

IndraMotion MTX 14VRS

Manuel de programmation

Référence
R911381559

Édition 04



Titre IndraMotion
MTX 14VRS
Manuel de programmation

Type de documentation Référence

Code de type de documentation DOK-MTX***-NC**PRO*V14-RE04-FR-P

Référence interne du fichier RS-32b2b2ff1c3483ba0a6846a50090ad76-4-fr-FR-6

Historique des modifications

Édition	Version	Remarque
Édition 04	2017-07	Extension de la syntaxe de langage standard pour ASUP interruptibles : ASPENA, ASPDIS Extension des notions fondamentales (ordres de saut) : RetTo – Saut en arrière depuis un sous-programme vers le label dans le programme principal et le sous-programme appelant

Mentions légales © Bosch Rexroth AG 2017

Tous droits réservés par Bosch Rexroth AG, y compris en cas de dépôt d'une demande de droit de propriété industrielle. Tout pouvoir de disposition, tel que droit de reproduction et de transfert, détenu par Bosch Rexroth.

Obligations Les données techniques fournies n'ont pour seul but que de décrire le produit ; elles ne sont pas à comprendre en tant que propriétés garanties au sens légal. Tous droits de modification de ce document et de disponibilité du matériel réservés.

Rédaction Développement de systèmes d'automatisation des fonctions de commande et CNC Motion, De (SyMu)

Table des matières

	Page
1 À propos de cette documentation.....	13
1.1 Validité de la documentation.....	13
1.2 Généralités.....	13
1.3 Documentation requise et documentation complémentaire MTX.....	15
1.3.1 Sélection/composition.....	15
1.3.2 Configuration.....	15
1.3.3 Mise en service.....	16
1.3.4 Utilisation.....	17
1.3.5 Développement OEM.....	18
1.3.6 Composants additionnels (addons).....	18
1.4 Présentation des informations.....	19
1.4.1 Consignes de sécurité.....	19
1.4.2 Symboles utilisés.....	19
1.4.3 Désignations et abréviations.....	20
1.5 Commentaires clients.....	20
2 Importantes consignes d'utilisation.....	21
2.1 Utilisation conforme aux spécifications.....	21
2.1.1 Introduction.....	21
2.1.2 Domaine d'application et de mise en œuvre.....	22
2.2 Utilisation non conforme aux spécifications.....	22
3 Notions de base pour la programmation CN.....	23
3.1 Introduction.....	23
3.2 Principes fondamentaux concernant la programmation standard et CPL.....	24
3.2.1 Possibilités de programmation.....	24
3.2.2 Identification des éléments CPL au sein d'un programme pièce.....	25
3.2.3 Éléments CPL au sein de la saisie manuelle, de la définition du bloc CN en mémoire tampon et de sous-programmes non liés.....	26
3.2.4 Respect des points d'interprétation entre les parties du langage CPL et standard !.....	26
3.3 Liaison de programmes CN.....	27
3.3.1 Table de liaison.....	27
3.3.2 Liaison lors de la sélection ou de l'exécution d'un programme.....	27
3.3.3 Liaison de sous-programmes – Liaison ultérieure.....	28
3.3.4 Influence du processus de liaison avec une identification DIN/CPL.....	29
3.4 Éléments de base d'un programme CN.....	29
3.4.1 Bloc-programme.....	29
3.4.2 Instructions.....	29
3.4.3 Conditions additionnelles.....	30
3.5 Mots de programme.....	31
3.5.1 Vue d'ensemble.....	31
3.5.2 Mots de programme provenant de fonctions CN.....	32
3.5.3 Mots de programme en tant que paramètres.....	35
3.5.4 Attributs de programmation.....	36

Table des matières

	Page
3.5.5	Utilisation des caractères de séparation entre 2 mots partiels..... 37
3.6	Fin du programme..... 38
3.7	Effet des mots de programme..... 39
3.7.1	Modal..... 39
3.7.2	Non modal..... 40
3.8	Éléments particuliers de configuration du programme..... 40
3.8.1	Identificateur de canal..... 40
3.8.2	Numéros de blocs..... 40
3.8.3	Lignes vides dans le code de programme..... 41
3.8.4	Commentaire dans un programme pièce..... 41
3.8.5	Remarques dans l'interface utilisateur..... 43
3.8.6	Sauts dans le déroulement du programme..... 44
3.9	Sous-programmes..... 45
3.9.1	Vue d'ensemble..... 45
3.9.2	Appel d'un sous-programme avec adresse P..... 46
3.9.3	Appel d'un sous-programme sans adresse P..... 47
3.9.4	Sous-programmes locaux..... 48
3.9.5	Appels de sous-programmes autodéfinis avec les codes G et M..... 51
3.9.6	Appels de sous-programmes modaux autodéfinis..... 51
3.9.7	Appel de sous-programme dans CPL via l'ordre CALL..... 53
3.9.8	Transmission de paramètres aux sous-programmes..... 54
3.9.9	Configuration et élimination des erreurs..... 61
3.10	Programmation de macros..... 62
3.10.1	Vue d'ensemble..... 62
3.10.2	Macros avec des paramètres..... 63
3.11	Programmation label et instructions de saut..... 64
3.11.1	Vue d'ensemble..... 64
3.11.2	Labels dans les blocs CN standards et blocs CPL..... 65
3.11.3	Saut en avant GoAhead (GOA) vers un bloc CN standard..... 65
3.11.4	Saut en arrière GoBack (GOB) vers un bloc CN standard..... 66
3.11.5	Saut conditionnel GoCond (GOC) vers un bloc CN standard..... 67
3.11.6	Saut inconditionnel GoTo vers un bloc CN standard..... 68
3.11.7	Saut CPL (GOTO) vers un bloc-programme quelconque..... 69
3.11.8	Saut en arrière RetTo dans un sous-programme vers un bloc-programme quelconque..... 69
3.12	Instructions de décision et de branchement..... 71
3.12.1	Vue d'ensemble..... 71
3.12.2	Fonction "Sauter séquence"..... 71
3.12.3	Instruction CPL : CASE-LABEL...LABEL-OTHERWISE-ENDCASE..... 72
3.12.4	Instruction CPL : IF-THEN-ELSE-ENDIF..... 73
3.13	Instructions de répétition..... 74
3.13.1	Vue d'ensemble..... 74
3.13.2	Instruction CPL : FOR-STEP-TO-NEXT..... 74
3.13.3	Instruction CPL : REPEAT-UNTIL..... 75
3.13.4	Instruction CPL : WHILE-DO-END..... 75
3.14	Programmation de variables..... 76
3.14.1	Noms des variables..... 76

Table des matières

	Page
3.14.2	Groupes de variables..... 77
3.14.3	Types de variables..... 82
3.14.4	Variables en programmation CN standard..... 84
3.14.5	Variable ERRNO pour l'analyse des erreurs des fonctions CPL "ERRNO"..... 85
3.15	Informations générales relatives au système..... 87
3.16	Mots d'ordre réservés..... 87
3.17	Instruction CPL..... 89
3.17.1	Attribution de valeur..... 89
3.17.2	Opérations mathématiques..... 90
3.17.3	Liaisons logiques..... 93
3.17.4	Conversion entre systèmes de nombres..... 94
3.17.5	Opérations de comparaison" =, >=, >, <>, <=, <"..... 94
3.18	Autres éléments de base CPL..... 95
3.18.1	Constantes..... 95
3.18.2	Caractères clés..... 95
3.19	Ordres de synchronisation du bloc CN 96
3.19.1	Vue d'ensemble..... 96
3.19.2	Fonctions de synchronisation de la préparation des blocs..... 96
3.19.3	Fonctions de synchronisation lors de la préparation des blocs..... 100
3.19.4	Écriture d'une date système lors de l'exécution..... 112
4	Aperçu des groupes modaux de fonctions CN..... 115
5	Fonctions CN avec syntaxe conforme DIN 66025 (extensions incl.)..... 119
5.1	Vue d'ensemble..... 119
5.2	Codes G..... 121
5.2.1	Interpolation linéaire en avance rapide "G0"..... 121
5.2.2	Interpolation linéaire en avance"G1"..... 122
5.2.3	Interpolation circulaire/hélicoïdale/hélicoïdale N"G2, G3"..... 124
5.2.4	Temporisation"G4"..... 130
5.2.5	Entrée tangentielle dans le cercle "G5"..... 131
5.2.6	Programmation de spline "G6"..... 131
5.2.7	Pente de la trajectoire ACTIVÉE "G8", pente de la trajectoire DÉSACTIVÉE "G9"..... 140
5.2.8	Profil de vitesse avec limitation des secousses "G8(SHAPE...)", "G9(SHAPE...)", "G9(ASHAPE...)", "G9(X... , Y... , ...)", "AsynchrShapeOrder, ASO"..... 141
5.2.9	Aucun plan "G16"..... 145
5.2.10	Changement de plan"G17, G18, G19"..... 146
5.2.11	Changement de plan avancé "G17(...), G18(...), G19(...)"..... 147
5.2.12	Libre sélection du plan (indépendamment du WCS) "G20"..... 149
5.2.13	Programmation du tracé de contour "G31"..... 149
5.2.14	Filetage"G33"..... 152
5.2.15	Correction de trajectoire de la fraise"G40, G41, G42"..... 155
5.2.16	Raccords entre segments de contour pour la correction de trajectoire de la fraise : arc de cer- cle "G43", point d'intersection "G44" arc de cercle..... 158
5.2.17	Correction de l'avance : Point de pénétration de la fraise "G45", centre de la fraise "G46"..... 159
5.2.18	Correction de la longueur d'outil "G47, G48"..... 160

Table des matières

	Page
5.2.19	Décalage d'origine programmable "G52"..... 162
5.2.20	Décalages d'origine (DO) "G53", "G53.1-G59.1" à "G53.5-G59.5"..... 164
5.2.21	Arrêt précis ACTIVÉ/DÉSACTIVÉ "G61, G62"..... 166
5.2.22	Taraudage sans mandrin de compensation "G63"..... 168
5.2.23	Programmation en pouces "G70", programmation locale en pouces "INCH(...)"..... 170
5.2.24	Programmation métrique "G71"..... 170
5.2.25	Approche des coordonnées du point de référence "G74"..... 171
5.2.26	Approche du point de référence "G74(HOME)"..... 172
5.2.27	Déplacement contre le palpeur de mesure "G75"..... 173
5.2.28	Approche de la position fixe des axes machine "G76"..... 174
5.2.29	Sous-programmes asynchrones : repositionnement de coordonnées individuelles "G77"..... 175
5.2.30	Commutation de correction ACTIVÉE "G78", commutation de correction DÉSACTIVÉE "G79".... 175
5.2.31	Programmation absolue "G90", programmation relative "G91", programmation absolue locale "AC(...)", programmation relative locale "IC(...)"..... 177
5.2.32	Programmation de temps "G93"..... 178
5.2.33	Programmation de l'avance (par min) "G94"..... 179
5.2.34	Programmation absolue de la vitesse "G94(LINS)" avec adaptation de la vitesse proportionnelle à la course..... 180
5.2.35	Programmation absolue de la vitesse "G94(LINT)" avec adaptation de la vitesse proportionnelle au temps..... 181
5.2.36	Programmation incrémentale de la vitesse "G94(...)" avec adaptation de l'accélération..... 182
5.2.37	Programmation absolue de la vitesse "G94(...)" avec adaptation de l'accélération..... 184
5.2.38	Programmation de l'avance (par rotation) "G95"..... 186
5.2.39	Programmation incrémentale de la vitesse "G95(...)" avec adaptation de l'accélération..... 187
5.2.40	Programmation de l'avance par rotation d'axe rond "G95(FRA=<axe>)"..... 189
5.2.41	Vitesse de coupe constante "G96", programmation directe de la vitesse de rotation "G97"..... 190
5.2.42	Correction 3D du rayon d'outil "G140, G141, G142"..... 194
5.2.43	Placement : plan incliné "G151 à G159.5"..... 196
5.3	Codes M..... 204
5.3.1	Interruption du programme (arrêt du programme) "M0, M00"..... 204
5.3.2	Interruption conditionnelle du programme (arrêt cond. progr.) "M1, M01"..... 204
5.3.3	Terminer le programme (fin du programme) "M2, M02, M30"..... 204
5.3.4	Rotation à droite de la broche "M3, M103, M203", rotation à droite de la broche et liquide de refroidissement ACTIVÉS "M13, M113, M213"..... 205
5.3.5	Rotation à gauche de la broche "M4, M104, M204", rotation à gauche de la broche et liquide de refroidissement ACTIVÉS "M14, M114, M214"..... 206
5.3.6	Arrêt de la broche "M5, M105, M205"..... 207
5.3.7	Orientation de la broche/Positionnement de la broche "M19, M119, M219"..... 208
5.3.8	Sélection automatique du rapport de réduction "M40, M140, M240"..... 210
5.3.9	Sélection manuelle du rapport de réduction "M41...44, M141...144, M241...244"..... 210
5.3.10	Débrayer le rapport de réduction "M48, M148, M248"..... 211
5.4	Avance et programmation de la vitesse de rotation..... 212
5.4.1	Définition de la vitesse de rotation ou de la vitesse modulo "<axe>=S<valeur>"..... 212
5.4.2	Adresse F "F"..... 213
5.4.3	Adresse F2 "F2"..... 214
5.4.4	Vitesse des axes asynchrones "FA"..... 215
5.4.5	Adresse "oméga" (avance)..... 215

Table des matières

	Page
5.4.6	Programmation de la vitesse de rotation de la broche "S, SSPG"..... 215
5.5	Correction d'outil..... 217
5.5.1	Correction D "D"..... 217
5.5.2	Correction ED "ED"..... 218
6	Fonctions CN avec syntaxe de langage standard..... 221
6.1	Vue d'ensemble..... 221
6.2	Programmation des facteurs du rayon d'action "ActRadFact, ARF"..... 222
6.2.1	Effet..... 222
6.2.2	Programmation..... 222
6.3	Comparaison des fonctions CN modales "Adjust, ADJ"..... 223
6.4	Signal programmable selon la distance..... 226
6.4.1	Généralités..... 226
6.4.2	Activation de la recherche d'une marque "ADSENA"..... 226
6.4.3	Désactivation de la recherche d'une marque "ADSDIS"..... 227
6.4.4	Définition d'une marque "ADSLAB"..... 227
6.4.5	Réinitialisation d'un signal de commutation "ADSCLR"..... 228
6.5	Extension du bloc-programme avant Retrace "AdvanceAppend, ADVA"..... 229
6.6	Remplacement du bloc-programme avant Retrace "AdvanceSubstitute, ADVS"..... 229
6.7	Surveillance de zone "Area, ARA"..... 230
6.8	Sous-programmes asynchrones..... 236
6.8.1	Sous-programmes asynchrones : désinscription "ASPCLR"..... 236
6.8.2	Sous-programmes asynchrones : mise hors service "ASPDIS"..... 236
6.8.3	Sous-programmes asynchrones : mise en service "ASPENA"..... 237
6.8.4	Sous-programmes asynchrones : Définir le point de redémarrage "ASPRTP"..... 238
6.8.5	Sous-programmes asynchrones : inscription "ASPSET"..... 238
6.8.6	Sous-programmes asynchrones : déclenchement par un programme "ASPSTA"..... 239
6.9	Attribution d'un nom d'axe logique "AssLogName, ALN"..... 240
6.10	Calibrage des cinématiques d'axe..... 241
6.10.1	Transformation en arrière "ATBWD"..... 241
6.10.2	Optimisation des paramètres "ATCAL"..... 241
6.10.3	Transformation en avant "ATFWD"..... 243
6.10.4	Lecture des paramètres de la commande numérique "ATGET"..... 244
6.10.5	Écriture de paramètres dans la commande numérique "ATPUT"..... 246
6.11	Exécution des fonctions auxiliaires actives de tous les groupes "AUXFUNC"..... 247
6.12	Modification de l'accélération maximale des axes "AxAcc, AAC", enregistrement temporaire de l'accélération maximale des axes "AxAccSave, AAS"..... 248
6.13	Chargement du tableau de correction des erreurs d'axe "AxCompTab"..... 250
6.14	Couplage d'axes "AxCouple, AXC"..... 252
6.15	Couplage d'avance d'axes "AxisFeedCouple, AFC"..... 256
6.16	Modification de la secousse maximale des axes "AxJerk, AJK", enregistrement temporaire de la secousse maximale des axes "AxJerkSave, AJS"..... 259
6.17	Mise hors service du mode axe C pour les broches "AxisToSpindle, ATS"..... 261
6.18	Mise hors service du mode axe C pour les broches et attente "AxisToSpindleWait, ATSW"..... 261
6.19	Remplacement d'axes "AxisReplace, AXR "..... 262

Table des matières

		Page
6.20	Modification de la vitesse maximale des axes "AxVel, AVE", enregistrement temporaire de la vitesse maximale des axes "AxVelSave, AVS".....	263
6.21	Paramètres de l'approximation spline B "BsaPar, BAP".....	264
6.22	Placement : correction de la position de la pièce à usiner "BcsCorr, BCR".....	265
6.23	Surveillance du bloc suivant en cas de définition du bloc CN en mémoire tampon "ChkNxtBlk, CNB".....	267
6.24	Programmation de chanfreins "ChLength", "CHL", "CHLL", "ChSection", "CHS", "CHSL".....	267
6.25	Détection automatique des coins et des droites "CLD".....	269
6.26	Surveillance de collision "Collision, CLN".....	271
6.27	Configuration de l'interpolation G0 "ConfG0, CG0".....	273
6.28	Sélection d'une transformation d'axe "Coord, CRD".....	274
6.29	Pointes d'angles intérieurs "Cornering, COR".....	275
6.30	Table de couplage spline "CoupleSplineTab, CST".....	277
6.31	Interpolation des paramètres des courbes "CurveParamInterpolation, CPI".....	279
6.32	Activation des tables de correction D "DcTSel, DCS".....	279
6.33	Acceptation des réglages d'axe issus des paramètres machine "DefAxis, DAX".....	280
6.34	Réinitialisation des broches du canal aux paramètres machine "DefSpindle, DSP".....	281
6.35	Enchaînement de blocs sans réduction de la vitesse "DefTangTrans, DTT".....	281
6.36	Programmation du diamètre "DiaProg, DIA", programmation du rayon "RadProg, RAD".....	282
6.37	Réaction temporisée en cas d'erreurs d'entraînement "DERDelay".....	283
6.38	Dimension du plan de travail "DimPlane, DimP".....	284
6.39	Réglage en hauteur pour la numérisation "DistCtrl, DCR".....	285
6.40	Déclenchement et arrêt de l'enregistrement du diagnostic "DREC_START", "DREC_STOP".....	287
6.41	Couplage de position finale "EndPosCouple, EPC".....	287
6.42	Génération de l'avance : masquage d'axes "FeedAd, FAD".....	289
6.43	Anticipation "FeedForward, FFW".....	290
6.44	Anticipation pour axes asynchrones "FFWAxis, FFWA".....	291
6.45	Génération de l'avance : groupe d'avance "FeedGrp, FDG".....	292
6.46	Génération de l'avance : type de groupe d'avance "FeedGrpType, FDGT".....	294
6.47	Mesure au vol "FlyMeas, FME".....	294
6.48	Mesure en butée fixe "FsProbe, FSP".....	296
6.49	Déplacement sur butée fixe "FsMove, FSM", "FsTorque, FST", "FsReset, FSR".....	298
6.50	Compensation des erreurs de rectitude et d'angle "GeoComp, GCT".....	301
6.51	Reprise d'axe "GetAxis, GAX".....	301
6.52	Création de broches de canal "GetSpindle, GSP".....	302
6.53	Changement de bloc au vol via le signal à grande vitesse "HsBlkSwitch, HSB".....	303
6.54	Changement de bloc avec interruption via le signal à grande vitesse "HsBlkSwitch(..,HSSTOP=..), HSB(..,HSSTOP=..)".....	306
6.55	Correction en ligne dans les coordonnées de la pièce à usiner "HWOC", "HWOCDIS".....	307
6.56	Initialisation de la mesure au vol "InitMeas, IME".....	308
6.57	Programmation du temps inversé "InvTime, ITIM".....	309
6.58	Présélection de la fenêtre d'arrêt précis "IPS1, IPS2, IPS3".....	309
6.59	Détermination des axes système pour la définition du filtre SHAPE optimal avec une limitation des secousses active "JerkAxlList, JAL".....	310
6.60	Activation de la limitation des secousses "JerkControl, JKC".....	312
6.61	Mode Jog dans les coordonnées de la pièce à usiner "JogWCSSelect".....	312
6.62	Suppression de la répartition cinématique des blocs CN "KinSplitDisable, KSD".....	313

Table des matières

	Page
6.63	Programmation KV "KvProg, KVP"..... 313
6.64	Compensation de poursuite "LCP"..... 315
6.65	Répartition du bloc de déplacement : longueur du segment de parcours "LEN"..... 315
6.66	Réglage de la puissance du laser en fonction de la vitesse sur trajectoire "LFP", "LFConf, LFC"..... 317
6.67	Commutation de la programmation linéaire/spline "LinearToSpline, LTS"..... 319
6.68	Prise d'origine de l'axe modulo linéaire "LinModZp, LMZ"..... 320
6.69	Profils de vitesse (fonctions rampe)..... 321
6.70	Changement de broche principale "MainSp, MSP"..... 323
6.71	Aide à la saisie : fonction miroir "Mirror(...), MIR(...)"..... 324
6.72	Mesure sur un contour (MOC)..... 326
6.73	Calcul modulo pour des axes sans fin à déplacement incrémental "Modulo, MOD"..... 329
6.74	Déplacement modulo pour les axes sans fin "ModuloShift, MODS"..... 330
6.75	Modification du palpeur de mesure de la configuration des flancs (MPEDGE)..... 331
6.76	Grignotage "Nibble, NIB"..... 332
6.77	Répartition du bloc de déplacement : nombre de segments de parcours "NUM"..... 334
6.78	Programmation de l'orientation "O()", ROTAX()", "phi, thème, psi"..... 334
6.78.1	Description générale..... 334
6.78.2	Mouvement d'orientation linéaire avec programmation des axes..... 335
6.78.3	Orientation linéaire avec programmation de coordonnées..... 336
6.78.4	Orientation du tenseur..... 338
6.78.5	Orientation vectorielle..... 342
6.79	Avance 100 % "OvrDis, OVD", "OvrEna, OVE"..... 344
6.80	Modification de l'accélération maximale sur trajectoire "PathAcc, PAC"..... 345
6.81	Activation des tableaux de placement "PmTSel, PMS"..... 346
6.82	Programmation des coordonnées polaires..... 347
6.82.1	Définition du pôle "PolarPol, POP"..... 347
6.82.2	Programmation des coordonnées polaires..... 348
6.83	Définition du point d'inversion miroir/de rotation "PoleSet, PLS"..... 351
6.84	Sortie à grande vitesse dépendant de la position "PosDepHSOut, PHS"..... 352
6.85	Mode de positionnement pour les axes sans fin "PosMode, PMD", mode de positionnement local pour les axes sans fin "DC", "ACP", "ACN"..... 354
6.86	Programmation de précision "PrecProg, PRP"..... 356
6.87	Traitement du programme en fonction de la préparation des blocs "PREP"..... 358
6.88	Temps de déclenchement de la course..... 360
6.88.1	Temps de déclenchement de la course (point d'arrivée de l'interpolation) "PtBlkEnd, PTE"..... 360
6.88.2	Temps de déclenchement de la course (restauration des valeurs par défaut) "PtDefault, PTD"..... 362
6.88.3	Temps de déclenchement de la course (fenêtre Inpos) "PtInpos, PTI"..... 363
6.89	Mouvement de déplacement PTP "PtpMove, PTP"..... 364
6.90	Poinçonnage "Punch, PUN"..... 366
6.91	Accélération radiale maximale "RadialAcc, RAC"..... 369
6.92	Enregistrement du vecteur de retrait en tant que donnée système "RecordRevVec, RRV"..... 369
6.93	Réduction de couple "RedTorque, RDT"..... 370
6.94	Poursuite du programme avec avance de bloc après interruption du programme "ReentContBlk, RCB, RCBD"..... 371
6.95	Suppression d'un axe d'un groupe d'axes "RemAxis, RAX"..... 373
6.96	Suppression d'un nom d'axe logique "RemLogName, RLN"..... 373

Table des matières

	Page
6.97 Retrait de broches du canal "RemSpindle, RSP".....	374
6.98 Répétition de bloc "REPEAT, RPT".....	375
6.99 Sous-programmes asynchrones.....	375
6.99.1 Définition du comportement de redémarrage "REPOSDEF".....	375
6.99.2 Définition du point de redémarrage dans un sous-programme asynchrone "REPOSTP".....	376
6.100 Recul Retrace, RETRACE.....	377
6.101 Extension du bloc-programme arrière Retrace "ReverseAppend, REVA".....	378
6.102 Remplacement du bloc-programme arrière Retrace "ReverseSubstitute, REVS".....	379
6.103 Aide à la saisie : rotation "Rotate(...), ROT(...)".....	380
6.104 Arrondissage des angles.....	382
6.104.1 Arrondissage des angles avec indication de la différence "RoundEps, RNE".....	382
6.104.2 Arrondissage des angles avec indication du rayon "Rounding", "RND", "RNDL".....	383
6.105 Synchronisation du couplage des axes système "SACSYNC".....	384
6.106 Aide à la saisie : mise à l'échelle "Scale(...), SCL(...)".....	387
6.107 Couplage sélectif additif de coordonnées.....	389
6.107.1 Couplage sélectif additif de coordonnées "SelCrdCouple, SCC".....	389
6.107.2 Couplage sélectif additif de coordonnées avec table "SelCrdCoupleTab, SCCT".....	390
6.108 Suppression de la surveillance de collision "SCLN".....	391
6.109 Initialisation de la position du programme "SetPos, SPS".....	392
6.110 Décalage programmé du contour "Shift, SHT".....	392
6.111 Limitation de la vitesse de rotation "SMin, SMN", "SMax, SMX".....	393
6.112 Comparaison des broches dans le programme de comparaison d'initialisation de l'avance de bloc "SpAdjSp".....	394
6.113 Détermination de la syntaxe de broche pour le programme de comparaison de l'avance de bloc "SpAdjStr".....	395
6.114 Libération/Reprise d'une broche réservée "SpAdmin, SPA".....	396
6.115 Définition (activation) d'un groupe de couplage, dissociation (désactivation) d'un groupe de couplage "SpCoupleConfig, SPCC".....	396
6.116 Écart de coulage de la broche esclave "SpCoupleDist, SPCD".....	398
6.117 Fenêtre du mode synchrone "SpCoupleErrWin, SPCE".....	399
6.118 Décalage angulaire avec un couplage actif "SpCouplePosOffs, SPCP".....	399
6.119 Attente d'un décalage angulaire "SpCouplePosOffs_Wait, SPCP_WAIT".....	400
6.120 Fenêtre du mode synchrone "SpCoupleSyncWin, SPCS".....	401
6.121 Attente du mode synchrone "SpCouple_Wait, SPC_WAIT".....	401
6.122 Définition/Dissociation de groupes de broches "SPG.., SPGALL".....	402
6.123 Écriture dans une queue de données système "SSDQ".....	403
6.124 Programmation des broches du système "SSp...".....	405
6.124.1 Généralités.....	405
6.124.2 Libération/Reprise d'une broche réservée "SSpAdm".....	406
6.124.3 Désactivation de la gestion de mouvement des broches pour certains canaux "SSpAdmOff".....	406
6.124.4 Changement du rapport de réduction "SSpGear".....	406
6.124.5 Limitation de la vitesse de rotation maximale "SSpMax".....	407
6.124.6 Limitation de la vitesse de rotation minimale "SSpMin".....	408
6.124.7 Commutation de l'interface de position/vitesse de rotation "SSpMode".....	408
6.124.8 Programmation de mouvement "SSpMove (SSPM)".....	408
6.124.9 Orientation de la broche "SSpOri (SSPO)".....	409
6.124.10 Programmation de la vitesse de rotation "SSpSpeed (SSPS)".....	409

Table des matières

	Page
6.124.11 Limitations de la broche spécifiques à l'outil "SSpToolLim".....	410
6.125 Activation du mode axe C pour les broches "SpindleToAxis, STA".....	411
6.126 Activer le mode axe C pour des broches et attendre "SpindleToAxisWait, STAW".....	411
6.127 Attente d'une programmation de la broche à la fin du bloc "SpindleWait, SPWAIT".....	412
6.128 Sélection d'une stratégie d'attente "SpindleWaitMode, SWM".....	413
6.129 Arrondissage des angles avec des splines.....	414
6.129.1 Arrondissage des angles avec des splines (modal) "SplineCornering, SCO".....	414
6.129.2 Arrondissage des angles avec des splines (local) "SplineCorneringLocal, SCOL".....	415
6.130 Définition du type de spline "SplineDef, SPD".....	415
6.131 Répartition du parcours programmable "Split, SPT".....	417
6.132 Broche : commutation de l'interface de position/vitesse de rotation "SpMode, SPM".....	418
6.133 Paramétrage de l'orientation statique de l'outil "StatToolOri, STO".....	419
6.134 Neutralisation des types de fonctionnement à étape unique "StepModeDisable, SMD".....	420
6.135 Limitations de la broche spécifiques à l'outil "SToolLim".....	423
6.136 Déplacement dans le TCS "TCM(, ,)".....	424
6.137 Guidage tangentiel de l'outil "TangTool, TTL".....	424
6.138 Orientation tangentielle de l'outil "TangToolOri, TTO".....	428
6.139 Remontée d'outil du taraudage "TappRet1, TappRet2".....	430
6.140 Sélection d'une broche pour le taraudage sans mandrin de compensation "TappSp, TSP".....	431
6.141 Comparaison des configurations d'axe "TargetAxisConf, TAX".....	432
6.142 Définition du TCS dans les coordonnées de programme "TcsDef, TCS".....	433
6.143 Fonctions additionnelles pour le filetage "ThreadSet, TST".....	435
6.143.1 Généralités.....	435
6.143.2 Configuration des données de remontée ThreadSet(...).....	435
6.143.3 Configuration de la dynamique ThreadSet(DYN).....	436
6.143.4 Libération de la remontée d'outil ThreadSet(ROn).....	436
6.143.5 Commutation du mode de fonctionnement de broche ThreadSet(SPC).....	437
6.143.6 Influence sur le signal IF du canal ThreadSet(TCI).....	437
6.144 Décalage des coordonnées du programme "Trans, TRS", décalage additif des coordonnées du programme "ATrans, ATR".....	437
6.145 Compensation volumétrique.....	439
6.145.1 Activation et désactivation "VCPEna".....	439
6.145.2 Chargement du tableau "VCPLoad".....	440
6.146 Entraînements virtuels "VirtAxisPos, VAP, VAPR".....	441
6.147 Diagnostic du guidage de vitesse "VREC_START", "VREC_STOP".....	441
6.148 Reprise d'axe et attente si nécessaire "WaitAxis, WAX".....	444
6.149 Écriture de paramètres Sercos avec un numéro d'identification Sercos élargi "Writeld3, WID3".....	445
6.149.1 Généralités.....	445
6.149.2 Écriture d'un paramètre dans un ou plusieurs entraînements (pour les broches et les axes asynchrones).....	446
6.149.3 Écriture d'un paramètre dans un ou plusieurs entraînements (pour les axes synchrones).....	446
6.149.4 Écriture de numéros d'identification élargis d'une liste ID dans un entraînement.....	447
6.149.5 Écriture des valeurs d'une liste de paramètres dans un entraînement.....	447
6.150 Activation des tables de décalage d'origine "ZoTSel, ZOS".....	448

Table des matières

	Page
7	Ordres CPL..... 449
7.1	Axes et coordonnées..... 449
7.1.1	Vue d'ensemble..... 449
7.1.2	Ordres de lecture des noms et propriétés..... 451
7.1.3	Ordres de lecture des positions de coordonnées et d'axes..... 457
7.1.4	Ordres de lecture des valeurs mesurées..... 464
7.2	Broches..... 467
7.2.1	Vue d'ensemble..... 467
7.2.2	Ordres de lecture des noms..... 467
7.3	Corrections et aides à la saisie..... 469
7.3.1	Décalages d'origine..... 469
7.3.2	Décalage de position et de contour..... 475
7.3.3	Correction de la position de la pièce à usiner et placements (plan incliné)..... 476
7.3.4	Aides à la saisie..... 483
7.3.5	Corrections d'outil..... 484
7.4	Données et processus de la commande..... 488
7.4.1	Version de la commande..... 488
7.4.2	Saisie du temps..... 488
7.4.3	Paramètres machine..... 491
7.4.4	Données système simples..... 492
7.4.5	Données système de types structurés..... 510
7.4.6	Interface Sercos..... 514
7.4.7	Interface API..... 520
7.4.8	Fonctions actives..... 523
7.4.9	Interface CN (NCS)..... 524
7.4.10	Tables XML générales..... 587
7.4.11	Base de données d'outils..... 589
7.4.12	Erreurs et avertissements..... 599
7.5	Fichiers et répertoires..... 605
7.5.1	Généralités..... 605
7.5.2	Répertoires..... 606
7.5.3	Opérations de fichier..... 609
7.5.4	Traitement d'un fichier..... 614
7.5.5	Propriétés du fichier..... 624
7.6	Traitement des chaînes de caractères..... 627
7.6.1	Notions de base..... 627
7.6.2	Conversion de type..... 629
7.6.3	Modification des chaînes de caractères..... 632
7.6.4	Exemples de programmation..... 639
7.7	Communication avec un client..... 641
7.7.1	MMC..... 641
8	Annexe..... 643
8.1	Tableau synoptique des fonctions CN..... 643
8.1.1	Classement par ordre alphanumérique selon le format long..... 643
8.2	Jeu de caractères ASCII..... 671

Table des matières

	Page
8.3 Codes de touches supplémentaires.....	674
8.4 Optimisation du temps d'usinage.....	675
8.4.1 Mesures ayant une influence directe sur la vitesse de traitement.....	675
8.4.2 Mesures ayant une influence indirecte sur la vitesse de traitement.....	677
8.4.3 Retard de la séquence dû à une programmation imprécise.....	680
 9 Service et assistance.....	 683
 Index.....	 685

À propos de cette documentation

1 À propos de cette documentation

1.1 Validité de la documentation

Vue d'ensemble des groupes cibles et des phases du produit

Les activités encadrées, les phases du produit et les groupes cibles figurant sur le graphique suivant se rapportent à la présente documentation.

Cette documentation permet au groupe cible "Programmeur" d'exécuter les activités "Optimiser et Tester" au cours de la phase du produit "Mise en service".

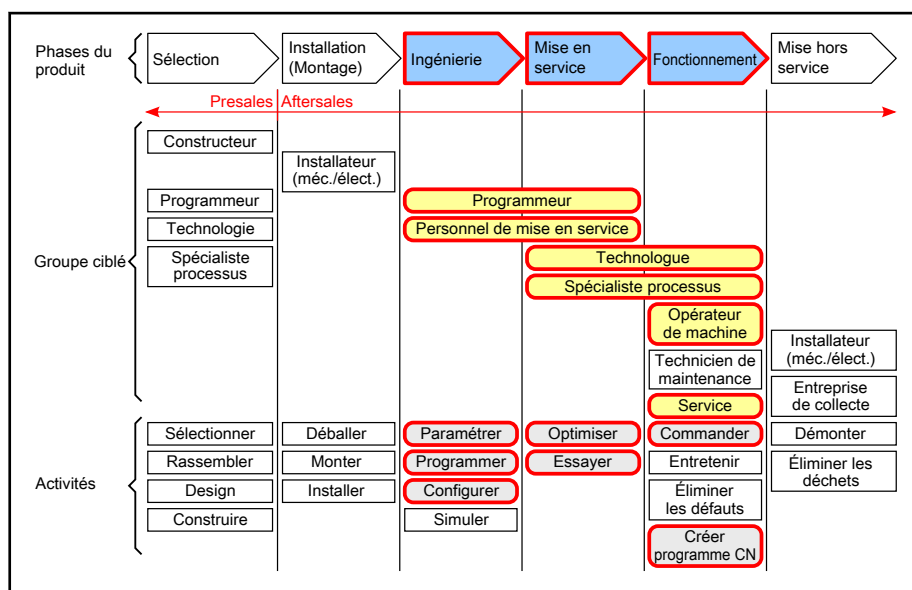


Fig. 1-1: Affectation de la présente documentation selon les phases du produit, les groupes cibles et les activités du groupe cible

Groupe cible

Cette documentation s'adresse à un utilisateur qui souhaite mettre en service une commande de type IndraMotion MTX.

Cette documentation fournit une aide à l'utilisateur au cours des phases

- Mise en service
- Utilisation

1.2 Généralités

Le présent manuel décrit la programmation standard de la commande Rexroth IndraMotion MTX. Outre les bases de la programmation CN, il décrit aussi bien l'application des fonctions CN selon la norme DIN 66025 que celle des fonctions CN avec syntaxe de langage standard et des fonctions CPL.

Ce manuel vous aide lors de l'application/activation de différentes fonctions de commande.

Un point essentiel repose sur la procédure requise lors du paramétrage.

Cela implique que le matériel complet (commande CNC, entraînements, API et alimentations en tension, câblage inclus) a déjà été intégré et que l'absence d'erreurs a été vérifiée conformément aux documents dont vous disposez sur l'application. En font aussi partie notamment tous les dispositifs de sécurité tels que les circuits d'ARRÊT D'URGENCE, contacts de validation, commutateurs de fins de course, etc. !

Dans l'idéal, la mise en service logicielle de tous les entraînements Sercos utilisés a déjà été réalisée, mais ce n'est pas une nécessité absolue puisque

À propos de cette documentation

la commande peut être configurée en téléchargeant les paramètres lors de la montée en régime des entraînements raccordés. Pour cela, il faut toutefois que les fichiers d'initialisation adaptés soient disponibles dans la commande.

Afin d'adapter les fichiers d'initialisation, vous devez disposer de la documentation appropriée (listes des réglages des paramètres Sercos de tous les entraînements, documentation relative aux entraînements, etc.).

Pour plus de clarté, toutes les fonctions sont attribuées à des domaines thématiques génériques (cf. table des matières).

Pour chaque fonction, vous apprendrez :

- son but ou son usage,
- les conditions préalables/restrictions d'utilisation,
- les influences possibles.

De nombreuses fonctions touchent plusieurs domaines à la fois (paramètres machine, fonctions CN pour la programmation, signaux d'interface, etc.). Par conséquent, il se peut qu'elles soient déjà décrites, de façon complète ou sous certains aspects seulement, dans d'autres manuels de ce produit.



Veuillez consulter également les autres manuels concernés.

Les manuels suivants complètent le présent manuel :

- Paramètres machine : "Rexroth IndraMotion MTX Paramètres machine"
- Fonctions CN et CPL : "Rexroth IndraMotion MTX Manuel de programmation"
- Signaux d'interface : "Rexroth IndraMotion MTX Interface API"

Gardez également à votre disposition la documentation relative à l'entraînement.

Vous devez par ailleurs disposer de bonnes connaissances sur :

- l'interface utilisateur standard de la commande,
- l'interface utilisateur Windows du panneau de configuration de l'ordinateur,
- les outils de configuration des paramètres machine.

Ces outils permettent de modifier la configuration de la commande. C'est pourquoi vous devez impérativement être familiarisé avec l'utilisation de tous les outils requis !

À propos de cette documentation

! ATTENTION

Un personnel insuffisamment ou non formé peut, par une utilisation incorrecte ou non conforme, causer des dommages graves sur la machine, une perte de données, voire des blessures aux personnes !

Par conséquent, le démarrage et l'utilisation de la machine, ainsi que les réglages ou modifications des paramètres de configuration ne doivent être effectués que par un personnel spécialisé suffisamment formé !

Ce personnel doit être en mesure de reconnaître et d'éviter les dangers susceptibles d'être provoqués par la modification des paramètres et plus généralement par l'équipement mécanique, électrique ou électronique.

Nous déclinons toute responsabilité en cas de dommages directs ou indirects causés par une utilisation incorrecte ou non conforme des pièces d'équipement.

1.3 Documentation requise et documentation complémentaire MTX

1.3.1 Sélection/composition

Titre de la documentation avec codification et référence du matériel

IndraMotion MTX 14VRS Systembeschreibung (IndraMotion MTX 14VRS Description du système) DOK-MTX***-SYS*DES*V14-PRRS-DE-P, R911342639 Cette documentation décrit le système de commande Rexroth IndraMotion MTX. Les versions, les caractéristiques techniques, les interfaces et la configuration des modules de commande y sont décrites.
IndraMotion MTX 14VRS SafeLogic Systemübersicht (IndraMotion MTX 14VRS Aperçu du système SafeLogic) DOK-MTX***-SL**SYS*V14-PRRS-DE-P, R911341697 Cette documentation décrit l'utilisation de la commande de sécurité SafeLogic dans le système IndraMotion MTX.

Tab. 1-1: Aperçu de la documentation MTX – Sélection/composition

1.3.2 Configuration

Titre de la documentation avec codification et référence du matériel

IndraMotion MTX 14VRS Maschinenparameter (IndraMotion MTX 14VRS Paramètres machine) DOK-MTX***-MA*PAR**V14-RERS-DE-P, R911342629 Cette documentation décrit la structure et l'adaptation des paramètres disponibles de la commande Rexroth IndraMotion MTX. Le fonctionnement et l'utilisation du configurateur CN y sont également décrits.
IndraMotion MTX 14VRS SPS-Interface (IndraMotion MTX 14VRS Interface API) DOK-MTX***-PLC*INT*V14-PRRS-DE-P, R911342621 Cette documentation décrit les signaux d'interface et les modules du programme pour l'API intégré.
IndraMotion MTX 14VRS Funktionsbeschreibung - Grundlagen (IndraMotion MTX 14VRS Description des fonctions – Notions fondamentales) DOK-MTX***-NC*F*BA*V14-RERS-DE-P, R911342625 Cette documentation décrit les fonctions de base de Rexroth IndraMotion MTX. Les principales étapes de mise en service et les fonctions de la commande sont expliquées sous forme de description et d'instructions de manipulation.

À propos de cette documentation

IndraMotion MTX 14VRS Funktionsbeschreibung - Erweiterung (IndraMotion MTX 14VRS Description des fonctions – Notions avancées)

DOK-MTX***-NC*F*EX*V14-RERS-DE-P, R911342627

Cette documentation décrit les fonctions avancées de la commande Rexroth IndraMotion MTX. Les principales étapes de mise en service et les fonctions de la commande sont expliquées sous forme de description et d'instructions de manipulation.

IndraMotion MTX 14VRS Funktionsbeschreibung - Sonderfunktionen (IndraMotion MTX 14VRS Description des fonctions – Fonctions spéciales)

DOK-MTX***-NC*F*SP*V14-RERS-DE-P, R911342623

Cette documentation décrit les fonctions spéciales de la commande Rexroth IndraMotion MTX. Les principales étapes de mise en service et les fonctions de la commande sont expliquées sous forme de description et d'instructions de manipulation.

IndraMotion MTX Freiformflächenbearbeitung (IndraMotion MTX Usinage des surfaces de formes libres)

DOK-MTX***-FREEFORM***-APRS-DE-P, R911341434

Cette documentation décrit le mode opératoire lors de l'usinage de surfaces de formes libres avec la commande MTX. Les programmes CNC générés à partir d'un système CAO/FAO servent de base à l'ensemble du processus d'usinage. Cette documentation donne une vue d'ensemble sur les thèmes suivants : description des fonctions MTX et CNC pour l'usinage de surfaces de formes libres, mode opératoire pour le paramétrage CN, conditions cadres pour la création de programmes CNC avec un logiciel de FAO.

IndraMotion MTX Konvertierung von MTX-Projekten (IndraMotion MTX Conversion de projets MTX)

DOK-MTX***-PROCONV****-PRRS-DE-P, R911342483

Cette documentation aide à convertir des projets IndraMotion MTX 1.x au format MTX 2G. Elle décrit le mode opératoire et les difficultés éventuellement rencontrées lors de la conversion.

Tab. 1-2: *Aperçu de la documentation MTX – Configuration*

1.3.3 Mise en service

Titre de la documentation avec codification et référence du matériel

IndraWorks 13VRS Simulation (IndraWorks 13VRS Simulation)

DOK-IWORKS-SIMU****V13-APRS-DE-P, R911339255

Cette documentation décrit les fonctions des composants de simulation View3D, du pupitre de commande virtuel et de la commande virtuelle ainsi que leur utilisation dans IndraWorks.

IndraMotion MTX 14VRS Inbetriebnahme (IndraMotion MTX 14VRS Mise en service)

DOK-MTX***-STARTUP*V14-CORS-DE-P, R911342619

Cette documentation décrit la mise en service de la commande Rexroth IndraMotion MTX. Elle donne une vue d'ensemble et décrit également la mise en service et la configuration des axes ainsi que l'interface utilisateur et les données API.

IndraWorks 14VRS Basis-Bibliotheken IndraLogic 2G (IndraWorks 14VRS Bibliothèques de base IndraLogic 2G)

DOK-IL*2G*-BASLIB**V14-LIRS-DE-P, R911343918

Ce manuel décrit les bibliothèques API qui couvrent l'ensemble du système.

IndraWorks 14VRS Feldbusse (IndraWorks 14VRS Bus de terrain)

DOK-IWORKS-FB*****V14-APRS-DE-P, R911341484

La documentation décrit les raccordements périphériques locaux et les circuits des bus de terrain pris en charge par les systèmes IndraLogic XLC, IndraMotion MLC et IndraMotion MTX. Cette documentation se concentre sur la configuration, le paramétrage, la mise en service et le diagnostic des différents raccordements périphériques. Elle constitue la base de l'aide en ligne.

À propos de cette documentation

IndraWorks 14VRS WinStudio (IndraWorks 14VRS WinStudio) DOK-IWORKS-WINSTUD*V14-APRS-DE-P, R911341584 Ce « Manuel de l'utilisateur et référence technique » doit vous aider à obtenir des résultats optimaux avec votre logiciel « Rexroth WinStudio™ ». Le document fournit des informations techniques et des instructions pas à pas pour la création de programmes HMI/SCADA compatibles Web.
IndraWorks 14VRS Software-Installation (IndraWorks 14VRS Installation du logiciel) DOK-IWORKS-SOFTINS*V14-CORS-DE-P, R911344285 Cette documentation décrit le processus d'installation du logiciel IndraWorks.
IndraWorks 14VRS Engineering (IndraWorks 14VRS Engineering) DOK-IWORKS-ENGINEE*V14-APRS-DE-P, R911343564 Cette documentation décrit l'utilisation du logiciel IndraWorks dans lequel sont intégrés les outils d'ingénierie Rexroth. Le travail avec IndraWorks et la commande de la fonction d'oscilloscope y sont décrits.
IndraWorks 12VRS FDT-Container (IndraWorks 12VRS Conteneur FDT) DOK-IWORKS-FDT*CON*V12-APRS-DE-P, R911334397 Cette documentation décrit les fonctionnalités du conteneur FDT d'IndraWorks. L'activation de ces fonctionnalités dans le projet et le travail avec les DTM y sont décrits.
IndraWorks 14VRS IndraLogic 2G SPS-Programmiersystem (IndraWorks 14VRS IndraLogic 2G Système de programmation API) DOK-IWORKS-IL2GPRO*V14-APRS-DE-P, R911343570 Cette documentation décrit l'outil de programmation API IndraLogic 2G et son utilisation. L'utilisation de base, les premières étapes, la visualisation, les points de menu et les éditeurs y sont décrits.
IndraWorks 14VRS HMI (IndraWorks 14VRS Interface HMI) DOK-IWORKS-HMI*****V14-APRS-DE-P, R911343567 Cette documentation décrit les fonctions, la configuration et la commande des interfaces utilisateurs IndraWorks HMI Engineering et IndraWorks HMI Operation.

Tab. 1-3: Aperçu de la documentation MTX – Mise en service

1.3.4 Utilisation

Titre de la documentation avec codification et référence du matériel

IndraMotion MTX 09VRS Werkstattprogrammierung Drehen und Fräsen (IndraMotion MTX 09VRS Programmation d'atelier pour le tournage et le fraisage) DOK-MTX***-SF*PROG*V09-AWxx-DE-P, R911324376 Cette documentation décrit l'utilisation et la programmation de la programmation CN graphique des technologies de tournage et de fraisage, y compris la simulation de la pièce à usiner.
IndraMotion MTX 12VRS Satzvorlauf (IndraMotion MTX 12VRS Avance de bloc) DOK-MTX***-BLK*RUN*V12-APRS-DE-P, R911334378 Cette documentation doit permettre au fabricant de la machine de configurer la fonction « Avance de bloc » sur sa machine pour l'utilisateur final.
IndraMotion MTX 14VRS Programmierhandbuch (IndraMotion MTX 14VRS Manuel de programmation) DOK-MTX***-NC**PRO*V14-RERS-DE-P, R911342633 Le présent manuel décrit la programmation standard de la commande Rexroth IndraMotion MTX. Outre les bases de la programmation CN, il décrit aussi bien l'application des fonctions CN selon la norme DIN 66025 que celle des fonctions CN avec syntaxe de langage standard et des fonctions CPL.

À propos de cette documentation

IndraMotion MTX 14VRS Standard-NC-Bedienung (IndraMotion MTX 14VRS Commande CN standard)

DOK-MTX***-NC*OP***V14-APRS-DE-P, R911342631

Cette documentation décrit la manipulation de l'interface utilisateur de la commande numérique Rexroth IndraMotion MTX. L'utilisation de l'interface, la création de programmes CN et la gestion des outils y sont décrites.

IndraMotion MTX 14VRS Standard-NC-Zyklen (IndraMotion MTX 14VRS Cycles CN standards)

DOK-MTX***-NC*CYC**V14-PRRS-DE-P, R911342637

Cette documentation décrit l'application des cycles standards des différentes technologies pour la commande Rexroth IndraMotion MTX.

xx

Édition actuelle correspondante

Tab. 1-4:

Aperçu de la documentation MTX – Utilisation

1.3.5 Développement OEM

Titre de la documentation avec codification et référence du matériel

IndraMotion MTX 13VRS Automation Interface (IndraMotion MTX 13VRS Interface d'automatisation)

DOK-MTX***-AUT*INT*V13-APRS-DE-P, R911337273

Cette documentation décrit l'accès basé sur des scripts aux données de projet IndraWorks via l'interface des interfaces d'automatisation. Les différents objets, y compris les exemples de code, y sont décrits. Une description de l'outil d'automatisation est jointe à cette documentation.

IndraMotion MTX 09VRS Integration von OEM-Applikationen (IndraMotion MTX 09VRS Intégration d'applications OEM)

DOK-MTX***-DEV*KIT*V09-AWxx-DE-P, R911324354

Cette documentation décrit l'intégration des applications OEM dans IndraWorks MTX et la communication via la norme industrielle OPC.

IndraMotion MTX 14VRS OPC-Kommunikation (IndraMotion MTX 14VRS Communication OPC)

DOK-MTX***-OPC*COM*V14-PRRS-DE-P, R911342635

Cette documentation décrit la syntaxe et la structure des éléments de communication avec les appareils Bosch Rexroth.

IndraMotion MTX 14VRS OPC UA Kommunikation (IndraMotion MTX 14VRS Communication OPC-UA)

DOK-MTX***-OPC*UA**V14-APRS-DE-P, R911342512

Cette documentation décrit la communication OPC-UA de la commande IndraMotion MTX.

xx

Édition actuelle correspondante

Tab. 1-5:

Aperçu de la documentation MTX – Développement OEM

1.3.6 Composants additionnels (addons)

Titre de la documentation avec codification et référence du matériel

IndraMotion MTX 11VRS Aktionsrekorder (IndraMotion MTX 11VRS Enregistreur d'actions)

DOK-MTX***-ACR*****V11-APxx-DE-P, R911329942

Cette documentation décrit l'enregistreur d'actions MTX. Sont non seulement décrites son installation et sa mise en service, signaux d'interface inclus, mais également son utilisation et sa commande.

IndraMotion MTX efficiency workbench 14VRS MTX cta, MTX ega (IndraMotion MTX Efficiency Workbench 14VRS MTX cta, MTX ega)

DOK-MTX***-EWB*****V14-APRS-DE-P, R911343915

Cette documentation décrit le mode de fonctionnement et le domaine d'application des outils d'analyse IndraMotion MTX cta et IndraMotion MTX ega.

À propos de cette documentation

IndraMotion MTX Remote Condition Monitoring (IndraMotion MTX Surveillance des conditions à distance)

DOK-MTX***-RCM*****V01-APRS-DE-P, R911334382

Cette documentation décrit l'utilisation du système de surveillance à distance des conditions.

IndraMotion MTX visIREC (IndraMotion MTX visIREC)

DOK-MTX***-VISIREC*V01-APRS-DE-P, R911344241

Cette documentation décrit l'outil d'analyse visIREC pouvant être utilisé pour optimiser le processus d'usinage de surfaces de formes libres. Affichage 2D ou 4D des données relatives à la trajectoire. Affichage 2D ou 4D des données relatives aux coordonnées. Analyse des domaines critiques (écart par rapport à la trajectoire et à l'orientation). Comparaison entre les jeux CN programmés et interpolés.

xx

Édition actuelle correspondante

Tab. 1-6:

Aperçu de la documentation MTX – Composants additionnels (ad-dons)

1.4 Présentation des informations

1.4.1 Consignes de sécurité

Dans la mesure où elles sont mentionnées dans la présente documentation des applications, les consignes de sécurité contiennent des mots signaux précis (danger, avertissement, attention, avis) et éventuellement aussi un panneau d'avertissement (selon ANSI Z535.6-2006).

Les mots signaux attirent l'attention sur la consigne de sécurité et identifient le niveau de dangerosité.

Le panneau d'avertissement (triangle d'avertissement avec un point d'exclamation) qui précède les mots signaux danger, avertissement et attention indique une mise en danger pour les personnes.

ATTENTION

Le non-respect d'une telle consigne de sécurité peut occasionner des blessures modérées ou légères.

AVIS

Le non-respect d'une telle consigne de sécurité peut entraîner des dégâts matériels.

1.4.2 Symboles utilisés

Remarque Les remarques sont représentées comme suit :



Il s'agit d'une remarque s'adressant à l'utilisateur.

Conseil

Les conseils sont représentés comme suit :



Il s'agit d'un conseil s'adressant à l'utilisateur.

À propos de cette documentation

1.4.3 Désignations et abréviations

Désignations et abréviations

Terme	Explication
IWE	IndraWorks Engineering
IWO	IndraWorks Operation
FO	Fibre optique
CN	Commande numérique
OEM	Original Equipment Manufacturer (fabricant d'équipement d'origine)
PROFIBUS	Liaison de communication
Sercos	Liaison de communication

Tab. 1-7: Désignations et abréviations utilisées

1.5 Commentaires clients

Nous accordons une grande valeur aux suggestions, souhaits et propositions d'amélioration de nos clients. Vous pouvez nous envoyer vos commentaires concernant cette documentation par e-mail à l'adresse Feedback.Documentation@boschrexroth.de. Vous pouvez saisir vos commentaires directement dans le document électronique au format PDF et nous renvoyer le fichier PDF ainsi commenté.

2 Importantes consignes d'utilisation

2.1 Utilisation conforme aux spécifications

2.1.1 Introduction

Les produits Rexroth sont conçus et fabriqués conformément à l'état de l'art applicable. Avant d'être livrés, ils sont soumis à un contrôle permettant de garantir leur sécurité de fonctionnement.

Les produits doivent impérativement être utilisés conformément aux spécifications. Toute utilisation non conforme aux spécifications risque d'entraîner des risques de dégâts matériels et de blessures.



En tant que fabricant, la société Bosch Rexroth décline toute responsabilité en cas de dommages résultant d'un emploi non conforme de ses produits. Ceci signifie que toute garantie ou demande de dommages-intérêts sera exclue en cas d'emploi incorrect des produits. Dans un tel cas, l'utilisateur sera seul responsable des risques encourus.

Avant d'utiliser les produits de la société Bosch Rexroth, il est indispensable de s'assurer que les conditions suivantes sont remplies afin de garantir l'utilisation conforme des produits :

- Toute personne amenée à manipuler l'un de nos produits, de quelque manière que ce soit, doit avoir lu et compris les consignes de sécurité correspondantes ainsi que l'utilisation conforme aux spécifications.
- S'il s'agit de matériel informatique, les produits en question doivent rester en leur état d'origine, c'est-à-dire qu'ils ne doivent subir aucune modification architecturale quelle qu'elle soit. Les logiciels ne doivent pas être décompilés et leurs codes sources ne doivent pas être modifiés.
- Les produits endommagés ou défectueux ne doivent en aucun cas être installés ou mis en service.
- Veiller à ce que les produits soient installés comme stipulé dans la documentation.



Afin d'écarter tout mouvement involontaire des axes, la justesse des données présentées par la commande ou saisies et lues par l'utilisateur doit être contrôlée avant de procéder à l'acceptation/activation. Il pourrait s'agir en particulier des données obsolètes ou non valides suivantes :

- Programmes pièces
- Tables de décalage d'origine
- Tables de correction
- Tables d'outils
- Variables CPL permanentes
- Données API rémanentes
- Données système permanentes

Importantes consignes d'utilisation

2.1.2 Domaine d'application et de mise en œuvre

La commande Rexroth IndraMotion MTX sert à :

- programmer les contours et une technologie d'usinage (avance sur trajectoire, vitesse de rotation de la broche, changement d'outil) d'une pièce à usiner ;
- guider un outil d'usinage le long d'une trajectoire programmée.

Les commandes d'avance, les broches et les axes auxiliaires d'une machine-outil sont contrôlés par l'intermédiaire d'une interface Sercos.



De plus, des composants E/S sont nécessaires pour l'API intégré qui, associé à la véritable CNC, commande l'ensemble du processus d'usinage et en surveille la sécurité d'un point de vue technique.

Les appareils doivent être utilisés exclusivement avec les configurations et combinaisons de composants matériels expressément indiquées, avec les logiciels et micrologiciels spécifiés et indiqués dans la documentation et les descriptions de fonctions correspondantes.

La commande Rexroth IndraMotion MTX constitue la solution système CNC idéale pour usiner et transformer selon les techniques suivantes :

- Tournage
- Fraisage
- Alésage
- Ponçage
- Cintrage
- Grignotage
- Poinçonnage
- Détourage
- Manipulation

2.2 Utilisation non conforme aux spécifications

Toute utilisation de la commande Rexroth IndraMotion MTX sortant du cadre des domaines d'application énumérés précédemment, exécutée dans d'autres conditions de service ou d'autres données techniques que celles indiquées dans la documentation correspondante est considérée comme "non conforme".

La commande Rexroth IndraMotion MTX ne doit en aucun cas être utilisée...

- si les conditions de service auxquelles elle est exposée ne correspondent pas aux conditions ambiantes prescrites. Il est, par exemple, absolument interdit de les utiliser sous l'eau, dans un environnement soumis à de très fortes variations de température ou encore avec des températures maximales extrêmes.
- dans des applications qui n'ont pas été autorisées expressément par Bosch Rexroth. À cet égard, les informations mentionnées dans les consignes de sécurité générales doivent impérativement être respectées !
- dans des installations ou sur des machines raccordées à Internet par une liaison réseau non sécurisée. Dans le cas contraire, des dysfonctionnements pourraient se produire suite à une panne de la commande due à un accès non autorisé.

3 Notions de base pour la programmation CN

3.1 Introduction

Une commande CN reçoit toutes les informations nécessaires au traitement d'une pièce à usiner sur une machine-outil par l'intermédiaire d'un programme CN (programme pièce).

La structure d'un tel programme CN peut varier, si bien qu'il est possible de traiter quasiment n'importe quelle pièce à usiner avec les technologies les plus diverses (fraisage, tournage, ponçage, etc.). Le programme pièce regroupe non seulement l'information de mouvement décrivant la trajectoire de l'outil par rapport à la pièce à usiner, mais également l'information sur la technologie.

L'**information sur le mouvement** est subdivisée en éléments de contour élémentaires individuels (droites, cercles, spirales, splines, nurbs, etc.).

Pour chacun de ces éléments de contour à géométrie simple, la commande peut exécuter les mouvements dans le cadre d'une seule étape de traitement si toutes les étapes de traitement sont définies dans le programme CN dans l'ordre correct et avec toutes les conditions aux limites requises. Les conditions aux limites requises se composent notamment des fonctions inhérentes à la technologie (vitesses, vitesses de rotation, etc.) et des fonctions auxiliaires des machines (liquide de refroidissement, serrage de l'axe, etc.).



Les directives fondamentales concernant la structure d'un programme CN sont stipulées dans la norme DIN 66025.

Le contenu de la norme DIN 66025 "Structure de programme pour machines à commande numérique" (parties 1 et 2) correspond à celui des normes internationales ISO/DIS 6983 et ISO/DP 6983 "Numerical control of machines".

IndraMotion MTX gère les programmes CN dans le "système de fichiers" de la commande. De plus, il est possible de connecter des lecteurs externes pour exécuter des programmes directement à partir de ces derniers.

Vous trouverez de plus amples informations sur le **système de fichiers** et les **droits d'accès**, ainsi que sur la création et l'édition de programmes pièces dans le mode d'emploi de la commande MTX.

Le nom de tout programme CN doit répondre aux conventions générales applicables aux noms de fichiers.

Les noms de programmes CN ne doivent pas dépasser une longueur maximale de 28 caractères, extension de fichier comprise. Ils peuvent se composer de lettres, de chiffres ainsi que des caractères spéciaux "." et "_".

- Par défaut, la commande MTX utilise l'extension de fichier ".npg" pour les programmes CN.
- En interne, aucune différence n'est faite entre le nom de fichier et l'extension de fichier.
- Les noms de fichiers sont généralement sensibles à la casse.
- Les désignateurs "." et ".." ne sont pas autorisés en tant que noms de fichier.
- Au sein d'un répertoire, les noms de fichiers doivent être univoques. Il est cependant possible d'attribuer le même nom à deux fichiers enregistrés dans des répertoires différents.

Notions de base pour la programmation CN

- Dans certains cas exceptionnels (dans la mesure où le programme concerné n'est jamais lié), il est également possible d'utiliser des noms de fichiers comptant jusqu'à 30 caractères.

3.2 Principes fondamentaux concernant la programmation standard et CPL

3.2.1 Possibilités de programmation

La commande offre deux possibilités de programmation :

- la programmation standard ou DIN,
- la programmation CPL (CPL : **C**ustomer **P**rogramming **L**anguage ou langage de programmation client).

En **programmation standard ou DIN**, vous décrivez la séquence de mouvements et ses conditions aux limites (géométrie, cinématique, dynamique, corrections, etc.). La programmation standard représente un langage d'ordres pur pour la commande de mouvements sur la machine et pour le pilotage des fonctions spécifiques de la machine.

La syntaxe de la commande IndraMotion MTX se compose d'ordres déterminés au sein de la norme DIN 66025 (codes G et M) ainsi que d'extensions essentielles dans le domaine des codes G et des éléments de syntaxe supplémentaires similaires au langage standard.

Les éléments de base de la programmation standard sont des fonctions dites CN, à chacune desquelles est affectée une syntaxe de programmation.

Il est possible d'affecter des paramètres supplémentaires à une fonction CN afin de paramétrer la fonction en question.

Exemple :

Fonction CN : G2 Interpolation circulaire antitrigonométrique

Paramètres : I, J, K, R Coordonnées du centre, rayon

La **programmation CPL** s'oriente d'après la norme du langage standard BASIC, mais elle contient également des éléments de structure similaires à PASCAL. De ce fait, elle s'apprend facilement.

Le langage CPL est un véritable langage de programmation et constitue de ce fait une extension des possibilités de programmation. Il comprend des éléments de programme qui ne se focalisent pas sur la commande de la machine, permettant ainsi d'accéder aux données système de la commande à partir de fonctions système spéciales.

La programmation CPL offre les possibilités suivantes :

- Programmation symbolique avec des variables
- Traitement des chaînes de caractères
- Traitement de fichiers
- Opérateurs mathématiques :
+, -, *, /,
et fonctions trigonométriques, etc.
- Opérateurs de comparaison :
=, <, >, etc.
- Liaisons logiques :
NOT, AND, OR, etc.
- Structures de contrôle pour la commande du déroulement du programme :

Notions de base pour la programmation CN

REPEAT, WHILE, FOR, IF, CASE, GOTO, etc.

- Fonctions système pour la détermination des états internes du système :
positions, fonctions actives, données d'outil, signaux d'interface, etc.
- Services de processus :
sélection du programme, remise à zéro, lancement du programme, définition du mode de fonctionnement

Il est ainsi possible de créer et d'enregistrer n'importe quels déroulements de l'édition avec une écriture variable.

En règle générale, les instructions CPL sont inscrites en lettres capitales en tenant compte des définitions formelles.

Le recours au langage CPL a pour conséquences :

- un raccourcissement des programmes CN lorsqu'il existe des répétitions et des parties de programme de même nature ;
- des variantes de programme dépendantes de l'état grâce à l'accès aux états du système de la commande.

Une différence essentielle entre la programmation standard et la programmation CPL réside dans le fait que toutes les parties CPL sont supprimées dès la lecture de la ligne du programme correspondante au moment de la préparation des blocs. Il n'existe donc plus de parties CPL lors du traitement ultérieur du bloc ni au moment de l'interpolation du bloc CN.

3.2.2 Identification des éléments CPL au sein d'un programme pièce

Au sein d'un programme pièce, il est possible d'utiliser soit la programmation CPL, soit la programmation standard. Pour des raisons de clarté et notamment aussi pour permettre une interprétation efficace du programme, les parties du programme CPL doivent être signalées spécialement :

- Les blocs-programmes contenant exclusivement des éléments CPL doivent être signalés directement au début du bloc par une valeur positive sans signe (numéro de ligne) (voir [chap. 3.8.2 "Numéros de blocs" à la page 40](#)).

Exemple :

20 AX1\$="U"

Dans le bloc CPL 20, attribuer la valeur "U" à la variable de chaîne "AX1\$".

- Si des éléments CPL sont programmés **au sein d'un bloc CN standard** (par ex. pour la programmation symbolique de variables), les parties CPL doivent être insérées **entre crochets** ("[" et "]"). Cela sert à attribuer une valeur paramétrable aux fonctions CN et aux paramètres des fonctions CN. C'est pourquoi seules les expressions pouvant se trouver à droite d'un signe "=" (variables, expressions mathématiques, fonctions système fournissant une valeur correspondante) dans la programmation CPL peuvent être insérées entre crochets.

Exemple :

N10 G1 Y[PCS("X")]

Dans le bloc CN N10, l'axe Y doit être déplacé sur la position de la pièce à usiner programmée en dernier pour l'axe X.

N20 ROT([360/A+SIN(B)])

Dans le bloc CN N20, activer l'aide à la saisie « Tourner ». L'angle de torsion est défini à l'aide d'une expression CPL.

Notions de base pour la programmation CN

```
30 XPOS=100.5
```

Dans le bloc CPL 30, la valeur 100.5 est d'abord affectée à la variable "XPOS".

```
N40 G1 X[XPOS] Y10
```

Dans le bloc CN N40, le contenu de la variable "XPOS" est transmis à l'axe X en tant que valeur de coordonnée.

3.2.3 Éléments CPL au sein de la saisie manuelle, de la définition du bloc CN en mémoire tampon et de sous-programmes non liés

Au sein d'une saisie manuelle, d'une définition du bloc CN en mémoire tampon ou de sous-programmes non liés (appel avec identification DIN), tous les ordres CPL peuvent être utilisés sans variables globales ou locales.

Exceptions :

- Programmation label et instructions de saut ([chap. 3.11 "Programmation label et instructions de saut" à la page 64](#))
- Instructions de décision et de branchement ([chap. 3.12 "Instructions de décision et de branchement" à la page 71](#))
- Instructions de répétition ([chap. 3.13 "Instructions de répétition" à la page 74](#))

3.2.4 Respect des points d'interprétation entre les parties du langage CPL et standard !

L'identificateur particulier des parties de langage CPL permet au programmeur de reconnaître immédiatement quelles parties du programme ont déjà été supprimées lors de la préparation des blocs.

Les éléments de la programmation standard ne montrent un effet visible de l'extérieur qu'au moment de l'exécution du bloc sur la machine.

Exemple :

```
N523 KvProg(X[@K VX])
```

Au moment de l'exécution du bloc N523, la valeur KV utilisée pour l'axe X est celle qui **était** attribuée à la variable permanente @K VX au moment de la préparation du bloc.

Cependant, une valeur tout à fait différente peut être attribuée à la variable @K VX jusqu'au moment de l'exécution du bloc N523 (par ex. une valeur inscrite par un autre canal) !

Afin d'éviter toute incertitude ici, il convient d'utiliser la fonction "WAIT". L'ordre WAIT bloque la préparation des blocs jusqu'à ce que le bloc CN programmé juste avant ait été entièrement traité. Ainsi la préparation des blocs est synchronisée avec l'état actif (moment de l'interpolation) de la commande. Le traitement se poursuit ensuite avec la préparation des blocs programmés suivants. À ce moment-là, il n'existe plus de blocs préparés puisqu'ils ont déjà tous été traités.

Exemple :

```
WAIT
```

Synchroniser la préparation des blocs

```
N523 KvProg(X[@K VX])
```

Au moment de l'exécution du bloc N523, la valeur KV utilisée pour l'axe X est celle qui était affectée à la variable permanente @K VX au moment de la préparation du bloc. Étant donné que la préparation des blocs a déjà été synchronisée, il s'agit maintenant de la valeur active de la variable.

3.3 Liaison de programmes CN

3.3.1 Table de liaison

Quand un programme a été sélectionné, sa syntaxe et les éventuelles destinations des sauts ou appels de sous-programmes sont d'abord vérifiés. De plus, les structures d'administration correspondantes auxquelles la commande accède en cours d'exécution sont créées pour les variables CPL. Ce processus est appelé **liaison** (ou préparation).

Quand une liaison a été exécutée avec succès, une **table de liaison** est créée pour le programme CN correspondant. Toutes les tables de liaison de la commande IndraMotion MTX sont archivées dans un répertoire spécial qui est défini dans le paramètre machine 3080 00004. Le nom de la table de liaison d'un programme se compose du nom du programme pièce auquel est ajoutée l'extension ".l" (l : liaison).

Au cours de la montée en régime de la commande, celle-ci recherche le programme CN correspondant à toutes les tables de liaison existantes. Pour cela, la recherche est effectuée selon le chemin de recherche configuré dans le paramètre NCO/FileOrg/SrchPathSubProg (MP 3080 00001). Les tables de liaison pour lesquelles aucun programme pièce n'est trouvé sont supprimées.



L'ordre du chemin de recherche doit être configuré de sorte que les chemins contenant la plupart des programmes CN soient définis le plus en amont possible dans l'ordre du paramètre NCO/FileOrg/SrchPathSubProg (MP 3080 00001).

Lors d'une nouvelle sélection d'un programme CN déjà lié, la commande IndraMotion MTX utilise une table de liaison déjà existante dans la mesure où le programme pièce n'a pas été modifié entre-temps. Une nouvelle liaison est nécessaire après chaque modification du programme.

3.3.2 Liaison lors de la sélection ou de l'exécution d'un programme

En principe, il existe deux possibilités pour sélectionner et exécuter un programme qui n'est pas encore lié : la seule variante de liaison lors de la sélection du programme qui était disponible jusqu'à la version 12 de MTX et la nouvelle liaison en cours d'exécution du programme qui est disponible depuis la version 13 de MTX.

La liaison lors de la sélection du programme exécute un processus de liaison. Le programme peut être démarré une fois la liaison terminée. La liaison en cours d'exécution du programme lance un processus de liaison lors de la sélection du programme. Contrairement à la liaison lors de la sélection du programme, cette variante n'attend pas la fin du processus de liaison. Le programme peut donc être lancé immédiatement.

Le paramètre NCO/FileOrg/ChFileOrg/Ch/KindOfLinking (MP 7070 00030) définit si le processus de liaison peut se dérouler pendant l'exécution du programme.

Si le programme sélectionné est lié pendant l'exécution du programme, il est alors possible que des parties du programme soient exécutées avant d'avoir été liées, c.-à-d. avant d'avoir été traitées par le processus de liaison. Dans certains cas, cela peut ralentir légèrement le traitement du programme.

Si de tels ralentissements perturbent le processus de production, il existe 2 possibilités pour empêcher la liaison pendant l'exécution du programme :

1. Désactivation globale de la liaison pendant l'exécution du programme : définir le paramètre NCO/FileOrg/ChFileOrg/Ch/KindOfLinking (MP

Notions de base pour la programmation CN

7070 00030) pour le canal correspondant sur 1 (la liaison sera toujours terminée lors de la sélection du programme).

2. Désactivation individuelle de la liaison pendant l'exécution du programme pour chaque (sous-)programmes : programmer "(CPL)" sur la première ligne du (sous-)programme. L'ensemble du sous-programme, y compris les sous-programmes qu'il appelle lui-même, sont liés intégralement. Cela peut également être effectué pendant l'exécution du programme (voir chap. 3.3.3 "Liaison de sous-programmes – Liaison ultérieure" à la page 28).

Si le programme est lié pendant son exécution, alors l'identification "(DIN)" utilisable jusqu'à présent sur la première ligne du (sous-)programme et l'identificateur d'appel de sous-programmes "P <SP> DIN" sont ignorés.

Si la liaison du programme est terminée au moment de sa sélection ou si un (sous-)programme est intégralement lié en raison de la présence de l'identification "(CPL)", alors l'identification "(DIN)" continue d'agir comme précédemment. Le sous-programme correspondant est retiré du processus de liaison et ne doit par conséquent en aucun cas contenir des instructions CPL nécessitant un processus de liaison.



Avec des programmes CN lourds, la liaison intégrale risque de ralentir considérablement la sélection du programme. Il convient donc de l'éviter.



La création de tables de liaison en mode "Mise en lien avec la durée de fonctionnement du programme" est exécutée avec une priorité moindre. En cas de forte sollicitation de la commande, par exemple si de nombreux canaux fonctionnent en parallèle, la création de la table de liaison accusera un léger ralentissement. Si le programme est interrompu avant que la table de liaison ne soit terminée ou si un autre programme est sélectionné sur le même canal, alors la création de la table de liaison sera interrompue et cette dernière ne sera donc pas enregistrée.



La liaison lors de la sélection du programme exécute un contrôle de la syntaxe. Si un programme est sélectionné en mode "Mise en lien avec la durée de fonctionnement du programme", alors ce contrôle de la syntaxe n'a pas lieu lors de la sélection du programme. Par conséquent, une éventuelle erreur de syntaxe ne sera détectée qu'avec un léger retard. Lors de la modification ou de la création de programmes CN, il est donc recommandé d'exécuter une fois intégralement la liaison du programme lors de la sélection du programme. Pour ce faire, l'identification "(CPL)" peut être programmée sur la première ligne.

3.3.3 Liaison de sous-programmes – Liaison ultérieure

Si des sous-programmes sont appelés dans le programme à lier, la commande IndraMotion MTX vérifie s'il existe déjà des tables de liaison valides. Si c'est le cas, ces sous-programmes ne sont pas liés de nouveau. Cela permet de raccourcir considérablement le processus de liaison.

Si un programme pièce ne contient aucune partie CPL (sauts, variables CPL, expressions CPL, etc.), mais uniquement des blocs CN (DIN) et des appels de sous-programmes, alors le programme ne devra pas être lié expressément avant son exécution.

Notions de base pour la programmation CN

Dans ce cas, la touche écran à bascule "CPL Prog/DIN Prog" peut être commutée sur le réglage "DIN Prog" lors de la sélection du programme. Le cas échéant, les sous-programmes sont liés de nouveau pendant l'exécution du programme, ce qui risque de ralentir le flux de traitement.

Si des appels de sous-programmes par des variables CPL (par ex. P[SP\$]) sont programmés, alors le processus de liaison pour ce sous-programme ne sera exécuté qu'au cours de l'exécution du programme étant donné que le nom de la variable ne pourra être supprimé qu'à ce moment. Un sous-programme appelé de cette manière est donc toujours lié ultérieurement dans la mesure où il n'existe encore aucune table de liaison.

3.3.4 Influence du processus de liaison avec une identification DIN/CPL

Dans les conditions décrites ci-dessus, il est également possible d'influencer directement la liaison avec des instructions dans le programme pièce, indépendamment de la position effective de la touche écran à bascule "CPL Prog/DIN Prog".

Pour cela, le mot-clé "**(DIN)**" doit être programmé au début de la première ligne du programme. Ici encore, les sous-programmes éventuels sont liés ultérieurement lors de l'exécution.

Il est également possible de programmer l'instruction (DIN) sur la première ligne de chaque sous-programme, ce qui a pour effet, ici encore, de ne pas lier le sous-programme ainsi caractérisé.

En alternative, le mot-clé "**DIN**" (sans parenthèses) peut aussi être inscrit dans le programme appelant, directement après l'appel de sous-programme (voir aussi à ce sujet [chap. 3.9 "Sous-programmes" à la page 45](#)).

Comme pour l'instruction (DIN), le mot-clé "**(CPL)**" peut également être programmé au début de la première ligne du programme. Cela force la génération d'une table de liaison pour le programme concerné, même s'il a été sélectionné avec le réglage "DIN Prog" ou appelé en tant que sous-programme avec l'identification DIN.

3.4 Éléments de base d'un programme CN

3.4.1 Bloc-programme

Un programme CN se compose d'au moins un **bloc-programme**.

Les points suivants s'appliquent aux blocs-programmes :

- Au maximum 1 élément de contour (par ex. droite, arc de cercle) peut être programmé par bloc-programme.
- Les lignes vides sont autorisées avant ou après un bloc-programme pour une meilleure structure/lisibilité du code de programme.
- Un bloc-programme ne doit pas contenir plus de 512 caractères.
- Un bloc-programme se termine par le caractère ASCII <LINEFEED>.
- Un bloc-programme se compose d'au moins un **mot de programme** (par ex. fonction CN, paramètre de fonction).

3.4.2 Instructions

On entend par « instructions » les mots de programme qui ont une influence directe ou indirecte sur la trajectoire de l'outil, le déroulement du programme, le statut, l'état ou la réaction de la commande. Toutes les fonctions CN sont des exemples typiques d'instructions.

Notions de base pour la programmation CN

Vous trouverez les fonctions CN disponibles avec les règles de syntaxe nécessaires dans le [chap. 6 "Fonctions CN avec syntaxe de langage standard"](#) à la page 221.

Les **fonctions de parcours** ont une position particulière parmi les instructions :

Les **fonctions de parcours** décrivent la manière dont une position doit être approchée (par ex. droite, cercle, avec ou sans interpolation des axes impliqués, mouvement d'approche en avance ou en avance rapide, etc.).

Exemples : G0, G1, G2

Les fonctions de parcours sont souvent programmées dans un bloc avec des définitions de position, de course ou de rayon. Dans ce cas, les fonctions de parcours déclenchent aussi toujours des mouvements de déplacement.

De même, les définitions de position ou de course programmées dans un bloc-programme sans fonction de parcours, déclenchent toujours des mouvements de déplacement puisqu'il y a toujours une fonction de parcours quelconque active.

Exemple : Fonction de parcours avec définition de coordonnées

G1	X40 Y50	Se déplacer en avance vers X40/Y50
		Définition de coordonnées
		Fonction de course

3.4.3 Conditions additionnelles

On entend par « conditions additionnelles » les mots de programme qui permettent de régler sur la machine les conditions aux limites requises pour l'usinage ou la technologie.

Mots de programme importants ayant l'effet d'une condition additionnelle :

F<chiffre>	influence l'avance d'axes synchrones.
FA<chiffre>	influence l'avance d'axes asynchrones.
S<chiffre>	influence la vitesse de rotation de la broche.
M<chiffre>	active les fonctions M (sélection du rapport de réduction, sens de rotation de la broche, appel de sous-programme, etc.).

Les fonctions auxiliaires aussi sont souvent programmées en tant que fonctions M.

T<chiffre> sélectionne des outils.

Les mots de programme sont expliqués plus en détail dans le [chap. 6 "Fonctions CN avec syntaxe de langage standard"](#) à la page 221.

Exemple : Informations sur la course avec conditions additionnelles

G01	G71	X40 Y50	F250	S500	T05	M03	Se déplacer en avance vers X40/Y50 avec la valeur programmée F (avance = 250 mm/min) et S (vitesse de rotation = 500 mm/min) ; avec broche tournant vers la droite et l'outil T05 à disposition dans le magasin outils.
							Conditions additionnelles



Bien que de telles conditions additionnelles ne servent pas directement à décrire le contour ou le tracé de la trajectoire, elles peuvent néanmoins influencer ou déclencher des mouvements sur la machine (par ex. mouvement du magasin d'outils) !

3.5 Mots de programme

3.5.1 Vue d'ensemble

Un mot de programme est :

1. une **fonction CN** ou
2. un **paramètre avec une valeur ou une liste de paramètres**. Un paramètre est adressé dans le programme pièce par sa syntaxe (adresse).



Les ordres de synchronisation des blocs CN ainsi que les appels de sous-programmes représentent des cas particuliers de fonctions CN parce qu'ils peuvent être programmés avec une liste de paramètres CPL.

Tous les mots de programme se composent toujours d'un ou deux **mots partiels** qui peuvent être combinés comme suit :

Mot de programme	Mot partiel 1	Mot partiel 2	Exemples
Fonction CN	Syntaxe de fonction	–	AxAcc G0 G52.1
	Syntaxe de fonction	Liste de paramètres	AxAcc(...) G0(NIPS)
	Syntaxe de fonction	Valeur	D5 M777
Ordre de synchronisation	Syntaxe d'ordre	–	OFFSTOPA
	Syntaxe d'ordre	Liste de paramètres CPL	WPV[@9=10]
Sous-programme	Syntaxe de sous-programme Nom de sous-programme	–	P UP1 G4711
	Syntaxe de sous-programme Noms de sous-programmes	Liste de paramètres CPL	P UP2[7,@25]
Paramètre	Syntaxe de paramètre	Valeur	X-23.45 Y=AC(40) S250 F5000
	Syntaxe de paramètre	Liste de paramètres	O(0,0,1) ROTAX(0,45)

Tab. 3-1: Combinaison de mots partiels

Notions de base pour la programmation CN



Les codes G et M peuvent apparaître non seulement avec, mais également sans mot partiel 2 :

- Pour les codes G et M ayant une fonctionnalité interne fixe, la valeur numérique fait partie de la syntaxe, par ex. G0, G17, G54, M0, M3, M19, etc.
- Pour les fonctions définies par l'utilisateur (appels de sous-programmes et fonctions auxiliaires), la valeur numérique ne fait pas partie de la syntaxe mais représente la valeur de la fonction. Ces fonctions sont configurées au sein des paramètres machine.

3.5.2 Mots de programme provenant de fonctions CN

Les points suivants s'appliquent de manière générale :

- La syntaxe des fonctions CN peut se composer de codes G et M, mais également d'éléments du langage standard.

Exemples :

"G0", "G41", "G141", "G52.0", "M30", "Mirror(...)"

Vous trouverez toutes les fonction CN disponibles avec les règles de syntaxe correspondantes dans le [chap. 6 "Fonctions CN avec syntaxe de langage standard" à la page 221](#).

- Les fonctions CN peuvent avoir des paramètres supplémentaires permettant de piloter le mode opératoire des fonctions CN.

Ici, 3 cas sont à distinguer :

- Les paramètres programmés en tant que mots de programme autonomes dans un bloc CN, par ex. "G2 X10 I1.3 J2.5 G94 F1000". Dans la plupart des cas, il s'agit de paramètres définis dans le cadre de la norme DIN 66025.
- Les paramètres programmés en tant que mots de programme, mais intégrés à une **liste de paramètres avec des éléments de syntaxe spécifiques** entre parenthèses, par ex. "KvProg(X1.2,Y1.2)".
- Les paramètres intégrés à une **liste de paramètres sans éléments de syntaxe spécifiques** entre parenthèses, pour lesquels seule la valeur à l'emplacement défini est programmée, par ex. "Coord(0,1)", "Rotate(45)", "GetAxis(Z1,Z,W,REV)".
- Les listes de paramètres peuvent être facultatives et contiennent au moins un paramètre possible. Si une liste de paramètres contient plusieurs paramètres, ceux-ci doivent être séparés les uns des autres par une virgule.



Dans la présente documentation, les paramètres existants qui s'excluent mutuellement (paramètres alternatifs) sont signalés par le caractère "|" entre deux alternatives. Le caractère "|" n'est pas programmé.

- Les paramètres des fonctions CN peuvent être optionnels. En règle générale, s'ils ne sont pas programmés, des réglages standards pouvant soit être codés de manière fixe, soit être enregistrés dans les paramètres machine, prennent effet.

Exemple :

Règle de syntaxe : `TangToolOri({ SYM<s>} , { ANG<a>})`

Notions de base pour la programmation CN

La règle de syntaxe vous permet de reconnaître que les paramètres "SYM" et "ANG" sont des paramètres optionnels. Une valeur appropriée doit être programmée à la place du caractère de substitution <s> pour le paramètre "SYM" et <a> pour le paramètre "ANG". Les valeurs permises pour <s> et <a> sont normalement indiquées dans la règle de syntaxe.

Par conséquent, un paramétrage possible d'après la règle de syntaxe mentionnée ci-dessus serait "TangToolOri (SYM4) ".



Dans la présente documentation :

- les paramètres optionnels sont signalés par des accolades ;
- les caractères de substitution destinés aux valeurs à programmer par des chevrons simples.

Les accolades et les chevrons simples ne sont pas programmés.

Listes de paramètres sans éléments de syntaxe spécifiques :

Dans ce type de liste, tous les paramètres sont des constantes et sont transmis directement en tant que valeur numérique ou en tant que nom. Ils doivent être programmés dans un ordre défini avec précision, car la signification de chaque paramètre est déterminée seulement par sa position au sein de la liste.

Exemple : Toute modification de l'ordre des paramètres peut aboutir à un mode opératoire totalement différent :

GetAxis (X1, X, Y3, Y)	Les axes nommés X1 et Y3 dans le système sont acceptés sur le canal et les noms des canaux "X" et "Y" leur sont attribués.
GetAxis (X1, Y3, X, Y)	Les axes nommés X1 et X dans le système sont acceptés sur le canal et les noms des canaux "Y3" et "Y" leur sont attribués.

Les **paramètres d'une liste de paramètres** sans éléments de syntaxe spécifiques peuvent être optionnels.

Dans ce cas, les points suivants s'appliquent :

- Les paramètres optionnels au début et au milieu de la liste peuvent être omis, mais la virgule correspondante doit être programmée. C'est le seul moyen de conserver la qualité univoque de la position des paramètres au sein de la liste.
- Les paramètres optionnels à la fin de la liste peuvent être omis et la liste peut être terminée avec ")".

Exemple :

GetAxis(<Phy.Ax1> {,<Log.Ax1>} {,<Phy.Ax2>} {,<Log.Ax2>} {,...})

Programmation sans paramètre- GetAxis(X,,Z)
 tres au milieu :

Programmation sans paramètre- GetAxis(X)
 tres à la fin :

Utilisation d'**expressions CPL dans les listes de paramètres** sans éléments de syntaxe spécifiques :

Si des éléments individuels doivent être transmis en tant qu'expression CPL, ceux-ci doivent toujours être insérés entre crochets.

Notions de base pour la programmation CN

Exemple : Noms de variables en tant que paramètres de transfert

10 AX1\$="U"	La variable de chaîne "AX1\$" reçoit le contenu "U".
20 AX2\$="V"	La variable de chaîne "AX2\$" reçoit le contenu "V".
:	
N100 G17 ([AX1\$], [AX2\$])	G17 sous-tend le plan de travail avec les axes U et V.

Listes de paramètres avec des éléments de syntaxe spécifiques :

Dans ce type de liste, l'ordre de tous les paramètres est libre, car la signification d'un paramètre est déterminée par sa syntaxe programmée.

Exemple :

"KvProg (X1.2, Y1.4, Z1.6)" et

"KvProg (Z1.6, Y1.4, X1.2)"

ont le même effet puisque chaque valeur KV peut être attribuée de manière univoque grâce aux adresses des axes X, Y et Z.

Les expressions CPL peuvent être utilisées pour les éléments de syntaxe comme pour les valeurs numériques. Elles doivent alors être insérées entre crochets "[" et "]".

Exemple :

Exemple :

20 VALEUR=1.2	Attribuer la valeur 1.2 à la variable "VALEUR".
N30 KvProg (Y[VALEUR], X1.2)	Programmer les valeurs KV des axes Y et X. L'axe Y reçoit la valeur contenue dans la variable "VALEUR" ; l'axe X reçoit la valeur 1.2 (constante).

Exemple :

40 ACHSB\$="X":FACT%=2	Attribuer la valeur "X" à la variable de chaîne "ACHSB\$" et la valeur 2 à la variable entière "FACT%".
N50 Scale ([ACHSB\$] [FACT%])	Activer la mise à l'échelle de facteur 2 pour l'axe "X". Correspond à la programmation : Scale(X2)

Exemple :

60 DIM PARAMETRE\$(10)	Créer un tableau de caractères pour une chaîne de 10 caractères au maximum.
70 PARAMETRE\$="X2"	
N80 AxAcc ([Parametre\$])	Attribuer l'accélération de l'axe 2 m/s ² à l'axe "X".

3.5.3 Mots de programme en tant que paramètres

Paramètres suivis d'une valeur : *Les points suivants s'appliquent généralement à de tels mots de programme :*

- Une adresse commence toujours par une lettre et peut être composée de plusieurs caractères.
- Les mots de programme avec une adresse et des chiffres sont utilisés pour programmer par exemple :
 - des désignations d'axes et de coordonnées (X..., Y..., Z..., B...)
 - des rayons (R...) et des paramètres d'interpolation (I..., J..., K...)
 - des valeurs d'avance ou de temps (F...)
 - des vitesses de rotation des broches et des vitesses de coupe (S..., Si=...)
 - des corrections d'outil externes (ED...)
 - des corrections D (D...)
 - des fonctions auxiliaires (M..., T...)
- Il n'est pas nécessaire de programmer les zéros non significatifs.
- Les chiffres à virgule flottante s'écrivent avec un point décimal ; les zéros non significatifs sont optionnels (par ex. "X100.500" est identique à "X100.5").
- Si aucun signe n'est programmé ou si un signe plus est programmé, la valeur qui suit est toujours interprétée comme positive. Le signe moins déclare une valeur négative.

Exemple : Mot de programme composé de lettres d'adresse et d'un chiffre
(ici : valeur de coordonnée de l'axe X)

X-2407.0458

Valeur après la virgule

Valeur avant le signe décimal

Signe (+/-)

Lettre d'adresse

Paramètres suivis d'une liste de paramètres : *Les points suivants s'appliquent généralement à de tels mots de programme :*

- En principe, les explications données pour les mots de programme composés d'une adresse et d'un chiffre s'appliquent aussi ici.
- La liste de paramètres est programmée après l'adresse et insérée entre parenthèses. Les éléments individuels sont séparés les uns des autres par des virgules.
- Les éléments de la liste de paramètres sont des constantes et sont transmis directement en tant que valeur numérique ou en tant que nom. Ils doivent être programmés dans un ordre défini avec précision, car la signification de chaque paramètre est déterminée seulement par sa position au sein de la liste.
- Si des éléments individuels doivent être transmis en tant qu'expressions CPL, ceux-ci doivent être insérés entre crochets.
- Les mots de programme avec une adresse et une liste de paramètres sont utilisés pour programmer par exemple :
 - des coefficients splines (X(...),..., Y(...),..., Z(...),..., B(...))
 - l'orientation vectorielle (O(...), ROTAX(...))

Notions de base pour la programmation CN

3.5.4 Attributs de programmation

Lors de la programmation de la position des axes et des coordonnées, il est possible d'indiquer en option un attribut de programmation à la valeur de position programmée.

Si aucun attribut de programmation n'est indiqué, la valeur de position est interprétée conformément à l'état modal actuel de la machine.

Si un attribut de programmation est indiqué, l'état modal peut être remplacé localement pour chaque axe ou pour chaque coordonnée.

IndraMotion MTX connaît les attributs suivants :

AC(...) :	L'indication de position programmée est interprétée de façon absolue, indépendamment de G90/G91.
IC(...) :	L'indication de position programmée est interprétée de façon incrémentale, indépendamment de G90/G91.
DC(...) :	La position programmée pour un axe sans fin est approchée en empruntant la course la plus courte, indépendamment des réglages dans les paramètres machine et de la programmation de la fonction "PosMode".
ACP(...) :	La position programmée pour un axe sans fin est approchée avec une rotation en sens positif, indépendamment des réglages dans les paramètres machine et de la programmation de la fonction "PosMode".
ACN(...) :	La position programmée pour un axe sans fin est approchée avec une rotation en sens négatif, indépendamment des réglages dans les paramètres machine et de la programmation de la fonction "PosMode".
INCH(...) :	L'indication de position programmée pour un axe asynchrone est interprétée en pouces, indépendamment de l'unité de mesure enregistrée dans les données de configuration. L'attribut peut également être utilisé pour l'avance asynchrone (adresse FA) dont la valeur programmée est alors interprétée en pouces/minute pour les axes linéaires, indépendamment de l'unité configurée pour les axes asynchrones.
MM(...) :	L'indication de position programmée pour un axe asynchrone est interprétée en mm, indépendamment de l'unité de mesure enregistrée dans les données de configuration. L'attribut peut également être utilisé pour l'avance asynchrone (adresse FA) dont la valeur programmée est interprétée en mm/minute pour les axes linéaires, indépendamment de l'unité configurée pour les axes asynchrones.
S(...) :	Programmation de la vitesse en tr/min ou en MOD/min pour les axes sans fin asynchrones.

Exemples :

N10 G90 G1 F1000 X10 Y=IC(15)

L'axe X atteint la position 10 de manière absolue ;
l'axe Y avance de 15 mm de plus de manière incrémentale.

N20 G91 X=AC(15) Y5

Malgré G91, l'axe X atteint la position absolue 15 ;
l'axe Y avance de 5 mm de plus de façon incrémentale.

Notions de base pour la programmation CN

N30 B=DC (90)	L'axe B (axe sans fin) doit atteindre la position 90 degrés en empruntant la course la plus courte.
N40 B=ACP (350)	L'axe B doit atteindre la position 350 degrés en tournant dans le sens positif (déplacement : 260 degrés).
N50 B=ACN (0)	L'axe B doit atteindre la position 0 degré en tournant dans le sens négatif (déplacement : 350 degrés).
N100 B=S (1000)	Tourner l'axe rond B asynchrone à 1000 tr/min.

3.5.5 Utilisation des caractères de séparation entre 2 mots partiels

Description : *Chaque mot partiel peut se composer d'un ou plusieurs caractères (chaîne de caractères). Chaque caractère programmable peut être classé d'après les groupes suivants :*

- Lettres :
"A" à "Z", "a" à "z"
- Chiffres (signe décimal inclus !) :
"0" à "9", "."
- Caractères de séparation :
" ", "=", "+", "-", "(", ")"
- Autres caractères spéciaux (ici sans pertinence)

Un caractère de séparation doit être programmé entre 2 mots partiels voisins si :

- le premier mot partiel se termine par une lettre **ou** un chiffre et que
- le deuxième mot partiel commence par une lettre **ou** un chiffre.

C'est le cas par exemple quand :

- une autre syntaxe suit une fonction CN sans valeur ni liste de paramètres ;
- ou -
- une syntaxe de paramètre se termine par un chiffre et une valeur numérique doit être attribuée au paramètre.

Exemples :

1. N10 OVE FeedForward(...)	Caractère de séparation approprié : " "
2. N20 X2=2	Caractère de séparation approprié : "="

Dans le 2e cas, on peut également utiliser " " ou "+" comme caractère de séparation ; si la valeur attribuée est négative, le signe "-" est également possible.

L'utilisation de caractères de séparation peut également être pertinente pour des listes de paramètres contenant des éléments spécifiques à la syntaxe !

Particularités de la désignation des axes et des coordonnées :

Les désignateurs d'axes et de coordonnées voulus sur votre commande doivent être déterminés dans les paramètres machine :

- Paramètre MAIN/Dr/SysDrName (MP 1003 00001) Désignation de l'axe du système
- Paramètre CHAN/Ch/Coord/Acs/ChAx/ChAxName (MP 7010 00010) Désignation de l'axe du canal
- Paramètre CHAN/Ch/Coord/Acs/OptAxName/OptChAxName (MP 7010 00020) Noms optionnels des axes du canal

Notions de base pour la programmation CN

- Paramètres CHAN/Ch/Coord/Wcs/LinCoord et CHAN/Ch/Coord/Wcs/OriCoord (MP 7080 00010) Noms de coordonnées du canal
- Paramètre CHAN/Ch/Coord/Mcs/CartCoord (MP 7080 00020) Noms de coordonnées cartésiennes du canal

Les désignations d'axes et de coordonnées commencent toujours par une lettre et peuvent :

- se composer d'une ou plusieurs lettres, mais le nom de la chaîne de caractères en résultant ne doit pas être identique à celui d'une fonction CN.
Exemples : "X", "PALETTE"
- se composer d'une ou plusieurs lettres et se terminer par un chiffre.
Exemples : "X1", "PALETTE1"
- Un caractère de séparation approprié doit être programmé ici comme caractère de séparation entre le désignateur et la valeur suivante : "=", "+", "-" ou une espace.

Exemple : Les désignateurs d'axe "X" et "X2" sont définis.

N40 G1 X10	L'axe X atteint la position 10.
N50 G1 X20	L'axe X atteint la position 20.
N60 G1 X2.5	L'axe X atteint la position 2.5.
N70 G1 X=2.2	L'axe X atteint la position 2.2.
N80 G1 X 2.8	L'axe X atteint la position 2.8.
N90 G1 X2	L'axe X atteint la position 2.
N100 G1 X2 1	L'axe X2 atteint la position 1.
N110 G1 X2=2.8	L'axe X2 atteint la position 2.8.
N120 G1 X2+3	L'axe X2 atteint la position 3.
N130 G1 X2-2.4	L'axe X2 atteint la position -2.4.

3.6 Fin du programme

La fin du programme ou du sous-programme est atteinte dans les cas suivants :

- à la fin du fichier ;
- ou -
- si une ligne du programme contient "M2", "M02" ou "M30".
Vous trouverez de plus amples informations sur les fonctions M dans le [chap. 5 "Fonctions CN avec syntaxe conforme DIN 66025 \(extensions incl.\)" à la page 119.](#)

Exemple :

```

:
N250 ...      N250 est le dernier bloc-programme
M30           Fin du programme.
```

À la fin d'un sous-programme, le système retourne dans le programme appelant. Tous les états modaux ("modal", voir [chap. 3.7.1 "Modal" à la page 39](#)) sont conservés.

À la fin d'un programme principal, le système retourne à son début et attend un nouveau "démarrage CN". Si "M2", "M02" ou "M30" a été utilisé pour terminer le programme principal, les états modaux sont définis sur la partie M30 de la chaîne d'initialisation.

3.7 Effet des mots de programme

3.7.1 Modal

Description : Les mots de programme peuvent avoir un effet "modal" ou "non modal".
"Modal" signifie qu'un mot de programme reste actif dans chacun des blocs-programmes suivants jusqu'à ce que vous...

- programmez le même mot de programme avec une autre valeur ;
- programmez un autre mot de programme annulant son effet ;
- ou -
- désactivez de façon ciblée la fonction du mot de programme.



Le terme "à autoretenue" est parfois utilisé comme synonyme de "modal".

Exemple d'un programme pièce CN :

N10 F1000	Régler la vitesse d'avance sur 1000 mm/min. F1000 a un effet modal.
N20 G0 X0 Y0	Interpolation linéaire en avance rapide sur la position X0/Y0. G0 a un effet modal.
N30 Z100	Interpolation linéaire en avance rapide sur la position Z100.
N40 G1(IPS1) X10 Y10	Interpolation linéaire (en avance ; avec fenêtre de positionnement exacte) à 1000 mm/min sur la position X10/Y10. G1 annule l'effet de G0. G1(IPS1) a un effet modal.
N50 X20	Interpolation linéaire (en avance ; avec fenêtre de positionnement exacte) à 1000 mm/min sur la position X20/Y10.
N60 G1(IPS2) X30 Y30	Interpolation linéaire (en avance ; avec fenêtre de positionnement approximative) à 1000 mm/min sur la position X30/Y30. G1 a été programmé avec une autre valeur (IPS2). G1(IPS2) a un effet modal.
N70 X40 Y40 F500	Interpolation linéaire (en avance ; avec fenêtre de positionnement approximative) à 500 mm/min sur la position X40/Y40. F a été programmé avec une autre valeur (500). F500 a un effet modal.

Notions de base pour la programmation CN

N80 G0 X0 Y0	Interpolation linéaire en avance rapide sur la position X0/Y0. G0 annule l'effet de G1(...). G0 a un effet modal.
N90 Scale (X2,Y2)	Activer la fonction de mise à l'échelle. La mise à l'échelle a un effet modal.
:	La fonction de mise à l'échelle reste active.
N200 Scale()	Désactiver la fonction de mise à l'échelle.

3.7.2 Non modal

"Non modal" signifie qu'un mot de programme n'a d'effet que dans le bloc-programme dans lequel il est programmé.

 Les termes "sans autoretenue" ou "local" sont parfois utilisés comme synonymes de "non modal".

Exemple d'un programme pièce CN :

N10 G1 F1000	Activer l'interpolation linéaire. G1 a un effet modal. Régler la vitesse d'avance sur 1000 mm/min. F1000 a un effet modal.
N20 G75.1 X100 Y100	Déplacement vers le 1er palpeur de mesure. (G75.1) a un effet non modal. Dans ce bloc, G75.1 a la priorité sur la fonction modale G1.
N30 Z100	Interpolation linéaire en avance vers Z100. G1 conserve son effet modal.

3.8 Éléments particuliers de configuration du programme

3.8.1 Identificateur de canal

Description : Cet outil permet de déterminer les canaux exclusifs dans lesquels le programme concerné peut être utilisé. Son lancement dans un autre canal entraîne une erreur d'exécution.

L'identificateur du canal est inscrit au début du programme.

Syntaxe :

\$<numéro du canal>

Tab. 3-2: Syntaxe de l'identificateur de canal

Exemple :	N10 \$2	Ce programme ne peut être exécuté que sur le canal 2.
	:	Instructions du programme
	M30	Fin du programme

3.8.2 Numéros de blocs

- Les blocs-programmes **peuvent** être caractérisés par un numéro de bloc afin d'améliorer la lisibilité d'un code de programme.

Notions de base pour la programmation CN

Dans la mesure où vous attribuez des numéros de blocs univoques, vous pouvez programmer des sauts vers ces numéros de blocs.

- Les blocs-programmes contenant exclusivement des éléments CPL **doivent** être identifiés par un numéro de bloc. De tels blocs sont également appelés **blocs CPL**.

Les points suivants s'appliquent ici :

- Les numéros de blocs doivent toujours être le premier mot CN programmé sur une ligne de programme.
- Les numéros de blocs des **blocs CN standards** se composent de la lettre d'adresse "N" suivie directement par un chiffre positif (exemple : N10, N10.2).

Veillez ce que les expressions CPL au sein de blocs CN standards soient insérées entre crochets.

- Les numéros de blocs des **blocs CPL purs** se composent exclusivement d'un chiffre positif sans signe (exemple : 10, 11.9).

L'instruction CPL ou l'accord CPL du bloc est programmé après le numéro de bloc.

- Si un bloc CPL se termine par ":", il convient d'ajouter un autre bloc CPL auquel aucun numéro de ligne n'est attribué.

3.8.3 Lignes vides dans le code de programme

Les lignes vides permettent de structurer le programme et d'augmenter ainsi sa clarté. Les lignes vides sont ignorées par la commande.

3.8.4 Commentaire dans un programme pièce

Lors du déroulement du programme, les commentaires sont ignorés par la commande.

Utilisez des commentaires pour :

- documenter le code du programme ou insérer des explications ;
- commenter des lignes entières du programme ou des passages individuels.

L'insertion de commentaires facilite et accélère le remaniement ultérieur du programme, par exemple lorsque celui-ci doit être modifié. Cependant, chaque caractère des commentaires augmente la taille du fichier de programme de 1 octet.

Commentaire de lignes entières du programme :

Vous pouvez masquer totalement chaque ligne du programme, qu'il s'agisse d'un bloc CN standard ou d'un bloc CPL.

Pour insérer des commentaires :

- programmez un point-virgule ";" au début du bloc ;
- ou -
- insérez la ligne complète entre parenthèses "(" et ")".

Commentaires dans un bloc CN standard :

Vous pouvez programmer un commentaire n'importe où à l'intérieur d'un bloc CN standard :

- Programmez un point-virgule ";" à l'endroit où le commentaire commence. La commande interprétera la ligne du programme comme un commentaire à partir du point-virgule et jusqu'à la fin de la ligne.

Notions de base pour la programmation CN

- Insérez des commentaires isolés à l'intérieur d'un bloc CN standard entre parenthèses "(" et ")". Il est alors possible de programmer des "commentaires imbriqués", par exemple si vous voulez masquer une séquence à l'intérieur d'une ligne du programme qui contient déjà un commentaire entre parenthèses.

Les parenthèses ne doivent pas être utilisées après des fonctions pour lesquelles il est possible de programmer une liste de paramètres entre parenthèses (par ex. G0, G1, G61) et pour lesquelles aucune liste de paramètres n'a été programmée.

Exemple :

N10 G0 (<texte de commentaire>)	Programmation non valide !
N20 G0 (NIPS) (<texte de commentaire>)	Programmation valide

- Vous pouvez insérer un commentaire après une fonction pouvant être suivie d'une liste de paramètres qui n'est pas programmée en l'introduisant par "//". Dans ce cas, vous pouvez refermer le commentaire par "\" si vous le voulez.

Commentaires a sein d'un bloc CPL :

Vous pouvez insérer un commentaire au sein d'un bloc CPL en l'introduisant par "REM". La commande interprétera la ligne du programme comme un commentaire à partir de l'instruction REM et jusqu'à la fin de la ligne.

Syntaxe :

REM <texte de commentaire>

Tab. 3-3: Syntaxe REM

Exemple :

```
...
10 REM ***Sous-programme pour démasquer le mot d'état***
...
```

Exemples : Commentaires dans un programme pièce

:	Code de programme
; <commentaire>	Ligne de commentaire
;N10 <bloc CN standard>	Masquer le bloc CN standard.
;20 <CPL>	Masquer le bloc CPL.
(<commentaire>)	Ligne de commentaire
(N30 <bloc CN standard>)	Masquer le bloc CN standard.
(40 <CPL>)	Masquer le bloc CPL.
N50 G1 X0 Y0 ; <commentaire>	Commentaire dans un bloc CN standard.
N60 G1(IPS) (<commentaire>) F1000	Commentaire inséré dans un bloc CN standard.
N70 X10 (Y10 (<commentaire>))	"Commentaire imbriqué" dans un bloc CN standard.
N80 G0 //<commentaire>	Commentaire inséré après une fonction pour laquelle des paramètres optionnels pourraient être programmés.

Notions de base pour la programmation CN

N90 G0 //<commentaire>\\ X0 Y0 Commentaire isolé après une fonction pour laquelle des paramètres optionnels pourraient être programmés.

100 REM <commentaire> Commentaire dans un bloc CPL.

3.8.5 Remarques dans l'interface utilisateur

Utilisez la programmation de remarques pour afficher des textes de remarque dans l'interface utilisateur de la commande numérique (max. 80 caractères).

Cela vous permet par exemple, au cours du déroulement du programme, d'indiquer à l'utilisateur de la machine :

- l'état actuel du programme ou
- des consignes de manipulation.

On distingue deux types de remarques :

- **les remarques spécifiques au canal :**

Elles sont effacées au moment de la désélection du programme ou de la remise à zéro du canal.

Syntaxe : MSG (<texte de remarque>)

- **les remarques multicanaux :**

Elles peuvent être effacées par une remise en zéro globale.

Syntaxe :

GMSG (<texte de remarque>)

Vous pouvez aussi programmer une remarque pour donner une consigne de manipulation à l'utilisateur de la machine. Pour cela, vous devez programmer un "M0" sur la même ligne ou sur la ligne suivante. Vous vous assurez ainsi que le programme sera interrompu dès que le texte de remarque s'affichera. Le programme ne reprendra son déroulement qu'après l'activation de "Démarrage CN".

Exemple :

:	Code de programme
N60 (MSG Mesurer pièce à usi-	Émettre une remarque spécifique au canal.
ner !)	
N70 M0	Attendre l'activation de la touche Démarrage CN.
:	Code de programme

Pour des raisons de compatibilité, il existe quelques variantes alternatives de la syntaxe pour la programmation de remarques, qui sont toutes équivalentes du point de vue de leur fonctionnalité :

- Variantes de la syntaxe des remarques spécifiques au canal :
 (MSG<texte de remarque>)
 (*MSG<texte de remarque>)
 (MSG,<texte de remarque>)
 (*MSG,<texte de remarque>)
- Variantes de la syntaxe des remarques multicanaux :
 (GMSG<texte de remarque>)
 (GMSG,<texte de remarque>)

Notions de base pour la programmation CN

Si aucun texte de remarque (ou une espace seulement) n'est programmé avec MSG ou GMSG, les remarques en suspens sont effacées. La fonction MSG supprime alors les remarques spécifiques au canal qui ont été créées avec MSG ou PRN(0,...). GMSG permet de supprimer les remarques non spécifiques à un canal.

Exemple :	:	Code de programme
	N60 MSG(Mesurer la pièce à usiner !)	Émettre une remarque spécifique au canal.
	N70 M0	Attendre l'activation de la touche Démarrage CN.
	N80 MSG()	Effacer la remarque.
	:	Code de programme

3.8.6 Sauts dans le déroulement du programme

Plus les programmes sont volumineux, plus une "programmation précise" est importante.

On entend par là notamment :

- une programmation structurée,
- une tolérance à l'égard des erreurs,
- une ergonomie du logiciel.

Les programmes structurés sont généralement plus clairs. Le regroupement de sections raisonnables ou des fonctions souvent requises dans des sous-programmes (paramétrés) ou sous une destination de saut dotée d'un désignateur (label) compréhensible, entraîne non seulement une meilleure lisibilité, mais aussi un travail plus efficace étant donné que ces programmes peuvent être réutilisés dans d'autres programmes.

Les options suivantes sont possibles :

- **Appels de sous-programmes**

(voir [chap. 3.9 "Sous-programmes" à la page 45](#))

Utilisez des sous-programmes quand une section de traitement précise apparaît plusieurs fois de façon identique ou similaire au cours du traitement.

Programmez cette section une seule fois (le cas échéant avec la possibilité de transmettre des paramètres), enregistrez-la en tant que programme et appelez-la tout simplement en cas de besoin.

Cela vous permet de réduire le code du programme et d'économiser de l'espace mémoire. Par ailleurs, vos programmes seront beaucoup plus clairs et beaucoup plus agréables à entretenir.

- **Instructions de sauts**

(voir [chap. 3.11 "Programmation label et instructions de saut" à la page 64](#))

Utilisez les instructions de sauts pour poursuivre le déroulement du programme au sein du programme en cours à un autre emplacement en fonction d'événements définis (par ex. résultats de calcul).

- **Instructions de décision/de branchement**

(voir [chap. 3.12 "Instructions de décision et de branchement" à la page 71](#))

Utilisez de tels ordres quand des blocs-programmes individuels doivent exécuter des sections de programme différentes ou des sous-programmes complets en fonction de conditions définies.

- **Instruction de répétition**

(voir [chap. 3.13 "Instructions de répétition" à la page 74](#))

Utilisez les instructions de répétition quand des sections de programme ou des sous-programmes complets doivent être exécutés à plusieurs reprises. Dans ce cas, il est généralement possible de définir aussi le nombre de répétitions.

3.9 Sous-programmes

3.9.1 Vue d'ensemble

On appelle sous-programmes (SP) les programmes appelés par un appel de sous-programme. Une fois le sous-programme exécuté, le programme appelant reprend son traitement en commençant après l'appel de sous-programme. On parle de sous-programme "récursif" lorsqu'un programme s'appelle lui-même.

On appelle programme principal (PP) le programme à partir duquel on passe au premier niveau de sous-programme (niveau SP).

D'un point de vue formel, on ne distingue pas les programmes principaux des sous-programmes, mais seuls les sous-programmes peuvent disposer de paramètres de transfert.

Les points suivants s'appliquent ici :

- Les sous-programmes peuvent contenir des blocs CN standards et des blocs CPL.
- Chaque programme pièce peut être appelé en tant que sous-programme par d'autres programmes. Cependant, un programme ne peut pas s'appeler lui-même en tant que sous-programme (appel récursif impossible).
- Le programme appelant peut transmettre des paramètres à un sous-programme.
- La profondeur d'imbrication maximale s'élève à 24, c.-à-d. que la commande peut maintenir un maximum de 24 niveaux de sous-programmes ouverts simultanément.
- La commande est sensible à la casse lorsqu'il s'agit des noms de sous-programmes.

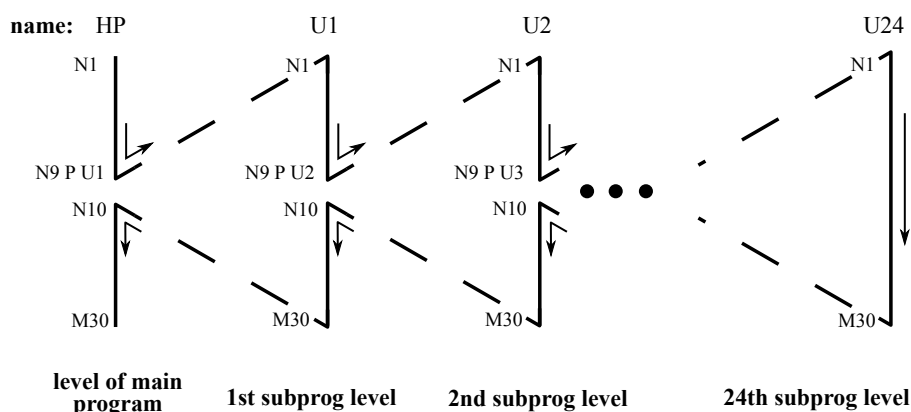


Pour plus d'informations sur la fin d'un sous-programme, voir [chap. 3.6 "Fin du programme" à la page 38](#).

La profondeur d'imbrication maximale admissible ne peut pas être configurée ou étendue directement. Cependant, différents paramètres machine peuvent empêcher d'atteindre la profondeur d'imbrication maximale. Pour plus d'informations à ce sujet, voir [chap. 3.9.9 "Configuration et élimination des erreurs" à la page 61](#).

Exemple : Représentation schématique de l'imbrication maximale admissible des sous-programmes. N1 est le premier bloc-programme dans chaque cas ; le sous-programme suivant est appelé dans le bloc N9. M30 termine le (sous)-programme correspondant.

Notions de base pour la programmation CN



On distingue les sous-programmes ayant un effet local de ceux ayant un effet modal :

- En règle générale, les sous-programmes ont un effet **local**. Dans ce cas, le sous-programme est appelé une fois au point d'appel.
- En revanche, si un sous-programme à effet **modal** est activé, celui-ci est rappelé à chaque mouvement de déplacement programmé suivant jusqu'à ce qu'il soit désélectionné. Les cycles de perçage en constituent un exemple d'application.

IndraMotion MTX offre les variantes d'appel de sous-programmes suivantes :

- Appel avec l'adresse P et le nom de sous-programme à partir d'un bloc CN standard (en option avec indication du chemin).
- Appel uniquement avec le nom de sous-programme à partir d'un bloc CN standard, sans indication de chemin et sans adresse P.
- Sous-programme en tant que fonction G ou M autodéfinie dans un bloc CN standard.
- Activation d'un sous-programme modal avec syntaxe autodéfinie à partir d'un bloc CN standard.
- Appel à partir d'un bloc CPL via la fonction CALL.

3.9.2 Appel d'un sous-programme avec adresse P

- Description :**
- Le nom du sous-programme est programmé directement après l'adresse P. En option, le répertoire dans lequel se trouve le sous-programme peut également être programmé.
 - Il est également possible de programmer le nom du sous-programme avec ou sans chemin après l'adresse P par une variable CPL. La variable CPL doit alors être insérée entre crochets.
 - Afin d'améliorer la lisibilité, une espace peut être programmée en tant que caractère de séparation entre l'adresse P et le nom du sous-programme.
 - L'appel de sous-programme doit être programmé à la fin du bloc.
- Les mouvements de déplacement programmés dans le même bloc sont exécutés avant que le sous-programme ne soit appelé (voir l'exemple).
- Au maximum, 1 appel de sous-programme peut être programmé dans un bloc.
 - L'appel de sous-programme est local (non modal).

Notions de base pour la programmation CN



Le recours à des appels de sous-programmes par des variables CPL comme "P [UP\$]" allonge la durée d'exécution du programme.

Syntaxe :

P{<chemin>}<nom> {DIN} ou P[<var. CPL>] {DIN}	
avec	
<chemin>	Répertoire dans lequel se trouve le sous-programme.
<nom>	Nom du programme à appeler.
<var. CPL>	Variable de chaîne CPL dont la valeur est le nom du sous-programme avec ou sans chemin.
DIN	Optionnel. Empêche la liaison du sous-programme. Les instructions CPL "simples" (voir chap. 3.2.3 "Éléments CPL au sein de la saisie manuelle, de la définition du bloc CN en mémoire tampon et de sous-programmes non liés" à la page 26) sont autorisées au sein du sous-programme non lié. Les appels de sous-programmes sont autorisés ; ceux-ci sont liés ultérieurement si nécessaire. Pour plus d'informations, voir l'ordre CALL (chap. 3.9.7 "Appel de sous-programme dans CPL via l'ordre CALL" à la page 53).

Tab. 3-4: Syntaxe avec adresse P

Exemple :

```

:
N40 P Gabarit de perçage Appel du programme "Gabarit de perçage".
N50 X100 Le bloc N50 suit la fin du sous-programme.
:
N140 GO X10 Y0 PUP1 D'abord le positionnement sur X10/Y0 en avance rapide.
N150 Z0 Ensuite l'appel du programme "UP1".
: Le bloc N150 suit à la fin du sous-programme.
200 DIM UP$ (20)
210 UP$="Gabarit de perçage"
N220 P[UP$]
```

3.9.3 Appel d'un sous-programme sans adresse P

Description : Les sous-programmes peuvent aussi être appelés **sans** adresse P placée en tête :

- Seul le nom du sous-programme est programmé directement.
- Le nom du sous-programme peut également être programmé avec une variable CPL. La variable CPL doit alors être insérée entre crochets.
- Il n'est pas possible d'indiquer le chemin.
- L'appel de sous-programme doit être programmé à la fin du bloc.

Les mouvements de déplacement programmés dans le même bloc sont exécutés avant que le sous-programme ne soit appelé (voir l'exemple).

Notions de base pour la programmation CN

- Au maximum, 1 appel de sous-programme peut être programmé dans un bloc.
- L'appel de sous-programme est local (non modal).

Syntaxe :

<nom> ou [<var. CPL>]	
avec	
<nom>	Nom du programme à appeler.
<var. CPL>	Variable de chaîne CPL contenant le nom du sous-programme sans chemin.

Tab. 3-5: Syntaxe sans adresse P



Assurez-vous que cette variante de programmation ne provoque aucune confusion avec la syntaxe normale !
Si le sous-programme à exécuter commence par une syntaxe valide, il doit être appelé avec une adresse P.

C'est pourquoi vous devez toujours utiliser des noms de sous-programmes univoques afin d'éviter toute erreur d'interprétation de la part de l'interpréteur de la commande.

Exemple :

```

:
N40 XUP                                Appel du programme "XUP".
N50 X100                               Le bloc N50 suit la fin du sous-programme.
:
100 DIM UP$ (20)
110 UP$="XUP"
N120 [UP$]
N100 X1UP
:
:
:
:
:
N140 G0 X10 Y0 XUP                     D'abord le positionnement sur X10/Y0 en avance
                                         rapide.
N150 Z0                                Ensuite l'appel du programme "XUP".
:                                       Le bloc N150 suit à la fin du sous-programme.
```

Attention !

Le nom de sous-programme "X1UP" programmé ici entraîne une erreur de syntaxe, car "X1" est interprété comme la coordonnée d'un axe nommé "X" et il n'existe aucun programme nommé "UP".



Une immersion dans les sous-programmes appelés par la variable CPL n'est **pas** possible dans la boîte de dialogue d'avance de bloc MTX.

3.9.4 Sous-programmes locaux

Description : À partir de la version MTX 11V02, il est également possible de définir des sous-programmes dans le fichier du programme principal. Un sous-programme local commence par l'ordre **LPS <nomSP>** (LocalProgStart) et se termine par l'ordre **PEND** (ProgrammEnd). Au sein d'une séquence de programme,

Notions de base pour la programmation CN

l'ordre CN **LP <nomSP>** (LocalProg) permet d'appeler un sous-programme local.

- Pour appeler des sous-programmes locaux, le nom du sous-programme local est programmé directement après l'adresse LP.
- Le nom du sous-programme peut également être programmé avec une variable CPL. La variable CPL doit alors être insérée entre crochets.
- L'appel de sous-programme doit être programmé à la fin du bloc.
 Les mouvements de déplacement programmés dans le même bloc sont exécutés avant que le sous-programme ne soit appelé (voir l'exemple).
- Au maximum, 1 appel de sous-programme peut être programmé dans un bloc.
- L'appel de sous-programme local a un effet local (non modal).
- Des paramètres de position ou d'adresse peuvent être transmis aux sous-programmes locaux. Dans ce cas, les paramètres sont transmis de la même façon que pour les sous-programmes "ordinaires".
- En option, l'appel de sous-programme LP local peut contenir le paramètre **DIN**. Le paramètre DIN permet de signaler que le sous-programme local contient exclusivement des "instructions CPL simples" ou DIN (voir [chap. 3.2.3 "Éléments CPL au sein de la saisie manuelle, de la définition du bloc CN en mémoire tampon et de sous-programmes non liés" à la page 26](#)) et ne doit par conséquent pas être lié.
- L'identification DIN/CPL peut être insérée entre parenthèses et programmée sur la première ligne du sous-programme local au lieu d'être programmée lors de l'appel du sous-programme local.
- Les sous-programmes locaux ne peuvent pas être appelés en tant que sous-programmes asynchrones (SPAS).
- Les sous-programmes locaux peuvent être imbriqués et contenir aussi des appels de sous-programmes ordinaires.
- Les sous-programmes locaux doivent se trouver à la fin du fichier, une fois le programme principal terminé.
- La fin du programme est caractérisée par M30. Si aucun M30 n'est programmé, alors la première programmation de LPS est interprétée implicitement comme la fin du programme.
- Les définitions des contours pour les cycles d'enlèvement des copeaux doivent être insérées dans le fichier avant le premier ordre LPS.

Application : Une application des sous-programmes locaux consiste à réduire les temps de liaison des programmes pièces volumineux en utilisant presque exclusivement une programmation DIN (par ex. pour les surfaces de formes libres). Pour cela, les parties DIN sont "délocalisées" dans un sous-programme local qui est ensuite appelé avec le paramètre **DIN**.

Syntaxe :

LP <nom> ou [<var. CPL>] {DIN}	
avec	
<nom>	Nom du programme à appeler.

Notions de base pour la programmation CN

<var. CPL>	Variable de chaîne CPL contenant le nom du sous-programme sans chemin.
DIN	Optionnel. Empêche la liaison du sous-programme. Utilisez ce paramètre uniquement si le sous-programme ne contient aucun bloc CPL. Par contre, les appels de sous-programmes sont autorisés ; ils sont liés ultérieurement si nécessaire.

Tab. 3-6: *Sous-programmes locaux*

Exemple :	:	
	10 I% = 5	
	N10 G0 X0 F1000	
	20 IF (...	
	:	
	LP Up1 (2, 3.4)	Appel du sous-programme local Up1 avec les paramètres de position (P1, P2)
	:	
	X10 LP Up2	Mouvement de déplacement sur X10 en avance rapide suivi de l'appel du sous-programme local Up2
	:	
	LP Up3 (MOD2, X3.4)	Appel du sous-programme local Up3 avec les paramètres d'adresse (MOD, X)
	M30	Fin du programme
	LPS Up1	Début du sous-programme local Up1
	:	
	PEND	Fin du sous-programme local Up1
	LPS Up2	Début du sous-programme local Up2
	:	
	PEND	Fin du sous-programme local Up2
	LPS Up3	Début du sous-programme local Up3
	(ADR_PARA)	Identificateur pour la transmission des paramètres d'adresse
	:	
	PEND	Fin du sous-programme local Up3



Une immersion dans les sous-programmes appelés par la variable CPL n'est **pas** possible dans la boîte de dialogue d'avance de bloc MTX.

3.9.5 Appels de sous-programmes autodéfinis avec les codes G et M

La commande vous permet de définir les éléments suivants en plus des appels de sous-programmes déjà mentionnés :

- des sous-programmes "non modaux" avec l'adresse M
(voir le paramètre NCP/SubProg/NonModalMFunc/MFunc/MFuncCode (MP 3090 00003) et le paramètre NCP/SubProg/NonModalMFunc/MFunc/MFuncSubProg (MP 3090 00004)) ;
- des sous-programmes "non modaux" avec l'adresse G
(voir le paramètre NCP/SubProg/NonModGFunc/GFunc/GFuncCode (MP 3090 00001) et le paramètre NCP/SubProg/NonModGFunc/GFunc/GFuncSubProg (MP 3090 00002)).

Il est possible d'attribuer des sous-programmes jusqu'à un maximum de 64 codes G et 8 codes M définis par l'utilisateur, ces sous-programmes étant appelés lors de la programmation des codes G ou M correspondants en tant que sous-programmes locaux (non modaux).

Les codes G et M configurés en tant qu'appels de sous-programmes ne doivent pas entrer en conflit avec les codes G et M définis de manière fixe.



Pour savoir quels appels de sous-programmes sont configurés spécialement pour votre machine, contactez le responsable de votre système.

*Les points suivants s'appliquent à la **programmation** d'appels de sous-programmes autodéfinis :*

- Seul le code G ou M correspondant est programmé dans le programme pièce. Le sous-programme appelé est déterminé dans la configuration des paramètres machine.
- L'appel de sous-programme doit être programmé à la fin du bloc.
Les mouvements de déplacement programmés dans le même bloc sont exécutés avant que le sous-programme ne soit appelé.
- Au maximum, 1 appel de sous-programme peut être programmé dans un bloc.
- L'appel de sous-programme est local (non modal).

3.9.6 Appels de sous-programmes modaux autodéfinis

Suite à leur premier appel, les sous-programmes modaux sont exécutés automatiquement après chaque mouvement de déplacement affecté par un bloc CN standard. Ceci est valable jusqu'à ce qu'ils soient désélectionnés par une fonction CN spéciale.

Les appels de sous-programmes modaux sont configurés dans les paramètres machine avec la syntaxe de la fonction de mise hors service (voir le paramètre NCP/SubProg/ModaSubProg/SwiOn/SwiOnSynSubProg (MP 3090 00005) et suivants).

Il est possible de configurer un maximum de 63 sous-programmes modaux. Les noms des sous-programmes modaux sont alors attribués aux syntaxes librement définissables. De plus, vous devez indiquer le nombre maximal de paramètres de sous-programme pouvant être transmis au sous-programme modal correspondant.

Les syntaxes des appels de sous-programmes modaux ne doivent pas entrer en conflit avec les fonctions CN définies de manière fixe.

Notions de base pour la programmation CN



Pour savoir quels appels de sous-programmes sont configurés spécialement pour votre machine, contactez le responsable de votre système.

*Les points suivants s'appliquent à la **programmation** d'appels de sous-programmes modaux :*

- Seule la syntaxe configurée correspondante du sous-programme modal est programmée dans le programme pièce. Le sous-programme appelé est déterminé dans la configuration des paramètres machine.
- Un seul appel de sous-programme peut être programmé dans un bloc.
- L'appel de sous-programme est modal, c'est-à-dire que le sous-programme est rappelé après chaque mouvement de déplacement programmé jusqu'à ce qu'il soit désélectionné.
- Les sous-programmes modaux sont désactivés par la syntaxe de désélection.
- Si un sous-programme modal est programmé tandis qu'un autre sous-programme modal est déjà activé, il est alors exécuté une seule fois à la position programmée comme s'il s'agissait d'un sous-programme non modal.
- Si le même sous-programme modal est programmé une nouvelle fois sans que la syntaxe de désélection n'ait été programmée auparavant, alors seuls les nouveaux paramètres sont importés. Le programme n'est lancé avec les nouveaux paramètres qu'avec le mouvement de déplacement suivant. Ce comportement correspond à la programmation de la syntaxe de désélection avant le nouveau paramétrage du sous-programme modal.

*Les points suivants s'appliquent aux **paramètres** des sous-programmes modaux :*

- Les paramètres programmés sont hérités d'un appel de sous-programme à l'autre, c.-à-d. que les paramètres programmés une fois s'appliquent à chaque appel de sous-programme modal jusqu'à leur désélection.
- Si un paramètre de transfert programmé d'un sous-programme modal est modifié dans le sous-programme modal, alors la valeur **modifiée** sera héritée.
- Au maximum, <n> paramètres programmés peuvent être hérités, où <n> représente le nombre de paramètres de transfert défini dans le paramètre NCP/SubProg/ModaSubProg/SwiOn/NofParSubProg (MP 3090 00007) pour le sous-programme modal correspondant. Si plus de <n> paramètres sont programmés, ils sont uniquement valables lors du premier appel de sous-programme ; la valeur n'est pas attribuée lors des appels modaux suivants.
- Si un sous-programme modal est activé et reprogrammé (sans avoir été désélectionné au préalable), alors les paramètres nouvellement programmés s'appliquent en plus des anciens paramètres hérités. Ceci est valable jusqu'à ce que le nombre maximal de paramètres pouvant être hérités soit atteint (voir plus haut).

Remarque : Les valeurs de paramètres non programmées dans l'appel ne sont jamais héritées !

3.9.7 Appel de sous-programme dans CPL via l'ordre CALL

Description : L'ordre CALL permet d'appeler des sous-programme à partir de programmes CPL purs.

Les points suivants s'appliquent à la programmation :

- Le sous-programme programmé après l'ordre CALL est appelé directement.
- L'ordre CALL doit être programmé dans un bloc CPL à part.
- L'appel de sous-programme est local.

Après le mot-clé CALL se trouve un nom de programme à la suite duquel on peut retrouver des paramètres de transfert entre crochets et l'identification "DIN" (pour influencer le processus de liaison) en dernière position.



Il est interdit d'utiliser le caractère ":" dans un bloc CPL contenant une instruction CALL. Les instructions CPL suivantes doivent être programmées dans un nouveau bloc CPL.

Exemple :

```
50 IF A% = 1 THEN
51 CALL P999
52 ENDIF
:
```

Influence sur le processus de liaison ("préparation") à l'aide de l'identification "DIN" :

Si vous programmez l'identification "DIN" en tant que terminaison lors d'un appel de sous-programme via l'ordre CALL, alors la commande n'exécute pas la liaison du sous-programme appelé. Cela permet d'accélérer considérablement le processus de liaison, par exemple dans le cas d'un programme principal appelant de nombreux sous-programmes.

Exemple :

```
50 IF A% = 1 THEN
51 CALL P999 DIN
52 ENDIF
M30
```

Le sous-programme "P999" n'est pas lié.

Nous recommandons de programmer l'identification "DIN" uniquement si le sous-programme appelé...

- se compose uniquement de blocs DIN ou d'instructions CPL simples (voir [chap. 3.2.3 "Éléments CPL au sein de la saisie manuelle, de la définition du bloc CN en mémoire tampon et de sous-programmes non liés" à la page 26](#)) et
- n'appelle aucun autre sous-programme.

Si un sous-programme n'a pas été lié à cause de l'identification "DIN" et qu'il contient des éléments CPL non autorisés, alors la commande émet un message d'erreur correspondant pendant l'exécution du programme.

En alternative, vous pouvez aussi insérer l'identification "DIN" en tant que **commentaire** sur la première ligne du sous-programme à appeler. Dans ce cas, la commande n'exécute pas la liaison du programme.

Exemple : Identification "DIN" dans le programme à appeler

Notions de base pour la programmation CN

Exemple : Identification "DIN" dans le programme à appeler

```

N10 (DIN)                Le sous-programme "P999" n'est pas lié.
N20 ...
:

```

3.9.8 Transmission de paramètres aux sous-programmes

Généralités

Les paramètres peuvent être transmis à un sous-programme de deux manières différentes :

- à l'aide de la **liste des paramètres CPL**,
- à l'aide de la **liste des paramètres d'adresse**.

Dans le sous-programme, la liste des paramètres CPL permet d'accéder aux valeurs transmises à l'aide des variables locales P1 à P<n>. Une valeur de paramétrage est attribuée au nom de la variable selon l'ordre de la programmation au sein du programme appelant. Dans la liste des paramètres d'adresse, le nom de la variable est également indiqué dans le sous-programme lors de l'appel de sous-programme, d'une part pour que l'ordre de la programmation ne joue aucun rôle dans l'appel, d'autre part pour que les noms "parlants" des variables puissent être utilisés dans l'appel et dans le sous-programme. Le fonctionnement de ces deux méthodes est équivalent.

Liste des paramètres d'adresse

Une **liste de paramètres d'adresse** permet de transférer des paramètres à un sous-programme. Pour ce faire, les paramètres sont insérés dans une liste entre parenthèses "(" et ")" et programmés directement après l'appel de sous-programme. Les différents paramètres sont séparés les uns des autres par des virgules.

Les paramètres à transférer sont programmés dans le programme appelant par la combinaison d'un mot d'adresse de paramètre et d'une valeur de paramétrage, éventuellement séparés par une espace ou par le caractère "=".

Exemples : Les exemples suivants sont équivalents et permettent de transmettre la valeur réelle "3.4" à l'adresse "A" du sous-programme :

```

N10 P Sous-programme (A 3.4)
N20 P Sous-programme (A=3.4)
N30 P Sous-programme (A3.4)

```

Unicité des noms d'adresses

Dans le bloc N30 cité en exemple, la séparation entre le paramètre d'adresse et la valeur n'est pas univoque : Dans le sous-programme, il est possible d'attribuer la valeur ".4" à la variable "A3" ou la valeur "3.4" à l'adresse (variable CPL) "A". Dans ce cas, la commande recherche toujours le dernier caractère alphanumérique comme séparation entre l'adresse et la valeur, puis elle interprète le reste comme valeur.

Si la valeur ".4" doit être attribuée à la variable "A3", une espace ou le caractère "=" doit être programmé entre "A3" et ".4" :

```

N40 P Sous-programme (A3 .4).

```

Ceci s'applique aussi notamment à la transmission de chaînes. Ces dernières doivent pouvoir être différenciées du paramètre d'adresse de manière univoque.

Notions de base pour la programmation CN

En général, les chaînes doivent être programmées entre guillemets ("). Si d'autres guillemets doivent être saisis dans une chaîne entre guillemets, il faut alors saisir une barre oblique (\) avant les nouveaux guillemets. Les barres obliques sont elles-mêmes représentées par deux barres obliques.

Pour les chaînes toutes simples (pas d'espace ouvrante ou fermante, pas de virgule, pas de "parenthèse fermante"), les guillemets ne sont pas obligatoires et les barres obliques (\) sont alors interprétées comme des caractères normaux.

Exemple:

Paramètres de chaîne

```
N10 P Up(ST="une chaîne (avec des parenthèses)")
N20 P Up(ST="une chaîne avec du \"texte\".")
N30 P Up(ST="chemin: \\mnt\\Test.npg")
N40 P Up(ST=ChaineSimple)
```

Dans ces exemples, le caractère d'attribution (=) peut aussi être remplacé par une espace.



Pour utiliser la liste de paramètres d'adresse pour transmettre des paramètres à un sous-programme, le mot-clé "(ADR_PARA)" doit être programmé sur la première ligne du programme appelé. Une identification "(DIN)" supplémentaire empêche la liaison du sous-programme et donc la transmission des valeurs de paramétrage.

Adresses de paramètres admissibles

- Variables CPL du sous-programme

Valeurs de paramétrage admissibles

- Chiffres
- Constantes de chaîne
- Expressions CPL arithmétiques (entre crochets CPL "[" et "]")

Pour transférer plusieurs paramètres, les couples mot d'adresse/valeur de paramètre sont séparés par une virgule :

```
N50 P Sous-programme(A3, B5.2).
```

Il n'est pas possible de saisir plusieurs virgules les unes à la suite des autres.

Contrairement à la liste des paramètres CPL, l'ordre des paramètres n'a aucune importance. Les adresses non programmées ne sont pas attribuées dans le sous-programme.

Une valeur peut être attribuée à chaque variable CPL dans le sous-programme. Dans le sous-programme, aucun nom de variable particulier comme P1, P2%, etc. ne doit être utilisé, comme c'est le cas dans la liste des paramètres CPL.

Exemple 1 :

Utilisation de noms de variables identiques

Notions de base pour la programmation CN

Programme principal :

```

10 A=5.5

N20 P Sous-programme (A=3.4)

N30 P Sous-programme (A=[A])

```

Sous-programme :

```

(ADR_PARA)

N10 G0 X0

N20 G1 X[A]

```

Explication :

Le programme principal attribue la valeur 5.5 à une variable locale "A". Dans le bloc N20, le sous-programme est appelé pour la première fois et la valeur 3.4 est attribuée à la variable locale "A" du **sous-programme**. Dans le bloc N20, le sous-programme déplace l'axe X de la position 0 à la position 3.4 définie par la variable locale "A" du sous-programme. Lors du deuxième appel de sous-programme dans le bloc N30, la valeur de la variable locale "A" du **programme appelant** est attribuée à la variable locale "A" du **sous-programme**.

Règle à retenir :

La variable locale du sous-programme se trouve à gauche devant le caractère de séparation ; les variables locales du programme appelant ou d'autres expressions admissibles se trouvent à droite du caractère de séparation, entre crochets.

Exemple 2 :**Utilisation de plusieurs paramètres, transmission de chaîne****Programme principal :**

```

10 A=5.5

N20 P Sous-programme (A=3.4, B[A*3], UP="UnSous-programme")

...

```

Sous-programme :

```

(ADR_PARA)

10 IF B=NUL THEN
20 B=0
30 ENDIF

N10 G1 X[A] Y[B]

40 DIM UP$(20)

N20 P [UP$]

```

Explication :

Notions de base pour la programmation CN

Le programme principal attribue la valeur 5.5 à une variable locale "A". Le sous-programme est appelé dans le bloc N20 et la valeur 3.4 est attribuée à la variable locale "A" du **sous-programme**. Le sous-programme attribue la valeur 0 à la variable B, si elle n'a pas été programmée. Ensuite, dans le bloc N10, l'axe X est déplacé sur la position 3.4 et l'axe Y sur la position $3 * 5.5 = 16.5$, car les valeurs 3.4 et 16.5 ont été attribuées respectivement à A et à B lors de l'appel de sous-programme. Le troisième paramètre UP définit qu'un autre sous-programme nommé "UnSous-programme" doit être appelé dans le sous-programme.



Si le programme appelant est sélectionné sans liaison et qu'une chaîne est transmise au sous-programme, l'ordre PDIM doit être utilisé (à la place de l'ordre DIM) (voir aussi la liste des paramètres CPL).

Liste des paramètres CPL

Description : Une **liste des paramètres CPL** permet de transmettre des paramètres à un sous-programme. Pour ce faire, les paramètres sont programmés dans une liste entre crochets CPL "[" et "]" directement après l'appel de sous-programme. Les différentes valeurs des paramètres sont séparées les unes des autres par des virgules.

Les paramètres transmis lors de l'appel de sous-programme sont toujours adressés dans le sous-programme par l'intermédiaire des variables P1, P2, P3, etc. dans l'ordre de la transmission des paramètres.

Il est ainsi possible d'adresser des paramètres (par ex. P1TEST, P2XYZ, etc.), mais les lettres majuscules suivant P1, P2, etc. sont alors ignorées (P1 = P1TEST = P1XYZ).

Les paramètres admissibles sont :

- les chiffres (pour les nombres à virgule flottante inférieurs à 1, le 0 doit impérativement être programmé avant la virgule, par ex. : 0.134.)
- les constantes de chaînes CPL (entre guillemets : "<constante de chaîne CPL>")
- les variables CPL
- les expressions CPL arithmétiques

Exemple :

- Dans le sous-programme P999, la valeur 2.75 est attribuée à P1,
- la valeur de la variable X% au moment de la transmission des paramètres est attribuée à P2,
- la valeur 0 est attribuée à P3.

Si P2 doit aussi représenter une valeur entière dans le sous-programme, c'est possible en ajoutant le caractère % à P2. Cet identificateur du type de variable peut aussi être utilisé de manière analogue pour les autres types de variables.

La valeur de chaque paramètre individuel peut être attribuée à d'autres variables dans le sous-programme.

Programme principal :

```
50 IF A% = 1 THEN
```

```
51 CALL P999 [2.75, X%, 0]
```

Appel de sous-programme avec transmission de paramètres.

Notions de base pour la programmation CN

```
52 ENDIF
```

```
M30
```

Sous-programme P999 :

```
1 FAKTOR=P1 : XWERT%=P2% : KORRTAB%=P3%
```

```
N1 G1 X[XWERT%*FAKTOR]
```

```
N2 G22 K[KORRTAB%]
```

L'ordre PDIM doit être utilisé quand :

- un sous-programme avec une constante de chaîne est appelé en tant que paramètres de transfert **et**
- que le programme appelant doit être sélectionné sans liaison.

Syntaxe :

PDIM <nom du paramètre>(<taille du tableau>)	
	Si la taille du tableau n'a pas été programmée ou si la valeur programmée est trop petite, alors la commande signale une erreur du programme pièce "VARIABLE INVALIDE".

Tab. 3-7: Syntaxe PDIM

Exemple : Programme principal :

```
N10 (DIN)
```

```
:
```

```
N50 P UP["Test"]
```

```
M30
```

Sous-programme :

```
10 PDIM P1$(4)
```

```
M30
```

La variable de chaîne P1\$ a la valeur "TEST".

Transmission des paramètres à des sous-programmes modaux

Comme les sous-programmes non modaux, les sous-programmes modaux peuvent être paramétrés par l'intermédiaire de paramètres de transfert. Il convient toutefois de tenir compte du point suivant :

Le nombre de paramètres transmis à un sous-programme modal est défini pour chaque sous-programme modal dans le paramètre NCP/SubProg/ModaSubProg/SwiOn/NofParSubProg (MP 3090 00007). La valeur qui y est configurée définit également le nombre de valeurs de paramètres pouvant être héritées lors de l'appel de sous-programme modal suivant.

En cas de recours à la liste des paramètres CPL, les valeurs de paramètres des variables P1 à Pn (n est défini dans le paramètre NCP/SubProg/ModaSubProg/SwiOn/NofParSubProg (MP 3090 00007)) sont transmises et héritées à l'appel de sous-programme suivant une fois le sous-programme modal terminé.

Si la liste des paramètres d'adresse est utilisée, les n premières adresses programmées sont transmises à l'appel suivant. Les adresses non programmées ne sont pas héritées.

Conversion d'un bloc CN dans une liste de paramètres

Lors de certains processus d'usinage, il est nécessaire de ne pas exécuter directement un mouvement de déplacement rapide, par ex. un mouvement d'avance G0, en le remplaçant par exemple par un mouvement d'avance al-

Notions de base pour la programmation CN

ternatif. C'est le cas, par exemple, sur les machines de découpe au jet réglées en hauteur lorsque le réglage en hauteur n'est pas assez dynamique pour empêcher avec certitude une collision de la tête d'usinage avec l'outil à une vitesse de réglage élevée. Ici, il faut alors aussi indiquer une "augmentation" des axes/coordonnées réglés en hauteur à la place d'un mouvement linéaire. Cet exemple est connu sous le nom de "Leapfrog" (saut de grenouille). Étant donné qu'il est souvent impossible de réécrire les programmes CN, une aide côté CNC est utile pour couvrir ce cas d'application. Évidemment, cette solution peut également s'appliquer à d'autres cas d'application.

La fonction CN "SubPConv" convertit la syntaxe des coordonnées et valeurs programmées après elle dans une liste de paramètres de sous-programme d'adresse. Dans le cas d'application décrit, il faut donc développer un sous-programme (par ex. "LeapFrog") qui attend le point de destination du mouvement d'avance G0 sous forme de paramètres et génère d'autres mouvements à partir de lui. Pour s'assurer que le programme CN ne puisse pas être modifié, il faut configurer une macro CN supplémentaire afin de remplacer la fonction CN "G0" par "SubPConv LeapFrog". Ici, la fonction CN "SubPConv" a l'effet d'un appel de sous-programme P normal. Le nom de sous-programme est programmé après la fonction CN et doit contenir une indication de chemin absolue (afin d'éviter toute ambiguïté due à l'ordre de recherche).

Pour la conversion d'un mouvement d'avance G0 (typique) de type "G0 X100 Y220", cet exemple montre que la chaîne CN "SubPConv LeapFrog X100 Y220" est d'abord générée par le remplacement de macro. Lors d'une deuxième étape, la fonction de conversion convertit la chaîne après le nom de sous-programme dans une liste de syntaxes de sous-programme valide. "SubPConv LeapFrog (X=100, Y=220)".

Le sous-programme "LeapFrog" a ainsi accès aux variables CPL "X" et "Y" et peut générer un nouveau mouvement de déplacement.

Restrictions : Les valeurs de coordonnées ou d'axe programmées devant la fonction CN "SubPConv" sont d'abord traitées dans le même bloc CN. Par exemple, "X100 Y200 SubPConv LeapFrog" déplace les axes sur (100,200), puis appelle le sous-programme "LeapFrog" auquel aucun paramètre n'est transmis.

Si une liste d'axes ou de coordonnées est interrompue par un élément de la syntaxe autre qu'un axe ou une coordonnée, alors la conversion de la liste de paramètres prend fin avant cet élément de syntaxe. L'ordre d'axe ou de coordonnée programmé après cet élément de syntaxe est alors exécuté avant l'appel de sous-programme. Exemple : "SubPConv LeapFrog X100 G91 Y200" entraîne un positionnement relatif (G91) sur 200 dans Y et un appel de sous-programme avec la liste de paramètres "LeapFrog (X=100)".

Si, par exemple, la syntaxe "G0" est remplacée par "SubPConv LeapFrog" à l'aide d'une fonctionnalité de macro, alors le remplacement s'applique à tout le système. Cependant, il n'est pas toujours souhaitable ou possible (par ex. dans une chaîne d'initialisation) d'appeler un sous-programme, si bien qu'aucun message d'erreur involontaire peut être émis. Veillez à ce que le texte de recherche de macro ne puisse pas apparaître dans des situations sensibles ou critiques !

Exemple d'application LeapFrog

Hypothèse 1 : Chaque mouvement de positionnement en avance rapide programmé avec G0 doit exécuter un "LeapFrog". G00 et G000 ne sont pas influencés pour permettre des mouvements d'avance linéaires normaux.

Notions de base pour la programmation CN

Hypothèse 2 : Le sous-programme "LeapFrog" spécifique au processus et à la machine accepte 2 paramètres : X et Y, puisque les mouvements d'avance (cartésiens) s'effectuent dans ces coordonnées.

Définition de macro (par ex. /usrfep/macro/MacroDef_G0.xml)*Programme:*

```
<MacroDef>
  <Macro Type="DIN" Overlay="Yes">
    <Name>G0</Name>
    <Replace>SubPConv LeapFrog</Replace>
    <Comment>Replace every occurrence of G0 with "SubPConv LeapFrog"</Comment>
  </Macro>
</MacroDef>
```

Sous-programme /usrfep/LeapFrog*Programme:*

```
(ADR_PARA)

; total retract height
10 RETRACTHEIGHT = 10
; height which has to be moved vertically
20 VLIFT = 2

; read current position (start point)
30 CUR_X = PCS("X")
40 CUR_Y = PCS("Y")
50 CUR_Z = PCS("Z")

; if no coordinate is given, the endpoint equals the start point
60 IF (X = NUL) THEN
70   X = CUR_X
80 ENDIF

90 IF (Y = NUL) THEN
100  Y = CUR_Y
110 ENDIF

; calculate the length of the desired movement in X/Y plane
130 DX = X - CUR_X
140 DY = Y - CUR_Y
150 LENG = SQRT(DX * DX + DY * DY)

; do some spline calculation (application dependent)
120 L_PARAM=2*(RETRACTHEIGHT - VLIFT)

; external influence: If @1 == 0, then enable Leap Frog,
; else do a linear G0 movement without Leap Frog

160 IF (@1 = 1) THEN
N170   G000 X=[X] Y=[Y]
180 ELSE
      ; Leap-Frog-Logik

190   IF (LENG < 2*RETRACTHEIGHT) THEN
      ; too short for leap frog. Do NOT use rapid mode (as the
      ; height control may not be dynamic enough even for short
      ; movements
N200   G1 X=AC([X]) Y=AC([Y])
210   ELSE
      ; generate 3 movements:
      ; vertikal upwards
      ; horizontal
      ; vertikal downwards
      ; use spline cornering (SCOL) to increase corner velocity
N220   G000 Z=AC([RETRACTHEIGHT + CUR_Z]) SCOL(L1=[L_PARAM], L2=[L_PARAM])
N230   G000 X=AC([X]) Y=AC([Y]) SCOL(L1=[L_PARAM], L2=[L_PARAM])
N240   G000 Z=AC([CUR_Z])
250   ENDIF

260 ENDIF
```

Exemple:

Exemples de programmation

N10 G0 X100 Y200 **génère (équivalent)** N10 P LeapFrog(X=100, Y=200)

N20 G0 X100 **génère (équivalent)** N20 P LeapFrog(X=100). La coordonnée Y reste en position.

N30 G00 X100 ne subit aucune modification (le fichier de définition de macro ci-dessus ne provoque aucun remplacement de G00), c.-à-d. qu'aucun sous-programme LeapFrog n'est exécuté.

N40 G0 X100 M992 Y100 **génère** N40 P LeapFrog (X=100) M992 Y100, c.-à-d. que l'axe Y est d'abord déplacé avec le code G initial (par ex. G1), puis le sous-programme LeapFrog est appelé.

3.9.9 Configuration et élimination des erreurs

Le réglage par défaut de la commande CN ne permet pas d'autoriser plus de 8 niveaux de sous-programmes. Si plus de 8 sous-programmes sont utilisés, les messages d'erreur suivants peuvent apparaître ; les étapes requises pour éliminer les erreurs sont alors décrites.

Erreur de canal 2085

Texte d'erreur :

Trop de fichiers ouverts

Cause :

Chaque sous-programme est ouvert lors de la liaison ou de l'édition. Le nombre de fichiers ouverts pour chaque canal est limité par un paramètre machine.

Correction :

Augmenter le paramètre machine 7060 00210.



En règle générale, nous recommandons de configurer le paramètre machine selon la profondeur d'imbrication maximale atteinte par les programmes, plus 3, plus le nombre de fichiers ouverts dans CPL.

Erreur de canal 1927

Texte d'erreur :

La préparation des blocs est limitée à NNN blocs.

Cause :

Toute entrée dans un sous-programme utilise un bloc CN. Si trop d'entrées de sous-programmes sont programmées sans mouvement de déplacement, alors les blocs CN utilisés ne peuvent pas être validés à temps, ce qui entraîne un manque de ressources.

Correction :

- Augmenter le paramètre /NCO/LookAh/Ch/NofBlkPrep (MP 7060 00110) ;
- ou -
- Désactiver l'aperçu des blocs pour la zone concernée avec G9 suivie éventuellement de l'activation de l'aperçu des blocs avec G8 ;
- ou -
- Programmer des mouvements de déplacement entre les entrées de sous-programmes, si possible.

Notions de base pour la programmation CN

Erreur de canal 2716 Texte d'erreur :

Pas assez de mémoire courante (pile) disponible.

Cause :

Lors de l'appel de sous-programme, les variables CPL locales du programme appelé sont stockées sur la pile CPL. Si cette erreur se produit, l'espace libre sur la pile CPL requis à cet effet dépasse la valeur configurée.

Correction :

- Augmenter le paramètre NCO/Mem/ChMemCh/CplStack (MP 7070 00010), index 1 ;
- ou -
- Utiliser moins de variables CPL et des variables CPL moins volumineuses, si possible.

Erreur de canal 825 ou 465 Texte d'erreur :

Mémoire insuffisante pour programme XXXX

Cause :

La mémoire des tables de liaison ne suffit pas au programme indiqué.

Correction :

Augmenter le paramètre NCO/Mem/ChMem/Ch/CplLinkBuff (MP 7070 00010), index 2

3.10 Programmation de macros

3.10.1 Vue d'ensemble

Les macros sont des aides à la saisie qui remplacent des parties du bloc CN.

Les macros se composent :

- d'un nom,
- d'un texte de remplacement,
- d'attributs (DIN ou CPL).

Les macros sont configurées dans des fichiers de définition. Pour connaître leur fonction précise et la procédure à suivre, consultez le manuel "Rexroth IndraMotion MTX Description des fonctions", chapitre "Programmation de macros".

Dans tous les cas, les macros sont uniquement actives dans le contexte (DIN ou CPL) pour lequel elles ont été configurées. Ceci dit, les macros sont utilisées comme les mots CPL ou CN.

Exemple:

Utilisation d'une macro

Nom de macro :	G0
Texte de remplacement :	ACC(X3.0, Y3.0, Z3.0) G0
Bloc programmé :	N10 G0(NIPS) M5
Effet :	N10 ACC(X3.0, Y3.0, Z3.0) G0(NIPS) M5

Vous trouverez des exemples plus complexes dans le manuel "Rexroth IndraMotion MTX Description des fonctions".



Les macros peuvent masquer des fonctions CN. Dans ces circonstances, les programmes pièces existants ne réagissent donc pas comme prévu.

Il vaut mieux éviter les macros qui masquent des fonctions CN.

Il est recommandé de donner des noms clairement identifiables aux macros.

3.10.2 Macros avec des paramètres

Pour concevoir des macros avec plus de flexibilité, l'utilisateur peut définir des macros paramétrables. Si une macro avec des paramètres doit être appelée dans le programme pièce, les paramètres doivent être insérés entre parenthèses après le nom de la macro. Les paramètres sont séparés les uns des autres par des virgules. Si un paramètre doit contenir une virgule ou une parenthèse, il doit être inscrit entre guillemets. Si des guillemets doivent être transmis en tant que tels dans le texte du paramètre, ils doivent être précédés d'une barre oblique inverse ("").

Exemple:

Utilisation d'une macro avec guillemets

Nom de macro :	Remarque
Texte de remplacement :	PRN#{0,{1}}
Bloc programmé :	10 Remarque(" \"valeur A\$=\", A\$ ")
Effet :	10 PRN#{0, "valeur A\$=", A\$ }

Les paramètres sont introduits dans le texte de remplacement en tant que chaînes de caractères, exactement comme ils ont été programmés. Aucun contrôle de l'exactitude syntaxique n'est effectué.

Exemple:

Utilisation d'une macro avec des paramètres

Nom de macro :	Course_X
Texte de remplacement :	G0 X[{1}]
Bloc programmé :	N10 Course_X(@J) M5
Effet :	N10 G0 X[@J] M5
Nom de macro :	Deplacement
Texte de remplacement :	G0(NIPS) ACC({2}) {1}
Bloc programmé :	N10 Deplacement(X100 Y100, "X3.0, Y3.0, Z3.0") M5
Effet :	N10 G0(NIPS) ACC(X3.0, Y3.0, Z3.0) X100 Y100 M5

Notions de base pour la programmation CN

Vous trouverez des exemples plus complexes dans le manuel "Rexroth IndraMotion MTX Description des fonctions".

3.11 Programmation label et instructions de saut

3.11.1 Vue d'ensemble

Les instructions de saut permettent de poursuivre le déroulement du programme à des points d'entrée définis.

Concernant les instructions de saut, la commande IndraMotion MTX offre les fonctionnalités suivantes :

-
- Programmation label pour les blocs CN standards (DIN)
 - Programmation label pour les blocs CPL
 - GoAhead (GOA) : saut en avant vers un bloc CN standard
 - GoBack (GOB) : saut en arrière vers un bloc CN standard
 - GoCond (GOC) : saut conditionnel vers un bloc CN standard
 - GoTo : saut inconditionnel vers un bloc CN standard
 - Saut CPL (GO-TO) : saut vers un bloc-programme quelconque
 - RetTo : saut en arrière vers un bloc-programme quelconque
-



Le saut CPL GOTO est traité beaucoup plus rapidement, car l'adresse de la destination du saut est déjà calculée pendant l'exécution de la liaison.

Pour les ordres DIN en revanche, la destination du saut n'est déterminée qu'au cours de l'exécution du programme. Pour cela, le label doit être recherché dans le programme, ce qui dure d'autant plus longtemps que la destination du saut est éloignée de l'ordre GOTO.

C'est pourquoi il vaut mieux privilégier l'ordre CPL.

Il est interdit de sauter vers une instruction

CASE-LABEL...LABEL-OTHERWISE-ENDCASE à l'aide du saut CPL GOTO.

Il est interdit d'entrer dans ou de sortir d'une instruction de répétition CPL ou d'une instruction de branchement CPL à l'aide des ordres de saut CN standards GoAhead, GoBack, GoCond et GoTo.

Les instructions CPL suivantes sont concernées :

- REPEAT – UNTIL
 - WHILE – DO – END
 - FOR – STEP – TO – NEXT
 - IF – THEN – ELSE – ENDIF
 - CASE – LABEL... LABEL – OTHERWISE – ENDCASE.
-

3.11.2 Labels dans les blocs CN standards et blocs CPL

Un label est un marqueur de saut pour un ordre de saut. La commande IndraMotion MTX distingue les labels dans un bloc CN standard (DIN) de ceux dans un bloc CPL.

Programmation label dans un bloc CN standard (DIN) :

- La destination du saut doit toujours être programmée au début du bloc.
- Dans les blocs ayant un numéro de bloc, la destination du saut se trouve directement derrière le numéro du bloc, séparée par un caractère de séparation.
- Le nom label peut être composé de 2 à 32 caractères. Sont autorisés les lettres, les traits de soulignement et les chiffres, mais les deux premiers caractères ne doivent pas être des chiffres. L'écriture du nom est sensible à la casse.
- Deux-points doivent être programmés derrière le nom label à la destination du saut.



Pour des raisons de compatibilité, les labels DIN devraient être programmés exclusivement **après le numéro du bloc**. Si un label est programmé directement au début du bloc et commence par la même chaîne de caractères qu'une fonction CN, cela provoquerait une erreur de syntaxe.

Étant donné que les nouvelles versions contiennent toujours aussi de nouvelles fonctions CN, il se peut que certains programmes existants ne fonctionnent plus dans les nouvelles versions à cause d'erreurs de syntaxe.

Il est vivement conseillé de programmer les labels DIN uniquement après le numéro de bloc afin d'éviter ce problème.

Programmation label dans un bloc CPL :

- La destination du saut est programmée directement après le numéro du bloc, dont elle est séparée par un caractère de séparation.
- Le nom label se compose d'un signe décimal suivi de caractères ASCII, en commençant par une majuscule.
- Un label ne doit pas être une variable.

3.11.3 Saut en avant GoAhead (GOA) vers un bloc CN standard

Description : Poursuit sans conditions le déroulement du programme au point de destination du saut (label).

Les points suivants s'appliquent ici :

- La destination du saut doit être définie.
- La destination du saut doit se trouver entre le bloc-programme actuel et la fin du fichier.

Pour en savoir plus sur la programmation label, voir [chap. 3.11 "Programmation label et instructions de saut"](#) à la page 64.

Syntaxe :

GoAhead <label>
Format abrégé : GOA

Notions de base pour la programmation CN

avec	
<label>	Nom de destination du saut 2 à 32 caractères Sont autorisés les lettres, les traits de soulignement et les chiffres, mais les 2 premiers caractères ne doivent pas être des chiffres.

Tab. 3-8: Syntaxe GoAhead

Exemple :

```

:
N40 GoAhead LABEL1      Saut en avant vers la destination de saut "LABEL1".
:
N80 LABEL1 :            Programmation label de la destination de saut "LA-
:                          BEL1".

```



Il est interdit d'entrer dans ou de sortir d'une instruction de répétition CPL ou d'une instruction de branchement CPL à l'aide de l'ordre de saut CN standard GoAhead.

Pour en savoir plus sur les instructions CPL, voir [chap. 3.11.1 "Vue d'ensemble" à la page 64.](#)

3.11.4 Saut en arrière GoBack (GOB) vers un bloc CN standard

Description : Poursuit sans conditions le déroulement du programme au point de destination du saut (label).

Les points suivants s'appliquent ici :

- La destination du saut doit être définie.
- La destination du saut doit se trouver entre le bloc-programme actif et le début du fichier.

Pour en savoir plus sur la programmation label, voir [chap. 3.11 "Programmation label et instructions de saut" à la page 64.](#)

Syntaxe :

GoBack <label>	
Format abrégé : GOB	
avec	
<label>	Nom de destination du saut 2 à 32 caractères Sont autorisés les lettres, les traits de soulignement et les chiffres, mais les 2 premiers caractères ne doivent pas être des chiffres.

Tab. 3-9: Syntaxe GoBack



Tenez compte du fait que, lors de la programmation de sauts en arrière, vous risquez de créer involontairement des boucles sans fin !

Notions de base pour la programmation CN

Exemple :

```

:
N40 LABEL1 :      Programmation label de la destination de saut "LA-
:                  BEL1".
N80 GoBack LABEL1 Saut en arrière vers la destination de saut "LABEL1".
:                  Si aucun autre ordre de saut n'est programmé entre
:                  N40 et N80, alors le programme se poursuit entre N40
:                  et N80 dans une boucle sans fin !
    
```



Il est interdit d'entrer dans ou de sortir d'une instruction de répétition CPL ou d'une instruction de branchement CPL à l'aide de l'ordre de saut CN standard GoBack.

Pour en savoir plus sur les instructions CPL, voir [chap. 3.11 "Programmation label et instructions de saut"](#) à la page 64.

3.11.5 Saut conditionnel GoCond (GOC) vers un bloc CN standard

Description : Poursuit le déroulement du programme au numéro de bloc indiqué dans la mesure où le signal d'entrée "Saut conditionnel" était actif sur l'interface du canal au moment du traitement du bloc.

Les points suivants s'appliquent ici :

- Le numéro de bloc indiqué doit exister.
- La position du numéro de bloc indiqué au sein du fichier de programme peut être choisie librement par rapport au bloc actuel du programme.

Syntaxe :

GoCond N<chiffre>	
Format abrégé : GOC	
avec	
<chiffre>	Numéro de bloc. Nombre au format entier ou réel avec 15 chiffres au maximum. Dans la mesure où le numéro du bloc cible est programmé avec les zéros non significatifs en tête (par ex. "N0020 . . ."), les zéros non significatifs en tête doivent également être programmés ici. ("GoCond N0020").

Tab. 3-10: Syntaxe GoCond

Exemple :

```

10 WAIT      Arrêter le traitement du bloc jusqu'à ce que tous les
N20 GoCond N090 blocs avant N20 aient été traités.
:
:           Ensuite, saut vers le bloc N090 si le signal d'interface
:           "Saut conditionnel" est actif au moment du traitement
:           du bloc de N20.
N090 ...
:
    
```

Notions de base pour la programmation CN



Il est interdit d'entrer dans ou de sortir d'une instruction de répétition CPL ou d'une instruction de branchement CPL à l'aide de l'ordre de saut CN standard GoCond.

Pour en savoir plus sur les instructions CPL, voir [chap. 3.11 "Programmation label et instructions de saut"](#) à la page 64.

Particularités et restrictions :

- À l'exception d'un numéro de bloc éventuellement programmé en tête, aucun autre mot de programme n'est autorisé dans le même bloc.
- Les modifications du signal d'interface du canal "Saut conditionnel" apportées entre la préparation et l'exécution du bloc ne sont pas prises en compte.

Dans la mesure où ce comportement n'est pas admissible pour votre application, vous devez programmer la fonction "WAIT" sur la ligne de programme précédente.



Tenez compte du fait que, lors de la programmation de sauts vers le début du fichier, vous risquez de créer involontairement des boucles sans fin !

3.11.6 Saut inconditionnel GoTo vers un bloc CN standard

Description : Poursuit sans conditions le déroulement du programme avec un numéro de bloc quelconque.

Les points suivants s'appliquent ici :

- Le numéro de bloc indiqué doit exister.
- La position du numéro de bloc indiqué au sein du fichier de programme peut être choisie librement par rapport au bloc actuel du programme.

Syntaxe :

GoTo N<chiffre>	
avec	
<chiffre>	Numéro de bloc. Nombre au format entier ou réel avec 15 chiffres au maximum. Dans la mesure où le numéro du bloc cible est programmé avec les zéros non significatifs en tête (par ex. "N0020 . . ."), les zéros non significatifs en tête doivent également être programmés ici. ("GoTo N0020").

Tab. 3-11: Syntaxe GoTo

Exemple :

```

:
N40 GoTo N080          Saut en avant vers le bloc N080.
:
N080 GoTo N40          Saut en arrière vers le bloc N40.
:                      Si aucun autre ordre de saut n'est programmé entre
                        N40 et N080, alors le programme continue entre N40
                        et N080 dans une boucle sans fin !

```

Particularités et restrictions :

À l'exception d'un numéro de bloc éventuellement programmé en tête, aucun autre mot de programme n'est autorisé dans le même bloc.



Tenez compte du fait que, lors de la programmation de sauts vers le début du fichier, vous risquez de créer involontairement des boucles sans fin !



Il est interdit d'entrer dans ou de sortir d'une instruction de répétition CPL ou d'une instruction de branchement CPL à l'aide de l'ordre de saut CN standard GoTo.

Pour en savoir plus sur les instructions CPL, voir [chap. 3.11.1 "Vue d'ensemble"](#) à la page 64.

3.11.7 Saut CPL (GOTO) vers un bloc-programme quelconque

Description : Poursuit sans conditions le déroulement du programme au point de destination du saut.

Les points suivants s'appliquent ici :

- La destination de saut peut être un numéro de bloc CPL, un numéro de bloc CN standard ou un "label" (marqueur de saut).
- La position de la destination de saut au sein du fichier de programme peut être choisie librement par rapport au bloc actuel du programme.

Syntaxe :

GOTO <destination>

Tab. 3-12: Syntaxe GOTO

Exemple :

10 GOTO N20	Saut vers le bloc N20
N20 X100	
30 GOTO 120	Saut vers le bloc CPL 120
...	
120 GOTO .CIBLE1	Saut vers le label .CIBLE1
...	
150 .CIBLE1	



Il est interdit de sauter vers une instruction CASE-LABEL...LABEL-OTHERWISE-ENDCASE à l'aide d'un saut CPL GOTO.

Pour connaître les conditions de programmation label, voir [chap. 3.11 "Programmation label et instructions de saut"](#) à la page 64.

3.11.8 Saut en arrière RetTo dans un sous-programme vers un bloc-programme quelconque

Description : Exécute sans conditions un saut en arrière d'un ou plusieurs niveaux dans un sous-programme et poursuit le déroulement du programme avec le bloc cible voulu.

Les points suivants s'appliquent ici :

- Le bloc cible est un numéro de bloc ou un marqueur de saut (respectivement pour les blocs CN standards ou les blocs CPL).
- La position du bloc cible au sein du fichier de programme cible par rapport à l'emplacement de l'appel de sous-programme n'a pas d'importance.
- Si aucun paramètre autre que le bloc cible n'est indiqué, il se produit alors un saut en arrière d'exactly un niveau de programme.

Notions de base pour la programmation CN

- En option, un saut en arrière de plusieurs niveaux de programme est aussi possible. Pour cela, il est possible d'indiquer le niveau cible à l'aide d'un paramètre de niveau indiquant soit l'écart par rapport au niveau de programme actuel (valeur négative), soit le niveau de programme en valeur absolue (valeur positive).

Syntaxe :

RetTo (<bloc cible>{, <niveau cible>})	
avec	
<bloc cible>	Numéro de bloc ou marqueur de saut (respectivement pour les blocs CN standards ou les blocs CPL)
<niveau cible>	Niveau de programme absolu ou incrémental (optionnel) > 0 : Saut en arrière vers le niveau de programme indiqué en valeur absolue = 0 : Saut en arrière vers le programme principal < 0 : Saut en arrière égal au nombre de niveaux de programme indiqué Si aucun niveau cible n'est indiqué, cela revient à indiquer -1, soit un saut en arrière d'un niveau de programme.

Tab. 3-13: Syntaxe RetTo

Particularités et restrictions : L'ordre RetTo n'est pas autorisé

- dans les programmes d'initialisation de canaux ("programme d'initialisation") et leurs sous-programmes ;
- dans le programme principal ;
- en tant que saisie directe au type de fonctionnement "saisie manuelle", ou dans le premier niveau de programme d'un programme appelé là.

De plus, l'ordre RetTo entraîne une erreur quand :

- aucun bloc cible n'est indiqué ;
- le bloc cible est introuvable ;
- le niveau cible indiqué en valeur absolue est inférieur au niveau de programme actuel ;
- le nombre de niveaux de programme cible indiqué en valeur incrémentale dans le saut en arrière est supérieur au niveau de programme actuel.

Les sous-programmes locaux constituent un niveau de programme autonome et sont donc traités comme tous les autres sous-programmes.

Exemples :

N100 RetTo (N500)	Saut en arrière d'un niveau vers le bloc CN standard N500
N100 RetTo (500)	Saut en arrière d'un niveau vers le bloc CPL 500
N100 RetTo (STDLABEL)	Saut en arrière d'un niveau vers le bloc CN standard avec le marqueur de saut "STDLABEL"
N100 RetTo (.CPLLABEL)	Saut en arrière d'un niveau vers le bloc CPL avec le marqueur de saut ".CPLLABEL"
N100 RetTo (N500, -1)	Saut en arrière d'un niveau vers le bloc CN standard N500 (identique au premier exemple)

Notions de base pour la programmation CN

N100 RetTo (500, -2)	Saut en arrière de deux niveaux vers le bloc CPL 500
N100 RetTo (STDLABEL, 2)	Saut en arrière jusqu'au niveau de programme 2 vers le bloc CN standard avec le marqueur de saut "STDLABEL"
N100 RetTo (.CPLLABEL, 0)	Saut en arrière dans le programme principal vers le bloc CPL avec le marqueur de saut ".CPLLABEL"



Il est absolument interdit de sauter dans une instruction CASE-LABEL...LABEL-OTHERWISE-ENDCASE à l'aide de l'ordre RetTo.



Dans le cas d'un saut en arrière vers un marqueur de saut de bloc CN standard, il est interdit de sauter dans une instruction CASE-LABEL...LABEL-OTHERWISE-ENDCASE à l'aide de l'ordre RetTo.

Pour connaître les conditions de programmation label, voir [chap. 3.11 "Programmation label et instructions de saut"](#) à la page 64.

3.12 Instructions de décision et de branchement

3.12.1 Vue d'ensemble

Les instructions de décision et de branchement servent à exécuter des blocs-programmes et extraits de programme individuels ou des sous-programmes complets en fonction de certains événements.

À cet effet, la commande IndraMotion MTX offre les possibilités suivantes :

- Fonction "Sauter séquence" pour les blocs CN standards (DIN)
- Instruction CPL IF-THEN-ELSE-ENDIF
- Instruction CPL CASE-LABEL...LABEL-OTHERWISE-ENDCASE

3.12.2 Fonction "Sauter séquence"

Description : Cette fonction permet à la commande d'ignorer des blocs CN standards (DIN). Pour ce faire, il suffit de programmer le caractère "/" au début des lignes de programme correspondantes.

Les blocs-programmes marqués ne sont ignorés que si le signal d'interface "qCh_BlockSlash" (Sauter séquence) est défini pour l'interface bit du canal concerné.

Exemple :

:	Le signal d'interface "Sauter séquence" est activé.
:	
/N100 ...	Le bloc N100 est ignoré.
:	Le signal d'interface "Sauter séquence" est désactivé.
:	
/N300 ...	Le bloc N300 est exécuté.
:	

Notions de base pour la programmation CN



La fonction "Sauter séquence" ne peut être utilisée que pour les blocs CN standards (DIN) !

3.12.3 Instruction CPL : CASE-LABEL...LABEL-OTHERWISE-ENDCASE

Description : Il est souvent nécessaire d'interroger **plus de 2 états** d'une expression INTEGER ou d'une variable INTEGER au sein d'un programme. Dans ce cas, une interrogation à l'aide de l'instruction IF n'est possible qu'à l'aide de plusieurs instructions IF imbriquées. Ceci requiert un temps de calcul élevé et réduit la clarté et la facilité d'entretien du programme.

La structure CASE permet d'éviter ces inconvénients :

```
CASE <expression à nombre entier> OF
LABEL <constante INT>[ , <constante INT>][ : <instruction>]
<instruction>
:
LABEL ...
:
[OTHERWISE <instruction>
<instruction>
:]
ENDCASE
```

Le déroulement du programme est ramifié après l'instruction CASE vers l'instruction LABEL pour laquelle une des <constantes INT> est égale à la valeur de l'<expression à nombre entier>. Toutes les instructions sont à présent exécutées jusqu'à la prochaine instruction LABEL ou OTHERWISE. Ensuite, le programme passe directement à l'instruction ENDCASE.

S'il n'existe aucune instruction LABEL remplissant cette condition, le programme est ramifié vers l'instruction OTHERWISE ou directement vers l'instruction ENDCASE (si OTHERWISE n'a pas été programmé).

Toutes les instructions CPL peuvent être utilisées dans le domaine <instruction> d'une structure CASE. Il est possible d'imbriquer un maximum de 10 structures CASE.

Exemples :

```
10 CASE A% OF
20 LABEL 0 : Y=1
30 LABEL 2
40 Y=Y*Y
50 LABEL 4 : Z=Y*Y
60 Y=Z*Z
70 OTHERWISE Y=0
80 ENDCASE
```

```
10 CASE (INT(X/Y)+C%) OF
20 LABEL 1,2 : X=1 : Y=2
30 LABEL 4.8
40 X=2 : Y=4
50 LABEL 0
```

Notions de base pour la programmation CN

```

60 X=0 : Y=1
70 OTHERWISE X=0 : Y=0
80 ENDCASE

10 CASE INTFELD%(1,2) OF
20 LABEL 1,2,3 : GOTO .MARKE1
30 LABEL 4,5,6 : GOTO .MARKE2
40 OTHERWISE GOTO .ENDE
50 ENDCASE

```



Il est interdit d'entrer dans ou de sortir d'une instruction CASE-LABEL...LABEL-OTHERWISE-ENDCASE à l'aide des ordres de saut CN standards GoAhead, GoBack, GoCond, GoTo.



Il est interdit de sauter vers une instruction CASE-LABEL...LABEL-OTHERWISE-ENDCASE à l'aide d'un saut CPL GOTO.

3.12.4 Instruction CPL : IF-THEN-ELSE-ENDIF

Description : Cette fonction est une instruction de branchement conditionnée simple :
"Si (IF) une condition définie est remplie, alors (THEN) une routine est exécutée, sinon (ELSE) une autre routine est exécutée !"

Syntaxe :

IF <condition> THEN <routine> [ELSE <routine alternative>]	
ENDIF	<i>Les points suivants s'appliquent ici :</i> • La condition se trouve sur la même ligne que "IF" et se termine sur la même ligne par "THEN". • Les routines THEN et ELSE sont des branches du programme qui ne sont pas nécessairement parcourues dans tous les cas. • Si l'on renonce à la partie ELSE, le programme poursuit son déroulement immédiatement après l'instruction ENDIF lorsque la condition n'est pas remplie.

Tab. 3-14: Syntaxe IF-THEN-ELSE-ENDIF

Comme pour les conditions d'interruption des instructions de boucles, les liaisons arithmétiques, trigonométriques et logiques peuvent être utilisées dans la condition de l'ordre IF. Une imbrication est également possible.

L'ordre IF doit toujours être terminé par une instruction ENDIF, sans quoi la fin de la routine ou de la routine alternative ne sera pas reconnue. Étant donné que le positionnement de l'instruction ENDIF dépend de la logique du déroulement du programme, la commande ne peut pas toujours reconnaître clairement l'absence d'une instruction ENDIF. Cela entraîne alors des messages d'erreur trompeurs. C'est pourquoi le programmeur doit toujours vérifier l'intégralité de l'ordre IF.

Exemple :

```

...
10 X = 1
20 .START
30 IF X>=100 THEN

```

Notions de base pour la programmation CN

```

40 GOTO .ENDE
50 ELSE X=X+2.75
60 GOTO .START
70 ENDIF
...
90 .FIN
...

```



Il est interdit d'entrer dans ou de sortir d'une instruction IF-THEN-ELSE-ENDIF à l'aide des ordres de saut CN standards GoAhead, GoBack, GoCond, GoTo.

3.13 Instructions de répétition

3.13.1 Vue d'ensemble

Si un ou plusieurs blocs-programmes doivent être traités à plusieurs reprises selon certaines conditions, il est possible de programmer cela à l'aide d'**instructions de répétition CPL**. L'exécution d'un programme à plusieurs reprises est également qualifiée de boucle.

À cet effet, la commande IndraMotion MTX offre les possibilités suivantes :

- Instruction CPL FOR-STEP-TO_NEXT
- Instruction CPL REPEAT-UNTIL
- Instruction CPL WHILE_DO_END

3.13.2 Instruction CPL : FOR-STEP-TO-NEXT

Description : Si la condition d'interruption de l'instruction de répétition découle directement du traitement de la routine, alors un compteur tournant en parallèle, par exemple, est nécessaire.

Celui-ci ne nécessite aucune programmation spéciale lors de la boucle FOR-NEXT. Il faut alors fixer une variable de comptage (nombre entier) dont les états de début et de fin doivent être indiqués. Si le pas de comptage est différent de 1, la longueur de pas (STEP) peut être déterminée à part.

Syntaxe :

FOR <var. comptage>=<valeur initiale> [STEP <longueur de pas>]
TO <valeur finale><routine>
NEXT [<variable de comptage>]

Tab. 3-15: Syntaxe FOR-STEP-TO-NEXT

Exemple :

```

10 FOR I%=1 TO 18
20 XSINUS(I%)=SIN(I%*10)
30 NEXT I%

```

À la fin de la boucle, la variable de comptage a une valeur supérieure à la valeur de fin (longueur de pas max.).

Ici, les valeurs sinus de 0 à 180 degrés sont inscrites dans le champ XSINUS. Le "I%" ajouté à "NEXT" sur la ligne 30 n'a qu'un rôle de clarification et peut éventuellement être omis.

Il est également possible de programmer des boucles FOR-NEXT avec une **longueur de pas variable**. La variable de la longueur de pas doit alors être de même type que la variable de comptage.

Exemple : *Programme:*

```
10 OPENW(1,"P222",130)
20 PAS%=2 : DEBUT%=1 : FIN%=3500 : NJUST
30 POUR COMPTEUR%=DEBUT% PAS PAS% À LA FIN%
40   PAS%=ROUND(PAS%*SQRT(PAS%))
50   PRN#(1,"COMPTEUR : ",COMPTEUR%,"LONGUEUR DE PAS : " ,PAS%)
60   NEXT
70 CLOSE(1)
```

À la fin de ce programme, il est inscrit ce qui suit dans le fichier "P222" :

Exemple : *Programme:*

```
COMPTEUR :      LONGUEUR DE PAS :      3
COMPTEUR :      4 LONGUEUR DE PAS :      5
COMPTEUR :      9 LONGUEUR DE PAS :     11
COMPTEUR :     20 LONGUEUR DE PAS :     36
COMPTEUR :     56 LONGUEUR DE PAS :    216
COMPTEUR :    272 LONGUEUR DE PAS :   3175
COMPTEUR :   3447 LONGUEUR DE PAS : 178902
```



Il est interdit d'entrer dans ou de sortir d'une instruction FOR-STEP-TO-NEXT à l'aide des ordres de saut CN standards GoAhead, GoBack, GoCond, GoTo.

3.13.3 Instruction CPL : REPEAT-UNTIL

Description : Si la condition d'interruption pour l'instruction de répétition ne doit être interrogée qu'après le premier traitement de la routine, il est possible d'utiliser la boucle REPEAT.

Syntaxe :

```
REPEAT <routine> UNTIL <condition>
```

Tab. 3-16: Syntaxe REPEAT-UNTIL

Exemple :

```
:
30 REPEAT
40 X=X+1          Boucle jusqu'à X = 100
50 UNTIL X=100
:
```



Il est interdit d'entrer dans ou de sortir d'une instruction REPEAT-UNTIL à l'aide des ordres de saut CN standards GoAhead, GoBack, GoCond ou GoTo.

3.13.4 Instruction CPL : WHILE-DO-END

Description : Si la condition d'interruption de l'instruction de répétition doit être interrogée avant la première exécution de la boucle, il est possible de le formuler comme suit : "Tant que (⇒ « while » en anglais) la condition est remplie, la routine est exécutée (⇒ « to do » en anglais) !"
La boucle WHILE est structurée de la manière suivante :

Syntaxe :

```
WHILE <condition> DO <routine> END
```

Tab. 3-17: Syntaxe WHILE-DO-END

Exemple :

```
:
30 WHILE DS(9)=0 DO      Boucle d'attente jusqu'à ce que DS(9) ait la valeur 0.
```

Notions de base pour la programmation CN

```
40 I=I+1
```

```
50 END
```

```
:
```

Tab. 3-18:



Il est interdit d'entrer dans ou de sortir d'une instruction WHILE-DO-END à l'aide des ordres de saut CN standards GoAhead, GoBack, GoCond, GoTo.

3.14 Programmation de variables

3.14.1 Noms des variables



La programmation de variables fait partie intégrante du langage CPL !

La programmation de variables en langage CPL permet le paramétrage des programmes et donc l'adaptation du déroulement du programme aux conditions actuelles.

- Les variables sont des noms de symboles quelconques auxquels s'appliquent cependant quelques conditions annexes particulières :
- - Les noms de variables doivent être univoques.
 - Les noms de variables ne doivent pas être identiques aux mots d'ordre CPL réservés.

En raison de la séparation formelle entre la programmation CN standard et la programmation CPL, les noms de variables peuvent théoriquement correspondre à des noms de fonctions CN ou à des paramètres de fonctions CN : il est par exemple possible de définir une variable "X" bien qu'il existe également un axe nommé "X" dans le système.

- Le nom de variable se compose d'une série quelconque de lettres majuscules et de chiffres, le premier caractère devant être une lettre majuscule. Un nom de variable se composant de la lettre "N" suivie par les chiffres n'est pas valable. Une telle suite de caractères correspond à l'identificateur d'un bloc CN.



Seuls les 8 premiers caractères du nom de variable sont significatifs, c.-à-d. que les noms de variables se distinguent les uns des autres seulement par leurs 8 premiers caractères (exception : variables permanentes définissables).

- Au total, il existe 3 groupes de variables déterminant la plage de validité des variables. Le groupe de variables est déterminé par un identificateur au début du nom de variable. Ce caractère compte toujours parmi les caractères significatifs du nom !

*Il existe les **groupes de variables** suivants avec leur identificateur correspondant :*

- Variables locales : aucun identificateur spécial
- Variables globales : "#"
- Variables permanentes : "@"

Notions de base pour la programmation CN

- Le **type de variable** est toujours déterminé par un identificateur spécifique à la fin du nom de la variable. Cela s'applique également si le nom de variable dépasse le nombre de caractères significatifs.

Toutefois, cet identificateur n'est pas nécessaire au sein des fichiers de déclaration pour les variables permanentes définissables ("wmhperm.dat" et "anwperm.dat"). Au lieu de cela, le **type de données CPL** correspondant doit être indiqué de manière explicite.

Il existe les types de variables suivants avec leur identificateur correspondant :

Type de variable	Type de données CPL	Identificateur de variable spécifique
Nombre entier	INT	"%"
Chiffre à virgule flottante avec simple précision	REAL	Aucun identificateur particulier
Chiffre à virgule flottante avec double précision	DOUBLE	"!"
Variable logique	BOOL	"?"
Caractères	CHAR	"\$"

Tab. 3-19: Variables

Exemples : Variables locales, globales et permanentes :

10 NOMBRE1% = 1	Variable locale de type CPL INT
20 #NOMBRE2% = 2	Variable globale de type CPL INT
30 @36% = 3	Variable permanente de type CPL INT
40 @ABCD% = 4	Variable permanente définie de type CPL INT

3.14.2 Groupes de variables

Introduction

Étant donné qu'il est possible d'utiliser des sous-programmes et éventuellement nécessaire d'enregistrer temporairement les valeurs des variables indépendamment du programme, il est nécessaire d'appliquer des conventions au domaine d'application des variables. À cet effet, on distingue les groupes de variables décrits ci-dessous.

Variables locales

Les variables locales ont un effet qui se limite au programme dans lequel elles sont convenues. À la fin de ce programme, ces variables sont effacées et l'espace mémoire occupé est libéré. Lors d'un appel de sous-programme, un nom de variable local pour le programme appelant n'est pas "visible" pour le sous-programme. C'est pourquoi il se peut que le nom de variable y soit convenu également de manière locale sans que les deux variables ne s'influencent mutuellement. Lors du retour dans le programme appelant, la variable locale originale est de nouveau disponible avec la valeur affectée immédiatement avant l'appel de sous-programme.

Variables globales

Les variables globales sont identifiées par le caractère # placé en tête. Après la première attribution d'une valeur à une variable globale, cette dernière peut être lue ou modifiée à partir de toutes les parties du programme pendant

Notions de base pour la programmation CN

toute la durée restante du programme complet. Les variables globales sont effacées à la fin du programme.

Variables permanentes

Les variables permanentes sont identifiées par le caractère @ placé en tête et suivi du nom de la variable. Elles peuvent être adressées par chaque programme actif. Les variables permanentes se trouvent dans une zone d'enregistrement à part et sont normalement conservées à la fin de tout programme, en cas de remise à zéro et après la mise hors/en service. Elle ne peuvent être effacées que par un écrasement ciblé.

Les variables permanentes de type CPL INT peuvent être adressées par les désignations @1 à @100 (pour connaître la signification du type CPL INT, voir [chap. 3.14.3 "Types de variables" à la page 82](#)). Afin d'améliorer la lisibilité du programme, la désignation de telles variables permanentes peut être complétée en ajoutant des lettres au chiffre.

De plus, il est possible d'utiliser la variable de tableau unidimensionnelle permanente @_R avec 100 éléments de type CPL DOUBLE. Les deux variables permanentes @_RES_DOUBLE et @_RES_DWORD sont réservées à des applications internes et ne doivent pas être utilisées.

Variables permanentes définissables

Les variables permanentes définissables sont également identifiées par le caractère @ placé en tête et suivi du nom de la variable.

Les différences par rapport aux "variables permanentes" sont les suivantes :

- Elles ne font pas automatiquement partie du logiciel du système, mais doivent être **déclarées manuellement** par inscription dans les fichiers "wmhperm.dat" (pour les données spécifiques à WMH) et "anwperm.dat" (pour les données spécifiques aux utilisateurs finaux). Vous trouverez la syntaxe pour la déclaration sous le point « Structure des fichiers "wmhperm.dat" et "anwperm.dat" ».

Au cours du démarrage, la commande recherche d'abord ces fichiers dans le répertoire racine, puis dans le FEPROM utilisateur et enfin dans le FEPROM.

La commande évalue le premier fichier trouvé pour chaque nom, puis elle génère des "variables permanentes définissables" à partir des inscriptions s'y trouvant, dans la mesure où elles n'existent pas encore. Les "variables permanentes définissables" existantes qui ne sont déclarées dans aucun des deux fichiers sont effacées.

Le nombre maximal autorisé de variables permanentes définissables est limité par l'espace mémoire disponible. S'il n'y a plus d'espace mémoire libre pour la génération des variables, la commande émet alors un message d'erreur correspondant.

- Les noms des "variables permanentes définissables" commencent toujours par le **caractère @ suivi d'une chaîne de caractères**. Cette chaîne de caractères se compose d'une lettre majuscule suivie de lettres majuscules ou de chiffres quelconques.

Pour les "variables permanentes définissables", seules les **16 premiers caractères** du nom de variable sont significatifs. Si elles ne se distinguent qu'à partir du 17e caractère, CPL les interprète comme une seule variable !

- Les variables permanentes définissables peuvent être de type CPL **INT**, **REAL**, **DOUBLE**, **BOOL** ou **CHAR**. Le type de variable est déterminé par un identificateur placé à la fin du nom de variable.

Notions de base pour la programmation CN

Cet identificateur doit être indiqué dans le programme pièce :

- @ABCD% Variable perm. déf. de type CPL INT
- @EFGH Variable perm. déf. de type CPL REAL (sans identificateur particulier)
- @IJKL! Variable perm. déf. de type CPL DOUBLE
- @MNOP? Variable perm. déf. de type CPL BOOL
- @QRST\$ Variable perm. déf. de type CPL CHAR
- Il est possible d'utiliser des **tableaux** unidimensionnels et bidimensionnels.

Pour les variables de tableau de type CPL INT, REAL, DOUBLE ou BOOL, l'**index de tableau maximal** s'élève à **65 535**. Pour les variables de tableau de type CPL CHAR, il s'élève à **1024**.

Exemples :

@WZNR% (1) = 4	La valeur 4 est attribuée à la première variable (ayant l'index 1) du tableau unidimensionnel @WZNR de type CPL INT.
@WZKOR (2, 2) = 0.2	La valeur 0.2 est attribuée à la variable (ayant l'indice 2,2) au sein du tableau bidimensionnel @WZKOR de type CPL REAL.

- Estimation du **nombre** de variables permanentes définissables disponible :

Espace mémoire total dédié aux variables permanentes : 100 kilooctets (102 400 octets)

Pos.	Réservé pour	Espace mémoire en octets	Remarque
1	toutes les variables permanentes	102 400	Espace mémoire total
Répartition réservée de ces octets			
2	@1 à @100 (variables permanentes)	800	Type de données : INT Afin d'améliorer la lisibilité du programme, la désignation de telles variables permanentes peut être complétée en ajoutant des lettres au chiffre.
3	les informations administratives	24	
4	toutes les variables permanentes définissables	101 576	(4) = (1) - (2) - (3)
Répartition réservée de ces octets			
5	@_R	823	Variable de tableau permanente avec 100 éléments du type CPL DOUBLE
6	@_RES_DOUBLE	40	Variable permanente de type CPL DOUBLE, réservée aux applications internes

Notions de base pour la programmation CN

Pos.	Réservé pour	Espace mémoire en octets	Remarque
7	@_RES_DWORD	35	Variable permanente de type CPL INT, réservée aux applications internes
8	les nouvelles variables permanentes définissables	100 678	(8) = (4) - (5) - (6) - (7)

Tab. 3-20: Espace mémoire dédié à toutes les variables permanentes

Pos.	Réservé pour	Espace mémoire en octets	Remarque
9	le nom des variables permanentes définissables	max. 16	1 octet par caractère
10	la valeur de la variable permanente définissable	1, 4 ou 8	INT : 4 octets DOUBLE : 8 octets REAL : 4 octets BOOL : 1 octet
11	les informations administratives	20	
12	une variable permanente définissable de type CPL DOUBLE avec un nom composé de 16 caractères	44	Exemple : occupation maximale de l'espace mémoire (9) + (10) + (11)

Tab. 3-21: Espace mémoire dédié aux variables permanentes définissables

Nombre de "variables permanentes définissables" des types CPL DOUBLE et INT :

Type de variable	Nombre de variables	Remarque
Type CPL DOUBLE avec un nom composé de 16 caractères au maximum	2288	$100\,678/44 = 2288$
Type CPL INT avec un nom composé de 16 caractères au maximum	2516	$100\,678/(16+4+20) = 2516$
Type CPL INT avec un nom composé de 8 caractères au maximum	3146	$100\,678/(8+4+20) = 3146$
Variables de tableau avec un nom composé de 16 caractères au maximum Type CPL INT	25 160	$(100\,678-16-20)/4 = 25\,160$
Variables de tableau avec un nom composé de 16 caractères au maximum Type CPL DOUBLE	12 580	$(100\,678-16-20)/8 = 12\,580$

Tab. 3-22: Nombre des "variables permanentes définissables"

Structure des fichiers de déclaration "wmhperm.dat" et "anwperm.dat" :

Les fichiers ne doivent contenir que les déclarations des "variables permanentes définissables". Chaque déclaration de variable est effectuée sur une ligne à part et doit se terminer par un **point-virgule**. Il est ensuite possible de saisir un commentaire optionnel à la fin de la ligne.

Une ligne de déclaration est toujours structurée de la manière suivante :

```
DEF <type de variable> @ <nom de variable> ; [<commentaire>]
```



Le caractère de séparation ";" doit être inscrit même si aucun commentaire n'est saisi !

Exemples de variables "wmhperm.dat" et "anwperm.dat"

```
DEF INT @CDEF
DEF INT @ABCD           ; variable simple de type CPL INT
DEF REAL @EFGH          ; variable simple de type CPL REAL
DEF DOUBLE @IJKL        ; variable simple de type CPL DOUBLE
DEF BOOL @MNOP           ; variable simple de type CPL BOOL
DEF CHAR @PSTR1(3)       ; variable de type CPL CHAR de longueur 3
DEF INT @WZNR(9)         ; tableau unidimensionnel de type CPL INT avec 9 variables
DEF REAL @WZKOR(9,2)     ; tableau bidimensionnel de type CPL REAL avec 18 variables
DEF CHAR @PSTR2(9,2)     ; tableau bidimensionnel de type CPL CHAR avec 9 chaînes partielles de 2 caractères chacune
```

Exemples d'application des variables permanentes :

```
10 @1 = 1
15 @2_COMPTEUR = 2
20 @ABCD% = 3
25 @EFGH = 4.1
30 @IJKL! = 5.12345
35 @MNOP? = TRUE
40 @PSTR1$ = "ABC"
45 @WZNR%(2) = 6
50 @WZKOR(3,2) = 7.6
55 @PSTR2$(3) = "DE"
```

Variables structurées

Les variables structurées peuvent être remplacées par les données système (DS), mais elles doivent être conservées pour des motifs de compatibilité (voir [chap. 7 "Ordres CPL" à la page 449](#)).

Les variables structurées sont identifiées par différents niveaux de la structure qui sont séparés par un point (".") dans la présentation. Les variables structurées commencent toujours avec "SV."

Toutes les variables structurées devant être accessibles dans une instruction CPL doivent être **déclarées** dans le fichier "machdef.dat" par une saisie **manuelle**. Au cours du démarrage, la commande recherche d'abord le fichier "machdef.dat" dans le répertoire racine, puis dans le FEPRM utilisateur et enfin dans le FEPRM. Le premier fichier trouvé est évalué lors du démarrage de la commande numérique, puis les données administratives et la mémoire dédiée aux données utiles des variables structurées sont créées. La mémoire est créée de nouveau à chaque montée en régime et toutes les valeurs sont remises à 0.

Le fichier "machdef.dat" sert uniquement à définir les variables structurées. Chaque définition se trouve sur une ligne à part et est terminée par un point-virgule.

Une ligne de définition est toujours structurée de la manière suivante :

Notions de base pour la programmation CN

```
DEF<type de variable> SV.<nom de variable> ; [<commentaire>]
```

Ici, <type de variable> et <nom de variable> sont des chaînes pouvant contenir des lettres majuscules et minuscules, des chiffres, des traits d'union et des traits de soulignement.



Dans tous les noms de variables, seuls les 14 premiers caractères du nom de variable (hormis "SV." en tête) sont significatifs. Si deux noms se distinguent uniquement à partir du 15e caractère, CPL ou l'API les interpréteront comme une seule et même variable !

Exemple :

```
DEF DBT1Rec_t SV.A; Définition d'une variable structurée de type
"DBT1Rec_t"
```

Les instructions CPL peuvent accéder non seulement à SV.A, mais également à tous les composants de SV.A. Les différents composants sont séparés les uns des autres par un ".".

```
10 SV.A.Hd=DBSEA("DBT1",-1,-1,"K1=1",FOUND%)
```

3.14.3 Types de variables

Variable en nombre entier (INT)

Une variable de type CPL INT occupe 32 bits de mémoire. Elle est identifiée par le caractère "%" ajouté au nom de la variable. La plage de valeurs s'étend de -2 147 483 647 à +2 147 483 647.

```
10 NOMBRE% = 4
```

Variable INTEGER

Variable à virgule flottante (REAL)

Si aucun identificateur particulier n'est placé après le nom de variable, alors la variable est interprétée comme une variable REAL de précision simple.

Dans ce cas, la variable occupe 32 bits de mémoire. La plage de valeur s'élève à $\pm 10^{38}$. Ceci correspond à 7 chiffres significatifs.

```
10 PI = 3.141593
```

Variable REAL de précision simple

Variable à virgule flottante (DOUBLE)

Si le caractère "!" est placé à la fin du nom de variable, alors la variable est interprétée comme une variable REAL de **précision double**.

Dans ce cas, la variable occupe 64 bits de mémoire. La plage de valeurs s'élève à $\pm 10^{308}$. Ceci correspond à 15 chiffres significatifs.

```
10 PI! = 3.141592653589793
```

Variable REAL de précision double

Variable logique (BOOL)

Ces variables sont identifiées par l'ajout du caractère "?" à la fin du nom de variable. Les variables logiques (variables booléennes) ne peuvent avoir que la valeur **TRUE** (= vrai) ou **FALSE** (= faux, non vrai). Les états ou conditions logiques requis pour la suite de l'exécution du programme y sont mémorisés.


```
10 START? = FALSE
```

Variable BOOLE

Variable de tableau (ARRAY)

Les variables ARRAY permettent de réserver un tableau unidimensionnel ou bidimensionnel (array) composé de plusieurs variables de même type dans la zone de mémoire sous un nom de variable unique.

Les définitions du tableau sont possibles pour les variables de type CPL INT, REAL, DOUBLE, BOOL et CHAR. Pour accéder aux différents éléments d'un tableau, il convient d'indiquer non seulement le nom de la variable de tableau, mais également son index ou son indice.

Exemple : Dimensionnement d'une variable ARRAY

```
10 DIM    TABLEAU (2,3)
```

Constantes INTEGER pour les tailles du tableau (index)
 Nom de variable (variable REAL)
 Mot d'ordre DIM

Accès à la variable ARRAY

```
100 VARTABLEAU(1,1) = MCS(1)
110 VARTABLEAU(2,1) = PCS(1)
120 VARTABLEAU(1,2) = MCS(2)
130 VARTABLEAU(2,2) = PCS(2)
140 VARTABLEAU(1,3) = MCS(3)
150 VARTABLEAU(2,3) = PCS(3)
```

Avant le premier accès à la variable de tableau, il convient de dimensionner la plage de l'index ou la taille du tableau avec une constante en nombre entier :

- Taille du tableau d'une variable de tableau de type CPL INT et REAL :
max. 65 536
- Taille du tableau d'une variable de tableau de type CPL CHAR :
max. 1024

DIM <nom de variable> (<taille du tableau1>[,<taille du tableau2>])



Le dimensionnement avec DIM ne doit pas être utilisé pour les "variables permanentes définissables". Ces variables sont dimensionnées dans le fichier "wmhperm.dat" ou "anwpwerm.dat".

Variables de type CPL CHAR et variables de chaîne

Une variable de type CPL CHAR est identifiée par le caractère "\$" placé en tête. Il est possible d'enregistrer dans ce type de variable aussi bien un caractère unique qu'une chaîne de caractères (STRING) complète.

Cependant, les instructions relatives aux chaînes de caractères (voir section "Traitement des chaînes de caractères") ne sont possibles que si une chaîne de caractères est enregistrée dans un tableau (array) unidimensionnel ou bidimensionnel de variables de type CPL CHAR. Pour ce faire, le tableau doit être déclaré à l'aide de l'instruction DIM.

Notions de base pour la programmation CN

Chaque variable de type CPL CHAR enregistrée dans ce tableau contient alors 1 seul caractère de la chaîne de caractères.

Un tableau unidimensionnel de type CPL CHAR est appelé variable STRING. Aucun index n'est indiqué lors de l'accès à des variables unidimensionnelles de type CPL CHAR. En revanche, un index doit être indiqué lors de l'accès à des variables bidimensionnelles.

Exemple :

```

1 REM variable STRING AB (longueur 10)
2 DIM AB$(10)
3 REM 3 variables STRING CD (longueur 5 chacune)
4 DIM CD$(3,5)
5 AB$ = "Z"
6 CD$(2) = "ABC"

```

Vue d'ensemble des variables

Groupe de variables	Nom de variable	Type CPL		Tableaux (AR-RAYS) possibles (X = oui)
Locales	max. 8 caractères significatifs	%	INT	X
			REAL	X
		!	DOUBLE	X
		?	BOOL	X
		\$	CHAR	X
Globales#	max. 8 caractères significatifs, caractère "#" inclus	%	INT	X
			REAL	X
		!	DOUBLE	X
		?	BOOL	X
		\$	CHAR	X
Permanentes@	1 à 100			
Permanentes définissables@	max. 16 caractères significatifs	%	INT	X
			REAL	X
		!	DOUBLE	X
		?	BOOL	X
		\$	CHAR	X

Tab. 3-23: Vue d'ensemble des variables

3.14.4 Variables en programmation CN standard

Il est souvent souhaitable de paramétrer des fonctions CN ou des paramètres de fonctions CN à l'aide de variables dans un programme pièce au sein de la programmation CN standard (DIN).

Lors d'une telle attribution de valeurs, il convient de veiller à ce que les expressions CPL au sein de la programmation CN standard soient inscrites entre crochets "[" et "]".

Exemple : Paramétrage de blocs CN standards :

Notions de base pour la programmation CN

```
10 ANGLE = 45
20 VAL1 = 1.5
30 VAL2 = 1.5
40 XPOS = 10.2
50 YPOS = 5.73
60 FEEDRATE = 1000
N70 Rotate([ANGLE])
N80 Scale (X[VAL1],Y[VAL2])
N90 G1 X[POS] Y[POS] F[FEEDRATE]
```



Il n'est pas possible de paramétrer le numéro de bloc à l'aide de variables CPL !



Toutes les adresses appelant un sous-programme ne sont pas destinées à l'écriture variable !

3.14.5 Variable ERRNO pour l'analyse des erreurs des fonctions CPL "ERRNO"

Il est possible de transmettre la variable CPL ERRNO dans laquelle la fonction appelée annonce une erreur en tant que paramètre de transfert à une fonction CPL qui génère une erreur d'exécution en cas d'erreur.

La variable CPL ERRNO est un paramètre optionnel qui peut être programmé dans les fonctions CPL dans lesquelles ERRNO est déclarée en tant que paramètre. Elle peut alors y être programmée en n'importe quelle position.

Si ERRNO n'est pas programmée, la fonction CPL génère une erreur d'exécution ou un avertissement en cas d'erreur.

Lors de la programmation de ERRNO, la fonction CPL ne génère **aucune** erreur d'exécution, ni aucun avertissement. Dans ce cas, la variable CPL ERRNO doit être analysée dans le programme CPL, et une erreur d'exécution ou un d'avertissement doit être générée explicitement, respectivement avec SETERR ou SETWARN.

En cas d'exécution correcte de la fonction CPL, ERRNO a la valeur 0. Les erreurs sont annoncées par des valeurs négatives. La signification des valeurs négatives est généralement valide. Cependant, toutes les valeurs ne peuvent pas s'appliquer à toutes les fonctions CPL. Les valeurs d'erreur possibles pour une fonction CPL sont mentionnées avec la fonction correspondante.

Liste des valeurs d'erreur généralement valables :

0 :	Accès OK
-1 :	Erreur de paramètre
-2 :	Coordonnée/Axe/Broche inexistants
-3 :	Coordonnée/Axe non admissibles sur le canal
-4 :	L'axe n'est pas une pseudo-coordonnée.
-5 :	Canal inexistant
-6 :	Appel de la fonction possible uniquement dans son propre canal
-7 :	Échec de la lecture des données
-8 :	Nom du fichier source, chemin inclus, trop long

Notions de base pour la programmation CN

-9 :	Impossible d'accéder au fichier source
-10 :	Nom du fichier cible, chemin inclus, trop long
-11 :	Nom de fichier (source ou cible) inadmissible
-12 :	Copie impossible
-13 :	Échec de l'écriture des données
-14 :	La table XML existe.
-15 :	Table XML introuvable
-16 :	Extension de fichier non valide
-17 :	Type de table non valide
-18 :	Position d'insertion erronée
-19 :	Nombre maximal d'axes dépassé
-20 :	Root-Tag erroné
-21 :	Impossible d'accéder au fichier
-22 :	Clé non valide
-23 :	Ensemble de données bloqué
-24 :	Aucun droit de lecture pour le fichier
-25 :	Aucun droit d'écriture pour le fichier
-26 :	Ordre avec paramètre invalide
-27 :	Ordre avec type de paramètre invalide
-28 :	Erreur d'accès au tableau XML
-29 :	Faux index du tableau
-30 :	Mémoire insuffisante.
-31 :	Condition de recherche erronée
-32 :	Un bloc de données de ce tableau de banque de données est verrouillé dans ce canal.
-33 :	Le bloc de données n'est pas verrouillé dans le canal.
-34 :	Variable non affectée sur le côté droit de l'attribution
-35 :	Erreur interne lors de l'accès aux données système
-36 :	Canal de service Sercos bloqué
-37 :	Numéro d'identification Sercos inconnu
-38 :	Numéro d'identification Sercos non valide
-39 :	Numéro d'axe non valide
-40 :	Bloc d'interruption (Freeze Block) inexistant
-41 :	Instruction non autorisée
-42 :	Valeur limite dépassée
-43 :	Nombre d'éléments non valide
-44 :	Aucune valeur mesurée existante pour l'axe
-45 :	Impossible de calculer la valeur mesurée des coordonnées. Absence de valeur mesurée pour au moins un axe membre de la transformation de l'axe
-46 :	Impossible de calculer la valeur mesurée des coordonnées. Absence de valeur mesurée pour au moins un axe participant à la rotation

Notions de base pour la programmation CN

Exemple : Lire la position de l'axe X du canal dans le premier canal. Pour cela, il est possible de programmer l'une des quatre lignes suivantes (alternatives), qui ont toutes le même effet :

```
10 POS = ACS ("X", 1, 1, ERRNO)
10 POS = ACS ("X", 1, ERRNO, 1)
10 POS = ACS ("X", ERRNO, 1, 1)
10 POS = ACS (ERRNO, "X", 1, 1)
```

3.15 Informations générales relatives au système

3.16 Mots d'ordre réservés

Les notions clés introduites ici doivent être insérées séparément ou être séparées par un caractère spécial. Cela permet de les distinguer en tant que mots d'ordre. Les mots d'ordre réservés ne doivent pas être utilisés lors de la sélection des noms de variables !

Exemple :	GOTO 10	Saut à la ligne 10
	GOTO10	Nom de symbole quelconque (variable) ; placé seul, il entraîne un message d'erreur "Erreur d'exécution 2167 = manque" parce que l'attribution d'une valeur est attendue pour la variable "GOTO10".

Notions clés :

Notions de base pour la programmation CN

A:	ABS ACOS ACS AND APOS ASC ASIN ATAN ATAN2 AXADR\$ AXINF AXO AXP	B:	BCD BIN BITIF	C:	CALL CASE CHR\$ CLOCK CLOSE CLRWARN COF COS CPROBE	D:	DATE DBLOAD DBMOVE DBSAVE DBSEA DBSEAX DBTAB DBTABX DBTABXL DCT DIM DIRCR DIRDEL DIRINF DO DPC
E:	ELSE END ENDIF ENDCASE EOF ERASE ERRNO	F:	FALSE FILEACCESS FILECOPY FILEDATE FILENO FILEPOS FILESIZE FOR FXC FXCR FXDEL FXINS	G:	GETERR GOTO	I:	IF INP# INSTR INT
L:	LABEL LEN LJUST	M:	MCA MCASE\$ MCODS MCOPS MCS MID\$ MMC	N:	NCF NEXT NJUST NOT NUL	O:	OF OPENR OPENW OTHERWISE OR

Notions de base pour la programmation CN

P:	PCS PCSPROBE PDIM PLC PMT PMV PPOS PRN# PROBE	R:	REM REPEAT REWRITE ROUND	S:	SCL SCS SCSL SCS3L SD SDLOAD SDR SDSAVE SEEK SETERR SETWARN SIN SPOS SQRT STEP STR\$	T:	TAN TCV THEN TIME TO TRIM\$ TRUE
U:	UNTIL	V:	VAL VARINF VERSINF\$	W:	WAIT WCS WHILE	X:	XOR XTAB XTABCR
Z:	ZOCDEL ZOCINS ZOT ZOTCR ZOV						

Tab. 3-24: Notions clés

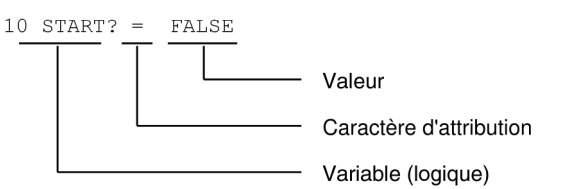
3.17 Instruction CPL

3.17.1 Attribution de valeur

Attribution

Il est possible d'attribuer des valeurs aux variables globales et locales. Pour cela, il faut utiliser le signe d'égalité "=".

Exemples : Attribution d'une valeur à une variable booléenne



Attribution d'une valeur à une variable à virgule flottante

Notions de base pour la programmation CN

```
1 X1MIN! = 2097.876
```

Valeur (à 7 chiffres au maximum)
Caractère d'attribution
Variable REAL de précision double

Attribution d'une valeur entre variables

```
1 XSOLL = X1MIN!
```

Valeur (variable REAL de précision double)
Caractère d'attribution
Variable REAL de précision simple

La variable à laquelle une valeur doit être attribuée doit se trouver à gauche du caractère d'attribution, la valeur correspondante à sa droite. Cet accord doit surtout être respecté si une valeur de variable doit être attribuée à une autre variable.

NUL

Si aucune valeur n'a été attribuée à une variable, celle-ci a alors la valeur NUL, c.-à-d. que la déclaration <variable>= NUL est vraie. Ici, il est possible de reconnaître que le signe égal peut aussi se trouver dans des comparaisons ou des conditions.

Une variable locale ou globale peut être effacée en lui attribuant la valeur NUL. Une variable permanente ne peut pas être effacée, mais seulement écrasée.

Exemple : Effacement d'une variable

Programme:

```
1 XSOLL = NUL
2 IF XSOLL = NUL THEN
3     PRN#(0,"Variable non occupée.")
4 ENDIF
```



Les variables auxquelles aucune valeur n'est attribuée, ou dont la valeur a été effacée par l'attribution de la valeur NUL, ne peuvent pas être attribuées à d'autres variables en tant que valeur ! Cependant, une requête de la variable sur NUL est possible à tout moment.

3.17.2 Opérations mathématiques

Fonctions simples

Outre l'attribution d'une valeur en tant qu'expression constante (chiffres) ou en tant que variable, il est également possible d'attribuer la valeur d'une expression CPL à une variable. Une expression CPL peut inclure les fonctions avec des constantes et des variables.

Les calculs arithmétiques élémentaires comptent parmi les fonctions les plus simples :

- Addition "+"
- Soustraction "-"
- Multiplication "**"

Notions de base pour la programmation CN

- Division "/"

Ici, "les opérations à points sont prioritaires sur les opérations à trait", c.-à-d. que les multiplications et les divisions sont calculées avant les additions et les soustractions. De plus, il est possible d'utiliser des parenthèses dont l'imbrication à 7 niveaux est possible pour les expressions simples (sans appels de fonctions).

Exemple :

1 I% = 25: XREEL = 10

2 XCONSIGNE = 150/(100-I%)+XREEL XCONSIGNE a la valeur 12.

Il est également possible d'appeler des fonctions mathématiques ayant un effet sur les variables, constantes ou expressions CPL, qui doivent être inscrites entre parenthèses immédiatement après le mot d'ordre concerné. La fonction se réfère toujours à la représentation interne des chiffres de la valeur entrée. Il est possible de la contrôler lors de l'exécution du programme sous "contrôle du programme". Si des expressions sont imbriquées, surtout si elles contiennent des appels de fonctions, il convient de respecter la profondeur d'imbrication maximale possible qui dépend de l'espace mémoire requis pour les expressions entre parenthèses en cours l'exécution.

ABS

Représente la valeur absolue du mot d'entrée, c.-à-d. que les valeurs négatives sont positives tandis que les valeurs positives restent positives.

Exemple :

1 I% = -125

2 XVALEUR = 2*SQRT(ABS(100+I%)) XVALEUR a la valeur 10.

EXP

Calcule la fonction exponentielle d'une base programmée quelconque ou d'une base par défaut e ($e=2.718281828...$) si aucune <base> (paramètre optionnel) n'est indiquée.

Syntaxe :

EXP(<exposant> [, <base>])

Tab. 3-25: Syntaxe EXP

Exemple :

Programme:

1 A! = EXP(2) : REM calcule e^2 et attribue le résultat A!
 2 B! = EXP(3,2) : REM calcule 2^3 et attribue le résultat B!

INT

INT convertit la valeur entrée (REAL) en nombre entier (type CPL INT) en tronquant les chiffres après la virgule (arrondissement). La valeur entrée peut être une constante ou une variable.

Exemple :

1 XVALEUR% = INT(10.9) XVALEUR a la valeur 10.

RANDOM

La fonction génère un nombre entier positif aléatoire. Par défaut, sa valeur est comprise dans une plage entre 0 et 32 767.

Les paramètres optionnels <ValeurMin> et <ValeurMax> permettent de pré-définir la plage de valeurs du chiffre aléatoire.

Les points suivants s'appliquent impérativement ici :

<ValeurMax> - <ValeurMin> <= 32 767

Notions de base pour la programmation CN

Syntaxe :

RANDOM ([[<ValeurMin>] [, <ValeurMax>]])	
<ValeurMin>	Optionnel : Valeur minimale du chiffre aléatoire (chiffre positif)
<ValeurMax>	Optionnel : Valeur maximale du chiffre aléatoire (chiffre positif)
ERRNO	Variable CPL Si la variable CPL ERRNO est programmée à un emplacement quelconque au sein de la liste de paramètres, l'ordre ne génère aucune erreur d'exécution interne. L'erreur est rendue par une valeur de retour correspondante de la variable. En cas d'erreur, le chiffre aléatoire 0 est généré. Les valeurs de retour suivantes sont possibles : 0 : Accès OK -1 : Erreur de paramètre Si la variable ERRNO n'est pas indiquée, une erreur d'exécution est générée en cas d'erreur de paramètre.

Tab. 3-26: Syntaxe RANDOM

Exemple :

```

1 REM chiffre aléatoire 0 ou 1
2 I%=RANDOM(0,1)
3 REM chiffre aléatoire entre 10 et 100
4 I%=RANDOM(10,100)

```

ROUND

ROUND convertit la valeur entrée en un nombre entier (type CPL INT) en l'arrondissant à l'entier supérieur ou inférieur. La valeur entrée peut être une expression REAL.

Exemple :

```

1      XVALEUR%      = XVALEUR a la valeur 11.
ROUND(10.9)

2      XVALEUR%      = XVALEUR a la valeur 6.
ROUND(5,5)

3      XVALEUR%      = XVALEUR a la valeur 5.
ROUND(5.49)

```

SQRT

Calcule la racine carrée de la valeur entrée. Pour l'instant, la valeur entrée ne doit pas être négative, étant donné que ceci n'est pas défini.

Exemple :

```

1 I% = 44

2      XCONSIGNE      = XCONSIGNE a la valeur 48.
4*SQRT(100+I%)

```

SIN, COS, TAN, ASIN, ACOS, ATAN

Dans le cas des fonctions trigonométriques qui traitent les angles en degrés, il est judicieux de caractériser les angles en tant que variables REAL à précision double.

La grandeur d'entrée des fonctions trigonométriques SIN, COS, TAN est un angle en degré et le résultat fourni des valeurs numériques. Par conséquent,

Notions de base pour la programmation CN

la grandeur d'entrée des fonctions inverses ASIN, ACOS, ATAN est une valeur numérique et le résultat un angle en degrés.

Il est judicieux de convenir des angles comme variables REAL double précision.

Les fonctions trigonométriques suivantes peuvent être utilisées :

•	SIN	Fonction sinus
•	COS	Fonction cosinus
•	TAN	Fonction tangente
•	ASIN	Fonction arc sinus
•	ACOS	Fonction arc cosinus
•	ATAN	Fonction arc tangente

Exemple :

```
1 ANGLE! = 30
2 RESULT! = SIN(ANGLE!)  RESULT! a la valeur 0,5.
3 ANGLE! = ASIN(RE-  ANGLE! a la valeur 30.
  SULT!)
4 ANGLE! = ATAN( 1/1 )  ANGLE! a la valeur 45 degrés.
5 ANGLE! = ATAN( 1/0 )  Non défini en raison de la division par zéro.
6 ANGLE! = ATAN( 0/-1 )  ANGLE! a la valeur 0 degrés.
7 ANGLE! = ATAN(-1/-1 )  ANGLE! a la valeur 45 degrés.
```

ATAN2

Constitue l'arc tangente dans le bon quadrant, c.-à-d. que le résultat est compris dans une plage de ± 180 degrés. Pour cela, ATAN2 doit être appelé avec deux arguments. L'arc tangente simple (ATAN) ne propose pas cette possibilité, son résultat se situe toujours à une plage de ± 90 degrés.

Il existe le rapport suivant entre ATAN et ATAN2 :

$\alpha_1 = \text{atan}(\text{cathète opposée}/\text{cathète adjacente}) = \text{atan}(y/x)$

avec $-90 < \alpha_1 < 90$

$\alpha_2 = \text{atan2}(\text{cathète opposée}, \text{cathète adjacente}) = \text{atan2}(y, x)$ avec $-180 < \alpha_2 < 180$

Exemple :

```
ANGLE = ATAN2(1,1)  L'ANGLE a la valeur 45 degrés.
ANGLE = ATAN2(1, 0)  L'ANGLE a la valeur 90 degrés.
ANGLE = ATAN2(0, -1)  L'ANGLE a la valeur 180 degrés.
ANGLE = ATAN2(-1, -1)  L'ANGLE a la valeur -135 degrés.
```

3.17.3 Liaisons logiques

Généralités

Il est possible d'exécuter des liaisons logiques binaires avec des variables logiques ou décimales avec des variables en nombre entier. Elles peuvent également être représentées, comme illustré dans la figure ci-dessous, avec les symboles d'opération habituels, à savoir le caractère "." et le caractère "+" (**non en CPL**). Ici aussi, "les opérations à point sont prioritaires sur les opérations à trait", la liaison AND prend donc effet avant la liaison OR. Une imbrication de parenthèses est possible sur un maximum de 7 niveaux.

Notions de base pour la programmation CN

NOT, AND, OR, XOR

CPL met à disposition quatre fonctions de liaison :

- Fonction NON **NOT**
- Fonction ET **AND**
- Fonction OU **OR**
- Fonction OU exclusif **XOR**

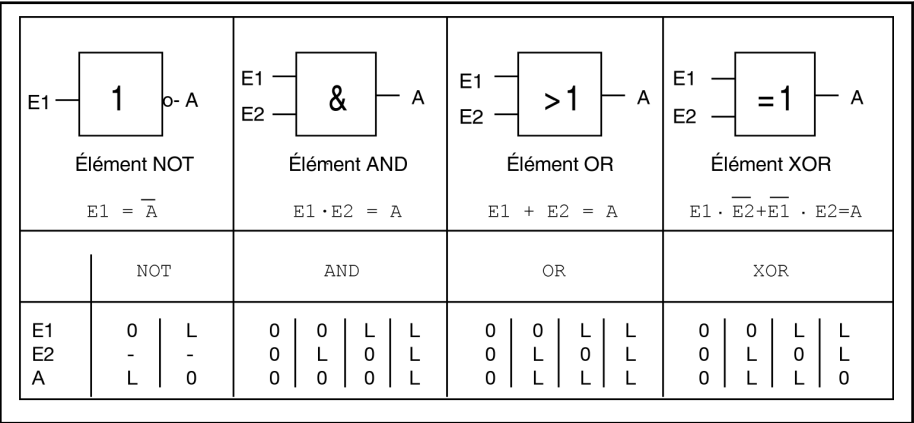


Fig. 3-1: Liaisons logiques

Les liaisons logiques peuvent être utilisées pour masquer des bits.

Exemple : Le bit 0 est-il défini dans @20 ?

```
...
20 IF @20 AND 1 <> 0 THEN GOTO . DÉFINI
30 ELSE GOTO . NON DÉFINI ENDIF
...
```

3.17.4 Conversion entre systèmes de nombres

BCD

Convertir une valeur binaire au format BCD.

Syntaxe :

<valeur BCD>=BCD (<valeur binaire>)

Tab. 3-27: Syntaxe BCD

Exemple : 1 BCD_VALEUR = BCD(49) BCD_VALEUR a la valeur 73.

BIN

Convertir des nombres codés BCD en valeur binaire.

Syntaxe :

<valeur binaire>=BIN (<valeur BCD>)

Tab. 3-28: Syntaxe BIN

Exemple : 1 BIN_VALEUR = BCD(49) BIN_VALEUR a la valeur 31.

3.17.5 Opérations de comparaison" =, >=, >, <>, <=, <"

Les opérateurs de comparaison suivants sont autorisés :

- = Égal à
- >= Supérieur ou égal à
- > Supérieur à

<>	Différent de
<=	Inférieur ou égal à
<	Inférieur à

Les opérations de comparaison sont utilisées pour décrire une **condition** (par ex. avec les instructions REPEAT-UNTIL, WHILE-DO-END, IF-THEN-ELSE-ENDIF) applicable à la **relation** ("remplie" ou "non remplie").

3.18 Autres éléments de base CPL

3.18.1 Constantes

Généralités

Il s'agit de valeurs numériques définies pour le déroulement du programme, qui doivent rester inchangées (constantes), mais que l'on peut écrire directement en nombres dans les instructions.

Constante en nombre entier (INT)

Les nombres entiers sont inscrits sans signe décimal.

Exemple : NOMBRE% = 4 Constante en nombre entier (de type CPL INT)

Constante à virgule flottante (DOUBLE)

Les nombres réels (nombres décimaux ou fractions) sont caractérisés par un signe décimal (virgule flottante). Si un chiffre à virgule flottante est inférieur à 1, le zéro doit impérativement être programmé devant la virgule. Exemple : 0.134.

Exemple : PI = 3.141593 Constante à virgule flottante (de type CPL REAL)

Les constantes à virgule flottantes ont toujours une précision double, c.-à-d. qu'elles sont représentées exactement avec 15 chiffres.

Constante de chaîne de caractères

Une constante de chaîne de caractères (constante de chaîne) est délimitée par des guillemets (").

Exemple : EXEMPLE\$ = "Ceci est une chaîne de caractères" Constante STRING

3.18.2 Caractères clés

CPL utilise les caractères clés suivants :

#	!	?	,	"	[	]	<	-	/	&
@	%	\$	:	(	)	=	>	+	*	

La virgule sert normalement de caractère de séparation. Cependant, elle est utilisée comme signe de ponctuation au sein des chaînes de caractères uniquement. Le point sert de signe décimal dans les chiffres décimaux et d'identificateur label pour les destinations de saut. Au sein des chaînes de caractères, le point est interprété comme un signe de ponctuation.

Notions de base pour la programmation CN

3.19 Ordres de synchronisation du bloc CN

3.19.1 Vue d'ensemble

Les ordres de synchronisation du bloc CN permettent :

- de synchroniser des programmes CN avec certains événements définis.
- de synchroniser la préparation des blocs avec l'exécution du programme.
- de limiter la prévisualisation des blocs (LookAhead) à un nombre de blocs défini.
- de synchroniser les programmes CN les uns avec les autres dans différents canaux.
- Les fonctions de synchronisation prennent effet soit au moment de la préparation des blocs, soit au moment actif de l'exécution du bloc.

Fonctions de synchronisation de la préparation des blocs :

• WAIT (sans paramètre) :	Arrête la préparation des blocs jusqu'à ce que tous les blocs précédents aient été traités.
• Fonction CPL WAIT(<temps d'attente>) :	La préparation des blocs est arrêtée pendant le temps indiqué.
• Fonction CPL WAIT(BITIF(...)) :	La préparation des blocs attend un signal défini au sein de l'interface bit API CN.
• BlkNmb :	Limite la prévisualisation des blocs au nombre de blocs programmé.

Fonctions de synchronisation au moment actif :

• WAITA / WAITO :	Attente d'un état défini sur l'interface bit API CN.
• WPV / WPVE :	Attente de la valeur d'une variable CPL permanente.
• SPV / SPVE :	Écriture de variables CPL permanentes.
• ASTOPA / ASTOPO :	Arrêt du mouvement jusqu'à ce qu'une position définie de l'axe soit atteinte.
• BSTOPA / BSTOPO :	Arrêt du mouvement jusqu'à ce que les positions définies soient atteintes dans le système de coordonnées de base de la pièce à usiner (BCS).
• WSTOPA / WSTOPO :	Arrêt du mouvement jusqu'à ce que les positions définies soient atteintes dans le système actuel de coordonnées de la pièce à usiner (WCS).
• OFFSTOPA / OFFSTOPO :	Suppression des conditions d'arrêt sur le canal de commande.

3.19.2 Fonctions de synchronisation de la préparation des blocs

WAIT (sans paramètres)

Description : La fonction "WAIT" **arrête** la **préparation des blocs** jusqu'à ce que tous les blocs programmés avant le bloc WAIT aient été traités entièrement. Elle est

Notions de base pour la programmation CN

impérativement nécessaire si le programme doit accéder par la suite à des données actuelles relatives à la machine ou au processus.

La préparation des blocs dans laquelle chaque ligne du programme est analysée et interprétée a toujours lieu avant l'exécution sur la machine (considération temporelle). Le laps de temps entre la préparation et l'exécution peut varier en fonction de plusieurs paramètres (avance, course de déplacement, Look-Ahead, etc.).

Si le programme doit réagir à un état réel de la machine ou du processus (par ex. position réelle actuelle, signal sur l'interface bit, etc.), la fonction WAIT doit garantir que ce laps de temps est exactement égal à "0" au moment de l'évaluation.

Exemple : WAIT (sans paramètres)

```
N10 X0
N100 (MSG, encore en marche)
N20 X150
30 WAIT                                Traitement du bloc arrêté
40 XPOS = MCS(1)-150
50 IF XPOS < 0.0001 THEN                "Position atteinte" est signalé sur X=150.
(MSG, position atteinte)
70 ENDIF
```



Il est possible de programmer la fonction "WAIT" (sans) paramètres aussi bien dans des blocs CN standards (DIN) que dans des blocs CPL. Aucun caractère ":" ne doit être programmé dans un bloc CPL avec une instruction WAIT. Les instructions CPL suivantes doivent être inscrites dans un nouveau bloc CPL.

Fonction CPL : WAIT(<temps d'attente>)

Description : Arrête la préparation des blocs jusqu'à ce que le temps d'attente programmé soit écoulé.

Syntaxe :

WAIT(, <temps d'attente>[, <var. résultat>])	
avec	
<temps d'attente>	Temps d'attente en millisecondes, sans chiffres après la virgule. Le temps d'attente peut également être programmé en tant qu'expression arithmétique en nombre entier.
<var. résultat>	Variable en nombre entier (optionnelle) Une fois le temps d'attente expiré, 1 est affecté à <var. résultat>.

Tab. 3-29: Syntaxe WAIT

Exemple : WAIT avec temps d'attente

```
10 WAIT(, Le traitement du bloc est arrêté pendant 1000 ms. Ensuite, la va-
1000,E%) leur en nombre entier "1" est attribuée à la variable E%.
```

```
10
WAIT(TEMPS% Le contenu de la variable en nombre entier TEMPS% détermine
%) combien de temps le traitement du bloc est arrêté. Aucune valeur
n'est renvoyée.
```

Notions de base pour la programmation CN

Particularités et restrictions : La fonction est négligée dans l’avance de bloc.

Fonction CPL : WAIT(BITIF(...))

Description : La préparation des blocs est arrêtée jusqu’à ce qu’un état défini se soit produit sur l’interface bit API CN.

En option, il est également possible de programmer un laps de temps. La préparation des blocs est alors arrêtée jusqu’à ce que l’état se produise sur l’interface bit ou jusqu’à ce que le laps de temps programmé ait expiré.

Syntaxe :	WAIT(<condition BITIF>[, [<délai d’expiration>] [, <var. résultat>]]) avec	
<condition BITIF>	Spécifie la condition devant être contrôlée sur l’interface bit API CN. Pour cela, la syntaxe suivante doit être respectée : [NOT () BITF (<paramètre>)] [= <état>]	
	<paramètre> Paramètres de transfert de la fonction BITIF (pour obtenir une description, voir fonction BITIF au chap. 7 "Ordres CPL" à la page 449).	
	<état> Expression booléenne à laquelle le résultat de la fonction BITIF est comparé. Si <état> n’est pas programmé, une comparaison avec TRUE a lieu.	
	Si la condition est remplie, le traitement du bloc reprend.	
<délai d’expiration>	Temps d’attente maximal en millisecondes, sans chiffres après la virgule. La préparation des blocs attend jusqu’à ce que la <condition BITIF> ait été remplie, mais au maximum jusqu’à ce que le <délai d’expiration> se soit écoulé.	
<var. résultat>	Variable en nombre entier (optionnelle) : Fournit en retour le moment où le résultat de l’attente est intervenu : 0 : La condition d’attente était déjà remplie au moment de l’appel. 1 : Le temps d’attente s’est écoulé sans que la condition d’attente n’ait été remplie. 2 : La condition d’attente a été remplie avant l’expiration du temps d’attente.	

Tab. 3-30: Syntaxe WAIT

Exemple : WAIT(BITIF(...))

```
10 WAIT (BITIF (1, 1, 1) = TRUE)
```

Attendre jusqu’à ce que le 2e signal d’entrée relatif au 1er axe soit défini.

```
11 WAIT (BITIF (2, 0, 2) = (E1? OR E2?))
```

Attendre jusqu’à ce que le 3e signal d’entrée relatif au 2e canal ait la valeur de l’expression logique (E1? OR E2?).

Notions de base pour la programmation CN

```
12 WAIT (NOT BITIF (3, 2, 1) , , C%)
```

Attendre jusqu'à ce que le 4e signal d'entrée relatif à la 1re broche ait la valeur FALSE.

```
20 IF C%=0 THEN
```

```
21 PRN# (0, "COND. DÉJÀ OK")
```

```
22 ENDIF
```

La variable C% fournit la valeur "0" quand la condition est déjà remplie lors de l'appel de la fonction WAIT.

```
30 WAIT (BITIF (4, 4, 1) =E7?, 250, RESULT%)
```

Attendre jusqu'à ce que le 5e signal de sortie relatif au 1er axe prenne la valeur de la variable E7? ou jusqu'à ce que le temps d'attente de 250 ms ait expiré.

```
40 IF RESULT%=0 THEN
```

```
41 PRN# (0, "NON ATTENDU")
```

```
42 ENDIF
```

La variable RESULT% fournit la valeur "0" quand la condition est déjà remplie lors de l'appel de la fonction WAIT.

```
50 IF RESULT%=1 THEN
```

```
51 PRN# (0, "ATTENDU >250 ms")
```

```
52 ENDIF
```

La variable RESULT% fournit la valeur "1" quand la condition n'a pas été remplie au cours de l'attente.

Particularités et restrictions : La fonction est négligée dans l'avance de bloc.

BlkNmb (BNB)

Description : La fonction CN standard "BlkNmb" permet de limiter le nombre maximal de blocs traités lors de la préparation des blocs.

La fonction "BlkNmb" permet, par exemple, de commander la poursuite du traitement des résultats de mesure déterminés lors de l'exécution dans le programme pièce suivant.

Syntaxe :

BlkNmb<nbre>	Limitation ACTIVÉE
BlkNmb(0)	Le nombre de blocs maximal autorisé dépend du paramètre /NCO/ LookAh/Ch/NofBlkPrep (MP 7060 00110).
Format abrégé : BNB . . . avec	
<nbre>	Nombre de blocs maximal voulu Nombre entier Champs de saisie : supérieur/égal à 0. 0 : Le nombre de blocs maximal autorisé dépend de MP 7060 00110.

Tab. 3-31: Syntaxe BlkNmb (BNB)

Particularités et restrictions :

- Si plus de <nbre> blocs ont déjà été préparés au moment de l'appel de la fonction, alors la préparation des blocs s'arrête jusqu'à ce que le nombre de blocs préparés soit inférieur à <nbre>.
- Si <nbre> définit un nombre de blocs supérieur à ce qui est prévu dans le cadre de la configuration de la commande, alors le nombre de blocs effectif sera automatiquement réduit au réglage de MP 7060 00110.

Notions de base pour la programmation CN

3.19.3 Fonctions de synchronisation lors de la préparation des blocs

Généralités

Un programme peut être exécuté sur chaque canal de la commande IndraMotion MTX. Si les séquences d'usinage sont subdivisées en différents programmes séparés et si ces programmes sont exécutés sur différents canaux, il est possible de piloter le déroulement de l'usinage de tout programme individuel par les fonctions de synchronisation dépendantes du déroulement de l'exécution.

Les fonctions de synchronisation lors de l'exécution du bloc avec les événements bit (REV, SEV, WEV et WREV) sont programmées avec des **listes de paramètres DIN**.



Les événements bit sont des niveaux (0 ou 1) valides dans tout le système, pouvant être modifiés ou utilisés avec les fonctions CN REV (ResetEvent), SEV (SetEvent), WEV (WaitForEvent) et WREV (WaitForAndResetEvent).

Toutes les autres fonctions de synchronisation lors de l'exécution du bloc ont des **listes de paramètres CPL** (similaires à des sous-programmes). Ces dernières comprennent les listes programmées d'expressions CPL insérées entre crochets "[" et "]".

Les points suivants s'appliquent de manière générale :

- Des événements bit valides dans tout le système sont utilisés dans REV, SEV, WEV et WREV. Le programmeur doit s'assurer qu'aucune interaction ne se produise.
- Les variables CPL permanentes utilisées dans les fonctions "WPV", "WPVE", "SPV", "SPVE" sont également valides dans tout le système. Par conséquent, le programmeur doit s'assurer ici aussi qu'elles sont utilisées correctement afin d'éviter toute interaction involontaire.
- Dans les fonctions CN proposées, seuls les types de variables CPL permanentes simples suivants peuvent être utilisés :
 - INT
 - BOOL
 - REAL
 - DOUBLE

Pour les tableaux, seuls les éléments individuels peuvent être adressés !



Les fonctions WEV, WREV, WAITA, WAITO, WPV et WPVE entraînent implicitement un front descendant à la fin du bloc. Des points de synchronisation mal définis peuvent entraîner un endommagement de la machine.

Il est donc recommandé de tester le déroulement du programme avant l'usinage concret afin de détecter d'éventuels problèmes de synchronisation au cours de son déroulement.

Attente d'un événement bit : WEV

Description : Lors de l'exécution, la fonction "WEV" attend (ou empêche la commutation au bloc suivant) jusqu'à ce qu'un événement bit défini se produise.

Notions de base pour la programmation CN

Syntaxe :

WEV (<n° événement> {, n° événement...})	
avec :	
<n° événement>	Numéro d'un bit d'événement. Valeurs valides : 1 à 96. Il est possible de programmer de nombreux <n° événements> dans une fonction WEV.

Tab. 3-32: Syntaxe WEV

Les événements au sein d'une fonction WEV sont reliés par un opérateur logique disjonctif (ou). Plusieurs fonctions WEV peuvent être programmées dans un bloc. Toutes les fonctions WEV programmées séparément sont reliées par un opérateur logique conjonctif (et). La programmation multiple d'un événement ne génère aucune erreur.

Si l'événement est déjà présent, alors la fonction WEV est tout simplement ignorée. Si l'utilisateur veut absolument attendre (réagir au front), il devra auparavant supprimer l'événement avec fonction REV. Ceci peut se produire dans le même bloc.

Exemple :

N10 X10 WEV(10,12)	Le déplacement s'effectue d'abord vers X10, puis le programme attend jusqu'à ce que l'événement 10 ou 12 se produise. Si l'événement 10 ou 12 se produit dès le déplacement vers X10 ou plus tôt, aucune attente n'aura lieu à la fin du bloc. (Une rampe vers le bas sera tout de même créée.)
N10 WEV(10,12) WEV(16, 18) WEV(20)	Le bloc ne sera terminé que lorsque les événements (10 ou 12) et (16 ou 18) et 20 seront présents.

WEV / WREV / SEV / REV

Les fonctions "WEV", "SEV" et "REV" permettent, par exemple, de synchroniser deux canaux en cours d'exécution à l'aide d'événements bit.

Si plusieurs fonctions sont programmées dans un bloc, elles prennent effet dans l'ordre suivant :

1. REV
2. SEV
3. WEV / WREV

Le bloc N10 REV(10) WEV(10) permet ainsi d'attendre un front de l'événement 10.

La séquence suivante est exécutée sur le canal 1 :

Canal 1 Programme:

```
;Canal 1
N10 REV(20, 21)      Supprimer les événements 20 et 21
N20 G1 F1000 X100 Y100 Approcher position 1
N30 WEV(20)
;attendre que l'événement 20 soit présent, ici synchronisation avec le canal 2 N30.. lors de
la durée d'exécution
N30 X200 Y200        Approcher position 2
N40 SEV(21)          déclenche l'événement 21, le canal 2 peut continuer maintenant
```

La séquence suivante est exécutée sur le canal 2 :

Canal 2 Programme:

```
;Canal 2
N20 G1 F1000 Z100 W100 Approcher position 1
N30 SEV(20)            déclenche l'événement 21, le canal 1 peut continuer maintenant
N30 Z200 W200          Approcher position 2
N40 WEV(21)
```

Notions de base pour la programmation CN

```
;attendre que le canal 1 déclenche l'événement 21, ici synchronisation avec le canal 1 N40..  
lors de la durée d'exécution
```

Attente et suppression d'un événement bit : WREV

Description : Lors de l'exécution, la fonction "WREV" attend (ou empêche la commutation au bloc suivant) jusqu'à ce qu'un événement bit défini se produise. Si la condition est remplie, tous les événements bit qui ont été attendus sont supprimés.

Syntaxe :	WREV (<n° événement> {, n° événement...})	
	avec :	
	<n° événement>	Numéro d'un bit d'événement. Valeurs valides : 1 à 96. Il est possible de programmer de nombreux <n° événement> dans une fonction WREV.

Tab. 3-33: Syntaxe WREV

Concernant la condition d'attente, la fonction WREV se comporte comme la fonction WEV (voir [chap. "Attente d'un événement bit : WEV" à la page 100](#)). Tous les événements bit qui ont été programmés dans la condition d'attente sont supprimés une fois la condition remplie. La condition d'attente peut être combinée à partir des fonctions WEV et WREV.



Entre la satisfaction de la condition et la suppression des événements bit, il s'écoule un certain temps pendant lequel d'autres canaux ou blocs sont exécutés.

Exemple : N10 WREV(11) est programmé sur le canal 1. Les blocs N20 SEV(11) puis N20 WEV(11) arrivent sur le canal 2. Une fois la fonction SEV exécutée, la condition d'attente de la fonction WREV est remplie et l'exécution du bloc se poursuit sur le canal 1. Mais il s'écoule encore un peu de temps jusqu'à ce que l'événement bit 11 soit supprimé. Si la fonction WEV a auparavant été exécutée sur le canal 2, alors ce dernier n'attend pas (car l'événement bit 11 est encore actif). Toutefois, si la suppression a lieu en premier, le canal 2 attend.

En de tels emplacements, il est recommandé de programmer une synchronisation via deux événements bit différents.

Exemple :

N10 X10 WREV(10,12)	Le déplacement s'effectue d'abord vers X10, puis le programme attend jusqu'à ce que l'événement 10 ou 12 se produise. Si l'événement 10 ou 12 se produit dès le déplacement vers X10 ou plus tôt, aucune attente n'aura lieu à la fin du bloc. (Une rampe vers le bas sera tout de même créée.) Les événements bit 10 et 12 seront ensuite supprimés.
N10 WREV(10,12) WEV(16, 18) WREV(20)	Le bloc ne sera terminé que lorsque les événements (10 ou 12) et (16 ou 18) et 20 seront présents. Les événements bit 10, 12 et 20 seront ensuite supprimés.

Déclencher un événement bit : SEV

Description : La fonction "SEV" permet de déclencher un événement bit (fixé) en cours d'exécution. Le niveau de l'événement programmé est ainsi défini sur 1 (peu importe s'il correspondait auparavant à 0 ou 1). Plusieurs fonctions SEV peuvent être programmées dans un bloc.

Notions de base pour la programmation CN

Syntaxe :	
SEV (<n° événement> {, <n° événement> ...})	
avec :	
<n° événement>	Numéro d'un bit d'événement Valeurs valides : 1 à 96 Il est possible de programmer de nombreux <n° événement> au sein d'une fonction SEV.

Tab. 3-34: Syntaxe SEV

Exemple :	N10 SEV(13)	L'événement 13 est défini.
	N10 SEV(15, 16)	Les événements 15 et 16 sont définis.
	N10 SEV(15) SEV(16)	Les événements 15 et 16 sont définis.

Suppression d'un événement bit : REV

Description : La fonction "REV" déclenche un événement bit en cours d'exécution. Plusieurs fonctions REV peuvent être programmées dans un bloc.

Syntaxe :	
REV (<n° événement> {, <n° événement> ...})	
avec :	
<n° événement>	Numéro d'un bit d'événement Valeurs valides : 1 à 96 Il est possible de programmer de nombreux <n° événement> au sein d'une fonction REV. Cas particuliers : REV(ALL) ou REV() suppriment tous les bits d'événement.

Tab. 3-35: Syntaxe REV

Exemple :	N10 REV(13)	L'événement 13 est supprimé.
	N10 REV(15, 16)	Les événements 15 et 16 sont supprimés.
	N10 REV(15) REV(16)	Les événements 15 et 16 sont supprimés.
	N10 REV(ALL)	Tous les événements sont supprimés.

Attente des états sur l'interface bit API CN : WAITA / WAITO

Description : La fonction "WAITA / WAITO" attend le moment d'exécution jusqu'à ce qu'un ou plusieurs signaux d'interface sur un maximum de 16 signaux aient accepté une valeur prédéfinie respectivement.

Selon la liaison, une programmation est possible avec plusieurs signaux d'interface :

- **WAITA :**
"Liaison ET" des différents signaux
Attendre jusqu'à ce que **tous** les signaux d'interface aient accepté la valeur prédéfinie.
- **WAITO :**
"Liaison OU" des différents signaux
Attendre jusqu'à ce qu'**au moins un** signal d'interface ait reçu la valeur prédéfinie.

Notions de base pour la programmation CN

Syntaxe :	
Attendre tous les signaux indiqués : WAITA[BITIF(<paramètre>){=<état>}, BITIF(<paramètre>){=<état>},...,<délai d'expiration>}]	
Attendre un des signaux indiqués : WAITO[BITIF(<paramètre>){=<état>}, BITIF(<paramètre>){=<état>},...,<délai d'expiration>}]	
avec :	
<BITIF>	La fonction BITIF interroge l'interface API CN. En option, 2 à 16 signaux d'interface peuvent être interrogés simultanément.
<paramètre>	Paramètre de transfert de la fonction BITIF (pour obtenir une description, voir fonction BITIF au chap. 7 "Ordres CPL" à la page 449)
<état>	Expression booléenne à laquelle le résultat de la fonction BITIF est comparé. Si <état> n'est pas programmé, une comparaison avec TRUE a lieu. Si la condition est remplie, le traitement du bloc reprend.
<délai d'expiration>	Optionnel. Temps en ms. Valeur par défaut = 0. Si le <délai d'expiration> est écoulé avant que la condition correspondante ne soit remplie, un avertissement est généré et l'attente se poursuit. Si aucun délai d'expiration n'a été programmé ou s'il est égal à 0, aucun avertissement n'est généré.

Tab. 3-36: Syntaxe BlkNmb (BNB)

Exemple : Le programme 1 sur le canal 1 traite la face avant d'une pièce à tourner. Programme 2 sur le canal 2 doit fraiser une rainure sur cette face et doit **attendre** que le programme 1 libère la pièce pour le programme 2. La libération de la pièce pour le programme 2 s'effectue par l'**initialisation** de certains signaux d'interface. Quand les signaux d'interface ont atteint l'état voulu, alors le canal 1 transmet la validation au canal 2. Pendant que le programme 2 est traité, le programme 1 attend le programme 2 pour poursuivre son traitement.

N10 WAITO[BITIF(10,1,1)=FALSE, BITIF(11,1,2)]	Attend activement jusqu'à ce que BITIF(10,1,1) ait pris la valeur 0 ou que BITIF(11,1,2) ait pris la valeur 1.
N10 WAITA[BITIF(10,1,1)=FALSE, BITIF(11,1,2)]	Attend activement jusqu'à ce que BITIF(10,1,1) ait pris la valeur 0 et que BITIF(11,1,2) ait pris la valeur 1.

Les points suivants s'appliquent aux fonctions WAITA et WAITO :
Si les fonctions WAITA et WAITO sont programmées dans un bloc CN, l'exécution du bloc sera alors suspendue jusqu'à ce que ces deux conditions soient remplies, la condition WAITO étant évaluée en priorité.



Les fonctions "WAITA", "WAITO", "WPV" et "WPVE" entraînent implicitement un front descendant à la fin du bloc.

Des points de synchronisation mal définis peuvent entraîner un endommagement de la machine.

Il est donc recommandé de tester le déroulement du programme avant l'usinage concret afin de détecter d'éventuels problèmes de synchronisation au cours de son déroulement.

Particularités et restrictions : La fonction est négligée dans l'avance de bloc.

Attendre la valeur d'une variable CPL permanente : WPV / WPVE

Description : En cours d'exécution, la fonction "WPV / WPVE" attend jusqu'à ce qu'une variable CPL permanente accepte une valeur de comparaison définie.

La valeur de comparaison peut être déterminée à différents moments :

- **WPV :**

La valeur de comparaison est une expression CPL calculée lors de l'exécution et ce, juste au moment où le bloc CN est activé. Toute modification apportée à la valeur de comparaison après que le bloc CN a été activé n'a plus d'effet. La valeur de comparaison est comparée avec la valeur de la variable permanente.

L'évaluation en cours d'exécution n'admet qu'une expression CPL simple.

- **WPVE :**

La valeur de comparaison est une expression CPL calculée lors de la préparation, mais elle n'est comparée avec la valeur de la variable permanente qu'au moment de l'exécution.

Syntaxe :

La valeur de comparaison n'est déterminée qu'au moment de l'exécution :	
WPV [<variable CPL perm.><opérateur de comparaison><expression CPL simple>{,<délai d'expiration>}]	
La valeur de comparaison n'est déterminée qu'au moment de la préparation :	
WPVE [<variable CPL perm.><opérateur de comparaison><expression CPL>{,<délai d'expiration>}]	
avec :	
<variable CPL perm.>	Variable permanente identifiée par le caractère "@" et suivie du nom de variable. Avec la fonction WPV, l'index des éléments d'un tableau ne peut être qu'une constante ou une variable permanente en nombre entier (de type CPL INT). Avec la fonction WPVE, l'index peut être une constante ou une variable CPL en nombre entier quelconque.

Notions de base pour la programmation CN

<opérateur de comparaison>	Les opérateurs de comparaison suivants sont possibles :	
	=	La variable CPL permanente est égale à la valeur de l'expression CPL. Judicieux seulement pour des valeurs entières ou booléennes.
	< >	La variable CPL permanente est différente de la valeur de l'expression CPL. Judicieux seulement pour des valeurs entières ou booléennes.
	<	La variable CPL permanente est inférieure à la valeur de l'expression CPL.
	≤	La variable CPL permanente est inférieure ou égale à la valeur de l'expression CPL.
	>	La variable CPL permanente est supérieure à la valeur de l'expression CPL.
	≥	La variante CPL permanente est supérieure ou égale à la valeur de l'expression CPL.
<expression CPL simple>	Afin de ne pas entraver la génération du mouvement, seules les expressions CPL simples peuvent être évaluées lors de l'exécution. Une expression CPL simple est une expression mathématique composée de données système, de variables CPL permanentes, de constantes et d'opérations mathématiques possibles en CPL. <i>Les types de données autorisés sont les suivants :</i> • INT • BOOL • REAL • DOUBLE	
<expression CPL>	Expression mathématique quelconque en langage de programmation CPL.	
<délai d'expiration>	Optionnel. Temps en ms. Valeur par défaut = 0. Si le <délai d'expiration> est écoulé avant que la condition correspondante ne soit remplie, un avertissement est généré et l'attente se poursuit. Si aucun délai d'expiration n'a été programmé ou s'il est égal à 0, aucun avertissement n'est généré.	

Tab. 3-37: Syntaxe WPV / WPVE

Exemple :

N10 WPV[@9=10]

Au moment actif, le programme attend jusqu'à ce que la variable @9 accepte la valeur 10.

N10 WPVE[@8=(5*#VAR2%)]

L'expression "5 * #VAR2%" est exploitée au moment de la préparation. La valeur alors déterminée est comparée à la variable permanente @8 au moment de l'exécution. Aucun nouveau bloc CN n'est activé tant que @8 ne correspond pas à la valeur déterminée.

WPV / SPV

Notions de base pour la programmation CN

Les fonctions "WPV" et "SPV" permettent entre autres de synchroniser deux canaux en cours d'exécution à l'aide de variables CPL permanentes.

La séquence suivante est exécutée sur le canal 1 :

Programme:

```

;Canal 1
N20 G1 F1000 X100 Y100      Approcher position 1
N30 WPV(@20=5)              Attendre jusqu'@20=5, ici synchronisation lors de la durée
                             d'exécution avec canal 2
                             N30..
N30 X200 Y200               Approcher position 2
N40 SPV(@20=6)              Poser @20=6, canal 2 doit continuer maintenant

```

La séquence suivante est exécutée sur le canal 2 :

Programme:

```

;Canal 2
N20 G1 F1000 Z100 W100      Approcher position 1
N30 SPV(@20=5)              Poser @20=5, canal 1 doit continuer maintenant
N30 Z200 W200               Approcher position 2
N40 WPV(@20=6)              Attendre jusqu'@20=6, ici synchronisation lors de la durée
                             d'exécution avec canal 1
                             N40..

```

Particularités et restrictions : La fonction est négligée dans l'avance de bloc.

Écriture d'une variable CPL permanente : SPV / SPVE

Description : La fonction "SPV / SPVE" écrit une valeur et l'affecte à une variable CPL permanente en cours d'exécution.

La valeur peut être déterminée à différents moments :

- **SPV :**
La valeur devant être attribuée à la variable permanente n'est déterminée qu'au moment de l'exécution.
L'évaluation lors de l'exécution n'admet qu'une expression CPL simple.
- **SPVE :**
La valeur devant être attribuée à la variable permanente est calculée lors de la préparation (temps d'interprétation CPL), mais elle n'est affectée à la variable CPL permanente qu'au moment de l'exécution.

Syntaxe :

La valeur à attribuer n'est déterminée qu'au moment de l'exécution :	
SPV [<variable CPL perm.> =<expression CPL simple>]	
La valeur à attribuer est déterminée au moment de la préparation :	
SPVE [<variable CPL perm.> =<expression CPL>]	
avec :	
<variable CPL perm.>	Variable permanente identifiée par le caractère "@" et suivie du nom de variable.

Notions de base pour la programmation CN

<expression CPL simple>	Afin de ne pas entraver la génération du mouvement, seules les expressions CPL simples peuvent être évaluées lors de l'exécution. Une expression CPL simple est une expression mathématique composée de données système, de variables CPL permanentes, de constantes et d'opérations mathématiques possibles en CPL. <i>Les types de données autorisés sont les suivants :</i> • INT • BOOL • REAL • DOUBLE
<expression CPL>	Expression mathématique quelconque en langage de programmation CPL.

Tab. 3-38: Syntaxe SPV / SPVE

Exemple :

N10 SPV[@6=1]	La variable permanente "@6" se voit affecter la valeur 1 en cours d'exécution.
N10 SPV[@5=(7*(@VARPERM1% +5))]	La valeur de l'expression (7*(@VARPERM1% + 5)) est calculée en cours d'exécution, puis attribuée à @5.
N10 SPVE[@5=(7*#VAR1%)]	La valeur de l'expression (7 * #VAR1%) est calculée en cours de préparation, puis attribuée à @5 en cours d'exécution.

Arrêt du mouvement jusqu'à ce qu'une position d'axe soit atteinte : ASTOPA / ASTOPO

Description : La fonction "ASTOPA / ASTOPO" permet de synchroniser des mouvements entre plusieurs canaux. En fonction de la **position** d'un ou plusieurs axes du **système de coordonnées de l'axe (ACS)** dans un canal, le mouvement synchrone dans un autre canal peut être arrêté, puis repris.

Restrictions :

- Les axes utilisés pour la synchronisation doivent appartenir à un autre canal que celui à piloter, sans quoi un auto-verrouillage risque de se produire.
- Le canal à piloter doit se trouver en mode "Automatique" ou "Saisie manuelle".
- Si des conditions ET et OU sont également affectées au canal, celui-ci s'arrêtera dès que la condition correspondante sera remplie pour au moins une des deux fonctions.

Une ou plusieurs conditions d'arrêt du canal peuvent être affectées à chaque canal pilotant :

- **ASTOPA :**
Le mouvement synchrone du canal à piloter est arrêté tant que **toutes les conditions** sont remplies (liaison logique ET).
- **ASTOPO :**
Le mouvement synchrone du canal à piloter est arrêté tant qu'**au moins une condition** est remplie (liaison logique OU).

Notions de base pour la programmation CN



L'affectation de nouvelles conditions ET et OU invalide toutes les conditions appliquées jusqu'à présent sur ce canal.

Syntaxe :

Conditions ET :	
ASTOPA [<numéro de canal> , <cond.1> { , <cond.2> } { . . { , <cond.8> } . }]	
Conditions OU :	
ASTOPO [<numéro de canal> , <cond.1> { , <cond.2> } { . . { , <cond.8> } . }]	
avec :	
<numéro de canal>	Numéro du canal à piloter (1 à n). Valeur ou variable en nombre entier.
<cond.1, cond. 2...cond.8>	Définition de 1 à 8 condition(s) sous la forme : <axe><opérateur de comparaison><valeur de comparaison> avec :
<axe>	Nom d'axe du système ou du canal, programmé en tant que constante de chaîne CPL ou variable de chaîne CPL.
<opérateur de comparaison>	Opérateurs admissibles : < , ≤ , > , .
<valeur de comparaison>	Valeur réelle ou expression CPL réelle. La valeur est calculée lors de la préparation et reste active en tant que valeur modale.

Tab. 3-39: Syntaxe



Des conditions ET/OU permettent d'arrêter un maximum de 4 canaux à partir d'un autre canal.

Exemple :

Utilisation de noms d'axes et de numéros :

10 NOAXE% = 2	<i>Définition :</i>
20 NOMAXE\$ = "X"	• Numéro de l'axe
30 CANALARRET% = 2	• Nom de l'axe
:	• Numéro du canal
N40 ASTOPO[CANALARRET%, NOAXE%<10]	
:	
N90 ASTOPO[CANALARRET%, "Z">20.3]	
:	
N150 ASTOPO[CANALARRET%, NOMAXE\$<1.5]	

Arrêt du mouvement jusqu'à ce que la pièce à usiner atteigne une position de base : BSTOPA / BSTOPO

Description :

La fonction "BSTOPA / BSTOPO" permet de synchroniser des mouvements entre plusieurs canaux. Il est possible d'arrêter, puis de reprendre le mouvement synchrone sur un canal en fonction de la **position** d'une ou plusieurs coordonnées du **système de coordonnées de base de la pièce à usiner (BCS)** sur un autre canal.

Notions de base pour la programmation CN

Restrictions :

- Les coordonnées utilisées pour la synchronisation doivent appartenir à un autre canal que celui à piloter, sans quoi un auto-verrouillage risque de se produire.
- Le canal à piloter doit se trouver en mode "Automatique" ou "Saisie manuelle".
- Si des conditions ET et OU sont également affectées au canal, celui-ci s'arrêtera dès que la condition correspondante sera remplie pour au moins une des deux fonctions.

Une ou plusieurs conditions d'arrêt du canal peuvent être affectées à chaque canal pilotant :

- **BSTOPA :**
Le mouvement synchrone du canal à piloter est arrêté tant que **toutes les conditions** sont remplies (liaison logique ET).
- **BSTOPO :**
Le mouvement synchrone du canal à piloter est arrêté tant qu'**au moins une condition** est remplie (liaison logique OU).



L'affectation de nouvelles conditions ET et OU invalide toutes les conditions appliquées jusqu'à présent sur ce canal.

Syntaxe :

Conditions ET :		
BSTOPA[<numéro de canal> , <cond.1> { , <cond.2> } { . . { , <cond.8> } . }]		
Conditions OU :		
BSTOPO[<numéro de canal> , <cond.1> { , <cond.2> } { . . { , <cond.8> } . }]		
avec :		
<numéro de canal>	Numéro du canal à piloter (1 à n) Valeur ou variable en nombre entier	
<cond.1, cond. 2...cond.8>	Définition de 1 à 8 condition(s) sous la forme : <coordonnée><opérateur de comparaison><valeur de comparaison> avec :	
	<coordonnée>	Concernant BCS : Nom de coordonnée ou index de coordonnées programmé en tant que constante de chaîne CPL ou variable de chaîne CPL.
	<opérateur de comparaison>	Opérateurs admissibles : < , ≤ , > , .
	<valeur de comparaison>	Valeur réelle ou expression CPL réelle. La valeur est calculée lors de la préparation et reste active en tant que valeur modale.

Tab. 3-40: Syntaxe BSTOPA / BSTOPO



Des conditions ET/OU permettent d'arrêter un maximum de 4 canaux à partir d'un autre canal.

Notions de base pour la programmation CN

Exemple : Activer une condition ET pour les coordonnées de base de la pièce à usiner :

N10 BSTOPA[3,"z"<12.0,"x">15]

Le canal 3 sera arrêté tant que les conditions suivantes sont appliquées sur le canal pilotant :

- Position de la coordonnée de base de la pièce à usiner $z < 12 \text{ mm ET}$
- Position de la coordonnée de base de la pièce à usiner $x > 15 \text{ mm}$

Arrêt du mouvement jusqu'à ce que la pièce à usiner atteigne une position : WSTOPA / WSTOPO

Description : La fonction "WSTOPA / WSTOPO" permet de synchroniser les mouvements entre plusieurs canaux. Il est possible d'arrêter, puis de reprendre le mouvement synchrone dans un canal en fonction de la **position** d'une ou plusieurs coordonnées du **système de coordonnées de la pièce à usiner (WCS)** sur un autre canal.

Restrictions :

- Les coordonnées utilisées pour la synchronisation doivent appartenir à un autre canal que celui à piloter, sans quoi un auto-verrouillage risque de se produire.
- Le canal à piloter doit se trouver en mode "Automatique" ou "Saisie manuelle".
- Si des conditions ET et OU sont également affectées au canal, celui-ci s'arrêtera dès que la condition correspondante sera remplie pour au moins une des deux fonctions.

Une ou plusieurs conditions d'arrêt du canal peuvent être affectées à chaque canal pilotant :

- **WSTOPA :**
Le mouvement synchrone du canal à piloter est arrêté tant que **toutes les conditions** sont remplies (liaison logique ET).
- **WSTOPO :**
Le mouvement synchrone du canal à piloter est arrêté tant qu'**au moins une condition** est remplie (liaison logique OU).



L'affectation de nouvelles conditions ET et OU invalide toutes les conditions appliquées jusqu'à présent sur ce canal.

Syntaxe :

Conditions ET :	
WSTOPA [<numéro de canal> , <cond.1> { , <cond.2> } { . . { , <cond.8> } . }]	
Conditions OU :	
WSTOPO [<numéro de canal> , <cond.1> { , <cond.2> } { . . { , <cond.8> } . }]	
avec :	
<numéro de canal>	Numéro du canal à piloter (1 à n) Valeur ou variable en nombre entier

Notions de base pour la programmation CN

<cond.1, cond. 2...cond.8>	Définition de 1 à 8 condition(s) sous la forme : <coordonnée><opérateur de comparaison><valeur de comparaison>	
	avec :	
	<coordonnée>	Concernant WCS : Nom de coordonnée ou index de coordonnées programmé en tant que constante de chaîne CPL ou variable de chaîne CPL.
	<opérateur de comparaison>	Opérateurs admissibles : < , ≤ , > , .
	<valeur de comparaison>	Valeur réelle ou expression CPL réelle La valeur est calculée lors de la préparation et reste active en tant que valeur modale.

Tab. 3-41: Syntaxe WSTOPA / WSTOPO



Des conditions ET/OU permettent d'arrêter un maximum de 4 canaux à partir d'un autre canal.

Exemple : Activer une condition ET pour les coordonnées de la pièce à usiner :

```
N10 WSTOPA[3,"Z"<12.0,"X">15]
```

Le canal 3 sera arrêté tant que les conditions suivantes sont appliquées sur le canal pilotant :

Position de la coordonnée Z de la pièce à usiner du canal inférieure à 12 mm ET position de la coordonnée X de la pièce à usiner supérieure à 15 mm

Suppression des conditions d'arrêt : OFFSTOPA / OFFSTOPO

Description : Suppression des conditions d'arrêt pour un canal de commande.

Syntaxe :

OFFSTOPA	Efface toutes les conditions d'arrêt ET
OFFSTOPO	Efface toutes les conditions d'arrêt OU

Tab. 3-42: Syntaxe OFFSTOPA / OFFSTOPO

Les points suivants s'appliquent aux fonctions de synchronisation :

- ASTOPO, BSTOPO, WSTOPO, OFFSTOPO
sont des fonctions modales qui s'annulent réciproquement.
- ASTOPA, BSTOPA, WSTOPA, OFFSTOPA
sont des fonctions modales qui s'annulent réciproquement.

Suppression des conditions d'arrêt à la fin du programme (M30) :

Les conditions d'arrêt activées dans un canal sont supprimées implicitement à la fin d'un programme pièce (M30), de même que lors de la remise à zéro de ce canal ou de la désélection du programme.

3.19.4 Écriture d'une date système lors de l'exécution

Description La fonction "SSD / SSDE" écrit une valeur en cours d'exécution et l'attribue à une date système.

Notions de base pour la programmation CN

La valeur peut être déterminée à différents moments :

- **SSD :**

La valeur devant être attribuée à la date système n'est déterminée qu'au moment de l'exécution.

L'évaluation en cours d'exécution n'admet qu'une expression CPL simple.

- **SSDE :**

La valeur devant être attribuée à la date système est calculée au moment de la préparation (temps d'interprétation CPL), mais elle n'est affectée à la variable CPL permanente qu'au moment de l'exécution. En plus des types de données simples, il est aussi possible d'attribuer des chaînes.

Il est généralement recommandé d'utiliser SSDE. L'ordre SSD devrait seulement être utilisé si le contenu de l'expression CPL attribuée ne doit être calculé qu'au moment de l'exécution.



L'adresse cible de la date système est **toujours** calculée lors de la préparation. Cela signifie que si la date système est spécifiée à l'aide d'index de tableau, alors ces index ne doivent pas être modifiés à partir d'autres canaux.

Exemple:

Index dans la date système

```
N10 SSDE[SD.tableau[@6].Int = 4]
```

Si la variable @6 n'est pas définie sur ce canal, une fonction WAIT doit être programmée avant N10 (en plus de la synchronisation du canal).

Syntaxe :

La valeur à attribuer n'est déterminée qu'au moment de l'exécution :	
SSD[<date système> = <expression CPL simple>]	
La valeur à attribuer est déterminée au moment de la préparation :	
SSDE[<date système> = <expression CPL>]	
avec :	
<date système>	Date système identifiée par "SD." et suivie du nom de la variable.

Notions de base pour la programmation CN

<expression CPL simple>	Afin de ne pas entraver la génération du mouvement, seules les expressions CPL simples peuvent être évaluées lors de l'exécution. Une expression CPL simple est une expression mathématique composée de données système, de variables CPL permanentes, de constantes et d'opérations mathématiques possibles en CPL. <i>Les types de données autorisés sont les suivants :</i> • INT • BOOL • REAL • DOUBLE
<expression CPL>	Expression mathématique quelconque en langage de programmation CPL. <i>Les types de données autorisés sont les suivants :</i> • INT • BOOL • REAL • DOUBLE • Tableaux CHAR

Tab. 3-43: Syntaxe SSD / SSDE

Exemple :

N10 SSDE[SD.VarTest=1]

La valeur 1 est attribuée à la date système "SD.VarTest" lors de l'exécution.

N10 SSDE[SD.Array[4].Pos=7.2*(SIN(ANG) +5)]

La valeur de l'expression "7.2*(SIN(ANG%) +5)" est attribuée à "SD.Array[4].Pos".

N10 SSDE[SD.Array[J%].Pos=4]

La valeur 4 est attribuée à "SD.Array[J%].Pos" lors de l'exécution.

N20 SPV[SD.MeasPos= 3.5*@MPOS! + 12.2]

La valeur de l'expression "3.5*@MPOS! + 12.2" est déterminée lors de l'exécution, puis elle est attribuée à "SD.MeasPos".

- Particularités et restrictions :
- En raison de l'accès aux données système, le temps de liaison des fonctions SSD et SSDE est plus long que pour des ordres comparables. Il est donc recommandé d'utiliser ces ordres avec parcimonie afin de ne pas prolonger inutilement le temps de liaison (même en cas de "liaison ultérieure").
 - Il est recommandé d'utiliser SSDE. L'ordre SSD devrait seulement être utilisé si le contenu de l'expression CPL attribuée ne doit être calculé qu'au moment de l'exécution.

4 Aperçu des groupes modaux de fonctions CN

Un grand nombre de fonctions CN proposées par la commande MTX est configuré dans les groupes de fonctions CN. Les différentes fonctions CN d'un groupe modal se désactivent réciproquement, si bien qu'une seule fonction d'un groupe modal peut avoir effet.

Groupe modal	Fonctions CN
Géométrie	G0, G1, G2, G3, G5, G6, G33
Guidage de vitesse	G8, G9, CFD, LNU, LND, SNU, SND, S2U, S2D
Niveau	G16, G17, G18, G19, G20
Programmation absolue/incrémentale	G90, G91
Programmation de l'avance	G93, G94, G95
Programmation de la vitesse de rotation de la broche	G96, G97
Pouce/Métrique	G70, G71
Correction de la longueur d'outil	G47, G48
Correction 2,5D du rayon d'outil	G40, G41, G42
Correction 3D du rayon d'outil	G140, G141, G142
Correction D	D0, D
Correction ED	ED0, ED
Correction 2,5D du rayon d'outil	G40, G41, G42
Angles extérieurs avec la correction 2,5D du rayon	G43, G44
Avance avec la correction 2,5D du rayon	G45, G46
Surveillance de collision pour la correction 2,5D du rayon	CLN, Off**
Inpos en avance	G61, G62
Fenêtre Inpos active	IPS1, IPS2, IPS3
Décalage d'origine banque 1	G52, G53.1, G54, G55, G56, G57, G58, G59
Décalage d'origine banque 2	G52.2, G53.2, G54.2 G55.2, G56.2, G57.2, G58.2, G59.2
Décalage d'origine banque 3	G52.3, G53.3, G54.3 G55.3, G56.3, G57.3, G58.3, G59.3
Décalage d'origine banque 4	G52.4, G53.4, G54.4 G55.4, G56.4, G57.4, G58.4, G59.4
Décalage d'origine banque 5	G52.5, G53.5, G54.5 G55.5, G56.5, G57.5, G58.5, G59.5
Placement, plan incliné banque 1	G151.1, G152.1, G153.1, G154.1 G155.1, G156.1, G157.1, G158.1, G159.1
Placement, plan incliné banque 2	G151.2, G152.2, G153.2, G154.2 G155.2, G156.2, G157.2, G158.2, G159.2
Placement, plan incliné banque 3	G151.3, G152.3, G153.3, G154.3 G155.3, G156.3, G157.3, G158.3, G159.3
Placement, plan incliné banque 4	G151.4, G152.4, G153.4, G154.4 G155.4, G156.4, G157.4, G158.4, G159.4
Placement, plan incliné banque 5	G151.5, G152.5, G153.5, G154.5 G155.5, G156.5, G157.5, G158.5, G159.5

Aperçu des groupes modaux de fonctions CN

Groupe modal	Fonctions CN
Placement, Correction de la position de la pièce à usiner	BCR, Off**
Décalage des coordonnées	TRS, Off*
Décalage additif des coordonnées	ATR, Off**
Aide à la saisie, décalage de contour	SHT, Off**
Aide à la saisie, mise à l'échelle	SCL, Off**
Aide à la saisie, fonction miroir	MIR, Off**
Aide à la saisie, tourner	ROT, Off**
Aide à la saisie, pôle	PLS, Off**
Aide à la saisie, couplage de programmation	PLP, EPC, Off**
Arrondissages et chanfreins	RND, RNE, CHF, CHL, Off**
Arrondissage avec splines	SCO, Off**
Remplacement des droites par des splines	LTS, Off**
Override	OVD, OVE
Programmation du diamètre	DIA, RAD
Affectation de correction	G78, G79
Pôle pour coordonnées polaires	POP, Off**
Vitesse de l'axe	AVE, Off**
Accélération de l'axe	AAC, Off**
Accélération sur trajectoire	PAC, Off**
Programmation KV	KVP, Off**
Anticipation	FFW, Off**
Mode de positionnement pour axes sans fin	PMD, Off**
Programmation du couple de réduction	RDT, Off**
Ajustement de l'avance	FAD, Off**
Poinçonnage	PUN, NIB, Off**
Guidage tangentiel de l'outil	TTL, Off**
Orientation tangentielle de l'outil	TTO, Off*
Surveillance de zone	ARA, Off**
Division	SPLIT, Off**
Couplage d'axes	AXC, Off**
Couplage sélectif de coordonnées	SCC, SCCT, Off**
Tables de compensation	GCT, Off**
Laser	LFP, Off**
Raccord tangentiel	DTT, Off**
Programmation de précision	PRP, Off**

Aperçu des groupes modaux de fonctions CN

Groupe modal	Fonctions CN
Sous-programme modal	G80, G81, G82, G83...
Interpolation G0 asynchrone	CG0(ASY1), CG0(ASY0)

Off**

Fonction de mise hors service par programmation de la fonction CN avec (0) ou (), par ex. AAC(0)

Tab. 4-1:

Aperçu des groupes modaux de fonctions CN

5 Fonctions CN avec syntaxe conforme DIN 66025 (extensions incl.)



Mouvement de déplacement non intentionnel !

De nombreuses fonctions CN attendent la programmation d'adresses d'axes ou de coordonnées. En règle générale, outre les adresses de coordonnées, seules les adresses des axes respectifs du canal, c'est-à-dire des axes synchrones, peuvent être programmées.

En général, la programmation de l'adresse d'un axe asynchrone au sein d'une liste de paramètres entraîne une erreur de syntaxe, tandis que la programmation en dehors d'une liste de paramètres entraîne un mouvement de déplacement asynchrone !

Exceptions :

- G74(Home) VA1 :
L'axe asynchrone VA approche le point de référence.
- GAX(VA) :
L'axe asynchrone VA est accepté sur le canal.
- FsMove, FsTorque, FsReset :
La fonctionnalité "Déplacement sur butée fixe" a également un effet sur les axes asynchrones.

5.1 Vue d'ensemble

La commande dispose d'un grand nombre de fonctions CN. Hormis les ordres définis dans la norme DIN 66025, elle contient également des extensions essentielles dans le domaine des codes G et des éléments de syntaxe additionnels similaires au langage standard.

Les fonctions CN de la commande comprennent :

- **les codes G à 1 et 2 chiffre(s) :**
Ils répondent en grande partie à la norme DIN 66025 et complètent son "vocabulaire des codes G" de manière judicieuse afin :
 - d'étendre les groupes de fonctions compris dans la norme DIN (par ex. **G52**, G53, G54...) ou
 - d'introduire de nouveaux groupes de fonctions en rapport avec les groupes de fonctions déjà existants.
- Les codes G à 1 chiffre peuvent toujours s'écrire avec 2 chiffres en utilisant un zéro non significatif en tête (par ex. G0=G00, G1=G01...).
- **les codes G à 3 chiffres** (cas exceptionnels) :
Ils sont utilisés lorsqu'une fonctionnalité préexistante dans le domaine 2D est introduite de façon analogue dans le domaine 3D (par ex. G41, G42 → G141, G142).
- **les codes G suivis de "." :**
Ils sont utilisés par exemple en combinaison avec des traits à points, des décalages d'origine ou des transformations de coordonnées. Ces dernières fonctions possèdent jusqu'à 5 banques qui peuvent être adressées/programmées via le suffixe ".".

Exemples :

Fonctions CN avec syntaxe conforme DIN 66025 (extensions incl.)

G1.2 Trait à 2 points en avance

G59.2 G59 Décalage d'origine de la banque 2

- **les fonctions M ayant une fonctionnalité CN définie**
par ex. M0, M30
- **les fonctions CN avec la syntaxe de langage standard :**
(voir [chap. 6 "Fonctions CN avec syntaxe de langage standard" à la page 221](#))



Vous trouverez un tableau synoptique de toutes les fonctions CN au [chap. 8 "Annexe" à la page 643](#).

En outre, toutes les fonctions CN sont répertoriées dans l'index.

Écritures utilisées : La présente documentation utilise l'écriture suivante pour la syntaxe des fonctions CN :

Police de caractères "Courier gras" ou "Courier"

Les chaînes de caractères dans cette police de caractères doivent être programmées comme indiqué.

Exemple : G0 (POL)

Crochets triangulaires < >

Ils identifient un caractère de substitution pour une expression/un paramètre à programmer. Le caractère de substitution est écrit en italique.

Exemple : <axe1>

Accolades { }

Elles identifient une expression/un paramètre optionnel.

De tels éléments de syntaxe peuvent être programmés, mais ce n'est pas une obligation.

Exemple : G0 { ({ POL, } { <par1> }) }

Caractère "|"

Il sépare les paramètres possibles qui s'excluent réciproquement (paramètres alternatifs).

Exemple : G0 { ({ POL, } { NIPS | IPS1 | IPS2 | IPS3 }) }

Unités de mesure : *Sauf indication contraire, la programmation CN repose sur les unités de mesure suivantes :*

- Positions translationnelles :
[mm] avec G71, [pouce] avec G70
- Positions rotatives :
[degré] pour les axes ronds
- Vitesses translationnelles :
[mm/min] avec G71, [pouces/min] avec G70
- Vitesses rotatives :
[degré/min]
- Accélérations translationnelles :
[m/sec²] avec G71, [1000 pouces/sec²] avec G70
- Accélérations rotatives :
[1000 degrés/sec²]

5.2 Codes G

5.2.1 Interpolation linéaire en avance rapide "G0"

Description La position programmée est atteinte par interpolation sur une droite à vitesse maximale sur trajectoire (= en avance rapide).

Les points suivants s'appliquent ici :

- Un axe au moins se déplace à la vitesse maximale ou en accélération. La vitesse des axes est réglée de sorte que tous atteignent la position cible au même moment.
- Si la fonction est activée, le signal IF du canal "Avance rapide active" est émis.
- La fonction est modale et supprime G1, G2, G3, G5, G6, G33.
- La distance de poursuite (décalage entre la valeur de consigne et la valeur réelle) à la fin du bloc peut être réduite à la taille de la fenêtre cible définie (= fonctionnalité "Arrêt précis").

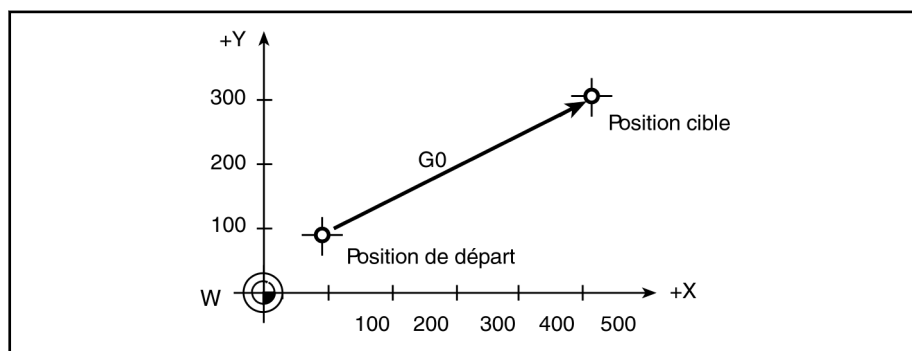


Fig. 5-1: Interpolation linéaire en avance rapide

Syntaxe : Fonction de base

G0	Avance rapide avec arrêt précis ACTIVÉ et programmation de la position finale en coordonnées cartésiennes. On tient compte pour cela de la dernière taille de la fenêtre d'arrêt précis réglée pour l'avance rapide (par défaut = exacte). À la fin du bloc, la commande ralentit d'abord à la vitesse de déplacement $v = 0$. Le bloc suivant n'est déplacé qu'après que cette fenêtre d'arrêt précis a été atteinte pour tous les axes participants.

Tab. 5-1: Syntaxe G0

Paramètres optionnels

G0 ({NIPS IPS1 IPS2 IPS3},{POL}) avec	
NIPS	Arrêt précis DÉSACTIVÉ Sauf condition contraire, pas de décélération à $v = 0$ à la fin du bloc.
IPS1	Comme G0 sans paramètres, mais avec taille de la fenêtre d'arrêt précis exacte .
IPS2	Comme G0 sans paramètres, mais avec taille de la fenêtre d'arrêt précis approximative .

Fonctions CN avec syntaxe conforme DIN 66025 (extensions incl.)

IPS3	Comme G0 sans paramètres, mais avec taille de la fenêtre d'arrêt précis sans fin . Cela signifie qu'aucune fenêtre de positionnement n'est vérifiée, mais que la vitesse est seulement ralentie à $v = 0$ à la fin du bloc.
POL	Active la programmation des coordonnées polaires et définit l'angle polaire 1 sur 0, l'angle polaire 2 sur 90 degrés. Pour savoir comment programmer des coordonnées polaires, voir chap. 6.82 "Programmation des coordonnées polaires" à la page 347.

Tab. 5-2: Paramètres optionnels G0

Exemple :

```

:
N40 X100 Y100          Position de départ
N50 G0 X500 Y300       Approcher la position cible en avance rapide

```

Particularités et restrictions :

- La fonction peut être programmée avec ou sans adresses d'axes dans le même bloc.
- La vitesse correspondante est déterminée via les paramètres machine pour la vitesse maximale de l'axe.
- Il est possible de limiter la vitesse de l'avance rapide à la valeur réduite réglée dans les paramètres machine avec le signal IF du canal "Avance rapide limitée" (qCh_RedRap).
- La vitesse de l'avance rapide peut également être limitée via la fonction "Avance rapide de test" pilotée par le signal IF du canal "Avance rapide de test" (qCh_TestRap).
- Selon le paramètre machine
/CHAN/Ch[]/Path/RapTravOvrd/EnableRapTravIOvrd "Activer l'Override avance rapide" (7030 00014), la vitesse peut être influencée avec le potentiomètre d'avance ou d'avance rapide.
- Les modifications de l'arrêt précis (ACTIVÉ/DÉSACTIVÉ, taille de la fenêtre d'arrêt précis) effectuées avec G0 ne s'appliquent qu'à l'avance rapide.
- Pour modifier la taille de la fenêtre d'arrêt précis de manière générale, c.-à-d. pour l'avance rapide **et** pour toutes les fonctions d'avance, il convient d'utiliser les fonctions CN IPS1, IPS2 et IPS3.



Pour plus d'informations sur l'arrêt précis, voir G1, G61/62, IPS1, IPS2, IPS3 dans la présente documentation, ainsi que dans la documentation "Description des fonctions IndraMotion MTX", chapitre "Précision", section "Arrêt précis".

5.2.2 Interpolation linéaire en avance"G1"

Description : La position programmée est atteinte par interpolation sur une droite avec l'avance effective (programmable par l'adresse F).

Le mouvement est coordonné de sorte que tous les axes participants atteignent le point d'arrivée programmé en même temps.

Les points suivants s'appliquent ici :

- La valeur d'avance (F) programmée a l'effet d'une avance sur trajectoire. De ce fait, si plusieurs axes sont déplacés, la participation de chaque axe individuelle est inférieure à F.
- La fonction est modale et révoque G0, G2, G3, G5, G6, G33.

Fonctions CN avec syntaxe conforme DIN 66025 (extensions incl.)

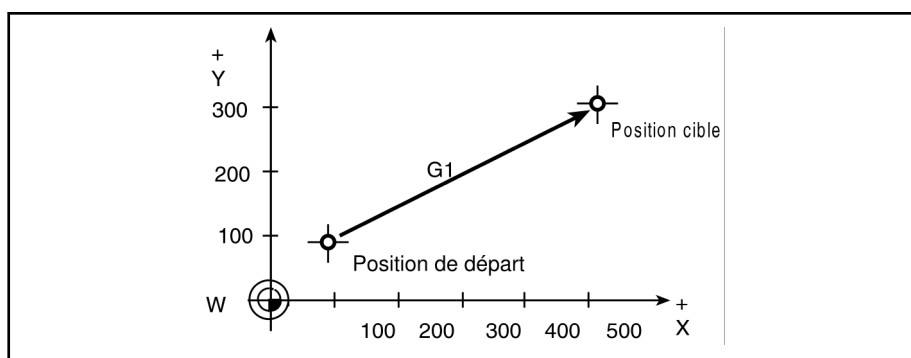


Fig. 5-2: Interpolation linéaire en avance

Syntaxe : **Fonction de base**

G1	Mouvement selon l'avance active et programmation de la position finale en coordonnées cartésiennes. L'arrêt précis dépend du fait que le code G61 (arrêt précis ACTIVÉ) ou G62 (arrêt précis DÉSACTIVÉ) est actif. On tient compte ici de la dernière taille de la fenêtre d'arrêt précis réglée pour l'avance (par défaut = exacte).

Tab. 5-3: Syntaxe G1

Paramètres optionnels

G1 ({IPS IPS1 IPS2 IPS3},{POL})	
avec	
IPS	Arrêt précis ACTIVÉ, indépendamment de G61/G62. On tient compte ici de la dernière taille de la fenêtre d'arrêt précis réglée pour le mode d'avance (par défaut = exacte). À la fin du bloc, la commande réduit d'abord la vitesse sur trajectoire en ralentissant à la vitesse $v = 0$. Le bloc suivant n'est déplacé que lorsque cette fenêtre de positionnement a été atteinte pour tous les axes participants. Cette activation de l'arrêt précis ne s'applique qu'à G1. Pour toutes les autres fonctions d'avance, G61/G62 restent valables.
IPS1	Comme IPS, mais avec taille de la fenêtre d'arrêt précis exacte . Le réglage de la taille de la fenêtre d'arrêt précis s'applique également à toutes les autres fonctions d'avance, exception faite de l'avance rapide (G0), tandis que l'arrêt précis est uniquement activé pour G1.
IPS2	Comme IPS, mais avec taille de la fenêtre d'arrêt précis approximative . Le réglage de la taille de la fenêtre d'arrêt précis s'applique également à toutes les autres fonctions d'avance, exception faite de l'avance rapide (G0), tandis que l'arrêt précis est uniquement activé pour G1.
IPS3	Comme IPS, mais avec taille de la fenêtre d'arrêt précis sans fin . Cela signifie qu'aucune fenêtre de positionnement n'est vérifiée, mais que la vitesse est seulement ralentie à $v = 0$ à la fin du bloc. Le réglage de la taille de la fenêtre d'arrêt précis s'applique également à toutes les autres fonctions d'avance, exception faite de l'avance rapide (G0), tandis que l'arrêt précis est uniquement activé pour G1.
POL	Active la programmation des coordonnées polaires et définit l'angle polaire 1 sur 0, l'angle polaire 2 sur 90 degrés. Pour savoir comment programmer des coordonnées polaires, voir chap. 6.82 "Programmation des coordonnées polaires" à la page 347.

Tab. 5-4: Paramètres optionnels G1

Fonctions CN avec syntaxe conforme DIN 66025 (extensions incl.)

Exemple : Fonction de base :

```

:
N40 X100 Y100          Position de départ
N50 G0 X500 Y300 F100 Approcher la position finale avec l'avance F100.
  
```

- Particularités et restrictions :**
- La fonction peut être programmée avec ou sans adresses d'axes dans le même bloc.
 - Si aucune avance n'a été activée jusque là, une avance doit être programmée sur la même ligne via l'adresse F.
Cette avance reste active jusqu'à ce qu'elle soit écrasée par une nouvelle valeur d'avance.
 - La vitesse sur trajectoire programmée peut être limitée à l'aide des paramètres machine.
 - Il est possible d'influencer la vitesse sur trajectoire à l'aide du potentiomètre d'avance.
 - La taille de la fenêtre d'arrêt rapide peut également être modifiée pour toutes les fonctions d'avance à l'aide des fonctions CN G61(IPS1), G61(IPS2) et G61(IPS3).
 - Pour modifier la taille de la fenêtre d'arrêt précis de manière générale, c.-à-d. pour l'avance rapide **et** pour toutes les fonctions d'avance, il convient d'utiliser les fonctions CN IPS1, IPS2 et IPS3.



Pour plus d'informations sur l'arrêt précis, voir G0, G61/62, IPS1, IPS2, IPS3 dans la présente documentation, ainsi que dans la documentation "Description des fonctions IndraMotion MTX", chapitre "Précision", section "Arrêt précis".

5.2.3 Interpolation circulaire/hélicoïdale/hélicoïdale N"G2, G3"

Généralités

Description Approche la position finale programmée dans le plan de travail actif par interpolation sur une trajectoire circulaire avec l'avance actuelle (programmable via l'adresse F).

- G2 : trajectoire circulaire dans le sens des aiguilles d'une montre (antitrigonométrique)
- G3 : trajectoire circulaire dans le sens inverse des aiguilles d'une montre (trigonométrique)

D'autres axes peuvent aussi être programmés en dehors du plan de travail, dont les mouvements linéaires sont co-interpolés.

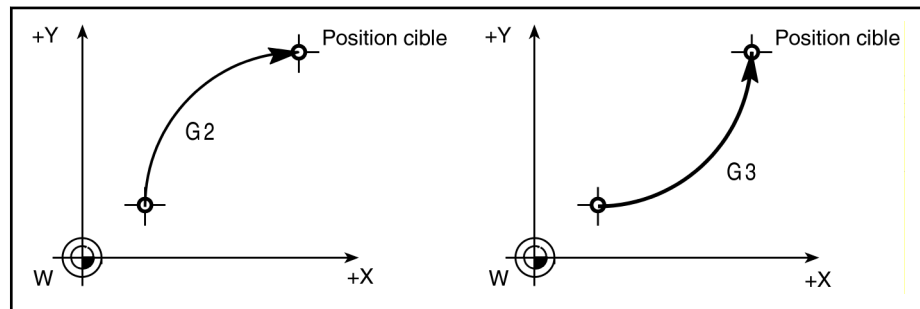


Fig. 5-3: G2 Trajectoire circulaire dans le sens antitrigonométrique ; G3 Trajectoire circulaire dans le sens trigonométrique

Fonctions CN avec syntaxe conforme DIN 66025 (extensions incl.)

Interpolation circulaire :

Seuls les axes du plan de travail actif y participent.

Interpolation hélicoïdale :

Outre les axes du plan de travail actif, un axe synchrone **supplémentaire** participe.

Dans la mesure où celui-ci est perpendiculaire au plan de travail actif (par ex. l'axe restant du système de coordonnées de la pièce à usiner), une trajectoire suivant la forme d'une spirale à pas constant est créée (interpolation hélicoïdale).

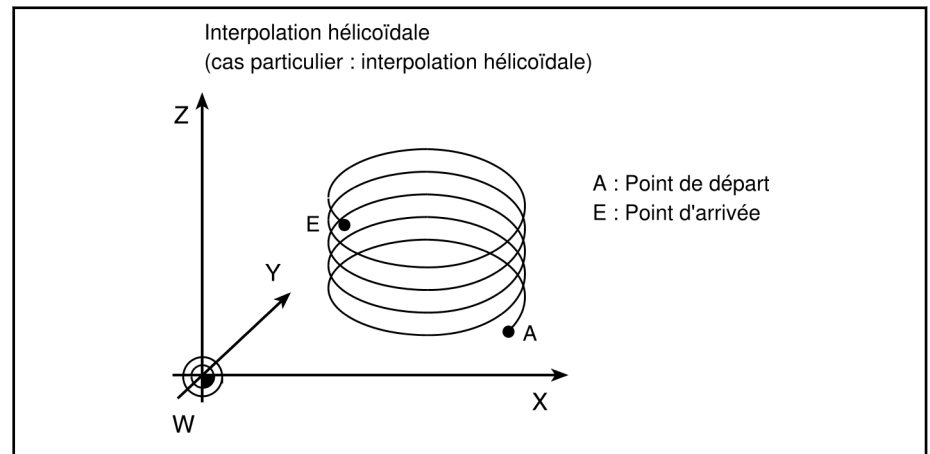


Fig. 5-4: Interpolation hélicoïdale

Interpolation hélicoïdale N :

Outre les axes du plan de travail actif, plusieurs axes synchrones **supplémentaires** sont programmés (6 au maximum).

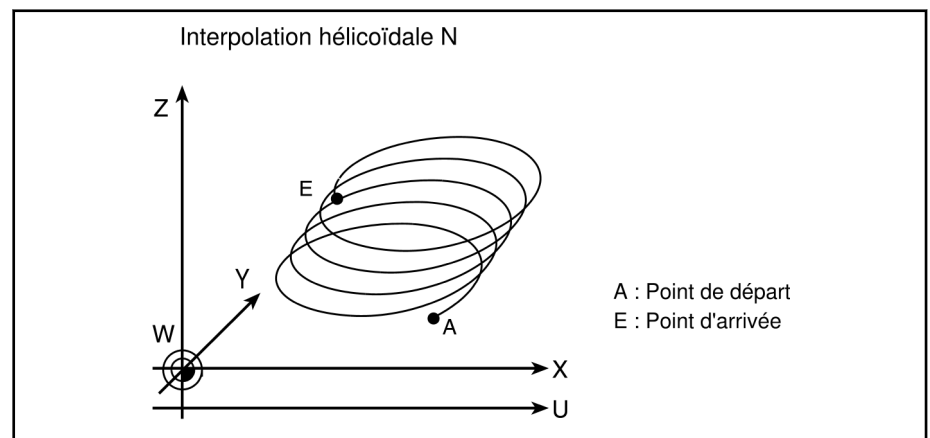


Fig. 5-5: Interpolation hélicoïdale N

Les points suivants s'appliquent ici :

- Les axes qui se déplacent sur l'arc de cercle sont définis clairement par les plans de travail sélectionnés (G17, G18, G19, G20).
- La vitesse d'avance (F) programmée a l'effet d'une avance sur trajectoire et se rapporte généralement à tous les axes à déplacer dans un bloc. Par conséquent, si plusieurs axes sont déplacés, la participation de chaque axe individuel est inférieure à F.
- L'avance réelle peut être limitée par l'accélération radiale maximale possible.

Fonctions CN avec syntaxe conforme DIN 66025 (extensions incl.)

- Les fonctions G0, G1, G2, G3, G5, G6 constituent un groupe modal et se révoquent donc réciproquement.
- *La trajectoire circulaire peut être programmée par :*
 - la programmation du rayon ou
 - la programmation du centre.

Programmation du rayon

Description : *Une trajectoire circulaire est définie à partir de la position actuelle (point de départ) en programmant :*

- le point d'arrivée du cercle et
- le rayon du cercle.

Le point d'arrivée du cercle doit donc être indiquée en tant que valeur de position absolue ou incrémentale.

Syntaxe : **Fonction de base :**

G2<EP> R<valeur>	Trajectoire circulaire antitrigonométrique
G3<EP> R<valeur>	Trajectoire circulaire trigonométrique
avec	
<EP>	Coordonnées du point d'arrivée

Fonctions CN avec syntaxe conforme DIN 66025 (extensions incl.)

<valeur>	Rayon de la trajectoire circulaire Le rayon doit être au moins égal à la moitié de la distance séparant le point de départ et le point d'arrivée. Utiliser le signe (+/-) pour définir si, dans l'éventualité où deux trajectoires circulaires sont possibles, la trajectoire devant être déplacée est celle avec le petit ou le grand arc de cercle : Rayon positif : arc de cercle ≤ 180 degrés Rayon négatif : arc de cercle > 180 degrés Si le rayon est exactement égal à la moitié de la distance qui sépare le point de départ et le point d'arrivée, on obtient alors un demi-cercle et le signe devant la valeur de rayon n'a aucune influence.

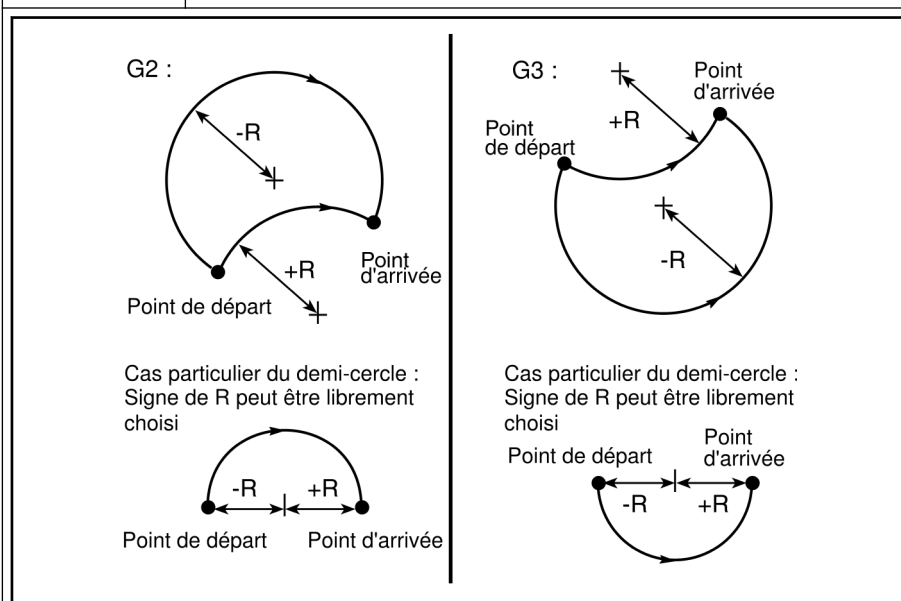


Fig. 5-6: Rayon des trajectoires circulaires G2 et G3

Tab. 5-5: Syntaxe G2, programmation du rayon

Particularités et restrictions :

- Il est impossible de générer des cercles complets.
 - La commande corrige automatiquement les indications de rayon imprécises selon la tolérance réglée dans les paramètres machine.
- Sinon, une erreur d'exécution est générée.

Programmation du centre

Fonction de base

Description : Une trajectoire circulaire est définie à partir de la position actuelle (point de départ) en programmant :

- le point d'arrivée du cercle et
- le centre du cercle.

Le centre du cercle peut être programmé de manière incrémentale ou absolue.

Fonctions CN avec syntaxe conforme DIN 66025 (extensions incl.)

Syntaxe : Fonction de base :

G2<EP> <IP>	Trajectoire circulaire antitrigonométrique
G3<EP> <IP>	Trajectoire circulaire trigonométrique
avec	
<EP>	Coordonnées du point d'arrivée Si le point de départ et le point d'arrivée sont identiques au sein du plan circulaire, alors un cercle complet est généré automatiquement.
<IP>	Paramètres d'interpolation I, J et K Pour la programmation relative locale, ces paramètres définissent pour chaque axe la distance qui sépare le point de départ A du cercle et le centre M du cercle. Le signe découle de l'orientation vectorielle de A vers M. La signification des coordonnées dans les paramètres machine permet de déduire quel paramètre d'interpolation est attribué à quel axe. Règle standard : I = M(X) - A(X) pour la coordonnée X J = M(Y) - A(Y) pour la coordonnée Y K = M(Z) - A(Z) pour la coordonnée Z Il est également possible de programmer le centre du cercle en coordonnées absolues à l'aide de la programmation absolue locale.

G2 :

+Y

A : Point de départ
E : Point d'arrivée
M : Centre

G3 :

+Y

A : Point de départ
E : Point d'arrivée
M : Centre

Fig. 5-7: Programmation du centre

Tab. 5-6: Syntaxe G2, programmation du centre

Paramètres optionnels

Syntaxe :	G2 / G3 (POL) <EP> R<valeur>	Programmation du rayon en coordonnées polaires
	G2 / G3 (POL) <EP> <IP>	Programmation du centre en coordonnées polaires

Fonctions CN avec syntaxe conforme DIN 66025 (extensions incl.)

N30 G0 G90 X-10 Y0 Z0	Approcher la position de départ.
N50 G2 I10 Z=IC([-PITCH])	Chaque spire avec le pas affecté.
RPT[TURN] F1000	

- Particularités et restrictions :**
- Si le point de départ et le point d'arrivée sont identiques, alors la commande génère automatiquement un cercle entier.
 - La commande corrige automatiquement les indications de centre imprécises selon la tolérance définie dans les paramètres machine.
Pour ce faire, elle décale la position du centre en conséquence.
 - Si des paramètres d'interpolation et un rayon de cercle sont programmés dans un même bloc, alors le système ne tient compte que du rayon de cercle (= Programmation du rayon).
 - Si des paramètres d'interpolation ne correspondant pas au plan sélectionné ont été programmés, alors la commande génère un message d'erreur d'exécution.

Exemple : G17 G2 X5 I9 K7 (erreur : K ne correspond pas au plan X/Y)

5.2.4 Temporisation"G4"

Description : Interrompt l'exécution du programme. Le bloc programmé suivant n'est exécuté qu'à la fin de la temporisation programmée.

La temporisation commence lorsque le bloc CN précédent a été complètement exécuté.

Les broches en rotation ou axes secondaires en déplacement ne sont pas arrêtés. Le cas échéant, les axes synchrones peuvent réduire leur décalage de poursuite.

La temporisation peut être programmée :

- en secondes
- ou -
- en rotations de la broche.

Pour déterminer la vitesse de rotation de la broche, la vitesse réelle actuelle de la broche principale est déterminée cycliquement et utilisée pour calculer le nombre de tours exécutés. Dans le cas de broches très puissantes, une certaine différence peut donc apparaître entre la vitesse de broche programmée et la vitesse réellement attendue au cours des phases d'accélération et de freinage.

Si la broche principale configurée est une broche analogique (sans retour de la vitesse de rotation), la vitesse de rotation de consigne est utilisée pour le calcul à la place de la vitesse de rotation réelle.

Syntaxe :

G4 (F<valeur>)	Temporisation en secondes.
G4 (S<valeur>)	Temporisation en rotations de la broche
avec	
<valeur>	Indication en secondes ou en rotations de la broche Saisir "0" entraîne le rejet du bloc G4 en interne.

Tab. 5-8: Syntaxe G4

- Particularités et restrictions :**
- La fonction doit être programmée dans un bloc à part, sans information sur la course. Seules des fonctions auxiliaires et additionnelles restent autorisées dans ce bloc.

Fonctions CN avec syntaxe conforme DIN 66025 (extensions incl.)

- Les rotations programmées de la broche font référence à la broche principale configurée dans les paramètres machine ou par la fonction "MainSp" (voir [chap. 6.70 "Changement de broche principale MainSp, MSP"](#) à la page 323).

5.2.5 Entrée tangentielle dans le cercle "G5"

Description : La commande calcule automatiquement une entrée tangentielle dans le cercle pour le bloc G5 à l'aide du dernier mouvement de déplacement programmé.

Un raccord n'est tangentiel que s'il ne présente aucune inversion de direction.



Si plusieurs mouvements G5 s'enchaînent, la 1^{re} tangente d'entrée influence tous les éléments de contour G5 suivants.

Les fonctions "G0, G1, G2, G3, G5, G6" constituent un groupe modal et se révoquent donc réciproquement.

AVIS

Des marqueurs de traitement peuvent apparaître au niveau de l'enchaînement des blocs en interpolation hélicoïdale/hélicoïdale N !

Le raccord tangentiel calculé se rapporte uniquement au plan circulaire ! La tangente dans l'espace peut sauter au niveau de l'enchaînement des blocs !

Syntaxe :

G5	<EP>
avec	
<EP>	Coordonnées du point d'arrivée du cercle

Tab. 5-9: Syntaxe G5

Particularités et restrictions :

- La programmation de G5 en mode "Saisie manuelle" n'est pas possible.
- Un bloc avec un mouvement de déplacement doit être programmé dans le même programme avant G5.
- Le plan actuel ne doit pas être commuté juste avant ni pendant l'activation de G5.

Exemples :

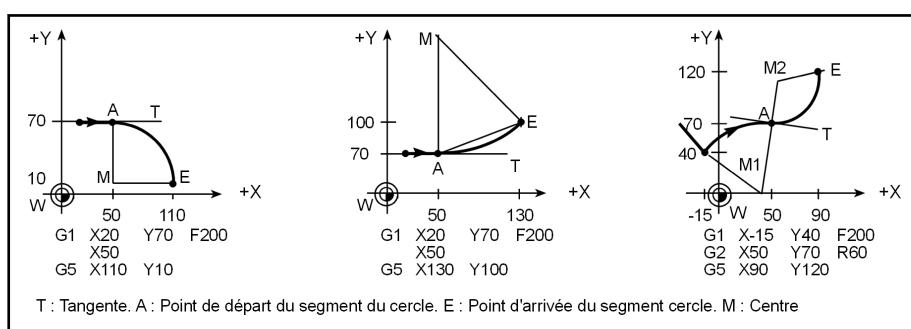


Fig. 5-10: Exemple - Entrée tangentielle dans le cercle

5.2.6 Programmation de spline "G6"

Aperçu – Types de splines

Description : Comparée à l'interpolation linéaire, l'interpolation spline fournit une meilleure qualité de surface et une plus grande précision du contour avec un nombre réduit de repères, car des courbes constantes sont calculées entre les points.

Fonctions CN avec syntaxe conforme DIN 66025 (extensions incl.)

La commande IndraMotion MTX prend en charge les types de splines suivants :

- **Type de spline 0 :**
Spline avec programmation de coefficients
(coefficients polynomiaux du système CAO/FAO)
- **Type de spline 1 :**
Splines cubiques C^1 constantes avec programmation de repères
(raccords tangentiels au niveau des repères)
- **Type de spline 2 :**
Splines cubiques C^2 constantes avec programmation de repères
(raccords à courbure constante au niveau des repères)
- **Type de spline 3 :**
Spline B avec programmation de points de contrôle
(tracé de la courbe proche des repères)
- **Type de spline 4 :**
Approximation de spline B (compresseur)

Le type de spline voulu est sélectionné et initialisé avec la fonction "Spline-Def" (SPD, voir [chap. 6.130 "Définition du type de spline SplineDef, SPD" à la page 415](#)). Ensuite, la programmation spline peut être activée à l'aide de la fonction G6.



Pour plus d'informations sur chaque type de spline, voir la documentation "Description des fonctions IndraMotion MTX".

Syntaxe :

G6	Activation du type de trajectoire "Spline".

Tab. 5-10: Syntaxe G6

En fonction du type de programmation requis, il est possible de programmer G6 pour les différents types de splines avec différents paramètres modaux.

Particularités et restrictions :

Les fonctions suivantes ne peuvent **pas** être programmées conjointement avec des splines :

- Orientation tensorielle
- Correction 2D de la trajectoire G41/G42
- Poinçonnage/Grignotage avec répartition de la trajectoire
- Chanfreins et arrondis
- Guidage tangentiel de l'outil
- Programmation de précision
- G5 à la suite d'une spline
- Effacement de la course restante

Spline avec programmation de coefficients (type de spline 0)

Programmation de coordonnées/d'axes

Description : *Chaque coordonnée du canal peut être déplacée comme suit :*

- en tant que spline, en saisissant des coefficients polynomiaux
- linéairement, en saisissant la position finale

Syntaxe : Spline par saisie des coefficients polynomiaux :

Fonctions CN avec syntaxe conforme DIN 66025 (extensions incl.)

<NomCoord>(<c0>,<c1>,...,<cn>)	
	Programmation des différentes coordonnées avec des coefficients polynomiaux.

Tab. 5-11: Mouvement de coordonnées en tant que spline

- ou -

Linéairement par saisie de la position finale :

<Nom-Coord>(<PosFin>)	Programmation de la position finale de chaque coordonnée/axe
avec	
<NomCoord>	Nom de la coordonnée/de l'axe
<c0>,<c1>,...,<cn>	Coefficients polynomiaux d'une coordonnée "n" correspond au degré de lissage paramétré dans "Spline-Def"
<PosFin>	Position finale de la coordonnée

Tab. 5-12: Mouvement de coordonnées "linéaire"

Exemple :
SplineDef(3)
G6 X(0.1,1.25,0.5,0.73) Y30 B(0.0,-1.0,0.1,-0.2)

Programmation d'un polynôme dénominateur

Syntaxe :	DN(<g ₀ >,<g ₁ >,...,<g _n >)	Polynôme dénominateur commun à toutes les coordonnées spline.
	avec	
	<g ₀ >, <g ₁ >, ..., <g _n >	Coefficients polynomiaux du polynôme dénominateur. "n" correspond au degré de lissage défini dans "SplineDef".

Tab. 5-13: Programmation d'un polynôme dénominateur

Exemple :
SplineDef(3)
G6 X(0.1,1.25,0.5,0.73) B(0.0,-1.0,0.1,-0.2) DN(1,0,1)

Programmation d'un vecteur d'orientation

Description : Ce type de programmation présuppose une transformation d'axe active qui prend en charge l'orientation vectorielle ("Coord(..)").

Syntaxe :	$O1(<o_{10}>, <o_{11}>, ..., <o_{1n}>)$	Composante x du vecteur d'orientation
	$O2(<o_{20}>, <o_{21}>, ..., <o_{2n}>)$	Composante y du vecteur d'orientation
	$O3(<o_{30}>, <o_{31}>, ..., <o_{3n}>)$	Composante z du vecteur d'orientation
	avec	
	$<o_{10}> , <o_{11}> , \dots , <o_{1n}> :$	Coefficients spline de la composante x du vecteur d'orientation.

Fonctions CN avec syntaxe conforme DIN 66025 (extensions incl.)

$\langle O_{20} \rangle, \langle O_{21} \rangle, \dots, \langle O_{2n} \rangle :$	Coefficients spline de la composante y du vecteur d'orientation.
$\langle O_{30} \rangle, \langle O_{31} \rangle, \dots, \langle O_{3n} \rangle :$	Coefficients spline de la composante z du vecteur d'orientation. "n" correspond au degré de lissage défini dans "SplineDef".

Tab. 5-14: Programmation d'un vecteur d'orientation

Exemple :

```

N00 ; coefficients spline pour l'orientation vectorielle
001 PI=3.14159:PIH=PI/2:PIHQ=PIH*PIH:PIHC=PIHQ*PIH
N10 G1 F30000 X0 Y0 Z0 B90 C0
N20 SplineDef(3)
N30 Coord(1) ; transformation géométrique à 5 axes avec
orientation vectorielle activée
N40 G6 PL[PIH] ; pour PL, voir la programmation de la
longueur des paramètres de la spline
N50 O1(1,0,-3/PIHQ,2/PIHC) O3(0,1,(3-PI)/PIHQ,(-2+PIH)/
PIHC)
N60 O1(0,0,3/PIHQ,-2/PIHC) O3(1,0,(-3+PIH)/PIHQ,(2-PIH)/
PIHC)
N70 O(0,1,0) ; orientation vectorielle normale
N80 G1
N90 Coord(0)

```

Programmation de la longueur des paramètres de la spline

Description : La longueur des paramètres de la spline est la longueur de l'intervalle de définition de "w". "w" s'étend de 0 à w_e . La valeur w_e est modale et reste valide pour tous les blocs CN jusqu'à la désélection de G6.

La fonction PL doit être programmée dans le premier bloc de déplacement après G6, ou un message d'erreur d'exécution est généré.

Syntaxe :	$\{ PL\langle w_e \rangle \}$	Programmation optionnelle de la longueur des paramètres de la spline
	avec	
	$\langle w_e \rangle$	Valeur quelconque > 0

Tab. 5-15: Programmation de la longueur des paramètres de la spline

Exemple :

```

G6 X(0.1,1.25,0.5,0.73) B(0.0,-1.0,0.1,-0.2) PL0.6
(X = 0.1 + 1.25 w + 0.5 w2 + 0.73 w3 et
B = 0.0 - 1.0 w + 0.1 w2 - 0.2 w3 avec
w s'étendant de 0 à 0.6)

```

Spline d'Hermite (type de spline 1)**Programmation de coordonnées/d'axes**

Description : Chaque coordonnée du canal peut être déplacée au choix :

- en tant que spline, en saisissant la position de la 1re dérivée
- ou -
- linéairement, en saisissant la position finale.

Fonctions CN avec syntaxe conforme DIN 66025 (extensions incl.)

Syntaxe : Spline par la saisie de la position et de la 1re dérivée :

<NomCoord>(<c0>,<c1>)	Programmation de coordonnées individuelles avec la position ($\pm c0$) et la première dérivée ($\pm c1$).

Tab. 5-16: Mouvement de coordonnées en tant que spline

- ou -

Linéairement par saisie de la position finale :

<NomCoord>(<PosFin>)	Programmation de la position finale de chaque coordonnée/axe.

Tab. 5-17: Mouvement de coordonnées "linéaire"

avec	
<NomCoord>	Nom de la coordonnée/de l'axe
<c0>,<c1>	Coefficients polynomiaux d'une coordonnée, où $c0 \pm$ position et $c1 \pm$ première dérivée
<PosFin>	Position finale de la coordonnée

Tab. 5-18: Mouvement de coordonnées

Exemple : SplineDef(1143)

G6 X(0.1,1.25) Y30 B(0.0,-1.0)

Programmation de la longueur des paramètres de la spline

Description : La longueur des paramètres de la spline est la longueur de l'intervalle de définition de "w". "w" s'étend de 0 à w_e . La valeur w_e a une validité locale dans le bloc programmé. La valeur utilisée par défaut est $w_e=1$.

On suppose que la dérivée est nulle au point de départ de la séquence de la spline (c.-à-d. dans le premier bloc de déplacement après G6).

Syntaxe :

{ PL< w_e > }	Programmation optionnelle de la longueur des paramètres de la spline
avec	
< w_e > :	Valeur quelconque > 0

Tab. 5-19: Programmation de la longueur des paramètres de la spline

Exemple : G6 X(0.1,1.25) B(0.0,-1.0) PL0.6

(w s'étend de 0 à 0.6)

Splines cubiques C1 et C2 constantes (types de splines 1 et 2)

Programmation de coordonnées

Description : Les points d'arrivée des coordonnées du canal sont programmés ici.

Tous les <membres> énumérés dans "SplineDef" se déplacent sur la courbe spline ; les coordonnées restantes qui ne sont pas énumérées dans "Spline-Def" se déplacent linéairement.

Syntaxe :

<NomCoord><PosFin>	Programmation des différentes coordonnées et de leurs valeurs.
avec	

Fonctions CN avec syntaxe conforme DIN 66025 (extensions incl.)

<NomCoord>	Nom de la coordonnée (y compris l'orientation O)
<PosFin>	Position finale de la coordonnée

Tab. 5-20: Programmation du point d'arrivée des coordonnées du canal

Exemple : Coordonnées x, y, z et coordonnées d'orientation phi, theta.

```
SplineDef(2203,x,y,z,phi,theta)
```

```
G6 x10 y20 phi20 theta30
```

```
x15 y22 O(1,1,2)
```

Exemple :

Axes X, Y, U

```
SplineDef(1213,X,Y)
```

```
G6 X10 Y10 U20 (X et Y se déplacent sous forme de spline, U linéairement)
```

Longueur des paramètres de la spline

Description : La commande numérique calcule la longueur des paramètres de la spline à partir des repères affectés. On utilise ici le déplacement affecté dans l'ID spline (paramétrage). Au besoin, la longueur des paramètres de la spline peut également être programmée :

Syntaxe :	{ PL<w _e >}	Programmation optionnelle de la longueur des paramètres de la spline quand la sélection du <paramétrage> doit être écrasée (voir chap. 6.130 "Définition du type de spline SplineDef, SPD" à la page 415, paramètre <id>).
	avec	
	<w _e >	Valeur quelconque > 0

Tab. 5-21: Programmation optionnelle de la longueur des paramètres de la spline

Conditions aux limites de départ et d'arrivée

Voir [chap. "Conditions aux limites de départ et d'arrivée"](#) à la page 138.

Splines B (NURBS) (type de splines 3)**Programmation de coordonnées**

Description : Les points d'arrivée des coordonnées du canal (points de contrôle) sont programmés ici. Tous les <membres> énumérés dans "SplineDef" se déplacent sur la courbe spline ; les coordonnées restantes qui ne sont pas énumérées dans "SplineDef" se déplacent linéairement.

Syntaxe :	<NomCoord><PosFin>	Programmation des différents points de contrôle (coordonnées) et de leurs valeurs.
	avec	
	<NomCoord>	Nom de la coordonnée (y compris l'orientation O)
	<PosFin>	Position finale de la coordonnée

Tab. 5-22: Programmation des points de contrôle (coordonnées/axes)

Exemples : Coordonnées x, y, z et coordonnées d'orientation

```
SplineDef(3103,x,y,z,O)
```

```
G6 x10 y20 z30 O(0.1,0,1.0)
```

Fonctions CN avec syntaxe conforme DIN 66025 (extensions incl.)

Axes X, Y, U

SplineDef (3102, X, Y) G6 X10 Y10 (X et Y se déplacent sous forme de spline, U linéairement)

Longueur des paramètres de la spline

Description : La commande numérique calcule automatiquement en interne la longueur des paramètres de la spline à partir des points de contrôle affectés. Le déplacement affecté dans l'ID spline (paramétrage = 1, 2 ou 3) est utilisé ici. Au besoin, la longueur des paramètres de la spline peut également être programmée :

Syntaxe :	{ PL<w _e >}	Programmation optionnelle de la longueur des paramètres de la spline quand la sélection du <paramétrage> doit être écrasée (voir chap. 6.130 "Définition du type de spline SplineDef, SPD" à la page 415, paramètre <id>).
	avec	
	<w _e >	Valeur quelconque > 0

Tab. 5-23: Programmation optionnelle de la longueur des paramètres de la spline

Poids ponctuel de la spline pour point de contrôle des splines B

Syntaxe :	{ PW<w>}	Programmation optionnelle de poids ponctuels. Cela permet de modifier la spline à proximité d'un point de contrôle.
	avec	
	<w>	Valeur par défaut : 1 0 < w < 1 : écarte la spline du point de contrôle. w > 1 : rapproche la spline du point de contrôle.

Tab. 5-24: Programmation optionnelle de poids ponctuels.

Exemple : Coordonnées x, y, z et coordonnées d'orientation

SplineDef (3103, x, y, z, O)
G6 x10 y20 z30 O(0.1, 0, 1.0) PW2.3

Approximation de spline B (type de spline 4)

Programmation de coordonnées

Description : Les points d'arrivée des coordonnées du canal sont programmés ici. Seuls les <membres> peuvent être programmés.

Syntaxe :	<NomCoord><PosFin>	Programmation de points des coordonnées
	avec	
	<NomCoord>	Nom de la coordonnée (y compris l'orientation O).
	<PosFin>	Position finale de la coordonnée

Tab. 5-25: Programmation de points

Exemple : L'exemple de programme CN suivant présente le <Q_i> respectif d'un point de l'espace programmé, y compris le vecteur d'orientation "x.. y.. z.. O(..)".

Fonctions CN avec syntaxe conforme DIN 66025 (extensions incl.)

N00 SplineDef(4213)	; définit le G06 ultérieur comme approximation de spline B
N01 BAP(E0.01, OE0.05)	; cote de tolérance égale à 0,01 mm, cote de tolérance du vecteur d'orientation égale à 0,05 mm
..	
N100 G0 <Q ₀ >	; point de départ de la courbe spline B
N101 G06 <Q ₁ >	
N102 <Q ₂ >	
N103 <Q ₃ >	
..	
N400 <Q ₃₀₀ >	; dernier point de la courbe spline B
G1	

Valeur de tolérance de l'approximation

Voir [chap. "Paramètre d'approximation de spline B"](#) à la page 139.

Conditions aux limites de départ et d'arrivée

Voir [chap. "Conditions aux limites de départ et d'arrivée"](#) à la page 138.

Longueur des paramètres de la spline

La commande numérique définit automatiquement la longueur des paramètres de la spline qui ne peut pas être programmée.

Conditions aux limites de départ et d'arrivée

Il est possible de définir les conditions aux limites de départ et/ou d'arrivée dans des blocs CN au sein de la séquence spline pour tous les types de splines, exception faite du type 0 (programmation de coefficients).

SBC(<type BC>{,<valeurs>})	Condition aux limites pour le point de départ d'un bloc spline de la séquence spline
EBC(<type BC>{,<valeurs>})	Condition aux limites pour le point de départ d'un bloc spline de la séquence spline
avec	

Fonctions CN avec syntaxe conforme DIN 66025 (extensions incl.)

<type BC>	0 (pour le type de spline 4) 1 (pour les types de splines 1, 2, 4) Indication de la direction de la tangente au point de départ ou d'arrivée de la séquence spline. Une valeur doit être indiquée dans la liste <valeurs> pour chaque membre spline (3 valeurs pour le vecteur d'orientation). 10 (pour les types de splines 1, 2, 3, 4) Les tangentes de départ et d'arrivée se dirigent vers la ligne qui joint le point de départ au point d'arrivée du bloc spline actuel. 11 (pour les types de splines 1, 2, 4) Avec SBC : la tangente au point de départ du bloc spline actuel se dirige vers la ligne qui joint le point de départ au point d'arrivée du bloc précédent. 2 (pour les types de splines 1, 2) Indication de la seconde dérivée au point de départ ou d'arrivée du bloc spline après le paramètre spline "w". Une valeur doit être indiquée dans la liste <valeurs> pour chaque membre spline (3 valeurs pour le vecteur d'orientation). 3 (uniquement pour le type de spline 2) La condition aux limites De Boor relie la deuxième dérivée à la première ou aux deux derniers repères. Ici, <valeurs> est généralement paramétré sur 1. Cette liste ne peut être programmée qu'au début ou à la fin de la séquence spline. 4 (uniquement pour le type de spline 2) Condition aux limites périodique : le dernier et le premier points de la séquence spline coïncident. SBC(4) requiert impérativement EBC(4) et vice versa. Toute la séquence spline doit se trouver dans la zone de prévisualisation, sans quoi un message d'erreur d'exécution est généré.
<valeurs>	Uniquement en combinaison avec les types BC = 1, 2 et 3 Valeurs par défaut : 0,...,0 pour les types BC = 1, 2 et 1,...,1 pour le type BC = 3 Aucune valeur pour les types BC 10, 11 et 4

Tab. 5-26: Conditions aux limites de départ et d'arrivée

Exemples : SBC(1, 1.0, 1.0, 0.2) pour SplineDef(1213,X,Y,B)

SBC(1, 1.0, 1.0, 0.2, 0.0, 0.5, 0.5) pour Spline-Def(2203,x,y,z,O)

EBC(2) pour SplineDef(2203,x,y,z,O)

EBC(10) pour SplineDef(3203,x,y,z,O)

Paramètre d'approximation de spline B

Syntaxe :

BsaPar(E...,OE...,CMP)	
Format abrégé : BAP(..)	
E (Error)	Si E est manquant, la valeur appliquée est celle enregistrée dans MP 80070030. Indique la cote de tolérance en mm ou en pouces.

Fonctions CN avec syntaxe conforme DIN 66025 (extensions incl.)

OE (OrientationError)	Cote de tolérance du vecteur d'orientation OE n'a d'importance que si l'orientation vectorielle est active. Si OE est manquant, la valeur appliquée est la valeur d'activation OE=E.
CMP (Compression)	Avec (CMP1) ou sans (CMP0) compression En l'absence de CMP, CMP1 s'applique.

Tab. 5-27: Syntaxe Paramètre d'approximation de spline B

5.2.7 Pente de la trajectoire ACTIVÉE "G8", pente de la trajectoire DÉSACTIVÉE "G9"

- Description :**
- Sans "pente de trajectoire", la commande exécute une pente ascendante (upslope) et descendante (downslope) complète (rampe de vitesse) à $v = 0$ au début et à la fin d'un bloc de déplacement.
- Cela réduit l'écart de contour lors de l'enchaînement des blocs, mais nécessite aussi un temps d'usinage plus long.
- Avec la fonction "Pente de la trajectoire", la commande tente de générer une vitesse aussi constante que possible, de l'ordre de l'avance programmée, même lors de l'enchaînement de blocs. Ceci réduit le temps d'usinage.

Le fait que le contour soit éventuellement "adouci" au niveau des angles peut être avantageux pour certaines opérations d'usinage (surface plus uniforme).

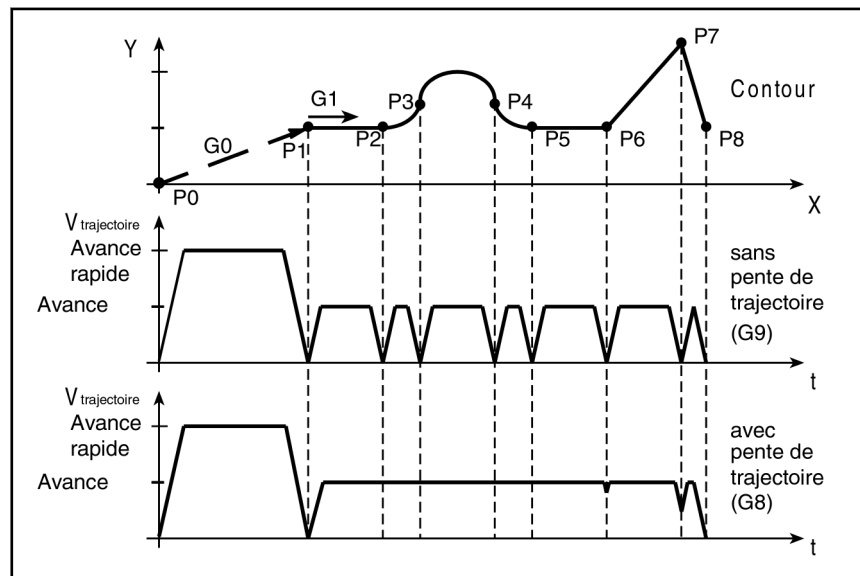


Fig. 5-11: Pente de trajectoire

Lors de la détermination du profil de vitesse optimal, la commande tient compte non seulement de la valeur de la capacité de saut de l'axe (MP 1010 00012), mais aussi du nombre de blocs-programmes suivants (prévisualisation de bloc ; MP 7060 00110 à 7060 00130), car un freinage doit être assuré à tout moment au sein de la marge dynamique de la machine.

Pour limiter les malfaçons de contour à l'endroit des angles réels, la capacité de saut de l'axe ne doit pas être trop importante. Au contraire, une capacité de saut de l'axe trop faible entraîne un ralentissement in-

Fonctions CN avec syntaxe conforme DIN 66025 (extensions incl.)

opportun à l'endroit des petits coudes de contours (raccords quasi-continus).

Un lissage de l'accélération sur trajectoire peut être saisi pour G8 en tant que réglage par défaut dans le paramètre machine 7050 00320 (voir aussi [chap. 6.60 "Activation de la limitation des secousses JerkControl, JKC" à la page 312](#)).

Syntaxe :

G8	Pente de la trajectoire ACTIVÉE
G9	Pente de la trajectoire DÉSACTIVÉE

Tab. 5-28: Syntaxe G8, G9

Particularités et restrictions :

- Si l'arrêt précis est activé, la vitesse est réduite à $v = 0$ après chaque bloc, bien que la fonction G8 soit active.
- Les fonctions auxiliaires peuvent réduire l'effet de G8 si leur temps d'exécution, validation comprise, n'est pas nettement inférieur au temps d'interpolation d'un bloc. Le cas échéant, la course de déplacement d'un bloc doit être prolongée ou son avance réduite.
- La pente de la trajectoire n'affecte que les axes synchrones (axes d'usinage).
- Les fonctions G8 et G9 constituent un groupe de fonctions modales avec les fonctions rampe et se révoquent réciproquement.

5.2.8 Profil de vitesse avec limitation des secousses "G8(SHAPE...)", "G9(SHAPE...)", "G9(ASHAPE...)", "G9(X... , Y... , ...)", "AsynchrShapeOrder, ASO"

Description générale



La fonction SHAPE a été remplacée par la limitation de secousses spécifique aux axes et son utilisation devrait être évitée autant que possible (cf. "Description des fonctions IndraMotion MTX", "Bases de la limitation des secousses spécifique aux axes"). Il est recommandé de laisser la commande calculer automatiquement la fonction SHAPE en interne par l'intermédiaire de la limitation des secousses spécifique aux axes (c.-à-d. avec JerkControl(1)). Quand la limitation des secousses est désactivée (c.-à-d. avec JerkControl(0)), il est possible, en alternative, de programmer soi-même la fonction SHAPE. L'utilisation simultanée de la fonction SHAPE et de la limitation des secousses est impossible.

Pour les axes asynchrones, toutes les données SHAPE sont ignorées lorsque $AX/Dr[i]/Jerk/MaxAxJerk <> 0$ a été défini.

Le profil de vitesse avec limitation des secousses lisse les sauts produits sur plusieurs cycles d'interpolation au fur et à mesure de l'accélération sur la trajectoire. Le nombre des cycles d'interpolation peut être programmé.

Il est ainsi possible d'obtenir des transitions de vitesse douces (limitation des secousses).

Le profil de vitesse avec limitation des secousses est possible aux types de fonctionnement suivants :

- "Mode trajectoire" (avec G8 actif)
- "Mode de positionnement" (avec G9 actif)
- "Mode asynchrone"

Fonctions CN avec syntaxe conforme DIN 66025 (extensions incl.)

Le profil de vitesse avec limitation des secousses est également appelé Fonction SHAPE.

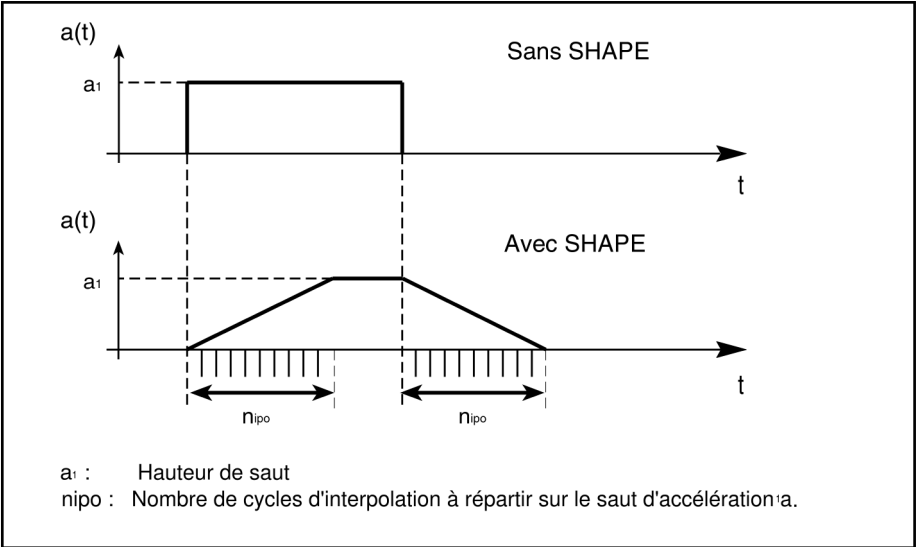


Fig. 5-12: Répartition de l'accélération avec/sans SHAPE

SHAPE pour axes asynchrones

Syntaxe :

AsynchShapeOrder <valeur>	
Format abrégé : ASO <valeur>	
avec	
<valeur>	Nombre de cycles d'interpolation sur lesquels la commande doit répartir linéairement un saut d'accélération. Plage de valeurs : 2 à 100 cycles ; nombre entier.

Tab. 5-29: Syntaxe AsynchShapeOrder

Exemples : N10 W150 FA20000 ASO(100)

La secousse de l'axe est obtenue avec :

$$\frac{MP101000001}{100 * MP903000001 * 10^{-6}} \left[\frac{m}{s^3} \right]$$

Si l'ordre SHAPE n'est pas indiqué dans le bloc, alors la valeur utilisée par défaut est celle du paramètre 1003 00008 "Ordre SHAPE pour l'axe".

N10 W150 FA1000 AxAcc(W0.5)

La secousse de l'axe est obtenue avec :

$$\frac{0.5}{MP100300008 * MP903000001 * 10^{-6}} \left[\frac{m}{s^3} \right]$$

Le paramètre 1003 00007 "Ordre SHAPE minimum pour limitation de jerk de trajectoire" doit impérativement être respecté. Si aucun ordre SHAPE n'est programmé et que les paramètres de préréglage et l'ordre SHAPE minimal sont affectés à 0 ou 1, alors la commande déplace l'axe sans limitation de secousses. L'indication de l'ordre SHAPE pour les axes asynchrones est efficace au sein du bloc et s'applique à tous les axes asynchrones programmés.

Fonctions CN avec syntaxe conforme DIN 66025 (extensions incl.)

SHAPE pour mode trajectoire – Programmation

Syntaxe :

G8 { (SHAPE<valeur>) }	
avec	
<valeur>	0 ou 1 : Désactiver SHAPE pour le mode trajectoire. 0 ou 1 : 2 à 100 (nombres entiers) : Nombre de cycles d'interpolation sur lesquels la commande doit répartir linéairement le saut (ordre SHAPE) . Sans paramètre SHAPE : Déplacement avec la valeur par défaut enregistrée pour SHAPE dans le paramètre machine 7050 00320.

Tab. 5-30: Syntaxe G8(SHAPE)



Une modification de l'ordre SHAPE entraîne toujours une vitesse d'enchaînement des blocs $v = 0$.

Exemple :

N30 G8 (SHAPE10)	Activer SHAPE au mode trajectoire et répartir les sauts d'accélération sur 10 cycles IPO
:	
:	
N50 G8 (SHAPE0)	Désactiver SHAPE au mode trajectoire
:	
:	
N70 G8 (SHAPE1)	Désactiver SHAPE au mode trajectoire
:	
:	
N90 G8	Déplacement avec la valeur par défaut enregistrée dans le paramètre machine 7050 00320 pour SHAPE

SHAPE pour mode de positionnement – Programmation

Syntaxe :

G9{(SHAPE<valeur>)}	Activer SHAPE linéaire pour le mode de positionnement.
avec	
<valeur>	Nombre de cycles d'interpolation sur lesquels la commande doit répartir linéairement un saut d'accélération. Plage de valeurs : 2 à 100 cycles ; nombre entier.
Sans paramètres SHAPE :	Déplacement avec la valeur par défaut enregistrée pour SHAPE dans le paramètre machine 7050 00340.
G9(SIN<valeur>)	Activer SHAPE de forme $\sin^2$ au mode de positionnement
avec	

Fonctions CN avec syntaxe conforme DIN 66025 (extensions incl.)

<valeur>	Nombre de cycles d'interpolation sur lesquels la commande doit répartir un saut d'accélération sous la forme sin² . Les valeurs suivantes sont admissibles : 5 : 5 cycles IPO 10 : 10 cycles IPO 15 : 15 cycles IPO 20 : 20 cycles IPO 40 : 40 cycles IPO
G9 (ASHAPE { , TYP<i> })	
	Attribue à chaque axe du canal l'ordre SHAPE qui lui est spécifique (nombre de cycles IPO) à partir de MP 1003 00008 et calcule bloc par bloc l'ordre SHAPE résultant spécifique au bloc respectif pour la trajectoire.
G9 (<Par ₁ >{,<Par ₂ >,...} {,TYP<i>})	
	Programmer l'ordre SHAPE axe par axe. Pour les axes non programmés, l'ordre SHAPE appliqué est celui indiqué dans MP 1003 00008.
avec	
<Par _x >	Nom d'axe logique avec l'ordre SHAPE devant être attribué à cet axe (maximum 100).
Exemple : N30 G9 (X4, Y6, Z10)	Ordre SHAPE (axe X) = 4 Ordre SHAPE (axe Y) = 6 Ordre SHAPE (axe Z) = 10
TYPE 0 ou aucun TYPE	Tout ordre SHAPE d'axe est évalué d'après le rapport entre l'accélération réelle de l'axe et l'accélération maximale de l'axe. L'ordre SHAPE le plus élevé ainsi déterminé devient l'ordre SHAPE de la trajectoire (voir ci-dessous).
TYPE 1	L'ordre SHAPE le plus élevé d'un axe devient l'ordre SHAPE de la trajectoire.

Tab. 5-31: Syntaxe G9(SHAPE) et G9(ASHAPE)

Ordre SHAPE résultant

Description : L'ordre SHAPE résultant de la trajectoire S_b correspond au maximum des ordres SHAPE effectifs S_i^{eff} de tous les axes participant à l'interpolation :

$S_b = \max \{S_1^{eff}, \dots, S_n^{eff}\}$

Pour G9(...,TYPE0) ou pour les cas où TYPE n'est pas indiqué, les ordres SHAPE effectifs des axes S_i^{eff} sont calculés à partir des ordres SHAPE programmés selon la formule :

$$S_i^{eff} = S_i^p \frac{\alpha_i^{eff}}{\alpha_i^{max}}$$

Les points suivants s'appliquent pour G9(...,TYPE1) :

$S_i^{eff} = S_i^p$

Fonctions CN avec syntaxe conforme DIN 66025 (extensions incl.)

Ici :

S_i^p	Ordre SHAPE des axes programmé avec G9 (...)
a_i^{eff}	Accélération effective de l'axe au sein du bloc CN courant En cas d'interpolation linéaire, cette valeur est fonction de la part actuelle de l'axe dans la trajectoire. Avec les autres types d'interpolations (par ex. circulaire, hélicoïdal), elle correspond à l'accélération de l'axe normalement programmée avec "AxAcc". Quant aux fonctions "Plan incliné" ou "Couplage d'axe", l'accélération effective de l'axe est généralement davantage réduite par rapport à la valeur définie dans "AxAcc" !
a_i^{max}	Accélération maximale de l'axe (paramètre AX/Dr/Acc/FeedAxAcc (MP 1010 00001))



"AxAcc" ne modifie pas l'accélération maximale des axes à partir du paramètre AX/Dr/Acc/FeedAxAcc (MP 1010 00001) !

Rapport entre l'ordre SHAPE et les secousses

Description : Avec les ordres SHAPE des axes S_i^{eff} , une secousse maximale r_i^{max} (dérivée de l'accélération en fonction du temps) est définie pour chaque axe, cette valeur n'étant pas dépassée quel que soit le mouvement.

La secousse est définie par :

$$r_i^{max} = \frac{a_i^{max}}{S_i^{eff} T_{ipo}}$$

T_{ipo} : Temps de cycle de l'interpolateur

Exemple : L'axe X présente une accélération maximale (paramètre AX/Dr/Acc/FeedAxAcc (MP 1010 00001)) de 10 m/s². L'ordre SHAPE programmé des axes est de 5 ms, le cycle d'interpolation de 4 ms.

Suivant la formule ci-dessus, une secousse maximale définie pour l'axe X est alors de 500 m/s³.

5.2.9 Aucun plan "G16"

Description : Le cas échéant, désactive un plan actif.

Les points suivants s'appliquent ici :

- Si un axe principal ou secondaire est supprimé d'un canal (par ex. en lien avec la fonctionnalité "Transfert d'axe"), la commande désactive automatiquement le plan sélectionné et active la fonction G16.

Toute interpolation circulaire/hélicoïdale devient impossible dans ce canal tant qu'aucun plan valide n'a été sélectionné.

- Certaines applications, certains types de machines et certaines unités de traitement ne nécessitent aucune configuration des plans quand une interpolation circulaire ou hélicoïdale est requise (par ex. dans le cas de canaux comptant un seul axe d'usinage).

Il est alors possible de saisir "0" (aucun rôle technique) pour chaque axe dans le paramètre CHAN/Ch/Coord/Acs/ChAxClass (MP 7010 00030) (signification des coordonnées).

Fonctions CN avec syntaxe conforme DIN 66025 (extensions incl.)

- Si aucune fonction de plan (G17, G18, G19, G20) n'est saisie alors qu'un canal est actif (paramètre CHAN/Ch/Ini/ChResetState (MP 7060 00020) et paramètre CHAN/Ch/Ini/NcResetState (MP 7060 00010)), alors la fonction G16 est automatiquement activée pour le canal correspondant.
- Les fonctions G16, G17, G18, G19 et G20 constituent un groupe de fonctions modales et se révoquent donc réciproquement.

Syntaxe :

G16

Tab. 5-32: Syntaxe G16

Particularités et restrictions : Si la correction de trajectoire de la fraise (G41/G42) est active, le plan actif ne doit pas être changé ni désactivé.

5.2.10 Changement de plan"G17, G18, G19"

Description : Le plan actif est sélectionné à l'aide de la fonction Changement de plan. Elle est définie par une coordonnée principale et une coordonnée secondaire ainsi que, le cas échéant, par un axe d'approche qui doit être perpendiculaire à ce plan. À cet effet, G17, G18 et G19 sont programmés sans paramètres.

Un système de coordonnées cartésiennes de la pièce à usiner (WCS) est sous-tendu par les axes X, Y et Z pour une **machine-outil cartésienne type**. Au total, il est possible de définir 3 plans de base différents avec 2 axes de coordonnées du système de coordonnées de la pièce à usiner.

Le plan actif est sélectionné à l'aide de la fonction Changement de plan. Elle est définie par une coordonnée principale et une coordonnée secondaire ainsi que, le cas échéant, par un axe d'approche qui doit être perpendiculaire à ce plan. À cet effet, G17, G18 et G19 sont programmés sans paramètres.

Ces coordonnées, de même que le plan actif, sont définies comme suit par G17, G18 et G19 :

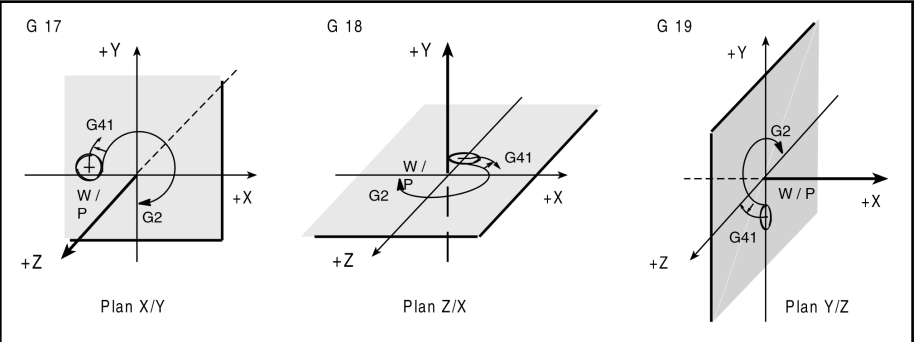


Fig. 5-13: Changement de plan

	Coordonnée principale	Coordonnée se- condaire	Coordonnée d'approche	Plan actif
G17	X	Y	Z	XY
G18	Z	X	Y	ZX
G19	Y	Z	X	YZ

Tab. 5-33: Aperçu des coordonnées G17, G18 et G19

Fonctions CN avec syntaxe conforme DIN 66025 (extensions incl.)

Syntaxe :

G17
G18
G19

Tab. 5-34: Syntaxe G17, G18, G19

Particularités et restrictions :

- Si la correction de trajectoire de la fraise (G41/G42) est active, le plan actif ne doit pas être changé ni désactivé.
- Les points suivants s'appliquent aux paramètres d'interpolation I, J et K pour G2/G3 et G33 :
 - "I" se rapporte à l'axe X.
 - "J" se rapporte à l'axe Y.
 - "K" se rapporte à l'axe Z.
- La correction de la longueur d'outil est attribuée au plan actif à l'aide de G47(ActPlane) de la manière suivante :
 - L1 à la coordonnée principale
 - L2 à la coordonnée secondaire
 - L3 à la coordonnée d'approche
 Cette attribution est mise à jour après le changement de plan actif.
- Les fonctions G16, G17, G18, G19 et G20 constituent un groupe de fonctions modales et se révoquent donc réciproquement.

Attribution indépendante des noms d'axe :

Les noms d'axes peuvent être paramétrés librement dans la commande IndraMotion MTX. Les axes devant sous-tendre le système de coordonnées de la pièce à usiner (WCS) sont donc choisis à l'aide du paramètre machine 7010 00030 (signification des coordonnées). Celui-ci définit, **indépendamment** des noms d'axes configurés, quels axes sous-tendent respectivement les coordonnées X, Y et Z du système de coordonnées de la pièce à usiner :

Axe avec "Signification d'axe X" : coordonnée X du WCS

Axe avec "Signification d'axe Y" : coordonnée Y du WCS

Axe avec "Signification d'axe Z" : coordonnée Z du WCS

En règle générale, on obtient ainsi l'attribution suivante pour G17, G18 et G19 :

	Coordonnée principale	Coordonnée secondaire	Coordonnée d'approche	Plan actif
G17	Coord. X WCS	Coord. Y WCS	Coord. Z WCS	Coord. XY WCS
G18	Coord. Z WCS	Coord. X WCS	Coord. Y WCS	Coord. ZX WCS
G19	Coord. Y WCS	Coord. Z WCS	Coord. X WCS	Coord. YZ WCS

Tab. 5-35: Aperçu de l'attribution des noms d'axe

5.2.11 Changement de plan avancé "G17(...), G18(...), G19(...)"

Description :

Lors du changement de plan avancé, outre la sélection du plan actif, les axes devant actuellement sous-tendre le système de coordonnées de base de la pièce à usiner (WCS) sont également définis.

Il est ainsi possible de définir, en cours d'exécution, un système de coordonnées de base de la pièce à usiner qui soit différent du réglage des paramètres machine, et de sélectionner le plan actuel sur la base de celui-ci.

Fonctions CN avec syntaxe conforme DIN 66025 (extensions incl.)

Le plan est sélectionné selon l'aperçu suivant :

	Coordonnée principale	Coordonnée secondaire	Coordonnée d'approche	Plan actif
G17	Coord. X WCS	Coord. Y WCS	Coord. Z WCS	Coord. XY WCS
G18	Coord. Z WCS	Coord. X WCS	Coord. Y WCS	Coord. ZX WCS
G19	Coord. Y WCS	Coord. Z WCS	Coord. X WCS	Coord. YZ WCS

Tab. 5-36: Aperçu de l'attribution des noms d'axes

Syntaxe :

G17 (<axe1>,<axe2>,<axe3>)	
G18 (<axe1>,<axe2>,<axe3>)	
G19 (<axe1>,<axe2>,<axe3>)	
avec	
<axe1>	"Coordonnée X" du système de coordonnées de la pièce à usiner WCS
<axe2>	"Coordonnée Y" du système de coordonnées de la pièce à usiner WCS
<axe3>	"Coordonnée Z" du système de coordonnées de la pièce à usiner WCS

Tab. 5-37: Syntaxe G17 à G19

Particularités et restrictions :

- Les axes peuvent être programmés avec une désignation d'axe logique (spécifique au canal) ou physique (spécifique au système) et doivent être différents les uns des autres. La saisie peut s'effectuer soit directement, soit en tant que variable chaîne CPL.
- Les axes programmés doivent être affectés au canal actuel.
- Lors de la programmation, il suffit d'indiquer les axes devant être reconfigurés en tant que paramètres. Pour tous les autres axes, les coordonnées correspondantes du WCS restent inchangées.
Exemple : G17 (, , W)
La coordonnée Z du WCS est définie au moyen de l'axe W.
Les coordonnées X et Y du WCS restent inchangées.
Le plan actif est sous-tendu par les coordonnées X et Y du WCS.
- La programmation d'une expression vide entre parenthèses rétablit le WCS configuré dans MP 7010 00030 avant d'activer le plan sélectionné.
Exemple : G17 ()
- La fonction ne doit pas être programmée, si la fonction Correction de trajectoire de la fraise (G41/G42) est active.
Si la correction de trajectoire de la fraise est active, il ne faut généralement pas changer ni désactiver le plan.
- Si la correction d'axes est active, aucun changement du plan avancé ne peut être programmé, car le WCS est sous-tendu par les coordonnées tridimensionnelles.
- Les fonctions G16, G17, G18, G19 et G20 constituent un groupe de fonctions modales et se révoquent donc réciproquement.

5.2.12 Libre sélection du plan (indépendamment du WCS) "G20"

Description : La libre sélection du plan est nécessaire aux applications pour lesquelles le niveau actif doit être découplé du système de coordonnées de la pièce à usiner (WCS) actuel. Elle permet de sous-tendre le système avec des axes quelconques, que ces derniers sous-tendent actuellement aussi le WCS ou non.

Il est ainsi possible, entre autres, de calculer les corrections d'outil (G41, G42, G47) dans les coordonnées principale, secondaire et d'approche du plan actuel, tandis que les transformations de la pièce à usiner (placements, corrections du plan sous-tendu) se rapportent aussi à un système de coordonnées de la pièce à usiner sous-tendu par d'autres axes.

Syntaxe :

G20 (<axe1>,<axe2>{,<axe3>})	
avec	
<axe1>	Coordonnée principale du plan à sous-tendre Elle est attribuée au paramètre d'interpolation "I".
<axe2>	Coordonnée secondaire du plan à sous-tendre Elle est attribuée au paramètre d'interpolation "J".
<axe3>	Coordonnée d'approche du plan à sous-tendre Elle peut être programmée si une coordonnée d'approche doit absolument être prévue pour la correction d'outil "G47(ActPlane)".

Tab. 5-38: Syntaxe G20

- Particularités et restrictions :**
- Les axes peuvent être programmés avec une désignation d'axe logique (spécifique au canal) ou physique (spécifique au système).
 - Une correction de trajectoire de la fraise G41/G42 se rapporte au plan sélectionné. C'est pourquoi le plan actif ne peut être changé que si la correction de trajectoire de la fraise (G40) est désactivée. Sinon, un message d'erreur est émis.
 - La correction de la longueur d'outil est attribuée au plan actif à l'aide de "G47(ActPlane)" de la manière suivante :
 L1 à la coordonnée principale
 L2 à la coordonnée secondaire
 L3 à la coordonnée d'approche
 Cette attribution est mise à jour après le changement de plan actif.
 - Les fonctions G16, G17, G18, G19 et G20 constituent un groupe de fonctions modales et se révoquent donc réciproquement.

5.2.13 Programmation du tracé de contour "G31"

Description : La programmation du tracé de contour constitue un type particulier de définition du contour qui permet de reprendre largement les cotes du dessin directement en tant que paramètres CN. Le contour peut alors se composer de segments de cercle et de droite.

La programmation du tracé de contour prend notamment en charge :

- la programmation de droites
- le tracés à 2, 3 et 4 points
- les chanfreins de raccord

Fonctions CN avec syntaxe conforme DIN 66025 (extensions incl.)

- la programmation de cercles avec des possibilités de programmation étendues par rapport à la programmation standard
- la définition automatique d'un point d'intersection en cas de segments de contour définis de manière incomplète pour les raccords droite-droite, cercle-cercle, droite-cercle ou cercle-droite
- les raccords de droite tangentiels sur les cercles ainsi que les cercles de connexion entre 2 arcs de cercle

Des chanfreins et arrondis de raccords peuvent être programmés en option à l'aide des fonctions CN CHSL, CHLL et RNDL.

Les contours G31 se rapportent toujours au plan de travail actif (G17, G18, G19).

Paramètres de fonction

X, Y, Z, ...	Noms des coordonnées (comme configuré)
A	Angle (syntaxe similaire à l'angle polaire 1 de la programmation de coordonnées polaires, configurable via les paramètres machine)
I, J, K	Paramètres d'interpolation pour le centre du cercle (avec G31, le centre du cercle est toujours indiqué de manière absolue)
R	Rayon du cercle
CW, CCW	Sens de rotation du cercle : sens antitrigonométrique, sens trigonométrique
TG	Raccord tangentiel ou connexion tangentielle
CS	Segment de chanfrein pour un chanfrein de raccord
CL	Longueur de chanfrein pour un chanfrein de raccord

Tab. 5-39: Paramètre de la fonction Programmation du tracé de contour G31

Syntaxe :

Variantes de la programmation de droites pour G31 :	
X	Droite avec point d'arrivée programmé (comme G1)
A <angle>	Droite avec programmation d'angle
A <angle> X ... ou Y ... ou Z ...	Droite avec angle et une coordonnée programmée (tracé à 2 points)
G31 A <angle> X ... Y ... ou Z... X... ou Y... Z ...	Droite avec indication d'angle et point d'arrivée entièrement programmé
A <angle> CL <longueur> X	Droite avec angle, longueur programmée et point d'arrivée programmé
A <angle> CS <section> X	Droite avec angle, segment de chanfrein programmé et point d'arrivée programmé
TG	Connexion tangentielle linéaire (par ex. entre 2 cercles)
TG X	Raccord tangentiel avec point d'arrivée programmé

Tab. 5-40: Variantes de la programmation de droites pour G31

Variantes de la programmation de cercles pour G31 :	
X I CW ou CCW	Cercle par défaut

Fonctions CN avec syntaxe conforme DIN 66025 (extensions incl.)

X I	Cercle sans indication du sens de rotation
X ... ou Y ... ou Z ... I CW ou CCW	Cercle avec indication de position incomplète
I CW ou CCW	Cercle avec point d'arrivée non défini
I R <rayon de cercle> CW ou CCW	Cercle non défini
R <rayon de cercle> TG	Cercle de connexion tangentiel
X R <rayon de cercle> TG CW ou CCW	Cercle avec entrée tangentielle

Tab. 5-41: Variantes de la programmation de cercles pour G31

Exemple :

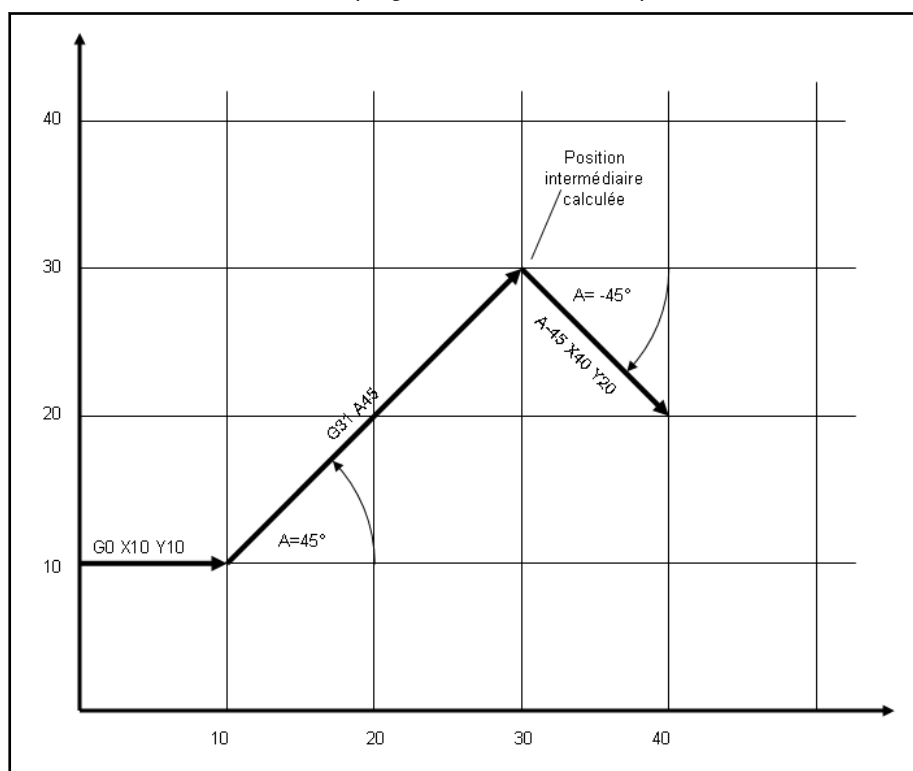


Fig. 5-14: Tracé à 3 points

Programme:

```
N10 G0 X0 Y10
N20 G0 X10 Y10
N30 G31 A45 F1000 ; tracé à 3 points
N40 A-45 X40 Y20 ; sur la position finale X40 Y20
M30
```

Particularités et restrictions :

- Les centres de cercles sont toujours programmés en coordonnées absolues.
- Il faut toujours programmer deux coordonnées pour le centre du cercle : la coordonnée d'axe principal et la coordonnée d'axe secondaire.
- Si des chanfreins et arrondis locaux sont programmés alors qu'un arrondi ou un chanfrein modal est activé, l'arrondi ou le chanfrein local masque l'arrondi ou le chanfrein modal.

Fonctions CN avec syntaxe conforme DIN 66025 (extensions incl.)

5.2.14 Filetage"G33"

- Description :

Le "filetage" constitue la découpe des filets suivants:

 - Filet longitudinal
(mouvement de coupe parallèle à l'axe principal du plan actif)
 - Filet plan
(mouvement de coupe parallèle à l'axe secondaire du plan actif)
 - Filet conique
(l'axe principal et l'axe secondaire du plan actif participent tous les deux au mouvement de coupe)

G33 est possible soit avec une broche asservie en vitesse de rotation, soit avec une broche asservie en position.

Le mouvement de coupe est toujours couplé à la broche principale active sur le canal correspondant (voir [chap. 6 "Fonctions CN avec syntaxe de langage standard" à la page 221](#)).

La vitesse d'avance du mouvement de coupe est la résultante de la vitesse de rotation actuelle de la broche et du pas en pourcentage programmé respectivement (constant, variable ; voir la syntaxe).

- Particularités :
- Possibilité de réaliser les filetages uniques ou multiples
 - Possibilité de programmer un pas de filetage constant ou variable
 - Possibilité de programmer un mouvement de remontée rapide
 - Possibilité de fabriquer des filets enchaînés

Le potentiomètre d'avance n'a aucun effet pendant G33.

Comme l'interpolation circulaire (G2, G3), le filetage est une fonction dépendante du plan actif (G17 à G20).

Syntaxe :

G33 <EP> <pas constant> {<pas var.>} {<angle de départ>}	
avec	
<EP>	Coordonnés de l'axe principal et de l'axe secondaire du plan actif. Le plan actif est déterminé par G17, G18, G19 ou G20. Exemple : Le plan actif pour G18 est normalement sous-tendu par les axes Z (axe principal) et X (axe secondaire).
<pas constant>	Détermine la course (en mm) qui sera parcourue par tour de broche en direction de l'axe principal ou de l'axe secondaire. La valeur est programmée avec le paramètre d'interpolation (I, J ou K) respectivement valide dans le plan actif. Concernant les filetages coniques, le pas de filetage saisi doit toujours se rapporter à la direction principale de coupe. Exemple : Pour G18, le paramètre K est affecté à l'axe principal et le paramètre I à l'axe secondaire. Dans le cas d'un filet longitudinal (pas dans le sens de l'axe principal), le pas de filetage constant est programmé avec l'adresse K.

Fonctions CN avec syntaxe conforme DIN 66025 (extensions incl.)

<pas variable>	Paramètre optionnel avec l'adresse DF Définit l'augmentation/la réduction du pas par tour de broche, en mm. Programmation : "DF<valeur>" avec <valeur> en mm.
<angle de départ>	Paramètres optionnels Si <angle de départ> n'est pas programmé, le système accepte un angle de 0 degré. L'angle de départ (décalage angulaire) est requis pour les filets multiples. L'adresse est représentée par le paramètre d'interpolation qui n'est pas associé au plan actif. Exemple : Pour G18, les adresses I et K sont affectées au plan. L'adresse de l'angle de départ est donc J.

Tab. 5-42: Syntaxe G33

Exemples : **Filet longitudinal**

G91 G18 G8 M3 S1000	Activer la programmation relative Activer le plan Z/X
G0 X-10	Mouvement d'avance de l'outil de coupe (1)
G33 Z-50 K2	Filetage (2) Point d'arrivée : incrément de -50 mm en direction Z Pas de filetage constant : 2 mm/tr Paramètres d'interpolation : ici K
G0 X10	Dégagement de l'outil de coupe (3)

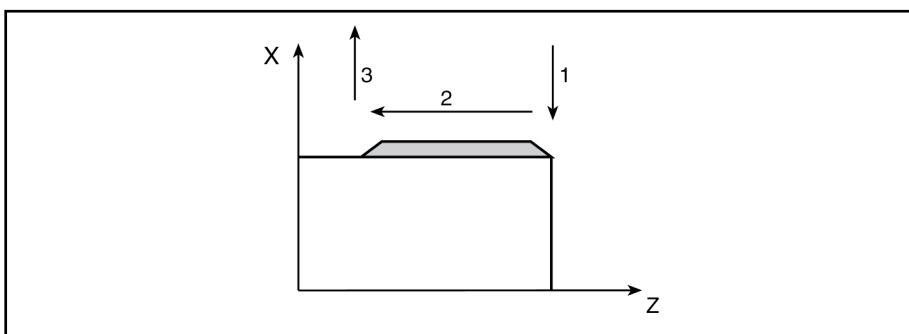


Fig. 5-15: Filet longitudinal

Filet plan

G91 G18 G8 M3 S1000	Activer la programmation relative Activer le plan Z/X
G0 Z-10	Mouvement d'avance de l'outil de coupe (1)
G33 X40 I2	Filetage (2) Point d'arrivée : incrément de +40 mm en direction X Pas de filetage constant : 2 mm/tr Paramètre d'interpolation : ici I
G0 Z10	Dégagement de l'outil de coupe (3)

Fonctions CN avec syntaxe conforme DIN 66025 (extensions incl.)

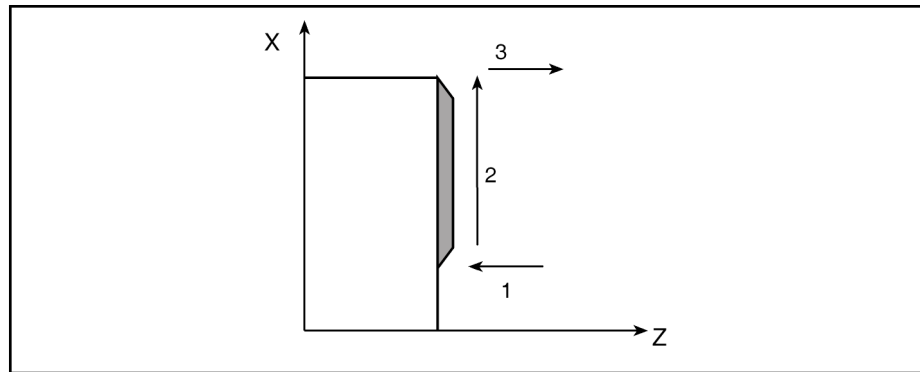


Fig. 5-16: Filet plan

Filet longitudinal conique

G91 G18 G8 M3 S1000	Activer la programmation relative Activer le plan Z/X
G0 X-20	Mouvement d'avance de l'outil de coupe (1)
G33 Z-50 X15 K2	Filetage (2) Point d'arrivée : incrément de -50 mm en direction Z et de +15 mm en direction X Pas de filetage constant : 2 mm/tr Paramètre d'interpolation : ici I
G0 X5	Dégagement de l'outil de coupe (3)

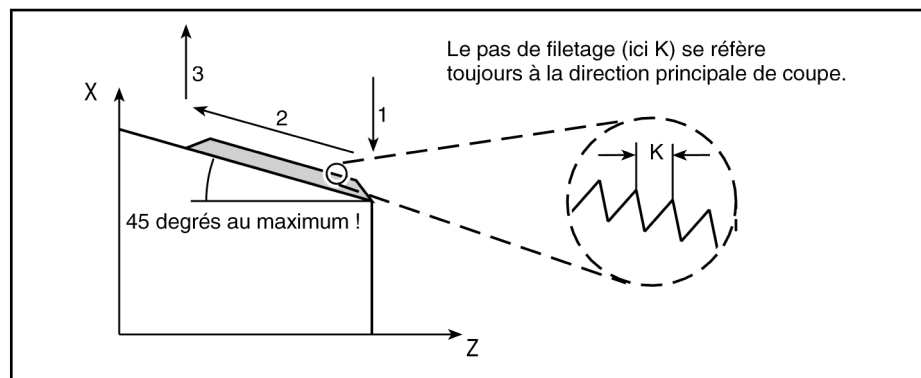


Fig. 5-17: Filet longitudinal conique

Filet enchaîné

- Ils peuvent être exécutés à partir de tous les modes de filetage mentionnés ci-dessus.
- Ils sont programmés par plusieurs blocs G33 consécutifs.

À chaque bloc G33 programmé, la commande numérique vérifie si le bloc suivant est un bloc G33 avec course. Si c'est le cas, le passage au bloc suivant doit s'effectuer sans arrêt de l'axe.

Filet multiple

Les filets multiples sont créés en décalant l'angle de départ (angle de départ, voir "Syntaxe :" à la page 152).

Fonctions CN avec syntaxe conforme DIN 66025 (extensions incl.)

Pour exécuter un filet quadruple, il faut générer quatre coupes décalées respectivement de 90 degrés (0, 90, 180, 270).

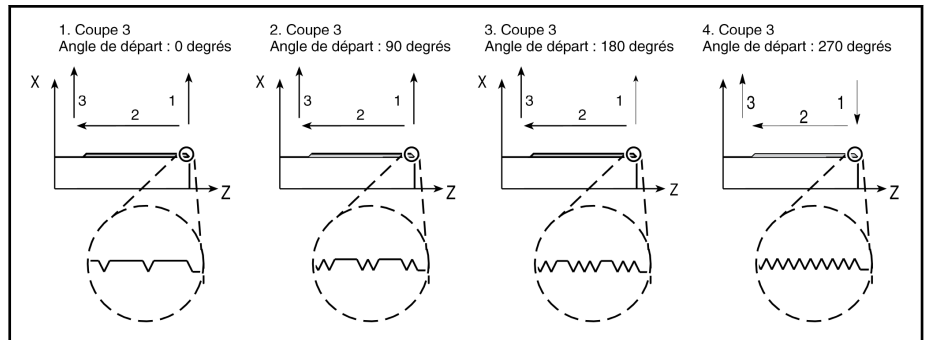


Fig. 5-18: Filet quadruple

Particularités

- Au début et à la fin de l'usinage d'un filet, les axes participants doivent être accélérés ou ralentis (jusqu'à l'arrêt).
 Par conséquent, il faut toujours prévoir un segment d'entrée (pour l'accélération des axes de coupe) et un segment de sortie (pour le ralentissement jusqu'à l'arrêt), ces segments devant être suffisamment dimensionnés.
- La fonction "ThreadSet (...)"
 (voir [chap. 6.70 "Changement de broche principale MainSp, MSP" à la page 323](#)) peut être utilisée en combinaison avec G33.
 - Adaptation de la dynamique du mouvement de coupe.
 - Configuration d'un mouvement de remontée pouvant être déclenché par le signal d'interface.
 - Commutation du mode de fonctionnement de la broche (asservissement en vitesse de rotation, régulation de position).
 - Définition d'un signal au niveau de l'interface canal (configurable au sein des signaux "Fonction active" via les paramètres machine).

5.2.15 Correction de trajectoire de la fraise "G40, G41, G42"

Description : La correction de trajectoire de la fraise guide l'outil sur une ligne équidistante parallèle à la trajectoire programmée lors de l'exécution d'un programme pièce (équidistante = trajectoire à distance constante perpendiculaire au contour programmé). L'écart entre la ligne équidistante et la trajectoire programmée est fonction de la valeur de correction du rayon active.

Le schéma suivant illustre le principe appliqué :

Fonctions CN avec syntaxe conforme DIN 66025 (extensions incl.)

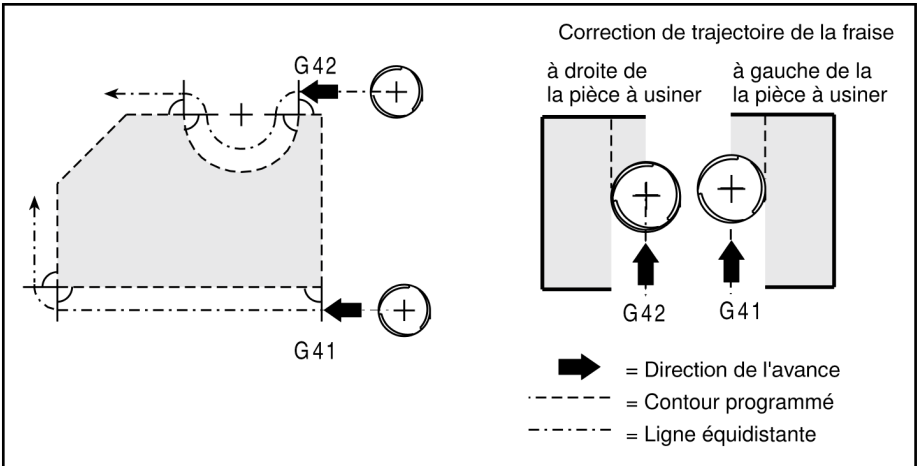


Fig. 5-19: Correction de trajectoire de la fraise

Syntaxe :

G40	Correction de trajectoire de la fraise DÉSACTIVÉE (état de mise en service) Le comportement des blocs G40 sans mouvement de déplacement programmé peut être configuré via les paramètres machine : • Désactivation de la correction, immédiatement et perpendiculairement au bloc de déplacement précédent (réglage par défaut). • Désactivation de la correction seulement à partir du prochain bloc de déplacement programmé. Si un mouvement de déplacement est programmé dans le bloc G40, la commande désactive la correction linéairement sur le parcours menant au point d'arrivée du mouvement de déplacement programmé. Selon qu'il s'agit d'un angle intérieur ou extérieur, la correction est désactivée à partir de la verticale finale du bloc précédent ou de la verticale initiale du bloc actuel. S'il s'agit d'un angle extérieur, un segment de trajectoire transitoire est inséré éventuellement pour les raccords entre segments de contour (G43/G44) en fonction de la stratégie actuelle. Ce comportement s'applique également pour la désactivation de la correction après un bloc G40 sans mouvement de déplacement, lorsque la correction doit être désactivée seulement lors du bloc de déplacement suivant.
G40(NOM)	Correction de trajectoire de la fraise DÉSACTIVÉE La fonction est programmée sans mouvement de déplacement. La correction n'est désactivée que lors du bloc de déplacement suivant. Il convient de tenir compte ici du fait qu'un déplacement incrémental programmé par la suite se rapporte au centre de l'outil. Par conséquent, l'outil se déplace effectivement selon le mouvement incrémental indiqué. Le cas échéant, le mouvement qui en résulte se distingue ainsi du mouvement de déplacement incrémental programmé dans un bloc directement avec G40.
G40(ORTH)	Correction de trajectoire de la fraise DÉSACTIVÉE Indépendamment du fait qu'il existe un angle intérieur ou extérieur au niveau du raccord entre segments du contour jusqu'au bloc de déconnexion, la correction est désactivée à partir de la verticale finale du bloc précédent. La correction est désactivée linéairement sur le parcours menant au point d'arrivée du mouvement de déplacement programmé.

Fonctions CN avec syntaxe conforme DIN 66025 (extensions incl.)

G41	Correction de trajectoire de la fraise à gauche de la pièce à usiner ACTIVÉE (avec des valeurs de correction positives dans le sens de l'usinage). Le comportement des blocs G41 sans mouvement de déplacement programmé peut être configuré via les paramètres machine : • Activation de la correction, immédiatement et perpendiculairement à la direction initiale du bloc de déplacement suivant (réglage par défaut). • Activation de la correction seulement lors du bloc de déplacement programmé suivant. Dans un bloc G41 avec mouvement de déplacement programmé, la correction est activée au sein de ce bloc. Pour activer la correction dans un bloc de déplacement, elle doit toujours l'être linéairement sur le parcours menant au point d'arrivée programmé. En cas d'angle intérieur, la correction est activée sur la verticale initiale du mouvement de déplacement programmé suivant. En cas d'angle extérieur, la correction est activée sur la verticale finale du bloc de mise en service. Un segment de trajectoire transitoire est éventuellement inséré en fonction de la stratégie actuelle pour le raccord entre segments de contour (G43/G44).
G41(ORTH)	Correction de trajectoire de la fraise à gauche de la pièce à usiner ACTIVÉE Indépendamment de l'existence d'un angle intérieur ou extérieur au niveau du raccord entre segments de contour suivant, la correction est activée sur la verticale initiale du bloc de déplacement suivant.
G42	Correction de trajectoire de la fraise à droite de la pièce à usiner ACTIVÉE Identique à G41 pour le reste
G42(ORTH)	Correction de trajectoire de la fraise à droite de la pièce à usiner ACTIVÉE Identique à G41(ORTH) pour le reste

Tab. 5-43: Syntaxe G40 à G42



Dans certaines circonstances, les valeurs de correction sont saisies ou obtenues immédiatement sans programmation d'un mouvement de déplacement à part. Ceci risque d'endommager la pièce à usiner ou l'outil.

C'est pourquoi toutes les informations fournies dans le présent chapitre doivent impérativement être observées !

Particularités et restrictions :

- Les fonctions G40, G41 et G42 sont des fonctions modales qui se révoquent réciproquement.
- Si G2, G3 ou G5 sont actives, les fonctions G40, G41 et G42 peuvent être programmées uniquement sans mouvement de déplacement.
- En cas d'activation et de désactivation de la correction, le comportement peut être influencé par les paramètres machine et les paramètres de fonction NOM et ORTH.
- *Si G41 et G42 sont actives, les fonctions suivantes sont **interdites** :*
 - G17 à G20 (changement de plan)
 - G63 (taroudage sans mandrin de compensation)
 - G74 (approche des coordonnées du point de référence)
 - G75 (déplacement contre le palpeur de mesure)
 - G76 (approche de la position fixe des axes machine)

Fonctions CN avec syntaxe conforme DIN 66025 (extensions incl.)

- G54.x à G59.x (décalages d'origine)
- G154.x à G159.x (plan incliné)
- G40, G41 et G42 n'ont aucune influence sur une correction active de la longueur d'outil.

5.2.16 Raccords entre segments de contour pour la correction de trajectoire de la fraise : arc de cercle "G43", point d'intersection "G44" arc de cercle

Description : Fonction pour la correction active de trajectoire de la fraise (G41, G42).

*La commande réalise un raccord entre segments de contour **aux angles extérieurs** soit sous la forme d'un*

- arc de cercle généré automatiquement (G43), soit sous la forme d'un
- point d'intersection des lignes équidistantes (G44).

G43 : Arc de cercle

Le vide entre les éléments de la trajectoire est comblé par un arc de cercle tangentiel de rayon "r" :

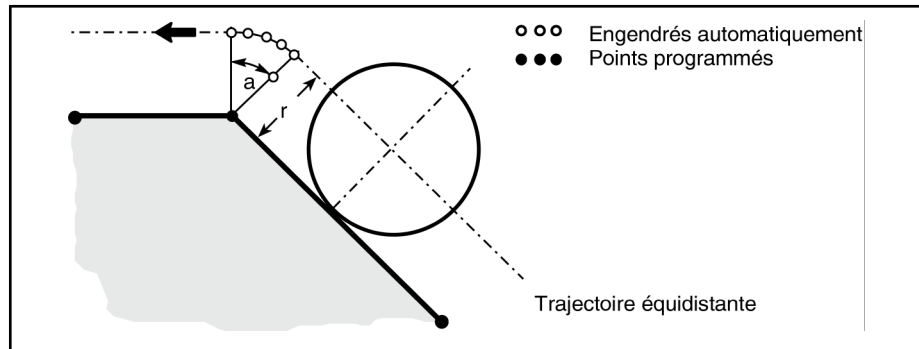


Fig. 5-20: Arc de cercle G43

G44 : Point d'intersection

La commande essaie de combler le vide entre les éléments de la trajectoire en définissant un point d'intersection des deux lignes équidistantes.

En fonction de l'écart "A" entre l'angle de contour "KE" et le point d'intersection "S", la commande procède de la façon suivante :

Fonctions CN avec syntaxe conforme DIN 66025 (extensions incl.)

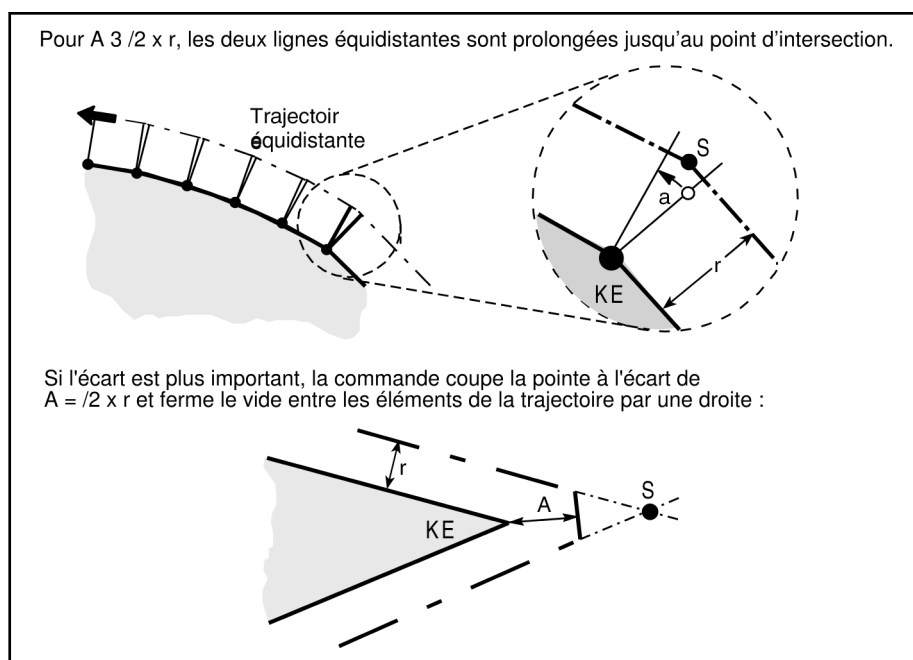


Fig. 5-21: Point d'intersection G44



Si aucun point d'intersection n'existe, la ligne est fermée par un arc de cercle comme avec G43.

Syntaxe :

G43	Raccord entre segments de contour sous la forme d'un arc de cercle
G44	Raccord entre segments de contour sous la forme d'un point d'intersection des lignes équidistantes

Tab. 5-44: Syntaxe G43, G44

Les points suivants s'appliquent ici :

- Les fonctions G43 et G44 sont des fonctions modales qui se révoquent réciproquement.
- L'état de mise en service peut être défini à l'aide de paramètres machine.

Particularités et restrictions : Les fonctions G43 ou G44 sont programmées sans condition de course.

5.2.17 Correction de l'avance : Point de pénétration de la fraise "G45", centre de la fraise "G46"

Description : Fonction pour la correction active de trajectoire de la fraise (G41, G42).

Ces fonctions définissent si, en cas d'interpolation circulaire, la commande doit maintenir l'avance programmée constante

- au point de pénétration de la fraise (trajectoire de coupe de la fraise) ou
- sur la trajectoire du centre de la fraise

Fonctions CN avec syntaxe conforme DIN 66025 (extensions incl.)

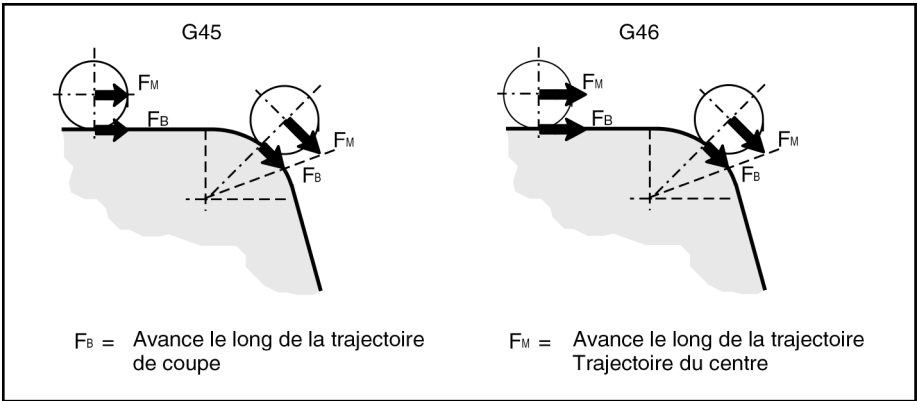


Fig. 5-22: Avance G45 et G46

Syntaxe :	G45	Maintenir constante l'avance F_B le long de la trajectoire de coupe.
	G46	Maintenir constante l'avance F_M le long de la trajectoire du centre de la fraise.

Tab. 5-45: Syntaxe G45, G46

Les points suivants s'appliquent ici :

- Les fonctions G45 ou G46 ont un effet modal et se révoquent réciproquement.
- L'état de mise en service peut être défini à l'aide de paramètres machine.

Particularités et restrictions : Étant donné que la vitesse d'avance peut augmenter fortement au niveau des contours circulaires, la fonction G45 doit être utilisée uniquement pour le fraisage de finition.

5.2.18 Correction de la longueur d'outil "G47, G48"

Description : La fonction :

- active/désactive la correction de la longueur d'outil ;
- commute en option l'affectation des valeurs de correction de la longueur L_1 , L_2 et L_3 pour chaque coordonnée (fonctionnement similaire à G78, voir [chap. 5.2.30 "Commutation de correction ACTIVÉE G78, commutation de correction DÉSACTIVÉE G79" à la page 175](#)).

Lorsque la correction de la longueur d'outil est activée, les éléments suivants ont un effet :

- les valeurs de correction de la longueur L_1 , L_2 et L_3 du bloc de correction D actuellement sélectionné (Dxx) ;
- les valeurs de correction de la longueur L_1 , L_2 et L_3 d'une correction externe de l'outil (EDxx) ;
- la fonction "Orientation statique de l'outil" (paramétrable avec STO, voir [chap. 6.133 "Paramétrage de l'orientation statique de l'outil StatToolOri, STO" à la page 419](#)) ;
- la fonction "Correction de la position de l'arête coupante", dans la mesure où une position de l'arête coupante est indiquée dans le bloc de correction D (Dxx) actuellement sélectionné et où la correction de trajectoire de la fraise G41/G42 est active (voir [chap. 5.2.15 "Correction de trajectoire de la fraise G40, G41, G42" à la page 155](#)).

Fonctions CN avec syntaxe conforme DIN 66025 (extensions incl.)

Syntaxe :

G47	Correction de la longueur d'outil ACTIVÉE
G47 (ActPlane)	Attribuer les valeurs de correction de la longueur L_1 , L_2 et L_3 des coordonnées principale, secondaire et normale du plan actif, puis activer la correction de la longueur d'outil. Reste active en tant que valeur modale et est reconfigurée automatiquement à chaque changement de plan ultérieur.
G47 ({ {-} <coord L1> } {, { {-} <coord L2> } {, { {-} <coord L3> } })	Modifier l'affectation des valeurs de correction de la longueur L_1 , L_2 et L_3 aux différentes coordonnées, puis activer la correction de la longueur d'outil.
G47 ()	Attribuer les valeurs de correction de la longueur L_1 , L_2 et L_3 conformément aux réglages des paramètres machine 7050 01300 et 7050 01310, puis activer la correction de la longueur d'outil.
G48	Correction de la longueur d'outil DÉSACTIVÉE.
avec	
<coord. Li>	Nom de la coordonnée du WCS à laquelle la correction L_i (avec $i = 1, 2, 3$) doit être attribuée dans le système de coordonnées de la pièce à usiner. Si la correction L_i (avec $i = 1, 2, 3$) doit agir sur les coordonnées du système de coordonnées de l'outil (TCS), il convient alors d'utiliser les désignateurs de coordonnées XTR, YTR et ZTR et d'activer la transformation des axes correspondante. Signe négatif optionnel : La correction est calculée dans le sens négatif. Les valeurs de correction L_i auxquelles aucune coordonnée n'est attribuée sont négligées.

Tab. 5-46: Syntaxe G47, G48

Particularités et restrictions :

- Les coordonnées qui ne sont pas comprises dans le réglage par défaut du canal ne sont jamais prises en compte dans la syntaxe "G47()".
- Les fonctions G47.. et G48 sont des fonctions modales qui se révoquent réciproquement.
- Le fait que les adresses de coordonnées programmées sous G47 se rapportent aux coordonnées de la pièce à usiner ou aux coordonnées de l'outil dépend du désignateur de coordonnées utilisé (voir <coordonnée i> ci-dessus).
- Une programmation mixte telle que "G47(X,,ZTR)" est interdite.
- Les fonctions G47/G48 peuvent être programmées conjointement avec d'autres conditions de course, informations de déplacement ou fonctions auxiliaires.

Fonctions CN avec syntaxe conforme DIN 66025 (extensions incl.)

Exemples :	G47 (X, , -Z)	La correction L ₁ est attribuée à la coordonnée X du système de coordonnées de la pièce à usiner (WCS) et calculée dans le sens positif. La correction L ₂ n'est pas calculée, puisqu'aucune coordonnée ne lui est attribuée. La correction L ₃ est attribuée à la coordonnée Z du WCS et calculée dans le sens négatif. Ensuite, la correction de la longueur d'outil est activée.
	G47 (, , ZTR)	La correction L ₁ et la correction L ₂ ne sont pas calculée puisqu'aucune coordonnée ne leur est attribuée. La correction L ₃ est attribuée à la coordonnée Z du système de coordonnées de l'outil (TCS) et calculée dans le sens positif. Ensuite, la correction de la longueur d'outil est activée.

5.2.19 Décalage d'origine programmable "G52"

Description : Le décalage d'origine programmable permet de décaler le système de coordonnées de la machine (MCS) dans l'espace grâce à une programmation dans le programme pièce.

Le décalage d'origine programmable peut être activé pour un total de 5 banques d'origines actives (groupes).

Les décalages issus des différentes banques d'origines ont toujours un effet additif.

Le décalage d'origine programmable complète les décalages d'origine G54.1 à G59.5, pour lesquels les valeurs de décalage doivent être chargées à partir des tables d'origine par une variante pour laquelle les valeurs de décalage sont programmées directement dans le programme pièce sans avoir été indiquées dans la table de décalage d'origine.

Le décalage d'origine programmable d'une banque constitue un groupe modal avec les décalages d'origine G54.x à G59.x de la banque correspondante. Les fonctions d'un groupe modal se révoquent réciproquement.

Le décalage d'origine d'une banque donnée est désactivé avec G53.x. G53 permet de désactiver tous les décalages d'origine.

Syntaxe :	G52.<banque décalages d'origine> (<coordonnées machine>) Décalage d'origine programmable de la <banque décalages d'origine> ACTIVÉ	
	Format abrégé : G52 pour le décalage d'origine programmable de la banque 1 avec	
	<banque décalages d'origine> 1 à 5, nombre entier	

Tab. 5-47: Syntaxe G52

Le décalage d'origine est désactivé avec :

G53 Décalages d'origine pour toutes les banques de correction DÉSACTIVÉS

G53.<banque DO> Décalage d'origine de la <banque DO> DÉSACTIVÉ

G54.<banque DO> à G59.<banque DO> Décalage d'origine (sur la base de la table) de la <banque DO> ACTIVÉ

Les points suivants s'appliquent ici :

- G52.1, G53.1, G54.1 à G59.1
sont des fonctions modales et se révoquent réciproquement.

Fonctions CN avec syntaxe conforme DIN 66025 (extensions incl.)

G52.2, G53.2, G54.2 à G59.2

sont des fonctions modales et se révoquent réciproquement.

G52.3, G53.3, G54.3 à G59.3

sont des fonctions modales et se révoquent réciproquement.

G52.4, G53.4, G54.4 à G59.4

sont des fonctions modales et se révoquent réciproquement.

G52.5, G53.5, G54.5 à G59.5

sont des fonctions modales et se révoquent réciproquement.

G53 active les fonctions G53.1 à G53.5 et

désactive par conséquent tous les décalages d'origine.

- Si le décalage d'origine est programmé seul, il n'entraîne aucun mouvement de déplacement. Seul le système de coordonnées de la machine (MCS) est décalé.
- Le décalage d'origine programmable peut être programmé conjointement avec un mouvement de déplacement. Le décalage d'origine modifié décale le MCS actuel, ce qui entraîne simultanément une modification du WCS actuel. Le mouvement de déplacement programmé dans le même bloc se réfère au nouveau WCS.

Exemple : Effet additif des décalages d'origine provenant de différentes banques de décalages d'origine.

Programme :

N20 G52.1 (X100, Y100)	Aucun mouvement de déplacement Décalage d'origine sur la position ACS X100 Y100 activé.
N30 G52.2 (X300, Y100)	Aucun mouvement de déplacement Décalage d'origine additif activé L'origine résultante de la machine est maintenant sur la position ACS X400 Y200.
N40 G90 G1 F1000 X10 Y15	Les axes se déplacent sur la position ACS X410 Y215.
N50 G53.1	La banque d'origine 1 est désactivée. L'origine de la machine est maintenant sur la position X300 Y100.
N60 X10 Y10	Les axes se déplacent sur la position ACS X310 Y110.
N70 G53	Tous les décalages d'origine sont désactivés.

Particularités et restrictions : L'origine programmable se rapporte aux coordonnées de la machine.
 Le décalage d'origine programmable peut être programmé dans un bloc avec d'autres conditions de course.
 Il est possible de programmer un mouvement de déplacement dans le même bloc qu'un décalage d'origine.

Fonctions CN avec syntaxe conforme DIN 66025 (extensions incl.)

5.2.20 Décalages d'origine (DO) "G53", "G53.1-G59.1" à "G53.5-G59.5"

Description : Le décalage d'origine permet de **décaler** le système de coordonnées de la machine dans l'espace.

 Si une correction du plan sous-tendu s'avère nécessaire, voir la fonctionnalité "BcsCorr".

Les "tables de décalage d'origine" comprennent les distances de décalage pour les coordonnées de la machine sur le canal :

- Une table de décalage d'origine comprend 5 banques de décalages d'origine (groupes) avec 6 décalages d'origine chacune (DO).
- Les décalages issus des différentes banques de décalages d'origine ont toujours un effet additif.
- Les décalages au sein d'une banque DO s'écrasent réciproquement.
- Les décalages ayant le numéro de canal 0 ont un effet dans tous les canaux dans lesquels ils ont été activés. Au contraire, les décalages ayant un numéro de canal > 0 ne peuvent avoir un effet que sur le canal indiqué. Si les mêmes coordonnées de machine sont saisies plusieurs fois pour différents canaux, veiller à ce que le numéro de canal 0 ne soit pas utilisé, faute de quoi tous les décalages suivants de cette coordonnée qui sont spécifiques aux canaux ne seront plus pris en compte.

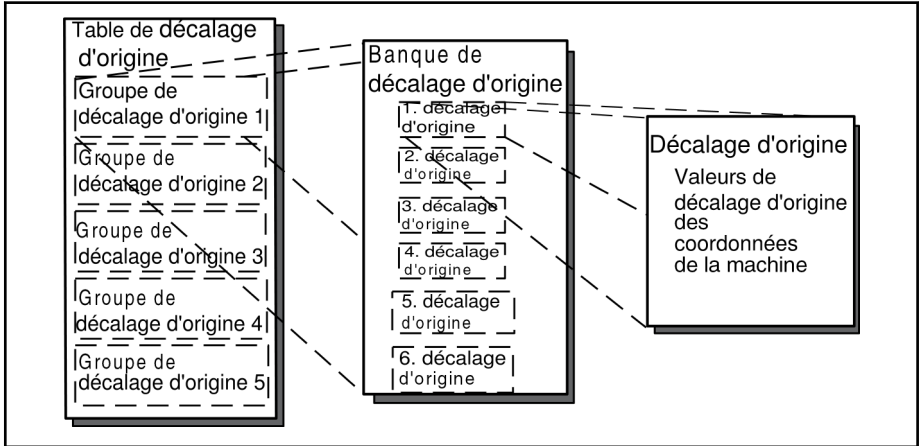


Fig. 5-23: Décalages d'origine (DO)

 Pour l'édition des tables de décalage d'origine, consulter le mode d'emploi.

Syntaxe : Activer la table DO voulue avec "ZoTSel" (voir [chap. 6.150 "Activation des tables de décalage d'origine ZoTSel, ZOS" à la page 448](#)).

Programmer la fonction requise :

G53	Tous les décalages d'origine pour toutes les banques de correction DÉSACTIVÉS
G53.<banque DO>	Tous les décalages d'origine de la <banque DO> DÉSACTIVÉS
G54.<banque DO>	1er décalage d'origine de la <banque DO> ACTIVÉ
G55.<banque DO>	2e décalage d'origine de la <banque DO> ACTIVÉ
G56.<banque DO>	3e décalage d'origine de la <banque DO> ACTIVÉ

Fonctions CN avec syntaxe conforme DIN 66025 (extensions incl.)

G57 . <banque DO>	4e décalage d'origine de la <banque DO> ACTIVÉ
G58 . <banque DO>	5e décalage d'origine de la <banque DO> ACTIVÉ
G59 . <banque DO>	6e décalage d'origine de la <banque DO> ACTIVÉ
Format abrégé : G54 à G59	Pour le décalage d'origine correspondant de la banque 1
avec	
<banque DO>	1 à 5, nombre entier

Tab. 5-48: Syntaxe G53 à G59

Les points suivants s'appliquent ici :

- G52.1, G53.1, G54.1 à G59.1
sont des fonctions modales et se révoquent réciproquement.
G52.2, G53.2, G54.2 à G59.2
sont des fonctions modales et se révoquent réciproquement.
G52.3, G53.3, G54.3 à G59.3
sont des fonctions modales et se révoquent réciproquement.
G52.4, G53.4, G54.4 à G59.4
sont des fonctions modales et se révoquent réciproquement.
G52.5, G53.5, G54.5 à G59.5
sont des fonctions modales et se révoquent réciproquement.
G53 active les fonctions G53.1 à G53.5 et
désactive par conséquent tous les décalages d'origine.
- Si les fonctions sont programmées seules, elles n'entraînent aucun mouvement de déplacement. Seul le système de coordonnées de la machine est décalé.
- Les fonctions peuvent également être programmées dans le même bloc que d'autres conditions de course. Dans ce cas, le décalage d'origine correspondant est activé, puis le déplacement vers la position programmée démarre.
- Les saisies DO pour un certain canal (canal > 0) sont actives uniquement sur le canal indiqué. Il est ainsi possible d'activer différentes origines pour le même désignateur des coordonnées de machine avec une table de décalage d'origine à part pour chaque canal.

Exemple : La table de décalage d'origine "Z01" contient les valeurs de décalage suivantes dans la banque de décalages d'origine 1 :

1er décalage d'origine : X100, Y100

2e décalage d'origine : X300, Y100

3e décalage d'origine : X500, Y100

4e décalage d'origine : X100, Y450

5e décalage d'origine : X300, Y450

6e décalage d'origine : X500, Y450

Effet des fonctions G54.1 à G59.1 :

Fonctions CN avec syntaxe conforme DIN 66025 (extensions incl.)

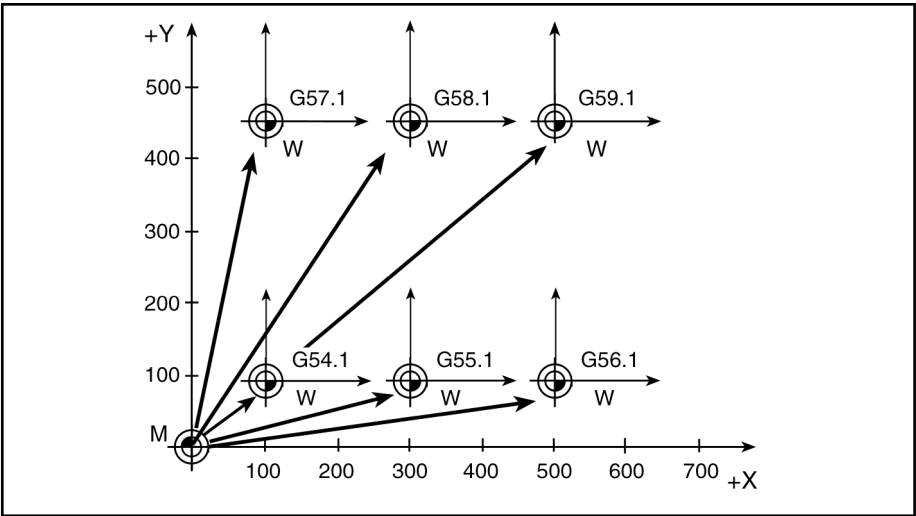


Fig. 5-24: Effet des fonctions G54.1 à G59.1

Effet additif des décalages d'origine issus de différentes banques de décalages d'origine.

La table de décalage d'origine "Z01" contient les valeurs de décalage suivantes :

Banque de décalages d'origine 1, 1er décalage d'origine : X100, Y100

Banque de décalages d'origine 1, 2e décalage d'origine : X300, Y100

Banque de décalages d'origine 2, 4e décalage d'origine : X200, Y350

:	Position actuelle de la machine 0,0
:	Aucun décalage d'origine
:	Toutes les indications de coordonnées suivantes sont des coordonnées de machine !
N40 ZOS (Z01)	Activer la table de décalage d'origine "Z01"
N50 G54.1	Aucun mouvement de déplacement Décalage sur X100, Y100 activé
N60 G55.1 X...Y...	Décalage sur X300, Y100 activé Puis mouvement de déplacement
N70 G57.2 X...Y...	Décalage sur X500, Y450 activé Puis mouvement de déplacement
N80 G53	Tous les décalages encore actifs DÉSACTIVÉS

5.2.21 Arrêt précis ACTIVÉ/DÉSACTIVÉ "G61, G62"

Description : Minimise l'erreur de poursuite à la fin du bloc.

Lors des mouvements d'outils, les valeurs de consigne et les valeurs réelles des différents axes sont décalées dans le temps en raison de la dynamique propre à la machine. Lors de l'usinage, cet "effet de poursuite" entraîne une erreur de poursuite dont l'ampleur dépend de la vitesse d'avance et du facteur KV (dynamique de l'axe). Si les raccords entre segments de contour (angles) sont irréguliers, cette erreur de poursuite se traduit par un "adoucissement" de l'angle.

Fonctions CN avec syntaxe conforme DIN 66025 (extensions incl.)

Les points suivants s'appliquent ici :

- La distance de poursuite (décalage entre la valeur de consigne et la valeur réelle) à la fin du bloc peut être réduite à la taille de certaines fenêtres cibles.
- G61 n'a d'effet que sur les mouvements en avance. Cette fonction n'agit pas sur des mouvements en avance rapide.
- Contrairement à la fonction d'arrêt précis G1, G61 active toutes les fonctions d'interpolation (donc également les interpolations circulaires/hélicoïdales, spline) à l'exception de G0.
- G61/G62 sont des fonctions modales qui se révoquent réciproquement.

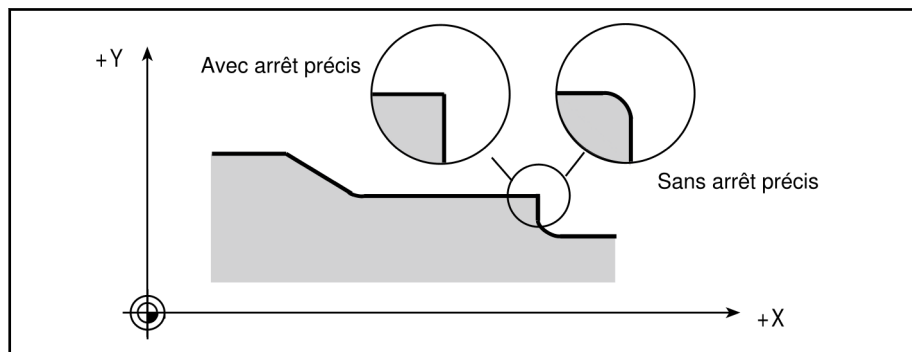


Fig. 5-25: Arrêt précis



Quand G61 (arrêt précis activé) est active, la commande ralentit toujours à $v = 0$ à la fin du bloc.

Syntaxe : Fonction de base :

G61	Arrêt précis ACTIVÉ
G62	Arrêt précis DÉSACTIVÉ

Tab. 5-49: Syntaxe G61, G62

Pour l'arrêt précis ACTIVÉ, la dernière fenêtre de positionnement active s'applique encore au mode trajectoire.

Valeur par défaut : Fenêtre de positionnement fin

Paramètres optionnels

Pour sélectionner la fenêtre de positionnement au mode d'avance.

G61 (IPS1 IPS2 IPS3)	
avec	
IPS1	Arrêt précis ACTIVÉ Attendre la fenêtre de positionnement fine. À la fin du bloc, la commande ralentit d'abord à la vitesse de déplacement $v = 0$. Le bloc suivant n'est déplacé que lorsque cette fenêtre de positionnement a été atteinte pour tous les axes participants.

Fonctions CN avec syntaxe conforme DIN 66025 (extensions incl.)

IPS2	Arrêt précis ACTIVÉ Attendre la fenêtre de positionnement brute. L'interface du canal indique "Zone Inpos 2 activée" (voir aussi la documentation "Interface API IndraMotion MTX"). À la fin du bloc, la commande ralentit d'abord à la vitesse de déplacement $v = 0$. Le bloc suivant n'est déplacé que lorsque cette fenêtre de positionnement a été atteinte pour tous les axes participants.
IPS3	Arrêt précis ACTIVÉ Décélération à $v = 0$ à la fin du bloc. À la fin du bloc, la commande ralentit à la vitesse sur trajectoire $v = 0$. Ensuite, le bloc suivant est déplacé sur une fenêtre de positionnement sans être vérifié.

Tab. 5-50: Syntaxe des paramètres optionnels de G61



Les paramètres "Fenêtre de positionnement fin" et "Fenêtre de pos. brute" peuvent être définis dans les fichiers Sercos pour la phase 3.

Vous trouverez de plus amples informations sur les fichiers Sercos dans la documentation "Paramètres machine IndraMotion MTX", sous "Initialisation Sercos".

Particularités et restrictions : Les fonctions G61/G62 doivent être programmées au plus tard dans le bloc dans lequel elles doivent s'appliquer.

Voir l'exemple suivant :

```

:
N40 G61                Arrêt précis ACTIVÉ
:
N50 Y200              Positionnement
:
N60 G62                Arrêt précis DÉSACTIVÉ
:
N70 Y0                Positionnement
:
N80 G61 Y200          Positionnement avec arrêt précis ACTIVÉ
:

```

5.2.22 Taraudage sans mandrin de compensation "G63"

- Description :**
- Synchronise l'interpolation linéaire de l'axe de perçage avec la broche commutée en mode axe C. Un mandrin de compensation qui, autrement, aurait dû capter les différences de vitesse entre l'axe de taraudage et la broche devient ainsi superflu.
 - La fonction G63 prend effet uniquement dans le bloc programmé.
 - Un signal peut être émis au niveau de l'interface du canal pendant le taraudage (configurable au sein des signaux "Fonction active" via les paramètres machine).
 - Au cours du taraudage, seul le potentiomètre d'avance est actif.

Fonctions CN avec syntaxe conforme DIN 66025 (extensions incl.)



En cas d'activation de G63, l'axe C de la broche doit être en mode broche.



Pour le taraudage avec plusieurs broches, voir aussi [chap. 6.140 "Sélection d'une broche pour le taraudage sans mandrin de compensation TappSp, TSP"](#) à la page 431.

Si des axes doivent être masqués pour générer l'avance, voir [chap. 6.42 "Génération de l'avance : masquage d'axes FeedAd, FAD"](#) à la page 289.

Syntaxe :

G63 (M<3 4> , S<vitesse de rotation> H<pas de filetage> , ANG<angle de départ>) <axe de perçage><profondeur du pas de pénétration> {F<avance>}
G63.2 (M<3 4> , S<vitesse de rotation> H<pas de filetage>) <axe de perçage><profondeur du pas de pénétration>{F<avance>}

Tab. 5-51: Syntaxe G63, G63.2

L'angle de départ peut uniquement être programmé avec G63.

Avec G63.2, la broche démarre toujours sur la position finale du bloc G63 précédent.

Le pas de filetage est le résultat du rapport entre l'avance sur trajectoire et la vitesse de rotation (F/S).



Des pas de filetage différents pour la pénétration et la remontée risquent d'endommager la pièce à usiner ou l'outil !

C'est pourquoi il convient de toujours programmer la pénétration et la remontée de sorte que les pas de filetage résultants soit identiques dans les deux cas !

Exemple : *Programme:*

```

N20 G0 X20 Y15 Z10 F1000          Positionnement
;----- Programmation de la vitesse -----
;   la pente est calculée par la commande : H=F/S
N30 G63(M3,S500) Z-10 F500        Pénétration (axe de perçage Z)
N40 G63(M4,S500) Z10 F500        Remontée d'outil (axe de perçage Z)
N50 G0 Z20
;----- Programmation de la pente -----
;   la vitesse de rotation est calculée par la commande : S=F/H
N60 G63(M3,H5) Z-10 F500          Pénétration (axe de perçage Z)
N70 G63(M4,H5) Z10               Remontée d'outil (axe de perçage Z)
N80 G0 Z100
:
;----- Perçage de trous profonds -----
N120 G0 X20 Y15 Z10 F1000         Positionnement pour le perçage de trous profonds
N130 G63(M3,S500) Z-20 F500       Pénétration section partielle 1
N140 G63(M4,S500) Z5 F500        Remontée d'outil section partielle 1
N150 G63(M3,S500) Z-40 F500       Pénétration section partielle 2
N160 G63(M4,S500) Z5 F500        Remontée d'outil section partielle 2
:
;----- Perçage avec angle de départ -----
N220 G0 X60 Y15 Z10              Positionnement
N230 G63(M3,S500,ANG60) Z-20     Perçage avec angle de départ 60 degrés
F500
N240 M257                        Notification à l'API
N250 G63.2(M4,S500) Z5 F500      Remontée d'outil G63.2
:

```

Particularités et restrictions :

- Les blocs de perçage et de remontée doivent être programmés directement l'un après l'autre.
- Si aucune autre valeur d'avance n'est indiquée (adresse F) dans le bloc G63, alors la commande utilise l'avance de trajectoire active.

Fonctions CN avec syntaxe conforme DIN 66025 (extensions incl.)

- Les adresses "M" et "S" sont actives uniquement dans le bloc G63 programmé.
- Ni "Orientation de la broche", ni "Arrêt de la broche" ne sont requis avant G63.
- Le passage au mode axe C s'effectue automatiquement.
Avant de démarrer, la commande attend en interne le message "IN-POS" de tous les axes participants. Si un axe dérive de sa zone INPOS, la fonction G63 ne sera pas lancée.
- À la fin du bloc de remontée, la broche retourne automatiquement en mode broche.
- G63 avec expressions CPL :
Une programmation sous la forme "G63 (M[MCODE%], S[DREHZ])" est interdite.
"G63 ([MCODE\$], S[DREHZ])" doit être programmé à la place.

5.2.23 Programmation en pouces "G70", programmation locale en pouces "INCH(...)"

Description : Permet d'indiquer des informations sur la course et sur l'avance ainsi que des accélérations en **pouces**.

Syntaxe :

G70	Les informations sur la course et sur l'avance ainsi que les accélérations sont interprétées en pouces.
<axe>=INCH(<valeur>)	Les informations sur la course de l'axe linéaire asynchrone sont interprétées en pouces.

Tab. 5-52: Syntaxe G70

Les points suivants s'appliquent ici :

- G70 est une fonction modale et révoque G71.
- L'état de mise en service peut être défini à l'aide de paramètres machine.
- La fonction G70 peut également être programmée dans le même bloc que d'autres conditions de course.
- *G70 se rapporte aux :*
 - courses de déplacement,
 - avances,
 - accélérations,
 - autres grandeurs auxiliaires géométriques telles que les paramètres d'interpolation I, J, K.

Exemple :

N40 G70	À partir de N40 inclus, toutes les informations relatives à la
:	course et à l'avance sont interprétées en pouces.
N50 W=INCH(2)	Déplacer l'axe asynchrone linéaire W en position 2 (pouces).

5.2.24 Programmation métrique "G71"

Description : Permet d'indiquer des informations sur la course et sur l'avance ainsi que des accélérations en **unités de mesure métriques**.

Fonctions CN avec syntaxe conforme DIN 66025 (extensions incl.)

Syntaxe :	<table border="1"> <tr> <td>G71</td><td>Les informations sur la course et sur l'avance ainsi que les accélérations sont interprétées en unités de mesure métriques.</td></tr> </table>	G71	Les informations sur la course et sur l'avance ainsi que les accélérations sont interprétées en unités de mesure métriques.
G71	Les informations sur la course et sur l'avance ainsi que les accélérations sont interprétées en unités de mesure métriques.		

Tab. 5-53: Syntaxe G71

Les points suivants s'appliquent ici :

- G71 est une fonction modale et révoque G70.
- L'état de mise en service peut être défini à l'aide de paramètres machine.
- La fonction G71 peut également être programmée dans le même bloc avec d'autres conditions de course.
- *G71 se rapporte aux :*
 - courses de déplacement,
 - avances,
 - accélérations,
 - autres grandeurs auxiliaires géométriques telles que les paramètres d'interpolation I, J, K.

Exemple :	<table border="1"> <tr> <td>N40 G71</td><td>À partir de N40 inclus, toutes les informations sur la course et sur l'avance sont interprétées en unités de mesure métriques.</td></tr> </table>	N40 G71	À partir de N40 inclus, toutes les informations sur la course et sur l'avance sont interprétées en unités de mesure métriques.
N40 G71	À partir de N40 inclus, toutes les informations sur la course et sur l'avance sont interprétées en unités de mesure métriques.		

5.2.25 Approche des coordonnées du point de référence "G74"

Description : Les axes programmés dans le même bloc avec G74 se déplacent simultanément vers leur position de référence.

La vitesse d'avance dépend de G0/G1 en avance rapide/en avance.

Avec G74, le système ne tient pas compte des cames ni des repères des points de référence. G74 est une **opération de positionnement pure sur les positions d'axes absolues**, et s'applique donc aussi aux axes avec codeurs à distances codées.



Les corrections éventuellement actives sont négligées lors de cette opération de positionnement !

Les points suivants s'appliquent ici :

- La fonction G74 agit bloc par bloc et est révoquée lorsque les axes machine programmés dans le bloc G74 ont atteint leur point de référence.
- Les valeurs réelles des axes ne sont pas réinitialisées lors de l'approche du point de référence avec G74.
- Les corrections éventuellement encore actives, les décalages d'origine et autres ne sont pas pris en compte dans le bloc G74 pour les axes programmés.

Syntaxe :	<table border="1"> <tr> <td>G74<coordonnées d'axe></td><td>Démarrer l'"Approche des coordonnées du point de référence"</td></tr> <tr> <td>avec</td><td></td></tr> <tr> <td><coordonnées d'axe></td><td>Les adresses des axes doivent être programmées avec une valeur numérique (par ex. X1 Y1 Z1). Cette valeur numérique n'a aucune influence sur la position du point de référence. Elle sert seulement à compléter le mot.</td></tr> </table>	G74<coordonnées d'axe>	Démarrer l'"Approche des coordonnées du point de référence"	avec		<coordonnées d'axe>	Les adresses des axes doivent être programmées avec une valeur numérique (par ex. X1 Y1 Z1). Cette valeur numérique n'a aucune influence sur la position du point de référence. Elle sert seulement à compléter le mot.
G74<coordonnées d'axe>	Démarrer l'"Approche des coordonnées du point de référence"						
avec							
<coordonnées d'axe>	Les adresses des axes doivent être programmées avec une valeur numérique (par ex. X1 Y1 Z1). Cette valeur numérique n'a aucune influence sur la position du point de référence. Elle sert seulement à compléter le mot.						

Tab. 5-54: Syntaxe G74

Fonctions CN avec syntaxe conforme DIN 66025 (extensions incl.)

Exemple :

N100 G74 X1 Y1 Z1

Les axes X, Y et Z démarrent simultanément leurs déplacements vers leurs positions de référence et les atteignent simultanément.

Particularités et restrictions :

G74 est programmé dans un bloc à part avec les axes à déplacer. Des fonctions auxiliaires et additionnelles peuvent être programmées dans le même bloc.

5.2.26 Approche du point de référence "G74(HOME)"

Description :



La fonction G74(HOME) s'applique aussi aux axes asynchrones !

Déclenche le référencement des axes programmés avec le programme pièce.

La fonction désactive la commande Sercos "Prise d'origine guidée par l'entraînement" (S-0-0148) respective correspondant aux entraînements des axes programmés.

L'entraînement se découple ensuite de la commande et génère lui-même ses données de position pour le référencement (prise d'origine). Pour cela, il utilise le paramètre Sercos S-0-0147 (paramètre de la prise d'origine), S-0-0041 (vitesse de la prise d'origine) et S-0-0042 (Accélération de la prise d'origine).



Pour une description détaillée, voir la documentation "Description des fonctions IndraMotion MTX".

Les points suivants s'appliquent ici :

- Si plusieurs axes sont programmés dans le bloc G74(HOME), ils approchent leur point de référence indépendamment les uns des autres (pas de mode trajectoire). Les points de référence ne sont donc pas atteints simultanément.
- Il est possible de programmer des axes **synchrones** et **asynchrones**. Le traitement de bloc est arrêté jusqu'à ce que tous les entraînements de la commande aient confirmé l'arrivée à leur point de référence.
- La fonction ne présente aucune différence par rapport à "Approche du point de référence" en mode "Réglage".

Syntaxe :

G74 (HOME) <coordonnées d'axe>	
	Déclencher la commande Sercos "Prise d'origine guidée par l'entraînement" pour les axes programmés.
avec	
<coordonnées d'axe>	Les adresses des axes doivent être programmées avec une valeur numérique (par ex. X1 Y1 Z1). Cette valeur numérique n'a aucune influence sur la position du point de référence. Elle sert seulement à compléter le mot.

Tab. 5-55: Syntaxe G74(HOME)

Exemple :

N1 G74 (HOME) X1 Y1 Z1

Envoyer la commande Sercos "Prise d'origine guidée par l'entraînement" aux entraînements des axes X, Y et Z.

Fonctions CN avec syntaxe conforme DIN 66025 (extensions incl.)

5.2.27 Déplacement contre le palpeur de mesure "G75"

Description : *La commande numérique a deux fonctions G75 :*

- Déplacement contre le 1er palpeur de mesure (G75.1 ou G75)
- Déplacement contre le 2e palpeur de mesure (G75.2)

Les deux fonctions G75 :

- déplacent et mesurent, c.-à-d. qu'il s'agit de fonctions générant un mouvement et exploitant la fonction du palpeur de mesure de l'entraînement ;
- agissent bloc par bloc et sont des fonctions locales ;
- utilisent exclusivement l'interpolation linéaire ;
- masquent le groupe modal G0, G1, G2, etc. dans le bloc précis dans lequel elles sont programmées ;
- arrêtent automatiquement le traitement du bloc sans nécessiter la programmation explicite de WAIT.

Syntaxe :

G75.1 <coordonnées d'axe> ou G75 <coordonnées d'axe>	Activer tous les axes de mesure du canal qui réagissent au palpeur de mesure 1 et déplacent les axes de déplacement sur les <coordonnées d'axe>.
G75.2 <coordonnées d'axe>	Activer tous les axes de mesure du canal qui réagissent au palpeur de mesure 2 et déplacent les axes de déplacement sur les <coordonnées d'axe>.

Tab. 5-56: Syntaxe G75

Exemple :

N100 G75.2 Y250 F500;	Déplacement sur la position Y250 à l'aide de F500
110 IF SD(9)=0 THEN;	Quand le 2e palpeur de mesure s'est déclenché
120 YPOS=PCSPROBE("Y");	Enregistrer la valeur mesurée de l'axe Y dans YPOS
:	
N130 MSG(CONTACT)	
140 ELSE	
150 SETWARN("AUCUN CONTACT!")	
N160 M0	
170 WAIT	
180 CLRWARN()	
190 ENDIF	
N200 ...	

Particularités et restrictions :

- **Mise en service :**
 Avant de pouvoir programmer une fonction G75, cette dernière doit être mise en service. Toutes les étapes requises sont expliquées dans la documentation "Description des fonctions IndraMotion MTX".
- La fonction G75 doit être programmée avec au moins une coordonnée d'axe, c.-à-d. avec au moins un axe de déplacement.

Fonctions CN avec syntaxe conforme DIN 66025 (extensions incl.)

- Les coordonnées définissent la profondeur de recherche maximale à laquelle le palpeur de mesure doit être déclenché au plus tard.
- Il est absolument interdit de programmer deux fonctions G75 dans un seul et même bloc CN. La commande numérique vérifie cette contrainte et génère une erreur si elle n'est pas satisfaite.
- *À la suite d'un bloc G75, l'utilisateur doit créer un programme CPL approprié lui permettant :*
 - de vérifier si le palpeur de mesure s'est déclenché ; avec SD(9)
 - d'accepter les valeurs mesurées dans le programme pièce ; avec PCSPROBE() ou PROBE()
 - de générer les avertissements et messages d'erreur requis ; avec SETWARN(), etc.

5.2.28 Approche de la position fixe des axes machine "G76"

Description : Permet la programmation et l'approche linéaire de positions dans le système de coordonnées de la machine sans devoir désactiver les éventuelles transformations et corrections actives dans le programme pièce.

Ceci peut s'avérer nécessaire, par exemple, pour un changement d'outil, les contrôles de rupture des outils, les cycles de mesure ou les changements de palettes.

G76 agit bloc par bloc en avance rapide (G0) ou en avance (G1), aussi en combinaison avec G93 (Programmation de temps), G94/G95 (Programmation de l'avance) et le mot F.

Les fonctions suivantes ne sont pas prises en compte avec G76 :

- Corrections d'outil (G41, G42, G47 Dxx, EDxx)
- Aides à la saisie (fonction miroir, mise à l'échelle, rotation, décalage)
- Décalages d'origine (G54.1 à G59.5)
- Placements (BcsCorr, G154.1 à G159.5)
- Transformations d'axes sur la base des coordonnées de machine, par ex. transformation à 5 axes
- Programmation relative (G91)
- Initialisation de la position du programme (SetPos)

Syntaxe :

G76<coordonnées de machine>	
avec	
<coordonnées de machine>	Coordonnées de la machine à approcher

Tab. 5-57: Syntaxe G76

Particularités et restrictions :

- La fonction G76 peut être programmée avec d'autres conditions de course (par ex. G0, G1, G93, G94, G95, mot F).
- Une programmation relative locale ("IC(...)") (voir [chap. 5.2.31 "Programmation absolue G90, programmation relative G91, programmation absolue locale AC\(...\), programmation relative locale IC\(...\)"](#) à la page 177) n'est pas autorisée avec G76 et entraîne une erreur d'exécution.

5.2.29 Sous-programmes asynchrones : repositionnement de coordonnées individuelles "G77"

Description : Les points suivants s'appliquent au sein d'un sous-programme asynchrone (ASUP) :

Les coordonnées programmées avec G77 sont déplacées à leur position de synchronisation respective (position à laquelle la coordonnée doit se trouver à la fin du sous-programme asynchrone). En option, il est possible d'indiquer un décalage pour la position de synchronisation sur laquelle la coordonnée doit être déplacée au sein du bloc G77.

En alternative, la position actuelle de la coordonnée peut être définie comme position de synchronisation.



Vous trouverez toutes les informations relatives à la mise en œuvre et à la programmation de sous-programmes asynchrones dans la documentation "Description des fonctions IndraMotion MTX".

Syntaxe :

G77<coord 1><mode><coord n><mode>... F<valeur>	
avec	
<coord i>	Nom de coordonnée (par ex. "x")
<mode>	1 : Déplacer la coordonnée sur la position de synchronisation 0 : Définir la position actuelle de la coordonnée en tant que position de synchronisation =IC(<décalage>) : Déplacer la coordonnée sur la valeur Position de synchronisation + <décalage>
<valeur>	Avance avec laquelle le point de redémarrage doit être atteint.

Tab. 5-58: Syntaxe G77

Particularités et restrictions :

- La fonction est prévue être utilisée dans un sous-programme asynchrone.
- Toutes les coordonnées non programmées avec G77 sont déplacées via une droite allant au point de redémarrage à la fin du sous-programme asynchrone.
- Les modifications de correction survenues entretemps au sein d'un sous-programme asynchrone sont prises en compte automatiquement pour le calcul interne du point de redémarrage requis.
- En alternative, il est également possible de programmer la syntaxe REPOS à la place de G77.

5.2.30 Commutation de correction ACTIVÉE "G78", commutation de correction DÉSACTIVÉE "G79"

Description : Affecte les **corrections de longueur** des fonctions de correction de la géométrie à des coordonnées individuelles.

Il s'agit ici des coordonnées :

- du système de coordonnées actuel de la pièce à usiner (WSC), ou
- du système de coordonnées de l'outil (TCS).

Fonctions CN avec syntaxe conforme DIN 66025 (extensions incl.)

Il est toujours possible d'attribuer les coordonnées du **système de coordonnées de la pièce à usiner (WCS)** quand l'outil est orienté perpendiculairement au plan de travail actuel et que son orientation par rapport au plan de travail reste constante pendant toute la durée de l'usinage.

Il est nécessaire d'attribuer les coordonnées du **système de coordonnées de l'outil (TCS)** quand l'orientation de l'outil change au cours de l'usinage, par ex. lors du fraisage d'une surface de forme libre. Une transformation d'axe (par ex. transformation à 5 axes ou à 6 axes) est nécessaire pour ce calcul de la correction. Les valeurs de correction sont calculées au sein de la transformation d'axe.

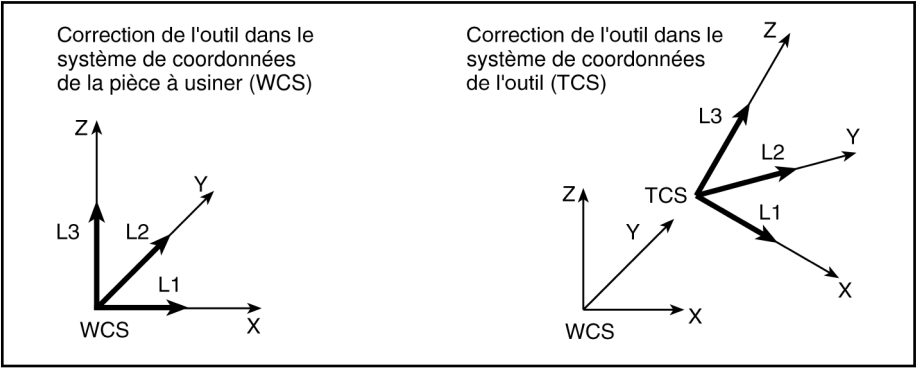


Fig. 5-26: Correction d'outil dans le système de coordonnées de la pièce à usiner WCS et dans TCS

 Pour activer les valeurs de correction actuelles pour l'usinage, voir G47 (chap. 5.2.18 "Correction de la longueur d'outil G47, G48" à la page 160).

Syntaxe :

G78 ({{-}<coordonnée1>},{{-}<coordonnée2>},{{-}<coordonnée3>})	
	Commutation de correction ACTIVÉE
avec	
<coordonnée i>	Nom de la coordonnée du WCS (nom d'axe logique/physique) auquel la correction L_i (avec $i = 1, 2, 3$) doit être activée. Si la correction L_i doit agir sur des coordonnées du système de coordonnées de l'outil (TCS), alors XTR, YTR et ZTR doivent être utilisés pour désigner les coordonnées. Signe négatif optionnel : La correction est calculée dans le sens négatif. Les valeurs de correction L_i auxquelles aucune coordonnée n'est attribuée sont négligées.
G78(ActPlane)	Les corrections L_1 , L_2 et L_3 sont affectées aux coordonnées principale, secondaire et normale du plan respectivement actif.
G79	Commutation de correction DÉSACTIVÉE Les réglages des paramètres machine sont de nouveau appliqués.

Tab. 5-59: Syntaxe G78, G79

Fonctions CN avec syntaxe conforme DIN 66025 (extensions incl.)

Exemples :	
G78(X, ,ZTR)	La correction L_1 est affectée à l'axe X du système de coordonnées de la pièce à usiner (WCS), la correction L_3 à l'axe Z du système de coordonnées de l'outil (TCS). Ces deux corrections sont calculées dans le sens positif.
G78(, , -Y)	La correction L_3 est affectée à l'axe Y du système de coordonnées de la pièce à usiner (WCS) et calculée dans le sens négatif.
G78(YA,YB)	La correction L_1 est affectée à l'axe YA et la correction L_2 à l'axe YB du système de coordonnées de la pièce à usiner (WCS). Ces deux corrections sont calculées dans le sens positif.
G79	Les corrections sont désactivées pour toutes les coordonnées par défaut du canal.

- Particularités et restrictions :**
- Les fonctions G78 et G79 sont des fonctions modales qui se révoquent réciproquement.
 - Les coordonnées qui ne font pas partie des réglages par défaut du canal ne sont pas prises en compte avec G79 !
 - Les adresses de coordonnées programmées sous G78 peuvent se rapporter aux coordonnées de la pièce à usiner ou aux coordonnées de l'outil (voir ci-dessus <coordonnée i>).
 - Les fonctions G78/G79 peuvent être programmées conjointement avec d'autres conditions de course, informations de déplacement ou fonctions auxiliaires.

5.2.31 Programmation absolue "G90", programmation relative "G91", programmation absolue locale "AC(...)", programmation relative locale "IC(...)"

Description : Définit si la commande doit interpréter les cotes indiquées pour les axes et les coordonnées en tant que valeurs absolues ou relatives (incrémentales).

- Les cotes indiquées en valeurs absolues font référence à l'origine actuelle (dans le système de coordonnées du programme).
- Les cotes indiquées en valeurs relatives font référence à la dernière position approchée.

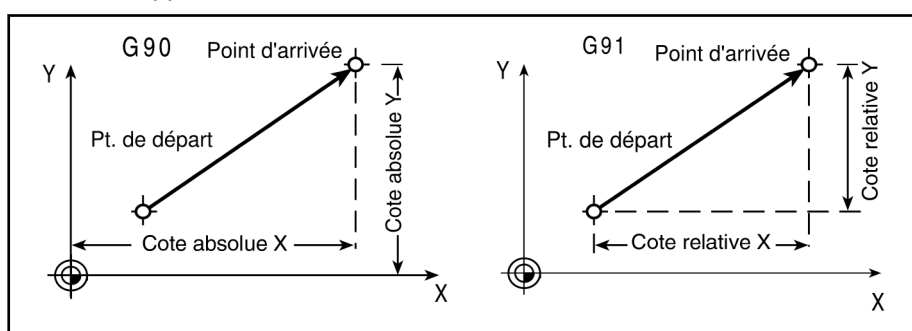


Fig. 5-27: Programmation absolue G90 et programmation relative G91

Les points suivants s'appliquent ici :

- Les fonctions G90 et G91 sont des fonctions modales qui se révoquent réciproquement.
- AC(...) et IC(...) agissent indépendamment d'un bloc G90/G91 actif uniquement pour l'information sur la course des axes dont la position est programmée avec AC(...) ou IC(...).

Fonctions CN avec syntaxe conforme DIN 66025 (extensions incl.)

Syntaxe :	G90	Programmation absolue
	G91	Programmation relative
	<axe>=AC (<valeur>)	Programmation absolue locale
	<axe>=IC (<valeur>)	Programmation relative locale
	avec	
	<axe>	Adresse d'axe logique
	<valeur>	Information sur la course pour <axe>

Tab. 5-60: Syntaxe G90, G91

Exemple :	N10 G1 G90	Programmation absolue activée
	N20 X100 Y100	Déplacement vers les coordonnées X100, Y100
	N30 G91	Programmation relative activée
	N40 X100 Y100	Déplacement vers les coordonnées X200, Y200
	N50 X=AC(50) Y50	Programmation absolue locale pour l'axe X Mouvement de déplacement au niveau de la machine vers X50, Y250

5.2.32 Programmation de temps "G93"

Description : La commande interprète les mots F en fonction de la valeur de la variable modale ITIM (voir chap. 6.57 "Programmation du temps inversé InvTime, ITIM" à la page 309), soit comme un temps d'usinage en secondes, soit comme un temps inversé en 1/min ou en 1/sec pour la trajectoire programmée (durée du bloc).

Par défaut, la durée est enregistrée en secondes (c.-à-d. ITIM=0).

Pour ITIM > 0, l'unité résulte de la formule :

$$F = \frac{v \left[\frac{mm}{ITIM \times sec} \right]}{s[mm]} = \frac{v \left[\frac{1}{min} \right]}{s[min]}_{ITIM=60} = \frac{v \left[\frac{1}{sec} \right]}{s[sec]}_{ITIM=1}$$

Les unités suivantes découlent des différentes valeurs ITIM :

Valeur ITIM	Nombre entier ; en fonction de la valeur ITIM, l'unité du mot F change au mode G93
0	Valeur F en [sec]
1	Valeur F en [1/sec]
60	Valeur F en [1/min]

Les points suivants s'appliquent ici :

Les fonctions G93, G94 et G95 sont des fonctions modales qui se révoquent réciproquement.



L'état de mise en service voulu (G93 (éventuellement ITIM), G94 ou G95) peut être défini dans les paramètres machine (par défaut : G94, ITIM=0).

Fonctions CN avec syntaxe conforme DIN 66025 (extensions incl.)

Syntaxe :

G93	Passer à la programmation de temps ou de temps inversé

Tab. 5-61: Syntaxe G93

Exemple :

N5 G93G1 X30 Y20F20	Valeur par défaut : ITIM non programmé ou ITIM=0
:	L'interpolation linéaire programmée dure 20 secondes.
N4 ITIM60	L'interpolation linéaire programmée dure 60/3 = 20 secondes.
N5 G93G1 X30 Y20F3	
:	
N4 ITIM1	L'interpolation linéaire programmée dure 1/5 = 200 millisecondes.
N5 G93G1 X30 Y20F5	
:	

Particularités et restrictions :

- En cas de commutation sur G94 ou G95, une valeur F programmée pendant G93 reste enregistrée dans la mémoire interne et est réactivée lors d'une nouvelle commutation sur G93.
- Le mot F défini dans les paramètres machine devient actif (par défaut : F0) suite à une montée en régime ou à une remise à zéro.
- La commande calcule en interne l'avance requise sur la base de la longueur de course du bloc de déplacement respectif et du temps d'usinage programmé.
Cependant, la commande peut limiter l'avance finalement active en fonction de la trajectoire programmée et des valeurs maximales des axes intéressés, de sorte à augmenter la durée du bloc.
- L'avance Override reste également active sous G93 selon la fonction OvrEna/OvrDis.

5.2.33 Programmation de l'avance (par min) "G94"

Description :

La commande interprète les mots F (voir [chap. 5.4.2 "Adresse F F" à la page 213](#) et [chap. 5.4.4 "Vitesse des axes asynchrones FA" à la page 215](#)) ou les mots Oméga (voir [chap. 5.4.5 "Adresse oméga \(avance\)" à la page 215](#)) en tant qu'avance pour une trajectoire programmée.

Les points suivants s'appliquent ici :

- Les fonctions G93, G94 et G95 sont des fonctions modales qui se révoquent réciproquement.
- L'avance programmée est interprétée en :
 - mm/min, si G71 est active ;
 - pouces/min, si G70 est active ;
 - degré/min pour les axes ronds.



Les paramètres machine permettent de régler le calibrage de l'avance pour G70/G71.



L'état de mise en service voulu (G93, G94 ou G95) peut être défini dans les paramètres machine (par défaut : G94).

Syntaxe :

G94	Commuter sur la programmation de l'avance

Tab. 5-62: Syntaxe G94

Fonctions CN avec syntaxe conforme DIN 66025 (extensions incl.)

Exemple :	N10 G71	Programmation métrique activée
	N20 G1G94X20 Y30F200	Déplacement avec une avance de 200 mm/min
	N30 G4 (F40)	Temporisation de 40 secondes
	N40 G70	Programmation en pouces activée
	N60 X300 Y400	L'avance F200 (en mm/min) est de nouveau active
	N70F100	Nouvelle valeur d'avance : 100 pouces/min
	:	

- Particularités et restrictions :
- En cas de commutation sur G93 ou G95, une valeur F ou Oméga programmée pendant G94 reste enregistrée dans la mémoire interne et est réactivée lors d'une nouvelle commutation sur G94.
 - Le mot F défini dans les paramètres machine devient actif (par défaut : F0) suite à une montée en régime ou à une remise à zéro.
La valeur Oméga est initialisée automatiquement sur "0" après chaque montée en régime.
 - L'avance effective peut être limitée par la commande à cause des vitesses maximales configurées pour les axes participants.
 - L'avance est influencée par l'avance Override en fonction de OvrEna/OvrDis.

5.2.34

Programmation absolue de la vitesse "G94(LINS)" avec adaptation de la vitesse proportionnelle à la course

Description : Dans un cas standard (G94 sans programmation de paramètres), lorsqu'un mot F est reprogrammé, le système effectue une interpolation vers la nouvelle valeur d'avance aussi rapidement que possible en tenant compte des valeurs effectives de l'accélération.

Avec G94(LINS), la vitesse de consigne qui en résulte est modifiée au sein du bloc proportionnellement à la course parcourue. Les valeurs effectives de l'accélération sont maintenues constantes. Dans la mesure où ces valeurs le permettent, la nouvelle vitesse de consigne est atteinte exactement à la fin du bloc, indépendamment de la valeur actuelle de l'Override.

Les points suivants s'appliquent ici :

- G94(LINS) a un effet modal et est révoquée par G94 (le cas standard (voir plus haut) s'applique de nouveau).
- L'avance programmée ne doit pas être programmée sur la même ligne que G94(LINS).
- L'avance programmée a un effet modal et absolu pour les blocs suivants.
- L'unité de l'avance programmée est le pouce/min ou le mm/min selon G70/G71.
- L'avance est influencée par l'avance Override en fonction de OvrEna/OvrDis.

Syntaxe :	<table><tr><td>G94 (LINS)</td><td>Activer la programmation directe de l'avance. Les mots F programmés ci-après entraînent une modification de la vitesse de consigne proportionnelle à la course parcourue.</td></tr></table>	G94 (LINS)	Activer la programmation directe de l'avance. Les mots F programmés ci-après entraînent une modification de la vitesse de consigne proportionnelle à la course parcourue.
G94 (LINS)	Activer la programmation directe de l'avance. Les mots F programmés ci-après entraînent une modification de la vitesse de consigne proportionnelle à la course parcourue.		

Tab. 5-63: Syntaxe G94 (LINS)

Fonctions CN avec syntaxe conforme DIN 66025 (extensions incl.)

Exemple :	N30 G94 G1 G71 F1000 X250 Y300	Le bloc est déplacé avec une avance sur trajectoire de 1000 mm/min.
	:	
	N50 G94(LINS) F2000 X300 Y200	L'avance sur trajectoire de consigne est augmentée à 2000 mm/min proportionnellement à la course jusqu'à la fin du bloc.
	:	
	N60 X350	Le bloc est déplacé avec une avance sur trajectoire de 2000 mm/min.
	:	
	N100 F500 X450	L'avance sur trajectoire de consigne est réduite à 500 mm/min proportionnellement à la course jusqu'à la fin du bloc.
	N110 G94	G94 dans un cas standard : les modifications de la vitesse sont interpolées aussi rapidement que possible en conservant les valeurs effectives de l'accélération.

- Particularités et restrictions :**
- Les valeurs limites existantes pour l'accélération ou la décélération sont surveillées. De ce fait, la vitesse finale qui en résulte ne peut éventuellement être atteinte que dans le bloc suivant.
 - La programmation d'une valeur de vitesse supérieure entraîne l'augmentation de la valeur modale pour la vitesse programmée dès le bloc actuel. Au contraire, la programmation d'une vitesse inférieure n'entraîne la modification de la valeur modale de la vitesse programmée qu'à partir du **bloc CN suivant**. Cela permet de garantir dans les deux cas que la vitesse voulue est atteinte à la **fin du bloc CN actuel** dans la mesure où les valeurs effectives de l'accélération le permettent.

5.2.35 Programmation absolue de la vitesse "G94(LINT)" avec adaptation de la vitesse proportionnelle au temps

Description : Dans un cas standard (G94 sans programmation de paramètres), lorsqu'un mot F est reprogrammé, le système effectue une interpolation vers la nouvelle valeur d'avance aussi rapidement que possible en tenant compte des valeurs effectives de l'accélération.

Avec G94(LINT), la vitesse de consigne qui en résulte est modifiée au sein du bloc proportionnellement au temps. Les valeurs effectives de l'accélération sont maintenues constantes. Dans la mesure où ces valeurs le permettent, la nouvelle vitesse de consigne est atteinte exactement à la fin du bloc, indépendamment de la valeur actuelle de l'Override. Les transitions de vitesse ainsi obtenues sont particulièrement douces (comparables à G94(DF...)).

Les points suivants s'appliquent ici :

- G94(LINT) a un effet modal et est révoquée par G94 (le cas standard (voir ci-dessus) s'applique de nouveau).
- L'avance programmée ne doit pas être programmée sur la même ligne que G94(LINT).
- L'avance programmée a un effet modal et absolu pour les blocs suivants.
- L'unité de l'avance programmée est le pouce/min ou le mm/min selon G70/G71.

Fonctions CN avec syntaxe conforme DIN 66025 (extensions incl.)

- L'avance est influencée par l'avance Override en fonction de OvrEna/OvrDis.

Syntaxe :

G94 (LINT)	Activer la programmation directe de l'avance. Les mots F programmés après modifient la vitesse de consigne proportionnellement au temps.

Tab. 5-64: Syntaxe G94 (LINT)

Exemple :

N30 G94 G1 G71 F1000 X250 Y300 :	Le bloc est déplacé avec une avance sur trajectoire de 1000 mm/min.
N50 G94 (LINT) F2000 X300 Y200 :	La consigne de l'avance sur trajectoire est augmentée à 2000 mm/min proportionnellement au temps jusqu'à la fin du bloc.
N60 X350 :	Le bloc est déplacé avec une avance sur trajectoire de 2000 mm/min
N100 F500 X450	La consigne de l'avance sur trajectoire est augmentée à 500 mm/min proportionnellement au temps jusqu'à la fin du bloc.
N110 G94	G94 dans un cas standard : les modifications de la vitesse sont interpolées aussi rapidement que possible en conservant les valeurs effectives de l'accélération.

Particularités et restrictions :

- Les valeurs limites existantes pour l'accélération ou la décélération sont surveillées. De ce fait, la vitesse finale qui en résulte ne peut éventuellement être atteinte que dans le bloc suivant.
- La programmation d'une valeur de vitesse supérieure entraîne l'augmentation de la valeur modale pour la vitesse programmée dès le bloc actuel. Au contraire, la programmation d'une vitesse inférieure n'entraîne la modification de la valeur modale de la vitesse programmée qu'à partir du **bloc CN suivant**. Cela permet de garantir dans les deux cas que la vitesse voulue est atteinte à la **fin du bloc CN actuel** dans la mesure où les valeurs effectives de l'accélération le permettent.

5.2.36 Programmation incrémentale de la vitesse "G94(...)" avec adaptation de l'accélération

Description : Modifie l'avance ou la vitesse de rotation relative par rapport à la dernière valeur effective active. L'accélération au sein du bloc G94(...) est adaptée de façon à ce que la vitesse ou la vitesse de rotation résultante ne soit atteinte qu'à la fin du bloc. Ceci entraîne un comportement d'accélération très doux.

Les points suivants s'appliquent ici :

- L'avance qui en résulte à la fin du bloc a un effet modal pour les blocs suivants.
- L'unité de l'avance incrémentale est le pouce/min ou le mm/min selon G70/G71.
- L'avance est influencée par l'avance Override en fonction de OvrEna/OvrDis.

Fonctions CN avec syntaxe conforme DIN 66025 (extensions incl.)

Syntaxe :

G94 (DF<va- leur F>)	Programmer de façon incrémentale la vitesse sur trajectoire (Delta Feed)
G94 (DS1=<va- leur S>)	Programmer de façon incrémentale la vitesse de rotation de la broche (Delta Speed) pour la broche 1
G94 (DF<valeur F>,DS7=<valeur S>)	
	Programmer la vitesse sur trajectoire et la vitesse de rotation de la broche 7 de façon incrémentale
G94 (DSA2=<axe>,DS2=<valeur S>)	
	Programmer de façon incrémentale la vitesse de rotation de l'axe sans fin <axe> (Delta Speed Axis)
avec	
<valeur F>	Vitesse sur trajectoire incrémentale Les valeurs positives augmentent et les valeurs négatives réduisent la vitesse sur trajectoire actuellement effective.
<valeur S>	<i>En fonction de l'affectation des axes, la valeur S signifie :</i> • Vitesse de rotation incrémentale DSi de la broche du canal du même index (sans indication d'axe) • Vitesse de rotation incrémentale DSi de l'axe sans fin de même index DSAi • Facteur de couplage incrémental DSi de l'axe sans fin de même index couplé en avance DSAi (voir aussi AFC) Les valeurs positives augmentent et les valeurs négatives réduisent la vitesse de rotation de la broche actuellement effective. Il est possible de programmer simultanément un maximum de 8 valeurs S (DS1 à DS8). La programmation incrémentale de la vitesse de rotation pour les axes sans fin peut entraîner une inversion de la direction.
<axe>	Désigne un axe sans fin asynchrone au mode d'asservissement en vitesse de rotation. Il est possible de prédéfinir simultanément des valeurs S incrémentales pour un maximum de 8 axes (DSA1 à DSA8).

Tab. 5-65: Syntaxe G94 (valeur F/valeur S)

Exemple :

N30 G94 (DF100) X250 Y300 Augmenter de manière linéaire l'avance sur trajectoire de 100 mm/min jusqu'à la fin du bloc
 :
 N50 G94 (DF-50) X300 Y200 Réduire de manière linéaire l'avance sur trajectoire de 50 mm/min jusqu'à la fin du bloc
 :
 N70 G94 (DS1=100) X25 Y30 Augmenter de manière linéaire la vitesse de rotation de la 1re broche à 100 tr/min jusqu'à la fin du bloc
 :

Augmenter de manière linéaire la vitesse sur trajectoire de 100 mm/min et la vitesse de rotation de la broche 2 de 150 tr/min jusqu'à la fin du bloc

Fonctions CN avec syntaxe conforme DIN 66025 (extensions incl.)

```
N90 G94 (DF100, DS7=150) X2 Y2
```

```
:
```

```
N100 B2=S (100) B3=S (50) Démarrer les axes sans fin B2 et B3 au mode  
d'asservissement en vitesse de rotation
```

Augmentation de la vitesse de rotation à
200 tr/min pour B2,

réduction de la vitesse de rotation à 30 tr/min pour
B3

jusqu'à la fin du bloc

```
N110 G94 (DSA1=B2, DS1=100, DSA2=B3, DS2=-20) X10
```

```
:
```

```
N120 B1=S (500) Démarrer l'axe sans fin B1 à 500 tr/min
```

Couplage de l'avance avec

0,5 rotation de B2/rotation de B1

```
N130 AFC (FRA=B1, FA=B2, CF=0.5) B2=S (1)
```

Augmentation à

0,8 rotation de B2/rotation de B1 jusqu'à la fin du
bloc

```
N140 G94 (DSA1=B2, DS1=0.3) X10 F1000
```

- Particularités et restrictions :**
- Les valeurs limites existantes pour l'accélération ou la décélération sont surveillées. De ce fait, la vitesse finale qui en résulte ne peut éventuellement être atteinte que dans le bloc suivant.
 - L'accélération calculée n'agit que dans le bloc G94(...). En cas d'interruption du bloc, la décélération est exécutée à l'accélération calculée.
 - L'avance absolue et l'avance incrémentale ne doivent pas être programmées simultanément dans le même bloc.
 - La programmation d'une valeur positive pour le changement de vitesse entraîne l'augmentation de la valeur modale de la vitesse programmée selon la valeur correspondante dès le bloc actuel. Par contre, la programmation d'un changement de vitesse négatif entraîne la diminution de la valeur modale de la vitesse programmée selon la valeur correspondante seulement dans le **bloc CN suivant**. L'adaptation de l'accélération permet de garantir dans les deux cas que la valeur de la vitesse qui en résulte est atteinte à la **fin du bloc CN actuel**.
 - En cas d'interruption anticipée du bloc ou si la spécification de course est manquante, la commande active immédiatement les données de vitesse de rotation incrémentales.

5.2.37 Programmation absolue de la vitesse "G94(...)" avec adaptation de l'accélération

Description : Modifie l'avance en continu sur plusieurs blocs, de la valeur actuelle jusqu'à l'avance cible. L'accélération jusqu'au bloc cible (nouvelle avance cible ou fin de rampe) est adaptée de sorte que la vitesse qui en résulte soit atteinte uni-

Fonctions CN avec syntaxe conforme DIN 66025 (extensions incl.)

quement au niveau du bloc cible. Ceci entraîne un comportement d'accélération très doux.

De même, les définitions de la vitesse de rotation de max. 8 broches du canal peuvent être modifiées en continu sur plusieurs blocs de la vitesse de rotation actuelle jusqu'à la vitesse de rotation cible en fonction de la course.

Les points suivants s'appliquent ici :

- La fonction G94 est modale, mais l'avance qui en résulte n'a pas d'effet modal sur les blocs suivants.
- Selon G70/G71, l'unité de l'avance cible est le pouce/min ou le mm/min.
- L'avance est influencée par l'avance Override en fonction de OvrEna/OvrDis. Cependant, il est généralement impossible d'atteindre une rampe continue avec un Override différent de 100 %.

Syntaxe :

G94 (TF<valeur F>)	Programmer la vitesse sur trajectoire cible en valeur absolue
G94 (TS1=<valeur S>)	Programmer la vitesse de rotation cible de la broche 1 en valeur absolue
G94 (TF<valeur F>,TS7=<valeur S>)	
	Programmer la vitesse sur trajectoire cible et la vitesse de rotation cible de la broche 7
G94 (TB)	Fin de rampe pour l'avance cible et les vitesses de rotation cibles
avec	
<valeur F>	Vitesse sur trajectoire cible
<valeur S>	Vitesse de rotation cible de la broche

Tab. 5-66: Syntaxe G94 (valeur TF/valeur TS)

Exemple :

N10 G94 F1000 M3 S1=1000 :	Avance sur trajectoire constante et vitesse de rotation constante.
N30 G94 (TF1500) X50 Y30 :	Augmenter l'avance sur trajectoire de F1000 à F1500 de manière linéaire jusqu'au bloc N50.
N50 G94 (TF500,TS1=2000) X70 Y50 :	Réduire l'avance sur trajectoire de F1500 à F500 de manière linéaire jusqu'au bloc N70. En même temps, la vitesse de rotation de la broche 1 est augmentée de 1000 tr/min à 2000 tr/min jusqu'au bloc N70.
N70 G94 (TB) X90 Y70 :	À partir du bloc cible, l'avance est de F500 et la vitesse de rotation de la broche 1 de 2000 tr/min.

Fonctions CN avec syntaxe conforme DIN 66025 (extensions incl.)

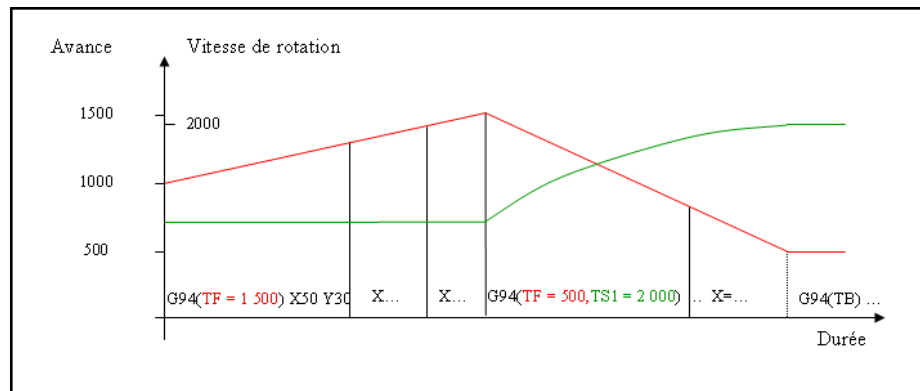


Fig. 5-28: Accélération et ralentissement continus

- Particularités et restrictions :**
- Les valeurs limites existantes pour l'accélération ou la décélération sont surveillées. De ce fait, la vitesse finale qui en résulte ne peut éventuellement être atteinte que dans le bloc suivant.
 - L'accélération calculée n'agit que dans le bloc G94(...). En cas d'interruption du bloc, le freinage est exécuté avec l'accélération maximale autorisée.
 - Dans une séquence d'avance cible (TF-TB), aucune avance incrémentale (DF) ni aucune avance modale (mot F) ne doivent être programmées.
 - Dans une séquence de vitesse de rotation cible (TS-TB), aucune vitesse de rotation incrémentale (DS) ni aucune vitesse de rotation modale de la broche (mot S) ne doivent être programmées.
 - Le nombre maximal autorisé de blocs d'avance/de vitesse de rotation cibles est limité par les paramètres de prévisualisation des blocs
NCO/LookAh/Ch[k]/MaxBlk/MaxBlkMiscFunc et
NCO/LookAh/Ch[k]/PercSplit/PercMiscFunc.
 - Les vitesses de rotation cibles peuvent uniquement être définies lors de la programmation directe de la broche (G97).

5.2.38 Programmation de l'avance (par rotation) "G95"

Description : La commande interprète les mots F (voir [chap. 5.4.2 "Adresse F F" à la page 213](#)) en tant qu'avance/rotation. Ceci est obligatoire pour la broche principale.

Les points suivants s'appliquent ici :

- Les fonctions G93, G94 et G95 sont des fonctions modales qui se révoquent réciproquement.
- *L'avance programmée est interprétée en :*
 - mm/tr, si G71 est active.
 - pouces/tr, si G70 est active.



- Les paramètres machine permettent de régler le calibrage de l'avance pour G70/G71.
- L'état de mise en service voulu (G93, G94 ou G95) peut être défini dans les paramètres machine (par défaut : G94).
- Pour la définition de la broche principale, voir MP 7020 00010 et la fonction "MainSp".

Fonctions CN avec syntaxe conforme DIN 66025 (extensions incl.)

Syntaxe :

G95	Commuter sur la programmation rotative de l'avance

Tab. 5-67: Syntaxe G95

Exemple :

N05 G71	Programmation métrique activée
N10 S200 M4	Vitesse de rotation de la broche de 200 tr/min, vers la gauche
N20 G1 G95 X20 Z30 F0.2	Déplacement avec une avance de 0,2 mm/tr
N30 G4 (S20)	Temporisation de 20 tours
N40 G70	Programmation en pouces activée
N60 X300 Z40	L'avance F0.2 (en mm/tr) s'applique de nouveau
N70 F0.1	Nouvelle avance de 0,1 pouce/tr
:	

Particularités et restrictions :

- G95 suppose une broche principale en rotation.
- L'avance effective est influencée d'une part par le potentiomètre de la broche, d'autre part par le potentiomètre d'avance.
- L'avance effective peut être limitée par la commande à cause des vitesses maximales configurées pour les axes participants.
- Une valeur F programmée pendant G95 est conservée dans la mémoire interne en cas de commutation sur G93 ou G94, et redevient active lors d'une nouvelle commutation sur G95.
- Le mot F défini dans les paramètres machine devient actif (par défaut : F0) suite à une montée en régime ou à une remise à zéro.

5.2.39 Programmation incrémentale de la vitesse "G95(...)" avec adaptation de l'accélération

Description :

Ici, l'accélération au sein du bloc G95(...) est adaptée de sorte que la vitesse ou vitesse de rotation qui en résulte ne soit atteinte qu'à la fin du bloc. Ceci entraîne un comportement d'accélération très doux.

Les points suivants s'appliquent ici :

- L'avance qui en résulte à la fin du bloc a un effet modal pour les blocs suivants.
- Selon G70/G71, l'unité de l'avance incrémentale est le pouce/tr ou le mm/tr.
- L'avance est influencée par l'avance Override en fonction de OvrEna/OvrDis.

Syntaxe :

G95 (DF<valeur F>)	Programmer de façon incrémentale la vitesse sur trajectoire (Delta Feed)
G95 (DS1=<valeur S>)	Programmer de façon incrémentale la vitesse de rotation de la broche (Delta Speed) pour la broche 1
G95 (DF<valeur F>, DS7=<valeur S>)	
	Programmer la vitesse sur trajectoire et la vitesse de rotation de la broche 7 de façon incrémentale
G95 (DSA2=<axe>, DS2=<valeur S>)	
	Programmer de façon incrémentale la vitesse de rotation d'un axe sans fin (Delta Speed Axis)

Fonctions CN avec syntaxe conforme DIN 66025 (extensions incl.)

avec	
<valeur F>	Vitesse sur trajectoire incrémentale en mm/tr ou en pouces/tr. Les valeurs positives augmentent et les valeurs négatives réduisent la vitesse sur trajectoire actuellement effective.
<valeur S>	En fonction de l'affectation des axes, la valeur S signifie : • Vitesse de rotation incrémentale DSi de la broche du canal de même index (sans indication d'axe) • Vitesse de rotation incrémentale DSi de l'axe sans fin de même index DSAi • Facteur de couplage incrémental DSi de l'axe sans fin de même index couplé en avance DSAi (voir aussi AFC) Les valeurs positives augmentent et les valeurs négatives réduisent la vitesse actuellement effective. Il est possible de programmer simultanément un maximum de 8 valeurs S (DS1 à DS8). La programmation incrémentale de la vitesse de rotation pour les axes sans fin peut entraîner une inversion de la direction.
<axe>	Désigne un axe sans fin asynchrone au mode d'asservissement en vitesse de rotation. Il est possible de prédéfinir simultanément des valeurs S incrémentales pour un maximum de 8 axes (DSA1 à DSA8).

Tab. 5-68: Syntaxe de la programmation de vitesse incrémentale avec G95

Exemple :

```

N10 G95 (DF=-0.5) G1 X100 ; Réduire l'avance dégressive de 0,5 mm/tr
                                ; jusqu'à la fin du bloc

...

N20 M3 S1=500 ; Démarrer la broche du canal S1 à 500 tr/min
N30 G95 (DS1=200) X10 ; Augmenter la vitesse de rotation S1 à 700 tr/min
F0.2 ; jusqu'à la fin du bloc

...

N70 B2=S (100) B3=S (50) ; Démarrer les axes sans fin B2 et B3
                                ; au mode d'asservissement en vitesse de rotation

                                ; Augmentation de la vitesse de rotation à
                                ; 200 tr/min pour B2,
                                ; réduction de la vitesse de rotation à 30 tr/min
                                ; pour B3
                                ; jusqu'à la fin du bloc

N80 G95 (DSA1=B2, DS1=100, DSA2=B3, DS2=-20) X10
...

```

Fonctions CN avec syntaxe conforme DIN 66025 (extensions incl.)

```

N100 B1=S(500) ; Démarrer l'axe sans fin B1 à 500 tr/min
N110 G1 G95(FRA=B1) F0.3 ; Référence de l'avance : 0,3 mm/tr de B1

; Couplage de l'avance : 0,5 tr de B2/
; tour de B1
N120 AFC(FA=B2,CF=0.5) B2=S(1)

; Augmente F à 0,4 mm/tr de B1 et
; CF à 0,7 tr de B2/tour de B1
; jusqu'à la fin du bloc
N130 G95(DF=0.1,DSA1=B2,DS1=0.2)X10

```

- Particularités et restrictions :**
- Les valeurs limites existantes pour l'accélération ou la décélération sont surveillées. De ce fait, la vitesse finale qui en résulte ne peut éventuellement être atteinte que dans le bloc suivant.
 - Une accélération régulière est uniquement atteinte dans le bloc G95(DF...) avec un Override de 100 %.
 - L'ordre SPLIT constitue la rampe d'avance programmée dans chaque segment de bloc.
 - En cas d'interruption anticipée du bloc ou si une spécification de course est manquante, la commande active immédiatement les données de vitesse de rotation incrémentales.

5.2.40 Programmation de l'avance par rotation d'axe rond "G95(FRA=<axe>)"

Description : Génère la référence de l'avance en [mm ou pouces/360°] pour un axe rond du système ou pour des axes linéaires sans fin [mm ou pouces/intervalle modulo] et modifie l'avance relative par rapport à la dernière valeur effective active. La référence de l'axe rond conserve son effet modal.

Les points suivants s'appliquent ici :

- Après la mise en service, la référence de la broche principale est générée avec G95.
- La fonction G95 avec l'axe de référence programmé a un effet modal.
- L'avance est influencée par l'avance Override en fonction de OvrEna/OvrDis.

Syntaxe :

G95(FRA=<nom axe système>)	
	Générer la référence de l'avance pour l'axe sans fin
G95(FRA=MSP)	Générer la référence de l'avance pour la broche principale (MainSp)

Tab. 5-69: Syntaxe de la programmation de l'avance pour chaque tour d'axe rond G95

Exemple :

```

N10 Z3=S(1000) ; Définition de la vitesse de rotation Z3 à
                1000 tr/min
N30 G95(FRA=Z3) F1.0 ; Référence de l'avance pour l'axe rond Z3

```

Fonctions CN avec syntaxe conforme DIN 66025 (extensions incl.)

N40 G1 X100

; Avance : 1000 mm/min à 100 % OVR

N50 G95 (FRA=MSP)

; Avance concernant la broche principale active

- Particularités et restrictions :**
- Seuls un axe sans fin ou la broche de référence actuelle (MainSp) sont autorisés comme axe de référence (FRA).
 - L'axe de référence ne doit en aucun cas être un axe du canal.

5.2.41 Vitesse de coupe constante "G96", programmation directe de la vitesse de rotation "G97"

Description : La commande interprète les mots *S* lors de l'usinage rotatif en tant que

- vitesse de coupe constante de l'outil (G96) ou
- rotation de l'axe de rotation de la pièce à usiner (G97).

Si la vitesse de rotation de la broche est définie, la vitesse de coupe dépend de la distance de l'outil de coupe par rapport à l'axe de rotation de la pièce à usiner.

Afin de compenser cet état de fait, la fonction G96 modifie automatiquement la vitesse de rotation de l'axe de rotation en fonction de la distance entre l'outil de coupe et l'axe de rotation de la pièce à usiner :

- avec G71 :

$$S_{act} \left[\min^{-1} \right] = \frac{S_{G96} \times 1000}{2 \times \pi \times \left| \text{Position}(\text{Coordonnée, référentiel}) - \text{Correction de l'outil} \right|} \left[\frac{m \times 1000}{\min \times mm} \right]$$

Fig. 5-29: G71

- avec G70 :

$$\left[\frac{feet \times 12}{\min \times inch} \right]$$

Fig. 5-30: G70

Par défaut, la commande numérique calcule la distance dans le système de coordonnées de la machine (référentiel MCS) :

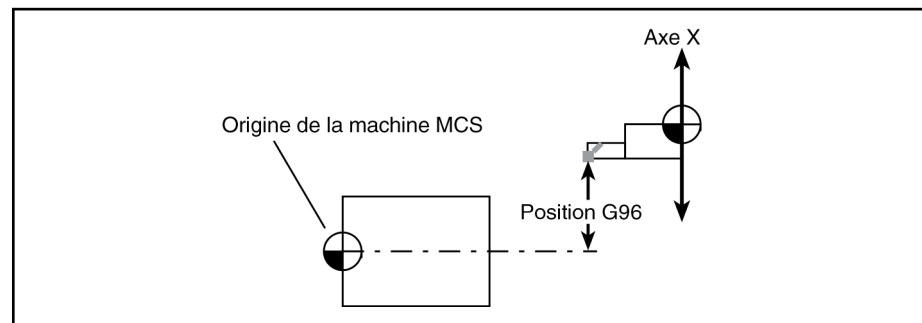


Fig. 5-31: Référentiel MCS

Si la broche doit être positionnée entre différents points de logement de l'outil, G96 peut intégrer des décalages d'origine correspondants qui sous-tendent un système de coordonnées local (référentiel LCS) :

Fonctions CN avec syntaxe conforme DIN 66025 (extensions incl.)

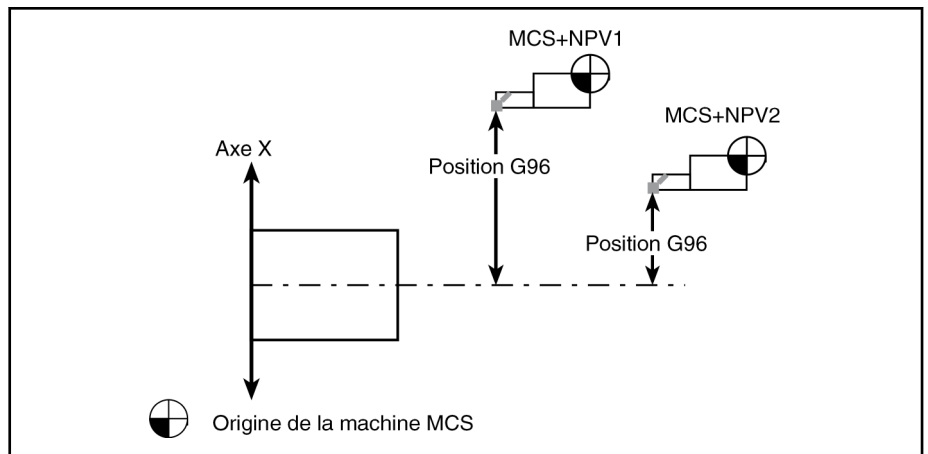


Fig. 5-32: *Référentiel LCS*

Pour les outils qui ne sont pas disposés parallèlement à la broche, il est également possible de tenir compte de décalages et de rotations (placements) de l'origine de la pièce à usiner (référentiel PCS) :

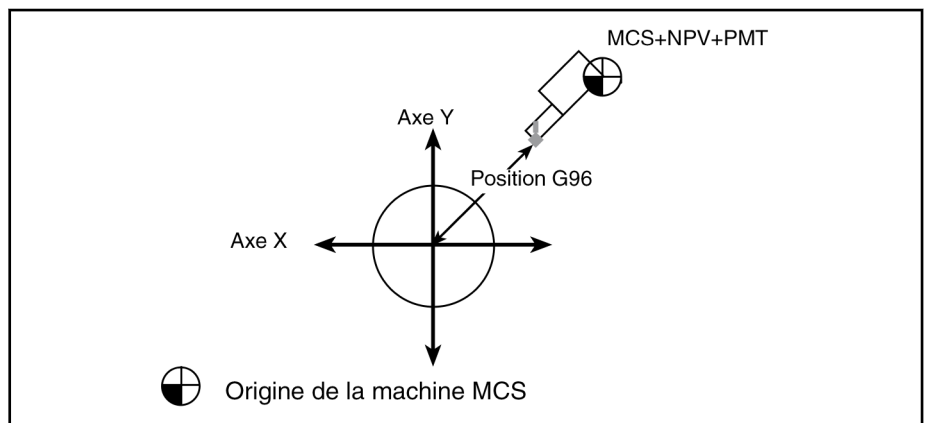


Fig. 5-33: *Référentiel PCS*

La vitesse de coupe sur la pièce à usiner n'est donc plus influencée par la distance entre l'outil de coupe et l'axe de rotation de la pièce à usiner.

Si ce comportement n'est pas nécessaire ou indésirable, il convient d'utiliser la fonction G97. La vitesse de l'axe de rotation de la pièce à usiner est alors définie exclusivement par le mot S programmé.

Les points suivants s'appliquent ici :

- L'axe de référence pour la distance entre la pointe de coupe et l'axe de rotation de la pièce à usiner est défini à l'aide du paramètre machine 7010 00110 (valeur par défaut).
- Cet axe de référence peut être reprogrammé lors de l'exécution.
- Le système de coordonnées de référence (point d'action) pour l'axe de référence est prédéfini par le paramètre machine 7010 00120.

Les alternatives suivantes sont possibles :

- PCS :
position dans le système de coordonnées du programme
- LCS :
position dans le système de coordonnées local de la machine (décalée par DO)

Fonctions CN avec syntaxe conforme DIN 66025 (extensions incl.)

- MCS :
position dans le système de coordonnées de la machine (valeur par défaut)
- Le point d'action de l'axe de référence peut être reprogrammé en cours d'exécution.
- *La vitesse de coupe programmée est interprétée en :*
 - m/min, si G71 est active.
 - pieds/min, si G70 est actif (1 pied = 12 pouces).
- En plus de l'Override de la broche, les fonctions de limitation de la vitesse de rotation (SMin, SMax, voir [chap. 6.111 "Limitation de la vitesse de rotation SMin, SMN, SMax, SMX" à la page 393](#)) sont également effectives.

Syntaxe :

G96 { ({ <axe de référence> { , <point d'action> } }) } ... S<i>=<V>	
	Les mots S des broches programmées dans le bloc G96 sont interprétés lors de l'usinage rotatif en tant que vitesse de coupe de consigne sur l'outil. Toutes les autres broches retombent à la programmation directe de la vitesse de rotation !
G96	Les derniers réglages programmés deviennent actifs ; si aucun réglage n'a été effectué, les valeurs par défaut sont actives.
G96 () ...	Les valeurs par défaut pour l'axe de référence et le point d'action sont réactivées.
G97 ... S<i>= <vitesse de rotation>	
	Seules les broches dont les mots S sont programmés dans le bloc G97 retombent à la programmation directe de la vitesse de rotation. Pour ces broches, les mots S programmés sont interprétés en tant que vitesse de rotation pour l'axe de rotation de la pièce à usiner.
G97	Toutes les broches retombent à la programmation directe de la vitesse de rotation.
avec	
<axe de référence>	Nom logique ou physique de l'axe de référence. La programmation est conservée jusqu'à ce qu'une nouvelle valeur ou la valeur par défaut soit activée.
<point d'action>	Système de coordonnées de l'axe de référence : PCS, LCS ou MCS La programmation est conservée jusqu'à ce qu'une nouvelle valeur ou la valeur par défaut soit activée.
<i>	Index de la broche. 1 : première broche ; 2 : deuxième broche, etc.
<V>	Vitesse de coupe de la broche concernée en m/min ou pieds/min.
<vitesse de rotation>	Vitesse de rotation de la broche concernée en tr/min.

Tab. 5-70: Syntaxe G96, G97

Fonctions CN avec syntaxe conforme DIN 66025 (extensions incl.)

Exemple : (2 broches sont configurées dans le système) :

N10 G71 S1=500 S2=500 :	Définir pour la première fois les vitesses de rotation de consigne des deux broches dans le programme.
N60 G96 S1=50 : : :	Activer G96 pour la 1re broche avec une vitesse de coupe de 50 m/min. La 2e broche tourne encore à la vitesse de rotation programmée. G96 est active
N100 S1=30 : : :	Réduire la vitesse de coupe de la 1re broche à 30 m/min. La 2e broche tourne encore à la vitesse de rotation programmée. G96 est active
N140 G96(Y,LCS) S2=100 : : : : :	L'axe de référence est Y, le point d'action est le système de coordonnées local (LCS). Activer la vitesse de coupe de 100 m/min pour la 2e broche. La 1re broche retombe à la vitesse de rotation programmée. G96 est active
N180 G97 : :	Toutes les broches retombent – si elles ne l'ont pas fait déjà - à la vitesse de rotation programmée. G97 est active.

Particularités et restrictions :

- Il est possible de commuter plusieurs broches dans un même bloc en programmant les mots S correspondants les uns après les autres (exemple : G96 S1=100 S2=1000).
- Pour modifier la vitesse de coupe d'une broche déjà commutée sur G96 au cours du déroulement du programme, il suffit de reprogrammer le mot S de la broche correspondante.
- La vitesse de coupe actuellement active d'une broche reste enregistrée en interne même après la commutation sur G97. Elle redevient active dès que la broche correspondante est "de nouveau commutée" sur G96.
- Lors de la commutation de G96 à G97, la commande accepte la vitesse de rotation actuelle en tant que nouvelle vitesse de rotation de consigne pour toutes les broches dont le mot S n'est pas programmé dans le bloc G97.
- Un changement de rapport de réduction éventuellement souhaité doit être exécuté avant l'activation de G96.
- Lorsque G96 est active, la fonction "Commutation automatique des rapports de réduction" ne change pas le rapport de réduction.
- Lorsque G96 est active, la commande vérifie s'il existe une "prise" de l'axe de référence sur le point d'action actif. Si ce n'est pas le cas (par ex. si l'axe quitte le canal), une erreur d'exécution est émise.

Fonctions CN avec syntaxe conforme DIN 66025 (extensions incl.)

5.2.42 Correction 3D du rayon d'outil "G140, G141, G142"

Description : La fonction déplace un outil à symétrie de révolution vers la gauche ou vers la droite de la trajectoire programmée tout en guidant son point d'intervention dans l'avance active sur la trajectoire programmée.

Lors des mouvements d'orientation purs, la position actuelle du point d'intervention reste fixe et le TCP se déplace.

Au niveau des angles extérieurs, le parcours de la trajectoire est fermé par un bloc intermédiaire généré automatiquement (arc de cercle). Au niveau des angles intérieurs, le point d'intersection est calculé.

Pour la correction, la commande tient compte :

- du rayon de l'outil "r" pour l'arête de coupe active (ED, DCT) et la correction D active (D, DcTSeI, DCS, DCT) ;
- d'une profondeur de plongée optionnelle "d" (voir syntaxe `INSDEP`) ;
- d'un mètre optionnel par rapport à la trajectoire, dont l'effet s'ajoute au rayon de l'outil (voir syntaxe `COFFS`) ;
- d'un taux de variation optionnelle du mètre (voir syntaxe `DCOFFS`).

Le schéma suivant illustre le principe appliqué :

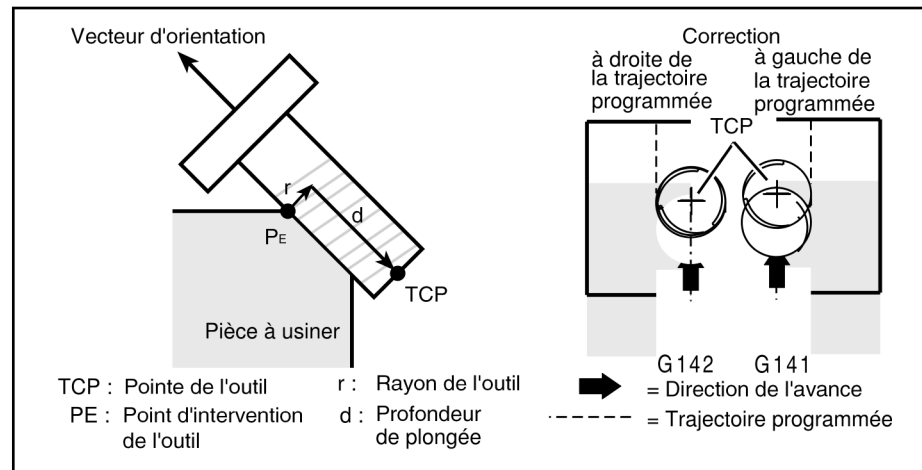


Fig. 5-34: Correction 3D du rayon d'outil

Quand les fonctions G141/G142 sont actives, la profondeur de plongée (`INSDEP`), le mètre (`COFFS`), la longueur d'outil ou le rayon de la fraise sont modifiés immédiatement et progressivement (par splines).

Le taux de variation (`DCOFFS`) du mètre agit uniquement lorsque les fonctions G141/G142 sont actives, mais pas dans les blocs d'activation de la correction. Les modifications du taux de variation prennent effet immédiatement et le mètre passe linéairement, en l'espace d'un bloc de déplacement, de sa valeur initiale au début du bloc à sa valeur de fin obtenue par la $\text{<valeur initiale> + <longueur de la trajectoire programmée> * <taux de variation du mètre>}$ à la fin du bloc. La valeur de fin devient la valeur initiale dans le bloc suivant, sauf si elle est écrasée dans le bloc suivant par la programmation de `COFFS`.



Pour une description détaillée de la fonction, voir la documentation "Description des fonctions IndraMotion MTX".

Fonctions CN avec syntaxe conforme DIN 66025 (extensions incl.)



Dans certaines circonstances, les valeurs de correction sont saisies ou obtenues immédiatement sans programmation d'un mouvement de déplacement à part. Ceci risque d'endommager la pièce à usiner ou l'outil.

C'est pourquoi toutes les informations mentionnées dans ce chapitre et dans la documentation "Description des fonctions IndraMotion MTX" doivent impérativement être respectées.

Syntaxe :

G140	Correction 3D du rayon d'outil DÉSACTIVÉE (État de mise en service après la montée en régime de la commande) La correction (rayon, profondeur de plongée) est désactivée par interpolation avec un éventuel mouvement de déplacement programmé (via les splines de 3 ^e ordre). G140 n'a aucune influence sur la correction active de la longueur d'outil.
G141	Correction 3D du rayon d'outil à gauche de la trajectoire ACTIVÉE (vue dans le sens d'usinage avec des valeurs de correction positives). Il est possible de programmer des mouvements de déplacement et d'orientation dans un même bloc. Le rayon et la profondeur de plongée sont utilisés si l'une des coordonnées tridimensionnelles actives est programmée dans le même bloc. <i>Dans le cas contraire, la commande applique la correction :</i> • seulement avec le bloc de déplacement suivant dans la mesure où G140 était active au préalable. • immédiatement (dans l'avance active), si G141/G142 était active au préalable.
G142	Correction 3D du rayon d'outil à droite de la trajectoire ACTIVÉE Identique à G141 pour le reste
INSDEP<PP>	Définit la profondeur de plongée <PP>. A un effet modal. État de mise en service après montée en régime de la commande : 0
COFFS<KA>	Définit le métré <KA>. A un effet modal et s'ajoute au rayon d'outil "r". État de mise en service après montée en régime de la commande : 0
DCOFFS<DA>	Définit le taux de variation <DA> du métré. A un effet modal et modifie le métré en continu le long de la trajectoire. État de mise en service après montée en régime de la commande : 0

Tab. 5-71: Syntaxe G140 - G142

Particularités et restrictions :

- Les fonctions G140, G141 et G142 sont des fonctions modales qui se révoquent réciproquement.

Fonctions CN avec syntaxe conforme DIN 66025 (extensions incl.)

- Activation et désactivation de la correction sur un contour quelconque (G2, G6, etc.)
- *Si G141 ou G142 est active, les fonctions suivantes sont **interdites** :*
 - G17 à G20 (changement de plan)
 - G70, G71 (commutation du système pouce/métrique)
 - G63 (taraudage sans mandrin de compensation)
 - G74 (approche des coordonnées du point de référence)
 - G75 (déplacement contre le palpeur de mesure)
 - G76 (approche de la position fixe des axes machine)
 - SetPos (initialisation de la position du programme)
 - G54.x à G59.x (décalages d'origine)
 - G154.x à G159.x (plan incliné)
 - G40 à G42 (correction du rayon d'outil)
 - Coord() (activer/désactiver/commuter la transformation d'axe)
 - Transferts des axes participant aux coordonnées tridimensionnelles actuelles via la transformation d'axes/de coordonnées.
- Aucun contrôle de collision n'est effectué si G141 ou G142 est active.
- Si le programme se termine sans M30, G140 doit être active à cet instant. Les valeurs actuelles de la profondeur de plongée, du métré et de son taux de variation sont conservées.
- G140 est activée automatiquement après une remise à zéro, une remise à zéro du système ou M30. Les valeurs actuelles de la profondeur de plongée, du métré et de son taux de variation sont paramétrées sur 0.

5.2.43 Placement : plan incliné "G151 à G159.5"

Description : Le placement "Plan incliné" peut décaler **et** orienter librement le système de coordonnées de la pièce à usiner dans l'espace. Le plan incliné a un effet sur les coordonnées avec les significations "X", "Y" et "Z" sur le canal correspondant.



Les désignations d'axes "X", "Y" et "Z" utilisées ci-dessous se rapportent aux axes ayant les **significations** X, Y et Z.

Fonctions CN avec syntaxe conforme DIN 66025 (extensions incl.)

Comme il existe 3 degrés de liberté pour l'orientation, chaque orientation peut être représentée par 3 rotations de base successives :

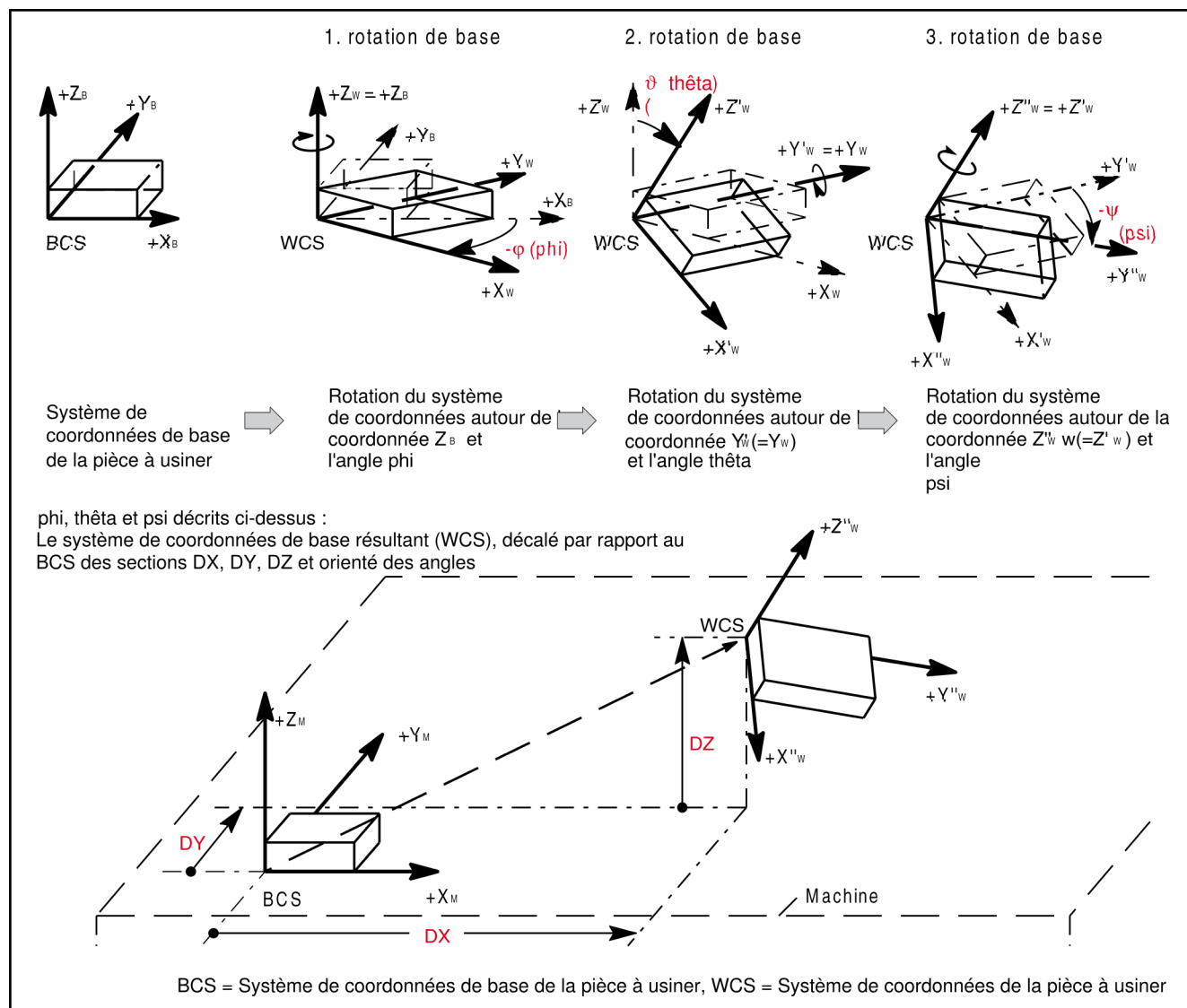


Fig. 5-35: Placement "Plan incliné"

La fonctionnalité résultante du "Plan incliné" est influencée par :

- 5 banques (1 à 5) qui agissent de façon additive les unes par rapport aux autres.
Elles permettent de générer jusqu'à 5 systèmes de coordonnées pour la pièce à usiner reposant les uns sur les autres.
- 8 jeux alternatifs par banque.
Chaque jeu contient toutes les données de décalage et d'orientation nécessaires pour un système de coordonnées de la pièce à usiner.
Au sein d'une banque, un seul jeu à la fois peut être actif, c.-à-d. que les jeux d'une banque se révoquent réciproquement.

L'effet du plan incliné s'ajoute au placement "Correction de la position de la pièce à usiner" et se trouve donc derrière la correction de la position de la pièce à usiner dans la "chaîne de calcul" :

Fonctions CN avec syntaxe conforme DIN 66025 (extensions incl.)

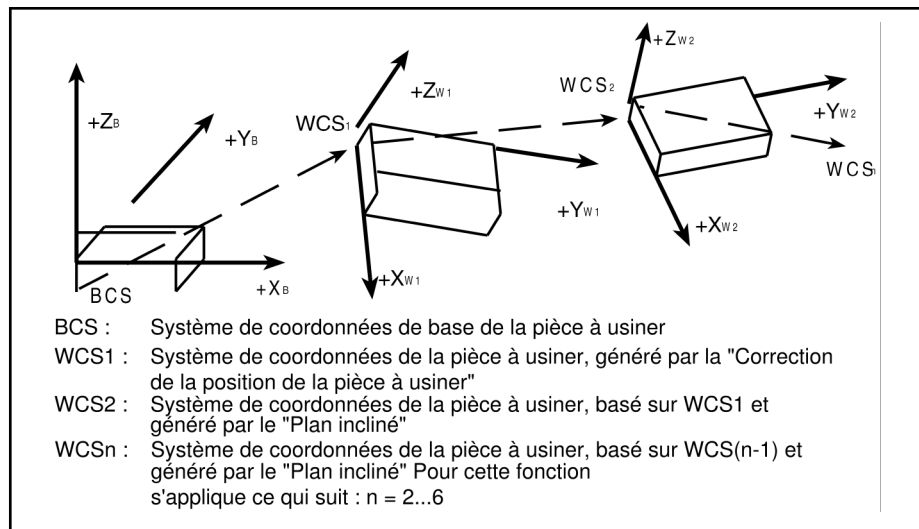


Fig. 5-36: Placement "Correction de la position de la pièce à usiner"

Syntaxe : Toutes les données nécessaires au décalage et à l'orientation d'un système de coordonnées de la pièce à usiner constituent un "jeu".

Les jeux peuvent être utilisés comme suit :

- Ils peuvent être activés implicitement dans le programme pièce d'après l'orientation actuelle de l'outil (G151 à G151.5).
- Ils peuvent être programmés dans le programme pièce en tant que paramètres de fonction (G152 à G152.5).
- Ils peuvent être inscrits dans un tableau de placement (G154 à G159.5). (voir [chap. 6.81 "Activation des tableaux de placement PmTsel, PMS" à la page 346](#))

Un tableau de placement peut contenir un maximum de 30 jeux (5 banques, 6 jeux par banque).

Activation implicite dans le programme pièce (G151 à G151.5) :

Si le plan incliné est perpendiculaire à l'outil, la position et l'orientation du nouveau système de coordonnées de la pièce à usiner à sous-tendre sont déterminées directement par le TCP et l'orientation de l'outil.

Fonctions CN avec syntaxe conforme DIN 66025 (extensions incl.)

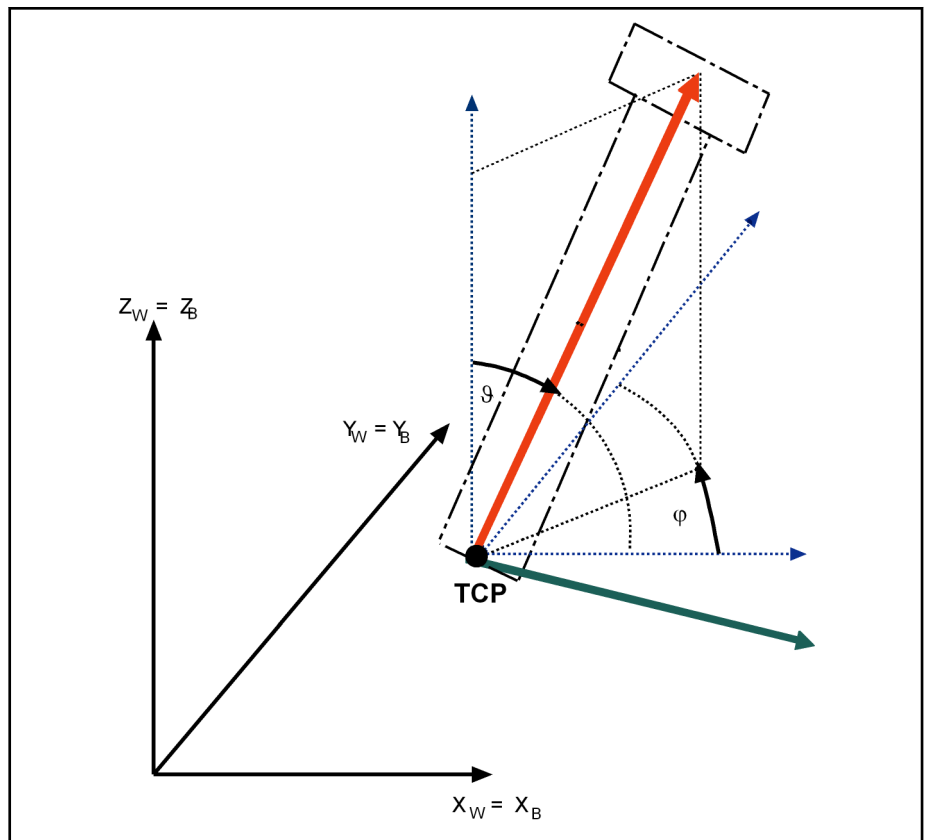


Fig. 5-37: Plan incliné perpendiculaire à l'outil

Une fois le plan incliné perpendiculaire à l'outil activé, l'origine du nouveau système de coordonnées de la pièce à usiner se trouve dans le TCP de l'outil.

Le nouvel axe Z_w se trouve en direction de l'axe longitudinal de l'outil.

La nouvelle coordonnée Y_w est parallèle au plan sous-tendu par les coordonnées de base précédentes X_B et Y_B .

Le nouvel axe X_w est obtenu d'après la définition mathématique issue du produit vectoriel des deux coordonnées restantes $X_w = Y_w \times Z_w$.

Fonctions CN avec syntaxe conforme DIN 66025 (extensions incl.)

La figure suivante montre la configuration après l'activation du plan incliné perpendiculaire à l'outil :

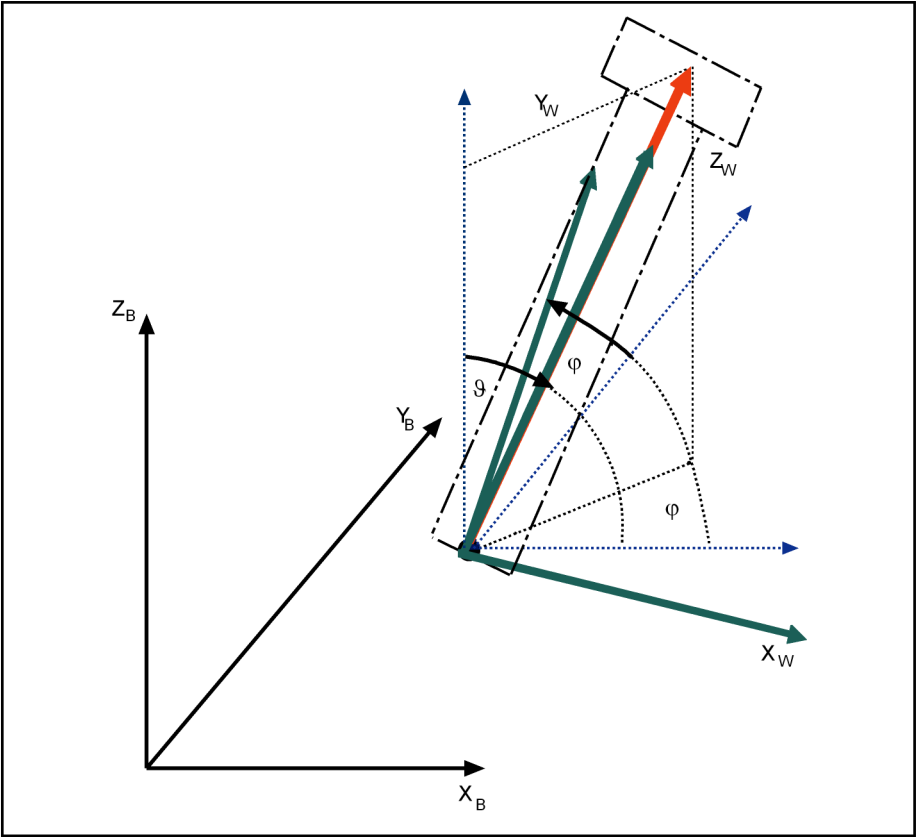


Fig. 5-38: Configuration après activation du plan incliné perpendiculaire à l'outil

La programmation optionnelle de l'angle de rotation psi permet de faire pivoter l'orientation du système de coordonnées autour de la nouvelle coordonnée Z_w.

G151.<banque>{(<angle3>)}	Plan incliné de la banque voulue ACTIVÉ Les valeurs de décalage sont obtenues à partir de la position actuelle de la pointe de l'outil (TCP) dans le système de coordonnées actuel du programme (PCS). Les angles de rotation sont le résultat direct de l'orientation actuelle de l'outil. Si aucune transformation d'axe orientable n'est active, les valeurs des angles d'Euler sont paramétrées sur 0.
Format abrégé : G151	Correspond à G151.1
G153.<banque>	Plan incliné de la banque programmée DÉSACTIVÉ
G153	Plan incliné DÉSACTIVÉ (désactiver toutes les banques)
avec	
<banque>	Numéro de la banque voulue Plage de valeurs : 1 à 5
<angle3>	Angle de rotation psi autour de la nouvelle coordonnée Z _w Plage de valeurs : 0 <= <angle3> <= 360 degrés

Tab. 5-72: Syntaxe G151, G153

Fonctions CN avec syntaxe conforme DIN 66025 (extensions incl.)

Programmation directe sur la ligne du programme en tant que paramètre (G152 à G152.5) :

En cas de programmation directe du plan incliné avec G152 à G152.5, la position et l'orientation du nouveau système de coordonnées de la pièce à usiner à sous-tendre sont programmées directement par le paramètre de position de la fonction.

G152.<banque> ({<décalage X_W >},{<décalage Y_W >},{<décalage Z_W >},{<angle1>},{<angle2>},{<angle3>}})	
	Plan incliné de la banque voulue ACTIVÉ, avec données de décalage/d'orientation programmées.
Format abrégé : G152	Correspond à G152.1
G153.<banque>	Plan incliné de la banque programmée DÉSACTIVÉ
G153	Plan incliné DÉSACTIVÉ (désactiver toutes les banques)
avec	
<banque>	Numéro de la banque voulue Plage de valeurs : 1 à 5
<décalage X_W >	Valeur de décalage dans la direction de la coordonnée principale
<décalage Y_W >	Valeur de décalage dans la direction de la coordonnée secondaire
<décalage Z_W >	Valeur de décalage dans la direction de la coordonnée normale
<angle1>	Angle de rotation autour de la coordonnée Z Plage de valeurs : $0 \leq \text{<angle1>} < 360$ degrés
<angle2>	Angle de rotation autour de la coordonnée Y' Plage de valeurs : $0 \leq \text{<angle2>} < 180$ degrés
<angle3>	Angle de rotation autour de la coordonnée Z'' Plage de valeurs : $0 \leq \text{<angle3>} < 360$ degrés

Tab. 5-73: Syntaxe G152, G153

Programmation associée à un tableau de placement :

Activer le tableau de placement voulu (voir [chap. 6.81 "Activation des tableaux de placement PmTsel, PMS" à la page 346](#)).

Programmer la fonction requise :

G154.<banque>	Plan incliné du jeu 1 programmé de la banque ACTIVÉ
G155.<banque>	Plan incliné du jeu 2 programmé de la banque ACTIVÉ
G156.<banque>	Plan incliné du jeu 3 programmé de la banque ACTIVÉ

Fonctions CN avec syntaxe conforme DIN 66025 (extensions incl.)

G157.<banque>	Plan incliné du jeu 4 programmé de la banque ACTI-VÉ
G158.<banque>	Plan incliné du jeu 5 programmé de la banque ACTI-VÉ
G159.<banque>	Plan incliné du jeu 6 programmé de la banque ACTI-VÉ
Format abrégé : G152 - G159	Correspond à G154.1 - G159.1
G153.<banque>	Plan incliné de la banque programmée DÉSACTIVÉ
G153	Plan incliné DÉSACTIVÉ (désactiver toutes les banques)
avec	
<banque>	Numéro de la banque voulue Plage de valeurs : 1 à 5

Tab. 5-74: Syntaxe G153 à G159

Exemple : Perçage de trous obliques avec G151

:	
N05 Coord(1)	La transformation d'axe 1 est activée dans le programme, puis l'outil est positionné et orienté en différentes positions dans l'espace. Un placement perpendiculaire à l'outil est activé pour chacune de ces positions, puis une percée est exécutée à l'aide du cycle d'alésage standard MTX. La profondeur de perçage (DT), le plan de sécurité (SL) et la hauteur de dégagement (RL) sont transmis au sous-programme en tant que paramètres.
N10 G1 x0 y0 z100 phi90 theta1 F2000	
N15 G151	
N20 P STC_G81(SL90, DT80, RL100) F500	
N25 G153	
N30 phi90 theta10	
N35 G151	
N40 P STC_G81(SL90, DT80, RL100) F500	
N45 G153	
N50 theta45	
N55 G151	
N60 P STC_G81(SL90, DT80, RL100) F500	
N65 G153	
N70 phi0	
N75 G151	
N80 P STC_G81(SL90, DT80, RL100) F500	
N85 G153	
N90 G2 I100 J0 ROTAX(0,0,1) O(-360)	
N95 G151	
N100 P STC_G81(SL90, DT80, RL100) F500	
N105 G153	
N110 G90G1	
N120 Coord(0)	
M30	

Fonctions CN avec syntaxe conforme DIN 66025 (extensions incl.)

Exemple : Programmation de placement directe avec G152

:	
N40 G152.1 (100,0,0,90)	Programmer le placement pour la banque 1
:	Aucun mouvement de déplacement
:	Le point d'origine est décalé de 100 mm en direction de X et le WCS tourné de 90 degrés autour de la coordonnée Z.
:	
:	
N180 G153	Désactiver toutes les banques (désactiver complètement le plan incliné)

Exemple : Activer le placement par tableau : G154 - G159.5

N40 PMS(Tab1)	Activer le tableau de placement "Tab1"
N50 G154.1	Activer la banque 1 avec le jeu 1
:	Aucun mouvement de déplacement
N90 G154.2 X1 Y2 Z3	Activer la banque 2 avec le jeu 1
:	(effet additif à la banque 1)
:	Approcher la position programmée P(1,2,3) dans le WCS résultant.
N120 G153.2	Désactiver la banque 2 (la banque 1 reste active)
:	
N180 G153	Désactiver toutes les banques (désactiver complètement le plan incliné)

- Particularités et restrictions :**
- Dans la mesure où une transformation d'axe est active, le plan incliné ne doit pas être programmé dans le même bloc qu'un mouvement de déplacement.
 - L'état de mise en service et le comportement en cas de remise à zéro sont configurés dans les paramètres machine 7060 00010 et 7060 00020.
 - Toutes les fonctions ont un effet modal au sein d'une banque (1 à 5) et se révoquent réciproquement.
 - L'activation et la désactivation du plan incliné interrompent la prévisualisation du bloc et ne doivent donc pas être programmées quand la correction de trajectoire de la fraise (G41/G42, voir [chap. 5.2.15 "Correction de trajectoire de la fraise G40, G41, G42" à la page 155](#)) est active.
 - Si la fonction "Plan incliné perpendiculaire à l'outil" est appelée, aucun contrôle de plausibilité n'est effectué pour cet appel. Cela signifie que c'est à l'utilisateur d'appeler la fonction dans la banque de placement correcte.
 - Actuellement, 2 axes au moins ayant une signification ("X", "Y", "Z") doivent exister sur un canal pour activer un placement Plan incliné sur un canal.

Fonctions CN avec syntaxe conforme DIN 66025 (extensions incl.)

5.3 Codes M

5.3.1 Interruption du programme (arrêt du programme) "M0, M00"

- Description :**
- Interrompt le programme CN.
 - Arrête les mouvements de la machine à la fin de l'exécution du bloc.
 - Émet le signal d'interface spécifique au canal "Arrêt programme M0".

L'état actuel du canal passe à "CN prête".

Une nouvelle activation de "Démarrage CN" reprend l'exécution du programme.



Pour interrompre un programme en fonction du signal d'interface spécifique au canal "Arrêt optionnel", voir M1/M01.

Syntaxe : M0 ou M00

Particularités et restrictions : Il est possible de programmer "Arrêt programme" dans le même bloc que d'autres fonctions CN. La fonction "Arrêt programme" est exécutée seulement après que toutes les autres fonctions programmées ont été appliquées.

5.3.2 Interruption conditionnelle du programme (arrêt cond. progr.) "M1, M01"

- Description :**
- Interrompt le programme CN.
 - Arrête les mouvements de la machine à la fin de l'exécution du bloc si le signal d'interface spécifique au canal "Arrêt optionnel" est actif.

L'état actuel du canal passe à "CN prête".

Une nouvelle activation de "Démarrage CN" reprend l'exécution du programme.

Syntaxe : M1 ou M01

Tab. 5-75: Syntaxe M1

Particularités et restrictions : Il est possible de programmer "Arrêt conditionnel du programme" dans le même bloc que d'autres fonctions CN. La fonction "Arrêt conditionnel du programme" est exécutée seulement après que toutes les autres fonctions programmées ont été appliquées.

5.3.3 Terminer le programme (fin du programme) "M2, M02, M30"

Description : Termine le programme.

Si le programme est en fait un sous-programme, les points suivants s'appliquent :

- La commande numérique émet la fonction auxiliaire (M2, M02 ou M30).
- La commande numérique retourne dans le programme appelant.
- La commande numérique poursuit le traitement du programme appelant.



Les états modaux modifiés dans le sous-programme ne sont pas réinitialisés !

Si le programme est un programme principal, les points suivants s'appliquent :

- La commande numérique fixe le signal d'interface spécifique au canal "Fin du programme M30".

Fonctions CN avec syntaxe conforme DIN 66025 (extensions incl.)

- La commande numérique retire le signal d'interface spécifique au canal "Programme en cours".
- Le cas échéant, la commande numérique désactive une "sélection automatique du rapport de réduction" (mais le rapport de réduction actuel reste sélectionné).
- La commande numérique passe à la "programmation directe de la vitesse de rotation" (G97).
- La commande numérique active tous les états définis dans le paramètre machine 7060 00020 "États de mise en service" pour un événement "M30".
- La commande numérique retourne au début du programme principal.
- La commande numérique attend un nouveau "Démarrage CN".

Réactiver "Démarrage CN" relance l'exécution du programme à partir du début.

Syntaxe :

M2 ou M02 ou M30

Tab. 5-76: Syntaxe M2

Particularités et restrictions :

La fonction doit être programmée sur une ligne de programme à part.



L'absence de définition des états de mise en service risque de causer des dommages !

Si certains états ou certaines fonctions sont nécessaires à la fin d'un programme principal, **il convient de s'assurer** que la chaîne d'initialisation de l'évènement "M30" est paramétrée correctement dans MP 7060 00020 !

La chaîne d'initialisation doit contenir toutes les fonctions utilisées pour placer la commande numérique à l'état requis/voulu à la fin d'un programme principal. Sachant cela, tenir compte du fait que les fonctions modales restent actives une fois le programme terminé !



Vous trouverez de plus amples informations sur les états de mise en service dans la documentation "Description des fonctions IndraMotion MTX".

5.3.4 Rotation à droite de la broche "M3, M103, M203", rotation à droite de la broche et liquide de refroidissement ACTIVÉS "M13, M113, M213"

Description :



La syntaxe décrite ne s'applique qu'aux réglages par défaut des paramètres machine 1040 001xx et 1040 002xx. Elle est librement configurable et peut donc être différente sur votre machine !

La documentation du constructeur de la machine-outil est donc toujours prioritaire.

Consultez votre responsable système pour savoir si les fonctions décrites ci-après sont également valables pour votre machine !

- Démarre une rotation à droite de la broche ("outil vers pièce à usiner" dans la direction du regard).
- Annule une régulation de position activée par "Orientation/Positionnement de la broche".
- Réserve la/les broche(s) correspondante(s) pour le canal actuel.

Fonctions CN avec syntaxe conforme DIN 66025 (extensions incl.)



Le mouvement de la broche démarre uniquement si un mot S ou SSPG supérieur à 0 a été programmé au préalable ou dans le même bloc (voir [chap. 5.4.6 "Programmation de la vitesse de rotation de la broche S, SSPG" à la page 215](#)).

Syntaxe :

M3	Agit sur le 1er groupe de broches
M103	Agit sur la 1re broche
M203	Agit sur la 2e broche
M13	Agit sur le 1er groupe de broches Liquide de refroidissement également activé
M113	Agit sur la 1re broche Liquide de refroidissement également activé
M213	Agit sur la 2e broche Liquide de refroidissement également activé

Tab. 5-77: Syntaxe M3, M13, M103, M113, M203 et M213

Particularités et restrictions :

- La fonction respective réserve automatiquement la/les broche(s) correspondante(s) pour le canal actuel. L'utilisation d'une ou plusieurs broches réservées par un autre canal ne redevient possible que lorsque la fonction "Arrêt broche" (voir [chap. 5.3.6 "Arrêt de la broche M5, M105, M205" à la page 207](#)) ou "SpAdmin" (SPA, voir [chap. 6.114 "Libération/Reprise d'une broche réservée SpAdmin, SPA" à la page 396](#)) est utilisée pour les broches concernées.
- La fonction correspondante est active jusqu'à ce qu'un nouvel état de mouvement soit programmé pour les mêmes broches (par ex. un autre sens de rotation, avec/sans liquide de refroidissement, "Arrêt broche" ou "Orientation broche").
- Des ordres concurrentiels entre des broches individuelles et des groupes de broches programmés dans un même bloc génèrent une erreur d'exécution
(exemple : M3 et M104 ne doivent pas être programmés dans le même bloc).
- Après un changement de rapport de réduction, la direction de rotation préalablement programmée de la broche est automatiquement réappliquée.

5.3.5 Rotation à gauche de la broche "M4, M104, M204", rotation à gauche de la broche et liquide de refroidissement ACTIVÉS "M14, M114, M214"

Description :



La syntaxe décrite ne s'applique qu'aux réglages par défaut des paramètres machine 1040 001xx et 1040 002xx. Elle est librement configurable et peut donc être différente sur votre machine !

La documentation du constructeur de la machine-outil est donc toujours prioritaire.

Consultez votre responsable système pour savoir si les fonctions décrites ci-après sont également valables pour votre machine !

- Démarre une rotation à gauche de la broche ("Outil vers la pièce à usiner" dans la direction du regard).

Fonctions CN avec syntaxe conforme DIN 66025 (extensions incl.)

- Annule une régulation de position activée par "Orientation/positionnement de la broche",
- Réserve la/les broche(s) correspondante(s) pour le canal actuel.



Le mouvement de la broche démarre uniquement si un mot S ou SSPG supérieur à 0 a été programmé au préalable ou dans le même bloc (voir [chap. 5.4.6 "Programmation de la vitesse de rotation de la broche S, SSPG" à la page 215](#)).

Syntaxe :

M4	Agit sur le 1er groupe de broches
M104	Agit sur la 1re broche
M204	Agit sur la 2e broche
M14	Agit sur le 1er groupe de broches Liquide de refroidissement également activé
M114	Agit sur la 1re broche Liquide de refroidissement également activé
M214	Agit sur la 2e broche Liquide de refroidissement également activé

Tab. 5-78: Syntaxe M4, M14, M104, M114, M204 et M214

Particularités et restrictions :

- La fonction respective réserve automatiquement la/les broche(s) correspondante(s) pour le canal actuel. L'utilisation d'une ou plusieurs broches réservées par un autre canal ne redevient possible que lorsque la fonction "Arrêt broche" (voir [chap. 5.3.6 "Arrêt de la broche M5, M105, M205" à la page 207](#)) ou "SpAdmin" (SPA, voir [chap. 6.114 "Libération/Reprise d'une broche réservée SpAdmin, SPA" à la page 396](#)) est utilisée pour les broches concernées.
- La fonction correspondante est active jusqu'à ce qu'un nouvel état de mouvement soit programmé pour les mêmes broches (par ex. un autre sens de rotation, avec/sans liquide de refroidissement, "Arrêt broche" ou "Orientation broche").
- Des ordres concurrentiels entre des broches individuelles et des groupes de broches programmés dans un même bloc génèrent une erreur d'exécution
(exemple : M3 et M104 ne doivent pas être programmés dans le même bloc).
- Après un changement de rapport de réduction, la direction de rotation de la broche préalablement programmée est automatiquement réappliquée.

5.3.6 Arrêt de la broche "M5, M105, M205"

Description :



La syntaxe décrite ne s'applique qu'aux réglages par défaut des paramètres machine 1040 001xx et 1040 002xx. Elle est librement configurable et peut donc être différente sur votre machine !

La documentation du constructeur de la machine-outil est donc toujours prioritaire.

Consultez votre responsable système si vous n'êtes pas sûr que les fonctions décrites ci-après sont également valables pour votre machine !

Fonctions CN avec syntaxe conforme DIN 66025 (extensions incl.)

- Arrête une ou plusieurs broches.
- Annule une éventuelle réservation des broches indiquées par le canal actif.
- Annule une régulation de position activée par "Orientation/Positionnement de la broche".

Syntaxe :

M5	Agit sur le groupe de broches 1
M105	Agit sur la 1re broche
M205	Agit sur la 2e broche

Tab. 5-79: Syntaxe M5, M105 et M205

- Particularités et restrictions :
- La fonction correspondante reste active jusqu'à ce qu'un nouvel état de mouvement soit programmé pour les mêmes broches (par ex. "Rotation à droite/gauche de la broche" ou "Orientation broche").
 - Des ordres concurrentiels entre des broches individuelles et des groupes de broches programmés dans un même bloc génèrent une erreur d'exécution
(exemple : M3 et M105 ne doivent pas être programmés dans le même bloc).

5.3.7 Orientation de la broche/Positionnement de la broche "M19, M119, M219"

Description :



La syntaxe décrite ne s'applique qu'aux réglages par défaut des paramètres machine 1040 001xx et 1040 002xx. Elle est librement configurable et peut donc être différente sur votre machine !
La documentation du constructeur de la machine-outil est donc toujours prioritaire.

Consultez votre responsable système si vous n'êtes pas sûr que les fonctions décrites ci-après sont également valables pour votre machine !

La broche est réservée pour le canal actif et positionnée en régulation de position sur une position définissable.

Le positionnement s'effectue

- à l'arrêt :
comme défini dans le paramètre S-0-0154 de l'entraînement.
- lors d'un mouvement rotatif actif :
en conservant le sens de rotation.

Syntaxe :

M19	Groupe de broches 1 : Positionner toutes les broches participantes sur leur angle de référence (= 0 degré)
M119	Positionner la 1re broche sur son angle de référence (= 0 degré)
M219	Positionner la 2e broche sur son angle de référence (= 0 degré)

Fonctions CN avec syntaxe conforme DIN 66025 (extensions incl.)

M19S<angle>	Groupe de broches 1 : Positionner toutes les broches participantes sur <angle>
M119 S1=<angle>	Positionner la 1re broche sur <angle>
M219 S2=<angle>	Positionner la 2e broche sur <angle>
<angle>	Position absolue voulue de la broche en degrés Plage de valeurs : $0^\circ \leq \text{Position de la broche} < 360^\circ$ Si une autre valeur est programmée en tant que position, elle est automatiquement convertie dans l'intervalle indiqué. Si la broche se trouve déjà à la position indiquée, aucun mouvement n'a lieu.

Tab. 5-80: Syntaxe M19, M119 et M219

Exemples :

N60 M19 :	Positionner toutes les broches du 1er groupe de broches sur 0 degré.
N70 M219 :	Positionner uniquement la 2e broche sur son angle de référence.
N80 M19 S180 :	Positionner toutes les broches du 1er groupe de broches sur 180 degrés.
N90 M119 S1=370 :	1re broche positionnée sur 10 degrés.
N95 M19 S1=10 S2=20 :	Dans la mesure où les 1re et 2e broches du 1er groupe de broches sont attribuées : 1re broche positionnée sur 10 degrés, 2e broche positionnée sur 20 degrés. Positionner toutes les autres broches du 1er groupe de broches sur leur angle de référence.

Particularités et restrictions :

- Le positionnement de la broche est uniquement pris en charge pour les broches Sercos.
- La fonction choisie ne doit pas être programmée dans le même bloc qu'une fonction de broche concurrentielle ("Rotation à droite/gauche de la broche", "Arrêt broche").
- Dans la mesure où le mot S est indispensable, il doit être programmé dans le même bloc.
- À la fin du processus de positionnement, la broche conserve la régulation de position. La régulation de position n'est annulée automatiquement que par "Rotation à droite/gauche de la broche" ou "Arrêt broche".
- Des ordres concurrentiels entre des broches individuelles et des groupes de broches programmés dans un même bloc génèrent une erreur d'exécution
(exemple : M3 et M119 ne doivent pas être programmés dans le même bloc).
- Si le signal d'interface "qSp_SValueSD" est défini, les positions sont définies via la donnée système "SysSpCmdData[n]/OriPos". La valeur programmée est ignorée.

Fonctions CN avec syntaxe conforme DIN 66025 (extensions incl.)

5.3.8 Sélection automatique du rapport de réduction "M40, M140, M240"

Description :



La syntaxe décrite ne s'applique qu'aux réglages par défaut des paramètres machine 1040 001xx et 1040 002xx. Elle est librement configurable et peut donc être différente sur votre machine !
La documentation du constructeur de la machine-outil est donc toujours prioritaire.

Consultez votre responsable système si vous n'êtes pas sûr que les fonctions décrites ci-après sont également valables pour votre machine !

La commande sélectionne automatiquement le rapport adapté parmi les rapports de réduction disponibles en fonction de la vitesse de rotation active.

Si les plages de vitesse de certains rapports se chevauchent, la commande choisira toujours le rapport inférieur (avec la vitesse de rotation supérieure du moteur).

Syntaxe :

M40	Activer la sélection automatique du rapport de réduction pour le 1er groupe de broches.
M140	Activer la sélection automatique du rapport de réduction pour la 1re broche.
M240	Activer la sélection automatique du rapport de réduction pour la 2e broche

Tab. 5-81: Syntaxe M40, M140 et M240

Particularités et restrictions :

- La programmation de la vitesse de rotation "0" n'entraîne aucun changement du rapport de réduction.
- M40, M41-M44, M48, M140, M141-M144 et M148
sont des fonctions modales et se révoquent réciproquement.
- M240, M241-M244 et M248
sont des fonctions modales et se révoquent réciproquement.
- M30 désactive une sélection automatique du rapport de réduction éventuellement active (mais le rapport de réduction actuel reste sélectionné).
- La commutation des rapports de réduction est uniquement prise en charge pour les broches Sercos.
Pour les broches qui ne disposent pas d'une interface Sercos, seule la fonction auxiliaire est transmise à l'API.

5.3.9 Sélection manuelle du rapport de réduction "M41...44, M141...144, M241...244"

Description :



La syntaxe décrite ne s'applique qu'aux réglages par défaut des paramètres machine 1040 001xx et 1040 002xx. Elle est librement configurable et peut donc être différente sur votre machine !
La documentation du constructeur de la machine-outil est donc toujours prioritaire.

Consultez votre responsable système si vous n'êtes pas sûr que les fonctions décrites ci-après sont également valables pour votre machine !

Sélectionne le rapport de réduction correspondant.

Fonctions CN avec syntaxe conforme DIN 66025 (extensions incl.)

Si la vitesse de rotation programmée se situe en dehors de la plage des vitesses de rotation du rapport de réduction, alors la commande délivre la vitesse de rotation minimale ou maximale du rapport de réduction correspondant.

Syntaxe :

M4<rapport>	Activer le rapport de réduction <rapport> pour le 1er groupe de broches
M14<rapport>	Activer le rapport de réduction <rapport> pour la 1re broche
M24<rapport>	Activer le rapport de réduction <rapport> pour la 2e broche
avec	
<rapport>	Plage de saisie : 1 à 4

Tab. 5-82: Syntaxe M4, M14 et M24

Particularités et restrictions :

- M40, M41-M44, M48, M140, M141-M144 et M148 sont des fonctions modales et se révoquent réciproquement.
- M240, M241-M244 et M248 sont des fonctions modales et se révoquent réciproquement.
- La commutation des rapports de réduction est uniquement prise en charge pour les broches Sercos.
Pour les broches qui ne disposent pas d'une interface Sercos, seule la fonction auxiliaire est transmise à l'API.

5.3.10 Débrayer le rapport de réduction "M48, M148, M248"

Description :



La syntaxe décrite ne s'applique qu'aux réglages par défaut des paramètres machine 1040 001xx et 1040 002xx. Elle est librement configurable et peut donc être différente sur votre machine !

La documentation du constructeur de la machine-outil est donc toujours prioritaire.

Consultez votre responsable système si vous n'êtes pas sûr que les fonctions décrites ci-après sont également valables pour votre machine !

Débrayer le rapport de réduction.

Le réducteur tourne ensuite à vide.

Syntaxe :

M48	Débrayer le rapport de réduction pour le 1er groupe de broches
M148	Débrayer le rapport de réduction pour la 1re broche
M248	Débrayer le rapport de réduction pour la 2e broche

Tab. 5-83: Syntaxe M48, M148 et M248

Particularités et restrictions :

- M40, M41-M44, M48, M140, M141-M144 et M148 sont des fonctions modales et se révoquent réciproquement.
- M240, M241-M244 et M248 sont des fonctions modales et se révoquent réciproquement.

Fonctions CN avec syntaxe conforme DIN 66025 (extensions incl.)

- La commutation des rapports de réduction est uniquement prise en charge pour les broches Sercos.
Pour les broches qui ne disposent pas d'une interface Sercos, seule la fonction auxiliaire est transmise à l'API.

5.4 Avance et programmation de la vitesse de rotation

5.4.1 Définition de la vitesse de rotation ou de la vitesse modulo "<axe>=S<valeur>"

Description : Un axe sans fin asynchrone est accéléré à la vitesse indiquée (tr/min ou intervalles modulo/min). Le mode de vitesse reste actif jusqu'à ce qu'il soit remplacé par une nouvelle saisie ou annulé par un mouvement de positionnement. Le signe de la vitesse définit le sens de la rotation ou du mouvement. Lorsque la vitesse est atteinte, le signal d'interface iAx_ProgSpReach = 1 est défini.

Le mouvement de positionnement en vol du mode de vitesse arrête l'axe sur la prochaine position possible indiquée dans la direction du mouvement. La logique de positionnement de l'axe et les attributs de positionnement (DC, ACN etc.) sont sans effet ici. Une indication de l'avance avec FA=<avance> permet d'accélérer le positionnement, mais pas de le temporiser.

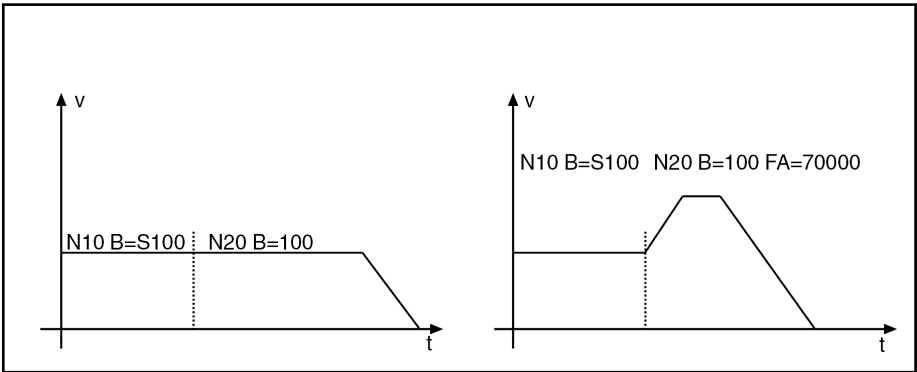


Fig. 5-39: Positionnement volant

Syntaxe :

<adresse d'axe>=S<valeur><valeur>	
avec	
<valeur>	Vitesse en tours/min ou intervalle modulo/min.

Tab. 5-84: Syntaxe de l'attribut S

Exemple :	N100 B=S1000	Accélérer l'axe B à 1000 tr/min et le déplacer de manière constante
	N200 B=-600	Ralentir l'axe B et l'accélérer dans le sens inverse à 600 tr/min.
	N300 B=100	Positionnement de l'axe B sur la position absolue 100° à une
	N300 B=100 FA=360000	vitesse de rotation ayant une vitesse de positionnement augmentée (correspondant à 1000 tr/min)
	N400 W=S100	Accélérer et déplacer de manière constante l'axe W linéaire à 100 plages modulo/min (200 mm/min) avec une plage modulo de 20 mm.

Fonctions CN avec syntaxe conforme DIN 66025 (extensions incl.)

- Particularités et restrictions :**
- La vitesse résultante dépend de l'Override spécifique à l'axe. La vitesse maximale est limitée par la vitesse maximale autorisée pour l'axe (AX/Dr[i]/Vel/MaxVel/) et par la moitié de la plage modulo (S-0-0103) par durée de cycle CN (SCS/ScsIff[1..2]/ScsCycTime).
 - Les attributs de programmation ACN, ACP, DC ou IC d'un mouvement de positionnement sont ignorés lorsque l'interpolation de la vitesse de rotation est ignorée.
 - La remise à zéro du système interrompt l'interpolation de la vitesse avec l'accélération d'axe autorisée.
 - Une limitation de secousses (ASO) peut uniquement être activée au début de l'interpolation de la vitesse de rotation. En cas de modifications de la vitesse de rotation ou d'un positionnement au vol, l'entrée est ignorée.

5.4.2 Adresse F "F"

Description : La commande interprète les adresses F d'après la fonction G actuellement active (G93, G94, G95) en tant que :

- temps d'interpolation en secondes
(voir G93, [chap. 5.2.32 "Programmation de temps G93" à la page 178](#))
- avance exprimée en mm/min ou pouces/min
(voir G94, [chap. 5.2.33 "Programmation de l'avance \(par min\) G94" à la page 179](#))
- avance en mm/tour
(voir G95, [chap. 5.2.38 "Programmation de l'avance \(par rotation\) G95" à la page 186](#))

Les points suivants s'appliquent ici :

F est une fonction modale pour G94 et G95.



Après une montée en régime ou une remise à zéro, la dernière valeur F active est éventuellement modifiée !

Après les événements mentionnés ci-dessus, la valeur F définie dans les paramètres machine 7060 00020 ou 7060 00010 devient active (valeur par défaut : F0). Il y est également indiqué si la fonction G93, G94 ou G95 doit être activée (valeur par défaut : G94).

C'est pourquoi il faut s'assurer que l'avance nécessaire est toujours programmée avant l'usinage !

Syntaxe :

F<valeur>	
avec	
<valeur>	Selon la fonction G active, est interprété en tant que durée d'interpolation, avance ou temporisation.

Tab. 5-85: Syntaxe F<valeur>

Particularités et restrictions : L'avance sur trajectoire programmée peut être écrasée à l'aide de la fonction "Avance d'essai".

Celle-ci est commandée via le signal IF du canal "qCh_TestFeed" (avance d'essai).

Fonctions CN avec syntaxe conforme DIN 66025 (extensions incl.)

5.4.3 Adresse F2 "F2"

Description Au mode G94, une deuxième avance F2 (avance2) est introduite en plus de la valeur F disponible jusqu'à présent (avance par défaut). L'avance2 peut être inférieure ou supérieure à l'avance par défaut en termes de valeur. La limitation de la vitesse ainsi que la préparation du profil de vitesse (Look-Ahead) reposent sur la valeur maximale des deux valeurs F pour leur prise en compte respective.

L'avance2 est activée et désactivée via l'API par le signal de sortie 13.2 (qCh_Feedrate2) de l'interface du canal. L'avance2 prend effet avec "qCh_Feedrate2 = 1". La modification de l'avance prend effet immédiatement, c.-à-d. que la vitesse sur trajectoire passe immédiatement à la nouvelle vitesse, dès le bloc actif.

La commande numérique signale à l'API que l'avance2 est activée par l'intermédiaire du signal d'entrée API 13.4 (iCh_Feedrate2) de l'interface du canal.

Lorsque G94 est active, la commande interprète les adresses F2 "F2" en tant que :

- avance en mm/min ou en pouces/min (voir G94, [chap. 5.2.33 "Programmation de l'avance \(par min\) G94" à la page 179](#)).

Les points suivants s'appliquent ici :

Pour G94, F2 a un effet modal.



Après une remise à zéro ou Reset, la dernière valeur F2 actuelle ne change pas !

C'est pourquoi il faut s'assurer que l'avance nécessaire est toujours programmée avant l'usinage !

Syntaxe :

F2=<valeur>	La syntaxe est "F2=".
	Il ne doit pas y avoir d'espace entre "F2" et "=".
avec	
<valeur>	Est interprétée en tant qu'avance lorsque G94 est active.

Tab. 5-86: Fig. 5.115 : Syntaxe F2=<valeur>

Exemple :

N10 G1 G94 X200 Z300 F200 F2=1000	Avance des axes synchrones : 1000 mm/min, si qCh_Feedrate2=1 et 200 mm/min, si qCh_Feedrate2=0

Particularités et restrictions : L'avance2 programmée peut être écrasée à l'aide de la fonction "Avance d'essai". Celle-ci est commandée via le signal IF du canal "qCh_TestFeed" (avance d'essai).

Comme avec l'avance par défaut, l'avance Override est également active avec l'avance2. De plus, l'avance2 peut être affectée d'une valeur Override indépendante (valeur entre 0 et 100 %) via la donnée système SD.SysChCmd[...].Feedrate2Ovr. L'effet est multiplicatif. Par défaut, SD.SysChCmd[...].Feedrate2Ovr est initialisé sur 100 et disponible pour chaque canal.

Une commutation entre deux valeurs d'avance (F ↔ F2) a le même effet qu'une modification Override.

Fonctions CN avec syntaxe conforme DIN 66025 (extensions incl.)

5.4.4 Vitesse des axes asynchrones "FA"

Description : Tous les mouvements de déplacement des axes asynchrones qui sont programmés dans un bloc FA ne sont pas exécutés en avance rapide, mais à la vitesse programmée.

L'unité d'interprétation de l'avance asynchrone pour les axes linéaires est définie par l'intermédiaire du paramètres de configuration "NCO/CorrUnit/AsynchrAxUnit" : mm/min ou pouces/min.

L'unité peut être changée pour chaque bloc individuel au moyen des attributs de programmation MM(...) et INCH(...).

Pour les axes ronds, l'avance FA programmée est interprétée en degrés/min, indépendamment de la configuration et de l'attribut de programmation.



Une programmation erronée est susceptible d'endommager la machine.

La vitesse indiquée s'applique uniquement au bloc FA actif.

Si des axes asynchrones sont programmés sans nouveau mot FA dans l'un des blocs suivants, alors les axes se déplacent de nouveau en avance rapide.

Syntaxe :

FA<valeur>	
avec	
<valeur>	Vitesse voulue

Tab. 5-87: Syntaxe FA

Exemple :

N10 G1 G94 X200 Z300 F200	Avance des axes synchrones : 200 mm/min
N11 UA400 VA140 FA250	Les axes asynchrones UA et VA se déplacent à une vitesse de 250 mm/min.
N12 UA0 WA10	Les axes asynchrones UA et WA se déplacent à nouveau en avance rapide.

5.4.5 Adresse "oméga" (avance)

Description : Dans la mesure où, dans un bloc, seuls des axes masqués pour la génération de l'avance sont déplacés (voir "FeedAd", [chap. 6.42 "Génération de l'avance : masquage d'axes FeedAd, FAD" à la page 289](#)), il est possible de régler leur **avance** via l'adresse "oméga".

Syntaxe :

Omega<valeur>	
avec	
<valeur>	Avance voulue

Tab. 5-88: Syntaxe Oméga

5.4.6 Programmation de la vitesse de rotation de la broche "S, SSPG"

Description : Quand G97 est active, définit la vitesse de rotation voulue :

- d'une broche individuelle (S...) ou
- d'un groupe de broches complet (SSPG...).

Fonctions CN avec syntaxe conforme DIN 66025 (extensions incl.)



La broche atteint la vitesse de rotation programmée uniquement si une instruction de rotation de broche est active (par ex. rotation à droite : M3, rotation à gauche : M4).

Syntaxe :

S<numéro>=<valeur>	Programmer la vitesse de rotation pour une broche individuelle.
SSPG<groupe>=<valeur>	Programmer la vitesse de rotation pour un groupe de broches complet.
S<valeur>	Écriture abrégée pour la programmation de la vitesse de rotation de la 1re broche. Agit exclusivement sur la 1re broche dans la mesure où elle n'est affectée à aucun groupe de broches par MP 1040 00002. Sinon, c'est tout le groupe de broches contenant la 1re broche qui est programmé.
avec	
<numéro>	Numéro de la broche (index de la broche) Champs de saisie : de 1 jusqu'au nombre de broches définies (définies par MP 1040 00001) Nombre entier
<groupe>	Numéro du groupe de broches Champs de saisie : 1 à 4
<valeur>	Vitesse de rotation de la broche voulue (unité par défaut : tr/min) Valeur saisie : ≥ 0

Tab. 5-89: Syntaxe S et SSPG

Particularités et restrictions :

- Lorsque G96 est active, un mot S est interprété en tant que vitesse de coupe.
Pour la syntaxe, voir [chap. 5.2.41 "Vitesse de coupe constante G96, programmation directe de la vitesse de rotation G97"](#) à la page 190.
- Combiné à la fonction "Orientation broche", un mot S est interprété en tant qu'angle de positionnement.
Pour la syntaxe, voir [chap. 5.3.7 "Orientation de la broche/Positionnement de la broche M19, M119, M219"](#) à la page 208.
- Il est possible de programmer les vitesses de rotation de plusieurs broches/groupe de broches dans un même bloc.
- La vitesse de rotation résultante dépend de l'Override spécifique à la broche.
- *La vitesse de rotation résultante est éventuellement limitée par :*
 - les fonctions SMin ou SMax
(voir [chap. 6.111 "Limitation de la vitesse de rotation SMin, SMN, SMax, SMX"](#) à la page 393)
 - les valeurs limites des rapports de réduction (MP 1040 00011, 1040 00012)

Fonctions CN avec syntaxe conforme DIN 66025 (extensions incl.)

- La vitesse de rotation maximale autorisée d'un groupe de couplage de broches (broches à marche synchrone) dépend du temps de cycle CN.
 Les points suivants s'appliquent ici : $S_{max} [tr/min] = 14400 / (MP\ 9030\ 00001 [ms])$
- La vitesse de rotation définie s'applique tant qu'elle n'est pas écrasée par une nouvelle vitesse de rotation prédéfinie pour la/les même(s) broche(s).
 Après la montée en régime de la commande, S=0 prend effet.
- Si le signal d'interface "qSp_SValueSD" est défini, la vitesse de rotation est définie via la donnée système "SysSpCmdData[n]/Speed". Les valeurs programmées via S, Sx=, SSPGx sont ignorées.

Exemple :		
N10	G97	Activer la programmation de la vitesse de rotation
N20	SSPG1=1000	Toutes les broches du 1er groupe de broches à une vitesse de rotation de 1000 tr/min
:		
N50	S1=2000 S2=60	La 1re broche à une vitesse de rotation de 2000 tr/min
:		La 2e broche à une vitesse de rotation de 60 tr/min
N80	S3=2000	La 3e broche à une vitesse de rotation de 2000 tr/min
:		
N90	S1500	La 1re broche à une vitesse de rotation de 1500 tr/min
:		Dans la mesure où la 1re broche est attribuée à un groupe de broches, la définition de la vitesse de rotation s'applique à tout le groupe de broches.

5.5 Correction d'outil

5.5.1 Correction D "D"

Description : La correction D appelle les valeurs de correction d'outil enregistrées dans des tables de correction D XML au sein de la commande IndraMotion MTX. Une table de correction peut contenir 99 blocs de données au maximum.

Chaque bloc de données contient les valeurs de correction suivantes :

- 3 longueurs d'outil L1, L2, L3,
- le rayon de coupe RAD,
- la position de l'arête coupante ORI.

La correction D est indiquée de la même manière pour les outils de perçage, de fraisage, de tournage et d'équerrage. Les 3 valeurs de décalage L1, L2 et L3 disponibles permettent de réaliser des décalages constants dans l'espace pour un seul outil ainsi que des corrections de longueur parallèles pour un maximum de 3 outils différents.

Une table de correction D est activée à l'aide de la fonction "DcTSel" (DCS), puis un bloc de correction est sélectionné parmi les max. 99 blocs de données via l'ordre CN "D".

Les valeurs de correction prennent effet avec la correction d'outil active G47 (longueurs d'outil et position de l'arête coupante) ou avec la correction de trajectoire de la fraise active G41/G42 ou G141/G142 (rayon de l'outil). Elles sont alors superposées de façon additive avec les valeurs de correction sélectionnées activement de la correction externe de l'outil ED.

Fonctions CN avec syntaxe conforme DIN 66025 (extensions incl.)

Les points suivants s'appliquent ici :

- Le bloc d'outil présélectionné a un effet modal. Une nouvelle programmation supprime le bloc d'outil présélectionné jusqu'à présent.
- Une correction D peut être programmée dans le même bloc que d'autres conditions de course, mouvements de déplacement ou fonctions auxiliaires.
- La correction d'outil n'est calculée que lorsque la fonction CN correspondante a été activée : G47, G41, G42, G141, G142.

Syntaxe :

D<NoBlocOutil>	Présélectionner le bloc de correction d'outil à partir de la table de correction D active
D0	Désélectionner le bloc de correction d'outil, sans en présélectionner un nouveau
avec	
<NoBlocOutil>	Numéro du bloc de données de correction Plage de saisie : 1 à 99

Tab. 5-90: Syntaxe D et D0

Exemple :

N10 D7 Présélectionner les valeurs de correction pour le bloc d'outil 7
:

5.5.2 Correction ED "ED"

Description : La correction d'outil externe (correction ED) appelle les valeurs de correction pour un maximum de 16 arêtes coupantes de l'outil. Les valeurs de correction peuvent être inscrites par l'API via le module de programme "MT_TCorr" ou avec l'ordre CPL "DTC" via le programme pièce.

Chacun des 16 blocs de données contient les valeurs de correction suivantes :

- 3 longueurs d'outil L1, L2, L3,
- le rayon de coupe RAD,
- la position de l'arête coupante ORI.

La correction ED est indiquée de la même manière pour les outils de perçage, de fraisage, de tournage et d'équerrage. Les 3 valeurs de décalage L1, L2 et L3 disponibles permettent de réaliser des décalages constants dans l'espace pour un seul outil ainsi que des corrections de longueur parallèles pour un maximum de 3 outils différents.

Un bloc de correction est sélectionné parmi les 16 blocs de données via l'ordre CN "ED".

Les valeurs de correction prennent effet avec la correction d'outil active G47 (longueur d'outil et position de l'arête coupante) ou avec la correction de trajectoire de la fraise active G41/G42 ou G141/G142 (rayon de l'outil). Elles sont alors superposées de façon additive avec les valeurs de correction sélectionnées activement de la correction D.

Les points suivants s'appliquent ici :

- L'arête coupante d'outil présélectionnée a un effet modal. Une nouvelle programmation supprime l'arête coupante d'outil présélectionnée jusqu'à présent.

Fonctions CN avec syntaxe conforme DIN 66025 (extensions incl.)

- Une correction ED peut être programmée dans le même bloc que d'autres conditions de course, mouvements de déplacement ou fonctions auxiliaires.
- La correction d'outil n'est calculée que lorsque la fonction CN correspondante a été activée :
G47, G41, G42, G141, G142.
- L'arête coupante d'outil présélectionnée peut être indiquée sur l'interface API du canal respectif :
iCh_ActFunc1..24

Syntaxe :

ED<NoArêteCoupanteOutil>	Présélectionner le bloc de correction
ED0	Désélectionner le bloc de correction, sans en présélectionner un nouveau.
avec	
<NoArêteCoupanteOutil>	Numéro de l'arête coupante de l'outil Plage de saisie : 1 à 16

Tab. 5-91: Syntaxe ED

Exemple :

N10 ED7 Présélectionner les valeurs de correction pour l'arête coupante 7
:

6 Fonctions CN avec syntaxe de langage standard



Mouvement de déplacement non intentionnel !

De nombreuses fonctions CN attendent la programmation d'adresses d'axes ou de coordonnées. En règle générale, outre les adresses de coordonnées, seules les adresses des axes du canal, c'est-à-dire les axes synchrones, peuvent être programmées.

Globalement, la programmation de l'adresse d'un axe asynchrone au sein d'une liste de paramètres entraîne une erreur de syntaxe, tandis que la programmation en dehors d'une liste de paramètres entraîne un mouvement de déplacement asynchrone !

Exceptions :

- G74(Home) VA1 :
L'axe asynchrone VA se déplace vers le point de référence.
- GAX(VA) :
L'axe asynchrone VA est accepté sur le canal.
- FsMove, FsTorque, FsReset :
La fonctionnalité "Déplacer sur la butée fixe" a également un effet sur les axes asynchrones.

6.1 Vue d'ensemble

La commande dispose d'un grand nombre de fonctions CN. Outre les ordres déterminés dans la norme DIN 66025, ce chapitre décrit également les extensions essentielles dans le domaine des codes G et les éléments de syntaxe additionnels similaires au langage standard.

Fonctions CN avec syntaxe de langage standard :

Dans le cas des fonctions CN avec syntaxe de langage standard, il est généralement possible de déduire la signification de la fonction à partir de son nom (par ex. "Scale" : mise à l'échelle).

En règle générale, il existe les variantes suivantes d'égale importance :

- Format complet avec écriture en majuscules/minuscules indifférente.
Afin d'améliorer la lisibilité, la première lettre des mots partiels concaténés est toujours inscrite en majuscule dans la présente documentation.
Exemple : "KvProg"
- Format abrégé, composé de trois lettres majuscules (exceptionnellement aussi quatre lettres majuscules).
Exemple :
Format abrégé de KvProg : KVP
Le format abrégé concerné est également affiché dans l'affichage des données de processus pour les fonctions CN modales actives.

Les deux variantes sont indiquées dans les descriptions de syntaxe suivantes.



Vous trouverez un tableau synoptique de toutes les fonctions CN au [chap. 8 "Annexe" à la page 643](#).

En outre, les fonctions CN sont indiquées dans l'index.

Fonctions CN avec syntaxe de langage standard

Écritures utilisées : La présente documentation utilise l'écriture suivante pour la syntaxe des fonctions CN :

Police de caractères "Courier gras" ou "Courier" :

Les chaînes de caractères dans cette police de caractères doivent être programmées comme indiqué.

Exemple : G0 (POL)

Crochets triangulaires < >

Ils identifient un caractère de substitution pour une expression/un paramètre à programmer. Le caractère de substitution est écrit en italique.

Exemple : <axe1>

Accolades { }

Elles identifient une expression/un paramètre optionnel.

De tels éléments de syntaxe peuvent être programmés, mais ce n'est pas une obligation.

Exemple : G0 { ({ POL, } { <Par1> }) }

Caractère "|"

Il sépare les paramètres possibles qui s'excluent réciproquement (paramètres alternatifs).

Exemple : G0 { ({ POL, } { NIPS | IPS1 | IPS2 | IPS3 }) }

6.2 Programmation des facteurs du rayon d'action "ActRadFact, ARF"

6.2.1 Effet

La longueur de trajectoire des blocs programmés est calculée pour calculer l'avance. Pour les axes/coordonnées linéaires, il s'agit de la distance géométrique ($\sqrt{L_1^2 + L_2^2 + L_3^2}$...). Pour les coordonnées/axes ronds, il est possible de définir un facteur supplémentaire par lequel la distance est multipliée, afin de modifier l'influence de l'axe sur l'avance ($\sqrt{L_1^2 + L_2^2 + L_3^2 + (R_1 \cdot F_1)^2 + (R_2 \cdot F_2)^2}$...). Si ce facteur est égal à 1, alors $1^\circ = 1$ mm. Cela correspond à un rayon d'action de 57,3 mm.

Exemple L'axe linéaire se déplace de 100 mm, l'axe rond se déplace de 100 degrés, les deux axes se déplacent avec l'avance programmée/racine(2) en [mm/min] ou en [degrés/min], car les deux déplacements sont considérés comme égaux.

Il est possible de définir un facteur de rayon d'action via les paramètres machine "Facteurs de mise à l'échelle spécifiques au canal pour les vitesses d'axes ronds" (NCO/VelScaleFact/Ch[k]/FactRotInch et NCO/VelScaleFact/Ch[k]/FactRotMetr).

Un facteur **supplémentaire** peut être saisi dans le programme pièce à l'aide de ActRadFact.

Outre les facteurs de rayon d'action pour les axes et les coordonnées, il est également possible de saisir un facteur de rayon d'action pour l'orientation, si la transformation active prend cette dernière en charge.

6.2.2 Programmation

Syntaxe :

ActRadFact (<NomAxe> <facteur> {, <NomAxe> <facteur>}...)
Format abrégé : ARF (. .)

Fonctions CN avec syntaxe de langage standard

avec	
<NomAxe>	Nom d'un axe rond, d'une coordonnée ou "O" (orientation)
<facteur>	Facteur du rayon d'action valable à partir de ce bloc (facteur >= 0)
ActRadFact ()	Tous les facteurs de rayon d'action du canal sont réinitialisés à 1.

Tab. 6-1: Syntaxe ActRadFact

Particularités et restrictions :

- Aucun facteur de rayon d'action ne doit être négatif.
- Après le démarrage du système, tous les facteurs de rayon d'action de ActRadFact sont réinitialisés à 1. Les facteurs configurés dans les paramètres machine reprennent effet.
- Si un axe est retiré du canal, il est également supprimé du facteur de rayon d'action programmé.
- Si le rayon d'action d'un axe/d'une coordonnée est égal à 0, aucun mouvement contenant uniquement des parties de cet axe/cette coordonnée ne doit être programmé.
- Si la transformation active est modifiée, les facteurs de rayon d'action de coordonnées identiques seront acceptés. Tous les autres facteurs de rayon d'action sont réinitialisés à la valeur 1.

Exemple :

G0 X0 A0	Position de départ
N010 G1 G91 F1000 X100 A100	X → 707[mm/min] A → 707[degrés/min]
N020 ARF(A0.5)	La partie A est divisée par deux.
N030 X100 A100	X → 894[mm/min] A → 894[degrés/min]
N040 ARF(A2)	La partie A est multipliée par deux.
N050 X100 A100	X → 447[mm/min] A → 447[degrés/min]
N060 ARF(A10)	La partie A est multipliée par dix.
N070 X100 A100	X → 99[mm/min] A → 99[degrés/min]
N080 ARF(A0)	La partie A est masquée.
N090 X100 A100	X → 1000[mm/min] A → 1000[degrés/min]
N100 ARF()	Réinitialisation du facteur de rayon d'action à 1 pour A

6.3 Comparaison des fonctions CN modales "Adjust, ADJ"

Description : La fonction de comparaison "Adjust" permet de comparer aisément les fonctions CN dans les programmes de comparaison après l'avance de bloc. La syntaxe d'une fonction CN modale est fournie comme paramètre à la fonction de comparaison. La commande active alors la fonction CN effective dans le groupe modal correspondant avant le bloc cible de l'avance de bloc.

Syntaxe :

ADJ (<nom FCN>)	
<nom FCN>	Syntaxe d'une fonction CN modale à partir du groupe devant être comparé.

Tab. 6-2: Syntaxe ADJ

Exemple :

ADJ(G0)

Fonctions CN avec syntaxe de langage standard

La fonction géométrique ayant un effet modal placée juste avant le bloc cible (G0, G1, G2, G3...) est activée et le groupe de fonctions CN est alors comparé.

Pour les fonctions CN mentionnées ci-après, les paramètres spécifiques à la fonction sont alors comparés simultanément :

Groupe modal	Fonctions CN	Paramètres pris en charge
Fonctions d'interpolation	G0	Programmation polaire et fenêtre d'arrêt précis (NIPS, IPS<n>)
	G1	Programmation polaire et fenêtre d'arrêt précis (IPS<n>)
	G2, G3	Programmation polaire
Pente de trajectoire	G8	Forme de trajectoire (Shape)
	G9	Forme de trajectoire, d'axe (Shape)
Plan	G17, G18, G19, G20	Affectation d'axe
Correction de la longueur d'outil	G47	Affectation des longueurs
Décalage d'origine 1	G52.1	Décalage d'origine programmable 1 [°])
	G53.1	Décalage d'origine 1 désactivé [°])
	G54.1 à G59.1	Banque de tableaux de décalage d'origine active 1 [°])
...		
Décalage d'origine 5	G52.5	Décalage d'origine programmable 5 [°])
	G53.5	Décalage d'origine 5 désactivé [°])
	G54.5 à G59.5	Banque de tableaux de décalage d'origine active 5 [°])
Placements : "Plan incliné" 1	G151.1	Orientation d'outil [°])
	G152.1	Plan incliné programmable 1 [°])
	G153.1	Plan incliné 1 désactivé [°])
	G154.1 à G159.1	Banque de tableaux plan incliné active 1 [°])
...		
Placements : "Plan incliné" 5	G151.5	Orientation d'outil 5 [°])
	G152.5	Plan incliné programmable 5 [°])
	G153.5	Plan incliné 5 désactivé [°])
	G154.5 à G159.5	Banque de tableaux plan incliné active 5 [°])
Arrêt précis	G61	Fenêtre d'arrêt précis (IPS<n>)
Affectation des longueurs d'outil	G78	ActPlane ou affectation des longueurs
Avance sur trajectoire	G93	Valeur F
	G94	Valeur F, valeur F2
	G95	Valeur F, axe de référence (FRA)
Programmation de la broche	G96	Axe de référence, affectation de la broche
	G97	Affectation de la broche
Correction D	D0	Correction D désactivée [°])

Fonctions CN avec syntaxe de langage standard

Groupe modal	Fonctions CN	Paramètres pris en charge
	D1 à D99	Correction D active***)
Accélération de l'axe	AAC	Accélérations d'axe programmées
Secousse d'axe	AJK	Secousse d'axe programmée
Décalage additif des coordonnées du programme	ATR	Coordonnées de l'origine du programme
Vitesse maximale de l'axe	AVE	Liste des vitesses de l'axe
Couplage d'axes du canal	AXC	Axes directeurs du canal de l'avance de bloc avec les rapports de couplage des axes esclaves et tableaux correspondants. Conditions préalables : Le bloc cible ne doit contenir aucun ordre de couplage. Le couplage doit être désactivé.
Position de la pièce à usiner	BCR	Coordonnées d'origine de la pièce à usiner, angles
Chanfreins/Arrondis	CHL CHS RND RNE	Longueur de chanfrein, segment de chanfrein Rayon, epsilon
Détection automatique des angles	CLD	Angle de coude, facteur d'angle, écart de points max., facteur d'écartement, barrière de précision
Détection de collision sur la trajectoire de la fraise	CLN	Type de surveillance, zone de prévisualisation
Transformation d'axe	COORD	Transformation d'axe 1 et 2 avec options de transformation
Programmation du diamètre/rayon	DIA RAD	Coordonnées du diamètre
Broches du canal et groupes de broches	DSP, GSP, RSP SPGn, SPGALL	Broches du canal et affectation des groupes du canal (correspond à SPADJSP)
Enchaînement de blocs sans réduction de la vitesse	DTT	Angle de raccordement
Axes de masquage de l'avance	FAD	Liste des axes de masquage
Coordonnées d'avance du groupe	FDG FDGT	Listes des coordonnées de trajectoire, type de groupe d'avance
Compensation des erreurs de rectitude et d'angle	GCT	-
Secousse d'axe active	JAL	Liste des axes de secousse d'axes pertinents pour la trajectoire
Limitation des secousses	JKC	-
Aide à la saisie – Fonction miroir	MIR	Axes de symétrie
Broche principale	MSP	Broche
Accélération sur trajectoire	PAC	Accélération sur trajectoire, décélération sur trajectoire
Définition du pôle	PLS	Point d'inversion miroir/de rotation

Fonctions CN avec syntaxe de langage standard

Groupe modal	Fonctions CN	Paramètres pris en charge
Mode de positionnement de l'axe rond	PMD	Mode de positionnement
Définition du pôle	POP	Coordonnées du pôle
Programmation de précision	PRP	Erreur de contour du rayon (EPS), poursuite (DIST)
Accélération radiale maximale	RAC	Accélération
Aide à la saisie – Rotation	ROT	Angle
Aide à la saisie – Mise à l'échelle	SCL	Facteurs d'axe
Aide à la saisie – Décaler	SHT	Coordonnées du point d'origine du contour
Définition Spline	SPD	Type de spline, coordonnées spline, condition de départ (SBC), condition d'arrivée (EBC), longueur des paramètres de la spline (PL)
Divisions géométriques	SPT	Mode, longueur
Coordonnées du programme, décalage	TRS	Coordonnées de l'origine du programme
Broche de taraudage	TSP	Broche

Tab. 6-3: Fonctions CN modales comparables et paramétrables

- *) Présélection implicite de tableaux de décalage d'origine (ZOS)
- **) Présélection implicite de tableaux de placement (PMS)
- ***) Présélection implicite de tableaux de correction D (DCS)

Particularités et restrictions :

- Pour les fonctions CN modales paramétrables qui ne sont pas citées ci-dessus, la comparaison devra s'effectuer le cas échéant à l'aide des données système de l'avance de bloc dans le programme de comparaison (InitAdjPrg) ou à l'aide du bloc d'action.
- La comparaison a lieu séparément pour chaque banque d'un décalage d'origine ou d'un plan incliné : ADJ(G54.1), ADJ(G54.2), etc.
- À l'état comparé, la correction du rayon d'outil G41/G42 n'est active qu'avec le bloc d'approche après la comparaison. Pendant la comparaison, l'effet est supprimé en interne et aucune correction n'est calculée.

6.4 Signal programmable selon la distance

6.4.1 Généralités

La fonction Advance-Distance-Signal (ADS) émet un signal avant d'atteindre une marque programmée. Les propriétés d'un signal sont définies par la donnée système /SysAdvDistSig/Sig[].



Vous trouverez de plus amples informations dans la documentation "Description des fonctions – Fonctions spéciales IndraMotion MTX", au chapitre "Signal programmable selon la distance".

6.4.2 Activation de la recherche d'une marque "ADSENA"

Description : Advance-Distance-Signal-ENable (ADSENA)

ADSENA reprend le contenu actuel de la donnée système "/SysAdvDistSig/Sig[]" et active la recherche de la marque programmée (voir ADSLAB).

Fonctions CN avec syntaxe de langage standard

Syntaxe :

ADSENA (<signal> {,<signal> ...})	
avec	
<signal>	1 à 8 : numéro d'un signal

Tab. 6-4: Syntaxe ADSENA

Exemple :

SignalConfig

```

N00 WAIT
001 SD.SysAdvDistSig.Unit = 0 : REM mm
002 REM
003 REM Configure Signal 3
004 SD.SysAdvDistSig.Sig[3].DistOn = 1.2 : REM 1.2 mm
005 SD.SysAdvDistSig.Sig[3].DistOff = 0.0 : REM 0.0 mm
006 SD.SysAdvDistSig.Sig[3].TimeOn = 0.0 : REM 0.0 msec
007 SD.SysAdvDistSig.Sig[3].TimeOff = 0.0 : REM 0.0 msec
008 SD.SysAdvDistSig.Sig[3].Duration = 40 : REM 40 msec
009 SD.SysAdvDistSig.Sig[3].DistCalc = 0 : REM calc. dist. in active plane
010 SD.SysAdvDistSig.Sig[3].Type = 0 : REM iCh_Custom2
011 SD.SysAdvDistSig.Sig[3].Index = 2 : REM iCh_Custom2

```

PartProg

```

...
N20 ADSENA(3) ; Activate look-ahead for label ADSLAB(3)
N21 G1 X... Y... F...
N22 G2 X... Y...
N23 G1 X... Y...
...
N50 G1 X... Y... ADSLAB(3) ; Set signal iCh_Custom2 1.2 mm before end of block
N51 G1 X... Y...
...
N80 ADSDIS(3) ; Deactivate look-ahead for label ADSLAB(3)
...

```

Particularités : La remise à zéro et M30 désactivent la fonction.

6.4.3 Désactivation de la recherche d'une marque "ADSDIS"

Description : Advance-Distance-Signal-DISable (ADSDIS)

ADSDIS désactive la recherche de la marque programmée (voir ADSLAB).

Syntaxe :

ADSDIS (<signal> {,<signal> ...})	
avec	
<signal>	1 à 8 : numéro d'un signal
ADSDIS ()	Désactive la recherche de toutes les marques.

Tab. 6-5: Syntaxe ADSDIS

Exemple : Voir "Activation de la recherche d'une marque "ADSENA"".

Particularités :

- ADSDIS ne met pas fin à l'émission d'un signal.
- Les signaux de commutation peuvent être supprimés avec la fonction ADSCLR(<signal>).

6.4.4 Définition d'une marque "ADSLAB"

Description : Advance-Distance-Signal-LABel (ADSLAB)

ADSLAB est la marque recherchée.

Fonctions CN avec syntaxe de langage standard

Syntaxe :	
ADSLAB (<signal> {,<signal> ...})	
avec	
<signal>	1 à 8 : Le signal définit "DistOn" + "TimeOn" avant la marque. -1 à -8 : Le signal réinitialise "DistOff" + "TimeOff" avant la marque.

Tab. 6-6: Syntaxe ADSLAB

Exemple : Voir "Activation de la recherche d'une marque "ADSENA"" et "Réinitialisation d'un signal de commutation "ADSCLR"".

6.4.5 Réinitialisation d'un signal de commutation "ADSCLR"

Description : Advance-Distance-Signal-CLeaR (ADSCLR)

Réinitialise les signaux de commutation.

Syntaxe :	
ADSCLR (<signal> {,<signal> ...})	
avec	
<signal>	1 à 8 : numéro d'un signal
ADSCLR ()	Réinitialise tous les signaux de commutation.

Tab. 6-7: Syntaxe ADSCLR

Exemple : *SignalConfig*

```

N00 WAIT
N01 SD.SysAdvDistSig.Unit = 0 : REM mm
N02 REM
N03 REM Configure Signal 3
N04 SD.SysAdvDistSig.Sig[3].DistOn = 1.2 : REM 1.2 mm
N05 SD.SysAdvDistSig.Sig[3].DistOff = 0.0 : REM 0.0 mm
N06 SD.SysAdvDistSig.Sig[3].TimeOn = 0.0 : REM 0.0 msec
N07 SD.SysAdvDistSig.Sig[3].TimeOff = 0.0 : REM 0.0 msec
N08 SD.SysAdvDistSig.Sig[3].Duration = -1 : REM switching signal
N09 SD.SysAdvDistSig.Sig[3].DistCalc = 0 : REM calc. dist. in active plane
N10 SD.SysAdvDistSig.Sig[3].Type = 0 : REM iCh_Custom2
N11 SD.SysAdvDistSig.Sig[3].Index = 2 : REM iCh_Custom2

```

PartProg

```

...
N20 ADSENA(3) ; Activate look-ahead for label ADSLAB(3)
N21 G1 X... Y... F...
N22 G2 X... Y...
N23 G1 X... Y...
...
...
N50 G1 X... Y... ADSLAB(3) ; Set signal iCh_Custom2 1.2 mm before end of block
N51 G1 X... Y...
N52 G1 X... Y... ADSLAB(-3) ; Reset signal at end of block
...
N80 ADSDIS(3) ; Deactivate look-ahead for label ADSLAB(3)
...

```

"Duration = -1" définit le signal 3 en tant que signal de commutation.

Quand le programme pièce "PartProg" est interrompu dans le bloc N51 par la remise à zéro, le signal 3, et donc aussi iCh_Custom2, restent définis. Quand l'ordre ADSCLR() est contenu dans la chaîne d'initialisation ou dans le programme d'initialisation, alors le signal est réinitialisé lors de la remise à zéro.

6.5 Extension du bloc-programme avant Retrace "AdvanceAppend, ADVA"

Description : Pour le fonctionnement en sens inverse, un bloc-programme avant est généré pour chaque bloc-programme pièce dans la zone Retrace de Retrace(1) à Retrace(0). La fonction "AdvanceAppend" insère les fonctions CN indiquées à la fin du bloc-programme avant actuel.

Syntaxe :

AdvanceAppend (<FCN> . . . <FCN>)	
Format abrégé : ADVA (...)	
avec	
<FCN>...<FCN>	Liste des fonctions CN

Tab. 6-8: Syntaxe AdvanceAppend (ADVA)



En cas de fonctionnement en sens inverse, toute modification du contour entraîne une erreur de commutation.

Les fonctions CPL et les appels de sous-programmes ne sont pas autorisés.

Exemple :

Programme pièce :	Programme avant :	Observation :
N100 X100 M10	N1 X=100	Les fonctions auxiliaires ne sont pas reprises.
N200 X0 ADVA (M125 M126)	N2 X=0 M125 M126	Dans le programme avant, il est nécessaire d'indiquer M125 et M126 à l'API.



À des fins de contrôle, il est possible d'enregistrer le programme avant modifié dans un fichier (voir [chap. 6.100 "Recul Retrace, RETRACE"](#) à la page 377).

6.6 Remplacement du bloc-programme avant Retrace "AdvanceSubstitute, ADVS"

Description : Un bloc-programme avant est généré pour chaque bloc-programme pièce dans la zone Retrace de Retrace(1) à Retrace(0). La fonction "AdvanceSubstitute" remplace le bloc-programme avant actuel par les fonctions CN indiquées.

Syntaxe :

AdvanceSubstitute (<FCN> . . . <FCN>)	
Format abrégé : ADVS (...)	
avec	
<FCN>...<FCN>	Liste des fonctions CN

Tab. 6-9: Syntaxe AdvanceSubstitute (ADVS)



En cas de fonctionnement en sens inverse, toute modification du contour entraîne une erreur de commutation.

Les fonctions CPL et les appels de sous-programmes ne sont pas autorisés.

Fonctions CN avec syntaxe de langage standard

Exemple :	Programme pièce :	Programme avant :	Observation :
	N5 XO F10 ADVS(XO F100)	N1 XO F100	Autoriser une avance supérieure dans le programme avant. Toujours ajouter les coordonnées du point d'arrivée programmées !



À des fins de contrôle, il est possible d'enregistrer le programme avant modifié dans un fichier (voir [chap. 6.100 "Recul Retrace, RETRACE"](#) à la page 377).

6.7 Surveillance de zone "Area, ARA"

Description : Définit, active ou désactive jusqu'à 32 zones mortes ou zones de travail rectangulaires, bidimensionnelles avec des limites parallèles aux axes et surveille leurs points d'intersection avec des mouvements de déplacement droits ou circulaires.

- **Zones mortes :**

Elles et leurs limites ne doivent ni être croisées, ni être touchées lors d'un mouvement de déplacement.

- **Zones de travail :**

Elles et leurs limites ne doivent pas être quittées lors d'un mouvement de déplacement.



Les données par défaut de toutes les zones de 1 à 10 sont définies dans le groupe de paramètres machine MP 8002.

Définir la donnée système optionnelle SysARACfg annule l'effet du groupe de paramètres machine MP 8002 : les données par défaut de toutes les zones doivent être configurées/pré-affectées dans SysARACfg.



Une possibilité de définir les zones de surveillance exclusivement par les données système est décrite dans la documentation "Description des fonctions – Fonctions spéciales IndraMotion MTX 14VRS".

Zones statiques : Normalement, les zones sont définies statiquement sur la machine. La détection d'un dépassement de zone entraîne une erreur et l'interruption du programme. Seules les zones actives sont surveillées (-> statut de surveillance). La donnée système SysARACfg (optionnelle) peut être définie pour les zones statiques. Celle-ci comprend toutes les pré-affectations éventuelles. La configuration par les paramètres machine devient alors inutile.

Zones dynamiques : Au cours du traitement du programme, les zones dynamiques peuvent devenir actives ou inactives (libérées), par ex. avec une remontée d'outil mécanique.

Les zones dynamiques sont surveillées de la même manière que les zones statiques.

Seules les zones actives sont surveillées (-> statut de surveillance).

Les données système (optionnelles) suivantes sont pertinentes et peuvent être définies pour les zones dynamiques :

Fonctions CN avec syntaxe de langage standard

- SysARACfg
- SysARA
- SysARAAdm

La propriété "dynamique" (d'une zone) est optionnelle et configurable, et peut être définie exclusivement dans la donnée système SysARACfg.DynamicArea[<i>] pour chaque zone <i>.

0 : Zone statique.

1 : Zone dynamique, peut uniquement être désactivée quand aucune autre zone dynamique est inactive.

2 : Zone dynamique, peut être activée/désactivée en fonction des autres zones dynamiques.

L'activation ou l'inactivation d'une zone dynamique doit être affectée (par l'API) par l'intermédiaire de la donnée système SysARA.PlcAreaDisabled[<i>] appliquée au moment de l'exécution :

0 : La zone est active.

1 : La zone est inactive/libérée.

Si un dépassement de zone dynamique est détecté, le programme n'est pas interrompu, mais le mouvement est arrêté avant le mouvement de déplacement critique. Un avertissement est alors émis (l'avertissement peut être rejeté en option avec SysARAAdm.MsgDisable=1) et le mouvement de déplacement est arrêté jusqu'à ce que la zone concernée soit libérée et que cela soit signalé par l'API.

L'état "Zone dépassée" est signalé au moment de l'exécution du bloc CN critique par un identificateur dans la donnée système (SysARA.AreaViolationOccured[<i>]=1). Après la libération (désactivation) de la zone concernée dans l'application (API -> interface de données système SysARA.PlcAreaDisabled[<i>]=1), le mouvement de déplacement critique est exécuté et l'état "Zone dépassée" est acquitté (SysARA.AreaViolationOccured[<i>]=0).

Si SysARA.PlcAreaDisabled[<i>]=1 est détecté avant d'atteindre le mouvement de déplacement critique, celui-ci est exécuté sans communication.

Pour que l'arrêt puisse se faire à temps à la limite d'une zone, il est possible de réduire la vitesse le cas échéant.

Il est recommandé de sélectionner une extension de zone qui empêche toute collision en cas de dépassement des axes.

Syntaxe : Activer, désactiver, libérer une zone de surveillance individuelle ou l'ensemble des zones de surveillance.

Area (<NoZone> , <statut>)
Format abrégé : ARA (...)
avec

Fonctions CN avec syntaxe de langage standard

<NoZone>	Numéro de la zone Nombre entier. Champ de saisie : -1, 1 à 32 -1 : Désactiver/activer toutes les zones
<statut>	Statut de surveillance voulu 0 : Désactiver la surveillance 1 : Activer la surveillance -1 : Désactiver et libérer la surveillance

Tab. 6-10: Syntaxe 1 Area (ARA)

Définition, activation, désactivation et libération d'une zone de surveillance individuelle.

Area (<NoZone> , <statut>{ , <mod> , {<P1>} , {<P2>} , {<D1>} , {<D2>} , {<axe1>} , {<axe2>}})	
Format abrégé : ARA (...)	
avec	
<NoZone>	Numéro de la zone. Nombre entier. Champ de saisie : 1 à 32.
<statut>	Statut de surveillance voulu 0 : Désactiver la surveillance de la zone <NoZone> 1 : Activer la surveillance de la zone <NoZone> -1 : Désactiver la surveillance de la zone <NoZone> et libérer la zone <NoZone>
<mod>	0 : La zone <NoZone> n'est pas utilisée. 1 : La zone <NoZone> est une zone morte. 2 : La zone <NoZone> est une zone de travail.
<P1>	Position dans le système de coordonnées de machine Détermine la valeur de la position du centre de la zone par rapport au premier axe du système participant à la zone (défini par MP 8002 00001 (en option : SysARACfg.Pos1) ou programmé avec axe1). Voir l'exemple suivant Unité de programmation, comme pour les coordonnées d'axes
<P2>	Position dans le système de coordonnées de machine Détermine la valeur de la position du centre de la zone par rapport au deuxième axe du système participant à la zone (défini par MP 8002 00002 (en option : SysARACfg.Pos2) ou programmé avec axe2). Voir l'exemple suivant. Unité de programmation, comme pour les coordonnées d'axes
<D1>	Détermine la longueur de la zone par rapport au premier axe du système participant à la zone (défini par MP 8002 00001 (optionnel : SysARACfg.Ext1) ou programmé avec axe1). Voir l'exemple suivant. Unité de programmation, comme pour les coordonnées d'axes

Fonctions CN avec syntaxe de langage standard

<D2>	Détermine la longueur de la zone par rapport au deuxième axe du système participant à la zone (défini par MP 8002 00002 (optionnel : SysARACfg.Ext2) ou programmé avec axe2). Voir l'exemple suivant. Unité de programmation, comme pour les coordonnées d'axes
<axe1>	Définit le premier axe du système participant à la zone (optionnel : SysARACfg.AreaSysAxisNoDim1) Numéro d'axe du système
<axe2>	Définit le deuxième axe du système participant à la zone (optionnel : SysARACfg.AreaSysAxisNoDim2) Numéro d'axe du système

Tab. 6-11: Syntaxe 2 Area (ARA)



Les valeurs non programmées restent conservées dans la mesure où elles ont déjà été indiquées une fois dans le déroulement du programme.

Si elles n'ont jamais été indiquées dans le déroulement du programme, la commande utilise les valeurs correspondantes issues du groupe de paramètres machine 8002 pour les zones 1 à 10 et les valeurs restent vides pour les zones 11 à 32 (en option : pré-affectation SysARACfg).

Exemple :

N100 Area(4,0,,100,200)	Désactiver la zone 4 et placer le centre de la zone sur les coordonnées de machine indiquées (100,200). Les longueurs de zone restent inchangées.
N120 Area(12,1,2,100,200,,1,3)	Activer la zone 12 avec le type Zone de travail et placer le centre de la zone sur les coordonnées de machine indiquées (100,200). Les longueurs de zone restent inchangées. Premier axe du système participant à la zone : 1er axe. Deuxième axe du système participant à la zone : 3e axe.
N150 Area(12,-1)	Désactiver la zone 12 et la libérer pour les autres canaux

Fonctions CN avec syntaxe de langage standard

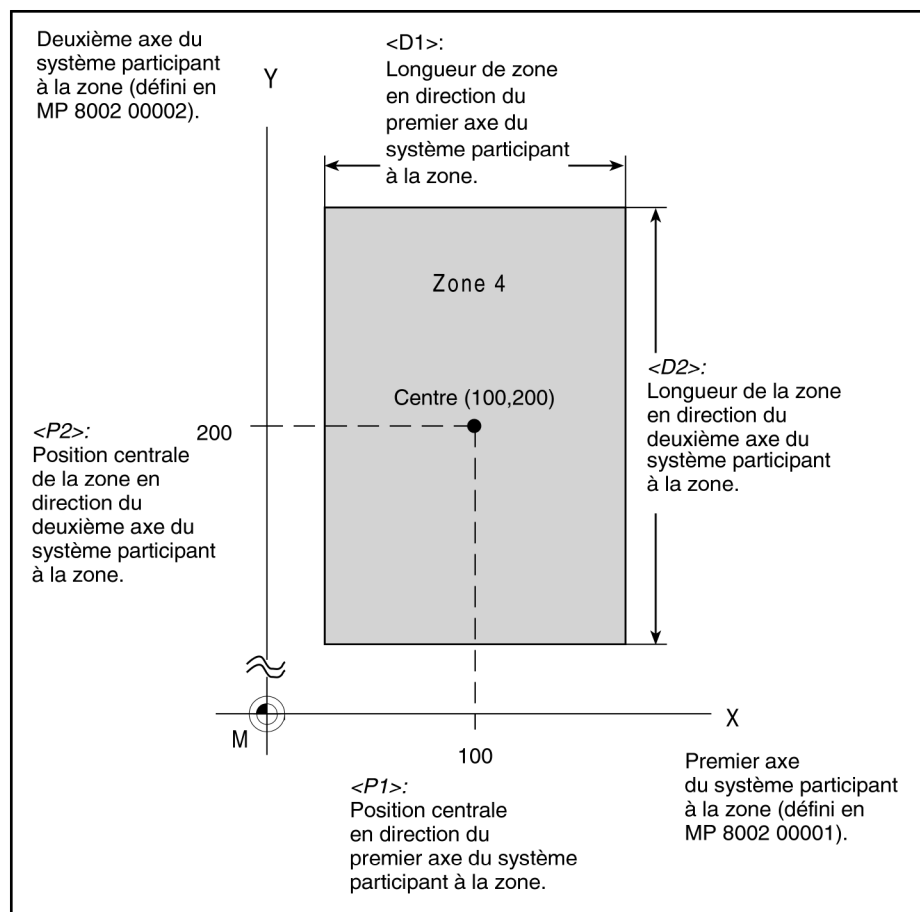


Fig. 6-1: Exemple Area (ARA)

Particularités et restrictions :

- Le point suivant s'applique aux zones 1 à 10 : tous les axes du système participant aux différentes zones doivent être définis dans MP 8002 00001 et MP 8002 00002 ou être programmés avec axe1 et axe2.
- Le point suivant s'applique aux zones 11 à 32 : tous les axes du système participant aux différentes zones doivent être programmés avec axe1 et axe2.
- Le réglage correspondant doit impérativement être enregistré dans MP 8002 00032 pour pouvoir influencer les zones 1 à 10 avec "Area".
- La fonction nécessite des axes référencés.
- Lors de l'activation d'une zone, les axes correspondants doivent se trouver sur le canal actuel.
- La fonction "Area" influence une zone uniquement sur le canal dans lequel "Area" est programmée. Si des axes d'une zone sont transmis à un autre canal, la zone sur le canal cible présente alors les valeurs pré-réglées par défaut dans le groupe de paramètres machine 8002 et elle est inactive. Les valeurs éventuellement programmées sur le canal source via "Area" ne sont pas transmises.
- Chaque zone existe une seule fois dans l'ensemble du système et peut être modifiée par un seul canal. Si une zone est utilisée par un autre canal, elle doit d'abord être libérée par le premier canal.
- En mode JOG : les axes déplacés à l'aide de la manivelle ne sont pas surveillés.

Fonctions CN avec syntaxe de langage standard

- En mode JOG : un seul des axes qui sous-tendent une zone morte peut être déplacé en mode JOG à la fois.
- Les dépassements de zone en mode JOG génèrent un avertissement. L'axe concerné reste à l'arrêt et ne peut être déplacé en mode JOG qu'en sens contraire.
- Si les dimensions d'une zone se trouvent au sein du plan actif, les mouvements linéaires et circulaires sont surveillés pour cette zone.
- Si les dimensions d'une zone se trouvent en dehors du plan actif, seuls les mouvements linéaires sont surveillés pour cette zone.
- Si les dimensions d'une zone se trouvent partiellement en dehors du plan actif, seuls les mouvements linéaires sont surveillés pour cette zone. Les mouvements circulaires ne peuvent pas être surveillés et déclenchent un avertissement.
- Sont surveillés les mouvements de déplacement linéaires dont les axes participants se trouvent sur le canal surveillé.
- Sont surveillés les mouvements de déplacement circulaires, dans la mesure où la zone de surveillance respective et ses deux dimensions d'axe se trouvent dans le plan actif.
- Si les deux dimensions d'une zone de surveillance se trouvent à l'extérieur du plan actif lors d'un mouvement circulaire, alors la surveillance n'a pas lieu pour cette zone. Si une seule dimension de la zone se trouve dans le plan actif, un avertissement est émis pour informer que le mouvement n'est pas surveillé.

Données système pour les zones de protection dynamiques

Les données système des zones de protection sont volatiles. Les données de configuration peuvent être pré-affectées à l'aide d'un fichier SDDatA-ra.xml.

Mode opératoire permettant de définir les données système des zones de protection :

Copier le fichier "feprom/schemas/sdara.xsd_" dans "usrfep/schemas/sdara.xsd", puis intégrer les définitions mentionnées ci-dessous dans un fichier de définition actif (par ex. usrfep/SDDefMtb.xml).

```
<Variable Dimension="32" Storage="volatile">
```

```
<Name>SysARA</Name>
```

```
<Type>ARA_t</Type>
```

Création des données système ARA :

```
...
```

```
</Variable>
```

```
<Variable Dimension="32" Storage="volatile">
```

```
<Name>SysARACfg</Name>
```

```
<Type>ARACfg_t</Type>
```

```
...
```

```
</Variable>
```

```
<Variable Storage="volatile">
```

```
<Name>SysARAAdm</Name>
```

```
<Type>ARAAdm_t</Type>
```

```
....
```

```
</Variable>
```

Fonctions CN avec syntaxe de langage standard

Les modèles de ces définitions sont enregistrés dans feprom/SDDe-fOem.xml.

6.8 Sous-programmes asynchrones

6.8.1 Sous-programmes asynchrones : désinscription "ASPCLR"

Description : Désinscrit un sous-programme asynchrone sur le canal actuel.

Il est impossible d'activer ou de désactiver des sous-programmes désinscrits.



Pour réinscrire des sous-programmes asynchrones désinscrits, voir fonction "ASPSET", [chap. 6.8.5 "Sous-programmes asynchrones : inscription ASPSET" à la page 238](#).



Vous trouverez toutes les informations sur l'utilisation et la programmation de sous-programmes asynchrones dans la documentation "Description des fonctions IndraMotion MTX".

Syntaxe :

ASPCLR (<NoSP>)	
avec	
<NoSP>	Numéro du sous-programme Champ de saisie : 1 à 8. Nombre entier

Tab. 6-12: Syntaxe ASPCLR

- Particularités et restrictions :**
- Le sous-programme correspondant doit être inscrit sur le canal cible (voir ASPSET, [chap. 6.8.5 "Sous-programmes asynchrones : inscription ASPSET" à la page 238](#)).
 - Pour désactiver temporairement un sous-programme, voir la fonction "ASPDIS", [chap. 6.8.2 "Sous-programmes asynchrones : mise hors service ASPDIS" à la page 236](#).

6.8.2 Sous-programmes asynchrones : mise hors service "ASPDIS"

Description : Désactive un sous-programme asynchrone sur le canal actuel.

Un sous-programme désactivé n'est pas appelé lorsque survient l'événement pertinent.



Vous trouverez toutes les informations sur l'utilisation et la programmation de sous-programmes asynchrones dans la documentation "Description des fonctions IndraMotion MTX".

Syntaxe :

ASPDIS (<NoSP> {,<NoSPInt>})	
avec	

Fonctions CN avec syntaxe de langage standard

<NoSP>	Numéro du sous-programme Champ de saisie : 1 à 8. Nombre entier
<NoSPInt>	Sous-programme asynchrone interruptible (en option) : Aucune indication : – Verrouiller l'interruption du programme pièce Numéro du sous-programme asynchrone interruptible : Champ de saisie : 1 à 8. Nombre entier – Verrouiller l'interruption du sous-programme asynchrone <NoSPInt>

Tab. 6-13: Syntaxe ASPDIS

- Particularités et restrictions :**
- Le sous-programme correspondant doit être inscrit sur le canal cible (voir ASPSET, [chap. 6.8.5 "Sous-programmes asynchrones : inscription ASPSET" à la page 238](#)).
 - Pour activer un sous-programme désactivé, voir la fonction "ASPENA", [chap. 6.8.3 "Sous-programmes asynchrones : mise en service ASPENA" à la page 237](#).

6.8.3 Sous-programmes asynchrones : mise en service "ASPENA"

Description : Active un sous-programme asynchrone sur le canal actuel.
 Seuls les sous-programmes activés peuvent être appelés lorsque survient l'événement pertinent.



Vous trouverez toutes les informations sur l'utilisation et la programmation de sous-programmes asynchrones dans la documentation "Description des fonctions IndraMotion MTX".

Syntaxe :

ASPENA (<NoSP> {,<NoSPInt>})	
avec	
<NoSP>	Numéro du sous-programme Champ de saisie : 1 à 8. Nombre entier
<NoSPInt>	Sous-programme asynchrone interruptible (en option) : Aucune indication : – Libérer l'interruption du programme pièce Numéro du sous-programme asynchrone interruptible : Champ de saisie : 1 à 8. Nombre entier – Libérer l'interruption du sous-programme asynchrone <NoSPInt>

Tab. 6-14: Syntaxe ASPENA

- Particularités et restrictions :**
- Le sous-programme correspondant doit être inscrit sur le canal cible (voir ASPSET, [chap. 6.8.5 "Sous-programmes asynchrones : inscription ASPSET" à la page 238](#)).
 - Pour désactiver un sous-programme, voir la fonction "ASPDIS", [chap. 6.8.2 "Sous-programmes asynchrones : mise hors service ASPDIS" à la page 236](#).

Fonctions CN avec syntaxe de langage standard

6.8.4 Sous-programmes asynchrones : Définir le point de redémarrage "ASPRTP"

Description : *Définit le positionnement de la commande à la fin d'un sous-programme asynchrone d'un bloc de déplacement éventuellement interrompu :*

- Point de départ
- Point d'arrivée
- Point d'interruption

Si aucun bloc de déplacement n'est actif au moment de l'interruption, la commande se positionne toujours sur les dernières coordonnées actives.



Vous trouverez toutes les informations sur l'utilisation et la programmation de sous-programmes asynchrones dans la documentation "Description des fonctions IndraMotion MTX".

Syntaxe :

ASPRTP (<NoSP> , <point>)	
avec	
<NoSP>	Numéro du sous-programme Champ de saisie : -1 ; 1 à 8. Nombre entier -1 : Définir le point de redémarrage voulu pour tous les sous-programmes asynchrones du canal actuel.
<point>	Point de redémarrage voulu : 1 : Point de départ 2 : Point d'arrivée 3 : Point d'interruption

Tab. 6-15: Syntaxe ASPRTP

Particularités et restrictions :

- Le point de redémarrage voulu et réglé est effacé par une remise à zéro ou par M30.
- Les modifications de correction survenues entretemps au sein d'un sous-programme asynchrone sont prises en compte automatiquement pour le calcul interne du point de redémarrage requis.
- La fonction "REPOSTP" (voir [chap. 6.99.2 "Définition du point de redémarrage dans un sous-programme asynchrone REPOSTP" à la page 376](#)) permet de masquer temporairement le point de redémarrage défini dans le sous-programme asynchrone.

6.8.5 Sous-programmes asynchrones : inscription "ASPSET"

Description : Inscrit un sous-programme asynchrone sur le canal actuellement actif et l'active (pour l'activation, voir aussi la fonction "ASPENA", [chap. 6.8.3 "Sous-programmes asynchrones : mise en service ASPENA" à la page 237](#)).

Seuls les sous-programmes inscrits et en service peuvent être utilisés.



Pour désinscrire des sous-programmes asynchrones inscrits, voir la fonction "ASPCLR", [chap. 6.8.1 "Sous-programmes asynchrones : désinscription ASPCLR" à la page 236](#).

Fonctions CN avec syntaxe de langage standard



Vous trouverez toutes les informations sur l'utilisation et la programmation de sous-programmes asynchrones dans la documentation "Description des fonctions IndraMotion MTX".

Syntaxe :

ASPSET (<NoSP> , <NomSP>{, <drapeaux>})	
avec	
<NoSP>	Numéro du sous-programme Champ de saisie : 1 à 8. Nombre entier
<NomSP>	Nom du sous-programme avec ou sans indication de chemin Si le chemin n'est pas indiqué, le fichier est recherché conformément au paramètre NCO/FileOrg/SrchPathSubProg (MP 3080 00001) (chemin de recherche pour les sous-programmes).
<drapeaux>	Commutateurs avec lesquels il est possible d'influencer le comportement après l'appel : 00 : Ni processus de liaison, ni démarrage automatique. 10 : Processus de liaison, mais pas de démarrage automatique. 01 : Pas de processus de liaison, mais démarrage automatique. 11 : Processus de liaison et démarrage automatique.

Tab. 6-16: Syntaxe ASPSET

Exemple :

```

:
N30 AS-      Inscription du programme ASUP1 en tant que premier sous-
PSET (1,ASUP1,10) programme asynchrone sur le canal actif. Après avoir été ap-
:           pelé (par ex. par le signal d'interface), il est lié automatique-
           ment (si nécessaire) et démarré explicitement via le démarra-
           ge CN.

```

6.8.6 Sous-programmes asynchrones : déclenchement par un programme "ASPSTA"

Description : Appelle un sous-programme asynchrone dans un canal quelconque par pilotage de programme.



Vous trouverez toutes les informations sur l'utilisation et la programmation de sous-programmes asynchrones dans la documentation "Description des fonctions IndraMotion MTX".

Syntaxe :

ASPSTA (<NoSP>{, <NoCanal>})	
avec	
<NoSP>	Numéro du sous-programme Champ de saisie : 1 à 8. Nombre entier
<NoCanal>	Canal cible dans lequel <NoSP> doit être appelé En l'absence de programmation, <NoSP> est appelé sur le canal actuel.

Tab. 6-17: Syntaxe ASPRTP

Particularités et restrictions :

- Les sous-programmes asynchrones ne doivent pas être imbriqués.

Fonctions CN avec syntaxe de langage standard

- Le sous-programme appelé doit être inscrit sur le canal cible (voir ASPSET, chap. 6.8.5 "Sous-programmes asynchrones : inscription ASPSET" à la page 238).
- Le sous-programme appelé ne peut pas être mis hors service sur le canal cible (voir ASPDIS, chap. 6.8.2 "Sous-programmes asynchrones : mise hors service ASPDIS" à la page 236 et ASPENA, chap. 6.8.3 "Sous-programmes asynchrones : mise en service ASPENA" à la page 237).

6.9 Attribution d'un nom d'axe logique "AssLogName, ALN"

Description : Attribue un nouveau nom d'axe logique à un axe synchrone sur le canal appelant. L'ancien nom logique d'axe perd sa validité.



Pour obtenir des informations détaillées sur la fonctionnalité "Transfert d'axe", voir la documentation "Description des fonctions IndraMotion MTX".

Syntaxe :

AssLogName (<NPA> <IPA> <NLA> , <NLAnouveau> { , <NPA> <IPA> <NLA> , <NLAnouveau>}...)	
Format abrégé : ALN (. .)	
avec	
<NPA>	Nom physique de l'axe Définit l'axe qui doit être renommé sur le canal actuel.
<IPA>	Index physique de l'axe Effet similaire à <NPA>
<NLA>	Nom logique de l'axe Effet similaire à <NPA>
<NLAnouveau>	Nouveau nom logique de l'axe L'axe indiqué à l'aide de <NPA> , <IPA> ou <NLA> reçoit le nom logique <NLAnouveau> sur le canal actuel. <NLAnouveau> doit être défini dans le paramètre CHAN/Ch/Coord/Acs/ChAx/ChAxName (MP 7010 00010) (nom de l'axe du canal) ou dans le paramètre CHAN/Ch/Coord/Acs/OptAxName/OptChAxName (MP 7010 00020) (désignation optionnelle de l'axe).

Tab. 6-18: Syntaxe AssLogName (ALN)

Exemple :

```
:
N030  ALN (YP,X, L'axe physique YP reçoit le nom logique X, le 3e axe physique
3,Y,B,Z)         reçoit le nom logique Y et l'axe logique B reçoit le nom logi-
:                que Z.
                La programmation ultérieure de B entraîne une erreur d'exécu-
                tion.
```

Particularités et restrictions :

- Un axe à renommer doit être arrêté. Si ce n'est pas le cas, la commande génère un message d'erreur et interrompt le programme.
- Les positions d'axe d'un même bloc doivent toujours être programmées d'après l'expression "AssLogName(...)".

6.10 Calibrage des cinématiques d'axe

6.10.1 Transformation en arrière "ATBWD"

Description : Sert à calculer les nouvelles coordonnées du système de coordonnées de la pièce à usiner (WCS) en positions d'axe réelles (ACS). Ce processus est appelé "Transformation en arrière".

Syntaxe :

AtBwd (<CoordAxe>, <coord>)	
avec	
<CoordAxe>	Tableau CPL permanent, global ou local. Type : Double Dimension : Au moins égal au nombre d'axes sur le canal actuel. Chaque variable du tableau contient les positions d'axe réelles dans l'ACS après l'appel.
<coord>	Tableau CPL permanent, global ou local. Type : Double Dimension : Au moins égal au nombre de coordonnées du canal dans le système de coordonnées généré par la transformation d'axe. Les variables du tableau doivent contenir les coordonnées de la pièce à usiner de toutes les coordonnées participant à la transformation d'axe dans le WCS.

Tab. 6-19: Syntaxe AtBwd

Particularités et restrictions : La transformation en arrière est exécutée pour la transformation d'axe active au moment de l'appel. Les paramètres de longueur et d'angle requis pour cela sont issus des paramètres machine.

6.10.2 Optimisation des paramètres "ATCAL"

Description : Associée à la fonctionnalité "Calibrer les cinématiques d'axe", sert à optimiser les paramètres d'angle et de longueur spécifiques à la cinématique d'axe. Ces données sont différentes pour chaque type de transformation d'axe et contenues dans le paramètre TRA/AxTrafo/JointParTrafo (MP 1030 00140) pour chaque transformation d'axe. En règle générale, elles doivent être lues avant l'optimisation avec la fonction "ATGET" (voir [chap. 6.10.4 "Lecture des paramètres de la commande numérique ATGET" à la page 244](#)), puis transférées de nouveau vers la commande numérique après l'optimisation par la fonction "ATPUT" (voir [chap. 6.10.5 "Écriture de paramètres dans la commande numérique ATPUT" à la page 246](#)).



Vous trouverez les informations complètes à propos du calibrage des cinématiques d'axe dans la documentation "Description des fonctions IndraMotion MTX".

Syntaxe :

ATCAL (<fichier>, <DonnéesOpt>, <masque>{, <info>}{, <NbIt>})	
avec	

Fonctions CN avec syntaxe de langage standard

<fichier>	Nom du fichier de calibrage ; avec ou sans indication de chemin. Si le chemin n'est pas indiqué, le fichier est recherché conformément au paramètre NCO/FileOrg/SrchPathSubProg (MP 3080 00001) (chemin de recherche pour les sous-programmes). <fichier> contient les données nécessaires à l'optimisation. Pour plus d'informations à ce sujet, voir la documentation "Description des fonctions IndraMotion MTX".
<Donnée-sOpt>	Tableau CPL permanent, global ou local. Type : Double Dimension : min. 16 Après le calcul, le tableau contient un bloc de paramètres avec les paramètres de longueur et d'angle optimisés. L'ordre des variables individuelles (index 1 à 16) correspond ici à l'index des éléments du paramètre TRA/AxTrafo/JointParTrafo (MP 1030 00140). Les valeurs ont la même unité que les paramètres individuels dans le paramètre TRA/AxTrafo/JointParTrafo (MP 1030 00140).
<masque>	Variable CPL Type : Nombre entier Masque de bits avec lequel les paramètres individuels à optimiser sont déterminés. Exemple :

← Tableau <OptData> avant l'optimisation →

Index:

1	2	3	4	5	6	7	8	...	15	16

Masque de bit: 1 = Optimiser un paramètre individuel ...

1	1	1	0	1	0	0	0	...	0	0

Priorité: 1 2 4 8 16 32 64 128 ...

Valeur requise en <Masque>: 1 + 2 + 4 + 16 = 23

← Tableau <OptData> après l'optimisation →

Index:

1	2	3	4	5	6	7	8	...	15	16

: Paramètre individuel optimisé

Fig. 6-2: Masque de bits

Fonctions CN avec syntaxe de langage standard

<info>	Tableau CPL permanent, global ou local. Type : Double. Dimension : min. 4. Contient les données suivantes après l'optimisation : <info>[1] Valeur de la fonction de qualité (écart quadratique) avant l'optimisation. Unité : mm <info>[2] Valeur de la fonction de qualité (écart quadratique) après l'optimisation. Unité : mm <info>[3] Écart max. avant l'optimisation (en mm) <info>[4] Écart max. après l'optimisation (en mm)
<Nblt>	Nombre maximal des pas d'itération pour l'optimisation. Dans la mesure où l'itération n'est pas indiquée ou indiquée avec "-1", elle n'est terminée que si l'écart entre deux blocs de paramètres calculés successivement dépasse un seuil interne à la commande numérique suffisamment petit (il n'existe plus aucun écart pertinent). Si "1" est indiqué, le calcul de compensation linéaire intervient.

Tab. 6-20: Syntaxe ATCAL

Exemple :

01 DIM PAR! (16)	Dimensionner le tableau CPL local avec 16 champs de type DOUBLE (pour les paramètres de longueur/d'angle optimisés).
02 DIM GA! (4)	Dimensionner le tableau CPL local avec 4 champs de type DOUBLE (pour la qualité et l'écart).
03 MASK%=2+4+32	Optimisation des paramètres individuels avec les indices de tableau 2, 3 et 6.
N4 ATCAL (CL.TXT, [PAR!], [MASK%], [GA!], -1)	Démarrer l'optimisation

6.10.3 Transformation en avant "ATFWD"

Description : Associée à la fonctionnalité "Calibrer les cinématiques d'axe", sert à recalculer les positions d'axe réelles dans les coordonnées d'un système de coordonnées généré lors de la transformation d'axe. Ce processus est également appelé "Transformation en avant".



Vous trouverez les informations complètes à propos du calibrage des cinématiques d'axe dans la documentation "Description des fonctions IndraMotion MTX".

Syntaxe :

ATFWD (<coord>, <CoordAxe>{, <DonnéesPar>})
avec

Fonctions CN avec syntaxe de langage standard

<coord>	Tableau CPL permanent, global ou local. Type : Double Dimension : Au moins égal au nombre de coordonnées du canal dans le système de coordonnées généré par la transformation d'axe. Les variables individuelles du tableau contiennent les coordonnées du canal résultant de la conversion dans le système de coordonnées transformé.
<CoordAxe>	Tableau CPL permanent, global ou local. Type : Double Dimension : Au moins égal au nombre d'axes sur le canal actuel. Les variables du tableau doivent contenir les positions d'axe réelles de tous les axes du canal qui participent à la transformation d'axe.
<DonnéesPar>	Tableau CPL permanent, global ou local. Type : Double Dimension : min. 16. Le tableau doit contenir un bloc de paramètres contenant tous les paramètres de longueur et d'angle pour une transformation d'axe. L'ordre de chaque variable (index 1 à 16) correspond à l'index des éléments de MP 1030 00140. Les valeurs doivent présenter la même unité que les paramètres individuels correspondants dans MP 1030 00140. En l'absence de programmation, le bloc de paramètres utilisé est celui de la transformation d'axe actuellement active. Dans la mesure où deux transformations d'axes sont actives simultanément (AT1, AT2, voir fonction "Coord" (chap. 6.28 "Sélection d'une transformation d'axe Coord, CRD" à la page 274), les données utilisées sont celle de AT2.

Tab. 6-21: Syntaxe ATFWD

Particularités et restrictions : Si <DonnéesPar> a été programmé et qu'aucune transformation d'axe n'est active, une erreur d'exécution s'affiche.

6.10.4 Lecture des paramètres de la commande numérique "ATGET"

Description : Associée à la fonctionnalité "Calibrer les cinématiques d'axe", sert à lire les paramètres machine spécifiques à la transformation d'axe.

Ces données diffèrent pour chaque type de transformation d'axe. En règle générale, elles doivent être lues avant l'optimisation (voir fonction "ATCAL", [chap. 6.10.2 "Optimisation des paramètres ATCAL" à la page 241](#)), puis transférées de nouveau vers la commande numérique après l'optimisation avec la fonction ATPUT (voir [chap. 6.10.5 "Écriture de paramètres dans la commande numérique ATPUT" à la page 246](#)).



Vous trouverez les informations complètes à propos du calibrage des cinématiques d'axe dans la documentation "Description des fonctions IndraMotion MTX".

Syntaxe :

```
AtGet (<DonnéesPar>{, <NoTransfoAxe>}{, <NoParamMach>})
```

avec

Fonctions CN avec syntaxe de langage standard

<DonnéesPar>	Tableau CPL permanent, global ou local. Type : Double Dimension : min. 8 (NoParamMach = 1030 00130) ou min. 16 (NoParamMach = 1030 00140). Une fois la fonction exécutée, le tableau contient les valeurs actuellement valables du paramètre machine "NoParamMach" de la transformation d'axe "NoTransfoAxe". L'ordre des variables individuelles (index 1, 2...) correspond à l'index des éléments du paramètre machine associé. Les valeurs ont la même unité que le paramètre machine correspondant.
<NoTransfoAxe>	Numéro de la transformation d'axe dont le paramètre doit être lu. En l'absence de programmation, les données lues sont celles de la transformation d'axe actuellement active. Dans la mesure où deux transformations d'axes sont actives simultanément (AT1, AT2, voir fonction "Coord" (chap. 6.28 "Sélection d'une transformation d'axe Coord, CRD" à la page 274), les données utilisées sont celle de AT2.
<NoParamMach>	Numéro du paramètre machine qui est lu. En l'absence de programmation, la valeur 1030 00140 est enregistrée par défaut, c.-à-d. que les paramètres d'angle et de longueur sont lus.

Tab. 6-22: Syntaxe AtGet

Exemple :

01 DIM LENPARAM! (16)	Tableau CPL local contenant 16 éléments pour les paramètres de longueur et d'angle
01 DIM ZEROPOS! (8)	Tableau CPL local contenant 8 éléments pour les positions de référence des axes du canal
	Les valeurs suivantes sont lues :
N1 ATGET (LENPARAM!)	MP 1030 00140 de la transformation d'axe actuelle
N1 ATGET (LENPARAM!, 3)	MP 1030 00140 de la transformation d'axe 3
N1 ATGET (LENPARAM!, 2, 103000140)	MP 1030 00140 de la transformation d'axe 2
N1 ATGET (ZEROPOS!, , 103000130)	MP 1030 00130 de la transformation d'axe actuelle
N1 ATGET (ZEROPOS!, 3, 103000130)	MP 1030 00130 de la transformation d'axe 3

Particularités et restrictions :

- Les valeurs admissibles pour <NoParamMach> sont 1030 00130 (positions d'axe de la position de référence) et 1030 00140 (paramètres d'angle et de longueur).
- L'ordre "ATGET" entraîne une erreur d'exécution, si NoTransfoAxe n'a pas été programmé et qu'aucune transformation d'axe (COORD(0,2) et COORD(0,1)) n'est active.

6.10.5 Écriture de paramètres dans la commande numérique "ATPUT"

Description : Associée à la fonctionnalité "Calibrer les cinématiques d'axe", sert à écraser les paramètres machine spécifiques à la transformation d'axe.

Ces données diffèrent pour chaque type de transformation d'axe. En règle générale, elles doivent être lues avant l'optimisation (voir fonction "ATCAL", chap. 6.10.2 "Optimisation des paramètres ATCAL" à la page 241) (voir fonction "ATGET", chap. 6.10.4 "Lecture des paramètres de la commande numérique ATGET" à la page 244), puis transférées de nouveau vers la commande numérique après l'optimisation avec la fonction "ATPUT".

 Vous trouverez les informations complètes à propos du calibrage des cinématiques d'axe dans la documentation "Description des fonctions IndraMotion MTX".

Syntaxe : AtPut (<DonnéesPar>{, <NoTransfoAxe>}{,<NoParamMach>})	
avec	
<DonnéesPar>	Tableau CPL permanent, global ou local. Type : Double Dimension : min. 8 (NoParamMach = 1030 00130) ou min. 16 (NoParamMach = 1030 00140). Le tableau doit contenir un bloc de paramètres complet du paramètre machine "NoParamMach" de la transformation d'axe "NoTransfoAxe". L'ordre des variables individuelles (index 1, 2...) correspond à l'index des éléments du paramètre machine associé. Les valeurs ont la même unité que le paramètre machine correspondant.
<NoTransfoAxe>	Numéro de la transformation d'axe dont le paramètre doit être écrit. En l'absence de programmation, les données de la transformation d'axe actuellement active sont écrasées. Dans la mesure où deux transformations d'axes sont actives simultanément dans la commande (AT1, AT2 ; voir fonction "Coord", chap. 6.28 "Sélection d'une transformation d'axe Coord, CRD" à la page 274), les données de AT2 sont modifiées.
<NoParamMach>	Numéro du paramètre machine qui est écrasé. En l'absence de programmation, la valeur 1030 00140 est enregistrée par défaut, c.-à-d. que les paramètres d'angle et de longueur sont modifiés.

Tab. 6-23: Syntaxe AtPut

Exemple :	01 DIM LENPARAM! (16)	Tableau CPL local contenant 16 éléments pour les paramètres de longueur et d'angle
	01 DIM ZEROPOS! (8)	Tableau CPL local contenant 8 éléments pour les positions de référence des axes du canal
	01 LENPARAM! (1) = 123.456 :	Attributions de valeur
	01 ZEROPOS! (8) = 1.0	Les valeurs suivantes sont écrasées :

Fonctions CN avec syntaxe de langage standard

N1 ATPUT (LENPARAM!)	MP 1030 00140 de la transformation d'axe actuelle
N1 ATPUT (LENPARAM!, 3)	MP 1030 00140 de la transformation d'axe 3
N1 ATPUT (LENPARAM!, 2, 103000140)	MP 1030 00140 de la transformation d'axe 2
N1 ATPUT (ZEROPOS!,, 103000130)	MP 1030 00130 de la transformation d'axe actuelle
N1 ATPUT (ZEROPOS!, 3, 103000130)	MP 1030 00130 de la transformation d'axe 3

- Particularités et restrictions :**
- Les valeurs admissibles pour <NoParamMach> sont 1030 00130 (positions d'axe de la position de référence) et 1030 00140 (paramètres d'angle et de longueur).
 - L'ordre "ATPUT" entraîne une erreur d'exécution, si NoTransfoAxe est manquant et qu'aucune transformation d'axe (COORD(0,2) et COORD(0,1)) n'est active.
 - L'exécution de la fonction "ATPUT" nécessite un niveau utilisateur CN permettant l'écrasement des paramètres machine.
 - Le bloc de paramètres inscrit ne devient actif qu'après la remise à zéro du système **et** la nouvelle programmation de COORD (<NoTransfoAxe>).

6.11 Exécution des fonctions auxiliaires actives de tous les groupes "AUXFUNC"

Description : Exécute les fonctions auxiliaires spécifiques au canal et intercanaux actuellement actives de tous les groupes de fonctions auxiliaires existants.

La fonction est pertinente en association avec l'avance de bloc (par ex. après une interruption de traitement) pour rétablir tous les états des fonctions auxiliaires à une position donnée du programme pièce.

Par exemple, si un programme pièce est interrompu au cours de son traitement, l'avance de bloc permet, dans certaines circonstances, de reprendre le traitement à partir du bloc qui était traité au moment de l'interruption.

Dans ce cas, le programme pièce est redémarré depuis le début avec l'avance de bloc, mais le traitement sur la machine ne s'effectue qu'à partir d'un bloc-programme défini.

Étant donné que la commande n'interpole pas et n'émet pas non plus de fonctions auxiliaires programmées pendant l'avance de bloc, tous les états des fonctions auxiliaires doivent impérativement être rétablis à la fin de l'avance de bloc.



Vous trouverez des informations complètes à propos de la mise en œuvre et du paramétrage des fonctions auxiliaires dans la documentation "Description des fonctions IndraMotion MTX".

Syntaxe :

AUXFUNC

Tab. 6-24: Syntaxe AUXFUNC

Fonctions CN avec syntaxe de langage standard

Exemple :	N100 T102	Sélectionner l'outil T102
	N110 M6	Remplacer l'outil T102
	:	
	N150 M3	Démarrer la 1re broche/le 1er groupe de broches
	:	Rotation à droite

Interruption du programme dans N160 :

- T102 et M3 étaient actifs.
- T102 est remplacé. La position de l'outil n'est pas modifiée.

Ensuite, le programme est sélectionné de nouveau et se poursuit sans mouvement de déplacement avec l'avance de bloc jusqu'à N150 inclus. Les fonctions auxiliaires programmées sont activées sans être utilisées.

Avant de traiter N160, les fonctions auxiliaires spécifiques au canal et inter-canaux actuellement actives de tous les groupes de fonctions auxiliaires sont exécutées avec AUXFUNC.

Dans cet exemple, AUXFUNC a l'effet suivant :

- Activer l'outil numéro T102 (l'outil est encore remplacé).
- Mettre en service la 1re broche/le 1er groupe de broches avec une rotation à droite.

6.12 Modification de l'accélération maximale des axes "AxAcc, AAC", enregistrement temporaire de l'accélération maximale des axes "AxAccSave, AAS"

- Description :**
- **AxAcc :**
Modifie les limites supérieures des accélérations maximales des axes. Pour cela, la fonction masque les valeurs d'accélération maximales des axes issues des paramètres machine avec les valeurs programmées.
 - **AxAccSave :**
Enregistre temporairement les valeurs d'accélération maximales actuelles de tous les axes dans une mémoire interne.
Cette mémoire interne est préinitialisée avec les valeurs des paramètres machine à chaque sélection du programme.

Syntaxe :

AxAccSave	
Format abrégé : AAS	Enregistrer temporairement toutes les valeurs d'accélérations actuelles des axes.
AxAcc ou AxAcc (1)	Réactiver les accélérations des axes enregistrées plus tôt à l'aide de "AxAccSave".
AxAcc(<valeurs> {,<référence>})	
	Modifier les accélérations maximales des axes
AxAcc () ou AxAcc (0)	Réactiver les accélérations des axes à partir des paramètres machine.
Format abrégé : AAC (. .)	
avec	

<valeurs>	Nom de l'axe et valeur de l'accélération Les indications pour plusieurs axes sont séparées par une virgule. En fonction de l'unité de mesure (G71/G70), la commande interprète les valeurs programmées en "1000 pouces/s²" ou "m/s²".
<référence>	En l'absence de programmation : • RapidAxAcc > 0.000 : L'accélération de l'avance est modifiée. • RapidAxAcc = 0.000 : Les valeurs de l'accélération sont modifiées dans le même sens en modes d'avance et d'avance rapide. RAP : L'accélération est modifiée pour l'avance rapide. ALL : Les accélérations sont modifiées pour l'avance et pour l'avance rapide.

Exemple : Situation de départ :

$$\vdots$$

Accélération max. de l'avance pour l'axe X : 1.0 m/s²

Accélération max. de l'avance pour l'axe Z : 2.1 m/s²

Les valeurs d'accélération de l'avance rapide restent inchangées à 10 m/s^2 .

$$\vdots$$

Enregistrer temporairement toutes les valeurs d'accélération maximales de l'axe actuellement actives.

Réactiver les valeurs issues des paramètres machine :

Valeurs d'accélération max. de l'avance (X,Y,Z) : 8 m/s²

Valeurs d'accélération max. de l'avance rapide (X,Y,Z) :
10 m/s²

•

Accélération max. de l'avance et de l'avance rapide de

l'axe Y : 5.0 m/s^2

Les accélérations maximales pour les axes X et Z restent inchangées.

:

Réactiver les valeurs enregistrées à l'aide de "AxAccSave" :

Accélération max. de l'avance pour l'axe X : 1.0 m/s²

Accélération max. de l'avance pour l'axe Y : 8.0 m/s²

Accélération max. de l'avance pour l'axe Z : 2.1 m/s²

Toutes les valeurs d'accélération de l'avance rapide sont définies sur 10.0 m/s^2 .

Fonctions CN avec syntaxe de langage standard

- Particularités et restrictions :**
- Si aucune valeur d'accélération séparée n'est réglée pour le mode d'avance rapide (AX/Dr[i]/Acc/RapidAxAcc = 0) et que le paramètre RAP n'est pas programmé, la fonction AxAcc/AAC modifie les valeurs d'accélération dans le même sens en modes d'avance et d'avance rapide. Dans ce cas, le comportement est compatible avec MTX V11 et les versions antérieures.



Les expressions entre parenthèses "(0)" et "(1)" ne sont pas des paramètres de la fonction CN. La valeur numérique entre parenthèses ne peut donc pas être transmise par des variables CPL.

6.13 Chargement du tableau de correction des erreurs d'axe "Ax-CompTab"

Description : La fonction CN permet de remplacer les tableaux de correction des erreurs d'axe (LSEC et CCOMP).

S'il existe un tableau de base pour la correction des erreurs d'axe d'un axe et que ce dernier est identifié comme LOADABLE, il est possible de charger un tableau de correction alternatif à l'aide de la fonction "AxCompTab".



Vous trouverez des informations à propos de la compensation des imprécisions mécaniques à l'aide de la compensation de croix (CCOMP) et de la compensation des erreurs de pas de la broche (LSEC) dans la documentation "Description des fonctions IndraMotion MTX".

Syntaxe : Activer un tableau chargeable ultérieurement :

AxCompTab (<CAX> , <table>)	
avec	
<CAX>	Axe compensé Nom ou numéro d'un axe système.
<table>	Nom (avec indication du chemin) d'un tableau CCOMP ou LSEC

Tab. 6-26: Syntaxe AxCompTab - Tableaux chargeables ultérieurement

Activer un tableau de base :

AxCompTab (<CAX> , <RAX>)	
avec	
<CAX>	Axe compensé Nom ou numéro d'un axe système.
<RAX>	Axe de référence Nom ou numéro d'un axe système. CAX et RAX sont identiques : tableau LSEC CAX et RAX sont différents : tableau CCOMP

Tab. 6-27: Syntaxe AxCompTab – Tableau de base

Activer tous les tableaux de base :

Activer tous les tableaux de base
AxCompTab (<CAX>)

Fonctions CN avec syntaxe de langage standard

avec	
<CAX>	Axe compensé Nom ou numéro d'un axe système.

Tab. 6-28: Syntaxe AxCompTab – Tous les tableaux de base

Exemple : Le tableau de base "cnc03r02.tab" contient uniquement les mots-clés permettant de valider les tableaux chargeables ultérieurement et non des valeurs de correction.

Le tableau "Reload0302.tab" est activé si nécessaire via la fonction CN "AxCompTab".

L'axe système 2 se nomme YA et l'axe 3 se nomme ZA.

Tableau CCOMP de base cnc03r02.tab

```
#STUP
LOADABLE
EMPTY
#ESTUP
```

Tableau CCOMP chargeable Reload0302.tab

```
#CONF
CAX 3 ; ZA est l'axe corrigé
RAX 2 ; YA est l'axe de référence
#ECONF
-20000.0 50000.0 ; Valeur initiale -20 mm, longueur de
pas 50 mm
7.0 7.0 ; Correction pour -20 mm
9.0 9.0 ; Correction pour 30 mm
9.8 9.8 ; Correction pour 80 mm
9.0 9.0 ; Correction pour 130 mm
7.0 7.0 ; Correction pour 180 mm
4.0 4.0 ; Correction pour 230 mm
0.0 0.0 ; Correction pour 280 mm
```

:

```
N010 AxCompTab(ZA, Re-      Le tableau CCOMP Reload0302.tab est actif.
load0302.tab)
```

:

```
N200 AxCompTab(ZA, YA)      Le tableau CCOMP cnc03r02.tab est actif.
```

:

Particularités et restrictions :

- Les tableaux des axes du canal concerné et des axes asynchrones peuvent être échangés.
- Le nom du tableau respectivement actif est enregistré dans une composante de la donnée système "SysAxActData[]".
- Un mouvement de l'axe se produit lors du changement de tableau !

Fonctions CN avec syntaxe de langage standard

Le mouvement est limité à max. 2 mm lors du chargement d'un tableau. Si un mouvement plus important est nécessaire, une erreur d'exécution est émise.

Le mouvement n'est pas limité en retournant vers le(s) tableau(x) de base.

- La fonction "AxComTab" n'est **pas exécutée dans l'avance de bloc**.

6.14 Couplage d'axes "AxCouple, AXC"

Description : Le couplage d'axes permet d'établir une relation particulière entre le mouvement d'un "axe maître" synchrone et d'un ou plusieurs (max. 7) "axes esclaves" synchrones.

Si les axes esclaves ne se trouvent pas sur la position de couplage au moment de l'activation du couplage, ils sont déplacés à cette position par un mouvement de déplacement linéaire généré en interne. Pour cela, l'avance sur trajectoire active et le potentiomètre d'Override sont effectifs.

Si l'axe maître se déplace, tous les axes esclaves bougent automatiquement selon le propre rapport avec l'axe maître défini respectivement pour chacun d'eux. C'est pourquoi l'axe maître et tous les axes esclaves participants sont appelés également "Groupe d'axes".

Rapports possibles entre l'axe maître et un axe esclave :

- Les positions de consigne de l'axe maître sont transformées en positions de consigne correspondantes des axes esclaves à l'aide d'un **décalage constant** (voir formule 1).

L'axe peut ainsi être déplacé par rapport à l'axe maître sur un itinéraire quelconque, toujours constant, en direction positive ou négative.

- Les positions de consigne de l'axe maître sont transformées en positions de consignes correspondantes des axes esclaves à l'aide d'un **facteur de couplage constant** (voir formule 1).

Il est ainsi possible de bouger l'axe esclave par rapport à l'axe maître selon un rapport défini.

- Les positions de consigne de l'axe maître sont transformées en positions de consigne **quelconques** correspondantes de l'axe esclave à l'aide d'une table (de couplage). Dans ce cas, des paires de repères sont déposées dans la table indiquant la position correspondante de l'axe esclave pour une position de l'axe maître et un décalage éventuellement nécessaire de l'axe maître (voir formule 2).

La commande peut déterminer les positions entre chaque repère à l'aide de la fonction spline cubique.

Fonctions CN avec syntaxe de langage standard

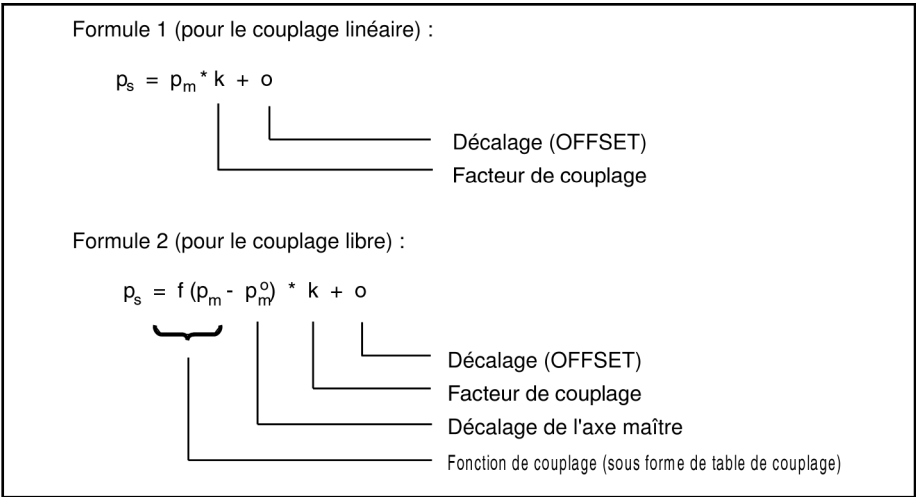


Fig. 6-3: Formules pour le couplage d'axes

Tous les rapports mentionnés peuvent être combinés à volonté.

Ceci permet de réaliser très simplement des axes parallèles (par ex. pour des plateaux d'usinage disposés en parallèle) ou des entraînements électro-
niques (1 tour de l'axe maître engendre par ex. 10 tours de l'axe esclave).

Les points suivants s'appliquent ici :

- Tous les axes d'un groupe d'axes doivent se trouver sur le même canal.
- Plusieurs groupes d'axes sont autorisés sur un même canal.



Vous trouverez des informations détaillées à propos de la fonctionnalité "Couplage d'axes" dans la documentation "Description des fonctions IndraMotion MTX".

Vous y trouverez également des informations concernant la table de couplage requise pour les "couplages libres".

Syntaxe :

AxCouple (<M><variante>, <S> ({<SO>}, {<SF>}, {<MO>}, <tab>}) {, ...})	
AxCouple() ou AxCouple(0)	Effacer tous les groupes d'axes sur le canal actuel.
Format abrégé : AXC (. .)	
avec	
<M>	Adresse logique de l'axe maître
<variante>	0 Créer un nouveau groupe d'axes 1 Modifier un groupe d'axes (ajouter des nouveaux axes esclaves ou modifier la désignation du couplage) -1 Effacer l'axe esclave ou supprimer complètement le groupe

Fonctions CN avec syntaxe de langage standard

<S>	Adresse logique de l'axe esclave Si <S> ne peut pas être distingué de la programmation d'un axe/ d'une coordonnée avec valeur, un signe d'égalité doit être inséré en- tre <S> et "(" : (...,<S>=(...)) Voir également l'exemple ci-dessous.
<SO>	Décalage de l'axe esclave : - Écart de couplage en mm ou en pouces ACT pour un couplage sur la position actuelle En l'absence de programmation, <SO> = 0 s'applique.
<SF>	Facteur de couplage de l'axe esclave. En l'absence de programmation, <SF> = 1 s'applique.
<MO>	Décalage de l'axe maître Uniquement pertinent en cas de "couplage libre" (voir formule 2 ci- dessus). En l'absence de programmation, <MO> = 0 s'applique.
<tab>	Nom de la table de couplage de l'axe esclave. Uniquement pertinent en cas de "couplage libre" (voir formule 2 ci- dessus).

Tab. 6-29: Syntaxe AxCouple

Exemples :	N100 AXC (C0 , A () , B ()) Créer un groupe d'axes C : axe maître A/B : axes esclaves Les deux axes esclaves fonctionnent avec un coupla- ge "linéaire". Les valeurs par défaut Écart de couplage 0.0 et Fac- teur de couplage 1.0 sont actives. N100 AXC (C0 , A (ACT) , B (ACT, 2)) Créer un groupe d'axes C : axe maître A/B : axes esclaves Les deux axes esclaves fonctionnent avec un coupla- ge "linéaire". A et B sont couplés à C sans mouvement d'approche. B est utilisé avec le facteur de couplage 2. N100 AXC (C0 , B (, 0.5 , , T_B)) Créer un groupe d'axes C : axe maître B : axe esclave L'esclave B fonctionne avec un couplage "libre". :

Fonctions CN avec syntaxe de langage standard

N150 AXC (C1, A (4.5, 2)) Élargir le groupe d'axes
:
A est ajouté en tant qu'axe esclave.
L'axe esclave A fonctionne avec un couplage "linéaire".
L'axe esclave B continue de fonctionner avec un couplage "libre".

N200 AXC (C-1, A ()) L'axe esclave A est retiré du groupe d'axes C.
:
N200 AXC (C-1) Le groupe d'axes C complet est supprimé.
:

Autre exemple :

Axe maître X, axe esclave Y2

L'axe esclave a un nom d'axe ambigu, Y2, qui ne se distingue pas de la programmation de l'axe Y avec la valeur 2.

AXC doit donc être programmée comme suit :

AXC (X, Y2= (. . . .))



La fonction déclenche un mouvement de déplacement de tous les axes esclaves programmés dans ce bloc !

Les axes esclaves se déplacent alors vers leur point de couplage spécifique (valeur guide) qui est défini par la position du maître et le rapport de couplage.

Particularités et restrictions :

- Tous les axes participant à un groupe d'axes doivent être des axes synchrones, au moins au moment du couplage des axes.
Les axes asynchrones ou les axes Hirth ne sont pas admis en tant qu'axes esclaves d'un groupe de couplage.
- Aucun mouvement de déplacement ne doit être programmé pour les axes esclaves ; cela génère un message d'erreur.
- Un axe esclave ne peut pas être en même temps l'axe maître d'un autre groupe d'axes.
- La fin du programme ne dissocie pas automatiquement un couplage d'axes existant.
- Si l'axe maître est un axe modulo, l'axe esclave doit également être un axe modulo s'il s'agit d'un rapport de couplage linéaire.
- Afin de permettre une approche des points de référence axe par axe, le couplage d'axes doit être ouvert.
- La zone de déplacement autorisée de l'axe maître peut être réduite par les axes esclaves couplés (par ex. si l'axe esclave atteint l'extrémité de ses zones avant l'axe maître ou si la zone de déplacement de l'axe esclave est plus petite que la zone de déplacement de l'axe maître).
- La dynamique maximale de l'axe "le plus faible" détermine la dynamique maximale du groupe d'axes complet.
- Le verrouillage d'axes est interdit quand un groupe d'axes est actif.
- Des axes couplés en mode test doivent être découplés avant la désactivation du mode test.

Fonctions CN avec syntaxe de langage standard

6.15 Couplage d'avance d'axes "AxisFeedCouple, AFC"

Description : Déduit l'avance pour un axe asynchrone à partir de la vitesse d'un axe ou d'une broche de référence. La fonction permet par exemple d'indiquer une avance de rotation (G95) pour des axes asynchrones. Le couplage est effectif en mode de positionnement comme en mode d'asservissement en vitesse de rotation. L'avance qui en résulte est calculée de la façon suivante :

$\text{Avance}[\text{axe asynchrone}] = |\text{Avance}[\text{axe de référence}]| \cdot \text{facteur de couplage}$

De plus, l'avance peut également être mise à l'échelle via l'Override d'axe.

Syntaxe :

AxisFeedCouple, AXISFEEDCOUPLE	
	Activer/Désactiver le couplage d'avance
Format abrégé : AFC (. . .)	
AFC (FRA=<axe de référence>, FA=<axe d'avance>, CF=<facteur>)	
Paramètres du couplage d'avance :	
	Couplage d'avance avec référence de l'axe système
AFC (FRA=<axe>, FA=<axe>, CF=<facteur>)	
	Couplage d'avance avec référence de la broche du canal n = 1 à 8
AFC (FRA=Sn, FA=<axe>, CF=<facteur>)	
	Couplage d'avance avec référence de la broche du système nn = 00 à 32
AFC (FRA=<broche du système>, FA=<axe>, CF=<facteur>)	
G95	Couplage d'avance avec référence actuelle G95
AFC (FA=<axe>, CF=<facteur>)	
AFC (FA=<axe>)	Désactiver le couplage d'avance pour <axe>
AFC ()	Désactiver les couplages d'avance de tous les axes du système
	La commande reprend une interpolation d'axe active avec l'avance programmée (valeur S/FA) ou paramétrée (vitesse de l'axe max.).
avec	
FRA=<axe de référence>	L'axe de référence peut être un axe du système, une broche du canal (S1 à S8) ou une broche du système. En l'absence de paramètre, la commande enregistre la broche de référence G95 ou l'axe de référence G95 actif.

Fonctions CN avec syntaxe de langage standard

FA=<axe d'avance>	Axe sans fin, rond ou linéaire asynchrone
CF=<facteur> ou CF=S(<facteur>)	Met la vitesse d'axe de référence à l'échelle de la vitesse d'avance requise. La vitesse d'axe de référence est disponible en °/min pour les axes ronds et les broches et en mm/min pour les axes linéaires. Les points suivants s'appliquent à une combinaison d'axe linéaire, rond et de broche : 1°/min correspond à 1 mm/min. Pour un axe de référence sans fin qui est l'axe esclave d'un couplage d'axes système, la vitesse de rotation "corrigée" peut être utilisée avec l'attribut "S(<facteur>)" pour une avance de tour régulière. Les parties de l'axe de positionnement ou de la table du couplage d'axes système ne sont pas comprises dans la vitesse de rotation "corrigée". Le <facteur> doit être supérieur à 0. Le facteur de couplage d'avance peut être modifié de manière dynamique à l'aide des fonctions "G94/G95(DSi=) Programmation incrémentale de la vitesse".

Tab. 6-30: Syntaxe AxisFeedCouple, AFC



Mouvement involontaire de l'axe possible suite à une désactivation du couplage d'avance !

Les axes d'avance couplés à un axe de référence vertical peuvent se déplacer de nouveau suite à la désactivation du couplage.

Avant de désactiver le couplage, terminez les mouvements de déplacement asynchrones en remettant le système à zéro ou arrêtez les axes sans fin en mode d'asservissement en vitesse de rotation avec une donnée S(0).

Exemple :

```

; Taillage par outil-pignon :
; Axe de course Z3, axe d'outil C1, avance du cylindre FW [mm/tr Z3]
; Un tour de Z3 correspond à la double course (DH) de l'outil-pignon
; Diamètre d'outil 100 mm
; Avance du cylindre 0,2 mm/DH

1 PI=3.141593 : DIA=100 : FW=0.2
; Facteur de couplage=FW/circonférence C1

1 FC=FW/
(Pi*DIA)

; Couplage de l'avance pour l'axe C1

N20 AFC (FRA=Z3, FA=C1, CF=[FC])

N30 C1=S(-1) ; Démarrer le mouvement d'avance : C1=S(-(Z3*FC))

...

; Avance de tour pour l'axe linéaire asynchrone VA

N50 G95 F0.5 ; Activer l'avance de tour du canal

```

Fonctions CN avec syntaxe de langage standard

```

; Avance de tour pour axe linéaire asynchrone
; Axe VA : 1 mm/t
N60 VA=10 AFC (FA=VA, CF=[1/360])
...

; Avance de tour de l'axe de transmission C2=SAC(B2,X,Z)
; Avance VA 1 mm/tour C2,
; Avec partie B2, mais sans partie X et Z
N80 VA=0 AFC (FRA=C2, FA=VA, CF=S ([1/360]))
...

; Couplage de l'avance avec différence de vitesse de rotation en fonction de la course
N100 G1
G95 (FRA=B1) ; G95 avec 1,5 mm/tr B1
F1.5

; B2 avec vitesse de rotation B1 dans le sens
; opposé (1:1 – couplage)
N110 AFC (FA=B2, CF=1.0) B2=S (-1)

; Augmenter la vitesse de rotation de B2 jusqu'à la fin du bloc
; (1:2 – couplage)
N120 G95 (DSA1=B2, DS=1.0) X10
...

; Désactiver le couplage pour un axe
; Arrêter l'axe C1 et
N150 C1=S (0)
AFC (FA=C1) ; arrêter le couplage de l'avance
...

; Arrêter le couplage de l'avance de tous les axes système
; Arrêter le couplage de l'avance B2 et VA,
N170 AFC () ; Modification de la vitesse de rotation B2 à 1 tr/min

```

- Particularités et restrictions :**
- Une avance asynchrone programmée simultanément (<valeur> FA) limite l'avance maximale du couplage.
 - L'arrêt du couplage de l'avance en mode d'asservissement en vitesse de rotation ou au cours d'un processus de positionnement active la va-

Fonctions CN avec syntaxe de langage standard

leur S ou la valeur FA programmées. Le mouvement continue avec l'avance modifiée.

- Le couplage de l'avance conserve un effet modal à la fin du mouvement. Lors d'une reprogrammation de l'axe, l'axe se déplace avec le couplage existant.
- En mode d'asservissement en vitesse de rotation d'un axe sans fin, la commande évalue uniquement le signe de la valeur S indiquant le sens de rotation lorsque le couplage d'avance est activé. La hauteur de la vitesse de rotation est déterminée par la donnée de couplage.
- L'activation ou la désactivation du couplage de l'avance ne démarre pas ni n'arrête l'axe d'avance, mais commute l'avance au vol avec l'accélération d'axe autorisée.
- La vitesse prise en compte pour le couplage est la vitesse réelle pour les broches de référence et la vitesse de consigne pour les axes système de référence.
- Le couplage de l'avance ne garantit aucune synchronisation de la position des axes couplés.
- Le couplage de l'avance n'a aucun effet en mode de configuration.
- La commutation de l'avance du canal (G93/G94/G95) n'influence pas les couplages d'avance de l'axe actifs.

6.16 Modification de la secousse maximale des axes "AxJerk, AJK", enregistrement temporaire de la secousse maximale des axes "AxJerkSave, AJS"

- Description :**
- **AxJerk :**
Modifie les limites supérieures des valeurs maximales des axes. Pour cela, la fonction masque les valeurs maximales des axes issues des paramètres machine avec les valeurs programmées.
 - **AxJerkSave :**
Enregistre les valeurs de secousse maximales actuelles de tous les axes dans une mémoire interne.
Cette mémoire interne est préinitialisée avec les valeurs des paramètres machine à chaque sélection du programme.

Syntaxe :

AxJerkSave	
Format abrégé : AJS	Enregistrer temporairement toutes les valeurs de secousse actuelles des axes.
AxAcc ou AxAcc (1)	Réactiver les valeurs de secousse enregistrées plus tôt à l'aide de "AxJerkSave".
AxJerk (<valeurs> {,<référence>})	
	Modifier les valeurs de secousse maximales de l'axe
AxJerk () ou AxJerk (0)	Réactiver les valeurs de secousses des axes issues des paramètres machine.
Format abrégé : AJK (. .)	
avec	

Fonctions CN avec syntaxe de langage standard

<valeurs>	Noms de l'axe et valeurs de secousse Les indications pour plusieurs axes sont séparées par une virgule. En fonction de l'unité de mesure (G71/G70), la commande interprète les valeurs programmées en "1000 pouces/s³" ou "m/s³".
<référence>	<i>En l'absence de programmation :</i> • RapidAxJerk > 0.000 : La secousse de l'avance est modifiée. • RapidAxJerk = 0.000 : Les valeurs de secousse sont modifiées dans le même sens en modes d'avance et d'avance rapide. RAP : La secousse est modifiée pour l'avance rapide. ALL : La secousse est modifiée pour l'avance et pour l'avance rapide.

Tab. 6-31: Syntaxe AxJerk, AJK

Exemple : Situation de départ :

Les axes X à Z sont attribués par défaut dans les paramètres machine avec la secousse d'avance 40 m/s³ et la secousse d'avance rapide 60 m/s³.

$$\vdots$$

N10 JKC (1) Activer la limitation des secousses

$$\vdots$$

N40 Secousse max. de l'avance de l'axe X : 30.0 m/s³

AJK (X30.0, Z20.0) Secousse max. de l'avance de l'axe Z : 20.0 m/s³

$$\vdots$$

La secousse max. d'avance de l'axe Y reste inchangée (40 m/s^3).

Les valeurs de secousse d'avance rapide restent inchangées à 60 m/s³.

N80 AxJerkSave	Enregistrer temporairement toutes les valeurs de secousse maximales de l'axe actuellement actives.

N90 AJK () Réactiver les valeurs issues des paramètres machine :

:

Valeurs de secousse max. de l'avance (X,Y,Z) : 40.0 m/s³

Valeurs de secousse max. de l'avance rapide (X,Y,Z) :
60.0 m/s³

N150_AJK(Y35,ALL) Secousse max. de l'avance et de l'avance rapide de
: l'axe Y : 35.0 m/s³

$$:$$

N200_AJK(1) Réactiver les valeurs enregistrées à l'aide de "AxJerkSave":

Secousse max. de l'avance de l'axe X : 30.0 m/s³

Secousse max. de l'avance de l'axe Y : 40.0 m/s³

Secousse max. de l'avance de l'axe Z : 20.0 m/s³

Toutes les valeurs de secousse de l'avance rapide sont définies sur 60,0 m/s³.

Fonctions CN avec syntaxe de langage standard

- Particularités et restrictions :**
- Si aucune valeur de secousse n'est réglée à part pour le mode d'avance rapide (AX/Dr[j]/Acc/RapidAxJerk = 0) et que le paramètre RAP n'est pas programmé, la fonction AxJerk/AJK modifie les valeurs de secousse dans le même sens en modes d'avance et d'avance rapide. Dans ce cas, le comportement est compatible avec MTX V11 et les versions antérieures.
 - Les valeurs limites de secousse agissent uniquement lorsque la limitation des secousses est activée (JerkControl /JKC).
 - Si un axe physique et un axe logique du canal actif portent le même nom au sein du système, la secousse influencée est toujours celle de l'axe logique.

6.17 Mise hors service du mode axe C pour les broches "AxisToSpindle, ATS"

Description : Commute une broche du mode axe C (voir [chap. 6.125 "Activation du mode axe C pour les broches SpindleToAxis, STA" à la page 411](#)) au mode broche.



Pour obtenir des informations détaillées sur la fonctionnalité "Transfert d'axe", voir la documentation "Description des fonctions IndraMotion MTX".

Syntaxe :

AxisToSpindle (<NPA> <IPA>{, <NPA> <IPA>}...)	
Format abrégé : ATS (. .)	
avec	
<NPA>	Nom physique de l'axe Détermine l'axe qui doit être commuté du mode d'axe C au mode broche.
<IPA>	Index physique de l'axe Effet similaire à <NPA>

Tab. 6-32: Syntaxe AxisToSpindle

Exemple :

```
:
N030 ATS (CH)    L'axe physique CH (c.-à-d. la broche ayant le nom CH en mode
:                axe) est commutée au mode broche.
```

- Particularités et restrictions :**
- Un axe indiqué doit être à l'arrêt et ne doit appartenir à aucun groupe d'axes.
Si ce n'est pas le cas, la commande génère un message d'erreur et interrompt le programme.
 - La position des axes au sein d'un même bloc doit toujours être programmée suivant l'expression AxisToSpindle(...).

6.18 Mise hors service du mode axe C pour les broches et attente "AxisToSpindleWait, ATSW"

Description : Commute une broche du mode axe C au mode broche et attend en mode broche jusqu'à ce que la broche soit active (voir [chap. 6.125 "Activation du mode axe C pour les broches SpindleToAxis, STA" à la page 411](#) et [chap.](#)

Fonctions CN avec syntaxe de langage standard

6.126 "Activer le mode axe C pour des broches et attendre SpindleToAxisWait, STAW" à la page 411).



Pour obtenir des informations détaillées sur la fonctionnalité "Transfert d'axe", voir la documentation "Description des fonctions IndraMotion MTX".

Syntaxe :

AxisToSpindleWait (<NPA> <IPA>{, <NPA> <IPA>}...)	
Format abrégé : ATSW (. .)	
avec	
<NPA>	Nom physique de l'axe Détermine l'axe qui doit être commuté du mode d'axe C au mode broche.
<IPA>	Index physique de l'axe Effet similaire à <NPA>

Tab. 6-33: Syntaxe AxisToSpindle

Exemple :

```

:
N030 ATSW (CH)    L'axe physique CH (c.-à-d. la broche ayant le nom CH en mode
                  axe) est commutée au mode broche.
N040 M3 S1000     La broche est démarrée.
:

```

Particularités et restrictions :

- Un axe indiqué ne doit appartenir à aucun groupe d'axes.
Si ce n'est pas le cas, la commande génère un message d'erreur et interrompt le programme.
- La position des axes au sein d'un même bloc doit toujours être programmée selon l'expression AxisToSpindleWait(...).

6.19 Remplacement d'axes "AxisReplace, AXR "

Description : Placée après l'avance du bloc au sein du "programme de comparaison cible", permet d'entrer dans le programme pièce avec une configuration modifiée par rapport à la configuration d'axe calculée en remplaçant des axes par des axes ayant une signification similaire.

Syntaxe :

AXR(<NAC1>, <NPA2>)	L'axe dont le nom d'axe du canal est <NAC1> est transmis depuis le canal ; l'axe dont le nom physique de l'axe est <NPA2> est intégré au canal, puis le nom d'axe du canal <NAC1> lui est attribué.

Tab. 6-34: Syntaxe AxisToSpindle

Exemple : **AXR(A,A2)**

L'axe dont le nom d'axe du canal est A est transmis depuis le canal ; l'axe dont le nom physique de l'axe est A2 est intégré au canal, puis le nom d'axe du canal A lui est attribué.

Particularités et restrictions :

- Seuls des axes de même type peuvent être remplacés les uns les autres.
- Les axes ayant une signification à des fins d'usinage ne peuvent pas être remplacés.
- Un nom d'axe du canal doit être disponible et programmé pour l'axe à remplacer.

Fonctions CN avec syntaxe de langage standard

- La fonction Remplacer les axes peut uniquement être programmée dans le "programme de comparaison cible" et un programme de comparaison Init doit également être configuré.
- La configuration d'axe doit être comparée avant l'appel de AXR.
- Un seul ordre AXR peut être programmé par ligne CN.
- Après un remplacement d'axe, il n'est plus possible de programmer un ordre WAX, RAX, GAX, RLN ou ALN.
- Il est possible de remplacer max. 4 axes au total.

6.20 Modification de la vitesse maximale des axes "AxVel, AVE", enregistrement temporaire de la vitesse maximale des axes "AxVelSave, AVS"

Description : **AxVel :**

Cette fonction modifie temporairement les limites supérieures des vitesses maximales de l'axe. Pour cela, la fonction masque les valeurs de vitesse maximales de l'axe issues des paramètres machine avec les valeurs programmées.

AxVelSave :

Enregistre temporairement les vitesses maximales actuelles de tous les axes dans une mémoire interne. Cette mémoire interne est toujours préinitialisée avec les valeurs des paramètres machine lors de la sélection du programme.

Syntaxe :

AxVelSave	Enregistre les valeurs de vitesse maximales actuellement actives.
Format abrégé : AVS	
AxVel (<valeurs>)	Modifier la vitesse maximale de l'axe
AxVel () ou AxVel (0)	
	Réactiver la vitesse maximale de l'axe issue des paramètres machine.
AxVel (1)	Activer la vitesse maximale de l'axe enregistrée par AxVelSave.
AxVelSave	Enregistre les valeurs de vitesse maximales actuellement actives.
Format abrégé :AVE (. .)	
avec	
<valeurs>	Nom de l'axe et valeur de vitesse

Tab. 6-35: Syntaxe AxVel, AxVelSave

Les indications pour plusieurs axes sont séparées par une virgule.

En fonction de l'unité de mesure active (G71/G70), la commande interprète les données programmées pour les axes linéaires en "pouces/min" ou en "mm/min".

Les axes ronds sont toujours pré réglés en "1000 degrés/min".



Les expressions entre parenthèses "(0)" et "(1)" ne sont pas des paramètres de la fonction CN. La valeur numérique entre parenthèses ne peut donc pas être transmise par des variables CPL.

Fonctions CN avec syntaxe de langage standard

Exemple : Situation de départ :

Dans les paramètres machine, la valeur de 50 000 mm/s est prédéfinie pour les axes X et Z.

:

N40

AVE (X10000, Z30000)

:

N80 AVS

:

N90 AVE ()

:

N200 AVE

:

Vitesse max. pour l'axe X : 10 000,0 mm/s

Vitesse max. pour l'axe Z : 30 000,0 mm/min

La vitesse max. de l'axe Y reste inchangée (50 000,0 mm/min).

Enregistrer temporairement toutes les valeurs de vitesse maximales des axes actuellement actives.

Réactiver les valeurs issues des paramètres machine.

Réactiver les valeurs enregistrées à l'aide de "AVS/AxVel-Save" :

Vitesse max. pour l'axe X : 10 000,0 mm/s

Vitesse max. pour l'axe Z : 30 000,0 mm/min

Vitesse max. pour l'axe Y : 50 000,0 mm/min

6.21

Paramètres de l'approximation spline B "BsaPar, BAP"

Description : Définit les paramètres de l'approximation spline B.

Syntaxe :

BsaPar (E... , OE... , CMP...)	
E	Rayon de la bande de tolérance Valeur d'activation : Paramètres SPL/BSplineApproxTol (MP 8007 00030)
OE	Valeur de tolérance du vecteur d'orientation Valeur d'activation : OE = E
CMP	1 : Avec compression du bloc CN 0 : Sans compression du bloc CN Valeur d'activation : CMP1
BsaPar (0) - ou - BsaPar ()	Réinitialiser les paramètres de l'approximation spline B aux valeurs d'activation
Format abrégé : BAP	

Tab. 6-36: Syntaxe BAP

Exemple : Définition des paramètres de l'approximation spline B

SPD(4203,x,y,z,O)

BAP(E0.02,OE0.1,CMP0)

; Définit le type de spline qui sera ensuite activé avec G6

; Définit les paramètres de l'approximation spline B :

; Tolérance de 0,02 mm

; Tolérance d'orientation de 0,1 mm

; Compression du bloc CN désactivée

Fonctions CN avec syntaxe de langage standard

```
G6 x.. y.. z.. phi.. theta.. ; Approximation spline B (spline 4203) activée
...
BAP() ; Retourner aux valeurs d'activation :
; Tolérance du paramètre SPL/BSplineApproxTol
(MP 8007 00030)
; Compression du bloc CN activée
...
G1 x.. y.. z.. phi.. theta.. ; Approximation spline B désactivée
```

6.22 Placement : correction de la position de la pièce à usiner "BcsCorr, BCR"

Description : Sert à corriger le plan sous-tendu.

Le placement "Correction de la position de la pièce à usiner" peut décaler **et** orienter le système de coordonnées de la pièce à usiner dans l'espace. La correction de la position de la pièce à usiner agit sur les coordonnées ayant les significations "X", "Y" et "Z" sur le canal correspondant.



Les désignations d'axes "X", "Y" et "Z" utilisées ci-dessous se rapportent aux axes ayant les **significations** X, Y et Z.

Les efforts de configuration sont considérablement réduits, si la position de la pièce à usiner est mesurée après être sous-tendue et corrigée au moyen de la correction de la position de la pièce à usiner.

Comme il existe 3 degrés de liberté pour l'orientation, chaque orientation peut être représentée par 3 rotations de base successives. Pour des raisons de simplification, seule la rotation de base autour de la coordonnée Z est représentée dans la figure suivante :

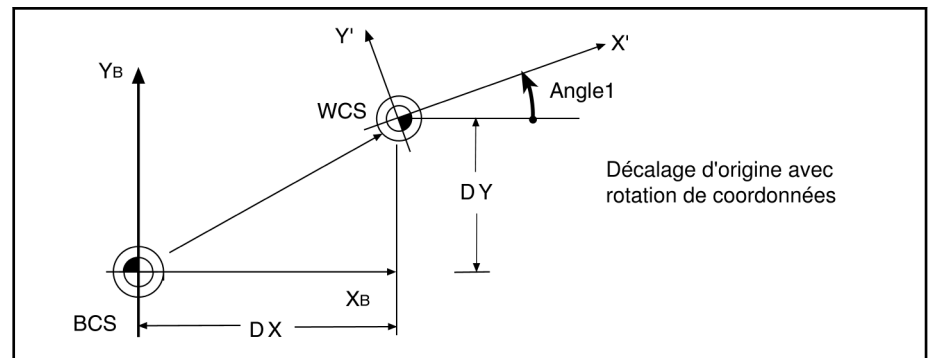


Fig. 6-4: Rotation de base autour de la coordonnée Z



Fonctionne de façon identique au placement "Plan incliné" (voir [chap. 5.2.43 "Placement : plan incliné G151 à G159.5" à la page 196](#)). Vous y trouverez également une représentation des 3 rotations de base.

Les autres placements (par ex. Plan incliné) ont un effet additif. Dans la "chaîne de calcul", la correction de la position de la pièce à usiner se trouve devant le plan incliné :

Fonctions CN avec syntaxe de langage standard

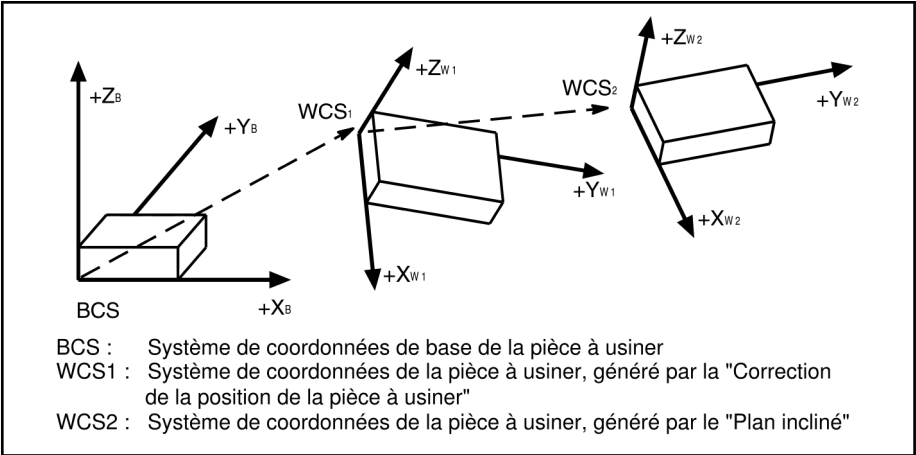


Fig. 6-5: Chaîne de calcul

Syntaxe :

BcsCorr ({<décalage X _W >},{<décalage Y _W >},{<décalage Z _W >},{<angle1>},{<angle2>},{<angle3>}})	
	Correction de la position de la pièce à usiner ACTIVÉE
BcsCorr () ou BcsCorr (0)	Correction de la position de la pièce à usiner DÉSACTIVÉE
Format abrégé : BCR (. .)	
avec	
<décalage X _W >	Valeur de décalage dans la direction de coordonnée principale X
<décalage Y _W >	Valeur de décalage dans la direction de coordonnée secondaire Y
<décalage Z _W >	Valeur de décalage dans la direction de coordonnée normale Z
<angle1>	Angle de rotation autour de la coordonnée Z Plage de valeurs : 0 ≤ <angle1> < 360 degrés
<angle2>	Angle de rotation autour de la coordonnée Y' Plage de valeurs : 0 ≤ <angle2> < 180 degrés
<angle3>	Angle de rotation autour de la coordonnée Z" Plage de valeurs : 0 ≤ <angle3> < 360 degrés

Tab. 6-37: Syntaxe BcsCorr (BCR)

Exemple :

N70 G40	Correction de trajectoire de la fraise DÉSACTIVÉE
N80	Correction de la position de la pièce à usiner ACTIVÉE
BCR(50,300,10,1.23)	L'origine du nouveau système de coordonnées de la pièce à usiner se trouve dans le BCS sur X50, Y300 et Z10. Les axes des coordonnées X et Y du nouveau système de coordonnées de la pièce à usiner sont tournés de 1,23 degré dans le sens trigonométrique autour du nouvel axe de coordonnée Z par rapport au BCS.
:	
:	
N200 BCR()	Correction de la position de la pièce à usiner DÉSACTIVÉE

Fonctions CN avec syntaxe de langage standard

- Particularités et restrictions :**
- L'activation et la désactivation de la correction de la position de la pièce à usiner interrompt la prévisualisation des blocs. Par conséquent, ces actions ne doivent pas être programmées lorsque la correction de trajectoire de la fraise est active (G41/G42, voir [chap. 5.2.15 "Correction de trajectoire de la fraise G40, G41, G42"](#) à la page 155).
 - L'état de mise en service et le comportement en cas de remise à zéro sont configurés dans les paramètres machine 7060 00010 et 7060 00020.

6.23 Surveillance du bloc suivant en cas de définition du bloc CN en mémoire tampon "ChkNxtBlk, CNB"

Description : La fonction ChkNxtBlk peut être utilisée pour surveiller le flux de données en cas de définition du bloc CN en mémoire tampon.

Le système vérifie si la définition du bloc CN enregistrée dans la mémoire tampon peut mettre à disposition le bloc suivant à temps. Si cela n'est pas le cas, une erreur d'exécution est générée.



Vous trouverez des informations à propos de l'interface OPC dans la documentation "Communication OPC IndraMotion MTX".

Syntaxe :

ChkNxtBlk (1)	Active la surveillance du bloc suivant.
ChkNxtBlk (0) ou ChkNxtBlk ()	Désactive la surveillance du bloc suivant.
Format abrégé : CNB (. .)	

Tab. 6-38: Syntaxe ChkNxtBlk (CNB)

- Particularités et restrictions :** La fonction est désactivée par une remise à zéro.

6.24 Programmation de chanfreins "ChLength", "CHL", "CHLL", "ChSection", "CHS", "CHSL"

Description : La fonction "Programmation de chanfreins" ajoute une phase de raccordement entre deux blocs CN successifs de type droite ou cercle, dont la longueur peut être indiquée soit en **longueur de chanfrein** absolue, soit en **longueur de segment de chanfrein**. Le chanfrein est généré au sein du plan de travail actif.

La fonction Programmation de chanfreins est disponible de manière modale (ChLength, CHL, ChSection, CHS) et par bloc (CHLL, CHSL). En cas de programmation par bloc, il est possible en option de définir une avance locale avec laquelle le chanfrein doit être usiné.

Les raccords de chanfrein suivants sont possibles :

- **Chanfrein entre deux droite sécantes**

Le chanfrein suit un tracé à angle droit avec la bissectrice entre deux segments de trajectoire consécutifs. La longueur du chanfrein est automatiquement corrigée (diminuée), s'il n'y a pas de point d'intersection avec les segments de trajectoire consécutifs programmés.

Fonctions CN avec syntaxe de langage standard

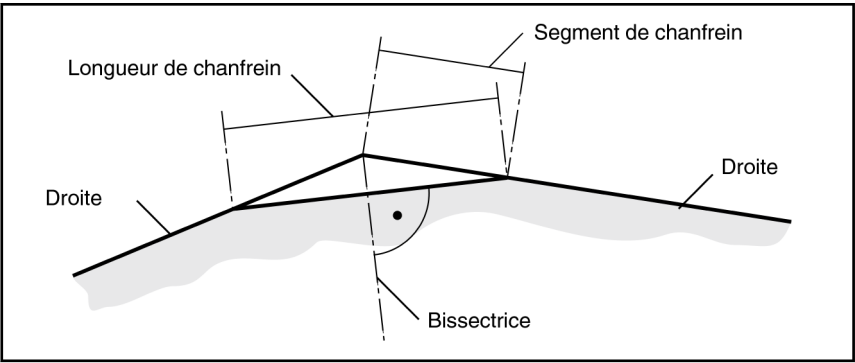


Fig. 6-6: *Chanfrein entre deux droites sécantes*

- **Chanfrein entre deux segments de cercles sécants**

En cas de raccords entre segments de contour avec des segments de cercles, les dimensions des chanfreins se rapportent aux tangentes finale et initiale respectives des segments de trajectoire participant au raccord entre segments de contour. La longueur de chanfrein effective est, entre autres, fortement dépendante des rayons des cercles participants et diffère donc plus ou moins des dimensions programmées.

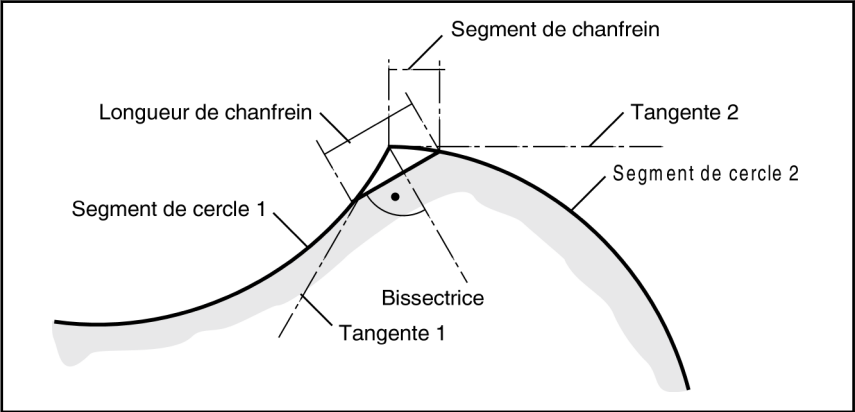


Fig. 6-7: *Chanfrein entre deux segments de cercles sécants*

Syntaxe :

ChLength (<longueur de chanfrein>)	Programmation de chanfreins ACTIVÉE Indiquer la <longueur de chanfrein> voulue en mm (G71) ou en pouces (G70).
ChLength() ou ChLength(0)	Programmation de chanfreins DÉSACTIVÉE
Format abrégé : CHL (..)	
ChSection (<segment de chanfrein>)	
	Programmation de chanfreins ACTIVÉE Indiquer le <segment de chanfrein> voulu en mm (G71) ou en pouces (G70).
ChSection() ou ChSection(0)	Programmation de chanfreins DÉSACTIVÉE
Format abrégé : CHS (..)	

Tab. 6-39: *Syntaxe ChLength (CHL) ou ChSection (CHS)*

Fonctions CN avec syntaxe de langage standard

CHLL (<longueur> {, <LocFeed> })	Programmation de chanfreins par bloc avec indication de la longueur de chanfrein
CHSL (<section> {, <LocFeed> })	Programmation de chanfreins par bloc avec indication du segment de chanfrein
avec	
<longueur>	Indiquer la <longueur de chanfrein> voulue en mm (G71) ou en pouces (G70).
<LocFeed>	Avance locale avec laquelle le chanfrein de raccord doit être déplacé.

Tab. 6-40: Syntaxe CHLL et CHSL

- Particularités et restrictions :**
- Les fonctions "ChLength", "ChSection", "RoundEps" et "Rounding" **ont un effet modal** et se révoquent mutuellement.
 - Les fonctions "CHLL" et "CHSL" **ont un effet local** (par bloc).
 - Si un chanfrein ou un arrondi agissent de manière modale et qu'un chanfrein local est programmé simultanément, alors la fonction locale écrase la fonction modale dans le bloc.
 - Le chanfrein se rapporte toujours uniquement au plan de travail actif (G17, G18, G19, G20). Le fait que d'autres axes participent éventuellement au mouvement n'a aucun effet sur les chanfreins.
 Étant donné que les coordonnées des blocs de déplacement des axes sont modifiées par les chanfreins au sein du plan de travail mais que les valeurs pour les axes en dehors du plan de travail restent inchangées, la direction dans l'espace peut changer, par exemple pour les droites.
 - La fonction n'agit que dans la zone de commande "Exécution", sous "Bloc suivant", "Bloc unique" ou "Étape unique".
 Étant donné que le "bloc-programme" se comporte comme une saisie manuelle, la programmation de chanfreins est ici sans effet.
 - Le comportement de mise en/hors service ainsi que le comportement pendant la remise à zéro sont déterminés par les chaînes d'initialisation des paramètres machine 7060 00010 et 7060 00020.
 - Si le plan, le décalage d'origine ou une transformation d'axe est modifié entre 2 blocs successifs, aucun chanfrein n'est généré.

6.25 Détection automatique des coins et des droites "CLD"

Description : La fonction "Détection automatique des coins et des droites" génère des coins et des droites au sein des séquences spline en fonction de paramètres programmés. Si un angle de coude est supérieur à l'angle limite programmé, un coude reste programmé, c'est-à-dire qu'il n'est pas arrondi par une courbe spline. Dans les blocs dont l'écart des points est supérieur à l'écart limite programmé, une spline linéaire (droite) est générée. Il est possible de définir également un facteur d'angle ou d'écartement (type de spline 4 uniquement). De même, un coude ou une droite est créé si un angle ou un écart dépasse la valeur moyenne respective indiquée dans la zone de prévisualisation. Par ailleurs, une tolérance d'angle peut être indiquée. Si une spline d'arrondi imaginaire est insérée avec cette tolérance dans un enchaînement de blocs de sorte qu'au moins l'un des deux côtés reste à plus de la moitié, la condition d'écartement sera remplie pour ce dernier. La fonction est uniquement active quand G6 (interpolation spline) est actif. Elle est active pour tous les types de splines à l'exception de la programmation spline de coefficients.

Fonctions CN avec syntaxe de langage standard

Syntaxe : La détection automatique des coins et des droites est mise en service à l'aide de la syntaxe **CornerLineDetection**.

CornerLineDetection(ANG<a>,AFACT<fa>,DIST<d>,DFACT<fd>)	
Les paramètres ont la signification suivante :	
ANG<a>	"a" est l'angle de coude maximal admissible pour le contour. La condition d'angle est remplie avec des angles de coude supérieurs à a.
AFACT<fa>	Uniquement pour le type de spline 4 : "fa" est le facteur d'angle. La condition d'angle est remplie si l'angle de coude dépasse l'angle de coude moyen du facteur "fa".
DIST<d>	"d" est l'écart maximal admissible entre les points. La condition d'écartement est remplie pour les écarts supérieurs à d.
DFACT<fd>	Uniquement pour le type de spline 4 : "fd" est le facteur d'écartement. La condition d'écartement est remplie si l'écart entre les points dépasse l'écart moyen du facteur "fd".
E<e>	"e" est une valeur de précision du contour. Si une spline d'arrondi avec cette tolérance est créée sur un angle de sorte qu'au moins l'un des deux côtés reste à plus de la moitié, la condition d'écartement est remplie pour ce dernier.

Tab. 6-41: Syntaxe CornerLineDetection

Chaque paramètre individuel est optionnel. Les conditions DIST, DFACT et E, ainsi que ANG et AFACT sont liées par un "OU" logique. Après avoir programmé CLD avec au moins un paramètre, la fonction est mise en service avec un effet modal et devient active dès que G6 est activé (exception : programmation spline des coefficients). Les paramètres peuvent à tout moment être modifiés par un nouveau paramétrage.

Le format abrégé est CLD(...).

La fonction est désactivée par une programmation sans le paramètre CornerLineDetection().

Exemple : **Spline cubique constante c2**

```
SPD(2203,x,y,z,O) CLD(ANG20,AFACT5,DIST5)
```

```
...
```

```
G6 x.. y.. z.. phi.. theta.. ; Spline 2203 ACTIVÉE, condition d'écartement absolue,  
                                ; conditions d'angle absolue et relative actives
```

```
x.. y.. z.. O(..)
```

```
...
```

```
x.. y.. z.. O() CLD(DIST8) ; Spline DÉACTIVÉE
```

```
G1 ; Désactivation des conditions d'angle,  
    ; nouvelle condition d'écartement absolue
```

Fonctions CN avec syntaxe de langage standard

...

G6

; Spline ACTIVÉE, le dernier CLD(DIST8) étant valable

6.26 Surveillance de collision "Collision, CLN"

Description : *Offre la possibilité :*

- de mettre en/hors service la surveillance de collision de la correction de trajectoire de la fraise G41/G42 ;
- de l'adapter à l'application en fonction de la zone de prévisualisation et du comportement en cas de collision.

Les points suivants s'appliquent ici :

- La surveillance de collision est seulement effective si la correction de trajectoire de la fraise G41/G42 est active, même si le rayon de correction a la valeur "0".
- Si la valeur de la correction du rayon ne permet pas de traiter des segments individuels du contour, la commande tente alors de modifier le tracé de la trajectoire correspondante de façon à ne pas endommager le contour.
- La surveillance de collision ne prend en considération que les coordonnées du plan de travail actif lors du tracé de contour. Si la programmation technique empêche les collisions, par exemple en modifiant la profondeur du pas de pénétration de l'outil, cela entraîne tout de même un adressage de la surveillance de collision au sein de la zone de prévisualisation actuelle.

Dans ce cas, il est possible de désactiver temporairement la surveillance de collision dans le secteur de traitement concerné.



Vous trouverez des informations détaillées à propos de la fonctionnalité "Surveillance de collision" dans la documentation "Description des fonctions IndraMotion MTX".



Nous recommandons d'inscrire le comportement voulu de la surveillance de collision dans MP 7060 00010 / MP 7060 00020.

Syntaxe :

Collision(1) ou CLN(1) ou CLN	Surveillance de collision ACTIVÉE Conserver le comportement actuel en cas de collision. Si ce comportement n'a pas encore été programmé ou s'il a été inscrit dans MP 7060 00010 / MP 7060 00020, le comportement correspond à celui de Collision(CollErr 0). CLN ("CLN sans parenthèses") correspond à CLN(1) ou à Collision(1).
Collision() ou Collision(0)	Surveillance de collision DÉACTIVÉE
Collision(DEF)	Prédéfinir une valeur pour la zone de prévisualisation de la surveillance de collision à 2 blocs.
Collision(CollErr<type>)	

Fonctions CN avec syntaxe de langage standard

	Activer la surveillance de collision et définir le comportement en cas de collision.
Collision (DLA<blocs>)	Prédéfinir la valeur pour la zone de prévisualisation de la surveillance de collision. Prend effet avec la programmation suivante de G41/G42.
Collision (LA<blocs>)	Modifier temporairement la zone de prévisualisation jusqu'à la programmation suivante de G41/G42.
Format abrégé : CLN (. . .)	
avec	
<type>	Définit le comportement de la commande en cas de détection d'une collision : 0 : Ne générer ni erreurs d'exécution, ni avertissements. 1 : Générer une erreur d'exécution Le traitement est interrompu. 2 : Émettre un avertissement Le traitement n'est pas interrompu.
<blocs>	Détermine la taille (nombre de blocs) de la zone de prévisualisation. Valeur entière. Zone de prévisualisation recommandée : 1 à 10 blocs.

Tab. 6-42: Syntaxe Collision (CLN)

Exemple :

```

:
N100 CLN (DLA 5)      Préréglage pour la zone de prévisualisation de la surveillance
:                      de collision à partir de la programmation suivante de G41/
                        G42 : 5 blocs.
N110 G41 D10          Correction de trajectoire de la fraise à gauche de la pièce à
                        usiner
N120 X10              Déplacement en avant
N130 X20
N140 X30
N150 G42              Commuter à la correction de la trajectoire de la fraise à droite
:                      de la pièce à usiner.
                        La prévisualisation de la surveillance de collision est terminée
                        à partir du bloc 150 et redémarrée ensuite.
N160 X20              Déplacer en arrière
N170 X10

```

Fonctions CN avec syntaxe de langage standard

```
N180 XO
N190 G40      Correction de trajectoire de la fraise DÉSACTIVÉE
:            La surveillance de collision devient ainsi inactive, mais elle
              n'est pas désactivée !
```

- Particularités et restrictions :**
- Le pré réglage pour la zone de prévisualisation s'élève à 2 blocs.
 - La zone de prévisualisation maximale dépend des paramètres machine 7060 00110 à 7060 00130.
 - Pour pouvoir effectuer un recul sans que la surveillance de collision ne réagisse lorsque la correction de trajectoire de la fraise est activée, la direction de correction active doit être inversée à partir du point d'inversion (programmer G42 pour G41 actif et G41 pour G42 actif).
- La commande termine automatiquement la zone de prévisualisation de la surveillance de collision dans un bloc G41 ou G42 et redémarre ensuite la prévisualisation.

6.27 Configuration de l'interpolation G0 "ConfG0, CG0"

Description : L'interpolation G0 est configurée de façon asynchrone. Les axes participants restent synchrones, mais le mouvement s'effectue bloc par bloc de manière asynchrone. Cela signifie qu'ils se déplacent vers la position finale programmée avec leurs valeurs limites dynamiques respectives. Les axes atteignent leur position finale à différents moments.

Le bloc CN est encore activé lorsque tous les axes ont atteint leur position finale programmée. En fonction de l'IPS ou NIPS actifs, l'atteinte de la position finale se rapporte à la position réelle ou de consigne.

La fonction comprend une fonctionnalité WAIT interne à la commande numérique.

Syntaxe :

ConfG0(ASY<statut>)	Le statut peut avoir la valeur "1" ou "0".
ConfG0(ASY1)	Activer l'interpolation G0 asynchrone
ConfG0(ASY0)	Désactiver l'interpolation G0 asynchrone
Format abrégé : CG0	

L'interpolation asynchrone est active pour tous les blocs G0 suivants jusqu'à la programmation de CG0(ASY0), c.-à-d. qu'elle reste active jusqu'à sa désactivation.

Exemple :

```

:            Activer l'interpolation G0 asynchrone
N10 CG0(ASY1) G0      Démarrer la position (100,100,50) de manière asyn-
N20 XA100 YA100 ZA50  chrone, attendre que l'axe le plus faible ait atteint sa
N30 XA200 YA300 ZA55  position
N40 CG0(ASY0)         Démarrer la position (200,300,55) de manière asyn-
N50 XA400 YA450 ZA60  chrone, attendre que l'axe le plus faible ait atteint sa
:            position
:            Désactiver l'interpolation G0 asynchrone
:            Approcher la position (400,450,60) de manière syn-
:            chrone
```

- Le mouvement est toujours effectué axe par axe. En cas de transformation active de l'axe aux points d'action 1 ou 2, la position finale des axes

Fonctions CN avec syntaxe de langage standard

correspondant aux coordonnées est calculée en interne et approchée de manière asynchrone, c.-à-d. qu'il n'y a aucun mouvement en coordonnées.

- La fonction génère une erreur d'exécution si un couplage d'axes système est actif.
- La fonction n'est pas compatible avec Retrace.

6.28 Sélection d'une transformation d'axe "Coord, CRD"

Description : Active ou désactive des transformations d'axes configurées dans le groupe de paramètres machine 1030.



La fonction est utilisée par exemple en lien avec la programmation des coordonnées tridimensionnelles (voir documentation "Description des fonctions IndraMotion MTX") ou, dans certaines circonstances, pour le calibrage de cinématiques d'axes (voir chap. 6.10.4 "Lecture des paramètres de la commande numérique ATGET" à la page 244, chap. 6.10.5 "Écriture de paramètres dans la commande numérique ATPUT" à la page 246 et chap. 6.10.3 "Transformation en avant ATFWD" à la page 243).

Dans la commande, une transformation d'axe doit être activée en 2 points d'action au maximum :

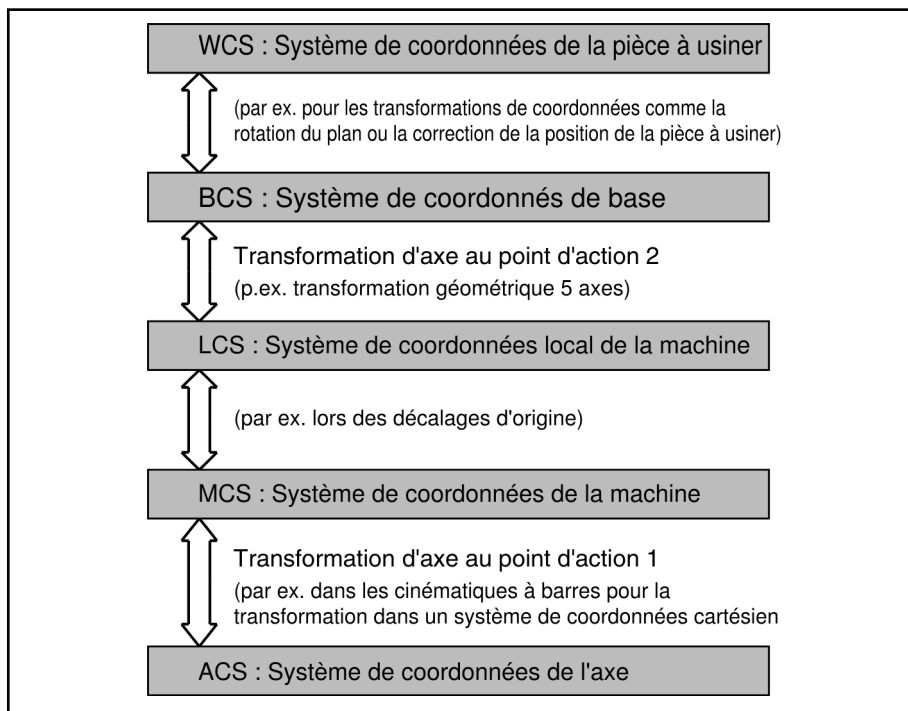


Fig. 6-8: Transformation d'axe



Pour chaque transformation d'axe disponible, il est déjà défini en interne sur quel point d'action cette transformation prendra effet.

Fonctions CN avec syntaxe de langage standard

Syntaxe :

Coord (<NoTransfoAxe>) ou Coord (<[SDTrfName]>)	Transformation d'axe ACTIVÉE
Coord () ou Coord (0)	Transformation d'axe au point d'action 2 DÉSACTIVÉE
Coord (0,<point d'action>)	Transformation d'axe au point d'action sélectionné DÉSACTIVÉE
Format abrégé :CRD (...)	
avec	
<NoTransfoAxe>	Numéro de la transformation d'axe Champ de saisie : 1 à 20. Nombre entier L'attribution entre NoTransfoAxe et le type de transformation d'axe est enregistrée dans MP 1030 00110.
<[SDTrfName]>	Nom de la donnée système (entre crochets) de type SysAxTrafo_t qui contient les données de transformation à activer. Le type de transformation d'axe est contenu directement dans la donnée système.
<point d'action>	Point d'action de la transformation d'axe à désactiver Saisie : 1 ou 2

Tab. 6-43: Syntaxe Coord (CRD)

Particularités et restrictions :

- Il est possible de commuter directement entre les différentes transformations d'axe au point d'action 2. Une désactivation préalable n'est pas nécessaire.
- Une transformation d'axe au point d'action 1 ne peut être commutée que si aucune transformation d'axe n'est active au point d'action 2.



Pour plus d'informations sur les types de transformations d'axes disponibles, voir la documentation "Description des fonctions IndraMotion MTX".

6.29 Pointes d'angles intérieurs "Cornering, COR"

Description : La fonction "Pointe d'angles intérieurs" permet de rendre des angles intérieurs pointus par fraisage avec des outils coniques.

Pour ébavurer et tailler des contours en forme d'âme avec une section triangulaire, on utilise une fraise conique avec un angle d'ouverture correspondant pour usiner les deux flancs de l'âme avec la correction du rayon d'outil "G41" ou "G42" active. Il en résulte une arête de coupe tranchante à la pointe de l'âme. Cependant, des résidus de matériau persistent dans les coins intérieurs (selon la géométrie, à cause du rayon de coupe actif à la pointe de l'âme), si bien que l'arête de coupe obtenue dans cette zone n'est pas tranchante.

Pour traiter les résidus de matériau, la fonction "Cornering, COR" conduit le centre de l'outil jusqu'à la pointe du coin selon une trajectoire calculée (en tenant compte du contour des angles antérieur et postérieur) et la ramène de même quand un angle intérieur est détecté. Ici, la pointe de l'outil est soulevée jusqu'à ce que l'outil soit en prise avec le contour d'angle antérieur comme avec le contour d'angle postérieur à la hauteur de la pointe de l'âme. Au

Fonctions CN avec syntaxe de langage standard

niveau du point angulaire, la pointe de l'outil se trouve (théoriquement) directement sur la pointe de l'âme.

Les contours d'angles antérieur et postérieur peuvent se composer chacun de plusieurs blocs CN, à savoir de blocs CN linéaires et/ou circulaires quelconques. Il est toutefois important que le nombre de blocs d'angles antérieurs et postérieurs pertinents soit compris dans la zone de prévisualisation paramétrée de la fonction "Surveillance de collision CLN". Par ailleurs, la surveillance de collision doit être activée. Il en va de même pour la fonction "Arc de cercle sur le raccord entre segments de contour G43".

Syntaxe :

Cornering([CAN<can>] [, MAN<man>] [, FF<ff>])	
avec	
<can>	Cone ANgle : Angle d'ouverture de l'arête coupante de l'outil, en degrés. Valeur par défaut : 90°
<man>	Min ANgle : Angle minimal entre deux blocs successifs dans l'enchaînement de blocs permettant de tailler un angle. Valeur par défaut : 10° Ce paramètre est uniquement pertinent pour les "angles simples". Un "angle simple" est un angle défini uniquement par les deux éléments de trajectoire limitrophes. En cas détection d'une collision par la surveillance de collision, l'angle minimal paramétré ne joue aucun rôle, puisqu'aucun angle géométrique ne doit se trouver obligatoirement au sein de la collision. Un amincissement de l'âme a toujours lieu quand une collision est détectée.
<ff>	Feed Factor : Facteur d'avance avec lequel l'avance programmée dans la zone de la pointe de l'angle est mise à l'échelle. Valeur par défaut : 1 Plage de valeurs : 0 < <ff> 1
Cornering(0)	Désactiver la pointe des angles intérieurs.

Tab. 6-44: Syntaxe Cornering

Exemple :

:	Activer l'arc de cercle au niveau du raccord entre segments de contour
N10 G43	Activer la surveillance de collision, définir la zone de surveillance sur 80 blocs
N20 CLN(DLA80)	
N30 CON(CAN87.8, MAN15, FF0.2)	Activer et paramétrer la fonction Pointe d'angle : angle d'ouverture 87,8°, angle minimal 15°, facteur d'avance 20 %
N40 G42 ED1	
:	Activer la correction du rayon d'outil, accepter les données d'outil provenant du bloc de correction ED1

Particularités et restrictions :

- La fonction Pointe d'angle a un effet modal.
- Une condition préalable à la fonction Pointe d'angle est que la "détection de collision CLN" doit être activée.
- La fonction Pointe d'angle fonctionne uniquement quand la fonction "Arc de cercle au niveau d'un raccord entre segments de contour G43" est active.

- Le rayon de correction de l'outil ne doit pas être modifié au sein de la fonction Pointe d'angle.

6.30 Table de couplage spline "CoupleSplineTab, CST"

Description : Lors du couplage de coordonnées ou d'axes, la fonction de couplage est enregistrée sous la forme de paires de repères dans une table de couplage. Pour calculer les positions de l'axe esclave entre les repères, la préparation des blocs génère une table spline.

La table spline est générée lors de l'interpolation de la syntaxe de couplage et enregistrée en tant que fichier dans le répertoire des tables de liaison.

Les tables splines sont générées automatiquement pour les couplages d'axes du canal (AXC, SCCT).

CoupleSplineTab(...) permet aussi de générer une table spline définie :

- **CST(TAB(<NomTab>, 1))**
Force explicitement la création d'une nouvelle table spline.
- **CST(TAB(<NomTab>, 0))**
Force la création d'une nouvelle table quand aucune table spline n'existe ou s'il en existe une qui est antérieure à la table de couplage.
- **CST(TAB(<NomTab>{,0|1}),LOAD)**
Charge également la table spline dans la mémoire des tables système du couplage d'axes système.
- **CST(TAB(<NomTab>{,0|1}),UNLOAD)**
Supprime la table spline de la mémoire des tables systèmes du couplage d'axes système.

CoupleSplineTab(...) permet également de générer une table spline sans qu'aucun groupe d'axes ne doivent exister (par ex. en cas de saisie manuelle).

Pour l'utilisation des tables dans le couplage d'axes système, voir aussi la documentation "Description des fonctions IndraMotion MTX", chapitre "Couplage d'axes système".

Le nom de la table spline est généré à partir du nom de la table de couplage active à ce moment, auquel on ajoute l'extension ".s". Par exemple, le nom de table de couplage **curve.fct** donnera le nom de table spline **curve.fct.s**.



La table de couplage est recherchée dans le chemin de recherche actuel. Le chemin de recherche est réglé dans le paramètre machine 3080 00001.

Le répertoire par défaut pour les tables de liaison est : /usr/lnk. Il peut être défini librement dans le paramètre machine 3080 00004.

Syntaxe :

CoupleSplineTab (TAB (<NomTab>{,<1 0>}) { , LOAD UNLOAD } { , <emplacement> })	
	Générer une table spline
Format abrégé : CST (...)	
avec	
<NomTab>	Nom de la table de couplage recherchée dans le chemin de recherche actuel et pour laquelle une table spline est générée.

Fonctions CN avec syntaxe de langage standard

<1 0>	En option : 0 : La table spline est uniquement générée si elle n'existe pas encore ou si elle est antérieure à la table de couplage. (par défaut) 1 : Une nouvelle table spline est générée.
LOAD UNLOAD	En option uniquement pour le couplage d'axes système : LOAD : La table spline est chargée dans la mémoire de tables intercanaux SysAxCoupleTab pour être utilisée dans le couplage d'axes système. Si aucun emplacement n'est indiqué, la saisie utilisée est celle dans "<NomTab>" ou, si cette dernière n'existe pas, la première saisie libre. UNLOAD : La table spline est supprimée de la mémoire des tables SysAxCoupleTab. Si aucun emplacement n'est indiqué, la saisie "<NomTab>" est recherchée et utilisée.
<emplacement>	En option uniquement pour le couplage d'axes système : Charge ou supprime la table de la mémoire des tables système SysAxCoupleTab[<emplacement>] Si un nom de table est indiqué lors d'une libération orientée emplacement (UNLOAD), il est ignoré. Le nombre d'emplacements n'est effectif qu'en combinaison avec LOAD / UNLOAD.

Tab. 6-45: Syntaxe CoupleSplineTab (CST)

Exemple :

CST (TAB (curve.fct))

Crée la table spline /<répertoire de liaison>/ curve.fct.s si nécessaire.

CST (TAB (curve.fct, 1))

Crée la table de liaison /<répertoire de liaison>/curve.fct.s indépendamment de la date ou de l'existence.

CST (TAB (curve.fct) , LOAD)

Crée la table spline si nécessaire et la charge dans la mémoire des tables système SysAxCoupleTab (donnée système).

CST (TAB (curve.fct) , UNLOAD)

Supprime la table de la mémoire des tables système.

CST (TAB (curve.fct) , LOAD, 5)

Charge la table curve.fct dans la mémoire des tables système SysAxCoupleTab[5]

CST (, UNLOAD, PLACE%)

Supprime la table de la mémoire des tables système SysAxCoupleTab[PLACE%]

6.31 Interpolation des paramètres des courbes "CurveParamInterpolation, CPI"

Description : Pour les éléments de contour avec des paramètres de courbes, à savoir les types spline 0 à 4 et les splines d'arrondi (syntaxe SCO), il est possible de sélectionner une interpolation des paramètres des courbes plutôt qu'une interpolation des longueurs de la trajectoire. Les coordonnées du canal ne suivent alors pas la trajectoire vers la course s de manière synchrone, mais de manière synchrone par rapport au paramètre de courbe w. Le paramètre des courbes w s'étend de $w = 0$ à $w = W_e$. Pour les splines, W_e est la longueur des paramètres de la spline. Elle est programmée avec PL ou calculée automatiquement en interne selon le paramétrage spline sélectionné.

Syntaxe :	CurveParamInterpolation(1)
	Activer l'interpolation des paramètres des courbes.
	CurveParamInterpolation(0)
	Repasser à l'interpolation des longueurs de la trajectoire.
	Format abrégé : CPI

Tab. 6-46: Syntaxe CoupleSplineTab (CST)

Exemple :

```

N10 SplineDef(2203)
N20 G1 F123 X0 Y0 ; L'avance est de 123 mm/min
N30 CPI(1) ; Activation de l'interpolation des paramètres des courbes
N40 G06 X10 Y10 PL5 ; La durée du bloc est de 5/123 min(*)
N50 X20 Y15 PL4 F88 ; La durée du bloc est de 4/88 min(*)
N60 G1 CPI(0) ; Repasser à l'interpolation des longueurs de la trajectoire

```

(*) À l'exception des opérations de freinage et d'accélération dans ce bloc

Particularités et restrictions : L'interpolation des longueurs de la trajectoire est activée après la montée en régime de la commande.

6.32 Activation des tables de correction D "DcTSel, DCS"

Description : Active une table de correction D (table de correction géométrique, table GEO). Les tables de correction D sont enregistrées dans le système de fichiers de la commande au format XML.

Syntaxe :	DcTSel ({<chemin>}<nom de fichier>)
	Format abrégé : DCS (. .)
	avec

Fonctions CN avec syntaxe de langage standard

<chemin>	Indication optionnelle du chemin d'accès au répertoire dans lequel <nom de fichier> est enregistré. Si aucun chemin n'est indiqué, la recherche s'effectue sous le chemin "/database". Si <nom de fichier> ne s'y trouve pas, la commande utilise le chemin de recherche des sous-programmes afin de rechercher <nom de fichier> dans d'autres répertoires.
<nom de fichier>	Nom du fichier de la table de correction D avec extension de fichier incluse. Les tables ayant un nom standard (DC <numéro>.dct) peuvent être activées directement par leur numéro, par ex. DcTSel(7) active la table DC7.dct.

Tab. 6-47: Syntaxe DcTSel (DCS)



Pour plus d'informations sur la création et l'édition des tables de correction D, reportez-vous au mode d'emploi de la commande !

Exemple :

```

:
N030 DCS (geotab.dct) Recherche la table de correction "geotab.dct" d'abord
:                        dans le répertoire "/database", puis dans le chemin de
                        recherche pour sous-programmes si la recherche a
                        été infructueuse. La première table de correction D
                        trouvée avec le nom "geotab.dct" est activée.

N130 DCS (/mnt/ge.dct) Recherche et active la table de correction D "ge.dct"
:                        dans le répertoire "/mnt". Si elle ne s'y trouve pas, un
                        message d'erreur est généré.

```

6.33 Acceptation des réglages d'axe issus des paramètres machine "DefAxis, DAX"

Description : Active la configuration par défaut de l'axe sur tous les canaux conformément à MP 1003 00002.



Pour obtenir des informations détaillées sur la fonctionnalité "Transfert d'axe", voir la documentation "Description des fonctions IndraMotion MTX".

Syntaxe :

DefAxis
Format abrégé : DAX

Tab. 6-48: Syntaxe DefAxis, DAX

Particularités et restrictions : Entraîne une erreur d'exécution, si un axe participant n'est pas libéré.



Étant donné que plusieurs canaux participent généralement au transfert d'axe, il est recommandé de saisir la fonction dans MP 7060 00020, après le mot-clé "#SysRes".

De cette manière, il est toujours possible de rétablir la configuration d'axe par défaut grâce à une "remise à zéro du système".

6.34 Réinitialisation des broches du canal aux paramètres machine "DefSpindle, DSP"

Description : Active le réglage par défaut des broches selon le paramètre machine MP 7020 00020.

Syntaxe : DefSpindle

Format abrégé : DSP

Particularités et restrictions : Si les fonctions auxiliaires en nombre entier spécifiques au canal CSP1 à CSP8 sont définies, l'API est informé via l'affectation actuelle de la broche du canal.

6.35 Enchaînement de blocs sans réduction de la vitesse "DefTangTrans, DTT"

Description : La fonction "Enchaînement de blocs sans réduction de la vitesse" limite l'influence de la capacité de saut de l'axe sur l'angle de raccordement le plus élevé. Un paramétrage correspondant permet de régler un comportement avec lequel le saut d'axe n'est pris en compte que pour les coudes de contour importants, tandis que les coudes de contour plus petits sont considérés comme quasi-continus et la vitesse sur trajectoire n'est pas réduite. Sur une machine réelle, ceci risque d'entraîner l'adoucissement du contour programmé.

Syntaxe :

DefTangTrans (<angle>)	
Format abrégé : DTT (<angle>)	
avec	
<angle>	Si la modification de la partie d'axe sur le vecteur de direction normalisé est supérieure à $\sin(\text{angle})$ au niveau du raccord entre segments de contour, alors la capacité de saut de l'axe est prise en compte pour le raccord entre segments de contour. Si la modification est inférieure à $\sin(\text{angle})$, la capacité de saut de l'axe est ignorée et la vitesse n'est pas réduite au raccord entre segments de contour (à condition que la distance de freinage disponible dans le cadre de la prévisualisation des blocs réglée suffise). Plage de valeurs : 0 à 50 degrés
DefTangTrans({0})	L'angle limite saisi dans le paramètre machine 7030 00310 devient actif.

Tab. 6-49: Syntaxe DefTangTrans (DTT)

Particularités et restrictions :

- Si un arrêt précis est actif, la vitesse est ralentie à $v = 0$ à chaque enchaînement de blocs.
- Le réglage issu de la chaîne de mise en service prend effet après une remise à zéro.
- Si la fonction "DefTangTrans" est utilisée, il convient de toujours veiller à la vitesse avec laquelle le déplacement doit se faire. Comme celle-ci peut être ignorée temporairement dans les limites configurées dans les paramètres machine, des erreurs d'asservissement peuvent se produire, notamment à des vitesses élevées et avec de grands angles de raccordement !

6.36 Programmation du diamètre "DiaProg, DIA", programmation du rayon "RadProg, RAD"

Description : Les indications des coordonnées pour les axes du plan (sur des tours ; généralement la coordonnée X) peuvent être interprétées au choix en tant que diamètre ou rayon. Il est ainsi possible de reprendre les cotes existantes correspondantes directement dans le programme pièce, sans les convertir.

Si la "Programmation du diamètre" est activée, le caractère du diamètre est placé en tête de la position de la pièce à usiner, la course restante, la position finale et la valeur du programme dans les indications d'axe de l'axe du plan.

La position de la machine, la valeur réelle de l'axe et la poursuite sont toujours affichées en tant que valeurs de rayon.



Une mauvaise interprétation des cotes est possible ! "DIA" n'agit que sur les coordonnées de diamètre indiquées/configurées. La programmation du diamètre n'a aucune influence sur les paramètres de l'interpolation circulaire I, J, K.

Il convient de toujours s'assurer que seules les cotes appropriées sont programmées.

Syntaxe :

DIA { ({ <coord1> { , . . . , <coord8> } }) }	
	Programmation du diamètre pour 8 axes ou coordonnées linéaires au maximum ACTIVÉE. La programmation du diamètre est désactivée pour toutes les coordonnées non indiquées.
avec	
<coordi>	Max. 8 axes linéaires ou coordonnées (i = 1 à 8) dont les mesures de course doivent être évaluées en tant qu'indications du diamètre.
DIA	Rétablit le dernier état avant RAD. Après la montée en régime, les valeurs appliquées sont celles par défaut des paramètres machine.
DIA ()	Les valeurs par défaut sont activées. Si aucune coordonnée de diamètre n'est définie dans les paramètres machine, l'axe de la classification X devient l'axe de diamètre. Si celui-ci n'existe pas non plus, la commande signale une erreur d'exécution.
RAD	Programmation du rayon pour toutes les coordonnées ACTIVÉE

Tab. 6-50: Syntaxe DiaProg (DIA) et RadProg (RAD)

Exemple :

N10 DIA ()	Activer les coordonnées de diamètre configurées
N20 DIA (Y1, W2)	Les coordonnées Y1 et W1 indiquées deviennent des coordonnées de diamètre ; toutes les coordonnées non programmées deviennent des coordonnées de rayon.
N30 RAD	La programmation du diamètre est arrêtée pour toutes les coordonnées.
N40 DIA :	Les coordonnées Y1 et W1 deviennent des coordonnées de diamètre (dernier état avant RAD).

Particularités et restrictions :

- Ces fonctions sont modales et se révoquent réciproquement.

Fonctions CN avec syntaxe de langage standard

- Si une coordonnée de diamètre est intégrée à une transformation d'axe, la commande numérique arrête la programmation du diamètre pour cet axe. Lorsqu'une transformation d'axe est activée, il est également possible de commuter des coordonnées dans la programmation du diamètre.
- Les points suivants s'appliquent aux modes de fonctionnement "Saisie manuelle" et "Exécution" :
Si des cotes sont indiquées pour les coordonnées du centre du cercle, les longueurs d'outil et les décalages d'origine, alors les coordonnées de l'axe du plan sont toujours interprétées en tant que valeurs de rayon.
- Les points suivants s'appliquent aux modes de fonctionnement "Manivelle" et "JOG" :
Il est possible de passer de la programmation du diamètre à la programmation du rayon, et inversement, à l'aide du signal d'interface d'axe "Longueur de pas – diamètre" (qAx_JogDia).

6.37 Réaction temporisée en cas d'erreurs d'entraînement "DERDelay"

Description Dans certaines applications, la commande numérique doit conserver brièvement le contrôle sur les entraînements, même en cas d'apparition d'erreurs d'entraînement, et arrêter les axes machine de manière coordonnée afin d'éviter tout endommagement de la machine.

La fonction "DERDelay" peut mettre en/hors service la réaction temporisée aux erreurs pour les entraînements identifiés comme "programmables" dans "MAIN/Dr[i]/DelayedErrorReaction" (MP 1001 00012).



Pour de plus amples informations à ce sujet, voir le chapitre "Réaction temporisée en cas d'erreurs d'entraînement non fatales" dans la documentation "Description des fonctions IndraMotion MTX".

Syntaxe

DERDelay (1)	Active la réaction temporisée en cas d'erreurs pour tous les entraînements identifiés comme programmables dans "MAIN/Dr[i]/DelayedErrorReaction" (MP 1001 00012).
DERDelay () - ou - DERDelay (0)	Désactive la réaction temporisée en cas d'erreurs pour tous les entraînements identifiés comme programmables dans "MAIN/Dr[i]/DelayedErrorReaction" (MP 1001 00012).
DERDelay ({ON (<liste des entraînements>) }, {OFF (<liste des entraînements>) })	
	Active ou désactive la réaction aux erreurs retardée pour les entraînements programmés.
avec	
<liste des entraînements>	Liste des numéros des entraînements système ou des noms d'axes système séparés par des virgules.

Tab. 6-51: Syntaxe DERDelay

Exemple

```

N10 DERDelay(ON(X,B)) ; Activer les axes X et B
...
N200 DERDelay(OFF(X,B)) ; Désactiver les axes X et B

```

Fonctions CN avec syntaxe de langage standard

- Particularités et restrictions**
- La fonction CN peut uniquement être utilisée pour les entraînements identifiés comme "programmables" dans les paramètres machine "MAIN/Dr[i]/DelayedErrorReaction" (MP 1001 00012).
 - Pour les ensembles Gantry, seul l'axe maître est programmé.
 - Une remise à zéro du système désactive de nouveau la réaction temporisée en cas d'erreurs activée par la fonction "DERDelay".

6.38 Dimension du plan de travail "DimPlane, DimP"

Description : Active le calcul des valeurs minimale et maximale dans les trois systèmes de coordonnées Acs (coordonnées d'axe), Mcs0 (coordonnées de machine (sans décalage d'origine)) et Bcs (coordonnées de base). Les valeurs extrêmes sont déterminées individuellement pour chaque coordonnée. On obtient ainsi la dimension des plans de travail.

Syntaxe :	DimPlane(1) ou DimPlane	Activer le calcul "Dimension du plan de travail"
	DimPlane(0) ou DimPlane()	Désactiver le calcul "Dimension du plan de travail"
Format abrégé : DimP(..)		

Tab. 6-52: Syntaxe DimPlane

Exemple : Programme:

```

:
N100   DimPlane(1)      ;Calculation "Dimension of working plane" on
:
N500   G1 x11 y22       ;programmed traversing motions
:
N900   DimPlane()       ;Calculation "Dimension of working plane" off
N1000  G0               ;target block (e.g. for block search)
:

```

Toutes les valeurs extrêmes apparues entre les blocs N100 et N900 sont déterminées.

- Particularités et restrictions :**
- Les valeurs extrêmes sont enregistrées en interne dans la donnée système volatile spécifique au canal "/SysDimPlane". Les éléments de structure correspondants sont définis dans le schéma "sddimp.xsd".

Données générales (SysDimPlane) :

Élément de structure	Description
MinAcs	Valeurs minimales des axes du canal
MaxAcs	Valeurs maximales des axes du canal
MinMcs0	Valeurs minimales des coordonnées de machine sur le canal (sans décalage d'origine)
MaxMcs0	Valeurs maximales des coordonnées de machine sur le canal (sans décalage d'origine)
MinBcs	Valeurs minimales des coordonnées de base sur le canal
MaxBcs	Valeurs maximales des coordonnées de base sur le canal

Tab. 6-53: Vue d'ensemble des éléments de structure SysDimPlane (données générales)

Fonctions CN avec syntaxe de langage standard

- La donnée système "/SysDimPlane" doit être créée par l'utilisateur même. Des modèles correspondants sont préparés dans la commande (voir documentation "Description des fonctions – Fonctions spéciales Rexroth IndraMotion MTX", chapitre "Dimension du plan de travail").
- Toutes les valeurs sont remises à zéro après la montée en régime de la commande.
- Toutes les valeurs sont initialisées avec l'activation (c.-à-d. avec la programmation de "DimPlane(1)") : Valeurs minimales = infini / Valeurs maximales = -infini.
- Les valeurs calculées restent en suspens jusqu'à la prochaine activation sur la donnée système.
- L'interpolation brute entre "DimPlane(1)" et "DimPlane(0)" est active dans l'opération de calcul de l'avance de bloc. Cela ralentit l'avance de bloc, mais la commande numérique est ainsi en mesure de déterminer les valeurs minimales et maximales sans mouvement de déplacement.

6.39 Réglage en hauteur pour la numérisation "DistCtrl, DCR"

Description : Maintient constant l'écart entre la surface balayée et le dispositif de mesure (par ex. laser) lors de la numérisation. Cela garantit que la zone de travail disponible pour le dispositif de mesure n'est pas dépassée.



Pour une description détaillée de la fonction, voir la documentation "Description des fonctions IndraMotion MTX".

Syntaxe :

DistCtrl l(1) ou DistCtrl	Démarre le réglage en hauteur et reprend la distance actuelle entre le dispositif de mesure et la surface en tant que valeur de référence. Les données de configuration définies dans le paramètre machine 7050 007xx prennent effet ici.
DistCtrl l() ou DistCtrl l(0)	Termine le réglage en hauteur, reprend la valeur de correction actuelle et arrête le mouvement des axes. S'il est programmé conjointement avec un mouvement de déplacement dans le même bloc, le réglage en hauteur n'est désactivé qu'après l'exécution du mouvement.
DistCtrl l(<fct>)	Masque quelques données de configuration spécifiques à la fonction dans les paramètres machine.
Format abrégé : DCR (...)	
avec	
<Fct>	DcAxis (<axe> ,) <corr>) Masque MP 7050 00702. Format abrégé : DCA (...)
	<axe> Nom ou numéro de l'axe du canal qui doit être réglé en hauteur.
	<corr> Direction de mouvement dans laquelle les valeurs de correction doivent être calculées : +1 ou 1 : Dans la direction de mouvement positive -1 : Dans la direction de mouvement négative

Fonctions CN avec syntaxe de langage standard

<fct>	DcFilter (<temps>) Masque MP 7050 00730. Format abrégé : DCF (...)	
	<temps>	Paramétrage des filtres pour le lissage des valeurs de la sonde. 0 : Filtre DÉSACTIVÉ > 0 : Filtre ACTIVÉ ; temps de lissage en ms
<fct>	DcLimit ({<vitesse>}, {<accél>}) Masque MP 7050 00740 ou MP 7050 00741. Format abrégé : DCL (...)	
	<vitesse>	Vitesse de modification maximale de la valeur de correction. Masque MP 7050 00740. Valeur entrée en fonction du système de mesure actif (G71,G70) en mm/min ou en pouces/min.
	<accél>	Pente (accélération) maximale de la valeur de correction. Masque MP 7050 00741. Valeur entrée en fonction du système de mesure actif (G71,G70) en m/s² ou en 1000 pouces/s².
<Fct>	DcMon ({<collision>}, {<trou>}) Masque MP 7050 00750 ou MP 7050 00751. Format abrégé : DCM (...)	
	<collision>	Plage de tolérance pour la détection de collision Masque MP 7050 00750. Valeur entrée en fonction du système de mesure actif (G71,G70) en mm ou en pouces. 0 : Détection de collision DÉSACTIVÉE
	<trou>	Plage de tolérance pour la détection de trous Masque MP 7050 00751 Valeur entrée en fonction du système de mesure actif (G71,G70) en mm ou en pouces. 0 : Détection de trous DÉSACTIVÉE
DcBreak	Interrompt le réglage en hauteur. La valeur de correction actuelle reste active.	
Format abrégé : DCB		
DcCont	Poursuit un réglage en hauteur interrompu plus tôt par DCB. La commande numérique équilibre la différence par rapport à la valeur de référence aussi vite que possible.	
Format abrégé : DCC		

Tab. 6-54: Syntaxe DistCtrl (DCR)



Les expressions entre parenthèses "(0)" et "(1)" ne sont pas des paramètres de la fonction CN. La valeur numérique entre parenthèses ne peut donc pas être transmise par des variables CPL.

6.40 Déclenchement et arrêt de l'enregistrement du diagnostic "DREC_START", "DREC_STOP"

Description : Les ordres CN "DREC_START" et "DREC_STOP" permettent de déclencher et de mettre fin à l'enregistrement du diagnostic (oscilloscope d'axe) via le programme pièce.

Si le lancement de l'enregistrement doit être retardé, cela est possible via la programmation de G4 <temporisation> dans le même bloc CN.

Syntaxe :

DREC_START	Déclencher l'amorce de démarrage pour l'enregistrement du diagnostic à partir d'un programme pièce.
DREC_STOP	Arrêter l'enregistrement du diagnostic.

Tab. 6-55: Syntaxe DREC_START, DREC_STOP

Exemple :

N50 DREC_START G4 F5 ; Déclencher l'enregistrement du diagnostic avec une temporisation de 5 s.

...

N100 DREC_STOP ; Arrêter l'enregistrement du diagnostic.

6.41 Couplage de position finale "EndPosCouple, EPC"

Description : Couple deux coordonnées synchrones actuelles au sein d'un canal selon la relation suivante :

Δ coordonnée esclave = facteur de couplage * Δ coordonnée maître.

Afin de pouvoir effectuer le calcul de la coordonnée esclave qui en résulte, les définitions de position de la coordonnée maître sont toujours converties automatiquement en courses incrémentales.

Pour chaque bloc CN dans lequel une définition de position de la coordonnée maître est programmée, la commande calcule la position finale nécessaire de la coordonnée esclave selon le rapport indiqué ci-dessus.

Par rapport aux autres fonctions CN, la coordonnée esclave se comporte comme si sa course de déplacement avait été inscrite spécialement dans le programme pièce au moyen de la programmation relative locale (IC). Cela signifie, par exemple, que l'avance actuelle agit sur le **mouvement sur trajectoire résultant**.

Syntaxe :

EndPosCouple (<coord. M>, <coord. S>, <facteur>)	
	Couplage de position finale ACTIVÉ
EndPosCouple () ou EndPosCouple (0)	Couplage de position finale DÉACTIVÉ
Format abrégé : EPC (...)	
avec	
<coord M>	Nom de la coordonnée maître

Fonctions CN avec syntaxe de langage standard

<coord S>	Nom de la coordonnée esclave
<facteur>	Rapport entre la course de déplacement de l'esclave et la course de déplacement du maître.

Tab. 6-56: Syntaxe EndPosCouple (EPC)

Exemple :	N10 G18 G0 Z0	Plan actif : z,x avec vitesse rapide sur z=0
	N20 G1 F1000 Z3 X1	Positionnement sur P1
	N30 EPC (Z, X, 1)	Activer le couplage de position finale
	:	Coordonnée maître : Z ;
	:	Coordonnée esclave : X
		Course de déplacement de l'esclave = course de déplacement du maître
	N40 Z4	Positionnement sur P2
	N50 X1	Positionnement sur P3
	N60 Z5	Positionnement sur P4
	N70 X3	Positionnement sur P5
	N80 Z0	Positionnement sur P6
	N90 EPC ()	Couplage de position finale DÉSACTIVÉ
	:	

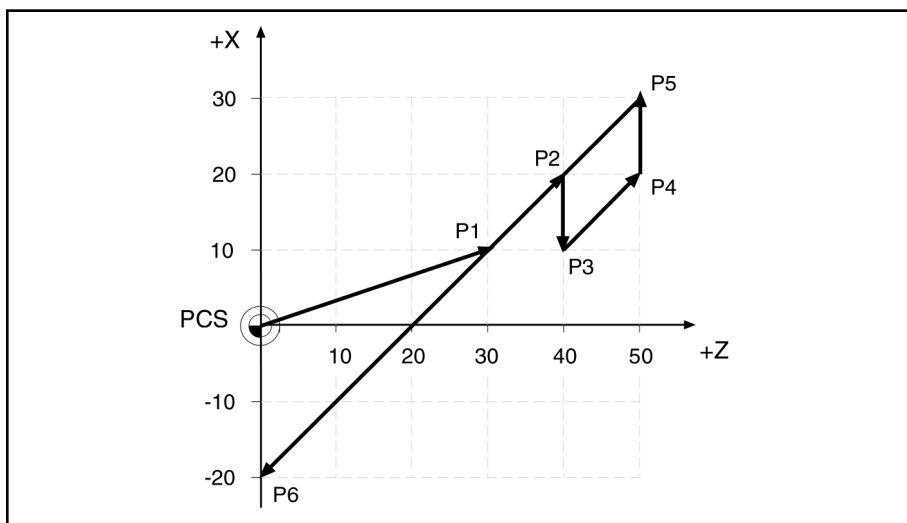


Fig. 6-9: Exemple EndPosCouple (EPC)

Particularités et restrictions :

- Le couplage de position finale n'est pas possible avec des coordonnées sans fin (coordonnées dont les valeurs sont transformées automatiquement dans la plage de valeurs définie lors du dépassement de leurs limites de zone).
- La programmation de coordonnées polaires n'est pas permise lorsqu'un couplage de position finale est activé.
- Si le couplage de position finale est actif, les coordonnées maître et esclave ne doivent pas être programmées dans le même bloc CN. La programmation exclusive de la coordonnée esclave dans un bloc CN est cependant admise.

Fonctions CN avec syntaxe de langage standard

- Ni la coordonnée maître, ni la coordonnée esclave ne doivent quitter le canal lorsque le couplage de la position finale est actif. Le remplacement d'axes et les modifications de la transformation active sont interdits.
- Même si la programmation du diamètre est activée pour la coordonnée maître (voir [chap. 6.36 "Programmation du diamètre DiaProg, DIA, programmation du rayon RadProg, RAD" à la page 282](#)), la valeur de rayon correspondante est utilisée pour le couplage.

6.42 Génération de l'avance : masquage d'axes "FeedAd, FAD"

Description : Masque les coordonnées issues de la génération de l'avance.
Retire toutes les coordonnées programmées ou les axes définis dans MP 1003 00020 de la génération de l'avance.
Les coordonnées retirées sont déplacées simultanément de manière synchrone.
Par conséquent, l'avance réelle peut augmenter par rapport à la valeur F programmée.

Syntaxe :	FeedAd (1) ou FeedAd	Retirer les axes définis dans le paramètre machine 1003 00020 de la génération de l'avance.
	FeedAd (<coordonnées>)	Retirer les coordonnées programmées de la génération de l'avance.
	FeedAd (0) ou FeedAd ()	Tenir compte de toutes les coordonnées lors de la génération de l'avance.
	avec <coordonnées>	Une liste des noms de coordonnées, y compris les coordonnées d'orientation Exemple : X, Y, Z, ..., x, y, z, phi, theta, O
	Format abrégé : FAD (. .)	

Tab. 6-57: Syntaxe FeedAd (FAD)

Exemple : (l'axe Y est enregistré dans MP 1003 00020)

:	
N100 G94 G0 X0 Y0	Positionnement sur P(0;0) en vitesse rapide
N110 FeedAd()	
N120 X100 Y100 F100	Positionnement sur P(100;100)
:	Avance sur trajectoire programmée : 100 mm/min
:	Avance sur trajectoire réelle : 100 mm/min
N150 FeedAd	Masquer Y (via MP)
N160 X200 Y200 F100	Positionnement sur P(200;200)
:	Avance sur trajectoire programmée : 100 mm/min
:	Avance sur trajectoire réelle : 141,42 mm/min
N190 FeedAd(X)	Masquer X
X300 Y300 F100	Positionnement sur P(300;300)
:	Avance sur trajectoire programmée : 100 mm/min
:	Avance sur trajectoire réelle : 141,42 mm/min

Fonctions CN avec syntaxe de langage standard

Particularités et restrictions : Dans la mesure où, dans un bloc, seules les coordonnées qui sont masquées pour la génération de l'avance sont déplacées, leur avance peut être réglée par l'adresse "Oméga" au lieu de l'adresse "F".
(voir [chap. 5.4.5 "Adresse oméga \(avance\)" à la page 215](#))

 Les expressions entre parenthèses "(0)" et "(1)" ne sont pas des paramètres de la fonction CN. La valeur numérique entre parenthèses ne peut donc pas être transmise par des variables CPL.

6.43 Anticipation "FeedForward, FFW"

Description : La fonction réduit la poursuite due au système en corrigeant les valeurs définies de l'interpolateur dans l'entraînement. Il est ainsi possible d'atteindre une meilleure fidélité de contour ou un "Fonctionnement sans erreur de poursuite".

 La fonction "Anticipation" est exécutée dans l'entraînement, de manière spécifique au fabricant, et n'est activée/désactivée que par la syntaxe d'ordre indiquée à partir du programme pièce.
Vous trouverez une description détaillée de la fonction "Anticipation" dans la documentation relative à l'entraînement.

 La possibilité d'activation de l'anticipation doit être libérée pour les axes correspondants à l'aide du paramètre machine 1003 00009.

Syntaxe :

FeedForward(1) ou FeedForward	Activer l'anticipation pour tous les axes libérés selon MP 1003 00009 (les entraînements pertinents sont commutés au mode de fonctionnement secondaire 1).
FeedForward(<adr><valeur>,...)	Activer/désactiver l'anticipation pour les axes programmés (effet dépendant de <valeur>).
FeedForward() ou FeedForward(0)	Désactiver l'anticipation pour tous les axes. (Tous les entraînements pertinents sont commutés au mode de fonctionnement principal.)
Format abrégé : FFW (. .)	
avec	
<adr>	Adresse physique ou logique de l'axe
<valeur>	Information sur la course pour <adr> 0 : Désactiver l'anticipation Différent de 0 : Activer l'anticipation

Tab. 6-58: Syntaxe Feedforward (FFW)

Fonctions CN avec syntaxe de langage standard

Exemple :	N10 FFW	Activer l'anticipation pour tous les axes du canal libérés selon le paramètre AX/Dr/FeedFwd/AxFeedFwd (MP 1003 00009)
	:	
	N50 FFW(Z0)	Désactivation l'anticipation pour Z
	:	(commuter Z au mode de fonctionnement principal)
	N90 FFW()	Désactiver l'anticipation de tous les axes du canal
	:	(commuter au mode de fonctionnement principal)

- Particularités et restrictions :**
- Lors de la désactivation de l'anticipation, tous les axes du canal sont commutés à leur mode de fonctionnement principal.
 - Le paramétrage de l'anticipation dans l'entraînement n'est possible que par l'écriture des paramètres d'entraînement pertinents.
- La fonction "Writeld3" disponible permet d'écrire des paramètres Sercos via le programme pièce (voir [chap. 6.149 "Écriture de paramètres Sercos avec un numéro d'identification Sercos élargi Writeld3, WID3" à la page 445](#)).



Les expressions entre parenthèses "(0)" et "(1)" ne sont pas des paramètres de la fonction CN. La valeur numérique entre parenthèses ne peut donc pas être transmise par des variables CPL.

6.44 Anticipation pour axes asynchrones "FFWAxis, FFWA"

Description : La fonction réduit la poursuite due au système en corrigeant les valeurs définies de l'interpolateur dans l'entraînement. Il est ainsi possible d'atteindre une meilleure fidélité de contour ou un "Fonctionnement sans erreur de poursuite".



La fonction "Anticipation" est exécutée dans l'entraînement, de manière spécifique au fabricant, et n'est activée/désactivée que par la syntaxe d'ordre indiquée à partir du programme pièce.

Vous trouverez une description détaillée de la fonction "Anticipation" dans la documentation relative à l'entraînement.



La possibilité d'activation de l'anticipation doit être libérée pour les axes correspondants via le paramètre machine AxFeedFwd (MP 1003 00009).

Syntaxe :

FFWAxis({ON(<ListeAxes>)}, OFF(<ListeAxes>))	
Format abrégé : FFWA (..)	
	Permet l'activation/la désactivation de l'anticipation des axes du canal même et des axes asynchrones (y compris les axes C asynchrones).
avec	
<ListeAxes>	Liste des noms ou numéros d'axes système séparés par des virgules.

Tab. 6-59: Syntaxe FeedforwardAxis (FFWA)

ou, en alternative,

Fonctions CN avec syntaxe de langage standard

FFWAxis (1)	Permet d'activer l'anticipation de tous les axes du canal même et tous les axes asynchrones (c.-à-d. les axes libérés par MP 1003 00009).
FFWAxis () ou FFWAxis (0)	Permet de désactiver l'anticipation de tous les axes du canal même et de tous les axes asynchrones.

Tab. 6-60: Syntaxe FeedforwardAxis (FFWA)

Exemple :	N10 FFWA (ON (B, AS10, AS12))	Activer l'anticipation pour l'axe B du canal et pour les axes asynchrones AS10 et AS12
:		
	N20 FFWA (OFF (AS10, AS12))	Désactiver l'anticipation pour les axes asynchrones AS10 et AS12
:		
	N80 FFWA (1)	Activer l'anticipation de tous les axes du canal et de tous les axes asynchrones
:		
	N90 FFWA (0)	Désactiver l'anticipation de tous les axes du canal et de tous les axes asynchrones
:		

- Particularités et restrictions :**
- Les esclaves Gantry sont influencés uniquement par la programmation du maître.
 - Pour les broches/axes C, la commutation concerne uniquement le mode axe.
Après la commutation "broche -> axe C", l'axe se trouve en l'état d'interface de position (sans anticipation).
 - Le paramétrage de l'anticipation dans l'entraînement n'est possible que par l'écriture des paramètres d'entraînement pertinents.
 - L'anticipation peut être commutée uniquement à l'arrêt.
Cela signifie que le programme pièce au début du bloc FFWA doit attendre l'arrêt de tous les axes asynchrones programmés.

6.45 Génération de l'avance : groupe d'avance "FeedGrp, FDG"

Description : Définit les coordonnées générant l'avance.

L'avance sur trajectoire programmée se rapporte aux coordonnées programmées.

Toutes les coordonnées non programmées sont déplacées simultanément de manière synchrone.

Par conséquent, l'avance réelle peut augmenter par rapport à la valeur F programmée.

Syntaxe :	FeedGrp (1)	Activer le groupe d'avance standard.
	- ou -	L'avance sur trajectoire se réfère aux coordonnées ayant les significations X, Y, Z liées au travail.
	FeedGrp	
	FeedGrp (<coor-données>)	Activer le groupe d'avance nommé. L'avance sur trajectoire se rapporte aux coordonnées programmées.

Fonctions CN avec syntaxe de langage standard

FeedGrp (0) - ou - FeedGrp ()	Aucun groupe d'avance programmé actif.
avec <coordonnées>	Une liste des noms de coordonnées, y compris les coordonnées d'orientation Exemple : X, Y, Z, ..., x, y, z, phi, theta, O phi, theta et O ont le même effet.
Format abrégé : FDG (. .)	

Tab. 6-61: Syntaxe FeedGrp (FDG)

Exemple:

FeedGrp (X, Y, Z)

Les coordonnées X, Y et Z génèrent les avances.

FeedGrp (x, y, z, O)

Les coordonnées x, y, z et le vecteur d'orientation O (et donc les coordonnées d'orientation phi et theta) génèrent les avances.

Exemple :

```

:
N100 G94 G0 X0 C0
N110 FeedGrp (X, Y)    Les coordonnées X et Y génèrent des avances.
N120 X100 C50 F100    Le mouvement X est exécuté avec F100. C est entraîné.
N130 C100 F100        Aucun mouvement XY présent.
:                      Le mouvement C est exécuté avec la valeur F.
N200 FeedGrp (1)      Le groupe d'avance standard (ici X, Y, Z) est actif.
N210 X200 C150 F100    Le mouvement X est exécuté avec F100. C est entraîné.
N220 C200 F100        Aucun mouvement XYZ présent.
:                      Le mouvement C est exécuté avec la valeur F.
N300 FeedGrp (C)      La coordonnée C génère des avances.
N310 X300 C250 F100    Le mouvement C est exécuté avec F100. X est entraîné.
N320 C300 F100        Le mouvement C est exécuté avec F100.
:

```

Particularités et restrictions :

- Si seules des coordonnées n'appartenant pas au groupe d'avance sont déplacées, l'avance programmée se rapporte à ces coordonnées.
- Si le groupe d'avance standard FeedGrp(1) est actif, le groupe d'avance est automatiquement adapté lors de la programmation du changement de plan avancé (G17(...), G18(...), G19(...)) et lors de la commutation d'une transformation d'axe (Coord(...)).
- Lorsque le groupe d'avance nommé est actif, les axes qui ont été retirés brièvement du canal sont réintégrés au groupe d'avance après leur retour sur le canal.

Fonctions CN avec syntaxe de langage standard

6.46 Génération de l'avance : type de groupe d'avance "FeedGrp-Type, FDGT"

Description : Quand le groupe d'avance FeedGrp/FDG est activé, il est possible d'influencer l'évolution de la vitesse sur trajectoire lors du changement de bloc.

Syntaxe :

FeedGrpType (SMOOTH) - ou - FeedGrpType (1)	La MTX calcule une avance sur trajectoire modifiée à partir de l'avance F programmée pour les membres de FeedGrp.
FeedGrp (STRICT) ou FeedGrpType (0)	L'avance F programmée se réfère directement aux membres du groupe d'avance. Les coordonnées restantes sont entraînées.
Format abrégé : FDGT (. .)	

Tab. 6-62: Syntaxe FeedGrpType (FDGT)

Exemple : Un groupe d'avance est actif. L'avance programmée se rapporte aux membres du groupe d'avance.

FeedGrpType(SMOOTH)

L'avance sur trajectoire interne est adaptée de telle sorte que les coordonnées programmées avec FeedGrp se déplacent avec l'avance programmée. Dans le passage du programme, les vitesses suivantes sont possibles :

Programme:

```
N10 G0 X0 V0
N20 FeedGrp(X) FeedGrpType(SMOOTH)
N20 G1 F1000 X10 V20 ; Fcont = 2236.068, Fx = 1000, Fv = 2000
N30 X20 V100 ; Fcont = 8062.258, Fx = 1000, Fv = 8000
```

Lors du passage du bloc N20 au bloc N30, l'avance sur trajectoire Fcont commence à augmenter de 2236 à 8062. Ceci entraîne un changement de vitesse de Fv relativement doux.

FeedGrpType(STRICT)

Dans le passage du programme, les vitesses suivantes sont possibles :

Programme:

```
N10 G0 X0 V0
N20 FeedGrp(X) FeedGrpType(STRICT)
N20 G1 F1000 X10 V20 ; Fcont = 1000, Fx = 1000, Fv = 2000
N30 X20 V100 ; Fcont = 1000, Fx = 1000, Fv = 8000
```

Lors du passage du bloc N20 au bloc N30, l'avance sur trajectoire Fcont ne change pas. Seule la réserve d'accélération de l'axe V limite la vitesse sur trajectoire effective au niveau du changement de bloc.

Particularités et restrictions : La variante SMOOTH est prédéfinie lors de la montée en régime de la commande.

6.47 Mesure au vol "FlyMeas, FME"

Description : La fonction « Mesure au vol » (FME) :

- complète un ordre de déplacement ;
- permet d'effectuer une mesure en cours de déplacement (d'où "Mesure au vol") ;
- utilise la fonction du palpeur de mesure de l'entraînement ;
- agit bloc par bloc ou est une fonction locale ;

Fonctions CN avec syntaxe de langage standard

- (ou l'ordre de déplacement) utilise le type d'interpolation du groupe modal actif G0, G1, G2, etc. ;
- n'arrête pas le traitement du bloc suivant ; le cas échéant, un bloc WAIT doit être programmé explicitement.

Syntaxe :

FME (MpiAxis <i>) <ordre de déplacement> ou FlyMeas (MpiAxis <i>) <ordre de déplacement>	
« MpiAxis »	Doit toujours être programmé avant l'index <i>.
<i>	Index d'axe système de l'axe de mesure
<ordre de déplacement>	Contient les coordonnées (d'axe) sur lesquelles le déplacement doit s'effectuer.

Tab. 6-63: Syntaxe FME (FlyMeas)

Exemple :

```

:
N100 IME(MpiAxis 1) ;           Initialisation de l'axe de mesure
:
N110 G0 X0 Y0 ;                 Déplacement sur X0, Y0
N120 FME(MpiAxis 1) G1 X20 Y20 ; FME lors du déplacement sur X20,
:                               Y20
----- ou -----
:
N100 IME(MpiAxis 1)
:
N110 G1 X0 Y0
N120 FME(MpiAxis 1) X20 Y20 ;   G1 a un effet modal
:
    
```

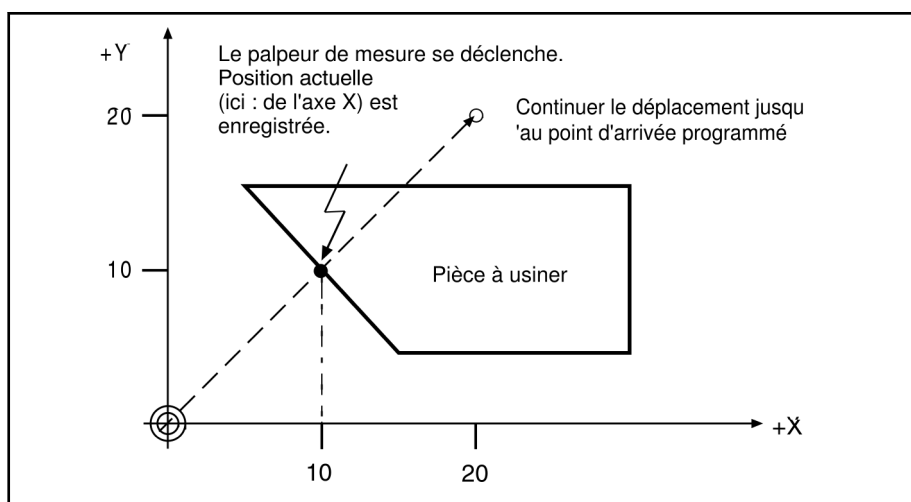


Fig. 6-10: Exemple FME (FlyMeas)

Particularités et restrictions :

- Mise en service :
 Avant de pouvoir programmer une fonction FME ou de pouvoir l'initialiser avec IME, la "Mesure au vol" (FME) doit être mise en service. Tou-

Fonctions CN avec syntaxe de langage standard

tes les étapes requises sont expliquées dans la documentation "Description des fonctions IndraMotion MTX".

- Avant la première programmation d'un axe de mesure avec la fonction FME, celle-ci doit d'abord être initialisée avec IME.
- L'utilisateur doit créer un programme CPL approprié à la suite d'un bloc FME, qui lui permette d'accepter la valeur mesurée dans le programme pièce (avec PCSPROBE() ou PROBE()).
- La valeur mesurée ne prend pas immédiatement effet dans le programme pièce. L'effet se produit au plus tôt après le nombre total de blocs défini dans le chemin NCO/LookAh/Ch[k]/**NofBlkPrep** pour la prévisualisation des blocs (Look-Ahead).

Le nombre de blocs cité peut être limité avec la fonction BlkNmb (BNB) ou un bloc WAIT doit être programmé explicitement le cas échéant.

- La fonction FME peut aussi être utilisée pour les axes sans fin linéaires si les coordonnées programmées ont un signe positif. Le recul avec le palpeur de mesure (programmation de coordonnées négatives) ne fournit pas de valeur univoque.

6.48 Mesure en butée fixe "FsProbe, FSP"

Description : Le couple actuel d'un axe sélectionné est surveillé tandis que la commande déplace tous les axes synchrones programmés par interpolation linéaire dans l'avance indiquée vers le point d'arrivée programmé.

Si le couple de cet axe dépasse une valeur limite configurable lors du mouvement, la commande déclenche les actions suivantes :

- Émission du signal IF de l'axe "Butée fixe atteinte"
- Mémorisation de la position réelle
- Freinage du mouvement sur trajectoire à $v = 0$ avec l'accélération maximale admissible
- Suppression de la course restante
- Suppression de "FsProbe" ("FsProbe" agit bloc par bloc)



N'utiliser "FsProbe" qu'en combinaison avec un programme CPL pour l'évaluation.

Syntaxe :

FsProbe (MfsAxis<i>) <coordonnées> <avance>	
	"Mesure en butée fixe" pour l'axe de mesure <i> ACTIVÉE. Déplacer les coordonnées programmées avec une interpolation linéaire.
- ou -	
FsProbe (MfsAxis (<k>{,<seuil>})) <coordonnées> <avance>	
	"Mesure en butée fixe" pour l'axe de mesure <k> ACTIVÉE. Si le <seuil> est programmé, cette valeur est utilisée pour l'axe de mesure <k>. Déplacer les coordonnées programmées avec une interpolation linéaire.
Format abrégé : FSP (. .)	
avec	

Fonctions CN avec syntaxe de langage standard

<i> - ou - <k>	Numéro d'axe système de l'axe de mesure Nom ou numéro d'axe système de l'axe de mesure
<seuil>	Seuil de couple Valeur entrée : en % du couple maximal Si <seuil> n'est pas programmé, le paramètre machine 1003 00031 prend effet (valeur limite de couple – butée fixe).
<coordon- nées>	Position sur laquelle doit avoir lieu le déplacement pendant la mesure.
<avance>	Avance sur trajectoire voulue Est limitée par MP 1005 00030 (Avance maximale pour le déplacement sur butée fixe) et par MP 1005 00002 (Vitesse maximale de l'axe et en avance rapide).

Tab. 6-64: Syntaxe FsProbe (FSP)

Exemple :

<pre> N100 FSP (MfsAxis (1, 30)) X100 F500 110 IF SD (9) = 0 THEN 120 XPOS = PROBE (1) : N130 (MSG, CONTACT) 140 GOTO N180 150 ENDIF N160 (MSG, AUCUN CONTACT) N170 MO N180 ... </pre>	Activer "Mesure sur butée fixe" pour le premier axe physique et approcher la position X100 à F500. Régler le seuil de couple sur 30 % du couple maximal. Interrogation pour savoir si le seuil de couple a été dépassé. Enregistrement de la position au moment de la commutation du 1er axe (axe X) dans la variable XPOS. ARRÊT du programme

Particularités et restrictions :

- Dans le bloc "FsProbe", les fonctions suivantes sont interdites :
 - G75 (Déplacement contre le palpeur de mesure),
 - InitMeas/FlyMeas (Mesure au vol),
 - RedTorque (Réduction du couple),
 - FsMove/FsReset/FsTorque (Déplacement sur butée fixe).
- « FsProbe » doit être programmé avec au moins une coordonnée. Sa valeur représente la profondeur de recherche maximale avant laquelle la "butée fixe" devra être atteinte au plus tard.
- « FsProbe » arrête implicitement la préparation des blocs suivants. La programmation de WAIT est donc inutile.
- L'évaluation consistant à déterminer si la "butée fixe" est atteinte, la poursuite du programme (suite à une erreur), les suivis de sécurité, la génération de messages d'erreur etc. doivent être réalisés au sein du programme CPL.
- SD(9) émet une requête pour savoir si la butée fixe a été atteinte.

Fonctions CN avec syntaxe de langage standard

- Il est possible d'interroger la position de la butée fixe avec PCSPROBE ou PROBE.

6.49 Déplacement sur butée fixe "FsMove, FSM", "FsTorque, FST", "FsReset, FSR"

Description :



La fonction "Déplacement sur butée fixe" s'applique également aux axes asynchrones !

La fonctionnalité globale "Déplacement sur butée fixe" comprend les fonctions partielles suivantes :

1. "Réduction du couple en butée fixe" : FsTorque, FST

Réglage du couple maximal que l'entraînement peut fournir après l'activation de "Déplacement sur butée fixe".

La valeur programmée agit uniquement sur le premier "déplacement sur butée fixe" qui suit.

Si "FsTorque" n'est pas utilisé, le paramètre MPS/Dr/MaxTorqPoStop (MP 1003 00031) prend effet.

2. "Déplacement sur butée fixe" : FsMove, FSM

Lancer le mouvement en direction de la butée fixe, en tenant compte du couple maximal autorisé.

Si le couple maximal autorisé (voir point 1.) est atteint au cours de ce mouvement, alors la chaîne d'actions suivante est déclenchée dans la commande :

- Émission du signal IF de l'axe "Butée fixe atteinte".
- Freinage du mouvement sur trajectoire à $v = 0$ avec l'accélération maximale admissible.
- Réglage de la position de consigne sur :
la position réelle + 0,1 mm (ou position réelle + 0,1 degré)
- Surveillance de la position de l'axe sur :
la position de la butée fixe + paramètre MPS/Dr/MaxDevPoStop (MP 1003 00032) ("Fenêtre de surveillance butée fixe en mm ou degrés")
- Maintenir le couple spécifié sur l'entraînement concerné.
- Poursuivre le traitement du programme pièce.

Si aucune butée fixe n'est atteinte avant la fin de la trajectoire programmée (atteinte du seuil de couple indiqué, voir "FsTorque"), alors la commande génère un message d'erreur.

"FsMove" reste active au-delà du bloc "FsMove" et n'est désactivée que par "FsReset".

3. "Suppression de la butée fixe" : FsReset, FSR

Débloquer les axes et quitter la butée fixe, le cas échéant.

- Dans la mesure où des axes synchrones et/ou asynchrones ont été programmés dans le bloc "FsReset", la commande déplace tous les axes sur les points d'arrivée programmés avec l'avance définie. Le couple maximal admissible des différents axes s'applique pour ce déplacement.

Fonctions CN avec syntaxe de langage standard

- Si aucun axe n'a été programmé dans le bloc "FsReset", seuls les axes synchrones sont débloqués. Les axes asynchrones pour lesquels le "Déplacement sur butée fixe" est encore actif ne peuvent être débloqués dans ce cas que par le signal d'interface "Suppression de la butée fixe".

Syntaxe :

FsTorque(<adr><couple>)	Activer pour l'axe <adr> le couple maximal admissible <couple>. Les axes synchrones et asynchrones sont permis.
Format abrégé : FST (. .)	
FsMove . .	Activer Déplacement sur butée fixe.
<coord syn> <avance> <coord asy> <avance asy>	
	Déplacer les axes synchrones et asynchrones sur les positions finales programmées.
Format abrégé : FSM . .	
FsReset . . <coord syn> <avance> <coord asy> <avance asy>	
	Désactiver Déplacement sur butée fixe. Déplacer les axes synchrones et asynchrones sur les positions finales programmées.
Format abrégé : FSR . .	
avec	
<adr>	Adresse physique ou logique de l'axe ; axes asynchrones également possibles.
<couple>	Couple maximal en % du couple d'arrêt de l'axe correspondant. Plage de valeurs : 0 à 500 %.
<coord syn>	Positions voulues des points d'arrivée des axes synchrones dans le système de coordonnées des axes (par ex. "X100 Y100 Z100"). Les axes se déplacent indépendamment, en tenant compte de l'<avance> et du paramètre MPS/Dr/MaxVelAccStop (MP 1010 00030) (accélération maximale "Déplacement sur butée fixe").
<avance>	Avance voulue pour les axes synchrones Est programmée via l'adresse F et limitée par le paramètre MPS/Dr/MaxVelPoStop (MP 1005 00030) (Avance maximale "Déplacement sur butée fixe") et par le paramètre AX/Dr/Vel/MaxVel (MP 1005 00002) (Vitesse maximale de l'axe et en avance rapide). L'avance a un effet à part sur chaque axe synchrone participant.

Fonctions CN avec syntaxe de langage standard

<coord asy>	Positions des points d'arrivée voulues dans le système de coordonnées d'axes des axes asynchrones. Elles sont toutes approchées séparément en tenant compte de l'<avance asy> et du paramètre MPS/Dr/MaxVelAccStop (MP 1010 00030) (accélération maximale "Déplacement sur butée fixe").
<avance asy>	Avance voulue pour les axes asynchrones. Est programmée via l'adresse "FA" et limitée par le paramètre MPS/Dr/MaxVelPoStop (MP 1005 00030) (Avance maximale "Déplacement sur butée fixe") et par le paramètre AX/Dr/Vel/MaxVel (MP 1005 00002) (Vitesse maximale de l'axe et en avance rapide). L'avance a un effet propre sur chaque axe asynchrone participant.

Tab. 6-65: Syntaxe FsMove (FSM), FsTorque (FST) et FsReset (FSR)

Exemple de programme de déplacement sur butée fixe

```

N100 FST(X20,Y20)      ; Restrict torque to 20% for X and Y
N110 FSM F200 X100 Y=IC(20) ; Activate moving to fixed stop for X and Y
;                       ; X travels along the axis position 100, Y travels incrementally
;                       ; Both axes each travel with the feed value 200
;
;                       ; Further part program processing
;
;
N500 FSR               ; Deactivate move to fixed stop and
;                       ; release all synchronous axes

```

- Particularités et restrictions :**
- Les fonctions suivantes ne sont pas permises lors du "Déplacement sur butée fixe" et dans le bloc "FsReset" :
 - G75 (Déplacement contre le palpeur de mesure)
 - InitMeas / FlyMeas (Mesure au vol)
 - Réduction du couple via l'interface API
 - FsProbe (Mesurer sur butée fixe)
 - FsMove doit être programmé avec au moins une coordonnée du point d'arrivée. Sa valeur représente la profondeur de recherche maximale avant laquelle la "butée fixe" devra être atteinte au plus tard.



Les positions d'axe programmées en combinaison avec FsMove et FsReset sont toujours interprétées en tant que valeurs de position absolues indépendamment de G90/G91 ! Si un déplacement incrémental est prédéfini pour un axe, l'attribut de programmation IC doit être programmé.



Les positions d'axes programmées sont des positions du système de coordonnées d'axe (ACS) et non des positions du système de coordonnées de la pièce à usiner (WCS) !



Tous les mouvements programmés en lien avec le "déplacement sur butée fixe" sont exécutés de manière asynchrone. Par conséquent, la vitesse des axes est toujours pondérée avec la position Override actuelle, indépendamment de OVD / OVE !

6.50 Compensation des erreurs de rectitude et d'angle "Geo-Comp, GCT"

Description : Active la compensation des erreurs de rectitude et d'angle pour compenser les erreurs géométriques causées par l'imprécision de la mécanique que les codeurs ne peuvent pas détecter.



Vous trouverez des informations détaillées sur la mise en œuvre et le paramétrage de la "compensation des erreurs de rectitude et d'angle" dans la documentation "Description des fonctions IndraMotion MTX".

Syntaxe :

GeoComp(...)	
Format abrégé : GCT (. .)	
GCT (1)	Activer la compensation des erreurs de rectitude et d'angle
GCT (0)	Désactiver la compensation des erreurs de rectitude et d'angle

Tab. 6-66: Syntaxe GCT(1)

- Particularités et restrictions :**
- L'utilisateur enregistre les données de compensation dans un schéma XML.
 - Ces fichiers sont sauvegardés dans le répertoire racine ou dans l'USR-FEP de la commande. Le nom de fichier peut être choisi librement ; l'extension de fichier doit être "gct".



Les expressions entre parenthèses "(0)" et "(1)" ne sont pas des paramètres de la fonction CN. La valeur numérique entre parenthèses ne peut donc pas être transmise par des variables CPL.

6.51 Reprise d'axe "GetAxis, GAX"

Description : Reprend un axe asynchrone sur le canal appelant. Ainsi l'axe asynchrone devient un axe synchrone.

L'axe peut alors être programmé sur le canal actuel via son nom physique ou logique.



Pour obtenir des informations détaillées sur la fonctionnalité "Transfert d'axe", voir la documentation "Description des fonctions IndraMotion MTX".

Syntaxe :

GetAxis (<NPA> <IPA>, {<NLA>}{, <NPA> <IPA>, {<NLA>}}...)	
Format abrégé : GAX (. .)	
avec	
<NPA>	Nom physique de l'axe Définit l'axe devant être repris sur le canal actuel.

Fonctions CN avec syntaxe de langage standard

<IPA>	Index physique de l'axe Effet similaire à <NPA>
<NLA>	Nom logique de l'axe Dans la mesure où celui-ci a été programmé, l'axe à reprendre sur le canal actuel reçoit le nom logique <NLA>. <NLA> doit être défini dans le paramètre CHAN/Ch/Coord/Acs/ChAx/ChAxName (MP 7010 00010, Nom de l'axe du canal) ou dans le paramètre CHAN/Ch/Coord/Acs/OptAxName/OptChAxName (MP 7010 00020, Désignation optionnelle de l'axe).

Tab. 6-67: Syntaxe GetAxis (GAX)

Exemple :

```

:
N030          Les axes physiques YP et ZP sont repris sur le canal appelant.
GAX (YP, , ZP, Z)  Tandis que YP est aussi adressé sur le canal appelant en tant
:                  qu'adresse YP, ZP reçoit l'adresse Z sur le canal appelant.

```

Particularités et restrictions :

- Un axe à reprendre doit être arrêté.
Si ce n'est pas le cas, la commande génère un message d'erreur et interrompt le programme.
– Contrairement à la fonction "WaitAxis" (voir [chap. 6.148 "Reprise d'axe et attente si nécessaire WaitAxis, WAX" à la page 444](#)) –
- Les positions d'axe dans un même bloc doivent toujours être programmées après GetAxis(...) et la programmation ne peut s'effectuer que si aucune transformation d'axe n'est active.
- Les axes à reprendre ne doivent participer à aucune zone de surveillance active (voir [chap. 6.7 "Surveillance de zone Area, ARA" à la page 230](#)).

6.52 Création de broches de canal "GetSpindle, GSP"

Description : Reprend une broche du système sur le canal appelant. La broche du système devient ainsi une broche du canal. Ensuite, la broche peut être programmée sur le canal actuelle à l'aide de son nom logique et de ses fonctions auxiliaires logiques.

Syntaxe :

GetSpindle (<NoBrSys> <NomBrSys> , <NoBrCan> <NomBrCan> , { . . . })	
Reprend une broche du système en tant que broche logique sur le canal (broche de canal).	
avec	
<NoBrSys>	1 à 32
<NomBrSys>	Nom de la broche du système
<NoBrCan>	1 à 8
<NomBrCan>	S1 à S8
Format abrégé : GSP	

Tab. 6-68: Syntaxe GetSpindle (GSP)

Fonctions CN avec syntaxe de langage standard

Exemple :

...	
N50 GSP (7, 2)	Reprend la 7e broche du système en tant que 2e broche de canal.
N60 M203 S2=500	Démarre la 2e broche du canal.
...	
N110 GSP (SSP12, S2)	Reprend la 12e broche du système en tant que 2e broche de canal.

Particularités et restrictions :

- Une broche du système peut exister logiquement sur plusieurs canaux.
- GetSpindle ne déclenche aucune erreur d'exécution si une broche de canal du nom indiqué existe déjà.

Exception :

La broche à remplacer fonctionne à une vitesse de coupe constante (G96).

- Si les fonctions auxiliaires en nombre entier spécifiques au canal CSP1 à CSP8 sont définies, l'API est informé de l'affectation actuelle de la broche de canal (par ex. CSP2 12 quand la broche de canal 2 est attribué à la broche du système 12).

6.53 Changement de bloc au vol via le signal à grande vitesse "HsBlkSwitch, HSB"

Description : Permet d'anticiper le changement de bloc via les entrées à grande vitesse ou les entrées client de l'interface du canal de la commande. Cela permet d'anticiper l'arrêt d'un déplacement **linéaire** en fonction d'événements externes.



Modification de la trajectoire programmée !

Dans la mesure où les points d'arrivée programmés du bloc "HsBlkSwitch" et du bloc suivant ne se trouvent pas sur une droite, l'apparition d'un événement externe génère toujours une modification de la trajectoire qu'il est impossible de prévoir avec exactitude !



Les signaux à grande vitesse peuvent être configurés à l'aide des paramètres machine.

Les points suivants s'appliquent au changement de bloc au vol :

- Sans suppression de la course restante aux modes de fonctionnement "Bloc suivant", "Bloc unique" et "Étape unique".
- Avec effacement de la course restante aux modes de fonctionnement "Bloc-programme" et "Saisie manuelle".
- **Dans la plupart des cas sans** arrêt de l'axe.
Le bloc est quitté à la vitesse actuelle (exceptions, voir "Particularités et restrictions").
- Aucun contrôle de la capacité de saut de l'axe maximale possible ainsi que
- Le mode d'action dépend du mode de fonctionnement actuel (Bloc suivant/Bloc séparé/Bloc-programme, Étape unique, Saisie manuelle). Voir "Particularités et restrictions".

Fonctions CN avec syntaxe de langage standard

Syntaxe :		HsBlkSwitch (HS<x>=<y>) ou HsBlkSwitch (CI<x>=<y>)	
		Changement de bloc au vol	
Format abrégé : HSB (. .)			
avec			
<x>		Numéro du signal à grande vitesse ou de l'entrée client (qCh_Custom<x>).	
		Plage de valeurs : 1 à 8, nombre entier.	
<y>		État du signal requis pour le changement de bloc.	
		0 : Low	
		1 : High	

Tab. 6-69: Syntaxe HsBlkSwitch (HSB)

Exemple : Avance dépendant d'un événement

L'avance effectuée doit être réduite à un itinéraire droit en fonction d'un événement externe.

Pour cela 3 blocs CN sont nécessaires dans lesquels sont définies les différentes indications d'avance. Étant donné qu'il ne faut pas ralentir à v = 0 lors de l'enchaînement de blocs, un changement de bloc au vol est nécessaire.

Tous les points d'arrivée programmés doivent impérativement se trouver sur la même droite en raison des données du problème (...sur un itinéraire droit...).



Les points d'arrivée d'un bloc suivant ne doivent pas être identiques à ceux du bloc précédent. Sinon aucun mouvement de déplacement n'est déclenché dans le bloc suivant !



La distance de déplacement programmée d'un bloc suivant influence la vitesse d'avance maximale possible au niveau de l'enchaînement de bloc. Des distances de déplacement trop courtes risquent de réduire automatique la vitesse sur trajectoire !

:		
N20G0 X0 Y0		Approche de la position de départ
N30 HSB(HS1=1) G1 X100 Y10 F500		Déplacement avec F500 jusqu'à ce que le signal à grande vitesse 1 atteigne "High" ou X100 Y10.
N40 HSB(HS1=0) X110 Y11 F100		Déplacement avec F100 jusqu'à ce que le signal à grande vitesse 1 atteigne "Low" ou X110 Y11.
N50 X120 Y12 F500		Déplacer la course restante jusqu'à X120 Y12 avec F500
:		

Fonctions CN avec syntaxe de langage standard

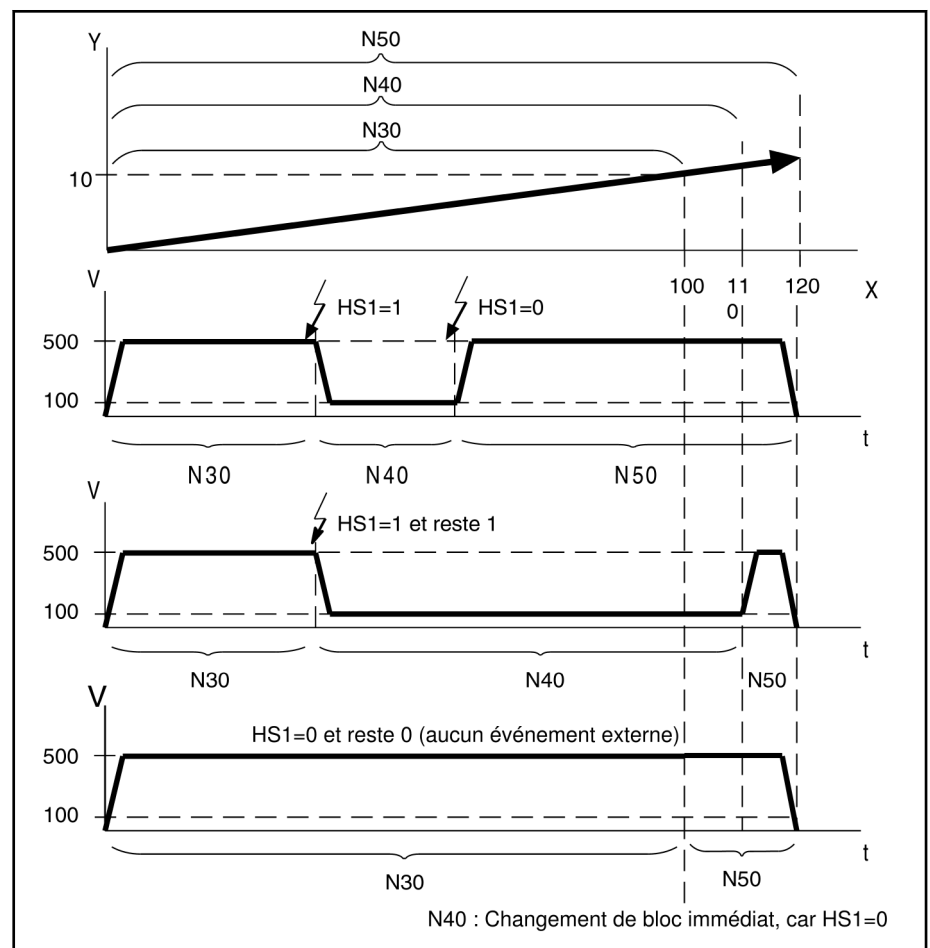


Fig. 6-11: Exemple HsBlkSwitch (HSB)

Particularités et restrictions :

- La fonction exige un mouvement de déplacement linéaire non seulement dans le bloc "HsBlkSwitch", mais également dans le bloc suivant.
- La fonction doit être écrite avec une information de course et peut être inscrite conjointement avec d'autres conditions de course.
- Aux modes de fonctionnement "Bloc suivant", "Bloc unique" et "Étape unique", les points d'arrivée de tels axes qui ne sont pas programmés dans le bloc suivant sont repris du bloc terminé par anticipation.
- *Il se produit "changement de bloc au vol" avec arrêt de l'axe dans les cas suivants :*
 - Coude de contour > 90° entre le bloc "HsBlkSwitch" et le bloc suivant.
 - L'arrêt précis, c.-à-d. G0(IPS...) ou G1(IPS...), est actif.
 - En raison d'une programmation supplémentaire, le bloc suivant commence avec v = 0 (par ex. quand la programmation KV ou l'anticipation est active).
 - Le "guidage de vitesse à limitation de secousse" (G8(SHAPE...), G9(SHAPE...) etc.) est actif.
 - Le mode de fonctionnement "Bloc unique", "Étape unique", "Bloc-programme" ou "Saisie manuelle" est actif.

6.54 Changement de bloc avec interruption via le signal à grande vitesse "HsBlkSwitch(..,HSSTOP=..), HSB(..,HSSTOP=..)"

Description : Permet d'anticiper un changement de bloc via les entrées à grande vitesse ou les entrées client de l'interface du canal de la commande. Cela permet d'anticiper l'arrêt d'un déplacement **linéaire** en fonction d'événements externes.

 **Modification de la trajectoire programmée !**
Dans la mesure où les points d'arrivée programmés du bloc "HsBlkSwitch" et du bloc suivant ne se trouvent pas sur une droite, l'apparition d'un événement externe génère toujours une modification de la trajectoire qu'il est impossible de prévoir avec exactitude !

 Les signaux à grande vitesse peuvent être configurés à l'aide des paramètres machine.

Les points suivants s'appliquent au changement de bloc avec interruption :

- Avec effacement de la course restante dans tous les modes de fonctionnement.
- Freinage toujours sur $v = 0$;
en option par saut de vitesse ou pente descendante (downslope)
Même si l'événement externe ne s'est pas produit, la vitesse est ralentie à $v = 0$.

Syntaxe :

HsBlkSwitch (HS<x>=<y>, HSSTOP=<z>) ou HsBlkSwitch (CI<x>=<y>, HSSTOP=<z>)	
	Changement de bloc avec interruption
Format abrégé : HSB (. . , HSSTOP= . .)	
avec	
<x>	Numéro du signal à grande vitesse ou de l'entrée client (qCh_Custom<x>). Plage de valeurs : 1 à 8, nombre entier
<y>	État du signal requis pour le changement de bloc. 0 : Low 1 : High
<z>	Type de freinage lors de l'apparition de l'événement : 0 : Décélération avec rampe à $v = 0$ avec accélération de freinage maximale -1 : Saut de vitesse à $v = 0$

Tab. 6-70: Syntaxe HsBlkSwitch (HSSTOP)

Exemple : Interruption du mouvement de déplacement en fonction d'un événement :

:
N20 G1 X0 Y0 F1000 Approche de la position de départ

Fonctions CN avec syntaxe de langage standard

N30 HSB (HS1=1, HSSTOP=-1) X10 F10	Déplacement de l'axe X avec F10 jusqu'à ce que le signal à grande vitesse 1 atteigne "High" ou X10. La survenue d'un événement déclenche le saut à v = 0 et la suppression de la course restante.
N40 HSB (HS2=1, HSSTOP=0) Y100 F200	Déplacement de l'axe Y avec F200 jusqu'à ce que le signal à grande vitesse 2 atteigne "High" ou Y100. La survenue d'un événement déclenche la décélération avec rampe à v = 0 avec accélération de freinage maximale et la suppression de la course restante.

- Particularités et restrictions :**
- La fonction exige un mouvement de déplacement linéaire non seulement dans le bloc "HsBlkSwitch", mais également dans le bloc suivant.
 - La fonction doit être écrite avec une information de course et peut être inscrite conjointement avec d'autres conditions de course.
 - Si "Guidage de vitesse à limitation de secousse" (G8(SHAPE...), G9(SHAPE...), etc.) est actif, **aucun** saut de vitesse ne se produit à HSSTOP = -1.

6.55 Correction en ligne dans les coordonnées de la pièce à usiner "HWOC", "HWOCDIS"

Description : La "correction en ligne" dans les coordonnées de la pièce à usiner permet de "superposer" les points suivants via une correction :

- les positions ou orientations actuelles dans le système de coordonnées de la pièce à usiner d'un canal ;
- les positions actuelles de l'axe longitudinal de l'outil dans la direction TCS-z.

La commande déduit la dimension de la correction à partir d'une manivelle raccordée (cas normal), d'un entraînement ou de la valeur d'une variable CPL.

Pour une description détaillée de la fonction, voir la documentation "Description des fonctions IndraMotion MTX".

Syntaxe :

HWOC ({ CHAN<NoCanal> } , CRDNO <NoCoord> , {STEP<incr>})	
	Correction en ligne ACTIVÉE
HWOCDIS { (CHAN<NoCanal>) }	
	Correction en ligne DÉSACTIVÉE ; la dimension de la correction actuelle reste enregistrée en interne.
HWOC () ou HWOC (0)	Correction en ligne DÉSACTIVÉE et suppression des valeurs de correction pertinentes.
avec	
<NoCanal>	Numéro du canal dans lequel la correction en ligne est activée/désactivée. Si CHAN<NoCanal> n'est pas programmé, alors le numéro actif est celui du canal actuel.

Fonctions CN avec syntaxe de langage standard

<No-Coord>	Coordonnée sur laquelle la correction en ligne doit agir. Plage de valeurs : 1 à 9 et 103. Nombre entier 1 à 8 : Numéro de coordonnée sur le canal indiqué 9 ou 103 : Coordonnées TCS
<incr>	Définition de la longueur de pas voulue par incrémentation. Seulement actif quand MP 7050 00926 = 0.

Tab. 6-71: Syntaxe HWOC, HWOC DIS

Particularités et restrictions :

- Le dépassement des commutateurs de fin de course logiciels n'est pas contrôlé pour une position de coordonnée générée par la correction en ligne.
C'est pourquoi un contrôle des commutateurs de fin de course devrait être activé côté entraînement en cas d'utilisation de la correction en ligne.
- La correction en ligne est incompatible avec les modes de fonctionnement "Configuration manuelle" (mode JOG) et "Configuration de l'approche du point de référence".
- G76 approche une position décalée de la valeur de correction actuelle.
- Les fonctions CPL PPOS et PCSPROBE ne tiennent pas compte de la valeur de la correction en ligne.
- G75 mesure la position réelle actuelle (correction en ligne incluse).
- "FsProbe" mesure la position réelle actuelle (correction en ligne incluse).



Les expressions entre parenthèses "(0)" et "(1)" ne sont pas des paramètres de la fonction CN. La valeur numérique entre parenthèses ne peut donc pas être transmise par des variables CPL.

6.56 Initialisation de la mesure au vol "InitMeas, IME"

Description : La fonction "Initialisation de la mesure au vol" (IME) :

- initialise la fonction du palpeur de mesure de l'entraînement pour la fonction "Mesure au vol" (FME) ;
- agit bloc par bloc ou est une fonction locale ;
- n'a aucune influence sur le type d'interpolation ni sur le groupe modal actif G0, G1, G2, etc.

Syntaxe :

IME (MpiAxis <i>) ou InitMeas (MpiAxis <i>)	
"MpiAxis"	Doit toujours être programmé avant l'index <i>.
<i>	Index d'axe système de l'axe de mesure

Tab. 6-72: Syntaxe IME (InitMeas)

Particularités et restrictions :

- Mise en service :

Fonctions CN avec syntaxe de langage standard

La "mesure au vol" doit être mise en service avant de pouvoir programmer la fonction IME. Toutes les étapes requises sont expliquées dans la documentation "Description des fonctions IndraMotion MTX", au chapitre "Fonctions de mesure".

- La fonction IME (InitMeas) doit être appelée avant le premier appel de la fonction FME (FlyMeas).

6.57 Programmation du temps inversé "InvTime, ITIM"

Description : ITIM agit exclusivement en mode G93. ITIM permet de définir l'unité du mot F programmé (voir [chap. 5.2.32 "Programmation de temps G93" à la page 178](#)).

Syntaxe :

ITIM <valeur>	
avec	
<valeur>	Nombre entier ; l'unité du mot F change au mode G93 en fonction de la <valeur>
0	Valeur F en [sec]
1	Valeur F en [tr/sec]
60	Valeur F en [tr/min]

Tab. 6-73: Syntaxe de la programmation du temps inversé

Pour ITIM > 0, l'unité se calcule à partir de la formule :

$$F = \frac{v \left[\frac{mm}{ITIM \times sec} \right]}{s[mm]} = \frac{v \left[\frac{1}{min} \right]_{ITIM=60}}{s[min]} = \frac{v \left[\frac{1}{sec} \right]_{ITIM=1}}{s[sec]}$$

Particularités et restrictions : La fonction ITIM peut être définie dans la chaîne d'initialisation.
 La valeur par défaut est "0" (programmation du temps en secondes).
 Si la base temporelle est commutée via ITIM, une valeur F peut être reprogrammée.

6.58 Présélection de la fenêtre d'arrêt précis "IPS1, IPS2, IPS3"

Description : Les ordres CN IPS1, IPS2 et IPS3 permettent de prérégler de façon globale la taille de la fenêtre d'arrêt précis, aussi bien pour l'avance rapide que pour l'avance.

IPS1	Fenêtre d'arrêt précis exacte
IPS2	Fenêtre d'arrêt précis approximative
IPS3	Fenêtre d'arrêt précis sans fin

Tab. 6-74: Fenêtres d'arrêt précis réglables avec IPS1, IPS2, IPS3

Si l'arrêt précis est actif pour la fonction CN actuellement active du groupe modal {G0, G1, G2, G3, G5, G6, G33}, alors ce préréglage prend effet immédiatement.

Fonctions CN avec syntaxe de langage standard

Syntaxe :

IPS1	Fenêtre d'arrêt précis exacte : À la fin du bloc, la commande décélère d'abord à la vitesse de déplacement $v = 0$. Le bloc suivant n'est déplacé qu'après que tous les axes participants ont atteint cette fenêtre d'arrêt précis.
IPS2	Fenêtre d'arrêt précis approximative : Pour le comportement, voir IPS1
IPS3	Fenêtre d'arrêt précis sans fin : À la fin du bloc, la commande ne décélère la vitesse sur trajectoire qu'à $v = 0$. Aucun contrôle de la fenêtre d'arrêt précis n'a lieu.

Tab. 6-75: Effet de IPS1, IPS2 et IPS3 si l'arrêt précis est actif pour la fonction CN actuellement active du groupe modal {G0, G1, G2, G3, G5, G6, G33}

Particularités et restriction : Si aucun arrêt précis n'est actif pour la fonction CN actuellement active du groupe modal {G0, G1, G2, G3, G5, G6, G33}, alors la taille de la fenêtre d'arrêt précis est prééglée à l'aide de IPS1, IPS2 et IPS3, pour l'avance rapide ainsi que pour l'avance normale, mais elle ne prend pas effet.

La taille de la fenêtre d'arrêt précis peut être modifiée comme suit par un nouveau paramétrage de IPS1, IPS2 et IPS3 :

- G0(IPS1), G0(IPS2), G0(IPS3) (seulement efficace pour l'avance rapide)
- G1(IPS1), G1(IPS2), G1(IPS3), G61(IPS1), G61(IPS2), G61(IPS3) (seulement efficace pour l'avance normale)



Pour plus d'informations sur l'arrêt précis, reportez-vous à G0, G1, G61/62 dans la présente documentation, ainsi qu'à la documentation "Description des fonctions IndraMotion MTX", chapitre "Précision", section "Arrêt précis".

6.59 Détermination des axes système pour la définition du filtre SHAPE optimal avec une limitation des secousses active "JerkAxList, JAL"

Description : Modifie ou définit la liste des axes système pouvant être utilisés pour le calcul d'un filtre SHAPE lorsque la limitation des secousses est active.

Cette fonction CN peut également être utilisée en tant que "paramètre machine".

Vous trouverez de plus amples informations au chapitre "Limitation des secousses spécifique aux axes" dans la documentation "Description des fonctions IndraMotion MTX".

Syntaxe :

Programmation absolue dans le programme pièce :	
JerkAxList() ou JerkAxList(DEF)	Activer la valeur par défaut issue de CHAN/Ch[k]/Ini/NcResetState
JerkAxList(<ListeAbs>)	
	Sélectionner les axes système programmés
Programmation relative dans le programme pièce :	

Fonctions CN avec syntaxe de langage standard

JerkAxlList ({DEF} {, -, <ListeMoins>} {, +, <ListePlus>})	
	Modifie la liste des axes système
Format abrégé : JAL (. .)	
avec	
DEF	Valeur par défaut issue de CHAN/Ch[k]/Ini/NcResetState
<ListeAbs> <ListeMoins> <ListeAbs>	Liste des noms d'axes système séparés par des virgules.
Programmation absolue dans CHAN/Ch[k]/Ini/NcResetState :	
JerkAxlList (<ListeAbs>)	
	Sélectionner les axes système programmés
Programmation relative dans CHAN/Ch[k]/Ini/NcResetState :	
JerkAxlList (-, <ListeMoins>)	
	Retirer les axes système programmés
Format abrégé : JAL (. .)	
avec	
<ListeAbs> <ListeMoins>	Liste des noms d'axes système séparés par des virgules.

Tab. 6-76: Syntaxe JerkAxlList, JAL

Exemple : Situation de départ :

Le 1er canal commande une unité de traitement avec 3 axes linéaires (XA, YA, ZA) et une table ronde (CA). L'axe VA est accepté temporairement sur le canal.

La table ronde (CA) est uniquement utilisée pour faire pivoter la pièce à usiner et n'a donc aucune importance pour l'usinage.

L'axe VA est important pour l'usinage.

Dans CHAN/Ch[k]/Ini/NcResetState (MP 7060 00010), saisir JerkAxlList(-,CA).

En alternative, il est également possible d'utiliser la variante suivante :

JerkAxlList(XA,YA,ZA,VA)

- Particularités et restrictions :**
- Cette fonction a un effet si elle est saisie dans CHAN/Ch[k]/Ini/NcResetState (MP 7060 00010), comme un paramètre machine.
Programmation absolue :
JerkAxlList(<ListeAxesSys>)
Programmation relative :
JerkAxlList(- , <ListeAxesSys>)

Fonctions CN avec syntaxe de langage standard

- Si cette fonction n'est pas programmée, tous les axes qui sont membres du canal ou qui deviennent membres du canal via le transfert d'axe sont utilisés pour le calcul d'un filtre SHAPE en cas de limitation des secousses active.

6.60 Activation de la limitation des secousses "JerkControl, JKC"

Description : Active la limitation des secousses sur le canal.



Pour une description détaillée de la fonction, voir la documentation "Description des fonctions IndraMotion MTX".

Syntaxe :

JerkControl ou JerkControl(1)	Activer la limitation des secousses
JerkControl() ou JerkControl(0)	Désactiver la limitation des secousses
Format abrégé : JKC (..)	

Tab. 6-77: Syntaxe JerkControl, JKC

Exemple :

N10 JKC(1) Activer la limitation des secousses
:
N90 JKC(0) Désactiver la limitation des secousses.

6.61 Mode Jog dans les coordonnées de la pièce à usiner "JogWCSSelect"

Description : Définit quelles coordonnées doivent être déplacées en Jog ou à l'aide d'une manivelle en mode de fonctionnement "Mode de réglage, mode Jog dans les coordonnées de la pièce à usiner".

Les coordonnées suivantes peuvent être sélectionnées pour le mode JOG :

- toutes les coordonnées d'axes de machine individuels (coordonnées pseudo), dans la mesure où aucune transformation d'axe n'est active.
- Toutes les coordonnées linéaires et d'orientation se rapportant au WCS actif, si une transformation d'axe adaptée à l'orientation est active.
- La coordonnée TCS-z, dans la mesure où une transformation d'axe apte à l'orientation est active.

Il dépend du réglage dans MP 7050 01010 que la vitesse d'avance et la longueur de pas (en mode JOG incrémental) sont à déduire des réglages d'un axe de machine existant ou qu'ils doivent être programmés explicitement.



Pour une description détaillée de la fonction, voir la documentation "Description des fonctions IndraMotion MTX".

Syntaxe :

JogWCSSelect{JWSCHAN<NoCanal>} JWSCOORD<NoCoord> {JWSFEED<valeur F>{ JWSSTEP<incr>}}
avec

Fonctions CN avec syntaxe de langage standard

<NoCa-nal>	Numéro du canal dans lequel doit être déplacée une coordonnée en mode JOG. Valeur par défaut : Numéro du canal dans lequel la fonction est programmée.
<No-Coord>	Sélection des coordonnées Plage de valeurs : 1 à 8 et 103 Nombre entier 1 à 8 : Numéro de la coordonnée à déplacer en mode JOG 103 : La coordonnée TCS doit être déplacée en mode JOG.
<valeur F>	Vitesse de Jog Valeur entrée en fonction du système de mesure actif (G71,G70), en mm/min, en pouces/min ou en degrés/min. Uniquement actif quand MP 7050 01010 = 0.
<incr>	Sélection d'un déplacement manuel incrémental en mode JOG et définition de la longueur de pas voulue en incréments. Uniquement actif quand MP 7050 01010 = 0.

Tab. 6-78: Syntaxe JogWCSSelect

Particularités et restrictions : La fonction est prévue pour l'emploi dans un programme spécial CN pour la sélection des coordonnées (voir également MP 7050 01110).
Pour des informations détaillées, voir la documentation "Description des fonctions IndraMotion MTX".

6.62 Suppression de la répartition cinématique des blocs CN "KinSplitDisable, KSD"

Description : Afin d'optimiser la durée de traitement, les blocs CN sont répartis en plusieurs blocs si nécessaire, pour que l'avance sur trajectoire puisse mieux s'ajuster aux différentes vitesses maximales qui dépendent de la course.

Dans certaines circonstances, ce comportement n'est pas souhaitable, par exemple si un bloc CN programmé doit être déplacé plus lentement mais à une vitesse constante. La répartition cinématique des blocs peut être désactivée à ces fins.

Syntaxe :

KinSplitDisable(1)	Désactive la répartition cinématique des blocs CN
KinSplitDisable(0)	Active la répartition cinématique des blocs CN
Format abrégé : KSD	

Tab. 6-79: Syntaxe KinSplitDisable

Particularités et restrictions : La répartition des blocs est activée après la montée en régime de la commande.

6.63 Programmation KV "KvProg, KVP"

Description : La fonction permet de modifier les valeurs KV de chaque axe via un programme. Il est possible que cette opération entraîne une augmentation temporaire de la rigidité des axes (par ex. pour le fraisage d'un alésage).

Après la montée en régime, la commande numérique lit et enregistre, entre autres, les valeurs KV actuelles de tous les entraînements raccordés.

Fonctions CN avec syntaxe de langage standard

Au cours de la programmation KV, la commande charge les valeurs KV programmées dans les entraînements pertinents (paramètre S-0-0104).

Lors de la désactivation de la programmation KV, la commande transfère les valeurs KV sauvegardées avant aux entraînements et rétablit ainsi l'état original.

Les points suivants s'appliquent ici :

- KV = (avance sur trajectoire en m/min) / (poursuite en mm)
- Avant un bloc avec modification KV, le système décélère toujours à $v = 0$, étant donné que la valeur KV dans l'entraînement ne doit être modifiée qu'à l'arrêt.
- Après l'ordre de commutation de KV, la commande attend toujours l'acquittement de tous les entraînements participants.
- La commutation de KV s'effectue immédiatement avant un mouvement de déplacement éventuellement programmé dans le même bloc.

Exemple :

```
:
N10 G1 F1000
:
N40 X40          Positionnement avec la valeur KV par défaut
N50 KVP(X2) X50  Transmettre la valeur KV 2 à l'axe X, puis exécuter le déplacement
:
```

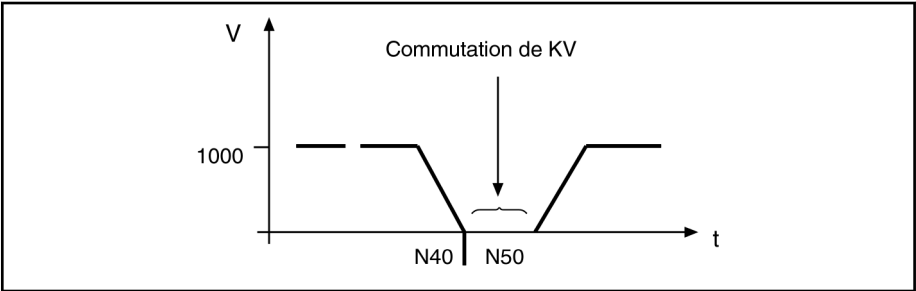


Fig. 6-12: Exemple 1 KvProg (KVP)

Syntaxe :

KvProg(<axe1>{,<axe2>}{,...})	
	Programmation KV ACTIVÉE
KvProg({0})	Programmation KV DÉSACTIVÉE
Format abrégé : KVP (. .)	
avec	
<axe>	Désignation physique (valide dans tout le système) ou logique (spécifique au canal), y compris celle de la valeur KV voulue. Valeur KV maximale programmable : 655.35

Tab. 6-80: Syntaxe KvProg (KVP)

Exemple :

```
:          Point de départ : une valeur KV de 1.0 est active dans tous les entraînements.
N10 G0 X0 Y0  Positionnement avec la valeur KV = 1.0
Z100
```


Fonctions CN avec syntaxe de langage standard

N20 Transmettre la valeur KV 2.1 à l'entraînement de l'axe physi-
KVP (Z2.1) que Z.
:
N30 G1 Z0 Positionnement avec la valeur KV = 2.1
N40 KVP () Désactiver la programmation KV
Z100 La commande numérique charge automatiquement de nouveau
: la valeur KV 1.0 dans tous les entraînements.
Ensuite, l'axe Z est déplacé.

Particularités et restrictions : Seuls des axes attribués actuellement au canal peuvent être programmés.

6.64 Compensation de poursuite "LCP"

Description : La fonction "LCP" (LagCompensation/compensation de poursuite) travaille sur la base de l'axe du système. Pour cette fonction CN, les axes doivent être adressés via leurs noms d'axes du système.

Syntaxe : **Activation de groupes :**

N100LCP (AbsMode,<esclave1>,<OnOff>,<esclave2>,<OnOff>...)		
avec		
AbsMode	Si "AbsMode" est programmé, les axes supplémentaires sont réglés autour de leurs positions zéro. Pour cela, la MLD tire les axes supplémentaires sur la position 0 lors de l'activation de la compensation. Si "AbsMode" n'est pas programmé, aucun mouvement de synchronisation n'est effectué. Les axes supplémentaires sont réglés autour de leurs positions actuelles.	
	<esclave i>	Noms d'axes du système des axes supplémentaires
	<OnOff>	0 : Désactiver le couplage 1 : Activer le couplage

Tab. 6-81: Syntaxe LCP

L'appel de cette fonction permet de coupler et découpler un ou plusieurs axes supplémentaires à leurs maîtres définis dans la configuration.

Désactivation de tous les groupes :

N200 LCP ()



La fonction est décrite en détail dans la documentation "Description des fonctions IndraMotion MTX".

6.65 Répartition du bloc de déplacement : longueur du segment de parcours "LEN"

Description : Répartit un bloc de déplacement programmé en plusieurs segments de parcours de même longueur.

Fonctions CN avec syntaxe de langage standard



Agit uniquement en combinaison avec les fonctions "Poinçonnage" (voir [chap. 6.90 "Poinçonnage Punch, PUN" à la page 366](#)) et "Grignotage" (voir [chap. 6.76 "Grignotage Nibble, NIB" à la page 332](#)).

Syntaxe :	LEN=<valeur>	
	avec	
	<valeur>	Pour les blocs linéaires : longueur du segment de parcours Pour les blocs circulaires : longueur du segment d'arc Unité de programmation, comme pour les coordonnées d'axes <valeur> ne doit pas être un diviseur entier du parcours programmé. En interne, la commande numérique génère toujours une valeur LEN effective inférieure ou égale à la valeur LEN programmée, si bien que la longueur effective du segment de parcours est toujours un diviseur entier du parcours programmé.

Tab. 6-82: Syntaxe LEN

- Exemple :
- G90 est active (programmation absolue).
 - Plan actif : X/Y
 - Position actuelle : X=0, Y=0, C=0
 - Poinçonnage/Grignotage DÉSACTIVÉ

:	
N20 X100 Y100 LEN=15	Diviser les blocs de déplacement consécutifs en segments de parcours égaux de 15 mm au maximum. LEN n'est pas encore actif, étant donné que le poinçonnage/grignotage est désactivé.
N30 X200 Y200 C180 Punch(1)	Poinçonnage ACTIVÉ LEN prend effet. La longueur de trajectoire est divisée en 10 blocs partiels. Positions de course résultante (X,Y,C) : P1 (110,110,18) P2 (120,120,36) : P10 (200,200,180)
N40 Y290 C210	La longueur de trajectoire (90 mm) est divisée en 6 blocs partiels. Positions de course résultante (X,Y,C) : P11 (200,215,185) P12 (200,230,190) : P16 (200,290,210)
N50 Punch()	Poinçonnage DÉSACTIVÉ
:	

Fonctions CN avec syntaxe de langage standard

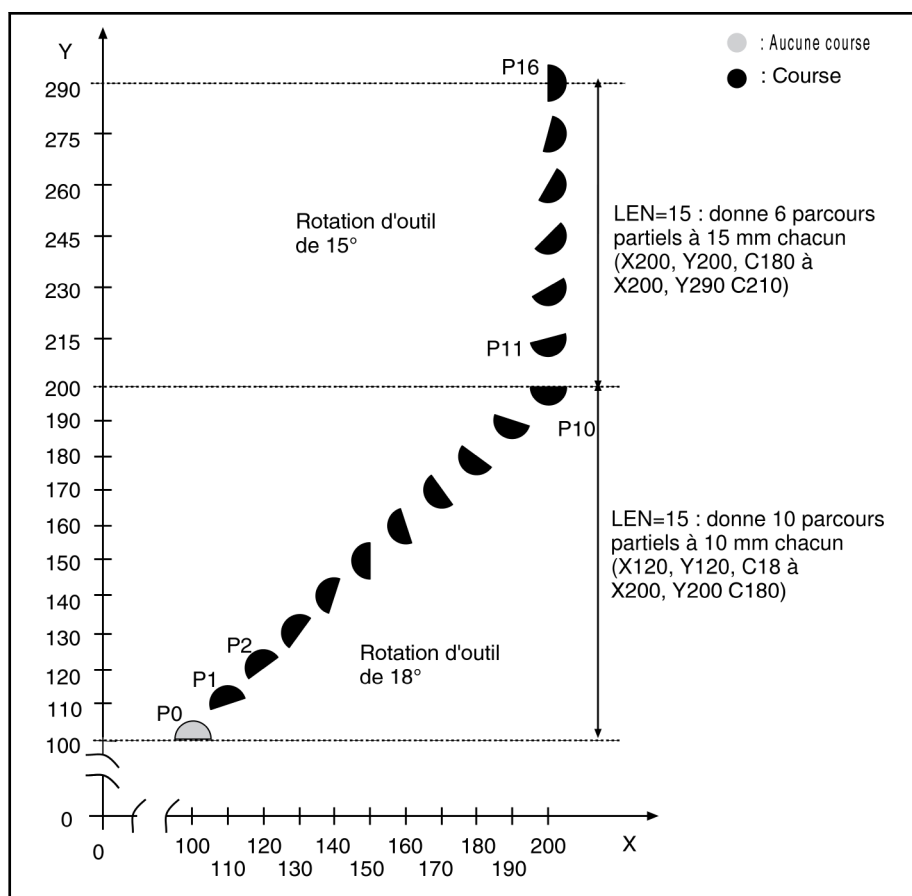


Fig. 6-13: Exemple LEN

Particularités et restrictions :

- Pour les blocs circulaires divisés, le mouvement s'effectue toujours linéairement d'une course à l'autre !
- LEN a un effet modal tant que la fonction Poinçonnage/Grignotage est active. LEN peut être masqué au sein d'un bloc par NUM (voir [chap. 6.77 "Répartition du bloc de déplacement : nombre de segments de parcours NUM"](#) à la page 334).
- LEN peut être programmé à tout moment lorsque les fonctions Poinçonnage/Grignotage (MP 8001 00010) sont libérées. La répartition des blocs ne commence toutefois qu'après l'activation du poinçonnage/grignotage.

6.66 Réglage de la puissance du laser en fonction de la vitesse sur trajectoire "LFP", "LFCConf, LFC"

Description : Règle la puissance d'un laser par signal de tension analogique (0-10 V) en fonction de l'avance sur trajectoire actuelle $v_{\text{Trajectoire}}$.



Pour une description détaillée de la fonction, voir la documentation "Description des fonctions IndraMotion MTX".

Syntaxe :

Fonctions CN avec syntaxe de langage standard

LFP (1) ou LFP	Réglage de la puissance du laser ACTIVÉ avec paramétrage actuel.
LFP ({LL (. . .)})	Réglage de la puissance du laser ACTIVÉ avec paramétrage supplémentaire, comme décrit sous LFCnt.
LFP (0)	Réglage de la puissance du laser DÉSACTIVÉ
LFCnt({LL(<Tens>,<Vmin>)}, {UL (<Tens>,<Vmax>)}, {PL (<mode>)}, {CD (<coord 1>{,<coord n>}...)} ,)	Paramétrer le réglage de la puissance du laser La sélection des coordonnées pertinentes se fait soit avec PL(..), soit avec CD(..).
Format abrégé : LFC (. .)	
avec	
<Tens>	Tension en % de la tension de sortie maximale (10 V) Associée à <Vmin> : Est indiquée si la vitesse sur trajectoire est inférieure à <Vmin>. Associée à <Vmax> : Est indiquée, si la vitesse sur trajectoire est supérieure à <Vmax>.
<Vmin>	Limite inférieure de la vitesse sur trajectoire Valeur entrée en fonction du système de mesure actif (G71,G70), en mm/min, en pouces/min ou en degrés/min.
<Vmax>	Limite supérieure de la vitesse sur trajectoire Valeur entrée en fonction du système de mesure actif (G71,G70), en mm/min, en pouces/min ou en degrés/min.
<mode>	Définit quelles coordonnées doivent être utilisées pour la détermination de v_{Tra-} jectoire : APL toutes les coordonnées du plan actuel. ASP toutes les coordonnées de l'espace actuel. CFD coordonnées correspondant à MP 7050 00820.
<coord x>	Définit quelles coordonnées doivent être utilisées pour la détermination de v_{Tra-} jectoire- Valeurs entrées : Noms des coordonnées tridimensionnelles participantes (transformation d'axe activée) ou noms logiques d'axe (pseudo-coordonnées ; transformation d'axe désactivée).

Tab. 6-83: Syntaxe LFP, LFCnt et LFC

Exemple :

LFP (LL (10,100), UL (60,700))	Réglage de la puissance du laser ACTIVÉ Paramétrage supplémentaire : Tension de sortie à 10 % (= 1 V), si la vitesse sur trajectoire tombe à une valeur inférieure à 100 mm/min, ou à 60 % (= 6 V), si la vitesse sur trajectoire augmente à une valeur supérieure à 700 mm/min.
LFP (0)	Réglage de la puissance du laser DÉSACTIVÉ

Fonctions CN avec syntaxe de langage standard

<code>LFConf (PL (APL))</code>	Paramétrage uniquement : La vitesse sur trajectoire est déduite du mouvement dans le plan actif.
<code>LFConf (CD (X , Z))</code>	Paramétrage uniquement : La vitesse sur trajectoire est déduite du mouvement des coordonnées X et Z.
<code>LFConf (UL (80, 500))</code>	Paramétrage uniquement : Limiter la tension de sortie à 80 % (= 8 V), si la vitesse sur trajectoire monte dépasse 500 mm/min.

- Particularités et restrictions :**
- Les sorties analogiques disponibles limitent le nombre de canaux pouvant utiliser la fonction.
 - Le signal de tension utilisé pour le réglage du laser tombe à 0 V dans les cas suivants :
 - Une erreur se produit (erreur d'exécution, erreur de classe d'état 1).
 - Le signal "Entraînement en service" d'un entraînement participant à la trajectoire est désactivé (aucun FG, entraînement désactivé).
 - Le signal "Arrêt de l'avance" devient actif.



Les expressions entre parenthèses "(0)" et "(1)" ne sont pas des paramètres de la fonction CN. La valeur numérique entre parenthèses ne peut donc pas être transmise par des variables CPL.

6.67 Commutation de la programmation linéaire/spline "Linear-ToSpline, LTS"

Description : La fonction convertit une interpolation linéaire programmée G1 en une programmation d'interpolation spline G6.

Syntaxe :

<code>LinearToSpline(1)</code>	Activer Tous les ordres G1 suivants sont convertis en G6.
<code>LinearToSpline(0)</code>	Désactiver Tous les ordres G1 suivants restent G1. Le format abrégé est LTS

Tab. 6-84: Syntaxe



Les expressions entre parenthèses "(0)" et "(1)" ne sont pas des paramètres de la fonction CN. La valeur numérique entre parenthèses ne peut donc pas être transmise par des variables CPL.

Exemple : L'extrait de programme

```
N10 SplineDef (2203, X, Y, Z) LTS (1)
N20 G1
a le même effet que
N10 SplineDef (2203, X, Y, Z)
N20 G6
```

6.68 Prise d'origine de l'axe modulo linéaire "LinModZp, LMZ"

Description : "LinModZp" vous permet de définir la position actuelle d'un axe modulo linéaire en tant que nouvelle origine du programme.

Ensuite, l'affichage des axes saute à la valeur "0". Une valeur de décalage est calculée dans la commande pour l'axe modulo linéaire, puis elle est ajoutée à toutes les positions d'axes programmées par la suite.

Syntaxe :	LinModZp{ (<AxeModLin>) }	
	Format abrégé : LMZ (. .)	
	avec	
	<AxeModLin>	Paramètre optionnel : Index de l'axe du système, nom de l'axe du système ou nom de l'axe du canal de l'axe modulo auquel la prise d'origine doit être appliquée. Si aucun axe n'est indiqué, alors l'origine est redéfinie pour tous les axes modulo linéaires du canal.

Tab. 6-85: Syntaxe LinModZp (LMZ)

Exemple :	LinModZp (LMZ)	
	N50 LMZ (1) X20	Définir la position actuelle de l'axe linéaire sans fin avec l'index d'axe du système 1 en tant que nouveau point d'origine du programme. Déplacer ensuite à la position X20.
	N90 LMZ (X) X-43.7	Définir la position actuelle de l'axe X en tant que nouveau point d'origine du programme. Déplacer ensuite l'axe X sur la position -43.7.
	N120 LMZ G91 X7.3	Remettre à zéro tous les axes modulo linéaires du canal, puis déplacer l'axe X de 7,3 mm dans le sens positif.

- Particularités et restrictions :**
- La plage de déplacement d'un axe modulo linéaire est limitée par les valeurs seuils "**Valeur modulo moins**" et "**Valeur modulo plus**" relatives à l'origine actuelle de l'axe.
 - La programmation d'une position à l'extérieur de la plage de déplacement définie par les valeurs seuils entraîne une erreur d'exécution.
 - Un axe modulo linéaire peut être déplacé indifféremment de façon absolue ou incrémentale à l'intérieur des valeurs seuils actuelles.
 - L'ordre LMZ permet d'accepter la position actuelle de l'axe comme nouvelle origine et de décaler la plage de déplacement en conséquence.
 - Une redéfinition incessante de l'origine permet de déplacer à l'infini un axe modulo linéaire dans une direction. Étant donné que les positions de consigne dirigées vers l'entraînement sont calculées modulo, tout dépassement des positions devient impossible.
 - Bien qu'un axe modulo linéaire soit un axe modulo côté entraînement, son comportement vis-à-vis du programme est identique à celui d'un axe linéaire ordinaire sur lequel les valeurs modulo agissent comme des commutateurs de fin de course. La seule particularité est que l'ordre LMZ permet de décaler l'origine, et donc aussi la plage de déplacement.
 - Un mouvement de déplacement peut être programmé sur une ligne avec l'ordre LMZ. Dans ce cas, l'ordre LMZ n'entraîne aucune interruption de la séquence de mouvement.



La valeur modulo est définie dans l'entraînement via le paramètre Sercos S-0-0103 lors de la montée en régime Sercos. Une valeur modulo modifiée ne prend effet qu'après une nouvelle montée en régime Sercos !

6.69 Profils de vitesse (fonctions rampe)

Description : L'étendue de la fonction permet de définir des séquences d'usinage devant être déplacées avec leurs propres profils de vitesse.

Les "modules" suivants sont disponibles pour cela :

- 1 interpolateur de marche constante
- 3 interpolateurs d'accélération et 3 interpolateurs de freinage (pour profil de vitesse de forme linéaire, sinusoïdale et $\sin^2$).

Toutes les fonctions décrites ci-dessous ont un effet modal, forment un groupe modal avec G8 et G9 et se révoquent réciproquement.

Les points suivants s'appliquent ici :

1. Interpolateurs d'accélération "LinUpFeed, LNU", "SinUpFeed, SNU", "Sin2UpFeed, S2U"

Partant de la vitesse V_0 (vitesse au début du bloc), la commande accélère sur toute la longueur de trajectoire programmée jusqu'à la vitesse cible V_1 .

Selon la fonction programmée, cela s'effectue avec une augmentation de vitesse de forme linéaire, sinusoïdale ou $\sin^2$.

La vitesse cible v_1 est atteinte au point d'arrivée programmé et s'ensuit de l'avance programmée, calibrée avec la valeur Override actuelle.

Elle est limitée par :

- l'accélération maximale sur trajectoire et
- la vitesse max. admissible sur trajectoire.

La commande calcule ces deux grandeurs en fonction des segments spécifiques à la trajectoire pour chaque bloc CN et exécute une prévisualisation pour 1 bloc en combinaison avec la vitesse maximale admissible. Ceci permet d'éviter un dépassement de la vitesse maximale des axes dans le bloc suivant respectif.



Si v_1 n'est pas supérieure à v_0 , l'appel de l'interpolateur d'accélération est ignoré dans le bloc actuel.

Comportement avec les modifications Override :

- Une augmentation de l'Override entraîne le recalcul de la rampe d'accélération.
 - Une réduction de l'Override sur des valeurs résultantes inférieures à la vitesse de départ V_0 produit les effets suivants :
 - Avec "LinUpFeed", la réduction entraîne le recalcul de la rampe de freinage qui se prolonge jusqu'au point d'arrivée programmé.
 - Avec "SinUpFeed" et "Sin2UpFeed", la réduction est ignorée.
- ### 2. Effet de l'interpolateur de marche constante "ConstFeed, CFD"

Fonctions CN avec syntaxe de langage standard

La commande tente d'atteindre la vitesse de consigne programmée en tenant compte de la vitesse maximale admissible sur trajectoire et de la position Override actuelle.

Les modifications de vitesse en cas de modifications Override sont effectuées avec les accélérations respectivement admissibles.

3. Interpolateurs de freinage "LinDownFeed, LND", "SinDownFeed, SND", "Sin2DownFeed, S2D"

Partant de la vitesse V_0 (vitesse au début du bloc), la commande freine sur toute la longueur de trajectoire programmée jusqu'à l'arrêt ($V_1 = 0$).

Selon la fonction programmée, cela s'effectue avec une réduction de vitesse de forme linéaire, sinusoïdale ou $\sin^2$.

Des modifications Override restent sans effet, exception faite dans les cas suivants :

Si l'Override a été fixé sur 0 % dans le bloc précédent et si, par conséquent, la vitesse de consigne 0 a été atteinte exactement au changement de bloc vers l'interpolateur de freinage, la commande arrête l'interpolateur de freinage jusqu'à ce que l'Override soit augmenté à une valeur > 0 .

La vitesse saute alors d'un cran d'accélération (conformément à l'accélération sur trajectoire admissible).

Sur la base de la vitesse résultante, la commande calcule la rampe de freinage nécessaire. La valeur Override effective reste alors sans influence jusqu'en fin de bloc.

Syntaxe :

LinUpFeed Format abrégé : LNU	Activer l'interpolateur d'accélération avec augmentation de vitesse linéaire.
SinUpFeed Format abrégé : SNU	Activer l'interpolateur d'accélération avec augmentation de vitesse sinusoïdale.
Sin2UpFeed Format abrégé : S2U	Activer l'interpolateur d'accélération avec augmentation de vitesse sous forme $\sin^2$.
ConstFeed Format abrégé : CFD	Activer l'interpolateur de la marche constante
LinDownFeed Format abrégé : LND	Activer l'interpolateur d'accélération avec réduction de vitesse linéaire.
SinDownFeed Format abrégé : SND	Activer l'interpolateur d'accélération avec réduction de vitesse sinusoïdale.
Sin2DownFeed Format abrégé : S2D	Activer l'interpolateur d'accélération avec réduction de vitesse sous forme $\sin^2$.

Tab. 6-86: Syntaxe LNU, SNU, S2U, CFD, LND, SND et S2D

Exemple : Profil de vitesse pour un cycle d'oscillation de l'axe U

Fonctions CN avec syntaxe de langage standard

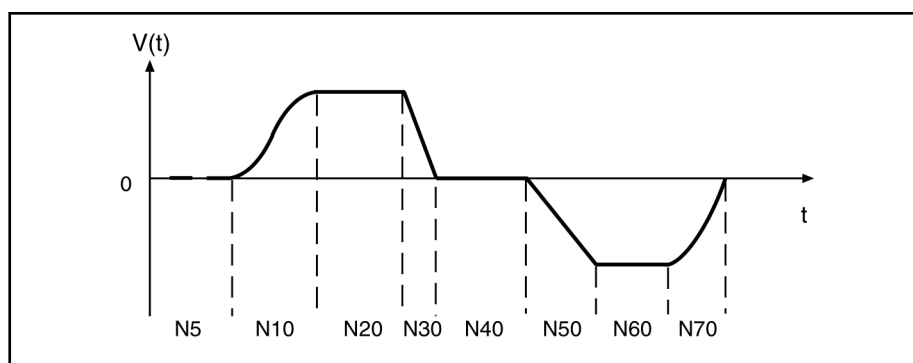


Fig. 6-14: Profil de vitesse pour un cycle d'oscillation de l'axe U

N5 G0 U10	Déplacer l'axe U à la position de départ (U = 10 mm)
N10 S2U U17 F500	Accélérer avec une courbe Sin ² jusqu'à la position U = 17 Avance de consigne au point d'arrivée : F = 500 mm/min
N20 CFD U23	Marche constante jusqu'à la position U = 23
N30 LND U29	Freinage linéaire jusqu'à la position U = 29 Vitesse finale : 0 mm/min
N40 G4 F0.5	Temporisation au point d'inversion
N50 LNU U20	Accélération linéaire jusqu'à la position U = 20
N60 CFD U17	Marche constante jusqu'à la position U = 17
N70 SND U10	Freinage sinusoïdal jusqu'à la position U = 10 Vitesse finale : 0 mm/min

Particularités et restrictions :

- La coordonnée du point d'arrivée voulu doit toujours être indiquée dans le bloc programmé.
- Toutes les fonctions rampe ne peuvent être utilisées qu'en modes "Automatique/Bloc suivant". Tout autre mode de fonctionnement (Saisie manuelle, Bloc unique, Étape unique ou Bloc-programme) entraînera une erreur d'exécution.
- Si des interpolateurs de freinage sont utilisés avec des courses de déplacement très courtes, il se peut que des dépassements d'accélération se produisent et engendrent une servoverreur.
Il convient donc de tenir compte de la dynamique maximale possible de la machine lors de l'élaboration du programme pièce.
- L'utilisation des fonctions auxiliaires ou des fonctions comme l'"arrêt précis" n'est pas autorisée lorsque les interpolateurs de marche constante et d'accélération sont actifs. L'utilisation de ces fonctions risque d'entraîner des chutes de vitesse !

Voici quelques exemples de fonctions interdites :

G0, G4, "KvProg", G63, G33, G61, "G1(IPS)", G75, "G74(HOME)", "HsBlkSwitch", "Writeld".

6.70 Changement de broche principale "MainSp, MSP"

Description : Définit la broche indiquée en tant que broche principale sur le canal actuel (réglage différent de celui dans MP 7020 00010).

Fonctions CN avec syntaxe de langage standard

De cette manière, il est possible de définir dynamiquement dans le programme pièce sur quelle broche les fonctions doivent prendre effet :

- G33 (Filetage)
- G95 (Programmation de l'avance en mm/tour) et
- G4 (Temporisation)

Syntaxe :

MainSp(<num>) ou MainSp(<broche>)	
Format abrégé : MSP...	
avec	
<num>	Numéro de la broche (index de la broche) Champ de saisie : -1 ; 1 à 8. Nombre entier -1 : Broche principale correspondant à MP 7020 00010
<broche>	Comme <num> – ou – nom de la broche (par ex. S1)

Tab. 6-87: Syntaxe MainSp (MSP)

6.71 Aide à la saisie : fonction miroir "Mirror(...), MIR(...)"

Description : La fonction "Miroir" compte parmi les aides à la saisie.

La commande traite un contour programmé ou, par exemple, un gabarit de perçage en l'inversant symétriquement.

La fonction se rapporte toujours au point d'inversion miroir actuel (voir fonction "PoleSet" ([chap. 6.83 "Définition du point d'inversion miroir/de rotation PoleSet, PLS" à la page 351](#)). S'il n'a pas été programmé explicitement, le point d'inversion miroir utilisé est le point d'origine actuel du programme.



La fonction miroir est une aide à la saisie et ne modifie donc pas le système actuel des coordonnées du programme. Une aide à la saisie n'est en fait qu'une alternative de saisie des coordonnées de programme.

La fonction miroir peut également être utilisée conjointement avec la mise à l'échelle et la rotation.

Les points suivants s'appliquent ici :

- La fonction a un effet modal. Elle reste valide jusqu'à ce qu'elle soit désactivée.
- Elle peut être programmée dans le même bloc que d'autres conditions de course et fonctions auxiliaires.

Syntaxe :

Mirror(<axe 1>1{,<axe n>1})	
	Activer la fonction miroir pour les axes indiqués.
Mirror(<axe 1>0{,<axe n>0})	
	Désactiver la fonction miroir pour les axes indiqués.
Mirror(0) ou Mirror()	Désactiver la fonction miroir pour tous les axes du canal. Les positions d'axe atteintes sont conservées jusqu'à ce qu'elles soient reprogrammées.
Format abrégé : MIR (. .)	

Fonctions CN avec syntaxe de langage standard

avec	
<axe 1>, <axe n>	La fonction active les adresses d'axes (par ex. X) en combinaison avec la valeur "1" : tous les ordres de course programmés suivants des axes correspondants (par ex. X100) sont multipliés en interne par la valeur "-1". La fonction miroir ne devient active qu'avec l'information de déplacement suivante.

Tab. 6-88: Syntaxe Mirror(...), MIR(...)

Exemples : Fonction miroir

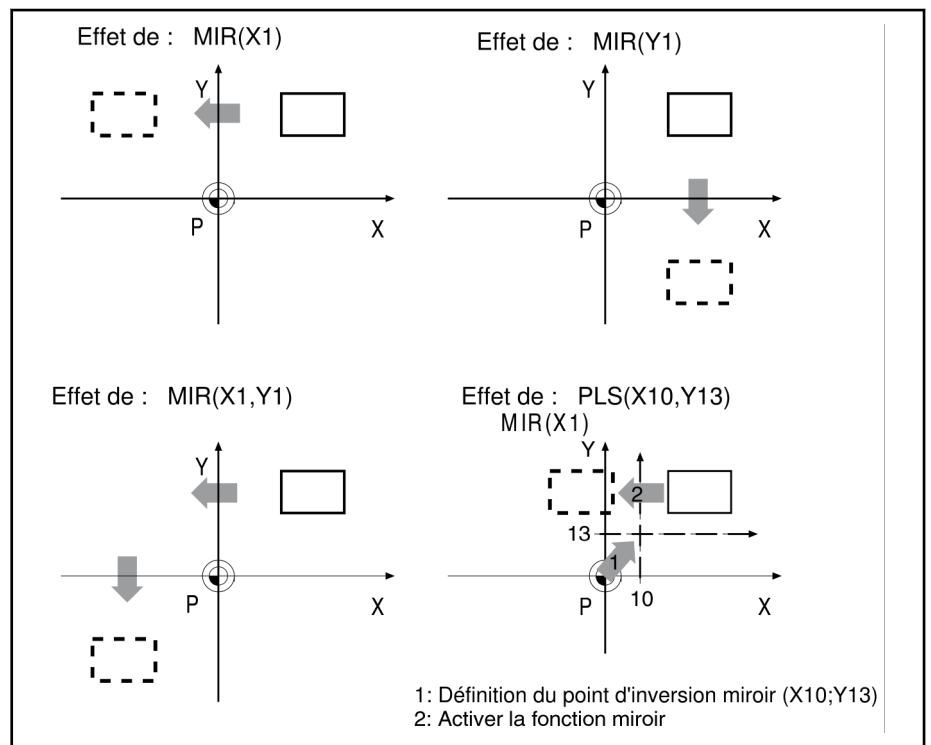


Fig. 6-15: Exemples de la fonction miroir

Particularités et restrictions :

- La fonction tient compte des paramètres d'interpolation avec une interpolation circulaire.
- Elle influence le décalage programmable des segments de contour (Shift).

Voir [chap. 6.110 "Décalage programmé du contour Shift, SHT"](#) à la page 392.

- Elle n'influence **pas** :

- les décalages d'origine (G54-G59.5 ; voir [chap. 5.2.20 "Décalages d'origine \(DO\) G53, G53.1-G59.1 à G53.5-G59.5"](#) à la page 164),
- les décalages des coordonnées de programme (Trans ou ATrans, voir [chap. 6.144 "Décalage des coordonnées du programme Trans, TRS, décalage additif des coordonnées du programme ATrans, ATR"](#) à la page 437),
- l'initialisation de la position du programme

Fonctions CN avec syntaxe de langage standard

("SetPos", voir chap. 6.109 "Initialisation de la position du programme SetPos, SPS" à la page 392),

- l'approche des coordonnées du point de référence
(G74, voir chap. 5.2.25 "Approche des coordonnées du point de référence G74" à la page 171),
- l'approche de la position fixe des axes machine
(G76, voir chap. 5.2.28 "Approche de la position fixe des axes machine G76" à la page 174),
- les valeurs de correction du rayon de la fraise et des longueurs d'outils.

Inversion d'un vecteur d'orientation :

Un vecteur d'orientation est inversé exclusivement composante par composante à l'aide de la syntaxe de fonction suivante.

Une mise à l'échelle éventuellement active ou un point d'inversion miroir/de rotation n'ont aucune influence sur le résultat.

Syntaxe :

<code>Mirror(0(<Sx>,<Sy>,<Sz>))</code>	
	Activer/Désactiver la fonction miroir pour les composantes indiquées d'un vecteur d'orientation.
Format abrégé : <code>MIR(. .)</code>	
avec	
<code><Sx>,<Sy>,<Sz></code>	Facteurs miroir pour les composantes de vecteur individuelles. 0 : Sans inversion miroir 1 : Avec inversion miroir

Tab. 6-89: Syntaxe Mirror

Pour le vecteur d'orientation s'applique ce qui suit :

- Les coordonnées polaires φ et ϑ ne peuvent pas être inversées.
- La programmation `MIR(phi1,theta1)` est interdite.

6.72 Mesure sur un contour (MOC)

Description : La fonction CN MOC permet d'effectuer une mesure le long d'une trajectoire quelconque. La trajectoire de mesure se compose de blocs générant le mouvement et est mise entre parenthèses par un MOC avec des paramètres et un sans parenthèses.



Il est recommandé d'écrire la désactivation de la mesure (`MOC()`) dans l'"état après la remise à zéro" (paramètre machine `CHAN/Ch[k]/Ini/ChResetState`). Sinon, la trajectoire de mesure ne serait pas terminée, surtout lors de la réaction Remise à zéro (RE3).

Lors de l'apparition de l'événement de mesure, les positions mesurées sont inscrites dans la donnée système `/SysMeasPos`. Les fonctions CPL de G75 (notamment `PROBE`, `SD(9)` ...) peuvent en plus être utilisées pour interroger les positions mesurées dans le programme pièce.



Si les fonctions CPL sont utilisées pour la lecture des valeurs de mesure, un `WAIT` doit obligatoirement être défini entre la fin de la trajectoire de mesure et les fonctions CPL.

Fonctions CN avec syntaxe de langage standard

Si un mouvement est programmé dans le bloc de démarrage de la mesure, la mesure commence au vol (donc sans arrêt intermédiaire). Ceci est également valable pour la fin de la trajectoire de mesure.

Dans tous les cas, un seul MOC est actif à la fois. Une mesure peut être terminée par la programmation d'une autre mesure.



Les mesures multiples sont impossibles. Cela signifie que, dans une trajectoire de mesure, la logique de MOC est uniquement prononcée jusqu'au premier événement de mesure avant d'être désactivée. Pour effectuer une autre mesure, MOC doit être reprogrammé.

Syntaxe :

MOC (MP<palpeur de mesure> CI<entrée client> HS<grande vitesse>{, ES<flanc>}, RE<réaction>{, AS<ASUP>}{, ER<erreur>}{, <NomAxeCanal1>{, <NomAxeCanal 2> ...}})	
avec	
<palpeur de mesure>	Sélectionne un palpeur de mesure (1 ou 2) comme source de signal.
<entrée client>	Sélectionne une entrée client dans l'interface du canal (1 à 8) comme source de signal.
<grande vitesse>	Sélectionne une entrée HS (1 à 8) comme source de signal.
<flanc>	Sélectionne des flancs descendants (1), ascendants (2) ou les deux (3) comme déclencheurs de la mesure. Si une entrée client ou HighSpeed a été sélectionnée comme source du signal, la sélection du flanc est un paramètre obligatoire. Si un palpeur de mesure est utilisé, la sélection du flanc est optionnelle. Si elle n'est pas programmée, alors le flanc configuré (voir les paramètres machine MEAS/Dr[i]/Probe1Edge, MEAS/Dr[i]/Probe2Edge et MPEDGE d'IndraMotion MTX, chap. 6.75 "Modification du palpeur de mesure de la configuration des flancs (MPEDGE)" à la page 331) sert de déclencheur pour MOC.
<réaction>	<i>L'une des réactions suivantes peut être déclenchée par l'événement de mesure :</i> 1. Commutation de la vitesse sur trajectoire 2. Déclencher le sous-programme asynchrone 3. Positions initiales du canal 4. Aucune réaction (mesure au vol) 5. Ignorer la trajectoire de mesure restante Les réactions possibles sont décrites en détail dans la documentation "Description des fonctions IndraMotion MTX", chapitre "Mesure sur un contour".
<ASUP>	Numéro d'un sous-programme asynchrone inscrit devant être exécuté après l'événement de mesure (uniquement avec RE2). Le sous-programme asynchrone doit être inscrit dans le système avant l'appel de MOC (par ex. avec ASPSET, voir chap. 6.8.5 "Sous-programmes asynchrones : inscription ASPSET" à la page 238).

Fonctions CN avec syntaxe de langage standard

<erreur>	Réaction lorsque l'événement de mesure n'est pas survenu sur la trajectoire de mesure. <i>Les réactions possibles sont les suivantes :</i> • 0 : Aucun message (définition) • 1 : Émettre un avertissement • 2 : Émettre un message d'erreur
<NomAxeCanalX>	Noms d'axes de tous les axes du canal à mesurer La donnée est pertinente uniquement si la mesure est effectuée avec un palpeur de mesure. Si aucun axe n'est indiqué, alors la mesure est exécutée avec tous les axes du canal configurés pour le palpeur de mesure (voir documentation "Paramètre machine IndraMotion MTX", MEAS/Dr[i]/EnablProbeFunc).
MOC ()	
	Fin de la trajectoire de mesure La mesure est désactivée et, si l'événement de mesure ne s'est pas produit, un message est généré selon ER.

Tab. 6-90: Syntaxe de la fonction CN MOC

Exemple :

N10 ASPSET(3, SAFE-POS01, 11)	; Inscrire le sous-programme asynchrone "SAFE-POS01" dans le système en tant que numéro 3.
N20 MOC(CI4, ES1, RE2, AS3, X, Z)	; Si l'entrée client 4 fournit un flanc descendant, l'ASUP 3 (SAFEPOS01) est exécuté et les axes X et Z sont mesurés
N30 X... Y... Z...	; Mesure en cours
N40 X... Y... Z...	; Mesure en cours
N50 X... Y... Z...	; Mesure en cours
N60 X... Y... Z...	; Mesure en cours
N70 MOC()	; Mesure terminée (le palpeur de mesure est désactivé)

Particularités et restrictions :

- Une seule source de signal (palpeur de mesure/entrée client/entrée HighSpeed) peut être sélectionnée.
- Si la réaction 2 (déclencher le sous-programme asynchrone) est utilisée dans une définition de bloc CN enregistrée en mémoire tampon, le tampon de blocs est supprimé après le sous-programme asynchrone. Cela signifie que la trajectoire de mesure prédéfinie est ensuite déplacée uniquement jusqu'à l'événement de mesure.
- Les axes de mesure situés sur la trajectoire de mesure ne doivent pas être supprimés du canal.
- La position mesurée dans le système de coordonnées du programme (PCS) est uniquement fournie une fois tous les axes du canal mesurés. La configuration (d'axe) active est alors prise en compte au moment de la mesure.
- Il est impossible de programmer un G75 ou un FlyMeas sur une trajectoire de mesure. Par ailleurs, il n'est pas permis de reconfigurer les flancs de mesure avec MPEDGE.

Fonctions CN avec syntaxe de langage standard

Particularités pour la réaction Passer la trajectoire de mesure restante (RE5) :

- Les décalages d'origine, les placements et les transformations ne doivent pas être modifiés sur la trajectoire de mesure.
- Les fonctions M dans les blocs passés sont activées, les fonctions modales sont commutées.
- Les fonctions CPL dans les blocs passés sont traitées.
- Si des blocs sont passés, aucun sous-programme asynchrone ne peut être démarré.
- Si aucun mouvement n'est programmé dans le bloc de la fin de la trajectoire de mesure, la commande numérique se déplace sur la dernière position de la trajectoire de mesure au sein de ce bloc avec la fonction de mouvement alors active.



Dans le bloc final de la trajectoire de mesure (`MOC()`), après les blocs passés, il y a un mouvement de déplacement, même si aucun mouvement n'est programmé explicitement.

Il est judicieux de programmer un mouvement de déplacement (y compris une fonction de mouvement) vers une position absolue et sûre dans le bloc final de la trajectoire de mesure. Cette dernière doit pouvoir être atteinte sans collision depuis chaque point de la trajectoire de mesure.



Il est recommandé d'enregistrer la fin de la trajectoire de mesure (`MOC()`) dans l'"état après la remise à zéro du canal" (paramètre machine `CHAN/Ch[k]/Ini/ChResetState`), car des erreurs d'exécution risquent de survenir pendant la remise à zéro si une erreur s'est produite lors du passage de blocs.

Particularités de l'utilisation d'un palpeur de mesure :

- Il est impossible de programmer 3 (deux flancs) comme sélection de flanc.
- Les flancs de mesure de tous les axes de mesure doivent être configurés de la même manière et correspondre à la programmation de `ES`.
- Seuls les axes dont le nom d'axe du canal est indiqué peuvent être mesurés. Si aucun nom d'axe du canal n'a été programmé, tous les axes du canal qui sont configurés pour l'utilisation du palpeur de mesure (paramètre machine `MEAS/Dr[i]/EnablProbeFunc`) sont mesurés.
- Si un axe de mesure n'est pas configuré pour l'utilisation du palpeur de mesure (paramètre machine `MEAS/Dr[i]/EnablProbeFunc`), une erreur d'exécution est générée.

6.73 Calcul modulo pour des axes sans fin à déplacement incrémental "Modulo, MOD"

Description : Si des axes sans fin sont déplacés de façon incrémentale, alors le calcul modulo est neutralisé au sein de la commande. Il est ainsi garanti que la course de déplacement réglée est vraiment parcourue. Le calcul modulo est seulement activé de manière permanente pour les valeurs de consigne de l'entraînement et les valeurs d'affichage. Pour les valeurs d'axe au sein de la commande, le calcul modulo n'est effectué que lors de la définition suivante des positions absolues.

Si un axe sans fin est déplacé de manière incrémentale dans la même direction, il finit par atteindre la limite interne de la plage de nombres présentable.

Fonctions CN avec syntaxe de langage standard

La commande acquitte cet état avec une erreur du commutateur de fin de course.

Si la somme des mouvements incrémentaux dépasse cette valeur seuil, il est nécessaire de réinitialiser de manière coordonnée les mémoires internes des axes dans la plage modulo. Cette tâche est assumée par la fonction "Modulo".

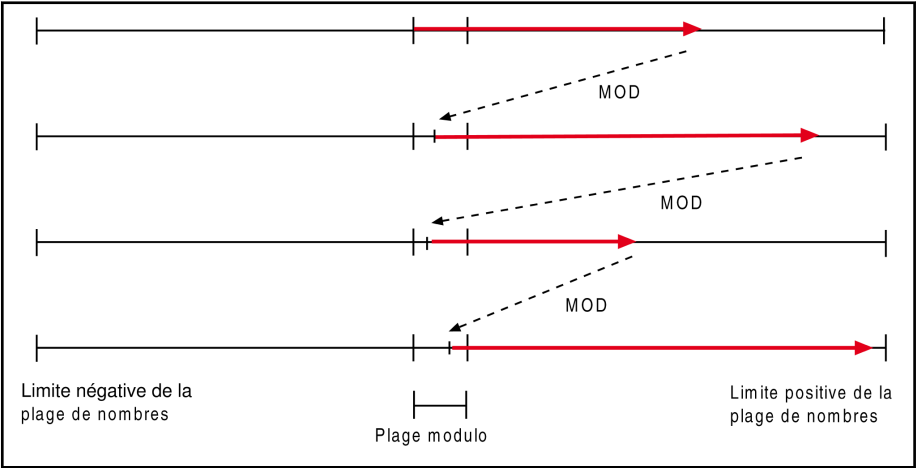


Fig. 6-16: Plage modulo

Syntaxe : Calcul modulo pour tous les axes sans fin du canal :

Modulo
Format abrégé : MOD

Tab. 6-91: Syntaxe MOD – pour tous les axes sans fin

Calcul modulo pour des axes sans fin individuels du canal :

Modulo(<NA1>{,<NA1>{, ...}})	
Format abrégé : MOD (. . .)	
avec	
<NAx>	Nom de l'axe modulo (nom d'axe du canal ou du système)

Tab. 6-92: Syntaxe MOD – pour certains axes sans fin



L'ordre modulo peut être programmé conjointement avec un mouvement de déplacement quelconque dans un même bloc. L'ordre modulo ne provoque aucune chute de vitesse.

L'ordre modulo peut aussi être appliqué aux axes ronds dans la mesure où le mode modulo a été configuré pour eux côté entraînement.

Exemple :
N030 MOD(A,B) A270 B90 F1000
:

6.74

Déplacement modulo pour les axes sans fin "ModuloShift, MODS"

Description : Permet de décaler d'un certain nombre de tours la position du programme d'un axe modulo ou d'un axe rond dont le mode modulo est configuré côté entraînement.

Fonctions CN avec syntaxe de langage standard

Syntaxe :	
MODS () { , ES1 ES2 }	
avec	
MP1 MP2	Sélectionne le palpeur de mesure 1 ou 2 pour le changement de configuration
ES1 ES2	Sélectionne le flanc descendant (ES1) ou ascendant (ES2) pour le palpeur de mesure. Si ni ES1, ni ES2 ne sont indiqués, MPEDGE réinitialise la sélection des flancs pour le palpeur de mesure au réglage de la configuration.
MPEDGE ()	
	Réinitialise les flancs de tous les palpeurs de mesure au réglage de la configuration.

Tab. 6-93: Syntaxe de la fonction CN MPEDGE

Exemple :
 :
 N030 MODS (A1, B-2)
 :

6.75 Modification du palpeur de mesure de la configuration des flancs (MPEDGE)

Description : Les paramètres machine suivants permettent de définir, pour un entraînement particulier, par quel flanc (ascendant ou descendant) de l'entrée du palpeur de mesure l'entraînement doit déclencher une mesure :

MEAS/Dr[i]/Probe1Edge et

MEAS/Dr[i]/Probe2Edge

La fonction CN MPEDGE permet de masquer cette configuration. Les nouveaux flancs agissent pour toutes les fonctions qui utilisent le palpeur de mesure (G75, MOC...).

Syntaxe :	
MPEDGE (MP1 MP2 { , ES1 ES2 })	
avec	
MP1 MP2	Sélectionne le palpeur de mesure 1 ou 2 pour le changement de configuration
ES1 ES2	Sélectionne le flanc descendant (ES1) ou ascendant (ES2) pour le palpeur de mesure. Si ni ES1, ni ES2 ne sont indiqués, MPEDGE réinitialise la sélection des flancs pour le palpeur de mesure au réglage de la configuration.
MPEDGE ()	
	Réinitialise les flancs de tous les palpeurs de mesure au réglage de la configuration.

Tab. 6-94: Syntaxe de la fonction CN MPEDGE

Exemple :	
N100 MPEDGE(MP2, ES2)	; Le palpeur de mesure 2 réagit au flanc ascendant
...	
N200 MPEDGE(MP1)	; Flanc du palpeur de mesure 1 réinitialisé aux paramètres de configuration

Fonctions CN avec syntaxe de langage standard

...	
N300 MPEDGE()	; Tous les palpeurs de mesure sont réinitialisés aux paramètres de configuration

- Particularités et restrictions :**
- MPEDGE s'arrête au début du bloc (vitesse sur trajectoire égale à zéro).
 - La configuration ne doit en aucun cas être modifiée sur la trajectoire de mesure d'un MOC.
 - Il est impossible de déclencher une mesure pour les deux flancs.

6.76 Grignotage "Nibble, NIB"

Description : Active ou désactive la fonction "Grignotage".

Si le grignotage est activé, une course est déclenchée dans les cas suivants :

- à la fin de chaque segment de parcours programmé ou généré par l'une des fonctions "NUM" (voir [chap. 6.77 "Répartition du bloc de déplacement : nombre de segments de parcours NUM" à la page 334](#)) ou "LEN" (voir [chap. 6.65 "Répartition du bloc de déplacement : longueur du segment de parcours LEN" à la page 315](#)) ;
- au début du premier segment de parcours, dans la mesure où aucun mouvement de déplacement n'a été programmé dans le plan actif ou si le grignotage a été désactivé.

Le mouvement de déplacement suivant n'est démarré qu'une fois la course terminée.



Fonctions influençant le temps de déclenchement de la course :

- PtDefault
(voir [chap. 6.88.2 "Temps de déclenchement de la course \(restauration des valeurs par défaut\) PtDefault, PTD" à la page 362](#))
- PtBlkEnd
(voir [chap. 6.88.1 "Temps de déclenchement de la course \(point d'arrivée de l'interpolation\) PtBlkEnd, PTE" à la page 360](#))
- PtInpos
(voir [chap. 6.88.3 "Temps de déclenchement de la course \(fenêtre Inpos\) PtInpos, PTI" à la page 363](#))

Syntaxe :

Nibble(1) ou Nibble	Activer le grignotage
Nibble(0) ou Nibble()	Désactiver le grignotage
Format abrégé : NIB (. .)	

Tab. 6-95: Syntaxe Nibble (NIB)

- Exemple :**
- G90 est active (programmation absolue).
 - Plan actif : X/Y
 - Position actuelle : X=0, Y=0, C=0

Fonctions CN avec syntaxe de langage standard

:	
N10 LEN=12	Diviser les blocs de déplacement en segments de parcours de 12 mm au maximum. LEN doit être programmé avant que le grignotage ne soit activé !
N20 C10 Nibble(1) G9(ASHAPE)	Activer le grignotage. La fonction modale LEN prend effet. Activer l'interpolateur avec la logique de poinçonnage L'axe C effectue une rotation à 10 degrés. Pas de course, car les axes X et Y ne sont pas programmés.
N30 X0	Pas de mouvement de déplacement, car l'axe X est déjà sur la position 0. Course, car l'axe X se situe dans le plan actif.
N40 X110	Le bloc de déplacement est divisé en 10 segments de parcours égaux de 11 mm. Course aux positions X11, X22, X33 à X99, X110. Course supplémentaire sur la position X0, car il n'y a eu aucun mouvement de déplacement dans N30.
N50 Y30 NUM=3	Masque la fonction modale LEN(N10) pour le bloc actuel. Diviser le bloc de déplacement en 3 segments de parcours égaux. Course à Y10, Y20, Y30.
N60 Y90	La fonction LEN de N10 reprend effet. Course à Y42, Y54, Y66, Y78, Y90.
N70 X50 Y50 Nibble() G8	Grignotage DÉSACTIVÉ Pente de la trajectoire ACTIVÉE
:	Mouvement de déplacement à X=50, Y=50.

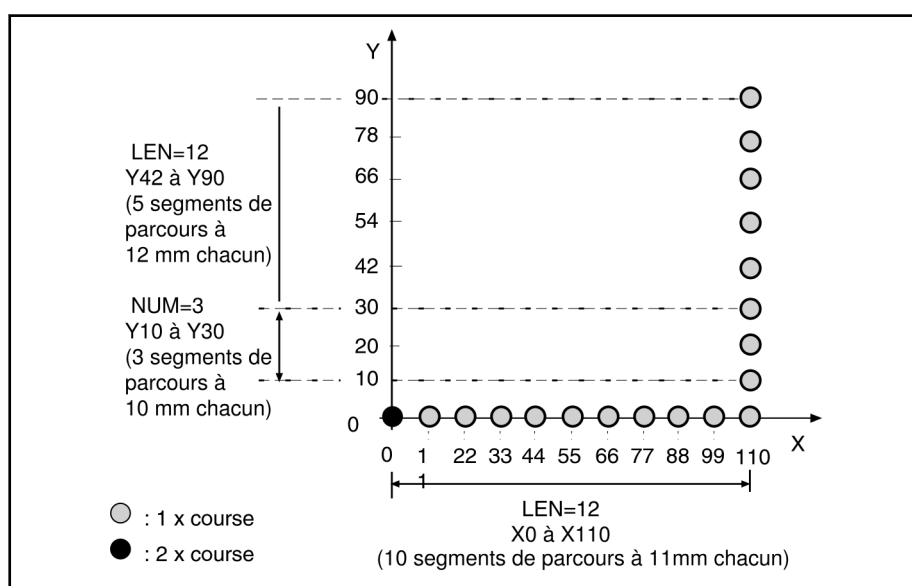


Fig. 6-17: Exemple de grignotage (NIB)



Pour utiliser la fonction Poinçonnage, un interpolateur prenant en charge la fonctionnalité de poinçonnage doit être activé :

G9(X... , Y... , ...)

G9(ASHAPE)

G9(SHAPE... , PUN)

Particularités et restrictions :

- Le grignotage doit être libéré via MP 8001 00010.

Fonctions CN avec syntaxe de langage standard

- La programmation de la fonction "NUM" ou "LEN" est impérative. Ainsi, la commande génère automatiquement des segments de parcours à partir de la course de déplacement programmée aux extrémités de laquelle une course est déclenchée.
- La fonction a un effet modal, même sur des mouvements de déplacement programmés dans le même bloc.
- Si la fonction Poinçonnage est activée (voir [chap. 6.90 "Poinçonnage Punch, PUN" à la page 366](#)), elle désélectionne la fonction "Grignotage".
- Les blocs qui ne contiennent pas de coordonnées d'axes du plan actif ne déclenchent pas de course.
- Dans la mesure où l'API neutralise le déclenchement de la course, le traitement s'arrête sur la position de déclenchement de la course jusqu'à ce que l'API libère à nouveau le déclenchement de la course.

6.77 Répartition du bloc de déplacement : nombre de segments de parcours "NUM"

Description : Répartit un bloc de déplacement programmé en un nombre défini de segments de parcours de même longueur.



Agit uniquement en combinaison avec les fonctions "Poinçonnage" (voir [chap. 6.90 "Poinçonnage Punch, PUN" à la page 366](#)) et "Grignotage" (voir [chap. 6.76 "Grignotage Nibble, NIB" à la page 332](#)).

Syntaxe :

NUM=<valeur>	
avec	
<valeur>	Nombre de segments de parcours. Valeur entrée : Nombre entier, supérieure à 0. NUM=1 n'entraîne aucune répartition.

Tab. 6-96: Syntaxe NUM

Exemple : Voir l'exemple au [chap. 6.65 "Répartition du bloc de déplacement : longueur du segment de parcours LEN" à la page 315](#).

- Particularités et restrictions :**
- Pour les blocs circulaires divisés, le mouvement s'effectue toujours linéairement d'une course à l'autre !
 - NUM a un effet uniquement dans le bloc programmé et masque une fonction LEN active (voir [chap. 6.65 "Répartition du bloc de déplacement : longueur du segment de parcours LEN" à la page 315](#)).
 - La programmation de NUM n'est possible que si le poinçonnage/grignotage est libéré (MP 8001 00010) et actif.

6.78 Programmation de l'orientation "O()", ROTAX()", "phi, thème, psi"

6.78.1 Description générale

Orienté un outil (fraise, perceuse, laser, pince) librement dans l'espace par rapport au système actuel des coordonnées du programme (PCS).



Si le TCP doit également être déplacé en plus du mouvement d'orientation de l'outil, voir la fonction complémentaire "Programmation des coordonnées tridimensionnelles" dans la documentation "Description des fonctions IndraMotion MTX".

Si un mouvement d'orientation de l'outil sans mouvement simultané du TCP est programmé, l'avance active (F) n'agit que sur le mouvement d'orientation. Le cas échéant, des mouvements de pseudo-coordonnées programmés sont également exécutés (mouvements de pseudo-coordonnées : mouvements provoqués par la programmation des positions d'axe directes). En alternative, il est possible d'effectuer une programmation "Oméga" si G94 est active.

Le vecteur d'orientation longe l'axe de symétrie de l'outil et est orienté vers le logement de l'outil. Il existe différents types de programmation d'orientation.

6.78.2 Mouvement d'orientation linéaire avec programmation des axes

Description :



S'applique uniquement à la transformation d'axe de type 3032101.

Pour programmer l'orientation de l'outil, les valeurs de position de deux axes ronds agissant sur l'outil sont programmées (par ex. B, C).

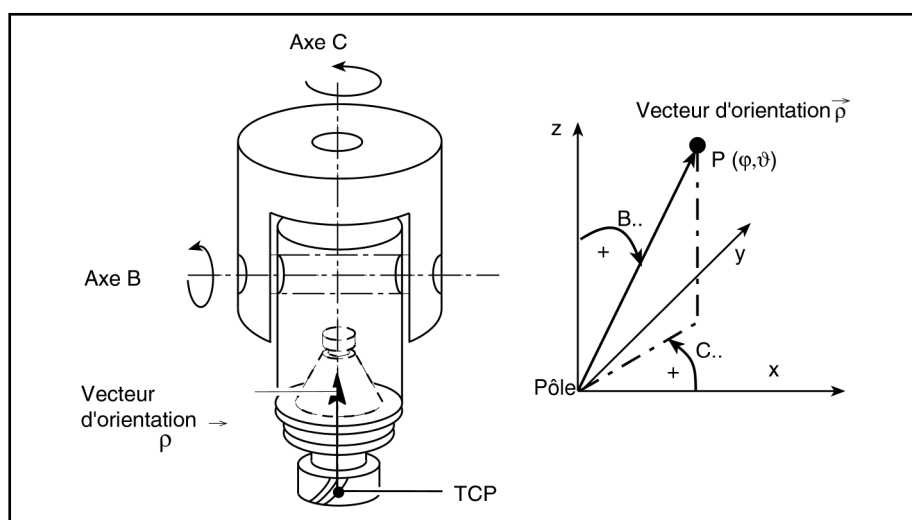


Fig. 6-18: Mouvement d'orientation linéaire avec programmation des axes

Les points suivants s'appliquent ici :

- Seules les cinématiques d'axes dont les valeurs de position des deux axes ronds peuvent être représentées à l'identique sur les coordonnées d'orientation du vecteur d'orientation (par ex. phi et theta) sont admises.
- Le mouvement d'orientation est exécuté sous la forme d'un mouvement linéaire dans les axes ronds.
- Fonction adaptée aux outils à symétrie de révolution.

Syntaxe :

1. Activer le type de transformation d'axe **3032101** avec "Coord(...)" (voir [chap. 6.28 "Sélection d'une transformation d'axe Coord, CRD" à la page 274](#)).
2. Utiliser la syntaxe indiquée ci-dessous.

{ <no> } { <mouv TCP> } { <pos> } { <C><pos> }
avec

Fonctions CN avec syntaxe de langage standard

<no>	Numéro de bloc Voir chap. 3.8.2 "Numéros de blocs" à la page 40.
<mouv TCP>	Mouvement TCP supplémentaire Pour la syntaxe, voir la documentation "Description des fonctions IndraMotion MTX", sous "Programmation des coordonnées tridimensionnelles".
,<C>	Adresses d'axe des axes ronds se trouvant sur l'outil.
<pos>	Position absolue de l'axe en degrés

Tab. 6-97: Syntaxe "Mouvement d'orientation linéaire avec programmation des axes"

Exemple : Exemple de paramétrage :
Paramètre TRA/AxTrafo/TrafoType (MP 1030 00110) pour la transformation d'axe 2 : 3032101
Paramètre CHAN/Ch/Coord/Wcs/LinCoord [1] (MP 7080 00010 [1]) : x
Paramètre CHAN/Ch/Coord/Wcs/LinCoord [2] (MP 7080 00010 [2]) : y
Paramètre CHAN/Ch/Coord/Wcs/LinCoord [3] (MP 7080 00010 [3]) : z

N10 G1 X0 Y0 Z0 B0 C0	Programmation des noms d'axes logiques/physiques
N20 Coord(2)	Activer la transformation d'axe 2 Les éléments suivants sont maintenant programmables : les coordonnées linéaires x, y, z ; les coordonnées d'orientation B et C.
N30 x100 y200 z300 B20 C60	Interpolation des coordonnées linéaires avec mouvement d'orientation supplémentaire
N40 G2 x.. y.. z.. I.. J.. B20 C60	Mouvement hélicoïdal TCP avec mouvement d'orientation supplémentaire
N50 G1 B20 C10	Mouvement d'orientation pur TCP reste constant.
N60 Coord(0) :	Transformation d'axe DÉSACTIVÉE

- Particularités et restrictions :**
- Le type de transformation d'axe actif 3032101 est nécessaire.
 - Une logique de recherche spéciale permet d'éviter les rotations de plus de 180 degrés pour les axes ronds.

6.78.3 Orientation linéaire avec programmation de coordonnées

Description :  S'applique uniquement à la transformation d'axe de type 3232101.

En alternative, l'orientation de l'outil peut également être programmée comme suit :

- Coordonnées d'orientation du vecteur d'orientation (par ex. phi et theta).

Fonctions CN avec syntaxe de langage standard

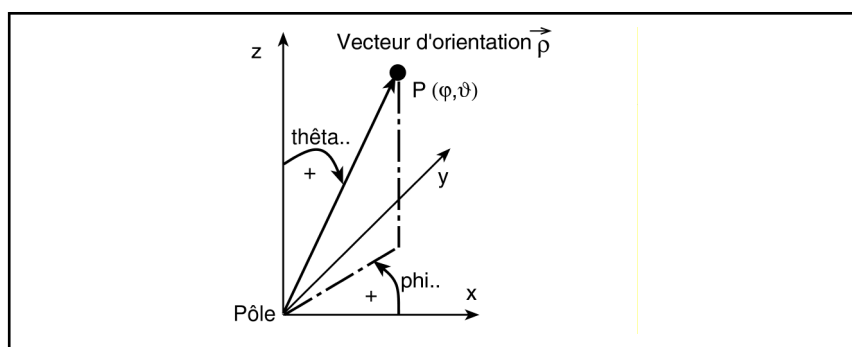


Fig. 6-19: Coordonnées d'orientation du vecteur d'orientation

- Fonction O(..) avec angles polaires (φ , ϑ) ou composantes cartésiennes (ρ_x , ρ_y , ρ_z) du vecteur d'orientation.

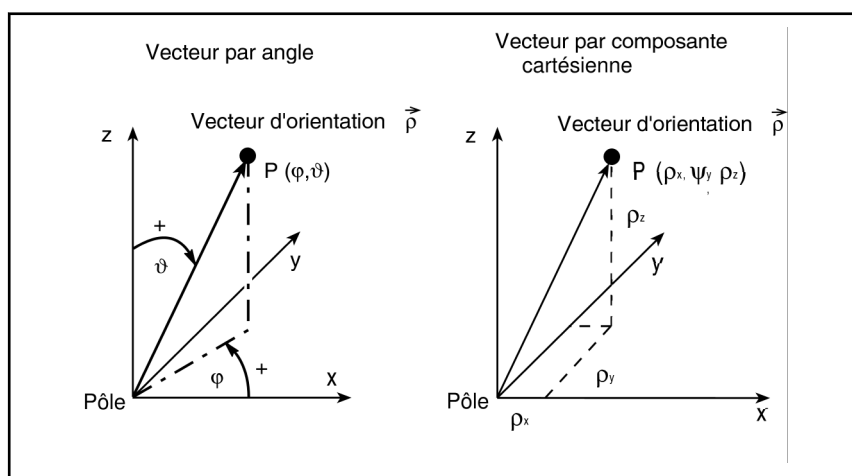


Fig. 6-20: Fonction O(..) avec angles polaires (φ , ϑ) ou composantes cartésiennes (ρ_x , ρ_y , ρ_z) du vecteur d'orientation

- La cinématique d'axe servant de base peut être choisie librement.
- Fonction adaptée aux outils à symétrie de révolution.
- Le mouvement d'orientation se produit en tant qu'interpolation linéaire dans φ et ϑ , c.-à-d. en tant que droite dans un plan φ/ϑ imaginaire.

- Syntaxe :**
1. Activer le type de transformation d'axe **3232101** avec "Coord(...)" (voir [chap. 6.28 "Sélection d'une transformation d'axe Coord, CRD" à la page 274](#)).
 2. Utiliser la syntaxe indiquée ci-dessous.

$\{ \langle \text{no} \rangle \} \{ \langle \text{mouv TCP} \rangle \} \{ \langle \text{phi} \rangle \langle \varphi \rangle \} \{ \langle \text{thêta} \rangle \langle \vartheta \rangle \}$ ou $\{ \langle \text{no} \rangle \} \{ \langle \text{mouv TCP} \rangle \} \circ (\langle \varphi \rangle, \langle \vartheta \rangle)$ ou $\{ \langle \text{no} \rangle \} \{ \langle \text{mouv TCP} \rangle \} \{ \circ (\langle \rho_x \rangle, \langle \rho_y \rangle, \langle \rho_z \rangle)$	
avec	
$\langle \text{no} \rangle$	Numéro de bloc Voir chap. 3.8.2 "Numéros de blocs" à la page 40
$\langle \text{mouv TCP} \rangle$	Mouvement TCP supplémentaire Pour la syntaxe, voir la documentation "Description des fonctions IndraMotion MTX", sous "Programmation des coordonnées tridimensionnelles".

Fonctions CN avec syntaxe de langage standard

<phi>	Nom d'angle saisi dans le paramètre CHAN/Ch/Coord/Wcs/OriCoord[1] (MP 7080 00010 [4]). Valeur par défaut : phi
<thêta>	Nom d'angle saisi dans le paramètre CHAN/Ch/Coord/Wcs/OriCoord [2] (MP 7080 00010 [5]). Valeur par défaut : thêta
<φ>,<θ>	Valeurs absolues des angles en degrés Plage de valeurs : $0^{\circ} \leq \varphi < 360^{\circ}$; $0^{\circ} \leq \vartheta \leq 180^{\circ}$
O (<φ>,<θ>)	Orientation à l'aide de la fonction "O(..)" et des angles polaires <φ> et <θ> du vecteur d'orientation.
O (<ρ _{xy} >,<ρ _z >)	Orientation à l'aide de la fonction "O(..)" et des composantes cartésiennes <ρ _{xyz} > du vecteur d'orientation en dimensions absolues. Standardisation automatique à 1. C'est pourquoi les indications présentées dans les exemples ci-dessous aboutissent à une orientation identique : O(1,2,4), O(2,4,8)

Tab. 6-98: Syntaxe "Mouvement d'orientation linéaire avec programmation des coordonnées"

Exemple : Exemple de paramétrage :

Paramètre TRA/AxTrafo/TrafoType (MP 1030 00110) pour la transformation d'axe 2 : 3232101

Paramètre CHAN/Ch/Coord/Wcs/LinCoord [1] (MP 7080 00010 [1]) : x

Paramètre CHAN/Ch/Coord/Wcs/LinCoord [2] (MP 7080 00010 [2]) : y

Paramètre CHAN/Ch/Coord/Wcs/LinCoord [3] (MP 7080 00010 [3]) : z

Paramètre CHAN/Ch/Coord/Wcs/OriCoord [1] (MP 7080 00010 [4]) : phi

Paramètre CHAN/Ch/Coord/Wcs/OriCoord [2] (MP 7080 00010 [5]) : thêta

N10 G1 X0 Y0 Z0 B0 C0	Programmation des noms d'axes logiques/physiques
N20 Coord(2)	Activer la transformation d'axe 2 Les éléments suivants sont désormais programmables : les coordonnées linéaires x, y, z et les coordonnées d'orientation phi, thêta.
N30 x1 y2 z3 phi5 thêta5	Interpolation des coordonnées linéaires avec mouvement d'orientation supplémentaire
N40 Coord(0)	Transformation d'axe DÉSACTIVÉE

- Particularités et restrictions :
- Le type de transformation d'axe actif 3232101 est nécessaire.
 - Une logique de recherche spéciale permet d'éviter les rotations de plus de 180 degrés pour les axes ronds.
 - La programmation de "ROTAX(... | O(<β>)" est impossible.

6.78.4 Orientation du tenseur

Description : L'orientation de l'outil est influencée par une orientation du système complet des coordonnées de l'outil (TCS) par rapport au PCS actuel.

Fonctions CN avec syntaxe de langage standard

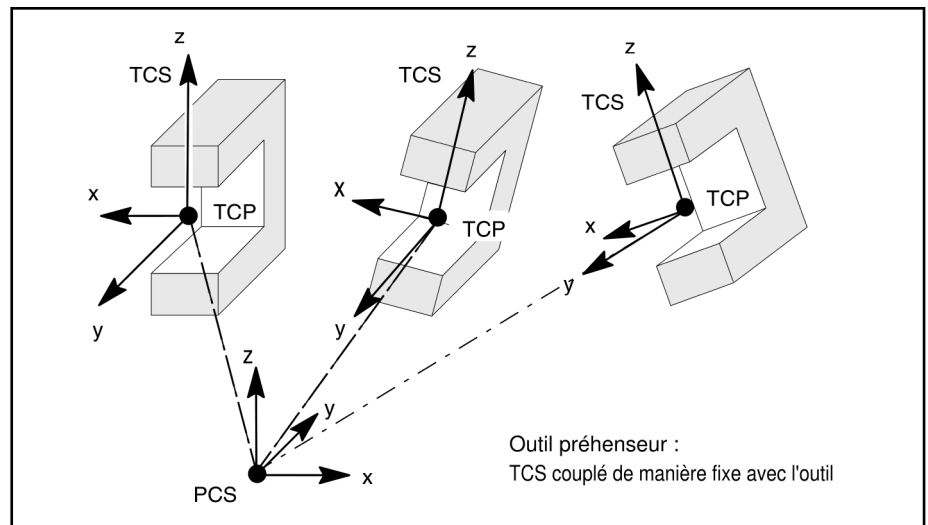


Fig. 6-21: Orientation du tenseur

- *En alternative, la programmation peut être réalisée via :*
 - les angles d'Euler ϕ , θ et ψ .
 Le TCS obtient son orientation en subissant des rotations successives avec les angles d'Euler.
 - 3×3 tenseurs d'orientation (matrice de rotation qui réoriente le TCS complet autour du TCP). Les composantes de ses vecteurs colonnes (O_x , O_y , O_z) définissent exactement l'orientation du TCS et peuvent être programmées en tant qu'angles polaires ou que composantes cartésiennes.
 - la fonction "ROTAX(..)" qui définit un axe de rotation orienté librement dans l'espace et autour duquel il est possible de faire pivoter le tenseur d'orientation selon un angle β défini à l'aide de la fonction "O(..)".
 Dans ce cas, l'axe de rotation peut être programmé avec des angles polaires (ϕ_u , θ_u), ainsi qu'à l'aide des composantes cartésiennes (u_x , u_y , u_z).

Les points suivants s'appliquent ici :

- L'orientation du tenseur peut être programmée uniquement en combinaison avec une transformation d'axe dont la 100e position est de type = 3.
- Le mouvement d'orientation est exécuté en tant que mouvement de rotation du tenseur d'orientation, de l'orientation de départ jusqu'à l'orientation finale programmée.
- La fonction n'est adaptée que pour des outils **sans** symétrie de révolution, puisque le TCS est relié ici de façon fixe à l'outil.

- Syntaxe :**
1. Activer le type de transformation d'axe autorisé (100e position de type = 3) avec "Coord(...)" (voir [chap. 6.28 "Sélection d'une transformation d'axe Coord, CRD" à la page 274](#)).
 2. Utiliser la syntaxe indiquée ci-dessous.

Fonctions CN avec syntaxe de langage standard

{<no>} {<TCP>} {<phi><φ>} {<thêta><θ>} {<psi><ψ>} ou {<no>} {<TCP>} O _x (<φx>, <θx>) O _y (<φy>, <θy>) ou {<no>} {<TCP>} O _x (<φx>, <θx>) O _z (<φz>, <θz>) ou {<no>} {<TCP>} O _y (<φy>, <θy>) O _z (<φz>, <θz>) ou {<no>} {<TCP>} O _x (<O11>, <O21>, <O31>) O _y (<O12>, <O22>, <O32>) ou {<no>} {<TCP>} O _x (<O11>, <O21>, <O31>) O _z (<O13>, <O23>, <O33>) ou {<no>} {<TCP>} O _y (<O12>, <O22>, <O32>) O _z (<O13>, <O23>, <O33>) ou {<no>} {<TCP>} ROTAX (<φu>, <θu>) O (<β>) ou {<no>} {<TCP>} ROTAX (<ux>, <uy>, <uz>) O (<β>)	
avec	
<no>	Numéro de bloc Voir chap. 3.8.2 "Numéros de blocs" à la page 40.
<mouv TCP>	Mouvement TCP supplémentaire Pour la syntaxe, voir la documentation "Description des fonctions IndraMotion MTX", sous "Programmation des coordonnées tridimensionnelles".
<phi>	Nom d'angle saisi dans le paramètre CHAN/Ch/Coord/Wcs/OriCoord [1] (MP 7080 00010 [4]). Valeur par défaut : phi
<thêta>	Nom d'angle saisi dans le paramètre CHAN/Ch/Coord/Wcs/OriCoord [2] (MP 7080 00010 [5]). Valeur par défaut : thêta
<psi>	Nom d'angle saisi dans le paramètre CHAN/Ch/Coord/Wcs/OriCoord [3] (MP 7080 00010 [6]). Valeur par défaut : psi
<φ>,<θ>,<ψ>	Angles d'Euler absolus en degrés La programmation AC/IC est permise. Plage de valeurs : 0° ≤ φ < 360° ; 0° ≤ θ ≤ 180° ; 0° ≤ ψ ≤ 360°. Les valeurs situées en dehors de ces limites sont converties automatiquement dans l'intervalle correspondant.
<φ..>,<θ..>	Valeurs absolues des angles en degrés Plage de valeurs : 0° ≤ φ.. < 360° ; 0° ≤ θ.. ≤ 180°. Si la valeur programmée pour θ.. se situe en dehors de la plage de valeurs, alors elle est automatiquement convertie dans l'intervalle indiqué.

Fonctions CN avec syntaxe de langage standard

$O_x (<O11> , <O21> , <O31>)$ $O_y (<O12> , <O22> , <O32>)$ $O_z (<O13> , <O23> , <O33>)$ $O_x (<\varphi_x> , <\vartheta_x>)$ $O_y (<\varphi_y> , <\vartheta_y>)$ $O_z (<\varphi_z> , <\vartheta_z>)$	Orientation à l'aide de la fonction $O_x(..)$, $O_y(..)$, $O_z(..)$ $O_x(..)$ définit par exemple la direction de la coordonnée x du TCS dans le système de coordonnées de référence. Il en va de même pour $O_y(..)$ et $O_z(..)$. La direction peut être définie soit à l'aide des angles polaires correspondants $<\varphi..>$ et $<\vartheta..>$, soit à l'aide des composantes cartésiennes des vecteurs colonnes du tenseur d'orientation. Seule les cotes absolues sont permises. Les valeurs des composantes des vecteurs colonnes (o..) sont automatiquement standardisées à 1.
$ROTAX (<\varphi_u> , <\vartheta_u>)$	Définition de l'axe de rotation à l'aide des angles polaires (φ_u, ϑ_u) .
$ROTAX (<u_x> , <u_y> , <u_z>)$	Définition de l'angle de rotation à l'aide des composantes cartésiennes $<u_x>$, $<u_y>$, $<u_z>$ en programmation absolue. Standardisation automatique à 1.
$O (<\beta>)$	Indique, en degrés, l'angle incrémental $<\beta>$ duquel le tenseur d'orientation doit se déplacer autour de l'axe de rotation. Les valeurs supérieures à 360 degrés sont permises. Le sens de rotation peut être sélectionné à l'aide des signes (+/-).

Tab. 6-99: Syntaxe d'orientation du tenseur

Exemple : Exemple de paramétrage :

Paramètre TRA/AxTrafo/TrafoType (MP 1030 00110) pour la transformation d'axe 2 : 3333301

Paramètre CHAN/Ch/Coord/Wcs/LinCoord [1] (MP 7080 00010 [1]) : x

Paramètre CHAN/Ch/Coord/Wcs/LinCoord [2] (MP 7080 00010 [2]) : y

Paramètre CHAN/Ch/Coord/Wcs/LinCoord [3] (MP 7080 00010 [3]) : z

Paramètre CHAN/Ch/Coord/Wcs/OriCoord [1] (MP 7080 00010 [4]) : phi

Paramètre CHAN/Ch/Coord/Wcs/OriCoord [2] (MP 7080 00010 [5]) : theta

Paramètre CHAN/Ch/Coord/Wcs/OriCoord [3] (MP 7080 00010 [6]) : psi

N100 G1 X10 Y20 Z30 $O_x(1,0,0)$ $O_y(0,0.707,-0.707)$

- ou -

N100 G1 X10 Y20 Z30 $O_x(1,0,0)$ $O_z(0,0.707,0.707)$

- ou -

N100 G1 X10 Y20 Z30 $O_y(0,0.707,-0.707)$ $O_z(0,0.707,0.707)$

Fonctions CN avec syntaxe de langage standard

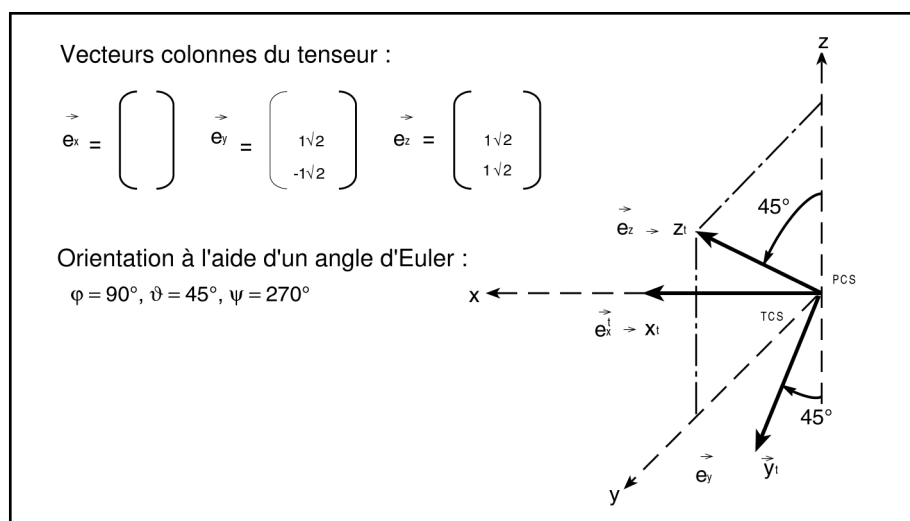


Fig. 6-22: Exemple d'orientation tensorielle

Particularités et restrictions :

- Hormis pour la programmation de ROTAX(..), le mouvement d'orientation est toujours exécuté en suivant la course la plus courte vers l'orientation finale.
Pour ROTAX(..), le sens de rotation dépend du signe (+/-) de $\langle \beta \rangle$.
- Afin de créer un tenseur d'orientation univoque, les restrictions suivantes doivent être observées :
Avec $\vartheta = 0^\circ$:
la somme de φ et ψ est requise pour définir une orientation.
Avec $\vartheta = 180^\circ$:
la différence de φ et ψ est nécessaire pour définir une orientation.
- Le tenseur d'orientation ne peut pas être calculé si deux vecteurs colonnes sont parallèles ou antiparallèles lors de la programmation des vecteurs colonnes du tenseur.
Une erreur d'exécution est signalée.

6.78.5 Orientation vectorielle

Description : En alternative, la programmation de l'orientation de l'outil est possible via :

- les coordonnées d'orientation du vecteur d'orientation (par ex. phi et theta).

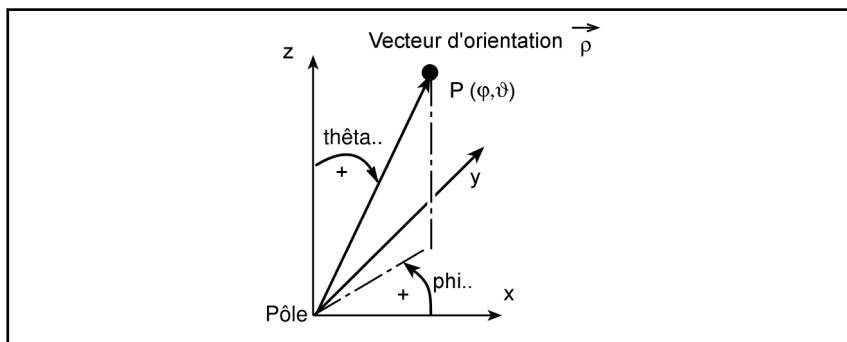


Fig. 6-23: Coordonnées d'orientation du vecteur d'orientation

- la fonction "O(..)" avec les angles polaires (φ, ϑ) ou les composantes cartésiennes (ρ_x, ρ_y, ρ_z) du vecteur d'orientation.

Fonctions CN avec syntaxe de langage standard

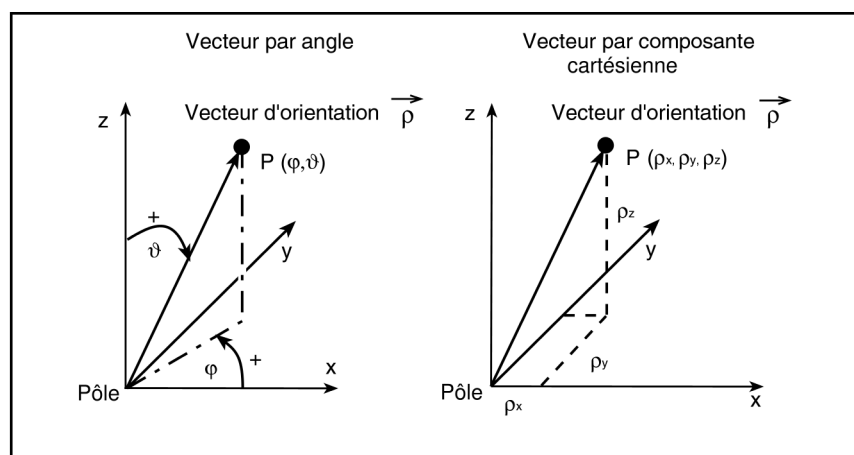


Fig. 6-24: Fonction $O(\dots)$ avec angles polaires (φ , ϑ) ou composantes cartésiennes (ρ_x , ρ_y , ρ_z) du vecteur d'orientation

- la fonction "ROTAX(...)" qui définit un axe de rotation orienté librement dans l'espace et autour duquel il est possible de faire pivoter le vecteur d'orientation d'un angle β défini à l'aide de la fonction "O(...)".

L'axe de rotation peut être programmé ici avec des angles polaires (φ_u , ϑ_u) ou avec des composantes cartésiennes (u_x , u_y , u_z).

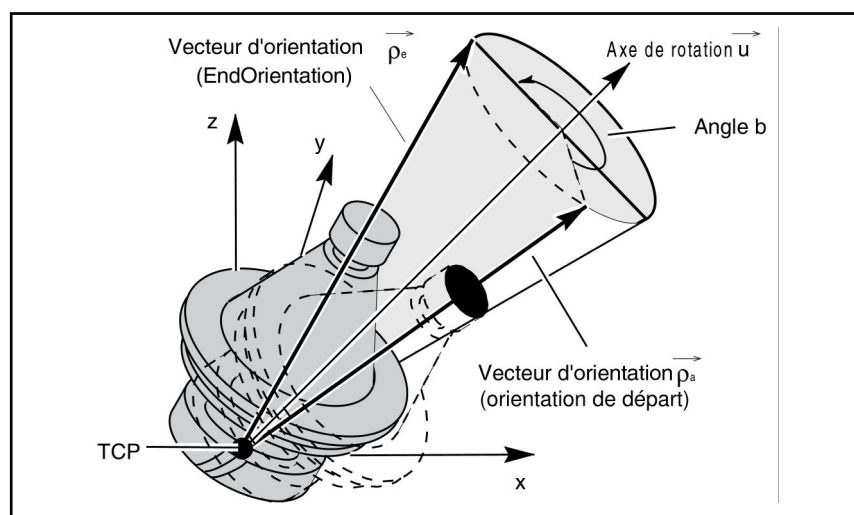


Fig. 6-25: Fonction $ROTAX(\dots)$ pour la définition d'un axe de rotation orienté librement dans l'espace

Les points suivants s'appliquent ici :

- L'orientation vectorielle peut maintenant être programmée avec une transformation d'axe dont la 100e position est de type = 2.
- Le mouvement d'orientation est exécuté du début à la fin de l'orientation programmée sous la forme d'un mouvement de rotation du vecteur d'orientation.
- Fonction adaptée aux outils à symétrie de révolution.

- Syntaxe :**
- Activer le type de transformation d'axe autorisé (100e position de type = 2) avec "Coord(...)" (voir [chap. 6.28 "Sélection d'une transformation d'axe Coord, CRD" à la page 274](#)).
 - Utiliser la syntaxe indiquée ci-dessous.

Fonctions CN avec syntaxe de langage standard

{<no>} {<mouv TCP>} {<phi><φ>} {<thêta><θ>} ou {<no>} {<mouv TCP>} O (<φ>, <θ>) ou {<no>} {<mouv TCP>} O (<px>, <py>, <pz>) ou {<no>} {<mouv TCP>} ROTAX (<φu>, <θu>) O (<β>) ou {<no>} {<mouv TCP>} ROTAX (<ux>, <uy>, <uz>) O (<β>)	
avec	
<no>	Numéro de bloc Voir chap. 3.8.2 "Numéros de blocs" à la page 40.
<mouv TCP>	Mouvement TCP supplémentaire Pour la syntaxe, voir la documentation "Description des fonctions IndraMotion MTX", sous "Programmation des coordonnées tridimensionnelles".
<phi>	Nom d'angle saisi dans MP 7080 00010[4]. Valeur par défaut : phi
<thêta>	Nom d'angle saisi dans MP 7080 00010[5]. Valeur par défaut : thêta
<φ>, <θ>, <φu>, <θu>	Valeurs absolues des angles en degrés. Plage de valeurs : $0^\circ \leq \varphi < 360^\circ$; $0^\circ \leq \theta \leq 180^\circ$. Si une autre valeur est programmée pour θ, elle est automatiquement convertie dans l'intervalle indiqué.
O (<φ>, <θ>)	Orientation à l'aide de la fonction O(..) et des angles polaires <φ> et <θ> du vecteur d'orientation.
O (<px>, <py>, <pz>)	Orientation à l'aide de la fonction O(..) et des composantes cartésiennes <px>, <py>, <pz> du vecteur d'orientation via les données absolues. Standardisation automatique à 1. C'est pourquoi les indications présentées dans les exemples ci-dessous aboutissent à une orientation identique : O(1,2,4), O(2,4,8)
ROTAX (<φu>, <θu>)	Définition de l'axe de rotation à l'aide des angles polaires (φ _u , θ _u).
ROTAX (<ux>, <uy>, <uz>)	Définition de l'angle de rotation à l'aide des composantes cartésiennes <ux>, <uy>, <uz> en cotes absolues. Standardisation automatique à 1.
O (<β>)	Indique, en degrés, l'angle incrémental <β> duquel le vecteur d'orientation doit se déplacer autour de l'axe de rotation. Les valeurs supérieures à 360 degrés sont permises. Le sens de rotation peut être sélectionné à l'aide des signes (+/-).

Tab. 6-100: Syntaxe de l'orientation vectorielle

Particularités et restrictions : L'orientation de départ/d'arrivée du vecteur d'orientation ne doit pas suivre un tracé parallèle ou antiparallèle, sauf pour la programmation de ROTAX(..).

6.79 Avance 100 % "OvrDis, OVD", "OvrEna, OVE"

Description : Influence l'effet du potentiomètre d'avance du canal (qCh_Override_xy) pour l'avance et l'avance rapide via le programme.

Ces fonctions agissent en modes "Saisie manuelle" et "Exécution".

Les axes asynchrones ne sont pas affectés par la fonction "OVD / OVE".

Fonctions CN avec syntaxe de langage standard

Syntaxe :	OvrDis	Potentiomètre d'avance DÉSACTIVÉ L'avance est définie sur 100 % de la valeur programmée, indépendamment du potentiomètre d'avance.
	Format abrégé : OVD	
	OvrEna	Potentiomètre d'avance ACTIVÉ L'avance dépend de la position du potentiomètre d'avance.
	Format abrégé : OVE	

Tab. 6-101: Syntaxe OvrDis (OVD) et OvrEna (OVE)

Les points suivants s'appliquent ici :

- Ces deux fonctions sont modales et se révoquent réciproquement.
- L'état de mise en service peut être défini à l'aide des paramètres machine.
- Les deux fonctions peuvent également être programmées dans le même bloc que d'autres conditions de course.
- Le signal d'interface "qAx_Override_100" permet d'influencer l'effet de l'Override sur les axes asynchrones de manière spécifique à l'axe.

Exemple :	:	Le potentiomètre d'avance est réglé sur 100 %.
	N40 OVD G1 X5 Z-2 F200 S100 M4	
		Désactiver le potentiomètre d'avance
	:	Passer le potentiomètre d'avance à 40 % ; l'avance est maintenue à 100 %.
	N80 OVE X100 Activer le potentiomètre d'avance	
	Y50	L'avance passe à 40 %.

6.80 Modification de l'accélération maximale sur trajectoire "PathAcc, PAC"

Description : Dans le programme pièce, réduit la limite supérieure pour :

- l'accélération sur trajectoire,
- l'accélération de freinage sur trajectoire.

Les deux valeurs d'accélération sont préréglées à l'aide des paramètres machine et peuvent être commutées dans le programme pièce de manière séparée ou couplée.



L'accélération sur trajectoire programmée ou préréglée peut être limitée par les accélérations sur trajectoire maximales des axes participants à la trajectoire.

Syntaxe :	PathAcc (ACC<valeur>)	Paramétrage commun de l'accélération sur trajectoire et de l'accélération de freinage sur trajectoire.
	PathAcc ({UP<valeur1>, } {DOWN<valeur2>})	Réglage séparé de l'accélération sur trajectoire et de l'accélération de freinage sur trajectoire.
	PathAcc () ou PathAcc (0)	Réactiver les valeurs d'accélération de MP 7030 00210 et 7030 00220.

Fonctions CN avec syntaxe de langage standard

Format abrégé : PAC (. .)	
avec	
<valeur>	Valeur d'accélération En fonction de l'unité de mesure (G71/G70), la commande interprète la valeur programmée en "1000 pouces/s ² " ou "m/s ² ".
<valeur1>	Valeur pour l'accélération sur trajectoire Identique à <valeur> pour le reste
<valeur2>	Valeur de l'accélération de freinage sur trajectoire. Identique à <valeur> pour le reste

Tab. 6-102: Syntaxe PathAcc (PAC)

Exemple :	
N30 G71	Activer la programmation métrique
N40 PAC (UP1.5)	Régler l'accélération sur trajectoire sur 1,5 m/s ²
:	
N140 PAC (ACC5)	Régler l'accélération sur trajectoire et l'accélération de freinage sur trajectoire à 5 m/s ²
:	
N200 PAC (UP3.5, DOWN2)	Régler l'accélération sur trajectoire sur 3,5 m/s ² et l'accélération de freinage sur trajectoire sur 2 m/s ²
:	
N240 PAC ()	Rétablir les valeurs d'accélération réglées dans les paramètres machine.
:	

- Particularités et restrictions :**
- Les valeurs d'accélération maximales programmables sont toujours limitées par les valeurs des paramètres machine.
 - Nous conseillons de programmer cette fonction dans un bloc à part.

6.81 Activation des tableaux de placement "PmTsel, PMS"

Description : Active les tableaux de placement (tables de correction du placement "Plan incliné"). Ces tables de correction sont archivées sous forme de fichiers XML dans le système de fichiers de la commande.

Syntaxe :	PmTsel ({<chemin>}<nom de fichier>)
	Format abrégé : PMS (. .)
	avec

Fonctions CN avec syntaxe de langage standard

<chemin>	Indication optionnelle du chemin d'accès au répertoire dans lequel <nom de fichier> est enregistré. Si aucun chemin n'est indiqué, la recherche s'effectue sous le chemin "/database". Si <nom de fichier> ne s'y trouve pas, la commande utilise le chemin de recherche des sous-programmes afin de rechercher <nom de fichier> dans d'autres répertoires.
<nom de fichier>	Nom du fichier du tableau de placement, extension de fichier incluse Les tableaux ayant un nom standard (PM<numéro>.pmt) peuvent être activés directement avec leur numéro, par exemple PmTsel(3) active le tableau PM3.pmt.

Tab. 6-103: Syntaxe PmTsel (PMS)



Pour de plus amples informations sur la création et l'édition des tableaux de placement, reportez-vous au mode d'emploi de la commande !

6.82 Programmation des coordonnées polaires

6.82.1 Définition du pôle "PolarPol, POP"

- Description :**
- Définit l'origine du système de coordonnées polaires par rapport au plan actif du système actif des coordonnées du programme.
 - Initialise l'angle polaire 1 sur 0 degré pour la programmation des coordonnées polaires (adresse standard : A ; paramétrable dans MP 8005 00001).
 - Initialise l'angle polaire 2 sur 90 degrés pour la programmation des coordonnées polaires (adresse standard : B ; paramétrable dans MP 8005 00002).

Syntaxe :

PolarPol (<valeurCP>,<valeurCS>)	
	Définir le pôle
PolarPol (ACT-POS)	Placer le pôle sur la position actuelle
PolarPol () ou PolarPol (0)	Placer le pôle sur la coordonnée 0,0
Format abrégé : POP (. .)	
avec	
<valeurCP>	Valeur du pôle pour la coordonnée principale
<valeurCS>	Valeur du pôle pour la coordonnée secondaire

Tab. 6-104: Syntaxe PolarPol (POP)

Exemple :

```

:
N10 G18           Commutation au plan ZX avec la coordonnée principale Z
:                et la coordonnée secondaire X.

```

Fonctions CN avec syntaxe de langage standard

N30 G0 X10 Z25	Mouvement de positionnement en avance rapide sur les coordonnées cartésiennes X10 et Z25.
:	
N40 POP (ACTPOS)	Les valeurs de position actuelles sont reprises pour la définition du pôle (valeurCP = 25 ; valeurCS = 10).
:	

- Particularités et restrictions :**
- Toutefois, si la fonction "PolarPol" n'est pas utilisée, la commande numérique utilise l'origine du système actif des coordonnées du programme en tant que pôle.
 - Un pôle programmé reste actif pour le plan actuel jusqu'au changement de plan suivant seulement (voir [chap. 5.2.10 "Changement de planG17, G18, G19" à la page 146](#)).

6.82.2 Programmation des coordonnées polaires

Description : La commande IndraMotion MTX permet de programmer des coordonnées polaires aussi bien dans un plan (2D) que dans l'espace (3D).

Le point d'arrivée voulu est indiqué à l'aide de :

- la valeur du rayon (distance du point cible par rapport au pôle),
- l'angle polaire 1,
- l'angle polaire 2.

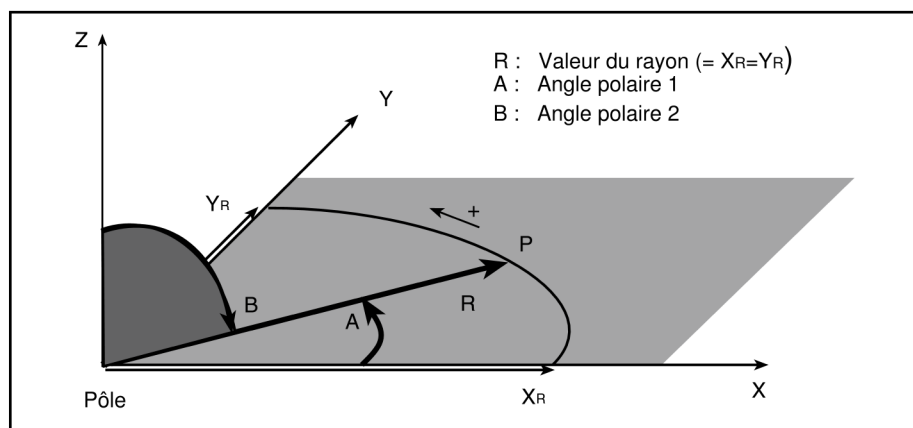


Fig. 6-26: Exemple : Coordonnées polaires sur le plan quand G17 est active

Valeur du rayon :

Elle est toujours programmée à l'aide d'une adresse d'axe du plan de travail actuel (par ex. "X" ou "Y" quand G17 est active). Cette coordonnée programmée est également appelée « Coordonnée de rayon ».

En cas de programmation absolue, elle indique la distance entre le point cible et le pôle ; en cas de programmation incrémentale, elle indique la distance entre le point cible et le point de départ.

Si les deux adresses d'axe possibles sont programmées dans un même bloc, une erreur d'exécution est générée.

Angle polaire 1 :

- Il est normalement programmé en tant qu'adresse "A" et est toujours situé dans le plan de travail actuel.
- L'angle polaire programmé 1 se rapporte toujours au **dernier axe programmé** de la coordonnée de rayon. Une valeur positive signifie une rotation dans le sens mathématique positif. La valeur reste active jusqu'à ce qu'une nouvelle valeur soit programmée.

Fonctions CN avec syntaxe de langage standard

Si une programmation incrémentale est active, l'angle programmé est calculé de façon incrémentale.

- En activant la programmation des coordonnées polaires avec POL, l'angle polaire 1 est défini sur la valeur 0 degré.

Angle polaire 2 :

- Il est normalement programmé en tant qu'adresse "B". Si la programmation absolue est active, il décrit l'angle entre l'axe de la normale (par rapport au plan de travail actuel) et le vecteur du pôle actuel par rapport au point cible. Il reste valide jusqu'à ce qu'une nouvelle valeur soit programmée.

Si une programmation incrémentale est active, l'angle programmé est calculé de façon incrémentale.

- En activant la programmation des coordonnées polaires avec POL, l'angle polaire 2 est défini sur la valeur 90 degrés.

Dans ce cas, le point cible se trouve exactement dans le plan de travail actuel. Les coordonnées polaires n'agissent maintenant qu'au niveau du plan. Toutes les autres coordonnées en dehors du plan de travail sont interprétées de façon cartésienne. Ceci s'applique notamment aux valeurs de coordonnées programmées pour l'axe actuel de la normale !

- Des "vraies" coordonnées polaires tridimensionnelles (coordonnées sphériques) sont créées, si l'angle polaire 2 devient différent de 90 degrés. La programmation des valeurs de coordonnées pour l'axe actuel de la normale génère une erreur d'exécution. Toutes les coordonnées en dehors de l'espace 3D, par contre, peuvent être programmées simultanément. Comme en 2D, elles sont interprétées ici aussi de façon cartésienne.
- Si l'angle polaire 2 est de 0 degré, la valeur de l'angle polaire 1 n'a aucune importance, puisque le point programmé est situé exactement à l'aplomb du pôle.

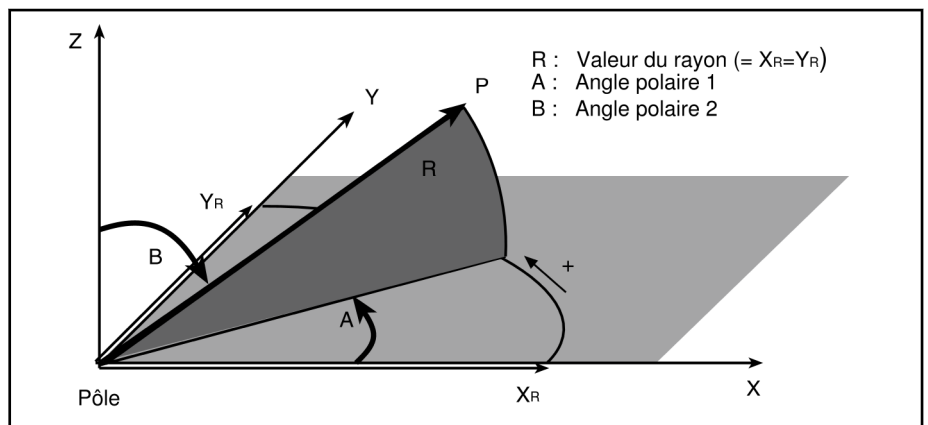


Fig. 6-27: Exemple : Coordonnées polaires dans l'espace quand G17 est active



Les adresses standards pour les angles polaires 1 et 2 peuvent être modifiées via les paramètres machine 8005 0001 et 8005 0002. C'est pourquoi elles peuvent être différentes sur votre installation.

Exemples : **Programmation des coordonnées polaires en valeur absolue**

Fonctions CN avec syntaxe de langage standard

N10 G18	Changement de plan
:	Ici : Commutation au plan ZX avec la coordonnée principale Z et la coordonnée secondaire X.
N40 POP (25,10)	Définition du pôle dans le plan actuel (ici : avec les coordonnées cartésiennes Z = 25 et X = 10).
:	
N50 G1 (POL)	Mouvement sur trajectoire en coordonnées polaires selon P2 (voir fig.).
Z20 A70	
:	P2 est défini par la distance absolue par rapport au pôle (ici : 20 mm) et par l'angle polaire 1 (ici : A = 70 degrés), en fonction de l'axe de rayon programmé (ici : l'axe Z).

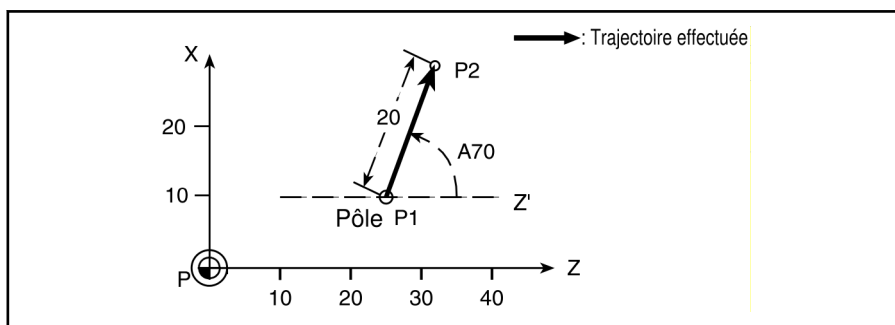


Fig. 6-28: Mouvement sur trajectoire en coordonnées polaires selon P2

N60 G1 (POL)	Mouvement sur trajectoire en coordonnées polaires selon P3 (voir fig.).
X20 A90	
:	P3 est défini par la distance absolue par rapport au pôle (ici : 20 mm) et par l'angle polaire 1 (ici : A = 90 degrés), en fonction de l'axe de rayon programmé (ici : l'axe X).
:	
:	
:	La programmation suivante est également possible :
N60 G1 (POL) Z20 A180	
:	
- ou -	
N60 Z20 A180	

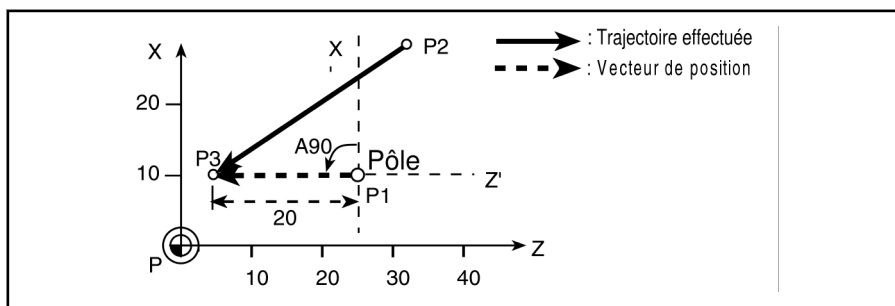


Fig. 6-29: Mouvement sur trajectoire en coordonnées polaires selon P3

Programmation incrémentale des coordonnées polaires

N10 G17 G90 G0 X0 Y0	Position de départ : P0
N20 G1 (POL) X200 A0	P1
N30 G91 Y200 A30	P2
N40 A60	P3
N50 A60	P4

Fonctions CN avec syntaxe de langage standard

```

N60 A60          P5
N70 A60          P6
N80 A60          P7
N90 X0           P0
N100 M30
    
```

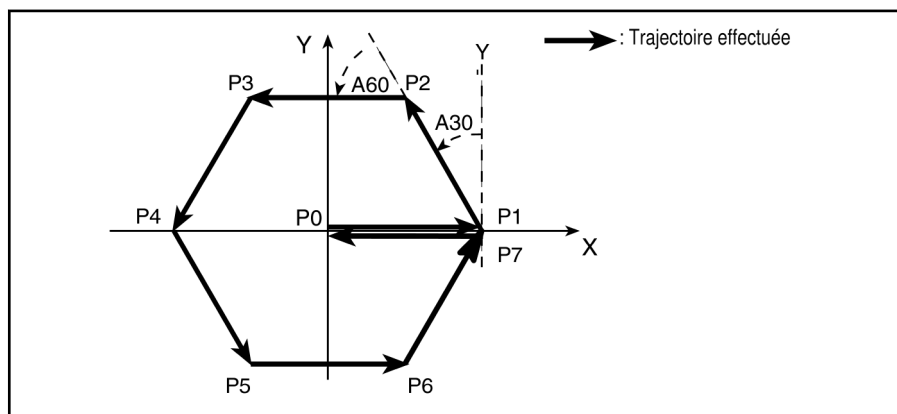


Fig. 6-30: Programmation incrémentale des coordonnées polaires

6.83 Définition du point d'inversion miroir/de rotation "PoleSet, PLS"

Description : Cette fonction définit la position absolue du pôle pour les aides à la saisie des fonctions Miroir, Mise à l'échelle et Rotation. Le pôle défini est alors le point de repère de ces aides à la saisie et représente donc le point d'inversion miroir/de rotation.



La fonction n'est pas requise si la fonction miroir/rotation doit être effectuée par rapport à l'origine du programme.

Les points suivants s'appliquent ici :

- PoleSet est une fonction modale. La position du pôle se rapporte au système actuel des coordonnées du programme et est maintenue jusqu'à ce qu'elle soit redéfinie ou réinitialisée sur l'origine des coordonnées du programme.
- La fonction n'entraîne aucun déplacement des axes.
- La fonction peut être programmée dans le même bloc que d'autres conditions de course et fonctions auxiliaires.
- Le pôle programmé se rapporte aux fonctions Miroir (Mirror), Mise à l'échelle (Scale) et Rotation (Rotate).

Syntaxe :

PoleSet (<coor-données>)	Placer le pôle sur les coordonnées indiquées. Les coordonnées indiquées doivent être séparées par des virgules (par ex. : PoleSet (X5, Y2)).
PoleSet (0) ou PoleSet ()	Réinitialiser le pôle sur l'origine du système de coordonnées du programme.
Format abrégé : PLS (..)	

Tab. 6-105: Syntaxe PoleSet (PLS)

Fonctions CN avec syntaxe de langage standard

Exemple :

```
:  
  
N30 PLS (X5, Y2)  Placer le point d'inversion miroir/de rotation sur la position  
:                X = 5 et Y = 2.  
  
N130 PLS ()      Réinitialiser le point d'inversion miroir/de rotation sur l'origine  
:                du système de coordonnées du programme.
```

Particularités et restrictions : La position du pôle doit être indiquée en valeur absolue, c.-à-d. par rapport à l'origine actuelle du programme.

6.84

Sortie à grande vitesse dépendant de la position "PosDepH-
SOut, PHS"

Description : Influence le statut d'une sortie à grande vitesse (sortie HS, HighSpeed) à l'aide d'un mouvement de déplacement programmé.

La sortie HS est définie ou réinitialisée selon la programmation dès que la **position de consigne CN** relative au début ou à la fin du bloc actuel a atteint une valeur programmable.

Il est possible de programmer pour la durée pendant laquelle la sortie doit rester initialisée après l'apparition de l'événement spécifique.



Paramétrage des sorties à grande vitesse disponibles via MP 4075 00102.

Syntaxe :	PosDepHSOut (<mode>{, {<distance>}{, <durée>}})	
	Format abrégé : PHS (. .)	
	avec	
	<mode>	Mode d'action voulu de la fonction Valeurs entrées : 0, 1 ou -1 0 : Enregistre de façon modale les données programmées dans <distance> et/ou <durée>. Cela ne modifie en rien l'état actuel de la sortie à grande vitesse. 1 : Enregistre de façon modale les données programmées dans <distance> et/ou <durée> et paramètre la sortie à grande vitesse. Pour les paramètres non programmés, les valeurs appliquées sont celles des données modales pertinentes. -1 : Réinitialise la sortie à grande distance. Une <distance> éventuellement programmée n'a qu'un effet local sur ce bloc. Une <durée> éventuellement programmée est ignorée.

Fonctions CN avec syntaxe de langage standard

<distance>	Distance par rapport au début/à la fin du bloc qui doit déclencher l'initialisation du signal lorsqu'elle est atteinte (en mm ou pouces). Valeur par défaut : 0 0 : à la fin du bloc Supérieure à 0 : distance par rapport au début du bloc Inférieure à 0 : distance par rapport à la fin du bloc
<durée>	Durée de fonctionnement maximale de la sortie à grande vitesse (en ms) Plage de valeurs : 0.5 à 10 000.0 Les valeurs programmées sont arrondies en interne au multiple entier supérieur du temps de cycle CN le plus proche.

Tab. 6-106: Syntaxe PosDepHSOut (PHS)

Exemple :

```

:
N05 G71
N10 PHS(0,-1.2,40)      Configurer la fonction
:
:                        Lors d'un appel ultérieur via la syntaxe PHS(1), la
:                        sortie à grande vitesse est définie à environ 1,2 mm
                        avant d'atteindre le point d'arrivée pour une durée
                        d'environ 40 ms.

N210 G1 G91 F1000
N220 X10 Y23 PHS(1)      Régler la sortie à grande vitesse conformément à la
:                        configuration dans le bloc N10
:
N330 X10 PHS(1,0.1,900)  Définir la sortie à grande vitesse à 0,1 mm après la
:                        position de départ pour une durée maximale de
                        900 ms
N350 X20 PHS(-1,-0.3)    Réinitialiser la sortie à grande vitesse à une dis-
:                        tance de 0,3 mm avant la position finale
    
```

Particularités et restrictions :

- Une sortie à grande vitesse est prise en charge par chaque canal.
- La fonction n'agit que sur un mouvement de déplacement programmé dans le même bloc.
- Une sortie à grande vitesse définie n'est réinitialisée qu'une fois la durée de fonctionnement écoulée. La remise à zéro du système n'affecte pas cet instant.
- <distance> se rapporte toujours à la position de consigne actuelle du point de vue de la commande numérique.
Les temporisations conditionnées par le système (par ex. provoquées par des fonctions de l'entraînement ou par la poursuite de l'axe) restent inconsiderées ici.
- Si un nouvel ordre PHS est reçu pendant la durée de mise en service, la commande numérique supprime l'ordre encore actif et exécute le nouvel ordre.



Si deux ordres PHS sont reçus consécutivement sans changement de signal sur la sortie à grande vitesse, alors le matériel informatique externe ne peut généralement pas distinguer le nouvel ordre.

Fonctions CN avec syntaxe de langage standard

6.85 Mode de positionnement pour les axes sans fin "PosMode, PMD", mode de positionnement local pour les axes sans fin "DC", "ACP", "ACN"

Description : Définit la direction dans laquelle les axes de type "Sans fin" doivent tourner lors d'un processus de positionnement.

Les points suivants s'appliquent ici :

- Contrairement à "PosMode", DC(...), ACP(...) et ACN(...) agissent bloc par bloc et masquent le mode de positionnement actif dans le bloc actuel pour l'axe programmé concerné.
- Les données de position programmées en relation avec les fonctions DC, ACP ou ACN sont toujours interprétées en tant que valeurs de position absolues.

Syntaxe :

PosMode (<adr><mode>)	Activer le mode de positionnement conforme à <mode> pour l'axe ayant l'adresse <adr>. Il est possible de programmer plusieurs axes entre les parenthèses. Activer le mode de positionnement défini dans MP 1003 00005 pour les axes ronds/sans fin non programmés sur le canal actuel.
PosMode () ou PosMode (0)	Activer le mode de positionnement conforme à MP 1003 00005 pour tous les axes sur le canal actuel.
Format abrégé : PMD (. .)	
<adr>=DC (<valeur>)	L'axe <adr> approche la position absolue <valeur> en suivant la course la plus courte.
<adr>=ACP (<valeur>)	L'axe <adr> approche la position absolue <valeur> dans le sens mathématique positif (voir remarque ci-dessous).
<adr>=ACN (<valeur>)	L'axe <adr> approche la position absolue <valeur> dans le sens mathématique négatif.
avec	
<adr>	Adresse de l'axe Les axes de type "Sans fin" sont permis.

Fonctions CN avec syntaxe de langage standard

<valeur>	Information sur la course pour <adr>.
<mode>	Mode de positionnement 0 : Aucune Logique de positionnement La direction de déplacement est toujours calculée à partir de la différence entre l'ancienne et la nouvelle position. 1 : Course la plus courte La course de déplacement maximale de l'axe ne dépasse pas la moitié de la valeur modulo pertinente. 2 : Selon le signe (+/-) programmé : "+" : rotation à droite ; "-" : rotation à gauche. 3 : Les commutateurs de fin de course logiciels actifs (1020 00001/3 et 1020 00002/4) déterminent la zone de déplacement autorisée et la direction possible. La zone de déplacement autorisée s'étend du commutateur de fin de course négatif au commutateur de fin de course positif. Les zones des commutateurs de fin de course ne deviennent actives qu'après la prise d'origine de l'axe. Les coordonnées peuvent être indiquées aussi bien dans la plage modulo positive que négative (par ex. 20° correspond à -340°). Le signe des coordonnées ainsi que l'attribut local des coordonnées "DC" sont ignorés. Les attributs "ACP", "ACN" et "IC", par contre, sont observés et génèrent un message d'erreur en cas de dépassement de la zone de travail.

Tab. 6-107: Syntaxe PosMode (PMD), DC, ACP et ACN



Sens mathématique positif :

Sens de rotation trigonométrique (inverse au sens des aiguilles d'une montre) vu d'un axe de coordonnées en direction de l'origine des coordonnées.

Exemple :

N40	Indépendamment de "PosMode", l'axe physique désigné "B"
B=ACP (-258)	s'approche de la position 258 degrés dans le sens mathématique positif. Le signe est ignoré.
:	
N80 PMD (A1, C2)	Mode de positionnement de :
:	Axe A : course la plus courte
:	Axe B : selon MP 1003 00005
:	Axe C : signe (+/-)

Particularités et restrictions :

- Exception faite du mode de positionnement "Zone de déplacement limitée", les commutateurs de fin de course logiciels pour axes sans fin doivent toujours être masqués.
- La commutation du mode de positionnement pour un axe sans fin n'est possible que si la commutation est libérée dans le paramètre MP 1003 00050.
- Il est possible de programmer plusieurs fonctions DC, ACP ou ACN (chacune pour des axes différents) dans un même bloc.

Fonctions CN avec syntaxe de langage standard

- Les fonctions DC, ACP ou ACN ne sont efficaces que pour les axes de type "Rond" ou "Sans fin". Si elles sont utilisées en relation avec d'autres types d'axe, la commande les ignore.
- Les signes négatifs des données de position des fonctions DC, ACP ou ACN sont ignorés.
- Les points suivants s'appliquent au mode de positionnement "Course limitée" :

Les angles finaux négatifs sont toujours indiqués dans la plage modulo positive (0 à 360°).

L'attribut de coordonnée DC "Course la plus courte" est ignoré. Au lieu de cela, la commande choisit la "Course autorisée" avec ajustement automatique de la direction.

Les positionnements de direction par signe, attributs de coordonnée ou incréments (+/-, ACP/ACN et G91/IC) ne sont pas ajustés et ne sont admis qu'au sein de la zone du commutateur de fin de course.

Les mouvements ne sont admis qu'au sein d'une rotation complète.

6.86 Programmation de précision "PrecProg, PRP"

Description : Réduit l'avance au niveau des raccords entre segments de contour et des segments circulaires de la trajectoire jusqu'à ce que la précision définie soit respectée.

La précision exigée peut également être influencée par l'indication :

- de l'**erreur de contour ou de rayon ϵ** maximale admissible au niveau du raccord entre segments de contour ou des arcs de cercles ;
- de la **poursuite de trajectoire δ** maximale admissible (distance des angles) qui ne doit pas être dépassée lors du déplacement sur un raccord entre segments de contour.

Fonctions CN avec syntaxe de langage standard

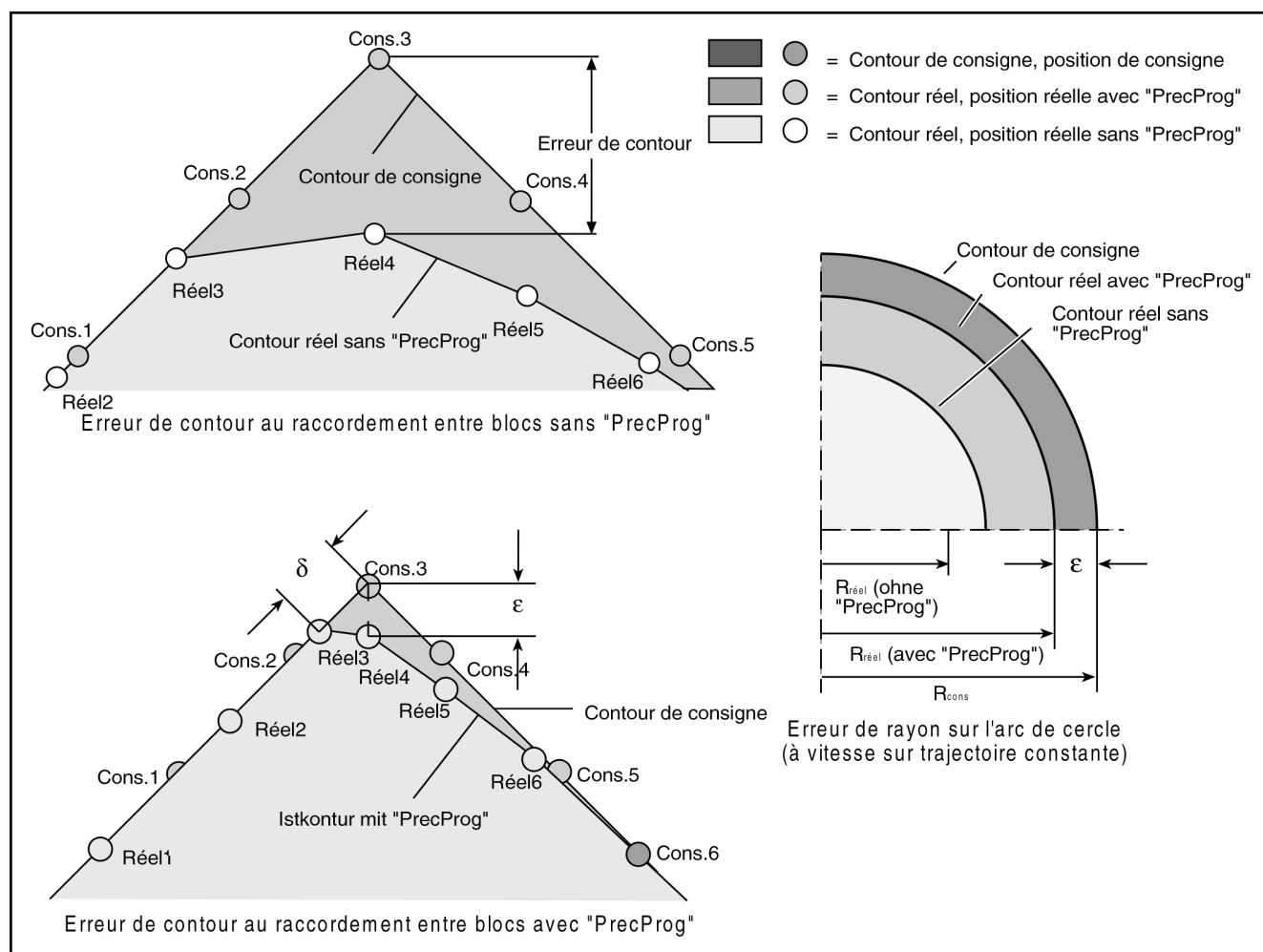


Fig. 6-31: Programmation de précision



Contrairement à la fonction "Arrêt précis" (G61/G62 ; voir [chap. 5.2.21 "Arrêt précis ACTIVÉ/DÉSACTIVÉ G61, G62" à la page 166](#)), la vitesse n'est pas nécessairement freinée jusqu'à 0 au niveau d'un enchaînement de blocs. Voir "[Particularités et restrictions :](#)" à la page 358.

Syntaxe :

PrecProg ou PrecProg(1)	Programmation de précision ACTIVÉE Erreur de contour/rayon maximale admissible ϵ selon MP 8003 00001.
PrecProg(EPS<e>)	Programmation de précision ACTIVÉE Erreur de contour/rayon maximal admissible : < ϵ >.
PrecProg(DIST<d>)	Programmation de précision ACTIVÉE Poursuite de trajectoire maximale admissible : < δ >. Pour les segments circulaires de la trajectoire : respecter la valeur ϵ issue de MP 8003 00001.
PrecProg() ou PrecProg(0)	Programmation de précision DÉSACTIVÉE
Format abrégé : PRP (..)	

Fonctions CN avec syntaxe de langage standard

avec	
<ε>	Erreur de contour/rayon maximale admissible en mm ou pouces (en fonction de G70/G71).
<δ>	Poursuite de trajectoire maximale admissible en mm ou pouces (en fonction de G70/G71).

Tab. 6-108: Syntaxe PrecProg (PRP)

- Particularités et restrictions :**
- Tous les axes concernés doivent être réglés avec une dynamique identique.
 - Tant que G8 et G62 ne sont pas actifs, la vitesse est freinée à $v = 0$ à chaque enchaînement de blocs.
 - La programmation de l'arrêt précis n'apporte des résultats sensés que si l'influence de l'accélération et de la poursuite peut être négligée.



Les expressions entre parenthèses "(0)" et "(1)" ne sont pas des paramètres de la fonction CN. La valeur numérique entre parenthèses ne peut donc pas être transmise par des variables CPL.

6.87 Traitement du programme en fonction de la préparation des blocs "PREP"

Description Si un passage critique avec des blocs très courts (en temps) se présente dans le programme pièce, il convient de s'assurer le cas échéant que la mémoire tampon des blocs préparés est largement remplie avant de traiter ce passage. Cela permet d'éviter les coupures immédiates (le traitement ne peut pas prendre un nouveau bloc car aucun n'est entièrement préparé) lorsque la préparation des blocs dure plus longtemps que le traitement du bloc.

De même, il peut s'avérer défavorable que la MTX démarre l'exécution d'un programme pièce dès que le premier bloc est entièrement préparé. Dans ce cas, la mémoire tampon de la préparation est encore vide. Si le premier bloc est extrêmement court, cela peut produire une coupure.

Enfin, la MTX propose de transférer un programme à la commande via la définition du bloc CN en mémoire tampon, puis de le faire exécuter. La communication est cependant traitée avec une priorité moindre et peut se dérouler de manière temporisée. Cette situation peut également entraîner une coupure.

Dans ces trois scénarios d'application, il est possible d'arrêter volontairement l'exécution avant le passage critique jusqu'à ce que la préparation des blocs ait terminé de préparer suffisamment de blocs. La fonction CN "PREP" est utilisée à cet effet. Le nombre de blocs à préparer est alors programmé avec la fonction.

La fonction retarde l'exécution des blocs de 2 secondes au maximum. S'il s'avère impossible de préparer le nombre de blocs programmé une fois ce temps écoulé, une erreur d'exécution est générée. Cette erreur peut être neutralisée en programmant le mot clé "SILENT".



Les blocs préparés sont enregistrés dans la mémoire tampon de la prévisualisation de blocs.

La taille de cette mémoire tampon est définie avec NCO / LookAh / Ch[k] / NofBlkPrep et doit être au moins aussi grande que le nombre de blocs programmé.

Fonctions CN avec syntaxe de langage standard

La fonction PREP ne doit en aucun cas être programmée **dans des sous-programmes asynchrones**.

Pour identifier explicitement la fin d'un passage devant être préparé entièrement, il est possible de programmer une fonction PREP () ici. Dans ce cas, il est possible d'inscrire un PREP (<N>) (avec un nombre de blocs à préparer <N> élevé) au début de l'extrait de programme et un PREP () à la fin. Le premier bloc suivant le premier PREP est exécuté seulement après que tous les blocs ont été entièrement préparés jusqu'au deuxième PREP.

Syntaxe :

PREP (<NbBlocs> { , SILENT })	
avec	
<NbBlocs>	Nombre de blocs à préparer
SILENT	Quand ce mot-clé est programmé, aucun message d'erreur n'est généré si le délai d'expiration est écoulé.
PREP ()	
	Fin d'un extrait de programme devant être préparé entièrement

Tab. 6-109: Syntaxe de la fonction CN PREP

Exemple : Programme:

```
N10 PREP(30)
N20 G1 X10 Y10 F12000 ; bloc 1
; bloc de commentaire pur -> n'est pas compris dans le calcul.
N30 X10.3 Y10.2 ; bloc 2
...
N300 X120.4 Y13.1 ; bloc 29
N310 X121.2 Y13.2 ; bloc 30, lorsque la préparation de ce bloc est terminée, l'exécution de N20
commence
```

Par exemple, si la mémoire tampon de la prévisualisation de blocs est trop petite ou si le programme est indiqué via la définition du bloc CN en mémoire tampon et qu'une temporisation plus importante apparaît, alors une erreur d'exécution est générée puisque le nombre de blocs requis n'a pas pu être préparé.

Pour supprimer l'erreur d'exécution, il est possible de programmer la ligne N10 PREP(30, SILENT) à la place de la ligne N10.

Particularités et restrictions :

- La mémoire tampon de la préparation des blocs doit être aussi grande que le nombre programmé de blocs à préparer.
- La fonction attend uniquement aux modes Saisie manuelle, Définition du bloc CN et Automatique. La fonction n'attend dans aucun autre type de fonctionnement (par ex. Bloc unique).
- Si un WAIT ou la fin du programme ont été trouvés au cours de l'attente, alors les blocs sont entièrement préparés jusqu'à ce point avant que l'exécution reprenne.
- Il n'y a généralement pas d'attente au cours d'une remise à zéro.
- L'ordre PREP ne doit pas être utilisée dans un sous-programme asynchrone. Une erreur d'exécution est générée en cas de programmation dans un sous-programme asynchrone.
- Une coupure de communication (en cas de définition du bloc CN en mémoire tampon) génère un message d'erreur si l'interrupteur SILENT n'a pas été programmé.
- Aucun état d'attente du canal n'est activé.

Fonctions CN avec syntaxe de langage standard

- Pour les blocs à préparer, seuls les blocs qui arrivent à l'interpolateur sont comptés. Cela signifie que les blocs CPL ou les blocs vides, par exemple, ne sont pas inclus dans le calcul. Pour les blocs qui sont divisés en raison de leur géométrie (Split), chaque pièce individuelle est comptée séparément.
- Si le traitement est effectué avec la définition du bloc CN, alors le premier bloc d'un message de bloc CN en mémoire tampon ne doit pas contenir de drapeau d'auto-démarrage ni de PREP.

6.88 Temps de déclenchement de la course

6.88.1 Temps de déclenchement de la course (point d'arrivée de l'interpolation) "PtBlkEnd, PTE"

Description :



Agit en combinaison avec les fonctions "Poinçonnage" (voir [chap. 6.90 "Poinçonnage Punch, PUN" à la page 366](#)) et "Grignotage" (voir [chap. 6.76 "Grignotage Nibble, NIB" à la page 332](#)).

La fonction :

- définit l'événement de référence temps "L'interpolateur CN atteint le point d'arrivée du mouvement de déplacement" pour les axes indiqués ;
- définit le laps de temps entre l'événement de référence temps et le déclenchement de la course.

Cela permet de décaler librement le moment de déclenchement de la course par rapport à la référence temps.

Les alternatives suivantes sont possibles :

- Déclenchement de course anticipé
(par ex. pour compenser un temps d'attente constant dépendant de l'application provoqué par le traitement du signal)
- Déclenchement de course retardé
(par ex. pour augmenter la précision du positionnement des axes à faible dynamique ou lors d'un poinçonnage avec serre-flans)



Autres fonctions influençant le temps de déclenchement de la course :

- PtBlkEnd (voir [chap. 6.88.1 "Temps de déclenchement de la course \(point d'arrivée de l'interpolation\) PtBlkEnd, PTE" à la page 360](#))
- PtInpos (voir [chap. 6.88.3 "Temps de déclenchement de la course \(fenêtre Inpos\) PtInpos, PTI" à la page 363](#))

Syntaxe :

PtBlkEnd (<axe><temps>{, <axe><temps>}...)
Format abrégé : PTE (. .)
avec

Fonctions CN avec syntaxe de langage standard

<axe>	Nom logique de l'axe
<temps>	Laps de temps voulu (en ms) entre l'événement de référence temps et le déclenchement de la course. 0 : déclenchement de la course au moment de l'événement Valeur négative : déclenchement de course anticipé Valeur positive : déclenchement de course retardé

Tab. 6-110: Syntaxe PtBlkEnd (PTE)

- Exemples :**
- Configuration de l'axe : X, Y, C
 - Durée de cycle Sercos : 3 ms

Réglage des temps de déclenchement de la course :

```

:
N10 PTE(X-10) Événement de référence temps pour l'axe X :
:
: "L'interpolateur CN atteint le point d'arrivée du mouvement de
: déplacement"
: Déclenchement de course anticipé de 10 ms
N20 Événement de référence temps pour les axes Y et C :
PTI(Y10,C2) "Fenêtre Inpos atteinte"
:
: (pour la fonction, voir chap. 6.88.3 "Temps de déclenchement
: de la course \(fenêtre Inpos\) PtInpos, PTI" à la page 363).
:
: Axe Y : déclenchement de course retardé de 10 ms
: Axe C : déclenchement de course retardé de 2 ms
    
```

Effet pendant le mouvement de déplacement :

```

:
N100 G1 Y20 Y définit le comportement de déclenchement de la course
C10 puisque, à références de temps égales, un temps d'attente su-
: périeur à été programmé (voir N20).
:
: Les temps d'attente étant arrondis à l'entier supérieur sur la
: grille de la durée de cycle Sercos, il en résulte un déclenche-
: ment de course retardé effectif de 12 ms.
N110 X20 C20 C détermine le comportement de déclenchement de la course
: puisque les axes avec la référence temps "Fenêtre Inpos" sont
: "plus faibles" que les axes avec la référence temps "Point d'ar-
: rivée de l'interpolation".
:
: Les temps d'attente étant arrondis à l'entier supérieur sur la
: grille de la durée de cycle Sercos, il en résulte un déclenche-
: ment de course retardé effectif de 3 ms.
N120 X30 X définit le comportement de déclenchement de la course
: puisque seul l'axe X est déplacé.
:
: Déclenchement de course anticipé effectif : -9 ms
N130 PTD Définir le déclenchement de la course de tous les axes du sys-
: tème conformément à MP 8001 00020 et MP 8001 00021
:
: (pour la fonction, voir chap. 6.88.2 "Temps de déclenchement
: de la course \(restauration des valeurs par défaut\) PtDefault,
: PTD" à la page 362).
    
```

Fonctions CN avec syntaxe de langage standard

- Particularités et restrictions :**
- Les fonctions PTD, PTE et PTI sont des fonctions modales qui se révoquent mutuellement.
 - Le poinçonnage ou grignotage doit être défini dans le paramètre MP 8001 00010.
 - Tous les temps programmés sont arrondis à l'entier supérieur sur la grille du temps de cycle Sercos.
 - Lors des mouvements de déplacement auxquels participent plusieurs axes avec des références temps et des temps d'attente différents, l'axe "le plus faible" définit le comportement effectif du déclenchement de la course.

Les points suivants s'appliquent ici :

- Les axes avec la référence temps "Fenêtre Inpos" (voir [chap. 6.88.3 "Temps de déclenchement de la course \(fenêtre Inpos\) PtInpos, PTI" à la page 363](#)) sont "plus faibles" que les axes avec la référence temps "Point d'arrivée de l'interpolation".
- À référence temps égale, l'axe ayant le temps d'attente le plus long est considéré comme "le plus faible".
- La fonction PtDefault (voir [chap. 6.88.2 "Temps de déclenchement de la course \(restauration des valeurs par défaut\) PtDefault, PTD" à la page 362](#)) permet de restaurer les valeurs par défaut du déclenchement de course pour tous les axes dans le système.

6.88.2 Temps de déclenchement de la course (restauration des valeurs par défaut) "PtDefault, PTD"

Description :



Agit en combinaison avec les fonctions "Poinçonnage" (voir [chap. 6.90 "Poinçonnage Punch, PUN" à la page 366](#)) et "Grignotage" (voir [chap. 6.76 "Grignotage Nibble, NIB" à la page 332](#)).

Définit le moment de déclenchement de la course de tous les axes dans le système sur les valeurs définies dans MP 8001 00020 et MP 8001 00021.

Syntaxe :

PtDefault

Format abrégé : PTD

Tab. 6-111: Syntaxe PtDefault, PTD

Exemple :

voir [chap. 6.88.1 "Temps de déclenchement de la course \(point d'arrivée de l'interpolation\) PtBlkEnd, PTE" à la page 360](#).



Autres fonctions influençant le temps de déclenchement de la course :

- PtBlkEnd
(voir [chap. 6.88.1 "Temps de déclenchement de la course \(point d'arrivée de l'interpolation\) PtBlkEnd, PTE" à la page 360](#))
- PtInpos
(voir [chap. 6.88.3 "Temps de déclenchement de la course \(fenêtre Inpos\) PtInpos, PTI" à la page 363](#))

Particularités et restrictions :

- Les fonctions "PTD", "PTE" et "PTI" sont des fonctions modales qui se révoquent mutuellement.
- Le poinçonnage ou grignotage doit être défini dans le paramètre MP 8001 00010.

6.88.3 Temps de déclenchement de la course (fenêtre Inpos) "PtInpos, PTI"

Description :



Agit en combinaison avec les fonctions "Poinçonnage" (voir [chap. 6.90 "Poinçonnage Punch, PUN" à la page 366](#)) et "Grignotage" (voir [chap. 6.76 "Grignotage Nibble, NIB" à la page 332](#)).

La fonction :

- définit l'événement de référence temps "Fenêtre Inpos atteinte" pour les axes indiqués ;
- définit le laps de temps entre l'événement de référence temps et le déclenchement de la course.

Cela permet de retarder librement l'instant de déclenchement de la course par rapport à la référence temps (par ex. pour augmenter la précision de positionnement sur les axes à faible dynamique ou lors d'un poinçonnage avec serre-flans).



Autres fonctions influençant le temps de déclenchement de la course :

- PtBlkEnd
(voir [chap. 6.88.1 "Temps de déclenchement de la course \(point d'arrivée de l'interpolation\) PtBlkEnd, PTE" à la page 360](#))
- PtInpos
(voir [chap. 6.88.3 "Temps de déclenchement de la course \(fenêtre Inpos\) PtInpos, PTI" à la page 363](#))

Syntaxe :

PtInpos (<axe><temps>{, <axe><temps>}...)	
Format abrégé : PTI (. .)	
avec	
<axe>	Nom logique de l'axe
<temps>	Laps de temps voulu (en ms) entre l'événement de référence temps et le déclenchement de la course.

Tab. 6-112: Syntaxe PtInpos (PTI)

Exemple :

voir [chap. 6.88.1 "Temps de déclenchement de la course \(point d'arrivée de l'interpolation\) PtBlkEnd, PTE" à la page 360](#).

Particularités et restrictions :

- Les fonctions "PTD", "PTE" et "PTI" sont des fonctions modales qui se révoquent mutuellement.
- Le poinçonnage ou grignotage doit être défini dans le paramètre MP 8001 00010.
- Tous les temps programmés sont arrondis à l'entier supérieur sur la grille du temps de cycle Sercos.
- Lors des mouvements de déplacement auxquels participent plusieurs axes avec des références temps et des temps d'attente différents, l'axe "le plus faible" définit le comportement effectif du déclenchement de la course.

Les points suivants s'appliquent ici :

- Les axes avec la référence temps "Fenêtre Inpos" (voir [chap. 6.88.3 "Temps de déclenchement de la course \(fenêtre Inpos\)](#)

Fonctions CN avec syntaxe de langage standard

- PtInpos, PTI" à la page 363) sont "plus faibles" que les axes avec la référence temps "Point d'arrivée de l'interpolation".
- À référence temps égale, l'axe ayant le temps d'attente le plus long est considéré comme "le plus faible".
 - La fonction "PtDefault" (voir chap. 6.88.2 "Temps de déclenchement de la course (restauration des valeurs par défaut) PtDefault, PTD" à la page 362) permet de restaurer les valeurs par défaut du déclenchement de course pour tous les axes dans le système.

6.89 Mouvement de déplacement PTP "PtpMove, PTP"

Description : Si une transformation d'axe est active, un mouvement de déplacement **PointToPoint** (PTP) est exécuté, c.-à-d. que le mouvement est exécuté dans le système MCS ou ACS sans désactiver la transformation d'axe. La position finale programmée peut être paramétrée dans le système WCS, MCS ou ACS. La commande réalise la conversion correspondante en interne.



Vous trouverez des informations complètes sur l'utilisation et du paramétrage de PTP dans la documentation "Description des fonctions IndraMotion MTX".

Syntaxe :

PTP (<SysPosFin>, <SysMouv>,TB<mode>) et programmation de la position finale dans EndposSys (en alternative : PtpMove (,))	
avec	
<SysPosFin>	Système de coordonnées dans lequel la position finale est programmée. Sont autorisés : WCS : Position finale programmée en coordonnées de la pièce à usiner MCS : Position finale programmée en coordonnées de machine AC : Position finale programmée en coordonnées d'axe

Fonctions CN avec syntaxe de langage standard

<SysMouv>	Système de coordonnées dans lequel le mouvement est effectué. Sont autorisés : MCS : Mouvement en coordonnées de machine ACS: Mouvement en coordonnées d'axe
TB<mode>	TrafoBranch Pour commander le changement de branche de transformation lors d'une transformation d'axe à deux branches. Possible uniquement avec la variante PTP(WCS,MCS). "Mode" peut avoir les valeurs suivantes : 0 (par défaut) : Entre les deux positions MCS possibles MCS(+1) et MCS(-1), la valeur approchée est la plus proche du point de départ. Pour cela, il peut être nécessaire de commuter la branche de transformation +1 sur -1 ou inversement. 10 : Conserve la branche de transformation lors du mouvement. 11 : Commute la branche de transformation lors du mouvement. +1 : Approche la branche de transformation +1, c.-à-d. la position MCS(+1). -1 : Approche la branche de transformation -1, c.-à-d. la position MCS(-1). Autres propriétés : TB est ignoré dans les transformations à une branche. Seule la variante PTP(WCS,MCS) prend en charge la programmation de TB. Une erreur de syntaxe est émise avec toutes les autres variantes.

Tab. 6-113: Syntaxe

La valeur par défaut pour ces fonctions est "MCS".

Exemple :

PTP (WCS,MCS) x50 y55	; Coord. WCS x = 50, y = 55 est approchée. ; Le mouvement se fait dans MCS.
PTP (WCS,ACS) x=IC(50)	; Déplacer la coord. WCS x de manière incrémentale de 50 ; Le mouvement se fait dans ACS.
PTP (MCS, MCS) X100 Y200	; Coord. MCS X = 100, Y = 200 est approchée. ; Le mouvement se fait dans MCS.
PTP (MCS, ACS) X101 Y201	; Coord. MCS X = 101, Y = 201 est approchée. ; Le mouvement se fait dans ACS.
PTP (ACS, MCS) XA50 C45	; Position ACS XA = 50, C = 45 est approchée. ; Le mouvement se fait dans MCS.

Fonctions CN avec syntaxe de langage standard

PTP (ACS, MCS) XA51 C46 ; Position ACS XA = 51, C = 46 est approchée.
; Le mouvement se fait dans ACS.

PTP () correspond à PTP (MCS, MCS) .

PTP (,) correspond à PTP (MCS, MCS) .

PTP (WCS,) correspond à PTP (WCS, MCS) .

PTP (,ACS) correspond à PTP (MCS, ACS) .

Exemple de la programmation de
TB avec la transformation à
5 axes 3232201 :

B10 C0	Approche de l'orientation de départ Phi = 0, Thê- ta = 30
COORD(1)	
G1 phi0 theta30	Les axes approchent B30, C0 (la branche de transfor- mation est +1).
PTP (WCS,,TB0) phi180 theta30	Parmi les deux positions MCS possibles MCS(+1) = (B30,C180) et MCS(-1) = (B-30,C0), MCS(-1) est approchée (course la plus courte avec changement de branche de transformation).
- ou -	
PTP (WCS) phi180 the- ta30	
PTP (WCS,,TB10) phi0 theta30	Aucun changement de la branche de transformation : la position MCS(-1) = (B-30,C180) est approchée.
PTP (WCS,MCS,TB11)	Changement de la branche de transformation : la position MCS(+1) = (B30,C0) est approchée. La position WCS reste Phi = 0, Thêta = 30.
PTP (WCS,,TB+1)	Approcher la branche de transformation +1. Aucun mouvement n'est produit puisque la branche de transformation est déjà +1.
PTP (WCS,MCS,TB-1)	Approcher la branche de transformation -1. La position MCS(-1) = (B-30,C180) est approchée. La position WCS reste Phi = 0, Thêta = 30.

Particularités et restrictions :

- Il s'agit d'une fonction locale, non modale.
- Un mouvement de déplacement PTP est toujours interprété en tant que programmation absolue, c.-à-d. que G91 est ignoré. Une programmation incrémentale est possible avec l'attribut de programmation "IC" (par ex. x=IC(50)).
- Lors d'un mouvement de déplacement PTP, la correction du rayon 3D doit être désactivée. Sinon, une erreur d'exécution est générée.
- Exactement un ordre PTP peut être programmé sur chaque ligne.

6.90 Poinçonnage "Punch, PUN"

Description : Active ou désactive la fonction "Poinçonnage".

Quand la fonction Poinçonnage est activée, une course est déclenchée à la fin pour :

- chaque mouvement de déplacement programmé (mode Course isolée) ou
- chaque segment de parcours généré par la fonction "NUM" (voir [chap. 6.77](#) "Répartition du bloc de déplacement : nombre de segments de

Fonctions CN avec syntaxe de langage standard

parcours NUM" à la page 334) ou "LEN" (voir chap. 6.65 "Répartition du bloc de déplacement : longueur du segment de parcours LEN" à la page 315).

Le mouvement de déplacement suivant n'est démarré qu'une fois la course terminée.



Fonctions influençant le temps de déclenchement de la course :

- PtDefault
(voir chap. 6.88.2 "Temps de déclenchement de la course (restauration des valeurs par défaut) PtDefault, PTD" à la page 362)
- PtBlkEnd
(voir chap. 6.88.1 "Temps de déclenchement de la course (point d'arrivée de l'interpolation) PtBlkEnd, PTE" à la page 360)
- PtInpos
(voir chap. 6.88.3 "Temps de déclenchement de la course (fenêtre Inpos) PtInpos, PTI" à la page 363)

Syntaxe :

Punch (1) ou Punch	Activer le poinçonnage
Punch (0) ou Punch ()	Désactiver le poinçonnage
Format abrégé : PUN (. .)	

Tab. 6-114: Syntaxe Punch (PUN)

Exemple :

- G90 est active (programmation absolue).
- Plan actif : X/Y
- Position actuelle : X=0, Y=0, C=0

:	
N10 C10	Activer le poinçonnage
Punch (1) G9 (ASHAPE)	Activer l'interpolateur avec la logique de poinçonnage L'axe C effectue une rotation à 10 degrés Pas de course, car les axes X et Y ne sont pas programmés.
N20 C60	L'axe C effectue une rotation à 60 degrés Pas de course, car les axes X et Y ne sont pas programmés.
N30 X0	Pas de mouvement de déplacement, car l'axe X est déjà sur la position 0. Course, car l'axe X se situe dans le plan actif.
N40 LEN=12	Diviser les blocs de déplacement suivants en segments de parcours égaux de 12 mm au maximum.
N50 X110	Le bloc de déplacement est divisé en 10 segments de parcours égaux de 11 mm. Course sur les positions X11, X22, X33 à X99, X110

Fonctions CN avec syntaxe de langage standard

N60 Y30 NUM=3

Masque la fonction LEN(N40) qui a un effet modal pour le bloc actuel.

Diviser le bloc de déplacement en 3 segments de parcours égaux.

Course à Y10, Y20, Y30

N70 Y90

LEN de N40 est de nouveau active.

Course à Y42, Y54, Y66, Y78, Y90.

N80 X50 Y50

Poinçonnage DÉSACTIVÉ

Punch(0) G8

Pente de la trajectoire ACTIVÉE

:

Mouvement de déplacement à X = 50, Y = 50

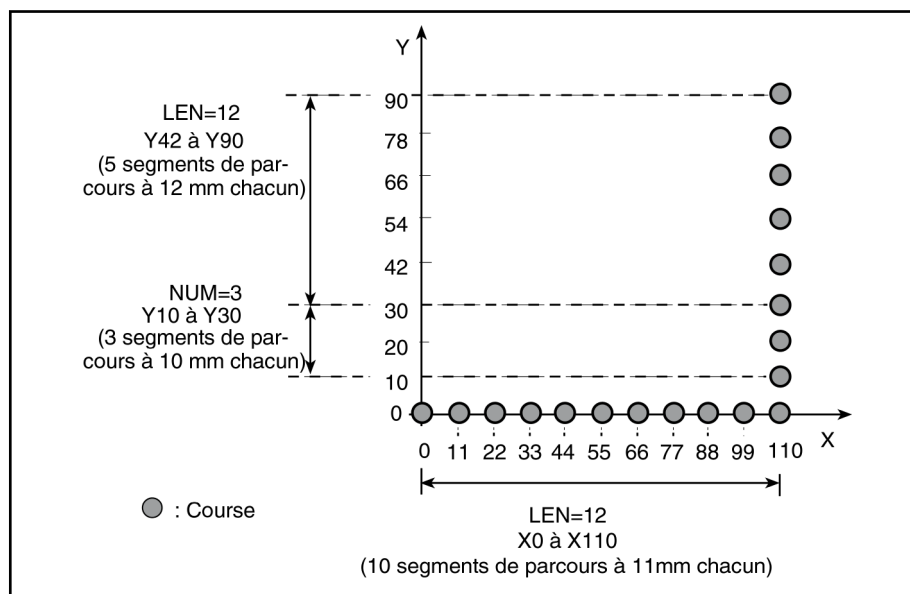


Fig. 6-32: Exemple Punch (PUN)



Pour utiliser la fonction Poinçonnage, un interpolateur prenant en charge la fonctionnalité de poinçonnage doit être activé :

G9(X... , Y... , ...)

G9(ASHAPE)

G9(SHAPE..., PUN)

Particularités et restrictions :

- Le poinçonnage doit être libéré via MP 8001 00010.
- La fonction a un effet modal, même sur des mouvements de déplacement programmés dans le même bloc.
- La fonction "NUM" ou "LEN" permet à la commande de générer automatiquement des segments de parcours à partir de la course de déplacement programmée aux extrémités de laquelle une course est déclenchée.
- Si la fonction Grignotage est activée (voir [chap. 6.76 "Grignotage Nibble, NIB" à la page 332](#)), elle révoque la fonction "Poinçonnage".
- Les blocs qui ne contiennent pas de coordonnées d'axes du plan actif ne déclenchent pas de course.

- Dans la mesure où l'API neutralise le déclenchement de la course, le traitement s'arrête sur la position de déclenchement de la course jusqu'à ce que l'API libère à nouveau le déclenchement de la course.

6.91 Accélération radiale maximale "RadialAcc, RAC"

Description : Réduit l'avance sur trajectoire pour les splines et les arcs de cercles jusqu'à ce que l'accélération radiale indiquée ne soit plus dépassée dans le système de coordonnées de la pièce à usiner (WCS).

La limitation de l'accélération radiale est désactivée après la montée en régime de la commande.

Syntaxe :

RadialAcc (<valeur>)	Paramétrage de l'accélération radiale maximale
RadialAcc(0)	Annuler la limitation de l'accélération radiale
Format abrégé : RAC (. .)	
avec	
<valeur>	Valeur d'accélération En fonction de l'unité de mesure (G71/G70), la commande interprète la valeur programmée en "1000 pouces/s ² " ou "m/s ² ".

Tab. 6-115: Syntaxe RadialAcc (RAC)

Exemple :

```

N20 RAC(0.1) ; Limiter l'accélération radiale à 0,1 m/s2
N30 G1 G8 G90 G94 F5000 ; Vitesse sur trajectoire maximale de
: 5000 mm/min
N50 G3 Y20 R10 ; Vmax = 1897 mm/min (Vmax=sqrt(RAC*R10))
N60 G2 Y30 R5 ; Vmax = 1341 mm/min (Vmax=sqrt(RAC*R5))
:
N100 RAC(0) ; Désactiver la limitation de l'accélération radiale

```

Particularités et restrictions :

- Pour les splines, la courbure maximale est déterminée uniquement par des échantillons au début, au centre et à la fin du bloc.
- La réduction est effectuée pour toute la trajectoire spline ou circulaire en tenant compte de toutes les coordonnées influençant l'avance.
- L'avance peut uniquement être réduite. Toutes les autres limitations d'avance par les paramètres machine ou par un programme pièce restent actives sans être influencées.

6.92 Enregistrement du vecteur de retrait en tant que donnée système "RecordRevVec, RRV"

Description : Active ou désactive la fonction "Enregistrement du vecteur de retrait en tant que donnée système".

Pour que cette fonction CN fonctionne, il doit exister une donnée système avec le nom <nom>RevVec</nom> et elle doit être définie avec <type>RevVec_t</type> et Dimension="Channel".

Si un bloc CN dans lequel cette fonction CN est programmée arrive à exécution, des zéros sont enregistrés dans la donnée système pour le vecteur

Fonctions CN avec syntaxe de langage standard

d'orientation et le vecteur binormal. Ceci ne se produit pas si la fonction CN est programmée dans la chaîne d'initialisation.

Si l'enregistrement est activé, le vecteur d'orientation et le vecteur binormal sont standardisés à 1 immédiatement après l'initialisation et enregistrés dans la donnée système durant le cycle d'interpolation.

Le vecteur binormal est le vecteur perpendiculaire au plan formé à partir de la direction du mouvement et du vecteur d'orientation. Lorsque G41/G42 sont actives, il s'agit par exemple du vecteur du point de pénétration de la fraise par rapport au centre de la fraise. Pour G41, il est orienté vers la gauche puisque la fraise se trouve à gauche du contour, pour G42 vers la droite.

 Voir à ce sujet les descriptions détaillées des fonctions G41/G42 et G141/G142 dans la documentation "Description des fonctions IndraMotion MTX".

Syntaxe :	RecordRevVec (1) ou RecordRevVec ()	Activer l'enregistrement des vecteurs d'orientation et binormal, le vecteur binormal pointant vers la gauche s'il n'existe aucune autre donnée due à une correction active du rayon.
	RecordRevVec (2)	Activer l'enregistrement des vecteurs d'orientation et binormal, le vecteur binormal pointant vers la droite s'il n'existe aucune autre donnée due à une correction active du rayon.
	RecordRevVec (0) ou RecordRevVec ()	Désactiver l'enregistrement des vecteurs binormal et d'orientation.
	Format abrégé : RRV (. .)	

Tab. 6-116: Syntaxe RecordRevVec (RRV)

Exemple :	N3 G17 (X,Y,Z) X0 Y0	Sélection du plan : traitement dans le plan X-Y.
	N5 RRV (2)	Activation avec un vecteur binormal vers la droite.
	:	Vecteur d'orientation = [0,0,0], vecteur binormal = [0,0,0]
	N7 G41 X8 ED3	Vecteur binormal vers la gauche à cause de G41.
	N8 X20	Vecteur d'orientation = [0,0,1], vecteur binormal = [0,1,0]

Particularités et restrictions : Si G20 est active, aucun enregistrement n'a lieu.

6.93 Réduction de couple "RedTorque, RDT"

Description : "RedTorque" masque, axe par axe, la valeur du paramètre machine 1003 00010 avec la valeur programmée.

La valeur du paramètre machine n'est pas modifiée.

De cette façon, le couple maximal, qui peut devenir effectif après un flanc positif du signal de l'interface d'axe "Réduction du couple" (qAx_TrqLim), est réduit d'un axe et n'est plus seulement limité fixement à la valeur du paramètre machine, mais devient réglable de manière "dynamique".

Syntaxe :	RedTorque (<axe1> {,<axe2>{, ...}})	Définir les valeurs du couple maximal réduit aux valeurs programmées.
	RedTorque (0) ou RedTorque ()	Définir les valeurs du couple maximal réduit aux valeurs des paramètres machine (MP 1003 00010).

Fonctions CN avec syntaxe de langage standard

Format abrégé : RDT (. .)	
avec	
<axe i>	Désignation physique ou logique de l'axe, y compris le couple maximal en % du couple d'arrêt de l'axe. Plage de valeurs : 0 à 500 %

Tab. 6-117: Syntaxe RedTorque (RDT)

Exemple :

N8 RDT (X5) : Le couple maximal de l'axe (ici : l'axe X) est limité à 5 % de son couple d'arrêt avec le prochain flanc positif du signal "Réduction du couple" pertinent spécifique à l'axe.

6.94 Poursuite du programme avec avance de bloc après interruption du programme "ReentContBlk, RCB, RCBD"

Description : Marque un emplacement ciblé dans le programme là où il peut se poursuivre après avoir été interrompu.

Syntaxe :

ReentContBlk, Format abrégé : RCB RCB(1 {,<cond. 1>{, <cond. 2>{, <cond. 3>}}}) RCBD(1)	
	Sélectionne l'emplacement comme emplacement possible de réentrée ; enregistre la syntaxe et les valeurs des conditions optionnelles. Voir les exemples 1, 3 et 4
Bed [1,2,3]	Variable CPL ou expression CPL : La variable ou l'expression peut être de type Bool, Int, Real ou Double. Si la variable est de type String ou String de champ, une évaluation a lieu dans la variable String ou String de champ de sorte que le contenu de la variable String soit interprété comme s'il avait été programmé directement dans la fonction. Voir l'exemple 4.
RCB(0)	Supprime le dernier emplacement pris en compte. Voir l'exemple 2. Le premier paramètre peut être remplacé par une variable en nombre entier. Cette dernière doit avoir la valeur 0 ou 1. Voir l'exemple 5.
RCBD(1)	Supprime le dernier emplacement pris en compte. Contrairement à RCB(0), agit aussi dans un sous-programme asynchrone.

Tab. 6-118: Syntaxe

Exemples :

Exemple 1 :

N130 RCB(1) Sélectionne le bloc N130 comme cible possible de l'avance de bloc. Aucune condition saisie.

Exemple 2 :

N190 RCB(0) Supprime la dernière cible définie de l'avance de bloc (celle définie dans l'exemple 1 dans notre exemple).

Fonctions CN avec syntaxe de langage standard

Exemple 3 :

310 FOR COURSE%=1 TO 10	Le sous-programme "PercageTrou" est exécuté 10 fois dans la boucle.
N320 X[10 * COURSE %]	Par exemple, si la boucle ou le sous-programme est interrompu au septième passage à cause d'un arrêt d'urgence, la valeur "COURSE% = 6" est enregistrée avec le dernier traitement réussi de RCB().
N330 P PercageTrou	
N340 RCB(1, COURSE %)	Si l'avance de bloc est lancée avec cette logique, alors le bloc cible N340 se trouverait dans le sixième passage de boucle, c.-à-d. que le démarrage CN lancerait le traitement dans le septième passage de boucle.
350 NEXT COURSE%	

Exemple 4 :

500 DIM STRVAR \$(30)	Est de type String.
510 STRVAR\$="COURSE%<8"	Les crochets sont nécessaires pour que l'interpréteur CPL annule la variable String et la remplace par "COURSE %<8".
...	La programmation de RCB(1, [STRVAR\$]) avec STRVAR\$="COURSE%<8" correspond donc exactement à RCB(1, COURSE%<8).
N550 RCB(1, [STRVAR\$])	
...	
STRVAR\$	

Exemple 5 :

N800 RCB([ENABLE%])	Paramétrage du premier paramètre.

Particularités et restrictions :

Les variables et expressions CPL sont uniquement évaluées au moment de la préparation des blocs. Après l'"Effacement de la course restante" et en mode Retrace, aucune réévaluation des expressions n'a lieu dans la mesure où ces dernières ont déjà été évaluées.

Les conditions peuvent uniquement être saisies lorsque le programme est lié. En cas de sélection en mode DIN, une erreur d'exécution est générée si des conditions doivent être évaluées.

Une condition peut uniquement être évaluée pour les types suivants :

Integer, Real, Double, Boolean.

Les types de champ et types structurés notamment ne doivent pas être saisis.

Les programmes participant à l'appel de RCB ne doivent pas être modifiés entre un traitement interrompu et l'"avance de bloc consécutive à un arrêt du programme". Participent tous les programmes qui étaient contenus dans la liste d'imbrication des sous-programmes du dernier ordre RCB exécuté, c.-à-d. le programme principal et au moins aussi le programme dans lequel l'ordre RCB était programmé.

L'ordre RCB est ignoré sans aucun commentaire dans les sous-programmes asynchrones. Si une marque de réentrée doit être supprimée dans un sous-programme asynchrone, utiliser la syntaxe « **RCBD(1)** » ou « **RCBDelete(1)** ». Il s'agit ici d'une extension fonctionnelle de la fonction RCB(0).

6.95 Suppression d'un axe d'un groupe d'axes "RemAxis, RAX"

Description : Supprime un axe synchrone du canal appelant. L'axe synchrone devient ainsi un axe asynchrone.

L'axe asynchrone peut ensuite être programmé dans un canal quelconque via son nom physique.



Pour obtenir des informations détaillées sur la fonctionnalité "Transfert d'axe", voir la documentation "Description des fonctions IndraMotion MTX".

Syntaxe :

RemAxis (<NPA> <IPA> <NLA>{, <NPA> <IPA> <NLA>}...)	
Format abrégé : RAX (. .)	
avec	
<NPA>	Nom physique de l'axe Définit l'axe qui doit être extrait du canal actuel.
<IPA>	Index physique de l'axe Effet similaire à <NPA>
<NLA>	Nom logique de l'axe Effet similaire à <NPA>

Tab. 6-119: Syntaxe RemAxis (RAX)

Exemple :

```
:
N030 RAX (XP,      L'axe physique XP, l'axe physique avec l'index 2 et l'axe logi-
2, Z)              que Z sont retirés du canal.
:
```

- Particularités et restrictions :**
- La préparation des blocs n'est pas suspendue lors du retrait d'un axe.
 - Les noms d'axes non valides génèrent un message d'erreur.
 - Aucun message d'erreur n'est généré si un axe à retirer est défini dans le système mais plus présent sur le canal actuel.
 - Les positions d'axe dans un même bloc doivent toujours être programmées après RemAxis(...).

6.96 Suppression d'un nom d'axe logique "RemLogName, RLN"

Description : Supprime le nom d'axe logique d'un axe synchrone sur le canal appelant.

L'axe subsiste sur le canal, mais il ne pourra plus y être programmé que par son nom physique ou par son index physique.



Pour obtenir des informations détaillées sur la fonctionnalité "Transfert d'axe", voir la documentation "Description des fonctions IndraMotion MTX".

Syntaxe :

RemLogName (<NPA> <IPA> <NLA>{, <NPA> <IPA> <NLA>}...)	
Format abrégé : RLN (. .)	
avec	

Fonctions CN avec syntaxe de langage standard

<NPA>	Nom physique de l'axe Définit l'axe dont le nom logique d'axe doit être supprimé du canal actuel.
<IPA>	Index physique de l'axe Effet similaire à <NPA>
<NLA>	Nom logique de l'axe Effet similaire à <NPA>

Tab. 6-120: Syntaxe RemLogName (RLN)

Exemple :

:

N030 RLN (YP, 3, Z) Les noms logiques de l'axe physique YP, du 3e axe physique et de l'axe logique Z sont retirés du canal appelant.

:

Particularités et restrictions :

- Un axe indiqué doit être à l'arrêt.
Si ce n'est pas le cas, la commande génère un message d'erreur et interrompt le programme.
- Les positions d'axe dans un même bloc doivent toujours être programmées après l'expression RemLogName(...).

6.97 Retrait de broches du canal "RemSpindle, RSP"

Description : Retire une broche logique du canal appelant.

Syntaxe :

RemSpindle (<NoBrSys> | <NomBrSys> | <NomBrCan> , {...})

Supprime une broche de canal du canal.

avec

<NoBrSys>	1 à 32
<NomBrSys>	Nom de la broche du système
<NomBrCan>	S1 à S8

Format abrégé : RSP

Tab. 6-121: Syntaxe RemSpindle (RSP)

Exemple :

...

N50 RSP (S2) Retire la 2e broche de canal du canal.

N60 M203 S2=500 Génère une erreur d'exécution, car il n'existe aucune 2e broche de canal.

...

Particularités et restrictions :

- Une broche exploitée à une vitesse de coupe constante (G96) ne peut pas être retirée du canal.
- Si les fonctions auxiliaires en nombre entier spécifiques au canal CSP1 à CSP8 sont définies, alors l'API est informé de l'affectation actuelle des broches de canal (par ex. CSP2 0 pour informer que la broche de canal 2 n'est plus attribuée).

6.98 Répétition de bloc "REPEAT, RPT"

Description : La fonction CN REPEAT(<facteur de répétition>) (format abrégé : RPT) permet de répéter des blocs CN.

Le facteur de répétition indique le nombre de fois que le bloc doit être exécuté (1 = une seule exécution). Les valeurs comprises entre 1 et 65 535 sont admises.

Si un appel de sous-programme est inclus, le bloc complet et le sous-programme seront exécutés à plusieurs reprises. Les sous-programmes peuvent contenir d'autres blocs avec des ordres RPT.

Si un mouvement de déplacement est programmé et qu'un sous-programme modal est actif, alors le bloc complet (sous-programmes inclus) peut être exécuté à plusieurs reprises.

RPT est possible en modes de fonctionnement Automatique et Saisie manuelle.

Les parties CPL sont parcourues uniquement lors du premier appel.

Syntaxe : RPT<facteur de répétition> ou REPEAT(<facteur de répétition>)

- <facteur de répétition> indique le nombre de fois que le bloc CN doit être exécuté.

Plage de valeurs : $1 \leq n \leq 65\,535$

- Les variables CPL sont permises.

Exemple :

```
...
N10 G91 X10 Y10 RPT5 ; Le bloc est exécuté 5 fois.
...
20 MAXCNT%=10
N20 G91 X5 Y5 ; Le bloc complet, sous-programme inclus, est exécuté 10 fois.
RPT[MAXCNT%] P Up
...
N10 G81
N20 G91 X20 Y20 ; Le bloc est exécuté 8 fois. Ce faisant, le sous-programme modal est appelé à chaque fois.
RPT(8)
...
```

Particularités et restrictions : RPT n'agit pas dans la chaîne d'initialisation, ni dans le bloc d'action de l'avance de bloc.

6.99 Sous-programmes asynchrones

6.99.1 Définition du comportement de redémarrage "REPOSDEF"

Description : Définit l'avance de redémarrage et/ou le point de redémarrage localement (au sein d'un sous-programme asynchrone).

Syntaxe :

REPOSDEF({F<avance> OF<avance>} , {TP<point>})	
avec	
F<avance>	Vitesse d'avance pour le mouvement de redémarrage automatique à la fin du sous-programme asynchrone.

Fonctions CN avec syntaxe de langage standard

OF<avance>	Vitesse d'avance optionnelle pour le mouvement de redémarrage automatique à la fin du sous-programme asynchrone. Cette valeur d'avance n'est utilisée que si aucune avance active n'est disponible lors du déclenchement du sous-programme asynchrone.
TP<point>	1 : Point de départ 2 : Point d'arrivée 3 : Point d'interruption

Tab. 6-122: Syntaxe

- Particularités et restrictions :**
- La fonction est prévue pour être utilisée dans un sous-programme asynchrone.
 - Le point de redémarrage paramétré au préalable avec ASPRTP (voir [chap. 6.8.5 "Sous-programmes asynchrones : inscription ASPSET" à la page 238](#)) pour le sous-programme asynchrone en cours reprend automatiquement effet à la fin du sous-programme.

6.99.2 Définition du point de redémarrage dans un sous-programme asynchrone "REPOSTP"

Description : *Définit si la commande doit, à la fin du sous-programme asynchrone dans lequel la fonction "REPOSTP" est programmée, se positionner sur :*

- le point de départ,
- le point d'arrivée,
- le point d'interruption

d'un bloc de déplacement éventuellement interrompu.

Si aucun bloc de déplacement n'est actif au moment de l'interruption, la commande se positionne toujours sur les dernières coordonnées actives.

REPOSTP masque ainsi un point de redémarrage paramétré préalablement avec ASPRTP pour le sous-programme asynchrone actuellement en cours (voir [chap. 6.8.4 "Sous-programmes asynchrones : Définir le point de redémarrage ASPRTP" à la page 238](#)).



Vous trouverez toutes les informations sur l'utilisation et la programmation de sous-programmes asynchrones dans la documentation "Description des fonctions IndraMotion MTX".

Syntaxe :

REPOSTP (<point>)	
avec	
<point>	Point de redémarrage voulu : 1 : point de départ 2 : point d'arrivée 3 : point d'interruption

Tab. 6-123: Syntaxe REPOSTP

- Particularités et restrictions :**
- La fonction est prévue pour être utilisée dans un sous-programme asynchrone.
 - Le point de redémarrage paramétré au préalable avec ASPRTP (voir [chap. 6.8.4 "Sous-programmes asynchrones : Définir le point de redémarrage ASPRTP" à la page 238](#)) pour le sous-programme asynchrone en cours reprend automatiquement effet à la fin du sous-programme.

Fonctions CN avec syntaxe de langage standard

- Les modifications de correction survenues entretemps au sein d'un sous-programme asynchrone sont prises en compte automatiquement pour le calcul interne du point de redémarrage requis.

6.100 Recul Retracer, RETRACE

Description : La fonction CN "Retracer" permet d'identifier des passages d'un programme pièce pour lesquels la fonctionnalité "Recul sur le contour" doit être autorisée.

Lors du traitement du programme pièce, la commande génère un programme avant et un programme arrière au sein de cet identificateur. Ces programmes peuvent être traités en fonctionnement en sens inverse.

Les programmes avant et arrière contiennent, ligne par ligne, les instructions de création du contour issues du programme pièce. Pour cela, les coordonnées programmées et une partie des fonctions CN relatives au contour sont transférées automatiquement aux programmes avant et arrière. Les mouvements d'axes asynchrones ne sont pas transférés. Pour l'ajustement des programmes avant et arrière spécifique à l'application, voir aussi ADVA, ADVS, REVA et REVS.

Le recul est commandé à l'aide des signaux d'interface.

Syntaxe :

Retracer(1)	Identificateur de début d'une zone pour laquelle le recul sur le contour doit être autorisé.
Retracer([<avt>][,<arr>])	Comme pour Retracer(1), mais avec l'enregistrement du fichier des programmes avant et arrière. Ces fichiers montrent l'image des programmes internes pour le fonctionnement en sens inverse. Les fichiers de programme sont générés en cours d'exécution. Les fichiers peuvent être visualisés à la fin du traitement de l'identification de la fin de la zone – Retracer(0).
avec	
<avt>	Nom du fichier de programme avant Retracer
<arr>	Nom du fichier de programme arrière Retracer
Retracer(0)	Identificateur de fin d'une zone pour laquelle le recul sur le contour doit être autorisé.
Format abrégé : -	

Tab. 6-124: Syntaxe Retrac, RETRACE

Le tableau ci-dessous présente les fonctions CN transférées automatiquement aux programmes avant et arrière :

Fonction CN	Groupe de fonctions
G0, G1, G2, G3	Fonctions d'interpolation *)
G8, G9	Pente de la trajectoire *)
G17, G18, G19	Présélection du plan *)
G40, G41, G42	Correction de trajectoire de la fraise *)
G43, G44	Segment de trajectoire transitoire
G45, G46	Avance au point de pénétration de la fraise/centre de la fraise

Fonctions CN avec syntaxe de langage standard

G47, G48	Correction d'outil *)
G53.x à G59.x	Décalages d'origines
G61, G62	Arrêt précis
G70, G1	Programmation en pouces/mètres
G90, G91	Programmation en valeurs absolues/relatives
G93, G94, G95	Programmation de l'avance
G140, G141, G142	Correction 3D du rayon d'outil
G153.x à G159.x	Placements
CHLL, CHSL	Programmation locale de chanfreins
RNDL	Arrondi local

*) Paramètres non pris en charge dans les programmes avant et arrière

Tab. 6-125: Fonctions CN commutables dans la zone Retrace

Exemple :

```
...
N100 Retrace(1)
N110 G1 F1000 X27.571 Y3.958
N120 X38.729 Y26.945
N130 M125 ADVA(M125) REVA(M126)
...
N200 Retrace(0)
...
```

Particularités et restrictions :

- La zone Retrace limite le fonctionnement en sens inverse. Le recul ne peut se produire qu'au sein de la plage active. Cela vaut aussi quand plusieurs zones sont définies dans le programme pièce.
- Les programmes avant et arrière contiennent la suite de blocs CN annulée sans instructions CPL et sans appels de sous-programmes.
- Aucun appel de sous-programme ne doit être inséré dans un programme avant ou arrière.
- Le fonctionnement en sens inverse prend en charge l'interpolation linéaire et circulaire.
- Les commutations modales dans la zone Retrace sont uniquement prises en charge pour les fonctions mentionnées au [Tab. 6-125 "Fonctions CN commutables dans la zone Retrace" à la page 377](#).



Les expressions entre parenthèses "(0)" et "(1)" ne sont pas des paramètres de la fonction CN. La valeur numérique entre parenthèses ne peut donc pas être transmise par des variables CPL.

6.101 Extension du bloc-programme arrière Retrace "ReverseAppend, REVA"

Description : En fonctionnement en sens inverse, un bloc-programme arrière est généré dans la zone Retrace de Retrace(1) à Retrace(0) pour chaque bloc-programme pièce. La fonction ReverseAppend ajoute les fonctions CN indiquées à la fin du bloc-programme arrière actuel.

Fonctions CN avec syntaxe de langage standard

Syntaxe :

ReverseAppend (<FCN>...<FCN>)	
Format abrégé : REVA (...)	
avec	
<FCN>...<FCN>	Liste des fonctions CN

Tab. 6-126: Syntaxe ReverseAppend (REVA)



En cas de fonctionnement en sens inverse, toute modification du contour entraîne une erreur de commutation.

Les fonctions CPL et les appels de sous-programmes ne sont pas autorisés.

Exemple :

Programme pièce :	Programme arrière :	Observation :
N100 X100 M10	N1 X=10	Les fonctions auxiliaires ne sont pas reprises.
N200 X0 REVA(M125 M126)	N2 X=100 M125 M126	Dans le programme arrière, M125 et M126 doivent être indiqués à l'API.



À des fins de contrôle, vous pouvez enregistrer le programme arrière modifié dans un fichier (voir [chap. 6.100 "Recul Retracer, RETRACE" à la page 377](#)).

6.102 Remplacement du bloc-programme arrière Retracer "ReverseSubstitute, REVS"

Description : En fonctionnement en sens inverse, un bloc-programme arrière est généré dans la zone Retracer de Retracer(1) à Retracer(0) pour chaque bloc-programme arrière. La fonction "ReverseSubstitute" remplace le bloc-programme arrière actuel par les fonctions CN indiquées.

Syntaxe :

ReverseSubstitute (<FCN>...<FCN>)	
Format abrégé : REVS (...)	
avec	
<FCN>...<FCN>	Liste des fonctions CN

Tab. 6-127: Syntaxe ReverseSubstitute (REVS)



En cas de fonctionnement en sens inverse, toute modification du contour entraîne une erreur de commutation.

Les fonctions CPL et les appels de sous-programmes ne sont pas autorisés.

Fonctions CN avec syntaxe de langage standard

Exemple :

Programme pièce :	Programme arrière :	Observation :
N1 X100	N1 X10	
N2 X0 F10 REVS(X100 F100)	N1 X100 F100	Autoriser une avance supérieure dans le programme arrière. Utiliser le point de départ pour les coordonnées programmées dans le programme arrière !



À des fins de contrôle, vous pouvez enregistrer le programme arrière modifié dans un fichier (voir [chap. 6.100 "Recul Retrace, RETRACE"](#) à la page 377).

6.103 Aide à la saisie : rotation "Rotate(...), ROT(...)"

Description : La fonction "Rotation" compte parmi les aides à la saisie.

La commande fait pivoter un contour programmé dans le plan actif (pour les plans, voir G17, G18, G19 ou G20).

La rotation se rapporte toujours au point de rotation actuel (voir fonction "PoleSet", [chap. 6.83 "Définition du point d'inversion miroir/de rotation PoleSet, PLS"](#) à la page 351). Si celui-ci n'a pas été programmé explicitement, le point d'origine actuel du programme sert de point de rotation.

Cette fonction permet de ne programmer qu'une seule fois les éléments de programme récurrents qui sont pivotés selon un angle défini.

De plus, il devient inutile de convertir les cotes des pièces à usiner angulaires en coordonnées du système de coordonnées de base de la pièce à usiner. Il suffit de reprendre directement les cotes des plans de construction et d'indiquer l'angle de rotation correspondant. La commande se charge du reste.



La rotation est une aide à la saisie et ne modifie donc pas le système actuel des coordonnées du programme. Une aide à la saisie n'est en fait qu'une alternative de saisie des coordonnées de programme.

La rotation peut également être utilisée conjointement avec la fonction miroir et la mise à l'échelle.

Les points suivants s'appliquent ici :

- La fonction a un effet modal. Elle reste valide jusqu'à ce qu'elle soit désactivée.
- Elle peut être programmée dans le même bloc que d'autres conditions de course et fonctions auxiliaires.

Syntaxe :

Rotate (<angle de rotation>)	Activer la rotation autour de l'<angle de rotation> indiqué.
Rotate (0) ou Rotate ()	Désactiver la rotation pour tous les axes du canal. Définir tous les angles de rotation sur "0". Les positions d'axe atteintes sont conservées jusqu'à ce qu'elles soient reprogrammées.
Format abrégé : ROT (. .)	

Fonctions CN avec syntaxe de langage standard

avec	
<angle de rotation>	> 0 : rotation trigonométrique < 0 : rotation antitrigonométrique = 0 : désactiver la rotation Toutes les coordonnées programmées suivantes du plan actif sont pivotées autour du point de rotation (voir PoleSet, chap. 6.83 "Définition du point d'inversion miroir/de rotation PoleSet, PLS" à la page 351). La rotation ne devient active qu'avec l'information de déplacement suivante.

Tab. 6-128: Syntaxe Rotate(...), ROT(...)

Exemple :

: Activer la rotation

N30 ROT(45) Toutes les coordonnées du plan actif programmées ensuite effectuent une rotation trigonométrique de 45 degrés autour d'un point de rotation éventuellement programmé.

:

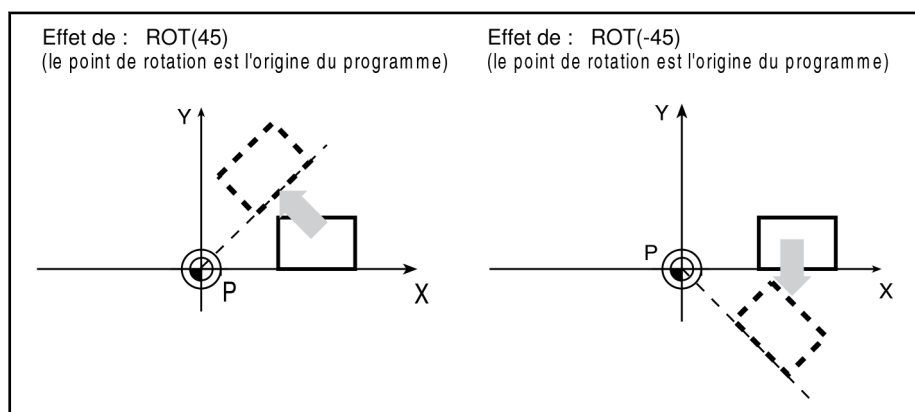


Fig. 6-33: Exemple Rotate(...), ROT(...)

Rotation d'un vecteur d'orientation :

La rotation d'un vecteur d'orientation s'effectue autour de la normale du plan actif.

Même si la fonction Mise à l'échelle ou un point d'inversion miroir/de rotation est actif, ceci n'a aucune influence sur le résultat.

La syntaxe est décrite sous "Programmation".

Particularités et restrictions :

- La fonction tient compte des paramètres d'interpolation avec une interpolation circulaire.
- Elle influence le décalage programmable des segments de contour (Shift).
 Voir [chap. 6.110 "Décalage programmé du contour Shift, SHT"](#) à la page 392.
- Elle n'influence **pas** :
 - les décalages d'origines (G54 à G59.5 ; voir [chap. 5.2.20 "Décalages d'origine \(DO\) G53, G53.1-G59.1 à G53.5-G59.5"](#) à la page 164),
 - les décalages des coordonnées du programme (Trans ou ATrans ; voir [chap. 6.144 "Décalage des coordonnées du programme Trans, TRS, décalage additif des coordonnées du programme ATrans, ATR"](#) à la page 437),

Fonctions CN avec syntaxe de langage standard

- l'initialisation de la position du programme ("SetPos" ; voir [chap. 6.109 "Initialisation de la position du programme SetPos, SPS"](#) à la page 392),
- l'approche des coordonnées du point de référence (G74 ; voir [chap. 5.2.25 "Approche des coordonnées du point de référence G74"](#) à la page 171),
- l'approche de la position fixe des axes machine (G76 ; voir [chap. 5.2.28 "Approche de la position fixe des axes machine G76"](#) à la page 174),
- les valeurs de correction du rayon de la fraise et des longueurs d'outils.

6.104 Arrondissage des angles

6.104.1 Arrondissage des angles avec indication de la différence "RoundEps, RNE"

Description : Cette fonction ajoute des arcs de cercles de raccordement tangentiels entre 2 blocs linéaires dans le niveau principal.

Certes, ceci modifie légèrement le contour programmé à l'endroit de tels angles, mais il est possible d'obtenir des profils de vitesse et d'accélération continus lors de l'interpolation.

Syntaxe :

RoundEps (<différence>)	Activer l'arrondissage des angles entre 2 blocs linéaires.
RoundEps (DEF)	Activer l'arrondissage des angles entre 2 blocs linéaires avec la différence par défaut issue de MP 7050 00110.
RoundEps (0) ou RoundEps ()	Désactiver l'arrondissage des angles entre 2 blocs linéaires.
Format abrégé : RNE (. .)	
avec	
<différence>	Différence maximale admissible (en mm) entre le contour modifié et le contour programmé. Les décimales sont permises. La commande calcule elle-même un arc de cercle de raccordement tangentiel approprié.

Tab. 6-129: Syntaxe RoundEps, RNE

- Particularités et restrictions :**
- Les fonctions "ChLength", "ChSection", "RoundEps" et "Rounding" ont un **effet modal** et se révoquent mutuellement.
 - *La commande n'exécute **pas** la fonction "Arrondissage des angles" si l'un des cas suivants se produit :*
 - au moins l'un des deux blocs consécutifs n'est pas un bloc linéaire ;
 - au moins l'un des deux blocs consécutifs a un segment de trajectoire situé en dehors du niveau principal sélectionné ;
 - au moins l'un des deux blocs consécutifs a une course de déplacement plus courte que la course réglée dans MP 7050 00120 (2 à 90 mm, valeur par défaut : 10 mm) ;

Fonctions CN avec syntaxe de langage standard

- l'enchaînement de blocs est considéré comme continu selon le paramètre machine, c.-à-d. si l'angle entre les deux blocs est inférieur à l'angle maximal défini dans MP 7050 00130 (valeur par défaut = 1 degré).
- Si le plan, un décalage d'origine ou une transformation d'axe sont modifiés entre 2 blocs successifs, alors aucun arrondi n'est généré.

6.104.2 Arrondissement des angles avec indication du rayon "Rounding", "RND", "RNDL"

Description : Cette fonction ajoute des arcs de cercles de raccordement tangentiels entre 2 blocs linéaires, circulaires ou hélicoïdaux dans le niveau principal.

Certes, ceci modifie légèrement le contour programmé à l'endroit de tels angles, mais il est possible d'obtenir des profils de vitesse et d'accélération continus lors de l'interpolation.

La fonctionnalité "Arrondissement des angles avec indication du rayon" peut avoir un effet modal (Rounding, RND) ou local au sein du bloc (RNDL). Pour un arrondi de bloc, il est possible de programmer en option une avance locale avec laquelle l'arrondi doit être usiné.

Syntaxe :

Rounding(<rayon>)	Activer l'arrondissement modal des angles entre 2 blocs linéaires/circulaires/hélicoïdaux.
Rounding(0) ou Rounding()	Désactiver l'arrondissement des angles entre 2 blocs linéaires/circulaires/hélicoïdaux.
Format abrégé : RND (. .)	
avec	
<rayon>	Rayon voulu pour l'arc de cercle de raccordement, les décimales étant permises.

Tab. 6-130: Syntaxe Rounding (RND)

RNDL(<rayon> {, <Avance-Loc>})	Arrondissement local des angles
avec	
<rayon>	Rayon voulu pour l'arc de cercle de raccordement, les décimales étant permises.
<LocFeed>	Avance locale avec laquelle l'arrondi doit être usiné.

Tab. 6-131: Syntaxe RNDL

Particularités et restrictions :

- Les fonctions "ChLength", "ChSection", "RoundEps" et "Rounding" ont un **effet modal** et se révoquent mutuellement.
- La fonction "RNDL" a un **effet local** (bloc par bloc).
- Si un chanfrein ou un arrondi agissent de manière modale et qu'un arrondi local est programmé simultanément, alors la fonction locale écrase la fonction modale au sein du bloc.
- *La commande n'exécute pas la fonction "Arrondissement des angles" si l'un des cas suivants se produit :*
 - au moins l'un des deux blocs consécutifs a une course de déplacement plus courte que la course réglée dans MP 7050 00120 (2 à 90 mm, valeur par défaut : 10 mm) ;

Fonctions CN avec syntaxe de langage standard

- l'enchaînement de blocs a une valeur continue selon les paramètres machine, c.-à-d. si l'angle entre deux blocs est inférieur à l'angle maximal indiqué dans MP 7050 00130 (valeur par défaut = 1 degré).
- Si le plan, un décalage d'origine ou une transformation d'axe sont modifiés entre 2 blocs successifs, alors aucun arrondi n'est généré.
- Seules les composantes du plan actif sont prises en compte pour l'arrondissement. De ce fait, la direction dans l'espace des droites tridimensionnelles est modifiée. Il en va de même pour les segments de trajectoire hélicoïdaux.

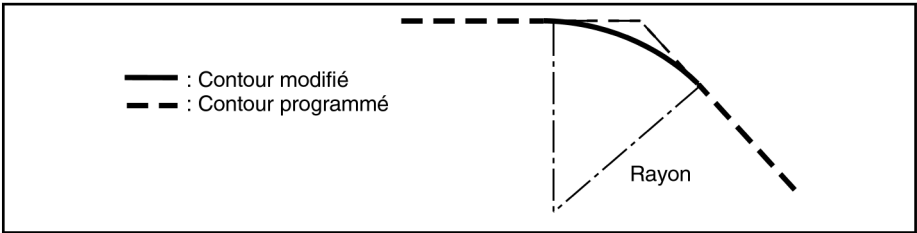


Fig. 6-34: Arrondissement des angles avec indication du rayon

6.105 Synchronisation du couplage des axes système "SACSYNC"

Description : Synchronise le programme CN à l'état du couplage des axes système (SAC) et applique les positions de découplage (\$Position de consigne) dans le système de coordonnées de la pièce à usiner sans déclencher de mouvement de l'axe.

Si un mouvement est programmé dans le même bloc, l'ordre réduit la vitesse sur trajectoire à zéro au niveau du point d'arrivée et agit une fois le point d'arrivée atteint.

La position est reprise et le traitement du programme se poursuit dès que le couplage d'axes système se trouve à l'état affecté.

Syntaxe :

SACSYNC	Attend l'arrêt du couplage d'axes système (SysAxCoupleSta[i].State=0) de tous les axes du canal et reprend la position de découplage dans le système de coordonnées de la commande numérique.
SACSYNC (TO=0)	Reprend les positions de découplage de tous les axes du canal découplés sans attendre.
SACSYNC (<axe1>=<état SAC>, ..., <axeN>=<état SAC>)	Attend jusqu'à ce que tous les états SAC indiqués pour les axes aient été atteints et reprend la position d'arrêt pour les axes découplés.
SACSYNC (TO=<temps>, <axe1>=<état SAC>, ..., <axeN>=<état SAC>)	Comme précédemment, mais avec une attente jusqu'à l'écoulement du temps <temps> en ms seulement. La commande dépose une erreur sans reprendre les positions de couplage lorsque la condition d'attente n'est pas remplie.

Fonctions CN avec syntaxe de langage standard

avec <axe i>	Désignateur d'axe du canal ou désignateur d'axe système ou désignateur de broche du canal (S1 à S8) ou désignateur de broche du système
où l'état SAC peut accepter les valeurs suivantes provenant de SysAxCoupleSta[i].State : 0 : arrêt (OFF) 1 : coupler (HUNTING) 2 : recoupler (RECOUPLING) 3 : synchrone (COUPLED) 4 : arrêter (STOPPING) 5 : arrêté (STOPPED) 6 : découpler (DISCOUPLE) 7 : erreur (ERROR) 1x : annulation (ABORT-x) avec x de 1-7 2x : commutation au temps réel (PENDING-x) avec x de 1-7 En tant que condition d'attente, il est possible de superposer aussi les valeurs issues de SysAxCoupleSta[i].StateAddOn (avec x de 1-7 et y de 1-2) dans l'état SAC : 1000, 10yx : Fenêtre du mode synchrone 1 atteinte (INPOS1) 2000, 20yx : Fenêtre du mode synchrone 2 atteinte (INPOS2) 3000, 30yx : Fenêtre du mode synchrone 1 et fenêtre du mode synchrone 2 atteintes (INPOS1 + INPOS2)	Remarque : Les conditions d'attente ne devraient pas utiliser les états statiques tels que : 0 : ARRÊT 3 : synchrone 5 : stoppé puisque les états de transition tels que : 1 : coupler 2 : recoupler 4 : arrêter 6 : découpler ne sont pas, ou brièvement seulement, acceptés selon la position et la vitesse d'axe. SACSYNC compare la condition d'attente à l'état SAC actuel de la manière suivante : - La condition d'attente est 0 ou contient uniquement des états possibles issus de SysAxCoupleSta[i].State : SACSYNC attend une concordance exacte de la condition d'attente avec SysAxCoupleSta[i].State. - La condition d'attente contient uniquement des états possibles issus de SysAxCoupleSta[i].StateAddOn : SACSYNC attend une concordance des états définis activement dans la condition d'attente avec SysAxCoupleSta[i].StateAddOn. - La condition d'attente contient les états suivants issus de SysAxCoupleSta[i].State et de SysAxCoupleSta[i].StateAddOn : SACSYNC attend la concordance des états définis activement dans la condition d'attente dans la zone StateAddOn avec SysAxCoupleSta[i].StateAddOn et la concordance exacte de la condition d'attente dans la zone State avec SysAxCoupleSta[i].State.

Tab. 6-132: Syntaxe SACSYNC

Exemple

```

; Activer le couplage d'axe XB avec comparaison de la position :
1 XB%=10 ; Index d'axe système 10
1 SD.SysAxCoupleCmd[XB%].SyncMode = 0 ; Coupler précisément sur la position
1 SD.SysAxCoupleCmd[XB%].OnOff = 1 ; Activer le couplage

```

Fonctions CN avec syntaxe de langage standard

```

1 SD.SysAxCoupleCtr.Validate = 1      ; Activer les données
SACSYNC (XB=3)                        ; Attendre le mode synchrone
...                                   ; Traitement
; Désactiver le couplage d'axe XB sans mouvement de comparaison :
1 SD.SysAxCoupleCmd[XB%].SyncMode = 2 ; Compensation de la vitesse
1 SD.SysAxCoupleCmd[XB%].OnOff = 0    ; Désactiver le couplage
1 SD.SysAxCoupleCtr.Valid = 1         ; Activer les données
SACSYNC                               ; Attend l'arrêt et accepte la po-
                                       sition de découplage
    
```

Tab. 6-133: Exemple d'utilisation de SACSYNC

Autres exemples

SACSYNC (XB=0)	; Attend que le couplage de l'axe XB soit désactivé.
SACSYNC (S1=3, SSP11=3)	; Attend jusqu'à ce que la broche du canal S1 et la broche du système SSP11 soient couplées à la valeur de consigne.
SACSYNC (XB=5)	; Attend que le couplage de l'axe XB soit arrêté.
SACSYNC (XB=1000)	; Attend que l'axe XB ait atteint la fenêtre INPOS1. Peu importe s'il a aussi atteint la fenêtre INPOS2.
SACSYNC (XB=3000)	; Attend que l'axe XB ait atteint la fenêtre INPOS1 et la fenêtre INPOS2.
SACSYNC (XB=2003)	; Attend que l'axe XB soit couplé à la valeur de consigne et qu'il ait atteint la fenêtre INPOS2. Peu importe s'il a aussi atteint la fenêtre INPOS1.
SACSYNC (XB=3003)	; Attend que l'axe XB soit couplé à la valeur de consigne et qu'il ait atteint la fenêtre INPOS1 et INPOS2.

Tab. 6-134: Exemples d'autres appels SACSYNC

Particularités et restrictions

- Les axes asynchrones doivent avoir terminé l'interpolation avant de procéder à la programmation d'une condition d'arrêt avec SACSYNC(<AxeAsynchr>=0). D'autres états peuvent également être prédéfinis lorsque l'interpolation est active.
- Si la fonction est interrompue par la remise à zéro du canal ou du système, le décalage de couplage reste actif si l'axe a été découplé avec SyncMode=2 (sans mouvement de compensation).
- Pour accepter les positions de découplage (\$Position de consigne) des axes du canal, même avec la remise à zéro, il est possible d'enregistrer SACSYNC(TO=0) dans la chaîne d'initialisation. Les couplages actifs ne sont pas désactivés lors de la remise à zéro.
- Le décalage de couplage des broches peut uniquement être désactivé par les interpolateurs de broche (M3/M4, M19, commutation SPCC) ou par les commutations du bloc de paramètres d'entraînement (mode d'axe C, changement du rapport de réduction, etc.).
- SACSYNC(...<axis>=<state>...) permet d'attendre l'activation conditionnelle (SwitchCond > 0) des données d'axe maître puisque PENDING (+20) est superposé à l'état de couplage actuel jusqu'à la commutation.
- SACSYNC peut uniquement attendre l'accès aux fenêtres du mode synchrone, mais pas les quitter.

6.106 Aide à la saisie : mise à l'échelle "Scale(...), SCL(...)"

Description : La fonction "Mise à l'échelle" compte parmi les aides à la saisie. La commande augmente ou réduit un contour programmé par rapport à l'origine du programme.

La mise à l'échelle se rapporte toujours au point d'inversion miroir actuel (voir fonction "PoleSet", [chap. 6.83 "Définition du point d'inversion miroir/de rotation PoleSet, PLS" à la page 351](#)). S'il n'a pas été programmé explicitement, le point d'inversion miroir utilisé est le point d'origine actuel du programme.

La fonction "Mise à l'échelle" permet de programmer des contours dans les programmes pièces avec une dimension définie (dimension normalisée). L'échelle du contour programmé est ainsi influencée pour chaque axe par les facteurs de mise à l'échelle avant d'appeler un programme pièce "normalisé" (par ex. en tant que sous-programme).

Ceci permet, par exemple, de compenser facilement les retraits des pièces à usiner lors de la fabrication de pièces moulées ou forgées.



La fonction Mise à l'échelle est une aide à la saisie et ne modifie donc pas le système actuel des coordonnées du programme. Une aide à la saisie n'est en fait qu'une alternative de saisie des coordonnées de programme.

La mise à l'échelle peut également être utilisée conjointement avec les fonctions Miroir et Rotation.

Les points suivants s'appliquent ici :

- La fonction a un effet modal. Elle reste valide jusqu'à ce qu'elle soit désactivée.
- Elle peut être programmée dans le même bloc que d'autres conditions de course et fonctions auxiliaires.

Syntaxe :

Scale (<axe1><facteur>{,...})	
	Activer la mise à l'échelle avec le facteur programmé pour les axes indiqués.
Scale (0) ou Scale ()	Désactiver la mise à l'échelle pour tous les axes du canal. Tous les facteurs de mise à l'échelle sont définis sur la valeur "1". Les positions d'axe atteintes sont conservées jusqu'à ce qu'elles soient reprogrammées.
Format abrégé : SCL (. .)	
avec	

Fonctions CN avec syntaxe de langage standard

<axe 1>	Adresses d'axes (par ex. X) devant être mises à l'échelle.
<facteur>	Programmer l'adresse d'axe avec un facteur positif active la fonction. Cela permet de multiplier en interne tous les ordres de course programmés par la suite pour l'axe correspondant (par ex. X10) par ce facteur : Facteur > 1 : le contour est agrandi. Facteur < 1 : le contour est réduit. Facteur = 1 : le contour reste inchangé. La mise à l'échelle elle-même ne déclenche aucun mouvement de déplacement et ne devient active qu'avec l'information de déplacement suivante.

Tab. 6-135: Syntaxe Scale(...), SCL(...)

Exemple :

:

N30 SCL(X3,Y0.5)

Activer la mise à l'échelle

:

Toutes les coordonnées X programmées ensuite seront multipliées par "3", les coordonnées Y par "0,5".

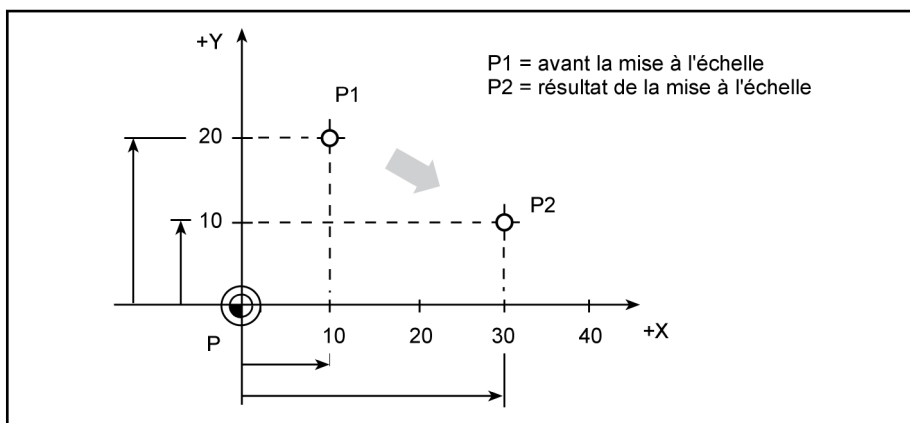


Fig. 6-35: Exemple (...), SCL(...)

Particularités et restrictions :

- Les facteurs de mise à l'échelle négatifs ne sont pas autorisés.
- Pour une interpolation circulaire/hélicoïdale/hélicoïdale N, les facteurs de mise à l'échelle de toutes les coordonnées participantes du plan circulaire doivent être identiques ! Si ce n'est pas le cas, un message d'erreur est généré.
- La fonction agit également sur les paramètres d'interpolation I, J, K et la valeur de l'adresse R (pour la programmation du rayon).
- Elle influence le décalage programmable des segments de contour (Shift).
Voir [chap. 6.110 "Décalage programmé du contour Shift, SHT"](#) à la page 392.
- Elle n'influence **pas** :
 - la programmation de l'avance ou l'avance active,
 - les décalages d'origines (G54 à G59.5 ; voir [chap. 5.2.20 "Décalages d'origine \(DO\) G53, G53.1-G59.1 à G53.5-G59.5"](#) à la page 164),

Fonctions CN avec syntaxe de langage standard

- les décalages des coordonnées du programme (Trans ou ATrans ; voir [chap. 6.144 "Décalage des coordonnées du programme Trans, TRS, décalage additif des coordonnées du programme ATrans, ATR"](#) à la page 437),
- l'initialisation de la position du programme ("SetPos" ; voir [chap. 6.109 "Initialisation de la position du programme SetPos, SPS"](#) à la page 392),
- l'approche des coordonnées du point de référence (G74 ; voir [chap. 5.2.25 "Approche des coordonnées du point de référence G74"](#) à la page 171),
- l'approche de la position fixe des axes machine (G76 ; voir [chap. 5.2.28 "Approche de la position fixe des axes machine G76"](#) à la page 174),
- les valeurs de correction du rayon de la fraise et des longueurs d'outils.

6.107 Couplage sélectif additif de coordonnées

6.107.1 Couplage sélectif additif de coordonnées "SelCrdCouple, SCC"

Description : Couple une coordonnée de la pièce à usiner du canal actuel (cible) à une coordonnée de la pièce à usiner d'un autre canal (source). La valeur de coordonnée de la source s'ajoute donc à la valeur de coordonnée de la cible.

Cela permet de superposer les mouvements programmés dans un autre canal aux mouvements sur le canal actuel.

Syntaxe :

SelCrdCouple(SC<canal>, CL(<S1>,<C1>{{,<Sn>,<Cn>}...}))	
	Couplage sélectif additif de coordonnées ACTIVÉ
SelCrdCouple(0) ou SelCrdCouple()	Tous les couplages de coordonnées actifs DÉSACTIVÉS
Format abrégé : SCC (. .)	
avec	
<canal>	Numéro du canal où se trouve <Sn>. Valeur entrée : Nombre entier <canal> doit être inférieur au numéro de canal du canal actuel.
<Sn>	Source Nom logique ou numéro logique de la coordonnée dans <canal>.
<Cn>	Cible Nom logique ou numéro logique de la coordonnée sur le canal actuel.

Tab. 6-136: Syntaxe SelCrdCouple (SCC)

Fonctions CN avec syntaxe de langage standard

Exemples :	<code>SCC(SC1,CL(YA, YB))</code>	Couple la coordonnée YB du canal actuel à la coordonnée YA du canal 1.
	<code>SCC(SC1,CL(1,1,2,2))</code>	Couple les coordonnées ayant les numéros logiques 1 et 2 sur le canal actuel aux coordonnées ayant les numéros logiques 1 ou 2 sur le canal 1.

6.107.2 Couplage sélectif additif de coordonnées avec table "SelCrdCoupleTab, SCCT"

Description : Couple une coordonnée de la pièce à usiner du canal actuel (cible) à une ou deux coordonnées de la pièce à usiner ou coordonnées machine d'un ou deux autres canaux (sources). Les valeurs de coordonnées des sources s'ajoutent donc à la valeur de coordonnée de la cible. Cela permet de superposer les mouvements programmés dans un ou deux autres canaux aux mouvements sur le canal actuel.

Pour plus d'informations, voir la documentation "Description des fonctions IndraMotion MTX", chapitre "Couplages", section "Couplage sélectif additif de coordonnées avec tables".

Syntaxe :

<code>SelCrdCoupleTab(DE(<nom>,<fact>),SO(<canal>,<nom>,<fact>,<tab>,<décal>),WCS MCS)</code> <code>{,SO2(<canal>,<nom>,<fact>,<tab>,<décal>),WCS MCS)}</code>	
Format abrégé : SCCT	
avec	
DE (DEstination)	
<nom>	Nom de la coordonnée cible
<fact>	Facteur de calibrage de la coordonnée cible (1.0 en l'absence de programmation)
SO (SOurce)	
<canal>	Numéro du canal source
<nom>	Nom de la coordonnée source
<fact>	Facteur de couplage (1.0 en l'absence de programmation)
<tab>	Nom de la table de couplage
<décal>	Valeur de décalage pour l'accès à la table (0.0 en l'absence de programmation)
WCS MCS	Système de coordonnées de la source : WCS = coordonnée de la pièce à usiner MCS = coordonnée de la machine Valeur par défaut : WCS
SO2 (SOurce)	Identique à SO

Tab. 6-137: Syntaxe SelCrdCoupleTab (SCCT)

La fonction SCCT peut être programmée plusieurs fois pour différentes coordonnées.

Fonctions CN avec syntaxe de langage standard

Exemples :	SCCT (DE (YB) , SO (1, YA) , SO2 (1, YC))	La position YB est calculée à partir de la somme des positions YB, YA et YC programmées.
	SCCT (DE (YB, 0) , SO (1, YA))	La position YB est identique à la position YA. La position YB programmée reste sans effet.
	SCCT (DE (ZB) , SO (1, ZA, , SCCTab.fct))	Couple la coordonnée ZA du canal 1 à la coordonnée ZB du canal actif via la table de couplage "SCCTab.fct".
Suppression d'un couplage :		
SCCT ()		
SCCT() ouvre tous les couplages générés avec SCCT(...).		

6.108 Suppression de la surveillance de collision "SCLN"

Description : La fonction SCLN supprime localement la surveillance de collision CLN de la correction du rayon d'outil. Il est alors possible d'usiner des boucles de contour géométriques fermées en elles identifiées en conséquence lorsque la surveillance de collision est active. Si la surveillance de collision rencontre un bloc identifié avec SCLN, alors la prévisualisation de son bloc prend fin.

Syntaxe :	SCLN [(<mode>)]	
	avec	
	<mode>	1 : La prévisualisation des blocs de la surveillance de collision est interrompue sur le bloc identifié. 2 : Pour identifier un bloc d'approche sur un contour fermé : la prévisualisation des blocs de la surveillance de collision est interrompue sur le bloc qui entre en collision avec le bloc d'approche. Valeur par défaut : 1

Tab. 6-138: Syntaxe SCLN

Exemple :	N100 CLN (DLA10)	Surveillance de collision avec prévisualisation de 10 blocs-programmes.
	N110 G0 X4.4645 Y-5.5355	Positionner en position de départ.
	N120 G1 F750 Z5	Approcher l'axe d'outil
	N130 G1 F1250 G42 ED1 X6.4645 Y-5.5355	Activer la correction du rayon d'outil
	N140 G2 R5 X5 Y0 SCLN (2)	Cercle d'approche tangentiel sur le contour. Collision avec N190. SCLN(2) a pour effet d'interrompre la surveillance de collision dans le bloc N190.
	N150 G1 X10 Y0	Début du contour
	N160 X10 Y10	
	N170 X0 Y10	
	N180 X0 Y0	
	N190 X6 Y0	Fin du contour
	N200 G2 R5 X8.5355 Y-3.5355	Abandon tangentiel du contour
	N210 G40 G1 X10.85355 Y-5.5355	Désactivation de la correction du rayon d'outil

Particularités et restrictions : La fonction SCLN agit au sein du bloc.

6.109 Initialisation de la position du programme "SetPos, SPS"

Description : Initialise l'origine actuelle du programme (par rapport au système actuel des coordonnées du programme et à l'origine active) sur la valeur programmée respectivement sans déclencher de mouvements d'axes.

Ensuite, les nouvelles valeurs de position sont affichées automatiquement.

Syntaxe :	SetPos (<coordonnées>)	Définir le point d'origine du programme pour les axes programmés sous <coordonnées>.
	SetPos	Supprimer tous les décalages déclenchés par SetPos.
	Format abrégé : SPS (. .)	
	avec	
	<coordonnées>	Placer les valeurs d'axe aux coordonnées indiquées. Les coordonnées indiquées doivent être séparées par des virgules (par ex. : (X0, Y0)).

Tab. 6-139: Syntaxe SetPos, SPS

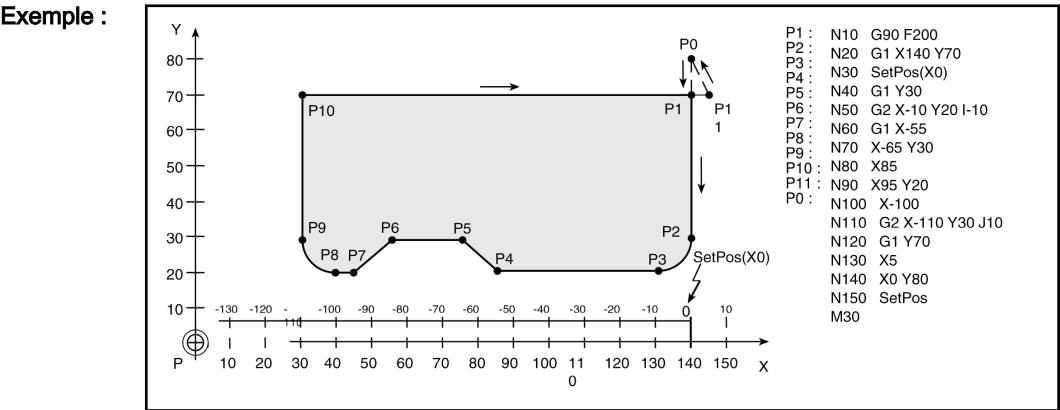


Fig. 6-36: Exemple : Initialisation de la position du programme

Particularités et restrictions : Les paramètres machine permettent de paramétrer pour chaque canal si les décalages SetPos doivent être conservés ou supprimés après une remise à zéro.

6.110 Décalage programmé du contour "Shift, SHT"

Description : La fonction "Shift" compte parmi les aides à la saisie.

La commande décale un contour programmé parallèlement aux axes du système de coordonnées du programme.

"Shift" est une aide à la saisie et ne modifie donc pas le système actuel des coordonnées du programme. Une aide à la saisie n'est en fait qu'une alternative de saisie des coordonnées de programme.

Les points suivants s'appliquent ici :

- Les coordonnées de décalage programmées restent effectives jusqu'à elles sont désactivées ou écrasées par un nouveau bloc Shift.
- Si la fonction "Shift" est programmée seule, elle n'entraîne aucun mouvement de déplacement, mais il est possible de programmer des informations de déplacement dans le même bloc.

Fonctions CN avec syntaxe de langage standard



Une programmation incorrecte peut entraîner un endommagement de la pièce à usiner et de la machine !

La fonction "Shift" est influencée par les fonctions Miroir, Mise à l'échelle et Rotation, c.-à-d. que les coordonnées de la nouvelle origine du contour programmées dans un bloc de décalage sont également inversées, mises à l'échelle et tournées !

Syntaxe :

Shift (<coordonnées>)	Activer le décalage de contour
Shift (0) ou Shift ()	Désactiver le décalage de contour
Format abrégé : SHT (. .)	
avec	
<coordonnées>	Coordonnées du programme de l'origine décalée du contour. Si plusieurs coordonnées sont indiquées, elles doivent être séparées par des virgules (par ex. : SHT (X5, Y2)).

Tab. 6-140: Syntaxe Shift (SHT)

Exemple :

N10 SHT (X10, Y10, Z50)	Nouvelle origine du contour à X10, Y10, Z50
:	Aucun déplacement des axes
N100 G1 X...Y...Z...	Déplacement des axes, compte tenu du décalage.
:	
N110 SHT (X20, Y20)	Nouvelle origine du contour à X20, Y20, Z50 (le décalage de Z est conservé !)
:	Aucun déplacement des axes
N210 SHT ()	Désactiver le décalage ; les axes se déplacent sur la position programmée.
X...Y...Z...	
:	

6.111 Limitation de la vitesse de rotation "SMin, SMN", "SMax, SMX"

Description : Définit la plage de vitesse de rotation dans laquelle la vitesse de rotation de la broche peut se déplacer avec la programmation directe de la vitesse de rotation G97 et une vitesse de coupe constante G96 (voir [chap. 5.2.41 "Vitesse de coupe constante G96, programmation directe de la vitesse de rotation G97" à la page 190](#)) au cours d'une section d'usinage. La plage de vitesse de rotation définie s'applique à tous les rapports de réduction. Lorsque la limitation de la vitesse de rotation est active, toutes les affectations de vitesses de rotation sont limitées aux valeurs limites programmées.

Le système n'applique les modifications de la vitesse de rotation (même celles provoquées par le potentiomètre de la broche) que lorsqu'elles se situent au sein de la plage de vitesse de rotation définie.

Syntaxe :

SMin (S<vitesse de rotation1>)	Active la <vitesse de rotation1> en tant que limite inférieure de la plage de vitesse de rotation admissible.
SMin (S0)	Supprime la limite inférieure de la première broche ou du premier groupe de broches.

Fonctions CN avec syntaxe de langage standard

Format abrégé : SMN (. .)	
SMax (S<vitesse de rotation2>)	Active la <vitesse de rotation2> en tant que limite supérieure de la plage de vitesse de rotation admissible.
SMax (S-1)	Supprime la limite supérieure de la première broche ou du premier groupe de broches.
Format abrégé : SMX (. .)	
avec	
<vitesse de rotation1>	Vitesse de rotation minimale admissible Plage de valeurs : > 0. Doit être inférieure à <vitesse de rotation2>
<vitesse de rotation2>	Vitesse de rotation maximale admissible Plage de valeurs : > 0. Doit être supérieure à <vitesse de rotation1>

Tab. 6-141: Syntaxe SMin, SMN, SMax et SMX

La syntaxe générale est la suivante :

SMin(S1=Valeur1 {,S2=Valeur2} ...) ou SMax(S1=Valeur1 {,S2=Valeur2} ...).

Exemple :	N50 X.. Y.. SMN(S1500)	La vitesse de rotation de la broche doit se situer dans une plage comprise entre 1500 et 2500 tr/min.
	N60 X.. Y.. SMX(S2500)	
	:	
	N90 X.. Y.. SMN(S0) SMX(S-1)	Désactiver les limites inférieure et supérieure (= limitation de la vitesse de rotation désactivée)

Particularités et restrictions : La limitation de la vitesse de rotation n'agit que si les limites de vitesse de rotation programmées sont comprises dans les limites des rapports de réduction.

6.112

Comparaison des broches dans le programme de comparaison d'initialisation de l'avance de bloc "SpAdjSp"

Description :	 Pour de plus amples informations sur la fonctionnalité "Avance de bloc", voir la documentation "Description des fonctions IndraMotion MTX".
	La fonction "SPADJSP" crée les broches logiques du canal lors de l'appel dans le premier programme de comparaison de l'avance de bloc.
Syntaxe :	SpAdjSp
Exemple :	Tab. 6-142: Syntaxe SpAdjSp
	Extrait d'une InitAdjPrg <pre>; création des broches du canal et groupes de broches ; envoie aussi les fonctions aux. CSPx à l'API ! N100 SpAdjSp ;</pre>

- Particularités et restrictions :**
- La fonction "SpAdjSp" peut être utilisée uniquement dans un programme de comparaison d'initialisation (voir SD.SysSRun.NameOfInitAdjPrg).
 - Si les fonctions auxiliaires en nombre entier spécifiques au canal CSP1 à CSP8 sont définies, l'API est informé via l'affectation actuelle de la broche du canal, comme pour GetSpindle, RemSpindle et DefSpindle.

6.113 Détermination de la syntaxe de broche pour le programme de comparaison de l'avance de bloc "SpAdjStr"

Description :



Pour de plus amples informations sur la fonctionnalité "Avance de bloc", voir la documentation "Description des fonctions IndraMotion MTX".

La fonction "SPADJSTR" <numéro broche> renseigne les composantes suivantes de la donnée système "SD.SysChSpSRun[]" lors de l'appel d'un programme de comparaison :

- SpeedStr
- MoveStr
- GearStr

Après l'appel, les variables SpeedStr, MoveStr et GearStr contiennent une fonction d'aide pertinente pour la broche programmée à la fin de l'avance du bloc.

Syntaxe :

SpAdjStr<broche>	
Les composantes de la donnée système sont renseignées avec les chaînes valides à la fin de l'avance de bloc (par ex. S1=, M103, M141).	
avec	
<broche>	Numéro de la broche (du canal) 1 à 8

Tab. 6-143: Syntaxe SpAdjStr

Exemple :

Extrait d'une InitAdjPrg

```
;
;   prédéfinir la vitesse de rotation ou de coupe
;
0000 DIM SADR$(8) :REM pour la syntaxe de broche      (p. ex. S1=)
;
0100 FOR SP% = 1 TO SD.SysChSpSRun.MaxSpNumber
0102   IF SD.SysChSpSRun.Sp[SP%].SpeedNotAdj <> 0 THEN
;
N104     SpAdjStr[SP%] ;charger les éléments de syntaxe de la broche actuel.
0106     SADR$ = SD.SysChSpSRun.SpeedStr
;
N108     [SADR$] [SD.SysChSpSRun.Sp[SP%].ProgSpeed]
0110     ENDIF
0112 NEXT SP%
;
```

Particularités et restrictions :

- La fonction SpAdjStr peut être utilisée uniquement dans un programme de comparaison (voir SD.SysSRun.NameOfInitAdjPrg et SD.SysSRun.NameOfAsup).
- Lors de l'appel de "SpAdjStr", la configuration de la broche doit être comparée (voir "SpAdjSp").

Fonctions CN avec syntaxe de langage standard

6.114 Libération/Reprise d'une broche réservée "SpAdmin, SPA"

Description : *Autorise :*

- la libération d'une broche actuellement réservée sur le canal actuel, sans arrêter une broche en cours de rotation ;
- puis son intégration dans un canal quelconque.



La fonction "Arrêt de la broche" sert également à libérer une broche (voir [chap. 5.3.6 "Arrêt de la broche M5, M105, M205" à la page 207](#)).

Syntaxe :

SpAdmin (S<num>=<mode>{ , S<num>=<mode>}...)

Format abrégé : SPA (...)

avec

<num>	Numéro de la broche (index de la broche)
	Champ de saisie : 1 à 8
	Nombre entier
<mode>	0 : libérer la broche
	1 : réserver et reprendre la broche libérée

Tab. 6-144: Syntaxe SpAdmin (SPA)

Exemple :

N60 SPA (S1=0, S2=0) Libérer la 1re et la 2e broches
:

Particularités et restrictions :

La remise à zéro ou M30 ne libèrent les broches réservées par le canal actuel que lorsque l'ordre "Arrêt de la broche" est inscrit à l'endroit adéquat dans MP 7060 00020.

6.115 Définition (activation) d'un groupe de couplage, dissociation (désactivation) d'un groupe de couplage "SpCoupleConfig, SPCC"



Vous trouverez de plus amples informations dans la documentation "Description des fonctions de base IndraMotion MTX", chapitre "Fonctionnement à synchronisation d'orientation des broches".

Description :

- Définit un groupe de couplage et active le couplage de ses broches. Pour cela, la commande des entraînements de broches participants passe automatiquement à l'interface de position.
- Ajoute des broches esclaves à un groupe de couplage existant, ou les retire d'un groupe de couplage existant.
Les entraînements de broches ajoutés sont automatiquement commutés à l'interface de position, les entraînements de broches retirés à l'interface de vitesse de rotation (dans la mesure où l'interface de vitesse de rotation a été activée avant le couplage pour les broches concernées).
- Définit le sens de rotation des broches esclave à coupler par rapport à la broche maître.

Fonctions CN avec syntaxe de langage standard

- Désactive le couplage des broches d'un groupe de couplage et dissocie le groupe de couplage complet.

Tous les entraînements de broches participants sont automatiquement commutés à l'interface de vitesse de rotation, dans la mesure où l'interface de vitesse de rotation a été activée avant le couplage pour les broches concernées.

La règle de couplage pour la position de la broche esclave est la suivante :

$$p_s = p_m * k + o_{dist} + o_{offs}$$

avec

p_s	Position de consigne de la broche esclave
p_m	Position de consigne de la broche maître
k	Sens de rotation de la broche esclave par rapport à la broche maître (1, -1)
o_{dist}	Écart de couplage prédéfini entre les broches maître et esclave (SPCD)
o_{offs}	Décalage dynamique s'ajoutant à l'écart de couplage (SPCP)

Syntaxe :

Définir le groupe de couplage	
SpCoupleConfig (CP=<groupe>, MA=<maître>, S<esclave>=1 -1 {{,S<esclave>=1 -1}...})	
Ajouter/Retirer des broches esclaves au/du groupe de couplage	
SpCoupleConfig (CP=<groupe>, S<esclave>=<mode> {{,S<esclave>=<mode>}...})	
Dissocier un groupe de couplage	
SpCoupleConfig (CP=<groupe>, MA=0)	
Format abrégé : SPCC (...)	
avec	
<groupe>	Numéro du groupe de couplage Champ de saisie : 1 à 4 Nombre entier
<maître>	Numéro de la broche maître (index de la broche) Champ de saisie : 1 à 8 Nombre entier
<esclave>	Numéro de la broche esclave (index de la broche) Champ de saisie : 1 à 8 Nombre entier
<mode>	0 : Retirer la broche esclave du <groupe> 1 : Ajouter la broche esclave au <groupe> dans le sens de rotation de la broche maître -1 : Ajouter la broche esclave au <groupe> dans le sens de rotation inverse de la broche maître

Tab. 6-145: Syntaxe SpCoupleConfig (SPCC)

Exemple :

Fonctions CN avec syntaxe de langage standard

CoupleAndDecouple

```
N010 SPCD(S3=0)                ;coupling distance 0°
...
N210 SPCC(CP=1, MA=1, S3=1) ;activate coupling 1
N220 SPC_WAIT(CP=1)           ;wait for synchronous mode
...
...
N310 M103 S1=2000
...
...
N760 SPCC(CP=1, MA=0)         ;deactivate coupling 1
N770 SPCC_WAIT(CP=1)         ;wait for end of decoupling
...
```

6.116 Écart de coulage de la broche esclave "SpCoupleDist, SPCD"

Description : Configure la différence de position entre la broche maître et la broche esclave au moment du couplage (lors de l'établissement du couplage) pour la fonctionnalité "Couplage de broches".

Lorsque la fonction n'est pas utilisée pour certaine une broche esclave, la différence de position par rapport à la broche maître est alors de 0 degré (au moment du couplage).

 Pour amorcer un décalage angulaire additif entre la broche maître et la broche esclave au cours d'un couplage de broches actif, voir la fonction "SpCouplePosOffs", [chap. 6.118 "Décalage angulaire avec un couplage actif SpCouplePosOffs, SPCP"](#) à la page 399.

Syntaxe :		SpCoupleDist (S<num>=<écart>{, S<num>=<écart>}...)
		Format abrégé : SPCD (...)
		avec
<num>		Numéro de la broche esclave (index de la broche) Champ de saisie : 1 à 8 Nombre entier
<écart>		Écart de couplage de la broche esclave : - Différence de position entre la broche maître et la broche esclave en degrés. Plage de valeurs : $0^{\circ} \leq \text{différence de position} < 360^{\circ}$ Si une autre valeur est programmée, elle est automatiquement convertie dans l'intervalle indiqué. ACT pour un couplage sur la position actuelle. En cas d'arrêt de la broche maître et de la broche esclave, cela permet d'éviter un mouvement de compensation perturbateur.

Tab. 6-146: Syntaxe SpCoupleDist (SPCD)

Exemple Voir "Définition (activation) et dissociation (désactivation) d'un groupe de couplage".

Particularités et restrictions : La fonction peut uniquement être programmée avec des broches qui ne font partie d'aucun groupe de couplage à ce moment.

6.117 Fenêtre du mode synchrone "SpCoupleErrWin, SPCE"

Description : Configure l'écart de position maximal autorisé entre la valeur de consigne et la valeur réelle d'une broche esclave pour la fonctionnalité "Couplage de broches".

Si la différence de position au cours d'un couplage actif se situe dans l'intervalle défini, alors le signal de sortie spécifique à la broche "**Marche synchrone 2**" est émis.

Lorsque la fonction n'est pas utilisée pour une certaine broche esclave, sa fenêtre d'erreur de marche synchrone est de +/- 10 degrés.



Voir aussi la fonction "SpCoupleSyncWin", chap. 6.120 "Fenêtre du mode synchrone SpCoupleSyncWin, SPCS" à la page 401.

Syntaxe :

SpCoupleErrWin(S<num>=<fenêtre>{, S<num>=<fenêtre>}...)	
Format abrégé : SPCE (...)	
avec	
<num>	Numéro de la broche esclave (index de la broche) Champ de saisie : 1 à 8 Nombre entier
<fenêtre>	Différence de position maximale admissible par rapport à la valeur de consigne en degrés. Champ de saisie : 0 à 359.9999

Tab. 6-147: Syntaxe SpCoupleErrWin (SPCE)

Exemple Voir "Fenêtre du mode synchrone "SpCoupleSyncWin, SPCS".

Particularités et restrictions : La fonction peut uniquement être programmée avec des broches qui ne font partie d'aucun groupe de couplage à ce moment.

6.118 Décalage angulaire avec un couplage actif "SpCouplePosOffs, SPCP"

Description : Amorce un décalage angulaire entre la broche maître et la broche esclave quand un couplage de broches est actif. La vitesse de rotation relative entre la broche maître et la broche esclave à laquelle s'applique le décalage peut être programmée en option.



Pendant la durée de cette rotation, la commande réinitialise le signal de sortie spécifique à la broche "Marche synchrone 1".

Syntaxe :

SpCouplePosOffs(S<num>=<décalage>{{, S<num>=<décalage>}...} {, POSVEL<vitesse de rotation>})	
Format abrégé : SPCP (...)	
avec	
<num>	Numéro de la broche esclave (index de la broche) Champ de saisie : 1 à 8 Nombre entier

Fonctions CN avec syntaxe de langage standard

<décalage>	Angle de rotation absolu entre la broche maître et la broche esclave en degrés. Plage de valeurs : -3600° à +3600°
<vitesse de rotation>	Vitesse de rotation relative entre la broche maître et la broche esclave à laquelle s'applique le décalage. L'unité et la valeur par défaut correspondent au paramètre d'entraînement S-0-0222. Une fois indiquée, la <vitesse de rotation> reste enregistrée en interne jusqu'à ce qu'elle soit modifiée par un nouvel appel de fonction.

Tab. 6-148: Syntaxe SpCouplePosOffs (SPCP)

Exemple : AngularOffset

<pre>N010 SPCD(S3=0) ;coupling distance 0° ... N210 SPCC(CP=1, MA=1, S3=1) ;activate coupling 1 N220 SPC_WAIT(CP=1) ;wait for synchronous mode N310 M103 S1=2000 N410 SPCP(POSVEL20, S3=195) ;rotate S3 in relation to S1 N420 SPCP_WAIT(CP=1) ;wait for end of rotation ...</pre>	

Particularités et restrictions : Le décalage angulaire s'ajoute à un écart de couplage éventuellement configuré (voir chap. 6.116 "Écart de coulage de la broche esclave SpCoupleDist, SPCD" à la page 398).

6.119 Attente d'un décalage angulaire "SpCouplePosOffs_Wait, SPCP_WAIT"

Description : Arrête le programme pièce jusqu'à ce que le décalage angulaire programmé d'un groupe de couplage (SPCP ; voir chap. 6.118 "Décalage angulaire avec un couplage actif SpCouplePosOffs, SPCP" à la page 399) soit effectivement amorcé sur la machine.

Syntaxe : SpCouplePosOffs_Wait (CP=<groupe>)	
Format abrégé : SPCP_WAIT (...)	
avec	
<groupe>	Numéro du groupe de couplage Champ de saisie : 1 à 4 Nombre entier

Tab. 6-149: Syntaxe SpCouplePosOffs_Wait (SPCP_WAIT)

Exemple Voir "Décalage angulaire avec un couplage actif "SpCouplePosOffs, SPCP"".

6.120 Fenêtre du mode synchrone "SpCoupleSyncWin, SPCS"

Description : Configure l'écart de position maximal autorisé entre la valeur de consigne et la valeur réelle d'une broche esclave pour la fonctionnalité "Couplage de broches" :

- Mode d'action lors de la création/modification d'un couplage (au début de la phase de synchronisation) :
Le programme pièce attend jusqu'à ce que la différence de position se situe dans l'intervalle défini.
- Mode d'action lors du couplage :
Si la différence de position se situe dans l'intervalle défini, alors le signal de sortie spécifique à la broche "**Marche synchrone 1**" est émis.

Lorsque la fonction n'est pas utilisée pour une certaine broche esclave, sa fenêtre de marche synchrone est de +/- 1 degré.



Pour une surveillance supplémentaire avec un couplage actif, voir la fonction "SpCoupleErrWin", [chap. 6.117 "Fenêtre du mode synchrone SpCoupleErrWin, SPCE"](#) à la page 399.

Syntaxe :

SpCoupleSyncWin (S<num>=<fenêtre>{, S<num>=<fenêtre>}...)	
Format abrégé : SPCS (...)	
avec	
<num>	Numéro de la broche esclave (index de la broche) Champ de saisie : 1 à 8 Nombre entier
<fenêtre>	Différence de position maximale admissible par rapport à la valeur de consigne en degrés. Champ de saisie : 0 à 20

Tab. 6-150: Syntaxe SpCoupleSyncWin (SPCS)

Exemple :

ConfigureAndCouple

```
;Configure slave spindle
N010 SPCD(S3=0)           ;coupling distance 0°
N020 SPCS(S3=0.5)         ;synchronous window 0.5°
N030 SPCE(S3=3.0)         ;error window 3°
...
N210 SPCC(CP=1, MA=1, S3=1) ;activate coupling
N220 SPC_WAIT(CP=1)       ;wait for synchronous mode
...
...
```

Particularités et restrictions :

La fonction peut uniquement être programmée avec des broches qui ne font partie d'aucun groupe de couplage à ce moment.

6.121 Attente du mode synchrone "SpCouple_Wait, SPC_WAIT"

Description : Arrête le programme pièce jusqu'à ce qu'un groupe de couplage ait été créé, reconfiguré ou dissocié avec succès (SPCC ; voir [chap. 6.115 "Définition \(activation\) d'un groupe de couplage, dissociation \(désactivation\) d'un groupe de couplage SpCoupleConfig, SPCC"](#) à la page 396).

Syntaxe :

SpCouple_Wait (CP=<groupe>)
Format abrégé : SPC_WAIT (...)

Fonctions CN avec syntaxe de langage standard

avec	
<groupe>	Numéro du groupe de couplage Champ de saisie : 1 à 4 Nombre entier

Tab. 6-151: Syntaxe SpCouple_Wait (SPC_WAIT)

Exemple Voir "Définition (activation) et dissociation (désactivation) d'un groupe de couplage".

6.122 Définition/Dissociation de groupes de broches "SPG.., SPGALL"

Description : Si la fonction "Broches du canal" est activée (MP 1040 00999 = 1), alors les groupes de broches sont automatiquement préréglés lors de la montée en régime de la commande.

La première broche du canal est membre du premier groupe de broches.

Les groupes de broches (aussi appelés broches parallèles) permettent de programmer simplement plusieurs broches avec les fonctions pour les "rapports de réduction", la "marche à droite/à gauche", l'"arrêt de la broche" et l'"orientation de la broche". Cela évite de programmer les broches une à une.



La fonctionnalité "Groupe de broches" ne doit **pas** être confondue avec un "Groupe de couplage" d'au moins deux broches asservies en position.

Tandis que toutes les broches d'un groupe de broches peuvent tourner à différentes vitesses de rotation, les broches d'un groupe de couplage fonctionnent en principe de manière synchrone.

Syntaxe :

SPG<groupe>(<numéros>)	Définit quelles broches du canal doivent être regroupées dans un groupe de broches sur le canal actuel.
SPG<groupe>(0)	Restaure les paramètres par défaut du groupe sur le canal actuel.
SPG<groupe>(-1)	Dissocie le groupe de broches correspondant sur le canal actuel.
SPGALL(0)	Restaure les paramètres par défaut de tous les groupes sur le canal actuel.
avec	
<groupe>	Numéro du groupe de broches Champ de saisie : 1 à 4 Nombre entier
<numéros>	Numéros de toutes les broches du canal devant être attribuées au groupe de broches correspondant. Les numéros sont séparés par des virgules. Champ de saisie : 1 à 8
SPG (<numéros>) est un format abrégé de SPG1(<numéros>)	

Tab. 6-152: Syntaxe SPG.. (SPGALL)

Fonctions CN avec syntaxe de langage standard

Exemple :	
GSP (SSP01, S1, SSP07, S2)	Les broches du canal 1 et 2 sont attribuées aux broches du système 1 et 7.
SPG1 (1, 2)	Le groupe de broches 1 se compose de S1 et S2.
M3	Démarrer S1 et S2 (SSP01 et SSP07) avec une marche à droite.
GSP (SSP14, S2)	La broche du système 14 est attribuée à la broche 2.
M4	Démarrer S1 et S2 (SSP01 et SSP14) avec une marche à gauche.
RSP (S1)	Aucune broche du système n'est plus attribuée à la broche du canal 1 (S1).
M3	Démarrer S2 (SSP14) avec une marche à droite.

- Particularités et restrictions :**
- En cas de transfert d'une broche (GSP, RSP, DSP), l'attribution passe de la broche du canal à la broche du système. Ce changement d'attribution a un effet direct sur le groupe de broches.
 - Les ordres concurrentiels entre les broches individuelles et les groupes de broches programmés dans un même bloc génèrent une erreur d'exécution (exemple : N10 M3 M104).

6.123 Écriture dans une queue de données système "SSDQ"

Description : La fonction "SSDQ" écrit un message programmé (un nombre entier) dans une queue dans les données système (un champ de données système "Int_t") au **moment de l'exécution**. La queue est alors utilisée comme mémoire tampon circulaire. Cela signifie que, si le dernier élément du champ de données système est décrit, la fonction saisit le message programmé dans le premier élément de la queue lors de l'appel suivant. Dans tous les autres cas, l'élément suivant est décrit.

Pour que l'utilisateur reçoive une information concernant les éléments de queue qui ont été décrits, la fonction "SSDQ" modifie simultanément une structure de la commande. Cette structure de commande se trouve aussi dans les données système et se compose des deux éléments "Int_t" **LastFilled** et **LastGet**.

Si la fonction a inscrit un message programmé dans la queue, l'élément **LastFilled** est défini sur l'index de l'entrée modifiée dans la queue. Cet index est basé sur un.

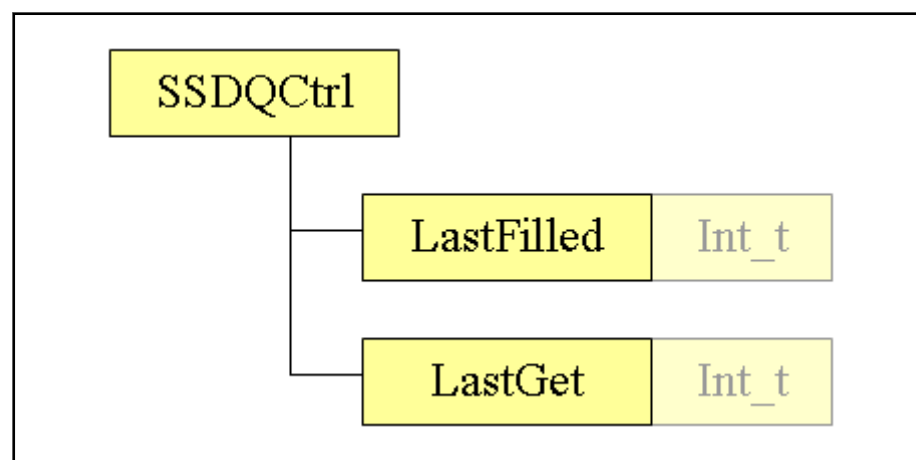


Fig. 6-37: Composition de la structure de commande de SSDQ

Fonctions CN avec syntaxe de langage standard

Le programmeur peut aussi bien prédéfinir la queue des données système que la structure de commande utilisée. Pour cela, il prédéfini les structures correspondantes comme XPath. Les deux structures des données système peuvent être des éléments racine ou faire partie d'une structure plus grande.



Il est aussi possible de créer des structures en fonction du canal. Si l'index du canal est omis avec une telle constellation, la fonction utilise automatiquement la structure partielle du canal actuel (dans lequel le programme pièce est exécuté).

Si les structures des données système ne sont pas prédéfinies explicitement, la fonction essaie d'utiliser "/SSDQueue" comme queue et "/SSDQCtrl" comme structure de commande.

Dans le cas d'une action d'écriture, un événement de modification est uniquement déclenché pour la structure de commande. La queue est certes modifiée, mais aucun événement de changement n'est déclenché.

Une fois la fonction "SSDQ" exécutée, l'utilisateur est responsable de la lecture du message à partir de la queue. Il doit alors définir dans la structure de commande l'élément **LastGet** sur l'index de l'élément de queue lu en dernier.

Si l'utilisateur lit la structure de commande et que les valeurs de **LastFilled** et **LastGet** diffèrent, cela signifie que de nouveaux messages sont apparus dans la queue. Si **LastFilled** est supérieure à **LastGet**, cela signifie que les nouveaux messages se trouvent dans les entrées de la queue de **LastGet+1** à **LastFilled** (dans l'ordre dans lequel ils ont été programmés). Si **LastFilled** est inférieure à **LastGet**, cela est dû au comportement de la mémoire tampon circulaire de la queue. Dans ce cas, les nouveaux messages se trouvent dans **LastGet+1** jusqu'à la fin de la queue ainsi que du début de la queue jusqu'à **LastFilled** (dans l'ordre dans lequel ils ont été programmés).

Si l'utilisateur ne parvient pas à définir **LastGet** à temps, il se peut que la queue soit pleine. Des messages seront alors perdus. C'est pourquoi la fonction "SSDQ" vérifie si d'autres éléments libres sont présents dans la queue (position d'écriture suivante différente de **LastGet**). Si ce n'est pas le cas, une erreur d'exécution est générée.

La fonction "SSDQInit" permet d'initialiser une structure de commande. Lors de la programmation de "SSDQInit", aussi bien **LastGet** que **LastFilled** sont définis sur 1 au moment de l'exécution.

Syntaxe :

SSDQ (M <message> {, Q <XPath>} {, C <XPath>})	
avec	
M <message>	Nombre entier devant être inscrit dans la queue.
Q <XPath>	XPath sur la donnée système ou la structure partielle qui est utilisée comme queue. Donnée : "/SSDQueue"
C <XPath>	XPath sur la donnée système ou la structure partielle qui est utilisée comme structure de commande. Donnée : "/SSDQCtrl"

Fonctions CN avec syntaxe de langage standard

SSDQInit ({C <XPath> })	
C <XPath>	XPath sur la donnée système ou la structure partielle qui doit être initialisée comme structure de commande. Donnée : "/SSDQCtrl"

Tab. 6-153: Syntaxe SSDQ et SSDQInit

Exemple : Pour SSDQ et SSDQInit

N10 SSDQ(M32)	Écriture du message "32" dans la queue "/SSDQueue" avec la structure de commande "/SSDQCtrl"
N20 SSDQ(M=17, Q / myQueue)	Écriture du message "17" dans la queue "/myQueue" avec la structure de commande "/SSDQCtrl"
N30 SSDQ(M 42, C=/ myCtrl)	Écriture du message "42" dans la queue "/SSDQueue" avec la structure de commande "/myCtrl"
40 ST\$="/Msg[1,,3]/ Queue"	Écriture du message dans I% dans la queue "/Msg[1,KANALNR,3]/Queue" avec la structure de commande "/SSDQCtrl"
N50 SSDQ(M[I%], Q=[ST\$])	Si ce programme est traité sur le canal 5, par exemple, alors la fonction écrit dans une queue "/Msg[1,5,3]/Queue".
N60 SSDQInit	Initialisation de la structure de commande "/SSDQCtrl" avec LastFilled=1 et LastGet=1
N70 SSDQInit(C=/ Msg[1,,2]/Ctrl	Initialisation de la structure de commande "/Msg[1,KANALNR,2]/Ctrl" avec LastFilled = 1 et LastGet = 1 Si ce bloc est traité sur le canal 4, par exemple, alors la fonction écrit dans la structure de commande "/Msg[1,4,3]/Ctrl".

- Particularités et restrictions :
- La queue doit être au moins aussi grande que 3 éléments.
 - La taille de la queue du message dépend de la mémoire libre de la cible.
 - Si l'exécution du mouvement programmé d'un bloc avec SSDQ dure moins longtemps qu'un cycle IPO, cela entraîne une "secousse".
 - Étant donné que la communication vers Windows via NCS a une priorité moindre, cela peut entraîner des temporisations lors de la lecture des messages.

6.124 Programmation des broches du système "SSp..."

6.124.1 Généralités

Les fonctions suivantes permettent de piloter les broches du système. Ces fonctions complètent le pilotage traditionnel des broches via des fonctions auxiliaires.

L'abréviation **SSp** vient de l'anglais « **S**ystem **S**pindle », c'est-à-dire broche du système.

Fonctions CN avec syntaxe de langage standard

6.124.2 Libération/Reprise d'une broche réservée "SSpAdm"

Description : Libère du canal actuel une ou plusieurs broches du système actuellement réservées par des instructions de mouvement ou d'orientation. Les broches libérées peuvent être reprises sur le canal actuel.

Cette fonction ne n'existe que pour des raisons de symétrie. Il est recommandé d'utiliser la fonction du canal "SpAdmin".

Syntaxe :

SSpAdm, SSPADM	
SSpAdm (<NoBrSys> <NomBrSys>, <mode>{ , <NoBrSys> <NomBrSys>, <mode>} ...)	
avec	
<NoBrSys>	1 à 32
<NomBrSys>	Nom de la broche du système
<mode>	1 reprend une broche libérée. 0 libère une broche sous certaines conditions.

Tab. 6-154: Syntaxe SSpAdm

Exemple :

SSpAdm (SSP05, 0, SSP08, 1) Libère la broche du système 5 et reprend la broche du système 8 libérée.

6.124.3 Désactivation de la gestion de mouvement des broches pour certains canaux "SSpAdmOff"

Description : Désactive la gestion de mouvement d'une broche du système pour certains canaux. Cette fonction a le même effet que le paramètre machine 1040 00070 "Gestion des broches".

Syntaxe :

SSpAdmOff, SSPADMOFF	
SSpAdmOff (<NoBrSys> <NomBrSys>, <NoCanal> {, <NoCanal> } ...)	
SSpAdmOff (<NoBrSys> <NomBrSys>)	
avec	
<NoBrSys>	1 à 32
<NomBrSys>	Nom de la broche du système
<NoCanal>	1 à 60 : numéros des canaux 0 : libération pour les ordres de l'interface bit

Tab. 6-155: Syntaxe SSpAdmOff

Exemple :

SSpAdmOff (SSP05, 1, 2, 4) Désactive la gestion de mouvement de la broche du système 5 pour les canaux 1, 2 et 4.
SSpAdmOff (5) Restaure l'état du MP 1040 00070 pour la gestion de mouvement pour la broche du système 5.

- Particularités et restrictions :**
- Contrairement au schéma applicable aux autres fonctions "SSp", cette fonction ne peut influencer qu'une seule broche du système à la fois.
 - Un pré réglage du canal 0 désactive la gestion de mouvement de cette broche du système pour les pré réglages de l'interface bit.

6.124.4 Changement du rapport de réduction "SSpGear"

Description : Change le rapport de réduction d'une ou plusieurs broches du système.

Fonctions CN avec syntaxe de langage standard

Syntaxe :

SSpGear, SSPGEAR	
SSpGear (<NoBrSys> <NomBrSys>, <rapport>{ , <NoBrSys> <NomBrSys>, <rapport>} ...)	
avec	
<NoBrSys>	1 à 32
<NomBrSys>	Nom de la broche du système
<rapport>	1 à 4 pour les rapports de réduction 1 à 4 8 pour la position neutre

Tab. 6-156: Syntaxe SSpGear

Exemple :

SSpGear (SSP05, 4) Change le 4e rapport de réduction pour la broche du système 5.

Particularités et restrictions :

- La commande numérique attend l'exécution du changement de rapport de réduction.
- La sélection automatique du rapport de réduction est une caractéristique propre au canal. C'est pourquoi il est impossible de commuter entre les modes Automatique et Manuel avec "SSpGear".
- Une sélection automatique du rapport de réduction active dans un canal n'est pas affectée par "SSpGear". Cela signifie que, si "M40" est actif, la programmation suivante du mot S exécute éventuellement un changement du rapport de réduction.
- La commutation des rapports de réduction est uniquement prise en charge pour les broches Sercos.

6.124.5 Limitation de la vitesse de rotation maximale "SSpMax"

Description :

Limite la vitesse de rotation maximale d'une ou plusieurs broches du système.

Syntaxe :

SSpMax, SSPMAX	
SSpMax (<NoBrSys> <NomBrSys>, <vitesse de rotation>{ , <NoBrSys> <NomBrSys>, <vitesse de rotation>} ...)	
avec	
<NoBrSys>	1 à 32
<NomBrSys>	Nom de la broche du système
<vitesse de rotation>	> 0 : limite supérieure de la vitesse de rotation en tr/min -1 : désactivation de la limitation de la vitesse de rotation

Tab. 6-157: Syntaxe SSpMax

Exemple :

SSpMax (SSp05, 2000, SSp08, 3000) Limite la vitesse de rotation maximale de la broche du système 5 à 2000 et la vitesse de rotation maximale de la broche du système 8 à 3000 tr/min.

SSpMax (SSp05, -1, SSp08, -1) Supprime la limite supérieure de la vitesse de rotation pour les broches du système 5 et 8.

Fonctions CN avec syntaxe de langage standard

6.124.6 Limitation de la vitesse de rotation minimale "SSpMin"

Description : Limite la vitesse de rotation minimale d'une ou plusieurs broches du système.**Syntaxe :**

SSpMin, SSPMIN	
SSpMin (<NoBrSys> <NomBrSys>, <vitesse de rotation>{ , <NoBrSys> <NomBrSys>, <vitesse de rotation>} ...)	
avec	
<NoBrSys>	1 à 32
<NomBrSys>	Nom de la broche du système

Tab. 6-158: Syntaxe SSpMin (SSPMIN)

Exemple :

SSpMin (5, 100, 8, 500) Limite la vitesse de rotation minimale de la broche du système 5 à 100 et la vitesse de rotation minimale de la broche du système 8 à 500 tr/min.

SSpMin (5, 0, 8, 0) Supprime la limite inférieure de la vitesse de rotation pour les broches du système 5 et 8.

6.124.7 Commutation de l'interface de position/vitesse de rotation "SSpMode"

Description : Commute l'entraînement des broches du système programmées entre le mode d'asservissement en vitesse de rotation et le mode d'asservissement en position.**Syntaxe :**

SSpMode, SSPMODE	
SSpMode (<NoBrSys> <NomBrSys>, <mode> { , <NoBrSys> <NomBrSys>, <mode>} ...)	
avec	
<NoBrSys>	1 à 32
<NomBrSys>	Nom de la broche du système
<mode>	1 : activer l'interface de position 0 : désactiver l'interface de position

Tab. 6-159: Syntaxe SSpMode

Exemple :

SSpMode (5, 1, 8, 0) Active l'interface de position pour la broche du système 5 et désactive l'interface de position pour la broche du système 8.

Particularités et restrictions : L'activation de l'interface de position se fait toujours quand la broche est à l'arrêt.

6.124.8 Programmation de mouvement "SSpMove (SSPM)"

Description : Démarre ou arrête une ou plusieurs broches du système.**Syntaxe :**

SSpMove, SSPMOVE, SSPM	
SSpMove (<NoBrSys> <NomBrSys>, <ordre>{ , <NoBrSys> <NomBrSys>, <ordre>} ...)	
avec	
<NoBrSys>	1 à 32

Fonctions CN avec syntaxe de langage standard

<NomBrSys>	Nom de la broche du système
<ordre>	3 - Rotation de la broche vers la droite 4 - Rotation de la broche vers la gauche 5 - Arrêt de la broche

Tab. 6-160: Syntaxe SSpMove (SSPM)

Exemple : SSpMove (SSP05, 3, SSP08, 5) Démarre la broche du système 5 avec une marche à droite et arrête la broche du système 8.

6.124.9 Orientation de la broche "SSpOri (SSPO)"

Description : Les broches du système programmées se positionnent sur la position angulaire programmée.

Syntaxe :

SSpOri, SSPORI, SSPO	
SSpOri (<NoBrSys> <NomBrSys>, <angle>{ , <NoBrSys> <NomBrSys>, <angle>} ...)	
avec	
<NoBrSys>	1 à 32
<NomBrSys>	Nom de la broche du système
<angle>	0.0 à 359.9999

Tab. 6-161: Syntaxe SSpOri (SSPO)

Exemple : SSpOri (5, 20, 8, 0) Oriente la broche du système 5 à 20 degrés et la broche du système 8 à 0 degré.

- Particularités et restrictions :**
- La commande numérique attend l'exécution du mouvement d'orientation.
 - Contrairement au mouvement d'orientation avec M19, M119, etc., l'angle doit toujours être programmé.
 - Si un mouvement défini avec SSpMove est programmé simultanément pour une broche dans le même bloc CN, alors SSpMove est ignoré (aucun message d'erreur).
 - La définition du signal d'interface "qSp_SValueSD" entraîne la définition de position via la donnée système "SysSpCmdData[n]/OriPos". La valeur programmée est ignorée.

6.124.10 Programmation de la vitesse de rotation "SSpSpeed (SSPS)"

Description : Préréglage de la vitesse de rotation pour une ou plusieurs broches du système.

Syntaxe :

SSpSpeed, SSPSPEED, SSPS	
SSpSpeed (<NoBrSys> <NomBrSys>, <vitesse de rotation>{ , <NoBrSys> <NomBrSys>, <vitesse de rotation>} ...)	
avec	
<NoBrSys>	1 à 32
<NomBrSys>	Nom de la broche du système

Tab. 6-162: Syntaxe SSpSpeed (SSPS)

Fonctions CN avec syntaxe de langage standard

Exemple :

SSpSpeed(SSp05, 1000,
SSp08, 2000)

Définit la vitesse de rotation de la broche du système 5 sur 1000 et la vitesse de rotation de la broche du système 8 sur 2000 tr/min.

- Particularités et restrictions :
- Par défaut, les vitesses de rotation programmées sont interprétées en tr/min.
 - Si G96 est active, la vitesse de rotation programmée est interprétée en tant que vitesse de coupe en m/min.
 - La commande limite la vitesse de rotation de consigne de manière à respecter les limites du rapport de réduction actif et de la limitation de vitesse de rotation programmée (voir SMin, SMax, SSpMin et SSpMax).
 - La vitesse de rotation réglée s'applique tant qu'elle n'a pas été écrasée par une nouvelle vitesse de rotation (effet modal).
 - Initialiser le signal d'interface "qSp_SValueSD" entraîne le préréglage de la vitesse de rotation via la donnée système "SysSpCmdData[n]/Speed". La valeur programmée est ignorée.

6.124.11 Limitations de la broche spécifiques à l'outil "SSpToolLim"

Description :

Active les limites spécifiques à l'outil d'une ou plusieurs broches du système, qui sont enregistrées dans la donnée système /SysSpCmdData[<BrSys>]/Tool.



Vous trouverez de plus amples informations dans la documentation "Description des fonctions – Fonctions de base IndraMotion MTX", chapitre "Fonctions de broche", sous "Limitations spécifiques à l'outil".

Syntaxe :

SSpToolLim, SSPTOOLLIM	
SSpToolLim(<NoBrSys> <NomBrSys>, <0 1>{ , <NoBrSys> <NomBrSys>, <0 1>} ...)	
avec	
<NoBrSys>	Numéro de la broche du système
<NomBrSys>	Nom de la broche du système
<0 1>	1 : active les limitations spécifiques à l'outil issues de /SysSpCmdData[]/Tool. 0 : désactive les limitations spécifiques à l'outil.

Tab. 6-163: Syntaxe SSpToolLim

Exemple :

SSpToolLim(3,1)

Active les limitations spécifiques à l'outil de la 3e broche du système.

- Particularités et restrictions :
- SSpToolLim arrête les broches programmées, puis les relance.
 - SSpToolLim informe les fonctions "SpindleWait, SPWAIT" et "SpindleWaitMode, SWM" d'un changement de rapport de réduction.
 - SSpToolLim contient une fonctionnalité WAIT interne à la commande numérique.

6.125 Activation du mode axe C pour les broches "SpindleToAxis, STA"

Description : *Commute une broche saisie dans les paramètres machine suivants au mode axe C :*

- MP 1001 00001 (type de fonction d'entraînement) en tant que broche/axe C
- MP 1040 00001 (sélection du type de broche) en tant que broche Ser-cos

À des fins d'usinage, la broche devient ainsi un axe asynchrone.

Sur l'écran, elle apparaît comme un axe asynchrone qui se situe dans d'abord sur une position quelconque entre 0 et 359.9999 degrés.



Pour obtenir des informations détaillées sur la fonctionnalité "Transfert d'axe", voir la documentation "Description des fonctions IndraMotion MTX".

Syntaxe :

SpindleToAxis (<NPA> <IPA>{ , <NPA> <IPA>}...)	
Format abrégé : STA (. .)	
avec	
<NPA>	Nom physique de l'axe Définit la broche qui doit être commutée au mode axe C.
<IPA>	Index physique de l'axe Effet similaire à <NPA>

Tab. 6-164: Syntaxe SpindleToAxis (STA)

Exemple :

```
:
N030      L'axe physique "CH", c.-à-d. la broche ayant le nom "CH" en mode axe,
STA (CH)  est converti en axe asynchrone.
:
```

Particularités et restrictions : Les positions d'axes d'un même bloc doivent toujours être programmées après l'expression "SpindleToAxis(...)".

6.126 Activer le mode axe C pour des broches et attendre "SpindleToAxisWait, STAW"

Description : *Active une broche saisie dans les paramètres machine suivants en mode axe C et attend que l'axe C devienne actif :*

- MP 1001 00001 (type de fonction d'entraînement) en tant que broche/axe C
- MP 1040 00001 (sélection du type de broche) en tant que broche Ser-cos

À des fins d'usinage, la broche devient ainsi un axe asynchrone.

Sur l'écran, elle apparaît comme un axe asynchrone qui se situe dans d'abord sur une position quelconque entre 0 et 359.9999 degrés.

Fonctions CN avec syntaxe de langage standard



Pour obtenir des informations détaillées sur la fonctionnalité "Transfert d'axe", voir la documentation "Description des fonctions IndraMotion MTX".

Syntaxe :

```
SpindleToAxisWait (<NPA>|<IPA>{ , <NPA>|<IPA>}...)
```

Format abrégé : STAW (. .)

avec

<NPA>	Nom physique de l'axe Définit la broche qui doit être commutée au mode axe C.
<IPA>	Index physique de l'axe Effet similaire à <NPA>

Tab. 6-165: Syntaxe SpindleToAxisWait (STAW)

Exemple :

:

N030 STAW (CH) L'axe physique "CH", c.-à-d. la broche ayant le nom "CH" en mode axe, est converti en axe asynchrone.

N040 CH90 L'axe asynchrone CH est positionné à 90 degrés.

:

Particularités et restrictions : Les positions d'axes d'un même bloc doivent toujours être programmées après l'expression "SpindleToAxisWait(...)".

6.127 Attente d'une programmation de la broche à la fin du bloc "SpindleWait, SPWAIT"

Description : À la fin du bloc, attend que la broche correspondante ait atteint l'état voulu.



Vous trouverez de plus amples informations dans la documentation "Description des fonctions IndraMotion MTX", chapitre "Fonctions de broche", sous "Mouvement sur trajectoire et fonctions de la broche".

Syntaxe :

```
SpindleWait (<broche> { , <condition>})
```

Format abrégé : SPWAIT (. .)

avec

<broche>	Définit la/les broche(s) du canal qu'il faut attendre. -2 : toutes les broches sauf la broche principale -1 ou MSP : la broche principale 0 ou ALL : toutes les broches 1 à 8 ou S1 à S8 : numéro ou nom d'une broche

Fonctions CN avec syntaxe de langage standard

<condition>	Optionnel. Liste des conditions d'attente avec le contenu maximal possible 3, 5, 19, 40 3 : attente de la vitesse de rotation atteinte avec M3 ou M4 5 : attente de la broche sur M5 19 : attente de la broche orientée avec M19 40 : attente de la fin d'un changement du rapport de réduction Valeur par défaut : Si aucune condition n'est programmée, toutes les conditions sont attendues.
SpindleWait	
	SpindleWait sans liste de paramètres attend d'atteindre les fonctions de broche programmées dans ce bloc.

Tab. 6-166: Syntaxe SpindleWait (SPWAIT)

Exemple :

```

:
N30 M3 S1000 SPWAIT           Attend à la fin d'un bloc jusqu'à ce que le
:                               message « Vitesse de rotation atteinte » s'affiche.
N260 M203 S2=500 G0 X1000     La broche 2 est démarrée.
N270 G0 Y10 SPWAIT(S2,3)      Attend à la fin du bloc jusqu'à ce que la broche 2 signale « Vitesse de rotation atteinte ».
N280 G1 ...
:

```

6.128 Sélection d'une stratégie d'attente "SpindleWaitMode, SWM"

Description : Sélectionne et active une stratégie d'attente pour la fonction "Mouvement sur trajectoire et fonctions de la broche".



Vous trouverez de plus amples informations dans la documentation "Description des fonctions IndraMotion MTX", chapitre "Fonctions de broche", sous "Mouvement sur trajectoire et fonctions de la broche", ainsi que dans la documentation "Paramètres machine IndraMotion MTX", chapitre "Broches".

Syntaxe :

SpindleWaitMode(<broche> , <activer> {, <stratégie>)	
Format abrégé : SWM (..)	
avec	
<broche>	Définit la/les broches pour lesquelles la stratégie doit être modifiée. -2 : toutes les broches sauf la broche principale -1 ou MSP : la broche principale 0 ou ALL : toutes les broches 1 à 8 ou S1 à S8 : numéro ou nom d'une broche

Fonctions CN avec syntaxe de langage standard

<activer>	0 : désactiver la stratégie d'attente 1 : activer la stratégie d'attente
<stratégie>	Optionnel. 1 à 3 : numéro de la stratégie 1 : stratégie prédéfinie 2 et 3 : SP/SpAdmin/BlkSwiCtrl

Tab. 6-167: Syntaxe SpindleWaitMode (SWM)

Exemple :	:	
	N010 SpindleWaitMode (MSP, 1, 2)	La stratégie 2 est sélectionnée et activée pour la broche principale.
	:	
	N150 SpindleWaitMode (MSP, 0)	La stratégie d'attente actuelle (2) de la broche principale est désactivée.
	:	
	N190 SpindleWaitMode (MSP, 1)	La stratégie d'attente actuelle (2) de la broche principale est réactivée.
	:	

6.129Arrondissage des angles avec des splines

6.129.1Arrondissage des angles avec des splines (modal) "SplineCornering, SCO"

Description : La fonction CN modale "Arrondissage des angles avec des splines" arrondit avec une spline tous les raccords entre segments de contour dont l'angle de coude est supérieur ou égal à un angle de coude minimal configuré.

Syntaxe :	SplineCornering (E<écart>)
	"Arrondissage des angles ACTIVÉ" avec indication de la tolérance E maximale admissible.
SplineCornering (DEF)	"Arrondissage des angles ACTIVÉ" avec une tolérance E issue de MP 7050 00110.
	SplineCornering (L1=<L1>, L2=<L2>)
	"Arrondissage des angles ACTIVÉ" avec indication d'un écart angulaire préliminaire et subséquent L1 resp. L2.
	SplineCornering(L<L>) "Arrondissage des angles ACTIVÉ" avec indication d'un trajet angulaire préliminaire et subséquent symétrique L ₁ = L ₂ = L SplineCornering(L<L>) "Arrondissage des angles ACTIVÉ" avec indication d'un trajet angulaire préliminaire et subséquent symétrique L ₁ = L ₂ = L SplineCornering(L<L>,E<écart>) "Arrondissage des angles ACTIVÉ" avec indication de la tolérance E en cas de limitation simultanée du trajet angulaire préliminaire et subséquent à L

Fonctions CN avec syntaxe de langage standard

SplineCornering(0)	"Arrondissage des angles DÉACTIVÉ"
Format abrégé : SCO	

Tab. 6-168: Syntaxe SplineCornering (SCO)

Particularités et restrictions :

- L'arrondissage se fait dans toutes les coordonnées (jusqu'à 8) d'un canal et pour tout type de contour.
- La fonction CN modale "Arrondissage des angles avec des splines" peut être masquée localement, c.-à-d. pour le passage au bloc suivant, par la fonction CN locale (non modale) "Arrondissage des angles avec des splines".
- Les paramètres machine applicables sont les mêmes que pour l'arrondissage circulaire des angles.

6.129.2 Arrondissage des angles avec des splines (local) "SplineCorneringLocal, SCOL"

Description : La fonction CN locale (non modale) "Arrondissage des angles avec des splines" arrondit le passage au bloc CN suivant avec une spline s'il s'agit d'un raccord entre segments de contour non tangentiel.

Syntaxe :

SplineCorneringLocal(E<écart>)	Arrondissage des angles avec indication de la tolérance E maximale admissible.
SplineCornering(DEF)	Arrondissage des angles avec tolérance E du MP 7050 00110.
SplineCornering(L1=<L1>, L2=<L2>)	Arrondissage des angles avec indication d'un écart angulaire préliminaire et subséquent L1 resp. L2.
SplineCornering(0)	Aucun arrondissage des angles pour le passage au bloc CN suivant.
Format abrégé : SCOL	

Tab. 6-169: Syntaxe SplineCornering (SCOL)

Particularités et restrictions :

- L'arrondissage se fait dans toutes les coordonnées (jusqu'à 8) d'un canal et pour tout type de contour.
- La fonction CN locale "Arrondissage des angles avec des splines" permet de masquer localement, c.-à-d. pour le passage au bloc suivant, la fonction CN modale "Arrondissage des angles avec des splines".
- Les paramètres machine applicables sont les mêmes que pour l'arrondissage circulaire des angles.

6.130 Définition du type de spline "SplineDef, SPD"

Description : La commande IndraMotion MTX prend en charge les types de splines suivants :

- **Type de spline 0 :**
spline avec programmation de coefficients
(coefficients des polynômes du système CAO/FAO)
- **Type de spline 1 :**

Fonctions CN avec syntaxe de langage standard

Splines cubiques constantes C1 avec programmation de repères et, le cas échéant, donnée de tangente (splines d'Hermite)

(raccords tangentiels aux repères)

- **Type de spline 2 :**

Splines cubiques constantes C2 avec programmation de repères

(raccords à courbure constante aux repères)

- **Type de spline 3 :**

Spline B avec programmation de points de contrôle

(tracé de la courbe près des repères)

- **Type de spline 4 :**

Approximation spline B

(bande de tolérance affectée autour des repères)

Le type de spline voulu est sélectionné et initialisé à l'aide de la fonction "SplineDef" (SPD). Ensuite, la programmation des splines peut être activée à l'aide de la fonction "G6".

Syntaxe :

SplineDef (<id>, {<membres>})	
	Initialiser le type de spline
Format abrégé : SPD (. .)	
avec	
<id>	Nombre entier à 4 chiffres au maximum pour la variante spline

Tab. 6-170: Syntaxe SplineDef (SPD)

Type de spline	Chiffre			
	Millier	Centaine	Dizaine	Unité
	Type de spline 0 à 4	Paramétrage 0 : aucun 1 : équidistant 2 : à la corde 3 : centripète	Calcul ou indication de la tangente 0 : aucun 1 : Bessel 2 : Akima 3 : rapport à la corde 4 : Hermite	Degré de lissage 1 à 5
Type de spline 0	0	0	0	1 à 5
Type de spline 1	1	1 à 3	0 à 4	1 à 5
Type de spline 2	2	1 à 3	0	3
Type de spline 3	3	1 à 3	0	1 à 5
Type de spline 4	4	2	0	1 à 5

Tab. 6-171: Aperçu des types de splines

Fonctions CN avec syntaxe de langage standard

<membres>	Non applicable au type de spline 0 et spline d'Hermite ! Liste contenant les noms des coordonnées/axes qui doivent participer au mouvement spline. Les coordonnées/axes non programmés dans <membres> sont déplacés linéairement. Il est également possible de programmer des coordonnées d'orientation en tant que spline. Dans ce cas, il suffit d'indiquer "O" pour l'orientation ou les coordonnées polaires "phi" et "thêta" dans <membres>. Si aucune coordonnée n'est programmée dans <membres>, les valeurs d'activation suivantes s'appliquent : Si une transformation à 5 axes est active : x,y,z,O Sinon : tous les membres du canal

Tab. 6-172: Suite de la syntaxe SplineDef (SPD)

Exemple :

N10 SplineDef(5)	Initialiser le type de spline 0 avec le degré de lissage 5
N15 SplineDef(1143)	Initialiser le type de spline 1 Spline d'Hermite avec le degré de lissage 3
N20 SplineDef(2203,X,Y,Z)	Initialiser le type de spline 2 avec le degré de spline 3 et un paramétrage à la corde. Les coordonnées X, Y et Z sont concernées.
N30 Spline- Def(3103,x,y,z,O)	Initialiser le type de spline 3 avec le degré de lissage 3 et un paramétrage équidistant. Les coordonnées x, y et z ainsi que les coordonnées d'orientation sont concernées.
N30 SplineDef(4203)	Initialiser le type de spline 4 avec le degré de lissage 3. Toutes les coordonnées du canal sont concernées.

6.131 Répartition du parcours programmable "Split, SPT"

Description : Répartit les blocs de déplacement programmables en plusieurs segments de parcours dès qu'ils dépassent une certaine longueur.

Syntaxe :

Split({<mode>},{, <longueur partielle>})	
Format abrégé : SPT (...)	
avec	
<mode>	0 : la fonction est sans effet. 1 : la fonction a un effet modal. 2 : La fonction a un effet uniquement dans le bloc programmé. En l'absence de programmation : effet identique à 0.
<longueur partielle>	Pour les blocs linéaires : longueur du segment de parcours Pour les blocs circulaires : longueur du segment d'arc Pour l'unité de programmation : voir les coordonnées d'axes Si <longueur de segment> n'est pas programmé, alors : la <longueur partielle> efficace est environ égale à la course maximale qu'il est possible de couvrir en 2 cycles d'interpolation consécutifs en fonction de l'avance actuelle.

Tab. 6-173: Syntaxe Split

Exemple :

Fonctions CN avec syntaxe de langage standard

:	
N20 Split(2,10) X100	Répartition du parcours efficace au sein du bloc pour N20 ACTIVÉE
:	Le mouvement de déplacement dans N20 est divisé en segments de parcours de 10 mm chacun.
N120 Split(1)	Répartition du parcours efficace de façon modale à partir de N120 ACTIVÉE
:	À partir de N120, les mouvements de déplacement sont divisés en segments de parcours qui peuvent être effectués en environ 2 cycles d'interpolation.
:	
N220 Split(1,3.0)	Répartition du parcours efficace de façon modale à partir de N220 ACTIVÉE
:	À partir de N220, les mouvements de déplacement sont divisés en segments de parcours de 3 mm chacun.
N320 Split()	Répartition du parcours programmable DÉSACTIVÉE
:	

- Particularités et restrictions :
- La répartition du parcours programmable ne divise un bloc de déplacement que lorsque sa longueur dépasse la <longueur partielle>.
 - Si la <longueur partielle> n'est pas un diviseur entier de la course, le parcours restant est également déplacé séparément.
 - Si la répartition du parcours programmable a un effet au sein du bloc seulement (<mode> = 2), alors un mouvement de déplacement doit être programmé dans le même bloc.

6.132 Broche : commutation de l'interface de position/vitesse de rotation "SpMode, SPM"

Description : Commute l'entraînement de la broche entre les modes d'asservissement en vitesse de rotation et d'asservissement en position.

Syntaxe :	SpMode (S<num>=<mode>{, S<num>=<mode>}...)	
	Format abrégé : SPM (...)	
	avec	
	<num>	Numéro de la broche (index de la broche) Champ de saisie : 1 à 8 Nombre entier
	<mode>	0 : mode d'asservissement en vitesse de rotation 1 : mode d'asservissement en position

Tab. 6-174: Syntaxe SpMode (SPM)

Exemple :

N60 SPM(S1=1, S2=1)

:

Commuter la 1re et la 2e broches au mode d'asservissement en position

- Particularités et restrictions :
- L'orientation de la broche est interrompue.
 - Une broche en cours de rotation est arrêtée pendant un court instant lors de la commutation au mode d'asservissement en position.

6.133 Paramétrage de l'orientation statique de l'outil "StatToolOri, STO"

Description : Les corrections de longueur d'outil L_1 , L_2 et L_3 sont attribuées aux directions de coordonnées L_1 , L_2 et L_3 par l'intermédiaire de G47(...) ou G78. Ces directions de coordonnées forment un système de coordonnées et le vecteur de correction la longueur d'outil L dispose des composantes L_1 , L_2 et L_3 dans ce système de coordonnées.

L'ordre "StatToolOri" permet de faire pivoter L dans ce système de coordonnées autour des angles d'Euler ϕ (φ), θ (ϑ) et ψ (ψ), c.-à-d. :

- d'abord autour de l'angle **phi** pour la direction de coordonnée L_3 ,
- ensuite autour de l'angle **thêta** pour la direction de coordonnée L_2 **qui a également subi la rotation précédente**,
- et enfin autour de l'angle **psi** pour la direction de coordonnée L_3 **qui a également subi les rotations précédentes**.

Les rotations suivantes de L aboutissent au même résultat :

- Tourner d'abord autour de l'angle **psi** pour la direction de coordonnée L_3 ,
- ensuite autour de l'angle **thêta** pour la direction de coordonnée L_2 **originelle, c.-à-d. qui n'a pas subi la rotation précédente**,
- et enfin autour de l'angle **phi** pour la direction de coordonnée L_3 **originelle, c.-à-d. qui n'a pas subi les rotations précédentes**.

Étant donné que les deux modes opératoires aboutissent au même résultat, vous pouvez utiliser celui qui vous parle le plus.

L'orientation statique de l'outil se rapporte toujours au vecteur de correction de l'outil non pivoté. Les orientations statiques successives de l'outil n'ont donc pas un effet additif.

Syntaxe :

StatToolOri ({< φ >} {< ϑ >} {< ψ >})	
Format abrégé : STO (. .)	
avec	
< φ >,< ϑ >,< ψ >	Angle d'Euler en degrés Plage de valeurs : $0^\circ \leq \varphi < 360^\circ$; $0^\circ \leq \vartheta \leq 180^\circ$; $0^\circ \leq \psi \leq 360^\circ$. Les valeurs situées en dehors de ces limites sont converties automatiquement dans l'intervalle correspondant.

Tab. 6-175: Syntaxe StatToolOri (STO)

Rotation du vecteur de correction de l'outil autour de l'angle Alpha pour la direction de coordonnée L_1 :

Direction de coordonnée L_1 :

StatToolOri (-90, <alpha>, 90)

Fonctions CN avec syntaxe de langage standard

Exemple :

```

1 DCT(1,1,0) = 100
1 DCT(2,1,0) = 0
1 DCT(3,1,0) = 0
; Composantes du vecteur de correction de l'outil
G0 X0 Y0 Z0 ; (L1,L2,L3)
G47(X,Y,Z) ED1 ; (100,0,0)
STO(90,0,0) ; (0,100,0)
STO(0,0,0) ; (100,0,0)
STO(0,90,0) ; (0,0,-100)
G48 ED0
M30

```

Exemple :

```

1 DCT(1,1,0) = 0
1 DCT(2,1,0) = 0
1 DCT(3,1,0) = 100
; Composantes du vecteur de correction de l'outil
G0 X0 Y0 Z0 ; (L1,L2,L3)
G47(X,Y,Z) ED1 ; (0,0,100)
STO(0,90,0) ; (100,0,0)
STO(-90,45,90) ; ( 0 , -50*sqrt(2) , 50*sqrt(2) )
STO(0,30,0) ; ( 50 , 0 , 50*sqrt(3) )
STO(45,30,0) ; (25*sqrt(2),25*sqrt(2),50*sqrt(3))
G48 ED0
M30

```

Particularités et restrictions :

- L'orientation statique de l'outil est paramétrée avec "STO" et activée avec "G47" (voir [chap. 5.2.18 "Correction de la longueur d'outil G47, G48" à la page 160](#)).
- Si "G47" est déjà active lors de l'exécution de STO, la modification de l'orientation de l'outil prend immédiatement effet.
- Si des désignateurs de coordonnées n'ont pas été attribués aux trois corrections de longueur d'outil L_1 , L_2 et L_3 via G47(...) ou G78, la commande ne tient compte que des coordonnées attribuées pour orienter l'outil et génère un avertissement.



Voir aussi "STP" dans la documentation "Cycles CN standards IndraMotion MTX".

6.134 Neutralisation des types de fonctionnement à étape unique "StepModeDisable, SMD"

Description :

La neutralisation de l'étape unique sert à traiter complètement des parties d'un programme CN en une seule fois dans l'un des types de fonctionnement "Étape unique", "Bloc unique", "Bloc-programme" et "Étape de débogage". Cela signifie que les blocs sont traités sans confirmer le démarrage CN.

Il est possible d'encapsuler entièrement les sous-programmes (asynchrones) et les cycles de sorte que l'ensemble du sous-programme (asynchrone) ou

Fonctions CN avec syntaxe de langage standard

du cycle soit traité avec un démarrage CN, même si le type de fonctionnement actuel ne le prévoit pas.

La neutralisation du type de fonctionnement à étape unique ne constitue pas une modification du type de fonctionnement, c.-à-d. que la géométrie et la vitesse sont calculées comme au type de fonctionnement sélectionné (par ex. Bloc unique), mais que l'attente du signal de démarrage CN est neutralisée.

Il est possible de commander séparément l'effet de la fonction pour les types de fonctionnement Bloc unique/Étape unique, Bloc-programme et Étape de débogage. Cela signifie qu'il est possible, par exemple, d'activer la neutralisation du démarrage CN pour Étape unique/Bloc unique et Bloc-programme sans l'activer pour le type de fonctionnement "Étape de débogage".

Programmation :

La neutralisation du bloc unique est programmée sous forme de fonction CN à effet modal. Il existe une fonction de mise en/hors service.

Syntaxe :

StepModeDisable(({variable={ } {valeur}} { , } {variable={ } {valeur}}))	
Format abrégé : SMD { (. .) }	
avec les variables programmables optionnelles	
SBLK	Les points suivants s'appliquent pour les types de fonctionnement "Bloc unique" et "Étape unique" : 0 - Le démarrage CN n'est pas neutralisé. 1 - Le démarrage CN est neutralisé. Préréglage : 1
PBLK	Valable pour le type de fonctionnement "Bloc-programme" 0 - Le démarrage CN n'est pas neutralisé. 1 - Le démarrage CN est neutralisé. Préréglage : 1
DBGS	Valable pour le type de fonctionnement "Étape de débogage" 0 - Le démarrage CN n'est pas neutralisé. 1 - Le démarrage CN est neutralisé. Préréglage : 1
NOW	Indique si la neutralisation est valable à partir du bloc dans lequel la fonction a été programmée. 0 - Le démarrage CN est uniquement neutralisé à partir du bloc suivant. 1 - Le démarrage CN est également neutralisé pour ce bloc. Préréglage : 0
Si une variable n'est pas programmée, les préréglages indiqués prennent effet. Si la valeur n'est pas programmée, la valeur 1 est attribuée à la variable. Si aucune variable n'est programmée, les parenthèses sont optionnelles. Les parenthèses vides n'agissent pas comme la fonction de mise hors service, mais comme la fonction de mise en service avec les valeurs de préréglage.	

Tab. 6-176: Syntaxe StepModeDisable (SMD)

Fonctions CN avec syntaxe de langage standard

StepModeEnable
Format abrégé : SME
Aucun paramètre programmable. Le bloc dans lequel SME est programmé compte encore parmi les blocs pour lesquels le démarrage CN est neutralisé.

Tab. 6-177: Syntaxe StepModeEnable (SME)

Exemples :

Exemple 1 : ... N50 SMD N60 X100 N70 SME ...	Dans tous les types de fonctionnement Bloc unique, les blocs N60 et N70 ne sont pas confirmés avec le démarrage CN.
Exemple 2 : ... N50 SMD X100 N60 Y100 N70 SME ...	Similaire à l'exemple 1
Exemple 3 : ... N50 SMD(NOW1) X100 N60 Y100 N70 SME ...	Contrairement aux exemples 1 et 2, le bloc N50 n'est pas confirmé par le démarrage CN malgré le mouvement de déplacement. De même, les blocs N60 et N70 ne sont pas confirmés par le démarrage CN.
Exemple 4 : ... N50 SMD(DBGS0) X100 N60 Y100 SME ...	Le bloc N60 peut être confirmé avec le démarrage CN au type de fonctionnement "Étape de débogage" ; le démarrage CN est neutralisé aux types de fonctionnement "Bloc-programme" et "Étape unique/Bloc unique".
Exemple 5 : Programme principal ... N50 P Up ... Sous-programme Up N10 SMD(NOW) N999 SME	Le démarrage CN dans le bloc N50 du programme principal traite entièrement le sous-programme Up à tous les types de fonctionnement Étape unique, ce qui signifie qu'aucun démarrage CN n'est nécessaire dans le sous-programme. Il convient de veiller ici à ce que la neutralisation du bloc unique soit activée avec la variable "NOW" dans le premier bloc du sous-programme et désactivée dans le dernier bloc.

Tab. 6-178: Exemples de programmation de la neutralisation du type de fonctionnement Étape unique

Fonctions CN avec syntaxe de langage standard

Particularités et restrictions : La neutralisation du type de fonctionnement Étape unique n'empêche pas de changer un type de fonctionnement par un type de fonctionnement verrouillé ! Le type de fonctionnement change, mais les blocs verrouillés continuent d'être traités sans démarrage CN.

La programmation d'une nouvelle SME écrase les variables d'une SME active plus tôt. Seule un SMD est requis.

SME est active à la fin de la montée en régime de la commande ; elle est également activée implicitement par la remise à zéro et la fin du programme. Cela signifie que la neutralisation du type de fonctionnement n'a pas d'effet modal sur un redémarrage du programme, ni sur un changement de programme. SME n'est toutefois pas activée globalement en cas de saut en arrière d'un sous-programme. Ceci s'applique également aux sous-programmes asynchrones. Il convient ici de veiller impérativement à ce que la fonction "SME" soit programmée pour désactiver une neutralisation, faut de quoi les blocs d'un programme CN ne pourront plus être confirmés involontairement dans un type de fonctionnement Bloc unique avec certains ordres de démarrage CN.

Chaque programme est d'abord démarré avec la fonction "SME". Pour des raisons de sécurité, ce comportement n'est pas réglable.

6.135 Limitations de la broche spécifiques à l'outil "SToolLim"

Description : Active les limitations spécifiques à l'outil d'une ou plusieurs broches du canal, qui sont enregistrées dans la donnée système `/SysSpCmdData[<SysSp>]/Tool`.



Vous trouverez de plus amples informations dans la documentation "Description des fonctions de base IndraMotion MTX", chapitre "Fonctions de broche", sous "Limitations spécifiques à l'outil".

Syntaxe :

SToolLim, STOOLLIM	
SToolLim (<nomBr> = <0 1>{ , <nomBr> = <0 1>} ...)	
avec	
<nomBr>	Nom de la broche (S, S1 à S8)
<0 1>	1 : active les limitations spécifiques à l'outil issues de <code>/SysSpCmdData[]/Tool</code> . 0 : désactive les limitations spécifiques à l'outil.

Tab. 6-179: Syntaxe SToolLim

Exemple :

`SToolLim (S2=1)` Active les limitations spécifiques à l'outil de la broche du système actuellement affectée à la 2e broche du canal.

Particularités et restrictions :

- SToolLim arrête les broches programmées, puis les relance.
- SToolLim informe les fonctions "SpindleWait, SPWAIT" et "SpindleWait-Mode, SWM" d'un changement du rapport de réduction.
- SToolLim contient une fonctionnalité WAIT interne à la commande numérique.

6.136 Déplacement dans le TCS "TCM(, ,)"

Description : Mouvement de déplacement linéaire dans le système de coordonnées de l'outil (TCS), même si la transformation d'axe 2 est active.

Il s'agit d'une programmation relative locale, c.-à-d. que l'indication de position programmée est interprétée de façon incrémentale, indépendamment de G90/G91.

Vous trouverez des informations complètes sur l'utilisation et le paramétrage de "TcsMove" dans la documentation "Description des fonctions IndraMotion MTX".

Syntaxe :	TCM ({<valX>},{<valY>},{<valZ>})	
	(en alternative : TcsMove (, ,))	
	avec	
	<valX>, <valY>, <valZ>	Valeurs à déplacer de manière incrémentale, par rapport au système de coordonnées de l'outil (TCS).

Tab. 6-180: Syntaxe TCM

Exemple :

TCM (,,-100)	; Immerger 100 [mm] en direction de l'outil
TCM (,,200)	; Retirer 200 [mm] en direction de l'outil

6.137 Guidage tangentiel de l'outil "TangTool, TTL"

Description : Guide un axe rotatif ou une coordonnée le long de la trajectoire programmée dans le plan actif avec un angle réglable. Cela permet de déplacer un outil sous l'angle d'attaque requis par rapport à la trajectoire programmée, à tout moment pendant l'intervention.

Pour faire tourner l'axe de l'outil sur l'angle d'attaque respectivement nécessaire, la fonction tient compte d'une symétrie d'outil éventuelle pour les outils à plusieurs arêtes coupantes. En outre, la commande peut, en cas de coude de contour lors de l'enchaînement de blocs, insérer automatiquement un bloc intermédiaire pour tourner l'axe d'outil de l'angle nécessaire.

Les points suivants s'appliquent ici :

- Lors d'une position à 0 degré de l'axe d'outil, la direction de coupe par défaut de l'outil est parallèle à la direction de déplacement positive de l'axe primaire du plan actuel.

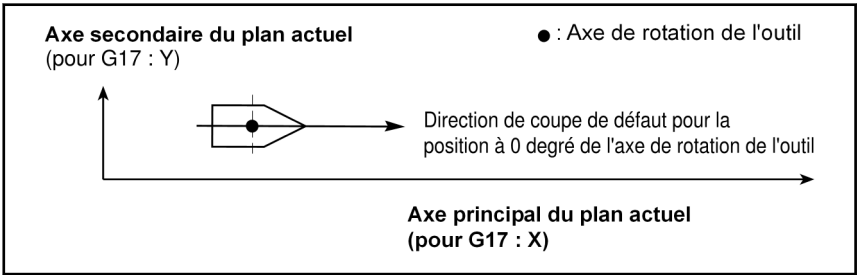


Fig. 6-38: Position 0 degré de l'axe de rotation de l'outil

- L'angle d'attaque indique la différence angulaire entre la trajectoire et la direction de coupe par défaut pour l'outil.
- Il est de 0 degré, lorsque la direction de coupe par défaut pour l'outil suit un tracé tangentiel par rapport à la trajectoire.

Fonctions CN avec syntaxe de langage standard

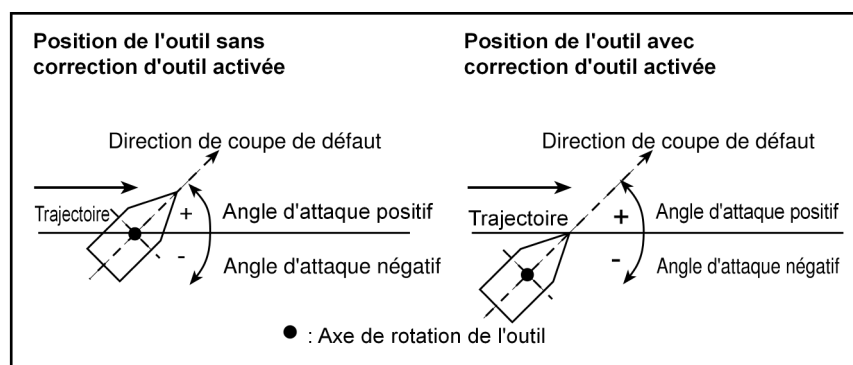


Fig. 6-39: Angle d'attaque

- "TangTool" ne provoque aucun mouvement de déplacement lorsqu'elle est activée.

Un axe d'outil programmé n'est réglé qu'à partir du mouvement de déplacement suivant. En fonction de l'angle de bloc intermédiaire programmé :

- soit un bloc intermédiaire pour la rotation de l'outil est déplacé au préalable ;
- soit un saut vers la position de réglage est exécuté au début du bloc.
- La rotation requise de l'axe d'outil à un angle d'attaque donné sur une trajectoire circulaire est calculée au cours du cycle d'interpolation. Pour ce type de segments de contour, l'axe de l'outil continue à tourner en cycle d'interpolation autour de l'angle d'attaque calculé actuel.



L'angle de bloc intermédiaire (ZSW) maximal autorisé ne doit pas dépasser l'angle que couvre l'axe concerné en une cadence Serco à la vitesse d'avance rapide :

$$\text{ZSW [°]} < \text{AXDr[i]/Vel/MaxVel} * \text{SCS/ScsIf[1]/ScsCycTime} * 1/60.000.000$$

Syntaxe :

TangTool { ({TAX{=} <coordonnée> } {,SYM{=} <s> } {,ANG{=} <a> } {,IA{=} <ZSW> } {,PLC{=} <p> }) }	
	Guidage tangentiel de l'outil ACTIVÉ
TangTool () ou TangTool (0)	Guidage tangentiel de l'outil DÉSACTIVÉ
Format abrégé : TTL (. .)	
avec	
<coordonnée>	Désignation de la coordonnée/l'axe sur lequel doit agir le guidage tangentiel de l'outil. Les éléments suivants sont permis : Nom ou numéro de la coordonnée du canal ou de l'axe du canal. En l'absence de programmation, MP 7050 00210 prend effet. Ici, seul un numéro d'axe système peut être saisi.

Fonctions CN avec syntaxe de langage standard

<s>	Symétrie de l'outil (en règle générale, le nombre d'arêtes coupantes). Valeur entrée : nombre entier, différent de 0. Un outil avec la symétrie <s> atteint une position techniquement équivalente lors d'une rotation de 360 degrés / <s>. Exemples : Outil rectangulaire : <s> = 2, Outil carré : <s> = 4. 1 : Outil asymétrique ou équipé d'une seule arête coupante. > 0 : Outil symétrique qui possède plusieurs arêtes coupantes situées à équidistance. En cas de coude de contour, l'outil tourne de juste assez pour que l'arête coupante la plus proche se positionne selon l'angle d'attaque par rapport au contour. < 0 : Lors d'une inversion de direction (coude à 180 degrés), l'outil n'effectue pas de rotation, quel que soit l'angle d'attaque. Identique à "> 0" pour le reste. En l'absence de programmation, MP 7050 00220 prend effet.
<a>	Angle d'attaque Plage de valeurs : -180 degrés à +180 degrés Indique la différence angulaire entre la trajectoire et la direction de coupe par défaut de l'outil. En l'absence de programmation, MP 7050 00250 prend effet. Ce paramètre détermine si l'angle d'attaque utilisé correspond à l'angle actuel de l'axe rotatif de l'outil ou à la valeur indiquée dans MP 7050 002540.

Fonctions CN avec syntaxe de langage standard

<zsw>	Angle de bloc intermédiaire Plage de valeurs : 0 à 180 degrés Définit à partir de quel angle de coude entre deux blocs, un bloc intermédiaire doit être inséré pour la rotation de l'axe d'outil. Avec un angle de coude inférieur à <zsw>, aucun bloc propre n'est inséré pour la rotation. L'outil saute vers la nouvelle position au début du bloc suivant. L'angle de bloc intermédiaire (ZSW) maximal autorisé ne doit pas dépasser l'angle que couvre l'axe concerné en une cadence Sercos à la vitesse d'avance rapide : $ZSW [^\circ] < AXDr[i]/Vel/MaxVel * SCS/ScsIf[1]/ScsCycTime * 1/60.000.000$ En l'absence de programmation, MP 7050 00230 prend effet.
<p>	Activer ou désactiver la communication CN-API au cours de l'exécution d'un bloc intermédiaire. 0 : Communication CN-API DÉSACTIVÉE. La commande numérique exécute le bloc intermédiaire sans condition. 1 : Communication CN-API ACTIVÉE. L'exécution d'un bloc de rotation est pilotée via la communication CN-API.

Tab. 6-181: Syntaxe TangTool (TTL)

Exemple :

TTL	Pour l'axe inscrit dans MP 7050 00210, le guidage tangentiel de l'outil est activé avec les valeurs de paramètre machine correspondantes.
TTL (TAX=C, SYM1, ANG90, IA20, PLC0)	Programmation avec nom d'axe du canal
TTL (TAX=theta, ANG45)	Programmation avec nom de la coordonnée
TTL (TAX3, SYM1, ANG90, IA20, PLC0)	Programmation avec numéro de la coordonnée
TTL (TAX[NAME\$], SYM1, ANG90, IA20)	Programmation avec une variable CPL

Particularités et restrictions :

- Le "guidage tangentiel de l'outil" ne doit pas être actif en même temps que l'"orientation tangentielle de l'outil" (voir [chap. 6.138 "Orientation tangentielle de l'outil TangToolOri, TTO"](#) à la page 428).
- Le "guidage tangentiel de l'outil" ne doit pas être programmé conjointement avec un mouvement de déplacement.



Les expressions entre parenthèses "(0)" et "(1)" ne sont pas des paramètres de la fonction CN. La valeur numérique entre parenthèses ne peut donc pas être transmise par des variables CPL.

Fonctions CN avec syntaxe de langage standard

6.138 Orientation tangentielle de l'outil "TangToolOri, TTO"

Description :

 Agit uniquement en combinaison avec les fonctions "Poinçonnage" (voir [chap. 6.90 "Poinçonnage Punch, PUN" à la page 366](#)) et "Grignotage" (voir [chap. 6.76 "Grignotage Nibble, NIB" à la page 332](#)).

Lorsque Poinçonnage/Grignotage est active, garantit qu'un outil de poinçonnage/grignotage se trouve à un angle définissable par rapport à la trajectoire programmée pour chaque processus de course.

Lors d'une course en début de bloc, un bloc pour la rotation de l'outil est ajouté automatiquement si nécessaire.

La commande calcule automatiquement la direction de rotation optimale (voie la plus courte).

L'axe de rotation est défini fixement via MP 7050 00210.

Syntaxe :

TangToolOri ({SYM<s>}, {ANG<a>}) ou TangToolOri (1) TangToolOri	
	Orientation tangentielle de l'outil ACTIVÉE
TangToolOri () ou TangToolOri (0)	Orientation tangentielle de l'outil DÉSACTIVÉE
Format abrégé : TTO . . .	
avec	
<s>	Symétrie de l'outil (en règle générale, le nombre d'arêtes coupantes) Valeur entrée : nombre entier, supérieure à 0 Un outil avec la symétrie <s> atteint une position techniquement équivalente lors d'une rotation de 360 degrés / <s>. Exemples : Outil rectangulaire : <s> = 2, Outil carré : <s> = 4. 1 : Outil asymétrique ou équipé d'une seule arête coupante. > 1 : Outil symétrique qui possède plusieurs arêtes coupantes situées à équidistance. SYM non programmée : effet identique à SYM1.
<a>	Angle d'attaque par rapport à la trajectoire programmée Plage de valeurs : -180 degrés à +180 degrés 0 : L'axe d'outil est positionné à un angle de 0 degré par rapport à la trajectoire programmée. ANG non programmée : effet identique à ANG0.

Tab. 6-182: Syntaxe TangToolOri, TTO

Fonctions CN avec syntaxe de langage standard

- Exemple :**
- G90 est active (programmation absolue).
 - Plan actif : X/Y
 - Axe d'outil : C (modulo 360)
 - Position actuelle : X=0, Y=0, C=0
 - Poinçonnage/Grignotage est désactivé.

:	
N10 TangToolOri(1)	Orientation tangentielle de l'outil ACTIVÉE, avec effet similaire à SYN1 et ANG0.
N20 G1 G91 X10 Y10	Pas encore d'orientation de l'axe C, car le poinçonnage/grignotage est désactivé.
N30 X10 Y10 Punch(1)	Poinçonnage ACTIVÉ L'axe C effectue une rotation à 45°.
N40 Y-10	L'axe C effectue une rotation à -90°. Suivant calcul modulo : C=270°
N50 Punch(0)	Poinçonnage DÉSACTIVÉ
N60 LEN=30 Nibble(1)	Grignotage ACTIVÉ Longueurs de sous-blocs voulues : 30 mm
N70 G2 X114.6 I57.3 J0	Demi-cercle avec une longueur de segment d'arc de 180 mm Le bloc est divisé en 6 blocs partiels d'une longueur d'arc de 30 mm chacun.
N80 TTO()	Orientation tangentielle de l'outil DÉSACTIVÉE
:	

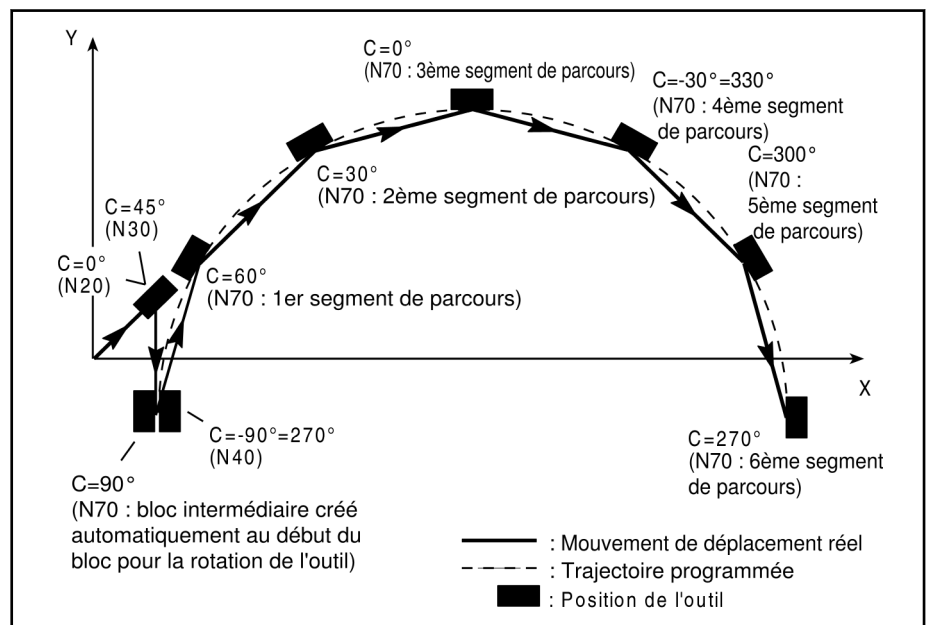


Fig. 6-40: Exemple TangToolOri (TTO)

- Particularités et restrictions :**
- La fonction ne doit pas être programmée conjointement avec la fonction "TangTool" (guidage tangentiel de l'outil, TTL).

Fonctions CN avec syntaxe de langage standard

- Lorsque la fonction est active, aucun "guidage tangentiel de l'outil" (TTL, voir [chap. 6.137 "Guidage tangentiel de l'outil TangTool, TTL" à la page 424](#)) n'est possible.



Les expressions entre parenthèses "(0)" et "(1)" ne sont pas des paramètres de la fonction CN. La valeur numérique entre parenthèses ne peut donc pas être transmise par des variables CPL.

6.139 Remontée d'outil du taraudage "TappRet1, TappRet2"

Description : Si la fonction "Taraudage sans mandrin de compensation" (G63) a été interrompue lors de la pénétration du taraud (par la remise à zéro ou une chute de tension), les fonctions "TappRet1" et "TappRet2" permettent de retirer le taraud de la percée. La programmation peut se faire manuellement ou dans un programme pièce/sous-programme (cycle).

Syntaxe :

TappRet1	Commute au mode d'asservissement en position la/les broches qui tournent de nouveau en mode d'asservissement en vitesse de rotation après la remise à zéro (montée en régime de la commande).
TappRet2 F...	Déclenche le mouvement de remontée effectif. Il est exécuté avec la valeur F programmée.

Tab. 6-183: Syntaxe TappRet1, TappRet2



"TappRet1" doit être programmée avant le mouvement de remontée !

Lors de l'application de la fonctionnalité "Remontée d'outil du taraudage", il faut distinguer 2 cas de figure différents :

- **Remontée d'outil après une remise à zéro** (dégagement automatique) :
Les données mémorisées au début du taraudage sont sauvegardées. Les broches sont commutées au mode asservissement en position à l'aide de la fonction "TappRet1". Si "TappRet2" est programmée ensuite, le taraud retourne à sa position initiale programmée en se dégageant du trou. Seule "TappRet2" doit être programmée conjointement avec l'avance voulue (valeur F).
- **Remontée d'outil après une panne de courant** (dégagement manuel) :
Les informations de remontée mémorisées au début du taraudage sont perdues. Les paramètres correspondants doivent être explicitement programmés avec "TappRet2".

G91 TappRet2 S<vitesse de rotation> F<avance> M3/M4
<axe de perçage> <course incrémentale>

Condition préalable à l'utilisation de TappRet1/TappRet2 :

- Les axes C participants doivent être définis en tant qu'axes ronds sans fin :
 - MP 1001 00004 (/AXSP/Dr[1]/AxFun/ModCalc/ENA_ModCalc)
 - Paramètre Sercos S-0-0076 = Ob1xxxxxxx.
- Au type de fonctionnement auxiliaire Sercos 1 (S-0-0033), le bit 8 pour le changement de type de fonctionnement piloté par l'entraînement ne doit pas être défini :
S-0-0033 = 0b000001011 ou 0b000001100.

Utilisation de sous-programmes pour la remontée du trou :

Fonctions CN avec syntaxe de langage standard

Afin de faciliter l'application de la fonction "Remontée d'outil du taraudage", il est recommandé d'écrire un sous-programme (cycle) pour la remontée automatique d'une part et pour la remontée manuelle d'autre part. Les paramètres machine suivants permettent d'attribuer un code G libre au choix à ces sous-programmes.

- 3090 00001 (/NCP/SubProg/GFun/Fun[1]/GCode)
- 3090 00002 (/NCP/SubProg/GFun/Fun[1]/SubProg)

6.140 Sélection d'une broche pour le taraudage sans mandrin de compensation "TappSp, TSP"

- Description :**
- Définit les broches auxquelles G63 doit se rapporter.
 La fonction "TappSp" est efficace jusqu'à ce qu'elle soit reprogrammée. Elle reste efficace même après une "remise à zéro" !
 - Si "TappSp" n'est pas programmée, alors G63 se rapporte toujours à la 1re broche.

Syntaxe :

TappSp (CAX<i>)	Activer le taraudage pour les différentes broches à l'aide du/des numéros de broches.
TappSp (GRP<j>)	Activer le taraudage pour toutes les broches d'un groupe de broches.
TappSp (GRP<j> { , CAX<i> })	
	Activer le taraudage pour toutes les broches d'un groupe de broches et pour des broches supplémentaires à l'aide du/des numéros de broches.
Format abrégé : TSP (. .)	
avec	
<i>	Numéro de la broche à laquelle se rapporte G63. Plage de valeurs : 1 à n, où n représente le numéro de broche le plus élevé du système (max. 8). Nombre entier. Plusieurs broches sont programmées via plusieurs CAX <i>, séparées par des virgules.
<j>	Numéro du groupe de broches auquel se rapporte G63. Plage de valeurs : 1 à n, où n représente le numéro du groupe de broches le plus élevé (max. 4). Nombre entier.

Tab. 6-184: Syntaxe TappSp (TSP)

Exemple :	N20 TappSp (CAX2)	Sélectionner pour G63 la broche avec le numéro de broche 2.
	:	
	N120 TappSp (CAX2, CAX4, CAX7)	Sélectionner pour G63 les broches avec les numéros de broches 2, 4 et 7.
	:	
	N220 TappSp (GRP2)	Sélectionner pour G63 le groupe de broches avec le numéro 2.
	:	
	N320 TappSp (GRP3, CAX4)	Sélectionner pour G63 le groupe de broches avec le numéro 3, et sélectionner également la broche avec le numéro 4.
	:	

- Particularités et restrictions :**
- Un seul groupe de broches peut être activé pour G63.

Fonctions CN avec syntaxe de langage standard

- Les numéros de broches et éventuellement 1 groupe de broches supplémentaire peuvent être combinés à volonté.

6.141 Comparaison des configurations d'axe "TargetAxisConf, TAX"

Description : Dans l'avance de bloc, la configuration des axes et des noms d'axe au sein des programmes de comparaison et du bloc d'action peuvent être comparés aisément à l'aide de la fonction CN TargetAxisConf (format abrégé : TAX), l.

La commande active la configuration d'axe avant le bloc cible de l'avance de bloc à l'aide de l'ordre TAX. S'il y a trop d'axes disponibles, ils sont retirés. Les axes requis sont conservés sur le canal. Même les noms des axes sont repris.

Si un axe est attribué ailleurs, il n'est pas pris en compte :

- Un axe ne peut pas être accepté s'il est réservé par un autre canal.
- Un axe C ne peut pas être accepté s'il se trouve en mode broche.
- Un axe ne peut pas être retiré s'il fait partie intégrante d'une transformation.

Les axes non pris en compte peuvent ensuite être traités avec des ordres simples. Les informations à ce sujet se trouvent dans les données système (Sys_ChSRun).

TAX peut aussi être appelé à plusieurs reprises, par exemple après la désactivation d'une transformation.

Syntaxe : TAX ou TargetAxisConf

Exemple : Au cours du calcul, le 2e axe système a été retiré du canal, le 3e axe système a reçu le nom "Y", les 5e et 11e axes système ont été acceptés sur le canal avec les noms "Z" et "U".

Avant l'opération de calcul		Au cours de l'opération de calcul	Après l'opération de calcul	
SysNr	Nom	...	SysNr	Nom
1	X	N50 RAX(Y)	1	X
2	Y	N60 ALN(Z, Y)	3	Y
3	Z	N70 WAX(5, Z, 11, U)	5	Z
7	C	...	7	C
			11	U

Programme de comparaison Init :

```

...
N100 X10 Y10 Z10 C10 ; Les 1er, 2e, 3e et 7e axes du système approchent la
                        position 10.
N110 TAX
N120 X20 Y20 Z20 C20 ; Les 1er, 3e, 5e, 7e et 11e axes du système appro-
                        chent la position 20.
...

```

Particularités et restrictions : TAX est autorisé uniquement dans la phase de comparaison de l'avance de bloc, c.-à-d. dans les programmes de comparaison et dans le bloc d'action.

6.142 Définition du TCS dans les coordonnées de programme "TcsDef, TCS"

Description : Génère un système de coordonnées de l'outil (TCS)_p pouvant être décalé et/ou tourné par rapport au dernier TCS actif quand une transformation à 6 axes est active.

La commande numérique convertit en interne les valeurs de coordonnées indiquées pour le TCS_p et les stocke dans la mémoire des corrections d'outil.

Le TCS_p généré peut être supprimé en désactivant la fonction. Dans ce cas, le dernier TCS actuel est réactivé.

- Syntaxe :**
1. Activer le type de transformation d'axe **3333301** si ce n'est pas déjà fait (voir la fonction "Coord(...)", [chap. 6.28 "Sélection d'une transformation d'axe Coord, CRD" à la page 274](#)).
 2. Utiliser la syntaxe indiquée ci-dessous.

TcsDef ({<position>},{<orientation>})	
	Définir et activer la position et l'orientation du TCS _p . Pour la syntaxe détaillée, voir ci-dessous.
TcsDef () ou TcsDef (0)	Supprimer le TCS _p et réactiver le dernier TCS actuel.
Format abrégé : TCS (...)	

Tab. 6-185: Syntaxe TcsDef (TCS)

Syntaxe détaillée :

TcsDef ({<x><px>},{<y><py>},{<z><pz>},{<phi><φ>},{<thêta><θ>},{<psi><ψ>} ou	
TcsDef ({<x><px>},{<y><py>},{<z><pz>},{O (<φ> , <θ> , <ψ>) } ou	
TcsDef ({<x><px>},{<y><py>},{<z><pz>},{Ox (<φ _x > , <θ _x >) Oy (<φ _y > , <θ _y >)	
- ou -	
TcsDef ({<x><px>},{<y><py>},{<z><pz>},{Ox (<φ _x > , <θ _x >) Oz (<φ _z > , <θ _z >)	
- ou -	
TcsDef ({<x><px>},{<y><py>},{<z><pz>},{Oy (<φ _y > , <θ _y >) Oz (<φ _z > , <θ _z >)	
- ou -	
TcsDef ({<x><px>},{<y><py>},{<z><pz>},{Ox (<o11> , <o21> , <o31>) Oy (<o12> , <o22> , <o32>) } ou	
TcsDef ({<x><px>},{<y><py>},{<z><pz>},{Ox (<o11> , <o21> , <o31>) Oz (<o13> , <o23> , <o33>) } ou	
TcsDef ({<x><px>},{<y><py>},{<z><pz>},{Oy (<o12> , <o22> , <o32>) Oz (<o13> , <o23> , <o33>) } ou	
TcsDef ({<x><px>},{<y><py>},{<z><pz>},{ROTAX (<φ _u > , <θ _u >) O (<β>) } ou	
TcsDef ({<x><px>},{<y><py>},{<z><pz>},{ROTAX (<u _x > , <u _y > , <u _z >) O (<β>) }	
avec	
<x>	Nom de coordonnée inscrit dans MP 7080 00010[1]. Valeur par défaut : x
<y>	Nom de coordonnée inscrit dans MP 7080 00010[2]. Valeur par défaut : y

Fonctions CN avec syntaxe de langage standard

<z>	Nom de coordonnée inscrit dans MP 7080 00010[3]. Valeur par défaut : z
<px>, <py>, <pz>	Coordonnées cartésiennes absolues des adresses d'axe <x>, <y> et <z> par rapport au PCS actuel. Les valeurs définissent l'origine du nouveau TCS _p . La programmation AC/IC est permise.
<phi>	Nom d'angle saisi dans MP 7080 00010[4]. Valeur par défaut : phi
<thêta>	Nom d'angle saisi dans MP 7080 00010[5]. Valeur par défaut : thêta
<psi>	Nom d'angle saisi dans MP 7080 00010[6]. Valeur par défaut : psi
<φ>,<θ>,<ψ>	Angle d'Euler absolu en degrés, par rapport au PCS actuel. Les valeurs définissent l'orientation du nouveau TCS _p . La programmation AC/IC est permise. Plage de valeurs : 0° ≤ φ < 360° 0° ≤ θ ≤ 180° 0° ≤ ψ ≤ 360° Les valeurs situées en dehors de cette plage sont automatiquement converties dans l'intervalle correspondant.
<φ..>,<θ..>	Valeurs absolues des angles en degrés. Plage de valeurs : 0° ≤ φ.. < 360° 0° ≤ θ.. ≤ 180° Si la valeur programmée pour θ.. se situe en dehors de la plage de valeurs, alors elle est automatiquement convertie dans l'intervalle indiqué.
Ox (<o11>, <o21>, <o31>) Oy (<o12>, <o22>, <o32>) Oz (<o13>, <o23>, <o33>) Ox (<φ _{xxOy (<φ_{yyOz (<φ_{zz<td>Orientation à l'aide de la fonction Ox(..), Oy(..), Oz(..). Ox(..) définit par exemple la direction de la coordonnée x du TCS_p dans le système de coordonnées de référence PCS. Il en va de même pour Oy(..) et Oz(..). La direction peut être définie soit à l'aide des angles polaires correspondants <φ..> et <θ..>, soit au moyen des composantes cartésiennes des vecteurs colonnes du tenseur d'orientation TCS_p. Seule les cotes absolues sont permises. Les valeurs des composantes des vecteurs colonnes (o..) sont automatiquement standardisées à 1.</td>}}}	Orientation à l'aide de la fonction Ox(..), Oy(..), Oz(..). Ox(..) définit par exemple la direction de la coordonnée x du TCS _p dans le système de coordonnées de référence PCS. Il en va de même pour Oy(..) et Oz(..). La direction peut être définie soit à l'aide des angles polaires correspondants <φ..> et <θ..>, soit au moyen des composantes cartésiennes des vecteurs colonnes du tenseur d'orientation TCS _p . Seule les cotes absolues sont permises. Les valeurs des composantes des vecteurs colonnes (o..) sont automatiquement standardisées à 1.
ROTAX (<φ _{uu<td>Définition de l'axe de rotation à l'aide d'angles polaires (φ_u, θ_u).</td>}	Définition de l'axe de rotation à l'aide d'angles polaires (φ _u , θ _u).
ROTAX (<u _{xyz<td>Définition de l'angle de rotation à l'aide des composantes cartésiennes <u_{xyz}</td>}	Définition de l'angle de rotation à l'aide des composantes cartésiennes <u _{xyz}
O (<β>)	Indique, en degrés, l'ange incrémental <β> duquel le tenseur d'orientation TCS _p doit être déplacé autour de l'axe de rotation à partir du tenseur d'orientation du dernier TCS actuel. Les valeurs supérieures à 360 degrés sont permises. Le sens de rotation peut être sélectionné à l'aide des signes (+/-).

Tab. 6-186: Syntaxe détaillée TcsDef (TCS)

Fonctions CN avec syntaxe de langage standard

- Particularités et restrictions :**
- Le type de transformation d'axe actif 3333301 est nécessaire.
 - Si la fonction "Coord(...)" (voir [chap. 6.28 "Sélection d'une transformation d'axe Coord, CRD" à la page 274](#)) est programmée de nouveau, un TCS_p généré plus tôt est automatiquement supprimé et le dernier TCS actif réactivé.
 - Le tenseur d'orientation ne peut pas être calculé si deux vecteurs colonnes sont parallèles ou antiparallèles lors de la programmation des vecteurs colonnes du tenseur.
- Une erreur d'exécution est signalée.



Les expressions entre parenthèses "(0)" et "(1)" ne sont pas des paramètres de la fonction CN. La valeur numérique entre parenthèses ne peut donc pas être transmise par des variables CPL.

6.143 Fonctions additionnelles pour le filetage "ThreadSet, TST"

6.143.1 Généralités

Description : "ThreadSet" permet d'adapter temporairement certaines zones partielles de G33.

La commande masque alors les valeurs statiques enregistrées dans les paramètres machine.

"ThreadSet" permet :

- d'adapter la dynamique et le mouvement de remontée ;
- de commuter le mode de fonctionnement de la broche (asservissement en vitesse de rotation, asservissement en position) ;
- d'initialiser un signal au niveau de l'interface du canal (configurable au sein des signaux "Fonction active" via les paramètres machine).

La remise à zéro ou M30 :

- annule les réglages masqués par "ThreadSet" ;
- supprime un signal IF initialisé par "ThreadSet" ;
- commute de nouveau la broche principale au mode d'asservissement en vitesse de rotation, dans la mesure où elle était auparavant asservie en position par "ThreadSet(SPC1)".



Toutes les fonctions partielles décrites ci-dessous peuvent être programmées conjointement dans un bloc "ThreadSet", séparées par des virgules.

6.143.2 Configuration des données de remontée ThreadSet(...)

Syntaxe :

ThreadSet (RD(<valeur AP>,<valeur AS>))	
Format abrégé : TST (. .)	
avec	
<valeur AP>	Course de remontée (incrémentale en mm) en direction de l'axe primaire du plan actuellement sélectionné (G17, G18, G19, G20). Cette valeur doit toujours être programmée, mais elle n'est pertinente que pour les filets longitudinaux et coniques.

Fonctions CN avec syntaxe de langage standard

<valeur AS>	Course de remontée (incrémentale en mm) en direction de l'axe secondaire du plan actuellement sélectionné. Cette valeur doit toujours être programmée, mais elle n'est pertinente que pour les filets plans et coniques.
ou :	
ThreadSet (RD(, -1))	
-1	Les données de remontée de MP 7050 00645 et MP 7050 00650 prennent effet.

Tab. 6-187: Syntaxe ThreadSet (TST), configuration des données de remontée outil

6.143.3 Configuration de la dynamique ThreadSet(DYN)

Syntaxe :

ThreadSet (DYN({<saut>},{<accél>},{<accél frein>})) ou TST (..)	
avec	
<saut>	Vitesse de saut max. admissible en mm/min Saisir "-1" réactive MP 7050 00610.
<accél>	Accélération en m/s ² Saisir "-1" réactive MP 7050 00615.
<accél frein>	Accélération de freinage en m/s ² Saisir "-1" réactive MP 7050 00620.

Tab. 6-188: Syntaxe ThreadSet (TST), configuration de la dynamique

6.143.4 Libération de la remontée d'outil ThreadSet(ROn)

Syntaxe :

ThreadSet (ROn<statut>) ou TST (..)	
avec	
<statut>	0 : désactiver la remontée rapide 1 : activer la remontée rapide

Tab. 6-189: Syntaxe ThreadSet (TST), libération de la remontée d'outil

6.143.5 Commutation du mode de fonctionnement de broche Thread-Set(SPC)

Syntaxe :

ThreadSet (SPC<statut>) ou TST (..)	
avec	
<statut>	0 : Commuter la broche principale au mode d'asservissement en vitesse de rotation. 1 : Commuter la broche principale au mode de régulation de position selon le paramétrage défini dans 7050 00600 [3].

Tab. 6-190: Syntaxe ThreadSet (TST), configuration du mode de fonctionnement de la broche

Pour les broches principales, voir [chap. 6.70 "Changement de broche principale MainSp, MSP" à la page 323](#).

6.143.6 Influence sur le signal IF du canal ThreadSet(TCI)

Syntaxe :

ThreadSet (TCI<statut>) ou TST (..)	
avec	
<statut>	> 0 : supprime le signal IF du canal. 1 : définit le signal IF du canal.

Tab. 6-191: Syntaxe ThreadSet (TST), influence sur le signal IF du canal



Le signal influencé au niveau de l'interface du canal peut être configuré au sein des signaux "Fonction active" via les paramètres machine.

6.144 Décalage des coordonnées du programme "Trans, TRS", décalage additif des coordonnées du programme "ATrans, ATR"

Description : "Trans" et "ATrans" décalent le système actif des coordonnées du programme par rapport au système actuel des coordonnées de la pièce à usiner.

Les deux décalages peuvent être activés et désactivés indépendamment l'un de l'autre. Si les deux sont actifs, ils ont un effet additif.

De ce fait, il est possible d'exécuter un programme pièce n'importe où dans le système de coordonnées de la pièce à usiner sans modifier le contour programmé. Les aides à la saisie construites sur le système actif des coordonnées du programme (par ex. décalage programmé du contour, mise à l'échelle, fonction miroir, rotation) sont indépendantes d'un décalage des coordonnées du programme et n'ont donc pas besoin d'être ajustées.

Fonctions CN avec syntaxe de langage standard

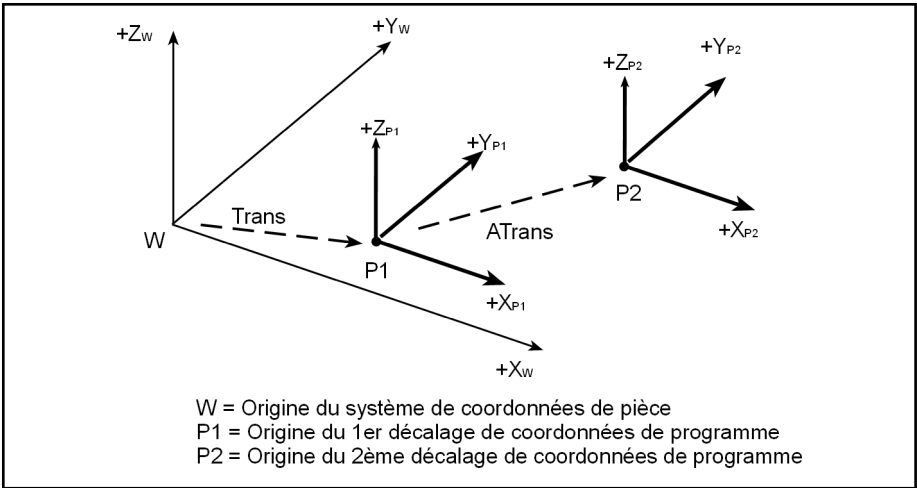


Fig. 6-41: Décalage des coordonnées du programme, décalage additif des coordonnées du programme

Les points suivants s'appliquent ici :

La programmation d'un décalage de coordonnées du programme n'entraîne aucun mouvement de déplacement.

Syntaxe :

Trans (<coordonnées>)	Décalage des coordonnées du programme ACTIVÉ L'origine du système de coordonnées du programme est décalée par rapport au système de coordonnées de la pièce à usiner selon les <coordonnées> programmées. Le décalage a un effet additif à "ATrans". Plusieurs axes de coordonnées peuvent être programmés entre parenthèses, séparés par des virgules. (exemple : TRS (X100, Y50, Z50)).
Trans () ou Trans (0)	Tous les décalages de coordonnées du programme DÉSACTIVÉS
Format abrégé : TRS (. .)	
ATrans (<coordonnées>)	Décalage additif des coordonnées du programme ACTIVÉ L'origine du système de coordonnées du programme est décalée par rapport au système de coordonnées de la pièce à usiner selon les <coordonnées> programmées. Le décalage a un effet additif à "Trans". Plusieurs axes de coordonnées peuvent être programmés entre parenthèses, séparés par des virgules. (exemple : ATR (X10, Z70)).
ATrans () ou ATrans (0)	Décalage additif des coordonnées du programme DÉSACTIVÉ
Format abrégé : ATR (. .)	

Tab. 6-192: Syntaxe Trans (TRS) et ATrans (ATR)

Fonctions CN avec syntaxe de langage standard

Exemple :	N10 ATR() TRS()	Les deux décalages de coordonnées du programme DÉSACTIVÉS.
	:	L'origine du système de coordonnées du programme est définie sur la position X0 Y0 Z0 dans le système actuel des coordonnées de la pièce à usiner.
	N30 TRS(X10,Y10,Z50)	L'origine du système de coordonnées du programme est définie sur la position X10 Y10 Z50 dans le système actuel des coordonnées de la pièce à usiner.
	:	L'origine du système de coordonnées du programme est définie sur la position X30 Y20 Z50 dans le système actuel des coordonnées de la pièce à usiner.
	N50 ATR(X20,Y10)	L'origine du système de coordonnées du programme est définie sur la position X60 Y20 Z80 dans le système actuel des coordonnées de la pièce à usiner.
	N70 ATR(X50,Z30)	Décalage des coordonnées du programme DÉSACTIVÉ
	:	L'origine du système de coordonnées du programme est définie sur la position X50 Y10 Z30 dans le système actuel des coordonnées de la pièce à usiner.
	N90 TRS()	

6.145 Compensation volumétrique

6.145.1 Activation et désactivation "VCPEna"

Description : La fonction CN permet d'activer et de désactiver la compensation volumétrique sur le canal. Pour l'activation, un fichier tabulaire doit d'abord être chargé avec la fonction CN VCPLoad ([chap. 6.145.2 "Chargement du tableau VCPLoad" à la page 440](#)).

Pour chaque canal, il existe deux espaces mémoire destinés aux fichiers tabulaires.

Syntaxe :	VCPEna (<index tab>)	Activer la compensation
	VCPEna	Activer la compensation du tableau 1
	VCPEna (0)	Désactiver la compensation
	VCPEna ()	
	avec	
	<index tab>	Numéro de l'espace mémoire dont le tableau doit être utilisé pour la compensation Plage de valeurs : 1 ou 2

Tab. 6-193: Syntaxe VCPEna

Exemple 1 : *Programme:*

```
N10 VCPLoad(/Machine.ofm) ; charger le fichier de tableau dans la mémoire
N20 VCPEna                ; active la compensation, tableau de 1
N30 ...                   ; édition
N40 VCPEna ()              ; désactiver la compensation
```

Exemple 2 : *Programme:*

```
N10 VCPLoad(/mnt/ToolHead1.ofm) ; charger le fichier dans la mémoire 1
N20 VCPLoad(/mnt/ToolHead2.ofm, 2) ; charger le fichier dans la mémoire 2
N30 VCPEna                      ; compensation activée, tableau 1
N40 ...                          ; édition
```

Fonctions CN avec syntaxe de langage standard

```

N50 VCPEna () ; désactiver la compensation
N60 ToolHeadChange (2) ; remplacer la tête d'outil 2
N70 VCPEna (2) ; compensation activée, tableau 2
N80 ... ; édition
N90 VCPEna () ; désactiver la compensation

```

- Particularités et restrictions :**
- Lors de l'activation et de la désactivation de la compensation, la valeur de correction doit être déplacée en max. 3 s. Par conséquent, aucun mouvement de la machine n'a lieu.
 - La durée du mouvement dépend de la valeur de correction et de la vitesse (sûre) autorisée.
 - Si les axes requis ne sont pas sur le canal ou si la transformation d'axe AT1 n'est pas active, un avertissement est généré lors de l'activation et la compensation n'a pas lieu.
 - Si la transformation d'axe AT1 est commutée ou désactivée, la compensation est désactivée et un avertissement est généré.
 - Si un axe compensé est retiré du canal, un avertissement s'affiche et la compensation est désactivée.
 - La fonction CN « ADJ(VCP) » peut être utilisée pour comparer VCPEna et VCPLoad après une avance de bloc.

6.145.2 Chargement du tableau "VCPLoad"

Description : La fonction CN permet de charger un fichier tabulaire destiné à la compensation volumétrique. La compensation est activée et désactivée avec la fonction CN VCPEna ([chap. 6.145.1 "Activation et désactivation VCPEna" à la page 439](#)).

Pour chaque canal, il existe deux espaces mémoire destinés aux fichiers tabulaires.

Syntaxe :	VCPLoad (<NomTab> {, <index tab> })	Chargement du fichier tabulaire
	avec	
	<NomTab>	Nom du fichier, y compris le chemin d'accès au fichier tabulaire à charger
	<index tab>	Numéro de l'espace mémoire dont le tableau doit être utilisé pour la compensation Plage de valeurs : 1 ou 2, par défaut : 1

Tab. 6-194: Syntaxe VCPLoad

Exemple 1 : Voir VCPLoad ([chap. 6.145.1 "Activation et désactivation VCPEna" à la page 439](#))

- Particularités et restrictions :**
- Si le tableau est actuellement actif sur l'emplacement de mémoire indiqué, un avertissement est généré et le tableau programmé n'est pas chargé.
 - Si un nouveau tableau doit être chargé sur le même emplacement de mémoire, immédiatement après la désactivation, un « WAIT » et un bref temps d'attente pour le retour de la correction doivent être programmés entre-deux pour permettre la synchronisation.
 - Le fichier de tableau peut se trouver n'importe où.
 - La mémoire mise à disposition pour un fichier tabulaire est limitée. Si le fichier est trop volumineux pour la mémoire, un avertissement est généré.

6.146 Entraînements virtuels "VirtAxisPos, VAP, VAPR"

Description : Définit la position des axes synchrones virtuels sur le canal actuel. Utiliser la syntaxe VAPR permet en plus de définir le signal d'interface "Point de référence connu".



Pour plus d'informations sur les entraînements virtuels, voir la documentation "Description des fonctions IndraMotion MTX".

Syntaxe :

VirtAxisPos (<axe 1><valeur>,<axe n><valeur>, ...)

Format abrégé : VAP (...)

avec

<axe x>	Adresse de l'axe virtuel

<valeur>	Information de position pour <axe x>

Tab. 6-195: Syntaxe VirtAxisPos (VAP)

Exemple :

VAP (VX150) Position de l'axe VX sur 150 mm

Particularités et restrictions : La programmation incrémentale locale (IC...) n'est pas permise.

6.147 Diagnostic du guidage de vitesse "VREC_START", "VREC_STOP"

Description : Lors du traitement des programmes de surfaces en formes libres avec le guidage de vitesse G8, de nombreuses grandeurs ont un impact direct ou indirect sur le profil de vitesse qui en résulte. Le diagnostic du guidage de vitesse permet à l'utilisateur de déterminer quelle grandeur a, à une position définie dans le programme pièce, une influence dominante sur la vitesse sur trajectoire atteinte.

Lors de la préparation des blocs, la fonction **VREC_START** consigne des données qui sont calculées bloc par bloc par le guidage de vitesse et ont un impact sur le profil de vitesse. Le nom et l'emplacement de stockage (répertoire) du fichier de protocole, ainsi que les unités physiques sur lesquelles les données doivent être consignées dans un procès-verbal, sont également indiqués lors de l'activation.

Le fichier de diagnostic généré peut être exploité hors ligne. Les données enregistrées permettent de déduire les paramètres à optimiser pour atteindre un profil de vitesse plus favorable.

Lors de l'activation de la fonction VREC, les valeurs limites de l'axe valides à ce moment sont émises sous l'intitulé "Limits of channel axes at VREC activation:" pour tous les axes situés sur le canal :

Index de colonne :	Titres de colonne :	Signification :
1	SysAxNo	Numéro d'axe du système
2	MaxAxVel	Vitesse maximale de l'axe
3	MaxAxAccUp	Accélération maximale de l'axe dans les phases d'accélération sur trajectoire telle qu'elle peut être programmée avec la fonction CN NcfAxAccUpSlope.

Fonctions CN avec syntaxe de langage standard

Index de colonne :	Titres de colonne :	Signification :
4	MaxAxAcc	Accélération maximale de l'axe dans les phases de freinage sur trajectoire
5	MaxAxJerk	Secousse maximale de l'axe

Tab. 6-196: Structure du fichier d'enregistrement

Ensuite, les données suivantes sont enregistrées bloc par bloc sous l'intitulé "Resulting dynamic limits:" :

Index de colonne :	Titres de colonne :	Signification :
1	Way	Course d'interpolation (longueur de trajectoire) dans le bloc CN
2	F100	Vitesse programmée, en intégrant le mouvement d'orientation (pour G93 et G94 dans l'unité sélectionnée, pour G95 en fonction des rotations de la broche).
3	Fmax	Vitesse sur trajectoire maximale, en intégrant le mouvement d'orientation (limité)
4	F100(TCP)	Vitesse sur trajectoire programmée de la pointe de l'outil (pour G93 et G94 dans l'unité sélectionnée, pour G95 en fonction des rotations de la broche).
5	Fmax(TCP)	Vitesse sur trajectoire maximale de la pointe de l'outil (limitée).
6	Fmax-Limiter	Propriété et numéro de l'axe système qui limite la vitesse sur trajectoire max. à Fmax. La vitesse max. de l'axe (Vel), l'accélération max. de l'axe (Acc) et la secousse d'axe max. (Jerk) sont prises en compte en tant que propriétés.
7	AccUp	Accélération sur trajectoire maximale de la pointe de l'outil (limitée). Cette valeur peut diverger de AccDown uniquement si une ou plusieurs accélérations max. de l'axe ont été modifiées par la fonction CN NcfAxAccUpSlope.
8	AccDown	Accélération de freinage sur trajectoire maximale de la pointe de l'outil (limitée).
9	Acc-Limiter	Propriété et numéro de l'axe système qui limite l'accélération de freinage sur trajectoire max. à AccDown. L'accélération max. de l'axe (Acc) et la secousse max. de l'axe (Jerk) sont prises en compte en tant que propriétés.
10	Vend	Vitesse sur trajectoire maximale à la fin du bloc ou de la rampe de freinage actuelle, en intégrant le mouvement d'orientation
11	Vend(TCP)	Vitesse sur trajectoire maximale de la pointe de l'outil à la fin du bloc ou de la rampe de freinage actuelle.
12	MaxStepVel	Vitesse maximale sur trajectoire qui serait théoriquement possible lors du passage au bloc suivant.
13	0Vel	yes : Pour le bloc CN : l'attribut définit que l'enchaînement de blocs suivant doit se faire à $v = 0$. Ledit attribut peut être paramétré, par exemple, pour les blocs devant être déplacés avec un arrêt précis.

Fonctions CN avec syntaxe de langage standard

Index de colonne :	Titres de colonne :	Signification :
14	Look ahead state	État de prévisualisation, y compris le nombre de blocs actuellement prévisualisés : 0 : Bloc fictif : aucune prévisualisation 1 : Bloc de déplacement sans prévisualisation, par ex. en cas d'arrêt précis ou si le bloc suivant est un bloc fictif. 2 : La prévisualisation réglée suffit à évaluer la distance de freinage. 3 : La prévisualisation réglée suffit à évaluer la distance de freinage. Cependant, un enchaînement de blocs $v = 0$ a été détecté au sein de la zone de prévisualisation.
15	Insert state	État d'insertion de bloc : - : aucun bloc d'insertion 1 : bloc d'insertion ordinaire 2 : First split block (pas de bloc d'insertion) 3 : Split block (bloc d'insertion) 4 : Last split block (bloc d'insertion)
16	Addr	Adresse de mémoire du bloc CN au sein de la commande
17	NC-Source	Texte source du bloc CN (parties CPL dissociées)

Tab. 6-197: Structure du fichier d'enregistrement

Syntaxe :

VREC_START({chemin d'accès/}<nom de fichier>{, unités})	
	Lancer l'enregistrement du diagnostic pour le guidage de vitesse.
avec :	
{chemin d'accès/}<nom de fichier>	Répertoire et nom du fichier dans lequel l'enregistrement du diagnostic doit être stocké.

Fonctions CN avec syntaxe de langage standard

Units	Le stockage est effectué dans les unités suivantes : Units = 0 ou sans prog. : Trajectoires en incréments Vitesses en incréments pour chaque cycle lpo Accélérations en incréments pour chaque cycle lpo² Secousses en incréments pour chaque cycle lpo³ Units = 1 et G70 actif : Trajectoires en pouces Vitesses en pouces/min Accélérations en pouces/sec² Secousses en pouces/sec³ Units = 1 et G71 actif : Trajectoires en mm Vitesses en mm/min Accélérations en m/sec² Secousses en m/sec³
VREC_STOP	Terminer l'enregistrement du diagnostic

Tab. 6-198: Syntaxe VREC_START, VRECSTOP

Exemple :

```
...
N30 VREC_START (/mnt/VDIAG1.txt)
...
N110 VREC_STOP
...
N300 VREC_START (/mnt/VDIAG2.txt, 1)
...
N500 VREC_STOP
...
M30
```

Particularités et restrictions :

Si le diagnostic du guidage de vitesse est utilisé, veiller à ce que la mémoire disponible soit suffisante pour le fichier de diagnostic. En général, la mémoire requise pour cela est 3 à 4 fois supérieure à la mémoire requise pour le segment correspondant du programme pièce à enregistrer.

6.148 Reprise d'axe et attente si nécessaire "WaitAxis, WAX"

Description :

Reprend un axe asynchrone sur le canal appelant. L'axe asynchrone devient ainsi un axe synchrone.
L'axe peut alors être programmé sur le canal actuel via son nom physique ou logique.

 Pour obtenir des informations détaillées sur la fonctionnalité "Transfert d'axe", voir la documentation "Description des fonctions IndraMotion MTX".

Fonctions CN avec syntaxe de langage standard

Syntaxe :

WaitAxis (<NPA> <IPA> , {<NLA>}{ , <NPA> <IPA> , {<NLA>}}...)	
Format abrégé : WAX (. .)	
avec	
<NPA>	Nom physique de l'axe Définit l'axe devant être repris sur le canal actuel.
<IPA>	Index physique de l'axe Effet similaire à <NPA>
<NLA>	Nom logique de l'axe Dans la mesure où celui-ci a été programmé, l'axe à reprendre sur le canal actuel reçoit le nom logique <NLA>. <NLA> doit être défini dans le paramètre CHAN/Ch/Coord/Acs/ChAx/ChAxName (MP 7010 00010, Nom de l'axe du canal) ou dans le paramètre CHAN/Ch/Coord/Acs/OptAxName/OptChAxName (MP 7010 00020, Désignation optionnelle de l'axe).

Tab. 6-199: Syntaxe WaitAxis (WAX)

Exemple :

N030 WAX (YP , , ZP , Z) : Le cas échéant, la préparation des blocs attend l'arrêt des axes physiques YP et ZP. Ensuite, les axes sont repris sur le canal appelant.

Tandis que YP est aussi adressé sur le canal appelant en tant qu'adresse YP, ZP reçoit l'adresse Z sur le canal appelant.

Particularités et restrictions :

- Si un axe à reprendre n'est pas encore à l'arrêt, la préparation des blocs attend l'arrêt de l'axe. Ensuite, l'axe est repris.
Contrairement à la fonction "GetAxis" (voir [chap. 6.51 "Reprise d'axe GetAxis, GAX" à la page 301](#)), aucun message d'erreur n'est émis et le programme n'est pas interrompu.
- Les positions d'axes d'un même bloc doivent toujours être programmées après WaitAxis(...) et leur programmation ne peut s'effectuer que si aucune transformation d'axe n'est active.
- Les axes à reprendre ne doivent participer à aucune zone de surveillance active (voir [chap. 6.7 "Surveillance de zone Area, ARA" à la page 230](#)).

6.149 Écriture de paramètres Sercos avec un numéro d'identification Sercos élargi "WritelD3, WID3"

6.149.1 Généralités

Description : Écrit le paramètre d'entraînement Sercos programmé (S-x-xxxx-xx.xx, P-x-xxxx.xx.xx) dans un ou plusieurs entraînements.

Outre des valeurs individuelles, il est possible d'inscrire des listes de valeurs complètes contenant 16 valeurs au maximum, ou des listes d'identifiants contenant 8 numéros d'identification au maximum.

Conditions préalables :

- Le paramètre doit être valide et modifiable dans la phase 4 Sercos.
- La valeur du paramètre se situe dans la plage autorisée.

Fonctions CN avec syntaxe de langage standard

- L'entraînement est connecté à la commande via l'interface Sercos ; la communication cyclique entre la commande et l'entraînement est active.



L'ordre CPL "SCS3" permet de lire les paramètres d'entraînement Sercos avec un numéro d'identification Sercos élargi.



La transmission de données peut être accélérée en utilisant le canal multiplex Sercos.



Des modifications inadaptées ou arbitraires des paramètres d'entraînement Sercos risquent d'endommager la pièce à usiner et/ou la machine, et de provoquer des réactions imprévues et dangereuses de la machine.

Pour une utilisation correcte, des informations sur les paramètres d'entraînement Sercos disponibles dans l'entraînement sont nécessaires. Consultez la documentation de l'entraînement à ce sujet.

6.149.2 Écriture d'un paramètre dans un ou plusieurs entraînements (pour les broches et les axes asynchrones)

Syntaxe :

WriteID3 (<par>, DRIVE (<AA1>, <Ang1>{<AAn>, <Angn>} ...))	
Format abrégé : WID3 (...)	
avec	
<par>	Paramètre standard (S-x-xxxx.xx.xx) - ou - Paramètre spécifique au produit (P-x-xxxx.xx.xx)
<AA1>...<AAn>	Index physique de l'axe ou nom physique de l'axe de l'entraînement Sercos. La syntaxe DRIVE(...) permet de définir un maximum de 8 entraînements.
<Ang1>...<Angn>	Valeur devant être inscrite dans <par>

Tab. 6-200: Syntaxe 2 WriteID3 (WID3)

Exemple :

N100 WID3 (S-0-0104.0.0, DRIVE (1, 2.22))

Écrire la valeur 2.22 dans S-0-0104.0.0 de l'entraînement 1

6.149.3 Écriture d'un paramètre dans un ou plusieurs entraînements (pour les axes synchrones)

Syntaxe :

WriteID3 (<par>, <ES1><Ang1>{, <ESn><Angn>}...)	
Format abrégé : WID3 (...)	
avec	
<par>	Paramètre standard (S-x-xxxx.xx.xx) - ou - Paramètre spécifique au produit (P-x-xxxx.xx.xx)

Fonctions CN avec syntaxe de langage standard

<ES1>...<ESn>	Adresse d'axe logique de l'entraînement Sercos
<Ang1>...<Angn>	Valeur devant être inscrite dans <par>

Tab. 6-201: Syntaxe 1 WritelD3 (WID3)

Exemple : N100 WID3 (S-0-0159.0.0,X10,Y20,Z30) Écrire S-0-0159.0.0 :
dans l'entraînement de l'axe X :
10
dans l'entraînement de l'axe Y :
20
dans l'entraînement de l'axe Z :
30

6.149.4 Écriture de numéros d'identification élargis d'une liste ID dans un entraînement

Syntaxe :

WriteID3(<par>,EID_LIST(<Axe1>,<id1>{,<idn>}...))	
Format abrégé : WID3 (...)	
avec	
<par>	Paramètre standard (S-x-xxxx.xx.xx) - ou - Paramètre spécifique au produit (P-x-xxxx.xx.xx) contenant une liste ID élargie.
<Axe1>...<Axe n>	Index physique de l'axe - ou - Nom physique de l'axe de l'entraînement Sercos.
<id1>...<idn>	8 numéros d'identification au maximum qui doivent être inscrits dans <par>.

Tab. 6-202: Syntaxe 4 WritelD3 (WID3)

Exemple : N100 WID3 (S-0-1050.0.20,EID_LIST(2,P-0-556,S-0-1050.0.6)) Écriture de deux numéros d'identification élargis dans S-0-1050.0.20 dans l'entraînement de l'axe 2.

- Particularités et restrictions :**
- La fonction ne doit jamais être programmée lors de l'usinage d'un contour.
 - L'inscription des paramètres programmés ne s'effectue qu'après l'immobilisation de l'entraînement correspondant.

6.149.5 Écriture des valeurs d'une liste de paramètres dans un entraînement

Syntaxe :

WriteID3(<par>,LIST(<Axe1>,Wl{,<Angn>}...))	
Format abrégé : WID3 (...)	
avec	
<par>	Paramètre standard (S-x-xxxx.xx.xx) - ou - Paramètre spécifique au produit (P-x-xxxx.xx.xx) contenant une liste de valeurs.

Fonctions CN avec syntaxe de langage standard

<Axe1>...<Axe n>	Index physique de l'axe ou nom physique de l'axe de l'entraînement Sercos.
<Ang1>...<Ang n>	16 valeurs au maximum qui doivent être inscrites dans <par>.

Tab. 6-203: Syntaxe 3 Writeld3(WID3)

Exemple :

N100 WID3	Écriture de 2 valeurs dans
(P-0-4006.0.0, LIST(1, 0.5, 0.2))	P-0-4006.0.0 dans l'entraînement 1

6.150 Activation des tables de décalage d'origine "ZoTSel, ZOS"

Description : Active une table de décalage d'origine (table NPV). Les tables NPV sont enregistrées au format XML dans le système de fichiers de la commande.

Syntaxe :

ZoTSel ({<chemin>}<nom de fichier>)	
Format abrégé : ZOS (. .)	
avec	
<chemin>	Indication optionnelle du chemin d'accès au répertoire dans lequel <nom de fichier> est enregistré. Si aucun chemin n'est indiqué, la recherche s'effectue sous le chemin "/database". Si <nom de fichier> ne s'y trouve pas, la commande utilise le chemin de recherche des sous-programmes afin de rechercher <nom de fichier> dans d'autres répertoires.
<nom de fichier>	Nom de fichier de la table NPV, extension du fichier incluse Les tables ayant des noms standards (ZO<numéro>.zot) peuvent être activées directement via leur numéro, par ex. ZoTSel(5) active le tableau ZO5.zot

Tab. 6-204: Syntaxe ZoTSel (ZOS)

Exemple :

:	
N030 ZOS (npvtab.zot)	Recherche d'abord la table NPV "npvtab.zot" dans le répertoire "/database" ; ensuite, si la recherche a été infructueuse, dans le chemin de recherche pour les sous-programmes. La première table NPV nommée "npvtab.zot" qui est trouvée est activée.
:	
N130 ZOS (/mnt/np.zot)	Recherche et active la table NPV "np.zot" dans le répertoire "/mnt". Si elle ne s'y trouve pas, un message d'erreur est généré.
:	

Particularités et restrictions : Lors de l'activation de la table, la commande peut vérifier en option si la configuration d'axe actuelle du canal correspond au contenu de la table (option "Attribution fixe" ; cette fonction est activée via l'éditeur de tables ou lors de la première génération de la table).

Si elle ne correspond pas au contenu de la table et que l'option "Attribution fixe" est activée, le programme est interrompu et un message d'erreur est généré.



Pour de plus amples informations sur la création et l'édition des tables NPV, consultez le mode d'emploi de la commande !

7 Ordres CPL

Certaines propriétés fondamentales des ordres CPL sont présentées dans un champ de propriétés placé avant la description d'un ordre. Les champs applicables à chaque ordre sont inscrits sur fond gris.

<type>	READ	WRITE	FUNCT	PROC	WAIT	ERRNO	!

Légende :

- **<type>** : Type de données CPL renvoyées par l'ordre CPL (return value), dans la mesure où l'ordre possède une valeur de retour : **DOUBLE**, **REAL**, **INT**, **BOOL**, **CHAR**
Si le type de données de la valeur de retour de l'ordre peut varier, le champ de propriétés présente la valeur **MULTYP**.
Si la valeur de retour de l'ordre fournit une structure ou une sous-structure de données système, le champ de propriétés présente la valeur **STRUCT**.
- **READ** : Ordre CPL permettant de lire une donnée interne à la commande, par ex. une position, un registre de correction ou l'élément d'une table.
- **WRITE** : Ordre CPL permettant d'écrire une donnée interne à la commande, par ex. un registre de correction ou l'élément d'une table.
- **FUNCT(ion)** : Ordre CPL déclenchant un calcul ou une action et fournissant une valeur de retour (return value).
- **PROC(edure)** : Ordre CPL déclenchant une action, mais ne fournissant aucune valeur de retour.
- **WAIT** : À des fins de synchronisation, le recours aux ordres CPL requiert la programmation judicieuse en amont d'un ordre WAIT puisqu'il est nécessaire, par exemple, d'accéder aux données du moment actif.
- **ERRNO** : Ordre CPL prenant en charge la programmation du paramètre ERRNO.
- **!** : Particularité devant être observée lors de l'utilisation de l'ordre CPL. La description qui suit présente cette particularité explicitement.

7.1 Axes et coordonnées

7.1.1 Vue d'ensemble

Le langage CPL met à disposition des ordres permettant d'accéder aux données des axes et des coordonnées.

Ces ordres se distinguent comme suit :

- Noms et propriétés
(AXINF, AXADR\$, AXP)
- Lecture des positions de coordonnées et d'axes
(PCS, WCS, MCS, ACS, SPOS, APOS),
- Lecture à partir du palpeur de mesure
(PCSPROBE, PROBE)

Les coordonnées interpolées au cours de l'exécution du programme sont toujours programmées dans le programme pièce. Les transformations d'axe calculent les valeurs de consigne respectives des axes concernés à partir des valeurs des coordonnées actuelles (coordonnées tridimensionnelles et coordonnées de machine).



Vous trouverez des informations détaillées sur le thème "Coordonnées, axes et transformations" dans la documentation "Description des fonctions IndraMotion MTX".

Diagramme de flux de données

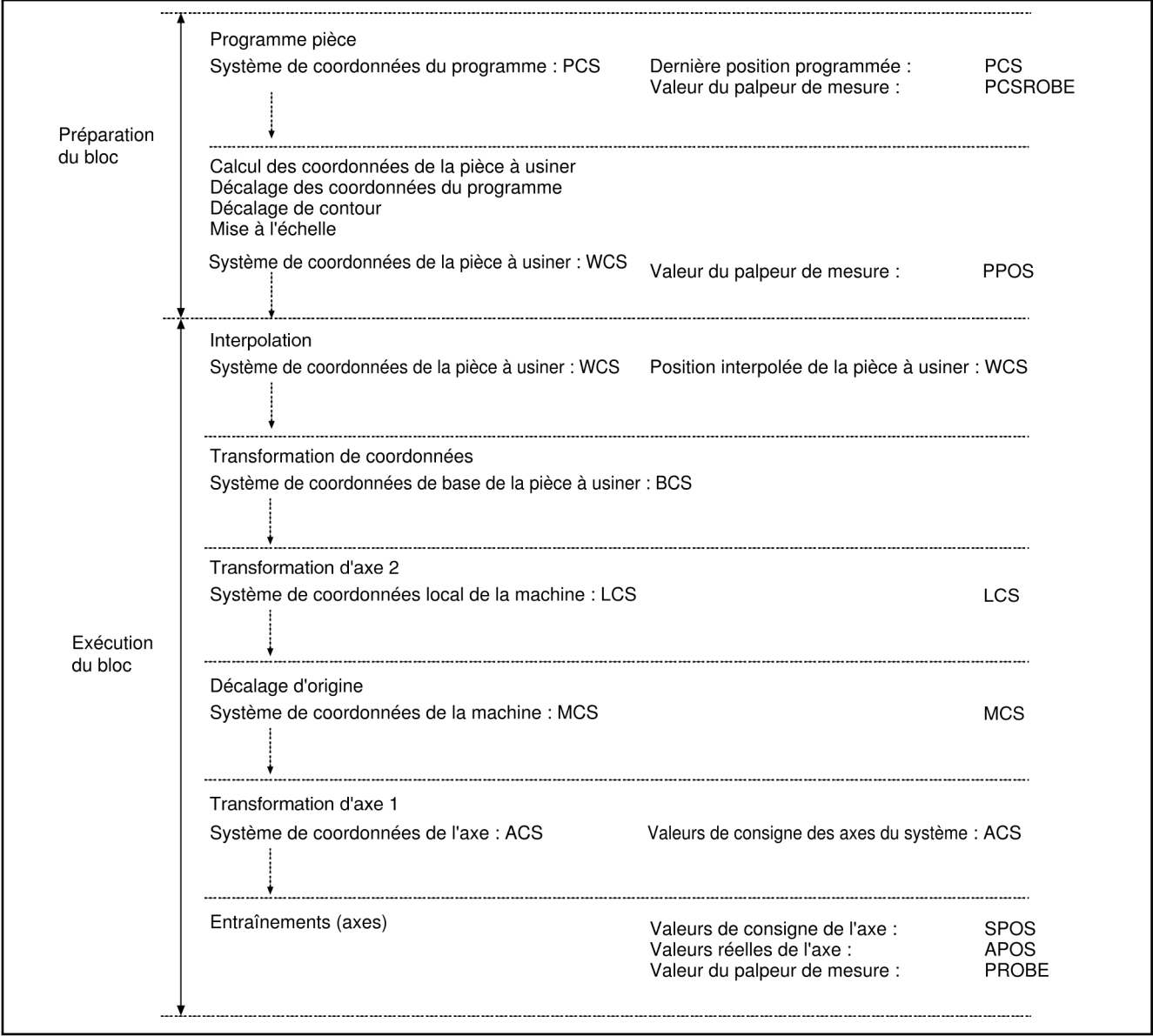


Fig. 7-1: Diagramme de flux de données

Unités de mesure des positions d'axes et de coordonnées fournies

Axes linéaires synchrones et

coordonnées tridimensionnelles translationnelles :

"mm" ou "Inch" ;
en fonction du réglage actuel (G71, G70)
sur le canal appelant.

Axes rotatifs synchrones et

coordonnées tridimensionnelles rotatives :

"Degré"

Axes linéaires asynchrones :

"Millimètres"

Axes ronds asynchrones :

"Degré"

Exemple : Configuration et attribution des noms d'axes aux canaux

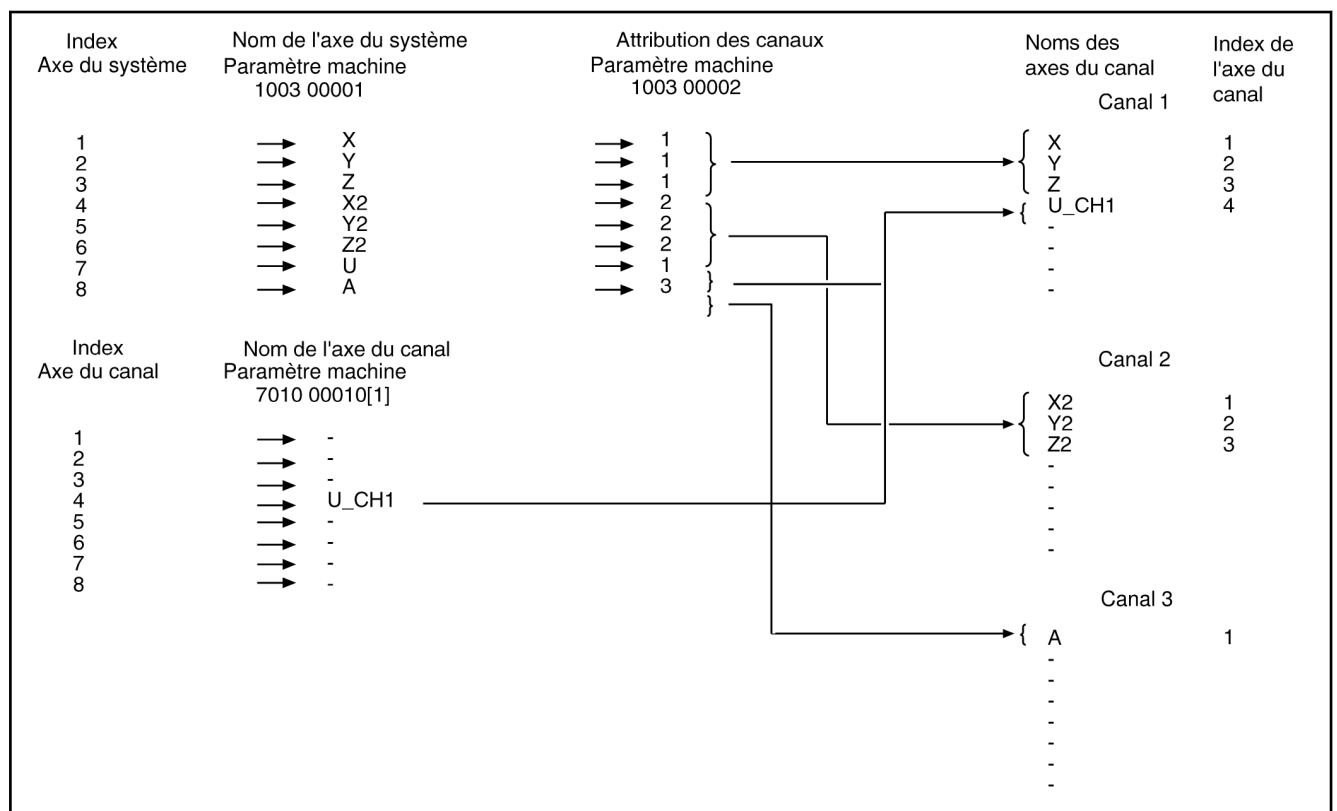


Fig. 7-2: Configuration et attribution des noms d'axes aux canaux

7.1.2 Ordres de lecture des noms et propriétés

AXADR\$

CHAR	READ	WRITE	FUNCT	PROC	WAIT	ERRNO	!

Description : **AXis ADdRess** – Adresse d'axes et de coordonnées

Les valeurs renvoyées par l'ordre CPL **AXADR\$** sont les noms des coordonnées de programme et de machine, ainsi que les noms d'axes du canal et du système.

Ordres CPL

Les paramètres entrés ici sont les suivants :

- Index d'axe du système
- Index SACI
- Index de coordonnée du programme (index de coordonnée du canal) pour les noms de coordonnées du programme
- Index de coordonnée de machine pour les noms de coordonnées de machine
- Index d'axe du canal pour les noms d'axes du canal et du système

L'ordre AXADR\$ prend en charge le paramètre ERRNO.

Syntaxe :

AXADR\$(<index> [, <mode de sélection>] [, <numéro du canal>])	
<index>	Index de coordonnée, d'axe du canal, d'axe du système ou SACI. L'interprétation s'effectue selon le <mode de sélection> indiqué.
<mode de sélection>	Optionnel : Définit la manière dont l'index transféré est interprété : 0 : Index d'axe du système L'ordre fournit le nom d'axe du système ou le nom SACI correspondant. 1 : Index de coordonnée du canal (par défaut) L'ordre fournit le nom de coordonnée du programme correspondant. 2 : Index de coordonnée de machine L'ordre fournit le nom de coordonnée de machine correspondant. 3 : Index de l'axe du canal L'ordre fournit le nom d'axe du canal correspondant. 4 : Index de l'axe du canal L'ordre fournit le nom d'axe du système de l'axe du système lié.

Ordres CPL

<numéro du canal>	Optionnel : Numéro du canal auquel se réfère l'index de coordonnée du canal ou d'axe du canal indiqué. -1 : L'ordre fournit le nom en vigueur dans le bloc d'interruption (uniquement en combinaison avec l'avance de bloc et les sous-programmes asynchrones). 0 : Canal actuel (par défaut). Le numéro du canal n'est permis que si <mode de sélection> = 1, 2, 3, 4. Combiné à <mode de sélection> = 0, le numéro du canal peut uniquement être indiqué si une valeur inférieure à 0 est indiquée pour <index>, c.-à-d. s'il faut accéder aux noms des coordonnées de machine et des coordonnées de la pièce à usiner configurés sur le canal. Si le numéro de canal indiqué est celui d'un canal tiers, alors l'ordre AXADR\$ fournit les noms en vigueur au moment actif du canal indiqué.
ERRNO	Variable CPL Si la variable CPL ERRNO est programmée à un emplacement quelconque au sein de la liste de paramètres, l'ordre ne génère aucune erreur d'exécution interne. L'erreur est renvoyée par une valeur de retour correspondante de la variable. <i>Les valeurs de retour suivantes sont possibles :</i> • 0 : Accès OK • < 0 : Erreur : pour une description de l'erreur, voir chap. 3.14.5 "Variable ERRNO pour l'analyse des erreurs des fonctions CPL ERRNO" à la page 85 Si la variable ERRNO n'est pas indiquée, une erreur d'exécution interne est automatiquement générée en cas d'erreur d'accès interne.

Tab. 7-1: Syntaxe AXADR\$

Exemples de syntaxe :

AXADR\$(2,0)	Fournit le nom d'axe du système du 2e axe du système.
AXADR\$(3)	Fournit le nom de coordonnée du programme de la 3e coordonnée du programme sur le canal actuel.
AXADR\$(1,2,2)	Fournit le nom de la 1re coordonnée de machine sur le canal 2.
AXADR\$(4,3,4)	Fournit le nom d'axe du canal du 4e axe du canal 4.
AXADR\$(5,4)	Fournit le nom d'axe du système du 5e axe du canal actuel.

AXINF

INT	READ	WRITE	FUNCT	PROC	WAIT	ERRNO	!

Description

AXis INFormation – Informations sur l'axe

Les valeurs renvoyées par l'ordre CPL **AXINF** sont différentes propriétés spécifiques aux axes et aux coordonnées.

Les paramètres de saisie suivants sont possibles :

- nom d'axe du système,
- numéro d'axe du système,

Ordres CPL

- nom de l'axe du canal et (en option) index du canal,
- index de l'axe du canal et (en option) index du canal.

L'ordre AXINF permet de déterminer les informations suivantes spécifiques aux axes et aux coordonnées :

- Index d'axe du système pour un axe du canal ou du système.
- Type de mouvement d'un axe (linéaire, rond, sans fin, modulo linéaire).
- Axe virtuel.
- Signification des coordonnées et des axes (signification de X, signification de Y, signification de Z).
- Programmation du diamètre.

L'ordre AXINF prend en charge le paramètre ERRNO.

Syntaxe :

AXINF(<axe> [, <type de coordonnée> [, <numéro du canal> [, <sélection>]]])	
<axe>	Numéro de l'axe du canal ou nom de l'axe : Le nom peut être soit le nom de l'axe du système, soit le nom de l'axe du canal. L'interprétation s'effectue selon le <mode de sélection> indiqué.
<type de coordonnée>	Optionnel : Définit si un nom programmé est interprété en tant que nom d'axe du système ou en tant que nom d'axe du canal : 0 : Système 1 : Canal (par défaut)
<numéro du canal>	Optionnel : Numéro du canal auquel se réfère l'index d'axe du canal indiqué ou le nom d'axe du canal indiqué. 0 : Canal actuel (par défaut). Le numéro du canal n'est permis que si <mode de sélection> = 1. Si le <mode de sélection> est défini sur 1 et qu'aucun numéro de canal n'est programmé, les données déterminées sont celles du canal actuel.

Ordres CPL

<sélection>	Optionnel : Sélection de l'information spécifique à l'axe qui doit être fournie par AXINF : 0 : Index de coordonnée ou d'axe (par défaut) 1 : Type de mouvement de l'axe <i>Valeurs de retour :</i> • 1 : Axe linéaire • 2 : Axe sans fin • 3 : Axe rond • 4 : Axe modulo linéaire 2 : Axe virtuel <i>Valeurs de retour :</i> • 0 : Axe réel • 1 : Axe virtuel 11 : Signification de l'axe <i>Valeurs de retour :</i> • 0 : Aucune signification de l'axe • 1 : Signification de X • 2 : Signification de Y • 3 : Signification de Z 12 : Programmation du diamètre <i>Valeurs de retour :</i> • 0 : Aucune programmation du diamètre • 1 : Programmation du diamètre
ERRNO	Variable CPL Si la variable CPL ERRNO est programmée à un emplacement quelconque au sein de la liste de paramètres, l'ordre ne génère aucune erreur d'exécution interne. L'erreur est renvoyée par une valeur de retour correspondante de la variable. <i>Les valeurs de retour suivantes sont possibles :</i> • 0 : Accès OK • < 0 : Erreur : pour une description de l'erreur, voir chap. 3.14.5 "Variable ERRNO pour l'analyse des erreurs des fonctions CPL ERRNO" à la page 85 Si la variable ERRNO n'est pas indiquée, une erreur d'exécution interne est automatiquement générée en cas d'erreur d'accès interne.

Tab. 7-2:

Syntaxe AXINF

Exemples de syntaxe :

AXINF("VA",0)	Fournit l'index d'axe du système de l'axe du système "VA".
AXINF("Z")	Fournit l'index d'axe du système de l'axe du canal "Z" sur le canal actuel.

Ordres CPL

AXINF("Y",1,5)	Fournit l'index d'axe du système de l'axe du canal "Z" sur le canal 5.
AXINF(4,1,0,ERRNO)	Fournit l'index d'axe du système du 4e axe du canal sur le canal actuel.
	Le code erreur est renvoyé dans la variable ERRNO.

AXP

---	READ	WRITE	FUNCT	PROC	WAIT	ERRNO	!

Description :

AXis Programming – Adresse d'axe variable
 L'ordre **AXP** permet d'écrire des programmes pièces et des programmes de mesure indépendamment du plan.

Syntaxe :

AXP (<numéro d'axe>,<information sur la course>[,<TypeAxe>])	
Cet ordre est utilisé dans un bloc CN. Il doit être inscrit entre crochets "[]" et est programmé à la place des valeurs d'adresse.	
<numéro d'axe>	Index de coordonnée de la pièce à usiner ou axe du système
<information sur la course>	Variable ou valeur de l'information sur la course
<type d'axe>	Optionnel : Définit la manière dont un index programmé sous <numéro d'axe> est interprété : 0 : Index d'axe du système 1 : Index de coordonnée de la pièce à usiner du canal sur lequel le programme est actuellement traité. En l'absence de programmation explicite, la définition utilisée est <type d'axe> = 1.

Tab. 7-3:
Syntaxe AXP

!



L'ordre AXP peut uniquement être programmé au sein de blocs CN s'il est inséré entre crochets []. Un appel est donc impossible avec le mode de fonctionnement Saisie manuelle !

Exemple :
Sous-programme :

10 A%=P1% : B%=P2%	Transférer le numéro d'axe de P1% et P2% à A% et B%
20 C=P3 :D=P4 :RA=P5	Transférer les valeurs de consigne pour G2
30 E=0	Constante pour pôle à G20

```
N40 G20 [AXP(A%,E)] [AXP(B%,E)]    Changement de plan avec G20 ;
                                     pôle sur 0,0
N50 G2 [AXP(A%,C)] [AXP(B%,D)]      Programmation du rayon avec G2
R[RA]] F1000
```

Définition du plan via A% et B%. Ensuite, changement de plan via G20. À la fin, les axes réalisent un arc de cercle avec F1000 déterminé par les variables C et D (point d'arrivée) et RA (rayon).

7.1.3 Ordres de lecture des positions de coordonnées et d'axes

Généralités

Les points suivants doivent être pris en compte ici :

- L'index de coordonnée sur le canal est toujours fixe pour les coordonnées tridimensionnelles et de machine.
- En revanche, des coordonnées peuvent être intégrées à ou retirées d'un canal grâce aux fonctions CN du transfert d'axe. Cela permet de modifier l'index de coordonnée des autres coordonnées sur le canal.
Le fait de définir l'index d'axe du système permet de travailler avec un indice fixe.
- Une erreur d'exécution est générée en cas de définition d'un axe du système non configuré.

Les ordres décrits fonctionnent tous avec des paramètres similaires qui sont donc expliqués préalablement ci-dessous :

<coordonnée>	Index ou nom d'une coordonnée : Un nom est interprété en tant que nom de coordonnée. Il est interprété en tant que nom de canal ou d'axe du système uniquement s'il n'existe aucun nom de coordonnée correspondant. Un index est interprété selon le <mode de sélection> indiqué. La programmation d'une coordonnée/d'un axe non configuré(e) entraîne une erreur d'exécution si la variable optionnelle ERRNO n'est pas programmée.
<axe>	Index ou nom d'un axe : Un nom est interprété en tant que nom d'axe du canal. S'il n'en existe aucun, c'est le nom d'axe du système qui est utilisé. Un index est interprété selon le <mode de sélection> indiqué. La programmation d'un axe non configuré entraîne une erreur d'exécution si la variable optionnelle ERRNO n'est pas programmée.
<mode de sélection>	Optionnel : Définit la manière dont est interprété un index programmé sous <coordonnée> ou <axe> : 0 : Index d'axe du système 1 : Index de coordonnée ou index d'axe du canal (par défaut) En l'absence de programmation explicite, la définition utilisée est <mode de sélection> = 1.

Ordres CPL

<canal>

Optionnel :

Numéro du canal, uniquement autorisé avec <mode de sélection> = 1.

Si les axes/coordonnées de canaux tiers doivent être lus et adressés à l'aide de leur index ou de leur nom, le numéro du canal auquel l'axe/la coordonnée est actuellement attribué(e) est indiqué dans <canal>.

Si aucun canal n'est indiqué, les coordonnées/axes utilisé(e)s sont ceux du canal actuel. Si un axe du système est adressé (avec son nom ou son index) et qu'un canal est également indiqué, alors un message d'erreur est généré, même si la variable ERRNO est programmée.

ERRNO

Variable CPL programmable à n'importe quel emplacement.

Avec ERRNO, aucune erreur d'exécution n'est générée en cas d'erreur ; les valeurs de retour possibles sont :

- 0 : Accès OK
- -1 : Erreur de paramètre
- -2 : La coordonnée/L'axe n'existe pas.
- -3 : Coordonnée/Axe inadmissible sur le canal
- -4 : L'axe n'est pas une pseudo-coordonnée.
- -5 : Le canal n'existe pas.
- -6 : L'ordre peut uniquement être appelé sur le canal concerné.
- -7 : Échec de la lecture des données

ACS

DOUBLE	READ	WRITE	FUNCT	PROC	WAIT	ERRNO	!

Description : **Axis Coordinate System** – Lecture des valeurs de consigne de l'axe

Fournit la position de consigne actuelle d'un axe.

Les axes peuvent être adressés via leur nom du canal ou leur index du canal.

En alternative, un axe peut être adressé via son nom du système ou son index du système.

Les points suivants s'appliquent ici :

- Si le couplage d'axes système est inactif pour cet axe, alors le résultat d'ACS correspond à l'ordre "SPOS".
- Si la valeur fournie doit être déterminée au moment de l'exécution du bloc, alors un ordre "WAIT" doit être programmé au préalable dans un bloc à part (voir aussi à ce sujet [chap. 3.19.2 "Fonctions de synchronisation de la préparation des blocs" à la page 96](#)).

Sans ordre WAIT, aucune valeur clairement prévisible ne sera fournie, car le "retard" exact de l'exécution du bloc par rapport à son traitement est inconnu.

- En cas d'accès à des valeurs d'axe d'un canal tiers, il convient le cas échéant de prendre des mesures de synchronisation afin de mesurer une position définie.

Syntaxe :

ACS (<Axe> [,<Mode de sélection>[,<Canal>]])	
	(Pour les paramètres, voir chap. 7.1.3 "Ordres de lecture des positions de coordonnées et d'axes" à la page 457 ; pour la programmation d'ERRNO, voir chap. 3.14.5 "Variable ERRNO pour l'analyse des erreurs des fonctions CPL ERRNO" à la page 85)

Tab. 7-4: Syntaxe ACS

Exemples de syntaxe :

ACS ("X", ERRNO)

ACS ("X", 1)

ACS ("X", 1, 1)

Canal 2 selon l'exemple de configuration (voir Fig. 7-2 "Configuration et attribution des noms d'axes aux canaux" à la page 451) :

```

N10 G0 G90 X2=150 Y2=100

20 WAIT

30 X2WERT=ACS ("X2", ERRNO)
31 IF ERRNO <> 0 THEN
32 PRN#(0, "Erreur :", ERRNO)
33 ENDIF

N40 G91 X2=10 Y2=10

50 WAIT

60 X2WERT=ACS (1, 1)

70 Y2WERT=ACS ("Y2", 1, 2)

80 XWERT=ACS (1, 0)
  
```

La position de l'axe actuellement interpolée de l'axe du canal X2 est attribuée à X2WERT.

La position de l'axe actuellement interpolée du 1er axe du canal actuel (axe X2) est attribuée à X2WERT.

La position de l'axe actuellement interpolée de l'axe du canal Y2 du 2e canal est attribuée à Y2WERT.

La position de l'axe actuellement interpolée du 1er axe du système est attribuée à XWERT.

APOS

DOUBLE	READ	WRITE	FUNCT	PROC	WAIT	ERRNO	!

Description : **Actual POSition** – Lecture des valeurs réelles de l'axe

Fournit la **position réelle** actuelle d'un axe par rapport au système de coordonnées de l'axe.

Tous les axes du système peuvent être adressés via leur nom du système ou leur index du système.

Les points suivants s'appliquent ici :

- Si la valeur fournie doit être déterminée au moment de l'exécution du bloc, alors un ordre WAIT doit être programmé au préalable dans un bloc à part (voir aussi à ce sujet chap. "WAIT (sans paramètres)" à la page 96).

Sans ordre WAIT, aucune valeur clairement prévisible ne sera fournie, car le "retard" exact de l'exécution du bloc par rapport à son traitement est inconnu.

Ordres CPL

- En cas d'accès à des valeurs d'axe d'un canal tiers, il convient le cas échéant de prendre des mesures de synchronisation afin de mesurer une position définie.

Syntaxe :

APOS (<axe>)	
<axe>	Index ou nom d'un axe du système La programmation d'un axe non configuré entraîne une erreur d'exécution.

Tab. 7-5: Syntaxe APOS

ERRNO n'est pas programmable.

Exemple : Canaux selon l'exemple de configuration (voir [Fig. 7-2 "Configuration et attribution des noms d'axes aux canaux" à la page 451](#)) :

30 AKT4=APOS (4)	La valeur réelle actuelle du 4e axe du système (axe X2 sur le canal 2) est attribuée à la variable AKT4.
:	
50 AKT8=APOS ("A")	La valeur réelle du 8e axe du système (axe A sur le canal 3) est attribuée à la variable AKT8.
:	

MCS

DOUBLE	READ	WRITE	FUNCT	PROC	WAIT	ERRNO	!

Description : **Machine Coordinate System** – Lecture des coordonnées de machine

Fournit la position actuelle de la machine (MCS) pour une coordonnée de machine sans compensation des erreurs de rectitude et d'angle.

Il est également possible d'accéder à des coordonnées hors canal.

Les points suivants s'appliquent ici :

- Si la valeur fournie doit être déterminée au moment de l'exécution du bloc, alors un ordre WAIT doit être programmé au préalable dans un bloc à part (voir aussi à ce sujet [chap. "WAIT \(sans paramètres\)" à la page 96](#)).
Sans ordre WAIT, aucune valeur clairement prévisible ne sera fournie, car le "retard" exact de l'exécution du bloc par rapport à son traitement est inconnu.
- En cas d'accès à des valeurs de coordonnées d'un canal tiers, il convient le cas échéant de prendre des mesures de synchronisation afin de mesurer une position définie.

Syntaxe :

MCS (<coordonnée> [,<mode de sélection>[,<canal>]])	
	(Pour les paramètres, voir chap. 7.1.3 "Ordres de lecture des positions de coordonnées et d'axes" à la page 457 ; pour la programmation d'ERRNO, voir chap. 3.14.5 "Variable ERRNO pour l'analyse des erreurs des fonctions CPL ERRNO" à la page 85)

Tab. 7-6: Syntaxe MCS

Exemples de syntaxe :
MCS ("X", ERRNO)
MCS ("X", 1)
MCS ("X", 1, 1)

Canal 2 selon l'exemple de configuration (voir Fig. 7-2 "Configuration et attribution des noms d'axes aux canaux" à la page 451) :

N10 G0 G90 X2=150 Y2=100	
20 WAIT	
30 X2WERT=MCS ("X2", ERRNO)	La position de la machine actuellement interpolée de la coordonnée X2 est attribuée à X2WERT.
31 IF ERRNO <> 0 THEN	
32 PRN#(0, "Erreur :", ERRNO)	
33 ENDIF	
N40 G91 X2=10 Y2=10	
50 WAIT	
60 X2WERT=MCS (1, 1)	La position de la machine actuellement interpolée de la 1re coordonnée de machine du canal actuel (coordonnée X2) est attribuée à X2WERT.
70 Y2WERT=MCS ("Y2", 1, 2)	La position actuellement interpolée de la coordonnée Y2 du 2e canal est attribuée à Y2WERT.
80 XWERT=MCS (1, 0)	La position de l'axe actuellement interpolée du 1er axe du système est attribuée à XWERT. Cet accès est autorisé uniquement si l'axe du système est identique à la coordonnée de machine.

PCS

DOUBLE	READ	WRITE	FUNCT	PROC	WAIT	ERRNO	!

Description : **Program Coordinate System** – Lecture de la valeur de coordonnée programmée

Fournit la dernière valeur programmée pour une coordonnée.

Les points suivants s'appliquent ici :

- Si le système de coordonnées de la pièce à usiner est commuté entre la programmation de la coordonnée et l'interrogation de la position, alors la valeur fournie tient déjà compte du nouveau système de coordonnées de la pièce à usiner.
- Seules les coordonnées du canal concerné peuvent être interrogées.

Syntaxe :

PCS (<coordonnée> [, <mode de sélection>])	
	(Pour les paramètres, voir chap. 7.1.3 "Ordres de lecture des positions de coordonnées et d'axes" à la page 457 ; pour la programmation d'ERRNO, voir chap. 3.14.5 "Variable ERRNO pour l'analyse des erreurs des fonctions CPL ERRNO" à la page 85)

Tab. 7-7: Syntaxe PCS

Exemples de syntaxe :
 PCS ("X", ERRNO)
 PCS ("X", 1)

Ordres CPL

Canal 2 selon l'exemple de configuration (voir [Fig. 7-2 "Configuration et attribution des noms d'axes aux canaux" à la page 451](#)) :

N1 G0 G90 X2=150 Y2=100	
02 X2WERT=PCS (1,ERRNO)	La position absolue programmée pour la 1re coordonnée sur le canal actuel est attribuée à X2WERT (X2WERT = 150).
03 IF ERRNO <>0 THEN	
04 PRN# (0,"Erreur :",ERRNO)	
05 ENDIF	
N10 G91 X2=10	
011 X2WERT=PCS (1,1)	La position absolue programmée pour la 1re coordonnée sur le canal actuel est attribuée à X2WERT (X2WERT = 160).
N12 X2=5 Y2=10	
13 Y2WERT=PCS ("Y2",1)	La position absolue programmée pour l'axe du système Y2 est attribuée à Y2WERT (Y2WERT = 110).
14 X2WERT=PCS ("X2")	La position absolue programmée pour la coordonnée X2 sur le canal actuel est attribuée à X2WERT (X2WERT = 165).
15 XWERT=PCS (1,0)	Erreur d'exécution : L'accès au 1er axe du système sur le canal 2 n'est pas autorisé. (L'axe est attribué au canal 1.)

SPOS

DOUBLE	READ	WRITE	FUNCT	PROC	WAIT	ERRNO	!

Description : **System axis POSition** – Lecture de la position de consigne d'un axe du système

Fournit la **position de consigne** actuelle d'un axe par rapport au système de coordonnées de l'axe.

Un couplage d'axes système actif est pris en compte dans la position de consigne, c.-à-d. que la position correspond à la valeur de l'axe esclave calculée à partir de la somme des parts actives de l'axe maître et de la position de consigne dans le système de coordonnées de l'axe.

Tous les axes du système peuvent être adressés via leur nom du système ou leur index du système.

Les points suivants s'appliquent ici :

- Si le couplage d'axes du système est inactif, SPOS et ACS fournissent des valeurs identiques.
- Si le couplage d'axes du système est actif, SPOS fournit la valeur de l'axe esclave SysAxCoupleSta[i].ActPos dans l'affichage \$PositionDe-Consigne.
- Si la valeur fournie doit être déterminée au moment de l'exécution du bloc, alors un ordre WAIT doit être programmé au préalable dans un bloc à part (voir aussi à ce sujet [chap. 3.19.2 "Fonctions de synchronisation de la préparation des blocs" à la page 96](#)).

Sans WAIT, aucune valeur clairement prévisible ne sera fournie, car le "retard" exact de l'exécution du bloc par rapport à son traitement est inconnu.

- En cas d'accès à des valeurs d'axe d'un canal tiers, il convient le cas échéant de prendre des mesures de synchronisation afin de mesurer une position définie.

Syntaxe :

SPOS (<axe>)	
<axe>	Index ou nom d'un axe du système La programmation d'un axe non configuré entraîne une erreur d'exécution.

Tab. 7-8: Syntaxe SPOS

ERRNO n'est pas programmable.

Exemple :

Canaux selon l'exemple de configuration (voir Fig. 7-2 "Configuration et attribution des noms d'axes aux canaux" à la page 451).

30 POS1=SPOS(1)	La valeur de consigne de l'axe actuelle du 1er axe du système (axe X sur le canal 1) est attribuée à la variable POS1.
:	
50 POS5=SPOS("Y2")	La valeur de consigne de l'axe actuelle du 5e axe du système (axe Y2 sur le canal 2) est attribuée à la variable POS5.
:	

WCS

DOUBLE	READ	WRITE	FUNCT	PROC	WAIT	ERRNO	!

Description :

Workpiece Coordinate System – Lecture de la position de la pièce à usiner
Fournit la position actuelle de la pièce à usiner sans valeurs de correction en ligne pour une coordonnée.

Les points suivants s'appliquent ici :

- Si la valeur fournie doit être déterminée au moment de l'exécution du bloc, alors un ordre "WAIT" doit être programmé au préalable dans un bloc à part (voir aussi à ce sujet chap. 3.19.2 "Fonctions de synchronisation de la préparation des blocs" à la page 96).

Sans ordre WAIT, aucune valeur clairement prévisible ne sera fournie, car le "retard" exact de l'exécution du bloc par rapport à son traitement est inconnu.

- En cas d'accès à des valeurs de coordonnées d'un canal tiers, il convient le cas échéant de prendre des mesures de synchronisation afin de mesurer une position définie.

Syntaxe :

WCS (<coordonnée> [,<mode de sélection>[,<canal>]])	
	(Pour les paramètres, voir chap. 7.1.3 "Ordres de lecture des positions de coordonnées et d'axes" à la page 457 ; pour la programmation d'ERRNO, voir chap. 3.14.5 "Variable ERRNO pour l'analyse des erreurs des fonctions CPL ERRNO" à la page 85)

Tab. 7-9: Syntaxe WCS

Exemples de syntaxe :

WCS ("X", ERRNO)

Ordres CPL

```
WCS ("X", 1)
```

```
WCS ("X", 1, 1)
```

Canal 2 selon l'exemple de configuration (voir [Fig. 7-2 "Configuration et attribution des noms d'axes aux canaux" à la page 451](#)) :

```
10 WAIT
```

```
20 Z2POS=WCS (3,1,2,ERRNO)
```

La position de la pièce à usiner actuellement interpolée de la 3e coordonnée du 2e canal est attribuée à Z2POS (axe Z2).

```
21 IF ERRNO <>0 THEN
```

```
22 PRN# (0, "Erreur :", ERRNO)
```

```
23 ENDIF
```

```
30 WAIT
```

```
110 YPOS = WCS ("Y")
```

La position de la pièce à usiner actuellement interpolée de la coordonnée Y du canal actuel est attribuée à YPOS.

```
120 XPOS = WCS (1)
```

La position de la pièce à usiner actuellement interpolée de la 1re coordonnée du canal actuel est attribuée à XPOS.

7.1.4 Ordres de lecture des valeurs mesurées

PCSPROBE

DOUBLE	READ	WRITE	FUNCT	PROC	WAIT	ERRNO	!

Description : **Program Coordinate System PROBE** – Valeur mesurée dans les coordonnées du programme

PCSPROBE() est utilisé après les fonctions CN suivantes :

- G75.1 et G75 (déplacement contre le 1er palpeur de mesure)
- G75.2 (déplacement contre le 2e palpeur de mesure)
- FME (mesure au vol)
- FSP (mesure sur butée fixe)

PCSPROBE() génère une erreur d'exécution par exemple dans les cas suivants :

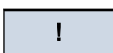
- Interrogation d'un axe asynchrone
- Interrogation d'un axe synchrone d'un canal tiers
- Erreur de syntaxe...

ERRNO est programmable (voir à ce sujet [chap. 3.14.5 "Variable ERRNO pour l'analyse des erreurs des fonctions CPL ERRNO" à la page 85](#)).

Syntaxe :

PCSPROBE (<coordonnée> [,<mode de sélection>])	
<coordonnée>	Voir chap. 7.1.3 "Ordres de lecture des positions de coordonnées et d'axes" à la page 457
<mode de sélection>	

Tab. 7-10: Syntaxe PCSPROBE



Il n'est pas nécessaire d'exécuter une synchronisation supplémentaire au moyen d'un ordre WAIT avant PCSPROBE si les fonctions CN Palpeur de mesure (G75.x) et Mesure sur butée fixe (FSP) sont programmées avant. Si

la mesure a réussi et qu'elle a fourni un résultat, une requête est lancée à l'aide de l'ordre CPL SD(9) !

Exemple : Canal 1 selon l'exemple de configuration (voir Fig. 7-2 "Configuration et attribution des noms d'axes aux canaux" à la page 451) :

```
N10 G75.1 X100 Y100 Z50
20 IF SD(9)=1 THEN
N30 (MSG, Palpeur de mesure 1 non déclenché !)
40 GOTO .ERREUR
50 ELSE
60 ZMESS=PCSPROBE("Z",ERR-    La valeur de la coordonnée Z de la position
NO) ;                          mesurée est attribuée à la variable "ZMESS".
61 IF ERRNO <>0 THEN
62 GOTO .Erreur
63 ENDIF
70 ENDIF
```

Particularités : PCSPROBE() vérifie la présence de toutes les valeurs mesurées des axes nécessaires au calcul de la valeur mesurée en tenant compte des fonctions CN suivantes :

- Transformation d'axe (CRD)
- Transformations de coordonnée (BCR, G152 à G159)
- Rotation (ROT)

Si cela a été programmé ainsi, une valeur mesurée de l'axe est indiquée via ERRNO 45 ou 46.

PPOS



À partir de la version MTX 14VRS, la position de mesure peut toujours être déterminée avec PCSPROBE().

Dans les nouvelles applications, il convient d'utiliser exclusivement les ordres **PCSPROBE()** et **PROBE()**.

DOUBLE	READ	WRITE	FUNCT	PROC	WAIT	ERRNO	!

Description : **Probe Position**

PPOS() est utilisé après les fonctions CN suivantes :

- G75.1 et G75 (déplacement contre le 1er palpeur de mesure)
- G75.2 (déplacement contre le 2e palpeur de mesure)
- FME (mesure au vol)
- FSP (mesure sur butée fixe)

PPOS() génère une erreur d'exécution par exemple dans les cas suivants :

- Interrogation d'un axe asynchrone
- Interrogation d'un axe synchrone d'un canal tiers
- Erreur de syntaxe...

ERRNO n'est pas programmable.

Ordres CPL

Syntaxe :

PPOS (<axe> [,<mode de sélection>])	
<axe>	Voir chap. 7.1.3 "Ordres de lecture des positions de coordonnées et d'axes" à la page 457
<mode de sélection>	

Tab. 7-11: Syntaxe PPOS

!

Il n'est pas nécessaire d'exécuter une synchronisation supplémentaire au moyen d'un ordre WAIT avant PPOS si les fonctions CN Palpeur de mesure (G75.x) et Mesure sur butée fixe (FSP) sont programmées avant. Si la mesure a réussi et qu'elle a fourni un résultat, une requête est lancée à l'aide de l'ordre CPL SD(9) !

Exemple : Canal 3 selon l'exemple de configuration (voir [Fig. 7-2 "Configuration et attribution des noms d'axes aux canaux"](#) à la page 451) :

```

N10 G75.2 A250 F500

20 IF SD(9) = 1 THEN

N30 (MSG, Palpeur de mesure 2 non déclenché !)

40 GOTO .ERREUR

50 ELSE

60 AMESS = PPOS(1,1) ;           La valeur mesurée du 1er axe du canal est
                                attribuée à la variable "AMESS".

70 ENDIF

```

Particularités : PPOS tient compte des corrections suivantes :

- Décalages d'origine des axes (G54 à G59)
- Corrections d'outil (G48, ED)
- Décalage des coordonnées du programme (Trans, ATrans)
- Compensations d'erreur de pas de la broche et de croix

Les fonctions suivantes ne sont pas prises en compte :

- Transformation d'axe (Coord)
- Transformations de coordonnée (BcsCorr, G152 à G159)
- Mise à l'échelle (PoleSet, Mirror, Scale, Rotate)

Sans transformation d'axe ou de coordonnée, la valeur fournie se rapporte au dernier système de coordonnées de la pièce à usiner (WCS) programmé.

PROBE

DOUBLE	READ	WRITE	FUNCT	PROC	WAIT	ERRNO	!

Description : PROBE – Valeur mesurée de l'entraînement

PROBE() est utilisé après les fonctions CN suivantes :

- G75.1 et G75 (déplacement contre le 1er palpeur de mesure)
- G75.2 (déplacement contre le 2e palpeur de mesure)
- FME (mesure au vol)
- FSP (mesure sur butée fixe)

PROBE() génère une erreur d'exécution par exemple dans les cas suivants :

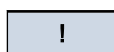
- Interrogation d'un axe asynchrone
- Interrogation d'un axe synchrone d'un canal tiers
- Erreur de syntaxe...

ERRNO n'est pas programmable.

Syntaxe :

PROBE (<axe> [, <mode de sélection>])	
<axe>	Voir chap. 7.1.3 "Ordres de lecture des positions de coordonnées et d'axes" à la page 457
<mode de sélection>	

Tab. 7-12: Syntaxe PROBE



Il n'est pas nécessaire d'exécuter une synchronisation supplémentaire au moyen d'un ordre WAIT avant PROBE si les fonctions CN Palpeur de mesure (G75.x) et Mesure sur butée fixe (FSP) sont programmées avant. Si la mesure a réussi et qu'elle a fourni un résultat, une requête est lancée à l'aide de l'ordre CPL SD(9) !

Exemple :

Canal 2 selon l'exemple de configuration (voir Fig. 7-2 "Configuration et attribution des noms d'axes aux canaux" à la page 451) :

```
N70 G75.2 Y2 250
80 IF SD(9) = 1 THEN
N90 (MSG, Palpeur de mesure 2 non déclenché !)
100 GOTO .ERREUR
110 ELSE
120 Y2MESS=PROBE(2) ;
130 ENDIF
```

La valeur mesurée du 2e axe du canal (ici : axe Y2 du canal 2) est attribuée à la variable "Y2MESS".

Particularités :

L'ordre PROBE() :

- fournit des valeurs mesurées qui se rapportent au système de coordonnées d'axe (ACS) ;
- tient toujours compte des compensations de croix et d'erreur de pas de la broche.

7.2 Broches

7.2.1 Vue d'ensemble

Le langage CPL met à disposition des ordres permettant d'accéder aux données des broches.

7.2.2 Ordres de lecture des noms

SPADR\$

CHAR	READ	WRITE	FUNCT	PROC	WAIT	ERRNO	!

Description :

SPindles ADdRess – Adresse de broche

Ordres CPL

La valeur renvoyée par l'ordre CPL **SPADR\$** contient les noms des broches du canal et du système.

Les paramètres entrés ici sont les suivants :

- Index de broche du système
- Index de broche du canal pour les noms de broches du canal et du système

L'ordre SPADR\$ prend en charge le paramètre ERRNO.

Syntaxe :

SPADR\$(<index> [, [<mode de sélection>] [, <numéro du canal>]])	
<index>	Index de la broche du canal ou de la broche du système L'interprétation s'effectue selon le <mode de sélection> indiqué.
<mode de sélection>	Optionnel : Définit la manière dont l'index transféré est interprété : 0 : Index de broche du système L'ordre fournit le nom de la broche du système correspondant. 1 : Index de la broche du canal (par défaut) L'ordre fournit le nom de la broche du canal correspondant. 2 : Index de la broche du canal L'ordre fournit le nom de la broche du système de la broche du système liée.
<numéro du canal>	Optionnel : Numéro du canal auquel se rapporte l'index de la broche du canal indiquée. -1 : L'ordre fournit le nom en vigueur dans le bloc d'interruption (uniquement en combinaison avec l'avance de bloc et les sous-programmes asynchrones). 0 : Canal actuel (par défaut) Le numéro du canal est uniquement autorisé si <mode de sélection> = 1 ou 2. Si le numéro du canal indiqué est celui d'un canal tiers, alors l'ordre SPADR\$ fournit les noms en vigueur au moment actif du canal indiqué.
ERRNO	Variable CPL Si la variable CPL ERRNO est programmée à un emplacement quelconque au sein de la liste de paramètres, l'ordre ne génère aucune erreur d'exécution interne. L'erreur est renvoyée par une valeur de retour correspondante de la variable. <i>Les valeurs de retour suivantes sont possibles :</i> • 0 : Accès OK • < 0 : Erreur : pour une description de l'erreur, voir chap. 3.14.5 "Variable ERRNO pour l'analyse des erreurs des fonctions CPL ERRNO" à la page 85 Si la variable ERRNO n'est pas indiquée, une erreur d'exécution interne est automatiquement générée en cas d'erreur d'accès interne.

Tab. 7-13: Syntaxe SPADR\$

Exemples de syntaxe :

SPADR\$(2,0)	Fournit le nom de broche du système de la 2e broche du système.
SPADR\$(3)	Fournit le nom de broche du canal de la 3e broche du canal sur le canal actuel.
SPADR\$(1,2,4)	Fournit le nom de broche du canal de la 1re broche du canal sur le canal 4.
SPADR\$(5,2)	Fournit le nom de broche du canal de la 5e broche du canal sur le canal actuel.

7.3 Corrections et aides à la saisie

7.3.1 Décalages d'origine

Généralités

Les ordres CPL suivants permettent de lire, de créer ou de modifier les tables de décalage d'origine (tables DO).

ZOTCR

Description :

Zero Offset Table CReate – Création d'une table DO

Crée une table de décalage d'origine XML sans blocs de correction au sein du système de fichiers de la MTX.

Syntaxe :

---	READ	WRITE	FUNCT	PROC	WAIT	ERRNO	!
ZOTCR (<table>,<canal/modèle>)							
<table>		Nom de la table XML, avec identification de l'extension de fichier ".zot" et, le cas échéant, indication du chemin En l'absence d'indication du chemin, la table est créée dans le répertoire racine.					
<canal/modèle>		Numéro du canal ou nom de la table XML modèle, le cas échéant avec indication du chemin En l'absence d'indication du chemin, la table utilisée est celle enregistrée dans le répertoire racine.					
ERRNO		Variable CPL Si la variable CPL ERRNO est indiquée à un emplacement quelconque dans la liste de paramètres, aucune erreur d'exécution n'est générée. L'erreur est fournie par une valeur correspondante de la variable. <i>Les valeurs de retour suivantes sont possibles :</i> • 0 : Accès OK • < 0 : Erreur (pour une description précise de l'erreur, voir chap. 3.14.5 "Variable ERRNO pour l'analyse des erreurs des fonctions CPL ERRNO" à la page 85) Si la variable CPL ERRNO n'est pas indiquée, une erreur d'exécution est générée en cas d'erreur d'accès.					

Tab. 7-14: Syntaxe ZOTCR

Ordres CPL

Exemple de syntaxe :

```
ZOTCR("/usr/user/tab.zot",1)
```

Création de la nouvelle table "/usr/user/tab.zot" contenant des colonnes pour chaque axe du premier canal.

ZOCINS

Description :

Zero Offset Column INSeRT – Ajout d'un axe dans une table DO

Ajoute un nouvel axe dans la table de décalages d'origine XML avant une colonne existante. Le nouvel axe reçoit autant de blocs de correction que le 1er axe de la table. Les valeurs sont prédéfinies sur 0.0.

Syntaxe :

---	READ	WRITE	FUNCT	PROC	WAIT	ERRNO	!
<pre>ZOCINS (<table>,<position>,<NomAxe> [,<TypeAxe>][,<CanalAxe>][,<CanalPos>]]])</pre>							
<table>	Nom de la table XML, le cas échéant avec indication du chemin En l'absence d'indication du chemin, la table utilisée est celle enregistrée dans le répertoire racine.						
<position>	Nom de coordonnée ou index de colonne du point d'insertion						
<NomAxe>	Nom de l'axe de la nouvelle colonne de table						
<TypeAxe>	Optionnel : type d'axe du nouvel axe 0 : Axe linéaire (valeur par défaut) 1 : Axe rond						
<CanalAxe>	Numéro du canal du nouvel axe Valeur par défaut : 0						
<CanalPos>	Nom du canal de la coordonnée au point d'insertion si un nom de coordonnée a été indiqué ; sinon sans signification						
ERRNO	Variable CPL Si la variable CPL ERRNO est indiquée à un emplacement quelconque dans la liste de paramètres, aucune erreur d'exécution n'est générée. L'erreur est fournie par une valeur correspondante de la variable. <i>Les valeurs de retour suivantes sont possibles :</i> • 0 : Accès OK • < 0 : Erreur (pour une description précise de l'erreur, voir chap. 3.14.5 "Variable ERRNO pour l'analyse des erreurs des fonctions CPL ERRNO" à la page 85) Si la variable CPL ERRNO n'est pas indiquée, une erreur d'exécution est générée en cas d'erreur d'accès.						

Tab. 7-15: Syntaxe ZOCINS

Exemples de syntaxe :

```
ZOCINS("/usr/user/tab.zot", "X", "Y")
```

Ajout d'un nouvel axe dans la table "/usr/user/tab.zot" avec la désignation de colonne "Y" avant la colonne de table ayant la désignation "X".

Variante avec nom de l'axe

```
ZOCINS("/ZOT_Trafo.zot", "X", "U", , 1, 1)
```

Ajout d'un nouvel axe linéaire (pour le canal 1) dans la table "/ZOT_Trafo.zot" avec la désignation de colonne "U" avant la colonne de table ayant la désignation "X" du canal 1.

Une fois cet ordre saisi, l'axe ayant la désignation "X" et le canal 1 est recherché dans la table et remplacé par le nouvel axe linéaire "U" pour canal 1.

Variante avec index de colonne du point d'insertion

```
ZOCINS("/ZOT_Trafo.zot", 1, "U", 1, , 2)
```

Ajout d'un nouvel axe rond (pour le canal 2) dans la table

"/ZOT_Trafo.zot" avec la désignation de colonne "U" dans la colonne de table 1.

ZOCDEL

Description : **Zero Offset Column DElete** – Suppression d'un axe d'une table DO
Suppression d'un axe d'une table de décalages d'origine XML.

Syntaxe :

ZOCDEL (<table>,<position>[,<canal>])	
<table>	Nom de la table XML, le cas échéant avec indication du chemin En l'absence d'indication du chemin, la table utilisée est celle enregistrée dans le répertoire racine.
<position>	Nom de l'axe ou index de colonne de l'axe à supprimer
<canal>	Numéro du canal de l'axe à supprimer Valeur par défaut : 0
ERRNO	Variable CPL Si la variable CPL ERRNO est indiquée à un emplacement quelconque dans la liste de paramètres, aucune erreur d'exécution n'est générée. L'erreur est fournie par une valeur correspondante de la variable. <i>Les valeurs de retour suivantes sont possibles :</i> • 0 : Accès OK • < 0 : Erreur (pour une description précise de l'erreur, voir chap. 3.14.5 "Variable ERRNO pour l'analyse des erreurs des fonctions CPL ERRNO" à la page 85) Si la variable CPL ERRNO n'est pas indiquée, une erreur d'exécution est générée en cas d'erreur d'accès.

Tab. 7-16: Syntaxe ZOCDEL

Exemples de syntaxe :

```
ZOCDEL("/usr/user/tab.zot", "X")
```

Suppression de l'axe ayant la désignation de colonne "X" dans la table "/usr/user/tab.zot".

Variante avec nom d'axe de l'axe à supprimer

```
ZOCDEL("/ZOT_Trafo.zot", "X", 1)
```

Ordres CPL

Suppression de l'axe avec la désignation de colonne "X" dans la table "/ZOT_Trafo.zot".

Une fois cet ordre saisi, l'axe ayant la désignation "X" (sur le canal 1) est recherché dans la table et supprimé.

Variante avec index de colonne de l'axe à supprimer

```
ZOCDEL("/ZOT_Trafo.zot",1)
```

Suppression de l'axe avec l'index de colonne 1 dans la table "/ZOT_Trafo.zot".

Une fois cet ordre saisi, le premier axe de la table est supprimé.

ZOT

DOUBLE	READ	WRITE	FUNCT	PROC	WAIT	ERRNO	!

Description : **Zero Offset Table** – Accès en lecture et en écriture à une table DO

Accès en lecture et en écriture à une table de décalages d'origine XML quelconque au sein du système de fichiers de la commande IndraMotion MTX. Seul l'accès à des éléments individuels est permis. L'écriture de modifications incrémentales est également possible.



L'ordre ZOT écrit directement dans la table indiquée. S'il s'agit d'une table active, les valeurs de correction inscrites prennent effet à partir du bloc CN suivant.

Prudence lors de la mise en œuvre des fonctionnalités "Sous-programme asynchrone" (ASUP), "Éloignement et rapprochement du contour", "Effacement de la course restante" et "Enregistrement avec écrasement" :

Si ces fonctions sont actives, tous les blocs CN déjà préparés sont de nouveau préparés à partir du point d'interruption ! Si des valeurs de correction ont été modifiées avec ZOT au sein de la zone des blocs CN déjà préparés, alors ces valeurs modifiées s'appliquent immédiatement à tous les nouveaux blocs CN préparés.

De même, les valeurs de correction modifiées prennent effet dès le sous-programme asynchrone appelé ou le mouvement d'approche de rapprochement ou d'éloignement.

Si un tel comportement n'est pas souhaitable, un ordre "WAIT" doit être programmé avant chaque ordre ZOT !

En alternative, le mécanisme d'actualisation des tables et registres de correction peut être neutralisé à l'aide de la donnée système **/SysCorrRefrDis**. Dans ce cas, les modifications des valeurs de correction prennent seulement effet à partir de la programmation DO suivante ou de l'activation suivante de la table.

Syntaxe :

ZOT (<sélection de colonne>,<code DO>[,<banque DO>][,<table>][,<unité>] [<canal>]]])	
<sélection de colonne>	Nom de la coordonnée de machine (axe) programmé entre guillemets ou index de colonne dans la table <i>En cas d'accès en écriture, il est possible :</i> - d'écraser la valeur de la table, - de saisir cette valeur de manière additive, si la <sélection de colonne> commence par un signe moins.
<code DO>	54 : 1er décalage d'origine 55 : 2e décalage d'origine 56 : 3e décalage d'origine 57 : 4e décalage d'origine 58 : 5e décalage d'origine 59 : 6e décalage d'origine
<banque DO>	Index de la banque de décalage d'origine (1 à 5) Valeur par défaut : 1
<table>	Nom de la table de décalage d'origine, le cas échéant avec indication absolue ou incrémentale du chemin (le suffixe ".zot" de la table ne doit pas être programmé) Si aucun chemin n'est indiqué, la recherche s'effectue au sein du chemin de recherche configuré. Pour les tables ayant les noms par défaut ZO1, ZO2, etc., il est possible d'indiquer uniquement la valeur numérique. Si aucun nom de table n'est indiqué, la table utilisée est la dernière table activée.
<unité>	0 ou "MM" : millimètres 1 ou "INCH" : pouces Lors d'un accès en écriture, la valeur attribuée est interprétée dans l'unité définie. Lors d'un accès en lecture, la valeur est convertie dans l'unité définie. Valeur par défaut : mm

Ordres CPL

<canal>	Attribution de canal (0 à 60) Le paramètre <canal> ne doit être indiqué que si la <sélection de colonne> est indiquée sous la forme d'un nom de coordonnée de machine. Si le paramètre <sélection de colonne> contient un index de colonne et qu'un <canal> est indiqué, alors une erreur est générée. Les coordonnées de la table DO spécifiques au canal qui ont le même nom peuvent être différenciées en leur attribuant un canal. Valeur par défaut : L'accès à la 1re colonne s'effectue selon la <sélection de colonne>. La saisie du canal est ignorée. L'accès à la 1re colonne trouvée s'effectue avec la <sélection de colonne> et le <canal>. En l'absence d'indication : message d'erreur
ERRNO	Variable CPL Si la variable CPL ERRNO est indiquée à un emplacement quelconque dans la liste de paramètres, aucune erreur d'exécution n'est générée. L'erreur est fournie par une valeur correspondante de la variable. <i>Les valeurs de retour suivantes sont possibles :</i> • 0 : Accès OK • < 0 : Erreur (pour une description précise de l'erreur, voir chap. 3.14.5 "Variable ERRNO pour l'analyse des erreurs des fonctions CPL ERRNO" à la page 85) Si la variable CPL ERRNO n'est pas indiquée, une erreur d'exécution est générée en cas d'erreur d'accès.

Tab. 7-17: Syntaxe ZOT

Exemples de syntaxe :

ZOT ("X", 54)	Accès au décalage G54 de la coordonnée d'axe/de machine "X" dans la dernière table DO activée du canal.
ZOT (2, 55, 2)	Accès au décalage G55.2 de la 2e coordonnée d'axe/de machine enregistrée dans la dernière table DO activée du canal.
ZOT (3, 57, , "V1")	Accès au décalage G57 de la 3e coordonnée d'axe/de machine enregistrée dans la table DO "V1.zot".
ZOT ("Z", 58, 4, 21, 1)	Accès en pouces au décalage G58.4 de la coordonnée d'axe/de machine "Z" dans la table DO "ZO21.zot".
ZOT ("Y", 59, 5, "/mnt/esmuser/de/HHGENIUS", "MM")	Accès métrique au décalage G59.5 de la coordonnée d'axe/de machine "Y" dans la table "HHGENIUS" du répertoire "/mnt/esmuser/de".

ZOT ("Y", 56, 1, "/mnt/ZOF", "MM", 2)

Accès métrique au décalage G56.1 de la coordonnée d'axe/de machine "Y" attribuée au canal 2 dans la table "ZOF" du répertoire "/mnt".

ZOV

DOUBLE	READ	WRITE	FUNCT	PROC	WAIT	ERRNO	!

Description : **Zero Offset Value** – Valeurs de décalage d'origine actives

Fournit la somme des dernières valeurs DO programmées et donc effectives d'une coordonnée de machine (axe) ou, en option, la valeur effective de chaque banque de décalage d'origine.

Syntaxe :

ZOV (<sélection d'axe>[,<banque DO>])	
<sélection d'axe>	Nom de la coordonnée de machine (axe) programmé entre guillemets – ou – Index de la coordonnée de machine (axe), valeurs 1 à 8
<banque DO>	Index de la banque de décalage d'origine, valeurs 1 à 5

Tab. 7-18: Syntaxe ZOV

Exemples de syntaxe :

ZOV ("X")	Fournit la somme de tous les décalages d'origine effectifs (derniers décalages d'origine programmés) pour la coordonnée de machine (axe) X.
ZOV (2, 3)	Fournit le décalage d'origine effectif de la 3e banque DO pour la 2e coordonnée de machine (axe).

7.3.2 Décalage de position et de contour AXO

DOUBLE	READ	WRITE	FUNCT	PROC	WAIT	ERRNO	!

Description : **AXis Offset** – Décalage de position

AXO fournit le décalage SETPOS actuel pour une coordonnée au moment de la préparation des blocs, c.-à-d. le dernier décalage activé au moment de l'interprétation du programme.

Syntaxe :

AXO (<coordonnée>[,<mode de sélection>])	
	(Pour les paramètres, voir chap. 7.1.3 "Ordres de lecture des positions de coordonnées et d'axes" à la page 457 ; ERRNO non programmable)

Tab. 7-19: Syntaxe AXO



AXO donne uniquement accès aux valeurs de décalage du canal concerné. Les axes asynchrones ne possèdent aucun décalage SETPOS, c'est pourquoi AXO ne peut pas être utilisé pour les axes asynchrones.

Exemple :

Canal 1 selon l'exemple de configuration (voir [Fig. 7-2 "Configuration et attribution des noms d'axes aux canaux"](#) à la page 451) :

Ordres CPL

N10 G1 G90 X100 F1000	
N20 SETPOS (X75, Y125)	
30 XD = AXO ("X")	Le dernier décalage SETPOS activé pour la coordonnée X du canal actuel est attribué à XD (XD=100-75=25).
40 YD = AXO (2, 0)	Le dernier décalage SETPOS activé pour le 2e axe du système est attribué à YD (YD=200-125= 5).
50 X2D = AXO (4, 0)	Erreur d'exécution puisque le 4e axe du système est attribué au canal 2.

COF

DOUBLE	READ	WRITE	FUNCT	PROC	WAIT	ERRNO	!

Description : **Contour Offset** – Décalage de contour

Fournit le dernier décalage programmé du contour (Shift) d'une coordonnée. Étant donné que le décalage de contour programmé agit uniquement sur les coordonnées du canal actuel, sélectionner une coordonnée inexistante sur le canal actuel génère un message d'erreur.

Les valeurs de correction sont fournies dans l'unité active sur le canal actuel, c.-à-d. en "pouces" avec G70 ou en "millimètres" avec G71. L'unité est toujours en "degrés" pour les axes ronds et les coordonnées tridimensionnelles rotatives.

Syntaxe :	COF (<coordonnée> [, <mode de sélection>])
	(Pour les paramètres, voir chap. 7.1.3 "Ordres de lecture des positions de coordonnées et d'axes" à la page 457 ; ERRNO non programmable)

Tab. 7-20: Syntaxe COF

Exemple :	10 A=COF (3)	Fournit le dernier décalage de contour programmé de la coordonnée avec le 3e index de coordonnée sur le canal actif.
	:	
	20 B=COF ("X")	Fournit le dernier décalage de contour programmé de la coordonnée X sur le canal actif.
	:	
	30 C=COF (2, 0)	Fournit le dernier décalage de contour programmé du 2e axe du système sur le canal actif.
	:	Cet accès n'est permis que si l'axe du système est identique à la coordonnée du WCS.
	100 C=COF (0)	Erreur d'exécution puisque 0 n'est pas un index de coordonnée valide.
	:	

7.3.3 Correction de la position de la pièce à usiner et placements (plan incliné)

DPC

DOUBLE	READ	WRITE	FUNCT	PROC	WAIT	ERRNO	!

Description : **Datum Point Correction** – Correction de la position de la pièce à usiner

Ordres CPL

Fournit les derniers paramètres programmés de la correction de la position de la pièce à usiner (BCR) d'une coordonnée (valeurs de décalage et angles de torsion).

Étant donné que la correction de position de la pièce à usiner n'agit que sur les coordonnées du canal actuel, sélectionner une coordonnées inexistante sur le canal actuel génère un message d'erreur.

Les valeurs de correction sont fournies dans l'unité de mesure active du canal actuel, c.-à-d. en "pouces" pour G70 et en "millimètres" pour G71. L'unité est toujours en "degrés" pour les axes ronds et les coordonnées tridimensionnelles rotatives.

Syntaxe :

DPC (<coordonnée> [,<mode de sélection>])	
(Pour les paramètres, voir chap. 7.1.3 "Ordres de lecture des positions de coordonnées et d'axes" à la page 457 ; ERRNO non programmable)	
<coordonnée>	Action supplémentaire des indications : "1" à "n" ou "Nom" fournissent la valeur de décalage. "0" fournit l'angle de rotation Phi.

Tab. 7-21: Syntaxe DPC

Exemple :

10 A=DPC (1)	Fournit la dernière correction de position de la pièce à usiner programmée pour la coordonnée avec le 1er index de coordonnée sur le canal.
:	
15 B=DPC ("X")	Fournit la dernière correction de position de la pièce à usiner programmée pour l'axe X/la coordonnée sur le canal.
:	
20 B=DPC (2)	Fournit la dernière correction de la position de la pièce à usiner programmée pour la coordonnée avec le 2e index de coordonnée sur le canal.
:	
25 B=DPC (2, 0)	Fournit la dernière correction de position de la pièce à usiner programmée pour le 2e axe du système sur le canal actif.
:	
30 ANGLE=DPC (0)	Fournit le dernier angle de rotation Phi programmé.
:	
100 C=DPC (9)	Erreur d'exécution puisque 9 n'est pas un index de coordonnée valide avec 8 axes dans le système.
:	

PMV

DOUBLE	READ	WRITE	FUNCT	PROC	WAIT	ERRNO	!

Description : PlaceMent Value – Valeurs de placement actives

Fournit la somme des dernières valeurs de placement programmées et donc effectives d'une coordonnée. Il est également possible de lire la valeur effective d'une banque de placement individuelle.

Syntaxe :

PMV (<sélection coord.>[,<banque pl.>])	
<sélection coord.>	Index de la coordonnée (1 à 6) ou désignateur fixe "X", "Y", "Z", "PHI", "THE", "PSI" pour les coordonnées du WCS
<banque pl.>	Index de la banque de placement (1 à 5)

Tab. 7-22: Syntaxe PMV

<table>	Nom du tableau de placement, le cas échéant avec indication absolue ou incrémentale du chemin (le suffixe ".pmt" des tableaux ne doit pas être programmé). Si aucun chemin n'est indiqué, la recherche s'effectue au sein du chemin de recherche configuré. Concernant les tableaux ayant un nom par défaut PM1, PM2, etc., il suffit d'indiquer leur valeur numérique. Si aucun nom de tableau n'est indiqué, le tableau utilisé est le dernier tableau activé.
<unité>	0 ou "MM" : millimètres 1 ou "INCH" : pouces Concernant les 3 coordonnées linéaires, la valeur attribuée est interprétée dans l'unité indiquée (en cas d'accès en écriture) ou transformée dans l'unité indiquée (en cas d'accès en lecture). Valeur par défaut : mm

Tab. 7-23: Syntaxe PMT

Exemples de syntaxe :

PMT ("Z", 155)	Accès au décalage G155 de la coordonnée "Z" dans le dernier tableau de placement activé du canal.
PMT (5, 157, 3)	Accès à l'angle d'Euler ϑ de G157.3 dans le dernier tableau de placement activé du canal.
PMT ("X", 154, , "P1")	Accès au décalage G154.1 de la coordonnée "X" dans le tableau de placement P1.
PMT (3, 158, 4, , 1)	Accès en pouces au décalage G158.4 de la coordonnée "Z" dans le dernier tableau de placement activé du canal.
PMT ("Y", 159, 5, "/mnt/esmuser/de/PL5", "MM")	Accès métrique au décalage G159.5 de la coordonnée "Y" dans le tableau "PL5" du répertoire "/mnt/esmuser/de".

ROT2EUL

---	READ	WRITE	FUNCT	PROC	WAIT	ERRNO	!

Description : **ROTation to EULer angles** – Calcul de l'angle d'Euler à partir de rotations autour des axes

ROT2EUL est un ordre CPL permettant de calculer des angles d'Euler. Cet ordre permet de définir des rotations successives autour des axes X, Y et/ou Z, qui sont ensuite converties en angles d'Euler pour les placements.

Contrairement à ABSROT2EUL (voir [chap. " ABSROT2EUL" à la page 480](#)), les rotations ne s'effectuent pas ici autour d'axes fixes, mais toujours autour des axes du nouveau système de coordonnées (tourné).

Ordres CPL

Syntaxe :

ROT2EUL (<angle d'Euler>, <axe de rotation1>, <angle1> [, <axe de rotation2>, <angle2> [, <axe de rotation3>, <angle3>]])	
<angle d'Euler>	Champ composé de trois valeurs DOUBLE Entrée : angle d'Euler autour duquel le système de coordonnées de départ est tourné. Sortie : angle d'Euler autour duquel le système de coordonnées résultant est tourné.
<axe de rotation1..3>	Axe autour duquel la rotation doit avoir lieu ("X", "Y" ou "Z").
<angle1..3>	Angle selon lequel la rotation doit avoir lieu, en degrés.

Tab. 7-24: Syntaxe ROT2EUL

Exemple :

Programme:

```

1 REM calcule l'angle d'Euler pour la rotation de X (20°) puis de Z (60°)
2 DIM EULER!(3)
3 EULER!(1) = 0 : EULER!(2) = 0 : EULER!(3) = 0
4 ROT2EUL(EULER!, "X", 20, "Z", 60)
5 REM résultat dans le champ EULER! (phi, theta, psi)

```

ABSROT2EUL

---	READ	WRITE	FUNCT	PROC	WAIT	ERRNO	!

Description :

ABSolute ROTation to EULER angles – Calcul des angles d'Euler à partir de rotations autour d'axes fixes

ABSROT2EUL est un ordre CPL permettant de calculer des angles d'Euler. Il est possible d'affecter un maximum de trois rotations autour des axes X, Y et/ou Z (fixes) du système de coordonnées de base (BCS), qui seront ensuite converties en angles d'Euler pour les placements.

Contrairement à ROT2EUL (voir [chap. " ROT2EUL " à la page 479](#)), la rotation ne s'effectue pas autour des axes du nouveau système de coordonnées, mais toujours autour des axes du système de coordonnées de base.

Syntaxe :

ABSROT2EUL (<angle d'Euler>, <axe de rotation1>, <angle1> [, <axe de rotation2>, <angle2> [, <axe de rotation3>, <angle3>]])	
<angle d'Euler>	Champ composé de trois valeurs DOUBLE Entrée : angle d'Euler autour duquel le système de coordonnées de départ est tourné. Sortie : angle d'Euler autour duquel le système de coordonnées résultant est tourné.
<axe de rotation1..3>	Axe autour duquel la rotation doit avoir lieu ("X", "Y" ou "Z").
<angle1..3>	Angle selon lequel la rotation doit avoir lieu, en degrés.

Tab. 7-25: Syntaxe ABSROT2EUL

Exemple :

Programme:

```

1 REM calcule l'angle d'Euler pour la rotation de Y (45°) d'un COS déjà tourné
2 DIM EULER!(3)
3 EULER!(1) = 90 : EULER!(2) = -30 : EULER!(3) = 270
4 ABSROT2EUL(EULER!, "Y", 45)
5 REM résultat dans le champ EULER! (phi, theta, psi)

```

PT2EUL

---	READ	WRITE	FUNCT	PROC	WAIT	ERRNO	!

Description : **PoinTs to EULer angles** – Calcul des angles d'Euler pour un plan incliné dont 3 points sont indiqués

PT2EUL est un ordre CPL permettant de calculer des angles d'Euler. Trois points (valeurs X, Y et Z) devant se situer sur un plan sont transmis à l'ordre. Les angles d'Euler sont calculés à partir de ces points de sorte que ce plan sous-tende le nouveau plan XY avec un placement. Il est ainsi possible, par exemple, d'effleurer une surface avec trois points et de déterminer les angles d'Euler pour un placement correspondant à l'aide de PT2EUL.

Les trois points doivent tous être différents les uns des autres et ils ne doivent pas être alignés sur la même droite. Bien que les paramètres suffisent à déterminer entièrement le nouveau plan, seuls les angles d'Euler calculés sont renvoyés. Étant donné que Pt1 est égal l'origine, il est possible de définir le plan directement avec les angles et Pt1.

Syntaxe :

PT2EUL (<angle d'Euler>, <Pt1X>, <Pt1Y> ,<Pt1Z>, <Pt2X>, <Pt2Y> ,<Pt2Z>, <Pt3X>, <Pt3Y> ,<Pt3Z>)	
<angle d'Euler>	Champ composé de trois valeurs DOUBLE Entrée : angle d'Euler autour duquel le système de coordonnées de départ est tourné. Sortie : angle d'Euler autour duquel le système de coordonnées résultant est tourné.
<Pt1XYZ>	Origine du nouveau plan par rapport au système de coordonnées d'origine
<Pt2XYZ>	Le nouvel axe X passe par Pt2 par rapport au système de coordonnées d'origine.
<Pt3XYZ>	Pt3 se situe sur le nouveau plan XY (par rapport au système de coordonnées d'origine).

Tab. 7-26: Syntaxe PT2EUL

Exemple : *Programme:*

```
1 REM Les points palpés se trouvent dans PT1! à PT3!
2 DIM EULER!(3)
3 EULER!(1) = 0 : EULER!(2) = 0 : EULER!(3) = 0
4 PT2EUL(EULER!, PT1!(1), PT1!(2), PT1!(3), PT2!(1), PT2!(2), PT2!(3), PT3!(1), PT3!(2), PT3!(3))
5 REM Résultat dans le champ EULER! (phi, theta, psi)
; le placement peut directement être défini avec G152
G152.1([PT1!(1)], [PT1!(2)], [PT1!(3)], [EULER!(1)], [EULER!(2)], [EULER!(3)])
```

VEC2EUL

---	READ	WRITE	FUNCT	PROC	WAIT	ERRNO	!

Description : **VEctors to EULer angles** – Calcul des angles d'Euler pour un plan incliné, dont le vecteur normal et la direction du nouvel axe X sont indiqués

VEC2EUL est un ordre CPL qui permet de calculer les angles d'Euler. Un plan est déterminé par un vecteur normal (perpendiculaire au plan). De plus, il est possible en option de programmer une direction pour l'axe X. L'ordre calcule l'angle d'Euler pour ce plan.

Si le deuxième vecteur n'est pas perpendiculaire au vecteur normal, sa projection verticale est utilisée pour déterminer l'axe X. Les deux vecteurs ne doivent en aucun cas être parallèles. Si aucun vecteur n'est programmé pour

Ordres CPL

l'axe X, alors l'axe Y est déterminé de sorte qu'il sera parallèle à l'ancien plan XY. L'angle formé correspond alors au deuxième angle d'Euler (identique à G151).

Syntaxe :

VEC2EUL (<angle d'Euler>, <Xnorm>, <Ynorm> ,<Znorm> [, <XaxeX>, <XaxeY> ,<XaxeZ>])	
<angle d'Euler>	Champ composé de trois valeurs DOUBLE Entrée : angle d'Euler autour duquel le système de coordonnées de départ est tourné. Sortie : angle d'Euler autour duquel le système de coordonnées résultant est tourné.
<XYZnorm>	Vecteur normal du nouveau plan par rapport au système de coordonnées d'origine
<XaxeXYZ>	Vecteur de direction du nouvel axe X par rapport au système de coordonnées d'origine

Tab. 7-27: Syntaxe VEC2EUL

Exemple :**Programme:**

```
1 REM calcule l'angle d'Euler avec un vecteur normal
2 DIM EULER!(3)
3 EULER!(1) = 0 : EULER!(2) = 0 : EULER!(3) = 0
4 VEC2EUL(EULER!, 1, 0.5, 0)
5 REM Résultat dans le champ EULER! (phi, theta, psi)
```

WCS2BCS

---	READ	WRITE	FUNCT	PROC	WAIT	ERRNO	!

Description :

WCS to BCS – Conversion d'un point du système de coordonnées de la pièce à usiner (WCS) dans le système de coordonnées de base (BCS)

WCS2BCS est un ordre CPL permettant de convertir un point. Un point transféré du système de coordonnées de la pièce à usiner (WCS) est converti dans le système de coordonnées de base (BCS). Les deux points se distinguent par leurs placements intercalés.

En option, il est possible de transférer les données de placement à l'ordre. Si ces données ne sont pas programmées, les placements utilisés pour le calcul sont ceux actuellement actifs (sans BCR).

Syntaxe :

WCS2BCS (<point> [, <placement>])	
<point>	Champ composé de trois valeurs DOUBLE Entrée : point dans le WCS Sortie : point dans le BCS
<placement>	Champ composé de six valeurs DOUBLE Données d'un placement (X, Y, Z, phi, theta, psi)

Tab. 7-28: Syntaxe WCS2BCS

Exemple :**Programme:**

```
; définis un placement
G151.1(10,20,30, 45,30,0)
WAIT
; l'origine d'un nouveau placement doit se trouver sur (10,0,50) par rapport au placement actuel
2 DIM PT!(3)
3 PT!(1) = 10 : PT!(2) = 0 : PT!(3) = 50
4 WCS2BCS(PT!)
; résultat dans le champ PT!
```



```
; définis le nouveau placement et écrase l'ancien
G151.1([PT!(1)], [PT!(2)], [PT!(3)], -45, 30, 0)
```

BCS2WCS

---	READ	WRITE	FUNCT	PROC	WAIT	ERRNO	!

Description : **BCS to WCS** – Conversion d'un point du système de coordonnées de base (BCS) dans le système de coordonnées de la pièce à usiner (WCS)

BCS2WCS est un ordre CPL permettant de convertir un point. Un point transféré du système de coordonnées de base (BCS) est converti dans le système de coordonnées de la pièce à usiner (WCS). Les deux points se distinguent par leurs placements intercalés.

En option, il est possible de transférer les données de placement à l'ordre. Si ces données ne sont pas programmées, les placements utilisés pour le calcul sont ceux actuellement actifs (sans BCR).

Syntaxe :

BCS2WCS (<point> [, <placement>])	
<point>	Champ composé de trois valeurs DOUBLE Entrée : point dans le BCS Sortie : point dans le WCS
<placement>	Champ composé de six valeurs DOUBLE Données d'un placement (X, Y, Z, phi, theta, psi)

Tab. 7-29: Syntaxe BCS2WCS

Exemple : *Programme:*

```
; définis un placement
G151.1(10,20,30, 45,30,0)
WAIT
; le point mesuré auparavant dans le champ PT! est présent dans le BCS et doit à présent être approché
2 BCS2WCS(PT!)
; résultat dans le champ PT!
; approche le point dans le nouveau système de coordonnées
G1 F500 X[PT!(1)] Y[PT!(2)] Z[PT!(3)]
```

7.3.4 Aides à la saisie

SCL

DOUBLE	READ	WRITE	FUNCT	PROC	WAIT	ERRNO	!

Description : **SCaLing** – Valeurs actives pour la mise à l'échelle, la fonction miroir, la rotation et le pôle

Fournit les derniers paramètres programmés des fonctions CN PLS, SCL, MIR et ROT(...) (coordonnées du pôle, facteurs de mise à l'échelle et angles de rotation) pour le canal actuel (ici : canal sur lequel le programme est exécuté avec l'ordre **SCL**).

Étant donné que PLS, SCL, MIR et ROT(...) agissent uniquement sur les coordonnées du canal actif, sélectionner une coordonnée inexistante sur le canal actuel génère un message d'erreur.

Les valeurs de position sont fournies dans l'unité de mesure active sur le canal actuel, c.-à-d. en "pouces" pour G70 et en "millimètres" pour G71. L'unité est toujours en "degrés" pour les axes rotatifs et les coordonnées tridimensionnelles rotatives.

Ordres CPL

Syntaxe :

SCL (<sélection>[,<coordonnée>[,<mode de sélection>]])	
<sélection>	0 : dernier angle de rotation programmé pour le plan principal 1 : dernier pôle programmé pour un axe du canal 2 : dernier facteur de mise à l'échelle programmé pour un axe du canal
<coordonnée>	Index ou nom d'une coordonnée : Un nom est interprété en tant que nom de coordonnée. S'il n'existe aucun nom de coordonnée correspondant, et seulement dans ce cas, le nom est interprété en tant que nom de canal ou d'axe du système. Un index est interprété selon le <mode de sélection> indiqué. La programmation d'une coordonnée/d'un axe non configuré(e) entraîne une erreur d'exécution si la variable optionnelle ERRNO n'est pas programmée.
<mode de sélection>	Optionnel : Définit la manière dont un index programmé sous <coordonnée> doit être interprété : 0 : index d'axe du système 1 : index de coordonnée (par défaut) Si aucun <mode de sélection> n'est programmé, l'index est interprété en tant qu'index de coordonnée !

Tab. 7-30: Syntaxe SCL

Exemple :

10 W=SCL(0)	Inscrit le dernier angle de rotation programmé dans la variable W.
:	
20 P=SCL(1,2)	Inscrit le 2e index de coordonnée sur le canal comme pôle de la coordonnée dans la variable P.
:	
30 F=SCL(2,2,1)	Inscrit le 2e index de coordonnée comme facteur de mise à l'échelle de la coordonnée dans la variable F.
:	
40 D=SCL(2,"X")	Inscrit le facteur de mise à l'échelle de la coordonnée X du canal actif dans la variable D.

7.3.5 Corrections d'outil

Généralités

Les ordres CPL suivants donnent accès aux données de correction d'outil internes (correction D) ainsi qu'aux données de correction d'outil externes (correction ED) auxquelles l'API a accès.

TCV

DOUBLE	READ	WRITE	FUNCT	PROC	WAIT	ERRNO	!

Description :

Tool Correction Values – Dernières valeurs de correction d'outil appliquées

Fournit les dernières valeurs de correction d'outil programmées soit en tant que valeur de la somme (correction D + mémoire de correction externe), soit en tant que valeur unique.

Syntaxe :

TCV (<sélection de valeurs>[,<sélection de correction>])	
<sélection de valeurs>	Numéro de la correction D/ED : 0 : dernière correction D/ED activée 1 ou "L1" : longueur de correction L1 2 ou "L2" : longueur de correction L2 3 ou "L3" : longueur de correction L3 4 ou "RAD" : rayon de l'outil/de coupe 5 ou "ORI" : position de l'arête coupante 6 ou "PHI" : angle d'Euler ϕ (correction ED uniquement) 7 ou "THE" : angle d'Euler ϑ (correction ED uniquement) 8 ou "PSI" : angle d'Euler ψ (correction ED uniquement)
<sélection de correction>	1 ou "D" : correction D (interne) 2 ou "E" : correction ED (externe) Valeur par défaut : somme des corrections D et ED

Tab. 7-31: Syntaxe TCV

Exemples de syntaxe :

TCV (0, "D")	Fournit la dernière correction D programmée.
TCV ("L1")	Lecture de la somme des valeurs L1 des dernières corrections D et ED programmées
TCV (4, 1)	Lecture du rayon de l'outil de la dernière correction D programmée
TCV (2, "E")	Lecture de la longueur de l'outil L2 de la dernière correction ED programmée

DCT

DOUBLE	READ	WRITE	FUNCT	PROC	WAIT	ERRNO	!

Description :

D-Correction Table – Accès aux tables de correction D et registres de correction ED

Accès en lecture et en écriture aux valeurs de correction d'outil dans une table de correction D quelconque ainsi qu'aux valeurs de correction externes (correction ED). L'écriture de modifications incrémentales est également possible.

Ordres CPL



L'ordre DCT écrit directement dans la table ou le registre de correction indiqué. S'il s'agit d'une table active ou d'un registre de correction actif, les valeurs de correction inscrites prennent effet à partir du bloc CN suivant.

Prudence lors de la mise en œuvre des fonctionnalités "Sous-programme asynchrone" (ASUP), "Éloignement et rapprochement du contour", "Effacement de la course restante" et "Enregistrement avec écrasement" :

Si ces fonctions sont actives, tous les blocs CN déjà préparés sont de nouveau préparés à partir du point d'interruption ! Si des valeurs de correction ont été modifiées avec DCT au sein de la zone des blocs CN déjà préparés, alors ces valeurs modifiées s'appliquent immédiatement à tous les nouveaux blocs CN préparés.

De même, les valeurs de correction modifiées prennent effet dès un sous-programme asynchrone appelé ou dès le mouvement d'approche de rapprochement ou d'éloignement.

Si un tel comportement n'est pas souhaitable, l'ordre "WAIT" doit être programmé avant chaque ordre DCT !

En alternative, le mécanisme d'actualisation des tables et registres de correction peut être neutralisé à l'aide de la donnée système **/SysCorrRefrDis**. Dans ce cas, les modifications des valeurs de correction prennent seulement effet lors de la correction suivante, de l'activation suivante de la table ou de la sélection suivante de l'arête de coupe.

Syntaxe :

DCT (<sélection de valeurs>,<jeu de données>[,<table>][,<unité>])	
<sélection de valeurs>	Numéro de la correction D/ED : 1 ou "L1" : longueur de correction L1 2 ou "L2" : longueur de correction L2 3 ou "L3" : longueur de correction L3 4 ou "RAD" : rayon de l'outil/de coupe 5 ou "ORI" : position de l'arête coupante 6 ou "PHI" : angle d'Euler ϕ (correction ED uniquement) 7 ou "THE" : angle d'Euler ϑ (correction ED uniquement) 8 ou "PSI" : angle d'Euler ψ (correction ED uniquement) La valeur peut être lue ou écrite. En cas d'accès en écriture, il est possible : - d'écraser la valeur de la table, - d'additionner cette valeur à la valeur de la table, si <sélection de valeurs> commence par un signe moins. L'écriture incrémentale n'est pas possible pour Position de l'arête coupante (5 ou "ORI").
<jeu de données>	1 à 99 : pour une correction D 1 à 16 : pour une correction ED

<table>	Nom de la table de correction D, le cas échéant avec indication absolue ou incrémentale du chemin (sans l'extension de table ".dct") Si aucun chemin n'est indiqué, la recherche s'effectue au sein du chemin de recherche configuré. Pour les tables avec le nom par défaut DC1, DC2, etc., il suffit d'indiquer la valeur numérique. Si aucune <table> n'est programmée, la table utilisée est la dernière table de correction D sélectionnée. 0 : accès à la correction ED du canal actuel -1 : accès à la correction ED du canal 1 -2 : accès à la correction ED du canal 2 -3 : accès à la correction ED du canal 3 ... -60 : accès à la correction ED du canal 60
<unité>	Uniquement pertinente pour les tables de correction D : 0 ou "MM" : millimètres (par défaut) 1 ou "INCH" : pouces Lors d'un accès en écriture, la valeur attribuée est interprétée dans l'unité définie. Lors d'un accès en lecture, la valeur est convertie dans l'unité définie. Lors d'un accès aux corrections ED, la valeur est toujours interprétée et fournie dans la dernière unité sélectionnée sur le canal appelant via G70/G71.

Tab. 7-32: Syntaxe DCT

Exemples de syntaxe :

DCT (1, 10, "K4")	Accès à la valeur de correction "L1" du jeu de données 10 dans la table de correction D "K4".
DCT ("RAD", 7)	Écriture incrémentale du rayon d'outil du jeu de données 7 de la dernière table activée.
DCT (3, 1, 0)	Accès à la correction "L3" de la correction d'outil externe du canal.
DCT ("L2", 16, 1, "INCH")	Accès en pouces à la correction "L2" du jeu de données 16 de la table DC1.dct.
DCT (4, 7, -3)	Accès au rayon de l'outil pour le bloc de correction 7 du 3e canal.



Lors de la lecture/de l'écriture des données de correction d'outil de canaux tiers, aucun accès concurrentiel ne doit provenir de canaux différents.



Lors d'un accès aux corrections ED, la valeur est toujours interprétée et fournie dans la dernière unité sélectionnée sur le canal appelant via G70/G71.

Ordres CPL

7.4 Données et processus de la commande

7.4.1 Version de la commande

VERSINF\$

CHAR	READ	WRITE	FUNCT	PROC	WAIT	ERRNO	!

Description : **VERS**ion **IN**formation – Version de la commande

VERSINF\$ permet d’interroger les données administratives de la commande IndraMotion MTX dans le programme CPL. La fonction fournit une valeur de type STRING.

Syntaxe : VERSINF\$(<index1>[,<index2>])	
<index1>	Type Constante INTEGER Mode d’ordre : 1 : lire la version logicielle 2 : lire la version matérielle
<index2>	Type INTEGER. Mode supplémentaire de l’ordre, qui dépend de la valeur du paramètre <index1>. <index1> = 1 : aucun mode supplémentaire disponible <index1> = 2 : mode supplémentaire disponible :
	1 : type de matériel (par défaut) 2 : n° du groupe de commutation 3 : index du n° du groupe de commutation

Tab. 7-33: Syntaxe VERSINF\$

Exemple :

```
10 DIM SYS_INFO$(50)
20 SYS_INFO$ = VERSINF$(1)
30 PRN#(0,"Version logicielle : ", SYS_INFO$)
M30
```

7.4.2 Saisie du temps

CLOCK

INT	READ	WRITE	FUNCT	PROC	WAIT	ERRNO	!

Description **CLOCK** – Temps système

Lit le temps système interne à la commande ou le compteur de cycles d’interpolation interne. Le temps d’exécution d’un extrait de programme peut être déterminé à partir de la différence entre deux résultats de mesure.

Syntaxe :

CLOCK(<mode>)	
<mode>	Optionnel (par défaut : 0) 0 : indication du temps système en millisecondes 1 : indication du temps système en microsecondes 2 : indication de l'état actuel du compteur de cycles d'interpolation multiplié par le cycle d'interpolation configuré, en millisecondes

Tab. 7-34: Syntaxe CLOCK

Exemple :

```

:
20 WAIT
30 TEMPS_INITIAL% = CLOCK(2)
N4 G1 X50 Y70
40 WAIT
50 TEMPS_FINAL%=CLOCK(2) : DIFF%=TEMPS_FINAL%-TEMPS_INITIAL%
:
    
```

L'état actuel du compteur de cycles d'interpolation des variables "TEMPS_INITIAL%" et "TEMPS_FINAL%" est attribué respectivement avant et après l'exécution du bloc N4. La différence des contenus des deux variables sert à déterminer le temps d'exécution du bloc N4 en millisecondes.

Attention, l'ordre "WAIT" est absolument nécessaire pour cela !



La mesure de la durée d'exécution d'un extrait de programme par le temps système interne de la commande ne fournit aucun résultat probant sur une émulation de commande MTX !

En revanche, une mesure effectuée via le nombre de cycles IPO entraîne des temps d'exécution quasiment identiques pour l'émulation de commande MTX et les commandes réelles dans la mesure où il n'y a aucune temporisation due aux blocs CN non préparés, par exemple en cas de problèmes avec le temps de cycle des blocs ou avec des passages CPL longs.

DATE

CHAR	READ	WRITE	FUNCT	PROC	WAIT	ERRNO	!

Description :

DATE – Date

Fournit des données relatives à la date actuelle.

Ordres CPL

Syntaxe :

DATE[(<i><mode></i>)]	
<i><mode></i>	Optionnel (par défaut : 0) 0 : Indication au format "JJ.MM" 1 : Indication au format "JJ.MM.AA" 2 : Indication du jour de la semaine : 1 = lundi, 2 = mardi, etc. 3 : Jour courant de l'année, par ex. "40" pour le 9 février 4 : Semaine calendaire de l'année

Tab. 7-35: Syntaxe DATE



Si DATE est utilisé plusieurs fois au sein d'un programme, il est nécessaire de dimensionner les variables de résultat correspondantes. Sinon, la dernière valeur de l'ordre DATE qui est lue sera attribuée à toutes les variables non dimensionnées qui contiennent le résultat d'une affectation de DATE.

Exemple :

:	
10 DIM A\$ (5)	Dimensionnement de la variable STRING A\$
20 DIM B\$ (8)	Dimensionnement de la variable STRING B\$
30 DIM C\$ (1)	Dimensionnement de la variable STRING C\$
40 DIM D\$ (3)	Dimensionnement de la variable STRING D\$
50 DIM E\$ (2)	Dimensionnement de la variable STRING E\$
60 A\$ = DATE	La date est attribuée à la variable STRING A\$ sous la forme "JJ.MM".
70 B\$ = DATE (1)	La date est attribuée à la variable STRING B\$ sous la forme "JJ.MM.AA".
80 C\$ = DATE (2)	Le jour de la semaine est attribué à la variable STRING C\$ sous la forme d'une valeur numérique comprise entre "1" et "7".
90 D\$ = DATE (3)	Le jour courant de l'année est attribué à la variable STRING D\$ sous la forme d'une valeur numérique comprise entre "1" et "366".
100 E\$ = DATE (4)	La semaine calendaire de l'année est attribuée à la variable STRING E\$ sous la forme d'une valeur numérique comprise entre "1" et "53".
:	

TIME

CHAR	READ	WRITE	FUNCT	PROC	WAIT	ERRNO	!

Description : **TIME** – Heure
Fournit la valeur actuelle de l'heure.

Exemple :

```

:
40 B$ = TIME      L'heure est attribuée à la variable STRING B$ sous la forme
:                  "HH.MM.SS".
    
```



Si TIME est utilisé plusieurs fois au sein d'un programme, il est nécessaire de dimensionner les variables de résultat correspondantes. Sinon, la dernière valeur de l'ordre TIME qui est lue sera attribuée à toutes les variables non dimensionnées qui contiennent le résultat d'une affectation de TIME.

7.4.3 Paramètres machine

MCA

MULTYP	READ	WRITE	FUNCT	PROC	WAIT	ERRNO	!

Description : **MAchine COnfiguration DAta** – Paramètres machine

Transfère le contenu d'un paramètre MACODA individuel. En fonction du type de donnée, cette valeur peut être de type "INT", "REAL", "DOUBLE" ou "CHAR". La variable dans laquelle la valeur transférée est enregistrée doit être de même type !

Les conflits de type entre la valeur transférée et la variable cible sont détectés lors de l'exécution du programme et acquittés avec un message d'erreur.

Syntaxe :

MCA (<IdBloc>, <index paramètre>[, <index classe>])	
<IdBloc>	Numéro d'un paramètre MACODA Un paramètre MACODA peut comprendre plusieurs paramètres MACODA individuels (liste de paramètres). Si un numéro inexistant est programmé, une erreur d'exécution est générée.
<index paramètre>	Index du paramètre MACODA individuel, commençant par "0" Si un numéro d'index inexistant est programmé, une erreur d'exécution est générée.
<index classe>	Optionnel : uniquement pour les paramètres machine spécifiques au canal, à la broche ou à la transformation. Ce paramètre permet d'affecter l'index du canal, de la broche ou de la transformation pour lequel la valeur du paramètre machine doit être lue. En l'absence de programmation et s'il s'agit d'un paramètre de canal, l'ordre fournit le paramètre MACODA individuel du canal sur lequel le programme CPL est actuellement traité. Si un index de classe non valide est programmé, une erreur d'exécution est générée.

Tab. 7-36: Syntaxe MCA

Exemple : **Ordre MCA**

```

10 BLOCKNO%=100300004
20 ERG%=MCA(BLOCKNO%,0)
    
```

Le contenu du premier paramètre individuel du paramètre machine 100300004 est attribué à la variable INTEGER ERG%.

Ordres CPL

7.4.4 Données système simples

SD (description générale)

	INT	READ	WRITE	FUNCT	PROC	WAIT	ERRNO	!
Description :	System Data – Données simples internes au système au format INTEGER Lit des données à partir d'une collection de données internes au système de la commande IndraMotion MTX au format INTEGER .							
Syntaxe :	SD (<groupe>[,<index1>[,<index2>[,<index3>]]])							
	L'ordre SD fournit des valeurs INTEGER.							

Tab. 7-37: Syntaxe SD

SD(2,...) – Position d'Override actuelle

Syntaxe :	Valeur de retour
SD(2,1)	Fournit la position actuelle de l'avance Override sur le canal actuel.
SD(2,2)	Fournit la position actuelle de l'avance rapide Override sur le canal actuel.
SD(2,3) ($\hat{=}$ SD(202,1))	Fournit la position actuelle de l'Override pour la broche 1. 0, si aucune broche n'est paramétrée.
SD(2,4) ($\hat{=}$ SD(202,1))	Fournit la position actuelle de l'Override pour la broche 2. 0, si aucune broche n'est paramétrée.

Tab. 7-38: SD(2,...)

SD(5,...) – Vitesse et vitesse de rotation

Syntaxe :	Valeur de retour
	Valeurs actives :
SD(5,1,1)	Fournit la vitesse d'avance active (valeur 100 %) arrondie à une valeur entière, en mm/min ou en pouces/min.
SD(5,2,1)	Fournit la vitesse d'avance rapide active (valeur 100 %) arrondie à une valeur entière, en mm/min ou en pouces/min.
SD(5,3,1) ($\hat{=}$ SD(205,1,2))	Fournit la vitesse de rotation actuelle de la broche (valeur de consigne) de la 1re broche, en tr/min (y compris l'évaluation de l'Override). 0, si aucune broche n'est paramétrée.
SD(5,4,1) ($\hat{=}$ SD(205,1,2))	Fournit la vitesse de rotation actuelle de la broche (valeur de consigne) de la 2e broche, en tr/min (y compris l'évaluation de l'Override). 0, si aucune broche n'est paramétrée.
	Valeurs programmées :
SD(5,1,2)	Fournit la dernière avance programmée, en unité de saisie par minute.

Syntaxe :	Valeur de retour
SD(5,3,2) ($\triangleq$ SD(205,2,1))	Fournit la dernière vitesse de rotation de la broche (valeur de consigne) programmée sur le canal de la 1re broche, en tr/min. Si aucune broche n'est paramétrée, la valeur renvoyée est 0.
SD(5,4,2) ($\triangleq$ SD(205,2,2))	Fournit la dernière vitesse de rotation de la broche (valeur de consigne) programmée sur le canal de la 2e broche, en tr/min. Si aucune broche n'est paramétrée, la valeur renvoyée est 0.
	Valeurs réelles :
SD(5,3,3) ($\triangleq$ SD(205,3,1))	Fournit la vitesse de rotation réelle de la 1re broche.
SD(5,4,3) ($\triangleq$ SD(205,3,2))	Fournit la vitesse de rotation réelle de la 2e broche.

Tab. 7-39: SD(5,...)

SD(8) – Interrogation du canal

Syntaxe :	Valeur de retour
SD(8)	Fournit le numéro de canal du canal appelant.

Tab. 7-40: SD(8)

SD(9) – Palpeur de mesure

Syntaxe :	Valeur de retour
	Informe si un palpeur de mesure a été déclenché après G75 ou G75.2, ou si "Mesure sur butée fixe" (FSP resp. FsProbe) est active :
SD(9)	• 0 : – après G75 ou G75.2 : palpeur de mesure activé – après FSP (FsProbe) : mesure sur butée fixe exécutée • 1 : – après G75 ou G75.2 : palpeur de mesure désactivé – après FSP (FsProbe) : mesure sur butée fixe non encore exécutée

Tab. 7-41: SD(9)

Ordres CPL

SD(10,...) – Index de la coordonnée sur laquelle la correction L1, L2 ou L3 agit

Syntaxe :	Valeur de retour
	Si la correction est calculée dans le système de coordonnées de la pièce à usiner, fournit à SD(10,...) l'index d'axe du canal de l'axe sur lequel la correction L1, L2 ou L3 agit. Si la correction est calculée dans le système de coordonnées de l'outil (TCS), fournit à SD(10,...) les valeurs de retour suivantes : 101 : calcul sur l'axe X du TCS $\triangleq$ XTR 102 : calcul sur l'axe Y du TCS $\triangleq$ YTR 103 : calcul sur l'axe Z du TCS $\triangleq$ ZTR
SD(10,1,<corr>) SD(10,1) ($\triangleq$ SD(10,1,3))	Au moment de la préparation du bloc
SD(10,2,<corr>) SD(10,2) ($\triangleq$ SD(10,2,3))	Au moment actif
	<corr> 1 : Fournit l'index pour la coordonnée sur laquelle la correction L1 agit. 2 : Fournit l'index pour la coordonnée sur laquelle la correction L2 agit. 3 : Fournit l'index pour la coordonnée sur laquelle la correction L3 agit.

Tab. 7-42: SD(10,...)

SD(11,...) – Index de l'axe du canal des axes principal, secondaire et d'approche

Syntaxe :	Valeur de retour
	Fournit l'index de l'axe du canal des axes principal, secondaire et d'approche en fonction du plan actif :
SD(11,<axe>,1)	Au moment de la préparation du bloc
SD(11,<axe>,2)	Au moment actif (exécution du bloc)
	<axe> 1 : Axe principal 2 : Axe secondaire 3 : Axe d'approche

Tab. 7-43: SD(11,...)

SD(12,...) – Sens de rotation de la broche

Syntaxe :	Valeur de retour
SD(12,1) ($\triangleq$ SD(212,1,1))	Sens de rotation actif de la broche : 0 : arrêt de la broche 3 : rotation à droite de la broche 4 : rotation à gauche de la broche 19 : orientation de la broche -1 : broche non appliquée
SD(12,2) ($\triangleq$ SD(212,2,1))	<i>Dernier sens de rotation programmé de la broche :</i> - Ordres identiques à "Sens de rotation actif de la broche" (SD(12,1)) - Aucune inversion active du sens de rotation via un signal d'interface n'est prise en compte !
SD(12,3) ($\triangleq$ SD(212,1,2))	<i>Sens de rotation actif de la broche (2e broche) :</i> Ordres identiques à "Sens de rotation actif de la broche" (SD(12,1))
SD(12,4) ($\triangleq$ SD(212,2,2))	<i>Dernier sens de rotation programmé de la broche (2e broche) :</i> - Ordres identiques à "Sens de rotation actif de la broche" (SD(12,1)). - Aucune inversion active du sens de rotation via un signal d'interface n'est prise en compte !

Tab. 7-44: SD(12,...)

SD(13) – Type de traitement au moment de l'interprétation

Syntaxe :	Valeur de retour
SD(13)	Types de traitement lors du traitement au moment de l'interprétation : 0 : bloc unique, étape unique 1 : bloc continu 2 : bloc-programme 11 : avance de bloc avec bloc continu (bloc sélectionné non encore interprété)

Tab. 7-45: SD(13)

SD(14) – Langue nationale

Syntaxe :	Valeur de retour
SD(14)	Fournit le numéro de la langue nationale active (paramètre machine 6010 00010).

Tab. 7-46: SD(14)

Ordres CPL

SD(15) – Essai sans mouvement

Syntaxe :	Valeur de retour
SD(15)	<i>Informe que l'essai sans mouvement est actif :</i> 0 = non 1 = oui

Tab. 7-47: SD(15)

SD(17,...) – Index du canal d'une coordonnée avec une signification prédéfinie

Syntaxe :	Valeur de retour								
SD(17,<classe>,1)	Fournit l'index spécifique au canal (1 à 8) de la coordonnée avec la signification <classe> au moment de la préparation des blocs.								
SD(17,<classe>,2)	Fournit l'index spécifique au canal de la coordonnée avec la signification <classe> au moment actif (exécution du bloc).								
	<table> <tr> <td><classe></td><td>Signification des coordonnées :</td></tr> <tr> <td></td><td>1 = "X"</td></tr> <tr> <td></td><td>2 = "Y"</td></tr> <tr> <td></td><td>3 = "Z"</td></tr> </table>	<classe>	Signification des coordonnées :		1 = "X"		2 = "Y"		3 = "Z"
<classe>	Signification des coordonnées :								
	1 = "X"								
	2 = "Y"								
	3 = "Z"								

Tab. 7-48: SD(17,...)

SD(20,...) – Nombre d'axes sur le canal appelant

Syntaxe :	Valeur de retour
	Fournit le nombre d'axes synchrones sur le canal appelant :
SD(20) – ou – SD(20,1)	Au moment de la préparation du bloc
SD(20,2)	Au moment actif (exécution du bloc)

Tab. 7-49: SD(20,...)

SD(21,...) – Nombre d'axes sur un canal

Syntaxe :	Valeur de retour		
	Fournit le nombre d'axes synchrones d'un canal :		
SD(21,<canal>) – ou – SD(21,<canal>,1)	Au moment de la préparation du bloc		
SD(21,<canal>,2)	Au moment actif (exécution du bloc)		
	<table> <tr> <td><canal></td><td>1 à 60 = index du canal</td></tr> </table>	<canal>	1 à 60 = index du canal
<canal>	1 à 60 = index du canal		

Tab. 7-50: SD(21,...)



Si le canal indiqué n'est pas actif, les axes de ce canal peuvent être attribués, ce qui signifie qu'ils peuvent être temporairement actifs sur un autre canal. Cependant, ces axes appartiennent tout de même au canal indiqué.

Exemple : L'axe X2 appartient au canal 2 (inactif) et X2 est actuellement déplacé de manière synchrone sur le canal 1. Dans les instructions SD(21,2,...), l'axe X2 est traité comme appartenant au canal 2.

SD(22,...) – Numéro d'axe du canal d'un axe du système du canal appelant

Syntaxe :	Valeur de retour	
	Fournit le numéro d'axe du canal d'un axe du système du canal appelant ou -1 :	
SD(22,<AxeSys> – ou – SD(22,<AxeSys>,1)	Au moment de la préparation du bloc	
SD(22,<AxeSys>,2)	Au moment actif (exécution du bloc)	
	<AxeSys>	1 à 250 = index d'axe du système – ou – Nom d'axe du système entre guillemets

Tab. 7-51: SD(22,...)

SD(23,...) – Numéro d'axe du système d'un axe du canal appelant

Syntaxe :	Valeur de retour	
	Fournit le numéro d'axe du système d'un axe du canal appelant ou -1 :	
SD(23,<AxeCan> – ou – SD(23,<AxeCan>,1)	Au moment de la préparation du bloc	
SD(23,<AxeCan>,2)	Au moment actif (exécution du bloc)	
	<AxeCan>	1 à 8 = index d'axe du canal – ou – Nom de l'axe du canal entre guillemets.

Tab. 7-52: SD(23,...)

Ordres CPL

SD(24,...) – Numéro d'axe du système d'un axe du canal

Syntaxe :	Valeur de retour	
	Fournit le numéro d'axe du système d'un axe du canal ou -1 :	
SD(24,<Axe-Can>,<canal>) – ou – SD(24,<Axe-Can>,<canal>,1)	Au moment de la préparation du bloc	
SD(24,<Axe-Can>,<canal>,2)	Au moment actif (exécution du bloc)	
	<AxeCan>	1 à 8 = index d'axe du canal – ou – Nom de l'axe du canal entre guillemets.
	<canal>	1 à 60 : Index du canal

Tab. 7-53: SD(24,...)



Si le canal indiqué n'est pas actif, les axes de ce canal peuvent être attribués, ce qui signifie qu'ils peuvent être temporairement actifs sur un autre canal. Cependant, ces axes appartiennent tout de même au canal indiqué.

Exemple : L'axe X2 appartient au canal 2 (inactif) et X2 est actuellement déplacé de manière synchrone sur le canal 1. Dans les instructions SD(24,2,...), l'axe X2 est traité comme appartenant au canal 2.

SD(25,...) – Canal d'un axe du système

Syntaxe :	Valeur de retour	
SD(25,<AxeSys>)	Fournit le canal d'un axe du système au moment actif. La valeur "-1" est renvoyée pour les axes asynchrones.	
	<AxeSys>	1 à 250 = index d'axe du système – ou – Nom d'axe du système entre guillemets

Tab. 7-54: SD(25,...)

SD(30) – Événement déclencheur de l'appel du programme d'initialisation

Syntaxe :	Valeur de retour
SD(30)	Fournit, sur chaque canal de traitement, l'événement déclencheur de l'appel du programme d'initialisation spécifique au canal : 1 : à la fin d'un programme principal (M30) 2 : après la remise à zéro du canal 3 : après la remise à zéro du système 0 : appel non pas en tant que programme d'initialisation, mais en tant que programme CN normal SD(30) est prévu pour être utilisé dans des programmes d'initialisation.

Tab. 7-55: SD(30)

SD(60,...) – Données de la spline

Syntaxe :	Valeur de retour
SD(60,1,<canal>,<TypeBloc>)	ID spline ayant été programmé avec SplineDef(<IdSpline>) Voir aussi SDR(60) Valeur >= 0 : pas d'erreur Valeur = -1 : TypeBloc erroné Valeur = -2 : échec lors de la détermination du bloc CN
	<canal> Optionnel : Index de l'axe du canal – ou – -1 pour le canal propre Valeur par défaut : -1
	<TypeBloc> Optionnel : Sélection de bloc CN : 1 : préparation du bloc 2 : exécution du bloc -1 : bloc de reconfiguration -2 : bloc d'interruption Valeur par défaut : 1 pour <canal> = canal propre 2 pour <canal> = canal tiers

Tab. 7-56: SD(60,...)



TypeBloc = 1 ne doit pas être utilisé pour le canal tiers. La valeur de retour est alors -1.

TypeBloc = -1, -2 ne fournissent des valeurs de retour judicieuses que pour l'effacement de la course restante, l'avance de bloc et ASUP. Sinon, la valeur fournie est -2.

Ordres CPL

SD(68,...) – Valeur de la somme du décalage des coordonnées du programme

Syntaxe :	Valeur de retour	
	Fournit la valeur de la somme (Trans + ATrans) du décalage des coordonnées du programme pour une coordonnée du canal.	
SD(68,1,<IndexCoord>)	Au moment de la préparation du bloc (dernière valeur programmée)	
SD(68,2,<IndexCoord>)	Au moment actif (exécution du bloc)	
	<IndexCoord>	1 à 8 = index de coordonnée

Tab. 7-57: SD(68,...)

SD(77) – Point de redémarrage du sous-programme asynchrone

Syntaxe :	Valeur de retour
SD(77)	<i>Fournit le point d'approche actuellement réglé d'un sous-programme asynchrone :</i> • 1 = point de départ • 2 = point d'arrivée • 3 = point d'interruption

Tab. 7-58: SD(77)

SD(131,...) – Guidage tangentiel de l'outil

Syntaxe :	Valeur de retour
SD(131,1)	Fournit le numéro d'axe du canal de l'axe de rotation.
SD(131,2)	Fournit la symétrie de l'outil.

Tab. 7-59: SD(131,...)

SD(168,...) – Décalage des coordonnées du programme

Syntaxe :	Valeur de retour	
	Fournit la valeur du décalage des coordonnées du programme (Trans) pour une coordonnée du canal.	
SD(168,1,<IndexCoord>)	Au moment de la préparation du bloc (dernière valeur programmée)	
SD(168,2,<IndexCoord>)	Au moment actif (exécution du bloc)	
	<IndexCoord>	1 à 8 = index de coordonnée

Tab. 7-60: SD(168,...)

SD(200,...) – Zones de travail et zones mortes

Syntaxe :	Valeur de retour
SD(200,1)	Fournit le nombre de zones actives sur le canal.
SD(200,1,<zone>)	<i>État de la zone sur le canal</i> • 1 : La zone est active sur le canal. • 0 : La zone est inactive sur le canal.

Syntaxe :	Valeur de retour	
SD(200,3,<zone>)	<i>Type de zone :</i> 0 : non défini 1 : zone morte 2 : zone de travail	
SD(200,11,<zone>)	Centre de la zone en unités de programmation (1er axe de la zone)	
SD(200,12,<zone>)	Centre de la zone en unités de programmation (2e axe de la zone)	
SD(200,21,<zone>)	Extension de la zone en unités de programmation (1er axe de la zone)	
SD(200,22,<zone>)	Extension de la zone en unités de programmation (2e axe de la zone)	
SD(200,31,<zone>)	Numéro de l'axe du système (1er axe de la zone)	
SD(200,32,<zone>)	Numéro de l'axe du système (2e axe de la zone)	
	<zone>	1 à 32 = index de la zone

Tab. 7-61: SD(200,...)

SD(202,...) – Override de la broche

Syntaxe :	Valeur de retour	
SD(202,<BrocheCan>)	Fournit la valeur Override active d'une broche du canal.	
	<BrocheCan>	1 à 8 = numéro de la broche du canal

Tab. 7-62: SD(202,...)

SD(205,...) – Vitesse de rotation d'une broche du canal

Syntaxe :	Valeur de retour	
SD(205,1,<BrocheCan>)	Fournit la vitesse de rotation de consigne active, y compris l'Override.	
SD(205,2,<BrocheCan>)	Fournit la dernière vitesse de rotation programmée.	
SD(205,3,<BrocheCan>)	Fournit la vitesse de rotation réelle actuelle.	
	<BrocheCan>	1 à 8 = numéro de la broche du canal

Tab. 7-63: SD(205,...)

SD(211,...) – Numéros des broches du canal d'un groupe de broches

Syntaxe :	Valeur de retour	
	Fournit la configuration actuelle d'un groupe de broches au moment de la préparation des blocs en tant que liste compressée.	
SD(211,<groupe>,0)	Fournit le nombre de membres du groupe (0 à 8).	

Ordres CPL

Syntaxe :	Valeur de retour	
SD(211,<groupe>,<N>)	Fournit le nombre de broches du canal du Nième membre du groupe.	
	<groupe>	1 à 4 = numéro de groupe
	<N>	1 à 8 = numéro de membre

Tab. 7-64: SD(211,...)

SD(212,...) – Fonction de mouvement d'une broche du canal

Syntaxe :	Valeur de retour	
	Fournit la fonction de mouvement d'une broche du canal.	
SD(212,1,<BrocheCan>)	Au moment de la préparation du bloc	
SD(212,2,<BrocheCan>)	Au moment actif	
	<BrocheCan>	1 à 8 = numéro de la broche du canal

Tab. 7-65: SD(212,...)

SD(222,...) – Numéro de broche du canal d'une broche du système

Syntaxe :	Valeur de retour	
	Fournit le numéro de broche du canal d'une broche du système sur le canal actif ou -1.	
SD(222,<BrocheSys>) – ou – SD(222,<BrocheSys>,1)	Valeur au moment de la préparation du bloc	
SD(222,<BrocheSys>,2)	Valeur au moment actif	
	<BrocheSys>	Nom de broche du système ou 1 à 60 = numéro de broche du système

Tab. 7-66: SD(222,...)

SD(223,...) – Numéro de broche du système d'une broche du canal

Syntaxe :	Valeur de retour	
	Fournit le numéro de broche du système d'une broche du canal sur le canal actif ou -1.	
SD(223,<BrocheCan>) – ou – SD(223,<BrocheCan>,1)	Valeur au moment de la préparation du bloc	
SD(223,<BrocheCan>,2)	Valeur au moment actif	
	<BrocheCan>	"S1" à "S8" ou 1 à 8 = numéro de broche du canal

Tab. 7-67: SD(223,...)

SD(230,...) – Numéro d'axe du système d'une broche du système

Syntaxe :	Valeur de retour	
SD(230,<BrocheSys>)	Fournit le numéro d'axe du système d'une broche du système ou -1.	
	<BrocheSys>	Nom de broche du système ou 1 à 60 = numéro de broche du système

Tab. 7-68: SD(230,...)

SD(231,...) – Numéro d'axe du système d'une broche du canal

Syntaxe :	Valeur de retour	
SD(231,<BrocheCan>)	Fournit le numéro d'axe du système d'une broche du canal ou -1.	
	<BrocheCan>	"S1" à "S8" ou 1 à 8 = numéro de broche du canal

Tab. 7-69: SD(231,...)

SD(232,...) – Numéro de broche du système d'un axe du système

Syntaxe :	Valeur de retour	
SD(232,<AxeSys>)	Fournit le numéro de broche du système d'un axe du système ou -1.	
	<AxeSys>	1 à 250 = index d'axe du système – ou – Nom d'axe du système entre guillemets

Tab. 7-70: SD(232,...)

SD(268,...) – Décalage additif des coordonnées du programme

Syntaxe :	Valeur de retour	
	Fournit la valeur du décalage additif des coordonnées du programme (ATrans) pour une coordonnée du canal.	
SD(268,1,<IndexCoord>)	Au moment de la préparation du bloc (dernière valeur programmée)	
SD(268,2,<IndexCoord>)	Au moment actif (exécution du bloc)	
	<IndexCoord>	1 à 8 = index de coordonnée

Tab. 7-71: SD(268,...)

Ordres CPL

SD(328,...) – Programmation de précision

Syntaxe :	Valeur de retour
SD(328,1) – ou – SD(328)	Dernière barrière de précision programmée de la fonction CN "PrecProg"
SD(328,2)	Dernière distance des angles programmée de la fonction CN "Prec-Prog"

Tab. 7-72: SD(328,...)

SD(581,...) – Couplage d'axes du canal

Syntaxe :	Valeur de retour				
SD(581,0,<maître>)	- Numéro d'axe du canal (<maître>) si <maître> est un axe maître. 0 : Si <maître> n'est pas un axe maître.				
SD(581,<esclave>,0)	- Numéro d'axe du canal de l'axe maître dont l'axe <esclave> est esclave. 0 : Si <esclave> n'est pas un axe esclave.				
SD(581,<esclave>,1)	- Décalage programmé de l'axe esclave, en unités de programmation, pour l'axe <esclave> 0 : Si <esclave> n'est pas un axe esclave.				
SD(581,<esclave>,2)	- Facteur de couplage programmé pour l'axe <esclave> 0 : Si <esclave> n'est pas un axe esclave.				
SD(581,<esclave>,3)	- Décalage programmé de l'axe maître, en unités de programmation, pour l'axe maître de l'axe <esclave> 0 : Si <esclave> n'est pas un axe esclave.				
	<table> <tr> <td><maître></td><td>Index d'axe du canal de l'axe maître (1 à 8)</td></tr> <tr> <td><esclave></td><td>Index d'axe du canal de l'axe esclave (1 à 8)</td></tr> </table>	<maître>	Index d'axe du canal de l'axe maître (1 à 8)	<esclave>	Index d'axe du canal de l'axe esclave (1 à 8)
<maître>	Index d'axe du canal de l'axe maître (1 à 8)				
<esclave>	Index d'axe du canal de l'axe esclave (1 à 8)				

Tab. 7-73: SD(581,...)

SD(594,...) – Groupe d'avance

Syntaxe :	Valeur de retour
	Fournit des informations sur les groupes d'avance (FeedGrp) et l'ajustement de l'avance (FeedAd).
SD(594,<coord>,0)	FeedGrp et FeedAd : 2 : La coordonnée est membre du groupe d'avance actuel. 1 : La coordonnée est entraînée. 0 : La coordonnée est un esclave (par ex. d'AXC). -1 : La coordonnée n'existe pas sur le canal. Par ex. SD(594,"Y",0)

Syntaxe :	Valeur de retour	
SD(594,<coord>,1)	FeedGrp : 1 : FeedGrp est actif et la coordonnée est un membre programmé. 0 : La coordonnée n'est pas membre ou FeedGrp est inactif. -1 : La coordonnée n'existe pas sur le canal. Par ex. SD(594,"Y",1)	
SD(594,<coord>,2)	FeedAd : 2 : FeedAd est actif et la coordonnée est un membre programmé. 1 : FeedAd est actif et la coordonnée est membre via MP. 0 : FeedAd est inactif ou la coordonnée n'est pas membre. -1 : La coordonnée n'existe pas sur le canal. Par ex. SD(594,"Y",2)	
	<coord>	Index d'axe du canal (1 à 8) – ou – Nom de coordonnée – ou – 9 (= coordonnée d'orientation O)

Tab. 7-74: SD(594,...)

SD – Exemple de programmation

Exemples :

```

30 A% = SD(2,1)      A% contient la position active du potentiomètre d'avance
:                    en pourcentage.

40 B% = SD(5,1,1)    B% contient la vitesse d'avance active.
:
```

Exemple de programme :

SD (9)

```

N4 G75.2 X120
60 IF SD(9) = 1 THEN
N7 (MSG, Palpeur de mesure 2 non déclenché.)
80 GOTO .ERREUR
90 ELSE
100 XMESS = PPOS(1)
110 ENDIF
```

Dans l'exemple du programme "SD (9)", l'axe X est déplacé en direction de la position indiquée. Si la position est atteinte sans que le palpeur de mesure n'ait été déclenché, un message (ligne N7) est généré et un saut vers le label .ERREUR est effectué. Par contre, si le palpeur de mesure a été déclenché, la position actuelle par rapport au système de coordonnées du programme peut être enregistrée dans XMESS.

Ordres CPL

SDR (description générale)

	REAL	READ	WRITE	FUNCT	PROC	WAIT	ERRNO	!
Description :	System Data Real – Données internes au système au format REAL							
	Lit dans une collection de données système de la commande IndraMotion MTX au format REAL . Le format de l'ordre et son application correspondent à l'ordre SD.							
Syntaxe :	SDR (<groupe>[,<index1>[,<index2>]])							
	L'ordre SDR fournit des valeurs REAL.							

Tab. 7-75: Syntaxe SDR

SDR(1,...) – Position cible après l'avance de bloc

Syntaxe :	Valeur de retour
SDR(1,<Index-Coord>)	Positions de la pièce à usiner de toutes les coordonnées qui ont été calculées pour l'avance de bloc/la réentrée. Si aucune avance de bloc n'est programmée, la valeur renvoyée est "0". Si un axe ou une coordonnée non paramétrés ou un axe auxiliaire sont adressés, une erreur d'exécution est générée.
	<IndexCoord> 1 à 8 = index de coordonnée

Tab. 7-76: SDR(1,...)

SDR(2,...) – Position actuelle de l'Override

Syntaxe :	Valeur de retour
SDR(2,1)	Fournit la position actuelle de l'avance Override sur le canal actuel, en pourcentage.
SDR(2,3)	Fournit la position actuelle de l'Override de la broche 1, en pourcentage. 0, si aucune broche n'est paramétrée.
SDR(2,4)	Fournit la position actuelle de l'Override de la broche 2, en pourcentage. 0, si aucune broche n'est paramétrée.

Tab. 7-77: SDR(2,...)

SDR(5,...) – Vitesse

Syntaxe :	Valeur de retour
SDR(5,1,1)	Fournit la vitesse d'avance active (valeur 100 %) arrondie à une valeur entière, en mm/min ou en pouces/min.
SDR(5,2,1)	Fournit la vitesse d'avance rapide active (valeur 100 %) arrondie à une valeur entière, en mm/min ou en pouces/min.
SDR(5,1,2)	Fournit la dernière avance programmée, en unité de saisie par minute.

Tab. 7-78: SDR(5,...)

SDR(60,...) – Données de la spline

Syntaxe :	Valeur de retour	
SDR(60,2,<canal>)	Tolérance de l'approximation spline B ayant été programmée avec BAP(E..)	
SDR(60,3,<canal>)	Tolérance d'orientation de l'approximation spline B ayant été programmée avec BAP(OE..)	
	La valeur de retour se rapporte au bloc de préparation sur le canal propre ou au bloc actif sur le canal tiers. Voir aussi SD(60,1) Valeur >= 0 : pas d'erreur Valeur = -2.0 : échec lors de la détermination du bloc CN	
	<canal>	Optionnel : Index de l'axe du canal – ou – -1 pour le canal propre Valeur par défaut : -1

Tab. 7-79: SDR(60,...)

SDR(68,...) – Valeur de la somme du décalage des coordonnées du programme

Syntaxe :	Valeur de retour	
	Fournit la valeur de la somme (Trans + Atrans) du décalage des coordonnées du programme pour une coordonnée du canal.	
SDR(68,1,<Index-Coord>)	Au moment de la préparation du bloc (dernière valeur programmée)	
SDR(68,2,<Index-Coord>)	Au moment actif (exécution du bloc)	
	<IndexCoord>	1 à 8 = index de coordonnée

Tab. 7-80: SDR(68,...)

SDR(77,...) – Position de redémarrage après ASUP

Syntaxe :	Valeur de retour	
	Fournit les coordonnées des points d'approche possibles après un sous-programme asynchrone : La position fournie permet, au sein d'un sous-programme asynchrone, de repositionner les coordonnées à proximité du point d'approche voulu à l'aide de "G76".	
SDR(77,<CoordIndex>,0) – ou – SDR(77)	Fournit le point d'approche actuel selon ASPRTP / REPOSTP d'une coordonnée de machine dans le bloc d'interruption.	
SDR(77,<CoordIndex>,1)	Fournit le point de départ d'une coordonnée de machine dans le bloc d'interruption.	

Ordres CPL

Syntaxe :	Valeur de retour	
SDR(77,<CoordIndex>,2)	Fournit le point d'arrivée d'une coordonnée de machine dans le bloc d'interruption.	
SDR(77,<CoordIndex>,3)	Fournit le point d'interruption d'une coordonnée de machine dans le bloc d'interruption.	
	<IndexCoord>	Index de coordonnée de machine

Tab. 7-81: SDR(77,...)

SDR(131,...) – Guidage tangentiel de l'outil

Syntaxe :	Valeur de retour
SDR(131,3)	Fournit l'angle d'attaque en degrés.
SDR(131,4)	Fournit l'angle de bloc intermédiaire en degrés.

Tab. 7-82: SDR(131,...)

SDR(168,...) – Décalage des coordonnées du programme

Syntaxe :	Valeur de retour	
	Fournit la valeur du décalage des coordonnées du programme (Trans) pour une coordonnée du canal.	
SDR(168,1,<IndexCoord>)	Au moment de la préparation du bloc (dernière valeur programmée)	
SDR(168,2,<IndexCoord>)	Au moment actif (exécution du bloc)	
	<IndexCoord>	1 à 8 = index de coordonnée

Tab. 7-83: SDR(168,...)

SDR(202,...) – Override de la broche

Syntaxe :	Valeur de retour	
SDR(202,<BrocheCan>)	Fournit la valeur Override active d'une broche du canal.	
	<BrocheCan>	1 à 8 = numéro de la broche du canal

Tab. 7-84: SDR(202,...)

SDR(205,...) – Vitesse de rotation d'une broche du canal

Syntaxe :	Valeur de retour	
SDR(205,1,<BrocheCan>)	Fournit la vitesse de rotation de consigne active, y compris l'Override.	
SDR(205,2,<BrocheCan>)	Fournit la dernière vitesse de rotation programmée.	
SDR(205,3,<BrocheCan>)	Fournit la vitesse de rotation réelle actuelle.	
	<BrocheCan>	1 à 8 = numéro de la broche du canal

Tab. 7-85: SDR(205,...)

SDR(219,...) – Position d'orientation d'une broche du canal

Syntaxe :	Valeur de retour	
SDR(219,<Broche-Can>)	Dernière position de broche programmée sur M19 (si M19 n'est pas programmé, la valeur renvoyée est 0)	
	<BrocheCan>	1 à 8 = numéro de la broche du canal

Tab. 7-86: SDR(219,...)

SDR(268,...) – Décalage additif des coordonnées du programme

Syntaxe :	Valeur de retour	
	Fournit la valeur du décalage additif des coordonnées du programme (ATrans) pour une coordonnée du canal.	
SDR(268,1,<Index-Coord>)	Au moment de la préparation du bloc (dernière valeur programmée)	
SDR(268,2,<Index-Coord>)	Au moment actif (exécution du bloc)	
	<IndexCoord>	1 à 8 = index de coordonnée

Tab. 7-87: SDR(268,...)

SDR(328,...) – Programmation de précision

Syntaxe :	Valeur de retour
SDR(328,1) – ou – SDR(328)	Dernière barrière de précision programmée de la fonction CN "PrecProg"
SDR(328,2)	Dernière distance des angles programmée de la fonction CN "Prec-Prog"

Tab. 7-88: SDR(328,...)

SDR(581,...) – Couplage d'axes du canal

Syntaxe :	Valeur de retour
SDR(581,0,<maître>)	- Numéro d'axe du canal (<maître>) si <maître> est un axe maître. 0 : Si <maître> n'est pas un axe maître.
SDR(581,<esclave>,0)	- Numéro d'axe du canal de l'axe maître dont l'axe <esclave> est esclave. 0 : Si <esclave> n'est pas un axe esclave.
SDR(581,<esclave>,1)	- Décalage programmé de l'axe esclave, en unités de programmation, pour l'axe <esclave> 0 : Si <esclave> n'est pas un axe esclave.
SDR(581,<esclave>,2)	- Facteur de couplage programmé pour l'axe <esclave> 0 : Si <esclave> n'est pas un axe esclave.

Ordres CPL

Syntaxe :	Valeur de retour	
SDR(581,<esclave>,3)	- Décalage programmé de l'axe maître, en unités de programmation, pour l'axe maître de l'axe <esclave> 0 : Si <esclave> n'est pas un axe esclave.	
	<maître>	Index d'axe du canal de l'axe maître (1 à 8)
	<esclave>	Index d'axe du canal de l'axe esclave (1 à 8)

Tab. 7-89: SDR(581,...)

7.4.5 Données système de types structurés

Généralités

Les données système de types structurés (abréviation : SD) constituent une catégorie de données dans le système global qui présente les caractéristiques suivantes :

- Les données système individuelles sont enregistrées soit dans la mémoire volatile, soit dans la mémoire non-volatile (données système permanentes).
- Le nombre, la taille et la structure peuvent être définis librement dans les limites de la mémoire existante.

Les données système permanentes sont initialisées une seule fois ; les données système volatiles sont initialisées lors de chaque montée en régime. Les valeurs peuvent être enregistrées dans un fichier d'initialisation. En l'absence de définition dans ce fichier, elles sont remises à "0".

Les variables structurées "SV" peuvent être remplacées par les données système (SD), mais elles sont tout de même conservées pour des raisons de compatibilité.

Vous trouverez de plus amples informations sur la mise en œuvre et la définition des données système dans la documentation "Description des fonctions Rexroth IndraMotion MTX".

Accès aux données système

Description : Au langage CPL, les données système sont accessibles avec le mot-clé SD. Contrairement à l'adressage XPath (accès standardisé aux données structurées), l'opérateur point "." est utilisé à la place du séparateur "/". Les tableaux sont adressés avec [], par exemple :

Xpath : /MyArrVar[2,3,4]/Sous-composant

CPL : SD.MyArrVar[2,3,4].Sous-composant

Exemple : 10 SD.MyChanVar=DBSEA("/dbt1/Rec",-1,-1,"Key1=1",I%)

Lorsqu'il s'agit de données système spécifiques au canal, l'index du canal peut être omis. Par exemple :

SD.MyArrVar[,3,4] – ou –

SD.MyChanVar sans [] si la SD du canal est unidimensionnelle.

Des expressions CPL fournissant une valeur entière (INTEGER) peuvent être programmées librement entre les crochets.

Des données système ayant des types de structures identiques peuvent être attribuées réciproquement.

Les règles d'attribution suivantes s'appliquent aux types de base :

Nom	Longueur en octets	Description	Type C	Type IndraLogic	CPL
String	2 * LongueurMax + 1	Chaîne de caractères (format UTF-8)	char []	String()	CHAR
IsoLatin1String	LongueurMax + 1	Chaîne de caractères (format Latin1)	char []	String()	CHAR
Byte_t	1	Nombre entier de 8 bits signé	char	SINT	INT
Short_t	2	Nombre entier de 16 bits signé	short	INT	INT
Int_t	4	Nombre entier de 32 bits signé	int	DINT	INT
UnsignedByte_t	1	Nombre entier de 8 bits non signé	unsigned char	USINT	INT
UnsignedShort_t	2	Nombre entier de 16 bits non signé	unsigned short	UINT	INT
UnsignedInt_t	4	Nombre entier de 32 bits non signé	unsigned int	UDINT	INT
Float_t	4	Nombre réel de 32 bits	float	REAL	REAL
Double_t	8	Nombre réel de 64 bits	double	LREAL	DOUBLE
Boolean_t	1	vrai, faux, 1, 0	char	BOOL	BOOL

Tab. 7-90: Règles d'attribution pour les types de base

VARINF

INT	READ	WRITE	FUNCT	PROC	WAIT	ERRNO	!

Description : **VARiable INformation** – Informations relatives aux variables

L'ordre **VARINF** permet d'envoyer une dans le programme pièce pour savoir s'il existe une donnée système structurée ou une variable permanente. Le résultat fourni par l'ordre est 1 si la variable indiquée existe. Si la variable indiquée n'existe pas, l'ordre fournit le résultat 0.

Syntaxe :

VARINF(<nom de variable>)	
avec	
<nom de variable>	Nom de la donnée système structurée ou de la variable permanente

Tab. 7-91: Syntaxe VARINF

Exemple :

```
10 DIM A$(20)
10 I% = VARINF("@_RES_DOUBLE")
...
40 A$ = "SD.SysSRun[1].Active"
50 J% = VARINF(A$)
...
```

VARVAL

MULTYP	READ	WRITE	FUNCT	PROC	WAIT	ERRNO	!

Ordres CPL

Description : **VARiable VALue** – Valeur de variables

L'ordre CPL renvoie le contenu d'une variable de données système ou d'une variable permanente. Le nom de la variable est transmise en tant qu'expression de chaîne de caractères. Il est ainsi possible d'adresser la variable indirectement. Par ailleurs, il n'est pas nécessaire que la variable existe pour lier le programme.

Si la variable indiquée n'existe pas ou si son type n'est pas valide, alors une erreur d'exécution est générée. Pour vérifier au préalable s'il existe une variable de données système ou une variable permanente, il faut utiliser l'ordre VARINF (voir [chap. "VARINF" à la page 511](#)).

L'ordre VARVAL prend en charge le paramètre ERRNO.

Syntaxe :

VARVAL(<nom de variable>)	
avec	
<nom de variable>	Nom de la donnée système structurée ou de la variable permanente
ERRNO	Variable CPL Si la variable CPL ERRNO est indiquée à un emplacement quelconque dans la liste de paramètres, aucune erreur d'exécution n'est générée. L'erreur est fournie par une valeur correspondante de la variable. <i>Les valeurs de retour suivantes sont possibles :</i> • 0 : Accès OK • < 0 : Erreur (pour une description précise de l'erreur, voir chap. 3.14.5 "Variable ERRNO pour l'analyse des erreurs des fonctions CPL ERRNO" à la page 85) Si la variable CPL ERRNO n'est pas indiquée, une erreur d'exécution est générée en cas d'erreur d'accès.

Tab. 7-92: Syntaxe VARVAL

Exemple : *Programme:*

```
1 DIM A$(30) : DIM B$(30) : IDX%=4
1 A$="SD.MySDVar[" : B$="].Active"
1 RES% = VARVAL(A$+STR$(IDX%)+B$)
; le contenu de SD.MySDVar[4].Active se trouve dans RES%
```

- Restrictions :**
- Seules les variables de données système et permanentes sont prises en charge (pas les variables CPL globales ou locales).
 - La chaîne de caractères transmise est le nom exact de la variable et non pas une expression devant être calculée.
 - Pour accéder à une variable de tableau, tous les indices doivent être indiqués en tant que constantes (chiffres) et non pas en tant qu'expressions.
 - Seuls des types simples (INTEGER, REAL, DOUBLE, BOOLEAN, STRING) peuvent être lus (et non pas des champs ou des structures).

SDLOAD

---	READ	WRITE	FUNCT	PROC	WAIT	ERRNO	!

Description : **System Data LOAD** – Chargement de données système structurées

L'ordre CPL SDLOAD sert à extraire les données système d'un fichier. Il est possible d'indiquer plusieurs éléments racine. Les sous-composants indiqués

sont ignorés, toutes les données sont toujours lues à partir de l'élément racine.

Les éléments du tableau racine peuvent être lus intégralement (sans index de champ) ou individuellement (avec index de champ).

Syntaxe :

SDLOAD (<NomFichier>,<ÉlémentRacine_1>[,...,<ÉlémentRacine_n>])	
<NomFichier>	Chemin du répertoire et nom du fichier cible Si aucun chemin n'est indiqué, le fichier est lu à partir du répertoire actuel.
<ÉlémentRacine>	Élément racine du groupe de données devant être lu
ERRNO	Variable CPL Si la variable CPL ERRNO est indiquée à un emplacement quelconque dans la liste de paramètres, aucune erreur d'exécution n'est générée. L'erreur est fournie par une valeur correspondante de la variable. Les valeurs de retour suivantes sont possibles : 0 : Accès OK -1 : Erreur de paramètre -7 : Échec de la lecture des données -9 : Impossible d'accéder au fichier source -10 : Fichier cible, y compris chemin, trop long -20 : Balise racine (root tag) erronée -21 : Impossible d'accéder au fichier -35 : Erreur interne lors de l'accès aux données système Si la variable CPL ERRNO n'est pas indiquée, une erreur d'exécution est générée en cas d'erreur d'accès.

Tab. 7-93: Syntaxe SDLOAD

Exemple de syntaxe :

```
SDLOAD ("/usrfep/SDDatUsr.xml", "DgnUsr[1]", ERRNO)
```

La donnée système spécifique au client "DgnUsr[1]" est lue à partir du fichier "SDDatUsr.xml" dans "/usrfep" (FEPROM utilisateur).

SDSAVE

---	READ	WRITE	FUNCT	PROC	WAIT	ERRNO	!

Description :

System Data SAVE – Sauvegarde des données système structurées

L'ordre CPL SDSAVE permet d'enregistrer les données système dans un **nouveau** fichier ; les données enregistrées dans un fichier de même nom pré-existant sont écrasées. Il est possible d'indiquer plusieurs éléments racine. Les sous-composants indiqués sont ignorés ; toutes les données sont toujours sauvegardées à partir de l'élément racine.

Les éléments du tableau racine peuvent être enregistrés intégralement (sans index de champ) ou individuellement (avec index de champ).

Ordres CPL

Syntaxe :

SDSAVE (<NomFichier>, <ÉlémentRacine_1>[, ..., <ÉlémentRacine_n>])	
<NomFichier>	Chemin du répertoire et nom du fichier cible Si aucun chemin n'est indiqué, le fichier est enregistré dans le répertoire actuel.
<ÉlémentRacine>	Élément racine du groupe de données à sauvegarder
ERRNO	Variable CPL Si la variable CPL ERRNO est indiquée à un emplacement quelconque dans la liste de paramètres, aucune erreur d'exécution n'est générée. L'erreur est fournie par une valeur correspondante de la variable. Les valeurs de retour suivantes sont possibles : 0 : Accès OK -1 : Erreur de paramètre -10 : Fichier cible, y compris chemin, trop long -13 : Échec lors de l'écriture des données -20 : Balise racine (root tag) erronée -21 : Impossible d'accéder au fichier -35 : Erreur interne lors de l'accès aux données système Si la variable CPL ERRNO n'est pas indiquée, une erreur d'exécution est générée en cas d'erreur d'accès.

Tab. 7-94: Syntaxe SDSAVE

Exemple de syntaxe :

```
10 SDSAVE("cplsave.xml","SysAPrg",ERRNO)
```

L'élément racine complet "SysAPrg" est enregistré dans le fichier "cplsave.xml" dans le répertoire actuel.

```
20 SDSAVE("SDDatUsr.xml", "DgnUsr[1]", ERRNO)
```

L'élément individuel "DgnUsr[1]" du tableau racine est enregistré dans le fichier "SDDatUsr.xml" dans le répertoire actuel.

```
30 SDSAVE("/SDDatUsr.xml", "DqnUsr", ERRNO)
```

Tous les éléments du tableau racine "DgnUsr" sont enregistrés dans le fichier "SDDatUsr.xml" dans le répertoire racine.

7.4.6 Interface Sercos

SCS3

INT	READ	WRITE	FUNCT	PROC	WAIT	ERRNO	!

Description : SerCoS 3 – Lecture de paramètres Sercos

Permet d'accéder aux paramètres d'entraînement Sercos avec un numéro d'identification élargi pour les lire.

Syntaxe :

SCS3 (<sélection d'axe>,<n° EID>)	
<sélection d'axe>	Index d'axe du système ou nom d'axe du système

Ordres CPL

<n° EID>	Numéro d'identification Sercos élargi (Sxxxxx.xx.xx, Pxxxxx.xx.xx)
ERRNO	Variable CPL Si la variable CPL ERRNO est indiquée à un emplacement quelconque dans la liste de paramètres, aucune erreur d'exécution n'est générée. L'erreur est fournie par une valeur correspondante de la variable. Les valeurs de retour suivantes sont possibles : 0 : Accès OK -1 : Erreur de paramètre -7 : Échec de la lecture des données -10 : Fichier cible, y compris chemin, trop long -21 : Impossible d'accéder au fichier -27 : Ordre avec type de paramètre non valide -30 : Espace mémoire insuffisant -36 : Canal de service bloqué -37 : Numéro d'identification inconnu -38 : Numéro d'identification non valide -39 : Numéro d'axe non valide Si la variable CPL ERRNO n'est pas indiquée, une erreur d'exécution est générée en cas d'erreur d'accès.

Tab. 7-95: Syntaxe SCS3

Le contenu du paramètre est fourni sans unité ni calibrage.



L'ordre est uniquement disponible pour les entraînements Sercos avec un numéro d'identification élargi.



Les paramètres contenant une liste (plusieurs valeurs séparées par des virgules) ne peuvent pas être lus. Dans ce cas, la commande génère un message d'erreur.



La transmission de données peut être accélérée en utilisant le canal multiplex Sercos.

Si les données d'entraînement se trouvent dans le télégramme d'entraînement Sercos, elles sont lues à partir de ce dernier (voir le manuel des paramètres de l'entraînement Sercos). Dans tous les autres cas, les données d'entraînement sont lues directement dans l'entraînement.

Si d'autres applications accèdent aux données d'entraînement, ces données ne sont pas accessibles à cet instant. Dans ce cas d'erreur, il est possible de réagir avec le paramètre ERRNO dans le programme pièce. Un nouvel accès peut fournir la donnée d'entraînement voulue.



Un accès permanent aux données d'entraînement peut bloquer l'accès pour d'autres applications !

L'évaluation de la variable d'erreur ERRNO permet de réagir au cas d'erreur dans le programme pièce.

Ordres CPL

Exemple :

```
10 POSITION%=SCS3(1,"S-0-51.0.0",ERRNO)

    La position réelle de la position du 1er axe est attribuée à la
    variable INTEGER POSITION%.

12 IF ERRNO = 0 THEN

    Analyse des erreurs

13 REM***La valeur de position réelle n'a pas pu être lue cor-
    rectement***

14 ELSE

15 REM***La valeur de position réelle n'a pas pu être lue***

16 ENDIF
```

SCS3L

---	READ	WRITE	FUNCT	PROC	WAIT	ERRNO	!

Description :

SerCoS 3 List – Lecture de listes de paramètres Sercos

Certains paramètres Sercos sont disponibles dans l'entraînement en tant que listes pouvant être lues à l'aide de l'ordre "SCS3L". Étant donné que la longueur (mémoire requise) d'une liste est inconnue, les éléments de la liste lus sont enregistrés dans des fichiers ASCII. Les données lues peuvent ensuite être exploitées à l'aide d'ordres de fichier CPL.

L'ordre SCS3L permet de créer le fichier indiqué dans l'ordre s'il n'existe pas encore. Si le fichier indiqué existe déjà, son contenu est écrasé.

Syntaxe :

SCS3L (<index d'axe>,<n° EID>,<nom de fichier>]	
<index d'axe>	Index d'axe du système ou nom d'axe du système
<n° EID>	Numéro d'identification Sercos élargi (Sxxxxx.xx.xx, Pxxxxx.xx.xx)

Ordres CPL

<nom de fichier>	Nom du fichier ASCII dans lequel la liste lue doit être enregistrée.
ERRNO	Variable CPL Si la variable CPL ERRNO est indiquée à un emplacement quelconque dans la liste de paramètres, aucune erreur d'exécution n'est générée. L'erreur est fournie par une valeur correspondante de la variable. Les valeurs de retour suivantes sont possibles : 0 : Accès OK -1 : Erreur de paramètre -7 : Échec de la lecture des données -10 : Fichier cible, y compris chemin, trop long -21 : Impossible d'accéder au fichier -27 : Ordre avec type de paramètre non valide -30 : Espace mémoire insuffisant -36 : Canal de service bloqué -37 : Numéro d'identification inconnu -38 : Numéro d'identification non valide -39 : Numéro d'axe non valide Si la variable CPL ERRNO n'est pas indiquée, une erreur d'exécution est générée en cas d'erreur d'accès.

Tab. 7-96: Syntaxe SCS3L

Il se peut que les données d'entraînement soient inaccessibles à certains moments si d'autres applications accèdent à ces données. Le paramètre <ERRNO> permet de réagir à ce cas d'erreur dans le programme pièce. Un nouvel accès peut fournir la donnée d'entraînement voulue.



L'ordre est uniquement disponible pour les entraînements Sercos qui ont un numéro d'identification élargi.



Un accès permanent aux données d'entraînement risque de bloquer l'accès pour d'autres applications.

Exemple :

```
10 SCS3L(1, "P-0-0021", "OUTFILE.txt")
```

Écrire la liste des numéros d'identification pour l'oscilloscope d'axe de l'entraînement 1 dans le fichier "OUTFILE.txt"

WSCS

---	READ	WRITE	FUNCT	PROC	WAIT	ERRNO	!

Description :

Permet d'accéder aux paramètres d'entraînement Sercos pour les écrire. Outre des valeurs individuelles, il est également possible d'écrire des listes pouvant contenir jusqu'à 16 valeurs.

Conditions préalables :

- Le paramètre est valide et peut être modifié dans la phase Sercos 4.
- La valeur du paramètre se situe dans la plage autorisée.

Ordres CPL

- L'entraînement est connecté à la commande via l'interface Sercos ; la communication entre la commande et l'entraînement fonctionne de manière cyclique.



Aucun bloc CN ne doit être en cours de traitement lors de l'appel de l'ordre WSCS. Pour cela, il suffit de programmer au préalable un ordre WAIT.



WSCS peut être utilisé comme alternative à la fonction CN "WriteID3(WID3)".



L'ordre CPL "SCS3" permet de lire les paramètres d'entraînement Sercos.



Des modifications inadaptées ou arbitraires des paramètres d'entraînement Sercos risquent d'endommager la pièce à usiner et/ou la machine, et de provoquer des réactions imprévues et dangereuses de la machine.

Pour un utilisation correcte, il est nécessaire d'obtenir des informations sur les paramètres d'entraînement Sercos disponibles dans l'entraînement. Pour cela, veuillez consulter la documentation relative à l'entraînement.

Syntaxe :

WSCS (<sélection d'axe>,<n° ID>,<longueur du tableau>,<valeur(s)>,<unité>)	
<sélection d'axe>	Index d'axe du système ou nom d'axe du système
<n° ID>	Numéro d'identification Sercos (S-x-xxxx, P-x-xxxx, S-x-xxxx.xx.xx, P-x-xxxx.xx.xx)
<longueur du tableau>	0 Écrire un paramètre individuel 1 à 16 Écrire une liste de paramètres
<valeur(s)>	Pour <longueur du tableau> = 0 : Expression CPL Pour <longueur du tableau> = 1 à 16 : Nom d'une variable de tableau

{<unité>}	Optionnel. 1 : (par défaut) écriture avec prise en compte de l'unité et du calibrage 0 : écriture sans unité ni calibrage
ERRNO	Variable CPL Si la variable CPL ERRNO est indiquée à un emplacement quelconque dans la liste de paramètres, aucune erreur d'exécution n'est générée. L'erreur est fournie par une valeur correspondante de la variable. Les valeurs de retour suivantes sont possibles : 0 : Accès OK -1 : Erreur de paramètre -13 : Échec lors de l'écriture des données -26 : Ordre avec valeur de paramétrage non valide -27 : Ordre avec type de paramètre non valide -36 : Canal de service bloqué -37 : Numéro d'identification inconnu -38 : Numéro d'identification non valide -39 : Numéro d'axe non valide -42 : Valeur limite dépassée -43 : Nombre d'éléments non valide Si la variable CPL ERRNO n'est pas indiquée, une erreur d'exécution est générée en cas d'erreur d'accès.

Tab. 7-97: Syntaxe WSCS

Écriture de paramètres individuels :

Une <longueur du tableau> égale à 0 permet d'écrire des paramètres individuels.

<valeur(s)> peut être n'importe quelle expression CPL.

Une valeur est inscrite dans un paramètre du 5e axe du système. Si le canal de service est attribué, 100 ms s'écoulent avant de répéter l'opération d'écriture :

```
...
N10 WAIT
000 .WRTORQUE
001 WSCS(5,"S-0-92",0,400.0,ERRNO)
002 IF ERRNO <> 0 THEN
003   IF ERRNO = -36 THEN
004     WAIT(,100)
005     GOTO .WRTORQUE
006   ELSE
007     REM   Error handling
009   ENDIF
010 ENDIF
...
```

Fig. 7-3: Écriture d'un paramètre individuel (exemple 1)

Écriture d'un paramètre de l'axe Z du canal :

```
...
N10 WAIT
000 SAX% = SD(23,"Z")
001 WSCS(SAX%,"S-0-92",0,400.0,ERRNO)
002 IF ERRNO <> 0 THEN
007   REM   Error handling
```

Ordres CPL

```
010 ENDIF
...
```

Écriture d'une liste de paramètres :

Fig. 7-4: Écriture d'un paramètre individuel (exemple 2)

Les listes de paramètres sont remplies par transfert du nom et du nombre d'éléments utilisés d'une variable de tableau.

La <longueur du tableau> (1 à 16) définit le nombre d'éléments de la liste devant être inscrits.

<valeur(s)> doit correspondre au nom d'une variable de tableau unidimensionnelle.

Dans l'extrait de programme ci-dessous, la variable de tableau `VALUE!` est créée et remplie avec des valeurs.

Le nom de la variable de tableau et le nombre d'éléments devant être utilisés sont transmis à `WSCS`.

Programme:

```
...
N10 WAIT
000 DIM VALUE!(5)
001 VALUE!(1) = 3.5
002 VALUE!(2) = 0.2
003 WSCS("X", "P-0-4006", 2, VALUE!, ERRNO)
004 IF ERRNO <> 0 THEN
005 REM Error handling
006 ENDIF
...
```

Fig. 7-5: Écriture d'une liste de paramètres

Particularités : {<unité>} = 0 permet à "SCS3" de réécrire les valeurs de paramètres lues dans l'entraînement.

7.4.7 Interface API
BITIF

BOOL	READ	WRITE	FUNCT	PROC	WAIT	ERRNO	!

Description : **BIT InterFace** – Interface bit

Cet ordre permet d'accéder à l'interface numérique entre la commande numérique et l'API :

- Toutes les entrées et sorties peuvent être lues.
- Les sorties client CPL Ch_Cpl01...16 dans l'interface du canal peuvent également être écrites.

Syntaxe :	BITIF (<SignalBit>[,<index>[,<unité IF>]])
	avec

Ordres CPL

<SignalBit>	Décalage de bit ou adresse symbolique Décalage de bit : 0 à 31 pour l'interface globale 0 à 111 pour l'interface du canal 0 à 95 pour l'interface de l'axe 0 à 95 pour l'interface de la broche 0 à 7 pour l'interface à grande vitesse (HighSpeed) Pour les adresses symboliques, voir la documentation "Interface API Rexroth IndraMotion MTX".
<index>	Index de l'axe, de la broche ou du canal <i>Si aucun index n'est programmé, l'élément suivant est utilisé pour la requête :</i> • canal actif pour l'interface du canal, • axe 1 pour l'interface de l'axe, • broche 1 pour l'interface de la broche.
<unité IF>	0 ou QCH : signaux de sortie API, spécifiques au canal 1 ou QAX : signaux de sortie API, spécifiques à l'axe 2 ou QSP : signaux de sortie API, spécifiques à la broche 3 ou ICH : signaux d'entrée API, spécifiques au canal (sorties client CPL de la commande numérique incluses) 4 ou IAX : signaux d'entrée API, spécifiques à l'axe 5 ou ISP : signaux d'entrée API, spécifiques à la broche 6 ou QGEN : signaux de sortie API, globaux 7 ou IGEN : signaux d'entrée API, globaux 8 ou QHS : signaux d'entrée API, grande vitesse 9 ou IHS : signaux de sortie API, grande vitesse

Tab. 7-98: Syntaxe BITIF

Exemples de syntaxe :

<code>I?=BITIF("QGEN_RESET")</code>	Lecture du signal de sortie global "Remise à zéro du système"
<code>I?=BITIF("ICH_RESET",1)</code>	Lecture du signal d'entrée du canal "Remise à zéro du canal" du canal 1
<code>I?=BITIF(0,1,"IAX")</code>	Lecture du signal d'entrée de l'axe "Point de référence connu" pour l'axe 1
<code>I?=BITIF(88,2,5)</code>	Lecture du signal d'entrée de la broche "Erreur – Classe d'état -1" pour la broche 2
<code>BITIF("ICH_CPL05")=TRUE</code>	Définir la sortie client CPL 5 sur TRUE
<code>BITIF(50,1,3)=TRUE</code>	
<code>BITIF("ICH_CPL02")=FALSE</code>	



Pour les signaux HighSpeed (HS), l'unité IF "QHS" (8) et les signaux de sortie "IHS" (7) sont indiqués avec les signaux d'entrée.

Ordres CPL

PLC

MULTYP	READ	WRITE	FUNCT	PROC	WAIT	ERRNO	!

Description : **Programmable Logic Controller** – Accès aux opérandes de l'API

Cet ordre permet d'accéder aux opérandes de l'API.

Syntaxe :

PLC (<type>,<param. vide>,<adresse>,<taille>)	
avec	
<type>	1 : entrée (E) 2 : sortie (S) 3 : marqueur (M) L'accès en lecture est permis pour tous les types. L'accès en écriture est uniquement autorisé pour le marqueur.
<param. vide>	(non attribué)
<adresse>	Adresse à octets pertinente à partir du début de la zone La commande vérifie le paramètre conformément à l'API actif.
<taille>	Taille du type de données : 1 : octet 2 : mot 4 : mot double
PLC (<type>, <param. vide>, <adresse>, <taille> [, <type de données>])	
<type de données>	Optionnel : Type pour l'accès aux opérandes de l'API 0 : INTEGER (par default) 1 : REAL
ERRNO	Variable CPL Si la variable CPL ERRNO est indiquée à un emplacement quelconque dans la liste de paramètres, aucune erreur d'exécution n'est générée. L'erreur est fournie par une valeur correspondante de la variable. Les valeurs de retour suivantes sont possibles : 0 : Accès OK < 0 : Erreur (pour une description précise de l'erreur, voir chap. 3.14.5 "Variable ERRNO pour l'analyse des erreurs des fonctions CPL ERRNO" à la page 85) Si la variable CPL ERRNO n'est pas indiquée, une erreur d'exécution est générée en cas d'erreur d'accès.

Tab. 7-99: Syntaxe PLC



En cas d'utilisation du paramètre <type de données>, le même type doit être utilisé pour les mêmes opérandes lors de l'accès en lecture et en écriture !

Exemple : :

30 REM Lire 2 octets à partir de l'entrée 10


```
40 I% = PLC(1,,10,2)
:
```

7.4.8 Fonctions actives

NCF

CHAR	READ	WRITE	FUNCT	PROC	WAIT	ERRNO	!

Description : **NC Functions** – Fonctions CN actives et fonctions auxiliaires modales

Fournit la syntaxe des fonctions CN programmées à la fin du groupe modal ou du groupe de fonctions auxiliaires de la <fonction CN>. Tous les groupes modaux de la commande ainsi que les fonctions auxiliaires définies dans les groupes de fonctions auxiliaires peuvent être interrogés.

En option, il est possible d'interroger également les états modaux du bloc d'interruption au sein d'un sous-programme asynchrone (réglage par un paramètre).

Le bloc d'interruption est :

- dans le cadre de l'avance des blocs, le bloc précédant le bloc cible sélectionné ;
- dans le cadre de sous-programmes asynchrones réguliers, le bloc CN à partir duquel le saut vers le sous-programme asynchrone a été effectué.

La variable dans laquelle le résultat doit être enregistré doit être de type "Tableau dimensionné".

Les conflits de type sont détectés au cours de l'exécution du programme et acquittés par un message d'erreur.

Syntaxe :

NCF (<fonction CN>)	
NCF (<fonction CN> [, <DataSelection>])	
<fonction CN>	Syntaxe d'une fonction CN ou d'une fonction auxiliaire quelconque Si la syntaxe programmée est inexistante, une erreur d'exécution est générée.
<DataSelection>	0 : (par défaut) L'ordre fournit la dernière fonction programmée. 1 : L'ordre fournit la fonction active dans le bloc d'interruption (uniquement en combinaison avec l'avance de bloc et les sous-programmes asynchrones).

Tab. 7-100: Syntaxe NCF

Exemple :

10 DIM A\$(4)	Dimensionner un tableau pour une chaîne de caractères d'une longueur maximale de 4 caractères
:	
20 A\$=NCF("G1")	La syntaxe de la dernière fonction CN programmée du groupe contenant "G1" en tant que syntaxe est attribuée à la variable STRING A\$.
:	
N80 [A\$]	La fonction CN interrogée auparavant est de nouveau programmée.
N10 M3 S1234 T2345	
123 DIM A\$(4)	

Ordres CPL

```

20 A$=NCF("M3")
N30 [A$]           La fonction CN interrogée auparavant est de nouveau pro-
                    grammée.
50 PRN#(0,"",A$," est actif dans le groupe M3")
N30 M5
33 A$=NCF("M3")
34 PRN#(0,"",A$," est actif dans le groupe M3")
N70 M30

```



La syntaxe de broche de la commande IndraMotion MTX peut être configurée librement ; elle est réglée dans les paramètres machine 1040 00101 et suivants.

7.4.9 Interface CN (NCS)

Généralités

Les ordres pour le couplage NCS permettent d'accéder aux services de processus et de données de l'interface NCS interne via le langage CPL.

MCODS (description générale)

	INT	READ	WRITE	FUNCT	PROC	WAIT	ERRNO	!
Description	Motion Control Data Service – Services de données Motion Control							

Appelle les services de données Motion Control de la NCS via le langage CPL. Cela permet d'extraire les états et les données de la commande numérique.



Toutes les valeurs extraites se réfèrent au moment où le bloc CPL est traité par le traitement du bloc.

Si le programme appelant contient également des blocs CN, la préparation des blocs est généralement en avance sur traitement par la machine. Si des fonctions déterminant les données machine actuelles ou les états de la machine actuels influencent le déroulement des programmes, la "temporisation" entre la préparation des blocs et l'état actuel de la machine doit être supprimée. Cependant, ceci ne concerne que la mise en œuvre des fonctions qui accèdent justement au canal sur lequel elles sont elles-mêmes exécutées.

Dans ce cas, utiliser l'ordre "WAIT" sur la ligne précédant l'appel. Cela permet d'arrêter le traitement du bloc jusqu'à ce que tous les blocs situés avant "WAIT" soient effectivement traités.

!

L'ordre "MCODS" ne peut pas être utilisé en combinaison avec l'avance de bloc ! La raison en est que MCODES accède aux données du moment actif (interpolation). L'interpolation est neutralisée au cours de l'avance de bloc. Par conséquent, l'état actif au début de la comparaison reste identique à l'état avant l'avance de bloc.

Pour un grand nombre d'ordres "MCODS", il existe des ordres CPL spéciaux qui fournissent des données identiques ou similaires. Ces ordres CPL sont mentionnés dans le tableau synoptique, dans la colonne "voir aussi sous".

Les ordres fournissent une valeur de retour (voir [chap. "Valeurs de retour pour une erreur" à la page 583](#)).

Syntaxe :

MCODS (<type> , <canal> , <version> , <mémoire tampon> , <taille> , <numéro d'axe> , <numéro d'identification> [, <P1>])	
<type>	Expression INTEGER Indique la fonction à exécuter. Le tableau ci-dessous présente la liste de toutes les fonctions disponibles.
<canal>	Expression INTEGER Indique le canal sur lequel la fonction doit agir.
<version>	Variable INTEGER ou REAL initialisée (ce n'est pas une constante !) Si le contenu de la variable lors de l'appel de l'ordre est égal à 0, la fonction déterminée par le <type> enregistre immédiatement les données demandées dans la <mémoire tampon>. En outre, l'ordre identifie la version des données fournies dans <version>. Si cette identification de la version est encore contenue dans la variable lors de l'appel de l'ordre suivant, alors la fonction écrit les données requises dans la <mémoire tampon> non pas immédiatement mais seulement après la modification suivante des données. Cela permet de poursuivre une boucle de programme jusqu'à ce qu'un canal ait atteint un certain état. Il convient toutefois d'intégrer ici une condition de dépassement de temps (par ex. compteur ou laps de temps écoulé) dans cette boucle afin d'éviter les boucles sans fin !
<mémoire tampon>	L'ordre renvoie les valeurs de données demandées dans la <mémoire tampon>. <i>En fonction du type de données, la <mémoire tampon> doit être :</i> • une variable simple de type "INTEGER", "REAL", "DOUBLE" ; • une variable de tableau de type "INTEGER", "REAL", "DOUBLE" ; • une variable STRING (tableau de caractères unidimensionnel).

Tab. 7-101: Syntaxe MCODES



Avec des variables STRING ou de champ, seul le nom de la variable doit être indiqué, sans dimension ni index !

Ordres CPL

<taille>	Expression INTEGER Définit la taille du tableau de la <mémoire tampon>. Si la <mémoire tampon> n'est pas une variable de tableau, mais une variable simple du type "INTEGER", "REAL" ou "DOUBLE", la valeur 1 doit être paramétrée dans <taille>.
<numéro d'axe>	Expression INTEGER Indique le numéro d'axe d'un axe du système.
<numéro d'identification>	Expression INTEGER Fournit la valeur d'un <numéro d'identification> à partir du télégramme d'axe cyclique pour tous les axes.

Tab. 7-102: Paramètres MCODS



La taille d'une variable de tableau utilisée doit être définie auparavant via l'ordre DIM et ne doit pas être dépassée dans le paramètre <taille> !

Ces variables doivent être dimensionnées avant l'ordre MCODS et dans le même fichier !

<P1>	Paramètre optionnel dépendant du <type>.

Tab. 7-103: Syntaxe 3 MCODS

Vue d'ensemble des ordres MCODS(...)

L'ordre fournit :	Syntaxe :	Voir aussi sous :
Positions		
Position de consigne de l'axe	MCODS(1..)	SPOS
Position de consigne de l'axe	MCODS(2..)	SPOS
Position réelle de l'axe de l'entraînement	MCODS(35..)	APOS
Position réelle de l'axe de la CNC	MCODS(147..)	–
Valeurs réelles de l'axe (système de coordonnées de la pièce à usiner)	MCODS(38..)	–
Poursuite	MCODS(4..)	–
Valeur du programme de l'axe (système de coordonnées du programme)	MCODS(37..)	PCS
Positions finales programmées, y compris les décalages	MCODS(16..)	WCS
Positions finales programmées, sans les décalages	MCODS(23..)	PCS
Coordonnées d'axe dans l'ACS1	MCODS(73..)	–
Coordonnées d'axe dans l'ACS0	MCODS(109..)	ACS
Course restante dans le système de coordonnées de la pièce à usiner (WCS)	MCODS(83..)	–
Course restante dans le système de coordonnées de l'axe (ACS1)	MCODS(85..)	–
Vitesse et vitesse de rotation		

Ordres CPL

L'ordre fournit :	Syntaxe :	Voir aussi sous :
Avance sur trajectoire	MCODS(3..)	SDR(5,1,2)
Avance sur trajectoire programmée	MCODS(24..)	SDR(5,1,2)
Avances sur trajectoire programmées	MCODS(142..)	
Avance sur trajectoire programmée active	MCODS(143..)	
Vitesses de Jog	MCODS(27..)	–
Vitesse de rotation de consigne/vitesse de coupe de la broche	MCODS(5..)	SD(205,2,...)
Vitesse de rotation réelle de la broche	MCODS(36..)	SDR(205,3,...)
Vitesse de rotation maximale de la broche	MCODS(19..)	–
Vitesse de rotation minimale de la broche	MCODS(20..)	–
Vitesses de rotation programmées de la broche	MCODS(25..)	SDR(205,2,...)
Vitesse actuelle de l'axe	MCODS(119..)	–
Vitesse de rotation actuelle de l'axe	MCODS(120..)	–
Vitesse de rotation de l'axe programmée	MCODS(121..)	–
Vitesse maximale de l'axe	MCODS(131..)	–
Valeur de crête et valeur maximale de la vitesse de l'axe	MCODS(145..)	–
Valeur de crête et valeur maximale de l'accélération de l'axe	MCODS(146..)	–
États		
État "InPos"	MCODS(6..)	BITIF("IAX_INPOS",...)
État "Point de référence approché"	MCODS(26..)	BITIF("IAX_REFKNOWN",...)
État "Mode test"	MCODS(29..)	BITIF("ICH_DRYRUN",...)
État SAV et IPO	MCODS(32..)	BITIF("ICH_STATE",...)
État "Temporisation active"	MCODS(39..)	–
État "Fonction auxiliaire avec acquittement obligatoire active"	MCODS(40..)	–
État "Validation d'enregistrement"	MCODS(41..)	BITIF("QCH_TRANSFERLOCK",...)
État "Instruction de déplacement"	MCODS(47..)	BITIF("IAX_TRVCMD",...)
État "Blocage d'avance"	MCODS(49..)	BITIF("QAX_DRVLOCK",...)
États d'attente du canal	MCODS(87..)	–
Remise à zéro du canal, en attente de	MCODS(129,..)	–
Potentiomètres		
Valeur des potentiomètres d'avance	MCODS(7..)	SDR(2,1)
Valeur des potentiomètres des broches	MCODS(8..)	SDR(202,...)
Valeur des potentiomètres des axes	MCODS(50..)	–
Corrections		
Numéro actif de la correction du rayon de l'outil	MCODS(11..)	TCV(0)
Correction active du rayon de l'outil	MCODS(12..)	TCV(4)

Ordres CPL

L'ordre fournit :	Syntaxe :	Voir aussi sous :
Nom de la table de correction d'outil active	MCODS(13..)	–
Nom de la table DO d'axe active	MCODS(14..)	–
Valeurs DO d'axe actives	MCODS(15..)	ZOV
Correction d'outil active	MCODS(54..)	–
Placement du jeu de données actif	MCODS(112..)	–
Chemin et nom du tableau de placement actif	MCODS(113..)	–
Correction de la position de la pièce à usiner	MCODS(114..)	DPC
Somme des placements	MCODS(115..)	PMV
Décalage actif des coordonnées du programme	MCODS(116..)	SDR(168)
Décalage additif actif des coordonnées du programme	MCODS(117..)	SDR(268)
Somme du décalage actif des coordonnées du programme	MCODS(118..)	–
Valeurs de correction en ligne	MCODS(89..)	–
État de la correction en ligne	MCODS(90..)	–
Compensation actuelle de la température	MCODS(132..)	–
Compensation actuelle de l'erreur de pas de la broche	MCODS(133..)	–
Compensation de croix actuelle	MCODS(134..)	–
Décalage de compensation actuel	MCODS(135..)	–
Modes de fonctionnement		
Mode de fonctionnement du canal	MCODS(31..)	BITIF("ICH_OPMODE",...)
Mode de fonctionnement de l'axe	MCODS(48..)	–
Structure du système		
Nombre d'axes d'avance, d'axes auxiliaires, de broches ; types de mouvements, types d'entraînements	MCODS(34..)	–
Nombre de canaux	MCODS(44..)	–
Nombre d'axes	MCODS(45..)	–
Noms des axes	MCODS(33..)	AXINF
Noms actifs des axes du canal	MCODS(59..)	AXADR\$
Noms des axes du canal (réglage par défaut)	MCODS(60..)	–
Attribution axe-canal	MCODS(43..)	SD(25,...)
Attribution axe-canal par défaut	MCODS(58..)	–
Numéro d'identification issu du télégramme d'axe cyclique	MCODS(62..)	–
Noms des coordonnées	MCODS(78..)	AXINF\$
Nombre de coordonnées	MCODS(82..)	–
Jeu de données actuel du décalage d'origine	MCODS(99..)	ZOV
Somme actuelle du décalage d'origine	MCODS(100..)	ZOV
Noms des coordonnées de machine	MCODS(106..)	–

L'ordre fournit :	Syntaxe :	Voir aussi sous :
Nombre de coordonnées de machine	MCODS(107..)	–
Unités de mesure		
Unités de mesure des axes (réglage par défaut)	MCODS(61..)	–
Unité de mesure des axes	MCODS(53..)	–
Type de programmation (pouces/métrique)	MCODS(18..)	NCF("G70")
Programmation du diamètre	MCODS(91..)	NCF("DIA")
Fonctions auxiliaires		
Numéros des groupes de fonctions auxiliaires	MCODS(94..)	–
Syntaxe active des groupes de fonctions auxiliaires	MCODS(65..)	–
Broches		
Fonctions de mouvement des broches	MCODS(63..)	SD(212,2)
Rapports de réduction des broches	MCODS(64..)	BITIF("ISP_GEAR1..4SELECT",...)
Attribution des broches du canal à des groupes	MCODS(65..)	–
Information sur l'activation de la commutation du réducteur	MCODS(66..)	–
Sélection automatique ou manuelle des rapports de réduction	MCODS(67..)	–
Noms des broches	MCODS(102..)	–
Nombre de broches	MCODS(103..)	–
Numéro du système de la broche	MCODS(104..)	–
Numéro d'axe de la broche	MCODS(105..)	–
Position réelle de la broche	MCODS(108..)	–
Couple de la broche	MCODS(110..)	–
Puissance de la broche	MCODS(111..)	–
Entraînement		
Version du fabricant	MCODS(55..)	–
Type de variateur	MCODS(56..)	–
Type de moteur	MCODS(57..)	–
Divers		
Messages dans le programme pièce	MCODS(28..)	–
Chemin et nom du programme principal	MCODS(30..)	–
Stratégie de redémarrage et enregistrement des déplacements en mode Jog	MCODS(46..)	–
Données spécifiques au client	MCODS(42..)	–
Arrêt optionnel (activé)	MCODS(68..)	BITIF("ICH_BLKSLASH",...)
Sauter séquence (activer)	MCODS(69..)	BITIF("ICH_OPTSTOP",...)
Resélection automatique du programme active	MCODS(70..)	BITIF("QCH_RESELOFF",...)
Angle de rotation	MCODS(122..)	SCL(0)

Ordres CPL

L'ordre fournit :	Syntaxe :	Voir aussi sous :
Position du pôle pour les fonctions Miroir et Rotation	MCODS(123..)	SCL(1,...)
Coordonnées de décalage	MCODS(124..)	COF
Facteurs de mise à l'échelle actifs	MCODS(125..)	SCL(2,...)
Inversion miroir des coordonnées active	MCODS(126..)	SCL(2,...)
Avance de bloc		
Statut de comparaison	MCODS(128,..)	–
Statut Marque de réentrée de l'avance de bloc	MCODS(127,..)	–

Tab. 7-104: Vue d'ensemble des ordres MCODES



Dans le tableau ci-dessous, des constantes INTEGER sont parfois indiquées comme paramètres de la syntaxe. Il est également possible de programmer des variables INTEGER à la place de ces constantes. Dans ce cas, la valeur indiquée doit toutefois leur être attribuée au moment de l'appel de l'ordre.

MCODS(1,...) – Position de consigne de l'axe

Description : Fournit dans la <mémoire tampon> les positions de consigne de tous les **axes d'avance et auxiliaires du système**, par ordre croissant et indépendamment du canal :

- en millimètres pour les axes linéaires ;
- en degrés pour les axes ronds.

Les valeurs tiennent compte du "réglage de la valeur réelle" (par ex. SetPos).

Mise à jour : Cyclique

Type de données de la <mémoire tampon> : DOUBLE, tableau

Syntaxe : MCODS(1,-1,<version>,<mémoire tampon>,<taille>)

Tab. 7-105: Syntaxe MCODES(1,...)

MCODS(2,...) – Position de consigne de l'axe (INTEGER)

Description : Fournit dans la <mémoire tampon> les positions de consigne de tous les **axes d'avance et auxiliaires du système**, par ordre croissant et indépendamment du canal :

- en 0,0001 mm pour les axes linéaires ;
- en 0,0001 degré pour les axes ronds.

Les valeurs tiennent compte du "réglage de la valeur réelle" (par ex. SetPos).

Mise à jour : Cyclique

Type de données de la <mémoire tampon> : INTEGER, tableau

Syntaxe : MCODS(2,-1,<version>,<mémoire tampon>,<taille>)

Tab. 7-106: Syntaxe MCODES(2,...)

MCODS(3,...) – Avance sur trajectoire

Description : Fournit dans la <mémoire tampon> 3 valeurs avec les avances sur trajectoire actuelles du <canal> (y compris le potentiomètre d'avance), en mm/min et par ordre croissant :

1. La vitesse de consigne définie de manière externe pour l'interpolateur.
2. La vitesse réelle de l'interpolateur (= vitesse sur trajectoire actuelle).
3. La vitesse de consigne interne de l'interpolateur. Elle peut être différente de celle prédéfinie de manière externe par une application (par ex. par la fonction Feed-Adapt).

Si la programmation de l'avance est exprimée in mm/tr (G95), elle fournit l'avance sur trajectoire en mm/min.

Mise à jour : Cyclique

Type de données de la <mémoire tampon> : REAL, tableau

Syntaxe :

MCODS(3,<canal>,<version>,<mémoire tampon>,3)

Tab. 7-107: Syntaxe MCODES(3,...)

MCODS(4,...) – Poursuite

Description : Fournit dans la <mémoire tampon> la poursuite de tous les axes du système, par ordre croissant :

- en millimètres pour les axes linéaires ;
- en degrés pour les axes ronds.

Si le transfert de la poursuite n'est pas pris en charge par les entraînements (via le paramètre Sercos), la valeur 0.0 est renvoyée.

Mise à jour : Cyclique

Type de données de la <mémoire tampon> : REAL, tableau

Syntaxe :

MCODS(4,-1,<version>,<mémoire tampon>,<taille>)

Tab. 7-108: Syntaxe MCODES(4,...)

MCODS(5,...) – Vitesse de rotation de consigne/Vitesse de coupe de la broche

Description : Fournit dans la <mémoire tampon> les vitesses de rotation de consigne des broches ou les vitesses de coupe de toutes les broches existantes sur le <canal>, par ordre croissant (S ou S1, S2, S3, etc.). Si <canal> = -1 ou 0, fournit dans la <mémoire tampon> les données des broches du système, sinon celles des broches du canal. Si G96 est active, les vitesses de coupe sont fournies en m/min, sinon les vitesses de consigne actuelles de la broche sont fournies en tr/min. Cela comprend le potentiomètre, les limitations de la vitesse de rotation (SMin, SMax) et les limitations dues au rapport de réduction. S'il n'existe aucune broche, 0.0 est indiqué dans la <mémoire tampon> à l'emplacement correspondant.

Mise à jour : Cyclique

Type de données de la <mémoire tampon> : REAL, tableau

Syntaxe :

MCODS(5,<canal>,<version>,<mémoire tampon>,<taille>)

Tab. 7-109: Syntaxe MCODES(5,...)

Ordres CPL

MCODS(6,...) – InPos

Description : Fournit dans la <mémoire tampon> la valeur 0 ou 1 en tant que signal InPos pour chaque **axe d'avance et auxiliaire**, par ordre croissant et indépendamment du canal :

L'axe est en position : 1

L'axe n'est pas en position : 0

Un axe est en position s'il se situe dans la fenêtre InPos paramétrée (MP 1015 00100) et si aucune instruction de déplacement (voir également MCODES(47...)) n'est en suspens.

Mise à jour : Cyclique

Type de données de la <mémoire tampon> : INTEGER, tableau

Syntaxe : **MCODS(6,-1,<version>,<mémoire tampon>,<taille>)**

Tab. 7-110: Syntaxe MCODES(6,...)

MCODS(7,...) – Valeur du potentiomètre d'avance

Description : Fournit dans la <mémoire tampon> la valeur actuelle du potentiomètre d'avance du <canal>, en 1/100 %.

Mise à jour : Cyclique

Type de données de la <mémoire tampon> : REAL

Syntaxe : **MCODS(7,<canal>,<version>,<mémoire tampon>,1)**

Tab. 7-111: Syntaxe MCODES(7,...)

MCODS(8,...) – Valeurs des potentiomètres des broches

Description : Fournit dans la <mémoire tampon> les valeurs actuelles des potentiomètres d'avance, en 1/100 % et par ordre croissant (S ou S1, S2, S3, etc.).

Si <canal> = -1 ou 0, fournit dans la <mémoire tampon> les données des broches du système, sinon celles des broches du canal.

Mise à jour : Cyclique

Type de données de la <mémoire tampon> : REAL, tableau

Syntaxe : **MCODS(8,<canal>,<version>,<mémoire tampon>,<taille>)**

Tab. 7-112: Syntaxe MCODES((8,...)

MCODS(9,...)

Description : Sans pertinence pour la MTX

MCODS(10,...)

Description : Sans pertinence pour la MTX

MCODS(11,...) – Numéro de la correction de rayon d'outil active

Description : Fournit dans la <mémoire tampon> le numéro de la correction du rayon d'outil actif pour le <canal>.

Si aucune correction du rayon n'est active, la valeur renvoyée est -1.

Mise à jour : Après chaque changement de bloc

Type de données de la <mémoire tampon> : INTEGER

Syntaxe :

MCODS(11,<canal>,<version>,<mémoire tampon>,1)

Tab. 7-113: Syntaxe MCODES(11,...)

MCODS(12,...) – Correction du rayon d'outil active

Description : Fournit dans la <mémoire tampon> la correction du rayon d'outil active sur le <canal>.

Si aucune correction du rayon d'outil n'est active, la valeur renvoyée est 0.0.

Mise à jour : Après chaque changement de bloc

Type de données de la <mémoire tampon> : REAL

Syntaxe :

MCODS(12,<canal>,<version>,<mémoire tampon>,1)

Tab. 7-114: Syntaxe MCODES(12,...)

MCODS(13,...) – Nom du tableau de la correction d'outil active

Description : Fournit dans la <mémoire tampon> le nom de la table de correction d'outil active sur le <canal>.

Si aucune table n'est active, une chaîne de caractères composée de 3 espaces est renvoyée.

Mise à jour : À chaque changement de bloc

Type de données de la <mémoire tampon> : CHARACTER, tableau

Syntaxe :

MCODS(13,<canal>,<version>,<mémoire tampon>,<taille>)

Tab. 7-115: Syntaxe MCODES(13,...)

MCODS(14,...) – Nom de la table DO d'axe active

Description : Fournit dans la <mémoire tampon> le nom de la table DO d'axe active sur le <canal>.

Si aucune table n'est active, une chaîne de caractères composée de 3 espaces est renvoyée.

Mise à jour : À chaque changement de bloc

Type de données de la <mémoire tampon> : CHARACTER, tableau

Syntaxe :

MCODS(14,<canal>,<version>,<mémoire tampon>,<taille>)

Tab. 7-116: Syntaxe MCODES(14,...)

MCODS(15,...) – Valeurs DO d'axe actives

Description : Fournit dans la <mémoire tampon> les valeurs DO actives sur le <canal> des 3 premières banques pour toutes les coordonnées de machine, en mm ou en degrés.

Si aucun décalage n'est actif, la valeur 0.0 est renvoyée.

L'ordre suivant s'applique ici :

- Décalage de la 1re coordonnée de machine dans la banque 1
- Décalage de la 2e coordonnée de machine dans la banque 1
- Décalage de la n-ième coordonnée de machine dans la banque 1
- Décalage de la 1re coordonnée de machine dans la banque 2

:

Ordres CPL

- Décalage de la n-ième coordonnée de machine dans la banque 2

:

- Décalage de la n-ième coordonnée de machine dans la banque 3

L'ordre MCODS(99, ...) fournit les valeurs pour l'ensemble des 5 banques de décalage.

Mise à jour : Après chaque changement de bloc

Type de données de la <mémoire tampon> : REAL, tableau

Syntaxe : MCODS(15,<canal>,<version>,<mémoire tampon>,<taille>)

Tab. 7-117: Syntaxe MCODS(15,...)

MCODS(16,...) – Position finale du bloc actuel, y compris les aides à la saisie

Description : Fournit dans la <mémoire tampon> les positions finales des blocs actifs de tous les **axes d'avance et auxiliaires** par rapport aux coordonnées de la pièce à usiner, par ordre croissant et indépendamment du canal :

- en millimètres pour les axes linéaires ;
- en degrés pour les axes ronds.

Toutes les valeurs de décalage sont prises en compte.

Le "réglage de la valeur réelle" (par ex. SetPos) n'est **pas** pris en compte dans les valeurs.

Mise à jour : À chaque changement de bloc

Type de données de la <mémoire tampon> : REAL, tableau

Syntaxe : MCODS(16,0,<version>,<mémoire tampon>,<taille>)

Tab. 7-118: Syntaxe MCODS(16,...)

MCODS(17,...)

Description : Sans pertinence pour la MTX

MCODS(18,...) – Programmation en pouces/métrique

Description : Fournit dans la <mémoire tampon> le mode de programmation des axes existants sur le <canal> :

- 0 : En pouces
- 1 : Système métrique
- 2 : En degrés
- 3 : Axe inexistant

Mise à jour : À chaque changement de bloc

Type de données de la <mémoire tampon> : INTEGER

Syntaxe : MCODS(18,<canal>,<version>,<mémoire tampon>,1)

Tab. 7-119: Syntaxe MCODS(18,...)

MCODS(19,...) – Vitesse de rotation maximale de la broche

Description : Fournit dans la <mémoire tampon> les vitesses maximales admissibles de la broche, en tr/min et par ordre croissant (S ou S1, S2, S3, etc.).

Si <canal> = -1 ou 0 est transféré, la <mémoire tampon> fournit les données des broches du système ; sinon, elle fournit les données des broches du canal.

Les limitations de la vitesse de rotation sont prises en compte.

Mise à jour : À chaque changement de bloc

Type de données de la <mémoire tampon> : REAL, tableau

Syntaxe : **MCODS(19,<canal>,<version>,<mémoire tampon>,<taille>)**

Tab. 7-120: Syntaxe MCODES(19,...)

MCODS(20,...) – Vitesse de rotation minimale de la broche

Description : Fournit dans la <mémoire tampon> les vitesses minimales admissibles de la broche, en tr/min et par ordre croissant (S ou S1, S2, S3, etc.).

Si <canal> = -1 ou 0 est transféré, la <mémoire tampon> fournit les données des broches du système ; sinon, elle fournit les données des broches du canal.

Les limitations de la vitesse de rotation sont prises en compte.

Mise à jour : À chaque changement de bloc

Type de données de la <mémoire tampon> : REAL, tableau

Syntaxe : **MCODS(20,<canal>,<version>,<mémoire tampon>,<taille>)**

Tab. 7-121: Syntaxe MCODES(20,...)

MCODS(21,...)

Description : Sans pertinence pour la MTX

MCODS(22,...)

Description : Sans pertinence pour la MTX

MCODS(23,...) – Positions finales programmées sans décalages

Description : Identique à MCODES(16,...), mais sans décalages.

Mise à jour : À chaque changement de bloc

Type de données de la <mémoire tampon> : REAL, tableau

Syntaxe : **MCODS(23,0,<version>,<mémoire tampon>,<taille>)**

Tab. 7-122: Syntaxe MCODES(23,...)

MCODS(24,...) – Avance sur trajectoire programmée

Description : Fournit dans la <mémoire tampon> l'avance sur trajectoire programmée du <canal>, en mm/min.

Mise à jour : Après chaque changement de bloc

Type de données de la <mémoire tampon> : REAL

Syntaxe : **MCODS(24,<canal>,<version>,<mémoire tampon>,1)**

Tab. 7-123: Syntaxe MCODES(24,...)

Ordres CPL

MCODS(25,...) – Vitesse de rotation programmée de la broche

Description : Fournit dans la <mémoire tampon> les vitesses de rotation programmées de la broche, en tr/min et par ordre croissant (S ou S1, S2, S3, etc.).

Si <canal> = -1 ou 0 est transféré, la <mémoire tampon> fournit les données des broches du système ; sinon, elle fournit les données des broches du canal.

Mise à jour : Après chaque changement de bloc

Type de données de la <mémoire tampon> : REAL, tableau

Syntaxe : **MCODS(25,<canal>,<version>,<mémoire tampon>,<taille>)**

Tab. 7-124: Syntaxe MCODES(25,...)

MCODS(26,...) – Point de référence approché

Description : Fournit dans la <mémoire tampon> la valeur 0 ou 1 en tant que signal "Point de référence approché" pour chaque **axe d'avance et auxiliaire**, par ordre croissant et indépendamment du canal :

Point de référence approché : 1

Point de référence non approché : 0

Mise à jour : Après modification

Type de données de la <mémoire tampon> : INTEGER, tableau

Syntaxe : **MCODS(26,-1,<version>,<mémoire tampon>,<taille>)**

Tab. 7-125: Syntaxe MCODES(26,...)

MCODS(27,...) – Vitesses de Jog

Description : Fournit dans la <mémoire tampon> les vitesses de Jog actuelles de tous les **axes d'avance et auxiliaires du système**, par ordre croissant et indépendamment du canal :

– en mm/min pour les axes linéaires

– en tr/min pour les axes ronds

Mise à jour : Après modification

Type de données de la <mémoire tampon> : REAL, tableau

Syntaxe : **MCODS(27,-1,<version>,<mémoire tampon>,<taille>)**

Tab. 7-126: Syntaxe MCODES(27,...)

MCODS(28,...) – Remarques (messages)

Description : Fournit dans la <mémoire tampon> la dernière remarque programmée avec l'ordre MSG, GMSG ou PRN#(0,...).

Le paramètre <canal> sélectionne la remarque multicanaux ou spécifique à un canal :

-1 Fournit la dernière remarque multicanaux (GMSG).

0 à 60 Fournit la dernière remarque de l'ordre MSG ou PRN# sur le canal indiqué.

Mise à jour : Après modification

Type de données de la <mémoire tampon> : CHARACTER, tableau

Syntaxe :

MCODS(28,<canal>,<version>,<mémoire tampon>,80)

Tab. 7-127: Syntaxe MCODES(28,...)

MCODS(29,...) – Mode test

Description : Fournit dans la <mémoire tampon> la valeur 1 si le mode test est activé. Sinon, fournit la valeur 0.

Mise à jour : Après modification

Type de données de la <mémoire tampon> : INTEGER

Syntaxe :

MCODS(29,0,<version>,<mémoire tampon>,1)

Tab. 7-128: Syntaxe MCODES(29,...)

MCODS(30,...) – Chemin et nom du programme principal

Description : Fournit dans la <mémoire tampon> le chemin contenant le nom du programme principal sélectionné sur le <canal> sous forme de chaîne de caractères.

Si une valeur est indiquée dans <version>, elle est ignorée lors de l'appel de l'ordre.

La somme des longueurs du nom de fichier et du chemin est indiquée dans <taille>. Dans le système de fichiers interne, le chemin contenant le nom de fichier peut contenir un maximum de 100 caractères.

Pour les fichiers sauvegardés dans les systèmes de fichiers mounted, la valeur indiquée dans le paramètre <taille> dépend du nombre maximal de caractères pris en charge par le système de fichier externe pour le chemin et le nom de fichier.

Mise à jour : Après modification

Type de données de la <mémoire tampon> : CHARACTER, tableau

Syntaxe :

MCODS(30,<canal>,<version>,<mémoire tampon>,<taille>)

Tab. 7-129: Syntaxe MCODES(30,...)

MCODS(31,...) – Mode de fonctionnement du canal

Description : Fournit dans la <mémoire tampon> les éléments actifs sur le <canal> suivants :

- le mode de fonctionnement principal,
- le mode de sous-fonctionnement.

Les valeurs suivantes sont définies pour le mode de fonctionnement principal :

- 0 : Aucun mode de fonctionnement et donc aucun processus n'est actif.
- 1 : Mode Jog
Les axes peuvent être déplacés en mode Jog (+/-).
- 2 : Approche du point de référence
Les axes peuvent être démarrés à l'aide des signaux Manuel+ / Manuel-.
- 3 : Réserve
- 4 : Saisie manuelle
Les blocs CN individuels peuvent être affectés au traitement.
- 5 : Automatique (Bloc continu)

Ordres CPL

Les programmes pièces sont exécutés entièrement.

- 6 : Automatique (Bloc-programme)

Les blocs individuels d'un programme pièce sont traités les uns après les autres. Chaque bloc individuel est préparé et lancé avec le démarrage CN.

- 7 : Automatique (Étape unique)

Dans certaines conditions, la commande numérique génère et prépare plusieurs blocs à partir d'un seul bloc CN du programme pièce. Avec ce mode de fonctionnement, le démarrage CN transmet toujours un bloc individuel à l'interpolateur pour être traité.

- 8 : Apprentissage

Les axes peuvent être déplacés en mode Jog (+/-) ou avec la manivelle. Les blocs CN individuels peuvent être affectés au traitement.

- 9 : Réserve

- 10 : Automatique (Bloc unique)

Tous les blocs générés et préparés à partir d'un bloc CN individuel du programme pièce sont transmis à l'interpolateur via le démarrage CN pour être traités.

- 11 : Redémarrage

Les axes peuvent être éloignés manuellement du contour, et rapprochés de lui automatiquement ou manuellement.

- 12 : Débogueur CPL (mode Bloc-programme)

Les blocs individuels sont traités individuellement tels qu'ils apparaissent dans le programme pièce.

- 13 : Débogueur CPL (mode Bloc continu)

Les blocs sont tous exécutés jusqu'au point d'interruption suivant.

Les valeurs suivantes sont définies pour le mode de sous-fonctionnement :

- 0 : Aucun mode de sous-fonctionnement actif
- 1 : Écrasement actif Les blocs CN individuels peuvent être affectés au traitement.

Mise à jour : Après modification

Type de données de la <mémoire tampon> : INTEGER, tableau

Syntaxe : MCODS(31,<canal>,<version>,<mémoire tampon>,2)

Tab. 7-130: Syntaxe MCODES(31,...)

MCODS(32,...) – État SAV et IPO

Description : Fournit dans la <mémoire tampon> du <canal> :

- l'état SAV (préparation des blocs),
- l'état de l'interpolateur.

Les valeurs suivantes sont définies en tant qu'état SAV :

- 1 : Le mode de fonctionnement n'est pas actif. Il est possible de sélectionner un processus.
- 2 : Le mode de fonctionnement est opérationnel. Il est possible de démarrer un processus.
- 3 : Le mode de fonctionnement est actif. Un programme ou un bloc CN est en cours de traitement.

- 4 : Réservé
- 5 : Réservé
- 6 : Une erreur s'est produite dans le mode de fonctionnement. La remise à zéro ou la désélection du programme est nécessaire pour y remédier.
- 7 : Réservé
- 8 : Remise à zéro en cours d'exécution
- 9 : Un programme est sélectionné et est en cours de préparation (par ex. lié).
- 10 : "Effacement de la course restante" a été déclenché et n'est pas encore terminé.
- 11 : Le mode de fonctionnement est actif et traite à nouveau les mémoires tampon existantes.
- 12 : Le mode de fonctionnement est opérationnel. Le processus se trouve au début du programme et peut être lancé.
- 13 : Tous les blocs sont traités lors de la définition du bloc CN en mémoire tampon. Attente de la définition suivante.

Les valeurs suivantes sont définies en tant qu'état IPO :

- 1 : Interpolateur en marche
- 2 : L'interpolateur s'arrête en raison de l'arrêt de l'avance.
- 3 : L'interpolateur a arrêté les axes.

Mise à jour : Après modification

Type de données de la <mémoire tampon> : INTEGER, tableau

Syntaxe :

MCODS(32,<canal>,<version>,<mémoire tampon>,2)

Tab. 7-131: Syntaxe MCODES(32,...)

MCODS(33,...) – Noms d'axes

Description : Si <canal> = -1, la <mémoire tampon> fournit les noms à 9 octets de tous les axes du système configurés (1 à 100), par ordre croissant et en les séparant avec le caractère "0" (octet 0).

Si <canal> = -2, la <mémoire tampon> fournit les noms à 9 octets de tous les axes du système configurés (101 à 200), par ordre croissant et en les séparant avec le caractère "0" (octet 0).

Si <canal> = -3, la <mémoire tampon> fournit les noms à 9 octets de tous les axes du système configurés (201 à 250), par ordre croissant et en les séparant avec le caractère "0" (octet 0).

Si <canal> = numéro du canal existant, la <mémoire tampon> fournit les noms de tous les **axes du canal indiqué**, par ordre croissant et en les séparant avec le caractère "0" (octet 0).

Les noms comportant moins de 8 caractères sont complétés avec des espaces jusqu'au 8e caractère.

<taille> indique la capacité de la <mémoire tampon>. Pour 100 axes, elle peut comporter au maximum (9*100) 900 octets.

Pour obtenir un exemple, voir [chap. "Exemples de programmation" à la page 585](#).

Mise à jour : Après la montée en régime de la commande

Type de données de la <mémoire tampon> : CHARACTER, tableau

Ordres CPL

Syntaxe : **MCODS(33,<canal>,<version>,<mémoire tampon>,<taille>)**

Tab. 7-132: *Syntaxe MCODES(33,...)*

MCODS(34,...)

Description : Sans pertinence pour la MTX

MCODS(35,...) – Position réelle de l'axe de l'entraînement

Description : Fournit dans la <mémoire tampon> les positions réelles de tous les **axes d'avance et auxiliaires** que les entraînements transmettent à la commande numérique par l'intermédiaire de l'interface Sercos, par ordre croissant et indépendamment du canal :

- en millimètres pour les axes linéaires ;
- en degrés pour les axes ronds.

Le "réglage de la valeur réelle" (par ex. SetPos) n'est **pas** pris en compte dans les valeurs.



Les valeurs fournies par MCODES(35..) se distinguent de celles fournies par MCODES(147..) au niveau des corrections d'erreur d'axe LSEC, CCOMP et de la compensation de la température.

Mise à jour : Cyclique

Type de données de la <mémoire tampon> : REAL, tableau

Syntaxe : **MCODS(35,-1,<version>,<mémoire tampon>,<taille>)**

Tab. 7-133: *Syntaxe MCODES(35,...)*

MCODS(36,...) – Vitesse de rotation d'une liste de broches

Description : Fournit dans la <mémoire tampon> la vitesse réelle des broches, en tr/min et par ordre croissant (S ou S1, S2, S3, etc.). Si <canal> = -1 ou 0, la <mémoire tampon> fournit les données des broches du système, sinon elle fournit celle des broches du canal. Cela comprend le potentiomètre, les limitations de la vitesse de rotation (SMin, SMax) et les limitations dues au rapport de réduction. S'il n'existe aucune broche, 0.0 est indiqué dans la <mémoire tampon> à l'emplacement correspondant.

Mise à jour : Cyclique

Type de données de la <mémoire tampon> : REAL, tableau

Syntaxe : **MCODS(36,<canal>,<version>,<mémoire tampon>,<taille>)**

Tab. 7-134: *Syntaxe MCODES(36,...)*

MCODS(37,...) – Valeurs de coordonnées programmées

Description : Fournit dans la <mémoire tampon> les positions de consigne de l'interpolateur par rapport à la pièce à usiner pour tous les **axes d'avance du système**, par ordre croissant et indépendamment du canal :

- en millimètres pour les axes linéaires ;
- en degrés pour les axes ronds.

Le "réglage de la valeur réelle" (par ex. SetPos) et le décalage d'origine de l'axe (G54.x à G59.x) ne sont **pas** pris en compte dans les valeurs.

Mise à jour : Cyclique

Type de données de la <mémoire tampon> : DOUBLE, tableau

Syntaxe :

MCODS(37,-1,<version>,<mémoire tampon>,<taille>)

Tab. 7-135: Syntaxe MCODES(37,...)

MCODS(38,...) – Valeurs réelles de l'axe dans le système de coordonnées de la pièce à usiner

Description : Fournit dans la <mémoire tampon> les "valeurs réelles" par rapport au système de coordonnées (cartésiennes) de la pièce à usiner du <canal>.

Les valeurs sont calculées à partir des positions réelles des axes au moyen de la transformation en avant cinématique spécifique à la machine et spécifiée par canal (transformation d'axe). Si aucune transformation cinématique d'axe n'est programmée, MCODES(38) fournit des valeurs identiques à celles de MCODES(147).

Pour pouvoir utiliser MCODES(38), le paramètre machine 9030 00002, qui permet de configurer si les valeurs réelles du système de coordonnées de la pièce à usiner sont calculées et à quelle fréquence elles le sont, doit être réglé en conséquence.

Mise à jour : Cyclique

Type de données de la <mémoire tampon> : DOUBLE, tableau

Syntaxe :

MCODS(38,<canal>,<version>,<mémoire tampon>,<taille>)

Tab. 7-136: Syntaxe MCODES(38,...)

MCODS(39,...) – Temporisation active

Description : Fournit dans la <mémoire tampon> la valeur 1 si une temporisation est active sur le <canal>. Sinon, fournit la valeur 0.

Mise à jour : Après modification

Type de données de la <mémoire tampon> : INTEGER

Syntaxe :

MCODS(39,<canal>,<version>,<mémoire tampon>,1)

Tab. 7-137: Syntaxe MCODES(39,...)

MCODS(40,...) – Fonction auxiliaire avec acquittement obligatoire active

Description : Fournit dans la <mémoire tampon> la valeur 1 si une fonction auxiliaire attend un acquittement sur le <canal>. Sinon, fournit la valeur 0.

Mise à jour : Après modification

Type de données de la <mémoire tampon> : INTEGER

Syntaxe :

MCODS(40,<canal>,<version>,<mémoire tampon>,1)

Tab. 7-138: Syntaxe MCODES(40,...)

MCODS(41,...) – Validation d'enregistrement

Description : Fournit dans la <mémoire tampon> la valeur 1 si le signal d'entrée CN "Blocage d'enregistrement" est défini sur le <canal>. Sinon, fournit la valeur 0.

Mise à jour : Après modification

Type de données de la <mémoire tampon> : INTEGER

Syntaxe :

MCODS(41,<canal>,<version>,<mémoire tampon>,1)

Tab. 7-139: Syntaxe MCODES(41,...)

Ordres CPL

MCODS(42,...) – Données spécifiques au client

Description : Fournit dans la <mémoire tampon> les données spécifiques au client issues du <canal>.

Lors de l'appel de l'ordre, il est possible de transférer une valeur INTEGER dans la plage de 0 à 65 535 au serveur client pour sélectionner certaines données.

L'ordre est prévu pour les développements propres spécifiques au client au niveau du "noyau CN".

Mise à jour : Après modification

Type de données de la <mémoire tampon> : En fonction du type de données spécifiques au client

Syntaxe : **MCODS(42, <canal>, <version>, <mémoire tampon>, <taille>, <P1>)**

Tab. 7-140: Syntaxe MCODES(42,...)

MCODS(43,...) – Attribution d'axe : canal

Description : Fournit dans la <mémoire tampon> les informations suivantes pour chaque axe du système :

- ≥ 0 :** Numéro du canal de l'axe synchrone
- 1 :** L'axe est asynchrone.
- 2 :** L'axe est une broche.
- 3 :** L'axe n'est pas défini.

Mise à jour : Après modification

Type de données de la <mémoire tampon> : INTEGER, tableau

Syntaxe : **MCODS(43, -1, <version>, <mémoire tampon>, <taille>)**

Tab. 7-141: Syntaxe MCODES(43,...)

MCODS(44,...) – Nombre de canaux

Description : Fournit dans la <mémoire tampon>, par ordre croissant :

- le nombre de canaux utilisateurs pouvant être utilisés ;
- le nombre de canaux sur l'interface ;
- le nombre de tous les canaux internes et externes.

Mise à jour : Après la montée en régime de la commande

Type de données de la <mémoire tampon> : INTEGER, tableau

Syntaxe : **MCODS(44, -1, <version>, <mémoire tampon>, 3)**

Tab. 7-142: Syntaxe MCODES(44,...)

MCODS(45,...) – Nombre d'axes

Description : Fournit dans la <mémoire tampon>, par ordre croissant :

- L'index maximal de tous les entraînements (axes et broches) dans le système. Il est requis pour le ponçage sur tous les entraînements. S'il n'existe aucune lacune, il s'agit du nombre de tous les entraînements disponibles dans le système.
- L'index maximal des axes dans le système. Il est requis, par exemple, pour la taille de l'interface d'axe. S'il n'existe aucune lacune, il s'agit du nombre d'axes existant dans le système.

	Le nombre maximal de broches dans le système. Il est requis, par exemple, pour la taille de l'interface de la broche.
Mise à jour :	Après la montée en régime de la commande
Type de données de la <mémoire tampon> :	INTEGER, tableau
Syntaxe :	MCODS(45,-1,<version>,<mémoire tampon>,3)

Tab. 7-143: Syntaxe MCODES(45,...)

MCODS(46,...) – Stratégie de redémarrage et enregistrement des déplacements en mode Jog

Description :	Fournit dans la <mémoire tampon> les éléments suivants concernant le <canal> indiqué, par ordre croissant : - le mode de fonctionnement de redémarrage, - le point de redémarrage, - l'état d'enregistrement des mouvements JOG. Les valeurs suivantes sont possibles pour le mode de fonctionnement de redémarrage : 1 : Redémarrage automatique 2 : Redémarrage avec bloc unique 3 : Redémarrage manuel Pour le point de redémarrage : 1 : Redémarrage au point de départ 2 : Redémarrage au point d'arrivée 3 : Redémarrage au point d'interruption Pour l'état d'enregistrement : 0 : Enregistrement inactif 1 : Enregistrement actif
Mise à jour :	Après modification
Type de données de la <mémoire tampon> :	INTEGER, modification
Syntaxe :	MCODS(46,<canal>,<version>,<mémoire tampon>,3)

Tab. 7-144: Syntaxe MCODES(46,...)

MCODS(47,...) – Instruction de déplacement

Description :	Si <canal> = -1, la <mémoire tampon> fournit les signaux d'instruction de déplacement de tous les axes du système, par ordre croissant. Si <canal> = numéro du canal existant, la <mémoire tampon> fournit les signaux d'instruction de déplacement de tous les axes du canal indiqué, puis de tous les axes asynchrones, par ordre croissant. Instruction de déplacement initialisée : 1 Instruction de déplacement non initialisée : 0 Une instruction de déplacement est toujours définie dès qu'un axe doit exécuter un mouvement de déplacement défini manuellement ou dans le programme pièce.
Mise à jour :	Cyclique
Type de données de la <mémoire tampon> :	INTEGER, tableau

Ordres CPL

Syntaxe :

MCODS(47,<canal>,<version>,<mémoire tampon>,<taille>)

Tab. 7-145: Syntaxe MCODES(47,...)

MCODS(48,...) – Mode de fonctionnement de l'axe

Description : Si <canal> = -1, la <mémoire tampon> fournit les modes de fonctionnement de tous les axes du système, par ordre croissant.

Si <canal> = numéro du canal existant, la <mémoire tampon> fournit les modes de fonctionnement de tous les **axes du canal indiqué, puis de tous les axes asynchrones**, par ordre croissant.

Valeurs de retour possibles pour les modes de fonctionnement :

0 :	Aucun mode de fonctionnement, et donc aucun processus, n'est actif.
1 :	Mode jog Les axes peuvent être déplacés en mode Jog (+/-).
2 :	Approche du point de référence Les axes peuvent être démarrés à l'aide des signaux Manuel + / Manuel-.
3 :	Réservé
4 :	Saisie manuelle Les blocs CN individuels peuvent être affectés au traitement.
5 :	Automatique (Bloc continu) Les programmes pièces sont exécutés entièrement.
6 :	Automatique (Bloc-programme) Les blocs individuels d'un programme pièce sont traités les uns après les autres. Chaque bloc individuel est préparé et lancé avec le démarrage CN.
7 :	Automatique (Étape unique) Dans certaines conditions, la commande numérique génère et prépare plusieurs blocs à partir d'un bloc CN individuel du programme pièce. Avec ce mode de fonctionnement, le démarrage CN transmet toujours un bloc individuel à l'interpolateur pour être traité.
8 :	Réservé
9 :	Réservé
10 :	Automatique (Bloc unique) Tous les blocs générés et préparés à partir d'un bloc CN individuel du programme pièce sont transmis à l'interpolateur via le démarrage CN pour être traités.
11 :	Redémarrage Les axes peuvent être éloignés manuellement du contour, et rapprochés de lui automatiquement ou manuellement.

Mise à jour : Après modification

Type de données de la <mémoire tampon> : INTEGER, tableau

Syntaxe :

MCODS(48,<canal>,<version>,<mémoire tampon>,<taille>)

Tab. 7-146: Syntaxe MCODES(48,...)

MCODS(49,...) – Blocage d'avance

Description :	Si <canal> = -1, la <mémoire tampon> fournit les signaux des blocages d'avance de tous les axes du système, par ordre croissant. Si <canal> = numéro du canal existant, la <mémoire tampon> fournit les blocages d'avance de tous les axes du canal indiqué, puis de tous les axes asynchrones, par ordre croissant. 1 : Blocage d'avance actif 0 : Blocage d'avance inactif
Mise à jour :	Après modification
Type de données de la <mémoire tampon> :	INTEGER, tableau
Syntaxe :	MCODS(49,<canal>,<version>,<mémoire tampon>,<taille>)

Tab. 7-147: Syntaxe MCODES(49,...)

MCODS(50,...) – Potentiomètre d'axe

Description :	Si <canal> = -1, la <mémoire tampon> fournit les valeurs des Overrides pertinents de chaque axe (potentiomètre d'axe ou de canal) pour tous les axes du système, par ordre croissant (en 0,01 %). Si <canal> = numéro du canal existant, la <mémoire tampon> fournit la valeur du potentiomètre du canal, puis les valeurs des potentiomètres de tous les axes asynchrones pour chaque axe sur le canal indiqué, par ordre croissant (en 0,01 %).
Mise à jour :	Après modification
Type de données de la <mémoire tampon> :	REAL, tableau
Syntaxe :	MCODS(50,<canal>,<version>,<mémoire tampon>,<taille>)

Tab. 7-148: Syntaxe MCODES(50,...)

MCODS(51,...)

Description : Sans pertinence pour la MTX

MCODS(52,...)

Description : Sans pertinence pour la MTX

MCODS(53,...) – Unité de mesure des axes

Description :

Si <canal> = -1, la <mémoire tampon> fournit les unités de mesure (métriques, pouces, degrés) de tous les axes du système, par ordre croissant.

Si <canal> = numéro du canal existant, la <mémoire tampon> fournit l'unité de mesure de chaque **axe du canal indiqué, puis de tous les axes asynchrones**, par ordre croissant.

Concernant les axes linéaires asynchrones aux modes de fonctionnement de l'axe "JOG" ou "Approche du point de référence", l'unité de mesure est définie par l'interface de l'axe. Si aucun mode de fonctionnement d'axe n'est défini, les valeurs sont fournies selon le "système métrique".

Concernant les axes linéaires synchrones aux modes de fonctionnement du canal "JOG" et "Approche du point de référence", l'unité de mesure est définie par l'interface de l'axe. Aux autres modes de fonctionnement, elle dépend de l'unité de mesure du canal (métrique/pouce – G70/G71).

Ordres CPL

Concernant les axes ronds et les broches, les valeurs sont fournies en degrés ; concernant les axes Hirth à programmation d'emplacement, les valeurs sont fournies dans une unité de mesure correspondante.

Valeurs de retour possibles pour les unités de mesure :

- 0 : En pouces
- 1 : Système métrique
- 2 : En degrés
- 3 : Axe inexistant
- 4 : Axe Hirth à programmation d'emplacement

Mise à jour : Après modification

Type de données de la <mémoire tampon> : INTEGER, tableau

Syntaxe : **MCODS(53,<canal>,<version>,<mémoire tampon>,<taille>)**

Tab. 7-149: Syntaxe MCODES(53,...)

MCODS(54,...) – Correction d'outil active

Description : Fournit dans la <mémoire tampon> les valeurs de la correction d'outil actives sur le <canal>.

Ordre :

- Correction du rayon
- Correction de la longueur L3
- Correction de la longueur L1
- Correction de la longueur L2
- position de l'arête coupante
- Type de correction

Les types de correction suivants sont définis :

- 0 : Aucune correction
- 1 : Outil de perçage
- 2 : Outil de fraisage
- 3 : Outil de tournage
- 4 : Outil d'équerrage

Mise à jour : Après un changement de bloc

Type de données de la <mémoire tampon> : REAL, tableau

Syntaxe : **MCODS(54,<canal>,<version>,<mémoire tampon>,<taille>)**

Tab. 7-150: Syntaxe MCODES(54,...)

MCODS(55,...) – Version du fabricant

Description : Indique la version du fabricant de l'entraînement.

L'axe est sélectionné en indiquant l'axe du système dans le paramètre <numéro d'axe> (n° 0 fournit également le 1er axe). La version du fabricant correspond au paramètre Sercos S-0-0030. Le paramètre <mémoire tampon> fournit un tableau de max. 40 caractères.

Mise à jour : Après la montée en régime de la commande

Type de données de la <mémoire tampon> : CHARACTER, tableau

Syntaxe : **MCODS(55,-1,<version>,<mémoire tampon>,<taille>,<numéro d'axe>)**

Tab. 7-151: Syntaxe MCODES(55,...)

MCODS(56,...) – Type de variateur

Description : Indique le type de variateur de l'entraînement.

L'axe est sélectionné en indiquant l'axe du système dans le paramètre <numéro d'axe> (n° 0 fournit également le 1er axe). Le type de variateur correspond au paramètre Sercos S-0-0140. Le paramètre <mémoire tampon> fournit un tableau de max. 40 caractères.

Mise à jour : Après la montée en régime de la commande

Type de données de la <mémoire tampon> : CHARACTER, tableau

Syntaxe : **MCODS(56,-1,<version>,<mémoire tampon>,<taille>,<numéro d'axe>)**

Tab. 7-152: Syntaxe MCODES(56,...)

MCODS(57,...) – Type de moteur

Description : Indique le type de moteur de l'entraînement.

La sélection d'axe s'effectue en indiquant l'axe du système dans le paramètre <numéro d'axe> (n° 0 fournit également le 1er axe). Le type de moteur correspond au paramètre Sercos S-0-0141. Le paramètre <mémoire tampon> fournit un tableau de max. 40 caractères.

Mise à jour : Après la montée en régime de la commande

Type de données de la <mémoire tampon> : CHARACTER, tableau

Syntaxe : **MCODS(57,-1,<version>,<mémoire tampon>,<taille>,<numéro d'axe>)**

Tab. 7-153: Syntaxe MCODES(57,...)

MCODS(58,...) – Attribution par défaut de l'axe : canal

Description : Fournit dans la <mémoire tampon> l'attribution par défaut suivante pour chaque axe du système :

≥ 0 : Numéro du canal de l'axe synchrone

-1 : L'axe est asynchrone.

-2 : L'axe est une broche.

-3 : L'axe n'est pas défini.

S'il existe 250 axes, la <mémoire tampon> doit avoir la <taille> 250 (INTEGER).

Mise à jour : Après une montée en régime

Type de données de la <mémoire tampon> : INTEGER

Syntaxe : **MCODS(58,-1,<version>,<mémoire tampon>,<taille>)**

Tab. 7-154: Syntaxe MCODES(58,...)

MCODS(59,...) – Noms actifs des axes du canal

Description : Si <canal> = -1, la <mémoire tampon> fournit les noms à 9 octets de tous les **axes actifs du canal** avec les index d'axes du système 1 à 100, par ordre croissant et en les séparant par le caractère "0" (octet 0).

Ordres CPL

Si <canal> = -2, la <mémoire tampon> fournit les noms à 9 octets de tous les **axes actifs du canal** avec les index d'axes du système 101 à 200, par ordre croissant et en les séparant par le caractère "0" (octet 0).

Si <canal> = -1, la <mémoire tampon> fournit les noms à 9 octets de tous les **axes actifs du canal** avec les index d'axes du système 201 à 250, par ordre croissant et en les séparant par le caractère "0" (octet 0).

Si <canal> = numéro du canal existant, la <mémoire tampon> fournit les noms de tous les **axes du canal indiqué**, par ordre croissant et en les séparant avec le caractère "0" (octet 0).

Les noms comportant moins de 8 caractères sont complétés avec des espaces jusqu'au 8e caractère.

<taille> indique la capacité de la <mémoire tampon>. Pour 100 axes, elle peut comporter (9*100) 900 octets au maximum.

Pour obtenir un exemple, voir [chap. " Exemples de programmation" à la page 585](#).

Mise à jour : Après modification

Type de données de la <mémoire tampon> : CHARACTER, tableau

Syntaxe :

MCODS(59,<canal>,<version>,<mémoire tampon>,<taille>)

Tab. 7-155: Syntaxe MCODES(59,...)

MCODS(60,...) – Réglage par défaut des noms des axes du canal

Description : Si <canal> = -1, la <mémoire tampon> fournit les noms à 9 octets des axes du canal (dans le réglage par défaut) avec les index des axes du système 1 à 100, par ordre croissant et en les séparant par le caractère "0" (octet 0).

Si <canal> = -2, la <mémoire tampon> fournit les noms à 9 octets des axes du canal (dans le réglage par défaut) avec les index des axes du système 101 à 200, par ordre croissant et en les séparant par le caractère "0" (octet 0).

Si <canal> = -3, la <mémoire tampon> fournit les noms à 9 octets des axes du canal (dans le réglage par défaut) avec les index des axes du système 201 à 250, par ordre croissant et en les séparant par le caractère "0" (octet 0).

Si <canal> = numéro du canal existant, la <mémoire tampon> fournit les noms de tous les **axes du canal indiqué**, par ordre croissant et en les séparant avec le caractère "0" (octet 0).

Les noms comportant moins de 8 caractères sont complétés avec des espaces jusqu'au 8e caractère.

<taille> indique la capacité de la <mémoire tampon>. Pour 100 axes, elle peut comporter au maximum (9*100) 900 octets.

Pour obtenir un exemple, voir [chap. " Exemples de programmation" à la page 585](#).

Mise à jour : Après la montée en régime de la commande

Type de données de la <mémoire tampon> : CHARACTER, tableau

Syntaxe :

MCODS(60,<canal>,<version>,<mémoire tampon>,<taille>)

Tab. 7-156: Syntaxe MCODES(60,...)

MCODS(61,...) – Réglage par défaut des unités de mesure des axes

Description : Si <canal> = -1, la <mémoire tampon> fournit les unités de mesure (métrique, pouces, degrés) de tous les axes du système dans le réglage par défaut.

Si <canal> = numéro du canal existant, la <mémoire tampon> fournit les unités de mesure (métrique, pouces, degrés) de tous les **axes sur le canal indiqué** dans le réglage par défaut.

- Concernant les axes linéaires **asynchrones**, les valeurs sont fournies dans le "système métrique".
- Concernant les axes linéaires **synchrones**, l'unité de mesure dépend de l'état de mise en service après la montée en régime (paramètre CHAN/Ch/Ini/NcResetState (MP 7060 00010)) : "métrique/pouces" (G70/G71)
- Concernant les axes ronds, les valeurs sont fournies en degrés ; concernant les axes Hirth à programmation d'emplacement, elles sont fournies dans une unité de mesure correspondante.

Valeurs de retour possibles pour les unités de mesure :

0 : En pouces
 1 : Système métrique
 2 : En degrés
 3 : Axe inexistant
 4 : Axe Hirth à programmation d'emplacement
 S'il existe 250 axes, la <mémoire tampon> doit avoir la <taille> 250 (INTEGER).

Mise à jour : Après la montée en régime de la commande

Type de données de la <mémoire tampon> : INTEGER, tableau

Syntaxe : MCODS(61,<canal>,<version>,<mémoire tampon>,<taille>)

Tab. 7-157: Syntaxe MCODES(61,...)

MCODS(62,...) – Numéro d'identification issu du télégramme d'axe cyclique

Description : Fournit la valeur d'un <numéro d'identification> à partir du télégramme d'axe cyclique pour tous les axes.

Si le <numéro d'identification> n'est pas contenu dans le télégramme cyclique, la valeur NCS_MCO_NOT_IN_CYCL_AT_C (-2147483648) est indiquée. La valeur est de type "INTEGER" en calibrage Sercos.

S'il existe 250 axes, la <mémoire tampon> doit avoir la <taille> 250 (INTEGER).

Mise à jour : Cyclique

Type de données de la <mémoire tampon> : INTEGER, tableau

Syntaxe : MCODS(62,-1,<version>,<mémoire tampon>,<taille>,<numéro d'identification>)

Tab. 7-158: Syntaxe MCODES(62,...)

MCODS(63,...) – Fonctions de mouvement des broches

Description : Fournit les fonctions de mouvement des broches.

Codage des fonctions de mouvement :

0 : Broche non définie

Ordres CPL

- 1 : Rotation vers la droite sans liquide de refroidissement
 - 2 : Rotation vers la droite avec liquide de refroidissement
 - 3 : Rotation vers la gauche sans liquide de refroidissement
 - 4 : Rotation vers la gauche avec liquide de refroidissement
 - 5 : Arrêt de la broche
 - 6 : Orientation de la broche
- S'il existe 32 broches, par exemple, la <mémoire tampon> doit avoir la <taille> 32 (INTEGER).
- Si <canal> = -1 ou 0 est transféré, la <mémoire tampon> fournit les données des broches du système ; sinon, elle fournit les données des broches du canal.

Mise à jour : Après modification

Type de données de la <mémoire tampon> : INTEGER, tableau

Syntaxe : **MCODS(63,<canal>,<version>,<mémoire tampon>,<taille>)**

Tab. 7-159: Syntaxe MCODES(63,...)

MCODS(64,...) – Rapports de réduction

Description : Fournit les rapports de réduction des broches.

Codage des rapports de réduction :

- 40 : Sélection automatique des rapports de réduction
- 41 : Rapport de réduction 1
- 42 : Rapport de réduction 2
- 43 : Rapport de réduction 3
- 44 : Rapport de réduction 4
- 48 : Débrayer le rapport de réduction

S'il existe 32 broches, par exemple, la <mémoire tampon> doit avoir la <taille> 32 (INTEGER).

Si <canal> = -1 ou 0 est transféré, la <mémoire tampon> fournit les données des broches du système ; sinon, elle fournit les données des broches du canal.

Mise à jour : Après modification

Type de données de la <mémoire tampon> : INTEGER, tableau

Syntaxe : **MCODS(64,<canal>,<version>,<mémoire tampon>,<taille>)**

Tab. 7-160: Syntaxe MCODES(64,...)

MCODS(65,...) – Attribution des broches du canal à des groupes

Description : Fournit l'attribution des broches du canal à des groupes :

Attribution de groupes :

- 0 : aucun groupe actif
- 1 à 4 : numéro de groupe

S'il existe 8 broches du canal, par exemple, la <mémoire tampon> doit avoir la <taille> 8 (INTEGER).

Si <canal> = -1 est transféré, la <mémoire tampon> fournit les données du canal actif ; sinon, elle fournit les données du canal sélectionné.

Mise à jour : Après modification

Type de données de la <mémoire tampon> : INTEGER, tableau

Syntaxe :

MCODS(65,<canal>,<version>,<mémoire tampon>,<taille>)

Tab. 7-161: Syntaxe MCODES(65,...)

MCODS(66,...) – Commutation des rapports de réduction

Description : Informe si la commutation des rapports de réduction est active :

- 0 : commutation des rapports de réduction inactive
- 1 : commutation des rapports de réduction active

S'il existe 32 broches, par exemple, la <mémoire tampon> doit avoir la <taille> 32 (INTEGER).

Si <canal> = -1 ou 0 est transféré, la <mémoire tampon> fournit les données des broches du système ; sinon, elle fournit les données des broches du canal.

Mise à jour : Après modification

Type de données de la <mémoire tampon> : INTEGER, tableau

Syntaxe :

MCODS(66,<canal>,<version>,<mémoire tampon>,<taille>)

Tab. 7-162: Syntaxe MCODES(66,...)

MCODS(67,...) – Commutation des rapports de réduction auto/manuelle

Description : Indique si les rapports de réduction sont sélectionnés automatiquement ou manuellement :

- 0 : manuellement
- 1 : automatiquement

S'il existe 32 broches, par exemple, la <mémoire tampon> doit avoir la <taille> 32 (INTEGER).

Si <canal> = -1 ou 0 est transféré, la <mémoire tampon> fournit les données des broches du système ; sinon, elle fournit les données des broches du canal.

Mise à jour : Après modification

Type de données de la <mémoire tampon> : INTEGER, tableau

Syntaxe :

MCODS(67,<canal>,<version>,<mémoire tampon>,<taille>)

Tab. 7-163: Syntaxe MCODES(67,...)

MCODS(68,...) – Sauter séquence

Description : Fournit dans la <mémoire tampon> l'état du signal de sortie CN **Activer sauter séquence** et du signal d'entrée **Sauter séquence** du <canal>.

Mise à jour : Après modification

Type de données de la <mémoire tampon> : INTEGER, tableau

Syntaxe :

MCODS(68,<canal>,<version>,<mémoire tampon>,2)

Tab. 7-164: Syntaxe MCODES(68,...)

MCODS(69,...) – Arrêt optionnel

Description : Fournit dans la <mémoire tampon> l'état du signal de sortie CN **Arrêt optionnel activé** et du signal d'entrée **Arrêt optionnel** du <canal>.

Ordres CPL

Mise à jour : Après modification
 Type de données de la <mémoire tampon> : INTEGER, tableau
 Syntaxe :

MCODS(69,<canal>,<version>,<mémoire tampon>,2)

Tab. 7-165: Syntaxe MCODES(69,...)

MCODS(70,...) – Resélection automatique du programme active

Description : Indique dans la <mémoire tampon> si Resélection automatique du programme est appliquée sur le canal indiqué :

0 : La fonction n'est pas appliquée.
 1 : La fonction est appliquée.

Mise à jour : Immédiatement après la requête
 Type de données de la <mémoire tampon> : INTEGER
 Syntaxe :

MCODS(70,<canal>,<version>,<mémoire tampon>,1)

Tab. 7-166: Syntaxe MCODES(70,...)

MCODS(71,...) – Coordonnées de la pièce à usiner

Description : Fournit dans la <mémoire tampon> les valeurs des **coordonnées de la pièce à usiner** (WCS) du canal indiqué : d'abord toutes les coordonnées tridimensionnelles, puis les pseudo-coordonnées du canal.

Mise à jour : Cyclique
 Type de données de la <mémoire tampon> : DOUBLE, tableau
 Syntaxe :

MCODS(71,<canal>,<version>,<mémoire tampon>,<taille>)

Tab. 7-167: Syntaxe MCODES(71,...)

MCODS(72,...) – Valeurs de consigne des coordonnées de base

Description : Fournit dans la <mémoire tampon> les **valeurs de consigne des coordonnées de base** (BCS) du canal indiqué : d'abord toutes les coordonnées tridimensionnelles, puis les pseudo-coordonnées du canal.

Mise à jour : Cyclique
 Type de données de la <mémoire tampon> : REAL, tableau
 Syntaxe :

MCODS(72,<canal>,<version>,<mémoire tampon>,<taille>)

Tab. 7-168: Syntaxe MCODES(72,...)

MCODS(73,...) – Coordonnées d'axe dans l'ACS1

Description : Fournit dans la <mémoire tampon> les **valeurs des coordonnées d'axe** (ACS1) du canal indiqué.

<canal> = -1 : toutes les coordonnées d'axe
 0 < <canal> ≤ Canal max. : données du canal défini

Mise à jour : Cyclique
 Type de données de la <mémoire tampon> : REAL, tableau
 Syntaxe :

MCODS(73,<canal>,<version>,<mémoire tampon>,<taille>)

Tab. 7-169: Syntaxe MCODES(73,...)

MCODS(74,...) – Coordonnées de machine

Description :	Fournit dans la <mémoire tampon> les valeurs des coordonnées de machine (MCS) du canal indiqué. <canal> = -1 : toutes les coordonnées d'axe 0 < <canal> ≤ Canal max. : données du canal défini
Mise à jour :	Cyclique
Type de données de la <mémoire tampon> :	REAL, tableau
Syntaxe :	MCODS(74,<canal>,<version>,<mémoire tampon>,<taille>)

Tab. 7-170: Syntaxe MCODES(74,...)

MCODS(75,...) – Valeurs réelles des coordonnées de base

Description :	Fournit dans la <mémoire tampon> les valeurs réelles des coordonnées de base (BCS) du canal indiqué : d'abord toutes les coordonnées tridimensionnelles, puis les pseudo-coordonnées du canal.
Mise à jour :	Cyclique
Type de données de la <mémoire tampon> :	REAL, tableau
Syntaxe :	MCODS(75,<canal>,<version>,<mémoire tampon>,<taille>)

Tab. 7-171: Syntaxe MCODES(75,...)

MCODS(76,...) – Points d'arrivée programmés des coordonnées

Description :	Fournit dans la <mémoire tampon> les points d'arrivée programmés des coordonnées du canal indiqué : d'abord toutes les coordonnées tridimensionnelles, puis les pseudo-coordonnées du canal.
Mise à jour :	Après chaque changement de bloc
Type de données de la <mémoire tampon> :	REAL, tableau
Syntaxe :	MCODS(76,<canal>,<version>,<mémoire tampon>,<taille>)

Tab. 7-172: Syntaxe MCODES(76,...)

MCODS(77,...) – Points d'arrivée des coordonnées

Description :	Fournit dans la <mémoire tampon> les points d'arrivée des coordonnées du canal indiqué, y compris les décalages : d'abord toutes les coordonnées tridimensionnelles, puis les pseudo-coordonnées du canal.
Mise à jour :	Après chaque changement de bloc
Type de données de la <mémoire tampon> :	REAL, tableau
Syntaxe :	MCODS(77,<canal>,<version>,<mémoire tampon>,<taille>)

Tab. 7-173: Syntaxe MCODES(77,...)

MCODS(78,...) – Noms des coordonnées

Description :	Fournit dans la <mémoire tampon> les noms des coordonnées actives du canal indiqué : d'abord toutes les coordonnées tridimensionnelles, puis les pseudo-coordonnées du canal. <taille> indique la capacité de la <mémoire tampon>. S'il existe 8 coordonnées, elle peut comporter (8*9) 72 octets au maximum.
Mise à jour :	Après modification

Ordres CPL

Type de données de la <mémoire tampon> : CHARACTER, tableau

Syntaxe :

MCODS(78,<canal>,<version>,<mémoire tampon>,<taille>)

Tab. 7-174: Syntaxe MCODES(78,...)

MCODS(79,...) – État INPOS pour les coordonnées

Description : Fournit dans la <mémoire tampon> l'**état INPOS des coordonnées** du canal indiqué : d'abord toutes les coordonnées tridimensionnelles, puis les pseudo-coordonnées du canal.

Pour une coordonnée tridimensionnelle, l'état est constitué à partir de la liaison ET des signaux des axes.

Mise à jour : Cyclique

Type de données de la <mémoire tampon> : INTEGER, tableau

Syntaxe :

MCODS(79,<canal>,<version>,<mémoire tampon>,<taille>)

Tab. 7-175: Syntaxe MCODES(79,...)

MCODS(80,...) – Référence des coordonnées

Description : Fournit dans la <mémoire tampon> l'**état du point de référence des coordonnées** du canal indiqué : d'abord toutes les coordonnées tridimensionnelles, puis les pseudo-coordonnées du canal.

Pour une coordonnée tridimensionnelle, l'état est constitué à partir de la liaison ET des signaux des axes.

Mise à jour : Après modification

Type de données de la <mémoire tampon> : INTEGER, tableau

Syntaxe :

MCODS(80,<canal>,<version>,<mémoire tampon>,<taille>)

Tab. 7-176: Syntaxe MCODES(80,...)

MCODS(81,...) – Unité de mesure des coordonnées

Description : Fournit dans la <mémoire tampon> les **unités de mesure des coordonnées** du canal indiqué : d'abord toutes les coordonnées tridimensionnelles, puis les pseudo-coordonnées du canal.

Valeurs de retour possibles pour les unités de mesure :

- 0 : En pouces
- 1 : Système métrique
- 2 : En degrés
- 3 : Coordonnée inexistante

Mise à jour : Après modification

Type de données de la <mémoire tampon> : INTEGER, tableau

Syntaxe :

MCODS(81,<canal>,<version>,<mémoire tampon>,<taille>)

Tab. 7-177: Syntaxe MCODES(81,...)

MCODS(82,...) – Nombre de coordonnées

Description : Fournit dans la <mémoire tampon> le **nombre de coordonnées/d'axes** du canal indiqué en 3 éléments :

- 1re valeur : nombre total des axes du canal ;
- 2e valeur : nombre de coordonnées tridimensionnelles + nombre de pseudo-coordonnées du canal ;
- 3e valeur : nombre de pseudo-coordonnées du canal.

Mise à jour : Après modification

Type de données de la <mémoire tampon> : INTEGER, tableau

Syntaxe : `MCODS(82,<canal>,<version>,<mémoire tampon>,<taille>)`

Tab. 7-178: Syntaxe MCODES(82,...)

MCODS(83,...) – Course restante des coordonnées dans le système de coordonnées de la pièce à usiner (WCS)

Description : Fournit dans la <mémoire tampon> les **courses restantes des coordonnées de la pièce à usiner** (WCS) du canal indiqué : d'abord toutes les coordonnées tridimensionnelles, puis les pseudo-coordonnées du canal.

Mise à jour : Cyclique

Type de données de la <mémoire tampon> : REAL, tableau

Syntaxe : `MCODS(83,<canal>,<version>,<mémoire tampon>,<taille>)`

Tab. 7-179: Syntaxe MCODES(83,...)

MCODS(84,...)

Description : Sans pertinence pour la MTX

MCODS(85,...) – Course restante des axes dans le système de coordonnées d'axe (ACS)

Description : Fournit dans la <mémoire tampon> les **courses restantes des axes dans le systèmes des coordonnées de l'axe** (ACS) du canal indiqué.

Mise à jour : Cyclique

Type de données de la <mémoire tampon> : REAL, tableau

Syntaxe : `MCODS(85,<canal>,<version>,<mémoire tampon>,<taille>)`

Tab. 7-180: Syntaxe MCODES(85,...)

MCODS(86,...)

Description : Sans pertinence pour la MTX

MCODS(87,...) – États d'attente du canal

Description : Fournit dans la <mémoire tampon> les **états d'attente** d'un canal.

Si un canal passe à un état d'attente, cet ordre permet d'en indiquer les motifs.

Les états d'attente actifs sont codés en bits.

Les constantes suivantes définissent les bits correspondants de la première valeur INTEGER, en commençant par la valeur la plus faible :

0 : Temporisation

1 : Fonction auxiliaire avec acquittement obligatoire

Ordres CPL

- 2 : Blocage d'enregistrement
 - 3 : Avance sur le canal égale à 0
 - 4 : Arrêt du programme avec M0/M1
 - 5 : Arrêt de l'avance sur le canal
 - 6 : Blocage d'avance sur le canal ou d'un axe du canal
 - 7 : Blocage d'enregistrement défini par le client
 - 8 : Arrêt de mouvement synchronisé entre canaux (ASTOP, etc.)
 - 9 : Attente de l'axe au cours d'un changement d'axes (WAX)
 - 10 : Attente de la variable permanente (WPV)
 - 11 : Attente du signal d'interface au moment actif (WAITA, etc.)
 - 12 : Attente du signal d'interface (WAIT(BITIF...)) ou pendant un laps de temps déterminé (WAIT(,TEMPS%)) lors de la préparation des blocs
 - 13 : Service de données Motion Control (MCODS(...))
 - 14 : Attente de la position INPOS des axes du canal
 - 15 : G95 : attente de la vitesse de rotation de la broche ou de l'axe rond
 - 16 : G33 : attente de la vitesse de rotation de la broche
 - 17 : Aucune vitesse de redémarrage, par ex. vitesse de Jog
 - 18 : Attente d'un mouvement de la broche
 - 19 : Événement programmable (SEV / WEV)
 - 20 : Attente de la libération des ressources internes (mémoire)
 - 21 : G63 : attente de la commutation d'axe C
- Voir l'exemple au [chap. "Exemples de programmation"](#) à la page 585.

Mise à jour : Après modification

Type de données de la <mémoire tampon> : INTEGER

Syntaxe : **MCODS(87,<canal>,<version>,<mémoire tampon>,1)**

Tab. 7-181: Syntaxe MCODES(87,...)

MCODS(88,...)

Description : Sans pertinence pour la MTX

MCODS(89,...) – Valeurs de correction en ligne (WCS)

Description : Fournit dans la <mémoire tampon> les valeurs de la correction en ligne (WCS) du canal indiqué : d'abord toutes les coordonnées tridimensionnelles, puis les pseudo-coordonnées du canal.

Mise à jour : Après modification

Type de données de la <mémoire tampon> : REAL, tableau

Syntaxe : **MCODS(89,<canal>,<version>,<mémoire tampon>,<taille>)**

Tab. 7-182: Syntaxe MCODES(89,...)

MCODS(90,...) – État de la correction en ligne (WCS)

Description :	Fournit dans la <mémoire tampon> l'état actuel de la correction en ligne (WCS) du canal indiqué : d'abord toutes les coordonnées tridimensionnelles, puis les pseudo-coordonnées du canal.
0 :	Inactif
1 :	Actif
Mise à jour :	Après modification
Type de données de la <mémoire tampon> :	INTEGER, tableau
Syntaxe :	MCODS(90,<canal>,<version>,<mémoire tampon>,<taille>)

Tab. 7-183: Syntaxe MCODES(90,...)

MCODS(91,...) – Programmation du diamètre

Description :	Fournit dans la <mémoire tampon> la valeur 1 pour les axes avec programmation du diamètre. Si la valeur Ncs_MCoNoChannel_C (-1) est affectée en tant que canal, alors les valeurs de tous les axes du système sont fournies. Si un numéro de canal est affecté, alors les valeurs de tous les axes du canal sont fournies.
Mise à jour :	Après chaque changement de bloc
Type de données de la <mémoire tampon> :	INTEGER, tableau
Syntaxe :	MCODS(91,<canal>,<version>,<mémoire tampon>,<taille>)

Tab. 7-184: Syntaxe MCODES(91,...)

MCODS(92,...) – Coordonnées de base de la pointe de l'outil (BCS-Tcp)

Description :	Si la correction de rayon 3D est active, fournit dans la <mémoire tampon> les valeurs de consigne de la pointe de l'outil dans le système de coordonnées de base (BCS-Tcp) : d'abord les coordonnées tridimensionnelles, puis les pseudo-coordonnées du canal indiqué. Si la correction de rayon 3D est désactivée, alors les valeurs sont identiques à celles de MCODES(72,...).
Mise à jour :	Cyclique
Type de données de la <mémoire tampon> :	REAL, tableau
Syntaxe :	MCODS(92,<canal>,<version>,<mémoire tampon>,<taille>)

Tab. 7-185: Syntaxe MCODES(92,...)

MCODS(93,...)

Description : Sans pertinence pour la MTX

MCODS(94,...) – Numéros des groupes de fonctions auxiliaires

Description :	Fournit dans <IdGroupe> les numéros des groupes HiFu des fonctions auxiliaires du canal indiqué qui sont indiquées dans la <syntaxe>. La <syntaxe> contient les fonctions auxiliaires, séparées par des espaces et terminées avec CHR\$(0) (voir "Exemple ." à la page 570).
Mise à jour :	Cyclique

Ordres CPL

Type de données de la <mémoire tampon> : CHARACTER, tableau

Syntaxe :

MCODS(94,<canal>,<version>,<IdGroupe>,<taille>,<syntaxe>)

Tab. 7-186: Syntaxe MCODES(94,...)

MCODS(95,...) – Syntaxe active des groupes de fonctions auxiliaires

Description : Fournit dans <syntaxe> la syntaxe active des numéros des groupes de fonctions auxiliaires indiqués dans <IdGroupe> du canal indiqué.

Un tableau contenant les numéros des groupes de fonctions auxiliaires voulus et se terminant en CHR\$(0) est transmis dans <IdGroupe> (voir l'exemple ci-dessous).

Mise à jour : Cyclique

Type de données de la <mémoire tampon> : CHARACTER, tableau

Syntaxe :

MCODS(95,<canal>,<version>,<syntaxe>,<taille>,<IdGroupe>)

Tab. 7-187: Syntaxe MCODES(95,...)

MCODS(96,...)

Description : Sans pertinence pour la MTX

MCODS(97,...)

Description : Sans pertinence pour la MTX

MCODS(98,...)

Description : Sans pertinence pour la MTX

MCODS(99,...) – Données de décalage d'origine actives

Description : Fournit dans la <mémoire tampon> le jeu de données actif pour le décalage d'origine des coordonnées de machine.

Pour chaque coordonnée, cinq valeurs DOUBLE sont fournies comme valeurs de décalage actives des banques DO 1 à 5. Si aucun décalage n'est actif, la valeur 0.0 est renvoyée.

Les données sont enregistrées dans la <mémoire tampon> comme suit :

- Décalage de la 1re coordonnée, banque 1
- Décalage de la 1re coordonnée, banque 2
- ...
- Décalage de la 1re coordonnée, banque 5
- Décalage de la n-ième coordonnée, banque 1
- ...
- Décalage de la n-ième coordonnée, banque 5

Mise à jour : Après modification

Type de données de la <mémoire tampon> : DOUBLE, tableau

Syntaxe :

MCODS(99,<canal>,<version>,<mémoire tampon>,<taille>)

Tab. 7-188: Syntaxe MCODES(99,...)

MCODS(100,...) – Somme des décalages d'origine

Description : Fournit dans la <mémoire tampon> la somme du décalage d'origine actif pour les coordonnées de machine. Une valeur DOUBLE est fournie pour chaque coordonnée.

Si aucun décalage n'est actif, la valeur 0.0 est renvoyée.

Mise à jour : Après modification

Type de données de la <mémoire tampon> : DOUBLE, tableau

Syntaxe :

MCODS(100,<canal>,<version>,<mémoire tampon>,<taille>)

Tab. 7-189: Syntaxe MCODES(100,...)

MCODS(101,...) – Points d'arrivée des coordonnées d'axe

Description : Fournit dans la <mémoire tampon> les **points d'arrivée des coordonnées d'axe** (ACS) du canal indiqué :

0 < <canal> ≤ Canal max. : données du canal défini

Mise à jour : Après chaque changement de bloc

Type de données de la <mémoire tampon> : REAL, tableau

Syntaxe :

MCODS(101,<canal>,<version>,<mémoire tampon>,<taille>)

Tab. 7-190: Syntaxe MCODES(101,...)

MCODS(102,...) – Noms des broches

Description : Fournit dans la <mémoire tampon> les noms des broches actives du canal indiqué. Si <canal> = -1 ou 0 est transféré, la <mémoire tampon> fournit les données des broches du système ; sinon, elle fournit les données des broches du canal.

Mise à jour : Après modification

Type de données de la <mémoire tampon> : CHARACTER, tableau

Syntaxe :

MCODS(102,<canal>,<version>,<mémoire tampon>,<taille>)

Tab. 7-191: Syntaxe MCODES(102,...)

MCODS(103,...) – Nombre de broches

Description : Fournit dans la <mémoire tampon> le nombre de broches du canal indiqué. Si <canal> = -1 ou 0 est transféré, la <mémoire tampon> fournit les données des broches du système ; sinon, elle fournit les données des broches du canal.

Mise à jour : Après modification

Type de données de la <mémoire tampon> : INTEGER

Syntaxe :

MCODS(103,<canal>,<version>,<mémoire tampon>,<taille>)

Tab. 7-192: Syntaxe MCODES(103,...)

Ordres CPL

MCODS(104,...) – Numéro du système de la broche

Description : Fournit dans la <mémoire tampon> les numéros du système des broches (numéro de broche du système), par ordre croissant (S ou S1, S2, S3, etc.). Si <canal> = -1 ou 0, la <mémoire tampon> fournit les données des broches du système ; sinon, elle fournit les données des broches du canal.

Mise à jour : Après modification

Type de données de la <mémoire tampon> : INTEGER, tableau

Syntaxe :

MCODS(104,<canal>,<version>,<mémoire tampon>,<taille>)

Tab. 7-193: Syntaxe MCODES(104,...)

MCODS(105,...) – Numéro d'axe de la broche

Description : Fournit dans la <mémoire tampon> les numéros d'axes des broches, par ordre croissant (S ou S1, S2, S3 etc.). Si <canal> = -1 ou 0 est transféré, la <mémoire tampon> fournit les données des broches du système ; sinon, elle fournit les données des broches du canal.

Mise à jour : Après modification

Type de données de la <mémoire tampon> : INTEGER, tableau

Syntaxe :

MCODS(105,<canal>,<version>,<mémoire tampon>,<taille>)

Tab. 7-194: Syntaxe MCODES(105,...)

MCODS(106,...) – Noms des coordonnées de machine

Description : Fournit dans la <mémoire tampon> les noms des coordonnées de machine actives du canal indiqué : d'abord toutes les coordonnées tridimensionnelles, puis les pseudo-coordonnées du canal dans le système de coordonnées de la machine.

<taille> indique la capacité de la <mémoire tampon>. S'il existe 8 coordonnées, elle peut comporter (8*9) 72 octets au maximum.

Mise à jour : Après modification

Type de données de la <mémoire tampon> : CHARACTER, tableau

Syntaxe :

MCODS(106,<canal>,<version>,<mémoire tampon>,<taille>)

Tab. 7-195: Syntaxe MCODES(106,...)

MCODS(107,...) – Nombre des coordonnées de machine

Description : Fournit dans la <mémoire tampon> le nombre de coordonnées/d'axes du canal indiqué en 3 éléments :

1re valeur : Nombre total des axes du canal

2e valeur : Nombre de coordonnées tridimensionnelles + nombre de pseudo-coordonnées du canal dans le système de coordonnées de la machine

3e valeur : Nombre de pseudo-coordonnées du canal dans le système de coordonnées de la machine

Mise à jour : Après modification

Type de données de la <mémoire tampon> : INTEGER, tableau

Syntaxe :

MCODS(107,<canal>,<version>,<mémoire tampon>,<taille>)

Tab. 7-196: Syntaxe MCODES(107,...)

MCODS(108,...) – Position réelle de la broche

Description : Fournit dans la <mémoire tampon> la position réelle des broches actives pour le canal indiqué.

Si <canal> = -1 ou 0 est transféré, la <mémoire tampon> fournit la position réelle des broches du système ; sinon, elle fournit la position réelle des broches du canal.

Mise à jour : Cyclique

Type de données de la <mémoire tampon> : REAL, tableau

Syntaxe :

MCODS(108,<canal>,<version>,<mémoire tampon>,<taille>)

Tab. 7-197: Syntaxe MCODES(108,...)

MCODS(109,...) – Coordonnées d'axe dans l'ACS0

Description : Fournit dans la <mémoire tampon> les **valeurs des coordonnées d'axe** (ACS0) du canal indiqué.

<canal> = -1 : toutes les coordonnées d'axe

0 < <canal> ≤ Canal max. : données du canal défini

Mise à jour : Cyclique

Type de données de la <mémoire tampon> : REAL, tableau

Syntaxe :

MCODS(109,<canal>,<version>,<mémoire tampon>,<taille>)

Tab. 7-198: Syntaxe MCODES(109,...)

MCODS(110,...) – Couple de broche

Description : Fournit dans la <mémoire tampon> les couples réel et maximal (valeur en pourcentage [%]) des broches actives pour le canal indiqué.

Si <canal> = -1 ou 0 est transféré, la <mémoire tampon> fournit le couple des broches du système, sinon le couple des broches du canal.

Ainsi, deux valeurs sont transférées pour chaque broche.

Voir aussi la documentation "Commande CN standard Rexroth IndraMotion MTX", chap. "Affichage de technologie".

Mise à jour : Cyclique

Type de données de la <mémoire tampon> : DOUBLE, tableau

Syntaxe :

MCODS(110,<canal>,<version>,<mémoire tampon>,<taille>)

Tab. 7-199: Syntaxe MCODES(110,...)

MCODS(111,...) – Puissance de la broche

Description : Fournit dans la <mémoire tampon> les puissances réelle et maximale (valeur en watts [W]) des broches actives pour le canal indiqué.

Si <canal> = -1 ou 0 est transféré, la <mémoire tampon> fournit la puissance des broches du système ; sinon, la puissance des broches du canal.

Ordres CPL

Ainsi, deux valeurs sont transférées pour chaque broche.

Voir aussi la documentation "Commande CN standard Rexroth IndraMotion MTX", chap. "Affichage de technologie".

Mise à jour : Cyclique

Type de données de la <mémoire tampon> : INTEGER, tableau

Syntaxe : **MCODS(111,<canal>,<version>,<mémoire tampon>,<taille>)**

Tab. 7-200: Syntaxe MCODES((111,...))

MCODS(112,...) – Données de placement actives

Description : Fournit dans la <mémoire tampon> le jeu de données actif des placements. Six valeurs DOUBLE sont fournies pour chaque plan ou banque. Si aucun placement n'est actif, la valeur 0.0 est renvoyée. Au total, 30 valeurs DOUBLE sont donc transférées.

Les données sont enregistrées dans la <mémoire tampon> comme suit :

- Banque 1 [X, Y, Z, phi, theta, psi]
- Banque 2 [X, Y, Z, phi, theta, psi]
- ...
- Banque 5 [X, Y, Z, phi, theta, psi]

Mise à jour : Après modification

Type de données de la <mémoire tampon> : DOUBLE, tableau

Syntaxe : **MCODS(112,<canal>,<version>,<mémoire tampon>,<taille>)**

Tab. 7-201: Syntaxe MCODES(112,...)

MCODS(113,...) – Chemin et nom du tableau de placement actif

Description : Fournit dans la <mémoire tampon> le nom du tableau de placement actif sur le <canal>.

Si aucune table n'est active, une chaîne de caractères composée de 3 espaces est renvoyée.

Mise à jour : Après chaque changement de bloc

Type de données de la <mémoire tampon> : CHARACTER, tableau

Syntaxe : **MCODS(113,<canal>,<version>,<mémoire tampon>,<taille>)**

Tab. 7-202: Syntaxe MCODES(113,...)

MCODS(114,...) – Correction de la position de la pièce à usiner

Description : Fournit dans la <mémoire tampon> la correction de position de la pièce à usiner active sur le <canal>.

Six valeurs DOUBLE sont fournies :

[décalage X_w , décalage Y_w , décalage Z_w , phi, theta, psi].

Si aucune correction de la position de la pièce à usiner n'est active, la valeur 0.0 est renvoyée.

Mise à jour : Après modification

Type de données de la <mémoire tampon> : DOUBLE, tableau

Syntaxe :

MCODS(114,<canal>,<version>,<mémoire tampon>,<taille>)

Tab. 7-203: Syntaxe MCODES(114,...)

MCODS(115,...) – Somme des placements

Description : Fournit dans la <mémoire tampon> la somme des placements actifs sur le <canal> et de la correction de position de la pièce à usiner.

Six valeurs DOUBLE sont fournies :

[décalage X_w , décalage Y_w , décalage Z_w , phi, θ , psi].

Si aucune correction de position de la pièce à usiner ni aucun placement ne sont actifs, la valeur 0.0 est renvoyée.

Mise à jour : Après modification

Type de données de la <mémoire tampon> : DOUBLE, tableau

Syntaxe :

MCODS(115,<canal>,<version>,<mémoire tampon>,<taille>)

Tab. 7-204: Syntaxe MCODES(115,...)

MCODS(116,...) – Décalage des coordonnées du programme actif

Description : Fournit dans la <mémoire tampon> le décalage des coordonnées du programme actif sur le <canal>.

Une valeur DOUBLE est fournie pour chaque coordonnée.

Si aucun décalage des coordonnées du programme n'est actif, la valeur 0.0 est renvoyée.

Mise à jour : Après modification

Type de données de la <mémoire tampon> : DOUBLE, tableau

Syntaxe :

MCODS(116,<canal>,<version>,<mémoire tampon>,<taille>)

Tab. 7-205: Syntaxe MCODES(116,...)

MCODS(117,...) – Décalage additif des coordonnées du programme actif

Description : Fournit dans la <mémoire tampon> le décalage additif des coordonnées du programme actif sur le <canal>.

Une valeur DOUBLE est fournie pour chaque coordonnée.

Si aucun décalage des coordonnées du programme n'est actif, la valeur 0.0 est renvoyée.

Mise à jour : Après modification

Type de données de la <mémoire tampon> : DOUBLE, tableau

Syntaxe :

MCODS(117,<canal>,<version>,<mémoire tampon>,<taille>)

Tab. 7-206: Syntaxe MCODES(117,...)

MCODS(118,...) – Somme des décalages des coordonnées du programme actifs

Description : Fournit dans la <mémoire tampon> la somme du décalage des coordonnées du programme actif sur le <canal>.

Une valeur DOUBLE est fournie pour chaque coordonnée.

Si aucun décalage des coordonnées du programme n'est actif, la valeur 0.0 est renvoyée.

Mise à jour : Après modification

Ordres CPL

Type de données de la <mémoire tampon> : DOUBLE, tableau

Syntaxe : MCODS(118,<canal>,<version>,<mémoire tampon>,<taille>)

Tab. 7-207: Syntaxe MCODES(118,...)

MCODS(119,...) – Vitesse actuelle de l'axe

Description : Fournit dans la <mémoire tampon> les **vitesses actuelles de l'axe** pour le canal indiqué.

Si la valeur -1 est affectée à <canal>, les vitesses d'axe de tous les axes du système sont fournies.

Sinon, l'ordre fournit les vitesses actuelles des axes du canal affecté.

Mise à jour : Cyclique

Type de données de la <mémoire tampon> : DOUBLE, tableau

Syntaxe : MCODS(119,<canal>,<version>,<mémoire tampon>,<taille>)

Tab. 7-208: Syntaxe MCODES(119,...)

MCODS(120,...) – Vitesse de rotation actuelle de l'axe

Description : Fournit dans la <mémoire tampon> les **vitesses de rotation actuelles des axes** de tous les axes du système.

Les vitesses de rotation des axes sont uniquement disponibles pour les axes ronds asynchrones, en rotation sans fin.

Mise à jour : Après chaque changement de bloc

Type de données de la <mémoire tampon> : DOUBLE, tableau

Syntaxe : MCODS(120,-1,<version>,<mémoire tampon>,<taille>)

Tab. 7-209: Syntaxe MCODES(120,...)

MCODS(121,...) – Vitesse de rotation programmée de l'axe

Description : Fournit dans la <mémoire tampon> les **vitesses de rotation programmées des axes** de tous les axes du système.

Les vitesses de rotation des axes sont uniquement disponibles pour les axes ronds asynchrones, en rotation sans fin.

Mise à jour : Après chaque changement de bloc

Type de données de la <mémoire tampon> : DOUBLE, tableau

Syntaxe : MCODS(121,-1,<version>,<mémoire tampon>,<taille>)

Tab. 7-210: Syntaxe MCODES(121,...)

MCODS(122,...) – Angle de rotation

Description : Fournit dans la <mémoire de rotation> l'**angle de rotation** pour le plan actif sur le <canal>.

Une valeur DOUBLE est fournie.

Si aucun angle de rotation n'est actif, la valeur 0.0 est renvoyée.

Mise à jour : Après modification

Type de données de la <mémoire tampon> : DOUBLE

Syntaxe :

MCODS(122,<canal>,<version>,<mémoire tampon>,<taille>)

Tab. 7-211: Syntaxe MCODES(122,...)

MCODS(123,...) – Position du pôle pour les fonctions Miroir et Rotation

Description : Fournit dans la <mémoire tampon> les **données polaires programmées** sur le <canal>.

Une valeur DOUBLE est fournie pour chaque coordonnée indiquée.

Si aucune donnée polaire n'est programmée, la valeur 0.0 est renvoyée.

Mise à jour : Après modification

Type de données de la <mémoire tampon> : DOUBLE, tableau

Syntaxe :

MCODS(123,<canal>,<version>,<mémoire tampon>,<taille>)

Tab. 7-212: Syntaxe MCODES(123,...)

MCODS(124,...) – Coordonnées de décalage actives

Description : Fournit dans la <mémoire tampon> les **coordonnées de décalage actives** sur le <canal>.

Une valeur DOUBLE est fournie pour chaque coordonnée.

Si aucune coordonnée de décalage n'est active, la valeur 0.0 est renvoyée.

Mise à jour : Après modification

Type de données de la <mémoire tampon> : DOUBLE, tableau

Syntaxe :

MCODS(124,<canal>,<version>,<mémoire tampon>,<taille>)

Tab. 7-213: Syntaxe MCODES(124,...)

MCODS(125,...) – Facteurs de mise à l'échelle actifs

Description : Fournit dans la <mémoire tampon> les **facteurs de mises à l'échelle actifs** sur le <canal>.

Une valeur DOUBLE est fournie pour chaque facteur.

Si aucun facteur de mise à l'échelle n'est actif, la valeur 1.0 est renvoyée.

Mise à jour : Après modification

Type de données de la <mémoire tampon> : DOUBLE, tableau

Syntaxe :

MCODS(125,<canal>,<version>,<mémoire tampon>,<taille>)

Tab. 7-214: Syntaxe MCODES(125,...)

MCODS(126,...) – Inversion miroir des coordonnées active

Description : Fournit dans la <mémoire tampon> l'**inversion miroir des coordonnées active** sur le <canal>.

Une valeur INTEGER est fournie pour chaque coordonnée.

Si aucune fonction miroir n'est active, la valeur 0 est renvoyée ; sinon c'est la valeur 1.

Mise à jour : Après modification

Type de données de la <mémoire tampon> : INTEGER, tableau

Ordres CPL

Syntaxe :

MCODS(126,<canal>,<version>,<mémoire tampon>,<taille>)

Tab. 7-215: Syntaxe MCODES(126,...)

MCODS(127,...) – Statut de la marque de réentrée de l'avance de bloc

Description : Fournit dans la <mémoire tampon> le statut de la **marque de réentrée de l'avance de bloc** présente sur le <canal>.

Si une marque de réentrée de l'avance de bloc a été générée ou modifiée, la valeur 1 est renvoyée. Si elle a été supprimée, la valeur 0 est renvoyée.

Mise à jour : Après modification

Type de données de la <mémoire tampon> : INTEGER

Syntaxe :

MCODS(127,<canal>,<version>,<mémoire tampon>,<taille>)

Tab. 7-216: Syntaxe MCODES(127,...)

MCODS(128,...) – Statut de la comparaison de l'avance de bloc

Description : Fournit dans la <mémoire tampon> le statut de la **comparaison de l'avance de bloc** réglé actuellement sur le <canal>.

Les statuts de comparaison suivants sont fournis :

- 0 : Avance de bloc inactive
- 5 : Opération de calcul active
- 10 : Bloc cible trouvé, en attente du démarrage CN
- 20 : Programme de comparaison Init actif
- 30 : Saisie manuelle active (écrasement après l'avance de bloc)
- 40 : Bloc d'action actif
- 60 : Programme de comparaison du bloc cible actif
- 80 : Mouvement d'approche actif

Mise à jour : Après modification

Type de données de la <mémoire tampon> : INTEGER

Syntaxe :

MCODS(128,<canal>,<version>,<mémoire tampon>,<taille>)

Tab. 7-217: Syntaxe MCODES(128,...)

MCODS(129,...) – Remise à zéro du canal

Description : Attente d'une **remise à zéro du canal** exécutée sur le <canal> . Aucune donnée initiale ne peut être lue, ce qui signifie que l'ordre ne réapparaîtra que lors de la prochaine remise à zéro du canal indiqué.

Le numéro du canal remis à zéro est renvoyé dans la <mémoire tampon>.

Mise à jour : Après modification

Type de données de la <mémoire tampon> : INTEGER

Syntaxe :

MCODS(129,<canal>,<version>,<mémoire tampon>,<taille>)

Tab. 7-218: Syntaxe MCODES(129,...)

MCODS(131,...) – Vitesse maximale de l'axe

Description : Fournit dans la <mémoire tampon> les **vitesse maximale des axes** pour le canal indiqué.

Si la valeur -1 est affectée à <canal>, les vitesses d'axe de tous les axes du système sont fournies.

Sinon, l'ordre fournit les vitesses des axes du canal affecté.

La vitesse maximale de l'axe est uniquement disponible pour les axes en mode Jog et pour les axes asynchrones.

Mise à jour : Cyclique
Type de données de la <mémoire tampon> : DOUBLE, tableau
Syntaxe :

MCODS(131,<canal>,<version>,<mémoire tampon>,<taille>)

Tab. 7-219: Syntaxe MCODES(131,...)

MCODS(132,...) – Compensation actuelle de la température

Description : Fournit dans la <mémoire tampon> la **compensation actuelle de la température**.

Si la valeur -1 est affectée à <canal>, l'ordre fournit les compensations de température des axes du système.

Sinon, il fournit les compensations de la température des axes du canal affecté.

Mise à jour : Cyclique
Type de données de la <mémoire tampon> : DOUBLE, tableau
Syntaxe :

MCODS(132,<canal>,<version>,<mémoire tampon>,<taille>)

Tab. 7-220: Syntaxe MCODES(132,...)

MCODS(133,...) – Compensation actuelle de l'erreur de pas de la broche

Description : Fournit dans la <mémoire tampon> la **compensation actuelle de l'erreur de pas de la broche**.

Si la valeur -1 est affectée à <canal>, l'ordre fournit les compensations de l'erreur de pas de la broche des axes du système.

Sinon, il fournit les compensations de l'erreur de pas de la broche des axes du canal affecté.

Mise à jour : Cyclique
Type de données de la <mémoire tampon> : DOUBLE, tableau
Syntaxe :

MCODS(133,<canal>,<version>,<mémoire tampon>,<taille>)

Tab. 7-221: Syntaxe MCODES(133,...)

MCODS(134,...) – Compensation de croix actuelle

Description : Fournit dans la <mémoire tampon> la **compensation de croix actuelle**.

Si la valeur -1 est affectée à <canal>, l'ordre fournit les compensations de croix des axes du système.

Sinon, il fournit les compensations de croix des axes du canal affecté.

Mise à jour : Cyclique
Type de données de la <mémoire tampon> : DOUBLE, tableau
Syntaxe :

MCODS(134,<canal>,<version>,<mémoire tampon>,<taille>)

Tab. 7-222: Syntaxe MCODES(134,...)

Ordres CPL

MCODS(135,...) – Décalage de compensation actuel

Description : Fournit dans la <mémoire tampon> le **décalage de compensation actuel**.
Le décalage de compensation est la somme des compensations effectives actives.
Si la valeur -1 est affectée à <canal>, l'ordre fournit le décalage pour les axes du système.
Sinon, il fournit le décalage actif pour les axes du canal affecté.

Mise à jour : Cyclique

Type de données de la <mémoire tampon> : DOUBLE, tableau

Syntaxe : **MCODS(135,<canal>,<version>,<mémoire tampon>,<taille>)**

Tab. 7-223: Syntaxe MCODES(135,...)

MCODS(142,...) – Avances sur trajectoire programmées

Description : Certaines fonctions CN comme G94 peuvent être programmées avec 2 avances. MCODES(142,...) permet de déterminer les différentes avances.
Les avances sont lues à partir du canal <canal> et renvoyées dans la <mémoire tampon> tableau REAL. La <taille> indique le nombre d'éléments du tableau.
L'index d'avance programmé valide à ce moment est fourni avec MCODES(143,...). MCODES(24,...) fournit toujours la première avance programmée seulement, c.-à-d. le premier élément qui est également fourni avec MCODES(142,...).

Mise à jour : Après chaque changement de bloc

Type de données de la <mémoire tampon> : REAL, tableau

Syntaxe : **MCODS(142,<canal>,<version>,<mémoire tampon>,<taille>)**

Tab. 7-224: Syntaxe MCODES(142,...)

MCODS(143,...) – Avance sur trajectoire programmée active

Description : Cet ordre fournit l'index de l'avance sur trajectoire programmée actuellement active. Ce tableau d'avance sur trajectoire peut être déterminé avec MCODES(142, ...). Étant donné que l'avance programmée active peut être commutée par un signal d'interface PLC, l'index de l'avance sur trajectoire programmée peut aussi être modifié (à plusieurs reprises) au sein d'un même bloc.

Mise à jour : Après modification

Type de données de la <mémoire tampon> : INTEGER

Syntaxe : **MCODS(143,<canal>,<version>,<mémoire tampon>,<taille>)**

Tab. 7-225: Syntaxe MCODES(143,...)

MCODS(145,...) – Valeur de crête et valeur maximale de la vitesse de l'axe

Description : Fournit dans la <Mémoire tampon> la valeur de crête, la valeur maximale et la valeur nominale des vitesses d'axe actuelles du canal indiqué. Si la valeur -1 est affectée à <canal>, l'ordre fournit les valeurs des vitesses d'axe de tous les axes du système. Sinon, il fournit les valeurs des vitesses des axes du canal affecté. Trois valeurs (valeur de crête, valeur maximale et valeur nominale) sont transmises pour chaque axe. La division de la valeur de crête et

de la valeur maximale par la valeur nominale permet de déterminer la taille de la valeur de crête et de la valeur maximale, en pourcentage.



La commande met à disposition les valeurs de diagnostic de la charge de l'axe pour la vitesse et pour l'accélération que lorsque les diagnostics sont activés par saisie dans les données système (/SysNcDgnCtrl/AxesLoadDgn/VelLoadDgn/DgnMode = 2, .../AccLoadDgn/DgnMode = 2).



S'ils ne sont plus nécessaires, les diagnostics de la charge de l'axe pour la vitesse et pour l'accélération doivent être désactivés par saisie dans les données système (.../VelLoadDgn/DgnMode = 0, .../AccLoadDgn/DgnMode = 0) ! Cela permet d'éviter de solliciter inutilement l'interrupteur d'interpolateur.

Mise à jour : Cyclique

Type de données de la <mémoire tampon> : REAL, tableau

Syntaxe :

MCODS(145,<canal>,<version>,<mémoire tampon>,<taille>)

Tab. 7-226: Syntaxe MCODES(145,...)

MCODS(146,...) – Valeur de crête et valeur maximale de l'accélération de l'axe

Description : Fournit dans la <mémoire tampon> la valeur de crête, la valeur maximale et la valeur nominale de l'accélération actuelle de l'axe du canal indiqué. Si la valeur -1 est affectée à <canal>, l'ordre fournit les valeurs de l'accélération d'axe pour tous les axes du système. Sinon, il fournit les valeurs de l'accélération d'axe pour le canal affecté. Trois valeurs (valeur de crête, valeur maximale et valeur nominale) sont transmises pour chaque axe. La division de la valeur de crête et de la valeur maximale par la valeur nominale permet de déterminer la taille de la valeur de crête et de la valeur maximale, en pourcentage.



La commande met à disposition les valeurs de diagnostic de la charge de l'axe pour la vitesse et pour l'accélération que lorsque les diagnostics sont activés par saisie dans les données système (/SysNcDgnCtrl/AxesLoadDgn/VelLoadDgn/DgnMode = 2, .../AccLoadDgn/DgnMode = 2).



S'ils ne sont plus nécessaires, les diagnostics de la charge de l'axe pour la vitesse et pour l'accélération doivent être désactivés par saisie dans les données système (.../VelLoadDgn/DgnMode = 0, .../AccLoadDgn/DgnMode = 0) ! Cela permet d'éviter de solliciter inutilement l'interrupteur d'interpolateur.

Mise à jour : Cyclique

Type de données de la <mémoire tampon> : REAL, tableau

Syntaxe :

MCODS(146,<canal>,<version>,<mémoire tampon>,<taille>)

Tab. 7-227: Syntaxe MCODES(146,...)

MCODS(147,...) – Position réelle des axes de la CNC

Description : Fournit dans la <mémoire tampon> les positions réelles de tous les **axes d'avance et auxiliaires** du point de vue de la CNC, par ordre croissant et indépendamment du canal :

Ordres CPL

- en millimètres pour les axes linéaires ;
- en degrés pour les axes ronds.

Le "réglage de la valeur réelle" (par ex. SetPos) n'est **pas** pris en compte dans les valeurs.



Les valeurs fournies par MCODS(147..) se distinguent de celles fournies par MCODS(35..) au niveau des corrections d'erreur d'axe LSEC, CCOMP et de la compensation de la température.

Mise à jour : Cyclique

Type de données de la <mémoire tampon> : REAL, tableau

Syntaxe : **MCODS(147,-1,<version>,<mémoire tampon>,<taille>)**

Tab. 7-228: Syntaxe MCODS(147,...=

Exemple de programmation pour MCODS

Exemple : Fonctions auxiliaires

Programme:

```

:
N10 M3 S1234 T5678
WAIT
10 DIM GROUPID$(64)
15 DIM SYNTAX$(256)
17 ERSION=0
18 CHAN%=1
20 REM Trouver les numéros de groupe M3 et T
25 SYNTAX$="T M3" +CHR$
30 ERR_VAR%=MCODS (94,CHAN%,VERSION,GROUPID$,64,SYNTAX$)
31 PRN#(0,"94 Erreur :",ERR_VAR%)
32 PRN#(0,"T M3:",ACS(MID$(GROUPID$,1,1),
    ACS(MID$(GROUPID$,2,1)))
33 REM Extraire groupes 2 et 1
36 GROUPID$=CHR$(2)+CHR$(1)+CHR$(0):
37 REM Extraire groupes 2 et 1
50 VERSION=0
58 CHAN%=1
60 ERR_VAR%=MCODS (95,CHAN%,VERSION,SYNTAX$,256,GROUPID$)
65 PRN#(0,"95 Erreur :",ERR_VAR%)
70 RN#(0,"Groupe 2 1 : ",SYNTAX$)
71 REM SYNTAX$ contient p.ex. "S1234 M3"

```

MCOPS (partie générale)

INT	READ	WRITE	FUNCT	PROC	WAIT	ERRNO	!

Description : **Motion Control Process Service** – Services de processus Motion Control

Appelle les services de processus Motion Control de l'interface NCS en utilisant le langage CPL. Cela permet de commander les canaux de la commande numérique.

Syntaxe : **MCOPS (<fct>,<canal>[[,[<P1>],[<P2>],[<P3>]],<P4>])**

<fct>	Expression INTEGER Indique la fonction à exécuter. Toutes les fonctions disponibles sont décrites dans le tableau ci-dessous :

<canal>	Expression INTEGER Indique le canal sur lequel la fonction doit agir.
<P1> à <P4>	Paramètres optionnels dépendant de <fct>. Les successions de virgules sont admissibles, mais aucune virgule ne doit être inscrite devant une parenthèse fermante.

Tab. 7-229: Syntaxe MCOPS



Des constantes INTEGER sont parfois indiquées en tant que paramètres de la syntaxe dans les tableaux ci-dessous. Il est également possible de programmer des variables INTEGER à la place de ces constantes. Dans ce cas, la valeur indiquée doit toutefois leur être attribuée au moment de l'appel de l'ordre.

MCOPS(1,...) – Effacement de la course restante

Description : **Motion Control Process Service(1,...)** – Service de processus Motion Control(1,...) : effacement de la course restante

Appelle le service de processus Motion Control de l'interface NCS pour l'"effacement de la course restante".

- Après le déclenchement de la fonction "Effacement de la course restante", tous les blocs CN préparés, y compris le reste du bloc actuel, sont rejetés et traités à nouveau.

Les blocs CPL ou les parties CPL ne sont pas pris en compte.

Exemple :

La variable CPL POS disposait de la valeur 10 lors de la préparation. Après "Effacement de la course restante", le mot CN X[POS] est interprété en tant que X10, bien que POS puisse avoir une valeur totalement différente au moment de ce saut temporel.

Les valeurs de correction éventuellement modifiées sont prises en compte :

- Le point d'arrivée affiché à l'écran est initialisé sur la position actuelle, ce qui supprime en même temps la course restante affichée. Le <canal> passe ensuite à l'état « CN prête » (signal d'entrée de l'API iCh_NCReady).
- Après le démarrage CN (signal de sortie de l'API qCh_NCStart), le programme reprend à partir du point d'interruption en tenant compte des nouvelles valeurs de correction.

Exemple d'utilisation de MCOPS(1,<canal>) : après une modification des tables de correction, si les nouvelles valeurs doivent également être appliquées aux blocs déjà préparés.

Syntaxe :

MCOPS(1,<canal>)	
<canal>	Expression INTEGER Indique le canal sur lequel la fonction doit agir.

Tab. 7-230: Syntaxe MCOPS(1,...)

MCOPS(2,...) – Remise à zéro

Description : **Motion Control Process Service(2,...)** – Service de processus Motion Control(2,...) : Remise à zéro

Appelle les services de processus Motion Control de l'interface NCS pour les "positions initiales".

Ordres CPL

Pour déclencher la remise à zéro : <canal> = -2

- Dans un premier temps, le canal n'accepte plus de nouvelles tâches telles que la sélection d'un programme ou la commutation du mode de fonctionnement.
- L'interpolateur est arrêté.
- Les tâches sur le canal qui n'ont pas encore été traitées sont rejetées.
- Le programme principal est désélectionné.
- Les paramètres machine modifiés qui ne nécessitent aucune montée en régime sont acceptés, par exemple le paramètre CHAN/Ch/Ini/ChResetState (MP 7060 00020) (commutateur de fin de course logiciel).
- Les erreurs et avertissements déclenchés par ce canal sont supprimés.
- L'interpolateur est redémarré.
- L'état de mise en service lors de la remise à zéro (paramètre machine 7060 00020) est accepté, c.-à-d. que les états modaux correspondants sont activés.
- Le canal émet le signal d'interface 0.2 "Remise à zéro exécutée" et est de nouveau prêt à recevoir de nouvelles tâches.

Syntaxe :

MCOPS(2,<canal> [, <variante de remise à zéro>])	
<canal>	Expression INTEGER Indique le canal sur lequel la fonction doit agir. <canal> = -2 : La remise à zéro globale est déclenchée.
<variante de remise à zéro>	Expression INTEGER Définit le comportement de la fonction. La liste ci-dessous contient tous les schémas de comportement définis. Un numéro d'identification est affecté à chaque schéma. Pour paramétrer un schéma particulier, le numéro d'identification correspondant doit être indiqué dans <variante de remise à zéro>. Si plusieurs schémas de comportement doivent être combinés, la somme de tous les numéros d'identification correspondants doit être transmise à la fonction dans <variante de remise à zéro>. Jusqu'à présent, la liste ne contient qu'un seul élément : Numéro d'identification : 2 : Si elle est active, la resélection automatique du programme est neutralisée.

Tab. 7-231: Syntaxe MCOPS(2,...)

Exemple : 10 ERR_VAR=MCOPS(2,2,2)
; Remise à zéro dans le 2e canal sans resélection automatique du programme.

MCOPS(3,...) – Rechercher séquence

Description : **Motion Control Process Service(3,...)** – Service de processus Motion Control(1,...) : rechercher séquence

Appelle les services de processus Motion Control de l'interface NCS pour "Recherche d'un bloc" :

- <bloc de démarrage> et <bloc d'arrivée> sont transmis en tant qu'expressions STRING.

Les conventions suivantes s'appliquent à la recherche du <bloc de démarrage> et du <bloc d'arrivée> :

- Le traitement commence avec le <bloc de démarrage> et se termine avec le <bloc d'arrivée>.

Si le <bloc de démarrage> est manquant ou introuvable, le traitement commence au début du programme.

Si le <bloc d'arrivée> est manquant ou introuvable, le traitement s'arrête à la fin du programme.

- Après "Rechercher séquence", l'état CN passe à PRÊT.

Syntaxe :

MCOPS(3,<canal> [, [<bloc de démarrage>] [, <bloc d'arrivée>]])	
<canal>	Expression INTEGER Indique le canal sur lequel la fonction doit agir.
<bloc de démarrage>	Expression STRING • Les espaces, <Tab> et <LF> au début d'un bloc CN sont ignorées. • Si le <bloc de démarrage> ou le <bloc d'arrivée> commencent par un chiffre et que l'expression est introuvable dans le programme dans lequel la recherche est effectuée, alors le système recherche à nouveau l'expression précédée du caractère "N". De cette façon, rechercher "50" permet de trouver aussi le bloc CN "N50X100". • Si <bloc de démarrage> ou le <bloc d'arrivée> se terminent par un chiffre, alors la recherche ne permet de trouver l'expression dans le programme ciblé que si ce chiffre n'est suivi directement par aucun autre chiffre, par ex. "G1X10" ne trouve pas le bloc CN "G1X100". • Si le <bloc de démarrage> ou le <bloc d'arrivée> se terminent par une lettre, alors l'expression n'est trouvée dans le programme dans lequel la recherche est effectuée que si cette lettre est suivie d'une espace. Par exemple, "50A" trouve le bloc CN "50A = 1", mais pas "50A = 1".
<bloc d'arrivée>	

Tab. 7-232: Syntaxe MCOPS(3,...)

Exemple : 10 ERR_VAR=MCOPS(3,2,"N50","N100")

; Déclenche "Rechercher séquence" sur le canal 2. Le programme principal doit être traité de N50 à N100 inclus.

MCOPS(4,...) – Sélection du programme ou de la saisie manuelle

Description : **Motion Control Process Service(4,...)** – Service de processus Motion Control(4,...) : Sélection du programme ou de la saisie manuelle

Appelle les services de processus Motion Control de l'interface NCS pour sélectionner un programme ou une saisie manuelle :

Sélectionne sur le <canal> programmé un programme à traiter ou une chaîne de caractères à traiter en mode de fonctionnement Saisie manuelle.

Par défaut, la sélection s'effectue sans marche à gauche.

Ordres CPL

Syntaxe :

MCOPS(4,<canal>[[,<string>]],<bloc de démarrage>,<bloc d'arrivée>]],<mode de sélection>))	
<canal>	Expression INTEGER Indique le canal sur lequel la fonction doit agir.
<string>	Expression STRING <i>Selon le <mode de sélection>, le système interprète le paramètre :</i> • en tant que nom de chemin (y compris le nom du programme pièce) d'un programme pièce à sélectionner (100 caractères au maximum) ; • si 32 est indiqué dans <mode de sélection> : en tant que bloc CN unique (taille max. 512 octets, octet final 0 inclus) devant être traité en mode de fonctionnement "Saisie manuelle" ; • si 32+4096 est indiqué dans <mode de sélection> : en tant que plusieurs blocs CN devant être traités en mode de fonctionnement "Saisie manuelle". Plusieurs blocs CN sont séparés par un saut de ligne ("\\n", Hex 0x0A). La taille maximale de tous les blocs CN ne doit pas dépasser 4096 octets, y compris l'octet final 0. • Si 32768 est indiqué dans <mode de sélection> : le nom ou le nom de chemin est indiqué pour la tâche "SETINT" pour les sous-programmes asynchrones.

<bloc de démarrage>	Expression STRING
<bloc d'arrivée>	Le <bloc de démarrage> et le <bloc d'arrivée> définissent les blocs de départ et d'arrivée du traitement au sein du programme pièce. Si le système interprète <string> en tant que saisie manuelle, le <bloc de démarrage> et le <bloc d'arrivée> sont ignorés. • Les espaces, <Tab> et <LF> au début d'un bloc CN sont ignorées. • Si le <bloc de démarrage> ou le <bloc d'arrivée> commencent par un chiffre et que l'expression est introuvable dans le programme dans lequel la recherche est effectuée, alors le système recherche à nouveau l'expression précédée du caractère "N". De cette façon, rechercher "50" permet de trouver aussi le bloc CN "N50X100". • Si <bloc de démarrage> ou le <bloc d'arrivée> se terminent par un chiffre, alors la recherche ne permet de trouver l'expression dans le programme ciblé que si ce chiffre n'est suivi directement par aucun autre chiffre, par ex. "G1X10" ne trouve pas le bloc CN "G1X100". • Si le <bloc de démarrage> ou le <bloc d'arrivée> se terminent par une lettre, alors l'expression n'est trouvée dans le programme dans lequel la recherche est effectuée que si cette lettre est suivie d'une espace. Par exemple, "50A" trouve le bloc CN "50A = 1", mais pas "50A = 1". Si <mode de sélection> = 32768, des définitions sont possibles pour les sous-programmes asynchrones. Ils peuvent être connectés, désactivés, réactivés, supprimés et lancés. • En cas de définitions pour les sous-programmes asynchrones (<mode de sélection> = 32768), le numéro ($1 \leq \text{numéro} \leq 8$) est indiqué en tant que chaîne (par ex. "1") dans le <bloc de démarrage>, et la tâche est indiquée dans le <bloc d'arrivée>. Les tâches pour les sous-programmes asynchrones sont : "SETINT" Connecter "DISABLE" Désactiver "ENABLE" Réactiver "CLRINT" Supprimer "START" Lancer

Ordres CPL

<mode de sélection>	Expression INTEGER Définit le comportement de la fonction. La liste ci-dessous contient tous les schémas de comportement définis. Un numéro d'identification est affecté à chaque schéma. Pour paramétrer un schéma particulier, ce numéro d'identification doit être transmis dans <mode de sélection>. Si plusieurs schémas de comportement doivent être combinés, la somme de tous les numéros d'identification correspondants doit être transmise à l'ordre dans <mode de sélection>. Si le <mode de sélection> est défini sur 2 (Attendre jusqu'à ce que l'état CN passe à PRÊTE) et que le programme à sélectionner n'existe pas ou ne peut pas être exécuté, alors le message d'erreur 6 est renvoyé. Dans tous les autres cas, aucune vérification n'a lieu pour contrôler si le programme est exécutable. L'ordre renvoie la valeur 0 (aucune erreur produite). Seule la liaison suivante génère une erreur d'exécution correspondante.
Valeurs du mode de sélection :	Signification :
1	Le système exécute une liaison lors de la sélection. Si elle n'existe pas encore, une table de liaison est générée pour le programme principal sélectionné. Les tables de liaison sont nécessaires s'il existe des appels de sous-programmes ou des instructions CPL dans le programme.
2	Le système acquitte l'autorisation d'une sélection seulement après que l'état CN est passé à PRÊTE. Normalement, la sélection est acquittée sans attendre le passage à l'état CN PRÊTE.
32	Le système interprète <string> comme un bloc de saisie manuelle.
64	Avant que l'ordre ne sélectionne le programme indiqué ou le bloc de saisie manuelle indiqué, un programme actif ou une saisie manuelle active doit d'abord être désélectionné(e).
256	Condition préalable au déplacement d'axes en mode de fonctionnement "JOG" ou à l'exécution de mouvements en mode de fonctionnement "JOG en coordonnées de la pièce à usiner".
512	Condition préalable au démarrage d'axes en mode de fonctionnement "Approche du point de référence".
2048	Un programme déjà actif est remplacé par le nouveau programme sélectionné. Tous les états modaux sont conservés. En cas de saisie manuelle, l'ancienne chaîne de caractères est remplacée par la nouvelle.
4096	La définition du bloc CN en mémoire tampon agit en combinaison avec le numéro d'identification 32. Il est possible d'affecter d'autres blocs tandis que les blocs précédents sont encore en cours d'exécution.
32768	Ce drapeau permet de piloter les sous-programmes asynchrones (voir <bloc de démarrage> et <bloc d'arrivée>).

65536	Avance de bloc Cet indicateur permet de paramétrer l'avance de bloc. Dans le bloc de démarrage, les données sont indiquées sous forme de chaîne XML. Elles sont incluses entre <SearchRun> et </SearchRun>. Pour saisir la position cible 100, par exemple, la chaîne XML suivante est transmise : <SearchRun><DesPos> 100 </DesPos></SearchRun>
131072	D'autres blocs suivent. Agit en relation avec la saisie manuelle (32) et la définition du bloc CN en mémoire tampon (4096). Une prévisualisation des blocs n'est pas interrompue si la mémoire tampon des blocs est vide (et génère une rampe descendante), mais elle attend jusqu'à ce que des blocs soient de nouveau affectés.
262144	Écrasement : S'il existe un arrêt d'avance lorsque le programme est actif, le bloc CN est transféré au canal dans <string>. Le canal passe au mode de sous-fonctionnement Écrasement et il est possible d'affecter et de traiter d'autres blocs. Si 2048 est ajouté, il est possible de modifier ou supprimer des blocs déjà affectés ou de réinitialiser une erreur de syntaxe. Cet indicateur permet d'affecter des blocs CN pour l'écrasement. <i>On distingue les cas suivants :</i> • Écrasement aux modes de fonctionnement automatiques La première indication permet de passer aux modes de fonctionnement automatiques en cas d'arrêt de l'avance actif dans un mode de fonctionnement interne Écrasement. D'autres blocs CN peuvent alors être affectés. • Écrasement après l'avance de bloc à la fin du programme de comparaison Init Si la commande numérique se trouve à l'état "En attente de la saisie manuelle" à la fin du programme de comparaison Init, ce numéro d'identification permet d'affecter les blocs CN pour la comparaison manuelle. L'ordre agit uniquement en combinaison avec l'avance de bloc (65536).

Tab. 7-233: MCOPS(4,...)

Exemples :

```
10 ERR_VAR=MCOPS(4,1,"sekt.cnc","N50","N100",1)
; Sélection du programme "sekt.cnc" sur le canal 1, y compris la recherche de blocs et la liaison.

20 ERR_VAR=MCOPS(4,1,/usr/"user/p1.cnc")
; Sélection du programme de " p1.cnc" sur le canal 1, sans recherche de bloc ni liaison.

30 ERR_VAR=MCOPS(4,1,"F1000G1X500",,,32)
; Sélectionne le bloc "F1000G1X500" sur le canal 1 en mode de saisie manuelle.

40 ERR_VAR=MCOPS(4,1,"S1000",,,262144)
; En cas d'arrêt de l'avance, commute au mode de sous-fonctionnement Écrasement et sélectionne le bloc « S1000 ». Il est traité avec le démarrage CN.
```

Ordres CPL

50 Flags=262144 + 2048

60 ERR_VAR=MCOPS(4,1,"STA(SP1)",,,Flags)

Un bloc erroné ou déjà affecté est réinitialisé et le bloc "STA(SP1)" est sélectionné.



Si ce n'est pas indiqué explicitement via le mode de sélection, la sélection s'effectue sans exécution de liaison.

MCOPS(5,...) – Désélection du programme

Description : **Motion Control Process Service(5,...)** – Service de processus Motion Control(5,...) : Désélection du programme

Appelle les services de processus Motion Control de l'interface NCS pour désélectionner le programme :

Désélectionne sur le <canal> programmé un programme préalablement sélectionné ou un bloc de saisie manuelle préalablement sélectionné.

Syntaxe :

MCOPS(5,<canal>, <variante de désélection>)	
<canal>	Expression INTEGER Indique le canal sur lequel la fonction doit agir.
<variante de désélection>	Expression INTEGER Définit le comportement de la fonction. La liste ci-dessous contient tous les schémas de comportement définis. Un numéro d'identification est affecté à chaque schéma. Pour régler un schéma particulier, le numéro d'identification correspondant doit être affecté dans <variante de désélection>. Si plusieurs schémas de comportement doivent être combinés, la somme de tous les numéros d'identification correspondants doit être transmise à la fonction dans <variante de désélection>. Jusqu'à présent, la liste ne contient qu'un seul numéro d'identification : 2 : Si la resélection automatique de programme est active, elle peut être neutralisée par la valeur 2 dans cette désélection de programme.

Tab. 7-234: Syntaxe MCOPS(6,...)

Exemple : 10 ERR_VAR=MCOPS(5,2,2)

; Désélection du programme sur le canal 2, sans resélection automatique du programme.

MCOPS(6,...) – Lancement du programme

Description : **Motion Control Process Service(6,...)** – Service de processus Motion Control(9,...) : lancement du programme

Appelle les services de processus Motion Control de l'interface NCS pour lancer le programme :

Démarre sur le <canal> programmé un programme préalablement sélectionné ou un bloc de saisie manuelle préalablement sélectionné.

Syntaxe :

MCOPS(6,<canal>)	
<canal>	Expression INTEGER Indique le canal sur lequel la fonction doit agir.

Tab. 7-235: Syntaxe MCOPS(6,...)

MCOPS(7,...) – Définition du mode de fonctionnement

Description : **Motion Control Process Service(7,...)** – Service de processus Motion Control(7,...) : Définition du mode de fonctionnement

Appelle les services de processus Motion Control de l'interface NCS pour définir le mode de fonctionnement :

Un mode de fonctionnement est ainsi affecté au canal programmé.

La modification du mode de fonctionnement n'est possible que dans les conditions suivantes :

- Le signal d'interface du canal "qCh_OpModePlc" (mode de fonctionnement de l'API) ne doit pas être initialisé.
- Aucun programme ou bloc n'est sélectionné au niveau de la commande numérique.

– ou –

La commutation ne doit s'effectuer qu'entre les modes de fonctionnement Automatique, Bloc continu, Bloc de programme, Étape unique ou Bloc unique.

Ordres CPL

Syntaxe :

MCOPS(7,<canal>,<mofe de fonctionnement>)	
<canal>	Expression INTEGER Indique le canal sur lequel la fonction doit agir.

<mode de fonctionnement>	Expression INTEGER 1. Mode JOG : Les axes peuvent être déplacés en Jog (+/-). Voir aussi MCOPS(4,...) sous <mode de sélection> : numéro d'identification 256. 2. Approche du point de référence : Les axes peuvent être démarrés à l'aide des signaux "Manuel+" ou "Manuel-". Voir aussi MCODS(4,...) sous <mode de sélection> : numéro d'identification 512. 3. Avance de bloc : Les programmes pièces sont traités jusqu'à un bloc cible sans mouvement. 4. Saisie manuelle : Des blocs individuels peuvent être définis pour le traitement. 5. Automatique (Bloc continu) : Les programmes pièces sont exécutés entièrement. 6. Automatique (Bloc-programme) : Les blocs d'un programme pièce sont préparés et traités individuellement avec le "Démarrage CN". 7. Bloc-programme (Étape unique) : Les blocs d'un programme pièce sont préparés de façon cohérente, mais traités individuellement avec le "Démarrage CN". Dans ce cas, les blocs CN qui ont été générés en interne sont aussi traités individuellement. 8. Apprentissage : Les axes peuvent être déplacés en Jog (+/-) ou avec la manivelle. Des blocs individuels peuvent être définis pour le traitement. 9. non attribué 10. Bloc-programme (Bloc unique) : Contrairement à l'étape unique, chaque démarrage CN transmet à l'interpolateur tous les blocs qui ont été générés en interne par un bloc CN programmé pour être traités. 11. Redémarrage : Les axes peuvent être éloignés manuellement du contour, puis rapprochés. 12. Débogueur CPL : Les blocs sont traités individuellement tels qu'ils apparaissent dans le programme pièce. 13. Débogueur CPL : Les blocs sont tous traités jusqu'au point d'arrêt suivant. 14. Mode Jog en coordonnées de la pièce à usiner : Les axes peuvent être déplacés en Jog selon les coordonnées de la pièce à usiner.

Tab. 7-236: Syntaxe MCOPS(7,...)

Ordres CPL

Exemple : 10 ERR_VAR=MCOPS(7,2,5)

; Changement du mode de fonctionnement sur le 2e canal en mode automatique (Bloc continu).

MCOPS(8,...) – Modification de la stratégie de redémarrage

Description : **Motion Control Process Service(8,...)** – Service de processus Motion Control(8,...) : modification de la stratégie de redémarrage

Appelle les services de processus Motion Control de l'interface NCS pour modifier la stratégie de redémarrage :

Cela permet de définir la stratégie de redémarrage sur le <canal> programmé.

Syntaxe :

MCOPS(8,<canal>,<mode>,<point d'approche>)	
<canal>	Expression INTEGER Indique le canal sur lequel la fonction doit agir.
<mode>	Expression INTEGER <i>Offre les possibilités suivantes :</i> • 1 : redémarrage automatique • 2 : redémarrage avec bloc unique • 3 : redémarrage manuel
<point d'approche>	Expression INTEGER <i>Offre les possibilités suivantes :</i> • 1 : le point de départ • 2 : le point d'arrivée • 3 : le point d'interruption

Tab. 7-237: Syntaxe MCOPS(8,...)

MCOPS(9,...) – Fin de l'enregistrement de redémarrage

Description : **Motion Control Process Service(9,...)** – Service de processus Motion Control(9,...) : Fin de l'enregistrement de redémarrage

Appelle les services de processus Motion Control de l'interface NCS pour terminer l'enregistrement de redémarrage :

Termine l'enregistrement de redémarrage sur le <canal> programmé. Les déplacements en mode JOG ne sont plus enregistrés.

Syntaxe :

MCOPS(9,<canal>)	
<canal>	Expression INTEGER Indique le canal sur lequel la fonction doit agir.

Tab. 7-238: Syntaxe MCOPS(9,...)

MCOPS(10,...)

Description : Sans pertinence pour la MTX

MCOPS(11,...) – Redémarrage du canal

Description : **Motion Control Process Service(11,...)** – Service de processus Motion Control(11,...) : redémarrage du canal

Appelle le service de processus Motion Control de l'interface NCS pour le "redémarrage du canal". Cette fonctionnalité redémarre le canal au début du

programme principal sélectionné. Aucune remise à zéro n'est exécutée, c.-à-d. que l'état de mise en service lors de la remise à zéro n'est pas repris non plus.

Si le programme attend par exemple le changement de valeur d'une variable permanente ("WPV"), l'attente est interrompue.



Les états modaux ne sont pas modifiés ! Exception : sous-programmes/cycles ayant un effet modal.

Dans le programme principal, toutes les fonctionnalités modales utilisées au cours du programme doivent être réinitialisées, car toutes les fonctions CN modales actives au moment du redémarrage du canal restent actives.

L'ordre MCOPS(11,...) ne réapparaît que si le redémarrage du canal a été exécuté ou si une erreur s'est produite.

Messages d'erreur :

Les messages d'erreur sont renvoyés sous forme de valeurs de retour. Les messages d'erreur possibles sont indiqués au [chap. "Valeurs de retour pour une erreur" à la page 583](#).

Code d'erreur 2 : Une tâche de redémarrage du canal est déjà exécutée sur ce canal.

Code d'erreur 37 : Le redémarrage du canal a échoué en raison d'une erreur.

Code d'erreur 38 : Le mode de fonctionnement actuel n'autorise pas le redémarrage du canal. Seuls les modes de fonctionnement dans lesquels un programme principal a été sélectionné sont autorisés ; la saisie manuelle et la définition du bloc CN en mémoire tampon, par exemple, ne sont pas autorisées. Par ailleurs, aucun sous-programme asynchrone ne doit être actif.

Déroulement :

- Dans un premier temps, le canal n'accepte plus de nouvelles tâches telles que la sélection d'un programme ou la commutation du mode de fonctionnement.
- L'interpolateur est arrêté.
- Les tâches sur le canal qui n'ont pas encore été traitées sont rejetées.
- Le programme actif est réinitialisé au début du programme principal.
- Les variables CPL locales et globales sont réinitialisées.
- Les sous-programmes modaux sont désactivés.
- L'interpolateur est redémarré.
- Le canal passe à l'état "CN PRÊTE" (signal d'entrée de l'API iCh_NCReady).

Syntaxe :

MCOPS(11,<canal>)	
<canal>	Expression INTEGER Indique le canal sur lequel la fonction doit agir.

Tab. 7-239: Syntaxe MCOPS(11,...)

Exemple :

10 ERR_VAR=MCOPS(11,1)
; Redémarrage du canal sur le 1er canal.

Valeurs de retour pour une erreur

Description :

Tous les appels d'ordres fournissent une valeur de retour à des fins de contrôle et de traitement des erreurs. Cette valeur peut être attribuée à une variable INTEGER ou à une variable REAL.

Ordres CPL

Exemples de syntaxe : `ERR_VAR% = MCOPS (...)`
`ERR_VAR% = MCODS (...)`

**Réactions erronées du programme possibles !**

Si les ordres appelés renvoient un code erreur, cela signifie parfois que des actions importantes pour la suite du déroulement du programme n'ont pas été exécutées ou qu'elles l'ont été partiellement seulement.

C'est pourquoi il est vivement recommandé de vérifier techniquement, après un appel d'ordre, si la fonction a pu être exécutée sans erreurs (par ex. via CASE). Le comportement ultérieur du programme dépend alors du type d'erreur et de la sévérité de l'erreur qui s'est produite.

Les valeurs de retour suivantes sont définies :

- 0 :** Aucune erreur ne s'est produite.
- 1 :** Le canal indiqué n'existe pas.
- 2 :** L'ordre ne peut pas être exécuté, car le canal indiqué est actuellement occupé (l'état actuel ne permet pas l'action).
- 3 :** La remise à zéro commencée sur le canal indiqué n'est pas encore terminée.
- 4 :** Le nom de programme indiqué est trop long (actuellement non utilisé).
- 5 :** L'ordre requiert des points de références approchés.
- 6 :** Le programme indiqué n'existe pas ou ne peut pas être exécuté.
- 7 :** La définition de blocs CN en mémoire tampon étant activée, l'écriture dans la mémoire tampon a été interrompue. Une deuxième instance essayait simultanément d'écrire dans la mémoire tampon.
- 8 :** L'ordre ne peut pas être exécuté dans le mode de fonctionnement actuel.
- 9 :** Le canal ne peut pas être démarré, car son état n'est pas OPÉRATIONNEL.
- 10 :** L'ordre ne peut pas être exécuté, car aucun programme n'est sélectionné.
- 11 :** Le programme indiqué ne peut pas être sélectionné, car l'état du canal ne le permet pas (par ex. état du traitement du bloc et de l'interpolateur est "EN COURS").
- 12 :** actuellement non utilisé
- 13 :** Le mode de fonctionnement ne peut pas être changé, car l'état du canal ne le permet pas.
- 14 :** La cible de "Rechercher séquence" n'a pas été trouvée.
- 15 :** "Rechercher séquence" n'est pas possible, car l'état du canal est bien OPÉRATIONNEL, mais le traitement du programme principal a déjà été lancé (par ex. programme sur M0).
- 16 :** Le nombre d'axes est trop élevé dans la définition des décalages d'origine externes.
- 17 :** Le nombre de groupes DO d'axes est trop élevé dans la définition des décalages d'origine externes.
- 18 :** La syntaxe définie est inconnue.
- 19 :** Index inadmissible dans la définition d'une correction d'outil externe
- 20 :** Le nombre des corrections est trop élevé dans la définition d'une correction d'outil externe (éventuellement en combinaison avec l'index de correction).

21 :	Format inadmissible dans la définition d'une correction d'outil externe
22 :	Position de l'arête coupante inadmissible dans la définition d'une correction d'outil externe
23 :	Groupe de correction inadmissible
24 :	L'axe adressé n'existe pas.
25 :	Une erreur d'exécution comme une erreur de syntaxe a été détectée dans une définition de bloc CN avec démarrage automatique.
26 :	La capacité de la mémoire tampon a été dépassée dans une définition de bloc CN en mémoire tampon.
27 :	La valeur définie pour le filtre de coordonnées est incorrecte.
28 :	La valeur définie pour la CN virtuelle n'est pas autorisée, car l'état n'est pas gelé.
29 :	Manque de mémoire sur le serveur. Les données requises n'ont pas pu être enregistrées.
30 :	Une valeur non admissible a été définie pour la fonction Ncs. Le serveur ne peut donc pas fournir les données.
31 :	Définitions erronées pour le sous-programme asynchrone.
32 :	Définitions erronées pour l'avance de bloc.
33 :	La valeur de correction n'a pas pu être transmise, car la dernière valeur de correction de l'interpolateur n'a pas été acceptée.
34 :	Mémoire client transférée trop petite.
35 :	Le Callchain traité n'est pas valide.
36 :	La valeur programmée pour une correction d'outil est trop élevée ou trop petite et cette dernière n'est donc plus judicieuse.
37 :	Le redémarrage du canal a échoué en raison d'une erreur.
38 :	Le redémarrage du canal n'est pas autorisé en raison du mode de fonctionnement actuel ou de l'état actuel du canal.
100 :	Le numéro Magic du télégramme est erroné.
101 :	La communication NCS est perturbée.
102 :	La fonction indiquée n'est pas disponible dans cette version du logiciel.
103 :	Une erreur interne s'est produite (actuellement non utilisé).



Si le programme appelant contient également des blocs CN, la préparation des blocs est généralement en avance sur traitement par la machine. Ensuite, si un service de processus est appelé via l'ordre MCOPS ou MCODS ou qu'un état de la machine est interrogé au moment de la préparation des blocs, les conditions préalables éventuellement requises pour cela ne sont pas encore remplies.

Cependant, ce problème ne concerne que les ordres qui accèdent justement au canal sur lequel ils sont exécutés eux-mêmes.

Dans ce cas, utiliser l'ordre "WAIT" sur la ligne précédant l'appel. Cela permet d'arrêter le traitement du bloc jusqu'à ce que tous les blocs situés avant "WAIT" soient effectivement traités.

Exemples de programmation

Exemple 1 : Interroger immédiatement l'état SAV et l'état de l'interpolateur du canal 2

Ordres CPL

10 DIM BUF%(2)	Créer un tableau
20 VERSION=0	Fournir immédiatement les données
30 ERR_VAR%=MCOVS(32,2,VERSION,BUF%,2)	Appel d'ordre

L'état SAV est inscrit dans BUF%(1), l'état IPO dans BUF%(2).

VERSION contient le numéro de version actuel des données (important dans l'exemple 2).

Beispiel 2: Warten, bis SAV-Zustand von Kanal 2 auf "Inaktiv" wechselt

<Code von Beispiel 1>

Programme:

```

.
10  INAKTIV = 1
20  WHILE BUF%(1) <> INAKTIV DO
30    ERR_VAR% = MCOVS(32,2,VERSION,BUF%,2)
40  END

```

Exemple 3 : Affichage des noms d'axe dans la fenêtre MSG

Programme:

```

:
30  VERSION=0
40  DIM AXNAME$(512)
50  REM Appeler tous les noms d'axe
60  ERR=MCOVS(33,-1,VERSION,AXNAME$,512)
70  IF ERR=0 THEN
80    REM Déterminer le nombre d'axes
90    DIM AXNMB%(3)
100   VERSION=0
110   ERR=MCOVS(45,-1,VERSION,AXNMB%,3)
120   ANZ=AXNMB%(2)
130  ENDIF
140  IF ERR<>0 THEN
150    PRN#(0,"Erreur survenue : ",ERR)
160  ELSE 170 REM Affichage des noms d'axe
180    FOR I%=0 TO (ANZ-1)
190      NAME$=MID$(AXNAME$,I%*9+1,8)
200      IF ASC(NAME$)<>0 THEN
210        REM Nom d'axe est défini
220        PRN#(0,I%+1,". Nom d'axe : ",NAME$)
230      ENDIF
240    NEXT
250  ENDIF
N260 M30

```

Exemple 4 : Canal/État d'attente

Programme:

```

10  CHAN%=1
20  VERSION%=0
30  STATES%=0
40  ERR=MCOVS(87,CHAN%,VERSION%,STATES%,1)
50  IF ERR=0 THEN
60    MASKE%=1
70    WHILE MASKE% <= STATES% DO
80      CASE (STATES% AND MASKE%) OF
90        LABEL 1:PRN#(0,"Temporisation")
100       LABEL 2:PRN#(0,"Fonction auxiliaire avec acquittement obligatoire")
110       LABEL 4:PRN#(0,"Blocage d'enregistrement")
120       LABEL 8:PRN#(0,"Avance dans le canal égale à 0")
130       LABEL 16:PRN#(0,"Arrêt du programme avec M0/M1")

```



```

140 LABEL 32:PRN#(0,"Arrêt d'avance dans le canal")
150 LABEL 64:PRN#(0,"Blocage d'avance dans le canal ou dans un axe de canal")
160 LABEL 128:PRN#(0,"Blocage d'enregistrement défini par le client")
170 LABEL 256:PRN#(0,"Arrêt de mouvement synchronisé entre canaux (ASTOP, ...)")
180 LABEL 512:PRN#(0,"Attente d'axe durant le remplacement des axes (G511)")
190 LABEL 1024:PRN#(0,"Attente de la variable permanente (WPV)")
200 LABEL 2048:PRN#(0,"Attente du signal d'interface au moment actif (WAITA, ...)")
210 LABEL 4096:PRN#(0,"Attente du signal IF (WAIT(BITIF(...)) ou")
220 PRN#(0," (WAIT(,ZEIT%)) dans la préparation de bloc")
230 LABEL 8192:PRN#(0,"Service de données Motion Control (MCODS(...))")
240 ENDCASE
250 MASKE%=MASKE%*2
260 END
270 ENDIF
M30

```

Exemple 5 : Attribution axe - canal

Programme:

```

10 REM Le programme interroge le nombre d'axes du système
15 REM et les numéros d'axes des axes du canal. Avec ces
20 REM informations, les axes du canal sont d'abord déplacés sur
30 REM position 0 et ensuite sur position <Numéro du
40 REM canal>.
50 CHAN%=SD(8) : REM Numéro de canal
60 IDCHAX%=43 : REM Type de fonction pour l'affectation axe - canal
70 IDMAXAX%=45 : REM Type de fonction pour nombre d'axes
80 DIM BUF%(16) : REM Mémoire tampon pour l'affectation axe - canal
90 SIZE%=16
100 ANZ%=0 : REM Index max. des axes du système
120 REM Déterminer les axes de canal du canal actif
130 VERSION=0
140 ERR=MCODS (IDCHAX%,CHAN%,VERSION,BUF%,SIZE%)
150 IF ERR=0 THEN
160 VERSION=0
170 ERR=MCODS (IDMAXAX%,CHAN%,VERSION,ANZ%,1)
180 IF ERR=0 THEN
190 FOR I%=1 TO ANZ%
200 IF BUF%(I%) = CHAN% THEN
N210 F1000 [AXP(I%,0,0)]; Déplacer les axes du canal sur 0
220 ENDIF
230 NEXT
240 FOR I%=1 TO ANZ%
250 IF BUF%(I%) = CHAN% THEN
N260 M0
N270 WAIT
N280 F1000 [AXP(I%,CHAN%,0)]; Déplacer les axes canal sur CHAN%
290 ENDIF
300 NEXT
310 ENDIF
330 ENDIF
N310 M30

```

7.4.10 Tables XML générales

XTAB

STRUCT	READ	WRITE	FUNCT	PROC	WAIT	ERRNO	!

Description : Xml TABLE – Accès en lecture et en écriture à une table XML quelconque

Contrairement aux fonctions spéciales pour les tables DO, tables de correction D et tableaux de placement, cet ordre permet également d'accéder à toutes les structures partielles d'une table dans la mesure où un type de donnée correspondant a été défini.

Ordres CPL

Syntaxe :

XTAB (<table>,<structure partielle>)	
<table>	Nom de la table XML, le cas échéant avec indication du chemin. En l'absence d'indication du chemin, la table utilisée est celle enregistrée dans le répertoire racine.
<structure partielle>	Structure partielle voulue, indiquée en tant qu'expression XPATH.

Tab. 7-240: Syntaxe XTAB

Exemple :

```
10X!=XTAB("/database/PM1.pmt","/PMT/Set[1]/G154/Corr/Trans/XWCS")  
  
20XTAB("/database/PM1.pmt","/PMT/Set[1]/G154/Corr/Trans/XWCS")=X!+1.0
```

Incrémenter la valeur de 1 dans la table

XTABCR

---	READ	WRITE	FUNCT	PROC	WAIT	ERRNO	!

Description : Xml TABLE CReate – Création d'une table XML générale

XTABCR (<table>,<type>[,<modèle de table>])	
<table>	Nom de la table XML, le cas échéant avec indication du chemin En l'absence d'indication du chemin, la table utilisée est celle enregistrée dans le répertoire racine.
<type>	Type de table (balise racine XML) <i>Les types de tables suivants sont pris en charge :</i> • /ZOT : table de décalage d'origine • /DCT : table de corrections D • /PMT : tableau de placements

Ordres CPL

<modèle de table>	Nom de la table XML modèle, le cas échéant avec indication du chemin Le <type> indiqué doit correspondre au type du <modèle de table>. En l'absence d'indication du chemin, la table utilisée est celle enregistrée dans le répertoire racine.
ERRNO	Variable CPL Si la variable CPL ERRNO est indiquée à un emplacement quelconque dans la liste de paramètres, aucune erreur d'exécution n'est générée. L'erreur est fournie par une valeur correspondante de la variable. Les valeurs de retour suivantes sont possibles : 0 : Accès OK -1 : Erreur de paramètre -13 : Échec lors de l'écriture des données -14 : La table XML existe. -15 : Table XML introuvable -16 : Extension de fichier non valide -17 : Type de table non valide Si la variable CPL ERRNO n'est pas indiquée, une erreur d'exécution est générée en cas d'erreur d'accès.

Tab. 7-241: Syntaxe XTABCR

Crée une table XML sans données au sein du système de fichiers de la MTX.

Syntaxe :

Exemple :

```
10 XTABCR("/database/ZO2.zot", "/ZOT", "/database/
ZO1.zot")
```

Une table de décalage d'origine ZO2.zot est créée dans le répertoire /database. La table /database/ZO1.zot sert ici de modèle.

7.4.11 Base de données d'outils

DBLOAD

---	READ	WRITE	FUNCT	PROC	WAIT	ERRNO	!

Description : **Data Base LOAD** – Chargement d'une base de données

Peut importer tout ou partie d'une table de base de données à partir d'un fichier de la base de données.

Ici, le paramètre <DataSelection> sert à déterminer si toutes les données du bloc de données ou seules les données spécifiques à l'outil ou à l'emplacement sont importées.

Syntaxe :

DBLOAD (<tableBD>,<Key1>,<Key2>,<nom de fichier>[,<DataSelection>])	
<tableBD>	Balise racine de la table de la base de données
<Key1>	Clé de la base de données 1
<Key2>	Clé de la base de données 2

Ordres CPL

<DataSelection>	Optionnel (par défaut : 0) : 0 : Toutes les données du bloc de données sont importées. 1 : Seules les données spécifiques à l'outil sont importées. 2 : Seules les données spécifiques à l'emplacement sont importées.
ERRNO	Variable CPL Si la variable CPL ERRNO est indiquée à un emplacement quelconque dans la liste de paramètres, aucune erreur d'exécution n'est générée. L'erreur est fournie par une valeur correspondante de la variable. Les valeurs de retour suivantes sont possibles : 0 : Accès OK < 0 : Erreur (Pour une description exacte de l'erreur, voir chap. 3.14.5 "Variable ERRNO pour l'analyse des erreurs des fonctions CPL ERRNO" à la page 85) Si la variable CPL ERRNO n'est pas indiquée, une erreur d'exécution est générée en cas d'erreur d'accès.

Tab. 7-242: Syntaxe DBLOAD

Exemple :

```
10 DBLOAD (" /DBT1", 1, 1, "/donneesdb.txt", 1, ERRNO)
```

Les données spécifiques à l'outil du jeu de données (1,1) sont importées à partir du fichier "donneesdb.txt".

DBMOVE

---	READ	WRITE	FUNCT	PROC	WAIT	ERRNO	!

Description :

Data Base MOVE – Déplacement d'un bloc de données au sein d'une base de données

Déplacement d'un bloc de données complet ou d'une structure partielle d'un bloc de données au sein d'une table de base de données.

Le paramètre <DataSelection> permet de déterminer si toutes les données du bloc de données ou uniquement les données spécifiques à l'outil ou à l'emplacement sont déplacées.

Syntaxe :

DBMOVE ((<tableBD>, <SrcKey1>, <SrcKey2>, <DestKey1>, <DestKey2> [, [<mode>] [,<DataSelection>]])	
<tableDB>	Balise racine (root tag) de la table XML
<SrcKey1>	1re clé du jeu de données source
<SrcKey2>	2e clé du jeu de données source
<DestKey1>	1re clé du jeu de données cible
<DestKey2>	2e clé du jeu de données cible

Ordres CPL

<mode>	Optionnel (par défaut : 0) 0 : Les données sélectionnées du jeu de données source sont déplacées vers le jeu de données cible, puis remises à zéro dans le jeu de données source. 1 : Les données sélectionnées du jeu de données source sont copiées dans le jeu de données cible. Le contenu du jeu de données source reste inchangé. 2 : Les données sélectionnées du jeu de données source et cible sont interverties.
<DataSelection>	Optionnel (par défaut : 0) 0 : Toutes les données sont déplacées. 1 : Les données spécifiques à l'outil sont déplacées. 2 : Les données spécifiques à l'emplacement sont déplacées.
ERRNO	Variable CPL Si la variable CPL ERRNO est indiquée à un emplacement quelconque dans la liste de paramètres, aucune erreur d'exécution n'est générée. L'erreur est fournie par une valeur correspondante de la variable. Les valeurs de retour suivantes sont possibles : 0 : Accès OK -1 : Erreur de paramètre -7 : Échec de la lecture des données -13 : Échec lors de l'écriture des données -20 : Balise racine (root tag) erronée -21 : Impossible d'accéder au fichier -22 : Clé non valide -23 : Bloc de données verrouillé Si la variable CPL ERRNO n'est pas indiquée, une erreur d'exécution est générée en cas d'erreur d'accès.

Tab. 7-243: Syntaxe DBMOVE

Exemple de syntaxe :

DBMOVE ("/DBT1", 1, 1, 2, 2) Le contenu du jeu de données (1,1) est décalé dans le jeu de données (2,2).

DBSAVE

Description :

Data Base SAVE – Sauvegarde de la base de données

DBSAVE sert à sauvegarder tout ou partie d'une table de base de données dans un fichier.

Syntaxe :

DBSAVE (<tableBD>,<Key1>,<Key2>,<NomFichier>[,<mode>])							
<tableDB>	Balise racine (root tag) de la table XML						

---	READ	WRITE	FUNCT	PROC	WAIT	ERRNO	!

Ordres CPL

<Key1>	1re clé du jeu de données
<Key2>	2e clé du jeu de données
<NomFichier>	Nom de fichier
<mode>	Optionnel (par défaut : 0) 0 : Les données sont écrites dans le fichier "NomFichier". Si le fichier existe déjà, son contenu est écrasé. 1 : Les données sont annexées au fichier existant "NomFichier".
ERRNO	Variable CPL. Si la variable CPL ERRNO est indiquée à un emplacement quelconque dans la liste de paramètres, aucune erreur d'exécution n'est générée. L'erreur est fournie par une valeur correspondante de la variable. Les valeurs de retour suivantes sont possibles : 0 : Accès OK -1 : Erreur de paramètre -7 : Échec de la lecture des données -13 : Échec lors de l'écriture des données -20 : Balise racine (root tag) erronée -21 : Impossible d'accéder au fichier -22 : Clé non valide -23 : Bloc de données verrouillé Si la variable CPL ERRNO n'est pas indiquée, une erreur d'exécution est générée en cas d'erreur d'accès.

Tab. 7-244: Syntaxe DBSAVE

Si les deux clés ont la valeur -1, alors tous les jeux de données sont écrits dans le fichier. Si la 1re clé a une valeur différente de -1 et que la 2e clé a la valeur -1, alors tous les jeux de données dont la 1re clé a la valeur indiquée sont écrits dans le fichier.

Exemple de syntaxe :

```
DBSAVE ("/DBT1", 1, 1, "/donneesdb.txt")
```

Le contenu du jeu de données (1,1) est sauvegardé dans le fichier "donneesdb.txt".

DBSEA

STRUCT	READ	WRITE	FUNCT	PROC	WAIT	ERRNO	!

Description : **Data Base SEArch** – Recherche d'un bloc de données

L'ordre CPL renvoie l'en-tête du premier jeu de données correspondant au critère de recherche. Dans ce cas, la variable <SearchRes> renvoie la valeur 1.

La recherche du bloc de données commence par le jeu de données défini dans

<StartKey1> et <StartKey2>.

Si la valeur de l'une des deux clés de démarrage est égale à -1, alors la recherche débute par le premier jeu de données de la table de base de données.

Syntaxe :

DBSEA (<tableDB>,<StartKey1>,<StartKey2>,<SearchCond>,<SearchRes>,[<ResVar>])	
<tableBD>	Nom de la table de la base de données
<StartKey1>	Key1 du bloc de données de démarrage
<StartKey2>	Key2 du bloc de données de démarrage
<SearchCond>	Critère de recherche sous forme de chaîne (voir chap. "DBSEAX" à la page 594)
<SearchRes>	Résultat de la recherche, variable de type INTEGER 0 : Aucun jeu de données correspondant au critère de recherche n'a été trouvé. 1 : Un jeu de données correspondant au critère de recherche a été trouvé.
<ResVar>	Variable de type INTEGER Si <ResVar> est indiquée, aucune erreur d'exécution n'est générée en cas d'erreur d'accès. Les valeurs de retour suivantes sont possibles : 0 : Accès OK 1 : Erreur lors de l'accès 2 : Espace mémoire insuffisant 3 : Type de variable non valide 4 : Critère de recherche erroné Si aucune <ResVar> n'est indiquée, une erreur d'exécution est générée en cas d'erreur d'accès.

Tab. 7-245: Syntaxe DBSEA

Exemple :

```

10 SD.DBRec.Hd=DBSEA("DBT1",-1,-1,"K1=1",FOUND%)
20 WHILE FOUND%=1 DO
30 SD.DBRec.Hd=DBSEA("DBT1", SD.DBRec.Hd.K1, SD.DBRec.Hd.K2, "K1=1",FOUND%)
40 END

```

La donnée système suivante est requise :

Programme:

```

<Variable Storage="volatile">
  <Name>DBRec</Name>
  <Type>DBT1Rec_t</Type>
</Variable>

```

Ordres CPL

DBSEAX

INT	READ	WRITE	FUNCT	PROC	WAIT	ERRNO	!

Description : **Data Base SEArch eXtended** – Recherche étendue de blocs de données

Recherche un ou plusieurs jeux de données dans une table de base de données d'outils. L'ordre fournit une liste des en-têtes des jeux de données qui correspondent au critère de recherche. Les en-têtes sont enregistrés dans une table de données système <HeaderArr> et triés selon Key1 et Key2. Le paramètre <ErgSize> sert à définir le nombre maximal d'en-têtes de blocs de données pouvant être recherchés. La valeur de retour de DBSEAX fournit le nombre de jeux de données trouvés.

La recherche de blocs de données commence par le jeu de données défini par
<StartKey1> et <StartKey2>.

Si la valeur de l'une des deux clés de démarrage est égale à "-1", alors la recherche débute par le premier jeu de données de la table de base de données.

Syntaxe :

DBSEAX (<tableDB>,<StartKey1>,<StartKey2>,<SearchCond>,<HeaderArr>,<ArrSize>)	
<tableBD>	Nom de la table de la base de données
<StartKey1>	Key1 du bloc de données de démarrage
<StartKey2>	Key2 du bloc de données de démarrage
<SearchCond>	Critère de recherche sous forme de chaîne de caractères (voir ci-dessous)
<HeaderArr>	Donnée système de type En-tête de bloc de données (par ex. DBT1Hd_t) Si plusieurs jeux de données sont recherchés, il s'agit d'un tableau. Le tableau est indexé par incrémentation du dernier index contenu dans les données SD, par ex. SD.HdArr[1,1] – Le deuxième index est incrémenté dans HdArr. SD.SDArr[1,1].RecArr[1].Hd – L'index de RecArr est incrémenté.

Ordres CPL

<ArrSize>	Nombre maximal d'en-têtes de jeux de données recherchés Remarque : <HeaderArr> doit être dimensionné de sorte à pouvoir intégrer ce nombre d'en-têtes.
ERRNO	Variable CPL Si la variable CPL ERRNO est indiquée à un emplacement quelconque dans la liste de paramètres, aucune erreur d'exécution n'est générée. L'erreur est fournie par une valeur correspondante de la variable. Les valeurs de retour suivantes sont possibles : 0 : Accès OK < 0 : Erreur (pour une description précise de l'erreur, voir chap. 3.14.5 "Variable ERRNO pour l'analyse des erreurs des fonctions CPL ERRNO" à la page 85) Si la variable CPL ERRNO n'est pas indiquée, une erreur d'exécution est générée en cas d'erreur d'accès.

Tab. 7-246: Syntaxe DBSEAX

Critère de recherche (<Search-Cond>)

La syntaxe suivante doit être respectée :

- Une liste de conditions individuelles à satisfaire absolument est prédéfinie.
- Une condition individuelle se présente sous la forme :
<nom de balise><opérateur de comparaison><valeur> avec
 - <opérateur de comparaison> :
 - "=" égal à
 - ">" supérieur à
 - ">=" supérieur ou égal à
 - "<" inférieur à
 - "<=" inférieur ou égal à
 - "<>" différent de

Seuls les opérateurs de comparaison suivants sont admissibles pour SKQ : "=" et "<>",
par ex. SKQ=4711
- Une condition individuelle pour un masque de bit se présente sous la forme :
<nom de balise>=<valeur>:<masque>
 - <valeur> : Valeur que les bits pertinents doivent avoir.
 - <masque> : Définition des bits pertinents.

Exemple : BQ1=0x307000:0xF0FF00 avec :

Seules les requêtes d'égalité sont possibles.
Les valeurs et les masques sont interprétés comme suit :

 - en tant que chiffre hexadécimal, s'ils commencent par "0x" ou "0X" ;
 - en tant que chiffre octal, s'ils commencent par "0" ;
 - en tant que chiffre décimal dans tous les autres cas.

Ordres CPL

Les espaces de tête ou de queue ne sont pas évalués au sein de <valeur>. Si une chaîne de caractères dans laquelle ces espaces ont une signification doit être prédéfinie, alors la chaîne de caractères partielle doit être inscrite entre guillemets (par ex. : SKQ=' ').

Pour utiliser un guillemet dans une chaîne de caractères partielle inscrite entre guillemets, ce guillemet doit être précédé d'une barre oblique inverse "\" (par ex. : SKQ='Achim\'s').

- Conditions aux limites
- Les conditions individuelles sont séparées les unes des autres par une virgule ",".
 - Toutes les conditions individuelles d'une liste doivent se référer à différents noms de <balise> (tag).
 - Seuls les noms de balise prédéfinis peuvent être utilisés.

Exemples de syntaxe :

```
K1=1
IKQ2=3
BQ2=0x1:0x1
```

Tous les jeux de données correspondant à un critère de recherche doivent être trouvés. Pour cela, la recherche doit être répétée jusqu'à ce que le nombre des jeux de données trouvés soit inférieur au nombre maximal de jeux de données recherchés. Pour toute nouvelle recherche, les clés de démarrage doivent être ajustées au dernier jeu de données trouvé.

Programme:

```
10 KEY1%=-1
10 KEY2%=-1
30 ARRSIZE%=10
31 REPEAT
33   FOUND% = DBSEAX("DBT1",KEY1%,KEY2%,"K1>0",SD.HdArr[1],ARRSIZE%,ERRNO)
35   IF ERRNO=0 THEN
37     IF FOUND%>=1 THEN
59       FOR I%= 1 TO FOUND%
43         PRN#(0,Secteur : ",SD.HdArr[I%].K1,", Endroit : ",SD.HdArr[I%].K2)
46       NEXT I%
49       KEY1%= SD.HdArr[FOUND%].K1
51       KEY2%= SD.HdArr[FOUND%].K2
53   ENDIF
55   ELSE
57     PRN#(0,"Erreur d'accès à la base de données des outils !")
61   ENDIF
63 UNTIL (FOUND%<ARRSIZE%) OR (ERRNO<>0)
```

La donnée système suivante est requise :

Programme:

```
<Variable Dimension="10" Storage="volatile">
  <Name>HdArr</Name>
  <Type>DBT1Hd_t</Type>
</Variable>
```

DBTAB

STRUCT	READ	WRITE	FUNCT	PROC	WAIT	ERRNO	!

Description :

Data Base TABLE – Accès en lecture ou en écriture à une base de données

Peut importer un jeu de données complet ou une structure partielle d'une table de base de données d'outils dans une variable CPL et réexporter un tel élément de la variable vers le jeu de données.

Syntaxe :

DBTAB (<tableBD>,<Key1>,<Key2>[,<ResVar>])	
<tableBD>	Nom de la table de base de données contenant des structures partielles (chaîne Xpath) Un point "." ou une barre oblique "/" peuvent être utilisés comme caractères de séparation entre les paramètres.
<Key1>	Clé de la base de données 1
<Key2>	Clé de la base de données 2
<ResVar>	Variable de type INTEGER Si <ResVar> est indiquée, aucune erreur d'exécution n'est générée en cas d'erreur d'accès. Les valeurs de retour suivantes sont possibles : 0 : Accès OK 1 : Erreur lors de l'accès 2 : Espace mémoire insuffisant 3 : Type de variable non valide Si aucune <ResVar> n'est indiquée, une erreur d'exécution est générée en cas d'erreur d'accès.

Tab. 7-247: Syntaxe DBTAB

Exemple :

```

10 SV.A=DBTAB("DBT1.Rec",1,1)
20 D!=DBTAB("DBT1.Rec.UD.Ed.Geo.L1",1,1)
30 SV.A.UD.Ed.Geo.L1 = 10
40 DBTAB("DBT1.Rec",1,1) = SV.A
    
```

DBTABX

STRUCT	READ	WRITE	FUNCT	PROC	WAIT	ERRNO	!

Description :

Data Base TABLE eXtended – Accès élargi à une table de base de données
 Peut importer un jeu de données complet ou une structure partielle d'une table de base de données d'outils dans une variable CPL et réexporter un tel élément de la variable vers le jeu de données. Ici, le paramètre <DataSelection> sert à déterminer si toutes les données du jeu de données ou seules les données spécifiques à l'outil ou à l'emplacement sont écrites.

Syntaxe :

DBTABX (<tableBD>,<Key1>,<Key2>[,<DataSelection>])	
<tableBD>	Nom de la table de base de données contenant des structures partielles (chaîne Xpath) Un point "." ou une barre oblique "/" peuvent être utilisés comme caractères de séparation entre les paramètres.
<Key1>	Clé de la base de données 1
<Key2>	Clé de la base de données 2

Ordres CPL

<Key1>	Clé de la base de données 1
<Key2>	Clé de la base de données 2
<DataSelection>	En cas d'accès en écriture, le verrouillage est supprimé indépendamment de <DataSelection>. Les valeurs suivantes sont possibles : 0 : Toutes les données du jeu de données sont écrites. 1 : Seules les données spécifiques à l'outil sont écrites. 2 : Seules les données spécifiques à l'emplacement sont écrites. 3 : Aucune donnée n'est écrite. Si <DataSelection> n'est pas indiqué, la valeur par défaut 0 est utilisée en cas d'accès en écriture. En cas d'accès en lecture, le paramètre <DataSelection> n'a aucune signification.
ERRNO	Variable CPL Si la variable CPL ERRNO est indiquée à un emplacement quelconque dans la liste de paramètres, aucune erreur d'exécution n'est générée. L'erreur est fournie par une valeur correspondante de la variable. Les valeurs de retour suivantes sont possibles : 0 : Accès OK < 0 : Erreur (pour une description précise de l'erreur, voir chap. 3.14.5 "Variable ERRNO pour l'analyse des erreurs des fonctions CPL ERRNO" à la page 85) Si la variable CPL ERRNO n'est pas indiquée, une erreur d'exécution est générée en cas d'erreur d'accès.

Tab. 7-249: Syntaxe DBTABXL

Exemple :

```

10 SV.A = DBTABXL("DBT1.Rec",1,1)
20 D! = DBTABXL("DBT1.Rec.UD.Ed.Geo.L1",1,1)
30 SV.A.UD.Ed.Geo.L1 = 10
40 DBTABXL("DBT1.Rec",1,1) = SV.A

```

7.4.12 Erreurs et avertissements

SETERR

---	READ	WRITE	FUNCT	PROC	WAIT	ERRNO	!

Description : SET ERROR – Définition d'une erreur d'exécution

Une erreur d'exécution peut être générée avec l'ordre CPL SETERR suite à l'évaluation d'ERRNO. Le programme pièce est alors interrompu à cette position. Le texte indiqué dans la <chaîne d'erreur> est toujours affiché, indépendamment du langage actuellement configuré.

Le paramètre optionnel <mode> permet de contrôler sur le canal actuel si le nom du programme pièce et la ligne actuelle du programme pièce sont également indiqués. Si la sortie s'effectue sur un autre canal, seul le texte indiqué s'affiche.

Ordres CPL

Syntaxe :

SETERR (<chaîne d'erreur> [, [<canal>] [, <mode>]])	
avec	
<chaîne d'erreur>	Texte affiché en tant qu'erreur sous le code erreur 3371
<canal>	Numéro du canal sur lequel l'erreur doit être signalée. Si le <canal> n'est pas indiqué, le canal actuel est utilisé comme valeur par défaut.
<mode>	Définit, si le nom du programme pièce et la ligne du programme pièce sont indiqués sur le canal actuel. 0 : Aucune sortie d'informations relatives au programme pièce 1 : Sortie du nom du programme pièce et de la ligne du programme pièce Si <mode> n'est pas indiqué, la valeur 0 est utilisée par défaut.
ERRNO	Variable CPL programmable n'importe où entre les parenthèses. Avec ERRNO, aucune erreur d'exécution n'est générée en cas d'erreur. Les valeurs de retour possibles sont : 0 : Accès OK -1 : Erreur de paramètre -5 : Canal inexistant

Tab. 7-250: Syntaxe SETERR

Exemple : Lire la position de l'axe X du canal sur le premier canal

```

10 POS = ACS("X",1,1,ERRNO)
20 IF ERRNO <> 0 THEN
30 SETERR("Échec lors de la lecture de la position")
40 ENDIF

```

SETWARN

---	READ	WRITE	FUNCT	PROC	WAIT	ERRNO	!

Description :

SET WARNING – Définition d'un avertissement quant à la durée d'exécution

Une erreur d'exécution peut être générée avec l'ordre CPL SETWARN suite à l'évaluation d'ERRNO, tandis que l'exécution du programme pièce continue. Le texte indiqué dans la <chaîne d'avertissement> est toujours indiqué, indépendamment du langage actuellement configuré.

Le paramètre optionnel <mode> permet de contrôler sur le canal actuel si le nom du programme pièce et la ligne actuelle du programme pièce sont également indiqués. Si la sortie s'effectue sur un autre canal, seul le texte indiqué s'affiche.

Syntaxe :

SETWARN (<chaîne d'avertissement> [, [<canal>] [, <mode>]])	
avec	
<chaîne d'avertissement>	Texte affiché en tant qu'AVERTISSEMENT sous le code erreur 3372

<canal>	Numéro du canal sur lequel l'avertissement doit être sorti. Si le <canal> n'est pas indiqué, le canal actuel est utilisé comme valeur par défaut.
<mode>	Définit, si le nom du programme pièce et la ligne du programme pièce sont indiqués sur le canal actuel. 0 : Aucune sortie d'informations relatives au programme pièce 1 : Sortie du nom du programme pièce et de la ligne du programme pièce Si <mode> n'est pas indiqué, la valeur 0 est utilisée par défaut.
ERRNO	Variable CPL programmable n'importe où entre les parenthèses. Avec ERRNO, aucune erreur d'exécution n'est générée en cas d'erreur. Les valeurs de retour possibles sont : 0 : Accès OK -1 : Erreur de paramètre -5 : Canal inexistant

Tab. 7-251: Syntaxe SETWARN

CLRWARN

---	READ	WRITE	FUNCT	PROC	WAIT	ERRNO	!

Description : **CLear WARNING** – Effacement des avertissements

L'ordre CLRWARN permet d'effacer tous les avertissements d'un canal générés par SETWARN.

Syntaxe :

CLRWARN([<canal>]) avec	
<canal>	Numéro du canal sur lequel tous les messages d'avertissement doivent être effacés. Si le <canal> n'est pas indiqué, le canal actuel est utilisé comme valeur par défaut.
ERRNO	Variable CPL programmable n'importe où entre les parenthèses. Avec ERRNO, aucune erreur d'exécution n'est générée en cas d'erreur. Les valeurs de retour possibles sont : 0 : Accès OK -1 : Erreur de paramètre -5 : Canal inexistant

Tab. 7-252: Syntaxe CLRWARN

Exemple :

```

10 CLRWARN (, ERRNO)
20 IF ERRNO <> 0 THEN
30 SETWARN("Avertissements non effacés")

```

Ordres CPL

40 ENDIF

GETERR

INT	READ	WRITE	FUNCT	PROC	WAIT	ERRNO	!

Description :

GET ERRor – Requête des erreurs

Cet ordre appelle l'erreur actuelle dans un programme CPL. Ceci comprend les éléments, le code erreur actuel, le numéro du canal sur lequel l'erreur s'est produite et la catégorie d'erreur correspondante.

Toute erreur qui se produit est enregistrée dans un tableau avec ses éléments. Le nombre maximal d'erreurs que le tableau peut inclure est limité au moyen du dimensionnement (DIM) du paramètre <code erreur>.

L'ordre GETERR renvoie les valeurs suivantes :

- 1 : Échec de l'exécution de l'ordre
- ≥ 0 : Nombre d'erreurs actuelles sur le <canal>

Syntaxe :

GETERR (<canal>,[<catégorie>],<n° erreur>[,<nombre>])	
<canal>	Numéro du canal interrogé -1 : tous les canaux > 0 : n° du canal

Ordres CPL

<catégorie>	0 : tous les avertissements et toutes les erreurs (par défaut) 1 : erreur mineure du système 2 : erreur de réglage ou d'entraînement 3 : erreur d'interpolateur 4 : erreur matérielle 5 : erreur ICL 6 : erreur du programme pièce 7 : avertissement quant à la durée d'exécution 8 : messages MZA : erreurs 9 : messages MZA : avertissements 10 : messages MZA : avis
<code erreur>	Variable de résultat : Tableau INTEGER bidimensionnel contenant au moins 3 éléments dans la deuxième dimension (DIM <code erreur>% (x, 3)), valeur par défaut : 0. L'ordre fournit les numéros des erreurs qui se sont produites sur le <canal>, par ordre chronologique décroissant. <i>Signification des 3 éléments de la 2e dimension :</i> • <code erreur>(x,1) : code erreur • <code erreur>(x,2) : canal d'erreur (-1 = multicanaux) • <code erreur>(x,3) : catégorie d'erreur (si déclarée à l'aide de l'ordre DIM) 0 = catégorie inconnue Autres valeurs telles que <catégorie> Exemple : DIM ERRNO% (100, 3).

Tab. 7-253: Syntaxe 1 GETERR



Seul le nom de la variable, sans dimension ni index, doit être indiqué !

<nombre>	Variable INTEGER (valeur par défaut : 1) Définit le nombre d'erreurs à lire. 1 : Valeur par défaut > 0 : La validité des valeurs de paramétrage n'est pas contrôlée, c.-à-d. qu'aucune erreur du programme pièce n'est générée si le nombre de données à lire est supérieur au dimensionnement du tableau. Exemple : <nombre> = 120, mais DIM ERRNO% (100, 3). Dans ce cas, 20 erreurs ne sont pas lues.

Tab. 7-254: Syntaxe 2 GETERR

Exemples : Interroger la dernière erreur du programme pièce dans le canal 2

Ordres CPL

Programme:

```

10 DIM ERR_CHAMP%(5,3): REM Tableau entier avec 5 éléments
20 REM Interroger la dernière erreur du programme pièce dans le canal 2
30 CHAN%=2: CATEGORY%=6
40 ERG%=GETERR(CHAN%,CATEGORY%,ERR_CHAMP%,1)

```

Exploitation du résultat pour 5 éléments de tableau.*Programme:*

```

10 DIM ERR_CHAMP%(5,3): REM Tableau entier avec 5 éléments
20 CHAN%=2: CATEGORY%=0
25 REM Interroger les avertissements et erreurs dans le canal 2
30 ERG%=GETERR(CHAN%,CATEGORY%,ERR_CHAMP%,5)
40 FOR I%=1 TO ERG%
50 IF ERR_CHAMP%(I%,3)=6 THEN
60 PRN#(0,"Erreur du programme pièce : ",ERR_CHAMP%(I%,1))
70 ENDIF
80 NEXT I%

```

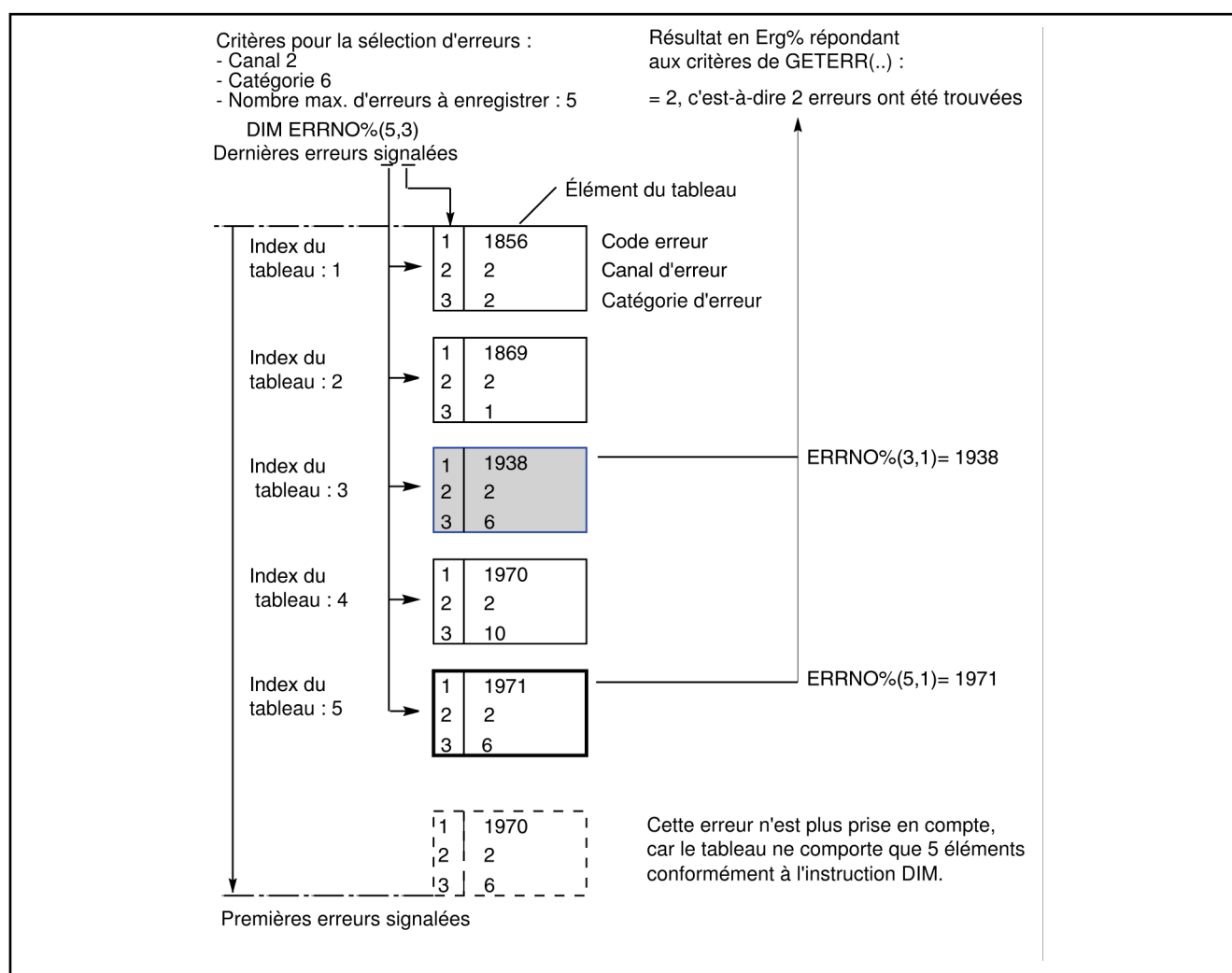


Fig. 7-6: Exemple

Le résultat indiqué dans la fenêtre MSG comprend les erreurs du programme pièce 1938 et 1971. La variable Erg% contient la valeur 2.



L'ordre GETERR permet notamment de comprendre l'ordre chronologique d'apparition d'une ou plusieurs erreurs afin de trouver la véritable cause de l'erreur.

7.5 Fichiers et répertoires

7.5.1 Généralités

Signification

Les fichiers sont des conteneurs qui contiennent des données. Au cours d'un programme CPL, les données peuvent être extraites des fichiers ou archivées dans des fichiers. Il est ainsi possible, par exemple, d'enregistrer d'abord des valeurs mesurées, puis de les affichées ou de les imprimer. Les données sont administrées dans le système de fichiers interne à la commande numérique. Elles sont archivées de manière organisée dans une structure hiérarchique. L'accès aux différentes données s'effectue via les chemins du répertoire. Il est possible d'administrer les fichiers sous différents modes de fonctionnement. Le fichier correspondant aux données à lire ou à écrire doit toujours être ouvert (voir les ordres OPENW, OPENR) seulement pour accéder à ces données en lecture ou en écriture ; il doit être refermé (voir l'ordre CLOSE) dès que l'accès aux données n'est plus utile.

Noms de fichiers

Description : Les conventions suivantes doivent être respectées lors de l'attribution d'un nom de fichier :

- La longueur maximale est de 30 caractères.

Aucune distinction n'est faite entre le nom et l'extension possible du nom de fichier. Toutes les lettres, tous les chiffres ainsi que les caractères spéciaux "." et "_" sont admissibles.

Les noms de fichiers des programmes pièces peuvent avoir une longueur maximale de 28 caractères, étant donné que la commande numérique génère un fichier lors de la liaison dont le nom se compose du nom de fichier initial plus 2 autres caractères.



Le caractère spécial "\$" ne doit être utilisé que pour les fichiers générés en interne. Les noms de fichiers externes (via l'interface utilisateur, DNC) ne doivent pas contenir le caractère "\$".

- Les noms de fichiers sont sensibles à la casse.

Exemples : Noms de fichiers

P123456789.PRГ

P12_Donnee_Dial

P12_DONNEE_DIAL

- Les noms de fichier "." et ".." sont interdits, étant donné qu'ils sont déjà utilisés en interne.
- Les noms de fichiers doivent être uniques au sein d'un répertoire. Différents fichiers peuvent cependant avoir le même nom s'ils sont enregistrés dans des répertoires différents.

Structure de fichier aléatoire

Description : Un fichier aléatoire possède des **composants** (enregistrements) d'une **longueur fixe définissable**. Un accès direct libre à un composant quelconque du fichier est ainsi possible. La répartition du fichier aléatoire dans les enregistrements de longueur fixe permet d'accéder directement à un enregistrement particulier. Comme pour les fichiers séquentiels, les données sont archivées en tant que caractères ASCII. Ceci permet non seulement d'y accéder com-

Ordres CPL

me d'habitude avec l'éditeur, mais également d'archiver et d'extraire les fichiers aléatoires.

Le fichier aléatoire présente l'avantage d'accéder plus rapidement aux données nécessaires. En plus, les fichiers d'un enregistrement peuvent être traités et/ou modifiés sans que la structure du reste du fichier ne doive être modifiée. Les enregistrements qui ne sont pas remplis complètement avec des données sont complétés par des espaces (→ Blanks) jusqu'à ce que la longueur définie soit atteinte.

Si une variable STRING dont la longueur est supérieure à la longueur d'un enregistrement doit être insérée dans un fichier aléatoire, alors les premiers caractères de la variable STRING sont remplis dans l'enregistrement jusqu'à ce que la longueur définie soit atteinte et les caractères restants sont rejetés.

Lors de la lecture du fichier, la fin du fichier est reconnue par "EOF". Les instructions REWRITE et CLOSE sont utilisées de la même manière que pour les fichiers séquentiels.

Un accès séquentiel à un fichier aléatoire est également possible.

Structure de fichier séquentielle

Description : Un fichier séquentiel contient une chaîne de **composants** (enregistrements) pouvant posséder une **longueur variable**. Si un certain enregistrement est recherché dans un fichier séquentiel, cet enregistrement doit être recherché dès le début du fichier. Un accès direct n'est pas possible. Si la longueur d'un enregistrement est modifiée dans un fichier séquentiel, alors tous les enregistrements suivants doivent être déplacés.

Contrairement aux fichiers aléatoires, les fichiers séquentiels contiennent des enregistrements de différentes longueurs (longueur maximale : 1024 caractères). La fin d'un enregistrement est identifiée par <LF> (non compté dans la longueur). <ETX><LF> avec un pointeur EOF est inséré à la fin du dernier enregistrement d'un fichier. Un pointeur EOF est un indicateur de fin des données exploitables (<ETX>) dans un fichier.

7.5.2 Répertoires

DIRINF

INT	READ	WRITE	FUNCT	PROC	WAIT	ERRNO	!

Description : **DIRectory INFormation** – Informations administratives d'un répertoire

DIRINF permet d'interroger les données administratives du système de fichiers de la commande IndraMotion MTX dans le programme CPL. L'ordre fournit une valeur de type "INTEGER" (si les résultats sont négatifs, le résultat réel est supérieur à la valeur INTEGER maximale, à savoir > 2 147 483 647).

La syntaxe change en fonction de la valeur du paramètre <index1>.

Syntaxe :

Pour <index1> = 1, 2 ou 3 : DIRINF (<index1>[, <index2>]) Pour <index1> = 4 ou 6 : DIRINF (<index1>, [<index2>], <VarRésultat>[, <numéro du fichier>]) Pour <index1> = 5, 7 ou 8 : DIRINF (<index1>, [<index2>], <VarRésultat>[])	
<index1>	Type Constante INTEGER Mode d'ordre : 1 : espace mémoire libre 2 : espace mémoire occupé 3 : nombre de fichiers existants dans le répertoire 4 : nom du fichier dans le répertoire avec l'index <numéro du fichier> 5 : nom de répertoire du programme principal actif 6 : nom de répertoire mounted avec l'index <numéro du fichier> 7 : nom de répertoire du programme ou sous-programme actif 8 : nom du programme appelant
<index2>	Type STRING Mode supplémentaire disponible : Nom de répertoire (par défaut : répertoire actuel)
<VarRésultat>	Type de variable STRING dimensionnée, uniquement pour les modes d'ordre 4 à 8. Fournit le nom d'un fichier dans le répertoire indiqué ou le nom du répertoire à la fin de l'exécution de l'ordre.
<numéro du fichier>	Type "INTEGER", uniquement pour les modes d'ordre 4 à 6. Dans le <numéro du fichier>, le numéro ordinal du fichier est indiqué dans le répertoire dont le nom doit être fourni dans <VarRésultat>. Si le numéro ordinal n'est pas autorisé, la valeur "NUL" est fournie dans <VarRésultat>.

Tab. 7-255: Syntaxe DIRINF

Valeur de retour :

<mode d'ordre 1>	Espace mémoire libre, en octets
<mode d'ordre 2>	Espace mémoire occupé, en octets
<mode d'ordre 3>	Nombre de fichiers existants dans le répertoire
<mode d'ordre 4>	0 (<VarRésultat> contient un nom de fichier) – ou – 1 (<VarRésultat> est NUL)
<mode d'ordre 5>	0 (<VarRésultat> contient un nom de répertoire) – ou – 1 (<VarRésultat> est NUL)
<mode d'ordre 6>	0 (<VarRésultat> contient un nom de répertoire) – ou – 1 (<VarRésultat> est NUL)

Ordres CPL

<mode d'ordre 7>	0 (<VarRésultat> contient un nom de répertoire) – ou – 1 (<VarRésultat> est NUL)
<mode d'ordre 8>	0 (<VarRésultat> contient un nom de fichier) – ou – 1 (<VarRésultat> est NUL)

Exemple : ; Afficher tous les noms de fichiers issus du répertoire
; actuel dans la fenêtre MSG

```

10 DIM FILENAME$(30)
20 DIM DIRNAME$(80)
30 LJUST
40 ERG%= DIRINF(5,,DIRNAME$)
45 IF ERG%=0 THEN
50 ANZ_FILES%= DIRINF(3,DIRNAME$)
60 FOR LNR%= 1 TO ANZ_FILES%
70 ERG%= DIRINF(4,DIRNAME$,FILENAME$,LNR%)
80 PRN#(0,LNR%,"": ",FILENAME$)
90 NEXT LNR%
95 ENDIF
M30

```

Exemple : ; Lire tous les répertoires mounted de la commande

```

10 LJUST
20 DIM T$(100)
30 FOR I%= 1 TO 100
40 ERG%= DIRINF(6,,T$,I%)
50 IF ERG%=0 THEN
60 PRN#(0,I%," : ",T$)
70 ELSE
80 PRN#(0,"Aucun autre répertoire mounted disponible !")
90 I%=100
100 ENDIF
110 NEXT I%
M30

```

DIRCR

INT	READ	WRITE	FUNCT	PROC	WAIT	ERRNO	!

Description : **DIrectory CReate** – Création du répertoire

DIRCR permet de créer un nouveau répertoire dans le programme CPL. L'ordre fournit une valeur de type "INTEGER".

Syntaxe :

DIRCR (<répertoire>)	
<répertoire>	Nom du répertoire avec chemin complet sous forme d'expression STRING. Si aucun chemin n'est indiqué, le répertoire actuel est mis en tête de l'expression STRING.

Tab. 7-256: Syntaxe DIRCR

Valeur de retour :

0 : Le répertoire a été créé.
1 : Impossible de créer le répertoire.

Exemple :

```
10 I% = DIRCR("/usr/user/test")
M30
```

DIRDEL

INT	READ	WRITE	FUNCT	PROC	WAIT	ERRNO	!

Description :

DIRectory DElete – Suppression du répertoire
DIRDEL permet de supprimer un répertoire vide dans le programme CPL. L'ordre fournit une valeur de type "INTEGER".

Syntaxe :

DIRDEL (<répertoire>)	
<répertoire>	Nom du répertoire avec chemin complet sous forme d'expression STRING.

Tab. 7-257: Syntaxe DIRDEL

Valeur de retour :

0 : Le répertoire a été supprimé.
1 : Impossible de supprimer le répertoire. Formatage identique à DIRCR.

Exemple :

```
10 I% = DIRDEL("/usr/user/test")
M30
```

7.5.3 Opérations de fichier

FILENO

INT	READ	WRITE	FUNCT	PROC	WAIT	ERRNO	!

Description :

FILE Numero – Numéro de fichier logique
Des numéros de fichiers logiques allant de 1 à 9 sont requis pour accéder aux fichiers, ce qui signifie qu'un maximum de 9 fichiers peuvent être ouverts simultanément via CPL. L'ordre FILENO permet d'interroger le numéro de fichier logique valide suivant. Si l'ordre renvoie la valeur "-1", cela signifie qu'aucun numéro de fichier logique n'est disponible. La valeur de retour est de type "INTEGER".

Exemple :

```
10 LOG_NR% = FILENO
20 IF (LOG_NR% <> (-1)) THEN
30 OPENW(LOG_NR%,"/user/usr/Test")
40 PRN#(LOG_NR%,"Date : ",DATE)
50 CLOSE(LOG_NR%)
60 ELSE
70 PRN#(0,"Aucun numéro log. libre n'est disponible !")
80 ENDIF
M30
```

Ordres CPL

OPENW, OPENR

---	READ	WRITE	FUNCT	PROC	WAIT	ERRNO	!

Description : **OPEN Write** – Ouvrir le fichier pour un accès en écriture

OPEN Read – Ouvrir le fichier pour un accès en lecture

L'ordre d'ouverture d'un fichier dépend du mode d'accès recherché :

- Accès en écriture : **OPENW**
- Accès en lecture : **OPENR**

Si le fichier à ouvrir n'existe pas, il sera créé lors de l'ouverture.

Il est possible d'ouvrir des fichiers en lecture qui sont déjà ouverts pour un accès en écriture : l'ordre "OPENR" doit être utilisé ici. Un fichier ouvert ne peut cependant pas être ouvert plusieurs fois pour un accès en écriture.

Pour ouvrir un fichier aléatoire, un paramètre supplémentaire définissant la longueur des enregistrements dans le fichier en octets (1 octet = longueur d'un caractère) est nécessaire. Sinon, la structure de l'ordre est identique à celle d'un fichier séquentiel.

Après un ordre OPENR, le pointeur du fichier est placé sur le premier enregistrement au niveau duquel l'accès en lecture est possible. Après un ordre OPENW, le pointeur du fichier est placé sur le pointeur EOF, soit après le dernier enregistrement du fichier.

Syntaxe :

OPENW (<n>,<nom de fichier>[,<param. vide>][,<longueur d'enregistrement>])	
OPENR (<n>,<nom de fichier>[,<longueur d'enregistrement>])	
<n>	Numéro logique sous lequel le fichier peut être adressé. Seules les valeurs de 1 à 9 peuvent être sélectionnées. Le numéro logique doit être programmé sous forme d'expression INTEGER. Un numéro logique ne doit pas être attribué simultanément pour la lecture et l'écriture d'un fichier. Au maximum, 9 fichiers peuvent être ouverts simultanément. Si la plage de valeurs n'est pas respectée, le message d'erreur "NUMÉRO LOGIQUE INVALIDE" s'affiche.
<nom de fichier>	Expression STRING : nom du fichier, en option avec chemin complet Si aucun chemin n'est indiqué, le fichier est recherché dans le répertoire actuel.
<param. vide>	(non attribué)
<longueur d'enregistrement>	Nombre d'octets d'un enregistrement Plage de valeurs : 1 à 1024 Si la plage de valeurs n'est pas respectée, le message d'erreur "LONGUEUR DE COMPOSANT INVALIDE" s'affiche.

Tab. 7-258: Syntaxe pour l'accès en lecture et en écriture

Exemples :

```

50 OPENW(1,"P500", "Ceci est mon meilleur programme")
40 A$="P500" : B1$="Ceci est mon meilleur programme"
50 OPENW(9,A$,B1$)
50 OPENW(7,"PDaten_Mes.DAT","Enregistrer les données mesurées")

```


Ordres CPL

Lors de l'ouverture du fichier en écriture, une vérification a lieu pour contrôler si la structure aléatoire reste maintenue. Si la structure a été détruite par l'éditeur, le message d'erreur "LONGUEUR DE COMPOSANT INVALIDE" s'affiche.

```
10 OPENW(2,"P200",,10)
20 FOR I% = 1 TO 3
30 PRN#(2,"TESTS")
40 NEXT I%
50 CLOSE(2)
```

Résultat : "P2"

```
TESTS <LF>
TESTS <LF>
TESTS <LF>
<ETX><LF>
```

Lors de l'ouverture du fichier en lecture, une vérification a lieu pour contrôler si la structure aléatoire reste maintenue. Tous les composants doivent avoir la même longueur définie dans l'ordre OPENR.

```
1 OPENW(2,"P200",,"TEST",10)
2 PRN#(2,"ABC")
3 CLOSE(2)
4 OPENR(1,"P2",5)
5 CLOSE(1)
```

Une vérification a lieu pour contrôler si la longueur d'enregistrement du fichier "P2" est égale à 5. La longueur d'enregistrement de ce fichier s'élève ici à 10.

P1 :

```
N10 G1F10000X1000Y1000Z1000
1 A$="01234567890123456789"
2 B$="TEST"
N20 X0
M30
```

P2 :

```
1 OPENW(1,"PMess_PRG",,"FICHIER_ALEATOIRE",10)
2 OPENR(2,"P1")
3 DIM A$(30)
4 FOR I% = 1 TO 5
5 INP#(2,A$)
6 PRN#(1,A$)
7 NEXT
8 CLOSE(2)
9 CLOSE(1)
```

RÉSULTAT : PMess_PRG :

Ordres CPL

```

N10 G1F100<LF>
1 A$="0123<LF>
2 B$="TEST<LF>
N20 X0<LF>
M30 <LF>
<ETX><LF><LF>

```

Si la structure a été détruite par l'éditeur, le message d'erreur "LONGUEUR DE COMPOSANT INVALIDE" s'affiche.

REWRITE

---	READ	WRITE	FUNCT	PROC	WAIT	ERRNO	!

Description : **REWRITE** – Écrasement du contenu du fichier

Si le fichier ouvert contient déjà des données, les nouvelles données sont normalement ajoutées aux données existantes lors de l'écriture. L'ordre "REWRITE" permet cependant de remplacer le contenu d'un fichier existant sans devoir supprimer au préalable le contenu qui n'est plus requis. Lors de l'écrasement, la zone réservée dans l'ordre OPENW est conservée dans la mémoire du programme pièce.

Syntaxe :

REWRITE (<n>)	
<n>	Numéro logique du fichier (plage de valeurs de 1 à 9)

Tab. 7-259: Syntaxe REWRITE

Un fichier doit être ouvert avant de pouvoir être écrasé.

CLOSE

---	READ	WRITE	FUNCT	PROC	WAIT	ERRNO	!

Description : "CLOSE" – Fermer un fichier

Au maximum, 9 fichiers peuvent être ouverts simultanément. Pour accéder à un fichier supplémentaire lorsque 9 fichiers sont déjà ouverts, il faut d'abord fermer un fichier.

C'est pourquoi, il convient généralement de refermer les fichiers ouverts dès que les opérations de lecture ou d'écriture sont terminées.

Syntaxe :

CLOSE (<n>)	
<n>	1 à 9 : Numéro logique du fichier qui doit être fermé.

Tab. 7-260: Syntaxe Close

Exemple :

```

:
90 DIM A$(35)
100 XPOS = MCS(1)
110 YPOS = MCS(2)
120 OPENW(1,"P5",500,"POSAXE")
130 REWRITE(1)
140 PRN#(1,"axe X",XPOS,YPOS,"axe Y",YPOS)
150 CLOSE(1)

```

```
160 OPENR(1,"P5")
170 INP#(1,A$)
180 CLOSE(1)
:
```

Dans cet exemple, les positions actuelles des axes X et Y sont transmises aux variables (lignes 90 à 110). Ensuite, le fichier 1 est ouvert et archivé en tant que programme pièce P5 (ligne 120).

Puis, le fichier est écrit ou écrasé avant d'être refermé (lignes 140 à 150). Le fichier est alors ouvert pour lecture et son contenu est affecté à la variable A\$. Une fois l'accès en lecture terminé, le fichier est refermé (lignes 160 à 180).

ERASE

Description :

ERASE – Suppression d'un fichier

Supprime le fichier du répertoire actuel.

Syntaxe :

INT	READ	WRITE	FUNCT	PROC	WAIT	ERRNO	!
ERASE (<nom de fichier>)>							
<nom de fichier>		Nom du fichier avec chemin complet sous forme d'expression STRING					
		Si aucun chemin n'est indiqué, le fichier est recherché dans le répertoire actuel. La valeur de retour fournie par l'ordre CPL est une valeur INTEGER.					

Tab. 7-261: Syntaxe Erase

L'ordre "ERASE" peut renvoyer les résultats suivants sous forme de valeur INTEGER, dans une variable INTEGER affectée ou dans une boucle et interrogation (WHILE, IF, etc.) :

- 0 : Le fichier a été supprimé.
 - 1 : Le fichier n'a pas été supprimé, car il n'existe pas.
 - 2 : Le fichier n'a pas été supprimé, car il est protégé contre la suppression.
 - 3 : Le fichier n'a pas été supprimé, car il est actif.
 - 4 : <nom de fichier> n'a pas été supprimé, car <nom de fichier> n'est pas un fichier mais un répertoire.
 - 5 : Le fichier n'a pas été supprimé, car il est ouvert.
- Si un fichier ne peut pas être supprimé, l'avertissement correspondant est émis et le traitement du programme se poursuit.

Exemples :

```
10 IF ERASE("P1") <> 0 THEN ...
10 I% = ERASE("P1")
10 WHILE ERASE("P1") <> 0 DO ...
```

```
10 OPENW(1,"P2",200)
11 OPENW(2,"P3",200)
20 PRN#(1,"TEST1 POUR ERASE")
21 PRN#(2,"TEST2 POUR ERASE")
31 CLOSE(1) 32 CLOSE(2)
40 ERASE("P2")
```

Ordres CPL

```
43 A$="P3"
44 ERASE (A$)
```

FILECOPY

---	READ	WRITE	FUNCT	PROC	WAIT	ERRNO	!

Description : **FILE COPY** – Copie d'un fichier

FILECOPY permet de copier un fichier dans le programme CPL.

Syntaxe :

FILECOPY (<fichier source>, <fichier cible>)	
<fichier source>	Nom de fichier, y compris le chemin, du fichier source sous forme d'expression STRING Si aucun chemin n'est indiqué, le fichier est recherché dans le répertoire actuel.
<fichier cible>	Nom de fichier, y compris le chemin, du fichier cible sous forme d'expression STRING Si aucun chemin n'est indiqué, le fichier est créé dans le répertoire actuel.
ERRNO	Variable CPL programmable à n'importe quel emplacement. Avec ERRNO, aucune erreur d'exécution n'est générée en cas d'erreur ; les valeurs de retour possibles sont :
0 :	Accès OK
-8 :	Nom du fichier source, y compris le chemin, trop long
-9 :	Impossible d'accéder au fichier source
-10 :	Nom du fichier cible, y compris le chemin, trop long
-11 :	Nom du fichier (source ou cible) inadmissible
-12 :	Copie impossible

Tab. 7-262: Syntaxe FILECOPY

Exemple : *Programme:*

```
10 FILECOPY("/usr/user/Test.txt", "/usr/user/Test.bak", ERRNO)
20 IF ERRNO = 0 THEN
30 PRN#(0,"Phase de copie ok")
40 ENDIF
```

7.5.4 Traitement d'un fichier

PRN#

---	READ	WRITE	FUNCT	PROC	WAIT	ERRNO	!

Description **PRN#** – Écriture dans un fichier

Le type de variable peut être sélectionné librement. Des variables indexées et des tableaux peuvent également être utilisés. Il est également possible de programmer des expressions REAL double précision sous forme d'expressions CPL quelconques.

Si le résultat d'une expression doit être édité avec un certain format, au moins une des expressions doit être de type "STRING". Le format peut être défini dans cette chaîne de format à l'aide des caractères "#" et ".". Les résultats sont inscrits à la place de l'instruction de format définie par "#". La première instruction de format contenue dans une expression STRING se réfère à la première expression suivante qui peut être éditée avec une indication du format. Les expressions booléennes ne peuvent pas être formatées. Le nombre de toutes les indications de format programmées doit être inférieur ou

égal au nombre d'expressions devant être éditées. Si cette condition n'est pas remplie, les "#" en excédent sont affichés. Si aucun format n'est indiqué, l'expression est éditée au format standard.

	Format standard	Nombre d'emplacements
Nombre INTEGER	9 positions	9 au maximum
Nombre REAL simple précision	4 chiffres avant et 3 chiffres après la virgule	7 au maximum
Nombre REAL double précision	9 chiffres avant et 6 chiffres après la virgule	15 au maximum

Si la sortie d'une expression est supérieure à 1024 caractères, le message d'erreur "BLOC SUPÉRIEUR À 1024 OCTETS" s'affiche.

Si le résultat ne peut pas être représenté au format indiqué, l'avertissement "FORMAT PRN INCORRECT" est émis, et des caractères "***" sont édités à la place du format erroné.

Si des caractères "#" doivent être créés dans le fichier lui-même, aucune expression à formater ne doit suivre la chaîne dans l'instruction PRN#. Le caractère "#" peut également être édité avec CHR\$(35).

CHR\$(13) permet de réaliser l'avance de ligne au cours de l'édition, c.-à-d. que l'édition de l'ordre PRN# se poursuit à la ligne suivante, c.-à-d. dans l'enregistrement suivant.

L'ordre CHR\$() permet également de transmettre d'autres signes de commande, par exemple en cas de sortie par l'intermédiaire d'une interface série.

Syntaxe :

PRN# (<n>, [<expression>][, <expression>][, <expression>][, ...][:])	
<n>	1 à 9 : Numéro logique du fichier dans lequel les données seront écrites. 0 : La sortie est déviée sur l'écran (comme pour la programmation des remarques à l'aide de l'ordre MSG).
<expression>	Caractères alphanumériques quelconques (texte entre guillemets), chaîne de format ou variable dont le contenu doit être enregistré/affiché.
;	Écrase l'ajout automatique de <CR><LF>. Les points suivants s'appliquent à un enregistrement écrasé via PRN# : Ordre PRN# avec point-virgule : Si la longueur des nouvelles données à écrire est inférieure à la longueur des anciennes données, les nouvelles données sont insérées et le reste des anciennes données est conservé. Ordre PRN# sans point-virgule : Si la longueur des nouvelles données à écrire est inférieure à la longueur des anciennes données, les nouvelles données sont insérées et le reste des anciennes données est écrasé par des espaces.

Tab. 7-263: Syntaxe PRN#

Exemples : **Ordre PRN# avec point-virgule**

Ordres CPL

```

1 OPENW(2,"PProg123.PRG",200,35)
2 PRN#(2,"TEST1 POUR ORDRE PRN AVEC POINT-VIRGULE")
3 PRN#(2,"TEST2 POUR ORDRE PRN AVEC POINT-VIRGULE")
4 PRN#(2,"TEST3 POUR ORDRE PRN AVEC POINT-VIRGULE")
6 SEEK(2,1)
7 PRN#(2,"ECRASER";)
8 CLOSE(2)

```

RÉSULTAT dans PProg123.PRG :

```

ORDRE ÉCRASER AVEC POINT-VIRGULE      <LF>
TEST2 POUR ORDRE PRN AVEC POINT-VIRGULE <LF>
TEST3 POUR ORDRE PRN AVEC POINT-VIRGULE <LF>
<ETX><LF>

```

Ordre PRN# sans point-virgule

```

1 OPENW(2,"P2",1000,36) 1 REWRITE(2)
2 PRN#(2,"TEST1 POUR ORDRE PRN SANS POINT-VIRGULE")
3 PRN#(2,"TEST2 POUR ORDRE PRN SANS POINT-VIRGULE")
4 PRN#(2,"TEST3 POUR ORDRE PRN SANS POINT-VIRGULE")
6 SEEK(2,1)
7 PRN#(2,"ECRASER")
8 CLOSE(2)

```

RÉSULTAT dans P2 :

```

ECRASER                                <LF>
TEST2 POUR ORDRE PRN SANS POINT-VIRGULE <LF>
TEST3 POUR ORDRE PRN SANS POINT-VIRGULE <LF>
<ETX><LF>

```

<ETX><LF> est ajouté après le dernier bloc du fichier.

Si la longueur de bloc de 1024 caractères est dépassée, le message d'erreur "BLOC SUPÉRIEUR À 1024 OCTETS" s'affiche.

Si la fin du fichier est atteinte lors de l'écriture d'un fichier séquentiel, alors le fichier est automatiquement copié et la plage réservée est augmentée de la longueur attribuée dans la mesure où l'espace mémoire du programme pièce le permet.

Étant donné que cela utilise beaucoup d'espace mémoire, il convient de réserver une longueur de fichier suffisante avec "OPENW" lors de la création du fichier.

```

1 OPENW(1,"P2",300,"TEST ORDRE PRN")
2 A$="TEST"
3 B$="POUR"
4 C$="ORDRE PRN"
5 PRN#(1,A$)
6 PRN#(1,B$)
7 PRN#(1,C$)

```

```
8 PRN# (1,A$;)
9 PRN# (1,B$;)
10 PRN# (1,C$;)
11 CLOSE(1)
```

Résultat :

P2 :

TEST<LF>

POUR<LF>

ORDRE PRN<LF>

TESTPOURORDREPRN<LF><ETX><LF>

```
10 DIM E$(50)
20 OPENW(1,"P2",300,"TEST2")
30 A% = 5000
40 R = 1.231
50 B! = 4/3
60 D$ = "ABCDE"
70 E$ = "CDEFGHI"
80 PRN# (1,"10";)
90 PRN# (1,"#####", "###.###", "#.#####", A%, R, B!, D$, E$)
95 CLOSE(1)
```

P2 :

```
10  5000  1.2311.33333ABCDECDEFGHI<LF>
    A%    R    B!    D$    E$
```

LJUST, NJUST

---	READ	WRITE	FUNCT	PROC	WAIT	ERRNO	!

Description : **Left JUSTify** – Formatage justifié à gauche

No JUSTify – Pas de formatage

LJUST permet de passer à l'alignement à gauche de la sortie des données. Elle agit sur toutes les sorties du fichier jusqu'à la fin de l'exécution du programme.

NJUST permet de repasser de manière prématurée à l'édition formatée.

Lors de l'édition de données dans des fichiers, le type de données "REAL" peut comporter un maximum de 7 positions (4 positions avant la virgule et 3 positions après) et le type de données "INTEGER" peut comporter un maximum de 9 positions. Les zéros de tête et de queue sont supprimés. Ceci s'applique aussi à la sortie justifiée à gauche.

LJUST permet de générer directement les programmes CN avec CPL qui peuvent être exécutés en mode de fonctionnement "TRAITEMENT", étant donné que les caractères sont supprimés entre l'adresse CN et la valeur.

INP#

---	READ	WRITE	FUNCT	PROC	WAIT	ERRNO	!

Ordres CPL

Description : **INPut#** – Lecture à partir d'un fichier

L'instruction INP# permet de lire les données ASCII d'un fichier ouvert sous forme d'enregistrements et de les attribuer à une ou plusieurs variables. L'ordre n'agit que sur les fichiers qui ont été ouverts avec "OPENR(..)".

Syntaxe :

INP# (<n> , <variable>[, <variable>][...];)	
<n>	1 à 9 : Numéro logique du fichier à partir duquel les données doivent être lues.
<variable>	Variable dans laquelle les données lues sont enregistrées.
;	Si un point-virgule est programmé, le pointeur du fichier reste dans l'enregistrement jusqu'à ce que sa fin soit atteinte. Il passe ensuite à l'enregistrement suivant. La lecture n'y est toutefois pas poursuivie automatiquement. Si aucun point-virgule n'est programmé, le pointeur passe automatiquement à l'enregistrement suivant.

Tab. 7-264: Syntaxe INP#

Le type de variable peut être sélectionné librement. Des variables indexées et des tableaux peuvent également être utilisés. Si une valeur différente de TRUE ou FALSE est attribuée à une variable, alors la valeur "NUL" lui est attribuée.

En cas des variables INTEGER ou REAL (simple ou double précision), les caractères "0" à "9", les signes "-", "+", les zéros en tête ou les espaces sont convertis en valeurs INTEGER ou REAL. Si un autre caractère est affecté à une variable INTEGER ou REAL, la valeur "NUL" est attribuée à la variable. Si la valeur "NUL" est attribuée à une variable, sa position ne change pas au sein du fichier.

Si une valeur trop élevée est affectée à une valeur INTEGER ou REAL, un message d'erreur correspondant apparaît :

"VALEUR ENTIER INVALIDE "

"VALEUR FLOTTANTE INVALIDE"

Exemples : **Instruction INP#**

P2 :

ABC 123456789 ABC

P3 :

1 OPENR(2,"P2") 2 DIM C\$(3) 3 DIM D\$(3) 4 INP#(2,I
%,J,L?,C\$,K%,D\$) 5 CLOSE(2)

RÉSULTAT :

I% = NUL

J = NUL

L? = NUL

C\$ = "ABC"

K% = 123456789

D\$ = "ABC"

Lecture d'un enregistrement à partir d'un fichier

1 OPENW(1,"P2",200,"Test",22)


```

2 PRN#(1,"-12TEST1.23V12ABCD2.4A")
3 PRN#(1,"-12TEST1.23V12ABCD2.4A")
4 PRN#(1,"-12TEST1.23V12ABCD2.4A")
5 CLOSE(1)
6 DIM A$(3)
7 DIM C$(5)
8 DIM D$(4)
9 DIM E$(4)
10 DIM G$(25)
11 DIM H$(7)
12 DIM I$(7)
13 DIM J$(25)
14 DIM R(1,2)
15 OPENR(2,"P2",22)
16 INP#(2,B%,D$,R(1,1),MID$(E$,1,1),R(1,2),A$,C$)
17 INP#(2,G$)
18 INP#(2,H$;)
19 INP#(2,I$;)
20 INP#(2,J$)
21 CLOSE(2)

```

RÉSULTAT :

```

B% = -12
D$ = "TEST" , car la longueur max. du tableau = 4
R(1,1) = 1.230
E$ = "V"
R(1,2) = 12.000
A$ = "ABC" , car la longueur max. du tableau = 3
C$ = "D2.4A"
G$ = "-12TEST1.23V12ABCD2.4A"
H$ = "-12TEST" , car la longueur max. du tableau = 7
I$ = "1.23V12" , car la longueur max. du tableau = 7
J$ = "ABCD2.4A"

```

SEEK

---	READ	WRITE	FUNCT	PROC	WAIT	ERRNO	!

Description : **SEEK-** Positionnement du pointeur de fichier

Positionne le pointeur de fichier à un emplacement précis du fichier ouvert. Il peut s'agir aussi bien d'un fichier séquentiel que d'un fichier aléatoire.

Les fichiers séquentiels doivent être ouverts avec l'ordre "OPENR(..)". Pour les fichiers aléatoires, l'ordre "OPENW(..)" est également admissible.

Ordres CPL

Syntaxe :

SEEK (<n> , <k>[, <o>])	
<n>	Numéro logique du fichier dans lequel le pointeur du fichier doit être positionné. Plage de valeurs : 1 à 9 Si la plage de valeurs n'est pas respectée, le message d'erreur "NUMÉRO DE FICHIER INVALIDE" s'affiche.
<k>	Numéro d'enregistrement d'un fichier aléatoire ou nombre d'octets d'un fichier séquentiel. Le pointeur du fichier est positionné sur <k>. Plage de valeurs : 0 jusqu'au dernier enregistrement existant ou 0 jusqu'au dernier octet existant. Le dernier enregistrement existant est celui avec le pointeur EOF. Si la valeur est égale à 0, le positionnement s'effectue sur le pointeur EOF. Si la plage de valeurs n'est pas respectée ou si l'enregistrement indiqué n'existe pas, le message d'erreur "COMPOSANT INVALIDE" s'affiche.
<o>	Décalage d'enregistrement Indique sur quel octet au sein d'un enregistrement le pointeur du fichier doit être positionné. Plage de valeurs : 1 à longueur d'enregistrement + 1 Si le décalage d'enregistrement n'est pas programmé pour les fichiers aléatoires, le pointeur de fichier est positionné sur le 1er octet de l'enregistrement <k>. Si la plage de valeurs n'est pas respectée, le message d'erreur "PARAMÈTRE INADMISSIBLE" s'affiche. Ce paramètre est autorisé uniquement pour les fichiers aléatoires. S'il est tout de même programmé pour un fichier séquentiel (ouvert en lecture), le message d'erreur "PARAMÈTRE INADMISSIBLE" s'affiche.

Tab. 7-265: Syntaxe SEEK

Exemples : SEEK et fichier séquentiel

```
1      DIM A$(1):LJUST:OPENW(1,"P271",130,"TEST"):FOR I%=1
      TO 10:
      PRN#(1,"!/-/!/-/!/-/!/-/!/-/!/-/!/-/!/-/!/-/!/-/!/"):NEXT:
      CLOSE(1):OPENR(2,"P271"):FOR I%=1 TO FILESIZE(2,2)-28:
      IF NOT (EOF(2)) THEN SEEK(2,I%):INP#(2,A$) ENDIF: IF
      (EOF(2)) THEN PRN#(0,"###",I%,". BYTE: <EOF>"): ELSE
      PRN#(0,"###",I%,". BYTE: <",<A$,<>>) ENDIF: NEXT
      I%:CLOSE(2)
```

M30

SEEK et fichier aléatoire

```
1      OPENW(1,27272,200,"TEST",1024):LJUST
2
3      FOR I%= 1 TO 10
4      PRN#(1,I%,"e enregistrement")
5      NEXT
```

```

5      SEEK(1,3,4) : REM positionné sur le 4e octet dans le 3e enregistrement
6      PRN#(1,"Ce texte écrase le 3e enregistrement à partir de l'octet 4")
7      SEEK(1,11):PRN#(1,"11e enregistrement")
8      SEEK(1,11,5):PRN#(1,"@@")
9      SEEK(1,0):PRN#(1,"<EOF>")
10     SEEK(1,0,1):PRN#(1,"nouveau <EOF>")
11     CLOSE(1)

```

FILEPOS

INT	READ	WRITE	FUNCT	PROC	WAIT	ERRNO	!

Description : **FILE POSition** – Définition de la position du pointeur de fichier

L'ordre "FILEPOS()" fournit le numéro d'enregistrement de l'enregistrement actuel d'un fichier aléatoire auquel on pourra accéder plus tard. De plus, il est possible de déterminer le décalage d'enregistrement au sein de l'enregistrement actuel (pour un fichier aléatoire) ou le décalage d'enregistrement par rapport à l'octet actuel (pour un fichier séquentiel) auquel on peut accéder. Le fichier peut aussi bien être un fichier séquentiel qu'un fichier aléatoire.

On entend par décalage le nombre d'octets dans le fichier, du début du fichier jusqu'à l'octet actuel. Le décalage d'enregistrement indique sur quel octet d'un enregistrement le pointeur doit être positionné. Le décalage d'enregistrement commence avec la valeur 1 (= 1er octet d'un enregistrement) et peut accepter au maximum la valeur de la longueur d'enregistrement + 1 (le dernier octet de cet enregistrement est <LF>). La valeur 1 est renvoyée si l'on se trouve sur le pointeur EOF.

Ordres CPL

Syntaxe :

FILEPOS (<n>[, <mode>])	
<n>	1 à 9 : Numéro logique du fichier dans lequel la position du pointeur de fichier doit être lue. Si la plage de valeurs n'est pas respectée, le message d'erreur "NUMÉRO DE FICHIER INVALIDE" s'affiche.
<mode>	Pour les fichiers aléatoires : Plage de valeurs : 1 à 3 <mode> = 1 : Fournit le décalage sur l'octet actuel qui peut être lu ou écrit. <mode> = 2 : Fournit le numéro de l'enregistrement actuel qui peut être lu ou écrit. Si l'on se trouve sur le pointeur EOF, le résultat est le suivant : nombre d'enregistrements + 1. <mode> = 3 : Fournit le décalage de l'enregistrement au sein de l'enregistrement actuel qui peut être lu ou écrit. Le décalage d'enregistrement commence avec la valeur 1 (⇒ 1er octet de cet enregistrement) et peut accepter au maximum la valeur de l'enregistrement + 1 (⇒ le dernier octet de cet enregistrement est <LF>). Si l'on se trouve sur le pointeur EOF, la valeur 1 est renvoyée et le fichier ne peut pas être lu. <mode> non programmé : Fournit le numéro de l'enregistrement actuel qui peut être lu ou écrit. Si l'on se trouve sur le pointeur EOF, le résultat est le suivant : nombre d'enregistrements + 1. Pour les fichiers séquentiels : Plage de valeurs : 1 <mode> = 1 ou non programmé : Fournit le décalage sur l'octet actuel qui peut être lu ou écrit.

Tab. 7-266: Syntaxe FILEPOS



Si la plage de valeurs de <mode> n'est pas respectée, le message d'erreur "PARAMÈTRE INADMISSIBLE" s'affiche.

Exemples :

FILEPOS et fichier séquentiel

```

1 OPENW(1,"P2",200,"TEST")
2 FOR I%= 1 TO 10
3 PRN#(1,"TEST POUR FILEPOS")
4 NEXT
5 CLOSE(1)
6 OPENR(1,"P2")
7 SEEK(1,3)
8 POS% = FILEPOS(1)
9 POS1% = FILEPOS(1,1)
11 SEEK(1,0) : POSITIONNER REM À LA FIN DU FICHIER

```

```
12 POS2% = FILEPOS(1)
13 POS3% = FILEPOS(1,1)
14 CLOSE(1)
```

Résultat :

POS% = 3	→ Nombre d'octets
POS1% = 3	→ Nombre d'octets
POS2% = 171	→ Nombre d'octets
POS3% = 171	→ Nombre d'octets

FILEPOS et fichier aléatoire

Programme:

```
1 OPENW(1,"P2",200,"TEST",1024)
2 FOR I% = 1 TO 10
3 PRN#(1,"TEST POUR FILEPOS")
4 NEXT
5 SEEK(1,3,2)
6 POS% = FILEPOS(1)
7 POS1% = FILEPOS(1,1)
8 POS2% = FILEPOS(1,2)
9 POS3% = FILEPOS(1,3)
10 PRN#(1,"Écrasement du 3e enregistrement à partir de l'octet 2 avec ce texte")
11 SEEK(1,0) : REM positionné sur fin de fichier
12 POS% = FILEPOS(1)
13 POS1% = FILEPOS(1,1)
14 POS2% = FILEPOS(1,2)
15 POS3% = FILEPOS(1,3)
16 CLOSE(1)
```

Résultat :

POS% = 3	→ Numéro de l'enregistrement dans lequel on se trouve.
POS1% = 258	→ Nombre d'octets
POS2% = 3	→ Numéro de l'enregistrement dans lequel on se trouve.
POS3% = 2	→ Position au sein du 3e enregistrement
POS% = 11	→ Numéro de l'enregistrement dans lequel on se trouve.
POS1% = 1281	POS1% = 1281
POS2% = 11	→ Numéro de l'enregistrement dans lequel on se trouve.
POS3% = 1	→ Position au sein du 3e enregistrement

EOF

BOOL	READ	WRITE	FUNCT	PROC	WAIT	ERRNO	!

Description : **End Of File** – Identification de la fin du fichier

L'ordre "EOF" permet d'interroger si la fin d'un fichier est atteinte.

L'ordre EOF renvoie la valeur logique "TRUE" lorsque la fin du fichier est atteinte au cours d'un accès en lecture. Dans le cas contraire, la valeur renvoyée est "FALSE".

Exemple :

```
: 9 DIM A$(10)
10 OPENR(1,"P",444) : I%=0
11 WHILE NOT (EOF(1)) DO
```

Ordres CPL

```

12 INP# (1,A$)
13 I%=I%+1
14 END
15 CLOSE (1)
M30

```

7.5.5 Propriétés du fichier

FILEACCESS

INT	READ	WRITE	FUNCT	PROC	WAIT	ERRNO	!

Description : **FILE ACCESS** – Droits d'accès du fichier

FILEACCESS permet de déterminer dans le programme CPL si un fichier existe et de quels droits d'accès il dispose.

Syntaxe :

FILEACCESS (<nom de fichier>)	
<nom de fichier>	Nom du fichier avec chemin complet sous forme d'expression STRING Si aucun chemin n'est indiqué, le fichier est recherché dans le répertoire actuel. La valeur de retour fournie par l'ordre CPL est une valeur INTEGER.
-1 :	Le fichier n'existe pas.
0 :	Fichier sans droits d'accès
Sinon :	Droits d'accès codés en binaire :
Bit0 :	Exécution possible (X)
Bit1 :	Écriture autorisée (W)
Bit2 :	Lecture autorisée (R)
Bit3 :	Le fichier est un répertoire (D).
Bit4 :	Le fichier est un programme actif (A).

Tab. 7-267: Syntaxe FILEACCESS

*Un **programme actif** est un fichier :*

- traité en tant que programme sur un canal ;
- traité en tant que sous-programme d'un programme traité sur un canal ;
- ouvert sur un canal via un ordre CPL.

Si les **droits d'accès** à une table de décalage d'origine ou à une table de correction d'outil actuellement utilisée dans un programme pièce sont interrogés par l'ordre "FILEACCESS", le bit 5 n'est **pas** défini.

Exemple : 10 I% = FILEACCESS ("/usrfep/test.cnc")

FILEDATE

CHAR	READ	WRITE	FUNCT	PROC	WAIT	ERRNO	!

Description : **FILE DATE** – Date

FILEDATE permet de déterminer la date d'un fichier dans le programme CPL.

Une erreur d'accès ne génère aucune erreur du programme pièce, mais l'ordre fournit une chaîne vide à la place.

Syntaxe :

FILEDATE (<nom de fichier>[, <DataSelection>])	
<nom de fichier>	Nom du fichier avec chemin complet sous forme d'expression STRING Si aucun chemin n'est indiqué, le fichier est recherché dans le répertoire actuel. La valeur renvoyée par l'ordre CPL est une expression STRING.
<DataSelection>	Variable INTEGER du mode d'ordre (par défaut = 1) : 1 = date du fichier, format : jj:mm:aa 2 = heure du fichier, format : hh:mm:ss

Tab. 7-268: Syntaxe FILEDATE

Exemple :

```

10    DIM DATUM$(10)
20    DATUM$ = FILEDATE("/usr/user/Test.txt",1)
30    IF LEN(DATE$)>0 THEN
40    PRN#(0,"Date du fichier : ",DATE$)
50    ENDIF
    
```

FILESIZE

INT	READ	WRITE	FUNCT	PROC	WAIT	ERRNO	!

Description :

FILE SIZE – Taille du fichier

Indique la taille d'un fichier ou la limite jusqu'à laquelle un fichier a déjà été écrit. Il peut s'agir aussi bien d'un fichier séquentiel que d'un fichier aléatoire. L'ordre n'agit que sur les fichiers qui ont été ouverts avec "OPENR(..)".

Ordres CPL

Syntaxe :

FILESIZE (<n>[, <k>])	
<n>	1 à 9 Numéro logique du fichier dont la taille doit être déterminée. Si la plage de valeurs est erronée, le message d'erreur "NUMÉRO DE FICHIER INVALIDE" s'affiche.
<k>	Pour les fichiers aléatoires : Plage de valeurs : 1 à 4 Pour les fichiers séquentiels : Plage de valeurs : 1 à 2 <k> = 1 : Taille totale de l'espace mémoire occupé par un fichier, en octets. <k> = 2 : Taille de l'espace mémoire occupé, du début de la plage de données jusqu'au pointeur EOF, en octets (taille du pointeur EOF non comprise). <k> = 3 : Nombre maximal d'enregistrements contenus dans un fichier. Ce résultat dépend de la longueur d'enregistrement avec laquelle le fichier a été ouvert. <k> = 4 : Nombre d'enregistrements, du début du fichier jusqu'au pointeur EOF. Ce résultat dépend de la longueur d'enregistrement avec laquelle le fichier a été ouvert. <k> non programmé : identique à <k> = 1 Si la plage de valeurs pour <k> est erronée, le message d'erreur "PARAMÈTRE INADMISSIBLE" s'affiche.

Tab. 7-269: Syntaxe FILESIZE

Exemples : FILESIZE et fichier séquentiel

1	OPENW (1, 2, 1000)
2	FOR I%= 1 TO 20
3	PRN# (1, "TESTS FILESIZE")
4	NEXT
5	CLOSE (1)
6	OPENR (2, 2)
7	A%=FILESIZE (2)
9	B%=FILESIZE (2, 1)
10	C%=FILESIZE (2, 2)
11	CLOSE (2)

La variable INTEGER A% contient la valeur : 302
La variable INTEGER B% contient la valeur : 302
La variable INTEGER C% contient la valeur : 300

FILESIZE et fichier aléatoire


```

1      OPENW(1,"P2",1000,10)
2      FOR I%= 1 TO 20
3      PRN#(1,"TESTS FILESIZE")
4      NEXT
5      CLOSE(1)
6      OPENR(2,2,10)
7      A%=FILESIZE(2)
9      B%=FILESIZE(2,1)
10     C%=FILESIZE(2,2)
10     D=FILESIZE(2,3)
10     E%=FILESIZE(2,4)
11     CLOSE(2)
    
```

La variable INTEGER A% contient la valeur : 222

La variable INTEGER B% contient la valeur : 222

La variable INTEGER C% contient la valeur : 220

La variable INTEGER D% contient la valeur : 20

La variable INTEGER E% contient la valeur : 20

7.6 Traitement des chaînes de caractères

7.6.1 Notions de base

DIM

---	READ	WRITE	FUNCT	PROC	WAIT	ERRNO	!

Description : **DIMension** – Dimensionner une chaîne de caractères

Pour créer un tableau, une variable CHARACTER doit être indexée via l'instruction DIM. De cette manière, il est possible de créer des tableaux de caractères d'une capacité maximale de 1024 caractères (plage de valeurs de l'index : 1 à 1024). Si la plage de valeurs n'est pas respectée, le message d'erreur "LIMITE DE CHAMP INVALIDE" s'affiche.

Exemples : 1 DIM VWX\$(14)

Dans cet exemple, le tableau VWX\$ composé de 14 variables CHARACTER individuelles est créé. Il est alors possible d'enregistrer des chaînes de caractères ayant une longueur maximale de 14 caractères dans VWX\$.

1 DIM ABC\$(1)	Tableau pour une chaîne de caractères ayant une longueur maximale de 1 caractère.
2 DIM BCDE\$(10)	Tableau pour une chaîne de caractères ayant une longueur maximale de 10 caractères.

INSTR

INT	READ	WRITE	FUNCT	PROC	WAIT	ERRNO	!

Description **IN STRing** – Rechercher dans une chaîne de caractères
 Parcourt une expression STRING selon un ordre prédéfini.

Ordres CPL

Syntaxe :	INSTR (<chaîne de caractères>,<expression STRING>[,<point de départ>])
	Parcourt une <expression SRING> à partir du <point de départ> après une <chaîne de caractères> et indique la position du premier caractère de la <chaîne de caractères> trouvée dans l'<expression STRING> sous forme de valeur INTEGER. Si la <chaîne de caractères> est introuvable, la valeur 0 est éditée. La <chaîne de caractères> peut être programmée en tant qu'expression STRING. La plage de valeurs pour le 3e paramètre est comprise entre 1 et 1024. Si la plage de valeurs n'est pas respectée, le message d'erreur "PARAMÈTRE INADMISSIBLE" s'affiche.

Tab. 7-270: Syntaxe INSTR

Exemple :	1 DIM A\$(8)	
	2 DIM B\$(16)	
	3 A\$="A" : MID\$(A\$,2)="UVWXYZ"	
	4 B\$="ABCDEF UVWXYZ GH"	
	5 POS1%=INSTR(MID\$(A\$,2),B\$,4)	La variable INTEGER POS1% a la valeur 8.
	6 POS2%=INSTR(MID\$(A\$,2,4),B\$,10)	La variable INTEGER POS2% a la valeur 0.
	7 POS3%=INSTR(MID\$(A\$,2),B\$)	La variable INTEGER POS3% a la valeur 8.

LEN

INT	READ	WRITE	FUNCT	PROC	WAIT	ERRNO	!

Description : LENGTH – Longueur d’une chaîne de caractères

Syntaxe :	LENLEN (<expression STRING>)
	Fournit le nombre de caractères d'une <expression STRING>. Le résultat est une valeur INTEGER. Si l'<expression STRING> est vide, LEN renvoie la valeur 0. Si l'<expression STRING> n'est pas définie, LEN renvoie la valeur -1.

Tab. 7-271: Syntaxe LEN

Exemple :	1 DIM XYZ\$(10)	
	2 XYZ\$="ABC"	
	3 I%=LEN(XYZ\$)	La variable INTEGER I% a la valeur 3.
	4 XYZ\$=" "	
	5 J%=LEN(XYZ\$)	La variable INTEGER J% a la valeur 0.

```
6 XYZ$=NUL
```

```
7 K%=LEN(XYZ$)
```

La variable INTEGER K% a
la valeur -1.

Comparaison d'expressions STRING

Description : Il est aussi possible de comparer des expressions STRING, c.-à-d. des constantes STRING, des variables STRING et des tableaux, entre eux.

Exemple :

```
1 DIM A$(10)
2 DIM B$(15)
3 A$="ANTON"
4 B$= "WILLI"
5 C$="ABCDE"
6 D$="VWXYZ"
7 IF A$ < B$ THEN ...
8 IF MID$(A$,2,3) = MID$(B$,1,3) THEN ...
9 Z?=A$ <> "TESTS"
10 IF "A" <= "C" THEN ...
11 IF C$ > D$ THEN ...
12 IF A$ = C$ THEN ...
13 IF "TE" < MID$(D$,2,2) THEN ...
```

L'ordre alphabétique des contenus des expressions STRING est contrôlé selon le codage ASCII de chaque caractère.

7.6.2 Conversion de type

ASC

INT	READ	WRITE	FUNCT	PROC	WAIT	ERRNO	!

Description : **AScii** – Définit le code ASCII d'un caractère

Syntaxe :

ASC (<chaîne de caractères>)	
	Édite le nombre ordinal du premier caractère (code ASCII) de la <chaîne de caractères> en tant que valeur INTEGER. Si la <chaîne de caractères> est vide ou non définie, ASC renvoie la valeur -1. La <chaîne de caractères> doit être une expression STRING. ASC est l'inversion de CHR\$.

Tab. 7-272: Syntaxe ASC

Exemple :

```
10 DIM A$(1)
20 A$ = "ABC"
30 B$ = "BCD"
40 I% = ASC(A$)
50 J% = ASC(B$)
```

La variable INTEGER I% a
la valeur 65.

La variable INTEGER J% a
la valeur 66.

Ordres CPL

```
60 A$ = ""
70 K% = ASC (A$)           La variable INTEGER K% a
                             la valeur -1.
80 A$ = NUL
90 L% = ASC (A$)           La variable INTEGER L% a
                             la valeur -1.
```

CHR\$

CHAR	READ	WRITE	FUNCT	PROC	WAIT	ERRNO	!

Description : **CHaRacter** – Détermination des caractères ASCII
CHR\$ est l'ordre inverse d'ASC.

Syntaxe :	CHR\$ (<expression INTEGER>)
	Convertit l'<expression INTEGER> au caractère ASCII correspondant.

Tab. 7-273: *Syntaxe CHR\$*
Vous trouverez toutes les valeurs décimales des caractères ASCII énumérés dans le tableau "Jeu de caractères ASCII" joint en annexe à la présente documentation.

Exemple :

```
10 DIM A$ (1)
20 I% = 65
40 A$ = CHR$ (I%)           La variable STRING A$ a la
                             valeur 65.
```

STR\$

BOOL	READ	WRITE	FUNCT	PROC	WAIT	ERRNO	!

Description **STRing** – Conversion d'un chiffre en une chaîne de caractères

Syntaxe :	STR\$ ([<chaîne de format> ,]<valeur>)
	Convertit l'expression numérique <valeur> en une chaîne de caractères qui peut être affectée uniquement à un tableau. L'affectation à une variable STRING entraîne une erreur d'exécution. La <valeur> peut être une expression REAL ou INTEGER simple ou double précision. Si la <chaîne de format> est programmée, la chaîne de caractères peut être éditée de manière formatée. Les chiffres sont représentés par le caractère "#" et les signes décimaux par un ".". Sans <chaîne de format>, l'édition s'effectue au format standard.

Tab. 7-274: *Syntaxe STR\$*

	Format standard	Nombre d'emplacements
Nombre INTEGER	9 positions	9 au maximum

Ordres CPL

Nombre REAL simple précision	4 chiffres avant et 3 chiffres après la virgule	7 au maximum
Nombre REAL double précision	9 chiffres avant et 6 chiffres après la virgule	15 au maximum

Exemple :

```
10 DIM A$(50)
20 DIM B$(21)
30 A$=STR$( "Chiffre=##.###", 37/3)
40 B$=STR$(2.5)
```

Contenu REM du tableau A\$:
"Chiffre=12.333"

Contenu du tableau B\$: "2 500"

VAL

INT	READ	WRITE	FUNCT	PROC	WAIT	ERRNO	!

Description

VALue – Conversion d'une chaîne de caractères en chiffre

Syntaxe :

VAL (<expression STRING>)	
	Fournit la valeur numérique d'une <expression STRING>. Si un caractère différent d'une espace de tête, des caractères de tête "+" ou "-", des chiffres 0 à 9 et du signe décimal "." apparaît, la conversion est exécutée jusqu'à ce caractère (différent). Comme les zéros de tête, les espaces de tête sont ignorés pour la création des valeurs. Si aucun des caractères mentionnés ci-dessus n'apparaît, la valeur "NUL" est renvoyée. Si un signe décimal est compris dans la chaîne de caractères, le résultat peut uniquement être affecté à une variable REAL ou à une variable REAL double précision. Dans ce cas, une attribution à une variable INTEGER génère le message d'erreur "ALLOCATION INVALIDE".

Tab. 7-275: Syntaxe VAL

Exemple :

```
1 I% = VAL("1.23DE")
2 K% = VAL("123DE")
3 J% = VAL("ABC")
4 R = VAL("-1.23DE")
5 Z = VAL("+ 000001234TEST4365")
6 X = VAL("ABC1.23DE")
7 D! = VAL("1234567.234567")
```

La ligne 1 déclenche un message d'erreur, car l'attribution doit se faire à une variable INTEGER.

La valeur de la variable INTEGER K% est 123.

Les chiffres 1, 2, 3 sont convertis en un nombre INTEGER.

Le caractère "D" interrompt la conversion, car il ne peut pas faire partie d'un nombre INTEGER. Les caractères suivants ne sont pas traités.

La valeur de la variable INTEGER "J%" est NUL ; aucune valeur n'est attribuée à la variable. Le caractère "A" interrompt le traitement de l'<expression STRING>.

Ordres CPL

La valeur de la variable REAL "R" est -1,23. Le caractère "-" est reconnu comme signe du nombre REAL. Le chiffre 1, le caractère ".", les chiffres 2 et 3 sont convertis en un nombre REAL. Le caractère "D" interrompt la conversion, car il ne peut pas faire partie d'un nombre REAL. Le caractère "E" ne sera pas traité.

La valeur de la variable REAL "Z" est 1234. Le caractère "+" est reconnu comme le signe du nombre REAL. Comme les zéros de tête, les espaces de queue sont ignorés pour la création des valeurs. Les chiffres 1, 2, 3 et 4 sont convertis en un nombre REAL. Le caractère "T" interrompt la conversion, car il ne peut pas faire partie d'un nombre REAL. Les caractères restants ne sont pas traités.

La variable REAL "X" est NUL, c.-à-d. qu'aucune valeur ne lui est attribuée. La reconnaissance du caractère "A" interrompt la conversion. La valeur de la variable REAL double précision "D!" est 1234567,234567.

7.6.3 Modification des chaînes de caractères

Affectation d'une expression STRING à un tableau

Description : Si l'expression STRING contient moins de caractères que la plage sélectionnée du tableau, la plage restante est considérée comme non attribuée. Cette plage restante n'est pas prise en compte dans la longueur du tableau.

Exemples :

```
1 DIM XYZ$(16)
2 XYZ$=" " Contenu du tableau XYZ$ : " "
Longueur du tableau XYZ$ : 1
```

Si la longueur de l'expression STRING dépasse la longueur maximale du tableau lors de l'attribution, alors les caractères en excédent sont rejetés.

```
1 DIM XYZ$(3)
2 XYZ$="ABCDEF"
Contenu du tableau XYZ$ : "ABC"
Longueur du tableau XYZ$ : 3 ---> Longueur maximale
```

```
1 DIM XYZ$(16)
2 A$="CECI"
3 B$="EST UN TEST"
4 C$="UN ŒUF"
5 MID$(XYZ$,1,4)=A$
```

CONTENU DU TABLEAU

Programme:

1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16
D	A	S													
D	A	S													

| Longueur = 4 | Zone non affectée |

```
6 MID$(XYZ$,5,6)=B$
```

CONTENU DU TABLEAU

Programme:

1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16
D	A	S		I	S	T		E	I						
Longueur = 10								Zone non affectée							

7 MID\$(XYZ\$,5,12)=B\$

CONTENU DU TABLEAU

Programme:

1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16
D	A	S		I	S	T		E	I	N		T	E	S	T
Longueur= 16 ; Zone est complètement affectée															

8 MID\$(XYZ\$,9,8)=C\$

CONTENU DU TABLEAU

Programme:

1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16
D	A	S		I	S	T		E	I	N		T	E	S	T
Longueur= 16 ; Zone est complètement affectée															

9 XYZ\$=MID\$(XYZ\$,1,4)

CONTENU DU TABLEAU

Programme:

1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16
D	A	S													
Longueur = 4				Zone non affectée											

Le tableau a été tronqué.

Enchaînement d'expressions STRING

Description : Le signe "+" permet d'enchaîner plusieurs expressions STRING. Le résultat doit être affecté à un tableau. La profondeur d'imbrication lors de l'enchaînement d'expressions STRING est de 3. Si elle est dépassée, le message d'erreur "ERREUR D'EXÉCUTION 2153 – IMBRICATION TROP PROFONDE" s'affiche.

Exemple : Enchaînement au sein d'ordres CPL

```
1 DIM A$(3)
2 DIM B$(3)
3 A$ = "ABC"
4 B$ = "DEF"
```

Ordres CPL

```

5 C$ = "GH"
6 D$ = "JKL"
7 OPENW(1,"P2",130,"TEST ENCHAÎNEMENT",10)
8 PRN#(1,A$+B$)
9 PRN#(1,A$+C$)
10 PRN#(1,C$+D$)
11 PRN#(1,A$+C$+"TEST")
12 PRN#(1,"UVW"+"XYZ")
13 CLOSE(1)

```

Contenu du fichier P2 :

ABCDEF	<LF>
ABCGH	<LF>
GHJKL	<LF>
ABCGHTEST	<LF>
UVWXYZ	<LF><ETX><LF>

Enchaînement de textes via les expressions STRING

```

10 DIM A$(100)
20 DIM B$(100)
30 DIM C$(10)
40 DIM D$(20)
51 DIM E$(30)
52 DIM F$(30)
53 DIM G$(30)
54 DIM H$(30)
55 DIM I$(30)
60 A$="CECI"
70 B$="EST UN TEST"
80 MID$(C$,1,6)=A$ + B$
90 MID$(D$,1,10)=MID$(A$,1,1) + MID$(B$,1,2)
92 E$=A$ + MID$(B$,1)
93 X$="ABC"
94 Y$="DE"
95 F$=X$ + Y$ 96 G$=X$ + A$
97 H$=X$ + A$ + "TEST"
98 I$="TES" + "T1"

```

Contenu de A\$:	"CECI "	Longueur 4
Contenu de B\$:	"EST UN TEST"	Longueur 12
Contenu de C\$:	"CECI EST"	Longueur 6
Contenu de D\$:	"CES"	Longueur 3

Ordres CPL

Contenu de E\$:	"CECI EST UN TEST"	Longueur 16
Contenu de F\$:	"ABCDE"	Longueur 5
Contenu de G\$:	"ABCDAS "	Longueur 7
Contenu de H\$:	"ABCDAS TEST"	Longueur 11
Contenu de I\$:	"TEST1"	Longueur 5

La programmation suivante entraîne des erreurs :

1 DIM A\$(3):A\$ = "ABC":B\$ = "CD":C\$ = "EF"	
2 D\$ = A\$ + B\$	Affectation non permise à une variable STRING non dimensionnée
3 D\$ = B\$ + C\$	
4 D\$ = A\$ + B\$ + "TEST"	
5 D\$ = "TEST" + "TEST1"	

STR\$

```
1 DIM A$(50) : DIM B$(21)
2 A$ = STR$("A$ = ##.###", (37/3)) : B$ = STR$(2.5)
```

Contenu du champ de caractères A\$: "A\$ = 12.333";

Tableau B\$: "2 500"

VAL

```
1 DIM FOLGE$(20) : FOLGE$="VALEUR X -0001.234 MM"
2 XR=VAL(MID$(FOLGE$,7)) : Z%=VAL(MID$(FOLGE$,7,6))
3 Y%=VAL(MID$(FOLGE$,15,5)) : X%=VAL(MID$(FOLGE$,18))
```

Contenu de la variable REAL XR : -1 234

Contenu de la variable INTEGER Z : -1
%

Contenu de la variable INTEGER Y : 34
%

Contenu de la variable INTEGER X : NUL
%

LEN

```
1 DIM Z$(10)
2 Z$ = "TEST"
3 S$ = "TEST"
4 A% = LEN("TEST")
5 B% = LEN(Z$)
6 C% = LEN(S$)
7 D% = LEN("TEST"+Z$+S$)
```

Ordres CPL

Contenu de la variable INTEGER A% : 4
 Contenu de la variable INTEGER B% : 4
 Contenu de la variable INTEGER C% : 4
 Contenu de la variable INTEGER D% : 12

Ordre MID\$ avec accès en lecture

```

10 DIM A$(4)
20 DIM B$(10)
30 DIM C$(10)
40 DIM D$(10)
50 DIM E$(10)
55 DIM F$(10)
60 A$ = "ABCD"

```

```

70 B$ = MID$(A$,2,2)          --> B$ = "BC"
80 C$ = MID$(A$,2,5)          --> C$ = "BCD"
95 E$ = MID$(A$,5,1)          --> E$ = NUL
97 F$ = MID$(A$,2)            --> F$ = "BCD"
98 F$ = MID$(F$,1,1)          --> F$ = "B"

```

Ordre MID\$ avec accès en écriture

```

10 DIM A$(4)
20 DIM B$(10)
30 DIM C$(10)
40 DIM D$(10)
60 A$ = "ABCD"
70 B$ = "1234567890"
80 C$ = "EFGHIJKLMNOP"
85 D$ = A$                    --> D$ = "ABCD"
90 MID$(D$,2,3) = B$          --> D$ = "A123"
95 MID$(D$,5,1) = C$          --> D$ = "A123E"
97 MID$(D$,4) = B$            --> D$ = "A121234567"

```

TRIM\$

```

1 DIM XYZ$(16)
2 XYZ$ = "XWERT = 0.123 "
3 A$ = MID$(XYZ$,8)
4 B$ = TRIM$(MID$(XYZ$,8))
5 C$ = TRIM$(MID$(XYZ$,8),"L")
6 D$ = TRIM$(MID$(XYZ$,8),"R")

```

Contenu de la variable STRING A\$:	" 0 123"
Contenu de la variable STRING B\$:	"0 123"
Contenu de la variable STRING C\$:	"0 123"
Contenu de la variable STRING D\$:	"0 123"

TRIM\$

CHAR	READ	WRITE	FUNCT	PROC	WAIT	ERRNO	!

Description **TRIM** – Suppression des espaces de tête ou de queue

Lors de l'affectation d'une zone du tableau de caractères, "TRIM\$()" fournit une chaîne de caractères sans espaces de tête (→ Index L) ou de queue (→ Index R) à une variable STRING ou à un tableau. L'ordre "TRIM" sans index masque aussi bien les espaces de tête que les espaces de queue. Si un enchaînement a lieu au sein de l'ordre TRIM (par ex. TRIM\$(A\$+B\$)), le résultat ne doit être attribué qu'à un seul tableau.

Syntaxe :

```
TRIM$ (<chaîne de caractères>)
TRIM$ (<chaîne de caractères>)> ,"L")
TRIM$ (<chaîne de caractères>)> ,"R")
```

Tab. 7-276: Syntaxe TRIM\$

Exemple :

```
1 A$ = " ABCDEF "
2 B$ = TRIM$ (A$,"L")
3 C$ = TRIM$ (A$,"R")
4 D$ = TRIM$ (A$)
5 PRN# (1,">",A$,"<")
6 PRN# (1,">",B$,"<")
7 PRN# (1,">",C$,"<")
8 PRN# (1,">",D$,"<")
```

mène aux lignes suivantes dans le fichier de numéro logique 1 ouvert pour écriture :

```
> ABCDEF <
>ABCDEF <
> ABCDEF<
>ABCDEF<
```

MID\$

CHAR	READ	WRITE	FUNCT	PROC	WAIT	ERRNO	!

Description : **MID** – Lecture des caractères d'une chaîne de caractères

Le résultat peut être transmis à une variable CHARACTER dimensionnée ou non dimensionnée :

- Une variable CHARACTER dimensionnée contient l'intégralité de la chaîne partielle déterminée par l'ordre MID.
- Une variable CHARACTER non dimensionnée contient uniquement l'adresse de départ et la longueur de la chaîne partielle déterminée. Si l'expression STRING à partir de laquelle la chaîne partielle a été extraite

Ordres CPL

change, la variable CHARACTER non dimensionnée est également modifiée en conséquence.

Si un enchaînement se produit au sein de l'ordre MID (par ex. MID\$ (A\$+B\$, 2, 3)), le résultat doit être attribué à un seul tableau.

Syntaxe :

MID\$ (<expression STRING>, <point de départ>[, <nombre de caractères>])	
<expression STR>	Expression STRING à partir de laquelle les parties doivent être extraites.
<point de départ>	Définit la position au sein du tableau <expression STRING> à partir de laquelle les caractères doivent être extraits.
<nombre de caractères>	Détermine le nombre de caractères qui sont extraits. Si le <nombre de caractères> n'est pas programmé, tous les caractères sont extraits jusqu'à la fin du tableau.

Tab. 7-277: Syntaxe MIDS\$

La plage de valeurs pour les 2e et 3e paramètres comprend les valeurs entières entre 1 et 1024. Si la plage de valeurs n'est pas respectée, le message d'erreur "PARAMÈTRE INADMISSIBLE" s'affiche.

En cas d'accès à une partie du tableau qui n'est pas encore attribuée, la valeur renvoyée est "NUL".

Exemple : Programme:

<pre>1 DIM A\$(10) 2 DIM B\$(5) 3 A\$="ABCDEFGHIJ" 4 B\$=MID\$(A\$,2,5) 5 C\$=MID\$(A\$,2,5) 6 REM Les variables B\$ ont C\$ le même contenu : BCDEF 7 A\$="QRSTUVWXYZ" 8 REM La variable B\$ a le contenu : BCDEF La variable C\$ a le contenu : RSTUV</pre>	

MCASE\$

CHAR	READ	WRITE	FUNCT	PROC	WAIT	ERRNO	!

Description :

Mixed CASE – Conversion des majuscules ou minuscules
L'ordre convertit une expression STRING en majuscules ou minuscules. Le résultat de la conversion est fourni par la valeur de retour de l'ordre.

Syntaxe :

MCASE\$ (<expression STRING>, <DataSelection>)	
<expression STRING>	Expression STRING qui doit être convertie.

<DataSelection>	Expression STRING "U" : conversion en majuscules "L" : conversion en minuscules
ERRNO	Variable CPL Si la variable CPL ERRNO est programmée à un emplacement quelconque au sein de la liste de paramètres, l'ordre ne génère aucune erreur d'exécution interne. L'erreur est renvoyée par une valeur de retour correspondante de la variable. Les valeurs de retour suivantes sont possibles : 0 : Accès OK -1 : Erreur de paramètre Si la variable ERRNO n'est pas indiquée, une erreur d'exécution est générée en cas d'erreur de paramètre.

Tab. 7-278: Syntaxe MCASE\$

Exemple :

```

1 DIM A$(20)
2 DIM B$(20)
3 A$="Bonjour le monde"
4 B$=MCASE$(A$,"U")
5 REM B$ a le contenu : "BONJOUR LE MONDE"
```

Particularités et restrictions : L'affectation de la valeur de retour à une variable CHARACTER dimensionnée entraîne une erreur d'exécution.

7.6.4 Exemples de programmation

Description : Il est possible d'attribuer une expression STRING à une variable STRING.

Exemples : **Programmation des variables STRING (sans dimensionnement préalable)**

```

1 A$="ABCDE"
2 B$=CHR$(10)
```

Les variables STRING (sans dimensionnement préalable) se comportent comme des renvois ou des pointeurs :

Programme:

```

10 DIM B$(10)
20 A$ = "Bleu"
30 B$ = "Jaune"
40 PRN#(0,A$,B$) : édition REM : BleuJaune
50 A$ = B$ : REM A$ renvoie à B$
60 B$ = "XXXX"
70 PRN#(0,A$,B$) : édition REM : XXXXXXXX
```

En cas d'accès en lecture, il est possible d'accéder à des parties de la variable STRING au moyen de l'ordre MID\$:

```

1 A$="ABCDEFGH IJKLMN"
2 B$=MID$(A$,2,1)
3 C$=MID$(A$,4,4)
```

Les programmations suivantes entraînent des erreurs :

```

4 MID$(A$,1,4)="ABCD"
4 A$=MID$(A$,1,3) + MID$(A$,4,1)
```

Ordres CPL

```
4 A$=B$ + A$
```

Pour poursuivre le traitement d'un tableau dimensionné, il est nécessaire d'accéder de manière ciblée à un ou plusieurs caractères qui se suivent. Il s'agit là du seul moyen possible d'affecter un tableau ou une partie du tableau à une variable STRING ou à un autre tableau.

L'accès en lecture et en écriture à une partie d'un tableau s'effectue via l'ordre MID\$. Si seul le nom du tableau est indiqué, l'ensemble du tableau est adressé.

Lecture d'un tableau

Pour accéder au <n>-ième caractère du tableau, procéder comme suit (n est inférieur ou égal à la longueur du tableau et au nombre de caractères du tableau) :

```
1 DIM VWX$(13)
2 VWX$="TEST TEST TES"
3 A$ = MID$(VWX$,12,1)
4 I%=12
5 A$=MID$(VWX$,I%,1)
```

Le 12e caractère ("E") du tableau VWX\$ est affecté à la variable STRING A\$.

Écriture d'un tableau

Si le contenu d'une variable STRING doit être transmis au tableau ou à une partie du tableau, l'affectation doit être modifiée.

Exemple : Écriture partielle d'un tableau

```
1 DIM XYZ$(15)
2 B$="ABCDE"
3 MID$(XYZ$,1,5)=B$
4 MID$(XYZ$,6,5)=B$
```

Le contenu de la variable STRING B\$ est affecté aux caractères 1 à 10 du tableau XYZ\$.

La programmation suivante entraînerait le message d'erreur "CHAMP DE CARACTÈRES NON OCCUPÉ", car les caractères 1 à 5 du tableau ne sont pas encore affectés :

```
1 DIM XYZ$(15)
2 B$="ABCDE"
4 MID$(XYZ$,6,5)=B$
```

Exemple : Écriture partielle d'un tableau

```
1 DIM XYZ$(100)
2 B$= "ABCDE"
3 MID$(XYZ$,1,10)=B$
```

Contenu de la variable "ABCDE"
 STRING B\$:

Contenu de la variable de tableau XYZ\$:

La variable de tableau a la longueur 5. Les 95 caractères restants ne sont pas attribués.

Si la longueur de la variable STRING est inférieure au tableau, alors le tableau XYZ\$ n'est écrit que sur la longueur de la variable STRING. En cas d'affectation de ce tableau à une variable STRING, ce n'est pas l'ensemble du tableau déterminé par l'instruction DIM qui est affecté, mais uniquement la plage écrite auparavant (→ Longueur du tableau).

Exemple :

```
1 DIM XYZ$(100)
3 MID$(XYZ$,1,10)="ABCDE"
4 MID$(XYZ$,6,3)="T"
```

Après le bloc 3, le contenu de la variable de tableau XYZ\$ est : "ABCDE". La variable de tableau a la longueur 5. Les 95 caractères restants ne sont pas attribués et ne sont donc pas pris en compte dans la longueur.

Après le bloc 4, le contenu de la variable de tableau XYZ\$ est : "ABCDET". La variable de tableau a maintenant la longueur 6. Les 94 caractères restants ne sont pas attribués et ne sont donc pas pris en compte dans la longueur.

Exemple : Écrasement d'un tableau

```
1 DIM XYZ$(100)
3 MID$(XYZ$,1,10)="1234567890"
4 MID$(XYZ$,3,3)="T"
```

Après le bloc 3, le contenu de la variable de tableau XYZ\$ est "1234567890". La variable de tableau a la longueur 10.

Après le bloc 4, le contenu de la variable de tableau XYZ\$ est "12T4567890".

La variable de tableau a la longueur 10. Le caractère "3" est écrasé par "T". Les caractères "4" et "5" sont conservés.

Exemple : Accès interdit au tableau

```
1 DIM XYZ$(100)
3 MID$(XYZ$,1,6)="ABCDEF"
5 MID$(XYZ$,9,5)="TESTS"
```

Après le bloc 3, le contenu de la variable de tableau XYZ\$ est "ABCDEF".

La variable de tableau a la longueur 6.

Une tentative d'affectation d'une constante aux composants 9 à 13 du tableau se produit après le bloc 5. Cette tentative conduit au message d'erreur "CHAMP DE CARACTÈRES NON OCCUPÉ", puisque les 7e et 8e composants ne sont pas attribués.

Pour accéder au tableau complet, il suffit d'indiquer le nom de variable.

7.7 Communication avec un client

7.7.1 MMC

INT	READ	WRITE	FUNCT	PROC	WAIT	ERRNO	!

Ordres CPL

Description : **Man Machine Communication** – Communication homme-machine

Envoie les informations concernant la durée d'un programme pièce à un client en cours d'exécution et attend le résultat de ce client. Cela s'effectue au moyen des variables CPL qui peuvent aussi bien envoyer les valeurs du programme pièce que renvoyer des valeurs au programme pièce.

Le programme pièce est arrêté lors de l'exécution du programme à l'endroit où l'ordre MMC est atteint.

Il existe les possibilités de traitement suivantes :

- Si aucun client capable de traiter les données de l'ordre MMC n'est connecté, la valeur de retour correspondante (= 1) est définie et le traitement du programme pièce se poursuit.
- S'il existe un client capable de traiter les données de l'ordre MMC, une affectation a lieu entre le programme pièce et le client. Une fois que le client a envoyé une réponse, la valeur de retour correspondante est définie et le traitement du programme pièce se poursuit.

L'ordre MMC peut avoir un maximum de 20 variables CPL comme paramètres. Le nom et les valeurs de ces variables sont transmis au client.

Syntaxe :

MMC (<VarCPL1>[, <VarCPL2>...[, <VarCPLn>]...])	
<VarCPL1>...	Variables CPL, n = max. 20
<VarCPLn>	Le client peut écrire de nouvelles valeurs dans les variables CPL indiquées dans l'ordre MMC. Ces variables CPL indiquées dans l'ordre MMC peuvent être utilisées dans le programme pièce.

Tab. 7-279: Syntaxe MMC

L'ordre MMC fournit les valeurs de retour suivantes :

0 :	OK
1 :	Aucun client disponible
2 :	Erreur sur le client
9 :	Client terminé

Exemple :

```

10  DIM PROGNAMES$ (50)
20  PROGNAMES$="WinProg"
30  INTPAR%=1
40  REALPAR=1.1
50  I%=MMC (PROGNAMES$, INTPAR%, REALPAR)

```

; Les variables CPL PROGNAMES\$, INTPAR% et REALPAR et leurs valeurs sont mises à la disposition du client.

; La préparation des blocs du programme pièce ne se poursuit en ligne 60 que si un message PRÊT correspondant est reçu.

```

60  IF I%=0 THEN
70  IF INTPAR%=2 THEN
80  ...
90  ELSE
100 ...
110 ENDIF
120 ENDIF

```


8 Annexe

8.1 Tableau synoptique des fonctions CN

8.1.1 Classement par ordre alphanumérique selon le format long

Nom de la fonction CN		Groupe	Désignation/Explication
Format long	Format abrégé	(« – » signifie : effet au sein du bloc)	
()	–	–	Commentaire : négliger le contenu des parenthèses
(MSG	–	–	Afficher le texte du message
/	–	–	Sauter séquence
//	–	–	Commentaire
;	–	–	Commentaire : négliger le reste de la ligne
ABS	–	CPL	ABS(<valeur entrée>)
AC(...)	–	–	Programmation absolue locale, par ex. X=AC(10)
ACOS	–	CPL	<valeur de fonction> = ACOS(<valeur entrée>) : Application de la fonction arc cosinus à la <valeur entrée>
ACS	–	CPL	ACS(<axe> [,<mode de sélection>[,<canal>]]) : Fournit la position de consigne actuelle d'un axe.
ActRadFact	ARF		ActRadFact(<nom de l'axe> <facteur> {,<nom de l'axe> <facteur>}...) Programmation des facteurs du rayon d'action
Adjust	ADJ	–	ADJ(<nom FCN>) : Comparaison des groupes modaux de fonctions CN
AND	–	CPL	<expression1> AND <expression2> : Liaison binaire de deux expressions booléennes ou INTEGER
APOS	–	CPL	APOS(<sélection d'axe>) : Transmet la valeur d'axe réelle actuelle par rapport à l'origine de la machine.
Area	ARA	–	Area(<NoBr>,<statut>{,<mode>{,<P1>},{<P2>},{<D1>},{<D2>}}) : Définit, active ou désactive jusqu'à 10 zones de travail ou zones mortes rectangulaires, bidimensionnelles avec des limites parallèles aux axes.
ASC	–	CPL	ASC(<chaîne de caractères>)
ASIN	–	CPL	<valeur de fonction> = ASIN(<valeur entrée>) : Application de la fonction arc sinus à la <valeur entrée>
ASPCLR	–	–	ASPCLR(<NoSP>) Sous-programmes asynchrones : désinscription
ASPDIS	–	–	ASPDIS(<NoSP>) Sous-programmes asynchrones : mise hors service
ASPENA	–	–	ASPENA(<NoSP>) Sous-programmes asynchrones : mise en service

Annexe

Nom de la fonction CN		Groupe (« – » signifie : effet au sein du bloc)	Désignation/Explication
Format long	Format abrégé		
ASPRTP	–	–	ASPRTP(<NoSP>,<point>) Sous-programmes asynchrones : définition du point de redémarrage
ASPSET	–	–	ASPSET(<NoSP>,<NomSP>{,<drapeaux>}) Sous-programmes asynchrones : inscription
ASPSTA	–	–	ASPSTA(<NoSP>{,<NoCanal>}) Sous-programmes asynchrones : déclenchement
AssLogName	ALN	–	ALN(...) : Affectation des noms d'axes logiques
ASTOPA	–	–	ASTOPA[<NoCanal>, <cond.1> {,<cond.2>}{...{,<cond.8>}}...] Instruction de synchronisation de canal : le canal à commander est arrêté tant que toutes les conditions sont remplies. Les conditions se rapportent aux positions d'axe, par ex. « Z » > 20.
ASTOPO	–	–	ASTOPO[<NoCanal>, <cond.1> {,<cond.2>}{...{,<cond.8>}}...] Instruction de synchronisation de canal : le canal à commander est arrêté tant qu' une condition est remplie. Les conditions se rapportent aux positions d'axe, par ex. « Z » > 20.
ATAN	–	CPL	<valeur de fonction> = ATAN(<valeur entrée>) : Application de la fonction arc tangente à la <valeur entrée>
ATAN2	–	CPL	<valeur de fonction> = TAN2(valeur_entrée1 , valeur_entrée2) Cette fonction permet de déterminer l'angle dans le bon quadrant.
ATBWD	–	–	AtBwd(<CoordAxe>,<coord>) Calibrage des cinématiques d'axe : transformation en arrière
ATCAL	–	–	ATCAL(<fichier>,<DonnéesOpt>,<masque> {,<info>}{,<nblt>}) Calibrage des cinématiques d'axe : optimisation des paramètres
ATFWD	–	–	ATFWD(<coord>,<CoordAxe>{,<DonnéesPar>}) Calibrage des cinématiques d'axe : conversion des paramètres
ATGET	–	–	ATGET(<DonnéesPar>{,<NoTransfoAxe>}) Calibrage des cinématiques d'axe : lecture des paramètres de la commande numérique
ATPUT	–	–	ATPUT(<DonnéesPar>{,<NoTransfoAxe>}) Calibrage des cinématiques d'axe : écriture de paramètres dans la commande numérique
ATrans	ATR	Décalage des coordonnées du programme	ATR(X..., Y..., Z) : Décalage additif des coordonnées du programme ACTIVÉ ATR () : Décalage additif des coordonnées du programme DÉSACTIVÉ
AUXFUNC	–	–	Exécution des fonctions auxiliaires actives de tous les groupes

Nom de la fonction CN		Groupe (« – » signifie : effet au sein du bloc)	Désignation/Explication
Format long	Format abrégé		
AxAcc	AAC	Accélération de l'axe	AAC(X..., Y..., Z) : Activer AAC(1) : Activer avec le réglage enregistré AAC() : Désactiver Programmer l'accélération de l'axe
AxAccSave	AAS	–	Enregistrement de l'accélération actuelle de l'axe
AxCompTab	–	–	AxCompTab(<CAX>, <table>) : Remplacement des tables de correction des erreurs d'axe (LSEC et CCOMP)
AxCouple	AXC	Couplage d'axe	AXC(<maître>, <esclave1>(...), <esclave2>(...), ...) : Activer le couplage d'axe AXC() : Désactiver le couplage d'axe
AXADR\$	–	CPL	AXADR\$(<index> [, [<mode de sélection>] [, <numéro du canal>]]) Adresses d'axes et de coordonnées
AXINF	–	CPL	La valeur de retour fournie est l'index d'axe du système pour un axe du canal ou du système.
AxisReplace	AXR		AXR(<KAN1>, <NPA2>) Remplacement d'axes
AxisToSpindle	ATS	–	ATS(<nom de l'axe>) : Commutation de l'axe au mode broche
AxisToSpindle- Wait	ATSW	–	AxisToSpindleWait(<NPA> <IPA>{, <NPA> <IPA>}...) Désactiver le mode d'axe C pour les broches et attendre
AxJerk	AJK	–	AxJerk(<valeurs>) Modification de la secousse maximale des axes AxJerk() ou AxJerk(0) Réactiver la secousse d'axe issue des paramètres machine
AXO	–	CPL	AXO((<sélection d'axe> [, <mode de sélection>]) : Transmet un décalage SetPos actif pour une coordonnée.
AXP	–	CPL	AXP(<numéro d'axe>, <information sur la course>) : Cette fonction permet d'écrire les programmes pièce et les programmes de mesure indépendamment du plan (programmer AXP au lieu des valeurs d'adresses).
AxVel	AVE	–	AxVel(<valeurs>) : Modification de la vitesse maximale de l'axe
AxVelSave	AVS	–	Enregistrement temporaire de la vitesse maximale de l'axe
BCD	–	CPL	<valeur BCD>=BCD(<valeur binaire>) : Convertir le format BCD au format binaire

Annexe

Nom de la fonction CN		Groupe (« – » signifie : effet au sein du bloc)	Désignation/Explication
Format long	Format abrégé		
BcsCorr	BCR	Correction de la position de la pièce à usiner	BCR({<décalage XW>},{<décalage YW>}{,<décalage ZW>},{<angle1>}{,<angle2>}{,<angle3>}}) : Activer BCR() : Désactiver Placement : correction de la position de la pièce à usiner
BIN	–	CPL	<valeur binaire> = BIN(<valeur BCD>) : Convertir le format binaire au format BCD
BITIF	–	CPL	BITIF(<SignalBit>[,<index>[,<unité IF>]]) : Accès à l'interface numérique entre la commande numérique et l'API
BlkNmb	BNB	–	BlkNmb(<nbre>) : Limite le nombre de blocs-programmes lus et pris en compte par la préparation des blocs.
BSTOPA	–	–	BSTOPA[<NoCanal>, <cond.1> {,<cond.2>}{...{,<cond.8>}}...] Instruction de synchronisation de canal : le canal à commander est arrêté tant que toutes les conditions sont remplies. Les conditions se rapportent aux positions de base de la pièce à usiner, par ex. "Z" > 20.
BSTOPO	–	–	BSTOPO[<NoCanal>, <cond.1> {,<cond.2>}{...{,<cond.8>}}...] Instruction de synchronisation de canal : le canal à commander est arrêté tant qu'une condition est remplie. Les conditions se rapportent aux positions de base de la pièce à usiner, par ex. "Z" > 20.
CALL	–	CPL	CALL <numéro du programme> [<paramètre de transfert1>,...] [DIN] : Appel de sous-programme à partir un programme CPL
CASE	–	CPL	CASE <expression INTEGER> OF LABEL <constante INT> [,<autre constante INT>] [: <instruction>]<instruction> LABEL ... OTHERWISE <instruction><instruction> ENDCASE Sélection conditionnelle parmi plusieurs alternatives
ChkNxtBlk	CNB	–	ChkNxtBlk (1) Activer la surveillance du bloc suivant ChkNxtBlk (0) Désactiver la surveillance du bloc suivant
ChLength	CHL	Chanfrein/Arrond.	CHL(<longueur de chanfrein>) : Activer CHL() : Désactiver Insérer des chanfreins de raccord avec la longueur de chanfrein définie

Nom de la fonction CN		Groupe (« – » signifie : effet au sein du bloc)	Désignation/Explication
Format long	Format abrégé		
CHR\$	–	CPL	CHR\$(<expression INTEGER>) : Fournit un caractère dont le nombre ordinal dans le tableau ASCII est égal à la valeur transmise par le paramètre <expression INTEGER>.
ChSection	CHS	Chanfrein/Arrond.	CHS(<segment de chanfrein>) : Activer CHS() : Désactiver Insérer des chanfreins de raccord avec le segment de chanfrein défini
CLOCK	–	CPL	<valeur de fonction> = CLOCK : Interrogation du compteur de temps, en millisecondes
CLOSE	–	CPL	CLOSE(<n>) : Ferme un fichier ouvert à la fin des opérations de lecture ou d'écriture.
CLRWARN	–	CPL	Supprimer tous les messages d'avertissement d'un canal générés avec SETWARN
COF	–	CPL	COF(<sélection d'axe>[,<mode de sélection>]) : Fournit le dernier décalage de contour programmé (Shift) d'une coordonnée pour le canal actuel.
COFFS	–	–	Décalage de contour pour correction de rayon 3D
Collision	CLN	Surveillance de collision	CLN(1) : Activer CLN(CollErr.. LA... DLA... DEF) : Activer avec paramétrage CLN() : Désactiver Surveillance de collision pour la correction 2D de trajectoire de la fraise
ConfG0	CG0	Interpolation	ConfG0(ASY<statut>) ConfG0(ASY1) Activer l'interpolation G0 asynchrone ConfG0(ASY0) Désactiver l'interpolation G0 asynchrone
ConstFeed	CFD	Profil en V	Vitesse constante
CornerLineDetection	CLD	–	Détection automatique des angles et des droites
Coord	CRD	–	CRD(x) : Activer la transformation d'axe CRD() : Désactiver la transformation d'axe, par ex. la transformation de surface frontale
COS	–	CPL	<valeur de fonction> = COS(<valeur entrée>) : Fonction cosinus pour la <valeur entrée>
CoupleSplineTab	CST	–	CST(STAB(...)) Création d'une table de couplage spline

Annexe

Nom de la fonction CN		Groupe (« – » signifie : effet au sein du bloc)	Désignation/Explication
Format long	Format abrégé		
CurveParamInterpolation	CPI	Interpolation	CurveParamInterpolation(1) Interpolation des paramètres des courbes activée CurveParamInterpolation(0) Repasser à l'interpolation des longueurs de la trajectoire
D	D	Correction D	Dx : Sélection de la correction D (1 à 99 outils) D0 : Désélection de la correction D
DATE	–	CPL	<variable STRING> = DATE : Affecte la date à une <variable STRING>, sous la forme JJ.MM.
DBLOAD	–	CPL	DBLOAD(<TableBD>,<clé1>,<clé2>), <nom de fichier>[,<mode>]) : Peut importer tout ou partie d'une table de base de données à partir d'un fichier de la base de données.
DBMOVE	–	CPL	DBMOVE(<TableBD>,<CléSrc1>,<CléSrc2>, <CléDest1>,<CléDest2>[,<mode>]) : Déplacement d'un jeu de données au sein d'une table de base de données
DBSAVE	–	CPL	DBSAVE(<TableBD>,<clé1>,<clé2>), <nom de fichier>[,<mode>]) : DBSAVE sert à sauvegarder tout ou partie d'une table de base de données dans un fichier.
DBSEA	–	CPL	DBSEA (<TableBD>,<clé1>,<clé2>, <CondRech>,<RésultRech>[,<VarRech>]) : Recherche les jeux de données dans une table de base de données.
DBSEAX	–	CPL	DBSEAX(<TableBD>,<clé1>,<clé2>, <CondRech>,<TableauRésult>,<TailleRésult>) : Recherche les jeux de données correspondant au critère de recherche dans une table de base de données d'outils.
DBTAB	–	CPL	DBTAB(<TableBD>,<clé1>,<clé2> [,<VarRésult>]) Importe un jeu de données complet ou une structure partielle d'une table de base de données dans une variable CPL ou le/la réexporte de la variable dans le jeu de données.
DBTABX	–	CPL	DBTABX(<TableBD>,<clé1>,<clé2> [,<mode>]) : Peut importer un jeu de données complet ou une structure partielle d'une table de base de données d'outils dans une variable CPL ou le/la réexporter de la variable dans le jeu de données.

Annexe

Nom de la fonction CN		Groupe (« – » signifie : effet au sein du bloc)	Désignation/Explication
Format long	Format abrégé		
DBTABXL	–	CPL	DBTABXL(<TableBD>,<clé1>,<clé2>[,<mode>]) : Peut importer un jeu de données complet ou une structure partielle d'une table de base de données d'outils dans une variable CPL ou le/la réexporter de la variable dans le jeu de données.
DCOFFS	–	–	Taux de variation du décalage de contour pour la correction de rayon 3D
DCT	–	CPL	DCT(<sélection de valeurs>, <jeu de données> [, [<table>] [, <unité>]]) : Accès en lecture et en écriture à une table de correction D quelconque ou aux valeurs de correction d'outil externes
DcTsel	DCS	–	DCS(({<chemin>}<nom de fichier>) : Sélection d'une table de correction D
DefAxis	DAX	–	Créer la configuration d'axe par défaut
DefSpindle	DSP	–	Restaurer les paramètres machine pour les broches du canal
DefTangTrans	DTT	Angle de raccordement	DTT(<angle de raccordement>) : Activer DTT() : Désactiver Définition du raccordement tangentiel entre segments de contour
DiaProg	DIA	Diamètre/Rayon	Programmation du diamètre
DIM	–	CPL	DIM <nom de variable> (<taille du tableau1> [,<taille du tableau2>]) : Détermination de la taille du tableau (dimensionnement) des variables ARRAY avec des constantes INTEGER
DIRCR	–	CPL	DIRCR(<répertoire>) : Crée un nouveau répertoire.
DIRDEL	–	CPL	DIRDEL(<répertoire>) : Supprime un répertoire vide.
DIRINF	–	CPL	DIRINF(4,[<index2>],<VarRésultat> [,<numéro de fichier>]) : Appelle les données administratives du système de fichiers.
DistCtrl	DCR	–	DCR(1) : Réglage en hauteur pour la numérisation ACTIVÉ, selon le paramètre machine DCR(<fct>) : ACTIVÉ avec ses propres données de configuration DCR() : Réglage en hauteur DÉSACTIVÉ
DPC	–	CPL	DPC(<sélection>[,<mode de sélection>]) : Fournit les derniers paramètres programmés de la correction de position de la pièce à usiner (BcsCorr) d'une coordonnée (valeurs de décalage et angle de rotation) pour le canal actuel.

Annexe

Nom de la fonction CN		Groupe (« – » signifie : effet au sein du bloc)	Désignation/Explication
Format long	Format abrégé		
ED	ED	Correction ED	ED x : Sélection de la correction d'outil externe (1 à 16 outils/arêtes coupantes) ED0 : Désélection de la correction d'outil externe
EndPosCouple	EPC	–	EPC(<coord M>,<coord S>,<facteur>) : Couplage de position finale ACTIVÉ EPC() : Couplage de position finale DÉSACTIVÉ
EOF	–	CPL	EOF(<n>) : Contrôle de la fin du fichier
ERASE	–	CPL	ERASE(<nom de fichier>) : Supprime les fichiers.
ERRNO	–	CPL	Message d'erreur de diverses fonctions CPL
EXP	–		EXP(<exposant> [, <base>]) Calcule la fonction exponentielle d'une base programmée quelconque ou d'une base par défaut e (e=2.718281828...), si aucune <base> (paramètre optionnel) n'est indiquée.
F	F	–	Adresse F pour G93, G94, G95
Adresse F2	F2		F2=<valeur> La syntaxe s'appelle "F2=". Il ne doit pas y avoir d'espace entre "F2" et "=".
FA	FA	–	Vitesse pour les axes asynchrones
FALSE	–	CPL	<variable booléenne> = FALSE : Valeur de vérité d'une variable BOOLE
FeedAd	FAD	Ajustement de l'avance	FAD(1) : Retirer l'axe DE la génération de l'avance FAD() : Axe contenu dans la génération de l'avance
FeedForward	FFW	Anticipation	FFW(X..., Y..., Z) : Activer FFW() : Désactiver Activation de l'anticipation
FeedGrp	FDG	Génération de l'avance	FeedGrp(1) ou FeedGrp Génération de l'avance : groupe d'avance
FeedGrpType	FDGT	Génération de l'avance	FeedGrpType(SMOOTH) ou FeedGrpType(1) Génération de l'avance : type de groupe d'avance
FILEACCESS	–	CPL	FILEACCESS(<nom de fichier>) : Détermine si un fichier existe et quels droits d'accès il possède.
FILECOPY	–	CPL	Copier les fichiers
FILEDATE	–	CPL	FILEDATE(<nom de fichier>,<mode>)] : Détermine la date/l'heure d'un fichier.
FILENO	–	CPL	Fournit le numéro de fichier logique valide suivant.

Annexe

Nom de la fonction CN		Groupe	Désignation/Explication
Format long	Format abrégé	(« – » signifie : effet au sein du bloc)	
FILEPOS	–	CPL	FILEPOS(<n>[,<mode>]) : Fournit le numéro d'enregistrement de l'enregistrement actuel et le décalage d'enregistrement d'un fichier aléatoire. Pour les fichiers séquentiels, fournit la position d'octet actuelle du pointeur du fichier.
FILESIZE	–	CPL	FILESIZE(<n>[,<k>]) : Indique la taille d'un fichier ou la limite jusqu'à laquelle un fichier a déjà été écrit.
FlyMeas	FME	Mesure au vol	FME(MpiAxis< i>) X... Y... Z... : Déplacement avec la mesure au vol
FOR NEXT	–	CPL	FOR <variable de comptage>=<valeur de départ> [STEP<longueur de pas>] TO <valeur d'arrivée><routine> NEXT [<variable de comptage>] : Construction en boucle avec compteur tournant automatiquement en parallèle
FsMove	FSM	–	Déplacement sur butée fixe (Fs = fixed stop)
FsProbe	FSP	–	Mesure sur butée fixe
FsReset	FSR	–	Supprimer la butée fixe
FsTorque	FST	–	Couple pour la butée fixe
G	–	–	Sous-programmes locaux définis par l'utilisateur
G0	G0	Interpolation	Interpolation linéaire (avance rapide)
G0(..)	G0(..)	Interpolation	Avance rapide avec options Paramètres optionnels : – NIPS : sans arrêt précis – IPS1 : fenêtre d'arrêt précis 1 – IPS2 : fenêtre d'arrêt précis 2 – IPS3 : Fenêtre d'arrêt précis 3 (freinage à v = 0 uniquement) – POL/POLAR : avec programmation des coordonnées polaires, par ex. G0(POL) X50 A45 B10, avec A, B = angle polaire 1/2
G1	G1	Interpolation	Interpolation linéaire (avance)
G2	G2	Interpolation	Interpolation circulaire antitrigonométrique (y compris hélicoïdal N)
G2(POL)	G2(..)	Interpolation	Interpolation circulaire antitrigonométrique (y compris hélicoïdal N) avec programmation des coordonnées polaires
G3	G3	Interpolation	Interpolation circulaire trigonométrique (y compris hélicoïdal N)
G3(POL)	G3(..)	Interpolation	Interpolation circulaire trigonométrique (y compris hélicoïdal N) avec programmation des coordonnées polaires

Annexe

Nom de la fonction CN		Groupe	Désignation/Explication
Format long	Format abrégé	(« – » signifie : effet au sein du bloc)	
G4	G4	–	G4(F..) : Temporisation, en secondes G4(S...) : Temporisation, en rotations de la broche, la broche de référence étant la broche principale (MainSp)
G5	G5	Interpolation	Entrée tangentielle dans le cercle (y compris hélicoïdal N)
G6	G6	Interpolation	Splines, nurbs
G8	G8	Pente de la trajectoire	Pente de la trajectoire ACTIVÉE avec forme de la trajectoire (SHAPE) programmée selon les paramètres machine
G8(SHAPE...)	–	Pente de la trajectoire	Pente de la trajectoire ACTIVÉE avec une forme de trajectoire (SHAPE) programmée
G9	G9	Pente de la trajectoire	Pente de la trajectoire DÉSACTIVÉE (sans SHAPE)
G9(..)	G9(..)	Pente de la trajectoire	G9(X..., Y) : Pente de la trajectoire DÉSACTIVÉE avec un ordre SHAPE programmé par axe G9(ASHAPE) : Pente de la trajectoire DÉSACTIVÉE avec un ordre SHAPE programmé par axe selon les paramètres machine G9(SHAPE ...) : Pente de la trajectoire DÉSACTIVÉE avec une forme (SHAPE) de trajectoire programmée G9(SIN ...) : Pente de la trajectoire DÉSACTIVÉE avec une forme sinusoïdale ² (ordres SHAPE possibles : 5, 10, 15, 20, 40)
G140	–	Correction de rayon 3D	Correction de rayon 3D DÉSACTIVÉE
G141	–	Correction de rayon 3D	Correction de rayon 3D à gauche du contour
G142	–	Correction de rayon 3D	Correction de rayon 3D à droite du contour
G152.1	G152	PI banque 1	Placement : plan incliné, programmable, banque 1
G152.2 à G152.5	–	PI banques 2-5	Placement : plan incliné, programmable, banques 2 à 5
G153	–	PI	Placement : plan incliné, complètement DÉSACTIVÉ
G153.1 à G153.5	–	PI banques 1-5	Placement : plan incliné, banques 1 à 5 DÉSACTIVÉ
G154.1	G154	PI banque 1	Placement : 1er plan incliné, banque 1 ACTIVÉ
G154.2 à G154.5	–	PI banques 2-5	Placement : 1er plan incliné, banques 2 à 5 ACTIVÉ
G155.1	G155	PI banque 1	Placement : 2e plan incliné, banque 1 ACTIVÉ
G155.2 à G155.5	–	PI banques 2-5	Placement : 2e plan incliné, banques 2 à 5 ACTIVÉ
G156.1	G156	PI banque 1	Placement : 3e plan incliné, banque 1 ACTIVÉ
G156.2 à G156.5	–	PI banques 2-5	Placement : 3e plan incliné, banques 2 à 5 ACTIVÉ

Nom de la fonction CN		Groupe	Désignation/Explication
Format long	Format abrégé	(« – » signifie : effet au sein du bloc)	
G157.1	G157	PI banque 1	Placement : 4e plan incliné, banque 1 ACTIVÉ
G157.2 à G157.5	–	PI banques 2-5	Placement : 4e plan incliné, banques 2 à 5 ACTIVÉ
G158.1	G158	PI banque 1	Placement : 5e plan incliné, banque 1 ACTIVÉ
G158.2 à G158.5	–	PI banques 2-5	Placement : 5e plan incliné, banques 2 à 5 ACTIVÉ
G159.1	G159	PI banque 1	Placement : 6e plan incliné, banque 1 ACTIVÉ
G159.2 à G159.5	–	PI banques 2-5	Placement : 5e plan incliné, banques 2 à 5 ACTIVÉ
G16	–	Sélection du plan	Aucun plan
G17	–	Sélection du plan	Sélection du plan XY
G17(...),G18(...); G19(...)	–	Sélection du plan	Changement de plan avancé G17/18/19(<axe1>,<axe2>,<axe3>) : Les axes entre parenthèses sous-tendent le WCS et reçoivent la signification X, Y et Z. Ensuite, le plan programmé est sélectionné. G17/18/19() : Restaurer le réglage par défaut du système de coordonnées de la pièce à usiner, puis sélectionner le plan programmé
G18	–	Sélection du plan	Sélection du plan ZX
G19	–	Sélection du plan	Sélection du plan YZ
G20	–	Sélection du plan	Libre sélection de plan (indépendamment du WCS) Les placements agissent sur le WCS ; le plan est défini indépendamment du WCS.
G33	–	Filet	Filetage
G40	–	Correction de trajectoire	Correction de trajectoire de la fraise DÉACTIVÉE
G41	–	Correction de trajectoire	Correction de trajectoire de la fraise à gauche de la pièce à usiner
G42	–	Correction de trajectoire	Correction de trajectoire de la fraise à droite de la pièce à usiner
G43	–	Stratégie d'insertion	Stratégie d'insertion – Arc de cercle
G44	–	Stratégie d'insertion	Stratégie d'insertion – Point d'intersection
G45	–	Vitesse de la fraise	Vitesse – Point de pénétration de la fraise
G46	–	Vitesse de la fraise	Vitesse – Centre de la fraise
G47	–	Correction d'outil	Activer la correction de la longueur d'outil
G47(..)	–	Correction d'outil	G47(<coord.L1>,<coord.L2>,<coord.L3>)G47(ActPlane) : Correction de la longueur d'outil ACTIVÉE, avec commutation de l'affectation de correction G47() : Correction de la longueur d'outil ACTIVÉE, avec commutation de l'affectation de correction selon les paramètres machine
G48	–	Correction d'outil	Correction de la longueur d'outil DÉACTIVÉE

Annexe

Nom de la fonction CN		Groupe (« – » signifie : effet au sein du bloc)	Désignation/Explication
Format long	Format abrégé		
G53	–	–	Tous les décalages d'origine DÉSACTIVÉS
G53.1 à G53.5	–	DO banques 1-5	Décalages d'origine, banques 1 à 5 DÉSACTIVÉS
G54.1	G54	DO banque 1	1er décalage d'origine, banque 1 ACTIVÉ
G54.2 à G54.5	–	DO banques 2-5	1er décalage d'origine, banques 2 à 5 ACTIVÉ
G55.1	G55	DO banque 1	2e décalage d'origine, banque 1 ACTIVÉ
G55.2 à G55.5	–	DO banques 2-5	2e décalage d'origine, banques 2 à 5 ACTIVÉ
G56.1	G56	DO banque 1	3e décalage d'origine, banque 1 ACTIVÉ
G56.2 à G56.5	–	DO banques 2-5	3e décalage d'origine, banques 2 à 5 ACTIVÉ
G57.1	G57	DO banque 1	4e décalage d'origine, banque 1 ACTIVÉ
G57.2 à G57.5	–	DO banques 2-5	4e décalage d'origine, banques 2 à 5 ACTIVÉ
G58.1	G58	DO banque 1	5e décalage d'origine, banque 1 ACTIVÉ
G58.2 à G58.5	–	DO banques 2-5	5e décalage d'origine, banques 2 à 5 ACTIVÉ
G59.1	G59	DO banque 1	6e décalage d'origine, banque 1 ACTIVÉ
G59.2 à G59.5	–	DO banques 2-5	6e décalage d'origine, banques 2 à 5 ACTIVÉ
G61		Arrêt précis	Arrêt précis ACTIVÉ
G61(IPS...)	–	Arrêt précis	Arrêt précis ACTIVÉ, avec fenêtre d'arrêt précis IPS1, IPS2 ou IPS3
G62	–	Arrêt précis	Arrêt précis DÉSACTIVÉ
G63	–	–	G63(M3/M4, S.../H...) F... Z : Taraudage sans mandrin de compensation
G70	–	Pouce/Métrique	Programmation en pouces, agit sur le parcours programmé, les longueurs, l'avances et les accélérations.
G71	–	Pouce/Métrique	Programmation métrique
G74	–	–	G74 X1 Y1 Z1... : Approche des coordonnées du point de référence
G74(Home)	–	–	G74(HOME) X1 Y1 Z1 : Approche du point de référence (vraie prise d'origine, même pour les axes asynchrones)
G75	–	–	Déplacement contre le palpeur de mesure (interruption du mouvement)
G76	–	–	Approche d'une position fixe de la machine (coordonnées de machine)
G77	–	–	G77 <coord 1><mode> <coord n><mode> ... F<valeur> Syntaxe alternative : REPOS Sous-programmes asynchrones : repositionnement de coordonnées individuelles
G78	–	–	G78({{-}<coordonnée1>},{-}<coordonnée2>},{-}<coordonnée3>}) : Commutation de correction ACTIVÉE

Nom de la fonction CN		Groupe (« – » signifie : effet au sein du bloc)	Désignation/Explication
Format long	Format abrégé		
G79	–	–	Commutation de correction DÉACTIVÉE
G90	–	Abs./Rel.	Programmation absolue
G91	–	Abs./Rel.	Programmation relative
G93	–	Programmation de l'avance	Programmation de temps
G94	–	Programmation de l'avance	Programmation de l'avance (par minute)
G94(...)	–	Programmation de l'avance	G94({DF <valeur> } { DS1 <valeur> , DS2 <valeur> , ... }) : Programmation incrémentale de la vitesse avec adaptation de l'accélération
G95	–	Programmation de l'avance	Programmation de l'avance (par tour)
G96	–	Programmation de la broche	G96({<axe de référence>{,<point d'action>}}) : Vitesse de coupe constante
G97	–	Programmation de la broche	Programmation directe de la vitesse de rotation
GCT(0)	–	–	Compensation des erreurs de rectitude et d'angle : mise hors service
GCT(1)	–	–	Compensation des erreurs de rectitude et d'angle : mise en service
GetAxis	GAX	–	GAX(...) : Reprendre l'axe libre sur le canal
GETERR	–	CPL	GETERR(<canal> [,<catégorie>], <code erreur> [,<nombre>]) : Fournit le code erreur, le numéro du canal et la catégorie d'erreur concernés par les erreurs actuelles.
GetSpindle	GSP	–	GetSpindle (<NoBrSys> <NomBrSys>,<NoBrCan> <NomBrCan> , {...}) : Création des broches du canal
GMSG	–	–	GMSG (<texte de remarque>) : Programmer des remarques dans l'interface utilisateur
GoAhead	GOA	–	Sauter en avant
GoBack	GOB	–	Sauter en arrière
GoCond	GOC	–	Saut conditionnel
GOTO	–	CPL	GOTO <destination de saut> : Sauts de programme inconditionnels vers un numéro de ligne, un numéro de bloc ou un label
GoTo	–	–	Saut inconditionnel
HsBlkSwitch	HSB	–	HSB(HS<X>=<Y>) : Changement de bloc au vol via le signal HighSpeed

Annexe

Nom de la fonction CN		Groupe (« – » signifie : effet au sein du bloc)	Désignation/Explication
Format long	Format abrégé		
HsBlkSwitch(..HS STOP=..)	HSB	–	HSB(...,HSSTOP=..) : Changement de bloc avec interruption via le signal HighSpeed
HWOC	–	–	HWOC({CHAN<NoCanal>},CRDNO <NoCoord.>,{ STEP<incr>}) : Correction en ligne dans le système de coordonnées de la pièce à usiner ACTIVÉE HWOCDIS({CHAN<no du canal>}) : Correction en ligne DÉSACTIVÉE, enregistrer les valeurs HWO() : Correction en ligne DÉSACTIVÉE, supprimer les valeurs
IC(...)	–	–	Programmation relative locale par ex. X=IC(5)
IF ENDIF	–	CPL	IF <condition> THEN <routine> [ELSE <routine alternative>] ENDIF : Saut conditionnel vers une routine ou une routine alternative
InitMeas	IME	–	IME(MpiAxis i) : Initialiser la mesure au vol
INP#	–	CPL	INP#(<n>,<variable>[,<variable>] [...] [:]): Accès en lecture à un fichier
INSDEP	–	–	Profondeur de plongée, correction de rayon 3D
INSTR	–	CPL	INSTR (<chaîne de caractères>,<expression STRING>[,<point de départ>]) : Recherche une <chaîne de caractères> dans une <expression STRING> à partir du <point de départ>, et affiche sa position de dé- part en tant que valeur INTEGER.
INT	–	CPL	<nombre INTEGER>=INT(<expression REAL>) : Convertit une <expression REAL> en un <nombre INTEGER> en tronquant les nombres décimaux.
InvTime	ITIM		ITIM <valeur> Programmation – Temps inversé
IPS1 ISP2 ISP3	–	–	IPS1 : fenêtre d'arrêt précis exacte IPS2 : fenêtre d'arrêt précis approximative IPS3 : fenêtre d'arrêt précis sans fin
JerkAxlList	JAL	–	Déterminer les axes système pour la définition du filtre Shape opti- mal en cas de limitation des secousses active
JerkControl	JKC	–	Mode Jog en coordonnées de la pièce à usiner
JogWCSSelect	–	–	Sélectionne les coordonnées pour le déplacement en mode JOG en coordonnées de la pièce à usiner (mode de réglage).

Nom de la fonction CN		Groupe	Désignation/Explication
Format long	Format abrégé	(« – » signifie : effet au sein du bloc)	
KinSplitDisable	KSD	–	KinSplitDisable(0) Activer la répartition cinématique des blocs CN KinSplitDisable(1) Désactiver la répartition cinématique des blocs CN
KvProg	KVP	Programmation KV	KVP(Y..., Y..., Z) : Programmation KV ACTIVÉE KVP() : Programmation KV DÉSACTIVÉE
LCP	–	–	Compensation de poursuite Permet de coupler un ou plusieurs axes supplémentaires aux maîtres définis dans la configuration (ou de les découpler).
LEN	–	CPL	LEN(<expression STRING>) : Fournit le nombre de caractères d'une expression STRING.
LEN	–	–	LEN=<valeur> : Divise le bloc de déplacement en plusieurs parcours partiels de même longueur.
LFCnf	LFC	–	LFC({LL(...)}) : Paramétrer le réglage de la puissance du laser
LFP	LFP	–	LFP, LFP(1), LFP({LL(...)}) : Réglage de la puissance du laser ACTIVÉ LFP(0) : Réglage de la puissance du laser DÉSACTIVÉ
LinDownFeed	LND	Profil en V	Freinage linéaire
LinearToSpline	LTS	–	Commutation de la programmation linéaire/spline LinearToSpline(1) : Activer LinearToSpline(0) : Désactiver
LinModZp	LMZ	–	LMZ{{LinModAxis i}} : Remise à zéro de l'axe modulo linéaire (Zp = zero point)
LinUpFeed	LNU	Profil en V	Accélération linéaire
LJUST	–	CPL	Commutation à l'affichage des données justifié à gauche, effectif jusqu'à la fin du déroulement du programme
M	M	–	Sous-programmes locaux définis par l'utilisateur
M0	M0	–	Arrêt du programme
M1	M1	–	Arrêt conditionnel du programme
M19, M119, M219	–	–	Orientation de la broche
M2	M2	–	Fin du programme principal
M3, M103, M203, M13, M113, M213	–	–	Rotation à droite de la broche Rotation à droite de la broche avec liquide de refroidissement ACTIVÉ
M30	M30	–	Fin du programme principal

Annexe

Nom de la fonction CN		Groupe (« – » signifie : effet au sein du bloc)	Désignation/Explication
Format long	Format abrégé		
M4, M104, M204, M14, M114, M214	–	–	Rotation à gauche de la broche Rotation à gauche de la broche avec liquide de refroidissement ACTIVÉE
M40, M140	–	–	Sélection automatique du rapport de réduction
M41 ... M44, M141... M241 ... M244	–	–	Sélection manuelle du rapport de réduction
M48, M148, M248	–	–	Débrayer le rapport de réduction
M5, M105, M205	–	–	Arrêt de la broche
MainSp	MSP	–	Sélection de la broche principale, par ex. pour G33 et G95
MCA	–	CPL	MCA(<bloc>,<index>[,<canal>]) : Transfère le contenu d'un paramètre individuel MACODA.
MCODS	–	CPL	MCODS(<type>,<canal>,<version>,<mémoire tampon>,<taille>[,<P1>]) : Appelle les services de données Motion Control de l'interface NCS. Cela permet d'extraire les états et les données de la commande numérique.
MCOPS	–	CPL	MCOPS(<fct>,<canal>[[,<P1>][,<P2>], [<P3>]],<P4>])) Appelle les services de processus Motion Control de l'interface NCS. Cela permet de commander les canaux de la commande numérique.
MCS	–	CPL	MCS(<coordonnée> [,<mode de sélection>[,<canal>]]) : Fournit la position actuelle de la machine (MCS) pour une coordonnée de machine sans compensation des erreurs de rectitude et d'angle.
MID\$	–	CPL	MID\$(<expression STRING>,<point de départ> [,<nombre de caractères>]) : Extrait une partie d'une expression STRING et l'affiche sous forme de texte. Le résultat peut être transmis à une variable STRING ou à un tableau dimensionné en conséquence. MID\$(<tableau>,<point de départ>[,<nombre de caractères>]) : Écrase des parties d'un tableau.
Miroir	MIR	Fonction miroir	MIR(X1, Y1,Z1,...) : Inversion des coordonnées ACTIVÉE MIR () : Inversion des coordonnées DÉSACTIVÉE
MMC	–	CPL	MMC(<VarCPL1>[,<VarCPL2>... [,<VarCPLn>]...]) : Au cours de l'exécution, envoie les informations concernant la durée d'un programme pièce à un client et attend le résultat de ce client.

Nom de la fonction CN		Groupe (« – » signifie : effet au sein du bloc)	Désignation/Explication
Format long	Format abrégé		
MOC			MOC(MP<palpeur de mesure> CI<entrée client> HS<HighSpeed>{, ES<flanc>}, RE<réaction>{, AS<ASUP>}{, ER<erreur>}{, <nom d'axe du canal1>{, <nom d'axe du canal2> ...}}) Mesure sur un contour
Modulo	MOD	–	Calcul modulo pour les axes sans fin à déplacement incrémental
MPEDGE			MPEDGE(MP1 MP2{,ES1 ES2}) Modifier le palpeur de mesure de la configuration des flancs
MSG	–	–	MSG (<texte de remarque>) Programmer des remarques dans l'interface utilisateur
NCF	–	CPL	NCF (<fonction CN>) Transmet la syntaxe de la fonction CN active au sein du groupe mo- dal CN de la <fonction CN>.
Nibble	NIB	Poinçonnage	NIB(NUM...) : Grignotage ACTIVÉ NIB() : Grignotage DÉSACTIVÉ
NJUST	–	CPL	Commutation prématurée de la sortie des données justifiée à gauche à la sortie formatée
NOT	–	CPL	NOT <expression> : Négation d'une expression BOOLE ou négation par bit d'une expres- sion INTEGER
NUL	–	CPL	<variable> = NUL : Effacement d'une variable
NUM	–	–	NUM=<valeur> : Divise le bloc de déplacement en un nombre défini de parcours par- tiels de même longueur.
O	O	–	O(...) : Mouvement d'orientation pour un outil
OFFSTOPA	–	–	Instruction de synchronisation de canal : supprime les « conditions d'arrêt ET » programmées sur le canal de commande (ASTOPA, BSTOPA, WSTOPA).
OFFSTOPO	–	–	Instruction de synchronisation de canal : supprime les "condition d'arrêt OU" programmées sur le canal de commande (ASTOPO, BSTOPO, WSTOPO).
Omega	–	–	Adresse oméga
OPENR	–	CPL	OPENR(<n>,<nom progr>[,<longueur d'enregistrement>]) : Ouvre un fichier pour y accéder en lecture.
OPENW	–	CPL	OPENW(<n>,<nom progr>[,<param. vide>][,<commentaire>][,<lon- gueur d'enregistrement>]) Ouvre un fichier pour y accéder en écriture.

Annexe

Nom de la fonction CN		Groupe (« – » signifie : effet au sein du bloc)	Désignation/Explication
Format long	Format abrégé		
OR	–	CPL	<expression> OR <expression> : Liaison binaire de deux expressions BOOLE ou INTEGER à l'aide de la fonction OU
OvrDis	OVD	Avance 100 %	Potentiomètre d'avance DÉSACTIVÉ
OvrEna	OVE	Avance 100 %	Potentiomètre d'avance ACTIVÉ
P	P	–	Appel de sous-programme (aussi possible directement sans P)
PL	–	–	Programmation de la longueur des paramètres de la spline
PathAcc	PAC	Accélération sur trajectoire	PAC(ACC... UP..., DOWN ...) : Activer PAC () : Désactiver Programmation de l'accélération sur trajectoire, soit de manière séparée pour l'accélération et le freinage, soit de manière globale
PCS	–	CPL	PCS(<coordonnée>[,<mode de sélection>]) : Transmet la dernière position absolue programmée d'une coordonnée.
PCSPROBE	–	CPL	PCSPROBE (<coordonnée>[,<mode de sélection>]) : Lit la valeur mesurée pour une coordonnée.
PDIM	–	CPL	PDIM <nom du paramètre>(<taille du tableau>) : Doit être utilisé : - si un sous-programme doit être appelé avec une constante STRING en tant que paramètre de transfert et - si le programme appelant est sélectionné sans liaison.
PLC	–	CPL	PLC(<type>,<param. vide>,<adresse>,<taille>) : Permet d'accéder aux opérandes de l'API.
PMT	–	CPL	PMT(<sélection de coord.>,<code de pos.>[,<banque de pos.>][,<table>][,<unité>]]]) : Accès en lecture et en écriture à une table de placement XML
PmTsel	PMS	–	PMS(<nom de tableau de placement>) : Sélection d'un tableau de placement
PMV	–	CPL	PMV(<sélection de coord.>[,<banque de pos.>]) : Fournit les valeurs de placement effectives pour une coordonnée.
PolarPol	POP	PolarPol	POP(X...,Y...,Z...) : Définir le pôle POP () : Pôle à l'origine Pôle pour la programmation de coordonnées polaires
PoleSet	PLS	Définir le pôle	PLS(X..., Y..., Z...) : Définir le pôle PLS () : Pôle à l'origine Pôle pour les aides à la saisie « Fonction miroir » et « Rotation »

Nom de la fonction CN		Groupe	Désignation/Explication
Format long	Format abrégé	(« – » signifie : effet au sein du bloc)	
PosDepHSOut	PHS	–	PHS(<mode>{,<distance>,<durée>}) : Sortie à grande vitesse programmable en fonction de la position
PosMode	PMD	Mode de positionnement	PMD(A..., B...) : Commuter le mode de positionnement PMD() : Mode de positionnement selon le paramètre machine Mode de positionnement pour axes sans fin
PPOS	–	CPL	PPOS(<sélection d'axe>[,<type d'axe>]) : Interrogation de la position réelle de l'axe au moment de la commutation du palpeur de mesure
PrecProg	PRP	Programmation de précision	PRP(DIST... EPS...) : Programmation de précision ACTIVÉE PRP() : Programmation de précision DÉSACTIVÉE
PREP	–	–	PREP(<NbBlocs> { , SILENT }) Traitement du programme en fonction de la préparation des blocs
PRN#	–	CPL	PRN#(<n>[,<expression>][,<expression>] [,<expression>][,...][;]) : Accès en écriture à un fichier
PROBE	–	CPL	PROBE(<sélection d'axe>[,<type d'axe>]) : Interrogation de la position de l'axe au moment de la commutation du palpeur de mesure, par rapport aux coordonnées d'origine de l'axe du MCS
PtBlkEnd	PTE	Course de poinçonnage	PTE(X..., Y..., Z) : Déclenchement de course à la fin du bloc
PtDefault	PTD	Course de poinçonnage	Déclenchement de la course selon le paramétrage
PtInpos	PTI	Course de poinçonnage	PTI(X..., Y..., Z) : Déclenchement de la course avec Inpos (Pt = punch time)
PtpMove	PTP	–	Si une transformation d'axe est active, un mouvement de déplacement point-à-point (PTP) est exécuté.
Punch	PUN	Poinçonnage	PUN(NUM...) : Découpage-poinçonnage ACTIVÉ PUN() : Découpage-poinçonnage DÉSACTIVÉ
RadProg	RAD	Diamètre/Rayon	Programmation du rayon
RadialAcc	RAC	–	RadialAcc (<valeur>) : Paramétrage de l'accélération radiale maximale RadialAcc(0) : Annuler la limitation de l'accélération radiale
RecordRevVec	RRV	–	RRV(...) Enregistrer le vecteur de retrait
RedTorque	RDT	–	RDT(<axe1>{,<axe2>{, ...}}) : Détermination du couple pour la réduction du couple
ReentContBlk	RCB	–	Poursuite du programme avec avance de bloc après l'arrêt du programme

Annexe

Nom de la fonction CN		Groupe (« – » signifie : effet au sein du bloc)	Désignation/Explication
Format long	Format abrégé		
REM	–	CPL	REM <texte du commentaire> : Commentaire au programme
RemAxis	RAX	–	RAX(...) : Extraire l'axe du canal
RemLogName	RLN	–	RLN(...) : Supprimer le nom d'axe logique
RemSpindle	RSP	–	RemSpindle (<NoBrSys> <NomBrSys> <NomBrCan> , {...}) : Supprimer les broches du canal
REPEAT	–	CPL	REPEAT <routine> UNTIL <condition> : Construction en boucle avec interrogation de la condition d'interruption après le premier traitement de la routine
REPEAT	RPT		RPT<facteur de répétition> Répétition du bloc
REPOSDEF	–	–	Définit localement (au sein d'un sous-programme asynchrone) l'avance de redémarrage et/ou le point de redémarrage.
REPOSTP	–	–	REPOSTP(<point>) Sous-programmes asynchrones : définir le point de redémarrage dans le sous-programme asynchrone.
RefTo	–	–	Saut en arrière d'un sous-programme vers un bloc-programme quelconque
REV	–		REV(<NoÉvénement> { , <NoÉvénement> ...}) : Un événement bit est effacé au cours de l'exécution.
REWRITE	–	CPL	REWRITE(<n>) : Écrasement d'un fichier existant
Rotate	ROT	Tourner	ROT(<angle>) : Rotation des coordonnées ACTIVÉE ROT() : Rotation des coordonnées DÉSACTIVÉE
ROTAX	–	–	Définir l'axe de rotation pour l'orientation vectorielle
ROUND	–	CPL	<nombre INTEGER> = ROUND(<expression REAL>) : Conversion d'une expression REAL en un nombre INTEGER, en l'arrondissant
RoundEps	RNE	Chanfrein/Arrond.	RNE(<valeur>) : Arrondissage ACTIVÉ RNE() : Arrondissage DÉSACTIVÉ Arrondissage avec écart de contour défini
Rounding	RND	Chanfrein/Arrond.	RND(<valeur>) : Arrondissage ACTIVÉ RND() : Arrondissage DÉSACTIVÉ Arrondissage avec rayon d'arrondissage défini

Nom de la fonction CN		Groupe (« - » signifie : effet au sein du bloc)	Désignation/Explication
Format long	Format abrégé		
S	S	–	S<numéro>=<valeur> S<valeur> : Programmer la vitesse de rotation de la broche
Scale	SCL	Mise à l'échelle	SCL(X..., Y..., Z) : Mise à l'échelle des coordonnées ACTIVÉE SCL() : Mise à l'échelle des coordonnées DÉACTIVÉE
SCL	–	CPL	SCL(<sélection SCL>[, <sélection> [, <mode de sélection>]]) : Fournit les derniers paramètres programmés des fonctions PLS et ROT pour le canal actuel (coordonnées polaires, facteurs d'échelle et angles de rotation).
SCS3	–	CPL	SCS3(<index d'axe>, <n° EID> [, <ERRNO>]) : Accès en lecture aux paramètres d'entraînement Sercos avec numé- ro d'identification élargi
SCSL3	–	CPL	SCSL3(<index d'axe>, <n° EID>, <nom de fichier>[, <ERRNO>]) : Création d'un fichier pour les listes de paramètres Sercos avec nu- méro d'identification élargi
SD	–	CPL	SD(<groupe>[, <index1>[, <index2> [, <index3>]]]) : Lecture des données système actives de la commande CN
SDLOAD	–	CPL	SDLOAD(<nom de fichier>, <élément_racine_1>[, ..., <élément_raci- ne_n>]) Lire les données système à partir d'un fichier
SDR	–	CPL	SDR(<groupe>[, <index1>[, <index2>]]) : Lecture des données système actives de la commande CN au for- mat REAL
SDSAVE	–	CPL	SDSAVE(<nom de fichier>, <élément_racine_1>[, ..., <élément_raci- ne_n>])
SEEK	–	CPL	SEEK(<n>, <k>[, <o>]) : Positionne le pointeur du fichier sur le <k>ième enregistrement d'un fichier aléatoire ou sur le <k>ième octet d'un fichier séquentiel.
SelCrdCouple	SCC	–	SCC(SC<canal>, CL(<Q1>, <Z1>, {...})) : Couplage sélectif additif de coordonnées ACTIVÉ SCC() : Tous les couplages de coordonnées DÉACTIVÉS
SelCrdCoupleTab	SCCT	–	Couplage sélectif additif de coordonnées avec table
SETERR	–	CPL	Générer une erreur d'exécution après l'évaluation d'ERRNO
SetPos	SPS	–	Initialiser la position du programme
SETWARN	–	CPL	Générer un message d'avertissement après l'évaluation d'ERRNO

Annexe

Nom de la fonction CN		Groupe (« - » signifie : effet au sein du bloc)	Désignation/Explication
Format long	Format abrégé		
SEV	-		SEV(<NoÉvénement> {, <NoÉvénement> ...}) : Un événement bit est déclenché (défini) au cours de l'exécution.
Shift	SHT	-	SHT(X..., Y..., Z) : Décalage de contour ACTIVÉ SHT() : Décalage de contour DÉSACTIVÉ
Sin2DownFeed	S2D	Profil en V	Freinage sous forme de $\sin^2$
Sin2UpFeed	S2U	Profil en V	Accélération sous forme de $\sin^2$
SinDownFeed	SND	Profil en V	Freinage sous forme de $\sin$
SinUpFeed	SNU	Profil en V	Accélération sous forme de $\sin$
Smax	SMX	-	SMX(<valeur>) : Vitesse de rotation maximale pour la broche
Smin	SMN	-	SMN(<VALEUR>) : Vitesse de rotation minimale pour la broche
SpAdjSp	-		SpAdjSp Comparer les broches dans le programme de comparaison Init de l'avance de bloc
SpAdjStr	-	-	SpAdjStr<broche> Déterminer la syntaxe de broche pour le programme de comparaison de l'avance de bloc
SpAdmin	SPA	-	SPA(Si=0 1) : Reprise ou libération conditionnelle de la broche réservée par un autre canal
SpCouple_Wait	SPC_WA IT	-	SPC_WAIT(CP=1..4) : Attente du mode synchrone du groupe de couplage indiqué
SpCoupleConfig	SPCC	-	SPCC(CP=1..4, MA=<maître>, Si=1, Sj=1, ...) : Définir le groupe de couplage, supprimer ou ajouter des broches esclaves, supprimer le groupe de couplage
SpCoupleDist	SPCD	-	SPCD(S1=<écart>, S2=<écart>, ...) : Définir l'écart de couplage pour les couplages de broches.
SpCoupleErrWin	SPCE	-	SPCE(S1=<fenêtre>, S2=<fenêtre>, ...) : Définir la fenêtre d'erreur du mode synchrone pour les couplages de broches
SpCouplePosOffs	SPCP	-	SPCP(S1=<décalage>, S2=<décalage>, ... {POSVEL<vitesse de rotation>}) : Définir le décalage angulaire pour le groupe de couplage de broches actif
SpCouplePosOffs_Wait	SPCP_W AIT	-	SPCP_WAIT(CP=1..4) : Attente du décalage angulaire pour le groupe de couplage programmé

Nom de la fonction CN		Groupe	Désignation/Explication
Format long	Format abrégé	(« – » signifie : effet au sein du bloc)	
SpCoupleSync-Win	SPCS	–	SPCS(S1=<fenêtre>, S2=<fenêtre>, ...) : Définir la fenêtre du mode synchrone pour les couplages de broches
SPG	–	–	SPG<groupe>(<numéros>) SPGALL(0) : Définir/dissocier les groupes de broches
SplineCornering	SCO	–	SplineCornering(E<écart>) SplineCornering(L1=<L1>,L2=<L2>) : Arrondissage des angles avec des splines
SplineCorneringLocal	SCOL	–	Arrondissage des angles avec des splines (localement)
SplineDef	SPD	–	Détermination d'une variante spline
SpindleToAxis	STA	–	STA(...) Commuter la broche au mode axe
SpindleToAxis-Wait	STAW	–	SpindleToAxisWait(<NPA> <IPA>{ ,<NPA> <IPA>}...) Activer le mode d'axe C pour les broches et attendre
Split	SPT	–	Split({<mode>},{<longueur partielle>}) : Divise les blocs de déplacement en plusieurs segments de parcours s'ils dépassent une certaine longueur.
SpindleWait	SPWAIT	–	SpindleWait(<broche> { ,<condition>}) : À la fin du bloc, attend jusqu'à ce que la broche correspondante ait atteint l'état voulu.
SpindleWaitMode	SWM	–	SpindleWaitMode(<broche>,<activer> { ,<stratégie>}) : Stratégie d'attente pour la fonction "Mouvement sur trajectoire et fonctions de la broche"
SpMode	SPM	–	SPM(S1=0 1, S2=0 1, ...) : Commuter manuellement l'interface d'entraînement de la broche entre le mode d'asservissement en vitesse de rotation et le mode d'asservissement en position
SPOS	–	CPL	SPOS(<sélection d'axe>) : Transmet la valeur de consigne actuelle de l'axe d'un axe physique.
SPV	–	–	SPV [<var. CPL perm.> = <expression CPL simple>] Instruction de synchronisation de canal : la variable CPL permanente est écrite lors de l'exécution.
SPVE	–	–	SPV E[<var. CPL perm.> = <expression CPL>] Instruction de synchronisation de canal : la variable CPL permanente est écrite lors de l'exécution. L'expression CPL peut inclure des ordres CPL, mais elle est déjà terminée lors de la préparation des blocs.

Annexe

Nom de la fonction CN		Groupe (« – » signifie : effet au sein du bloc)	Désignation/Explication
Format long	Format abrégé		
SQRT	–	CPL	<valeur de fonction> = SQRT(<valeur entrée>) : Application de la fonction racine carrée à la <valeur entrée>.
SSD	–	CPL	SSD[<date du système> =<expression CPL simple>] La valeur à attribuer n'est déterminée que pendant la durée d'exécution.
SSDE	–	CPL	SSDE[<date du système> =<expression CPL>] La valeur à attribuer est déterminée au moment de la préparation.
SSDQ	–	–	SSDQ(M <message> {, Q <XPath>} {, C <XPath>}) Écriture dans une queue de données système
SSpAdm	–	–	SSpAdm(<NoBrSys> <NomBrSys>, <mode> {, <NoBrSys> <NomBrSys>, <mode>} ...): Valider/Reprendre la broche réservée
SSpAdmOff	–	–	SSpAdmOff(<NoBrSys> <NomBrSys>, <NoCanal> {, <NoCanal> } ...)SSpAdmOff(<NoBrSys> <NomBr- Sys>): Désactiver la gestion de mouvement des broches pour certains ca- naux
SSPG	–	–	SSPG<groupe>=<valeur> : Définition de la vitesse de rotation pour le groupe de broches, voir également « S<valeur> »
SSpGear	–	–	SSpGear(<NoBrSys> <NomBrSys>, <rapport>{, <NoBrSys> <NomBrSys>, <rapport>} ...): Changer le rapport de réduction
SSpMax	–	–	SSpMax(<NoBrSys> <NomBrSys>, <vitesse de rotation>{, <NoBr- Sys> <NomBrSys>, <vitesse de rotation>} ...): Limiter la vitesse de rotation maximale
SSpMin	–	–	SSpMin(<NoBrSys> <NomBrSys>, <vitesse de rotation>{, <NoBr- Sys> <NomBrSys>, <vitesse de rotation>} ...): Limiter la vitesse de rotation minimale
SSpMode	–	–	SSpMode(<NoBrSys> <NomBrSys>, <mode> {, <NoBrSys> <NomBrSys>, <mode>} ...): Commutation entre l'interface de position/de vitesse de rotation
SSpMove	SSPM	–	SSpMove(<NoBrSys> <NomBrSys>, <ordre>{, <NoBrSys> <NomBrSys>, <ordre>} ...): Programmation de mouvement
SSpOri	SSPO	–	SSpOri(<NoBrSys> <NomBrSys>, <angle>{, <NoBrSys> <NomBr- Sys>, <angle>} ...): Orientation de la broche

Nom de la fonction CN		Groupe	Désignation/Explication
Format long	Format abrégé	(« – » signifie : effet au sein du bloc)	
SSpSpeed	SSPS	–	SSpSpeed(<NoBrSys> <NomBrSys>, <vitesse de rotation> { , <NoBrSys> <NomBrSys>, <vitesse de rotation> } ...) : Programmation de la vitesse de rotation
StatToolOri	STO	–	StatToolOri({<φ>} {,<θ>} {,<ψ>}) Paramétrage de l'orientation statique de l'outil
StepModeDisable	SMD	–	StepModeDisable({{variable={}<valeur>}}{,<valeur>}}) : Sert à traiter les éléments d'un programme CN en une seule fois dans l'un des types de fonctionnement Étape unique, Bloc unique, Bloc-programme et Étape de débogage. Cela signifie que les blocs sont traités sans confirmer le démarrage CN.
StepModeEnable	SME	–	StepModeEnable Il n'existe aucun paramètre programmable. Le bloc dans lequel SME est programmé compte encore parmi les blocs pour lesquels le démarrage CN est supprimé.
STR\$	–	CPL	STR\$([<chaîne de caractères de format>,<valeur>) : Transforme l'expression numérique <valeur> en une chaîne de caractères qui ne peut être affectée qu'à un seul tableau. Si la <chaîne de caractères de format> est programmée, la chaîne de caractères peut être affichée de manière formatée.
TAN	–	CPL	<valeur de fonction> = TAN(<valeur entrée>) : Application de la fonction tangente à la <valeur entrée>.
TangTool	TTL	Orientation de l'outil	TTL(TAX..., SYM..., ANG..., IA..., PLC...) : Guidage tangentiel de l'outil ACTIVÉ TTL() : Guidage tangentiel de l'outil DÉSACTIVÉ
TangToolOri	TTO	Orientation tangentielle de l'outil	TTO(SYM..., ANG...) : Orientation tangentielle de l'outil ACTIVÉE TTO() : Orientation tangentielle de l'outil DÉSACTIVÉE
TappRet1	–	–	Commute au mode d'asservissement en position les broches qui tournent de nouveau en mode d'asservissement en vitesse de rotation après la remise à zéro (montée en régime de la commande).
TappRet2	–	–	Déclenche le mouvement de remontée effectif. Ceci s'effectue avec la valeur F programmée.
TappSp	TSP	–	TSP(CAXi, ..., GRPj, ...) : Sélection de broches pour le taraudage sans mandrin de compensation
TargetAxisConf	TAX		TAX ou TargetAxisConf Comparaison de la configuration d'axe
TCM(, ,)	–	–	TCM({<valX>},{<valY>},{<valZ>}) : Déplacement dans le TCS

Annexe

Nom de la fonction CN		Groupe (« – » signifie : effet au sein du bloc)	Désignation/Explication
Format long	Format abrégé		
TcsDef	TCS	–	Déterminer explicitement la position du système de coordonnées de l'outil (correction de la longueur de l'outil implicite en combinaison avec la transformation d'axe correspondante)
TCV	–	CPL	TCV(<sélection de valeurs>[,<sélection de correction>]) : Fournit les dernières valeurs de correction d'outil programmées.
ThreadSet	TST	–	Filetage, fonctions supplémentaires
TIME	–	CPL	<variable STRING> = TIME : Attribue l'heure à une variable STRING sous la forme HH.MM.SS.
Trans	TRS	Décalage des coordonnées du programme	TRS(X ..., Y..., Z) : Décalage des coordonnées du programme ACTIVÉ TRS () : Décalage des coordonnées du programme DÉACTIVÉ
TRIM\$	–	CPL	TRIM\$(<chaîne de caractères>) TRIM\$(<chaîne de caractères>,,L") TRIM\$(<chaîne de caractères>,,R") Fournit une chaîne de caractères sans espace de tête (index « L ») ou de queue (index « R ») lors de l'attribution d'une zone du tableau à une variable STRING ou à un tableau. S'il n'existe aucun index, les espaces de tête et les espaces de queue sont masqués.
TRUE	–	CPL	<variable booléenne> = TRUE : Valeur de vérité d'une variable BOOLE
VAL	–	CPL	VAL(<expression STRING>) : Fournit la valeur numérique de l'expression STRING.
VARINF	–	–	VARINF(<nom de variable>) Nom de la donnée système structurée ou de la variable permanente
VERSINF\$	–	CPL	VERSINF\$(<index1>[,<index2>]) : Appelle des données administratives, par ex. la version du logiciel.
VirtAxisPos	VAP	–	Définit la position d'axe des axes synchrones virtuels sur le canal actuel.
VREC_START	–	–	Démarrer l'enregistrement du diagnostic pour le guidage de vitesse
VREC_STOP	–	–	Terminer l'enregistrement du diagnostic
WAIT	–	–	WAIT dans le bloc CN (identique à CPL WAIT)

Nom de la fonction CN		Groupe (« – » signifie : effet au sein du bloc)	Désignation/Explication
Format long	Format abrégé		
WAIT	–	CPL	WAIT(sans paramètres) : Arrête le traitement de bloc jusqu'à ce que tous les blocs programmés avant WAIT aient été traités. WAIT(<temps d'attente>) : Arrête le traitement de bloc jusqu'à ce que l'intervalle déterminé se soit écoulé. WAIT(BITIF(...)) : Arrête le traitement de bloc jusqu'à ce qu'un état précis se présente à l'interface de bit CN-API.
WAITA	–	–	WAITA[BITIF(<paramètre>){=<état>} {...}{,<délai d'expiration>}] Instruction de synchronisation de canal : en cours d'exécution, attendre que tous les signaux d'interface interrogés soient émis ou que le délai d'expiration se soit écoulé.
WaitAxis	WAX	–	WAX(...) : Attendre que l'axe soit libéré, puis le reprendre sur le canal.
WAITO	–	–	WAITO[BITIF(<paramètre>){=<état>} {...}{,<délai d'expiration>}] Instruction de synchronisation de canal : en cours d'exécution, attendre que l'un des signaux d'interface interrogés soit émis ou que le délai d'expiration se soit écoulé.
WCS	–	CPL	WCS(<coordonnée> [,<mode de sélection>[,<canal>]]) : Fournit la position actuelle de la pièce à usiner sans valeurs de correction en ligne pour une coordonnée.
WEV	–		WEV(<NoÉvénement> {, NoÉvénement ...}) : En cours d'exécution, permet d'attendre qu'un événement bit défini se produise (ou d'empêcher la commutation de bloc).
WHILE	–	CPL	WHILE <condition> DO <routine> END : Construction en boucle avec interrogation de la condition d'interruption avant la première exécution de la boucle.
WPV	–	–	WPV[<var. CPL perm.><op. comparaison> <expression CPL simple>{,<délai d'expiration>}] Instruction de synchronisation de canal : en cours d'exécution, attendre que l'expression soit remplie ou que le délai d'expiration se soit écoulé.
WPVE	–	–	WPVE[<var. CPL perm.><op. comparaison> <expression CPL>{,<délai d'expiration>}] Instruction de synchronisation de canal : en cours d'exécution, attendre que l'expression soit remplie ou que le délai d'expiration se soit écoulé. L'expression CPL peut inclure des ordres CPL, mais elle est déjà terminée lors de la préparation des blocs.

Annexe

Nom de la fonction CN		Groupe (« – » signifie : effet au sein du bloc)	Désignation/Explication
Format long	Format abrégé		
Writeld3	WID3	–	WID3(S-0-0104.0.0, X..., Y..., Z) : Écriture des paramètres Sercos avec numéro d'identification élargi
WSTOPA	–	–	WSTOPA[<NoCanal>,<cond.1>{,<cond.2>} {...,<cond.8>}...] Instruction de synchronisation de canal : le canal à commander est arrêté jusqu'à ce que toutes les conditions soient remplies. Les conditions se rapportent aux positions de la pièce à usiner, par ex. "Z" > 20.
WSTOPO	–	–	WSTOPO[<NoCanal>,<cond.1>{,<cond.2>} {...,<cond.8>}...] Instruction de synchronisation de canal : le canal à commander est arrêté tant que l'une des conditions est remplie. Les conditions se rapportent aux positions de la pièce à usiner, par ex. "Z" > 20.
XOR	–	CPL	<expression> XOR <expression> : Liaison binaire de deux expressions BOOLE ou INTEGER avec la fonction OU EXCLUSIF.
XTAB	–	CPL	XTAB(<table>,<structure partielle>) : Accès en lecture et en écriture à une table XML quelconque de type DCT, ZOT, PMT ou GCT
XTABCR	–	CPL	XTABCR(<table>,<type> [,<définition de table>]) : Crée une table XML sans données au sein du système de fichiers de la MTX.
ZOCDEL	–	CPL	ZOCDEL(<table>,<position>[,<canal>]) : Suppression d'un axe dans une table de décalage d'origine XML.
ZOCINS	–	CPL	ZOCINS(<table>,<position>,<nom d'axe> [,<TypeAxe>][,<CanalAxe>][,<CanalPos>]) : Ajout d'un nouvel axe dans la table de décalages d'origine XML avant une colonne existante
ZOT	–	CPL	ZOT(<sélection de colonne>,<code de décalage d'origine> [,<banque de décalage d'origine>][,<table>][,<unité>]) : Accès en lecture et en écriture à une table de décalage d'origine XML quelconque
ZOTCR	–	CPL	ZOTCR(<table>,<canal/modèle>) : Crée une table de décalage d'origine XML sans blocs de correction au sein du système de fichiers de la MTX.

Nom de la fonction CN		Groupe	Désignation/Explication
Format long	Format abrégé	(« – » signifie : effet au sein du bloc)	
ZoTSel	ZOS	–	ZOS(<nom DO>) : Sélection d'une table de décalage d'origine
ZOV	–	CPL	ZOV(<sélection d'axe>[.<banque de décalage d'origine>]) : Fournit les valeurs de décalage d'origine effectives pour un axe/une coordonnée de machine.

Tab. 8-1: Classement par ordre alphanumérique selon le format long

8.2 Jeu de caractères ASCII

ASCII signifie "American Standard Code for Information Interchange" (Code américain normalisé pour l'échange d'informations). Les signes comportant les numéros 00 à 31 et le signe avec le numéro 127 sont des signaux de commande.

Déc.	Hex.	ASCII
0	00	NUL
1	01	SOH
2	02	STX
3	03	ETX
4	04	EOT
5	05	ENQ
6	06	ACK
7	07	BEL
8	08	BS
9	09	HT
10	0A	LF
11	0B	VT
12	0C	FF
13	0D	CR
14	0E	SO
15	0F	SI
16	10	DLE
17	11	DC1
18	12	DC2
19	13	DC3
20	14	DC4
21	15	NAK
22	16	SYN
23	17	ETB

Annexe

Déc.	Hex.	ASCII
24	18	CAN
25	19	EM
26	1A	SUB
27	1B	ESC
28	1C	FS
29	1D	GS
30	1E	RS
31	1F	US
32	20	SP
33	21	!
34	22	"
35	23	#
36	24	\$
37	25	%
38	26	&
39	27	'
40	28	(
41	29	)
42	2A	*
43	2B	+
44	2C	,
45	2D	-
46	2E	.
47	2F	/
48	30	0
49	31	1
50	32	2
51	33	3
52	34	4
53	35	5
54	36	6
55	37	7
56	38	8
57	39	9
58	3A	:
59	3B	;
60	3C	<
61	3D	=

Annexe

Déc.	Hex.	ASCII
62	3E	>
63	3F	?
64	40	@
65	41	A
66	42	B
67	43	C
68	44	D
69	45	E
70	46	F
71	47	G
72	48	H
73	49	I
74	4A	J
75	4B	K
76	4C	L
77	4D	M
78	4E	N
79	4F	O
80	50	P
81	51	Q
82	52	R
83	53	S
84	54	T
85	55	U
86	56	V
87	57	W
88	58	X
89	59	Y
90	5A	Z
91	5B	[
92	5C	\
93	5D	]
94	5E	↑ (^)
95	5F	→ (→)
96	60	'
97	61	a
98	62	b
99	63	c

Annexe

Déc.	Hex.	ASCII
100	64	d
101	65	e
102	66	f
103	67	g
104	68	h
105	69	i
106	6A	j
107	6B	k
108	6C	l
109	6D	m
110	6E	n
111	6F	o
112	70	p
113	71	q
114	72	r
115	73	s
116	74	t
117	75	u
118	76	v
119	77	w
120	78	x
121	79	y
122	7A	z
123	7B	{
124	7C	
125	7D	}
126	7E	~
127	7F	DEL

Sur fond jaune : caractères normalement ignorés lors de la lecture.

Tab. 8-2: Jeu de caractères ASCII

8.3 Codes de touches supplémentaires

Code de touche (déc.)	Signification
134	CURSEUR VERS LE HAUT
135	CURSEUR VERS LE BAS
136	CURSEUR À DROITE
137	CURSEUR À GAUCHE
139	RETOUR DE NIVEAU

Code de touche (déc.)	Signification
141	SOFTKEY1
142	SOFTKEY2
143	SOFTKEY3
144	SOFTKEY4
145	SOFTKEY5
146	SOFTKEY6
147	SOFTKEY7
148	SOFTKEY8

Tab. 8-3: Codes de touches supplémentaires

8.4 Optimisation du temps d'usinage

8.4.1 Mesures ayant une influence directe sur la vitesse de traitement

Introduction

Ces mesures agissent directement sur la vitesse de traitement.

Pente de la trajectoire G8/G9 avec arrêt précis DÉSACTIVÉ "G62"

Sans "Pente de la trajectoire" (G9), la commande exécute un cycle de pente ascendante (upslope) et descendante (downslope) complet (rampe de vitesse) au début et à la fin d'un bloc de déplacement à $v = 0$.

Avec "Pente de la trajectoire" (G8), la commande tente de générer une vitesse la plus constante possible équivalente à l'avance programmée, même au niveau des enchaînements de blocs. Ceci réduit le temps d'usinage.

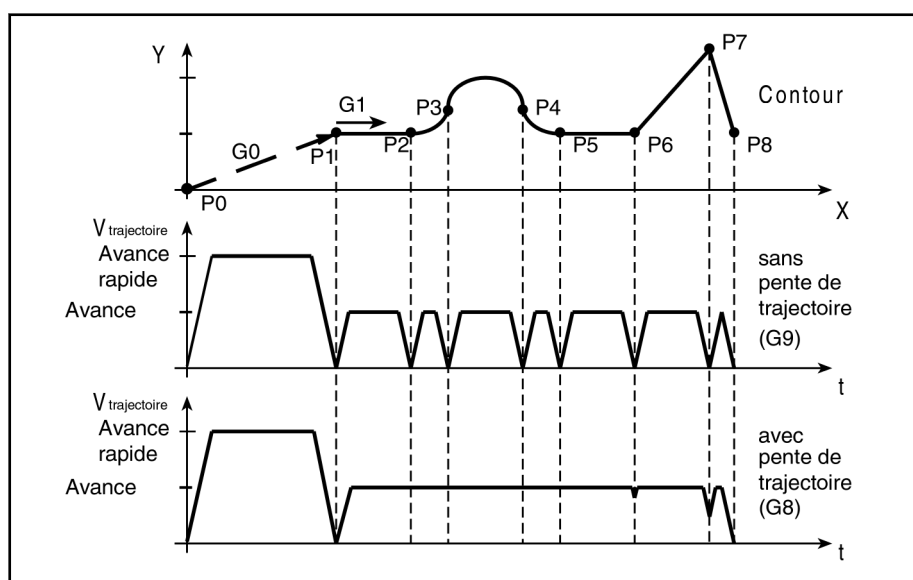


Fig. 8-1: Pente de la trajectoire

Pour que G8 soit effective, l'arrêt précis doit être mis hors service (G62).

Fenêtre INPOS avec arrêt précis ACTIVÉ "G61"

"Arrêt précis ACTIVÉ" doit être utilisé uniquement là où il faut vraiment attendre qu'une position exacte soit atteinte.

Annexe

Noter que plus il faut de temps pour atteindre la fenêtre INPOS au niveau des enchaînements de blocs, plus la zone INPOS configurée est petite et plus la qualité de régulation du circuit de réglage de la position est faible.

G61 (IPS1 IPS2 IPS3)	
avec	
IPS1	Arrêt précis ACTIVÉ Attendre Fenêtre de positionnement fine. À la fin du bloc, la commande freine d'abord à la vitesse de déplacement $v = 0$. Le bloc suivant n'est déplacé que lorsque cette fenêtre de positionnement a été atteinte pour tous les axes participants.
IPS2	Arrêt précis ACTIVÉ Attendre Fenêtre de positionnement brute. "Zone Inpos 2 activée" s'affiche sur l'interface du canal (voir aussi la documentation "Interface API IndraMotion MTX"). À la fin du bloc, la commande freine d'abord à la vitesse de déplacement $v = 0$. Le bloc suivant n'est déplacé que lorsque cette fenêtre de positionnement a été atteinte pour tous les axes participants.
IPS3	Arrêt précis ACTIVÉ Freinage à $v = 0$ à la fin du bloc. À la fin du bloc, la commande freine à la vitesse sur trajectoire $v = 0$. Ensuite, le bloc suivant est déplacé sur une fenêtre de positionnement sans être vérifié.

Tab. 8-4: Paramètres optionnels G61

Pour l'arrêt précis ACTIVÉ, la dernière fenêtre de positionnement active reste valide pour le mode trajectoire.

Fenêtre INPOS avec G0

G61 agit uniquement sur les mouvements en avance. Il n'agit pas sur des mouvements en avance rapide.

Avec G0, la fenêtre INPOS avec G0(NIPS) peut être désactivée.

Enchaînements de blocs continus

Des transitions lisses et constantes sont parcourues plus vite que des angles. Il est possible d'atteindre des transitions lisses avec les fonctions :

- SCO (arrondissement des angles avec des splines)
- RND (arrondissement des angles avec rayon)

Blocs CN

Chaque bloc CN programmé va généralement à l'interpolateur et demande donc un temps d'exécution moindre, même si aucun mouvement de déplacement n'a été programmé (au moins 1 cycle d'interpolation).

Il est judicieux de programmer les informations de non-déplacement avec les informations de déplacement dans un même bloc.

Exemple:

Économiser des cycles d'interpolation

Au lieu d'écrire :

N10 G91

N20 G1 F1000
N30 X100 Y50
Il faudrait écrire :
N10 G91 G1 X100 Y50 F1000

Communication Sercos

Chaque écriture d'un numéro d'identification (ordre WID3) demande un temps de traitement et entraîne le freinage de l'interpolation à la vitesse 0.

Usinage des surfaces de formes libres

En général, les blocs CN utilisés pour les surfaces de formes libres présentent des longueurs de course très courtes. Pour éviter que la vitesse sur trajectoire ne soit réduite en excès ici, il est recommandé d'activer la fonctionnalité « Ignorer les blocs CN » via le paramètre machine.

Les paramètres "Nombre max. de blocs pour le tampon de blocs (706000120)" et "Nombre de séquences en pourcentage pour le tampon de séquence (706000130)" indiquent le nombre maximal de blocs ignorés. La commande calcule respectivement la moyenne des vitesses maximales de chaque bloc via une fenêtre avec le nombre de blocs configuré. Si la fenêtre contient au moins 1 bloc relativement long avec une vitesse maximale supérieure, elle annule les vitesses maximales respectives des blocs voisins plus courts si aucune valeur limite n'est dépassée.

8.4.2 Mesures ayant une influence indirecte sur la vitesse de traitement

Introduction

Ces mesures agissent sur la vitesse de traitement uniquement lorsque le travail s'effectue sur la limite de temps du cycle de bloc.

Syntaxe abrégée

Chaque caractère programmé dans le programme pièce demande du temps d'exécution. Par conséquent, la syntaxe dite abrégée est optimale par rapport au format long, en ce qui concerne l'interprétation des blocs.

Exemple:

Syntaxe abrégée et format long

FFW < FeedForward

SCCT < SelCrdCoupleTab

Fonctions modales

Les fonctions modales ne doivent être programmées que si elles sont modifiées.

Dans la mesure du possible, la modification des fonctions modales doit être programmée dans un bloc CN avec l'information de déplacement suivante.

Exemple:

Fonctions modales

N10 G90

N20 1 X50 F1000

N30 G90 G1 X70 Y20 F1000

Annexe

N40 G90 G1 X100 Y50 F1000

N50 G91

N60 G1 X-100 F1000

N70 G91 G0 Y-50

Exemple:

Fonctions modales optimales

N10 G90 G1 X50 F1000

N20 X70 Y20

N30 X100 Y50

N40 G91 X-100

N50 G0 Y-50

Commentaires

- Pour gagner du temps, les passages de commentaires relativement longs doivent être ignorés avec un ordre GOTO.

```
10 GOTO .TETE
; =====
; Avant d'exécuter le programme CN, tenir impérative-
; ment compte de la documentation
;
; ...
; PRG TETE
; =====
20 .TETE
```

- Si certaines lignes doivent être considérées comme des commentaires, le mieux est de le faire en inscrivant un ";" directement au début de la ligne.

Ordre WAIT

En règle générale, les passages CPL ne doivent être séparés du passage DIN qui les précède par un ordre "WAIT" que si le passage CPL accède aux valeurs actuelles de l'interpolation (par ex. dans le cas d'une requête concernant les positions actuelles).

Si seules des valeurs programmées sont lues ou si des données sont seulement écrites au sein de la partie CPL, aucun ordre "WAIT" supplémentaire n'est requis puisque CPL est exécuté de manière synchronisée au sein du traitement du bloc.

De même, aucun ordre "WAIT" n'est requis après un passage CPL si CPL écrit des données qui doivent être traitées par le passage DIN consécutif (par ex. corrections d'outil).

Les fonctions suivantes présentent une fonctionnalité WAIT interne :

ATBWD	Calibrage des cinématiques d'axe : transformation en arrière
ATCAL	Calibrage des cinématiques d'axe : optimisation des paramètres

ATFWD	Calibrage des cinématiques d'axe : transformation en avant
ATGET	Calibrage des cinématiques d'axe : lecture des paramètres de la commande numérique
ATPUT	Calibrage des cinématiques d'axe : écriture de paramètres dans la commande numérique
DCR(0)	Désactiver le réglage en hauteur pour la numérisation
EPC(0)	Désactiver le couplage de position finale
FSM	Déplacement sur butée fixe
FSP	Mesure sur butée fixe
FSR	Restaurer le déplacement sur butée fixe
G74(HOME)	Approche du point de référence
G75, G75.1, G75.2	Palpeur de mesure
HSB	Changement de bloc avec interruption via le signal High-Speed
HWOC(0)	Désactiver la correction en ligne dans les coordonnées de la pièce à usiner
LCP	Compensation de poursuite
SCC(0)	Désactiver le couplage sélectif additif de coordonnées
SCCT(0)	Désactiver le couplage sélectif additif de coordonnées avec table
SCPOWAIT	Nouveau décalage du couplage de broches actif
SCWAIT	Attente du couplage de broches synchrone
SPC_WAIT	Attente du mode synchrone
SPCP_WAIT	Attente du décalage angulaire
TappRet1, TappRet2	Remontée d'outil du taraudage
TSP	Activer le taraudage pour des broches individuelles
VAP	Entraînements virtuels

Accès au fichier

Il est toujours plus lent d'accéder à des fichiers mounted qu'à des fichiers du système de fichiers internes de la commande. Les programmes ou cycles fréquemment appelés devraient toujours être enregistrés dans "usr/user".

Attente d'une variable CPL permanente

S'il faut attendre dans un programme qu'un certain événement se produise, les ordres "WPV" et "SPV" doivent être privilégiés par rapport à une vérification à l'intérieur d'une boucle CPL.

Exemple:

```

WPV, SPV
10 WPV[@1=1]
20 SPV[@2=3]
```

Annexe

Paramètres de la version avec "MCODS"

Si le contenu de la variable de la version est égal à 0 lors de l'appel de MCODES, alors la fonction fournit immédiatement les données demandées. Cette version est automatiquement augmentée à chaque modification des données. Si le contenu de la version $\neq$ 0 lors de l'appel de MCODES, alors la fonction attend la modification suivante des données.

Exemple:

L'appel de MCODES fournit des données immédiatement :

```
10 VERSION=0
20 REPEAT
30 ERR = MCODES(32,1,VERSION,SAVSTATE%,1)
40 UNTIL SAVSTATE%=6
```

8.4.3 Retard de la séquence dû à une programmation imprécise**Introduction**

Si la syntaxe de programmation employée est spéciale, il se peut que le programme soit ralenti. En raison du comportement temporel des fonctions programmées, il est possible qu'un événement soit déjà traité alors que le programme attend cet événement à un autre emplacement (sur un autre canal). Le programme s'arrête alors à cet emplacement.

Reprise d'une broche sur le canal en tant qu'axe

En cas de commutation "Broche vers axe" (STA) ou inversement (ATS), il est généralement recommandé de programmer un ordre WAIT sans paramètres après l'ordre en question.

Exemple:

Reprise d'une broche en tant qu'axe

```
N10 STA(C)
N20 WAIT
N30 WAX(C)
```

Programmation d'un axe rond via la définition de la vitesse

Un axe rond est programmé via la définition de la vitesse, puis il est repris sur le canal. Avant l'ordre WAX, ajouter obligatoirement un ordre WAIT.

Exemple:

Axe rond avec définition de la vitesse de rotation

```
N10 RAX(C1)
N20 C1=S(500)
N30 G4 F5
N40 C1=S(0)
N50 WAIT
N60 WAX(C1)
```

Boucles CPL

Les boucles CPL interrompues par un événement externe doivent être pourvues de l'instruction "WAIT(<temps d'attente>)". Cela permet d'éviter le blocage des tâches moins prioritaires (affichage de la position).

Exemple:

Boucle CPL

10 REPEAT

20 ...

30 WAIT(,200)

40 UNTIL SD(9)=0

9 Service et assistance

Nous disposons d'un réseau de services international dense pour vous offrir une assistance rapide et optimale. Nos experts se tiennent à votre entière disposition. Vous pouvez nous contacter **7j/7 et 24h/24, même les week-ends et jours fériés**.

Service après-vente Allemagne

Nos centre de compétences technologique à Lohr satisfait à toutes les attentes autour du service après-vente pour les entraînements et commandes électriques.

Vous pouvez joindre notre **numéro d'urgence** et notre **Bureau d'assistance** au:

Téléphone:	+49 9352 40 5060
Fax:	+49 9352 18 4941
E-mail:	service.svc@boschrexroth.de
Site web:	http://www.boschrexroth.com

Nos sites Internet comportent aussi des consignes supplémentaires relatives au service, à la réparation (adresses de livraison, par ex.) et à la formation.

Service après-vente international

En dehors de l'Allemagne, veuillez d'abord contacter votre interlocuteur. Vous trouverez les numéros des lignes d'urgence parmi les coordonnées des distributeurs sur Internet.

Préparation des informations

Nous pouvons vous aider de manière rapide et efficace si vous tenez à disposition les informations suivantes:

- Une description détaillée de la panne et des circonstances
- Les informations figurant sur la plaque signalétique des produits concernés et notamment les références et les numéros de série
- Vos coordonnées (téléphone, fax et adresse e-mail)

Index

Symboles

(ADR_PARA).....	55
<axe>=S<valeur>.....	212
\$<numéro du canal>.....	40

A

À propos de cette documentation.....	13
Présentation des informations.....	19
Validité de la documentation.....	13
AAC.....	248
AAS.....	248
Abréviations.....	20
ABS.....	91
ABSR0T2EUL.....	480
AC(...).....	178
Accélération maximale des axes	
Enregistrement temporaire.....	248
Modification.....	248
Accélération radiale maximale.....	369
Acceptation du réglage de l'axe issus des paramètres machine.....	280
Accès à l'interface numérique.....	520
Accès au fichier.....	679
accès aux opérandes.....	522
Accès aux tables XML.....	587
ACN.....	354
ACOS.....	93
ACP.....	354
ACS.....	458
Activation de la limitation des secousses.....	312
Activation des tableaux de placement.....	346
Activation des tables de correction D.....	279
Activation des tables de décalage d'origine.....	448
Activation du mode axe C pour les broches.....	411
Activer le mode axe C pour les broches et attendre.....	411
ActRadFact.....	222
ADJ.....	223
Adresse F.....	213
Adresse oméga (avance).....	215
ADSCLR.....	228
ADSDIS.....	227
ADSENA.....	227
ADSLAB.....	228
ADVA.....	229
AdvanceAppend.....	229
AdvanceSubstitute.....	229
ADVS.....	229
AFC.....	256
Affectation d'une expression STRING.....	632
Affectation d'une expression STRING à un tableau.....	632
Aide à la saisie : décalage programmé du contour.....	392
Aide à la saisie : fonction miroir.....	324

Inversion d'un vecteur d'orientation.....	326
Aide à la saisie : mise à l'échelle.....	387
Aide à la saisie : rotation.....	380
Rotation d'un vecteur d'orientation.....	381
Aides à la saisie	
SCL.....	483
AJK.....	259
AJS.....	259
ALN.....	240
AND.....	94
Angles, arrondissement	
Avec indication de la différence.....	382
Avec indication du rayon.....	383
Anticipation.....	290, 291
anwperm.dat.....	80
APOS.....	459, 460
Appel d'un sous-programme avec adresse P.....	46
Appel d'un sous-programme sans adresse P.....	47
Appel de MCODS.....	680
Appel de sous-programme autodéfini avec les codes G et M.....	51
Appel de sous-programme autodéfini modal.....	51
Appel de sous-programme dans CPL via CALL.....	53
Identification DIN dans le programme à appeler.....	53
Processus de liaison avec DIN.....	53
Approche de la position fixe des axes machine.....	174
Approche des coordonnées du point de référence.....	171
Approche du point de référence.....	172
Approximation de spline B.....	137
Longueur des paramètres.....	138
Programmation de coordonnées.....	137
Valeur de tolérance.....	138
ARA.....	231, 232
Area.....	231, 232
ARF.....	222
Arrêt conditionnel du programme.....	204
Arrêt de l'enregistrement du diagnostic.....	287
Arrêt de la broche.....	208
Arrêt du mouvement jusqu'à ce qu'une position d'axe soit atteinte.....	108
ASTOPA.....	108
ASTOPO.....	108
Arrêt du mouvement jusqu'à ce que la pièce à usiner atteigne une position.....	111
WSTOPA.....	111
WSTOPO.....	111
Arrêt du mouvement jusqu'à ce que la pièce à usiner atteigne une position de base.....	109
BSTOPA.....	110
BSTOPO.....	110
Arrêt précis ACTIVÉ.....	675

Index

Arrêt précis ACTIVÉ/DÉSACTIVÉ.....	166	Attributs de programmation.....	36
Arrêt programme.....	204	Aucun plan.....	145
Arrondissement des angles		Autoretenue, à.....	39
Avec indication de la différence.....	382	Autoretenue, sans.....	40
Avec indication du rayon.....	383	AUXFUNC.....	247
Arrondissement des angles avec des splines		Avance.....	122
Local.....	415	Avance 100 %.....	344
Modal.....	414	Avance dépendant d'un événement.....	304
Arrondissement à l'entier supérieur ou infé- rieur.....	92	Avance et programmation de la vitesse de rotation.....	212
ASC.....	629	Adresse F.....	213
ASIN.....	93	Adresse F2 F2.....	214
ASPCLR.....	236	Vitesse des axes asynchrones.....	215
ASPDIS.....	236	AVE.....	263
ASPENA.....	237	AVS.....	263
ASPRTP.....	238	AxAcc.....	248
ASPSET.....	239	AxAccSave.....	248
ASPSTA.....	239	AXADR\$.....	451, 452
Assistance.....	683	AXC.....	253
AssLogName.....	240	AxCompTab.....	250
ASTOPA.....	109	AxCouple.....	253
ASTOPO.....	109	Axe modulo, prise d'origine.....	320
AsynchrShapeOrder, ASO.....	141	Axes asynchrones	
AsynchShapeOrder.....	142	Vitesse.....	215
ATAN.....	93	Axes et coordonnées.....	449
ATAN2.....	93	AXINF.....	453, 454
ATBWD.....	241	AxisFeedCouple.....	256
ATCAL.....	241	AxisToSpindle.....	261
ATFWD.....	243	AxisToSpindleWait.....	262
ATGET.....	244	AxJerk.....	259
ATPUT.....	246	AxJerkSave.....	259
ATR.....	438	AXO.....	475
ATrans.....	438	AXP.....	456
ATS.....	261	AXR.....	262
ATSW.....	262	AxVel.....	263
Attente d'un événement bit.....	100	AxVelSave.....	263
Attente d'une variable CPL permanente.....	679		
Attente de la programmation de la broche à la fin du bloc.....	412	B	
Attente de la valeur d'une variable CPL per- manente.....	105	BAP.....	139
WPV.....	105	Base de données d'outils	
WPVE.....	105	Accès élargi.....	597
Attente des états sur l'interface bit API CN		Accès élargi avec verrouillage.....	598
WAITA.....	103	Accès en lecture ou en écriture.....	596
WAITO.....	103	Chargement.....	589
Attente des états sur l'interface bit API CN :		Déplacement du bloc de données.....	590
WAITA / WAITO.....	103	Enregistrement.....	591
Attente du mode synchrone.....	401	Recherche étendue.....	594
Attente et suppression d'un événement bit.....	102	Rechercher un bloc de données.....	592
Attribut S.....	212	BCD.....	94
Attribution d'un nom d'axe logique.....	240	BCR.....	266
Attribution de valeur.....	89	BCS2WCS.....	483
Attribution.....	89	BcsCorr.....	266
Entre variables.....	90	BIN.....	94
NUL.....	90	BITIF.....	520
Variable BOOL.....	89	BlkNmb.....	99
Variable REAL.....	89	Bloc-programme – Programme CN.....	29
		Blocs CN.....	676

BNB.....	99	CHR\$.....	630
Boucles CPL.....	681	CHS.....	268
Broche : commutation entre les modes d'as- servissement en position et en vitesse de ro- tation.....	418	ChSection.....	268
Broche esclave		CHSL.....	269
Décalage angulaire avec un couplage ac- tif.....	399	CLD.....	270
Écart de couplage.....	398	CLN.....	271, 272
Broches.....	467	CLOCK.....	488
BsaPar.....	139	CLOSE.....	612
BSTOPA.....	110	CLRWARN.....	601
BSTOPO.....	110	CNB.....	267
Bureau d'assistance.....	683	Codes de touches.....	674
C		Codes G.....	121
Calcul modulo pour des axes sans fin à dé- placement incrémental.....	329	Approche de la position fixe des axes machine.....	174
Plage modulo.....	330	Approche des coordonnées du point de référence.....	171
Calcul modulo pour les axes sans fin à dé- placement incrémental		Approche du point de référence.....	172
Pour des axes sans fin individuels du ca- nal.....	330	Arrêt précis ACTIVÉ/DÉSACTIVÉ.....	166
Pour tous les axes sans fin du canal.....	330	AsynchrShapeOrder.....	141
Calibrage des cinématiques d'axe.....	241	Aucun plan.....	145
Écriture de paramètres dans CN.....	246	Changement de plan.....	146
Lecture des paramètres de CN.....	244	Changement de plan avancé.....	147
Optimisation des paramètres.....	241	Commutation de correction.....	175
Transformation en arrière.....	241	Conditions aux limites de départ et d'arri- vée.....	138
Transformation en avant.....	243	Correction 3D du rayon d'outil.....	194
Caractères clés.....	95	Correction de l'avance.....	159
Caractères de séparation entre 2 mots parti- iels.....	37	Correction de la longueur d'outil.....	160
Désignations des axes et des coordon- nées.....	37	Correction de trajectoire de la fraise.....	155
CASE.....	72	Décalage d'origine programmable.....	162
CASE-LABEL...LABEL-OTHERWISE-ENDCA- SE.....	72	Déplacement contre le palpeur de mesu- re.....	173
Centre de la fraise – G46.....	159	Entrée tangentielle dans le cercle.....	131
CFD.....	321	Filetage.....	152
CG0.....	273	Interpolation circulaire/hélicoïdale/hélicoï- dale N.....	124
Changement de bloc au vol via le signal à grande vitesse.....	303, 304	Interpolation linéaire en avance.....	122
Changement de bloc avec interruption via le signal à grande vitesse.....	306	Interpolation linéaire en avance rapide.....	121
Interruption du mouvement de déplace- ment en fonction d'un événement.....	306	Libre sélection du plan.....	149
Changement de broche principale.....	323	Paramètre d'approximation de spline B.....	139
Changement de plan.....	146	Pente de la trajectoire ACTIVÉE.....	140
Attribution indépendante des axes.....	147	Pente de la trajectoire DÉSACTIVÉE.....	140
Changement de plan avancé.....	147	Placement : plan incliné.....	196
Attribution des noms d'axes.....	148	Profil de vitesse avec limitation des se- courses.....	141
Changement du rapport de réduction.....	406	Programmation absolue de la vitesse avec adaptation de l'accélération.....	184
ChkNxtBlk.....	267	Programmation absolue de la vitesse avec adaptation de la vitesse proportion- nelle à la course.....	180
CHL.....	268	Programmation absolue de la vitesse avec adaptation de la vitesse proportion- nelle au temps.....	181
ChLength.....	268	Programmation absolue locale.....	177
CHLL.....	269	Programmation de l'avance.....	179
		Programmation de l'avance par rotation.....	186
		Programmation de l'avance par rotation d'un axe rond.....	189

Index

Programmation de spline.....	131	Communication Sercos.....	677
Programmation de temps.....	178	Commutation de broches au mode axe C.....	411
Programmation directe de la vitesse de rotation.....	190	Commutation de correction ACTIVÉE.....	176
Programmation du tracé de contour.....	149	Commutation de correction DÉSACTIVÉE.....	176
Programmation en pouces.....	170	Commutation de l'interface de position/vitesse de rotation.....	408
Programmation incrémentale de la vitesse avec adaptation de l'accélération....	182, 187	Commutation de la broche vers l'axe.....	680
Programmation métrique.....	170	Commutation de la programmation linéaire/spline.....	319
Programmation relative.....	177	Commutation du mode de fonctionnement de broche.....	437
Programmation relative locale.....	177	Comparaison d'expressions STRING.....	629
Raccords entre segments de contour pour la correction de trajectoire de la fraise.....	158	Compensation de poursuite.....	315
Sous-programmes asynchrones : repositionnement de coordonnées individuelles....	175	Compensation des erreurs de rectitude et d'angle.....	301
Spline avec programmation de coefficients.....	132	Activation.....	301
Spline d'Hermite.....	134	Compensation volumétrique.....	439
Splines B (NURBS) (type de spline 3).....	136	Conditions additionnelles.....	30
Splines cubiques C1 et C2 constantes.....	135	Informations sur la course avec condition additionnelle.....	30
Taraudage sans mandrin de compensation.....	168	Mots de programme.....	30
Temporisation.....	130	Conditions aux limites de départ et d'arrivée....	138
Vitesse de coupe constante.....	190	ConfG0.....	273
Codes M.....	204	Configuration de l'interpolation G0.....	273
Adresse oméga (avance).....	215	Configuration de la dynamique.....	436
Arrêt de la broche.....	207	Configuration des données de remontée.....	435
Débrayer le rapport de réduction.....	211	Configuration du programme	
Définition de la vitesse de rotation.....	212	Sauts dans le déroulement du programme....	44
Définition de la vitesse modulo.....	212	Configuration du programme, éléments.....	40
Interruption conditionnelle du programme...	204	Commentaires au sein d'un bloc CPL.....	42
Interruption du programme.....	204	Commentaires dans un bloc CN standard....	41
Orientation de la broche/Positionnement de la broche.....	208	Commentaires dans un programme pièce....	41
Programmation de la vitesse de rotation de la broche.....	215	Commenter des lignes du programme.....	41
Rotation à droite de la broche.....	205	Identificateur de canal.....	40
Rotation à droite de la broche et liquide de refroidissement ACTIVÉS.....	205	Lignes vides dans le code de programme....	41
Rotation à gauche de la broche.....	206	Numéros de blocs.....	40
Rotation à gauche de la broche et liquide de refroidissement ACTIVÉS.....	206	Remarques dans l'interface utilisateur.....	43
Sélection automatique du rapport de réduction.....	210	Configuration par défaut de l'axe.....	280
Sélection manuelle du rapport de réduction.....	210	Constante à virgule flottante (DOUBLE).....	95
Terminer le programme.....	204	Constante de chaîne.....	95
COF.....	476	Constante de chaîne de caractères.....	95
COFFS.....	195	Constante en nombre entier (INT).....	95
Collision.....	271	Constante, vitesse de coupe.....	190
Commentaires.....	678	Constantes.....	95
Commentaires au sein d'un bloc CPL.....	42	Constante de chaîne.....	95
Commentaires clients.....	20	Constantes à virgule flottante (DOUBLE).....	95
Commentaires dans un bloc CN standard.....	41	Constantes de chaînes de caractères.....	95
Commentaires dans un programme pièce.....	41	Constantes en nombre entier (INT).....	95
Commenter des lignes du programme.....	41	ConstFeed.....	321
Communication avec un client.....	641	Contour, décalage programmé.....	392
		Conversion automatique de la syntaxe dans une liste de paramètres d'adresse.....	58
		Conversion de type	
		Caractères ASCII.....	630
		Code ASCII.....	629
		Conversion de caractères.....	630
		Conversion de chiffre.....	631
		Conversion entre systèmes de nombres.....	94

Format BCD.....	94	Exemple de programmation – Attendre jusqu'à ce que l'état SAV du canal 2 passe à INACTIF.....	586
Format BIN.....	94	Exemple de programmation – Attribution axe-canal.....	587
Nombres codés BCD en valeur binaire.....	94	Exemple de programmation – Canal/État d'attente.....	586
Valeur binaire au format BCD.....	94	Exemples de programmation – Interroger immédiatement l'état SAV et l'état de l'interpolateur du canal 2.....	585
Coord.....	275	Valeurs de retour pour une erreur.....	583
Coordonnée de la pièce à usiner		Couplage sélectif additif de coordonnées.....	389
Déplacement en Jog.....	312	Avec table.....	390
CornerLineDetection.....	270	SelCrdCouple, SCC.....	389
Correction D.....	217	Couplage sélectif additif de coordonnées avec table.....	390
Correction d'outil.....	217	Couplage sélectif additif de coordonnées SelCrdCouple, SCC.....	389
Correction D.....	217	CoupleSplineTab.....	277
Correction ED "ED".....	218	CRD	275
Correction de l'avance.....	159	Création de broches de canal.....	302
Centre de la fraise G46.....	159	Critiques.....	20
Point de pénétration de la fraise G45.....	159	CST.....	277
Correction de la longueur d'outil.....	160	CurveParamInterpolation.....	279
Correction de la position de la pièce à usiner.....	476		
Correction de la position de la pièce à usiner et placements (plan incliné).....	476	D	218
Calcul des angles d'Euler.....	479, 480, 481	D.....	489
Conversion d'un point.....	482, 483	DATE.....	280
Correction de la position de la pièce à usiner.....	476	DAX.....	589
Tableau de placement.....	478	DBLOAD.....	590
Correction de position de la pièce à usiner et placements (plan incliné)		DBMOVE.....	591
Valeurs de placement actives.....	477	DBSAVE.....	592, 593
Correction de trajectoire de la fraise.....	155	DBSEA.....	594
Raccords entre segments de contour.....	158	DBSEAX.....	597
Correction du plan sous-tendu.....	265	DBTAB.....	597
Correction du rayon d'outil 3-D.....	194	DBTABX.....	598
Correction ED.....	218	DBTABXL.....	354
Correction en ligne dans les coordonnées de la pièce à usiner.....	307	DC.....	286
Corrections d'outil.....	484	DCB.....	286
Valeurs de correction d'outil actives.....	484	DcBreak.....	286
Corrections de l'outil		DCC.....	286
Tables et registres de correction de l'outil...	485	DcCont.....	195
COS.....	93	DCOFFS.....	285
Couplage d'avance d'axes.....	256	DCR.....	279
Couplage d'axes.....	252	DCS.....	485, 486
Décalage constant.....	252	DCT.....	279
Facteur de couplage constant.....	252	DcTsel.....	211
Couplage de broche		Débrayer le rapport de réduction.....	399
Décalage angulaire avec un couplage actif.....	399	Décalage angulaire avec un couplage actif.....	400
Couplage de broches		Décalage angulaire, attente d'un.....	162
Attente d'un décalage angulaire.....	400	Décalage d'origine programmable.....	476
Attente du mode synchrone.....	401	Décalage de position et de contour	
Écart de coulage de la broche esclave.....	398	Décalage de contour.....	475
Fenêtre du mode synchrone.....	399, 401	Décalage de position.....	475
Couplage de position finale.....	287	Décalage de position et décalage de coordonnée.....	475
Couplage NCS		Décalage programmé du contour.....	392
Exemple de programmation – Affichage des noms des axes dans la fenêtre MSG....	586		

Index

Décalages d'origine.....	164, 469
Accès en lecture et en écriture à une table DO.....	472
Ajout d'un axe dans une table DO.....	470
Génération d'une table DO.....	469
Suppression d'un axe d'une table DO.....	471
Valeurs DO actives.....	475
Déclenchement d'un événement bit.....	102
Déclenchement de course anticipé (point d'arrivée de l'interpolation)	
Déclenchement de course anticipé.....	360
Déclenchement de course retardé (point d'arrivée de l'interpolation)	
Déclenchement de course retardé.....	360
Déclenchement de l'enregistrement du diagnostic.....	287
DefAxis.....	280
Définition (activation) d'un groupe de couplage.....	396
Définition de la vitesse de rotation.....	212
Positionnement volant.....	212
Définition de la vitesse modulo.....	212
Définition des paramètres de l'approximation spline B.....	264
Définition du comportement de redémarrage...	375
Définition du point d'inversion miroir/de rotation.....	351
Définition du point de redémarrage dans un SP asynchrone.....	376
Définition du pointeur de fichier	
SEEK et fichier aléatoire.....	620
SEEK et fichier séquentiel.....	620
Définition du pôle.....	347
Définition du TCS dans les coordonnées de programme.....	433
Définition du type de spline.....	415
Définition/Dissociation de groupes de broches.....	402
DefSpindle.....	281
DefTangTrans.....	281
Déplacement contre le palpeur de mesure.....	173
Déplacement dans le TCS.....	424
Déplacement en butée fixe.....	298
Déplacement modulo pour les axes sans fin...	330
Déplacement sur butée fixe	
Déplacement sur butée fixe.....	298
Réduction du couple en butée fixe.....	298
Suppression de la butée fixe.....	298
DERDelay.....	283
Désactivation de la gestion de mouvement des broches pour certains canaux.....	406
Détection automatique des coins et des droites.....	269
Spline cubique constante c2.....	270
Détermination de la syntaxe de broche pour le programme de comparaison de l'avance de bloc.....	395
Détermination de la taille d'un fichier	
FILESIZE et fichier aléatoire.....	626
FILESIZE et fichier séquentiel.....	626
Détermination des axes système pour la définition du filtre SHAPE optimal avec une limitation des secousses active.....	310
DIA.....	282
Diagnostic du guidage de vitesse.....	441
DiaProg.....	282
DIM.....	627
Dimension du plan de travail.....	284
DIRCR.....	608, 609
DIRDEL.....	609
Directe, programmation de la vitesse de rotation.....	190
DIRINF.....	606, 607
Dissociation (désactivation) d'un groupe de couplage.....	397
DistCtrl.....	285
DN.....	133
DO.....	164
Données de la commande.....	488
Base de données d'outils.....	589
Données système de types structurés.....	510
Données système simple.....	492
Erreurs et avertissements.....	599
Fonctions actives.....	523
Fonctions auxiliaires modales actives.....	523
Interface API.....	520
Interface Sercos.....	514
Paramètres machine.....	491
Saisie du temps.....	488
Tables XML générales.....	587
Version de la commande.....	488
Données et processus de la commande	
Interface CN(NCS).....	524
Données système de types structurés	
Accès.....	510
Chargement des données système structurées.....	512
Informations relatives aux variables.....	511
Sauvegarde des données système structurées.....	513
Valeur de variables.....	511
Données système simples	
SD.....	492
SDR.....	506
Données système, types structurés	
Règles d'attribution pour les types de base.....	511
DPC.....	477
DREC_START.....	287
DREC_STOP.....	287
Droits d'accès aux fichiers.....	23
DSP.....	281
DTT.....	281

E

EBC.....	138
Écart de couplage de la broche esclave.....	398
Écriture d'un fichier	
Ordre PRN# avec point-virgule.....	615
Ordre PRN# sans point-virgule.....	616
Écriture d'une date système lors de l'exécution.....	112
SSD.....	113
SSDE.....	113
Écriture d'une variable CPL permanente.....	107
SPV.....	107
SPVE.....	107
Écriture dans une queue de données système.....	403
LastFilled.....	403
LastGet.....	403
Écriture de numéros d'identification élargis d'une liste ID dans un entraînement	
Writeld3, WID3.....	447
Écriture de paramètres dans un entraînement	
Axes synchrones (Writeld3, WID3).....	446
Broches et axes asynchrones (Writeld3, WID3).....	446
Écriture de paramètres Sercos	
Numéros d'identification élargis d'une liste ID.....	447
Pour un ou plusieurs entraînements (axes synchrones).....	446
Pour un ou plusieurs entraînements (broches et axes asynchrones).....	446
Valeurs d'une liste de paramètres.....	447
Writeld3, WID3.....	445
WSCS.....	517
Écriture des valeurs d'une liste de paramètres dans un entraînement	
Writeld3, WID3.....	447
ED.....	219
Éléments CPL.....	25
Éléments de base CPL	
Caractères clés.....	95
Éléments de base CPL	
Constantes.....	95
Enchaînement d'expressions STRING.....	633
Enchaînement au sein d'ordres CPL.....	633
Enchaînement de textes via les expressions STRING.....	634
Ordre MID\$ avec accès en écriture.....	636
Ordre MID\$ avec accès en lecture.....	636
Enchaînement de blocs sans réduction de la vitesse.....	281
Enchaînements de blocs continus.....	676
EndPosCouple.....	287
Enregistrement du vecteur de retrait.....	369
Entraînements virtuels.....	441
Entrée tangentielle dans le cercle.....	131
EOF.....	623

EPC.....	287
ERASE.....	613
Erreur de poursuite.....	166
Erreur et catégories d'erreur "GETERR"	
Exploitation du résultat pour 5 éléments de tableau.....	604
Interroger la dernière erreur du programme pièce dans le canal 2.....	603
Erreurs et avertissements	
Avertissement quant à la durée d'exécution.....	600
Erreur d'exécution.....	599
Requête.....	602
Suppression des avertissements.....	601
Exécution des fonctions auxiliaires actives de tous les groupes.....	247
Exécution du bloc	
Éléments CPL.....	26
Parties de langage DIN-CPL.....	26
Exemple d'application LeapFrog	
Exemple d'application LeapFrog.....	59
Exemples de programmation – Suppression des espaces de tête et de queue	
Écriture d'un tableau.....	640
Lecture d'un tableau.....	640
Programmation de variables STRING (sans dimensionnement préalable).....	639
Exemples de programmation – Suppression des espaces de tête ou de queue	
Écrasement d'un tableau.....	641
Extension du bloc-programme arrière Retrace.....	378
Extension du programme avant Retrace.....	229
F	
F.....	213
FA.....	215
FAD.....	289
FDG.....	293
FDGT.....	294
Feed-back.....	20
FeedAd.....	289
FeedForward.....	290
FeedGrp.....	292
FeedGrpType.....	294
Fenêtre d'arrêt précis, présélection.....	309
Fenêtre du mode synchrone.....	399, 401
Fenêtre INPOS avec arrêt précis ACTIVÉ	
G61.....	675
Fenêtre INPOS avec G0.....	676
FFW.....	290
FFWA.....	291
Fichiers et répertoires.....	605
Noms de fichiers.....	605
Opérations de fichier.....	609
Propriétés du fichier.....	624
Répertoires.....	606

Index

Structure de fichier aléatoire.....	605	Activation du mode axe C pour les broches.....	411
Structure de fichier séquentielle.....	606	Activer le mode axe C pour les broches et attendre.....	411
Traitement d'un fichier.....	614	Aide à la saisie : fonction miroir.....	324
FILEACCESS.....	624	Aide à la saisie : mise à l'échelle.....	387
FILECOPY.....	614	Aide à la saisie : rotation.....	380
FILEDATE.....	624, 625	Anticipation.....	290
FILENO.....	609	Anticipation des axes asynchrones.....	291
FILEPOS.....	621, 622	Approximation spline B.....	264
FILESIZE.....	625, 626	Arrondissage des angles.....	382
Filetage.....	152	Arrondissage des angles avec des splines.....	414
Filet enchaîné.....	154	Attente d'un décalage angulaire.....	400
Filet longitudinal.....	153	Attente d'une programmation de la broche à la fin du bloc.....	412
Filet multiple.....	154	Attente du mode synchrone.....	401
Filet plan.....	153	Attribution d'un nom d'axe logique.....	240
Filet plan conique.....	154	Avance 100 %.....	344
Filetage, fonctions additionnelles.....	435	Broche : commutation de l'interface de position/vitesse de rotation.....	418
Fin de programme.....	38	Calcul modulo pour des axes sans fin à déplacement incrémental.....	329
Fin de sous-programme.....	38	Calibrage des cinématiques d'axe.....	241
Fin du programme.....	204	Changement de bloc au vol via le signal à grande vitesse.....	303
FlyMeas.....	295	Changement de bloc avec interruption via le signal à grande vitesse.....	306
FME.....	295	Changement de broche principale.....	323
Fonction de synchronisation lors de l'exécution du bloc.....	100	Chargement du tableau de correction des erreurs d'axe.....	250
Fonction exponentielle.....	91	Commutation de la programmation linéaire/spline.....	319
Fonction miroir.....	324	Comparaison de fonctions CN modales.....	223
Fonctions actives		Comparaison des broches dans le programme de comparaison d'initialisation de l'avance de bloc.....	394
NCF.....	523	Comparaison des configurations d'axe TargetAxisConf, TAX.....	432
Fonctions additionnelles pour le filetage.....	435	Compensation de poursuite.....	315
Commutation du mode de fonctionnement de broche.....	437	Compensation des erreurs de rectitude et d'angle.....	301
Configuration de la dynamique.....	436	Compensation volumétrique.....	439
Configuration des données de remontée.....	435	Configuration de l'interpolation G0.....	273
Influence sur le signal IF du canal.....	437	Correction en ligne dans les coordonnées de la pièce à usiner.....	307
Libération de la remontée d'outil.....	436	Couplage d'avance d'axes.....	256
Fonctions CN – Notions de base.....	23	Couplage d'axes.....	252
Fonctions CN – Vue d'ensemble.....	643	Couplage de position finale.....	287
Fonctions CN avec la syntaxe de langage standard		Couplage sélectif additif de coordonnées.....	389
Remplacement du programme avant Retrace.....	229	Création d'une broche de canal.....	302
Fonctions CN avec syntaxe conforme à DIN 66025		Décalage additif des coordonnées du programme.....	437
Avance et programmation de la vitesse de rotation.....	212	Décalage angulaire avec un couplage actif.....	399
Codes M.....	204	Décalage des coordonnées du programme.....	437
Correction d'outil.....	217	Décalage programmé du contour.....	392
Fonctions CN avec syntaxe conforme à la norme DIN 66025			
Codes G.....	121		
Fonctions CN avec syntaxe conforme DIN 66025.....	119		
Fonctions CN avec syntaxe de langage standard			
Accélération radiale maximale.....	369		
Acceptation des réglages d'axe issus des paramètres machine.....	280		
Activation de la limitation des secousses.....	312		
Activation des tableaux de placement.....	346		
Activation des tables de correction D.....	279		
Activation des tables de décalage d'origine.....	448		

Définition (activation) d'un groupe de couplage.....	396	Initialisation de la position du programme	392
Définition du point d'inversion miroir/de rotation.....	351	Interpolation des paramètres des courbes.....	279
Définition du TCS dans les coordonnées de programme.....	433	Libération/Reprise d'une broche réservée.....	396
Définition du type de spline.....	415	Limitation de la vitesse de rotation.....	393
Définition/Dissociation de groupes de broches.....	402	Limitations de la broche spécifiques à l'outil.....	410, 423
Démarrage de l'enregistrement du diagnostic.....	287	Mesure en butée fixe.....	296
Déplacement dans le TCS.....	424	Mesure sur un contour.....	326
Déplacement modulo pour les axes sans fin.....	330	Mesures au vol.....	294
Déplacement sur butée fixe.....	298	Mise hors service du mode axe C pour les broches.....	261
Détection automatique des coins et des droites.....	269	Mise hors service du mode axe C pour les broches et attente.....	261
Détermination de la syntaxe de broche pour le programme de comparaison de l'avance de bloc.....	395	Mode de positionnement pour les axes sans fin.....	354
Détermination des axes système pour la définition du filtre SHAPE optimal avec une limitation des secousses active.....	310	Mode Jog dans les coordonnées de la pièce à usiner.....	312
Diagnostic du guidage de vitesse.....	441	Modification de l'accélération maximale des axes.....	248
Dimension du plan de travail.....	284	Modification de l'accélération maximale sur trajectoire.....	345
Dissociation (désactivation) d'un groupe de couplage.....	396	Modification de la secousse maximale des axes.....	259
Écart de couplage de la broche esclave.....	398	Modification de la vitesse maximale des axes.....	263
Écriture dans une queue de données système.....	403	Modification du palpeur de mesure de la configuration des flancs.....	331
Écriture de paramètres Sercos.....	445	Mouvement de déplacement PTP.....	364
Enchaînement de blocs sans réduction de la vitesse.....	281	Neutralisation des types de fonctionnement à étape unique.....	420
Enregistrement du vecteur de retrait.....	369	Orientation tangentielle de l'outil.....	428
Enregistrement temporaire de l'accélération maximale des axes.....	248	Paramétrage de l'orientation statique de l'outil.....	419
Enregistrement temporaire de la secousse maximale des axes.....	259	Placement : correction de la position de la pièce à usiner.....	265
Enregistrement temporaire de la vitesse maximale des axes.....	263	Poinçonnage.....	366
Entraînements virtuels.....	441	Pointes d'angles intérieurs.....	275
Exécution des fonctions auxiliaires actives de tous les groupes.....	247	Poursuite du programme avec avance de bloc après interruption du programme.....	371
Extension du bloc-programme arrière Retrace.....	378	Présélection de la fenêtre d'arrêt précis.....	309
Extension du programme avant Retrace.....	229	Prise d'origine de l'axe modulo (axe sans fin linéaire).....	320
Fenêtre du mode synchrone.....	399, 401	Profils de vitesse (fonction rampe).....	321
Fin de l'enregistrement du diagnostic.....	287	Programmation de chanfreins.....	267
Fonctions additionnelles pour le filetage.....	435	Programmation de l'orientation.....	334
Génération de l'avance : groupe d'avance.....	292	Programmation de précision.....	356
Génération de l'avance : masquage d'axes.....	289	Programmation des broches du système....	405
Génération de l'avance : type de groupe d'avance.....	294	Programmation des coordonnées polaires.....	347
Grignotage.....	332	Programmation du diamètre.....	282
Guidage tangentiel de l'outil.....	424	Programmation du rayon.....	282
Initialisation de la mesure au vol.....	308	Programmation du temps inversé.....	309
		Programmation KV.....	313
		Réaction temporisée en cas d'erreurs d'entraînement.....	283

Index

Recul Retrace, RETRACE.....	377	Arrêt du mouvement jusqu'à ce que la pièce à usiner atteigne une position de base.....	109
Réduction de couple.....	370	Fonctions de synchronisation lors de l'exé- cution d'un bloc	
Réglage de la puissance du laser en fonction de la vitesse sur trajectoire.....	317	Arrêt du mouvement jusqu'à ce que la pièce à usiner atteigne une position.....	111
Réglage en hauteur pour la numérisation....	285	Attente de la valeur d'une variable CPL permanente.....	105
Réinitialisation des broches du canal aux paramètres machine.....	281	Attente des état sur l'interface bit API CN....	103
Remontée d'outil du taraudage.....	430	Attente et suppression d'un événement bit.....	102
Remplacement d'axes.....	262	Déclenchement d'un événement bit.....	102
Remplacement du programme arrière Retrace.....	379	Écriture d'une variable CPL permanente....	107
Répartition du bloc de déplacement : lon- gueur du segment de parcours.....	315	Suppression d'un événement bit.....	103
Répartition du bloc de déplacement : nombre de segments de parcours.....	334	Suppression des conditions d'arrêt.....	112
Répartition du parcours programmable.....	417	Suppression des conditions d'arrêt à la fin du programme (M30).....	112
Répétition de bloc.....	375	Fonctions de synchronisation lors de l'exé- cution du bloc	
Reprise d'axe.....	301	Attente d'un événement bit.....	100
Reprise d'axe et attente si nécessaire.....	444	Fonctions disponibles, couplage NS	
Retrait de broches du canal.....	374	MCODS, vue d'ensemble des fonctions.....	526
Sélection d'une broche pour le taraudage sans mandrin de compensation.....	431	Fonctions mathématiques simples.....	90
Sélection d'une stratégie d'attente	413	Fonctions modales.....	677
Sélection d'une transformation d'axe.....	274	FOR-STEP-TO-NEXT.....	74
Signal programmable selon la distance.....	226	FSM.....	299
Sortie à grande vitesse dépendant de la position.....	352	FsMove.....	299
Sous-programmes asynchrones.....	236, 375	FSP.....	296
Suppression d'un axe d'un groupe d'axes.....	373	FsProbe.....	296
Suppression d'un nom d'axe logique.....	373	FSR.....	299
Suppression de la répartition cinématique des blocs CN.....	313	FsReset.....	299
Suppression de la surveillance de colli- sion.....	391	FST.....	299
Surveillance de collision.....	271	FsTorque.....	299
Surveillance de zone.....	230	G	
Surveillance du bloc suivant en cas de définition du bloc CN en mémoire tam- pon	267	G0.....	121
Synchronisation du couplage des axes système.....	384	G1.....	122, 123
Table de couplage spline.....	277	G2, trajectoire circulaire en sens antitrigono- métrique.....	124
Temps de déclenchement de la course.....	360	G3, trajectoire circulaire en sens trigonomé- trique.....	124
Traitement du programme en fonction de la préparation des blocs.....	358	G4.....	130
Fonctions de synchronisation de la prépara- tion des blocs.....	96	G5.....	131
BikNmb (BNB).....	99	G6.....	132
WAIT (<temps d'attente>).....	97	G8	141
WAIT (BITIF(...)).....	98	G8(SHAPE...).....	143
WAIT (sans paramètres).....	96	G9.....	141
Fonctions de synchronisation en cours d'exécution d'un bloc		G9(ASHAPE...).....	143
Arrêt du mouvement jusqu'à ce qu'une position d'axe soit atteinte.....	108	G9(SHAPE...).....	143
		G9(X... , Y... , ...).	141
		G16.....	146
		G17.....	147
		G17(...).....	148
		G18.....	147
		G18(...).....	148
		G19.....	147

G19(...)	148	G141	195
G20	149	G142	195
G33	152	G151	200
G40	156	G152	201
G41	157	G153	200
G42	157	G153 à G159	202
G43	159	GAX	301
G43 – Arc de cercle	158	GCT	301
G44	159	GCT(0)	301
G44 – Point d'intersection	158	GCT(1)	301
G45	160	Génération de l'avance : groupe d'avance	292
G46	160	Génération de l'avance : masquage d'axes	289
G47	161	Génération de l'avance : type de groupe d'avance	294
G48	161	Générer un nombre entier positif aléatoire	91
G52	162	GeoComp	301
G53. à G59	164	GetAxis	301
G53.1	165	GETERR	602
G53.5	165	GetSpindle	302
G54.1	165	GMSG	43
G54.1 à G59.1	165	GOA	65
G54.5	165	GoAhead	65
G55.1	165	GOB	66
G55.5	165	GoBack	66
G56.1	165	GOC	67
G56.5	165	GoCond	67
G57.1	165	GoTo	68
G57.5	165	GOTO	69
G58.1	165	Grignotage	332
G58.5	165	Groupes cibles	13
G59.1	165	Groupes de variables	77
G59.5	165	Variables globales	77
G61	167, 676	Variables locales	77
G62	167, 675	Variables permanentes	78
G63	169	Variables permanentes définissables	78
G63.2	169	Variables structurées	81
G70	170	Groupes modaux de fonctions – Aperçu	115
G71	171	GSP	302
G74	171	Guidage tangentiel de l'outil	424
G74(HOME)	172	Angle d'attaque	425
G75	173	Position du guidage de l'outil	424
G76	174		
G77	175		
G78	176	H	
G79	176	HSB	304
G90	178	HSB(...,HSSTOP=..)	306
G91	178	HsBlkSwitch	304
G93	179	HsBlkSwitch(...,HSSTOP=..)	306
G94	179	HWOC	307
G94 (LINS)	180	HWOCDIS	307
G94 (LINT)	182		
G94 (valeur F/valeur S)	183	I	
G94 (valeur TF/valeur TS)	185	IC(...)	178
G95	187	Identificateur de canal	40
G95(FRA=<axe>)	189	IF-THEN-ELSE-ENDIF	73
G95(valeur F/valeur S)	187	IME	308
G96	190	INCH	170
G97	190	Influence sur le signal IF du canal	437
G140	195		

Index

Information sur le mouvement.....	23	Fonction de base.....	123
Informations relatives au système.....	87	Paramètres optionnels.....	123
Initialisation de la mesure au vol.....	308	Interpolation linéaire en avance rapide	
Initialisation de la position du programme.....	392	Paramètres optionnels.....	121
InitMeas.....	308	Interpolation linéaire en avance rapide "G0".....	121
INP#.....	617, 618	Interruption conditionnelle du programme.....	204
INSDEP.....	195	Interruption du programme.....	204
INSTR.....	627, 628	InvTime.....	309
Instruction CPL : CASE-LABEL...LABEL- OTHERWISE-ENDCASE.....	72	IPS1.....	310
Instruction CPL : FOR-STEP-TO-NEXT.....	74	IPS2.....	310
Instruction CPL : IF-THEN-ELSE-ENDIF.....	73	IPS3.....	310
Instruction CPL : REPEAT-UNTIL.....	75	ITIM.....	309
Instruction CPL : WHILE-DO-END.....	75	J	
Instructions – Programme CN.....	29	JAL.....	311
Fonctions de parcours.....	30	JerkAxList.....	310
Fonctions de parcours avec définition de coordonnées.....	30	JerkControl.....	312
Instructions CPL.....	89	Jeu de caractères ASCII.....	671
Attributions de valeurs.....	89	JKC.....	312
Conversion entre systèmes de nombres.....	94	JogWCSSelect.....	312
Liaisons logiques.....	93	K	
Opérations de comparaison.....	94	KinSplitDisable.....	313
Opérations mathématiques.....	90	KVP.....	314
Instructions de décision et de branchement.....	71	KvProg.....	313, 314
Instruction CPL : CASE-LABEL...LABEL- OTHERWISE-ENDCASE.....	72	L	
Instructions CPL : IF-THEN-ELSE-ENDIF.....	73	Labels dans des blocs CPL.....	65
Sauter séquence.....	71	Labels dans les blocs CN standards.....	65
Instructions de répétition CPL.....	74	LagCompensation.....	315
Instruction CPL : WHILE-DO-END.....	75	LCP.....	315
Instructions CPL : FOR-STEP-TO-NEXT.....	74	Lecture d'un fichier	
Instructions CPL : REPEAT-UNTIL.....	75	Instruction INP#.....	618
Instructions de saut.....	64	Lecture d'un enregistrement à partir d'un fichier.....	618
INT.....	91	Lecture de la position du pointeur de fichier	
Interface API.....	520	FILEPOS et fichier aléatoire.....	623
Accès aux opérandes.....	522	FILEPOS et fichier séquentiel.....	622
Interface bit.....	520	Lecture des valeurs mesurées.....	464
Interface CN (NCS)		Valeur mesurée dans les coordonnées d'axe.....	465
Exemples de programmation.....	585	Valeur mesurée dans les coordonnées du programme.....	464
Services de données.....	524	Valeur mesurée de l'entraînement.....	466
Services de processus.....	570	LEN.....	316, 628
Interface Sercos		LFC.....	318
Écriture de paramètres Sercos.....	517	LFConf.....	318
Lecture des listes de paramètres Sercos.....	516	LFP.....	318
Lecture des paramètres Sercos.....	514	Liaison avec une identification DIN/CPL.....	29
Interpolation circulaire.....	125	Liaison de programmes CN.....	27
Interpolation circulaire/hélicoïdale/hélicoïda- le N.....	124	Liaison avec une identification DIN/CPL.....	29
Paramètres optionnels.....	128	Liaison de sous-programmes.....	28
Programmation du centre.....	127	Liaison en arrière-plan.....	27
Programmation du rayon.....	126	Table de liaison.....	27
Interpolation des paramètres des courbes.....	279	Liaison de sous-programmes.....	28
Interpolation G0, configuration.....	273	Liaison ultérieure.....	28
Interpolation hélicoïdale.....	125	Liaison ultérieure.....	28
Interpolation hélicoïdale N.....	125		
Interpolation linéaire.....	122		
Interpolation linéaire en avance.....	122		

Liaisons logiques.....	93
Fonction ET AND.....	94
Fonction NON NOT.....	94
Fonction OU exclusif XOR.....	94
Fonction OU OR.....	94
Libération de la remontée d'outil.....	436
Libération/Reprise d'une broche réservée.....	396, 406
Libre sélection du plan (indépendamment du WCS).....	149
Ligne directe.....	683
Lignes vides dans le code de programme.....	41
Limitation de la vitesse de rotation.....	393
Limitation de la vitesse de rotation maximale.....	407
Limitation de la vitesse de rotation minimale.....	408
LinDownFeed.....	322
LinearToSpline.....	319
LinModZp.....	320
LinUpFeed.....	321
Liste de paramètres d'adresse	
Utilisation de noms de variables identi-ques.....	55
Utilisation de plusieurs paramètres.....	56
Valeurs de paramétrage admissibles.....	55
Liste des paramètres CPL.....	57
Liste des paramètres d'adresse.....	54
Adresses de paramètres admissibles.....	55
Noms d'adresses, unicité.....	54
LJUST.....	617
LJUSTi.....	617
LMZ.....	320
LND.....	322
LNU.....	321
Local.....	40
Logiques, liaisons.....	93
Longueur des paramètres de la spline.....	136, 137
LTS.....	319

M

M0.....	204
M00.....	204
M1.....	204
M01.....	204
M2.....	205
M02.....	205
M3.....	206
M4.....	207
M5.....	208
M13.....	206
M14.....	207
M19.....	208
M30.....	205
M40.....	210
M41...44.....	211
M48.....	211
M103.....	206
M104.....	207

M105.....	208
M113.....	206
M114.....	207
M119.....	208
M140.....	210
M141...144.....	211
M148.....	211
M203.....	206
M204.....	207
M205.....	208
M213.....	206
M214.....	207
M219.....	208
M240.....	210
M241...244.....	211
M248.....	211
machdef.dat.....	81
Macros.....	62
Macros avec des paramètres.....	63
MainSp.....	324
MCA.....	491
MCASE\$.....	638
MCODS.....	524, 525
Angle de rotation.....	564
Arrêt optionnel.....	551
Attribution d'axe : canal.....	542
Attribution par défaut de l'axe : canal.....	547
Avance sur trajectoire.....	531
Avance sur trajectoire active.....	568
Avance sur trajectoire programmée.....	535
Avances sur trajectoire programmées.....	568
Blocage d'avance.....	545
Broche du canal : attribution de groupes.....	550
Broches : commutation des rapports de réduction.....	551
Chemin et nom du tableau de placement actif.....	562
Compensation actuelle de l'erreur de pas de la broche.....	567
Compensation actuelle de la température... ..	567
Compensation de croix actuelle.....	567
Coordonnées d'axe dans l'ACS0.....	561
Coordonnées d'axe dans l'ACS1.....	552
Coordonnées de base de la pointe de l'outil (BCS-Tcp).....	557
Coordonnées de décalage actives.....	565
Coordonnées de la pièce à usiner.....	552
Coordonnées de machine.....	553
Correction d'outil active.....	546
Correction de la position de la pièce à usiner.....	562
Correction du rayon active.....	533
Correction en ligne dans le WCS.....	556
Couple de broche.....	561
Course restante des axes dans l'ACS.....	555
Course restante des coordonnées dans le WCS.....	555

Index

Décalage additif des coordonnées du programme actif.....	563	Position réelle de la broche.....	561
Décalage de compensation actuel.....	568	Position réelle des axes de la CNC.....	569
Décalage des coordonnées du programme actif.....	563	Potentiomètre d'avance.....	532
Données de décalage d'origine actives.....	558	Potentiomètre d'axe.....	545
Données de placement actives.....	562	Potentiomètre de la broche.....	532
Données spécifiques au client.....	542	Poursuite.....	531
État de la correction en ligne dans le WCS.....	557	Programmation du diamètre.....	557
État INPOS pour les coordonnées.....	554	Programme principal : nom et chemin.....	537
États d'attente du canal.....	555	Puissance de la broche.....	561
Exemple de programmation.....	570	Rapport de réduction de la broche.....	550
Facteurs de mise à l'échelle actifs.....	565	Référence des coordonnées.....	554
Fonction auxiliaire avec acquittement obligatoire active.....	541	Remarques (messages).....	536
Fonction de mouvement de la broche.....	549	Remise à zéro du canal.....	566
InPos.....	532	Resélection automatique du programme.....	552
Instruction de déplacement.....	543	Sauter séquence.....	551
Inversion miroir des coordonnées active.....	565	Sélection auto/manuelle des rapports de réduction.....	551
Lecture du numéro d'identification issu du télégramme d'axe cyclique.....	549	Somme des décalages d'origine.....	559
Mode de fonctionnement de l'axe.....	544	Somme des décalages des coordonnées du programme actifs.....	563
Mode de fonctionnement du canal.....	537	Somme des placements.....	563
Mode de fonctionnement IPO.....	538	Statut de la comparaison de l'avance de bloc.....	566
Mode de fonctionnement SAV.....	538	Statut de la marque de réentrée de l'avance de bloc.....	566
Mode test.....	537	Stratégie de redémarrage.....	543
Nom de la table DO.....	533	Syntaxe active des groupes de fonctions auxiliaires.....	558
Nom du tableau de la correction d'outil.....	533	Système en pouces/métrique.....	534
Nombre d'axes.....	542	Temporisation active.....	541
Nombre de broches.....	559	Type de moteur.....	547
Nombre de canaux.....	542	Type de variateur.....	547
Nombre de coordonnées.....	555	Unité de mesure des axes.....	545
Nombre des coordonnées de machine.....	560	Unité de mesure des coordonnées.....	554
Noms actifs des axes du canal.....	547	Unités de mesure des axes (réglage par défaut).....	549
Noms d'axes.....	539	Valeur de crête et valeur maximale de l'accélération de l'axe.....	569
Noms des axes du canal (réglage par défaut).....	548	Valeur de crête et valeur maximale de la vitesse de l'axe.....	568
Noms des broches.....	559	Valeurs de consigne des coordonnées de base.....	552
Noms des coordonnées.....	553	Valeurs de coordonnées programmées.....	540
Noms des coordonnées de machine.....	560	Valeurs DO actives.....	533
Numéro d'axe de la broche.....	560	Valeurs réelles de l'axe dans le WCS.....	541
Numéro de la correction de rayon.....	532	Valeurs réelles des coordonnées de base.....	553
Numéro du système de la broche.....	560	Validation d'enregistrement.....	541
Numéros des groupes de fonctions auxiliaires.....	557	Version du fabricant.....	546
Point de référence approché.....	536	Vitesse actuelle de l'axe.....	564
Points d'arrivée des coordonnées.....	553	Vitesse de rotation actuelle de l'axe.....	564
Points d'arrivée des coordonnées d'axe.....	559	Vitesse de rotation d'une liste de broches... ..	540
Points d'arrivée programmés des coordonnées.....	553	Vitesse de rotation de la broche.....	531
Position de consigne de l'axe.....	530	Vitesse de rotation maximale de la broche.....	534
Position de consigne de l'axe (INTER).....	530	Vitesse de rotation minimale de la broche.....	535
Position finale programmée.....	535	Vitesse de rotation programmée de l'axe....	564
Position finale, y compris les aides à la saisie.....	534		
Position réelle de l'axe de l'entraînement....	540		

Vitesse de rotation programmée de la broche.....	536	Mots de programme – Effet.....	39
Vitesse maximale de l'axe.....	566	Modal.....	39
Vitesses de Jog.....	536	Non modal.....	40
MCOPS.....	570	Mots de programme en tant que paramètres.....	35
Définition du mode de fonctionnement.....	579	Paramètres suivis d'une liste de paramètres.....	35
Désélection du programme.....	578	Paramètres suivis d'une valeur.....	35
Effacement de la course restante.....	571	Mots de programme provenant de fonctions CN.....	32
Lancement du programme.....	578	Expressions CPL dans les listes de paramètres.....	33
Modification de l'enregistrement de redémarrage.....	582	Paramètres d'une liste de paramètres.....	33
Modification de la stratégie de redémarrage.....	582	Mots de programme provenant des fonctions CN	
Rechercher séquence.....	572	Listes de paramètres avec des éléments de syntaxe spécifiques.....	34
Redémarrage du canal.....	582	Listes de paramètres sans éléments de syntaxe spécifiques.....	33
Remise à zéro.....	571	Mouvement d'orientation linéaire avec programmation des axes.....	335
Sélection de la saisie manuelle.....	573	Mouvement d'orientation linéaire avec programmation des coordonnées.....	336
Sélection du programme.....	573	Mouvement de déplacement linéaire dans le système de coordonnées de l'outil (TCS).....	424
MCS.....	460	Mouvement de déplacement PTP.....	364
Mesure au vol.....	294	MPEDGE.....	331
Initialisation.....	308	MSG.....	43
Mesure en butée fixe.....	296	MSP.....	324
MID\$.....	637, 638		
MIR.....	324, 326	N	
Mirror.....	324	NCF.....	523
Mise à l'échelle.....	387	Neutralisation de l'étape unique.....	420
Mise hors service du mode axe C et attente.....	261	NIB.....	332
Mise hors service du mode axe C pour les broches.....	261	Nibble.....	332
MMC.....	641, 642	NJUST.....	617
MOC.....	327	Noms de fichiers.....	605
MOD.....	330	Noms des variables.....	76
Modal – Mots de programme.....	39	Non modal – Mots de programme.....	40
À autoretenue.....	39	Local.....	40
Mode de positionnement pour les axes sans fin.....	354	Sans autoretenue.....	40
Mode Jog dans les coordonnées pièces à usiner.....	312	NOT.....	94
Modification de l'accélération maximale sur trajectoire.....	345	Notions clés.....	87
Modification des chaînes de caractères		Notions de base du traitement des chaînes de caractères	
Affectation d'une expression STRING.....	632	Comparaison d'expressions STRING.....	629
Conversion entre majuscules et minuscules.....	638	Dimensionnement.....	627
Enchaînement d'expressions STRING.....	633	Longueur d'une chaîne de caractères.....	628
Lecture des caractères d'une chaîne de caractères.....	637	Recherche dans une chaîne de caractères.....	627
Suppression des espaces.....	637	NUL.....	90
Modulo.....	330	NUM.....	334
Mots d'ordre réservés.....	87	Numérisation	
Notions clés.....	87	Réglage en hauteur.....	285
Mots de programme.....	31	Numéro d'urgence.....	683
Attributs de programmation.....	36	Numéro du canal.....	40
Caractères de séparation entre 2 mots partiels.....	37	Numéros de blocs.....	40
En tant que paramètres.....	35	Blocs CN standards.....	41
Mots partiels.....	31	Blocs CPL.....	41
provenant de fonctions CN.....	32		

Index

NURBS..... 136

O

O()..... 344

OFFSTOPA..... 112

OFFSTOPO..... 112

Oméga..... 215

OPENR..... 610

OPENW..... 610

Opérations de comparaison..... 94

Opérations de fichier

Copie..... 614

Écrasement..... 612

Fermeture..... 612

Numéro de fichier logique..... 609

Ouverture..... 610

Suppression..... 613

Opérations mathématiques..... 90

ABS..... 91

ACOS..... 93

Arrondissement à l'entier supérieur ou inférieur..... 92

ASIN..... 93

ATAN..... 93

ATAN2..... 93

COS..... 93

EXP..... 91

Fonction exponentielle..... 91

Fonction trigonométrique..... 92, 93

Fonctions simples..... 90

INT..... 91

Racine carrée de la valeur entrée..... 92

RANDOM..... 91

ROUND..... 92

SIN..... 93

SQRT..... 92

TAN..... 93

Troncature des chiffres après la virgule..... 91

Valeur absolue d'un mot d'entrée..... 91

Optimisation du temps d'usinage..... 675

OR..... 94

Ordre CALL..... 53

Ordre GOTO..... 678

Ordre SHAPE résultant..... 144

Ordre SPV..... 679

Ordre WAIT..... 678

Ordre WID3..... 677

Ordre WPV..... 679

Ordres CPL..... 449, 475

Adresse d'axe variable..... 456

Adresses d'axes et de coordonnées..... 451

Adresses de broches..... 467

Aides à la saisie..... 483

Axes et coordonnées..... 449

Broches..... 467

Communication avec un client..... 641

Correction de la position de la pièce à

usiner et placements (plan incliné)..... 476

Corrections d'outil..... 484

Décalages d'origine..... 469

Données et processus de la commande..... 488

Fichiers et répertoires..... 605

Informations sur l'axe..... 453

Lecture des valeurs mesurées..... 464

Positions de coordonnées et d'axes..... 457

Traitement des chaînes de caractères..... 627

Orientation de la broche..... 409

Orientation de la broche/Positionnement de

la broche..... 208

Orientation du tenseur..... 338

O(...)...... 339

phi, theta, psi..... 339

ROTAX(...)...... 339

Vecteurs colonnes (Ox, Oy, Oz)...... 339

Orientation tangentielle de l'outil..... 428

Orientation vectorielle..... 342

Coordonnées d'orientation..... 342

O(...)...... 342

phi, theta, psi..... 342

ROTAX(...)...... 343

Ouverture d'un fichier

OPENR..... 610

OPENW..... 610

OVD..... 345

OVE..... 345

OvrDis..... 345

OvrEna..... 345

P

P..... 47

PAC..... 346

Palpeur de mesure, déplacement contre le..... 173

Paramétrage de l'orientation statique de

l'outil..... 419

Paramètre d'approximation de spline B..... 139

Paramètres de l'approximation spline B..... 264

Paramètres de la version avec MCODS..... 680

Paramètres machine

MCA..... 491

Paramètres optionnels

Interpolation circulaire/hélicoïdale/hélicoï-

dale N..... 128

Interpolation linéaire en avance..... 123

Interpolation linéaire en avance rapide..... 121

Parcours programmable, répartition..... 417

Parties de langage CPL..... 26

Parties de programme CPL..... 25

Parties du langage DIN..... 26

Pas de filetage..... 169

PathAcc..... 345

PCS..... 461

PCSPROBE..... 464

PDIM..... 58

Pente de la trajectoire ACTIVÉE.....	140	PrecProg.....	357
Pente de la trajectoire DÉSACTIVÉE.....	140	PREP.....	359
Pente de la trajectoire G8 avec arrêt précis		Présélection de la fenêtre d'arrêt précis.....	309
DÉSACTIVÉ G62.....	675	Présentation des informations	
Pente de la trajectoire G9 avec arrêt précis		Consignes de sécurité.....	19
DÉSACTIVÉ G62.....	675	Désignations et abréviations.....	20
PHS.....	352	Symboles utilisés.....	19
PL.....	134, 135, 136, 137	Prise d'origine de l'axe modulo (axe sans fin	
Placement : correction de la position de la		linéaire).....	320
pièce à usiner.....	197, 265	PRN#.....	615
Chaîne de calcul.....	265	PROBE.....	466, 467
Rotation de base autour de la coordon-		Processus de la commande.....	488
née Z.....	265	Profil de vitesse avec limitation de secousses	
Placement : plan incliné.....	196	SHAPE pour axes asynchrones.....	142
Activation implicite dans le programme		Profil de vitesse avec limitation des secous-	
pièce.....	198	ses.....	141
Configuration après activation du plan in-		Ordre SHAPE résultant.....	144
cliné perpendiculaire à l'outil.....	200	Rapport ordre SHAPE-secousses.....	145
Plan incliné perpendiculaire à l'outil.....	199	SHAPE pour mode de positionnement.....	143
Programmation associée à un tableau de		SHAPE pour mode trajectoire.....	143
placement.....	201	Profils de vitesse	
Programmation directe sur la ligne du		Définition propre.....	321
programme en tant que paramètre.....	201	Profils de vitesse (fonction rampe).....	321
Plan incliné.....	196	Cycle d'oscillation de l'axe U.....	322
Plan, sélection libre.....	149	Interpolateur de marche constante, effet.....	321
PLC.....	522	Interpolateurs d'accélération.....	321
PLS.....	351	Interpolateurs de freinage.....	322
PMD.....	354	Programmation	
PMS.....	346	SHAPE pour mode de positionnement.....	143
PMT.....	478	SHAPE pour mode trajectoire.....	143
PmTSel.....	346	Programmation absolue.....	177
PMV.....	477	Programmation absolue locale.....	178
Poids ponctuel de la spline pour point de		Programmation CN.....	24
contrôle des splines B.....	137	Programmation CPL.....	24
Poinçonnage.....	366	Programmation d'axes.....	132, 134
Point de pénétration de la fraise – G45	159	Programmation d'un axe rond via la défini-	
Pointes d'angles intérieurs.....	275	tion de la vitesse.....	680
Points d'interprétation.....	26	Programmation d'un polynôme dénomina-	
PolarPol.....	347	teur.....	133
PoleSet.....	351	Programmation d'un vecteur d'orientation.....	133
POP.....	347	Programmation de chanfreins.....	267
PosDepHSOut.....	352	Chanfrein entre deux droites sécantes.....	267
Position fixe des axes machine, approche		Chanfrein entre deux segments de cer-	
de la.....	174	cles sécants.....	268
Positions de coordonnées et d'axes		Longueur de chanfrein.....	267
Lecture de la position de consigne d'un		Longueur du segment de chanfrein.....	267
axe du système.....	462	Programmation de coordonnées.....	132, 134, 135,
Lecture de la position de la pièce à usi-			136
ner.....	463	Programmation de l'avance (par min).....	179
Lecture des coordonnées de machine.....	460	Programmation de l'avance par rotation.....	186
Lecture des coordonnées du programme.....	461	Programmation de l'avance par rotation d'un	
Lecture des valeurs réelles de l'axe.....	459	axe rond.....	189
Position de consigne de l'axe.....	458	Programmation de l'orientation.....	334
PosMode.....	354	Mouvement d'orientation linéaire avec	
Possibilités de programmation DIN et CPL.....	24	programmation des axes.....	335
Poursuite du programme avec avance de		Mouvement d'orientation linéaire avec	
bloc après interruption du programme.....	371	programmation des coordonnées.....	336
PPOS.....	465, 466	Orientation du tenseur.....	338

Index

Orientation vectorielle.....	342	Dans un bloc CPL (DIN).....	65
Programmation de la longueur des paramètres de la spline.....	134, 135	Programmation label et instructions de saut.....	64
Programmation de la vitesse avec adaptation de l'accélération		Labels dans les blocs standards et CPL.....	65
Absolue.....	184	Saut conditionnel vers un bloc CN standardGoCond (GOC).....	67
Incrémentale.....	182, 187	Saut CPL (GOTO) vers un bloc-programme quelconque.....	69
Programmation de la vitesse de rotation.....	409	Saut en arrière dans un sous-programme vers un bloc-programme quelconque	
Programmation de la vitesse de rotation de la broche.....	215	RetTo.....	69
Programmation de macros.....	62	Saut en arrière vers un bloc CN standardGoBack (GOB).....	66
Avec des paramètres.....	63	Saut en avant vers un bloc CN standardGoAhead (GOA).....	65
Programmation de mouvement.....	408	Saut inconditionnel vers un bloc CN standard GoTo.....	68
Programmation de précision.....	356	Programmation locale en pouces.....	170
Programmation de spline.....	131	Programmation métrique.....	170
Programmation de temps.....	178	Programmation relative.....	177
Programmation de variables.....	76	Programmation relative locale.....	178
Groupes de variables.....	77	Programmation standard.....	24
Noms des variables.....	76	Programme CN	
Types de variables.....	82	Bloc-programme.....	29
Variable CPL ERRNO pour l'analyse des erreurs.....	85	Conditions additionnelles.....	30
Variables en programmation CN standard.....	84	Éléments de base.....	29
Programmation des broches du système.....	405	Instructions.....	29
Programmation de la vitesse de rotation.....	409	Propriétés du fichier	
Programmation des broches du système SSp...		Date.....	624
Changement du rapport de réduction.....	406	Droits d'accès.....	624
Commutation de l'interface de position/vitesse de rotation.....	408	Taille du fichier.....	625
Désactivation de la gestion de mouvement des broches pour certains canaux.....	406	PRP.....	357
Libération/Reprise d'une broche réservée.....	406	PT2EUL.....	481
Limitation de la vitesse de rotation maximale.....	407	PtBlkEnd.....	360
Limitation de la vitesse de rotation minimale.....	408	PTD.....	362
Orientation de la broche.....	409	PtDefault.....	362
Programmation de mouvement.....	408	PTE.....	360
Programmation des coordonnées polaires.....	348	PTI.....	363
Angle polaire 1.....	348	PtInpos.....	363
Angle polaire 2.....	349	PTP.....	364
Définition du pôle.....	347	PUN.....	367
En valeur absolue.....	349	Punch.....	367
Programmation incrémentale.....	350	PW.....	137
Programmer des coordonnées polaires.....	348		
Valeur du rayon.....	348	R	
Programmation DIN.....	24	RAC.....	369
Programmation directe de la vitesse de rotation.....	190	Raccords entre segments de contour pour la correction de trajectoire de la fraise.....	158
Programmation du centre.....	127	G43 – Arc de cercle.....	158
Programmation du diamètre.....	282	G44 – Point d'intersection.....	158
Programmation du rayon.....	126, 282	Racine carrée de la valeur entrée.....	92
Programmation du tracé de contour.....	149	RAD.....	282
Programmation en pouces.....	170	RadialAcc.....	369
Programmation KV.....	313	RadProg.....	282
Programmation label		RANDOM.....	92
Dans un bloc CN standard (DIN).....	65	RAX.....	373

Réaction temporisée en cas d'erreurs d'en- traînement.....	283	Programmation d'un axe rond via la défi- nition de la vitesse.....	680
Réclamations.....	20	Reprise d'une broche sur le canal en tant qu'axe.....	680
RecordRevVec.....	370	Retrace.....	377
Recul Retrace.....	377	Retrait de broches du canal.....	374
RedTorque.....	370	RetTo.....	70
Réduction de couple.....	370	REV.....	103
ReentContBlk.....	371	REVA.....	379
Référentiel LCS.....	190	ReverseAppend.....	379
Référentiel MCS.....	190	ReverseSubstitute.....	379
Référentiel PCS.....	191	REVS.....	379
Réglage de la puissance du laser		REWRITE.....	612
En fonction de la vitesse sur trajectoire.....	317	RLN.....	373
Réglage de la puissance du laser en fonc- tion de la vitesse sur trajectoire.....	317	RND.....	383
Réglage en hauteur pour la numérisation.....	285	RNDL.....	383
Réinitialisation des broches du canal aux paramètres machine.....	281	RNE.....	382
REM.....	42	ROT(...).....	380
Remarques dans l'interface utilisateur.....	43	ROT2EUL.....	479, 480
Multicanaux.....	43	Rotate(...).....	380
Spécifiques au canal.....	43	Rotation.....	380
RemAxis.....	373	Rotation à droite de la broche.....	205
RemLogName.....	373	Rotation à droite de la broche et liquide de refroidissement ACTIVÉS.....	206
Remontée d'outil du taraudage.....	430	Rotation à gauche de la broche.....	206
Remontée d'outil après la panne de cou- rant.....	430	Rotation à gauche de la broche et liquide de refroidissement ACTIVÉS.....	207
Remontée d'outil après une remise à zé- ro.....	430	Rotation d'un contour.....	380
Remplacement du programme arrière Retra- ce.....	379	ROTAX().....	344
Remplacement du programme avant Retra- ce.....	229	ROUND.....	92
RemSpindle.....	374	RoundEps.....	382
Répartition du bloc de déplacement : lon- gueur du segment de parcours.....	315	Rounding.....	383
Répartition du bloc de déplacement : nom- bre de segments de parcours.....	334	RRV.....	370
Répartition du parcours programmable.....	417	RSP.....	374
REPEAT.....	375		
REPEAT-UNTIL.....	75	S	
Répertoires		S.....	216
Création.....	608	S2D.....	322
Informations sur le répertoire.....	606	S2U.....	321
Suppression.....	609	SACSYNC.....	384
REPOSDEF.....	375	Saisie du temps	
Repositionnement de coordonnées indivi- duelles.....	175	Date.....	489
REPOSTP.....	376	Heure.....	490
Reprise d'axe.....	301	Temps système.....	488
Reprise d'axe et attente si nécessaire.....	444	Saut conditionnel vers un bloc CN standard.....	67
Reprise d'une broche sur le canal en tant qu'axe.....	680	Saut CPL vers un bloc-programme quel- conque.....	69
Réservés, mots d'ordre.....	87	Saut en arrière dans un sous-programme vers un bloc-programme quelconque.....	69
Retard de la séquence, programmation im- précise.....	680	Saut en arrière vers un bloc CN standardGo- Back (GOB).....	66
Boucles CPL.....	681	Saut en avant vers un bloc CN standard.....	65
		Saut inconditionnel vers un bloc CN standard....	68
		Sauter séquence pour les blocs CN stan- dards.....	71
		Sauts dans le déroulement du programme.....	44
		Appels de sous-programmes.....	44

Index

Instructions de décision/de branchement.....	44	Override.....	492
Instructions de répétition.....	45	Override de la broche.....	501
Instructions de sauts.....	44	Palpeur de mesure.....	493
SBC.....	138	Point de redémarrage.....	500
Scale(...).	387	Programmation de précision.....	504
SCC.....	389	Sens de rotation de la broche.....	495
SCCT.....	390	Signification des coordonnées.....	496
SCL.....	483, 484	Type de traitement.....	495
SCL(...).	387	Vitesse.....	492
SCO.....	415	Vitesse de rotation.....	492
SCOL.....	415	Vitesse de rotation des broches du canal....	501
SCS3.....	514	Zones de travail et zones mortes.....	500
SCS3L.....	516	SDLOAD.....	512, 513
SD.....	492	SDR.....	506
Avance de bloc : position cible.....	506	Couplage d'axes du canal.....	509
Canal actuel.....	493	Décalage additif des coordonnées du	
Canal d'un axe du système.....	498	programme (ATrans).....	509
Configuration des plans – Axes du canal		Décalage des coordonnées du program-	
affectés.....	494	me (Trans + ATrans).....	507
Correction de la longueur – Coordonnées		Décalage des coordonnées du program-	
affectées.....	494	me (Trans).....	508
Couplage d'axes du canal.....	504	Données de la spline.....	507
Décalage additif des coordonnées du		Guidage tangentiel de l'outil.....	508
programme (ATrans).....	503	Override.....	506
Décalage des coordonnées du program-		Override de la broche.....	508
me (Trans + ATrans).....	500	Position d'orientation d'une broche du ca-	
Décalage des coordonnées du program-		nal.....	509
me (Trans).....	500	Position de redémarrage après ASUP.....	507
Données de la spline.....	499	Programmation de précision.....	509
Essai sans mouvement.....	496	Vitesse.....	506
Événement déclencheur du programme		Vitesse de rotation d'une broche du ca-	
d'initialisation.....	499	nal.....	508
Exemple de programmation.....	505	SDSAVE.....	514
Fonction de mouvement des broches du		Secousse maximale des axes	
canal.....	502	Enregistrement temporaire.....	259
Groupe d'avance.....	504	Modification.....	259
Guidage tangentiel de l'outil.....	500	Secousses.....	145
Langue nationale.....	495	SEEK.....	619, 620
Nombre d'axes sur le canal.....	496	SelCrdCouple.....	389
Nombre d'axes sur le canal actuel.....	496	SelCrdCoupleTab.....	390
Numéro d'axe du canal d'un axe du sys-		Sélection automatique du rapport de réduc-	
tème.....	497	tion.....	210
Numéro d'axe du système d'un axe du		Sélection d'une broche pour le taraudage	
canal.....	498	sans mandrin de compensation.....	431
Numéro d'axe du système d'un axe du		Sélection d'une stratégie d'attente.....	413
canal sur le canal actuel.....	497	Sélection d'une transformation d'axe.....	274
Numéro d'axe du système d'une broche		Points d'action.....	274
du canal.....	503	Sélection du rapport de réduction	
Numéro d'axe du système d'une broche		Automatique.....	210
du système.....	503	Manuelle.....	210
Numéro de broche du canal d'une broche		Sélection manuelle du rapport de réduction....	210
du système.....	502	SETERR.....	599, 600
Numéro de broche du système d'un axe		SetPos.....	392
du système.....	503	SETWARN.....	600
Numéro de broche du système d'une		SEV.....	103
broche du canal.....	502	SHAPE pour axes asynchrones.....	142
Numéros des broches du canal d'un		Shift.....	393
groupe de broches.....	501	SHT.....	393

Signal à grande vitesse, changement de bloc au vol.....	303	SpCoupleConfig.....	397
Signal à grande vitesse, changement de bloc avec interruption.....	306	SpCoupleDist.....	398
SIN.....	93	SpCoupleErrWin.....	399
Sin2DownFeed.....	322	SpCouplePosOffs.....	399
Sin2UpFeed.....	321	SpCouplePosOffs_Wait.....	400
SinDownFeed.....	322	SpCoupleSyncWin.....	401
SinUpFeed.....	321	SPCP.....	399
SMax.....	394	SPCP_WAIT.....	400
SMD.....	421	SPCS.....	401
SME.....	422	SPD.....	416
SMin.....	393	SPG.....	402
SMN.....	394	SPGALL.....	402
SMX.....	394	SpindleToAxis.....	411
SND.....	322	SpindleToAxisWait.....	412
SNU.....	321	SpindleWait.....	412
Sortie à grande vitesse		SpindleWaitMode.....	413
Dépendant de la position.....	352	Spline	
Sortie à grande vitesse dépendant de la position.....	352	Cubique C1 constante.....	135
Sous-programmes.....	45	Cubique C2 constante.....	135
Appel autodéfini avec les codes G et M.....	51	Spline avec programmation de coefficients.....	132
Appel autodéfini modal.....	51	Programmation d'axes.....	132
Appel avec adresse P.....	46	Programmation d'un polynôme dénominateur.....	133
Appel dans CPL via l'ordre CALL.....	53	Programmation d'un vecteur d'orientation... ..	133
Appel sans adresse P.....	47	Programmation de coordonnées.....	132
Effet local.....	46	Programmation de la longueur des paramètres de la spline.....	134
Effet modal.....	46	Spline B (NURBS).....	136
Élimination des erreurs.....	61	Longueur des paramètres de la spline.....	137
Profondeur d'imbrication.....	61	Poids ponctuel de la spline pour point de contrôle.....	137
Sous-programmes locaux.....	48	Programmation de coordonnées.....	136
Transmission de paramètres aux sous-programmes.....	54	Spline d'Hermite.....	134
Sous-programmes asynchrones		Programmation d'axes.....	134
Déclenchement par un programme.....	239	Programmation de coordonnées.....	134
Définition d'un point de redémarrage.....	238	Programmation de la longueur des paramètres de la spline.....	135
Définition du comportement de redémarrage.....	375	SplineCornering.....	414
Définition du point de redémarrage dans un SP asynchrone.....	376	SplineCorneringLocal.....	415
Désinscription.....	236	SplineDef.....	416
Inscription.....	238	Splines cubiques C1 et C2 constantes.....	135
Mise en service.....	237	Conditions aux limites de départ et d'arrivée.....	136
Mise hors service.....	236	Longueur des paramètres de la spline.....	136
Sous-programmes asynchrones : repositionnement de coordonnées individuelles.....	175	Programmation de coordonnées.....	135
Sous-programmes locaux.....	48	Split.....	417
Sous-programmes modaux.....	58	SPM.....	418
SPA.....	396	SpMode.....	418
SpAdjStr.....	395	SPOS.....	462, 463
SpAdmin.....	396	SPS.....	392
SPADR\$.....	467, 468	SPT.....	417
SPC_WAIT.....	401	SPV.....	107
SPCC.....	397	SPVE.....	107
SPCD.....	398	SPWAIT.....	412
SPCE.....	399	SQRT.....	92
SpCouple_Wait.....	401	SSD.....	113
		SSDE.....	113
		SSDQ.....	403

ThreadSet(...)	435
ThreadSet(DYN)	436
ThreadSet(ROn)	436
ThreadSet(SPC)	437
ThreadSet(TCI)	437
Traitement d'un fichier	
Conversion de type	629
Écriture	614
Traitement de fichier	
Définition de la position du pointeur de fichier	621
Formatage	617
Identification de la fin du fichier	623
Lecture	617
Positionnement du pointeur de fichier	619
Traitement des chaînes de caractères	
Modification des chaînes de caractères	632
Notions de base	627
Trans	438
Transformation en arrière	241
Transformation en avant	243
Transmission de paramètres aux sous-programmes	54
Liste des paramètres CPL	57
Transmission des paramètres à des sous-programmes	
Conversion dans une liste de paramètres d'adresse	58
Transmission des paramètres à des sous-programmes modaux	58
Transmission des paramètres à des sous-programmes modaux	58
Transmission des paramètres aux sous-programmes	
Liste des paramètres d'adresse	54
TRIM\$	637
Troncature des chiffres après la virgule	91
TRS	438
TSP	431
TST	435
TTL	425
TTO	428
Type BC	139
Type de spline 0	132
Type de spline 1	134
Type de spline 3	136
Type de spline 4	137
Types de splines	132, 415
Types de splines 1 et 2	135
Types de variables	82
Variable à virgule flottante (DOUBLE)	82
Variable à virgule flottante (REAL)	82
Variable de tableau (ARRAY)	83
Variable en nombre entier (INT)	82
Variable logique (BOOL)	82
Variables CHAR	83
Vue d'ensemble	84

U

Unité de mesure	
Métrique	170
Pouces	170
Unités de mesure	
Positions d'axes et de coordonnées four-nies	450
Usinage des surfaces de formes libres	677
Utilisation conforme	
Cas de mise en œuvre	22
Domaines d'application	22
Introduction	21
Utilisation non conforme	22
Conséquences, exclusion de responsabi-lité	21

V

Valeur absolue du mot d'entrée	91
Valeurs d'axes et de coordonnées	
Unités de mesure	450
VAP	441
Variable à virgule flottante (DOUBLE)	82
Variable à virgule flottante (REAL)	82
Variable ARRAY	83
Variable BOOL	82
Variable CHAR	83
Variable CPL ERRNO	85
Variable de tableau (ARRAY)	83
Variable DOUBLE	82
Variable en nombre entier (INT)	82
Variable ERRNO	85
variable INTEGER	82
Variable logique (BOOL)	82
Variable REAL	82
Variable STRING	84
Variables en programmation CN standard	84
Variables globales	77
Variables locales	77
Variables permanentes	78
Variables permanentes définissables	78
Variables structurées	81
VARINF	511
VARVAL	511, 512
VCPEna	439
VCPLoad	440
VEC2EUL	481, 482
VERSINF\$	488
Version de la commande	
VERSINF\$	488
VirtAxisPos	441
Virtuels, entraînements	441
Vitesse de coupe constante	190
Vitesse de rotation, limitation	393
Vitesse de traitement, influence directe	675
Blocs CN	676
Communication Sercos	677
Enchaînements de blocs continus	676

Index

Fenêtre INPOS avec arrêt précis ACTI-VÉ.....	675	ZOT.....	472, 473
Fenêtre INPOS avec G0.....	676	ZOTCR.....	469
Pente de la trajectoire G8/G9 avec arrêt précis DÉACTIVÉ G62.....	675	ZoTSel.....	448
Usinage des surfaces de formes libres.....	677	ZOV.....	475
Vitesse de traitement, influence indirecte.....	677		
Accès au fichier.....	679		
Attente d'une variable CPL permanente.....	679		
Commentaires.....	678		
Fonctions modales.....	677		
Ordre WAIT.....	678		
Paramètres de la version avec MCODES.....	680		
Syntaxe abrégée.....	677		
Vitesse des axes asynchrones.....	215		
Vitesse maximale des axes			
Enregistrement temporaire.....	263		
Modification.....	263		
Vol, mesure au.....	294		
Volumétrie, compensation.....	439		
VREC_START.....	441		
VREC_STOP.....	444		
W			
WAIT	96		
WAIT (sans paramètres).....	97		
WAIT avec temps d'attente.....	97		
WAIT(<temps d'attente>).....	97		
WAIT(BITIF(...)).....	98		
WAITA.....	104		
WaitAxis.....	445		
WAITO.....	104		
WAX.....	445		
WCS.....	463		
WCS2BCS.....	482		
WEV.....	101		
WHILE-DO-END.....	75		
WID3.....	446, 447		
wmhperm.dat.....	80		
WPV.....	105		
WPVE.....	105		
WREV.....	102		
WritelD3.....	446		
WritelD3.....	446, 447		
WSCS.....	518		
WSTOPA.....	111		
WSTOPO.....	111		
X			
XOR.....	94		
XTAB.....	587		
XTABCR.....	588		
Z			
ZOCDEL.....	471		
ZOCINS.....	470		
ZOS.....	448		

Notes

Bosch Rexroth AG

Electric Drives and Controls

P.O. Box 13 57

97803 Lohr, Germany

Bgm.-Dr.-Nebel-Str. 2

97816 Lohr, Germany

Phone +49 9352 18 0

Fax +49 9352 18 8400

www.boschrexroth.com/electrics



R911381559