

MTX 14VRS

Funktionsbeschreibung - Erweiterung

Referenz
R911342627

Ausgabe 08



Titel	MTX 14VRS Funktionsbeschreibung - Erweiterung
Art der Dokumentation	Referenz
Dokumentations-Type	DOK-MTX***-NC*F*EX*V14-RE08-DE-P
Interner Ablagevermerk	RS-c8d9a1014d820ef80a347ea5007e2c65-7-de-DE-6
Änderungsverlauf	Ausgabe 08, 2021-04 Siehe Tab. 1-1 "Änderungsverlauf" auf Seite 1
Schutzvermerk	© Bosch Rexroth AG 2021 Alle Rechte vorbehalten, auch bzgl. jeder Verfügung, Verwertung, Reproduktion, Bearbeitung, Weitergabe sowie für den Fall von Schutzrechtsanmeldungen.
Verbindlichkeit	Die angegebenen Daten dienen allein der Produktbeschreibung und sind nicht als zugesicherte Eigenschaften im Rechtssinne zu verstehen. Änderungen im Inhalt der Dokumentation und Liefermöglichkeiten der Produkte sind vorbehalten.
Redaktion	Entwicklung Software und Steuerungs-Firmware - Steuerungsfunktionen CNC, HaBu (MiSc)

Inhaltsverzeichnis

	Seite
1 Zu dieser Dokumentation.....	1
1.1 Gültigkeit der Dokumentation.....	1
1.2 Allgemeines.....	2
1.3 Erforderliche und ergänzende Dokumentationen MTX.....	3
1.3.1 Auswählen/Zusammenstellen.....	3
1.3.2 Konfigurieren.....	4
1.3.3 In Betrieb nehmen.....	4
1.3.4 Bedienen.....	5
1.3.5 OEM-Entwicklung.....	6
1.3.6 AddOns.....	7
1.4 Darstellung von Informationen.....	7
1.4.1 Gebrauch der Sicherheitshinweise.....	7
1.4.2 Bezeichnungen und Abkürzungen.....	9
1.5 Kundenfeedback.....	9
2 Wichtige Gebrauchshinweise.....	11
2.1 Bestimmungsgemäßer Gebrauch.....	11
2.1.1 Einführung.....	11
2.1.2 Einsatz- und Anwendungsbereiche.....	12
2.2 Nicht-bestimmungsgemäßer Gebrauch.....	12
3 Antriebe (Achsen, Spindeln).....	13
3.1 Lineare Moduloachse	13
3.1.1 Beschreibung.....	13
3.1.2 Handlungsanweisung: Appliziert eine Linearachse als Moduloachse.....	14
3.2 Hirthachse.....	15
3.2.1 Beschreibung.....	15
3.2.2 Handlungsanweisung: Appliziert eine Hirthachse.....	16
4 Bedienfunktionen (manueller Betrieb).....	19
4.1 Tippen in Werkstückkoordinaten	19
4.1.1 Beschreibung.....	19
4.1.2 Handlungsanweisung: Tippen in Werkstückkoordinaten.....	20
4.2 Handrad in Werkstückkoordinaten.....	23
4.2.1 Allgemeines.....	23
4.2.2 Handlungsanweisung: Handrad in Werkstückkoordinaten.....	27
5 Kanalsteuerung.....	31
5.1 Weg- und Wiederanfahren.....	31
5.1.1 Beschreibung.....	31
5.1.2 Optionen zum Wiederanfahren.....	34
5.1.3 Handlungsanweisung: Weg- und Wiederanfahren an die Kontur.....	37
5.2 Satzvorlauf.....	40

	Seite
5.2.1	Allgemeines zum Satzvorlauf..... 40
5.2.2	Beschreibung..... 43
5.2.3	Vorgehensweise beim Satzvorlauf..... 44
5.2.4	Projektierung des Satzvorlaufs..... 48
5.2.5	Umschaltung von modalen Zuständen im Abgleichprogramm oder Aktionssatz..... 59
5.2.6	Anwendungen zum Satzvorlauf..... 62
5.2.7	Spindeln..... 64
5.2.8	Systemdaten..... 65
5.2.9	Interface-Signale..... 68
5.2.10	Look Ahead..... 68
5.2.11	Allgemein..... 68
5.2.12	Beispiele..... 70
5.3	Satzwechsel über High-Speed..... 76
5.3.1	Beschreibung..... 76
5.3.2	Handlungsanweisung: Satzwechsel über High-Speed-Signal mit dem Löschen des Restwegs..... 80
5.3.3	Handlungsanweisung: Satzwechsel über High-Speed-Signal mit fliegendem Wechsel..... 81
5.4	Programmfortführung mit Satzvorlauf nach Programmabbruch..... 83
5.4.1	Beschreibung..... 83
5.4.2	Aktivieren..... 84
5.4.3	Deaktivieren..... 84
5.4.4	Handlungsanweisungen..... 84
6	Dynamik und Geschwindigkeitsführung..... 85
6.1	Skalieren von Rundachsgeschwindigkeiten..... 85
6.1.1	Konfiguration von Skalierungsfaktoren für Rundachsgeschwindigkeiten..... 85
6.1.2	Programmierung von Skalierungsfaktoren für Rundachsgeschwindigkeiten..... 88
6.2	Bahnlängen- und Kurvenparameterinterpolation..... 90
6.2.1	Beschreibung..... 90
7	Sicherheit / Überwachung..... 93
7.1	Momenten-Reduktion..... 93
7.1.1	Beschreibung..... 93
7.1.2	Handlungsanweisung: Momenten-Reduktion aktivieren/deaktivieren..... 94
7.2	Reduzierter Eilgang..... 94
7.2.1	Beschreibung..... 94
7.2.2	Handlungsanweisung: Reduzierter Eilgang..... 95
7.3	Test-Vorschub, Test-Eilgang 96
7.3.1	Beschreibung..... 96
7.3.2	Handlungsanweisung: Test-Vorschub (DryRun)..... 98
7.3.3	Handlungsanweisung: Test-Eilgang (DryRun)..... 99
7.4	1-dimensionale Achskollisionsüberwachung 101
7.4.1	Beschreibung..... 101
7.4.2	Handlungsanweisung: 1-dimensionale Achskollisionsüberwachung..... 103
7.5	Achspositionsüberwachung..... 106
7.5.1	Beschreibung..... 106

	Seite
7.5.2 Handlungsanweisung: Achspositionsüberwachung.....	107
7.6 Distanz vor Endpunkt.....	107
7.6.1 Beschreibung.....	107
7.6.2 Handlungsanweisung: Distanz vor Endpunkt.....	108
7.7 Rückzugsvektor als Systemdatum.....	109
7.7.1 Beschreibung.....	109
7.7.2 Handlungsanweisung: Rückzugsvektor als Systemdatum.....	112
8 Genauigkeit.....	115
8.1 Genauigkeitsprogrammierung.....	115
8.1.1 Beschreibung.....	115
8.1.2 Handlungsanweisung: Genauigkeitsprogrammierung.....	117
9 Bahnbewegung.....	119
9.1 Spline.....	119
9.1.1 Überblick.....	119
9.1.2 Spline mit Koeffizientenprogrammierung	123
9.1.3 Spline mit Stützpunktprogrammierung und tangentialen Übergängen (C1-stetige kubische Splines)	127
9.1.4 Kubische Hermite-Splines.....	133
9.1.5 Spline mit Stützpunktprogrammierung und krümmungsstetigen Übergängen (C2-stetige kubische Splines)	135
9.1.6 Spline mit Kontrollpunktprogrammierung (B-Spline)	140
9.1.7 Approximierende B-Splines mit Toleranzangabe.....	146
9.1.8 Spline programmieren	148
9.1.9 Handlungsanweisung: Spline applizieren.....	153
9.1.10 Handlungsanweisung: Spline aktivieren.....	154
9.1.11 Handlungsanweisung: Spline deaktivieren.....	154
9.2 Spline-Zusatzfunktionen.....	155
9.2.1 Automatische Ecken- und Geradenerkennung.....	155
9.3 Automatische Linear-Zirkular-Spline-Wandlung.....	157
9.3.1 Beschreibung.....	157
9.3.2 Applizieren.....	159
9.3.3 Aktivieren.....	160
9.3.4 Deaktivieren.....	160
10 Werkzeugorientierung und Orientierungsbewegung	161
10.1 Allgemeines	161
10.1.1 Beschreibung.....	161
10.2 Definitionen zur Orientierung.....	161
10.2.1 Werkzeugkoordinatensystem TCS.....	161
10.2.2 Eulerwinkel	162
10.2.3 Orientierungsvektor.....	166
10.2.4 Orientierungstensor.....	168
10.3 Orientierungsbewegung.....	169

Inhaltsverzeichnis

	Seite
10.3.1	Beschreibung..... 169
10.3.2	Vektororientierungsbewegung 170
10.3.3	Lineare Orientierungsbewegung..... 177
10.3.4	Tensororientierungsbewegung..... 180
10.4	Tangentiale Werkzeugführung..... 184
10.4.1	Beschreibung..... 184
11	Achs-Transformation..... 187
11.1	Einleitung..... 187
11.1.1	Erläuterungen..... 187
11.1.2	Prinzip der Achs-Transformation..... 187
11.1.3	Definitionen..... 188
11.1.4	Wirkstellen 190
11.1.5	Konfigurierung von Achs-Transformationen 191
11.1.6	Ein- und Ausschalten von Achs-Transformationen 192
11.1.7	Anzeige der aktiven Transformations-Daten..... 194
11.1.8	Übersicht Achs-Transformationen (tabellarisch)..... 197
11.2	Bipod-Achs-Transformationen 200
11.2.1	Beschreibung..... 200
11.2.2	Handlungsanweisung: 3/4-Achs-Transformation Bipod..... 202
11.3	Scara-Achs-Transformation 209
11.3.1	Beschreibung..... 209
11.4	Handlungsanweisung: 3-/4-Achs-Transformation SCARA..... 212
11.5	Nichtorthogonale 2- und 3-Achs-Transformation..... 214
11.5.1	Variante 1..... 214
11.5.2	Variante 2..... 216
11.5.3	Variante 3..... 217
11.5.4	Handlungsanweisung: Nichtorthogonale 2/3-Achs-Transformation..... 218
11.6	4-Achs-Transformation - Typ 1: SCARA auf Linearachse..... 223
11.6.1	Beschreibung..... 223
11.6.2	Handlungsanweisung..... 225
11.7	2-dimensionale Delta-Kinematik mit zusätzlicher Drehachse in der Spitze (RRR)..... 231
11.7.1	Beschreibung..... 231
11.7.2	Handlungsanweisung..... 244
11.8	4-Achs-Transformation - Typ 2: LLLR..... 247
11.8.1	Beschreibung..... 247
11.8.2	Handlungsanweisung..... 249
11.9	4-Achs-Transformation - Typ 3: RLLL..... 253
11.9.1	Beschreibung..... 253
11.9.2	Handlungsanweisung..... 257
11.10	5-Achs-Transformation - Typ 1: LLLRR..... 260
11.10.1	Allgemeines zu Varianten 1-3..... 260
11.10.2	Variante 1 (mit Vektororientierung)..... 261
11.10.3	Variante 2 (mit linearer Orientierung)..... 262
11.10.4	Variante 3 (mit linearer Bewegung der rotatorischen Achsen) 263
11.10.5	Gemeinsame Eigenschaften der Varianten 1-3..... 263

	Seite
11.10.6 Handlungsanweisung: 5-Achs-Transformation LLLRR.....	268
11.10.7 Variante 4: 5-Achs-Transformation LLLAB.....	272
11.11 5-Achs-Transformation - Typ 2: RLLLR (mit Rundtisch)	278
11.11.1 Beschreibung.....	278
11.11.2 Handlungsanweisung: 5-Achs-Transformation RLLLR.....	285
11.12 5-Achs-Transformation - Typ 3: RRLLL (mit Schwenk-/Rundtisch)	288
11.12.1 Beschreibung.....	288
11.12.2 Handlungsanweisung: 5-Achs-Transformation RRLLL.....	294
11.13 5-Achs-Transformation - Typ 4: Kardanischer Kopf	298
11.13.1 Beschreibung.....	298
11.13.2 Handlungsanweisung: 5-Achs-Transformation Kardanische LLLRR.....	305
11.14 5-Achs-Transformation - Typ 4.1: Kardanischer Kopf mit linearer Orientierungsbewegung.....	308
11.14.1 Beschreibung.....	308
11.14.2 Handlungsanweisung: Kardanischer Kopf mit linearer Orientierungsbewegung.....	310
11.15 5-Achs-Transformation - Typ 5: Minderbestückte RRLLL (mit Schwenk-/Rundtisch).....	310
11.15.1 Beschreibung.....	310
11.15.2 Applizieren.....	314
11.15.3 Aktivieren.....	316
11.16 5-Achs-Transformation - Typ 6: Kardanische RLLLR	320
11.16.1 Beschreibung.....	320
11.16.2 Handlungsanweisung: 5-Achs-Transformation Kardanische RLLLR	326
11.17 3-Achstransformation LLR.....	330
11.17.1 Beschreibung.....	330
11.17.2 Applizieren.....	333
11.17.3 Aktivieren.....	336
11.18 6-Achs-Transformation.....	336
11.18.1 Beschreibung.....	336
11.18.2 Variante 1 (mit Tensor-Orientierung)	337
11.18.3 Variante 2 (mit linearer Bewegung der Rotationsachsen)	338
11.18.4 Gemeinsame Eigenschaften der Varianten 1, 2.....	339
11.18.5 Handlungsanweisung: 6-Achs-Transformation LLLRR.....	348
11.19 3-Achs-Zylindermantel-Transformation.....	353
11.19.1 Beschreibung.....	353
11.19.2 Variante 1 (Abwicklung)	353
11.19.3 Variante 2 (Ebenenprojektion)	355
11.19.4 Gemeinsame Eigenschaften der Varianten 1, 2.....	356
11.19.5 Handlungsanweisung: 3-Achs-Transformation Zylindermantel.....	358
11.20 4-Achs-Zylindermantel-Transformation	362
11.20.1 Beschreibung.....	362
11.20.2 Handlungsanweisung: 4-Achs-Zylindermanteltransformation.....	367
11.21 5-Achs-Zylindermantel-Transformation.....	370
11.21.1 Beschreibung.....	370
11.21.2 Handlungsanweisung: 5-Achs-Zylindermanteltransformation	376
11.22 Stirnseiten-Transformation.....	379
11.22.1 Beschreibung.....	379
11.22.2 Variante 1 (2-Achs-Transformation).....	379

Inhaltsverzeichnis

	Seite
11.22.3 Variante 2 (3-Achs-Transformation).....	380
11.22.4 Gemeinsame Eigenschaften der Varianten 1, 2.....	380
11.22.5 Handlungsanweisung: 2/3-Achs-Transformation Stirnseite.....	386
11.23 3-Achs-Transformation LLR-kartesisch.....	389
11.23.1 Beschreibung.....	389
11.23.2 Handlungsanweisung.....	394
11.24 Polarkinematik.....	397
11.24.1 Beschreibung.....	397
11.24.2 Handlungsanweisung: Polarkinematik.....	400
11.25 Achs-Kinematikkalibrierung	404
11.25.1 Beschreibung.....	404
11.25.2 Messverfahren.....	405
11.25.3 Handlungsanweisung.....	423
12 Technologien.....	425
12.1 Gewindebohren.....	425
12.1.1 Beschreibung.....	425
12.1.2 Handlungsanweisung: Gewindebohren ohne Ausgleichsfutter applizieren.....	431
12.2 Gewindeschneiden.....	433
12.2.1 Beschreibung.....	433
12.2.2 Relevante Systemdaten.....	439
12.2.3 Applizieren.....	439
12.2.4 Aktivieren.....	439
12.2.5 Deaktivieren.....	439
12.3 Zirkulares Eckenrunden.....	440
12.3.1 Beschreibung.....	440
12.3.2 Handlungsanweisung: Ecken verrunden mit Angabe der Abweichung applizieren und aktivieren/deaktivieren.....	442
12.3.3 Handlungsanweisung: Ecken verrunden mit Radiusangabe applizieren und aktivieren/deaktivieren.....	443
12.4 Eckenrunden mit Splines	445
12.4.1 Beschreibung.....	445
12.4.2 Handlungsanweisung: Eckenrunden mit Splines.....	448
12.5 Stanzen und Nibbeln.....	450
12.5.1 Beschreibung.....	450
12.5.2 Handlungsanweisung: Stanzen und Nibbeln aktivieren.....	453
12.6 Tangentiale Werkzeugorientierung.....	455
12.6.1 Beschreibung.....	455
12.6.2 Aktivieren.....	456
12.6.3 Deaktivieren.....	456
12.7 Laserbearbeitung.....	457
12.7.1 Bahngeschwindigkeitsabhängige Laserleistungssteuerung.....	457
12.7.2 Handlungsanweisung: Bahngeschwindigkeitsabhängige Laserleistungssteuerung.....	460
13 Achskonfiguration im Kanal.....	465
13.1 Achsübergabe.....	465

	Seite
13.1.1 Beschreibung.....	465
13.1.2 Handlungsanweisung: Achsübergabe.....	471
14 Kopplungen.....	485
14.1 Gantry-Achsen.....	485
14.1.1 Beschreibung.....	485
14.1.2 Handlungsanweisung: Gantry-Achsverband applizieren.....	498
14.1.3 Handlungsanweisung: Gantry-Verband aktivieren.....	511
14.1.4 Handlungsanweisung: Maßbezug bei einem absoluten Messsystem herstellen.....	514
14.1.5 Handlungsanweisung: Gantry-Achsverband E-Stopp applizieren.....	515
14.2 Selektive additive Koordinatenkopplung.....	517
14.2.1 Beschreibung.....	517
14.2.2 Handlungsanweisung: Selektive additive Koordinatenkopplung.....	519
14.3 Selektive additive Koordinatenkopplung mit Tabellen.....	521
14.3.1 Beschreibung.....	521
14.3.2 Aktivieren.....	525
14.3.3 Deaktivieren.....	525
14.4 Kopplungstabellen.....	526
14.4.1 Beschreibung.....	526
14.5 Systemachskopplung.....	530
14.5.1 Beschreibung.....	530
14.5.2 Applizieren der Systemachskopplung.....	534
14.5.3 Aktivierung/Deaktivierung.....	560
14.5.4 Synchronisierung.....	563
14.5.5 Systemachskopplung mit Spindeln.....	565
14.5.6 Ausnahmebehandlung.....	567
14.5.7 Anwendungszyklen.....	576
14.5.8 Systemdatenüberblick.....	580
14.6 Kanalachskopplung.....	588
14.6.1 Beschreibung.....	588
14.6.2 Applizieren.....	591
14.6.3 Aktivieren.....	591
14.6.4 Deaktivieren.....	591
15 Messfunktionen.....	593
15.1 Fahren gegen Messtaster (G75).....	593
15.1.1 Beschreibung.....	593
15.1.2 Inbetriebnahme.....	595
15.1.3 Handlungsanweisung: Messen mit Messtasterfunktionen G75.1 und G75.2.....	600
15.2 Fliegendes Messen (FME).....	603
15.2.1 Beschreibung.....	603
15.2.2 Inbetriebnahme.....	605
15.2.3 Handlungsanweisung: Fliegendes Messen.....	608
15.3 Messen auf einem Satz (MOB).....	610
15.3.1 Beschreibung.....	610

	Seite
15.3.2 Applizieren, Aktivieren und Deaktivieren.....	612
15.4 Messen auf einer Kontur (MOC).....	612
15.4.1 Beschreibung.....	612
15.4.2 Applizieren.....	617
15.4.3 Aktivieren.....	617
15.4.4 Deaktivieren.....	617
15.4.5 Handlungsanweisung: Messen auf einer Kontur.....	618
15.4.6 MOC mit "Schnellhalt über Messtastereingang".....	620
15.5 Digitalisieren (Aufnehmen von Messwerten).....	620
15.5.1 Beschreibung.....	620
15.5.2 Handlungsanweisung: Digitalisieren (Aufnehmen von Messwerten).....	631
15.6 Die Aufzeichnungsdatei.....	634
15.6.1 Beschreibung.....	634
15.7 Höhenregelung für Digitalisieren	637
15.7.1 Beschreibung.....	637
15.7.2 Abweichungen überwachen.....	643
15.7.3 Handlungsanweisung: Höhenregelung für Digitalisieren.....	645
15.7.4 Handlungsanweisung: Höhenregelung für Digitalisieren Parameter per CPL anpassen.....	648
15.8 Messen am Festanschlag.....	650
15.8.1 Beschreibung.....	650
15.8.2 Handlungsanweisung: Messen am Festanschlag.....	650
16 Synchronisationsfunktionen.....	653
16.1 Einleitung.....	653
16.2 Synchronisation mit der NC-Vorbereitung.....	655
16.2.1 NC-Vorbereitung und NC-Ausführung (WAIT).....	655
16.2.2 NC-Vorbereitung und Interface/SPS (WAIT(BITIF(...))).....	656
16.2.3 Interface/SPS und NC-Vorbereitung (BITIF, PLC, SD).....	657
16.2.4 Oberfläche (OPC) und NC-Vorbereitung (SD).....	662
16.3 Synchronisation mit der NC-Ausführung.....	664
16.3.1 NC-Ausführung und NC-Vorbereitung.....	664
16.3.2 NC-Ausführung zwischen Kanälen (WEV/WREV/SEV, WPV/SPV).....	666
16.3.3 NC-Ausführung und Interface/SPS (WAITA/WAITO).....	668
16.3.4 NC-Ausführung und Positionen (*STOPA/*STOPO).....	670
16.3.5 SPS und NC-Ausführung (Hilfsfkt., SSD).....	672
16.3.6 Oberfläche (OPC) und NC-Ausführung (SSD/SSDE/SSDQ).....	674
17 Service und Support.....	681
Glossar.....	683
Index.....	685

1 Zu dieser Dokumentation

Ausgaben dieser Dokumentation

Ausgabe	Stand	Bemerkung
01	2014-03	Erstausgabe
02	2015-10	Änderungen eingepflegt
03	2016-07	Änderungen eingepflegt
04	2017-01	Ergänzung in Kap. "Satzvorlauf" Neues Kap. "2-dimensionale Delta-Kinematik mit zusätzlicher Drehachse in der Spitze (RRR)"
05	2018-04	Kap. "Gewindeschneiden": "Gewindeschneiden mit Spindel-Override" ergänzt Kap. "Messfunktionen": "Messen auf einem Satz (MOB)" ergänzt
06	2018-10	Kap. "Bahnbewegung": "Automatische Linear-Zirkular-Spline-Wandlung" ergänzt Kap. "Achs-Transformation": ergänzt
07	2020-10	Kap. "Satzvorlauf": "Projektierung des Satzvorlaufs" ergänzt, Behebung Defdb00176533 Kap. "Spline Programmierung" um Abkündigung der Spline-Typen 4201, 4202, 4203, 4204 und 4205 ergänzt, Behebung Defdb00208667 Kap. "Achstransformationen": "6-Achs-Transformation" korrigiert
08	2021-03	Änderungen eingepflegt Änderungen in Kapitel "Kopplungen"

Tab. 1-1: Änderungsverlauf

1.1 Gültigkeit der Dokumentation

Zweck	Diese Dokumentation beschreibt die Funktionen der Rexroth IndraMotion MTX. Erläutert werden grundlegende Inbetriebnahmeschritte und die Funktionen der Steuerung als Beschreibung und Handlungsanweisung.
Übersicht über Zielgruppen und Produktphasen	In der folgenden Grafik beziehen sich die umrandeten Aktivitäten, Produktphasen und Zielgruppen auf die vorliegende Dokumentation. In der Produktphase "Engineering" kann die Zielgruppe "Programmierer" mit Hilfe dieser Dokumentation die Aktivitäten "parametrieren, programmieren, konfigurieren und simulieren" ausführen.

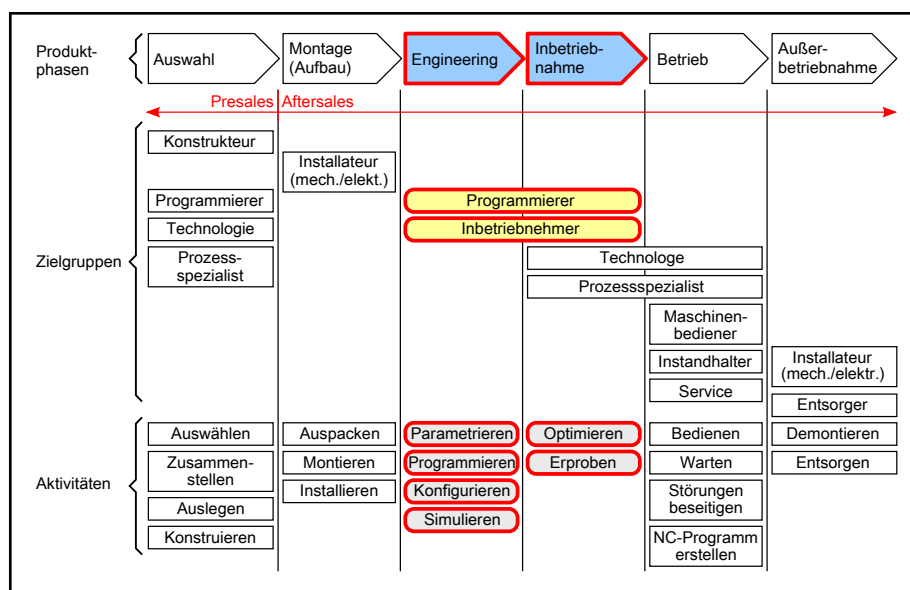


Abb. 1-1: Zuordnung der vorliegenden Dokumentation zu den Zielgruppen, Produktphasen und den Aktivitäten der Zielgruppe

1.2 Allgemeines

Vorliegendes Handbuch unterstützt Sie beim Applizieren/Aktivieren verschiedener Steuerungsfunktionen.

Schwerpunkt ist die erforderliche Vorgehensweise bei der Parametrierung.

Das setzt voraus, dass die komplette Hardware-Einbindung von CNC, Antrieben, SPS und Spannungsversorgungen bereits vollzogen und entsprechend Ihren Applikationsunterlagen inkl. Verkabelung auf Fehlerfreiheit geprüft worden ist. Dazu zählen insbesondere auch alle sicherheitsbezogenen Einrichtungen wie z.B. NOT-AUS-Kreise, Freigabkontakte, Endschalter usw.!

Eine schon erfolgte Software-Inbetriebnahme aller verwendeten Sercos Antriebe wäre zwar ideal, ist aber nicht unbedingt notwendig, weil die Steuerung während des Hochlaufs angeschlossene Antriebe per Parameterdownload konfigurieren kann. Dazu müssen allerdings passende Initialisierungsdateien in der Steuerung existieren.

Um die Initialisierungsdateien anpassen zu können, sollten Sie über entsprechende Unterlagen verfügen (z.B. Listen mit erforderlichen Sercos Parametereinstellungen aller Antriebe, Antriebsdokumentation usw.).

Zur besseren Übersicht sind alle Funktionen verschiedenen übergeordneten Themenbereichen zugeordnet (siehe Inhaltsverzeichnis).

Für jede Funktion erfahren Sie:

- Zweck bzw. Verwendungsmöglichkeit
- Voraussetzungen / Restriktionen zur Nutzung
- Beeinflussungsmöglichkeiten.

Viele Funktionen berühren gleichzeitig mehrere Teilgebiete (Maschinenparameter, NC-Funktionen zur Programmierung, Interface-Signale usw.) und können deshalb auch in anderen Handbüchern dieses Produkts bereits funktional komplett oder in Teilaspekten beschrieben sein.



Schlagen Sie bitte zusätzlich im betreffenden Handbuch nach.

Folgende Handbücher ergänzen das vorliegende Handbuch:

- Maschinenparameter: "Rexroth IndraMotion MTX Maschinenparameter"
- NC- und CPL-Funktionen: "Rexroth IndraMotion MTX Programmierhandbuch"
- Interface-Signale: "Rexroth IndraMotion MTX SPS-Interface"

Außerdem sollten Sie die Antriebs-Dokumentation bereithalten.

Im weiteren Verlauf benötigen Sie gute Kenntnisse über

- die Standard-Bedienoberfläche der Steuerung,
- die Windows-Benutzeroberfläche des PC-Bedienfelds und
- die Werkzeuge zur Konfiguration der Maschinenparameter.

Über diese Werkzeuge ist die Änderung der Steuerungskonfiguration möglich. Sie müssen deshalb mit der Bedienung aller erforderlichen Werkzeuge vertraut sein!

⚠ VORSICHT

Ungenügend oder nicht geschultes Personal kann durch falschen oder nicht sachgemäßen Umgang schwere Schäden an der Maschine, Datenverlust oder sogar Personenschäden verursachen!

Deshalb dürfen Start und Bedienung, sowie Einstellungen oder Änderungen von Konfigurationsparametern nur durch entsprechend geschultes Fachpersonal durchgeführt werden!

Dieses Personal muss in der Lage sein, Gefahren, welche durch Parameteränderungen und allgemein durch die mechanische, elektrische oder elektronische Ausrüstung verursacht werden können, zu erkennen und zu vermeiden.

Wir haften nicht für Schäden oder Folgeschäden, die durch falschen oder nicht sachgemäßen Umgang mit den Ausrüstungsteilen hervorgerufen werden.

1.3 Erforderliche und ergänzende Dokumentationen MTX

1.3.1 Auswählen/Zusammenstellen

Dokumentationstitel mit Typenschlüssel und Materialnummer

MTX 14VRS Systembeschreibung DOK-MTX***-SYS*DES*V14-PRRS-DE-P, R911342639 Diese Dokumentation beschreibt das Steuerungssystem MTX. Beschrieben werden die Ausführungen, Technischen Daten, Schnittstellen und die Konfiguration der Steuerungsbaugruppen.
MTX 14VRS SafeLogic Systemübersicht DOK-MTX***-SL**SYS*V14-PRRS-DE-P, R911341697 Diese Dokumentation beschreibt den Einsatz der Sicherheitssteuerung SafeLogic im System MTX.

Tab. 1-2: MTX-Dokumentationsübersicht - Auswählen/Zusammenstellen

1.3.2 Konfigurieren

Dokumentationstitel mit Typenschlüssel und Materialnummer

MTX 14VRS Maschinenparameter DOK-MTX***-MA*PAR**V14-RERS-DE-P, R911342629 Diese Dokumentation beschreibt den Aufbau und die Anpassung der verfügbaren Parameter der MTX. Ebenfalls beschrieben werden Funktion und Bedienung des NC-Konfigurators.
MTX 14VRS SPS-Interface DOK-MTX***-PLC*INT*V14-PRRS-DE-P, R911342621 Diese Dokumentation beschreibt die Interface-Signale und die Programmbausteine für die integrierte SPS.
MTX 14VRS Funktionsbeschreibung - Grundlagen DOK-MTX***-NC*F*BA*V14-RERS-DE-P, R911342625 Diese Dokumentation beschreibt die Basisfunktionen der MTX. Erläutert werden grundlegende Inbetriebnahmeschritte und die Funktionen der Steuerung als Beschreibung und Handlungsanweisung.
MTX 14VRS Funktionsbeschreibung - Erweiterung DOK-MTX***-NC*F*EX*V14-RERS-DE-P, R911342627 Diese Dokumentation beschreibt die erweiterten Funktionen der MTX. Erläutert werden grundlegende Inbetriebnahmeschritte und die Funktionen der Steuerung als Beschreibung und Handlungsanweisung.
MTX 14VRS Funktionsbeschreibung - Sonderfunktionen DOK-MTX***-NC*F*SP*V14-RERS-DE-P, R911342623 Diese Dokumentation beschreibt die Sonderfunktionen der MTX. Erläutert werden grundlegende Inbetriebnahmeschritte und die Funktionen der Steuerung als Beschreibung und Handlungsanweisung.
MTX Freiformflächenbearbeitung DOK-MTX***-FREEFORM***-APRS-DE-P, R911341434 Diese Dokumentation beschreibt die Vorgehensweise bei der Freiformflächenbearbeitung mit der MTX-Steuerung. CNC-Programme, die von einem CAD/CAM-System generiert werden, dienen als Grundlage für den gesamten Bearbeitungsprozess. Es wird ein Überblick über folgende Themen gegeben: Beschreibung der MTX- bzw. CNC-Funktionen zur Freiformflächenbearbeitung, Vorgehensweise bei der NC-Parametrierung, Rahmenbedingungen bei der Erzeugung von CNC-Programmen mit CAM-Software.
MTX Konvertierung von MTX-Projekten DOK-MTX***-PROCONV****-PRRS-DE-P, R911342483 Diese Dokumentation unterstützt bei der Konvertierung von MTX 1.x Projekten nach MTX 2G. Es wird die Vorgehensweise beschrieben und auf mögliche Schwierigkeiten bei der Konvertierung hingewiesen.

Tab. 1-3: MTX-Dokumentationsübersicht - Konfigurieren

1.3.3 In Betrieb nehmen

Dokumentationstitel mit Typenschlüssel und Materialnummer

IndraWorks 13VRS Simulation DOK-IWORKS-SIMU****V13-APRS-DE-P, R911339255 Diese Dokumentation beschreibt die Funktionen der Simulationskomponenten View3D, virtuelles Bedienfeld und virtuelle Steuerung und deren Bedienung in IndraWorks.
MTX 14VRS Inbetriebnahme DOK-MTX***-STARTUP*V14-CORS-DE-P, R911342619 Diese Dokumentation beschreibt die Inbetriebnahme der Steuerung MTX. Neben einer Gesamtübersicht werden die Inbetriebnahme und Konfiguration der Achsen und Bedienoberfläche und der SPS-Daten erläutert.

IndraWorks 14VRS Basis-Bibliotheken IndraLogic 2G DOK-IL*2G*-BASLIB**V14-LIRS-DE-P, R911343918 Das vorliegende Handbuch beschreibt die systemübergreifenden SPS-Bibliotheken.
IndraWorks 14VRS Feldbusse DOK-IWORKS-FB*****V14-APRS-DE-P, R911341484 Die Dokumentation dient der Beschreibung der von den Systemen XLC, MLC und MTX unterstützten Feldbus- und lokalen Peripherieanschlüssen. Der Schwerpunkt dieser Dokumentation liegt in der Konfiguration, Parametrierung, Inbetriebnahme und Diagnose der verschiedenen Peripherieanschlüssen. Sie ist die Grundlage für die Online-Hilfe.
IndraWorks 14VRS WinStudio 7.4 DOK-IWORKS-WINSTUD*V14-APRS-DE-P, R911341584 Dieses "Benutzerhandbuch und Technische Referenz" soll Sie dabei unterstützen, mit Ihrer Software "Rexroth WinStudio"™ optimale Ergebnisse zu erzielen. Das Dokument bietet technische Informationen und Schritt-für-Schritt-Anleitungen zum Anlegen von webfähigen HMI/SCADA-Programmen.
IndraWorks 14VRS Software-Installation DOK-IWORKS-SOFTINS*V14-CORS-DE-P, R911344285 Die Dokumentation beschreibt den Installationsprozess von IndraWorks.
IndraWorks 14VRS Engineering DOK-IWORKS-ENGINEE*V14-APRS-DE-P, R911343564 Diese Dokumentation beschreibt die Verwendung von IndraWorks, in das sich die Engineering-Tools integrieren. Beschrieben werden das Arbeiten mit IndraWorks und die Bedienung der Oszilloskop-Funktion.
IndraWorks 12VRS FDT-Container DOK-IWORKS-FDT*CON*V12-APRS-DE-P, R911334397 Diese Dokumentation beschreibt die FDT-Container-Funktionalität von IndraWorks. Beschrieben wird das Aktivieren der Funktionalität im Projekt und das Arbeiten mit DTMs.
IndraWorks 14VRS SPS-Programmiersystem IndraLogic 2G DOK-IWORKS-IL2GPRO*V14-APRS-DE-P, R911343570 Diese Dokumentation beschreibt das SPS-Programmierungswerkzeug IndraLogic 2G und seine Verwendung. Beschrieben werden die Basisverwendung, Erste Schritte, die Visualisierung, Menüpunkte und die Editoren.
IndraWorks 14VRS HMI DOK-IWORKS-HMI*****V14-APRS-DE-P, R911343567 Diese Dokumentation beschreibt die Funktionen, die Konfiguration und die Bedienung der Bedienoberflächen IndraWorks HMI Engineering und IndraWorks HMI Operation.

Tab. 1-4: MTX-Dokumentationsübersicht - In Betrieb nehmen

1.3.4 Bedienen

Dokumentationstitel mit Typenschlüssel und Materialnummer

MTX 09VRS Werkstattprogrammierung Drehen und Fräsen DOK-MTX***-SF*PROG*V09-AWxx-DE-P, R911324376 Diese Dokumentation beschreibt die Bedienung und die Programmierung der Grafischen NC-Programmierung der Technologien Drehen und Fräsen, einschließlich der Werkstücksimulation.
MTX 12VRS Satzvorlauf DOK-MTX***-BLK*RUN*V12-APRS-DE-P, R911334378 Die Doku soll den Maschinenhersteller in die Lage versetzen, die Funktion "Satzvorlauf" an seiner Maschine für den Endanwender einzurichten.

Zu dieser Dokumentation

MTX 14VRS Programmierhandbuch

DOK-MTX***-NC**PRO*V14-RERS-DE-P, R911342633

Das vorliegende Handbuch beschreibt die Standard-Programmierung der Steuerung MTX. Beschrieben werden neben den Grundlagen der NC-Programmierung sowohl die Anwendung der NC-Funktionen nach DIN 66025, als auch der NC-Funktionen mit Hochsprachensyntax und CPL-Funktionen.

MTX 14VRS Standard-NC-Bedienung

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

Diese Dokumentation beschreibt die Bedienung der Standard-Bedienoberfläche der NC-Steuerung MTX. Beschrieben werden die Bedienung der Oberfläche, die NC-Programmerstellung und die Werkzeugverwaltung.

MTX 14VRS Multitouch

DOK-MTX***-MULTI***V14-APRS-DE-P, R911384734

In dieser Dokumentation wird die Bedienung der Multitouch-Bedienoberfläche der NC-Steuerung MTX beschrieben.

MTX 14VRS Standard-NC-Zyklen

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

Diese Dokumentation beschreibt für die Steuerung MTX die Anwendung der Standard-Zyklen der verschiedenen Technologien.

MTX 14VRS NC-Simulation

DOK-MTX***-NC*SIM**V14-APRS-DE-P, R911379628

Diese Dokumentation beschreibt für die Steuerung MTX die NC-Simulation.

xx entsprechender Ausgabestand
Tab. 1-5: MTX-Dokumentationsübersicht - Bedienen

1.3.5 OEM-Entwicklung

Dokumentationstitel mit Typenschlüssel und Materialnummer

MTX 13VRS Automation Interface

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

Diese Dokumentation beschreibt den scriptbasierten Zugriff auf IndraWorks-Projektdateien über die Schnittstelle des Automation Interfaces. Beschrieben werden die verschiedenen Objekte, einschließlich Code-Beispielen. Eine Beschreibung des Automation Builders schließt sich an.

MTX 14VRS OPC-Kommunikation

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

Diese Dokumentation beschreibt die Syntax und den Aufbau der Items für die Kommunikation mit den Bosch Rexroth-Geräten.

IndraWorks OPC UA Kommunikation

DOK-IWORKS-OPC*UA*****-APRS-DE-P, R911379308

Diese Dokumentation beschreibt die OPC UA Kommunikation der Steuerungssysteme MLC und MTX.

Tab. 1-6: MTX-Dokumentationsübersicht - OEM-Entwicklung

1.3.6 AddOns

Dokumentationstitel mit Typenschlüssel und Materialnummer

MTX 11VRS Aktionsrekorder DOK-MTX***-ACR*****V11-APxx-DE-P, R911329942 Diese Dokumentation beschreibt den MTX Aktionsrekorder. Beschrieben werden sowohl die Installation und die Inbetriebnahme, einschließlich Schnittstellensignale, als auch die Anwendung und die Bedienung.
MTX 14VRS efficiency workbench MTX cta, MTX ega DOK-MTX***-EWB*****V14-APRS-DE-P, R911343915 Diese Dokumentation beschreibt die Funktionsweise und den Einsatzbereich der Analysetools MTX cta und MTX ega.
MTX Remote Condition Monitoring DOK-MTX***-RCM*****V01-APRS-DE-P, R911334382 Diese Dokumentation beschreibt die Bedienung des Remote Condition Monitoring Systems.
MTX visIREC Anwenderdokumentation DOK-MTX***-VISIREC*V01-APRS-DE-P, R911344241 Diese Dokumentation beschreibt das Analysewerkzeug visIREC, das zur Optimierung des Freiformflächenbearbeitungsprozesses eingesetzt werden kann. 2D- bzw. 4D-Anzeige der bahnbezogenen Daten. 2D- bzw. 4D-Anzeige der koordinatenbezogenen Daten. Analyse von kritischen Bereichen (Bahn- und Orientierungsabweichung). Vergleich zwischen den programmierten und den interpolierten NC-Sätzen.

xx entsprechender Ausgabestand
Tab. 1-7: MTX-Dokumentationsübersicht - AddOns

1.4 Darstellung von Informationen

1.4.1 Gebrauch der Sicherheitshinweise

Aufbau der Sicherheitshinweise

Die Sicherheitshinweise sind wie folgt aufgebaut:

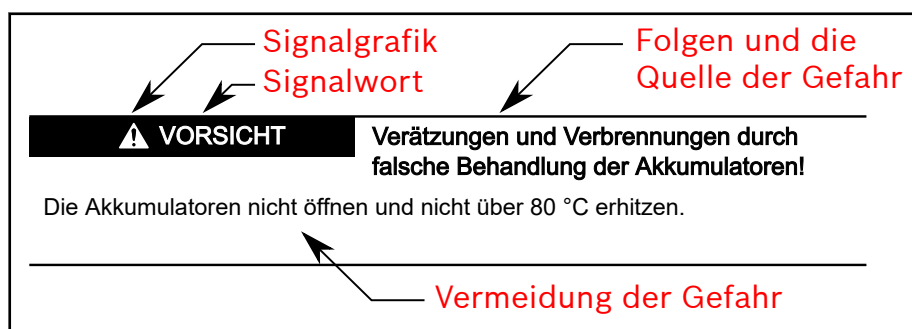


Abb. 1-2: Aufbau der Sicherheitshinweise

Erläuterung der Signalwörter und der Signalgrafik

Die Sicherheitshinweise in der vorliegenden Dokumentation beinhalten bestimmte Signalwörter (Gefahr, Warnung, Vorsicht, Hinweis) und gegebenenfalls eine Signalgrafik (nach ANSI Z535.6-2006).

Das Signalwort soll die Aufmerksamkeit auf den Sicherheitshinweis lenken und bezeichnet die Schwere der Gefährdung.

Die Signalgrafik (Warndreieck mit Ausrufezeichen), welche den Signalwörtern Gefahr, Warnung und Vorsicht vorangestellt wird, weist auf Gefährdungen für Personen hin.

⚠ GEFAHR

Bei Nichtbeachtung dieses Sicherheitshinweises **werden** Tod oder schwere Körpervletzung eintreten.

⚠ WARNUNG

Bei Nichtbeachtung dieses Sicherheitshinweises **können** Tod oder schwere Körpervletzung eintreten.

⚠ VORSICHT

Bei Nichtbeachtung dieses Sicherheitshinweises können mittelschwere oder leichte Körpervletzung eintreten.

HINWEIS

Bei Nichtbeachtung dieses Sicherheitshinweises können Sachschäden eintreten.

Verwendete Symbole

Fingerzeige werden wie folgt dargestellt:



Dies ist ein Hinweis.

Tipps werden wie folgt dargestellt:



Dies ist ein Tipp.

Erläuterung der Signalgrafik auf dem Gerät



Wenn dieses Symbol am Gerät angebracht ist, beachten Sie unbedingt die Dokumentation zu dem Gerät. In der jeweiligen Dokumentation finden Sie die Art der Gefährdung sowie die notwendigen Schritte zur Vermeidung der Gefährdung.

1.4.2 Bezeichnungen und Abkürzungen

Begriff	Erklärung
IWE	IndraWorks Engineering
IWO	IndraWorks Operation
LWL	Lichtwellenleiter
NC	Numerical Control
OEM	Original Equipment Manufacturer
PROFIBUS	Kommunikationsverbindung
Sercos	Kommunikationsverbindung
MP	Maschinenparameter

Tab. 1-8: Verwendete Bezeichnungen und Abkürzungen

1.5 Kundenfeedback

Anregungen, Wünsche oder Verbesserungen von unseren Kunden haben bei uns einen hohen Stellenwert. Senden Sie uns Ihre Anmerkungen zu den Dokumentationen per E-Mail an Feedback.Documentation@boschrexroth.de. Sie können direkt im elektronischen PDF-Dokument Kommentare einfügen und uns die PDF-Datei zusenden.

2 Wichtige Gebrauchshinweise

2.1 Bestimmungsgemäßer Gebrauch

2.1.1 Einführung

Die Produkte von Bosch Rexroth werden nach dem jeweiligen Stand der Technik entwickelt und gefertigt. Vor ihrer Auslieferung werden sie auf ihren betriebssicheren Zustand hin überprüft.

Die Produkte dürfen nur bestimmungsgemäß eingesetzt werden. Wenn sie nicht bestimmungsgemäß eingesetzt werden, dann können Situationen entstehen, die Sach- und Personenschaden nach sich ziehen.



Für Schäden bei nicht-bestimmungsgemäßigem Gebrauch der Produkte leistet Bosch Rexroth als Hersteller keinerlei Gewährleistung, Haftung oder Schadensersatz; die Risiken bei nicht-bestimmungsgemäßigem Gebrauch der Produkte liegen allein beim Anwender.

Bevor Sie die Produkte der Firma Bosch Rexroth einsetzen, müssen die folgenden Voraussetzungen erfüllt sein, um einen bestimmungsgemäßen Gebrauch der Produkte zu gewährleisten:

- Jeder, der in irgendeiner Weise mit einem unserer Produkte umgeht, muss die entsprechenden Sicherheitsvorschriften und den bestimmungsgemäßen Gebrauch lesen und verstehen.
- Sofern es sich bei den Produkten um Hardware handelt, müssen sie in ihrem Originalzustand belassen werden; d. h. es dürfen keine baulichen Veränderungen an ihnen vorgenommen werden. Softwareprodukte dürfen nicht dekompiert werden und ihre Quellcodes dürfen nicht verändert werden.
- Beschädigte oder fehlerhafte Produkte dürfen nicht eingebaut oder in Betrieb genommen werden.
- Es muss gewährleistet sein, dass die Produkte entsprechend den in der Dokumentation genannten Vorschriften installiert sind.



Die der Steuerung vorliegenden oder vom Benutzer eingegebenen bzw. eingelesenen Daten sind vor der Übernahme/Aktivierung auf ihre Richtigkeit zu prüfen, um ungewollte Bewegungen der Achsen auszuschließen. Im Einzelnen kann es sich um folgende ungültige oder veraltete Daten handeln:

- Teileprogramme
- NPV-Tabellen
- Korrekturtabellen
- Werkzeugtabellen
- Permanente CPL-Variablen
- Remanente SPS-Daten
- Permanente Systemdaten

2.1.2 Einsatz- und Anwendungsbereiche

Die MTX-Steuerung dient zum

- Programmieren von Kontur und Bearbeitungstechnologie (Bahnvorschub, Spindeldrehzahl, Werkzeugwechsel) eines Werkstückes.
- Führen eines Bearbeitungswerkzeuges entlang einer programmierten Bahn.

Vorschubantriebe, Spindeln und Hilfsachsen einer Werkzeugmaschine werden per Sercos Schnittstelle angesteuert.



Zusätzlich werden E/A-Komponenten für die integrierte SPS benötigt, die – in Kombination mit der eigentlichen CNC – den Bearbeitungsprozess ganzheitlich steuert und auch in sicherheitstechnischer Hinsicht überwacht.

Der Betrieb darf nur in den ausdrücklich angegebenen Konfigurationen und Kombinationen der Hardware-Komponenten und mit der in den jeweiligen Dokumentationen und den Funktionsbeschreibungen angegebenen und spezifizierten Soft- und Firmware erfolgen.

Das CNC-System MTX bietet die perfekte CNC-Systemlösung für Zerspanen und Umformen für die folgenden Technologien:

- Drehen
- Fräsen
- Bohren
- Schleifen
- Biegen
- Nibbeln
- Stanzen
- Formschneiden
- Handhaben

2.2 Nicht-bestimmungsgemäßer Gebrauch

Die Verwendung des CNC-Systems MTX außerhalb der vorgenannten Anwendungsgebiete oder unter anderen als den in der Dokumentation beschriebenen Betriebsbedingungen und angegebenen technischen Daten gilt als "nicht-bestimmungsgemäß".

Das CNC-System MTX darf nicht eingesetzt werden, wenn sie ...

- Betriebsbedingungen ausgesetzt wird, die die vorgeschriebenen Umgebungsbedingungen nicht erfüllen. Untersagt sind z. B. der Betrieb unter Wasser, unter extremen Temperaturschwankungen oder extremen Maximaltemperaturen.
- Außerdem darf das CNC-System MTX nicht bei Anwendungen eingesetzt werden, die von Bosch Rexroth nicht ausdrücklich freigegeben sind. Beachten Sie hierzu bitte unbedingt die Aussagen in den allgemeinen Sicherheitshinweisen!
- Das CNC-System MTX darf nicht an Anlagen oder Maschinen eingesetzt werden, die über eine ungesicherte Netzwerkverbindung mit dem Internet verbunden ist. Andernfalls besteht die Gefahr von Störungen bzw. einem Steuerungsausfall aufgrund unautorisierter Zugriffe.

3 Antriebe (Achsen, Spindeln)

3.1 Lineare Moduloachse

3.1.1 Beschreibung

Funktion

Appliziert eine Linearachse als Endlosachse. Erreicht die Linearachse den Modulowert, wird ihr Sollwert automatisch auf "0" gesetzt. Das verhindert den rechnerischen Überlauf der Achswerte und gestattet ein programmierbares Nullsetzen des Programmkoordinatensystems auf die aktuelle Achsposition durch LinModZp (LMZ).

Auf diese Weise kann die Achse theoretisch endlos in eine Richtung verfahren werden. Die Funktion eignet sich daher für förderbandähnliche Anwendungen wie z.B. Förderung eines Drahtes von einer Trommel, Wickeln von Federn usw.

Geschwindigkeitsvorgabe für asynchrone lineare Endlosachsen

Bei asynchronen Endlosachsen ist eine Geschwindigkeitsprogrammierung im Lageregelungsbetrieb möglich. Diese kann insbesondere für Leitachsen in elektronischen Getrieben (Systemachskopplung) genutzt werden.

Die Angabe der Geschwindigkeit erfolgt mit dem S-Attribut (<Achse>=S<Geschwindigkeit>) in der Einheit Modulintervalle/Minute. Mit dem Erreichen der Geschwindigkeitsvorgabe wird das Achsinterface-Signal „iAx_ProgSpReach“ gesetzt. Die Geschwindigkeitsvorgabe ist beliebig änderbar und kann fliegend in eine Positionierbewegung überführt werden.

Restriktionen

- Der relevante Antrieb muss in der Lage sein, die Modulorechnung selbstständig durchzuführen. Der Modulowert lässt sich antriebsseitig in der Regel per S-0-0103 parametrieren.
- In einem Verfahrssatz dürfen nur Positionswerte programmiert werden, die kleiner/gleich dem Modulowert sind.
- Negative Positionswerte dürfen nur dann programmiert werden, wenn sie kleiner als der Modulowert sind.
- Grundstellung setzt den Programmiernullpunkt auf den Achsnullpunkt zurück.
- Messen (InitMeas(IME) / FlyMeas(FME)) in Verbindung mit linearen Moduloachsen ist nur bei Verfahrbewegungen in positiver Richtung erlaubt.
- Die Grenzgeschwindigkeit bei Moduloachsen kann den halben Modulobereich (S-0-103) pro Interpolationstakt nicht überschreiten.

Beispiel:

S-0-103=20 mm, Sercos Takt 4 ms entspricht der Grenzgeschwindigkeit 150000 mm/min bzw. einem S-Wert von 7500 MOD/MIN.

Relevante NC-Funktionen

LinModZp(...)	Nullsetzen Moduloachse
<Achse>=S(...)	Geschwindigkeitsprogrammierung in Modulintervalle/MIN

Tab. 3-1: Relevante NC-Funktionen

Relevante Sercos Parameter

S-0-1050.1.6	Konfigurationsliste AT Dort muss je nach verwendetem Geber der Sercos Parameter S-0-0051 (Lageistwert1 Motorgeber) oder S-0-0053 (Lageistwert2 externer Geber) eingetragen sein.
S-0-1050.0.6	Konfigurationsliste MDT Dort muss der Sercos Parameter S-0-0047 (Lagesollwert) eingetragen sein. In Verbindung mit IndraDrive-Antrieben kann zusätzlich P-0-100 (Lagesollwert Erweiterung) eingetragen werden.
S-0-0032	Hauptbetriebsart Muss auf die Betriebsart "Lageregelung" eingestellt sein.
S-0-0033	Nebenbetriebsart 1 Muss auf die Betriebsart "Lageregelung schleppfehlerfrei" eingestellt sein.
S-x-0103	Modulowert Modulowert eintragen

Tab. 3-2: Relevante Sercos Parameter

Relevante IF-Signale

qAx_SwLimOff	Ausblenden der Software-Endschalter
iAx_ProgSpReach	Vorgegebene Geschwindigkeit der Endlosachse erreicht.

Tab. 3-3: Relevante IF-Signale

Relevante Maschinenparameter (MP)

1003 00004	Achsbewegungstyp.

Tab. 3-4: Relevante Maschinenparameter

3.1.2 Handlungsanweisung: Appliziert eine Linearachse als Moduloachse

Appliziert eine Linear Achse als Endlosachse. Erreicht die Linearachse den Modulowert, wird ihr Sollwert automatisch auf "0" gesetzt. Das verhindert einen rechnerischen Überlauf.

NC: Linearachse anlegen

Bevor die Modulorechnung aktiviert werden kann, muss eine Linearachse angelegt werden.

		Instruction
Instruction:		Linearachse anlegen

IW-Engineering / Configuration: Parameter editieren

- **SysDrName**
"Physikalische Antriebsbezeichnung" (1003 00001): den Systemachsennamen eintragen.
- **EnablDr**
"Aktiviere Antrieb" (1001 00001): relevanten Antrieb mit Wert "Ja" aktivieren.
- **EnablModCalc**

"Aktiviere Modulorechnung" (1003 00004): auf "Ja" setzen.

- **ModePosLog**

"Positionierlogik" (1003 00005): die gewünschte Positionierlogik auswählen.

- **EnablChgPosLog**

"Positionierlogik per NC-Prog. Umschaltbar" (1003 00050): legt fest, ob mit Hilfe von PosMode(..), die Positionierlogik im NC-Programm umschaltbar ist.



Der relevante Antrieb muss in der Lage sein, die Modulorechnung selbständig durchzuführen. Der Modulwert lässt sich antriebsseitig in der Regel per Sercos Parameter **S-0-0103** "Modulwert" parametrieren.

In einem Verfahrenssatz dürfen nur Positionswerte zwischen -Modulwert und +Modulwert programmiert werden.

Messen (**InitMeas(...)** "Initialisierung der Messtasterlogik" / **FlyMeas(...)** "Fliegendes Messen") in Verbindung mit linearen Moduloachsen ist nur bei Verfahrensbewegungen in positiver Richtung erlaubt.

Solange **G91** aktiv ist (Relativmaß-Programmierung) wird die Modulorechnung für synchrone Moduloachsen unter den Betriebsarten Automatik und Handeingabe bis auf den ausgegebenen Sollwert und dessen Anzeige NC-Intern unterdrückt.

Maßeinheit gegebenenfalls anpassen

	Documentation	Instruction
Instruction:		Editieren von Maschinenparametern
Documentation:	MTX Maschinenparameter	Achsparemeter

IW-Operation / NC-Programmierung: Nullsetzen Moduloachse

LinModZp(...) gestattet ein programmierbares Nullsetzen des Programmkoordinatensystems auf die aktuelle Achsposition

	Documentation	Instruction
Instruction:		Editieren von SPS-Signale
Documentation:	MTX Maschinenparameter	Linearachssignale

3.2 Hirthachse

3.2.1 Beschreibung

Funktion

Appliziert eine Rund- oder Endlosachse als Hirthachse. Hirthachsen fahren nur Positionen an, die auf einem festgelegten Raster liegen (Hirthraster; ⇒ MP 1003 00055).

Die Funktion eignet sich daher z.B. für Achsen von Werkzeugwechslern oder Achsen mit Verzahnung.

Wahlweise ist die Programmierung in Grad oder in Plätzen einstellbar (⇒ MP 1003 00056).

Bei **Platzprogrammierung** geben Sie die Zielposition als ganzzahlige Platznummer ohne Kommastellen an.

Platz 1 entspricht immer der Position 0 Grad. Alle Plätze sind in Abhängigkeit von MP 1003 00055 durchnummeriert und werden innerhalb der Positionsanzeige auch als Platznummer angezeigt. Negative Platznummern, Platznummer 0 oder Platznummern größer als in MP 1003 00055 angegeben, lassen sich nur bei aktivem G91 programmieren (Relativmaß-Programmierung).

Sofern die Hirthachse **in Grad programmiert** werden soll, ist eine automatische Endpunktkorrektur möglich ($\Rightarrow$ MP 1003 00057). Hierbei korrigiert die NC eine programmierte Position selbstständig auf die nächstliegende Position des Hirthrasters. Ohne Endpunktkorrektur erzeugt die NC in solchen Fällen eine Fehlermeldung.

Restriktionen

- Eine Hirthachse ist nicht als Folgeachse eines Kopplungsverbandes verwendbar. Als Führungsachse darf sie verwendet werden.
- Kein Handradbetrieb erlaubt.
- Eine synchrone Hirthachse kann nicht Bestandteil der Achsen sein, die mit der Funktion "Schiefe Ebene" angewählt werden.
- Referenzpunkt, Kontur- und Nullpunktverschiebungen der Hirthachse müssen auf dem Hirthraster liegen
- Beim "inkrementalen Tippen" (Schrittmaß ist aktiv), nach "Vorschub-Halt", "Vorschub-Sperre", "Not-Aus" oder "Grundstellung" kann die Hirthachse auch zwischen den erlaubten Positionen zum Stehen kommen.

Relevante IF-Signale

qAx_NextNotch	"Nächste Rasterposition" (Achsinterface) High-Signal startet die Verfahrbewegung, sofern Betriebsart "Tippen" aktiv ist, die Hirthachse aber momentan nicht manuell getippt wird. Die aktuelle Verfahrriichtung meldet die NC über das Achsinterface iAx_TrDirNeg "Negative Verfahrriichtung".
iAx_NotchPos	"Achse auf Rasterposition" (Achsinterface) Bei nicht referenzierter Achse ist das Signal immer "0". Bei referenzierter Achse ist es nur dann gesetzt, wenn sich die Achse im "In-Positions-Fenster" befindet.

Tab. 3-5: Relevante IF-Signale

Relevante Maschinenparameter (MP)

1003 00055	Plätze pro Kreis für Hirthachse
1003 00056	Platzprogrammierung bei Hirthachse
1003 00057	Endpunktkorrektur für Hirthachsen

Tab. 3-6: Relevante Maschinenparameter

3.2.2 Handlungsanweisung: Appliziert eine Hirthachse

Appliziert eine Rund- oder Endlosachse als Hirthachse. Hirthachsen fahren nur Positionen an, die auf einem fest gelegten Raster liegen. Die Funktion eignet sich daher z.B. für Achsen von Werkzeugwechslern oder Achsen mit Verzahnung.

NC: Rund- oder Endlosachse applizieren

Die Achse ist bereits als Rund- oder Endlosachse im System vorhanden und bezogen auf Geschwindigkeit, Dynamik und ggf. Positionierlogik - entsprechend Ihren Anforderungen - korrekt parametriert.

		Instruction
Instruction:		Rund- und Endlosachse applizieren

IW-Engineering / Configuration: Parameter editieren

- **ModelLocProg** "Art der Platzprogrammierung" (1003 00056):
legt die gewünschte Programmierung fest.
- **NofLoc** "Anzahl der Plätze" (1003 00055):
legt das erforderliche Hirthraster fest.
- **EnablAutoCorr** "Aktiviere autom. Platzkorrektur" (1003 00057):
kann gegebenenfalls die Endpunktkorrektur aktiviert werden.

	Documentation	Instruction
Instruction:		Editieren von Maschinenparametern
Documentation:	MTX Maschinenparameter	Platzprogrammierung



- Die notwendigen achsbezogenen Interface-Signale werden innerhalb des SPS-Programms ausgewertet oder gesetzt.
- Maßeinheit gegebenenfalls anpassen.

IW-Engineering / IndraLogic: Relevante Interface-Signale (SPS → NC)

- **qAx_NextNotch**:
"Nächste Rasterposition" (Achsinterface). High-Signal startet die Verfahrbewegung, sofern Betriebsart "Tippen" aktiv ist, die Hirthachse aber momentan nicht manuell getippt wird. Die aktuelle Verfahrrichtung meldet die NC über das Achsinterface **iAx_TrVDirNeg** "Negative Verfahr-richtung".
- **qAx_JogPlus** "Manuell +" / **qAx_JogMinus** "Manuell -":
wirken für Kanalachsen (synchrone Achsen) und Hilfsachsen (asynchrone Achsen) in den Betriebsarten "Einrichten Manuell" und "Einrichten Referenzpunkt anfahren". Darüber hinaus dienen die Signale zum manuellen Wegtippen von synchronen Achsen beim Wegfahren von der Kontur. Das aktive Teileprogramm ist dabei mit Vorschub Halt angehalten.

	Documentation	Instruction
Instruction:		Editieren von SPS-Signale
Documentation:	MTX SPS-Interface	Hirthachssignale

IW-Engineering / IndraLogic: Interface-Signale zur Kontrolle (NC → SPS)

- **iAx_NotchPos**:
"Achse auf Rasterposition" (Achsinterface). Bei nicht referenzierter Achse ist das Signal immer 0. Bei referenzierter Achse ist es nur dann gesetzt, wenn sich die Achse im "In-Positions-Fenster" befindet.

- **iAx_TrDirNeg:**

"Negative Verfahrriichtung". Gesetzt bedeutet, dass die betreffende Achse eine negative Verfahrbewegung ausführen soll oder ausgeführt hat.

	Documentation	Instruction
Instruction:		Editieren von SPS-Signale
Documentation:	MTX SPS-Interface	Hirhachssignale

Problemlösung: Bekannte Zustandsmeldungen

Zustand: Hirhachse steht nicht auf dem Hirhraster...

...nach "Vorschub-Halt" oder "Vorschub-Sperre":

1. "Vorschub-Halt" oder "Vorschub-Sperre" wegnehmen.
⇒ Die zuvor programmierte Endposition wird sofort auf dem Hirhraster angefahren.

...nach "Not-Aus" oder "Grundstellung":

1. Handvorschub auf "Eilgang", "Schnell", "Mittel" oder "Langsam" stellen.
2. Tippbetrieb aktivieren.
3. Hirhachse kurz tippen (per Achsinterface-Signale qAx_JogPlus "Manuell+" / qAx_JogMinus "Manuell-").
⇒ Die NC fährt je nach Tipprichtung (+/-) auf die nächstmögliche Position des Hirhrasters.

- oder -

1. Tippbetrieb aktivieren
2. Achsinterface-Signal "Fahre zu nächster Rasterposition" setzen.
⇒ Die NC fährt auf die nächstmögliche Position des Hirhrasters. Die Verfahrriichtung wird automatisch durch den aktuellen Zustand des Achsinterface-Signals iAx_TrDirNeg "Negative Verfahrriichtung" bestimmt.

...nach "inkrementalem Tippen" (Schrittmaß ist aktiv):

1. Handvorschub auf "Eilgang", "Schnell", "Mittel" oder "Langsam" stellen.
2. Hirhachse kurz tippen (per Achsinterface-Signal qAx_JogPlus "Manuell+" oder qAx_JogMinus "Manuell-").
⇒ Die NC fährt je nach Tipprichtung (+/-) auf die nächstmögliche Position des Hirhrasters.

- oder -

3. Achsinterface-Signal "Fahre zu nächster Rasterposition" setzen.
⇒ Die NC fährt auf die nächstmögliche Position des Hirhrasters. Die Verfahrriichtung wird automatisch durch den aktuellen Zustand des Achsinterface-Signals iAx_TrDirNeg "Negative Verfahrriichtung" bestimmt.

	Documentation	
Documentation:	MTX Funktionsbeschreibung	Hirhachse

4 Bedienfunktionen (manueller Betrieb)

4.1 Tippen in Werkstückkoordinaten

4.1.1 Beschreibung

Funktion

Mit dieser Funktion können Sie die Werkstückkoordinaten (WCS) und, falls eine orientierungsfähige Achs-Transformation aktiv ist, die Z-Richtung des Werkzeugs (TCS) tippen. Hierfür steht die Betriebsart **"Einrichtbetrieb, Tippen in Werkstückkoordinaten"** zur Verfügung.

Die Anwahl der zu tippenden Koordinate erfolgt über ein spezielles Koordinatenanwahlprogramm (dessen Name wird in MP 705001110 hinterlegt). Alternativ kann die Anwahl auch von der SPS über NC-Satzvorgabe oder über ein beliebiges Teileprogramm erfolgen. Die Anwahl erfolgt in allen Fällen mit der NC-Funktion "JogWCSSelect."



Zu einem Zeitpunkt kann immer nur eine Koordinate bewegt werden.

Vor dem Tippen muss die Betriebsart "Einrichtbetrieb, Tippen in Werkstückkoordinaten" angewählt sein (per Kanal-IF-Signal qCh_OpModeSel_00 "Anw. Betriebsart Bit 0" bis qCh_OpModeSel_03 "Anw. Betriebsart Bit 3").

Tipp-Varianten:

- **Kontinuierliches Tippen:**

Beim Tippen wird für die angewählte Koordinate eine positive oder negative Endposition vorgegeben. Die Bewegung wird beim Erreichen eines Achsendschalters oder einer Arbeitsraumgrenze gestoppt. Beim Loslassen des Tiptasters wird die Bewegung gestoppt.

- **Inkrementelles Tippen:**

Beim Betätigen des Tiptasters wird das angewählte Inkrement vorgegeben.



Kann das angewählte Weginkrement nicht verfahren werden, fährt die NC bis zum maximal möglichen Endpunkt!

Soll für das Tippen eine besondere Umgebung hergestellt werden (Achs-Transformation, Placements, usw), ist das per **Initialisierungsprogramm** möglich. Der Name dieses Programms wird in MP 70500 1100 eingetragen. Dieses Programm wird zusammen mit der ersten Tippbewegung ausgeführt.

Relevante NC-Funktionen

JogWCSSelect	Informationen über die Syntax (siehe Programmierhandbuch)

Tab. 4-1: Relevante NC-Funktionen

Relevante IF-Signale

qCh_JogPlusWcs	WCS Manuell+
qCh_JogMinusWcs	WCS Minus-
qCh_OpModeSel_00 bis qCh_OpModeSel_03	Anw. Betriebsart Bit 0 bis 3

Tab. 4-2: Relevante IF-Signale

Relevante Maschinenparameter

1010 00002	Tipp-Beschleunigung
7050 01000	Tippen WCS: Handvorschübe
7050 01010	Tippen WCS: Vorschub und Schrittmaß, Systemachsnummer
7050 01020	Tippen WCS: Auflösen eines Inkrements
7050 01030	Tippen WCS: Variabler Inkrementen-Schritt
7050 01100	Tippen WCS: Programmname für die Standardkonfiguration
7050 01110	Tippen WCS: Programmname für die Koordinatenwahl

Tab. 4-3: Relevante Maschinenparameter

4.1.2 Handlungsanweisung: Tippen in Werkstückkoordinaten

Applizieren:

IW-Operation / NC-Programmierung: zu bewegende Koordinaten anwählen

- mit einem **Koordinatenanwahlprogramm**:

Eine Koordinate wird mit Hilfe eines Koordinatenanwahlprogramms bestimmt. Der Name des Programms (Koordinatenanwahlprogramm) wird in den Parameter "**CoordProgName** - Programmname für die Koordinatenanwahl (7050 01110)" eingetragen. Dieses Programm wird bei jeder Betätigung des Tiptasters ausgeführt.

Der Name des Initialisierungsprogramms ist in den Parameter "**TrafoProgName** "Programmname für die Standardkonfiguration (7050 01100)" anzugeben. Dieses Programm wird zusammen mit der ersten Tippbewegung ausgeführt.

Initialisierungsprogramm "InitstringForJogWCS"

Programm:

```
N10 Coord (@AXTRAFO)
```

Koordinatenanwahlprogramm "CoordSelectForJogWCS"

Programm:

```
10 COORDNO% = PLC(<Typ>,<DB-Nummer>,<Adresse>,<Gr÷e>)  
N10 JogWCSSelect JWSCoord[COORDNO%]
```

- mit einem **beliebigen Teileprogramm**
- oder -
- über die **SPS mit NC-Satzvorgabe** per SPS-Baustein **MT_NcBlk**
- oder -
MT_NcBlkExt.



- **JogWCSSelect** siehe "MTX Programmierhandbuch".
- Die erstellten Programme müssen im Suchpfad abgelegt werden.

	Documentation	Instruction
Documentation:	MTX Programmierhandbuch	Koordinatenanwahlprogramm

IW-Engineering / Configuration: Parameter editieren

- **AccAxJog**
"Tipp-Beschleunigung" (1010 00002)
- **WcsJogIf**
"Systemachsnr. des Interfaces für Tippen in WCS" (7050 01010)
- **IncrWcsJog**
"Gewichtung eines Tipp-Inkrementes" (7050 01020)
- **DistWcsJog**
"Variable Tipp-Distanz" (7050 01030)
- **SlowVelWcs**
"Tipp-Geschw. Langsam" (7050 01000 [1])
- **MedVelWcs**
"Tipp-Geschw. Mittel" (7050 01000 [2])
- **FastVelWcs**
"Tipp-Geschw. Schnell" (7050 01000 [3])
- **RapidVelWcs**
"Tipp-Geschw. Eilgang" (7050 01000 [4])
- **DefStepVelWcs**
"Tipp-Geschw. definierte Schritte" (7050 01000 [5])
- **VarStepVelWcs**
"Tipp-Geschw. variable Schritte" (7050 01000 [6])
- **TrafoProgName**
"Programmname für die Standardkonfiguration" (7050 01100)
- **CoordProgName**
"Programmname für die Koordinatenanwahl" (7050 01110)



Die erstellten Programme müssen im Suchpfad abgelegt werden.

Die sich hieraus ergebenden Tippvarianten sind:

Bedienfunktionen (manueller Betrieb)

Tippmodus ⁽¹⁾	Länge des Fahrweges ⁽²⁾	wirksame Geschwindigkeit	wirksame Beschleunigung
Eilgang	abhängig von der Dauer des gesetzten Kanal-IF-Signals "Manuell -" bzw. "Manuell +"	RapidVelWcs	AccAxJog (falls mit "0" belegt, gilt der Wert aus FeedAxAcc)
Schnell		FastVelWcs	
Mittel		MedVelWcs	
Langsam		SlowVelWcs	
x Inkremente	pro L/H-Flanke an "Manuell +" bzw. "Manuell -": DistWcsJog * IncrWcsJog	VarStepVelWcs	
1000 Inkremente	pro L/H-Flanke an "Manuell +" bzw. "Manuell -": 1000 * IncrWcsJog	DefStepVelWcs	
100 Inkremente	pro L/H-Flanke an "Manuell +" bzw. "Manuell -": 100 * IncrWcsJog		
10 Inkremente	pro L/H-Flanke an "Manuell +" bzw. "Manuell -": 10 * IncrWcsJog		
1 Inkremente	pro L/H-Flanke an "Manuell +" bzw. "Manuell -": 1 * IncrWcsJog		

(1) Anwahl durch Setzen der durch WcsJogIf "Systemachsnr. des Interfaces für Tippen in WCS" bestimmten Ax-IF-Signale qAx_ManFeed_00 bis qAx_ManFeed_03 "Handvorschub Bit0 bis Bit3".

(2) Die Funktionen G70/G71 und DIA/RAD werden beim inkrementellen Tippen berücksichtigt. Wenn eine lineare Koordinate bewegt wird und G70 aktiv ist, multipliziert die NC den in "IncrWcsJog" eingetragenen Wert mit dem Faktor 2,54. Ist für die gewählte Koordinate die Funktion DIA aktiv, teilt die NC den in "IncrWcsJog" eingetragenen Wert durch 2.

Tab. 4-4: Übersicht Parameter Tippen in WCS

	Documentation	Instruction
Instruction:		Editieren von Maschinenparameter
Documentation:	MTX Maschinenparameter	Tippen in WCS

Aktivieren: IW-Operation: Betriebsart anwählen

Anwahl der Kanal-Betriebsart "Einrichten manuell Werkstückkoordinaten" über die Interface-Signale qCh_OpModeSel_00 .. qCh_OpModeSel_03 zu wählen

Betriebsart	Anwahl Betriebsart			
	Bit 3 (qCh_OpModeSel_03)	Bit 2 (qCh_OpModeSel_02)	Bit 1 (qCh_OpModeSel_01)	Bit 0 (qCh_OpModeSel_00)
Einrichten manuell Werkstückkoordinaten	1	1	1	0

Example	Documentation	Instruction
Example:		Kanalbetriebsarten in WCS

Instruction:		Editieren von SPS-Signalen
Documentation:	MTX SPS-Interface	Betriebsart wählen

IW-Operation / IndraLogic: Geschwindigkeitsstufe oder Schrittweite anwählen

- An dem durch **WcsJogIf** "Systemachsnr. des Interfaces für Tippen in WCS" bestimmten Achsinterface durch Setzen der Signale qAx_ManFeed_00 bis qAx_ManFeed_03 "Handvorschub Bit0 bis Bit3" die "Geschwindigkeitsstufe" oder die "Schrittweite" anwählen.

	Documentation	Instruction
Instruction:		Editieren von SPS-Signalen
Documentation:	MTX SPS-Interface	Geschwindigkeitsstufe / Schrittweite

IW-Operation / IndraLogic: Bewegung starten

Die Bewegung durch Setzen von **qCh_JogPlusWcs** "WCS Manuell +" oder **qCh_JogMinusWcs** "WCS Manuell -" starten.

	Documentation	Instruction
Instruction:		Editieren von SPS-Signalen
Documentation:	MTX SPS-Interface	Bewegung starten

Deaktivieren:

IW-Operation / IndraLogic: Bewegung abbrechen

Eine kontinuierliche Bewegung wird durch Loslassen des entsprechenden Tipptasters gestoppt, wenn das Kanalinterface-Signal **qChJogPlusWcs** "WCS Manuell +" oder **qCh_JogMinusWcs** "WCS Manuell -" zurückgesetzt wird.

	Documentation	Instruction
Instruction:		Editieren von SPS-Signalen
Documentation:	MTX SPS-Interface	Bewegung stoppen

IW-Operation / IndraLogic: Tippen verlassen

- Verlassen der Kanal-Betriebsart "Einrichten manuell Werkstückkoordinaten" beendet jede Tippbewegung.
- Kanalgrundstellung und Systemgrundstellung brechen eine aktive Tippbewegung ab.

	Documentation	Instruction
Instruction:		Editieren von SPS-Signalen
Documentation:	MTX SPS-Interface	Tippen verlassen

4.2 Handrad in Werkstückkoordinaten

4.2.1 Allgemeines

Funktion

Mit der Funktion "Handrad in Werkstückkoordinaten" bewegt das Handrad eine Werkstückkoordinate eines Kanals. Optional kann durch Setzen des Systemdatums "SysHandwWCS" eine Bewegung in der aktiven Ebene ausgeführt werden. Als Konturelemente stehen Gerade und Kreis zur Verfügung.

*Beispiel:***Handrad in Werkstückkoordinaten**

Ist eine entsprechende Achs-Transformation aktiv, kann dadurch ein Werkzeug in TCS-Z-Richtung durch kombinierte Bewegung der linearen Raumkoordinaten (x, y, z) per Handrad verfahren werden.

Koordinatenanwahl im WCS:

Die Anwahl der Koordinate oder des Konturelements erfolgt über ein spezielles Koordinatenanwahlprogramm dessen Name in CoordProgName (MP 705010110) hinterlegt ist. Dieses Programm wird bei jeder Handradanwahl aufgerufen.

Alternativ kann die Anwahl auch von der SPS oder über ein beliebiges Teilprogramm erfolgen.

Die Anwahl erfolgt in allen Fällen mit der NC-Funktion "JogWCSSelect" und dem Systemdatum "SysHandwWCS". Eine Auswahl über "SysHandwWCS" hat Vorrang vor der Auswahl durch "JogWCSSelect".



Bei jeder Anwahl des Handrads wird die aktuelle Vorwahl der Koordinate oder des Konturelements übernommen.

Handrad-Anwahl:

Für die zu bewegende Koordinate ergeben sich folgende Rahmenbedingungen bei der Handrad-Anwahl:

- Betriebsart "Einrichtbetrieb, Tippen in Werkstückkoordinaten" (relevante Betriebsart für "Handrad in WCS") muss angewählt sein.
- Gleichzeitig kann nur eine Werkstückkoordinate bewegt werden.
- Nur innerhalb der ermittelten Positionsgrenzen (maximale positive und negative Endposition) kann die Koordinate bewegt werden.
- Der Zustand des Interface-Signals qCh_HandwPosMode wird bei der Handradanwahl übernommen.
- Wird durch die Handradvorgabe die für die angewählte Koordinate maximal mögliche Bahngeschwindigkeit überschritten, dann ergibt sich das Verhalten aus dem Zustand des IF-Signals qCh_HandwPosMode zum Zeitpunkt der Handradanwahl:
 - qCh_HandwPosMode nicht gesetzt:
Die Geschwindigkeit wird begrenzt und der aufsummierte Bahnweg entsprechend reduziert.
 - qCh_HandwPosMode gesetzt:
Die Geschwindigkeit wird begrenzt.
- Aktive Endpositionskopplung (EPC) der angewählte Koordinate wird berücksichtigt.
- G70 (Inch) aktiv: für lineare Koordinaten werden die Handradinkremente mit dem Faktor 2,54 multipliziert (0,001 mm → 0,0001 Inch).
- Aktive Durchmesserprogrammierung (DIA) der angewählten Koordinate wird berücksichtigt.
- Handradbewegungen werden ignoriert bei aktiver/aktivem
 - Vorschub Sperre oder Vorschub Halt (bei Bahnbewegung)
 - Vorschub Sperre, Antrieb Aus oder Sicherheitsabschaltung (bei Achsbewegung einer beteiligten Achse)

Soll für den Handradbetrieb eine besondere Umgebung hergestellt werden (Achs-Transformation, Placements, usw.), ist das per Initialisierungsprogramm möglich. Der Name dieses Programms wird in MP 70500 1100 eingetragen. Dieses Programm bei der ersten Handradanwahl ausgeführt.

Restriktion

- Nur die Koordinaten, die auch für Onlinekorrektur freigeschaltet sind, können mittels Handrad bewegt werden.
Siehe Dokumentation "MTX Funktionsbeschreibung Sonderfunktionen", Kap. "Online-Korrektur mit Handrad (in Werkstückkoordinaten)".
- Die Koordinaten der folgenden Achs-Transformationen können nicht durch das Handrad bewegt werden:
Transformation-Typ 3232201: phi und theta
Transformation-Typ 3232202: phi und theta
Transformation-Typ 3232203: phi und theta
Transformation-Typ 3232211: phi und theta
Transformation-Typ 3021002: x und y
Transformation-Typ 2011001: x und y
Transformation-Typ 2103011: x und y

Relevante NC-Funktionen

JogWCSSelect	Koordinatenauswahl und andere Parameter Informationen über die Syntax (siehe "MTX Programmierhandbuch").

Tab. 4-5: Relevante NC-Funktionen

Relevante IF-Signale

qCh_OpModeSel_00 bis qCh_OpModeSel_03	"Anw. Betriebsart Bit 0 ... 3"
qCh_OpModePLC	"Betriebsart von SPS"
qCh_HandwSelWcs00	"Handradauswahl Bit 0"
qCh_HandwSelWcs01	"Handradauswahl Bit 0"
qCh_HandwDirWcs	"Handraddrehrichtung"
qChHandwPosMode	"Positionshandrad"
iCh_OpMode_00 bis iCh_OpMode_03	"Aktive Betriebsart Bit 0 ... 3"
qAx_ManFeed_00 bis qAx_ManFeed_03	"Handvorschub Bit 0 bis 3" Achsennummer siehe MP 7050 01010

Tab. 4-6: Relevante IF-Signale

Relevante Maschinenparameter

1010 00002	Tippen WCS: Tippbeschleunigung der, für den Kanal relevanten Koordinate.
7050 01010	Tippen WCS: Vorschub und Schrittmaß, Systemachsnummer.
7050 01020	Tippen WCS: Auflösen eines Inkrements.
7050 01030	Tippen WCS: Variabler Inkrementen-Schritt.
7050 01100	Tippen WCS: Programmname für die Standardkonfiguration (Initialisierungsprogramm).
7050 01110	Tippen WCS: Programmname für die Koordinatenwahl.
9060 00001	Anzahl der Inkremente pro Umdrehung (Handrad).

Tab. 4-7: Relevante Maschinenparameter

Relevante Systemdaten

Elementname	Beschreibung	Typ	Größe in Byte	Schreibrecht
PathMode	Konturhandrad? 0: Nein 1: Ja	UnsignedByte_t	1	jeder
ContourType	Art der Kontur? 0: Gerade 1: Kreisbogen	UnsignedByte_t	1	jeder
Angle	Richtung der Geraden Winkel zwischen der positiven Hauptachsrichtung und der Geraden in Grad. Für ContourType = 0.	Float_t	4	jeder
MainCP	Lage des Kreismittelpunkts in der Hauptachse in mm. Für ContourType = 1.	Float_t	4	jeder
ScndCP	Lage des Kreismittelpunkts in der Nebenachse in mm. Für ContourType = 1.	Float_t	4	jeder

Tab. 4-8: Strukturelemente SysHandwWCS

Durch Setzen des Systemdatums

SysHandwWCS[<Kanalnr.>].PathMode

werden die zu bewegendenden Koordinaten aus der Haupt- und Nebenachse der aktiven Ebene bestimmt. Die Art der Bewegung ergibt sich aus dem Inhalt des Systemdatums

SysHandwWCS[<Kanalnr.>].ContourType.

ContourType = 0

ergibt eine Gerade. Die positive Richtung der Geraden in der aktuellen Ebene wird durch den Winkelparameter SysHandwWCS[<Kanalnr.>].Angle bestimmt.

ContourType = 1

ergibt einen Kreisbogen. Die Lage des Mittelpunkts des Kreisbogens wird über die Systemdaten SysHandwWCS[<Kanalnr.>].MainCP und .ScndCP festgelegt.

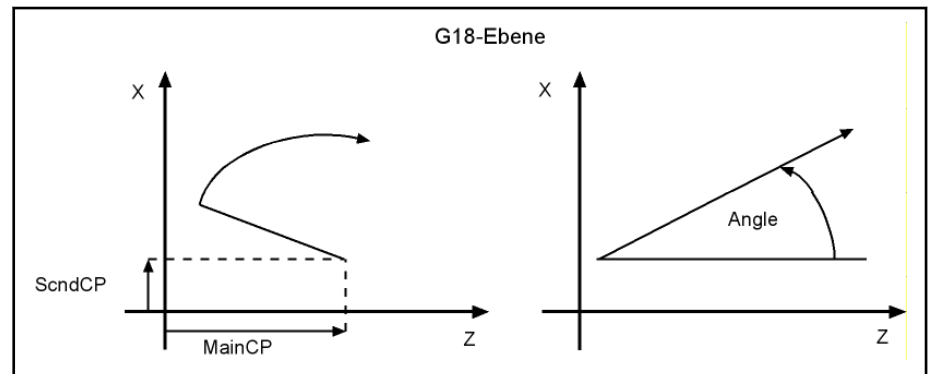


Abb. 4-1: G18-Ebene

4.2.2 Handlungsanweisung: Handrad in Werkstückkoordinaten

Applizieren: IW-Operation / NC-Programmierung: zu bewegende Koordinate anwählen

- mit einem **Koordinatenanwahlprogramm**:

Eine Koordinate wird mit Hilfe eines Koordinatenanwahlprogramms bestimmt. Der Name des Programms (Koordinatenanwahlprogramm) wird in den Parameter "**CoordProgName** - Programmname für die Koordinatenanwahl (7050 01110)" eingetragen. Dieses Programm wird bei jeder Handradanwahl ausgeführt.

Der Name des Initialisierungsprogramms ist in den Parameter **TrafoProgName** "Programmname für die Standardkonfiguration" (7050 01100) anzugeben. Diese Programm wird zusammen mit der ersten Handradanwahl ausgeführt.

Initialisierungsprogramm "InitstringForJogWCS"

Programm:

```
N10 Coord (@AXTRAFO)
```

Koordinatenanwahlprogramm "CoordSelectForJogWCS"

Programm:

```
10 COORDNO% = PLC(<Typ>,<DB-Nummer>,<Adresse>,<Gr÷e>)
N10 JogWCSselect JWSCoord[COORDNO%]
```

- mit einem **beliebigen Teileprogramm**
 - oder -
 - über die **SPS mit NC-Satzvorgabe** per SPS-Baustein **MT_NcBlk**
 - oder -
- MT_NcBlkExt.**



- JogWCSselect** siehe "MTX Programmierhandbuch".
- Die erstellten Programme müssen im Suchpfad abgelegt werden.

	Documentation	Instruction
Documentation:	MTX Programmierhandbuch	Koordinatenanwahlprogramm

IW-Engineering / Configuration: Parameter editieren

- **AccAxJog**
"Tipp-Beschleunigung" (1010 00002)
- **IncrPerRev**
"Anzahl der Inkremente/Umdrehung" (9060 00001)
- **WcsJogIf**
"Systemachs. des Interfaces für Tippen in WCS" (7050 01010)
- **IncrWcsJog**
"Gewichtung eines Tipp-Inkrementes" (7050 01020)
- **DistWcsJog**
"Variable Tipp-Distanz" (7050 01030)
- **TrafoProgName**
"Programmname für die Standardkonfiguration" (7050 01100)
- **CoordProgName**
"Programmname für die Koordinatenanwahl" (7050 01110)



Die erstellten Programme müssen im Suchpfad abgelegt werden.

Schrittmaß ⁽¹⁾	Länge des Fahrweges ⁽²⁾	wirksame Geschwindigkeit
x Inkremente	pro Skalenteil am Handrad: $\text{DistWcsJog} * \text{IncrWcsJog}$	abhängig von der Drehgeschwindigkeit des Handrades.
1000 Inkremente	pro Skalenteil am Handrad: $1000 * \text{IncrWcsJog}$	
100 Inkremente	pro Skalenteil am Handrad: $100 * \text{IncrWcsJog}$	
10 Inkremente	pro Skalenteil am Handrad: $10 * \text{IncrWcsJog}$	
1 Inkremente	pro Skalenteil am Handrad: $1 * \text{IncrWcsJog}$	

- (1) Anwahl durch Setzen der durch WcsJogIf "Systemachsnr. des Interfaces für Tippen in WCS" bestimmten Ax-IF-Signale qAx_ManFeed_00 bis qAx_ManFeed_03 "Handvorschub Bit0 bis Bit3".
- (2) Die Funktionen G70/G71 und DIA/RAD werden bei der Berechnung der Länge des Fahrweges berücksichtigt. Wenn eine lineare Koordinate bewegt wird und G70 aktiv ist, multipliziert die NC den in "IncrWcsJog" eingetragenen Wert mit dem Faktor 2,54. Ist für die gewählte Koordinate die Funktion DIA aktiv, teilt die NC den in "IncrWcsJog" eingetragenen Wert durch 2.

Tab. 4-9: Übersicht Parameter in WCS

	Documentation	Instruction
Instruction:		Editieren von Maschinenparametern
Documentation:	MTX Maschinenparameter	Handrad in WCS

Aktivieren: IW-Operation / IndraLogic: Betriebsart wählen

Anwahl der Kanalbetriebsart "Einrichten manuell Werkstückkoordinaten" über die Interface-Signale **qCh_OpModeSel_00...qCh_OpModeSel_03** zu wählen.



- Per Interface-Signal **qCh_HandwPosMode** zwischen den Handradbetriebsarten "Positionsgenau" oder "Bewegungsgenau" wählen.
- Die Kanalinterface-Signale **qCh_JogPlus** "Manuell+" bzw. **qCh_JogMinus** "Manuell-" sind nicht aktiv.

Bedienfunktionen (manueller Betrieb)

Betriebsart	Anwahl Betriebsart			
	Bit 3 (qCh_OpMode- Sel_03)	Bit 2 (qCh_OpMode- Sel_02)	Bit 1 (qCh_OpMode- Sel_01)	Bit 0 (qCh_OpMode- Sel_00)
Einrichten manuell Werkstückkoordinaten	1	1	1	0

	Documentation	Instruction
Instruction:		Editieren von SPS-Signalen
Documentation:	MTX SPS-Interface	Handrad auswählen

IW-Engineering / IndraLogic: Handrad auswählen

Anwahl per Achsinterface-Signale **qCh_HandwSelWcs_00** "Handradauswahl Bit 0" oder **qCh_HandwSelWcs_01** "Handradauswahl Bit 1".



- Per Interface-Signal **qCh_HandwPosMode** zwischen den Handradbetriebsarten "Positionsgenau" oder "Bewegungs-genau" wählen.
- Die Interface-Signale **qCh_JogPlusWcs** "Manuell+" bzw. **qCh_JogMinusWcs** "Manuell-" sind nicht aktiv.

	Documentation	Instruction
Instruction:		Editieren von SPS-Signalen
Documentation:	MTX SPS-Interface	Handrad auswählen

IW-Engineering / IndraLogic: Drehrichtung vorgeben

Geben Sie per Interface-Signal **qCh_HandwDirWcs** "Handrad Drehrichtung" vor, wie die Steuerung die Drehrichtung des Handrades interpretieren soll.

	Documentation	Instruction
Instruction:		Editieren von SPS-Signalen
Documentation:	MTX SPS-Interface	Drehrichtung vorgeben

Hand Wheel: Verfahrbewegung auslösen

Verfahrbewegung durch Drehen des Handrades auslösen.

Deaktivieren: IW-Operation / IndraLogic: deaktivieren

Setzen Sie das entsprechende Achsinterface-Signal **qAx_HandwSelWcs_00** "Handradauswahl Bit 0" oder **qAx_HandwSelWcs_01** "Handradauswahl Bit 1" zurück.



Nach Abwahl der Betriebsart "Tippen in Werkstückkoordinaten" oder nach Kanal- oder Systemgrundstellung muss das Handrad erneut angewählt werden, da zwischen Koordinate und Handrad kein Bezug mehr existiert.

	Documentation	Instruction
Instruction:		Editieren von SPS-Signalen
Documentation:	MTX SPS-Interface	Handrad deaktivieren

5 Kanalsteuerung

5.1 Weg- und Wiederanfahren

5.1.1 Beschreibung

Funktion

Die Funktion gestattet das Weg- und Wiederanfahren an die Kontur nach Unterbrechung eines Teileprogramms oder einer Satzfolge in der Betriebsart "MDI-Satzeingabe" mit "Vorschub Halt". Das kann z. B. nach Werkzeugbruch, zur manuellen Werkstück-/Werkzeugvermessung oder für eine optische Kontrolle erforderlich sein.

Für die Wiederanfahrstrategie ist frei wählbar:

- der Anfahrmodus
(**wie** soll die NC die Kontur wieder anfahren: Manuell, Automatik, Einzelsatz)
- Anfahrpunkt
(**an welcher Stelle** soll die NC die Kontur wieder anfahren: Unterbrechungspunkt, Satzanfang, Satzende)
- die Anfahrstrategie
legt fest, ob vom sog. Aufsetzpunkt aus linear oder tangential im Kreisbogen an den ausgewählten Anfahrpunkt angefahren wird.



Per MP OPF/ReturnPath/Ch[k]/ReturnMode (7060 00310) und MP OPF/ReturnPath/Ch[k]/ReturnPoint (7060 00320) lassen sich gewünschter Anfahrmodus und Anfahrpunkt voreinstellen.

Ausführliche Informationen über die Wirkungsweise der einzelnen Optionen finden Sie in [Kap. 5.1.2 "Optionen zum Wiederanfahren"](#) auf Seite 34.

Wird ein Teileprogramm (oder eine Satzfolge in der Betriebsart "MDI-Satzeingabe") durch "Vorschub Halt" unterbrochen, kann die Funktion "Weg- und Wiederanfahren" an die Kontur verwendet werden. Dazu muss zunächst der Modus "Wegfahren von der Kontur" aktiviert werden.

Ein Wegfahren von der Kontur löst intern immer ein Restweg löschen und damit einen Achsstillstand aus. Anschließend werden alle Sätze, die vor dem Wegfahren bereits vorbereitet waren, verworfen.

Nach dem Wiederanfahren werden alle verworfenen Sätze wieder neu vorbereitet.



Wurden innerhalb des Bereichs der bereits vorbereiteten Sätze Korrekturwerte verändert (z. B. Nullpunkte oder Werkzeugdaten), so wirken diese geänderten Werte jetzt auf alle neu aufzubereitenden Sätze.

Ist ein solches Verhalten nicht erwünscht, muss vor jedem die Korrekturwerte beeinflussenden Befehl wie z. B. ZOT oder DCT eine separate Zeile mit einem "WAIT" programmiert werden!

Alternativ kann der Refresh-Mechanismus von Tabellen und Korrekturregistern mit Hilfe des Systemdatums /SysCorrRefrDis auch global unterdrückt werden. Änderungen von Korrekturwerten werden in diesem Fall erst mit der nächsten NPV-Programmierung bzw. der nächsten Tabellen-Aktivierung wirksam.



Wie viele Sätze in der Satzvorbereitung gehalten werden sollen, lässt sich per Maschinenparameter NCO/LookAh/Ch[k]/NofBk-Prep (7060 00110) einstellen.

Ab diesem Zeitpunkt "merkt" sich die Steuerung alle Verfahrbewegungen, die Sie über die achsbezogenen IF-Signale qAx_JogPlus "Manuell+" oder qAx_JogMinus "Manuell-" auslösen. Bei Achsstillstand gibt die Steuerung das Kanal-IF-Signal iCh_ReadyReEnter "Bereit zum Wiederanfahren" aus.



Per MP OPF/ReturnPath/Ch[k]/NofJogSteps (7060 00330) kann die maximale Anzahl der zu merkenden Verfahrbewegungen konfiguriert werden.

Sollen keine weiteren Verfahrbewegungen mehr aufgezeichnet werden, aktivieren Sie den Modus "Wiederanfahren". Die NC setzt daraufhin das Kanal-IF-Signal iCh_RemoveFinish "Wegfahren beendet". Das Tippen der Achsen ist in diesem Modus weiterhin möglich.

In Abhängigkeit der gewählten Anfahrstrategie (siehe [Kap. 5.1.2 "Optionen zum Wiederanfahren" auf Seite 34](#)) lässt sich per "NC-Start" das Wiederanfahren an die Kontur starten. Nach dem Ende der Wiederanfahrbewegung wird automatisch in die zuvor aktive Betriebsart zurückgeschaltet und die Bearbeitung fortgesetzt.

Beim Wiederanfahren an die Kontur spielt der "Aufsetzpunkt" eine wichtige Rolle, weil die Strecke zwischen Aufsetzpunkt und dem Anfahrpunkt (Wiedereintrittspunkt auf der Kontur) zum Einrechnen erforderlicher Korrekturwerte dient.

Der Aufsetzpunkt entspricht beim

- Anfahrmodus "Manuell" der aktuellen Position
- Anfahrmodus "Automatisch" oder "Einzelsatz" der Position am Ende der ersten aufgezeichneten Verfahrbewegung.

Durch günstige Wahl des Aufsetzpunktes kann man die Strecke Aufsetzpunkt-Anfahrpunkt so gestalten, dass ein möglichst "weicher" Kontureintritt erzielt wird.



Im Modus "Wegfahren von der Kontur" lässt sich die Bearbeitung jederzeit am eingestellten Anfahrpunkt durch "NC-Start" fortsetzen, sofern das Eingangssignal "Wiederanfahren an die Kontur" (Kanalinterface "qCh_RetCont" nicht gleichzeitig gesetzt ist.

Die Steuerung führt in diesem Fall direkt von der aktuellen Position auf den eingestellten Anfahrpunkt, rechnet hierbei evtl. erforderliche Korrekturwerte ein und setzt die Bearbeitung fort.

Durch günstige Wahl des Aufsetzpunktes kann ein möglichst weicher Kontureintritt erzielt werden.

Nach NC-Start zur Bearbeitung im Aufsetzpunkt wird ein asynchrones Unterprogramm (ASUP) aufgerufen, wenn es in den Systemdaten angemeldet ist. Der Name des asynchronen Unterprogramms wird im Systemdatum **SysReentry** in der Komponente **NameOfAsup** abgelegt.

Die Anmeldung des asynchronen Unterprogramm erfolgt z. B. im Teileprogramm (CPL):

```
10 SD.SysReentry.NameOfAsup="/usrfep/ReentryAsup.cnc"
```

Am Ende des asynchronen Unterprogramms wird, je nach Einstellung, auf den Start-, Unterbrechungs- oder Endpunkt des Abbruchsatzes gefahren. Die

Kontur kann linear oder tangential angefahren werden. Danach wird die normale Bearbeitung fortgesetzt.

Während das asynchrone Unterprogramm läuft, ist das Systemdatum "SD.SysReentry.AsupActive" auf 1 gesetzt.

Für die Anfahrstrategie tangential im Kreisbogen erfolgt das Anfahren derart, dass der Kontureintritt über einen Kreisbogen mit einem, über ein Systemdatum, vorgebbaren Radius erfolgt. Der Kreisbogen liegt dabei innerhalb der aktiven Arbeitsebene (G17, G18, G19). Bezogen auf diese Arbeitsebene erfolgt der Kontureintritt tangential.

Die vorgelagerte Bewegung vom Aufsetzpunkt an den Anfahrkreis erfolgt auf einer Geraden derart, dass bezogen auf die aktive Arbeitsebene ebenfalls ein tangentialer Übergang von dieser Geraden auf den Kreisbogen erfolgt.

Anhand der Komponente **FeedHold** wird eingestellt, ob nach dem Wiederaufahren an die Kontur automatisch ein Vorschubhalt gesetzt werden soll.

Restriktionen

- Beim Wegfahren wirken die Tipp-Signale (qAx_JogPlus "Manuell+" bzw. qAx_JogMinus "Manuell-") direkt auf die jeweiligen Maschinenachsen. Eine evtl. aktive Achstransformation oder eine Drehung des Programmkoordinatensystems (PCS) bezogen auf das Basiskoordinatensystems (BCS) wird nicht berücksichtigt!
- Wiederaufahren ist nicht in mehreren Kanälen gleichzeitig möglich.
- Wiederaufahren ist erst nach Ende eines ggf. aktiven CPL-Wartezustandes (z. B. WAIT) möglich. Ist ein CPL-Wartezustand zum Unterbrechungszeitpunkt aktiv, gibt die Steuerung eine Warnung aus.
- G63 (Gewindebohren) darf nicht aktiv sein.
- Laufende Spindeln müssen nach der Unterbrechung entweder manuell (per Schaltfunktion) oder durch die SPS gestoppt und rechtzeitig vor Beginn der Bearbeitung wieder gestartet werden.
- Weg- und Wiederaufahren bricht den Rückwärtsbetrieb ab.
- Beim Wiederaufahren ist bis zum Erreichen des Anfahrpunktes erneutes Weg- und Wiederaufahren nicht möglich. Ebenso ist Umschalten auf Rückwärtsbetrieb nicht möglich.

Relevante IF-Signale

am Achsinterface	
qAx_JogPlus	Manuell+
qAx_JogMinus	Manuell-
am Kanalinterface	
qCh_NCStart	NC-Start
qCh_OpModeSel_00 bis qCh_OpModeSel_03	Betriebsart Anwahl Bit 0 bis 3 Erforderliche Betriebsart: "Wiederaufahren"
qCh_OpModePlc	Betriebsartvorgabe von SPS
qCh_RetCont	Wiederaufahren (an die Kontur)
qCh_FeedHold	Vorschub Halt

iCh_OpMode_00 bis iCh_OpMode_03	Aktive Betriebsart Bit 0 bis 3 (Rückmeldung für Betriebsart Anwahl Bit 0 bis 3)
iCh_ReadyReEnter	Bereit zum Wiederanfahren
iCh_ReEnterAct	Wiedereintritt aktiv
iCh_RemoveFinish	Wegfahren beendet

Tab. 5-1: Relevante IF-Signale

Relevante Maschinenparameter (MP)

OPF/ReturnPath/ Ch[k]/ReturnMode (7060 00310)	Betriebsart Wiederanfahren Voreinstellung des Anfahrmodus (Manuell, Automatik oder Einzelsatz)
OPF/ReturnPath/ Ch[k]/ReturnPoint (7060 00320)	Anfahrpunkt Wiederanfahren Voreinstellung des Wiederanfahrpunktes auf der Kontur (Satzanfang, Unterbrechungspunkt oder Satzende)
OPF/ReturnPath/ Ch[k]/NofJogSteps (7060 00330)	Maximale Aufzeichnung für Wiederanfahren in Sätzen. Legt fest, wie viele Verfahrbewegungen nach Aktivieren des Modus "Wegfahren von der Kontur" maximal aufgezeichnet werden können.

Tab. 5-2: Relevante Maschinenparameter

Relevante Systemdaten

Strukturelement	Beschreibung
FeedVal	Individuelle Wiederanfahrgeschwindigkeit
OptFeedVal	Optionale Geschwindigkeit falls noch kein Vorschub programmiert war
TangCircleRadius	Wiederanfahrradius
NameOfAsup	Name des Abgleichprogramms der Wiederanfahrlogik (ReentryAsup)
Approach	Wiederanfahrkontur
FeedMode	Wiederanfahrgeschwindigkeits-Auswahl
FeedHold	Aktivierung von Vorschubhalt nach Wiederanfahren an die Kontur
AsupActive	Abgleichprogramm der Wiederanfahrlogik (ReentryAsup) aktiv

Tab. 5-3: Übersicht Strukturelemente SysReentry

5.1.2 Optionen zum Wiederanfahren

Wahl des Anfahrpunktes

Verfügbare Optionen:

- Satzanfang
- Unterbrechungspunkt
- Satzende

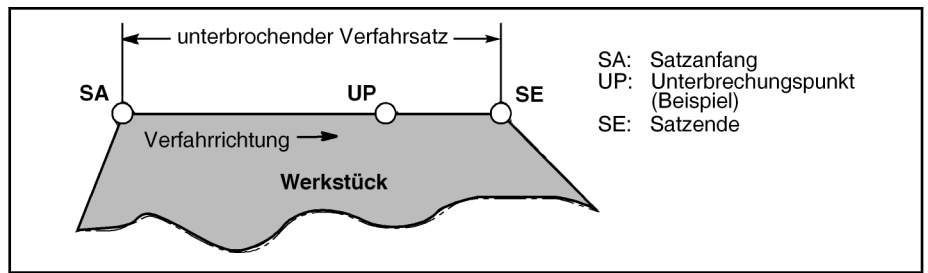


Abb. 5-1: Wahl des Anfahrpunktes

HINWEIS

Schäden an der Maschine oder am Werkstück möglich!

Bedenken Sie bei Nutzung der Option "Satzende", dass die Kontur ab dem Unterbrechungspunkt noch nicht bearbeitet wurde!

Wahl des Anfahrmodus

Verfügbare Optionen:

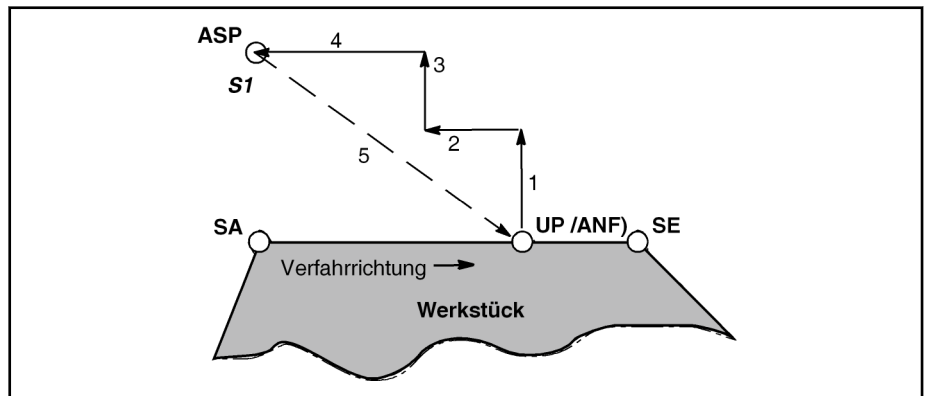
- Manuell
- Automatik
- Einzelsatz



Für die nachfolgend gezeigten Beispiel-Abbildungen gilt:

Wahl des Anfahrpunktes ist auf "Unterbrechungspunkt" eingestellt.

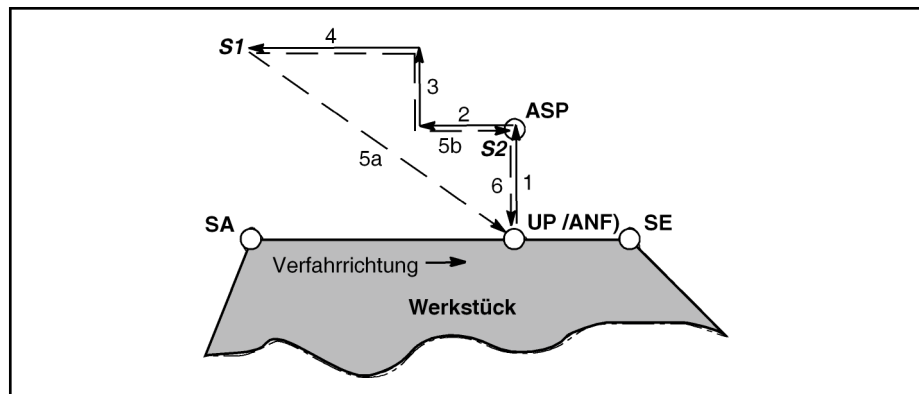
Manuell:



- 1 ... 4: Wegfahren durch Tippbewegungen. Die Position am Ende einer Tippbewegung wird immer zum (Neuen) Aufsetzpunkt.
- S1:** Auslösen von NC-Start
- Asup:** Falls ein asynchrones Unterprogramm in den Systemdaten angemeldet ist, wird es hier aufgerufen.
- 5:** Anfahrbewegung zum Anfahrpunkt (hier: Unterbrechungspunkt) und sofortige Weiterbearbeitung.
- ASP:** Aufsetzpunkt (Beispiel)
- SA:** Satzanfang
- UP:** Unterbrechungspunkt (Beispiel)
- ANF:** Anfahrpunkt
- SE:** Satzende

Abb. 5-2: Anfahrmodus - Manuell

Automatik:



1 ... 4: Wegfahren durch Tipfbewegungen

S1, S2: Auslösen von NC-Start

5a: Anfahrbewegung mit sofortiger Weiterbearbeitung am Anfahrpunkt (hier Unterbrechungspunkt), wenn das Kanal-Eingangssignal qCh_RetCont "Wiederanfahren" an die Kontur beim Auflösen von NC-Start (S1) nicht gesetzt ist. Ist ein asynchrones Unterprogramm in den Systemdaten angemeldet, wird es vor der Anfahrbewegung aufgerufen.

5b: Anfahrbewegung zum Aufsetzpunkt, wenn das Kanal-Eingangssignal qCh_RetCont "Wiederanfahren" an die Kontur beim Auslösen von NC-Start (S1) gesetzt ist. Per Tipfbewegung kann der Aufsetzpunkt danach noch verändert werden. Per NC-Start (S2) fährt die NC anschließend den Anfahrpunkt (hier: Unterbrechungspunkt) an und setzt die Bearbeitung fort. Ist ein asynchrones Unterprogramm in den Systemdaten angemeldet, wird es nach NC-Start (S2) vor der Anfahrbewegung aufgerufen.

ASP: Aufsetzpunkt (Beispiel)

SA: Satzanfang

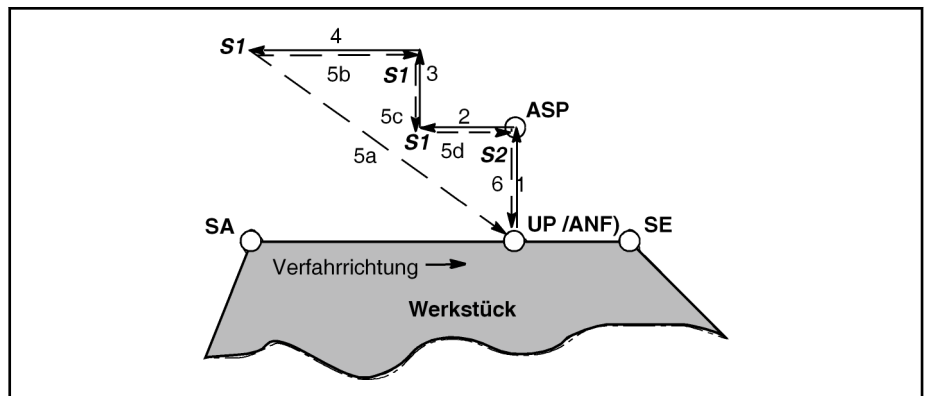
UP: Unterbrechungspunkt (Beispiel)

ANF: Anfahrpunkt

SE: Satzende

Abb. 5-3: Anfahrmodus - Automatik

Einzelsatz:



- 1 ... 4: Wegfahren durch Tippbewegungen
 S1, S2 : Auslösen von NC-Start
 5a: Anfahrbewegung mit sofortiger Weiterbearbeitung am Anfahrpunkt (hier Unterbrechungspunkt), wenn das Kanal-Eingangssignal qCh_RetCont "Wiederauffahren" an die Kontur beim Auslösen von NC-Start (S1) nicht gesetzt ist. Ist ein asynchrones Unterprogramm in den Systemdaten angemeldet, wird es vor der Anfahrbewegung aufgerufen.
 5b ... 5d: Separate Anfahrbewegung zum Aufsetzpunkt, wenn das Kanal-Eingangssignal qCh_RetCont "Wiederauffahren" an der Kontur beim Auslösen von NC-Start (S1) gesetzt ist. Sofern das Eingangssignal Wiederauffahren an die Kontur beim Auslösen von NC-Start (S1) nicht gesetzt ist, führt die NC den Anfahrpunkt direkt an und setzt die Bearbeitung fort. Per Tippbewegungen kann der Aufsetzpunkt (ASP) noch verändert werden. Per NC-Start (S2) führt die NC anschließend den Anfahrpunkt (hier: Unterbrechungspunkt) an und setzt die Bearbeitung fort. Ist ein asynchrones Unterprogramm in den Systemdaten angemeldet, wird es vor der Anfahrbewegung aufgerufen nach: - NC-Start (S1), wenn das Kanal-Eingangssignal qCh_RetCont "Wiederauffahren" an der Kontur nicht gesetzt ist. - NC-Start (S2)
 ASP: Aufsetzpunkt (Beispiel)
 SA: Satzanfang
 UP: Unterbrechungspunkt (Beispiel)
 ANF: Anfahrpunkt
 SE: Satzende

Abb. 5-4: Anfahrmodus - Einzelsatz

5.1.3 Handlungsanweisung: Weg- und Wiederauffahren an die Kontur

Applizieren:

HINWEIS**Schäden an der Maschine oder am Werkstück möglich!**

Während der Bearbeitung darf keine Systemgrundstellung ausgelöst werden.

1. MP 7060 00310, 7060 00320 und 7060 00330 entsprechend der gewünschten Standard-Wiederauffahrstrategie anpassen.
2. Systemgrundstellung auslösen.
3. Gegebenenfalls das Systemdatum /SysReentry gemäß gewünschter Wiederauffahrstrategie anpassen.

Aktivieren:

Kanalgrundstellung bricht einen aktivierten Modus ab!

Hierbei werden bereits gespeicherte Wegfahrbewegungen gelöscht.

Modus "Wegfahren von der Kontur per SPS" aktivieren:*Voraussetzungen:*

- Das aktive Teileprogramm ist durch "Vorschub Halt" unterbrochen.
- Das Kanal-IF-Signal iCh_RemoveFinish "Wegfahren beendet" ist nicht gesetzt.
- Das Kanal-IF-Signal qCh_RetCont "Wiederauffahren an die Kontur" ist nicht gesetzt.
- Das Kanal-IF-Signal qCh_OpModePlc "Betriebsart von SPS" ist gesetzt.

Über die Kanal-IF-Signale qCh_OpModeSel_00 bis qCh_OpModeSel_03 die Kanalbetriebsart "Wiederauffahren aktivieren".

Ab diesem Zeitpunkt "merkt" sich die Steuerung alle Verfahrbewegungen, die über die achsbezogenen IF-Eingangssignale qAx_JogPlus "Manuell+" bzw. qAx_JogMinus "Manuell-" ausgelöst werden. Bei Achsstillstand gibt die Steuerung das Kanal-IF-Signal iCh_ReadyReEnter "Bereit zum Wiederauffahren" aus.

Modus "Wiederauffahren per SPS" aktivieren:*Voraussetzungen:*

- Modus "Wegfahren von der Kontur per SPS" (siehe oben) wurde aktiviert.
- Positive Flanke am IF-Eingang qCh_RetCont "Wiederauffahren an die Kontur" geben.

Die NC setzt daraufhin das IF-Ausgangssignal iCh_RemoveFinish "Wegfahren beendet". Weitere, durch Tippen ausgelöste Verfahrbewegungen sind möglich, werden aber nicht mehr aufgezeichnet.

In Abhängigkeit der gewählten Anfahrstrategie (siehe [Kap. 5.1 "Weg- und Wiederauffahren" auf Seite 31](#)) kann das Wiederauffahren an die Kontur per "NC-Start" gestartet werden.

Die Funktion gestattet das Weg- und Wiederauffahren an die Kontur nach Unterbrechung eines Teileprogramms mit "Vorschub Halt" (NC-Stop). Das kann z. B. nach Werkzeugbruch, zur manuellen Werkstück-/Werkzeugvermessung oder für eine optische Kontrolle erforderlich sein.

IW-Engineering / Configuration: Modus zum Wiederauffahren konfigurieren:

- In den Parameter **ReturnMode** "Anfahrstrategie" (7060 00310) die gewünschte Anfahrstrategie festlegen.
- Im Parameter **ReturnPoint** "Anfahrpunkt" (7060 00320) den gewünschten Anfahrpunkt auswählen.
- In den Parameter **NofJogStep** "Max. Anzahl der aufzuzeichnenden Tippbewegungen" (7060 00330) die Anzahl der maximal möglichen Tippbewegungen festlegen, die beim Wegfahren aufgezeichnet werden können.



In Abhängigkeit der gewählten Anfahrstrategie entstehen mehrere Varianten zum Wiederauffahren. Nähere Information siehe [Kap. 5.1 "Weg- und Wiederauffahren" auf Seite 31](#).

	Documentation	Instruction
Instruction:		Editieren von Maschinenparametern
Documentation:	MTX Maschinenparameter	Wiederaufnahmepunkt definieren

IW-Engineering / IndraLogic: Wegfahren von der Kontur:

- "Vorschub Halt" (NC-Stop) auslösen, durch das Setzen des Interface-Signals **qCh_FeedHold = 1** "Vorschub Halt".
- Das Interface-Signal **qCh_RetCont** "Wiederaufnehmen an die Kontur" muss logisch "0" gesetzt sein.
- Das Interface-Signal **qCh_OpModePlc** "Betriebsart von SPS" ist auf logisch "1" zu setzen.
qCh_OpModePlc = 1:
Die Betriebsart wird von der SPS über **qCh_OpModeSel_00...03** vorgegeben. Es kann über die NC-Bedienoberfläche keine Betriebsart ausgewählt werden.
- Über die Interface-Signale **qCh_OpModeSel_00** bis **qCh_OpModeSel_03** ist die Kanalbetriebsart "Wiederaufnehmen" mit der Kodierung **0b1011** zu aktivieren.



Das Interface-Signal **iCh_RemoveFinish** "Wegfahren beendet" ist nicht gesetzt.

Betriebsart	Anwahl Betriebsart			
	Bit 3 (qCh_OpModeSel_03)	Bit 2 (qCh_OpModeSel_02)	Bit 1 (qCh_OpModeSel_01)	Bit 0 (qCh_OpModeSel_00)
Wiederaufnehmen	1	0	1	1

	Documentation	Instruction
Instruction:		SPS-Signal editieren
Documentation:	MTX SPS-Interface	Wegfahren von der Kontur

IW-Engineering / IndraLogic: Wegfahren von der Kontur kontrollieren bzw. beenden

Das Wegfahren von der Kontur und somit die Aufzeichnung der Punkte ist beendet, wenn das Interface-Signal **qCh_RetCont** "Wiederaufnehmen" gesetzt ist.

- oder -

ein Pegelwechsel von 0 auf 1 am Interface-Signal **qCh_RetCont** "Wiederaufnehmen" die Aufzeichnung der Wegfahrbewegung beendet.



- Ist das Wegfahren abgeschlossen, setzt die NC daraufhin das Interface-Signal **iCh_RemoveFinish** "Wegfahren beendet".
- Weitere, durch Tippen ausgelöste, Verfahrbewegungen sind möglich, werden aber nicht mehr aufgezeichnet.

	Documentation	Instruction
Instruction:		SPS-Signal editieren
Documentation:	MTX SPS-Interface	Wegfahren von der Kontur kontrollieren

IW-Engineering / IndraLogic: Wiederanfahren der Kontur

- Der Modus "Wegfahren von der Kontur per SPS" wurde durchgeführt.
- Das Setzen des Interface-Signals **qCh_RetCont** "Wiederanfahren an die Kontur" wird wie folgt interpretiert:
qCh_RetCont = 1:
Nach "NC-Start" wird auf der gespeicherten Wegfahrbewegung an die Kontur zurückgefahren.
qCh_RetCont = 0:
Nach "NC-Start" wird die Bearbeitung direkt fortgesetzt.
- In Abhängigkeit der gewählten Anfahrtsstrategie entstehen mehrere Varianten zum Wiederanfahren. Nähere Information siehe [Kap. 5.1 "Weg- und Wiederanfahren"](#) auf Seite 31.



Das Interface-Signal **iCh_ReadyReEnter** "Bereit zum Wiederanfahren" ist auf logisch "1", wenn das System bereit zum Wiederanfahren ist.

	Documentation	Instruction
Instruction:		SPS-Signal editieren
Documentation:	MTX SPS-Interface	Wegfahren von der Kontur

5.2 Satzvorlauf

5.2.1 Allgemeines zum Satzvorlauf

Begriffsdefinition

Satzvorlauf

Bei der Bearbeitung eines Werkstücks kann es zu Situationen kommen (Werkzeugbruch, Peripheriefehler usw.), die einen Programmabbruch und Neustart des NC-Programms zur Folge haben. In den meisten Fällen wird das Werkstück durch die Bearbeitungsunterbrechung nicht wesentlich beschädigt, die Bearbeitung kann nach Behebung der Unterbrechungsursache fortgesetzt werden.

Ohne die Funktion "Satzvorlauf" müsste die gesamte Bearbeitung wieder von vorne beginnen, d. h. der schon abgearbeitete Programmteil würde unnötigerweise ein weiteres mal verfahren. Um Bearbeitungszeit zu sparen, wählt der Maschinenbediener einen beliebigen NC-Satz in der Nähe der Abbruchstelle als Zielsatz aus (kann auch in einem Unterprogramm sein), um die Bearbeitung an dieser Stelle wieder zu beginnen.

Das CNC-System MTX berechnet das gesamte NC-Programm vom Programmanfang bis zu diesem Zielsatz neu (Rechenlauf), ohne Achsbewegungen oder Hilfsfunktionsausgaben zu generieren. Allerdings wird der logische Zustand der Steuerung so modifiziert, wie er bei regulärem Programmablauf am Zielsatz auch gewesen wäre. So sind z. B. die modalen NC-Funktionen, die programmierten Geschwindigkeiten und Drehzahlen richtig gesetzt, so dass die Bearbeitung am ausgewählten Zielsatz fortgesetzt werden kann.

Rechenlauf

Die Abarbeitung des NC-Programms vom Startsatz bis zum ausgewählten Zielsatz ohne die Generierung von Achsbewegungen oder Hilfsfunktionsausgaben wird Rechenlauf genannt. Im NC-Programm kann abgefragt werden, ob gerade eine echte Programmabarbeitung oder ein reiner Rechenlauf stattfindet. Der Rechenlauf endet mit dem ausgewählten Zielsatz. Dadurch werden die korrekten Bedingungen hergestellt, um das unterbrochene Programm mit dem Zielsatz starten zu können.

Abgleich

Nach Erreichen des Zielsatzes muss sichergestellt werden, dass der physikalische Zustand der Maschine dem logischen Zustand der Steuerung entspricht. Die Achsen müssen von ihren momentanen Istpositionen auf die im Satzvorlauf berechneten Sollpositionen des NC-Programms verfahren werden. Gleichzeitig müssen auch die Spindeln wieder ihre zuletzt beauftragten Drehzahlen, Drehrichtungen und Zustände erreichen. Alle modalen NC-Funktionen müssen wieder hergestellt werden. Dieser Abgleich der Zustände wird teilweise automatisch von der NC, sowie durch Ausführung spezieller Abgleichprogramme durchgeführt. Es ist auch möglich, einzelne Abgleichfunktionen durch direkte Handeingabe auszulösen. Spezielle Abgleichbildschirme liefern dem Bediener die Information, welche Abgleichaktionen notwendig sind. Anschließend wird dann das NC-Programm am Zielsatz so fortgesetzt, als wäre es ohne Satzvorlauf gestartet worden.

Bedienung

In der Standardapplikation wird der Satzvorlauf über einen M-Key aktiviert.

Kanalsteuerung

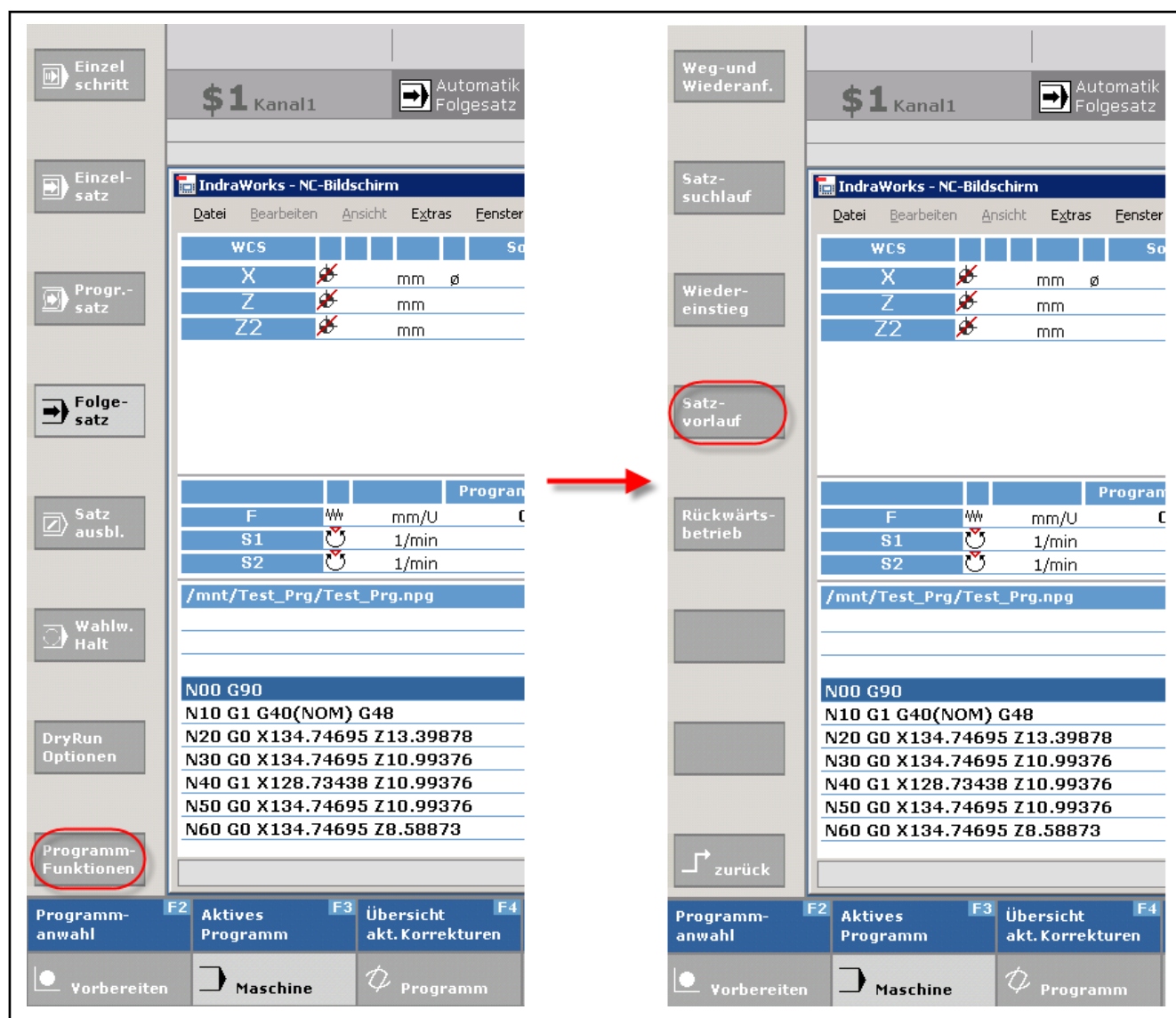


Abb. 5-5: Satzvorlauf aktivieren

Weitere Informationen zu diesem Thema finden Sie in der Dokumentation "MTX xxVRS Standard-NC-Bedienung" (DOK-MTX***-NC*OP***Vxx-APRS-DE-P), Kapitel "Satzsuchlauf/Satzvorlauf".

Abgrenzung zum Satzsuchlauf

Satzsuchlauf

Der Satzsuchlauf erlaubt dem Bediener, sein angewähltes Teileprogramm ab einem von ihm definierten Programmsatz zu starten. Das angewählte Programm muss auf Programmanfang stehen, damit der Bediener den gewünschten Zielsatz auswählen kann. Diese Funktion wirkt wie ein absoluter Sprung vom Programmanfang zum Zielsatz. Das bedeutet, alle dazwischen liegenden Anweisungen werden ignoriert.

Mit der Auslösung von "NC-Start" beginnt ab diesem Satz die Programmabarbeitung auf Basis der aktuellen Maschinen- und Steuerungszustände. Alle vorausgehenden Sätze werden nicht ausgeführt und inhaltlich nicht berücksichtigt.

Falls zum Eintrittssatz Bedingungen an der Maschine oder zum Ablauf bereit stehen müssen, so können diese Einstellungen zuvor unter dem "MDI-Betrieb" geschaffen werden.

Abgrenzung des Satzsuchlaufs zum Satzvorlauf

Die Funktionalität "Satzvorlauf" durchläuft rechnerisch das komplette Teileprogramm, einschließlich seiner Unterprogramme, vom Anfang bis zum Zielsatz und berücksichtigt alle bis dahin programmierten an- bzw. abgewählten modalen Bedingungen, Korrekturen und Systemeinstellungen. Wie im Satzsuchlauf erfolgen keine Achsbewegungen und keine Hilfsfunktionsausgaben.

5.2.2 Beschreibung

Funktion

Nach der Anwahl eines Teileprogramms besteht die Möglichkeit, über den M-Key "Abgleichfunktionen" einen Satzvorlauf auszuführen.

Zu diesem Thema siehe auch Dokumentation "MTX Standard-NC-Bedienung", Kapitel "Satzsuchlauf/Satzvorlauf".

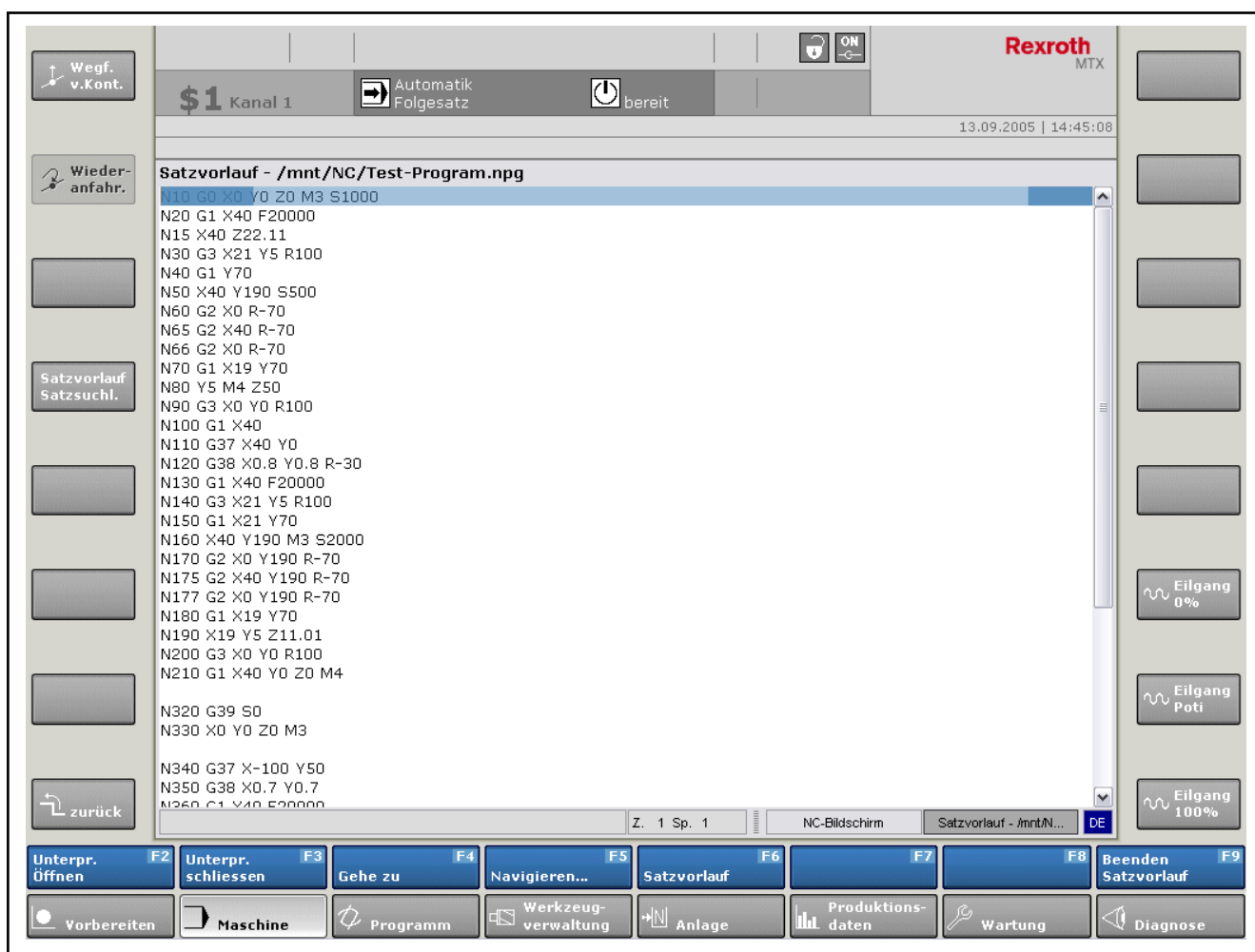


Abb. 5-6: Bildschirmlayout Satzsuchlauf/Satzvorlauf

Beim Satzvorlauf wird zunächst ein Zielsatz angegeben. Der Zielsatz kann im Hauptprogramm oder in Unterprogrammen liegen. Nach anschließendem Auslösen der Funktion "Satzvorlauf" erfolgt eine Berechnung der NC-Sätze des Teileprogramms bis zum Satz "Zielsatz-1", wobei jedoch **keine** Verfahrbewegungen ausgeführt werden. Der Zielsatz-1 wird auch als "Abbruchsatz" bezeichnet. Während des Satzvorlaufs findet keine Interpolation statt, es wird lediglich ein Rechenlauf durchgeführt. Verweilzeiten, WAIT-Funktionen mit Parametern, NC-Synchronisationsfunktionen und Hilfsfunktionen werden während des Satzvorlaufs überlesen, hingegen findet die Sercos Kommunikation (Schreiben von Sercos Id-Nummern) statt.

Als Zielsatz kann über die Bedienoberfläche ein beliebiger NC-Satz oder CPL-Satz ausgewählt werden.

Nach einem erfolgten Rechenlauf kann Satzvorlauf fortgesetzt werden, wobei der Zielsatz nicht verändert werden muss. Hiermit besteht die Möglichkeit, innerhalb von mehrfach zu durchlaufenden Programmpassagen (Schleifen) einen definierten Umlauf auszuwählen.

Nach Satzvorlauf steht der Kanalzustand auf "Bereit". Ein NC-Start führt dazu, dass die Achsen des Kanals in einer linearen Anfahrbewegung von ihrer aktuellen Position auf die berechnete Start- oder Zielposition des Zielsatzes verfahren und daran anschließend das Teileprogramm, beginnend mit dem Zielsatz, bearbeitet wird. Die Auswahl des Start- oder des Endpunktes wird durch ein vom Teileprogramm beschreibbares Systemdatum festgelegt. Die Wahl, d. h. das Systemdatum muss vor Beginn des Abgleichs feststehen.

Optional besteht die Möglichkeit, dass nach NC-Start ein "Initialisierungs-Abgleichprogramm" (InitAsup) ausgeführt wird. Das InitAsup wird in den Systemdaten angemeldet. Der Pfad kann in SysSRun.NameOfInitAdjPrg eingetragen werden.

Das InitAsup setzt auf den Zustand auf, der vor der Programmanwahl aktiv war. In ihm kann schrittweise auf den Zustand vor dem Zielsatz umgeschaltet werden. Dies betrifft vor allem Achs- und Spindelkonfiguration, Koordinatentransformation oder modale Zustände im Kanal. Am Ende des InitAsups können zusätzlich ein oder mehrere Handeingabesätze abgearbeitet werden. Dazu ist der entsprechende Menüpunkt in der Bedienoberfläche auszuwählen. Am Ende des Abgleichs wird auf den Zustand vor dem Zielsatz zurückgeschaltet.

Optional besteht zusätzlich die Möglichkeit, dass vor der Anfahrbewegung ein "Aktionssatz" und/oder ein (asynchrones) Unterprogramm als "Abgleichprogramm" ausgeführt werden.

5.2.3 Vorgehensweise beim Satzvorlauf

Aktivieren des Satzvorlaufes

Nach einer Programmanwahl kann der Satzvorlauf über M-Keys aufgerufen werden. Nach Markierung des Zielsatzes erfolgt mit der Funktionstaste "Satz übernehmen" die Zielsatzübernahme, die gleichzeitig den Rechenlauf startet ...

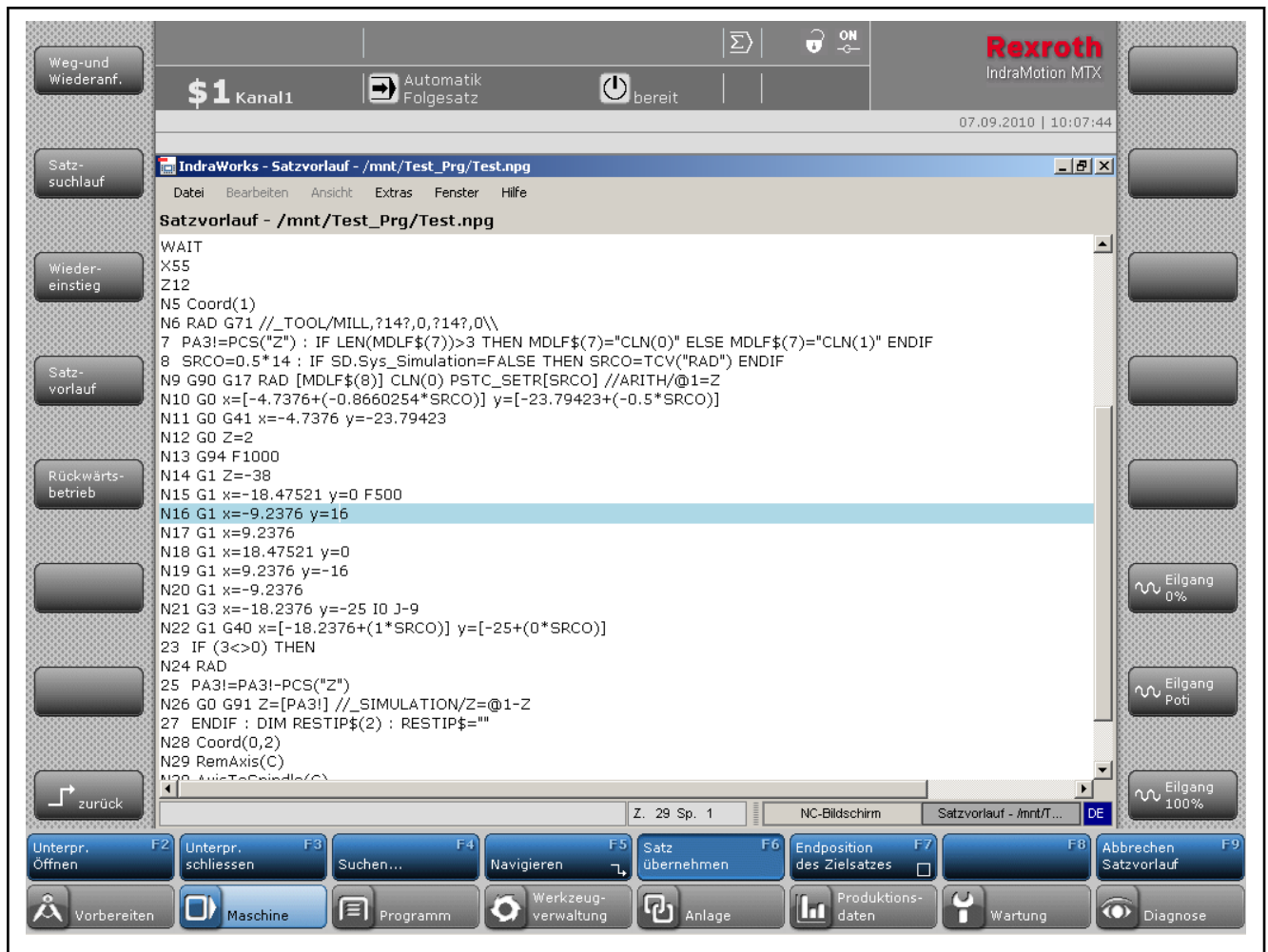


Abb. 5-7: Satz übernehmen

...und nach fehlerfreiem Durchlauf in das Grundbild "Maschine" wechselt. Aufgrund individueller Konfigurationen ist häufig für einzelne Bereiche ein zusätzlicher Abgleich vor der eigentlichen Anfahrbewegung erforderlich. Dieser Abgleich wird durch zusätzliche Symbolik in der Ansicht dargestellt.

Kanalsteuerung

1 Kanal1 Automatik Satzvorlauf bereit 07.09.2010 | 10:08:49

IndraWorks - NC-Bildschirm

WCS			Sollposition	Endposition	Restweg	Programm
X	mm	ø	0.000	0.000	0.000	0.000
Z	mm		0.000	0.000	0.000	0.000
Z2	mm		0.000	0.000	0.000	0.000

		Programm	Sollwert	Istwert	Override	Max	Getr.St.
F	mm/U	0.00	0.00	0.00	100%	120%	
S1	1/min	0.0	0.0	0.0	100%	120%	1
S2	1/min	0.0	0.0	0.0	100%	120%	1
S3	1/min	0.0	0.0	0.0	100%	120%	1

/mnt/Test_Prg/Test.npg

N13 G94 F1000
 N14 G1 Z=-38
 N15 G1 x=-18.47521 y=0 F500
 N16 G1 x=-9.2376 y=16
 N17 G1 x=9.2376
 N18 G1 x=18.47521 y=0
 N19 G1 x=9.2376 y=-16
 N20 G1 x=-9.2376
 N21 G3 x=-18.2376 y=-25 I0 J-9
 N22 G1 G40 x=[-18.2376+(1*SRCO)] y=[-25+(0*S...

WZ-Aktiv	akt. Schn.	D-Nummer
8	ED0	D0

NC-Bildschirm DE

Programmanwahl F2 Aktives Programm F3 Abgleichzustände F4 Nullpunktverschiebung F5 Aktive D-Korrekturen F6 Aktive Variablen F7 weitere Funktionen F8 F9

Vorbereiten Maschine Programm Werkzeugverwaltung Anlage Produktionsdaten Wartung Diagnose

Abb. 5-8: Abgleichbildschirm



Mit den linken M-Keys können die einzelnen Abgleichgruppen (Achsen, Spindeln ...) näher betrachtet werden.

Ein "NC-Start" löst in diesem Zustand den Satzvorlauf aus, der vor der Anfahrbewegung optional mit dem Abgleich beginnt.

Vorgänge beim Rechenlauf

Schematische Darstellung

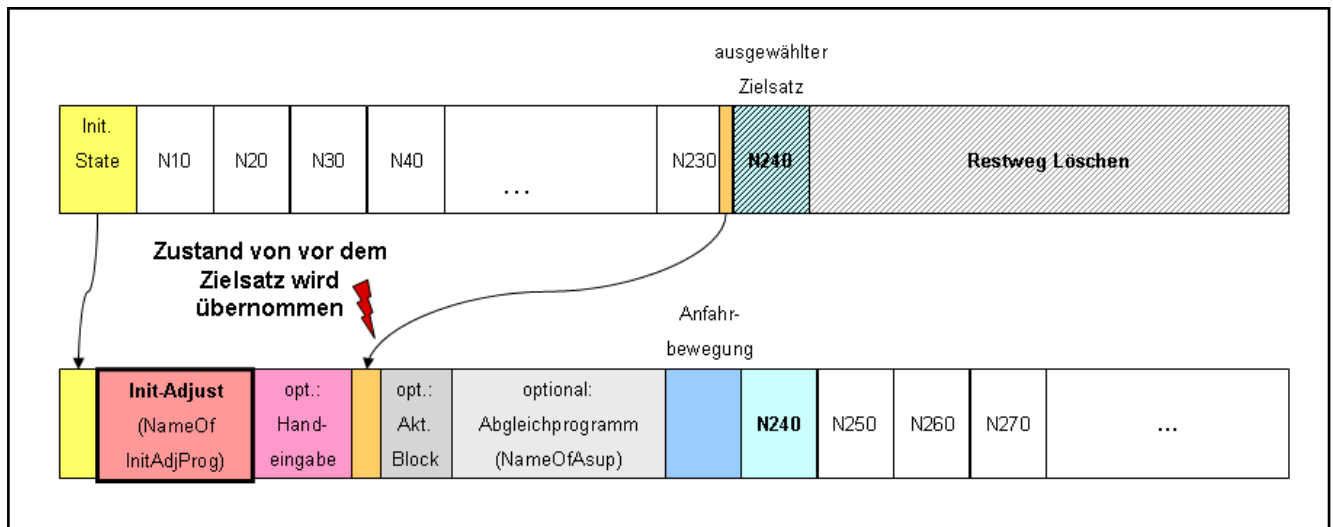


Abb. 5-9: Abgleich Satzvorgang

Ablauf

Der Rechenlauf in der Betriebsart Satzvorgang ist definiert durch seine Vorgaben in den Systemdaten (siehe [Kapitel 5.2.4 Projektierung des Satzvorgangs](#), Seite 48). Hier berechnet sich die steuerungsinterne Satzvorbereitung neu und vollzieht zunächst ein "Restweg löschen". Im Anschluss werden die Abgleichbedingungen entsprechend den projizierten Systemdaten ausgeführt, bevor es letztlich zur Anfahrbewegung auf den Zielsatz kommt.

Die Umsetzung der Abgleichbedingungen erfolgt in zwei Schritten:

- 1. Programmbedingungen (Zustände) vor bzw. im Zielsatz erzeugen
- 2. optional ergänzende Bedingungen hinzufügen

Beginnend mit dem "Initialisierungs-Status" und dem Abgleichprogramm "InitAdjustPrg" werden die Grundbedingungen aller Programmsätze bis zum Zielsatz erfüllt. Hierzu zählt insbesondere die Achs-, Koordinaten- und Spindelkonfiguration. Darüber hinaus sind die NC-Funktionen, Werkzeuge und ggf. Hilfsfunktionen abzugleichen. Nach dem InitAdjustPrg besteht noch die Möglichkeit spezielle Funktionen und Zustände über ein optionales Systemdatum "Handeingabe" manuell abzugleichen. Wird diese Option genutzt, ist an dessen Ende ein weiteres "NC-Start" zu setzen, um den Abgleichvorgang im Satzvorgang fortzusetzen.

Nach Abschluss dieses ersten Schrittes beim Abgleichvorgang erfolgt eine Überprüfung, ob die Achskonfiguration mit dem Zielsatzzustand abgeglichen ist.

Der jetzt abgespeicherte Zustand des Beginns des Zielsatzes wird in der steuerungsinternen Satzvorbereitung wieder eingesetzt. Könnten einzelne Zustände nicht abgeglichen werden, kann es zu Fehlermeldungen oder - im Extremfall - zu einem Fehlverhalten an der Maschine kommen.

Im zweiten Schritt können optional in einem Aktionssatz und/oder einem zusätzlichen Abgleichprogramm „TargetAdjustPrg“ weitere Aktionen wie z. B. Hilfsfunktionen aktiviert werden. Als Abschluss des Abgleichvorgangs im Satzvorgang kommt es dann zur Anfahrbewegung, also zum automatischen Anfahren der Achsen auf ihre Zielposition (Positionsabgleich). Hierbei werden einige modale NC-Funktionsgruppen automatisch abgeglichen, falls diese im Aktionssatz oder unter "TargetAdjustPrg" verändert wurden.

⚠ VORSICHT

Alle abzugleichenden Achsen werden gemeinsam verfahren

Falls dies nicht möglich ist, muss der Achsabgleich im TargetAdjustPrg über G77 manuell hergestellt werden, siehe Beispiel.

5.2.4 Projektierung des Satzvorlaufs

Systemdaten

Bedeutung

Die Parametrierung des Satzvorlaufes wird mit Hilfe von Systemdaten durchgeführt. Alle Einstellungen sowie Statusabfragen werden in den Systemdatenstrukturen abgelegt.

In dieser Dokumentation werden nur die wichtigsten, für den Satzvorlauf relevanten Systemdaten näher beschrieben.

Übersicht der Systemdaten für den Satzvorlauf

Weitere Informationen zu diesem Thema siehe auch Dokumentation "MTX Maschinenparameter", Kapitel "Systemdaten".

Satzvorlaufdaten für Achsen "/
SysAxSRun"

Strukturelement	Beschreibung
NotAdjusted	Achse nicht abgeglichen
PosNotAdjusted	Achsposition nicht abgeglichen
ConfNotAdjusted	Kanalzuordnung nicht abgeglichen
ReqPosAdjust	Positionsabgleich erforderlich
TargChan	Zielkanal
ReleaseChan	Abgabekanal

Tab. 5-4: Übersicht Strukturelemente SysAxSRun

Spindel-Satzvorlaufdaten "/
SysChSpSRun"

Die Systemdatenstruktur "SysChSpSRun" wird für jeden Kanal im flüchtigen Speicher angelegt.

Diese Systemdatenstruktur setzt sich aus 3 Teilen zusammen:

Allgemeine Daten (SysChSpSRun):

Strukturelement	Beschreibung
MaxSpNumber	Höchste im Kanal vorhandene Spindelnummer
PrgSpNumber	Höchste im Vorlauf programmierte Spindelnummer
PrgCpNumber	Höchste im Vorlauf programmierte Kopplungsnummer
ConstCutStatus	G-Funktionsgruppe G96/97 abgleichen
SpeedStr	Drehzahlsyntax einer Spindel
MoveStr	Bewegungssyntax einer Spindel
GearStr	Getriebestufensyntax einer Spindel
SpConfNotAdjusted	Abgleich der Spindelkonfiguration ist erforderlich

Tab. 5-5: Übersicht Strukturelemente SysChSpSRun (allgemeine Daten)

Daten für Kanalspindeln (SysChSpSRun.Sp[<Kanalspindelnummer>]):

Strukturelement	Beschreibung
SysSpNo	Nummer der Systemspindel
DriveNo	Nummer des Antriebs
IsProg	Spindel im Vorlauf programmiert
NotAdjusted	Abgleich erforderlich
MoveNotAdj	Abgleichstatus der Bewegungsfunktion
GearNotAdj	Abgleichstatus der Getriebestufe
SpeedNotAdj	Abgleichstatus der Drehzahl
LimitNotAdj	Abgleichstatus der Drehzahlbegrenzung
CcvNotAdj	Abgleichstatus des G96-Zustands
CAxisNotAdj	Abgleichstatus der C-Achse erforderlich
OpModeNotAdj	Abgleichstatus der Antriebsbetriebsart
CpDataNotAdj	Abgleichstatus der Kopplungsdaten
CoupleNotAdj	Abgleichstatus des Kopplungszustands
Move	Bewegungsfunktion
Gear	Getriebestufe
AutoGearShift	Automatischer Getriebestufenwechsel
CAxisOn	C-Achse einschalten
CAxisType	C-Achs-Typ
CcvOn	Konstante Schnittgeschwindigkeit einschalten
CcvAxis	Kanal-Bezugsachse für konstante Schnittgeschwindigkeit
CcvCoordSystem	Koordinatensystem für konstante Schnittgeschwindigkeit
PosCtrlOn	Lageschnittstelle einschalten
OriPos	Richtposition
CcvVelocity	Schnittgeschwindigkeit
ProgSpeed	Programmierte Drehzahl
ProgMinSpeed	Untere Grenzdrehzahl
ProgMaxSpeed	Obere Grenzdrehzahl
CoupleDistance	Koppelabstand
SyncWindow	Synchronlauffenster für Kopplung
SyncErrorWindow	Synchronlauffehlerfenster für Kopplung
CouplePosOffset	Winkelversatz bei aktiver Kopplung
CouplePosOffsetSpeed	Drehzahl zum Einfahren des Winkelversatzes

Tab. 5-6: Übersicht Strukturelemente SysChSpSRun (Daten für Kanalspindeln)

Daten für Spindelkopplungen (SysChSpSRun.Cp[<Kopplungsnummer>]):

Kanalsteuerung

Strukturelement	Beschreibung
IsProg	Koppelverband im Vorlauf programmiert
NotAdjusted	Abgleichstatus der Spindelkopplung
CoupleOn	Koppelverband einschalten
Master	Kanalspindelnummer der Master-Spindel
Slave[1..8]	Kanalspindelnummer der Slave-Spindel

Tab. 5-7: Übersicht Strukturelemente SysChSpSRun (Daten für Spindelkopplungen)

Kanalspezifische Satzvorlaufdaten
"/SysChSRun"

Strukturelement	Beschreibung
Prep/ChAxNames[1..8]	Kanalachsamen in der Satzvorbereitung
Act/ChAxNames[1..8]	Aktive Kanalachsamen im Abgleich
Targ/ChAxNames[1..8]	Kanalachsamen im Zielsatz
PosAdjust[1..8]	Achsamen für Positionsabgleich
TargPos[1..8]	Zielposition des Anfahrsetzes
AcsTargPos[1..8]	Zielposition im Achskoordinatensystem
ActAxConf[1..8]	Aktuelle Achskonfiguration im Abgleich
PrepAxConf[1..8]	Achskonfiguration in der Satzvorbereitung
TargAxConf[1..8]	Achskonfiguration im Zielsatz
TechAxAdjust	Technologischer Achsabgleich
Active	Rechenlauf aktiv
ActMode	Satzvorlauf-Modus
TargAdjPrgActive	Zielsatz-Abgleichprogramm (TargAdjPrg) aktiv
InitAdjPrgActive	Init-Abgleichprogramm (InitAdjPrg) aktiv
LookAheadPreRun	Satzvorlauf nach Zielsatz im Look-Ahead-Bereich aktiv
ChanNotAdjusted	Kanal nicht abgeglichen
NcfNotAdjusted	Modale NC-Funktionen nicht abgeglichen
AuxNotAdjusted	Modale Hilfsfunktionen nicht abgeglichen
AxConfNotAdjusted	Achskonfiguration nicht abgeglichen
CoordNotAdjusted	Achstransformationen nicht abgeglichen
ActStatus	Aktiver Abgleichzustand
TargBlockSrc	Zielsatz-Code
ReentryInfoDir	Dateipfad zu den Wiedereinstiegsdaten
TargetBlkAvailable	Gesicherter Zielsatz vorhanden
LockTargBlock	Zielsatz gesperrt

Tab. 5-8: Übersicht Strukturelemente SysChSRun

Daten der Wiederanfahrlogik "/
SysReentry"

Strukturelement	Beschreibung
FeedVal	Individuelle Wiederanfahrsgeschwindigkeit
OptFeedVal	Optionale Geschwindigkeit falls noch kein Vorschub programmiert war
TangCircleRadius	Wiederanfahradius
NameOfAsup	Name des Abgleichprogramms der Wiederanfahrlogik (ReentryAsup)
Approach	Wiederanfahrkontur
FeedMode	Wiederanfahrsgeschwindigkeits-Auswahl
AsupActive	Abgleichprogramm der Wiederanfahrlogik (ReentryAsup) aktiv

Tab. 5-9: Übersicht Strukturelemente SysReentry

Allgemeine Satzvorlaufdaten "/ SysSRun"

Strukturelement	Beschreibung
Active	Siehe: SysChSRun.Active
AsupActive	Abgleichprogramm aktiv
LookAheadPreRun	Siehe: SysChSRun.LookAheadPreRun
Reserved01	Reserviert
TargPos	Siehe: SysChSRun.TargPos
TargBlockSrc	Siehe: SysChSRun.TargBlockSrc
ReEntryIndex	Zuletzt benutzter Ablageindex der Abbruchinformation
LockTargBlock	Siehe: SysChSRun.LockTargBlock
LockWarning	Hinweistext bei gesperrtem Zielsatz
NameOfAsup	Name des Zielsatzabgleichprogramms (Asup)
NameOfInitAdjPrg	Name des Init-Abgleichprogramms
ActionBlock	Aktionssatz
Timeout	Laufzeitüberwachung für Satzvorlauf
RetPaStrategy	Anfahrstrategie nach Satzvorlauf
TargetBlkCheck	Gesicherter Unterbrechungspunkt ist vorhanden
ManWaitForStart	Handeingabe nach Satzvorlauf
SuppressMsg	Unterdrückung von Nachrichten im Satzvorlauf
ReEntryStore	Sichern der abgebrochenen Informationen für einen späteren Wiedereinstieg

Tab. 5-10: Übersicht Strukturelemente SysSRun

Aktives und vorgewähltes Werkzeug bei Satzvorlauf "/SysToolS- Run"

Strukturelement	Beschreibung
ActTool.K1	Werkzeugdatenelement K1 des aktiven Werkzeugs
ActTool.K2	Werkzeugdatenelement K2 des aktiven Werkzeugs

Strukturelement	Beschreibung
ActTool.SKQ	Werkzeugdatenelement SKQ des aktiven Werkzeugs
ActTool.IKQ1	Werkzeugdatenelement IKQ1 des aktiven Werkzeugs
ActTool.IKQ2	Werkzeugdatenelement IKQ2 des aktiven Werkzeugs
ActTool.IKQ3	Werkzeugdatenelement IKQ3 des aktiven Werkzeugs
ActTool.IQ1	Werkzeugdatenelement IQ1 des aktiven Werkzeugs
ActTool.IQ2	Werkzeugdatenelement IQ2 des aktiven Werkzeugs
ActTool.IQ3	Werkzeugdatenelement IQ3 des aktiven Werkzeugs
ActTool.BQ1	Werkzeugdatenelement BQ1 des aktiven Werkzeugs
ActTool.BQ2	Werkzeugdatenelement BQ2 des aktiven Werkzeugs
ActTool.BQ3	Werkzeugdatenelement BQ3 des aktiven Werkzeugs
PreTool.K1	Werkzeugdatenelement K1 des vorbereiteten Werkzeugs
PreTool.K2	Werkzeugdatenelement K2 des vorbereiteten Werkzeugs
PreTool.SKQ	Werkzeugdatenelement SKQ des vorbereiteten Werkzeugs
PreTool.IKQ1	Werkzeugdatenelement IKQ1 des vorbereiteten Werkzeugs
PreTool.IKQ2	Werkzeugdatenelement IKQ2 des vorbereiteten Werkzeugs
PreTool.IKQ3	Werkzeugdatenelement IKQ3 des vorbereiteten Werkzeugs
PreTool.IQ1	Werkzeugdatenelement IQ1 des vorbereiteten Werkzeugs
PreTool.IQ2	Werkzeugdatenelement IQ2 des vorbereiteten Werkzeugs
PreTool.IQ3	Werkzeugdatenelement IQ3 des vorbereiteten Werkzeugs
PreTool.BQ1	Werkzeugdatenelement BQ1 des vorbereiteten Werkzeugs
PreTool.BQ2	Werkzeugdatenelement BQ2 des vorbereiteten Werkzeugs
PreTool.BQ3	Werkzeugdatenelement BQ3 des vorbereiteten Werkzeugs
TargEd	Nummer der aktiven Schneide im Zielsatz
TargDNr	Nummer der aktiven D-Korrektur im Zielsatz
TCorrNotAdj	Werkzeugkorrektur nicht abgeglichen

Tab. 5-11: Übersicht Strukturelemente SysToolSRun

Wichtige Systemdaten für die Projektierung des Satzvorlaufes

Weitere Informationen zu diesem Thema siehe auch Dokumentation "MTX Maschinenparameter", Kapitel "Systemdaten".

Rechenlauf aktiv	Active

Liefert kanalspezifisch die Information darüber, dass im Kanal der Rechenlauf von Satzvorlauf aktiv ist.

Durch Abfrage im Programm können NC- oder CPL-Teile (wie z. B. ein Werkzeugwechsel oder Änderungen in der Datenbank) während des Rechenlaufs ausgeblendet werden.

Pfad	/SysChSRun[<Channel>]/Active	
Bedeutung der Parameterwerte	0	Satzvorlauf ist nicht aktiv.
	1	Satzvorlauf ist aktiv.

Zielsätze sperren	LockTargBlock

Ermöglicht es im jeweiligen Kanal, im Teileprogramm Bereiche für die Zielsatzauswahl zu sperren.

Im Teileprogramm können damit Bereiche gesperrt werden, in denen Zielsätze nicht erlaubt sind. Bei gesetzter Sperre wird der Zielsatz zwar akzeptiert, NC-Start aber nicht angenommen. Es wird in diesem Fall entweder eine allgemeine Warnung oder - falls vorhanden - ein individueller Hinweistext (siehe Systemdatum **/SysSRun/LockWarning**) ausgegeben. Das Fortsetzen des Satzvorlaufs auf einem anderen Zielsatz ist möglich.

Das Datum wird typischer Weise im NC-Programm geschrieben.

Pfad	/SysChSRun[<Channel>]/LockTargBlock	
Bedeutung der Parameterwerte	0	Zielsätze sind nicht gesperrt.
	1	Zielsätze sind gesperrt.

Individuelle Wiederanfahrsgeschwindigkeit	FeedVal

Gibt eine individuelle Wiederanfahrsgeschwindigkeit vor. Diese wird wirksam, wenn in "/SysReentry/FeedMode" der Wert 2 eingetragen ist.

Der Wert kann im Teileprogramm vorgegeben werden.

Pfad	/SysReentry/FeedVal
Bedeutung der Variablenwerte	Anfahrsgeschwindigkeit

Optionale Wiederanfahrsgeschwindigkeit	OptFeedVal

Gibt eine optionale Wiederanfahrsgeschwindigkeit vor. Diese wird wirksam, wenn in "/SysReentry/FeedMode" der Wert 1 eingetragen ist jedoch im Unterbrechungssatz kein gültiger F-Wert wirksam ist, oder aber Zeitprogrammierung (G93) aktiv ist.

Ist für die optionale Wiederanfahrsgeschwindigkeit kein gültiger Wert parametrisiert, wird als Notfallstrategie die **Tippgeschwindigkeit langsam** der jeweils ersten Achse im Kanal als optionale Wiederanfahrsgeschwindigkeit verwendet.

Es wird empfohlen, einen geeigneten Geschwindigkeitswert einzutragen. Der Wert kann auch im Teileprogramm programmiert werden.

Pfad	/SysReentry/OptFeedVal
Bedeutung der Variablenwerte	Anfahrsgeschwindigkeit

Wiederanfahrsgeschwindigkeits-Auswahl	FeedMode

Legt fest, auf welche Art die Anfahrsgeschwindigkeit ermittelt wird.

Der Wert kann im Teileprogramm vorgegeben werden.

Pfad	/SysReentry/FeedMode	
Bedeutung der Variablenwerte	0	Anfahrsgeschwindigkeit = Tippgeschwindigkeit
	1	Anfahrsgeschwindigkeit = aktiver Vorschub im Unterbrechungssatz (programmierter Vorschub)

2 Anfahrgeschwindigkeit = Wert aus "/SysReentry/FeedVal"

Hinweistext bei gesperrtem Zielsatz	LockWarning

Definiert kanalspezifisch einen individuellen Hinweistext bei gesperrtem Zielsatz.

Wird hier nichts angegeben, wird die sprachabhängige allgemeine Warnung "Dieser Zielsatz ist gesperrt." ausgegeben. Im Text kann die Kennung "%s" verwendet werden. Sie wird durch die allgemeine Warnung ersetzt.

Pfad /SysSRun[<Channel>]/LockWarning

Bedeutung der Variablenwerte

<Leerstring>	Bei gesperrtem Zielsatz wird eine allgemeine Warnung ausgegeben.
<Hinweistext>	Bei gesperrtem Zielsatz wird dieser Hinweistext ausgegeben.

Name des Zielsatz-Abgleichprogramms	NameOfAsup

Definiert kanalspezifisch den Namen inkl. Pfadangabe des Programms, das als Zielsatz-Abgleichprogramm aufgerufen werden soll.

Wird kein Programm angegeben, wird vor der Anfahrbewegung kein Zielsatz-Abgleichprogramm aufgerufen.

Pfad /SysSRun[<Channel>]/NameOfAsup

Bedeutung der Variablenwerte

<Leerstring>	Vor der Anfahrbewegung wird kein Zielsatz-Abgleichprogramm aufgerufen.
<gültiger Programmname>	Vor der Anfahrbewegung wird das angegebene Zielsatz-Abgleichprogramm aufgerufen.

Name des Init-Abgleichprogramms	NameOfInitAdjPrg

Definiert kanalspezifisch den Pfad und den Namen des Unterprogramms, das als Initialisierungs-Abgleichprogramm aufgerufen werden soll.

Wird kein Unterprogramm angegeben, wird vor der Anfahrbewegung kein Init-Abgleichprogramm aufgerufen.

Pfad /SysSRun[<Channel>]/NameOfInitAdjPrg

Bedeutung der Variablenwerte

<Leerstring>	Vor der Anfahrbewegung wird kein Init-Abgleichprogramm aufgerufen.
<gültiger Programmname>	Vor der Anfahrbewegung wird das angegebene Init-Abgleichprogramm aufgerufen.

Aktionssatz	ActionBlock

Definiert kanalspezifisch einen Aktionssatz.

Wird hier ein NC-Satz angegeben, wird dieser mit Beginn der Bearbeitung nach Satzvorlauf noch vor einem gegebenenfalls definierten Abgleichprogramm abgearbeitet.

Pfad /SysSRun[<Channel>]/ActionBlock

Bedeutung der Variablenwerte

<Leerstring>	Es liegt kein Aktionssatz vor.
<Aktionssatz>	Mit Beginn der Bearbeitung nach Satzvorlauf wird zunächst der angegebene Aktionssatz abgearbeitet.

Laufzeitüberwachung für Satzvorlauf	Timeout

Ermöglicht es im jeweiligen Kanal, den Satzvorlauf nach Ablauf der Überwachungszeit abubrechen.

Die Laufzeitüberwachung kann eingesetzt werden, wenn ein Zielsatz aufgrund einer nicht erfüllten Bedingung in einem Programm übersprungen und der Satzvorlauf wegen einer im Programm enthaltenen Endlosschleife nicht automatisch beendet wird. Programme, welche potentiell Endlos-Schleifen enthalten, sollten dieses Systemdatum zu Beginn auf einen ausreichend großen Wert setzen und anschließend wieder restaurieren.



Ist die Steuerung stark belastet, kann es zu einem unberechtigten Abbruch des Satzvorlaufes aufgrund einer Überschreitung der Laufzeit kommen, wenn der Timeout-Wert nur geringfügig über der Laufzeit des Satzvorlaufes der unbelasteten Steuerung eingestellt ist.

Hier ist ein ausreichend großer Wert einzustellen oder die Überwachung komplett zu deaktivieren.

Pfad	/SysSRun[<Channel>]/Timeout	
Bedeutung der Variablenwerte	0	Die Laufzeitüberwachung ist deaktiviert.
	> 0	Die Laufzeitüberwachung ist aktiv (z. B. 10000 ms).

ActTool.K1	ActTool.K1

Werkzeugdatenelement K1 des aktiven Werkzeugs

Pfad /SysTool[<Channel>]/ActTool/K1



Das Datum ist frei belegbar, wird aber in der Regel für die Sektorauswahl verwendet und dient dann z. B. zur Unterscheidung von Spindel, Greifer, Werkzeugliste.

ActTool.K2	ActTool.K2

Werkzeugdatenelement K2 des aktiven Werkzeugs

Pfad /SysTool[<Channel>]/ActTool/K2



Das Datum ist frei belegbar, wird aber in der Regel für die Platzauswahl verwendet und dient dann z. B. zur Unterteilung der Sektoren in Plätze.

Nummer aktive Schneide Zielsatz	TargEd

Nummer der aktiven Schneide im Zielsatz.

Pfad	/SysToolSRun[<Channel>]/TargEd	
Bedeutung der Variablenwerte	0	Keine Schneide aktiv
	1..16	Nummer der aktiven Schneide im Zielsatz

Nummer aktive D-Korrektur Zielsatz	TargDNr

Nummer der aktiven D-Korrektur im Zielsatz.

Pfad	/SysToolSRun[<Channel>]/TargDNr	
Bedeutung der Variablenwerte	0	Keine Korrektur aktiv
	1..99	Nummer der aktiven D-Korrektur im Zielsatz

Abgleich modaler NC-Funktionen "Adjust, ADJ"

Beschreibung: Mit Hilfe der Abgleichfunktion "Adjust" lassen sich modale NC-Funktionen innerhalb der Abgleichprogramme nach Satzvorlauf komfortabel abgleichen. Der Abgleichfunktion wird als Parameter die Syntax einer modalen NC-Funktion mitgegeben. Die Steuerung aktiviert daraufhin die in der entsprechenden Modalgruppe vor dem Zielsatz des Satzvorlaufs wirksame NC-Funktion.

Syntax:

ADJ (<NCF-Name>)	
<NCF-Name>	Syntax einer modalen NC-Funktion aus der Gruppe, die abgeglichen werden soll.

Tab. 5-12: Syntax ADJ

Beispiel: ADJ(G0)

Es wird die unmittelbar vor dem Zielsatz modal wirksame Geometriefunktion (G0, G1, G2, G3, ...) aktiviert und die NC-Funktionsgruppe somit abgeglichen.

Bei nachfolgend aufgeführten NC-Funktionen werden dabei gleichzeitig die funktionsspezifischen Parameter mit abgeglichen:

Modalgruppe	NC-Funktionen	Unterstützte Parameter
Interpolationsfunktionen	G0	Polarprogrammierung und Genauhaltfenster (NIPS, IPS<n>)
	G1	Polarprogrammierung und Genauhaltfenster (IPS<n>)
	G2, G3	Polarprogrammierung
Bahnslope	G8	Bahnshape
	G9	Bahn-,Achs-Shape
Ebene	G17, G18, G19, G20	Achszuordnung
Werkzeuglängenkorrektur	G47	Längenzuordnung
Nullpunktverschiebung 1	G52.1	programmierbare NPV 1 ^{*)}
	G53.1	NPV 1 Aus ^{*)}
	G54.1,...,G59.1	aktive NPV Tabellenbank 1 ^{*)}
	...	
Nullpunktverschiebung 5	G52.5	programmierbare NPV 5 ^{*)}
	G53.5	NPV 5 Aus ^{*)}
	G54.5,...,G59.5	aktive NPV Tabellenbank 5 ^{*)}
Placements: "Schiefe Ebene" 1	G151.1	Werkzeugorientierung ^{**)}
	G152.1	programmierbare Schiefe Ebene 1 ^{**)}
	G153.1	Schiefe Ebene 1 Aus ^{**)}

Modalgruppe	NC-Funktionen	Unterstützte Parameter
	G154.1,...,G159.1	aktive Schiefe Ebene Tabellenbank 1 ^{**})
...		
Placements: "Schiefe Ebene" 5	G151.5	Werkzeugorientierung 5 ^{**})
	G152.5	programmierbare Schiefe Ebene 5 ^{**})
	G153.5	Schiefe Ebene 5 Aus ^{**})
	G154.5,...,G159.5	aktive Schiefe Ebene Tabellenbank 5 ^{**})
Genauhalt	G61	Genauhaltfenster (IPS<n>)
Werkzeuglängenzuordnung	G78	ActPlane bzw. Längenzuordnung
Bahnvorschub	G93	F-Wert
	G94	F-Wert, F2-Wert
	G95	F-Wert, Referenzachse (FRA)
Spindelprogrammierung	G96	Bezugsachse, Spindelzuordnung
	G97	Spindelzuordnung
D-Korrektur	D0	D-Korrektur Aus ^{***})
	D1,...,D99	aktive D-Korrektur ^{***})
Achsbeschleunigung	AAC	programmierte Achsbeschleunigungen
Achsruck	AJK	programmierte Achsrucke
additive Programmkordinaten Verschiebung	ATR	Programmnullpunktkoordinaten
maximale Achsgeschwindigkeit	AVE	Liste der Achsgeschwindigkeiten
Werkstücklage	BCR	Werkstücknullpunktkoordinaten, Winkel
Fasen-/Rundungen	CHL CHS RND RNE	Fasenlänge, Fasenabschnitt Radius, Epsilon
automatische Eckenerkennung	CLD	Knickwinkel, Winkelfaktor, max. Punkt- abstand, Abstandsfaktor, Genauigkeits- schranke
Fräsebahn -Kollisions- erkennung	CLN	Überwachungsart, Vorschaubereich
Achstransformation	COORD	Achstransformation 1 und 2 mit Trafo- Optionen
Radius-/Durchmesser- programmierung	DIA RAD	Durchmesserkoordinaten
Kanalspindeln und Spindelgruppen	DSP, GSP, RSP SPGn, SPGALL	Kanalspindeln und Kanalgruppenzuord- nung (entspricht SPADJSP)

Modalgruppe	NC-Funktionen	Unterstützte Parameter
Satzübergang ohne Geschwindigkeitsabsenkung	DTT	Übergangswinkel
Vorschub-Ausblendachsen	FAD	Liste der Ausblendachsen
Vorschubkoordinaten Gruppe	FDG FDGT	Liste der Bahnkoordinaten, Vorschubgruppentyp
Geradheit-/Winkelfehler-Kompensation	GCT	-
aktive Achsrucke	JAL	Liste der bahnrelevanten Achsruckachsen
Ruckbegrenzung	JKC	-
Eingabehilfe Spiegeln	MIR	Spiegelachsen
Hauptspindel	MSP	Spindel
Bahnbeschleunigung	PAC	Bahnbeschleunigung, Bahnverzögerung
Pol-Definition	PLS	Spiegel-/Drehpunkt
Rundachspositioniermode	PMD	Positioniermode
Pol-Definition	POP	Pol-Koordinaten
Genauigkeitsprogrammierung	PRP	Radius-Konturfehler (EPS), Nachlauf (DIST)
maximale Radialbeschleunigung	RAC	Beschleunigung
Eingabehilfe Drehen	ROT	Winkel
Eingabehilfe Skalieren	SCL	Achsfaktoren
Eingabehilfe Verschieben	SHT	Konturnullpunktkoordinaten
Spline-Definition	SPD	Spline-Typ, Spline-Koordinaten, Startbedingung (SBC), Endbedingung (EBC), Spline-Parameterlänge (PL)
Geometrisches Splitten	SPT	Mode, Länge
Programmkoordinaten Verschiebung	TRS	Programmnullpunktkoordinaten
Gewindebohrspindel	TSP	Spindel

Tab. 5-13: Abgleichbare modale parametrierbare NC-Funktionen

- *) implizite NPV Tabellenvorwahl (ZOS)
 **) implizite Placement Tabellenvorwahl (PMS)
 ***) implizite D-Korrektur Tabellenvorwahl (DCS)

Besonderheiten und Restriktionen:

- Für die oben nicht namentlich erwähnten modalen parametrierbaren NC-Funktionen muss der Abgleich ggf. unter Verwendung von Satzvorlauf-Systemdaten im Abgleichprogramm (InitAdjPrg) oder mit Hilfe des Aktionssatzes erfolgen.

- Für jede Bank einer Nullpunktverschiebung bzw. Schiefen Ebene erfolgt der Abgleich separat: ADJ(G54.1), ADJ(G54.2) usw.
- Die Werkzeugradiuskorrektur G41/G42 wird im abgeglichenen Zustand erst mit dem Anfahrssatz nach dem Abgleich wirksam. Während des Abgleichs wird die Wirkung intern unterdrückt; es erfolgt keine Korrekturberechnung.

Modale NC-Funktionsgruppen die während der Anfahrbewegung wieder hergestellt werden

Folgende modalen Funktionen und Zustände werden im Anfahrssatz automatisch restauriert, für den Fall, dass diese im Aktionssatz oder im TargetAdjustPrg umgeschaltet bzw. verändert wurden.

Modalgruppe	NC-Funktionen	Unterstützte Parameter
Fräserbahnkorrektur	G40	Fräserbahnkorrektur AUS
	G41	Fräserbahnkorrektur links vom Werkstück
	G42	Fräserbahnkorrektur rechts vom Werkstück
Werkzeuglängenkorrektur	G47	Werkzeuglängenkorrektur EIN
	G48	Werkzeuglängenkorrektur AUS
Genauhalt	G61	Genauhalt EIN
	G62	Genauhalt AUS
Abs/Rel	G90	Absolutmaßprogrammierung
	G91	Relativmaßprogrammierung
Vorschubprogrammierung	G93	Zeitprogrammierung
	G94	Vorschubprogrammierung in mm
	G95	Vorschubprogrammierung in Umdrehung

Tab. 5-14: Modale NC-Funktionen während der Anfahrbewegung

5.2.5 Umschaltung von modalen Zuständen im Abgleichprogramm oder Aktionssatz

Im Aktionssatz oder im ersten Satz des Abgleichprogramms werden einige NC-Funktionen implizit zurückgesetzt (ausgeschaltet), falls sie im Rahmen des Satzvorlaufs aktiviert wurden:

- G41/G42 → G40
- G47 → G48

Im Gegenschritt dazu werden diese Funktionen im Anfahrssatz automatisch reaktiviert. Außerdem werden folgende modalen Funktionen und Zustände restauriert (für den Fall, dass diese im Aktionssatz oder Abgleichprogramm umgeschaltet bzw. verändert wurden:

- G90/G91
- G93/G94/G95
- F-Wert

Die Zustände aller weiteren modalen Funktionen müssen von Hand im Abgleichprogramm verwaltet werden, sofern sie hier verändert werden. Hierzu bietet sich u. a. der CPL-Befehl "NCF" an, mit dessen Hilfe die aktiven Funk-

tionen der einzelnen Modalgruppen sowie der konfigurierten Hilfsfunktionsgruppen gelesen und in Variablen gespeichert werden können.

Alle Hilfsfunktionen, die nach Satzvorlauf aktiviert werden sollen, müssen daher Hilfsfunktionsgruppen zugewiesen sein.

Die veränderten Gruppen müssen dann spätestens am Ende des Abgleichprogramms restauriert werden. Geschieht dies nicht, wird der veränderte Modalzustand mit in das angewählte Teileprogramm übernommen.

In diesem Zusammenhang sei auch auf die NC-Funktion "AUXFUNC" hingewiesen, die die aktuellen Hilfsfunktionen aller Hilfsfunktionsgruppen ausgibt.

Zur Kommunikation nach außen stehen spezielle Systemdaten, sowie Signale am Bitinterface zur Verfügung. Während die Systemdaten zur Steuerung des Satzvorlaufs sowie des Abgleichs dienen und zum Zeitpunkt der Satzvorbereitung (prep) gesetzt werden, erfolgt die Ausgabe der Interface-Signale jeweils zum aktiven Zeitpunkt (act).

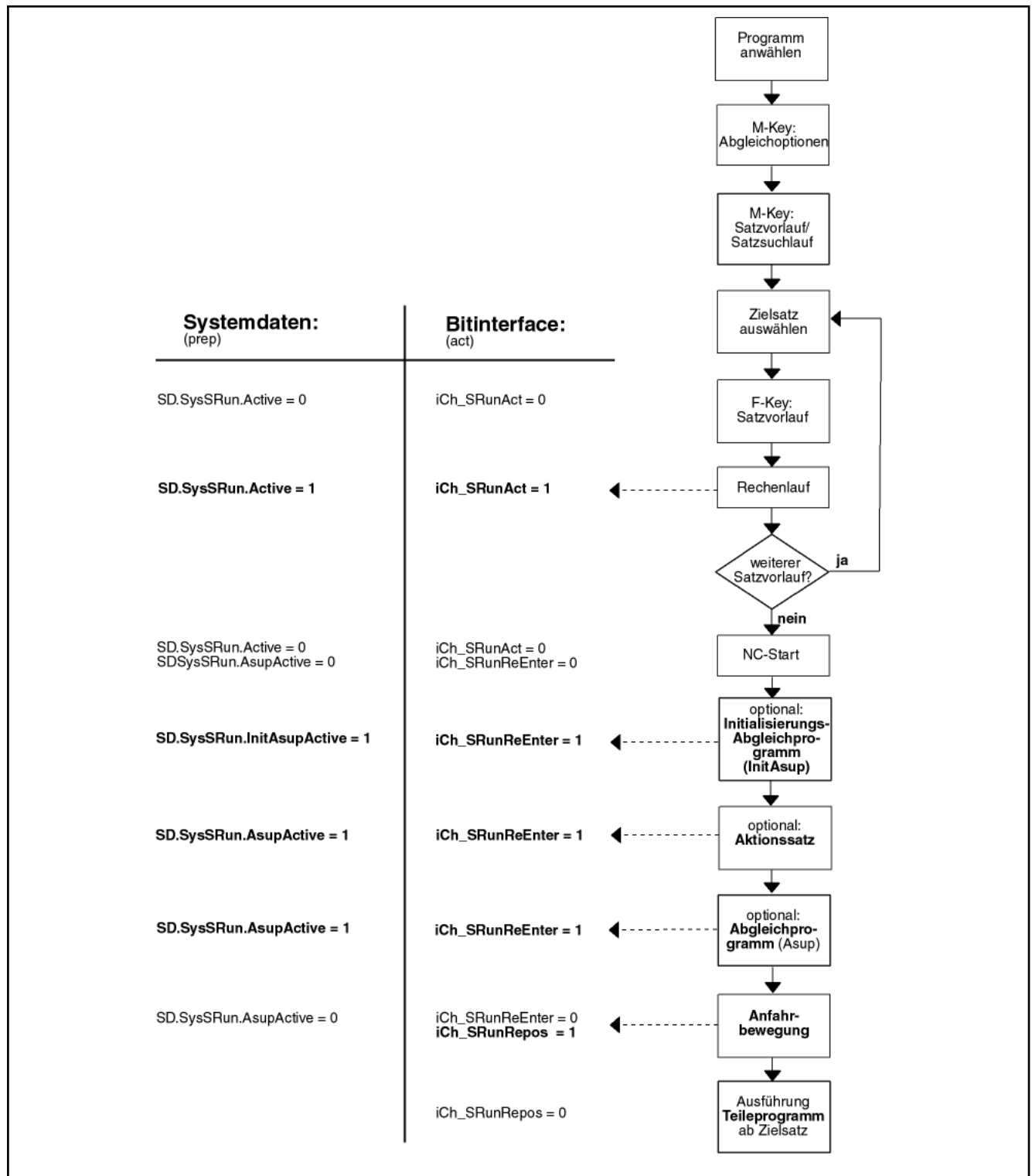


Abb. 5-10: Umschaltung von modalen Zuständen im Abgleichprogramm oder Aktionssatz

5.2.6 Anwendungen zum Satzvorlauf

Handeingabe

Nach dem Satzvorlauf können über Handeingabe ein oder mehrere NC-Sätze vorgegeben und abgearbeitet werden. Hierdurch kann ein manueller Abgleich der NC-Zustände durchgeführt werden.



Die Handeingabe ist optional und kann nach dem InitAdjustPrg aufgerufen werden.

Ein vorangehendes InitAdjustPrg ist zwingend notwendig.

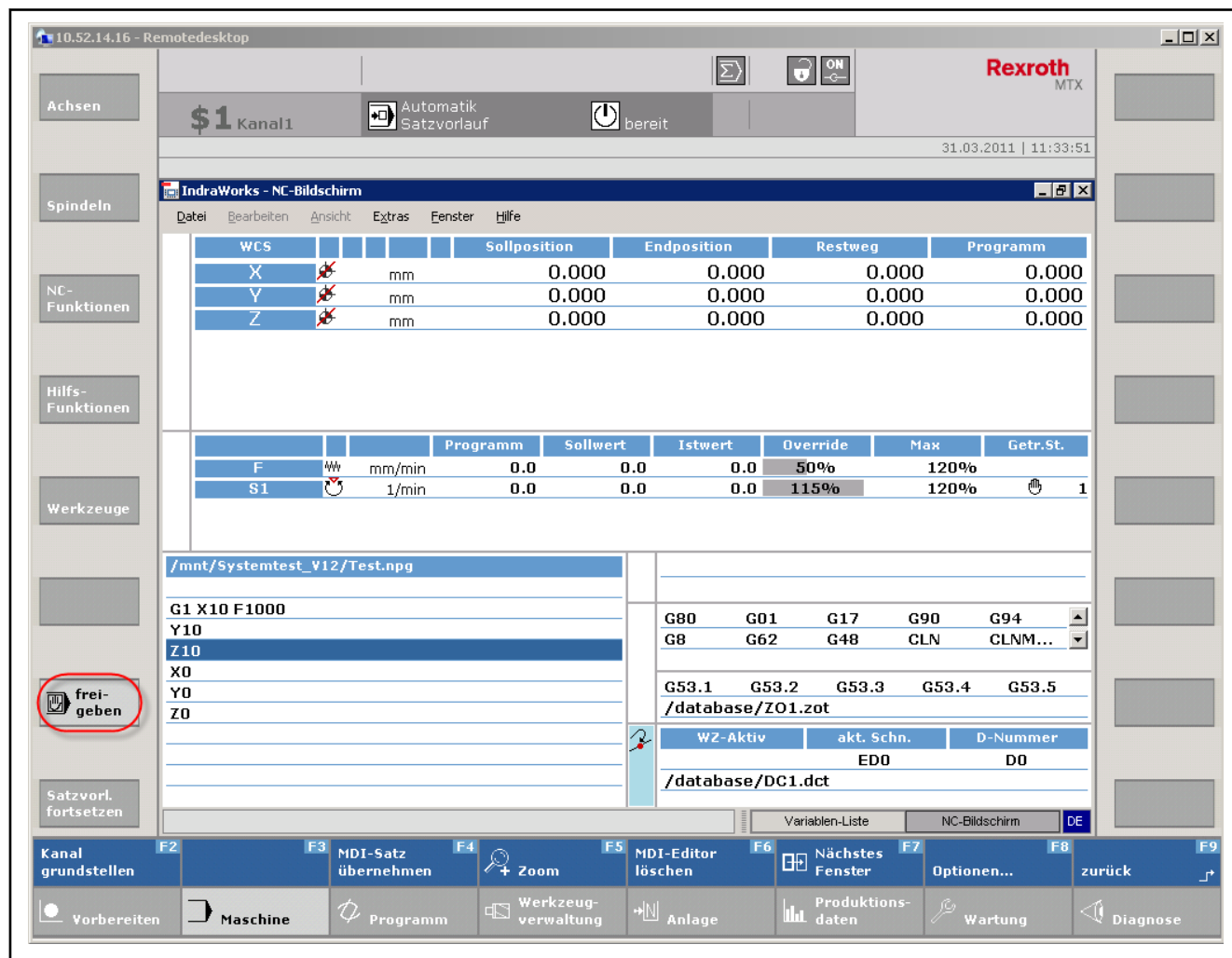


Abb. 5-11: Freigabe der Handeingabe

Ist die Handeingabe freigegeben, wird nach Beenden des Abgleichprogramms (InitAdjustPrg) im Übersichts-Abgleichbildschirm die Satzfortschrittsanzeige durch den Handeingabe-Editor automatisch ersetzt. Analog zur Betriebsart MDI können NC-Sätze eingeben, an die NC übertragen und mit der Taste NC-Start (Maschinentafel) abgearbeitet werden.

Ohne vorherige Eingabe eines NC-Satzes führt der nächste NC-Start zum Aufruf des Aktionssatzes und des Abgleichprogramms (TargetAdjustPrg), soweit diese parametrisiert sind. Die Handeingabe ist abgeschlossen, der Handeingabe-Editor im Abgleichbildschirm wird durch die Satzfortschrittsanzeige ersetzt.

Einschränkungen	Die Handeingabe steht nur zur Verfügung wenn ein InitAdjustPrg verwendet wird. Es sind keine CPL Anweisungen und keine Unterprogrammaufrufe möglich. Es sind nur einzelne Sätze möglich, keine gepufferte NC-Satzvorgabe. Es sind nur einfache NC-Funktionen programmierbar. Weitere Informationen zu diesem Thema finden Sie in der Dokumentation "MTX xxVRS Standard-NC-Bedienung" (DOK-MTX***-NC*OP***Vxx-APRS-DE-P), Kapitel "Satzsuchlauf/Satzvorlauf".

Wiedereinstiegsmarken RCB-Marken

Beschreibung	RCB markiert im Programm eine Stelle, an der das Programm nach einem Abbruch gezielt fortgeführt werden kann. Im Unterschied zum normalen Satzvorlauf mit manueller Angabe des Zielsatzes ist es damit möglich, gezielt erfolgreich abgearbeitete Schleifendurchläufe zu überspringen oder das Finden des Zielsatzes nur unter gewissen Voraussetzungen zuzulassen.
Anwendung	Wird ein NC-Programm während der Abarbeitung aus verschiedenen Gründen abgebrochen, so soll an einer vordefinierten Stelle mit der Abarbeitung sofort wieder begonnen werden können. Gründe für einen Abbruch können Werkzeugbruch, Stromausfall, NC-Steuerungsfehler oder anderes sein. Der NC-Programmierer kann für solche Fälle bestimmen, an welcher Stelle (NC-Satz) ein Wiederaufstart im NC-Programm sinnvoll ist. Zu diesem Thema siehe auch Dokumentation "MTX Programmierhandbuch", Kapitel "Programmfortführung mit Satzvorlauf nach Programmabbruch RCB".

Init-Abgleichprogramm "InitAdjPrg"

Im Abgleichprogramm "InitAdjPrg" muss der Zustand des Zielsatzes (-1) hergestellt werden.

Abzugleichen sind:

- Achskonfiguration
- Koordinatenkonfiguration
- Spindeln
- Kopplungen
- Werkzeuge
- NC-Funktionen
- Hilfsfunktionen
- Achs- und Achsnamenskonfiguration (sofern der NC-Befehl "TAX" nicht verwendet wird!)
- Sonstiges



Der NC-Befehl "TAX" stellt die Achs- und Achsnamenskonfiguration automatisch her!

Die Steuerung aktiviert mit dem TAX Befehl die Achskonfiguration vor dem Zielsatz des Satzvorlaufs. Zuviel vorhandene Achsen werden abgegeben. Benötigte Achsen werden in den Kanal herein geholt. Auch die Achsnamen werden mit übernommen.

Ist eine Achse anderweitig belegt, dann wird sie nicht berücksichtigt:

- Eine Achse kann nicht aufgenommen werden, wenn sie von einem anderen Kanal belegt ist.

- Eine C-Achse kann nicht aufgenommen werden, wenn sie sich im Spindelbetrieb befindet.
- Eine Achse kann nicht abgegeben werden, wenn sie Bestandteil einer Transformation ist.

Zu diesem Thema siehe auch Dokumentation "MTX Programmieranleitung", Kapitel "Abgleich Achskonfiguration, TAX".

Reihenfolge im Adjustprogramm

Mögliche funktionale Reihenfolge für den Abgleich im InitAdjPrg.

- Kanalspindelkonfiguration herstellen
- Aktive Transformationen abschalten
- C-Achsen in Spindel wandeln
- Kanalspindel abgeben
- Überflüssige Achsen abgeben (RAX)
- Achsen aufnehmen (WAX)
- Spindelkonfiguration herstellen
- C-Achsen aufnehmen
- Transformation aktivieren
- Werkzeugwechsel
- Spindel abgleichen
- ggf. weitere NC-Funktionen abgleichen
- Hilfsfunktionen abgleichen

VORSICHT

Die Reihenfolge des Abgleichs ist abhängig von der Applikation

Applikation genau betrachten und den Abgleich anpassen. Genaues Prüfen des Zustandes der Maschine nach dem Abgleich, bevor eine Verfahrbewegung ausgelöst wird.

5.2.7 Spindeln

Beschreibung

Funktion

Spindeln sind asynchrone Antriebe. Während eines Satzvorlaufs werden die Spindeln nicht angesteuert. Die programmierten Daten werden NC-intern protokolliert und am Ende des Rechenlaufs in dem Systemdatum

SD.SysChSpSRun[<Kanalnr.>]

abgelegt.

Ebenfalls in diesem Systemdatum legt die NC für den Spindelabgleich notwendige Informationen ab. Diese Abgleichinformationen werden beim Abarbeiten der unter SysSRun[<Kanalnr.>].NameOfInitAdjPrg und SysSRun[<Kanalnr.>].NameOfAsup hinterlegten Abgleichprogramme bei jeder Spindelprogrammierung aktualisiert.

Restriktionen

Die folgenden Spindelfunktionen werden nicht unterstützt:

- SpAdmin / SPA
- SSpAdm
- SSpAdmOff

Relevante NC-Funktionen

Adjust(..) / ADJ(..)

ADJ(G96) gleicht die NC-Funktionsgruppe für die konstante Schnittgeschwindigkeit (G96) und die direkte Spindeldrehzahlprogrammierung (G97) ab.

SpAdjSp

Die Funktion SpAdjSp führt den Abgleich der logischen Spindeln des Kanals durch (Automatische Abfolge von RemSpindle und GetSpindle).

SpAdjStr <SpindelNr.>

Die Funktion SpAdjStr <SpindelNr.> füllt beim Aufruf im Abgleichprogramm folgende Komponenten des Systemdatums SD.SysChSpSRun:

- SpeedStr
- MoveStr
- GearStr

Beispiel

Siehe [Kap. "Spindelabgleich" auf Seite 70.](#)

Dies ist ein Beispielprogramm für den Abgleich einfacher Spindelfunktionen.

Dieses Programm kann als Unterprogramm eines Abgleichprogramms verwendet werden.

5.2.8 Systemdaten

Die Satzvorlaufdaten werden im Systemdatum "SysSRun" zusammengefasst. Die Satzvorlaufdaten können von der Bedienoberfläche, der SPS oder vom Teileprogramm gelesen oder geändert werden. Damit kann während Satzvorlauf und im Abgleichprogramm flexibel reagiert werden.

Elementname	Beschreibung	Typ	Größe in Byte	Schreibrecht
Active	Hat während des Satzvorlauf den Wert 1, ansonsten 0. Dient zum Abfragen im Teileprogramm. Damit können Programmpassagen während des Satzvorlaufs übersprungen werden, z. B. ein Werkzeugwechsel oder Änderungen in Tabellen.	Byte_t	1	NC-Kern
AsupActive	Hat bei Abarbeitung des Abgleichprogramms (Asup) nach Satzvorlauf den Wert 1, ansonsten 0. Dient zum Abfragen im Abgleichprogramm. Damit können z. B. allgemeine Unterprogramme als Abgleichprogramm verwendet werden, die sich im Asup des Satzvorlaufs anders verhalten.	Byte_t	1	NC-Kern
InitAdjPrgActive	Hat den Wert 1, wenn InitAsup nach SV aktiv ist, sonst 0. Zum Abfragen im Programm. Damit können z. B. allgemeine Unterprogramme verwendet werden, die sich im InitAsup des SVs anders verhalten können.	Byte_t	1	NC-Kern

Kanalsteuerung

Elementname	Beschreibung	Typ	Größe in Byte	Schreibrecht
LookAheadPreRun	Hat den Wert 1, wenn der Satzvorlauf im Rahmen des eingestellten LookAheads über den Zielsatz hinaus Sätze vorbereitet. Dient zum Abfragen im Teileprogramm. Mit Hilfe dieser Information kann vermieden werden, dass CPL-Teile die hinter dem Zielsatz folgen zweimal durchlaufen werden (während Satzvorlauf und während der eigentlichen Vorbereitung nach Satzvorlauf).	Byte_t	1	NC-Kern
TargPos	In dieses Datenfeld schreibt die Steuerung nach Satzvorlauf die Zielposition des Anfahrsatzes, bezogen auf das aktuelle Werkstückkoordinatensystem vom (Zielsatz-1).	Double_t	8x8	NC-Kern
TargBlockSrc	Zielsatz Nach Erreichen des Zielsatzes wird hier der Zielsatzstring eingetragen. CPL-Teile sind darin bereits aufgelöst. Ist der Zielsatz ein CPL-Satz, dann wird der nachfolgende DIN-Satz eingetragen.	IsoLatin1String	512	NC-Kern
LockTargBlock	Mit Hilfe dieses Datums lassen sich Bereiche im Teileprogramm für die Auswahl als Zielsatz sperren. Bei gesetzter Sperre (Wert = 1) wird der Zielsatz zwar akzeptiert, aber es wird kein NC-Start angenommen. Es wird eine allgemeine Warnung ausgegeben. Ist ein Hinweistext im Datum "LockWarning" vorhanden, wird dieser ausgegeben. Das Fortsetzen des Satzvorlaufs auf einen anderen Zielsatz ist möglich.	Int_t	4	jeder
LockWarning	Hinweistext bei gesperrtem Zielsatz Wird hier nichts angegeben, wird dies sprachabhängige allgemeine Warnung "Dieser Zielsatz ist gesperrt." ausgegeben. Im Text kann die Kennung "%s" verwendet werden. Sie wird durch die allgemeine Warnung ersetzt.	IsoLatin1String	80	jeder
NameOfAsup	Name des Unterprogramms (inkl. Pfad), das als Abgleichprogramm aufgerufen werden soll. Ist kein Name angegeben, wird vor der Anfahrbewegung kein Abgleichprogramm aufgerufen.	IsoLatin1String	100	jeder
NameOfInitAdjPrg	Namen des Init-Abgleichprogramms. Wird hier der Pfadnamen eines Init-Abgleichprogramms angegeben, wird es automatisch aktiviert.	isoLatin1String	100	jeder

Elementname	Beschreibung	Typ	Größe in Byte	Schreibrecht
ActionBlock	Aktionssatz Wird hier ein Aktionssatz angegeben, wird dieser mit Beginn der Bearbeitung nach Satzvorlauf noch vor einem gegebenenfalls definierten Abgleichprogramm ausgeführt.	IsoLatin1String	512	jeder
Timeout	Dauer, nachdem der Satzvorlauf abgebrochen wird. Der Wert wird in Millisekunden (ms) angegeben. Der Wert "0" bedeutet, dass die Laufzeit-Überwachung deaktiviert wird. Die Laufzeitüberwachung kann eingesetzt werden, wenn ein Zielsatz aufgrund einer nicht erfüllten Bedingung in einem Programm übersprungen und der Satzvorlauf wegen einer im Programm enthaltenen Endlosschleife nicht automatisch beendet wird. Programme, die potentiell Endlos-Schleifen enthalten, sollten dieses Systemdatum zu Beginn auf einen ausreichend großen Wert setzen und anschließend wieder restaurieren. Achtung: Ist die Steuerung stark belastet, kann es zu einem unberechtigten Abbruch des Satzvorlaufes aufgrund einer Überschreitung der Laufzeit kommen, wenn der Timeoutwert nur geringfügig über der Laufzeit des Satzvorlaufes der unbelasteten Steuerung eingestellt ist. Hier ist ein ausreichend großer Wert einzustellen oder die Überwachung komplett zu deaktivieren.	Int_t	4	jeder
RetPaStrategy	0 - Anfahren auf Startpunkt (Standard) 1 - Anfahren auf Endpunkt Die Anfahrbewegung wird in Abhängigkeit des Systemdatums auf den Start- oder Endpunkt des Zielsatzes durchgeführt. Das Systemdatum muss zwingend vor dem Beginn des Abgleich-Zyklus festgelegt worden sein. Es kann im Satzvorlauf gesetzt werden, jedoch nicht mehr im Init-Abgleichprogramm oder im Abgleichprogramm.	Byte_t	1	jeder
ManWaitForStart	Handeingabe nach Satzvorlauf 0 - Handeingabe deaktivieren 1 - Handeingabe aktivieren Nach Abarbeiten des Init-Abgleichprogramms können nach Aktivieren der Handeingabe NC-Sätze zum manuellen Abgleich vorgegeben werden.	Byte_t	1	jeder

Tab. 5-15: Systemdaten

5.2.9 Interface-Signale

Folgende Interface-Signale signalisieren der SPS den Zustand von Satzvorlauf am Kanalinterface:

SrunAct	Satzvorlauf aktiv Das Signal steht an, solange das Programm im "Rechenlauf" bis zum Abbruchsatz (Zielsatz-1) abgearbeitet wird.
SrunReEnter	Wiedereintritt aktiv Das Signal wird ausgegeben, solange der Aktionssatz oder das Abgleichprogramm abgearbeitet werden.
SrunRepos	Wiederauffahren aktiv Das Signal steht an, solange der Anfahrssatz an den Zielsatz aktiv ist.

Tab. 5-16: Interface-Signale

5.2.10 Look Ahead

Während des Satzvorlaufs können vor dem Zielsatz NC-Funktionen aktiv sein, die eine Satz vorausschau benötigen (Look Ahead), z. B. die Fräsbahnkorrektur G41/G42. Die NC-Sätze hinter dem Zielsatz werden daher soweit vorbereitet, bis der letzte Satz vor dem Zielsatz (der Abbruchsatz) aktiv geworden ist.

Innerhalb des Look Aheads werden auch CPL-Ausdrücke ausgeführt, wobei z. B. Variablen ihre Werte ändern können.

Mit Beginn der Bearbeitung werden diese Programmpassagen erneut durchlaufen. Zuvor werden die Inhalte der **CPL-Variablen** auf den ursprünglichen Zustand zurückgesetzt.

Die Inhalte von **permanenten Variablen** und **Systemdaten** bleiben hingegen erhalten!

Sollen Passagen im Rahmen der Satz vorausschau hinter einem Zielsatz nicht durchlaufen werden, muss das Teileprogramm modifiziert und diese Passagen mittels Abfrage des Systemdatums "SD.SysS-Run.LookAheadPreRun" programmtechnisch verriegelt werden.

5.2.11 Allgemein

Funktionale Einschränkungen

Zur Zeit gelten nachfolgende funktionale Einschränkungen:

- Nur das Init-Abgleichprogramm beginnt mit dem Zustand vor der Programmanwahl. Aktionssatz bzw. das Abgleichprogramm springen auf den Zustand, der sich mit dem Zielsatz ergibt. Im Init-Abgleichprogramm ist dafür zu sorgen, dass die beiden Zustände abgeglichen (so ähnlich wie möglich) sind.
- Keine C-Achsumschaltung
- Kein Satzvorlauf in Gewindebohren (G63): Im TP verriegeln
- Kein Satzvorlauf in eine Spline-Sequenz hinein, jedoch darf der Zielsatz ein Umschaltsatz auf G6 sein
- Kein Satzvorlauf in die Mitte einer Rampeninterpolatorsequenz hinein, wohl aber auf den Beginn einer solchen Sequenz
- Kein Satzvorlauf auf Gewindeschneiden G33 (siehe auch G63).
- Keine 3D-Werkzeugkorrektur (G140 - G142)

- Kein Festanschlag
- Kein Achsentausch
- Achskopplung AXC

Einstieg in Unterprogramme

Im MTX-Satzvorlaufdialog ist ein Abtauchen in Unterprogramme, die per CPL-Variable aufgerufen werden, nicht möglich.

Vorschubprogrammierung (pro Umdr.) "G95"

G95 bedingt eine sich drehende Hauptspindel.

Wenn im Rechenlauf als letztes ein G95 aktiviert wurde, aber noch keine Spindeldrehzahl programmiert ist, so ist der Vorschub der Kanalachsen = 0. Bei der Anfahrbewegung der Achsen am Ende des Satzvorlaufes wird ein Vorschub benötigt. Dieser kann in diesem speziellen Fall nicht über die Systemdaten "FeedVal" oder "OptFeedVal" gebildet werden.

Der Achsenabgleich wird nicht durchgeführt.



Den Zielsatz so wählen, dass eine Spindeldrehzahl im Rechenlauf aktiviert wird.

Den Achsenabgleich manuell in der Handeingabe durchführen.

Den Achsenabgleich mittels G77 im TargetAsupPrg ausführen.

Relevante Interface-Signale

iCh_SrunAct	Satzvorlauf aktiv Das Signal steht an, solange das Programm im "Rechenlauf" bis zum Abbruchsatz (Zielsatz-1) abgearbeitet wird.
iCh_SrunReEnter	Wiedereintritt aktiv Das Signal wird ausgegeben, solange der Aktionssatz oder das Abgleichprogramm abgearbeitet werden.
iCh_SrunRepos	Wiederanfahren aktiv Das Signal steht an, solange der Anfahrsatz an den Zielsatz aktiv ist.

Tab. 5-17: Relevante Interface-Signale

Relevante Systemdaten

SysSRun.Active	Dient zum Abfragen im Teileprogramm. Damit können Programmpassagen während des Satzvorlaufs übersprungen werden, z. B. ein Werkzeugwechsel oder Änderungen in Tabellen.
SysSRun.AsupActive	Dient zum Abfragen im Abgleichprogramm um. Damit können z. B. allgemeine Unterprogramme als Abgleichprogramm verwendet werden, die sich im Asup des Satzvorlaufs anders verhalten.
SysSRun.InitAdjPrgActive	Dient zum Abfragen im Abgleichprogramm. Damit können z. B. allgemeine Unterprogramme als InitAsup verwendet werden, die sich im InitAsup des Satzvorlaufs anders verhalten.

SysS-Run.LookAheadPreRun	Dient zum Abfragen im Teileprogramm. Mit Hilfe dieser Information kann vermieden werden, dass CPL-Teile, die hinter dem Zielsatz folgen, zweimal durchlaufen werden (während Satzvorlauf und während der eigentlichen Vorbereitung nach Satzvorlauf).
SysSRun.TargPos	In dieses Datenfeld schreibt die Steuerung nach Satzvorlauf die Zielposition des Anfahrsatzes bezogen auf das aktuelle Werkstückkoordinatensystem vom (Zielsatz-1).
SysSRun.LockTargBlock	Mit Hilfe dieses Datums lassen sich Bereiche im Teileprogramm für die Auswahl als Zielsatz sperren.
SysSRun.LockWarning	Hinweistext bei gesperrtem Zielsatz
SysSRun.NameOfAsup	Name des Unterprogramms (inkl. Pfad), das als Abgleichprogramm aufgerufen werden soll.
SysSRun.NameOfInitAdjPrg	Name des Unterprogramms (inkl. Pfad), das als Init-Abgleichprogramm aufgerufen werden soll.
SysSRun.ActionBlock	Aktionssatz

Tab. 5-18: Relevante Systemdaten

5.2.12 Beispiele

Achskonfiguration und Koordinatenkonfiguration

Die Achskonfiguration kann im InitAdjPrg mit dem NC-Befehl TAX hergestellt werden. Eine Achstransformation wird vorher abgeschaltet, und nach dem Achsabgleich wieder eingeschaltet. Die Koordinatenkonfiguration wird mit dem Adjust-Befehl "COORD" hergestellt.

Achs- und Koordinatenabgleich

Programm:

```
; switch off transformers
G40 G48 COORD(0)
; generate axis configuration
TAX
; adjust transformer
ADJ(COORD)
ADJ(G47)
ADJ(G40)
```



Werkzeugwechsel und Spindelabgleich müssen applikationsabhängig berücksichtigt werden.

Spindelabgleich

Der Spindelabgleich muss applikationsabhängig in verschiedene Abschnitte unterteilt werden. In dem Beispiel soll ein grober Überblick und eine mögliche Reihenfolge des Abgleichs gegeben werden.

Kanalspindeln und Spindelgruppen herstellen

```
0000 DIM SADR$(8) :REM For spindle syntax      (e.g., S1=)
0001 DIM GADR$(8) :REM for gear                (e.g., M142)
;
N100 SPADJSP
```

Alle nicht abgeglichenen C-Achsen abschalten

```

0200 FOR SP% = 1 TO SD.SysChSpSRun.MaxSpNumber
0201     IF SD.SysChSpSRun.Sp[SP%].CAxisNotAdj <> 0 THEN
N202         RAX([SD.SysChSpSRun.Sp[SP%].DriveNo])
N203         ATSW([SD.SysChSpSRun.Sp[SP%].DriveNo])
0204     ENDIF
0290 NEXT

```

Spindeln abgleichen

```

0300 FOR SP% = 1 TO SD.SysChSpSRun.MaxSpNumber
0301     IF SD.SysChSpSRun.Sp[SP%].NotAdjusted <> 0 THEN
N302         SpAdjStr[SP%]
0303         SADR$ = SD.SysChSpSRun.SpeedStr
0305         GADR$ = SD.SysChSpSRun.GearStr
0306         MOVE% = SD.SysChSpSRun.Sp[SP%].Move
;
;         Gear
0310         IF SD.SysChSpSRun.Sp[SP%].GearNotAdj <> 0 THEN
0311             IF SD.SysChSpSRun.Sp[SP%].AutoGearShift THEN
0312                 SYSSPNO% = SD.SysChSpSRun.Sp[SP%].SysSpNo
N313                 SSPGEAR([SYSSPNO%],[SD.SysChSpSRun.Sp[SP%].Gear])
0314             ELSE
N315                 [GADR$]
0316             ENDIF
0317         ENDIF
;
;         Speed limit
0320         IF SD.SysChSpSRun.Sp[SP%].LimitNotAdj <> 0 THEN
N321             SMX([SADR$][SD.SysChSpSRun.Sp[SP%].ProgMaxSpeed])
N322             SMN([SADR$][SD.SysChSpSRun.Sp[SP%].ProgMinSpeed])
0323         ENDIF
;
;         SERCOS operation mode
0330         IF SD.SysChSpSRun.Sp[SP%].OpModeNotAdj <> 0 THEN
0331             IF SD.SysChSpSRun.Sp[SP%].PosCtrlOn THEN
N332                 SPM([SADR$]1)
0333             ELSE
N334                 SPM([SADR$]0)
0335             ENDIF
0336         ENDIF
0337     ENDIF
0390 NEXT SP%
N400 ADJ(G96)

```

S-Werte und, falls notwendig, AutoGear vorgeben

```

0500 FOR SP% = 1 TO SD.SysChSpSRun.MaxSpNumber
0501     IF SD.SysChSpSRun.Sp[SP%].NotAdjusted <> 0 THEN
N502         SpAdjStr[SP%]
;
;         Speed or cutting velocity
0510         IF SD.SysChSpSRun.Sp[SP%].SpeedNotAdj <> 0 THEN
0511             SADR$ = SD.SysChSpSRun.SpeedStr
N512             [SADR$][SD.SysChSpSRun.Sp[SP%].ProgSpeed]
0513         ENDIF
;
;         AutoGear
0520         IF SD.SysChSpSRun.Sp[SP%].AutoGearShift THEN
0521             GADR$ = SD.SysChSpSRun.GearStr
N522             [GADR$]
0523         ENDIF
0524     ENDIF
0590 NEXT SP%

```



Falls eine Spindelkopplung vorhanden ist, muss diese auch abgeglichen werden. Hierzu eine mögliche Abfolge im nächsten Beispiel.

Alle notwendigen C-Achsen asynchron einschalten

```

0700 FOR SP% = 1 TO SD.SysChSpSRun.MaxSpNumber
0701     IF SD.SysChSpSRun.Sp[SP%].CAxisNotAdj AND 2 = 2 THEN
0702         IF SD.SysChSpSRun.Sp[SP%].CAxisType AND 32 = 32 THEN
0703             SETERR("Block pre-run cannot be executed in a G63 sequence!")
0704         ENDIF

```

Kanalsteuerung

```

0705     IF SD.SysChSpSRun.Sp[SP%].CAxisType AND 16 = 16 THEN
0706         SETWARN("Block pre-run in turret operation! Intentionally?")
0707         M0 ; Turret operation
0708         STAW([SD.SysChSpSRun.Sp[SP%].DriveNo])
0709     ELSE
0710         STAW([SD.SysChSpSRun.Sp[SP%].DriveNo])
0711     ENDIF
0712 ENDIF
0760 NEXT SP%

```

Abgleich erfolgreich

```

0800 WAIT
0801 IF SD.SysChSpSRun.SpConfNotAdjusted <> 0 THEN
0802     SETERR("Incorrect channel spindles.")
0803 ENDIF
0811 FOR SP% = 1 TO SD.SysChSpSRun.MaxSpNumber
0812     IF SD.SysChSpSRun.Sp[SP%].NotAdjusted <> 0 THEN
0813         IF SD.SysChSpSRun.Sp[SP%].GearNotAdj <> 0 THEN
0814             SETERR("Incorrect gear.")
0815         ENDIF
0816         IF SD.SysChSpSRun.Sp[SP%].LimitNotAdj <> 0 THEN
0817             SETERR("Incorrect speed limit.")
0818         ENDIF
0819         IF SD.SysChSpSRun.Sp[SP%].CAxisNotAdj <> 0 THEN
0820             SETERR("Incorrect C-axes.")
0821         ENDIF
0822         IF SD.SysChSpSRun.Sp[SP%].OpModeNotAdj <> 0 THEN
0823             SETERR("Incorrect drive operating mode.")
0824         ENDIF
0825         IF SD.SysChSpSRun.Sp[SP%].CpDataNotAdj <> 0 THEN
0826             SETERR("Incorrect coupling data.")
0827         ENDIF
0828         IF SD.SysChSpSRun.Sp[SP%].CoupleNotAdj <> 0 THEN
0829             SETERR("Incorrect coupling state.")
0830         ENDIF
0831         IF SD.SysChSpSRun.Sp[SP%].SpeedNotAdj <> 0 THEN
0832             SETERR("Incorrect S-value.")
0833         ENDIF
0834     ENDIF
0835 NEXT SP%

```

Das Starten der Spindeln wird erst am Ende des Abgleichs durchgeführt

Drehzahl und Bewegungsfunktion abgleichen

```

0000 DIM SADR$(8) :REM for spindle syntax      (z.B. S1=)
0001 DIM MADR$(8) :REM for motion function (e.g. M103)
;
;
; Specify speed and direction of travel
0100 FOR SP% = 1 TO SD.SysChSpSRun.MaxSpNumber
0101     IF SD.SysChSpSRun.Sp[SP%].NotAdjusted <> 0 THEN
0102         SpAdjStr[SP%] ; load syntax elements of current spindle
0103         SADR$ = SD.SysChSpSRun.SpeedStr
0104         MADR$ = SD.SysChSpSRun.MoveStr
0105         MOVE% = SD.SysChSpSRun.Sp[SP%].Move
;
; Speed or cutting velocity
0110     IF SD.SysChSpSRun.Sp[SP%].SpeedNotAdj <> 0 THEN
0111         [SADR$][SD.SysChSpSRun.Sp[SP%].ProgSpeed]
0112     ENDIF
;
; Motion function
0120     IF SD.SysChSpSRun.Sp[SP%].MoveNotAdj <> 0 THEN
0121         IF MOVE% = 19 THEN
0122             [MADR$][SADR$][SD.SysChSpSRun.Sp[SP%].OriPos]
0123         ELSE
0124             [MADR$]
0125         ENDIF
0126     ENDIF
0180 ENDIF
0190 NEXT SP%
;
; Adjustment successful?
0200 WAIT
0201 FOR SP% = 1 TO SD.SysChSpSRun.MaxSpNumber
0202     IF SD.SysChSpSRun.Sp[SP%].NotAdjusted <> 0 THEN
0203         SETERR("Spindle adjustment failed.")
0204     ENDIF
0205 NEXT SP%

```



Wenn noch ein TargetAdjustPrg aktiv wird, dann sollte der Abgleich der Drehzahl und Bewegungsfunktion in diesem Abgleichprogramm erfolgen.

Spindelkopplung

Bei einer Spindelkopplung müssen beim Abgleich verschiedene Schritte durchgeführt werden. Eine grobe Einteilung mit Beispiel soll den Zusammenhang aufzeigen.

- Kopplungsdaten der Spindeln abgleichen
- Koppelzustand abgleichen
- Auswertung Master und Slave
- Neue Kopplung ein- oder alte Kopplung abschalten
- Winkelversatz einfahren

⚠ VORSICHT

Spindeln bewegen sich.

Für ausreichend Sicherheit sorgen.

Spindelkopplung abgleichen

```

0000 DIM SADR$(8) :REM For spindle syntax      (e.g. S1=)
;
; Adjust coupling data of spindles
0100 FOR SP% = 1 TO SD.SysChSpSRun.MaxSpNumber
0101   IF SD.SysChSpSRun.Sp[SP%].CpDataNotAdj <> 0 THEN
N102     SpAdjStr[SP%]
0103     SADR$ = SD.SysChSpSRun.SpeedStr
N104     SPCS([SADR$][SD.SysChSpSRun.Sp[SP%].SyncWindow])
N105     SPCE([SADR$][SD.SysChSpSRun.Sp[SP%].SyncErrorWindow])
N106     SPCD([SADR$][SD.SysChSpSRun.Sp[SP%].CoupleDistance])
0107   ENDIF
0108 NEXT
;
; Adjust coupling state
0200 FOR CP% = 1 TO SD.SysChSpSRun.PrgCpNumber
0201   IF SD.SysChSpSRun.Cp[CP%].NotAdjusted <> 0 THEN
0202     NADJ% = SD.SysChSpSRun.Cp[CP%].NotAdjusted
0203     MAST% = SD.SysChSpSRun.Cp[CP%].Master
0300     IF NADJ% = 1 THEN
; Incorrect master, delete coupling and generate new coupling
0301       IF SD.SysChSpSRun.Cp[CP%].CoupleOn THEN
N302         SPCC(CP=[CP%],MA=0)
N303         SPC_WAIT(CP=[CP%])
0304         INDEX% = 1
0305         SP% = SD.SysChSpSRun.Cp[CP%].Slave[INDEX%]
0310         WHILE SP% <> 0 DO
N311           SpAdjStr[SP%]
0312           SADR$ = SD.SysChSpSRun.SpeedStr
N313           SPCC(CP=[CP%],MA=[MAST%],[SADR$]1)
0314           INDEX% = INDEX% + 1
0315           IF INDEX% <= 8 THEN
0316             SP% = SD.SysChSpSRun.Cp[CP%].Slave[INDEX%]
0317           ELSE
0318             SP% = 0
0319           ENDIF
0320         END
N330         SPC_WAIT(CP=[CP%])
0331       ELSE
0332         SETERR("MTX error: Cp[.].NotAdjusted=1 und CoupleOn=FALSE")
0333       ENDIF
0334     ENDIF
0400     IF NADJ% = 2 THEN
; Incorrect slaves
0401       IF SD.SysChSpSRun.Cp[CP%].CoupleOn THEN
0410         FOR SP% = 1 TO SD.SysChSpSRun.MaxSpNumber
0411           CPNOTADJ% = SD.SysChSpSRun.Sp[SP%].CoupleNotAdj
0412           IF (CPNOTADJ% = 1) OR (CPNOTADJ% = 4) THEN
N413             SpAdjStr[SP%] ;Load syntax elements of the current spindle
0414             SADR$ = SD.SysChSpSRun.SpeedStr

```

Kanalsteuerung

```

N415             SPCC (CP=[CP%],MA=[MAST%],[SADR$]0)
N416             ENDIF
N417             NEXT
N418             SPC_WAIT (CP=[CP%])
N420             FOR SP% = 1 TO SD.SysChSpSRun.MaxSpNumber
N421                 CPNOTADJ% = SD.SysChSpSRun.Sp[SP%].CoupleNotAdj
N422                 IF CPNOTADJ% = 2 THEN
N423                     SpAdjStr[SP%] ;Load syntax elements of the current spindle
N424                     SADR$ = SD.SysChSpSRun.SpeedStr
N425                     SPCC (CP=[CP%],MA=[MAST%],[SADR$]1)
N426                 ENDIF
N427             NEXT
N428             SPC_WAIT (CP=[CP%])
N430             ELSE
N431                 SETERR("MTX error: Cp[].NotAdjusted=2 und CoupleOn=FALSE")
N432             ENDIF
N433         ENDIF
N500     IF NADJ% = 3 THEN
N501         ; New coupling
N502         IF SD.SysChSpSRun.Cp[CP%].CoupleOn THEN
N503             INDEX% = 1
N504             SP% = SD.SysChSpSRun.Cp[CP%].Slave[INDEX%]
N505             WHILE SP% <> 0 DO
N506                 SpAdjStr[SP%]
N507                 SADR$ = SD.SysChSpSRun.SpeedStr
N508                 SPCC (CP=[CP%],MA=[MAST%],[SADR$]1)
N509                 INDEX% = INDEX% + 1
N510                 IF INDEX% <= 8 THEN
N511                     SP% = SD.SysChSpSRun.Cp[CP%].Slave[INDEX%]
N512                 ELSE
N513                     SP% = 0
N514                 ENDIF
N515             END
N516             ELSE
N517                 Resolve coupling
N518                 SPCC (CP=[CP%],MA=0)
N519             ENDIF
N518             SPC_WAIT (CP=[CP%])
N519         ENDIF
N519     ;
N520     ; If required, position the phase offset of coupling
N521     ; Do not confuse with coupling distance
N522     IF SD.SysChSpSRun.Cp[CP%].CoupleOn THEN
N523         FOR SP% = 1 TO SD.SysChSpSRun.MaxSpNumber
N524             ; Is the phase offset correct?
N525             IF SD.SysChSpSRun.Sp[SP%].CoupleNotAdj AND 16 = 16 THEN
N526                 ; Adjust phase offset and speed
N527                 SpAdjStr[SP%] ;Load syntax elements of the current spindle
N528                 SADR$ = SD.SysChSpSRun.SpeedStr
N529                 SPEED = SD.SysChSpSRun.Sp[SP%].CouplePosOffsetSpeed
N530                 SPCP ([SADR$][SD.SysChSpSRun.Sp[SP%].CouplePosOffset],POSVEL[SPEED])
N531             ELSE
N532                 IF SD.SysChSpSRun.Sp[SP%].CoupleNotAdj = 8 THEN
N533                     ; Adjust phase offset
N534                     SpAdjStr[SP%] ;Load syntax elements of the current spindle
N535                     SADR$ = SD.SysChSpSRun.SpeedStr
N536                     SPCP ([SADR$][SD.SysChSpSRun.Sp[SP%].CouplePosOffset])
N537                 ENDIF
N538             ENDIF
N539         NEXT
N540         SPCP_WAIT (CP=[CP%])
N541     ENDIF
N542     ENDIF
N543     NEXT
N544     NEXT

```

Werkzeuge

Je nach Applikation kann die Abarbeitung des Werkzeugwechsels im Satzvorlauf sehr unterschiedlich aufgebaut sein. Im Werkzeugwechselprogramm kann über die Systemdaten zwischen Satzvorlauf und Programmabarbeitung unterschieden werden.

Abfrage im Werkzeugwechselprogramm

```

100 IF SD.SysChSRun[1].Active=1 THEN
110     ...
120     ...
130 ENDIF

```


Im Satzvorlauf wird der Werkzeugwechsel nicht real durchgeführt. Einige Daten müssen aber für die Berechnungen im Satzvorlauf gesetzt werden. Für die Bahnkorrektur betrifft dies unter anderem die DCT-Korrekturen. Das letzte aktive Werkzeug kann in einem Systemdatum gespeichert werden. Die E- und D-Korrekturen werden automatisch in den Systemdaten abgelegt.

Werkzeugdaten im Satzvorlauf

```

300 FOR I% = 1 TO EDGE%
350   SD.EdGeo = SD.ToolPre.UD.Ed[I%].Geo
360   SD.EdLife = SD.ToolPre.UD.Ed[I%].Life
370   SD.EdWear = SD.ToolPre.UD.Ed[I%].Wear
380   SD.EdOffset = SD.ToolPre.UD.Ed[I%].Offset
410   DCT(1,I%,0)=SD.EdGeo.L1 + SD.EdWear.L1 + SD.EdOffset.L1
420   DCT(2,I%,0)=SD.EdGeo.L2 + SD.EdWear.L2 + SD.EdOffset.L2
430   DCT(3,I%,0)=SD.EdGeo.L3 + SD.EdWear.L3 + SD.EdOffset.L3
440   DCT(4,I%,0)=SD.EdGeo.Rad + SD.EdWear.Rad + SD.EdOffset.Rad
450   DCT(5,I%,0)=SD.EdGeo.Ori
460 NEXT I%
810 SD.SysToolSRUN[1].ActTool.K1=1
810 SD.SysToolSRUN[1].ActTool.K2=SD.ToolPre.Hd.IKQ3

```

Im InitAdjPrg kann dann der Werkzeugwechsel durchgeführt werden.

Werkzeugwechsel

```

N10 M6 T[SD.SysToolSRUN[1].ActTool.K2]
100 IF SD.SysToolSRUN[1].TargDNr = 0 THEN
N50   D0
110 ELSE
N60   D[SD.SysToolSRUN[1].TargDNr]
120 ENDIF
150 IF SD.SysToolSRUN[1].TargEd = 0 THEN
N80   ED0
160 ELSE
N90   ED[SD.SysToolSRUN[1].TargEd]
170 ENDIF

```

NC-Funktionen

Ein Teil der modal wirkenden NC-Funktionen kann über den Adjust-Befehl abgeglichen werden. Teilweise werden modale NC-Funktionen auch im Anfahrtsatz berücksichtigt, siehe oben.

Abgleich NC-Funktionen

```

...
ADJ(G47)
ADJ(G1)
ADJ(G61)
ADJ(G8)
ADJ(RAD)
ADJ(G53.1)
ADJ(G94)
ADJ(G96)
ADJ(G16)
ADJ(G40)
...

```

Hilfsfunktionen

Hilfsfunktionen werden im Rechenlauf unterdrückt. Die anstehenden Hilfsfunktionen, wie z. B. Kühlflüssigkeit, können z. B. im TargetAdjustPrg ausgeführt werden.

Das Systemdatum "/SysChSRUN/AuxNotAdjusted" gibt an, dass noch nicht alle modal wirkenden Hilfsfunktionen abgeglichen sind.

Einzelachsenabgleich mit G77

Bei der Anfahrbewegung werden alle abzugleichenden Achsen gemeinsam verfahren. An manchen Maschinentypen ist dies nicht erwünscht. Hier müssen die Achsen bearbeitungsabhängig angefahren werden.

An einer Drehmaschine darf z. B. bei einer Innenbearbeitung die Z-Achse erst verfahren werden, wenn die X-Achse schon abgeglichen ist.

Im TargetAdjustPrg können einzelne Achsen mit dem NC-Befehl G77 abgeglichen werden. Im folgenden Beispiel wird eine Unterscheidung der Abgleichreihenfolge gezeigt. Abhängig einer SPS-Variable, die z. B. über M-Keys gesetzt werden kann, wird im TargetAdjustPrg die Abgleichreihenfolge der Achsen beeinflusst.

Achsenabgleich mit G77 im TargetAdjustPrg

```

WAIT
1   CASE @MODEG77% OF
1   LABEL 1
1       PRN#(0,"Mode 1: first X then Z")
          G77 X1 F2000
          G77 Z1 F2000
1   LABEL 2
1       PRN#(0,"Mode 2: first Z then X")
          G77 Z1 F2000
          G77 X1 F2000
1   LABEL 3
1       PRN#(0,"Mode 3: X and Z together")
          G77 X1 Z1 F2000
1   OTHERWISE
1       SETERR("Invalid startup mode G77")
1   ENDCASE
WAIT

```

5.3 Satzwechsel über High-Speed

5.3.1 Beschreibung

Funktion

Durch die Funktion Satzwechsel über "High-Speed-Signal" (HsBlkSwitch/HSB) lässt sich per High-Speed-Eingang oder einem Kundeneingang des Kanalinterfaces ein "vorzeitiger" Satzwechsel in der NC auslösen.

Auf diese Weise können z. B. externe Steuerungs-/Überwachungseinrichtungen eine aktive lineare Verfahrbewegung abbrechen. Anschließend führt die Steuerung die Bearbeitung mit dem nächsten programmierten Satz fort.

Die Funktion erlaubt:

- "fliegenden" Satzwechsel:
programmierte Endpunkte von Achsen werden nicht gelöscht.
Sind im Nachfolgesatz diese Achsen nicht mit neuen Endpunkten programmiert, übernimmt die Steuerung deren Endpunkte aus dem vorzeitig beendeten Satz und interpoliert diese Achsen mit.
Die aktuelle Geschwindigkeit im vorzeitig beendeten Satz wird am Satzwechsel nicht beeinflusst (Ausnahmen siehe unter [Kap. "Restriktionen" auf Seite 77](#)). Eine evtl. notwendige Geschwindigkeitsänderung leitet die NC erst im Nachfolgesatz, allerdings ohne Überprüfung des Achsprungvermögens ein.
- Satzwechsel mit "Restweg löschen":
programmierte Endpunkte von Achsen werden gelöscht.
Falls man im Nachfolgesatz diese Achsen nicht gezielt erneut programmiert, werden sie auch nicht weiter verfahren.
Die aktuelle Bahngeschwindigkeit des vorzeitig beendeten Satzes wird am Satzwechsel immer auf v=0 reduziert. Hierbei ist sowohl ein Downslope mit der aktuellen Beschleunigung, als auch ein Geschwindigkeitsprung auf v=0 per Parameter HSSTOP wählbar.

"Fliegenden" Satzwechsel können Sie z. B. dann anwenden, wenn entlang einer geraden Strecke in Abhängigkeit eines externen Ereignisses mit unterschiedlichen Vorschüben gefahren werden soll.

Satzwechsel mit "Restweg löschen" eignet sich z. B. bei Anwendungen, innerhalb derer ein druck- oder drehmomentgesteuertes "Fahren auf Festanschlag" erforderlich ist.

Beispiele finden Sie unter HsBlkSwitch (HSB) im "MTX Programmierhandbuch".

HINWEIS

Durch die Funktion wird die programmierte Kontur verändert! Das kann Schäden am Werkstück und/oder Werkzeug zur Folge haben!

Setzen Sie die Funktion nur dann ein, wenn sich das Werkzeug nicht im Eingriff befindet, oder Schäden am Werkstück und/oder Werkzeug ausgeschlossen werden können!

Restriktionen

- Sowohl im vorzeitig beendeten Satz, als auch im Nachfolgesatz muss eine lineare Verfahrbewegung programmiert sein (während aktivem G0 oder G1).
- Schnittstelle für schnelle Signalankopplung erforderlich ("Highspeed I/O").
- Wird der Satzwechsel über einen High-Speed-Eingang ausgelöst, muss dies in **PLC/DigIO/FuncAllocIn/DigIOAllocIn[1..8]** (MP 4075 00101) mit der Funktionalität "HsBlkSwitch/HSB" appliziert sein.
- *"Fliegender" Satzwechsel innerhalb der Betriebsart "Folgesatz" findet im Achsstillstand statt, wenn*
 - zwischen dem abzubrechenden Satz und dem Nachfolgesatz ein Konturknick > 90 Grad existiert
 - der HsBlkSwitch-Satz aufgrund der momentan wirksamen Interpolationsart immer mit v=0 enden würde (bei aktivem G0)
 - die Funktion "Genauhalt" aktiv ist (G61)
 - der Nachfolgesatz wegen einer Zusatzprogrammierung mit v=0 beginnt (z. B. KvProg/KVP, FeedForward/FFW oder Writeld/WID)
 - die Funktion "SHAPE" aktiv ist (siehe unter G8 bzw. G9).
- In den Betriebsarten "Einzelsatz" und "Einzelschritt" wird trotz programmiertem "fliegender" Satzwechsel auf v=0 abgebremst, da die Steuerung pro NC-Start nur jeweils einen einzelnen Programmsatz abarbeitet.
- In den Betriebsarten "Programmsatz" und "Handeingabe" wird trotz programmiertem "fliegender" Satzwechsel auf v=0 abgebremst **und der Restweg gelöscht!** Ein nachfolgender NC-Satz mit Inkrementalprogrammierung führt deshalb nicht zum zuvor programmierten Endpunkt!

Relevante NC-Funktionen

HsBlkSwitch/HSB	"Satzwechsel über High-Speed-Signal" für die Dauer des HsBlkSwitch-Satzes aktivieren. Die gewünschten Auslösebedingungen und die Auswahl des Digital-Eingangs sind per Parameter HS einstellbar:	
	HSx=0:	Low-Pegel (x: Nummer des Eingangs)
	HSx=1:	High-Pegel
	-oder-	
	per Parameter CI einstellbar:	
	CIx=0:	Low-Pegel (x: Nummer des Eingangs)
	CIx=1:	High-Pegel
	Parameter HSSTOP beeinflusst die Wirkungsweise der Funktion:	
	ohne HSSTOP:	"fliegender Satzwechsel". Keine Geschwindigkeitsänderung am Satzübergang.
	HSSTOP=0:	Satzwechsel mit "Restweg löschen" und Downslope auf v=0.
	HSSTOP=-1:	Satzwechsel mit "Restweg löschen" und Geschwindigkeitssprung auf v=0.
	Ändert sich innerhalb der Bearbeitungszeit für den HsBlkSwitch-Satz der Pegel des verwendeten High-Speed-Eingangs nicht, wird der Satz entsprechend seiner Programmierung (ohne oder mit Parameter HSSTOP) bis zur Endposition verfahren. Danach arbeitet die NC den nächsten Programmsatz ab.	

Tab. 5-19: Relevante INC-Funktionen

Relevante IF-Signale

HSI-0	schneller Digitaleingang, einstellbar per MP DigIOAllocIn[1..8] (4075 00101). Satzwechsel auslösen bei z. B. HSB(HS1=1). Der Eingang wird im Interpolatortakt abgefragt.
qCh_Customx	Kundeneingang x Satzwechsel auslösen bei z. B. HSB(CI1=1)

Tab. 5-20: Relevante IF-Signale

Relevante Maschinenparameter (MP)

PLC/DigIO/ FuncAllocIn/ DigIOAllocIn[1..8] (4075 00101)	Zuordnung der Digitaleingänge Legt u.a. fest, welcher schnelle Digital-Eingang der Funktion "HsBlkSwitch/HSB" zugeordnet wird.

Tab. 5-21: Relevante Maschinenparameter

*Beispiel:***"fliegender" Satzwechsel**

Die resultierende Verfahrbewegung zum Endpunkt (X50/Y20) kann sich in Abhängigkeit der Signalwechsel am High-Speed-Eingang zwar ändern, der Endpunkt aber selbst bleibt unverändert.

Beim Satzwechsel bleibt die programmierte Geschwindigkeit erhalten.

N05G1F1000 X0 Y5	Endpunkt: X0 Y5
N10HSB(HS1=1) G90 X50	Endpunkt: X50 Y5
N20HSB(HS1=0) G91 Y10	Endpunkt: X50 Y15 (abs. Y5+inkr. Y10)
N30 G91 Y5	Endpunkt: X50 Y20 (abs. Y15+inkr. Y5)

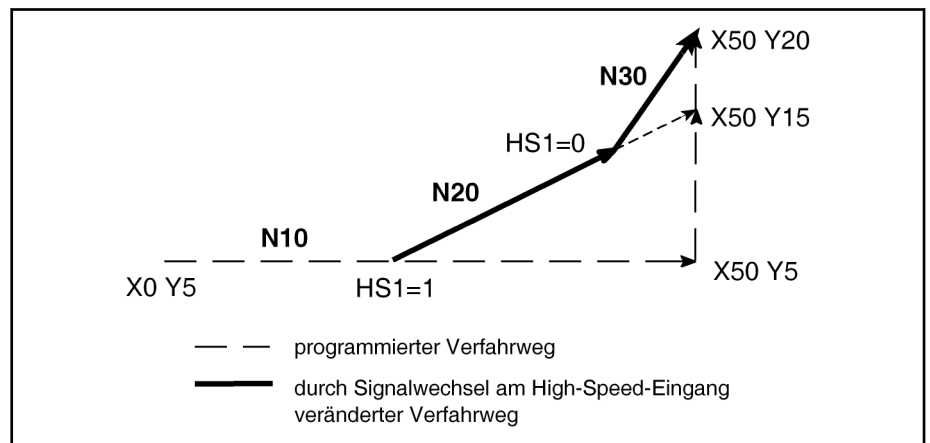


Abb. 5-12: *Beispiel "fliegender" Satzwechsel*

*Beispiel:***Satzwechsel mit "Restweg löschen"**

Der programmierte Endpunkt (X50/Y20) bleibt nur dann erhalten, wenn kein Signalwechsel am High-Speed-Eingang stattfindet. In allen anderen Fällen ist der Endpunkt vom Signalwechsel am High-Speed-Eingang abhängig.

Beim Satzwechsel wird per Downslope auf $v=0$ abgebremst.

N05G1F1000 X0 Y5	Endpunkt: X0 Y5
N10HSB(HS1=1) HSSTOP=0 G90 X50	Endpunkt: externes Ereignis bei Xp50
N20HSB(HS1=0)HSSTOP=0 G91 Y10	Endpunkt: externes Ereignis bei Yp10
N30 G91 Y5	Endpunkt: vom letzten externen Ereignis inkremental um Y5 "entfernt".

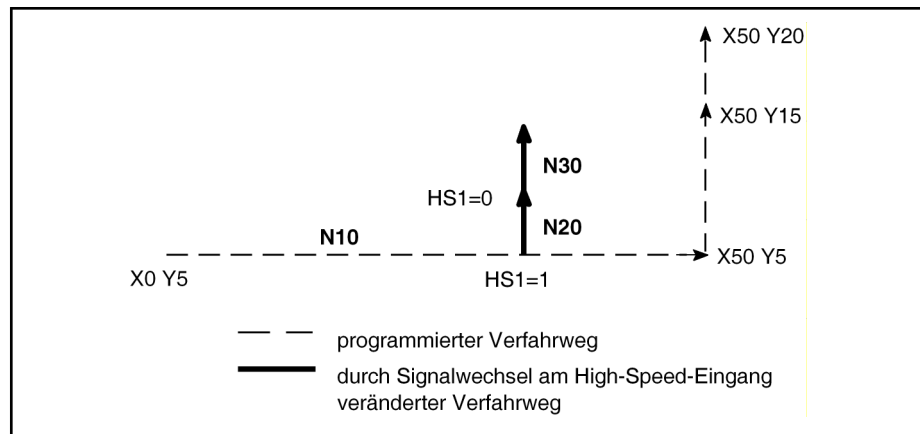


Abb. 5-13: Beispiel - Satzwechsel mit "Restweg löschen"

5.3.2 Handlungsanweisung: Satzwechsel über High-Speed-Signal mit dem Löschen des Restwegs

Applizieren:

HINWEIS**Schäden an der Maschine oder am Werkstück möglich!**

Während der Bearbeitung darf kein Steuerungs-Reset ausgelöst werden.

1. Stellen Sie sicher, dass alle High-Speed-Eingänge, die Sie im Zusammenhang mit der Funktion "Satzwechsel über High-Speed-Signal" nutzen wollen, in MP 4075 00101 mit dem Wert "0003" belegt sind.
2. Sofern Sie den Konfigurationsparameter geändert haben, Steuerungs-Reset auslösen.

Aktivieren:

Voraussetzung:

Es ist eine lineare Interpolationsart aktiv (G0 oder G1).

HsBlkSwitch/HSB in Verbindung mit den Parametern HS oder CI und ggf. HSSTOP programmieren (siehe "Relevante NC-Funktionen").

Die Funktion ist bis zum ausgelösten Satzwechsel, höchstens jedoch für die Dauer des HsBlkSwitch-Satzes aktiv und wird anschließend, sofern kein neuer HsBlkSwitch-Satz unmittelbar folgt, automatisch ausgeschaltet.

Durch die Funktion "Satzwechsel über High-Speed-Signal" (HsBlkSwitch/HSB) lässt sich per High-Speed-Eingang oder einem Kundeneingang des Kanalinterfaces ein "vorzeitiger" Satzwechsel in der NC auslösen.

IW-Engineering / Configuration: Dig-IO parametrieren

Für die Nutzung der Funktion "Satzwechsel über High-Speed-Signal" den gewünschten Digitaleingang über den Parameter **DigIOAllocIn** "Zuordnung der Digitaleingänge [1..8]" (4075 00101) den Wert **"Satzwechsel über HS-Signal"** zuweisen.



- Einstellen der verwendeten Hardware für Digital-IO über den Parameter **DigIOHw** "Hardware-Auswahl" (4075 00200)
- In den Parameter **DigIOStartAddrIn** "Anfangsadresse der Digitaleingänge" (4075 00105) die Anfangsadresse für Dig-IO eintragen.

	Documentation	Instruction
Instruction:		Editieren von Maschinenparametern
Documentation:	MTX Maschinenparameter	Digitale IO's parametrieren

IW-Operation / NC-Programmierung:**NC-Satz programmieren (Restweg löschen)**

Die NC-Funktion **HsBlkSwitch(HS<x>=<y>,HSSTOP=<z>)** bzw.

HSB(HS<x>=<y>,HSSTOP=<z>) "Satzwechsel mit Abbruch" programmieren.

- <x>:** Nummer des Highspeed-Signals (1..8)
- <y>:** Zum Satzwechsel erforderlicher logischer Signalzustand (0 oder 1)
- <z>:** Abbremsart beim Auftreten des Ereignisses
 ("0" = Abrampen auf v=0 mit max. Bremsbeschleunigung und
 "1" = Geschwindigkeitssprung auf V=0)



- Die Funktion erfordert eine lineare Verfahrbewegung (G0 oder G1) sowohl im "HsBlkSwitch"-Satz, als auch in dessen Nachfolgesatz.
- Die Funktion muss mit einer Weginformation und kann mit anderen Wegbedingungen zusammen geschrieben werden.
- Genauhalt ist aktiv, d.h. G0(IPS...) oder G1(IPS...).
- Schnittstelle für schnelle Signalankopplung erforderlich ("Highspeed I/O").

	Documentation	Example
Example:		Beispiel für fliegender Satzwechsel
Documentation:	MTX Programmierhandbuch	Fliegender Satzwechsel mit Restweg löschen programmieren

Digital Interface: Per Digitaleingang den Satzwechsel auslösen

Für die gewünschte Art des Satzwechsels das entsprechende logische Signal (0 oder 1) an den Digitaleingang geben.



Es ist möglich einen Digitaleingang einem Interface-Signal **qAx_Custom1 ... 8** "Kundeneingang 1...8" zuzuweisen.

	Documentation	Instruction
Instruction:		Editieren von SPS-Signalen
Documentation:	MTX SPS-Interface	Satzwechsel auslösen

5.3.3 Handlungsanweisung: Satzwechsel über High-Speed-Signal mit fliegender Wechsel

Durch die Funktion "Satzwechsel über High-Speed-Signal" (HsBlkSwitch/HSB) lässt sich per High-Speed-Eingang oder einem Kundeneingang des Kanalinterfaces ein "vorzeitiger" Satzwechsel in der NC auslösen.

IW-Engineering / Configuration: Dig-IO parametrieren

Für die Nutzung der Funktion "Satzwechsel über High-Speed-Signal" den gewünschten Digitaleingang über den Parameter **DigIOAllocIn** "Zuordnung der Digitaleingänge [1..8]" (4075 00101) den Wert **"Satzwechsel über HS-Signal"** zuweisen.



- Einstellen der verwendeten Hardware für Digital-IO über den Parameter DigIOHw "Hardware-Auswahl" (4075 00200)
- In den Parameter DigIOStartAddrIn "Anfangsadresse der Digitaleingänge" (4075 00105) die Anfangsadresse für Dig-IO eintragen.

	Documentation	Instruction
Instruction:		Editieren von Maschinenparametern
Documentation:	MTX Parameterbeschreibung	Digitale IOs parametrieren

IW-Operation / NC-Programmierung: NC-Satz programmieren (fliegender Wechsel)

Die NC-Funktion **HsBlkSwitch(HS<x>=<y>)** bzw. **HSB(HS<x>=<y>)** "Fliegender Satzwechsel" programmieren.

<x>: Nummer des Highspeed-Signals (1..8)

<y>: Zum Satzwechsel erforderlicher logischer Signalzustand (0 oder 1)



- Die Funktion erfordert eine lineare Verfahrbewegung (G0 oder G1) sowohl im "HsBlkSwitch"-Satz, als auch in dessen Nachfolgesatz.
- Die Funktion muss mit einer Weginformation und kann mit anderen Wegbedingungen zusammen geschrieben werden.
- Genauhalt ist aktiv, d. h., G0(IPS...) oder G1(IPS...).
- Schnittstelle für schnelle Signalankopplung erforderlich ("Highspeed I/O").

	Documentation	Example
Example:		Beispiel fliegender Satzwechsel
Documentation:	MTX Programmierhandbuch	Fliegender Satzwechsel programmieren

Digital Interface: Per Digitaleingang den Satzwechsel auslösen

Für die gewünschte Art des Satzwechsels das entsprechende logische Signal (0 oder 1) an den Digitaleingang geben.



Es ist möglich einen Digitaleingang einem Interface-Signal **qAx_Custom1 ... 8** "Kundeneingang 1..8" zuzuweisen.

	Documentation	Instruction
Instruction:		Editieren von SPS-Signalen
Documentation:	MTX SPS-Interface	Satzwechsel auslösen

5.4 Programmfortführung mit Satzvorlauf nach Programmabbruch

5.4.1 Beschreibung

Funktion

Wird ein NC-Programm während der Abarbeitung abgebrochen, kann das Programm an der letzten, vordefinierten Stelle wieder fortgeführt werden. Diese Stelle wird im NC-Programm durch Programmierung der NC-Funktion "RCB" festgelegt. Nachdem der Interpolator diesen Satz abgearbeitet hat, kann diese Stelle im Satzvorlauf als Zielsatz verwendet werden.

Werden mehrere Stellen zum Wiederanfahren mit RCB im NC-Programm programmiert, wird immer die zuletzt erfolgreich abgearbeitete Stelle als Vorgabe für den Satzvorlauf verwendet. Es ist möglich, durch einen bestimmten Parameterwert die zuletzt gemerkte Stelle zu löschen, so dass nach Abarbeitung dieses Satzes ein Satzvorlauf an die zuvor gemerkte Stelle nicht mehr möglich ist.

Es können optional bis zu 3 CPL-Bedingungen der RCB-Funktion übergeben werden. Diese Bedingungen werden zum Satzvorbereitungszeitpunkt ausgewertet und zum Satzarbeitungszeitpunkt intern gespeichert. Der Satzvorlauf verwendet dann dieselben Bedingungen, um zu entscheiden, ob der Zielsatz gefunden wurde. Im Unterschied zum normalen Satzvorlauf mit manueller Angabe des Zielsatzes ist es damit möglich, gezielt erfolgreich abgearbeitete Schleifendurchläufe zu überspringen oder das Finden des Zielsatzes nur unter gewissen Voraussetzungen zuzulassen.

Restriktionen

Die Auswertung der Bedingungen wird zum Satzvorbereitungszeitpunkt bzw. zum Zeitpunkt des Satzvorlaufes durchgeführt. Unter Umständen muss mit der WAIT-Funktion vor der Programmierung von RCB die Satzvorbereitung mit dem IPO synchronisiert werden, damit der Zustand der CPL-Variablen konsistent ist. Dies gilt ebenso für den Satzvorlauf. Hier sollten eventuelle externe Signale bereits vor dem Start des Satzvorlaufes korrekt anliegen, da der Zeitpunkt der Auswertung von RCB hier nur von der Abarbeitungsgeschwindigkeit der Steuerung abhängt.

Das direkte aktivierende Programmieren von RCB ist nicht im Handeingabebetrieb möglich. Trotzdem kann ein aktivierendes RCB in einem im Handeingabebetrieb aufgerufenen Programm verwendet werden.

Es kann immer nur eine Stelle pro Kanal aktiv sein, welche im Satzvorlauf als Zielsatz verwendet werden kann.

Aufrufe von aktivierenden RCB-Funktionen in ASUPs sind erlaubt, aber wirkungslos.

Bei Verwendung von Bedingungen muss das Programm gelinkt worden sein.

Relevante NC-Funktionen

RCB(1, {Bed. 1}, {Bed. 2}, {Bed. 3}) bzw. ReentContBlk(1, {Bed. 1}, {Bed. 2}, {Bed. 3})	Markiert eine Stelle, die der Satzsuchlauf finden kann. Bedingung 1 bis 3 sind optional. Die Gültigkeit einer eventuell zuvor gemerkten Stelle wird überschrieben.
RCB(0) bzw. ReentContBlk(0)	Löscht die zuletzt markierte Stelle. Ein Satzvorlauf an die letzte aktivierende Stelle ist nicht mehr möglich.

Tab. 5-22: Relevante NC-Funktionen

5.4.2 Aktivieren

Programmieren von **RCB(1)** (ohne Bedingung) bzw. von **RCB(1, Bed. 1, Bed. 2, Bed. 3}}** (mit max. 3 optionalen Bedingungen)

5.4.3 Deaktivieren

Programmieren von **RCB(0)**.

5.4.4 Handlungsanweisungen

Wurde das Programm nach der Abarbeitung einer aktivierenden RCB-Programmierung und vor einer deaktivierenden RCB-Programmierung abgebrochen, ist es möglich, durch IndraWorks Operation einen Satzvorlauf mit Verwendung der zuletzt gemerkten Stelle auszulösen.

Dazu muss das abgebrochene Hauptprogramm angewählt werden und der Satzvorlauf mit Verwendung der intern gespeicherten Stelle gestartet werden. Wird der Zielsatz gefunden, kann das Programm an dieser Stelle mittels NC-Start fortgeführt werden. Eine eventuelle Fehlermeldung zeigt an, dass entweder der Zielsatz nicht gefunden wurde oder der Zielsatz gefunden wurde, aber eine der angegebenen Bedingungen nicht erfüllt war.

Bemerkung 1: Zielsatz nicht gefunden:

Kann nur auftreten, wenn das zuletzt abgearbeitete RCB in einem Unterprogramm, welches im Satzvorlauf nicht aufgerufen wurde, programmiert war.

Bemerkung 2: Zielsatz gefunden, aber eine Bedingung war nicht erfüllt:

Es werden im Satzvorlauf alle angegebenen Bedingungen geprüft, ob der Zustand im Satzvorlauf identisch ist mit dem Zustand, welcher zuvor bei der normalen Abarbeitung bestand. Ist dies z. B. wegen veränderter Eingangssignale oder Variablenbelegungen der Fall, muss der ursprüngliche Variablenzustand wiederhergestellt werden, damit der Satzvorlauf den Zielsatz findet und anhält.

6 Dynamik und Geschwindigkeitsführung

6.1 Skalieren von Rundachsgeschwindigkeiten

6.1.1 Konfiguration von Skalierungsfaktoren für Rundachsgeschwindigkeiten

Beschreibung

Funktion

Stellt sicher, dass die Steuerung Geschwindigkeiten synchroner Rund- und Endlosachsen sowohl bei metrischen, als auch bei Inch-bezogenen Programmen korrekt vorgibt.

Vorhandene "landesspezifische" Teileprogramme können dadurch ohne Modifikation abgearbeitet werden.

Durch die Skalierung der Rundachsgeschwindigkeit wird festgelegt, wie die Wegvorgabe einer Rund-/Endlosachse in Grad im Verhältnis zu einer Linearachse mit einer Wegvorgabe in mm bzw. Inch geschwindigkeitsmäßig gewichtet werden soll.

In Abhängigkeit von G70/G71 lässt sich dabei über Maschinenparameter einstellen, ob 1 Grad Winkelvorgabe wie 1 mm oder 1 Inch Wegvorgabe behandelt werden soll. In Abhängigkeit der Einstellung ergibt sich somit bei gleicher programmierter Bahngeschwindigkeit eine unterschiedliche Bewegungsdauer.

Dazu wird für G70 (Inch-Programmierung) und G71 (Metrisch-Programmierung) getrennt eingestellt, ob für die Geschwindigkeitsberechnung 1 Grad wie 1 mm oder, wie 1 Inch behandelt werden soll.

1. Europäische Einstellung:

[1]: 1

[2]: 1

- Sowohl bei Metrisch- als auch bei Inch-Programmierung entspricht 1 Grad stets 1 mm. Ist bei G71 die Rundachse allein programmiert, wird der Vorschub in Grad/min interpretiert.
- Bei G70 und alleiniger Rundachsenprogrammierung verfährt die Rundachse um den Faktor 25.4 zu schnell (Vorschubwert wird in Grad/min interpretiert,

Beispiel:

$F100 = 100 \text{ Inch/min} = 2540 \text{ mm/min} = 2540 \text{ Grad/min}$.

- Fahren Rundachse und Linearachse zusammen, werden beide bei G71 gleich gewichtet.

Beispiel:

Linearachse verfährt 100 mm, Rundachse verfährt 100 Grad, beide Achsen verfahren mit 0.7071^* programmierten Vorschub in [mm/min] bzw. [Grad/min], da beide Verfahrenswege gleich groß betrachtet werden.

- Verfahren Linearachse um 100 Inch und Rundachse um 100 Grad bei G70, fährt die Linearachse mit nahezu 100% der programmierten Geschwindigkeit, da der Weg der Linearachse um den Faktor 25.4 größer als der der Rundachse gesehen wird.

2. Amerikanische Einstellung:

[1]: 2

[2]: 2

- Sowohl bei Metrisch- als auch bei Inch-Programmierung entspricht 1 Grad stets 1 Inch. Ist bei G70 die Rundachse alleine programmiert, wird der Vorschub in Grad/min interpretiert.
- Bei G71 und alleiniger Rundachsenprogrammierung verfährt die Rundachse um den Faktor 25.4 zu langsam (Vorschubwert wird nicht in Grad/min interpretiert,
Beispiel:
 $F100=100 \text{ mm/min}=3.937 \text{ Inch/min}=3.397 \text{ Grad/min}$).
- Fahren Rundachse und Linearachse zusammen, werden beide bei G70 gleich gewichtet.
Beispiel:
Linearachse verfährt 100 Inch, Rundachse verfährt 100 Grad, beide Achsen verfahren mit $0.7071 \cdot$ programmierten Vorschub in [Inch/min] bzw. [Grad/min], da beide Verfahrwege gleich groß betrachtet werden.
- Verfahren bei G71 Linearachse um 100 mm und Rundachse um 100 Grad, fährt die Rundachse mit nahezu 100% der programmierten Geschwindigkeit, da der Weg der Rundachse um den Faktor 25.4 größer als der der Linearachse gesehen wird.

3. Allgemeine Einstellung: [1]: 2

[2]: 1

- Sowohl bei G70 als auch bei G71 wird der Vorschub bei alleiniger Programmierung der Rundachse in Grad/min interpretiert.
- Fahren Rundachse und Linearachse zusammen, werden beide bei G70 und G71 gleich gewichtet.

Beispiel:

G70

Linearachse verfährt 100 Inch, Rundachse verfährt 100 Grad, beide Achsen verfahren mit $0.7071 \cdot$ programmierten Vorschub in [Inch/min] bzw. [Grad/min].

Beispiel:

G71

Linearachse verfährt 100 mm, Rundachse verfährt 100 Grad, beide Achsen verfahren mit $0.7071 \cdot$ programmierten Vorschub in [mm/min] bzw. [Grad/min].

Bedeutung der Parameterwerte:

1	1 Grad wird wie 1 mm gewichtet
2	1 Grad wird wie 1 Inch gewichtet

Bedeutung der Einzelparameter:

[1]	Skalierung für G70
[2]	Skalierung für G71

Restriktionen

- Die Funktion FeedAd (FAD) darf nicht aktiv sein ($\Rightarrow$ alle synchronen Achsen eines Kanals werden für die Berechnung des Bahnvorschubes berücksichtigt).
Um sicherzustellen, dass nach Hochlauf und nach Grundstellung Feed-Ad abgewählt ist, empfehlen wir "FAD()" in MP 7060 00020 (Grundstellung) mit aufzunehmen.

- Die Funktion wirkt nur für synchrone Achsen vom Achsbewegungstyp "rund" oder "endlos" (⇒ MP 1003 00004).
- Die Funktion bezieht sich ausschließlich auf Geschwindigkeiten und nicht auf Positionen.

Relevante NC-Funktionen

FeedAd(...)	Berücksichtigung aller Kanalachsen bei der Vorschubbildung

Tab. 6-1: Relevante NC-Funktionen

Relevante Maschinenparameter (MP)

7040 00110	Skalierung der Rundachsgeschwindigkeit bei G70/G71.

Tab. 6-2: Relevante Maschinenparameter

Handlungsanweisung: Skalieren von Rundachsgeschwindigkeiten

Applizieren:

HINWEIS

Schäden an der Maschine oder am Werkstück möglich!

Während der Bearbeitung darf keine Systemgrundstellung ausgelöst werden.

1. MP 7040 00110 anpassen

Hierbei gilt:

- Steuerung befindet sich an einem Standort, an dem ausschließlich metrische Teileprogramme eingesetzt werden (G71 ist aktiv; z.B. innerhalb Europas):
7040 00110 [1]: 1
7040 00110 [2]: 1
- Steuerung befindet sich an einem Standort, an dem ausschließlich Inch-bezogene Teileprogramme eingesetzt werden (G70 ist aktiv; z.B. innerhalb USA):
7040 00110 [1]: 2
7040 00110 [2]: 2
- Steuerung befindet sich an einem Standort, an dem sowohl metrische, als auch Inch-bezogene Teileprogramme eingesetzt werden:
7040 00110 [1]: 2
7040 00110 [2]: 1

2. Systemgrundstellung auslösen

Ermöglicht die Verwendung von metrischen und nicht metrischen Teileprogrammen bei Rundachsen

IW-Engineering / Configuration: Rundachsgeschwindigkeit anpassen

- Parameter **FactRotInch** "Skalierungsfaktor für nicht metr. Rundachsgeschw. (7040 00110[1])" und **FactRotMetr** "Skalierungsfaktor für metr. Rundachsgeschw. (7040 00110[2])"
 - G70 aktiv
 - **RotInch** (7040 00110[1]): 1 Grad = 1 inch
 - **RotMetr** (7040 00110[2]): 1 Grad = 1 inch
 - G71 aktiv
 - **RotInch** (7040 00110[1]): 1 Grad = 1 mm

- **RotMetr** (7040 00110[2]): 1 Grad = 1mm
- Bei Einsatz von metrischen und inch-bezogenen Teileprogramme
 - **RotInch** (7040 00110[1]): 1 Grad = 1 inch
 - **RotMetr** (7040 00110[2]): 1 Grad = 1 mm



Die NC-Funktion **FeedAd** (..) darf nicht aktiv sein.

6.1.2 Programmierung von Skalierungsfaktoren für Rundachsgeschwindigkeiten

Beschreibung

Funktion

Für die Vorschubberechnung wird die (virtuelle) Bahnlänge **L** der programmierten Sätze berechnet. Die gefahrene Geschwindigkeit einer Achse **va** ergibt sich dann als Anteil der Bahnlänge **La** geteilt durch die gesamte Bahnlänge **L** mal dem programmierten Vorschub **v**.

$$v_a = L_a / L * v$$

Wird für eine Bewegung die Bahnlänge berechnet, verwendet die MTX die geometrische Distanz ($\sqrt{A_1^2 + A_2^2 + A_3^2 \dots}$). Dabei wird implizit angenommen, dass sowohl lineare Achsen als auch Rundachsen ein rechtwinkliges Koordinatensystem aufspannen.

Beispiel:

Berechnung Vorschub linearer Achsen

Achse X fährt [300]mm, Achse Y fährt 400[mm], programmierter Vorschub 2000[mm/min]

$$\text{Bahnlänge } L = \sqrt{300^2 + 400^2} = 500[\text{mm}]$$

$$\text{Achse X: } v_a = 300/500 * 2000 = 1200[\text{mm/min}]$$

$$\text{Achse Y: } v_a = 400/500 * 2000 = 1600[\text{mm/min}]$$

Rundachsen, rotatorische Koordinaten und die Orientierung werden in Grad programmiert. Bei der Bahnlängenberechnung wird nun 1 Grad = 1 mm gesetzt. Geometrisch betrachtet, entspricht das einem Wirkradius von 57,3 mm. Das bedeutet, würde ein Werkstück 57,3 mm entfernt von der Drehachse eines Rundtisches montiert, bewegt es sich bei Drehung der Rundachse mit exakt dem programmierten Bahnvorschub auf der Kreisbahn.

Beispiel:

Berechnung Vorschub linearer und Rundachsen

Achse X fährt [100]mm, Achse C fährt 50[Grad], programmierter Vorschub 500[mm/min]

$$\text{Bahnlänge } L = \sqrt{100^2 + 50^2} \approx 111,8[\text{mm}]$$

$$\text{Achse X: } v_a = 100/111,8 * 500 \approx 447,2[\text{mm/min}]$$

$$\text{Achse C: } v_a = 50/111,8 * 500 \approx 223,6[\text{Grad/min}]$$

In bestimmten Fällen soll der Anteil der Rundachsenbewegung am Vorschub stärker (wenn die sich die Achsen zu schnell bewegen) oder weniger stark (wenn die exakte Orientierung nicht von Bedeutung ist) gewichtet werden.

Aus diesem Grund kann Rundachsen/-koordinaten ein zusätzlicher Faktor definiert werden, um den Einfluss der Achse auf den Vorschub zu modifizie-

ren ($\sqrt{L_1^2 + L_2^2 + L_3^2 + (R_1 \cdot F_1)^2 + (R_2 \cdot F_2)^2 \dots}$). Ist dieser Faktor gleich 1, gilt $1^\circ = 1 \text{ mm}$. Das entspricht dem Wirkradius von 57,3 mm.

Mit der NC-Funktion ActRadFact kann dieser Faktor direkt im Teilprogramm programmiert werden. Zusätzlich besteht die Möglichkeit in den Konfigurationsparameter den Wirkradiusfaktor auf Inch/Metrisch ([Kap. 6.1.1 "Konfiguration von Skalierungsfaktoren für Rundachsgeschwindigkeiten" auf Seite 85](#)) anzupassen.

Beispiel:

Berechnung Vorschub linearer und Rundachsen mit ARF (verringert)

Achse X fährt [100]mm, Achse C fährt 50[Grad], programmierter Vorschub 500[mm/min], ARF(C0.5)

Bahnlänge L = $\sqrt{100^2 + (50 \cdot 0,5)^2} \approx 103,1[\text{mm}]$

Achse X: $v_a = 100/103,1 \cdot 500 \approx 485,9[\text{mm/min}]$

Achse C: $v_a = 50/103,1 \cdot 500 \approx 242,5[\text{Grad/min}]$

Beispiel:

Berechnung Vorschub linearer und Rundachsen mit ARF (erhöht)

Achse X fährt [100]mm, Achse C fährt 50[Grad], programmierter Vorschub 500[mm/min], ARF(C3)

Bahnlänge L = $\sqrt{100^2 + (50 \cdot 3)^2} \approx 180,3[\text{mm}]$

Achse X: $v_a = 100/180,3 \cdot 500 \approx 277,3[\text{mm/min}]$

Achse C: $v_a = 50/180,3 \cdot 500 \approx 138,7[\text{Grad/min}]$

Restriktionen

- Kein Wirkradiusfaktor darf negativ sein.
- Nach Systemstart sind alle Wirkradiusfaktoren von ActRadFact auf 1 gesetzt. Es wirken wieder die in den Maschinenparametern konfigurierten Faktoren.
- Wird eine Achse aus dem Kanal entfernt, geht aus der Wirkradiusfaktor verloren.
- Ist der Wirkradius einer Achse/Koordinate gleich 0, darf keine Bewegung programmiert werden, die nur Anteile dieser Achse/Koordinate enthält.
- Wird die aktive Transformation geändert, so werden die Wirkradiusfaktoren gleicher Koordinaten übernommen. Alle anderen Wirkradiusfaktoren haben wieder den Wert 1.
- Die Funktion wirkt nur für synchrone Achsen vom Achsbewegungstyp "rund" oder "endlos" (MP 1003 00004) bzw. auf rotatorische Koordinaten und die Orientierung.
- Die Funktion bezieht sich ausschließlich auf Geschwindigkeiten und nicht auf Positionen.
- Funktion wirkt nicht bei Eilgang.

Relevante NC-Funktionen

ActRadFact , ARF	Programmieren eines Wirkradiusfaktors

Tab. 6-3: Relevante NC-Funktionen

Relevante Maschinenparameter (MP)

7040 00110	Skalierung der Rundachsgeschwindigkeit bei G70/G71.

Tab. 6-4: Relevante Maschinenparameter

Handlungsanweisung**Fall 1**

Problem Orientierung im Teileprogramm ändert sich sehr abrupt -> Rundachsen fahren ruckartig und zu schnell

Lösung Wichtung der Rundachsen in den kritischen Bereichen erhöhen

Programm:

N0010 G1 F800

```
...
N1010 X563.3 Y325.2 A34.1 C270.0
N1020 X254.1 Y325.2 A34.1 C270.0
N1030 X253.3 Y326.1 A34.1 C225.0 ARF(C2.5) ; Scaling of c-factor 2.5
N1040 X252.6 Y327.2 A34.1 C180.0
N1050 X252.6 Y414.4 A34.1 C180.0 ARF() ; Scaling of C again on 1.0
...
```

Ergebnis Die MTX fährt langsamer um die Kurve, die Rundachsen bewegen sich sanfter.

Fall 2

Problem Orientierung im Teileprogramm ist bei Positionierbewegung relativ unwichtig - > Rundachsen bremsen die Bewegung aus

Lösung Wichtung der Rundachsen während der Positionierbewegung verringern

Programm:

```
...
N1010 G1 F1000 X550 Y220 A0 C0 ARF(A0.1, C0.1) ; Scaling reduced
N1020 F355 X554.1 Y225.2 A34.1 C270.0
N1030 X556.1 Y227.2 A34.1 C270.0 ARF() ; Scaling again on 1.0
...
```

Ergebnis Die MTX fährt schneller an die Zielposition.

6.2 Bahnlängen- und Kurvenparameterinterpolation**6.2.1 Beschreibung****Funktion**

Für Konturelemente mit Kurvenparameter – das sind die Spline-Typen 0 bis 4 und die Rundungs-Splines (Syntax SCO) – kann an Stelle der Bahnlängeninterpolation eine Kurvenparameterinterpolation gewählt werden. Die Kanalkoordinaten folgen der Bahn dann nicht synchron zum Weg s, sondern synchron zum Kurvenparameter w. Eine aktivierte Kurvenparameterinterpolation bleibt ohne Wirkung bei den Bahntypen ohne Kurvenparameter (Gerade, Helix)

Ein Beispiel zeigt nachfolgende Abbildung. Bei Kurvenparameterinterpolation ergibt sich eine xy-Geschwindigkeit proportional zu den eingezeichneten Kurvenpunktabständen.

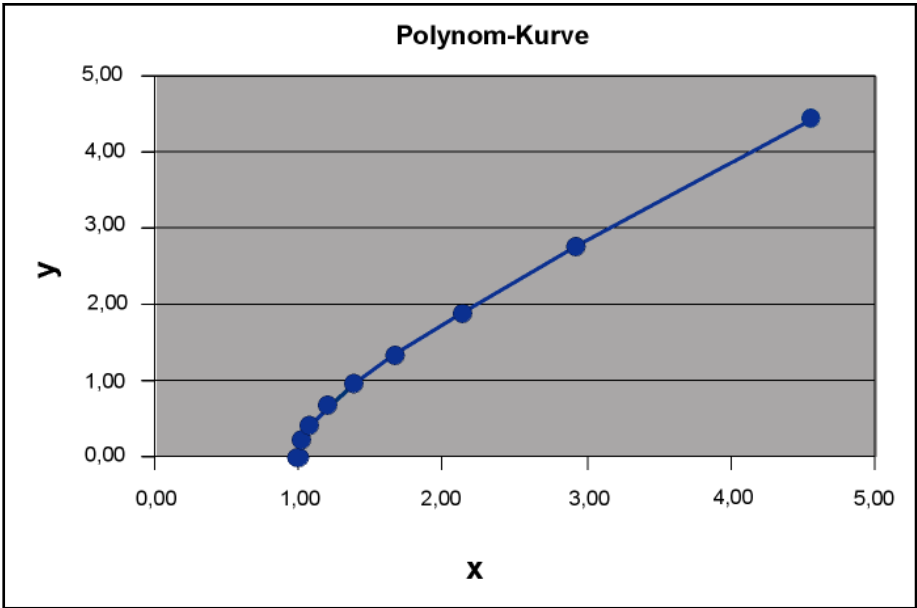


Abb. 6-1: Polynom-Kurve
Polynom-Kurve beschrieben durch

$$x(w) = \frac{1+w^2}{1-w^2} \quad y(w) = \frac{2w}{1-w^2} \quad w \in [0, 0.8]$$

Eingezeichnet sind die Kurvenpunkte für äquidistantes $\Delta w = 0.1$.
Die Kurve entspricht dem folgenden NC-Programmausschnitt:

N10 SplineDef(0002)
N20 G1 x1 y0
N20 G06 x(1,0,1) y(0,2) DN(1,0,-1) PL0.8

Relevante NC-Funktionen

CPI(1)	Einschalten der Kurvenparameterinterpolation
CPI(0)	Zurückschalten auf Bahnlängeninterpolation

Tab. 6-5: Relevante NC-Funktionen

7 Sicherheit / Überwachung

7.1 Momenten-Reduktion

7.1.1 Beschreibung

Funktion

Die Momenten-Reduktion erlaubt eine temporäre Beeinflussung des eingestellten Drehmoment-Grenzwerts einer Achse (S-0-0092: Drehmoment-Grenzwert bipolar).

Merkmale:

- Die Funktion ist per achsbezogenem Interface-Signal qAx_TrqLim "Momenten-Reduktion" ein-/ausschaltbar.
- Rückmeldung des Funktionsstatus erfolgt über das achsbezogene Interface-Signal iAx_TrqLim "Momenten-Reduktion aktiv".
- Das reduzierte Drehmoment, das mit dem Setzen des Interface-Signals "Momenten-Reduktion" an den Antrieb übertragen wird, wird über den MP 1003 00010 eingestellt.

Der MP-Wert lässt sich per NC-Funktion "RedTorque" programmgesteuert verändern. Ein per RedTorque veränderter Drehmoment-Grenzwert wird mit der nächsten positiven Flanke des achsbezogenen Interface-Signals qAx_TrqLim aktiv.

Mit Rücksetzen des Interface-Signals wird dem Antrieb wieder der zum Zeitpunkt des Steuerungshochlaufs aktuelle Momentengrenzwert (S-0-0092) übertragen.

Restriktion

- Ein-/Ausschalten der Momentenreduktion ist nur bei Achsstillstand erlaubt.
- S-0-0092 muss im Antrieb als % vom Nennmoment interpretiert werden (in S-0-0086 ist die Wichtungsart "prozentuale Wichtung" einzustellen).

Relevante NC-Funktionen

RedTorque	Momenten-Reduktion

Tab. 7-1: Relevante NC-Funktionen

Relevante IF-Signale

qAx_TrqLim	"Momentenreduktion" (Achsinterface) L/H Flanke: Momenten-Reduktion einschalten. H/L-Flanke: Momenten-Reduktion ausschalten. Beim Ausschalten wird im Antrieb wieder derjenige Drehmoment-Grenzwert gesetzt, der nach dem letzten Sercos Phasenhochlauf in S-0-0092 eingestellt war.
iAx_TrqLim	"Moment reduziert" (Achsinterface) ist aktiv.

Tab. 7-2: Relevante IF-Signale

Relevante Maschinenparameter

1003 00010	Voreinstellung für reduziertes Maximalmoment der Achse.

Tab. 7-3: Relevante Maschinenparameter

7.1.2 Handlungsanweisung: Momenten-Reduktion aktivieren/deaktivieren

Applizieren: Engineering / Configuration: Reduziertes Drehmoment bei aktiver Momentenreduktion

Maschinenparameter anpassen

- **RedMaxTorq** "Reduziertes Drehmoment bei aktiver Momentenreduktion" (1003 00010)

	Documentation	Instruction
Instruction:		Editieren von Maschinenparametern
Documentation:	MTX Maschinenparameter	Momenten-Reduktion



S-0-0092 "Drehmoment/Kraft-Grenzwert bipolar" muss im Antrieb als % vom Nennmoment interpretiert werden (in **S-0-0086** ist die Wichtungsart "prozentuale Wichtung" einzustellen).

Aktivieren/Deaktivieren: IW-Operation / NC-Programmierung: Reduziertes Drehmoment bei aktiver Momentenreduktion

Mit der NC-Funktion **RedTorque(<Achsenname><Wert>)** kann der Wert des reduzierten Drehmoments verändert werden.

	Documentation	Instruction
Documentation:	MTX Programmierhandbuch	Momenten-Reduktion

IW-Engineering / IndraLogic: Momenten-Reduktion aktivieren/deaktivieren

- **qAx_TrqLim = 1:**
Überblendet den Sercos Parameter S-0-0092 "Drehmoment/Kraft-Grenzwert bipolar" mit dem Wert aus dem Parameter **RedMaxTorq** "Reduziertes Drehmoment bei aktiver Momenten-Reduktion" (1003 00010) oder der NC-Funktion **RedTorque**.
- **qAx_TrqLim = 0:**
Der originale Momentengrenzwert aus dem Sercos Parameter **S-0-0092** "Drehmoment/Kraft-Grenzwert bipolar" wird zum Antrieb übertragen.

	Documentation	Instruction
Instruction:		Editieren von SPS-Signalen
Documentation:	MTX SPS-Interface	Momenten-Reduktion

7.2 Reduzierter Eilgang

7.2.1 Beschreibung

Funktion

Beim Einfahren von Teileprogrammen ist es oft nicht erwünscht, dass Eilgangsätze mit maximaler Geschwindigkeit verfahren werden. Eine Reaktion

des Bedieners mittels "Override" kann bei fehlerhafter Programmierung zu spät sein, um einen möglichen Schaden zu vermeiden.

Deshalb reduziert die Steuerung in Abhängigkeit des kanalbezogenen IF-Signals qCh_RedRap "Reduzierter Eilgang" alle Eilgangbewegungen synchroner Achsen im entsprechenden Kanal auf

- einen parametrierbaren Wert (sofern MP 7030 00110 > 0)
- oder -
- auf die zuletzt im Kanal programmierte Vorschubgeschwindigkeit (sofern MP 7030 00110 = 0).



Sie können die Funktion auch zur Eilgangreduktion verwenden, solange noch keine Referenzpunkte angefahren wurden.

Ist die Funktion "Override" im relevanten Kanal aktiviert, wirkt sie auch auf den reduzierten Eilgang. Ein Override von 100% entspricht in diesem Fall der reduzierten Eilganggeschwindigkeit.

Restriktionen

- Wirkt in den Betriebsarten "Handeingabe NC-Satz", "Folgesatz", "Programmsatz", "Einzelsatz" und "Einzelschritt".
- Eine Änderung des IF-Signals qCh_RedRap wirkt nicht auf Programmsätze, die zum Änderungszeitpunkt bereits durch die Satzvorbereitung bearbeitet wurden.

Ist dieses Verhalten nicht gewünscht, müssen Sie die Bearbeitung mit "Vorschub Halt" anhalten, bereits vorbereitete Sätze per "Restweg löschen" verwerfen und die Bearbeitung per NC-Start fortsetzen.

Relevante IF-Signale

qCh_RedRap	"Reduzierter Eilgang" (Kanalinterface)

Tab. 7-4: Relevante IF-Signale

Relevante Maschinenparameter

7030 00110	Wert für reduzierte Eilganggeschwindigkeit

Tab. 7-5: Relevante Maschinenparameter

7.2.2 Handlungsanweisung: Reduzierter Eilgang

Die Funktion reduziert die maximale Geschwindigkeit der Eilgangbewegung. (wird z.B. für Testzwecke genutzt)

Applizieren: IW-Engineering / Configuration: Reduzierter Eilgang konfigurieren

Dem Parameter **RedChVel** "Reduzierte Eilganggeschwindigkeit" (7030 00110) die gewünschte reduzierte Eilganggeschwindigkeit zuweisen.

	Documentation	Instruction
Instruction:		Editieren von Maschinenparametern
Documentation:	MTX Maschinenparameter	Reduzierter Eilgang

IW-Operation / Maschine: Systemgrundstellung auslösen

Zur Übernahme der geänderten Maschinenparameter ist eine Systemgrundstellung notwendig.

Aktivieren:

HINWEIS**Schäden an der Maschine oder am Werkstück möglich!**

Während der Bearbeitung darf keine Systemgrundstellung ausgelöst werden.

IW-Engineering / IndraLogic: Reduzierter Eilgang aktivierenAchsinterface-Signal **qCh_RedRap Reduzierter Eilgang** setzen:

- **qCh_RedRap = 1:**
zum Aktivieren der Funktion "Reduzierter Eilgang".



Wirkt nicht auf Spindeldrehzahlen oder auf den aktiven Bahnvor-schub.

Eine Änderung des Interface-Signals **qCh_RedRap** wirkt nicht auf Programmsätze, die zum Änderungszeitpunkt bereits durch die Satzvorbereitung bearbeitet wurden.

Ist die Funktion "Override" im relevanten Kanal aktiviert, wirkt sie auch auf den reduzierten Eilgang. Ein Override von 100% entspricht in diesem Fall der reduzierten Eilgangsgeschwindigkeit.

	Documentation	Instruction
Instruction:		Editieren von SPS-Signalen
Documentation:	MTX Parameterbeschrei-bung	Reduzierter Eilgang

Deaktivieren: **IW-Engineering / IndraLogic: Reduzierter Eilgang aktivieren/deaktivieren**Achsinterface-Signal **qCh_RedRap Reduzierter Eilgang** löschen:

- **qCh_RedRap = 0:**
zum Deaktivieren der Funktion "Reduzierter Eilgang".

Eine Änderung des Interface-Signals **qCh_RedRap** wirkt nicht auf Programmsätze, die zum Änderungszeitpunkt bereits durch die Satzvorbereitung bearbeitet wurden.

	Documentation	Instruction
Instruction:		Editieren von SPS-Signalen
Documentation:	MTX Parameterbeschrei-bung	Reduzierter Eilgang

7.3 Test-Vorschub, Test-Eilgang

7.3.1 Beschreibung

Funktion

Überblendet für die Dauer der jeweils aktivierten Funktion die zuvor im Kanal wirksame Vorschub- bzw. Eilgangsgeschwindigkeit, sofern G8 oder G9 aktiv ist.

Hierbei gilt:

- Per Funktion "Test-Vorschub" lässt sich der wirksame Vorschub temporär auf einen definierten Wert erhöhen oder erniedrigen.

- Per Funktion "Test-Eilgang" lässt sich die wirksame Eilgangsgeschwindigkeit temporär auf einen definierten Wert reduzieren.

HINWEIS

Unzulässig hoher Vorschub während Werkzeugeingriff möglich!

Durch Aktivieren des Test-Vorschubs während der Bearbeitung kann der technologisch maximal erlaubte Vorschub überschritten werden.

Schäden am Werkstück, Werkzeug oder an der Maschine können die Folge sein.

Verwenden Sie die Funktion Test-Vorschub deshalb nur zu Testzwecken und nicht zur Teilebearbeitung!

Merkmale

- Die Funktionen sind sowohl per kanalspezifischen Interface-Signalen, als auch per Benutzeroberfläche beeinflussbar.
Die Steuerung meldet den jeweils aktiven Zustand am kanalspezifischen Interface zurück.
- Die während aktivem Test-Vorschub bzw. Test-Eilgang wirksamen Geschwindigkeitswerte werden pro Kanal jeweils in einem Systemdatum abgelegt (in mm/min).
- Der aktuelle Vorschub-Override im Kanal bleibt auch bei Test-Vorschub und Test-Eilgang wirksam.
- Wirkt unter G93 (Zeitprogrammierung), G94 (direkte Vorschubprogrammierung) und G95 (Vorschub/Umdrehung).
- Beide Funktionen wirken auf die Sollgeschwindigkeit unmittelbar nach der Aktivierung. Hierbei berücksichtigt die Steuerung für erforderliche Geschwindigkeitsänderungen die aktuell eingestellte Beschleunigung.

Restriktionen

- G8 oder G9 muss aktiv sein.
- Wirkt nicht im Zusammenhang mit der Funktionalität "Geschwindigkeitsprofile" (Rampenfunktionen).
- Änderungen der Geschwindigkeitswerte in den Systemdaten werden während laufendem Teileprogramm erst im nächsten vorzubereitenden NC-Satz wirksam.
- Ist der per Systemdatum vorgegebene Test-Eilgang größer als der steuerungsintern für die Bahn berechnete Eilgangswert, wirkt der steuerungsintern für die Bahn berechnete Eilgangswert.

Relevante Systemdaten

SysTestMode[<Kanalnr>].Enable	
	Ein im Systemdatum SysTestMode[<Kanalnr>].TestFeedrate hinterlegter und gegenüber dem programmierten Vorschub höherer Geschwindigkeitswert kann nur erreicht werden, wenn SysTestMode[<Kanalnr>].Enable den Wert 1 hat. Während der Bearbeitung sollte SysTestMode[<Kanalnr>].Enable hingegen auf den Wert 0 gesetzt sein; ansonsten kann es zu einer im Bearbeitungsbetrieb unerwünschten Absenkung der Bahnbeschleunigung kommen.

SysTestMode[<KanalNr>].TestFeedrate	
	Geschwindigkeit in mm/min für Test-Vorschub in Kanal <KanalNr>
SysTestMode[<KanalNr>].TestRapid	
	Geschwindigkeit in mm/min für Test-Eilgang in Kanal <KanalNr>

Tab. 7-6: Relevante Systemdaten

Relevante IF-Signale

qCh_TestFeed	"Test-Vorschub" (Kanalinterface) Schaltet die Funktion "Test-Vorschub" ein bzw. aus.
qCh_TestRap	"Test-Eilgang" (Kanalinterface) Schaltet die Funktion "Test-Eilgang" ein bzw. aus.
iCh_TestFeed	"Test-Vorschub aktiv" (Kanalinterface) Spiegelt den Status der Funktion "Test-Vorschub".
iCh_TestRap	"Test-Eilgang aktiv" (Kanalinterface) Spiegelt den Status der Funktion "Test-Eilgang".

Tab. 7-7: Relevante IF-Signale

7.3.2 Handlungsanweisung: Test-Vorschub (DryRun)

Aktivieren/deaktivieren von der Funktion Test-Vorschub (DryRun). Die Funktion Test-Vorschub überblendet die zuvor im Kanal wirksame Vorschubgeschwindigkeit.

Aktivieren: **IW-Operation / NC-Programmierung: Einem Systemdatum einen Wert zuweisen**

SD.SysTestMode[<Kanalnummer>].TestFeedrate = <Geschwindigkeit in mm/min>

Test-Vorschubgeschwindigkeit im Systemdatum ablegen.

	Documentation	
Documentation:	MTX Programmierhandbuch	Test-Vorschub

IW-Operation / HMI / M-Panels: M-Key belegen

		Instruction
Instruction:		M-Key belegen

IW-Operation / Programm: Test-Vorschub aktivieren

Aktivieren per Bedienoberfläche:

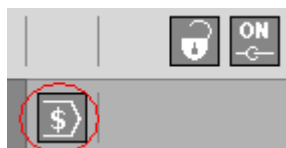
- Den Linken M-Key **"DryRun Optionen"** (L7) auswählen



- Mit dem Linken M-Key **"Test-Vorschub"** (L4) aktivieren



- Test-Vorschub ist aktiv, wenn das „\$“-Zeichen im Statusfeld erscheint



oder per SPS:

qCh_TestFeed = 1:

aktivieren des Test-Vorschubs mit dem kanalspezifischen Geschwindigkeitswert aus der CPL-Variablen **TEST_FEEDRATE**.



G8 oder G9 muss aktiv sein.

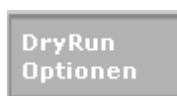
Das Interface-Signal **iCh_TestFeed** "Test-Vorschub aktiv" (Kanal-interface) spiegelt den Status der Funktion "Test-Vorschub" wieder.

	Documentation	
Documentation:	IndraWorks HMI	Test-Vorschub

Deaktivieren: IW-Operation / Programm: Test-Vorschub deaktivieren

Deaktivieren per Bedienoberfläche:

- Den Linken M-Key "DryRun Optionen" (L7) auswählen



- Mit dem Linken M-Key "Test-Vorschub" (L4) abwählen



oder per SPS:

qCh_TestFeed = 0:

deaktivieren des Test-Vorschubs

	Documentation	
Documentation:	IndraWorks HMI	Test-Vorschub

7.3.3 Handlungsanweisung: Test-Eilgang (DryRun)

Die Funktion "Test-Eilgang" reduziert die im Kanal konfigurierte Eilganggeschwindigkeit bei G0 zum Einfahren eines NC-Programms mit der (niedrigeren) Test-Eilganggeschwindigkeit.

Aktivieren: IW-Operation / NC-Programmierung: Einem Systemdatum einen Wert zuweisen

`SD.SysTetsSD.SysTestMode[<Kanalnummer>].TestRapid = <Geschwindigkeit in mm/min>`

Test-Eilganggeschwindigkeit im Systemdatum ablegen.

	Documentation	
Documentation:	MTX Programmierhandbuch	Test-Vorschub

IW-Operation / HMI / M-Panels: M-Key belegen

		Instruction
Instruction:		M-Key belegen

IW-Operation / Programm: Test-Eilgang aktivieren

Aktivieren per Bedienoberfläche:

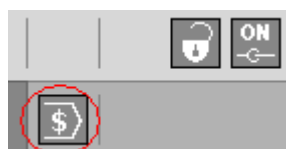
- Den Linken M-Key "**DryRun Optionen**" (L7) auswählen



- Mit dem Linken M-Key "**Test-Eilgang**" (L4) aktivieren



- Test-Vorschub ist aktiv, wenn das „\$“-Zeichen im Statusfeld erscheint



oder per SPS:

qCh_TestRap = 1:

Aktivieren des Test-Eilgangs mit dem kanalspezifischen Geschwindigkeitswert aus der CPL-Variablen **TEST_RAPID**.



G8 oder G9 muss aktiv sein.

Das Interface-Signal **iCh_TestRap** "Test-Vorschub aktiv" (Kanal-interface) spiegelt den Status der Funktion "Test-Vorschub" wieder.

	Documentation	
Documentation:	IndraWorks HMI	Test-Vorschub

Deaktivieren: IW-Operation / Programm: Test-Eilgang deaktivieren

Deaktivieren per Bedienoberfläche:

- Den Linken M-Key "**DryRun Optionen**" (L7) auswählen



- Mit dem Linken M-Key "**Test-Eilgang**" (L4) abwählen



oder per SPS:

qCh_TestRap = 0:

deaktivieren des Test-Eilgangs

	Documentation	
Documentation:	IndraWorks HMI	Test-Vorschub

7.4 1-dimensionale Achskollisionsüberwachung

7.4.1 Beschreibung

Funktion

Überwacht die Differenz oder Summe zweier Achspositionen

Pos(1) und Pos(2):

Unterer Grenzwert $\leq$ Pos(1) $\pm$ Pos(2) $\leq$ Oberer Grenzwert

Beim Versuch, den Überwachungsbereich zu verlassen, wird ein Fehler gemeldet.

Hiermit lassen sich beispielsweise 2 beliebige parallel verlaufende lineare Systemachsen überwachen auf

- Kollision
- oder -
- zu weites Auseinanderlaufen.

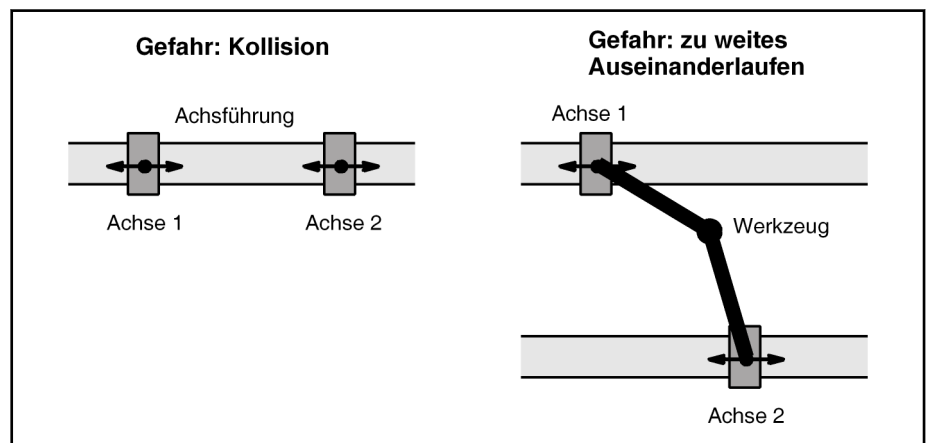


Abb. 7-1: Anwendungsbeispiele

Merkmale

- 10 Verbände zu jeweils 2 Achsen können gleichzeitig überwacht werden. Eine Achse darf hierbei mehreren Verbänden angehören.
- Die Achsen eines Verbandes dürfen unterschiedlichen Kanälen zugeordnet sein.
- Die Überwachung berücksichtigt auch erforderliche Bremswege. Dadurch werden vorgegebene Grenzwerte beim Stillsetzen der relevanten Achsen nicht überschritten.

Restriktionen

- Maximal 10 Verbände systemweit möglich.
- Die Achskollisionsüberwachung ist für einen Verband nur dann wirksam, wenn die beiden beteiligten Achsen referenziert sind.
- Moduloachsen sind nicht erlaubt.
- Zu große Achsgeschwindigkeiten (v_{Achse1} , v_{Achse2}) und/oder Interpolator-Zykluszeiten ($tlpo$) können auch dann zum Ansprechen der Achskollisionsüberwachung führen, wenn die programmierten, rein positionsbezogenen Daten nicht zu einer Kollision führen würden.

Ursache hierfür ist die Unterschreitung eines intern gebildeten Sicherheitsabstands:

$$d_{\min} = 4 * |\max(v_{\text{Achse1}}, v_{\text{Achse2}})| * t_{\text{ipo}}$$

Relevante Maschinenparameter (MP)

8008 00001	Überwachungsverband aktiv Schaltet den entsprechenden Verband für die Überwachung frei. Die Überwachung wird aktiv, sobald die beiden beteiligten Achsen referenziert sind.
8008 00002	Systemachsnummer der Überwachungsachse 1 Physikalischer Achsindex der ersten Achse des überwachten Verbandes.
8008 00003	Systemachsnummer der Überwachungsachse 2 Physikalischer Achsindex der zweiten Achse des überwachten Verbandes. Überwachungsachse 1 und 2 dürfen sich in unterschiedlichen Kanälen befinden.
8008 00004	Legt fest, ob die Differenz oder die Summe der beiden Achspositionen überwacht wird: 0: Differenz wird überwacht, also: $\text{unterer Grenzwert} \leq \text{Pos}(1) - \text{Pos}(2) \leq \text{oberer Grenzwert}$ 1: Es wird die Summe überwacht, also: $\text{unterer Grenzwert} \leq \text{Pos}(1) + \text{Pos}(2) \leq \text{oberer Grenzwert}$
8008 00005	Überwachung des oberen Grenzwerts aktivieren. 0: oberer Grenzwert (MP 8008 00007) wird nicht überwacht, also: $\text{unterer Grenzwert} \leq \text{Pos}(1) \pm \text{Pos}(2)$ 1: oberer Grenzwert (MP 8008 00007) wird überwacht, also: $\text{unterer Grenzwert} \leq \text{Pos}(1) \pm \text{Pos}(2) \leq \text{oberer Grenzwert}$ Ob der untere Grenzwert überwacht wird, wird in MP 8008 00006 festgelegt.
8008 00006	Überwachung des unteren Grenzwerts aktivieren. 0: unterer Grenzwert (MP 8008 00008) wird nicht überwacht, also: $\text{Pos}(1) \pm \text{Pos}(2) \leq \text{oberer Grenzwert}$ 1: unterer Grenzwert (MP 8008 00008) wird überwacht, also: $\text{unterer Grenzwert} \leq \text{Pos}(1) \pm \text{Pos}(2) \leq \text{oberer Grenzwert}$ Ob der obere Grenzwert überwacht wird, wird in MP 8008 00005 festgelegt.
8008 00007	oberer Grenzwert Eingetragener Zahlenwert ist nur dann wirksam, wenn unter MP 8008 00005 der Wert 1 eingetragen ist.
8008 00008	unterer Grenzwert Eingetragener Zahlenwert ist nur dann wirksam, wenn unter MP 8008 00006 der Wert 1 eingetragen ist.

Tab. 7-8: Relevante Maschinenparameter (MP)

7.4.2 Handlungsanweisung: 1-dimensionale Achskollisionsüberwachung

Die 1-dimensionale Achskollisionsüberwachung überwacht 2 beliebige lineare Systemachsen auf Kollision oder zu weites Auseinanderlaufen. Die Systemachsen bewegen sich auf derselben Bahn hintereinander oder auf parallel liegenden Bahnen.

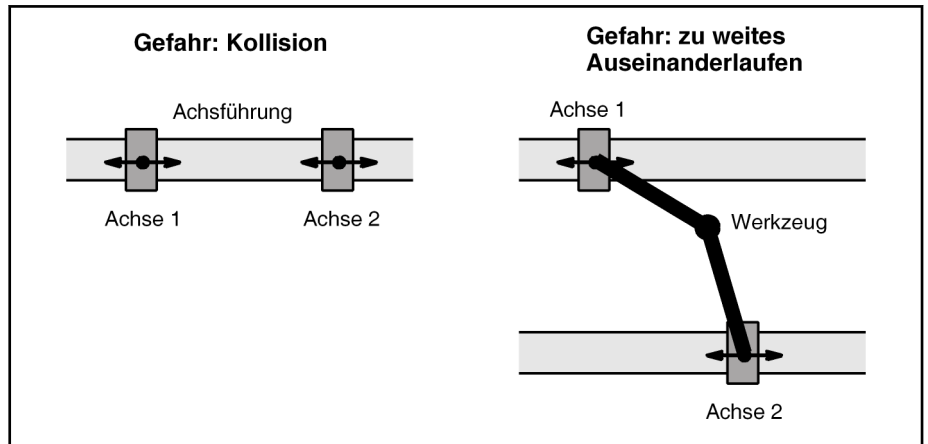


Abb. 7-2: Gefahr der Kollision bzw. des Auseinanderlaufens

Applizieren:

HINWEIS

Gefahr auf Kollision bzw. unzulässig weites Auseinanderlaufen! Maschinenschäden möglich.

Stellen Sie sicher, dass ein rechtzeitiges Anhalten der Achsen jederzeit und unproblematisch möglich ist!



Maximal 10 Verbände systemweit möglich.
Moduloachsen sind nicht erlaubt.

Zu große Achsgeschwindigkeiten $[v(\text{Achse1}), v(\text{Achse2})]$ und/oder Interpolator-Zykluszeiten ($tlpo$) können auch dann zum Ansprechen der Achskollisionsüberwachung führen, wenn die programmierten, rein positionsbezogenen Daten nicht zu einer Kollision führen würden. Ursache hierfür ist die Unterschreitung eines intern gebildeten Sicherheitsabstandes: $d_{min} = 4 * \max[v(\text{Achse1}), v(\text{Achse2})] * tlpo$

IW-Engineering / Configuration: Achskollisionsüberwachung konfigurieren

Führen Sie die nachfolgend beschriebene Parametrierung für jeden einzelnen Überwachungsverband durch:

- Im Parameter **MonAx1** "Überwachungsachse 1" (8008 00002) die Systemachsnummer für die zu überwachende Achse 1 eintragen.
Im Parameter **MonAx2** "Überwachungsachse 2" (8008 00003) die Systemachsnummer für die zu überwachende Achse 2 eintragen.
Geben Sie die Ausrichtung der Koordinatensysteme beider beteiligten Achsen im Parameter **Dir**: "Verfahrrichtung" (MP 8008 00004) an.
- Stellen Sie sicher, dass die relevanten Achsen referenziert sind.
- Fahren Sie die 2 gewünschten Achsen vorsichtig bis zur gewünschten "Kollisionsgrenze". Die Kollisionsgrenze entspricht hierbei dem kritischen Abstand beider Achsen im Gefahrenfall.

- Notieren der Achsposition beider Achsen:
Pos1: Position der Überwachungsachse 1
Pos2: Position der Überwachungsachse 2
Sie benötigen beide Werte im weiteren Verlauf!
- Überwachungstyp festlegen; dabei entspricht log. "0" = Nein und log. "1" = Ja.
ActUpLim "Überwache obere Schranke" (8008 00005)
ActLowLim "Überwache untere Schranke" (8008 00006)

Überwachungstyp	erforderliche Einstellung von MP	
	ActUpLim	ActLowLim
keine Überwachung	0	0
Überwachung auf Kollision, sofern Achse 1 der Achse 2 vorausläuft *)	0	1
Überwachung auf Kollision, sofern Achse 2 der Achse 1 vorausläuft *)	1	0
Überwachung auf Auseinanderlaufen	1	1

*) bezogen auf pos. Verfahrrichtung von Achse 2 bei gleichläufiger Verfahrrichtung beider Achsen.

Tab. 7-9: Überwachungstyp

- Grenzwerte festlegen; Verwenden Sie dazu die oben (Punkt 3.) notierten Werte für Pos1 und Pos2:
PosUpLim "Oberer Grenzwert" (8008 00007)
PosLowLim "Unterer Grenzwert" (8008 00008)

Gefahr	bei Einstellung der Parameter			Berechnung der Werte für Parameter	
	Dir	ActUpLim	ActLowLim	PosUpLim	PosLowLim
Kollision	0 (gleich)	0	1	-	Pos1 - Pos2
		1	0	Pos1 - Pos2	-
Kollision	1 (gegenläufig)	0	1	-	Pos1 + Pos2
		1	0	Pos1 + Pos2	-
Auseinander-laufen	0 (gleich)	0	1	Pos1 - Pos2	-
		1	0	-	Pos1 - Pos2
Auseinander-laufen	1 (gegenläufig)	0	1	Pos1 + Pos2	-
		1	0	-	Pos1 + Pos2

	Documentation	Instruction
Instruction:		Editieren von Maschinenparameter
Documentation:	MTX Maschinenparameter	Achskollisionsüberwachung applizieren

Aktivieren: **Voraussetzung:**
Die Funktion ist korrekt appliziert.



Die Überwachung eines Verbandes wird erst aktiv, wenn beide beteiligten Achsen referenziert sind.

HINWEIS

Schäden an der Maschine oder am Werkstück möglich!

Während der Bearbeitung darf keine Systemgrundstellung ausgelöst werden!

IW-Engineering / Configuration: Achskollisionsüberwachung aktivieren

Der Parameter **EnablCollMon** "Aktiviere Eindimensionale Kollisionsüberwachung" (8008 00001) ist wie folgt einzustellen:

- **EnablCollMon = Ja:**

Aktiviert die Kollisionsüberwachung

	Documentation	Instruction
Instruction:		Editieren von Maschinenparameter
Documentation:	MTX Maschinenparameter	Achskollisionsüberwachung aktivieren

IW-Operation / Maschine: Systemgrundstellung auslösen

Zur Übernahme der geänderten Maschinenparameter ist eine Systemgrundstellung oder ein Steuerungshochlauf notwendig.

Wird ein aktiver Überwachungsverband bereits unmittelbar nach Systemgrundstellung nicht eingehalten, erfolgt eine Warnung. Bewegungen in Richtung auf den zulässigen Bereich sind danach erlaubt. Bewegungen in Richtung vom zulässigen Bereich weg sind verboten, und führen zu einer Fehlermeldung.



Moduloachsen sind nicht erlaubt.

Zu große Achsgeschwindigkeiten $[v(\text{Achse1}), v(\text{Achse2})]$ und/oder Interpolator-Zykluszeiten ($tlpo$) können auch dann zum Ansprechen der Achskollisionsüberwachung führen, wenn die programmierten, rein positionsbezogenen Daten nicht zu einer Kollision führen würden. Ursache hierfür ist die Unterschreitung eines intern gebildeten Sicherheitsabstandes:

$$dmin = 4 * |\max[v(\text{Achse1}), v(\text{Achse2})]| * tlpo$$

Deaktivieren:

HINWEIS

Schäden an der Maschine oder am Werkstück möglich!

Während der Bearbeitung darf keine Systemgrundstellung ausgelöst werden!

IW-Engineering / Configuration: Achskollisionsüberwachung deaktivieren

Der Parameter **EnablCollMon** "Aktiviere Eindimensionale Kollisionsüberwachung" (8008 00001) ist wie folgt einzustellen:

- **EnablCollMon = Nein:**

Deaktiviert die Kollisionsüberwachung

	Documentation	Instruction
Instruction:		Editieren von Maschinenparameter
Documentation:	MTX Maschinenparameter	Achskollisionsüberwachung aktivieren

IW-Operation / Maschine: Systemgrundstellung auslösen

Zur Übernahme der geänderten Maschinenparameter ist eine Systemgrundstellung oder ein Steuerungshochlauf notwendig.

7.5 Achspositionsüberwachung

7.5.1 Beschreibung

Funktion

Meldet per Interface-Signal, in welchem Bereich sich eine Achse bezogen auf eine parametrierbare Position ($\Rightarrow$ MP 2010 00110) befindet. Die Position bezieht sich hierbei stets auf das Achskoordinatensystem (ACS).

Eventuell aktive LSEC-Korrekturen berücksichtigt die Steuerung für die aktuelle Achsposition automatisch.

Für die Signalerzeugung gilt:

- aktuelle Achsposition $\geq$ parametrisierte Position: Signal gesetzt
- aktuelle Achsposition $<$ parametrisierte Position: Signal nicht gesetzt

Die Funktion eignet sich zum SPS-seitigen Ansteuern anderer Funktionalitäten.

- Merkmale:**
- Pro Achse lassen sich bis zu 8 Positionen mit jeweils einem korrespondierenden Interface-Signal definieren.
 - Bei Rundachsen kann sich die Überwachungsposition auf den kompletten Verfahrbereich oder auf den Bereich zwischen 0 und 360 Grad beziehen.
 - Ob die Überwachung nur bei referenzierter oder bei nicht referenzierter Achse erfolgen soll, ist einstellbar.

Restriktion

- Überwachungspositionen beziehen sich auf das Achskoordinatensystem
- Maximal 64 Überwachungspositionen systemweit

Relevante IF-Signale

iAx_PosSwitch1 bis iAx_PosSwitch8	
	"Positionsschaltpunkt 1 ... 8" (Achsinterface). High-Signal: akt. Achsposition $\geq$ parametr. Position Low-Signal: akt. Achsposition $<$ parametr. Position

Tab. 7-10: Relevante IF-Signale

Relevante Maschinenparameter (MP)

2010 00100	überwachte Achse
2010 00110	Position des Punktes
2010 00120	Interface-Signal des Punktes
2010 00130	zyklische Überwachung bei Rundachsen

Tab. 7-11: Relevante Maschinenparameter (MP)

7.5.2 Handlungsanweisung: Achspositionsüberwachung

Applizieren: IW-Engineering / Configuration: Achspositionsüberwachung einstellen

Je Point[i] (Wegschaltpunkt[i]) sind folgende Parameter anzupassen:

- **Dr** "Achsnnummer der physikalischen Achse" (2010 00100)
- **Pos** "Position" (2010 00110)
- **MonType** "Art der Überwachung" (2010 00100)
- **MonRotAx** "Überwachung bei Rundachsen" (2010 00130)
- **IfSig** **Interface-Signal des Punktes** (2010 00120), enthält das gewünschte Interface-Signal **iAx_PosSwitch1** bis **iAx_PosSwitch8**.



Überwachungspositionen beziehen sich auf das Achskoordinatensystem (ACS).

Maximal 64 Überwachungspositionen Systemweit möglich.

	Documentation	Instruction
Instruction:		Editieren von Maschinenparametern
Documentation:	MTX Maschinenparameter	Achspositionsüberwachung

Aktivieren: IW-Engineering / IndraLogic: Achspositionsüberwachungssignal benutzen

Je Achse können bis zu 8 Interface-Signale als Positionsschaltpunkt genutzt werden:

iAx_PosSwitch1 bis **iAx_PosSwitch8** "Positionsschaltpunkt 1 ... 8" (Achsinterface)

- **High-Signal:** aktuelle Achsposition $\geq$ parametrisierte Position
- **Low-Signal:** aktuelle Achsposition $<$ parametrisierte Position

	Documentation	Instruction
Instruction:		Editieren von SPS-Signalen
Documentation:	MTX SPS-Interface	Achspositionsüberwachung

7.6 Distanz vor Endpunkt

7.6.1 Beschreibung

Funktion

Ausgabe eines Achs-IF-Signals, sobald die aktuelle Distanz der Sollposition zum programmierten Endpunkt eines NC-Satzes eine definierte Strecke unterschreitet.

Auf diese Weise können z. B. eigene Funktionen durch die SPS schon initialisiert oder vorbereitet werden, bevor die Achse ihren tatsächlichen Endpunkt erreicht. Dadurch verkürzt sich unter Umständen der Gesamtzeitbedarf eines Fertigungsprozesses.

Restriktion

"Distanz vor Endpunkt" ist ohne Funktion für

- Slaveachsen bei aktiver Achskopplung
- Handradbetrieb
- Betriebsart "Referenzpunkt anfahren"

Relevante IF-Signale

iAx_DistCtrl	"Achse vor Endpunkt" (Achsinterface)

Tab. 7-12: Relevante IF-Signale

Relevante Maschinenparameter (MP)

1015 00010	Distanz vor Endpunkt.

Tab. 7-13: Relevante Maschinenparameter (MP)

7.6.2 Handlungsanweisung: Distanz vor Endpunkt

Dient zur Ausgabe eines Achsinterface-Signals, sobald die aktuelle Distanz der Sollposition zum programmierten Endpunkt eines NC-Satzes eine definierte Strecke unterschreitet.

Applizieren:

HINWEIS

Schäden an der Maschine oder am Werkstück möglich!

Während der Bearbeitung darf kein Steuerungs-Reset ausgelöst werden.

IW-Engineering / Configuration: Parameter editieren

- **EnablAxDistCtrl** "Aktiviere Distanzüberwachung" (1015 00010)
EnablAxDistCtrl = Ja: Aktivieren von "Distanz vor Endpunkt"
- Im Parameter **DistEndPoint** "Abstand zum programmierten Endpunkt" (1015 00010) die gewünschte Strecke eintragen, wo bei Unterschreitung dieser Strecke das Achsinterface-Signal **iAx_DistCtrl** "Achse vor Endpunkt" gesetzt werden soll.

	Documentation	Instruction
Instruction:		Editieren von Maschinenparametern
Documentation:	MTX Maschinenparameter	Distanz vor Endpunkt

IW-Operation / Diagnose: Steuerungshochlauf auslösen

Zur Aktivierung der Funktion ist ein Steuerungshochlauf erforderlich.

IW-Engineering / IndraLogic: Signal Distanz vor Endpunkt benutzen

Das Achsinterface-Signal **iAx_DistCtrl** "Achse vor Endpunkt" wird bei Unterschreiten der im Parameter **DistEndPoint** "Abstand zum programmierten Endpunkt" (1015 00010) definierten Strecke gesetzt.



"Distanz vor Endpunkt" ist ohne Funktion für:

- Slaveachsen bei aktiver Achskopplung
- Handradbetrieb
- Betriebsart "Referenzpunkt anfahren"

	Documentation	Instruction
Instruction:		Editieren von SPS-Signalen
Documentation:	MTX Maschinenparameter	Distanz vor Endpunkt

Deaktivieren:

HINWEIS

Schäden an der Maschine oder am Werkstück möglich!

Während der Bearbeitung darf kein Steuerungs-Reset ausgelöst werden.

IW-Engineering / Configuration: Parameter editieren

- **EnablAxDistCtrl** "Aktiviere Distanzüberwachung" (1015 00010)
EnablAxDistCtrl = Nein: Deaktivieren von "Distanz vor Endpunkt"

	Documentation	Instruction
Instruction:		Editieren von Maschinenparametern
Documentation:	MTX Maschinenparameter	Distanz vor Endpunkt

7.7 Rückzugsvektor als Systemdatum

7.7.1 Beschreibung

Funktion

Tritt während der Bearbeitung ein Programmabbruch auf, ist es meist wünschenswert, das Werkzeug senkrecht zur bearbeiteten Werkstückoberfläche vom Werkstück abzuheben. Dies ist z. B. nach Stromausfällen von Bedeutung. Ist die Stromversorgung wiederhergestellt, möchte der Kunde das Werkzeug von der Werkstückoberfläche abheben, ohne diese zu beschädigen.

Um dies zu erleichtern, ist es möglich, die Speicherung folgender Daten im Interpolatortakt durch eine NC-Funktion zu aktivieren:

- Speicherung des Orientierungsvektors in das Systemdatum SD.RevVec.Orientation[1..3]
- Speicherung des Binormalenvektors in das Systemdatum SD.RecVec.Binormal[1..3]

Bei aktiver Radiuskorrektur ist das der normierte Radiuskorrekturvektor.

Bei nicht-aktiver Radiuskorrektur ist es der Einheitsvektor senkrecht zu Bewegungsrichtung und Orientierungsvektor.

Eine Ausnahme sind hier der G41/G42-Korrekturaufbausatz sowie der G40-Korrekturabbausatz. Hier steht der Binormalenvektor senkrecht zur Fräsermittelpunktsbahn, was aber in der Praxis kaum von Bedeutung sein dürfte, da man in diesen Sätzen noch keinen Kontakt zum Werkstück hat.

Das Systemdatum **SD.RevVec** wird nicht per Default mitgeliefert. Wenn der Anwender die hier beschriebene Funktionalität nutzen möchte, muss er es selbst wie folgt anlegen:

```
<Variable Dimension="Channel" Storage="permanent">
  <Name> RevVec </Name>
  <Type> RevVec_t </Type>
</Variable>
```

Wählt der Anwender den falschen Typ, erhält er beim nächsten Hochlauf den leichten Systemfehler 4144.

Wählt der Anwender die falsche Dimension, erhält er beim nächsten Hochlauf den leichten Systemfehler 4145.

Legt der Anwender das Systemdatum gar nicht an, erfolgt beim nächsten Hochlauf kein Hinweis.

Aktiviert der Anwender RRV, ohne dass ein geeignetes Systemdatum vorhanden ist, wird der Fehler 4147 gemeldet.

Aktiviert der Anwender RRV und der Typ des Systemdatums ist falsch, wird der Fehler 4148 gemeldet.

Aktiviert der Anwender RRV und die Dimension des Systemdatums ist falsch, wird der Fehler 4149 gemeldet.

Anstelle von Storage = "permanent" ist auch Storage = "volatile" zulässig. Der Inhalt des Systemdatums geht dann bei Stromausfall verloren.

Bezogen auf die Bewegungsrichtung (per Definition nach vorn) und auf den vom Werkstück weg weisenden Orientierungsvektor (nach oben), weist die Senkrechte zu diesen beiden Richtungen entweder nach links oder nach rechts. Welche dieser beiden Möglichkeiten gewünscht ist, muss bei der Aktivierung der Speicherung festgelegt werden.

Die Aktivierung der Speicherung mit einem nach links weisenden Binormalenvektor erfolgt mit dem NC-Befehl (vgl. Dokumentation MTX Programmierhandbuch):

RecordRevVec(1) oder alternativ **RRV(1)** , **RRV**

Die Aktivierung der Speicherung mit einem nach rechts weisenden Binormalenvektor erfolgt mit dem NC-Befehl (vgl. Dokumentation MTX Programmierhandbuch):

RecordRevVec(2) oder alternativ **RRV(2)**

Sobald der NC-Satz, in dem die Aktivierung erfolgt ist, zur Abarbeitung gelangt, wird das Systemdatum SD.RevVec zunächst auf Null initialisiert, und dann folgt die Abspeicherung des aktuellen Orientierungs- und Binormalenvektors im Interpolatortakt.

Ist eine Radiuskorrektur aktiv, wird die Entscheidung, ob der Binormalenvektor nach links oder rechts weisen soll, nicht durch den Parameter der NC-Funktion **RecordRevVec** getroffen, sondern durch die Radiuskorrektur selbst.

G41, G141: Werkzeug steht links vom Werkstück. Binormalenvektor weist ebenfalls nach links. Beides gilt auch bei negativem Fräserradius.

G42, G142: Werkzeug steht rechts vom Werkstück. Binormalenvektor weist ebenfalls nach rechts. Beides gilt auch bei negativem Fräserradius.

In diesem Fall ist es also egal, ob man den Parameter 1, 2 oder gar keine Klammern programmiert.

Wird eine Radiuskorrektur deaktiviert, während die Speicherung aktiv ist, wird die Entscheidung, ob der Binormalenvektor nach links oder rechts weisen soll, durch den Parameter der aktiven NC-Funktion RecordRevVec getroffen.

Die beiden Vektoren beziehen sich grundsätzlich auf das Basiskoordinatensystem (BCS). Werden sie also benutzt, um das Werkzeug vom Werkstück abzuheben, dürfen dabei weder Koordinatentransformationen noch Eingabehilfen aktiv sein.

Die Speicherung des Systemdatums wird deaktiviert mit dem NC-Befehl:

RecordRevVec(0)

oder alternativ

RecordRevVec(), RRV(0), RRV()

Sobald der NC-Satz, in dem die Deaktivierung erfolgt, zur Abarbeitung gelangt, wird in das Systemdatum SD.RevVec nichts mehr geschrieben. Die zuletzt dort abgelegten Werte bleiben unverändert erhalten.

Bemerkungen

- Der Orientierungsvektor steht nicht zwingend senkrecht zur Bewegungsrichtung.
- Negative Fräserradien werden z. B. benötigt, wenn das CAD-System bereits die Fräsermittelpunktsbahn für einen bestimmten Fräserradius liefert, und der Anwender sich entschließt, mit einem anderen Fräserradius zu arbeiten. In diesem Fall ist die vorhandene Bahn mit folgendem "Fräserradius" zu korrigieren:

$$(\text{Tatsächlicher Fräserradius}) - (\text{CAD-Fräserradius})$$
 Die Rückzugsrichtung muss jedoch immer dieselbe bleiben, egal ob dieser "Fräserradius" > 0 , $= 0$ oder < 0 ist.
- Ist Mirror bezüglich einer oder aller drei Raum-Koordinaten aktiv, wird der Binormalenvektor invertiert.
- Sind G20 und RRV gleichzeitig aktiv, wird das Systemdatum genullt und die Warnung 4146 ausgegeben, da bei aktivem G20 keine Aussage zum Reversevektor möglich ist.

Beispiel: Programmabbruch durch Stromausfall

Programm:

```
001 DCT(4,1,0)=10
N10 RRV(1)
N20 G71 G17 G1 F1000 X-100 Y0 Z0
N30 X-70
N40 X-30 G41 ED1
N50 X0
N60 X40 Y30 ; Power failure during block processing
...
```

Das folgende NC-Programm sei an der markierten Stelle durch einen Stromausfall unterbrochen worden:

Beispiel: Gezielter Rückzug

Nach dem Steuerungshochlauf kann z. B. mit folgendem NC-Programm oder NC-Zyklus um 17.0 mm seitlich von der Werkstückoberfläche zurückgezogen werden:

Programm:

```
10 A = SD.RevVec.Binormal[1] :REM = -0.6
20 B = SD.RevVec.Binormal[2] :REM = 0.8
30 C = SD.RevVec.Binormal[3] :REM = 0.0
40 W = 17.0 :REM Retract path
```

```

N50 G91 G17 G1 F100
N60 X[W*A] Y[W*B] Z[W*C]

; It is assumed that after control startup
; no milling path correction, no coordinate transformation
; and no input help is active. The decision about the
; retract distance, that it the distance that may be retracted
; safely is the user`s responsibility.

```

Relevante NC-Funktionen

RecordRevVec, RRV, G16..G20, G40, G41, G42, G141, G142

7.7.2 Handlungsanweisung: Rückzugsvektor als Systemdatum

Aktivieren: Aktivierung der Speicherung für Binormalenvektor nach links:

RRV(1)

Aktivierung der Speicherung für Binormalenvektor nach rechts:

RRV(2)

Deaktivieren: **RRV(0)**

Der Anwender soll in die Lage versetzt werden nach Unterbrechung eines NC-Programms, z. B. durch Stromausfall, das Werkzeug vom Werkstück zurückziehen, ohne dieses zu beschädigen.

1. Fügen Sie an der Stelle im NC-Programm, ab der der Orientierungs- und Binormalenvektor kontinuierlich aufgezeichnet werden soll, den NC-Befehl RRV(1) bzw. RRV(2) ein.
2. Benötigen Sie ab einer bestimmten Stelle im NC-Programm die Aufzeichnung des Orientierungs- und des Binormalenvektors nicht mehr, fügen Sie an dieser Stelle den NC-Befehl kurz RRV(0) ein.
3. Programmieren Sie einen Zyklus zum Zurückziehen des Werkzeugs bei NC-Programm-Unterbrechung

Beispiel: Falls die Unterbrechung des NC-Programms bei aktiver Fräserbahnkorrektur geschah, wird man den Fräser in Richtung des Binormalenvektors vom Werkstück zurückziehen wollen. In diesem Fall könnte dieser Zyklus wie folgt aussehen:

Programm:

```

20 A = SD.RevVec.Binormal[1]
21 B = SD.RevVec.Binormal[2]
22 C = SD.RevVec.Binormal[3]
23 W = 17.0                                :REM Retract path

G91 G17 G40 G1 F100

X[W*A] Y[W*B] Z[W*C]

```



Die Entscheidung darüber, wie weit in der vorgegebenen Richtung sicher zurückgezogen werden darf (im Beispiel 17.0 mm), müssen Sie anhand der spezifischen Werkstückgeometrie entscheiden.

Orientierungs- und Binormalenvektor beziehen sich stets auf das Basiskoordinatensystem (BCS).

Falls also bei der Aufzeichnung der beiden Vektoren eine Achstransformation aktiv war, muss diese auch bei der Durchführung aktiviert sein.

Falls aber der Aufzeichnung Eingabehilfen, eine Radiuskorrektur oder Koordinatentransformationen, z. B. Placements, aktiv waren, dürfen diese bei der Durchführung des Rückzugs nicht aktiviert werden.

8 Genauigkeit

8.1 Genauigkeitsprogrammierung

8.1.1 Beschreibung

Funktion

In bestimmten Anwendungsfällen kann ein **kontrolliertes** "Verschleifen" von Konturübergängen ausdrücklich erwünscht sein (z. B. bei der Bearbeitung von Freiformflächen).

Die "Genauigkeitsprogrammierung" beschränkt deshalb

1. an Konturübergängen den
 - Konturfehler (ϵ)
 - oder -
 - Bahn-Nachlauf (δ)
 und
2. bei zirkularen Bahnsegmenten (Kreise, Helical, HelicalN) den
 - Radiusfehler (ϵ)

auf einen programmierbaren Wert (in mm bzw. Inch; je nach aktivem G71 bzw. G70).

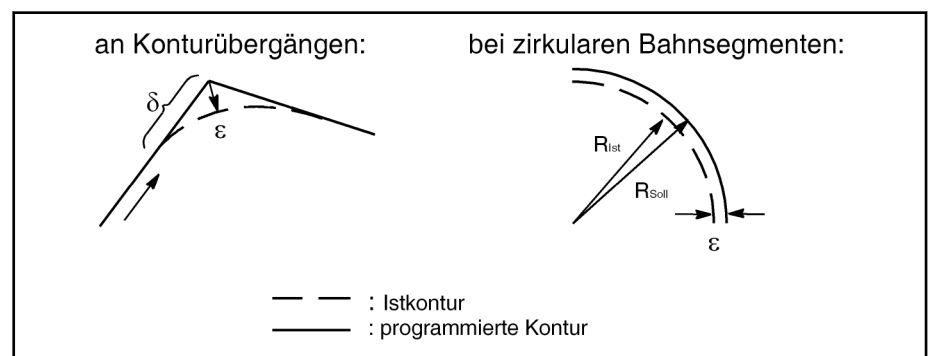


Abb. 8-1: Genauigkeitsprogrammierung

Dazu berechnet die Steuerung für jeden Konturübergang bzw. für jedes zirkulare Bahnsegment die maximale Bahngeschwindigkeit, die zur Einhaltung des jeweils angegebenen Grenzwerts nicht überschritten werden darf.

Restriktionen

- Die Funktion "Genauigkeitsprogrammierung" wirkt für Konturübergänge nur dann, wenn G8 aktiv ist. Bei zirkularen Bahnsegmenten wirkt sie immer.
- Die Funktion "Genauhalt" muss ausgeschaltet sein (G62 ist aktiv).
- Zur Berechnung der max. Bahngeschwindigkeit werden nur die Reglerparameter der ersten Achse im Kanal herangezogen. Alle an der Bahn beteiligten Achsen müssen deshalb bezüglich der Dynamik identisch parametrisiert sein.
- Zur Berechnung der max. Bahngeschwindigkeit geht die Steuerung von einem linearen Zusammenhang zwischen Geschwindigkeit und Nachlauf aus. Der Einfluss von Beschleunigungsvorgängen auf den aktuellen Nachlauf wird nicht berücksichtigt.
- Bei im Antrieb aktivem "Feed Forward" (schleppfehlerfreies Fahren; Vorsteuerung) berücksichtigt die Steuerung nur in Verbindung mit Ser-

vodyn-Antrieben Sercos Parameter P-0-0500 zur Berechnung der zulässigen Bahngeschwindigkeit. Dazu liest sie den Parameterwert während des Sercos Phasenhochlaufs aus dem Antrieb und speichert den Wert ab.

Ist P-0-0500 mit Werten > 99% belegt, wird ein Schleppfehler von 1% des Schleppfehlers ohne Vorsteuerung angenommen.

Bei anderen Antrieben geht die Steuerung für die Berechnung der zulässigen Bahngeschwindigkeit davon aus, dass sich der Nachlauf bei (im Antrieb) aktivierter Vorsteuerungsfunktion um die Hälfte reduziert.

Relevante NC-Funktionen

PrecProg - oder -	
PrecProg(EPS <Konturfehler>) - oder -	
PrecProg(DIST <Bahn-Nachlauf>)	
	Jeweils Genauigkeitsprogrammierung EIN. Wird nur PrecProg programmiert, beschränkt die Steuerung an Konturübergängen bzw. bei zirkularen Bahnsegmenten den Kontur-/Radiusfehler ϵ. Als Grenzwert dient der, in MP 8003 00001 parametrisierte Wert. Wird PrecProg in Verbindung mit EPS programmiert, beschränkt die Steuerung an Konturübergängen bzw. bei zirkularen Bahnsegmenten den Kontur-/Radiusfehler ϵ auf den nach EPS programmierten Wert. Wird PrecProg in Verbindung mit DIST programmiert, beschränkt die Steuerung an Konturübergängen den Bahn-Nachlauf δ auf den nach DIST programmierten Wert. Beim Verfahren von zirkularen Bahnsegmenten beschränkt die Steuerung jetzt den Radiusfehler ϵ entsprechend MP 8003 00001. DIST und EPS dürfen nicht gleichzeitig programmiert werden.
PrecProg(0)	Genauigkeitsprogrammierung AUS.

Tab. 8-1: Relevante NC-Funktionen

Relevante Maschinenparameter (MP)

8003 00001	Toleranz für Konturtreue bei Genauigkeitsprogrammierung PrecProg .

Tab. 8-2: Relevante Maschinenparameter

Relevante SD-Funktionen

SD(328,1)	Liefert den zuletzt programmierten maximal erlaubten Kontur-/Radiusfehler ϵ im aktiven Kanal als gerundeten Integerwert. Alternative Programmierung: SD(328).
SD(328,2)	Liefert den zuletzt programmierten maximal erlaubten Bahn-Nachlauf δ im aktiven Kanal als gerundeten Integerwert.
SDR(328,1)	wie SD(328,1), Rückgabewert jedoch im Realformat. Alternative Programmierung: SDR(328).
SDR(328,2)	wie SD(328,2), Rückgabewert jedoch im Realformat.

Tab. 8-3: Relevante SD-Funktionen

Ist die Funktion "Genauigkeitsprogrammierung" nicht aktiv, liefern alle Funktionen den Wert 0.

8.1.2 Handlungsanweisung: Genauigkeitsprogrammierung

Aktivieren/deaktivieren der Genauigkeitsprogrammierung. Die Genauigkeitsprogrammierung führt zu einer Beschränkung der maximalen Abweichung an Konturübergängen und bei zirkularen Bahnsegmenten (Kreise, Helical, HelicalN).

IW-Engineering / Configuration: Toleranz festlegen

Der Parameter **ContDevTol** "Toleranz für Konturtreue" (8003 00001) legt fest, wie groß der Bahnfehler am Konturübergang und der Radiusfehler am Kreisbogen maximal sein darf.

	Documentation	Instruction
Instruction:		Editieren von Maschinenparametern
Documentation:	MTX Maschinenparameter	Genauigkeitsprogrammierung

IW-Operation / NC-Programmierung: aktivieren/modifizieren

- **PrecProg(1)**
- oder -
- **PrecProg(EPS<Konturfehler ϵ >)**
- oder -
- **PrecProg(DIST<Bahn-Nachlauf δ >)**
programmieren.



Die Funktion "Genauigkeitsprogrammierung" wirkt für Konturübergänge nur dann, wenn G8 aktiv ist. Bei zirkularen Bahnsegmenten wirkt sie immer.

Die Funktion "Genauhalt" muss ausgeschaltet sein (G62 ist aktiv).

Alle an der Bahn beteiligten Achsen müssen bezüglich der Dynamik identisch parametrisiert sein.

DIST und EPS dürfen nicht gleichzeitig programmiert werden.

	Documentation	
Documentation:	MTX Maschinenparameter	Genauigkeitsprogrammierung

IW-Operation / NC-Programmierung: deaktivieren

PrecProg(0) programmieren

	Documentation	
Documentation:	MTX Programmierhandbuch	Genauigkeitsprogrammierung

9 Bahnbewegung

9.1 Spline

9.1.1 Überblick

Funktion

Im Vergleich zur Linear-Interpolation erreicht die Spline-Interpolation mit einer geringeren Anzahl von Stützpunkten die gleiche Oberflächen- oder Konturgenauigkeit, da zwischen den Punkten **Kurven** erzeugt werden.

Die Spline-Funktion bietet

- verschiedene Verarbeitungsoptionen von Spline-Daten, die vom CAD/CAM- oder vom Digitalisierungssystem kommen
- verschiedene Spline-Typen für praxisgerechte Anforderungen (z. B. 3- und 5-achsige Freiformflächenbearbeitung)
- Parametrierung von Randbedingungen für die anwendungsgerechte Spline-Bearbeitung im NC-Programm

Ursprung der Spline-Daten: Punktfolgen können wie folgt vorgegeben werden:

- manuell im NC-Programm programmieren
- Spline per CAD-Programm mit Postprozessor-Funktion erzeugen und nur die Spline-Koeffizienten im NC-Programm übermitteln

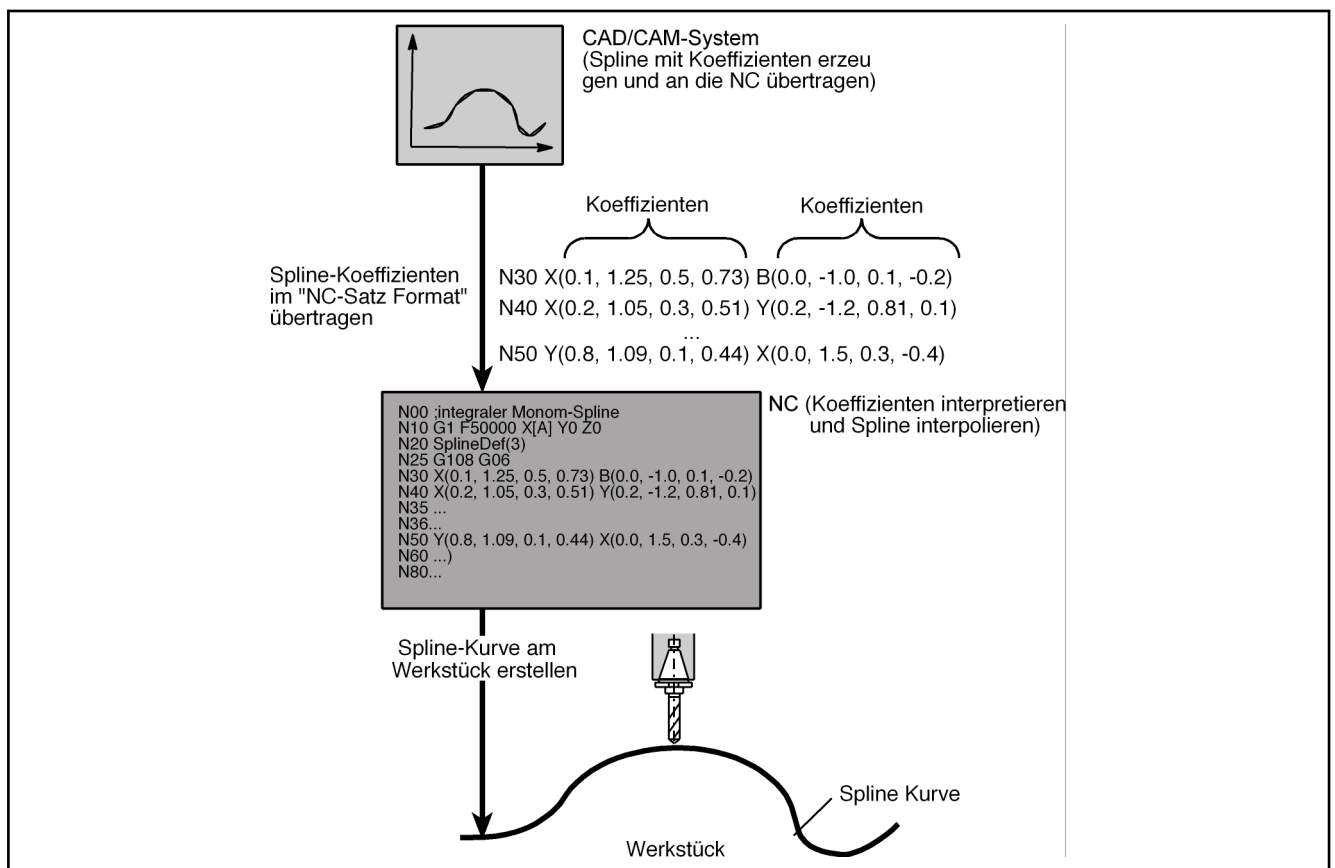


Abb. 9-1: CAD/CAM-System (Spline mit Koeffizienten erzeugen und an die NC übertragen)

Bahnbewegung

- ein CAD/CAM-System erzeugt für das NC-Programm Stütz- oder Kontrollpunkte, aus denen die NC Spline-Kurven generiert werden (siehe folgende Abbildung)
- beim Digitalisieren: Oberflächen als einzelne Punkte erfassen (siehe folgende Abbildung)

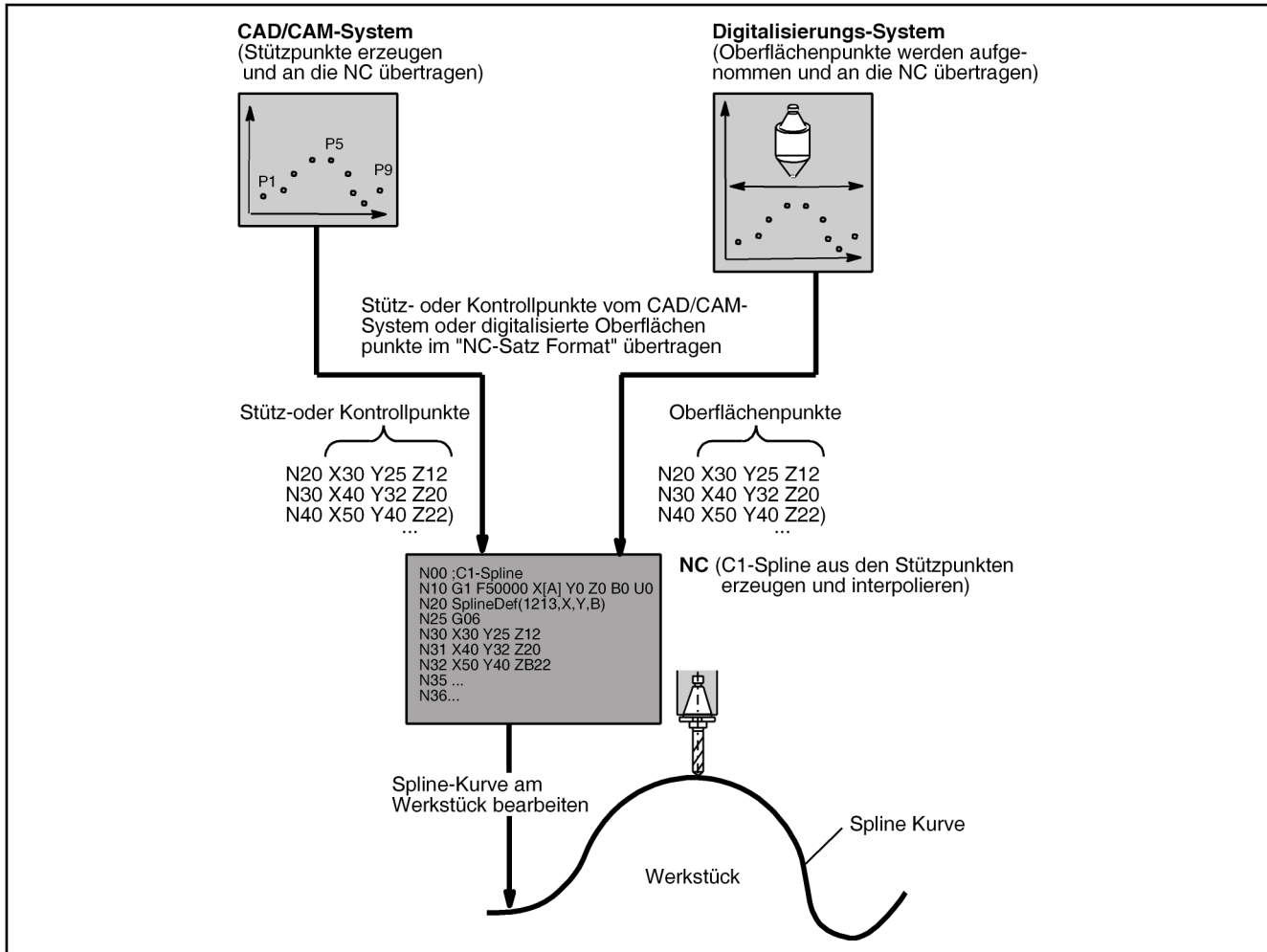


Abb. 9-2: Digitalisieren

- Vorteile:**
- Weichere Linienführung der Kontur und glattere Oberflächen am Werkstück
 - Die Anzahl der vom CAD/CAM-Programm oder von der Digitalisierung erzeugten Punkte kann unter Einhaltung der erforderlichen Genauigkeit ggf. verringert werden
 - Die Maschine und ihre bewegten Teile werden im Gegensatz zum eckigen Übergang bei Linearinterpolation durch die Spline-Interpolation **geringer** belastet, da die Richtungsänderung des Werkzeugs weniger abrupt verläuft

Spline-Typen (Überblick): Die NC unterstützt folgende Spline-Programmierungen:

- **Typ 0: Splines mit Koeffizientenprogrammierung**
Jeder Spline kann als Polynom (z. B. Parabel = quadratisches Polynom) dargestellt werden. Dessen Koeffizienten bestimmen das Aussehen der Kurve, den Start- und den Endpunkt. Die Koeffizienten werden vom

CAD/CAM-System geliefert. Dort werden sie einzeln als NC-Sätze programmiert, an die NC übertragen und in der NC interpoliert.

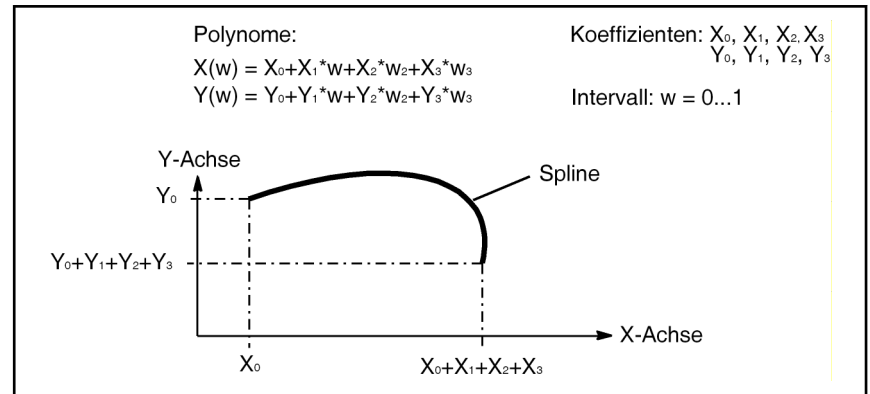


Abb. 9-3: Splines mit Koeffizientenprogrammierung

Vorteile:

Die vom CAD/CAM-System gelieferten Daten bestimmen den Spline.

Nachteile:

Das CAD/CAM-System muss über eine entsprechende Postprozessor-Schnittstelle verfügen. Per Parametrierung in der NC sind keine Korrekturen möglich. Mögliche Abweichungen sind erst während des Bearbeitungsprozesses erkennbar.

- **Typ 1: c¹-stetige kubische Splines mit Stützpunktprogrammierung**

Die MTX unterscheidet zwei Unterarten:

- Im NC-Programm stehen ausschließlich die Stützpunkte. Die NC erzeugt daraus kubische Splines, die an den Stützpunkten tangential sind, d. h. c¹-stetig. Die Tangenten an den Übergängen werden nach dem Akima-, Bessel- oder Sehnen-Verfahren berechnet und wirken sich auf maximal vier Stützpunkte aus.
- Hermite-Splines: Im NC-Programm stehen neben den Stützpunkten auch die Tangenten (Ableitung der Koordinate nach der Bahnlänge). Die NC erzeugt daraus kubische Splines, die mindestens c¹-stetig sind.

Vorteile:

Geringer Rechenaufwand in der NC. Nur ein Satz Look-Ahead notwendig (lokaler Spline).

Nachteile:

Die Übergänge an den Stützstellen sind nicht krümmungsstetig, dadurch treten Sprünge in der Achsbeschleunigung auf.

- **Typ 2: c²-stetige kubische Splines mit Stützpunktprogrammierung**
- Im NC-Programm stehen ausschließlich die Stützpunkte. Die NC erzeugt daraus kubische Splines, die an den Stützpunkten krümmungsstetig (c²-stetig) sind.

Vorteile:

Keine Sprünge in der Achsbeschleunigung.

Nachteile:

Hoher Rechen- und Speicheraufwand in der NC. Große Anzahl von Sätzen im Look-Ahead (globaler Spline).

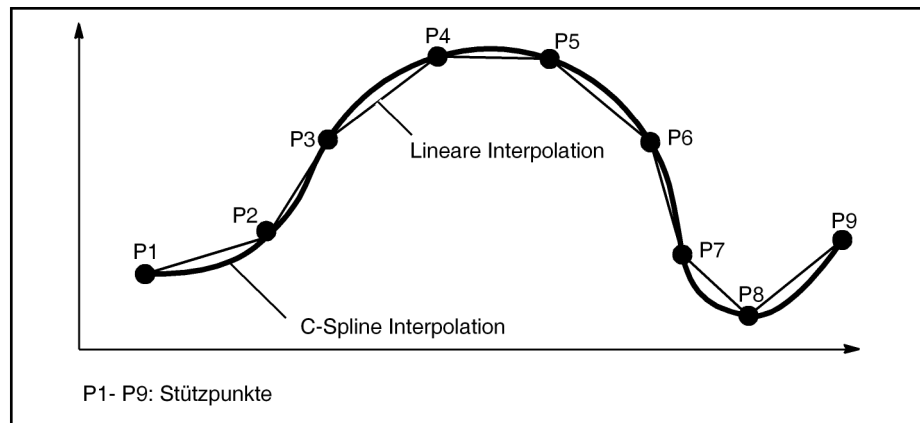


Abb. 9-4: kubische Splines

- **Typ 3: B-Splines mit Kontrollpunktprogrammierung**

Bei der B-Spline-Programmierung (z. B. NURBS) stehen im NC-Programm die Kontrollpunkte der B-Spline-Kurve. Die Kurve verläuft nicht durch die Kontrollpunkte, sondern nähert sich mehr oder weniger dem Kontrollpunktpolygon an (siehe folgende Abbildung).

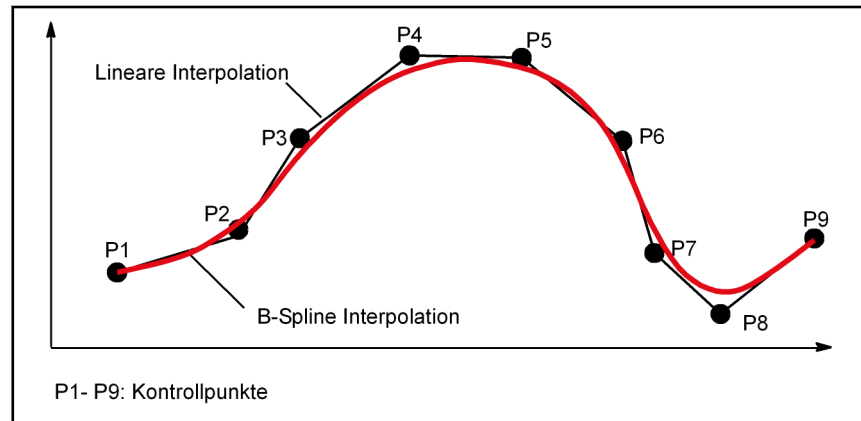


Abb. 9-5: B-Splines mit Kontrollpunktprogrammierung

Vorteile:

- Geringerer Rechenaufwand in der NC als bei C²-stetigen kubischen Splines
- Geringere Anzahl von Sätzen im Look-Ahead (lokaler Spline)
- Alle Kurven, die durch Kegelschnitte (Ellipse, Kreis, Parabel, Hyperbel) definiert sind, können exakt durch B-Splines dargestellt werden

Nachteile:

Die Kurve geht nicht durch die Kontrollpunkte, sondern nähert sich ihnen nur an.

- **Typ 4: Approximierende B-Splines mit Toleranzangabe (Kompressor)**

Die sich ergebende Spline-Kurve approximiert die programmierten Punkte mit einer vorgegebenen Genauigkeit.

Damit werden

1. ungenau programmierte Punkte geglättet
2. kurze NC-Sätze zusammengefasst (Kompressor)

9.1.2 Spline mit Koeffizientenprogrammierung

Allgemeines



Die Koeffizientenprogrammierung von Splines setzt ein entsprechendes CAD/CAM-Programmiersystem voraus, das die Koeffizienten in einem NC-Programm generieren kann.

Integrale Monom-Splines mit Koeffizientenprogrammierung

Der Spline wird durch ein Polynom bis zum Grad $n=5$ dargestellt.

Die $n+1$ Koeffizienten des Polynoms werden **direkt** im NC-Satz vorgegeben.

$$\underbrace{\vec{r}(w)}_{\text{Vektor}} = \underbrace{\sum_{j=0}^n r_j w^j}_{\text{Polynom}} \quad w \in [0, w_e]$$

Beispiel:

Monom-Splines (z. B. kubischer Spline: Grad 3) mit Koeffizientenprogrammierung. Hierbei wurde angenommen, dass die Kanalachsen X, Y, Z, A und B die Spline-Bewegung bilden.

Vektor Kanalachsen Polynom 3.Grades

$$\vec{r}(w) = \begin{pmatrix} X(w) \\ Y(w) \\ Z(w) \\ A(w) \\ B(w) \end{pmatrix} = \begin{pmatrix} X_0 \\ Y_0 \\ Z_0 \\ A_0 \\ B_0 \end{pmatrix} + \begin{pmatrix} X_1 \\ Y_1 \\ Z_1 \\ A_1 \\ B_1 \end{pmatrix} \cdot w + \begin{pmatrix} X_2 \\ Y_2 \\ Z_2 \\ A_2 \\ B_2 \end{pmatrix} \cdot w^2 + \begin{pmatrix} X_3 \\ Y_3 \\ Z_3 \\ A_3 \\ B_3 \end{pmatrix} \cdot w^3$$

Intervallgröße $w \in [0, w_e]$

3+1 Koeffizienten

Beispiel:

Vorgabe von Koeffizienten im NC-Programm für die Kanalachsen X und B

G06 X(0.1, 1.25, 0.5, 0.73) B(0.0, -1.0, 0.1, -0.2)

Daraus ergibt sich für die Kanalachse X und B:

$$X(w) = 0.1 + 1.25w + 0.5w^2 + 0.73w^3$$

$$B(w) = 0.0 - 1.0w + 0.1w^2 - 0.2w^3$$

Aus den Koeffizienten werden im Definitionsintervall 0 bis w_e die Spline-Kurve erzeugenden Vektoren errechnet, deren Komponenten die Kanalkoordinaten darstellen. Aus den Kanalkoordinaten werden, gegebenenfalls unter Berücksichtigung einer aktiven Achstransformation, NC-intern die Achspositionen der physikalischen Achsen berechnet.

Beispiel:

Spline-Kurve

$\vec{r}(w)$ mit den beiden Kanalkoordinaten x und y

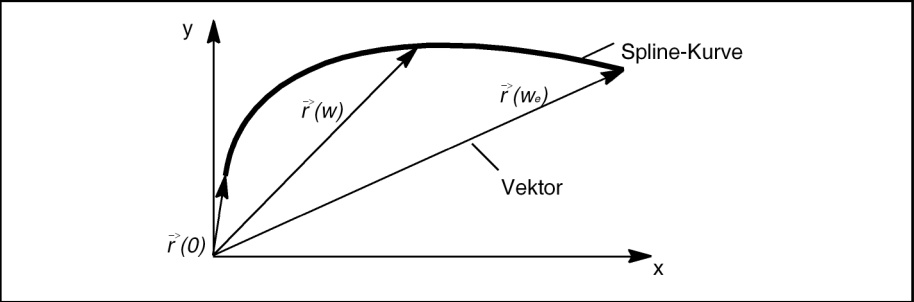


Abb. 9-6: Spline-Kurve mit den beiden Koordinaten x und y

Spline-Parameterlänge

Die Spline-Parameterlänge definiert die Länge des **Definitionsintervalls** von **w**. Programmiert wird das **Definitionsintervall** im NC-Satz über die Vorschrift **PL...**

Beispiel:

Spline_Parameterlänge

G06 X(0.1, 1.25, 0.5, 0.73) B(0.0, -1.0, 0.1, -0.2) **PL0.6**

Gemischte Programmierung

Innerhalb eines jeden NC-Satzes können Koordinaten wahlweise als **Spline** oder als **Gerade** programmiert werden.

Beispiel:

Gemischte Programmierung - Anwendungsbeispiel

```
N10 G1 X100 Y100 Z20 B10 U0 F1000
N20 SplineDef(3)
N30 G2 X200 Y100 Z30 I100 J0
N40 G06 X(200,1,0.5,-0.1) Y(100,-1,0.2,0.2) B(10,1.0) Z60 U10 PL10
N50 X(160,20) Y(310,4.0,0.2,-0.01) Z0
N60 G1 X10 Y20 Z30 B40
N70 G06 Y(...) ..... PL12
```

N20	Der Syntax G06 wird ein kubischer Spline zugewiesen.
N40	Umschaltung vom Bahntyp Helix auf Spline. X, Y und B haben Koeffizientenprogrammierung und werden in diesem Satz als Spline-Mitglieder betrachtet. Z und U haben Endpunktprogrammierung und bewegen sich linear. Der Spline-Parameter "w" durchläuft das Intervall [0,10].

N40	$X(w)=200+1.0w+0.5w^2-0.1w^3$	Endpunkt:	$X(10)=160$
	$Y(w)=100-1.0w+0.2w^2+0.2w^3$	Endpunkt:	$Y(10)=310$
	$B(w)=10+1.0w+0.0w^2-0.0w^3$	Endpunkt:	$B(10)=20$
N50	zweiter Spline im NC-Programm Die Spline-Parameterlänge ist 10 (modale Wirkung von PL aus N40) X und Y sind Spline-Mitglieder Z bewegt sich linear		
N60	Umschaltung auf Bahntyp Gerade		
N70	Umschaltung auf Bahntyp Spline PL muss erneut programmiert werden		

Rationale Monom-Splines

Wird das Polynom des integralen Monom-Splines mit einem gemeinsamen **Nenner-Polynom** erweitert, erhält man einen rationalen Monom-Spline.

$$\begin{array}{c}
\text{Vektor} \quad \text{Polynom} \\
\begin{array}{c} \underbrace{\hspace{1cm}} \end{array} \quad \begin{array}{c} \underbrace{\hspace{1cm}} \\ \sum_{j=0}^n \vec{r}_j \cdot w^j \end{array} \\
\vec{r}(w) = \frac{\sum_{j=0}^n \vec{r}_j \cdot w^j}{\underbrace{\sum_{j=0}^n r_j \cdot w^j}_{\text{Nenner-Polynom}}} \quad w \in [0, w_e]
\end{array}$$

Programmiert wird das Nenner-Polynom durch Koeffizientenprogrammierung im NC-Satz über die Vorschrift DN(...).

Beispiel:

Vorgabe von Koeffizienten im NC-Programm für die Kanalachsen X und B
G06 X(0.1, 1.25, 0.5, 0.73) B(0.0, -1.0, 0.1, -0.2) **DN(1.0,0.0,1.0)**

Daraus ergibt sich für die Kanalachse:

$$X(w) = \frac{0.1 + 1.25w + 0.5w^2 + 0.73w^3}{1 + w^2}$$

und für

$$B(w) = \frac{0.0 - 1.0w + 1.0w^2 - 0.2w^3}{1 + w^2}$$

Beispiel:

Anwendungsbeispiel

N00 ;Ellipse als rationaler Monom-Spline

0001 A=100

0002 B=50

N10 G1 F50000 X[A] Y0 Z0

```

N20 SplineDef(2)
N25 G108 G06
N30 X(A,0,-A) Y(0,2*B) DN(1,0,1)PL1
N40 X(0,-2*A,A) Y(2*B,-2*B) DN(2,-2,1)
N50 X(-A,0,A) Y(0,-2*B) DN(1,0,1)
N60 X(0,2*A,-A) Y(-2*B,2*B) DN(2,-2,1)
N80 G1

```



Alle Kegelschnitte (Ellipse,...) lassen sich durch rationale Monom-Splines darstellen.

Die Splines haben darüber hinaus folgende Eigenschaften:

- Eine C^0 -Stetigkeit liegt vor, wenn der Startpunkt eines Splines mit dem Endpunkt im Vorgängersatz identisch ist. Durch ungenaue Koeffizientenprogrammierung kann es zu Differenzen zwischen Start- und Endpunkt kommen. Für den Satzübergang kann man in MP 8007 00010 einen Toleranzwert e_0 einstellen.
- Die Bahngeschwindigkeit auf der Spline-Kurve wird aus dem Zusammenhang zwischen der Bahnlänge und dem Spline-Parameter w intern über eine "Bahnlängenfunktion" angenähert.
In MP 8007 00020 kann dazu eine Toleranz ϵ_v als **maximal auftretender relativer Bahngeschwindigkeitsfehler** eingestellt werden. Mit zunehmender Genauigkeitsanforderung muss ein kleinerer Wert eingegeben werden. Das führt jedoch auch zu einer steigenden NC-Blockzykluszeit. Wird die vorgegebene Genauigkeit nicht erreicht, erzeugt die NC eine Warnung, die Bearbeitung wird jedoch fortgesetzt.
Wird MP 8007 00020 auf 100 % gesetzt, wird die Bahnlängenfunktion durch eine Gerade approximiert.
- Auch an der Spline-Kurve beteiligte Achsen können aus der Vorschubbildung (FeedGrp-Programmierung) herausgenommen werden. Es darf dann jedoch nicht der Fall auftreten, dass die Spline-Kurve einen Bahnpunkt enthält, indem alle Vorschub bildenden Koordinaten/Achsen eine sogenannte Null-Tangente (d. h. keine Verschiebung in Achsrichtung) haben. In diesem Fall wird ein Laufzeitfehler erzeugt.

Beispiel:

Die Raumkurve in der folgenden Abbildung wird als Kurve der Achsen X und Y betrachtet.

Der Spline $\vec{r}(w)$ hat zwei Bahnpunkte mit einer Tangente entlang der X-Achse. Ist nun X nicht Vorschub bildend (FeedGrp(Y)), so tritt genau der Fehlerfall auf.

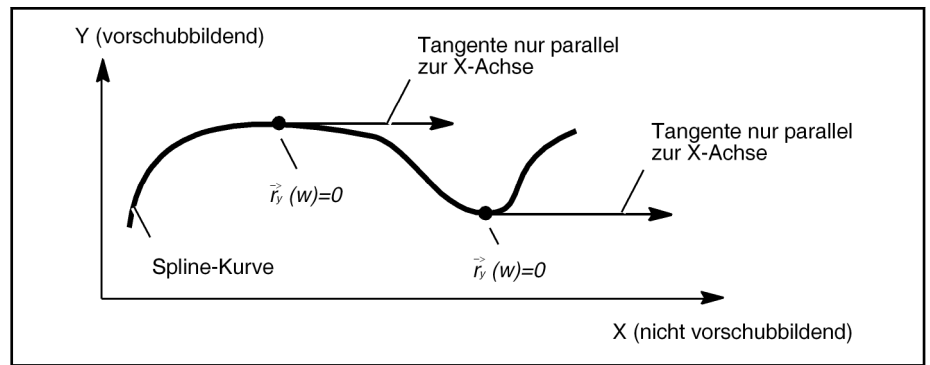


Abb. 9-7: Fehlerfall

9.1.3 Spline mit Stützpunktprogrammierung und tangentialen Übergängen (C1-stetige kubische Splines)

Funktion

Die Splines werden aus vorgegebenen Punkten (n-Stützstellen) und den Tangenten dieser Punkte berechnet. Die Punkte werden im NC-Programm vorgegeben.

Die Tangenten in den Punkten $\vec{r}_i (i = 2, \dots, n-1)$ werden aus jeweils drei aufeinander folgenden Punkten $\vec{r}_{i-1}$, $\vec{r}_i$, $\vec{r}_{i+1}$ berechnet.

Die Tangenten an den Punkten $\vec{r}_1$ und $\vec{r}_n$ ergeben sich aus den Start- und Endbedingungen (siehe [Kap. "Tangentenberechnung" auf Seite 128](#)).

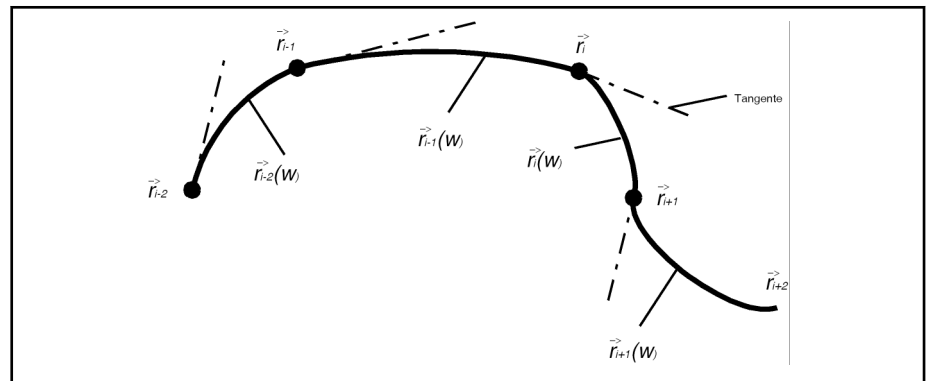


Abb. 9-8: Spline mit Stützpunktprogrammierung und tangentialen Übergängen

Die Splines haben folgende Eigenschaften:

- Aus n Punkten und einer Start- und Endbedingung der Spline-Sequenz entstehen n-1 Splines.

Ein Spline $\vec{r}_i(w)$ verbindet die Punkte $\vec{r}_i$ und $\vec{r}_{i+1}$.

- Sie sind C¹-stetig (tangential) an den Übergangsstellen $\vec{r}_i$.
- Sie sind lokal, d. h. die Veränderung eines Punktes $\vec{r}_i$ beeinflusst nur eine beschränkte Anzahl von Splines in seiner Nachbarschaft.

Es sind hier die Splines $\vec{r}_{i-2}(w)$, $\vec{r}_{i-1}(w)$, $\vec{r}_i(w)$, $\vec{r}_{i+1}(w)$.

Umgekehrt wird ein Spline nur durch die Punkte $\vec{r}_{i-1}$, $\vec{r}_i$, $\vec{r}_{i+1}$, $\vec{r}_{i+2}$ festgelegt.

Tangentenberechnung

Der tangentielle Übergang wird mittels Parametrierung (Spline-Id siehe [Kap. "Funktion" auf Seite 140](#)) einstellt. Hierfür stehen die folgenden Verfahren zur Verfügung:

Bessel-Verfahren:

Durch die programmierten Punkte $\vec{r}_{i-1}$, $\vec{r}_i$, $\vec{r}_{i+1}$ wird eine Parabel (Spline 2. Grades) gelegt. Daraus folgt die Tangente in $\vec{r}_i$.

Länge und Richtung der Tangenten hängen von der gewählten Spline-Parametrisierung ab (siehe [Kap. "Tangentenberechnung" auf Seite 128](#))

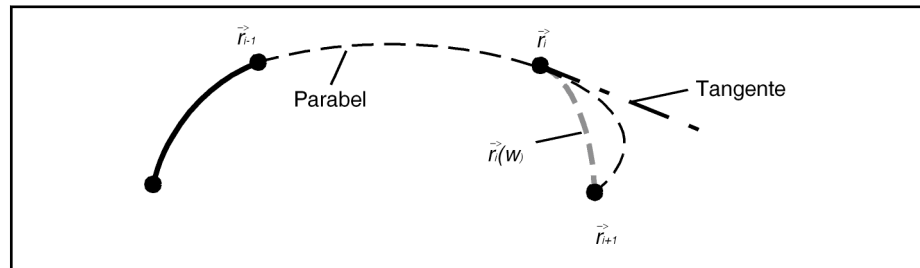


Abb. 9-9: Bessel-Verfahren

Akima-Verfahren:

Die Tangente in $\vec{r}_i$ zeigt in die Richtung der Winkelhalbierenden der beiden Sehnen $\vec{r}_{i-1}\vec{r}_i$ und $\vec{r}_i\vec{r}_{i+1}$.

Die Tangentenlänge hängt von der gewählten Spline-Parametrisierung ab (siehe [Kap. "Tangentenberechnung" auf Seite 128](#)).

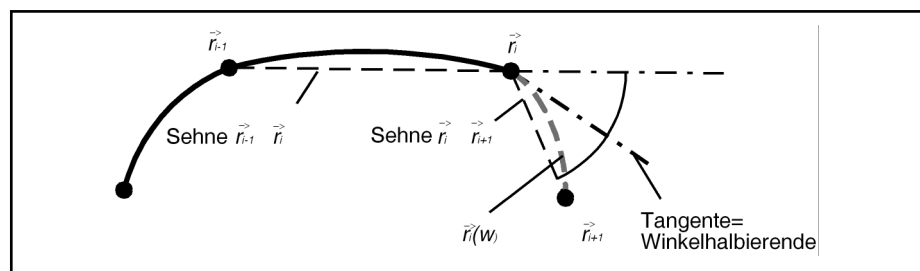


Abb. 9-10: Akima-Verfahren

Sehnen-Verfahren:

Die Tangente in $\vec{r}_i$ zeigt in die Richtung der Sehne $\vec{r}_{i-1}\vec{r}_{i+1}$.

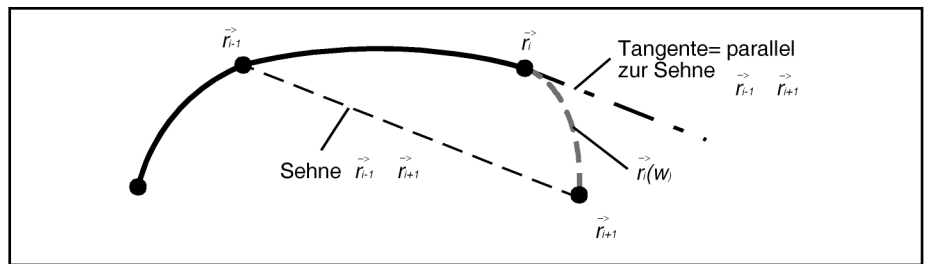


Abb. 9-11: Sehnen-Verfahren

Start- und Endbedingung: An beliebigen Stellen innerhalb einer Spline-Sequenz – in den meisten Fällen wird dies der Beginn und das Ende sein – können Randbedingungen programmiert werden.

Die Programmierung erfolgt mit der Syntax

- "Start Boundary Condition" **SBC**((<Typ>[,<Values>]) und
- "End Boundary Condition" **EBC**((<Typ>[,<Values>])).

Für Start- und Endbedingungen sind auswählbar:

- Typ = 1:** Vorgabe der Tangentenrichtung am Start-/Endpunkt der Spline-Sequenz. Nur das Verhältnis der Werte der Spline-Mitglieder ist von Bedeutung. Die Tangentenlänge wird intern berechnet (Bei chordaler Parametrisierung ist die Tangentenlänge = 1).
- Typ = 2:** Vorgabe der zweiten Ableitung r_{ii} nach w am Start-/Endpunkt der Spline-Sequenz.
- Typ = 10:** Die Spline-Tangentenrichtung ist entlang der Verbindungslinie Startpunkt – Endpunkt.
- Typ = 11:** Die Spline-Tangentenrichtung ist entlang der Verbindungslinie Startpunkt – Endpunkt des Vorgängersatzes (bei SBC) oder des Nachfolgersatzes (bei EBC).

Durch Verwenden von SBC und EBC innerhalb einer Spline-Sequenz können gezielt Ecken auf der Spline-Kurve erzeugt werden. Ohne programmierte Start- und Endbedingung wird die sogenannte natürliche Randbedingung $SBC(2,0,...,0)$ oder $EBC(2,0,...,0)$ verwendet.

Bei Verwendung des Typs 1 und 2 ist für jedes Spline-Mitglied ein Wert in der Liste <Values> anzugeben. Ist die Vektororientierung als Spline aktiv, ist in den Randbedingungen auch der Orientierungsvektor enthalten.

Beispiel:

```
SplineDef(1213,x,y,z,O)
```

```
G06 ..
```

```
... EBC(1,1.0,1.5,-0.5,0.0,-1.0.0.0)
```

Das erste Werte-Tripel (1.0,1.5,-0.5) ist die Tangentenrichtung der Raumbewegung x,y,z.

Das zweite Tripel (0.0,-1.0.0.0) ist die Tangentenrichtung des Orientierungsvektors

Spline-Parameterlänge: Die Spline-Parameterlänge berücksichtigt die Stützpunktweite und Spline-Kurvenkrümmung. Je nach Anwendung sind drei unterschiedliche Parametrisierungen zulässig:

- **Äquidistante** Parametrisierung ($w=1$) :

Diese Parametrisierung ist für gleiche Spline-Stützpunktstände geeignet. Sind die Spline-Stützpunkte unterschiedlich weit entfernt, führt dies insbesondere bei Krümmungsstetigkeit verstärkt dazu, dass die Kurve "ausbeult" (siehe nächste Abbildung).

$$w = \left| \vec{r}_{i+1} - \vec{r}_i \right|$$

- **Chordale** Parametrisierung

Sind die Spline-Stützpunkte unterschiedlich weit entfernt, wird der Abstand und die Richtung der Stützpunkte berücksichtigt.

$$w = \sqrt{\left| \vec{r}_{i+1} - \vec{r}_i \right|^2}$$

- **Zentripetale** Parametrisierung

Diese Parametrisierung wird eingesetzt, wenn die Spline-Stützpunkte unterschiedlich weit entfernt sind und die Splinekurven starke Krümmungen aufweisen.

Der Kurvenverlauf ist umso glatter und hat umso weniger unerwartete Krümmungen, je besser Lage und Abstände der Stützpunkte berücksichtigt werden. Daher kann eine Kombination der o. g. Parametrisierungen je nach Stützpunktvorgabe erforderlich sein.

Die Spline-Parameterlänge kann statt der NC-berechneten Parametrisierung auch mit dem Parameter PL<w> manuell vorgegeben werden.

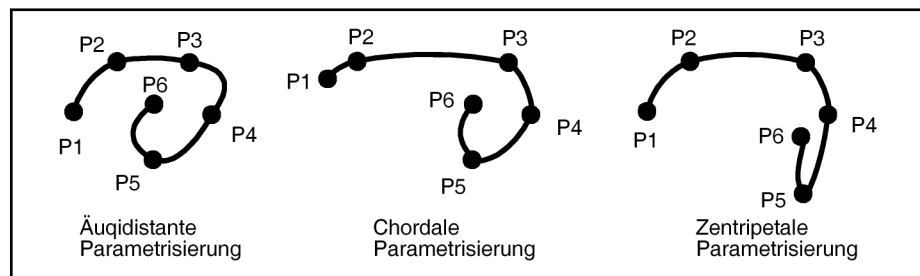


Abb. 9-12: Spline-Parameterlänge

Bewegung der Werkzeugorientierung

Nicht nur die Werkzeugspitze kann auf einer Spline-Kurve verfahren, sondern auch die **Werkzeugorientierung**, wenn es sich um ein rotationssymmetrisches Werkzeug handelt, das zusammen mit einer entsprechenden 5-Achstransformation programmiert wird. Man unterscheidet zwischen Vektororientierung und linearer Orientierung

(siehe auch [Kap. 3 "Antriebe \(Achsen, Spindeln\)"](#) auf Seite 13).



Die Orientierung nichtrotationssymmetrischer Werkzeuge (Tensororientierung in Verbindung mit der entsprechenden 6-Achstransformation) kann nicht als Spline programmiert werden.

- **Vektororientierung als C¹-stetiger kubischer Spline:**

Bei der Vektororientierung als C¹-stetiger kubischer Spline wird in jedem NC-Satz ein Spline in den sechs Koordinaten x, y, z, p_x, p_y, p_z mit einer gemeinsamen Parametrisierung (äquidistant, chordal, zentripetal) erzeugt. Das in der Funktion "SplineDef" angegebene Tangentenverfahren (Bessel-, Akima- und Sehnenverfahren) wird auch für den Orientierungsvektor

benutzt. An den Satzübergängen sind sowohl die Bahnbewegung (x, y, z) als auch der Orientierungsvektor C¹-stetig.

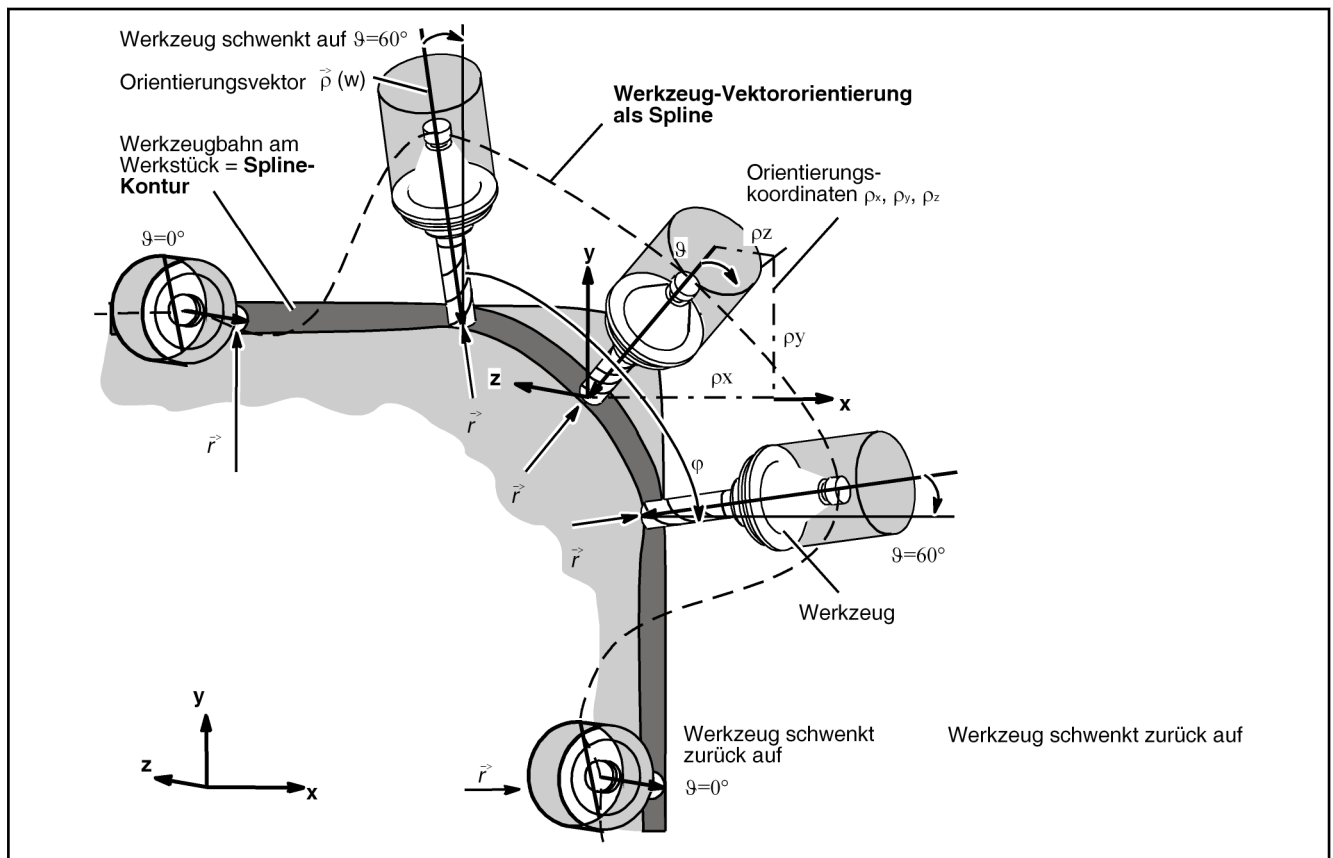


Abb. 9-13: Vektororientierung

- **Lineare Orientierungsbewegung als C^1 -stetiger kubischer Spline**

Bei der linearen Orientierungsbewegung werden Splines in den fünf Koordinaten x , y , z , ϕ , θ erzeugt. Die Splines sind mathematisch äquivalent zu Splines bei einer Achsbewegung in drei Linear- und zwei rotatorischen Achsen (siehe [Kap. "Bewegung der Werkzeugorientierung"](#) auf Seite 130).

N0 x0 y0 z0 phi0 theta0

N0 G06 G8

N1 y20

N2 y40 theta30

N3 y60 theta0

N4 y80

N5 ...

N1	In diesem Satz beginnt bereits die theta-Orientierungsbewegung des Werkzeugs.
N4	Erst in diesem Satz endet die theta-Orientierungsbewegung des Werkzeugs.

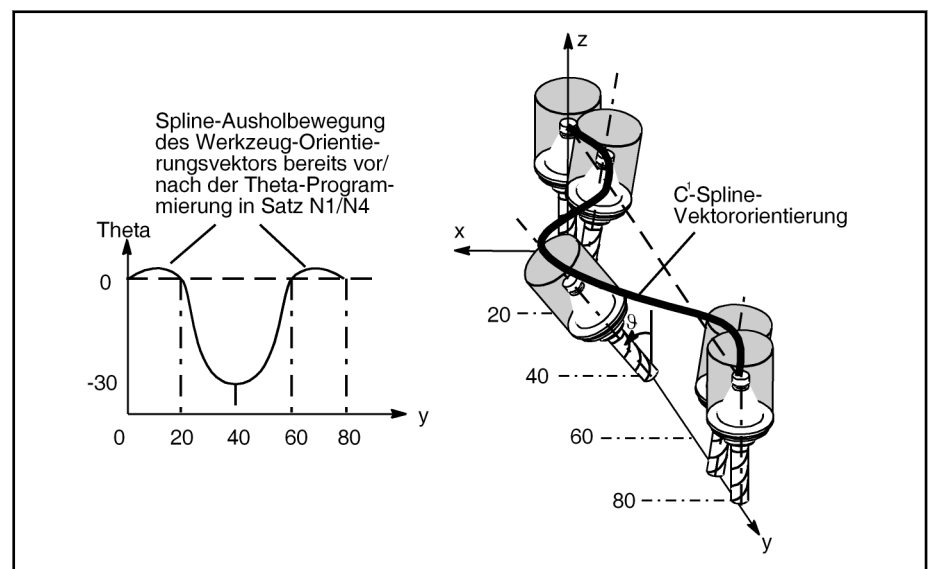


Abb. 9-15: Beispiel

9.1.4 Kubische Hermite-Splines

Allgemeines



Hermite-Splines setzen ein entsprechendes CAD/CAM-System voraus, das sowohl die Punkte als auch deren Ableitungen generieren kann

Ein kubischer Spline wird allgemein durch folgendes Monom-Polynom dargestellt

$$\vec{r}(w) = \vec{c}_0 + \vec{c}_1 \cdot w + \vec{c}_2 \cdot w^2 + \vec{c}_3 \cdot w^3; \quad w \in [0, W]$$

mit dem Kurvenparameter w und den vier Spline-Koeffizienten c_0, c_1, c_2, c_3 . Dabei entspricht c_0 dem Startpunkt und c_1 der Starttangente. c_2, c_3 sind nicht direkt geometrisch interpretierbar.

Eine zu dem Monom-Polynom mathematisch äquivalente Darstellung der Hermite-Splines erfolgt über Bézier-Polynome

$$\vec{r}(w) = \sum_{i=0}^3 \vec{P}_i B_{i,3}(u); \quad u \in [0, 1]$$

mit den Bernstein-Basispolynomen

$$B_{i,3}(u) = \binom{3}{i} u^i (1-u)^{3-i}$$

Generell werden bei Hermite-Splines sowohl die Stützpunkte r_i ($i=1,\dots,n$) als auch die Tangenten an den Stützpunkten (Ableitungen der Koordinaten nach dem Weg) von einem CAD/CAM-System berechnet und anschließend der NC vorgegeben (siehe nachfolgende Abbildung).

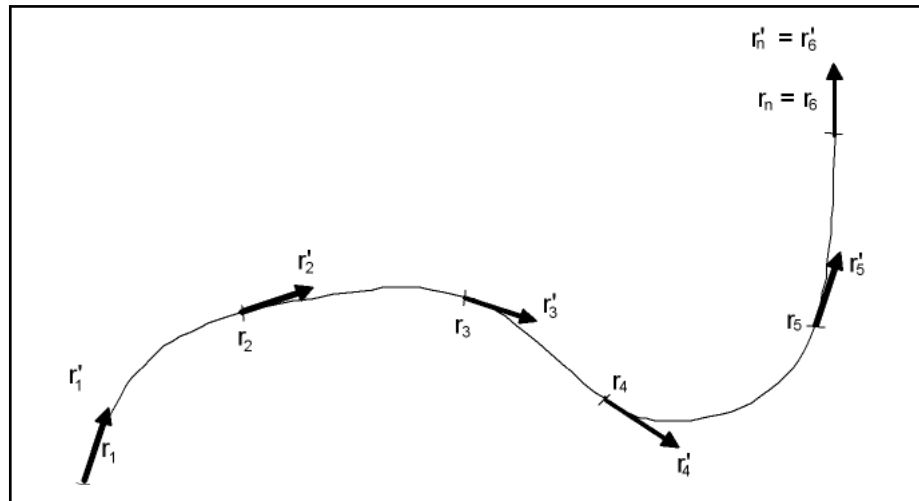


Abb. 9-16: Hermite-Spline

Damit hat man steuerungsintern für jedes kubische Teil-Polynom $r_i(w)$ vier Bedingungen r_i , r'_i , r_{i+1} , r'_{i+1} gegeben, aus denen sich die Koeffizienten berechnen lassen. Die Satzübergänge sind mindestens c^1 -stetig. An dem Startpunkt r_1 wird von einer Starttangente mit dem Wert=0 ausgegangen.

Jede an der Splinebewegung beteiligte Kanalachse oder Koordinate wird mit ihrem Positionswert und ihrem Ableitungswert programmiert. Diese Werte werden in Klammern hinter dem Kanalachs- oder Koordinatennamen angegeben. Die Parameterlänge kann, wie bei den Splines mit Koeffizientenprogrammierung, mit programmiert werden

Beispiel:

N10 SplineDef(1143)	G6 wird ein Hermite-Spline zugewiesen
N20 G1 X100 Y100 Z0 F60	Ausgangsposition anfahren
N25 G6 CPI(1) CVP	Spline, Kurvenparameterinterpolation "CPI(1)" und Sprunginterpolator "CVP" aktivieren
N30 G6 N40 X(200.0,1.0) Y(100.0,-1.0) PL10	Param.-Länge = 10 (entspricht 10 s wegen F = 60)
N50 X(160.0,20.0) Y(310.0,4.0) Z(100.0,2.0)	Param.-Länge = 1 (da kein PL programmiert)

N70 F60000 X(210.0,2.0) Y(220.0,2.1) PL3	Param.-Länge = 3 (entspricht 3 ms wegen F = 60000)
N90 G1 X10 Y20 Z30	Spline beenden, Endposition anfahren

Die Hermite-Splines haben folgende Eigenschaften:

- Aus n Punkten und n Tangenten der Spline-Sequenz entstehen n-1 Splines. Ein Spline $r_i(w)$ verbindet die Punkte r_i und r_{i+1} .
- Sie sind C^1 -stetig (tangential) an den Übergangsstellen.
- Sie sind lokal, d. h. die Veränderung eines Punktes r_i beeinflusst nur eine beschränkte Anzahl von Splines in seiner Nachbarschaft. Es sind hier die Splines $r_{i-1}(w), r_i(w)$ und $r_{i+1}(w)$.
- Unter zwei Zusatzbedingungen kann bei den Hermite-Splines die Parameterlänge PL als Ausführungszeit interpretiert werden. Dazu muss, wie im Beispiel-Programm in Zeile N25 angegeben, die Kurvenparameterinterpolation "CPI(1)" und der Sprunginterpolator "CVP" aktiviert werden. Bei F=60 entspricht der PL-Wert der Zeit in [s] und bei F=60000 entspricht der PL-Wert der Zeit in [ms].

9.1.5 Spline mit Stützpunktprogrammierung und krümmungsstetigen Übergängen (C2-stetige kubische Splines)

Funktion

Die Splines werden aus vorgegebenen Punkten $\vec{r}_i$ mit $i=1, \dots, n$ und einer Start- und Endbedingung der Spline-Sequenz berechnet.

Die Splines haben folgende Eigenschaften:

- Aus n Punkten und einer Start- und Endbedingung der Spline-Sequenz entstehen n-1 Splines. Ein Spline $\vec{r}_i(w)$, $w \in [0, w_e]$ verbindet die Punkte $\vec{r}_i$ und $\vec{r}_{i+1}$.
- Sie sind C^2 -stetig (krümmungsstetig) an den Übergangsstellen $\vec{r}_i$.
- Sie sind global, d. h. die Veränderung eines Punktes $\vec{r}_i$ beeinflusst alle Splines der Spline-Sequenz. Der Einfluss nimmt jedoch mit zunehmendem Abstand vom Punkt $\vec{r}_i$ stark ab.

Die Berechnung der Splines erfordert wegen der globalen Spline-Eigenschaft einen deutlich höheren Vorausschaubereich als bei den C^1 -stetigen kubischen Splines.

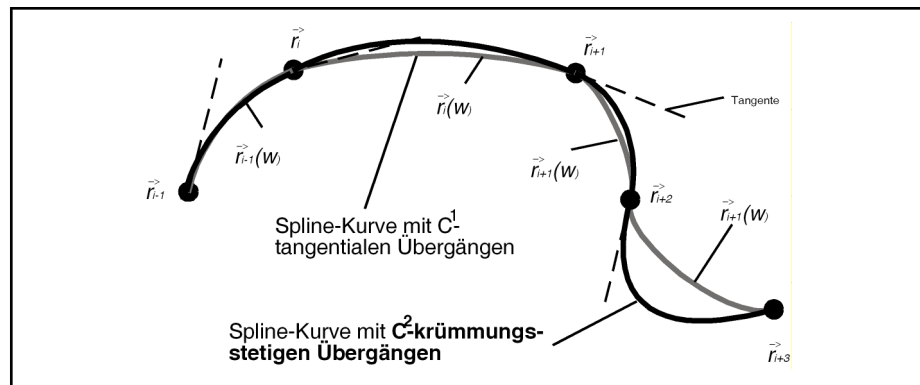


Abb. 9-17: Spline-Kurve

Start- und Endbedingung: Neben den Randbedingungen Typ 1, 2, 10 und 11, die auch für C¹-stetige Splines möglich sind (siehe ["Start- und Endbedingung:"](#) auf Seite 129), existieren zwei weitere Bedingungen:

- Typ =3:
Die **De-Boor-Randbedingung** verknüpft die zweiten Ableitungen an den ersten beiden Stützpunkten oder an den letzten beiden Stützpunkten miteinander.
SBC(3,<v>) oder EBC(3,<v>)
v ist üblicherweise =1
- Typ =4:
Die **periodische Randbedingung** wird für geschlossene Kurven (z. B. Ellipse) benutzt. In diesem Fall muss der letzte und der erste Punkt der Spline-Sequenz übereinstimmen.
Die periodische Randbedingung kann nicht gemischt mit einer anderen Randbedingung auftreten, d. h. ein SBC(4) am Beginn erfordert zwingend ein EBC(4) am Ende.
Die Spline-Sequenz von SBC(4) bis EBC(4) muss im Vorausschaubereich der NC liegen. Ist dies nicht der Fall, wird ein Laufzeitfehler erzeugt.

SBC und EBC können auch innerhalb einer Spline-Sequenz (G06,...,G1) programmiert werden. Damit wird die ursprüngliche Sequenz in zwei Teilsequenzen zerlegt, um gezielt Ecken innerhalb der Spline-Kurve zu erzeugen.

Ohne programmierte Start- und Endbedingung wird die natürliche Randbedingung SBC(2,0,...,0) oder EBC(2,0,...,0) verwendet.

Spline-Parameterlänge: Siehe C¹-stetiger kubischer Spline (siehe [Kap. "Bewegung der Werkzeugorientierung"](#) auf Seite 130).

Bewegung der Werkzeugorientierung als C²-stetiger kubischer Spline: Es gelten alle Voraussetzungen wie für den C¹-stetigen kubischen Spline (siehe [Kap. "Bewegung der Werkzeugorientierung"](#) auf Seite 130).

Die Vektororientierung lässt sich ebenfalls als C²-stetiger kubischer Spline programmieren. Es wird hierbei zwischen zwei Varianten unterschieden, die in MP 800700099[3] festgelegt werden können:

- **Vektororientierung (Variante 1) als gemeinsamer C²-kubischer Spline mit dem Raum-Spline der Linearkoordinaten x, y, z.**
Bei dieser Variante wird ein Spline in den sechs Koordinaten x, y, z, ρ_x , ρ_y , ρ_z mit einer gemeinsamen Parametrisierung verwendet.

In diesem Fall hat die Spline-Vektororientierung folgende Eigenschaften:

- Die $\vec{r}$ -Bewegung ($\vec{r} = (x, y, z)$) ist abhängig von der $\vec{\rho}$ -Bewegung und umgekehrt. Dies bedeutet, dass z. B. in Sätzen, in denen nur $\vec{\rho}$ programmiert ist, eine zusätzliche $\vec{r}$ -Bewegung auftritt (s. "Vektororientierung (Variante 2)" auf Seite 137).
- Der Spline-Parameter $w \in [0, w_e]$ gilt sowohl für $\vec{r}(w)$ als auch für $\vec{\rho}(w)$.

- **Vektororientierung (Variante 2) als eigenständiger C²-kubischer Spline unabhängig vom Raum-Spline der Linearkoordinaten x, y, z.**

Diese Variante erfordert eine hohe Anzahl von Sätzen im Vorausschau-bereich. Die Anzahl der Sätze (MP 706000120[3]) sollte etwa drei Mal so hoch wie bei der Variante 1 sein.

Die Variante 2 hat folgende spezielle Eigenschaften:

- Die $\vec{r}$ -Bewegung ($\vec{r} = (x, y, z)$) ist unabhängig von der $\vec{\rho}$ -Bewegung. Eine $\vec{\rho}$ -Bewegung erfolgt nur in Sätzen mit Startorientierung ungleich Endorientierung (siehe Abb. 9-18 "Vektororientierung - nach Variante 1" auf Seite 139).
- $\vec{\rho}(w)$, $w \in [0, w_e]$ hat immer eine eigene chordale Parametrisierung. $\vec{w_e}$ ist dabei der Winkel zwischen dem Start- und dem Endorientierungsvektor des Satzes. Dieses $\vec{w_e}$ des Spline $\vec{p_i} = (w)$ ist verschieden von w_e des Spline $\vec{r_i}(w)$.
- Spezielle Randbedingungen für den Orientierungsvektor $\vec{\rho}$ können nicht programmiert werden.

Beispiel:

Vektororientierung (Variante 2)

Wird das Programm mit der Einstellung MP 8007 00099[3] = 1 abgearbeitet, ergibt sich der in der nächsten Abbildung gezeigte theta-y- Verlauf.

Vektororientierung und Bahnbewegung als gemeinsamer Spline;

(Variante 1) N0 SplineDef(2203,x,y,z,O)

;Linearkoordinaten u. Orientierung sind Spline-Mitglieder (C²)

N0Coord(1); 5-Achstransformation mit **Vektororientierung**

N0 x0 y0 z0 phi0 theta0

N0 G06 G8

Bahnbewegung

N1 y20

;in N1 beginnt bereits die theta-Orientierungsbewegung des Werkzeugs

N2 y40 theta30

N3 y60 theta0 N4 y80

;in N4 endet erst die theta-Orientierungsbewegung des Werkzeugs

N5 ...

Wird das Programm mit der Einstellung MP 8007 00099[3] = 2 abgearbeitet,

ergibt sich der in der nächsten Abbildung gezeigte theta-y- Verlauf. Eine  - Bewegung ergibt sich hier auch in den Sätzen N1 und N4, in denen der Orientierungsvektor nicht programmiert wurde.

;Vektororientierung als eigenständiger Spline

;(Variante 2)

N0 SplineDef(**2203**,x,y,z,O)

;Linearkoordinaten u. Orientierung sind Spline-Mitglieder (C²)

N0 **Coord(1)**; 5-Achstransformation mit **Vektororientierung**

N0 x0 y0 z0 phi0 theta0

N0 G06 G8

N1 y20

N2 y40 theta30

;in N2 beginnt die theta-Orientierungsbewegung des Werkzeugs

N3 y60 theta0

;in N3 endet die theta-Orientierungsbewegung des Werkzeugs N4 y80

N5 ...

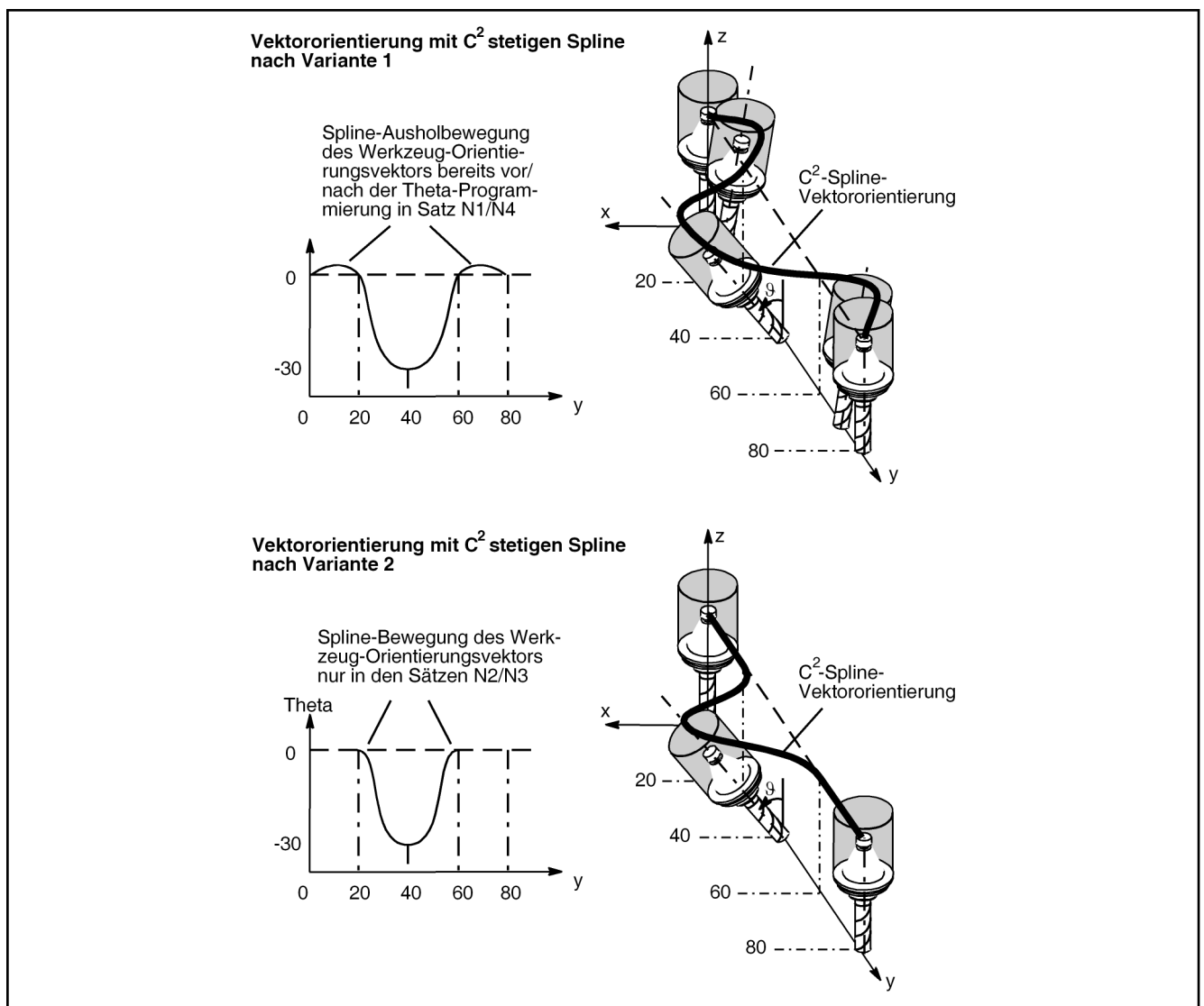


Abb. 9-18: Vektororientierung - nach Variante 1

- **Lineare Orientierungsbewegung als C^2 -stetiger kubischer Spline**
 - Für die lineare Orientierungsbewegung als C^2 -stetiger kubischer Spline gilt das gleiche wie unter der linearen Orientierungsbewegung als C^1 -stetiger kubischer Spline (siehe ab [Kap. 9.1.3 "Spline mit Stützpunktprogrammierung und tangentialen Übergängen \(\$C^1\$ -stetige kubische Splines\)"](#) auf Seite 127).
 - Bei langen Spline-Sequenzen (Anzahl der Sätze zwischen SBC und EBC), die nicht vollständig im Vorausschaubereich (siehe [Kap. "Relevante Maschinenparameter \(MP\)"](#) auf Seite 152) der NC liegen, werden durch ein geeignetes Verfahren zwei oder mehrere Teil-Spline-Sequenzen erzeugt, die wiederum C^2 -stetig ineinander übergehen. Damit können die Splines für beliebig lange Konturen (Freiformflächenbearbeitung) benutzt werden.
- Die Anzahl der Teilsequenzen sollte minimal gehalten werden, d. h. der MP 800700099[2] sollte so groß wie möglich eingestellt werden.

- Bei periodischen Randbedingungen ist eine Sequenzunterbrechung nicht möglich. Hier muss die gesamte Spline-Sequenz im Vorausschaubereich liegen.

9.1.6 Spline mit Kontrollpunktprogrammierung (B-Spline)

Funktion

B-Splines werden durch sogenannte "Kontrollpunkte" erzeugt. Im Gegensatz zu den C^1/C^2 -Splines verläuft die Spline-Kurve bei B-Splines nicht durch vorgegebene Stützpunkte, sondern "nähert" sich den vorgegebenen "Kontrollpunkten" an (siehe folgende Abbildung).

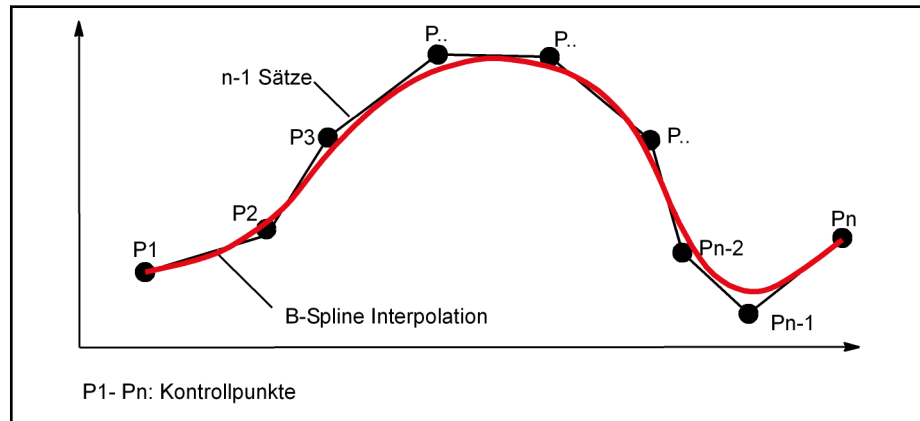


Abb. 9-19: Spline mit Kontrollpunktprogrammierung (B-Spline)



NURBS (NonUniform Rational B-Spline) sind eine spezielle Ausprägung der B-Splines. Sie werden hier unter dem Begriff "B-Spline" behandelt. Für spezielle Eigenschaften der NURBS sei auf die einschlägige Literatur verwiesen.

Die B-Splines haben folgende Eigenschaften:

- Aus n -Punkten, d. h. aus " $n-1$ " NC-Sätzen entstehen " $n-p$ " Splines, wobei " p " den Grad der B-Splines bezeichnet (siehe nachfolgendes Beispiel).
- Im Allgemeinen treten **kleinere** Krümmungen auf, als bei den globalen kubischen Splines, da die B-Splines, außer in Spezialfällen, nicht durch die Kontrollpunkte gehen. Sie neigen weniger zum Überspringen und sind trotzdem am Übergang krümmungsstetig.
- Durch Programmieren von Punktgewichten mit der Funktion PW können die Splines in der Umgebung eines Punktes verändert werden. Ein Punktgewicht >1 "zieht" den Spline zum Punkt hin, ein Punktgewicht <1 "drückt" ihn vom Punkt weg. Man spricht in diesem Fall vom rationalen B-Spline oder NURBS.
- B-Splines wirken lokal, d. h. die Veränderung eines Punktes beeinflusst eine vom Grad des B-Splines abhängige Anzahl Splines.
- Als Start- und Endrandbedingung ist der Typ 10 (Tangente entlang Start-Endpunkt-Verbindungsline, vgl. [Kap. 9.1 "Spline" auf Seite 119](#)) möglich.

Kontrollpunkte: Für eine B-Spline-Kurve vom Grad p müssen mindestens " $p+1$ " verschiedene Kontrollpunkte definiert sein. Innerhalb der Spline-Sequenz müssen mindestens " p " NC-Sätze (mit Endpunkt ungleich Startpunkt) vorhanden sein.

Doppelte Kontrollpunkte werden nicht unterstützt, d. h. sie werden von der NC herausgefiltert. Es entsteht dabei ein Spline ohne Verfahrbewegung.

Spline-Parameterlänge: Siehe C¹-stetiger kubischer Spline (siehe [Kap. "Bewegung der Werkzeugorientierung" auf Seite 130](#)).

B-Splines können äquidistant, chordal und zentripetal parametrisiert werden (siehe [Kap. "Bewegung der Werkzeugorientierung" auf Seite 130](#)). Bei der äquidistanten Parametrisierung spricht man von uniformen B-Splines, in den beiden anderen Fällen von nicht uniformen B-Splines. Darüber hinaus kann der Knotenvektor (siehe [Kap. 9.1.6 "Spline mit Kontrollpunktprogrammierung \(B-Spline\)" auf Seite 140](#)) gezielt durch Programmieren der Spline-Parameterlänge PL verändert werden.

Punktgewichte: Die Kontrollpunkte können mit einem Punktgewicht (**Point Weight**) mit der Syntax **PW ..** versehen werden. Durch unterschiedliche Wichtung kann so an jedem Kontrollpunkt (abhängig vom Grad des Splines auch im Umkreis des Kontrollpunktes) der Verlauf der Spline-Kurve beeinflusst werden. Ein Punktgewicht von 0 ist nicht erlaubt.

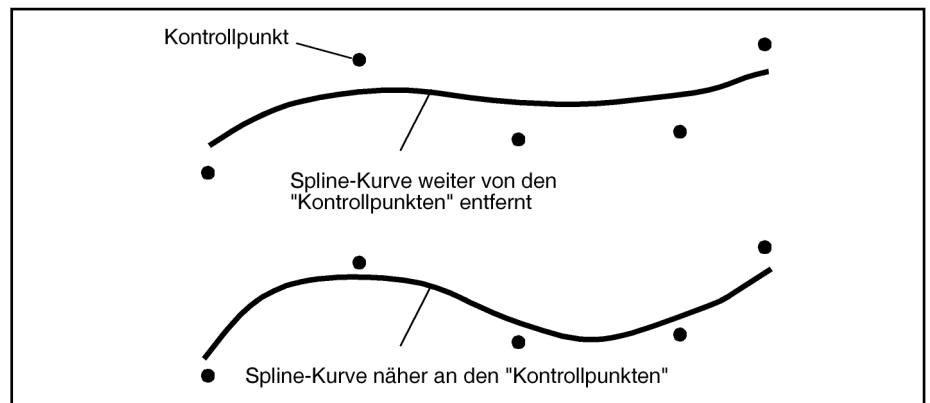


Abb. 9-20: Kontrollpunkte

Beispiel:

B-Spline 3. Grades:

;Auswahl B-Spline 3. Grades:

N0 SplineDef(3213,,x,y,z,O)

;5-Achstransformation mit Vektororientierung:

N0 Coord(1)

N0 x0 y0 z0 phi0 theta0

N0 G06 G108

N1 x10y20z11 O(0.1,0,1.0)

N2 x.. y.. z.. O(..)

;1. Spline:

N3 x.. y.. z.. O(..)

;2. Spline, Punktgewicht 2.3:

N4 x.. y.. z.. O(..) PW2.3

...

... ...

;18. Spline:

N20 x.. y.. z.. O(..)

N21 G1

Erläuterung:

N0	Auswahl eines B-Splines 3. Grades
N1	Startpunkt der Spline-Kurve tangential zur Verbindungslinie Startpunkt - Endpunkt
N1-N2	Nichtverfahrensatz
N1-N20	21 Kontrollpunkte festlegen
N3-N20	18 Splines, die abgearbeitet werden
N20	Endpunkt der Spline-Kurve tangential zur Verbindungslinie Startpunkt - Endpunkt.

Detaillierte B-Spline-Beschreibung (für Entwickler)**Definition und Eigenschaften von B-Spline-Kurven in der NC:**

Im Folgenden wird die Umsetzung einer B-Spline-Sequenz aus einer CAD/CAM-Ausgabedatei in das NC-Programmformat beschrieben. Das Kapitel richtet sich daher speziell an Postprozessor-Entwickler.

Die NC unterstützt B-Splines bis zum Grad $p = 5$. Grad $p = 3$ ist Standard.

Eine B-Spline-Kurve (Nurbs-Kurve) vom Grad p ist definiert durch

- $n+1$ Kontrollpunkte (bei n NC-Sätzen),
- einen Knotenvektor mit $n+p+2$ Elementen
- $n+1$ Punktgewichten
- Knotenspanne $PL = 1$ (default)
- Punktgewicht $PW = 1$ (default)

Der Knotenvektor ist uniform mit der Knotenspanne gleich 1.

Er hat die Form

$$U = \{0, \dots, 0, 1, 2, \dots, n+1-p, \dots, n+1-p\}$$

Der Knotenvektor beginnt mit $p+1$ Nullen und hat am Ende $p+1$ Knoten mit dem Wert $n+1-p$. Die Anzahl der Knotenelemente des Knotenvektors ist $m = n+p+2$.

Die Anzahl der Splines ist $n+1-p$, also gleich der Anzahl der Knotenspannen ungleich Null.

Für einen kubischen uniformen B-Splines $p = 3$ hat der Default-Knotenvektor die allgemeine Form

$$U = \{0, 0, 0, 0, 1, 2, \dots, n-3, n-2, n-2, n-2, n-2\}$$

Die Anzahl der Splines (Anzahl der Nutzknoten) ist $n-2$.

Beispiel:

Im folgenden Beispiel des NC-Programms mit 5-Achsbearbeitung steht $\langle P0 \rangle$, $\langle P1 \rangle$... für einen Syntaxausdruck der Form $x.. y.. z.. O(.., .., ..)$

Die drei Linearkoordinaten x , y und z beschreiben die Position der Werkzeugspitze (TCP). Der Orientierungsvektor des Werkzeugs wird durch $O(..)$ beschrieben (s. a. Vektororientierungsbewegung).

Damit ergibt sich:

$p = 3$: Grad des kubischen B-Splines
 $n = 7$: Anzahl der Spline NC-Sätze
 $n+1 = 8$: Anzahl der Kontrollpunkte
 $n+1-p = 5$: Anzahl der Splines (Nutzknoten)
 $n+p+2 = m$: Anzahl der Knotenelemente des Knotenvektors
 12:

Der Knotenvektor hat die Form $U = \{0,0,0,0,1,2,3,4,5,5,5,5\}$

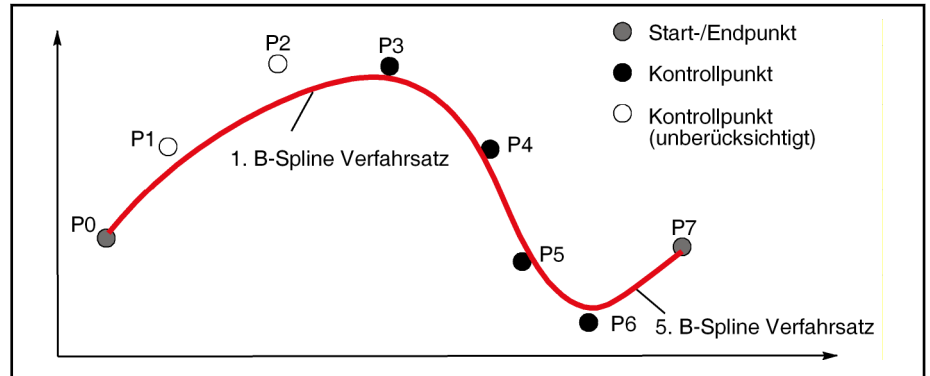


Abb. 9-21: Beispiel

N01 SplineDef(3213,,x,y,z,O)	;Kubischer uniformer B-Spline für die Koordinaten x, y, z und O
N02 COORD(1)	;5-Achstransformation mit Vektororientierung ;(z. B. 3232201)
N10 G1 <P0>	;Anfahren des Kontrollpunkts P0
N11 G06 <P1>	;Spline ein, Kontrollpunkt P1
N12 <P2>	;Kontrollpunkt P2
N13 <P3>	;Kontrollpunkt P3, ;1.Spline-Verfahrenssatz
N14 <P4>	
N15 <P5>	
N16 <P6>	
N17 <P7>	;Kontrollpunkt P7; ;5. Spline-Verfahrenssatz
N18 G1	;Spline aus

Da die Anzahl der Spline-NC-Sätze (N11 bis N17) um zwei größer als die Anzahl der Splines ist, entstehen am Anfang der Spline-Sequenz zwei interne Nichtverfahrensätze (N11 und N12).

Möglichkeiten zur Beeinflussung des B-Splines:

- **Verändern des Knotenvektors**

Mit der Syntax **PL** kann der Knotenvektor verändert werden (siehe Beispiel 1):

- Eine Knotenspanne muss erst ab dem ersten Verfahrenssatz (im folgenden Beispiel ab N13) programmiert werden.

Bahnbewegung

- Eine Programmierung von PL in den Nichtverfahrensätzen N11 und N12 bleibt ohne Wirkung.
- Mehrfachknoten werden mit PL0 erzeugt (siehe Beispiel 2).

Beispiel 1:

Veränderte Knotenspanne mit PL.

```

N01          Spline- ;kubischer B-Spline für x, y, z und O
Def(3213,,x,y,z,O)

N02 COORD(1)    ;5-Achstransformation mit Vektororientierung
                ;(z. B. 3232201)

N10 G1 <P0>     ;Anfahren des Kontrollpunkts P0
N11 G06 <P1>    ;Spline ein, 1. Kontrollpunkt
N12 <P2>        ;2. Kontrollpunkt
N13 <P3> PL2.3  ;1. Knotenspanne ist 2.3
N14 <P4>        ;2. Knotenspanne ist 1.0 (Default)
N15 <P5> PL0.8  ;3. Knotenspanne ist 0.8
N16 <P6> PL1.2  ;4. Knotenspanne ist 1.2
N17 <P7> PL2.0  ;5. Knotenspanne ist 2.0
N18 G1          ;Spline aus

Der Knotenvektor hat dann die Form
U = {0,0,0,0,2.3,3.3,4.1,5.3,7.3,7.3,7.3,7.3}

```

Beispiel 2:

Veränderte Knotenspanne durch Mehrfachknoten mit PL0

```

N01          Spline- ;kubischer B-Spline für die Raumkurve x,y,z
Def(3213,,x,y,z,O)    ;und den Orientierungsvektor p

N02 Coord(1)    ;5-Achstransformation mit Vektororientierung
                ;(z. B. 3232201)

N10 G1 <P0>     ;Anfahren des Kontrollpunkts P0
N11 G06 <P1>    ;Spline ein, Kontrollpunkt P1
N12 <P2>        ;Kontrollpunkt P2
N13 <P3> PL2.3   ;1. Knotenspanne ist 2.3
N14 <P4> PL0    ;2. Knotenspanne ist 0.0
                ;(Mehrfachknoten)
N15 <P5> PL0.8   ;3. Knotenspanne ist 0.8
N16 <P6> PL1.2   ;4. Knotenspanne ist 1.2
N17 <P7> PL2.0   ;5. Knotenspanne ist 2.0
N18 G1          ;Spline aus

Dies ergibt den Knotenvektor
U = {0,0,0,0,2.3,2.3,3.1,4.3,6.3,6.3,6.3,6.3}

```

Tab. 9-1: *Beispiel 2*

Man beachte, dass ein doppelter Knoten die Anzahl der Splines um 1 erniedrigt. Im Beispiel 2 ist demnach die Anzahl der erzeugten Splines gleich 4.

Verändern der Punktgewichte: Jedem Kontrollpunkt (mit Ausnahme <P0>) wird mit der Syntax **PW** ein Gewicht zugeordnet. Default ist PW=1.

Beispiel:

```
N01 SplineDef(3213,,x,y,z,O)
N02 Coord(1)
N10 G1 <P0>           ;Anfahren des Kontrollpunkts P0
N11 G06 <P1> PW1.8    ;P1-Gewicht = 1.8
N12 <P2> PW 2.5        ;P2-Gewicht = 2.5
N13 <P3> PL2.3          ;P3-Gewicht = 1
N14 <P4> PL0            ;P4-Gewicht = 1
N15 <P5> PL0.8          ;P5-Gewicht = 1
N16 <P6> PL1.2 PW0.6  ;P6-Gewicht = 0.6
N17 <P7> PL2.0          ;P7-Gewicht = 1
N18 G1                 ;Spline aus
```

Umsetzbeispiel einer CAD/CAM-Nurbs-Sequenz in das NC-Programmformat der MTX:

Beispiel: APT-Nurbs-Format von Catia Version 5 für fünfschsiges Fräsen.
Für einen Siemens-Postprozessor hat das APT-Format folgende Struktur:

```
BEGIN NURBS_SIEMENS (D=3, F=xxxx, AXIS=VAR, LENGTH=100.00) ;
N0, XT=xt0, YT=yt0, ZT=zt0, XH=xh0, YH=yh0, ZH=zh0, DK=dk0, W=w0;
N1, XT=xt1, YT=yt1, ZT=zt1, XH=xh1, YH=yh1, ZH=zh1, DK=dk1, W=w1;
N2, XT=xt2, YT=yt2, ZT=zt2, XH=xh2, YH=yh2, ZH=zh2, DK=dk2, W=w2;
.../...
Nn, XT=xtn, YT=yt n, ZT=zt n, XH=xhn, YH=yhn, ZH=zhn, DK=dkn, W=wn;
END NURBS;
```

Die Umsetzung der Parameter des **Postprozessor APT-Format** in das MTX-Format kann wie folgt aussehen:

Parameter	Postprozessor APT-Format Elemente	MTX-Programmformat Bosch Rexroth
Header	BEGIN NURBS_SIEMENS(D=3,F=xxxx,AXIS=VAR,LENGTH=100.00);	SplineDef(3213,x,y,z,O) Coord (1)
Grad des Nurbs	D	p
Bahnvorschub	F	F
Achsentyp	AXIS = VAR	z. B. x, y ,z, O
Kontrollpunkt-Position und WZ-Orientierung	Kontrollpunkt-Position = Position der Werkzeugspitze (XTi,YTi,ZTi) und einer Position entlang der Werkzeugsymmetrieachse (XHi,YHi,ZHi) LENGTH = Abstand zwischen XTi,YTi,ZTi und XHi,YHi,ZHi mit i = 0 bis n	x<xti> y<yti> z<zti> O(<xhi-xti>,<yhi-yti>,<zhi-z>) mit i = 0 bis n.
Knotenspanne	dk i mit i = 0 bis n	PL<dk i> mit i = 0 bis n
Punktgewicht	wi mit i = 0 bis n	PW<wi> mit i = 0 bis n

Tab. 9-2: MTX-Format

Die Kontrollpunkt-Position wird im Postprozessor durch die Position der Werkzeugspitze (X_{Ti}, Y_{Ti}, Z_{Ti}) und einer weiteren Position entlang der Werkzeugsymmetrieachse (X_{Hi}, Y_{Hi}, Z_{Hi}) dargestellt.

Auf das MTX-Format umgesetzt, bedeutet die Postprozessor-Kontrollpunkt-Position die MTX-Position des TCP und ein zur TCP-Position gehörender Orientierungsvektor:

$$\langle P_i \rangle = x \langle x_{ti} \rangle \ y \langle y_{ti} \rangle \ z \langle z_{ti} \rangle \ O(\langle x_{hi} - x_{ti} \rangle, \langle y_{hi} - y_{ti} \rangle, \langle z_{hi} - z_{ti} \rangle)$$

Der Orientierungsvektor muss nicht auf "Eins" normiert werden, d. h. LENGTH bleibt unberücksichtigt. Im MTX-Format sieht obiges Postprozessor NC-Programm wie folgt aus:

```
SplineDef (3203, , x, y, z, O) ;B-Spline für x, y, z und O
Coord (1) ;Auswahl 5-Achstransformation mit Vektororientierung
G1 <P0> ;Anfahren des Startpunkts der Nurbs-Sequenz
G06 <P1> PL<dk0> PW<w1> ;Spline ein, PL<dk0> kann weggelassen werden
F..
<P2> PL<dk1> PW<w2> ;PL<dk1> kann weggelassen werden
<P3> PL<dk2> PW<w3>
..
<Pn> PL<dkn-1> PW<wn>
G1 <Pn+1> ;Spline aus, ggf. ist Pn+1 = P0 der nächsten Nurbs-Sequenz
```

9.1.7 Approximierende B-Splines mit Toleranzangabe

Die NC-Programmerstellung (CAD/CAM-Postprozessor-Kette) für die Bearbeitung von Freiformflächen erzeugt in vielen Fällen nicht optimale NC-Programme bezüglich Kurvenglattheit und Punktabständen. Diese Defizite können von der NC nicht ohne weiteres kompensiert werden. Ein wesentlicher Punkt ist die häufig fehlende oder mangelhafte Unterstützung der Spline-Bearbeitung. An Stelle von Splines werden für alle Raumkurven und Orientierungsbewegungen kurze Linearsätze (G1) programmiert, welche die folgende Problematik aufweisen:

1. Die programmierten Kurvenpunkte sind oft ungenau, z. B. werden NC-Satz-Punkte mit nur zwei Nachkommastellen erzeugt.
2. Der Abstand benachbarter Punkte ist so gering, dass dadurch eine Absenkung der Bahngeschwindigkeit seitens der NC erfolgt. Die Bearbeitungszeit wird dadurch unnötig erhöht. Bei schwankenden Punktabständen kann es außerdem zu einem unruhigen Bahngeschwindigkeitsverlauf kommen, wenn Sätze mit und ohne Geschwindigkeitsabsenkung einander abwechseln
3. Die einfache Lösung, pauschal Spline-Interpolation (G6) an Stelle von Linearinterpolation (G1) zu verwenden, scheitert oft daran, dass wiederum einige wenige ausgezeichnete Sätze aus geometrischen Gründen tatsächlich mit G1 verfahren werden müssen

Die Verwendung approximierender B-Splines löst diese Probleme.

Die Funktion hat die folgenden Eigenschaften:

Im NC-Programm vorgegebene Punkte und Orientierungen werden durch eine glatte B-Spline-Kurve mit vorgegebenem Grad 2 bis 5 approximiert. Die Approximation erfolgt so, dass jeder programmierte Punkt von der B-Spline-

Kurve einen Abstand hat, der kleiner als ein programmierbares Toleranzmaß ist. Ein entlang der Kurve gelegter "Schlauch" mit Radius gleich Toleranzmaß erfasst alle programmierten Punkte. Damit werden zum einen ungenau programmierte Punkte geglättet, zum anderen werden je nach geforderter Genauigkeit wenige, lange Spline-Sätze gebildet. Hauptanwendungsgebiet der Funktion ist das Bearbeiten von Freiformflächen.

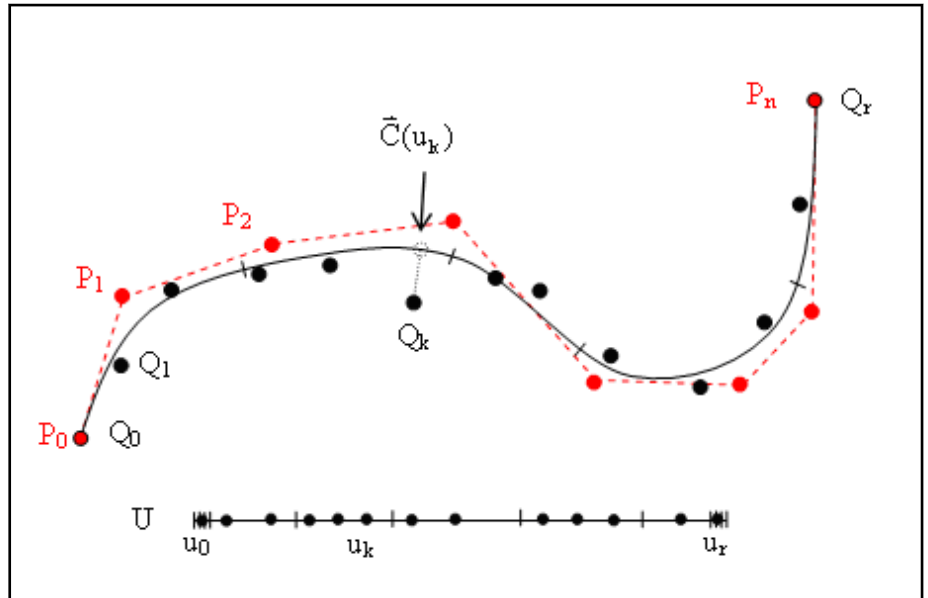


Abb. 9-22: B-Spline-Kurve

Die programmierten Punkte Q_0 bis Q_r werden durch eine B-Spline-Kurve mit berechneten Kontrollpunkten P_0 bis P_n ($n < r$) approximiert.

Das Toleranzmaß kann in den B-Spline-Approximationsparametern angegeben werden.

Diese werden mit der folgenden **Syntax** programmiert:

BsaPar(E...,OE...) Kurzform: **BAP(..)**

Die einzelnen Parameter haben die folgende Bedeutung:

1. **E (Error)**

Gibt das Toleranzmaß in mm oder Zoll an.

Bei fehlendem "E" gilt der Wert im MP 80070030.

2. **OE (OrientationError)**

Toleranzmaß der Orientierungsbewegung in mm oder Zoll.

Ist kein Wirkradiusfaktor für den Orientierungsvektor programmiert, entspricht ein mm einem Grad. "OE" ist nur bei aktiver Vektororientierung von Bedeutung.

Bei fehlendem "OE" gilt als Einschaltwert $OE = E$. Beim Sturzfräsen ist $OE = 5 \cdot E$ ein sinnvoller Wert.

Start- und Endrandbedingungen

Neben den Randbedingungen 1, 10 und 11 (siehe ["Start- und Endbedingung:" auf Seite 129](#)) ist bei der B-Spline-Approximation die Randbedingung 0 möglich:

- **SBC(0):**

Im aktuellen Satz beginnt eine neue Spline-Sequenz (Konturknick am Satzstartpunkt)

- EBC(0):

Im aktuellen Satz endet die aktuelle Spline-Sequenz (Konturknick am Satzendpunkt)

Die B-Spline-Approximation ist ein weiterer Spline-Typ 4 in der Reihe der MTX-Splines (Tab. 9-4 "Relevante NC-Funktionen - Spline mit Koeffizientenprogrammierung" auf Seite 149).

Ein Sonderfall ist die Variante mit der Spline-Id 4233. Hier wird eine besonders gute Glättung der Kurve erzeugt durch ein zusätzliches Geometriefilter. Die Spline-Id 4233 wird für die Freiformflächenbearbeitung empfohlen.

9.1.8 Spline programmieren

Funktion

Alle verwendbaren Splines können im Programm parametrisiert werden mit:

SplineDef(<Id>,<Members>)

Die Tabelle zeigt eine Übersicht der vorhandenen Möglichkeiten:

Spline-Id = <Typ>+<Parametrisierung>+<Untertyp>+<Grad>

Spline-Typ	Parametrisierung 0 = ohne Bedeutung 1 = äquidistant 2 = chordal 3 = zentripetal	Untertyp	Grad	Bedeutung	Empfohlene Kombination
0	0	0 = ohne Bedeutung	1,...,5	Polynom-Koeffizienten- Programmierung	SplineDef(3)
1	1, 2, 3	Tangentenberechnung: 1 = Bessel 2 = Akima 3 = Sehnen 4 = Hermite	3	c1-stetiger lokaler kubischer Spline aus vorgegebenen Punkten mit unterschiedlicher Tangenten-Berechnung oder -Vorgabe	SplineDef (1213,...) bzw. SplineDef (1143)
2	1, 2, 3	0 = ohne Bedeutung	3	c2-stetiger globaler kubischer Spline aus vorgegebenen Punkten	SplineDef(2203,...)
3	1, 2, 3	0 = ohne Bedeutung	1,...,5	B-Spline und Nurbs aus vorgegebenen Kontrollpunkten und Gewichten PW	SplineDef(3203,...)
4	2	0 = ohne Bedeutung	1,...,5	B-Spline-Approximation (BSA)	SplineDef(4203,...) *)
		3	3	BSA mit Glättungsfilter	SplineDef(4233,...)

*) Die Splinetypen 4201, 4202, 4203, 4204 und 4205 werden ab MTX 14VRS nicht mehr unterstützt.

Tab. 9-3: Spline-Id

Restriktionen



Die zum Zyklus G591 gehörenden Splinetypen 4201, 4202, 4203, 4204 und 4205 werden ab MTX 14V22 nicht mehr unterstützt.

Folgende Funktionen können **nicht** mit Splines programmiert werden:

- Tensororientierung: dies betrifft sowohl die Tensorsyntax Ox(..), Oy(..) und Oz(..) als auch die Eulerwinkel phi, theta, psi
- 2D-Bahnkorrektur G41/G42 (3D-Bahnkorrektur G141/G142 ist zu verwenden)
- Fasen und Rundungen (SCO ist zu verwenden)
- Tangentiale Werkzeugführung

Relevante NC-Funktionen

- Spline mit Koeffizientenprogrammierung:

G06 mit den modalen Parametern:	
<CoordName >(..)	Koeffizienten-Programmierung einzelner Koordinaten (Coord-Name) mit ihren Polynom-Koeffizienten.
O1(..), O2(..), O3(..)	Koeffizienten-Programmierung der Vektororientierung (vorher muss Raumkoordinatenprogrammierung(Coord(..)) aktiviert sein)
DN(..)	Nenner-Polynom bei Koeffizienten-Programmierung für die Beschreibung von rationalen Monom-Splines (Bezier-Spline, rationaler B-Splines (NURBS), alle Kegelschnitte).
PL(..)	Spline-Parameterlänge

Tab. 9-4: Relevante NC-Funktionen - Spline mit Koeffizientenprogrammierung
Beispiel:

N10 SplineDef(2) ;Koeffizientenprogrammierung Grad 2

N20 G1 X0 Y0

N30 G06 X(0,1) Y(0,0,1) PL3.0 ; Parabel $X=w$, $Y=w^2$, $0 \leq w \leq 3$

- C¹-stetige kubische Splines:

G06 mit den modalen Parametern:	
<CoordName>	Programmierung der Koordinaten; werden nur für die Berechnung des Splines einbezogen, wenn sie unter dem Parameter <Members> in der NC-Funktion SplineDEF(..) aufgelistet sind. Alle anderen werden linear mitgeführt.
<Achsen>	Programmierung der Achsen; siehe oben <CoordName>
<Orientierungskordinaten>	Programmierung der Vektororientierung (5-Achstransformation) als Spline
SBC(..)	Startrandbedingungen (Typ = 1, 10, 11) für den Startpunkt einer Spline-Sequenz
EBC(..)	Endrandbedingungen (Typ = 1, 10, 11 für den Endpunkt einer Spline-Sequenz
PL(..)	Spline-Parameterlänge

Tab. 9-5: Relevante NC-Funktionen - c¹-stetige kubische Splines
Beispiel:

N10 SplineDef(1213) ;c¹-stetiger Spline 3. Grades

N20 G1 X0 Y0

N30 G06 X1 Y1

N40 X2 Y4

..

N50 G1

- Hermite Splines

G06 mit den modalen Parametern:	
<CoordName>	Koeffizienten-Programmierung einzelner Koordinaten (Coord-Name) mit ihrem Positions- und Ableitungswert
PL(..)	Spline-Parameterlänge

Tab. 9-6: Relevante NC-Funktionen - Hermite-Splines

Beispiel:

N10 SplineDef(1143) ;Hermite-Spline

N20 G1 X0 Y0 Z0 F1000

N30 G06 X(20,1) Y(10,-1) PL3.0

N40 X(30.5,2.1) Y(10.8,-0.8) Z(5,1.3) PL4.2

...

N1000 G1

- C²-stetige kubische Splines:

G06 mit den modalen Parametern:	
<CoordName>	Programmierung der Koordinaten; werden nur für die Berechnung des Splines einbezogen, wenn sie unter dem Parameter <Members> in der NC-Funktion Splinedef(..) aufgelistet sind. Alle anderen werden linear mitgeführt.
<Achsen>	Programmierung der Achsen; siehe oben <CoordName>
<Orientierungskordinaten>	Programmierung der Vektororientierung (5-Achstransformation) als Spline
SBC(..)	Startrandbedingungen (Typ = 1,2,3,4,10,11)
EBC(..)	Endrandbedingungen (Typ = 1,2,3,4,10,11)
PL(..)	Spline-Parameterlänge

Tab. 9-7: Relevante NC-Funktionen - C²-stetige kubische Splines

Beispiel:

N10 SplineDef(2203) ;c²-stetiger Spline 3. Grades

N20 G1 X0 Y0

N30 G06 X1 Y1

N40 X2 Y4

..

N50 G1

- B-Splines (NURBS):

G06 mit den modalen Parametern:	
<CoordName>	Programmierung der Koordinaten; werden nur für die Berechnung des Splines einbezogen, wenn sie unter dem Parameter <Members> in der NC-Funktion Splinedef(..) aufgelistet sind. Alle anderen werden linear mitgeführt.
<Achsen>	Programmierung der Achsen; siehe oben <CoordName>
<Orientierungskordinaten>	Programmierung der Vektororientierung (5-Achstransformation) als Spline
SBC(..)	Startrandbedingungen (Typ = 10)
EBC(..)	Endrandbedingungen (Typ = 10)

Tab. 9-8: Relevante NC-Funktionen

- SplineDef(<Id>, <Members>):
Definition des Spline-Typ

Es gilt:

<Id>	Spline-Typ, (siehe Tab. 9-3 "Spline-Id" auf Seite 148)
<Members>	Legt die Koordinaten- oder Achsnamen fest, die an der Spline-Bewegung teilnehmen. Nicht aufgeführte Koordinaten/Achsen können nur linear bewegt werden. Wird Members nicht programmiert, werden alle Koordinaten/Achsen des Kanals für die Spline-Bewegung benutzt. Die Angabe von <Members> entfällt für den Typ 0.

Tab. 9-9: Relevante NC-Funktionen - SplineDef(<Id>, <Members>)

Beispiel:

N10 SplineDef(3203) ;B-Spline 3. Grades

N20 G1 X0 Y0

N30 G06 X1 Y1

N40 X2 Y4

..

N50 G1

B-Spline-Approximation

G06 mit den modalen Parametern	
<CoordName>	Programmierung der Koordinaten. Es dürfen nur Koordinaten aus der Liste <Members> benutzt werden.
<Orientierungskordinaten>	Programmierung der Vektororientierung O, falls in <Members> enthalten
SBC(..)	Startrandbedingung (Typ = 0)
EBC(..)	Endrandbedingung (Typ = 0)
BAP(..)	B-Spline-Approximationsparameter

Tab. 9-10: Relevante NC-Funktionen - B-Spline-Approximation

Bahnbewegung

Beispiel:

N10 SplineDef(4233) BAP(E0.02,OE0.1) ;B-Spline-Approximation

N20 G1 X0 Y0 O(1,1,1)

N30 G06 X1 Y1 O(1.2,0.8,1)

N40 X2 Y4 O(1,3,0.7,0.95)

..

N50 G1

Einen Überblick über die Programmiermöglichkeiten der einzelnen Spline-Typen gibt die folgende Tabelle:

Spline-Typ	Endpunktprogrammierung	PL	PW	SBC/EBC
0	Koeffizientenprogrammierung, Nennerpolynom für rationale Splines	muss	nein	nein
1	Kanalkoordinaten als Stützpunkte	optional	nein	Typ = 1, 2, 10, 11
2	Kanalkoordinaten als Stützpunkte	optional	nein	Typ = 1, 2, 3, 4, 10, 11
3	Kanalkoordinaten als Kontrollpunkte	optional	ja	Typ = 10
4	Kanalkoordinaten (nur Members) als zu approximierende Punkte	nein	nein	Typ = 0

Tab. 9-11: Programmiermöglichkeiten der Spline-Typen

Relevante Maschinenparameter (MP)

7060 00120[3]	Maximale Satzanzahl für Satzvorausschau (Look-Ahead) Empfohlene Werte: Für Spline-Typ 2 : 30 Sätze Für Spline-Typ 4 : 80 Sätze Für alle anderen Spline-Typen : 10 Sätze
7060 00130[3]	Prozentuale Aufteilung Vorausschau (Look Ahead)
8007 00010	Toleranzwert ϵ_0 in Mikrometer für die C ⁰ -Stetigkeit bei Splines mit Koeffizientenprogrammierung. Stimmt der Startpunkt nicht mit dem Endpunkt des Vorgänger-Splines überein, wird bei einer Abweichung, die größer als der eingestellte Toleranzwert ϵ_0 ist, ein Laufzeitfehler erzeugt. Der Wert sollte so klein wie möglich (hängt von der Genauigkeit des Programmiersystems ab) gewählt werden, da ansonsten Achssprünge an den Segmentübergängen auftreten können.

8007 00020	max. zulässiger relative Bahngeschwindigkeitsfehler δv in %. Das Verhältnis (inverse Bahnlängenfunktion) von der "Spline-Parameterlänge w" und der "Bahnlänge der Kurve" wird NC-intern durch Splines angenähert. Der Bahngeschwindigkeitsfehler gibt die Genauigkeit dieser Annäherung wieder. Empfohlener Wert ist 0,1 %. Wird der Maschinenparameter auf 100% gesetzt, wird die Bahnlängenfunktion w(s) durch einen einzigen quintischen Spline approximiert.	
8007 00099	Spline-Konfiguration (nur für C ² -stetigen kubischen Spline relevant)	
	[1] Grad des Splines zwischen zwei Teilsequenzen:	
	3:	kubischer Spline (nur am Startpunkt C ¹ -stetig)
	4:	quadratischer Spline (C ² -stetig zum Vorgänger-Spline)
	[2] Überlappung (Anzahl NC-Sätze) zweier Teilsequenzen	
	[3] Spline-Variante für Vektororientierung:	
	1:	Spline für Vektororientierung gemeinsam mit Raum-Spline
800700030	2:	Eigenständiger Spline für Vektororientierung
	[4]...[8] nicht relevant	
	Toleranzmaß in mm bei der B-Spline-Approximation	

Tab. 9-12: Relevante Maschinenparameter

9.1.9 Handlungsanweisung: Spline applizieren

Für Spline-Kurven sind die folgenden Maschinenparameter relevant:

IW-Engineering/Configuration: Parameter editieren

- **Spline** (7060 00120) "Max. Satzanzahl für Spline-Funktionen"
Empfohlen werden 60 NC-Sätze
- **Spline** (7060 00130) "Prozentuale Satzanzahl für Spline-Funktionen"
Empfohlen wird eine Prozentzahl, die 60 NC-Sätze ergibt
- **MaxPosDev** (8007 00010)
"Max. zulässige Start-/Endpunktabweichung"
Nur für Spline-Typ 0 relevant
- **MaxVelDev** (8007 00020)
"Max. zulässiger relativer Bahngeschwindigkeitsfehler"
Empfohlen wird 0,1 %
- **BSplineApproxTol** (8007 00030)
"Toleranzmaß in mm bei der B-Spline-Approximation"
Nur für Spline-Typ 4 relevant
Empfohlen wird 0,02 mm
- **ConSplineType** (8007 00099)
"Grad des Verbindungs-Splines"
Nur für Spline-Typ 2 relevant

- **NofOverlapBI** (8007 00099)
"Überlappung zweier Teilsequenzen"
Nur für Spline-Typ 2 relevant
- **VectOri** (8007 00099)
"Spline-Variante für Vektororientierung"
Nur für Spline-Typ 2 relevant

Die Parameter **MaxPosDev** bis **VectOri** müssen über den Parameter **spl** "Splines" im Setup (SUP) aktiviert werden.

9.1.10 Handlungsanweisung: Spline aktivieren

IW-Operation / NC-Programmierung: Initialisieren

Vor der erstmaligen Benutzung von G06 im NC-Programm ist die Initialisierung **SplineDef** zu programmieren, sofern dies nicht bereits im Maschinenparameter-Initstring enthalten ist

SplineDef(<Id>,{<Members>})

- **<Id>**:
maximal 4-stellige ganze Zahl für die Spline-Variante:
- **<Members>**:
Liste mit Namen der Koordinaten/Achsen, die an der Spline-Bewegung teilnehmen sollen. In <Members> nicht programmierte Koordinaten/Achsen werden linear mitgeführt. Orientierungskordinaten lassen sich auch als Spline programmieren. Hierzu wird unter <Members> „O“ für die Orientierung, oder alternativ die Polarkordinaten "phi" und "theta" angegeben. Members ist nicht relevant für Spline-Typ 0.
Members kann auch weggelassen werden. Es gelten dann die folgenden Default-Einstellungen:
5-Achstransformation ein: Members = Koordinaten der Transformation (x,y,z,phi,theta)
Keine 5-Achstransformation: Members = Alle Kanalkoordinaten/-achsen

Beispiel: SplineDef(3203,X,Y,Z)

	Documentation	
Documentation:	MTX Maschinenparameter	Spline initialisieren

IW-Operation / NC-Programmierung: Spline aktivieren

G06 programmieren. Je nach erforderlicher Programmierart lässt sich **G06** für die unterschiedlichen Spline-Typen mit unterschiedlichen, modal wirksamen Parametern programmieren.

	Documentation	
Documentation:	MTX Programmierhandbuch	Spline aktivieren

9.1.11 Handlungsanweisung: Spline deaktivieren

IW-Operation / NC-Programmierung: Spline deaktivieren

Der Bahntyp "Spline" wird deaktiviert durch das Programmieren eines anderen Bahntyps, also z.B. durch **G1**, **G2**, **G3** etc. Die entsprechende Funktion muss appliziert sein.

	Documentation	
Documentation:	MTX Programmierhandbuch	Spline deaktivieren

9.2 Spline-Zusatzfunktionen

9.2.1 Automatische Ecken- und Geradenerkennung

Funktion

Hintergrund In vielen Fällen erzeugen CAD/CAM-Postprozessoren für die Bearbeitung von Freiformflächen NC-Programme, die ausschließlich Linearsätze (G1) enthalten. Eine hohe Oberflächengüte zusammen mit einer kurzen Bearbeitungszeit ist jedoch nur auf glatten Kurven, also z. B. mit Splines zu erreichen. Die einfache Lösung, pauschal Spline-Interpolation (G6) an Stelle von Linearinterpolation (G1) zu verwenden, scheitert oft daran, dass wiederum einige wenige ausgezeichnete Sätze aus geometrischen Gründen tatsächlich mit G1 verfahren werden müssen. Die NC-Funktion "Automatische Ecken- und Geradenerkennung" löst dieses Problem.

Per NC-Befehl können Winkel- und Abstandsbedingungen vorgegeben werden. Damit erzeugt die Spline-Generierung automatisch Ecken und Geraden innerhalb der sonst glatten Spline-Sequenz. Die Funktion ist nicht möglich beim Spline-Typ 0 (Koeffizientenprogrammierung) und wird nicht empfohlen beim Spline-Typ 4 (B-Spline-Approximation).

Winkelbedingung: Absolute Winkelbedingung

Vorgegeben wird ein maximal zulässiger Konturknickwinkel $\alpha^{\max}$.

Überschreitet der Knickwinkel α_i am Endpunkt $\vec{P}_i$ des aktuellen Satzes, ist die Winkelbedingung erfüllt. Bei erfüllter Winkelbedingung wird die Spline-Sequenz unterbrochen. Sie verhält sich damit so, als ob eine Endrandbedingung EBC im Satz i und eine entsprechende Startrandbedingung SBC im Satz i+1 programmiert worden wäre. Dies ergibt am Endpunkt des aktuellen Satzes einen Konturknick (siehe [Abb. 9-23 "Spline Verhalten bei erfüllter Winkel- und Abstandsbedingung"](#) auf Seite 156).

Relative Winkelbedingung

Vorgegeben wird ein Winkelfaktor f_α . Der mittlere Knickwinkel im Vorausschaubereich sei $\bar{\alpha}$. Die Winkelbedingung an einem Punkt $\vec{P}_i$ lautet dann $\alpha_i > f_\alpha \cdot \bar{\alpha}$.

Bei der Berechnung des mittleren Knickwinkels $\bar{\alpha}$ werden die α_i mit erfüllter Winkelbedingung nicht mitgezählt.

Die absolute und die relative Winkelbedingung sind durch ein logisches "Oder" verknüpft.

Abstandsbedingung: Absolute Abstandsbedingung

Vorgegeben wird ein maximal zulässiger Punktabstand $d^{\max}$. Die Abstandsbedingung in einem Satz j mit Startpunkt $\vec{P}_{j-1}$ und Endpunkt $\vec{P}_j$ ist erfüllt, wenn der Punktabstand $d_j = |\vec{P}_j - \vec{P}_{j-1}|$ größer als $d^{\max}$ ist.

Bei erfüllter Abstandsbedingung werden die beiden Punkte mit einem Spline ersten Grades (Linearsatz) verbunden.

Relative Abstandsbedingung

Vorgegeben wird ein Abstands faktor f_d . Der mittlere Punktabstand im Vorausschaubereich sei $\bar{d}$. Die Abstandsbedingung für einen Satz j lautet dann $d_j > f_d \cdot \bar{d}$.

Bei der Berechnung des mittleren Punktabstands $\bar{d}$ werden die d_j mit erfüllter Abstandsbedingung nicht mitgezählt.

Die absolute und die relative Abstandsbedingung sind durch ein logisches "Oder" verknüpft

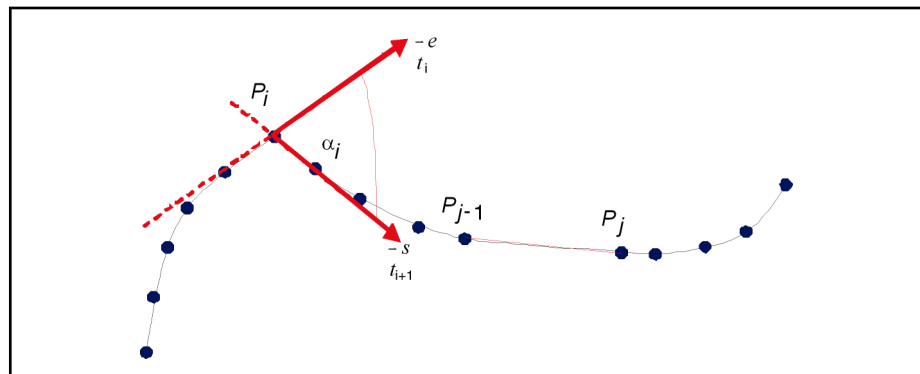


Abb. 9-23: Spline Verhalten bei erfüllter Winkel- und Abstandsbedingung

Syntax. Die Automatische Linear-Spline-Umschaltung wird eingeschaltet mittels der Syntax CornerLineDetection (Kurzform CLD)

CLD(ANG<a>,AFACT<fa>,DIST<d>,DFACT<fd>)

Die Parameter haben die Bedeutung:

- ANG<a>** **a** ist der maximal zulässige Konturknickwinkel $\alpha^{\max}$.
Bei Knickwinkeln größer a ist die Winkelbedingung erfüllt.
- AFACT<fa>** **fa** ist der Winkelfaktor. Übersteigt der Knickwinkel den durchschnittlichen Knickwinkel um den Faktor fa , ist die Winkelbedingung erfüllt.
- DIST<d>** **d** ist der maximal zulässige Punktabstand $d^{\max}$.
Bei Abständen größer als d ist die Abstandsbedingung erfüllt.
- DFACT<fd>** **fd** ist der Abstands faktor.
Übersteigt der Punktabstand den durchschnittlichen Punktabstand um den Faktor fd , ist die Abstandsbedingung erfüllt.

Die Parameter können einzeln weggelassen werden.

Nach dem Programmieren von **CLD** mit mindestens einem Parameter ist die Funktion modal eingeschaltet und wirkt, sobald G06 aktiv ist (Ausnahme: Spline-Koeffizientenprogrammierung).

Die Parameter können jederzeit durch erneutes Programmieren verändert werden.

Die Funktion wird abgeschaltet durch Programmieren ohne Parameter **CLD()**.

Beispiel:

NC-Programm-Beispiel

c²-stetiger kubischer Spline

```
SPD(2203,x,y,z,O) CLD(ANG20,DIST5)
```

```
...
```

```
G06 x.. y.. z.. phi.. theta..
```

```
;Spline 2203 ein
```

absolute Abstandsbedingung, absolute und relative Winkelbedingung sind aktiv

```
x.. y.. z.. O(..)
```

```
...
```

```
x.. y.. z.. O() CLD(DIST8)
```

Ausschalten der Winkelbedingung, neue absolute Abstandsbedingung.

```
...
```

```
G01
```

```
;Spline aus
```

```
...
```

```
G06
```

```
;Spline ein, es gilt das letzte CLD(DIST8)
```

9.3 Automatische Linear-Zirkular-Spline-Wandlung

9.3.1 Beschreibung

Funktion

Die Funktion wandelt eine mit Geraden und Kreisen programmierte Kontur in eine Abfolge von Splines um.

Jede mit G1 (Geradeninterpolation mit Vorschub) und mit G2/G3/G5 (Kreis-/ Helical-Interpolation) vorgebende Bewegung wird in einen Punktezug zerlegt. Die Anzahl der Punkte ist dabei von der vorgegebenen Toleranz (BAP(E<Tol>) abhängig. Die so erzeugten Punkte werden anschließend durch Splines verbunden.

Es werden sowohl programmierte, als auch MTX-intern erzeugte Geraden und Kreise (Rundungen, Fasen, eingefügte Sätze der Fräserbahnkorrektur) gewandelt.

Bei der Umwandlung wird der Spline-Typ 4233 (B-Spline-Approximation mit Filter) verwendet.

Die Abfolge ist:

1. Programmierte Geraden und Kreise einlesen.
2. Konturveränderung durch aktive Fasen- oder Rundungsfunktion.
3. Konturanpassung durch aktive 2D-Fräserbahnkorrektur.
4. Wandlung der resultierenden Kontur in eine Abfolge von Splines.

Beispiel *Programm:*

```
...
N10 T1 ;select rough tool
N15 G0 G90 G54 X50 Y50 Z100 S12000 M3 ;go to start
N20 LCTS BAP(E0.1) ;activate conversion with rough tolerance
N25 P part.npg ;call machining subroutine
N30 LCTS() ;deactivate conversion
N35 T2 ;select finish tool
N40 G0 G90 G54 X50 Y50 Z50 S20000 M3 ;go to start
N45 LCTS BAP(E0.01) ;activate conversion with finish tolerance
N50 P part.npg ;call machining subroutine
N55 LCTS() ;deactivate conversion
...
```

Besonderheiten

Es gelten die Besonderheiten der verwendeten Spline-Variante mit der Spline-Id 4233 "Approximierende B-Splines mit Toleranzangabe und Geometriefilter".

Lokale bewegungserzeugende Funktionen sind von der Wandlung ausgenommen. Dies sind:

- G63, G63.2 "Gewindebohren ohne Ausgleichsfutter"
- G74 "Referenzpunktkoordinaten anfahren"
- G74(HOME) "Referenzpunkt anfahren"
- G75, G75.2 "Fahren gegen Messtaster"
- G76 "Maschinenfeste Achsposition anfahren"
- G77 "Repositionieren einzelner Koordinaten"
- AXC / AxCouple "Achskopplung" (Anfahrbewegung der Kopplung)
- FME / FlyMeas "Fliegendes Messen"
- FSM / FsMove "Fahren auf Festanschlag"
- FSP / FsProbe "Messen an Festanschlag"
- FSR / FsReset "Aufheben Festanschlag"
- PTP / PtpMove "PTP-Verfahrbewegung"
- TCM / TcsMove "Verfahren im TCS"

Durch bewegungsbegleitende Funktionen ergänzte modale Bewegungen sind von der Wandlung ausgenommen. Die Ergänzungsfunktionen sind:

- MOB "Messen auf einem Satz"
- HSB / HsBlkSwitch "Fliegender Satzwechsel"
- PHS / PosDepHsOut "Positionsabhängiger HighSpeed-Ausgang"

Bewegungen mit einer Genauhalt-Funktion sind von der Wandlung ausgenommen:

- G1(IPS1), G1(IPS2)
- G1, G2, G3 bei aktivem G61

Restriktionen

Ein NC-Satz, der zusätzlich zu einer Bahnbewegung eine der nachfolgend aufgeführten NC-Funktionen enthält, wird nicht in einen Spline gewandelt.

- Hilfsfunktion M, S, T, ...
- Eine asynchrone Achsbewegung
- Eine Antriebskommunikation wie z. B. FFW, IPS1, IPS2, KVP, WID3
- Synchronisationsfunktionen zur Satzausführungszeit
 - Bitereignisse (Events) REV, SEV, WEV, WREV
 - Bitinterface WAITA, WAITO
 - Permanente Variablen SPV, WPV, WPVE
Hinweis: Der Befehl "SPVE" kann oft als Ersatz für SPV verwendet werden.
 - Systemdaten SSD
Hinweis: Der Befehl "SSDE" kann oft als Ersatz für SSD verwendet werden.
 - Systemdatenqueue SSDQ, SSDQInit

- Achsübergabe DAX, GAX, RAX, WAX
- Alle eine Kanal- oder Systemspindel betreffende Vorgaben

Relevante NC-Funktionen

- **LCTS**
LCTS oder LCTS(1) schaltet die automatische Linear-Zirkular-Spline-Wandlung ein.
LCTS() oder LCTS(0) schaltet die automatische Linear-Zirkular-Spline-Wandlung aus.
- **BAP(E..)**
Der Parameter hat folgende Bedeutung:
"E" gibt das Toleranzmaß in mm oder Zoll an.
Bei fehlendem "E" gilt der Wert im Maschinenparameter BSplineApprox-Tol (MP 80070030).

Relevante CPL-Funktionen

- **SD(60,11)**
SD(60,11) liefert die von LCTS verwendete Spline-Id.
- **SDR(60,2)**
SDR(60,2) liefert das mit BAP(E..) vorgegebene Toleranzmaß der B-Spline-Approximation in mm oder Zoll.
- **SDR(60,3)**
SDR(60,3) liefert das mit BAP(OE..) vorgegebene Toleranzmaß der B-Spline-Approximation in mm oder Zoll oder -1.0.

Relevante Maschinenparameter

Siehe [Kap. 9.3.2 "Applizieren"](#) auf Seite 159.

9.3.2 Applizieren

Der Speicherplatzbedarf der Funktion "Automatische Linear-Zirkular-Spline-Wandlung" ist sehr groß. Die Datei startup.xml ist entsprechend anzupassen. Ein geeigneter Eintrag ist:

```
<AdditionalNCBlockMem>4000000</AdditionalNCBlockMem>
```

Von der B-Spline-Approximation verwendete Maschinenparameter müssen auf einen sinnvollen Wert eingestellt sein. Dies sind:

- **NofBlkPrep (7060 00110)** "Gesamtanzahl Sätze für Satzvorbereitung und Interpolation"
Ein genügend großer Wert, der den Bedarf aller Funktionen (MaxBlkVelCalc, MaxBlkBuffered, MaxBlkSpline, MaxBlkTCorr3D und MaxBlkMiscFunc) abdeckt.
- **MaxBlkSpline (7060 00120)** "Max. Satzanzahl für Spline-Funktionen"
Empfohlen werden 120 Sätze.
- **PercSpline (7060 00130)** "Prozentuale Satzanzahl für Spline-Funktionen"
Empfohlen wird eine Prozentzahl, die 120 NC-Sätze ergibt.
- **MaxVelDevSpline (8007 00020)** "Max. zulässiger relativer Bahngeschwindigkeitsfehler"
Empfohlen wird 0,1 %.

- BSplineApproxTol (8007 00030) "Toleranzmaß in mm bei der B-Spline-Approximation"
Empfohlen wird 0,02 mm.

9.3.3 Aktivieren

"LCTS" aktiviert die Funktion.

Bei Bedarf kann gleichzeitig durch Programmieren von "BAP(E<Tol>)" der Toleranzschlauch angepasst werden.

9.3.4 Deaktivieren

LCTS() deaktiviert die Funktion.

Zeitweises Deaktivieren

LCTS kann innerhalb eines Zyklus oder eines Unterprogramms durch Verwendung der oben beschriebenen CPL- und NC-Funktionen zeitweise abgeschaltet werden.

Programm:

```
;Save parameters and deactivate LCTS
0001 DIM NCFTS$(25)
0002 NCFTS$ = NCF("LCTS")
0003 IF NCFTS$ = "LCTS" THEN
0004     TSETOL! = SDR(60,2)
0005     TSOTOL! = SDR(60,3)
0006     LCTS()
0007 ENDIF
...
...
...
;Activate LCTS again
0990 IF NCFTS$ = "LCTS" THEN
0991     LCTS BAP(E[TSETOL!],OE[TSOTOL!])
0992 ENDIF
```

10 Werkzeugorientierung und Orientierungsbewegung

10.1 Allgemeines

10.1.1 Beschreibung

Von Orientierungsbewegungen spricht man, wenn das Werkzeug interpolatorisch bezüglich des Werkstückkoordinatensystems (WCS oder PCS) ausgerichtet werden soll. Dabei ist eine entsprechende Achstransformation (5-/6-Achstransformation, Hexapod), welche die Orientierungsbewegung in Achsbewegungen umsetzen kann, notwendig.

10.2 Definitionen zur Orientierung

10.2.1 Werkzeugkoordinatensystem TCS

Das Werkzeugkoordinatensystem (TCS: Tool Coordinate System) ist ein mit dem Werkzeug fest verbundenes kartesisches Koordinatensystem mit den Koordinaten X_t , Y_t und Z_t . Sein Ursprung liegt im TCP (Tool Center Point). Seine Z_t -Koordinate zeigt immer in Richtung der Werkzeugaufnahme. Bei rotationssymmetrischen Werkzeugen, wie Fräser, Laserstrahl, etc., sind die Koordinaten X_t und Y_t ohne Bedeutung. Bei nicht rotationssymmetrischen Werkzeugen, wie z. B. einem Greiferwerkzeug zeigen X_t und Y_t in spezifische Symmetrierichtungen senkrecht zueinander und zu Z_t . Entscheidend ist hierbei die Referenz-Pose der zugehörigen Achstransformation. In der Referenzpose müssen das TCS und das Basiswerkstückkoordinatensystem BCS übereinstimmen.

Beispiel:

Orientierung des TCS

- Rotationssymmetrisches Werkzeug in einer 5-Achskinematik, siehe Kapitel 11 Achs-Transformation → 5-Achs-Transformation - Typ 1: LLLRR: Fräswerkzeug mit drei Positionskoordinaten (x , y , z) und zwei Orientierungskoordinaten (ϕ , θ).
- Nichtrotationssymmetrisches Werkzeug in einer 6-Achskinematik, siehe Kapitel 11 Achs-Transformation → 6-Achs-Transformation: Greiferwerkzeug mit drei Positionskoordinaten (x , y , z) und drei Orientierungskoordinaten (ϕ , θ , ψ).

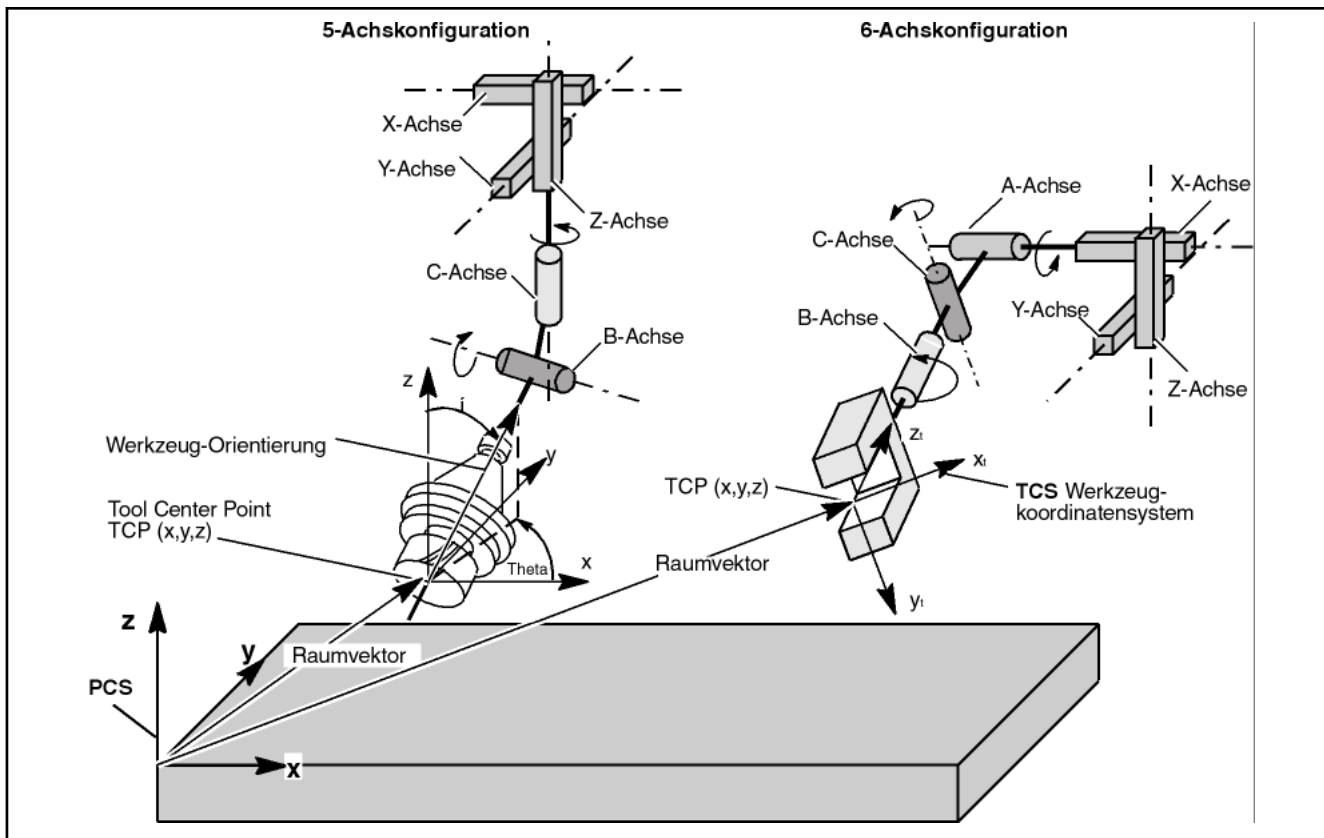


Abb. 10-1: Orientierung des TCS bezüglich des PCS in einer 5- und 6-Achsenkinematik

10.2.2 Eulerwinkel

Allgemeines

Die Orientierung des TCS bezüglich des WCS wird im Allgemeinen durch drei Eulerwinkel ϕ , θ und ψ beschrieben. Bei Rotationssymmetrie entfällt der dritte Winkel ψ . Um die Eulerwinkel eines TCS mit vorgegebener Orientierung zu bestimmen, wird ausgehend von einem TCS(0), welches mit dem WCS übereinstimmt, durch drei hintereinander ausgeführte Basisdrehungen um die Eulerwinkel das TCS(3) (Ziel-TCS) erreicht. Hierbei gibt es drei gleichwertige Varianten mit demselben (ϕ , θ , ψ)-Satz, um das Ziel-TCS zu erreichen.

Varianten zur Eulerwinkelorientierung mit den Winkeln ϕ , θ und ψ

Von den **drei Varianten** ist Variante 1 die verbreitetste Variante.

Variante 1: Drehung um die jeweils neue Koordinate

1. $D_{Z_t}(\phi)$: Drehung um die Z_t -Koordinate des TCS(0) = WCS mit Winkel ϕ
2. $D_{Y_t}(\theta)$: Drehung um die Y_t -Koordinate des TCS(1) mit Winkel θ
3. $D_{Z_t}(\psi)$: Drehung um die Z_t -Koordinate des TCS(2) mit Winkel ψ

Beispiel:

Greiferwerkzeug-Orientierung

Zur besseren Übersicht ist das Werkzeugkoordinatensystem auf den Nordpol des Programmkoordinatensystems verschoben.

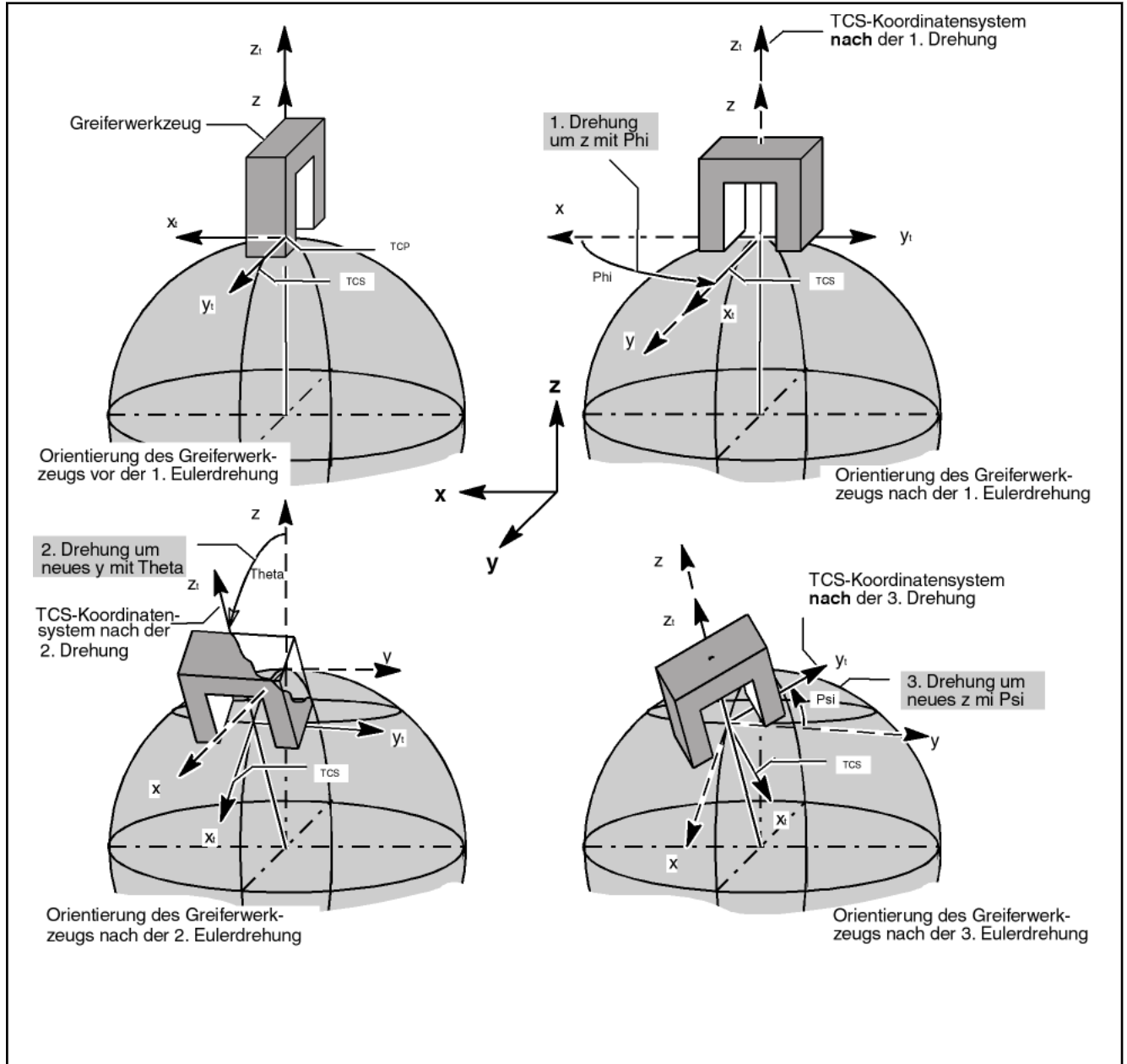


Abb. 10-2: Drehung um die jeweils neue Koordinate des Greiferwerkzeug-TCS: $D_{z_i}(\phi)$, $D_{y_i}(\theta)$ und $D_{z_i}(\psi)$

Variante 2: Drehung um raumfeste Koordinaten

1. $D_z(\psi)$: Drehung um die z -Koordinate des festen WCS mit Winkel ψ
2. $D_y(\theta)$: Drehung um die y -Koordinate des festen WCS mit Winkel θ
3. $D_z(\phi)$: Drehung um die z -Koordinate des festen WCS mit Winkel ϕ

Werkzeugorientierung und Orientierungsbewegung

Beispiel:

Greiferwerkzeug-Orientierung

Zur besseren Übersicht ist das Werkzeugkoordinatensystem auf den Nordpol des Programmkoordinatensystems verschoben.

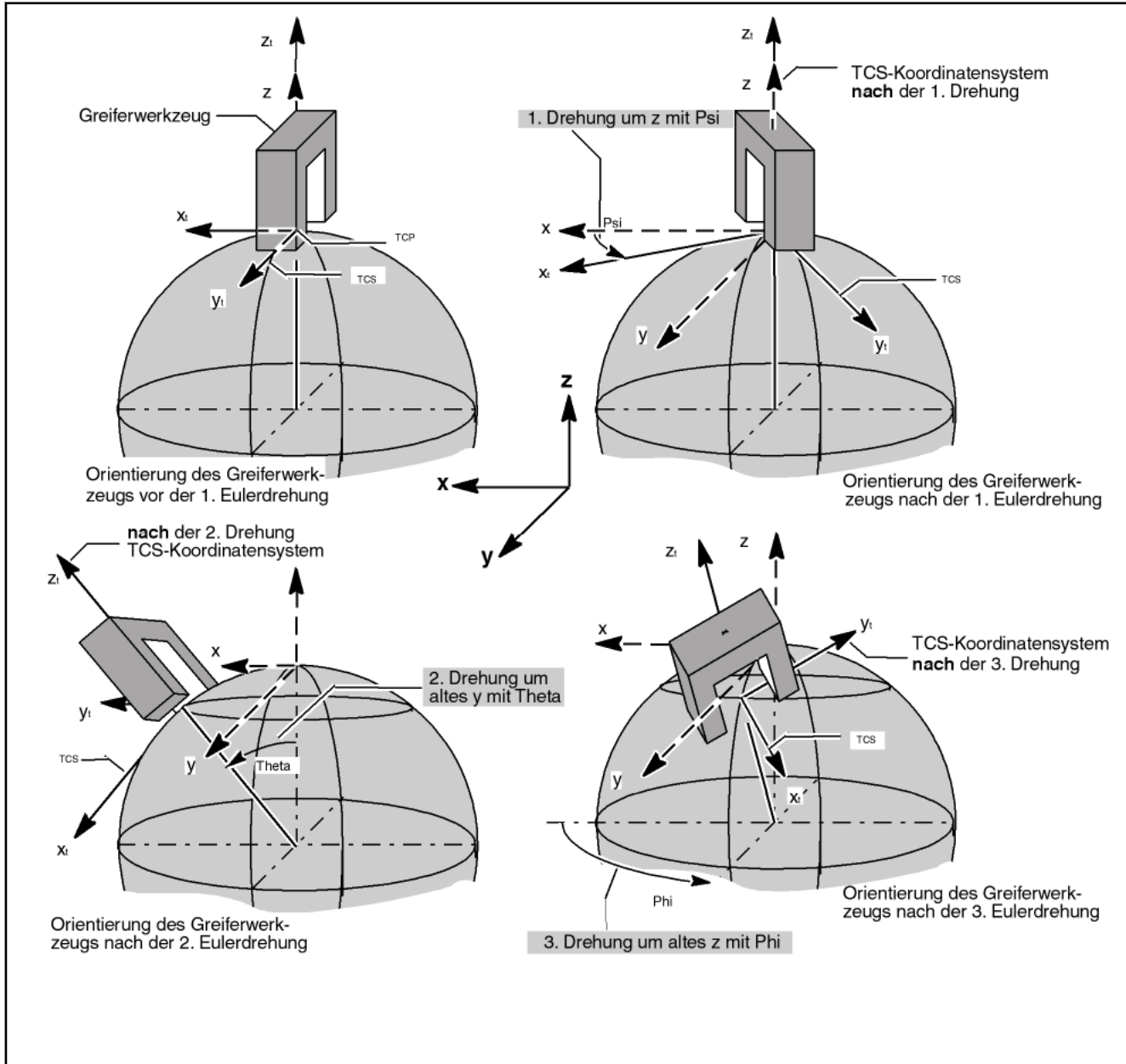


Abb. 10-3: Drehung um feste WCS-Koordinaten: $D_z(\psi)$, $D_y(\theta)$ und $D_z(\phi)$

Variante 3: Gemischte Drehung

1. $D_y(\theta)$: Drehung um die y -Koordinate des festen WCS mit Winkel θ
2. $D_z(\phi)$: Drehung um die z -Koordinate des festen WCS mit Winkel ϕ
3. $D_{z_t}(\psi)$: Drehung um die neue z_t -Koordinate des "TCS(2)" mit Winkel ψ

Damit ist das Zielkoordinatensystem erreicht.

Beispiel:

Greiferwerkzeug-Orientierung

Zur besseren Übersicht ist das Werkzeugkoordinatensystem auf den Nordpol des Programmkoordinatensystems verschoben.

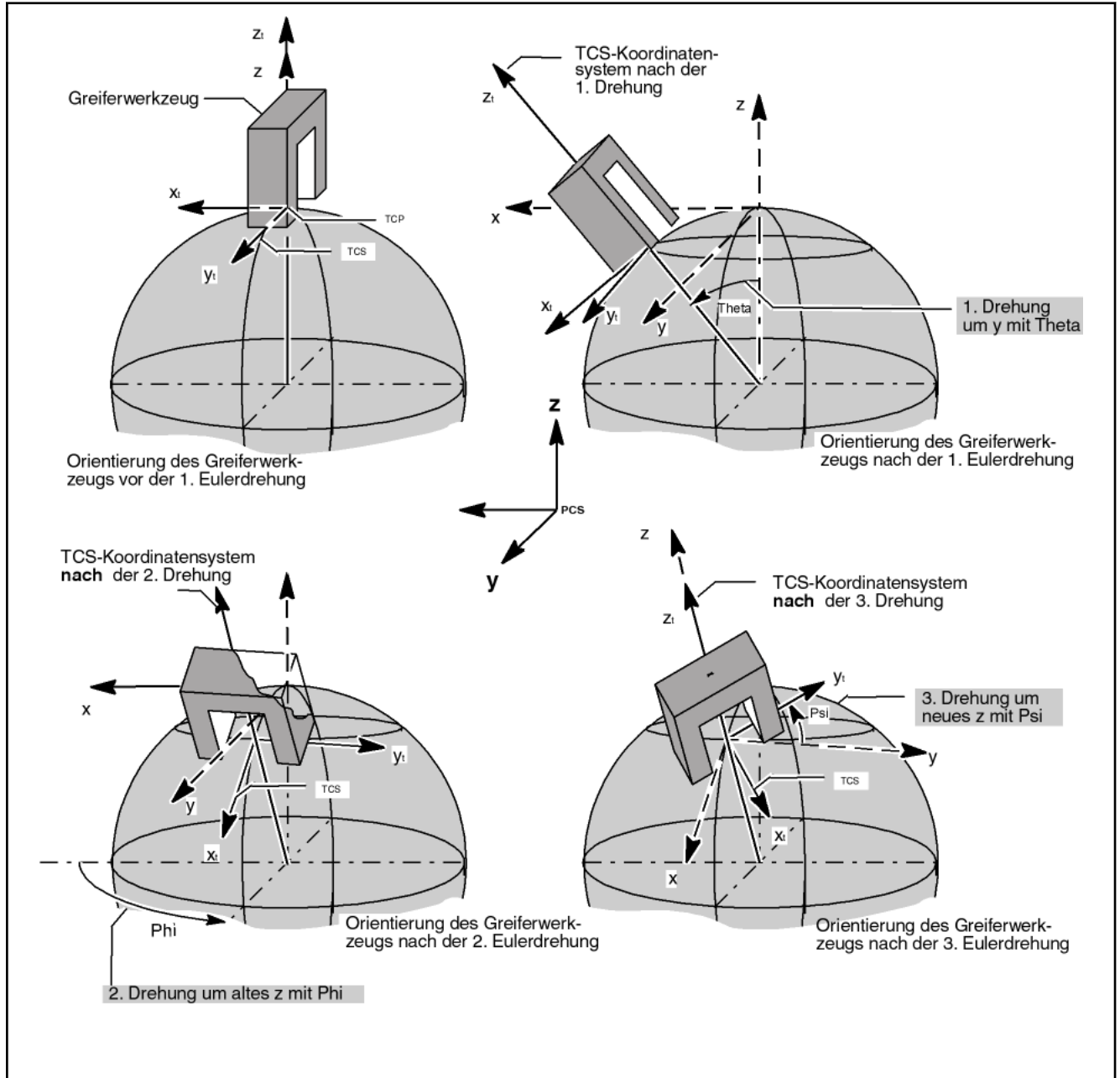


Abb. 10-4: Gemischte Drehung: $D_y(\theta)$, $D_z(\phi)$ und $D_{z'}(\psi)$

Beschreibung

Die Eulerwinkel lassen sich auch aus dem Ziel-TCS nachträglich wie folgt ablesen:

- theta: "Breitengrad" von Z_t (Winkel zwischen Nordpol und Z_t)
- phi: Längengrad von Z_t (Längengrad von x ist 0 Grad)
- psi: Abweichung von X_t von der Südrichtung

Für die Eulerwinkel phi, theta und psi gelten folgende Wertebereiche:

phi: $0^\circ \leq \text{phi} < 360^\circ$
 theta: $0^\circ \leq \text{theta} \leq 180^\circ$
 psi: $0^\circ \leq \text{psi} < 360^\circ$

Die eingeschränkten Wertebereiche bewirken, dass für jede beliebige Orientierung eines Zielkoordinatensystems ein eindeutiges Wertetripel für die Eulerwinkel bezüglich des Ausgangskoordinatensystems existiert. Bei der Programmierung der Eulerwinkel können jedoch beliebige Werte programmiert werden (z. B. negative theta-Werte). Zur Anzeige werden diese automatisch in die gültigen Wertebereiche umgerechnet.

Eulerwinkel werden auch verwendet, um das Werkstückkoordinatensystem WCS bezüglich des Basis-Werkstückkoordinatensystems BCS beliebig im Raum zu orientieren (siehe Dokumentation "MTX Funktionsbeschreibung - Grundlagen", Kap. "Placement: Schiefe Ebene G152.1...G159.5").



Die Eulerwinkel werden von Programmiersystemen unterschiedlich interpretiert. Beispielsweise können sich die Eulerwinkel auf andere Achsen, um die gedreht wird, beziehen.

Stellen Sie daher sicher, dass das verwendete Programmiersystem Eulerwinkel nach MTX-Programmiervorschrift unterstützt!

10.2.3 Orientierungsvektor

Die Orientierung rotationssymmetrischer Werkzeuge wird durch die zwei Eulerwinkel phi und theta beschrieben. Durch sie wird ein **Orientierungsvektor** O der Länge 1 definiert, der die Orientierung -bezogen auf das Programmkoordinatensystem- entlang der Werkzeuglängsachse zur Werkzeugaufnahme vorgibt. In der Freiformflächenbearbeitung (5-Achsbearbeitung) wird vorwiegend der Orientierungsvektor im NC-Programm verwendet.

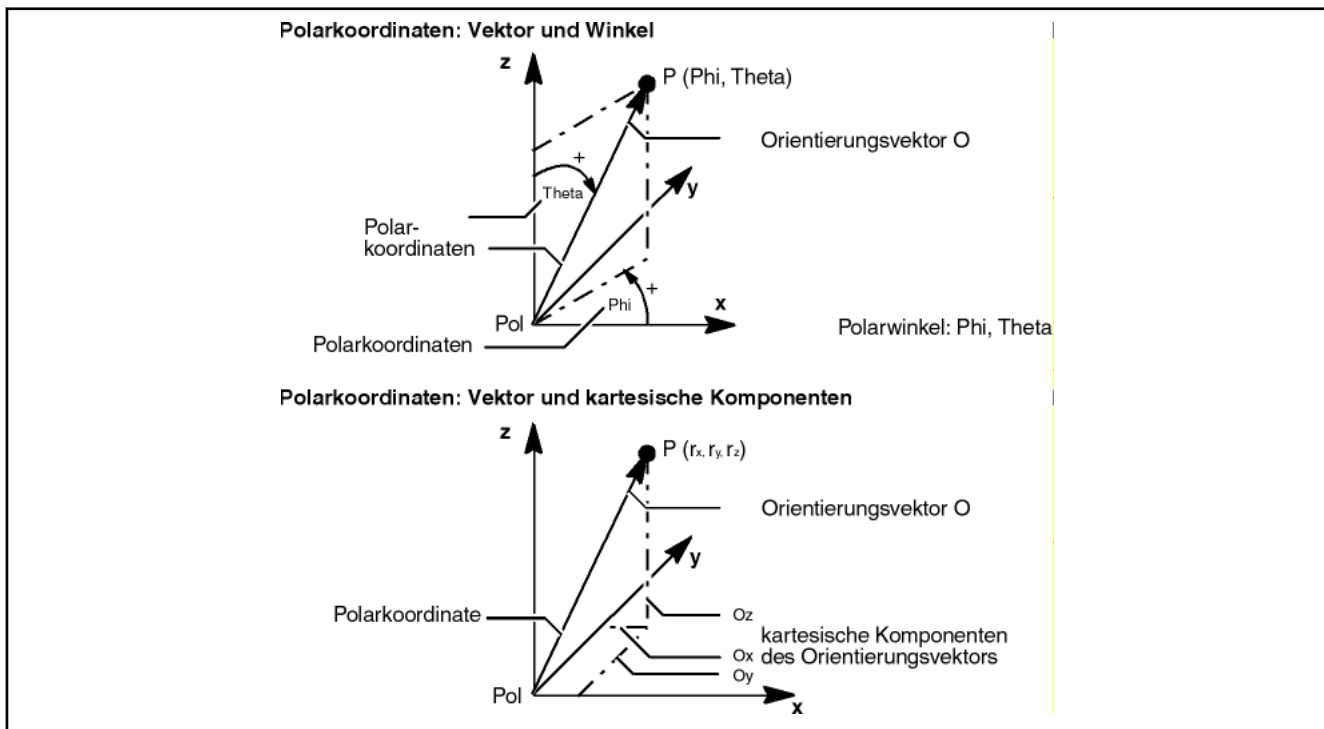


Abb. 10-5: Vektororientierung

Zwischen den Eulerwinkeln und den kartesischen Komponenten (O_x , O_y , O_z) des Orientierungsvektors O (mit der normierten Länge 1) besteht die Beziehung:

$$O = \begin{pmatrix} O_x \\ O_y \\ O_z \end{pmatrix} = \begin{pmatrix} \sin \Theta \cdot \cos \Phi \\ \sin \Theta \cdot \sin \Phi \\ \cos \Theta \end{pmatrix}$$

Beispiel:

Eulerwinkel und Orientierungsvektor

Die folgende Abbildung zeigt die Winkel- und die Vektor-Variante.

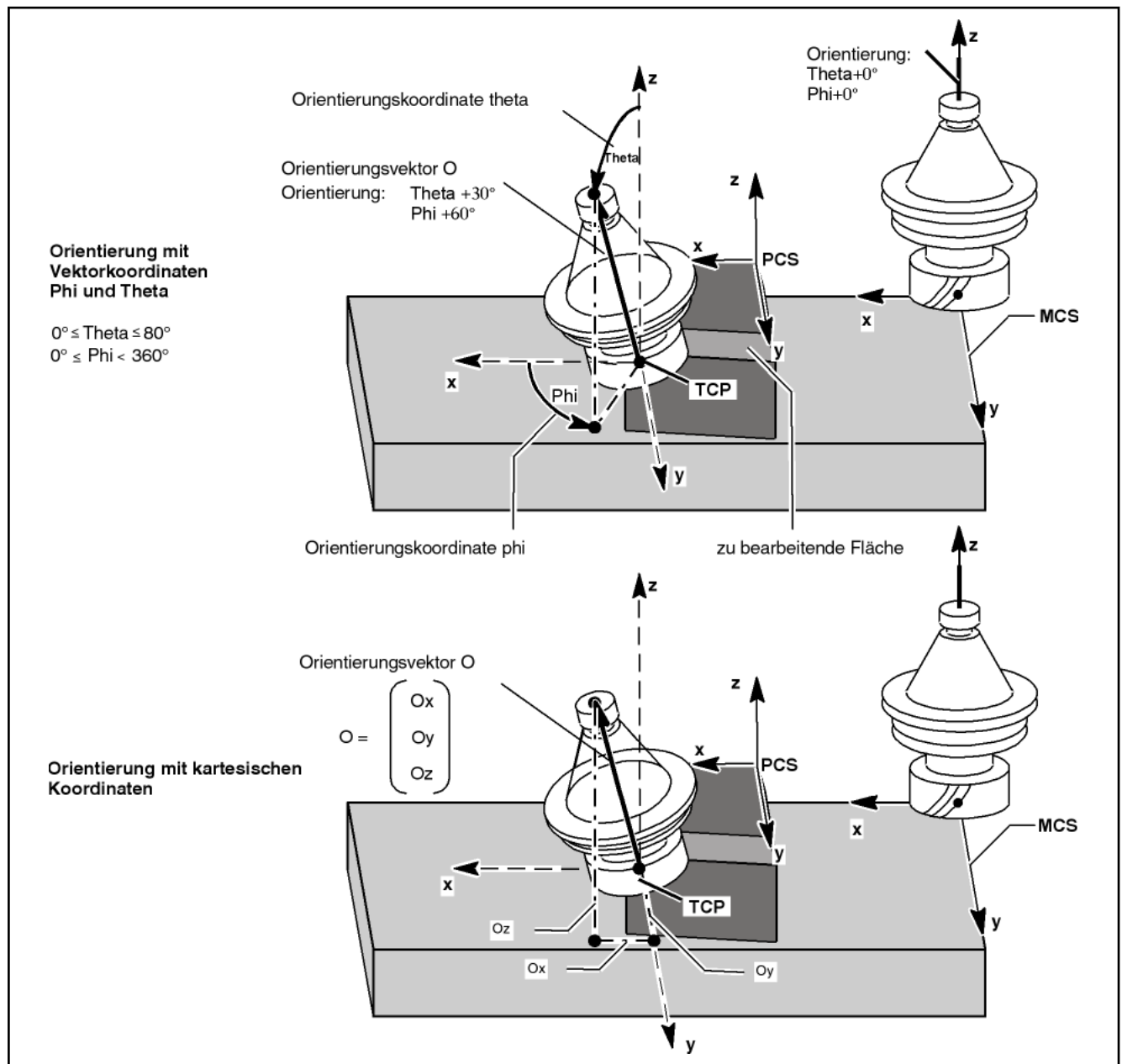


Abb. 10-6: Eulerwinkel und Orientierungsvektor

- Die Orientierung des Werkzeugs habe bezüglich des PCS die Eulerwinkel $\phi = 60^\circ$ und $\theta = 30^\circ$. Im NC-Programm kann diese Orientierung mit "phi60 theta30" oder mit "O(60,30)" angegeben werden.
- Eine zweite Möglichkeit besteht darin, die Orientierung der Werkzeuglängsachse als **kartesische Position** der Spitze des Orientierungsvektors O bzgl. des in den TCP verschobenen PCS anzugeben. Einsetzen der Winkel in die obige Formel für O ergibt $O_x = 0,25$, $O_y = 0,433$ und $O_z = 0,866$. Im NC-Programm kann diese Orientierung also auch mit "O(0.25,0.433,0.866)" angegeben werden.

10.2.4 Orientierungstensor

Die Orientierung nicht rotationssymmetrischer Werkzeuge wird durch die drei Eulerwinkel ϕ , θ und ψ beschrieben. Durch diese Eulerwinkel wird eine 3x3-Drehmatrix, die **Eulermatrix** festgelegt. Diese wird auch als **Orientierungstensor** O bezeichnet.

Die Eulermatrix oder der Orientierungstensor setzt sich wie folgt zusammen:

$$\vec{O} = \begin{bmatrix} O_{11} & O_{12} & O_{13} \\ O_{21} & O_{22} & O_{23} \\ O_{31} & O_{32} & O_{33} \end{bmatrix}$$

Zeilen und Spalten des Tensors haben die Länge "1" und stehen jeweils senkrecht aufeinander.

Die Spalten des Tensors sind die Vektorkomponenten der **Basisvektoren** e_x , e_y , e_z des verdrehten TCS bezüglich des Werkstückkoordinatensystems:

$$\vec{O} = \begin{pmatrix} O_{11} & O_{12} & O_{13} \\ O_{21} & O_{22} & O_{23} \\ O_{31} & O_{32} & O_{33} \end{pmatrix}$$

$$\vec{e}_x = \begin{pmatrix} O_{11} \\ O_{21} \\ O_{31} \end{pmatrix} \quad \vec{e}_y = \begin{pmatrix} O_{12} \\ O_{22} \\ O_{32} \end{pmatrix} \quad \vec{e}_z = \begin{pmatrix} O_{13} \\ O_{23} \\ O_{33} \end{pmatrix}$$

Beispiel:

3x3-Orientierungstensor eines nicht-rotationssymmetrischen Werkzeugs im Raum

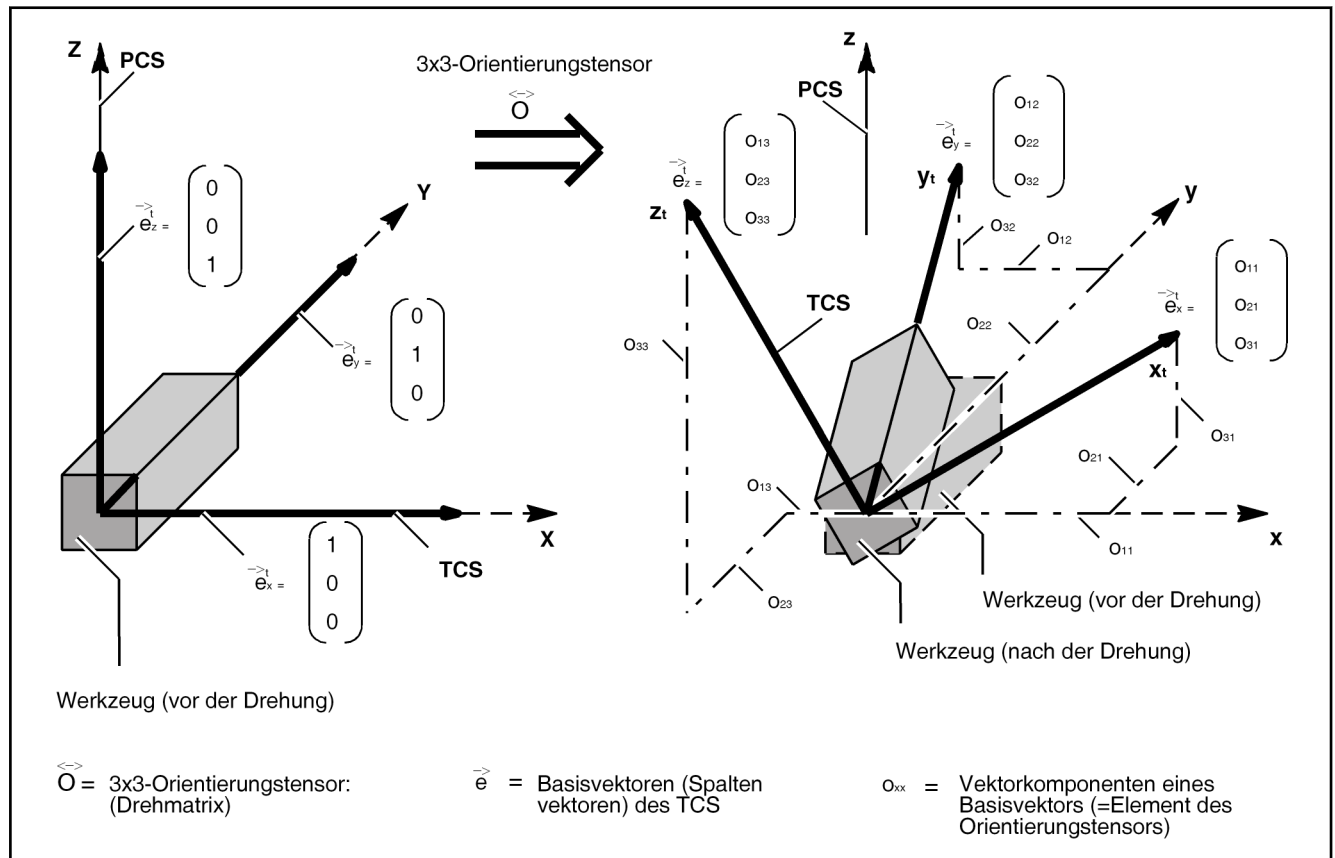


Abb. 10-7: Beispiel - Orientierungstensor



Der Orientierungstensor hat den Nachteil, dass seine neun Komponenten nicht unabhängig voneinander sind, da für eine allgemeine Orientierung drei Koordinaten ausreichen. Im NC-Programm werden daher die drei Eulerwinkel bevorzugt programmiert. Alternativ kann die Orientierung als Quaternion programmiert werden.

10.3 Orientierungsbewegung

10.3.1 Beschreibung

Allgemeines

Werkzeugorientierungen werden mit Hilfe der Eulerwinkel beschrieben (siehe [Kap. 10.2 "Definitionen zur Orientierung" auf Seite 161](#)).

Die Art der Orientierungsbewegung ist eine Eigenschaft der jeweils verwendeten Achstransformation. Die 100er-Stelle im Achstransformationstyp gibt dabei an, welche Orientierungsbewegungsart aktiviert wird.

100er-Stelle des Achstransformationstyps	Orientierungsbewegungsart
0	Keine Orientierung oder Orientierungsbewegung mit einer einzigen Winkelkoordinate. Diese Spezialfälle sind in den entsprechenden Achstransformationskapiteln beschrieben
1	Lineare Orientierungsbewegung mit den Eulerwinkeln phi, theta
2	Vektororientierungsbewegung
3	Tensororientierungsbewegung

Tab. 10-1: Orientierungsbewegungsart in Abhängigkeit des Achstransformationstyps

Beispiel:

Achstransformation und Orientierungsbewegungsart

Die erste konfigurierte Achstransformation in den Maschinenparametern habe den Typ 3232201 (5-Achstransformation LLLRR). Mit dem NC-Befehl COORD(1) wird diese Achstransformation und die Vektororientierungsbewegung aktiviert, da die 100er-Stelle des Typs gleich 2 ist.

Für die gleiche 5-Achskinematik existiert auch der Typ 3232101. Mit dieser 5-Achskinematik wird die Lineare Orientierungsbewegung aktiviert.

10.3.2 Vektororientierungsbewegung

Funktion

Bei rotationssymmetrischen Werkzeugen kann die interpolatorische Orientierungsbewegung über die beiden Eulerwinkel phi und theta oder über den Orientierungsvektor erfolgen. Im ersten Fall spricht man von Linearer Orientierungsbewegung, im zweiten Fall von Vektororientierungsbewegung.

Relevante NC-Funktionen

Coord(<i>)</i>	Achstransformation i aktivieren Orientierungsprogrammierung (phi, theta oder O(..)) ist nur in Zusammenhang mit einer entsprechenden Achstransformation i möglich
Coord(0)	Achstransformation und damit Orientierungsprogrammierung deaktivieren

Programmierung des Orientierungsvektors mit 3 Alternativen:

phi<phi> theta<theta>	Orientierung mit den Eulerwinkeln phi und theta. Die Namen "phi" und "theta" sind im Maschinenparameter CHAN/Ch[x]/Coord/Wcs/OriCoord[1..3] (MP7080 00010) festgelegt. "phi" und "theta" müssen immer zusammen programmiert werden. Programmierung ist nur absolut in Grad möglich. Beispiel: G1 x10 y50 z30 phi90 theta90
O(<phi>,<theta>)	siehe Angaben bei "phi, theta" Beispiel: G1 x10 y50 z30 O(90,90)
O(<x>,<y>,<z>)	Orientierung mit kartesischen Komponenten O_x, O_y, O_z des Orientierungsvektors. Programmierung absolut. Beispiel: G1 x10 y50 z30 O(0,1,0)

Tab. 10-2: Relevante NC-Funktionen

Programmierung des Orientierungsvektors mit Spline:

Innerhalb der Spline-Koeffizientenprogrammierung kann auch der Orientierungsvektor als Polynom programmiert werden.

O1(<o ₁₀ >,<o ₁₁ >,..., <o _{1n} >)	x-Komponente des Orientierungsvektors mit den Spline-Koeffizienten o ₁₀ bis o _{1n}
O2(<o ₂₀ >,<o ₂₁ >,..., <o _{2n} >)	y-Komponente des Orientierungsvektors mit den Spline-Koeffizienten o ₂₀ bis o _{2n}
O3(<o ₃₀ >,<o ₃₁ >,..., <o _{3n} >)	z-Komponente des Orientierungsvektors mit den Spline-Koeffizienten o ₃₀ bis o _{3n}

Tab. 10-3: Programmierung des Orientierungsvektors mit Spline-Koeffizienten



Das gemeinsame Nenner-Polynom (DN) gilt nicht für die Vektororientierung (siehe [Kap. 10.3.2 "Vektororientierungsbewegung"](#) auf Seite 170).

Programmierung der Orientierungsbewegung des Orientierungsvektors:

ROTAX(<R_x>,<R_y>,<R_z>) O(<β>) - oder - ROTAX(<phi_R>,<theta_R>) O(<β>)	ROTAX(..) definiert die Orientierung der Drehachse, um die der Orientierungsvektor dreht. Die Richtung kann in kartesischen Vektorkomponenten R_x, R_y, R_z oder in Polarkoordinaten phi_R, theta_R angegeben werden. O(β) definiert den Winkel β, um den der Anfangsorientierungsvektor um die Drehachse ROTAX dreht. Programmierung von O(β) ist nur inkremental und in Grad möglich. β kann beliebige Werte annehmen, d. h. es sind auch mehrere Umdrehungen möglich. Beispiel: ROTAX(0,45) O(720)

Bewegung des Orientierungsvektors:

Außer bei Spline-Interpolation ist die Bewegung des Orientierungsvektors immer eine Drehung um eine Drehachse durch den TCP (Ursprung des TCS).

Folgende Größen sind beteiligt:

- O_s : Anfangsorientierungsvektor
- O_e : Endorientierungsvektor
- R : Drehachse
- β : Drehwinkel

Um eine Orientierungsbewegung von einem Ausgangszustand des Orientierungsvektors O auszuführen, sind folgende Größen vorzugeben:

Orientierungsbewegung durch Vorgabe des Endorientierungsvektors O_e :

- Der Endorientierungsvektor O_e wird mit einer der drei Alternativen von [Tab. 10-2 "Relevante NC-Funktionen" auf Seite 171](#) programmiert.
- Der Orientierungsvektor dreht vom Anfangsvektor O_s um eine intern berechnete Drehachse R auf den Endvektor O_e .
- Die Drehachse R steht immer im rechten Winkel auf dem Orientierungsvektor O , der sich von der Anfangsorientierung auf die Endorientierung dreht. Die Drehachse geht durch den TCP.
- Die Bewegung des Orientierungsvektors überstreicht den Drehwinkel $\beta \leq 180^\circ$ und nimmt damit immer die kürzeste Verbindung zwischen Anfangs- und Endorientierung.
- Die Orientierungsbewegung erfolgt immer so, dass der Orientierungsvektor in derjenigen Ebene bleibt, die von O_s und O_e aufgespannt wird. Die Fläche, die vom Orientierungsvektor überstrichen wird, ist eben. Die Spitze des Orientierungsvektors läuft dabei auf einem Großkreis der Einheitskugel. Die Orientierungsbewegung ist unabhängig von der Programmierart in [Tab. 10-2 "Relevante NC-Funktionen" auf Seite 171](#).
- Damit eine Drehachse R bestimmbar ist, darf O_s und O_e nicht parallel oder antiparallel stehen.

Orientierungsbewegung mit vorgegebener Drehachse:

- Der Orientierungsvektor dreht vom Anfangswert O_s um eine programmierte Drehachse R mit einem programmierten Drehwinkel β .
- Die Drehachse wird mit ROTAX(..), der Drehwinkel mit O(..) programmiert.
Beispiel: N10 ROTAX(0,0,1) O(40)
- Der programmierte Drehwinkel kann beliebige Werte annehmen. Es entfällt die Einschränkung $\beta \leq 180^\circ$. Er wird generell als inkrementaler Wert interpretiert. Ein Winkel z. B. dreht von O(360) den Anfangsorientierungsvektor O_s genau eine Umdrehung weit auf seine ursprüngliche Position.
- Durch positive und negative Drehwinkel lässt sich der gewünschte Drehsinn erzeugen.
- Der Orientierungsvektor überstreicht einen Kegelmantel.

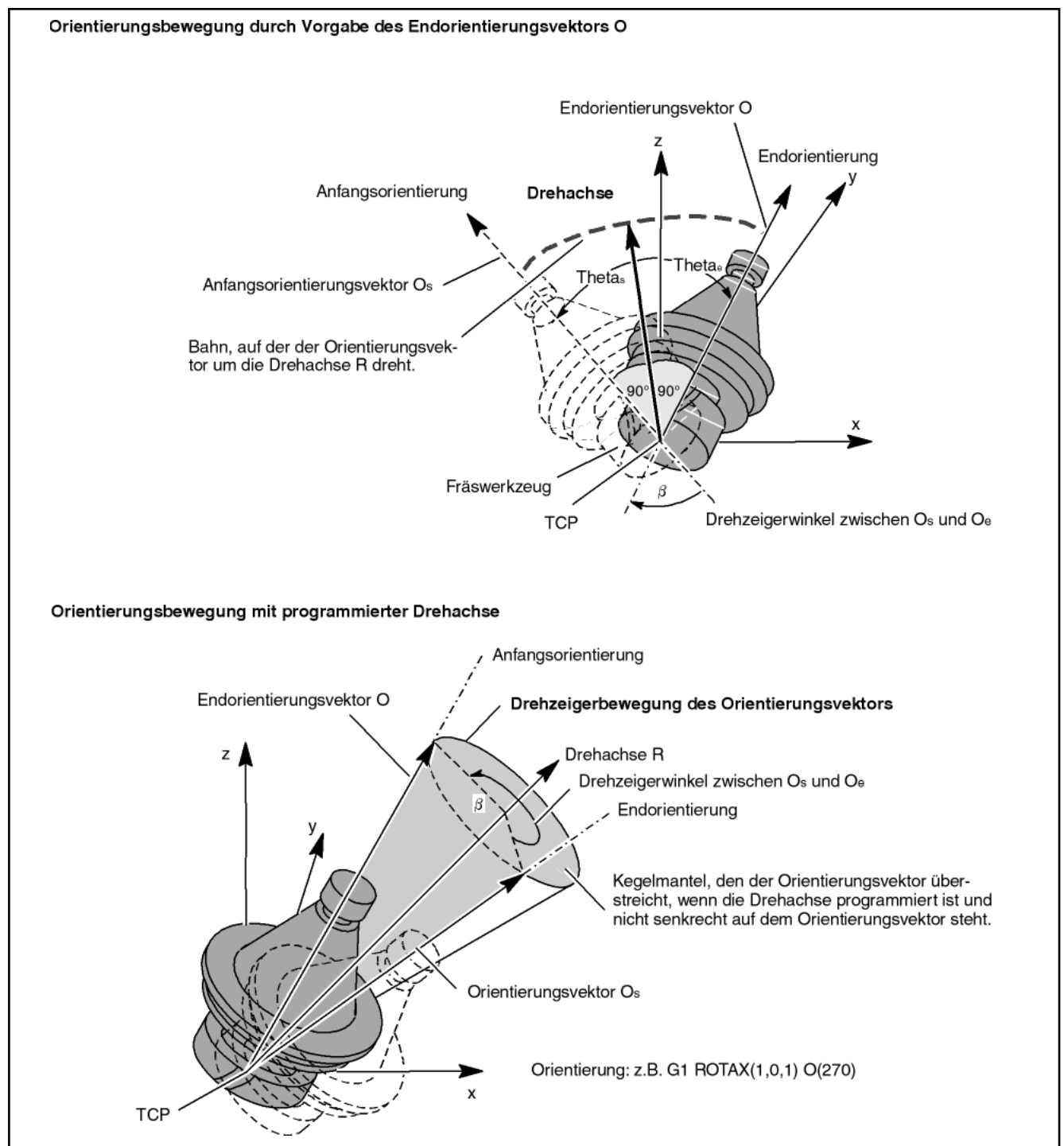


Abb. 10-8: Orientierungsbewegung ohne und mit programmierter Drehachse

Drehgeschwindigkeit des Orientierungsvektors:

Der Orientierungsvektor hat eine Länge von 57,23 mm ($= 360/2\pi$). In einem mit dem TCP als Ursprung mitbewegten, zum WCS parallelen Koordinatensystem, bewegt sich damit die Vektorspitze um 1 mm bei einer Drehung um 1 Grad. Der programmierte Vorschub F bezieht sich anteilig auf die Geschwindigkeit der Vektorspitze. Dazu die folgenden drei Beispiele.

Beispiel:

Reine Orientierungsbewegung

N10 O(0,0,1)	Ausgangszustand theta = 0
N20 O(0,1,0) F180	Bewegung auf phi=90 theta=90 mit Vorschub 180 mm/min

Tab. 10-4: Reine Orientierungsbewegung

Die Spitze des Orientierungsvektors wird mit 180 mm/min bewegt. Da die Spitze eine Bogenlänge von 90 mm überfährt, dauert die Bewegung ca. 0,5 min.

Beispiel:

Raum- plus Orientierungsbewegung , Orientierungsvektor Mitglied in der Vorschubgruppe

N10 FeedGrp(x, y, z, phi, theta)	Vorschubgruppe: x, y, z und O
N20 x0 O(1,0,0)	Ausgangszustand phi0 theta90
N30 x90 O(0,0,1) F90	Bewegung in x und O mit Vorschub 90 mm/min

Tab. 10-5: Raum- plus Orientierungsbewegung , Orientierungsvektor Mitglied in der Vorschubgruppe

Im Satz N30 bewegen sich x und O jeweils um 90 mm. Die Weglänge des Satzes ist daher $S^2 = 90^2 + 90^2$, also $S = 1,41 \cdot 90$. Die Bewegung dauert also $T = S/F = 1,41$ min.

Beispiel:

Raum- plus Orientierungsbewegung , Orientierungsvektor nicht Mitglied in der Vorschubgruppe

N10 FeedGrp(x, y, z)	Vorschubgruppe: x, y, z.
N20 x0 phi0 theta90	
N30 x90 phi0 theta0 F90	Bewegung in x und O mit Vorschub 90 mm/min

Tab. 10-6: Raum- plus Orientierungsbewegung , Orientierungsvektor nicht Mitglied in der Vorschubgruppe

Die Weglänge wird nur über die Mitglieder der Vorschubgruppe gebildet, d. h. $S = 90$. Die Bewegung dauert also $T = S/F = 1$ min. Die Orientierungsbewegung wird entsprechend mitgeschleppt.

Begrenzung der Geschwindigkeit des Orientierungsvektors (Omega-Programmierung)

Die Geschwindigkeit des Orientierungsvektors kann durch Programmieren von "Omega" begrenzt werden. Dies ist unter Umständen wichtig, wenn bei FeedGrp(x,y,z) eine Bewegung mit kleinem Raumanteil und großem Orientierungsanteil ausgeführt wird. Die Orientierungsbewegung würde dann nur noch durch die maximalen Achsgeschwindigkeiten begrenzt werden.

Omega-Programmierung

Omega<Wert>	Wert > 0: Die Geschwindigkeit der Vektorspitze wird auf Wert begrenzt. Wert hat die gleiche Einheit wie der Vorschub F.
Omega0	Omegaprogrammierung aus: Die Orientierungsgeschwindigkeit wird auf den Vorschub F begrenzt. Dies gilt auch, wenn Omega nicht programmiert ist.

Die Omega-Limitierung ist nur bei Bewegungen im Vorschub aktiv. Sie gilt nicht bei Eilgang (G0).

Beispiel:

Orientierungsbewegung mit Begrenzung durch Omega

N10 Omega180	
N20 G1 O(0,0,1)	Ausgangszustand theta = 0
N30 O(0,1,0) F600	Bewegung auf phi=90 theta=90 mit Vorschub 180 mm/min

Tab. 10-7: Orientierungsbewegung mit Begrenzung durch Omega

Der Vorschub im Satz N30 wird auf 180 mm/min begrenzt.

Bei gemischten Bewegungen wird die resultierende Bahngeschwindigkeit gegebenenfalls so begrenzt, dass der Orientierungsanteil den Omega-Wert nicht überschreitet.

Besonderheiten

Singularität der Bewegung des Orientierungsvektors:

Bei Achskonfigurationen, wie in der folgenden Abbildung dargestellt, kommt es zu nicht-stetigen Achsbewegungen, wenn der Orientierungsvektor nicht-tangential durch die Stellung $O = (0, 0, \pm 1)$ bewegt wird.

Beispiel:

Vermeidung eines Sprungs der C-Achse von 0 Grad auf 90 Grad.

Die MTX ist in solchen Fällen in der Lage **automatisch** einen "**Singularitätssatz**" einzufügen, der die C-Achse zunächst auf die erforderliche Position dreht, ehe die Positionierung fortgeführt wird. Siehe folgende Abbildung:

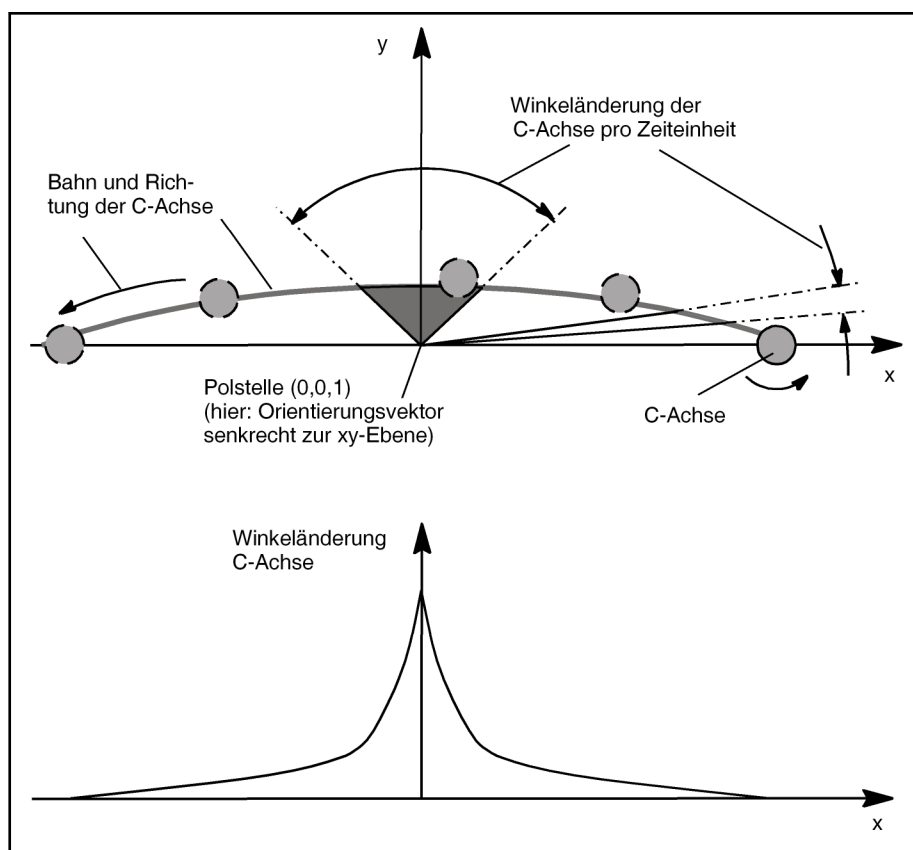


Abb. 10-10: Projektion der O-Bewegung auf die xy-Ebene

Für diesen Fall kann im Maschinenparameter TRA/AxTrafo[x]/EpsAxTrafo (MP 1030 00 60) "Epsilon-Umgebungen" ein Fangbereich (in Grad) um den Pol (0,0,1) definiert werden. Orientierungsbewegungen durch diesen Bereich werden dann so modifiziert, dass sie exakt durch die Polstelle gehen. Die Modifikation erfolgt so, dass an keiner Stelle die Differenz zwischen der ursprünglichen und der modifizierten O-Bahn größer als der Parameterwert ist.

Restriktionen

Eine Online-Korrektur von phi und theta ist nicht möglich.

Relevante Maschinenparameter

TRA/AxTrafo[x]/ EpsAxTrafo MP 1030 00160	Epsilon-Umgebung Definiert einen Fangbereich (in Grad), um die Orientierung $\theta = 0^\circ$. Orientierungsbewegungen durch die Epsilon-Umgebung werden dann exakt durch die Stelle $\theta = 0^\circ$ umgeleitet. Der Parameter ist nur für vektororientierungsfähige Achstransformationen relevant.

Tab. 10-8: Relevante Maschinenparameter

10.3.3 Lineare Orientierungsbewegung

Funktion

Die **lineare Orientierungsbewegung** benutzt ebenfalls die Koordinaten phi und theta. Die Bewegung erfolgt jedoch als eine Gerade in einer gedachten phi-theta-Ebene.

Auf der Einheitskugel beschreibt die Spitze des Orientierungsvektors im Gegensatz zur Vektororientierung keinen Großkreis, sondern eine Geradenbe-

wegung in den Koordinaten phi und theta. In der nachfolgenden Abbildung ist der Unterschied am Beispiel einer Bewegung von "phi0 theta45" auf "phi90 theta45" zu sehen.

Die lineare Orientierungsbewegung erfordert von der zugehörigen Achstransformation keine Singularitätenbehandlung (siehe [Kap. "Besonderheiten" auf Seite 175](#)), da kein phi-Sprung auftreten kann.

Im Folgenden sind die speziellen Eigenschaften der linearen Orientierungsbewegung und ihre Unterschiede zur Vektororientierung zusammengestellt:

- Voraussetzung ist eine in den Maschinenparametern konfigurierte Achstransformation mit zwei Orientierungskordinaten und der Orientierungskennung 1 (z. B. 3232101). Das Einschalten einer solchen Achstransformation mittels Coord(<n>) aktiviert automatisch die lineare Orientierungsbewegung.
- Die Orientierung bezieht sich immer auf das Basiskoordinatensystem, d. h. Koordinatentransformationen (Schiefe Ebene, Placement, Aufspannlagenkorrektur) werden im Gegensatz zur Vektororientierung **nicht** unterstützt.
- Die Programmierung geschieht über die Koordinatennamen phi und theta oder über die Vektorsyntax O(O_x, O_y, O_z). Beide Programmierarten sind äquivalent.
- Eine spezielle Wegsuchlogik sorgt dafür, dass keine Drehungen größer als 180° erfolgen.
- Eine Umrechnung der phi-theta-Werte in das Definitionsintervall $0^\circ \leq \text{phi} < 360^\circ$, $0^\circ \leq \text{theta} < 180^\circ$, erfolgt nicht.
In der Anzeige treten damit phi-theta-Werte außerhalb des Definitionsintervalls auf.
- Das Programmieren einer Drehachse mit den Funktionen "ROTAX(..) O(..)" ist **nicht** erlaubt.
- Die Abbildungsfunktionen "Spiegeln" und "Drehen" werden wie bei der Vektororientierung programmiert.
- Die Funktion "Online-Korrektur" kann für phi und theta benutzt werden.

Die folgende Abbildung zeigt die "Lineare Orientierungsbewegung" im Vergleich zur "Vektororientierungsbewegung":

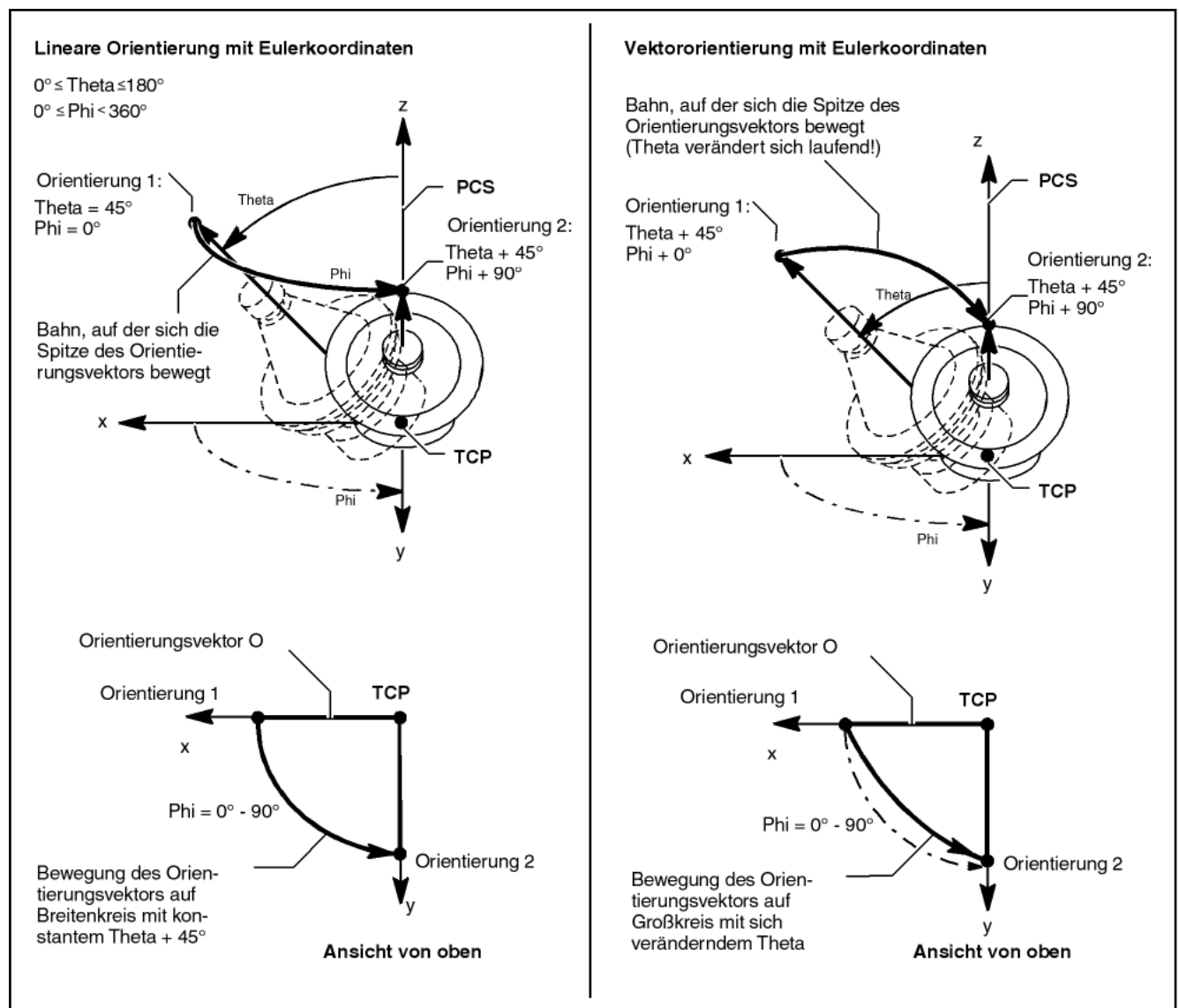


Abb. 10-11: Lineare Orientierung - Vektororientierung

Relevante NC-Funktionen

Coord(<i>)</i>	Achstransformation i aktivieren. Raumkoordinatenprogrammierung ist nur im Zusammenhang mit einer aktiven Achstransformation i möglich. Es dürfen keine Achsadressen programmiert werden, die durch das Umschalten auf Raumkoordinaten betroffen sind. i: Nummer der zu aktivierenden Achstransformation.
Coord(0)	Achstransformation und damit Raumkoordinatenprogrammierung deaktivieren.
Programmierung der linearen Orientierung mit zwei Alternativen:	

$\text{phi}<\text{phi}> \text{theta}<\text{theta}>$	Orientierung mit den beiden ersten Eulerwinkeln (Richtungswinkeln) phi, theta. Die Namen "phi" und "theta" sind im Maschinenparameter CHAN/Ch[x]/Coord/Wcs/OriCoord[1..3] (MP7080 00010) festgelegt. Programmierung ist nur absolut in Grad möglich. Beispiel: G1 x10 y50 z30 phi90 theta90
$O(<\text{phi}>, <\text{theta}>)$	siehe Angaben mit "phi, theta" Beispiel: G1 x10 y50 z30 O(90,90)
$O(O_x, O_y, O_z)$	Orientierung mit kartesischen Komponenten O_x, O_y, O_z des Orientierungsvektors. Programmierung absolut. Beispiel: G1 x10 y50 z30 O(0,1,0.707)

Tab. 10-9: Relevante NC-Funktionen

10.3.4 Tensororientierungsbewegung

Funktion

Bei der Orientierung von nicht rotationssymmetrischen Werkzeugen (z. B. Robotergreifer) spricht man von der Tensororientierung.

Ein fest mit dem Werkzeug verbundenes Werkzeugkoordinatensystem (TCS) hat in Bezug auf das Programmkoordinatensystem (PCS) eine Orientierung, die durch die Eulerwinkel phi, theta und psi oder durch den "3x3-Orientierungstensor" O festgelegt ist (siehe [Kap. 10.2.4 "Orientierungstensor" auf Seite 168](#)).

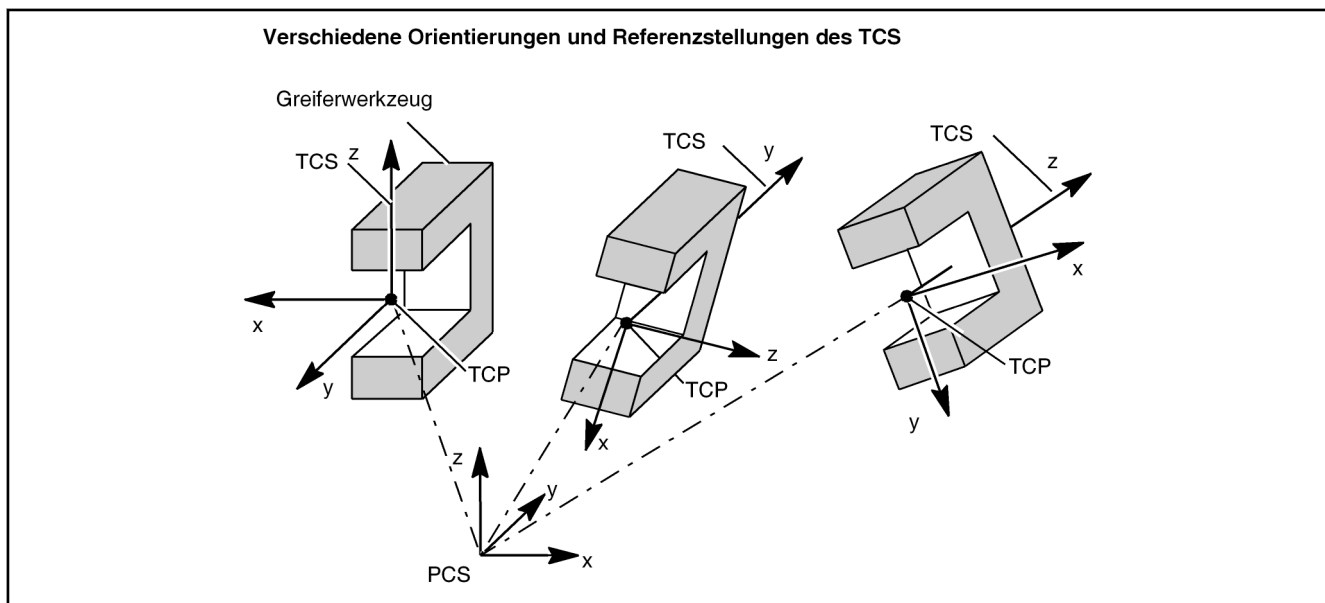


Abb. 10-12: Orientierungen des TCS

Relevante NC-Funktion

Die Tensororientierung wird mittels einer entsprechenden Achstransformation (siehe [Tab. 10-1 "Orientierungsbewegungsart in Abhängigkeit des Achstransformationstyps" auf Seite 170](#)) aktiviert (z. B. Typ 3333301).

Die Spaltenvektoren

$$\vec{e}_x = \begin{pmatrix} O_{11} \\ O_{21} \\ O_{31} \end{pmatrix} \quad \vec{e}_y = \begin{pmatrix} O_{12} \\ O_{22} \\ O_{32} \end{pmatrix} \quad \vec{e}_z = \begin{pmatrix} O_{13} \\ O_{23} \\ O_{33} \end{pmatrix}$$

des Orientierungstensors brauchen nicht 1-normiert angegeben zu werden. Die Normierung auf "1" erfolgt automatisch innerhalb der NC.

Die **TCS-Orientierung** kann mit den folgenden Alternativen programmiert werden:

phi<phi> theta<theta> psi<psi>	Orientierung mit Eulerwinkeln phi, theta und psi. Eulerwinkelnamen sind im Maschinenparameter CHAN/Ch[x]/Coord/Wcs/OriCoord[1..3] (MP7080 00010) festgelegt. Beispiel: G1 x10 y50 z30 phi90 theta90 psi45
O_x(<O₁₁>,<O₂₁>,<O₃₁>) O_y(<O₁₂>,<O₂₂>,<O₃₂>) O_z(<O₁₃>,<O₂₃>,<O₃₃>) - oder - O_x(<phi_x>,<theta_x>) O_y(<phi_y>,<theta_y>) O_z(<phi_z>,<theta_z>)	O_x(..) definiert die Richtung der x-Koordinate des TCS im Programmkoordinatensystem. Die Richtung kann in kartesischen Vektorkomponenten O_x(<O₁₁>,<O₂₁>,<O₃₁>) oder in Polarkoordinaten O_x(<phi_x>,<theta_x>) angegeben werden. Entsprechendes gilt für die Definition der Spaltenvektoren O_y(..) und O_z(..). Es brauchen nur zwei der drei TCS-Koordinaten programmiert werden. Beispiel: G1 x10 y50 z30 Ox(1,0,0) Oy(0,0.707,-0.707)

ROTAX(<R _x >,<R _y >,<R _z >) O(<β>) - oder - ROTAX(<phi _u >,<theta _u >) O(<β>)	ROTAX(..) definiert die Orientierung der Drehachse, um die der Orientierungstensor dreht. Die Richtung kann in kartesischen Vektorkomponenten R _x , R _y , R _z oder in Polarkoordinaten phi _x , theta _x angegeben werden. Die Programmierung von ROTAX(..) ist nur absolut möglich. O(..) definiert den Winkel β, um den der Anfangsorientierungstensor dreht. Programmierung von O(..) ist nur inkremental und in Grad möglich. β kann beliebige Werte annehmen, d. h. es sind auch mehrere Umdrehungen möglich. Beispiel: ROTAX(0,45) O(720) ROTAX(1,0,1) O(720)
O(<w>,<x>,<y>,<z>)	Quaternionprogrammierung, mit $w = \cos\beta/2$ $x = R_x \sin\beta/2$ $y = R_y \sin\beta/2$ $z = R_z \sin\beta/2$ Beschreibt eine Orientierung des TCS, welche durch Drehung um die Rotationsachse R = (R _x , R _y , R _z) mit Drehwinkel β aus der Stellung TCS = PCS erfolgt.

Tab. 10-10: Programmieralternativen - TCS-Orientierung



Eine detaillierte Darstellung der einzelnen Parameter finden Sie in der Dokumentation "MTX Programmierhandbuch".

Bewegung des Orientierungstensors:

Bei der Bewegung des Orientierungstensors sind folgende Größen beteiligt:

- O_s: Anfangsorientierungstensor (Anfangsorientierung des TCS)
- O_e: Endorientierungstensor (Endorientierung des TCS)
- R: Drehachse
- β: Drehwinkel

Um eine Orientierungsbewegung auszuführen, kann man folgende Größen vorgeben:

Orientierungsbewegung bei Vorgabe von O_e

Der Anfangsorientierungstensor O_s wird um eine intern berechnete Drehachse R auf den Endorientierungstensor gedreht. Für den intern berechneten Drehwinkel β gilt: $\beta \leq 180^\circ$.

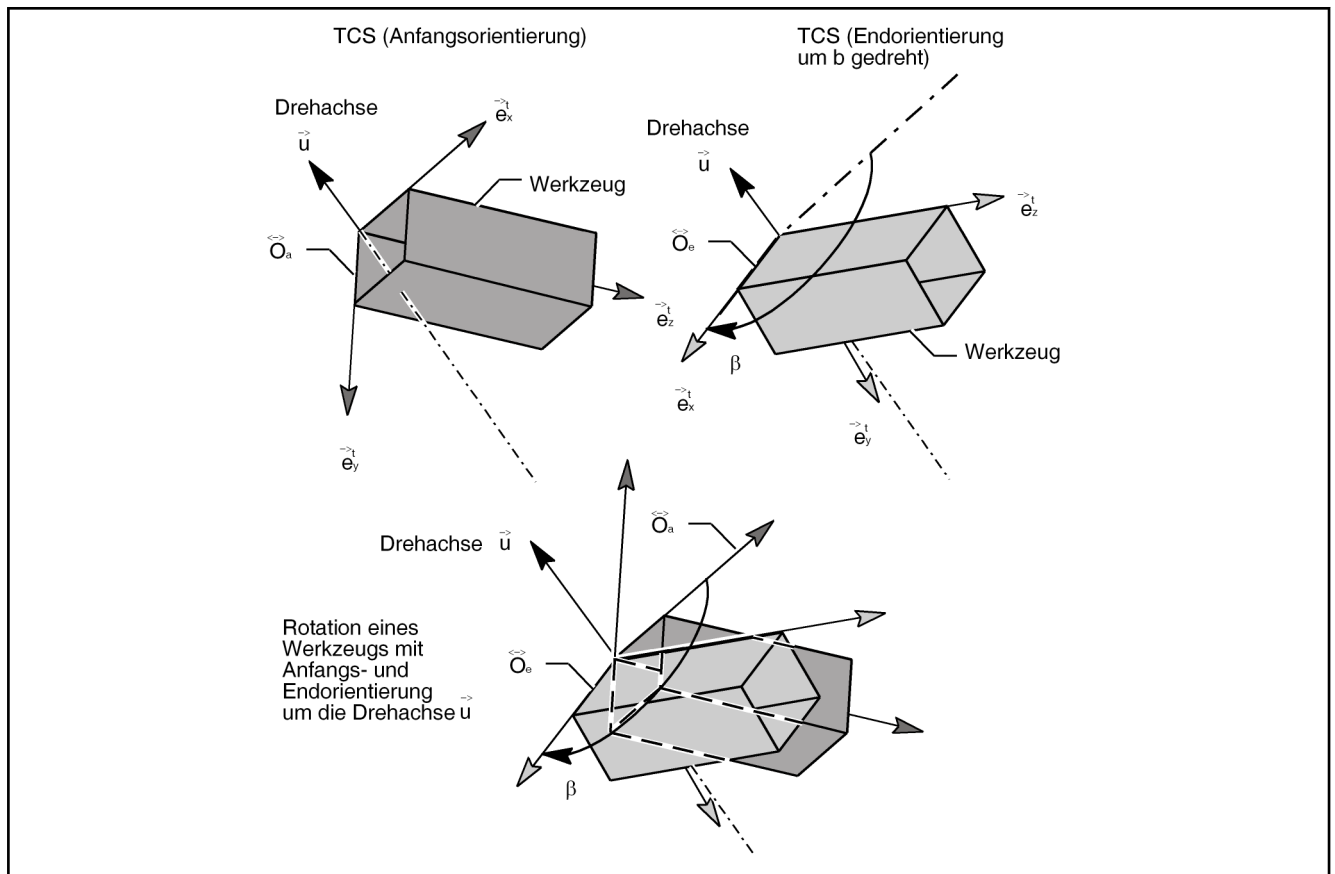


Abb. 10-13: Orientierungsbewegung

Orientierungsbewegung bei Vorgabe von Drehachse R und Drehwinkel β :

- Der Anfangsorientierungstensor $\vec{O}_s$ wird um die programmierte Drehachse R mit dem Winkel β gedreht.
- Die Orientierung der Drehachse wird mit "ROTAX(...)" programmiert.
- Die Bewegungen des Orientierungstensors wird mit "ROTAX(...) O(...)" erzeugt.

Beispiel:

ROTAX(...) O(360) dreht den Anfangsorientierungstensor $\vec{O}_s$ genau eine Umdrehung weit auf seine ursprüngliche Position. Der programmierte Drehwinkel wird generell als inkrementaler Wert interpretiert.

- Durch positive und negative Drehwinkel lässt sich der gewünschte Drehsinn erzeugen.

Die TCS-Koordinaten e_x , e_y , e_z beschreiben während der Orientierungsbewegung im Allgemeinen eine Kegelfläche um die Drehachse R.

Ausnahme:

R steht parallel zu e_x , e_y , e_z . Dann wird die Kegelfläche zu einer Kreisfläche.

Drehgeschwindigkeit des Orientierungstensors:

Wie bei der Vektororientierungsbewegung gilt die Zuordnung 1 mm $\triangleq$ 1 Grad. Bei einer reinen Orientierungsbewegung ergibt beispielsweise ein Vorschub $F=100$ mm/min eine Winkelgeschwindigkeit von 100 Grad/min der Drehzeigerbewegung um die Drehachse R. Bei einer Raum- plus Orientierungsbewegung hängt die Drehgeschwindigkeit davon ab, ob O Mitglied der Vorschubgruppe ist oder nicht. Es gelten hier entsprechend die Beispiele von Kap. "Relevante NC-Funktionen" auf Seite 170.

Die Geschwindigkeit der Orientierungsbewegung wird durch Omega (siehe [Kap. "Relevante NC-Funktionen" auf Seite 170](#)) begrenzt.

Besonderheiten: **Singularität der Bewegung des Orientierungstensors:**

Das Auftreten von Singularitäten hängt von der verwendeten Achstransformation ab. Bei der 6-Achstransformation vom Typ 3333301 tritt z. B. eine Singularität auf, wenn der Orientierungstensor zu einer Position von $\pm 90^\circ$ für die mittlere der drei Rundachsen führt.

Nicht stetige Orientierungsbewegungen an dieser Stelle werden durch **automatisches Einfügen eines NC-Satzes** überbrückt (siehe auch [Kap. 10.3.2 "Vektororientierungsbewegung" auf Seite 170](#)).

10.4 Tangentiale Werkzeugführung

10.4.1 Beschreibung

Funktion

Die tangentielle Werkzeugführung führt eine Werkzeugachse (Rund- oder Endlosachse) oder rotatorische Koordinate tangential zur programmierten Bahn in der jeweils aktiven Ebene eines Kanals mit.

Merkmale:

- Parametrierbarer Anstellwinkel (Winkel zwischen Symmetrieachse des Werkzeugs und der Bahntangente).
- Berücksichtigung der Werkzeugsymmetrie. Dadurch kann die Steuerung die bei Konturknicken erforderliche Anstellrotation des Werkzeuges selbstständig optimieren ("Kürzeste Weg"-Logik).
- Automatisches Einfügen von Zwischensätzen zur Drehung des Werkzeugs, wenn der Knickwinkel zwischen zwei Konturelementen einen parametrierbaren Wert überschreitet.
- Kommunikation mit der SPS, wenn bei einem Konturknick ein Zwischensatz automatisch eingefügt werden muss. Auf diese Weise kann die SPS vor der Ausführung des Zwischensatzes notwendige Schritte ausführen (z. B. Werkzeug anheben), bevor die NC das Werkzeug dreht.
- Komplette Parametrierung der "Tangentialen Werkzeugführung" per TangTool-Befehl (TTL) im Teileprogramm oder durch Maschinenparameter.
- Lesender Zugriff auf aktive Parameter von TangTool per CPL-SD-Funktion möglich.

Restriktionen

- Ist nur in Verbindung mit Gerade-Kreis-Interpolation anwendbar, nicht bei Splines.
- Funktion TangToolOri (TTO) "Tangentiale Werkzeugorientierung" (für Nibbeln/Stanzen; siehe [Kap. 12.6 "Tangentiale Werkzeugorientierung" auf Seite 455](#)) darf nicht gleichzeitig aktiv sein.
- Als Werkzeugachsen sind nur Rund- oder Endlosachsen sowie rotatorische Koordinaten, wie z. B. phi, theta, erlaubt.
- TangTool darf nicht in einem Verfahrssatz programmiert werden.
- Das automatische Einfügen eines Zwischensatzes ist nicht möglich, wenn bei aktiver "Tangentialer Werkzeugführung" ein einzelner WAIT-Satz zwischen 2 Verfahrssätzen programmiert ist.

- Tangentenstartwinkel und Tangentenendwinkel des programmierten Konturabschnittes müssen innerhalb des Endschalterbereichs der Werkzeugachse liegen (sofern die Werkzeugachse eine Rundachse ist).

Relevante NC-Funktionen

TangTool(1) -oder- TTL(1)	Einschalten der "Tangentialen Werkzeugführung" mit Initialwerten aus den Maschinenparametern. Das Anstellen der Werkzeugachse erfolgt mit dem nächsten Verfahrssatz. In Abhängigkeit vom aktiven Zwischensatzwinkel dreht die Werkzeugachse entweder innerhalb eines automatisch generierten Zwischensatzes oder sprunghaft auf die Anstellposition. Die Parameter von TTL sind im Programmierhandbuch beschrieben.
TangTool(0) - oder - TTL(0)	Ausschalten der "Tangentialen Werkzeugführung".

Tab. 10-11: Relevante NC-Funktionen

Relevante IF-Signale

iCh_TangTRotCmd	"Werkzeugdrehung (TangTool)" (Kanalinterface). High-Signal zeigt der SPS an, dass der Knickwinkel zwischen zwei Konturelementen den aktuell eingestellten Zwischensatzwinkel überschreitet. Ob die NC zur Ausführung des Zwischensatzes auf die Freigabe durch die SPS warten soll, legen Sie per MP 7050 00260 oder durch den TangTool-Parameter "PLC" fest.
qCh_TangTRotRel	"TangTool (TTL) Freigabe" (Kanalinterface) Freigabe für TangTool Werkzeugdrehung. High-Signal gibt die Ausführung des Zwischensatzes durch die NC frei, sofern MP 7050 00260 oder TangTool-Parameter "PLC" mit dem Wert "1" belegt ist. Auf den Zwischensatz folgende Sätze führt die Steuerung erst dann aus, wenn die SPS dieses Signal wieder zurücksetzt.

Tab. 10-12: Relevante IF-Signale

Relevante SD-Funktionen

SD(131,1)	liefert die Kanalkoordinatennummer (Kanalachsnummer) der Drehachse
SD(131,2)	liefert die Werkzeugsymmetrie
SD(131,3)	liefert den Anstellwinkel in Grad
SD(131,4)	liefert den Zwischensatzwinkel in Grad

Tab. 10-13: Relevante SD-Funktionen

Relevante Maschinenparameter (MP)

Die Werte der Maschinenparameter wirken als Initialisierungswerte, wenn TangTool ohne Parameter programmiert ist. Sie können die Initialisierungswerte im TangTool-Satz durch die in Klammern angegebenen Parameter überschreiben (Syntax siehe Dokumentation "MTX Programmierhandbuch"):

Werkzeugorientierung und Orientierungsbewegung

7050 00210	Systemachsnummer der Drehachse (hier kann keine Koordinate angegeben werden)	(TAX)
7050 00220	Werkzeugsymmetrie	(SYM)
7050 00230	Zwischensatzwinkel	(IA)
7050 00240	Anstellwinkel	(ANG)
7050 00250	Initialisierung des Anstellwinkels	-
7050 00260	NC-SPS-Kommunikation für Zwischensatz	(PLC)

Tab. 10-14: Relevante Maschinenparameter (MP)

11 Achs-Transformation

11.1 Einleitung

11.1.1 Erläuterungen

Im Kapitel **Achs-Transformation** werden die zurzeit anwendbaren Transformationen in der NC beschrieben. Eine Zusammenstellung aller Achs-Transformationen finden Sie in [Kap. 11.1.8 "Übersicht Achs-Transformationen \(tabellarisch\)" auf Seite 197](#).

Zusätzlich befindet sich im letzten Unterkapitel die **Achs-Kinematikkalibrierung**, die die Genauigkeit der Positionierung der transformierten Achsen beträchtlich erhöht.

11.1.2 Prinzip der Achs-Transformation

Verlaufen die Koordinatenachsen (=Programmkoordinaten) des werkstückbezogenen Koordinatensystems nicht mehr parallel zu den Maschinenachsen, kann die NC zunächst keine Achssollwerte direkt für die Maschinenachsen herleiten. Das Erzeugen der Achssollwerte müsste in diesem Fall ein Programmiersystem bewerkstelligen und im Programm hinterlegen. Dies hat den Nachteil, dass das Programm nur auf einer Maschine mit einem bestimmten Werkzeug lauffähig wäre. Diese Einschränkungen umgeht die NC per **Achs-Transformation**.

Die Programmkoordinaten werden als "Systemkoordinaten" interpretiert, wenn Achs-Transformationen aktiv sind. Die Systemkoordinaten unterteilen sich außerdem in Raumkoordinaten und kartesische Koordinaten. Solche Koordinaten beziehen sich auf ein maschinenunabhängiges Koordinatensystem, das irgendwo im Raum positioniert und orientiert ist (siehe dazu Dokumentation "IndraMotion MTX 14VRS Funktionsbeschreibung - Grundlagen", Kap. "Achsen, Koordinaten, Koordinatensysteme").

Die Umsetzung der Programmkoordinaten in das reale Achssystem der Maschine stellt eine maschinenspezifische Achs-Transformation her, die die geometrisch-kinematischen Zusammenhänge von Maschine und Werkzeug berücksichtigt bzw. definiert. Hierbei berechnet die Achs-Transformation für jede anzufahrende Position im Raum die notwendigen Anteile aller beteiligten realen Achsen.

Zur Bearbeitung in einer "schiefen Ebene" können ggf. bis zu 5 Achsen notwendig sein, um eine Oberfläche plan zu fräsen. Programmiert wird in diesem Fall nur der Satz "G1x 100 phi 100 theta 45" mit den transformierten Koordinaten x, phi und theta. Diese wirken sowohl auf die Linearachsen X, Y und Z, als auch auf die Rundachsen B und C.

Für die Achs-Transformation müssen die Programmkoordinaten intern durch "Rückwärtstransformation" "aufbereitet" werden (siehe auch folgende Abbildung):

1. Sind die programmierten Koordinaten, bezogen auf das Programmkoordinatensystem, durch Korrekturen verschoben und/oder verdreht, werden die Korrekturen zunächst durch eine "Koordinatentransformation" herausgerechnet. Dadurch erhält man Raumkoordinaten, die sich auf das Basis-Werkstückkoordinatensystem (BCS) beziehen.
2. Diese Raumkoordinaten sind Eingangsgröße für die Achs-Transformation AT2.
3. Werden diese Raumkoordinaten nach "Herausrechnen" der Achs-Transformation AT2 auf kartesische Koordinaten rückwärts transformiert, erhält man die Eingangsgröße für die Achs-Transformation AT1.

Diese kartesischen Koordinaten ergeben nun durch "Herausrechnen" der Achs-Transformation AT1 die eigentlichen Achssollwerte.

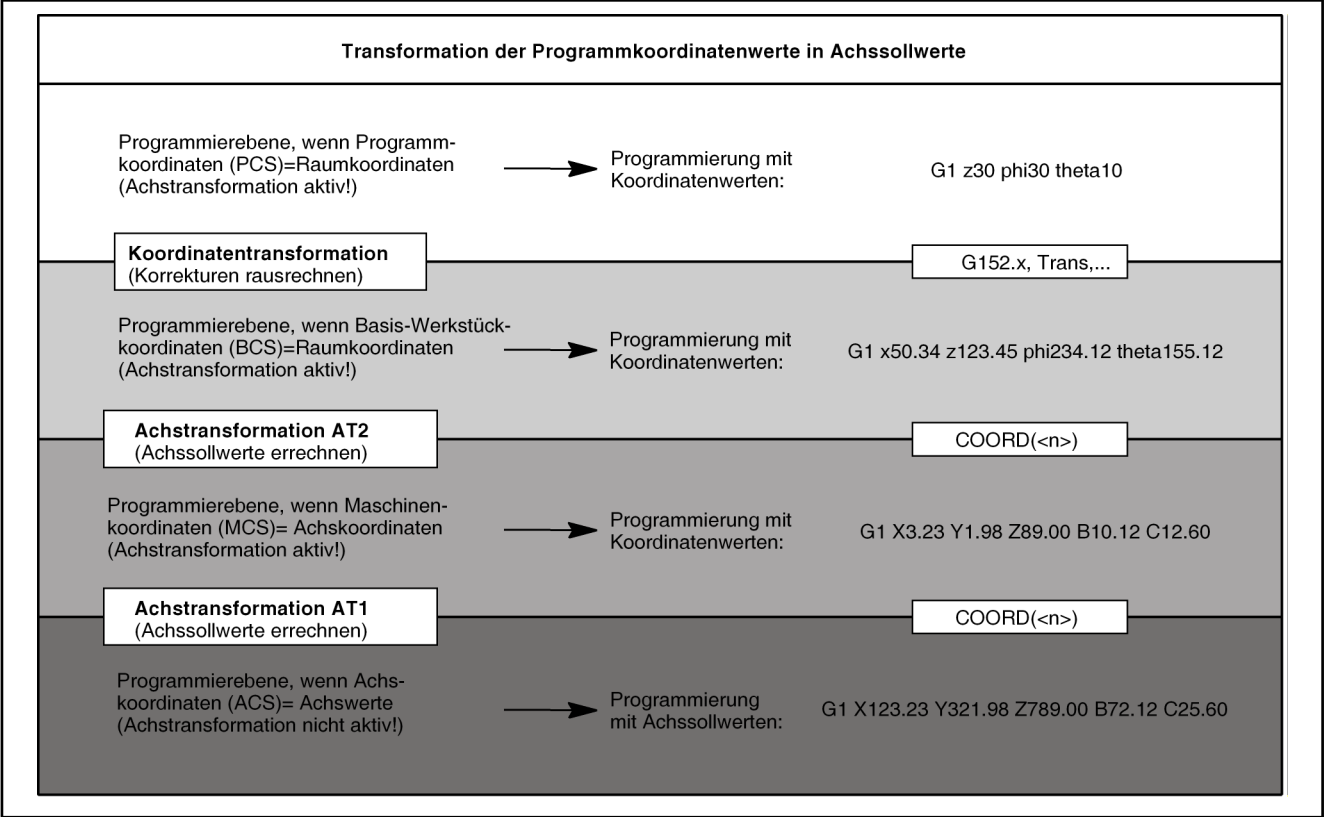


Abb. 11-1: Transformation der Programmkoordinatenwerte in Achssollwerte

Aus der Art und Weise, wie Achssollwerte per Achs-Transformation aus den Programmkoordinaten erzeugt werden, erkennt man, dass Raumkoordinaten bzw. kartesische Koordinaten mit jeder Achskonfiguration individuell verschieden auf die physikalische Achskonfiguration wirken können. Dies lässt sich durch unterschiedliche Einstellungen der Achs-Transformationen in MP realisieren.

11.1.3 Definitionen

Typ-Kennung

Alle Achs-Transformationen werden durch eine Typ-Kennung unterschieden und werden in MP 1030 00110 eingetragen.

Die Typ-Kennung ist eine siebenstellige Zahl, deren Ziffern folgende Bedeutung haben:

Ziffer	1	2	3	4	5	6 u. 7
	A _l	A _r	E _l	E _r	O	NN

Ziffer	Abk.	Bedeutung	Optionen
1	A _l	Ausgänge der Vorwärtstransformation, linear	0..3
2	A _r	Ausgänge der Vorwärtstransformation, rotatorisch	0..3

Ziffer	Abk.	Bedeutung	Optionen
3	E _l	Eingänge der Vorwärtstransformation, linear	0..3
4	E _r	Eingänge der Vorwärtstransformation, rotatorisch	0..3
5	O	Kennung der unterstützten Orientierungsbewegung	0: Keine Orientierungsbewegung 1: Lineare Orientierungsbewegung in den rotatorischen Achsen 2: Vektorrotation (Drehzeigerbewegung) des Orientierungsvektors 3: Tensorrotation der Orientierung
6 und 7	NN	Fortlaufende Nummerierung für interne Zwecke	01 .. 99 Nummer ist typbezogen

Tab. 11-1: Typ-Kennung - Bedeutung der Ziffern

Aus der Typ-Kennung lässt sich ablesen, welche Funktionen bei Anwahl der Achs-Transformation wirksam sind:

- Die Achs-Transformation mit der Nummer i (aus dem i-ten Parametersatz im MP) wird per Befehl COORD(<i>) aktiviert, sofern die Konfiguration in den MP keine fehlerhaften Einträge enthält.
- Unterstützt die aktivierte Achs-Transformation **Vektor- oder Tensororientierung**, werden auch die entsprechenden NC-Funktionen für die Orientierungsbewegung aktiviert.
- Die Programmierung von **Koordinatennamen** wird aktiviert, d. h. es erfolgt eine Umschaltung der Achsen auf Koordinaten.
- Achsen, die an der Achs-Transformation beteiligt sind, dürfen bei aktiver Achs-Transformation nicht mehr programmiert werden.

Beispiel:

Koordinatennamen: | x| y| z| phi| theta| psi| U| V|

Achsamen: | X| Y| Z| A| B| C| U| V|

Die Achs-Transformation legt fest, wie viele der maximal sechs Raumkoordinaten programmiert werden können. Die Raumkoordinaten bestehen aus maximal drei translatorischen "Positionskoordinaten" und maximal drei rotatorischen "Orientierungskoordinaten", deren Namen sich in MP frei einstellen lassen (im weiteren Verlauf wird "x", "y" und "z" für Positionskoordinaten, und "phi", "theta" und "psi" für Orientierungskoordinaten verwendet). U und V sind im Beispiel weitere an der Achs-Transformation nicht beteiligte Kanalachsen.



Die Umschaltung auf Achs-Transformation bewirkt, dass die Anzeige der Werkstückkoordinaten von "Achspositionen" auf "Koordinatenwerte" springt. Die Achsnamen des Achsfeldes wechseln in die Koordinatennamen wie im obigen Beispiel gezeigt.

Kennung für Achsanordnungen

Um die Anordnung, Typ und Anzahl der Achsen, die an 5- und 6-Achs-Transformationen beteiligt sind, schnell zu erfassen, wird folgende Kennung definiert:

L = Bezeichnung für lineare Achsen

R = Bezeichnung für Rotationsachsen

Beispiel 1:

LLLRR bedeutet drei lineare Achsen gefolgt von zwei Rotationsachsen

Beispiel 2:

RLLLR bedeutet eine Rotationsachse dreht drei lineare Achsen gefolgt von einer Rotationsachse (meist Rundtisch)

Eingangsgrößen, Ausgangsgrößen

Eingangsgrößen: Bezeichnen die transformierten Koordinaten, aus denen die Achs-Transformation durch Rückwärtstransformation reale Achsen/Koordinaten-Positionen erzeugt.

Ausgangsgrößen: Bezeichnen die Achsen oder Koordinaten, aus denen die Achs-Transformation transformierte Eingangsgrößen erzeugt.

Beispiel:

5-Achs-Transformation

Ausgangsgrößen im MCS: X,Y,Z,R2,R1

Eingangsgrößen im BCS: x, y, z, φ , ϑ

11.1.4 Wirkstellen

Eigenschaften

- Die Wirkstellen AT1 und AT2 bestimmen die Stellen im System, an denen Achs- und/oder Koordinatenwerte eines Bezugssystems in Koordinatenwerte eines anderen Bezugssystems umgerechnet (transformiert) werden.
- An jeder Wirkstelle dürfen nur ganz bestimmte Transformationen wirken.
- Eine Wirkstelle wird indirekt durch die entsprechende Transformation festgelegt und kann daher nicht programmiert werden.

Wirkstelle AT1:

Eine Achs-Transformation an der Wirkstelle AT1 transformiert zwischen dem Achskoordinatensystem (ACS) und einem fest mit dem Maschinenkörper verbundenem kartesischen Maschinenkoordinatensystem (MCS). Eine AT1 ist bei Maschinenkinematiken ohne kartesische Achsanordnung wie z. B. Stabkinematiken notwendig.

Zur Zeit sind dies die folgenden Achs-Transformationen an AT1:

- Bipod-Transformation (Typ: 2121001, 3131001)
- SCARA-Transformation (Typ: 2103011)
- Nichtorthogonale Achsen (Typ: 2020001, 2020002, 3030001)

Wird einer der oben angeführten Typen im NC-Programm aktiviert, platziert sich die entsprechende Achs-Transformation automatisch auf die Wirkstelle AT1.

Einer Achs-Transformation an AT1 kann eine geeignete Achs-Transformation AT2 (zwischen MCS und BCS) nachgeschaltet werden.

Beispiel:

Die Greiferkoordinaten (x, y, z, ϕ , ϑ) eines SCARA-Roboters (AT1) werden mit Hilfe der 5-Achs-Transformation (AT2) programmiert.

Wirkstelle AT2: Eine Achs-Transformation an der Wirkstelle AT2 transformiert zwischen dem Maschinenkoordinatensystem MCS und dem Basis-Werkstückkoordinatensystem BCS (= WCS(1)).

An der AT2 kommen beispielsweise folgende Achs-Transformationen zum Einsatz:

- Stirnseiten-Transformation (Typ: 2011001)
- Zylindermantel-Transformation (Typ: 3021001, 3021002)
- 5-Achs-Transformation LLLRR (Typ: 3232201, 3232101, 3032001)
- 5-Achs-Transformation RLLLR (Typ: 3232202)
- 5-Achs-Transformation RRLLL (Typ: 3232203)
- 5-Achs-Transformation LLLRR-Kardanisch (Typ: 3232211)
- 6-Achs-Transformation LLLRRR (Typ: 3333301, 3033001)

Wird einer der oben angeführten Transformationstypen im NC-Programm aktiviert, platziert sich die entsprechende Achs-Transformation automatisch auf die Wirkstelle AT2. Je nach Maschinenkinematik muss einer AT2 eine AT1 vorgeschaltet sein (siehe Beispiele in der Dokumentation "IndraMotion MTX 14VRS Funktionsbeschreibung - Grundlagen", Kap. "Achsen, Koordinaten, Koordinatensysteme"), die kartesische Koordinaten im MCS zur Verfügung stellt.

11.1.5 Konfigurierung von Achs-Transformationen

Mittels des MP 1030 00120 "Systemachsen/-koordinaten der Transformation" werden die Eingangsgrößen für die Vorwärtstransformation festgelegt.

Beispiel:

Bipod-Transformation an der Wirkstelle 1

MP 1030 00120

[1] 1	(Achse YL)
[2] 2	(Achse YR)
[3] 3	(Achse Za)
[4] 9	(Achse Sc)

Beispiel:

Achs-Transformation an der Stelle AT2, aufgesetzt auf Bipod-Transformation

MP 1030 00120

[1] -6	(Systemkoordinate X)
[2] -5	(Systemkoordinate Y)
[3] -4	(Systemkoordinate Z)
[4] 7	(Achse C)
[5] 6	(Achse B)

Beispiel:

5-Achs-Transformation an der Stelle AT2, Keine Transformation an Stelle AT1

MP 1030 00120

[1] 1	(Achse Xa)
[2] 2	(Achse Ya)
[3] 3	(Achse Za)
[4] 7	(Achse C)
[5] 6	(Achse B)

Bei MP 103000130 "Achs-/Koordinatenpositionen der Referenzstellung" beziehen sich die Positionen für Achs-Transformationen an AT2 auf das MCS. Für Achs-Transformationen an AT1 beziehen sie sich auf das ACS.

Im MP 1030 00140 werden die Längen und Winkelparameter für die aktuelle Achs-Transformation eingestellt. Diese können mit Hilfe der Funktion "Achs-Kinematikkalibrierung" bzgl. der **exakten** Ausrichtung der Achsen/Koordinaten optimiert werden (siehe Dokumentation "IndraMotion MTX 14VRS Funktionsbeschreibung - Grundlagen", Kap. "Achsen, Koordinaten, Koordinatensysteme").

11.1.6 Ein- und Ausschalten von Achs-Transformationen

- Bedingungen:**
- Einschalten: AT1 muss vor AT2 aktiviert werden
 - Ausschalten: AT2 muss vor AT1 deaktiviert werden

Aktivieren der Achs-Transformation an AT1 mit:

COORD(i) Die Nummer i (i=1,...,20) verweist auf einen der 20 Achs-Transformationssätze in den Maschinenparametern (siehe Dokumentation "IndraMotion MTX 14VRS Maschinenparameter").

Die Programmierung von COORD(i) löst folgende Aktionen aus:

1. Die Achs-Transformation an AT1 mit der Nummer i wird aktiviert, sofern die Konfigurierung in den Maschinenparametern keine fehlerhaften Einträge enthält.
2. Koordinatennamen und -positionen im MCS und in allen nachfolgenden Koordinatensystemen werden entsprechend umgeschaltet (siehe Beispiele in der Dokumentation "IndraMotion MTX 14VRS Funktionsbeschreibung - Grundlagen", Kap. "Achsen, Koordinaten, Koordinatensysteme").

Aktivieren der Achs-Transformation an AT2 mit:

COORD(i) Die Nummer i (i=1,...,20) verweist auf einen der 20 Achs-Transformationssätze in den Maschinenparametern (siehe Dokumentation "IndraMotion MTX 14VRS Maschinenparameter").

Die Programmierung von COORD(i) löst folgende Aktionen aus:

1. Die Achs-Transformation mit der Nummer i wird aktiviert, sofern die Konfigurierung in den Maschinenparametern keine fehlerhaften Einträge enthält.
2. Unterstützt die aktivierte Achs-Transformation die Vektor- oder Tensororientierung, wird die entsprechende Orientierungs-NC-Funktion aktiviert.

3. Die Koordinatennamen und -Positionen im WCS werden entsprechend umgeschaltet.

Aktivieren der Achs-Transformation per Systemdatum:

Alternativ zu der oben beschriebenen Maschinenparameter-Nummer kann eine Achs-Transformation auch per Systemdatum aktiviert werden. Dazu existiert der steuerungsinterne Typ **SysAxTrafo_t**, der alle achstransformations-spezifischen Maschinenparameter enthält.

Der Strukturtyp SysAxTrafo_t ist in der Schemadatei

feprom/schemas/sdaxtrf.xsd definiert (siehe [Kap. 11.1.7 "Anzeige der aktiven Transformations-Daten" auf Seite 194](#)). Die zugehörige SD-Variable vom Typ SysAxTrafo_t wird vom WMH in der Datei **usrfep/SDDefMTB.xml** definiert (z. B. mit Namen MTBAxTrafo). Es kann sich dabei auch um ein Array vom Typ SysAxTrafo_t handeln.

Initialisierungswerte können in der Datei **usrfep/SDDefMTB.xml** abgelegt werden.

Beispiel:

Aktivieren der Achs-Transformation per Systemdatum

Innerhalb eines NC-Programms kann mit SD.MTBAxTrafo[3].LenParam[5] der fünfte Längenparameter des dritten Achstransformations-Parametersatzes angesprochen werden. Da dies sowohl lesend als auch schreibend möglich ist, hat man damit auch die Möglichkeit, Transformationsparameter online zu ändern.

Programmiert man COORD mit dem Namen einer Transformations-SD-Variablen, werden die Transformationsdaten aus dem entsprechenden Systemdatum entnommen.

COORD(<[SDTrfName]>) "SDTrf Name" ist der Name des Systemdatums vom Typ SysAxTrafo_t, das die zu aktivierenden Transformationsdaten enthält.

Abwählen einer Achs-Transformation:

COORD (0,<TrafoLocation>)]	Ausschalten der aktiven Transformation Der zweite Parameter "TrafoLocation" bezeichnet die Wirkstelle der auszuschaltenden Transformation, kann also die Werte 1 oder 2 annehmen. Default: TrafoLocation = 2
COORD(0)	Kurzform für das Ausschalten der aktiven Transformation 2
Beispiel:	
COORD(4)	;Aktivieren einer AT1
COORD(3)	;Aktivieren einer AT2
x.. y.. z..	
COORD(0)	;Ausschalten von AT2, identisch mit COORD(0,2)
COORD(0,1)	;Ausschalten von AT1
COORD ([SD.MTBAxTrafo[4]]	;Aktivieren der in dem Systemdatum SD.MTBSxTrafo[4] hinterlegten Achs-Transformation

Die Programmierung von COORD(0,1) löst folgende Aktionen aus:

1. Die zuvor aktive Achs-Transformation an AT1 wird deaktiviert.
2. Die Koordinatennamen und -Positionen im MCS und in allen nachfolgenden Koordinatensystemen werden auf ACS-Namen und ACS-Positionen zurückgeschaltet

Die Programmierung von COORD(0,2) löst folgende Aktionen aus:

1. Die aktive Achs-Transformation an AT2 wird deaktiviert.
2. Ist eine Orientierungs-NC-Funktion aktiv, wird sie deaktiviert.
3. Die Koordinatennamen und -Positionen im WCS werden auf MCS-Namen und MCS-Positionen zurückgeschaltet.



Das direkte Umschalten von einer Achs-Transformation an AT2 auf eine andere Achs-Transformation ohne vorheriges Deaktivieren ist möglich.

Programmbeispiel: Am Programmanfang sind beide Achs-Transformationen abgeschaltet:

N01 G1 YL10 YR10 U20 V30	;G1 wirkt im WCS.
	;Es gelten die WCS-Bezeichner YL, YR, U, V
N02 COORD(4)	;Einschalten Bipod-Transformation
N03 X20 Y30 Z40 psi0 B120 C120	;WCS-Bezeichner X, Y, Z, y, B, C
N04 G06 YL1.0 YR1.5	;G06 wirkt im ACS
N05 COORD(3)	;Einschalten 5-Achs-Transformation
N06 x10 y10 z10 phi20 theta30 U80	;WCS-Bezeichner x, y, z, φ, θ
N07 G76 X0 Y0 Z0	;MCS-Bezeichner X, Y, Z
N08 COORD(0,2)	;Ausschalten 5-Achs-Transformation
	; (Wirkstelle;AT2)
N09 X0 Y0	
N10 COORD(0,1)	;Ausschalten Bipod-Transformation
	; (Wirkstelle;AT1)
N11 YL0 YR10	

11.1.7 Anzeige der aktiven Transformations-Daten

Es gibt zwei MTX-interne SD-Variablen, in denen die Daten der (zum Zeitpunkt der Satzvorbereitung) aktiven Achs-Transformation abgespeichert sind:

- SysAxTrafo1 und
- SysAxTrafo2

Sie sind vom Typ "SysAxTrafo_t" und in der Datei "feprom/SDDef.xml" definiert. Es handelt sich um kanalbezogene SD-Variablen.

SysAxTrafo1 enthält die Daten der an der Wirkstelle 1 aktiven Achs-Transformation (AT1) und SysAxTrafo2 die Daten der an der Wirkstelle 2 aktiven Achs-Transformation (AT2). Ist keine Achs-Transformation aktiv, ist die Variable komplett genullt.

Der Strukturtyp SysAxTrafo_t der Systemdaten SysAxTrafo1 und SysAxTrafo_2 ist in der Schemadatei feprom/schemas/sdaxtrf.xsd definiert. Er beinhaltet die Achs-Transformationspezifischen Maschinenparameter 1030 00110, 1030 00120, 1030 00125, 1030 00140, 1030 00150, 1030 00160.

Im Detail hat SysAxTrafo_t folgendes Aussehen:

```
<?xml version="1.0"?>
<xs:schema xmlns:xs="http://www.w3.org/2001/XMLSchema"
  elementFormDefault="qualified" attributeFormDefault="unqualified">
  <xs:include schemaLocation="basic_ty.xsd"/>
  <xs:complexType name="SysAxTrafo_t">
    <xs:annotation>
      <xs:documentation>Konfigurationsdaten der Achs-Transformation
    </xs:documentation> <xs:documentation>Autor: KE</xs:documentation>
      <xs:documentation>Achtung!
        Werden hier Änderungen vorgenommen, muss gleichzeitig acb1sysd.h geändert
        werden.
      </xs:documentation>
    </xs:annotation>
    <xs:sequence>

      <xs:element name="AxTrafoNo" type="Int_t">
        <xs:annotation>
          <xs:documentation>Nummer der Achstransformation:
            0 keine Achstransformation aktiv
            1..20 z. B. Coord (4) aktiv(ieren)
          </xs:documentation>
        </xs:annotation>
      </xs:element>

      <xs:element name="Type" type="Int_t" <xs:annotation>
        <xs:documentation>Transformationstyp entspricht MP 1030 00110
        </xs:documentation>
      </xs:annotation>
    </xs:element>

    <xs:element name="AxisAssignment" type="Byte_t" minOccurs="0" maxOccurs="8">
      <xs:annotation>
        <xs:documentation>Systemachsen/-koordinaten der Transformation entspricht
        MP 1030 00120
        </xs:documentation>
      </xs:annotation>
    </xs:element>

    <xs:element name="AxisClassification" type="Int_t" minOccurs="0" maxOccurs="8">
      <xs:annotation>
        <xs:documentation>Achsklassifikation der Transformationsachsen entspricht
        MP 1030 00125
        </xs:documentation>
      </xs:annotation>
    </xs:element>
```

```

<xs:element name="AxZeroPos" type="Double_t" minOccurs="0" maxOccurs="8">
  <xs:annotation>
    <xs:documentation>Achspositionen der Referenzstellung, entspricht MP 1030 00130
    </xs:documentation>
  </xs:annotation>
</xs:element>

<xs:element name="LenParam" type="Double_t" minOccurs="0" maxOccurs="16">
  <xs:annotation>
    <xs:documentation>Längen- und Winkelparameter entspricht MP 1030 00140
    </xs:documentation>
  </xs:annotation>
</xs:element>

<xs:element name="TCSOrientation" type="Byte_t" minOccurs="0" maxOccurs="8">
  <xs:annotation>
    <xs:documentation>Referenzlage des Werkzeugkoordinatensystems entspricht
    MP 1030 00150
    </xs:documentation>
  </xs:annotation>
</xs:element>

<xs:element name="EpsilonRanges" type="Float_t">
  <xs:annotation>
    <xs:documentation>Epsilon-Umgebungen entspricht MP 1030 00160
    </xs:documentation>
  </xs:annotation>
</xs:element>
</xs:sequence>
</xs:complexType>
</xs:schema>

```

Beispiel:

SysAxTrafo_t

```

101 TYPCHAN3%   = SD.SysAxTrafo2[3].Type
102 TYPACTCH%   = SD.SysAxTrafo2.Type
103 AT1NO%      = SD.SysAxTrafo1[2].AxTrafoNo

```

```

TYPCHAN3%      liefert den Type der aktiven AT2 des dritten Kanals (entspricht
MP 1030 00110, z. B. 32322201).

TYPACTCH%      liefert den Type der aktiven AT2 des aktuellen Kanals.

```

AT1NO%

liefert die AxTrafoNo der aktiven AT1 des zweiten Kanals.

Dabei bedeutet:

AT1NO% = 0 keine Achstransformation aktiv

AT1NO% = 1..20 Coord (AT1NO%) aktiv (z. B. Coord (4)

AT1NO% = 1000 z. B. Coord (SD.MTBxAxTrafo[3]) aktiv.

Wurde die Achs-Transformation per SD-Variable aktiviert, wird immer die Nummer "1000" angezeigt

Die Variablen dürfen im NC-Programm nur lesend verwendet werden.

11.1.8 Übersicht Achs-Transformationen (tabellarisch)

Die folgende Tabelle gibt den Stand der aktuell verfügbaren Achs-Transformationen wieder:

Beschreibung	Kinematik-Typ	Typ	Achszahl	Wirkstelle	in Kapitel ...
Bipod-Stabkinematik	Bipod	3131001	4/3	1	Kap. 11.2 "Bipod-Achs-Transformationen" auf Seite 200
	Bipod	2121001	3/2	1	Kap. 11.2 "Bipod-Achs-Transformationen" auf Seite 200
SCARA	SCARA	2103011	3	1	Kap. 11.3 "Scara-Achs-Transformation" auf Seite 209
	SCARA	3113011	4	1	
LLR - Kartesisch	LLR	3021005	3	1	
Nichtorthogonale Achsen	nichtorthogonal	3030001	3	1	Kap. 11.5.1 "Variante 1" auf Seite 214
	nichtorthogonal	2020001	2	1	Kap. 11.5.2 "Variante 2" auf Seite 216
	nichtorthogonal	2020002	2	1	Kap. 11.5.3 "Variante 3" auf Seite 217
3-/4-Achs-Transformation SCARA auf Linearachse	LRR	2112001	3	2	Kap. 11.6 "4-Achs-Transformation - Typ 1: SCARA auf Linearachse" auf Seite 223
	LLRR	3122001	4	2	Kap. 11.6 "4-Achs-Transformation - Typ 1: SCARA auf Linearachse" auf Seite 223
3-Achs-Transformation LLR (Hohlkugelfräsen)	LLR	2121002	3	2	Kap. 11.17 "3-Achs-transformation LLR" auf Seite 330
3-Achs-Transformation LLR - Kartesisch	LLR	3021004	3	2	

Achs-Transformation

Beschreibung	Kinematik-Typ	Typ	Achsanzahl	Wirkstelle	in Kapitel ...
4-Achs-Transformation mit Schwenkachse	LLLR	3131002	4	2	Kap. 11.8 "4-Achs-Transformation - Typ 2: LLLR" auf Seite 247
4-Achs-Transformation mit Schwenktisch	RLLL	3131003	4	2	Kap. 11.9 "4-Achs-Transformation - Typ 3: RLLL" auf Seite 253
5-Achs-Transformation mit Schwenk-Drehachse LLRRR	LLRRR (Vektororientierung)	3232201	5	2	Kap. 11.10.2 "Variante 1 (mit Vektororientierung)" auf Seite 261
	LLRRR (lineare Orientierung)	3232101	5	2	Kap. 11.10.3 "Variante 2 (mit linearer Orientierung)" auf Seite 262
	LLRRR (lineare Bewegung der rotatorischen Achsen)	3032001	5	2	Kap. 11.10.4 "Variante 3 (mit linearer Bewegung der rotatorischen Achsen)" auf Seite 263
5-Achs-Transformation mit Rundtisch RLLLR	RLLLR (Vektororientierung)	3232202	5	2	Kap. 11.11 "5-Achs-Transformation - Typ 2: RLLLR (mit Rundtisch)" auf Seite 278
	RLLLR (lineare Orientierung)	3232102	5	2	Kap. 11.11 "5-Achs-Transformation - Typ 2: RLLLR (mit Rundtisch)" auf Seite 278
5-Achs-Transformation mit Schwenk-/Rundtisch RRLLL	RRLLL (Vektororientierung)	3232203	5	2	Kap. 11.12 "5-Achs-Transformation - Typ 3: RRLLL (mit Schwenk-/Rundtisch)" auf Seite 288
	RRLLL (lineare Orientierung)	3232103	5	2	Kap. 11.12 "5-Achs-Transformation - Typ 3: RRLLL (mit Schwenk-/Rundtisch)" auf Seite 288
	RRLLL (minderbestückt)	3030013	3 (5)	2	Kap. 11.15 "5-Achs-Transformation - Typ 5: Minderbestückte RRLLL (mit Schwenk-/Rundtisch)" auf Seite 310

Beschreibung	Kinematik-Typ	Typ	Achszahl	Wirkstelle	in Kapitel ...
Kardanische 5-Achs-Transformation mit Schwenk-Drehachse LLLLR	LLLLR (Vektororientierung)	3232211	5	2	Kap. 11.13 "5-Achs-Transformation - Typ 4: Kardanischer Kopf " auf Seite 298
	LLLLR (lineare Orientierung)	3232111	5	2	Kap. 11.14 "5-Achs-Transformation - Typ 4.1: Kardanischer Kopf mit linearer Orientierungsbewegung" auf Seite 308
Kardanische 5-Achs-Transformation mit Rundtisch RLLLL	RLLLL (Vektororientierung)	3232212	5	2	Kap. 11.16 "5-Achs-Transformation - Typ 6: Kardanische RLLLL " auf Seite 320
	RLLLL (lineare Orientierung)	3232112	5	2	Kap. 11.16 "5-Achs-Transformation - Typ 6: Kardanische RLLLL " auf Seite 320
6-Achs-Transformation	LLLLRR (Eulerwinkel-Orientierung)	3333301	6	2	Kap. 11.18.2 "Variante 1 (mit Tensor-Orientierung) " auf Seite 337
	LLLLRR (lineare Bewegung der rotatorischen Achsen)	3033001	6	2	Kap. 11.18.3 "Variante 2 (mit linearer Bewegung der Rotationsachsen) " auf Seite 338
3-Achs-Zylindermantel-Transformation	Abwicklung	3021001	3	2	Kap. 11.19.2 "Variante 1 (Abwicklung) " auf Seite 353
	Projektion	3021002	3	2	Kap. 11.19.3 "Variante 2 (Ebenenprojektion) " auf Seite 355
4-Achs-Zylindermantel-Transformation	Abwicklung	3131021	4	2	Kap. 11.20 "4-Achs-Zylindermantel-Transformation " auf Seite 362
	Projektion	3131022	4	2	Kap. 11.20 "4-Achs-Zylindermantel-Transformation " auf Seite 362
5-Achs-Zylindermantel-Transformation	Abwicklung Linear-Orientierung	3232121	5	2	Kap. 11.21 "5-Achs-Zylindermantel-Transformation" auf Seite 370

Achs-Transformation

Beschreibung	Kinematik-Typ	Typ	Achsanzahl	Wirkstelle	in Kapitel ...
	Abwicklung Vektor-Orientierung	3232221	5	2	Kap. 11.21 "5-Achs-Zylindermantel-Transformation" auf Seite 370
	Projektion Linear-Orientierung	3232122	5	2	Kap. 11.21 "5-Achs-Zylindermantel-Transformation" auf Seite 370
	Projektion Vektor-Orientierung	3232222	5	2	Kap. 11.21 "5-Achs-Zylindermantel-Transformation" auf Seite 370
Stirnseitentransformation	X-C-Achs-Kinematik für Stirnseite	2011001	2	2	Kap. 11.22 "Stirnseiten-Transformation" auf Seite 379
	X-C-Achs-Kinematik für Stirnseite und zus. Z-Achse	3021003	3	2	Kap. 11.22 "Stirnseiten-Transformation" auf Seite 379

Tab. 11-2: Übersicht Achs-Transformationen

Derzeit können 20 verschiedene Parametersätze einer Achs-Transformation pro Maschine **gleichzeitig** in MP abgelegt und mit COORD(<i=1..20>) aufgerufen werden.

11.2 Bipod-Achs-Transformationen

11.2.1 Beschreibung

Erläuterung zu 3- und 4-Achs-Bipod

Name der Transformation	Transformationstyp	Wirkstelle	Koordinatensystem
4-Achs-Bipod	3131001	AT1	ACS, MCS
3-Achs-Bipod	2121001	AT1	ACS, MCS

Tab. 11-3: Bipod-Achs-Transformation

Beim **4-Achs-Bipod** (siehe nachfolgende Abbildung) sind zwei parallel zur MCS-Y-Koordinate verschiebbare Linearachsen (YL und YR) durch je ein Drehgelenk "A0" und "B0" mit den Elementen I1 und I2 verbunden (siehe nachfolgende Abbildung). Die Elementen I1 und I2 sind ihrerseits durch ein drittes Drehgelenk "A" verbunden. Die Drehachsen aller drei Drehgelenke stehen senkrecht zur XY-Ebene des MCS. Am Element I2 ist eine Spindel-C-Achse SC mit Werkzeug (z. B. Greifer) senkrecht zur Spindel-C-Achse stehenden XY-Ebene des MCS montiert. Durch Verschieben der Linearachsen YL und YR wird das Werkzeug in der XY-Ebene bewegt.

Die Zustellung des Werkzeugs erfolgt beim 4-Achs-Bipod durch Verschieben des gezeigten Aufbaus längs einer zur XY-Ebene senkrechten Linearachse Z (Z-Richtung des MCS).

Beim **3-Achs-Bipod** entfällt die Linearachse Z (Z-Richtung des MCS).

Sonst ist die Transformation identisch zu 3131001.

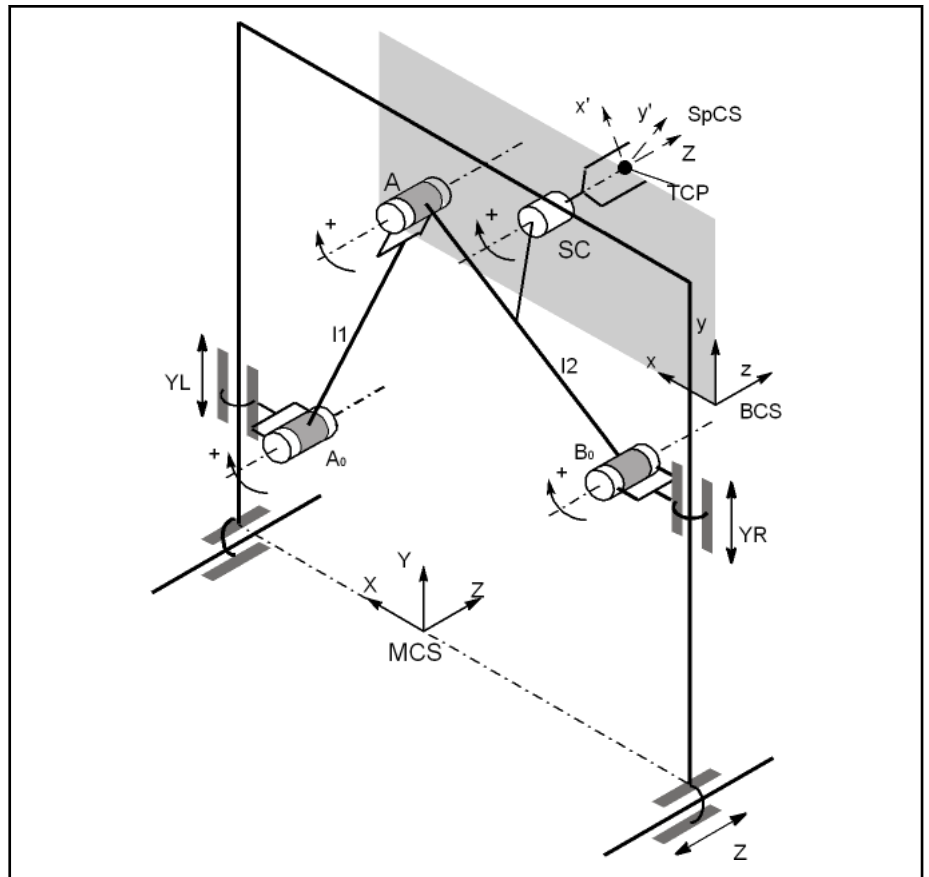


Abb. 11-2: 3- und 4-Achs-Bipod

Die Bipod-Transformation kennt zwei Betriebszustände:

- Mit Spindel-C-Achsnachführung
- Ohne Spindel-C-Achsnachführung

Das Werkzeug dreht sich während der Bewegung in der XY-Ebene **auch** um seine zur Z-Richtung parallele Längsachse.

Wird die Spindel-C-Achse nachgeführt, ist die Drehposition der Spindel-C-Achse eine Funktion der SC-Achsenposition in der XY-Ebene. Die Drehposition der Spindel-C-Achse ist auf den im Maschinenelement I2 festen Nullpunkt bezogen.

Für den Fall, dass sich die Spindel als C-Achse im Interpolationsverband des Kanals befindet, muss also die axiale Drehung des Werkzeugs durch eine entgegengesetzte Drehung der Spindel-C-Achse kompensiert werden.

Wird die Spindel-C-Achse nicht nachgeführt, ergibt sich die Drehposition der Spindel-C-Achse relativ zu ihrem im MCS festen Nullpunkt.

Die Beziehung der beiden Nullpunkte zueinander sind in MP 10300130 festgelegt.

Die Bipod-Transformation wählt bei ihrer Aktivieren automatisch den richtigen Betriebszustand aus. Wird die Spindel bei aktiver Bipod-Transformation in den Interpolationsverband des Kanals aufgenommen oder aus diesem entfernt, schaltet die Bipod-Transformation ihren Betriebszustand entsprechend um.

Relevante NC-Funktionen

COORD(i)	Mit i=1..20 wird eine Bipod-Transformation aktiviert, die in MP 1030 00110 definiert ist.
COORD(0,1)	Ausschalten der Bipod-Transformation (Wirkstelle AT1)

Tab. 11-4: Relevante NC-Funktionen

Relevante Maschinenparameter

1030 00110	definiert den Transformationstyp
1030 00120	Systemachsen der Transformation
1030 00130	Angabe der Achspositionen der Referenzstellung (z. B. der Achsen YL, YR, (SC), Z)
1030 00140	definiert Längen- und Winkelparameter (geometrische Beziehungen der Maschinenelementpositionen ermittelt bei der Kalibrierung der Referenzposition)

Tab. 11-5: Relevante Maschinenparameter

Die Achsen YL, YR, (Z), SC sind so zu positionieren, dass alle Koordinaten (im BCS) der Transformation den Wert Null erreichen ($X = Y = (Z) = C = 0$).

Referenzstellung

Die Achsen YL, YR, (Z), SC sind so zu positionieren, dass alle Koordinaten (im BCS) der Transformation den Wert Null erreichen ($X = Y = (Z) = C = 0$).

Die Bestimmung der Referenzstellung erfolgt mit der Kalibrierung des gesamten Systems (siehe [Kap. 11.2.2 "Handlungsanweisung: 3/4-Achs-Transformation Bipod"](#) auf Seite 202).

Mehrdeutige Achspositionen

Das Drehgelenk "A" muss sich immer oberhalb von Punkt "A₀" und oberhalb von Punkt "B₀" befinden. Dadurch entstehen keine Singularitäten und keine Mehrdeutigkeiten.

Werkzeugkorrektur

Für Achstransformationen an der Wirkstelle 1 (AT1) wird generell keine Werkzeugkorrektur verrechnet. Diese muss in der nachgeschalteten AT2 realisiert werden.

11.2.2 Handlungsanweisung: 3/4-Achs-Transformation Bipod

Applizieren: Das Bipod ist zunächst in eine Referenzstellung zu bringen, in der eine Kalibrierung durchgeführt wird. Die daraus gewonnenen Einstellungen für Längen- und Winkelparameter der Bipod-Maschinenelemente sind in die relevanten Maschinenparameter einzutragen.

Daten der Referenzstellung ermitteln:

Die nachfolgende Abbildung zeigt beispielhaft die geometrischen Abhängigkeiten der Bipodkinematik. Sie dient als Orientierung zur Kalibrierung und zum Feststellen der Referenzstellung des Bipods.

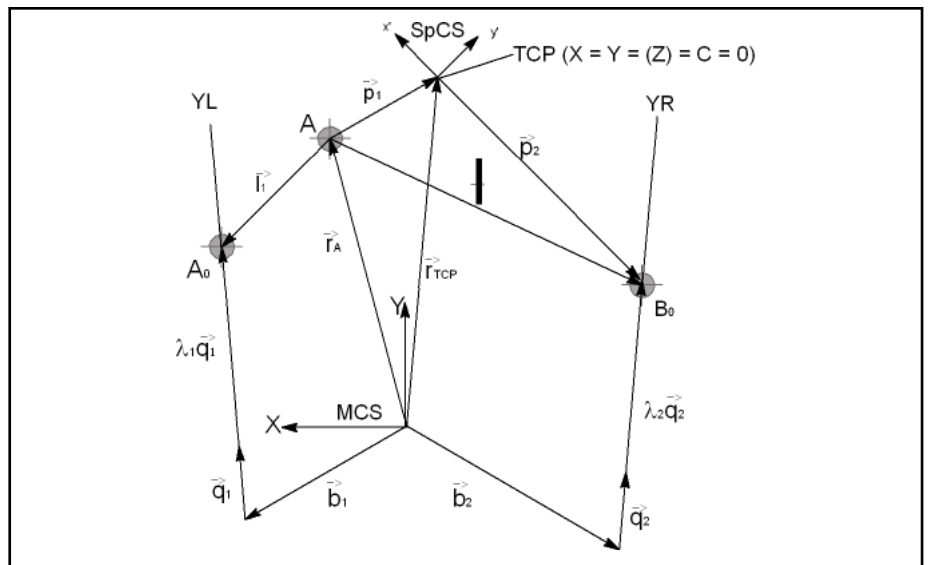


Abb. 11-3: Daten der Referenzstellung ermitteln

1. Mit den Achsen YL und YR den Tool Center Point (TCP) eines geeigneten Werkzeugs in die gewünschte Referenzposition bringen. An dieser Position soll später bei eingeschalteter Transformation folgendes gelten: Koordinaten ($X = Y = (Z) = C = 0$).

2. Die Achspositionen der Achsen YL und YR werden aus den Längen- und Winkelparameter (unter Punkt 3. bis 6.) ermittelt.

Bedingung ist, dass die Achsen YL und YR Nullpunkt-Positionen einnehmen.

3. $\vec{q}_1, \vec{q}_2$

Richtungsvektor der linken und rechten Führung (YL und YR) im MCS bestimmen (nur bei Nichtparallelität der beiden Linearachsen!)

4. $\vec{b}_1, \vec{b}_2$

Nullpunkte der Bewegungslinien der Gelenkmittelpunkte A_0 und B_0 im MCS bestimmen = Abstand MCS-Nullpunkt zu den Nullpunkt-Positionen der Achsen YL und YR.

Die Koordinaten der Gelenkmittelpunkte A_0 und B_0 im MCS ergeben

sich aus den Achspositionen λ_1 und λ_2 zu $\lambda_1 \vec{q}_1$ bzw. $\lambda_2 \vec{q}_2$.

Die Achspositionen $\lambda_1(0)$ und $\lambda_2(0)$, für die der TCP den Koordinaten-Nullpunkt erreicht ($r_{TCP} = 0$) ergeben sich intern aus den Kalibrierparametern.

5. $\vec{p}_1, \vec{p}_{2x'}$

Bestimmen der Koordinaten der Gelenkmittelpunkte A und B_0 bezogen auf das Spindelkasten-Koordinatensystem SpCS ($x'y'z'$ -Koordinatensystem).

Achs-Transformation

Für $\vec{p_2}$ im SpCS gilt stets die y' -Komponente $\vec{p_{2y'}}$ = 0, daher ist nur die x' -Komponente $\vec{p_{2x'}}$ von Bedeutung.

$$\left| \vec{l_1} \right|$$

6.

Abstand der Gelenkmittelpunkte A_0 und A der Koppellemente bestimmen.

7. Da die Linearachsen YL und YR auf denjenigen Achspositionen stehen, für die Koordinaten $x = y = 0$ sind, muss nun die Position der Z-Achse (nur für 4-Achs-Bipod) und die Position der SC-Achse ermittelt werden, für die die Koordinaten $(X = Y = (Z) = C = 0)$ einnehmen werden. Hierzu muss das gesamte System ggf. in Z-Richtung verschoben und die SC-Achse um die C-Achse gedreht werden.

Die ermittelten Werte sind in die MP einzutragen (siehe folgendes Beispiel):

Die Achstransformation kann über die relevanten Maschinenparameter appliziert werden. Dazu muss die Funktion Achstransformation über den Parameter **tra** "Achstransformationen" im Setup (SUP) aktiviert werden. Der Pfad des MP-Namens ist immer "TRA/AxTrafo[1..20]/...".

Alternativ dazu kann die Achstransformation über ein benutzerdefiniertes Systemdatum vom Typ SysAxTrafo_t appliziert werden.

Die beiden Varianten sind inhaltlich identisch.

Die Achspositionen der Referenzstellung müssen ermittelt werden.

In der nachfolgenden Tabelle sind jeweils die äquivalenten MP-Nummern, MP-Name und SD-Elemente angegeben. Um eine Transformation zu applizieren, muss eine der Varianten entsprechend besetzt werden.

1030 00110	Transformationstyp	
	MP-Name: TrafoType	
	SD-Elem. : Type	
1030 00120	Transfor- mation AT1:	3131001 (4-Achs-Bipod) 2121001 (3-Achs-Bipod)
	Systemachsen der 4-Achs-Bipod-Transformation:	
	MP-Name: FwdInCoordIndTrafo[1..8]	
	SD-Elem. : AxisAssignment[1..8]	
	[1]	Systemachsindex der YL-Achse
	[2]	Systemachsindex der YR-Achse
	[3]	Systemachsindex der Z-Achse
	[4]	Systemachsindex der Spindel-C-Achse SC
	[5..8]	nicht relevant

1030 00120	Systemachsen der 3-Achs-Bipod-Transformation:	
	MP-Name: FwdInCoordIndTrafo[1..8]	
	SD-Elem. : AxisAssignment[1..8]	
	[1]	Systemachsindex der YL-Achse
	[2]	Systemachsindex der YR-Achse
1030 00130	[3]	Systemachsindex der Spindel-C-Achse
	[4..8]	nicht relevant
	Achspositionen der Referenzstellung für 4-Achs-Bipod	
	MP-Name: RefPosTrafo[1..8]	
	SD-Elem. : AxZeroPos[1..8]	
	[1]	nicht relevant, da aus MP 1030 00140 hervorgehend
1030 00130	[2]	nicht relevant, da aus MP 1030 00140 hervorgehend
	[3]	Position für Z0
	[4]	Position für SC0
	[5..16]	nicht relevant
	Achspositionen der Referenzstellung für 3-Achs-Bipod	
	MP-Name: RefPosTrafo[1..8]	
1030 00130	SD-Elem. : AxZeroPos[1..8]	
	[1]	nicht relevant, da aus MP 1030 00140 hervorgehend
	[2]	nicht relevant, da aus MP 1030 00140 hervorgehend
	[3]	Position für SC0
	[4..16]	nicht relevant
1030 00130	Die erforderlichen Achspositionen YL und YR unter [1], [2] sind durch die in MP 1030 00140 nachfolgend beschriebenen Längen- und Winkelparameter bestimmt, so dass [1] und [2] nicht ausgewertet werden! 4-Achs-Bipod: Unter [3] ist die Position der Z-Achse und unter [4] die Position der SC-Achse einzutragen, für die die Koordinaten ($X = Y = (Z) = C = 0$) werden und, wenn die Achsen YL, YR bereits die hierfür erforderlichen Positionen einnehmen, d.h. wenn die x- und die y-Koordinate bereits auf Null stehen. 3-Achs-Bipod: Unter [3] ist die Position SC-Achse einzutragen, für die die Koordinaten ($X = Y = C = 0$) werden und, wenn die Achsen YL, YR bereits die hierfür erforderlichen Positionen einnehmen, d.h. wenn die x- und die y-Koordinate bereits auf Null stehen.	

Achs-Transformation

1030 00140	Längen- und Winkelparameter	
	MP-Name: JointParTrafo[1..16]	
	SD-Elem. : LenParam[1..16]	
	Die oben gezeigten Vektoren $\vec{q}_i$... werden anteilig bzgl. der Koordinaten von MCS und SpCS hier eingetragen:	
	Werte der Parameter $q_1, q_2, b_1, b_2, p_{1x}, p_{1y}, p_{2x}, p_{2y}$ aus der Maschinenkalibrierung der Referenzstellung:	
	[1]	q_{1x}
	[2]	q_{1y}
	[3]	q_{2x}
	[4]	q_{2y}
	[5]	b_{1x}
	[6]	b_{1y}
	[7]	b_{2x}
	[8]	b_{2y}
	[9]	$p_{1x'}$
	[10]	$p_{1y'}$
7080 00020	[11]	$p_{2x'}$
	[12]	l_1
	[13..16]	nicht relevant
	Optionale Namen der Systemkoordinaten bei aktiver 4-Achs-Bipod:	
	[1]	z.B.: X
	[2]	z.B.: Y
	[3]	z.B.: Z
	[4]	z.B.: C
	[5 .. 6]	nicht relevant
	Optionale Namen der Systemkoordinaten bei aktiver 3-Achs-Bipod:	
	[1]	z.B.: X
	[2]	z.B.: Y
	[3]	nicht relevant
	[4]	z.B.: C
	[5 .. 6]	nicht relevant

Tab. 11-6: Daten der Referenzstellung ermitteln

Aktivieren: Voraussetzungen:

Die Aktivieren der Bipod-Transformation setzt eine Kalibrierung zur Feststellung der Referenzstellung voraus.

Programmierung der Spindel-C-Achsnachführung:

Die Bipod-Transformation ist bereits beim Hochlauf aktiv und die SC-Achse wird mit Spindel-C-Achsnachführung ausgeführt, d.h. sie befindet sich im Interpolationsverband des Kanals.

Beispiel einer Programmierung:

```

M3 S5000                ;Spindel im Drehzahlbetrieb
                        ;(die SC-Achse wird nicht nachgeführt)

M5
...
SpindleToAxis(SC)       ;Umschalten der SC-Achse in den C-Achsmode
WAIT
WaitAxis(SC)           ;Wartet, bis SC-Achse vom Kanal übernommen wurde
...
G01 C45 F100           ;Bearbeitung wird durchgeführt
                        ;(SC-Achse wird nachgeführt)
...
RemAxis(SC)            ;SC-Achse aus dem Kanal freigeben
AxisToSpindle(SC)      ;Umschalten der SC-Achse in den Spindel-Mode
...
                        ;Bearbeitung wird durchgeführt
                        ;(SC-Achse wird nicht nachgeführt)

```



Die Spindel-C-Achse muss sich im Positionierbetrieb (C-Achsbetrieb) befinden, damit sie in den Interpolationsverband eines Kanals aufgenommen wird.

Spindel Positionieren (M19):

Auch wenn sich die **Spindel** im Drehzahlbetrieb befindet, benötigt **sie** eine Spindel-Nachführung, wenn mit M19 positioniert werden soll.

Beispiel:

```

M19 SC54 G0 X7 Y2      Anfahren der Ziel-Position X = 7, Y = 2, Z = 9 und gleich-
Z9                     zeitig Spindel so abbremesen, dass sie in Winkelposition
                        C = 54 (= 54°) zum Stehen kommt.

```

Beide Prozesse werden zu Beginn des Satzes gleichzeitig gestartet, beenden sie jedoch in der Regel nicht gemeinsam.

Beim 4-Achs-Bipod sind zwei parallel zur Y-Achse verschiebbare Linearachsen durch je ein Drehgelenk mit Elementen verbunden. Die Elemente sind ihrerseits durch ein drittes Drehgelenk verbunden. Die Drehachsen aller 3 Drehgelenke stehen senkrecht zur XY-Ebene des MCS. An einem Element ist eine Spindel-C-Achse mit Werkzeug (z.B. Greifer) senkrecht zur Spindel-C-Achse stehenden XY-Ebene des MCS montiert. Durch Verschieben der Linearachsen wird das Werkzeug in der XY-Ebene bewegt. Die Zustellung des Werkzeugs erfolgt beim 4-Achs-Bipod durch Verschieben längs einer zur XY-Ebene senkrechten Linearachse (Z-Richtung des MCS).

Beim 3-Achs-Bipod entfällt die senkrechte Linearachse (Z-Richtung des MCS). Sonst ist die Transformation identisch zur 4-Achs-Bipod-Transformation.

- Nullpunkte der Bewegungslinien der Gelenkmittelpunkte A0 und B0 im MCS bestimmen = Abstand MCS-Nullpunkt zu den Nullpunkt-Positionen der Achsen YL und YR. Die Koordinaten der Gelenkmittelpunkte A0 und B0 im MCS ergeben sich aus den Achspositionen. Die Achspositionen, für die der TCP den Koordinaten-Nullpunkt erreicht ($r_{TCP} = 0$) ergeben sich intern aus den Kalibrier-Parametern.
- Bestimmen der Koordinaten der Gelenkmittelpunkte A und B0 bezogen auf das Spindelkasten-Koordinatensystem SpCS (x'y'Z-Koordinatensystem).
Für p2 im SpCS gilt stets y'-Komponente $p2Y = 0$, daher ist nur die x'-Komponente p2X von Bedeutung.
- Abstand der Gelenkmittelpunkte A0 und A der Koppellemente bestimmen.
- Weil die Linearachsen YL und YR auf denjenigen Achspositionen stehen, für die Koordinaten $x = y = 0$ sind, muss nun die Position der Z-Achse (nur für 4-Achs-Bipod) und die Position der SC-Achse ermittelt werden, für die Koordinaten ($X = Y = Z = C = 0$) einnehmen werden. Hierzu muss das gesamte System ggf. in Z-Richtung verschoben und die SC-Achse um die C-Achse gedreht werden.



- Mit **COORD(i)** oder **CRD(i)** wird die Bipod-Achs-Transformation aktiviert. Ein einmal eingestellter Zweig bleibt bis zum Ausschalten der Achs-Transformation erhalten
- Per Befehl **SpindleToAxis (STA)** und **AxisToSpindle (ATS)** kann die Spindel/C-Achse in einer Spindel und wieder zurück umgeschaltet werden.
- Über die Hilfsfunktion M19 kann die Spindel positioniert werden.
- Mit **COORD(0,1)** oder **CRD(0,1)** wird die Bipod-Achs-Transformation deaktiviert.

Deaktivieren **COORD(0,1)**

schaltet die Achstransformation im NC-Programm aus.

11.3 Scara-Achs-Transformation

11.3.1 Beschreibung

Funktion

Name der Transformation	Transformationstyp	Wirkstelle	Koordinatensystem
3-Achs-SCARA (ohne Z-Achse)	2103011	AT1	ACS, MCS
4-Achs-SCARA (mit Z-Achse)	3113011		

Tab. 11-7: SCARA-Achs-Transformation

Drei miteinander verbundene Rundachsen A, B und C bewegen ein Werkzeug in der XY-Ebene des Maschinenkoordinatensystems (MCS) auf eine bestimmte Position mit einer vorgegebenen Orientierung PHI.

PHI ist der Winkel zwischen I_3 -Arm und MCS-X.

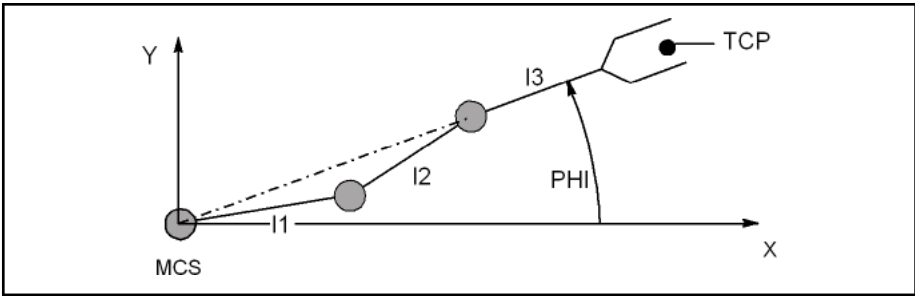


Abb. 11-6: SCARA-Transformation

Die Scara-Transformation kann wahlweise mit und ohne Z-Achse (Linearachse) verwendet werden. Bei Vorhandensein einer Z-Achse wird diese eins zu eins auf die z-Koordinate durchgereicht.

Relevante NC-Funktionen

COORD(<i>)	mit i=1..20; aktiviert die SCARA-Achs-Transformation, die in MP 1030 00110 definiert ist.
COORD(0,1)	Die aktive SCARA-Transformation wird deaktiviert.

Tab. 11-8: Relevante NC-Funktionen

Relevante Maschinenparameter

1030 00110	definiert den Transformationstyp
1030 00120	Definiert die an der Transformation beteiligten Systemachsen/-koordinaten der Transformation.
1030 00130	definiert die Achspositionen der Referenzstellung
1030 00140	Definiert die Armlängen der Rundachsen l_1 , l_2 und l_3 und den vektoriellen Abstand der ersten Rundachse zum MCS-Nullpunkt ($\vec{l}_0$).

Tab. 11-9: Relevante Maschinenparameter

Referenzstellung

Die 3 Rundachsen sind parallel zur Z-Koordinate des MCS angeordnet. Der Längenvektor $\vec{l}_0$ und die Längen l_1 , l_2 und l_3 definieren die exakten Referenzpositionen a_0 , a_1 , a_2 der 3 Rundachsen in der XY-Ebene des MCS. Der Vektor $\vec{l}_0$ definiert die Lage a_0 der ersten Rundachse bezüglich des Ursprungs im MCS.

Für die Längen gilt:

l_1 , l_2 und $l_3 > 0$.

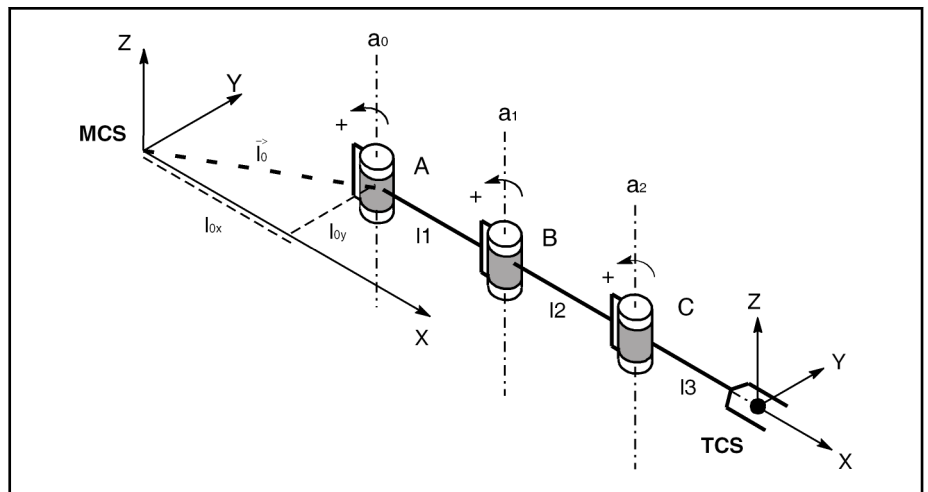


Abb. 11-7: Referenzstellung SCARA

Mehrdeutige Achspositionen

Eine Werkzeugposition (TCP-Position) kann mit zwei unterschiedlichen Positionen der Rundachsen und ihrer I_1 , I_2 -Arme erreicht werden:

Linksknick: Transformationszweig -1

Rechtsknick: Transformationszweig 1

Ein Übergang von Zweig -1 auf 1 und umgekehrt ist bei eingeschalteter SCARA-Transformation nicht möglich. Der Zweig ist beim Aktivieren der Transformation festgelegt und bleibt bis zum Ausschalten erhalten.

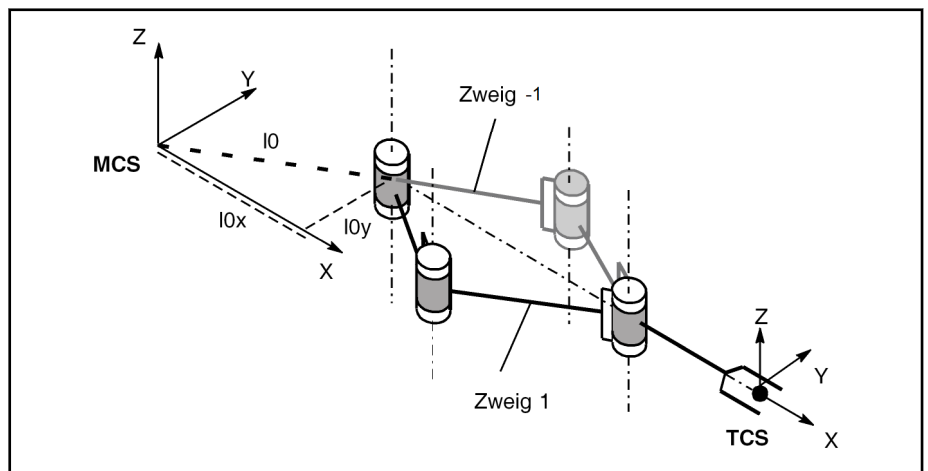


Abb. 11-8: Übergang von Zweig -1 auf Zweig 1

Beispiel:

Ein Rechtsknick (Transformationszweig 1) führt dazu, dass die Achsen gegen das Hindernis fahren. In diesem Fall darf nur der Transformationszweig -1 gewählt werden, um Position 2 anfahren zu können.

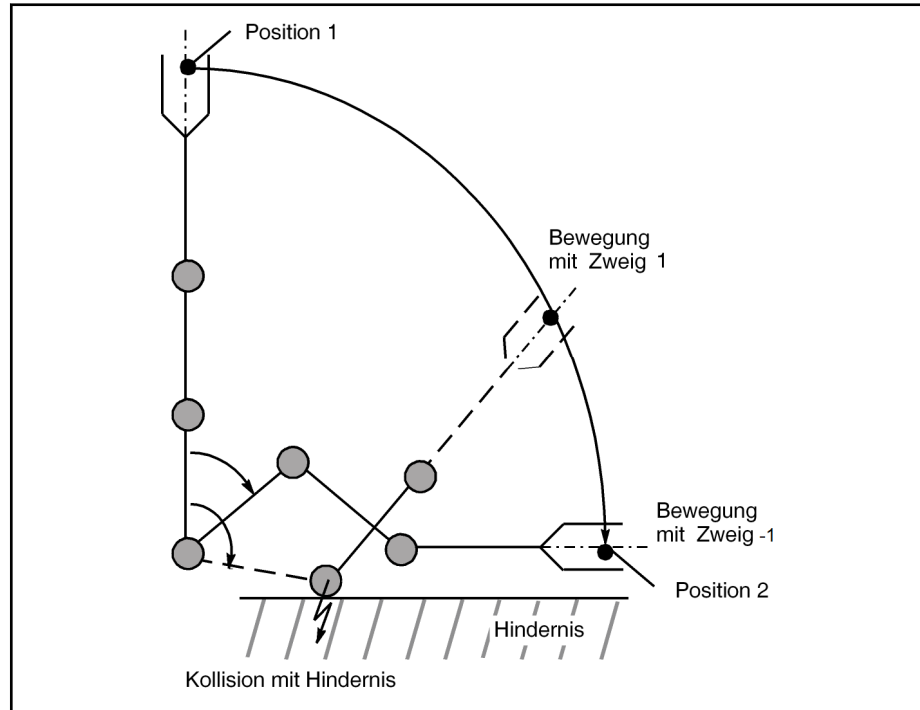


Abb. 11-9: *Beispiel - Rechtsknick*

Werkzeugkorrektur

Für Achstransformationen an der Wirkstelle 1 (AT1) wird generell keine Werkzeugkorrektur verrechnet. Diese muss in der nachgeschalteten AT2 realisiert werden.

11.4 Handlungsanweisung: 3-/4-Achs-Transformation SCARA

Applizieren Die Achstransformation kann über die relevanten Maschinenparameter appliziert werden. Dazu muss die Funktion Achstransformation über den Parameter **tra** "Achstransformationen" im Setup (SUP) aktiviert werden. Der Pfad des MP-Namens ist immer "TRA/AxTrafo[1..20]/...".

Alternativ dazu kann die Achstransformation über ein benutzerdefiniertes Systemdatum vom Typ SysAxTrafo_t appliziert werden.

Die beiden Varianten sind inhaltlich identisch.

Die Achspositionen der Referenzstellung müssen ermittelt werden.

In der nachfolgenden Tabelle sind jeweils die äquivalenten MP-Nummern, MP-Name und SD-Elemente angegeben. Um eine Transformation zu applizieren, muss eine der Varianten entsprechend besetzt werden.

Transformations-relevante Parameter

MP-Nummer	Beschreibung	
1030 00110	Transformationstyp MP-Name: TrafoType SD-Elem. : Type	
	2103011	3-Achs-Scara
	3113011	4-Achs-Scara
1030 00120	Systemachsen/-koordinaten der Transformation MP-Name: FwdInCoordIndTrafo[1..8] SD-Elem. : AxisAssignment[1..8]	
	[1]	Systemachsnummer von A
	[2]	Systemachsnummer von B
	[3]	Systemachsnummer von C
	[4]	Systemachsnummer von Z (falls vorhanden)
	[5..8]	nicht relevant
1030 00130	Achspositionen der Referenzstellung MP-Name: RefPosTrafo[1..8] SD-Elem. : AxZeroPos[1..8]	
	[1]	A ⁰
	[2]	B ⁰
	[3]	C ⁰
	[4]	Z ⁰ (falls vorhanden)
	[5..8]	nicht relevant
1030 00140	Längen- und Winkelparameter MP-Name: JointParTrafo[1..16] SD-Elem. : LenParam[1..16]	
	Definiert die Längenvektoren l_1, l_2 in [mm].	
	[1]	l_{0x}
	[2]	l_{0y}
	[3]	l_1
	[4]	l_2
	[5]	l_3
	[6..16]	nicht relevant

Tab. 11-10: Transformations-relevante Parameter

Achs-Transformation

Aktivieren *Ist die Transformation konfiguriert per*

- Maschinenparameter, dann wird die Transformation im NC-Programm aktiviert mit (i=1,..20):

COORD(<i>)

- Systemdatum, dann wird die Transformation im NC-Programm aktiviert mit (Typ "SysAxTrafo_t"):

COORD(<SD.MtbAxTrafo>)

Programmbeispiel:

Die Transformation ist im 4. Maschinenparametersatz definiert. Am Programmmanfang ist keine Achstransformation aktiv:

Transformation aktivieren

```

N100 G1 F1000 A10 B10 C30      ;The axes A, B and C are traversed
N110 COORD(4)                  ;Switching on the axis transformation
N130 X20 Y30                   ;Traversing the coordinates X and Y
N140 X10 Y10 PHI45             ;Traversing the coordinates X, Y and PHI
      :
N999 COORD(0,1)                ;Switching-off the axis transformation
                                ;Axes are again programmed from here

```

Deaktivieren **COORD(0,1)**

schaltet die Achstransformation im NC-Programm aus.

11.5 Nichtorthogonale 2- und 3-Achs-Transformation

11.5.1 Variante 1

Funktion

Name der Transformation	Transformationstyp	Wirkstelle	Koordinatensystem
3-Achsen - nichtorthogonal	3030001	AT1	ACS, MCS

Tab. 11-11: Funktion - Variante 1

Die nichtorthogonale 3-Achs-Transformation zeichnet sich dadurch aus, dass ihre Linearachsen Xa, Ya und Za nichtorthogonal, d. h. nicht rechtwinklig zueinander orientiert sind oder sein können.

Jede Achse wird durch ihre Polarkoordinaten φ (phi) und ϑ (theta) im MCS definiert:

- Xa-Achse: von φ_x und ϑ_x
- Ya-Achse: von φ_y und ϑ_y
- Za-Achse: von φ_z und ϑ_z

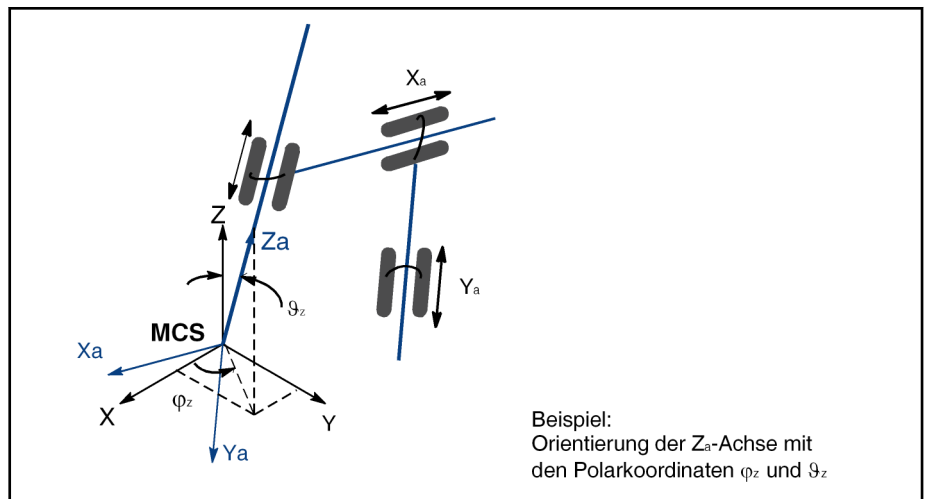


Abb. 11-10: Beispiel

Relevante NC-Funktionen

COORD(1) TRFOPT(< φ_x >, < ϑ_x >, < φ_y >, < ϑ_y >, < φ_z >, < ϑ_z >)	
	Mit $i=1..20$ wird eine, in MP 1030 00110 definierte, nichtorthogonale 2-Achs-Transformation aktiviert. Es besteht die Option, die in MP definierten Achsorientierungen beim Einschalten mit TRFOPT (..) zu verändern (ein nicht programmierter Winkel behält seinen MP-Wert).
COORD(0,1)	Die aktive Transformation wird deaktiviert.

Tab. 11-12: Relevante NC-Funktionen

Relevante Maschinenparameter

1030 00110	Definiert den Transformationstyp.
1030 00120	Definiert die an der Transformation beteiligten Systemachsen/-koordinaten der Transformation.
1030 00130	Definiert die Achspositionen der Referenzstellung.
1030 00140	Definiert die Achsrichtungen der nichtorthogonalen Achsen im MCS.

Tab. 11-13: Relevante Maschinenparameter

Referenzstellung

In der Referenzstellung der Achs-Transformation befindet sich der TCP im MCS-Nullpunkt. Die Achspositionen sind in diesem Fall X_{a0} , Y_{a0} und Z_{a0} .

Werkzeugkorrektur

Für Achstransformationen an der Wirkstelle 1 (AT1) wird generell keine Werkzeugkorrektur verrechnet. Diese muss in der nachgeschalteten AT2 realisiert werden.

11.5.2 Variante 2

Funktion

Name der Transformation	Transformationstyp	Wirkstelle	Koordinatensystem
2-Achsen - nichtorthogonal	2020001	AT1	ACS, MCS

Tab. 11-14: Funktion - Variante 2

Die Kinematik der nichtorthogonalen 2-Achs-Transformation besteht aus zwei Linearachsen X_a und Y_a , die nichtorthogonal, d. h. nicht rechtwinklig zueinander orientiert sind oder sein können.

Die Orientierung der Linearachsen wird durch ihre Polarkoordinaten φ (phi) auf die MCS-X-Koordinate bezogen:

- X_a -Achse: φ_x
- Y_a -Achse: φ_y

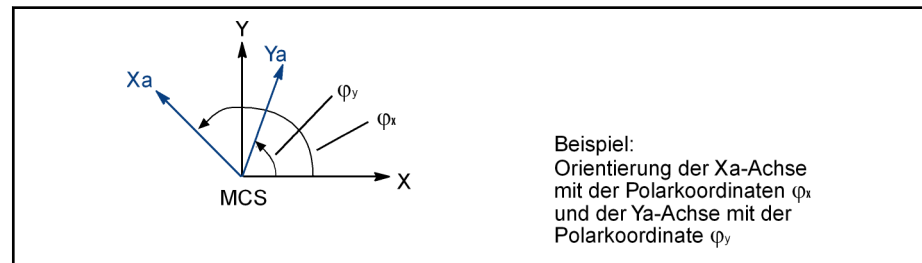


Abb. 11-11: Beispiel

Relevante NC-Funktionen

COORD(<i>) TRFOPT(< φ_x >,< φ_y >):	
	Mit $i=1..20$ wird eine in MP 1030 00110 definierte, nichtorthogonale 2-Achs-Transformation aktiviert. Es besteht die Option, die in MP definierten Achsorientierungen beim Einschalten mit TRFOPT (..) zu verändern (ein nicht programmierter Winkel behält seinen MP-Wert).
COORD(0,1)	Die aktive Transformation wird deaktiviert.

Tab. 11-15: Relevante NC-Funktionen

Relevante Maschinenparameter

1030 00110	Definiert den Transformationstyp.
1030 00120	Definiert die an der Transformation beteiligten Systemachsen/-koordinaten der Transformation.
1030 00130	Definiert die Achspositionen der Referenzstellung
1030 00140	Definiert die Achsrichtungen der nichtorthogonalen Achsen im MCS.

Tab. 11-16: Relevante Maschinenparameter

Referenzstellung

In der Referenzstellung der Achs-Transformation befindet sich der TCP im MCS-Nullpunkt. Die Achspositionen seien in diesem Fall X_{a0} und Y_{a0} .

Werkzeugkorrektur

Für Achstransformationen an der Wirkstelle 1 (AT1) wird generell keine Werkzeugkorrektur verrechnet. Diese muss in der nachgeschalteten AT2 realisiert werden.

11.5.3 Variante 3

Funktion

Name der Transformation	Transformationstyp	Wirkstelle	Koordinatensystem
2-Achsen - nichtorthogonal	2020002	AT1	ACS, MCS

Tab. 11-17: Funktion - Variante 3

Die Kinematik der nichtorthogonalen 2-Achs-Transformation besteht aus zwei Linearachsen X_a und Z_a , die nichtorthogonal, d. h. nicht rechtwinklig zueinander orientiert sein können.

Die Orientierung der Linearachsen wird jeweils durch ihre Polarkoordinaten ϑ (theta) auf die MCS-Z-Koordinate bezogen:

- X_a -Achse: ϑ_x
- Z_a -Achse: ϑ_z

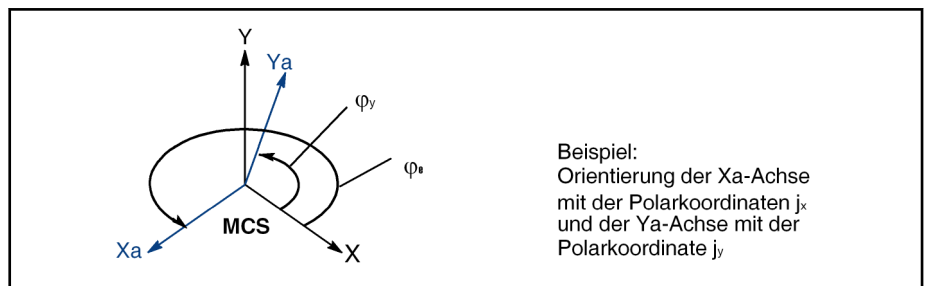


Abb. 11-12: Beispiel

Relevante NC-Funktionen

COORD(<i>) TRFOPT(< ϑ_x >,< ϑ_z >):	
	Mit $i=1..20$ wird eine, in MP 1030 00110 definierte, nichtorthogonale 2-Achs-Transformation aktiviert. Es besteht die Option, die in MP definierten Achsorientierungen beim Einschalten mit TRFOPT (..) zu verändern (ein nicht programmierter Winkel behält seinen MP-Wert).
COORD(0,1)	Die aktive Transformation wird deaktiviert.

Tab. 11-18: Relevante NC-Funktionen

Relevante Maschinenparameter

1030 00110	Definiert den Transformationstyp.
1030 00120	Definiert die an der Transformation beteiligten Systemachsen/-koordinaten der Transformation.
1030 00130	Definiert die Achspositionen der Referenzstellung.
1030 00140	Definiert die Achsrichtungen der nichtorthogonalen Achsen im MCS.

Tab. 11-19: Relevante Maschinenparameter

Referenzstellung

In der Referenzstellung der Achs-Transformation befindet sich der TCP im MCS-Nullpunkt. Die Achspositionen seien in diesem Fall Xa_0 und Ya_0 .

Werkzeugkorrektur

Für Achstransformationen an der Wirkstelle 1 (AT1) wird generell keine Werkzeugkorrektur verrechnet. Diese muss in der nachgeschalteten AT2 realisiert werden

11.5.4 Handlungsanweisung: Nichtorthogonale 2/3-Achs-Transformation

Applizieren: Die Achstransformation kann über die relevanten Maschinenparameter appliziert werden. Dazu muss die Funktion Achstransformation über den Parameter **tra** "Achstransformationen" im Setup (SUP) aktiviert werden. Der Pfad des MP-Namens ist immer "TRA/AxTrafo[1..20]/...".

Alternativ dazu kann die Achstransformation über ein benutzerdefiniertes Systemdatum vom Typ SysAxTrafo_t appliziert werden.

Die beiden Varianten sind inhaltlich identisch.

Die Achspositionen der Referenzstellung müssen ermittelt werden.

In der nachfolgenden Tabelle sind jeweils die äquivalenten MP-Nummern, MP-Name und SD-Elemente angegeben. Um eine Transformation zu applizieren, muss eine der Varianten entsprechend besetzt werden.

Es gibt insgesamt 3 Varianten der nichtorthogonalen Achs-Transformation, deren Parameter alle nachfolgend beschrieben sind.

Variante 1 (nichtorthogonale 3-Achs-Transformation)

An folgenden Beispielen soll eine Applizieren einer nichtorthogonalen 3-Achs-Transformation (3030001) gezeigt werden.

1030 00110	Transformationstyp	
	MP-Name: TrafoType SD-Elem. : Type 3030001	
1030 00120	Systemachsen/-koordinaten der Transformation	
	MP-Name: FwdInCoordIndTrafo[1..8] SD-Elem. : AxisAssignment[1..8]	
	[1]	Systemachsnummer der Xa-Achse
	[2]	Systemachsnummer der Ya-Achse
	[3]	Systemachsnummer der Za-Achse
	[4..8]	nicht relevant
1030 00130	Achspositionen der Referenzstellung	
	MP-Name: RefPosTrafo[1..8] SD-Elem. : AxZeroPos[1..8]	
	[1]	Position für Xa^0
	[2]	Position für Ya^0
	[3]	Position für Za^0
	[4..8]	nicht relevant

1030 00140	Längen- und Winkelparameter	
	MP-Name: JointParTrafo[1..16]	
	SD-Elem. : LenParam[1..16]	
	Definiert die Achsrichtungen im MCS, Einheit in Grad.	
	[1]	φ_x
	[2]	ϑ_x
	[3]	φ_y
	[4]	ϑ_y
	[5]	φ_z
	[6]	ϑ_z
	[7..16]	nicht relevant

Tab. 11-20: Applizieren - Variante 1

An folgenden Beispielen wird die Einstellung der Winkelparameter einer nichtorthogonalen 3-Achs-Transformation gezeigt:

Beispiel:

Einstellung der Winkelparameter einer nichtorthogonalen 3-Achsfunktion - Beispiel 1

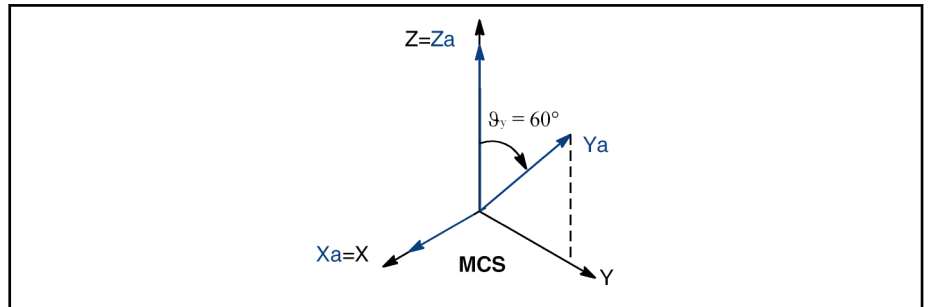


Abb. 11-13: Beispiel 1

Die Achsrichtungen Xa und Za fallen mit den kartesischen MCS-Koordinaten X und Z zusammen.

Ya liegt in der YZ-Ebene. Die Winkel der Achsrichtungen in 103000140 sind wie folgt einzustellen

Achs-Transformation

1030 00140

Einheiten in Grad

[1] $\varphi_x = 0$

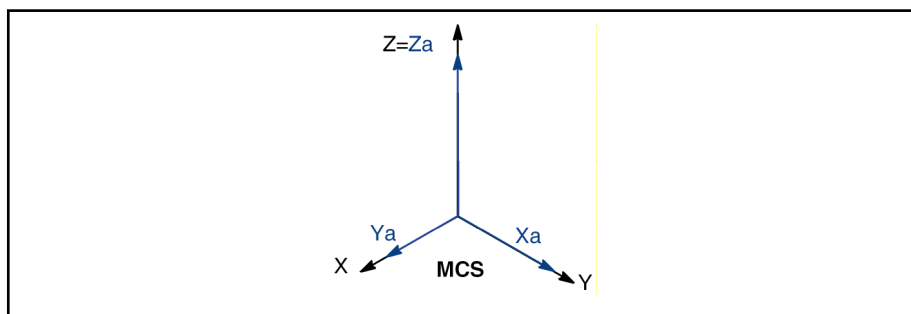
[2] $\vartheta_x = 90$

[3] $\varphi_y = 90$

[4] $\vartheta_y = 60$

[5] $\varphi_z = \text{beliebig}$

[6] $\vartheta_z = 0$

*Beispiel:*Einstellung der Winkelparameter einer nichtorthogonalen 3-Achsfunktion -
Beispiel 2Abb. 11-14: *Beispiel 2*

Hier wird die Achs-Transformation benutzt, um von einem linkshändigen kartesischen Achsenkoordinatensystem (ACS) auf ein rechtshändiges kartesisches Maschinenkoordinatensystem (MCS) zu transformieren. Die Winkel sind in diesem Fall:

1030 00140

Einheiten in Grad

[1] $\varphi_x = 90$

[2] $\vartheta_x = 90$

[3] $\varphi_y = 0$

[4] $\vartheta_y = 90$

[5] $\varphi_z = \text{beliebig}$

[6] $\vartheta_z = 0$

Variante 2 (nichtorthogonale 2-Achs-Transformation in XY-Ebene)

An folgendem Beispiel soll eine Applizieren einer nichtorthogonalen 2-Achs-Transformation (2020001) gezeigt werden.

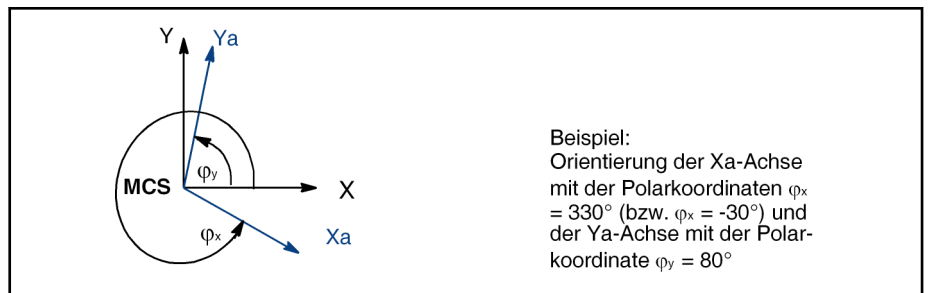


Abb. 11-15: Applizieren - Variante 2

Daten der Referenzstellung ermitteln:

Die Referenzstellung setzt voraus, dass die Koordinatenorientierung aller beteiligten Achsen und die Achspositionen der Referenzstellung bekannt sind.

Dazu müssen die folgenden MP gesetzt werden (Beispiel):

1030 00110	Transformationstyp	
	MP-Name: TrafoType SD-Elem. : Type	
	2020001	
1030 00120	Systemachsen/-koordinaten der Transformation	
	MP-Name: FwdInCoordIndTrafo[1..8] SD-Elem. : AxisAssignment[1..8]	
	[1]	Systemachsnummer der Xa-Achse
	[2]	Systemachsnummer der Ya-Achse
	[3..8]	nicht relevant
1030 00130	Achspositionen der Referenzstellung	
	MP-Name: RefPosTrafo[1..8] SD-Elem. : AxZeroPos[1..8]	
	[1]	Position für Xa ⁰
	[2]	Position für Ya ⁰
	[3..8]	nicht relevant
1030 00140	Längen- und Winkelparameter	
	MP-Name: JointParTrafo[1..16] SD-Elem. : LenParam[1..16] Definiert die Achsrichtungen im MCS, Einheit in Grad.	
	[1]	φ_x
	[2]	φ_y
	[3..16]	nicht relevant

Tab. 11-21: Applizieren - Variante 2

Variante 3 (nichtorthogonale 2-Achs-Transformation in XZ-Ebene)

An folgendem Beispiel soll eine Applizieren einer nichtorthogonale n2-Achs-Transformation (2020002) gezeigt werden.

Achs-Transformation

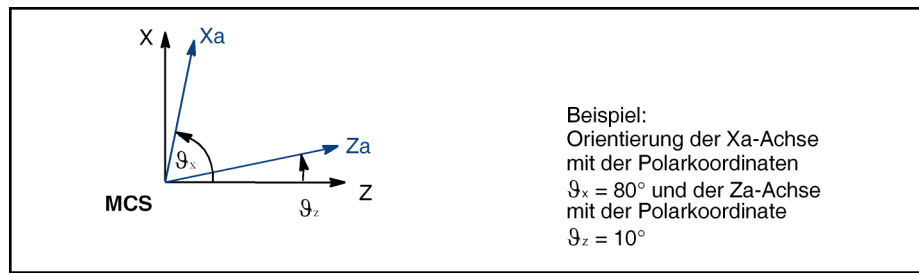


Abb. 11-16: Applizieren - Variante 3

Daten der Referenzstellung ermitteln:

Die Referenzstellung setzt voraus, dass die Koordinatenorientierung aller beteiligten Achsen und die Achspositionen der Referenzstellung bekannt sind.

Dazu müssen die folgenden MP gesetzt werden (Beispiel):

1030 00110	Transformationstyp	
	MP-Name: TrafoType SD-Elem. : Type 2020002	
1030 00120	Systemachsen/-koordinaten der Transformation	
	MP-Name: FwdInCoordIndTrafo[1..8] SD-Elem. : AxisAssignment[1..8]	
	[1]	Systemachsnummer der Xa-Achse
	[2]	Systemachsnummer der Za-Achse
1030 00130	Achspositionen der Referenzstellung	
	MP-Name: RefPosTrafo[1..8] SD-Elem. : AxZeroPos[1..8]	
	[1]	Position für Xa^0
	[2]	Position für Za^0
1030 00140	Längen- und Winkelparameter	
	MP-Name: JointParTrafo[1..16] SD-Elem. : LenParam[1..16] Definiert die Achsrichtungen im MCS, Einheit in Grad.	
	[1]	θ_x
	[2]	θ_z
	[3..16]	nicht relevant

Tab. 11-22: Applizieren - Variante 3

Aktivieren Ist die Transformation konfiguriert per

- Maschinenparameter, dann wird die Transformation im NC-Programm aktiviert mit (i=1,..20):

COORD(<i>)

- Systemdatum, dann wird die Transformation im NC-Programm aktiviert mit (Typ "SysAxTrafo_t"):

COORD(<SD.MtbAxTrafo>)

Programmbeispiel:

Die Transformation ist im 4. Maschinenparametersatz definiert. Am Programmmanfang ist keine AchsTransformation aktiv:

Transformation aktivieren

```
N100 G1 F1000 XA10 YA10 ZA5 ;axes XA, YA and ZA are traversed
N110 COORD(4) TRFOPT(...) ;Switching-on the axis transformation
N130 X20 Y30 Z10 ;Cart. coordinates X,Y and Z are traversed
N140 X10 Y50 Z15 ;Cart. coordinates X,Y and Z are traversed
      :
N999 COORD(0,1) ;Switching-off the axis transformation
      ;Axes are again programmed from here
```

Deaktivieren **COORD(0,1)**

schaltet die Achstransformation im NC-Programm aus.

11.6 4-Achs-Transformation - Typ 1: SCARA auf Linearachse

11.6.1 Beschreibung

Funktion

Bei dieser Transformation handelt es sich um einen SCARA-Arm auf einer Linearachse. Sie ist als 3-Achs- und als 4-Achsvariante realisiert.

- 3-Achsvariante in (y,z,φ)
- 4-Achsvariante in (x,y,z,φ)

Die wesentlichen Transformationseigenschaften sind bei beiden Varianten identisch. Bei der 4-Achsvariante wird zusätzlich die x-Koordinate (inklusive x-Werkzeuglängenkorrektur) mit berücksichtigt. Die nachfolgenden Skizzen und Beschreibungen beziehen sich auf die 4-Achsvariante.

Name der Transformation	Transformationstyp	Wirkstelle	Koordinatensystem
SCARA auf Linearachse LLRR	3122001	AT2	MCS, BCS
SCARA auf Linearachse LRR	2112001	AT2	MCS, BCS

Tab. 11-23: Funktion - 3-Achsvariante SCARA auf Linearachse

Ausgangsgrößen im MCS: (X),Z,R₁,R₂

Eingangsgrößen im BCS: (x),y,z,φ

Die Konfiguration besitzt zwei (eine) Linearachsen (X) und Z. Darauf sitzen zwei rotatorische Achsen R₁ und R₂. Die Drehachse R₁ schwenkt die Achse R₂, R₂ schwenkt das Werkzeug. Die Achsen R₁ und R₂ drehen beide um X (Typ A-Achse). Die y-Koordinate ergibt sich durch gleichzeitiges Verfahren der Z- und R₁-Achse. Die Achse R₂ dient der Orientierung.

- Der Orientierungsvektor "p" wird berechnet (in Polarkoordinaten).
- Eine lineare Orientierungsbewegung in "φ" ist möglich.
- Eine Online-Korrektur ist für alle Koordinaten möglich.

Achs-Transformation

- Die rotatorischen Achsen können als Endlos- oder als Rundachsen definiert werden.

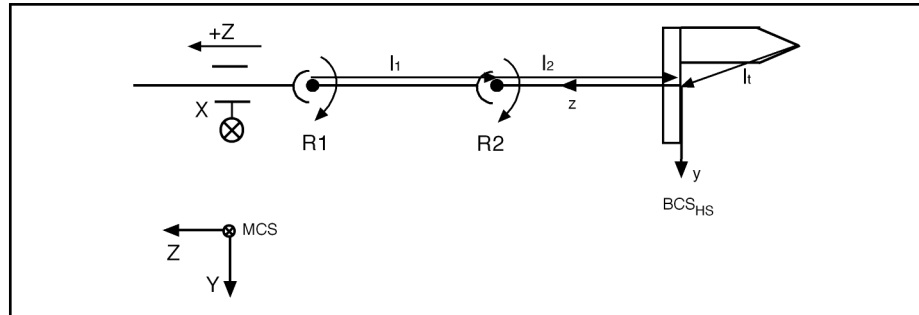


Abb. 11-17: Referenzstellung der Hauptspindel-Konfiguration

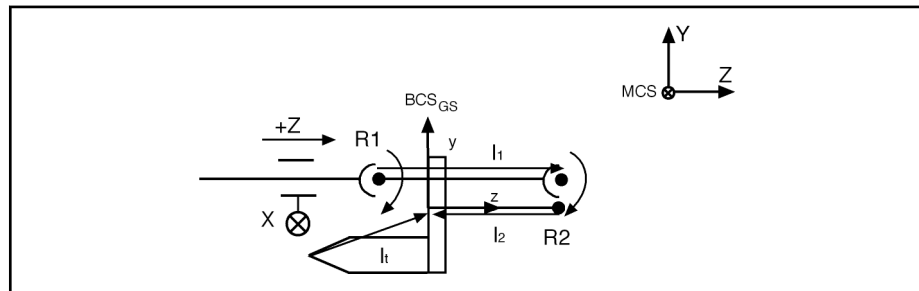


Abb. 11-18: Referenzstellung der Gegenspindel-Konfiguration

Die Vektoren $\vec{l}_1$ und $\vec{l}_2$ definieren die Geometrie der 4-Achs-Kinematik. Sie beschreiben die Armlängen der Kinematik und haben die Form $\vec{l}_1 = (0,0,l_{1x})$ und $\vec{l}_2 = (0,0,l_{2x})$.

$\vec{l}_1$ ist ein Vektor entlang z von der R₁-Achse zur R₂-Achse. $\vec{l}_2$ ist der $\vec{l}_1$ -Fortsetzungsvektor von der R₂-Achse zum Nullpunkt des BCS (d. h. $l_{1z} \neq 0$, $l_{2z} \neq 0$, $l_{1x} = l_{1y} = l_{2x} = l_{2y} = 0$).

Besonderheiten: Wertebereich:

Der Orientierungswinkel ϕ ist auf $\pm 89^\circ$ begrenzt.

Mehrdeutigkeiten:

Während der Bearbeitung ist der Verfahrbereich der Achse R₁ sowohl bei der Haupt- als auch bei der Gegenspindel auf $\pm 89^\circ$ begrenzt. Somit ergeben sich keine Mehrdeutigkeiten der Achs-Transformation.

Die Verfahrbereiche der übrigen Achsen sind unbegrenzt.

Werkzeugrevolver:

Der Werkzeug-Revolver wird von einer NC-Achse angesteuert. Für jede Werkzeugposition muss eine eigene Achs-Transformation definiert werden. Diese unterscheiden sich nur in der Achsposition der Referenzstellung der Achse R₂ (Maschinenparameter 1030 00130 [4]). Bei einem Wechsel auf eine neue Revolverposition muss zunächst die Achse R₂ auf ihre Referenzposition gefahren werden, anschließend wird die entsprechende Achs-Transformation aktiviert.

Ist R₂ als Endlosachse konfiguriert, müssen Werte von 0..359 in der Referenzstellung eingetragen werden.

Haupt- /Gegenspindel:

Haupt- und Gegenspindel können mit der gleichen Achs-Transformation realisiert werden. Sie unterscheiden sich nur in der Parametrisierung.

Es gilt:

Hauptspindel: $l_{1z} < 0, l_{2z} < 0, Z = +3$

Gegenspindel: $l_{1z} > 0, l_{2z} < 0, Z = -3$

Referenzstellung: Die Abb. 11-17 "Referenzstellung der Hauptspindel-Konfiguration" auf Seite 224 zeigt die Referenzstellung der Hauptspindel-Konfiguration und die Abb. 11-18 "Referenzstellung der Gegenspindel-Konfiguration" auf Seite 224 zeigt die Referenzstellung der Gegenspindel-Konfiguration. Das heißt, nimmt die Maschine die in der Abbildung dargestellte Stellung ein, müssen (bei $l_t=0$) die BCS-Koordinaten $(x,y,z,\varphi) = (0,0,0,0)$ sein, bei zugehörigen MCS-Koordinaten $(X,Z,R_1,R_2) = (X^0,Z^0,R_1^0,R_2^0)$ aus MP 1030 00130. Die positiven Dreh-/Verfahrrichtungen der Achsen sind ebenfalls in der Abbildung dargestellt. Diese sind entsprechend einzustellen.

In dieser Stellung steht die Spitze des rotationssymmetrischen Werkzeugs (Tool Center Point TCP) auf dem Ursprung des Basiskoordinatensystems (BCS) und ist entlang der z-Koordinate ausgerichtet. Die Ausrichtung des BCS ist durch die Achs-Transformation eindeutig festgelegt.

Werkzeugkorrektur:

$$\vec{l}_t = \begin{pmatrix} l_x \\ l_y \\ l_z \end{pmatrix}$$

Eine Werkzeugkorrektur ist möglich. Der Korrekturvektor $\vec{l}_t$ zeigt von der Werkzeugspitze zur Werkzeugaufnahme.

Relevante NC-Funktionen

COORD ()	Die 4-Achs-Transformation LLRR wird aktiviert. Dies kann wahlweise per Maschinenparametersatz ($i=1..20$) oder per MTB-Systemdatum (vom Typ SysAxTrafo_t) erfolgen.
COORD (0,2)	Die aktive Achstransformation wird deaktiviert.

Tab. 11-24: Relevante NC-Funktionen

11.6.2 Handlungsanweisung

Applizieren Die Achstransformation kann über die relevanten Maschinenparameter appliziert werden. Dazu muss die Funktion Achstransformation über den Parameter **tra** "Achstransformationen" im Setup (SUP) aktiviert werden. Der Pfad des MP-Namens ist immer "TRA/AxTrafo[1..20]/...".

Alternativ dazu kann die Achstransformation über ein benutzerdefiniertes Systemdatum vom Typ SysAxTrafo_t appliziert werden.

Die beiden Varianten sind inhaltlich identisch.

Die Achspositionen der Referenzstellung müssen ermittelt werden.

In den beiden nachfolgenden Tabellen sind jeweils die äquivalenten MP-Nummern, MP-Name und SD-Elemente angegeben. Um eine Transformation zu applizieren, muss eine der Varianten entsprechend besetzt werden. Die erste Tabelle beschreibt die 3-Achs-Variante LRR und die zweite Tabelle beschreibt die 4-Achs-Variante LLRR.

3-Achs-Variante LRR: Transformations-relevante Parameter

MP-Nummer	Beschreibung	
1030 00110	Transformationstyp MP-Name: TrafoType SD-Elem. : Type	
	2112001	Scara auf Linearachse LRR
1030 00120	Systemachsen/-koordinaten der Transformation MP-Name: FwdInCoordIndTrafo[1..8] SD-Elem. : AxisAssignment[1..8]	
	[1]	Systemachsnummer von Z
	[2]	Systemachsnummer von R ₁ (A-Achse)
	[3]	Systemachsnummer von R ₂ (A-Achse)
	[4..8]	nicht relevant

MP-Nummer	Beschreibung																				
1030 00125	Achsklassifikationen der Transformationsachsen MP-Name: CoordClass[1..8] MP-Name: CoordDir[1..8] SD-Elem. : AxisClassification[1..8] Definiert die an der Transformation beteiligten Achsen inklusive Bewegungsrichtung. - MP-Nr. oder SD-Element: Wert ">0" bedeutet: Die Achse verfährt/dreht positiv im Sinne der obigen Abbildungen. Wert "<0" bedeutet: Die Achse verfährt/dreht negativ im Sinne der obigen Abbildungen. <i>Mögliche Werte sind:</i> +/-1 = X-Achse +/-2 = Y-Achse +/-3 = Z-Achse +/-100 = A-Achse +/-200 = B-Achse +/-300 = C-Achse - MP-Name: <i>Mögliche Werte für CoordClass sind:</i> - X,Y,Z (linear) - A,B,C (rotatorisch) <i>Mögliche Werte für CoordDir sind:</i> - positiv (bzgl. Abb. "Referenzstellung") - negativ (bzgl. Abb. "Referenzstellung") Ist ein Parameter unbesetzt, gelten folgende Default-Einstellungen: <table><tr><th></th><th>MP-Nr / SD-Elem.:</th><th>CoordClass:</th><th>CoordDir:</th></tr><tr><td>[1]</td><td>3</td><td>Z</td><td>positiv</td></tr><tr><td>[2]</td><td>100</td><td>A</td><td>positiv</td></tr><tr><td>[3]</td><td>100</td><td>A</td><td>positiv</td></tr><tr><td>[4..8]</td><td>nicht relevant</td><td>0</td><td>positiv</td></tr></table>		MP-Nr / SD-Elem.:	CoordClass:	CoordDir:	[1]	3	Z	positiv	[2]	100	A	positiv	[3]	100	A	positiv	[4..8]	nicht relevant	0	positiv
	MP-Nr / SD-Elem.:	CoordClass:	CoordDir:																		
[1]	3	Z	positiv																		
[2]	100	A	positiv																		
[3]	100	A	positiv																		
[4..8]	nicht relevant	0	positiv																		

Achs-Transformation

MP-Nummer	Beschreibung
1030 00130	Achspositionen der Referenzstellung MP-Name: RefPosTrafo[1..8] SD-Elem. : AxZeroPos[1..8]
	[1] Z^0
	[2] R_1^0
	[3] R_2^0
	[4..8] nicht relevant
1030 00140	Längen- und Winkelparameter MP-Name: JointParTrafo[1..16] SD-Elem. : LenParam[1..16] Definiert die Längenvektoren l_1, l_2 in [mm].
	[1] ohne Bedeutung
	[2] ohne Bedeutung
	[3] Armlänge 1: Hauptspindel $l_{1z} < 0$, Gegenspindel $l_{1z} > 0$
	[4] ohne Bedeutung
	[5] ohne Bedeutung
	[6] Armlänge 2: Hauptspindel und Gegenspindel $l_{2z} < 0$
	[7..16] nicht relevant
1030 00160	Epsilon-Umgebungen
	ohne Bedeutung

Tab. 11-25: Transformations-relevante Parameter

4-Achs-Variante LLRR: Transformations-relevante Parameter

MP-Nummer	Beschreibung
1030 00110	Transformationstyp MP-Name: TrafoType SD-Elem. : Type
	3122001 Scara auf Linearachse LLRR
1030 00120	Systemachsen/-koordinaten der Transformation MP-Name: FwdInCoordIndTrafo[1..8] SD-Elem. : AxisAssignment[1..8]
	[1] Systemachsnummer von X
	[2] Systemachsnummer von Z
	[3] Systemachsnummer von R_1 (A-Achse)
	[4] Systemachsnummer von R_2 (A-Achse)
	[5..8] nicht relevant

MP-Nummer	Beschreibung																								
1030 00125	Achsklassifikationen der Transformationsachsen MP-Name: CoordClass[1..8] MP-Name: CoordDir[1..8] SD-Elem. : AxisClassification[1..8] Definiert die an der Transformation beteiligten Achsen inklusive Bewegungsrichtung. - MP-Nr. oder SD-Element: Wert ">0" bedeutet: Die Achse verfährt/dreht positiv im Sinne der obigen Abbildungen. Wert "<0" bedeutet: Die Achse verfährt/dreht negativ im Sinne der obigen Abbildungen. <i>Mögliche Werte sind:</i> +/-1 = X-Achse +/-2 = Y-Achse +/-3 = Z-Achse +/-100 = A-Achse +/-200 = B-Achse +/-300 = C-Achse - MP-Name: <i>Mögliche Werte für CoordClass sind:</i> - X,Y,Z (linear) - A,B,C (rotatorisch) <i>Mögliche Werte für CoordDir sind:</i> - positiv (bzgl. Abb. "Referenzstellung") - negativ (bzgl. Abb. "Referenzstellung") Ist ein Parameter unbesetzt, gelten folgende Default-Einstellungen: <table><tr><th></th><th>MP-Nr / SD-Elem.:</th><th>CoordClass:</th><th>CoordDir:</th></tr><tr><td>[1]</td><td>1</td><td>X</td><td>positiv</td></tr><tr><td>[2]</td><td>3</td><td>Z</td><td>positiv</td></tr><tr><td>[3]</td><td>100</td><td>A</td><td>positiv</td></tr><tr><td>[4]</td><td>100</td><td>A</td><td>positiv</td></tr><tr><td>[5..8]</td><td>nicht relevant</td><td>0</td><td>positiv</td></tr></table>		MP-Nr / SD-Elem.:	CoordClass:	CoordDir:	[1]	1	X	positiv	[2]	3	Z	positiv	[3]	100	A	positiv	[4]	100	A	positiv	[5..8]	nicht relevant	0	positiv
	MP-Nr / SD-Elem.:	CoordClass:	CoordDir:																						
[1]	1	X	positiv																						
[2]	3	Z	positiv																						
[3]	100	A	positiv																						
[4]	100	A	positiv																						
[5..8]	nicht relevant	0	positiv																						

Achs-Transformation

MP-Nummer	Beschreibung	
1030 00130	Achspositionen der Referenzstellung	
	MP-Name: RefPosTrafo[1..8]	
	SD-Elem. : AxZeroPos[1..8]	
	[1]	X^0
	[2]	Z^0
	[3]	R_1^0
1030 00140	[4]	R_2^0
	[5..8]	nicht relevant
	Längen- und Winkelparameter	
	MP-Name: JointParTrafo[1..16]	
	SD-Elem. : LenParam[1..16]	
	Definiert die Längenvektoren l_1 , l_2 in [mm].	
	[1]	ohne Bedeutung
	[2]	ohne Bedeutung
	[3]	Armlänge 1: Hauptspindel $l_{1z} < 0$, Gegenspindel $l_{1z} > 0$
	[4]	ohne Bedeutung
1030 00160	[5]	ohne Bedeutung
	[6]	Armlänge 2: Hauptspindel und Gegenspindel $l_{2z} < 0$
	[7..16]	nicht relevant
	Epsilon-Umgebungen	
	ohne Bedeutung	

Tab. 11-26: Transformations-relevante Parameter

Aktivieren Ist die Transformation konfiguriert per

- Maschinenparameter, dann wird die Transformation im NC-Programm aktiviert mit (i=1,..20):
COORD(<i>)
- Systemdatum, dann wird die Transformation im NC-Programm aktiviert mit (Typ "SysAxTrafo_t"):
COORD(<SD.MtbAxTrafo>)

Programmbeispiel:

Die 3-Achs-Transformation LRR ist im vierten Maschinenparametersatz definiert. Am Programmmanfang ist keine Achstransformation aktiv:

Transformation aktivieren

```

N100 G1 F1000 Z10 A1=10 A2=30 ;Axes Z, A1 and A2 are traversed
N110 COORD(4) ;Switching on the axis transformation
N130 z20 y30 phi0 ;Coordinates z, y and phi are traversed
      :
N999 COORD(0) ;Switching-off the axis transformation
      ;Axes are again programmed from here

```

Deaktivieren COORD(0) bzw. COORD(0,2)
schaltet die Achstransformation im NC-Programm aus.

11.7 2-dimensionale Delta-Kinematik mit zusätzlicher Drehachse in der Spitze (RRR)

11.7.1 Beschreibung

Funktion

Name der Transformation	Transformationstyp	Wirkstelle
2-dimensionale Delta-Kinematik Zusätzliche Drehachse in der Spitze (RRR)	2103002	AT2

Die Basis dieser Transformation bildet eine 2-dimensionale Delta-Kinematik aus zwei zueinander parallelen rotatorischen Achsen R1 und R2, die über drei dazu parallele Drehgelenke G1, G2, G3 und die Verbindungselemente e11, e22, e1, e2 miteinander verbunden sind:

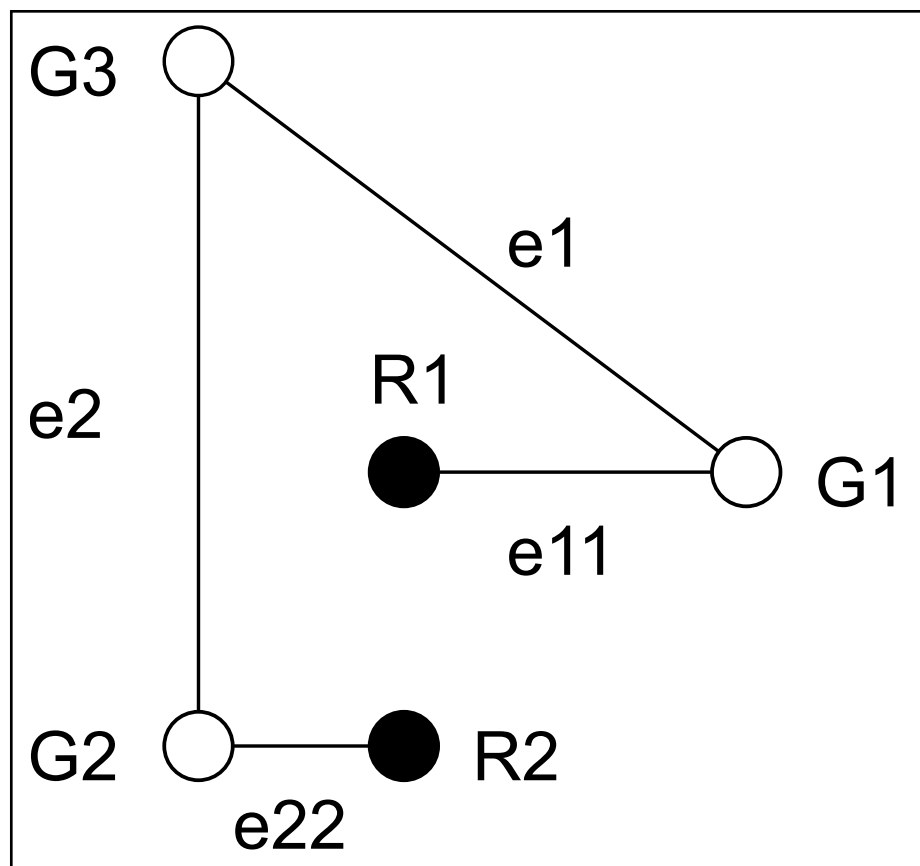


Abb. 11-19: Elementare 2-dimensionale Delta-Transformation

Durch Drehen von R1 und/oder R2 kann das mittlere Drehgelenk G3 innerhalb gewisser Grenzen an beliebige Positionen verschoben werden.

Zusätzlich ist konzentrisch mit dem Drehgelenk G3 eine weitere rotatorische Achse R3 installiert. Durch diese lässt sich das Werkzeug drehen. [Abb. 11-20 "2-dimensionale Delta-Transformation mit zusätzlicher rotatorischer Achse in der Spitze"](#) auf Seite 233 zeigt ein Beispiel.

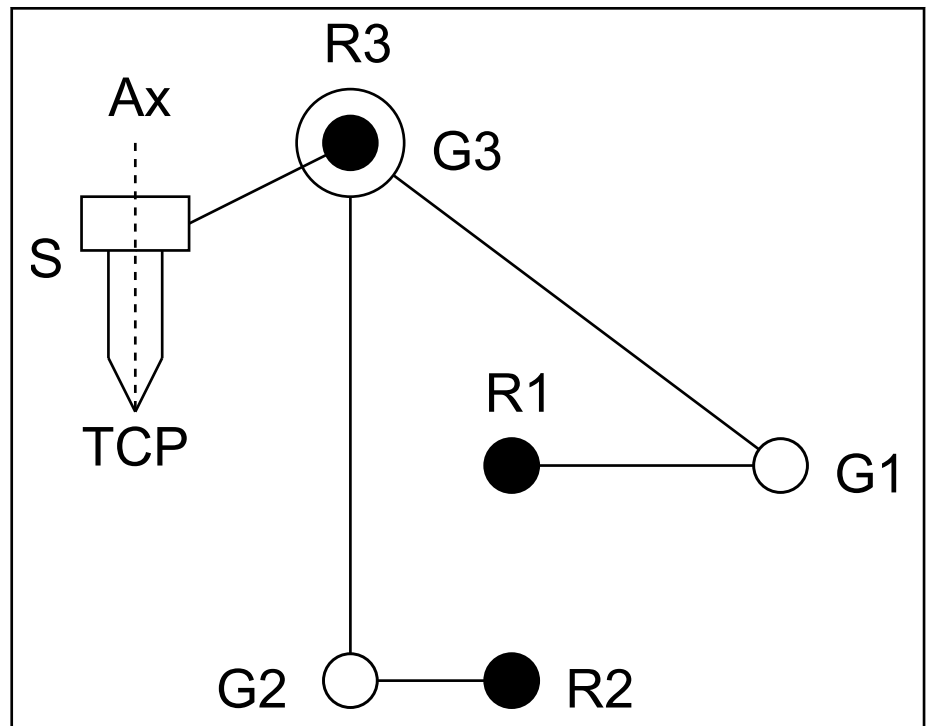


Abb. 11-20: 2-dimensionale Delta-Transformation mit zusätzlicher rotatorischer Achse in der Spitze

Die Achse R3 ist mechanisch so installiert, dass sich ihr Gehäuse durch Bewegung der Achsen R1 und/oder R2 zwar verschiebt, aber nicht dreht.

Die Spindelnase ist mit "S" gekennzeichnet, und die Werkzeugspitze (Tool Center Point) mit "TCP". Der Werkzeugreferenzpunkt ist der Schnittpunkt zwischen Werkzeugachse "Ax" und Spindelnase.

Abb. 11-21 "Auswirkung einer Drehung der Achsen R1 und R2, aber ohne Drehung der Achse R3" auf Seite 234 zeigt die in Abb. 11-20 "2-dimensionale Delta-Transformation mit zusätzlicher rotatorischer Achse in der Spitze" auf Seite 233 gezeigte Konfiguration nach einer Drehung der Achsen R1 und R2, aber ohne Drehung der Achse R3.

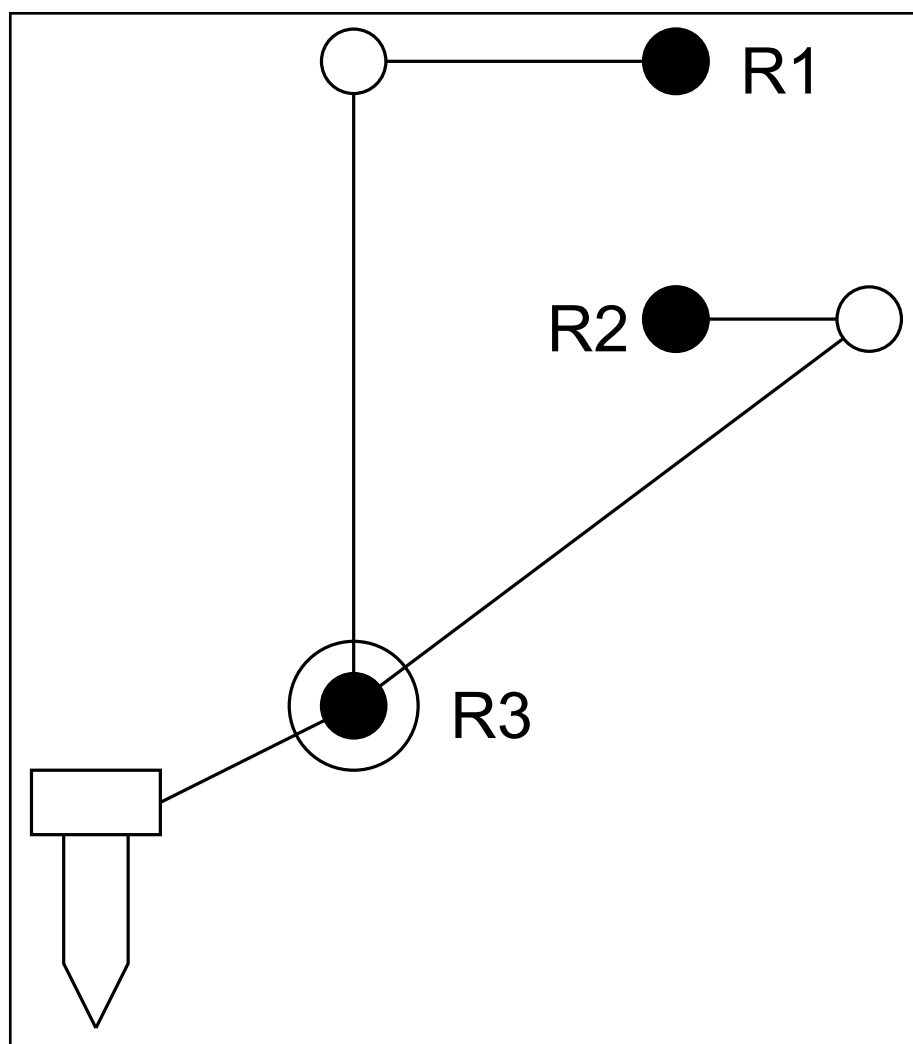


Abb. 11-21: Auswirkung einer Drehung der Achsen R1 und R2, aber ohne Drehung der Achse R3

Abb. 11-22 "Auswirkung einer zusätzlichen Drehung von Achse R3" auf Seite 235 zeigt diese Konfiguration nach einer zusätzlichen Drehung der Achse R3.

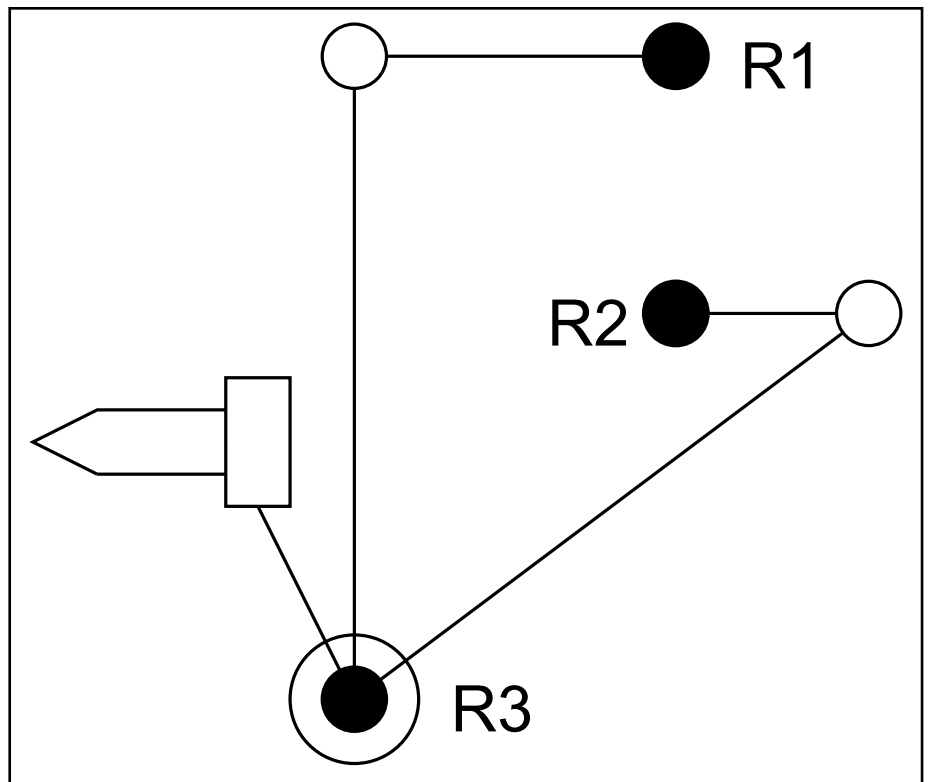


Abb. 11-22: Auswirkung einer zusätzlichen Drehung von Achse R3

Der Transformationstyp wird im Maschinenparameter 103000110 "Transformationstyp" festgelegt:

MP 103000110 = 2103002

Im Maschinenparameter 103000120 werden die "Systemachsen/-koordinaten der Transformation" festgelegt:

Systemachsnummer der R1-Achse: MP 103000120 (1)

Systemachsnummer der R2-Achse: MP 103000120 (2)

Systemachsnummer der R3-Achse: MP 103000120 (3)

Die restlichen Einträge MP 103000120 (4..8) sind nicht relevant.

Varianten

Als Achstransformation an Wirkstelle 2 (AT2) wirkt diese zwischen dem Basis-Koordinatensystem BCS als Eingangskoordinatensystem und dem Maschinen-Koordinatensystem MCS = (R1, R2, R3) als Ausgangskoordinatensystem.

Das Eingangskoordinatensystem (BCS) kann durch den Maschinenparameter 103000125 "Achsklassifikationen der Transformationsachsen" wie folgt gewählt werden:

MP 103000125 (1..3) = ± 100 :

Die BCS-Eingangskoordinaten der Transformation sind y, z, phi.

Die Achsen R1, R2, R3 haben für die Achstransformation die Bedeutung A.

Die positive Drehrichtung einer mit +100 parametrisierten Achse ist die Drehrichtung, in der man die positive y-Richtung um 90° drehen müsste, um zur positiven z-Richtung zu gelangen.

Die positive Drehrichtung einer mit -100 parametrisierten Achse ist umgekehrt.

MP 103000125 (1..3) = ± 200 :

Achs-Transformation

Die BCS-Eingangskoordinaten der Transformation sind z, x, phi.

Die Achsen R1, R2, R3 haben für die Achstransformation die Bedeutung B.

Die positive Drehrichtung einer mit +200 parametrisierten Achse ist die Drehrichtung, in der man die positive z-Richtung um 90° drehen müsste, um zur positiven x-Richtung zu gelangen.

Die positive Drehrichtung einer mit -200 parametrisierten Achse ist umgekehrt.

MP 103000125 (1..3) = ±300:

Die BCS-Eingangskoordinaten der Transformation sind x, y, phi.

Die Achsen R1, R2, R3 haben für die Achstransformation die Bedeutung C.

Die positive Drehrichtung einer mit +300 parametrisierten Achse ist die Drehrichtung, in der man die positive x-Richtung um 90° drehen müsste, um zur positiven y-Richtung zu gelangen.

Die positive Drehrichtung einer mit -300 parametrisierten Achse ist umgekehrt.

Die restlichen Einträge MP 103000125 (4..8) sind nicht relevant.

Programmiert werden je nach Vorgabe die BCS-Eingangskoordinaten x, y, phi oder z, x, phi oder y, z, phi.

Die Steuerung berechnet daraus die MCS-Ausgangskoordinaten R1, R2, R3.

Beispiel:

MP 103000125 (1) = -300

MP 103000125 (2) = 300

MP 103000125 (3) = 300

Referenzstellung

Die Referenzstellung ist für diese Achstransformationen, wie auch für die meisten anderen Achstransformationen, dadurch definiert, dass der Werkzeugreferenzpunkt im Nullpunkt des Eingangskoordinatensystems (BCS) steht.

Für MP 103000125 (1..3) = ±100 ist das die Position $y = 0$, $z = 0$, $\phi = 0$.

Für MP 103000125 (1..3) = ±200 ist das die Position $z = 0$, $x = 0$, $\phi = 0$.

Für MP 103000125 (1..3) = ±300 ist das die Position $x = 0$, $y = 0$, $\phi = 0$.

In dieser Stellung sind die Ausgangskoordinatenwerte, also die Positionen der 3 Achsen im MCS, abzulesen, und als "Achspalten der Referenzstellung" einzutragen in den folgenden Maschinenparametern.

103000130 (1) für die R1-Achse

103000130 (2) für die R2-Achse

103000130 (3) für die R3-Achse

Geometrie

Für MP 103000125 (1..3) = ±300 haben die einzelnen Parameter von MP 103000140 folgende Bedeutung:

MP 103000140 (1): Lage von Achse R1 im BCS: x-Koordinate

MP 103000140 (2): Lage von Achse R1 im BCS: y-Koordinate

MP 103000140 (3): Abstand zwischen Achse R1 und Drehgelenk G1

MP 103000140 (4): Abstand zwischen Drehgelenk G1 und Drehgelenk G3

MP 103000140 (5): Lage von Achse R2 im BCS: x-Koordinate

MP 103000140 (6): Lage von Achse R2 im BCS: y-Koordinate

MP 103000140 (7): Abstand zwischen Achse R2 und Drehgelenk G2
MP 103000140 (8): Abstand zwischen Drehgelenk G2 und Drehgelenk G3
MP 103000140 (9): Lage von Achse R3 im BCS in Referenzstellung: x-Koordinate
MP 103000140 (10): Lage von Achse R3 im BCS in Referenzstellung: y-Koordinate
MP 103000140 (11): Minimal zulässiger Winkel im Drehgelenk G1
MP 103000140 (12): Minimal zulässiger Winkel im Drehgelenk G2
MP 103000140 (13): Minimal zulässiger Winkel im Drehgelenk G3
MP 103000140 (14):

0 oder 1:

Die positive Drehrichtung ist die Drehrichtung, in der man die positive x-Richtung um 90° drehen müsste, um zur positiven y-Richtung zu gelangen..

-1: Die positive Drehrichtung ist entgegengesetzt dazu.

Für MP 103000125 (1..3) = ± 100 gilt abweichend davon:

MP 103000140 (1): Lage von Achse R1 im BCS: z-Koordinate
MP 103000140 (2): Lage von Achse R1 im BCS: x-Koordinate
MP 103000140 (5): Lage von Achse R2 im BCS: z-Koordinate
MP 103000140 (6): Lage von Achse R2 im BCS: x-Koordinate
MP 103000140 (9): Lage von Achse R3 im BCS in Referenzstellung: z-Koordinate
MP 103000140 (10): Lage von Achse R3 im BCS in Referenzstellung: x-Koordinate
MP 103000140 (14):

0 oder 1:

Die positive Drehrichtung ist die Drehrichtung, in der man die positive z-Richtung um 90° drehen müsste, um zur positiven x-Richtung zu gelangen.

-1: Die positive Drehrichtung ist entgegengesetzt dazu.

MP 103000140 (1): Lage von Achse R1 im BCS: y-Koordinate
MP 103000140 (2): Lage von Achse R1 im BCS: z-Koordinate
MP 103000140 (5): Lage von Achse R2 im BCS: y-Koordinate
MP 103000140 (6): Lage von Achse R2 im BCS: z-Koordinate
MP 103000140 (9): Lage von Achse R3 im BCS in Referenzstellung: y-Koordinate
MP 103000140 (10): Lage von Achse R3 im BCS in Referenzstellung: z-Koordinate
MP 103000140 (14):

0 oder 1:

Die positive Drehrichtung ist die Drehrichtung, in der man die positive y-Richtung um 90° drehen müsste, um zur positiven z-Richtung zu gelangen.

-1: Die positive Drehrichtung ist entgegengesetzt dazu.

MP 103000140(11..13) dient zum Schutz der Mechanik. Dort ist jeweils der kleinste Winkel einzutragen, den die Mechanik noch zulässt.

MP 103000160 ist nicht relevant.

Werkzeugkorrekturen

Die Achstransformation unterstützt eine Werkzeugkorrektur im Werkzeugkoordinatensystem (TCS): G47(XTR,YTR,ZTR)

Das TCS bewegt sich mit dem Werkzeug. In Referenzstellung stimmen BCS und TCS überein.

Der Werkzeugkorrekturvektor t zeigt von der Werkzeugspitze zum Werkzeugreferenzpunkt. Er setzt sich aus den Längenkorrekturen L_1 , L_2 , L_3 und aus der D-Korrekturtabelle zusammen.

Relevante NC-Funktionen

COORD(...)	Die 2-dimensionale Delta-Kinematik mit zusätzlicher Drehachse in der Spitze (RRR) wird aktiviert. Dies kann wahlweise per Maschinenparametersatz ($i=1..20$) oder per Mtb-Systemdatum (vom Typ SysAxTrafo_t) erfolgen.
COORD(0,2) - oder - COORD(0,1)	Die aktive Transformation wird deaktiviert.

Tab. 11-27: Relevante NC-Funktionen

Relevante Maschinenparameter

In den Maschinenparametern können bis zu 20 unterschiedliche Achstransformationen per nachfolgendem Maschinenparametersatz deklariert werden.

1030 00110	Definiert den Transformationstyp.
1030 00120	Definiert die an der Transformation beteiligten Systemachsen/-koordinaten der Transformation.
1030 00125	Definiert die Achsklassifikation der Transformationsachsen.
1030 00130	Definiert die Achspositionen der Referenzstellung.
1030 00140	Definiert die Längen- und Winkelparameter.
1030 00160	Epsilon-Umgebungen

Tab. 11-28: Relevante Maschinenparameter

Relevante Systemdaten

Die Achstransformation kann alternativ zu den Maschinenparameter auch per Systemdatum definiert werden. Dazu muss der Benutzer eine Systemvariable vom Typ "SysAxTrafo_t" anlegen (siehe Dokumentation "IndraMotion MTX 14VRS Maschinenparameter", Kapitel "Systemdaten"). Der Strukturtyp "SysAxTrafo_t" ist in der XML-Schemadateien "/feprom/schemas/sdaxtrf.xsd" steuerungsintern definiert und enthält, analog zu den relevanten Maschinenparametern, die nachfolgenden transformationsrelevanten Strukturelemente.

Strukturelement	Beschreibung
Type	Definiert den Transformationstyp.
AxisAssignment	Definiert die an der Transformation beteiligten Systemachsen/-koordinaten der Transformation.
AxisClassification	Definiert die Achsklassifikation der Transformationsachsen.

AxZeroPos	Definiert die Achspositionen der Referenzstellung.
LenParam	Definiert die Längen- und Winkelparameter.
EpsilonRanges	Epsilon-Umgebungen

Tab. 11-29: Relevante Systemdaten

Beispiele

Parametrierung

Wir gehen davon aus, dass die in [Abb. 11-19 "Elementare 2-dimensionale Delta-Transformation"](#) auf Seite 232, [Abb. 11-20 "2-dimensionale Delta-Transformation mit zusätzlicher rotatorischer Achse in der Spitze"](#) auf Seite 233, [Abb. 11-21 "Auswirkung einer Drehung der Achsen R1 und R2, aber ohne Drehung der Achse R3"](#) auf Seite 234 und [Abb. 11-22 "Auswirkung einer zusätzlichen Drehung von Achse R3"](#) auf Seite 235 gezeigte Kinematik wie folgt parametrierung sei:

MP 103000110 = 2103002

MP 103000120 (1) = 3

MP 103000120 (2) = 8

MP 103000120 (3) = 1

Die positive Drehrichtung der Achse R1 soll im Uhrzeigersinn sein.

Die positive Drehrichtung der Achse R2 soll im Gegenuhrzeigersinn sein.

Die positive Drehrichtung der Achse R3 soll im Gegenuhrzeigersinn sein.

Die Drehrichtung, in der man die positive x-Richtung um 90° drehen müsste, um zur positiven y-Richtung zu gelangen, ist die Drehrichtung im Gegenuhrzeigersinn.

Also:

MP 103000125 (1) = -300

MP 103000125 (2) = 300

MP 103000125 (3) = 300

Bevor wir die weiteren Parameter eingeben können, ist zuerst das BCS festzusetzen. Es ist also festzusetzen, welche Position der Werkzeugreferenzpunkt haben soll, wenn für die beiden Linearkoordinaten des BCS, im Beispiel also x und y, der Wert Null programmiert wird, und welche Orientierung die Senkrechte zur Spindelnase haben soll, wenn $\phi = 0$ programmiert wird. Das ist der Nullpunkt des BCS. Auf diesen Nullpunkt ist die Kinematik zu verfahren. Damit wird die Referenzstellung der Achstransformation erreicht. Als Beispiel betrachten wir die in [Abb. 11-22 "Auswirkung einer zusätzlichen Drehung von Achse R3"](#) auf Seite 235 gezeigte Stellung als Referenzstellung.

In dieser Stellung sind die Ausgangskordinatenwerte (MCS) abzulesen, und als "Achspositionen der Referenzstellung" einzutragen in:

MP 103000130 (1) für die Achse R1, also für Systemachse 3

MP 103000130 (2) für die Achse R2, also für Systemachse 8

MP 103000130 (3) für die Achse R3, also für Systemachse 1

Für das Beispiel nehmen wir folgende Werte an:

MP 103000130 (1) = 17

MP 103000130 (2) = 31

MP 103000130 (3) = 23

Ebenfalls auf die Referenzstellung beziehen sich die Maschinenparameter:

MP 103000140 (9): Lage von Achse R3 im BCS in Referenzstellung: x-Koordinate

MP 103000140 (10): Lage von Achse R3 im BCS in Referenzstellung: y-Koordinate

In Abb. 11-23 "Beispiel für eine Referenzstellung der 2-dimensionalen Delta-Kinematik mit zusätzlicher Drehachse in der Spitze (RRR)" auf Seite 240 ist das gewünschte BCS eingezeichnet. Der Werkzeugreferenzpunkt steht im Nullpunkt des BCS.

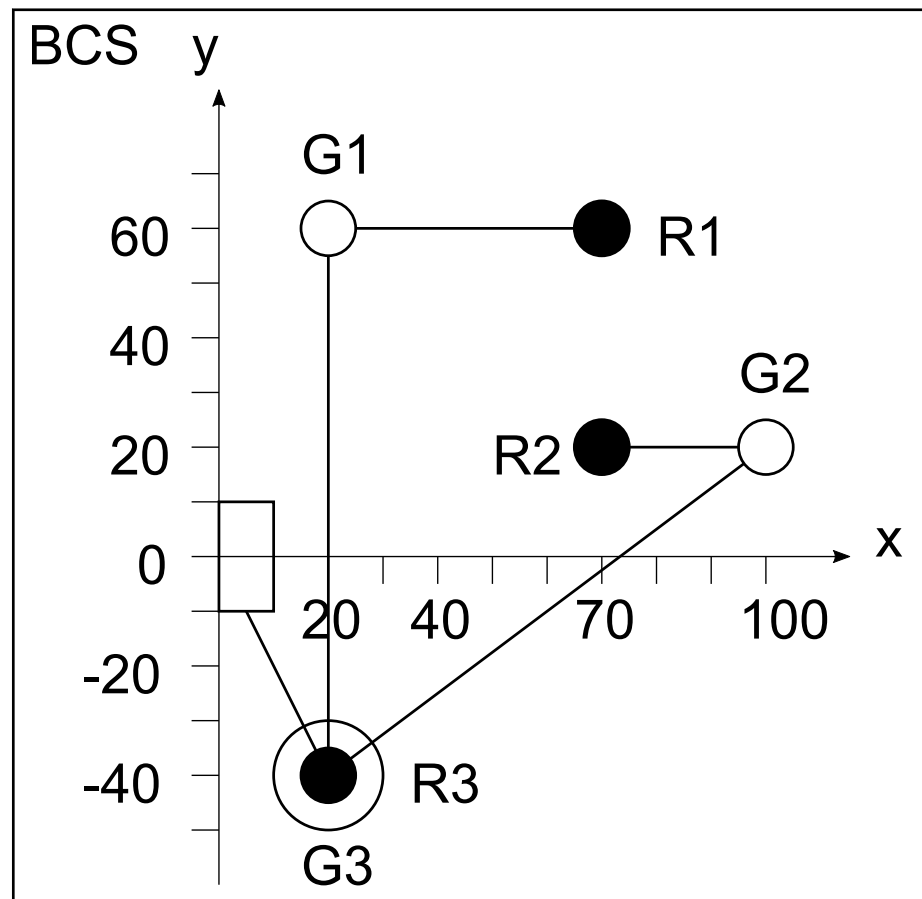


Abb. 11-23: Beispiel für eine Referenzstellung der 2-dimensionalen Delta-Kinematik mit zusätzlicher Drehachse in der Spitze (RRR)

Wir lesen folgende Werte ab:

MP 103000140 (1) = 70

MP 103000140 (2) = 60

MP 103000140 (3) = 50

MP 103000140 (4) = 100

MP 103000140 (5) = 70

MP 103000140 (6) = 20

MP 103000140 (7) = 30

MP 103000140 (8) = 100

MP 103000140 (9) = 20

MP 103000140 (10) = -40

Für dieses Beispiel nehmen wir an, dass die Mechanik folgende Minimalwinkel in den Drehgelenken G1, G2 und G3 zulässt:

MP 103000140 (11) = 30°

MP 103000140 (12) = 30°

MP 103000140 (13) = 45°

In Abb. 11-23 "Beispiel für eine Referenzstellung der 2-dimensionalen Delta-Kinematik mit zusätzlicher Drehachse in der Spitze (RRR)" auf Seite 240 haben wir in Drehgelenk G1 den Winkel 90° , in Drehgelenk G2 etwa 36.87° und in Drehgelenk G3 etwa 51.13° .

Ausgehend von der in Abb. 11-23 "Beispiel für eine Referenzstellung der 2-dimensionalen Delta-Kinematik mit zusätzlicher Drehachse in der Spitze (RRR)" auf Seite 240 gezeigten Referenzstellung $x = y = 0$, $\phi = 0$ soll sich die Senkrechte zur Spindelnase im Gegenuhrzeigersinn drehen, wenn ϕ vergrößert wird. Das ist die Drehrichtung, in der man die positive x-Richtung um 90° drehen müsste, um zur positiven y-Richtung zu gelangen. Daher können wir die Voreinstellung für diesen Wert unverändert lassen, also:

MP 103000140 (14) = 0

Werkzeugkorrektur

Bei nicht aktiver Werkzeugkorrektur gilt für die in Abb. 11-24 "Beispiel für BCS- und MCS-Position mit und ohne Werkzeugkorrektur" auf Seite 242 gezeigte Stellung:

BCS: $x = 80$, $y = 140$, $\phi = -90^\circ$

MCS: $R1 = 17^\circ + 180^\circ = 197^\circ$, $R2 = 31^\circ - 180^\circ = -149^\circ$, $R3 = 23^\circ - 90^\circ = -67^\circ$

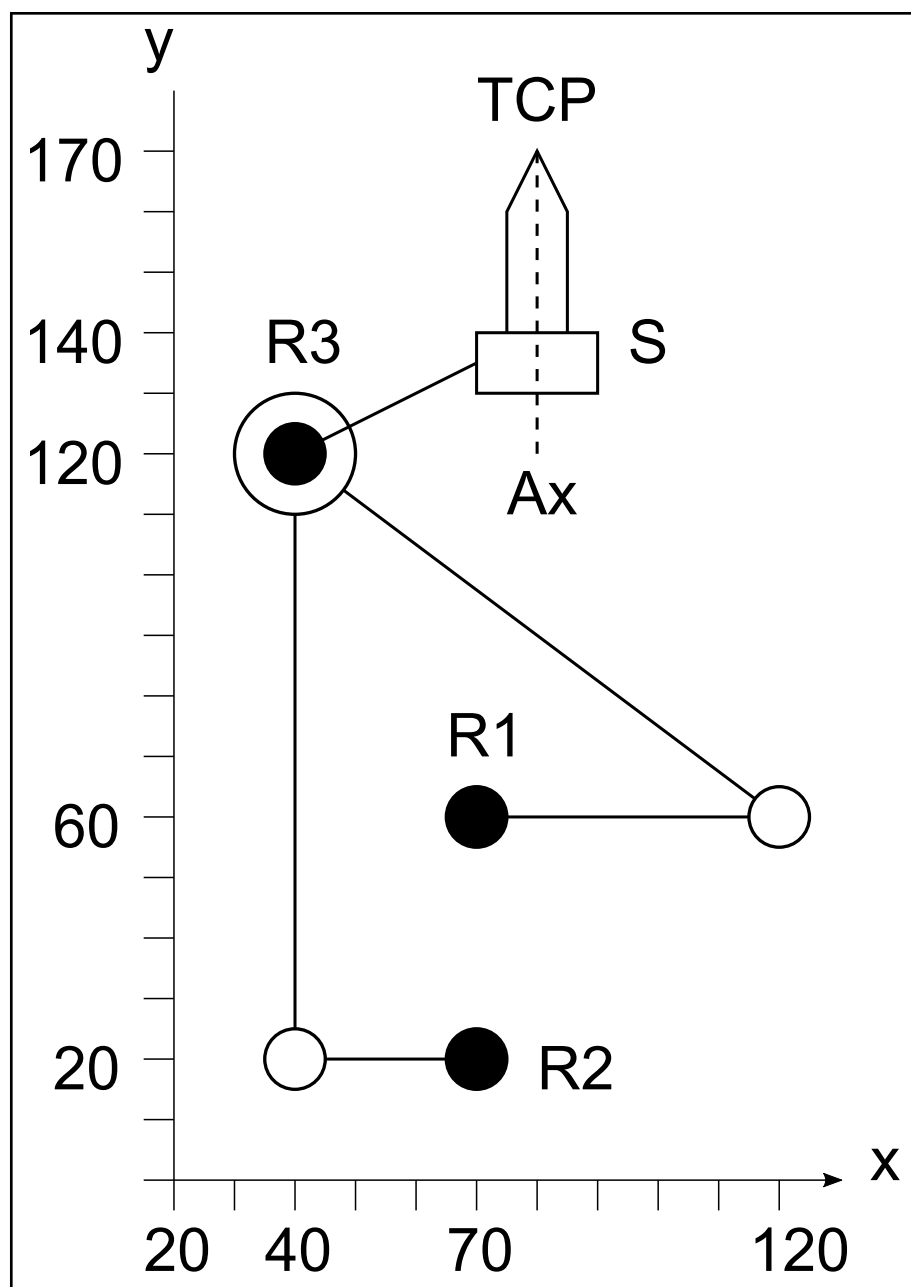


Abb. 11-24: Beispiel für BCS- und MCS-Position mit und ohne Werkzeugkorrektur

Bei aktiver Achstransformation führt also beispielweise der Befehl $G1\ x=80\ y=140\ \phi=-90$ dazu, dass die zuvor genannte Position erreicht wird.

Die Werkzeugkorrektur für das in Abb. 11-24 "Beispiel für BCS- und MCS-Position mit und ohne Werkzeugkorrektur" auf Seite 242 gezeigte Werkzeug ist wie folgt zu aktivieren:

$DCT(1,1,0) = 30$

$G47(XTR) ED1$

In der in Abb. 11-23 "Beispiel für eine Referenzstellung der 2-dimensionalen Delta-Kinematik mit zusätzlicher Drehachse in der Spitze (RRR)" auf Seite 240 gezeigten Referenzstellung würde der Werkzeugkorrekturvektor für dieses Werkzeug in die positive x-Richtung weisen. Da in Referenzstellung BCS und TCS übereinstimmen, verläuft der Werkzeugkorrekturvektor im TCS in

die positive x-Richtung. Wäre diese Werkzeugkorrektur aktiv, hätten wir in Abb. 11-24 "Beispiel für BCS- und MCS-Position mit und ohne Werkzeugkorrektur" auf Seite 242 $x = 80$, $y = 170$, $\phi = -90^\circ$.

Parametrierung bei einem anderen BCS

Um die Auswirkung des BCS auf die Parametrierung zu verdeutlichen, gehen wir jetzt von der in Abb. 11-25 "Beispiel eine Referenzstellung in einem anderen BCS" auf Seite 243 gezeigten Referenzstellung aus.

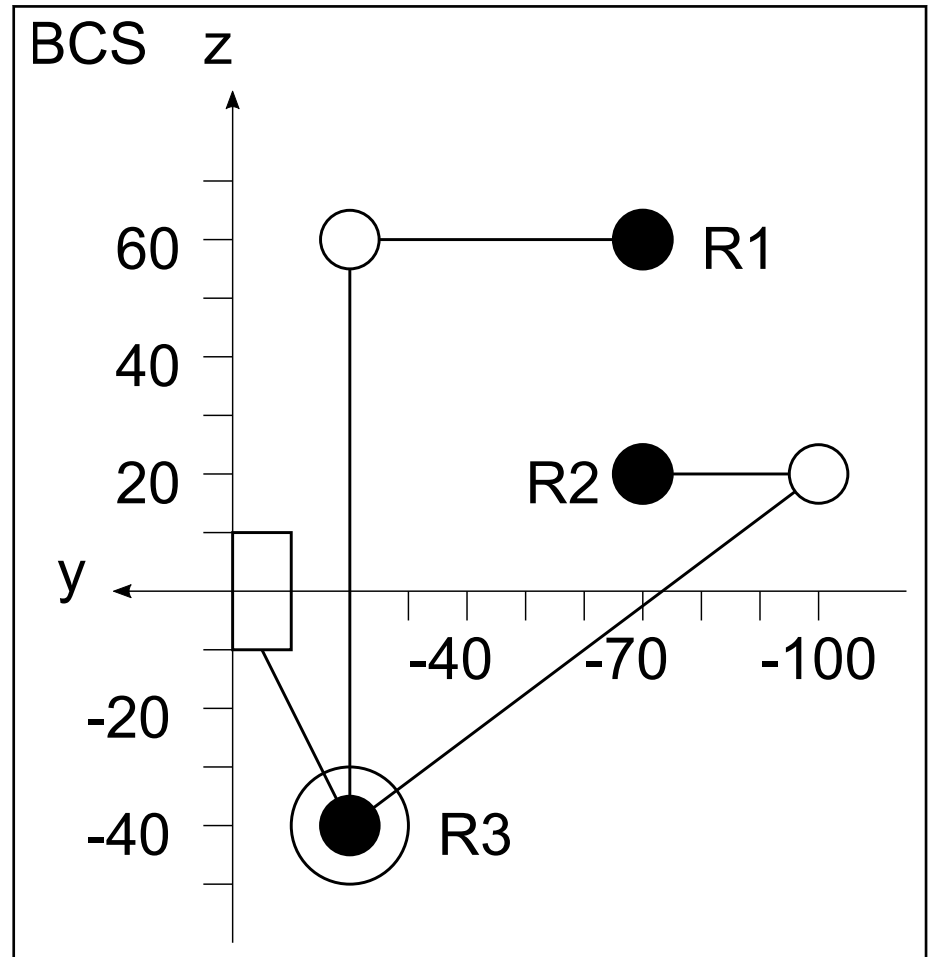


Abb. 11-25: Beispiel eine Referenzstellung in einem anderen BCS

Auch hier soll gelten:

Die positive Drehrichtung der Achse R1 soll im Uhrzeigersinn sein.

Die positive Drehrichtung der Achse R2 soll im Gegenuhrzeigersinn sein.

Die positive Drehrichtung der Achse R3 soll im Gegenuhrzeigersinn sein.

Ausgehend von der Referenzstellung $x = y = 0$, $\phi = 0$ soll sich die Senkrechte zur Spindelnase im Gegenuhrzeigersinn drehen, wenn ϕ vergrößert wird.

Die Drehrichtung, in der man die positive y-Richtung um 90° drehen müsste, um zur positiven z-Richtung zu gelangen, ist hier jedoch die Drehrichtung im Uhrzeigersinn.

Damit ergeben sich folgende Änderungen in der Parametrierung:

MP 103000125 (1) = 100

MP 103000125 (2) = -100

MP 103000125 (3) = -100

MP 103000140 (1) = -70

MP 103000140 (5) = -70

MP 103000140 (9) = -20

MP 103000140 (14) = -1

11.7.2 Handlungsanweisung

Applizieren Die Achstransformation kann über die relevanten Maschinenparameter appliziert werden. Dazu muss die Funktion Achstransformation über den Parameter **tra** "Achstransformationen" im Setup (SUP) aktiviert werden. Der Pfad des MP-Namens ist immer "TRA/AxTrafo[1..20]/...".

Alternativ dazu kann die Achstransformation über ein benutzerdefiniertes Systemdatum vom Typ SysAxTrafo_t appliziert werden.

Die beiden Varianten sind inhaltlich identisch.

Die Achspositionen der Referenzstellung müssen ermittelt werden.

In der nachfolgenden Tabelle sind jeweils die äquivalenten MP-Nummern, MP-Name und SD-Elemente angegeben. Um eine Transformation zu applizieren, muss eine der Varianten entsprechend besetzt werden.

Transformations-relevante Parameter

MP-Nummer	Beschreibung	
1030 00110	Transformationstyp	
	MP-Name: TrafoType SD-Elem. : Type	
	2103002	Wirkstelle AT2
1030 00120	Systemachsen/-koordinaten der Transformation	
	MP-Name: FwdInCoordIndTrafo[1..8] SD-Elem. : AxisAssignment[1..8]	
	[1]	Systemachsnummer von R1
	[2]	Systemachsnummer von R2
	[3]	Systemachsnummer von R3
	[4..8]	nicht relevant

MP-Nummer	Beschreibung			
1030 00125	Achsklassifikationen der Transformationsachsen MP-Name: CoordClass[1..8] MP-Name: CoordDir[1..8] SD-Elem. : AxisClassification[1..8]			
	Definiert die an der Transformation beteiligten Achsen inklusive Bewegungsrichtung.			
	- MP-Nr. oder SD-Element: Wert ">0" bedeutet: Die Achse verfhrt/dreht positiv im Sinne der obigen Abbildungen. Wert "<0" bedeutet: Die Achse verfhrt/dreht negativ im Sinne der obigen Abbildungen. <i>Mgliche Werte sind:</i> ±100 = A-Achse ±200 = B-Achse ±300 = C-Achse - MP-Name: <i>Mgliche Werte fr CoordClass sind:</i> - A,B,C (rotatorisch) <i>Mgliche Werte fr CoordDir sind:</i> - positiv - negativ			
	Ist ein Parameter unbesetzt, gelten folgende Default-Einstellungen:			
		MP-Nr / SD-Elem.:	CoordClass:	CoordDir:
	[1]	300	C	positiv
[2]	300	C	positiv	
[3]	300	C	positiv	
[4..8]	Nicht relevant	Nicht relevant	Nicht relevant	
1030 00130	Achspositionen der Referenzstellung MP-Name: RefPosTrafo[1..8] SD-Elem. : AxZeroPos[1..8]			
	[1]	R1 ⁰		
	[2]	R2 ⁰		
	[3]	R3 ⁰		
	[4..8]	Nicht relevant		

Achs-Transformation

MP-Nummer	Beschreibung
1030 00140	Längen- und Winkelparameter
	MP-Name: JointParTrafo[1..16]
	SD-Elem. : LenParam[1..16]
	Definiert die Geometrie der Kinematik, Einheit mm bzw. Grad
	[1] Lage von Achse R1 im BCS: Hauptkoordinate
	[2] Lage von Achse R1 im BCS: Nebenkoordinate
	[3] Abstand zwischen Achse R1 und Drehgelenk G1
	[4] Abstand zwischen Drehgelenk G1 und Drehgelenk G3
	[5] Lage von Achse R2 im BCS: Hauptkoordinate
	[6] Lage von Achse R2 im BCS: Nebenkoordinate
	[7] Abstand zwischen Achse R2 und Drehgelenk G2
	[8] Abstand zwischen Drehgelenk G2 und Drehgelenk G3
	[9] Lage von Achse R3 im BCS in Referenzstellung: Hauptkoordinate
	[10] Lage von Achse R3 im BCS in Referenzstellung: Nebenkoordinate
	[11] Minimal zulässiger Winkel im Drehgelenk G1
	[12] Minimal zulässiger Winkel im Drehgelenk G2
	[13] Minimal zulässiger Winkel im Drehgelenk G3
	[14] 0 oder 1: Die positive Drehrichtung ist die Drehrichtung, in der man die positive x-Richtung um 90° drehen müsste, um zur positiven y-Richtung zu gelangen. -1: Die positive Drehrichtung ist entgegengesetzt dazu.
	[15..16] Nicht relevant
1030 00160	Epsilon-Umgebungen
	MP-Name: EpsAxTrafo
	SD-Elem. : EpsilonRanges
	nicht relevant

Tab. 11-30: Transformationsrelevante Parameter

Aktivieren Ist die 2-dimensionale Delta-Kinematik mit zusätzlicher Drehachse in der Spitze konfiguriert per

- Maschinenparameter, dann wird die Transformation im NC-Programm aktiviert mit (i=1,..20):

COORD(<i>)

- Systemdatum, dann wird die Transformation im NC-Programm aktiviert mit (Typ "SysAxTrafo_t"):

COORD(<SD.MtbAxTrafo>)

Deaktivieren **COORD(0)** bzw. **COORD(0,2)**

schaltet die Achstransformation im NC-Programm aus.

11.8 4-Achs-Transformation - Typ 2: LLLR

11.8.1 Beschreibung

Funktion

Bei dieser Transformation handelt es sich um eine Konfiguration mit drei Linearachsen und einer Schwenkachse. Die Schwenkachse R_2 bewirkt eine Verdrehung des Werkzeugs. Die Transformation kommt an der Wirkstelle 2 zum Einsatz.

Name der Transformation	Transformationstyp	Wirkstelle	Koordinatensystem
4-Achs-Transformation LLLR	3131002	AT2	MCS, BCS

Ausgangsgrößen im MCS: X, Y, Z, R_2

Eingangsgrößen im BCS: x, y, z, θ

Die linearen MCS-Koordinaten X, Y, Z spannen ein rechtshändiges kartesisches Koordinatensystem auf. In der Referenzstellung (BCS parallel zum MCS) dreht R_2 wahlweise um X (Typ A-Achse) oder um Y (Typ B-Achse). Die Achse R_2 dient der Orientierung.

Die nachfolgenden Abbildungen zeigen die LLLR-Konfigurationen in der Referenzstellung mit $R_2 = A$ bzw. $R_2 = B$. D. h. jeweils: Nimmt die Maschine die in der Abbildung dargestellte Stellung ein, so müssen die BCS-Koordinaten $(x, y, z, \theta) = (0, 0, 0, 0)$ sein, bei zugehörigen MCS-Koordinaten $(X, Y, Z, R_2) = (X^0, Y^0, Z^0, R_2^0)$. Die positiven Dreh-/Verfahr-Richtungen der Achsen sind in den Abbildungen dargestellt. Diese sind entsprechend einzustellen! In dieser Stellung steht die Spitze (TCP tool center point) des rotationssymmetrischen Werkzeugs (Fräser oder Laser) auf dem Ursprung des Basiskoordinatensystems (BCS) und ist entlang der z -Koordinate ausgerichtet. Die Ausrichtung des BCS ist durch die Achs-Transformation eindeutig festgelegt.

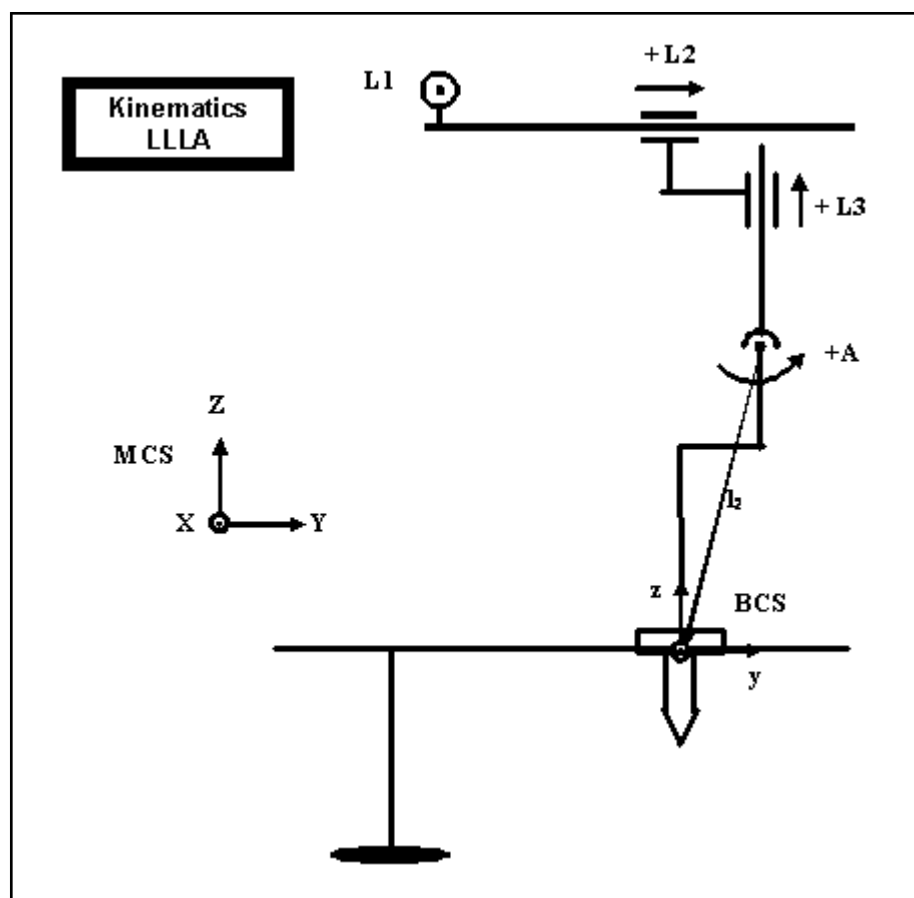


Abb. 11-26: LLLR (Typ A-Achse)

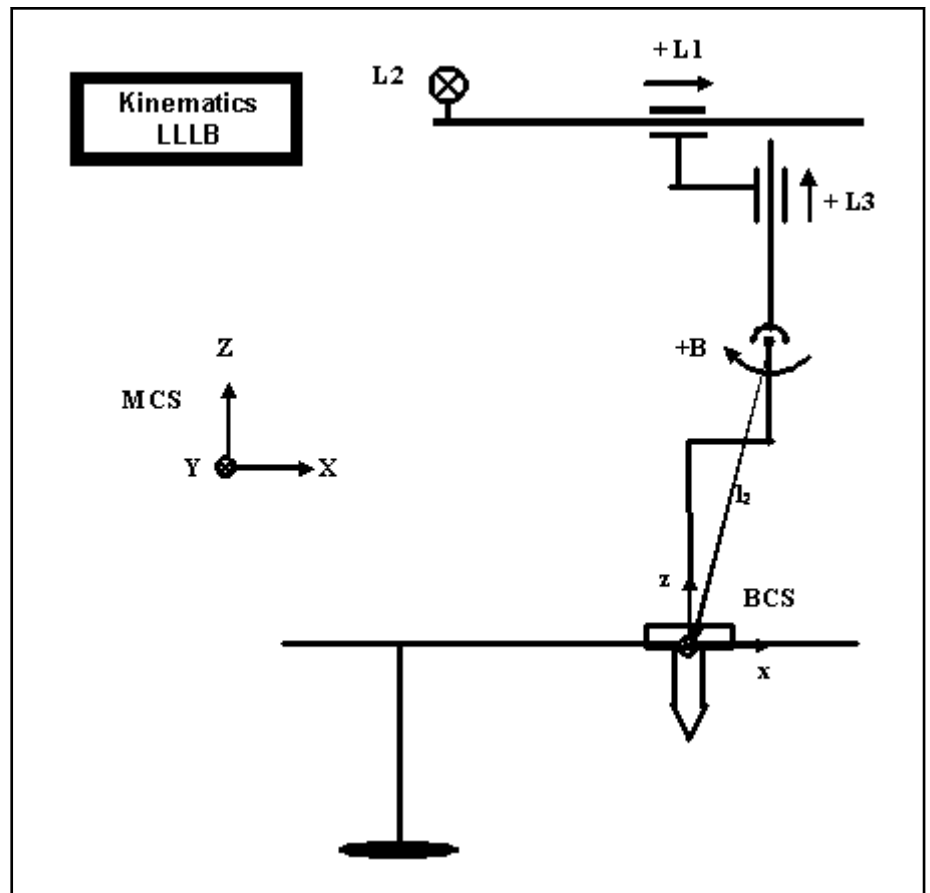


Abb. 11-27: LLLR (Typ B-Achse)

Der Vektor $\mathbf{l}_2$ definiert die Geometrie der 4-Achs-Kinematik. Er hat die Form $\mathbf{l}_2 = (l_{2x}, l_{2y}, l_{2z})$. $\mathbf{l}_2$ ist der Vektor von der R_2 -Achse (A/B-Achse) zum Nullpunkt des BCS.

Die rotatorische Achse R_2 kann sowohl Endlos- als auch Rundachse sein.

Werkzeugkorrektur: Eine Werkzeugkorrektur ist möglich. Der Korrekturvektor zeigt von der Werkzeugspitze zur Werkzeugaufnahme.

Relevante NC-Funktionen

COORD ()	Die 4-Achs-Transformation LLLR wird aktiviert. Dies kann wahlweise per Maschinenparametersatz (i=1..20) oder per MTB-Systemdatum (vom Typ SysAxTrafo_t) erfolgen.
COORD (0,2)	Die aktive Achstransformation wird deaktiviert.

Tab. 11-31: Relevante NC-Funktionen

11.8.2 Handlungsanweisung

Applizieren Die Achstransformation kann über die relevanten Maschinenparameter appliziert werden. Dazu muss die Funktion Achstransformation über den Parameter **tra** "Achstransformationen" im Setup (SUP) aktiviert werden. Der Pfad des MP-Namens ist immer "TRA/AxTrafo[1...20]/...".

Alternativ dazu kann die Achstransformation über ein benutzerdefiniertes Systemdatum vom Typ `SysAxTrafo_t` appliziert werden.

Die beiden Varianten sind inhaltlich identisch.

Die Achspositionen der Referenzstellung müssen ermittelt werden.

In der nachfolgenden Tabelle sind jeweils die äquivalenten MP-Nummern, MP-Name und SD-Elemente angegeben. Um eine Transformation zu applizieren, muss eine der Varianten entsprechend besetzt werden.

4-Achs-Variante LLRR: Transformations-relevante Parameter

MP-Nummer	Beschreibung	
1030 00110	Transformationstyp MP-Name: TrafoType SD-Elem. : Type	
	3131002	4-Achs-Transformation LLRR
1030 00120	Systemachsen/-koordinaten der Transformation MP-Name: FwdInCoordIndTrafo[1..8] SD-Elem. : AxisAssignment[1..8]	
	[1]	Systemachsnummer von X
	[2]	Systemachsnummer von Y
	[3]	Systemachsnummer von Z
	[4]	Systemachsnummer von R ₂ (A-/B-Achse)
	[5..8]	nicht relevant

MP-Nummer	Beschreibung																								
1030 00125	Achsklassifikationen der Transformationsachsen MP-Name: CoordClass[1..8] MP-Name: CoordDir[1..8] SD-Elem. : AxisClassification[1..8] Definiert die an der Transformation beteiligten Achsen inklusive Bewegungsrichtung. - MP-Nr. oder SD-Element: Wert ">0" bedeutet: Die Achse verfährt/dreht positiv im Sinne der obigen Abbildungen. Wert "<0" bedeutet: Die Achse verfährt/dreht negativ im Sinne der obigen Abbildungen. <i>Mögliche Werte sind:</i> +/-1 = X-Achse +/-2 = Y-Achse +/-3 = Z-Achse +/-100 = A-Achse +/-200 = B-Achse +/-300 = C-Achse - MP-Name: <i>Mögliche Werte für CoordClass sind:</i> - X,Y,Z (linear) - A,B,C (rotatorisch) <i>Mögliche Werte für CoordDir sind:</i> - positiv (bzgl. Abb. "Referenzstellung") - negativ (bzgl. Abb. "Referenzstellung") Ist ein Parameter unbesetzt, gelten folgende Default-Einstellungen: <table><tr><th></th><th>MP-Nr / SD-Elem.:</th><th>CoordClass:</th><th>CoordDir:</th></tr><tr><td>[1]</td><td>1</td><td>X</td><td>positiv</td></tr><tr><td>[2]</td><td>2</td><td>Y</td><td>positiv</td></tr><tr><td>[3]</td><td>3</td><td>Z</td><td>positiv</td></tr><tr><td>[4]</td><td>200</td><td>B</td><td>positiv</td></tr><tr><td>[5..8]</td><td>nicht relevant</td><td>0</td><td>positiv</td></tr></table>		MP-Nr / SD-Elem.:	CoordClass:	CoordDir:	[1]	1	X	positiv	[2]	2	Y	positiv	[3]	3	Z	positiv	[4]	200	B	positiv	[5..8]	nicht relevant	0	positiv
	MP-Nr / SD-Elem.:	CoordClass:	CoordDir:																						
[1]	1	X	positiv																						
[2]	2	Y	positiv																						
[3]	3	Z	positiv																						
[4]	200	B	positiv																						
[5..8]	nicht relevant	0	positiv																						

Achs-Transformation

MP-Nummer	Beschreibung
1030 00130	Achspositionen der Referenzstellung MP-Name: RefPosTrafo[1..8] SD-Elem. : AxZeroPos[1..8]
	[1] X^0
	[2] Y^0
	[3] Z^0
	[4] R_2^0
	[5..8] nicht relevant
1030 00140	Längen- und Winkelparameter MP-Name: JointParTrafo[1..16] SD-Elem. : LenParam[1..16] Definiert die Längenvektoren l_1, l_2 in [mm].
	[1] nicht relevant
	[2] nicht relevant
	[3] nicht relevant
	[4] l_{2x}
	[5] l_{2y}
	[6] l_{2z}
	[7..16] nicht relevant
1030 00160	Epsilon-Umgebungen
	nicht relevant

Tab. 11-32: Transformations-relevante Parameter

Aktivieren Ist die Transformation konfiguriert per

- Maschinenparameter, dann wird die Transformation im NC-Programm aktiviert mit (i=1,..20):

COORD(<i>)

- Systemdatum, dann wird die Transformation im NC-Programm aktiviert mit (Typ "SysAxTrafo_t"):

COORD(<SD.MtbAxTrafo>)

Programmbeispiel:

Die 4-Achs-Transformation LLLR ist im 4. Maschinenparametersatz definiert. Am Programmanfang ist keine Achstransformation aktiv:

Transformation aktivieren

```

N100 G1 F1000 Z10 X=10 B=30 ;Axes Z, X and B are traversed
N110 COORD(4) ;Switching on the axis transformation
N130 z20 y30 theta10 ;Coordinates z, y and theta are traversed
      :
N999 COORD(0) ;Switching-off the axis transformation
      ;Axes are again programmed from here

```

Deaktivieren **COORD(0)** bzw. **COORD(0,2)**

schaltet die Achstransformation im NC-Programm aus.

11.9 4-Achs-Transformation - Typ 3: RLLL

11.9.1 Beschreibung

Funktion

Bei dieser Transformation handelt es sich um eine Konfiguration mit drei Linearachsen und einer Schwenkachse. Die Schwenkachse R_2 bewirkt eine Verdrehung des Werkstücks. Die Transformation kommt an der Wirkstelle 2 zum Einsatz.

Name der Transformation	Transformationstyp	Wirkstelle	Koordinatensystem
4-Achs-Transformation RLLL	3131003	AT2	MCS, BCS

Ausgangsgrößen im MCS: R_2, X, Y, Z

Eingangsgrößen im BCS: x, y, z, θ

Die linearen MCS-Koordinaten X, Y, Z spannen ein rechtshändiges kartesisches Koordinatensystem auf. In der Referenzstellung (BCS parallel zum MCS) dreht R_2 wahlweise um X (Typ A-Achse) oder um Y (Typ B-Achse). Die Achse R_2 dient der Orientierung.

Die nachfolgenden Abbildungen zeigen die RLLL-Konfigurationen in der Referenzstellung mit $R_2 = A$ bzw. $R_2 = B$. D. h. jeweils: Nimmt die Maschine die in der Abbildung dargestellte Stellung ein, so müssen die BCS-Koordinaten $(x, y, z, \theta) = (0, 0, 0, 0)$ sein, bei zugehörigen MCS-Koordinaten $(R_2, X, Y, Z) = (R_2^0, X^0, Y^0, Z^0)$. Die positiven Dreh-/Verfahr-Richtungen der Achsen sind in den Abbildungen dargestellt. Diese sind entsprechend einzustellen! In dieser Stellung steht die Spitze (TCP tool center point) des rotationssymmetrischen Werkzeugs (Fräser oder Laser) auf dem Ursprung des Basiskoordinatensystems (BCS) und ist entlang der z -Koordinate ausgerichtet. Die Ausrichtung des BCS ist durch die Achs-Transformation eindeutig festgelegt.

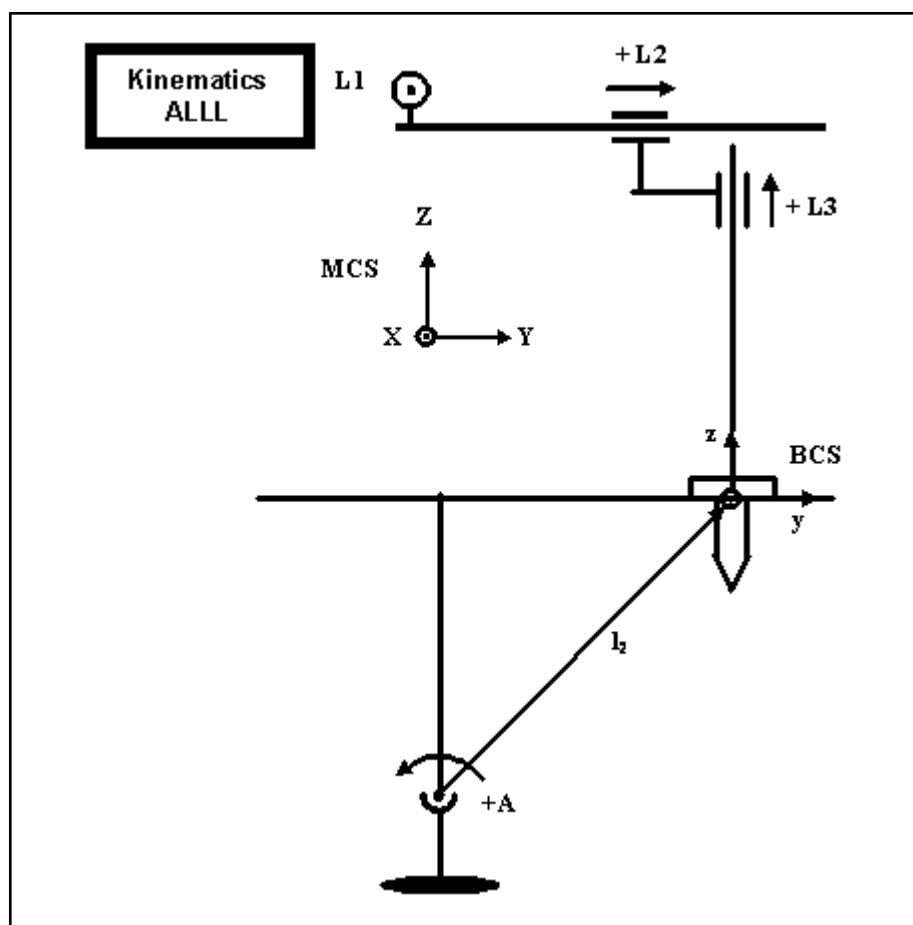


Abb. 11-28: RLL (Typ A-Achse)

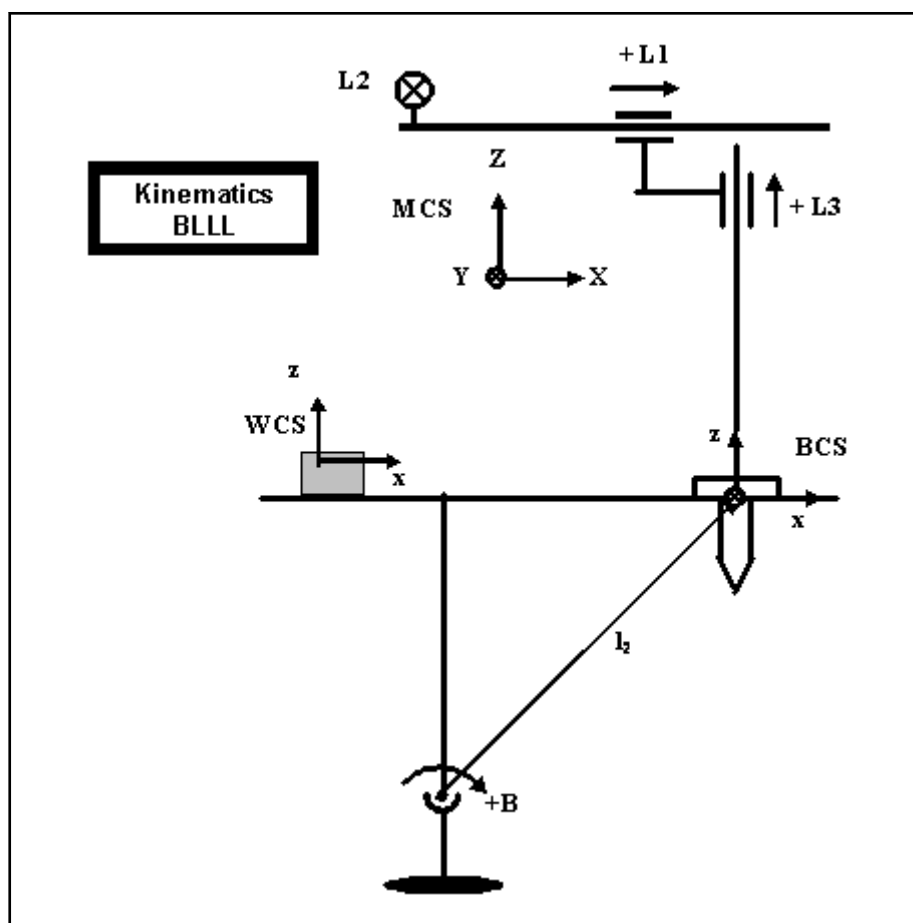


Abb. 11-29: RLL (Typ B-Achse)

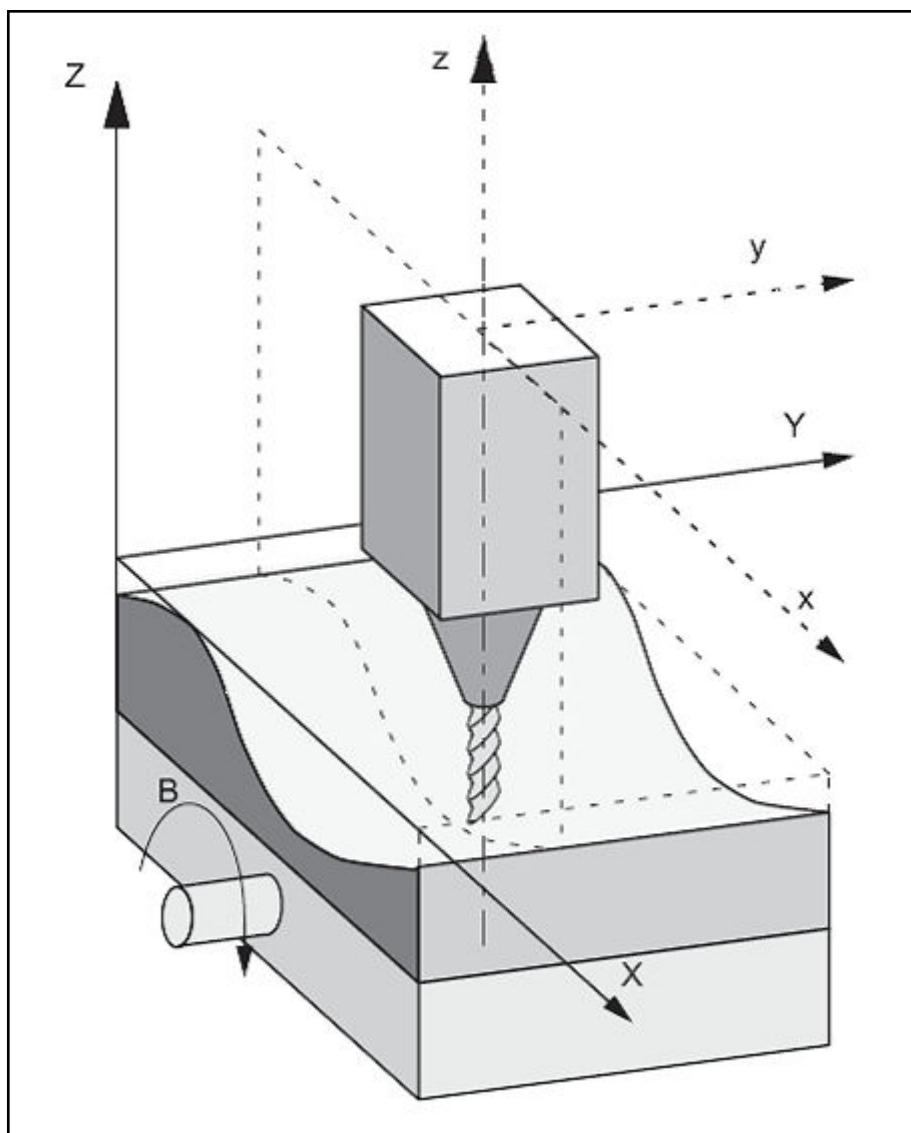


Abb. 11-30: RLLL (Typ B-Achse)

Der Vektor l_2 definiert die Geometrie der 4-Achs-Kinematik. Er hat die Form $l_2 = (l_{2x}, l_{2y}, l_{2z})$. l_2 ist der Vektor von der R_2 -Achse (A-/B-Achse) zum Nullpunkt des BCS.

Die rotatorische Achse R_2 kann sowohl Endlos- als auch Rundachse sein.

Werkzeugkorrektur:

Eine Werkzeugkorrektur ist möglich. Der Korrekturvektor zeigt von der Werkzeugspitze zur Werkzeugaufnahme.

Relevante NC-Funktionen

COORD ()	Die 4-Achs-Transformation RLLL wird aktiviert. Dies kann wahlweise per Maschinenparametersatz ($i=1..20$) oder per MTB-Systemdatum (vom Typ SysAxTrafo_t) erfolgen.
COORD (0,2)	Die aktive Achstransformation wird deaktiviert.

Tab. 11-33: Relevante NC-Funktionen

11.9.2 Handlungsanweisung

Applizieren Die Achstransformation kann über die relevanten Maschinenparameter appliziert werden. Dazu muss die Funktion Achstransformation über den Parameter **tra** "Achstransformationen" im Setup (SUP) aktiviert werden. Der Pfad des MP-Namens ist immer "TRA/AxTrafo[1..20]/...".

Alternativ dazu kann die Achstransformation über ein benutzerdefiniertes Systemdatum vom Typ SysAxTrafo_t appliziert werden.

Die beiden Varianten sind inhaltlich identisch.

Die Achspositionen der Referenzstellung müssen ermittelt werden.

In der nachfolgenden Tabelle sind jeweils die äquivalenten MP-Nummern, MP-Name und SD-Elemente angegeben. Um eine Transformation zu applizieren, muss eine der Varianten entsprechend besetzt werden.

4-Achs-Variante LLRR: Transformations-relevante Parameter

MP-Nummer	Beschreibung	
1030 00110	Transformationstyp MP-Name: TrafoType SD-Elem. : Type	
	3131003	4-Achs-Transformation RLLL
1030 00120	Systemachsen/-koordinaten der Transformation MP-Name: FwdInCoordIndTrafo[1..8] SD-Elem. : AxisAssignment[1..8]	
	[1]	Systemachsnummer von R ₂ (A-/B-Achse)
	[2]	Systemachsnummer von X
	[3]	Systemachsnummer von Y
	[4]	Systemachsnummer von Z
	[5..8]	nicht relevant

Achs-Transformation

MP-Nummer	Beschreibung																								
1030 00125	Achsklassifikationen der Transformationsachsen MP-Name: CoordClass[1..8] MP-Name: CoordDir[1..8] SD-Elem. : AxisClassification[1..8] Definiert die an der Transformation beteiligten Achsen inklusive Bewegungsrichtung. - MP-Nr. oder SD-Element: Wert ">0" bedeutet: Die Achse verfährt/dreht positiv im Sinne der obigen Abbildungen. Wert "<0" bedeutet: Die Achse verfährt/dreht negativ im Sinne der obigen Abbildungen. <i>Mögliche Werte sind:</i> +/-1 = X-Achse +/-2 = Y-Achse +/-3 = Z-Achse +/-100 = A-Achse +/-200 = B-Achse +/-300 = C-Achse - MP-Name: <i>Mögliche Werte für CoordClass sind:</i> - X,Y,Z (linear) - A,B,C (rotatorisch) <i>Mögliche Werte für CoordDir sind:</i> - positiv (bzgl. Abb. "Referenzstellung") - negativ (bzgl. Abb. "Referenzstellung") Ist ein Parameter unbesetzt, gelten folgende Default-Einstellungen: <table><tr><th></th><th>MP-Nr / SD-Elem.:</th><th>CoordClass:</th><th>CoordDir:</th></tr><tr><td>[1]</td><td>200</td><td>B</td><td>positiv</td></tr><tr><td>[2]</td><td>1</td><td>X</td><td>positiv</td></tr><tr><td>[3]</td><td>2</td><td>Y</td><td>positiv</td></tr><tr><td>[4]</td><td>3</td><td>Z</td><td>positiv</td></tr><tr><td>[5..8]</td><td>nicht relevant</td><td>0</td><td>positiv</td></tr></table>		MP-Nr / SD-Elem.:	CoordClass:	CoordDir:	[1]	200	B	positiv	[2]	1	X	positiv	[3]	2	Y	positiv	[4]	3	Z	positiv	[5..8]	nicht relevant	0	positiv
	MP-Nr / SD-Elem.:	CoordClass:	CoordDir:																						
[1]	200	B	positiv																						
[2]	1	X	positiv																						
[3]	2	Y	positiv																						
[4]	3	Z	positiv																						
[5..8]	nicht relevant	0	positiv																						

MP-Nummer	Beschreibung
1030 00130	Achspositionen der Referenzstellung MP-Name: RefPosTrafo[1..8] SD-Elem. : AxZeroPos[1..8]
	[1] R_2^0
	[2] X^0
	[3] Y^0
	[4] Z^0
	[5..8] nicht relevant
1030 00140	Längen- und Winkelparameter MP-Name: JointParTrafo[1..16] SD-Elem. : LenParam[1..16] Definiert die Längenvektoren l_1, l_2 in [mm].
	[1] nicht relevant
	[2] nicht relevant
	[3] nicht relevant
	[4] l_{2x}
	[5] l_{2y}
	[6] l_{2z}
	[7..16] nicht relevant
1030 00160	Epsilon-Umgebungen
	nicht relevant

Tab. 11-34: Transformations-relevante Parameter

Aktivieren Ist die Transformation konfiguriert per

- Maschinenparameter, dann wird die Transformation im NC-Programm aktiviert mit (i=1,..20):
COORD(<i>)
- Systemdatum, dann wird die Transformation im NC-Programm aktiviert mit (Typ "SysAxTrafo_t"):
COORD(<SD.MtbAxTrafo>)

Programmbeispiel:

Die 4-Achs-Transformation LLLR ist im 4. Maschinenparametersatz definiert.
Am Programmanfang ist keine Achstransformation aktiv:

Transformation aktivieren

```

N100 G1 F1000 Z10 X=10 B=30 ;Axes Z, X and B are traversed
N110 COORD(4) ;Switching on the axis transformation
N130 z20 y30 theta10 ;Coordinates z, y and theta are traversed
      :
N999 COORD(0) ;Switching-off the axis transformation
      ;Axes are again programmed from here

```

Deaktivieren **COORD(0)** bzw. **COORD(0,2)**

schaltet die Achstransformation im NC-Programm aus.

11.10.2 Variante 1 (mit Vektororientierung)

Funktion

Name der Transformation	Transformationstyp	Wirkstelle	Koordinatensystem
5-Achs-Transformation mit Werkzeug-Dreh- und Schwenkachse	Variante 1: 3232201 mit Vektororientierung	AT2	MCS, BCS

Tab. 11-35: Funktion - Variante 1

Ausgangsgrößen im MCS: X,Y,Z,R2,R1

Eingangsgrößen im BCS: x, y, z, phi, theta

Die Transformation kommt an der Wirkstelle 2 zum Einsatz, d. h. sie wirkt zwischen dem Maschinenkoordinatensystem MCS = (X,Y,Z,R2,R1) und dem Basiskoordinatensystem BCS = (x,y,z,phi,theta).

Die Achs-Transformation unterstützt die Vektororientierungsbewegung (Drehung des Orientierungsvektors $\vec{\rho}$) um eine raumfeste Drehachse, Großkreisbewegung der Vektorspitze. Wird ein anderer Wert für phi und/oder theta programmiert, verändert sich die Werkzeugorientierung.

Die Orientierung dreht immer um die Werkzeugspitze (TCP, Tool Center-Point). Das heißt, es wird angenommen, dass der TCP fest steht und sich die Werkzeugachse um diesen Fixpunkt dreht.

Bei der Orientierungsänderung können unter Umständen alle fünf Achsen beteiligt sein. Die Orientierung zeigt immer in Richtung der Werkzeugaufnahme.

Der Orientierungsvektor $\vec{\rho}$ kann darüber hinaus als Spline programmiert werden.

Restriktion

Eine Online-Korrektur für θ und ϕ ist **nicht** möglich.

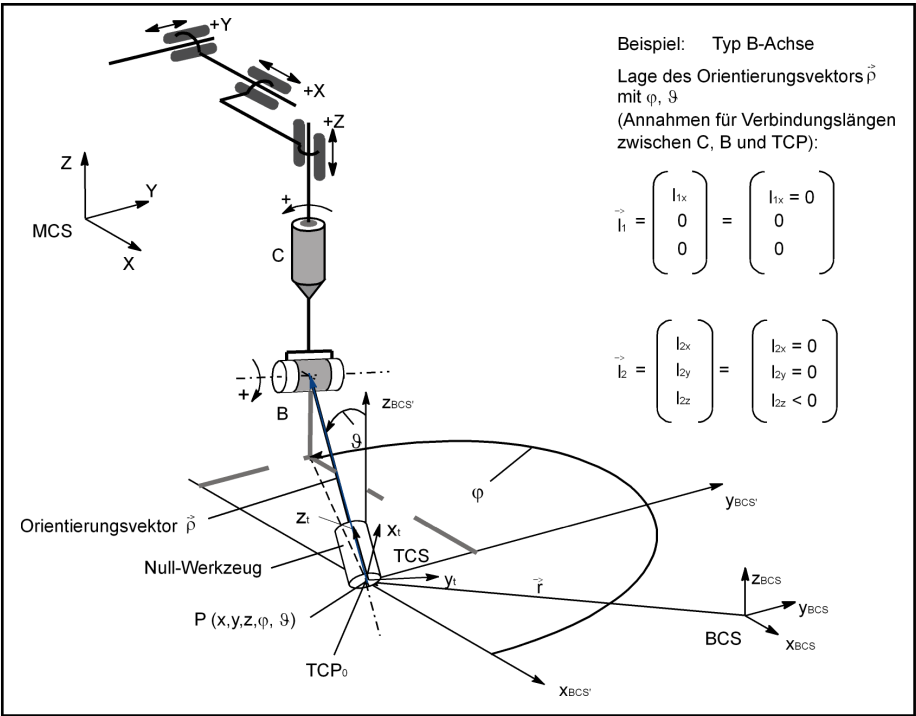


Abb. 11-32: Restriktion - Beispiel

11.10.3 Variante 2 (mit linearer Orientierung)

Funktion

Name der Transformation	Transformationstyp	Wirkstelle	Koordinatensystem
5-Achs-Transformation mit Werkzeug-Dreh- und Schwenkachse	Variante 2: 3232101 mit linearer Orientierung	AT2	MCS, BCS

Tab. 11-36: Funktion - Variante 2

Ausgangsgrößen im MCS: X,Y,Z,R2,R1

Eingangsgrößen im BCS: x, y, z, φ, ϑ

Die Transformation kommt an der Wirkstelle 2 zum Einsatz, d. h. sie wirkt zwischen dem Maschinenkoordinatensystem MCS = (X,Y,Z,R2,R1) und dem Basiskoordinatensystem BCS = (x,y,z,phi,theta).

Die Achs-Kinematik ist identisch mit dem Typ 3232201. Die Achs-Transformation unterstützt jedoch die lineare Orientierungsbewegung (Gerade in einer gedachten phi-theta-Ebene), also **keine** Vektororientierung. Bei allen Orientierungsbewegungen wird ein stetiges phi erwartet.

Beide Polarkoordinaten phi, theta können auch als Spline programmiert werden.

Eine Online-Korrektur ist für alle Koordinaten möglich.

11.10.4 Variante 3 (mit linearer Bewegung der rotatorischen Achsen)

Funktion

Name der Transformation	Transformationstyp	Wirkstelle	Koordinatensystem
5-Achs-Transformation mit Werkzeug-Dreh- und Schwenkachse	Variante 3: 3032001 mit linearer Bewegung der rotatorischen Achsen	AT2	MCS, BCS

Tab. 11-37: Funktion - Variante 3

Ausgangsgrößen im MCS: X,Y,Z,R2,R1

Eingangsgrößen im BCS: x, y, z, C, B

Die Transformation kommt an der Wirkstelle 2 zum Einsatz, d. h. sie wirkt zwischen dem Maschinenkoordinatensystem MCS = (X,Y,Z,R2,R1) und dem Basiskoordinatensystem BCS = (x,y,z,B,C).

Diese Transformation ist im wesentlichen identisch zum Typ 3232201, unterstützt jedoch keine programmierbaren Orientierungskordinaten, sondern orientiert das Werkzeug durch direktes Programmieren der rotatorischen Achsen B und C.

Die Orientierungsbewegung erfolgt linear in den Rundachspositionen.

11.10.5 Gemeinsame Eigenschaften der Varianten 1-3

Relevante NC-Funktionen

COORD(<i>)	Mit i=1..20 wird eine 5-Achs-Transformation aktiviert, die in MP 1030 00110 definiert ist.
COORD(0,2) - oder - COORD(0)	Die aktive Transformation wird deaktiviert.

Tab. 11-38: Relevante NC-Funktionen

Relevante CPL-Funktionen

SDR(300)	Wirksame Epsilon-Umgebung. Der wirksame Wert kann größer als der in EpsAxTrafo (MP 1030 00160) eingestellte Wert sein.

Tab. 11-39: Relevante CPL-Funktionen

Relevante Maschinenparameter

1030 00110	Definiert den Transformationstyp.
1030 00120	Definiert die an der Transformation beteiligten Systemachsen/-koordinaten der Transformation.
1030 00125	Definiert die Achsklassifikation der Transformationsachsen.
1030 00130	Definiert die Achspositionen der Referenzstellung.
1030 00140	Definiert die Längen- und Winkelparameter.
1030 00160	Epsilon-Umgebungen

Tab. 11-40: Relevante Maschinenparameter

Referenzstellung

In der Referenzstellung aller 5-Achskonfigurationen (im MCS: X, Y, Z, R1 = C und R2 = B) ist die Spitze (TCP) des rotationssymmetrischen Werkzeugs (Fräser oder Laser) auf dem Ursprung des Basiskoordinatensystems (BCS) entlang der z-Koordinate ausgerichtet. Die C-Achse dreht den B-Achskörper und das damit verbundene Werkzeug um die z-Koordinate. Die B-Achse dreht das Werkzeug um die y-Koordinate des BCS. Die Richtungspfeile an den Achskörpern geben jeweils die positive Drehrichtung an.

Die Ausrichtung des BCS ist durch die Achs-Transformation eindeutig festgelegt.

Der Achsvektor $\vec{r}_m$ entspricht der Distanz zwischen dem MCS-Nullpunkt und der Referenzposition (Nullpunkt des BCS).

Die Winkel, die die Rundachsen jeweils drehen müssen, um die Referenz-Orientierung von phi, theta = 0 in der Referenz-Raumposition x,y,z = 0 zu erreichen, werden auch auf den MCS-Nullpunkt bezogen.

Achsverbindungsvektoren

Die Achsverbindungsvektoren $\vec{l}_1$ und $\vec{l}_2$ definieren die Geometrie der 5-Achs-Kinematik in der Referenzstellung (siehe folgende Abb.)

- $\vec{l}_1$ ist ein Vektor entlang der X-Achse von der R1-Achse (C-Achse) zur R2-Achse (B-Achse) und hat daher nur die Komponente l_{1x} .

$$\vec{l}_1 = \begin{bmatrix} l_{1x} \\ 0 \\ 0 \end{bmatrix}$$

- $\vec{l}_2$ ist der Fortsetzungsvektor von der R2-Achse (B-Achse) zum Nullpunkt des BCS (in der Referenzstellung) und setzt sich aus bis zu 3 Komponenten l_{2x} , l_{2y} und l_{2z} zusammen. Die Richtung der Komponenten sind entgegengesetzt bzgl. der BCS-Koordinatenrichtung.

$$\vec{l}_2 = \begin{bmatrix} l_{2x} \\ l_{2y} \\ l_{2z} \end{bmatrix}$$

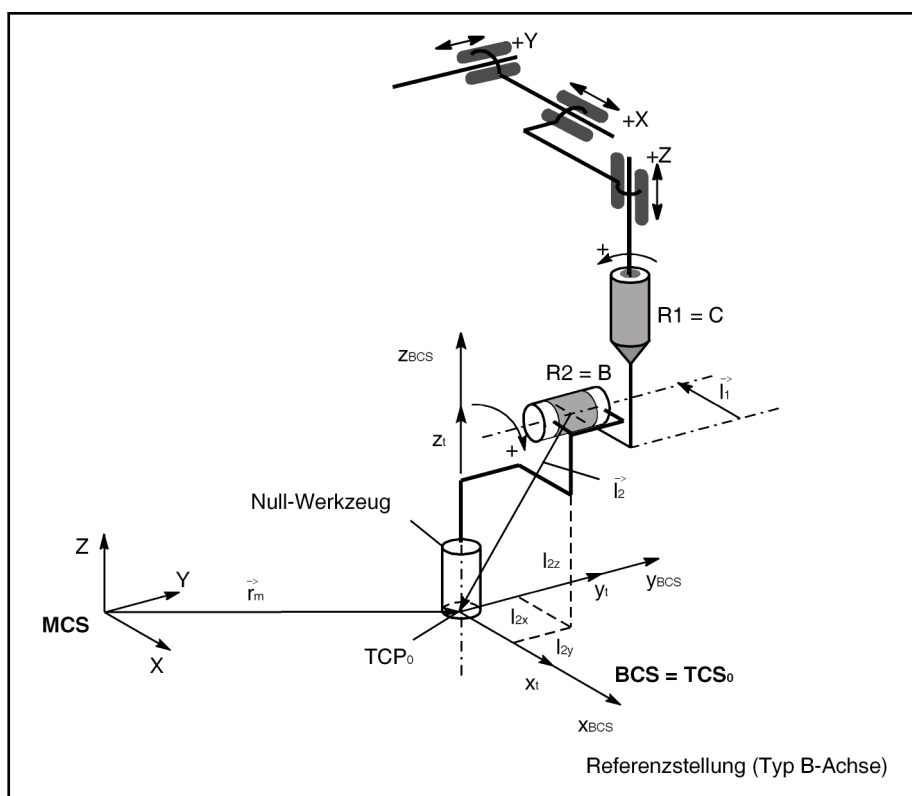
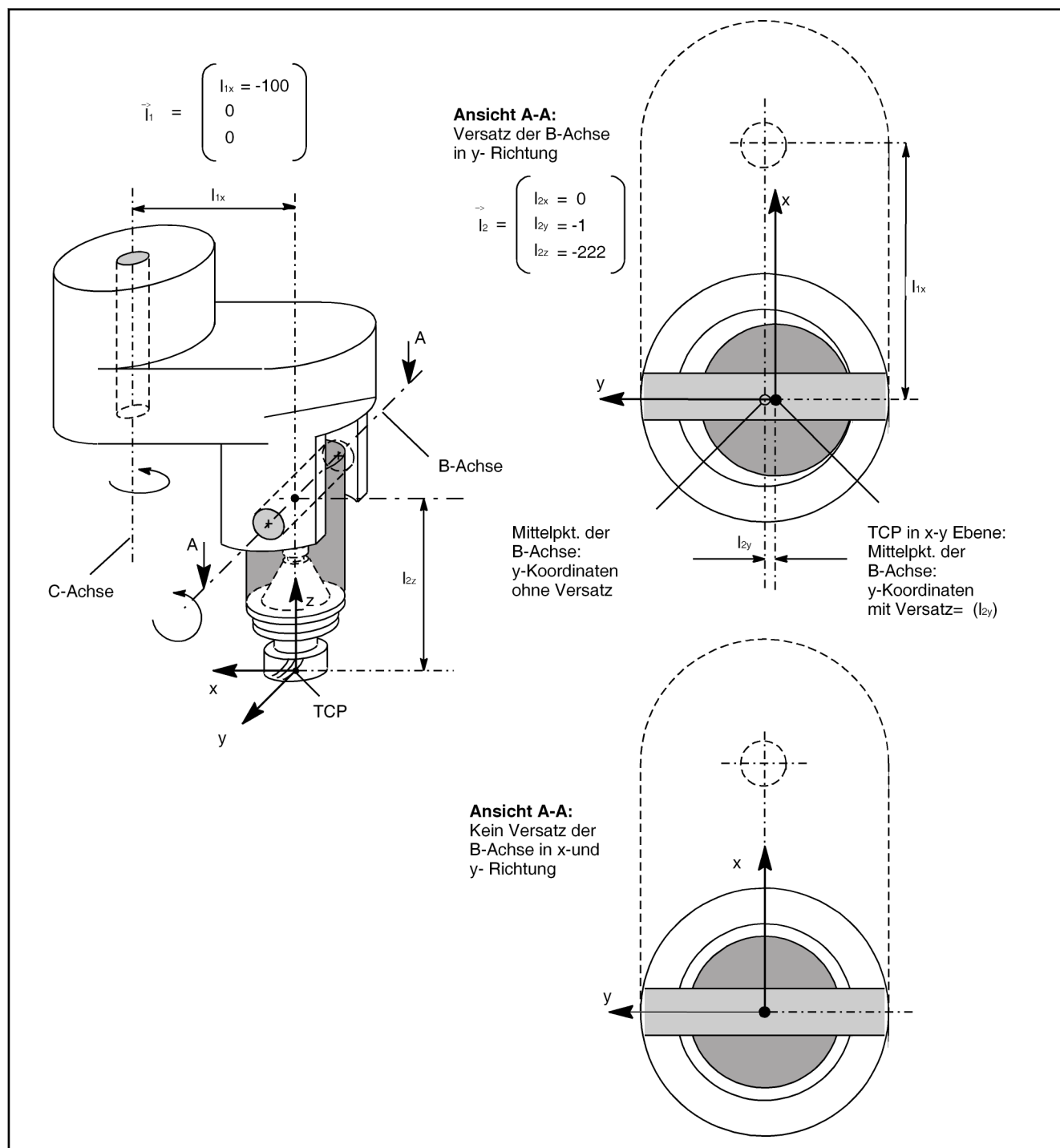


Abb. 11-33: Referenzstellung (Typ B-Achse)

Achs-Transformation

Beispiel:

5-Achsdreh-Schwenkkopf mit und ohne Versatz der B-Achse

Abb. 11-34: *Beispiel*

Besonderheiten

- Mehrdeutige Achspositionen:

Im Allgemeinen kann eine TCP-Position $\vec{r}^+$ und eine Werkzeugorientierung $\vec{\rho}$ durch zwei verschiedene Achsposen (Transformationszweige) erreicht werden. Die Zweige unterscheiden sich in den B-Achspositionen.

Welche der beiden Posen (Transformationszweige) benutzt wird, entscheidet sich beim Einschalten der Achs-Transformation. Ein Übergang von Zweig -1 nach 1 und umgekehrt ist danach nur beim Durchfahren der Orientierungsstelle $\theta = 0^\circ$ (Nordpolsingularität) möglich.

- **Nordpolsingularität:**

Bei der 5-Achs-Transformation werden Singularitäten der Orientierungsbewegung berücksichtigt. Bei nichttangentialem Durchgang von $\vec{\rho}$ durch die Stellen $\vec{\rho} = (0,0,0)$, d. h. $\theta = 0$ (Nordpolsingularität) wird ein spezieller Drehsatz für die C-Achse automatisch von der NC erzeugt.

Dreht der Orientierungsvektor nahe an der Singularität vorbei, ergibt sich eine hohe C-Achsgeschwindigkeit, d. h. die Bahngeschwindigkeit muss stark herabgesetzt werden, damit die C-Achse ihre maximal zulässige Geschwindigkeit nicht überschreitet. Um sich daraus ergebende Kriechbewegungen zu vermeiden, wird eine ε -Umgebung um die Stelle $\theta = 0^\circ$ eingeführt. Durchquert der Orientierungsvektor die ε -Umgebung, wird die Orientierungsbewegung durch den Nordpol geleitet. Der Wert für ε wird in MP hinterlegt.

- **Vorschub:**

Nach dem Einschalten der 5-Achs-Transformation wird auf Raumkoordinatenprogrammierung geschaltet. Der programmierte Vorschub (F) bezieht sich nur noch auf die programmierbaren Positionskoordinaten, d. h. mit dem F-Wort wird die Bahngeschwindigkeit des Werkzeugmittelpunktes (TCP) programmiert. Zusätzliche Orientierungs- und Pseudokoordinatenbewegungen verändern diese Bahngeschwindigkeit nicht. Orientierungs- und Pseudokoordinatenbewegung werden synchron mitgeführt, d. h., die Endposition wird für alle Koordinaten zeitgleich erreicht. Die mitgeführte Bewegung der Orientierungs- und Pseudokoordinaten kann jedoch zu einer zusätzlichen Limitierung der Bahnkinematik (maximale Bahngeschwindigkeit und -Beschleunigung) führen, da die Grenzwerte für alle an der Bewegung beteiligten Achsen überwacht werden.

TCS, TCP

TCS bezeichnet das Werkzeugkoordinatensystem.

TCP_0 / TCP_1 bezeichnet den Tool Center Point (Werkzeugmittelpunkt) **ohne** / **mit** Korrektur.

Werkzeugkorrektur

Die Achs-Transformation unterstützt eine Werkzeugkorrektur im Werkzeugkoordinatensystem (TCS). In der Referenzstellung stimmen BCS und TCS_0

überein. Der Werkzeugkorrekturvektor $\vec{t}_t$ zeigt vom korrigierten Werkzeugmittelpunkt TCP_1 zum unkorrigierten TCP_0 . Er setzt sich aus den Längenkorrekturen L1, L2 und L3 aus der D-Korrekturtabelle zusammen.

Beispiel:

zur Werkzeugkorrektur

Eine Fräser-Werkzeugkorrektur (Länge, Radius) bedeutet hierbei eine räumliche Verschiebung (um $L3$ und $L1$) des TCP entlang eines Korrekturvektors $\vec{l}_t$ vom TCP_1 zum TCP_0 (positiver $L3$ -Wert verlängert das Werkzeug).

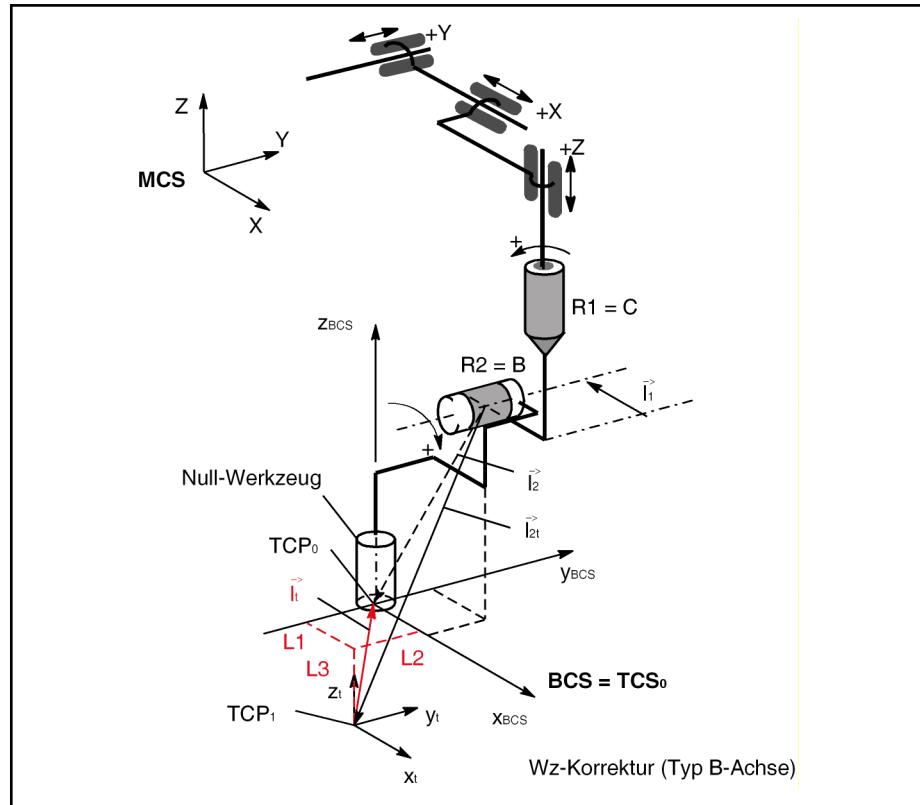


Abb. 11-35: Werkzeugkorrektur Typ B-Achse

11.10.6 Handlungsanweisung: 5-Achs-Transformation LLLRR

Applizieren Die Achstransformation kann über die relevanten Maschinenparameter appliziert werden. Dazu muss die Funktion Achstransformation über den Parameter **tra** "Achstransformationen" im Setup (SUP) aktiviert werden. Der Pfad des MP-Namens ist immer "TRA/AxTrafo[1..20]/...".

Alternativ dazu kann die Achstransformation über ein benutzerdefiniertes Systemdatum vom Typ SysAxTrafo_t appliziert werden.

Die beiden Varianten sind inhaltlich identisch.

Die Achspositionen der Referenzstellung müssen ermittelt werden.

In der nachfolgenden Tabelle sind jeweils die äquivalenten MP-Nummern, MP-Name und SD-Elemente angegeben. Um eine Transformation zu applizieren, muss eine der Varianten entsprechend besetzt werden.

Transformations-relevante Parameter

MP-Nummer	Beschreibung	
1030 00110	Transformationstyp MP-Name: TrafoType SD-Elem. : Type	
	3232201	LLLRR mit Vektor-Orientierung
	3232101	LLLRR mit Linear-Orientierung
	3032001	LLLRR mit Linear-Bewegung der rotatorischen Achsen
1030 00120	Systemachsen/-koordinaten der Transformation MP-Name: FwdInCoordIndTrafo[1..8] SD-Elem. : AxisAssignment[1..8]	
	[1]	Systemachsnummer von X
	[2]	Systemachsnummer von Y
	[3]	Systemachsnummer von Z
	[4]	Systemachsnummer von C
	[5]	Systemachsnummer von B
	[6..8]	nicht relevant

MP-Nummer	Beschreibung																												
1030 00125	Achsklassifikationen der Transformationsachsen MP-Name: CoordClass[1..8] MP-Name: CoordDir[1..8] SD-Elem. : AxisClassification[1..8] Definiert die an der Transformation beteiligten Achsen inklusive Bewegungsrichtung. - MP-Nr. oder SD-Element: Wert ">0" bedeutet: Die Achse verfährt/dreht positiv im Sinne der obigen Abbildungen. Wert "<0" bedeutet: Die Achse verfährt/dreht negativ im Sinne der obigen Abbildungen. Mögliche Werte sind: +/-1 = X-Achse +/-2 = Y-Achse +/-3 = Z-Achse +/-100 = A-Achse +/-200 = B-Achse +/-300 = C-Achse - MP-Name: - Mögliche Werte für CoordClass sind - X,Y,Z (linear) - A,B,C (rotatorisch) - Mögliche Werte für CoordDir sind: - positiv (bzgl. Abb. "Referenzstellung") - negativ (bzgl. Abb. "Referenzstellung") Ist ein Parameter unbesetzt, gelten folgende Default-Einstellungen: <table><tr><th></th><th>MP-Nr / SD-Elem.:</th><th>CoordClass:</th><th>CoordDir:</th></tr><tr><td>[1]</td><td>1</td><td>X</td><td>positiv</td></tr><tr><td>[2]</td><td>2</td><td>Y</td><td>positiv</td></tr><tr><td>[3]</td><td>3</td><td>Z</td><td>positiv</td></tr><tr><td>[4]</td><td>300</td><td>C</td><td>positiv</td></tr><tr><td>[5]</td><td>200</td><td>B</td><td>positiv</td></tr><tr><td>[6..8]</td><td>nicht relevant</td><td>0</td><td>positiv</td></tr></table>		MP-Nr / SD-Elem.:	CoordClass:	CoordDir:	[1]	1	X	positiv	[2]	2	Y	positiv	[3]	3	Z	positiv	[4]	300	C	positiv	[5]	200	B	positiv	[6..8]	nicht relevant	0	positiv
	MP-Nr / SD-Elem.:	CoordClass:	CoordDir:																										
[1]	1	X	positiv																										
[2]	2	Y	positiv																										
[3]	3	Z	positiv																										
[4]	300	C	positiv																										
[5]	200	B	positiv																										
[6..8]	nicht relevant	0	positiv																										

MP-Nummer	Beschreibung
1030 00130	Achspositionen der Referenzstellung MP-Name: RefPosTrafo[1..8] SD-Elem. : AxZeroPos[1..8]
	[1] X^0
	[2] Y^0
	[3] Z^0
	[4] C^0
	[5] B^0
	[6..8] nicht relevant
1030 00140	Längen- und Winkelparameter MP-Name: JointParTrafo[1..16] SD-Elem. : LenParam[1..16] Definiert die Längenvektoren l_1, l_2 in [mm].
	[1] l_{1x}
	[2] nicht relevant
	[3] nicht relevant
	[4] l_{2x}
	[5] l_{2y}
	[6] l_{2z}
	[7..16] nicht relevant
1030 00160	Epsilon-Umgebungen MP-Name: EpsAxTrafo SD-Elem. : EpsilonRanges
	0.01

Tab. 11-41: Transformations-relevante Parameter

Aktivieren

Ist die 5-Achs-Transformation konfiguriert per

- Maschinenparameter, dann wird die Transformation im NC-Programm aktiviert mit (i=1,..20):

COORD(<i>)

- Systemdatum, dann wird die Transformation im NC-Programm aktiviert mit (Typ "SysAxTrafo_t"):

COORD(<SD.MtbAxTrafo>)**Programmbeispiel:**

Die 5-Achs-Transformation ist im 4. Maschinenparametersatz definiert. Am Programmstart ist keine Achstransformation aktiv:

Transformation aktivieren

```

N100 G1 F1000 XA10 YA10 C30 ;Axes XA, YA and C are traversed
N110 COORD(4) ;Switching on the axis transformation
N130 x20 y30 phi0 ;the coordinates x,y,phi are traversed
N140 x10 z10 theta45 ;the coordinates x,z,theta are traversed
;
```

N999 COORD(0)

;Switching-off the axis transformation
;Axes are again programmed from here**Deaktivieren** COORD(0) bzw. COORD(0,2)

schaltet die Achstransformation im NC-Programm aus.

11.10.7 Variante 4: 5-Achs-Transformation LLLAB

Funktion

Name der Transformation	Transformationstyp	Wirkstelle
LLLAB		
Vektor-Orientierung	3232201	AT2
Linear-Orientierung	3232101	AT2

Die Kinematik der 5-Achs-Transformation LLLAB besteht aus drei Linearachsen X, Y, Z und zwei rotatorischen Achsen R_1 und R_2 . Die Transformation kommt an der Wirkstelle 2 zum Einsatz, d. h. sie wirkt zwischen dem Maschinen-Koordinatensystem MCS = (X,Y,Z, R_1 , R_2) und dem Basis-Koordinatensystem BCS = (x,y,z,phi,theta).

- Programmiert werden die BCS-Eingangsgrößen x, y, z, phi, theta
- Berechnet werden die MCS-Ausgangsgrößen X, Y, Z, R_1 , R_2

Die Linearachsen spannen ein rechtshändiges Koordinatensystem auf. Die Rotationsachse R_1 (Typ A-Achse) dreht das Werkzeug um die X-Achse und die Rotationsachse R_2 (Typ B-Achse) schwenkt das Werkzeug um die Y-Achse (siehe nachfolgende Abbildung).

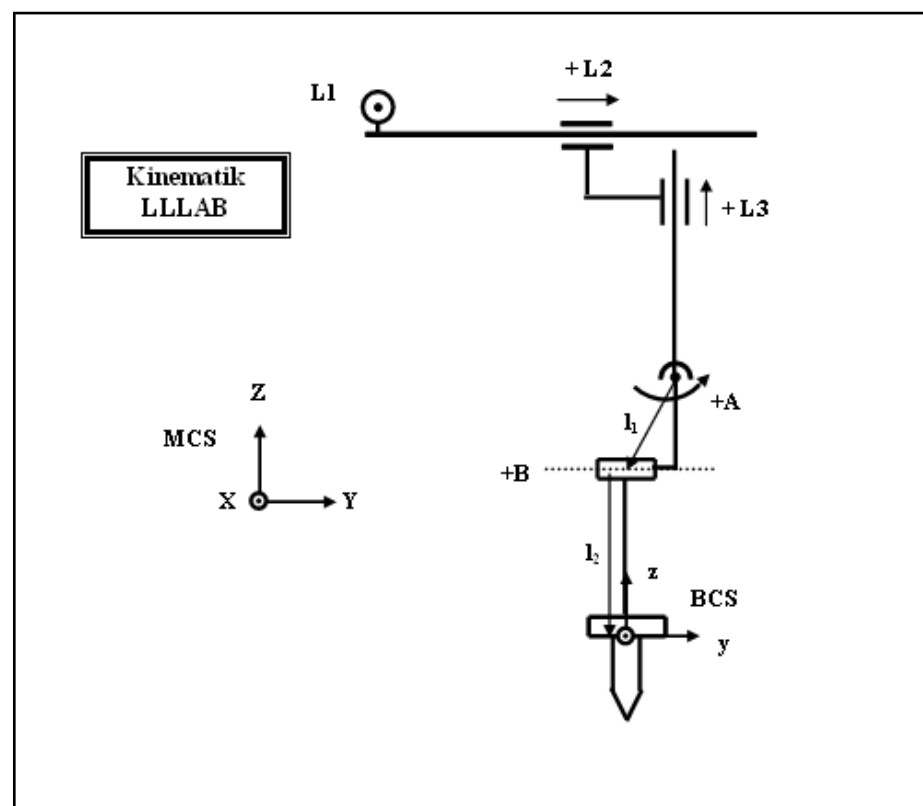


Abb. 11-36: Funktion 5-Achs-Transformation LLLAB

Referenzstellung

In der Referenzstellung steht die Werkzeugaufnahme (Spindelnase) im Nullpunkt des Basis-Koordinatensystems (BCS) und ist entlang der z-Koordinate ausgerichtet, d. h. das Werkzeug zeigt senkrecht nach unten (vergl. Abbildung).

In dieser Stellung sind die BCS-Koordinaten $(x,y,z,\phi,\theta) = (0,0,0,0,0)$. Um dies zu erreichen, müssen die in dieser Stellung angezeigten MCS-Koordinaten $(X^0,Y^0,Z^0,R_1^0,R_2^0)$ als Referenzposition in den Maschinenparametern eingetragen werden (s. u.).

Die positiven Dreh-/Verfahr-Richtungen der Achsen sind in der Abbildung dargestellt. Diese sind entsprechend einzustellen! In der Referenzstellung dreht die A-Achse um die x-Koordinate und die B-Achse um die y-Koordinate.

Die Vektoren l_1 und l_2 definieren die Geometrie der 5-Achs-Kinematik. Sie haben die Form

$$\vec{l}_1 = \begin{pmatrix} 0 \\ l_{1y} \\ l_{1z} \end{pmatrix} \quad \text{und} \quad \vec{l}_2 = \begin{pmatrix} l_{2x} \\ l_{2y} \\ l_{2z} \end{pmatrix}$$

Abb. 11-37: Form der Vektoren l_1 und l_2

l_1 ist ein Vektor von der A-Achse zur B-Achse.

l_2 ist der l_1 -Fortsetzungsvektor von der B-Achse zum Nullpunkt des BCS.

Orientierung

Mit Einschalten der 5-Achs-Transformation LLLAB wird entweder die Vektor-Orientierungsbewegung (Typ 3232202) oder die lineare Orientierungsbewegung (Typ 3232102) aktiviert. In Abhängigkeit vom Orientierungstyp ergeben sich unterschiedliche Eigenschaften.

Vektor-Orientierung

Bei der Vektor-Orientierungsbewegung erfolgt jede Orientierungsänderung durch Programmierung von ϕ , θ oder einer entsprechenden Alternativsyntax (siehe Spez. Orientierungsbewegung) durch eine Drehung des Orientierungsvektors um eine raumfeste Drehachse (Großkreis-Bewegung der Vektorspitze).

ϕ , θ können Werte aus folgenden Bereichen annehmen:

$$0 \leq \phi < 360 ; 0 \leq \theta < 180$$

Die rotatorischen Achsen können sowohl Endlos- als auch Rundachsen sein.

Linear-Orientierung

Die Winkel ϕ und θ werden linear vom Anfangs- zum Endpunkt interpoliert. Das Verhalten der "kürzeste Weg"-Logik hängt von der Programmierungsart der Orientierungsbewegung ab. Bei ϕ -, θ -Programmierung ist die "kürzeste Weg"-Logik ausgeschaltet. Bei der O(...)-Programmierung ist die "kürzeste Weg"-Logik eingeschaltet.

ϕ , θ können Werte aus den folgenden Bereichen annehmen:

$$-\infty < \phi < \infty ; -180 \leq \theta < 180$$

Die rotatorischen Achsen müssen als Rundachsen konfiguriert sein (keine Modulo-Endlosachse).



Die Linear-Orientierung ist nicht "Schiefe-Ebenen"-tauglich. Programmierte Orientierungen beziehen immer auf das Basis-Koordinatensystem BCS und nicht, wie sonst üblich, auf das Werkstück-Koordinatensystem WCS!

Besonderheiten

Beschränkung der B-Achse

Der Verfahrbereich der B-Achse wird auf $|B| \leq 90^\circ$ beschränkt. Mit dieser Einschränkung ist die Rückwärtstransformation BWD eindeutig lösbar.

Werkzeugkorrektur

Die Achstransformation unterstützt eine Werkzeugkorrektur im Werkzeugkoordinatensystem TCS (Tool-Coordinate-System). In der Referenzstellung stimmen BCS und TCS₀ (Spindelnase) überein. Der Werkzeugkorrekturvektor I_t zeigt vom korrigierten Werkzeugmittelpunkt TCP₁ zum unkorrigierten TCP₀. Er setzt sich aus den Längskorrekturen L1, L2 und L3 zusammen.

Restriktionen

Eine Online-Korrektur ist ausschließlich für x, y, z möglich, nicht für phi, theta. Der Verfahrbereich der B-Achse ist auf $|B| \leq 90^\circ$ beschränkt.

Relevante NC-Funktionen

COORD(...)	Die 5-Achs-Transformation LLLAB wird aktiviert. Dies kann wahlweise per Maschinenparametersatz (i=1..20) oder per Mtb-Systemdatum (vom Typ SysAxTrafo_t) erfolgen.
COORD(0) - oder - COORD(0,2)	Die aktive Transformation wird deaktiviert.

Tab. 11-42: Relevante NC-Funktionen

Relevante Maschinenparameter

In den Maschinenparametern können bis zu 20 unterschiedliche Achstransformationen per nachfolgendem Maschinenparametersatz deklariert werden.

1030 00110	Definiert den Transformationstyp
1030 00120	Definiert die an der Transformation beteiligten Systemachsen/-koordinaten der Transformation
1030 00125	Definiert die Achsklassifikation der Transformationsachsen
1030 00130	Definiert die Achspositionen der Referenzstellung
1030 00140	Definiert die Längen- und Winkelparameter
1030 00160	Epsilon-Umgebungen

Tab. 11-43: Relevante Maschinenparameter

Relevante Systemdaten

Die Achstransformation kann alternativ zu den Maschinenparameter auch per Systemdatum definiert werden. Dazu muss der Benutzer eine Systemvariable vom Typ "SysAxTrafo_t" anlegen (siehe Dokumentation "IndraMotion MTX 14VRS Maschinenparameter", Kapitel "Systemdaten"). Der Strukturtyp "SysAxTrafo_t" ist in der XML-Schemadatei "/feprom/schemas/sdaxtrf.xsd" steuerungsintern definiert und enthält, analog zu den relevanten Maschinenparametern, die nachfolgenden transformationsrelevanten Strukturelemente.

Strukturelement	Beschreibung
Type	Definiert den Transformationstyp
AxisAssignment	Definiert die an der Transformation beteiligten Systemachsen/-koordinaten der Transformation
AxisClassification	Definiert die Achsklassifikation der Transformationsachsen
AxZeroPos	Definiert die Achspositionen der Referenzstellung
LenParam	Definiert die Längen- und Winkelparameter
EpsilonRanges	Epsilon-Umgebungen

Tab. 11-44: Relevante Systemdaten

Handlungsanweisung: 5-Achs-Transformation LLLAB

Applizieren Die Achstransformation kann über die relevanten Maschinenparameter appliziert werden. Dazu muss die Funktion Achstransformation über den Parameter **tra** "Achstransformationen" im Setup (SUP) aktiviert werden. Der Pfad des MP-Namens ist immer "TRA/AxTrafo[1..20]/...".

Alternativ dazu kann die Achstransformation über ein benutzerdefiniertes Systemdatum vom Typ SysAxTrafo_t appliziert werden.

Die beiden Varianten sind inhaltlich identisch.

Die Achspositionen der Referenzstellung müssen ermittelt werden.

In der nachfolgenden Tabelle sind jeweils die äquivalenten MP-Nummern, MP-Name und SD-Elemente angegeben. Um eine Transformation zu applizieren, muss eine der Varianten entsprechend besetzt werden.

Transformations-relevante Parameter

MP-Nummer	Beschreibung	
1030 00110	Transformationstyp MP-Name: TrafoType SD-Elem. : Type	
	3232201	5-Achs-Trafo LLLAB mit Vektor-Orientierung
	3232101	5-Achs-Trafo LLLAB mit Linear-Orientierung
1030 00120	Systemachsen/-koordinaten der Transformation MP-Name: FwdInCoordIndTrafo[1..8] SD-Elem. : AxisAssignment[1..8]	
	[1]	Systemachsnummer von X
	[2]	Systemachsnummer von Y
	[3]	Systemachsnummer von Z
	[4]	Systemachsnummer von A
	[5]	Systemachsnummer von B
	[6..8]	nicht relevant

Achs-Transformation

MP-Nummer	Beschreibung																												
1030 00125	Achsklassifikationen der Transformationsachsen MP-Name: CoordClass[1..8] MP-Name: CoordDir[1..8] SD-Elem. : AxisClassification[1..8] Definiert die an der Transformation beteiligten Achsen inklusive Bewegungsrichtung - MP-Nr. oder SD-Element: Wert ">0" bedeutet: Die Achse verfährt/dreht positiv im Sinne der obigen Abbildung. Wert "<0" bedeutet: Die Achse verfährt/dreht negativ im Sinne der obigen Abbildungen. Mögliche Werte sind: +/-1 = X-Achse +/-2 = Y-Achse +/-3 = Z-Achse +/-100 = A-Achse +/-200 = B-Achse - - MP-Name: - Mögliche Werte für CoordClass sind - X,Y,Z (linear) - A,B,C (rotatorisch) - Mögliche Werte für CoordDir sind: - positiv (bzgl. Abb. "Referenzstellung") - negativ (bzgl. Abb. "Referenzstellung") Stimmen die Verfahrrichtungen mit der Abbildung überein, sind die Parameter folgendermaßen zu setzen: <table><tr><th></th><th>MP-Nr / SD-Elem.:</th><th>CoordClass:</th><th>CoordDir:</th></tr><tr><td>[1]</td><td>1</td><td>X</td><td>positiv</td></tr><tr><td>[2]</td><td>2</td><td>Y</td><td>positiv</td></tr><tr><td>[3]</td><td>3</td><td>Z</td><td>positiv</td></tr><tr><td>[4]</td><td>100</td><td>A</td><td>positiv</td></tr><tr><td>[5]</td><td>200</td><td>B</td><td>positiv</td></tr><tr><td>[6..8]</td><td>nicht relevant</td><td>0</td><td>positiv</td></tr></table>		MP-Nr / SD-Elem.:	CoordClass:	CoordDir:	[1]	1	X	positiv	[2]	2	Y	positiv	[3]	3	Z	positiv	[4]	100	A	positiv	[5]	200	B	positiv	[6..8]	nicht relevant	0	positiv
	MP-Nr / SD-Elem.:	CoordClass:	CoordDir:																										
[1]	1	X	positiv																										
[2]	2	Y	positiv																										
[3]	3	Z	positiv																										
[4]	100	A	positiv																										
[5]	200	B	positiv																										
[6..8]	nicht relevant	0	positiv																										

MP-Nummer	Beschreibung
1030 00130	Achspositionen der Referenzstellung MP-Name: RefPosTrafo[1..8] SD-Elem. : AxZeroPos[1..8]
	[1] X^0
	[2] Y^0
	[3] Z^0
	[4] A^0
	[5] B^0
	[6..8] nicht relevant
1030 00140	Längen- und Winkelparameter MP-Name: JointParTrafo[1..16] SD-Elem. : LenParam[1..16] Definiert die Längenvektoren l_1, l_2 in [mm].
	[1] nicht relevant
	[2] l_{1y}
	[3] l_{1z}
	[4] l_{2x}
	[5] l_{2y}
	[6] l_{2z}
	[7..16] nicht relevant
1030 00160	Epsilon-Umgebungen MP-Name: EpsAxTrafo SD-Elem. : EpsilonRanges
	nicht relevant

Tab. 11-45: Transformations-relevante Parameter

Aktivieren

Ist die 5-Achs-Transformation konfiguriert per

- Maschinenparameter, dann wird die Transformation im NC-Programm aktiviert mit (i=1,..20):

COORD(<i>)

- Systemdatum, dann wird die Transformation im NC-Programm aktiviert mit (Typ "SysAxTrafo_t"):

COORD(<SD.MtbAxTrafo>)**Programmbeispiel:**

Die 5-Achs-Transformation ist im 4. Maschinenparametersatz definiert. Am Programmstart ist keine Achstransformation aktiv:

Transformation aktivieren

```

N100 G1 F1000 XA10 YA10 A30 ;Axes XA,YA and A are traversed
N110 COORD(4) ;Activating axis transformation
N130 x20 y30 phi0 ;Coordinates x,y,phi are traversed
N140 x10 z10 theta45 ;Coordinates x,z,theta are traversed :
N999 COORD(0) ;Deactivating axis transformation
;from here, axes are programmed

```

Deaktivieren COORD(0) bzw. COORD(0,2)

schaltet die Achstransformation im NC-Programm aus.

11.11 5-Achs-Transformation - Typ 2: RLLLR (mit Rundtisch)

11.11.1 Beschreibung

Funktion

Name der Transformation	Transformationstyp	Wirkstelle	Koordinatensystem
5-Achs-Transformation mit Rundtisch			
Vektororientierung	3232202	AT2	MCS, BCS
lineare Orientierung	3232102	AT2	MCS, BCS

Tab. 11-46: Funktion

Ausgangsgrößen im MCS:

X,Y,Z,R2,R1

Eingangsgrößen im BCS:

x, y, z, φ , ϑ

Die verteilte 5-Achs-Transformation (RLLLR) besitzt drei Linearachsen (X, Y, Z) und zwei rotatorische Achsen (R1, R2).

Die rotatorischen Achsen R1 und R2 können sowohl Endlos- als auch Rundachsen sein.

Sie kommt an der Wirkstelle 2 zum Einsatz, d. h. sie wirkt zwischen dem Maschinenkoordinatensystem MCS = (X,Y,Z,R2,R1) und dem Basiskoordinatensystem BCS = (x, y, z, φ , ϑ).

Die drei Linearachsen spannen ein rechtshändiges Koordinatensystem auf. Die Rotationsachse R_2 schwenkt das Werkzeug und die Rotationsachse R_1 dreht das (auf der Achse aufgespannte) Werkstück.

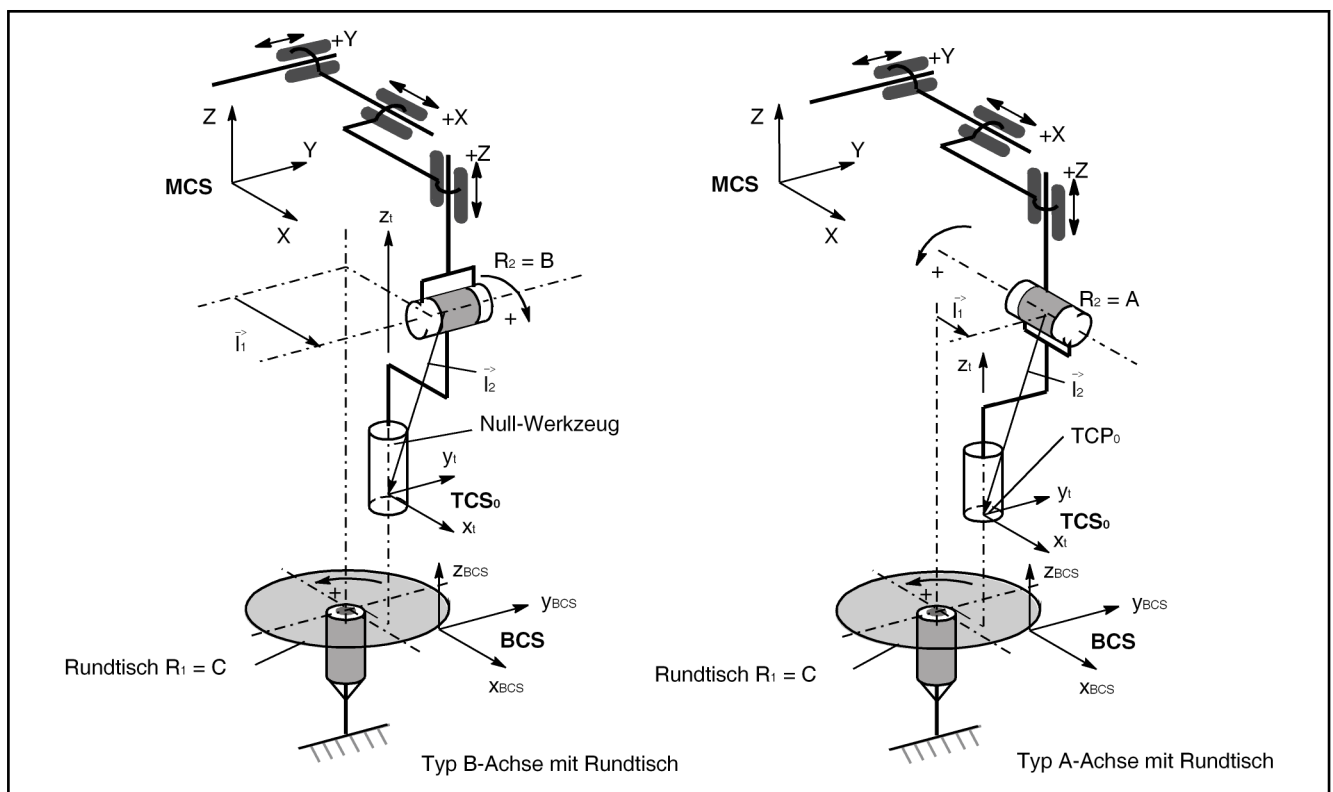


Abb. 11-38: RLLLR (mit Rundtisch)

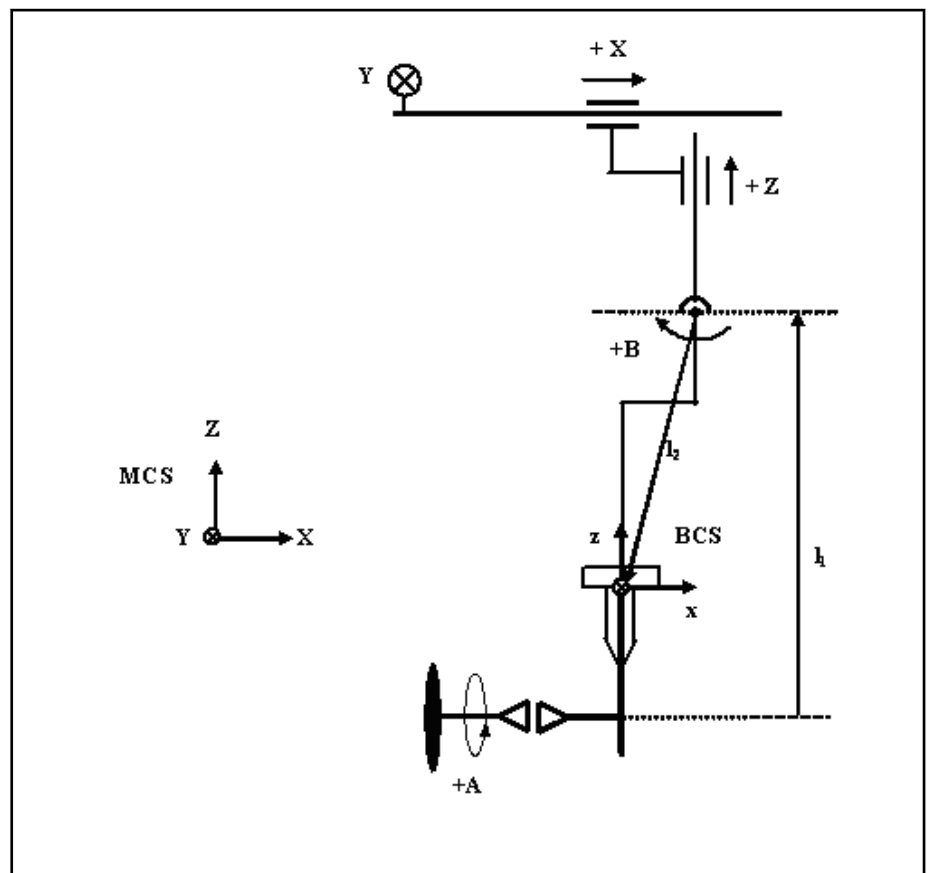


Abb. 11-39: ALLLB

Die Vektoren $\vec{i}_1$ und $\vec{i}_2$ definieren die Geometrie der 5-Achs-Kinematik. $\vec{i}_1$ ist ein Vektor von der R_1 -Achse (C-/A-Achse) zur R_2 -Achse (B-/A-Achse), d. h. im Fall $R_1=C$ ist $\vec{i}_1$ ein Vektor entlang x_{BCS} und im Fall $R_1=A$ entlang z_{BCS} . $\vec{i}_2$ ist der Fortsetzungsvektor von der R_2 -Achse (B-/A-Achse) zum Nullpunkt des BCS (in der Referenzstellung).

Es gibt grundsätzlich **drei verschiedene Varianten** der verteilten 5-Achs-Transformation RLLLR. Diese unterscheiden sich durch den Typ der Rundachsen (der Typ hängt davon ab, um welche der BCS-Koordinaten x,y,z die Rundachse in der Referenzstellung dreht):

- **$R_1 = C\text{-Achse}, R_2 = A\text{-Achse}$:**
 R_1 dreht in der Referenzstellung um die z-Koordinate und R_2 dreht in der Referenzstellung um die x-Koordinate.
- **$R_1 = C\text{-Achse}, R_2 = B\text{-Achse}$:**
 R_1 dreht in der Referenzstellung um die z-Koordinate und R_2 dreht in der Referenzstellung um die y-Koordinate.
- **$R_1 = A\text{-Achse}, R_2 = B\text{-Achse}$:**
 R_1 dreht in der Referenzstellung um die x-Koordinate und R_2 dreht in der Referenzstellung um die y-Koordinate.

Zu diesen Achsanordnungen existieren noch weitere **Untervarianten**, die sich jeweils nur dadurch unterscheiden, dass der Rundtisch R_1 auf einer der Linearachsen X oder Y montiert ist. Sie sind aber immer einer der drei obigen Varianten zuzuordnen und unterscheiden sich letztendlich nur durch die Verfahrrichtung der R_1 -bewegenden Linearachse.

In der folgenden Abbildung ist exemplarisch die Variante $R_1 = C\text{-Achse}$ auf X-Achse, $R_2 = B\text{-Achse}$ dargestellt. Da sich der Rundtisch R_1 auf der linearen X-Achse bewegt, dreht sich die positive Verfahrrichtung der X-Achse um.

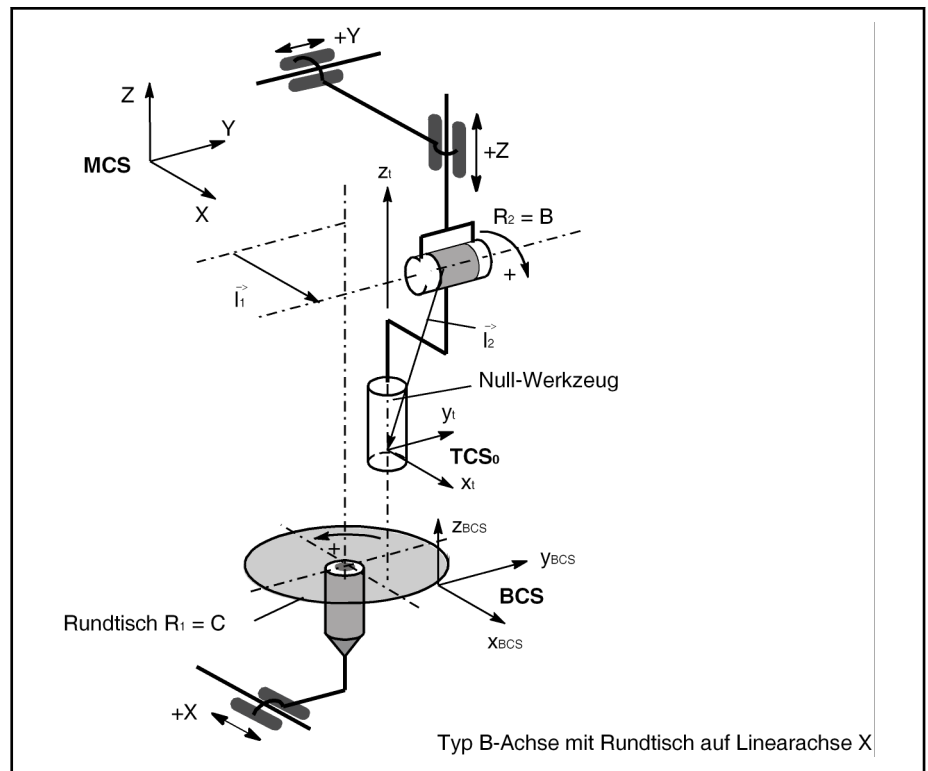


Abb. 11-40: Typ B-Achse mit Rundtisch auf Linearachse X

Eigenschaften der Achs-Transformation:

Bei aktiver 5-Achs-Transformation werden die Koordinaten x, y, z, φ, θ des Basiskoordinatensystem BCS in das Maschinenkoordinatensystem MCS transformiert.

Eingangsgrößen sind die Linearkoordinaten x , y , z und die Orientierungs-
koordinaten φ , θ . Mit Einschalten dieser Achs-Transformation wird entweder
die Vektororientierungsbewegung (Typ 3232202) oder die lineare Orientie-
rungsbewegung (Typ 3232102) aktiviert.

Der Orientierungsvektor $\vec{\phi}$ kann als Spline programmiert werden (siehe [Kap. 9.1.8 "Spline programmieren"](#) auf Seite 148).

In Abhängigkeit vom Orientierungstyp ergeben sich unterschiedliche Eigenschaften

Vektor-Orientierung:

Bei der Vektororientierungsbewegung erfolgt jede Orientierungsänderung durch Programmierung von φ , θ oder einer entsprechenden Alternativsyntax (s. Spez. Orientierungsbewegung) durch eine Drehung des Orientierungsvek-

tors $\vec{\rho}(\varphi, \theta)$ um eine raumfeste Drehachse (Großkreis-Bewegung der Vektorspitze). Die rotatorischen Achsen können sowohl Endlos- als auch Rundachsen sein.

Linear-Orientierung:

Die Winkel φ und θ werden linear vom Anfangs- zum Endpunkt interpoliert. Das Verhalten der "kürzeste Weg"-Logik hängt von der Programmierungsart der Orientierungsbewegung ab. Bei φ -, θ -Programmierung ist die "kürzeste Weg"-Logik ausgeschaltet. Bei der $O(\dots)$ -Programmierung ist die "kürzeste Weg"-Logik eingeschaltet. φ , θ können Werte aus den folgenden Bereichen annehmen:

$\varphi \in [-\text{unendl}, +\text{unendl}]$, $\vartheta \in [-180, +180]$.

Die rotatorischen Achsen müssen als Rundachsen konfiguriert sein (keine Modulo-Endlosachse).

Restriktion

Eine Online-Korrektur für φ und ϑ ist nicht möglich.

Relevante NC-Funktionen

COORD(<i>)	Mit i=1..20 wird eine 5-Achs-Transformation aktiviert, die in MP 1030 00110 definiert ist.
COORD(0,2) - oder - COORD(0)	Die aktive Transformation wird deaktiviert.

Tab. 11-47: Relevante NC-Funktionen

Relevante CPL-Funktionen

SDR(300)	Wirksame Epsilon-Umgebung. Der wirksame Wert kann größer als der in EpsAxTrafo (MP 1030 00160) eingestellte Wert sein.

Tab. 11-48: Relevante CPL-Funktionen

Relevante Maschinenparameter

1030 00110	Definiert den Transformationstyp.
1030 00120	Definiert die an der Transformation beteiligten Systemachsen/-koordinaten der Transformation.
1030 00125	Definiert die Achsklassifikation der Transformationsachsen.
1030 00130	Definiert die Achspositionen der Referenzstellung.
1030 00140	Definiert die Längen- und Winkelparameter.
1030 00160	Epsilon-Umgebungen

Tab. 11-49: Relevante Maschinenparameter

Referenzstellung

In der Referenzstellung der folgenden Konfiguration (im MCS: X, Y, Z, R1 = C und R2 = B) ist die Spitze (TCP) des rotationssymmetrischen Werkzeugs (Fräser oder Laser) auf dem Ursprung des Basiskoordinatensystems (BCS) entlang der z-Koordinate ausgerichtet. Die Ausrichtung des BCS ist durch die Achs-Transformation eindeutig festgelegt.

In dieser Stellung müssen die BCS-Koordinaten $(x, y, z, \varphi, \vartheta) = (0, 0, 0, 0, 0)$ sein, bei zugehörigen MCS-Koordinaten $(X, Y, Z, R2, R1) = (X^0, Y^0, Z^0, R2^0, R1^0)$ aus MP 1030 00130.

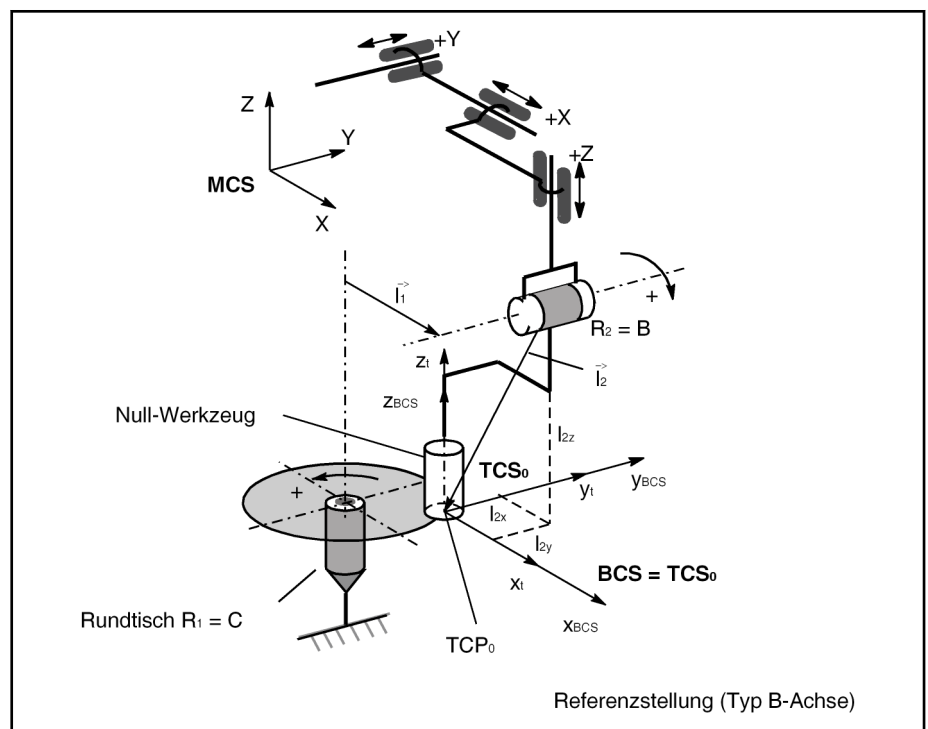


Abb. 11-41: Referenzstellung

TCS, TCP

TCS bezeichnet das Werkzeugkoordinatensystem.

TCP_0/TCP_1 bezeichnet den Tool Center Point (Werkzeugmittelpunkt) **ohne/mit** Korrektur.

Werkzeugkorrektur

Die Achs-Transformation unterstützt eine Werkzeugkorrektur im Werkzeugkoordinatensystem (TCS). In der Referenzstellung stimmen BCS und TCS_0

überein. Der Werkzeugkorrekturvektor $\vec{l}_t$ zeigt vom korrigierten Werkzeugmittelpunkt TCP_1 zum unkorrigierten TCP_0 . Er setzt sich aus den Längenkorrekturen L_1 , L_2 und L_3 aus der D-Korrekturtabelle zusammen.

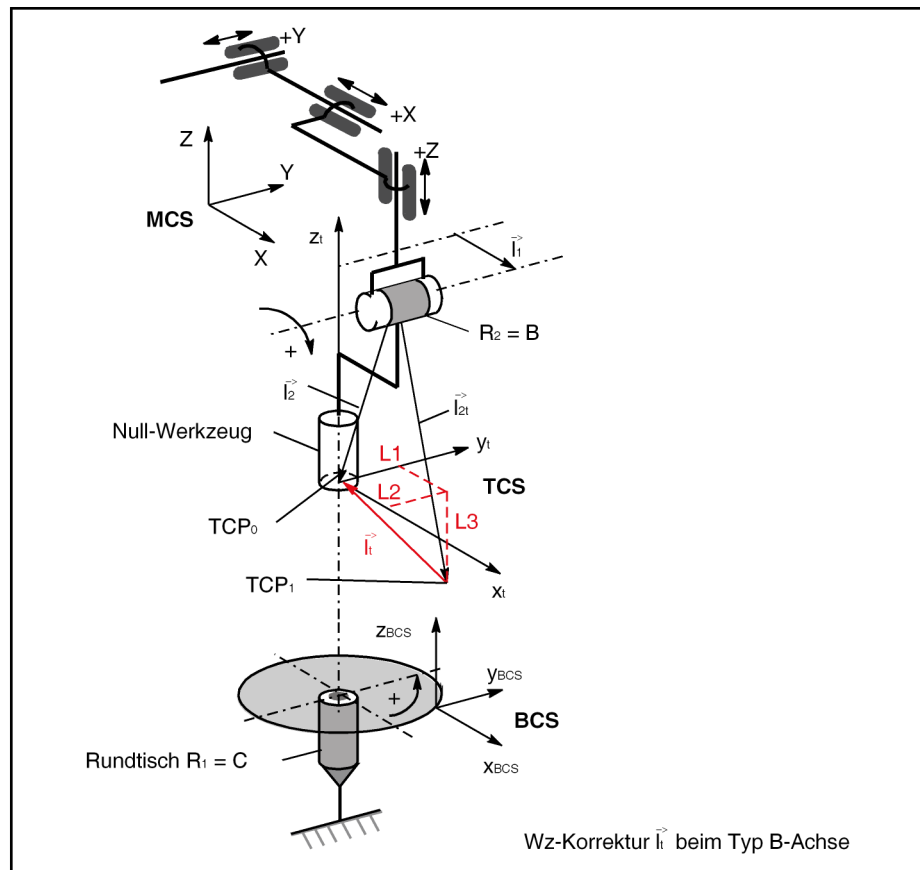


Abb. 11-42: Werkzeugkorrektur

Besonderheiten

- Mehrdeutige Achspositionen:**

Im Allgemeinen kann eine TCP-Position und eine Werkzeugorientierung durch zwei verschiedene Achsposen (Transformationszweige) erreicht werden. Die Zweige unterscheiden sich in den B-Achspositionen (bzw. A-Achsposition).

Vektor-Ori:

Welche der beiden Posen (Transformationszweige) benutzt wird, entscheidet sich beim Einschalten der Achs-Transformation. Ein Übergang von Zweig 1 nach -1 und umgekehrt ist danach nur beim Durchfahren der Orientierungsstelle $\theta = 0$ (Nordpolsingularität) möglich.

Linear-Ori:

Der Transformationszweig wird immer so gewählt, dass die B-Achse (A-Achse) und der Orientierungswinkel θ das gleiche Vorzeichen besitzen, d. h. es gilt:

- $\theta \geq 0 \triangleq B \geq 0$ (Zweig 1)
- $\theta \leq 0 \triangleq B \leq 0$ (Zweig -1)

Dadurch ist die Rückwärtstransformation eindeutig lösbar. Ein Übergang von Zweig 1 nach -1 und umgekehrt ist ohne weiteres beim Durchfahren der Orientierungsstelle $\theta = 0$ (Nordpolsingularität) möglich.

- Nordpol-Singularität:**

Vektor-Ori:

Der Nordpol stellt eine Singularität dar.

Im Nordpol wirkt der Maschinenparameter 103000160 Epsilon-Umgebungen, d. h. Orientierungsbewegungen durch die ε -Umgebung werden zwangsweise durch die Stelle $\theta = 0$ geleitet.

Bei $\theta = 0$ kann φ programmiert werden, jedoch erfolgt keine Bewegung.

Linear-Ori:

Der Nordpol stellt keine Singularität dar.

Der Nordpol kann ohne weiteres durchfahren werden. Der Maschinenparameter 103000160 Epsilon-Umgebungen hat keine Wirkung, d. h. Orientierungsbewegungen durch die ε -Umgebung werden nicht zwangsweise durch die Stelle $\theta = 0$ geleitet.

Bei $\theta = 0$ kann φ programmiert werden und die gewünschte Bewegung (Drehung der C-Achse) wird durchgeführt.

11.11.2 Handlungsanweisung: 5-Achs-Transformation RLLLR

Applizieren Die Achstransformation kann über die relevanten Maschinenparameter appliziert werden. Dazu muss die Funktion Achstransformation über den Parameter **tra** "Achstransformationen" im Setup (SUP) aktiviert werden. Der Pfad des MP-Namens ist immer "TRA/AxTrafo[1..20]/...".

Alternativ dazu kann die Achstransformation über ein benutzerdefiniertes Systemdatum vom Typ SysAxTrafo_t appliziert werden.

Die beiden Varianten sind inhaltlich identisch.

Die Achspositionen der Referenzstellung müssen ermittelt werden.

In der nachfolgenden Tabelle sind jeweils die äquivalenten MP-Nummern, MP-Name und SD-Elemente angegeben. Um eine Transformation zu applizieren, muss eine der Varianten entsprechend besetzt werden.

Transformations-relevante Parameter

MP-Nummer	Beschreibung	
1030 00110	Transformationstyp	
	MP-Name: TrafoType	
	SD-Elem. : Type	
	3232202	RLLLR mit Vektor-Orientierung
	3232102	RLLLR mit Linear-Orientierung
1030 00120	Systemachsen/-koordinaten der Transformation	
	MP-Name: FwdInCoordIndTrafo[1..8]	
	SD-Elem. : AxisAssignment[1..8]	
	[1]	Systemachsnummer von R_1 (C-/A-Achse)
	[2]	Systemachsnummer von X
	[3]	Systemachsnummer von Y
	[4]	Systemachsnummer von Z
	[5]	Systemachsnummer von R_2 (A-/B-Achse)
	[6..8]	nicht relevant

Achs-Transformation

MP-Nummer	Beschreibung																												
1030 00125	Achsklassifikationen der Transformationsachsen MP-Name: CoordClass[1..8] MP-Name: CoordDir[1..8] SD-Elem. : AxisClassification[1..8] Definiert die an der Transformation beteiligten Achsen inklusive Bewegungsrichtung. - MP-Nr. oder SD-Element: Wert ">0" bedeutet: Die Achse verfährt/dreht positiv im Sinne der obigen Abbildungen. Wert "<0" bedeutet: Die Achse verfährt/dreht negativ im Sinne der obigen Abbildungen. Mögliche Werte sind: +/-1 = X-Achse +/-2 = Y-Achse +/-3 = Z-Achse +/-100 = A-Achse +/-200 = B-Achse +/-300 = C-Achse - MP-Name: - Mögliche Werte für CoordClass sind - X,Y,Z (linear) - A,B,C (rotatorisch) - Mögliche Werte für CoordDir sind: - positiv (bzgl. Abb. "Referenzstellung") - negativ (bzgl. Abb. "Referenzstellung") Ist ein Parameter unbesetzt, gelten folgende Default-Einstellungen: <table><tr><th></th><th>MP-Nr / SD-Elem.:</th><th>CoordClass:</th><th>CoordDir:</th></tr><tr><td>[1]</td><td>300</td><td>C</td><td>positiv</td></tr><tr><td>[2]</td><td>1</td><td>X</td><td>positiv</td></tr><tr><td>[3]</td><td>2</td><td>Y</td><td>positiv</td></tr><tr><td>[4]</td><td>3</td><td>Z</td><td>positiv</td></tr><tr><td>[5]</td><td>200</td><td>B</td><td>positiv</td></tr><tr><td>[6..8]</td><td>nicht relevant</td><td>0</td><td>positiv</td></tr></table>		MP-Nr / SD-Elem.:	CoordClass:	CoordDir:	[1]	300	C	positiv	[2]	1	X	positiv	[3]	2	Y	positiv	[4]	3	Z	positiv	[5]	200	B	positiv	[6..8]	nicht relevant	0	positiv
	MP-Nr / SD-Elem.:	CoordClass:	CoordDir:																										
[1]	300	C	positiv																										
[2]	1	X	positiv																										
[3]	2	Y	positiv																										
[4]	3	Z	positiv																										
[5]	200	B	positiv																										
[6..8]	nicht relevant	0	positiv																										

MP-Nummer	Beschreibung
1030 00130	Achspositionen der Referenzstellung MP-Name: RefPosTrafo[1..8] SD-Elem. : AxZeroPos[1..8]
	[1] R_1^0
	[2] X^0
	[3] Y^0
	[4] Z^0
	[5] R_2^0
	[6..8] nicht relevant
1030 00140	Längen- und Winkelparameter MP-Name: JointParTrafo[1..16] SD-Elem. : LenParam[1..16] Definiert die Längenvektoren l_1, l_2 in [mm].
	[1] l_{1x}
	[2] l_{1y}
	[3] 0
	[4] l_{2x}
	[5] l_{2y}
	[6] l_{2z}
	[7..16] nicht relevant
1030 00160	Epsilon-Umgebungen MP-Name: EpsAxTrafo SD-Elem. : EpsilonRanges
	0.01

Tab. 11-50: Transformations-relevante Parameter

Aktivieren

Ist die 5-Achs-Transformation konfiguriert per

- Maschinenparameter, dann wird die Transformation im NC-Programm aktiviert mit (i=1,..20):

COORD(<i>)

- Systemdatum, dann wird die Transformation im NC-Programm aktiviert mit (Typ "SysAxTrafo_t"):

COORD(<SD.MtbAxTrafo>)**Programmbeispiel:**

Die 5-Achs-Transformation ist im 4. Maschinenparametersatz definiert. Am Programmanfang ist keine Achstransformation aktiv:

Transformation aktivieren

```

N100 G1 F1000 XA10 YA10 C30 ;Axes XA, YA and C are traversed
N110 COORD(4) ;Switching on the axis transformation
N130 x20 y30 phi0 ;the coordinates x,y,phi are traversed
N140 x10 z10 theta45 ;the coordinates x,z,theta are traversed
:
```

N999 COORD(0)

;Switching-off the axis transformation
;Axes are again programmed from here**Deaktivieren** COORD(0) bzw. COORD(0,2)

schaltet die Achstransformation im NC-Programm aus.

11.12 5-Achs-Transformation - Typ 3: RRLLL (mit Schwenk-/Rundtisch)

11.12.1 Beschreibung

Funktion

Name der Transformation	Transformationstyp	Wirkstelle	Koordinatensystem
5-Achs-Transformation mit Schwenk-/Rundtisch			
Vektororientierung	3232203	AT2	MCS, BCS
Lineare Orientierung	3232103	AT2	MCS, BCS

Tab. 11-51: Funktion

Koordinaten im Maschinenkoordinatensystem MCS: R1, R2, X, Y, Z

Koordinaten im Basiskoordinatensystem BCS: x, y, z, ϕ , ϑ

Die verteilte 5-Achs-Transformation (RRLLL) besitzt drei Linearachsen (X, Y, Z) und zwei rotatorische Achsen (R1, R2).

Sie kommt an der Wirkstelle 2 zum Einsatz. Sie wirkt also zwischen dem MCS = (R1, R2, X, Y, Z) und dem BCS = (x, y, z, ϕ , ϑ).

Die rotatorischen Achsen R1 und R2 können als Endlos- oder als Rundachsen betrieben werden.

Das Werkzeug wird durch die linearen Achsen X, Y, Z bewegt. Die Werkzeuglängsachse steht parallel zur Z-Achse. Der Werkstücktisch (Dreh-Schwenk-Tisch) mit dem darauf befindlichen Werkstück wird durch die rotatorischen Achsen R1 und R2 gegenüber dem Werkzeug orientiert.

Es gibt 3 Varianten:

- Variante CBLLL: Der Werkstücktisch mit dem damit verbundenen BCS wird von der Rotationsachse R1 = C-Achse gedreht. Der Werkstücktisch zusammen mit der C-Achse wird von der B-Achse (parallel zur MCS-Y-Achse) geschwenkt. Die C-Achse steht in Referenzstellung parallel zur MCS-Z-Achse, wird aber von der B-Achse aus dieser Orientierung herausgeschwenkt.
- Variante CALLL: Der Werkstücktisch mit dem damit verbundenen BCS wird von der Rotationsachse R1 = C-Achse gedreht. Der Werkstücktisch zusammen mit der C-Achse wird von der A-Achse (parallel zur MCS-X-Achse) geschwenkt. Die C-Achse steht in Referenzstellung parallel zur MCS-Z-Achse, wird aber von der A-Achse aus dieser Orientierung herausgeschwenkt.
- Variante BALLL: Der Werkstücktisch mit dem damit verbundenen BCS wird von der Rotationsachse R1 = B-Achse gedreht. Der Werkstücktisch zusammen mit der B-Achse wird von der A-Achse (parallel zur MCS-X-Achse) geschwenkt. Die B-Achse steht in Referenzstellung parallel zur MCS-Y-Achse, wird aber von der A-Achse aus dieser Orientierung herausgeschwenkt.

Das BCS ist mit dem Werkstücktisch fest verbunden. In Referenzstellung steht die x-Richtung des BCS parallel zur MCS-X-Achse, die y-Richtung des

BCS parallel zur MCS-Y-Achse und die z-Richtung des BCS parallel zur MCS-Z-Richtung.

Die Vektoren $\vec{l}_1$ und $\vec{l}_2$ beziehen sich auf die Referenzstellung. Sie definieren die Geometrie der 5-Achs-Kinematik für die verschiedenen Varianten wie folgt:

Variante CBLLL: $\vec{l}_1$ ist der Vektor entlang der MCS-X-Achse von der R1-Achse (C-Achse) zur R2-Achse (B-Achse) und hat daher nur die Komponente l_{1x} . $\vec{l}_2$ ist der Fortsetzungsvektor von der R2-Achse (B-Achse) zum Nullpunkt des BCS und hat die drei Komponenten l_{2x} , l_{2y} und l_{2z} .

Variante CALLL: $\vec{l}_1$ ist der Vektor entlang der MCS-Y-Achse von der R1-Achse (C-Achse) zur R2-Achse (A-Achse) und hat daher nur die Komponente l_{1y} . $\vec{l}_2$ ist der Fortsetzungsvektor von der R2-Achse (A-Achse) zum Nullpunkt des BCS und hat die drei Komponenten l_{2x} , l_{2y} und l_{2z} .

Variante BALLL: $\vec{l}_1$ ist der Vektor entlang der MCS-Z-Achse von der R1-Achse (B-Achse) zur R2-Achse (A-Achse) und hat daher nur die Komponente l_{1z} . $\vec{l}_2$ ist der Fortsetzungsvektor von der R2-Achse (A-Achse) zum Nullpunkt des BCS und hat die drei Komponenten l_{2x} , l_{2y} und l_{2z} .

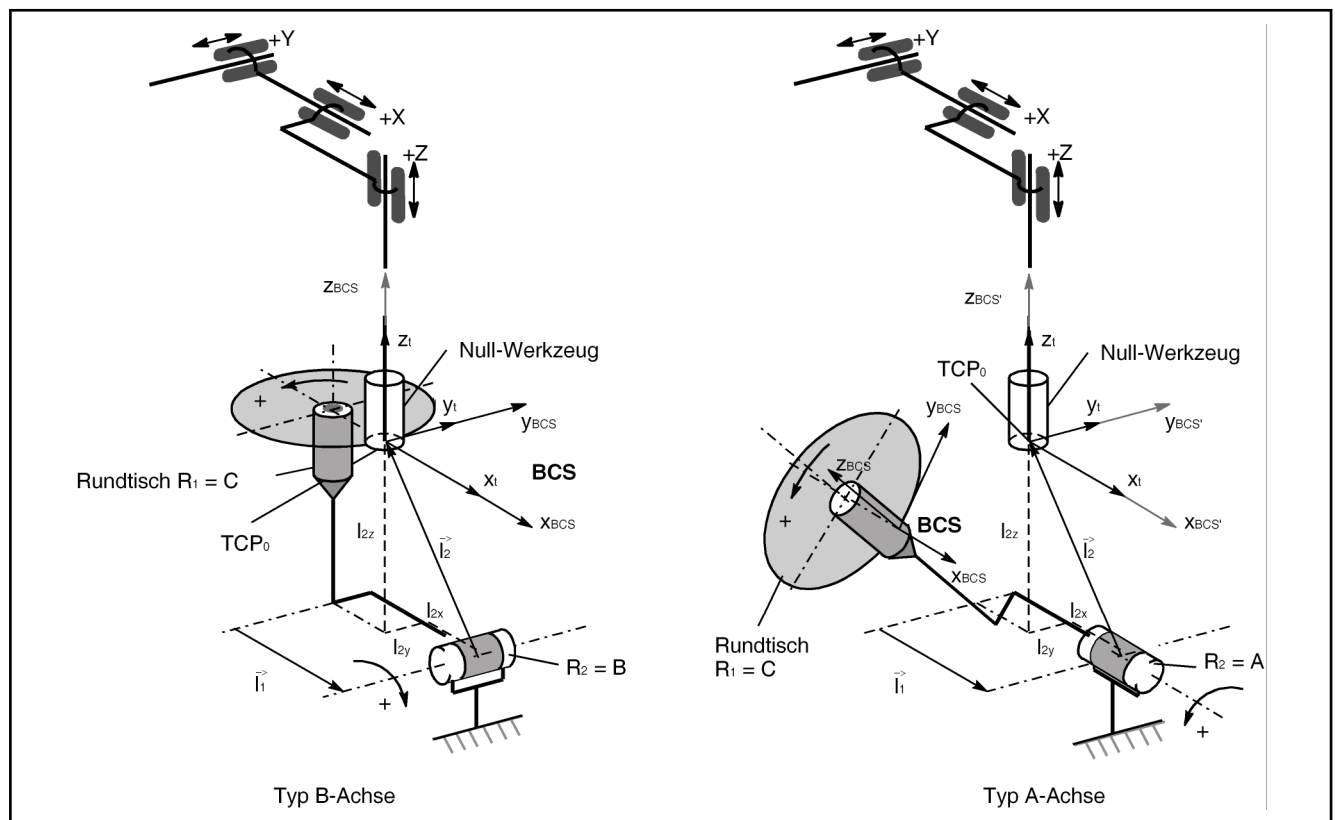


Abb. 11-43: Varianten CBLLL und CALLL der 5-Achs-Transformation RRLLL

Eigenschaften der Achs-Transformation

Beim Einschalten der 5-Achstransformation werden die aktuellen Koordinaten R1, R2, X, Y, Z des MCS in die Koordinaten x, y, z, φ , θ des BCS transformiert (Vorwärtstransformation).

Bei aktiver 5-Achs-Transformation werden die programmierten Koordinaten x, y, z, φ , θ des BCS in die Koordinaten R1, R2, X, Y, Z des MCS transformiert (Rückwärtstransformation).

Die Koordinaten des BCS setzen sich zusammen aus den 3 Linearkoordinaten x, y, z und den 2 Orientierungskordinaten φ , θ .

Mit Einschalten dieser Achs-Transformation wird entweder die Vektororientierungsbewegung (Typ 3232203) oder die lineare Orientierungsbewegung (Typ 3232103) aktiviert. Letztere ist jedoch nur für die Varianten CBLL und CALLL zulässig.

Die Orientierungskordinaten können auch als Spline programmiert werden (siehe [Kap. 9.1.8 "Spline programmieren " auf Seite 148](#)).

In Abhängigkeit vom Orientierungstyp ergeben sich unterschiedliche Eigenschaften

Vektor-Orientierung:

Bei der Vektororientierungsbewegung erfolgt jede Orientierungsänderung durch Programmierung von φ , θ oder einer entsprechenden Alternativsyntax (s. Spez. Orientierungsbewegung) durch eine Drehung des Orientierungsvektors $\vec{\varphi}$ (φ, θ) um eine raumfeste Drehachse (Großkreis-Bewegung der Vektorspitze). Die rotatorischen Achsen können sowohl Endlos- als auch Rundachsen sein.

tors $\vec{\varphi}$ (φ, θ) um eine raumfeste Drehachse (Großkreis-Bewegung der Vektorspitze). Die rotatorischen Achsen können sowohl Endlos- als auch Rundachsen sein.

Linear-Orientierung:

Die Winkel φ und θ werden linear vom Anfangs- zum Endpunkt interpoliert. Das Verhalten der "kürzeste Weg"-Logik hängt von der Programmierungsart der Orientierungsbewegung ab. Bei φ -, θ -Programmierung ist die "kürzeste Weg"-Logik ausgeschaltet. Bei der O(...)-Programmierung ist die "kürzeste Weg"-Logik eingeschaltet. φ , θ können Werte aus den folgenden Bereichen annehmen:

$\varphi \in [-\text{unendl}, +\text{unendl}]$, $\theta \in [-180, +180]$.

Die rotatorischen Achsen müssen als Rundachsen konfiguriert sein (keine Modulo-Endlosachse).

Referenzstellung

In der Referenzstellung ist die Spitze (TCP) des rotationssymmetrischen Werkzeugs (Fräser oder Laser) auf dem Ursprung des BCS entlang der Richtung der z-Koordinate des BCS ausgerichtet. In dieser Stellung müssen die BCS-Koordinaten $(x, y, z, \varphi, \theta) = (0, 0, 0, 0, 0)$ sein. Die zugehörigen MCS-Koordinaten (R1, R2, X, Y, Z), für die das erreicht wird, werden mit $(R1^0, R2^0, X^0, Y^0, Z^0)$ bezeichnet, und sind als Achspositionen der Referenzstellung einzutragen in MP 1030 00130.

Achs-Transformation

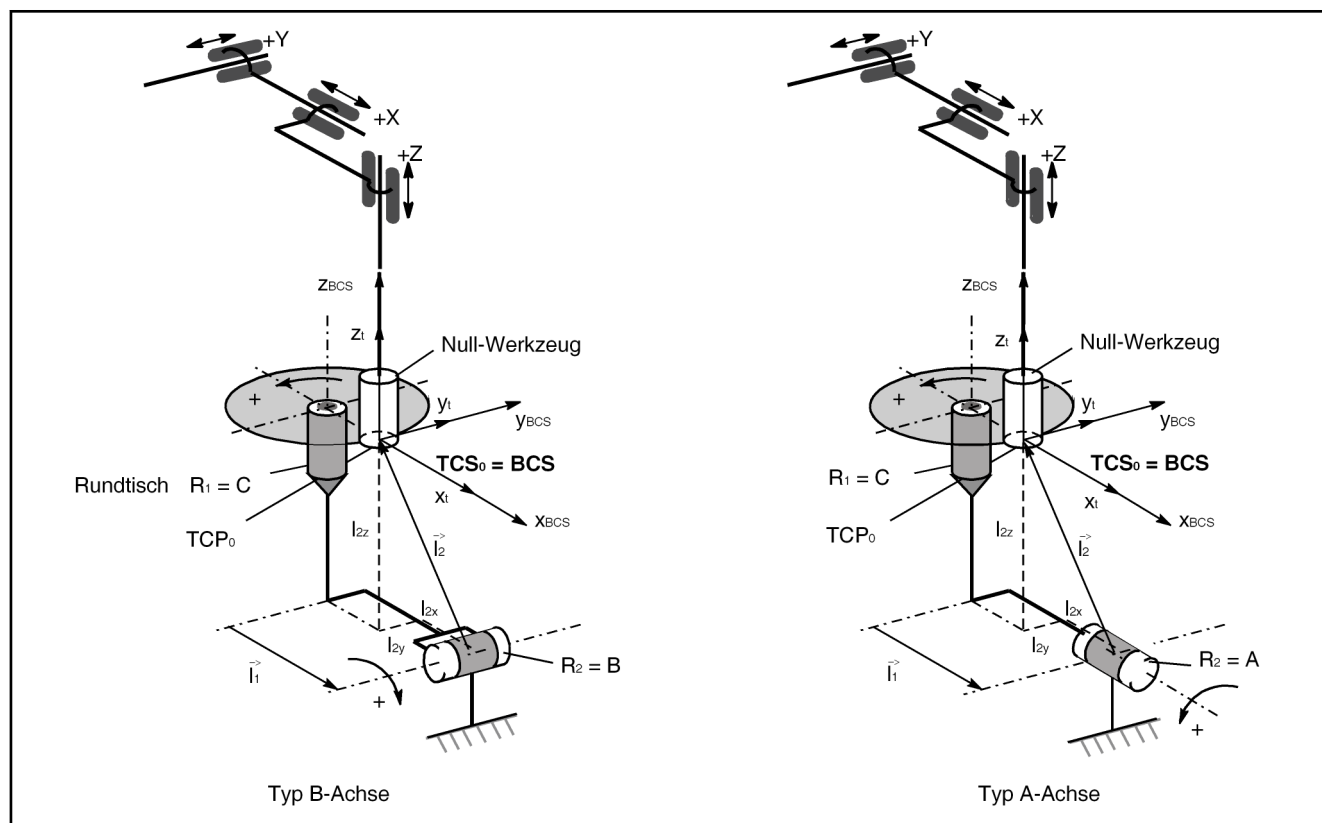


Abb. 11-45: Referenzstellung

TCS, TCP

TCS bezeichnet das Werkzeugkoordinatensystem.

TCP₀/TCP₁ bezeichnet den Tool Center Point (Werkzeugmittelpunkt) **ohne/mit** Korrektur.

Werkzeugkorrektur

Die Achs-Transformation unterstützt eine Werkzeugkorrektur im Werkzeugkoordinatensystem (TCS). In der Referenzstellung stimmen TCS₀ und BCS

überein. Der Werkzeugkorrekturvektor $\vec{l}_t$ zeigt vom korrigierten Werkzeugmittelpunkt TCP₁ zum unkorrigierten TCP₀. Er setzt sich aus den Längenkorrekturen L1, L2 und L3 aus der D-Korrekturtabelle zusammen.

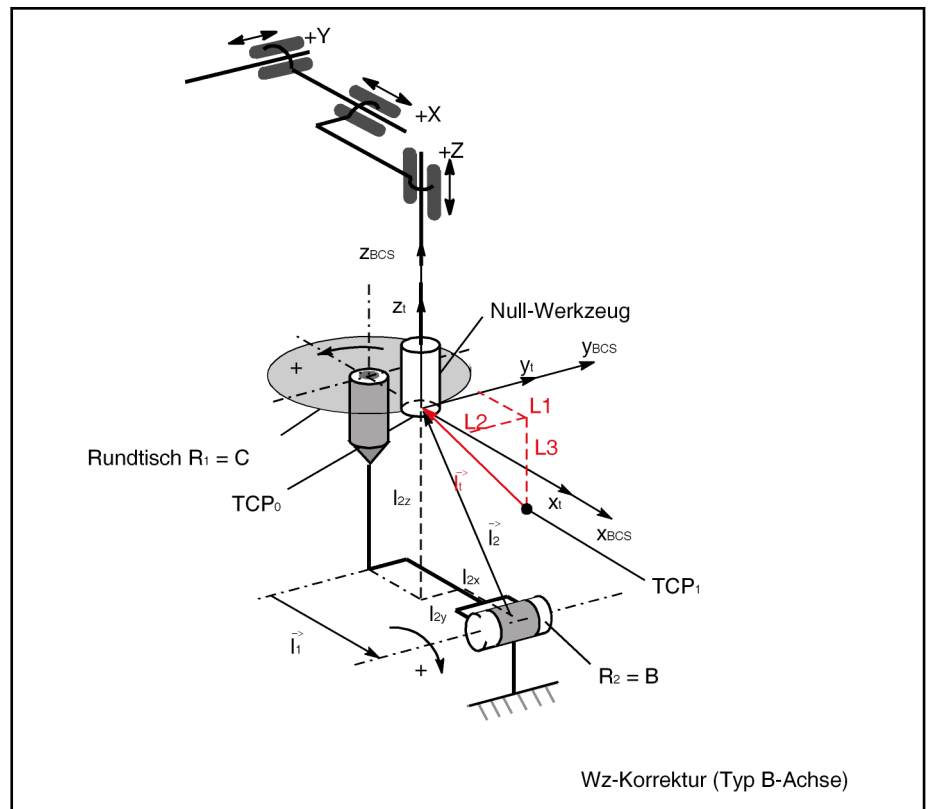


Abb. 11-46: Werkzeugkorrektur

Besonderheiten

- **Mehrdeutige Achspositionen:**

Bei den Varianten CBLLL und CALLL kann eine TCP-Position und eine Werkzeugorientierung im allgemeinen durch zwei verschiedene Achsposen (Transformationszweige) erreicht werden. Die Zweige unterscheiden sich in den B-Achspositionen (bzw. A-Achsposition). Für die Variante BALLL gibt es keine Mehrdeutigkeit.

Vektor-Orientierung:

Welche der beiden Posen (Transformationszweige) benutzt wird, entscheidet sich beim Einschalten der Achs-Transformation. Ein Übergang von Zweig 1 nach -1 und umgekehrt ist danach nur beim Durchfahren der Orientierungsstelle $\theta = 0$ (Nordpolsingularität) möglich.

Lineare Orientierung:

Der Transformationszweig wird immer so gewählt, dass die B-Achse (A-Achse) und der Orientierungswinkel θ das gleiche Vorzeichen besitzen, d. h. es gilt:

- $\theta \geq 0 \triangleq B \geq 0$ (Zweig 1)
- $\theta \leq 0 \triangleq B \leq 0$ (Zweig -1)

Dadurch ist die Rückwärtstransformation eindeutig lösbar. Ein Übergang von Zweig 1 nach -1 und umgekehrt ist ohne weiteres beim Durchfahren der Orientierungsstelle $\theta = 0$ (Nordpolsingularität) möglich.

- **Nordpol-Singularität:**

Vektor-Orientierung bei den Varianten CBLLL und CALLL:

Der Nordpol stellt eine Singularität dar.

Im Nordpol wirkt der Maschinenparameter 103000160 Epsilon-Umgebungen, d. h. Orientierungsbewegungen durch die ε -Umgebung werden zwangsweise durch die Stelle $\theta = 0$ geleitet.

Bei $\theta = 0$ kann φ programmiert werden, es erfolgt jedoch keine Bewegung.

Lineare Orientierung bei den Varianten CBLL und CALL:

Der Nordpol stellt keine Singularität dar.

Der Nordpol kann ohne weiteres durchfahren werden. Der Maschinenparameter 103000160 Epsilon-Umgebungen hat keine Wirkung, d. h. Orientierungsbewegungen durch die ε -Umgebung werden nicht zwangsweise durch die Stelle $\theta = 0$ geleitet.

Bei $\theta = 0$ kann φ programmiert werden und die gewünschte Bewegung (Drehung der C-Achse) wird durchgeführt.

Vektor-Orientierung bei der Variante BALL:

Der Nordpol stellt keine Singularität dar.

Der Nordpol kann ohne weiteres durchfahren werden.

Die Orientierung $\varphi = 90^\circ$, $\theta = 90^\circ$ darf jedoch nicht programmiert werden, und auch nicht auf anderen Wegen erreichbar sein.

11.12.2 Handlungsanweisung: 5-Achs-Transformation RRL

Applizieren Die Achstransformation kann über die relevanten Maschinenparameter appliziert werden. Dazu muss die Funktion Achstransformation über den Parameter **tra** "Achstransformationen" im Setup (SUP) aktiviert werden. Der Pfad des MP-Namens ist immer "TRA/AxTrafo[1..20]/...".

Alternativ dazu kann die Achstransformation über ein benutzerdefiniertes Systemdatum vom Typ SysAxTrafo_t appliziert werden.

Die beiden Varianten sind inhaltlich identisch.

Die Achspositionen der Referenzstellung müssen ermittelt werden.

In der nachfolgenden Tabelle sind jeweils die äquivalenten MP-Nummern, MP-Name und SD-Elemente angegeben. Um eine Transformation zu applizieren, muss eine der Varianten entsprechend besetzt werden.

Transformations-relevante Parameter

MP-Nummer	Beschreibung	
1030 00110	Transformationstyp	
	MP-Name: TrafoType	
	SD-Element: Type	
	3232203	RRL mit Vektor-Orientierung
	3232103	RRL mit Linear-Orientierung (Für BALL unzulässig)

MP-Nummer	Beschreibung
1030 00120	Systemachsen/-koordinaten der Transformation MP-Name: FwdInCoordIndTrafo[1..8] SD-Elem. : AxisAssignment[1..8]
	[1] Systemachsnummer von R_1 (C-Achse für CBLL, C-Achse für CALL, B-Achse für BALL)
	[2] Systemachsnummer von R_2 (B-Achse für CBLL, A-Achse für CALL, A-Achse für BALL)
	[3] Systemachsnummer von X
	[4] Systemachsnummer von Y
	[5] Systemachsnummer von Z
	[6..8] nicht relevant

Achs-Transformation

MP-Nummer	Beschreibung		
1030 00125	Achsklassifikationen der Transformationsachsen		
	MP-Name: CoordClass[1..8]		
	MP-Name: CoordDir[1..8]		
	SD-Elem. : AxisClassification[1..8]		
	Definiert die an der Transformation beteiligten Achsen inklusive Bewegungsrichtung.		
	- MP-Nr. oder SD-Element: Wert ">0" bedeutet: Die Achse verfährt/dreht positiv im Sinne der obigen Abbildungen. Wert "<0" bedeutet: Die Achse verfährt/dreht negativ im Sinne der obigen Abbildungen. Mögliche Werte sind: +/-1 = X-Achse +/-2 = Y-Achse +/-3 = Z-Achse +/-100 = A-Achse +/-200 = B-Achse +/-300 = C-Achse - MP-Name: - Mögliche Werte für CoordClass sind - X,Y,Z (linear) - A,B,C (rotatorisch) - Mögliche Werte für CoordDir sind: - positiv (bzgl. Abb. "Referenzstellung") - negativ (bzgl. Abb. "Referenzstellung")		
	Ist ein Parameter unbesetzt, gelten folgende Default-Einstellungen:		
		MP-Nr / SD-Elem.:	CoordClass: CoordDir:
	[1]	300	C positiv
	[2]	200	B positiv
	[3]	1	X positiv
	[4]	2	Y positiv
	[5]	3	Z positiv
	[6..8]	nicht relevant	0 positiv

MP-Nummer	Beschreibung
1030 00130	Achspositionen der Referenzstellung MP-Name: RefPosTrafo[1..8] SD-Elem. : AxZeroPos[1..8]
	[1] R_1^0
	[2] R_2^0
	[3] X^0
	[4] Y^0
	[5] Z^0
	[6..8] nicht relevant
1030 00140	Längen- und Winkelparameter MP-Name: JointParTrafo[1..16] SD-Elem. : LenParam[1..16] Definiert die Längenvektoren l_1, l_2 in [mm].
	[1] l_{1x} (Nur relevant für CBLLL)
	[2] l_{1y} (Nur relevant für CALLL)
	[3] l_{1z} (Nur relevant für BALLL)
	[4] l_{2x}
	[5] l_{2y}
	[6] l_{2z}
	[7..16] nicht relevant
1030 00160	Epsilon-Umgebungen MP-Name: EpsAxTrafo SD-Elem. : EpsilonRanges
	0.01

Tab. 11-55: Transformations-relevante Parameter

Aktivieren

Ist die 5-Achs-Transformation konfiguriert per

- Maschinenparameter, dann wird die Transformation im NC-Programm aktiviert mit (i=1,..20):

COORD(<i>)

- Systemdatum, dann wird die Transformation im NC-Programm aktiviert mit (Typ "SysAxTrafo_t"):

COORD(<SD.MtbAxTrafo>)**Programmbeispiel:**

Die 5-Achs-Transformation ist im 4. Maschinenparametersatz definiert. Am Programmanfang ist keine Achstransformation aktiv:

Transformation aktivieren

```

N100 G1 F1000 XA10 YA10 C30 ;Axes XA, YA and C are traversed
N110 COORD(4) ;Switching on the axis transformation
N130 x20 y30 phi0 ;the coordinates x,y,phi are traversed
N140 x10 z10 theta45 ;the coordinates x,z,theta are traversed
:
```

N999 COORD(0)

;Switching-off the axis transformation
;Axes are again programmed from here**Deaktivieren** COORD(0) bzw. COORD(0,2)

schaltet die Achstransformation im NC-Programm aus.

11.13 5-Achs-Transformation - Typ 4: Kardanischer Kopf

11.13.1 Beschreibung

Funktion

Name der Transformation	Transformationstyp	Wirkstelle	Koordinatensystem
Kardanische 5-Achs-Transformation	3232211	AT2	MCS, BCS

Tab. 11-56: Funktion

Ausgangsgrößen im MCS: X,Y,Z,C,B

Eingangsgrößen im BCS: x, y, z, φ , ϑ

Die Achs-Kinematik der kardanischen Achs-Transformation umfasst drei Linearachsen (z. B. X, Y und Z), die ein rechtshändiges kartesisches Koordinatensystem bilden, und einen kardanischen Bearbeitungskopf, der das Werkzeug mit zwei rotatorischen Achsen (B und C) im Raum orientiert.

Die C-Achse ist parallel zur Z-Achse des MCS angeordnet. Die B-Achse ist gegenüber der xy-Ebene des MCS geneigt. Der Neigungswinkel variiert im Bereich $0^\circ \leq \alpha \leq 89^\circ$. In den meisten praktischen Fällen ist $\alpha = 45^\circ$. Für den Fall $\alpha = 0^\circ$ ist die kardanische Achs-Transformation mit der 5-Achs-Transformation vom Typ 3232201 identisch.

$\vec{i}_1$ ist ein Vektor entlang x_{BCS} von der C-Achse zur B-Achse, d. h., $\vec{i}_1$ hat nur eine Komponente l_{1x} .

$\vec{i}_2$ ist die Fortsetzung von der B-Achse zum Nullpunkt des BCS und setzt sich aus bis zu drei Komponenten l_{2x} , l_{2y} und l_{2z} zusammen.

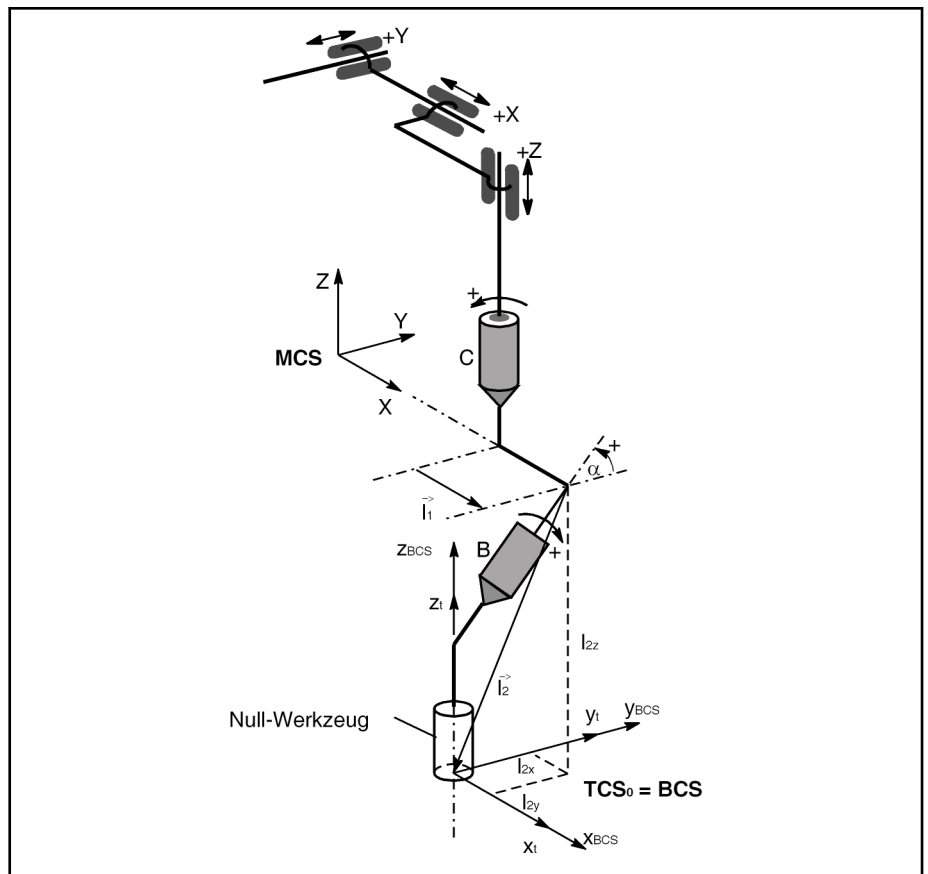


Abb. 11-47: Funktion Typ 4: Kardanischer Kopf

Restriktion

Eine Online-Korrektur für φ und ϑ ist nicht möglich.

Relevante NC-Funktionen

COORD(<i>)	Mit $i=1..20$ wird eine 5-Achs-Transformationen aktiviert, die in MP 1030 00110 definiert sind.
COORD(0,2)	Die aktive Transformation wird deaktiviert.

Tab. 11-57: Relevante NC-Funktionen

Relevante CPL-Funktionen

SDR(300)	Wirksame Epsilon-Umgebung. Der wirksame Wert kann größer als der in EpsAxTrafo (MP 1030 00160) eingestellte Wert sein.

Tab. 11-58: Relevante CPL-Funktionen

Relevante Maschinenparameter

1030 00110	definiert den Transformationstyp
1030 00120	Definiert die, an der Transformation beteiligten Systemachsen/-koordinaten der Transformation.
1030 00130	definiert die Achspositionen der Referenzstellung

1030 00140	definiert die Längen- und Winkelparameter
1030 00160	Epsilon-Umgebungen

Tab. 11-59: Relevante Maschinenparameter

Eigenschaften der Kardanischen Achs-Transformation

Eingangsgrößen sind die Linearkoordinaten x , y , z und die Orientierungskoordinaten φ und ϑ . Mit Einschalten dieser Achs-Transformation wird automatisch die Vektororientierungsbewegung aktiviert, d. h. jede Orientierungsänderung durch Programmierung von φ und ϑ oder einer entsprechenden Alternativsyntax (siehe DIN-Programmieranleitung) erfolgt durch eine Drehung

des Orientierungsvektors $\vec{\rho}$ (φ , ϑ) um eine raumfeste Drehachse (Großkreis-Bewegung der Vektorspitze). Der Orientierungsvektor $\vec{\rho}$ kann darüber hinaus als Spline programmiert werden.

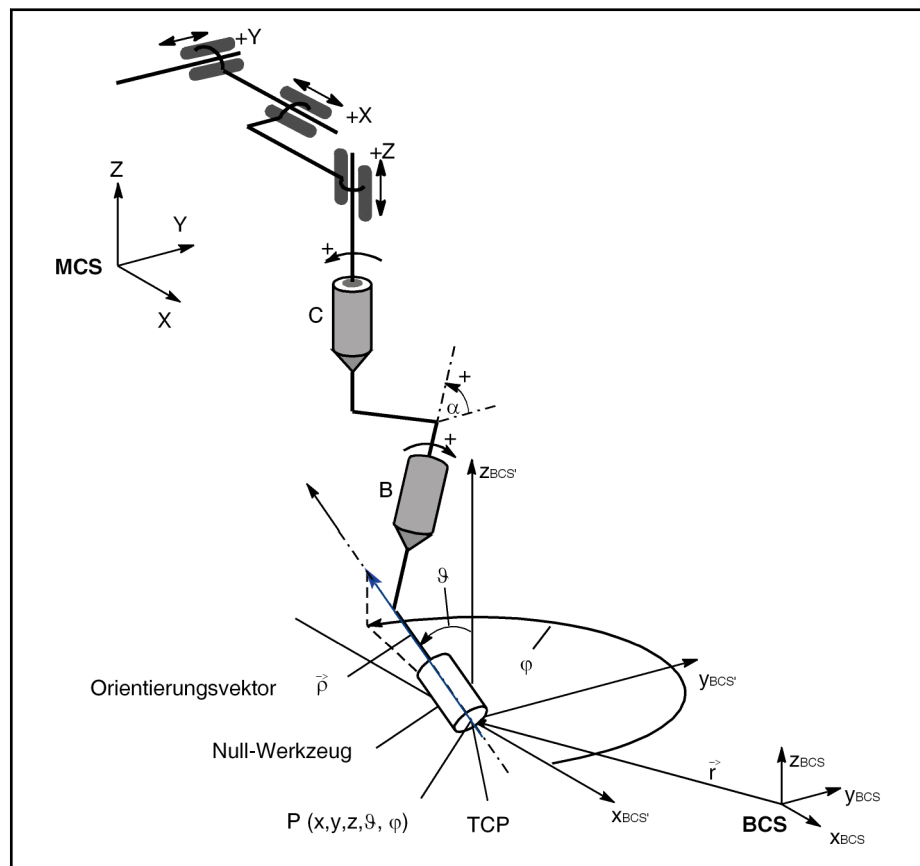


Abb. 11-48: Kardanische Achs-Transformation

Bedingt durch die geneigte B-Achse ist der Winkel ϑ zwischen z_{BCS} und der Werkzeugsymmetrieachse nicht größer als

$$\vartheta_{\max} = 2(90^\circ - \alpha)$$

Infolge dessen ist die z-Komponente des Orientierungsvektors $\vec{\rho}_z$ nicht kleiner als

$$\vec{\rho}_z^{\min} = \cos \vartheta_{\max}$$

Für $\vartheta_{\max}$ bzw. $\vartheta_{\min}$ erhält man dann ausgehend von der Referenzstellung eine Orientierung, die B um $\pm 180^\circ$ dreht (Referenzstellung vorausgesetzt).

Beispiel:

$\alpha = 45^\circ$: $\vartheta_{\max} \rightarrow 90^\circ$; also ist $\vartheta_{\min} = 0$ in der B-Achsstellung $\pm 180^\circ$

Referenzstellung

Die Achspositionen aller Achsen sind in der Referenzstellung durch folgende Eigenschaften definiert:

- Der Werkzeugnullpunkt (Werkzeugaufnahme oder TCP eines Nullwerkzeugs) liegt im Ursprung des Basiskoordinatensystems (BCS)
- Die Werkzeugsymmetrieachse liegt entlang z
- Die B-Achse liegt in der yz-Ebene
- Neigungswinkel α der B-Achse $\vec{l}_1$ ist der Vektor von der C-Achse zur B-Achse.

$\vec{l}_2$ ist der Fortsetzungsvektor zum BCS.

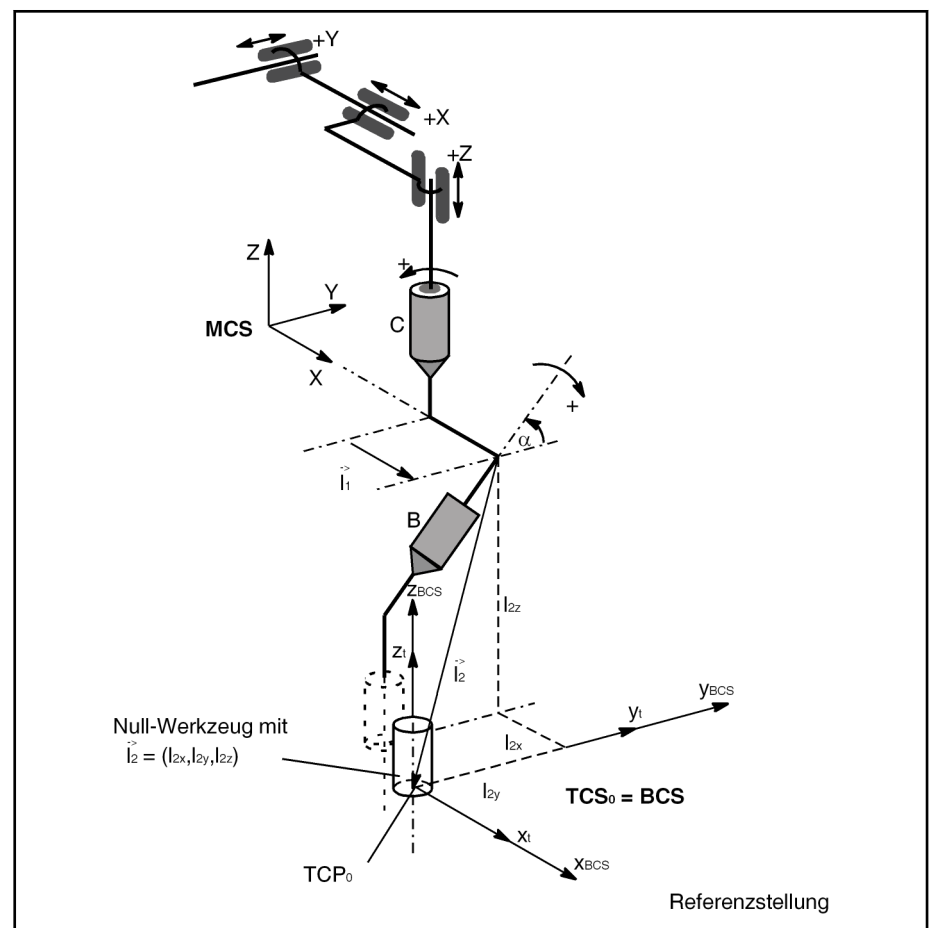


Abb. 11-49: Referenzstellung

TCS, TCP

TCS bezeichnet das Werkzeugkoordinatensystem.

TCP₀/TCP₁ bezeichnet den Tool Center Point (Werkzeugmittelpunkt) **ohne/mit** Korrektur.

Werkzeugkorrektur

Die Achs-Transformation unterstützt eine Werkzeugkorrektur im Werkzeugkoordinatensystem (TCS). Das TCS stimmt in der Referenzstellung mit dem

BCS überein. Der Werkzeugkorrekturvektor $\vec{l}_t$ zeigt vom korrigierten Werkzeugmittelpunkt TCP₁ zum unkorrigierten TCP₀. Er setzt sich aus den Längskorrekturen L1, L2 und L3 aus der D-Korrekturtabelle zusammen.

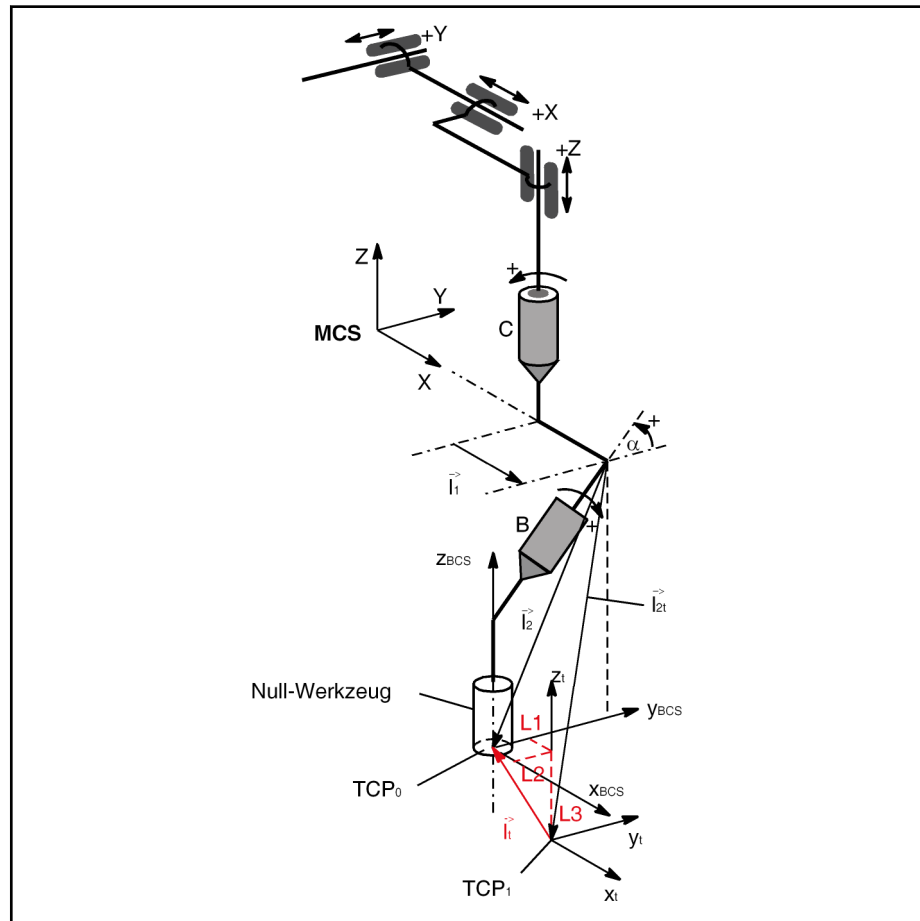


Abb. 11-50: Werkzeugkorrektur

Besonderheiten

Bei der kardanischen Achs-Transformation werden Singularitäten der Orientierungsbewegung berücksichtigt, d. h., nicht tangentialer Durchgang von $\vec{\rho}$ durch die Stellen $\vec{\rho} = (0,0,1)$ (=Nordpolsingularität) und $\vec{\rho} = (*,*, \vec{\rho}_{\text{min}})$ (=Äquatorsingularität).

- Mehrdeutige Achspositionen:**

Im Allgemeinen kann eine TCP-Position $\vec{r}$ und eine Werkzeugorientierung $\vec{\rho}$ durch zwei verschiedene Achsposen (Transformationszweige) erreicht werden. Die Zweige unterscheiden sich in den B-Achspositionen.

Achs-Transformation

Bei Orientierungsbewegungen tangential durch den Breitenkreis $\vartheta = \vartheta_{\max}$ wird der Transformationszweig nur dann gewechselt, wenn die B-Achse einen Verfahrbereich von mindestens $B\epsilon [-360^\circ, +360^\circ]$ hat. Der Verfahrbereich ergibt sich aus dem negativen und positiven ersten Endschalter der B-Achse.

Ausblenden des Software-Endschalters oder Umschalten auf die zweiten Endschalter haben hierauf keine Auswirkung.

Bei Bewegungen in Richtung $\vartheta = \vartheta_{\max}$ ergeben sich hohe B-Achsgeschwindigkeiten. NC-intern wird das Bahngeschwindigkeitsprofil in diesen Fällen so verändert, dass sich eine flüssige B-Achsbewegung ergibt.

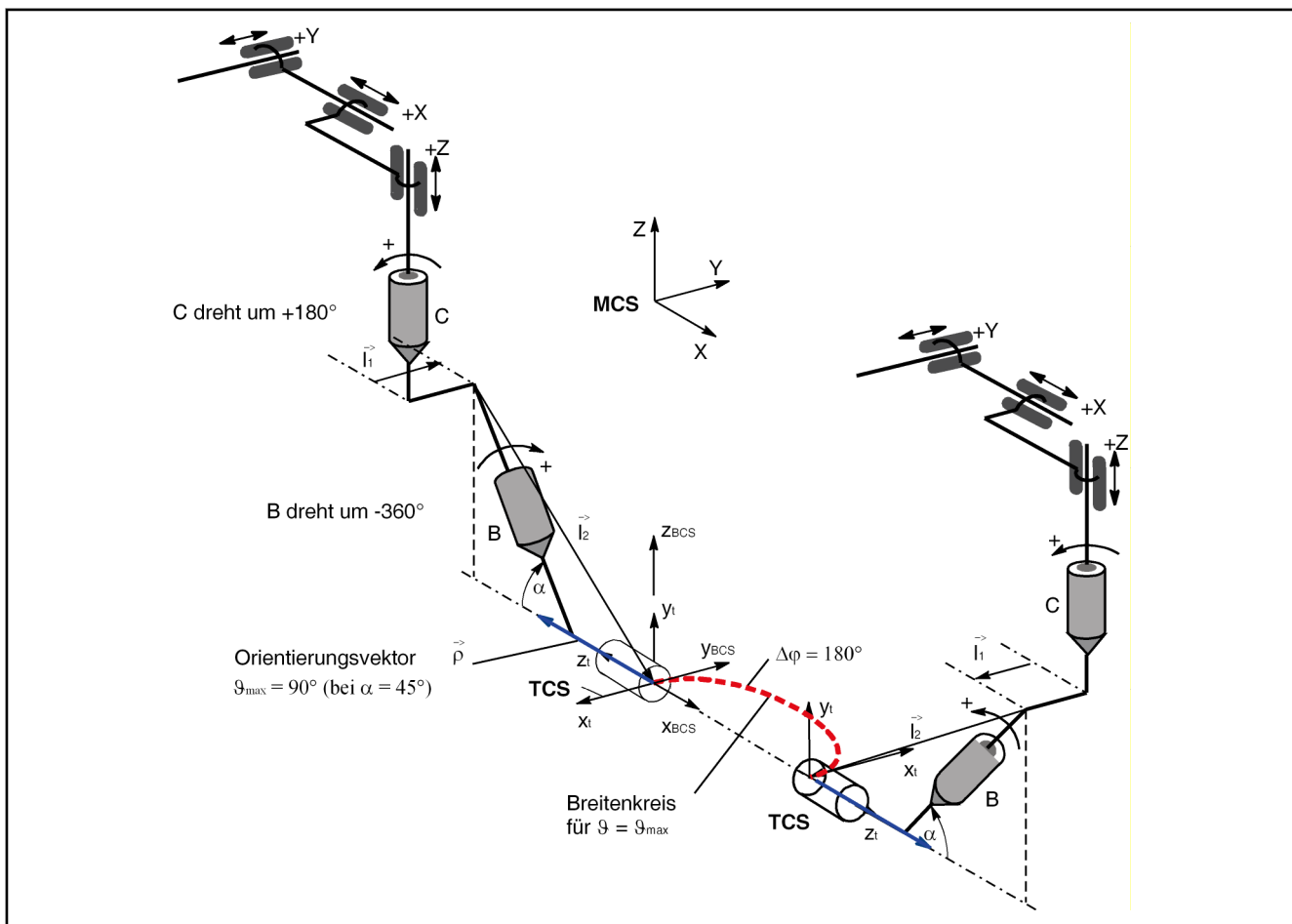


Abb. 11-52: Äquatorsingularität

- Vorschub:**

Nach dem Einschalten der 5-Achs-Transformation wird auf Raumkoordinatenprogrammierung geschaltet. Der programmierte Vorschub (F) bezieht sich nur noch auf die programmierbaren Positionskoordinaten, d. h. mit dem F-Wort wird die Bahngeschwindigkeit des Werkzeugmittelpunktes (TCP) programmiert.

Zusätzliche Orientierungs- und Pseudokoordinatenbewegungen verändern diese Bahngeschwindigkeit nicht. Orientierungs- und Pseudokoordinatenbewegung werden synchron mitgeführt, d. h., die Endposition wird für alle Koordinaten zeitgleich erreicht. Die mitgeführte Bewegung der Orientierungs- und Pseudokoordinaten kann jedoch zu einer zusätz-

lichen Limitierung der Bahnkinematik (maximale Bahngeschwindigkeit und -Beschleunigung) führen, da die Grenzwerte für alle, an der Bewegung beteiligten Achsen überwacht werden.

11.13.2 Handlungsanweisung: 5-Achs-Transformation Kardanische LLLRR

Applizieren Die Achstransformation kann über die relevanten Maschinenparameter appliziert werden. Dazu muss die Funktion Achstransformation über den Parameter **tra** "Achstransformationen" im Setup (SUP) aktiviert werden. Der Pfad des MP-Namens ist immer "TRA/AxTrafo[1..20]/...".

Alternativ dazu kann die Achstransformation über ein benutzerdefiniertes Systemdatum vom Typ SysAxTrafo_t appliziert werden.

Die beiden Varianten sind inhaltlich identisch.

Die Achspositionen der Referenzstellung müssen ermittelt werden.

In der nachfolgenden Tabelle sind jeweils die äquivalenten MP-Nummern, MP-Name und SD-Elemente angegeben. Um eine Transformation zu applizieren, muss eine der Varianten entsprechend besetzt werden.

Transformations-relevante Parameter

MP-Nummer	Beschreibung	
1030 00110	Transformationstyp MP-Name: TrafoType SD-Elem. : Type	
	3232211	kardanische LLLRR mit Vektor-Orientierung
	3232111	kardanische LLLRR mit Linear-Orientierung
1030 00120	Systemachsen/-koordinaten der Transformation MP-Name: FwdInCoordIndTrafo[1..8] SD-Elem. : AxisAssignment[1..8]	
	[1]	Systemachsnummer von X
	[2]	Systemachsnummer von Y
	[3]	Systemachsnummer von Z
	[4]	Systemachsnummer von C
	[5]	Systemachsnummer von B
	[6..8]	nicht relevant

MP-Nummer	Beschreibung		
1030 00125	Achsklassifikationen der Transformationsachsen		
	MP-Name: CoordClass[1..8]		
	MP-Name: CoordDir[1..8]		
	SD-Elem. : AxisClassification[1..8]		
	Definiert die an der Transformation beteiligten Achsen inklusive Bewegungsrichtung.		
	- MP-Nr. oder SD-Element: Wert ">0" bedeutet: Die Achse verfährt/dreht positiv im Sinne der obigen Abbildungen. Wert "<0" bedeutet: Die Achse verfährt/dreht negativ im Sinne der obigen Abbildungen. Mögliche Werte sind: +/-1 = X-Achse +/-2 = Y-Achse +/-3 = Z-Achse +/-100 = A-Achse +/-200 = B-Achse +/-300 = C-Achse - MP-Name: - Mögliche Werte für CoordClass sind - X,Y,Z (linear) - A,B,C (rotatorisch) - Mögliche Werte für CoordDir sind: - positiv (bzgl. Abb. "Referenzstellung") - negativ (bzgl. Abb. "Referenzstellung")		
	Ist ein Parameter unbesetzt, gelten folgende Default-Einstellungen:		
		MP-Nr / SD-Elem.:	CoordClass: CoordDir:
	[1]	1	X positiv
	[2]	2	Y positiv
	[3]	3	Z positiv
	[4]	300	C positiv
	[5]	200	B positiv
	[6..8]	nicht relevant	0 positiv

MP-Nummer	Beschreibung
1030 00130	Achspositionen der Referenzstellung MP-Name: RefPosTrafo[1..8] SD-Elem. : AxZeroPos[1..8]
	[1] X^0
	[2] Y^0
	[3] Z^0
	[4] C^0
	[5] B^0
	[6..8] nicht relevant
1030 00140	Längen- und Winkelparameter MP-Name: JointParTrafo[1..16] SD-Elem. : LenParam[1..16] Definiert die Längenvektoren l_1, l_2 in [mm].
	[1] l_{1x}
	[2] nicht relevant
	[3] nicht relevant
	[4] l_{2x}
	[5] l_{2y}
	[6] l_{2z}
	[7] α
	[8..16] nicht relevant
1030 00160	Epsilon-Umgebungen MP-Name: EpsAxTrafo SD-Elem. : EpsilonRanges
	0.01

Tab. 11-60: Transformations-relevante Parameter

Aktivieren

Ist die 5-Achs-Transformation konfiguriert per

- Maschinenparameter, dann wird die Transformation im NC-Programm aktiviert mit (i=1,..20):
COORD(<i>)
- Systemdatum, dann wird die Transformation im NC-Programm aktiviert mit (Typ "SysAxTrafo_t"):
COORD(<SD.MtbAxTrafo>)

Programmbeispiel:

Die 5-Achs-Transformation ist im 4. Maschinenparametersatz definiert. Am Programmanfang ist keine Achstransformation aktiv:

Transformation aktivieren

```

N100 G1 F1000 XA10 YA10 C30 ;Axes XA, YA and C are traversed
N110 COORD(4) ;Switching on the axis transformation
N130 x20 y30 phi0 ;the coordinates x,y,phi are traversed
N140 x10 z10 theta45 ;the coordinates x,z,theta are traversed

```

```

:
N999  COORD(0)                                ;Switching-off the axis transformation
                                           ;Axes are again programmed from here

```

Deaktivieren COORD(0) bzw. COORD(0,2)
 schaltet die Achstransformation im NC-Programm aus.

11.14 5-Achs-Transformation - Typ 4.1: Kardanischer Kopf mit linearer Orientierungsbewegung

11.14.1 Beschreibung

Funktion

Name der Transformation	Transformationstyp	Wirkstelle	Koordinatensystem
Kardanische 5-Achs-Transformation mit linearer Orientierungsbewegung	3232111	2	MCS, BCS

Tab. 11-61: Funktion

Die Achskonfiguration der kardanischen Achs-Transformation mit linearer Orientierungsbewegung und die der kardanischen Achs-Transformation mit Vektororientierungsbewegung stimmen mechanisch überein. Beide haben den in der Abbildung dargestellten Aufbau.

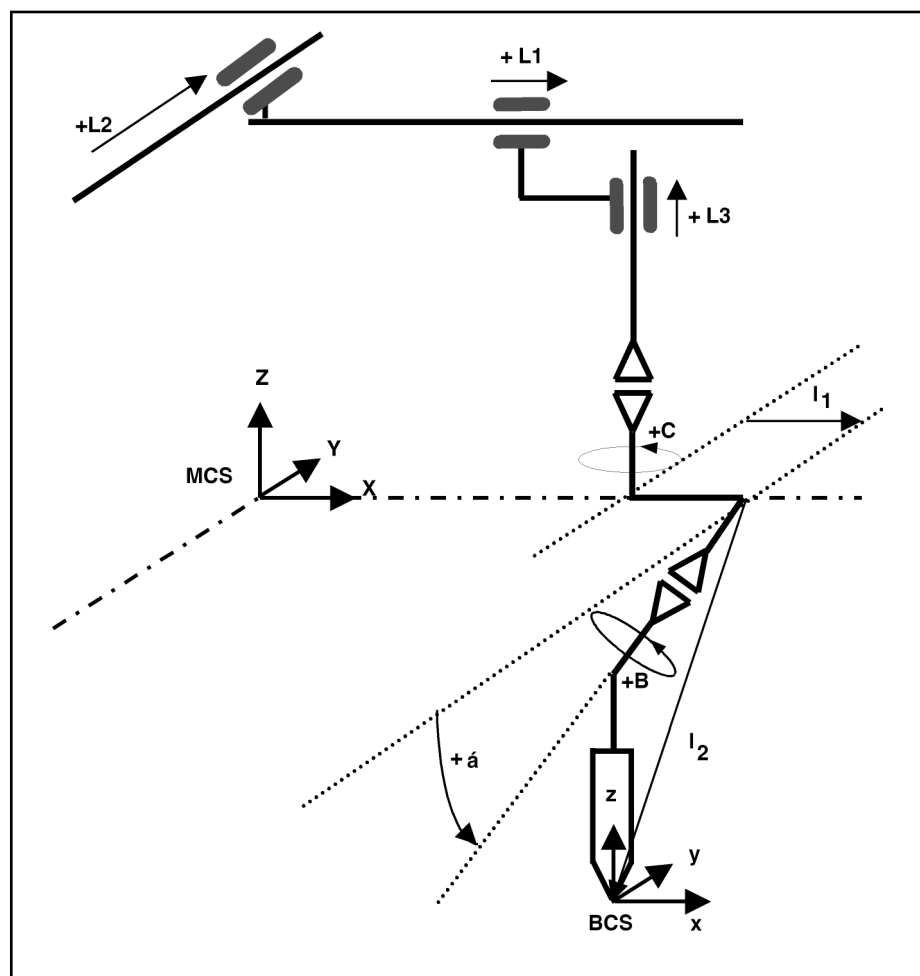


Abb. 11-53: Aufbau

In diesem Kapitel werden nur die, durch die verschiedenen Orientierungsbewegungen bedingten Unterschiede dargestellt. Ansonsten sei auf das [Kap. 11.13 "5-Achs-Transformation - Typ 4: Kardanischer Kopf "](#) auf Seite 298 verwiesen.

Die kardanische Achs-Transformation mit linearer Orientierung hat den Transformations-Typ 3232111.

Mit Einschalten dieser Achs-Transformation wird automatisch die lineare Orientierungsbewegung aktiviert, d. h. jede Orientierungsänderung durch Programmierung von φ und ϑ erfolgt durch eine lineare Interpolation von φ und ϑ .

Bei der Orientierungsbewegung kann die "kürzeste Weg"-Logik per MP Längenparameter [8] abgeschaltet werden.

φ und ϑ können Werte aus den folgenden Bereichen annehmen:

$$\vartheta \in [-\vartheta_{\max}, \vartheta_{\max}] \quad \text{und} \quad \varphi \in [-\infty, +\infty]$$

Die rotatorischen Achsen B und C müssen als Rundachsen konfiguriert sein (keine Modulo-Endlosachsen).

Mehrdeutigkeiten

Im Allgemeinen kann eine TCP-Position $\vec{r}$ und eine Werkzeugorientierung $\vec{o}$ durch zwei verschiedene Achsposen (Transformationszweige) erreicht werden. Die Zweige unterscheiden sich in den B-Achsenpositionen. Der Transformationszweig wird immer so gewählt, dass die B-Achse und der Orientierungswinkel ϑ das gleiche Vorzeichen besitzen, d. h., es gilt:

$$\vartheta \geq 0 \Leftrightarrow B \geq 0 \text{ (Zweig 1)}$$

$$\vartheta < 0 \Leftrightarrow B < 0 \text{ (Zweig -1)}$$

Dadurch ist die Rückwärtstransformation eindeutig lösbar. Ein Übergang von Zweig 1 nach -1 und umgekehrt ist ohne weiteres beim Durchfahren der Orientierungsstelle $\vartheta = 0^\circ$ möglich.

Nordpol

Der Nordpol kann ohne Weiteres durchfahren werden. Der Maschinenparameter 103000160 Epsilon-Umgebungen hat keine Wirkung, d. h. Orientierungsbewegungen durch die ε -Umgebung werden nicht zwangsweise durch die Stelle $\vartheta = 0^\circ$ geleitet. Bei $\vartheta = 0^\circ$ kann φ programmiert werden und die gewünschte Bewegung (Drehung der C-Achse) wird durchgeführt.

Restriktionen

Schiefe Ebene:

Die lineare Orientierungsbewegung ist für die Werkstücklagenkorrektur (BCR) und die Placement-Funktionalität ungeeignet, wenn diese eine Verdrehung des Koordinatensystems WCS bewirken. Die programmierten Positionen x,y,z beziehen sich korrekterweise auf das WCS, die Orientierungen φ und ϑ aber immer auf das BCS, d. h. nur die Bearbeitung mit konstanter Orientierung können sinnvoll ausgeführt werden. Stellen BCR und Placement eine reine Verschiebung dar, ist die Wirkungsweise korrekt.

Relevante NC-Funktionen

Coord (...)

Relevante Maschinenparameter (MP)

Die Kardanische Achs-Transformation mit linearer Orientierungsbewegung wird in den Parametern 103000110 bis 103000160 abgelegt.

1030 00110 Transformationstyp:	
3232111	(kardanische Achstransformation mit linearer Orientierungsbewegung)
1030 00160 Epsilon-Umgebungen	
	Ohne Bedeutung

Tab. 11-62: Relevante Maschinenparameter

11.14.2 Handlungsanweisung: Kardanischer Kopf mit linearer Orientierungsbewegung

Kardanische 5-Achs-Transformation in den Maschinenparametern oder per Systemdatum konfigurieren.

Im NC-Programm die kardanische 5-Achs-Transformation mit Coord (...) aktivieren.

11.15 5-Achs-Transformation - Typ 5: Minderbestückte RRLLL (mit Schwenk-/Rundtisch)

11.15.1 Beschreibung

Funktion

Name der Transformation	Transformationstyp	Wirkstelle	Koordinatensystem
Minderbestückte 5-Achs-Transformation mit Schwenk-/Rundtisch	3030013	AT2	MCS, BCS

Tab. 11-63: Funktion - minderbestückte RRLLL

Ausgangsgrößen im MCS:

$X, Y, Z, R_{2\text{fix}}, R_{1\text{fix}}$

Eingangsgrößen im BCS:

$x, y, z, \varphi_{\text{fix}}, \vartheta_{\text{fix}}$

Die minderbestückte 5-Achs-Transformation RRLLL besitzt drei Linearachsen (X,Y,Z) und zwei rotatorische Achsen (R1,R2). Die beiden rotatorischen Achsen werden nicht von der NC angesteuert, sondern sind ausschließlich unter Kontrolle der SPS. Für die NC existieren diese Achsen nicht.

Die anzufahrenden Positionen liegen jedoch allgemein in Werkstückkoordinaten (x,y,z,φ,θ) vor, d. h. sie sind durch fünf Freiheitsgrade (drei Positionen und zwei Orientierungen) festgelegt.

Während der eigentlichen Bearbeitung können die rotatorischen Achsen mechanisch nicht bewegt werden. Es ist nur möglich, sie vor der Bearbeitung per SPS zu positionieren.

Für einen speziellen Bearbeitungszyklus sind φ und θ immer konstant. Man benötigt die aus φ, θ resultierende Stellung der rotatorischen Achsen R₁, R₂, um die Ausgangsposition vor der Bearbeitung anfahren zu können. Dazu wird die minderbestückte 5-Achs-Transformation verwendet.

Die Transformation kommt an der Wirkstelle 2 zum Einsatz, d. h. sie wirkt zwischen dem Maschinenkoordinatensystem MCS = (X,Y,Z,R_{2fix}, R_{1fix}) und dem Basiskoordinatensystem BCS = (x,y,z,φ_{fix}, θ_{fix}).

Die rotatorischen Achsen können als Endlos- oder als Rundachsen definiert werden.

Bei dem Grundaufbau handelt es sich um die Konfiguration RRLLL (Transformationstyp 3232203). Das Werkzeug wird durch die drei Linearachsen X,Y,Z bewegt. Das Werkstück wird von der Rotationsachse R₁ (Rundtisch) gedreht. Die Schwenkachse R₂ schwenkt den Rundtisch.

In der Referenzstellung (BCS parallel zum MCS) dreht R₁ um Z (Typ C-Achse). Die minderbestückte 5-Achs-Transformation unterscheidet zwei Varianten der R₂-Achsanordnung (siehe nachfolgende Abbildungen):

- in der Referenzstellung dreht R₂ um X (Typ A-Achse)
- in der Referenzstellung dreht R₂ um Y (Typ B-Achse)

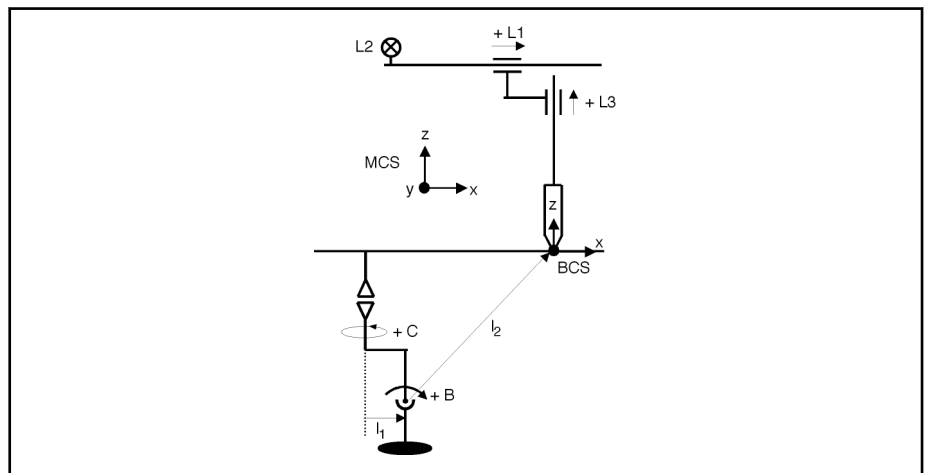


Abb. 11-54: RRLLL (Typ B-Achse)

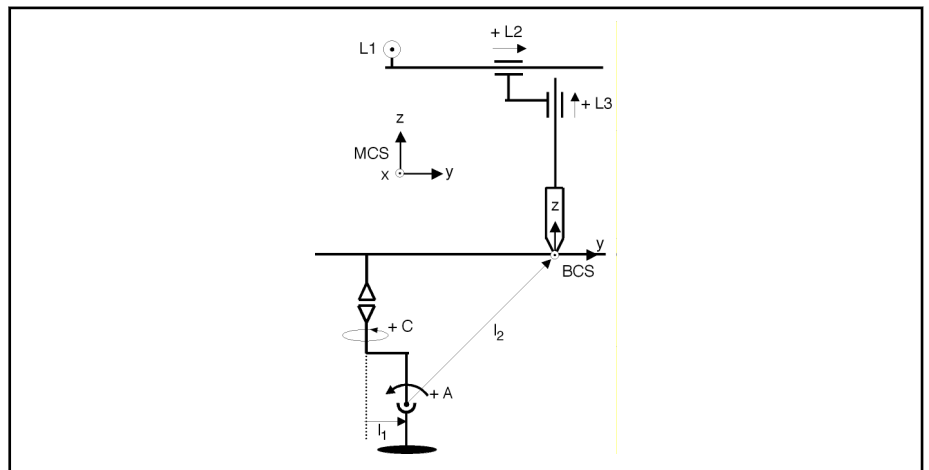


Abb. 11-55: RRLLL (Typ A-Achse)

Die Vektoren $\vec{l}_1$ und $\vec{l}_2$ definieren die Geometrie der minderbestückten

$$\vec{l}_1 = \begin{pmatrix} l_{1x} \\ l_{1y} \\ 0 \end{pmatrix} \quad \vec{l}_2 = \begin{pmatrix} l_{2x} \\ l_{2y} \\ l_{2z} \end{pmatrix}$$

5-Achs-Kinematik. Sie haben die Form

und

Achs-Transformation

Beim Typ B-Achse ist ein Vektor entlang x von der C-Achse zur B-Achse (d. h. $l_{1x} \neq 0$, $l_{1y} = 0$), beim Typ A-Achse ist ein Vektor entlang y von der C-Achse

zur A-Achse (d. h. $l_{1x} = 0$, $l_{1y} \neq 0$). $\vec{l}_2$ ist der $\vec{l}_1$ -Fortsetzungsvektor von der R_2 -Achse (B-/A-Achse) zum Nullpunkt des BCS.

Optionale Transformationsparameter:

Eingangsgrößen für die minderbestückte 5-Achs-Transformation sind die Liniarkoordinaten x, y, z. Beim Einschalten werden entweder die Orientierungskoordinaten (φ , ϑ) oder die Achspositionen (R_1 , R_2) inklusive Transformationszweig als fixe Transformationsparameter mit übergeben. Dies geschieht beim Aktivieren mittels drei optionaler Transformationsparameter per TRFOPT.

- Coord () TRFOPT ([PHI!], [THETA!], 1) oder
- Coord () TRFOPT ([C!], [B!], 0)

Im ersten Fall wird die Achstransformation mit fest vorgegebenem (φ , ϑ) und vorgegebenem Transformationszweig aktiviert.

Im zweiten Fall wird eine feste Achsstellung der rotatorischen Achsen (R_1 , R_2) übergeben.

Anhand des dritten Parameters wird steuerungintern erkannt, ob die Orientierungswinkel (Parameterwert = ± 1) oder die Achspositionen (Parameterwert = 0) übergeben werden.

Die fünf korrespondierenden Werte (φ , ϑ , R_1 , R_2 , Transformationszweig) werden in den Längenparametern (MP 1030 00140) gehalten und können nach Aktivierung der minderbestückten 5-Achs-Transformation innerhalb eines NC-Programms über die SD-Variable der aktiven Achstransformation 2 ausgelesen werden.

SD.SysAxTrafo2.LenParam[8]	φ_{fix}
SD.SysAxTrafo2.LenParam[9]	ϑ_{fix}
SD.SysAxTrafo2.LenParam[10]	$R_{1\text{fix}}$ (C-Achse)
SD.SysAxTrafo2.LenParam[11]	$R_{2\text{fix}}$ (A-/B-Achse)
SD.SysAxTrafo2.LenParam[12]	Transformationszweig (1 oder -1)

Tab. 11-64: SD-Variable

Besonderheiten: Mehrdeutigkeiten:

Im Allgemeinen kann eine TCP-Position und eine Werkzeugorientierung durch zwei verschiedene Achsposen (Transformationszweige) erreicht werden. Welche der beiden Posen (Transformationszweige) benutzt wird, entscheidet sich beim Einschalten der Achs-Transformation.

Werden die Orientierungskoordinaten (φ , ϑ) mit übergeben, entspricht der dritte optionale Parameter dem Transformationszweig.

Dabei bedeutet:

- Transformationszweig = +1
Es wird immer die Lösung gewählt, bei der die R_2 -Achse (A/B-Achse) einen positiven Wert hat.
- Transformationszweig = -1
Es wird immer die Lösung gewählt, bei der die R_2 -Achse (A/B-Achse) einen negativen Wert hat.

Werden die Achspositionen (R_1 , R_2) mit übergeben, muss der dritte optionale Parameter auf den Wert 0 gesetzt werden. Der Transformationszweig ergibt sich dann automatisch aus der Stellung der Achsen. In der Singularität (d. h. $R_2 = 0$) wird der Transformationszweig +1 festgelegt.

Ein Übergang von Zweig 1 nach -1 und umgekehrt ist nach Aktivieren der Achs-Transformation nicht mehr möglich.

Wertebereiche:

Die Orientierungswinkel (φ , ϑ) werden intern begrenzt auf:

$$0 \leq \varphi \leq 360$$

$$0 \leq \vartheta \leq 180 ,$$

für die Achsen ergeben sich immer Werte im Hauptbereich, d. h. bei Endlosachsen

$$0 \leq R_1 \leq 360$$

$$0 \leq R_2 \leq 360$$

und bei rotatorischen Achsen

$$\text{Zweig 1: } -180 \leq R_1 \leq 180$$

$$\text{Zweig -1: } -360 \leq R_1 \leq 0$$

$$0 \leq R_2 \leq 180$$

$$-180 \leq R_2 \leq 0 .$$

Referenzstellung: Die obigen Abbildungen zeigen die beiden RRLLL-Konfigurationen, Typ B-Achse mit $R_1=C$, $R_2=B$ und Typ A-Achse mit $R_1=C$, $R_2=A$, jeweils in der Referenzstellung. d. h. nimmt die Maschine die, in der Abbildung dargestellte Stellung ein, müssen die BCS-Koordinaten (x , y , z , φ , ϑ) = (0,0,0,0,0) sein,

bei zugehörigen MCS-Koordinaten (X, Y, Z, R_1, R_2) = ($X^0, Y^0, Z^0, R_1^0, R_2^0$) aus MP 1030 00130. Die positiven Dreh-/Verfahrrichtungen der Achsen sind ebenfalls in der Abbildung dargestellt. Diese sind entsprechend einzustellen.

In dieser Stellung steht die Spitze des rotationssymmetrischen Werkzeugs (Tool Center Point TCP) auf dem Ursprung des Basiskoordinatensystems (BCS) und ist entlang der z-Koordinate ausgerichtet. Die Ausrichtung des BCS ist durch die Achs-Transformation eindeutig festgelegt.

Werkzeugkorrektur: Eine Werkzeugkorrektur ist wie bei allen 5-Achs-Transformationen möglich.

$$\vec{l}_t = \begin{pmatrix} l_x \\ l_y \\ l_z \end{pmatrix}$$

Der Korrekturvektor zeigt von der Werkzeugspitze zur Werkzeugaufnahme.

Relevante NC-Funktionen

COORD () TRFOPT (... ,...)	
	Die minderbestückte 5-Achs-Transformation wird mit drei optionalen Parametern aktiviert. Dies kann wahlweise per Maschinenparametersatz ($i=1..20$) oder per MTB-Systemdatum (vom Typ SysAxTrafo_t) erfolgen.
COORD (0,2)	Die aktive Achstransformation wird deaktiviert.

Tab. 11-65: Relevante NC-Funktionen

Relevante Maschinenparameter (MP)

103000110	Transformationstyp
103000120	Systemachsen/-koordinaten der Transformation
103000125	Achsklassifikationen der Transformationsachsen
103000130	Achspositionen der Referenzstellung
103000140	Längen- und Winkelparameter
103000150	Referenzlage des Werkzeugkoordinatensystems
103000160	Epsilon-Umgebungen

Tab. 11-66: Relevante Maschinenparameter (MP)

11.15.2 Applizieren

Daten der Referenzstellung ermitteln:

Zum Einstellen der Werte müssen die Achsen in die Referenzstellung gebracht werden.

Einstellen der Maschinenparameter:

103000110	Transformationstyp eintragen 3030013 (minderbestückte RRLLL)
103000120	Systemachsen/-koordinaten der Transformation: Definiert die, an der Transformation beteiligten Systemachsen. Es gilt: [1] Systemachsnummer von X [2] Systemachsnummer von Y [3] Systemachsnummer von Z [4..8] nicht relevant

103000125	Achsklassifikationen der Transformationsachsen Definiert die Verfahrrichtung der Linearachsen und die implizit an der Transformation beteiligten rotatorischen Achsen inklusive Drehrichtung. Wert ">0" bedeutet: Die Achse verfährt positiv im Sinne der obigen Abbildung. Wert "<0" bedeutet: Die Achse verfährt negativ im Sinne der obigen Abbildung. Wert ">0" bedeutet: Die Achse verfährt positiv im Sinne der obigen Abbildung . Wert "<0" bedeutet: Die Achse verfährt negativ im Sinne der obigen Abbildung. Definiert sind: +/-1 = X-Achse +/-2 = Y-Achse +/-3 = Z-Achse +/-100 = A-Achse (rotatorisch, keine NC-Achse) +/-200 = B-Achse (rotatorisch, keine NC-Achse) +/-300 = C-Achse (rotatorisch, keine NC-Achse) +/-111 = A-Achse (endlos, keine NC-Achse) +/-222 = B-Achse (endlos, keine NC-Achse) +/-333 = C-Achse (endlos, keine NC-Achse)												
103000125	Ist ein Parameter unbesetzt, gelten folgende Default-Einstellungen: <table border="1" data-bbox="831 1205 1517 1464"> <tr> <td data-bbox="831 1205 963 1249">[1] 1</td><td data-bbox="963 1205 1517 1249">(alternativ -1)</td></tr> <tr> <td data-bbox="831 1249 963 1294">[2] 2</td><td data-bbox="963 1249 1517 1294">(alternativ -2)</td></tr> <tr> <td data-bbox="831 1294 963 1339">[3] 3</td><td data-bbox="963 1294 1517 1339">(alternativ -3)</td></tr> <tr> <td data-bbox="831 1339 963 1384">[4] 300</td><td data-bbox="963 1339 1517 1384">(alternativ -300, +/-333)</td></tr> <tr> <td data-bbox="831 1384 963 1429">[5] 200</td><td data-bbox="963 1384 1517 1429">(alternativ -200, +/-222, +/-100, +/-111)</td></tr> <tr> <td data-bbox="831 1429 963 1464">[6..8]</td><td data-bbox="963 1429 1517 1464">nicht relevant</td></tr> </table>	[1] 1	(alternativ -1)	[2] 2	(alternativ -2)	[3] 3	(alternativ -3)	[4] 300	(alternativ -300, +/-333)	[5] 200	(alternativ -200, +/-222, +/-100, +/-111)	[6..8]	nicht relevant
[1] 1	(alternativ -1)												
[2] 2	(alternativ -2)												
[3] 3	(alternativ -3)												
[4] 300	(alternativ -300, +/-333)												
[5] 200	(alternativ -200, +/-222, +/-100, +/-111)												
[6..8]	nicht relevant												
103000130	Achspositionen der Referenzstellung Definiert die Achspositionen der Referenzstellung. [1] X^0 [2] Y^0 [3] Z^0 [4] C^0 [5] R^2_0 (A^0 oder B^0) [6..8] nicht relevant												

103000140	Längen- und Winkelparameter	
	Definiert die Längenvektoren $\vec{l}_1, \vec{l}_2$ in [mm], Kopplungsfaktor, Default-Orientierungswinkel und Rundachspositionen in [Grad].	
	[1] l_{1x}	(Typ B-Achse)
	[2] l_{1y}	(Typ A-Achse)
	[3]	ohne Bedeutung
	[4] l_{2x}	
	[5] l_{2y}	
	[6] l_{2z}	
	[7] K_{BC}	Kopplungsfaktor von B- zur C-Achse
	[8] φ	Default-Wert für Orientierungswinkel φ , wird ggf. von TRFOPT überschrieben bzw. aus (R_1, R_2) berechnet.
	[9] ϑ	Default-Wert für Orientierungswinkel, wird ggf. von TRFOPT überschrieben bzw. aus (R_1, R_2) berechnet.
	[10] R_1	Default-Wert für Position der C-Achse, wird ggf. von TRFOPT überschrieben bzw. aus (φ, ϑ) berechnet
	[11] R_2	Default-Wert für Position der R_2 -Achse (A- oder B-Achse), wird ggf. von TRFOPT überschrieben bzw. aus (φ, ϑ) berechnet.
	[12] TrfBr	Default-Wert für den Transformationszweig (Wert 1 oder -1), wird ggf. von TRFOPT überschrieben bzw. aus (R_1, R_2) berechnet.
	[13..16]	ohne Bedeutung
103000150	Referenzlage des Werkzeugkoordinatensystems ohne Bedeutung	
103000160	Epsilon-Umgebungen ohne Bedeutung	

Tab. 11-67: Einstellen der Maschinenparameter

11.15.3 Aktivieren

Voraussetzung: Minderbestückte 5-Achs-Transformation ist per MP 1030 00110 bis MP 1030 00160 oder per MTB-Systemdatum vom Typ SysAxTrafo_t. konfiguriert.

Im NC-Programm wird die minderbestückte 5-Achs-Transformation für ein spezielles PHI!, THETA!, BRANCH! mit Coord (...) TRFOPT ([PHI!], [THETA!], [BRANCH!]) aktiviert und bleibt bis zum Ausschalten (COORD(0)) eingeschaltet.

Beispiel:

1

Orientierungswinkel (φ, ϑ) vorgegeben

Aufgabe: zugehörige Achspositionen (B, C) ermitteln und in

; Werkstückkoordinaten verfahren.

;

; Achs-Transformation mit gewünschtem phi, theta, Transformationszweig = -1 aktivieren

;

001 PHI! = 45

002 THETA! = 30

N001 Coord(1) TRFOPT ([PHI!], [THETA!], -1) ; minderbest. 5-Achstransformation wird aktiviert mit phi=45, theta=30, Transformationszweig=-1 (d. h. B-Achse wird immer negativ berechnet)

; die aus (phi=45, theta=30) resultierenden Positionen der C- und B-Achse
; werden eingelesen

020 WAIT

021 C! = SD.SysAxTrafo2.LenParam [10]

022 B! = SD.SysAxTrafo2.LenParam[11]

;

; jetzt müssen die errechneten Positionen der B- und C-Achse per SPS
; angefahren werden

; erst danach stimmen die angezeigten WCS-Koordinaten mit der Maschine
; überein

;

; erst danach stimmen die angezeigten WCS-Koordinaten mit der Maschine
; überein

;

N310 G1 F1000 x100 z200 ; x, z sind WCS-Koordinaten, d. h., die Werte
; beziehen sich auf das Werkstück-Koordinatensystem

:

: sonstige Verfahrssätze, alle bezüglich Orientierung (phi=45, theta=30)

:

N380 x50 y60 z70

N399 Coord(0) ;Achs-Transformation ausschalten

; _____ Ende Zyklus mit (phi=45, theta=30) _____

Beispiel:

2

Achspositionen (B, C)

vorgegeben Aufgabe: zugehörige Orientierungen (φ , ϑ) ermitteln

; und in Werkstückkoordinaten verfahren

;

; vorgegebene Position der B- und C-Achse per SPS anfahren.

; Achs-Transformation mit vorgegebenem B, C aktivieren

; zugehöriges phi, theta auslesen

;

Achs-Transformation

```

401 B! = 5
402 C! = 20
N403 Coord(1) TRFOPT([C!], [B!], 0) ; minderbestückte 5-Achs-Transforma-
tion wird aktiviert mit B=5, C=20, "0"=Kennung "B,C ist programmiert"
; die aus (B=5, C=20) resultierenden Orientierungen phi, theta werden
; eingelesen
404 WAIT
405 PHI! = SD.SysAxTrafo2.LenParam [8]
406 THETA! = SD.SysAxTrafo2.LenParam [9]
; _____
; es kann jetzt direkt in Werkstück-Koordinaten programmiert und
; verfahren werden.
; die angezeigten WCS-Koordinaten stimmen mit der Maschine überein
; _____
N410 G1 F1000 x100 z200 ; x, z sind WCS-Koordinaten, d. h. die Werte be-
ziehen sich auf das Werkstück-Koordinatensystems
:
: sonstige Verfahrssätze, alle bezüglich Orientierung (B=5, C=20)
:
N480 x50 y60 z70 N499 Coord(0)
; Achs-Transformation ausschalten
; _____ Ende Zyklus mit (B=5, C=20) _____

```

Beispiel:

3

Schräge Bohrungen

Bei schrägen Bohrungen ist folgende Vorgehensweise zu empfehlen:

Annahme: $\varphi \neq 0$, $\vartheta \neq 0$.

Es soll im Punkt (x_{pos} , y_{pos} , 0) des Basis-Werkstückkoordinatensystems BCS gebohrt werden. Man setzt im Punkt (x_{pos} , y_{pos} , 0) eine schiefe Ebene mit (φ , ϑ). Ab da kann man, wie in der nachfolgende Abbildung gezeigt, in den Koordinaten (x' , y' , z') des WCS_1 programmieren und verfahren. Die Position (0,0,10) liegt jetzt exakt 10 [mm] über dem Bohrloch. Die Position (0,0,-100) entspricht z. B. der tiefsten Bohrposition.

```

; _____
; Achs-Transformation mit gewünschtem phi, theta, Transformations-
zweig = -1 aktivieren,
; normales Vorgehen wie im Beispiel 2 "Achspositionen (B, C) vorgegeben"
; _____
N601 Coord(1) TRFOPT ([PHI!], [THETA!], -1)
; _____
; Mit dem nachfolgenden Befehl G152 wird eine schiefe Ebene im Punkt
(100,200,0); eingeschaltet.
; Es erfolgt keine Verfahrbewegung.
; Das neue Programmkoordinatensystem entspricht jetzt dem System WCS1

```


; der Abbildung.

;Der Nullpunkt liegt an der gewünschten Bohrstelle.

;

1000 XPOS! = 100

1001 YPOS! = 200

N602 G152 ([XPOS!], [YPOS!], 0, [PHI!], [THETA!], 0)

N603 G1 F1000 N604 x0 y0 z10 ; Position 10 [mm] über der Bohrposition anfahren

:

N650 z-100 ; 100 [mm] in das Teil hinein fahren (bohren)

:

N690 G153 ; schiefe Ebene ausschalten

N699 Coord(0) ; Achs-Transformation ausschalten

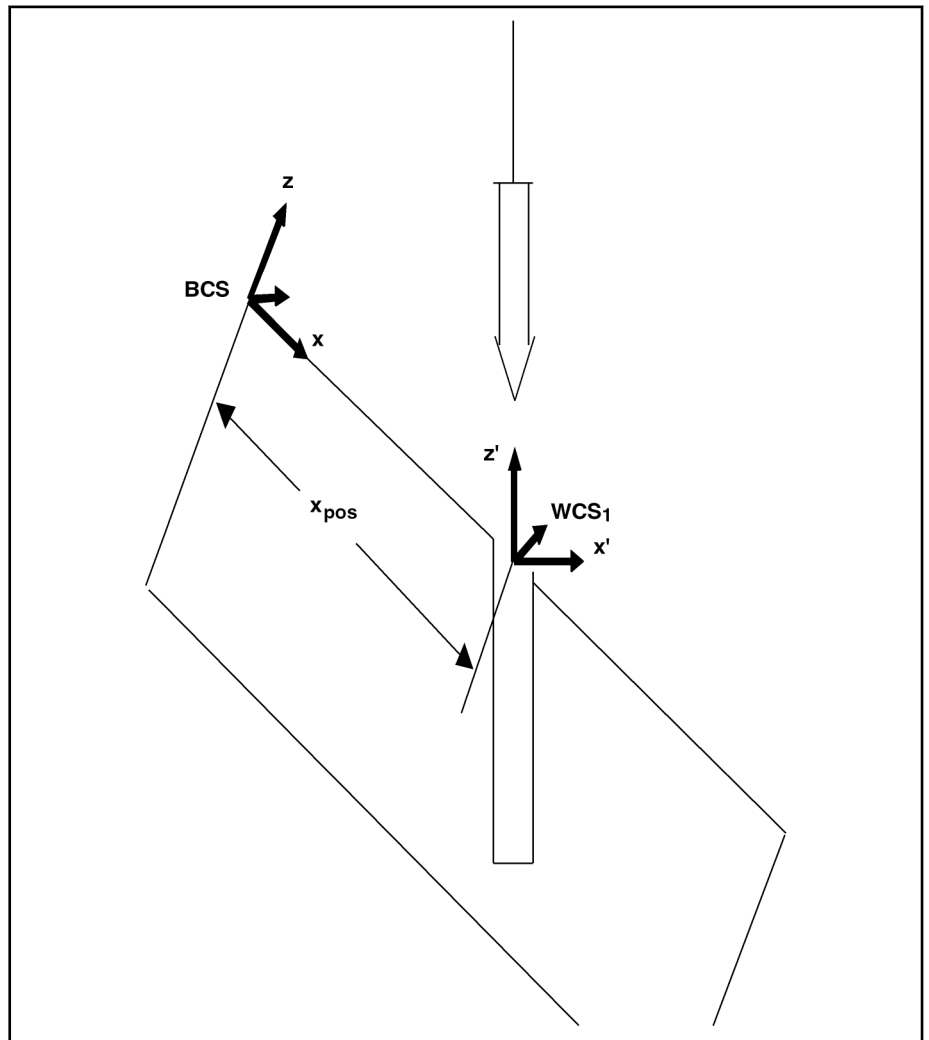


Abb. 11-56: Schräge Bohrung

11.16 5-Achs-Transformation - Typ 6: Kardanische RLLLR

11.16.1 Beschreibung

Funktion

Name der Transformation	Transformationstyp	Wirkstelle	Koordinatensystem
Kardanische 5-Achs-Transformation RLLLR			
Vektororientierung	3232212	AT2	MCS, BCS
lineare Orientierung	3232112	AT2	MCS, BCS

Tab. 11-68: Funktion

Ausgangsgrößen im MCS: X, Y, Z, R_2, R_1

Eingangsgrößen im BCS: x, y, z, φ, θ

Die kardanische 5-Achs-Transformation (RLLLR) besitzt drei Linearachsen (X, Y, Z) und zwei rotatorische Achsen (R_1, R_2).

Die rotatorischen Achsen R_1 und R_2 können sowohl Endlos- als auch Rundachsen sein.

Die Achs-Transformation kommt an der Wirkstelle 2 zum Einsatz, d. h. sie wirkt zwischen dem Maschinenkoordinatensystem MCS = (X, Y, Z, R_2, R_1) und dem Basiskoordinatensystem BCS = (x, y, z, φ, θ).

Die drei Linearachsen spannen ein rechtshändiges Koordinatensystem auf. Die Rotationsachse R_2 schwenkt das Werkzeug und die Rotationsachse R_1 dreht das (auf der Achse aufgespannte) Werkstück.

Bei der R_1 -Achse handelt es sich um eine C-Achse, d. h. in der Referenzstellung dreht sie um die Z-Achse. R_2 ist eine B-Achse, sie dreht um die Y-Achse. Die B-Achse ist gegenüber der xy-Ebene des MCS geneigt. Der Neigungswinkel α variiert im Bereich $0^\circ \leq \alpha \leq 89^\circ$. In den meisten praktischen Fällen ist $\alpha = 45^\circ$. Für den Fall $\alpha = 0^\circ$ ist die kardanische Achs-Transformation RLLLR mit der 5-Achs-Transformation vom Typ 3232202 identisch.

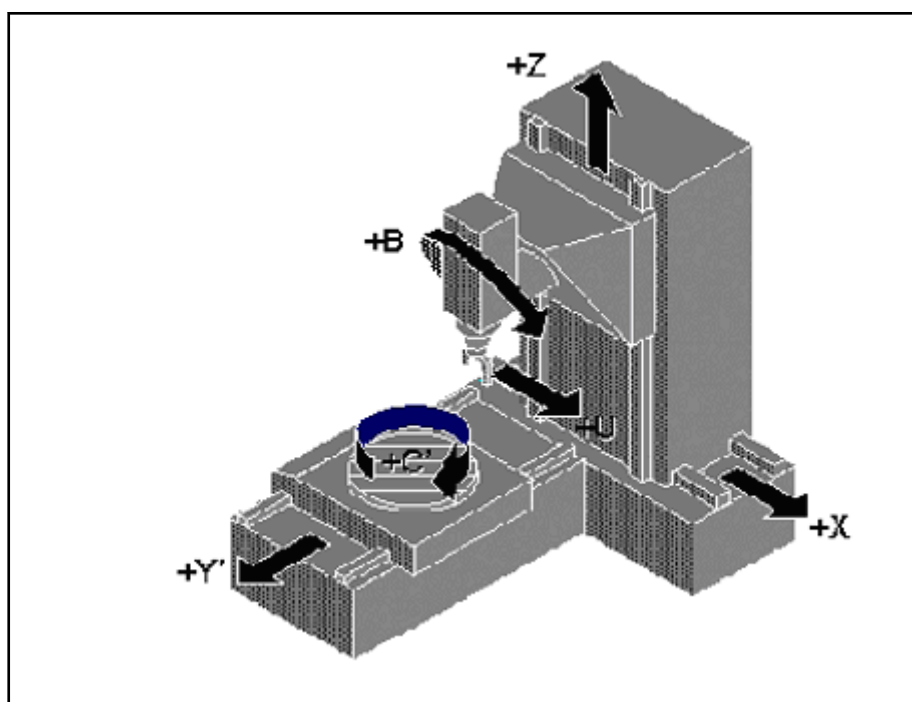


Abb. 11-57: Kardanische CLLLB (schematisch)

Achs-Transformation

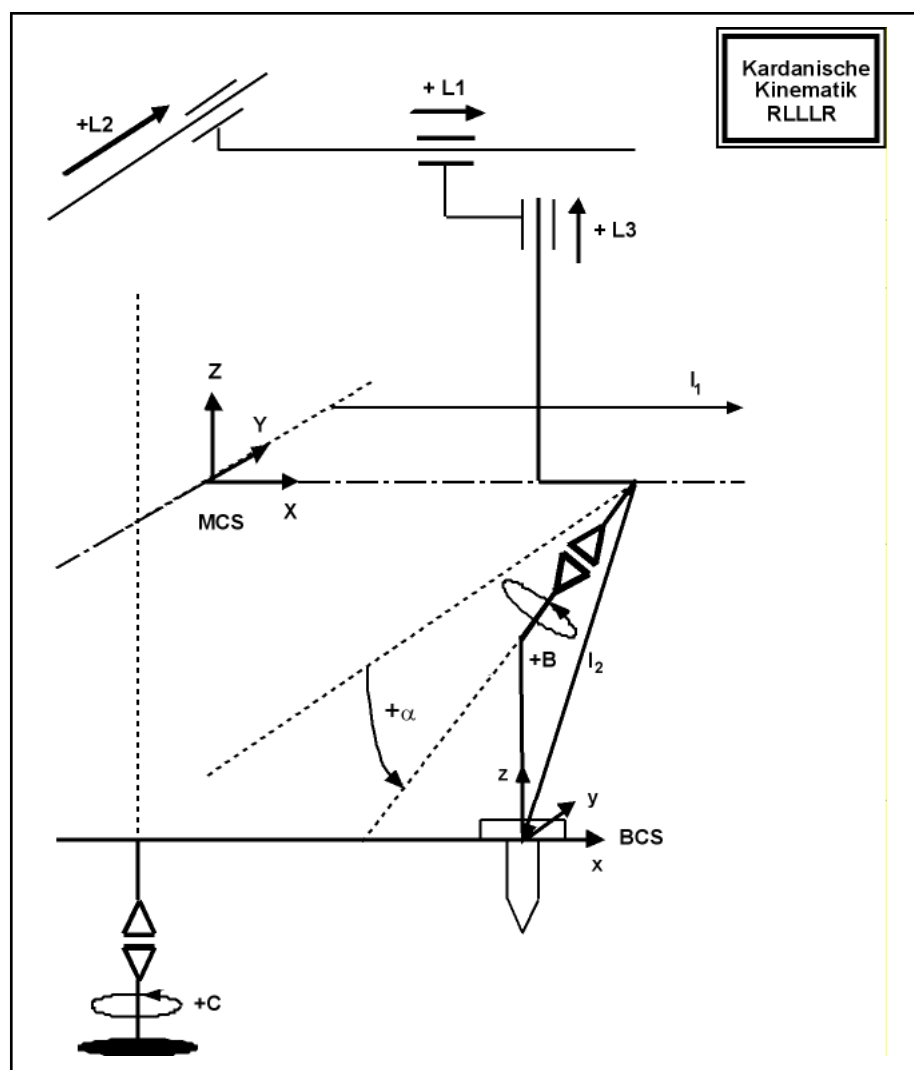


Abb. 11-58: Kardanische CLLLB (Referenzstellung)

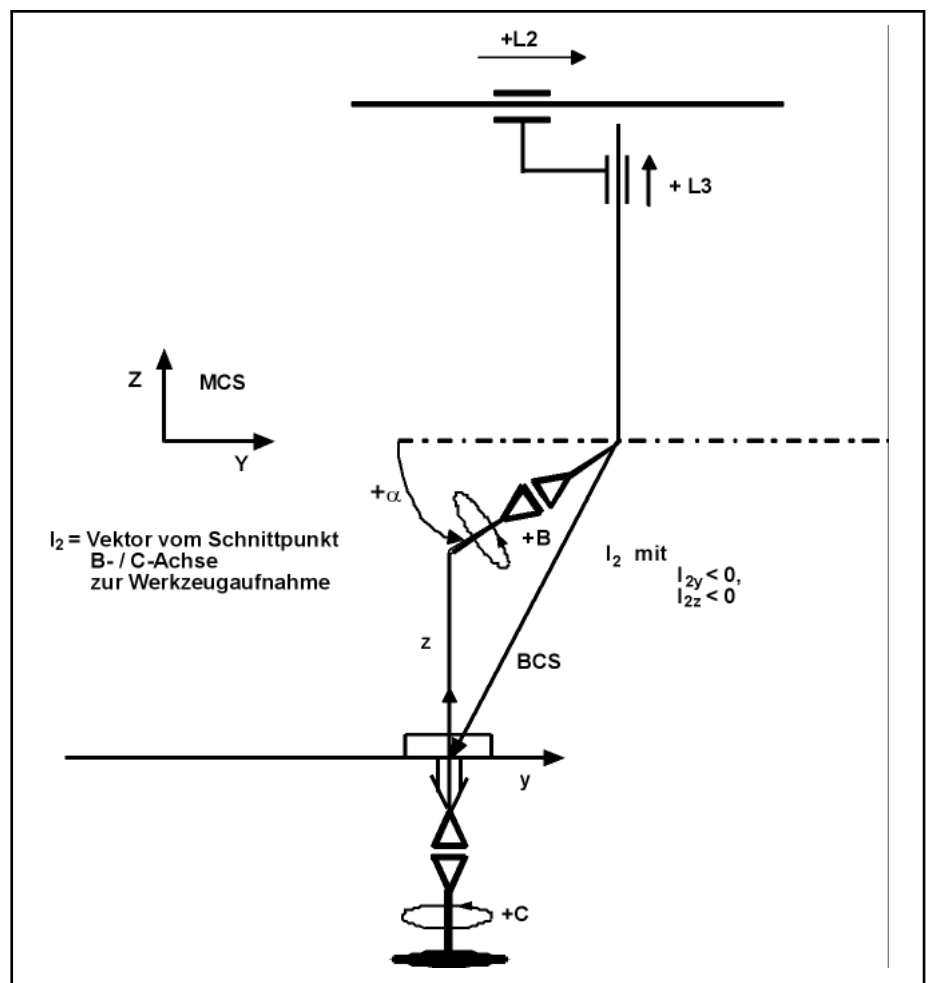


Abb. 11-59: Kardanische CLLLB (Seitenansicht Referenzstellung)

Der Neigungswinkel α und die Vektoren $\vec{i}_1$ und $\vec{i}_2$ definieren die Geometrie der 5-Achs-Kinematik.

$\vec{l}_1$ ist ein Vektor entlang X von der R_1 -Achse (C-Achse) zur R_2 -Achse (B-Achse) (d. h. $l_{1x} \neq 0$, $l_{1y} = l_{1z} = 0$).

$\vec{l}_2$ ist der $\vec{l}_1$ -Fortsetzungsvektor von der R_2 -Achse (B-Achse) zum Nullpunkt des BCS (d. h. $l_{1x} \neq 0, l_{1y} \neq 0, l_{1z} \neq 0$).

Bedingt durch die geneigte B-Achse ist der Winkel θ zwischen z_{BCS} und der Werkzeugsymmetrieachse nicht größer als $\theta_{\text{max}} = 2(90^\circ - \alpha)$

Eigenschaften der Achstransformation:

Bei aktiver 5-Achs-Transformation werden die Koordinaten x, y, z, φ, θ des Basiskoordinatensystem BCS in das Maschinenkoordinatensystem MCS transformiert.

Eingangsgrößen sind die Linearkoordinaten x , y , z und die Orientierungskoordinaten φ und θ . Mit Einschalten dieser Achs-Transformation wird entweder die Vektororientierungsbewegung (Typ 3232212) oder die lineare Orientierungsbewegung (Typ 3232112) aktiviert.

Der Orientierungsvektor $\vec{\phi}$ kann als Spline programmiert werden (siehe [Kap. 9.1.8 "Spline programmieren "](#) auf Seite 148).

In Abhängigkeit vom Orientierungstyp ergeben sich unterschiedliche Eigenschaften

Vektor-Ori:

Bei der Vektororientierungsbewegung erfolgt jede Orientierungsänderung durch Programmierung von φ , θ oder einer entsprechenden Alternativsyntax (s. Spez. Orientierungsbewegung) durch eine Drehung des Orientierungsvektors $\vec{\phi}$ (φ, θ) um eine raumfeste Drehachse (Großkreis-Bewegung der Vektorspitze). φ , θ können Werte aus den folgenden Bereichen annehmen:

$\varphi \in [0, +360]$, $\theta \in [0, \theta_{\max}]$.

Die rotatorischen Achsen B und C können sowohl Endlos- als auch Rundachsen sein.

Linear-Ori:

Die Winkel φ und θ werden linear vom Anfangs- zum Endpunkt interpoliert. Das Verhalten der "kürzeste Weg"-Logik hängt von der Programmierungsart der Orientierungsbewegung ab. Bei φ -, θ -Programmierung ist die "kürzeste Weg"-Logik ausgeschaltet. Bei der O(...)-Programmierung ist die "kürzeste Weg"-Logik eingeschaltet. φ, θ können Werte aus den folgenden Bereichen annehmen:

$\varphi \in [-\text{unendl}, +\text{unendl}]$, $\theta \in [-\theta_{\max}, \theta_{\max}]$.

Die rotatorischen Achsen B und C müssen als Rundachsen konfiguriert sein (keine Modulo-Endlosachse).

Restriktion

Eine Online-Korrektur für φ und θ ist nicht möglich.

Relevante NC-Funktionen

COORD(<i>)	Mit $i=1..20$ wird eine Achs-Transformationen aktiviert, die in MP 1030 00110 definiert ist.
COORD(0,2)	Die aktive Transformation wird deaktiviert.

Tab. 11-69: Relevante NC-Funktionen

Relevante CPL-Funktionen

SDR(300)	Wirksame Epsilon-Umgebung. Der wirksame Wert kann größer als der in EpsAxTrafo (MP 1030 00160) eingestellte Wert sein.

Tab. 11-70: Relevante CPL-Funktionen

Relevante Maschinenparameter

1030 00110	Definiert den Transformationstyp.
1030 00120	Definiert die, an der Transformation beteiligten Systemachsen/-koordinaten der Transformation.
1030 00125	Definiert die an der Transformation beteiligten rotatorischen Achsen inklusive Drehrichtung und die Verfahrrichtung der Linearachsen.
1030 00130	Definiert die Achspositionen der Referenzstellung

1030 00140	Definiert die Längen- und Winkelparameter
1030 00160	Epsilon-Umgebungen

Tab. 11-71: Relevante Maschinenparameter

Referenzstellung

Die Referenzstellung der kardanischen 5-Achs-Transformation RLLLR ist in Abb. 11-58 "Kardanische CLLLLB (Referenzstellung)" auf Seite 322 und in Abb. 11-59 "Kardanische CLLLLB (Seitenansicht Referenzstellung)" auf Seite 323 dargestellt. In der Referenzstellung der Konfiguration (MCS: X, Y, Z, $R_1 = C$ und $R_2 = B$) ist die Spitze (TCP) des rotationssymmetrischen Werkzeugs (Fräser oder Laser) auf dem Ursprung des Basiskoordinatensystems (BCS) entlang der z-Koordinate ausgerichtet. Die Ausrichtung des BCS ist durch die Achs-Transformation eindeutig festgelegt (siehe Abb.)

In dieser Stellung müssen die BCS-Koordinaten $(x,y,z,\varphi,\theta) = (0,0,0,0,0)$ sein, bei zugehörigen MCS-Koordinaten $(X, Y, Z, R_2, R_1) = (X^0, Y^0, Z^0, R_2^0, R_1^0)$ aus MP 1030 00130.

Werkzeugkorrektur

Die Achs-Transformation unterstützt eine Werkzeugkorrektur im Werkzeugkoordinatensystem (TCS). Das TCS stimmt in der Referenzstellung mit dem BCS überein. Der Werkzeugkorrekturvektor $\vec{t}_t$ zeigt vom korrigierten Werkzeugmittelpunkt TCP_1 zum unkorrigierten TCP_0 . Er setzt sich aus den Längenkorrekturen L1, L2 und L3 aus der D-Korrekturtabelle zusammen.

Besonderheiten

- **Mehrdeutige Achspositionen:**

Im Allgemeinen kann eine TCP-Position und eine Werkzeugorientierung durch zwei verschiedene Achsposen (Transformationszweige) erreicht werden. Die Zweige unterscheiden sich in den B-Achspositionen.

Vektor-Ori:

Welche der beiden Posen (Transformationszweige) benutzt wird, entscheidet sich beim Einschalten der Achs-Transformation. Ein Übergang von Zweig 1 nach -1 und umgekehrt ist danach nur beim Durchfahren der Orientierungsstelle $\theta = 0$ (Nordpolsingularität) oder $\theta = \theta_{\max}$ (Äquator-singularität) möglich.

Linear-Ori:

Der Transformationszweig wird immer so gewählt, dass die B-Achse und der Orientierungswinkel θ das gleiche Vorzeichen besitzen, d. h. es gilt:

- $\theta \geq 0 \triangleq B \geq 0$ (Zweig 1)
- $\theta \leq 0 \triangleq B \leq 0$ (Zweig -1)

Dadurch ist die Rückwärtstransformation eindeutig lösbar. Ein Übergang von Zweig 1 nach -1 und umgekehrt ist ohne weiteres beim Durchfahren der Orientierungsstelle $\theta = 0$ (Nordpolsingularität) möglich.

- **Nordpolsingularität:**

Vektor-Ori:

Der Nordpol stellt eine Singularität dar.

Im Nordpol wirkt der Maschinenparameter 103000160 Epsilon-Umgebungen, d. h. Orientierungsbewegungen durch die ε -Umgebung werden zwangsweise durch die Stelle $\theta = 0$ geleitet.

Bei $\theta = 0$ kann φ programmiert werden, es erfolgt jedoch keine Bewegung.

Linear-Ori:

Der Nordpol stellt keine Singularität dar.

Der Nordpol kann ohne weiteres durchfahren werden. Der Maschinenparameter 103000160 Epsilon-Umgebungen hat keine Wirkung, d. h. Orientierungsbewegungen durch die ε -Umgebung werden nicht zwangsweise durch die Stelle $\theta = 0$ geleitet.

Bei $\theta = 0$ kann φ programmiert werden und die gewünschte Bewegung (Drehung der C-Achse) wird durchgeführt.

- **Äquatorsingularität:**

Bei Orientierungsbewegungen tangential durch den Breitenkreis $\theta = \theta_{\max}$ wird der Transformationszweig nur dann gewechselt, wenn die B-Achse einen Verfahrbereich von mindestens $B \in [-360^\circ, +360^\circ]$ hat. Der Verfahrbereich ergibt sich aus dem negativen und positiven ersten Endschalter der B-Achse.

11.16.2 Handlungsanweisung: 5-Achs-Transformation Kardanische RLLLR

Applizieren Die Achstransformation kann über die relevanten Maschinenparameter appliziert werden. Dazu muss die Funktion Achstransformation über den Parameter **tra** "Achstransformationen" im Setup (SUP) aktiviert werden. Der Pfad des MP-Namens ist immer "TRA/AxTrafo[1..20]/...".

Alternativ dazu kann die Achstransformation über ein benutzerdefiniertes Systemdatum vom Typ SysAxTrafo_t appliziert werden.

Die beiden Varianten sind inhaltlich identisch.

Die Achspositionen der Referenzstellung müssen ermittelt werden.

In der nachfolgenden Tabelle sind jeweils die äquivalenten MP-Nummern, MP-Name und SD-Elemente angegeben. Um eine Transformation zu applizieren, muss eine der Varianten entsprechend besetzt werden.

Transformations-relevante Parameter

MP-Nummer	Beschreibung	
1030 00110	Transformationstyp	
	MP-Name: TrafoType	
	SD-Elem. : Type	
	3232212	kardanische RLLLR mit Vektor-Orientierung
	3232112	kardanische RLLLR mit Linear-Orientierung

MP-Nummer	Beschreibung	
1030 00120	Systemachsen/-koordinaten der Transformation	
	MP-Name: FwdInCoordIndTrafo[1..8]	
	SD-Elem. : AxisAssignment[1..8]	
	[1]	Systemachsnummer von C
	[2]	Systemachsnummer von X
	[3]	Systemachsnummer von Y
	[4]	Systemachsnummer von Z
	[5]	Systemachsnummer von B
	[6..8]	nicht relevant

Achs-Transformation

MP-Nummer	Beschreibung		
1030 00125	Achsklassifikationen der Transformationsachsen		
	MP-Name: CoordClass[1..8]		
	MP-Name: CoordDir[1..8]		
	SD-Elem. : AxisClassification[1..8]		
	Definiert die an der Transformation beteiligten Achsen inklusive Bewegungsrichtung.		
	- MP-Nr. oder SD-Element: Wert ">0" bedeutet: Die Achse verfährt/dreht positiv im Sinne der obigen Abbildungen. Wert "<0" bedeutet: Die Achse verfährt/dreht negativ im Sinne der obigen Abbildungen. Mögliche Werte sind: +/-1 = X-Achse +/-2 = Y-Achse +/-3 = Z-Achse +/-100 = A-Achse +/-200 = B-Achse +/-300 = C-Achse - MP-Name: - Mögliche Werte für CoordClass sind - X,Y,Z (linear) - A,B,C (rotatorisch) - Mögliche Werte für CoordDir sind: - positiv (bzgl. Abb. "Referenzstellung") - negativ (bzgl. Abb. "Referenzstellung")		
	Ist ein Parameter unbesetzt, gelten folgende Default-Einstellungen:		
		MP-Nr / SD-Elem.:	CoordClass: CoordDir:
	[1]	300	C positiv
	[2]	1	X positiv
	[3]	2	Y positiv
	[4]	3	Z positiv
	[5]	200	B positiv
	[6..8]	nicht relevant	0 positiv

MP-Nummer	Beschreibung
1030 00130	Achspositionen der Referenzstellung MP-Name: RefPosTrafo[1..8] SD-Elem. : AxZeroPos[1..8]
	[1] C ⁰
	[2] X ⁰
	[3] Y ⁰
	[4] Z ⁰
	[5] B ⁰
	[6..8] nicht relevant
1030 00140	Längen- und Winkelparameter MP-Name: JointParTrafo[1..16] SD-Elem. : LenParam[1..16] Definiert die Längenvektoren l ₁ , l ₂ in [mm].
	[1] l _{1x}
	[2] nicht relevant
	[3] nicht relevant
	[4] l _{2x}
	[5] l _{2y}
	[6] l _{2z}
	[7] α
	[8..16] nicht relevant
1030 00160	Epsilon-Umgebungen MP-Name: EpsAxTrafo SD-Elem. : EpsilonRanges
	0.01

Tab. 11-72: Transformations-relevante Parameter

Aktivieren

Ist die 5-Achs-Transformation konfiguriert per

- Maschinenparameter, dann wird die Transformation im NC-Programm aktiviert mit (i=1,..20):
COORD(<i>)
- Systemdatum, dann wird die Transformation im NC-Programm aktiviert mit (Typ "SysAxTrafo_t"):
COORD(<SD.MtbAxTrafo>)

Programmbeispiel:

Die 5-Achs-Transformation ist im 4. Maschinenparametersatz definiert. Am Programmanfang ist keine Achstransformation aktiv:

Transformation aktivieren

```

N100 G1 F1000 XA10 YA10 C30 ;Axes XA, YA and C are traversed
N110 COORD(4) ;Switching on the axis transformation
N130 x20 y30 phi0 ;the coordinates x,y,phi are traversed
N140 x10 z10 theta45 ;the coordinates x,z,theta are traversed

```

```

:
N999 COORD(0) ;Switching-off the axis transformation
;Axes are again programmed from here

```

Deaktivieren COORD(0) bzw. COORD(0,2)

schaltet die Achstransformation im NC-Programm aus.

11.17 3-Achstransformation LLR

11.17.1 Beschreibung

Funktion

Diese Transformation kommt an Wirkstelle 2 zum Einsatz. Sie wirkt also zwischen dem Maschinenkoordinatensystem (MCS) und dem Basiskoordinatensystem (BCS).

Seitens des MCS werden 2 Linearkoordinaten und eine rotatorische Koordinate benötigt, die im folgenden mit X_{MCS} , Y_{MCS} , C_{MCS} bezeichnet werden, deren Namen jedoch per Maschinenparameter beliebig festgesetzt werden können.

Daraus ergeben sich im BCS 2 Linearkoordinaten und eine rotatorische Koordinate, die im folgenden mit x_{BCS} , y_{BCS} , ϕ_{BCS} bezeichnet werden. Auch die Namen dieser Koordinaten können per Maschinenparameter beliebig festgesetzt werden. Dabei ist (x_{BCS}, y_{BCS}) die Position der Werkzeugspitze (Tool Center Point = TCP) und ϕ_{BCS} die Werkzeug-Orientierung, d.h., der Winkel zwischen der x_{BCS} -Richtung und dem von der Werkzeugspitze längs der Werkzeugachse in Richtung Werkzeughalterung weisenden Vektor.

Das BCS ist fest mit dem Werkstück verbunden.

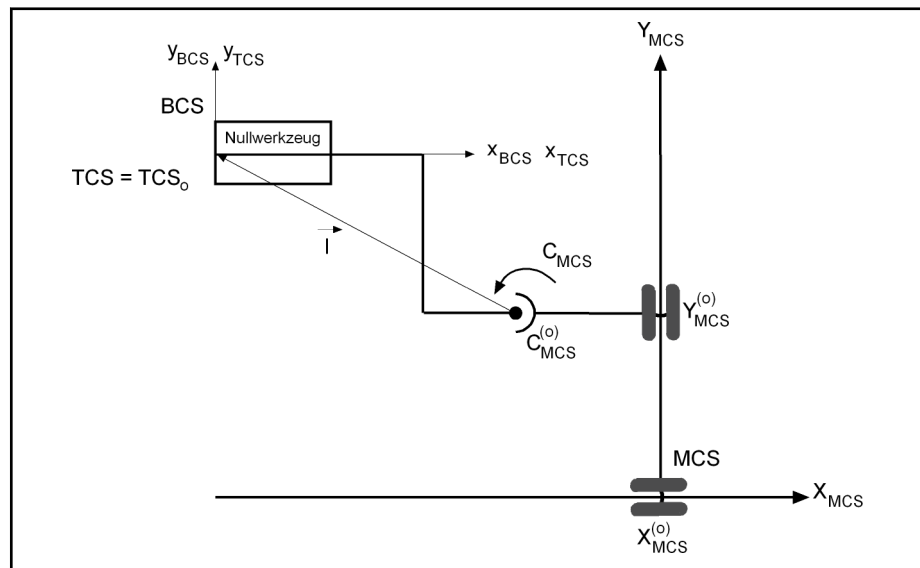


Abb. 11-60: 3-Achstransformation - Referenzlage mit Nullwerkzeug

In Referenzlage (Abb. 11-60 "3-Achstransformation - Referenzlage mit Nullwerkzeug" auf Seite 330) gilt:

- Die Orientierung des Werkzeugs ist gegeben durch $\phi_{BCS} = 0$.
- Der Vektor vom Drehpunkt der rotatorischen MCS-Koordinate C_{MCS} zur Position $x_{BCS} = y_{BCS} = 0$ ist gegeben durch den Vektor I .

- Die Komponenten I_x und I_y von I im BCS sind im Maschinenparameter JointParTrafo (1030 00 140) einzutragen. Die obige Abbildung zeigt den Fall $I_x < 0$, $I_y > 0$.
- Die MCS-Positionen sind gegeben durch $X_{MCS}^{(0)}$, $Y_{MCS}^{(0)}$, $C_{MCS}^{(0)}$, die im Maschinenparameter RefPosTrafo (1030 00 130) einzutragen sind.
- Die Spitze des Nullwerkzeugs steht bei $x_{BCS} = y_{BCS} = 0$ (Definition des Nullwerkzeugs). Die Abbildung zeigt diesen Fall. Für ein Werkzeug mit anderen Werkzeuglängen hat die Werkzeugschneidspitze eine andere Position.

Das Werkzeugkoordinatensystem (TCS) hat seinen Nullpunkt in der Werkzeugschneidspitze und ist fest mit dem Werkzeug verbunden. Für das Nullwerkzeug ist das TCS in der Referenzlage deckungsgleich mit dem BCS und wird mit TCS_0 bezeichnet. Für ein Werkzeug mit anderen Werkzeuglängen ist das TCS in der Referenzlage gegenüber dem BCS verschoben.

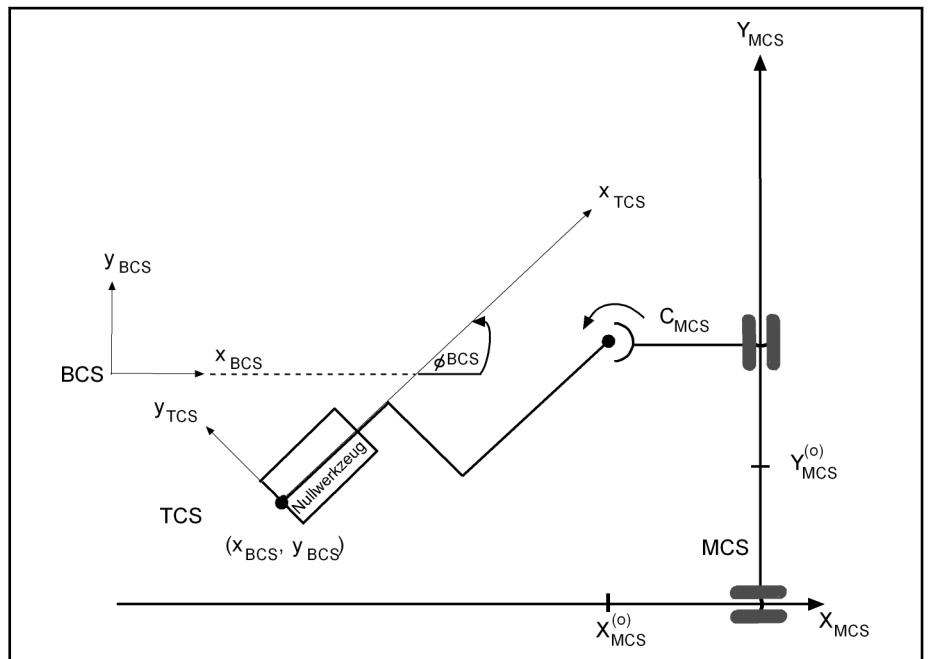


Abb. 11-61: 3-Achstransformation - Beliebige Lage mit Nullwerkzeug

Die Abbildung [Abb. 11-61 "3-Achstransformation - Beliebige Lage mit Nullwerkzeug" auf Seite 331](#) zeigt eine beliebige Lage der Achstransformation mit dem Nullwerkzeug.

Die Achstransformation unterstützt eine Werkzeugkorrektur im TCS. Der Werkzeugkorrekturvektor I_t zeigt von der aktuellen Werkzeugschneidspitze (TCP) zur Werkzeugschneidspitze des Nullwerkzeugs (TCP₀). Seine Komponenten I_{tx} , I_{ty} , I_{tz} werden im TCS angegeben. Siehe Abbildung [Abb. 11-62 "3-Achstransformation - Referenzlage mit beliebigem Werkzeug" auf Seite 332](#).

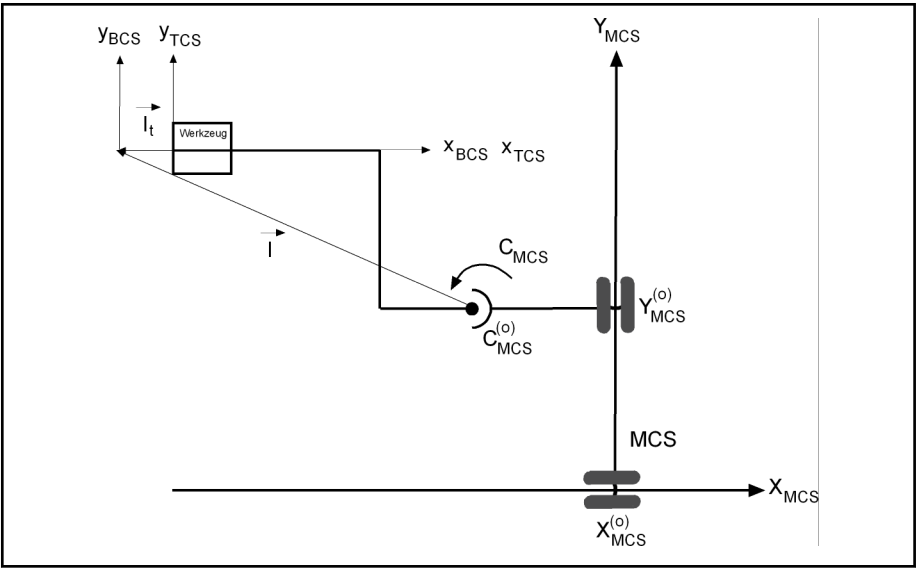


Abb. 11-62: 3-Achstransformation - Referenzlage mit beliebigem Werkzeug

Da zur Achstransformation nur 2 Linearkoordinaten des BCS gehören, kann im BCS immer nur eine von den 3 Ebenen G17, G18 und G19 aktiviert werden. Ob dies G17, G18 oder G19 ist, hängt nicht von den Namen der BCS-Koordinaten ab, für die hier beispielhaft x_{BCS} , y_{BCS} , ϕ_{BCS} gewählt wurde, sondern wird durch den Maschinenparameter $FwdInCoord[1..3]/CoordClass$ (1030 00 125) festgelegt.

Relevante NC-Funktionen

Das Ein- und Ausschalten von Achstransformationen erfolgt mit dem NC-Befehl "Coord", der in der folgenden Tabelle beschrieben ist.

Coord(i)	Es wird die Achstransformation mit der Nummer i (i=1,...,20) aktiviert. Soll das die hier beschriebene Achstransformation sein, ist diese unter der Nummer i in den Maschinenparametern zu parametrieren.
Coord(0)	Ist eine Transformation an Wirkstelle 2 aktiv, wird diese deaktiviert.
Coord(0,2)	Ist eine Transformation an Wirkstelle 2 aktiv, wird diese deaktiviert.

Tab. 11-73: 3-Achstransformation: Ein- und Ausschalten

Die Aktivierung der Werkzeugkorrektur im TCS erfolgt mit dem NC-Befehl G47 ($\{ \langle TCS\text{-Koordinate, auf die sich L1 bezieht} \rangle, \langle TCS\text{-Koordinate, auf die sich L2 bezieht} \rangle, \langle TCS\text{-Koordinate, auf die sich L3 bezieht} \rangle \}$). Dabei steht "XTR" für die durch den Maschinenparameter $LinCoord[1]$ bezeichnete TCS-Koordinate, "YTR" für " $LinCoord[2]$ " und "ZTR" für " $LinCoord[3]$ ".
Daneben kann die Werkzeugkorrektur auch auf WCS-Koordinaten bezogen werden, was in der Dokumentation "MTX Funktionsbeschreibung - Grundlagen", Kap. "Werkzeugkorrekturen" beschrieben ist.

Relevante Maschinenparameter

CHAN	Ch[k]	Coord	Mcs	CartCoord[1...6]
			Wcs	LinCoord[1..3]
				OriCoord[1...3]

TRA	AxTrafo[1...20]	TrafoType		
		FwdInCoord	FwdInCoordIndTrafo[1...8]	
		JointPar	JointParTrafo[1...8]	
		RefPos	RefPosTrafo[1...8]	
		FwdInCoordClass	FwdInCoord[1...8]	CoordClass
				CoordDir
			EpsAxTrafo	
		TcsRefOri	TcsX	CoordTcsXBcs
				DirTcsXBcs
			TcsY	CoordTcsYBcs
				DirTcsYBcs
			TcsZ	CoordTcsZBcs
				DirTcsZBcs

Tab. 11-74: Baumstruktur der für die 3-Achstransformation relevanten Maschinenparameter

7080 00020	Mcs	Maschinenkoordinaten
7080 00010	Wcs	Werkstückkoordinaten
1030 00110	TrafoType	Achstransformationstyp
1030 00120	FwdInCoord	Eingangsachsen/-koordinaten der Vorwärtstransformation
1030 00140	JointPar	Gelenkparameter (Längen-/Winkelparameter)
1030 00130	RefPos	Achs-/Maschinenpositionen der Referenzstellung
1030 00125	FwdInCoordClass	Arbeitstechn. Bedeutung der Eingangsachsen/-koordinaten
1030 00150	TcsRefOri	Referenzorientierung des WZ-Koordinatensystems TCS

Tab. 11-75: Macoda-Nummern der relevanten Maschinenparameter

11.17.2 Applizieren

Zum Einstellen der Maschinenparameter muss die Referenzlage der Achstransformation angefahren werden.

Zuerst ist die Nummer i ($i=1...20$) festzulegen, unter der die Achstransformation mit $\text{Coord}(i)$ zu aktivieren sein soll. Damit sind folgende Maschinenparameter im Verzeichnis $/\text{TRA}/\text{AxTrafo}[i]$ einzutragen. "k" bezeichnet durchweg die Kanalnummer:

Achs-Transformation

TrafoType (1030 00 110)		
	Hier ist für diese Achstransformation 2121002 einzutragen.	
FwdInCoordIndTrafo[1] (1030 00 120)		
	Nummer der 1. MCS-Koordinate der Transformation (oben mit X_{MCS} bezeichnet)	
FwdInCoordIndTrafo[2] (1030 00 120)		
	Nummer der 2. MCS-Koordinate der Transformation (oben mit Y_{MCS} bezeichnet)	
FwdInCoordIndTrafo[3] (1030 00 120)		
	Nummer der 3. MCS-Koordinate der Transformation (oben mit C_{MCS} bezeichnet)	
Für FwdInCoordIndTrafo[1] und FwdInCoordIndTrafo[2] (1030 00 120) sind folgende Zahlenwerte zulässig:		
	-6:	Es ist die, durch CartCoord[1] (7080 00 020) bezeichnete MCS-Koordinate einer aktiven Achstransformation an Wirkstelle 1 gemeint.
	-5:	Es ist die, durch CartCoord[2] (7080 00 020) bezeichnete MCS-Koordinate einer aktiven Achstransformation an Wirkstelle 1 gemeint.
	-4:	Es ist die, durch CartCoord[3] (7080 00 020) bezeichnete MCS-Koordinate einer aktiven Achstransformation an Wirkstelle 1 gemeint.
	1...99:	Die MCS-Koordinate entspricht 1:1 (bis auf die NPV) einer Systemachse und es ist die Systemachsnummer dieser Systemachse gemeint. Es muss sich dabei um eine Linearachse handeln.
Für FwdInCoordIndTrafo[3] (1030 00 120) sind folgende Zahlenwerte zulässig:		
	-3:	Es ist die, durch CartCoord[4] (7080 00 020) bezeichnete MCS-Koordinate einer aktiven Achstransformation an Wirkstelle 1 gemeint.
	-2:	Es ist die, durch CartCoord[5] (7080 00 020) bezeichnete MCS-Koordinate einer aktiven Achstransformation an Wirkstelle 1 gemeint.
	-1:	Es ist die, durch CartCoord[6] (7080 00 020) bezeichnete MCS-Koordinate einer aktiven Achstransformation an Wirkstelle 1 gemeint.
	1...99:	Die MCS-Koordinate entspricht 1:1 (bis auf die NPV) einer Systemachse und es ist die Systemachsnummer dieser Systemachse gemeint. Es muss sich dabei um eine Rund- oder Endlosachse handeln.

Tab. 11-76: Applizieren

Für den Maschinenparameter FwdInCoord[1..3]/CoordClass (1030 00125) sind nur die folgenden Einstellungen (Tab. 11-77 "Zusammenhang der auswählbaren Ebene und den für die TCS-Werkzeugkorrektur zulässigen Parametern mit den Maschinenparametern RefPosTrafo und LinCoord" auf Seite 335) zulässig, wobei jede Spalte einer Einstellmöglichkeit entspricht. Die Auswirkungen auf die Bedeutung der Maschinenparameter LinCoord und OriCoord sowie auf die auswählbare Ebene sind jeweils darunter aufgelistet.

Auswählbare Ebene und zulässige Parameter für die TCS-Werkzeugkorrektur

FwdInCoord[1]/CoordClass (1030 00 125)	X	Z	Y
FwdInCoord[2]/CoordClass (1030 00 125)	Y	X	Z
FwdInCoord[3]/CoordClass (1030 00 125)	C	B	A
Name der lin. BCS-Hauptkoord.	LinCoord[1]	LinCoord[3]	LinCoord[2]
Name der lin. BCS-Nebenkoord.	LinCoord[2]	LinCoord[1]	LinCoord[3]
Name der rotatorischen BCS-Koord.	LinCoord[4]	LinCoord[4]	LinCoord[4]
Einzigste auswählbare Ebene	G17	G18	G19
Parameter der TCS-Werkzeug- korrektur	XTR, YTR	ZTR, XTR	YTR, ZTR

Tab. 11-77: Zusammenhang der auswählbaren Ebene und den für die TCS-Werkzeugkorrektur zulässigen Parametern mit den Maschinenparametern RefPosTrafo und LinCoord

FwdInCoord[1]/CoordDir (1030 00 125)	Vorzeichen, mit dem die 1. MCS-Koordinate der Transformation berücksichtigt wird.
FwdInCoord[2]/CoordDir (1030 00 125)	Vorzeichen, mit dem die 2. MCS-Koordinate der Transformation berücksichtigt wird.
FwdInCoord[3]/CoordDir (1030 00 125)	Vorzeichen, mit dem die 3. MCS-Koordinate der Transformation berücksichtigt wird.

Tab. 11-78: Vorzeichen

Die Bilder in der oben stehenden Beschreibung der Grundlagen der Achs-transformation gehen davon aus, dass für FwdInCoord[1..3]/CoordDir (1030 00125) jeweils "positiv" eingetragen wird. Verläuft eine MCS-Koordinatenrichtung entgegengesetzt dazu, ist für diese "negativ" einzutragen.

RefPosTrafo[1] (1030 00 130)	Referenzposition der 1. MCS-Koordinate der Transformation (oben mit $X_{MCS}^{(0)}$ bezeichnet)
RefPosTrafo[2] (1030 00 130)	Referenzposition der 2. MCS-Koordinate der Transformation (oben mit $Y_{MCS}^{(0)}$ bezeichnet)
RefPosTrafo[3] (1030 00 130)	Referenzposition der 3. MCS-Koordinate der Transformation (oben mit $C_{MCS}^{(0)}$ bezeichnet)
JointParTrafo[1] (1030 00 140)	1. Längen- und Winkelparameter der Transformation (oben mit l_x bezeichnet)
JointParTrafo[2] (1030 00 140)	2. Längen- und Winkelparameter der Transformation (oben mit l_y bezeichnet)
CoordTcsXBcs, DirTcsXBcs (1030 00 150)	CoordTcsXBcs ist die Koordinate des BCS (x, y oder z), deren, durch DirTcsXBcs gegebene, Richtung (positiv oder negativ) in Referenzlage mit Nullwerkzeug mit der x-Koordinate des TCS zusammenfällt.

CoordTcsYBcs, DirTcsYBcs (1030 00 150)	CoordTcsYBcs ist die Koordinate des BCS (x, y oder z), deren, durch DirTcsYBcs gegebene, Richtung (positiv oder negativ) in Referenzlage mit Nullwerkzeug mit der y-Koordinate des TCS zusammenfällt
CoordTcsZBcs, DirTcsZBcs (1030 00 150)	CoordTcsZBcs ist die Koordinate des BCS (x, y oder z), deren, durch DirTcsZBcs gegebene, Richtung (positiv oder negativ) in Referenzlage mit Nullwerkzeug mit der z-Koordinate des TCS zusammenfällt.

Tab. 11-79: Applizieren

In den obigen 3 Abbildungen ist der Fall:

CoordTcsXBcs = x,
 CoordTcsYBcs = y,
 CoordTcsZBcs = z,
 DirTcsXBcs = positiv,
 DirTcsYBcs = positiv,
 DirTcsZBcs = positiv
 dargestellt.

11.17.3 Aktivieren

1. Maschinenparameter für die Achstransformation eintragen.
2. NC-Neustart.
3. Achstransformation mit Coord(i) aktivieren.
4. Alle im NC-Programm verwendeten NC-Funktionen auf ihre Anwendbarkeit bei aktiver Transformation 2121002 überprüfen.

11.18 6-Achs-Transformation

11.18.1 Beschreibung

Allgemeines

Die 6-Achs-Transformation wird auf solchen Maschinen angewendet, deren Werkzeug

- zeitgleich translatorische und rotatorische Bahnbewegungen im Raum ausführen muss
- zusätzlich zur Bahnbewegung noch Orientierungsbewegungen durchführt, die das Werkzeug in jedem Bahnpunkt ausrichten

Mit Hilfe der 6-Achs-Transformation werden die Programmkoordinaten in Achspositionen der beteiligten Achsen gewandelt.

Zu der, in der NC realisierten 6-Achs-Transformation gehören:

- drei Linearkoordinaten (z. B. x, y, z)
- drei Orientierungskoordinaten φ , ϑ , ψ (z. B. phi, theta, psi),
- drei Linearachsen (z. B. X, Y und Z)
- drei rotatorische Achsen (z. B. A, B und C)

Für die 6-Achs-Transformation existieren zwei Versionen (siehe [Kap. 11.18.2 "Variante 1 \(mit Tensor-Orientierung\)"](#) auf Seite 337 und [Kap. 11.18.3 "Variante 2 \(mit linearer Bewegung der Rotationsachsen\)"](#) auf Seite 338).

11.18.2 Variante 1 (mit Tensor-Orientierung)

Funktion

Name der Transformation	Transformationstyp	Wirkstelle	Koordinatensystem
6-Achs-Transformation mit Tensor-Orientierung	3333301	AT2	MCS, BCS

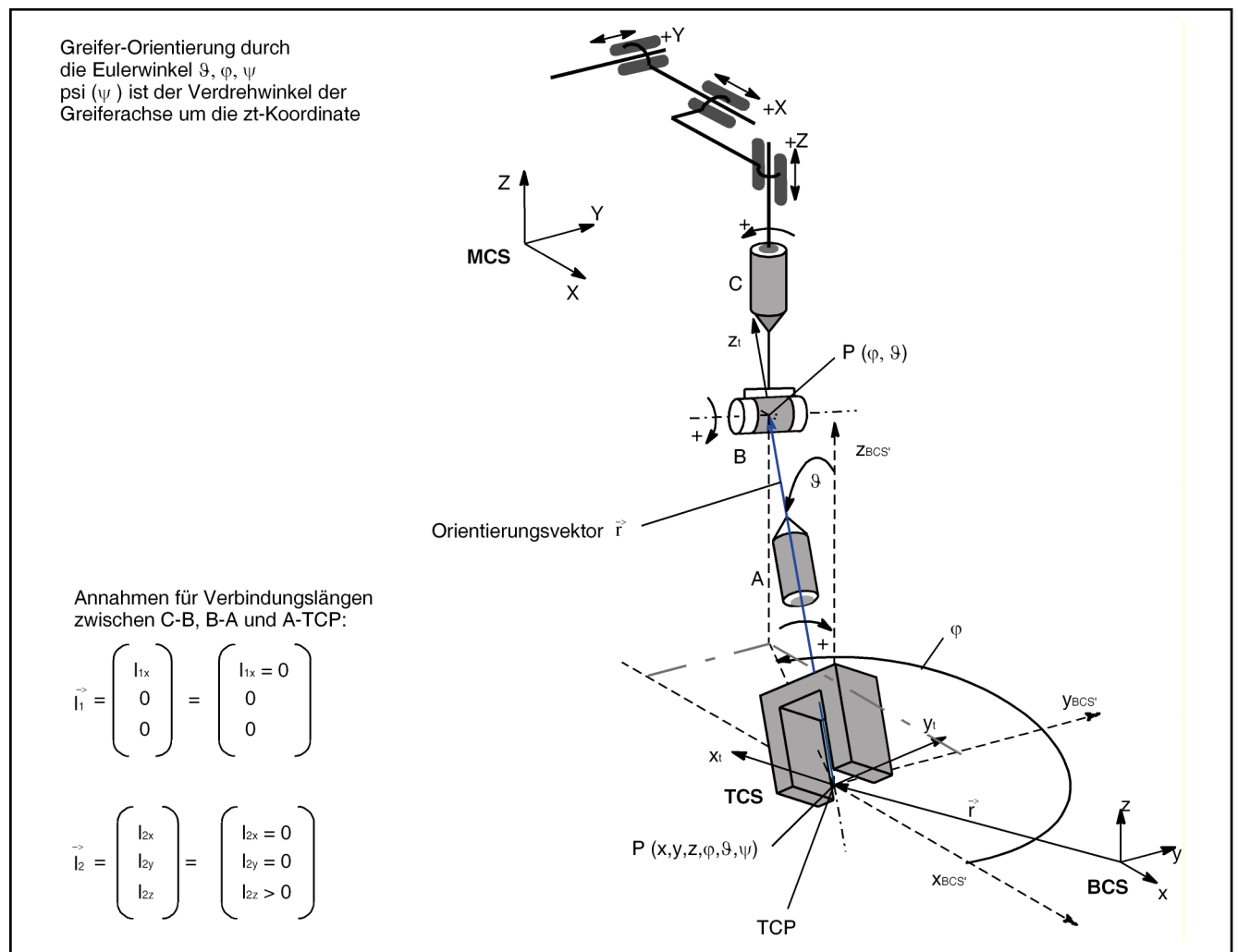
Tab. 11-80: Funktion - Variante 1

Ausgangsgrößen im MCS: X,Y,Z,C,B,A

Eingangsgrößen im BCS: x, y, z, φ , ϑ , ψ

Typ 3333301 erlaubt die TCP-Programmierung über drei Linearkoordinaten und die Orientierung des Werkzeugs (TCS) durch Programmieren der Eulerwinkel φ (phi), ϑ (theta) und ψ (psi) (Eulerwinkel siehe [Kap. 10.2.2 "Eulerwinkel"](#) auf Seite 162).

Die Orientierungsbewegung erfolgt als TCS-Drehung um eine raumfeste Drehachse (siehe folgende Abb.).



11.18.3 Variante 2 (mit linearer Bewegung der Rotationsachsen)

Funktion

Name der Transformation	Transformationstyp	Wirkstelle	Koordinatensystem
6-Achs-Transformation mit linearer Bewegung der Rotationsachsen	3033001	AT2	MCS, BCS

Tab. 11-81: Funktion - Variante 2

Ausgangsgrößen im MCS: X,Y,Z,C,B,A

Eingangsgrößen im BCS: x,y,z,C,B,A

Typ 3033001 unterstützt die TCP-Programmierung über drei Linearkoordinaten und die Orientierung des Werkzeugs durch Programmieren der drei Rotationsachsen.

Die Orientierungsbewegung erfolgt linear in den Rundachsenpositionen.

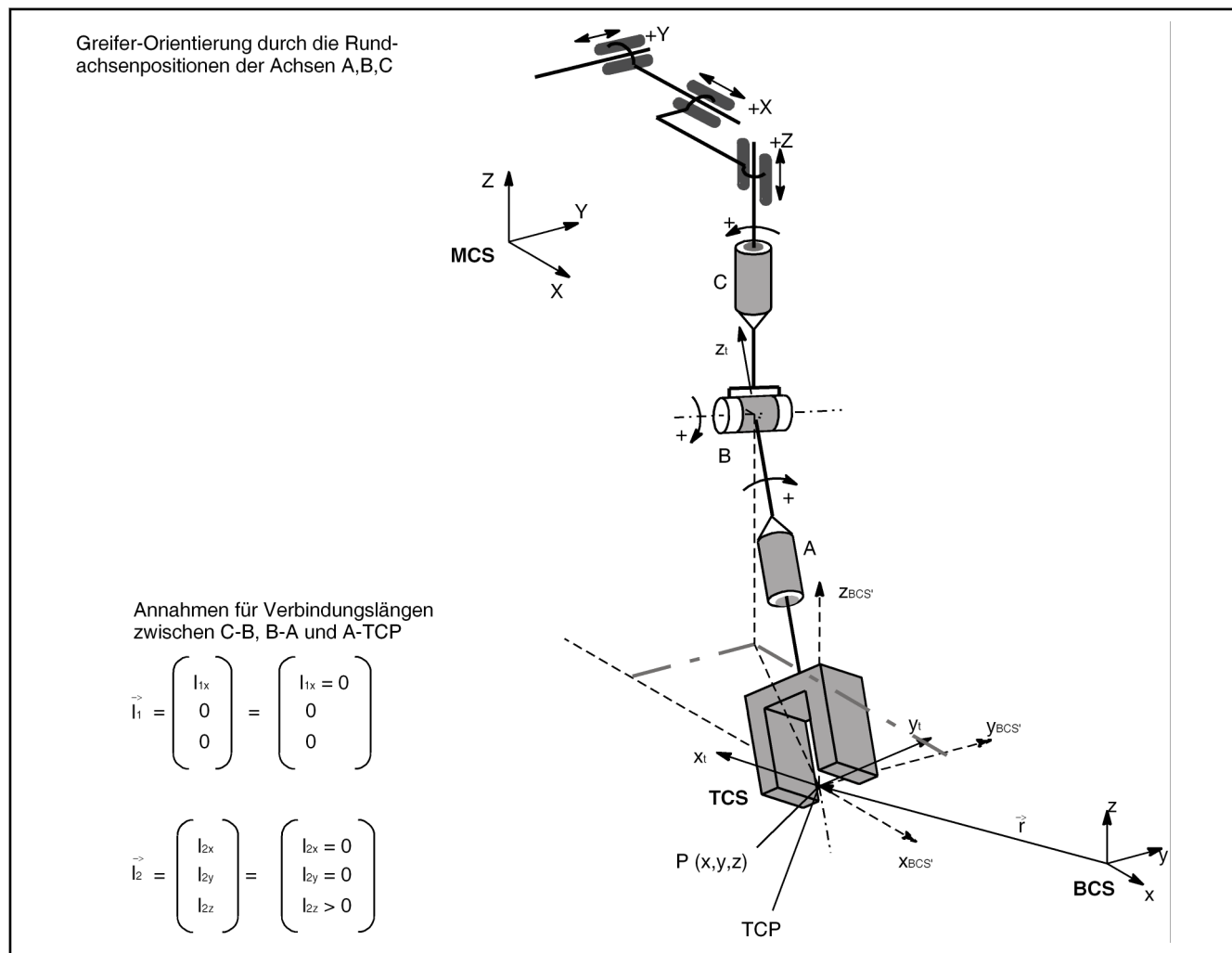


Abb. 11-64: lineare Bewegung der Rotationsachsen

11.18.4 Gemeinsame Eigenschaften der Varianten 1, 2

Relevante NC-Funktionen

COORD(<i>)	Mit i=1..20 wird eine 5-Achs-Transformation aktiviert, die in MP 1030 00110 definiert ist.
COORD(0,2)	Die aktive Transformation wird deaktiviert.

Tab. 11-82: Relevante NC-Funktionen

Relevante Maschinenparameter

1030 00110	Definiert den Transformationstyp
1030 00120	Definiert die an der Transformation beteiligten Systemachsen/-koordinaten der Transformation.
1030 00125	Definiert die Achsklassifikation der Transformationsachsen
1030 00130	Definiert die Achspositionen der Referenzstellung
1030 00140	Definiert die Längen- und Winkelparameter
1030 00150	Referenzlage des Werkzeugkoordinatensystems TCS
1030 00160	Epsilon-Umgebungen

Tab. 11-83: Relevante Maschinenparameter

Typ und Position der Rotationsachsen

Die Rotationsachsen sind hintereinander am Werkzeug angeordnet und stehen senkrecht aufeinander. In der Referenzstellung drehen sie das Werkzeug jeweils um eine der linearen Basiskoordinaten x, y oder z in positiver Drehrichtung.

Die Positionen der Rotationsachsen werden mit α_i , α_j , α_k bezeichnet. Die Indizes i, j und k (1, 2 oder 3) geben jeweils den Typ der Achse an.

Der Typ wird wie folgt definiert:

Rotationsachsentyt	Erklärung	Achsklassifikation in MP 1070 00125
1	Die Rotationsachse dreht um x. Ihre Position wird mit α_1 bezeichnet.	100
2	Die Rotationsachse dreht um y. Ihre Position wird mit α_2 bezeichnet.	200
3	Die Rotationsachse dreht um z. Ihre Position wird mit α_3 bezeichnet.	300

Tab. 11-84: Rotationsachsentyt

Reihenfolge der Rotationsachsen bzgl. des Werkzeugs:

Die Rotationsachsen vom Typ 1, 2 oder 3 können in **sechs** verschiedenen Reihenfolgen am Werkzeug angeordnet sein.

Es ergeben sich folgende **Subkonfigurationen**:

Subkonfigurations- typ	Position der ersten Rotationsachse (am weitesten vom TCP entfernt)	Position der zwei- ten Rotationsachse	Position der dritten Rotationsachse (nahe am TCP)
123	α_1	α_2	α_3
132	α_1	α_3	α_2
213	α_2	α_1	α_3
231	α_2	α_3	α_1
312	α_3	α_1	α_2
321	α_3	α_2	α_1

Tab. 11-85: Subkonfigurationen

Beispiel:

Subkonfigurationen

Typ 132 = die erste Rotationsachse dreht um die x-Koordinate des BCS, die zweite um die z-Koordinate und die dritte um die y-Koordinate.

Subkonfigurationen der Rotationsachsen-Anordnung:

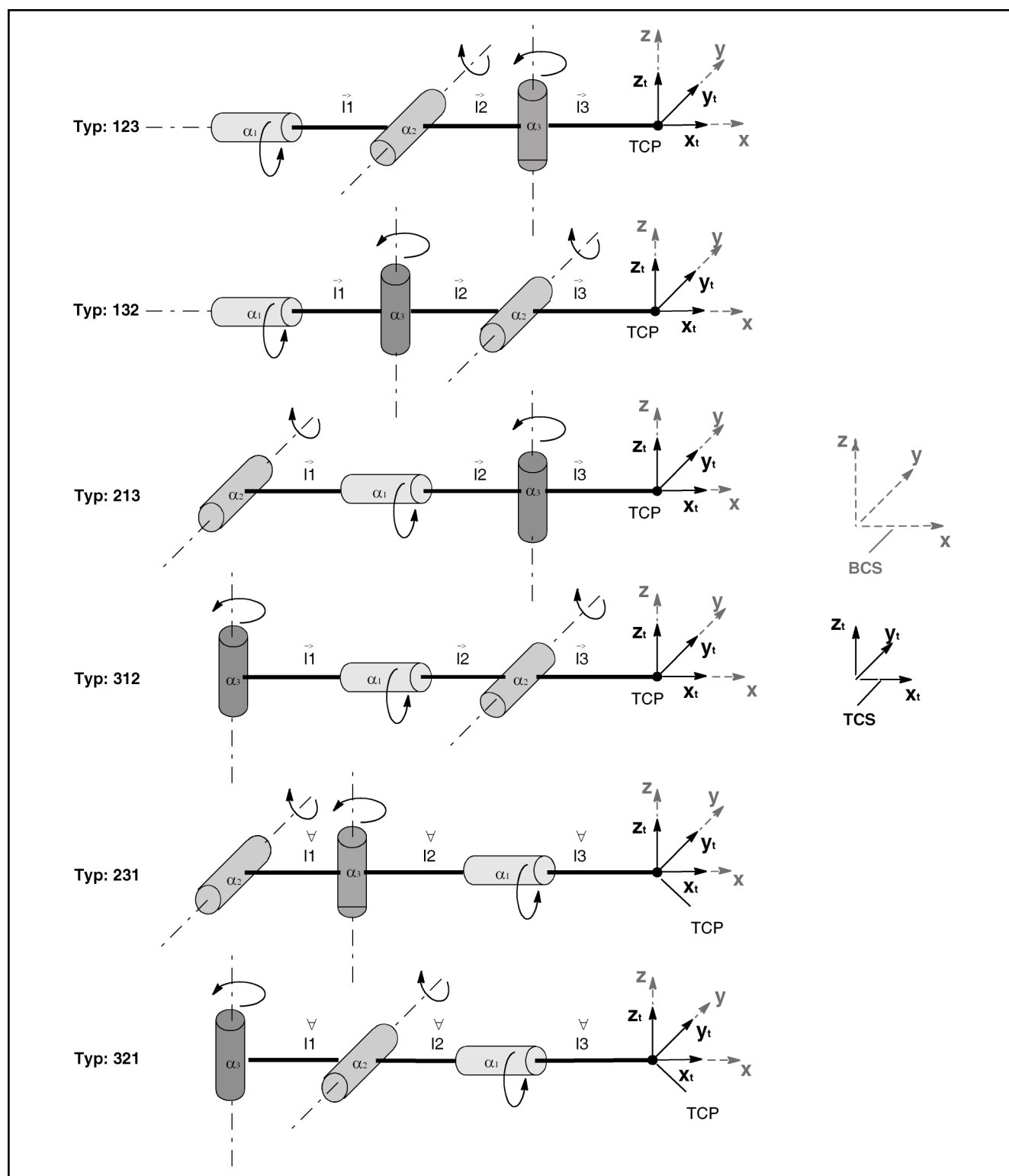


Abb. 11-65: Subkonfigurationen der Rotationsachsen-Anordnung

Referenzstellung

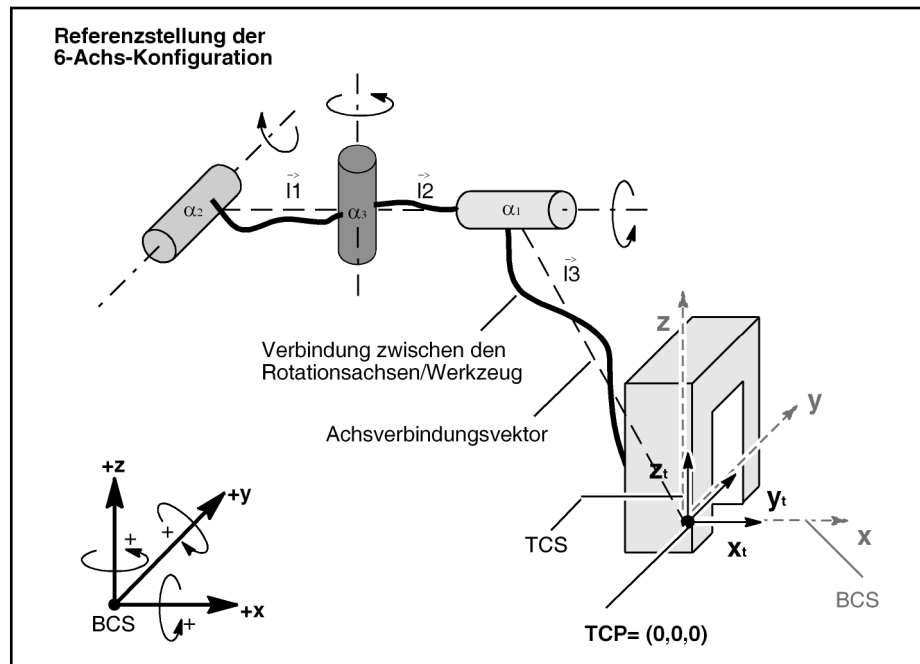


Abb. 11-66: Referenzstellung der 6-Achskonfiguration

Für die Konfigurationsparameter der Achs-Kinematik ist die **Referenzstellung** von Bedeutung.

Die Referenzstellung ist durch folgende Eigenschaften gekennzeichnet:

- Der TCP steht auf der **Raumposition (0,0,0)** im BCS (Nullpunkt des Basis-Werkstückkoordinatensystems).
- Die Koordinaten (x_t , y_t , z_t) des Werkzeugkoordinatensystem (TCS) stehen **parallel** oder **antiparallel** zu den Koordinaten des Basis-Werkstückkoordinatensystems.
- Die Rotationsachsen stehen so, dass ihre Drehachsen entsprechend ihrem jeweiligen Typ um x, y und z des BCS drehen. Sie sind **hintereinander** am Werkzeug angeordnet und stehen **senkrecht** aufeinander. Die Drehrichtung der Rotationsachsen ist mathematisch positiv, wenn man von einer Koordinatenspitze in Richtung Koordinatenursprung schaut und die Drehrichtung im Gegenuhrzeigersinn verläuft.
- Die Linearachsen (X, Y, Z) bilden ein rechtshändiges Achskoordinatensystem, das parallel zum BCS liegt.
- **Achsverbindungsvektoren**

In der Referenzstellung wird die Geometrie zur 6-Achs-Kinematik durch

drei Achsverbindungsvektoren $\vec{l}_1$, $\vec{l}_2$ und $\vec{l}_3$ beschrieben. Die Vektoren können beliebige x-, y- und z-Komponenten bezogen auf das BCS haben.

Der Verbindungsvektor $\vec{l}_i$ ($i=1,2,3$) sieht wie folgt aus:

$$\vec{l}_i = \begin{bmatrix} l_{ix} \\ l_{iy} \\ l_{iz} \end{bmatrix}$$

- $\vec{l}_1$ liegt zwischen der ersten und zweiten Rotationsachse. Die erste Rotationsachse ist diejenige Rundachse, die am weitesten vom Werkzeug (TCP) entfernt ist. Ihr folgt die zweite Rotationsachse, die mit der dritten Rotationsachse über $\vec{l}_2$ verbunden ist. $\vec{l}_3$ verbindet die dritte Rotationsachse mit dem Werkzeug (TCP).
Die l_{ix} -, l_{iy} - und l_{iz} -Komponenten der Achsverbindungsvektoren werden im MP 1030 00140 abgelegt.
Lage der Achsverbindungsvektoren und deren x-, y- und z-Komponenten:

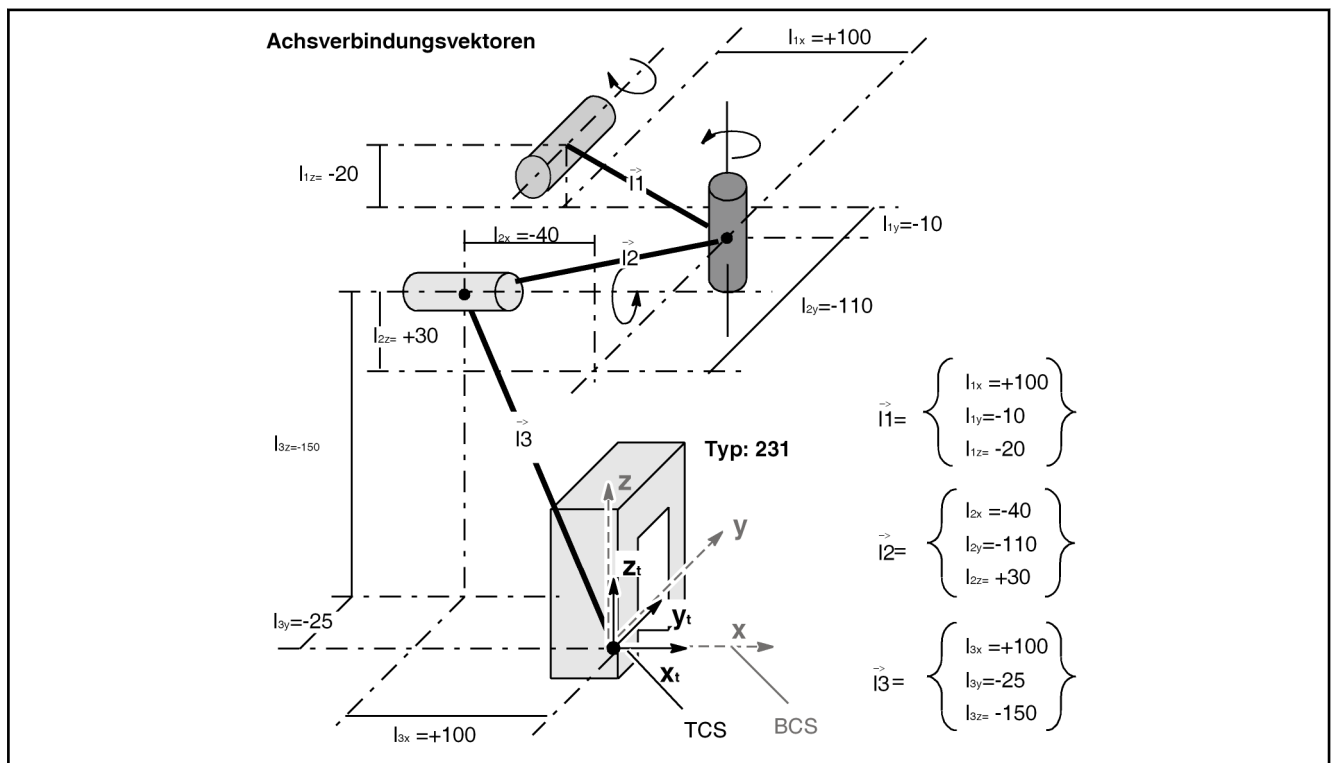


Abb. 11-67: Achsverbindungsvektoren

- Referenzlagen des Werkzeugkoordinatensystems (TCS)
Die **Orientierung** des TCS (x_t , y_t , z_t) bezüglich des BCS (x , y , z) in der Referenzstellung kann mit dem MP 1030 00150 konfiguriert werden.
Folgende Bedingungen gelten für die Referenzlagen des TCS:
 - die Hauptrichtungen (x_t , y_t , z_t) des TCS müssen parallel oder antiparallel zu den Hauptrichtungen (x , y , z) des BCS liegen.
 - Die Hauptrichtungen des TCS und des BCS müssen ein rechtshändiges, rechtwinkliges Koordinatensystem ergeben.
 - Die Voreinstellung des MP 1030 00150 (Referenzlage des Werkzeugkoordinatensystems) definiert ein TCS0, das in der Referenzstellung der Achs-Transformation mit der Orientierung des BCS übereinstimmt:

[1] 1
[2] 2
[3] 3

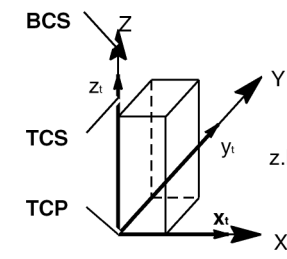
Achs-Transformation

Abweichend von dieser Voreinstellung kann in MP 1030 00150 eine Lage für das **TCS₁** definiert werden, die eine der 24 Referenzlagen (Orientierungen) bezüglich des BCS einnimmt.

Zur besseren **Übersicht** sind die 24 möglichen Orientierungen des TCS₁ wie folgt gegliedert:

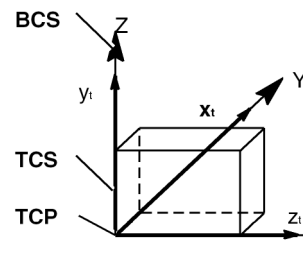
Die x_t -Koordinate des TCS₁ verläuft parallel oder antiparallel zu einer der 3 Hauptrichtungen x, y oder z des BCS. Daraus ergeben sich für **jede** der 3 Hauptrichtungen 8 verschiedene TCS₁-Orientierungen.

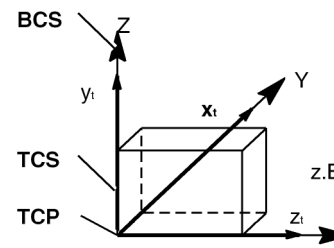
x_t liegt entlang der **x-Koordinate** des BCS:

			Referenzlage des Werkzeugkoordinatensystems (TCS) im MP 1030 00150		
x_t [1] liegt entlang der BCS-Koordinate:	y_t [2] liegt entlang der BCS-Koordinate:	z_t [3] liegt entlang der BCS-Koordinate:	[1]	[2]	[3]
x	y	z	1	2	3
x	z	-y	1	3	-2
x	-z	y	1	-3	2
x	-y	-z	1	-2	-3
-x	z	y	-1	3	2
-x	-y	z	-1	-2	3
-x	y	-z	-1	2	-3
-x	-z	-y	-1	-3	-2

Tab. 11-86: Orientierung des TCS - x-Koordinate

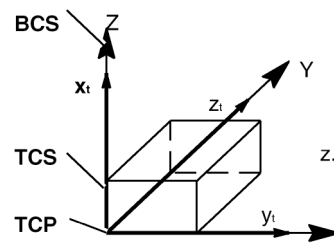
x_t liegt entlang der **y-Koordinate** des BCS:

			Referenzlage des Werkzeugkoordinatensystems (TCS) im MP 1030 00150		
x_t [1] liegt entlang der BCS-Koordinate:	y_t [2] liegt entlang der BCS-Koordinate:	z_t [3] liegt entlang der BCS-Koordinate:	[1]	[2]	[3]
y	z	x	2	3	1
y	x	-z	2	1	-3
y	-x	z	2	-1	3

 z.B. TCS_i-Orientierung: 231 TCS-Haupttrichtung x_t --> y-Koordinate des BCS			Referenzlage des Werkzeugkoordinatensystems (TCS) im MP 1030 00150		
x_t [1] liegt entlang der BCS-Koordinate:	y_t [2] liegt entlang der BCS-Koordinate:	z_t [3] liegt entlang der BCS-Koordinate:	[1]	[2]	[3]
y	-z	-x	2	-3	-1
-y	x	z	-2	1	3
-y	-z	x	-2	-3	1
-y	z	-x	-2	3	-1
-y	-x	-z	-2	-1	-3

Tab. 11-87: Orientierung des TCS - y-Koordinate

 x_t liegt entlang der z-Koordinate des BCS:

 z.B. TCS_i-Orientierung: 312 TCS-Haupttrichtung x_t --> z-Koordinate des BCS			Referenzlage des Werkzeugkoordinatensystems (TCS) im MP 1030 00150		
x_t [1] liegt entlang der BCS-Koordinate:	y_t [2] liegt entlang der BCS-Koordinate:	z_t [3] liegt entlang der BCS-Koordinate:	[1]	[2]	[3]
z	x	y	3	1	2
z	y	-x	3	2	-1
z	-y	x	3	-2	1
z	-x	-y	3	-1	-2
-z	y	x	-3	2	1
-z	-x	y	-3	-1	2
-z	x	-y	-3	1	-2
-z	-y	-x	-3	-2	-1

Tab. 11-88: Orientierung des TCS - z-Koordinate

Besondere Eigenschaften:

- Vorschub:**

Bei aktiver 6-Achs-Transformation bezieht sich der programmierte Vorschub (F) nur noch auf die programmierbaren Positionskoordinaten, d. h. mit dem F-Wort wird die Bahngeschwindigkeit des Werkzeugmittelpunktes (TCP) programmiert.

Zusätzliche Orientierungs- und Pseudokoordinatenbewegungen verändern diese Bahngeschwindigkeit nicht. Die Orientierungs- und die Pseudokoordinatenbewegung werden synchron mitgeführt, d. h., die Endposition wird für alle Koordinaten zeitgleich erreicht.

Die mitgeführte Bewegung der Orientierungs- und Pseudokoordinaten kann jedoch zu einer zusätzlichen Limitierung der Bahnkinematik führen (maximale Bahngeschwindigkeit und -beschleunigung), da die Grenzwerte für alle an der Bewegung beteiligten Achsen überwacht werden.

- Die rotatorischen Achsen A, B und C können sowohl Endlos- als auch Rundachsen sein.

TCS, TCP

TCS bezeichnet das Werkzeugkoordinatensystem.

TCP₀ / TCP₁ bezeichnet den Tool Center Point (Werkzeugmittelpunkt) **ohne/ mit** Korrektur.

Werkzeugkorrektur

Werkzeugkorrektur beim 6-Achs-Transformationstyp 3333301:

Durch Werkzeugkorrekturen kann das TCS₁ sowohl verschoben als auch gedreht werden. Es entsteht dadurch ein Werkzeugkoordinatensystem TCS_c:

- Die Verschiebung (Korrekturvektor) des TCS_c bzgl. TCS₁ wird durch

$$\vec{l}_t$$

definiert.

- Die Orientierung des TCS_c im Koordinatensystem TCS₁ wird durch die

Drehmatrix $\vec{T}_t$ oder äquivalent dazu durch die drei Eulerwinkel φ , ϑ , ψ definiert. Der Korrekturvektor und die Eulerwinkel können z. B. mittels des CPL-Befehls "TC" gesetzt werden.

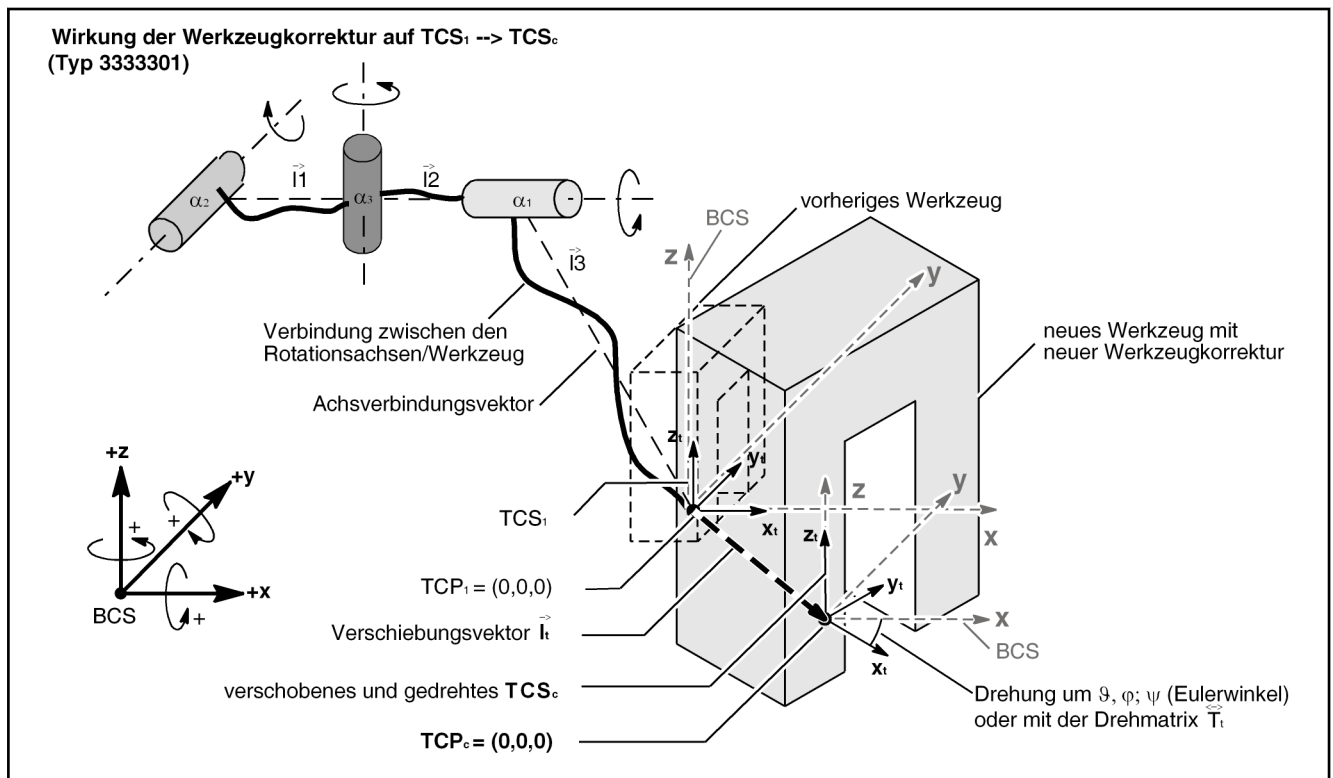


Abb. 11-68: Wirkung der Werkzeugkorrektur

Die Werkzeugkorrektur hat hier die gleiche Wirkung, wie das Ersetzen des Konfigurationsvektors $\vec{l}_3$ durch $\vec{l}_t + \vec{l}_3$.

Werkzeugkorrektur beim 6-Achs-Transformationstyp 3033001:

Durch Werkzeugkorrekturen wird das TCS_1 beliebig räumlich verschoben (hier ist immer $TCS_1 = TCS_0$). Es entsteht dadurch ein Werkzeugkoordinatensystem TCS_c :

Die Verschiebung (Korrekturvektor) des TCS_c bzgl. TCS_0 oder TCS_1 wird durch $\vec{t}_i$ definiert.



Eine Drehung des TCS wie beim Typ 3333301 ist **nicht** möglich.

Achs-Transformation

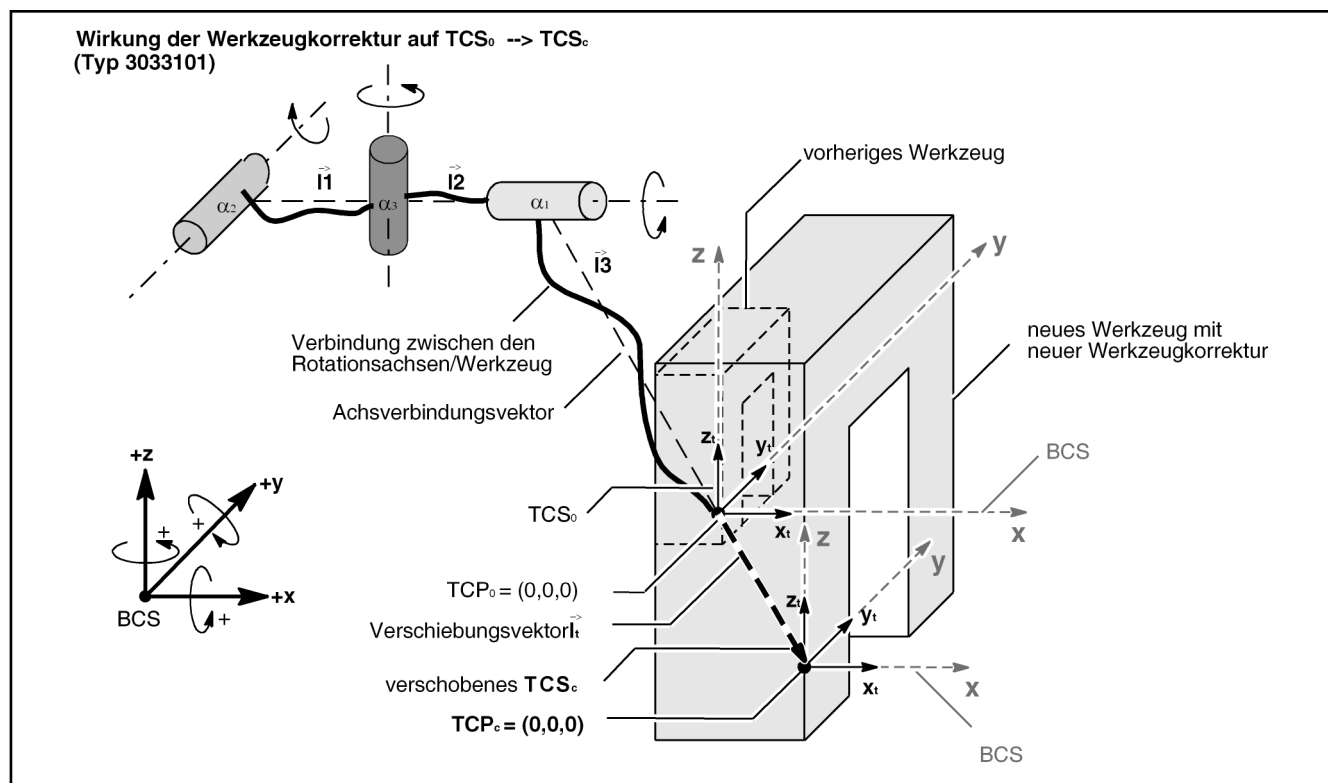


Abb. 11-69: Wirkung der Werkzeugkorrektur

Die Werkzeugkorrektur hat hier die gleiche Wirkung wie das Ersetzen des Konfigurationsvektors $\vec{l}_3$ durch $\vec{l}_t + \vec{l}_3$.

11.18.5 Handlungsanweisung: 6-Achs-Transformation LLLRRR

Applizieren Die Achstransformation kann über die relevanten Maschinenparameter appliziert werden. Dazu muss die Funktion Achstransformation über den Parameter **tra** "Achstransformationen" im Setup (SUP) aktiviert werden. Der Pfad des MP-Namens ist immer "TRA/AxTrafo[1..20]/...".

Alternativ dazu kann die Achstransformation über ein benutzerdefiniertes Systemdatum vom Typ SysAxTrafo_t appliziert werden.

Die beiden Varianten sind inhaltlich identisch.

Die Achspositionen der Referenzstellung müssen ermittelt werden.

In der nachfolgenden Tabelle sind jeweils die äquivalenten MP-Nummern, MP-Name und SD-Elemente angegeben. Um eine Transformation zu applizieren, muss eine der Varianten entsprechend besetzt werden.

Transformations-relevante Parameter

MP-Nummer	Beschreibung
1030 00110	Transformationstyp MP-Name: TrafoType SD-Elem. : Type
3333301	LLLRRR mit Tensor-Orientierung
3033001	LLLRRR mit Linear-Bewegung der rotatorischen Achsen

MP-Nummer	Beschreibung	
1030 00120	Systemachsen/-koordinaten der Transformation	
	MP-Name: FwdInCoordIndTrafo[1..8]	
	SD-Elem. : AxisAssignment[1..8]	
	[1]	Systemachsnummer von X
	[2]	Systemachsnummer von Y
	[3]	Systemachsnummer von Z
	[4]	Systemachsnummer von R_1 (am weitesten vom Werkzeug (TCP) entfernt)
	[5]	Systemachsnummer von R_2
	[6]	Systemachsnummer von R_3 (am Werkzeug)
	[7..8]	nicht relevant

MP-Nummer	Beschreibung		
1030 00125	Achsklassifikationen der Transformationsachsen		
	MP-Name: CoordClass[1..8]		
	MP-Name: CoordDir[1..8]		
	SD-Elem. : AxisClassification[1..8]		
	Definiert die an der Transformation beteiligten Achsen inklusive Bewegungsrichtung.		
	- MP-Nr. oder SD-Element: Wert ">0" bedeutet: Die Achse verfährt/dreht positiv im Sinne der obigen Abbildungen. Wert "<0" bedeutet: Die Achse verfährt/dreht negativ im Sinne der obigen Abbildungen. Mögliche Werte sind: +/-1 = X-Achse +/-2 = Y-Achse +/-3 = Z-Achse +/-100 = A-Achse +/-200 = B-Achse +/-300 = C-Achse - MP-Name: - Mögliche Werte für CoordClass sind - X,Y,Z (linear) - A,B,C (rotatorisch) - Mögliche Werte für CoordDir sind: - positiv (bzgl. Abb. "Referenzstellung") - negativ (bzgl. Abb. "Referenzstellung")		
	Ist ein Parameter unbesetzt, gelten folgende Default-Einstellungen:		
		MP-Nr / SD-Elem.:	CoordClass: CoordDir:
	[1]	1	X positiv
	[2]	2	Y positiv
	[3]	3	Z positiv
	[4]	200	B positiv
	[5]	100	A positiv
	[6]	300	C positiv
	[7..8]	nicht relevant	0 positiv

MP-Nummer	Beschreibung
1030 00130	Achsenpositionen der Referenzstellung MP-Name: RefPosTrafo[1..8] SD-Elem. : AxZeroPos[1..8]
	[1] X^0
	[2] Y^0
	[3] Z^0
	[4] R_1^0
	[5] R_2^0
	[6] R_3^0
	[7..8] nicht relevant
1030 00140	Längen- und Winkelparameter MP-Name: JointParTrafo[1..16] SD-Elem. : LenParam[1..16] Definiert die Längenvektoren l_1, l_2, l_3 in [mm].
	[1] l_{1x}
	[2] l_{1y}
	[3] l_{1z}
	[4] l_{2x}
	[5] l_{2y}
	[6] l_{2z}
	[7] l_{3x}
	[8] l_{3y}
	[9] l_{3z}
	[10..16] nicht relevant
1030 00150	Referenzlage des Werkzeugkoordinatensystems Definiert die Orientierung des TCS ₁ bezüglich des BCS in der Referenzstellung. MP-Name: CoordTcsXBcs, CoordTcsYBcs, CoordTcsZBcs MP-Name: DirTcsXBcs, DirTcsYBcs, DirTcsZBcs SD-Elem. : TCSOrientation[1..8]
	[1] Richtung von x_t
	[2] Richtung von y_t
	[3] Richtung von z_t
	[4..8] nicht relevant
	Werte: $\pm 1, \pm 2, \pm 3$. Die Orientierung des TCS muss ein rechtshändiges Koordinatensystem ergeben.
	Der Parameter 103000150 ist nicht relevant für den Typ 3033001!

MP-Nummer	Beschreibung
1030 00160	Epsilon-Umgebungen MP-Name: EpsAxTrafo SD-Elem. : EpsilonRanges 0.01

Tab. 11-89: Transformations-relevante Parameter

Beispiel:**Applizieren der 6-Achs-Transformation**

Im folgenden Beispiel werden die Transformationstypen 3333301 und 3033001 behandelt.



Alle einzugebenden Werte beziehen sich auf die Referenzstellung.

- Legen Sie in MP 1030 00110 (Nr. 1) als Transformationstyp den Wert 3333301 ab, in MP 1030 00110 (Nr. 2) den Wert 3033001.
- In MP 103000120 (Nr. 1 und Nr. 2) sind die Systemachsnummern einzutragen.
Achten Sie darauf, dass die 1. Rundachse am weitesten vom Werkzeug (TCP) entfernt liegt und die 3. Rundachse die kürzeste Verbindung zum TCP hat.
- Die Komponenten der Vektoren $\vec{l}_1$, $\vec{l}_2$ und $\vec{l}_3$ sind geeignet zu ermitteln. Dies kann durch Vermessen oder aus Konstruktionszeichnungen direkt ermittelt werden. Die Längen- und Versatzwerte sind in MP 1030 00140[1..9] sowohl für Nr. 1 als auch für Nr. 2 abzulegen.
- Achsnullstellung ermitteln: Fahren Sie die Referenzstellung an. Die Achspositionen in der Referenzstellung werden im Parameter 1030 00130[1..5] sowohl für Nr. 1 als auch für Nr. 2 abgelegt.
- Die Referenzlage des TCS wird im Parameter 1030 00150 für Nr. 1 eingetragen. für Nr. 2 (Typ = 3033001) ist keine TCS-Definition notwendig.
Es ist darauf zu achten, dass die Werte ein rechtshändiges Koordinatensystem ergeben. Ein linkshändiges Koordinatensystem erzeugt einen Laufzeitfehler.
- Parameter mit Systemgrundstellung übernehmen.
- Testen Sie die 6-Achs-Transformation in einem NC-Programm mit COORD(1) und COORD(2) auf Funktionalität. Erfolgt kein Laufzeitfehler, sind die Parameter in sich konsistent.

Aktivieren Ist die 6-Achs-Transformation konfiguriert per

- Maschinenparameter, dann wird die Transformation im NC-Programm aktiviert mit (i=1,..20):
COORD(<i>)
- Systemdatum, dann wird die Transformation im NC-Programm aktiviert mit (Typ "SysAxTrafo_t"):
COORD(<SD.MtbAxTrafo>)

Programmbeispiel:

Die 6-Achs-Transformation ist im 4. Maschinenparametersatz definiert. Am Programmanfang ist keine Achstransformation aktiv:

Transformation aktivieren

```

N100 G1 F1000 XA10 YA10 C30 ;Axes XA, YA and C are traversed
N110 COORD(4) ;Switching on the axis transformation
N130 x20 y30 phi0 psi10 ;the coordinates x,y,phi,psi are traversed
N140 x10 z10 theta45 ;the coordinates x,z,theta are traversed
      :
N999 COORD(0) ;Switching-off the axis transformation
                        ;Axes are again programmed from here

```

Deaktivieren **COORD(0)** bzw. **COORD(0,2)**
 schaltet die Achstransformation im NC-Programm aus.

11.19 3-Achs-Zylindermantel-Transformation

11.19.1 Beschreibung

Allgemeines

Die Zylindermantel-Transformationen wird für die Programmierung der Bearbeitungen auf der Mantelfläche eines zylindrischen Werkstücks verwendet.

Dazu existieren 2 Varianten:

- die programmierte Kontur bezieht sich auf die abgewinkelte Zylindermantelfläche sowie
- die programmierte Kontur bezieht sich auf die Ebenenprojektion auf die Zylindermantelfläche.

11.19.2 Variante 1 (Abwicklung)

Funktion

Name der Transformation	Transformationstyp	Wirkstelle	Koordinatensystem
Zylindermantel-Transformation (Abwicklung)	3021001 (RLL)	AT2	MCS, BCS

Tab. 11-90: Funktion - Variante 1 (Abwicklung)

Ausgangsgrößen im MCS: X,Z,C

Eingangsgrößen im BCS: x, y, z

Die Zylindermantel-Transformation 3021001 wirkt an der Wirkstelle AT2 und transformiert zwischen dem Maschinenkoordinatensystem MCS = (X,Z,C) und dem Basiskoordinatensystem BCS = (x,y,z).

Die Maschinen-Konfiguration besteht aus zwei linearen MCS-Koordinaten X, Z, die das Fräswerkzeug positionieren, sowie einer Rundachse C, auf die der zu bearbeitende Zylinder rotationssymmetrisch aufgespannt ist.

Die C-Achse kann eine Endlos- oder auch Rundachse sein.

Achs-Transformation

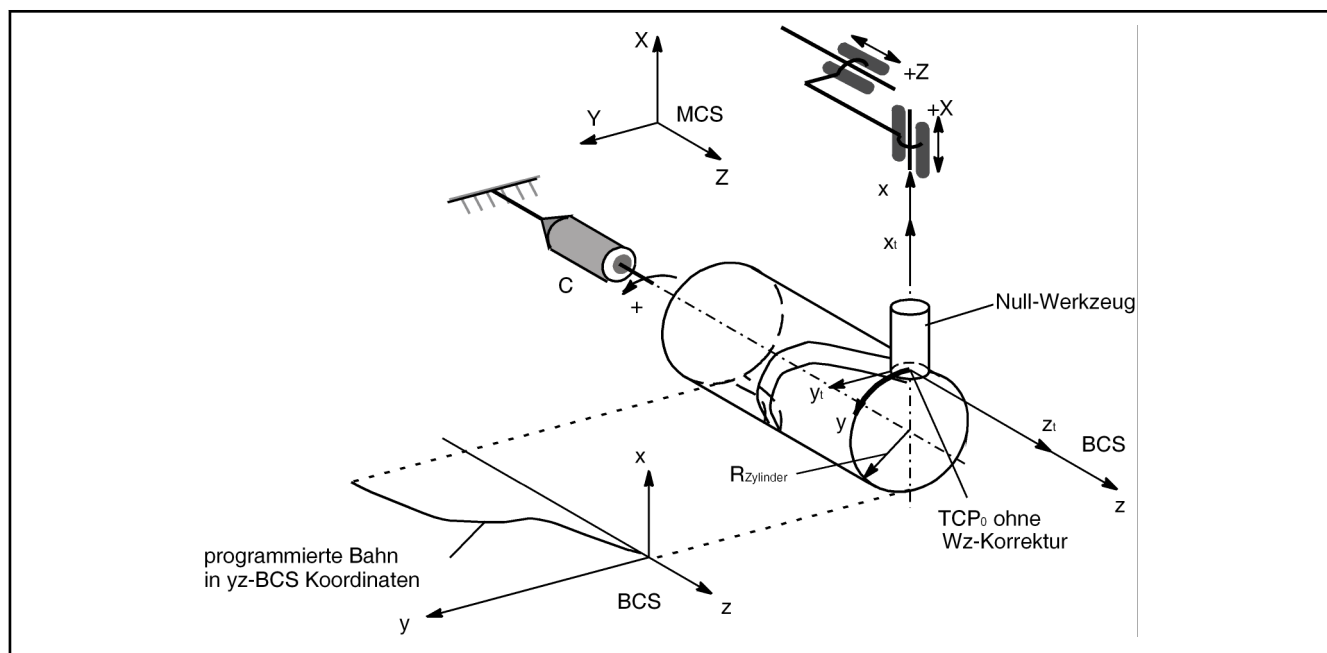


Abb. 11-70: Zylindermantel-Transformation (Abwicklung)

Die Eigenschaft der Transformation liegt in der Konturbeschreibung der y-Koordinate:

- Die Programmierung der y-Koordinate führt mit der z-Koordinate zu einer Kontur, die man als Abwicklung auf einer Zylindermantelfläche bezeichnen kann.
- die y-Koordinate läuft auf der Manteloberfläche über den Umfang des Zylinders.
- Die y,z-Kontur wird bezogen auf die abgewickelte Zylindermantelfläche programmiert. Der Wertebereich der y-Achse ist unbeschränkt:

Beispiel:

Spiralförmige Nuten

Die Winkellage $C = 20^\circ$ und $C = 380^\circ$ beschreiben unterschiedliche Positionen.

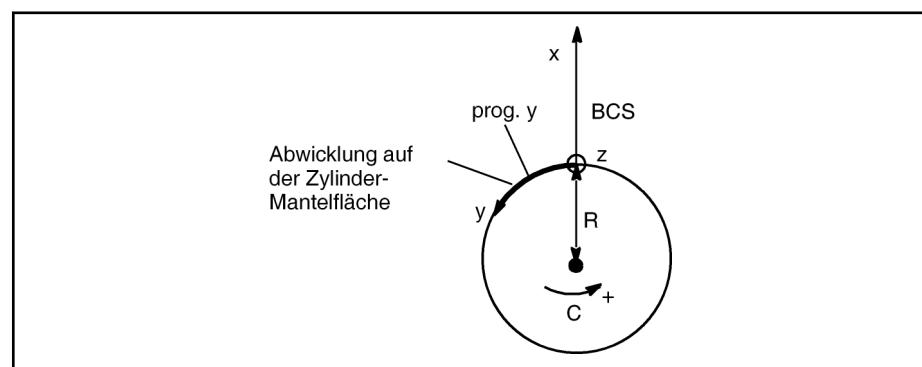


Abb. 11-71: Beispiel - Spiralförmige Nuten

11.19.3 Variante 2 (Ebenenprojektion)

Funktion

Name der Transformation	Transformationstyp	Wirkstelle	Koordinatensystem
Zylindermantel-Transformation (Ebenenprojektion)	3021002 (RLL)	AT2	MCS, BCS

Tab. 11-91: Funktion - Variante 2 (Ebenenprojektion)

Ausgangsgrößen im MCS: X,Z,C

Eingangsgrößen im BCS: x, y, z

Die Zylindermantel-Transformation 3021002 wirkt an der Wirkstelle AT2 und transformiert zwischen dem Maschinenkoordinatensystem MCS = (X,Z,C) und dem Basiskoordinatensystem BCS = (x,y,z).

Die Eigenschaft dieser Transformationen liegen in der Konturbeschreibung der y-Koordinate:

- Die Programmierung der y-Koordinate fährt im Gegensatz zur Zylindermantel-Transformation 3021001 mit Abwicklung zu einer Kontur auf einer Zylindermantelfläche, die durch **Ebenenprojektion** zustande kommt. Die Programmierung einer Kontur erfolgt in einer Ebene, die z. B. durch die "Schiefe Ebene"-Funktion beliebig im Raum orientiert sein kann. Die Abbildung (Projektion) dieser Kontur auf der Mantelfläche ergibt die tatsächliche Kontur am Werkstück.
- Die (y,z)-Kontur wird bezogen auf eine (beliebige) Ebenenprojektion programmiert (z. B. Rohr in Rohr).

Beispiel:

Projektion auf das Basiskoordinatensystem BCS

; d. h. es ist kein Placement aktiv (BCS = WCS).

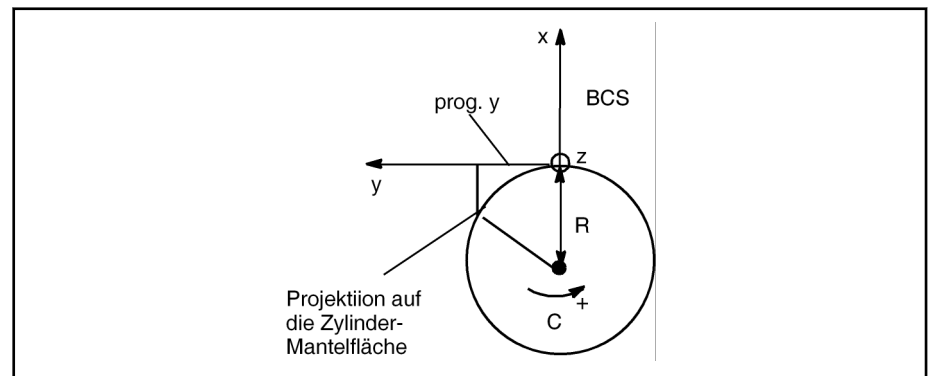


Abb. 11-72: Beispiel Projektion auf das Basiskoordinatensystem BCS

Die Rückwärtstransformation ist für $|y| < R$ definiert. Bei $|y| \geq R$ wird ein Laufzeitfehler generiert. Dieser Bereich ist für die Praxis irrelevant.

Beispiel:

Projektion auf das BCS

Die folgende Abbildung zeigt den Fall der Projektion auf das BCS, wenn die Funktion Placement (Schiefe Ebene) aktiv ist (BCS nicht identisch mit WCS).

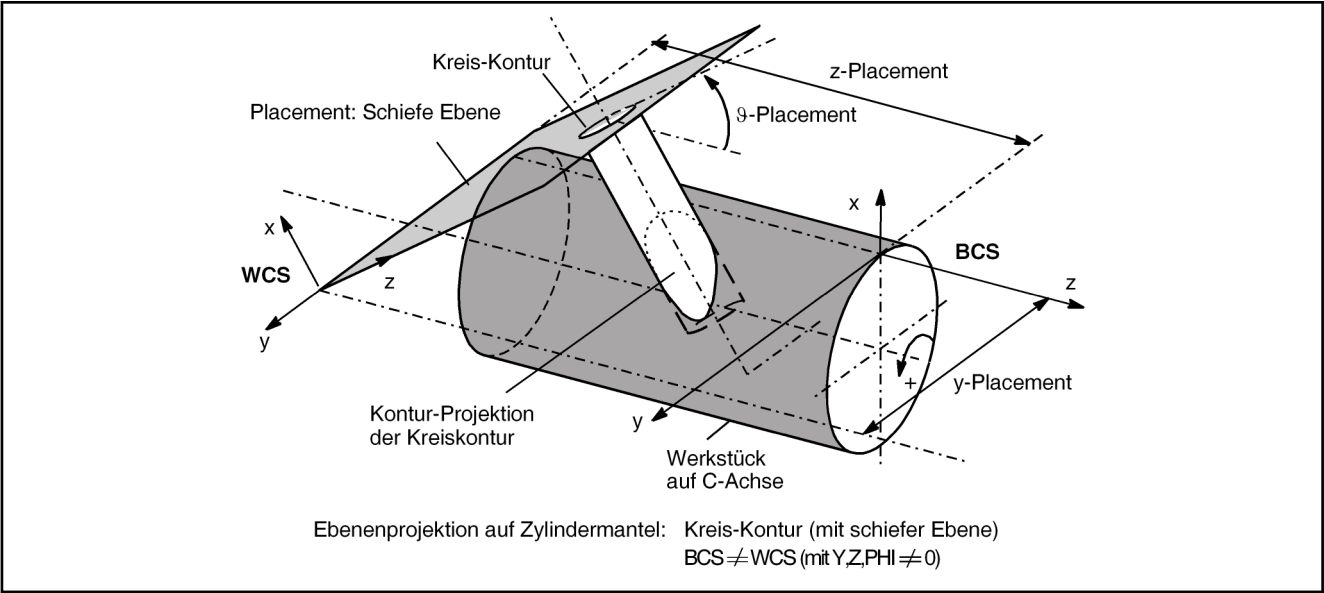


Abb. 11-73: Ebenenprojektion auf Zylindermantel

11.19.4 Gemeinsame Eigenschaften der Varianten 1, 2

Relevante NC-Funktionen

COORD (<I>) TRFOPT(<RADIUS>)	
	mit I=1..20. Aktivierung der in MP 1030 00110 (Achs-Transformationstyp) definierten Zylindermantel-Transformation. Mit TRFOPT kann der Zylinderradius (RADIUS > 0) mit übergeben werden. Er überschreibt den Wert des MP 1030 00140 [1].
Nur für Ebenenprojektion 3021002:	
G152.x	Placement (Schiefe Ebene) aktivieren und damit Festlegung der Projektionsebene. Placement muss nach dem Aktivieren der Transformation eingeschaltet werden. Hinweis: Im Zusammenhang mit der Zylindermantel-Transformation hat die Placement-Funktion "Schiefe Ebene " G152.x kein "klassisches" Placementverhalten: G152.x definiert die (Y,Z)-Projektionsebene grundsätzlich mit "X Placement) = 0". X-Werte ungleich Null haben keine Wirkung, da die X Placement)-Komponente steuerungsintern zur Ermittlung des zugehörigen Punktes auf der Manteloberfläche benötigt wird.
G153	Ausschalten Placement (Schiefe Ebene), d. h. BCS = WCS

Tab. 11-92: Relevante NC-Funktionen

Relevante Maschinenparameter

Die Maschinenparameter sind für beide Varianten im Wesentlichen identisch.

1030 00110	Achs-Transformationstyp
1030 00120	Systemachsen der Transformation.
1030 00125	Achsklassifikation der Transformationsachsen
1030 00130	Achspositionen der Referenzstellung

Tab. 11-93: Relevante Maschinenparameter

Referenzstellung

In der Referenzstellung befindet sich die Werkzeugaufnahme in der Rotationsachse C. Der BCS-Nullpunkt liegt dann auf der Manteloberfläche bei $x = y = z = 0$.

Steht die Werkzeugspitze (incl. Längenkorrektur) auf diesem Punkt, entspricht dies der zugehörigen MCS-Koordinaten

$$(X, Z, C) = (\text{Radius}_{\text{Zylinder}} + l_{tx} + X_0, Z_0, C_0).$$

Das heißt, die Referenzstellung für den Werkzeugaufnahmepunkt (TCP_0) ist vom BCS-Nullpunkt um den Radius R des Zylinders in negative x-Richtung verschoben.

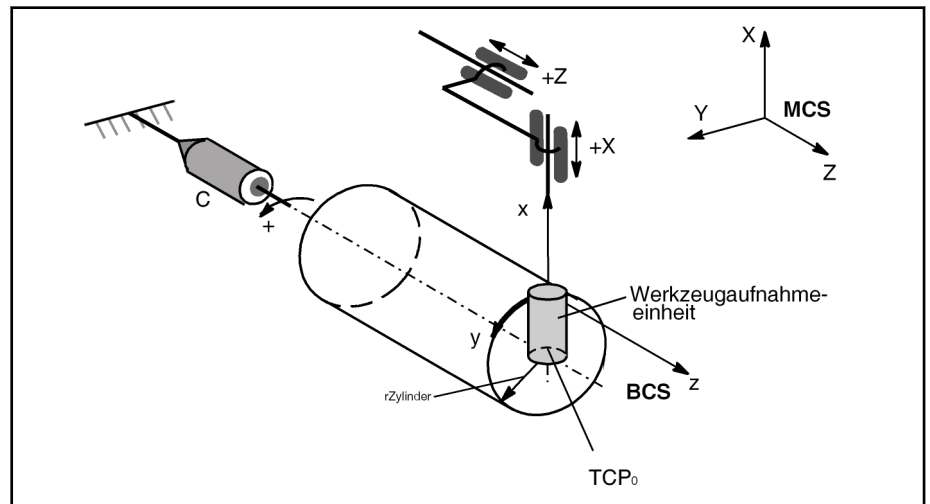


Abb. 11-74: Referenzstellung

TCS, TCP

TCS bezeichnet das Werkzeugkoordinatensystem.

TCP_0 / TCP_1 bezeichnet den Tool Center Point (Werkzeugmittelpunkt) **ohne/ mit** Korrektur.

Werkzeugkorrektur

Steht der Werkzeugaufnahmepunkt TCP_0 in der Referenzposition ($x = y = z = 0$ im BCS), befindet sich der Tool Center Point (TCP) mit aktiver WZ-Längenkorrektur auf der Position

$$(x, y, z) = (-\text{Radius}_{\text{Zylinder}} - l_{tx}, 0, 0)$$

bezogen auf den BCS-Nullpunkt (siehe obige Abbildung).

Die WZ-Längenkorrektur darf bei der Zylindermantel-Transformation nur in Richtung x-Koordinate wirken. Der WZ-Längenkorrekturvektor $\vec{l}_{tx}$ zeigt also von der Werkzeugspitze des TCS zur Werkzeugaufnahme des TCS₀.

Transformations-relevante Parameter

MP-Nummer	Beschreibung	
1030 00110	Transformationstyp	
	MP-Name: TrafoType	
	SD-Elem. : Type	
	3021001	Abwicklung
	3021002	Projektion
1030 00120	Systemachsen/-koordinaten der Transformation	
	MP-Name: FwdInCoordIndTrafo[1..8]	
	SD-Elem. : AxisAssignment[1..8]	
	[1]	Systemachsnummer von X
	[2]	Systemachsnummer von C
	[3]	Systemachsnummer von Z
	[4..8]	nicht relevant

Achs-Transformation

MP-Nummer	Beschreibung		
1030 00125	Achsklassifikationen der Transformationsachsen		
	MP-Name: CoordClass[1..8]		
	MP-Name: CoordDir[1..8]		
	SD-Elem. : AxisClassification[1..8]		
	Definiert die an der Transformation beteiligten Achsen inklusive Bewegungsrichtung. - MP-Nr. oder SD-Element: Wert ">0" bedeutet: Die Achse verfährt/dreht positiv im Sinne der obigen Abbildungen. Wert "<0" bedeutet: Die Achse verfährt/dreht negativ im Sinne der obigen Abbildungen. Mögliche Werte sind: +/-1 = X-Achse +/-300 = C-Achse - MP-Name: - Mögliche Werte für CoordClass sind - X (linear) - C (rotatorisch) - Mögliche Werte für CoordDir sind: - positiv (bzgl. Abb. "Referenzstellung") - negativ (bzgl. Abb. "Referenzstellung")		
	Ist ein Parameter unbesetzt, gelten folgende Default-Einstellungen:		
		MP-Nr / SD-Elem.:	CoordClass: CoordDir:
	[1]	1	X positiv
	[2]	300	C positiv
	[3..8]	nicht relevant	0 positiv
1030 00130	Achspositionen der Referenzstellung		
	MP-Name: RefPosTrafo[1..8]		
	SD-Elem. : AxZeroPos[1..8]		
	[1]	X ⁰	
	[2]	C ⁰	
	[3]	Z ⁰	
	[6..8]	nicht relevant	

MP-Nummer	Beschreibung	
1030 00140	Längen- und Winkelparameter	
	MP-Name: JointParTrafo[1..16]	
	SD-Elem. : LenParam[1..16]	
	Definiert die Längenvektoren l_1, l_2 in [mm].	
	[1]	Radius Default-Wert für den Zylinderradius. Wird beim Aktivieren der Achstransformation TRFOPT programmiert, wird dieser Wert damit überschrieben.
	[2]	yMod Nur relevant für die Variante 1, wenn es sich bei der C-Achse um eine Endlos-Modulo-Achse handelt.
		0 der Einschaltwert der y-Koordinate liegt im Intervall $[-2\pi \cdot \text{Radius}, 0]$
		1 der Einschaltwert der y-Koordinate liegt im Intervall $[-\pi \cdot \text{Radius}, \pi \cdot \text{Radius}]$ (empfohlen).
	[3..16]	nicht relevant

Tab. 11-94: Transformations-relevante Parameter

Aktivieren

Ist die Zylindermantel-Transformation konfiguriert per

- Maschinenparameter, dann wird die Transformation im NC-Programm aktiviert mit (i=1,..20):

COORD(<i>) TRFOPT(<Zylinderradius>)

- Systemdatum, dann wird die Transformation im NC-Programm aktiviert mit (Typ "SysAxTrafo_t"):

COORD(<SD.MtbAxTrafo>) TRFOPT(<Zylinderradius>)

Programmbeispiel:

Die Zylindermantel-Transformation ist im 4. Maschinenparametersatz definiert. Am Programmanfang ist keine Achstransformation aktiv:

Transformation aktivieren

```

N100 G1 F1000 XA10 ZA10 C30 ;the axes XA,ZA,C are traversed
N110 COORD(4) TRFOPT(100) ;switching-on axis transformation with a radius of 100
N130 x20 y30 ;traversing the coordinates x,y
N140 x10 z5 ;the coordinates x,z are traversed
:
N999 COORD(0) ;Switching-off the axis transformation
;Axes are again programmed from here

```



Der Parameter TRFOPT(<Zylinderradius>) muss mit einem Zylinderradius-Wert > 0 programmiert werden.

Per Default ist der Wert Null unterlegt, dieser führt aber zu einem Laufzeitfehler.



Projektion

- **G152.x** Placement (Schiefe Ebene) aktivieren, damit wird die Projektionsebene festgelegt. Das **Placement muss nach** dem Aktivieren der Transformation eingeschaltet werden. Mit G153 wird das Placement (Schiefe Ebene) ausgeschaltet.
- Bei der Zylindermantel-Transformation mit Ebenenprojektion ist darauf zu achten, dass die Transformation nur für y-Werte funktioniert, die im Durchmesserbereich des Zylinders liegen, d.h. $-\text{Zylinderradius} < y < \text{Zylinderradius}$.

Deaktivieren

COORD(0) bzw. **COORD(0,2)**

schaltet die Achstransformation im NC-Programm aus.

11.20 4-Achs-Zylindermantel-Transformation

11.20.1 Beschreibung

Funktion

Name der Transformation	Transformationstyp	Wirkstelle
4-Achs-Zylindermantel		
Abwicklung	3131021	AT2
Projektion	3131022	AT2

Die 4-Achs-Zylindermantel-Transformationen werden für die Programmierung der Bearbeitungen auf der Mantelfläche eines zylindrischen Werkstücks verwendet. Sie ermöglichen Nutwandkorrekturen (z. B. parallele Nutwände), siehe dazu folgende Abbildung.

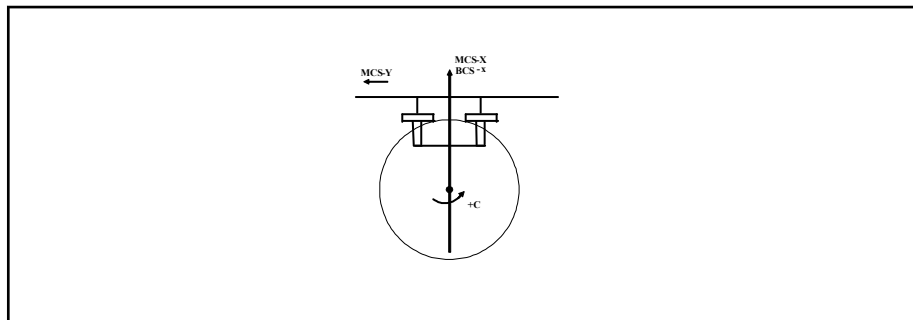


Abb. 11-76: Parallele Nutwände

Es existieren grundsätzlich zwei Varianten bezüglich der y-Koordinate:

- die programmierte Kontur bezieht sich auf die abgewinkelte Zylindermantelfläche (Standard)
- die programmierte Kontur bezieht sich auf die Ebenenprojektion auf die Zylindermantelfläche.

Die Maschinenkonfiguration besteht aus drei linearen MCS-Koordinaten X, Y, Z, welche ein rechtshändiges kartesisches Koordinatensystem aufspannen. Das Werkzeug ist an der X-Achse angebracht. Auf der Rundachse C ist der zu bearbeitende Zylinder rotationssymmetrisch angebracht. Die Z-Achse läuft parallel zur Längsrichtung des Zylinders (siehe [Abb. 11-77 "TCP in BCS-Nullposition" auf Seite 363](#) - [Abb. 11-80 "Position und Orientierung phi-Abwicklung" auf Seite 364](#)).

Der BCS-Nullpunkt liegt auf der Manteloberfläche (siehe Abb. 11-77 "TCP in BCS-Nullposition" auf Seite 363). Die (negative) x-Koordinate beschreibt die Eintauchtiefe.

In beiden Varianten (Abwicklung und Projektion) wird die y-Bewegung von der Rundachse C erzeugt (siehe Abb. 11-78 "Orientierung phi" auf Seite 363). Die lineare Y-Achse erzeugt am TCP die Orientierung phi (siehe Abb. 11-78 "Orientierung phi" auf Seite 363).

Die beiden 4-Achs-Zylindermantel-Transformationen kommen an der Wirkstelle 2 zum Einsatz, d. h. sie transformieren zwischen dem Basiskoordinatensystem (BCS) mit den Koordinaten x, y, z , phi und dem Maschinenkoordinatensystem (MCS) mit den Koordinaten X, Y, Z, C .

- Programmiert werden die BCS-Eingangsgrößen x, y, z, phi
- Berechnet werden die MCS-Ausgangsgrößen X, Y, Z, C

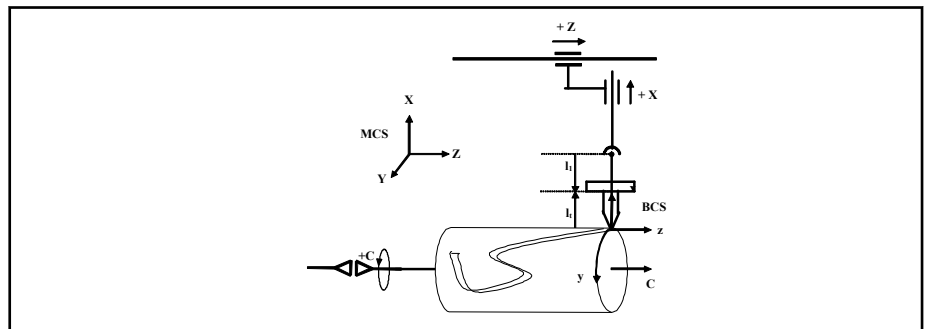


Abb. 11-77: TCP in BCS-Nullposition

Der Werkzeugkorrekturvektor zeigt vom TCP zur Werkzeugaufnahme.

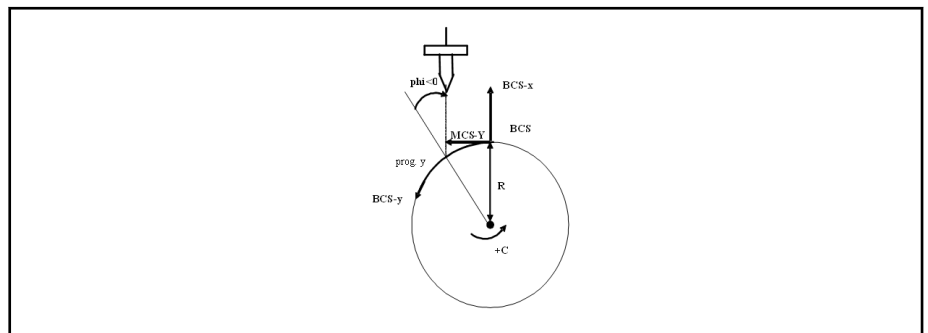


Abb. 11-78: Orientierung phi

Die Referenzposition der Z-Achse liegt in der C-Achse, vergl. folgende Abbildung.

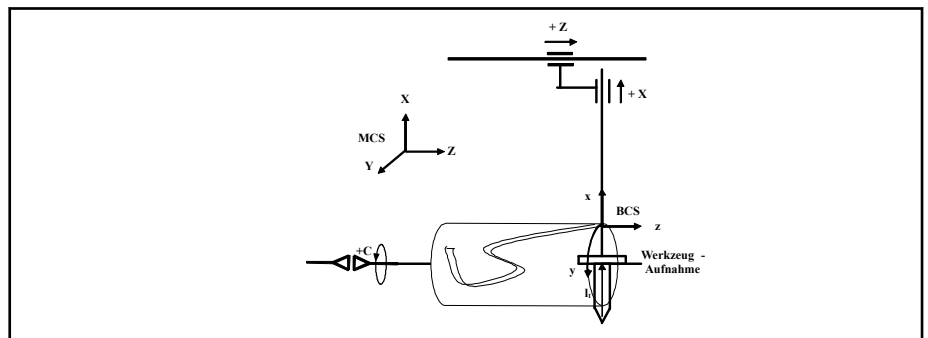


Abb. 11-79: Referenzposition Z-Achse

Achs-Transformation

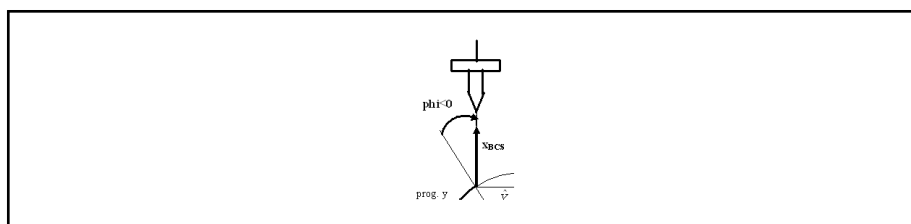


Abb. 11-80: Position und Orientierung phi-Abwicklung

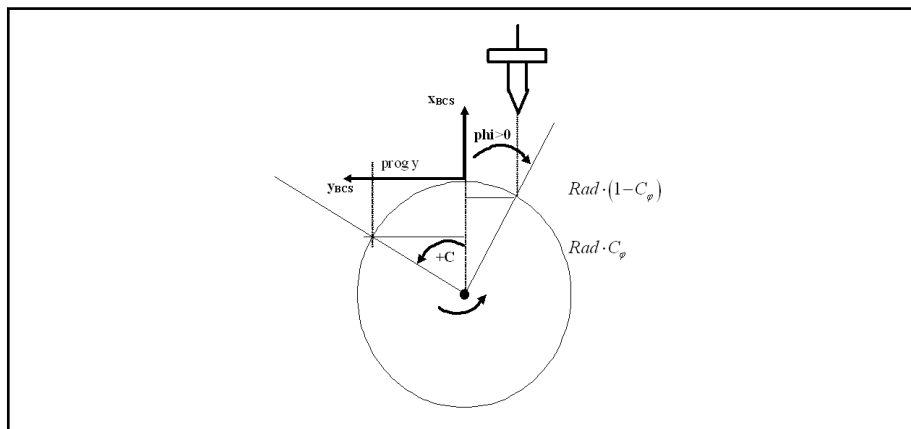


Abb. 11-81: Position und Orientierung phi-Projektion

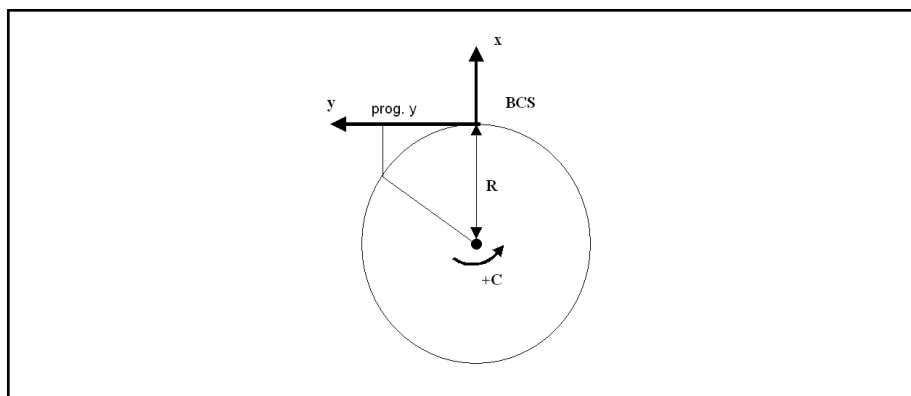


Abb. 11-82: Projektion

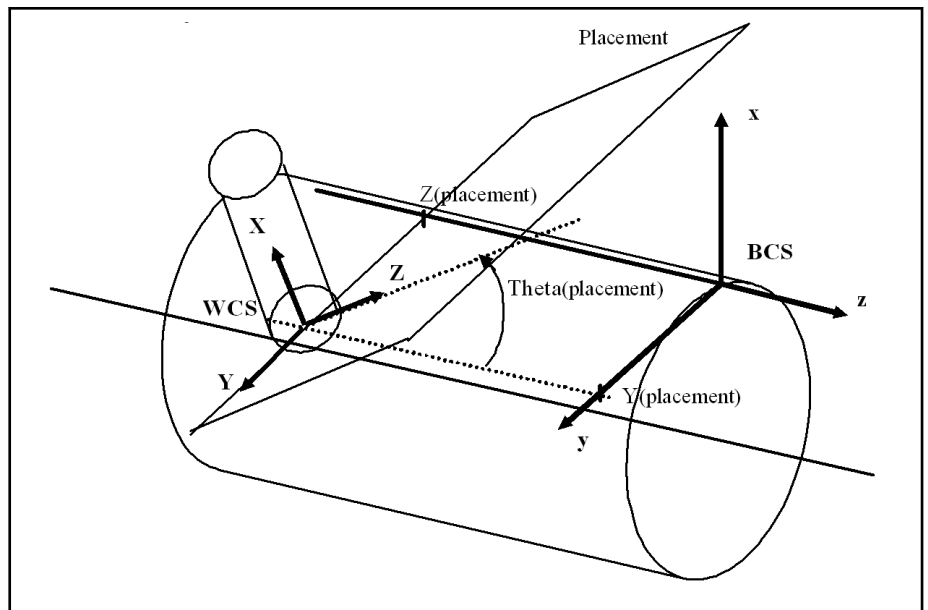


Abb. 11-83: Projektion

Referenzstellung

In der Referenzstellung befindet sich die Werkzeugaufnahme (TCP₀) in der Rotationsachse C ([Abb. 11-80 "Position und Orientierung phi-Abwicklung" auf Seite 364](#)). Der BCS-Nullpunkt liegt dann auf der Manteloberfläche mit (x,y,z)=(0,0,0) ([Abb. 11-77 "TCP in BCS-Nullposition" auf Seite 363](#)).

Steht die Werkzeugspitze (inkl. Längenkorrektur) auf diesem Punkt, entspricht dies den zugehörigen MCS-Koordinaten $(X, C, Z, Y, B) = (\text{RadiusZylinder} + \text{MTX} + X_0, C_0, Z_0, Y_0, B_0)$. D. h., die Referenzstellung der Werkzeugaufnahme ist vom BCS-Nullpunkt um den Radius R des Zylinders in negative x-Richtung verschoben (vergl. Abb. 11-77 "TCP in BCS-Nullposition" auf Seite 363, Abb. 11-80 "Position und Orientierung phi-Abwicklung" auf Seite 364).

Die Werkzeugorientierung ist entlang X ausgerichtet (d. h. $\phi=0$).

Varianten

Es gibt zwei verschiedene Varianten der 4-Achs-Zylindermanteltransformation.

Diese unterscheiden sich durch unterschiedliche y -Koordinaten:

1. die programmierte Kontur bezieht sich auf die abgewinkelte Zylindermantelfläche (Standard)
2. die programmierte Kontur bezieht sich auf eine Ebenenprojektion auf die Zylindermantelfläche

Die positiven Dreh-/Verfahr-Richtungen der Achsen sind in den Abbildungen dargestellt. Diese sind entsprechend einzustellen!

Orientierung

Mit Einschalten der 4-Achs-Zylindermantel-Transformation wird die lineare Orientierungsbewegung aktiviert. Der Winkel phi wird linear vom Anfangs- zum Endpunkt interpoliert.

Besonderheiten

- Die C-Achse kann als Endlos- oder als Rundachse betrieben werden.

- **Werkzeugkorrektur**

Beide Achstransformationen unterstützen eine Werkzeugkorrektur. Der Werkzeugkorrekturvektor $l_t = (l_{tx}, 0, 0)$ zeigt vom korrigierten Werkzeugmittelpunkt TCP_1 zum unkorrigierten TCP_0 . Nur die Längenkorrektur $L1 = l_{tx}$ ist kompensierbar, d. h. es wird nur in Richtung der x-Koordinate korrigiert.

- **TRFOPT (<Radius>)**

Beim Aktivieren der Zylindermanteltransformation kann der Zylinderradius (Wert > 0) mit Hilfe des optionalen Transformations-Parameters TRFOPT übergeben werden. Er überschreibt den Wert des MP 1030 00140[2].

Relevante NC-Funktionen

COORD(...)	Die 4-Achs-Zylindermantel-Transformation wird aktiviert. Dies kann wahlweise per Maschinenparametersatz (i=1..20) oder per Mtb-Systemdatum (vom Typ SysAxTrafo_t) erfolgen.
COORD(0) - oder - COORD(0,2)	Die aktive Transformation wird deaktiviert.
G152.x	Placement (Schiefe Ebene) aktivieren und damit Festlegung der Projektionsebene. Placement muss nach dem Aktivieren der Transformation eingeschaltet werden. Hinweis: Im Zusammenhang mit der Zylindermantel-Transformation hat die Placement-Funktion "Schiefe Ebene" G152.x kein "klassisches" Placementverhalten: G152.x definiert die (Y, Z)-Projektionsebene grundsätzlich mit "X-Placement = 0". X-Werte ungleich Null haben keine Wirkung, da die X-Placement-Komponente steuerungsintern zur Ermittlung des zugehörigen Punktes auf der Manteloberfläche benötigt wird.
G153	Ausschalten Placement (d. h. Bcs=Wcs)

Tab. 11-95: Relevante NC-Funktionen

Relevante Maschinenparameter

In den Maschinenparametern können bis zu 20 unterschiedliche Achstransformationen per nachfolgendem Maschinenparametersatz deklariert werden.

1030 00110	Definiert den Transformationstyp.
1030 00120	Definiert die an der Transformation beteiligten Systemachsen/-koordinaten der Transformation.
1030 00125	Definiert die Achsklassifikation der Transformationsachsen.
1030 00130	Definiert die Achspositionen der Referenzstellung.
1030 00140	Definiert die Längen- und Winkelparameter.
1030 00160	Epsilon-Umgebungen

Tab. 11-96: Relevante Maschinenparameter

Relevante Systemdaten

Die Achstransformation kann alternativ zu den Maschinenparameter auch per Systemdatum definiert werden. Dazu muss der Benutzer eine Systemvariable vom Typ "SysAxTrafo_t" anlegen (siehe Dokumentation "MTX Maschinenparameter", Kapitel "Systemdaten"). Der Strukturtyp "SysAxTrafo_t" ist in der XML-Schemadateien "/feprom/schemas/sdaxtrf.xsd" steuerungsinternen definiert und enthält, analog zu den relevanten Maschinenparametern, die nachfolgenden transformationsrelevanten Strukturelemente.

Strukturelement	Beschreibung
Type	Definiert den Transformationstyp.
AxisAssignment	Definiert die an der Transformation beteiligten Systemachsen/-koordinaten der Transformation.
AxisClassification	Definiert die Achsklassifikation der Transformationsachsen.
AxZeroPos	Definiert die Achspositionen der Referenzstellung.
LenParam	Definiert die Längen- und Winkelparameter.
EpsilonRanges	Epsilon-Umgebungen

Tab. 11-97: Relevante Systemdaten

11.20.2 Handlungsanweisung: 4-Achs-Zylindermanteltransformation

Applizieren

Applizieren Die Achstransformation kann über die relevanten Maschinenparameter appliziert werden. Dazu muss die Funktion Achstransformation über den Parameter **tra** "Achstransformationen" im Setup (SUP) aktiviert werden. Der Pfad des MP-Namens ist immer "TRA/AxTrafo[1..20]/...".

Alternativ dazu kann die Achstransformation über ein benutzerdefiniertes Systemdatum vom Typ SysAxTrafo_t appliziert werden.

Die beiden Varianten sind inhaltlich identisch.

Die Achspositionen der Referenzstellung müssen ermittelt werden.

In der nachfolgenden Tabelle sind jeweils die äquivalenten MP-Nummern, MP-Name und SD-Elemente angegeben. Um eine Transformation zu applizieren, muss eine der Varianten entsprechend besetzt werden.

Transformations-relevante Parameter

MP-Nummer	Beschreibung	
1030 00110	Transformationstyp	
	MP-Name: TrafoType	
	SD-Elem. : Type	
	3131021	4-Achs-Zylindermantel Abwicklung
	3131022	4-Achs-Zylindermantel Projektion

Achs-Transformation

1030 00120	Systemachsen/-koordinaten der Transformation			
	MP-Name: FwdInCoordIndTrafo[1..8]			
	SD-Elem. : AxisAssignment[1..8]			
	[1]	Systemachsnummer von X		
	[2]	Systemachsnummer von C		
	[3]	Systemachsnummer von Z		
1030 00125	[4]	Systemachsnummer von Y		
	[5..8]	nicht relevant		
	Achsklassifikationen der Transformationsachsen			
	MP-Name: CoordClass[1..8]			
	MP-Name: CoordDir[1..8]			
	SD-Elem. : AxisClassification[1..8]			
	Definiert die an der Transformation beteiligten Achsen inklusive Bewegungsrichtung.			
	- MP-Nr. oder SD-Element: Wert ">0" bedeutet: Die Achse verfährt/dreht positiv im Sinne der obigen Abbildungen. Wert "<0" bedeutet: Die Achse verfährt/dreht negativ im Sinne der obigen Abbildungen. <i>Mögliche Werte sind:</i> +/-1 = X-Achse +/-2 = Y-Achse +/-3 = Z-Achse +/-100 = A-Achse +/-200 = B-Achse +/-300 = C-Achse <i>MP-Name:</i> <i>Mögliche Werte für CoordClass sind:</i> - X, Y, Z (linear) - A, B, C (rotatorisch) <i>Mögliche Werte für CoordDir sind:</i> - positiv (bzgl. Abb. "Referenzstellung") - negativ (bzgl. Abb. "Referenzstellung")			
	Ist ein Parameter unbesetzt, gelten folgende Default-Einstellungen:			
		MP-Nr / SD-Elem.:	CoordClass:	CoordDir:
	[1]	1	X	positiv
	[2]	300	C	positiv
	[3]	3	Z	positiv
	[4]	2	Y	positiv
	[5..8]	nicht relevant	0	positiv

1030 00130	Achspositionen der Referenzstellung		
	MP-Name: RefPosTrafo[1..8]		
	SD-Elem. : AxZeroPos[1..8]		
	[1]	X ⁰	
	[2]	C ⁰	
	[3]	Z ⁰	
	[4]	Y ⁰	
[5..8]	nicht relevant		
1030 00140	Längen- und Winkelparameter		
	MP-Name: JointParTrafo[1..16]		
	SD-Elem. : LenParam[1..16]		
	Definiert die Längenvektoren l ₁ , l ₂ in [mm].		
	[1]	Radius	Default-Wert für den Zylinderradius. Wird beim Aktivieren der Achstransformation TRFOPT programmiert, wird dieser Wert damit überschrieben
	[2]	yMod	Nur relevant für die Variante 1 (Abwicklung), wenn es sich bei der C-Achse um eine Endlos-Modulo-Achse handelt.
		0	der Einschaltwert der y-Koordinate liegt im Intervall [-2*Pi*Radius, 0]
		1	der Einschaltwert der y-Koordinate liegt im Intervall [-Pi*Radius, +Pi*Radius] (empfohlen)
1030 00160	[3..16]	nicht relevant	
	Epsilon-Umgebungen		
1030 00160	MP-Name: EpsAxTrafo		
	SD-Elem. : EpsilonRanges		
	nicht relevant		

Tab. 11-98: Transformations-relevante Parameter

Aktivieren

Aktivieren Ist die 4-Achs-Transformation konfiguriert per

- Maschinenparameter, dann wird die Transformation im NC-Programm aktiviert mit (i=1,..20):

COORD(<i>)

- Systemdatum, dann wird die Transformation im NC-Programm aktiviert mit (Typ "SysAxTrafo_t"):

COORD(<SD.MtbAxTrafo>)

Programmbeispiel:

Die 4-Achs-Transformation ist im 4. Maschinenparametersatz definiert. Am Programmanfang ist keine Achstransformation aktiv:

N01 G1 F1000 XA110 YA10 C30	;die Achsen XA, YA, B, C werden verfahren
N02 COORD(4) TRFOPT(100)	;4-Achs-Zyl-Transformation mit Radius R=100 [mm] aktivieren
N03 x-5 y30 z50 phi10	; (y,z)-Position mit Eintauchtiefe 5 und Orientierung phi anfahren
N04 z10 phi20	;die Koordinaten z, phi werden verfahren
N99 COORD(0)	;4-Achstransformation ausschalten ;ab hier werden wieder die Achsen programmiert

Deaktivieren

Deaktivieren **COORD(0) bzw. COORD(0,2)**
schaltet die Achstransformation im NC-Programm aus.

11.21 5-Achs-Zylindermantel-Transformation

11.21.1 Beschreibung

Funktion

Name der Transformation	Transformationstyp	Wirkstelle
5-Achs-Zylindermantel		
Abwicklung (lineare Orientierung)	3232121	AT2
Projektion (lineare Orientierung)	3232122	AT2

Die 5-Achs-Zylindermantel-Transformationen werden für die Programmierung der Bearbeitungen auf der Mantelfläche eines zylindrischen Werkstücks verwendet. Sie ermöglichen Nutwandkorrekturen (z. B. parallele Nutwände), siehe dazu folgende Abbildung.

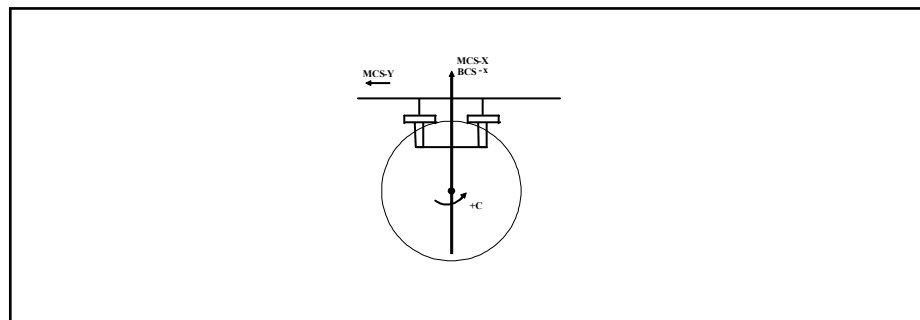


Abb. 11-84: Parallele Nutwände

Es existieren grundsätzlich zwei Varianten bezüglich der y-Koordinate:

- die programmierte Kontur bezieht sich auf die abgewinkelte Zylindermantelfläche (Standard)
- die programmierte Kontur bezieht sich auf die Ebenenprojektion auf die Zylindermantelfläche.

Die Maschinenkonfiguration besteht aus drei linearen MCS-Koordinaten X, Y, Z, welche ein rechtshändiges kartesisches Koordinatensystem aufspannen. Das Werkzeug ist an der Schwenkachse B angebracht. Auf der Rundachse C ist der zu bearbeitende Zylinder rotationssymmetrisch angebracht. Die Z-Achse läuft parallel zur Längsrichtung des Zylinders (siehe [Abb. 11-85](#) "TCP

in BCS-Nullposition" auf Seite 371 - Abb. 11-88 "Referenzposition Z-Achse" auf Seite 372).

Der BCS-Nullpunkt liegt auf der Manteloberfläche (siehe [Abb. 11-85 "TCP in BCS-Nullposition" auf Seite 371](#)). Die (negative) x-Koordinate beschreibt die Eintauchtiefe.

In beiden Varianten (Abwicklung und Projektion) wird die y-Bewegung von der Rundachse C erzeugt (siehe [Abb. 11-86 "Orientierung phi" auf Seite 371](#)). Die lineare Y-Achse erzeugt am TCP die Orientierung phi (siehe [Abb. 11-86 "Orientierung phi" auf Seite 371](#)), die Schwenkachse B generiert die Orientierung theta (siehe [Abb. 11-87 "Orientierung theta" auf Seite 372](#)).

Die beiden 5-Achs-Zylindermantel-Transformationen kommen an der Wirkstelle 2 zum Einsatz, d. h. sie transformieren zwischen dem Basiskoordinatensystem (BCS) mit den Koordinaten x, y, z, ϕ, θ und dem Maschinenkoordinatensystem (MCS) mit den Koordinaten X, Y, Z, B, C .

- Programmiert werden die BCS-Eingangsgrößen x, y, z, ϕ, θ
- Berechnet werden die MCS-Ausgangsgrößen X, Y, Z, B, C

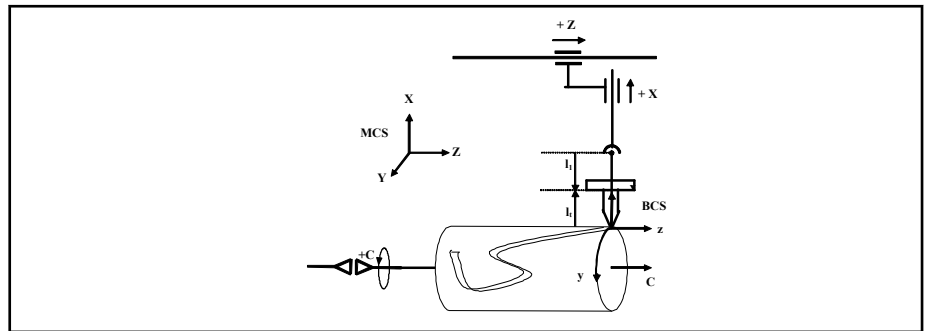


Abb. 11-85: TCP in BCS-Nullposition

Der Vektor $l_1 = (l_{1x}, 0, 0)$ zeigt von der B-Achse zur Werkzeugaufnahme, $l_t = (l_{tx}, 0, 0)$ ist der Werkzeugkorrekturvektor und zeigt vom TCP zur Werkzeugaufnahme.

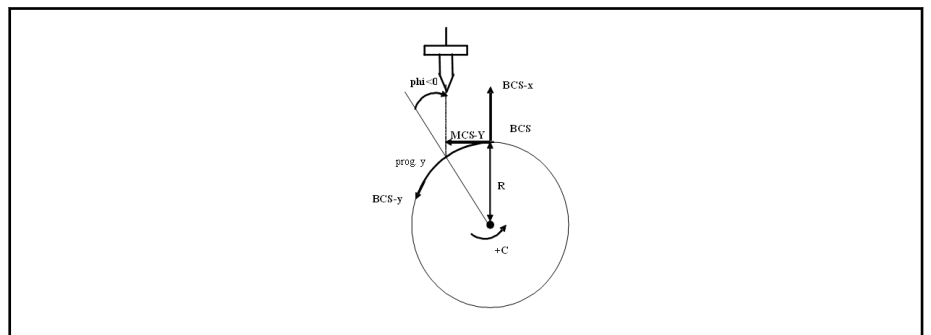


Abb. 11-86: Orientierung phi

Achs-Transformation

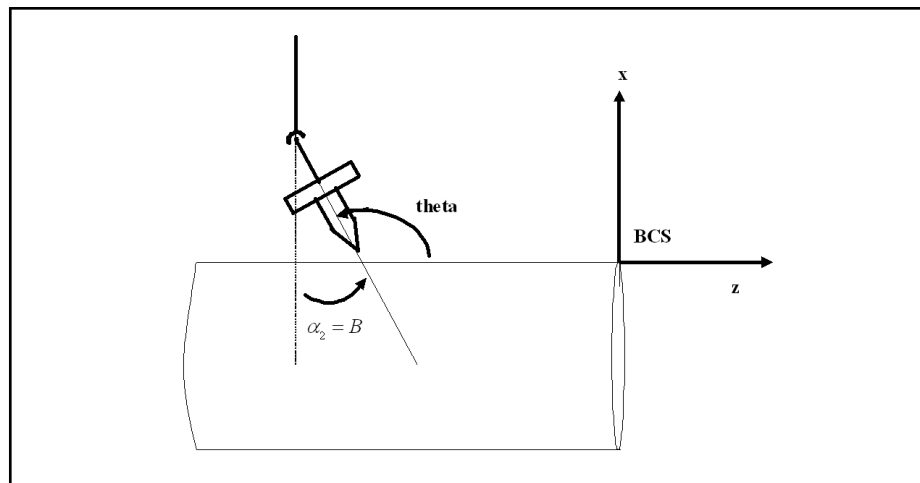


Abb. 11-87: Orientierung theta

Die Referenzposition der Z-Achse liegt in der C-Achse, vgl. folgende Abbildung.

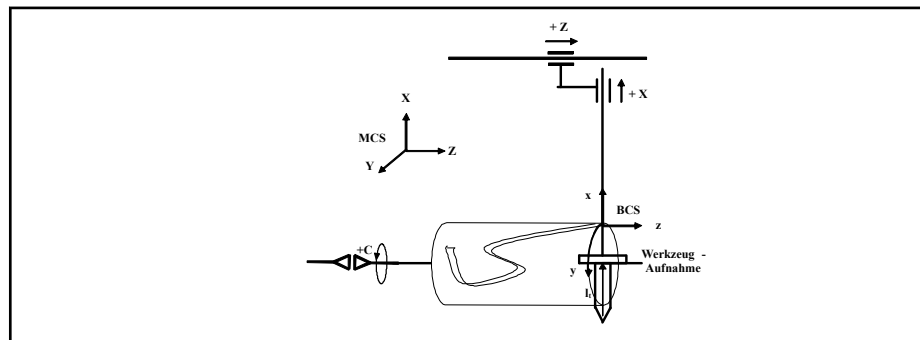


Abb. 11-88: Referenzposition Z-Achse

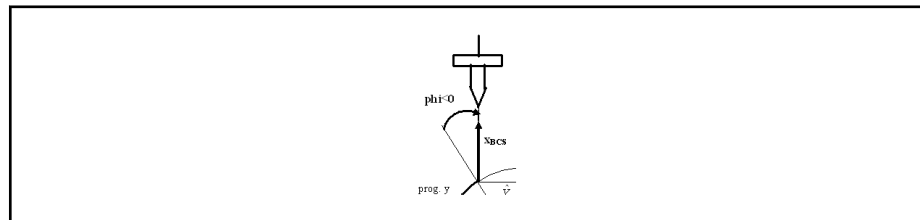


Abb. 11-89: Position und Orientierung phi-Abwicklung

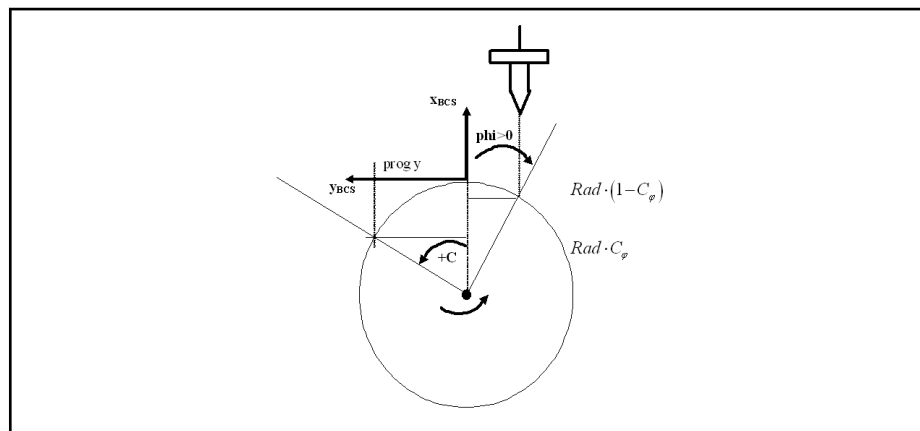


Abb. 11-90: Position und Orientierung phi-Projektion

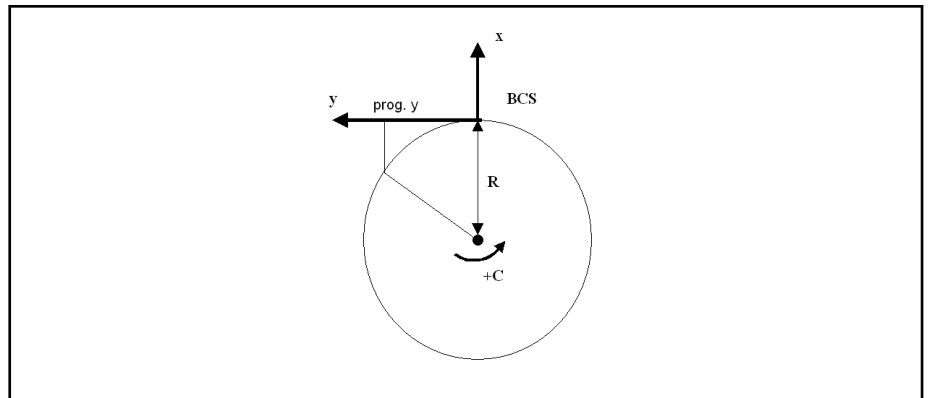


Abb. 11-91: Projektion

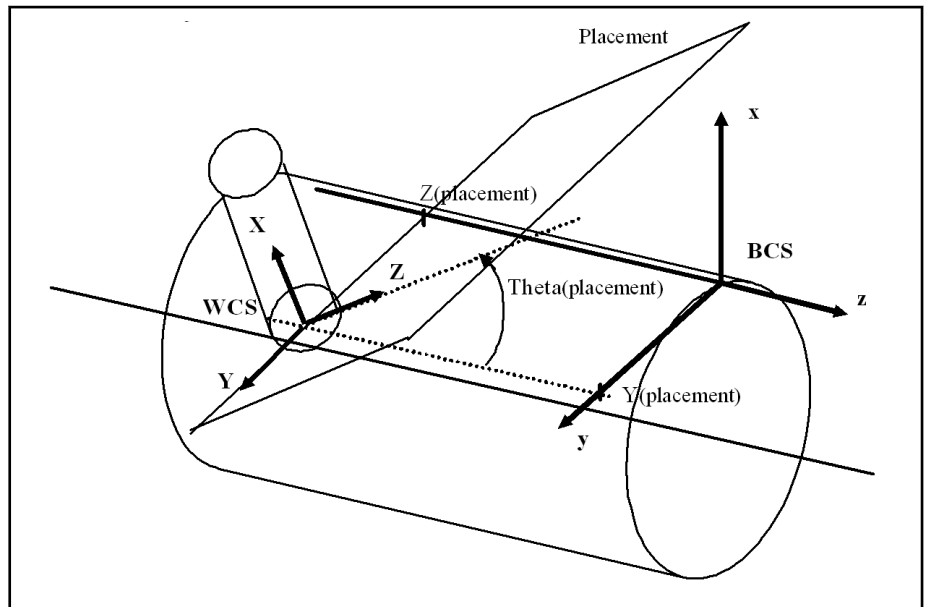


Abb. 11-92: Projektion

Referenzstellung

In der Referenzstellung zeigt die B-Achse mit $B=0$ senkrecht nach unten. Die Werkzeugaufnahme (TCP_0) befindet sich in der Rotationsachse C (Abb. 11-88 "Referenzposition Z-Achse" auf Seite 372). Der BCS-Nullpunkt liegt dann auf der Manteloberfläche mit $(x, y, z)=(0,0,0)$ (Abb. 11-85 "TCP in BCS-Nullposition" auf Seite 371).

Steht die Werkzeugspitze (inkl. Längenkorrektur) auf diesem Punkt, entspricht dies den zugehörigen MCS-Koordinaten $(X, C, Z, Y, B) = (\text{RadiusZylinder} + l_{tx} + X_0, C_0, Z_0, Y_0, B_0)$. D. h., die Referenzstellung der Werkzeugaufnahme ist vom BCS-Nullpunkt um den Radius R des Zylinders in negative x -Richtung verschoben (vgl. Abb. 11-85 "TCP in BCS-Nullposition" auf Seite 371, Abb. 11-88 "Referenzposition Z-Achse" auf Seite 372).

Die Werkzeugorientierung ist entlang X ausgerichtet (d. h. $\phi=0$, $\theta=0$).

Varianten

Es gibt zwei verschiedene Varianten der 5-Achs-Zylindermanteltransformation. Diese unterscheiden sich durch unterschiedliche y -Koordinaten:

1. die programmierte Kontur bezieht sich auf die abgewinkelte Zylindermantelfläche (Standard)

2. die programmierte Kontur bezieht sich auf eine Ebenenprojektion auf die Zylindermantelfläche

Der Vektor $I_1 = (I_{1x}, 0, 0)$ definiert in beiden Fällen die Geometrie der 5-Achs-Kinematik. In der Nullposition zeigt I_1 jeweils von der B-Achse zum Nullpunkt des BCS (siehe [Abb. 11-85 "TCP in BCS-Nullposition" auf Seite 371](#)).

Die positiven Dreh-/Verfahr-Richtungen der Achsen sind in den Abbildungen dargestellt. Diese sind entsprechend einzustellen!

Orientierung

Linear-Orientierung

Mit Einschalten der 5-Achs-Zylindermantel-Transformation wird die lineare Orientierungsbewegung (Typ 3232121, 3232122) aktiviert.

Die Winkel phi und theta werden linear vom Anfangs- zum Endpunkt interpoliert. Das Verhalten der "kürzeste Weg"-Logik hängt von der Programmierungsart der Orientierungsbewegung ab. Bei phi-, theta-Programmierung ist die "kürzeste Weg"-Logik ausgeschaltet. Bei der O(...)-Programmierung ist die "kürzeste Weg"-Logik eingeschaltet.

phi, theta können Werte aus den folgenden Bereichen annehmen:

-unendlich < phi < unendlich ; -180 ≤ theta < 180



Die Linear-Orientierung ist nicht "Schiefe-Ebenen"-tauglich. Programmierte Orientierungen beziehen immer auf das Basis-Koordinatensystem BCS und nicht, wie sonst üblich, auf das Werkstück-Koordinatensystem WCS!

Besonderheiten

- Der Verfahrbereich der B-Achse ist auf +/-90 Grad begrenzt.
- Die C-Achse kann als Endlos- oder als Rundachse betrieben werden.
- **Werkzeugkorrektur**
Beide Achstransformationen unterstützen eine Werkzeugkorrektur. Der Werkzeugkorrekturvektor $I_t = (I_{tx}, 0, 0)$ zeigt vom korrigierten Werkzeugmittelpunkt TCP_1 zum unkorrigierten TCP_0 . Nur die Längenkorrektur $L1 = I_{tx}$ ist kompensierbar, d. h. es wird nur in Richtung der x-Koordinate korrigiert.
- **TRFOPT (<Radius>)**
Beim Aktivieren der Zylindermanteltransformation kann der Zylinderradius (Wert > 0) mit Hilfe des optionalen Transformations-Parameters TRFOPT übergeben werden. Er überschreibt den Wert des MP 1030 00140[2].

Relevante NC-Funktionen

COORD(...)	Die 5-Achs-Zylindermantel-Transformation wird aktiviert. Dies kann wahlweise per Maschinenparametersatz (i=1..20) oder per Mtb-Systemdatum (vom Typ SysAxTrafo_t) erfolgen.
COORD(0) - oder - COORD(0,2)	Die aktive Transformation wird deaktiviert.

G152.x	Placement (Schiefe Ebene) aktivieren und damit Festlegung der Projektionsebene. Placement muss nach dem Aktivieren der Transformation eingeschaltet werden. Hinweis: Im Zusammenhang mit der Zylindermantel-Transformation hat die Placement-Funktion "Schiefe Ebene" G152.x kein "klassisches" Placementverhalten: G152.x definiert die (Y, Z)-Projektionsebene grundsätzlich mit "X-Placement = 0". X-Werte ungleich Null haben keine Wirkung, da die X-Placement-Komponente steuerungsintern zur Ermittlung des zugehörigen Punktes auf der Manteloberfläche benötigt wird.
G153	Ausschalten Placement (d. h. Bcs=Wcs)

Tab. 11-99: Relevante NC-Funktionen

Relevante Maschinenparameter

In den Maschinenparametern können bis zu 20 unterschiedliche Achstransformationen per nachfolgendem Maschinenparametersatz deklariert werden.

1030 00110	Definiert den Transformationstyp.
1030 00120	Definiert die an der Transformation beteiligten Systemachsen/-koordinaten der Transformation.
1030 00125	Definiert die Achsklassifikation der Transformationsachsen.
1030 00130	Definiert die Achspositionen der Referenzstellung.
1030 00140	Definiert die Längen- und Winkelparameter.
1030 00160	Epsilon-Umgebungen

Tab. 11-100: Relevante Maschinenparameter

Relevante Systemdaten

Die Achstransformation kann alternativ zu den Maschinenparameter auch per Systemdatum definiert werden. Dazu muss der Benutzer eine Systemvariable vom Typ "SysAxTrafo_t" anlegen (siehe Dokumentation "MTX Maschinenparameter", Kapitel "Systemdaten"). Der Strukturtyp "SysAxTrafo_t" ist in der XML-Schemadateien "/feprom/schemas/sdaxtrf.xsd" steuerungsintern definiert und enthält, analog zu den relevanten Maschinenparametern, die nachfolgenden transformationsrelevanten Strukturelemente.

Strukturelement	Beschreibung
Type	Definiert den Transformationstyp.
AxisAssignment	Definiert die an der Transformation beteiligten Systemachsen/-koordinaten der Transformation.
AxisClassification	Definiert die Achsklassifikation der Transformationsachsen.
AxZeroPos	Definiert die Achspositionen der Referenzstellung.
LenParam	Definiert die Längen- und Winkelparameter.
EpsilonRanges	Epsilon-Umgebungen

Tab. 11-101: Relevante Systemdaten

11.21.2 Handlungsanweisung: 5-Achs-Zylindermanteltransformation

Applizieren

Applizieren Die Achstransformation kann über die relevanten Maschinenparameter appliziert werden. Dazu muss die Funktion Achstransformation über den Parameter **tra** "Achstransformationen" im Setup (SUP) aktiviert werden. Der Pfad des MP-Namens ist immer "TRA/AxTrafo[1..20]/...".

Alternativ dazu kann die Achstransformation über ein benutzerdefiniertes Systemdatum vom Typ SysAxTrafo_t appliziert werden.

Die beiden Varianten sind inhaltlich identisch.

Die Achspositionen der Referenzstellung müssen ermittelt werden.

In der nachfolgenden Tabelle sind jeweils die äquivalenten MP-Nummern, MP-Name und SD-Elemente angegeben. Um eine Transformation zu applizieren, muss eine der Varianten entsprechend besetzt werden.

Transformations-relevante Parameter

MP-Nummer	Beschreibung	
1030 00110	Transformationstyp	
	MP-Name: TrafoType	
	SD-Elem. : Type	
	3232121	5-Achs-Zylindermantel Abwicklung (Linear-Orientierung)
	3232122	5-Achs-Zylindermantel Projektion (Linear-Orientierung)
1030 00120	Systemachsen/-koordinaten der Transformation	
	MP-Name: FwdInCoordIndTrafo[1..8]	
	SD-Elem. : AxisAssignment[1..8]	
	[1]	Systemachsnummer von X
	[2]	Systemachsnummer von C
	[3]	Systemachsnummer von Z
	[4]	Systemachsnummer von Y
	[5]	Systemachsnummer von B
	[6..8]	nicht relevant

MP-Nummer	Beschreibung			
1030 00125	Achsklassifikationen der Transformationsachsen MP-Name: CoordClass[1..8] MP-Name: CoordDir[1..8] SD-Elem. : AxisClassification[1..8]			
	Definiert die an der Transformation beteiligten Achsen inklusive Bewegungsrichtung. - MP-Nr. oder SD-Element: Wert ">0" bedeutet: Die Achse verfährt/dreht positiv im Sinne der obigen Abbildungen. Wert "<0" bedeutet: Die Achse verfährt/dreht negativ im Sinne der obigen Abbildungen. <i>Mögliche Werte sind:</i> +/-1 = X-Achse +/-2 = Y-Achse +/-3 = Z-Achse +/-100 = A-Achse +/-200 = B-Achse +/-300 = C-Achse - MP-Name: <i>Mögliche Werte für CoordClass sind:</i> - X, Y, Z (linear) - A, B, C (rotatorisch) <i>Mögliche Werte für CoordDir sind:</i> - positiv (bzgl. Abb. "Referenzstellung") - negativ (bzgl. Abb. "Referenzstellung")			
	Ist ein Parameter unbesetzt, gelten folgende Default-Einstellungen:			
		MP-Nr / SD-Elem.:	CoordClass:	CoordDir:
	[1]	1	X	positiv
	[2]	300	C	positiv
	[3]	3	Z	positiv
	[4]	2	Y	positiv
	[5]	200	B	positiv
	[6..8]	nicht relevant	0	positiv

Achs-Transformation

MP-Nummer	Beschreibung	
1030 00130	Achspositionen der Referenzstellung	
	MP-Name: RefPosTrafo[1..8]	
	SD-Elem. : AxZeroPos[1..8]	
	[1]	X^0
	[2]	C^0
	[3]	Z^0
	[4]	Y^0
1030 00140	[5]	B^0
	[6..8]	nicht relevant
	Längen- und Winkelparameter	
	MP-Name: JointParTrafo[1..16]	
	SD-Elem. : LenParam[1..16]	
	Definiert die Längenvektoren l_1, l_2 in [mm].	
	[1]	Radius
1030 00160	[2]	yMod
		Default-Wert für den Zylinderradius. Wird beim Aktivieren der Achstransformation TRFOPT programmiert, wird dieser Wert damit überschrieben
		Nur relevant für die Variante 1 (Abwicklung), wenn es sich bei der C-Achse um eine Endlos-Modulo-Achse handelt.
		0
		1
1030 00160	[3]	der Einschaltwert der y-Koordinate liegt im Intervall $[-2 \cdot \pi \cdot \text{Radius}, 0]$
		der Einschaltwert der y-Koordinate liegt im Intervall $[-\pi \cdot \text{Radius}, +\pi \cdot \text{Radius}]$ (empfohlen)
		Wert < 0 eingeben
1030 00160	[4..16]	nicht relevant
	Epsilon-Umgebungen	
	MP-Name: EpsAxTrafo	
1030 00160	SD-Elem. : EpsilonRanges	
	nicht relevant	

Tab. 11-102: Transformations-relevante Parameter

Aktivieren

Aktivieren Ist die 5-Achs-Transformation konfiguriert per

- Maschinenparameter, dann wird die Transformation im NC-Programm aktiviert mit (i=1,..20):

COORD(<i>)

- Systemdatum, dann wird die Transformation im NC-Programm aktiviert mit (Typ "SysAxTrafo_t"):

COORD(<SD.MtbAxTrafo>)

Programmbeispiel:

Die 5-Achs-Transformation ist im 4. Maschinenparametersatz definiert. Am Programmstart ist keine Achstransformation aktiv:

N01 G1 F1000 XA110 YA10 B20	;die Achsen XA, YA, B, C werden verfahren C30
N02 COORD(4) TRFOPT(100)	;5-Achs-Zyl-Transformation mit Radius R=100 [mm] aktivieren
N03 x-5 y30 z50 phi10 theta70	; (y,z)-Position mit Eintauchtiefe 5 und Orien- tierung phi, theta anfahren
N04 z10 phi20 theta30	;die Koordinaten z, phi, theta werden verfah- ren
N99 COORD(0)	;5-Achstransformation ausschalten ;ab hier werden wieder die Achsen program- miert

Deaktivieren

Deaktivieren **COORD(0)** bzw. **COORD(0,2)**
schaltet die Achstransformation im NC-Programm aus.

11.22 Stirnseiten-Transformation

11.22.1 Beschreibung

Allgemeines

Die Stirnseiten-Transformation wird für die Koordinatenprogrammierung einer Kontur auf der Stirnseite eines sich drehenden Werkstücks verwendet.

Dazu existieren 2 Varianten:

- 2-Achs-Transformation in (x,y)
- 3-Achs-Transformation in (x,y,z)

Die wesentlichen Transformationseigenschaften, was beispielsweise Transformationszweig oder Singularitäten angeht, sind bei beiden Varianten identisch. Bei der 3-Achsvariante wird nur zusätzlich die z-Koordinate (inklusive z-Werkzeuglängenkorrektur) mit interpoliert.

11.22.2 Variante 1 (2-Achs-Transformation)

Funktion

Name der Transformation	Transformationstyp	Wirkstelle	Koordinatensystem
Stirnseiten-Transformation 2-Achs-Transformation	2011001	AT2	MCS, BCS

Tab. 11-103: Funktion - Variante 1: 2-Achs-Transformation

Ausgangsgrößen im MCS: X, C

Eingangsgrößen im BCS: x, y

11.22.3 Variante 2 (3-Achs-Transformation)

Funktion

Name der Transformation	Transformationstyp	Wirkstelle	Koordinatensystem
Stirnseiten-Transformation 3-Achs-Transformation	3021003	AT2	MCS, BCS

Tab. 11-104: Funktion - Variante 2: 3-Achs-Transformation

Ausgangsgrößen im MCS: X, Z, C

Eingangsgrößen im BCS: x, y, z

11.22.4 Gemeinsame Eigenschaften der Varianten 1, 2

Allgemeines

Mit der Stirnseiten-Transformation ist eine Koordinatenprogrammierung (x,y) einer Kontur auf der Stirnseite eines Werkstücks möglich.

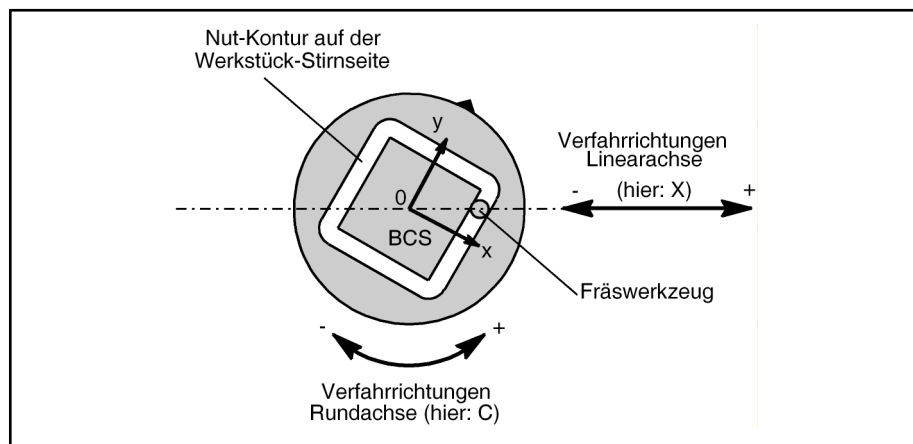


Abb. 11-93: Stirnseiten-Transformation

Dafür stellt die Transformation eine Beziehung zwischen den Koordinaten x und y im BCS und den entsprechenden Achskoordinaten der beteiligten Achsen (hier: X, C) her, wenn folgende Konfiguration erfüllt ist:

- das Werkstück wird durch eine Rund- oder Endlosachse bewegt, die sich um die Werkzeugzustellkoordinate dreht.

Beispiel:

Rund- oder Endlosachse "C" als Spindel/C-Achse (Spindel muss vor dem Aktivieren der Stirnseiten-Transformation in C-Achsbetrieb umgeschaltet und im Kanal übernommen werden).

- das Werkzeug (oder das Werkstück selbst) wird durch eine Linearachse bewegt, die horizontal parallel zur Werkstückaufspanfläche verläuft. Das kann eine Achse entlang einer kartesischen Hauptrichtung oder eine zur Hauptrichtung parallele Achse sein.

Beispiel:

Linearachse "X"

- Die Rund-/Endlosachse dreht sich um die Werkzeugzustellachse des BCS (hier: z).
- Die beteiligten Achsen (hier: X,C) müssen sich im gleichen Kanal befinden, wenn die Stirnseiten-Transformation aktiviert wird.

Relevante NC-Funktionen

COORD(<i>)</i>	Mit $i=1..20$ wird eine Stirnseiten-Transformation aktiviert, die in MP 1030 00110 definiert ist.
COORD(0,2) oder COORD(0)	
	Die aktive Stirnseiten-Transformation wird deaktiviert.

Tab. 11-105: Relevante NC-Funktionen

Relevante Maschinenparameter

In den Maschinenparametern können bis zu 20 unterschiedliche Achstransformationen per nachfolgendem Maschinenparametersatz deklariert werden.

1030 00110	Definiert den Transformationstyp
1030 00120	Definiert die an der Transformation beteiligten Systemachsen/-koordinaten der Transformation
1030 00125	Definiert die Achsklassifikation der Transformationsachsen
1030 00130	Definiert die Achspositionen der Referenzstellung
1030 00140	Definiert die Längen- und Winkelparameter
1030 00160	Epsilon-Umgebungen

Tab. 11-106: Relevante Maschinenparameter

Relevante Systemdaten

Die Achstransformation kann alternativ zu den Maschinenparametern auch per Systemdatum definiert werden. Dazu muss der Benutzer eine Systemvariable vom Typ "SysAxTrafo_t" anlegen (siehe Dokumentation "IndraMotion MTX 14VRS Maschinenparameter", Kapitel "Systemdaten"). Der Strukturtyp "SysAxTrafo_t" ist in der XML-Schemadatei "/feprom/schemas/sdaxtrf.xsd" steuerungsintern definiert und enthält, analog zu den relevanten Maschinenparametern, die nachfolgenden transformationsrelevanten Strukturelemente.

Strukturelement	Beschreibung
Type	Definiert den Transformationstyp
AxisAssignment	Definiert die an der Transformation beteiligten Systemachsen/-koordinaten der Transformation
AxisClassification	Definiert die Achsklassifikation der Transformationsachsen
AxZeroPos	Definiert die Achspositionen der Referenzstellung
LenParam	Definiert die Längen- und Winkelparameter
EpsilonRanges	Epsilon-Umgebungen

Tab. 11-107: Relevante Systemdaten

Referenzstellung

In der Referenzstellung befindet sich die Rund-/Endlosachse (hier: C) und die Linearachse (hier: X) auf der Position $x = y = 0$ des BCS. Diese Position liegt auf der Drehachse der beteiligten Rund-/Endlosachse (hier: C). Gleichzeitig ist dort die Position des TCP (Tool Center Point), wenn $x = y = z = 0$ ist.

Achs-Transformation

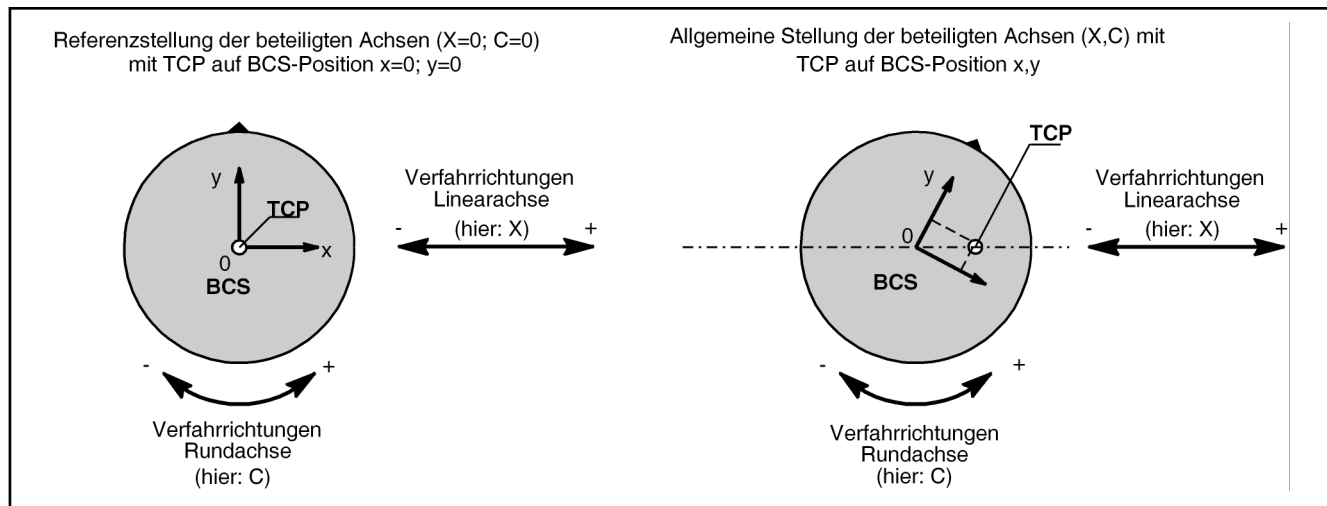


Abb. 11-94: Referenzstellung und allgemeine Stellung der Maschinenachsen

Die Ausrichtung des BCS ist mit der Stirnseite des Werkstücks fest verbunden. Dadurch dreht sich das BCS, wenn das Werkstück dreht.

Der Nullpunkt der BCS-Koordinatenachsen x und y liegt auf der Drehachse der beteiligten Rund-/Endlosachsen (hier: C). Die Rund-/Endlosachse dreht sich um die Werkzeugzustellachse des BSC (hier: z).

TCS, TCP

TCS bezeichnet das Werkzeugkoordinatensystem.

TCP₀ / TCP₁ bezeichnet den Tool Center Point (Werkzeugmittelpunkt) **ohne/ mit** Korrektur.

Der TCP (Tool Center Point) befindet sich in Referenzstellung, wenn das Werkzeug die Position x = y = z = 0 im BCS einnimmt.

Werkzeugkorrektur

Werkzeuglängenkorrektur L_t :

Die Werkzeuglängenkorrektur wirkt im TCS, wobei der Korrekturvektor $\vec{l}_t$ vom korrigierten Werkzeugmittelpunkt (TCPC) zum unkorrigierten TCP₀ zeigt.

Bei der 2-Achsvariante ist die Werkzeuglängenkorrektur in Richtung der Linearachse X realisiert (entspricht L1-Wert), d. h. der Korrekturvektor hat die Form

$$\vec{l}_t = \begin{pmatrix} L_1 \\ 0 \end{pmatrix}$$

Bei der 3-Achsvariante ist die Werkzeuglängenkorrektur in Richtung der Linearachsen X, Z realisiert (entspricht L1-, L3-Wert), d. h. der Korrekturvektor hat die Form

$$\vec{l}_t = \begin{pmatrix} L_1 \\ 0 \\ L_3 \end{pmatrix}$$

Der Korrekturwert $L2 \neq 0$ ist für die Praxis nicht relevant. Er entspricht einem Werkzeug, das das Zentrum nicht bearbeiten kann. Ein Kreis mit Radius $L2$ um den BCS-Nullpunkt wäre unerreichbar.

Werkzeugradiuskorrektur

Bei aktiver Werkzeugradiuskorrektur bewegt sich der Werkzeugeingriffspunkt auf einer zur Linearachsen-Verfahrrichtung parallelen Geraden, deren Verlängerung durch die Drehachse der beteiligten Rund-/Endlosachse läuft.

Die Linearachse bewegt sich horizontal parallel zur Werkstückaufspannfläche.

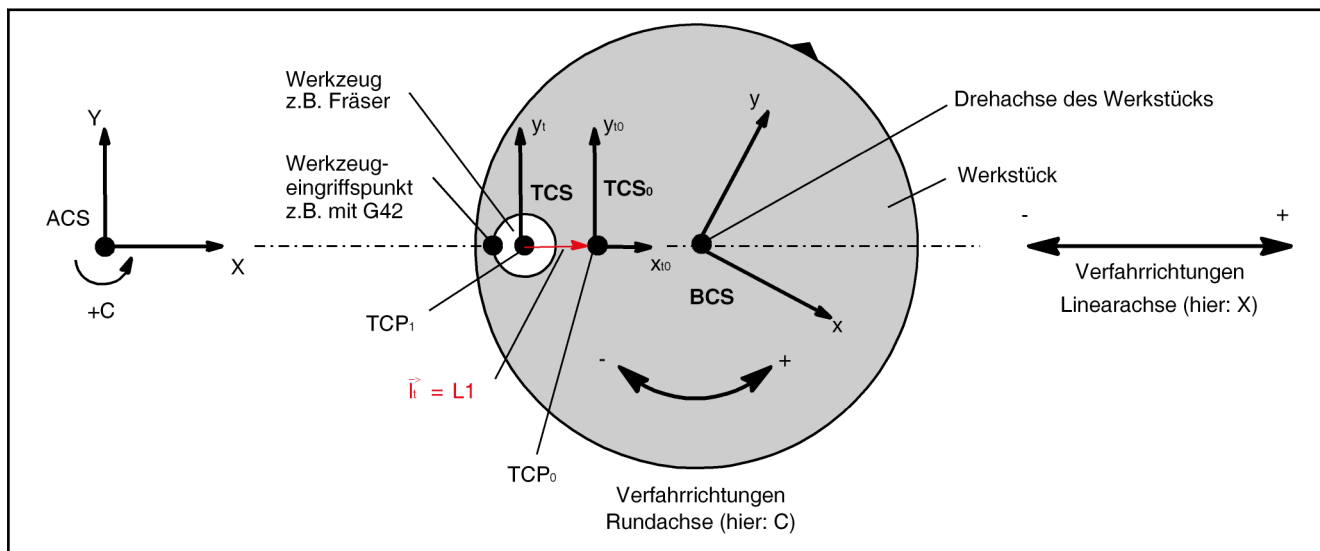


Abb. 11-95: Werkzeugradiuskorrektur

Transformationszweige und automatischer Zweigwechsel

Jede TCP-Position, ausgenommen der BCS-Nullpunkt, kann durch zwei unterschiedliche Maschinenkoordinatenpaare der beteiligten Achsen abgebildet werden (siehe folgende Abbildung). Deshalb sind in der Steuerung zur Berechnung der Transformation zwei "Transformationszweige" implementiert.

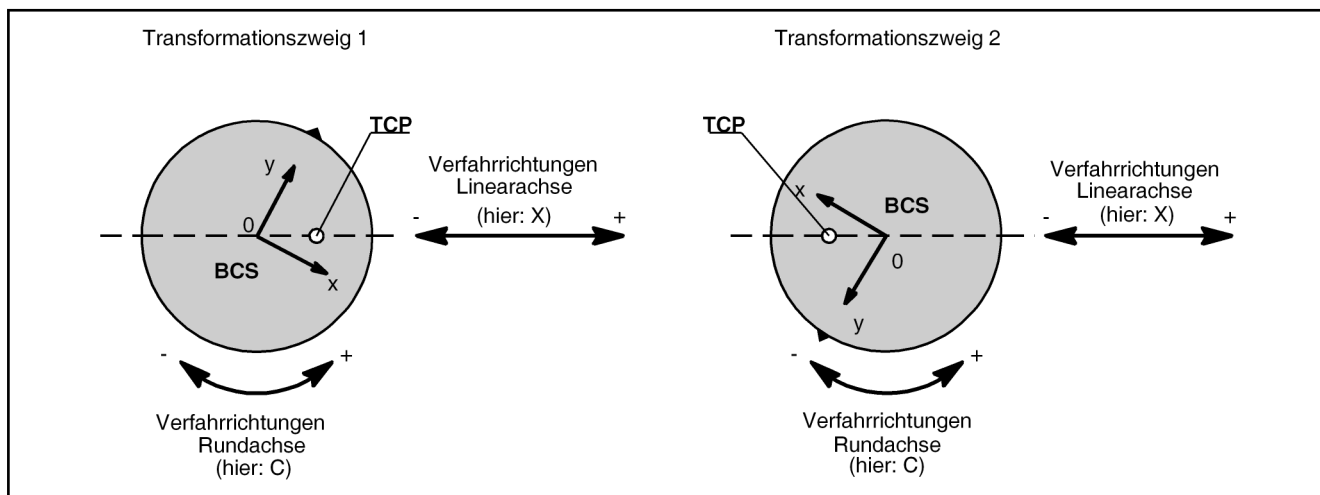


Abb. 11-96: Identische Position des TCP im BCS; abbildbar durch 2 unterschiedliche Maschinenkoordinatenpaare

Welcher Zweig für die internen Transformationsberechnungen benutzt wird, entscheidet sich beim Einschalten der Stirnseiten-Transformation. Ein auto-

matischer Übergang zwischen beiden Zweigen ist danach nur beim Durchfahren des Nullpunktes der BCS-Koordinatenachsen x und y möglich.

Der **automatische Zweigwechsel** ist immer dann von Vorteil, wenn der BCS-Nullpunkt ($x = 0$, $y = 0$) durchfahren werden soll. Wäre in der Steuerung nur ein einziger Transformationszweig implementiert, müsste in solchen Fällen prinzipbedingt ein Zwischensatz eingefügt werden, der die Rund-/Endlosachse um 180° dreht. Durch den automatischen Zweigwechsel ist diese 180° -Drehung überflüssig.

Allerdings gibt es auch Fälle, in denen ein automatischer Zweigwechsel unerwünscht ist. So ist es z. B. notwendig, den automatischen Zweigwechsel zu verbieten, wenn die Linearachse nur einen eingeschränkten Verfahrbereich hat, der zur Bearbeitung der kompletten Stirnfläche nicht ausreicht.

In diesem Fall erzeugt die NC automatisch einen Zwischensatz mit Drehung der Rund-/Endlosachse um 180° . Dadurch kann der benötigte Verfahrbereich der Linearachse auf die Hälfte reduziert werden.

Ob ein automatischer Zweigwechsel erlaubt ist, können Sie per MP 1030 00140 konfigurieren.

Besonderheiten

Bahnbewegungen durch die Drehachse der beteiligten Rund-/Endlosachse:

Die Drehachse der beteiligten Rund-/Endlosachse kann

1. **ohne Richtungswechsel** der Werkzeugbahn (tangential) oder
2. **mit Richtungswechsel** der Werkzeugbahn (nichttangential) überfahren werden.

- Zu 1.:

Bei erlaubtem Zweigwechsel schaltet die Steuerung automatisch den Zweig um. Eine 180° Grad-Drehung der beteiligten Rund-/Endlosachse ist somit nicht erforderlich.

Bei verbotenen Zweigwechsel zerlegt die NC den Satz automatisch in drei Teilsätze:

Erster Teilsatz: Bahnbewegung bis zur BCS-Koordinate 0,0.

Zweiter Teilsatz: 180° -Drehung der beteiligten Rund-/Endlosachse.

Dritter Teilsatz: Restbahnbewegung von 0,0 bis zum Endpunkt.

- Zu 2.:

Wechselt die Werkzeugbahn an der BCS-Koordinate 0,0 die Richtung, führt das zu einer Unstetigkeit; die Rund-/Endlosachsenposition würde an dieser Stelle "springen".

Die Stirnseiten-Transformation überwacht die Sollbahn auf diesen Effekt und fügt bei Bedarf automatisch einen Zwischensatz ein, in dem die Rund-/Endlosachse entsprechend gedreht wird.

Bei der Erzeugung des passenden Zwischensatzes berücksichtigt die NC auch einen ggf. erlaubten Zweigwechsel.

Beispiel:

N05x100 y0

N10x0

N20x-100 y100

Zwischen den Sätzen N10 und N20 dreht die NC die Rundachse von Position 0 Grad auf 45° , bevor sie die TCP-Bewegung mit N20 fortsetzt. Dabei findet ein Wechsel des Transformationszweiges statt.

Wäre der automatische Zweigwechsel verboten, würde die NC die Rundachse auf Position 225° drehen. Sehen Sie dazu die folgende Abbildung.

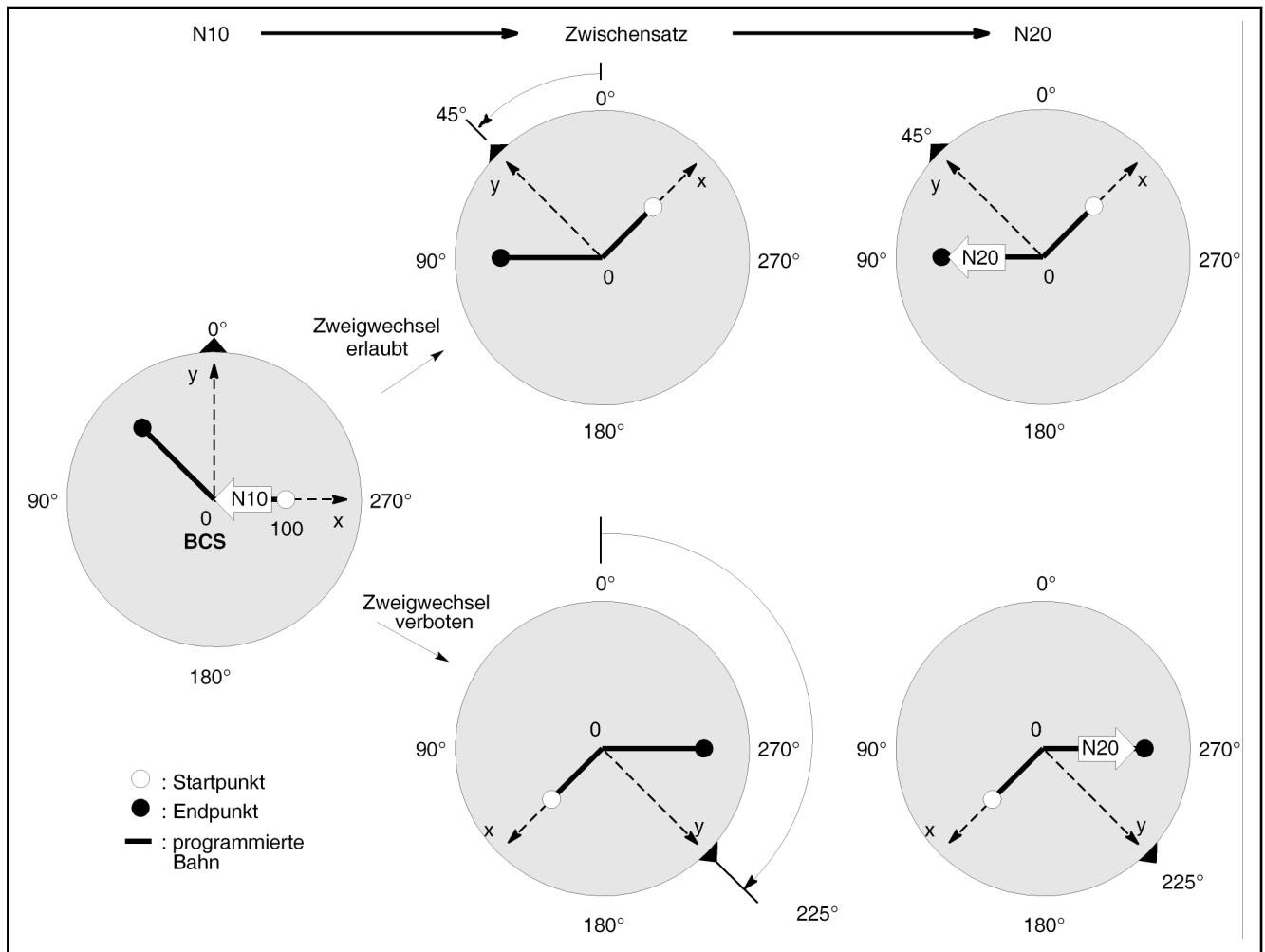


Abb. 11-97: Erzeugung eines Zwischensatzes bei Satzübergang im BCS-Nullpunkt

Bahnbewegungen nahe der Drehachse der beteiligten Rund-/Endlosachse:

Bei der Annäherung der Werkzeugbahn an die Drehachse der beteiligten Rundachse werden in Abhängigkeit vom Abstand zur Drehachse verschiedene Strategien angeboten:

- Abstand ≤ 0.01 mm an der Drehachse:

Werkzeugbahnen werden von der NC zwangsweise durch den BCS-Nullpunkt ($x=y=0$) geführt.

Für vorbeiführende Bahnen gilt:

- Der Satz wird in zwei Teilsätze zerlegt. Bei einem Kreis werden nach Möglichkeit zwei Teilkreise mit dem selben Radius wie der Ursprungskreis generiert.
- Der erste Teilsatz endet, der zweite Teilsatz beginnt im Nullpunkt.

- Abstand > 0.01 mm an der Drehachse

Diese Werkzeugbahnen erfordern prinzipbedingt eine Absenkung der max. Bahngeschwindigkeit und Bahnbeschleunigung. Die Absenkung ist

umso größer, je näher die Bahn in die Umgebung der 0.01 mm-Grenze kommt.

Damit diese Absenkung nicht auf der kompletten Bahnlänge wirkt (Kriechbewegung über den gesamten Satz), wird die Bahn in Teilsätze zerlegt. Damit ergibt sich nur für die Nullpunkt-nahen Teilsätze eine starke Absenkung.

11.22.5 Handlungsanweisung: 2/3-Achs-Transformation Stirnseite

Applizieren Die Achstransformation kann über die relevanten Maschinenparameter appliziert werden. Dazu muss die Funktion Achstransformation über den Parameter **tra** "Achstransformationen" im Setup (SUP) aktiviert werden. Der Pfad des MP-Namens ist immer "TRA/AxTrafo[1..20]/...".

Alternativ dazu kann die Achstransformation über ein benutzerdefiniertes Systemdatum vom Typ SysAxTrafo_t appliziert werden.

Die beiden Varianten sind inhaltlich identisch.

Die Achspositionen der Referenzstellung müssen ermittelt werden.

In der nachfolgenden Tabelle sind jeweils die äquivalenten MP-Nummern, MP-Name und SD-Elemente angegeben. Um eine Transformation zu applizieren, muss eine der Varianten entsprechend besetzt werden.

Transformations-relevante Parameter

MP-Nummer	Beschreibung	
1030 00110	Transformationstyp	
	MP-Name: TrafoType	
	SD-Elem. : Type	
	2011001	2-Achs-Variante
	3021003	3-Achs-Variante
1030 00120	Systemachsen/-koordinaten der Transformation	
	MP-Name: FwdInCoordIndTrafo[1..8]	
	SD-Elem. : AxisAssignment[1..8]	
	[1]	Systemachsnummer von X
	[2]	Systemachsnummer von C
	[3]	Systemachsnummer von Z (nur 3-Achs-Variante)
	[4..8]	nicht relevant

MP-Nummer	Beschreibung																				
1030 00125	Achsklassifikationen der Transformationsachsen MP-Name: CoordClass[1..8] MP-Name: CoordDir[1..8] SD-Elem. : AxisClassification[1..8] Definiert die an der Transformation beteiligten Achsen inklusive Bewegungsrichtung. - MP-Nr. oder SD-Element: Wert ">0" bedeutet: Die Achse verfährt/dreht positiv im Sinne der obigen Abbildungen. Wert "<0" bedeutet: Die Achse verfährt/dreht negativ im Sinne der obigen Abbildungen. Mögliche Werte sind: +/-1 = X-Achse +/-2 = Y-Achse +/-3 = Z-Achse +/-100 = A-Achse +/-200 = B-Achse +/-300 = C-Achse - MP-Name: - Mögliche Werte für CoordClass sind - X,Y,Z (linear) - A,B,C (rotatorisch) - Mögliche Werte für CoordDir sind: - positiv (bzgl. Abb. "Referenzstellung") - negativ (bzgl. Abb. "Referenzstellung") Ist ein Parameter unbesetzt, gelten folgende Default-Einstellungen: <table><tr><th></th><th>MP-Nr / SD-Elem.:</th><th>CoordClass:</th><th>CoordDir:</th></tr><tr><td>[1]</td><td>1</td><td>X</td><td>positiv</td></tr><tr><td>[2]</td><td>300</td><td>C</td><td>positiv</td></tr><tr><td>[3]</td><td>3</td><td>Z</td><td>positiv</td></tr><tr><td>[4..8]</td><td>nicht relevant</td><td>0</td><td>positiv</td></tr></table>		MP-Nr / SD-Elem.:	CoordClass:	CoordDir:	[1]	1	X	positiv	[2]	300	C	positiv	[3]	3	Z	positiv	[4..8]	nicht relevant	0	positiv
	MP-Nr / SD-Elem.:	CoordClass:	CoordDir:																		
[1]	1	X	positiv																		
[2]	300	C	positiv																		
[3]	3	Z	positiv																		
[4..8]	nicht relevant	0	positiv																		

Achs-Transformation

MP-Nummer	Beschreibung
1030 00130	Achspositionen der Referenzstellung MP-Name: RefPosTrafo[1..8] SD-Elem. : AxZeroPos[1..8]
	[1] X ⁰
	[2] C ⁰
	[3] Z ⁰
	[4..8] nicht relevant
1030 00140	Längen- und Winkelparameter MP-Name: JointParTrafo[1..16] SD-Elem. : LenParam[1..16] Hier kann festgelegt werden, ob die Linearachse X den Werkstückmittelpunkt überschreiten darf oder nicht.
	[1] 0 : Linearachse X darf Mittelpunkt überfahren +1 : Linearachse X auf positive Positionen beschränkt -1 : Linearachse X auf negative Positionen beschränkt
	[2..16] nicht relevant
1030 00160	Epsilon-Umgebungen MP-Name: EpsAxTrafo SD-Elem. : EpsilonRanges
	0.01

Tab. 11-108: Transformations-relevante Parameter

Aktivieren

Ist die Stirnseiten-Transformation konfiguriert per

- Maschinenparameter, dann wird die Transformation im NC-Programm aktiviert mit (i=1,..20):

COORD(<i>)

- Systemdatum, dann wird die Transformation im NC-Programm aktiviert mit (Typ "SysAxTrafo_t"):

COORD(<SD.MtbAxTrafo>)**Programmbeispiel:**

Die Stirnseiten-Transformation ist im 4. Maschinenparametersatz definiert. Am Programmstart ist keine Achstransformation aktiv:

Transformation aktivieren

```

N100 G1 F1000 XA10 ZA10 C30 ;the axes XA,ZA,C are traversed
N110 COORD(4) ;Switching on the axis transformation
N130 x20 y30 ;the coordinates x,y,phi are traversed
N140 x10 z10 ;the coordinates x,z,theta are traversed
      :
N999 COORD(0) ;Switching-off the axis transformation
      ;Axes are again programmed from here

```

Deaktivieren**COORD(0) bzw. COORD(0,2)**

schaltet die Achstransformation im NC-Programm aus.

11.23 3-Achs-Transformation LLR-kartesisch

11.23.1 Beschreibung

Funktion

Name der Transformation	Transformationstyp	Wirkstelle
LLR - kartesisch 3-Achs-Transformation	3021005	AT1
LLR - kartesisch 3-Achs-Transformation	3021004	AT2

Die Kinematik der 3-Achs-Transformation LLR besteht aus zwei Linearachsen L_1 , L_2 und einer rotatorischen Achse R_1 . Die Linearachsen L_1 und L_2 stehen senkrecht aufeinander. Die Rotationsachse R_1 dreht das Werkzeug über einen Ausleger um die L_2 -Achse und verschiebt es damit auch in der Raumrichtung, in der es mit L_1 und L_2 allein nicht verschoben werden könnte. Programmiert werden die drei Raumkoordinaten des Werkzeugreferenzpunkts (Spindelnase) oder bei aktiver Werkzeugkorrektur G47(XTR,YTR,ZTR) die der Werkzeugspitze (Tool Center Point, TCP). Da sich die Orientierung des Werkzeugs bei der R_1 -Drehung ändert, kommt die Achstransformation nur für Werkzeuge in Frage, bei denen es nicht auf die Orientierung ankommt.

Die Transformation kann sowohl an Wirkstelle 1 als auch an Wirkstelle 2 zum Einsatz kommen.

Als Achstransformation (AT) an Wirkstelle 1 (AT1) wirkt sie zwischen dem Achs-Koordinatensystem ACS = (L_1, L_2, R_1) und dem Maschinen-Koordinatensystem MCS = (X,Y,Z).

- Programmiert werden die MCS-Eingangskordinaten X, Y, Z
- Berechnet werden die ACS-Ausgangskordinaten L_1 , L_2 , R_1

Als Achstransformation an Wirkstelle 2 (AT2) wirkt sie zwischen dem Maschinen-Koordinatensystem MCS = (L_1, L_2, R_1) und dem Basis-Koordinatensystem BCS = (x,y,z).

- Programmiert werden die BCS-Eingangskordinaten x, y, z
- Berechnet werden die MCS-Ausgangskordinaten L_1 , L_2 , R_1

Die Wirkstelle wird im Maschinenparameter 103000110 "Transformationstyp" festgelegt:

Wirkstelle 1: MP 103000110 = 3021005

Wirkstelle 2: MP 103000110 = 3021004

Im Maschinenparameter 103000120 werden die "Systemachsen/-koordinaten der Transformation" festgelegt:

Systemachsnummer der L_1 -Achse: MP 103000120 (1)

Systemachsnummer der L_2 -Achse: MP 103000120 (2)

Systemachsnummer der R_1 -Achse: MP 103000120 (3)

Die restlichen Einträge MP 103000120 (4..8) sind nicht relevant.

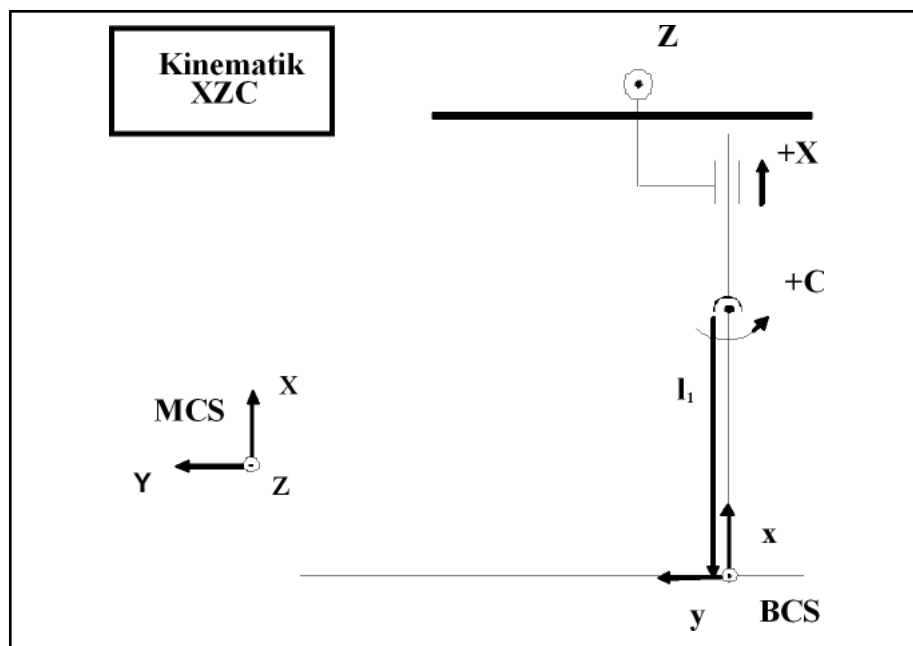


Abb. 11-98: Kinematik XZC

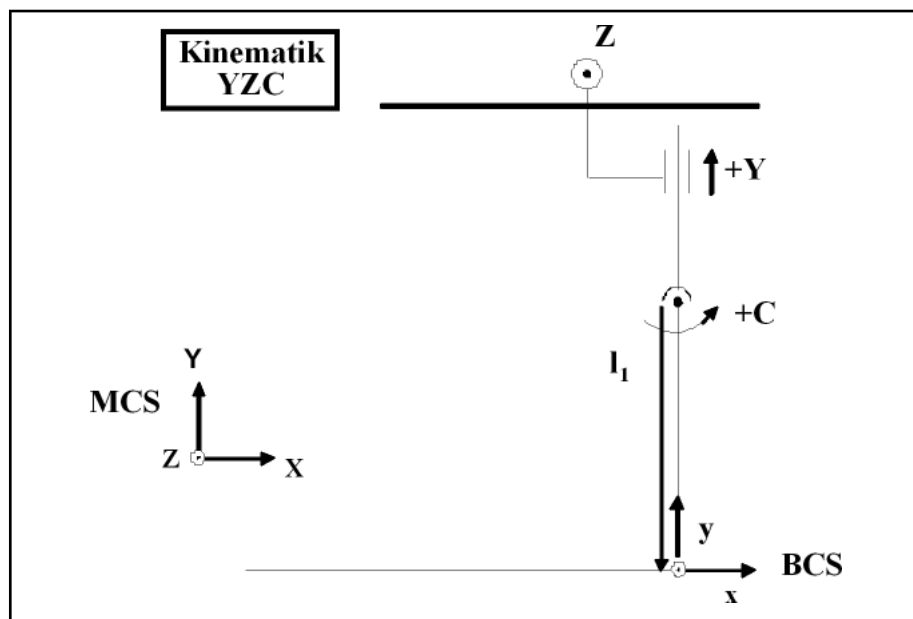


Abb. 11-99: Kinematik YZC

Varianten

Durch den Maschinenparameter 103000125 "Achsklassifikationen der Transformationsachsen" wird für jede beteiligte Achse festgelegt, welche Bedeutung sie für die AT haben soll:

Die Achse, die für die AT die Bedeutung X hat, legt die x/X-Richtung des Eingangskoordinatensystems fest.

Die Achse, die für die AT die Bedeutung Y hat, legt die y/Y-Richtung des Eingangskoordinatensystems fest.

Die Achse, die für die AT die Bedeutung Z hat, legt die z/Z-Richtung des Eingangskoordinatensystems fest.

Die nicht auf diese Weise festgelegte Richtung des Eingangskordinatensystems steht senkrecht zu den beiden anderen, so dass sich ein rechtshändiges Eingangskordinatensystem ergibt.

Kinematik XZC:

Die L_1 -Achse hat für die AT die Bedeutung X: MP 103000125 (1) = 1

Die L_2 -Achse hat für die AT die Bedeutung Z: MP 103000125 (2) = 3

Die R_1 -Achse hat für die AT die Bedeutung C: MP 103000125 (3) = 300

Damit legen bei dieser Kinematik die L_1 -Achse die x/X-Richtung und die L_2 -Achse die z/Z-Richtung des Eingangskordinatensystems fest. Die y/Y-Richtung des Eingangskordinatensystems steht senkrecht zu den beiden anderen, so dass sich ein rechtshändiges Koordinatensystem ergibt. [Abb. 11-98 "Kinematik XZC" auf Seite 390](#) zeigt diese Kinematik für Wirkstelle 2, also mit dem BCS als Eingangskordinatensystem.

Kinematik YZC:

Die L_1 -Achse hat für die AT die Bedeutung Y: MP 103000125 (1) = 2

Die L_2 -Achse hat für die AT die Bedeutung Z: MP 103000125 (2) = 3

Die R_1 -Achse hat für die AT die Bedeutung C: MP 103000125 (3) = 300

Damit legen bei dieser Kinematik die L_1 -Achse die y/Y-Richtung und die L_2 -Achse die z/Z-Richtung des Eingangskordinatensystems fest. Die x/X-Richtung des Eingangskordinatensystems steht senkrecht zu den beiden anderen, so dass sich ein rechtshändiges Koordinatensystem ergibt. [Abb. 11-99 "Kinematik YZC" auf Seite 390](#) zeigt diese Kinematik für Wirkstelle 2, also mit dem BCS als Eingangskordinatensystem.

Kinematik YXA:

Die L_1 -Achse hat für die AT die Bedeutung Y: MP 103000125 (1) = 2

Die L_2 -Achse hat für die AT die Bedeutung X: MP 103000125 (2) = 1

Die R_1 -Achse hat für die AT die Bedeutung A: MP 103000125 (3) = 100

[Abb. 11-98 "Kinematik XZC" auf Seite 390](#) zeigt diese Kinematik, wenn dort X durch Y, Z durch X, C durch A, x durch y, y durch z ersetzt wird, an Wirkstelle 2

Kinematik ZXA:

Die L_1 -Achse hat für die AT die Bedeutung Z: MP 103000125 (1) = 3

Die L_2 -Achse hat für die AT die Bedeutung X: MP 103000125 (2) = 1

Die R_1 -Achse hat für die AT die Bedeutung A: MP 103000125 (3) = 100

[Abb. 11-99 "Kinematik YZC" auf Seite 390](#) zeigt diese Kinematik, wenn dort Y durch Z, Z durch X, C durch A, y durch z, x durch y ersetzt wird, an Wirkstelle 2.

Kinematik ZYB:

Die L_1 -Achse hat für die AT die Bedeutung Z: MP 103000125 (1) = 3

Die L_2 -Achse hat für die AT die Bedeutung Y: MP 103000125 (2) = 2

Die R_1 -Achse hat für die AT die Bedeutung B: MP 103000125 (3) = 200

[Abb. 11-98 "Kinematik XZC" auf Seite 390](#) zeigt diese Kinematik, wenn dort X durch Z, Z durch Y, C durch B, x durch z, y durch x ersetzt wird, an Wirkstelle 2.

Kinematik XYB:

Die L_1 -Achse hat für die AT die Bedeutung X: MP 103000125 (1) = 1

Die L_2 -Achse hat für die AT die Bedeutung Y: MP 103000125 (2) = 2

Die R_1 -Achse hat für die AT die Bedeutung B: MP 103000125 (3) = 200

Abb. 11-99 "Kinematik YZC" auf Seite 390 zeigt diese Kinematik, wenn dort Y durch X, Z durch Y, C durch B, y durch x, x durch z ersetzt wird, an Wirkstelle 2.

Die restlichen Einträge MP 103000125 (4..8) sind nicht relevant.

Für jeden Eintrag im Maschinenparameter 103000125 ist auch ein negatives Vorzeichen zulässig. In diesem Fall wird die entsprechende Achse für die AT in umgekehrter Richtung berücksichtigt, also beispielsweise:

Kinematik YZC:

Die - L_1 -Achse hat für die AT die Bedeutung Y: MP 103000125 (1) = - 2

Die L_2 -Achse hat für die AT die Bedeutung Z: MP 103000125 (2) = 3

Die - R_1 -Achse hat für die AT die Bedeutung A: MP 103000125 (3) = - 300

Referenzstellung

Abb. 11-98 "Kinematik XZC" auf Seite 390 und Abb. 11-99 "Kinematik YZC" auf Seite 390 zeigen die Achstransformationen jeweils in Referenzstellung.

In dieser Stellung steht der Werkzeugreferenzpunkt im Nullpunkt des Eingangskoordinatensystems und in dieser Stellung sind die Ausgangskoordinatenwerte abzulesen und als "Achspositionen der Referenzstellung" einzutragen in den Maschinenparameter 103000130 (1) für die L_1 -Achse, 103000130 (2) für die L_2 -Achse und 103000130 (3) für die R_1 -Achse.

Der Abstand zwischen R_1 -Achse und Werkzeugreferenzpunkt wird für die Kinematik XZC festgelegt durch MP 103000140 (1), weil der zugehörige Abstandsvektor von der R_1 -Achse zum Werkzeugreferenzpunkt bei Referenzstellung (Abb. 11-98 "Kinematik XZC" auf Seite 390) parallel zur der x/X-Richtung des Eingangskoordinatensystems ausgerichtet ist. MP 103000140 (1) kann sowohl positiv, als auch als negativ gewählt werden.

Im positiven Fall zeigt der Abstandsvektor bei Referenzstellung in die x/X-Richtung des Eingangskoordinatensystems, im negativen Fall in die -x/X-Richtung.

Somit zeigt Abb. 11-98 "Kinematik XZC" auf Seite 390 den Fall MP 103000125 (1) > 0 und MP 103000140 (1) < 0.

Entsprechend wird dieser Abstand für die Kinematik YZC festgelegt durch MP 103000140 (2) (Abb. 11-99 "Kinematik YZC" auf Seite 390), für YXA durch MP 103000140 (2), für ZXA durch MP 103000140 (3), für ZYB durch MP 103000140 (3) und für XYB durch MP 103000140 (1).

Alle anderen Einträge in MP 103000140 sind nicht relevant.

Der Maschinenparameter 103000160 ist ebenfalls nicht relevant.

Besonderheiten

Mehrdeutige Achspositionen

Gibt man eine Position im Eingangskoordinatensystem vor, so kann diese Stellung immer durch zwei unterschiedliche Positionen im Ausgangskoordinatensystem erreicht werden. Diese beiden Achsstellungen bezeichnet man als Transformations-Zweige. Der Transformations-Zweig ist durch den Wert der R_1 -Achse festgelegt.

Werkzeugkorrektur

Werkzeugkorrekturen werden nur an Wirkstelle 2 unterstützt, also nur für die AT 3021004.

Werden Werkzeugkorrekturen mit G47(XTR,YTR,ZTR) aktiviert, wirken diese im Werkzeugkoordinatensystem (TCS). Das TCS fällt bei Referenzstellung mit dem BCS zusammen. Es dreht sich jedoch mit dem zuvor erwähnten Abstandsvektor von der R₁-Achse zum Werkzeugreferenzpunkt mit.

Relevante NC-Funktionen

COORD(...)	Die 3-Achs-Transformation LLR wird aktiviert. Dies kann wahlweise per Maschinenparametersatz (i=1..20) oder per Mtb-Systemdatum (vom Typ SysAxTrafo_t) erfolgen.
COORD(0,2) - oder - COORD(0,1)	Die aktive Transformation wird deaktiviert.

Tab. 11-109: Relevante NC-Funktionen

Relevante Maschinenparameter

In den Maschinenparametern können bis zu 20 unterschiedliche Achstransformationen per nachfolgendem Maschinenparametersatz deklariert werden.

1030 00110	Definiert den Transformationstyp.
1030 00120	Definiert die an der Transformation beteiligten Systemachsen/-koordinaten der Transformation.
1030 00125	Definiert die Achsklassifikation der Transformationsachsen.
1030 00130	Definiert die Achspositionen der Referenzstellung.
1030 00140	Definiert die Längen- und Winkelparameter.
1030 00160	Epsilon-Umgebungen

Tab. 11-110: Relevante Maschinenparameter

Relevante Systemdaten

Die Achstransformation kann alternativ zu den Maschinenparameter auch per Systemdatum definiert werden. Dazu muss der Benutzer eine Systemvariable vom Typ "SysAxTrafo_t" anlegen (siehe Dokumentation "IndraMotion MTX 14VRS Maschinenparameter", Kapitel "Systemdaten"). Der Strukturtyp "SysAxTrafo_t" ist in der XML-Schemadateien "/feprom/schemas/sdaxtrf.xsd" steuerungsinternen definiert und enthält, analog zu den relevanten Maschinenparametern, die nachfolgenden transformationsrelevanten Strukturelemente.

Strukturelement	Beschreibung
Type	Definiert den Transformationstyp.
AxisAssignment	Definiert die an der Transformation beteiligten Systemachsen/-koordinaten der Transformation.
AxisClassification	Definiert die Achsklassifikation der Transformationsachsen.
AxZeroPos	Definiert die Achspositionen der Referenzstellung.
LenParam	Definiert die Längen- und Winkelparameter.
EpsilonRanges	Epsilon-Umgebungen

Tab. 11-111: Relevante Systemdaten

11.23.2 Handlungsanweisung

Applizieren Die Achstransformation kann über die relevanten Maschinenparameter appliziert werden. Dazu muss die Funktion Achstransformation über den Parameter **tra** "Achstransformationen" im Setup (SUP) aktiviert werden. Der Pfad des MP-Namens ist immer "TRA/AxTrafo[1..20]/...".

Alternativ dazu kann die Achstransformation über ein benutzerdefiniertes Systemdatum vom Typ SysAxTrafo_t appliziert werden.

Die beiden Varianten sind inhaltlich identisch.

Die Achspositionen der Referenzstellung müssen ermittelt werden.

In der nachfolgenden Tabelle sind jeweils die äquivalenten MP-Nummern, MP-Name und SD-Elemente angegeben. Um eine Transformation zu applizieren, muss eine der Varianten entsprechend besetzt werden.

Transformations-relevante Parameter

MP-Nummer	Beschreibung	
1030 00110	Transformationstyp	
	MP-Name: TrafoType	
	SD-Elem. : Type	
	3021005	Wirkstelle AT1
1030 00120	3021004	Wirkstelle AT2
	Systemachsen/-koordinaten der Transformation	
	MP-Name: FwdInCoordIndTrafo[1..8]	
	SD-Elem. : AxisAssignment[1..8]	
	[1]	Systemachsnummer von L ₁
	[2]	Systemachsnummer von L ₂
	[3]	Systemachsnummer von R ₁
	[4..8]	nicht relevant

MP-Nummer	Beschreibung																				
1030 00125	Achsklassifikationen der Transformationsachsen MP-Name: CoordClass[1..8] MP-Name: CoordDir[1..8] SD-Elem. : AxisClassification[1..8] Definiert die an der Transformation beteiligten Achsen inklusive Bewegungsrichtung. - MP-Nr. oder SD-Element: Wert ">0" bedeutet: Die Achse verfährt/dreht positiv im Sinne der obigen Abbildungen. Wert "<0" bedeutet: Die Achse verfährt/dreht negativ im Sinne der obigen Abbildungen. <i>Mögliche Werte sind:</i> +/-1 = X-Achse +/-2 = Y-Achse +/-3 = Z-Achse +/-100 = A-Achse +/-200 = B-Achse +/-300 = C-Achse - MP-Name: <i>Mögliche Werte für CoordClass sind:</i> - X,Y,Z (linear) - A,B,C (rotatorisch) <i>Mögliche Werte für CoordDir sind:</i> - positiv (bzgl. Abb. "Referenzstellung") - negativ (bzgl. Abb. "Referenzstellung") Ist ein Parameter unbesetzt, gelten folgende Default-Einstellungen: <table><tr><th></th><th>MP-Nr / SD-Elem.:</th><th>CoordClass:</th><th>CoordDir:</th></tr><tr><td>[1]</td><td>1</td><td>X</td><td>positiv</td></tr><tr><td>[2]</td><td>3</td><td>Z</td><td>positiv</td></tr><tr><td>[3]</td><td>300</td><td>C</td><td>positiv</td></tr><tr><td>[4..8]</td><td>nicht relevant</td><td>0</td><td>positiv</td></tr></table>		MP-Nr / SD-Elem.:	CoordClass:	CoordDir:	[1]	1	X	positiv	[2]	3	Z	positiv	[3]	300	C	positiv	[4..8]	nicht relevant	0	positiv
	MP-Nr / SD-Elem.:	CoordClass:	CoordDir:																		
[1]	1	X	positiv																		
[2]	3	Z	positiv																		
[3]	300	C	positiv																		
[4..8]	nicht relevant	0	positiv																		

Achs-Transformation

MP-Nummer	Beschreibung
1030 00130	Achspositionen der Referenzstellung MP-Name: RefPosTrafo[1..8] SD-Elem. : AxZeroPos[1..8]
	[1] L_1^0
	[2] L_2^0
	[3] R_1^0
	[4..8] nicht relevant
1030 00140	Längen- und Winkelparameter MP-Name: JointParTrafo[1..16] SD-Elem. : LenParam[1..16] Definiert den Längenvektoren l_i in [mm].
	[1] Nur relevant für die Kinematiken XZC und XYB
	[2] Nur relevant für die Kinematiken YZC und YXA
	[3] Nur relevant für die Kinematiken ZXA und ZYB
	[4..16] Grundsätzlich nicht relevant
1030 00160	Epsilon-Umgebungen MP-Name: EpsAxTrafo SD-Elem. : EpsilonRanges
	nicht relevant

Tab. 11-112: Transformations-relevante Parameter

Aktivieren Ist die 3-Achs-Transformation konfiguriert per

- Maschinenparameter, dann wird die Transformation im NC-Programm aktiviert mit (i=1,..20):

COORD(<i>)

- Systemdatum, dann wird die Transformation im NC-Programm aktiviert mit (Typ "SysAxTrafo_t"):

COORD(<SD.MtbAxTrafo>)

Programmbeispiel:

Die 3-Achs-Transformation ist im 4. Maschinenparametersatz definiert. Am Programmanfang ist keine Achstransformation aktiv:

Einstellung der Maschinenparameter für ein Programm in Kanal 1:

1003 00 001 (1..8) XS1 XS2 XS3 XS4 XS5 XS6 CS1 CS2

1003 00 002 (1..8) 1 1 1 1 1 1 1 1

1030 00 110 (4) 3021004

1030 00 120 (4) 5 3 8

1030 00 125 (4) 3 2 200

1030 00 130 (4) 0 0 0

1030 00 140 (4) 0 0 90

1030 00 160 (4) 0.1

7010 00 010 (1) X1 X2 X3 X4 X5 X6 C1 C2

7080 00 010 (1) x y z phi theta psi

Programm:

```

010  A!  = 100 * SQRT(3)
020  A3! = A! / 3
030  AM! = -A!           :  A3M! = -A3!
N040  Coord(0,2)
N050  Coord(0,1)
N060  G53 G48 ED0
N070  X1=0 X2=0 X3=0 X4=0 X5=0 X6=0 C1=0 C2=0 G1 F10000
080  DCT(1,5,0) = A3!
090  DCT(2,5,0) = -30
100  DCT(3,5,0) = -10
N110  Coord(4)
;
N120  G47(XTR,YTR,ZTR) ED5 ; =>   XS5 XS3 CS2
N130  x[A3M!] y0 z10      ; =>   0   0   0
N140  x[A3!]              ; =>   0  -30  0
N150  G48 ED0
N160  Coord(0)
      M30

```

Deaktivieren **COORD(0) bzw. COORD(0,2)**

schaltet die Achstransformation im NC-Programm aus.

11.24 Polarkinematik

11.24.1 Beschreibung

Funktion

Name der Transformation	Transformationstyp	Wirkstelle
Polarkinematik		
2-Achs-Variante	2011003	AT2
3-Achs-Variante	3021006	AT2

Die Maschinenkonfiguration der Polarkinematik besteht aus einer Rundachse C, auf der eine Linearachse X montiert ist. Die Linearachse wird von der Rundachse mitgedreht. Wahlweise kann noch eine Vertikalachse Z vorhanden sein (vergl. [Abb. 11-100 "2-/3-Achs-Polarkinematik \(Seitenansicht\)" auf Seite 398](#), [Abb. 11-101 "2-/3-Achs-Polarkinematik \(Draufsicht\)" auf Seite 398](#)). Die Transformation kommt an der Wirkstelle 2 zum Einsatz, d. h. sie wirkt zwischen dem Maschinen-Koordinatensystem MCS = (C,X (Z)) und dem Basis-Koordinatensystem BCS = (x,y (z)).

- Programmiert werden die BCS-Eingangsgrößen x,y (z)
- Berechnet werden die MCS-Ausgangsgrößen C,X (Z)

Achsen-Transformation

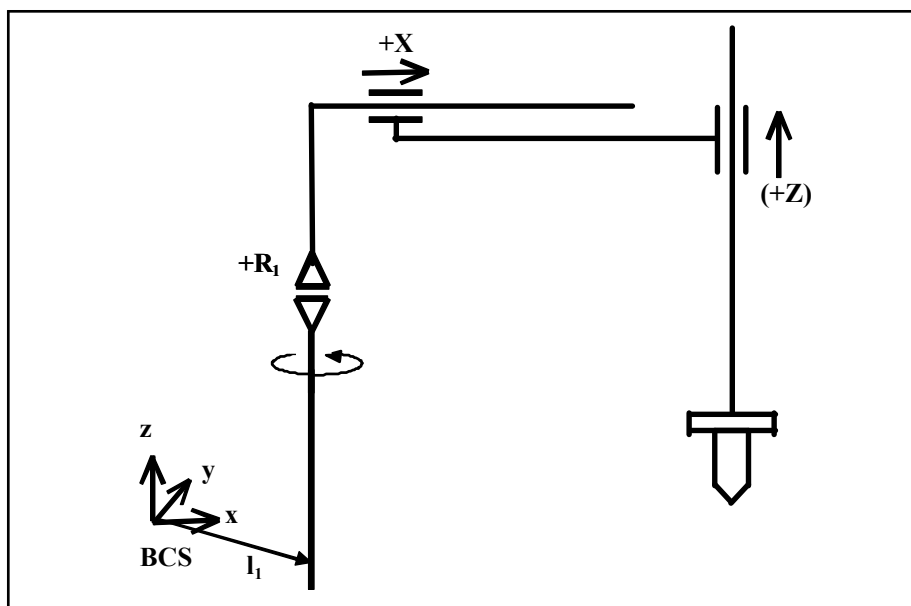


Abb. 11-100: 2-/3-Achs-Polarkinematik (Seitenansicht)

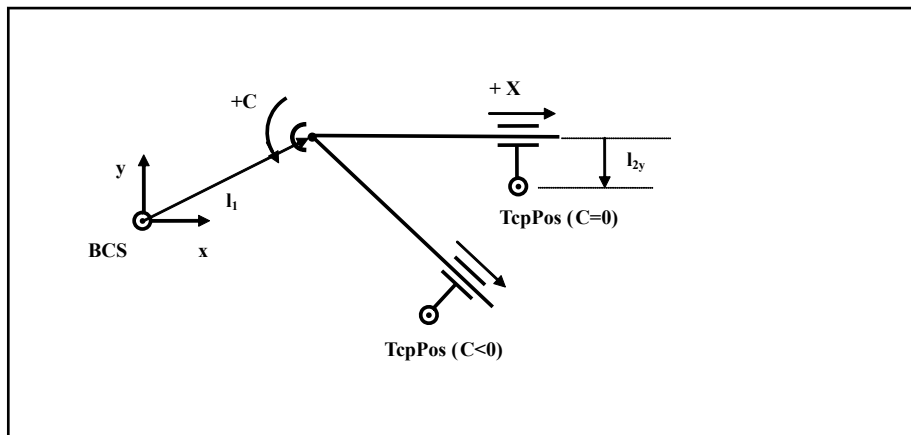


Abb. 11-101: 2-/3-Achs-Polarkinematik (Draufsicht)

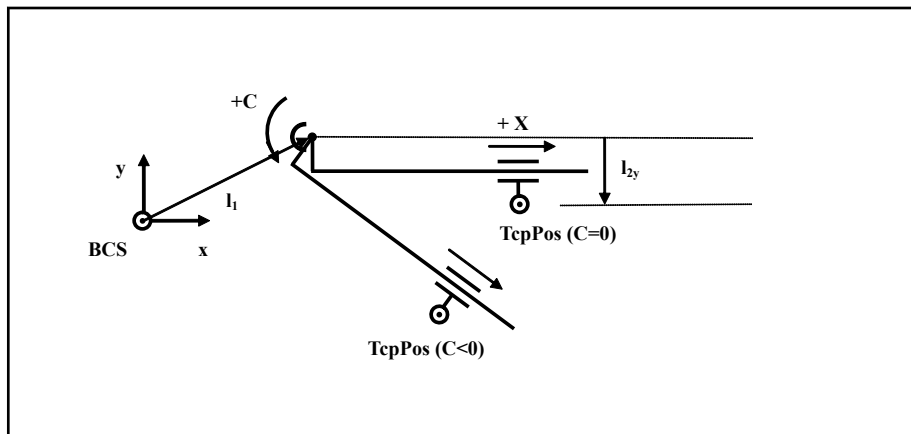


Abb. 11-102: 2-/3-Achs-Polarkinematik (Draufsicht)

Der BCS-Nullpunkt kann sich an einer beliebigen Position der Arbeitsfläche befinden (siehe [Abb. 11-100 "2-/3-Achs-Polarkinematik \(Seitenansicht\)"](#) auf [Seite 398](#)).

Die Vektoren I_1 und I_2 definieren die Geometrie der Polarkinematik. I_1 zeigt vom BCS-Nullpunkt zur Rotationsachse C. I_2 beschreibt die Außermittigkeit des ToolCenterPoint (TCP) zur X-Achse.

Referenzstellung

- In der Referenzstellung ($C=0$) ist die X-Achse parallel zur x-Koordinate des BCS ausgerichtet.
- Der Nullpunkt der X-Achse liegt in der Rundachse C.
- Das Werkzeug zeigt in negative Z-Richtung (falls vorhanden), d. h. das Werkzeug zeigt senkrecht nach unten.
- Das BCS liegt parallel zum MCS.

Typischerweise liegt der BCS-Nullpunkt in der Rotationsachse R_1 .

Besonderheiten

Mehrdeutige Achspositionen

Gibt man eine TCP-Position (x,y (z)) vor, so kann diese Stellung immer durch zwei unterschiedliche Achspositionen erzielt werden. Diese bezeichnet man als Trafo-Zweige. Der Trafo-Zweig ist i.w. durch das Vorzeichen der X-Achse festgelegt:

- Zweig +1: $X \geq 0$
- Zweig -1: $X < 0$
- Das BCS liegt parallel zum MCS.

Welche der beiden Trafo-Zweige benutzt wird, entscheidet sich beim Einschalten der Achstransformation durch die vorgegebene Stellung der X-Achse. Ein Übergang von Zweig +1 nach -1 und umgekehrt ist danach nur beim Durchfahren der Singularität bei $X=0$ möglich.

Werkzeugkorrektur

Die Achstransformation unterstützt eine Werkzeugkorrektur im Werkzeugkoordinatensystem TCS (Tool-Coordinate-System). In der Referenzstellung stimmen BCS und TCS_0 (Spindelnase) überein. Der Werkzeugkorrekturvektor I_1 zeigt vom korrigierten Werkzeugmittelpunkt TCP_1 zum unkorrigierten TCP_0 . Er setzt sich aus den Längenkorrekturen L1, L2 und L3 aus der D-Korrekturtable zusammen.

Relevante NC-Funktionen

COORD(...)	Die Achs-Transformation der Polarkinematik wird aktiviert. Dies kann wahlweise per Maschinenparametersatz ($i=1..20$) oder per Mtb-Systemdatum (vom Typ SysAxTrafo_t) erfolgen.
COORD(0) - oder - COORD(0,2)	Die aktive Transformation wird deaktiviert.

Tab. 11-113: Relevante NC-Funktionen

Relevante Maschinenparameter

In den Maschinenparametern können bis zu 20 unterschiedliche Achstransformationen per nachfolgendem Maschinenparametersatz deklariert werden.

1030 00110	Definiert den Transformationstyp.
1030 00120	Definiert die an der Transformation beteiligten Systemachsen/-koordinaten der Transformation.

1030 00125	Definiert die Achsklassifikation der Transformationsachsen.
1030 00130	Definiert die Achspositionen der Referenzstellung.
1030 00140	Definiert die Längen- und Winkelparameter.
1030 00160	Epsilon-Umgebungen

Tab. 11-114: Relevante Maschinenparameter

Relevante Systemdaten

Die Achstransformation kann alternativ zu den Maschinenparameter auch per Systemdatum definiert werden. Dazu muss der Benutzer eine Systemvariable vom Typ "SysAxTrafo_t" anlegen (siehe Dokumentation "MTX Maschinenparameter", Kapitel "Systemdaten"). Der Strukturtyp "SysAxTrafo_t" ist in der XML-Schemadatei "/feprom/schemas/sdaxtrf.xsd" steuerungsinternen definiert und enthält, analog zu den relevanten Maschinenparametern, die nachfolgenden traforelevanten Strukturelemente.

Strukturelement	Beschreibung
Type	Definiert den Transformationstyp.
AxisAssignment	Definiert die an der Transformation beteiligten Systemachsen/-koordinaten der Transformation.
AxisClassification	Definiert die Achsklassifikation der Transformationsachsen.
AxZeroPos	Definiert die Achspositionen der Referenzstellung.
LenParam	Definiert die Längen- und Winkelparameter.
EpsilonRanges	Epsilon-Umgebungen

Tab. 11-115: Relevante Systemdaten

11.24.2 Handlungsanweisung: Polarkinematik

Applizieren

Applizieren Die Achstransformation kann über die relevanten Maschinenparameter appliziert werden. Dazu muss die Funktion Achstransformation über den Parameter **tra** "Achstransformationen" im Setup (SUP) aktiviert werden. Der Pfad des MP-Namens ist immer "TRA/AxTrafo[1..20]/...".

Alternativ dazu kann die Achstrafo über ein benutzerdefiniertes Systemdatum vom Typ SysAxTrafo_t appliziert werden.

Die beiden Varianten sind inhaltlich identisch.

Die Achspositionen der Referenzstellung müssen ermittelt werden.

In der nachfolgenden Tabelle sind jeweils die äquivalenten MP-Nummern, MP-Name und SD-Elemente angegeben. Um eine Transformation zu applizieren, muss eine der Varianten entsprechend besetzt werden.

Trafo-relevante Parameter

MP-Nummer	Beschreibung	
1030 00110	Transformationstyp	
	MP-Name: TrafoType	
	SD-Elem. : Type	
	2011003	2-Achs-Variante
	3021006	3-Achs-Variante

MP-Nummer	Beschreibung	
1030 00120	Systemachsen/-koordinaten der Transformation	
	MP-Name: FwdInCoordIndTrafo[1..8]	
	SD-Elem. : AxisAssignment[1..8]	
	[1]	Systemachsnummer von C
	[2]	Systemachsnummer von X
	[3]	Systemachsnummer von Z (falls vorhanden)
	[4..8]	nicht relevant

Achs-Transformation

MP-Nummer	Beschreibung																					
1030 00125	Achsklassifikationen der Transformationsachsen																					
	MP-Name: CoordClass[1..8]																					
	MP-Name: CoordDir[1..8]																					
	SD-Elem. : AxisClassification[1..8]																					
	Definiert die an der Transformation beteiligten Achsen inklusive Bewegungsrichtung. - MP-Nr. oder SD-Element: Wert ">0" bedeutet: Die Achse verfährt/dreht positiv im Sinne der obigen Abbildungen. Wert "<0" bedeutet: Die Achse verfährt/dreht negativ im Sinne der obigen Abbildungen. <i>Mögliche Werte sind:</i> +/-1 = X-Achse +/-2 = Y-Achse +/-3 = Z-Achse +/-100 = A-Achse +/-200 = B-Achse +/-300 = C-Achse - MP-Name: <i>Mögliche Werte für CoordClass sind:</i> - X, Y, Z (linear) - A, B, C (rotatorisch) <i>Mögliche Werte für CoordDir sind:</i> - positiv (bzgl. Abb. "Referenzstellung") - negativ (bzgl. Abb. "Referenzstellung") Ist ein Parameter unbesetzt, gelten folgende Default-Einstellungen: <table> <tr> <th></th><th>MP-Nr / SD-Elem.:</th><th>CoordClass:</th><th>CoordDir:</th></tr> <tr> <td>[1]</td><td>300</td><td>C</td><td>positiv</td></tr> <tr> <td>[2]</td><td>1</td><td>X</td><td>positiv</td></tr> <tr> <td>[3]</td><td>3 (falls vorhanden)</td><td>Z</td><td>positiv</td></tr> <tr> <td>[4..8]</td><td>nicht relevant</td><td>0</td><td>positiv</td></tr> </table>				MP-Nr / SD-Elem.:	CoordClass:	CoordDir:	[1]	300	C	positiv	[2]	1	X	positiv	[3]	3 (falls vorhanden)	Z	positiv	[4..8]	nicht relevant	0
	MP-Nr / SD-Elem.:	CoordClass:	CoordDir:																			
[1]	300	C	positiv																			
[2]	1	X	positiv																			
[3]	3 (falls vorhanden)	Z	positiv																			
[4..8]	nicht relevant	0	positiv																			

MP-Nummer	Beschreibung
1030 00130	Achspositionen der Referenzstellung MP-Name: RefPosTrafo[1..8] SD-Elem. : AxZeroPos[1..8]
	[1] C^0
	[2] X^0
	[3] Z^0 (falls vorhanden)
	[4..8] nicht relevant
1030 00140	Längen- und Winkelparameter MP-Name: JointParTrafo[1..16] SD-Elem. : LenParam[1..16] Definiert die Längenvektoren l_1, l_2 in [mm].
	[1] l_{1x}
	[2] l_{1y}
	[3] l_{1z}
	[4] l_{2x}
	[5] l_{2y}
	[6] l_{2z}
	[7..16] nicht relevant
1030 00160	Epsilon-Umgebungen MP-Name: EpsAxTrafo SD-Elem. : EpsilonRanges
	0.01

Tab. 11-116: Trafo-relevante Parameter

Aktivieren

Aktivieren *Ist die 5-Achs-Trafo konfiguriert per*

- Maschinenparameter, dann wird die Transformation im NC-Programm aktiviert mit (i=1,..20):

COORD(<i>)

- Systemdatum, dann wird die Transformation im NC-Programm aktiviert mit (Typ "SysAxTrafo_t"):

COORD(<SD.MtbAxTrafo>)

Programmbeispiel:

Die 2-Achs-Polarkinematik ist im 4. Maschinenparametersatz definiert. Am Programmstart ist keine Achstrafo aktiv:

N01 G1 F1000 XA10 C30	;die Achsen XA, C werden verwendet
N02 COORD(4)	;Einschalten der 2-Achs-Trafo
N03 x20 y30	;die Koordinaten x, y werden verwendet
N04 x10 y100	;die Koordinaten x, y werden verwendet

N99 COORD(0)

;Ausschalten 2-Achs-Transformation

;ab hier werden wieder die Achsen programmiert

Deaktivieren

Deaktivieren

COORD(0) bzw. COORD(0,2)

schaltet die Achstransformation im NC-Programm aus.

11.25 Achs-Kinematikkalibrierung

11.25.1 Beschreibung

Grundlagen

Mit der Funktion Achs-Kinematikkalibrierung werden Konfigurationsparameter von Achs-Transformationen so optimiert, dass eine höhere Positionierungsgenauigkeit erreicht wird. Diese Ermittlung erfolgt automatisch mit Hilfe von Messzyklen.

Im Messzyklus werden bei aktiver Achs-Transformation mehrere Messpunkte angefahren und die Istwerte der Koordinaten mittels eines geeigneten Messsystems (z. B. Messtaster) gemessen. Die Messdaten bestehend aus einer Anzahl von Ist- und Soll-Achs-/Koordinatenwerten und werden je nach Messverfahren entweder direkt in einer Kalibrierdatei verrechnet, oder zuvor in einer Messzyklusdatei verarbeitet und anschließend in einer Kalibrierdatei verrechnet.

Die NC ermittelt dann aus den Messdaten die optimalen Konfigurationsparameter.

Optimierbare Konfigurationsparameter:

Eine Achs-Kinematik wird i. d. R. aus folgenden Konfigurationsparametern bestimmt:

MP 103000110:	Achs-Transformationstyp
MP 103000120:	Systemachsen der Transformation
MP 103000130:	Achspositionen der Referenzstellung
MP 103000140:	Längen- und Winkelparameter
MP 103000150:	Referenzlage des Werkzeugkoordinatensystems

Von den beiden fehlerbehafteten Parametern MP 103000130 und MP103000140 werden ausschließlich die in MP 103000140 abgelegten **Längen- und Winkelparameter** mit der Funktion Achs-Kinematikkalibrierung optimiert.

Eine Optimierung des MP 103000130 "Achspalten der Referenzstellung" ist nicht vorgesehen.

Auswahl der Achs-Transformation:

Die Achs-Kinematikkalibrierung kann bei allen Achs-Transformationen - sowohl AT1 als auch AT2 - angewendet werden.

Die Transformation zwischen dem Achskoordinatensystem ACS und dem kartesischen Maschinenkoordinatensystem MCS ist üblicherweise notwendig, wenn das ACS nicht kartesisch aufgebaut ist (typische Kinematiken für die Wirkstelle AT1: Stabkinematiken wie Bipod, SCARA und nichtorthogonale Linearachsen).

Das kartesische MCS, das mit der Achs-Transformation AT1 entsteht, wird bei Anwendung einer weiteren Transformationen AT2 (z. B. 5-Achs-Transformation) auf das Basiskoordinatensystem (BCS) transformiert.

Für die Achs-Kinematikkalibrierung bedeutet dies:

Wird der Messzyklus mit aktiver Achs-Transformation AT2 abgearbeitet, ist diese automatisch die zu kalibrierende Achs-Transformation.

Ist keine Achs-Transformation AT2 vorhanden, beziehen sich die Messdaten auf die Achs-Transformation AT1.

11.25.2 Messverfahren

Funktion

Bei den Messverfahren zur Achs-Kinematikkalibrierung werden zwei Verfahren zum Messen des TCP (tool center point) unterschieden:

- **Direktes Messen:**
Bei dieser Messung ist das Messergebnis unabhängig von den Konfigurationsparametern der aktuellen Achs-Transformation (z. B. Laser-Interferometrie).
- **Indirektes Messen:**
Der TCP (tool center point) wird indirekt gemessen. Bei dieser Messung (z. B. Vermessen einer Lochplatte mit Achsmesstaster) ist das Ergebnis parameterabhängig, da aus den getasteten Achspositionen per Vorwärtstransformation die zugehörigen Maschinen- oder Basiskoordinaten auf Grundlage des aktuellen Parametersatzes berechnet werden.

Direktes Messen: Einem Messpunkt i seien der vorgegebene Koordinatensollwert (exakt bekannte Koordinatenposition im WCS) $\vec{a}^e = (p_{i,1}^e, p_{i,2}^e, \dots, p_{i,c}^e)$ und der direkt gemessene WCS-Koordinaten-Istwert im WCS $\vec{a}^m = (p_{i,1}^m, p_{i,2}^m, \dots, p_{i,c}^m)$ zugeordnet.

Die Abweichung $|\vec{a}^e - \vec{a}^m|$ kommt durch einen ungenauen Parametersatz $\vec{\rho} = (P_1, P_2, \dots, P_n)$ (= Längen- und Winkelparameter, ...) in der aktuellen Achs-Transformation zustande.



Weitere, von der Achs-Transformation nicht korrigierbare Fehler (Spindelsteigungsfehler, krumme Achsführungen) werden hier nicht berücksichtigt.

Dem Koordinatenpunkt $\vec{a}^e$ zugeordnet ist ein Achspunkt $\vec{q}^e$ im ACS. Beide sind über die Vorwärtstransformation $\vec{a}^e = \vec{f}(\vec{q}^e, \vec{\rho})$ miteinander verknüpft (siehe dazu folgende Abb.).

Achs-Transformation

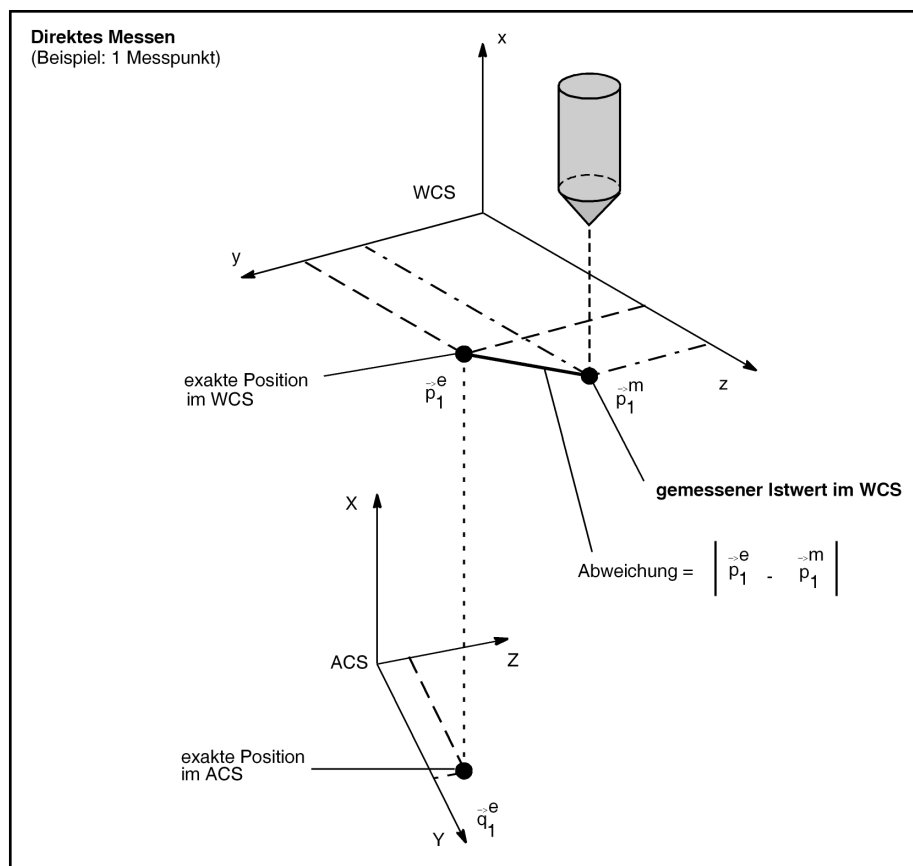


Abb. 11-103: Direktes Messen

Bei der Rechnung (iterierte lineare Ausgleichsrechnung) wird so bestimmt, dass die "quadratische Abweichung" (Gütefunktion) zwischen den Koordinaten-Ist- und -Sollwerten **minimal** wird.

$$E(\vec{P}) = \sum_{i=1}^k \left| \vec{p}_i - \vec{p}_i^e \right|^2 = \sum_{i=1}^k \left| \vec{p}_i - f(q, \vec{P}^e) \right|^2$$

"k" die Anzahl der Messpunkte.

Bei der Iteration sind die Koordinatenpunkte $\vec{p}_i^m$ und die Achspunkte $\vec{q}_i^e$ konstant. Variiert wird der Parametersatz $\vec{p}$.

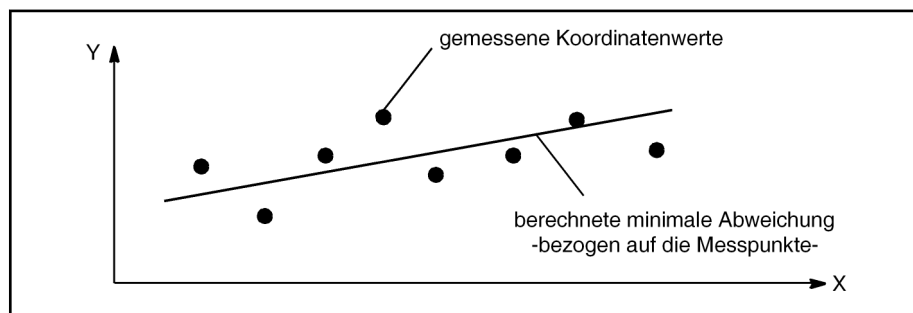


Abb. 11-104: Iteration

Indirektes Messen (Variante 1):

Die indirekte Messmethode misst exakt bekannte WCS-Positionen $\vec{p}^e$, z. B. Bohrungen auf einer Lochplatte (siehe [Kap. "Kalibrierzyklus " auf Seite 418](#)).

Ein Messtaster liefert beim Messen der Punkte zunächst die Achspunkte $\vec{q}^m$ im ACS, die noch auf Koordinatenpunkte mittels $\vec{p}^m = \vec{f}(\vec{q}^m, \vec{\rho})$ umgerechnet werden müssen. Die Gütefunktion ist in diesem Fall

$$E(\vec{P}) = \sum_{i=1}^k \left| \vec{p}^m_i - \vec{p}^e_i \right|^2 = \sum_{i=1}^k \left| \vec{p}^m_i - f(\vec{q}^m_i, \vec{P}) \right|^2$$

Ermittlung einer Bohrungsposition:

Eine Bohrung i wird durch Abtasten von drei Achspunkten $\vec{q}^{m,1}_i, \vec{q}^{m,2}_i, \dots, \vec{q}^{m,3}_i$ vermessen. Zur Bestimmung des Mittelpunktes $\vec{p}^m_i$ werden zunächst die drei Tastkoordinatenpunkte mittels $\vec{p}^{m,i}_j = \vec{f}(\vec{q}^{m,i}_j, \vec{\rho})$ bestimmt. Aus diesen kann dann der Kreismittelpunkt $\vec{p}^m_i$ berechnet werden, der seinerseits vom verwendeten Parametersatz $\vec{\rho}$ abhängt.

Achs-Transformation

Indirektes Messen

(Beispiel: 1 Messpunkt am Lochkreis)

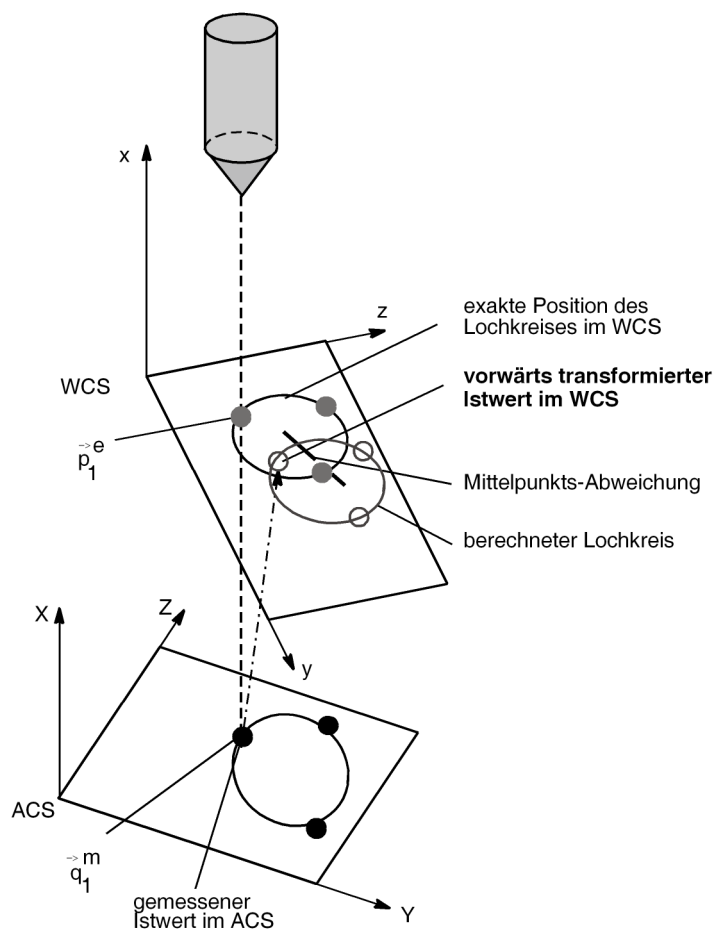


Abb. 11-105: Indirektes Messen

Indirektes Messen (Variante 2):

Die nachfolgend beschriebene Variante 2 der indirekten Messmethode sollte der Variante 1 vorgezogen werden, da sie bessere Konvergenzeigenschaften besitzt. Die Variante 2 basiert auf dem Antasten einer Messkugel. Das Vorhandensein dieser Messkugel ist zwingend erforderlich. Darüber hinaus müssen alle transformations-relevanten Achsen per Messtaster erfasst werden, d. h. alle transformations-relevanten Achsen sind an den Messtaster angeschlossen.

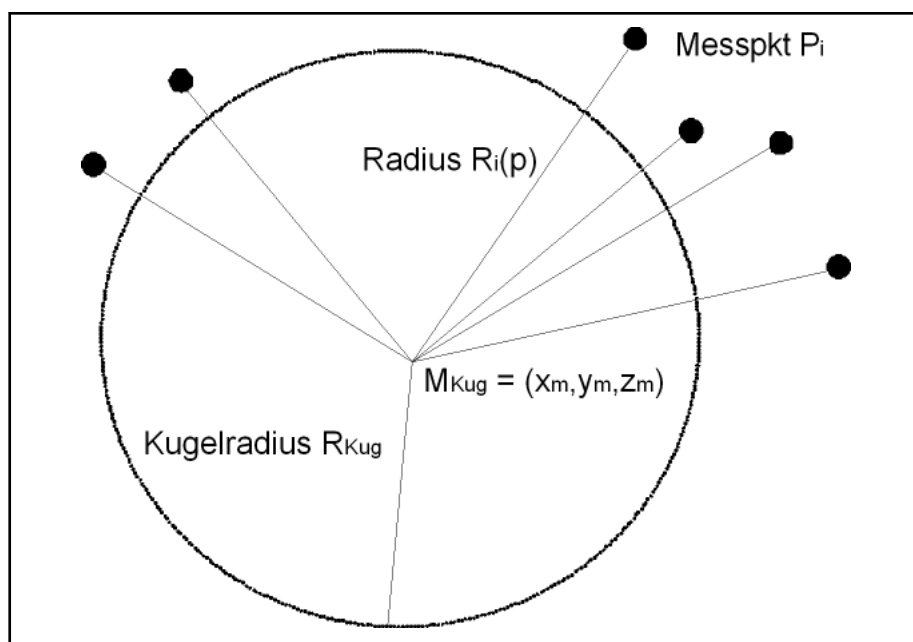


Abb. 11-106: Kalibrierung: Indirektes Verfahren mittels Messkugel

Es werden Messpositionen auf der Kugel berechnet. Diese werden mittels Messtaster angefahren. Das Anfahren muss radial erfolgen, um den Einfluss des Messtasterkugel-Radius zu eliminieren. Er wirkt dann immer in Werkzeug-z-Richtung. Beim Auslösen des Messtasters werden die Achspositionen

$\vec{q}_i$ abgespeichert. Daraus werden steuerungintern die zugehörigen Koordinatenpositionen $\vec{q}_i$ berechnet.

Relevante NC-Funktionen

Die folgenden NC-Funktionen erlauben es, sowohl für das direkte als auch indirekte Messverfahren eine Parameteroptimierung auf Basis eines einzigen Messzyklus durchzuführen:

ATCAL(..)	Berechnung eines optimalen Parametersatzes
ATGET(..)	Lesen der aktuell gültigen Parameter (aus MP in Systemdaten übernommen)
ATPUT(..)	Schreiben der optimierten Konfigurationsparameter in den MP 1030 00140
ATFWD(..)	Ausführen einer Vorwärtstransformation mit vorgegebenen Parametern (beliebiger Achspunkt (im ACS) mit vorgegebenen Konfigurationsparametern auf einen Koordinatenpunkt (im WCS) umrechnen.

Tab. 11-117: Relevante NC-Funktionen

NC-Befehle der Funktion "Kalibrieren von Achs-Kinematiken"

Es wird eine Reihe von NC-Befehlen angeboten, die es sowohl für das direkte als auch indirekte Messverfahren erlauben, eine Parameteroptimierung auf Basis eines einzigen Messzyklus durchzuführen. Struktur und Beispiele von Kalibrierzyklen sind in [Kap. "Kalibrierzyklus"](#) auf Seite 418 beschrieben.

Berechnung eines optimalen Parametersatzes:

Die Optimierungsrechnung der Konfigurationsparameter erfolgt mit dem NC-Befehl ATCAL (**A**xis**T**ransformation**C**ALibration)

```
ATCAL(<CalibFile>,<LenParamVar>,<LenParamMaskVar>
[,<ResultVar>][,n])
```

Die Parameter haben folgende Bedeutung:

CalibFile

Name der Kalibrierdatei, in der die Koordinatenwerte $\vec{p}_0^e$, $\vec{p}_0^m$ und gegebenenfalls die Achswerte $\vec{q}_i^e$ oder $\vec{q}_i^m$ abliegen. Der Dateiname kann mit oder ohne Pfadangabe übergeben werden. Bei fehlender Pfadangabe gilt der in den Maschinenparametern eingestellte Unterprogrammsuchpfad. Der Aufbau der Kalibrierdatei ist in [Kap. "Kalibrierzyklus" auf Seite 418](#) beschrieben.

LenParamVar

Gibt den Namen eines CPL-Variablen-Arrays an, in den die ATCAL-Logik die optimierten Längen- und Winkelparameter ablegt. Die Variable muss den CPL-Typ DOUBLE und die Dimension ≥ 16 haben. Die Werte haben die gleiche Einheit wie der Maschinenparameter 1030 00 40.

Beispiele:

- (i) Als permanente Variable (Eintrag in wmhperm.dat)
DEF DOUBLE @LENPARAM(16)
- (ii) Als globale Variable im Kalibrierzyklus
DIM #LENPARAM!(16)
- (iii) Als lokale Variable im Kalibrierzyklus
DIM LENPARAM!(16)

LenParamMaskVar

Gibt den Namen einer CPL-Variablen an, in der eine Bitmaske für die zu optimierenden Einzelparameter LENPARAM ablegt. Die Variable hat den CPL-Typ INT.

Beispiel:

```
LENPARAMASK% = 1+2+4+16: REM Parameter 1, 2, 3 und 5 optimieren
ATCAL("CalFile",LENPARAM!,LENPARAMASK%)
```

ResultVar

Diese Variable kann optional angegeben werden. Sie ist vom Typ DOUBLE und hat die Dimension ≥ 4 . Nach dem Aufruf von ATCAL beinhaltet sie die Werte der Gütefunktion und die maximale Soll-Ist-Abweichung. Es kann eine lokale, globale oder permanente Variable benutzt werden.

Im Einzelnen gilt:

ResultVar(1):

Wert der Gütefunktion $\sqrt{E(\vec{p}_0)}$ vor der Optimierung, also mit dem nicht optimierten Parametersatz $\vec{p}_0$.

Die Einheit ist mm.

ResultVar(2):

Wert der Gütefunktion $\sqrt{E(\vec{P}_1)}$ nach der Optimierung, also mit dem optimierten Parametersatz $\vec{P}_1$.

Die Einheit ist mm.

ResultVar(3):

Wert der maximalen Abweichung $\max_i |\vec{P}_i^m - \vec{P}_i^e|$ vor der Optimierung in mm.

ResultVar(4):

Wert der maximalen Abweichung $\max_i |\vec{P}_i^m - \vec{P}_i^e|$ nach der Optimierung in mm.

n

Mit dem vierten Parameter kann die Maximalanzahl der Iterationen von vornherein begrenzt werden. Bei $n = 1$ liegt der Fall der linearen Ausgleichsrechnung vor. Die Anzahl der Iterationen ist unbegrenzt, wenn "n" nicht angegeben wird oder wenn $n = -1$ ist. Die Iteration wird in diesem Fall erst beendet, wenn die Abweichung zweier aufeinanderfolgender Parametersätze praktisch Null ist. Hierbei wird eine NC-interne Schranke benutzt.

Beispiel:

```
01 DIM LENPARAM! (16)
01 DIM RESULTS! (4)
01 LENPARAMASK% = 2+4+32
N1 ATCAL ("CalibData.txt", LENPARAM!, LENPARAMASK%,
RESULTS!, -1)
```

Lesen der aktuellen Parameter aus der NC:

Die aktuell gültigen Konfigurationsparameter der Achs-Transformation werden mit ATGET aus den Maschinenparametern ausgelesen.

```
ATGET (<McdParamVar>[, <AxTrafoNo>] [, <McdParamNo>])
```

McdParamVar

Vgl. Abschnitt "LenParamVar". Gibt den Namen eines CPL-Variablen-Arrays an, der nach Aufruf die aktuell gültigen Werte des Maschinenparameters "McdParamNo" der Achs-Transformation "AxTrafoNo" enthält. Die Variable muss den CPL-Typ DOUBLE und die Dimension ≥ 8 (McdParamNo = 1030 00130) bzw. ≥ 16 (McdParamNo = 1030 00140) haben. Die Werte haben die gleiche Einheit wie der entsprechende Maschinenparameter.

AxTrafoNo

Nummer der Achs-Transformation, deren Parameter gelesen werden sollen. Wird AxTrafoNo nicht angegeben, wird die aktive Achs-Transformation benutzt. Hier gilt wieder die Reihenfolge AT2 vor AT1.

Der Befehl "ATGET" führt zu einem Laufzeitfehler, wenn AxTrafoNo fehlt und keine Achs-Transformation (COORD(0,2) und COORD(0,1)) aktiv ist.

McdParamNo

Nummer des Maschinenparameters, der gelesen wird. Zulässige Werte für McdParamNo sind 1030 00130 (Achspalten der Referenzstellung) und 1030 00140 (Längen- und Winkelparameter). Wird McdParamNo nicht programmiert, wird per Default 1030 00140 unterlegt, d. h., die Längen- und Winkelparameter werden eingelesen.

Achs-Transformation

Beispiele:

01 DIM LENPARAM! (16)	; Folgende Werte werden eingelesen:
01 DIM ZEROPOS! (8)	
N1 ATGET (LENPARAM!)	; MP 1030 00140 der aktuellen Achs-Transformation
N1 ATGET (LENPARAM!, 3)	; MP 1030 00140 der Achs-Transformation 3
N1 ATGET (LENPARAM!, 2, 103000140)	; MP 1030 00140 der Achs-Transformation 2
N1 ATGET (ZEROPOS!, , 103000130)	; MP 1030 00130 der aktuellen Achs-Transformation
N1 ATGET (ZEROPOS!, 3, 103000130)	; MP 1030 00130 der Achs-Transformation 3

Schreiben der optimierten Parameter in die NC:

Die Übernahme der optimierten Parameter in die NC erfolgt mit ATPUT.

ATPUT (<McdParamVar>[, <AxTrafoNo>] [, <McdParamNo>])

McdParamVar

Siehe "ATGET".

AxTrafoNo

Nummer der Achs-Transformation, für die die Parameterübernahme erfolgen soll. Es gilt das Gleiche wie bei "ATGET".

McdParamNo

Siehe "ATGET".

Der Parametersatz, der in der Variablen McdParamVar abliegt, wird in den entsprechenden Maschinenparameter der Achs-Transformation AxTrafoNo übernommen.



Der neue Parametersatz wird erst nach Systemgrundstellung und erneutem Programmieren von COORD(<AxTrafoNo>) aktiv.

Für den Befehl ATPUT ist ein genügend hohes Benutzer-Level zum Schreiben des Maschinenparameters notwendig.

Ausführen einer Vorwärtstransformation mit vorgegebenen Parametern:

Mit ATFWD (**A**xis**T**ransformation**F**or**W**ar**D**) kann ein beliebiger Achspunkt $\vec{q}$ (ACS) mit vorgegebenen Konfigurationsparametern $\vec{p}$ auf einen Koordinatenpunkt $\vec{\rho}$ (WCS) umgerechnet werden, d. h. $\vec{\rho} = \vec{f}_{\text{Fwd}}(\vec{q}, \vec{p})$.

Die Syntax ist:

ATFWD (<CoordVar>, <AxisVar>[, <LenParamVar>])

CoordVar

Name eines CPL-Variablen-Array vom Typ DOUBLE, in den die WCS-Koordinatenwerte $\vec{\rho}$ zurückgeliefert werden. Die Dimension der Variablen muss mindestens so groß, wie die Anzahl der Kanalkoordinaten sein.

AxisVar

Name eines CPL-Variablen-Array vom Typ DOUBLE, in dem bei Aufruf die

ACS-Achswerte $\vec{q}$ abliegen. Die Dimension der Variablen muss mindestens so groß, wie die Anzahl der Kanalachsen sein.

LenParamVar

Ausführen einer Rückwärtstransformation mit aktuellen Maschinenparametern:

Enthält die vorgegebenen Konfigurationsparameter $\vec{\rho}$ (AT2 vor AT1). Wird LenParamVar nicht programmiert, werden die aktuell gültigen Parameter benutzt.

Ist LenParamVar programmiert, muss die Achs-Transformation eingeschaltet sein. Andernfalls wird ein Laufzeitfehler erzeugt.

Mit ATBWD (**A**xis**T**ransformation**B**ack**W**ar**D**) kann ein beliebiger Koordinatenpunkt $\vec{\rho}$ (WCS) mit den aktuell gültigen Konfigurationsparametern $\vec{\rho}$ auf einen Achspunkt $\vec{q}$ (ACS) umgerechnet werden, d. h. $\vec{\rho} = \vec{f}_{\text{Bwd}}(\vec{q}, \vec{\rho})$.

Die Syntax ist:

ATBWD(<AxisVar>,<CoordVar>)

AxisVar

Name eines CPL-Variablen-Array vom Typ DOUBLE, in den die ACS-Achswerte $\vec{q}$ zurückgeliefert werden. Die Dimension der Variablen muss mindestens so groß, wie die Anzahl der Kanalachsen sein.

CoordVar

Name eines CPL-Variablen-Array vom Typ DOUBLE, in dem bei Aufruf

WCS-Koordinatenwerte $\vec{\rho}$ abliegen. Die Dimension der Variablen muss mindestens so groß, wie die Anzahl der Kanalkoordinaten sein.

Zum Aufrufzeitpunkt von ATBWD muss die entsprechende Achs-Transformation eingeschaltet sein.

Kalibrierdatei

Die Kalibrierdatei enthält die Daten eines Kalibrierzyklus. Der Name kann beliebig gewählt werden (z. B. "CalibDirect.cnc"). Die Datei muss im RAM-Dateisystem abliegen.

Aufbau der Kalibrierdatei:

In Abhängigkeit der verwendeten Achs-Transformation können in der Kalibrierdatei folgende Zeilen enthalten sein:

#1 <FixCoordMask> ; z. B. "5" entspricht den Koordinaten 1 und 3

#2 <DepCoordMask>

#3 <AxisMask>; z. B. "21" entspricht den Achsen 1, 3 und 5

#4 <MeasBallGeo> ; Geometrie-Daten der Messkugel

; Messkugel-Radius, x-, y-, z-Koordinaten des Mittelpunktes

#11 <FixCoordPos> ; Parameter unabhängige Koordinatenwerte eines Messpunktes

#12 <DepCoordPos> ; Parameter abhängige Koordinatenwerte eines Messpunktes

#22 <AxisPos> ; zu #12 alternative Achsposition

#32 <EstAxisPos> ; geschätzte Achsposition

Bedeutung der Zeilen:

- #1 legt fest, welche WCS-Koordinaten in der Zeile #11 abliegen.
Die aus FixCoordMask sich ergebende Koordinatenanzahl muss mit der Anzahl der Einträge in #11 übereinstimmen.
- #2 legt fest, welche WCS-Koordinaten in der Zeile #12 abliegen.
Die aus DepCoordMask sich ergebende Koordinatenanzahl muss mit der Anzahl der Einträge in #12 übereinstimmen.
- #3 legt fest, welche ACS-Koordinaten (Kanalachsen 1 bis 8) in den Zeilen #22 und #32 abliegen.
- #4 enthält als erster Eintrag den Radius der Messkugel und als Eintrag 2,3,4 die festen x-,y-,z-Koordinatenwerte des Messkugelmittelpunktes im BCS.
Die Einzelwerte sind durch Leerzeichen oder Tabs getrennt.
Das indirekte Messverfahren (Variante 2) erwartet diesen Eintrag. Er ist nur ein Mal pro Kalibrierdatei vorzunehmen.
- #11 enthält die festen Koordinatenwerte $\vec{p}_i^m$ beim direkten oder $\vec{p}_i^e$ beim indirekten Messverfahren für einen gegebenen Messpunkt i.
Die Einzelwerte sind durch Leerzeichen oder Tabs getrennt.
- #12 enthält die Parameter abhängigen Koordinatenwerte $\vec{p}_i^e$ beim direkten oder $\vec{p}_i^m$ beim indirekten Messverfahren für einen gegebenen Messpunkt i.
Die Einzelwerte sind durch Leerzeichen oder Tabs getrennt.
- #22 enthält Achswerte $\vec{q}_i^e$ bzw. $\vec{q}_i^m$, die an Stelle von #12 verwendet werden können, wenn die Achs-Transformation einen mehrdeutigen Rückwärtszweig hat (z. B. 5-Achs-Transformation).
#12 und #22 **schließen** sich gegenseitig **aus**.
- #32 enthält Achsschätzwerte als Zusatz zu #12. Die Zeile ist nur notwendig, wenn die Achs-Transformation mehrdeutig ist und die #22-Werte nicht bekannt sind.
#32 muss in diesem Fall zusammen mit #12 in der Kalibrierdatei enthalten sein.
#22 und #32 **schließen** sich gegenseitig **aus**.

Das Kommentarzeichen in der Tabelle ist das Semikolon.

Die Kalibrierdatei bietet vier Definitionen für die Strukturierung der Messpunkte "i" an:

- für eine eindeutige Achs-Transformation bestimmt sich der Messpunkt durch:
#11 <FixCoordPos>
#12 <DepCoordPos> oder #22 <AxisPos>
- für eine mehrdeutige Achs-Transformation bestimmt sich der Messpunkt durch:
#11 <FixCoordPos>
#22 <AxisPos>

- für eine mehrdeutige Achs-Transformation, wenn <AxisPos> **nicht** genau bekannt ist:
 #11 <FixCoordPos>
 #12 <DepCoordPos>
 #32 <EstAxisPos>
- Beim indirekten Messverfahren (Variante 2) für eine Achstransformation bestimmt sich der Messpunkt durch:
 #22 <AxisPos>

Beispiele für Kalibrierdateien:**Beispiel 1:**

Direktes Messverfahren bei 5-Achs-Transformation.

Kanalachsen: X, Y, Z, U, V, W, B, C.

Die 5-Achs-Transformation 3032001 transformiert zwischen den Achsen X, Y, Z, B, C und den Koordinaten x, y, z, B, C. B und C werden direkt durchgereicht.

Für die Kalibrierung werden die Punkte $\vec{\rho}_1^e$ (entspricht x, y, z, B, C) angefahren. Die tatsächliche TCP-Position $\vec{\rho}_1^m$ (entspricht x,y,z) wird gemessen.

Da die Achs-Transformation eindeutig ist, können die Zeilen #11 und #12 für die Messpunkte benutzt werden.

Die Kalibrierdatei enthält daher folgende Angaben:

#1	7						$\vec{\rho}_1^m$	enthält die Koordinaten ; 1, 2 und 3 (x, y, z) = 7 bitcodiert
#2	199						$\vec{\rho}_1^e$	-Koordinaten 1, 2, 3, 7 und 8 ; (X, Y, Z, B, C) = 199 bitcodiert
#11	20.3	510.24	80.1				$\vec{\rho}_1^m$	
#12	20.4	510.14	80.1	20.0	181.5		$\vec{\rho}_1^e$	
#11	40.9	510.23	20.1				$\vec{\rho}_2^m$	
#12	50.0	510.13	19.0	40.0	165.0		$\vec{\rho}_2^e$	
...								
#11	100.32	23.2	220.1				$\vec{\rho}_k^m$	
#12	100.22	23.3	220.41	76.2	-36.1		$\vec{\rho}_k^e$	

Beispiel 2:

Direktes Messverfahren bei Kardanischer Achs-Transformation (3232211).

Achs-Transformation

Kanalachsen: X, Y, Z, B und C

Die kardanische Achs-Transformation transformiert zwischen den Koordinaten 1, 2, 3, 4 und 5 (Koordinatennamen x, y, z, phi, theta) und den Achsen X, Y, Z, B und C.

Für die Kalibrierung werden die Punkte $\vec{\rho}^e$ (entspricht x, y, z, ϑ , φ) angefahren. Gemessen werden die tatsächlichen TCP-Positionen $\vec{\rho}^m$ (entspricht x,y,z).

Die Achs-Transformation ist mehrdeutig, so dass an Stelle der Koordinaten-

Sollpunkte $\vec{\rho}^e$ (#12) die Achspunkte $\vec{q}^e$ (#22) benutzt werden müssen. Die

Achspunkte erhält man nach Anfahren der Position $\vec{\rho}^e$ mit dem CPL-Befehl SPOS.

Kalibrierdatei:

#1	7						$\vec{\rho}^m$	enthält die Koordinaten
								; 1, 2 und 3 (x, y, z) = 7 bitcodiert
#3	199						$\vec{q}^e$	Kanalachsen 1, 2, 3, 7 und 8
								; (X, Y, Z, B, C) = 199 bitcodiert
#11	20.3	510.24	80.1				$\vec{\rho}^m$	Koordinaten x, y, z
#22	100.54	28.3	470.22	0.0	181.5		$\vec{q}^e$	Achspositionen X,Y,Z,B,C
...								
..								
.								
#11	100.32	23.2	220.1				$\vec{\rho}^m$	
#22	200.23	49.3	522.81	76.2	-36.1		$\vec{q}^e$	

Beispiel 3:

Indirektes Messverfahren (Variante 1) bei Bipod-Achs-Transformation (3131001).

Im Kanal befinden sich die Achsen YL, YR, Za, B, C. Hierbei ist C eine Spindel-/C-Achse.

Die Achs-Kinematik hat die Koordinaten 1, 2, 3 und 6 (Koordinatennamen X, Y, Z, psi) und die Achsen YL, YR, Za und C.

Für die Kalibrierung werden die parameterabhängigen Lochmittelpunkte $\vec{p}_i$ (Koordinaten X, Y) mittels Messtaster gemessen. Die exakten Lochmittelpunkte $\vec{p}_i$ sind bekannt.

Kalibrierdatei:

#1	3		$\vec{p}_i$	Koordinaten 1 und 2 (X, Y) = 3 bitcodiert
#2	3		$\vec{p}_i$	Koordinaten 1 und 2 (X, Y) = 3 bitcodiert
#11	20.3	510.2	$\vec{p}_i$	exakte Position (X,Y) des ersten Lochs
#12	20.3	510.3	$\vec{p}_i$	exakte Position (X,Y) des zweiten Lochs
...				
..				
.				
#11	100.5	10.2	$\vec{p}_k$	
#12	100.6	10.1	$\vec{p}_k$	

Beispiel 4:

Indirektes Messverfahren (Variante 2) bei Kardanischer Achstransformation (3232211).

Kanalachsen: X, Y, Z, B und C

Die kardanische Achstransformation transformiert zwischen den Koordinaten 1, 2, 3, 4 und 5 (Koordinatennamen x, y, z, phi, theta) und den Achsen X, Y, Z, B und C.

Die Geometriedaten der Messkugel sind bekannt. Kugelradius=23 mm, Kugelmittelpunkts-Koordinaten (x,y,z) = (10.10, -21.85, 185.26).

Für die Kalibrierung werden die Punkte $\vec{p}_i = (x, y, z, \text{phi}, \text{theta})$ auf der Messkugel angefahren. Gemessen werden die Achspositionen $\vec{q}_i$ (#22).

Kalibrierdatei:

#1	31		$\vec{p}_i$	-Koordinaten 1,...,5 (x,y,z,phi,theta)
#3	199		$\vec{q}_i$	Kanalachsen 1, 2, 3, 7 und 8
				; (X, Y, Z, B, C) = 199 bitcodiert

Achs-Transformation

#4	23.0	10.10	-21.85	185.26		; Kugelradius, x-, y-, z-Koordinate des Kugelmittelpunkts.
#11	14.1	-6.0	213.6	0.0	10.0	; $\vec{A} = (x,y,z,\phi,\theta)$ exakte angefahrne Kugelposition
#22	-306.0	-176.8	-33.3	10.0	270.0	; $\vec{q}^i$ gemessene Achspositionen X,Y,Z,B,C
...						
..						
.						
#11	26.9	-11.5	205.7	330.0	50.0	; $\vec{A}$
#22	-302.6	-61.1	-91.5	50.0	288.0	; $\vec{q}^i$

Kalibrierzyklus

Kalibrierzyklus beim direkten Messverfahren:

Die folgenden Abläufe beschreiben die Maßnahmen zur Durchführung eines Kalibrierzyklus mit dem direkten und indirekten Messverfahren.

Prinzipieller Aufbau eines CPL-Kalibrierzyklus beim direkten Messverfahren:

1. Auswahl und Aktivieren der zu kalibrierenden Achs-Transformation (z. B. COORD(3))
2. Kalibrierdatei zum Schreiben öffnen
3. Verwendete Koordinaten und Achsen in die Datei schreiben (#1, #2, #3)

4. Messpunkte (Koordinatensollwerte $\vec{A}$) anfahren und Koordinatenpositionen $\vec{q}^i$ messen.

$\vec{q}^i$ in Zeile #11 und $\vec{A}$ in Zeile #12 eintragen.

Ist die Transformation mehrdeutig, wird an Stelle von #12 der Eintrag

#22 $\vec{q}^i$ benutzt. Die angefahrne Achsposition kann mit dem CPL-Befehl SPOS erfragt werden.

5. Messdatei schließen.
6. Ermitteln der aktuellen Konfigurationsparameter mit (ATGET(LENPAR!)).
7. Zu optimierende Einzelparameter als Bitmaske in LENPARMASK% ablegen.
8. Starten der Kalibrierrechnung mit (ATCAL("CalibData.txt",LENPAR!,LENPARMASK%))
9. Die berechneten Daten werden in der Variablen LENPAR!(1,...,16) zurückgeliefert.

10. Gegebenenfalls Plausibilitätsprüfung der Parameter durchführen. Hier kann ggf. eine Limitierung der Parameter durchgeführt werden.
11. Schreiben der neuen Daten (N.. ATPUT(LENPAR!) in die Maschinenparameter 103000140.
12. Kalibrierzyklusende
13. Systemgrundstellung durchführen

Danach ist das Kalibrieren beendet. Die neuen Daten stehen im MP 103000140 der benutzten Achs-Transformation.

Beispiel:

Kalibrierzyklus: kardanische 5-Achs-Transformation 3232211.

Die Achskonfiguration im Kanal ist {Xa, Ya, Za, U, V, W, B ,C}.

Durch die Transformation werden die Achsen {Xa, Ya, Za, B ,C} in die Koordinaten {x, y, z, phi, theta} transformiert.

An jedem Messpunkt werden die Koordinatenpositionen x, y, und z gemessen.

Für die zu optimierenden Parameter wird eine permanente Variable @LENPAR!(16) angelegt.

Optimiert werden die Parameter 1, 4, 5, 6 und 7.

Der Zyklus generiert eine Kalibrierdatei "Calib.txt", die wie das Beispiel 2 (siehe [Kap. "Kalibrierdatei " auf Seite 413](#)) aufgebaut ist.

Kalibrierzyklus-Programm **CalibDirect.cnc** zum Kalibrieren:

```
000 DIM AXPOS!(5)
N10 X0Y0Z0 B170 C-180 F10000 G1
;
; Kalibrierdatei öffnen und Zeilen #1,#3 eintragen
010 OPENW(1,"CalibData.txt",512)
010 REWRITE(1)
010 PRN#(1,"#1 7") : REM Koordinaten 1,2,3
010 PRN#(1,"#3 199") : REM Kanalachsen 1,2,3,7,8
N20 Coord(2)
020 FOR THETA=80.0 STEP -35.0 TO 45.0
020 FOR PHI=0.0 STEP 60.0 TO 300.0
; Messpunkt anfahren
N30 x10 y10 z0 phi[PHI] theta[THETA]
N40 WAIT
; Achspositionen des angefahrenen Messpunkts auslesen
040 AXPOS!(1) = SPOS(1)
040 AXPOS!(2) = SPOS(2)
040 AXPOS!(3) = SPOS(3)
040 AXPOS!(4) = SPOS(7)
040 AXPOS!(5) = SPOS(8)
;
; TCP-Position messen (direktes Messverfahren)
```


Die Ergebnisse der Messung werden in einer Messdatei mit der folgenden Struktur (Beispiel) abgelegt:

Loch-Nr.	Exakte Pos.	Achspunkt 1	Achspunkt 2	Achspunkt 3
1	$\vec{P}_1^m$	$\vec{q}_{1.1}^m$	$\vec{q}_{1.2}^m$	$\vec{q}_{1.3}^m$
2	$\vec{P}_2^m$	$\vec{q}_{2.1}^m$	$\vec{q}_{2.2}^m$	$\vec{q}_{2.3}^m$
...				
k	$\vec{P}_k^m$	$\vec{q}_{k.1}^m$	$\vec{q}_{k.2}^m$	$\vec{q}_{k.3}^m$

mit

$\vec{O}^e$

Exakter Mittelpunkt der Bohrung.

$\vec{q}_{ij}^m$

Mit Messtaster ermittelte Achspunkte.

$\vec{P}_{ij}^m$

Aus $\vec{q}_{ij}^m$ berechnete Koordinatenwerte.

$\vec{P}_i^m$

Aus $\vec{P}_{ij}^m$ berechneter Mittelpunkt des grauen Kreises.

Der Messzyklus muss nur einmalig ausgeführt werden, da alle Werte der Messdatei unabhängig von den Konfigurationsparametern der benutzten Achs-Transformation sind.

Der darauf folgende Kalibrierzyklus benutzt die Messdatei als Basis für die Parameteroptimierung.

Prinzipieller Aufbau eines CPL-Kalibrierzyklus beim indirekten Messverfahren:

1. Auswahl und Aktivieren der zu kalibrierenden Achs-Transformation (z. B. COORD(3))
2. Ermitteln der aktuellen Konfigurationsparameter mit (ATGET(LENPAR!))
3. Messdatei zum Lesen öffnen
4. Kalibrierdatei zum Schreiben öffnen
5. Verwendete Koordinaten und Achsen in die Kalibrierdatei schreiben (#1, #2, #3)

6. Exakte Position $\vec{O}^e$ von der Messdatei in die Kalibrierdatei übertragen (#11-Zeile).

7. Die Achstastpunkte $\vec{q}_{i.1}^m$, $\vec{q}_{i.2}^m$ und $\vec{q}_{i.3}^m$ in Koordinatenpunkte $\vec{P}_{i.1}^m$, $\vec{P}_{i.2}^m$ und $\vec{P}_{i.3}^m$ umrechnen mit dem NC-Befehl ATFWD.

8. Lochmittelpunkt $\vec{P}_i^m$ aus den Koordinatenpunkten $\vec{P}_{ij}^m$ berechnen.
9. In die #12-Zeile der Kalibrierdatei schreiben.
Ist die Achs-Transformation mehrdeutig, kann zusätzlich die #32-Zeile mit einem der Achstastpunkte beschrieben werden.
10. Weiter bei Punkt 6, bis die letzte Zeile der Messdatei erreicht ist.
11. Mess- und Kalibrierdatei schließen.
12. Zu optimierende Einzelparameter als Bitmaske in LENPARMASK% ablegen.
13. Starten der Kalibrierrechnung
N...ATCAL("CalibData.txt", LENPAR!,LENPARMASK%,RESULTS!)
Die optimierten Parameter werden in der Variablen LENPAR!(1,...,16) zurückgeliefert
14. Weicht RESULTS!(1) nur geringfügig von RESULTS!(2) ab, ist der Zyklus beendet.
Andernfalls geht es bei Punkt 3 weiter.
15. Gegebenenfalls Plausibilitätsprüfung der Parameter.
Hier kann ggf. eine Limitierung der Parameter durchgeführt werden.
16. Schreiben der neuen Daten (N.. ATPUT(LENPAR!) in den Maschinenparameter 103000140.
17. Kalibrierzyklusende
18. Systemgrundstellung durchführen.

Danach ist das Kalibrieren beendet. Die neuen Daten stehen in MP 103000140 der benutzten Achs-Transformation.

Beispiel:

Kalibrierzyklus für die Bipod-Transformation vom Typ 3131001.

Die Achskonfiguration im Kanal ist {YL, YR, Za, B, C}.

Die Transformation transformiert zwischen den Achsen {YL, YR, Za, C} und den Koordinaten {X, Y, Z, psi}. An jedem Messpunkt werden die Koordinatenpositionen (X, Y) gemessen. Für die zu optimierenden Parameter sei eine permanente Variable @LENPAR!(16) angelegt. Optimierte werden die Parameter 1 bis 12. Der Zyklus generiert eine Kalibrierdatei "Calib.txt", die wie das Beispiel 3 ("[Beispiele für Kalibrierdateien:](#)" auf Seite 415) aufgebaut ist. Es wird angenommen, dass vorher ein Messzyklus gelaufen ist, der die Messdatei "Holes.txt" mit der Struktur des vorherigen Abschnitts generiert hat.

Kalibrierzyklus-Programm **CalibIndirect.cnc** zum Kalibrieren:

```
000 DIM AXPOS!(5) : DIM COPOS!(5)
000 DIM AP!(2,3) : DIM CP!(2,3)
000 DIM ECP!(2) : DIM MCP!(2)
000 DIM RESULTS!(4)

N10 COORD(1)
N20 ATGET(@LENPAR!)
020 REPEAT
```



```

;
; Messdatei zum Lesen öffnen
020 OPENR(2,"Holes.txt")
;
; Kalibrierdatei zum Schreiben öffnen und Zeilen #1,#3 eintragen
020 OPENW(1,"CalibData.txt",512)
020 REWRITE(1)
020 PRN#(1,"#1 3") : REM Koordinaten 1,2
020 PRN#(1,"#2 3") : REM Koordinaten 1,2

020 WHILE NOT (EOF(2)) DO
; Exakte Lochposition aus Holes.txt in ECP!(2) einlesen
; Drei Achstastpunkte aus Holes.txt in AP!(2,3) einlesen
020 INP#(2,.....)
; Achstastpunkte AP!(2,3) in Koordinatenpunkte CP!(2,3) umrechnen
020 FOR J% = 1 TO 3
020 AXPOS!(1) = AP!(1,J%) : AXPOS!(2) = AP!(2,J%)
020 FOR I%=3 TO 5 : AXPOS!(I%) = 0.0 : NEXT I%
N30 ATFWD(COPOS!,AXPOS!,@LENPAR!)
030 CP!(1,J%) = COPOS!(1) : CP!(2,J%) = COPOS!(2)
030 NEXT J%
; Aus CP!(2,3) Lochmittelpunkt MCP!(2) berechnen
<Berechnungsalgorithmus liefert MCP!>
; Koordinatenpositionen ECP und MCP in CalibData.txt schreiben
030 PRN#(1,"#11 ";) : PRN#(1,ECP!(1),ECP!(2))
030 PRN#(1,"#12 ";) : PRN#(1,MCP!(1),MCP!(2))
030 END
030 CLOSE(1) : CLOSE(2)

; Kalibrierrechnung durchführen
040 LENPARMASK% = 1+2+4+8+16+32+64+128+256+512+1024+2048 :
REM P1 bis P12
N40 ATCAL("CalibData.txt",@LENPAR!,RESULTS!)
; Verbesserung der Güte überprüfen
040 DIFF! = ABS(RESULTS!(1) - RESULTS!(2))
040 UNTIL (DIFF! < 0.1)
N50 ATPUT(@LENPAR!)
M30

```

Nach Beenden des Programms ist **Systemgrundstellung** durchzuführen.

11.25.3 Handlungsanweisung

- Applizieren**
- Alle transformations-relevanten Achsen werden per Messtaster erfasst, d. h. alle transformations-relevanten Achsen sind an den Messtaster angeschlossen.

Achs-Transformation

- Für die zu kalibrierende Maschine ist sicherzustellen, dass die Referenzpunkte aller Achsen und die Referenzstellung in der angewählten Achstransformation richtig eingestellt ist.
- Die für die Kalibrierung relevante Achstransformation ist zu aktivieren
- Bei Vorhandensein zweier Achstransformation, d. h., sowohl an Wirkstelle AT1 als auch an AT2, ist **nur** die Achstransformation an AT2 zu aktivieren.
- Bei der direkten Messmethode sind im relevanten Bearbeitungsraum für ein Werkstück verschiedene Koordinatenpunkte festzulegen und in Bezug auf die Maschinenkoordinaten exakt zu vermessen.
- Bei der indirekten Messmethode (Variante 1) ist z. B. eine Lochplatte auf der Werkstückaufnahme so auszurichten, dass ein Bezug zu WCS entsteht.

Bei der Nutzung der direkten Messmethode ist ein Kalibrierzyklus-Programm zu programmieren.

Bei der Nutzung der indirekten Messmethode (Variante 1) ist zusätzlich zum Kalibrierzyklus-Programm ein Messzyklusprogramm für die Erfassung der Messpunkte z. B. von Bohrungsabtautungen auf einer Lochplatte zu erstellen.

Die Programme sollten so geschrieben sein, dass die Messwerte automatisch ermittelt werden, die Abweichung daraus berechnet wird und eine Optimierung der Längen- und Winkelparameter automatisch in MP übernommen werden kann.

Aktivieren	Kalibrierzyklus-Programm und bei indirektem Messen das Messzyklusprogramm starten. Nach Beenden der Programme sind alle Werte durch Systemgrundstellung zu übernehmen.
Deaktivieren	Ohne Bedeutung.

12 Technologien

12.1 Gewindebohren

12.1.1 Beschreibung

Allgemeines

Gewindebohren kann auf zwei Arten durchgeführt werden:

- mit Ausgleichsfutter
- ohne Ausgleichsfutter.

Hier wird nur das Gewindebohren ohne Ausgleichsfutter beschreiben.

Gewindebohren ohne Ausgleichsfutter ist nur mit Sercos Spindeln möglich.

Funktion

Funktion 1: Gewindebohren ohne Ausgleichsfutter

Höhere Qualitätsanforderungen an die Gewindegüte erfordern ein exaktes Zusammenspiel zwischen Bohrachse und Spindel.

Für die Erzeugung eines Gewindes ohne Ausgleichsfutter synchronisiert die NC die lineare Zustellbewegung der Bohrachse mit der Drehbewegung der Spindel. Hierbei entsteht **kein Versatz**, da die Zustellposition der Bohrachse und die Spindeldrehung miteinander interpolieren, d. h. Bohrachse und Spindel beschleunigen/verzögern **zeitgleich** im selben Verhältnis.

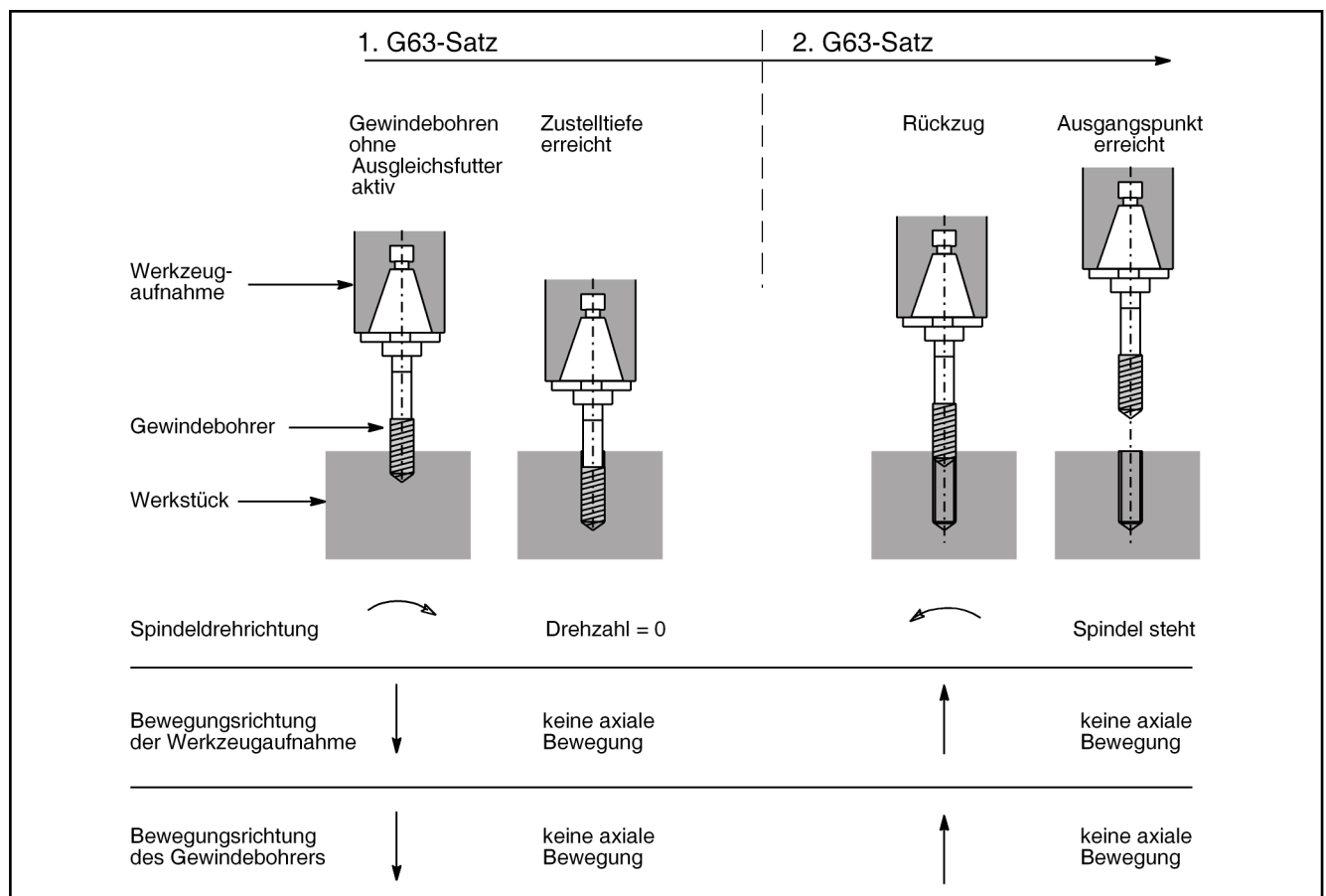


Abb. 12-1: Gewindebohren ohne Ausgleichsfutter

"Gewindebohren ohne Ausgleichsfutter" besteht immer aus **zwei** aufeinanderfolgenden Sätzen

- G63 zum "Einbohren" und
- G63 bzw. G63.2 für den "Rückzug" aus der Bohrung.

Die Gewindesteigung wird wie folgt vorgegeben:

- als Verhältnis von Bahnvorschub der Bohrachse zur Spindeldrehzahl (F/S).

Programmierung:

G63 (M..., S...) F...

- oder -

- durch die direkte Angabe der Gewindesteigung (H).

Programmierung:

G63 (M..., H...)

Funktion 2: Rückzug aus Gewindebohrung (ohne Ausgleichsfutter)

Beim Gewindebohren kann es vorkommen, dass der Vorgang unterbrochen wird. Verschiedene Ursachen können dafür verantwortlich sein:

Ursache der Unterbrechung	Gewindebohrdaten vorhanden	Fortsetzung des Gewindebohrens	Programmierung
Vorschubpoti = 0	ja	Ja: Vorschubpoti >0 setzen	-
Grundstellung ausgelöst	ja	Nein: automatisches Herausfahren (Rückzug) des Gewindebohrers	TappRet1 und TappRet2 mit Parameter F
Spannungsausfall (Hochlauf der NC erforderlich)	nein	Nein: manuelles Herausfahren (Rückzug) des Gewindebohrers	TappRet1 und TappRet2 mit den Parametern S, F, M3/M4, Bohrachse und inkrementalem Rückzugsweg
Werkzeugbruch	ja	Nein: automatisches Herausfahren (Rückzug) des Gewindebohrers nach Grundstellung	TappRet1 und TappRet2 mit F

Tab. 12-1: Ursachen der Unterbrechung

Der Rückzug (automatisch und manuell) wird grundsätzlich durch die Funktion "TappRet1" (Umschaltung der Spindel auf C-Achsbetrieb) eingeleitet. Mit "TappRet2" wird die Rückzugbewegung gestartet.

Die zugehörigen Parameter kommen entweder automatisch von der Steuerung (nach Grundstellung) oder sind manuell anzugeben (nach Spannungsausfall und anschließendem Steuerungshochlauf):

- Startposition des 1. Gewindebohrsatzes (mit G63)
- Gewindesteigung (F/S)
- Drehrichtung
- beteiligte Spindel

Die Programmierung von TappRet1 und TappRet2 kann per Handeingabe und per Zyklus erfolgen.

Um die Anwendung der Funktionalität "Rückzug aus Gewindebohrung" zu vereinfachen, wird empfohlen, für den automatischen und den manuellen Rückzug jeweils ein Unterprogramm (Zyklus) zu schreiben. Mit den Maschinenparametern 3090 00001 und 3090 00002 lassen sich diese Unterprogramme dann einem beliebigen freien G-Code zuordnen:

Beispielzyklus für einen automatischen Rückzug:

AutoTR[Vorschub]

```
N010
0020 DIM FCODE$(4)
0030 IF P1=NUL THEN
N040 (MSG, ** P1 KEIN VORSCHUB PROGRAMMIERT **)
N050 M0
0060 GOTO N40
0070 ELSE
0080 FCODE$=NCF("G94")
0090 FVORSCH%=SD (5,1,2)
N100 TappRet1
N110 G94 TappRet2 F[P1]
N120 [FCODE$]
0130 IF FCODE$="G94" THEN
N140 F[FVORSCH%]
0150 ENDIF
0160 ENDIF
M30
```

Beispielzyklus für einen manuellen Rückzug:

ManTR[Bohrachsennummer, Weg, Originalsteigung, Vorschub]

```
0010 DIM GABS_INK$(4)
0010 DIM GVORSCH$(4)
0020 IF P1=NUL THEN
N030 (MSG, ** P1 ACHSE IST NICHT PROGRAMMIERT **)
N040 M0
0050 GOTO N30
0060 ENDIF
0070 IF P2=NUL THEN
N080 (MSG, ** P2 WEG IST NICHT PROGRAMMIERT **)
N090 M0
0100 GOTO N80
0110 ENDIF
0120 IF P3=NUL THEN
N130 (MSG, ** P3 STEIGUNG IST NICHT PROGRAMMIERT **)
N140 M0
0150 GOTO N130
0160 ENDIF
0170 IF P4=NUL THEN
```

```

N180 (MSG, ** P4 VORSCHUB IST NICHT PROGRAMMIERT **)
N190 M0
0200 GOTO N180
0210 ENDIF
0220 BACHSE%=ROUND(P1)
0230 IF P3>0 THEN
0240 MCODE1$="M4"
0250 ELSE
0260 MCODE1$="M3"
0270 ENDIF
0280 STEIG=ABS(P3)
0290 GABS_INK$=NCF("G90")
0300 GVORSCH$ =NCF("G94")
0310 FVORSCH% =SD (5,1,2)
N320 TappRet1
N330 G91 G94 TappRet2 [AXP(BACHSE%,P2)] H[STEIG] F[P4]
[MCODE1$]
N340 [GABS_INK$] [GVORSCH$]
0350 IF GVORSCH$="G94" THEN
N360 F[FVORSCH%]
0370 ENDIF
M30

```

Wenn gewünscht, können die Zyklen mit nachfolgenden Einstellungen G-Funktionen zugewiesen werden:

Parameter: 3090 00001

0 6301

1 6302

2 ...

Parameter: 3090 00002

0 AutoTR

1 ManTR

2 ...

Beispiel:

G6301[1000]	Automatischer Rückzug mit Vorschub F1000 mm/min
G6302[3,-100,0.5,500]	Manueller Rückzug bei Gewindesteigung 0.5, Rückzugsweg -100 mm, F500, Bohrachse

Restriktionen

- Einbohren und Rückzug müssen immer mit identischer Gewindesteigung (F/S bzw. H) programmiert werden.

- Wird zwischen "Einbohren" und "Rückzug" eine "Pause" benötigt, z. B. Verweilzeit oder Hilfsfunktion, muss der Rückzug mit G63.2 programmiert werden.
- Die Rückzugsbewegung kann nach einer Unterbrechung bei aktiven Placements oder Werkstücklagenkorrektur (G152.x, G154.x..G159.x oder BCR) nur **manuell** erfolgen.

Relevante NC-Funktionen

G63	Gewindebohren ohne Ausgleichsfutter (satzweise wirksam) mit Angabe der Bohrachse, Zustelltiefe, Vorschub, Spindeldrehrichtung, Drehzahl, Gewindesteigung, Startwinkel.
G63.2	Folgesatz von Gewindebohren ohne Ausgleichsfutter.

Tab. 12-2: Relevante NC-Funktionen

G84	Gewindebohren
G89	Gewindebohren mit Spanbruch

Tab. 12-3: Relevante NC-Zyklen

Relevante Sercos Parameter

S-0-1050.1.6	Konfigurationsliste AT Dort muss je nach verwendetem Geber der Sercos Parameter S-0-0051 (Lage-Istwert1 Motorgeber) oder S-0-0053 (Lage-Istwert2 externer Geber), S-0-0040 (Drehzahl-Istwert) und S-0-0144 (Signalstatuswort) eingetragen sein.
S-0-1050.0.6	Konfigurationsliste MDT Dort müssen die Sercos Parameter S-0-0047 (Lagesollwert) und S-0-0036 (Drehzahlsollwert) eingetragen sein.
S-0-0026	Konfigurationsliste Signal-Statuswort S-0-0331 (Spindel steht) muss eingetragen werden. Wir empfehlen, in S-0-0026 den Sercos Parameter S-0-0403 (Status Lage-Istwerte) einzutragen.
S-0-0032	Hauptbetriebsart Muss auf die Betriebsart "Geschwindigkeitsregelung" eingestellt sein.
S-0-0033 bis S-0-0034	Nebenbetriebsart 1 bis 2. Muss auf die Betriebsart "Lageregelung..." eingestellt sein.
S-0-0044, S-0-0076 und S-0-0160	Wichtungsart Als Wichtungsart ist "rotatorische Wichtung im Moduloformat" erforderlich.
S-0-0057	Positionierfenster
S-0-0261	Positionierfenster grob
S-0-0103	Modulowert Als Modulowert ist 360 Grad einzustellen.

Wenn der antriebsgeführte Betriebsartenwechsel eines IndraDrive aktiviert ist (Bit 8 in S-0-0034):	
P-0-0152	Aufsynchronisieren beendet P-0-0152 muss zusätzlich in S-0-0026 (Konfigurationsliste Signal-Statuswort) enthalten sein.
P-0-0142	Aufsynchronisier-Beschleunigung Beschleunigungswert zum Abbau der Positionsdivergenz bei antriebsgeführtem Betriebsartenwechsel.
P-0-0143	Aufsynchronisier-Geschwindigkeit Maximalgeschwindigkeit zum Abbau der Positionsdivergenz bei antriebsgeführtem Betriebsartenwechsel.

Tab. 12-4: Sercos Parameter

Relevante IF-Signale

am Kanalinterface

Das Signal "G63 Aktiv" kann am Kanaleingangsinterface unter den aktiven NC-Funktionen konfiguriert werden und steht dann für die Dauer des Gewindebohrers an.

am Spindelinterface

iSp_TurnCmd	"Drehbefehl" für die Dauer von "C-Achsumschaltung aktiv"
iSp_CAxSwitch	"C-Achsumschaltung" ist aktiv
iSp_CAxAct	"C-Achse ist aktiv" für die Dauer des Gewindebohrers

Tab. 12-5: Relevante IF-Signale am Spindelinterface

am Achsinterface

iAx_TrVCmd	"Fahrbefehl" für die C-Achse und Bohrachse für die Dauer des Gewindebohrers
iAx_InPos	"Achse in Position" wird am Beginn und am Ende jeder Bohrbewegung überprüft.

Tab. 12-6: Relevante IF-Signale am Achsinterface

Relevante Maschinenparameter (MP)

100100001	Antriebsfunktionstyp (Antrieb bewegt eine Spindel, die temporär auch auf C-Achsbetrieb umgeschaltet werden kann.)
100300004	Achsbewegungstyp: C-Achse/Spindel ist eine Rundachse mit Modulorechnung
104000001	Auswahl der Spindelart (1 = Sercos Spindel)
105000001	Gibt für jeden Antrieb den Typ an.
104000009	Antriebsparametersätze pro G-Stufe (nur für Indradrive)
104000012	Max. Spindeldrehzahl der Getriebestufe in U/min
104000051	Lageschnittstelle: 1. Beschleunigung in rad/s ²

Tab. 12-7: Relevante Maschinenparameter

12.1.2 Handlungsanweisung: Gewindebohren ohne Ausgleichsfutter applizieren

Applizieren:

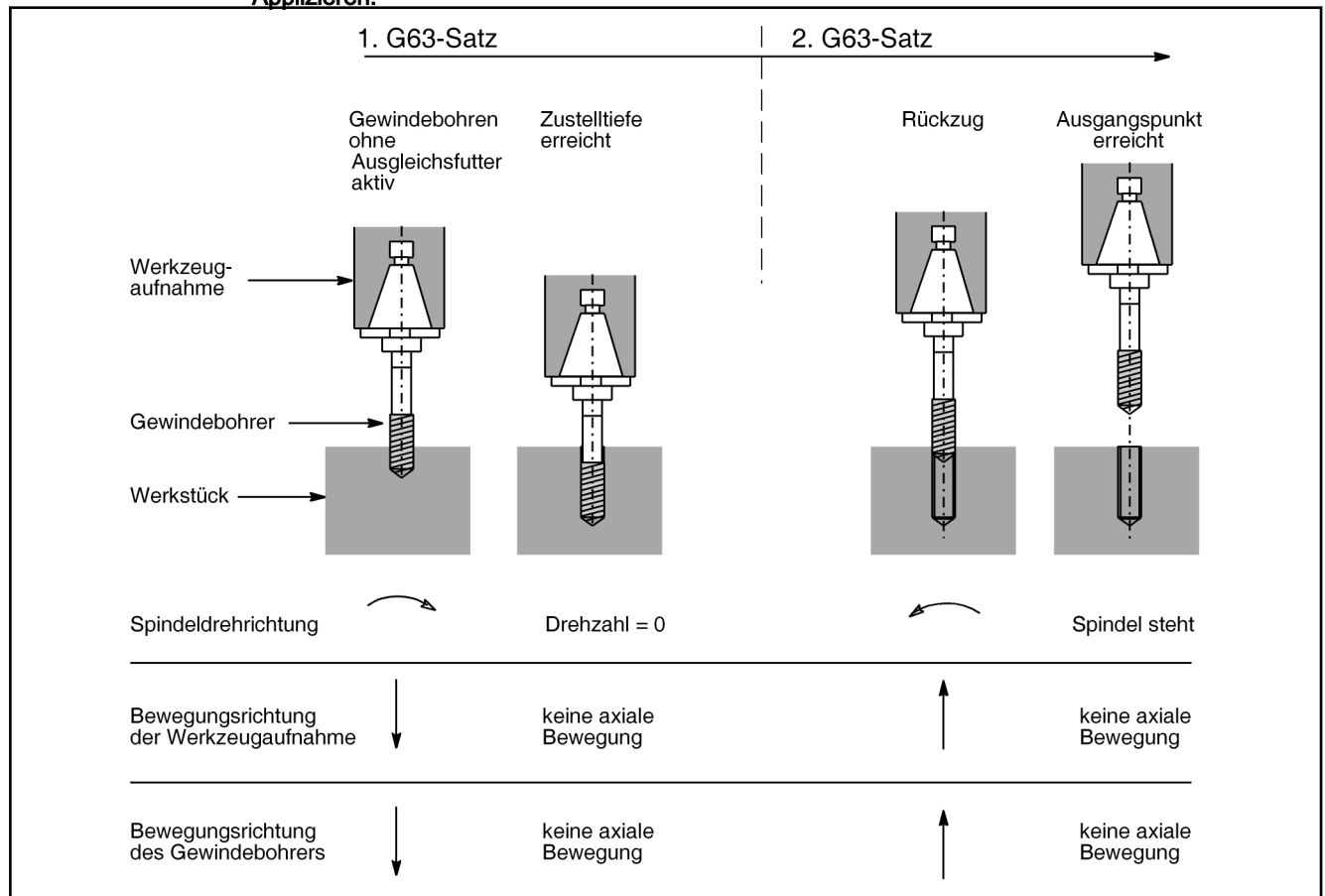


Abb. 12-2: Übersicht Gewindebohren

IW-Engineering / Configuration: Spindel/C-Achse anlegen

		Instruction
Instruction:		C-Achse anlegen

IW-Engineering / Configuration: Parameter editieren

Die Spindeldynamik beim Gewindebohren wird von folgenden Parametern bestimmt:

- **MaxSpAccPosCtrl[1]** "Beschleunigung im Lagereglerbereich[1] von GearStep[1..4]" (1040 00051)
- **MaxSpSpeed** "Maximale Drehzahl von GearStep[1..4] von GearStep[1..4]" (1040 00012)
- **MaxSpJerkPosCtrl** "Max. Spindelruck im Lageregelbetrieb von GearStep[1..4]" (1040 00055)

Wirksam bei aktiver NC-Funktion **JerkControl**.



Wenn **MaxSpAccPosCtrl[1]** den Wert 0 enthält, gilt:

Die Spindeldynamik wird von **MaxAxAcc** "Maximale Achsbeschleunigung" (1010 00001) und **MaxVel** "Maximale Achsgeschwindigkeit" (1005 00002) bestimmt.

Zur weiteren Optimierung des Gewindebohrers kann im Antrieb ein angepasster Sercos Parametersatz verwendet werden.

- **NofScsParSets** "Antriebsparametersätze pro Getriebestufe" (1040 00009[1])
- **PosCtrlSpSet** "Parametersatz für G63 u. lageger. Spindelbetrieb" (1040 00009 [2])

	Documentation	Instruction
Instruction:		Editieren von Maschinenparametern
Documentation:	MTX Maschinenparameter	Gewindebohren

IW-Engineering / IndraLogic: Ggf. Statusinterface-Signale verknüpfen

- Am Kanalinterface:
 - **iCh_ActFunc07** "Gewindebohren aktiv" zeigt an, ob Gewindebohren G63 aktiv ist
- Am Spindelinterface:
 - **iSp_TurnCmd** "Drehbefehl" für die Dauer von "C-Achsumschaltung aktiv".
 - **iSp_CAxSwitch** "C-Achsumschaltung" ist aktiv.
 - **iSp_CAxAct** "C-Achse ist aktiv" für die Dauer des Gewindebohrers.
- Am Achsinterface:

Für die Dauer des Gewindebohrers ist das Achsinterface von Bedeutung:

 - **iAx_TrVCmd** "Fahrbefehl" für die C-Achse und Bohrachse für die Dauer des Gewindebohrers.
 - **iAx_InPos** "Achse in Position" wird am Beginn und am Ende jeder Bohrbewegung überprüft.

	Documentation	Instruction
Instruction:		Editieren von SPS-Signalen
Documentation:	MTX SPS-Interface	Gewindebohren

Aktivieren: **IW-Operation / NC-Programmierung: Funktion Gewindebohren verwenden**



Es stehen folgende Funktionen für das Gewindebohren zur Verfügung:

- G63 und G63.2 – Gewindebohren ohne Ausgleichsfutter
- G84 – Zyklus für Gewindebohren
- G89 – Zyklus für Gewindebohren mit Spanbrechen
- TappRet1/TappRet2 – Rückzug aus Gewindebohrung bei Gewindebohren ohne Ausgleichsfutter

Die Zyklen zum Gewindebohren sind in dem Handbuch "Standard-NC-Zyklen" beschrieben.

	Documentation	
Documentation:	MTX Programmierhandbuch	Gewindebohren

Deaktivieren: Nach Beendigung des letzten G63- oder G63.2-Satzes wechselt die Spindel vom "C-Achsbetrieb" zum "Spindelbetrieb" zurück.

Damit ist Gewindebohren deaktiviert.

12.2 Gewindeschneiden

12.2.1 Beschreibung

Allgemeines

Zum Herstellen geschnittener Gewinde stehen mehrere Drehzyklen zur Verfügung.

Die im Handbuch "Standard-NC-Zyklen" beschriebenen Zyklen sind:

- **G165** Gewinde Einzelschnitt
- **G166** Gewindekette
- **G167** Gewindedrehen
- **G168** Plangewinde
- **G169** Kegelgewinde

Diese Zyklen verwenden intern die Funktion Gewindeschneiden "G33".

Funktion

Mit der Funktion Gewindeschneiden können ein- und mehrgängige Längs-, Plan- und konische Gewinde geschnitten werden. Die Gewindesteigung kann hierbei konstant, zunehmend oder abnehmend sein.

Alle Gewindetypen können innerhalb eines geketteten Gewindes verwendet werden.

Gewindeschneiden ist eine modale NC-Funktion. Sie ist zusammen mit G0, G1, G2, G3, usw. in einer NC-Funktionsgruppe.

Gewindeschneiden ist wie die Zirkularinterpolation (G2, G3) eine von der aktiven Ebene (G17 ... G20) abhängige Funktion.

Die Vorschubgeschwindigkeit der Schneidbewegung ergibt sich aus der aktuellen Spindeldrehzahl und den programmierten Steigungsanteilen (konstant und variabel). Das Vorschubpotentiometer hat keine Wirkung.

Die allgemeine Syntax lautet:

G33 <EP> <feste Steigung> {<var. Steigung>} {<Startwinkel>}	
mit	
<EP>	Achskoordinaten der Haupt- und Nebenachse der aktiven Ebene. Die aktive Ebene ist durch G17, G18, G19 oder G20 bestimmt. Beispiel: Für G18 wird die aktive Ebene üblicherweise durch die Achsen Z (Hauptachse) und X (Nebenachse) aufgespannt.
<feste Steigung>	Bestimmt den Weg (in mm), der in Richtung der Haupt- oder Nebenachse pro Spindelumdrehung zurückgelegt wird. Man programmiert den Wert mit dem in der aktiven Ebene jeweils gültigen Interpolationsparameter (I, J oder K). Bei konischen Gewinden muss sich die angegebene Gewindesteigung immer auf die Hauptschneidrichtung beziehen. Beispiel: Für G18 ist der Parameter K der Hauptachse und I der Nebenachse zugeordnet. Für ein Längsgewinde (Steigung in Hauptachsrichtung) wird die feste Gewindesteigung mit der Adresse K programmiert.
<var. Steigung>	optionaler Parameter mit der Adresse DF Definiert Zuwachs/Abnahme der Steigung pro Spindelumdrehung in mm. Programmierung: "DF<Wert>" mit <Wert> in mm.
<Startwinkel>	optionaler Parameter Ist <Startwinkel> nicht programmiert, wird 0 Grad angenommen. Der Startwinkel(-versatz) wird für mehrgängige Gewinde benötigt. Als Adresse dient derjenige Interpolationsparameter, der nicht der aktiven Ebene zugeordnet ist. Beispiel: Für G18 sind die Adressen I und K der Ebene zugeordnet. Die Adresse des Startwinkels ist somit J.

Tab. 12-8: Syntax G33

Ein Beispiel für ein Längsgewinde:

G91 G18 G8 M3 S1000	Relativmaß-Programmierung aktivieren Z/X-Ebene aktivieren
G0 X-10	Zustellbewegung des Schneidwerkzeuges (1)

G33 Z-50 K2

Gewindeschneiden (2)

Endpunkt: inkremental um -50 mm in Z-Richtung

feste Gewindesteigung: 2 mm/Umdr.

Interpolationsparameter: hier K

G0 X10

Schneidwerkzeug freifahren (3)

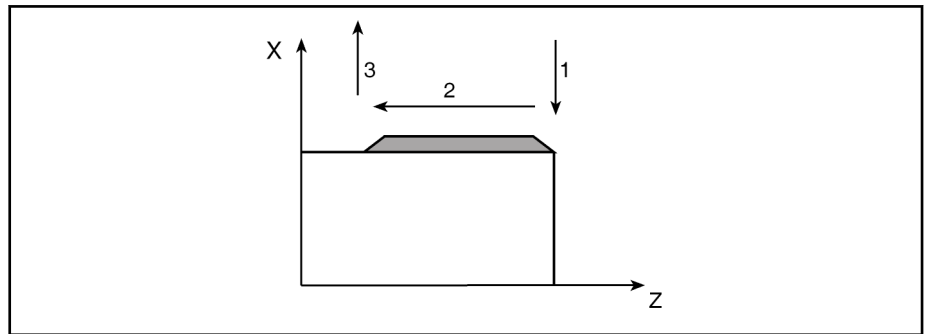


Abb. 12-3: Längsgewinde

Dynamisches Verhalten

Zu Beginn und am Ende einer Gewindebearbeitung müssen die beteiligten Achsen beschleunigt bzw. abgestoppt werden.



Planen Sie deshalb immer eine ausreichend groß bemessene Einlaufstrecke (zum Beschleunigen der Schneidachsen) und Auslaufstrecke (zum Abstoppen) ein.

Die Dynamik der Schneidbewegung kann auf 2 Arten festgelegt werden:

- Start/Ende der Schneidbewegung mit automatisch berechneter Dynamik:
Geschwindigkeitssprung, Anfahr- und Bremsbeschleunigung werden von der NC berechnet.
- Start/Ende der Schneidbewegung mit individuell eingestellter Dynamik:
Bei Bedarf kann das dynamische Verhalten bezüglich Geschwindigkeitssprung, Anfahr- und Bremsbeschleunigung individuell eingestellt werden:
 - statisch per Maschinenparameter (siehe THRD/Ch[k]/Vel, THRD/Ch[k]/Acc und THRD/Ch[k]/Dec).
 - dynamisch per Teileprogramm mit "ThreadSet(DYN)" (siehe Programmierhandbuch "Zusatzfunktionen zum Gewindeschneiden ThreadSet, TST").

Der Geschwindigkeitssprung wird von der NC automatisch begrenzt.

Die Steuerung berechnet sich aus dem programmierten Startwinkel in Abhängigkeit der Steilheit der Beschleunigungsrampe einen versetzten Startwinkel. Dadurch ist sichergestellt, dass unabhängig von der Größe der Beschleunigung immer der gleiche Gewindegang geschnitten wird.

Am Ende des Gewindes werden die Schneidachse(n) von der Spindel abgekoppelt und entsprechend der aktiven Bremsbeschleunigung abgebremst.

Schneller Rückzug

In Verbindung mit G33 ist ein "schneller Rückzug" nutzbar.

Wenn Rückzugsdaten konfiguriert und aktiviert sind

- im Teileprogramm mit "ThreadSet(RD(,),RON1))" (siehe Programmierhandbuch "Zusatzfunktionen zum Gewindeschneiden ThreadSet, TST")
- in den Maschinenparametern (siehe "THRD/Ch[k]/Retr").

löst eine positive Flanke am Kanalinterface-Signal qCh_Retract "Schneller Rückzug" die Rückzugsbewegung mit folgendem Ablauf aus:

1. Der Schneidbewegung wird eine senkrecht zur Hauptschneidrichtung orientierte Bewegung überlagert.
2. Sind mehr als 90% des Rückzugswegs abgefahren, werden die Schneidachse(n) von der Spindel abgekoppelt und abgebremst.



- Rückzugsbewegungen erfolgen immer senkrecht zur Hauptschneidrichtung in Richtung der Nebenschneidachse.
- Eine konfigurierte Rückzugsbewegung wird beim Auftreten der Ereignisse "Kanal-", "System-" und "Spindelgrundstellung" von der NC automatisch ausgelöst!



Wurde der Rückzug ausgelöst, kann dieser Zustand nur durch "Grundstellung" oder "Wegfahren von der Kontur" verlassen werden.

Ein Beispiel für eine Rückzug aus einem Längsgewinde mit Außenbearbeitung:

G18	Z/X-Ebene aktivieren (G18).
TST (RON1)	schnellen Rückzug aktivieren (RON1).
TST (RD (0, 5))	Rückzugsbewegung (RD...) um +5 mm in Nebenschneidrichtung (hier X).
:	
:	
G33 Z20 K1	Gewindeschneiden (G33)

Relevante NC-Funktionen

G33	Gewindeschneiden mit Angabe des Endpunkts, der Steigung und des Startwinkels.
MainSp Kurzform: MSP	Hauptspindelumschaltung Gewindeschneiden verwendet immer die aktive Hauptspindel
ThreadSet Kurzform: TST	Konfigurieren einer Rückzugsbewegung Rückzugsbewegung konfigurieren. Dynamik am Beginn und Ende der Schneidbewegung beeinflussen.

Tab. 12-9: Relevante NC-Funktionen

G165	Gewinde Einzelschnitt
G166	Gewindekette
G167	Gewindedrehen
G168	Plangewinde
G169	Kegelgewinde

Tab. 12-10: Relevante NC-Zyklen

Relevante Sercos Parameter

S-0-1050.1.6	Konfigurationsliste AT Die Istposition der Spindel muss übertragen werden.
S-0-0043 und S-0-0055	Geschwindigkeits-Polarität und Lage-Polarität der Spindel müssen gleich sein.

Tab. 12-11: Sercos Parameter

Relevante IF-Signale

qCh_CtrlReset	Grundstellung Wenn ein schneller Rückzug konfiguriert ist, dann wird dieser ausgelöst und anschließend das Teileprogramm beendet.
qCh_Retract	Schneller Rückzug Wenn ein schneller Rückzug konfiguriert ist, dann kann dieser über dieses Signal ausgelöst werden.
qCh_Override100	Dieses Signal wird bei aktivem G33 nicht ausgewertet.
qCh_Override_xx	Diese Signale werden bei aktivem G33 nicht ausgewertet.
iCh_ActFunc06	Die Standardbelegung dieses Signals ist "TST" (laut PLC/NcFuncBitIf/NcFunc[1..24]) Dieses Signal kann von ThreadSet(TCI1) gesetzt werden.
iCh_ActFunc08	Die Standardbelegung dieses Signals ist "G33" (laut PLC/NcFuncBitIf/NcFunc[1..24]) Dieses Signal wird bei aktivem G33 gesetzt.

Tab. 12-12: Relevante IF-Signale am Kanalinterface

qSp_Reset	Spindelgrundstellung Wenn ein schneller Rückzug konfiguriert ist, dann wird dieser ausgelöst.
qSp_Override100	Default: Eine Änderung dieses Signals wird bei aktivem G33 nicht ausgewertet. Option: Durch Setzen von "/SysSpCmdData[]/ThreadOverride = 1" kann die Auswertung dieses Signals eingeschaltet werden.
qSp_Override_xx	Default: Eine Änderung dieser Signale wird bei aktivem G33 nicht ausgewertet. Option: Durch Setzen von "/SysSpCmdData[]/ThreadOverride = 1" kann die Auswertung dieser Signale eingeschaltet werden.

Tab. 12-13: Relevante IF-Signale am Spindelinterface

Relevante Maschinenparameter (MP)

1040 00001	Spindelart (SP/SpFunc/Sp[1..32]/Base/SpType) Die Funktion Gewindeschneiden (G33) erfordert eine Spindel mit Positionsrückführung (SpType 1=Sercos, 4=Extern mit Geber oder 5=Analog mit Geber).
7020 00010	Hauptspindel (CHAN/Ch[k]/Couple/MainSp) Die Hauptspindel kann im Teileprogramm mit "MainSp()" verändert werden.
Für Sonderfälle von Bedeutung sind:	
7050 00610	Geschwindigkeitssprung (THRD/Ch[k]/Vel/VelJumpThrd) Legt den maximalen Geschwindigkeitssprung am Anfang und am Ende der Schneidbewegung fest.
7050 00615	Bahnbeschleunigung (THRD/Ch[k]/Acc/AccThrd) Gibt die Bahnbeschleunigung am Beginn der Schneidbewegung an. Im Allgemeinen ergibt der Defaultwert (0.000m/s2) das beste Ergebnis.
7050 00620	Bahnbremsbeschleunigung (THRD/Ch[k]/Dec/DecThrd) Gibt die Bahnbremsbeschleunigung am Ende der Schneidbewegung an. Im Allgemeinen ergibt der Defaultwert (0.000m/s2) das beste Ergebnis.
7050 00640	Aktiviere Schnellrückzug (THRD/Ch[k]/Retr/EnablRetrThrd) Dieser Parameter kann zur Laufzeit durch "ThreadSet(ROn)" überblendet werden.
7050 00645	Achsnummer der Hauptachse (THRD/Ch[k]/Retr/DrIndRetrAx) und Achsnummer der Nebenachse (THRD/Ch[k]/Retr/DrIndSeco-nAx) Legt die Rückzugsachsen fest. Diese Achsen müssen innerhalb des Kanals eine Ebene repräsentieren. Dieser Parameter und MP 705000650 können zur Laufzeit durch "ThreadSet(RD(,))" überblendet werden.
7050 00650	Rückzugsvektor der Hauptachse (THRD/Ch[k]/Retr/RetrVect-RetrAx) und Rückzugsvektor der Nebenachse (THRD/Ch[k]/Retr/RetrVect-Seco-nAx) Legt für die in 705000645 festgelegten Achsen den Rückzugsweg, inkl. Vorzeichen, fest. Dieser Parameter und MP 705000645 können zur Laufzeit durch "ThreadSet(RD(,))" überblendet werden.

Tab. 12-14: Relevante Maschinenparameter

12.2.2 Relevante Systemdaten

/SysSpCmdData[<Spindle>]/ThreadOverride	1: Während des Gewindeschneidens wirkt der Spindel-Override.

Tab. 12-15: Relevante Systemdaten



Beachten Sie bitte, dass bei aktiviertem Spindeloverride eine schnelle Änderung des Override-Werts während der Bearbeitung zu einer Beschädigung des Gewindes führen kann.

12.2.3 Applizieren

Normalerweise müssen für die Funktion Gewindeschneiden keine besonderen Schritte durchgeführt werden.

Die Defaultwerte der Maschinenparameter "Gewindeschneiden (THRD)" (siehe THRD/Ch[k]/) können beibehalten werden.

Voraussetzung:

Die Funktion Gewindeschneiden (G33) erfordert eine Spindel mit Positionsrückführung (SpType 1=Sercos, 4=Extern mit Geber oder 5=Analog mit Geber).

12.2.4 Aktivieren

Es stehen folgende Funktionen für das Gewindeschneiden zur Verfügung:

- G33 "Gewindeschneiden"
- G165 "Gewinde Einzelschnitt"
- G166 "Gewindekette"
- G167 "Gewindedrehen"
- G168 "Plangewinde"
- G169 "Kegelgewinde"

Die Zyklen zum Gewindeschneiden (G165 ... G169) sind in dem Handbuch "Standard-NC-Zyklen" beschrieben.

Der normale Ablauf ist:

1. Hauptspindel starten
2. Startposition anfahren
3. Gewindeschneiden G33 oder einen geeigneten Drehzyklus ausführen.

Wenn für den Notfall die Funktion "Schneller Rückzug" verwendet werden soll, dann müssen mit der Funktion "ThreadSet" Rückzugsdaten konfiguriert und aktiviert werden.



Die Rückzugsbewegung muss vom Werkstück wegführen (Vorzeichen beachten).

12.2.5 Deaktivieren

Die Anwahl einer beliebigen anderen Geometriefunktion (G0, G1, G2, G3, G6) wählt die Funktion Gewindeschneiden G33 ab.

12.3 Zirkulares Eckenrunden

12.3.1 Beschreibung

Funktion

Die Steuerung kann mit Hilfe der Funktionen "Rounding (RND)" und "RoundEps (RNE)" tangential zwischen Linear- oder Zirkular-/Helicalsätzen in der aktiven Ebene Übergangsbögen einfügen.

Auf diese Weise wird zwar einerseits die programmierte Kontur am Satzübergang geringfügig modifiziert, andererseits kann man aber (bei genügend großem Radius) kontinuierlichere Geschwindigkeits- und Beschleunigungsverläufe auf der Bahn erreichen.

Unter welchen Bedingungen die Steuerung Übergangsbögen einfügen soll, ist per Maschinenparameter "MinLen" und "MaxAng" einstellbar.

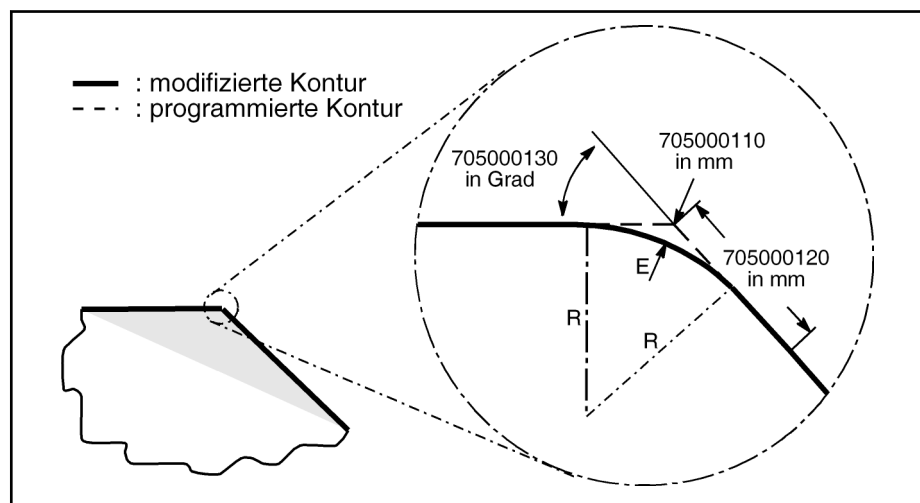


Abb. 12-4: Zirkulares Eckenrunden

Restriktionen

- RoundEps (RNE) lässt sich nur in Verbindung mit zwei Linearsätzen einsetzen, während Rounding (RND) zusätzlich auch für Zirkularsätze und Helicalsätze wirkt.
- Bei aktivem Eckenverrunden kann nicht gestanzt werden.
- Beide benachbarten Sätze müssen in Verbindung mit RNE in der ausgewählten Hauptebene liegen, um die Verrundung durchzuführen. Bei Helicalsätzen werden nur die Komponenten der Kreisebene für die Verrundung berücksichtigt.

Relevante NC-Funktionen

Rounding (RND)	"Ecken verrunden" zwischen Linear- oder Zirkularsätzen einschalten. Der gewünschte Radius des Übergangsbogens wird beim Aufruf von Rounding/RND in Klammern hinter der Funktion programmiert (z. B. Rounding(5)). Er wirkt solange, bis er durch erneute Programmierung überschrieben wird.
RoundEps/RNE	"Ecken verrunden" zwischen 2 Linearsätzen einschalten. Die maximal zulässige Abweichung zwischen modifizierter und programmierter Kontur wird in runden Klammern hinter der Funktion programmiert (z. B. RoundEps(2)). Soll die Voreinstellung aus Maschinenparameter MaxDev wirken, ist innerhalb der runden Klammer das Schlüsselwort "DEF" zu programmieren.
Rounding(0) oder RND(0) bzw. RoundEps(0) oder RNE(0)	
	"Ecken verrunden" ausschalten. Beide Funktionen sind identisch und wählen alle Varianten von Fasen- und Rundungsprogrammierung ab.

Tab. 12-16: Relevante NC-Funktionen



Die NC-Funktionen "Rounding/RND" und "RoundEps/RNE" bilden zusammen mit den Funktionen "ChLength/CHL" und "ChSection/CHS" eine modale Gruppe und wählen sich deshalb gegenseitig ab.

Relevante Maschinenparameter

NCP/CornRound/CH[x]/MaxDev (705000110)	
	Maximal zulässige Abweichung von der programmierten Kontur. Daraus errechnet die Steuerung automatisch den passenden Radius des Übergangsbogens.
NCP/CornRound/CH[x]/MinLen (705000120)	
	Mindestsatzlänge, ab welcher verrundet wird. Ist einer der benachbarten Verfahrssätze kleiner, wird kein Übergangsbogen erzeugt.
NCP/CornRound/CH[x]/MaxAng (705000130)	
	Maximalwinkel, bis zu welchem Satzübergänge als quasikontinuierlich gelten. Bei kleineren Winkeln wird kein Übergangsbogen erzeugt.

Tab. 12-17: Relevante Maschinenparameter

12.3.2 Handlungsanweisung: Ecken verrunden mit Angabe der Abweichung applizieren und aktivieren/deaktivieren

Diese Funktion fügt zwischen 2 Linearsätzen in der Hauptebene tangentielle Übergangsbögen ein. Dadurch wird zwar die programmierte Kontur an diesen Ecken geringfügig modifiziert, für die Interpolation ergeben sich jedoch kontinuierliche Geschwindigkeitsverläufe.

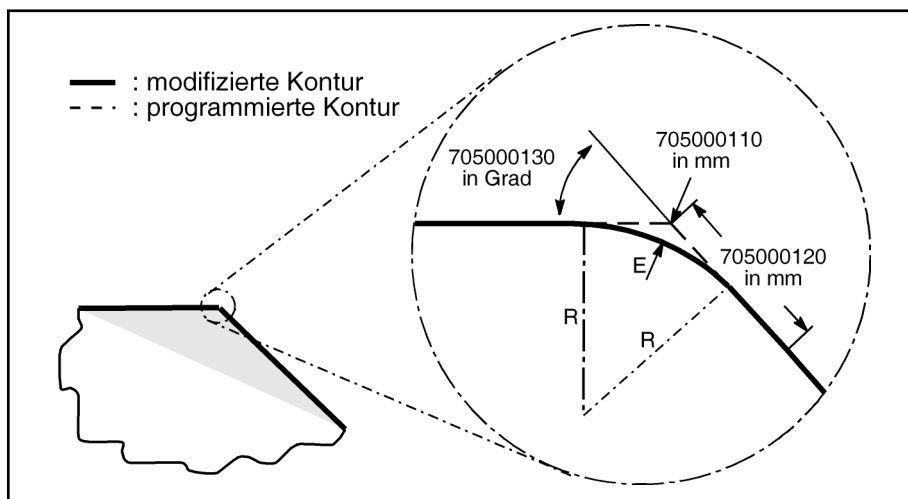


Abb. 12-5: Skizze zu Ecken verrunden

Applizieren: IW-Engineering / Configuration: Parameter editieren

- Aus dem Parameter **MaxDev** "Maximal zulässige Abweichung von der Kontur" (7050 00110) errechnet die Steuerung automatisch den passenden Radius des Übergangsbogens.
- Der Parameter **MinLen** "Mindestsatzlänge, ab welcher verrundet wird" (7050 00120) enthält die Mindestsatzlänge, ab welcher verrundet wird. Falls einer der benachbarten Verfahrssätze kleiner ist, wird kein Übergangsbogen erzeugt.
- Der Parameter **MaxAng** "Winkel, bis zu dem Satzübergänge als kont. gelten" (7050 00130) enthält den Maximalwinkel, bis zu welchem Satzübergänge als quasikontinuierlich gelten. Bei kleineren Winkeln wird kein Übergangsbogen erzeugt.

	Documentation	Instruction
Instruction:		Editieren von Maschinenparametern
Documentation:	MTX Maschinenparameter	Ecken verrunden

IW-Operation / Maschine: Systemgrundstellung auslösen

Zur Übernahme der geänderten Maschinenparameter ist eine Systemgrundstellung oder ein Steuerungshochlauf notwendig.

Aktivieren:

HINWEIS

Schäden an der Maschine oder am Werkstück möglich!

Während der Bearbeitung darf keine Systemgrundstellung ausgelöst werden.

IW-Operation / NC-Programmierung: aktivieren

- **RoundEps(<Abweichung>)** bzw. **RNE(<Abweichung>)** "Ecken verrunden mit Angabe der Abweichung"
 <Abweichung> Maximal zulässige Abweichung (in mm) zwischen modifizierter und programmierter Kontur. Nachkommastellen sind erlaubt. Die Steuerung berechnet sich einen passenden tangentialen Übergangsbogen selbst.
- oder -
- **RoundEps(DEF)** Ecken verrunden zwischen 2 Linearsätzen einschalten mit Default-Abweichung aus dem Parameter **MaxDev** "Maximal zulässige Abweichung von der Kontur" (7050 00110).



Die Funktionen **ChLength()**, **ChSection()**, **RoundEps()** und **Rounding()** wirken modal und wählen sich gegenseitig ab.

Es werden nur Konturübergänge von Sätzen berücksichtigt, deren Bahn ausschließlich innerhalb der aktiven Arbeitsebene liegt.

	Documentation	
Documentation:	MTX Programmierhandbuch	Ecken verrunden

Deaktivieren: **IW-Operation / NC-Programmierung: deaktivieren**
RoundEps(0) bzw. **RNE(0)**

	Documentation	
Documentation:	MTX Programmierhandbuch	Ecken verrunden

12.3.3 Handlungsanweisung: Ecken verrunden mit Radiusangabe applizieren und aktivieren/deaktivieren

Diese Funktion fügt zwischen 2 Linear-, Zirkular- bzw. Helicalsätzen in der Hauptebene tangential Übergangsbögen ein. Dadurch wird zwar die programmierte Kontur an diesen Ecken geringfügig modifiziert, für die Interpolation ergeben sich jedoch kontinuierliche Geschwindigkeitsverläufe.

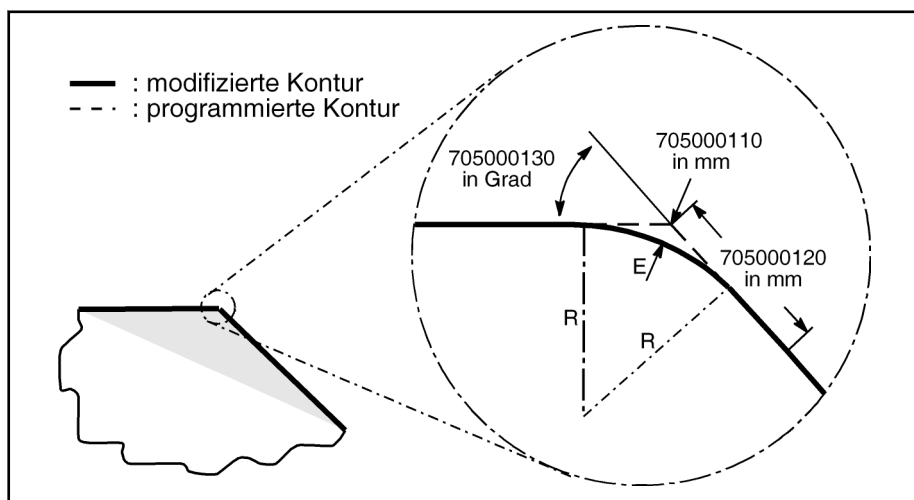


Abb. 12-6: Skizze zu Ecken verrunden

Applizieren: **IW-Engineering / Configuration: Parameter editieren**

- Aus dem Parameter **MaxDev** "Maximal zulässige Abweichung von der Kontur" (7050 00110) errechnet die Steuerung automatisch den passenden Radius des Übergangsbogens.
- Der Parameter **MinLen** "Mindessatzlänge, ab welcher verrundet wird" (7050 00120) enthält die Mindestsatzlänge, ab welcher verrundet wird. Falls einer der benachbarten Verfahrssätze kleiner ist, wird kein Übergangsbogen erzeugt.
- Der Parameter **MaxAng** "Winkel, bis zu dem Satzübergänge als kont. gelten" (7050 00130) enthält den Maximalwinkel, bis zu welchem Satzübergänge als quasikontinuierlich gelten. Bei kleineren Winkeln wird kein Übergangsbogen erzeugt.

	Documentation	Instruction
Instruction:		Editieren von Maschinenparametern
Documentation:	MTX Maschinenparameter	Ecken verrunden

IW-Operation / Maschine: Systemgrundstellung auslösen

Zur Übernahme der geänderten Maschinenparameter ist eine Systemgrundstellung oder ein Steuerungshochlauf notwendig.

Aktivieren:

HINWEIS

Schäden an der Maschine oder am Werkstück möglich!

Während der Bearbeitung darf keine Systemgrundstellung ausgelöst werden.

IW-Operation / NC-Programmierung: aktivieren

Rounding(<Radius>) bzw. **RND(<Radius>)** "Ecken verrunden mit Radiusangabe"

<Radius> Gewünschter Radius des Übergangskreisbogens, Nachkommastellen sind erlaubt.



Die Funktionen **ChLength()**, **ChSection()**, **RoundEps()** und **Rounding()** wirken modal und wählen sich gegenseitig ab.

Es werden nur die Komponenten der aktiven Arbeitsebene für die Verrundung berücksichtigt. Dies führt bei räumlichen Geraden dazu, dass sich die Richtung im Raum ändert. Ähnliches gilt sinngemäß für Helical-Bahnsegmente.

	Documentation	
Documentation:	MTX Programmierhandbuch	Ecken verrunden aktivieren

Deaktivieren:

IW-Operation / NC-Programmierung: deaktivieren

Rounding(0) bzw. **RND(0)**

	Documentation	
Documentation:	MTX Programmierhandbuch	Ecken verrunden

12.4 Eckenrunden mit Splines

12.4.1 Beschreibung

Funktion

Die NC-Funktionen "Eckenrunden mit Splines" verrunden alle Konturübergänge, deren Knickwinkel größer gleich einem konfigurierten Minimalknickwinkel sind, mit einem Spline. Hierfür stehen eine modale und eine nichtmodale (lokale) NC-Funktion zur Verfügung.

Eigenschaften:

- Die modale NC-Funktion bleibt nach ihrer Aktivierung solange aktiv, bis sie wieder deaktiviert wird.

Die lokale NC-Funktion ist nur für den Übergang von dem Satz, in dem sie programmiert ist, zum nächsten Satz aktiv, und macht damit die modale NC-Funktion, falls diese ebenfalls aktiv ist, für diesen einen Übergang unwirksam.

- Die Übergänge am Spline-Start- und -Endpunkt sind c^2 -stetig, d. h. beim Überfahren mit konstanter Bahngeschwindigkeit treten keine Achsbeschleunigungssprünge auf.
- Es kann eine maximale Abweichung "e" der Spline-Kontur von der Ecke angegeben werden (siehe [Abb. 12-7 "Eckenrunden bei Vorgabe einer maximalen Toleranz" auf Seite 445](#)). Die Steuerung erzeugt dann einen symmetrischen Spline der eine Strecke "l" vor der Ecke (auf dem Voreckensatz) beginnt und nach derselben Strecke hinter der Ecke (auf dem Nacheckensatz) endet. Ist eine der beiden Eckenkonturen zu kurz, wird sowohl die Vor- als auch die Nacheckenstrecke auf ein gemeinsames Maß l gekürzt. l bleibt immer kleiner oder gleich als die Hälfte der kürzeren der beiden Eckenkonturen, d. h. $l \leq \frac{1}{2} \min(S_1, S_2)$, mit den Bahnlängen S_1 und S_2 des Vor- bzw. Nacheckensatzes.

Es ist möglich gleichzeitig e und l (s. unten) vorzugeben. Dann wird in jedem Falle die Vor- und Nacheckenstrecke auf den Wert "L" begrenzt.

- Wahlweise können an Stelle einer Toleranz e Vor- und Nacheckenstrecken l_1 bzw. l_2 oder eine symmetrische Vor-/Nacheckenstrecke L auf den ursprünglichen Konturen angegeben werden (s. Abbildung "Eckenrunden bei Vorgabe einer Vor- und Nacheckenstrecke l_1 bzw. l_2 ").
- Das Verrunden erfolgt bei jedem Konturtyp.

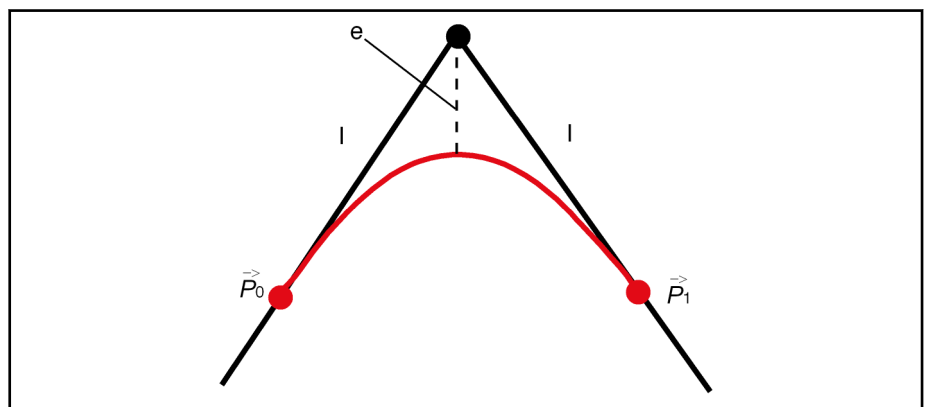


Abb. 12-7: Eckenrunden bei Vorgabe einer maximalen Toleranz

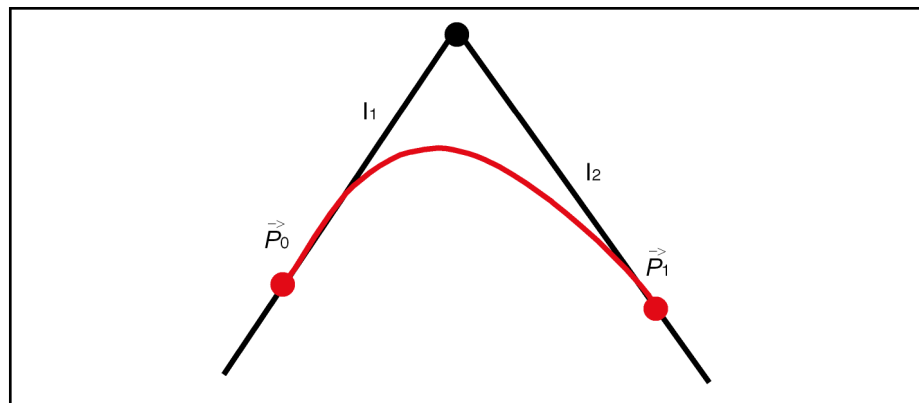


Abb. 12-8: Eckenrunden bei Vorgabe einer Vor- und Nacheckenstrecke l_1 bzw. l_2

Das Runden wird nicht durchgeführt, wenn eine der folgenden Bedingungen erfüllt ist:

- Die Vor- oder Nacheckenkontur hat eine Bahnlänge, die kleiner ist als der Wert im Maschinenparameter MinLen. Insbesondere ist dies auch bei Nichtverfahrensätzen der Fall.
- Die Bewegung in den beiden Eckensätzen gehören zu unterschiedlichen Vorschub bildenden Koordinatengruppen.

Beispiel:

Der Voreckensatz enthält eine Bewegung in den Koordinaten x, y, z und der Nacheckensatz eine reine Orientierungsbewegung phi, theta.

- Der Knickwinkel ist kleiner als der Wert im Maschinenparameter MaxAng (quasitangentialer Übergang).
- Es liegt eine Richtungsumkehr vor.

Besonderheiten:

Bei zu großen Eckenstrecken ergeben sich jeweils in der Mitte der ursprünglichen Eckensätze "Nichtverfahrensätze" (siehe nachfolgende Abbildung).

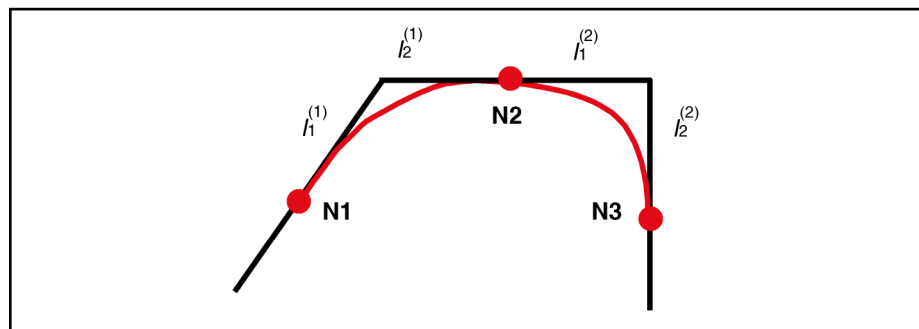


Abb. 12-9: Aufeinanderfolgende Ecken

$l_2^{(1)}$ und $l_1^{(2)}$ sind auf die Hälfte der N2-Bahnlänge gekürzte Strecken.

Restriktionen

Die gleichzeitige Vorgabe von Eckenstrecken l_1 , l_2 und einer Eckentoleranz e ist nicht möglich. Die gleichzeitige Vorgabe von l und e ist jedoch erlaubt

Relevante modale NC-Funktion

SplineCornering(E<Abstand>)

Ecken runden ein (Aktivierung) mit Angabe einer maximal zulässigen Toleranz E.

SplineCornering(DEF)

Ecken runden ein (Aktivierung) mit Toleranz E aus Maschinenparameter MaxDev.

SplineCornering(L1=<L1>,L2=<L2>)

Ecken runden ein (Aktivierung) mit Angabe einer Voreckenstrecke L_1 und Nacheckenstrecke L_2 .

SplineCornering(L<L>)

Eckenrunden ein mit Angabe einer symmetrischen Vor- und Nacheckenstrecke $L_1 = L_2 = L$.

SplineCornering(L<L>,E<Abstand>)

Eckenrunden ein mit Angabe eines maximal zulässigen Abstands E zwischen Eckpunkt und Spline bei gleichzeitiger Begrenzung der Vor- und Nacheckenstrecke auf den Wert L.

SplineCornering(0)

Eckenrunden aus (Deaktivierung)

SCO

Kurzform

Relevante lokale (nichtmodale) NC-Funktion

SplineCorneringLocal(E<Abstand>)

Ecken runden für den Übergang zum nächsten NC-Satz mit Angabe einer maximal zulässigen Toleranz E.

SplineCorneringLocal(DEF)

Ecken runden für den Übergang zum nächsten NC-Satz mit Toleranz E aus Maschinenparameter MaxDev.

SplineCorneringLocal(L1=<L1>,L2=<L2>)

Ecken runden für den Übergang zum nächsten NC-Satz mit Angabe einer Voreckenstrecke L_1 und Nacheckenstrecke L_2 .

SplineCorneringLocal(L<L>)

Eckenrunden ein mit Angabe einer symmetrischen Vor- und Nacheckenstrecke $L_1 = L_2 = L$.

SplineCorneringLocal(L<L>,E<Abstand>)

Eckenrunden ein mit Angabe eines maximal zulässigen Abstands E zwischen Eckpunkt und Spline bei gleichzeitiger Begrenzung der Vor- und Nacheckenstrecke auf den Wert L.

SplineCorneringLocal(0)

Kein Eckenrunden für den Übergang zum nächsten NC-Satz.

SCOL

Kurzform

Wechselwirkung zwischen modaler und lokaler NC-Funktion

Wird in einem NC-Satz, in dem die modale NC-Funktion aktiv ist, die lokale NC-Funktion programmiert, wirkt beim Übergang zum nächsten Satz nur die lokale. Die modale bleibt zwar weiterhin aktiv, ist jedoch an diesem Übergang unwirksam.

Relevante Maschinenparameter

Es werden die gleichen Maschinenparameter wie beim zirkularen Runden benutzt:

NCP/CornRound/CH[x]/MaxDev (705000110)	
	Maximal zulässige Abweichung von der programmierten Kontur. Ist-Einschaltzustand für die Toleranz E.
NCP/CornRound/CH[x]/MinLen (705000120)	
	Mindestsatzlänge, ab welcher verrundet wird. Ist einer der benachbarten Verfahrssätze kleiner, wird nicht gerundet.
NCP/CornRound/CH[x]/MaxAng (705000130)	
	Maximalwinkel, bis zu welchem Satzübergänge als quasikontinuierlich gelten. Bei kleineren Knickwinkeln wird nicht gerundet.

Tab. 12-18: Relevante Maschinenparameter

12.4.2 Handlungsanweisung: Eckenrunden mit Splines

Nutzung der NC-Funktion im NC-Programm

1. Schritt NC-Konfigurator: Maschinenparameter editieren

- Maschinenparameter MaxDev "Maximal zulässige Abweichung von der programmierten Kontur" ausfüllen. Dies ist nur notwendig, wenn SCO(DEF) benutzt werden soll.
- Maschinenparameter MinLen "Mindestsatzlänge, ab welcher verrundet wird" und MaxAng "Maximalwinkel, bis zu welchem Satzübergänge als quasikontinuierlich gelten" ausfüllen.



Diese Parameter gelten gleichermaßen für das zirkulare Eckenrunden.

2. Schritt NC-Programm

- Im NC-Programm die Funktion mit SCO(E<Toleranz>), SCO(DEF) oder SCO(L1=<L1>,L2=<L2>) einschalten und mit SCO() ausschalten.

Beispiel:

1 - Im Kanal befinden sich die Linearachsen X, Y, Z, sowie die Rundachse A.

NC-Sätze	Bemerkung
N0 FeedGrp(X,Y,Z,A)	Vorschubgruppe ist X,Y,Z,A
N1 X0 Y0	
N2 G91 SCO(E0.1)	Eckenrunden ein mit Toleranz 0,1 mm Kein Runden bei N2/N3, da N2 Bahnlänge = 0 hat
N3 X10	Runden bei N3/N4 mit Toleranz 0,1 mm

NC-Sätze	Bemerkung
N4 Y10 SCOL(DEF)	Runden bei N4/N5 mit Toleranzwert aus MP 705000110
N5 X20	Runden bei N5/N6 mit Toleranz 0,1 mm
N6 Y0 SCOL(0)	Kein Runden bei N6/N7
N7 X30	Runden bei N7/N8 mit Toleranz 0,1 mm
N8 Y10 SCOL(L1=1,L2=2)	Kein Runden bei N8/N9 wegen tangentialem Übergang
N9 Y10	Runden bei N9/N10 mit Toleranz 0,1 mm Bem.: Bei FeedGrp(X,Y,Z) würde nicht gerundet werden
N10 Y10 A10	Kein Runden bei N10/N11, da N11 Bahnlänge 0 hat
N11 SCO()	Eckenrunden aus

Tab. 12-19: Beispiel 1

Beispiel:

2 - Im Kanal befinden sich die Raumkoordinaten x, y, z, phi, theta.

NC-Sätze	Bemerkung
N0 FeedGrp(x,y,z)	Vorschubgruppe ist x,y,z
N1 G90 x0 y0 z0 phi0 theta45	Ausgangszustand
N2 SCO(DEF) x10	Eckenrunden ein mit Toleranzwert aus MP705000110 Runden bei N2/N3
N3 y10 phi0 theta50	Runden bei N3/N4 Knick durch xy-Bewegung
N4 x-10 phi0 theta70	Kein Runden bei N4/N5 Unterschiedliche Vorschub bildende Koordinatengruppen
N5 O(1,0,0)	Runden bei N5/N6 Knick durch Orientierungsbewegung
N6 O(0,1,0)	Kein Runden bei N6/N7 N7 hat Bahnlänge 0
N7 SCO()	Eckenrunden aus

Tab. 12-20: Beispiel 2

Beispiel:

3 - Schnelles Umfahren von Ecken (Überschleifen)

NC-Sätze	Bemerkung
N0 G91	inkrementelle Programmierung
N1 SCO(L1=5,L2=15) Y20	Spline-Beginn bei Y=15

NC-Sätze	Bemerkung
N2 SCO(L1=15,L2=5) X100	Spline-Ende bei X=15 Spline-Beginn bei X=85
N3 SCO() Y-20	Spline-Ende bei Y=15

Tab. 12-21: Beispiel 3

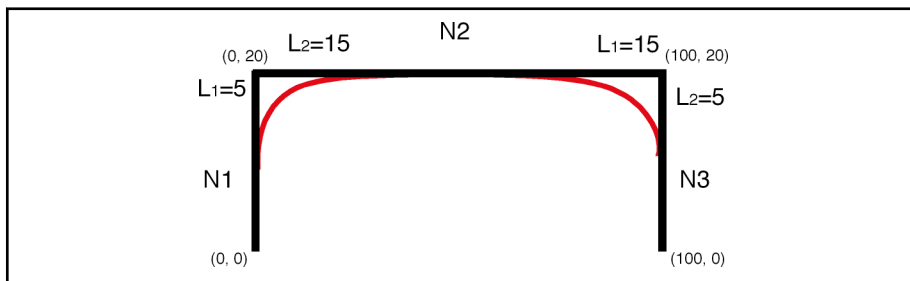


Abb. 12-10: Eckenrunden zum schnellen Umfahren von Ecken (Überschleifen)

12.5 Stanzen und Nibbeln

12.5.1 Beschreibung

Funktion

Erweitert die NC um verschiedene Funktionen, die für deren Einsatz in Stanz- bzw. Nibbelmaschinen mit hydraulischen Stößeln erforderlich sind.

Merkmale:

- Programmgesteuertes Ein- und Ausschalten von Stanzen/Nibbeln durch die NC-Funktionen "Nibble/NIB" (Nibbeln) und "Punch/PUN" (Stanzen). Die Funktionen bilden eine modale Gruppe, d. h. zu einem Zeitpunkt kann nur eine der Funktionen aktiv sein.
- Automatische Zerlegung eines programmierten Verfahrenssatzes in lineare Teilstrecken mit gleicher Länge (Satzzerlegung). Die Endpunkte der Teilstrecken befinden sich hierbei immer auf der programmierten Bahn.
- Automatische tangentielle Anstellung des Stanz-/Nibbelwerkzeuges an die programmierte Bahn am Satzende per Funktion "TangToolOri/TTO" (Tangentiale Werkzeugorientierung). Siehe ab [Kap. 12.6 "Tangentiale Werkzeugorientierung"](#) auf Seite 455.
- Externe Stanzeinheit/Stanzsteuerung: Bereitstellen eines schnellen Ein- und Ausgangssignals zur Kommunikation mit einer hubansteuernden Stanzsteuerung über die "Schnittstelle für schnelle Signalankoppelung" (Highspeed I/O).
- Interne Stanzeinheit: Verwenden von permanenten CPL-Variablen zur Kommunikation mit dem huberzeugenden Kanal.
- Der Signalverlauf zwischen NC und Stanzsteuerung kann NC-seitig durch eine Hubzeitüberwachung kontrolliert werden.
- NC-interne High-Speed-Logik zum Informationsaustausch zwischen Stanzsteuerung, NC und SPS.
- Hubanforderung sowohl durch die NC, als auch durch die SPS möglich. Zur Kommunikation mit der SPS stellt die NC am Globalinterface verschiedene Signale zur Verfügung. Unter anderem kann die SPS dadurch auch eine Hubsperrung auslösen.
- Verschiebung der Hubauslösezeiten durch die NC (Hubvor- oder Hubnachausslösung). Als Zeitreferenz lässt sich entweder das Erreichen des

Endpunktes einer Verfahrbewegung durch den Interpolator, oder das Erreichen des Inpos-Fensters aller Achsen definieren.

- Auslassen von Stanzhüben durch Setzen eines Steuerdatums. Solange das Systemdatum /SysPunch/Cmd/doSkipPunch den Wert 1 hat, werden Stanzhübe ausgelassen: Die Stanzeinheit erhält keine Signale, das Programm wird ohne Stanzhub fortgesetzt.

Stanzen und Nibbeln unterscheiden sich in folgenden Punkten:

Funktion	beim Stanzen	beim Nibbeln
Satzerlegung	bei Bedarf möglich	erforderlich
Hubauslösung am Satzende	nur am Ende des programmierten Verfahrssatzes erforderlich	am Ende jeder Teilstrecke erforderlich
Hubauslösung am Anfang eines programmierten Verfahrssatzes	nicht erforderlich	Nur erforderlich, wenn der Vorgängersatz kein Verfahrssatz war, oder im Vorgängersatz weder Stanzen noch Nibbeln aktiv war.

Tab. 12-22: Funktion beim Stanzen und beim Nibbeln

Bei aktivem Stanzen oder Nibbeln wird in der Regel an einem Satzende immer ein Hub ausgelöst.

Ausnahmen:

- Der aktuelle Satz enthält keine Achskoordinate in der aktiven Ebene. Die Bearbeitung wird ohne Hub mit dem nächsten Satz fortgesetzt.
- Die SPS unterdrückt die Hubauslösung. Die Bearbeitung wird in diesem Fall erst nach Freigabe durch die SPS mit dem Hub fortgesetzt.
- Die Funktion Hub überspringen (Skip stroke) ist aktiv. Die Bearbeitung wird in diesem Fall ohne Hub und Quittierung fortgesetzt.

Restriktionen

- Nibbeln und Stanzen eignet sich nicht für Werkzeugmaschinen mit freilaufendem Stößel.
- Nibbeln und Stanzen sind bei aktivem Eckenverrunden nicht erlaubt.
- Nibbeln oder Stanzen ist nur einschaltbar, wenn MP 8001 00010 (Funktion "Stanzen appliziert") mit dem Wert "1" belegt ist.
- Nibbeln und Stanzen kann zu einem Zeitpunkt nur in einem einzigen Kanal verwendet werden.
- Voraussetzung für Stanzen/Nibbeln ist eine stanzfähige Geschwindigkeitsinterpolation (G9 mit SHAPE).
- Externe Stanzsteuerung: Schnittstelle für schnelle Signalan Kopplung erforderlich ("Highspeed I/O").
- Je ein schneller Digitalein- und -ausgang ist zur Kommunikation mit der Stanzsteuerung notwendig (konfigurierbar per MP 4075 00101 und MP 4075 00102).
- Interne Stanzeinheit (elektrischer Stanzkopf): Kein Digitaleingang und Digitalausgang, sondern Anbindung über permanente CPL-Variablen erforderlich:

Die permanenten Variablen @HSIN und @HSOUT müssen angelegt sein.

Relevante NC-Funktionen

Punch/PUN	Stanzen einschalten
Nibble/NIB	Nibbeln einschalten Die Satzzerlegung muss zum Nibbeln aktiviert sein ("LEN=" > 0).
PtInpos/PTI	Festlegung achsspezifischer Hubauslösezeiten bezogen auf die Zeitreferenz "Inpos-Fenster erreicht". Hubvorauslösung ist hier nicht möglich.
PtBikEnd/PTE	Festlegung achsspezifischer Hubauslösezeiten bezogen auf die Zeitreferenz "Interpolator-Endpunkt erreicht". Sowohl Hubvorauslösung, als auch Hubnachausslösung möglich.
PtDefault/PTD	Hubauslösezeiten und Zeitreferenz aller Achsen auf die Maschinenparameter-Werte setzen (MP 8001 00020 und MP 8001 00021).

Tab. 12-23: Relevante NC-Funktionen

Relevante IF-Signale

HSO-0	schneller Digital-Ausgang (einstellbar per MP 4075 00102). NC fordert Hubauslösung
HSI-0	schneller Digital-Eingang (einstellbar per MP 4075 00101). Stanzsteuerung quittiert
iGen_StrokeIntend	"Hub beabsichtigt" (Globalinterface) Die NC zeigt der SPS an, dass sie einen Hub auslösen möchte.
iGen_NoStroke	"Hub läuft nicht" (Globalinterface) Abbild von HSI-0 für die SPS.
qGen_StrokeInhibit	"Hubsperrung" (Globalinterface) SPS verhindert das Setzen von HSO-0.
qGen_StrokeReserv	"Hubreservierung" (Globalinterface) SPS reserviert sich HSO-0 für eigene Hubauslösung.
qGen_StrokeRel	"Hub ein" (Globalinterface) SPS beauftragt die NC, einen Hub auszulösen.

Tab. 12-24: Relevante IF-Signale

Relevante Maschinenparameter (MP)

4075 00101	Zuordnung der Digitaleingänge
4075 00102	Zuordnung der Digitalausgänge
8001 00010	Funktion Stanzen appliziert

8001 00020	achsspezifische Hubauslösezeiten
8001 00021	Referenz der Hubauslösezeiten
8001 00030	Hubüberwachungszeit

Tab. 12-25: Relevante Maschinenparameter

Relevante Systemdaten

/SysPunch/Cmd/doSkipPunch	=1 Hübe werden übersprungen
/SysPunch/Cmd/ suppressSkipMsg	=1 keine Warnung, wenn Funktion "Hübe überspringen" aktiv ist

Tab. 12-26: Relevante Systemdaten

12.5.2 Handlungsanweisung: Stanzen und Nibbeln aktivieren

Das Stanz- Nibbelpaket erweitert die NC um verschiedene Funktionen, die für deren Einsatz in Stanz- bzw. Nibbelmaschinen mit extern gesteuerten Stößeln erforderlich sind.

Applizieren:

HINWEIS

Schäden an der Maschine oder am Werkstück möglich!

Während der Bearbeitung darf kein Steuerungs-Reset ausgelöst werden.

1. MP 8001 00010 (Funktion "Stanzen appliziert") auf den Wert "1" setzen.
2. Zeitreferenz (MP 8001 00021) und Hubauslösezeiten (MP 8001 00020) entsprechend Ihrer Applikation einstellen.
3. Falls gewünscht, Hubüberwachung (MP 8001 00030) parametrieren.
4. Per MP 4075 00101 und MP 4075 00102 einen schnellen Digital-Ein- und -Ausgang reservieren.
5. Um den Funktionszustand beim Hochlauf und nach Grundstellung festzulegen, empfehlen wir, in MP 7060 00010 (Hochlauf) und MP 7060 00020 (Grundstellung) als Einschaltzustand folgende Einstellungen aufzunehmen:
PUN(0) oder NIB(0) Nibbeln/Stanzen AUS
PTD Hubauslösezeiten und Zeitreferenz laut MP 8001 00020 und MP 8001 00021
TTO(0) Tangentiale Werkzeugorientierung AUS.
6. Steuerungs-Reset auslösen.

Aktivieren:

Voraussetzung:

- Stanzen/Nibbeln ist appliziert.
 - Schneller Digital-Ein- und -Ausgang ist per MP 4075 00101 bzw. MP 4075 00102 reserviert.
 - G9 mit SHAPE ist aktiv.
 - Notwendige IF-Signale werden ausgewertet bzw. generiert.
1. Zum Stanzen optional, zum Nibbeln erforderlich:
Länge der Teilsätze per "LEN=" angeben.
 2. Um Stanzen zu aktivieren, Punch(1) oder PUN(1) programmieren
- oder -
um Nibbeln zu aktivieren, Nibble(1) oder NIB(1) programmieren.

Deaktivieren: Punch(0) oder PUN(0) bzw. Nibble(0) oder NIB(0) programmieren (beide Funktionen sind gleichwertig und schalten sowohl Nibbeln als auch Stanzen ab).

	Documentation	Instruction
Instruction:		Editieren von Maschinenparametern
Documentation:	MTX Maschinenparameter	Stanzen und Nibbeln

IW-Operation / NC-Programmierung: Funktion LEN programmieren

- **LEN=<Wert>** programmieren ist für Nibbeln notwendig und für Stanzen optional. **LEN** zerlegt die programmierten Verfahrbewegungen in Teilsegmente gemäß dem vorgegebenen **<Wert>**.
 - Bei Linearsätzen: **<Wert>** = Teilstreckenlänge
 - Bei Zirkularsätzen: **<Wert>** = Bogenlänge



Programmiereinheit wie für die Achskoordinaten.

<Wert> muss kein ganzer Teiler der programmierten Wegstrecke sein. NC-intern wird automatisch ein effektiver LEN-Wert kleiner/gleich dem programmierten LEN-Wert ermittelt, so dass die berechneten Teilsegmente gleiche Weglängen aufweisen (Der **<Wert>** wird steuerungsintern entsprechend angepasst.)

	Documentation	
Documentation:	MTX Maschinenparameter	Funktion LEN

IW-Operation / NC-Programmierung: Stanzen oder Nibbeln aktivieren

- Um Stanzen zu aktivieren, **Punch(1)** oder **PUN(1)** programmieren.
- oder -
- Um Nibbeln zu aktivieren **Nibble(1)** oder **NIB(1)** programmieren.

IW-Engineering / Configuration: Parameter editieren

- Einen schnellen digitalen Eingang per Parameter **DigIOAllocIn** "Zuordnung der Digitaleingänge" (4075 00101) und einen digitalen Ausgang per Parameter **DigIOAllocOut** "Zuordnung der Digitalausgänge" (4075 00102) die Funktion "Stanzen" zuweisen.
- Um den Funktionszustand beim Hochlauf und nach Grundstellung festzulegen, wird empfohlen in den Parameter **NcResetState** "Zustand nach Hochlauf (Einschaltzustand)" (7060 00010) und den Parameter **ChResetState** "Zustand nach Kanal-Grundstellung" (7060 00020) als Einschaltzustand folgende Einstellungen aufzunehmen:
 - **PUN(0)** oder **NIB(0)** Nibbeln/Stanzen AUS
 - **PTD** Hubauslösezeiten und Zeitreferenz laut **TimeOffs** "Hubauslösezeit" (8001 00020) und **TimeRef** "Referenz der Hubauslösezeiten" (8001 00021)
 - **TTO(0)** Tangentiale Werkzeugorientierung AUS.
- Die Funktion Stanzen und Nibbeln muss über den Parameter **pun** "Stanzen und Nibbeln" im Setup (SUP) aktiviert werden.
- **EnablPun** "Aktiviere Stanzen und Nibbeln" (8001 00010)
- **StrokeMonTime** "Hub-Überwachungszeit" (8001 00030)
- **TimeRef** "Referenz der Hubauslösezeiten" (8001 00021)

- **TimeOffs** "Hubauslösezeit" (8001 00020)

	Documentation	
Documentation:	MTX Maschinenparameter	Stanzen und Nibbeln aktivieren

12.6 Tangentiale Werkzeugorientierung

12.6.1 Beschreibung

Funktion

Stellt bei aktivem "Stanzen" (Punch/PUN) oder "Nibbeln" (Nibble/NIB) für jeden Hub eine synchrone Werkzeugachse vom Achsbewegungstyp "endlos" (%MP 1003 00004) tangential zur programmierten Bahn in der aktiven Ebene eines Kanals an.

Die Werkzeugachse bewegt sich dazu synchron mit den ebenenaufspannenden Linearachsen und erreicht zeitgleich mit deren Zielposition den erforderlichen Anstellwinkel. Der Ausdruck "Zielposition" bezieht sich hier sowohl auf die Endposition eines programmierten Satzes, als auch auf die Endpositionen von Teilsätzen, die die NC im Zusammenhang mit der automatischen Satzzerlegung generieren kann.

Muss bereits am Anfang des programmierten Satzes ein Hub ausgelöst werden (kann beim Nibbeln erforderlich sein), generiert die NC automatisch einen separaten Satz, in dem die Werkzeugachse zuvor auf den erforderlichen Anstellwinkel gedreht wird.

Merkmale:

- Parametrierbarer Anstellwinkel (Winkel zwischen Symmetrieachse des Werkzeuges und der Bahntangente).
- Berücksichtigung der Werkzeugsymmetrie. Dadurch kann die Steuerung bei Konturknicken die für einen Hub erforderliche Ansteldrehung des Werkzeuges selbstständig optimieren ("Kürzeste Weg"-Logik).
- Anstellwinkel und Werkzeugsymmetrie sind im Programm parametrierbar

Restriktionen

- Funktion "Tangentiale Werkzeugführung" (TangTool bzw. TTL; siehe [Kap. 10.4 "Tangentiale Werkzeugführung" auf Seite 184](#)) darf nicht gleichzeitig aktiv sein.
- Die "Tangentiale Werkzeugorientierung" kann jederzeit programmiert und eingeschaltet werden. Anstellbewegungen der Werkzeugachse werden aber nur dann ausgeführt, wenn Nibble/NIB ("Nibbeln EIN") oder Punch/PUN ("Stanzen EIN") aktiv sind.
- TangToolOri/TTO muss in einem separaten Satz programmiert werden.
- Die Werkzeugachse muss per MP 7050 00210 definiert sein. Eine programmgesteuerte Änderung der Werkzeugachse ist nicht möglich.
- Als Werkzeugachsen sind nur synchrone Endlosachsen erlaubt.

Relevante NC-Funktionen

TangToolOri/TTO	Initialisieren und Einschalten der "Tangentialen Werkzeugorientierung". TangToolOri führt zu keiner Verfahrbewegung. Das Anstellen der Werkzeugachse startet erst mit dem nächsten Stanz- bzw. Nibbel-Verfahrssatz, sofern am Satzanfang kein Hub ausgelöst werden muss (vgl. Kap. "Funktion" auf Seite 455).
TangToolOri(0) - oder - TTO(0)	Ausschalten der "Tangentialen Werkzeugorientierung"

Tab. 12-27: Relevante NC-Funktionen

Relevante Maschinenparameter (MP)

Die Werte der Maschinenparameter wirken als Initialisierungswerte, wenn TangToolOri ohne Parameter programmiert ist. Sie können die Initialisierungswerte im TangToolOri-Satz durch die in Klammern angegebenen Parameter überschreiben. Syntax siehe "MTX Programmieranleitung".

7050 00210	Nummer der Drehachse
7050 00220	Werkzeugsymmetrie (SYM)
7050 00240	Anstellwinkel (ANG)

Tab. 12-28: Relevante Maschinenparameter

12.6.2 Aktivieren

Voraussetzung:

- Die Drehachse ist bereits als synchrone Endlosachse im System vorhanden und, bezogen auf Geschwindigkeit, Dynamik und ggf. Positionierlogik, entsprechend Ihren Anforderungen korrekt parametrier.
 - MP 7050 00210 (Nummer der Drehachse) ist entsprechend Ihren Anforderungen parametrier.
 - Die notwendigen achs- und kanalbezogenen IF-Signale werden ausgewertet bzw. generiert.
1. Falls aktiv, Funktion "Tangentiale Werkzeugführung" (TangTool) für die entsprechende Achse ausschalten (TangTool(0) oder TTL(0)).
 2. Nibble/NIB ("Nibbeln EIN") oder Punch/PUN ("Stanzen EIN") aktivieren.
 3. "Tangentiale Werkzeugorientierung" per TangToolOri-Satz aktivieren.

Im gleichen Satz die Initialisierungswerte von Werkzeugsymmetrie und Anstellwinkel per Parameter "SYM" bzw. "ANG" entsprechend Ihrer Applikation einstellen.

Syntax siehe "MTX Programmieranleitung".



Die "Tangentiale Werkzeugorientierung" kann auch bei nicht aktivem Stanzen und Nibbeln parametrier und eingeschaltet werden.

Anstellbewegungen der Werkzeugachse erfolgen jedoch nur, wenn Nibbeln oder Stanzen aktiv sind!

12.6.3 Deaktivieren

TangToolOri(0) bzw. TTO(0) programmieren.

12.7 Laserbearbeitung

12.7.1 Bahngeschwindigkeitsabhängige Laserleistungssteuerung

Funktion

Die Funktion steuert die Leistung eines Lasers abhängig vom aktuellen Istwert des Vorschubs (V_{Bahn}). Dazu wird, gemäß der Konfiguration im Maschinenparameter AnalOAllocOut[1..4], an einem Analogausgang ein entsprechender Spannungswert ausgegeben.

Unterhalb einer bestimmten Bahngeschwindigkeit V_{Bahn} wird die Laserleistung auf ein einstellbares Minimum beschränkt. Oberhalb einer bestimmten Bahngeschwindigkeit V_{Bahn} wird die Laserleistung auf einem einstellbaren Maximum gehalten.

Zwischen V_{Min} und V_{Max} wird die Laserleistung bahngeschwindigkeitsabhängig angepasst, siehe Abbildung:

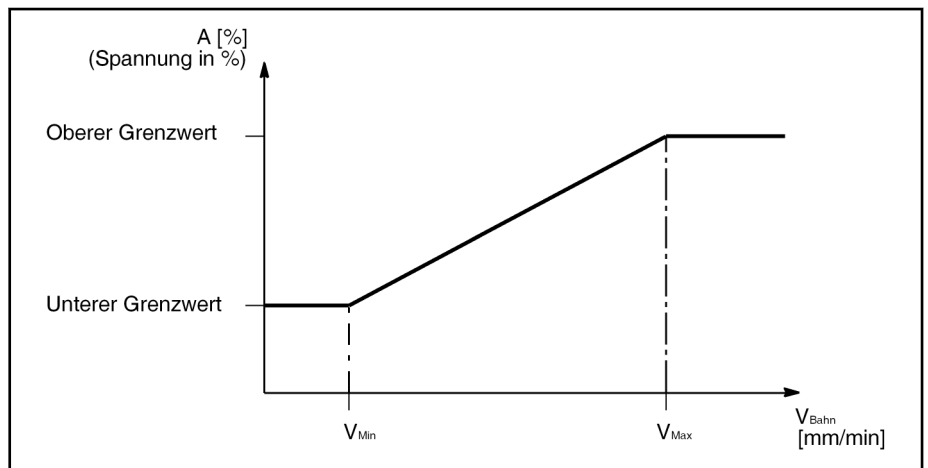


Abb. 12-11: Laserbearbeitung

Der Ausgabewert der Spannung wird in % angegeben.

100% Ausgabewert der Spannung entspricht 10 Volt am Analogausgang.

Die effektive Bahngeschwindigkeit V_{Bahn} wird aus den Geschwindigkeiten der ausgewählten Koordinaten gebildet:

- durch Anwählen der aktiven Ebene (APL) oder des aktiven Raumes (ASP). Hierbei wird auch eine ggf. aktive Achstransformation oder Koordinatentransformation (Schiefe Ebene) berücksichtigt.
- durch direkte Anwahl von Werkstückkoordinaten im Teileprogramm.

Restriktionen

- Die zur Verfügung stehenden Analogausgänge begrenzen die Anzahl der Kanäle, die die Funktion verwenden können.
- Beim Einsatz der Funktion "Schiefe Ebene" kann für die Koordinatenauswahl nur "aktiver Raum" mit PL(ASP) zur Bildung der Geschwindigkeit V_{Bahn} verwendet werden.
- Beim Auftreten eines Fehlers (Laufzeitfehler, Zustandsklasse1-Fehler), beim Weggehen von "Antrieb in Betrieb" eines an der Bahn beteiligten Antriebs (Keine FG, Antrieb AUS) und bei "Vorschub Halt" wird keine Laserspannung ausgegeben.

Relevante NC-Funktionen

LFP	Startet die bahngeschwindigkeitsabhängige Laserleistungssteuerung. Alleine programmiert wirken die per Maschinenparameter definierten Konfigurationsdaten. Optional lassen sich zusätzliche Parameter innerhalb der Parameterliste der Funktion programmieren.	
LFCnf/LFC	Parametriert die aktive Laserleistungssteuerung per Teileprogramm. Parameter für LFP und LFCnf:	
	LL(<%Spannung>], [<VMin>])	
		Untere Leistungsgrenze: Der hier festgelegte Spannungswert wird unterhalb der angegebenen Bahngeschwindigkeit ausgegeben.
	UL(<%Spannung>], [<VMax>])	
		Obere Leistungsgrenze: Der hier festgelegte Spannungswert wird oberhalb der angegebenen Bahngeschwindigkeit ausgegeben.
	mit	
	<%Spannung>	0% .. 100%: entspricht 0 .. 10 Volt
	<VMin>	Unterer Eckwert der Bahngeschwindigkeit in mm/min oder inch/min.
	<VMax>	Oberer Eckwert der Bahngeschwindigkeit in mm/min oder inch/min.
	PL(<Ebenenbezeichnung>)	Koordinatenauswahl für die Geschwindigkeitsberechnung per Ebenenauswahl.
	mit	
	<Ebenenbezeichnung>	"APL": aktuelle Ebene (G17, G18, G20) "ASP": aktueller Raum "MCD": Maschinenparameter-Werte
	CD(<Koordinate1>, [<Koordinate2>], ... , [<Koordinate n>])	
		Koordinatenauswahl für die Geschwindigkeitsberechnung direkt per logischem Namen.
	mit	
	<Koordinatex>	x = 1..n logische Namen der beteiligten Raumkoordinaten oder Pseudo-koordinaten (Achsen)
LFP(0)	Beendet die bahngeschwindigkeitsabhängigen Laserleistungssteuerung	

Tab. 12-29: Relevante NC-Funktionen

Relevante Sercos Parameter

S-0-0040	Geschwindigkeits-Istwert Ist S-0-0040 nicht im Antriebstelegramm enthalten, wird die Geschwindigkeit aus der Veränderung der Achsistposition (S-0-0051 oder S-0-0053) gewonnen.

Tab. 12-30: Relevante Sercos Parameter

Relevante IF-Signale

am Kanalinterface	
qCh_FeedStop	Vorschub Sperre
qCh_FeedHold	Vorschub Halt
am Achsinterface	
qAx_DrvOn	Antrieb Ein
qAx_DrvLock	Vorschub Sperre

Tab. 12-31: Relevante IF-Signale am Kanal-/Achsinterface

Relevante Maschinenparameter

PLC/AnaIO/FuncAllocOut/AnaIOAllocOut[1..4] (4075 00104)	
	Zuordnung der Analogausgänge 701: Kanal 1 .. 712: Kanal 60
VDO/Ch[k]/Out/AnaOutUpVelLim (7050 00810)	
	oberer Leistungsgrenzwert in %
VDO/Ch[k]/Out/AnaOutLowVelLim (7050 00810)	
	unterer Leistungsgrenzwert in %
VDO/Ch[k]/Vel/UpVelLim (7050 00815)	
	oberer Geschwindigkeitsgrenzwert in mm/min
VDO/Ch[k]/Vel/LowVelLim (7050 00815)	
	unterer Geschwindigkeitsgrenzwert in mm/min
VDO/Ch[k]/CoordSel (7050 00820)	
	Koordinatenauswahl: • keine Koordinaten • aktive Arbeitsebene • aktiver Arbeitsraum

Tab. 12-32: Relevante Maschinenparameter

12.7.2 Handlungsanweisung: Bahngeschwindigkeitsabhängige Laserleistungssteuerung

Applizieren: **Voraussetzung:**

Analogausgänge (z.B. per PROFIBUS) müssen vorhanden sein.

Die Funktion steuert die Leistung eines Lasers abhängig vom aktuellen Istwert des Vorschubs (V_{Bahn}). Dazu wird, gemäß der Konfiguration in Parameter **AnalOAllocOut** "Zuordnung der Analogausgänge" (4075 00104), an einem Analogausgang ein entsprechender Spannungswert ausgegeben.

IW-Engineering / Configuration: Parameter editieren

- Dem Parameter **AnalOAllocOut** "Zuordnung der Analogausgänge" (4075 00104) einen Wert von 0701...0712 (für Kanal 1..60) zuweisen.
- Die Funktion Geschwindigkeitsproportionaler Ausgang muss über die optionale Parametergruppe **VDO** "Geschwindigkeitsproportionaler Ausgang" im Setup (SUP) aktiviert werden, um folgende Parameter für die Benutzung der Analogausgänge zu konfigurieren:
 - **AnaOutUpVelLim** "Sollwert bei Erreichen des oberen Geschwindigkeitsgrenzwerts" (7050 00810)
 - **AnaOutLowVelLim** "Sollwert bei Erreichen des unteren Geschwindigkeitsgrenzwerts" (7050 00810)
 - **UpVelLim** "Oberer Geschwindigkeitsgrenzwert" (7050 00815)
 - **LowVelLim** "Unterer Geschwindigkeitsgrenzwert" (7050 00815)
 - **CoordSel** "Koordinatenauswahl für die Bahngeschwindigkeitsberechnung" (7050 00820)



Die Antriebstelegramme der beteiligten Achsen sollten den Sercos Parameter **S-0-0040** "Geschwindigkeitsistwert" enthalten.

Ist der Sercos Parameter **S-0-0040** "Geschwindigkeitsistwert" nicht im Antriebstelegramm enthalten, wird die Geschwindigkeit aus der Veränderung der Achsistposition (**S-0-0051/S-0-0053** "Lageistwert Geber 1/2") gewonnen.

Example	Documentation	Instruction
Instruction:		Editieren von Maschinenparametern
Documentation:	MTX Maschinenparameter	Laserleistungssteuerung

Aktivieren: Nach Steuerungshochlauf ist die Laserleistungssteuerung ausgeschaltet und die Einstellungen aus den relevanten Maschinenparametern sind angewählt.

Vorgaben aus Maschinenparameter NCResetState "Einschaltzustand nach Hochlauf" und Maschineparameter ChResetState "Einschaltzustand nach Grundstellung" überschreiben diese Voreinstellung.

IW-Operation / NC-Programmierung: Laserleistungssteuerung aktivieren

Die Laserleistungssteuerung kann mit den in den Maschinenparametern hinterlegten Voreinstellungen aktiviert werden und, falls notwendig, im Teileprogramm mit Parametern zusätzlich beeinflusst werden:

1. Einschalten der Laserleistungssteuerung mit **LFP{(1)}** (mit den Default-Parametern).
- oder -

2. Einschalten der Laserleistungssteuerung mit zusätzlichen Parametern LFP(LL(..),UL(..),PL(..)).

Die aktive Laserleistungssteuerung kann im Teileprogramm mit LFC(LL(..),UL(..),PL(..)) eingestellt werden.

Aktivieren/Deaktivieren der bahngeschwindigkeitsabhängigen Laserleistungssteuerung mit Voreinstellung. Während des Betriebes können die Parameter im Teileprogramm beeinflusst werden. Die Funktion steuert die Leistung eines Lasers per analogem Spannungssignal (0...10 V) in Abhängigkeit des aktuellen Bahnvorschubes v_{Bahn} .

Beispiel:

Einstellung der Maschinenparameter:

AnalOAllocOut[3]	701	Kanal 1 verwendet den 3. Analogausgang
AnaOutLowVelLim	20	unterer Leistungsgrenzwert 20% (= 2 V)
AnaOutUpVelLim	90	oberer Leistungsgrenzwert 90% (= 9 V)
LowVelLim	500	unterer Geschw.-grenzwert 500 mm/min
UpVelLim	2000	oberer Geschw.-grenzwert 2000 mm/min
CoordSel	2	Die Bahn-Istgeschwindigkeit wird aus der aktiven Ebene gebildet.

Tab. 12-33: Einstellung der Maschinenparameter Beispiel

Teileprogramm 1:

N010 G17 G8 G62 F2500

N020 G0 X0 Y0 Z0

N110 G1 X10 Laser einschalten
LFP(1)

N120 Y10

N130 X0

N140 Y0 LFP(0) Laser aus

Achtung:

Die Achsen haben ihren Endpunkt nicht erreicht.

N200 G54

N210 X10 LFP(1) Laser einschalten

N220 Y10

N230 X0

N240 Y0

N250 G4 F0.5 Laser aus nach Ablauf der Verweilzeit
LFP(0)

N300 G55

N310 X10 LFP(1) Laser einschalten

N320 Y10

N330 X0

N340 Y0 G1(IPS) Laser aus nach Erreichen des Inpositionsfensters
LFP(0)

M30

Erklärung:

Die Ausgangsspannung bewegt sich zwischen 2 und 9 V.

Wenn die Bahngeschwindigkeit in der aktiven Ebene 2000 mm/min überschreitet, werden 9 V ausgegeben.

Wenn die Bahngeschwindigkeit 500 mm/min unterschreitet, werden 2 V ausgegeben, z.B. beim Anfahren an den Ecken (abhängig vom Achssprungvermögen) am Ende der Bewegung.

Ausschalten des Lasers:

N140 Beim Ausschalten des Lasers sind die Achsen wegen des Nachlaufs noch nicht am Endpunkt angelangt. Dadurch ist das bearbeitete Teil fehlerhaft.

N250 Das Ausschalten des Lasers wird um eine programmierte Zeit verzögert.

N340 Der Laser wird nach Erreichen des Inpositionsfensters aller Achsen ausgeschaltet.

Teileprogramm 2:

N010 G17 G8 G62 F2500

N020 G0 X0 Y0 Z0

N030 LFPConf(UL(, oberer Geschwindigkeitsgrenzwert 1800 mm/min
1800)

N040 Koordinatenauswahl X und Y
LFPConf(CD(X,Y)

N050 G1 X10 Laser einschalten
LFP(1)

N060 Y10

N070 X0

N080 Y0 G1(IPS) Laser aus, nach Erreichen des Inpositionsfensters.
LFP(0)

N090 G54

N100 LFP(PL(APL) Aktive Ebene auswählen und Laser ein.

N110 G1 X10

N120 Y10

N130 X0

N140 Y0 G1(IPS) Laser aus, nach Erreichen des Inpositionsfensters.
LFP(0)

M30 Alle Maschinenparameter-Daten sind wieder wirksam.

Erklärung:

Die Ausgangsspannung bewegt sich zwischen 2 und 9 V.

N030 oberer Geschwindigkeitsgrenzwert 1800 mm/min

N040 Überschreitet die Bahngeschwindigkeit in den gewählten Koordinaten (X und Y) mit der aktiven Ebene (N100) 1800 mm/min, werden 9 V ausgegeben.

Bei Unterschreitung von 500 mm/min werden 2 V ausgegeben.

Example	Documentation	Example
Example:		Beispiel zur Laserbearbeitung
Documentation:	MTX Programmierhandbuch	Laserleistungssteuerung

Deaktivieren: IW-Operation / NC-Programmierung: Laserleistungssteuerung deaktivieren

1. **LFP(0)** schaltet die Leistungssteuerung am Ende eines NC-Satzes ab.
2. Nach Grundstellung und M2/M30 ist die Leistungssteuerung ausgeschaltet und die Einstellungen aus den relevanten Maschinenparametern werden wieder wirksam.

	Documentation	
Documentation:	MTX Programmierhandbuch	Laserleistungssteuerung

13 Achskonfiguration im Kanal

13.1 Achsübergabe

13.1.1 Beschreibung

Funktion

Innerhalb eines Kanals werden Achsen, die in einem bearbeitungstechnischen Zusammenhang stehen, als **Achsgruppe** bezeichnet. Sie bilden einen Interpolationsverbund. Für bestimmte Bearbeitungsaufgaben ist es notwendig die **Achszuordnung** zu ändern, d.h. Achsen aus einer Achsgruppe zu entfernen und zeitweise in die Achsgruppe eines anderen Kanals zu integrieren. Damit verbunden besteht die Möglichkeit, die Achsbezeichnungen innerhalb einer Achsgruppe zu verändern.

*Folgende **Achsübergabe-Funktionen** stehen bereit:*

- Wechseln von Achsen zwischen Achsgruppen, d.h. synchrone Achse bleibt synchrone Achse.
- Achsen aus einer Achsgruppe entfernen, d.h. synchrone Achse wird zur asynchronen Achse.
- Achsen in eine Achsgruppe übernehmen, d.h. asynchrone Achse wird zur synchronen Achse.
- Umbenennen der Achsnamen innerhalb einer Achsgruppe.
- Umschalten der Achsklassifikation (arbeitstechnische Bedeutung).

Notwendige Begriffe:

Achse übernehmen: Achse wird in die Achsgruppe **integriert**.

Achse leihen: Achse **wechselt**, d.h. Kanal leiht sich Achse von einem inaktiven Kanal.

Achse entfernen: Achse wird von ihrer bisherigen Achsgruppe **freigegeben**.



Passen Sie den Maschinenparameter Einschaltzustand nach Grundstellung (/CHAN/Ch[...]/Ini/ChReset) für diejenigen Kanäle, welche Achsen abgeben oder aufnehmen derart an, dass mit der Grundstellung wieder eine definierte Achskonfiguration hergestellt wird. Das Eintragen des Befehls DefAxis bzw. DAX für die Systemgrundstellung sorgt dabei dafür, dass die Achsen koordiniert aus den entsprechenden Kanälen abgegeben und anschließend von ihren Defaultkanälen wieder aufgenommen werden.

Die Bedeutung der Begriffe Antriebe, Kanal, Achsen, Achsgruppe, Hilfsachsen, synchrone und asynchrone Achsen, System- und Kanalachsen, System- und Kanalachsindex, Spindeln siehe [Kap. 3 "Antriebe \(Achsen, Spindeln\)" auf Seite 13](#).

Wechseln von Achsen zwischen Achsgruppen:

Beim Wechseln von Achsen zwischen zwei Kanälen sind zwei Fälle zu unterscheiden:

- **Der Kanal der "Quell-Achsgruppe" ist nicht aktiv.**

Ist der Kanal, aus der die Achse kommt, nicht aktiv, kann eine Achse jederzeit in eine zweite Achsgruppe integriert werden, d.h. die Achse wird

Achskonfiguration im Kanal

an den zweiten Kanal "verliehen". Wird sie dort wieder freigegeben, dann wird sie automatisch wieder in die Quell-Achsgruppe integriert.

Wird in der Quell-Achsgruppe ein Programm oder Handeingabe ausgewählt, während eine Achse verliehen ist, kommt es zu einer Fehlermeldung.

Mit dieser Form der Achsübergabe kann z.B. nach einer Systemgrundstellung ein Anwender-Rücksetzprogramm in einem Kanal gestartet werden, das alle Systemachsen auch aus anderen Kanälen ausleiht und sie in eine Grundposition fährt.

- **Der Kanal der "Quell-Achsgruppe" ist aktiv.**

Beim Wechseln einer Achse aus einem aktiven Kanal (Achsgruppe 1) wird zuerst die Achse entfernt und in einem zweiten Schritt wird sie in einen anderen Kanal (Achsgruppe 2) integriert.

Um sicher zu stellen, dass die Übernahme der Achse ohne Wartezeiten in der Satzvorbereitung durchgeführt werden kann, muss die Achse aus der Achsgruppe 1 bereits entfernt sein. Dies ist der Fall, wenn zur Zeit der Übergabe der Achse in die Achsgruppe 2 der entsprechende NC-Satz in der Achsgruppe 1 bereits aktiv war.

Beispiel:

Die X-Achse von Kanal 1 wird an Kanal 2 übergeben (siehe folgende Abbildung).

In Kanal 1 ist die Vorbereitung bereits auf Satz N1310 und der Satz N1220 ist aktiv. Damit ist die Freigabe der X Achse abgeschlossen.

Zu diesem Zeitpunkt ist im 2. Kanal der Satz N2110 aktiv. Die Vorbereitung befindet sich auf Satz N2220 und will die Systemachse XP (ehemals X aus Kanal 1) übernehmen.

Die Achse XP bekommt dabei den Namen ZA. Da dieser Name im Kanal 2 bereits bekannt ist, entstehen keine Wartezeiten bei der Übernahme.

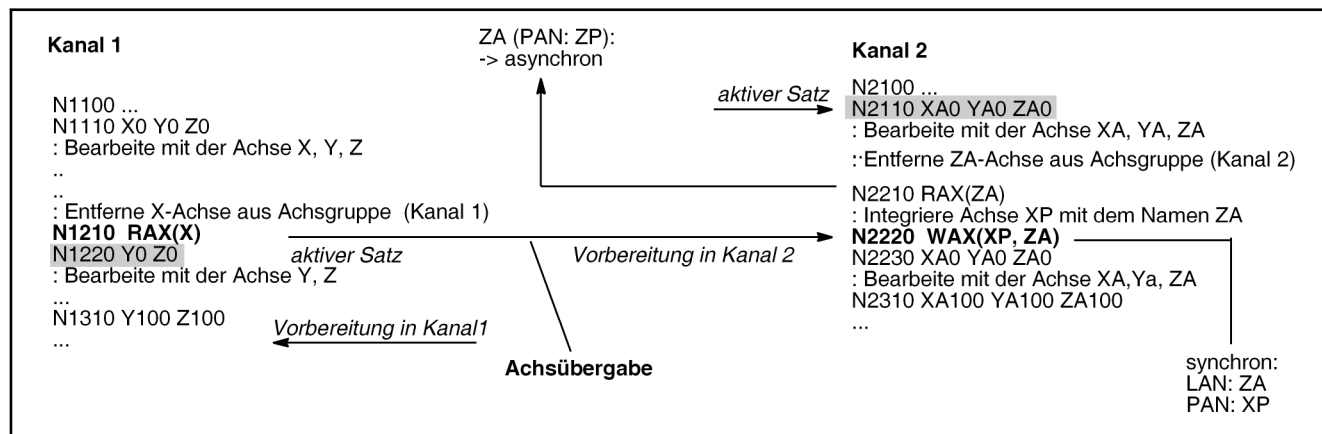


Abb. 13-1: Beispiel - Wechsel von Achsen zwischen Achsgruppen

Achsen aus einer Achsgruppe entfernen:

Mit "RemAxis" (RAX) wird eine Achse aus einer Achsgruppe entfernt. Damit wird aus einer synchronen Kanalachse eine **asynchrone** Achse. Die Satzvorbereitung wird bei dieser Aktion nicht angehalten.

Beispiel:

Die Kanalachse Y wird freigegeben, sie ist danach als asynchrone Achse YP in allen Kanälen programmierbar. Die Kanalachsen Z und B erhalten neue Kanalachsennummern. In einer Kanalachsanzeige verschwindet die Y Achse. Programmierung der Kanalachse Y führt zu einer Fehlermeldung.

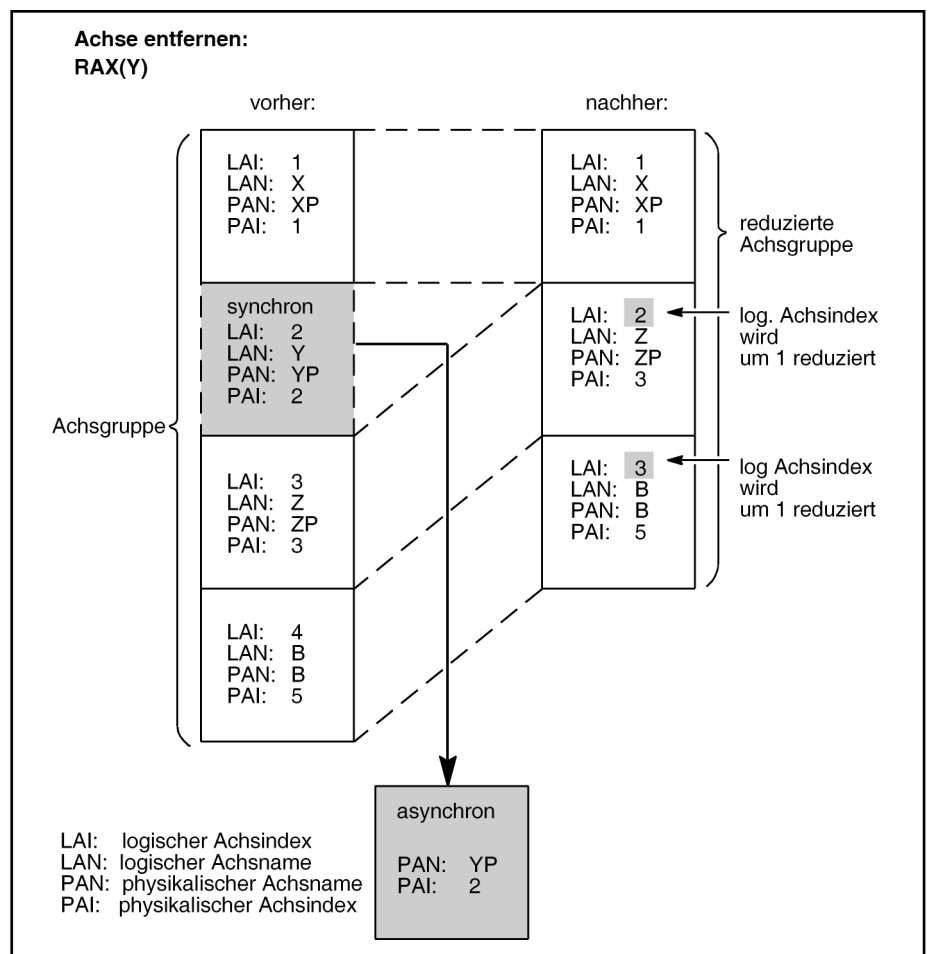


Abb. 13-2: Beispiel - Achse entfernen

Achsen in eine Achsgruppe übernehmen:

Mit "GetAxis" (GAX) oder "WaitAxis" (WAX) wird eine asynchrone Achse in eine Achsgruppe übernommen. Sie wird hierdurch zu einer synchronen Kanalachse.

- Bei "GetAxis" (GAX) muss die Achse stehen, sonst kommt es zu einer Fehlermeldung und die Satzvorbereitung wird angehalten.
- Bei "WaitAxis" (WAX) wird implizit auf den Stillstand der Achse gewartet.
- Wahlweise kann ein neuer Kanalachsname angegeben werden, mit dem die Achse im Kanal angesprochen wird. Dieser Kanalachsname muss in einem der beiden kanalspezifischen Maschinenparametern 701000010 "Namen der Kanalachsen" oder 701000020 "Optionale Achsbezeichnung" vordefiniert sein.

Beispiel:

In der folgenden Abbildung wird die Systemachse YP in eine Achsgruppe integriert. Sie kann danach als asynchrone Achse YP nicht mehr in anderen Kanälen angesprochen werden. Im eigenen Kanal kann sie als synchrone Achse wahlweise mit dem Kanal- oder dem Systemachsamen programmiert werden. In einer Kanalachsanzeige erscheint zusätzlich die Y Achse.

Die Kanalachsen Z und B erhalten neue Kanalachsindizes.

Achskonfiguration im Kanal

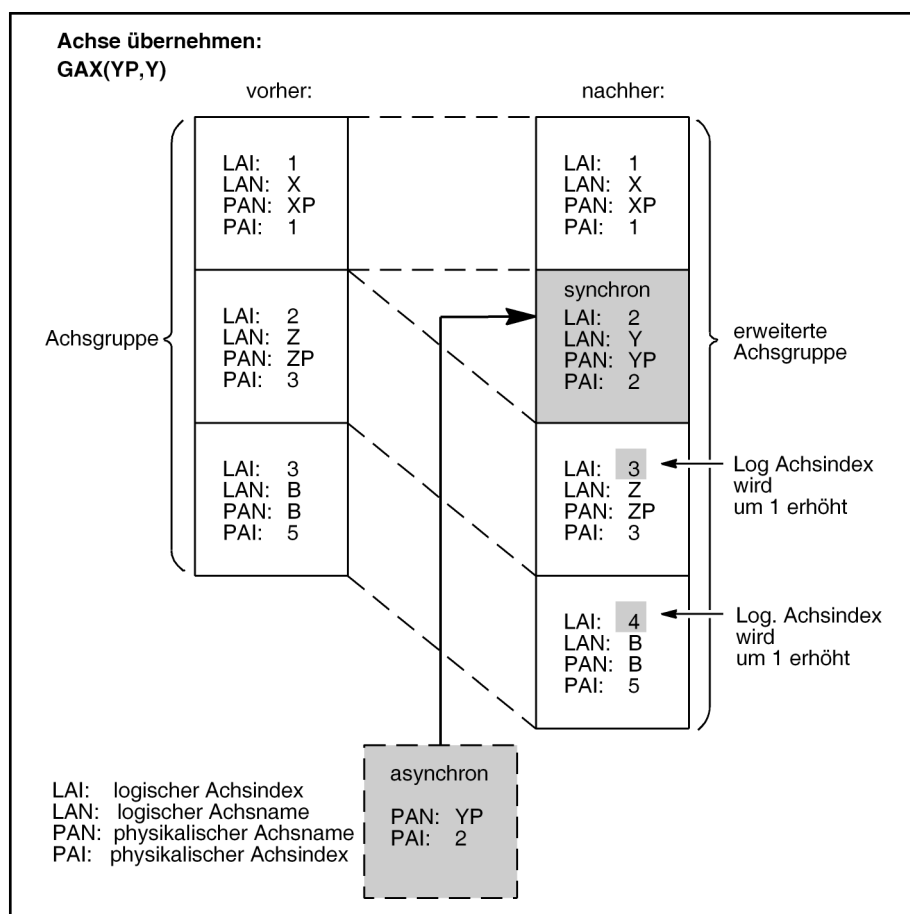


Abb. 13-3: Beispiel - Achsen in eine Achsgruppe übernehmen

Eine übernommene Achse hat keine arbeitstechnische Bedeutung, wenn sie nicht zur Default-Konfiguration des Kanals gehört. Ansonsten erhält sie ihre Default-Klassifikation aus den Maschinenparametern, sofern diese Bedeutung nicht bereits im Kanal vergeben ist.

Umbenennen der Achsnamen innerhalb einer Achsgruppe:

- Beim Übernehmen einer Achse in eine Achsgruppe per GAX bzw. WAX kann wahlweise ein neuer Kanalachsname angegeben werden.
- Mit "AssLogName" (ALN) kann direkt der Kanalachsname geändert werden.



Der logische Name muss in MP 7010 00010 "Namen der Kanalachsen" oder MP 7010 00020 "Optionale Achsbezeichnung" vordefiniert sein.

Vergabe einer arbeitstechnischen Bedeutung:

Per MP 7010 00030 kann man den Kanalachsen eines Kanals eine **arbeitstechnische Bedeutung** (Achsbedeutung) zuweisen:

- Haupt-, Neben-, Normalenachse des Werkstückkoordinatensystems
- Zuordnung der Interpolationsparameter (I, J, K)

Per Achsübergabe kann die Achsbedeutung beeinflusst werden:

- Wird eine Achse aus einer Achsgruppe **entfernt**, die innerhalb der angeählten Ebene liegt, erfolgt ein implizites Umschalten auf G16 "keine Ebene". G16 kann im Teileprogramm verwendet werden.
- Wird eine Achse in eine Achsgruppe **übernommen**, erhält sie nur dann eine arbeitstechnische Bedeutung, wenn

- sie innerhalb der Maschinenparameter diesem Kanal zugeordnet ist.
- ihr innerhalb der Maschinenparameter eine Achsbedeutung zugewiesen ist und diese Achsbedeutung im Kanal aktuell an keine andere Achse vergeben ist.



Durch Programmierung von G17(...), G18(...) oder G19(...) kann die arbeitstechnische Bedeutung von Achsen verändert werden.

Beispiel: G17(...) "Achsklassifikation ändern"

N100 G17 ()	Default Achsbedeutung: X=1, Y=2, Z=3
N200 RAX (Y)	Y wird aus Achsgruppe entfernt. Es wird implizit auf G16 umgeschaltet. Zirkularinterpolation ist nicht mehr möglich.
N210 WAX (YA)	Es wird die Achse YA in die Achsgruppe aufgenommen. Sie erhält keine arbeitstechnische Bedeutung.
N220 G17 (X, YA)	YA erhält die Achsbedeutung 2 und die X/YA-Ebene wird aktiviert.
N240 G2 X... YA...	Zirkularinterpolation ist möglich.

Achsanzeige bei Achsübergabe:

Bei Achsübergabe ändert sich ggf. die Achsanzeige.

Gründe dafür sind (siehe auch Beispiele in [Kap. 13.1 "Achsübergabe" auf Seite 465](#)):

- Die Reihenfolge der aktuellen Achsnamen und ihrer Achszuordnungen zum Kanal hat sich geändert.
- Achsen wurden aus der Achsgruppe entfernt oder in die Achsgruppe aufgenommen.
- Die Namen der Kanalachsen haben sich bei der Übergabe ggf. geändert

Im MP 6005 00020 ist die Reihenfolge in der Achsanzeige festgelegt (1..16). Ohne Änderungen in der Reihenfolge kann es vorkommen, dass durch die Übergabe die Achsnamen in der Anzeige springen.

Die entsprechende Achse wird nur angezeigt, wenn es sich um eine synchrone Kanalachse oder eine asynchrone Achse handelt. Ist der Wert negativ im Parameter eingetragen, wird sie immer angezeigt.

Restriktionen

- Die Funktionen GAX, WAX, RAX und DAX ("DefAxis") müssen im NC-Satz vor einer Achse programmiert werden.
- Ebenenauswahl G17(...), G18(...), G19(...):
arbeitstechnische Bedeutung
- Achskopplung "AxCouple" (AXC):
Entfernen einer mit AXC gekoppelten Achse aus einer Achsgruppe führt zu einem Laufzeitfehler.
- Tangentiale Werkzeugführung "TangTool" (TTL):
Entfernen einer Drehachse bei aktivem TTL aus einer Achsgruppe führt zu einem Laufzeitfehler.
- Gegebenenfalls müssen NPV-Tabellen angepasst werden.

Relevante NC-Funktionen

GAX(..)	Achse übernehmen Fehlermeldung, wenn die Achse in ihrer bisherigen Achsgruppe nicht freigegeben ist.
WAX(..)	Achse übernehmen mit Warten, bis Achse freigegeben ist.
RAX(..)	Achse aus einer Achsgruppe entfernen.
DAX	Default Achseinstellung aus Maschinenparameter übernehmen.
ALN(..)	Neuen Kanalachsamen vergeben. Dieser muss in MP 7010 00010 "Namen der Kanalachsen" oder MP 7010 00020 "Optionale Achsbezeichnung" vordefiniert sein.
RLN(..)	Kanalachsamen in der rufenden Achsgruppe wieder entfernen Die Ebenenauswahl ausschalten. Zirkularinterpolation ist nicht mehr möglich. Haupt- oder Nebenachsen können aus der Achsgruppe entfernt werden.
G16	
G17(..), G18(..), G19(..)	Arbeitstechnische Bedeutung ändern und eine Ebene anwählen.

Tab. 13-1: Relevante NC-Funktionen

Relevante CPL-Funktionen

SD(20..)	Liefert die Anzahl synchroner Achsen des rufenden Kanals.
SD(21..)	Liefert die Anzahl synchroner Achsen eines Kanals.
SD(22..)	Liefert die Kanalachsnummer einer Systemachse, falls die Systemachse eine Achse des rufenden Kanals ist.
SD(23..)	Liefert die Systemachsnummer einer Kanalachse des rufenden Kanals.
SD(24..)	Liefert die Systemachsnummer einer Kanalachse.
SD(25..)	Liefert den Kanal einer Systemachse zum aktiven Zeitpunkt.
MCODS(43, ...)	Zuordnung Achse - Kanal (aktive synchrone Achsen eines Kanals ermitteln)
FXC(..):	Direkter Zugriff auf Achsnulldpunktverschiebungswerte
SCS-, APOS- und SPOS	Verschiedene Befehle, bei denen der Systemachsindex (1 .. 16) oder der Systemachsamen programmiert wird.
PCS; AXO;	
WCS; MCS; ACS;	
PPOS; PROBE;	
COF; DPC;	
SCL	Verschiedene Befehle, bei denen der Kanal- oder der Systemachsindex programmiert wird.

Tab. 13-2: Relevante CPL-Funktionen

Relevante IF-Signale

iAx_ChIndex_00 ... iAx_ChIndex_03	In diesen Achsausgangssignalen wird die Kanalnummer des zugehörigen Kanals codiert. 0 = Achse ist keinem Kanal zugeordnet, z.B. wenn sie als asynchrone Achse betrieben wird.

Tab. 13-3: Relevante IF-Signale

Relevante Maschinenparameter (MP)

1003 00001	Namen der Systemachsen. Legt die Achsnamen für maximal 99 Systemachsen (tatsächliche Achsen an der Maschine) fest.
1003 00002	Default Achszuordnung zu den Kanälen.
7010 00010	Namen für Kanalachsen. Als logische Achsen werden die Koordinatenachsen des Maschinen- bzw. Werkstück- oder Programmkoordinatensystems bezeichnet. Sie sind auf 8 Achsen pro Kanal beschränkt.
7010 00020	Optionale Achsbezeichnungen. Werden bei der Achsübernahme oder beim Umbenennen von Achsnamen nicht bekannte Kanalachsbezeichnungen verwendet, müssen sie hier voreingestellt werden.
6005 00022	Anzeige-Priorität für Achsen und Raumkoordinaten des Kanals, die an keiner Achstransformation beteiligt sind, oder wenn keine Achstransformation aktiv ist. 6005 00022 hat gegenüber 6005 00020 Vorrang.
6005 00020	Anzeige-Priorität für asynchrone und kanalfremde Achsen über Vorzeichen: +: Achse wird nur angezeigt, wenn sie dem Kanal angehört -: Achse wird auch dann angezeigt, wenn sie einem fremden Kanal zugewiesen ist.
6005 00021	Anzeige-Priorität für Raumkoordinaten fremder Kanäle, oder für Raumkoordinaten des eigenen Kanals, wenn sie nicht an vorderster Stelle angezeigt werden sollen.

Tab. 13-4: Relevante Maschinenparameter



Innerhalb eines Kanals ist die Reihenfolge der Anzeige aller Raumkoordinaten untereinander fest. Nur die gesamte Einheit kann in der Anzeige verschoben werden.

13.1.2 Handlungsanweisung: Achsübergabe

Applizieren: Für bestimmte Bearbeitungsaufgaben ist es notwendig die Achszuordnung zu ändern, d. h., Achsen aus einer Achsgruppe zu entfernen und zeitweise in die Achsgruppe eines anderen Kanals zu integrieren.

IW-Engineering / Configuration: Parameter editieren

- Systemachsamen vergeben

SysDrName

"Physikalische Antriebsbezeichnung" (1003 00001)

Achskonfiguration im Kanal

- Kanalachsennamen vergeben
ChAxName
"Achsname" (7010 00010)
- Zusätzlich benötigte (optionale) Kanalachsbezeichnungen einstellen
OpChAxName
"Optionale Achsbezeichnung" (7010 00020)
- Zuordnung der Systemachsen zu den Kanälen festlegen
DefaultCh
"Kanalzuordnung" (1003 00002)
- Anzeige-Reihenfolge für Achsen und Raumkoordinaten festlegen
ModeAddAx
"Achse im Zusatzfenster anzeigen" (6005 00020)
PrioAddAx
"Reihenfolge der Achse im Zusatzfenster" (6005 00020)
ModeChPack
"Kanalkoordinatenpaket im Zusatzfenster anzeigen" (6005 00021)
PrioChPack
"Reihenfolge des Kanalkoord.-pakets im Zusatzfenster" (6005 00021)
ModeAxChWin
"Achse im Kanalfenster anzeigen" (6005 00022)
PrioAxChWin
"Reihenfolge der Achse im Kanalfenster" (6005 00021)
- Zum Wiederherstellen der Default-Achskonfiguration bei Grundstellung ist folgender Parameter zu konfigurieren:
In den Parameter **ChResetState** "Zustand nach Kanalgrundstellung" (7060 00020) folgendes Schlüsselwort eintragen:
"#Reset:DAX", wenn die Funktion DAX "Achseinstellungen aus MP übernehmen" bei Kanalgrundstellung
oder
"#SysRes:DAX", wenn die Funktion DAX "Achseinstellungen aus MP übernehmen" bei Systemgrundstellung
ausgeführt werden soll.



Die Funktion DefAxis (DAX) hinter dem Schlüsselwort **"#SysRes:DAX"** aktiviert kanalübergreifend die Defaultachskonfiguration laut dem Parameter DefaultCh "Kanalzuordnung" (1003 00002).

	Documentation	Instruction
Instruction:		Editieren von Maschinenparametern
Documentation:	MTX Maschinenparameter	Achsübergabe

IW-Engineering / IndraLogic: Interface Signale verknüpfen

Achsinterface-Signale **iAx_ChIndex_00...iAx_ChIndex_03** in der SPS verknüpfen, um dieser die Information über die Achszuordnung (synchron/asynchron/Kanalzuordnung) bereitzustellen.

	Documentation	Instruction
Instruction:		SPS-Signale verknüpfen
Documentation:	MTX SPS-Interface	Achszuordnung

IW-Operation / Programm: Nullpunktverschiebung anpassen

Gegebenenfalls die zu übergebenden Achsen in die entsprechenden Nullpunktverschiebungstabellen des betreffenden Kanals eintragen.

		Instruction
Instruction:		Nullpunktverschiebungen

IW-Operation / Diagnose: NC-Neustart ausführen

Zur Übernahme der eingestellten Achskonfiguration ist ein Steuerungshochlauf notwendig.

Aktivieren: Voraussetzung:

Programmtechnisch müssen die in unterschiedlichen Kanälen arbeitenden Programme bezüglich der Achsübergabe synchronisiert werden.

Z.B. muss die zu übergebende Achse freigegeben sein, ehe sie vom anderen Kanal übernommen wird, ggf. muss das Programm eines Kanals auf einen Übergabezustand des Programms im anderen Kanal warten.

Dazu dienen, je nach Anwendungsfall, folgende Funktionen:

- **GetAxis(...), GAX(...)**Achse anfordern
- **WaitAxis(...), WAX(...)**Achse anfordern und gegebenenfalls warten
- **RemAxis(...), RAX(...)**Achse freigeben

GetAxis, GAX

Übernimmt eine asynchrone Achse in den rufenden Kanal. Aus der asynchronen wird dabei eine synchrone Achse. Die Achse ist danach im aktuellen Kanal über ihren System- oder Kanal-Achsname programmierbar.

Die aufzunehmende Achse muss vom System freigegeben sein, d.h., sie darf keinem anderen Kanal angehören und darf aktuell auch an keiner Achsbewegung teilhaben.

IW-Engineering / Programmieren:

GetAxis(<SAN> | <SAI>,{<KAN>},{<SAN> | <SAI>,{<KAN>}}...)

Kurzform **GAX(...)**

<SAN> System-Achsname. Bestimmt die Achse, die in den aktuellen Kanal übernommen werden soll.

<SAI> System-Achsindex. Wirkung wie <SAN>.

<KAN> Kanal-Achsname. Sofern programmiert, erhält die aufzunehmende Achse im aktuellen Kanal den Kanal-Achsname <KAN>; <KAN> muss im Parameter **ChAxName** "Achsnamen" (7010 00010) oder im Parameter **OptChAxName** "Optionale Achsbezeichnung" (7010 00020) definiert sein



Sofern eine zu übernehmende Achse noch nicht freigegeben ist, erfolgt im Gegensatz zur Funktion **WaitAxis(...)** eine Fehlermeldung.

Achspositionen im selben Satz müssen immer nach **GetAxis(...)** programmiert werden, und dürfen nur programmiert werden, wenn keine Achstransformation aktiv ist.

Zu übernehmende Achsen dürfen an keinem aktiven Überwachungsbereich beteiligt sein.

Es ist zu beachten, dass nach Aufnahme einer Achse das WCS u.U. neu aufgespannt werden soll. Dies hat Auswirkungen auf folgende Funktionen:

- Placements
- Aktive Ebene
- Werkzeugkorrekturen

	Documentation	
Documentation:	MTX Programmierhandbuch	Achse übernehmen (WAX)

WaitAxis, WAX

Übernimmt eine asynchrone Achse in den rufenden Kanal. Aus der asynchronen wird dabei eine synchrone Achse. Die Achse ist danach im aktuellen Kanal über ihren System- oder Kanal-Achsamen programmierbar.

Ist die aufzunehmende Achse noch nicht freigegeben, wartet das System selbstständig auf die Freigabe.

IW-Engineering / Programmieren:

WaitAxis(<SAN> | <SAI>,{<KAN>},{<SAN> | <SAI>,{<KAN>}}...)

Kurzform **WAX(...)**

<SAN> System-Achsname. Bestimmt die Achse, die in den aktuellen Kanal übernommen werden soll.

<SAI> System-Achsindex. Wirkung wie <SAN>.

<KAN> Kanal-Achsname. Sofern programmiert, erhält die aufzunehmende Achse im aktuellen Kanal den Kanal-Achsamen <KAN>; <KAN> muss im Parameter **ChAxName** "Achsenname" (7010 00010) oder im Parameter **OptChAxName** "Optionale Achsbezeichnung" (7010 00020) definiert sein



Sofern eine zu übernehmende Achse noch nicht freigegeben ist, wartet die Satzvorbereitung den Stillstand der Achse ab. Anschließend wird die Achse übernommen. Im Gegensatz zur Funktion **GetAxis(...)** erfolgt keine Fehlermeldung und kein Programmabbruch.

Achspositionen im selben Satz müssen immer nach **WaitAxis(...)** programmiert werden, und dürfen nur programmiert werden, wenn keine Achstransformation aktiv ist.

Zu übernehmende Achsen dürfen an keinem aktiven Überwachungsbereich beteiligt sein.

Es ist zu beachten, dass nach Aufnahme einer Achse das WCS u.U. neu aufgespannt werden soll. Dies hat Auswirkungen auf folgende Funktionen:

- Placements
- Aktive Ebene
- Werkzeugkorrekturen

	Documentation	
Documentation:	MTX Programmierhandbuch	Achse übernehmen (WAX)

RemAxis, RAX

Entfernt eine synchrone Achse aus dem rufenden Kanal. Aus der synchronen wird dabei eine asynchrone Achse. Die asynchrone Achse ist danach in jedem Kanal über ihren System-Achsnamen programmierbar.

IW-Engineering / Programmieren: Achse aus Achsgruppe entfernen

RemAxis(<SAN> | <SAI> | <KAN>{,<SAN> | <SAI> | <KAN>}...)

Kurzform **RAX(...)**

<SAN> System-Achsnamen. Bestimmt diejenige Achse, die aus dem aktuellen Kanal entfernt werden soll.

<SAI> System-Achsindex. Wirkung wie <SAN>.

<KAN> Kanal-Achsnamen. Wirkung wie <SAN>.



Die Satzvorbereitung wird beim Entfernen einer Achse nicht angehalten.

Ungültige Achsnamen erzeugen eine Fehlermeldung.

Ist eine zu entfernende Achse im System definiert, aber bereits nicht mehr im aktuellen Kanal vorhanden, erfolgt keine Fehlermeldung.

Achspositionen im selben Satz müssen immer nach **RemAxis(...)** programmiert werden.

Achsen, die in einer aktiven Achstransformation verknüpft sind, können nicht aus dem Kanal abgegeben werden.

Es ist zu beachten, dass nach Abgabe einer Achse, die das WCS mit aufgespannt, das aktuelle Werkstückkoordinatensystem nicht erhalten werden kann. Dies kann Auswirkungen auf folgende Funktionen haben:

- Placements
- Aktive Ebene
- Werkzeugkorrekturen

	Documentation	
Documentation:	MTX Programmierhandbuch	Achse übernehmen (WAX)

Beispiel:

1 - Achse wird temporär an einen 2. Kanal abgegeben

Zum Zeitpunkt t1 wird im 1. Kanal mit den Kanalachsen X, Y, Z und im 2. Kanal mit den Achsen U, B bearbeitet.

Im 1. Kanal wirkt die NPV-Tabelle `z01.zot` und im 2. die Tabelle `z02.zot` mit den Spalten U und B.

Im Satz N2220 wird gewartet bis die YP-Achse freigegeben wird. Dies geschieht zum aktiven Zeitpunkt von Satz N1210. Ab da wird die Bearbeitung im 2. Kanal fortgesetzt. YP soll als 2. Kanalachse angesprochen werden, es wirkt die entsprechende Spalte aus der NPV-Tabelle `z02.zot`.

Zum Zeitpunkt t2 ist die Achsübergabe abgeschlossen. In der Achsanzeige des 2. Kanals erscheint jetzt YP als 2. Kanalachse.

Im Satz N1310 wird YP-Achse wieder in den ersten Kanal integriert. Es muss programmtechnisch sichergestellt sein, dass zu diesem Zeitpunkt die YP-Achse vom 2. Kanal freigegeben ist. Ansonsten kommt es zu einem Laufzeitfehler.

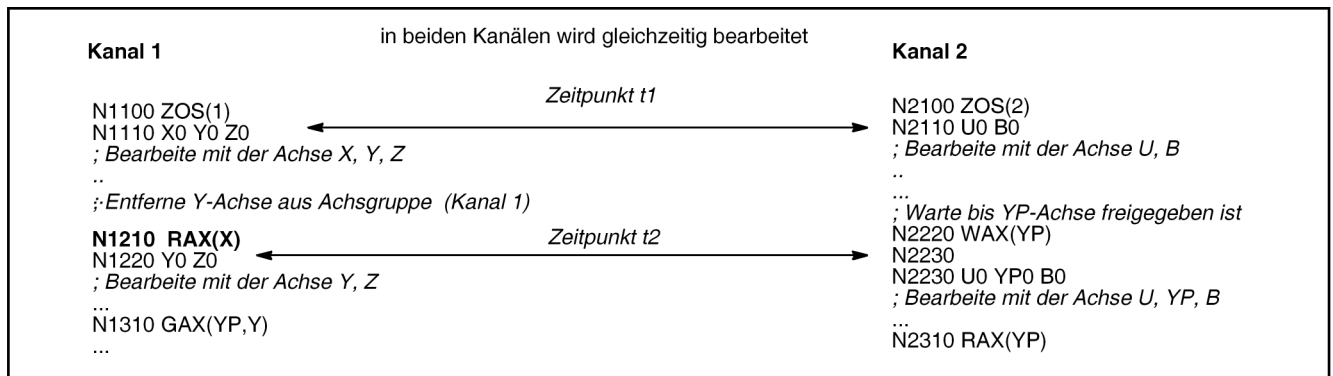


Abb. 13-4: Beispiel 1 - Achse wird temporär an einen 2. Kanal abgegeben

Achskonfiguration im Kanal

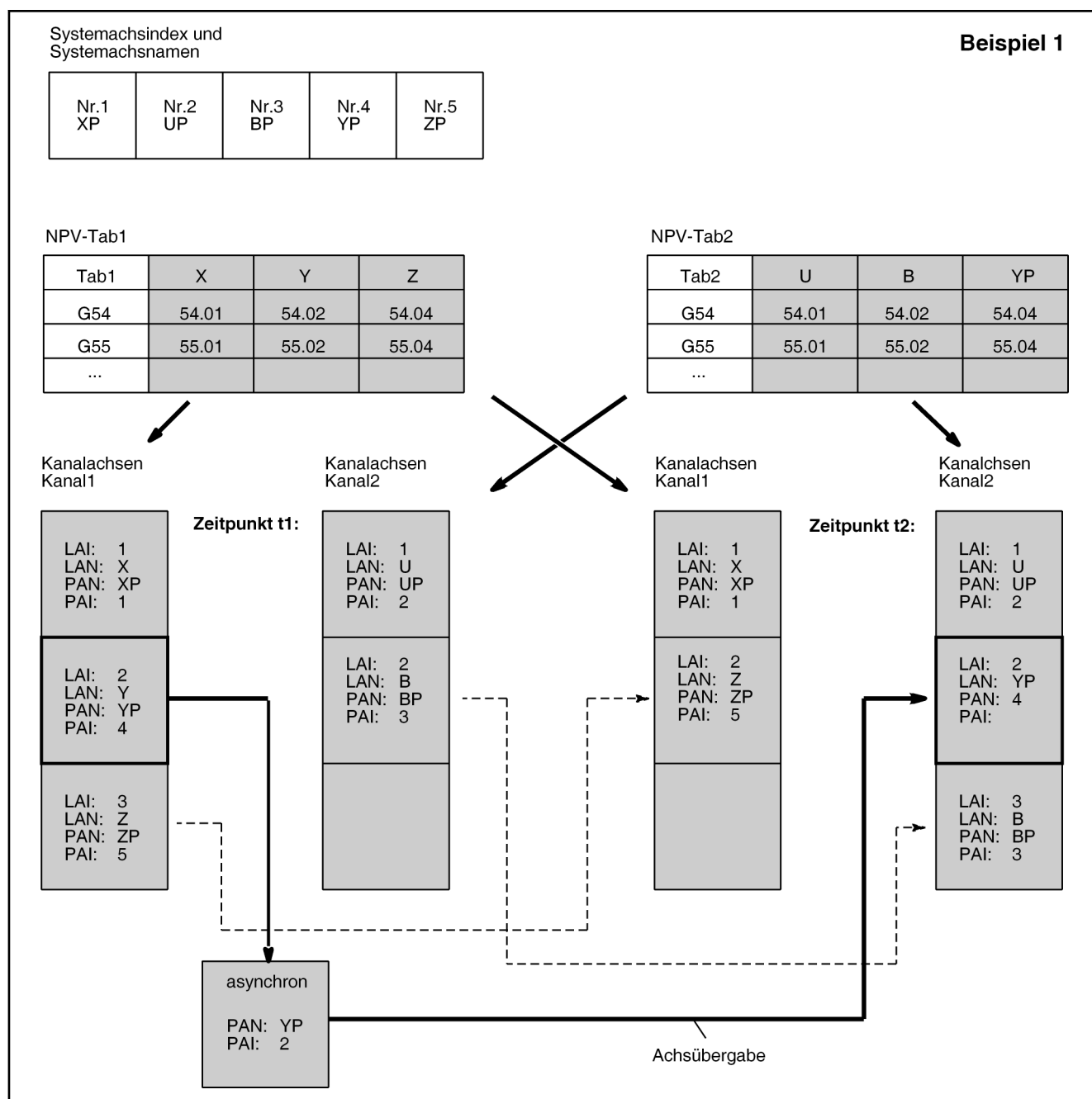


Abb. 13-5: Beispiel 1

Beispiel:**2 - Achsübergabe bei 2 realen Bearbeitungskanälen**

Zum Zeitpunkt t1 wird im 1. Kanal mit den Kanalachsen X, Y, Z und im 2. Kanal mit den Achsen U, V, W bearbeitet. Im 1. Kanal wirkt die NPV-Tabelle z01.zot und im 2. die Tabelle z02.zot.

In den Sätzen N1210 und N2210 werden die Y- und V-Achse freigegeben. Beide Achsen waren an der Ebenenauswahl beteiligt. Es wurde deshalb in beiden Kanälen auf G16 umgeschaltet.

Im Satz N1220 bekommt die ZP-Achse den neuen Kanalachsennamen Y.

In den Sätzen N1230 und N2220 wird gewartet, bis die angeforderten Achsen freigegeben sind.

In Satz N1240 wird die Achsbedeutung neu definiert und die Ebene neu gesetzt. In der Achsanzeige des 1. Kanals wird X, Y und Z angezeigt, wobei sich hinter Y die Achse ZP und hinter Z die Achse VP verbirgt.

Die Reihenfolge der Achsanzeige kann in den Maschinenparametern festgelegt werden.

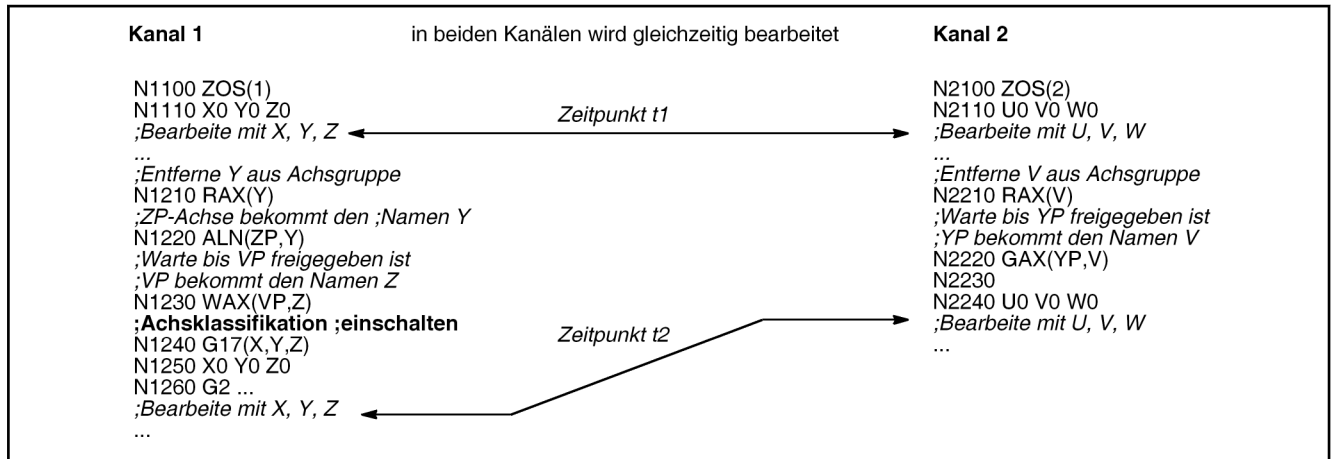


Abb. 13-6: Beispiel 2 - Achsübergabe bei 2 realen Bearbeitungskanälen

Achskonfiguration im Kanal

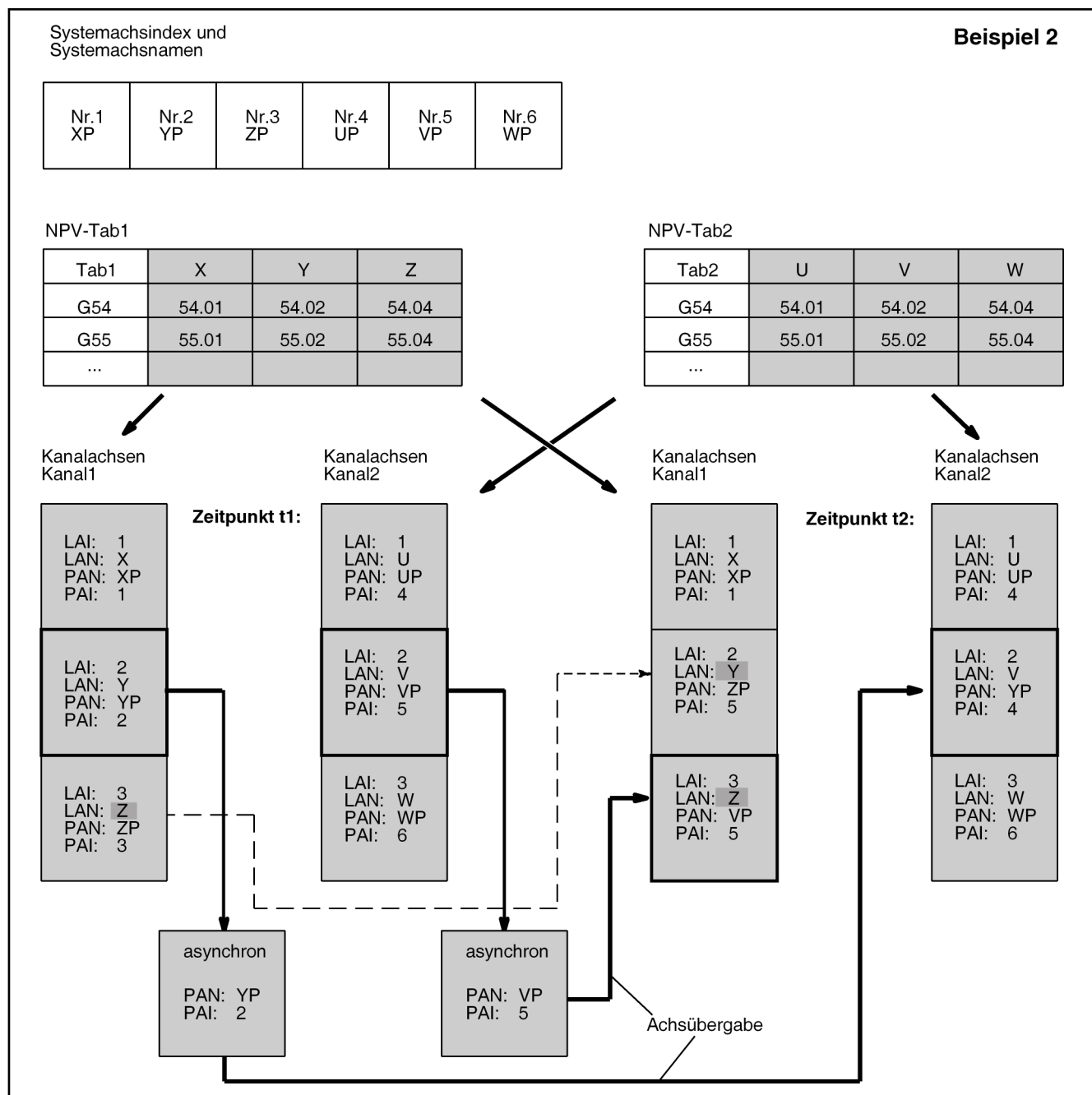


Abb. 13-7: Beispiel 2

Beispiel:**3 - Tausch zweier gleichbedeutender Achsen in einem Kanal**

Zum Zeitpunkt t1 wird mit den Kanalachsen X, Y und Z gearbeitet. Hinter der X-Achse verbirgt sich die Systemachse X1.

In Satz N1210 wird die X1-Achse freigegeben und in Satz N1220 wird die X2-Achse mit Namen X integriert.

Ab Zeitpunkt t2 kann mit der X2-Achse bearbeitet werden. Für diese Achse gelten die NPV-Werte aus der Spalte mit dem Namen X2.

In der Achsanzeige ändert sich nichts.

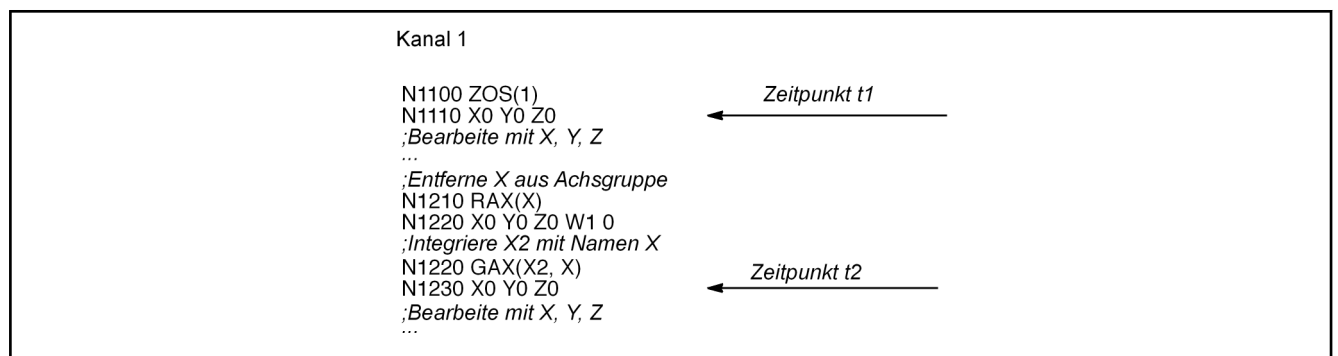


Abb. 13-8: Beispiel 3 -Tausch zweier gleichbedeutender Achsen in einem Kanal

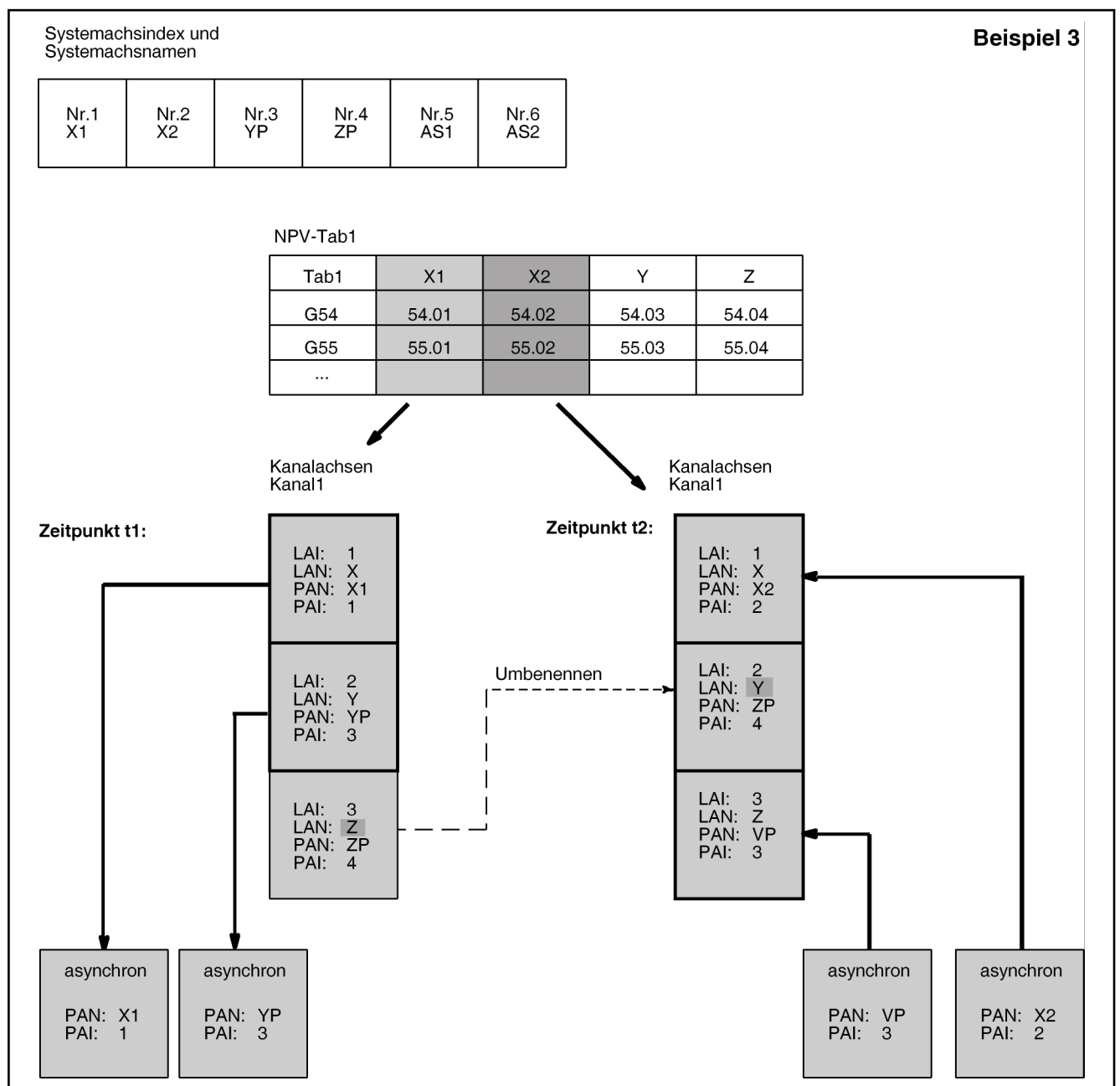


Abb. 13-9: Beispiel 3

Achskonfiguration im Kanal

*Beispiel:***4 - Mehr als 8 synchrone Achsen in einem Kanal**

Zum Zeitpunkt t1 wird mit den Kanalachsen X, Y, Z, W1, W2, W3, W4 und W5 gearbeitet.

In Satz N1210 werden die W2-, W3-, W4- und W5-Achse freigegeben und in Satz N1230 werden die Systemachsen SY6, SY7, SY8 und SY9 mit den Namen W2, W3, W4 und W5 übernommen.

Ab jetzt wirken die NPV-Werte der Spalten SY6 bis SY9.

Ab Zeitpunkt t2 kann mit der SY6- bis SY9-Achse bearbeitet werden.

Kanal 1

```

N1100 ZOS(1)
N1110 X0 Y0 Z0 W1 0 W2 0 W3 0 W4 0 W5 0 ← Zeitpunkt t1
;Bearbeite mit X, Y, Z, W1, W2, W3, W4, W5
...
;Entferne W2, W3, W4, W5 aus Achsgruppe
N1210 RAX(W2, W3, W4, W5)
N1220 X0 Y0 Z0 W1 0
;Bearbeite mit X, Y, Z, W1
;Integriere SY6, SY7, SY8, SY9
;mit Namen W2, W3, W4, W5
N1230 GAX(SY6,W2, SY7,W3, SY8,W4, SY9,W5)
N1240 X0 Y0 Z0 W1 0 W2 0 W3 0 W4 0 W5 0 ← Zeitpunkt t2
;Bearbeite mit X, Y, Z, W1, W2, W3, W4, W5
...

```

Abb. 13-10: Beispiel 4 - Mehr als 8 synchrone Achsen in einem Kanal

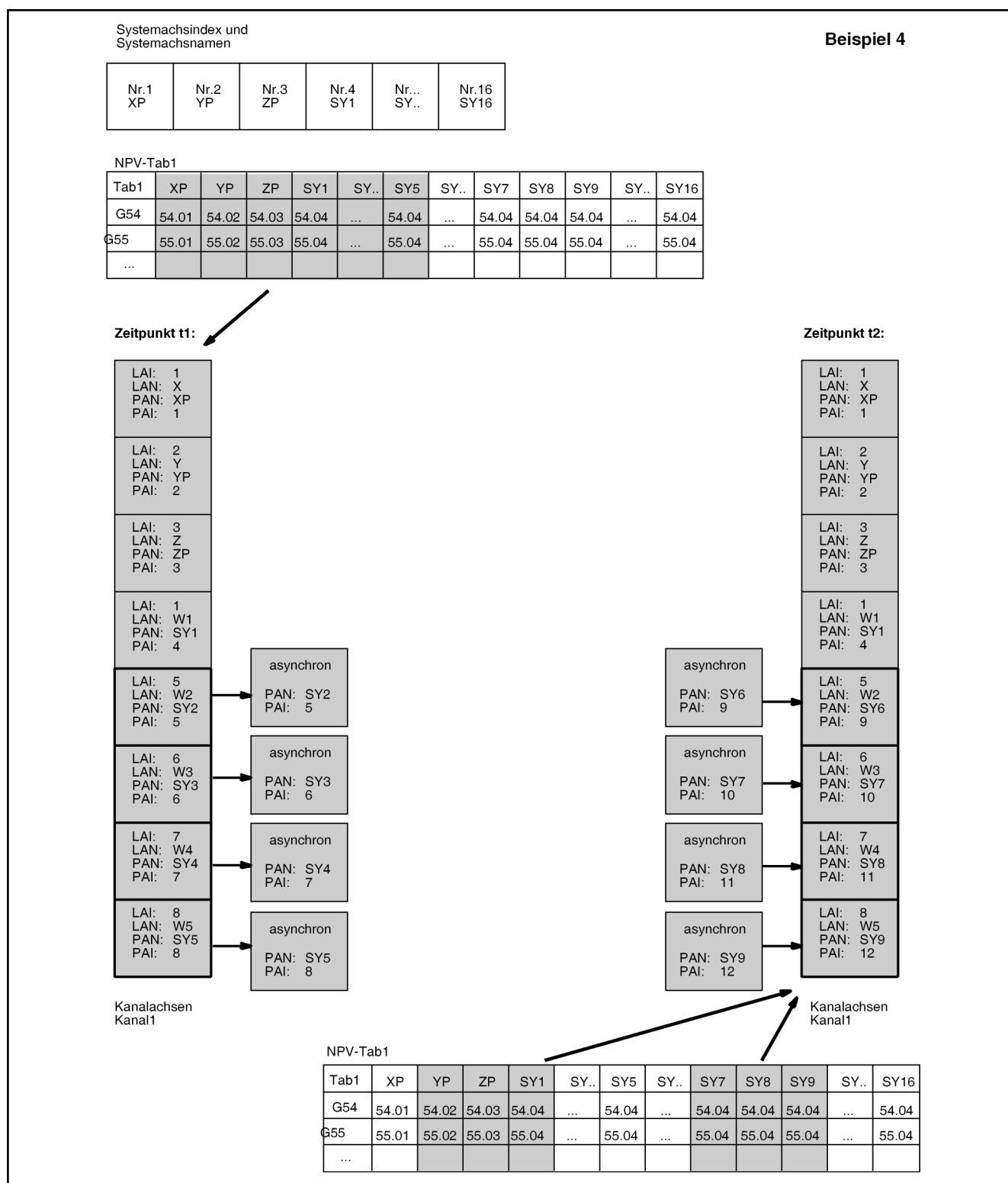


Abb. 13-11: Beispiel 4

Deaktivieren: IW-Engineering / Programmieren:
 DefAxis
 Kurzform DAX

Es sind verschiedene Methoden der Programmierung zu unterscheiden:

- **"DefAxis" (DAX)** direkt im Teileprogramm oder unter Handeingabe programmieren:

Die Programmierung führt zum Laufzeitfehler, wenn die Achse noch nicht freigegeben ist.

Abhilfe: DAX in den Initstring schreiben, wenn die Funktion nur bei Grundstellung (Schlüsselwort #Reset:) oder Gesamtgrundstellung (Schlüsselwort #SysRes:) ausgeführt werden soll.

- Mit **Grundstellung** wird die aktuelle Achszuordnung im Kanal beibehalten. Die zugeordneten Achsen werden jedoch freigegeben, d.h. dem Restsystem zur Verfügung gestellt. Andere Kanäle können diese Achsen anfordern. Bei einer Programmanwahl oder einer Vorgabe in Handeingabe müssen die Achsen von den anderen Kanälen wieder freigegeben worden sein, sonst gibt es eine Fehlermeldung.

Mit **Gesamtgrundstellung** kann wieder auf die Achszuordnung der Maschinenparameter umgeschaltet werden.

- **Programmende (M30):**

Mit NC-Funktionen können fremde Achsen freigegeben oder die Achszuordnung aus den Maschinenparametern übernommen werden. Letzterer Fall kann aber zu Fehlermeldungen führen, wenn Achsen anderweitig zugeordnet sind.

- **Kanalgrundstellung:**

Analog Programmende (M30). Auch hier kann es zu Fehlermeldungen kommen, wenn Achsen anderweitig zugeordnet sind.

14 Kopplungen

14.1 Gantry-Achsen

14.1.1 Beschreibung

Funktion

Gantry-Achsen sind mechanisch starr gekoppelte Achsen, die über getrennte Antriebe parallel verfahren werden. Die Sollwerte des Slaves sind fest an die des Masters gekoppelt (Sollwert-Kopplung). Bauartbedingt kann es bei dem Gantry-Verband zu einer Schräglage, d. h. zu einem Istwert-Versatz der Antriebe untereinander kommen.

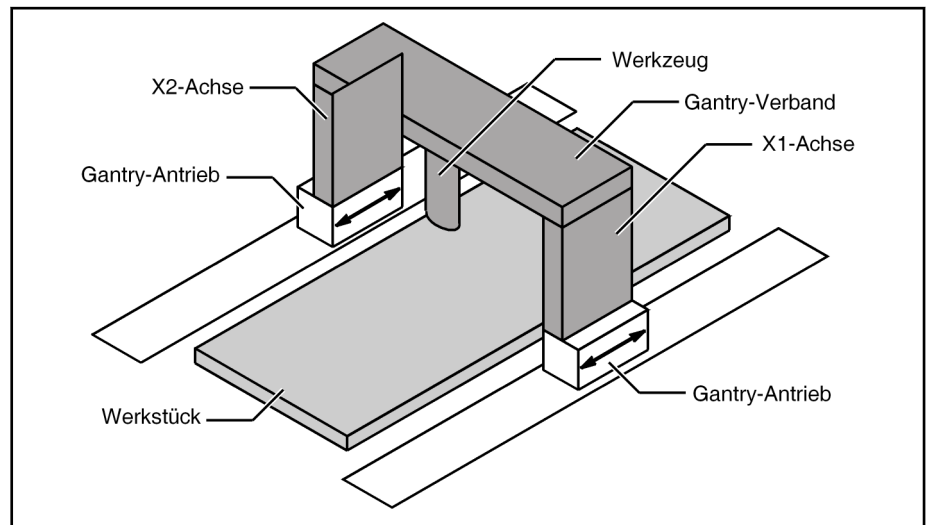


Abb. 14-1: Gantry-Achsen

Voraussetzungen

Eigenschaften des Gantryverbands:

- Schnittstelle:
Die Antriebe sind digitale Antriebe gemäß Sercos.
- Antriebsdaten:
Die Antriebsdaten starr gekoppelter Achsen müssen identisch sein, damit sie sich auch im asynchronen Betrieb (Tippen, Referenzpunktanfahren) identisch verhalten.
Alle Antriebsdaten, die die Achsdynamik betreffen, müssen auch bei Programmierung für alle Gantry-Achsen eines Verbands identisch gehalten werden. Die Daten dürfen nur bei Stillstand der Achsen geändert werden, weil eine Umschaltung nicht in allen Antrieben gleichzeitig (im selben Sercos Takt) erkannt und aktiv wird.
- Sollwert-Kopplung:
Master- und Slaveachse in einem Gantry-Verband fahren gemeinsam in einer Sollwert-Kopplung.
- Messsysteme:
Alle Gantry-Achsen erfordern identische absolute oder inkrementelle (relative) Messsysteme.

Innerhalb eines Gantry-Verbandes sind folgende Achstyp-Konstellationen erlaubt:

Master:	linear	- Slave:	linear
Master:	linear modulo	- Slave:	linear modulo
Master:	endlos	- Slave:	endlos
Master:	rund	- Slave:	rund

Bei Moduloachsen müssen die Modulo-Werte von Master und Slave identisch sein.

Die Masterachse muss eine synchrone Achse sein; für die Slaveachse wird die Einstellung als asynchrone Achse empfohlen!

Sollwert-Kopplung

Damit sich Master und Slaves gemeinsam bewegen, werden deren Sollwerte aneinander gekoppelt ($\text{Sollwert}_{\text{Slave}} = \text{Sollwert}_{\text{Master}}$).

Kommt es zu einem Istwert-Versatz, kann dieser unter bestimmten Bedingungen von den Sollwerten der Achsen als Sollwert-Versatz übernommen werden. In diesem Fall erhält der Slave einen anderen Sollwert als der Master (nämlich einen, um den Sollwert-Versatz versetzten, Sollwert des Masters).

Bei ausgeschalteten Antrieben ist die Sollwert-Kopplung unwirksam. Nach Zuschalten der Antriebe werden die Sollwerte (und damit auch der Sollwert-Versatz) sofort wieder wirksam. Kommt es bei offenem Regler zum Wegdriften der Achsen, entsteht ohne weitere Einflussnahmen ggf. ein Antriebsfehler mit exzessiver Sollwertvorgabe (F2037) oder exzessiver Regelabweichung (F2028), da der Slave die Sollwerte des Masters einnehmen will.

Ein auftretender Versatz der Lageistwerte von Master und Slave wird von der Steuerung von allein nicht ausgeglichen. Ein aus dem Istwert-Versatz gebildeter Sollwert-Versatz kann aktiv sein, jedoch wird auch er nicht von allein von der Steuerung ausgeglichen.

Maßbezug und Messsysteme im Gantry-Verband

Voraussetzung für die Ermittlung der Achsistwerte sind **korrekt** eingestellte Maßbezüge aller Gantry-Achsen bei der Inbetriebnahme (IB).

Die Achsistwerte sind ihrerseits Voraussetzung, um den Versatz zwischen den Gantry-Achsen zu ermitteln und damit als Vorgabe für den Sollwert-Versatz zu dienen.

Die Ermittlung der Maßbezüge hängt vom eingesetzten Messsystem ab.

Zum Einsatz kommen absolute und inkrementelle (mit/ohne abstandskodiertem) Messsysteme:

- Bei absolutem Messsystem ermittelt jede Achse für sich im Stillstand die momentane Achsposition. Sie wird zum achsbezogenen Lageistwert.
Eine eventuell vorhandene Schräglage des Verbands wird nicht ausgeglichen!
- Bei relativem Messsystem mit abstandscodierten Referenzmarken bewegt sich der Achsverband zwischen 2 Referenzmarken. Bei gefundener Marke schalten die Antriebe automatisch auf achsbezogenen Lageistwert um (jede Achse erhält ihren Istwert). Eine eventuell vorhandene Schräglage des Verbands wird nicht ausgeglichen!
- Bei relativem Messsystem ohne abstandscodierten Referenzmarken bewegt sich der Achsverband zum Referenzpunkt und schaltet an definierter Position automatisch auf achsbezogenen Lageistwert um (jede Achse erhält ihren Istwert). Bei Gantry-Achsen muss "Positionieren" appliziert sein (siehe Sercos ID S-0-0147). Bei Auswertung des Referenz-

schalters müssen alle Antriebe an einen Referenzschalter angeschlossen sein!

Eine eventuell vorhandene Schräglage des Verbands wird mit Anfahren des Referenzpunktes ausgeglichen!



Die Maßbezüge werden entsprechend der Antriebsdokumentation Rexroth IndraDrive (Firmware für Antriebsregelgeräte MPH-02, MPB-02, MPD-02) ermittelt.

Schräglagen

Entstehen von Schräglagen: *Eine Schräglage ist ein **Istwert-Versatz**, wenn sichergestellt ist, dass*

- die Achsen korrekt in Betrieb genommen wurden,
- für alle Achsen ist nach dem Hochlauf "Maßbezug hergestellt"
- noch kein Sollwert-Versatz existiert

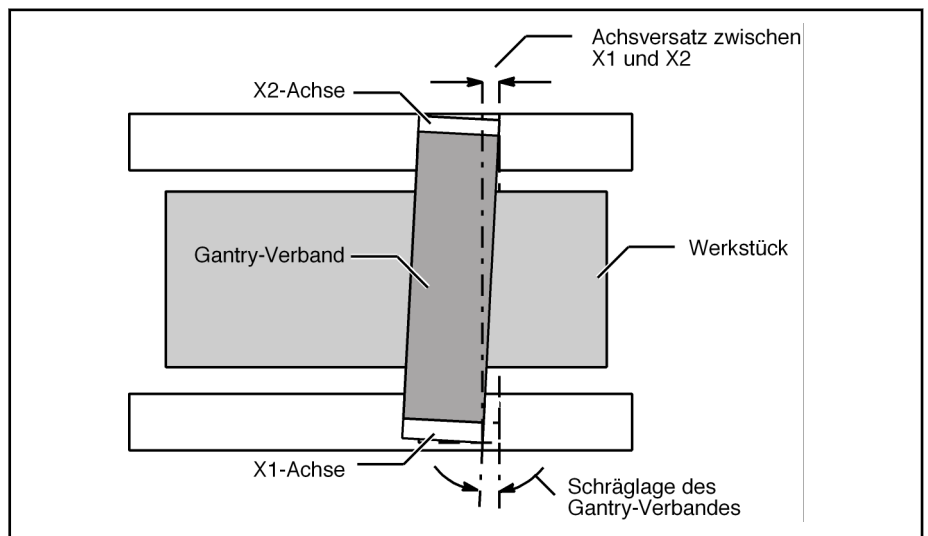


Abb. 14-2: Schräglage

*Eine Schräglage entsteht **im Betrieb** bei einer:*

- unsymmetrischen Blockade
- ungleichen Lastverteilung

Die Istwerte jeder einzelnen Achse folgen den Sollwerten unterschiedlich. Dadurch entsteht der Istwert-Versatz. Da die Sollwerte für Master und Slave jedoch identisch sind, hat die Steuerung außer der **Schräglagenüberwachung** der Istwerte keine Eingriffsmöglichkeit.

Kopplungen

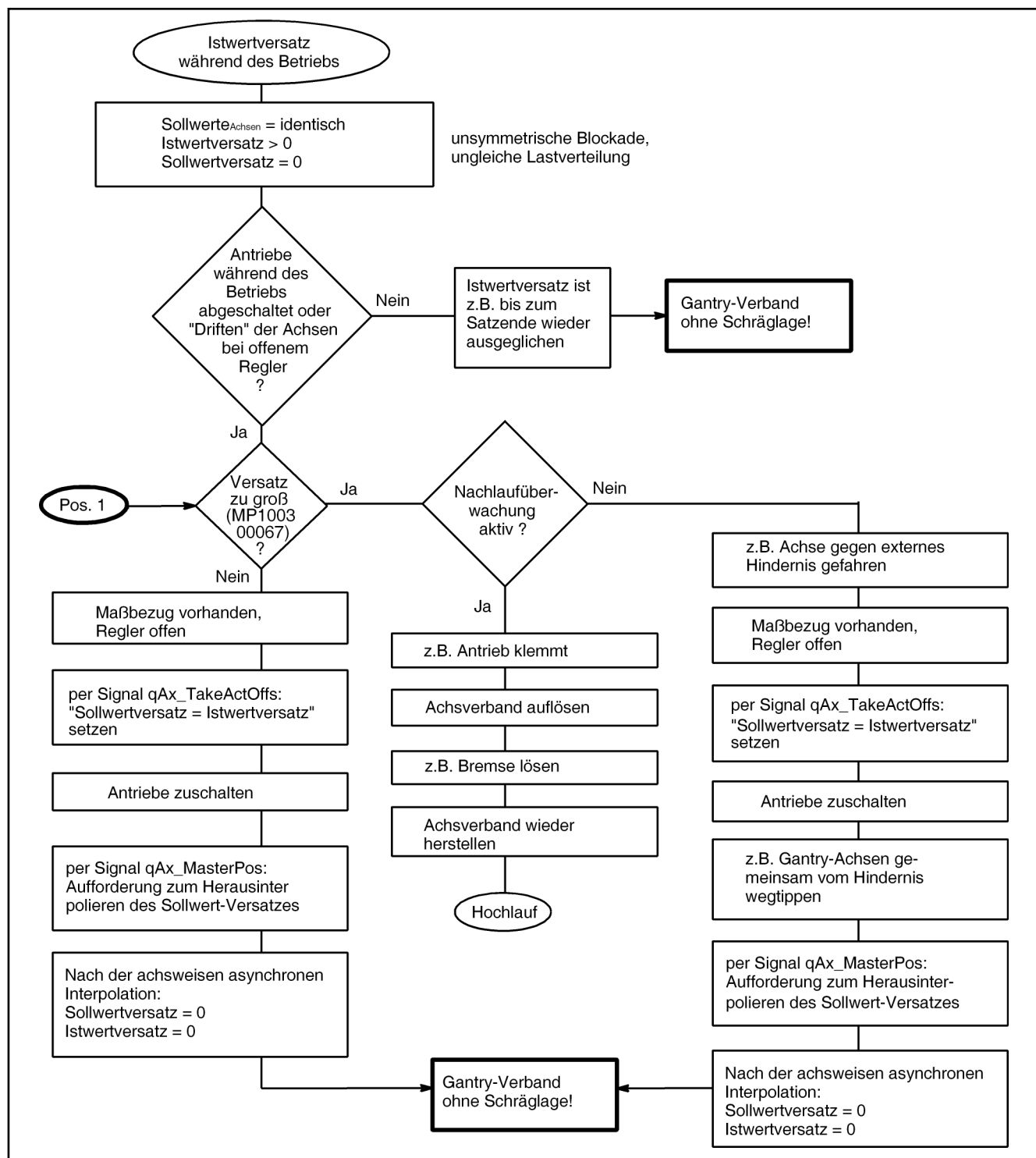


Abb. 14-3: Schräglagenüberwachung

Eine Schräglage bei **ausgeschalteten Antrieben** oder **offenem Regler** entsteht durch:

- Mechanisches Verschieben im antriebslosen Zustand
- Driften von Achsen bei offenem Regler
- Unterschiedliches Bremsverhalten während des Abbremsens

- Einen Istwert-Versatz, der während einer Bahnbewegung nicht mehr ausgeglichen werden konnte, weil die Antriebe ausgeschaltet wurden

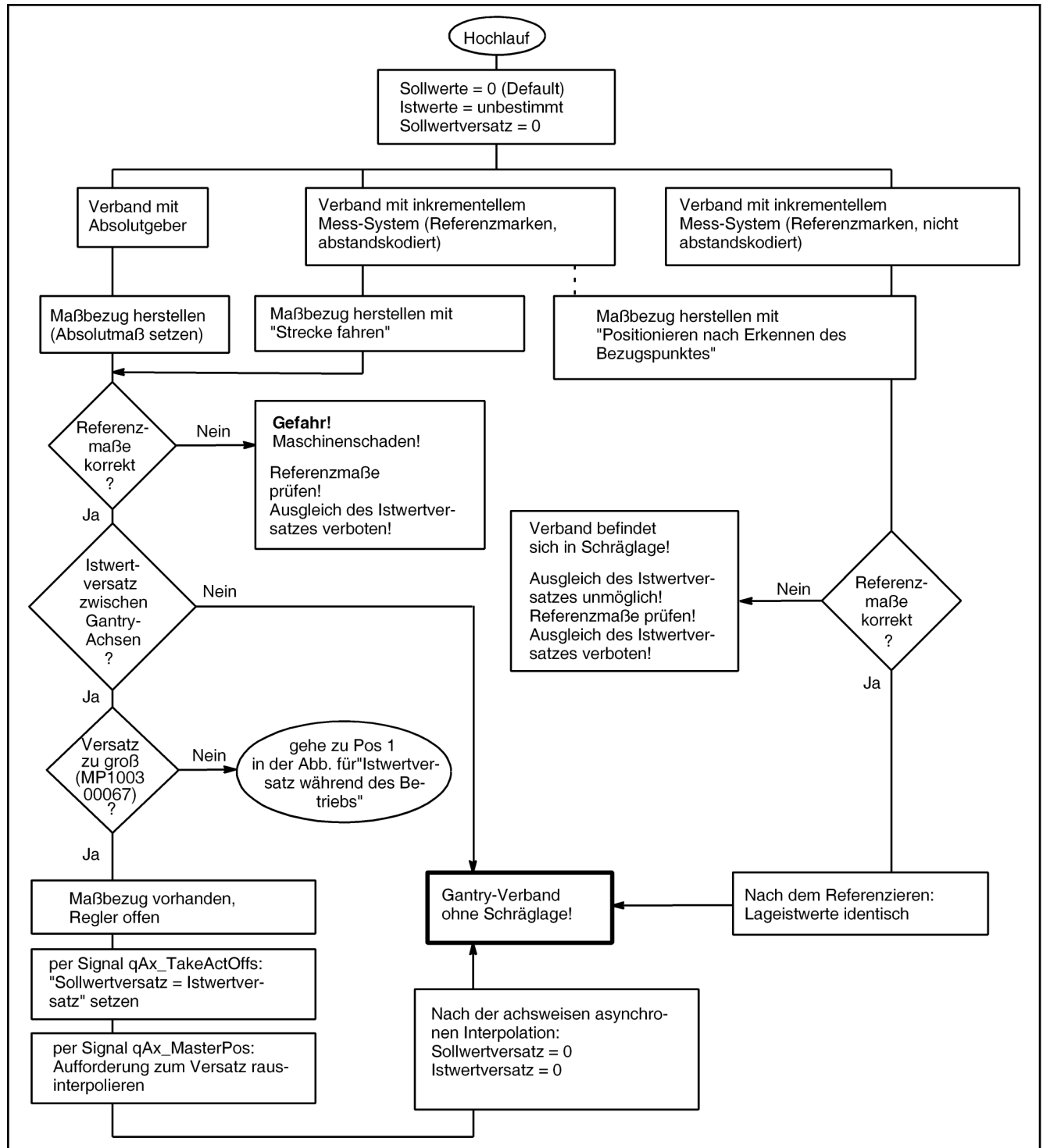


Abb. 14-4: Schräglage bei eingeschalteten Antrieben oder offenem Regler

Ausgleich von Schräglagen:

- **Ausgleich während der Bahnbewegung:**

Entsteht während der Bahnbewegung eine Schräglage, weil die Antrieb-istwerte ihren Sollwerten unterschiedlich folgen, wird dieser Istwert-Versatz selbständig ausgeglichen, wenn die Antriebe den Sollwerten wieder folgen können (z. B. bis Satzende).



Die Schräglage kann von den Antrieben nicht mehr ausgeglichen werden, wenn es vorher zum Laufzeit- oder zum Sercos Fehler kommt oder die Antriebe ausgeschaltet werden.

- **Ausgleich durch Ausgleichsbewegung:**

Die hier dargestellte Funktion führt zum **Ausgleich einer Schräglage** durch eine steuerungsseitig initiierte **interpolatorische Ausgleichsbewegung** der Achsen.

Der Sollwert-Versatz wird an definierten Punkten aus dem Versatz der Istwerte gebildet:

1. Bei Antrieben mit Absolutgebern beim Steuerungshochlauf, wenn der Maßbezug der Antriebe hergestellt ist,
2. bei Antrieben mit inkrementellen Gebern, beim Herstellen des Maßbezugs (Referenzieren) oder
3. bei vorhandenem Maßbezug durch ein Interface-Signal (qAx_TakeActOffs), wenn die entsprechenden Voraussetzungen erfüllt sind.

Existiert ein Sollwert-Versatz, kann er interpolatorisch ausgeglichen werden.

Ein existierender Istwert-Versatz wird durch eine asynchrone interpolatorische Ausgleichsbewegung einer Achse im Gantry-Verband ausgeglichen.

Folgende Bedingungen sind für die Ausgleichsbewegung einzuhalten:

- Die Achsen sind in Referenz (der Maßbezug ist hergestellt).
- Die Achsen stehen. Keine Verfahrbewegung ist aktiv.
- Die NC hat die Schräglage als Sollwert-Versatz übernommen (iAx_CmdOffsExst = 1).
- Der Schleppabstand muss im zulässigen Bereich liegen (MP 100300060) oder das Achsinterface-Signal qAx_LagErrOff "Kopplungsfehler ausblenden" muss gesetzt sein.
- Der Sollwert-Versatz muss im zulässigen Bereich liegen (MP 100300066, 1003 00067) (iAx_CmdOffsExceed = 0).
- Ist durch die Sicherheitstechnik "sichere Lage" vorgegeben, dann wird nicht ausgeglichen.
- Ist durch die Sicherheitstechnik "sichere Geschwindigkeit" vorgegeben, dann wird mit sicherer Geschwindigkeit ausgeglichen.
- Der Ausgleich des Sollwert-Versatzes läuft mit der in "Handvorschub Mittel" (MP 1005 00004) eingestellten Ausgleichsgeschwindigkeit und mit der Tippbeschleunigung (MP 1010 00002). Das Kanal-Overridepoti ist wirksam.



Werden die Antriebe wieder zugeschaltet (ohne Ausgleich des Istwert-Versatzes), werden für die Slaves sofort die Sollwerte des Masters wirksam. Bei einem zu großen Versatz (außerhalb der Grenzen) kann es dann zu Sercos Fehlern kommen (F2028: exzessive Regelabweichung oder F2037: exzessive Sollwertvorgabe).

Signal-Ablauf der Ausgleichsbewegung:

- Master- und Slaveantrieb müssen ausgeschaltet sein.
- Auftrag zur Übernahme des Istwert-Versatzes als Sollwert-Versatz bei offenem Regler mit dem Achsinterface-Signal qAx_TakeActOffs, Flanke 0 → 1 "Istwert-Versatz übernehmen".

- Der Sollwert-Versatz wird von der NC aus dem Istwert-Versatz gebildet:
iAx_CmdOffsExst "Gantry-Sollwert-Versatz aktiv" ist logisch 1
- Master- und Slaveantrieb zuschalten.
Die Antriebe müssen aktiv sein (iAx_DrvAct).
- Mit dem Achsinterface-Signal qAx_MasterPos, Flanke 0 → 1 "Gantry Slave auf Masterposition" wird der Ausgleich des Sollwert-Versatzes durch asynchrone Interpolation angestoßen.
Die Bewegung erfolgt erst, wenn die Antriebe der Sollwertvorgabe folgen (Antriebsstatus: Bit3==TRUE).
- Der Sollwert-Versatz wird von der Steuerung interpolatorisch herausgefahren:
iAx_Trvcmd "Fahrbefehl " ist logisch 1.

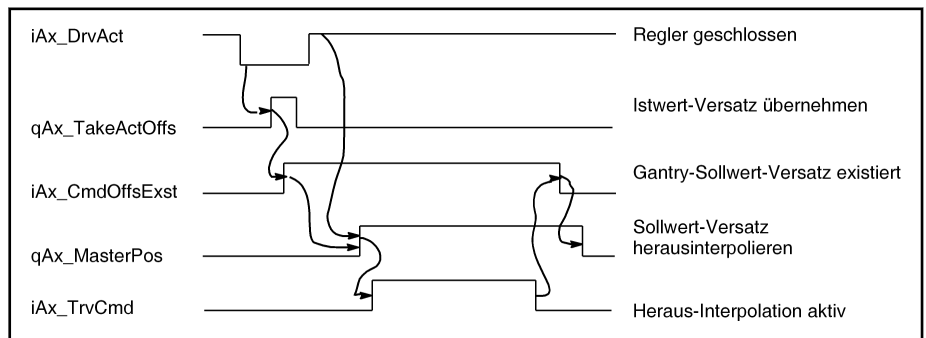


Abb. 14-5: Signal-Ablauf der Ausgleichsbewegung

Abbruch des Ausgleichs:

Die Ausgleichsbewegung wird durch eine fallende Flanke "Gantry-Slave auf Masterposition" (qAx_MasterPos Flanke 1 → 0) abgebrochen. Wird die Ausgleichsbewegung abgebrochen, bevor der Sollwert-Versatz heraus interpoliert ist, bleibt ein Sollwert-Versatz übrig und das Signal "Gantry-Sollwert-Versatz aktiv" (iAx_CmdOffsExst) bleibt auf logisch 1.



Ein Tausch der Antriebe, Geber oder der Antriebs-Software erfordert eine erneute Inbetriebnahme!

Restriktionen

- Eine Schräglage, die sich nicht durch die Lageistwerte erkennen lässt, kann nicht ausgeglichen werden.
- Erreichen die Istwerte einer Achse des Gantry-Verbands die vorgegebenen Sollwerte nicht, dann ist eine Achse blockiert (statische Schräglage). Die Sollwert-Kopplung verhindert einen Ausgleich, da die Blockade nicht erkannt wird.
In diesem Fall muss die Blockade zunächst beseitigt werden, d. h. Antriebe ausschalten, blockierte Achse befreien und danach "Versatz-Ausgleich aller Achsen" oder, wenn Achsverband aufgelöst wurde, "Referenzieren mit Versatz-Ausgleich aller Achsen" durchführen.
- Wird die Schräglage nach dem Hochlauf nicht ausgeglichen, verfahren die Achsen weiterhin versetzt.
- Für die Antriebe ist **kein** "Maßbezug hergestellt". Der Sollwert-Versatz kann nicht ermittelt werden (fehlende Daten für Istwert-Versatz), wodurch die Schräglage unbestimmt ist. Die Steuerung kann den Ausgleich des Sollwert-Versatzes nicht durchführen.

- für die Antriebe ist "Maßbezug hergestellt", d. h. die Bezugsmaße sind **korrekt** übernommen:

Ein Sollwert-Versatz von Achsen, die bei inaktiver Steuerung weggedriftet sind, ist ermittelbar und kann von der Steuerung interpolatorisch ausgeglichen werden.

- Für die Antriebe ist "Maßbezug hergestellt", aber die Bezugsmaße sind **nicht korrekt** übernommen. Man erkennt dies daran, dass die Lageistwerte der Gantry-Achsen unterschiedlich sind. Die Sollwerte des Masters wirken sofort und ohne Versatz auf die Slaves. Dadurch entsteht ein Sercos Fehler (exzessive Regelabweichung bzw. exzessive Sollwertvorgabe).

Die Steuerung könnte ausgleichen, **darf aber nicht**, da der Sollwert-Versatz willkürlich ist (Schräglage durch falsch ermitteltes Bezugsmaß). Der Fehler liegt in der Inbetriebnahme.



Eine "unzulässige Schräglage" der Gantry-Achsen muss vor dem Steuerungshochlauf mechanisch beseitigt werden, wenn die Gantry-Achsen in dieser Schräglage nicht verfahren werden dürfen.

Relevante NC-Funktionen

Alle NC-Funktionen, die eine sinnvolle Programmierung des Gantry-Verbands ermöglichen, sind erlaubt.

Relevante Sercos Parameter

S-0-0041	Referenzfahr-Geschwindigkeit
S-0-0042	Referenzfahr-Beschleunigung
S-0-0115	Geberart 2
S-0-0116/117	Auflösung Geber 1/2
S-0-0052/54	Referenzmaß 1/2
S-0-0147	Referenzfahrparameter
S-0-0150/151	"Referenzmaß-Offset 1/2" bei Motor-Geber/externen Geber. Wird der Referenzmaß-Offset (Sercos Parameter S-x-0150 bei Motor-Geber, S-x-0151 bei ext. Geber) der Masterachse verändert, muss für alle Slaves der Referenzmaß-Offset so angepasst werden, dass die Differenz der Referenzmaß-Offsets (zum Master!) konstant bleiben.
S-0-0051/53	Lageistwert Geber 1/2: Zeigt die aktuelle Position des Motorgebers/ext. Gebers.
S-0-0177/178	"Absolutmaß-Offset" bei Motor-Geber Abstand vom Maschinen-Nullpunkt zum Nullpunkt des Motorgebers mit Absolutmessung.
S-0-0400	Schaltzustand des am Regelgerät angeschlossenen Referenzschalters.
S-0-0403	Status Lageistwert
S-0-0189	Schleppabstand für die Schleppabstands-Überwachung, falls im zyklischen Telegramm enthalten.

S-0-0084	Drehmoment-Istwert für die Momentenüberwachung.
S-0-1050.1.6	Aufnahme des Drehmoment-Istwertes S-0-0084 und des Schleppabstandes S-0-0189 in das zyklische Telegramm in den Sercos Dateien (p2*.scs) für die entsprechenden Gantry-Achsen.

Tab. 14-1: Relevante Sercos Parameter



Die Inbetriebnahme der Gantry-Achsen ist auf IndraDrive-Antriebe mit Sercos Schnittstelle zugeschnitten. Bei der Verwendung von Antrieben (mit Sercos Schnittstelle) anderer Hersteller muss überprüft werden, ob die entsprechende Funktionalität gewährleistet ist und welcher Parameternummer zugeordnet ist!

Relevante IF-Signale

Kanal-IF-Signale	
qCh_OpModeSel_00 bis qCh_OpModeSel_03	
	Anwahl der Betriebsart (hier: Referenzpunktfahren bei inkrementeller Geber mit/ohne abstandskodiertem Messsystem)
qCh_OpModePlc	Betriebsartvorgabe von SPS (hier: Referenzpunkt anfahren inkrementeller Geber mit/ohne abstandskodiertem Messsystem)
Achsinterface-Signale	
iAx_DrvAct	Antrieb in Betrieb (Regler geschlossen)
iAx_MasterIndex0	Index des Masters Bit 0 ...
iAx_MasterIndex6	Index des Masters Bit 6
qAx_TakeActOffs	Istwert-Versatz übernehmen Istwert-Versatz wird für diesen Slave als Sollwert-Versatz übernommen.
iAx_CmdOffsExst	Sollwert-Versatz existiert für diesen Slave (logisch1)
iAx_CmdOffsExceed	Ausgleichbarer Sollwert-Versatz für diesen Slave überschritten (logisch1).
qAx_MasterPos	Gantry auf Masterposition Der ausgewählte Slave erhält den gleichen Sollwert wie der Master. Das Signal bewirkt eine Änderung in der Position des Slaves, wenn die Sollwerte von Master und Slave gleich sind. Wird das Signal zurückgesetzt, bevor der neue Sollwert aktiv ist, bleibt ein Sollwert-Offset erhalten (nur bei interpolatorischen Ausgleich möglich).
iAx_Trvcmd	Fahrbefehl (Interpolatorische Verfahrbewegung aktiv)

Kopplungen

iAx_CoupleLag	Nachlauffehler Jede Achse des Gantry-Verbands erhält dieses Ausgangs-Signal. Es wird ständig aktualisiert, d. h., es bleibt nicht statisch anstehen, wenn ein Fehler aufgetreten war und wieder verschwunden ist! iAx_CoupleLag ist unabhängig von qAx_LagErrOff				
iAx_TrqExceed	Stillstandsfehler Jede Achse des Gantry-Verbands erhält dieses Ausgangs-Signal. Es wird ständig aktualisiert, d. h., es bleibt nicht statisch anstehen, wenn ein Fehler aufgetreten war und wieder verschwunden ist! Im Gantry-Verband ist das Signal iAx_TrqExceed nur für die Masterachse relevant. Die Signale für die Slaveachsen werden nicht ausgewertet. <table> <tr> <td>Low:</td><td>Die Momentensumme aller Achsen liegt innerhalb des erlaubten Bereichs.</td></tr> <tr> <td>High:</td><td>Die Momentensumme aller Achsen hat den erlaubten Bereich überschritten.</td></tr> </table> (iAx_TrqExceed ist unabhängig von qAx_TrqErrOff!)	Low:	Die Momentensumme aller Achsen liegt innerhalb des erlaubten Bereichs.	High:	Die Momentensumme aller Achsen hat den erlaubten Bereich überschritten.
Low:	Die Momentensumme aller Achsen liegt innerhalb des erlaubten Bereichs.				
High:	Die Momentensumme aller Achsen hat den erlaubten Bereich überschritten.				
qAx_LagErrOff	Kopplungsfehler ausblenden Jede Achse des Gantry-Verbands erhält dieses Eingangssignal, es ist jedoch nur für die Masterachse relevant. Ist dieses Signal für die Masterachse gesetzt, wird bei Überschreiten des Nachlaufs die Fehlerauswertung innerhalb der NC unterdrückt. <table> <tr> <td>Low:</td><td>Im Fehlerfall, Fehlerbehandlung/Fehlermeldung durch die NC.</td></tr> <tr> <td>High:</td><td>Im Fehlerfall, keine Fehlerbehandlung/Fehlermeldung durch die NC.</td></tr> </table>	Low:	Im Fehlerfall, Fehlerbehandlung/Fehlermeldung durch die NC.	High:	Im Fehlerfall, keine Fehlerbehandlung/Fehlermeldung durch die NC.
Low:	Im Fehlerfall, Fehlerbehandlung/Fehlermeldung durch die NC.				
High:	Im Fehlerfall, keine Fehlerbehandlung/Fehlermeldung durch die NC.				
qAx_TrqErrOff	Stillstandsfehler ausblenden Jede Achse des Gantry-Verbands erhält dieses Eingangssignal, es ist jedoch nur für die Masterachse relevant. Ist dieses Signal für die Masterachse gesetzt, wird bei Überschreiten des Stillstandsmoments die Fehlerauswertung innerhalb der NC unterdrückt. <table> <tr> <td>Low:</td><td>Im Fehlerfall Fehlerbehandlung/Fehlermeldung durch die NC.</td></tr> <tr> <td>High:</td><td>Im Fehlerfall keine Fehlerbehandlung/Fehlermeldung durch die NC.</td></tr> </table>	Low:	Im Fehlerfall Fehlerbehandlung/Fehlermeldung durch die NC.	High:	Im Fehlerfall keine Fehlerbehandlung/Fehlermeldung durch die NC.
Low:	Im Fehlerfall Fehlerbehandlung/Fehlermeldung durch die NC.				
High:	Im Fehlerfall keine Fehlerbehandlung/Fehlermeldung durch die NC.				
qAx_DrvLock	Vorschub Sperre Vor dem Starten des Referenzierens muss die SPS alle Gantry-Achsen eines Verbandes sperren.				
qAx_JogPlus qAx_JogMinus	Manuell +/- In der Betriebsart "Referenzpunkt anfahren" setzt die SPS die Tipp-Signale für alle Gantry-Achsen.				

Tab. 14-2: Relevante IF-Signale



Zu den Interface-Signalen:

- Das Achsinterface muss sowohl für die Masterachse als auch für die Gantry-Slaveachsen von der SPS verarbeitet werden (das Interface einer Gantry-Slaveachse verhält sich wie das Interface einer durch Programmierung mit einer G-Funktion gekoppelten Slaveachse).
- *Die folgenden antriebsbezogenen Interface-Signale (NC-Eingänge) müssen innerhalb eines Koppelverbands immer identisch gehalten werden:*
 - Antrieb Ein
 - Vorschubsperr

Dies muss von der SPS gewährleistet werden!

Relevante Maschinenparameter (MP)

1003 00060	max. Schleppabstand für gekoppelte Achsen <i>Definiert für jede Achse:</i> • ob die Schleppabstandsüberwachung bei aktiver Achskopplung ein- oder ausgeschaltet werden soll. • den max. erlaubten Schleppabstand einer Slaveachse.
1003 00061	Index der Gantry-Leitachse Definiert, ob es sich bei einer Achse in einem Gantry-Verband um eine Master-oder Slaveachse handelt.
1003 00062	max. Stillstands-Momentensumme eines Gantry-Verbands <i>Definiert für jede Achse, sofern sie die Leitachse eines Gantry-Verbandes ist,</i> • ob die Stillstands-Momenten-Überwachung des entsprechenden Verbandes aktiviert werden soll • den Überwachungsgrenzwert
1003 00063	Synchronlauf-Überwachung beim Gantry-Referenzieren Der Parameter legt fest, ob beim Referenzieren von Gantry-Achsen überwacht wird, ob alle Achsen innerhalb eines Überwachungsfensters ihren Referenzpunkt erreicht haben. Der Parameter ist nur beim Master des Verbandes einzutragen.
1003 00064	Fenster Synchronlauf-Überwachung beim Gantry-"Maßbezug herstellen": Der Parameter gibt die Größe des Überwachungsfensters in msec an. Es wird immer die nächstgrößere Vielfache des Interpolatortakts wirksam.
1003 00065	Referenzschalter-Überwachung beim Gantry-"Maßbezug herstellen" Der Parameter legt fest, ob beim "Maßbezug herstellen" von Gantry-Achsen überwacht wird, ob eine Achse beim Start des "Maßbezug herstellen" auf dem Referenzschalter steht. Der Parameter ist nur beim Master des Verbandes einzutragen.

1003 00066	Gantry-Sollwert-Versatz ausgleichen Parameter legt fest, ob der Sollwert-Versatz zwischen Master- und Slaveachse eines Gantry-Verbands durch Setzen des entsprechenden Interface-Signals ausgeglichen werden darf und wo die maximale ausgleichbare Distanz festgelegt ist. Gilt für jede Slaveachse.
1003 00067	Gantry-Sollwert-Versatz ausgleichen: Obergrenze Legt fest, bis zu welcher maximalen Distanz der Sollwert-Versatz zwischen Master- und Slaveachse durch Setzen des entsprechenden Interface-Signals ausgeglichen werden darf. Gilt für jede Slaveachse.

Tab. 14-3: Relevante Maschinenparameter

Besonderheiten

- Die Gantry-Achsen müssen vor einem Steuerungshochlauf mechanisch momentenfrei (d. h. kein Klemmen) ausgerichtet sein, so dass eine parallele Verfahrbewegung der Achsen nach dem Steuerungs- und Sercos Hochlauf möglich ist.
- Nach jedem Tausch von beteiligten Motoren, Messsystemen und/oder Antriebssoftware ist die Inbetriebnahme der Gantry-Achsen erneut durchzuführen.
- Die für die Inbetriebnahme genannten herstellerspezifischen Sercos Parameter (P-x-xxxx) sind nur für IndraDrive-Antriebe relevant. Bei anderen Antrieben müssen äquivalente Parameter verwendet werden. Die Funktionalität muss vom Antriebshersteller sichergestellt sein.

Überwachungsfunktionen

Überwachungsfunktionen für den Gantry-Verband:

- IB-Überwachung:**
Über ein Zeitfenster (MP) kann der Referenziervorgang aller beteiligten Achsen kontrolliert werden. Hat die erste Achse ihr Referenzieren beendet, startet das Zeitfenster. Innerhalb der vorgegeben Zeit müssen alle anderen Achsen ihr "Referenzieren" beendet haben, sonst wird der Referenziervorgang mit Fehler abgebrochen.
Der relevante MP sollte nach der IB deaktiviert werden, weil sonst das Ausgleichen einer Schräglage beim Referenzieren durch Fehler-Abbruch unterdrückt wird.



Für die Erstinbetriebnahme sollte die Identnummer "Status Lage-istwerte" (S-0-0403) im zyklischen AT enthalten sein. Damit kann im Fehlerfall die Reaktionszeit der Steuerung minimiert werden.

- Überwachung des Referenzschalters:**
Eine Achse darf beim Starten des "Maßbezug herstellen" niemals auf dem Referenzschalter stehen, da dies eine Richtungsumkehr beim "Maßbezug herstellen" zur Folge hätte. Die Gantry-Achsen würden sich ggf. in entgegengesetzte Richtungen bewegen.
Eine Überwachung verhindert das Starten des "Maßbezug herstellen" mit einem Laufzeitfehler, wenn eine der Gantry-Achsen auf dem Referenzschalter steht (Schaltzustand "betätigt" des Referenzschalters im Parameter S-0-0400 im zyklischen AT).
- Überwachung Schleppabstand:**

Während der Inbetriebnahme und im Betrieb kann mit MP 100300060 (Toleranzmaß = max. Schleppabstand für gekoppelte Achsen) der Schleppabstand überwacht werden.

Dazu verwendet die Steuerung die Nachlauf-Informationen von Master und Slave aus dem zyklischen Sercos Parameter S-0-0189. Ist S-0-0189 nicht im zyklischen Sercos Telegramm enthalten, wird der Schleppabstand aus den Lageistwerten der Gantry-Achsen errechnet. Bei Überschreiten des maximal zulässigen Schleppabstandes zwischen Master und Slave wird der Verband bis zum Stillstand abgebremst und ein Koppelfehler erzeugt.

Fehlerrückmeldung:

Beim Zustandekommen einer unzulässigen Schräglage wird für die betroffene(n) Achse(n) das Interface-Signal iAx_CoupleLag "Nachlauffehler" gesetzt. Bei Unterschreiten des Schleppabstandes wird das Signal wieder weggenommen. Der Fehlerzustand in der NC kann mit Setzen des Achssignals qAx_LagErrOff für die Masterachse unterdrückt werden. In diesem Fall muss in der SPS eine Überwachung des Koppelnachlaufs programmiert werden.



Die Kopplung wird auch im Fehlerfall nicht geöffnet.

• Momenten-/Stromüberwachung:

Die Momenten-/Stromüberwachung (Stillstandsüberwachung) funktioniert nur im **statischen Zustand**, d. h. alle Gantry-Achsen sind "in Position" und es steht kein Fahrbefehl von der NC an.

Hierbei werden aus dem Sercos Telegramm die aktiven Momente der Achsen ermittelt.

Die Momentenüberwachung reagiert, wenn die **Summe** der Momente aller Antriebe eines Gantry-Verbandes die in dem MP 003 00062 festgelegte **max. Stillstands-Momentensumme** überschreitet. Die Summe der Momente im "Stillstand" ist ein qualitatives Maß für die Verspannung der Mechanik. Die quantitativen Größen müssen vom Maschinenhersteller festgelegt werden. Sie sind über Testreihen zu ermitteln.



Eine Auswertung des Momentenverlauf der Antriebe ist während der Bewegung nicht sinnvoll, da die Antriebsmomente währenddessen zu stark schwanken können.

Sercos Einstellung:

Damit die Momentenüberwachung durchgeführt werden kann, muss der Drehmoment-Istwert (S-0-0084) der Gantry-Antriebe im zyklischen Sercos Telegramm übertragen werden (Eintrag in der Sercos Datei "p2<xxx>.scs").

Fehlerrückmeldung:

Zeigt die Momenten-/Stromüberwachung eine außerordentliche Schräglage an, ist zunächst die **Fehlerursache** zu beseitigen, ehe die Schräglage durch Referenzieren beseitigt wird! Ein Referenzierversuch im "fehlerhaften" Zustand führt zum erneuten Ansprechen der Überwachung.

Beim Überschreiten der Momentensumme:

- wird das Achssignal iAx_TrqExceed für die Masterachse gesetzt, beim Unterschreiten wieder zurückgesetzt.

- kommt es zu einer Fehlermeldung durch die NC, die mit Grundstellung gelöscht werden kann.

Diese Fehlermeldung kann durch Setzen des Achssignals qAx_TrqErrOff unterdrückt werden. Das Ausgangssignal iAx_TrqExceed ist unabhängig von qAx_TrqErrOff!

Anzeige der Gantry-Slaveachsposition:

Eine Positionsanzeige für eine Gantry-Slaveachse ist nur auf Basis von Istpositionen möglich. Die von der NC generierten Sollwerte erscheinen nicht in allen Koordinatensystemen, so dass eine Anzeige der Slavepositionen im Allgemeinen keinen Sinn ergibt. Sie sollte ausgeblendet werden.

Empfehlung:

Ein Gantryslave sollte, wenn möglich, als asynchrone Achse vereinbart werden.

14.1.2 Handlungsanweisung: Gantry-Achsverband applizieren

Applizieren: Ablauf einer Gantry-Inbetriebnahme und "Maßbezug herstellen".

Voraussetzungen:

Für alle Achsen des Gantry-Verbands werden Maßbezüge ermittelt. Der auf diese Maßbezüge basierende Vorgang zum "Maßbezug herstellen" während der Erst-IB sollte mit schwachem KV, niedriger Geschwindigkeit, niedriger Beschleunigung bzw. Verzögerung sowie einer Momentenüberwachung vollzogen werden.

VORSICHT

Maschinenschaden und/oder Personenschaden durch falsch ermittelte Maßbezüge!

Erst-Inbetriebnahme mit angemessenen Achsparametern durchführen, um größere Schäden zu vermeiden!

Der Gantry-Verband muss sich in einem mechanisch momentenfrei ausgerichteten Zustand befinden:

Bei **schweren Portal-Gantry-Verbänden** müssen einzelne Gantry-Achsen mechanisch oder durch Tippen in eine parallele Lage zueinander gebracht werden. Dies erfordert jedoch, dass die Gantrykopplung der Achsen im MP 1003 00061 (alle Parameterwerte = 0) aufgehoben ist, damit Master- und Slaveachsen nach dem Hochlauf **einzel**n getippt werden können.

Wählen Sie dazu folgenden Ablauf:

1. NC und Antriebe hochlaufen lassen.
2. Stellen Sie den MP 1003 00061 so ein, dass keine Achse einem Gantry-Verband angehört.
3. NC und Antriebe erneut hochlaufen lassen.
4. Tippen Sie die einzelnen Achsen auf eine Position, auf der sie parallel zueinander stehen und eine ggf. noch vorhandene Schräglage durch das anschließende Referenzieren ausgeglichen werden kann.
5. Definieren Sie die Zugehörigkeit der Gantry-Achsen im MP 1003 00061 erneut.

Bei **relativ starren Gantry-Verbänden** ist davon auszugehen, dass keine Verspannungskräfte vorhanden sind.

Gantry-Achsen definieren:

MP 1003 00061 Index der Gantryleitachse. Definiert für **jede** Achse, ob sie zu einem Gantry-Verband gehört. Für alle Achsen eines Gantry-Verbands (inkl. der Leitachse) geben Sie den Achsindex (=die Achsnummer) der Leitachse ein.

Maßbezug herstellen (Referenzieren):

Zum Einsatz kommen absolute und inkrementelle (mit/ohne abstandskodiertem) Messsysteme. Die Maßbezüge werden entsprechend der Antriebsdokumentation (für IndraDrive) ermittelt.



Für die Inbetriebnahme ist die Ermittlung des Maßbezugs für einzelne Antriebe ausführlich in der Dokumentation "Rexroth IndraDrive Firmware für Antriebsregelgeräte MPH-02, MPB-02, MPD-02" dargestellt.



Zum Ermitteln des Maßbezugs in der IB dürfen die Achsen keine Schräglage aufweisen.

Maßbezug beim absoluten Messsystem:

Inbetriebnahme:

Sind die Achsen ausgerichtet, wird durch Setzen identischer Lageistwerte der Maßbezug aller Gantry-Achsen gesetzt ("Absolutmaß setzen"). Dies sollte ohne Reglerfreigabe geschehen. Da bei Reglerfreigabe die Sollwerte der Steuerung wieder wirken kann es zu einem Sollwertsprung kommen wenn die Istwerte vor dem "Absolutmaß setzen" unterschiedlich waren. Dies kann nur durch einen Steuerungshochlauf verhindert werden.

Maßbezug herstellen:

Beim Steuerungs- bzw. Sercos Hochlauf wird Maßbezug hergestellt, d.h. die Lageistwerte jeder Achse sind **sofort** bekannt. Für jeden Antrieb ist "Maßbezug hergestellt", wenn die Bezugsmaße der Achsen (mit Absolutmaße setzen bei der IB) korrekt übernommen wurden.

"Versatz" ausgleichen

Der Ausgleich eines, im Betrieb oder im Ausschaltzustand entstandenen Istwertversatzes wird nur durch das Setzen von dem Interface-Signal **qAx_TakeActOffs** "Istwert-Versatz übernehmen" initiiert. Das nachfolgende "Herausinterpolieren" des Versatzes wird mit dem Setzen des Interface-Signals **qAx_MasterPos** "Gantry-Slave auf Masterposition" gestartet.

Sind die Lageistwerte der Gantry-Achsen nach der Inbetriebnahme nach dem Steuerungshochlauf im Gantry-Verband unterschiedlich (z. B. Driften bei offenem Regler), entspricht der Istwert-Versatz einer Schräglage zwischen den Gantry-Achsen.

Der Versatz kann nun durch das Signal **qAx_TakeActOffs** Flanke 0 → 1 "Istwert-Versatz übernehmen" als Sollwert-Versatz übernommen werden.

Mit Signal **qAx_MasterPos**, Flanke 0 → 1 "Gantry-Slave auf Masterposition" wird der Istwert-Versatz durch asynchrone Interpolation durch die Steuerung ausgeglichen.



Der Gantry-Verband muss sich in einem mechanisch momentenfrei ausgerichteten Zustand befinden.

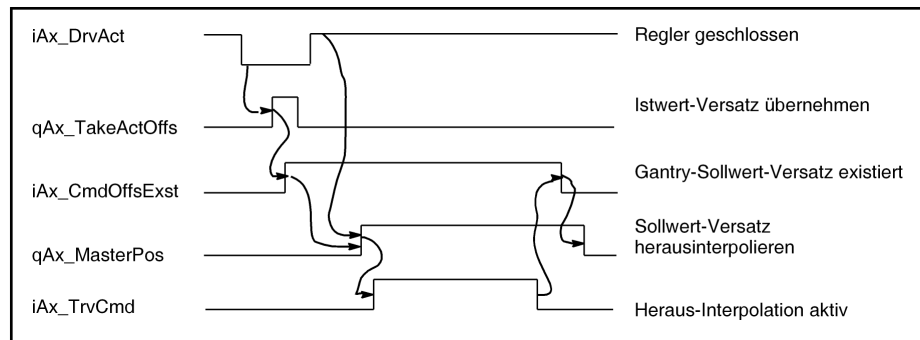


Abb. 14-6: Signalflussplan

Nach dem Versatzausgleich sind die Lageistwerte aller Gantry-Achsen wieder identisch.

Restriktionen:

- Ist kein Bezugspunkt verfügbar, ist für die Antriebe kein "Maßbezug hergestellt". Der Sollwert-Versatz ist unbestimmt und ein Ausgleich der Schräglage durch die Steuerung ist nicht möglich.
- Ist für die Antriebe "Maßbezug hergestellt", aber das Absolutmaß ist falsch ermittelt, dann ist der Sollwert-Versatz unbestimmt. Wird versucht, den Istwert-Versatz auszugleichen, führt dies ggf. zu Maschinenschäden.

HINWEIS

Maschinenschäden durch falsches "Absolutmaß setzen" möglich!

Inbetriebnahme wiederholen!

Maßbezug bei inkrementellem Messsystem:

Nur Antriebe mit inkrementellen Messsystemen dürfen referenziert werden.

Das Referenzieren darf keinesfalls direkt über ein Sercos Kommando gestartet werden (z.B. über Drivetop).

Ablauf des Referenzierens:

- Vor dem Starten des "Maßbezug herstellen" muss die SPS alle Gantry-Achsen eines Verbands über das Interface-Signal **qAx_DrvLock** "Vorschubsperrung" sperren. (**qAx_DrvLock = 1**).
- Dann setzt die SPS die, am Interface für alle Gantry-Achsen die Betriebsart "Referenzpunkt anfahren" und setzt die Jogg-Signale für diese Achsen.
- Stehen am Interface für alle Gantry-Achsen die Fahrbefehle an, nimmt die SPS das Signal "Vorschubsperrung" weg.
- Jetzt fahren die Antriebe die Referenzpunkte an.

Achtung:

Der Signal-Ablauf muss zwingend eingehalten werden!

Maßbezug beim abstandskodierten inkrementellen Messsystem:

Bei Verwendung von abstandskodierten inkrementellen Messsystemen sind die Lageistwerte nach dem Steuerungs- bzw. Sercos Hochlauf **nicht** bekannt. Bevor kein "Maßbezug herstellen" hergestellt ist, existiert kein gültiger Sollwert-Versatz zwischen den Achsen.

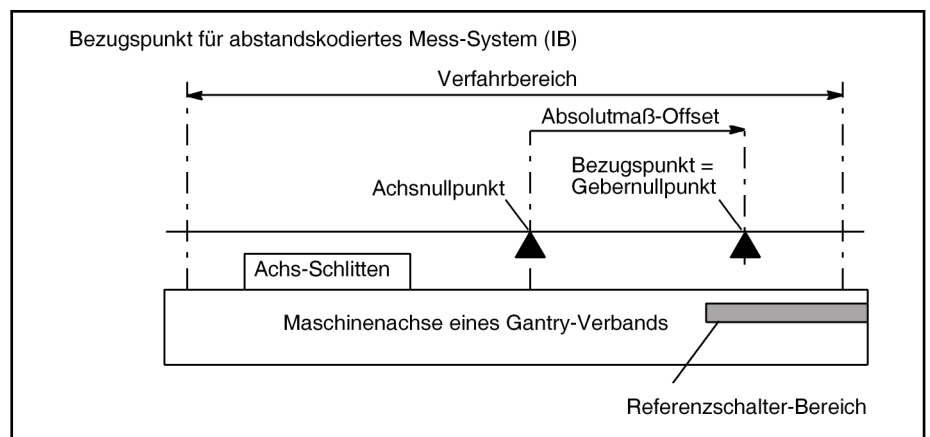


Abb. 14-7: Maßbezug beim abstandskodierten inkrementellen Messsystem

Inbetriebnahme:

Bei der Inbetriebnahme **muss** der Parameter S-0-147 für jede Achse auf **"Strecke fahren"** (Bit 7 = 0, Bit 8 = 1) gesetzt sein, damit die Antriebe nach dem "Maßbezug herstellen" auf einer beliebigen Position stehen dürfen. Der Maßbezug wird durch das Überfahren zweier Referenzmarken hergestellt. Nach dem Beenden von "Maßbezug herstellen" ist der Lageistwert des Gebernullpunkts gleich dem in S-0-177 bzw. S-0-178 festgelegten **Absolutmaß-Offset**. Der maschinenspezifische Achsnullpunkt muss durch die Vorgabe in S-0-177 bzw. S-0-178 angepasst werden. Wird später mit "Positionieren nach Erkennen des Bezugspunkts" ein Maßbezug hergestellt, muss das Referenzmaß (S-0-52/S-0-54) eingetragen werden.

Maßbezug herstellen:

Nach dem Steuerungs- bzw. Sercos Hochlauf werden die Lageistwerte, und damit ein eventueller Istwert-Versatz, durch "Maßbezug herstellen" mit "Strecke fahren" oder "Positionieren nach Erkennen des Bezugspunkts" ermittelt.



Vor dem Starten des "Maßbezug herzustellen" muss die SPS alle Gantry-Achsen eines Verbands über das Interface-Signal "Vorschubsperrung" sperren.

Wird mit **"Strecke fahren"** ein Maßbezug hergestellt, fahren die Achsen hierfür ein kurzes Stück über 2 abstandskodierte Referenzmarken und ermitteln daraus den Lageistwert (für einzelne Antriebe ist "Maßbezug hergestellt", wenn die Bezugsmaße der Achsen korrekt übernommen wurden).

Sind die Lageistwerte der Achsen des Gantry-Verbands nach "Maßbezug herstellen" unterschiedlich, entspricht der Istwert-Versatz einer Schräglage zwischen den Gantry-Achsen. Der Versatz kann nun durch das Signal qAx_TakeActOffs, Flanke 0 → 1 "Istwert-Versatz übernehmen" als Sollwert-Versatz übernommen werden. Mit Signal qAx_MasterPos, Flanke 0 → 1 "Gantry Slave auf Masterposition" wird der Istwert-Versatz durch asynchrone Interpolation ausgeglichen.

Nach Versatzausgleich sind die Lageistwerte aller Gantry-Achsen identisch.

Wird mit "Positionieren nach Erkennen des Bezugspunkts" Maßbezug hergestellt, muss das Referenzmaß (S-0-52/S-0-54) richtig eingetragen sein. Ist die IB korrekt erfolgt, ist eine evtl. vorhandene Schräglage, die sich vor dem "Maßbezug herstellen" eingestellt hatte, mit dem Vorgang "Maßbezug herstellen" beseitigt.

Beispiel:

Maßbezug herstellen bei Gantry-Achsen mit abstandskodierten inkrementellen Messsystem:

1. Maßbezug herstellen der Gantry-Achsen bei der Inbetriebnahme:

Voraussetzung:

- Die Referenzfahrparameter der Gantry-Achsen (S-x-0041: Geschwindigkeit und S-x-0042 Beschleunigung) müssen so vorgegeben werden, dass auch bei Fehlern bei der Inbetriebnahme (z.B. falsche Versatzmaße) während "Maßbezug herstellen" keine Schäden an der Mechanik auftreten. Z. B. können Überwachungsmechanismen (Momenten/Stromüberwachung) die Bewegung rechtzeitig abbrechen.
- Referenzfahrparameter: S-0-0147 Bit 7 = 0 und Bit 8 = 1.
Diese Einstellung muss solange erhalten bleiben, bis die korrekten Referenz-Versatzmaße in die Sercos Dateien eingetragen und von den Antrieben durch einen Sercos oder Steuerungshochlauf übernommen worden sind.
- Vor dem Starten des "Maßbezug herstellen" muss die SPS alle Gantry-Achsen eines Verbandes über das Interface-Signal **"Vorschubsperr"** sperren.
Dann setzt die SPS am Interface für alle Gantry-Achsen die Betriebsart "Referenzpunkt anfahren" und die Tipp-Signale für diese Achsen.
Stehen am Interface für alle Gantry-Achsen die Fahrbefehle an, nimmt die SPS das Signal "Vorschubsperr" weg. Danach fahren die Antriebe die Referenzpunkte an.



Der Signal-Ablauf muss zwingend eingehalten werden!

Für Master und Slave(s) wird jeweils der **Referenzmaß-Offset** ermittelt. Da die Geber nie gleich exakt montiert sind (IB-Distanz), muss diese IB-Distanz ebenfalls bei den Referenzmaß-Offsets der Antriebe berücksichtigt werden.

Durchführen des "Antriebsgeführten Referenzierens":

- Kommando "Antriebsgeführtes Referenzieren" wird gesetzt und freigegeben, nachdem den notwendigen Steuer- und Statussignalen über den Service-Kanal vom Master Echtzeitbits zugewiesen sind.
- Der Antrieb schaltet auf interne Lageregelung, löscht den Status Lageistwerte (S-0-0403) und ignoriert die zyklische Sollwertvorgabe.
- Unter Berücksichtigung der im Referenzfahrparameter S-0-0147 vorgegebenen Anfahrrichtung beschleunigt der Antrieb mit der Referenzfahr-Beschleunigung S-0-0042 auf die Referenzfahr-Geschwindigkeit S-0-0041.
- Der Antrieb bremst mit dem Wert aus S-0-0042 bis zum Stillstand. Gleichzeitig berechnet er den Lageistwert und setzt den Status Lageistwerte (S-0-0403).
Ggf. müssen die Achspositionen bzgl. des Maschinen-Nullpunktes vermessen werden.

2. Sercos Parameter setzen

- Tragen Sie die ermittelten **Referenzmaß-Offsets** der Achsen in das Sercos Datei **p3*.scs** ein.

- Nach Abschluss des "Maßbezug herstellen" bei der Inbetriebnahme muss der Sercos **Referenzfahrparameter** für das "Anfahren des Referenzpunktes" wieder gesetzt werden (S-0-0147 Bit 7 = 1), damit der Gantry-Verband nach dem Referenzieren ausgerichtet ist.
- Der Parameter P-0-0504 **Zustand Nockenposition** (Referenzpunktschalter) wird entweder durch das Kommando "Arbeitsspeicher sichern" (S-0-0264) gesichert oder er muss durch den Inbetriebnehmer gesetzt werden.



Für die Sercos Dateien **p3*.scs** aller Gantry-Achsen gilt:

Bis auf das Referenzversatzmaß S-x-0150, S-x-0151, S-0-0177, S-0-0178 (abhängig vom Geber-Typ) müssen alle Werte identisch sein.

3. Parameter übernehmen durch erneuten Hochlauf.



Sind die **Referenzmaß-Offsets der Achsen noch nicht korrekt eingetragen und übernommen** und der Sercos Parameter **Referenzpunktzustand (P-0-0504)** ist noch nicht im Antrieb gesichert.

Dann darf der Referenzpunkt nicht angefahren werden (S-0-0147 Bit 7 = 0)!

HINWEIS

Schäden an der Mechanik des Gantry-Verbands!

Die Referenzierparameter der Gantry-Achsen (S-X-0041: Geschwindigkeit und S-X-0042 Beschleunigung) müssen so vorgegeben werden, dass auch bei Fehlern bei der Inbetriebnahme (z.B. falsche Versatzmaße) während des "Maßbezug herstellen" keine Schäden an der Mechanik auftreten.

Sorgen Sie für eine geeignete Überwachung (z. B. Momenten-/Stromüberwachung), um ggf. die Bewegung rechtzeitig abbrechen zu können!

HINWEIS

Maschinenschäden durch falschen Maßbezug möglich!

Inbetriebnahme wiederholen!

Maßbezug beim nicht abstandskodierten inkrementellen Messsystem:

Bei Verwendung von nicht abstandskodierten inkrementellen Messsystemen sind die Lageistwerte nach dem Steuerungs- bzw. Sercos Hochlauf nicht bekannt. Bevor kein "Maßbezug herstellen" hergestellt ist, existiert kein gültiger Sollwert-Versatz zwischen den Achsen.



Vor dem Starten des "Maßbezug herzustellen" muss die SPS alle Gantry-Achsen eines Verbands über das Interface-Signal "Vorschubsperrung" sperren.

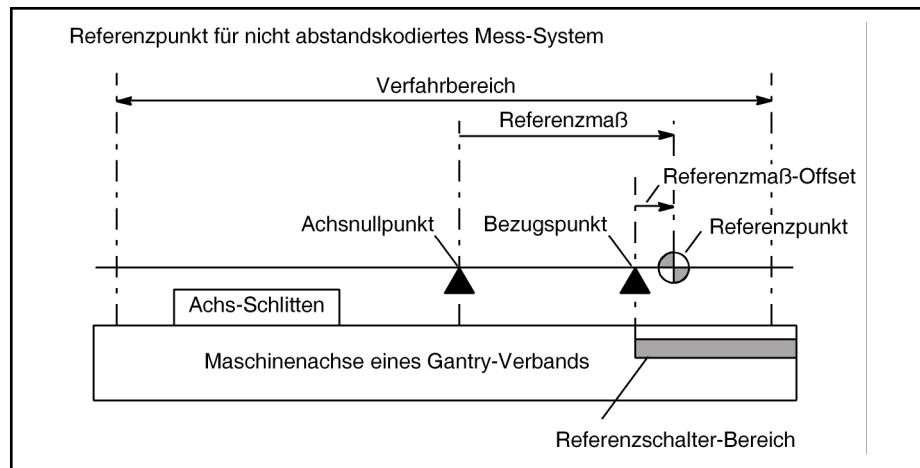


Abb. 14-8: Referenzpunkt für nicht abstandskodiertes Messsystem

Inbetriebnahme:

Es ergeben sich zwei Möglichkeiten zur Ermittlung der korrekten Bezugsmaße:

- Messsystem ist ausgerichtet:

Ist das Messsystem an jeder Achse **ausgerichtet**, sind die Nullmarker für jede Achse identisch. Der Referenzmarken-Offset und das Referenzmaß zum Referenzpunkt kann beliebig, aber für alle Antriebe identisch, eingestellt werden.

- Messsystem ist nicht ausgerichtet:

Folgende Vorgehensweise wird empfohlen (momentenfreie Gantry-Achsen vorausgesetzt):

- Ist das Messsystem an der Achse **nicht ausgerichtet**, muss in der IB das Referenzmaß-Offset mit "Marker suchen" P-0-14 für alle Gantry-Achsen ermittelt werden. Die Achsen können z. B. im Tipbetrieb bewegt werden, bis der Nullmarker aller Gantryachsen Masterachse sicher erkannt sind. Das entsprechende Maß ist in die Parameter einzutragen (Referenzversatzmaß, siehe IndraDrive-Dokumentation).
- für das Referenzieren muss "Positionieren nach Erkennen des Bezugspunkts" eingestellt sein (S-0-147: Bit 7 0 = 1 Bit 8 = 0).
- Zum Erkennen des Nullmarkers wird jede Achse im Tipbetrieb oder durch inkrementelle Programmierung bewegt, bis der Nullmarker (Referenz-Marke) der **Masterachse** sicher durch den Referenzschalter erkannt wird (da Nullmarker mehrfach im gesamten Verfahrbereich auftreten können). Der Verfahrweg muss größer sein, als der Referenzmarkenabstand!

Diese "Markerposition" A wird für jede Achse aus den Antriebsparameter (P-0-173) gelesen. (siehe auch Dokumentation "Rexroth IndraDrive Firmware für Antriebsregelgeräte MPH-02, MPB-02, MPD-02").

- Das Referenzmaß wird nun für alle Achsen identisch auf einen Wert gesetzt, der größer ist, als der Weg einer Geberumdrehung + Referenzier-Bremsweg!

Damit fahren die Achsen am Ende des "Maßbezug herstellen" nach dem Erfassen der Referenzmarke auf eine Referenzlage ohne die Richtung umzukehren.

- Das Referenzmaß-Offsets für jede Achse ergibt sich, indem vom Referenzmaß die jeweilige Markerposition abgezogen wird.
- Die Daten sind in der Sercos Datei zu speichern und durch einen Sercos Hochlauf zu übernehmen (optional mit dem Tool "DriveTop").

Anmerkung zum "Gittersprung":

Gegebenenfalls wird die, beim "Marker suchen", **erste** ermittelte Referenzmarke aller Gantry-Achsen beim anschließenden Referenzieren nicht **zuerst** erfasst, sondern erst nach einer weiteren Geberumdrehung, was zu einem mechanischen Verziehen der Achsen führt. Der bei der IB ermittelte Referenzmaß-Offset ist somit unzulässig



Diese Fehlerquelle kann bei der IB nicht ausgeschlossen werden.

Um dies abzuwenden und um größeren Schaden zu verhindern ist die "IB-Überwachung" (siehe weiter unten) zu aktivieren. Ist diese aktiviert, wird erfasst, welche Achse beim Referenzieren als erste ihre Referenzmarke erkannt hat. Im Fehlerfall wird diese Achse angezeigt. Diese Achse muss das größte Referenzversatzmaß haben. Ist dies nicht der Fall, müssen alle Referenzversatzmaße die kleiner oder gleich diesem Wert sind, um den Referenzmarkenabstand erhöht werden (Beachte: Geberart S-0-0115 und Geberauflösung S-0-0116, S-0-0117).



Besitzt ein Geber mehrere Referenzmarken, muss der Abstand zweier Referenzmarken konstant sein.

Maßbezug herstellen:

Das "Maßbezug herstellen" nach der IB erfolgt durch Anfahren des Referenzpunktes. Hierfür muss "Positionieren nach Erkennen des Bezugspunkts" eingestellt sein. Außerdem muss das Referenzmaß (S-0-52/S-0-54) richtig eingetragen sein. Eine Schräglage ist in der Referenzpunktposition nicht vorhanden.



Vor dem Starten des "Maßbezug herzustellen" muss die SPS alle Gantry-Achsen eines Verbands über das Interface-Signal "Vorschubsperrung" sperren.

Restriktionen:

- Beim "Maßbezug herstellen" der Gantry-Achsen darf der Nullmarker nicht angefahren werden, da die Lage des Null-Markers einbauabhängig ist.
- Nach der IB muss für das "Maßbezug herstellen" "Positionieren nach Erkennen des Bezugspunkts" eingestellt sein, das Referenzmaß (S-0-52/S-0-54) muss richtig eingetragen sein.
- Halten nach Erkennen des Bezugspunktes ist **nicht** erlaubt.
- Das Referenzmaß-Offset (S-0-150 bzw. S-0-151) der Achsen ist korrekt zu übernehmen, da sonst danach kein eindeutiges "Maßbezug herstellen" durchführbar ist (Eine Überwachung seitens der Steuerung existiert nicht!). Ist bei der IB ein Fehler unterlaufen (z.B. Referenzmaß-Offset falsch), haben die Gantry-Achsen nach dem Referenzieren unterschiedliche Lageistwerte, aber dieser Versatz führt **nicht** zu einem gültigen Sollwert-Versatz.



Die Sollwerte des Masters wirken sofort und ohne Versatz auf die Slaves. Dabei kann es zum Sercos Fehler (exzessive Reglerabweichung oder exzessive Sollwertvorgabe) kommen.

- Der Referenzpunkt muss immer hinter dem Bezugspunkt (relevante Nullmarker) liegen, um zu verhindern, dass der Vorgang auf dem Referenzschalter startet (S-0-0400, Referenzschalter betätigt).
- Es ist zu beachten, dass die Referenzierrichtung für alle Gantryantriebe immer die gleiche sein muss, damit die Antriebe nach dem Erfassen der Referenzmarke zum Anfahren des Referenzpunktes nicht die Richtung umkehren. Dies wird erreicht, indem der Referenzschalter ausgewertet wird und ein entsprechendes Positionieren vor dem eigentlichen "Maßbezug herstellen" erfolgt.
- Da für die Inbetriebnahme der Gantry-Achsen IndraDrive Sercos Parameter (P-x-xxxx) verwendet werden, muss bei der Verwendung von anderen Antrieben überprüft werden, ob die entsprechende Funktionalität existiert und mit welcher Parameternummer sie zu aktivieren ist!

Aktivieren: Voraussetzung:

Alle Arbeiten zur IB der Gantry-Antriebe wurden durchgeführt.

Betrieb mit Gantry-Achsen:



Vor dem Starten von "Maßbezug herzustellen" muss die SPS alle Gantry-Achsen eines Verbands über das Interface-Signal "Vorschubsperrung" sperren.

1. Lassen Sie das System hochlaufen.
Nach dem Systemhochlauf ist die Gantrykopplung **aktiv**.
2. Für die Gantry-Achsen ist Maßbezug herzustellen
3. Ein Ausgleich eines im Betrieb oder im Ausschaltzustand entstandenen Istwertversatzes von Gantry-Achsen wird nur initiiert, wenn das Signal qAx_TakeActOffs ("Istwert-Versatz wird für diesen Slave als Sollwert-Versatz übernommen") gegeben wird.
Das nachfolgende "Herausinterpolieren" des Versatzes wird mit dem Signal "qAx_MasterPos" ("Gantry auf Masterposition") gestartet.

Deaktivieren: Der Gantry-Verband kann nicht per Programmierung deaktiviert werden.

Ein Abschalten der Gantry-Achsen erfolgt ausschließlich über den MP 1003 00061 (Index der Gantryleitachse). Jede Achse, die zu einem Gantry-Verband gehört, erhält den Parameter "0". Damit wird der Verband aufgelöst. Nach einem erneuten Hochlauf des Systems existiert der Gantry-Verband nicht mehr.



Ein Schräglagen-Ausgleich eines Gantryportals ist mit der oben beschriebenen Funktion im entkoppelten Zustand nicht möglich.

**optionale Sicherheitsmaßnahme
E-Stopp bei Gantryachsen**

Der "E-Stopp" ist eine hardwaremäßige Kopplung der Antriebsverstärker, welche einem Gantry-Verband angehören. Die hardwaremäßige Kopplung dient einer schnelleren Reaktion auf fatale Fehler einer Achse (Y oder Y2); um Schäden an der Mechanik zu vermeiden, müssen beide Achsen möglichst synchron und schnell abgebremst werden.

Antriebsverstärker: Hardwaremäßiges Verdrahten

Der **Ausgang X32:7** der Y-Achse wird auf den **Eingang X32:6** der Y2-Achse verdrahtet. Der **Ausgang X32:7** der Y2-Achse wird auf den **Eingang X32:6** der Y-Achse verdrahtet.

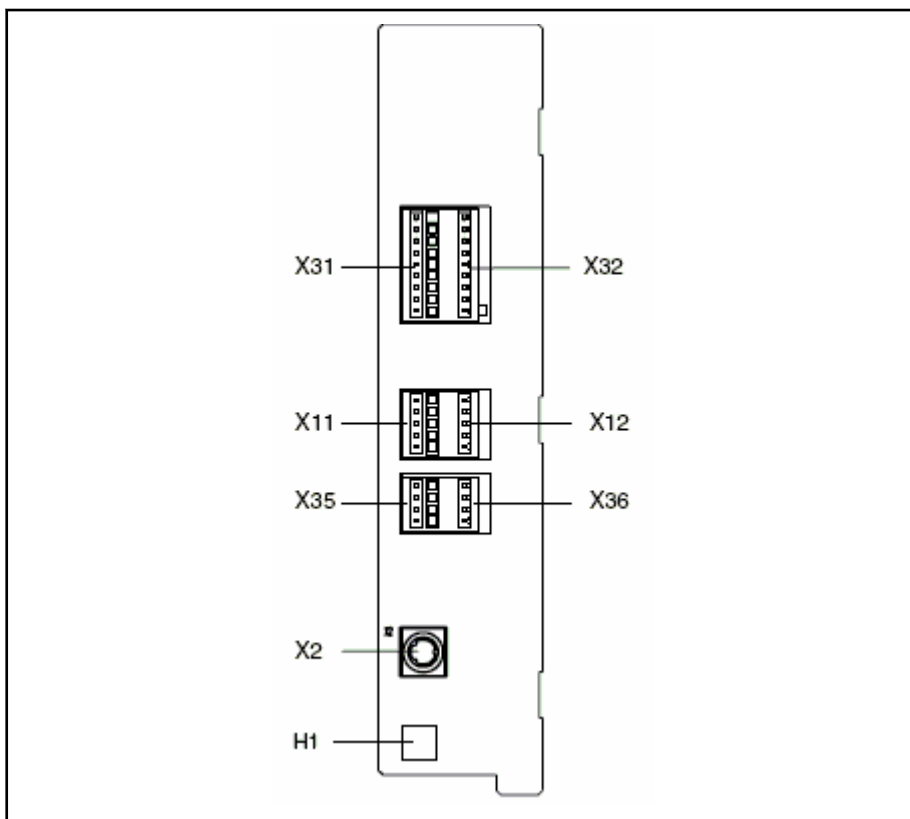


Abb. 14-9: Übersicht der Anschlussstecker

IW-Engineering / IW-Drive: Sercos Signale auflegen

E/A 7b muss als Ausgang parametrieren, diesem Ausgang wird der Sercos Parameter **P-0-0115** "Gerätesteuerung: Statuswort", **Bit 0** zugewiesen. **E/A 6b** wird als Eingang parametrieren und diesem der Sercos Parameter **P-0-0223** "E-Stopp-Eingang" das **Bit 0** zugewiesen. Damit wird das Signal, dass für das BB Relais benutzt wird, zum Auslösen des E-Stopp am jeweils anderen Regler genutzt.

		Example
Example:		Digitale Ein-/Ausgänge

IW-Engineering / IW-Drive: E-Stopp Funktion parametrieren

- Die Auswertung des E-Stopp Signals ist auf aktiv zu setzen.
- Der Interpretation des E-Stops ist als Warnung einzustellen.

		Example
Example:		E-Stopp Signal

IW-Engineering / IW-Drive: Fehlerreaktion parametrieren

Die Fehlerreaktion ist auf **"Sofort bestmögliches Stillsetzen"**, und die bestmögliche Stillsetzung auf **"Geschwindigkeits-Sollwert-Nullschaltung mit Rampe und Filter"** einzustellen. Die Verzögerung **"Schnell-Halt"** wird auf den maximalen Achsbeschleunigungswert gesetzt.

	Example	
Example:	Fehlerreaktion	Kapitelüberschrift

IW-Engineering / Configuration: Parameter editieren

Die Funktion "Elektronische Kopplung" muss über den Parameter **coup** "Elektronische Kopplung" im Setup (SUP) aktiviert werden, um folgende Parameter editieren zu können:

- **MaxFollErr**
"Maximaler Schleppabstand" (1003 00060)
- **MastInd**
"Achsgruppen-Index" (1003 00061)
- **MaxGrTroq**
"Max. Stillstands-Momentensumme der Gruppe" (1003 00062)
- **ENASynrMon**
"Synchronlauf-Überwachung ein" (1003 00063)
- **SynchrMonWin**
"Synchronlauf-Überwachungsfenster" (1003 00064)
- **ENRefSwiMon**
"Überwachung des Referenzschalters" (1003 00065)
- **CmdOffslpoMode**
"Gantry-Sollwertversatz ausgleichen" (1003 00066)
- **MaxRedCmdOffs**
"Obergrenze für Gantry-Sollwertversatz" (1003 00067)

	Documentation	Instruction
Instruction:		Editieren von Maschinenparametern
Documentation:	MTX Maschinenparameter	Gantry-Kopplung konfigurieren

IW-Operation / Programm: Sercos Dateien der Phase 2 anpassen

Für jeden Antrieb sind folgende Sercos Parameter einzustellen:

- **S-0-0041** "Referenzfahr-Geschwindigkeit"
- **S-0-0042** "Referenzfahr-Beschleunigung"
- **S-0-0115** "Lagegeberart 2" ist die verwendete Geberart einzustellen
- **S-0-0116/117** "Geber 1/2 Auflösung"
- **S-0-0052/54** "Referenzmaß 1/2":

In diesen Parameter wird der gewünschte Lageistwert bezüglich einer bestimmten Achsposition (Bezugsposition) eingetragen.

- **S-0-0147** "Referenzfahr-Parameter"

In diesem Parameter werden Festlegungen für die Kommandos:

- S-0-0146, C4300 Kommando "NC-geführtes Referenzieren"
- S-0-0148, C0600 Kommando "Antriebsgeführtes Referenzieren"
- S-0-0171, C4400 "Verschiebung berechnen"
- S-0-0172, C4500 "Verschiebung ins Referenzsystem"

bezogen auf die Anlagen-, NC- und Antriebsinstallation getroffen.

- **S-0-0150/151** "Referenzmaß-Offset 1/2"
bei Motor-Geber/externen Geber.
Wird der Referenzmaß-Offset (Sercos Parameter S-x-0150 bei Motor-Geber, S-x-0151 bei ext. Geber) der Masterachse verändert, muss für alle Slaves der Referenzmaß-Offset so angepasst werden, dass die Differenz der Referenzmaß-Offsets (zum Master!) konstant bleiben.
$$X_{\text{Referenzpunkt}} = X_{\text{Bezugspunkt}} + (S-0-0150 \text{ bzw. } S-0-0151)$$
- **S-0-0051/53** "Lageistwert Geber 1/2":
Zeigt die aktuelle Position des Motorgebers/ext. Gebers.
- **S-0-0177/178** "Absolutmaß-Offset" bei Motor-Geber.
Abstand vom Maschinen-Nullpunkt zum Nullpunkt des Motorgebers mit Absolutmessung.
- **S-0-0400** "Referenzschalter":
Schaltzustand des, am Regelgerät angeschlossenen Referenzschalters
- **S-0-0403** "Status Lageistwert":
In diesem Parameter werden die Statusmeldungen der Lageistwerte der angeschlossenen Geber angezeigt.
- **S-0-0189** "Schleppabstand"
für die Schleppabstands-Überwachung, falls im zyklischen Telegramm enthalten. Ist im Antrieb eine Betriebsart mit antriebsinterner Lageregelung aktiviert, wird die aktuelle Differenz zwischen Lagesollwert (vgl. **P-0-0434** "Lagesollwert Regler") und Lageistwert (vgl. **S-0-0051/S-0-0053** "Lageistwert Geber 1/2") angezeigt.
- **S-0-0084** "Drehmoment-/Kraft-Istwert"
In diesen Parameter den Drehmoment-Istwert für die Momentenüberwachung eingeben.
$$\text{Drehmoment-/Kraft-Istwert} = P-0-0043 \text{ "drehmomentbildender Strom, Istwert"} * \text{Drehmomentfaktor}$$
- **S-0-0016** "Konfigurationsliste Antriebstelegramm"
Aufnahme des Drehmoment-Istwertes **S-0-0084** und des Schleppabstandes **S-0-0189** in das zyklische Telegramm in den Sercos Dateien (p2*.scs) für die entsprechenden Gantry-Achsen.

	Documentation	Instruction
Instruction:		Editieren von SCS-Parametern
Documentation:	Rexroth IndraDrive Sercos Parameterbeschreibung	Gantry-Kopplung konfigurieren

IW-Operation / IndraLogic: Interface-Signale ggf. setzen

- Kanalinterface:
 - **qCh_OpModeSel_00** bis **qCh_OpModeSel_03** "Anwahl der Betriebsart"
(hier: Referenzpunkt anfahren bei inkrementelle Geber mit/ohne abstandskodiertem Messsystem)
 - **qCh_OpModePlc** "Betriebsartvorgabe von SPS"
(hier: Referenzpunkt anfahren inkrementelle Geber mit/ohne abstandskodiertem Mess-System)
- Achsinterface:

- **qAx_TakeActOffs** "Istwert-Versatz übernehmen":
Istwert-Versatz wird für diesen Slave als Sollwert-Versatz übernommen
- **qAx_MasterPos** "Gantry auf Masterposition":
Der ausgewählte Slave erhält den gleichen Sollwert wie der Master. Das Signal bewirkt keine Änderung in der Position des Slaves, wenn die Sollwerte von Master und Slave gleich sind. Wird das Signal zurückgesetzt, bevor der neue Sollwert aktiv ist, bleibt ein Sollwert-Offset erhalten (nur bei interpolatorischen Ausgleich möglich).
- **qAx_LagErrOff** "Kopplungsfehler ausblenden":
Jede Achse des Gantry-Verbands erhält dieses Eingangs-Signal, es ist jedoch nur für die Masterachse relevant. Ist dieses Signal für die Masterachse gesetzt, wird bei Überschreiten des Stillstandsmoments die Fehlerauswertung innerhalb der NC unterdrückt.
Low: Im Fehlerfall Fehlerbehandlung/Fehlermeldung durch die NC.
High: Im Fehlerfall keine Fehlerbehandlung/Fehlermeldung durch die NC.
- **qAx_TrqErrOff** "Stillstandsfehler ausblenden":
Jede Achse des Gantry-Verbands erhält dieses Eingangs-Signal, es ist jedoch nur für die Masterachse relevant. Ist dieses Signal für die Masterachse gesetzt, wird bei Überschreiten des Stillstandsmoments die Fehlerauswertung innerhalb der NC unterdrückt.
Low: Im Fehlerfall Fehlerbehandlung/ Fehlermeldung durch die NC.
High: Im Fehlerfall keine Fehlerbehandlung/Fehlermeldung durch die NC.
- **qAx_DrvLock** "Vorschub Sperre":
Vor dem Starten des Referenzierens muss die SPS alle Gantry-Achsen eines Verbandes sperren.
- **qAx_JogPlus/qAx_JogMinus** "Manuell +/-":
In der Betriebsart "Referenzpunkt anfahren" setzt die SPS die Tipp-Signale für alle Gantry-Achsen.



Das Achsinterface muss sowohl für die Masterachse als auch für die Gantry-Slaveachsen von der SPS verarbeitet werden (das Interface einer Gantry-Slaveachse verhält sich wie das Interface einer, durch Programmierung mit einer G-Funktion gekoppelten Slaveachse).

Die folgenden antriebsbezogenen Interface-Signale (NC-Eingänge) müssen innerhalb eines Koppelverbands immer identisch gehalten werden (Dies muss von der SPS gewährleistet werden!):

- Antrieb EIN
- Vorschub Sperre

	Documentation	Instruction
Instruction:		Editieren von SPS-Signalen
Documentation:	MTX SPS-Interface	Gantry-IF-Signale setzen

IW-Operation / IndraLogic: Interface-Signale zur Kontrolle

- Achsinterface:
 - **iAx_DrvAct** "Antrieb in Betrieb" (Regler geschlossen)
 - **iAx_MasterIndex0** "Index des Masters Bit 0" ... **iAx_MasterIndex6** "Index des Masters Bit 6"
 - **iAx_CmdOffsExst** "Sollwert-Versatz existiert" für diesen Slave (logisch1)
 - **iAx_CmdOffsExceed** "Ausgleichbarer Sollwert-Versatz" für diesen Slave überschritten (logisch1)
 - **iAx_TrpCmd** "Fahrbefehl" (Interpolatorische Verfahrbewegung aktiv)
 - **iAx_TrqExceed** "Stillstandsfehler":
Jede Achse des Gantry-Verbands erhält dieses Ausgangssignal. Es wird ständig aktualisiert, d.h. es bleibt nicht statisch anstehen, wenn ein Fehler aufgetreten war und wieder verschwunden ist!
 - **iAx_CoupleLag** "Nachlauffehler": Jede Achse des Gantry-Verbands erhält dieses Ausgangs-Signal. Es wird ständig aktualisiert, d.h. es bleibt nicht statisch anstehen, wenn ein Fehler aufgetreten war und wieder verschwunden ist! Im Gantry-Verband ist das Signal **iAx_CoupleLag** nur für die Masterachse relevant. Die Signale für die Slaveachsen werden nicht ausgewertet.
Low: Die Momentensumme aller Achsen liegt innerhalb des erlaubten Bereichs.
High: Die Momentensumme aller Achsen hat den erlaubten Bereich überschritten (**iAx_CoupleLag** ist unabhängig von **qAx_TrqErOff**).

	Documentation	Instruction
Instruction:		Editieren von SPS-Signalen
Documentation:	MTX SPS-Interface	Gantry-IF-Signale setzen

14.1.3 Handlungsanweisung: Gantry-Verband aktivieren

Diese Handlungsanweisung erklärt das Aktivieren eines Gantry-Verbands.

Drive: Achsinbetriebnahme

Alle Arbeiten zur Inbetriebnahme der Gantry-Antriebe wurden durchgeführt.



Alle Achsen des Gantry-Verbands müssen identische Parameter besitzen (Verstärkungsfaktoren, Beschleunigung/Verzögerung, Geschwindigkeit,...).

Die, für die Inbetriebnahme genannten herstellereigenen Sercos Parameter (P-x-xxxx) sind nur für IndraDrive-Antriebe relevant. Bei anderen Antrieben müssen äquivalente Parameter verwendet werden. Die Funktionalität muss vom Antriebshersteller sichergestellt sein.

Nach jedem Tausch von beteiligten Motoren, Messsystemen und/oder Antriebssoftware ist die Inbetriebnahme der Gantry-Achsen erneut durchzuführen.

IW-Engineering / Configuration: Ggf. Kopplung aktivieren

- Die Funktion Elektronische Kopplung muss über den Parameter **coup** "Elektronische Kopplung" im Setup (SUP) aktiviert sein

- In dem Parameter **MastInd** "Achsgruppen-Index" (1003 00061) ist die Achsnummer der Masterachse, bei allen dem Gantry-Verband angehörigen Achsen, auszutragen und ggf. für die Reaktivierung zu notieren.

	Documentation	Instruction
Instruction:		Editieren von Maschinenparametern
Documentation:	MTX Maschinenparameter	Gantry-Kopplung deaktivieren

E-Stopp applizieren

Der E-Stopp ist eine hardwaremäßige Kopplung der Antriebsverstärker, die einem Gantry-Verband angehören. Die hardwaremäßige Kopplung dient einer schnelleren Reaktion auf fatale Fehler.

		Instruction Kap. 14.1.2 "Handlungsanweisung: Gantry-Achsverband applizieren" auf Seite 498
Instruction:		E-Stopp applizieren

Gantry-Verband: ausrichten

Tippen Sie die einzelnen Achsen auf eine Position, auf der sie parallel zueinander stehen und eine ggf. noch vorhandene Schräglage durch das anschließende Referenzieren ausgeglichen werden kann.



Der Gantry-Verband muss sich in einem mechanisch momentenfrei ausgerichteten Zustand befinden.

Bei schweren Portal-Gantry-Verbänden müssen einzelne Gantry-Achsen mechanisch oder durch Tippen in eine parallele Lage zueinander gebracht werden.

IW-Engineering / Configuration: Kopplung aktivieren

- Die Funktion "Elektronische Kopplung" ist über den Parameter **coup** "Elektronische Kopplung" im Setup (SUP) aktiviert.
- In dem Parameter **MastInd** "Achsgruppen-Index" (1003 00061) ist die Achsnummer der Masterachse allen, dem Gantry-Verband angehörigen, Achsen zu zuweisen.



z.B. Masterachse ist Achse 5, somit ist der **MastInd** "Achsgruppen-Index" (1003 00061) = 5 . Der Achsgruppen-Index der Masterachse ist auch in den **MastInd** "Achsgruppen-Index" (1003 00061) aller dazugehörigen Achsen (Slaveachsen) einzutragen.

Folgende Interface-Signale müssen innerhalb eines Koppelverbandes immer identisch gehalten werden:

- **qAx_DrvLock** "Vorschub Sperre"
- **qAx_DrvOn** "Antrieb ein"

Die Gantry-Achsen müssen vor einem Steuerungshochlauf mechanisch momentenfrei (d.h. kein Klemmen) ausgerichtet sein, so dass eine parallele Verfahrbewegung der Achsen nach dem Steuerungs- und Sercos Hochlauf möglich ist.

	Documentation	Instruction
Instruction:		Editieren von Maschinenparametern
Documentation:	MTX Maschinenparameter	Gantry-Kopplung aktivieren

Drive: Maßbezug herstellen

- Vor dem Starten des "Maßbezug herstellen" muss die SPS alle Gantry-Achsen eines Verbands über das Interface-Signal **qAx_DrvLock** "Vorschubsperrung" sperren. (**qAx_DrvLock** = 1).
- Beim Steuerungs- bzw. Sercos Hochlauf wird Maßbezug hergestellt, d.h. die Lageistwerte jeder Achse sind sofort bekannt. Für jeden Antrieb ist "Maßbezug hergestellt", wenn die Bezugsmaße der Achsen (mit Absolutmaße setzen bei der Inbetriebnahme) korrekt übernommen wurden. Sind die Achsen ausgerichtet, wird durch Setzen identischer Lageistwerte der Maßbezug aller Gantry-Achsen gesetzt ("Absolutmaß setzen").



Der auf diese Maßbezüge basierende Vorgang zum "Maßbezug herstellen", während der Erst-Inbetriebnahme, sollte mit schwachem KV, niedriger Geschwindigkeit, niedriger Beschleunigung bzw. Verzögerung sowie einer Momentenüberwachung vollzogen werden.

IW-Engineering / IndraLogic: "Schräglage" ausgleichen

Der Ausgleich eines, im Betrieb oder im Ausschaltzustand entstandenen Istwertversatzes wird nur durch das Setzen von dem Interface-Signal **qAx_TakeActOffs** "Istwert-Versatz übernehmen" initiiert. Das nachfolgende "Rausinterpolieren" des Versatzes wird mit dem Setzen des Interface-Signals **qAx_MasterPos** "Gantry-Slave auf Masterposition" gestartet.



Der Gantry-Verband muss sich in einem mechanisch momentenfrei ausgerichteten Zustand befinden.

Bei schweren Portal-Gantry-Verbänden müssen einzelne Gantry-Achsen mechanisch oder durch Tippen in eine parallele Lage zueinander gebracht werden.

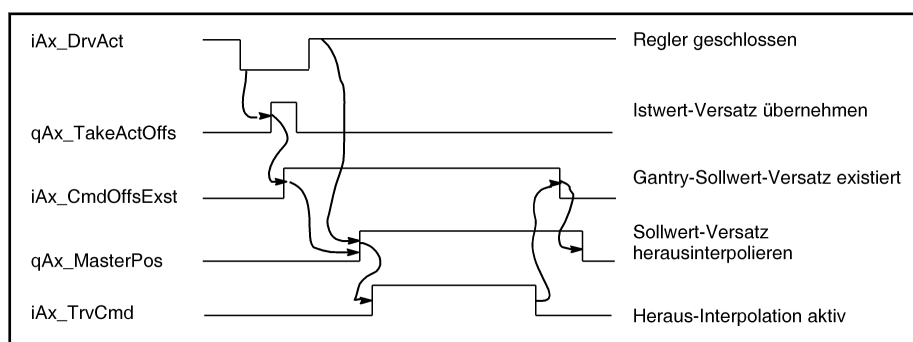


Abb. 14-10: Signalfussplan

	Documentation	Instruction
Instruction:		Editieren von SPS-Signalen
Documentation:	MTX SPS-Interface	Schräglage bei Gantry ausgleichen

IW-Engineering / Configuration: Überwachung parametrieren

- Die Funktion "Elektronische Kopplung" muss über den Parameter **coup** "Elektronische Kopplung" im Setup (SUP) aktiviert werden.
- **MaxFollErr** "Maximaler Schleppabstand" (1003 00060)
- **MastInd** "Achsgruppen-Index" (1003 00061)
- **MaxGrTroq** "Max. Stillstands-Momentensumme der Gruppe" (1003 00062)
- **ENASynrMon** "Synchronlauf-Überwachung ein" (1003 00063)
- **SynchrMonWin** "Synchronlauf-Überwachungsfenster" (1003 00064)
- **ENRefSwiMon** "Überwachung des Referenzschalters" (1003 00065)
- **CmdOffsIpoMode** "Gantry-Sollwertversatz ausgleichen" (1003 00066)
- **MaxRedCmdOffs** "Obergrenze für Gantry-Sollwertversatz" (1003 00067)

	Documentation	Instruction
Instruction:		Editieren von Maschinenparametern
Documentation:	MTX Maschinenparameter	Überwachung programmieren

14.1.4 Handlungsanweisung: Maßbezug bei einem absoluten Messsystem herstellen

Diese Handlungsanweisung erklärt das Herstellen eines Maßbezugs bei einem absoluten Messsystem

Drive: Maßbezug herstellen

- Wurde die Inbetriebnahme durchgeführt, so sind nach Steuerungs- bzw. Sercos Hochlauf die Lageistwerte jeder Achse sofort bekannt. Für jeden Antrieb ist "Maßbezug hergestellt", wenn die Bezugsmaße der Achsen (mit Absolutmaße setzen bei der Inbetriebnahme, ohne Reglerfreigabe) korrekt übernommen wurden. ("Absolutmaß setzen").

Drive: "Schräglage" ausgleichen

Sind die Lageistwerte der Gantry-Achsen nach dem Sercos Hochlauf im Gantry-Verband unterschiedlich (z.B. Driften bei offenem Regler), entspricht der Istwert-Versatz einer Schräglage zwischen den Gantry-Achsen. Der Versatz kann nun durch das Setzen des Interface-Signals **qAx_TakeActOffs** "Istwert-Versatz übernehmen" als Sollwert-Versatz übernommen werden. Mit dem Setzen des Interface-Signals **qAx_MasterPos** "Gantry-Slave auf Masterposition" wird der Istwert-Versatz durch asynchrone Interpolation durch die Steuerung ausgeglichen. Nach dem Versatzausgleich sind die Lageistwerte aller Gantry-Achsen wieder identisch.

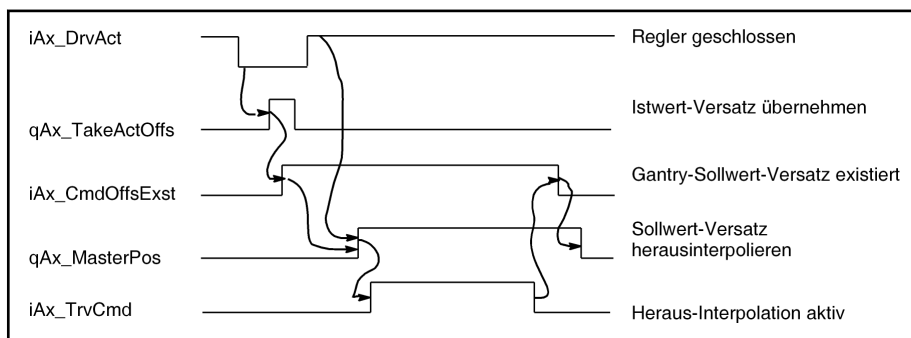


Abb. 14-11: Signalflussplan

	Documentation	Instruction
Instruction:		Editieren von SPS-Signalen
Documentation:	MTX SPS-Interface	Schräglage bei Gantry ausgleichen

Drive: Gleiche Antriebsparameter setzen

Alle Achsen des Gantry-Verbands müssen identische Parameter besitzen (Verstärkungsfaktoren, Beschleunigung/Verzögerung, Geschwindigkeit,...).

IW-Engineering / Configuration: Kopplung aktivieren

- Die Funktion "Elektronische Kopplung" ist über den Parameter **coup** "Elektronische Kopplung" im Setup (SUP) aktiviert.
- In dem Parameter **MastInd** "Achsgruppen-Index" (1003 00061) ist die Achsnummer der Masterachse, allen dem Gantry-Verband angehörigen Achsen zu zuweisen.



z.B. Masterachse ist Achse 5, somit ist der **MastInd** "Achsgruppen-Index" (1003 00061) = 5. Der Achsgruppen-Index der Masterachse ist auch in den **MastInd** "Achsgruppen-Index" (1003 00061) aller dazugehörigen Achsen (Slaveachsen) einzutragen.

Folgende Interface-Signale müssen innerhalb eines Koppelverbandes immer identisch gehalten werden:

- **qAx_DrvLock** "Vorschub Sperre"
- **qAx_DrvOn** "Antrieb ein"

Die Gantry-Achsen müssen vor einem Steuerungshochlauf mechanisch momentenfrei (d.h. kein Klemmen) ausgerichtet sein, so dass eine parallele Verfahrbewegung der Achsen nach dem Steuerungs- und Sercos Hochlauf möglich ist.

Nach jedem Tausch von beteiligten Motoren, Messsystemen und/oder Antriebssoftware ist die Inbetriebnahme der Gantry-Achsen erneut durchzuführen.

Die für die Inbetriebnahme genannten herstellerspezifischen Sercos Parameter (P-x-xxxx) sind nur für IndraDrive-Antriebe relevant. Bei anderen Antrieben müssen äquivalente Parameter verwendet werden. Die Funktionalität muss vom Antriebshersteller sichergestellt sein.

14.1.5 Handlungsanweisung: Gantry-Achsverband E-Stopp applizieren

Der "E-Stopp" ist eine hardwaremäßige Kopplung der Antriebsverstärker, welche einem Gantry-Verband angehören. Die hardwaremäßige Kopplung dient einer schnelleren Reaktion auf fatale Fehler einer Achse (Y oder Y2); um Schäden an der Mechanik zu vermeiden, müssen beide Achsen möglichst synchron und schnell abgebremst werden.

Antriebsverstärker: Hardwaremäßiges Verdrahten

Der **Ausgang X32:7** der Y-Achse wird auf den **Eingang X32:6** der Y2-Achse verdrahtet. Der **Ausgang X32:7** der Y2-Achse wird auf den **Eingang X32:6** der Y-Achse verdrahtet.

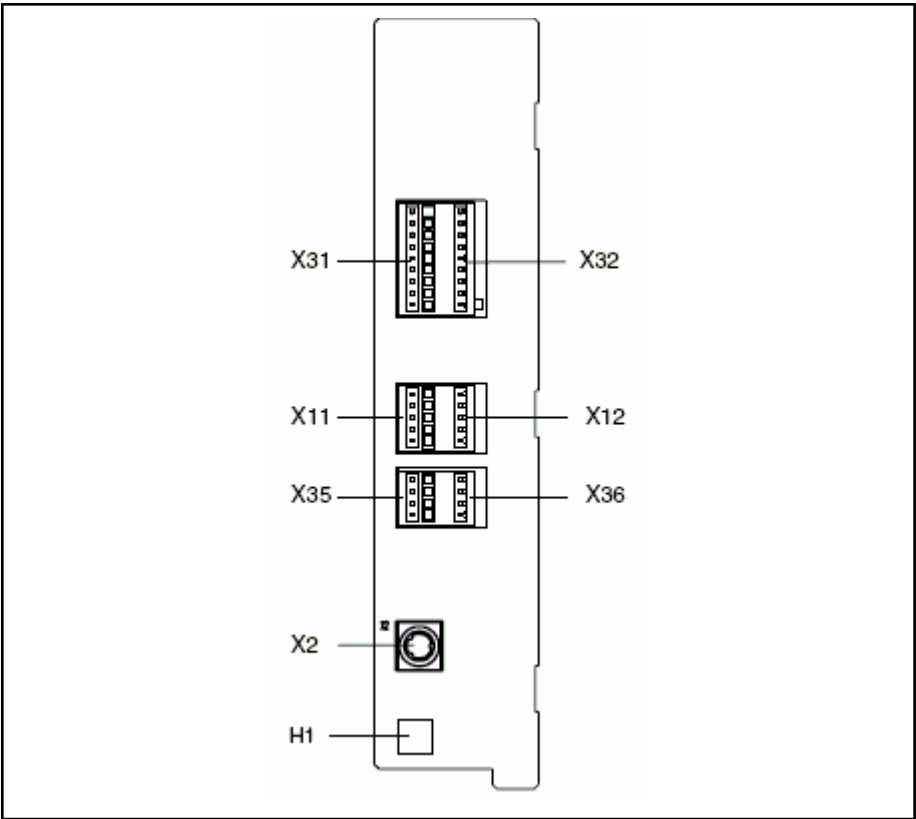


Abb. 14-12: Übersicht der Anschlussstecker

IW-Engineering / IW-Drive: Sercos Signale auflegen

E/A 7b muss als Ausgang parametrieren werden, diesem Ausgang wird der Sercos Parameter **P-0-0115** "Gerätesteuerung: Statuswort", **Bit 0** zugewiesen. **E/A 6b** wird als Eingang parametrieren und diesem der Sercos Parameter **P-0-0223** "E-Stopp-Eingang" das **Bit 0** zugewiesen. Damit wird das Signal, dass für das BB Relais benutzt wird, zum Auslösen des E-Stopp am jeweils anderen Regler genutzt.

		Example
Example:		Digitale Ein-/Ausgänge

IW-Engineering / IW-Drive: E-Stopp Funktion parametrieren

- Die Auswertung des E-Stopp Signals ist auf aktiv zu setzen.
- Der Interpretation des E-Stops ist als Warnung einzustellen.

		Example
Example:		E-Stopp Signal

IW-Engineering / IW-Drive: Fehlerreaktion parametrieren

Die Fehlerreaktion ist auf **"Sofort bestmögliches Stillsetzen"**, und die bestmögliche Stillsetzung auf **"Geschwindigkeits-Sollwert-Nullschaltung mit Rampe und Filter"** einzustellen. Die Verzögerung **"Schnell-Halt"** wird auf den maximalen Achsbeschleunigungswert gesetzt.

	Example	
Example:	Fehlerreaktion	Kapitelüberschrift

14.2 Selektive additive Koordinatenkopplung

14.2.1 Beschreibung

Funktion

Eine oder mehrere Werkstückkoordinaten eines Kanals (Quelle) können Werkstückkoordinaten anderer Kanäle (Ziele) additiv beeinflussen. Sofern im Quellkanal eine Verfahrbewegung stattfindet, lassen sich auf diese Art die Bahnbewegungen in den einzelnen Zielkanälen mit der des Quellkanals überlagern.

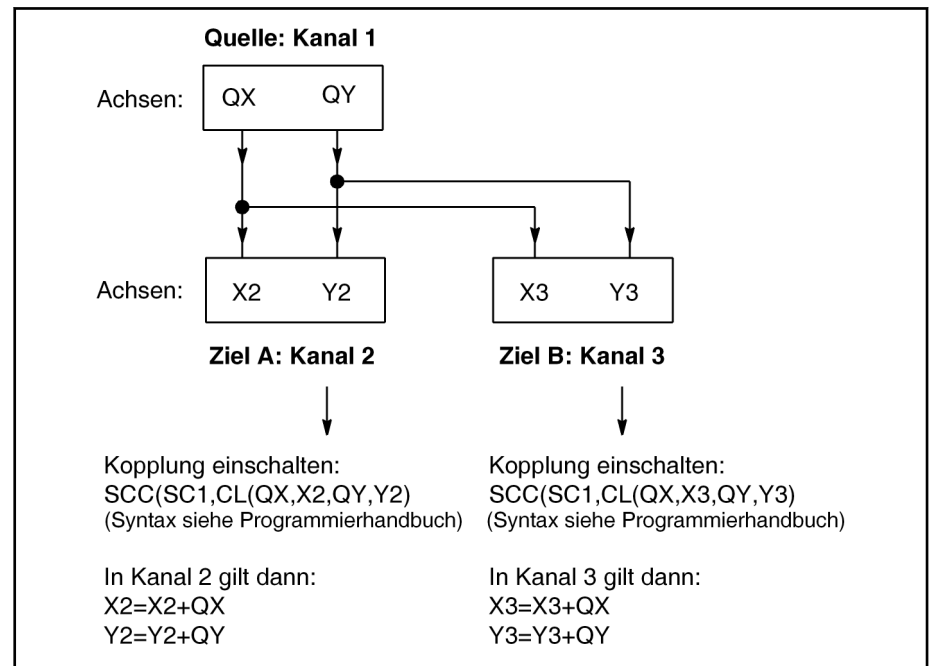


Abb. 14-13: Quellkanal

Restriktionen und Besonderheiten

- Ein Quellkanal muss eine niedrigere Kanalnummer als der Zielkanal besitzen.
- Sowohl der Quellkanal, als auch dessen Koordinaten können gleichzeitig mehreren Zielkanälen zugeordnet sein.
Ein Zielkanal darf aber zu einem Zeitpunkt nur mit einem einzigen Quellkanal verkoppelt sein.
- Beim Herstellen und beim Beenden einer Kopplung ($SCC(..)$) darf keine Veränderung der Quellkoordinaten stattfinden.
- Durch die Überlagerung zweier Bahnbewegungen kann die max. erlaubte(n) Bahngeschwindigkeit, Bahnbeschleunigung und Achsgeschwindigkeiten im Zielkanal überschritten werden.
- Für die überlagerte Bewegung wird keine Endschalterprüfung durchgeführt.
- Beim Ein-, Aus- oder Umschalten einer Koordinatentransformation/Achstransformation im Quell- oder Zielkanal wird die Koordinatenkopplung beendet und im Zielkanal ein Laufzeitfehler ausgegeben.

*Beispiel:***Selektive additive Koordinatenkopplung - Restriktion**

Eine Maschine hat zwei rechtwinklig zueinander angeordnete Bearbeitungsachsen, deren Bewegungen durch die Überlagerung zweier Teileprogramme (siehe unten) erzeugt werden soll.

Die Maschine wird mit zwei virtuellen (VX,VY) und zwei realen Achsen(X,Y) konfiguriert. Die virtuellen Achsen werden dem 1. Kanal, die realen Achsen dem 2. Kanal zugeordnet. Achs- oder Koordinatentransformationen sind nicht aktiv.

Erforderliche Maschinenparameter:

		Kanal 2		Kanal 1	
100100001	"Antriebsfunktionstyp":	1,	1,	1,	1
100100010	"Virtueller Antrieb":	0,	0,	1,	1
100300001	"Physikalische Achsbezeichnung":	X,	Y,	VX,	VY
100300002	"Kanalzuordnung":	2,	2,	1,	1

Erforderliche Teileprogramme:

Das Teileprogramm in Kanal 2 soll den Ablauf steuern.

Teileprogramm von Kanal 1 (Quellkanal) "TEST1":

```

10      VAP (-VX[@AXPOS], VY[@AYPOS] ;Achsposition laden
      POS])
20      BITIF("ICH_CUSTOM1") = 1 ;CPL-Kundenausgang 1 setzen
N20     WPV[@STARTALL=1] ;Warten bis die Bewegung beginnen
      darf
N20     G1 G91 F1000 VY50
20      BITIF("ICH_CUSTOM1") = 0
M30

```

Teileprogramm von Kanal 2 (Zielkanal):

```

10      @AXPOS=ACS( "X") ;Achsposition abspeichern
      @AYPOS=ACS( "Y")
      @STARTALL=0
21      WAIT
10      ERR=MCOPS(7,1,5) ;Betriebsart Automatik in Kanal 1 an-
      wählen
20      IF ERR <> 0 THEN
      <Fehlerroutine>
      ENDIF
30      WAIT
40      ERR=MCOPS(4,1, "TEST1",,, ;Programm TEST1 in Kanal 1 anwählen
      67)
50      IF ERR <> 0 THEN
      <Fehlerroutine>
      ENDIF

```



```

60      WAIT                                     ;Warten bis Kanal 1 NC-Bereit meldet
        (BITIF( "ICH_NCREADY",1))
70      ERR=MCOPS(6,1)                           ;Teileprogramm in Kanal 1 starten
80      IF ERR <> 0 THEN
        <Fehlerroutine>
        ENDIF
90      WAIT                                     ;Warten bis Kanal 1 den CPL-Kunden-
        (BITIF( "ICH_CUSTOM1",1))                ausgang 1 gesetzt hat
N10     SCC(SC1,CL(VX,X,VY,Y))                   ;Koordinaten VX und VY ankoppeln
N20     SPV[@STARTALL = 1]                       ;Bewegung starten
N30     G1 G91 F1000 X50
N40     SPV[@STARTALL = 0]                       ;Variable zurücksetzen
N50     SCC(0)                                    ;Kopplung aufheben
M30

```

Relevante NC-Funktionen

SCC(..)	Koordinaten-Kopplung herstellen/aufheben

Tab. 14-4: Relevante NC-Funktionen

Relevante Maschinenparameter (MP)

100100001	Antriebsfunktionstyp
100100010	virtueller Antrieb
100300001	physikalische Achsbezeichnung
100300002	Kanalzuordnung
101000001	Maximale Beschleunigung im Vorschubbetrieb
101000003	Maximale Beschleunigung im Eilgangbetrieb
703000210 und 703000220	Bahnbeschleunigung
706000020	Einschaltzustand nach Grundstellung

Tab. 14-5: Relevante Maschinenparameter

14.2.2 Handlungsanweisung: Selektive additive Koordinatenkopplung

Applizieren

Eine oder mehrere Werkstückkoordinaten eines Kanals (Quelle) können Werkstückkoordinaten anderer Kanäle (Ziele) additiv beeinflussen. Sofern im Quellkanal eine Verfahrbewegung stattfindet, lassen sich auf diese Art die Bahnbewegungen in den einzelnen Zielkanälen mit der des Quellkanals überlagern.

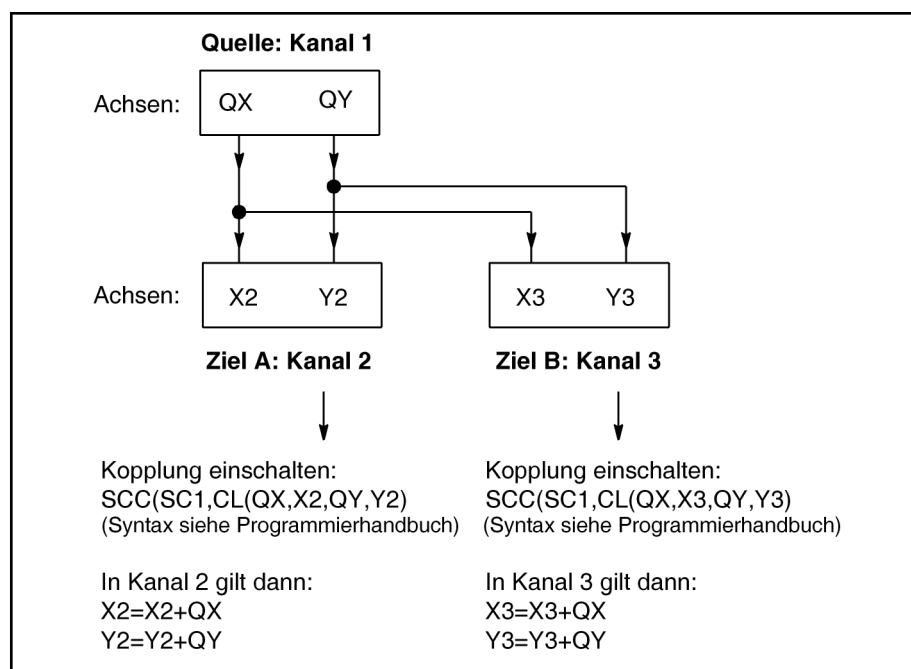


Abb. 14-14: Beispiel Übersicht: Selektive additive Koordinatenkopplung

IW-Engineering / Configuration: Parameter editieren

Die vorzunehmenden Einstellungen in den folgenden Parametern sind vom konkreten Anwendungsfall abhängig:

- **EnabVirtMode** "Virtueller Antrieb" (1001 00010)
- **RefPos** "Position des Referenzpunkts" (1015 00110)
- **SysDrName** "Physikalische Antriebsbezeichnung" (1003 00001):
den Systemachsennamen anpassen.
- **DefaultCh** "Kanalzuordnung" (1003 00002):
Achse dem gewünschten Kanal zuordnen.
- **FeedAxAcc** "Maximale Beschleunigung im Vorschubbetrieb" (1010 00001):
Festlegen der maximalen Achsbeschleunigung im Vorschubbetrieb.
- **RapidAxAcc** "Maximale Beschleunigung im Eilgangbetrieb" (1010 00003):
Festlegen der maximalen Achsbeschleunigung im Eilgangbetrieb.
- **MaxChVel** "Maximale Bahngeschwindigkeit" (7030 00120):
- **MaxChAcc** "Maximale Bahnbeschleunigung" (7030 00210):
- **MaxChDec** "Maximale Bahnbremsbeschleunigung" (70330 00220):
- **ChResetState** "Zustand nach Kanal-Grundstellung" (7060 00020): Falls die Funktion nach **M2/M30, Kanalgrundstellung** oder **Systemgrundstellung** deaktiviert werden soll, muss in den Parameter **ChResetState** an geeigneter Stelle die Funktion **SCC()** eingetragen werden.

	Documentation	Instruction
Instruction:		Editieren von Maschinenparametern
Documentation:	MTX Maschinenparameter	Koordinatenkopplung

Aktivieren/Deaktivieren **IW-Operation / NC-Programmierung: Koordinatenkopplung aktivieren**
SelCrdCouple(SC<Kanal>,CL(<Q1>,<Z1>{{,<Qn>,<Zn>}...})) bzw.
SCC(SC<Kanal>,CL(<Q1>,<Z1>{{,<Qn>,<Zn>}...})) programmieren



Ein Quellkanal muss eine niedrigere Kanalnummer als der Zielkanal besitzen.

Sowohl der Quellkanal, als auch dessen Koordinaten können gleichzeitig mehreren Zielkanälen zugeordnet sein. Ein Zielkanal darf aber zu einem Zeitpunkt nur mit einem einzigen Quellkanal verkoppelt sein.

Beim Herstellen und beim Beenden einer Kopplung (SCC(..)) darf keine Bewegung in den Quellkoordinaten stattfinden.

Durch die Überlagerung zweier Bahnbewegungen kann die max. erlaubte Bahngeschwindigkeit, Bahnbeschleunigung und Achsgeschwindigkeit im Zielkanal überschritten werden.

Für die überlagerte Bewegung wird keine Endschalterprüfung durchgeführt.

Beim Ein-, Aus- oder Umschalten einer Koordinatentransformation/Achstransformation im Quell- oder Zielkanal wird die Koordinatenkopplung beendet und im Zielkanal ein Laufzeitfehler ausgegeben.

IW-Operation / NC-Programmierung: Koordinatenkopplung deaktivieren

SelCrdCouple() bzw. **SCC()** programmieren.

Wenn die Kopplung nach **M2/M30**, **Kanalgrundstellung** oder **Systemgrundstellung** deaktiviert werden soll, dann muss im Parameter **ChResetState** an geeigneter Stelle die Funktion **SCC()** eingetragen werden.

	Documentation	Instruction
Documentation:	MTX Programmierhandbuch	Koordinatenkopplung aktivieren/deaktivieren

14.3 Selektive additive Koordinatenkopplung mit Tabellen

14.3.1 Beschreibung

Funktion

Diese Funktion erweitert die Möglichkeit von SelCoordCouple/SCC.

Die Erweiterungen sind:

- Ankoppeln von zwei Quellkoordinaten an eine Zielkoordinate.
- Werkstück- und Maschinenkoordinaten können als Quelle verwendet werden.
- Die Kopplungsbeziehung kann über einen Faktor und eine Tabelle angepasst werden.
- Die Zielkoordinate kann mit einem Wichtungsfaktor versehen werden.

Diese Funktion ermöglicht die additive Korrektur der Werkzeugposition (WCS) eines Kanals durch Ankopplung an die Werkstück- oder Maschinenposition(en) eines anderen Kanals. Es besteht die Möglichkeit, eine oder mehrere Koordinaten anzukoppeln. Beim Ankoppeln wird keine Verfahrbewegung ausgeführt.

Kopplungen

Einige Regeln sind bei der Ankopplung zu beachten:

- Ein Kanal kann sich nur an die Koordinaten eines Kanals mit einer niedrigeren Kanalnummer ankopplern.
- Beim Herstellen einer Kopplung muss sichergestellt sein, dass im Quellkanal keine Bewegung stattfindet.
- Für maximal 4 Zielkoordinaten können Tabellenkopplungen eingeschaltet werden.
- Eine Koordinate, die bereits Zielkoordinate einer bestehenden Koordinatenkopplung ist, darf nicht zusätzlich zur Zielkoordinate einer anderen Koordinatenkopplung gemacht werden. Die bestehende Koordinatenkopplung muss also in diesem Fall zuerst deaktiviert werden.
- Eine Koordinate, die bereits Quellkoordinate in einer Koordinatenkopplung ist, darf zusätzlich auch in einer anderen Koordinatenkopplung zur Quellkoordinate gemacht werden.

Bei aktiver Kopplung ist folgendes zu beachten:

- Die Bewegung der Koordinaten und Achsen des Zielkanals ergibt sich aus der Überlagerung zweier Bewegungen. Dieses muss bei der Wahl der Bahngeschwindigkeit (F), der Bahnbeschleunigung (PathAcc, MP 703000210 und 703000220) und der Achsbeschleunigung (AxAcc, MP 101000001) berücksichtigt werden.
- Beim Ein-, Aus- oder Umschalten einer Koordinatentransformation ("Schiefe Ebene") im Zielkanal wird die Koordinatenkopplung beendet und im Zielkanal wird ein Laufzeitfehler ausgegeben.
- Beim Ein-, Aus- oder Umschalten einer Koordinatentransformation ("Schiefe Ebene") im Quellkanal wird, falls eine Werkstückkoordinate verwendet wird, die Koordinatenkopplung beendet und im Zielkanal ein Laufzeitfehler ausgegeben.
Beim Ein-, Aus- oder Umschalten einer Achstransformation 1 im Quell- oder Zielkanal wird die Koordinatenkopplung beendet und im Zielkanal ein Laufzeitfehler ausgegeben.
- Beim Ein-, Aus- oder Umschalten einer Achstransformation 2 im Zielkanal wird die Koordinatenkopplung beendet und im Zielkanal ein Laufzeitfehler ausgegeben.
- Beim Ein-, Aus- oder Umschalten einer Achstransformation 2 im Quellkanal wird, falls eine Werkstückkoordinate verwendet wird, die Koordinatenkopplung beendet und im Zielkanal ein Laufzeitfehler ausgegeben.

Beim Beenden der Kopplung ist zu beachten, dass im Quellkanal keine Bewegung der Quellkoordinaten stattfindet.

Mögliche Kopplungsbeziehungen bei Verwendung einer Quellkoordinate sind:

$$P_{De} = P_{De}^0 + (P_{De}^{prog} - P_{De}^0) * \text{Zielfaktor} + (P_{So} - P_{So}^0) * \text{Faktor}$$

$$P_{De} = P_{De}^0 + (P_{De}^{prog} - P_{De}^0) * \text{Zielfaktor} + (\text{Tab} (P_{So} - \text{TabOffs}) - \text{Tab} (P_{So}^0 - \text{TabOffs})) * \text{Faktor}$$

Mit:

P_{De} = aktuelle Position der Zielkoordinate

P_{De}^0 = Position der Zielkoordinate beim Herstellen der Kopplung

P_{De}^{prog} = aktuelle Position der Zielkoordinate,
die sich ohne Kopplung ergeben würde

Zielfaktor = Wichtungsfaktor für die Zielkoordinate (Default = 1.0)

P_{So}	= aktuelle Position der Quellkoordinate
P_{So}^0	= Position der Quellkoordinate beim Herstellen der Kopplung
Faktor	= Kopplungsfaktor (Default = 1.0)
Tab	= tabellierte Funktion
TabOffs	= Offset für den Tabellenzugriff (Default = 0.0)

Bei Verwendung zweier Quellkoordinaten kommt ein zweiter analoger Summand für die zweite Quellkoordinate hinzu, d. h.:

$$\dots + (P_{So2} - P_{So2}^0) * \text{Faktor2}$$

oder:

$$\dots + (\text{Tab} (P_{So2} - \text{TabOffs2}) - \text{Tab} (P_{So2}^0 - \text{TabOffs2})) * \text{Faktor2}$$

Relevante NC-Funktionen (G-Codes)

Herstellen einer Kopplung:

SelCrdCoupleTab(DE(<Name>,<Zielfakt>),SO(<Kanal>,<Name>,<Fakt>,<Tab>,<Offs>),WCS|MCS)
{ ,SO2(<Kanal>,<Name>,<Fakt>,<Tab>,<Offs>),WCS|MCS})

Die Kurzform lautet: **SCCT**

Relevante NC-Funktionen (G-Codes) - Herstellen einer Kopplung

Mit den Parametern:

DE (Destination)

<Name> Name der Zielkoordinate

<Zielfakt> Wichtungsfaktor der Zielkoordinate (1.0 falls nicht programmiert)

SO (Source)

<Kanal> Nummer des Quellkanals

<Name> Name der Quellkoordinate

<Fakt> Kopplungsfaktor (1.0 falls nicht programmiert)

<Tab> Name der Kopplungstabelle

<Offs> Offsetwert für den Tabellenzugriff (0.0 falls nicht programmiert)

WCS|MCS Koordinatensystem der Quellkoordinate

WCS = Werkstückkoordinate

MCS = Maschinenkoordinate

Default: WCS

SO2 (Source 2) analog zu SO (Source)

Beispiele: Herstellen einer Kopplung

SCCT(DE(YB), SO(1,YA))

Die tatsächliche YB-Position ergibt sich als Summe aus der durch die Programmierung im aktiven Kanal vorgegebenen YB-Position und der YA-Position.

SCCT(DE(YB,0), SO(1,YA))

Die tatsächliche YB-Position ist gegeben durch die YA-Position. Die durch die Programmierung im aktiven Kanal vorgegebene YB-Position bleibt ohne Auswirkung.

SCCT(DE(YB), SO(1,YA, ,SCCTab.fct))

Koppelt die Werkstückkoordinate YA des Kanals 1 über die Kopplungstabelle SCCTab.fct an die Werkstückkoordinate YB des aktiven Kanals.

SCCT(DE(YB), SO(1,YM,0.3, ,MCS))

Koppelt die Maschinenkoordinate YM des Kanals 1 über den Kopplungsfaktor 0.3 an die Werkstückkoordinate YB des aktiven Kanals.

Kopplungen

Erzeugen einer Spline-Tabellendatei:

Aufheben einer Kopplung:

SelCrdCouple() oder SelCrdCoupleTab() oder SCC() oder SCCT()

Eine Spline-Tabellendatei kann unabhängig von der Herstellung einer Koordinatenkopplung mit

CoupleSplineTab(TAB(<Tab>,<1|0>))

erstellt werden.

Die Kurzform lautet: **CST**

Mit den Parametern:

TAB <Tab> Name der Kopplungstabelle

Beispiel:

CST(TAB(curve.fct)) Erstellt, falls notwendig, die Splinetabelle /
 <Linkverzeichnis>/curve.fct.s.

Ein-/Ausschaltverhalten, Grundstellung

Steuerungshochlauf: Diese Funktion ist inaktiv.

Grundstellung Kanalgrundstellung und Systemgrundstellung haben keine direkte Wirkung auf diese Funktion. Soll bei Kanalgrundstellung oder Systemgrundstellung diese Funktion deaktiviert werden, muss in MP 706000020 "Einschaltzustand nach Grundstellung" an geeigneter Stelle (vor #Reset:, hinter #Reset: oder hinter #SysRes:) SCCT() oder SCC() eingetragen werden.

M30 / M2: Soll M2/M30 die Funktion abschalten, muss in MP 706000020 "Einschaltzustand nach Grundstellung" vor #Reset: SCC() oder SCCT() eingetragen werden.

Beispiel:

Eine Maschine mit zwei Bearbeitungsachsen deren Bewegung durch Überlagerung zweier Teileprogramme erzeugt wird. Die beiden Achsen seien rechtwinklig angeordnet. Konfiguriert wird die Maschine mit zwei virtuellen und zwei realen Achsen. Die beiden virtuellen Achsen werden dem ersten Kanal und beiden realen Achsen dem zweiten Kanal zugeordnet.

MP 100100001	"Antriebsfunktionstyp"	1,	1,	1,	1
MP 100100010	"Virtueller Antrieb"	0,	0,	1,	1
MP 100300001	"Physikalische Achsbezeichnung"	X,	Y,	VX,	VY
MP 100300002	"Kanalzuordnung"	2,	2,	1,	1

Es ist keine Achstransformation und keine Koordinatentransformation aktiv. Das Teileprogramm in Kanal 2 steuert den Ablauf.

Teileprogramm von Kanal 1 (Quellkanal) "TEST1":

```

10 VAP VX[@MXPOS] VY[@MYPOS]           ;Achsposition laden
20 BITIF("ICH_CPL01") = 1               ;CPL-Kundenausgang1 setzen
N20 WPV[@STARTALL=1]                   ;Warten bis die Bewegung beginnen darf

N20 G1 G91 F1000 VY50

```

```
20 BITIF("ICH_CPL01") = 0
```

```
M30
```

Teileprogramm von Kanal 2:

```
10 @MXPOS=ACS("X") ;Achspositionen abspeichern
11 @MYPOS=ACS("Y")

20 @STARTALL=0
21 WAIT

10 ERR=MCOPS(7,1,5) ;Betriebsart Automatik in Kanal 1 anwählen
20 IF ERR <> 0 THEN Fehlerbehandlung ENDIF
30 WAIT
40 ERR=MCOPS(4,1,"TEST1", , , 67) ;Programm TEST1 in Kanal 1 anwählen
50 IF ERR <> 0 THEN Fehlerbehandlung ENDIF
60 WAIT(BITIF("ICH_NCREADY",1)) ;Warten bis Kanal 1 NC-Bereit meldet
70 ERR=MCOPS(6,1) ;Teileprogramm in Kanal 1 starten
80 IF ERR <> 0 THEN Fehlerbehandlung ENDIF
90 WAIT(BITIF("ICH_CPL01",1)) ;Warten bis Kanal1 CPL-Kundenausgang1 gesetzt hat

N10 SCCT(DE(X),SO(1,VX)) ;Koordinaten VX ankoppeln
N11 SCCT(DE(Y),SO(1,VY)) ;Koordinaten VY ankoppeln

N20 SPV[@STARTALL = 1] ;Bewegung starten
N30 G1 G91 F1000 X50
N40 SPV[@STARTALL = 0] ;Variable zurücksetzen
N50 SCCT() ;Kopplung aufheben

M30
```

Kompatibilität

Die Funktion "SelCrdCoupleTab" ergänzt die bisherige Funktion "SelCrdCouple".

14.3.2 Aktivieren

Durch Programmieren der NC-Funktion "SelCrdCoupleTab (...)" wird die Funktion aktiviert.

14.3.3 Deaktivieren

Die Funktion "SelCrdCoupleTab() / SCCT()" beendet die Kopplung.

14.4 Kopplungstabellen

14.4.1 Beschreibung

Funktion

Die Kopplungstabellen werden für die Koordinatenkopplung (SCCT), die Achskopplung (AXC) und die Systemachskopplung verwendet.

Eine Kopplungstabelle beinhaltet die Kopplungsbeziehung zwischen zwei Positionen. In Form von Stützpunktpaaren (p_{mi} , f_i) ($i=1,\dots,n$) wird die Kopplungsfunktion tabellarisch definiert.

Aus den Stützpunktpaaren errechnet der NC-Interpolator die Funktionswerte zwischen den Stützpunkten.

Die Berechnung der Positionen zwischen den Stützpunkten kann mit folgenden Approximationen (Näherungen) gewählt werden:

- linear – als Strecke zwischen zwei Stützpunkten
- kubischer Spline – als Spline-Kurve zwischen zwei Stützpunkten unter Berücksichtigung der vorausgehenden und nachfolgenden Stützpunkte.

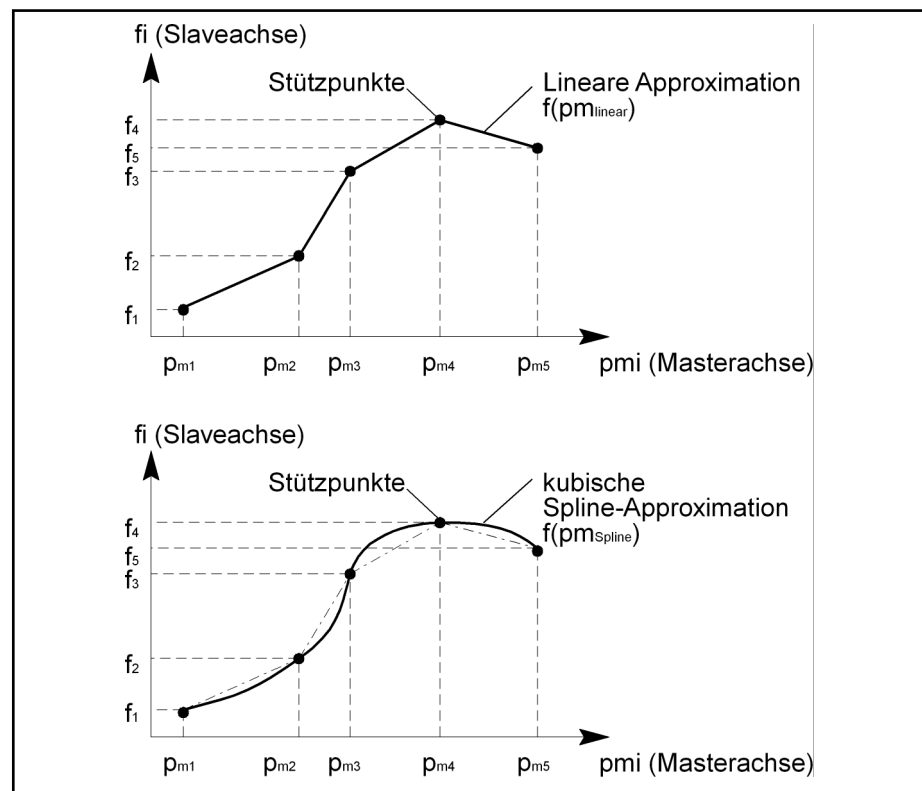


Abb. 14-15: Lineare Approximation - kubische Spline-Approximation

Die Spline-Approximation wird vorgezogen, wenn zwischen Stützpunkten ein kurviger Verlauf erwünscht ist und keine Daten über den exakten Verlauf vorliegen. Die kubische Spline-Näherung ermöglicht einen Kurvenverlauf, der weiche Übergänge an den Stützpunkten erzeugt.

Syntax der Kopplungstabelle

Die Kopplungstabellen haben folgenden Aufbau:

```
#1      <Interpolationsart der Stützpunkte>
        ; 1 linear, 3 kubischer Spline
```


#11	<Einheit pmi-Werte> ; -3 mm, -2 cm, -1 dm, 0 m, 1 Inch, 2 grad, 3 rad
#12	<Einheit der fi-Werte> ; wie bei #11
#20	<Periodisch> ; 0 Nichtperiodisch, 1 Periodisch
#32	<Leitachsgeschwindigkeitslimitierung> ; 1 vollständige Limitierung ; 0 unvollständige Limitierung
#100	<pm1> <f₁> ; 1. Stützpunktpaar
#100	<pm2> <f₂> ; 2. Stützpunktpaar
.	
#100	<p_{mn}> <f_n> ; n. Stützpunktpaar
#1	legt fest, auf welche Weise zwischen den Stützpunkten interpoliert wird. Möglich ist eine lineare Interpolation (Wert = 1) oder eine kubische Spline-Interpolation (Wert = 3). Der voreingestellte Wert ist 1, also Linearinterpolation. Der Wert 1 darf für die Kanalachskopplung (AxCouple, AXC) nicht verwendet werden.
#11	gibt die Einheit der pm-Werte an. Bei Längeneinheiten (Werte -3 bis +1) darf die Tabelle nur für lineare Führungsachsen verwendet werden. Entsprechend darf sie nur für rotatorische Führungsachsen verwendet werden bei Winkелеinheiten (Werte 2 und 3). Der voreingestellte Wert ist -3 (mm).
#12	gibt die Einheit der f-Werte an (Einheiten wie bei #11). Für lineare und rotatorische Folgeachsen dürfen nur jeweils die Einheiten -3 bis +1 bzw. 2 und 3 verwendet werden. Der voreingestellte Wert ist -3 (mm).
#20	gibt an, ob die Kopplungsfunktion periodisch ist oder nicht. Bei einer periodischen Kopplungsfunktion (Wert = 1) definiert der letzte p_m-Wert die Periode. Überschreitet die Position der Führungsachse die Periode, so wird zur Bestimmung des Funktionswertes f(p_m) pm in das Periodenintervall per Modulo-rechnung zurückgerechnet. Bei einer nichtperiodischen Kopplungsfunktion ist die Modulo-rechnung abgeschaltet. Der voreingestellte Wert ist 0, also nichtperiodisch. <i>Folgende Regeln sind zu beachten:</i> • #20 = 0, nichtperiodisch:

Der Endschalterbereich der Führungsachse wird auf das Intervall $[p_{m1}, p_{mn}]$ eingeschränkt.

Moduloachsen (lineare Moduloachse oder Endlosachse) sind nicht als Führungsachsen erlaubt.

- #20 = 1, periodisch:

Die p_m -Werte müssen bei 0 beginnen, d. h. $p_{m1} = 0$. Der letzte p_m -Wert definiert die Periode.

Die f -Werte des ersten und letzten Stützpunktpaares müssen gleich sein, d. h., $f(p_{m1}) = f(p_{mn})$.

Ist die Führungsachse eine Moduloachse, muss $AxModVal \bmod \text{Periode} = 0$ sein, wobei $AxModVal$ der achsspezifische Modulowert ist (Antriebsparameter).

#32 Dieser Parameter wird nicht mehr unterstützt!

#100 legt ein Stützpunktpaar fest.

In die Tabelle können beliebig viele Stützpunkte geschrieben werden.

Die p_m -Werte müssen in aufsteigender Reihenfolge abliegen.

Das Kommentarzeichen in der Tabelle ist das Semikolon.

Approximation von $f(p_m)$ zwischen den Stützpunkten

Die Funktion $f(p_m)$ wird zwischen jeweils zwei Stützpunkten $p_{m,i}$ und $p_{m,i+1}$ durch ein Polynom approximiert:

$$f_i(p_m) = \sum_{j=0}^3 c_{ij} \cdot (p_m - p_{m,i})^j \quad p_{m,i} \leq p_m < p_{m,i+1}, i = 1, \dots, n$$

Bei einer linearen Approximation sind die Koeffizienten c_{i2} und c_{i3} gleich Null. Bei der kubischen Splineapproximation sind im Allgemeinen alle Koeffizienten von Null verschieden. Die $4n$ Koeffizienten c_{ij} ($i=1, \dots, n$; $j=0, \dots, 3$) werden aus den Stützpunktpaaren berechnet und in einer eigenen Spline-Tabelle (s. nächsten Abschnitt) abgelegt. Der NC-Interpolator greift auf ein Abbild dieser Spline-Tabelle zu und berechnet daraus nach obiger Formel die Funktionswerte.

Erzeugen der zugehörigen Spline-Tabelle

Die Spline-Tabelle wird in der Satzvorbereitung bei der Interpretation der Kopplungssyntax oder bei "**CoupleSplineTab**(<TabName>{<1|0>})" erzeugt. Die Spline-Tabelle hat folgenden Aufbau:

#-3	<Größe der Kopplungstabelle>
#-2	<Pfad und Name der Kopplungstabelle>
#-1	<Zeitstempel der Kopplungstabelle>
#1	<Interpolationsart der Stützpunkte>
	; wie Kopplungstabelle
#10	<Splineanzahl>
	; Anzahl der Splines
#20	<Periode>
	; Einheit Inkremente

#30	<Mastertyp> ; Erlaubte Führungs(achs)typen
#31	<Slavetyp> ; Erlaubte Folge(achs)typen
#200	<pm1> ; 1. Führungsachsposition
#201	<c10> <c11> <c12> <c13> ; Koeffizienten für 1. Spline
#200	<pm2> ; 2. Führungsachsposition
#201	<c20> <c21> <c22> <c23> ; Koeffizienten 2. Spline
.....	
#200	<pmn> ; n. Führungsachsposition
#201	<cn0> <cn1> <cn2> <cn3> ; Koeffizienten n. Spline

Die Spline-Tabellen werden schreibgeschützt im Verzeichnis der Linktabellen abgelegt. Sie haben den gleichen Namen wie die Kopplungstabellen, jedoch mit der Erweiterung ".s".

Beispiel:

Kopplungstabelle "curve.fct", Spline-Tabelle "curve.fct.s"

Die Spline-Tabelle wird in folgenden Fällen neu angelegt:

- Sie ist im Linkverzeichnis nicht vorhanden:
- Einer der Einträge #-1, #-2 oder #-3 stimmt nicht mit dem entsprechenden Attribut der zugehörigen Kopplungstabelle überein.
- Das Erzeugen wurde mit CoupleSplineTab(<Name>, 1) explizit programmiert.

Die Einträge #-3, #-2 und #-1 haben folgende Bedeutung:

- **#-3 <Größe der Kopplungstabelle>**
Hier wird die Größe der Kopplungstabelle in Bytes abgelegt.
- **#-2 <Name der Kopplungstabelle>**
Hier wird der komplette Name der Kopplungstabelle mit Pfad abgelegt.
Beispiel: #-2 /mnt/CoCo/nocken.fct
- **#-1 <Zeitstempel der Kopplungstabelle>**
Hier wird der Zeitstempel der Kopplungstabelle abgelegt. Diese drei Daten werden benutzt, um zu entscheiden, ob die zugehörige Spline-Tabelle neu erzeugt werden muss. Eine Verschiebung der Kopplungstabelle in ein anderes Verzeichnis, eine Änderung ihres Zeitstempels oder ihrer Größe bewirken eine Neuerzeugung der Spline-Tabelle.

14.5 Systemachskopplung

14.5.1 Beschreibung

Die Systemachskopplung bietet die Möglichkeit, kanalübergreifend elektronische Getriebe aufzubauen. Für jede Systemachse sind bis zu fünf Leitachs-Abhängigkeiten vorgesehen. Zusätzlich kann der Interpolationswert der Folgeachse berücksichtigt werden.

Restriktionen und Besonderheiten:

- Die Folgeachse kann nicht gleichzeitig zum Fahren gegen Festanschlag aktiviert werden.
- Die Folgeachse kann nicht gleichzeitig referenziert werden.
- Durch die Überlagerung mehrerer Leitachsbewegungen kann die Folgeachsgeschwindigkeit oder die Folgeachsbeschleunigung überschritten werden.
- Für die überlagerte Folgeachsbewegung wird keine Endschalterprüfung durchgeführt.
- Die Geschwindigkeit von Rundachsen und Spindeln ist auf das halbe Modulintervall (meist 180°) pro Sercos Takt begrenzt (z. B. Sercos Takt 2 ms entspricht max. 15000 U/min). Für die Erhöhung der Grenzdrehzahl bei asynchronen Endlosachsen, siehe Dokumentation Funktionsbeschreibung - Grundlagen, Kapitel 4 Antriebe (Achsen, Spindeln).

Im folgenden Beispiel soll ein elektronisches Getriebe mit den 3 Leitachsen C, Z, A und der Folgeachse B aufgebaut werden. Dabei sollen die Leitachsen jeweils in einem bestimmten Drehzahlverhältnis gekoppelt sein:

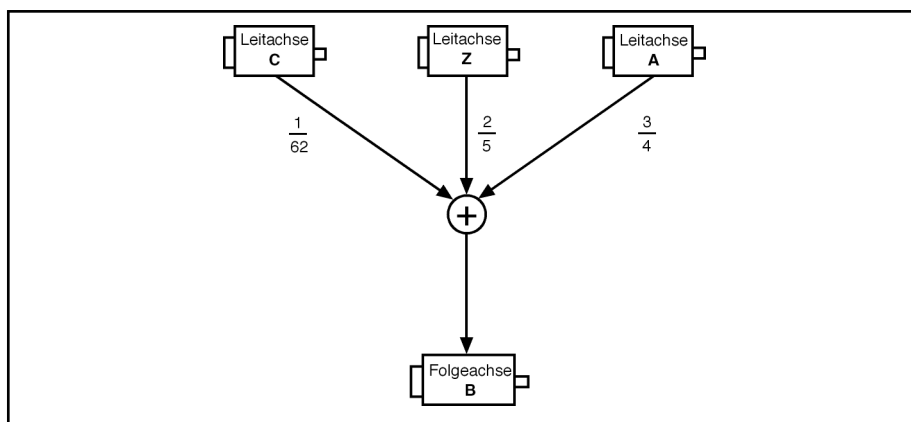


Abb. 14-16: Kopplungsbeispiel

Die Definition dieses Koppelverbandes erfolgt über eine Systemdatenstruktur. Der Anwender kann dadurch die Steuerdaten über das SPS-Programm, das CNC-Teileprogramm oder auch über die Bedienoberfläche vorgeben. Das folgende Teileprogramm definiert die oben gezeigte Beispielpkopplung:

Programm:

```

100 C%=3 : Z%=4 : A%=5 : B%=1 : REM System axis indices
101 SD.SysAxCoupleCmd[B%].OnOff = 1
102 SD.SysAxCoupleCmd[B%].Master[1].AxIndex = C%
103 SD.SysAxCoupleCmd[B%].Master[1].N_Input = 62
104 SD.SysAxCoupleCmd[B%].Master[1].N_Output = 1
105 SD.SysAxCoupleCmd[B%].Master[1].Enable = TRUE
106 SD.SysAxCoupleCmd[B%].Master[2].AxIndex = Z%
107 SD.SysAxCoupleCmd[B%].Master[2].N_Input = 5
108 SD.SysAxCoupleCmd[B%].Master[2].N_Output = 2
109 SD.SysAxCoupleCmd[B%].Master[2].Enable = TRUE
110 SD.SysAxCoupleCmd[B%].Master[3].AxIndex = A%
111 SD.SysAxCoupleCmd[B%].Master[3].N_Input = 4
112 SD.SysAxCoupleCmd[B%].Master[3].N_Output = 3
  
```

```

113 SD.SysAxCoupleCmd[B%].Master[3].Enable = TRUE
114 SD.SysAxCoupleCtr.Validate=1

```

Die Kopplungsangaben erfolgen ausschließlich für die Folgeachse B. Achsbezüge werden immer über die jeweiligen Systemachsnummern hergestellt.

Linear-, Rundachsen und lagegeregelte Spindeln sind als Folgeachsen und Leitachsen mit Sollwertkopplung zulässig. Spindeln in Drehzahlregelung dagegen können nur als Leitachsen mit Istpositionskopplung eingesetzt werden. Die Besonderheiten der Spindelkopplung werden in [Kap. 14.5.5 "Systemachskopplung mit Spindeln"](#) auf Seite 565 behandelt.

Eine Folgeachse selbst kann wiederum als Leitachse für ein weiteres Getriebe benutzt werden. Auf diese Weise können auch kaskadierte Verbände bis zu einer Tiefe von vier Leitachsebenen definiert werden.

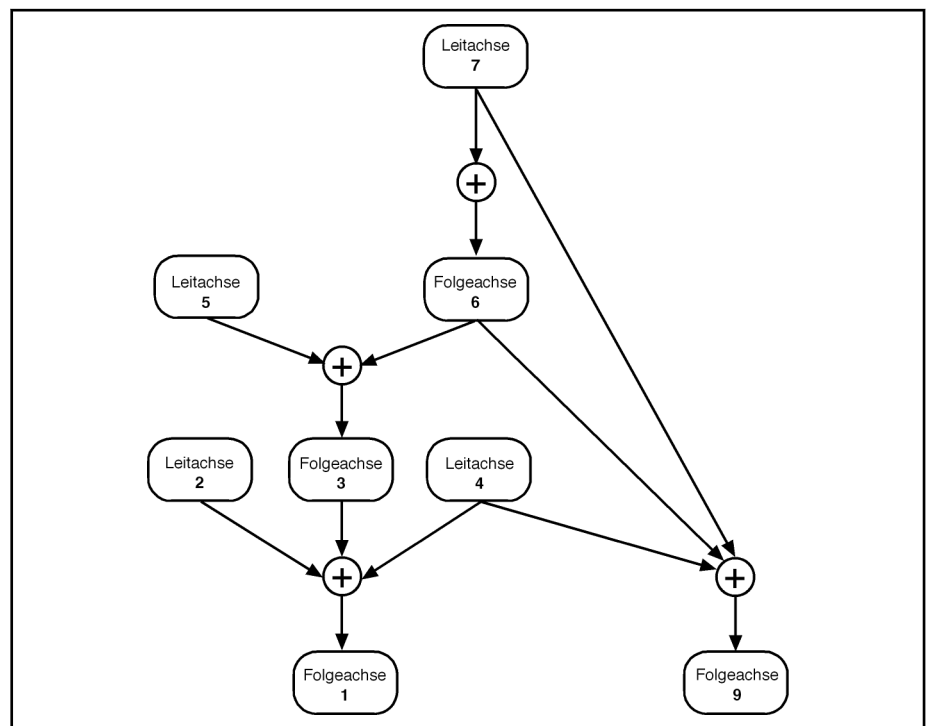


Abb. 14-17: Kaskadierte Kopplung mit drei Leitachsebenen

Die Systemachskopplungsbeziehung im Beispiel oben wird für die Achsen 1, 3, 6 und 9 in CPL wie folgt angegeben:

Programm:

```

1 SD.SysAxCoupleCmd[1].Master[1].AxIndex = 2
1 SD.SysAxCoupleCmd[1].Master[2].AxIndex = 3
1 SD.SysAxCoupleCmd[1].Master[3].AxIndex = 4
1 SD.SysAxCoupleCmd[3].Master[1].AxIndex = 5
1 SD.SysAxCoupleCmd[3].Master[2].AxIndex = 6
1 SD.SysAxCoupleCmd[6].Master[1].AxIndex = 7
1 SD.SysAxCoupleCmd[9].Master[1].AxIndex = 4
1 SD.SysAxCoupleCmd[9].Master[2].AxIndex = 6
1 SD.SysAxCoupleCmd[9].Master[3].AxIndex = 7

```

Neben der Koppelbeziehung besitzt die Systemachskopplung Steuerelemente:

SAC-Steuerelemente

OnOff	- Ein-/Ausschalten der Kopplung
IpoEnable	- Folgeachsinterpolationswert berücksichtigen

Kopplungen

Master[m].Enable	- Leitwert[m] berücksichtigen
AccCouple	- maximale Beschleunigung
VelCouple	- maximale Folgeachsgeschwindigkeit

Die Systemachskopplung ist unabhängig von der Betriebsart der, den Achsen zugeordneten Kanäle, d. h. die Folgeachsen bleiben beim Tippen, in der Handeingabe oder im Automatikbetrieb in gleicher Weise gekoppelt.

Verschiebungen der Folgeachse gegenüber dem Leitwert können zusätzlich im Tippbetrieb oder auch per Teilprogramm vorgegeben werden, da die gekoppelte Folgeachse weiterhin programmierbar bleibt.



Vorschub-Halt oder Override 0% kann eine durch Leitachsen bewegte Folgeachse nicht zum Stillstand bringen.

Das Abschalten aller Leitwerte bremst die Folgeachse auf Geschwindigkeit = 0.

Grundsätzlich kann die Systemachskopplung auch mit anderen Kopplungsarten kombiniert werden. Die folgende Übersicht zeigt, dass die Koordinatenkopplung (SCC/SCCT) und die Kanalachskopplung der Systemachskopplung vorgeschaltet ist, während die Gantrykopplung der Systemachskopplung nachgeschaltet ist.

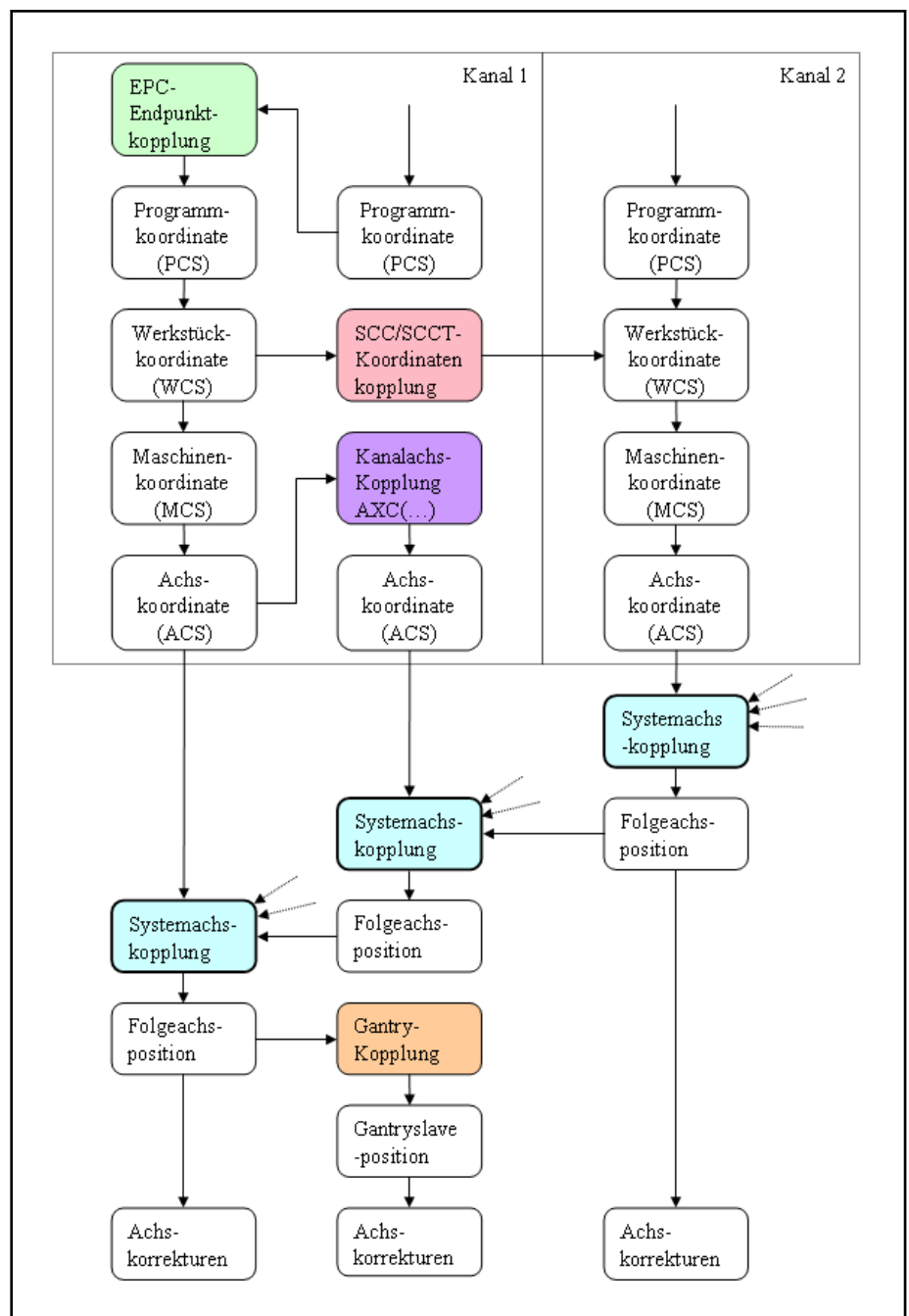


Abb. 14-18: Übersicht der MTX-Kopplungsarten

Mit dem Zu- oder Abschalten der Systemachskopplung wird ein Interpolator aktiv, der die Folgeachse an den aktiven Leitwert (Geschwindigkeit und Position im Kopplungsverhältnis der Leitachsen) heranführt.



Das Zu- oder Abschalten bzw. Verändern der Kopplung führt bei Geschwindigkeits- und Lagedifferenzen zwischen Leitwert und Folgeachseposition zu Ausgleichsbewegungen der Folgeachse.

Die Kopplung außerhalb des Gefahrenbereichs an- und abschalten.

14.5.2 Applizieren der Systemachskopplung

Systemdatenübersicht: Die Systemachskopplung wird gesteuert durch die Systemdatenstruktur

SysAxCoupleCmd

Die Struktur legt für jede Systemachse die Abhängigkeit zu Leitachsen mit den entsprechenden Kopplungsverhältnissen fest und signalisiert dem Interpolator das Zu- und Abschalten der Kopplung oder einzelner Leitwerte. Die Struktur bleibt über das Ausschalten hinaus bestehen, so dass nach dem Steuerungshochlauf die Kopplungsdaten erneut vorgegeben werden können.

Mit Hilfe der Kontrollstruktur

SysAxCoupleCtr

werden die aktuellen Steuerdaten aller Systemachsen in SysAxCoupleCmd für gültig erklärt. Fehlerhafte Daten lehnt das System mit einer Fehlernummer und dem entsprechenden Elementindex ab. Sind alle Elemente gültig, werden die Steuerdaten für die Achskopplung kopiert und frei geschaltet. Die Steuerdaten erscheinen nach der erfolgreichen Aktivierung auch in den Statusdaten.

Diese Statusinformationen sowie der Zustand der Kopplung werden mit jeder Änderung aktualisiert und sind zusammengefasst in der Struktur

SysAxCoupleSta

Die Statusdaten werden mit dem Aus-/Einschalten der Steuerung gelöscht und stehen mit dem Aktivieren der Kopplung wieder zu Verfügung.

**Steuerung der
Systemachskopplung:**

Die kleinste Kopplungseinheit - das Systemachskopplungselement (SAC) aus der Sicht der Folgeachse - liegt in der Systemdatenstruktur **SysAxCoupleCmd[i]** und steuert das An- und Abkoppeln der indexgleichen Systemachse:

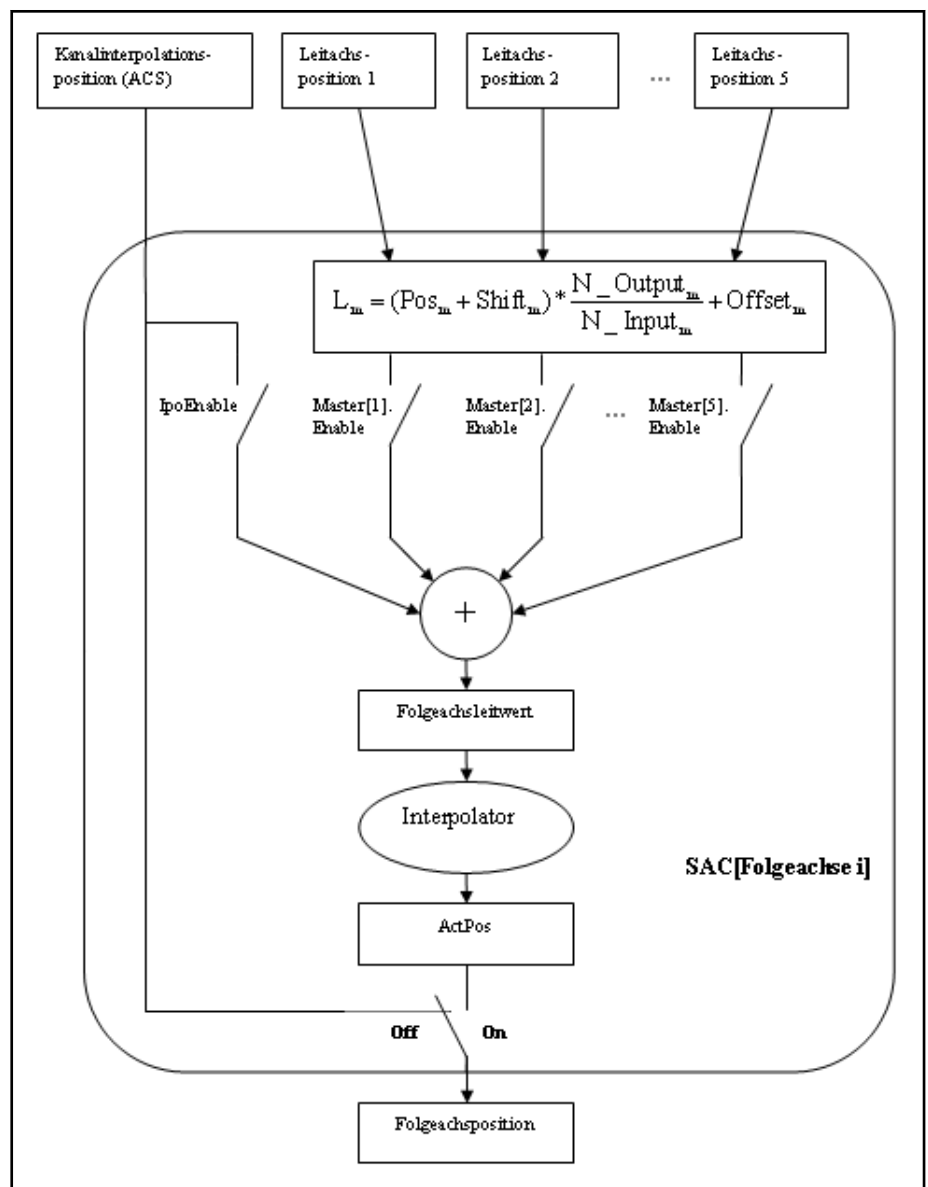


Abb. 14-19: Prinzipschaltbild der Systemachskopplung

Nach dem Systemhochlauf sind alle SAC's im inaktiven Zustand. Der **OnOff**-Schalter aktiviert die Systemachskopplung. Leitachs- und Interpolationswerte der Folgeachsen sind dabei getrennt zu- und abschaltbar (**Master[m].Enable**, **IpoEnable**). Der SAC-Interpolator sorgt dafür, dass die Folgeachse an die Folgeachsleitposition und Folgeachsleitgeschwindigkeit herangeführt wird. Das An- /Abschalten der Systemachskopplung, sowie das Zu-/Wegschalten von Leitwerten kann im Stillstand, als auch fliegend - bei sich stetig ändernden Leitwerten - erfolgen.

Der SAC-Interpolator berücksichtigt bei der Leitwertankopplung entweder die Geschwindigkeit **VelCouple** und die Beschleunigung **AccCouple** aus den Systemdaten (**Rigidity=0**) oder die maximale Achsgeschwindigkeit und -beschleunigung aus den Maschinenparameter (**Rigidity=1**).

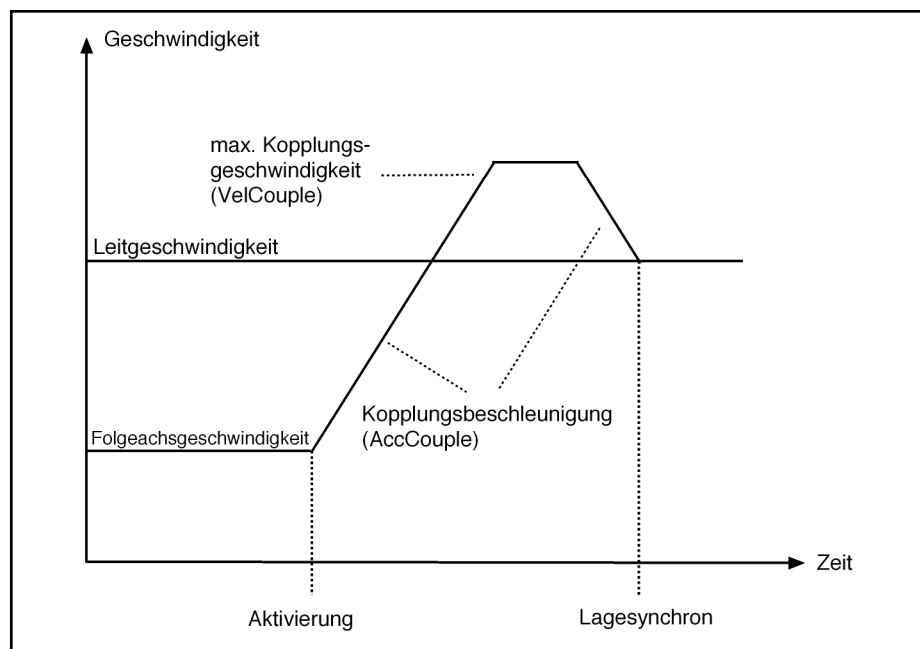


Abb. 14-20: Geschwindigkeitsprofil der Ankopplung

Erreicht die Folgeachse im Berechnungstakt des Interpolators die Leitgeschwindigkeit und die Leitposition, ist die Kopplung hergestellt. Maximale Geschwindigkeit und Beschleunigung werden im gekoppelten Zustand immer auf die Maschinenparameter gesetzt. Bei zu großen Leitwertänderungen kann die Kopplung abreißen. Ist das Wiederankoppeln (**RecoupleEnable**) erlaubt, versucht der SAC-Interpolator den Leitwert einzufangen. Andernfalls wird ein Fehler generiert, der zu einer Ausnahmebehandlung und schließlich zum Abbruch der Kopplung führen kann.

Mit dem Abschalten aller Leitachswerte (`Master[m].Enable=FALSE`) und dem Interpolationswert (`IpoEnable=FALSE`) im aktivem SAC-Element bremst die Folgeachse mit der achszulässigen Verzögerung (`Rigidity=1`) bzw. mit `AccCouple` (`Rigidity=0`) bis zum Stillstand.

Über das Ausschalten (`OnOff=0`) der Systemachskopplung bringt der SAC-Interpolator die Folgeachse wieder in die Interpositionsposition. Sobald Geschwindigkeit und Position im Berechnungstakt übereinstimmen, schaltet sich die Kopplung ab und die Interpositionsposition wird wieder direkt zur Achsvorgabe.



Die aktuelle Folgeachseposition ist in der MTX-Standard-Bedienoberfläche als "\$Sollposition" visualisierbar (siehe Dokumentation "MTX Standard-NC-Bedienung", Kap. "Positions-Anzeige").

ACS				Sollposition	Endposition	Restweg	\$ Sollposition
X			mm	10.000	--	0.000	15.000
Y			mm	5.000	--	0.000	5.000
Z			mm	0.000	--	0.000	0.000
VA		a	mm	0.000	--	0.000	0.000

ACS				Sollgeschw.	Solldrehzahl	Istdrehzahl	Sollposition	\$ Sollposition	Istposition
B1		a	o	0.000	0.000	0.000	0.000	0.000	0.000
B2		a	o	0.000	0.000	0.000	0.000	0.000	0.000
B4		a	o	0.000	0.000	0.000	0.000	0.000	0.000

		Programm	Sollwert	Istwert	Override	Max	Getr.St.
F		mm/min	0.0	10000.0	0.0	100%	120%
S1		1/min	0.0	0.0	0.0	100%	120%


```

/mnt/Test_utility/sac/sysac.npg
; -- Activate SAC --
1 SD.SysAxCoupleCmd[X%].OnOff = 1
1 SD.SysAxCoupleCmd[X%].IpoEnable = TRUE
1 SD.SysAxCoupleCmd[X%].RecoupleEnable = TRUE
1 SD.SysAxCoupleCmd[X%].Master[1].Enable = TRUE
1 SD.SysAxCoupleCmd[X%].Master[1].AxIndex = Y%
1 SD.SysAxCoupleCmd[X%].Master[1].N_Input = 1
1 SD.SysAxCoupleCmd[X%].Master[1].N_Output = 1
1 SD.SysAxCoupleCtr.Validate = 1
SACSYNC(X=3)
G1 X10 Y5 F10000 M0
WAIT
; -- Deactivate SAC --

```


G80	G01	G17	G90	G94	G8
G62	G48	FDG	MOC	G40	G71
OVE					
G53.1	G53.2	G53.3	G53.4	G53.5	
/database/ZO1.zot					
/database/DC1.dct					

WZ-Aktiv	akt. Schn.	D-Nummer
	ED0	D0

Abb. 14-21: Anzeige der Folgeachseposition mit \$Sollposition

Den aktuellen Zustand der Kopplung zeigt die Systemachskopplung in **SysAxCoupleSta[i].State** an. Zustandsänderungen können durch neue Systemdatenvorgaben oder durch den SAC-Interpolator selbst ausgelöst werden.

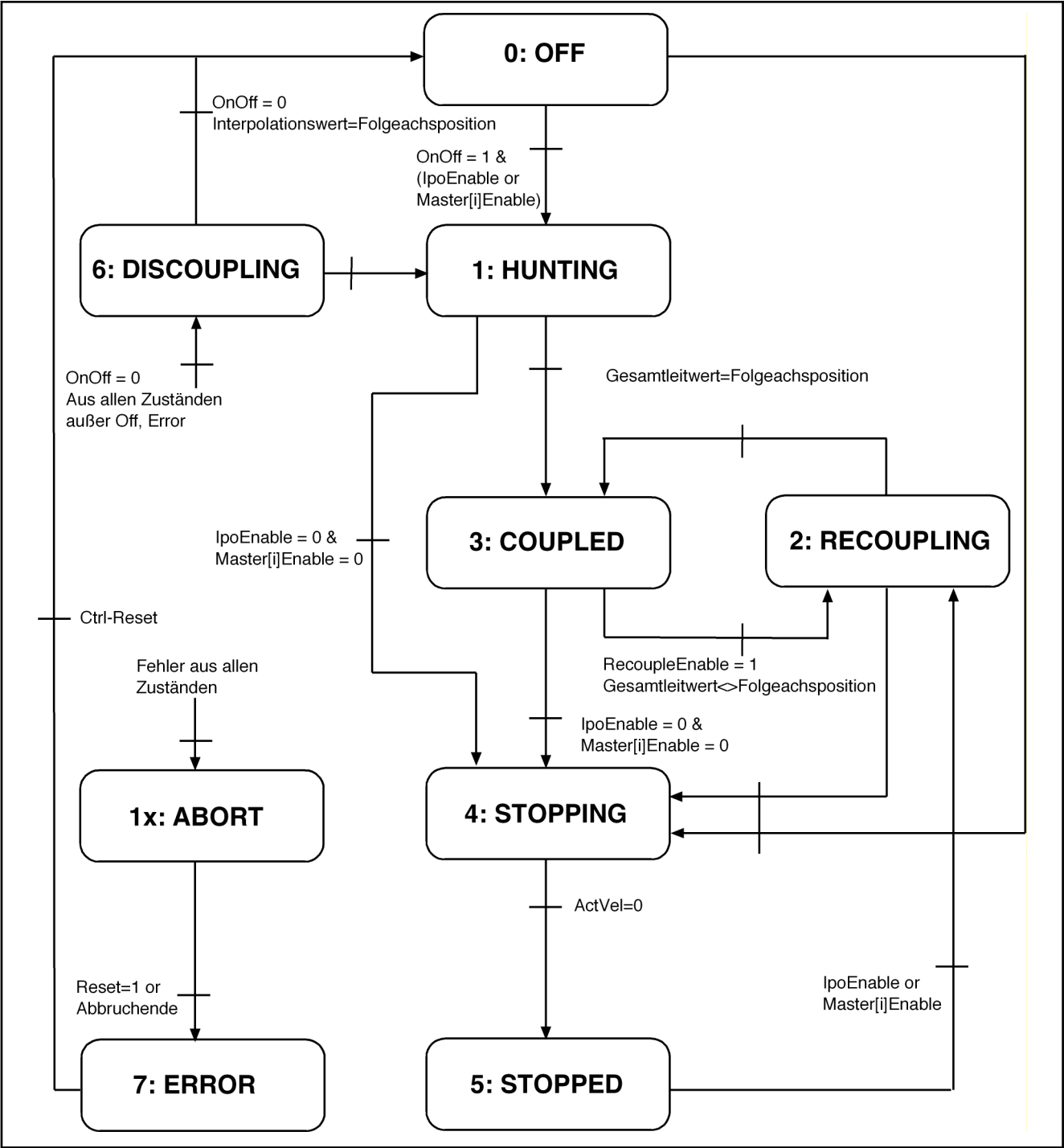


Abb. 14-22: Zustandsübergang der Systemachskopplung

Zustände der Systemachskopplung:

0	OFF	-	Die Systemachskopplung ist inaktiv.
1	HUNTING	-	Der SAC-Interpolator versucht den Leitwert erstmals zu erreichen.
2	RECOUPLING	-	Der SAC-Interpolator versucht den Kopplungszustand wieder herzustellen

3	COUPLED	-	Folgeachswert und Leitwert stimmen in Position und Geschwindigkeit im Berechnungstakt überein.
4	STOPPING	-	Alle Leitwerte wurden deaktiviert, der SAC-Interpolator stoppt die Folgeachse.
5	STOPPED	-	Die Folgeachse hat keinen Leitwert und steht still.
6	DISCOUPLING	-	Die Systemachskopplung soll abgeschaltet werden, der SAC-Interpolator führt die Achse auf den Interpolationswert zurück.
7	ERROR	-	Es ist eine Ausnahmesituation aufgetreten, die Kopplung wurde abgebrochen.
1x	ABORT	-	Es wurde eine Ausnahmebehandlung ausgelöst. Die Ausnahmebehandlung benutzt den SAC-Interpolator, so dass sich ABORT mit den Zuständen 1-7 überlagert (z. B. 14 ABORT-STOPPING).
2x	PENDING	-	Es ist mindestens eine bedingte Umschaltung (SwitchCond > 0) aktiv und noch nicht abgeschlossen. Der bisherige Zustand der Systemkopplung wird bis zur Umschaltung überlagert (z. B. 23 PENDING-COUPLED).

Tab. 14-6: Zustände der Systemachskopplung

Koppelverhältnisse und Leitwertbestimmung:

Die Leitachsen lassen sich mit der Leitachsverschiebung (Shift) in dem Verhältnis Ausgangsumdrehung (N_Output)/ Eingangsumdrehung (N_Input) plus einem Leitwertversatz (Offset) koppeln. Die Summe aller bewerteten Leitachsen mit dem Achswert der Folgeachse ergeben den Gesamtleitwert:

$$\begin{aligned}
 \text{Folgeachs}_{\text{Leitwert}} &= \text{Folgeachs}_{\text{Interpolationswert}} \\
 &+ (\text{Leitachssollposition}_1 + \text{Shift}_1) \cdot \frac{N\_Output_1}{N\_Input_1} + \text{Offset}_1 \\
 &+ (\text{Leitachssollposition}_2 + \text{Shift}_2) \cdot \frac{N\_Output_2}{N\_Input_2} + \text{Offset}_2 \\
 &\dots \\
 &+ (\text{Leitachssollposition}_5 + \text{Shift}_5) \cdot \frac{N\_Output_5}{N\_Input_5} + \text{Offset}_5;
 \end{aligned}$$

Abb. 14-23: Leitwertberechnung

Die Kopplungsfaktoren können auch als reelle Zahlen angegeben werden, um z. B. Verschiebungen von linearen Leitachse zu berücksichtigen.



Für die Kopplung zweier endlos drehenden Rundachsen sollten die Verhältnissfaktoren N_Output und N_Input nach Möglichkeit immer ganzzahlig angegeben werden. Andernfalls kann sich der Darstellungsfehler eines Faktors (14 Stellen) mit jeder weiteren Umdrehung der Leitachse zu einer messbaren Abweichung aufakkumulieren.

Bei Zahnradbearbeitungen wie dem Wälzfräsen oder dem Zahnradstoßen wiederholt sich der Eingriffsbereich des Werkzeugs für jede Zahnücke, so dass ein schnelleres Ankoppeln an eine beliebige Zahnücke Synchronisationszeit einspart. Die Teilung des Zahnrads an der Folgeachse gibt der Anwender mit dem Systemdatum GearTeeth bekannt. Die Steuerung koppelt somit an den nächstliegenden Teilbereich.

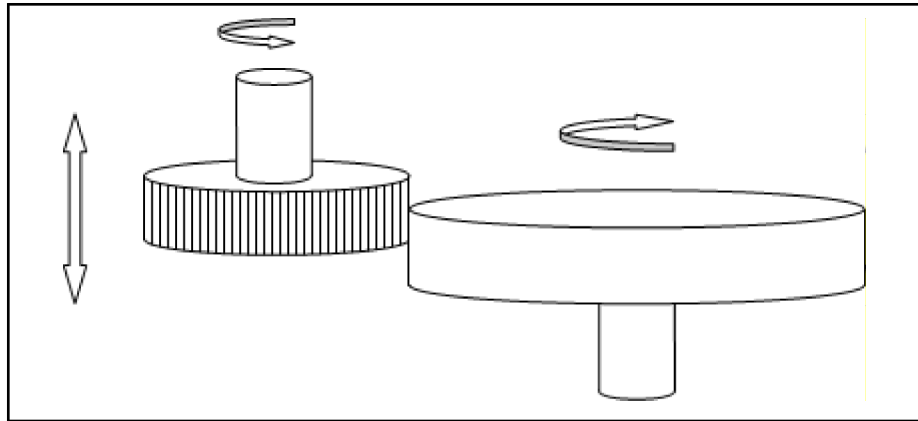


Abb. 14-24: Zahnradstoßen

Der Zahnrad-Rohling wird durch die Folgeachse angetrieben. Das Zahnrad wird mit 62 Zähnen gefertigt, wobei das Werkzeug 42 Zähne besitzt. Das Übersetzungsverhältnis kann mit $N\_Input = 62$ und $N\_Output = -42$ angegeben werden. Die Teilung des Zahnrads mit $GearTeeth = 62$. Die maximale Lageabweichung der Folgeachse zum Leitwert beträgt somit nur noch den halben Teilbereich: $360^\circ / (2 \cdot 62) = 2,9032^\circ$ anstatt bis zu 180° bei absoluter Kopplung.

Mit $GearTeeth=0$ wird eine absolute Kopplung erzwungen, welche beim Umschalten von 0 geleisteten Eingangsumdrehungen der Leitachse ausgeht.

Masterkopplung mit Tabellen:

Mit Hilfe der Tabellenkopplung können nichtlineare Kopplungszusammenhänge zwischen Leit- und Folgeachse abgebildet werden.

Die Systemachskopplung erlaubt für jede Leitachse ($Master[m]$) die Angabe einer separaten Kopplungstabelle.

Diese ist im Systemdatum **SysAxCoupleCmd[i].Master[m].Tab** einzutragen.

Der Gesamtleitwert einer Folgeachse ergibt sich bei aktiver Tabellenkopplung mit:

$$\begin{aligned}
 Folgeachs_{Leitwert} &= Folgeachs_{Interpolationwert} \\
 &+ Tabellenwert_1 (Leitachssollposition_1 + Shift_1) \cdot \frac{N\_Output_1}{N\_Input_1} + Offset_1 \\
 &+ Tabellenwert_2 (Leitachssollposition_2 + Shift_2) \cdot \frac{N\_Output_2}{N\_Input_2} + Offset_2 \\
 &\dots \\
 &+ Tabellenwert_s (Leitachssollposition_s + Shift_s) \cdot \frac{N\_Output_s}{N\_Input_s} + Offset_s
 \end{aligned}$$

Abb. 14-25: Leitwertberechnung mit Tabellen

Mit $SysAxCoupleCmd[i].Master[m].Shift$ kann eine Phasenverschiebung zur Leitachssposition realisiert werden. Dies ermöglicht z. B. die Nutzung einer Sinus-Tabelle zur Erzeugung der Kosinusfunktion durch eine 90° -Verschiebung.

Fehlt die Angabe der Mastertabelle, gilt:

$Tabellenwert_i = Leitachssollposition_i + Shift_i$.

Aufbau und Wirkungsweise von Kopplungstabellen sind im [Kap. 14.4 "Kopplungstabellen"](#) auf Seite 526 näher erläutert.

Um eine Tabelle in der Systemachskopplung verwenden zu können, muss diese mit dem NC-Befehl **CoupleSplineTab(TAB(<tabelle>),LOAD)** in den Systemtabellenspeicher **SysAxCoupleTab** geladen werden.

Die geladenen Tabellen können per Tabellennamen ohne Pfadangabe oder über die Platznummer im Systemtabellenspeicher SysAxCoupleTab adressiert werden:

```
;Erste namensgleiche Tabelle in SysAxCoupleTab referenzieren
CST(TAB(/mnt/tables/stroke.fct),LOAD)
200 SysAxCoupleCmd[1].Master[1].Tab = "stroke.fct"
...
;Tabelle per Platz referenzieren
CST(TAB(/mnt/tables/stroke.fct),LOAD,[ PLACE% ])
400 SysAxCoupleCmd[1].Master[2].Tab = $STR(PLACE%)
```

Beispiel:

Die mechanische Nockenkopplung XB → YB soll durch die Systemachstabel-
lenkopplung ersetzt werden:

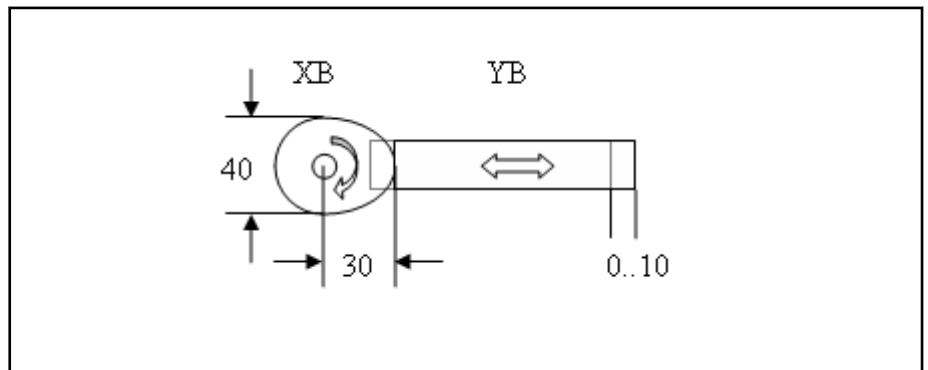


Abb. 14-26: Nockenkopplung

In der Tabellendatei "nocken.fct" werden dazu für XB-Winkel im Abstand von 5° Nockenradien eingetragen. Für Winkel zwischen den Stützstellen soll der Nockenradius kubisch interpoliert werden. Ein Offset von -20.0 verschiebt die YB-Position in das vorbestimmte Intervall [0..10] mm.

Tabellendatei "nocken.fct":

```
#1 3 ;kubische Interpolation
#20 1 ;periodische Tabelle
#100 0.0 30.0 ;Stützstelle
#100 5.0 29.73 ;Stützstelle
#100 10.0 28.99;Stützstelle
....
#100 360.0 30.0;Stützstelle
```

Programm:

```
;-----
; Displacing the YB axis by a rotatory cam motion ; of the XB axis
;-----
; System axis indices of the XB and YB axis: 1 XB%=4 : YB%=5 ...
;-----
; 1. Table file loaded in the system table memory (cache):
;-----
; CoupleSplineTab(TAB(nocken.fct),LOAD) ...
;-----
; 2. Activating cam table in the coupling relation
;-----
1 SD.SysAxCoupleCmd[YB%].Master[1].Enable = TRUE
```

Kopplungen

```

1 SD.SysAxCoupleCmd[YB%].Master[1].AxIndex = XB%
1 SD.SysAxCoupleCmd[YB%].Master[1].Tab      = "nocken.fct"
1 SD.SysAxCoupleCmd[YB%].Master[1].Offset
    = -20.0 1 SD.SysAxCoupleCmd[YB%].Master[1].N_Input
    = 1.0 1 SD.SysAxCoupleCmd[YB%].Master[1].N_Output
    = 1.0 1 SD.SysAxCoupleCtr.Validate=1 ...
;-----
; 3. Synchronous motion of the YB axis with XB rotation
;-----
; XB=S(100) ...
;-----
; 4. Removing table from the coupling relation
;-----
1 SD.SysAxCoupleCmd[YB%].Master[1].Enable = FALSE
1 SD.SysAxCoupleCmd[YB%].Master[1].Tab    = ""
1 SD.SysAxCoupleCtr.Validate=1 ...
;-----
; 5. Removing table from system table cache
;-----
; CoupleSplineTab(TAB(nocken.fct),UNLOAD) ...

```

Erzeugen und Laden einer Spline-Tabellendatei:

Bis zu 100 unterschiedliche Tabellen können gleichzeitig in der Systemachskopplung verwendet werden. Nach dem Einschalten der Steuerung befinden sich keine Tabellen im Speicher. Für die Erstellung und Verwendung von Tabellen gelten die oben genannten Bedingungen

Siehe [Kap. 14.4 "Kopplungstabellen" auf Seite 526](#).

Im Systemdatum **SysAxCoupleTab[1..100]** wird für jede geladene Tabelle der Ablageort, Zeitstempel und Name der geladenen Quelldatei, sowie der Zustand der Tabelle und die Größe des belegten Tabellenspeichers angezeigt.

Zustände der Kopplungstabellen:

- **0 - Tabellenspeicher nicht belegt:**
Der Tabellenspeicher ist noch frei und kann mit
CoupleSplineTab(TAB(<tab>),LOAD) belegt werden.
- **1 - Tabelle wird geladen:**
Der Ladevorgang für die Tabelle ist noch nicht abgeschlossen, die Tabelle kann noch nicht verwendet werden.
- **2 - Tabelle ist geladen:**
Die Tabelle kann jetzt in der Systemachskopplung verwendet werden, d. h. sie wird für mindestens einen Master in
SysAxCoupleCmd[axis].Master[m].Tab eingetragen und mit
SysAxCoupleCtr.Valid=1 aktiviert.
Nicht mehr benötigte Tabellen können in diesem Zustand mit CoupleSplineTab(TAB(<tab>),UNLOAD) entfernt werden.
- **3 - Tabelle wird benutzt:**
Die Tabelle wird in der Systemachskopplung verwendet und kann nicht aus dem Tabellenspeicher entfernt werden. Weitere Leitachskopplungen können ebenfalls diese Tabelle benutzen.
Falls die Tabelle in allen Kopplungen deaktiviert wurde, setzt die Tabellenverwaltung den Zustand "2-Tabelle geladen" wieder ein.
- **4 - Tabelle wird entfernt:**
Dieser Zustand wird kurzfristig beim Entfernen der Tabelle mit
CoupleSplineTab(TAB(<tab>),UNLOAD) eingenommen.

Systemdatendefinition des Tabellenspeichers SysAxCoupleTab[i]:

Element	Typ	Wertebereich	Beschreibung
.FileName	String	1..100	Dateiname mit Pfadangabe der Quelldatei
.State	Short	0..4	0 - nicht belegt 1 - Tabelle wird geladen 2 - Tabelle ist geladen 3 - Tabelle wird benutzt 4 - Tabelle wird entfernt
.FileTime	String	dd.mm.jj hh:mm:ss	Zeitstempel der geladenen Quelltabellenda- tei
.CacheSize	Int	-	Größe des belegten Tabellenspeichers

Tab. 14-7: Koppeltabellenspeicher

Mit dem NC-Befehl

CoupleSplineTab(TAB(<Tab>,<1|0>),LOAD{,<Platz>}) wird eine Tabelle vorbereitet und in den Tabellenspeicher geladen.

Ist kein Speicher mehr frei, kann mit

CoupleSplineTab(TAB(<Tab>,<1|0>),UNLOAD{,<Platz>})

eine nicht mehr benötigte Tabelle aus dem Speicher entfernt werden.

Die Kurzform lautet: CST

Mit den Parametern:

TAB <Tab>	Name der Kopplungstabelle
<1 0>	0 - Tabelle nur vorbereiten und laden, wenn Tabellendatei neuer als Tabelle im Speicher 1 - Tabelle immer neu vorbereiten und laden
LOAD	Tabelle in Systemtabellenspeicher laden
UNLOAD	Tabelle aus Systemtabellenspeicher entfernen
Platz	Systemtabellenspeichereintrag SysAxCoupleTab[Platz] laden bzw. freigeben

Beispiele:

CST(TAB(curve.fct),LOAD)	Erstellt, falls notwendig, die Spline-Tabelle / <Linkverzeichnis>/curve.fct.s. und lädt die Tabelle in den Systemspeicher.
CST(,UNLOAD,5)	Entferne Tabelle SysAxCoupleTab[5] aus dem Systemta- bellenspeicher

Formelkopplungen

Bewegungsgesetze und geometrische Kompensationen, die sich mit mathematischen Formeln beschreiben lassen, können direkt mit der Formelverarbeitung der Systemachskopplung im Systemdatum **SysAxCoupleCmd[axis].Formula** angegeben werden.

Die Formel erlaubt die Verknüpfung von Einzelleitwerten mit mathematischen Operationen und Funktionen. Ein Einzelleitwert ist jeweils durch das Systemdatum SysAxCoupleCmd[axis].Master[m] definiert und enthält bereits die Skalierung, Verschiebung und Tabellenbewertung der Leitachspolition. Das Ergebnis der Formel ist der Folgeachsleitwert.

*Beispiel:***Ausgleichsbewegung durch Systemachsformelkopplung**

Ein Werkzeug soll der nichtzentrischen Bewegung eines Drehteils mit dem Radius "R" und der Außermittigkeit "r" in Abhängigkeit der Werkstückachse C mit der linearen Zustellbewegung X folgen. Die Drehmitte der C-Achse befindet sich im Koordinatenursprung der X-Achse.

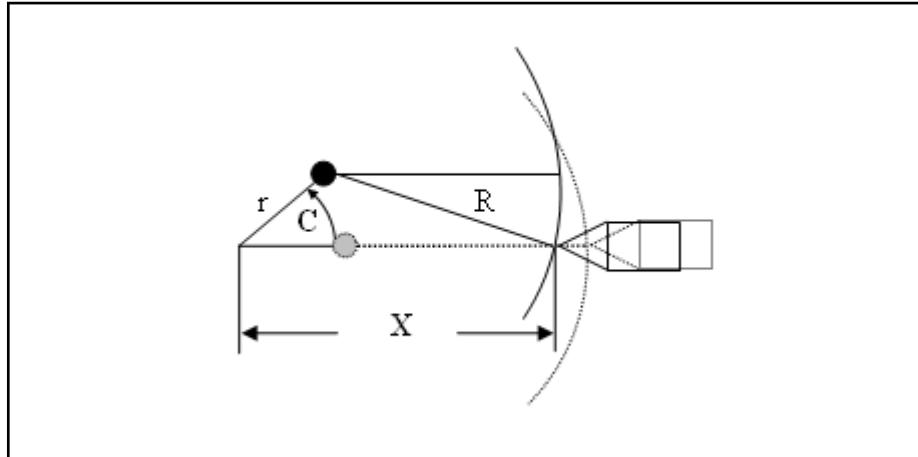


Abb. 14-27: *Kompensation eines außermittig rotierenden Drehteils*

Das Kopplungsgesetz zur Auslenkung der X-Achse als Funktion der C-Achse lautet:

$$X = r \cos(C) + R \sqrt{1 - \left(\frac{r}{R}\right)^2 \sin^2(C)}$$

Abb. 14-28: *Schubkurbelformel*

Das elektronische Getriebe kann entsprechend mit Hilfe der Systemachsformelkopplung direkt definiert werden:

Programm:

```
; System axis indices for the master axis (C) and the slave axis (X)
10 X%=1 : C%=7

; master value definition (L1)
20 SD.SysAxCoupleCmd[X%].Master[1].AxIndex = C%
30 SD.SysAxCoupleCmd[X%].Master[1].N_Input = 1.0
40 SD.SysAxCoupleCmd[X%].Master[1].N_Output = 1.0

; modulo calculation in the C-axis to
; prevent overflow of the master value:
50 SD.SysAxCoupleCmd[X%].Master[1].Mode = 1

; constant parameter: r=1.5, R=10
60 SD.SysAxCoupleCmd[X%].Formula="1.5*COS(L1)+10*SQRT(1-0.0225*SQR(SIN(L1)))"

; Aktivierung
70 SD.SysAxCoupleCtr.Validate = 1
```

In jedem Interpolationstakt wird über die Formel aus dem zeitdiskreten Winkel der C-Achse eine Folgeachsposition X ermittelt:

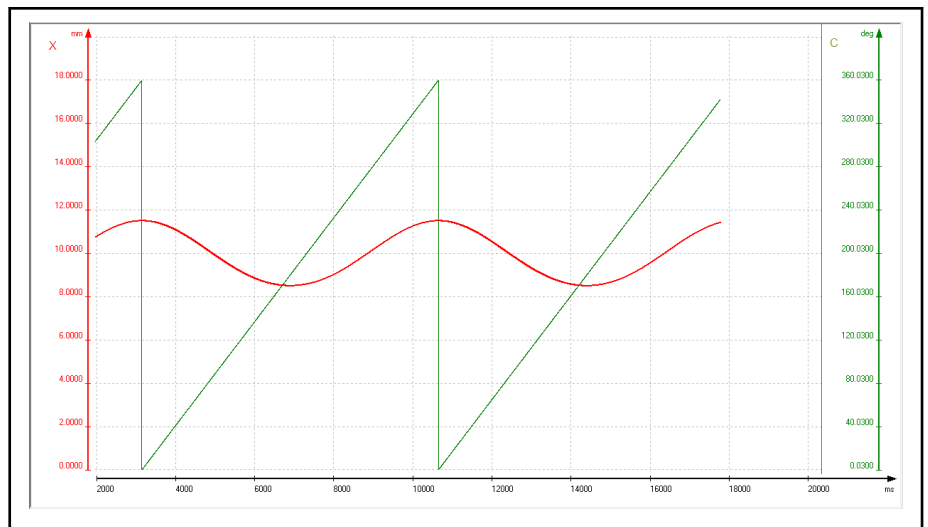


Abb. 14-29: Oszilloskopaufzeichnung zur Formelkopplung

Mit der Ersetzung von R durch den Folgeachs_{Interpolationswert} "AC" kann auch die programmierte X-Zustellung des Werkzeugs berücksichtigt werden:

Programm:

```
60 SD.SysAxCoupleCmd[X%].Formula="1.5*COS(L1)+AC*SQR(1-SQR(1.5/AC*SIN(L1)))"
```

Die Standard-Leitwertberechnung bildet aus der Summe der aktiven Einzel-leitwerte den Gesamtleitwert für die Folgeachse.

Die Formelverarbeitung ersetzt die Summenbildung durch die folgeachsspezifische Berechnungsvorschrift des Anwenders:

$$\text{Folgeachs}_{\text{Leitwert}} = \text{Formel}(\text{AC}, L1, \dots, L5)$$

mit den Operanden:

$$\text{AC}[\text{mm}/^\circ] = \text{Folgeachs}_{\text{Interpolationswert}}$$

$$L1[\text{mm}/^\circ] = \text{Tabellenwert}_1(\text{Leitachssollposition}_1 + \text{Shift}_1) \cdot \frac{N_{\text{Output}_1}}{N_{\text{Input}_1}} + \text{Offset}_1$$

...

$$L5[\text{mm}/^\circ] = \text{Tabellenwert}_5(\text{Leitachssollposition}_5 + \text{Shift}_5) \cdot \frac{N_{\text{Output}_5}}{N_{\text{Input}_5}} + \text{Offset}_5$$

Abb. 14-30:

Restriktionen:

- Eine ausgiebige Nutzung der Formelverarbeitung bei kurzen NC-Zykluszeiten kann zur Überlastung des Interpolators führen.
- In Formeln der Systemachskopplung sind nur interpolationssynchron veränderte permanente Variablen und Systemdaten erlaubt. Diese Bedingung erfüllen Variablen einer interpolationssynchronen FA-Task und vom Interpolator aktualisierte Systemdaten. Der Formelstring selbst kann mit der CPL-Stringverarbeitung zusammengesetzt werden.
- Die Operationen der Formel werden in Gleitkommaarithmetik in doppelter Genauigkeit ausgeführt. Der maximale Wertebereich aller Funktionen und Operationen ist auf +/-9.9999999999E99 begrenzt. Bei Überschreitung des maximalen Wertebereichs wird der jeweilige Grenzwert eingesetzt und ein Überlaufer diagnostiziert. Der kleinste erlaubte Wert ungleich Null ist +/- 1.0E-99. Bei Unterschreitung des Wertebereichs wird der jeweilige Grenzwert eingesetzt und ein Unterlaufer diagnostiziert.

Kopplungen

reichs wird das Ergebnis zu Null gesetzt. Bei Multiplikationen löst der Unterlauf eine Fehlerdiagnose aus.

- Positionen und Leitwerte werden in der Einheit mm (Linearachsen) bzw. ° (Rundachsen) verrechnet. Geschwindigkeiten liegen entsprechend in der Einheit mm/IPO-Takt bzw. °/IPO-Takt vor. Für die Normierung von konstanten Geschwindigkeiten und Beschleunigungen in die IPO-Taktzeit stehen die Funktionen FEED, SPEED und ACC zur Verfügung.
- Die trigonometrischen Funktionen verarbeiten Winkleinheiten.
- Endlos drehende Leitachsen, die per Formel an eine Linearachse gekoppelt werden, können zu einem Überlauf des Leitwerts führen, wenn sie nicht mit SysAxCoupleCmd[axis].Master[m].Mode=1 in das Leitachsenmodulintervall überführt werden.
- Unstetige oder sich stark ändernde Funktionsverläufe der Formel führen bei Überschreitung der maximalen Folgeachsgeschwindigkeit oder der maximalen Folgeachsbeschleunigung zum Abreißen der Kopplung.
- In Formeln kann die Folgeachssollposition SPOS aus dem vorherigen Interpolationstakt verwendet werden. Die Benutzung von SPOS kann zu unerwünschten Schwingungen beim Ankoppeln führen.

Definition und Aktivierung der Formelverarbeitung

Die Definition der Formel erfolgt im Systemdatum SysAxCoupleCmd[axis].Formula der Systemachskopplung. Die Formel wird als String für die betreffende Folgeachse hinterlegt.

Beispiele für Formeldefinitionen sind:

Programm:

```
;Default master value calculation
100 SD.SysAxCoupleCmd[X%].Formula="AC+L1+L2+L3+L4+L5"

;involute: x= r*(cos(phi)+phi*sin(phi)),
;          y= r*(sin(phi)-phi*cos(phi))
;          (for phi in [rad])
;          With r=10.0; phi=L1 [°], is the result:
200 SD.SysAxCoupleCmd[X%].Formula="10.0*(COS(L1)+L1/180*PI*SIN(L1))"
210 SD.SysAxCoupleCmd[Y%].Formula="10.0*(SIN(L1)-L1/180*PI*COS(L1))"

;Polynomial 4th degree: f(x) = ax^4 + bx^3 + cx^2 + dx + e
;          acc. to the Horner scheme with
;          x=L1, a=3.02E-12, b= -1.012E-10, c=4.25E-8, d=-1.33E-6 e=0.7E-3
300 SD.SysAxCoupleCmd[X%].Formula=
  "(((3.02E-12*L1-1.012E-10)*L1+4.25E-8)*L1-1.33E-6)*L1+0.7E-3"

;conditioned expression
400 SD.SysAxCoupleCmd[C%].Formula="ABS(L1)>0?ATAN(L2/L1):90.0"

;disable formula (add master values)
500 SD.SysAxCoupleCmd[C%].Formula=""
```

Mit der Datenübernahme (SysAxCoupleCtr.Validate=1) wird die Formel überprüft und für den Interpolator vorbereitet. Übernahmefehler liegen in SysAxCoupleCtr.CfgErrNum ab:

- 29 (Syntax Fehler)
Die Formelsyntax wurde nicht eingehalten. Operanden oder Operatoren sind unbekannt oder wurden in einer falschen Reihenfolge programmiert. Die Abbruchposition bei syntaktischen Formelfehlern liegt im Systemdatum SysAxCoupleCtr.CfgErrMaster.
- 30 (Laufzeit überschritten)

Die Laufzeitmessung aller programmierten Formeln der Systemachskopplung hat eine unzulässige Systembelastung ergeben. Entweder muss die NC-Zykluszeit verlängert oder eine leistungsfähigere Hardware eingesetzt werden.

- 31 (Fehler bei der Formelverarbeitung)

Die Schachtelungstiefe (z. B. Klammerebenen) ist möglicherweise zu groß. Die Abbruchposition liegt im Systemdatum SysAxCoupleCtr.CfgErrMaster ab.

Das nachfolgende CPL-Beispiel zeigt eine Syntaxfehlerdarstellung:

```
; Invalid character (#) in the formula definition:
100 SD.SysAxCoupleCmd[Y%].Formula="10.0#*(SIN(L1)-L1/180*PI*COS(L1))"
110 LP VALIDATE
...
; transfer program
LPS VALIDATE
210 SD.SysAxCoupleCtr.Validate = 1
220 ERRNO% = SD.SysAxCoupleCtr.CfgErrNum
230 AXIS% = SD.SysAxCoupleCtr.CfgErrAx
240 MASTER% = SD.SysAxCoupleCtr.CfgErrMaster
250 CASE ERRNO% OF
...
300 LABEL 29: ERRSTR$ = "Formula Syntax error"
310 LABEL 30: ERRSTR$ = "Formula runtime exhausted"
320 LABEL 31: ERRSTR$ = "Formula compile error"
...
410 IF MASTER% <> 0 THEN
420 IF (ERRNO% = 29) OR (ERRNO% = 31) THEN
430 FORMEL$=SD.SysAxCoupleCmd[AXIS%].Formula
440 ERRSTR$="SAC (" +STR$ ("##",AXIS%)+" ) "+MID$ (FORMEL$,1,MASTER%)+ "^
      "+ERRSTR$
...
510 IF ERRNO% <> 0 THEN
520 SETERR (ERRSTR$)
...
```

Das Übernahmeprogramm VALIDATE erzeugt den Teileprogrammfehler "SAC(5) 10.0#^ Formula Syntax error".

Basiswertebereich der Formelverarbeitung

Für alle mathematischen Funktionen und Operationen ist der erlaubte Wertebereich auf die Intervalle [-9.999999999E99 .. -1.0E-99], [1.0E-99 .. 9.999999999E99] und den Wert 0 beschränkt. Dieser Bereich wird im Folgenden als **Basiswertebereich** [BW] bezeichnet.

Diagnose und Laufzeitfehler der Formelverarbeitung

Eine aktivierte Formel liefert Aufgrund der Eingangsdaten und Rechenoperationen in bestimmten Fehlersituationen kein verwendbares Ergebnis. In diesem Fall kann die Kopplung mit einer definierten Abbruchbehandlung beendet werden. Formelverarbeitungsfehler zeigt die Systemachskopplung im Systemdatum **SysAxCoupleSta[axis].FormulaDiag** an:

0 -	kein Fehler
1 -	Nulldivision Eine Nulldivision ist aufgetreten. Die Formelverarbeitung wird mit dem Operationsergebnis Null fortgesetzt.
2 -	Überlauffehler Der zulässige Basiswertebereich [BW] für mathematische Operationen und Funktionen wurde überschritten. Die Formelverarbeitung wird mit dem überschrittenen Grenzwert (+/-9.999999999E99) als Operationsergebnis fortgesetzt.

- 3 -** Definitionsbereichsverletzung
Die mathematische Operation ist für das Argument nicht definiert.
Beispiel: "...ASIN(3.0)...".
Die Formelverarbeitung wird mit dem funktionsspezifischen Ersatzergebnis fortgesetzt.
- 4 -** Unterlauffehler
Das Ergebnis einer Multiplikation oder Potenzierung ist ungleich Null und betragsmäßig kleiner als der erlaubte Basiswertebereich:
 $0 < | \text{Ausdruck1} * \text{Ausdruck2} | < 1.0\text{E-}99$
Die Formelverarbeitung wird mit dem Operationsergebnis Null fortgesetzt.
- 5..9 -** Formelinterpretationsfehler
Die angegebene Formel kann nicht vollständig abgearbeitet werden. Das Formelergebnis wird zu Null gesetzt. Diese Fehler deuten auf ein internes Problem hin, das dem Kundendienst gemeldet werden sollte.

Die Formelfehler 1, 2, 3, 4 und 5-9 sind für die Abbruchbehandlung mit SysAxCoupleCmd[axis].AbortEnable Bit16-Bit20 selektiv freischaltbar.

Die Operatorposition im Formelstring, an der der Fehler auftrat, liegt im Systemdatum **SysAxCoupleSta[axis].FormulaDiagPos** ab.

Zusätzlich wird bei Laufzeitformelfehlern im Systemdatum SysAxCoupleSta[axis].Error der Fehlercode 7 – FORMULA_ERROR abgelegt.

Zur Überprüfung von Zwischenergebnissen in einer Formel steht die Funktion WATCH im Formelstring zur Verfügung.

Programm:

```
1 SD.SysAxCoupleCmd[X%].Formula="10.0*(COS(L1)+WATCH(L1*SIN(L1)))"
```

Die Funktion legt im Systemdatum SysAxCoupleSta[axis].FormulaWatch das Ergebnis des übergebenen Ausdrucks ab und liefert das Ergebnis für die Weiterverrechnung zurück. Im Beispiel wird das Ergebnis des Ausdrucks „L1*SIN(L1)“ in FormulaWatch abgelegt.

Operanden der Formelverarbeitung

In Formeln der Systemachskopplung dürfen nur numerische Konstanten, Echtzeitvariablen und die von der Systemachskopplung zur Verfügung gestellten Operanden und Systemkonstanten verwendet werden:

- Leitwerte (AC,L1,..L5)
- Leitachsgeschwindigkeiten (V1,...,V5)
- Letzte Folgeachssollposition (SPOS)
- Systemkonstanten (PI, TI)
- Numerische Konstanten (+/-nnn.nnnE+/-nn)
- Echtzeitvariablen (@<var>, SD.<var>)

Der Folgeachsinterpolationswert AC und die Leitwerte Lm der Masteranschlüssen SysAxCoupleCmd[axis].Master[m] liegen in der Einheit [mm] oder [°] vor. Die unbewerteten Leitgeschwindigkeiten Vm der Masterachsen beziehen sich auf die Positionsdivergenz pro Interpolationstakt.

Die folgende Tabelle zeigt die möglichen Formeloperanden:

Operand	Bedeutung	Einheit
AC	Folgeachselinterpolationswert (Achskoordinate); bei gesperrtem Interpolationswert (SysAxCmd[axis].IpoEnable = FALSE) liefert AC=0	[mm,°]
L1 ... L5	Leitwert der Leitachse m: $L_m = Tab_m \left(Pos_m + Shift_m \right) \cdot \frac{N\_Output_m}{N\_Input_m} + Offset_m$ Leitachsen (SysAxCmd[axis].Master[m].Enable = FALSE) liefern Lm=0	[mm,°]
PI	Kreiszahl π	-
SPOS	Folgeachs _{Sollposition} aus vorherigem Ipo-Takt	[mm,°]
TI	NC-Zykluszeit	[s]
V1 ... V5	Geschwindigkeit der Leitachse m: $V_m = \text{Leitachssollposition}_m \cdot T_{ipo} - \text{Leitachssollposition}_m \cdot T_{ipo} - 1$	$\left[\frac{mm,^\circ}{T_{ipo}} \right]$
<Konstante>	Integer- oder doppelt genaue Realkonstante im Basiswertebereich Beispiele: -1, 5, 1.4142135623731, -1.0E-50	-
<Echtzeitvariable>	Zulässige Echtzeitvariablen sind: • permanente Variablen (@<name>) vom Typ – INT – REAL – DOUBLE • Systemdaten (SD.<name>) vom Typ – Boolean_t – Byte_t, UnsignedByte_t – Short_t, UnsignedShort_t – Int_t, UnsignedInt_t – Float_t – Double_t Feldvariablen sind mit Konstanten zu indizieren Beispiele: @10, @Offset, @PermArray(1,3) SD.PlcFA.Offset, SD.SysAxCoupleSta[2].ActPos	-

Tab. 14-8: Formeloperanden



In der Formelkopplung verwendete Echtzeitvariablen dürfen nur synchron zur Interpolation, z.B. durch eine FA-Task (siehe Dokumentation "MTX Funktionsbeschreibung - Sonderfunktionen", Kap. "Schnelle Aktionen"), beschrieben werden! Bei Nichtbeachtung können sporadisch ungültige Variablenwerte gelesen werden!

Verwenden Sie ausschließlich Variablen, die interpolationssynchron verändert werden.

Formeloperationen

Formeln der Systemachskopplung erlauben:

- Grundrechenarten ("+", "-", "*", "/", MOD)
- mathematische Funktionen (z. B. SIN, SQRT)
- Vergleiche ("=", "<>", "<", ">", "<=", ">=")
- logische Operatoren (NOT, AND, OR)
- bedingte Ausdrücke (Bedingung ? Ausdruck1 : Ausdruck2)

Aufeinander folgende Operationen unterliegen Vorrangigkeitsregeln (z. B. Punkt- vor Strichrechnung), soweit diese nicht durch Klammerung außer Kraft gesetzt werden:

Operator	Rang	Bindung bei gleichem Rang	Beispiel	Ausführung
(Ausdruck)	1	-	(1.0 - L1) * 3	(1.0 - L1) * 3
"-" Vorzeichen	2	links	-- -3	-(-(-3))
"*", "/", MOD	3	links	1.0 * 20 MOD 2	(1.0 * 20) MOD 2
"+", "-"	4	links	L2 + L3 - L5	(L2 + L3) - L5
Vergleich "=", "<>", "<", ">", "<=", ">="	5	nicht erlaubt	1 < 2 < 3	Syntaxfehler
NOT	6	rechts	2 OR NOT NOT 1	2 OR NOT(NOT 1)
AND	7	links	0 AND 1 AND 2	(0 AND 1) AND 2
OR	8	links	0 OR 1 OR 2	(0 OR 1) OR 2
"?", ":", "."	9	rechts	1?2:3?4:5	1?2:(3?4:5)

Tab. 14-9: Formeloperatorrangfolge

Mathematische Berechnungen

Im Formelausdruck werden Operanden durch mathematische Operatoren und Funktionen unter Berücksichtigung der Vorrangigkeitsregeln verknüpft. Die Operanden Op sowie Op1 und Op2 in den nachfolgenden Tabellen können hierbei einfache Operanden (Konstanten, Leitwerte) oder auch zusammengesetzte Ausdrücke darstellen. Der Basiswertebereich [BW] ist für alle Berechnungen verbindlich. MAX bezeichnet das Betragsmaximum (9.999999999999999E99) und MIN das Betragsminimum (1.0E-99) im Basiswertebereich [BW]. Für einige Operationen gilt der Ganzzahlbereich von -2147483647 (MININT) .. 2147483647 (MAXINT).

Operatoren zur Grundarithmetik:

Syntax	Bedeutung	Definitionsbe- reich	Ergebniswerte- bereich	Diagnose: Ersatzergebnis
Op1 + Op2 Op1 – Op2	Addition Subtraktion	Op1 und Op2 im [BW]	[BW]	Überlauf : +/- MAX
Op1 * Op2	Multiplikation	Op1 und Op2 im [BW]	[BW]	Überlauf : +/- MAX Unterlauf: 0
Op1 / Op2	Division	Op1 im [BW] Op2 im [BW] ohne Null	[BW]	Überlauf : +/- MAX Nulldivision: 0
Op1 MOD Op2	Modulo	Op1 im [BW] Op2 im [BW] ohne Null	[BW]	Nulldivision: 0

Tab. 14-10: Formelgrundarithmetik

Betrags-, Ganzzahl und Signumfunktion:

Syntax	Bedeutung	Definitionsbe- reich	Ergebniswerte- bereich	Diagnose: Ersatzergebnis
ABS(Op)	Op	Op im [BW]	[BW]	-
INT(Op)	Ganzzahl(Op)	Op in [MININT.. MAXINT]	[MININT.. MAXINT]	Überlauf Op < MININT: MININT Op > MAXINT: MAXINT
SGN(Op)	Signum Op < 0 : -1 Op = 0 : 0 Op > 0 : 1	Op im [BW]	[-1,0,1]	-

Tab. 14-11: Formelbetrag und Ganzzahlarithmetik

Für die Normierung von Vorschüben, Drehzahlen, und Beschleunigungen in die Interpolations-Taktzeit stehen die Funktionen FEED, SPEED und ACC zur Verfügung. Längenangaben in Zoll können mit INCH in die Einheit mm gewandelt werden.

Normierungsfunktionen:

Syntax	Bedeutung	Definitionsbe- reich	Ergebniswerte- bereich	Diagnose: Ersatzergebnis
ACC(Op)	Wandelt Be- schleunigung von m/s^2 in mm/IPO-Takt^2	Op im [BW]	[BW]	-
FEED(Op)	Wandelt Vor- schub von mm/min in mm/IPO-Takt	Op im [BW]	[BW]	Überlauf : +/- MAX
INCH(Op)	Wandelt inch in mm	Op im [BW]	[BW]	Überlauf : +/- MAX
SPEED(Op)	Wandelt Dreh- zahl von U/min in $^\circ/\text{IPO-Takt}$	Op im [BW]	[BW]	Überlauf : +/- MAX

Tab. 14-12: Formelnormierungsfunktionen

Die trigonometrischen Funktionen SIN, COS und TAN verarbeiten das Winkelargument in Grad. ASIN, ACOS und ATAN liefern entsprechend den Winkel in Grad.

Trigonometrische Funktionen

Syntax	Bedeutung	Definitionsbe- reich	Ergebniswerte- bereich	Diagnose: Ersatzergebnis
ACOS(Op)	Arcuscosinus	Op in [-1..+1]	[0..180]	Undefiniert Op < -1 : 0 Op > 1 : 180
ASIN(Op)	Arcussinus	Op in [-1..+1]	[-90..90]	Undefiniert Op < -1 : -90 Op > 1 : 90
ATAN(Op)	Arcustangens	Op im [BW]	-90]..[90	-
COS(Op)	Cosinus	Op im [BW]	[-1..1]	-
SIN(Op)	Sinus	Op im [BW]	[-1..1]	-
TAN(Op)	Tangens	Op Periodenabbild in -90]..[90	[BW]	Undefiniert Op <= 90 : TAN(-MAXPHI) Op >= 90 : TAN(MAXPHI) mit MAXPHI= 90 - 1.0E-14

Tab. 14-13: Trigonometrische Funktionen

Trigonometrische Hyperbel- und Areafunktionen

Syntax	Bedeutung	Definitionsbe- reich	Ergebniswerte- bereich	Diagnose: Ersatzergebnis
ACOSH(Op)	Areacsinus Hyperbolicus	Op in [1..Max]	[BW]	Undefiniert Op < 1 : 0
ASINH(Op)	Areasinus Hy- perbolicus	Op im [BW]	[BW]	-
ATANH(Op)	Areatangens Hyperbolicus	Op in -1]..[1	[BW]	Undefiniert Op <= -1 : ATANH(- MAXP) Op >= 1 : ATANH(MAXP) mit MAXP= 1.0 - 1.0E-16
COSH(Op)	Cosinus Hyper- bolicus	Op in [-230..230]	[1..MAX]	Überlauf : COSH(230)
SINH(Op)	Sinus Hyperbo- licus	Op in [-230..230]	[BW]	Überlauf Op < -230 : SINH(-230) Op > 230 : SINH(230)
TANH(Op)	Tangens Hy- perbolicus	Op in [BW]	-1]..[1	-

Tab. 14-14: Hyperbelfunktionen

Funktionen zur Polarkoordinatenbestimmung

Zur Umrechnung von kartesischen Koordinaten in Polarkoordinaten können ATAN2 und DIST verwendet werden:

Syntax	Bedeutung	Definitionsbe- reich	Ergebniswerte- bereich	Diagnose: Ersatzergebnis
ATAN2(Op1, Op2)	Koordinaten- winkel $\text{atan2}(y,x)$	Op1 und Op2 im [BW]	[-180..+180]	-
DIST(Op1, Op2)	Hypotenuse $\sqrt{(\text{Op1}^2 + \text{Op2}^2)}$	Op1 und Op2 im [BW]	[BW]	-

Tab. 14-15: Polarkoordinatenbestimmung

Potenzfunktionen, Wurzelfunktionen und Logarithmen:

Syntax	Bedeutung	Definitionsbe- reich	Ergebniswerte- bereich	Diagnose: Ersatzergebnis
CBRT(Op)	Kubikwurzel $\sqrt[3]{Op}$	Op im [BW]	[BW]	-
CUB(Op)	Op^3	Op im [BW]	[BW]	Überlauf : +/- MAX Unterlauf: 0
EXP(Op)	e^{Op}	Op in [-MAX.. 230]	[0..Max]	Überlauf : EXP(230)
EXP(Op1, Op2)	$Op1^{Op2}$	1. $Op1 < 0$, $Op2$ aus N 2. $Op1$ oder $Op2 < 0$ 3. $ \ln(Op1) * Op2 $ <230	[BW]	Undefiniert 0^0 : 0 $-z^{n.nnn}$: 0 (complex) Überlauf : +/- MAX
LN(Op)	Logarithmus zur Basis e	Op in [MIN..MAX]	[-227.956.. 230.259]	Undefiniert Op < MIN : LN(MIN)
SQR(Op)	Op^2	Op in [BW]	[BW]	Überlauf : MAX Unterlauf: 0
SQRT(Op)	$\sqrt{Op}$	Op in [0..MAX]	[0..MAX]	Undefiniert Op < 0 : 0

Tab. 14-16: Arithmetikfunktionen

Bedingte Ausdrücke

Bedingte Ausdrücke dienen dazu, Definitionsbereiche von mathematischen Funktionen sicher zu stellen oder bereichsabhängige Kopplungen zu realisieren. Die Bedingung kann Vergleiche und logische Operationen enthalten.

Beispiel:

Die Endlosachse C [0°..360°] soll die Linearachse X über den Bereich [0..100mm] im Intervall 0°..180° mit einer Sinus²-Funktion auslenken:

Programm:

```

1 SD.SysAxCoupleCmd[X%].Master[1].AxIndex = C%
...
1 SD.SysAxCoupleCmd[X%].Formula="L1>=0 AND L1<=180?SQR(SIN(L1))*100:0.0"
...
```

Außerhalb des Bereichs wird der Leitwert 0.0 geliefert:

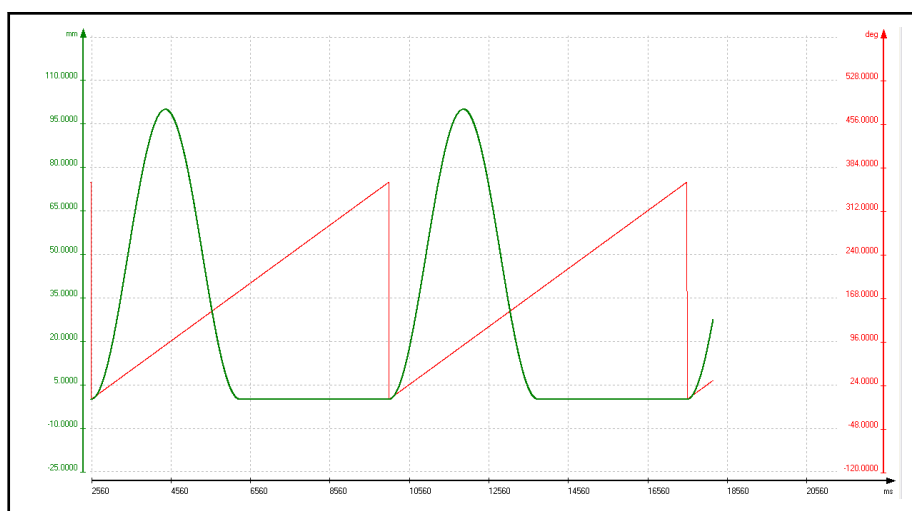


Abb. 14-31: Bedingte Formelkopplung

Die nachfolgenden Tabellen zeigen die möglichen Vergleichsrelationen und logische Operatoren:

Relationen:

Syntax	Bedeutung	Definitionsbe- reich	Ergebniswerte- bereich	Diagnose: Ersatzergebnis
$Op1 = Op2$	prüft Gleichheit $ Op2-Op1 < 1E-13 : 1$ $ Op2-Op1 \geq 1E-13 : 0$	$Op1$ und $Op2$ im [BW]	0,1	-
$Op1 \neq Op2$	prüft Ungleich- heit $ Op2-Op1 \geq 1E-13 : 1$ $ Op2-Op1 < 1E-13 : 0$	$Op1$ und $Op2$ im [BW]	0,1	-
$Op1 < Op2$ $Op1 \leq Op2$ $Op1 \geq Op2$ $Op1 > Op2$	$Op1 < Op2$ $Op1 \leq Op2$ $Op1 \geq Op2$ $Op1 > Op2$	$Op1$ und $Op2$ im [BW]	0,1	-

Tab. 14-17: Formelrelationen

Logische Operatoren

Syntax	Bedeutung	Definitionsbe- reich	Ergebniswerte- bereich	Diagnose: Ersatzergebnis
NOT Op	Negiert Aus- druck	OP in [MI- NINT.. MA- XINT]	0,1	Überlauf: 0
Op1 AND Op2	Logisch UND: Op1 >= 1 und Op2 >= 1 : 1 sonst : 0	Op1 und Op2 [MININT.. MA- XINT]	0,1	Überlauf MA- XINT begrenzt Op1 und Op2
Op1 OR Op2	Logisch ODER: Op1 >= 1 oder Op2 >= 1 : 1 sonst : 0	Op1 und Op2 [MININT.. MA- XINT]	0,1	Überlauf MA- XINT begrenzt Op1 und Op2

Tab. 14-18: Logische Operatoren

Logische Ausdrücke stellen insbesondere die Bedingung der "?" - ":" - Operation.

Bedingte Ausdrücke

Syntax	Bedeutung	Definitionsbe- reich	Ergebniswerte- bereich	Diagnose: Ersatzergebnis
Op1?Op2:Op3	Op1<>0: Op2 sonst: Op3	Op1,Op2 und Op3 im [BW]	[BW]	-
MIN(Op1,Op2)	Op1<Op2: Op1 sonst: Op2	Op1 und Op2 im [BW]	[BW]	-
MAX(Op1,Op2)	Op1>Op2: Op1 sonst: Op2	Op1 und Op2 im [BW]	[BW]	-

Tab. 14-19: Bedingte Ausdrücke

Mit Hilfe der Formelkopplung können komplexe Interpolationsverbände appliziert werden. Beachten Sie jedoch, dass die Leitachsen durch die Folgeachsen nicht limitiert werden. Überlastungen bzgl. der Geschwindigkeit und Beschleunigung führen immer zum Abkoppeln der Folgachse und damit zur Verletzung der Formelvorgabe. Insbesondere bedingte Ausdrücke können leicht zu Unstetigkeiten des Leitwertes führen. Die Steuerung versucht den Leitwert nach der Abkopplung wieder zu erreichen (RECOUPLING), falls keine Fehlerbehandlung (AbortEnable) aktiviert wurde.

Formelinterpolationsverbände der Systemachskopplung haben keine Bahnsteuerung. Bahngeschwindigkeiten und Bahnbeschleunigungen ergeben sich immer implizit durch die Leitachsenvorgaben und den durch die Kopplungsvorschrift mitgeschleppten Folgeachsen.

Synchronisiermode für
Folgeachse:

Die Ankopplung der Folgeachse an den vorgegebenen Leitwert kann auf verschiedene Arten erfolgen.

Insbesondere bei Moduloachsen ergeben sich grundsätzlich zwei Wege zum Erreichen der Synchronposition:

- Durch das Überhöhen der Folgeachsgeschwindigkeit gegenüber der Leitgeschwindigkeit wird die vorausseilende Synchronposition eingeholt.
- Durch das Abbremsen der Folgeachsgeschwindigkeit wartet die Folgeachse auf die nächste nachteilende Synchronposition.

Die Zeit, die die Folgeachse zum Einholen der Synchronposition benötigt, hängt zusätzlich noch von der Begrenzung der maximalen Geschwindigkeit und Beschleunigung der Folgeachse ab, so dass der kürzeste Weg nicht unbedingt der schnellste Weg ist.

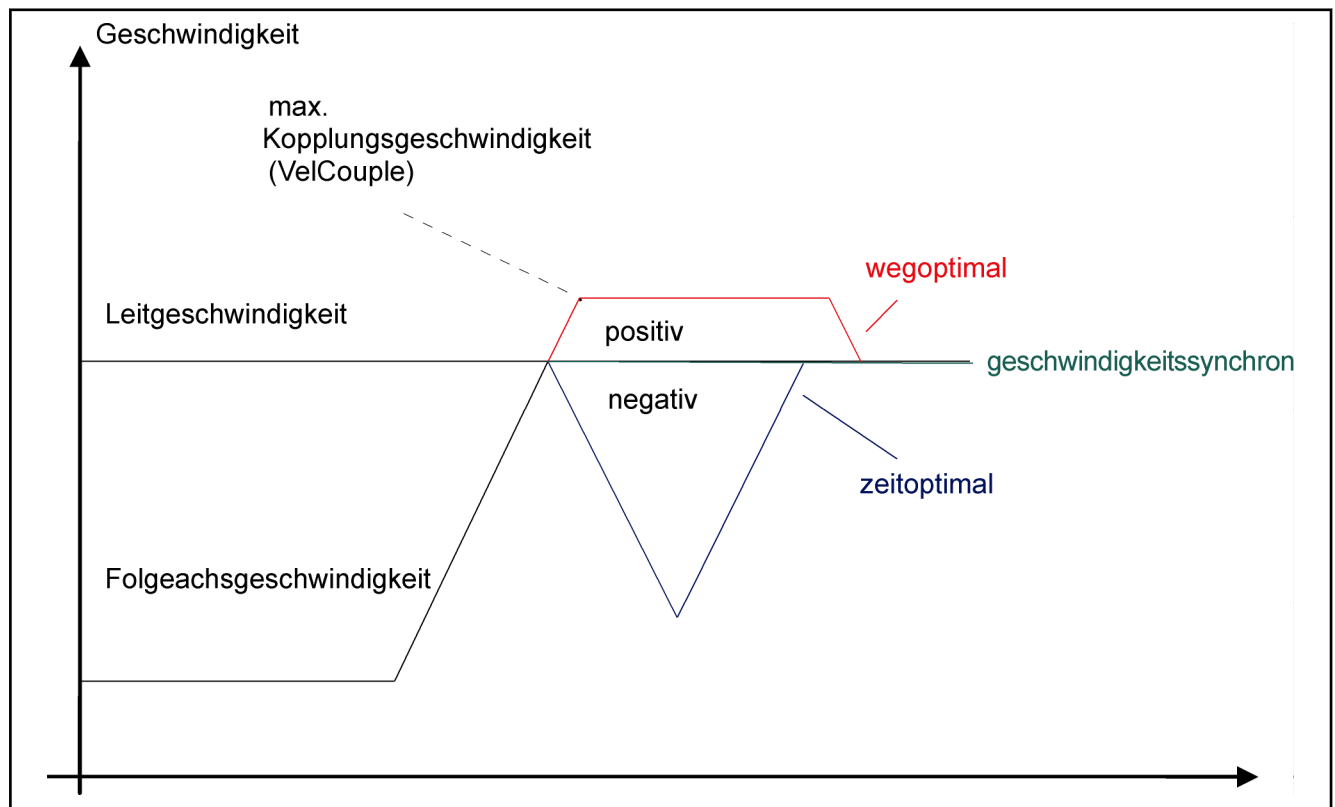


Abb. 14-32: Alternative Ankopplung bei Moduloachsen

Folgende Ankopplungsarten werden unterschieden:

- **Geschwindigkeitsausgleich**
Es wird auf den Lageausgleich verzichtet und stattdessen ein Offset nach Erreichen der Leitgeschwindigkeit gespeichert und mitgeführt. Dies ist die schnellste Ankopplungsart, die außerdem bei Stillstand der Leitachsen und Folgeachse ohne Bewegung an- und abkoppelt.
Mit dem Abschalten der Kopplung (SysAxCoupleCmd[i].OnOff=0) behält die Achse zunächst eine relative Kopplungsverschiebung zwischen Interpolatorposition (Sollposition) und Achsposition (\$Sollposition). Im Kanal kann dieser Offset mit dem NC-Befehl "SACSYNC" abgebaut werden.
- **Wegoptimal**
Es wird der kürzeste Weg zum Lageausgleich gewählt.
(Anmerkung: Ohne Geschwindigkeitsbegrenzung ist dies auch der schnellste Weg.)

und speziell nur für Moduloachsen:

- Zeitoptimal
Es wird der schnellste Weg unter Berücksichtigung gegebener Folgeachsbeschränkungen gewählt.
- Positiv
Der Lageausgleich erfolgt in positiver Geschwindigkeitsrichtung.
- Negativ
Der Lageausgleich erfolgt in negativer Geschwindigkeitsrichtung.



Der Synchronisiermode hat Einfluss auf die Bewegung der Folgeachse beim An- und Abkoppeln, sowie beim Ein- und Ausschalten der Kopplung. Mit der wegoptimalen, zeitoptimalen oder richtungsbezogenen Synchronisierung erfolgt auch bei Stillstand der Achsen eine Ausgleichsbewegung, wenn eine kopplungsbedingte Lagedifferenz besteht!

Wählen Sie den SyncMode "geschwindigkeitssynchron", um die Ausgleichsbewegung bei Stillstand der Achsen zu vermeiden. Übernehmen Sie nach dem Abschalten der Kopplung die Achsposition mit dem NC-Befehl SACSIN in den Kanal der Achse.

Der Synchronisationsmode wird für die jeweilige Folgeachse im Systemdatum **SysAxCoupleCmd[i].SyncMode** vorgegeben.

Dabei bedeutet:

- 0 - kürzester Weg
- 1 - zeitoptimaler Weg
- 2 - geschwindigkeitssynchron
- 3 - positiver Weg
- 4 - negativer Weg

Bei der Geschwindigkeitsankopplung (2 - geschwindigkeitssynchron) ist die Folgeachse bei Synchronität um einen Offset gegenüber dem absoluten Leitwert versetzt. Dieser Offset liegt in **SysAxCoupleSta[i].ActOffset** ab und kann z. B. für eine wiederholte absolute Ankopplung verrechnet werden. Die Aktualisierung des Leitwertversatzes erfolgt mit jeder Kopplungsänderung der betreffenden Folgachse, so dass dieser im Synchronzustand abgespeichert werden muss.



Nach dem Abkoppeln im Geschwindigkeitsmode ist die Synchronlage zunächst verloren. Auch mit der Verrechnung des Leitwertversatzes lässt sich bei Moduloachsen die positionsgenaue Kopplung nicht wiederherstellen. Mit dem Leitwertversatz kann jedoch zahngenau, d. h. an einem beliebigen Zahn, exakt angekoppelt werden, wenn eine Zahnteilung (GearTeeth > 1) vorliegt.

Programm:

```

;-----
; Couple slave axis CL velocity synchronous to
; B master axis
;-----
1 SD.SysAxCoupleCmd[CL%].SyncMode      = 2
1 SD.SysAxCoupleCmd[CL%].Master[1].Enable = TRUE
1 SD.SysAxCoupleCmd[CL%].Master[1].AxIndex = B%
1 SD.SysAxCoupleCmd[CL%].Master[1].Offset = 0
1 SD.SysAxCoupleCmd[CL%].Master[1].N_Input = 21
1 SD.SysAxCoupleCmd[CL%].Master[1].N_Output = -1
;-----

```



```

; Data transfer
;-----
CALL SAC_VALID
;-----
; Wait synchronism
;-----
CALL SAC_LEADSYNC[CL%]
;-----
; Store recoupling reference value offset
;-----
OFFSET! = SD.SysAxCoupleSta[CL%].ActOffset
;-----
; Command and decouple CL axis
;-----
...
1 SD.SysAxCoupleCmd[CL%].Master[1].Enable = FALSE
...
; time-optimum and teeth-exact recoupling
;-----
1 SD.SysAxCoupleCmd[CL%].SyncMode = 1
1 SD.SysAxCoupleCmd[CL%].GearTeeth = 21
1 SD.SysAxCoupleCmd[CL%].Master[1].Enable = TRUE
1 SD.SysAxCoupleCmd[CL%].Master[1].Offset = OFFSET!
...

```

Synchronlaufüberwachung der Folgeachse:

Das Synchronlaufverhalten im Verband ist für die Güte der Bearbeitung entscheidend, da Abweichungen vom Kopplungsverhältnis direkten Einfluss auf die Werkstückgenauigkeit haben. Für die Überwachung wird deshalb der IST-Leitwert mit der IST-Folgeachseposition verglichen. Der IST-Leitwert errechnet sich aus den IST-Positionen der Leitachsen, bewertet durch die Kopplungsbeziehung. Von virtuellen Leitachsen fließt immer der letzte Sollwert in den Vergleich ein. Die aktuelle Differenz des IST-Leitwert zur IST-Folgeachseposition liegt in **SysAxCoupleSta[i].Lag** ab.

Es können zwei Genauigkeitsschranken in **SysAxCoupleCmd[i].PosWin[1..2]** hinterlegt werden. **SysAxCoupleCmd[i].PosWinSel** wählt die aktive Genauigkeitsschranke in folgender Weise aus:

SysAxCoupleCmd[i].PosWinSel	Bedeutung
0	Synchronlaufüberwachung deaktiviert.
1	Synchronlaufüberwachung mit Genauigkeitsschranke 1 gewählt (PosWin[1]).
2	Synchronlaufüberwachung mit Genauigkeitsschranke 2 gewählt (PosWin[2]).
3	Synchronlaufüberwachung mit beiden Genauigkeitsschranken gewählt (PosWin[1] und PosWin[2], mit Hystereseverhalten). Dieser Modus ist z. B. bei stark rauschendem Lage-Istwert sinnvoll.

Tab. 14-20: Auswahl der Genauigkeitsschranke der Synchronlaufüberwachung

Die gewählte Synchronlaufüberwachung wird nur dann aktiv, wenn im Status-element **SysAxCoupleSta[i].State** einer der folgenden Zustände vorliegt:

SysAxCoupleSta[i].State	Bedeutung
1	Erstankopplung (HUNTING)

2	Wiederankopplung (RECOUPLING)
3	Sollwerte gekoppelt (COUPLED)

Tab. 14-21: Voraussetzung für Synchronlaufüberwachung

Ist die Synchronlaufüberwachung aktiv, zeigt die Steuerung das Ergebnis der Synchronlaufüberwachung im Status-Element **SysAxCoupleSta[i].InPos** an (1 = Synchronlauf, 0 = kein Synchronlauf).

Weitergehende Informationen über das Ergebnis der Synchronlaufüberwachung sind im Status-Element **SysAxCoupleSta[i].StateAddOn** zu finden:

Modus der Synchronlaufüberwachung	Ergebnis der Synchronlaufüberwachung	
	SysAxCoupleSta[i].InPos (1 = Synchronlauf, 0 = kein Synchronlauf)	SysAxCoupleSta[i].StateAddOn
nicht aktiv (SysAxCoupleCmd[i].PosWinSel = 0 oder SysAxCoupleSta[i].State ≠ 1, 2, 3)	0	0
eine Genauigkeitsschranke aktiv (SysAxCoupleCmd[i].PosWinSel = 1 oder 2)	1: Differenz zwischen IST-Leitwert und IST-Folgeachseposition liegt innerhalb der gewählten Genauigkeitsschranke 1 oder 2 ($\leq \text{PosWin}[\text{<PosWinSel>}]$) 0: sonst	0: keine Zusatzinformation vorhanden 1000: Genauigkeitsschranke 1 wird eingehalten 2000: Genauigkeitsschranke 2 wird eingehalten
beide Genauigkeitsschranken aktiv (SysAxCoupleCmd[i].PosWinSel = 3)	Hystereseverhalten 1: Differenz zwischen IST-Leitwert und IST-Folgeachseposition liegt innerhalb beider Genauigkeitsschranken 1 und 2 ($\leq \text{PosWin}[1]$ und $\leq \text{PosWin}[2]$) 0: Differenz zwischen IST-Leitwert und IST-Folgeachseposition liegt außerhalb beider Genauigkeitsschranken 1 und 2 ($> \text{PosWin}[1]$ und $> \text{PosWin}[2]$) keine Änderung: Differenz zwischen IST-Leitwert und IST-Folgeachseposition liegt zwischen den Genauigkeitsschranken 1 und 2	0: keine Zusatzinformation vorhanden 1000: Genauigkeitsschranke 1 wird eingehalten 2000: Genauigkeitsschranke 2 wird eingehalten 3000: Genauigkeitsschranken 1 und 2 werden eingehalten

Tab. 14-22: Ergebnis der Synchronlaufüberwachung

14.5.3 Aktivierung/Deaktivierung

Die vorbereiteten Daten aus **SysAxCoupleCmd[i]** werden gleichzeitig mit Hilfe der Kontrollstruktur **SysAxCoupleCtr** an den Interpolator übergeben. Das fol-

gende CPL- Beispielprogramm zeigt die Datenübergabe mit anschließender Fehlerauswertung:

Programm:

SACVALID

```

1 DIM ERRSTR$(100)
WAIT
1 SD.SysAxCoupleCtr.Validate=1
1 ERRNO% = SD.SysAxCoupleCtr.CfgErrNum
1 AXIS% = SD.SysAxCoupleCtr.CfgErrAx
1 MASTER% = SD.SysAxCoupleCtr.CfgErrMaster
1 CASE ERRNO% OF
1 LABEL 0: ERRSTR$ = "NoError"
1 LABEL 1: ERRSTR$ = "Circularity"
1 LABEL 2: ERRSTR$ = "Max Depth"
1 LABEL 3: ERRSTR$ = "Illegal Axis"
1 LABEL 4: ERRSTR$ = "Master enable without AxIndex"
1 LABEL 5: ERRSTR$ = "Master enable without N_INPUT"
1 LABEL 6: ERRSTR$ = "Master enable without N_OUTPUT"
1 LABEL 7: ERRSTR$ = "PosWinSel out of range [0..2]"
1 LABEL 8: ERRSTR$ = "ASUP Number out of range [0,1..8]"
1 LABEL 9: ERRSTR$ = "ASUP Channel invalid [0,1..MAX_CHAN]"
1 LABEL 10: ERRSTR$ = "Invalid AbortAction combination"
1 LABEL 11: ERRSTR$ = "Invalid SyncMode for axis type"
1 LABEL 12: ERRSTR$ = "Invalid VelCouple (NAN)"
1 LABEL 13: ERRSTR$ = "Invalid AccCouple (NAN)"
1 LABEL 14: ERRSTR$ = "Invalid N_Input (NAN)"
1 LABEL 15: ERRSTR$ = "Invalid N_Output (NAN)"
1 LABEL 16: ERRSTR$ = "Invalid Offset (NAN)"
1 LABEL 17: ERRSTR$ = "Invalid RetractPos (NAN)"
1 LABEL 18: ERRSTR$ = "Invalid PosWin[1] (NAN)"
1 LABEL 19: ERRSTR$ = "Invalid PosWin[2] (NAN)"
1 LABEL 20: ERRSTR$ = "Tab not found"
1 LABEL 21: ERRSTR$ = "Invalid SwitchCond"
1 LABEL 22: ERRSTR$ = "Invalid SwitchMin (NAN)"
1 LABEL 23: ERRSTR$ = "Invalid SwitchMax (NAN)"
1 LABEL 24: ERRSTR$ = "Invalid switch range"
1 LABEL 25: ERRSTR$ = "Invalid tab/master modulo range to slave axis"
1 LABEL 26: ERRSTR$ = "Invalid master modulo range to tab"
1 OTHERWISE ERRSTR$ = "Unkown system axis couple configuration error"
1 ENDCASE
1 IF MASTER% = 0 THEN
1 ERRSTR$="SAC("+STR$("##",AXIS%)+") : "+ERRSTR$
1 ELSE
1 ERRSTR$="SAC("+STR$("##",AXIS%)+") "+STR$(".Master( #) :",MASTER%)+ERRSTR$
1 ENDIF
1 IF ERRNO% <> 0 THEN
1 SETERR(ERRSTR$)
1 ENDIF

```

Die Steuerdaten SysAxCoupleCmd der Systemachskopplung können ohne Einfluss auf die aktuelle Kopplung beliebig geändert werden. Die Prüfung und Übernahme aller Änderungen erfolgt erst mit der Gültigstellung in der Kontrollstruktur durch:

SysAxCoupleCtr.Validate = 1

Fehlerhafte Daten werden nicht übernommen und von der Steuerung mit dem Index des SAC in **SysAxCoupleCtr.CfgErrAx** , dem Index des Master **SysAxCoupleCtr.CfgErrMaster** und einem entsprechenden Fehlercode (>0) in **SysAxCoupleCtr.CfgErrNum** quittiert. Die Datenprüfung setzt den Übernahmeauftrag mit **SysAxCoupleCtr.Validate = 0** am Ende zurück.

Die im Interpolator aktuell wirkenden Systemdaten sind nach der erfolgreichen Übernahme auch in der Status-Struktur **SysAxCoupleSta[i].Cmd** sichtbar.

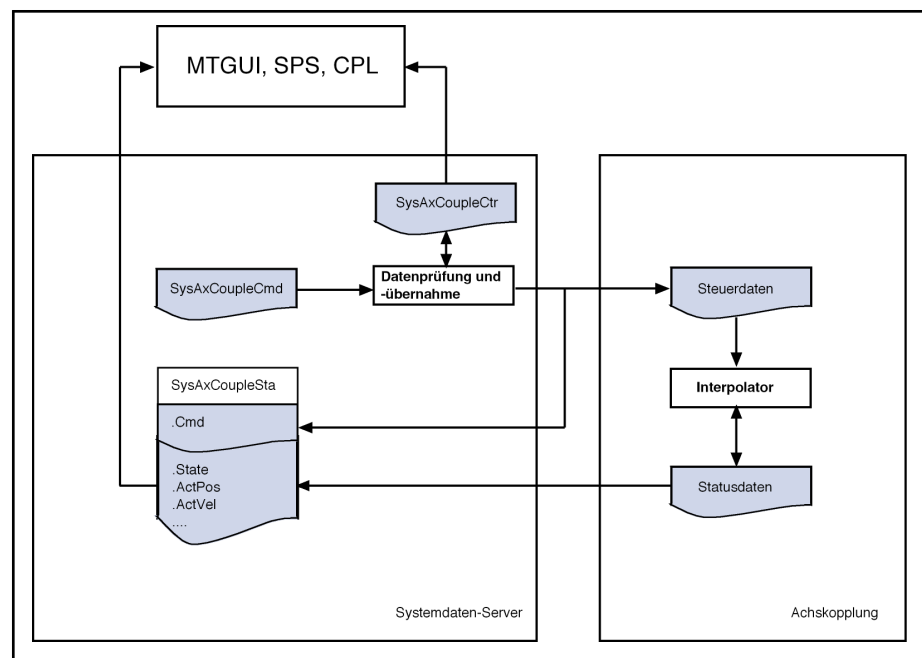


Abb. 14-33: Systemdatenübernahme

**Echtzeitschaltung
einer Leitachse**

Für eine interpolatorgenaue Umschaltung der Kopplungsvorschrift in einem vorgegebenen Positions- oder Geschwindigkeitsbereich kann pro Master eine Übernahmebedingung in **SysAxCoupleCmd[i].Master[m].SwitchCond** vorgegeben werden:

- 0 - unbedingte Umschaltung
- 1 - positionsbedingte Umschaltung
- 2 - geschwindigkeitsbedingte Umschaltung
- 100..131 - extern getriggerte Umschaltung

Die bedingte Umschaltung umfasst die Kopplungsdaten:

- SysAxCoupleCmd[i].Master[m].AxIndex
- SysAxCoupleCmd[i].Master[m].N_Input
- SysAxCoupleCmd[i].Master[m].N_Output
- SysAxCoupleCmd[i].Master[m].Offset
- SysAxCoupleCmd[i].Master[m].Tab

Ohne Umschaltbedingung (SwitchCond=0) werden die Kopplungsdaten mit der Datenübernahme (Validate=1) sofort aktiv.

Für die positions- und geschwindigkeitsbedingte Umschaltung wird der Triggerbereich in

SysAxCoupleCmd[i].Master[m].SwitchMin und

SysAxCoupleCmd[i].Master[m].SwitchMax festgelegt.

Die umschaltrelevante Position oder Geschwindigkeit wird von der vorgegebenen Systemachse in SysAxCoupleCmd[i].Master[m].AxIndex abgeleitet. Triggerpositionen sind auch außerhalb des Modulintervalls der Masterachse möglich, sodass die Umschaltung erst nach N Umdrehungen der Achse erfolgt.

Alternativ kann eine externe Triggerbedingung aus der Bit-Maske SysAxCoupleCtr.ExtCond herangezogen werden. Die Umschaltbedingung ist dann erfüllt, wenn das Bit SwitchCond-100 in der Maske ExtCond z. B. durch die SPS gesetzt wurde.

Der Schalter SysAxCoupleCmd[i].Master[m].Enable wirkt unbedingt. Mit SysAxCoupleCmd[i].Master[m].Enable=0 werden die aktiven Leitachsvorgaben gelöscht und mit SysAxCoupleCmd[i].Master[m].Enable=1 entsprechend der Umschaltbedingung aktiviert.

Die aktiven Echtzeit-Kopplungsdaten liegen im Systemdatum SysAxCoupleSta[i].ActMaster[m] ab.

Eine ausstehende Übernahme der Leitachsumschaltung wird in SysAxCoupleSta[i].State mit PENDING (+20) angezeigt.

Beispielprogramm zur Echtzeitumschaltung:

```

; -----
; function:      SAC_REALTIME
;
;   Description: Real-time table switching in the position
;                range 180°-270° at active system axis coupling
; -----
; --- system axis index of slave axis Z:
1 Z%=3
; --- preparing table switching in the range 180°-270°
1 SD.SysAxCoupleCmd[Z%].Master[1].SwitchCond = 1
1 SD.SysAxCoupleCmd[Z%].Master[1].SwitchMin  = 180.0
1 SD.SysAxCoupleCmd[Z%].Master[1].SwitchMax  = 270.0
1 SD.SysAxCoupleCmd[Z%].Master[1].Tab       = "stroke.fct"
; --- Data transfer
WAIT
1 SD.SysAxCoupleCtr.Validate = 1
; --- Ensure activation in the interpolator
WAIT
; --- synchronization to conditioned switching
SACSYNC (Z=3)

```

14.5.4 Synchronisierung

Der Kopplungsvorgang startet sofort nach der Übernahme der Steuerdaten und findet asynchron zum Ablauf der Teileprogramme in den Kanälen statt. Positionierbewegungen können parallel zum An-/Abkopplungsvorgang durchgeführt werden um Nebenzeiten einzusparen.

Vor dem Bearbeitungseingriff oder dem Aufspannen des Werkstücks muss die Synchronität sichergestellt sein. Dazu kann der Zustand der Systemachskopplung (SysAxCoupleSta[i].State) mit dem NC-Befehl SACSYNC als Wartebedingung abgefragt werden:

Programm:

```

SACSYNC_COUPLED

1 SLAVE% = P1%
; -----
; Wait for synchronous run of the slave axis
; -----
SACSYNC ( [AXP (AXIS%, 3, 0)] )

```

Im Hauptprogramm kann der Zyklus für beliebige Folgeachsen benutzt werden:

```

;Aufspannachse CR synchron zur Gegenspindel CL ?
1 CALL SACSYNC_COUPLED(CR%)
;Werkstück in CL klemmen
...
;Aufspannachse CL Synchron zur Werkzeugachse B?
1 CALL SACSYNC_COUPLED(CL%)
;Bearbeitung mit Wälzfräser
...

```

Kopplungen

Bei Bedarf kann zusätzlich die Synchronlaufüberwachung einbezogen werden.

Der NC-Befehl SACSUNC kann auch mehrere unterschiedliche SAC-Zustände gleichzeitig überwachen:

SACSUNC (TO=200, CL=3, CR=5, A=0)

In einem Zeitfenster von 200 ms wird erwartet, dass die CL-Achse gekoppelt (3), die CR-Achse gestoppt (5) und die A-Achskopplung abgeschaltet (0) ist. Treten alle Zustände innerhalb von 200 ms ein, wird die Bearbeitung fortgesetzt, andernfalls setzt die Steuerung einen Laufzeitfehler ab.

Für abgeschaltete Achsen eliminiert SACSUNC die durch SyncMode=2 (geschwindigkeitssynchron) bestehende Abkopplungsverschiebungen, so dass die Kanalachsen wieder absolut positioniert werden können.

Auf die bedingte Aktivierung der Leitachsendaten kann ebenfalls mit SACSUNC(...<axis|spindel>=<state>...) gewartet werden, da eine ausstehende bedingte Übernahme (<state>+20) die Fortsetzungsbedingung nicht erfüllt.

Beispiel: Kopplungsverschiebung abschalten

;Kopplung ohne Lageausgleich abschalten:

1 SD.SysAxCoupleCmd[XB%].SyncMode = 2 geschwindigkeitssynchron abkoppeln

1 SD.SysAxCoupleCmd[XB%].OnOff = 0 Kopplung abschalten

1 SD.SysAxCoupleCtr.Valid = 1 Aktivierung

SACSUNC ;auf Abschaltung aller Kanalachsen warten und
; Kopplungsverschiebung übernehmen

G1 G90 XB=100 F1000 Achse absolut positionieren

Synchronisation mit Status Synchronlaufüberwachung

Der Befehl SACSUNC kann bei der Angabe des SAC-Zustands auch auf das Erreichen einer Genauigkeitsschranke der Synchronlaufüberwachung warten. Dies ermöglicht z. B. eine frühere Fortsetzung der Programmabarbeitung, wenn die gewünschte Genauigkeitsschranke (INPOS1, INPOS2) erreicht wurde, die eigentliche Kopplung aber noch nicht vorliegt.

Die folgenden Beispiele veranschaulichen die unterschiedliche Wirkung der Wartebedingungen:

SACSUNC (XB=0) ;warte, bis die Kopplung der XB-Achse abgeschaltet ist.

SACSUNC (XB=3) ;warte, bis die XB-Achse Sollwertgekoppelt ist.

SACSUNC (XB=5) ;warte, bis die Kopplung der XB-Achse gestoppt ist.

SACSUNC (XB=1000) ;warte, bis die XB-Achse das INPOS1-Fenster erreicht hat. Das Erreichen des INPOS2-Fensters sowie die anderen Kopplungszustände werden nicht ausgewertet.

SACSUNC (XB=3000) ;warte, bis die XB-Achse sowohl das INPOS1-Fenster als auch das INPOS2-Fenster erreicht hat. Die anderen Kopplungszustände werden nicht ausgewertet.

- SACSYNC (XB=2003) ;warte, bis die XB-Achse sollwertgekoppelt ist und das IN-POS2-Fenster erreicht hat. Das Erreichen des INPOS1-Fensters wird nicht ausgewertet.
- SACSYNC (XB=3003) ;warte, bis die XB-Achse sollwertgekoppelt ist und sowohl das INPOS1-Fenster als auch das INPOS2-Fenster erreicht hat.

14.5.5 Systemachskopplung mit Spindeln

Spindelbetriebsarten

In der Systemachskopplung werden Spindeln wie endlos drehende Rundachsen behandelt. Folgespindeln können nur in der Achsbetriebsart "Lageregelung" (siehe Dokumentation "MTX Funktionsbeschreibung - Grundlagen", Kap. "Umschaltung auf lagegeregelte Spindel") gekoppelt werden.



Bei Leitspindeln führt der Wechsel der Achsbetriebsart (Drehzahlregelung/ Lageregelung) zum Übergang von Istposition zu Sollposition und umgekehrt. Je nach Geschwindigkeit der Leitspindel und Kopplungsvorschrift der Folgespindel/ Folgeachse kann dies zur Abkopplung führen.

Ändern Sie die Achsbetriebsart, wenn möglich, nur im Stillstand. Schalten Sie weder Leit- noch Folgespindeln von Synchronverbänden in Drehzahlregelung.

	Spindel in Drehzahlregelung	Spindel in Lageregelung	C-Achsbetrieb
Leitspindel	Istposition	Sollposition	Sollposition
Folgespindel	nicht erlaubt	Sollposition	Sollposition

Tab. 14-23: Achsbetriebsartabhängige Leit- und Folgespindelposition

Für lagegeregelte Folgespindeln führt die Achsbetriebsartumschaltung und insbesondere der Antriebsparametersatzwechsel in der Regel zum Abschalten der Systemachskopplung. Erlaubt ist nur der Wechsel in den C-Achsbetrieb im Stillstand ohne Positionssprung. Beachten Sie, dass sich mit einer Umschaltung auch Geschwindigkeits- und Beschleunigungsgrenzwerte ändern können.

Besonderheiten und Restriktionen

Die Spindelfunktionen

- Drehzahlbetrieb (M3/M4/M5)
- Positionieren (M19)
- Gewindeschneiden (G33)
- Gewindebohren (G63)
- Konstante Schnittgeschwindigkeit (G96)
- Spindelkopplung (SPCC)

erzeugen und verwenden ausschließlich die Interpolationsposition ohne Berücksichtigung der Systemachskopplungsanteile.

Die Spindelzustandssignale von lagegeregelten Folgespindeln

- "Drehzahl erreicht" (iSp_ProgSpReach)
- "Spindel Positionieren aktiv" (iSp_OrientateAct)
- "Spindel Positionieren beendet" (iSp_OrientateFinish)
- "Spindel Linkslauf aktiv" (iSp_TurnM4)

Kopplungen

- "Drehzahl aktiv" (iSp_TurnCmd)
- "Synchronfenster 1/2" (iSp_Synchr1/2)

beziehen sich bei aktiver Systemachskopplung auf die Interpolationsposition ohne Systemachskopplungsanteile.

Das Signal "Spindel Stopp" (iSp_Stop) zeigt jedoch weiterhin den Antriebszustand (S-0-0331: n_ist=0).

Spindelgrundstellung (qSp_Reset) beendet nicht die Systemachskopplung. Aufgrund der aktiven Leitvorgabe kann die Spindel in Bewegung bleiben.

Wie bei Achsen kann auch eine Folgespindel mit SyncMode=2 ohne Ausgleichsbewegung abgekoppelt werden.

Eine dadurch entstandene Abschaltverschiebung kann bei inaktiver Systemachskopplung mit M19 (Spindel Positionieren) abgebaut werden. Bei aktiver SPCC-Spindelkopplung bleibt die Abschaltverschiebung jedoch bestehen.

Leitspindelinterpolatoren werden durch die Folgespindeln nicht begrenzt. Bei Überschreitung der Maximalgeschwindigkeit oder der Maximalbeschleunigung kommt es zur Abkopplung.

Spindelkopplungsbeispiele

Systemachsgekoppelte Folgespindeln können selbst programmiert werden. Dies ermöglicht überlagerte Drehzahlvorgaben. Im Beispiel soll die Spindel S2 mit einer konstanten Relatvdrehzahl zum Werkstück in Spindel S1 eine Bohrung herstellen:

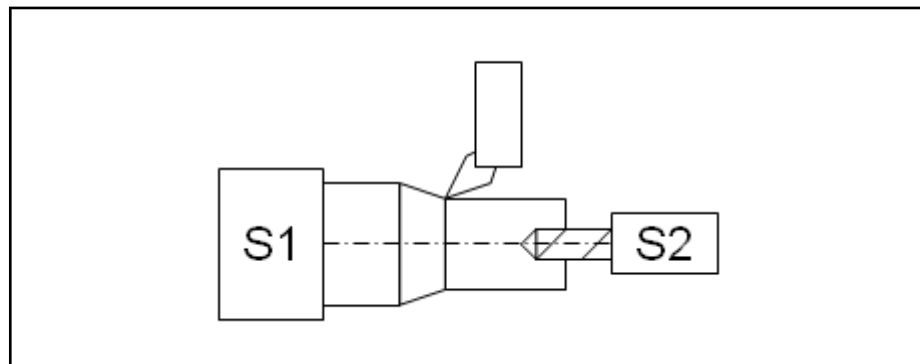


Abb. 14-34: Spindelkopplung

S2 wird dazu kontinuierlich mit der sich ändernden Drehzahl der Spindel S1 überlagert.

Programm:

```

...
0100 S1%=3 : S2%=4
N110 SPM(S2=1)          ;switch slave spindles to position control
...
                        ;coupling on
0140 SD.SysAxCoupleCmd[S2%].OnOff      = 1
                        ;S2 activate interpolation specification
0150 SD.SysAxCoupleCmd[S2%].IpoEnable  = TRUE
                        ;activate S1 override
0160 SD.SysAxCoupleCmd[S2%].Master[1].Enable = TRUE
0170 SD.SysAxCoupleCmd[S2%].Master[1].AxIndex = S1%
                        ;negative feedback S2 = -S1
0180 SD.SysAxCoupleCmd[S2%].Master[1].N_Input  = 1
0190 SD.SysAxCoupleCmd[S2%].Master[1].N_Output = -1
                        ;data application
0200 SD.SysAxCoupleCtr.Validate = 1
N210 SACSUNC(S2=3)        ;wait till override is active
N230 G97 M203 S2=500      ;boring speed 500 rpm
N240 G95 G1 Z=-20 F2      ;feed at 100% S2 potentiometer 1000 mm/min
...
                        ;decoupling
0280 SD.SysAxCoupleCmd[S2%].OnOff = 0
0290 SD.SysAxCoupleCtr.Validate = 1

```



```
N300 SACSUNC(S2=0)      ;waiting for decoupling
...
```

Eine weitere Anwendung der Systemachs-Spindelkopplung ist das Mehrkantdrehen. Werkzeug- und Werkstückspindel sind je nach Geometrie und Anzahl der Schneiden in einem ganzzahligen Verhältnis synchron zu koppeln:

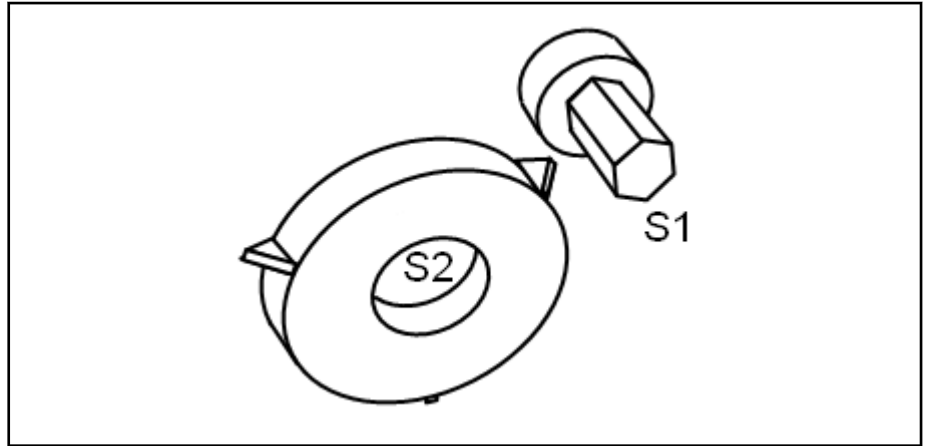


Abb. 14-35: Mehrkantbearbeitung

Im Beispiel ergibt sich das Kopplungsverhältnis von Werkzeug zu Werkstück mit 2:1 zur Erzeugung der Sechskantgeometrie.

Programm:

```
0100 S1%=3 : S2%=4
N110 SPM(S1=1,S2=1) ;switch master and slave spindle to position control
...
                                ;enable coupling
0140 SD.SysAxCoupleCmd[S2%].OnOff      =      1
                                ;no S2 interpolation specification
0150 SD.SysAxCoupleCmd[S2%].IpoEnable  = FALSE
                                ;activate master spindle S1
0160 SD.SysAxCoupleCmd[S2%].Master[1].Enable = TRUE
0170 SD.SysAxCoupleCmd[S2%].Master[1].AxIndex = S1%
                                ;coupling ratio S1:S2
0180 SD.SysAxCoupleCmd[S2%].Master[1].N_Input  =      1
0190 SD.SysAxCoupleCmd[S2%].Master[1].N_Output =      2
0200 SD.SysAxCoupleCtr.Validate        =      1
N210 SACSUNC(S2=3)      ;waiting until coupling is active
...
```

14.5.6 Ausnahmebehandlung

Antriebsgeführtes Stillsetzen und antriebsgeführte Rückzugsbewegung:

Ausnahmebehandlungen, wie Notrückzug oder Stillsetzen des Antriebs, können vom Antrieb oder auch von der NC durchgeführt werden.

Über den Antriebsparameter P-0-0119 "Bestmögliche Stillsetzung" kann ein Antrieb angehalten oder auf eine bestimmte Position gebracht werden.

Beim Rückzug generiert der Antrieb ein Lage-Sollwertprofil, um den gewünschten Verfahrensweg im Fehlerfall abzufahren. Im Fehlerfall wird ein relativer Verfahrenssatz aktiviert, der durch die Parameter "P-0-0055, Rückzugsweg", "P-0-0056, Rückzugsgeschwindigkeit", "P-0-0057, Rückzugsbeschleunigung" und "P-0-0058, Rückzugsruck" definiert ist.

Näheres hierzu ist der Antriebsdokumentation zu entnehmen.

Steuerungsgeführte Reaktionen:

In einer Koppel-Kaskade kann in jedem Kopplungselement ein Fehler auftreten, der zu einer Notreaktion führen soll. Auch externe Ereignisse, wie z. B. von einer Werkzeugbruchüberwachung können eine Notreaktion bewirken. Für jede Achse, die an einer Ausnahmebehandlung teilnehmen soll, ist eine Systemachskopplung zu definieren. Dazu muss das "Abbruch-SAC" einen SAC-Verbandteilnehmer als Leitachse oder eine Achse des zu überwachen-

den Kanals referenzieren (AxIndex=i). Der Leitwert selbst kann abgeschaltet werden (Master[m].Enable=FALSE), so dass keine "echte" Kopplung besteht. Das "Abbruch-SAC" befindet sich nach der Aktivierung (OnOff=1) im Verband und wird bei Störungen automatisch informiert.

Im Koppelverband auftretende Störungen können in jedem SAC-Objekt explizit behandelt oder auch ignoriert werden.

Auftretende Störungen sind in folgende Signalquellen unterteilt:

- Achsfehler
- Kopplungsfehler
- Kanalfehler
- Externer Fehler
- Folgeachs- bzw. Folgeverbandfehler
- Leitachs- bzw. Leitverbandfehler
- Formelfehler
- Verlassen des ausgewählten Synchronlauffensters

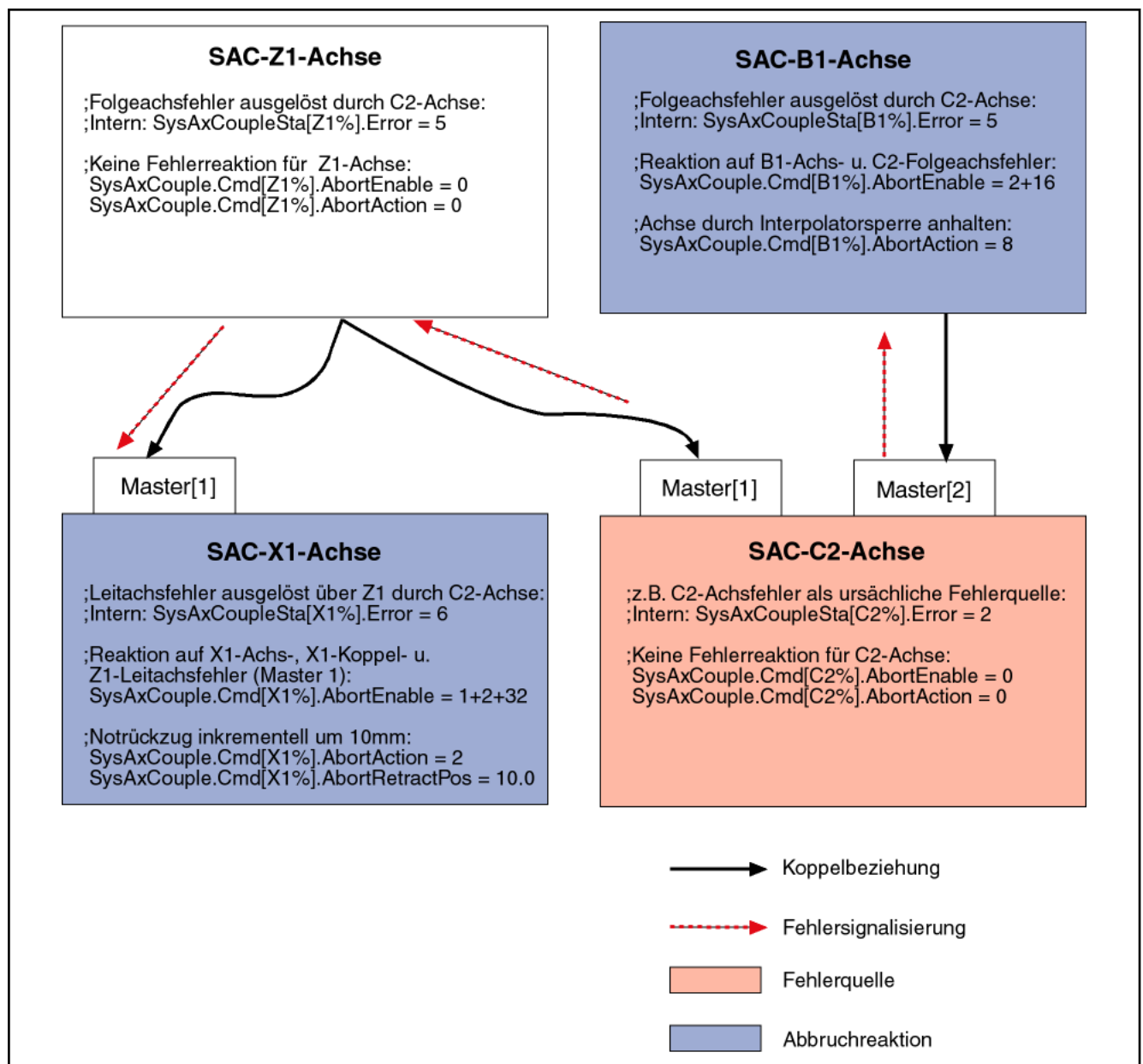


Abb. 14-36: Fehlerverarbeitung in einer Kopplungskaskade

Über das Systemdatum **SysAxCoupleCmd[i].AbortEnable** können die einzelnen Signalquellen freigeschaltet werden:

- 0
Keine Fehlerreaktion durchführen
direkte Fehlerquellen (Error=1..4)
- Bit 0 (# 1)
Reaktion bei Fehler des Antriebs
- Bit 1 (# 2)
Reaktion bei Fehler der Kopplung
- Bit 2 (# 4)
Reaktion bei externem Fehler (ExtError<>0)

Kopplungen

- Bit 3 (# 8)
Reaktion bei Fehler im Interpolationskanal (Kanalfehler)

indirekte Fehlerquellen (Error=5,6)

- Bit 4 (# 16)
Reaktion bei Fehler einer SAC-Folgeachse
- Bit 5 (# 32)
Reaktion bei Fehler der SAC-Leitachse 1
- Bit 6 (# 64)
Reaktion bei Fehler der SAC-Leitachse 2
- Bit 7 (#128)
Reaktion bei Fehler der SAC-Leitachse 3
- Bit 8 (#256)
Reaktion bei Fehler der SAC-Leitachse 4
- Bit 9 (#512)
Reaktion bei Fehler der SAC-Leitachse 5

Formelfehler (Error=7):

- Bit 16 (# 65536)
Reaktion bei Nulldivision (FormulaDiag=1)
- Bit 17 (# 131072)
Reaktion bei Überlauf (FormulaDiag=2)
- Bit 18 (# 262144)
Reaktion bei Definitionsbereichsverletzung (FormulaDiag=3)
- Bit 19 (# 524288)
Reaktion bei Unterlauf (FormulaDiag=4))
- Bit 20 (#1048576)
Reaktion auf Abarbeitungsfehler (FormulaDiag=5..9)

InPos-Fehler (Error=8)

- Bit 24 (#16777216)
Reaktion auf Verletzung der InPos-Bedingung

Mit dem Auftreten der Störung an einem SAC im Koppelverband werden automatisch alle anderen Leit- und Folge-SAC-Objekte informiert. Bei frei geschalteter Signalquelle beginnt die Ausnahmebehandlung noch im gleichen Interpolationstakt.

Ist dies nicht erwünscht kann sie um **SysAxCoupleCmd[i].AbortDelay** ms verzögert werden.

*Nach Ablauf der Verzögerung wird die (bitcodierte) Ausnahmebehandlung aus **SysAxCoupleCmd[i].AbortAction** eingeleitet:*

- (BIT 0)
Rückzug absolut auf die Position **SysAxCoupleCmd[i].AbortRetractPos**
- oder -
- (BIT 1)
Rückzug inkrementell um **SysAxCoupleCmd[i].AbortRetractPos**
- oder -

- (BIT 2)
Kanalfehler mit Störungsart absetzen
- oder -
- (BIT 3-6)
Kopplung normal bzw. modifiziert fortsetzen

Bei einem Achsfehler kann weder die Abbruchart "Rückzug" noch "Kopplung" aktiviert werden. Deshalb wird hier nur ein Kanalfehler mit der Störungsart ausgegeben.

HINWEIS

Falsch parametrierte Abbruchreaktionen können zu Werkzeug- und Maschinenschäden führen!

Testen Sie die Abbruchreaktion ohne Werkzeugeingriff außerhalb des Gefahrenbereichs.

Parallel zu den genannten Reaktionen kann ein asynchrones Programm aus **SysAxCoupleCmd[i].AbortAsupNo = [1..8]** folgendermaßen ausgelöst werden:

- im Achskanal **SysAxCoupleCmd[i].AbortAsupChan = 0**
- oder -
- im Kanal **SysAxCoupleCmd[i].AbortAsupChan = [1..60]**

Für die Installation und Ausführung eines ASUP gilt die allgemeine Funktionsbeschreibungen.

Während der Abbruchbehandlung ist der SAC-Zustand in **SysAxCoupleSta[i].State** mit ABORT (+10) überlagert.

Ablauf der Abbrucharten aus **SysAxCoupleCmd[i].AbortAction:**

1. Rückzug
In **SysAxCoupleSta[i].AbortPos** wird die aktuelle Position zum Zeitpunkt der Unterbrechung abgelegt.
Der Rückzug erfolgt mit maximaler Geschwindigkeit und Beschleunigung auf die in angegebene **SysAxCoupleCmd[i].AbortRetractPos** Position. Dazu wird als einziger und neuer Leitwert die Rückzugsposition vorgegeben. Das SAC-Objekt wird über ABORT-RECOUPLE bis ABORT-COUPLED an die Rückzugsposition interpoliert.
Ein Kanalfehler zeigt im Anschluss das Ende des Rückzugs an. Bei aktivem Abort-ASUP wird der Kanalfehler erst nach ASUP-Ende generiert.
2. Kanalfehler
Ist die SAC-Achse eine Kanalachse, wird ein Fehler mit der Störungsquelle an den Kanal gemeldet. Das SAC-Objekt geht ohne Berücksichtigung der Leitwerte sofort in den Zustand ABORT-ERROR, wo die Achse mit maximaler Verzögerung auf Geschwindigkeit Null abgebremst wird.
3. Kopplung
Diese Abbruchart gilt, wenn keine andere Abbruchart angegeben wurde. Die Achskopplung wird mit den Beziehungen aus den Systemdaten fortgesetzt. Zusätzlich können einzelne oder auch alle Leitwerte über die Bits #3-#6 in **SysAxCoupleCmd[i].AbortAction** gesperrt werden. Die Ausnahme-Kopplung wird mit **SysAxCoupleCmd[i].Reset = 1** abgebrochen. Wie beim Rückzug erfolgt eine Kanal-Fehlermeldung unter Berücksichtigung eines gestarteten ASUP.

Kopplungen

Nach der Fehlerbehandlung befindet sich das SAC-Objekt immer im Zustand ABORT-ERROR, von dem aus es nur nach Fehlerlöschen (Grundstellung) und anschließendem Reset mit **SysAxCoupleCmd[i].Reset = 1** wieder auf inaktiv (SysAxCoupleSta[i].State=0) geschaltet werden kann.

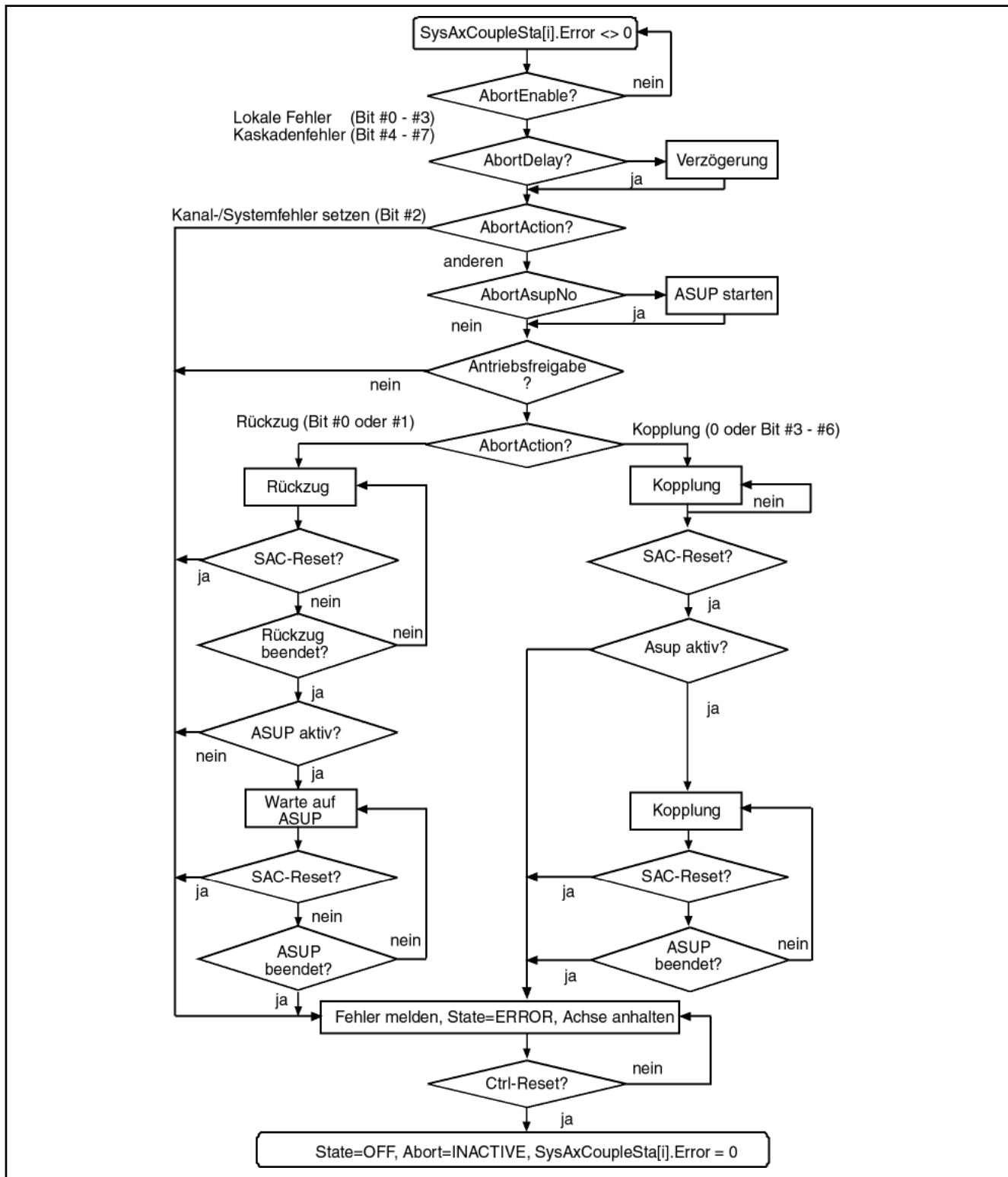


Abb. 14-37: Ablauf der Fehlerverarbeitung



Ein ASUP-Programm darf in einem Kanal nur einmal aktiviert werden. Parametrieren Sie deshalb nur ein SAC-Element im Verband zur ASUP-Aktivierung in einem bestimmten Kanal. Die Fehler anderer SAC im Verband werden als Folge- bzw. Leitachsfehler automatisch untereinander ausgetauscht.

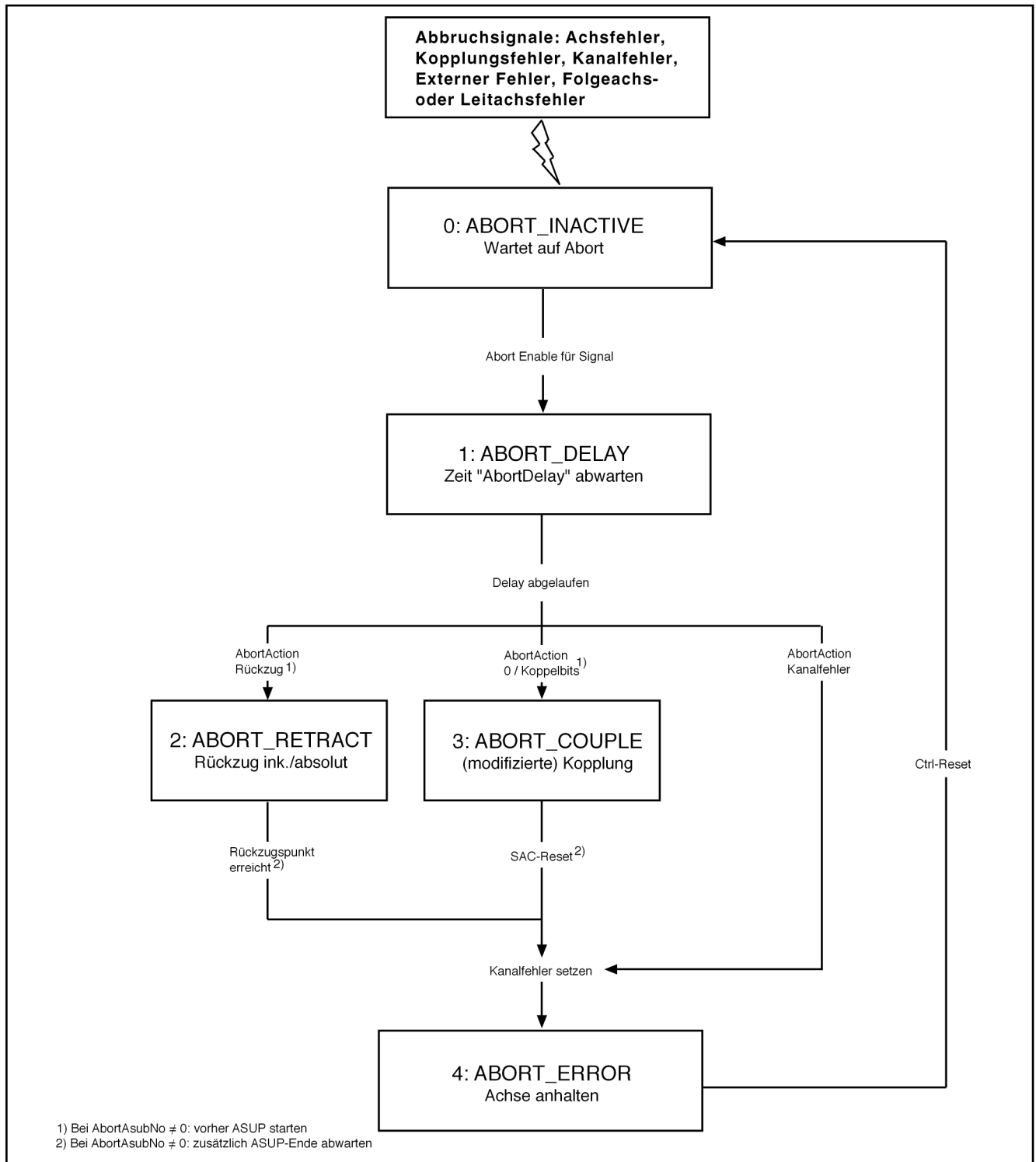


Abb. 14-38: Zustandsdiagramm der Ausnahmebehandlung

0	ABORT_INACTIVE	-	Der Abbruchmechanismus ist inaktiv.
1	ABORT_DELAY	-	Abbruch eingeleitet, Wartezeit aktiv.
2	ABORT_RETRACT	-	Rückzug der Systemachse aktiv.
3	ABORT_COUPLE	-	Systemachse bleibt in (modifizierter) Kopplung.
4	ABORT_ERROR	-	Die Ausnahmebehandlung ist beendet und wartet auf das Rücksetzen durch Grundstellung .

Beispiel einer Fehlerbehandlung:

```

; Coupling relation:
;
; +-----+
; |Z axis |
; +-----+
;
; |
; +-----+
; | V      | V
; +-----+ +-----+
; |YB axis | |VM axis |
; +-----+ +-----+
;
; |
; +-----+
; |
; +-----+
; |
; +-----+
; |XB axis |
; +-----+
;
;
; System data for SAC error treatment:
;
; 1. Enabling abort
; -----
; AbortEnable: (parallel)
; 0 - not enabled
; Bit #Val (variable)
; 0 # 1 - Axis error (AXENA)
; 1 # 2 - Coupling error (COENA)
; 2 # 4 - External error (EXENA)
; 3 # 8 - Channel error (CHENA)
; 4 # 16 - Follower group (SLENA)
; 5 # 32 - M1 group (M1ENA)
; 6 # 64 - M2 group (M2ENA)
; 7 # 128 - M3 group (M3ENA)
;
; Example: Enabling abort for
; axis error, external error and channel error
; 1 SD.SysAxCoupleCmd[1].AbortEnable = 1+4+8
;
; 2. Abort action
; -----
; AbortAction:
; 0 - Coupling continuous
; BIT #Val (alternative ----) (variable)
; -----
; 0 # 1 - Retract absolute (ABSRT)
; -----
; 1 # 2 - Retract relativw (INCRT)
; -----
; 2 # 4 - Set channel error (CHERR)
; -----
; 3 # 8 - IPO Disable (DISIP)
; 4 # 16 - Master1 Disable (DISM1)
; 5 # 32 - Master2 Disable (DISM2)
; 6 # 64 - Master3 Disable (DISM3)
; -----
;
; Example: absolute retract motion when abort:
; 1 SD.SysAxCoupleCmd[1].AbortAction = 1
;
; Axis stop when abort:
; 1 SD.SysAxCoupleCmd[1].AbortAction = 8+16+32+64

```



```

;
;-----
; System axis indices of axes Z, XB, YB and VM:
1 Z%=3 : XB%=4 : YB%=5 : VM%=64
;-----
; AbortEnable Defines
1 AXENA%=1 : COENA%=2 : EXENA%=4
1 CHENA%=8 : SLENA%=16
1 M1ENA%=32 : M2ENA%=64 : M3ENA%=128
;-----
; AbortAction Defines
1 ABSRT%=1 : INCRT%=2 : CHERR%=4
1 DISIP%=8 : DISM1%=16 : DISM2%=32 : DISM3%=64
;-----
; Slave axis XB: Master YB and VM
;-----
;
; After 100ms report channel error in channel XB at YB / VM
; or Z error and start ASUP 1 in channel 3
1 SD.SysAxCoupleCmd[XB%].AbortEnable = M1ENA%+M2ENA%
1 SD.SysAxCoupleCmd[XB%].AbortDelay = 100
1 SD.SysAxCoupleCmd[XB%].AbortRetractPos = 0.0
1 SD.SysAxCoupleCmd[XB%].AbortAction = CHERR%
1 SD.SysAxCoupleCmd[XB%].AbortAsupNo = 1%
1 SD.SysAxCoupleCmd[XB%].AbortAsupChan = 3
;
;-----
; Slave axis YB: Master Z (CH1)
;-----
;
; If external error (SysAxCoupleCmd[YB%].ExtError <> 0,
; YB axis is retracted by 10 mm
1 SD.SysAxCoupleCmd[YB%].AbortEnable = EXENA%
1 SD.SysAxCoupleCmd[YB%].AbortDelay = 0
1 SD.SysAxCoupleCmd[YB%].AbortRetractPos = -10
1 SD.SysAxCoupleCmd[YB%].AbortAction = INCRT%
1 SD.SysAxCoupleCmd[YB%].AbortAsupNo = 0
1 SD.SysAxCoupleCmd[YB%].AbortAsupChan = 0
;
;-----
; Slave axis: Master Z (CH1)
;-----
;
; Immediate stop of axis VM
; if error of master and slave axes (XB, YB, Z)
1 SD.SysAxCoupleCmd[VM%].AbortEnable = SLENA%+M1ENA%
1 SD.SysAxCoupleCmd[VM%].AbortDelay = 0
1 SD.SysAxCoupleCmd[VM%].AbortRetractPos = 64.0
1 SD.SysAxCoupleCmd[VM%].AbortAction = DISIP%+DISM1%
;
;-----
; Slave axis Z: no master, IpoEnable=TRUE
;-----
;
; Immediate absolute retract of Z axis if error
; of slave axes
1 SD.SysAxCoupleCmd[Z%].AbortEnable = SLENA%
1 SD.SysAxCoupleCmd[Z%].AbortDelay = 0
1 SD.SysAxCoupleCmd[Z%].AbortRetractPos = 100.0
1 SD.SysAxCoupleCmd[Z%].AbortAction = ABSRT%
;

```

Der aufgetretene Fehler im SAC wird mit SysAxCoupleSta[i].Error abgelegt. Dieser Fehler wird unabhängig von der Freigabe der Ausnahmebehandlung (AbortEnable) eingetragen und an abhängige Folge- und Leitachsmitglieder weitergeleitet.

Fehler in SysAxCoupleSta[i].Error:

0	NO_ERROR	-	Es liegt kein Fehler vor.
1	UNCOUPLED	-	Die Kopplung kann nicht aufrecht erhalten werden und es liegt keine Wiederankopplungsfreigabe (RecoupleEnable=FALSE) vor.
2	AXIS_ERROR	-	Die Folgeachse hat einen Zustandsklasse-1-Fehler anstehen.
3	CHAN_ERROR	-	Der Kanal der Folgeachse hat einen Fehler anstehen.

4	EXT_ERROR	-	In SysAxCoupleCmd[i].ExtError wurde ein Fehlercode <> 0 eingetragen.
5	SLAVE_ERROR	-	In einer abhängigen Folgeachse bzw. im Folgeachsverband ist ein Fehler aufgetreten.
6	MASTER_ERROR	-	In einer Leitachse bzw. im Leitachsverband ist ein Fehler aufgetreten.
7	FORMULA_ERROR	-	Fehler bei der Berechnung der Formel in: SysAxCoupleCmd[i].Formula Der Detailfehlercode liegt in SysAxCoupleSta[i].FormulaDiag ab.
8	INPOS_VIOLATION	-	Die Synchronlaufüberwachung hat festgestellt, dass die Istposition der Folgeachse das gewählte Genauigkeitsfenster verlassen hat und dass keine Wiederkopplungsfreigabe (RecoupleEnable=FALSE) vorliegt.

Tab. 14-25: Fehler in SysAxCoupleSta[i].Error

Erst mit der Freigabe (AbortEnable) wird die parametrisierte Ausnahmebehandlung für die oben genannten Fehler durchgeführt. Sobald die Störung beseitigt ist, kann der Fehler mit SysAxCoupleCmd[i].Reset gelöscht werden.

14.5.7 Anwendungszyklen

Übergabe der Systemdaten:

Programm:

```

;-----
;Program name: SAC_VALID
;Function:     SAC - system data transfer
;Parameters:   None
;-----
;
1 DIM ERRSTR$(100)
1 DIM FORMEL$(100)
WAIT
1 SD.SysAxCoupleCtr.Validate=1
1 ERRNO% = SD.SysAxCoupleCtr.CfgErrNum
1 AXIS% = SD.SysAxCoupleCtr.CfgErrAx
1 MASTER% = SD.SysAxCoupleCtr.CfgErrMaster
1 IF ERRNO% <> 0 THEN
1 CASE ERRNO% OF
1 LABEL 1: ERRSTR$ = "Circulation (auto-coupling)"
1 LABEL 2: ERRSTR$ = "Nesting too deep"
1 LABEL 3: ERRSTR$ = "Axis cannot become slave axis (e.g. spindle)"
1 LABEL 4: ERRSTR$ = "Master enabled with invalid master index"
1 LABEL 5: ERRSTR$ = "Master enabled without master input revolution"
1 LABEL 6: ERRSTR$ = "Master enabled without master output revolution"
1 LABEL 7: ERRSTR$ = "Monitoring window index invalid"
1 LABEL 8: ERRSTR$ = "ASUP number is invalid"
1 LABEL 9: ERRSTR$ = "ASUP channel does not exist"
1 LABEL 10: ERRSTR$ = "AbortAction contains an invalid combination"
1 LABEL 11: ERRSTR$ = "Syncmode (for axis type) not allowed"
1 LABEL 12: ERRSTR$ = "Invalid coupling velocity"
1 LABEL 13: ERRSTR$ = "Invalid coupling acceleration"
1 LABEL 14: ERRSTR$ = "Invalid input speed"
1 LABEL 15: ERRSTR$ = "Invalid output speed"
1 LABEL 16: ERRSTR$ = "Invalid offset"
1 LABEL 17: ERRSTR$ = "Invalid retract position"
1 LABEL 18: ERRSTR$ = "Invalid InPos window1"
1 LABEL 19: ERRSTR$ = "Invalid InPos window2"
1 LABEL 20: ERRSTR$ = "Table is not loaded"
1 LABEL 21: ERRSTR$ = "Invalid switching condition"
1 LABEL 22: ERRSTR$ = "Invalid lower switching limit"
1 LABEL 23: ERRSTR$ = "Invalid upper switching limit"
1 LABEL 24: ERRSTR$ = "Switching range too small"
1 LABEL 25: ERRSTR$ = "Invalid modulo range: Table to slave axis"
1 LABEL 26: ERRSTR$ = "Invalid modulo range: Master to table"
1 LABEL 27: ERRSTR$ = "Invalid master axis shift"
1 LABEL 28: ERRSTR$ = "Invalid master value mode"
1 LABEL 29: ERRSTR$ = "Formula syntax error"

```

```

1 LABEL 30: ERRSTR$ = "Formula timeout"
1 LABEL 31: ERRSTR$ = "Formula interpretation error"
1 OTHERWISE ERRSTR$ = "Error at system axis coupling data transfer"
1 ENDCASE
1 IF MASTER% = 0 THEN
1 ERRSTR$="SAC (" +STR$ ("##",AXIS%)+" ) : "+ERRSTR$
1 ELSE
1 IF (ERRNO% = 29) OR (ERRNO% = 31) THEN
1 FORMEL$=SD.SysAxCoupleCmd[AXIS%].Formula
1 START%=1
1 IF (MASTER% > 40) THEN
1 START%=MASTER%-40
1 ENDIF
1 ERRSTR$="SAC (" +STR$ ("##",AXIS%)+" )
1 " +MID$ (FORMEL$,START%,MASTER%)+"^ " +ERRSTR$
1 ELSE
1 ERRSTR$="SAC (" +STR$ ("##",AXIS%)+" )
1 " +STR$ (".Master (#) :",MASTER%) +ERRSTR$
1 ENDIF
1 ENDIF
1 SETERR (ERRSTR$)
1 ENDIF

```

Synchronisieren der Abschaltung:

Programm:

```

;-----
; program name: SAC_OFFSYNC
; function: At SAC - State OFF and/or ERROR
; synchronization
; parameters: P1% - Number of system axes
; P2% - 0: All SAC in OFF state
; or ERROR
; 1: All SAC in OFF state
;-----
1 MAXAX% = P1%
1 NOERR% = P2%
1 REPEAT
1 WAIT(,20)
1 SYN% = 0
1 FOR I%=1 TO MAXAX%
1 CASE SD.SysAxCoupleSta[I%].State OF
1 LABEL 0: SYN%=SYN%
1 LABEL 7: SYN%=SYN%+NOERR%
1 LABEL 17: SYN%=SYN%+NOERR%
1 OTHERWISE SYN%=SYN%+1
1 ENDCASE
1 NEXT
1 UNTIL (SYN%=0)

```

Zurücksetzen der Systemachskopplung und löschen der Systemdaten:

Programm:

```

;-----
; program name: SAC_OFF
; function: Switch off all SAC couplings and
; completely delete the SAC system data
; parameters: None
;-----
;
1 DIM AXINFO%(3)
1 VERSION=0
1 ERR_VAR%=MCOFS(45,-1,VERSION,AXINFO%,3)
1 IF ERR_VAR% <> 0 THEN
1 SETERR("Unknown system axis count")
1 ENDIF
1 MAXAX%=AXINFO%(1)
;-----
; Initialize system data, enabling abort
; lock, activate SAC reset
;-----
1 FOR I%=1 TO MAXAX%
1 SD.SysAxCoupleCmd[I%].OnOff = 0
1 SD.SysAxCoupleCmd[I%].IpoEnable = FALSE
1 SD.SysAxCoupleCmd[I%].VelCouple = 0.0
1 SD.SysAxCoupleCmd[I%].AccCouple = 0.0
1 SD.SysAxCoupleCmd[I%].AbortEnable = 0
1 SD.SysAxCoupleCmd[I%].AbortAsupNo = 0
1 SD.SysAxCoupleCmd[I%].AbortAsupChan = 0

```

Kopplungen

```

1  SD.SysAxCoupleCmd[I%].AbortDelay      = 0
1  SD.SysAxCoupleCmd[I%].AbortAction     = 0
1  SD.SysAxCoupleCmd[I%].AbortAction     = 0.0
1  SD.SysAxCoupleCmd[I%].Reset           = 1
1  SD.SysAxCoupleCmd[I%].ExtError        = 0
1  SD.SysAxCoupleCmd[I%].RecoupleEnable  = TRUE
1  SD.SysAxCoupleCmd[I%].GearTeeth       = 0
1  SD.SysAxCoupleCmd[I%].SyncMode        = 2
1  SD.SysAxCoupleCmd[I%].Rigidity        = 0
1  SD.SysAxCoupleCmd[I%].PosWinSel       = 0
1  SD.SysAxCoupleCmd[I%].PosWin[1]       = 0.0
1  SD.SysAxCoupleCmd[I%].PosWin[2]       = 0.0
1  FOR J%=1 TO 3
1      SD.SysAxCoupleCmd[I%].Master[J%].Enable = FALSE
1      SD.SysAxCoupleCmd[I%].Master[J%].AxIndex = 0
1      SD.SysAxCoupleCmd[I%].Master[J%].N_Input = 0
1      SD.SysAxCoupleCmd[I%].Master[J%].N_Output = 0
1      SD.SysAxCoupleCmd[I%].Master[J%].Tab = ""
1      SD.SysAxCoupleCmd[I%].Master[J%].SwitchCond = 0
1      SD.SysAxCoupleCmd[I%].Master[J%].SwitchMin = 0.0
1      SD.SysAxCoupleCmd[I%].Master[J%].SwitchMax = 0.0
1  NEXT
1  NEXT
1  CALL SAC_VALID
1  CALL SAC_OFFSYNC[MAXAX%,0]
;
; -----
; Still active pending exception treatments
; abort
; -----
1  FOR I%=1 TO MAXAX%
1      IF SD.SysAxCoupleSta[I%].Abort = 3 THEN
1          SD.SysAxCoupleCmd[I%].Reset = 1
1      ENDIF
1  NEXT
1  CALL SAC_VALID
1  CALL SAC_OFFSYNC[MAXAX%,0]
;
; Delete pending errors after reset
; -----
1  FOR I%=1 TO MAXAX%
1      SD.SysAxCoupleCmd[I%].Reset = 1
1  NEXT
1  CALL SAC_VALID
1  CALL SAC_OFFSYNC[MAXAX%,1]
;
; Revoke reset
; -----
1  FOR I%=1 TO MAXAX%
1      SD.SysAxCoupleCmd[I%].Reset = 0
1  NEXT
1  CALL SAC_VALID
;
; Apply coupling offset in preparation
; -----
SACSYNC
WAIT

```

Parametrieren einer Folgeachskopplung:*Programm:*

```

; -----
; Program name: SAC_INIT
; Function:      Preparing and transferring a coupling cascade as example:
;
;               Z axis +
;               |
;               BV axis -> B axis +->CL axis ->CR axis
;
; Parameters:    none
; -----
1  CALL SAC_OFF
;
; -----
; System axis index of the Z-, B-, X- and CL axis:
1  Z%=3 : B%=8 : CL%=9 : CR%=10 : BV%=12
; Synchronization modes
1  SYNSHORT%=0 : SYNTIME%=1 : SYNVEL%=2
1  SYNPOS%=3 : SYNNEG%=4
; -----

```

```

;
; -----
; Slave axis B (Master BV)
; -----
1 SD.SysAxCoupleCmd[B%].OnOff = 1
1 SD.SysAxCoupleCmd[B%].IpoEnable = TRUE
1 SD.SysAxCoupleCmd[B%].VelCouple = 5600000.0
1 SD.SysAxCoupleCmd[B%].AccCouple = 120.0
1 SD.SysAxCoupleCmd[B%].GearTeeth = 1
1 SD.SysAxCoupleCmd[B%].SyncMode = SYNVEL%
1 SD.SysAxCoupleCmd[B%].RecoupleEnable = TRUE
; ----- Master 1 (B) active -----
1 SD.SysAxCoupleCmd[B%].Master[1].Enable = FALSE
1 SD.SysAxCoupleCmd[B%].Master[1].AxIndex = BV%
1 SD.SysAxCoupleCmd[B%].Master[1].Offset = 0
1 SD.SysAxCoupleCmd[B%].Master[1].N_Input = 1
1 SD.SysAxCoupleCmd[B%].Master[1].N_Output = 1
; ----- No error reaction -----
1 SD.SysAxCoupleCmd[B%].AbortEnable = 0
1 SD.SysAxCoupleCmd[B%].AbortDelay = 0
1 SD.SysAxCoupleCmd[B%].AbortRetractPos = 0.0
1 SD.SysAxCoupleCmd[B%].AbortAction = 0
1 SD.SysAxCoupleCmd[B%].AbortAsupNo = 0
1 SD.SysAxCoupleCmd[B%].AbortAsupChan = 0
; ----- Synchronous running window CR -----
1 SD.SysAxCoupleCmd[B%].PosWin[1] = 0.02
1 SD.SysAxCoupleCmd[B%].PosWin[2] = 0.01
1 SD.SysAxCoupleCmd[B%].PosWinSel = 1
;
; -----
; Slave axis CL (master B)
; -----
1 SD.SysAxCoupleCmd[CL%].OnOff = 1
1 SD.SysAxCoupleCmd[CL%].IpoEnable = TRUE
1 SD.SysAxCoupleCmd[CL%].VelCouple = 5600000.0
1 SD.SysAxCoupleCmd[CL%].AccCouple = 120.0
1 SD.SysAxCoupleCmd[CL%].GearTeeth = 21
1 SD.SysAxCoupleCmd[CL%].SyncMode = SYNTIME%
1 SD.SysAxCoupleCmd[CL%].RecoupleEnable = TRUE
; ----- Master 1 (B) active -----
1 SD.SysAxCoupleCmd[CL%].Master[1].Enable = FALSE
1 SD.SysAxCoupleCmd[CL%].Master[1].AxIndex = B%
1 SD.SysAxCoupleCmd[CL%].Master[1].Offset = 0
1 SD.SysAxCoupleCmd[CL%].Master[1].N_Input = 21
1 SD.SysAxCoupleCmd[CL%].Master[1].N_Output = -1
; ----- Master 1 (Z) active -----
1 SD.SysAxCoupleCmd[CL%].Master[2].Enable = FALSE
1 SD.SysAxCoupleCmd[CL%].Master[2].AxIndex = Z%
1 SD.SysAxCoupleCmd[CL%].Master[2].Offset = -0.42
1 SD.SysAxCoupleCmd[CL%].Master[2].N_Input = 1
1 SD.SysAxCoupleCmd[CL%].Master[2].N_Output = 0.321
; ----- No error reaction -----
1 SD.SysAxCoupleCmd[CL%].AbortEnable = 0
1 SD.SysAxCoupleCmd[CL%].AbortDelay = 0
1 SD.SysAxCoupleCmd[CL%].AbortRetractPos = 0.0
1 SD.SysAxCoupleCmd[CL%].AbortAction = 0
1 SD.SysAxCoupleCmd[CL%].AbortAsupNo = 0
1 SD.SysAxCoupleCmd[CL%].AbortAsupChan = 0
; ----- Synchronous running window CR -----
1 SD.SysAxCoupleCmd[CL%].PosWin[1] = 0.02
1 SD.SysAxCoupleCmd[CL%].PosWin[2] = 0.01
1 SD.SysAxCoupleCmd[CL%].PosWinSel = 1
;
; -----
; Slave axis CR (master CL)
; -----
1 SD.SysAxCoupleCmd[CR%].OnOff = 1
1 SD.SysAxCoupleCmd[CR%].IpoEnable = TRUE
1 SD.SysAxCoupleCmd[CR%].VelCouple = 5600000.0
1 SD.SysAxCoupleCmd[CR%].AccCouple = 120.0
1 SD.SysAxCoupleCmd[CR%].GearTeeth = 0
1 SD.SysAxCoupleCmd[CR%].SyncMode = SYNTIME%
1 SD.SysAxCoupleCmd[CR%].RecoupleEnable = TRUE
; ----- Master 1 (B) active -----
1 SD.SysAxCoupleCmd[CR%].Master[1].Enable = FALSE
1 SD.SysAxCoupleCmd[CR%].Master[1].AxIndex = CL%
1 SD.SysAxCoupleCmd[CR%].Master[1].Offset = 0
1 SD.SysAxCoupleCmd[CR%].Master[1].N_Input = 1
1 SD.SysAxCoupleCmd[CR%].Master[1].N_Output = 1
; ----- Error reactions CL(none) -----
1 SD.SysAxCoupleCmd[CR%].AbortEnable = 0
1 SD.SysAxCoupleCmd[CR%].AbortDelay = 0
1 SD.SysAxCoupleCmd[CR%].AbortRetractPos = 0.0

```

Kopplungen

```
1 SD.SysAxCoupleCmd[CR%].AbortAction      = 0
1 SD.SysAxCoupleCmd[CR%].AbortAsupNo      = 0
1 SD.SysAxCoupleCmd[CR%].AbortAsupChan    = 0
; ----- Synchronous running window CL -----
1 SD.SysAxCoupleCmd[CR%].PosWin[1]        = 0.02
1 SD.SysAxCoupleCmd[CR%].PosWin[2]        = 0.01
1 SD.SysAxCoupleCmd[CR%].PosWinSel        = 1
;
; -----
; Apply coupling data
; -----
CALL SAC_VALID
```

Zuschaltung einer Leitachskopplung

Programm:

```
; -----
; Program name: SAC_LEADON
; function:
;   couple leading axis P2% to follower axis P1%.
;   require already executed SAC
;   parameterization:
;
;   SD.SysAxCoupleCmd[P1%].OnOff          = 1
;   SD.SysAxCoupleCmd[P1%].N_INPUT        = input speed
;   SD.SysAxCoupleCmd[P1%].N_OUTPUT       = output speed
;   SD.SysAxCoupleCmd[P1%].AxIndex        = system axis index
;
; parameter:   P1% - index of follower axis
;              P2% - index of leading axis
; -----
1 AXIS% = P1%
1 MASTER% = P2%
1 SD.SysAxCoupleCmd[AXIS%].Master[MASTER%].Enable = TRUE
1 CALL SAC_VALID
```

Synchronisieren einer Leitachskopplung

Programm:

```
; -----
; Program name: SAC_LEADSYNC
; function:   wait until follower axis P1%
;             is reference value synchron.
;
; parameter:   P1% - index of follower axis
; -----
1 REPEAT
1   WAIT(,20)
1 UNTIL (SD.SysAxCoupleSta[SLAVE%].State=3)
WAIT
```

14.5.8 Systemdatenüberblick

SAC-Kontrollstruktur: SysAxCoupleCtr

Element	Type	Einheit	Wertebereich	Beschreibung
.Validate	Byte	-	0, 1	0: Übernahme beendet 1: Übernahme aktivieren

.CfgErrNum	Byte	-	-	0: kein Fehler 1: Zirkulation (Selbstkopplung) 2: Verschachtelung zu tief 3: Achse kann nicht Folgeachse werden (z. B. Spindel) 4: Masterfreigabe bei ungültigem Masterachsindex 5: Master-Freigabe ohne Master-Eingangsumdrehung 6: Master-Freigabe ohne Master-Ausgangsumdrehung 7: Überwachungsfensterindex ungültig 8: ASUP-Nummer ist ungültig 9: Der ASUP-Kanal existiert nicht 10: AbortAction enthält eine ungültige Kombination 11: Syncmode (für Achstyp) nicht zulässig 12: ungültige Koppelgeschwindigkeit 13: ungültige Koppelbeschleunigung 14: ungültige Eingangsdrehzahl 15: ungültige Ausgangsdrehzahl 16: ungültiger Offset 17: ungültige Rückzugsposition 18: ungültiges InPos-Fenster1 19: ungültiges InPos-Fenster2 20: Tabelle ist nicht geladen 21: ungültige Umschaltbedingung 22: ungültige untere Umschaltgrenze 23: ungültige obere Umschaltgrenze 24: Umschaltbereich zu klein 25: Modulobereich Tabelle-Folgeachse ungültig 26: Modulobereich Master-Tabelle ungültig 27: Ungültige Leitachsverschiebung 28: Ungültiger Leitwertmode 29: Syntaxfehler im Formelausdruck 30: Formellaufzeitüberschreitung 31: Formelinterpretationsfehler
.CfgErrAx	Unsigned Byte	-	0..250	1-250: Index des fehlerhaften SAC

Kopplungen

.CfgErrMaster	Byte	-	0..250	0 - nicht masterrelevant 1..5 - fehlerhafter Master Formelfehler (CfgErrNum 29,31): 1..250 - Syntaxfehlerposition in Formula
.ExtCond	Unsigned Int	-		Bitmaske zur externen Echtzeitschaltung: Bit 2 ⁰ - aktiviert SwitchCond 100 Bit 2 ¹ - aktiviert SwitchCond 101 ... Bit 2 ³¹ - aktiviert SwitchCond 131

Tab. 14-26: Elemente der Struktur SysAxCoupleCtr

SAC-Steuerstruktur:

SysAxCoupleCmd[i]

Element	Type	Einheit	Wertebereich	Beschreibung
.OnOff	Byte	-	0,1	0: Ausschalten der Kopplung 1: Einschalten der Kopplung
.Reset	Byte	-	0,1	0: kein Rücksetzen 1: Rücksetzen des SAC
.IpoEnable	Boolean	-	TRUE, FALSE	TRUE: Der interpolierte Achswert wird zum Leitwert addiert FALSE: Der Achswert wird nicht berücksichtigt
.RecoupleEnable	Boolean	-	TRUE, FALSE	TRUE: Wiederankopplung erlaubt FALSE: Abreißen der Kopplung löst Fehler aus
.AccCouple	Double	m /s ² 1000°/s ²	0: AxMaxAcc >0.. AxMaxAcc	Maximale Beschleunigung beim An- und Abkoppeln
.VelCouple	Double	mm/min °/min	0: AxMaxVel >0.. AxMaxVel	Maximale Geschwindigkeit beim An- und Abkoppeln
.GearTeeth	Int	-	0.. MaxInt	0: absolute Ankopplung >1: Zahnzahl des Folgeachs Zahnrad
.SyncMode	Short	-	0..4	Ankopplungsmodus: 0: wegoptimal 1: zeitoptimal 2: geschwindigkeitssynchron 3: positive Geschwindigkeitsrichtung 4: negative Geschwindigkeitsrichtung

.Rigidity	Byte	-	0,1	0: weiche Kopplung mit max. AccCouple und VelCouple 1: harte Kopplung mit parametrierter max. Achsbeschleunigung und max. Achsgeschwindigkeit
.PosWinSel	Byte	-	0..3	0: keine Synchronlaufüberwachung 1: Nachlaufüberwachung mit PosWin[1] 2: Nachlaufüberwachung mit PosWin[2] 3: Nachlaufüberwachung mit PosWin[1] und PosWin[2], Hystereseverhalten
.PosWin[i], i = [1,2]	Double	mm,°	>0..MaxPos	Zwei Genauigkeitsfenster für die Synchronlaufbestimmung
.ExtError	Unsigned Short	-	0..65535	Externen Fehler absetzen 0: kein Fehler >0: Externer Fehlercode
.AbortEnable	Int	BITMASK	0 oder Summe Bit 0..24	Reaktionsfreigabe für Fehlerquelle: 0: keine Fehlerbehandlung #1 Bit 0: Fehler des Antriebs #2 Bit 1: Fehler der Achskopplung #4 Bit 2: externer Fehler BOF/SPS/CPL #8 Bit 3: Fehler im Interpolationskanal #16 Bit 4: Fehler in Folgeachs(kaskad)e #32 Bit 5: Fehler in 1. Leitachs(kaskad)e #64 Bit 6: Fehler in 2. Leitachs(kaskad)e #128 Bit 7: Fehler in 3. Leitachs(kaskad)e #256 Bit 8: Fehler in 4. Leitachs(kaskad)e #512 Bit 9: Fehler in 5. Leitachs(kaskad)e # 65536 Bit 16: Formelfehler Nulldivision # 131072 Bit 17: Formelfehler Überlauf # 262144 Bit 18: Formelfehler undefiniert # 524288 Bit 19: Formelfehler Unterlauf #1048576 Bit 20: Formelfehler Abarbeitung #16777216 Bit 24: Verlassen des gewählten Genauigkeitsfensters der Synchronlaufüberwachung
.AbortAsupNo	Byte	-	0..8	0: keine ASUP starten 1..8: asynchrones Unterprogramm im Fehlerfall starten
.AbortAsupChan	Byte	-	0..60	0: Kanal der Achse (bei synchr. Achsen) 1..60: Kanal in dem ASUP gestartet werden soll.
.AbortDelay	Int	ms	0..MaxInt	Abbruchwartezeit in ms bevor eine Abbruchreaktion (AbortAction) ausgelöst wird.

Kopplungen

.AbortAction	Int	BITMASK	0 oder Bit 0..6	Abbruchbehandlung nach Wartezeit: 0: Kopplung läuft weiter - oder - #1 Bit 0: Rückzug absolut nach AbortRetractPos - oder - #2 Bit 1: Rückzug relativ um AbortRetractPos - oder - #4 Bit 2: Kanalfehler absetzen oder eine Kombination der Bits 3-8: #8 Bit 3: Interpolatorleitwert abschalten #16 Bit 4: Masterachsleitwert 1 abschalten #32 Bit 5: Masterachsleitwert 2 abschalten #64 Bit 6: Masterachsleitwert 3 abschalten #128 Bit 7: Masterachsleitwert 4 abschalten #256 Bit 8: Masterachsleitwert 5 abschalten Anmerkung: Mit dem Abschalten aller Leitwerte wird die Folgeachse gestoppt.
.AbortRetractPos	Double	mm,°		Rückzugsposition
.Formula	String	-	Max. 250 Zeichen	Leitwertformel
.Master[m], m=[1..5]				Unterstruktur zur Leitachsdefinition

.Enable	Boolean	-	TRUE, FALSE	TRUE: Leitwert für Folgeachse berücksichtigen FALSE: Leitwert nicht berücksichtigen
.AxIndex	Unsigned Byte	-	0..250	0: Leitachse nicht definiert 1..250: Systemachsindex der Leitachse
.SwitchCond	Short	-	0, 1, 2, 100, ..., 131	0 - unbedingte Umschaltung 1 - positionsbedingte Umschaltung 2 - geschwindigkeitsbedingte Umschaltung 100 - Umschalt-Trigger ExtCond Bit 2 ⁰ ... 131 - Umschalt-Trigger ExtCond Bit 2 ³¹
SwitchMin	Double	mm, ° mm/min, °/min		Untere Schaltposition bzw. Schaltgeschwindigkeit
SwitchMax	Double	mm, ° mm/min, °/min		Obere Schaltposition bzw. Schaltgeschwindigkeit
.Offset	Double	mm, °		Versatz der Folgeachsposition gegenüber der Leitachse.
.N_Input	Double	-		Anzahl der Leitachsumdrehungen
.N_Output	Double	-		Anzahl der Folgeachsumdrehungen
.Shift	Double	mm, °		Leitachsverschiebung (Phasenverschiebung für Tabelle)
.Mode	Unsigned Int	-	0,1	0 - Leitwertmodulintervall ist folgeachsabhängig 1 - Leitwertmodulintervall ist leitachsabhängig
.Tab	String	-	max. 30 Zeichen	Name der Kopplungstabelle ohne Pfad "" - Keine Tabellenkopplung "<Tab>" - Tabellenkopplung mit <Tab> "<Platz>" - Tabellenkopplung mit SysAxCoupleTab[<Platz>]

Tab. 14-27: Elemente der Struktur SysAxCoupleCmd[i]

SAC-Statusstruktur:**SysAxCoupleSta[i]**

Element	Type	Einheit	Wertebereich	Beschreibung
.Cmd	Struct SysAxCoupleCmd	-		Im Interpolator aktive Steuerdaten.

Kopplungen

.State	Byte	-	0..7 11..17 21..27	Zustand der Systemachskopplung: 0: Aus (OFF) 1: Ankoppeln (HUNTING) 2: Wiederankoppeln (RECOUPLING) 3: Synchron (COUPLED) 4: Stoppen (STOPPING) 5: Gestoppt (STOPPED) 6: Entkoppeln (DISCOUPLE) 7: Fehler (ERROR) 1x: Abbruch (ABORT-x) 2x: bedingte Umschaltung (PENDING-x)
.StateAddOn	Int	-	0, 1000, 2000, 3000	Zusatzinformation zum Zustand der Systemachskopplung: 0: keine Zusatzinformation vorhanden 1000: Genauigkeitsschranke 1 der Synchronlaufüberwachung wird eingehalten 2000: Genauigkeitsschranke 2 der Synchronlaufüberwachung wird eingehalten 3000: Beide Genauigkeitsschranken 1 und 2 der Synchronlaufüberwachung werden eingehalten
.Abort	Byte	-	0..4	Zustand der Abbruchbehandlung: 0:- Inaktiv 1: Abbruchverzögerung 2: Rückzug aktiv 3: (modifizierte) Kopplung aktiv 4: Abbruch beendet, Fehlerzustand
.FormulaDiag	Byte	-	0..9	Formellaufzeitfehler: 0: Kein Fehler 1: Nulldivision 2: Überlauf (Ergebnis > 1.0E99) 3: Wertebereich nicht definiert 4: Unterlauf (Ergebnis < 1.0E-99) 5-9: Formelabbruch (interne Fehler)
.FormulaDiagPos	Byte	-	0..250	Operator- oder Operandposition in SysAxCoupleSta[axis].Cmd.Formula, der zum Fehler führte.

.Error		Int	-	0..8	Abbruchfehler bzw. Fehlerquelle: 0: Kein Fehler 1: Kopplung abgerissen 2: Achsfehler 3: Kanalfehler 4: Fehler durch externes Ereignis 5: Fehler in einer Folgachse 6: Fehler in einer Leitachse 7: Fehler in der Formelberechnung 8: Fehler in Synchronlaufüberwachung; die Istposition der Folgeachse hat das gewählte Genauigkeitsfenster verlassen und es liegt keine Wiederankopplungsfreigabe (RecoupleEnable=FALSE) vor.
.InPos		Int	-	0..1	Zustand der Synchronlaufüberwachung: a) PosWinSel = 1 oder 2: InPos = 0, wenn $ IST_{Leitwert} - IST_{Folgeachse} > PosWin[PosWinSel]$ InPos = 1, wenn $ IST_{Leitwert} - IST_{Folgeachse} \leq PosWin[PosWinSel]$ b) PosWinSel = 3: InPos = 0, wenn $ IST_{Leitwert} - IST_{Folgeachse} > PosWin[1]$ und $ IST_{Leitwert} - IST_{Folgeachse} > PosWin[2]$ InPos = 1, wenn $ IST_{Leitwert} - IST_{Folgeachse} \leq PosWin[1]$ und $ IST_{Leitwert} - IST_{Folgeachse} \leq PosWin[2]$ ansonsten behält InPos den bisherigen Wert Anmerkung: Bei PosWinSel = 0 oder SysAxCoupleCmd[i].State = 0 oder > 3 wird InPos immer auf 0 gesetzt.
.Lag		Double	mm, °		Differenz $ IST_{Leitwert} - IST_{Folgeachse} $
.AbortPos		Double	mm, °	-	Position zum Zeitpunkt der letzten Rückzugsauslösung.
.FormulaWatch		Double	-		Formelteilergebnis (WATCH-Funktion)
.ActPos		Double	mm, °	-	Aktuelle Sollposition der Folgeachse
.ActVel		Double	mm/min, °/min	-	Aktuelle Geschwindigkeit der Folgeachse
.ActOffset		Double	mm, °		Aktuelle Lagedifferenz der Folgeachse zum Leitwert.
.ActMaster[m], m=[1..5]					Unterstruktur mit aktiven Master
	.AxIndex	Short	-	0..250	aktive Leitachse: 0: Leitachse nicht definiert 1..250: Systemachsindex der Leitachse

Kopplungen

.Offset	Double	-		aktiver Leitwertversatz.
.N_Input	Double	-		aktive Eingangsdrehzahl
.N_Output	Double	-		aktive Ausgangsdrehzahl
.Shift	Double	mm,°		aktive Leitachsverschiebung
.Tab	String	-	30 Zeichen	aktive Kopplungstabelle

Tab. 14-28: Elemente der Struktur SysAxCoupleSta[i]

SAC-Tabellenverwaltung

SysAxCoupleTab[i]

Element	Type	Einheit	Wertebereich	Beschreibung
.FileName	String	-	100 Zeichen	Quelldateiname der Kopplungstabelle mit Pfad-angabe.
.State	Short	-	0..4	Zustand der Tabelle 0 - nicht belegt 1 - Tabelle wird geladen 2 - Tabelle ist geladen 3 - Tabelle wird benutzt 4 - Tabelle wird entfernt
.FileTime	String	dd:mm:yy hh:mm:ss	18 Zeichen	Zeitstempel der geladenen Kopplungstabellen-datei.
.CacheSize	Int	Bytes		Durch die Tabelle belegter Speicher.

Tab. 14-29: Elemente der Struktur SysAxCoupleTab[i]

14.6 Kanalachskopplung

14.6.1 Beschreibung

Funktion

Die Kanalachskopplung stellt eine feste Beziehung zwischen den Achskoordinaten (ACS) einer Master- und einer Slaveachse her.

Master- und Slaveachsen werden zu einem Achsverband gekoppelt. Jeder Achsverband hat genau eine Masterachse und maximal sieben Slaveachsen. Alle Achsen eines Achsverbandes müssen in einem Kanal liegen. Es können auch mehrere Achsverbände einem Kanal angehören.

Stehen die Slaveachsen beim Aktivieren der Kopplung nicht auf der Koppelposition, werden sie durch eine intern generierte lineare Verfahrbewegung auf diese Position bewegt. Dabei sind der aktive Bahnvorschub und das Override-Poti wirksam.

Grundsätzlich kann die Kanalachskopplung auch mit anderen Kopplungsarten kombiniert werden. Die Reihenfolge der Kopplungen ist:

- Kanalachskopplung
- Systemachskopplung (siehe [Kap. 14.5 "Systemachskopplung" auf Seite 530](#))
- Gantrykopplung (siehe [Kap. 14.1 "Gantry-Achsen" auf Seite 485](#))

In einem Achsverband sind gleichzeitig Parallelachsen, elektronische Getriebe und andere beliebige Koppelbeziehungen möglich.

Man unterscheidet zwischen linearer und beliebiger Kopplung der Achspositionen zueinander.

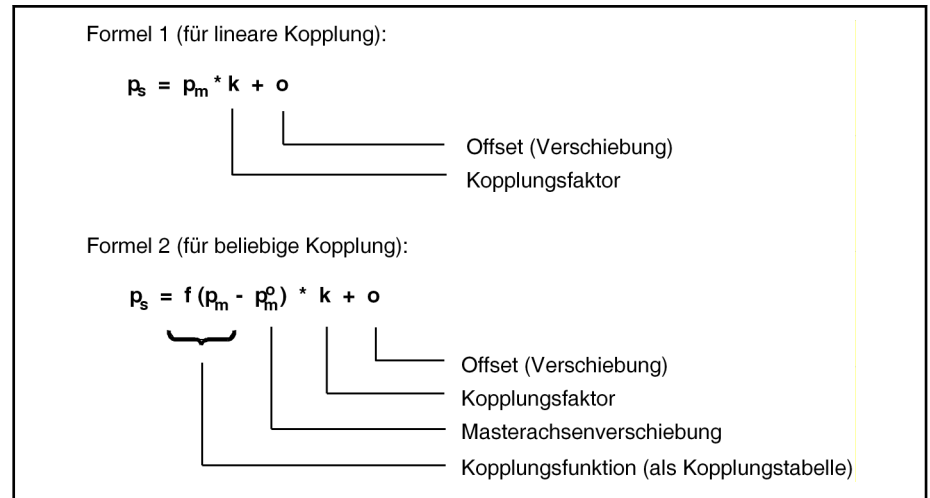


Abb. 14-39: Formeln zur Achskopplung

Die Funktion $f(p)$ liegt dabei als Funktionstabelle (Kopplungstabelle) im Speicher der NC ab.

Aufbau und Wirkungsweise von Kopplungstabellen sind im [Kap. 14.4 "Kopplungstabellen"](#) auf Seite 526 näher erläutert.

Ein Achsverband besteht aus einer Masterachse und einer oder mehreren Slaveachsen. Jede Slaveachse steht in einer eigenen Kopplungsbeziehung nach (Formel 1) oder (Formel 2) zur Masterachse.

Die rechten Seiten der Formeln 1 und 2 stellen von der Masterposition abhängige parametrisierte Leitwerte dar, die in jedem Interpolationstakt als Positionsvorgabe für die Slaveachsen benutzt werden.

Beispiel für Koppelbeziehungen:

- Parallelachsen (parallel angeordnete Bearbeitungstische)
- Elektronisches Getriebe (Achsen, die sich in einem bestimmten Verhältnis zueinander bewegen)

Besonderheiten und Restriktionen

- Ein Mitglied einer Achstransformation kann nicht Slaveachse einer Achskopplung sein.
- Eine Hirthachse kann nicht Slaveachse einer Achskopplung sein.
- Ein Gantry-Slave kann nicht Slaveachse einer Achskopplung sein.
- Eine Slaveachse kann nicht gleichzeitig die Masterachse einer anderen Achskopplung sein.
- Lineare Kopplung:
Ist die Masterachse eine Moduloachse, dann muss die Slaveachse ebenfalls eine Moduloachse sein. Es gilt folgende Einschränkung für den Modulwert:
 $(\text{Modulwert}_{\text{Master}} * \text{Koppelfaktor}) \bmod \text{Modulwert}_{\text{Slave}} = 0$
- Beliebige Kopplung:
Ist die Masterachse eine Moduloachse, dann muss die Kopplungstabelle periodisch sein (Tabelleneintrag #20 = 1). Der Modulwert der Masterachse muss ein Vielfaches der Periode der Tabelle sein.
 $\text{Modulwert}_{\text{Master}} \bmod \text{Modulwert}_{\text{Tabelle}} = 0$

- Eine Nachlaufüberwachung findet nicht statt.

Relevante NC-Funktionen

AxCouple(..)	Kanalachskopplung herstellen/aufheben
CoupleSplineTab(..)	Kopplungstabelle kompilieren

Tab. 14-30: Relevante NC-Funktionen, Kanalachskopplung

Relevante CPL-Funktionen

SD(582,0,m)	Kanalachsnummer m, wenn die Achse m Masterachse einer Achskopplung ist. 0: wenn m keine Masterachse ist.
SD(582,s,0)	Nummer der Masterachse zu der die Achse s eine Slaveachse ist. 0: wenn s keine Slaveachse ist.
SDR(582,s,1)	Programmierte Slaveachsenverschiebung in aktueller Programmiereinheit (mm, inch oder Grad). 0: wenn s keine Slaveachse ist.
SDR(582,s,2)	Programmierter Kopplungsfaktor in aktueller Programmiereinheit. 0: wenn s keine Slaveachse ist.
SDR(582,s,3)	Programmierte Masterachsverschiebung in aktueller Programmiereinheit. 0: wenn s keine Slaveachse ist.
mit: m: Nummer der Masterachse 1..8 s: Nummer der Slaveachse 1..8	

Tab. 14-31: Relevante CPL-Funktionen, Kanalachskopplung

Relevante IF-Signale

iAx_MasterAxIndex_00 .. iAx_MasterAxIndex_06

Der Wert dieser 7 Signale zeigt die binär kodierte Systemachsnummer der Masterachse einer Kanalachskopplung oder einer Gantrykopplung an.

- Ist die Achse nicht gekoppelt, hat die Nummer der Master-Achse den Wert 0.
- Ist die Achse selbst Master, so ist die Nummer der Master-Achse identisch mit der Achsnummer.

Beispiele

N100 AXC(C0,A(),B())

Koppeln mit Abstand 0.

C: Masterachse

A/B: Slaveachsen

Beide Slaveachsen laufen mit "linearer" Kopplung.

Die Defaultwerte Koppelabstand 0.0 und Koppelfaktor 1.0 sind wirksam.

	N100 AXC (C0, A (ACT) , B (ACT, 2))	Koppeln bei den aktuellen Achspositionen. C: Masterachse A/B: Slaveachsen Beide Slaveachsen laufen mit "linearer" Kopplung. A und B werden ohne Anfahrbewegung an C angekoppelt. B wird mit Koppelfaktor 2 betrieben.
	N100 AXC (C0, B (, 0.5, , T_B))	Achsverband erzeugen C: Masterachse B: Slaveachse Slave B läuft mit "beliebiger" Kopplung.
Achskopplung erzeugen	N100 AXC (C0, B ())	B wird als Slaveachse an C angekoppelt. Slaveachse B läuft mit "linearer" Kopplung.
Achskopplung erweitern	N150 AXC (C1, A (4.5, 2))	A wird als Slaveachse hinzugefügt. Slaveachse A läuft mit "linearer" Kopplung.
Achskopplung reduzieren	N200 AXC (C-1, A ())	Die Slaveachse A wird aus dem Achsverband C entfernt.
Achskopplung auflösen	N250 AXC (C-1)	Der komplette Achsverband C wird gelöscht.

14.6.2 Applizieren

Wenn für einen Koppelverband Kopplungstabellen benötigt werden, dann müssen diese auf geeignete Weise erstellt werden.

Weiterführende Informationen zu Kopplungstabellen liefert das [Kap. 14.4 "Kopplungstabellen"](#) auf Seite 526.

14.6.3 Aktivieren

Durch Programmieren der NC-Funktion "AxCouple(...)" wird die Funktion aktiviert (siehe obige Beispiele).

Wenn mit Kopplungstabellen gearbeitet wird, dann kann diese mit der NC-Funktion "CoupleSplineTab(...)" vorab überprüft und kompiliert werden.

14.6.4 Deaktivieren

Die Funktion "AxCouple" ermöglicht das Abschalten einer bestimmten Kopplung oder aller Kopplungen:

- AxCouple(<Masterachsname>, -1) beendet die Kopplung mit der programmierten Masterachse.
- AxCouple() beendet alle Kopplungen.

15 Messfunktionen

15.1 Fahren gegen Messtaster (G75)

15.1.1 Beschreibung

Zusammenfassung

G75 dient zum automatisierten Messen der Werkstück-Position. Anstelle des Werkzeugs wird ein Messtaster eingespannt. Bei einem G75-Satz fährt die NC in die programmierte Richtung, bis der Messtaster gegen das Werkstück stößt. In diesem Moment wird umgehend die Position des Messtasters gespeichert und die Bewegung gestoppt.

Die NC besitzt zwei G75-Funktionen:

- Fahren gegen ersten Messtaster (G75.1 bzw. G75)
- Fahren gegen zweiten Messtaster (G75.2)

Begriffe bei G75

Messtaster:

- sind all diejenigen Sensoren, die die Werkstückposition prüfen, z. B. mechanische Messtaster, Hall-Sensoren, etc.
- kennen nur zwei Zustände: "Werkstück" oder "kein Werkstück"
- formen diese Zustände in ein elektrisches Ausgangssignal um

Fahrachsen:

- werden als Achskoordinaten im G75-Satz programmiert (z. B. bei G75.1 X100 Y200 F1000 sind X und Y die Fahrachsen)
- sollten gleichzeitig Messachse sein - müssen aber nicht

Messachsen:

- können nur solche Sercos Antriebe sein, die selbst die Messtaster-Funktion unterstützen
- werden über einen Maschinenparameter zu einer solchen definiert (MP **EnablProbeFunc** "Messtasterfunktion verfügbar")
- speichern beim Empfang der Messtaster-Flanke den Messwert
- sollten gleichzeitig Fahrachse sein - müssen aber nicht

Der Messwert:

- ist der von der Messachse bei Empfang einer Messtaster-Flanke gespeicherte Wert
- stammt immer von der Quelle, die zuvor festgelegt wurde - mittels S-0-0426 "Signal-Auswahl Messtaster 1" bzw. S-0-0427 "Signal-Auswahl Messtaster 2"
- kennt als Quelle nur den Wert des aktiven Geber-Systems

Was passiert im G75-Satz?

G75 fährt und misst.

G75 startet die Messtaster-Funktion in allen Messachsen des Kanals.

Die Fahrachsen fahren geradlinig mit dem programmierten Vorschub auf den Punkt zu, der sich aus den programmierten Koordinaten ergibt.

Alle Messachsen, die während dessen eine Messtaster-Flanke empfangen, speichern umgehend den gefundenen Messwert.

Gleichzeitig senden sie das Signal "Messtaster-Flanke erkannt" zur NC.

Die NC prüft permanent, ob sie das Signal "Messtaster-Flanke erkannt" von der Messachse mit der Messachse empfängt.

Wenn ja: stoppt die NC sofort alle Fahrachsen, liest die Messwerte aller Messachsen ein und löscht den verbliebenen Restweg ab.

Wenn nein: fahren die Fahrachsen bis exakt auf die programmierten Koordinaten und stoppen dort.

Abschließend fährt die NC mit der nächsten Programmzeile fort.

Warum hat der Messtaster nicht ausgelöst?

Mögliche Ursachen hierfür wären:

- Es wurde über die Achskoordinaten eine zu geringe Suchtiefe zum Werkstück angegeben.
- Der Messtaster hat ausgelöst, aber die Messachsen erhalten keine Messtaster-Flanke; u.U. liegt ein Verdrahtungsfehler vor. - In diesem Fall könnte der Messtaster beschädigt werden.

Was passiert nach dem G75-Satz?

Im Anschluss an einen G75-Satz muss der Anwender:

- prüfen, ob der Messtaster ausgelöst hat oder nicht
- sicherstellen, dass die Messwerte ins Teileprogramm übernommen werden
- dafür sorgen, dass die benötigten Warnungen und Fehlermeldungen generiert werden

Hierzu stehen geeignete CPL-Funktionen zur Verfügung.

SPS-Interface

Die SPS ist in der Lage festzustellen, ob ein G75-Satz aktiv ist oder nicht. Hierfür muss der Anwender ein entsprechendes Bit am "NC-funktionsspezifischen Bit-Interface" vereinbaren. Dieses Bit wird dann für die Zeitdauer des G75.1- (G75-) bzw. G75.2-Satzes aktiv.

Relevante NC-Funktionen

G75.1 (bzw. G75)	Fahren gegen ersten Messtaster
G75.2	Fahren gegen zweiten Messtaster

Tab. 15-1: Relevante NC-Funktionen

Relevante CPL-Funktionen

SD(9)	Liefert die Information, ob der Messtaster ausgelöst hat.
PROBE(), PCSPROBE()	Liefern den Messwert einer (synchrone) Messachse - aber jeweils in unterschiedlicher Form.

Tab. 15-2: Relevante CPL-Funktionen

Relevante Maschinenparameter (MP)

Probe1Edge	Pfad: MEAS / Dr[i] / Probe1Edge Name: "Flanke Messtaster 1" (entspricht Macoda-ID 1003 00011)
EnablProbeFunc	Pfad: MEAS / Dr[i] / EnablProbeFunc Name: "Messtasterfunktion verfügbar" (entspricht Macoda-ID 1003 00012)
Probe2Edge	Pfad: MEAS / Dr[i] / Probe2Edge Name: "Flanke Messtaster 2" (entspricht Macoda-ID 1003 00013)
NcFunc[1..24]	Pfad: PLC / NcFuncBitIf / NcFunc[1..24] Name: "NC-funktionsspezifisches Bit-Interface" (entspricht Macoda-ID 3020 00001/ 3020 00002)

Tab. 15-3: Relevante Maschinenparameter (MP)

15.1.2 Inbetriebnahme Maschinenparameter (MP)

Bestimmen der Messachsen:

Alle Systemachsen, die der Anwender als Messachsen nutzen möchte, muss er über MP **EnablProbeFunc** "Messtasterfunktion verfügbar" zu einer solchen bestimmen, d. h.:

- wird die Systemachse nur für **G75.1** (G75) benötigt, ist in IndraWorks Engineering MEAS / Dr[i] / EnablProbeFunc = "Nur Messtaster 1" zu setzen
- wird die Systemachse sowohl für **G75.1** (G75) als auch für **G75.2** benötigt, ist in IndraWorks Engineering MEAS / Dr[i] / EnablProbeFunc = "Beide Messtaster" zu setzen

Bestimmen der Messtaster-Flanken:

Sobald eine Systemachse zur Messachse bestimmt wurde, muss festgelegt werden, auf welche Messtaster-Flanken sie reagiert. Dies geschieht über IndraWorks Engineering.

- **G75.1** (G75) gehört immer zu Messtaster 1, dessen Flanke über MP **Probe1Edge** "Flanke Messtaster 1" gewählt wird.
Es gibt nur zwei Möglichkeiten:
MEAS / Dr[i] / Probe1Edge = "positive Flanke"
- oder -
MEAS / Dr[i] / Probe1Edge = "negative Flanke"
- **G75.2** gehört immer zu Messtaster 2, dessen Flanke über MP **Probe2Edge** "Flanke Messtaster 2" gewählt wird.
Es gibt nur zwei Möglichkeiten:
MEAS / Dr[i] / Probe2Edge = "positive Flanke"
- oder -
MEAS / Dr[i] / Probe2Edge = "negative Flanke"

Mindestens eine Messachse:

- Beim **G75.1-** (G75-) Satz gilt:

Messfunktionen

Im Kanal muss mindestens eine Messachse vorhanden sein.

- Beim **G75.2**-Satz gilt:
Im Kanal muss mindestens eine Messachse vorhanden sein, die den Messtaster 2 unterstützt.
- Ansonsten kommt es zu:
Fehler 2105 "Es ist keine Kanalachse für Messtaster freigegeben." -
Es sei denn, der Kanal ist im Test-Betrieb.

Gleiche Messtaster-Flanke:

- Alle Messachsen, deren Messwerte benötigt werden, incl. der Messachse mit dem kleinsten Systemachs-Index, müssen bei der gleichen Messtaster-Flanke auslösen.
Die Einstellungen des Anwenders werden von der NC **nicht** überprüft.
- Hintergrund:
Bei einem G75-Satz werden alle zugehörigen Messachsen für die nächste Messung freigegeben:
 - egal, welche Messtaster-Flanke eingestellt wurde
 - egal, ob alle Messachsen auch wirklich das Signal von ein und demselben Messtaster erhalten
 - egal, ob die Messachse selbst auch Fahrachse ist
 - egal, ob der Messwert jeder Messachse hernach auch genutzt wird oder wirkungslos bleibt

Bestimmen des Signals "G75 aktiv":

- Für ein und dieselbe G75-Funktion kann der Anwender maximal ein "aktiv"-Bit am kanalspezifischen SPS-Interface vereinbaren.
Bei G75 ergeben sich daher 4 Möglichkeiten:
 - Kein Bit; weder für G75.1 noch für G75.2
 - Nur ein Bit für G75.1
 - Nur ein Bit für G75.2
 - Zwei Bits; eins für G75.1 und eins für G75.2
- Die Vereinbarung erfolgt in IndraWorks Engineering über:
PLC / NcFuncBitIf / NcFunc[1..24]
Hier schreibt der Anwender den Namen der G75-Funktion in den betreffenden Pfad. Die Index-Nummer entspricht der Bit-Nummer, d. h. durch Eingabe von z. B. "G75.2" unter NcFunc[21] wird dem Bit21 des NC-funktionsspezifischen Interfaces das Signal "G75.2-aktiv" zugewiesen.
- Die NC-Funktionen G75.1 und G75 sind identisch und benutzen daher das selbe Bit - egal, ob das Bit in Form von "G75" oder "G75.1" vereinbart wurde.
- Werden für ein und dieselbe G75-Funktion zwei oder mehr "aktiv"-Bits vereinbart, dann bedient die NC lediglich das Bit mit dem kleinsten Index. Die anderen bleiben wirkungslos.

Sercos Parameter

Die Messtaster-Funktion des Antriebs ist von Sercos International spezifiziert. Die NC selbst benutzt folgende Sercos Parameter für die G75-Sätze:

Sercos Parameter:		Zugeordnet:		Relevant für
IDN	Name	G75.1	G75.2	SCS-File
S-0-0130	Messwert 1 positive Flanke	X	-	Optional
S-0-0131	Messwert 1 negative Flanke	X	-	Optional
S-0-0132	Messwert 2 positive Flanke	-	X	Optional
S-0-0133	Messwert 2 negative Flanke	-	X	Optional
S-0-0169	Messtaster-Steuerparameter	X	X	-
S-0-0170	Kommando Messtasterzyklus	X	X	-
S-0-0405	Messtaster 1 Freigabe	X	-	-
S-0-0406	Messtaster 2 Freigabe	-	X	-
S-0-0409	Messtaster 1 positiv erfasst	X	-	Optional
S-0-0410	Messtaster 1 negativ erfasst	X	-	Optional
S-0-0411	Messtaster 2 positiv erfasst	-	X	Optional
S-0-0412	Messtaster 2 negativ erfasst	-	X	Optional
S-0-0426	Signalauswahl Messtaster 1	X	-	Pflicht
S-0-0427	Signalauswahl Messtaster 2	-	X	Pflicht

X Ja
- Nein

Tab. 15-4: Sercos Parameter für die G75-Sätze

Anmerkung:

- Besitzt der Antrieb die für den G75-Satz benötigten Sercos Parameter, dann unterstützt er auch die jeweilige Messtaster-Funktion.
- Besitzt er sie nicht, bricht der entsprechende Antrieb den Sercos Hochlauf ab und weist über eine entsprechende Fehler-Meldung auf die fehlende Sercos IDN und damit auf die fehlende Messtaster-Funktion hin.
- Nichts desto trotz können noch zusätzliche Einstellungen und Überprüfungen am Antrieb notwendig sein. Diese sind vom Anwender zu erledigen, bevor er die Messtaster-Funktion der NC in Betrieb nimmt. Details hierzu finden sich in der Dokumentation des Antriebs-Herstellers.

SCS-File bzw. Sercos Datei

Mit dem SCS-File setzt die NC für jede Messachse:

- die Pflicht Sercos Parameter und
- die optionalen Sercos Parameter, die der Anwender benötigt.

SCS-Files müssen grundsätzlich angepasst werden - entweder vom Werkzeugmaschinen-Hersteller oder u.U. auch vom Anwender. Sie stehen somit in der Verantwortung des jeweiligen Editors.

Pflicht-Parameter im SCS-File

- Für Teileprogramme mit G75.1- (G75-) Sätzen muss **S-0-0426** "Signal-Auswahl Messtaster 1" im SCS-File parametrierbar sein.
- Für Teileprogramme mit G75.2-Sätzen muss **S-0-0427** "Signal-Auswahl Messtaster 2" im SCS-File parametrierbar sein.

- Beide Parameter S-0-0426 und S-0-0427 können unabhängig von einander gesetzt werden - entweder auf S-0-0051 "Lage-Istwert-Geber 1" oder S-0-0053 "Lage-Istwert-Geber 2".

Optionale Parameter im SCS-File

G75.1:

Gibt es im SCS-File keine optionalen Parameter, die dem G75.1- (G75-) Satz zugeordnet sind, wird bei jedem G75.1-Satz automatisch:

- das Echtzeit-Status-Bit 1 für
"Messtaster 1 positiv erfasst" (S-0-0409) oder
"Messtaster 1 negativ erfasst" (S-0-0410) genutzt -
je nach dem, ob MP **Probe1Edge** positive oder negative Messtaster-Flanke vorgibt.
- der Messwert über den (Sercos) Service-Kanal gelesen, d. h. entweder
"Messwert 1 positive Flanke" (S-0-0130) oder
"Messwert 1 negative Flanke" (S-0-0131) -
je nach dem, ob MP **Probe1Edge** positive oder negative Messtaster-Flanke vorgibt.

Das heißt, in diesem Falle gibt es im SCS-File weder S-0-0409, S-0-0410, S-0-0130 noch S-0-0131.

G75.1:

Will der Anwender den Messwert vom G75.1- (G75-) Satz zyklisch lesen, weil dies schneller geht als über den Service-Kanal, muss er im SCS-File S-0-0015 "Telegrammart" und

S-0-1050.1.6 "Konfigurations-Liste AT" wie folgt setzen:

- **S-0-0015** auf "frei konfigurierbares Telegramm"
- **S-0-1050.1.6** hinzufügen von
"Messwert 1 positive Flanke" (**S-0-0130**) oder
"Messwert 1 negative Flanke" (**S-0-0131**) -
je nach dem, ob MP **Probe1Edge** positive oder negative Messtaster-Flanke vorgibt.

In diesem Falle wird der Messwert nicht über den Service-Kanal gelesen.

G75.1:

Will der Anwender das Echtzeit-Status-Bit 1 nicht für den G75.1- (G75-) Satz sondern für andere Zwecke nutzen, muss er im SCS-File

S-0-0015 "Telegrammart"

S-0-1050.1.6 "Konfigurations-Liste AT"

S-0-0026 "Konfigurations-Liste Signal-Statuswort" und

S-0-0328 "Bit-Zuweisungs-Liste Signal-Statuswort" wie folgt setzen:

- **S-0-0015** auf "frei konfigurierbares Telegramm"
- **S-0-1050.1.6** hinzufügen von Signal-Statuswort (**S-0-0144**)
- **S-0-0026** hinzufügen von
"Messtaster 1 positiv erfasst" (**S-0-0409**) oder
"Messtaster 1 negativ erfasst" (**S-0-0410**) -

je nach dem, ob MP **Probe1Edge** positive oder negative Messtaster-Flanke vorgibt.

- **S-0-0328** hinzufügen von "0" an der selben Listen-Position, die S-0-0409 bzw. S-0-0410 in der S-0-0026-Liste inne hat.

G75.2:

Gibt es im SCS-File keine optionalen Parameter, die dem G75.2-Satz zugeordnet sind, wird bei jedem G75.2 Satz automatisch:

- das Echtzeit-Status-Bit 1 für
"Messtaster 2 positiv erfasst" (S-0-0411) oder
"Messtaster 2 negativ erfasst" (S-0-0412) genutzt -
je nach dem, ob MP **Probe2Edge** positive oder negative Messtaster-Flanke vorgibt.
- der Messwert über den (Sercos) Service-Kanal gelesen, d. h. entweder
"Messwert 2 positive Flanke" (S-0-0132) oder
"Messwert 2 negative Flanke" (S-0-0133) -
je nach dem, ob MP **Probe2Edge** positive oder negative Messtaster-Flanke vorgibt.

Das heißt, in diesem Falle gibt es im SCS-File weder S-0-0411, S-0-0412, S-0-0132 noch S-0-0133.

G75.2:

Will der Anwender den Messwert vom G75.2-Satz zyklisch lesen, weil dies schneller geht als über den Service-Kanal, muss er im SCS-File

S-0-0015 "Telegrammart" und

S-0-1050.1.6 "Konfigurations-Liste AT" wie folgt setzen:

- **S-0-0015** auf "frei konfigurierbares Telegramm"
- **S-0-1050.1.6** hinzufügen von
"Messwert 2 positive Flanke" (**S-0-0132**) oder
"Messwert 2 negative Flanke" (**S-0-0133**) -
je nach dem, ob MP **Probe2Edge** positive oder negative Messtaster-Flanke vorgibt.

In diesem Falle wird der Messwert nicht über den Service-Kanal gelesen.

G75.2:

Will der Anwender das Echtzeit-Status-Bit 1 nicht für den G75.2-Satz sondern für andere Zwecke nutzen, muss er im SCS-File

S-0-0015 "Telegrammart"

S-0-1050.1.6 "Konfigurations-Liste AT"

S-0-0026 "Konfigurations-Liste Signal-Statuswort" und

S-0-0328 "Bit-Zuweisungs-Liste Signal-Statuswort" wie folgt setzen:

- **S-0-0015** auf "frei konfigurierbares Telegramm"
- **S-0-1050.1.6** hinzufügen von Signal-Statuswort (**S-0-0144**)
- **S-0-0026** hinzufügen von
"Messtaster 2 positiv erfasst" (**S-0-0411**) oder

"Messtaster 2 negativ erfasst" (**S-0-0412**) -

je nach dem, ob MP **Probe2Edge** positive oder negative Messtaster-Flanke vorgibt.

- **S-0-0328** hinzufügen von "0" an der selben Listen-Position, die S-0-0411 bzw. S-0-0412 in der S-0-0026-Liste inne hat.

Irrelevante Parameter für SCS-File

Beim Sercos Phasen-Hochlauf wird automatisch:

- aus MP **EnablProbeFunc**, MP **Probe1Edge** und MP **Probe2Edge** der "Messtaster-Steuerparameter" (S-0-0169) erzeugt und zum Antrieb geschrieben
- an Hand des MP **EnablProbeFunc** im Antrieb das Echtzeit-Steuer-Bit 1 mit "Messtaster 1 Freigabe" (S-0-0405) und das Echtzeit-Steuer-Bit 2 mit "Messtaster 2 Freigabe" (S-0-0406) belegt oder nicht
- aus MP **EnablProbeFunc** das "Kommando Messtasterzyklus" (S-0-0170) erzeugt und zum Antrieb geschrieben.

Deshalb dürfen S-0-0169, S-0-0170, S-0-0405 und S-0-0406 niemals im SCS-File stehen.

15.1.3 Handlungsanweisung: Messen mit Messtasterfunktionen G75.1 und G75.2

Hinweis

Das folgende Beispiel zeigt Ihnen, wie Sie die Messtaster-Funktionen G75.1 und G75.2 in Betrieb nehmen.

Das Beispiel dient lediglich als Vorlage. Sie sind verpflichtet, dasselbe Ihren Anforderungen anzupassen.

Die Inbetriebnahme der Messtaster-Funktionen G75.1 und G75.2 ist nur als Ergänzung der regulären Inbetriebnahme zu betrachten.

Aufgabe

Gegeben seien die Systemachsen: 1, 2, 3, 4, 5 und 6.

- Die Systemachsen 2, 3, 5 und 6 sollen als Messachsen für G75.1 definiert werden und auf die positive Messtaster-Flanke reagieren.
 - Der interne Geber dient als Quelle für den Messwert.
 - Die Messwerte werden zyklisch übertragen.
 - Kein separates Echtzeit-Status-Bit.
 - "G75.1 aktiv" wird für Kanal 1 am Bit 23 des NC-funktionsspezifischen SPS-Interfaces ausgegeben.
- Die Systemachsen 5 und 6 sollen zusätzlich als Messachsen für G75.2 definiert werden und auf die negative Messtaster-Flanke reagieren.
 - Der externe Geber dient als Quelle für den Messwert.
 - Die Messwerte werden nicht zyklisch übertragen.
 - Kein separates Echtzeit-Status-Bit.
 - "G75.2 aktiv" wird für Kanal 1 am Bit 24 des NC-funktionsspezifischen SPS-Interfaces ausgegeben.

Antriebe

- Stellen Sie sicher, dass die Antriebe, die den Systemachsen 2, 3, 5 und 6 zugeordnet sind die Messtaster-Funktionalität unterstützen. Nutzen Sie hierzu die vom Hersteller zur Verfügung gestellte Antriebs-Dokumentation.
- Stellen Sie sicher, dass die Messtaster-Funktionalität in den Antrieben so zur Verfügung steht, wie es der Antriebs-Hersteller fordert. Gegebenenfalls sind hierfür zusätzliche Tätigkeiten erforderlich.
- Stellen Sie sicher, dass das Signal für Messtaster 1 ordnungsgemäß erzeugt wird und am korrekten Hardware-Eingang der Antriebe 2, 3, 5 und 6 angeschlossen ist.
- Stellen Sie sicher, dass das Signal für Messtaster 2 ordnungsgemäß erzeugt wird und am korrekten Hardware-Eingang der Antriebe 5 und 6 angeschlossen ist.

Maschinenparameter (MP)

Setzen Sie die Maschinenparameter in den aufgeführten IndraWorks-Engineering-Pfaden auf die vorgegebenen Werte.

Vergessen Sie nicht, Ihre Änderungen zu speichern.

MEAS / Dr[1] /

- EnablProbeFunc = "Nein"
- Probe1Edge = nicht relevant
- Probe2Edge = nicht relevant

MEAS / Dr[2] /

- EnablProbeFunc = "Nur Messtaster 1"
- Probe1Edge = "positive Flanke"
- Probe2Edge = nicht relevant

MEAS / Dr[3] /

- EnablProbeFunc = "Nur Messtaster 1"
- Probe1Edge = "positive Flanke"
- Probe2Edge = nicht relevant

MEAS / Dr[4] /

- EnablProbeFunc = "Nein"
- Probe1Edge = nicht relevant
- Probe2Edge = nicht relevant

MEAS / Dr[5] /

- EnablProbeFunc = "Beide Messtaster"
- Probe1Edge = "positive Flanke"
- Probe2Edge = "negative Flanke"

MEAS / Dr[6] /

- EnablProbeFunc = "Beide Messtaster"
- Probe1Edge = "positive Flanke"
- Probe2Edge = "negative Flanke"

Für Kanal 1:

PLC / NcFuncBitlf /

- NcFunc[23] = "G75.1"
- NcFunc[24] = "G75.2"

SCS-Files - bzw. Sercos Dateien

Führen Sie folgende Ergänzungen in den SCS-Files der Antriebe 1 bis 6 durch und speichern Sie die geänderten SCS-Files.

SCS-File Antrieb 1:

- Stellen Sie sicher, dass keine für G75 relevante Sercos Parameter im SCS-File vorhanden sind, also keiner der in [Tab. 15-4 "Sercos Parameter für die G75-Sätze" auf Seite 597](#) aufgeführten Parameter.

SCS-File Antrieb 2:

- S-0-0426 = (51) ; S-0-0051 "Lage-Istwert-Geber 1"
- S-0-0015 = (0b111) ; Binär 7: "frei konfigurierbares Telegramm"
- S-0-1050.1.6 = (... , 130, ...) ; S-0-0130 "Messwert 1 positive Flanke" hinzufügen
- Stellen Sie sicher, dass außer S-0-0426 und S-0-0130 keiner der in [Tab. 15-4 "Sercos Parameter für die G75-Sätze" auf Seite 597](#) aufgeführten Parameter im SCS-File vorhanden ist.

SCS-File Antrieb 3:

- Wie SCS-File Antrieb 2.

SCS-File Antrieb 4:

- Wie SCS-File Antrieb 1.

SCS-File Antrieb 5:

- S-0-0426 = (51) ; S-0-0051 "Lage-Istwert-Geber 1"
- S-0-0427 = (53) ; S-0-0053 "Lage-Istwert-Geber 2"
- S-0-0015 = (0b111) ; Binär 7: "frei konfigurierbares Telegramm"
- S-0-1050.1.6 = (... , 130, ...) ; S-0-0130 "Messwert 1 positive Flanke" hinzufügen
- Stellen Sie sicher, dass außer S-0-0426, S-0-0427 und S-0-0130 keiner der in [Tab. 15-4 "Sercos Parameter für die G75-Sätze" auf Seite 597](#) aufgeführten Parameter im SCS-File vorhanden ist.

SCS-File Antrieb 6:

- Wie SCS-File Antrieb 5.

Abschließende Arbeit

Lassen Sie Steuerung und Antriebe neu hochlaufen.

Die SCS-Files werden im Zuge des Sercos Hochlaufs von der Steuerung an die zugehörigen Antriebe übergeben.

Aktivieren

Eine Messung kann nun im Teileprogramm gestartet werden. Dazu werden die NC-Funktionen G75.1 oder G75.2 programmiert.

Im folgenden Beispiel soll die Werkstückposition der Systemachsen 2, 3 (im Teileprogramm als Y und Z bezeichnet) mit Hilfe des Messtaster 1 ermittelt werden. Der Messtaster ist konfiguriert, dass er bei Kontakt mit dem Werkstück eine positive Flanke erzeugt.

Programm:

```

N10 G0 Y100 Z0          ; approach start position for measurement
N20 F500                ; set feed for measurement
N30 G75.1 Y200 Z100     ; approach position in workpiece
110 IF SD(9)=0 THEN     ; if touch probe triggered
120   YPOS = PCSPROBE("Y") ; save measured value of y-axis in YPOS
130   ZPOS = PCSPROBE("Z") ; save measured value of z-axis in ZPOS
N140 MSG(CONTACT)       ; output message
150 ELSE                ; touch probe did not trigger
160   SETWARN("NO CONTACT!") ; output warning
N170 M0                  ; wait for NC start
180 WAIT                ; synchronization with NC preparation
190 CLRWARN()           ; delete warning
200 ENDIF

```



Sobald der Messtaster auslöst (also Kontakt zum Werkstück hat), stoppt die NC die Bewegung um Beschädigungen am Taster zu verhindern. Die programmierte Zielposition wird somit nicht erreicht.

15.2 Fliegendes Messen (FME)

15.2.1 Beschreibung

Zusammenfassung

FME ergänzt einen Fahr-Befehl (G0, G1, etc.). FME misst während der Fahr-Bewegung die Werkstück-Position und trägt daher die Bezeichnung "Fliegendes Messen". Anstelle des Werkzeugs wird ein Messtaster eingespannt. Wenn der Messtaster während des Fahrens gegen das Werkstück stößt, wird umgehend die Position des Messtasters gespeichert. Das Messen ist nur bei einer einzelnen Achse möglich. Der Fahr-Befehl wird immer vollständig ausgeführt.

Die NC besitzt zwei FME-Funktionen:

- Initialisierung Fliegendes Messen (InitMeas bzw. IME) und
- Fliegendes Messen (FlyMeas bzw. FME).

Beide benutzen ausschließlich den ersten Messtaster.

Begriffe bei FME

Messtaster:

- sind all diejenigen Sensoren, die die Werkstückposition prüfen, z. B. mechanische Messtaster, Hall-Sensoren, etc.
- kennen nur zwei Zustände: "Werkstück" oder "kein Werkstück"
- formen diese Zustände in ein elektrisches Ausgangssignal um

Die Messachse:

- kann nur ein solcher Sercos Antrieb sein, der selbst die Messtaster-Funktion unterstützt
- wird über einen Maschinenparameter zu einer solchen definiert (MP **EnablProbeFunc** "Messtasterfunktion verfügbar") - und über den IME-Satz initialisiert
- speichert beim Empfang der Messtaster-Flanke den Messwert
- sollte gleichzeitig Fahrachse sein - muss aber nicht

Der Messwert:

- ist der von der Messachse bei Empfang einer Messtaster-Flanke gespeicherte Wert

- stammt immer von der Quelle, die zuvor festgelegt wurde - mittels S-0-0426 "Signal-Auswahl Messtaster 1"
- kennt als Quelle nur den Wert des aktiven Geber-Systems

Was passiert im IME-Satz?

IME (InitMeas) initialisiert FME (FlyMeas):

Die in FME programmierte Messachse muss zuvor mit Hilfe der Funktion "InitMeas" (IME) einmalig initialisiert werden.

Was passiert im FME-Satz?

FME (FlyMeas) ergänzt einen Fahr-Befehl:

Der Fahr-Befehl wird ausgeführt.

Parallel zu diesem startet FME die Messtaster-Funktion der programmierten Messachse.

Empfängt die Messachse eine Messtaster-Flanke, speichert sie umgehend den gefundenen Messwert.

Der Fahr-Befehl ist beim Erreichen der programmierten Position abgeschlossen.

FME prüft, ob der Messtaster ausgelöst hat oder nicht.

Wenn ja:

liest die NC den Messwert der Messachse ein - und fährt mit der nächsten Programmzeile fort.

Wenn nein:

gibt die NC eine Warnung aus und wartet solange, bis das Messereignis eintritt.

Warum hat der Messtaster nicht ausgelöst?

Mögliche Ursachen hierfür wären:

- Im Fahr-Befehl sind die Koordinaten nicht korrekt gewählt und das Werkstück wurde vom Messtaster nicht erreicht.
- Der Messtaster hat ausgelöst, aber die Messachse erhält keine Messtaster-Flanke; u.U. liegt ein Verdrahtungsfehler vor.

Was passiert nach dem FME-Satz?

Im Anschluss an einen FME-Satz, muss der Anwender:

- sicherstellen, dass der Messwert ins Teileprogramm übernommen wird.

Hierzu stehen geeignete CPL-Funktionen zur Verfügung.

SPS-Interface

Die SPS ist in der Lage festzustellen, ob ein FME-Satz aktiv ist oder nicht. Hierfür muss der Anwender ein entsprechendes Bit am "NC-funktionsspezifischen Bit-Interface" vereinbaren. Dieses Bit wird dann für die Zeitdauer des FME-Satzes aktiv.

Restriktionen

- Fliegendes Messen kann nicht zusammen mit der Funktion "Schnellhalt über Messtastereingang" eingesetzt werden.
Bei aktivierter Antriebsfunktion "Schnellhalt über Messtastereingang" bremst der Antrieb die Messachse bei Eintreffen des Messereignisses selbständig ab. Dieses Verhalten steht im Widerspruch zur Funktion Fliegendes Messen.

Relevante NC-Funktionen

InitMeas (IME)	Initialisierung Fliegendes Messen
FlyMeas (FME)	Fliegendes Messen

Tab. 15-5: Relevante NC-Funktionen

Relevante CPL-Funktionen

PROBE() PCSPROBE()	Liefern den Messwert der (synchronen) Messachse - aber jeweils in unterschiedlicher Form.

Tab. 15-6: Relevante CPL-Funktionen

Relevante Maschinenparameter (MP)

Probe1Edge	Pfad: MEAS / Dr[i] / Probe1Edge Name: "Flanke Messtaster 1" (entspricht Macoda-ID 1003 00011)
EnablProbeFunc	Pfad: MEAS / Dr[i] / EnablProbeFunc Name: "Messtasterfunktion verfügbar" (entspricht Macoda-ID 1003 00012)
NcFunc[1..24]	Pfad: PLC / NcFuncBitIf / NcFunc[1..24] Name: "NC-funktionsspezifisches Bit-Interface" (entspricht Macoda-ID 3020 00001/ 3020 00002)

Tab. 15-7: Relevante Maschinenparameter (MP)

15.2.2 Inbetriebnahme

Maschinenparameter (MP)

Bestimmen der Messachsen:

- Alle Systemachsen, die der Anwender als Messachsen nutzen möchte, muss er über MP **EnablProbeFunc** "Messtasterfunktion verfügbar" zu einer solchen bestimmen, d. h. er muss in IndraWorks Engineering MEAS / Dr[i] / EnablProbeFunc = "Nur Messtaster 1" setzen.
- Wird hingegen in IndraWorks Engineering MEAS / Dr[i] / EnablProbeFunc = "Beide Messtaster" gesetzt, führt dies zu keinem Fehler, obwohl der zweite Messtaster nicht für das Fliegende Messen genutzt werden kann.

Bestimmen der Messtaster-Flanke:

- Sobald eine Systemachse zur Messachse bestimmt wurde, muss festgelegt werden, auf welche Messtaster-Flanke sie reagiert. Dies geschieht über IndraWorks Engineering.
- Fliegendes Messen gehört immer zu Messtaster 1, dessen Flanke über MP **Probe1Edge** "Flanke Messtaster 1" gewählt wird.

Es gibt nur zwei Möglichkeiten:

entweder

MEAS / Dr[i] / Probe1Edge = "positive Flanke"

oder

MEAS / Dr[i] / Probe1Edge = "negative Flanke"

Bestimmen des Signals "FME aktiv":

- Für die FME-Funktion kann der Anwender maximal ein "aktiv"-Bit am kanalspezifischen SPS-Interface vereinbaren.
- Die Vereinbarung erfolgt in IndraWorks Engineering über PLC / NcFunc-Bitlf / NcFunc[1..24]
Hier schreibt der Anwender "FME" in den betreffenden Pfad. Die Index-Nummer entspricht der Bit-Nummer, d. h. durch Eingabe von z. B. "FME" unter NcFunc[21] wird dem Bit21 des NC-funktionsspezifischen Interfaces das Signal "FME-aktiv" zugewiesen.
- Werden für die FME-Funktion zwei oder mehr "aktiv"-Bits vereinbart, dann bedient die NC lediglich das Bit mit dem kleinsten Index. Die anderen bleiben wirkungslos.

Sercos Parameter

Die Messtaster-Funktion des Antriebs ist von Sercos International spezifiziert. Das heißt, die NC steuert die Messtaster-Funktion des Antriebs über Sercos Parameter. Für das Fliegende Messen benutzt die NC die folgenden:

Sercos Parameter:		Relevant für
IDN	Name	SCS-File:
S-0-0130	Messwert 1 positive Flanke	Optional
S-0-0131	Messwert 1 negative Flanke	Optional
S-0-0169	Messtaster-Steuerparameter	-
S-0-0170	Kommando Messtasterzyklus	-
S-0-0405	Messtaster 1 Freigabe	-
S-0-0409	Messtaster 1 positiv erfasst	Optional
S-0-0410	Messtaster 1 negativ erfasst	Optional
S-0-0426	Signalauswahl Messtaster 1	Pflicht

- Nein

Tab. 15-8: Sercos Parameter für Fliegendes Messen

Anmerkung:

- Besitzt der Antrieb die für Fliegendes Messen benötigten Sercos Parameter, dann unterstützt er auch die Messtaster-Funktion.
- Besitzt er sie nicht, bricht der entsprechende Antrieb den Sercos Hochlauf ab und weist über eine entsprechende Fehler-Meldung auf die fehlende Sercos IDN und damit auf die fehlende Messtaster-Funktion hin.
- Nichts desto trotz können noch zusätzliche Einstellungen und Überprüfungen am Antrieb notwendig sein. Diese sind vom Anwender zu erledigen, bevor er die Messtaster-Funktion der NC in Betrieb nimmt. Details hierzu finden sich in der Dokumentation des Antriebs-Herstellers.

SCS-File bzw. Sercos Datei*Mit dem SCS-File setzt die NC für die Messachse:*

- den Pflicht Sercos Parameter und
- die optionalen Sercos Parameter, die der Anwender benötigt.

SCS-Files müssen grundsätzlich angepasst werden - entweder vom Werkzeugmaschinen-Hersteller oder unter Umständen auch vom Anwender. Sie stehen somit in der Verantwortung des jeweiligen Editors.

Pflicht-Parameter im SCS-File

- Für Teileprogramme mit Fliegendem Messen (FME) muss **S-0-0426** "Signal-Auswahl Messtaster 1" im SCS-File parametrierung sein.
- Parameter S-0-0426 wird gesetzt -
entweder auf:
S-0-0051 "Lage-Istwert-Geber 1"
oder
S-0-0053 "Lage-Istwert-Geber 2".

Optionale Parameter im SCS-File

Gibt es im SCS-File keine optionalen Parameter für Fliegendes Messen:

- wird bei jedem IME-Satz automatisch:
das Echtzeit-Status-Bit 1 mit
"Messtaster 1 positiv erfasst" (S-0-0409) oder
"Messtaster 1 negativ erfasst" (S-0-0410) belegt -
je nach dem, ob MP **Probe1Edge** positive oder negative Messtaster-
Flanke vorgibt.
- wird bei jedem FME-Satz automatisch:
der Messwert über den (Sercos) Service-Kanal gelesen, d. h. entweder
"Messwert 1 positive Flanke" (S-0-0130) oder
"Messwert 1 negative Flanke" (S-0-0131) -
je nach dem, ob MP **Probe1Edge** positive oder negative Messtaster-
Flanke vorgibt.

Das heißt, in diesem Falle gibt es im SCS-File weder S-0-0409, S-0-0410, S-0-0130 noch S-0-0131.

FME:

Will der Anwender den Messwert zyklisch lesen, weil dies schneller geht als über den Service-Kanal, muss er im SCS-File

S-0-0015 "Telegrammart" und

S-0-1050.1.6 "Konfigurations-Liste AT" wie folgt setzen:

- **S-0-0015** auf "frei konfigurierbares Telegramm"
- **S-0-1050.1.6** hinzufügen von
"Messwert 1 positive Flanke" (**S-0-0130**) oder
"Messwert 1 negative Flanke" (**S-0-0131**) -
je nach dem, ob MP **Probe1Edge** positive oder negative Messtaster-
Flanke vorgibt.

In diesem Falle wird der Messwert nicht über den Service-Kanal gelesen.

FME:

Will der Anwender das Echtzeit-Status-Bit 1 nicht für das Fliegende Messen sondern für andere Zwecke nutzen, muss er im SCS-File

S-0-0015 "Telegrammart"

S-0-1050.1.6 "Konfigurations-Liste AT"

S-0-0026 "Konfigurations-Liste Signal-Statuswort" und

S-0-0328 "Bit-Zuweisungs-Liste Signal-Statuswort" wie folgt setzen:

- **S-0-0015** auf "frei konfigurierbares Telegramm"
- **S-0-1050.1.6** hinzufügen von Signal-Statuswort (**S-0-0144**)
- **S-0-0026** hinzufügen von
 "Messtaster 1 positiv erfasst" (**S-0-0409**) oder
 "Messtaster 1 negativ erfasst" (**S-0-0410**) -
 je nach dem, ob MP **Probe1Edge** positive oder negative Messtaster-Flanke vorgibt.
- **S-0-0328** hinzufügen von "0" an der selben Listen-Position, die S-0-0409 bzw. S-0-0410 in der S-0-0026-Liste inne hat.

Irrelevante Parameter für SCS-File

Beim Sercos Phasen-Hochlauf wird automatisch

- aus MP **EnablProbeFunc** und MP **Probe1Edge** der "Messtaster-Steuerparameter" (S-0-0169) erzeugt und zum Antrieb geschrieben
- an Hand von MP **EnablProbeFunc** im Antrieb das Echtzeit-Steuer-Bit 1 mit "Messtaster 1 Freigabe" (S-0-0405) belegt oder nicht
- aus MP **EnablProbeFunc** das Kommando "Messtasterzyklus" (S-0-0170) erzeugt und zum Antrieb geschrieben.

Deshalb dürfen S-0-0169, S-0-0170 und S-0-0405 niemals im SCS-File stehen.

15.2.3 Handlungsanweisung: Fliegendes Messen

Hinweis

Das folgende Beispiel zeigt Ihnen, wie Sie die Messtaster-Funktion "Fliegendes Messen" in Betrieb nehmen.

Das Beispiel dient lediglich als Vorlage. Sie sind verpflichtet, dasselbe Ihren Anforderungen anzupassen.

Die Inbetriebnahme der Messtaster-Funktion "Fliegendes Messen" ist nur als Ergänzung der regulären Inbetriebnahme zu betrachten.

Aufgabe

Gegeben seien die Systemachsen: 1, 2, 3, 4, 5 und 6.

- Die Systemachse 2 soll als Messachse für FME definiert werden und auf die positive Messtaster-Flanke reagieren.
 - Der interne Geber dient als Quelle für den Messwert.
 - Die Messwerte sind zyklisch zu übertragen.
 - Kein separates Echtzeit-Status-Bit.
 - "FME-aktiv" wird für Kanal 1 am Bit 22 des NC-Funktionsspezifischen SPS-Interfaces ausgegeben.

Antriebe

- Stellen Sie sicher, dass der Antrieb, der der Systemachse 2 zugeordnet ist, die Messtaster-Funktionalität unterstützt. Nutzen Sie hierzu die vom Hersteller zur Verfügung gestellte Antriebs-Dokumentation.
- Stellen Sie sicher, dass die Messtaster-Funktionalität im Antrieb so zur Verfügung steht, wie es der Antriebs-Hersteller fordert. Gegebenenfalls sind hierfür zusätzliche Tätigkeiten erforderlich.

- Stellen Sie sicher, dass das Signal für Messtaster 1 ordnungsgemäß erzeugt wird und am korrekten Hardware-Eingang des Antriebs 2 angeschlossen ist.

Maschinenparameter (MP)

Setzen Sie die Maschinenparameter in den aufgeführten IndraWorks-Engineering-Pfaden auf die vorgegebenen Werte.

Vergessen Sie nicht, Ihre Änderungen zu speichern.

MEAS / Dr[1] /

- EnablProbeFunc = "Nein"
- Probe1Edge = nicht relevant

MEAS / Dr[2] /

- EnablProbeFunc = "Nur Messtaster 1"
- Probe1Edge = "positive Flanke"

MEAS / Dr[3] /

- EnablProbeFunc = "Nein"
- Probe1Edge = nicht relevant

MEAS / Dr[4] /

- EnablProbeFunc = "Nein"
- Probe1Edge = nicht relevant

MEAS / Dr[5] /

- EnablProbeFunc = "Nein"
- Probe1Edge = nicht relevant

MEAS / Dr[6] /

- EnablProbeFunc = "Nein"
- Probe1Edge = nicht relevant

Für Kanal 1:

PLC / NcFuncBitlf /

- NcFunc[22] = "FME"

SCS-Files - bzw. Sercos Dateien

Führen Sie folgende Ergänzungen in den SCS-Files der Antriebe 1 bis 6 durch und speichern Sie die geänderten SCS-Files.

SCS-File Antrieb 1:

- Stellen Sie sicher, dass keine für Fliegendes Messen relevante Sercos Parameter im SCS-File vorhanden sind, also keiner der in [Tab. 15-8 "Sercos Parameter für Fliegendes Messen" auf Seite 606](#) aufgeführten Parameter.

SCS-File Antrieb 2:

- S-0-0426 = (51) ; S-0-0051 "Lage-Istwert-Geber 1"
- S-0-0015 = (0b111) ; Binär 7: "frei konfigurierbares Telegramm"
- S-0-1050.1.6 = (... , 130, ...) ; S-0-0130 "Messwert 1 positive Flanke" hinzufügen
- Stellen Sie sicher, dass außer S-0-0426 und S-0-0130 keiner der in [Tab. 15-8 "Sercos Parameter für Fliegendes Messen" auf Seite 606](#) aufgeführten Parameter im SCS-File vorhanden ist.

SCS-File Antrieb 3:

- Wie SCS-File Antrieb 1.

SCS-File Antrieb 4:

- Wie SCS-File Antrieb 1.

SCS-File Antrieb 5:

- Wie SCS-File Antrieb 1.

SCS-File Antrieb 6:

- Wie SCS-File Antrieb 1.

Abschließende Arbeit

Lassen Sie Steuerung und Antriebe neu hochlaufen.

Die SCS-Files werden im Zuge des Sercos Hochlaufs von der Steuerung an die zugehörigen Antriebe übergeben.

Aktivieren

Eine Messung kann nun im Teileprogramm gestartet werden. Dazu wird erst die NC-Funktion InitMeas (IME) und danach FlyMeas (FME) programmiert.

Im folgenden Beispiel soll die Werkstückposition der Systemachse 2 (im Teileprogramm als Y bezeichnet) mit Hilfe des Messtasters ermittelt werden. Der Messtaster ist konfiguriert, dass er bei Kontakt mit dem Werkstück eine positive Flanke erzeugt.

Programm:

```

N10 G0 Y100 Z0                ; Approaching starting position for measurement
N20 IME(MpiAxis 2)            ; Initialize flying measurement for system axis 2
N20 F500                      ; Set feed for measurement
N30 G1 FME(MpiAxis 2) Y200 Z100 ; Approach position behind the workpiece
N40 WAIT                      ; Synchronization with NC preparation
110 IF SD(9)=0 THEN           ; if touch probe triggered
120   YPOS = PPOS("Y")        ; Storing measured value of y-axis in YPOS
N130 MSG(CONTACT)             ; Output message
140 ELSE                      ; Touch probe not triggered
150 SETWARN("NO CONTACT!")    ; Outputting warning
N160 M0                       ; Waiting on NC start
170 WAIT                      ; Synchronization with NC preparation
180 CLRWARN()                 ; Delete warning
190 ENDIF

```



Im Gegensatz zu G75 wird beim fliegenden Messen die Messbewegung komplett abgearbeitet. Das geschieht unabhängig davon, ob der Messtaster auslöst (also Kontakt zum Werkstück hat) oder nicht.

15.3 Messen auf einem Satz (MOB)**15.3.1 Beschreibung****Funktion**

MOB ist eine satzweise wirksame Funktion, die zum Messen einer Werkstückposition verwendet werden kann. MOB wird zusammen mit einer Fahrbewegung (G1, TcsMove, ...) programmiert.

Das Messen auf einem Satz ähnelt der Funktion "Fahren gegen Messtaster (G75)" ([Kap. 15.1 "Fahren gegen Messtaster \(G75\)" auf Seite 593](#)). Genau wie G75 misst MOB während der Fahrbewegung die Werkstückposition. Nach Ansprechen des Messtasters wird die Position gespeichert und die Bewegung gestoppt.

"MOB G1 X100 F100" ist im Ergebnis gleichwertig mit "G75 X100 F100".

Wenn die Funktion "Fahren gegen Messtaster (G75)" konfiguriert ist, dann ist die Funktion "Messen auf einem Satz (MOB)" auch konfiguriert.

Restriktionen und Besonderheiten

Wenn für einen Antrieb die Funktion "Schnellhalt über Messtastereingang" aktiviert ist, dann bremst dieser Antrieb beim Eintreffen des Messereignisses selbständig ab.

Relevante NC-Funktionen

MOB / MOB(MPx)	Messen auf einem Satz

Tab. 15-9: Relevante NC-Funktionen

Relevante CPL-Funktionen

SD(9)	Liefert die Information, ob der Messtaster ausgelöst hat.
PROBE(), PCSPROBE()	Liefern den Messwert einer Messachse - aber jeweils in unterschiedlicher Form.

Tab. 15-10: Relevante CPL-Funktionen

Relevante IF-Signale

MOB(MP1) / MOB verwendet das der NC-Funktion G75.1 (G75) zugeordnete funktionsspezifische IF-Signal.

MOB(MP2) verwendet das der NC-Funktion G75.2 zugeordnete funktions-spezifische IF-Signal.

Relevante Maschinenparameter (MP)

EnablProbeFunc	Pfad: MEAS / Dr[i] / EnablProbeFunc Name: "Messtasterfunktion verfügbar" (entspricht Macoda-ID 1003 00012)
Probe1Edge	Pfad: MEAS / Dr[i] / Probe1Edge Name: "Flanke Messtaster 1" (entspricht Macoda-ID 1003 00011)
Probe2Edge	Pfad: MEAS / Dr[i] / Probe2Edge Name: "Flanke Messtaster 2" (entspricht Macoda-ID 1003 00013)
NcFunc[1..24]	Pfad: PLC / NcFuncBitIf / NcFunc[1..24] Name: "NC-funktionsspezifisches Bit-Interface" (entspricht Macoda-ID 3020 00001/ 3020 00002) <i>Die Einträge G75 und G75.2 werden mitverwendet:</i> - Eintrag "G75": MOB bzw. MOB(MP1) setzt dieses Signal - Eintrag "G75.2": MOB(MP2) setzt dieses Signal

Tab. 15-11: Relevante Maschinenparameter (MP)

Relevante Sercos-Parameter

Es gelten die gleichen Sercos-Parameter wie für die Funktionen G75 und G75.2. Siehe [Kap. "Sercos Parameter" auf Seite 596](#).

Die Spalte "G75.1" gilt für MOB/MOB(MP1) und die Spalte "G75.2" gilt für MOB(MP2).

Beispiel

Messen mit einer Linearbewegung

Programm:

```
..
N100 G0 Y100 Z0
N110 MOB G1 Y200 Z100 F100 ;Measuring with linear traversing
0120 IF SD(9) = 0 THEN
0130 YPOS! = PCSPROBE("Y")
0140 ZPOS! = PCSPROBE("Z")
0150 ELSE
0160 SETWARN("NO CONTACT")
N170 M0
0180 ENDIF
..
```

Messen in Werkzeugrichtung

Programm:

```
N100 CRD(1) ;Activate an axis transformation
..
N200 G0 x10 Y20 z50
N220 MOB TCM(,, -20) F100 ;MOB with traversing in TCSz direction
0230 IF SD(9) = 0 THEN
0240 ZPOS! = PCSPROBE("z")
0250 ELSE
0260 SETWARN("NO CONTACT")
N270 M0
0280 ENDIF
..
```

15.3.2 Applizieren, Aktivieren und Deaktivieren

Alle notwendigen Handlungen sind im [Kap. 15.1.2 "Inbetriebnahme" auf Seite 595](#) beschrieben.

Eine Steuerung, die mit G75 messen kann, kann auch mit MOB messen.

15.4 Messen auf einer Kontur (MOC)

15.4.1 Beschreibung

Zusammenfassung

Das Messen auf einer Kontur ähnelt den Funktionen "Fahren gegen Messtaster (G75)" ([Kap. 15.1 "Fahren gegen Messtaster \(G75\)" auf Seite 593](#)) und "Fliegendes Messen (FME)" ([Kap. 15.2 "Fliegendes Messen \(FME\)" auf Seite 603](#)). Genau wie die beschriebenen Funktionen misst MOC während der Fahrbewegung die Werkstückposition. Im Gegensatz zu G75 und FME kann beim Messen auf einer Kontur eine beliebige Bahn als Messstrecke benutzt werden.

Messen auf einer Kontur ist eine modale NC-Funktion, die am Beginn der Messstrecke eingeschaltet (ein MOC mit Parametern) und am Ende wieder ausgeschaltet (ein MOC mit leeren Klammer oder das nächste MOC) wird. Während sie aktiv ist, können beliebige Fahrbefehle programmiert werden. Andere Messfunktionen sind dagegen nicht erlaubt.

Als Auslöser der Messung kann ein Messtaster, ein Kundeneingang im Kanalinterface oder ein HighSpeed-Eingang genutzt werden. Wird ein Messtaster verwendet, wird dieser anstelle des Werkzeugs eingespannt. Stößt der Messtaster während des Fahrens gegen das Werkstück, wird umgehend die Position des Messtasters gespeichert.

Das Messen ist nur mit synchronen Achsen eines Kanals möglich.

Begriffe bei MOC

Messtaster:

- sind all diejenigen Sensoren, die die Werkstückposition prüfen, z. B. mechanische Messtaster, Hall-Sensoren etc.
- kennen nur zwei Zustände: "Werkstück" oder "kein Werkstück"
- formen diese Zustände in ein elektrisches Ausgangssignal um

Messsignalquelle:

- kann ein Messtaster sein
- oder -
- ein Kundeneingang im Kanalinterface (qCh_Custom1..8)
- oder -
- ein HighSpeed-Eingang

Messereignis:

- ist ein Zeitpunkt, an dem die Messsignalquelle eine Pegeländerung (Flanke) aufweist und diese Flanke in MOC als Auslöser programmiert wurde.
- startet das Ermitteln der Messwerte.
- löst verschiedene weitere Reaktionen von MOC aus (siehe [Kap. "Mögliche Reaktionen auf das Messereignis" auf Seite 613](#))
- wird nur einmal auf der Messstrecke registriert. Nach dem ersten Messereignis überwacht MOC die Signalquelle nicht länger.

Die Messachsen:

- können nur synchrone Achsen des Kanals sein.
- werden im MOC-Satz angegeben (wenn nicht programmiert, werden alle konfigurierten Kanalachsen gemessen).
- dürfen auf der Messstrecke nicht aus dem Kanal entfernt werden.
- müssen folgende Zusatzbedingungen erfüllen, wenn ein Messtaster als Signalquelle benutzt wird:
 - es können nur solche Sercos Antriebe sein, die selbst die Messtasterfunktion unterstützen.
 - müssen über Maschinenparameter "Messtasterfunktion verfügbar" (MEAS/Dr[i]/EnablProbeFunc) freigeschaltet werden.

Die Messwerte:

- sind bei Kundeninterface und HighSpeed-Eingang immer für alle Kanalachsen vorhanden.
- sind bei Messtaster nur für die Messachsen verfügbar.
- werden in einem Systemdatum gespeichert.
- werden in Programmkoordinaten (PCS) umgerechnet, wenn alle Kanalachsen gemessen wurden.
- können auch im Teileprogramm mittels CPL-Funktionen (SD(9), PROBE(), ...) abgefragt werden.

Mögliche Reaktionen auf das Messereignis

Die NC-Funktion MOC kann beim Eintreten des Messereignisses (neben dem Speichern der Messwerte) eine zusätzliche Reaktion ausführen. Diese Reaktion wird mit dem MOC-Befehl am Anfang der Messstrecke als Zahlencode programmiert.

1. Geschwindigkeit ändern

Die NC fährt im Allgemeinen mit dem programmierten Vorschub (F-Wert). In G94-Betrieb kann zusätzlich ein zweiter Vorschub programmiert werden (F2-Wert), der statt dessen benutzt wird, wenn ein SPS-Bit (qCh_Feedrate2) gesetzt ist.

Wird die Reaktion 1 (Geschwindigkeit ändern) in MOC programmiert, fährt die NC den Beginn der Messstrecke mit dem zweiten Vorschub. Das geschieht unabhängig davon, ob das SPS-Bit aktiv ist oder nicht. Ab dem Messereignis ist wieder die normale Logik aktiv. Das bedeutet, dass danach wieder das SPS-Bit entscheidet, mit welchem Vorschub gefahren wird.

2. Asynchrones Unterprogramm ausführen

Als Reaktion 2 kann MOC beim Auftreten des Messereignisses ein asynchrones Unterprogramm (siehe Dokumentation "MTX Funktionsbeschreibung - Grundlagen", Kap. "Asynchrone Unterprogramme") ausführen. Das Unterprogramm muss vor dem Aufruf von MOC im System angemeldet sein (z. B. per ASPSET).

Wird ein asynchrones Unterprogramm in der Betriebsart "gepufferte NC-Satzvorgabe" gestartet, gehen alle Sätze nach dem Unterprogramm verloren. Wenn also die komplette Messstrecke in einer Nachricht an die NC gegeben wurde, wird diese nur bis zum Punkt, als das Messereignis auftrat, gefahren.

3. Grundstellung

Die dritte Reaktion von MOC löst beim Auftreten des Messereignisses eine Kanalgrundstellung aus.



Da das Teileprogramm bei einer Kanalgrundstellung abgebrochen wird, kommt nach dem Messereignis das Ende der Messstrecke (MOC()) nicht mehr und das Messen auf einer Kontur bleibt aktiv. Aus diesem Grund wird empfohlen, ein MOC() in den "Zustand nach Kanalgrundstellung" (Maschinenparameter CHAN/Ch[k]/Ini/ChResetState) zu schreiben.

4. Fliegendes Messen

Bei Reaktion 4 führt MOC keine weiteren Aktionen neben dem Messen aus. Das bedeutet, dass die Messstrecke komplett (ohne Zwischenstopp) abgearbeitet wird.

5. verbleibende Messstrecke übergehen

MOC kann als Reaktion 5 die Messstrecke nach dem Messereignis übergehen. Wurde diese Reaktion programmiert, hält die NC beim Auftreten des Messereignis auf der Bahn an. Danach werden automatisch die verbleibenden Sätze ohne Bewegung übergangen. Alle CPL- und Hilfsfunktionen dieser Sätze werden jedoch abgearbeitet.

Enthält der Endesatz der Messstrecke (MOC()) eine Bewegung, wird diese danach ausgeführt. Wurde keine Bewegung in diesem Satz programmiert, fährt das Messen auf einer Kontur mit der letzten aktiven Bewegungsfunktion zu dem Endpunkt der Messstrecke.



Bei Reaktion 5 verfährt die NC-Funktion im Endesatz der Messstrecke die Maschine auf die letzte Position der Messstrecke, auch wenn keine Bewegung programmiert wurde.

Es wird empfohlen, im Endesatz eine sichere absolute Position per Gerade zu programmieren (G90 G0). Diese Position muss von jedem Punkt der Kontur kollisionsfrei erreichbar sein.



Es wird empfohlen, das Ende der Messstrecke (`MOC()`) in den "Zustand nach Kanalgrundstellung" (Maschinenparameter `CHAN/Ch[k]/Ini/ChResetState`) aufzunehmen, da es sonst während der Grundstellung zu Laufzeitfehlern kommen kann, wenn während des Übergehens von Sätzen ein Fehler auftrat.

Ist Reaktion 5 gewählt, dürfen auf der Messstrecke weder Transformationen bzw. Placements umgeschaltet noch die aktive Nullpunktverschiebungen geändert werden.

Messereignis nicht eingetreten

Ist das Messereignis auf der gesamten Messstrecke nicht eingetreten, kann das Messen auf einer Kontur, je nach Programmierung, verschieden reagieren.

- 0 Es erfolgt keine Reaktion. (Vorgabe)
- 1 Es wird eine Warnung ausgegeben, dass das Messereignis nicht eingetreten ist.
- 2 Ein Laufzeitfehler wird ausgelöst.

Ermitteln der Messwerte per Systemdatum

Die Messwerte werden bei jedem Messvorgang in der Einheit Millimeter in das Systemdatum `"/SysMeasPos"` geschrieben. Das Systemdatum muss vom Anwender angelegt werden, bevor `MOC` genutzt werden kann (siehe Dokumentation "MTX Funktionsbeschreibung - Sonderfunktionen", Kapitel "Systemdaten"). Eine Vorlage für die SD-Struktur ist in `"/feprom/schemas/sdmoc.xsd"` zu finden. Das Systemdatum sollte als kanalgroßes Feld definiert werden (siehe `"/feprom/SDDefMoc.xml"`).

<code>/SysMeasPos[Channel]/ACSPos[1..8]</code>	double	gemessene Positionen der Kanalachsen in mm
<code>/SysMeasPos[Channel]/PCSPos[1..8]</code>	double	gemessene Positionen der Kanalachsen umgerechnet auf Programmkoordinaten in mm
<code>/SysMeasPos[Channel]/Version</code>	uint	fortlaufende Versionsnummer der Messung
<code>/SysMeasPos[Channel]/Valid</code>	uint	ist 1, wenn Messwerte gültig sind

Tab. 15-12: Systemdatenstruktur für Messen auf einer Kontur

Die Messwerte werden in die Felder `ACSPos` und `PCSPos` unter der Koordinatennummer eingetragen. Das bedeutet, die gemessene Position der ersten Koordinate im Kanal ist unter `ACSPos[1]` zu finden.

Nicht gemessene Achspositionen werden auf 1000000mm gesetzt. Die PCS-Position wird nur dann berechnet, wenn alle Achsen des Kanals gemessen wurden. Die Korrekturen basieren dabei auf den aktuellen Ist-Positionen.

Über die Versionsnummer kann geprüft werden, ob eine Messung ausgelöst wurde. In diesem Fall hat sich die Versionsnummer um 1 erhöht.

Eine externe Anwendung kann das Änderungsereignis dieses Systemdatums abonnieren (OPC-Item `NC.SystemDataEvent`, (siehe Dokumentation "MTX OPC-Kommunikation") und wird somit geweckt, wenn die Messung beendet wurde.

Ermitteln der Messwerte per CPL-Funktionen

Mit der CPL-Funktion `SD(9)` kann im Teileprogramm geprüft werden, ob das Messereignis eingetreten ist. Die Funktionen `PROBE()` und `PCSPROBE()` liefern dann den Messwert einer Achse.



Im Allgemeinen muss zwischen dem Ende der Messstrecke und der ersten CPL-Funktion, die sich auf die Messung bezieht, ein `WAIT` zur Synchronisation programmiert werden.

Besonderheiten und Restriktionen

Bei Auswahl der Messsignalquelle "Messtaster (MT)" und aktivierter Antriebsfunktion "Schnellhalt über Messtastereingang" bremsen die Messachsen bei Eintreffen des Messereignisses selbständig ab ([Kap. 15.4.6 "MOC mit Schnellhalt über Messtastereingang" auf Seite 620](#)).

Alle MOC-Reaktionen, die ohne Achsstillstand ablaufen, können nicht zusammen mit "Schnellhalt über Messtastereingang" verwendet werden. Beim Auftreten des Messereignisses gibt die MTX einen Laufzeitfehler aus.

Dies sind die Reaktionen:

- 1 Geschwindigkeit ändern
- 4 Fliegendes Messen

Relevante NC-Funktionen

MPEDGE	Umkonfigurieren der Messflanke bei Messtastereinsatz
MOC	Messen auf einer Kontur

Tab. 15-13: Relevante NC-Funktionen

Relevante CPL-Funktionen

SD(9)	Ermitteln, ob das Messereignis eingetreten ist.
PROBE(), PCSPROBE()	Liefern den Messwert der (synchronen) Messachse - aber jeweils in unterschiedlicher Form.

Tab. 15-14: Relevante CPL-Funktionen

Relevante Maschinenparameter (MP)

EnablProbeFunc	Pfad: MEAS / Dr[i] / EnablProbeFunc Name: "Messtasterfunktion verfügbar" (entspricht Macoda-ID 1003 00012)
Probe1Edge	Pfad: MEAS / Dr[i] / Probe1Edge Name: "Flanke Messtaster 1" (entspricht Macoda-ID 1003 00011)
Probe2Edge	Pfad: MEAS / Dr[i] / Probe2Edge Name: "Flanke Messtaster 2" (entspricht Macoda-ID 1003 00013)

Tab. 15-15: Relevante Maschinenparameter (MP)

15.4.2 Applizieren

Das Systemdatum `"/SysMeasPos"` muss mit der entsprechenden Struktur angelegt werden (siehe Dokumentation "MTX Funktionsbeschreibung - Sonderfunktionen", Kapitel "Systemdaten"). Die Vorlagen für Struktur und Definition sind in `"/feprom/schemas/sdmoc.xsd_"` und `"/feprom/SDDefMoc.xml_"` zu finden.

Soll ein Messtaster als Signalquelle genutzt werden, müssen alle Messachsen entsprechend konfiguriert werden. Dies erfolgt analog zu den Funktionen "Fahren gegen Messtaster (G75) ", siehe [Kap. 15.1 "Fahren gegen Messtaster \(G75\)" auf Seite 593](#) und "Fliegendes Messen (FME) ", siehe [Kap. 15.2 "Fliegendes Messen \(FME\)" auf Seite 603](#). Der Messtaster sollte mit dem Messtastereingang aller Messachsen verbunden sein.

Soll ein Kundeneingang im Kanalinterface die Signalquelle sein, muss dieser durch das SPS-Programm entsprechende Messflanken erhalten.

Ist ein HighSpeed-Eingang die Signalquelle, muss dieser entsprechend verdrahtet sein.



Das Ende der Messstrecke (`MOC()`) sollte in den "Zustand nach Kanalgrundstellung" (Maschinenparameter `CHAN/Ch[k]/Ini/ChResetState`) geschrieben werden, damit eine Grundstellung eine eventuell laufende Messung beendet.

15.4.3 Aktivieren

Im Teileprogramm wird am Beginn der Messstrecke ein `MOC` mit Parametern programmiert. Die Messung beginnt ab diesem Satz. Wenn der Satz selbst eine Bewegung enthält, erfolgt der Start "fliegend" (also ohne Stop).

Beispiel

`N100 MOC(MT1, ES1, RE2, AS4, ER1, X, Z)`

; Die Positionen der Achsen X und Z sollen gemessen werden, wenn der Messtaster 1 eine fallende Flanke registriert. Zusätzlich soll dann das asynchrone Unterprogramm 4 ausgeführt werden. Falls der Messtaster auf der Messstrecke keine fallende Flanke erfasst, soll eine Warnung ausgegeben werden.

Enthält der `MOC`-Satz eine Bewegung, erfolgt der Start "fliegend" (ohne Stop). Ab dem Beginn der Bewegung ist die Messung scharf.

15.4.4 Deaktivieren

Ein Satz mit einem `MOC` ohne Parameter (mit leeren Klammern) stellt das Ende der Messstrecke dar. Spätestens hier wird die Messlogik deaktiviert. Je nach programmierter Fehlerreaktion wird in diesem Satz eine Warnung oder Fehlermeldung erzeugt, wenn das Messereignis nicht eingetreten ist.

Beispiel

`N500 MOC()`

; Ende der Messstrecke

Enthält der `MOC`-Satz eine Bewegung, erfolgt das Ende "fliegend" (ohne Stop). Ab dem Beginn der Bewegung ist die Messung beendet.

Eine Messstrecke kann auch durch eine nachfolgende Messstrecke beendet werden. In diesem Fall stellt ein `MOC` mit Parametern sowohl das Ende der alten Messstrecke als auch den Anfang der neuen dar.

Beispiel	N100 MOC(CI6, ES3, RE1)	; Messung 1 wird gestartet
	...	; Messung 1 läuft
	N200 MOC(HS2, ES2, RE4, ER2)	; Messung 1 wird beendet, Messung 2 wird gestartet
	...	; Messung 2 läuft
	N300 MOC()	; Messung 2 wird beendet

15.4.5 Handlungsanweisung: Messen auf einer Kontur

Die folgenden Beispiele stellen die Programmierung im Teileprogramm dar. Es wird davon ausgegangen, dass alle Messachsen korrekt konfiguriert sind und das Systemdatum angelegt ist.

Im ersten Beispiel soll die Geschwindigkeitsänderung dargestellt werden. Das SPS-Bit für den zweiten Vorschub soll durchgehend auf Null gesetzt sein.

Beispiel Geschwindigkeitsänderung:

N10 G01 F100 G94	; setze programmierte Geschwindigkeit F auf 100 mm/s
N20 F2=500	; setze zweite Geschwindigkeit F2 auf 500 mm/s
N20 MOC(MP1, ES1, RE1)	; ab diesem Satz wird mit F2=500 gefahren, sobald Messtaster 1 eine fallende Flanke meldet, wird auf F=100 umgeschaltet
N30 X... Y... Z...	; es wird mit F2=500 gefahren, falls noch keine Flanke vom MP1 kam, oder mit F=100, falls eine da war
N40 X... Y... Z... F=70	; es wird mit F2=500 gefahren, falls noch keine Flanke vom MP1 kam, oder mit F=70, falls eine da war
N50 X... Y... Z... F2=400	; es wird mit F2=400 gefahren, falls noch keine Flanke vom MP1 kam, oder mit F=70, falls eine da war
N60 MOC()	; es wird ab jetzt mit F=70 gefahren

Das zweite Beispiel zeigt die Reaktion "asynchrones Unterprogramm ausführen". Diese Variante könnte z. B. zur Crashüberwachung eingesetzt werden. Auf Basis der Messposition kann in dem asynchronen Unterprogramm eine Wegfahrtstrategie ermittelt werden.

Beispiel asynchrones Unterprogramm

N10 ASPSET(3, SAFEPOS01, 11)	; Asynchrones Unterprogramm "SAFEPOS01" im System als Nummer 3 anmelden.
N20 MOC(MP2, ES2, RE2, AS3)	; wenn der Messtaster 2 eine steigende Flanke liefert, wird das Asup 3 (SAFEPOS01) ausgeführt
N30 X... Y... Z...	; es wird gemessen
N40 X... Y... Z...	; es wird gemessen
N50 X... Y... Z...	; es wird gemessen
N60 X... Y... Z...	; es wird gemessen
N70 MOC()	; Messen ist beendet (der Messtaster wird deaktiviert)

Im folgenden Beispiel wird "fliegend" gemessen und die Messwerte werden mit CPL ausgelesen.

Beispiel Fliegendes Messen

N10 G1 F1000 X... Y... Z...	; Startposition anfahren
N20 MOC(MP2, RE4, X, Z) X... Z...	; wenn der Messtaster 2 die vorkonfigurierte Flanke liefert, wird gemessen, der Start erfolgt "fliegend"
N30 X... Y... Z...	; es wird gemessen
N40 X... Y... Z...	; es wird gemessen
N50 X... Y... Z...	; es wird gemessen → hier könnte der Messtaster auslösen • die Messwerte für X und Z werden gelatcht • die gelatchten Werte werden in das Systemdatum geschrieben • die Bewegung wird nicht unterbrochen
N60 X... Y... Z...	; es wird nicht mehr gemessen
N70 MOC() X... Y... Z...	; Messen wird "fliegend" beendet
N80 WAIT	; Synchronisation
90 XPOS = PCSPROBE(1)	; Messwert der X-Achse speichern
100 ZPOS = PCSPROBE(3)	; Messwert der Z-Achse speichern

In diesem Beispiel wird die Reaktion 5 (verbleibende Messstrecke übergehen) dargestellt.

Beispiel verbleibende Messstrecke übergehen

N10 G1 F1000	; Initialisierung
N20 MOC(HS5, ES3, RE5)	; wenn der HighSpeed-Eingang 5 eine beliebige Flanke liefert, wird die Bewegung angehalten, danach geht es mit dem Satz N70 weiter
N30 X... Y... Z...	; es wird gemessen
N40 X... Y... Z...	; es wird gemessen
N50 X... Y... Z...	; es wird gemessen → hier könnte der Messtaster auslösen • es wird auf der Bahn angehalten (Vorschub halt) und die Messwerte gelatcht • die gelatchten Werte werden in das Systemdatum geschrieben • es wird eine Gerade mit Eilgang zum Punkt X10 Y20 Z30 gefahren, weil das so im Endesatz der Messstrecke steht
N60 X... Y... Z...	; diese Bewegung wird nicht ausgeführt, der Satz wird übergangen
N70 MOC() G90 G0 X10 Y20 Z30	; Messen ist beendet, sicheren Punkt anfahren
N80 X... Y... Z...	

15.4.6 MOC mit "Schnellhalt über Messtastereingang"

Funktion	In Einzelfällen ist ein besonders schneller Achsenstopp nach Antasten eines Werkstücks durch einen Messtaster notwendig. Hierfür kann im IndraDrive für den Messtaster 1 die Funktion "Schnellhalt über Messtastereingang" aktiviert werden.
Restriktionen	Alle MOC-Reaktionen, die ohne Achsstillstand ablaufen, können nicht zusammen mit "Schnellhalt über Messtastereingang" verwendet werden (Kap. "Besonderheiten und Restriktionen" auf Seite 616).
Reaktion auf das Messereignis	Die MTX überprüft beim Eintreffen des Messereignisses, ob für einige oder alle Achsen des Kanals ein Schnellhalt durchgeführt wird. <i>Wenn eine oder mehrere Achsen des Kanals mit Schnellhalt gestoppt werden, reagiert die MTX mit:</i> 1. Abbremsen der Bahnbewegung. Die restlichen, nicht von Schnellhalt gestoppten Achsen, werden angehalten 2. Warten auf Stillstand aller Achsen und Übernahme der Istposition der "Schnellhalt"-Achsen 3. Bearbeitung fortsetzen <i>Wenn alle Achsen des Kanals mit Schnellhalt gestoppt werden, reagiert die MTX mit:</i> 1. Abbruch der Bahnbewegung 2. Warten auf Stillstand aller Achsen und Übernahme der Istposition der "Schnellhalt"-Achsen 3. Bearbeitung fortsetzen
Applizieren	Voraussetzung: Messtaster 1 ist in Betrieb genommen. Über P-0-0226 "Messtaster, erweitertes Steuerwort" wird die Funktion eingeschaltet und konfiguriert. Der Funktionsumfang ist abhängig von der verwendeten IndraDrive-Firmware. Es gilt immer die Beschreibung der verwendeten Firmware! S-0-0124 "Stillstandsfenster" muss auf einen sinnvollen Wert gesetzt werden.

15.5 Digitalisieren (Aufnehmen von Messwerten)

15.5.1 Beschreibung

Funktion

Die Funktion "Digitalisieren" stellt die technischen Voraussetzungen zum Aufnehmen eines Oberflächenprofils und zum Abspeichern erfasster Achspositionen bereit:

- Erfassen des Werkstücks in X/Y-Ebene und Z-Richtung mit **automatischem Niveau-Ausgleich** durch gleichzeitige **Online-Korrektur** der Z-Achse:

Um eine vorgegebene Oberfläche (z. B. Freiformfläche) zu erfassen, muss ein möglichst dichtes Punkte-Raster von der Oberfläche aufgenommen werden. Mit einem geeigneten Sensor (z. B. Laser) wird das Werkstück mäanderförmig abgefahren. Dabei wird die Sensorzustellachse Z der Höhenlinie des Oberflächenprofils ständig "online" angepasst ("Höhenregelung"; siehe auch unter "Vorgang des Digitalisierens").

- **Achspositionen in Echtzeit** über Sercos erfassen:

Durch Aufzeichnung der Achspositionen X, Y und der in Echtzeit korrigierten Achsposition Z werden die Höhenlinien durch eine Aneinanderreihung von Punkten digitalisiert.

Die Aufnahme des Oberflächenprofils kann bei aktiver 5-Achstransformation auch mit einer Aufzeichnung der Achspositionen von 5 Achsen durchgeführt werden.

- **Abspeichern der Echtzeitdaten** (Achsenpositionen und Z-Korrekturwerte) in eine Aufzeichnungsdatei:

Speichern auf Festplatte des PC-Panel oder eines externen PC's über Ethernet-Netzwerk.

- **Vorgang** des Digitalisierens **steuerbar** per NC-Programm:
Das Aufnehmen der Achspositionen kann per NC-Anweisung im Programm gestartet, unterbrochen, fortgesetzt und beendet werden.
- Digitalisieren mit **programmierbaren Voreinstellungen** beeinflussen.
- **Synchronisieren eines externen Gerätes** mit der NC-internen Aufzeichnung der Messwerte. Dazu lässt sich einer der High-Speed-Digitalausgänge der Steuerung verwenden.
- In der Aufzeichnungsdatei stehen zusätzliche Informationen (**Normierungsfunktionen**) für eine anschließende Aufbereitung der Echtzeitdaten zur Verfügung.
- Das Programmieren der Laserbahn, auf der das Werkstück "getastet" wird, ist **nicht** Bestandteil der Funktion "Digitalisieren". Dies muss programmtechnisch so gelöst werden, dass die aufgezeichneten Daten für eine spätere Bearbeitung mit einem CAD-System tauglich sind.

Anwendungsmöglichkeiten:

- **Übernahme der Daten** in ein geeignetes CAD/CAM-System (Werkstück visualisieren, verändern und Design-Wünsche umsetzen).
- **Erzeugen eines NC-Programms** zum "direkten Abarbeiten" **aus den Aufzeichnungsdaten** (Teil kopieren, auf gleicher oder anderer Maschine).

Erforderliche Hardware-Komponenten für das Digitalisieren:

- Sercos Antrieb mit zusätzlichem Eingang für externes Messsystem.
- interne Messsysteme (Antriebe) zur Aufnahme der Echtzeitpositionen
- externes Messsystem (z. B. Laser) für die Online-Korrektur
- Ethernet-Netzwerk (TCP/IP) für die Datenübertragung auf PC
- NC mit möglichst großen freien RAM-Speicher
- PC-Panel für Bedienung
- freier Festplattenspeicher auf PC-Panel zum Speichern der Echtzeitdaten, falls kein externer PC (über NFS) vorhanden
- ext. PC mit NC vernetzt zum Speichern der Echtzeitdaten auf Festplatte.

Messfunktionen

Beispiel:

Aufbau und Verbindung der NC- und Antriebskomponenten zum Digitalisieren auf einer 3-Achs-Maschine.

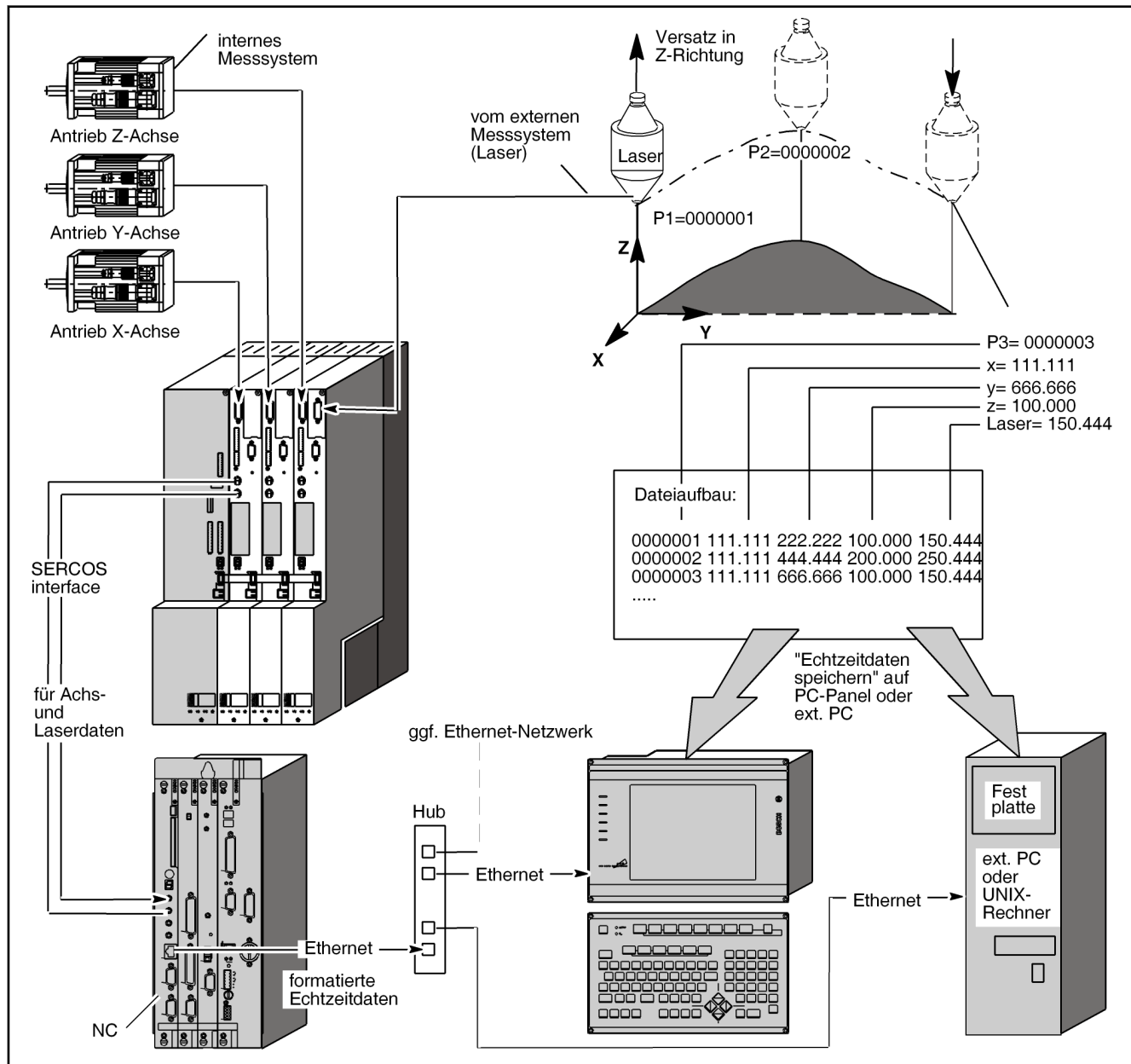


Abb. 15-1: Aufbau und Verbindung der NC- und Antriebskomponenten zum Digitalisieren

Vorgang des Digitalisierens:

- Der Vorgang des Digitalisierens muss im NC-Programm initialisiert werden. Das Digitalisieren selbst ist programmgesteuert, d. h. der Vorgang kann im Programm gestartet, unterbrochen, fortgesetzt und gestoppt werden.
- Zum Digitalisieren benötigt man einen Taster (z. B. Laser), der auf einer Mäanderbahn das Werkstück berührungslos abtastet (siehe folgende Abb.). Der Laser verfügt über ein eigenes Messsystem, das dazu benutzt wird, einen voreingestellten Abstand (Offset) zur Oberfläche wäh-

rend des Abtastens **konstant** zu halten, d. h. die Signale des Laser-Messsystems werden mit dem internen Messsystem der Laser-Zustellachse (z. B. Z-Achse) in Echtzeit verrechnet. Dadurch kann die Z-Achse der Höhenlinie der Oberfläche im IPO-Takt nachgeführt werden (**Höhenregelung**).



Erweiterte Möglichkeiten im Zusammenhang mit der (Laser-)Zustellachse (z. B. Kollisionserkennung, Locherkennung) stehen Ihnen durch den Einsatz der Funktion "Höhenregelung für Digitalisieren" zur Verfügung (siehe [Kap. 15.7 "Höhenregelung für Digitalisieren"](#) auf Seite 637).

Das "Externe Messsystem" (Laser) wird am Antriebsmodul eingespeist (siehe vorherige Abb.).

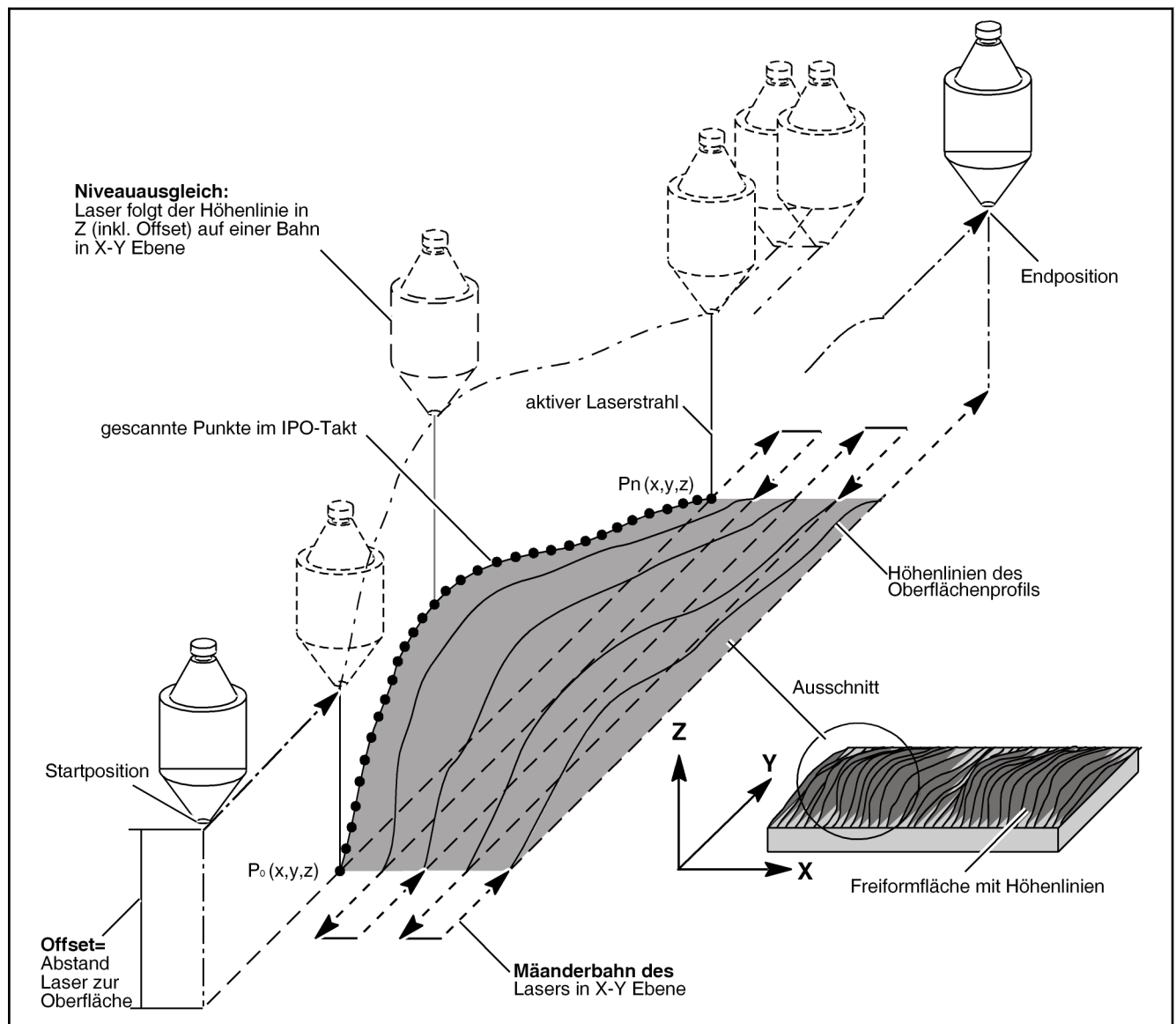


Abb. 15-2: Externes Messsystem

- Während des Abtastens werden alle erforderlichen Achspositionen (z. B. X, Y und Z-Achse) im Interpolatortakt (oder ein Mehrfaches davon) aufgenommen und auf die Festplatte eines PC-Panels oder eines externen PCs gespeichert.

Messfunktionen

- Der Vorgang des Abspeicherns wird über sog. Digitalisierungspuffer bewerkstelligt. Hierbei werden die vom Antrieb kommenden Echtzeit-Positionsdaten portionsweise in den freien RAM-Speicherbereich der NC geschrieben. Bereits "gefüllte" Puffer werden in der Ausgabedatei (siehe RECFILE) auf Festplatte eines angeschlossenen PC-Panels oder externen PCs (über Netzwerk) gespeichert. Die Zeitdauer des Speichervorgangs ist abhängig von der aktuellen Netzwerkbelastung bzw. der Systemauslastung der Steuerung.

In dieser Zeit können mehrere Puffer von dem "Echtzeitteil" (über Sercos ankommende Daten) gefüllt werden (siehe folgende Abb. "bereits gefüllte Puffer").

Beispiel:

Pufferung ankommender und abfließender Echtzeitdaten

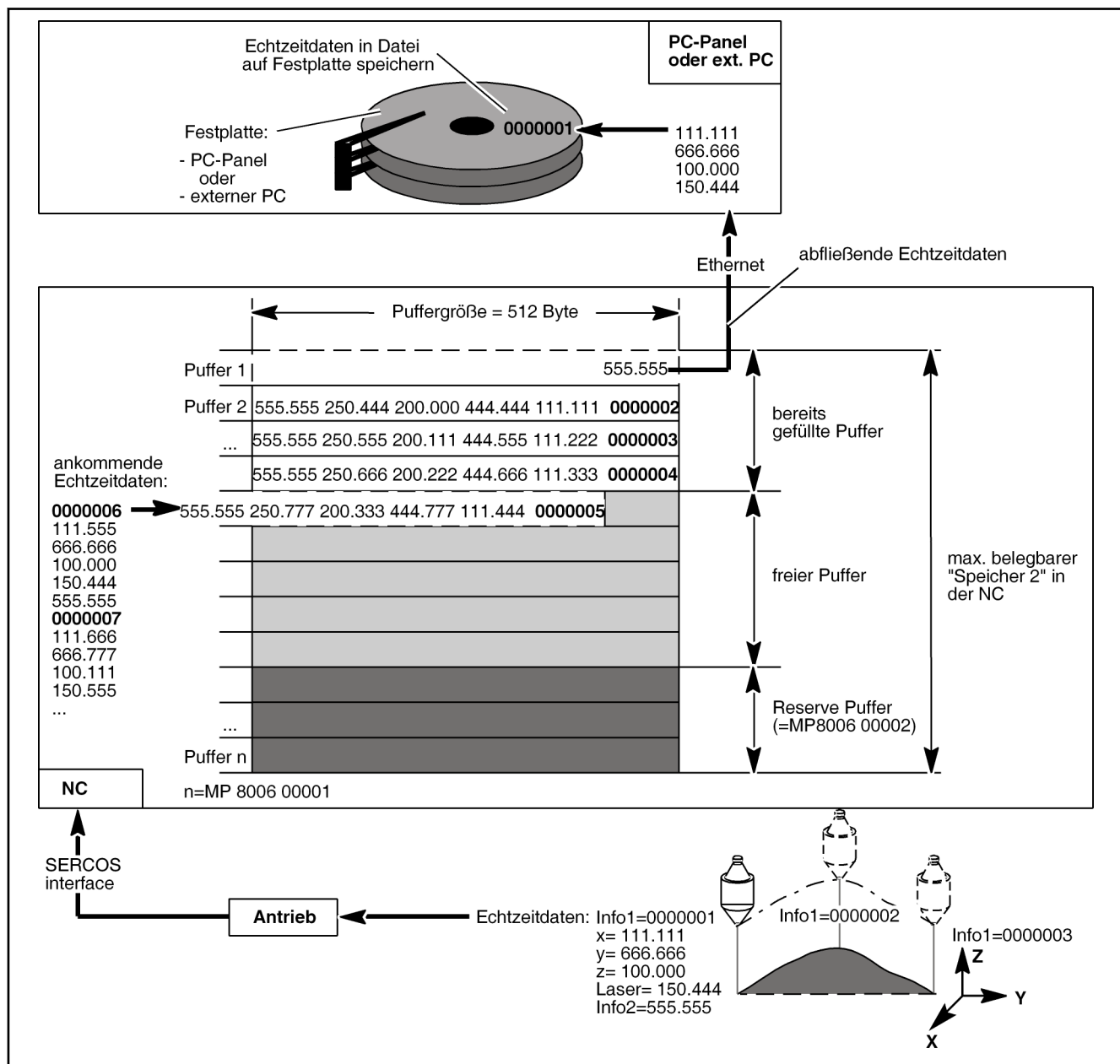


Abb. 15-3: Beispiel - Pufferung ankommender und abfließender Echtzeitdaten

Bei hoher Netzwerkbelastung gelangen die Daten nur verzögert auf die Festplatte. Dann entsteht ein Datenstau im Digitalisierungspuffer, d. h. es kommen mehr Daten an, als Daten abgespeichert werden.

Werden so viele Puffer benötigt, dass die Anzahl der noch freien Puffer die "Anzahl der Reservepuffer" unterschreitet, reagiert die Steuerung entsprechend der Parametrierung im Befehl

RECFILE(, [<File-I/O>]) und RECTIME(, [<AutoStop>],) des RecordSet:

- – Die Steuerung setzt den Kanalvorschub auf Null, so dass das Einlesen der Echtzeitdaten unterbrochen wird. Gefüllte Puffer können so zunächst gespeichert werden.
(siehe RECFILE(, <File-I/O=1 oder 2>))
- Bei jedem Stillstand der Kanalachsen wird das Einlesen der Echtzeitdaten automatisch unterbrochen. Dadurch können bereits gefüllte Puffer abgespeichert werden (siehe RECTIME(, <AutoStop=1>)).
- Wird die "Anzahl der Reservepuffer" nicht beachtet und sollten die Daten nicht rechtzeitig abgespeichert werden können, kommt es zum "Überlauf" der Digitalisierungspuffer.

Abhilfe:

Anzahl der Digitalisierungspuffer in MP 8006 00001 vergrößern (siehe RECFILE(, <File-I/O=0>) oder Parameter RECTIME in der NC-Funktion "RECORDSet" verändern.

Restriktionen

- Die Funktion "Digitalisieren" darf in der NC systemweit zu einem gegebenen Zeitpunkt nur einmal aktiv sein. Mehrmaliges Starten (z. B. in verschiedenen Kanälen) ist nicht erlaubt.
- Fehlerhafte Angaben (z. B. angegebene Identnummer ist nicht im zyklischen Achstelegramm konfiguriert) oder Probleme bei der Dateibearbeitung (z. B. nicht genügend Speicher auf der Festplatte) führen zu einem Laufzeitfehler.
- Existiert die gewünschte Aufzeichnungsdatei bereits, wird sie (je nach Parametrierung) durch eine neue Digitalisierung überschrieben.
- Achsnummern können systemweit ausgewählt werden. Es besteht keine Beschränkung auf die Achsen des Kanals, in dem das Digitalisierungsprogramm gestartet wurde.
- Eine aktive Digitalisierung wird spätestens beim Programmende beendet. Dabei wird die Aufzeichnungsdatei geschlossen.
- Mit der Anweisung RecordBreak wird die Digitalisierung angehalten, d. h. die Steuerung unterbricht das Einlesen der Echtzeitdaten. Die geöffnete Aufzeichnungsdatei bleibt weiterhin offen.
- Ist der Sollvorschub des Kanals und der Ist-Vorschub aller Achsen dieses Kanals gleich 0, wird die Erzeugung redundanter Daten vermieden (siehe Parameter RECTIME(...[AutoStop])).
- Ist durch zu hohe Netzwerklast das Abspeichern nicht möglich, wird der Vorschub des Kanals zu Null gesetzt (s.o.) und bei Stillstand das Aufzeichnen der Daten unterbrochen. Bereits gefüllte Puffer mit Echtzeitdaten können dann trotz eventueller Netzwerkbelastungen gespeichert werden (siehe RECFILE(, [File-I/O])).
- Bis zu 8 "Messpunkte" (Achsnummern, Achspositionen,...) können definiert werden (siehe RECPROBE1 ... RECPROBE8).

Messfunktionen

- Die erfassten Werte können mit einem Faktor für die weitere Datenaufbereitung (z. B. für ein CAD-System) gewichtet werden.
- Eine formatierte Ausgabe der erfassten Werte ist möglich.
- Es können bis zu 8 zusätzliche "Messpunkte" (z. B. Offsets,...) definiert werden (RECPROBE101 ... RECPROBE108) die sich arithmetisch (+/-) mit den RECPROBE1 ... RECPROBE8 verknüpfen lassen.
Diese Messpunkte werden nicht in der Datei abgespeichert.
- Bei fehlendem Umkehrspielausgleich im Antrieb (z. B. von Fremdhersteller) werden falsche Positionswerte ausgegeben.

"Digitalisieren" initialisieren:

Die Funktion "Digitalisieren" wird mit den NC-Funktionen "RecordSetProbe" und "RecordSet" initialisiert und steht am Anfang desjenigen NC-Programms, das die Verfahrbewegungen zum Erfassen der Werkstückoberfläche enthält.

Hierfür werden ein oder mehrere NC-Sätze mit RecordSetProbe zum Einstellen der Messpunkte gefolgt von einem NC-Satz mit RecordSet zum Einstellen des Speichervorgangs programmiert.

```
N10 RecordSetProbe RECPROBE1(..)
N20 RecordSetProbe RECPROBE2(..)
N30 RecordSet RECFILE(..) RECTIME(..)
```

Die NC-Funktion "RecordSetProbe"

- Mit Hilfe von RecordSetProbe werden die Messpunkte (Positionsdaten) **vor** dem Befehl RecordSet parametrisiert. RecordSetProbe kann mehrfach programmiert werden. Hierbei können bereits parametrisierte RECPROBE_x nicht überschrieben werden.
- Die Übernahme aller Messpunkte erfolgt mit dem Befehl RecordSet. Damit ist das Digitalisieren initialisiert und kann gestartet werden.

RecordSetProbe RECPROBE _x (..)

Tab. 15-16: NC-Funktion RecordSetProbe

Parameter von RecordSetProbe

RECPROBE _x (<Number>, <Name>, {<Factor>}, {<Format>}, {<±RECPROBE _y >})		
mit		
x: 1 .. 8 und 101 .. 108		
<Number>	0:	Für nicht von Achsen abgeleitete Messpunkte
	1..99	Achsnummer oder Angabe des physikalischen Achsnamens
<Name>	Mit dem Parameter <Name> werden die Positionsdaten der unter <Number> bezeichneten Achse übertragen Folgende Optionen existieren:	
	S-0-xxxx	Standard Sercos Parameter
	P-0-xxxx	Produktspezifischer Sercos Parameter

	I-ACTPOS	Antriebs-Istposition - LSEC - CCOMP - Temp.-Komp. in der Einheit 0.0001 mm
	I-HSINPUTx	Mit x: 1..8 High-Speed-Eingang der NC
	I-CMDPOS	Sollposition der Achse in der Einheit 0.0001 mm
	@aaaaa	Eine permanente CPL-Variable
<Factor>	Optional Die Einheit der ankommenden Positionsdaten wird mit einem Umrechnungsfaktor in eine andere Einheit umgerechnet (z. B. 1 mm = 0.001 mm). Die Positionsdaten werden in der neuen Einheit in die Aufzeichnungsdatei gespeichert. Das Ergebnis wird gerundet.	
<Format>	Optional, Default: "%d", Formatierung der ankommenden Positionsdaten in eine der nachfolgenden Optionen:	
	%d	4-Byte-Integer auch %<Länge>d. Beispiel: %6d
	%f	4-Byte-Float auch %<Länge>.<Nachkommastellen>f. Beispiel: %10.4f
	%e	4-Byte-Float (Exponentialdarstellung)
<±RECPROBEy>	Optional RECPROBEy (y=101, 102, ... ,108). Der Wert der Position wird um "+" oder "-" RECPROBEy korrigiert und das Ergebnis wird in der Aufzeichnungsdatei gespeichert (z. B. Offset des Lasers zur Oberfläche). Die Angabe der Korrektur erfolgt durch die RECPROBE-Nummer "y" (z. B. RECPROBE1(1, I-ACTPOS, , , +101) addiert den RECPROBE101 zu dem Probe1). Der korrigierte Wert wird in der Aufzeichnungsdatei gespeichert.	

Tab. 15-17: Parameter der NC-Funktion RecordSetProbe



- Ein als Messpunkt angegebener Sercos Parameter muss im zyklischen Antriebstelegramm vereinbart sein.
- RECPROBE101 ... RECPROBE108 werden nicht in der Aufzeichnungsdatei gespeichert. Diese Messpunkte (z. B. Offset) können nur zur Verrechnung mit RECPROBE1 bis RECPROBE8 verwendet werden.
- Bei gleichzeitiger Aufzeichnung von I-ACTPOS und I-CMDPOS muss berücksichtigt werden, dass sich die Messwerte auf verschiedene Zeitpunkte beziehen.

Die NC-Funktion "RecordSet"

Mit RecordSet wird der Speichervorgang und das Verhalten der NC während des Digitalisierens festgelegt.

- Alle Parameter von RecordSet müssen in eine Zeile geschrieben werden.
- Nach RecordSet ist das Digitalisieren initialisiert und kann gestartet werden.

RecordSet RECFILE (..) RECTIME (..) RECSEPARATOR (..)

Tab. 15-18: NC-Funktion RecordSet

Parameter von RecordSet

RECFILE (<Name>, {<Rewrite>}, {<File-I/O>})		
<Name>	Default: /mnt/<Dateiname> Name der Aufzeichnungsdatei inkl. Pfad (max. 100 Zeichen)	
{<Rewrite>}	Optional, Default: 0, Optionen: "Datei speichern" unter dem angegebenen <Name> überschreibt (=1) vorhandene Datei mit dem selben Namen oder wird abgelehnt (=0)	
{<File-I/O>}	Optional, Default: 1 Ist durch Netzwerkbelastungen das Abspeichern nicht möglich, wird der Vorschub des Kanals zu Null gesetzt. Sind die Kanäle im Stillstand, wird das Einlesen der Echtzeitdaten unterbrochen. Dadurch können bereits gefüllte Puffer abgespeichert werden. Die "Anzahl der Reservepuffer" muss groß genug sein, um alle während des Bremsvorgangs noch anfallenden Daten zwischenspeichern zu können.	
	0:	Der Vorschub wird nicht zu Null gesetzt
	1	Nach dem Abspeichern der gefüllten Puffer wird der Vorgang automatisch fortgesetzt
	2	Nach dem Abspeichern der gefüllten Puffer wird der Vorgang erst nach NC-Start fortgesetzt

{RECTIME (<Time>, {<AutoStop>}, {<Poti>}) }		
	RECTIME ist optional . Ist RECTIME nicht programmiert, wird <Time> auf die aktuelle NC-Zykluszeit, [<AutoStop>] und [<Poti>] auf Null gesetzt	
<Time>	0.5 bis 10000.0 ms. Angabe des Zeittaktes, indem die Positionsdaten gelesen werden sollen (z. B. im Interpolatortakt oder im ganzzahligen Vielfachen davon. Wert wird automatisch auf ein ganzzahliges Vielfaches vom Interpolatortakt aufgerundet	
{<AutoStop>}	Optional, Default: 0 Optionen:	
	0	Das Abspeichern wird nicht unterbrochen
	1	Die Digitalisierung unterbricht das Abspeichern der Daten, solange die Achsen des Kanals, der die Digitalisierung gestartet hat, stehen
{<Poti>}	Optional, Default: 0 Optionen:	
	0	Der Wert des Kanalpotis wirkt nicht
	1	In Abhängigkeit des Wertes des Kanalpotis wird die Abtastrate der Digitalisierung verändert (z. B. Poti 50%: Abtastzeit verdoppeln)
{RECSEPARATOR (<Char>) }		
	RECSEPARATOR ist optional . Ist RECSEPARATOR nicht programmiert, wird ein Tabulator als Trennzeichen verwendet	
<Char>	Trennzeichen zwischen den aufgezeichneten Positionsdaten, für den Fall, dass die Daten später exportiert werden sollen (z. B. in eine Microsoft Excel-Tabellendatei). Achtung: Geben Sie das Trennzeichen immer als CPL-String in der Form RECSEPARATOR ([" , "]) an! Empfohlene Zeichen sind: Leerzeichen: "" Tabulator: "\t" Komma: "," Semikolon: ";" Doppelpunkt: ":"	
	Bei Verwendung des "Punktes" als Trennzeichen kann es zu Fehlinterpretationen der Daten kommen, da im Fließkomma-Format ebenfalls der Punkt verwendet wird (z. B. 10.102)	

Tab. 15-19: Parameter der NC-Funktion RecordSet

"Digitalisieren" steuern: Mit den NC-Funktionen "RecordOn", "RecordOff", "RecordBreak", "RecordContinue" wird das Abspeichern der Echtzeitdaten während des Abarbeitens des NC-Programms beeinflusst.

NC-Funktion "RecordOn"

RecordOn	"Digitalisieren" starten. Aufzeichnungsdatei wird geöffnet. RecordOn startet das Erfassen und Speichern der Messwerte (Positionsdaten). Notwendige Parameter müssen mit RecordSet vorher gesetzt sein. Ist die Funktion "Digitalisieren" bereits aktiviert, führen weitere RecordSet-Anweisungen zu einem Laufzeitfehler.

Tab. 15-20: NC-Funktion RecordOn

NC-Funktion "RecordOff"

RecordOff	"Digitalisieren" beenden. RecordOff beendet das Aufzeichnen der Messwerte und schließt die Aufzeichnungsdatei. Nach dem Befehl RecordOff ist ein erneuter Start der Digitalisierung mit RecordOn oder eine weitere Initialisierung mit RecordSet(...) möglich. Weitere Funktionen zum Beenden des Digitalisierens: • Programmende • Kanal- oder Systemgrundstellung.

Tab. 15-21: NC-Funktion RecordOff

NC-Funktion "RecordBreak"

RecordBreak	"Digitalisieren" unterbrechen. RecordBreak unterbricht den Digitalisierungsvorgang, wobei das Erfassen und Abspeichern der Messwerte nicht mehr durchgeführt wird; die Aufzeichnungsdatei bleibt jedoch geöffnet. Der Vorschub ist hiervon nicht betroffen, so dass das NC-Programm weiter abgearbeitet wird. Wird während der Aufzeichnungsunterbrechung "Programmende" (z. B. M30), Grundstellung oder RecordOff erkannt, wird die Aufzeichnungsdatei geschlossen.

Tab. 15-22: NC-Funktion RecordBreak

NC-Funktion "RecordContinue"

RecordContinue {RECTIME(<Time>, {<AutoStop>}, {<Poti>}}}	"Digitalisieren" fortführen. RecordContinue führt den Digitalisierungsvorgang, d. h. Erfassen und Abspeichern der Messwerte fort. Mit der Programmierung von RecordContinue kann eine Abtastzeit angegeben werden. Dadurch ist es möglich, die in RECTIME angegebene Zeit zu ändern. Parameter RECTIME siehe NC-Funktion "RecordSet".

Tab. 15-23: NC-Funktion RecordContinue

Zusätzlich kann das Digitalisieren durch die folgenden Einstellungen beeinflusst werden:

- "Speichern der Echtzeitdaten" automatisch anhalten, wenn keine Verbewegung vorhanden ist:

- Einstellung in RecordSet(..., RECTIME(<Autostop=1>),...)
- Aufzeichnungsdatei automatisch bei Programmende schließen:
→ Programmieren von M30 im NC-Programm.

Relevante Maschinenparameter (MP)

8006 00001	Anzahl der Digitalisierungspuffer (Puffergröße 512 Byte). Die max. Anzahl der Digitalisierungspuffer ergibt sich nach der Formel: max. Anzahl Puffer = "Größe freier Speichers 2" / 512.
8006 00002	Anzahl Reservepuffer. Empfehlung: ca. 30% von MP 8006 00001. Bei Datenstau kann die Anzahl der Reservepuffer erhöht werden und/oder ggf. mit geringerem Vorschub gearbeitet werden.
4075 00102	Zuordnung der Digitalausgänge.

Tab. 15-24: Relevante Maschinenparameter

15.5.2 Handlungsanweisung: Digitalisieren (Aufnehmen von Messwerten)

Applizieren: IW-Engineering / Configuration: Parameter editieren

- Die Funktion "Digitalisieren" muss über den Parameter **dig** "Digitalisieren" im Setup (SUP) aktiviert werden.
- Im Parameter **NofDigBuff** "Anzahl der Digitalisierungspuffer" (8006 00001) ist die Anzahl der Digitalisierungspuffer (Puffergröße 512 Byte) einzutragen.
Wenn in der Aufzeichnungsdatei in der Sektion "[Statistics]" die Differenz zwischen LimitUsedBuffers und MaxUsedBuffers sehr klein ist, dann sollte der Wert von **NofDigBuff** erhöht werden.
- In dem Parameter **NofReservBuff** "Anzahl der Reservepuffer" (8006 00002) die Anzahl der Reservepuffer festlegen.
Empfehlung: ca. 30% vom Parameter **NofDigBuff** "Anzahl der Digitalisierungspuffer" (8006 00001). Bei Datenstau kann die Anzahl der Reservepuffer erhöht werden und/oder ggf. mit geringerem Vorschub gearbeitet werden.
- Im Parameter **DigIOAllocOut** "Zuordnung der Digitalausgänge" (4075 00102) den Wert "Digitalisieren" einstellen, somit kann der Digitalausgang zur Synchronisation der Aufzeichnung mit einem externen Gerät genutzt werden.

	Documentation	Instruction
Instruction:		Editieren von Maschinenparametern
Documentation:	MTX Maschinenparameter	Digitalisieren

IW-Engineering / Sercos: SCS-File der Phase 2 anpassen

Durch ein entsprechend konfiguriertes Achstelegramm (Sercos) steht sowohl der Achsistwert als auch der Sensoristwert in der NC zur Verfügung. Hierzu ist die Sercos Datei zu modifizieren, indem Sie die Sercos Parameter für den Sensor-Istwert des angeschlossenen externen Messsystems im Abschnitt für das zyklische Antriebstelegramm mit aufnehmen.

	Documentation	Instruction
Instruction:		Editieren von SCS-Parametern
Documentation:	Rexroth IndraDrive Parameterbeschreibung	Digitalisieren

Aktivieren: Voraussetzung:

RecordSet ist bzgl. der Maschinenkonfiguration eingestellt.

NC-Programm erstellen:

1. Name für NC-Programm wählen.
2. Den "parametrierten RecordSet" am NC-Programmbeginn einfügen.
3. Bahn programmieren.
4. Ab wann Positionsdaten aufgezeichnet werden sollen, legen Sie durch Programmieren des Befehls "RecordOn" fest.
5. Programmieren Sie den Befehl "RecordBreak", um das Digitalisieren temporär zu unterbrechen (z. B. zum Verfahren außerhalb der Werkstückoberfläche).
Programmierer Sie "RecordContinue", um das Digitalisieren fortzuführen.
6. Programmieren Sie das Ende des Digitalisierens mit "RecordOff" oder mit "Programmende M30".
7. **Abarbeiten:**
Starten des Digitalisiervorgangs mit Erfassen und Speicherung der Positionsdaten.



Stockt die Bewegung beim Abarbeiten, kann ein Übertragungsengpass beim Transfer der Echtzeitdaten über das Netzwerk die Ursache sein.

Hinweise über die Ursache der Störung finden Sie ggf. anhand der Daten aus dem Bereich "[Statistics]" in der Aufzeichnungsdatei.

Oftmals können solche Probleme durch Modifikation der Einstellungen in "RecordSet" gelöst werden.



Wenn in der Sektion "[Statistics]" der Aufzeichnungsdatei die Differenz zwischen LimitUsedBuffers und MaxUsedBuffers sehr klein ist, dann sollte der Wert des Maschinenparameters **NofDigBuff** "Anzahl der Digitalisierungspuffer" erhöht werden.

Beispiel:

1 - Programmierung der Istposition mit dem internen Messpunkt "I-ACTPOS" der NC und (S-0-0053)

```
RecordSetProbe RECPROBE1 (1,I-ACTPOS,0.0001,%9.3f)
```

```
RecordSetProbe RECPROBE2 (2,I-ACTPOS,0.0001,%9.3f)
```

```
RecordSetProbe RECPROBE3 (3,I-ACTPOS,0.0001,%9.3f)
```

```
RecordSetProbe RECPROBE4 (3,S-0-53,0.0001,%9.3f)
```

```
RecordSet      RECFILE (/mnt/digit/beispiel.dig,1,0)      REC-
TIME (8,1,0)
```

Die Parameter haben die Bedeutung:

- Die Echtzeitdaten werden in der Aufzeichnungsdatei "beispiel.dig" abgelegt. Sie ist im Pfad "/mnt/digit" gespeichert.
Eine bereits existierende Datei gleichen Namens wird vor Aufzeichnungsstart gelöscht.
- Die Erfassung erfolgt alle 8 msec.
- Da <File-I/O> gleich 0 ist, werden bei Netzwerkproblemen keine Maßnahmen getroffen, um einen Speicherüberlauf zu verhindern.
- Von den Achsen mit den physikalischen Achsnummern 1, 2 und 3 werden die korrigierten Istpositionen erfasst.
- Von der Achse 3 wird zusätzlich der "zweite" Istwert aufgezeichnet.
- Da <AutoStop> gleich 1 ist, werden im Stillstand der Achsen keine Daten aufgenommen.
- Da <Poti> gleich 0 ist, verändert das Poti nicht die Abtastzeit.

Beispiel:

2 - Programmierung der Istposition mit der Istposition des Motorgebers (S-0-0051)

```
RecordSetProbe RECPROBE1 (1,S-0-51)
RecordSetProbe RECPROBE2 (2,S-0-51)
RecordSetProbe RECPROBE3 (3,S-0-51)
RecordSet RECFILE (/mnt/digit/beispiel.dig,1) RECTIME (8)
```

Die Parameter haben die Bedeutung:

- Die Echtzeitdaten werden in der Aufzeichnungsdatei "beispiel.dig" abgelegt. Sie ist im Pfad "/mnt/digit" gespeichert. Eine bereits existierende Datei gleichen Namens wird vor Aufzeichnungsstart gelöscht.
- Die Erfassung erfolgt alle 8 msec.
- Von den Antrieben mit der physikalischen Achsnummer 1, 2 und 3 wird die Istposition des Motorgebers (S-0-51) erfasst.
- Da <File-I/O> nicht angegeben ist, wird bei Netzwerkproblemen der Vorschub auf Null reduziert. Nach dem Abspeichern bereits gefüllter Puffer wird das Teileprogramm automatisch fortgesetzt.
- Da <AutoStop> gleich 0 ist, werden auch im Stillstand der Achsen Daten aufgenommen.
- Da <Poti> gleich 0 ist, verändert das Poti nicht die Abtastzeit.

Beispiel:

3 - Programmierung der Istposition mit dem internen Messpunkt "I-ACTPOS" der NC und einem Offset (S-0-0053) (Lageistwert 2)

```
RecordSetProbe RECPROBE1 (X,I-ACTPOS,0.0001,%9.3f)
RecordSetProbe RECPROBE2 (Y,I-ACTPOS,0.0001,%9.3f)
RecordSetProbe RECPROBE3 (Z,I-ACTPOS,0.0001,%9.3f,-101)
RecordSetProbe RECPROBE101 (Z,S-0-53,[FAKTOR])
```

```
RecordSet      RECFILE (/mnt/digit/beispiel.dig,1,2)      REC-
TIME(8,1,0)  RECREPARATOR([","])
```

Die Parameter haben die Bedeutung:

- Von den Achsen mit den physikalischen Achsnamen X, Y und Z werden die korrigierten Istpositionen erfasst.
- Es wird dem RECPROBE101 die Identnummer S-0-53 (Lageistwert2) der Achse Z zugewiesen. Im Beispiel wird mit Hilfe der CPL-Variablen FAKTOR der Wert der Identnummer S-0-53 passend zu RECPROBE3 normiert.
- Bei der Achse Z wird der Wert von RECPROBE101 subtrahiert.
- Die Messwerte werden in der Datei "/mnt/digit/beispiel.dig" abgelegt. Eine bereits existierende Datei gleichen Namens wird vor Aufzeichnungsstart gelöscht.
- Da <File-I/O> gleich 2 ist, wird bei Netzwerkproblemen der Vorschub auf Null reduziert. Nach dem Abspeichern bereits gefüllter Puffer wird das Teileprogramm erst nach "NC-Start" fortgesetzt.
- Da <AutoStop> gleich 1 ist, werden im Stillstand der Achsen keine Daten aufgenommen.
- Da <Poti> gleich 0 ist, verändert das Poti nicht die Abtastzeit.
- Als Trennzeichen wird ein "," verwendet.

Deaktivieren: Der Digitalisiervorgang lässt sich beenden per

- RecordOff,
- Programmende,
- Kanalgrundstellung oder
- Systemgrundstellung.

Die Funktion "Digitalisieren" stellt die technischen Voraussetzungen zum Aufnehmen eines Oberflächenprofils und zum Abspeichern erfasster Achspositionen bereit.

15.6 Die Aufzeichnungsdatei

15.6.1 Beschreibung

Funktionen

Dateiart: Das Digitalisierungsergebnis wird als "Aufzeichnungsdatei" im ASCII-Format gespeichert.

Dateiname und Pfad: In "<Name>" des Parameters **RECFILE** (siehe Initialisierungsfunktion "RecordSet") Dateiname und Pfad angeben, unter dem die Aufzeichnungsdatei gespeichert werden soll.

Beispiel: <Name> entspricht /mnt/digit/beispiel.dig

Datei öffnen Aufzeichnungsdatei öffnen: Befehl **RecordOn**.

(Start der Aufzeichnung):

Ankommende Positionsdaten werden in der weiter unten angegebenen Struktur gespeichert.

Datei schließen
(Beenden der Aufzeichnung):

Aufzeichnungsdatei schließen: Befehl **RecordOff**.

Noch im Puffer befindliche Positionsdaten werden gespeichert.

Struktur der Datei:

Die Struktur der Aufzeichnungsdatei ist in drei Abschnitte gegliedert. Zur Kennung eines Abschnittsbeginns dienen folgende **Schlüsselwörter**:

1. **[Settings]:**

Enthält Informationen über Konfiguration, Aufzeichnungsdatei und Teileprogramm, mit dem das Werkstück digitalisiert wurde.

2. **[Data]:**

Enthält Echtzeitdaten, organisiert in Zeilen und Spalten.

Die Spalten enthalten diejenigen aufgezeichneten Werte (z. B. Achsposition der X-Achse) , die mit RECPROBE1..8 definiert wurden.

Jede Zeile enthält einen kompletten Satz von RECPROBE1..8 eines einzelnen Abtastzeitpunktes. Innerhalb der Daten werden programmierte Unterbrechungen mit RecordBreak und einem Kommentar gekennzeichnet. Ist in Verbindung mit dem Befehl RecordBreak kein Kommentar angegeben, entfällt dieser Eintrag komplett.

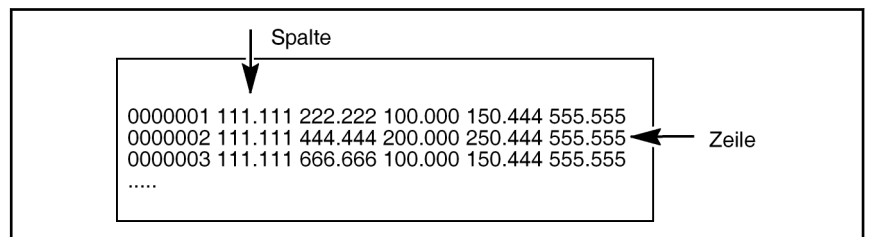


Abb. 15-4: Struktur der Datei

3. **[Statistics]:**

Enthält statistische Informationen, die während des Digitalisierens aufgetreten sind.

Detailbeschreibung des Dateiaufbaus:**[Settings]****[DATE]**

<Erstellungsdatum der Ausgabedatei>

[PROGRAM]

<Dateiname des Teileprogramms>

[RECFILE]

(siehe auch RecordSet RECFILE)

Name: <Dateiname>, Rewrite: <Rewrite>, File-I/O: <File-I/O>

[RECTIME]

siehe RecordSet RECTIME

RECTIME: <Time>, AutoStop: <AutoStop>, Poti: <Poti>

[RECPROBE1]

siehe RecordSet RECPROBE1

Axis: <Number>, Name: <Name>, Factor: <Factor>, Format: <Format>

....

[RECPROBE8]**[RECSEPARATOR]****[Data]**

<RECPROBE1(<Name>)> {<RECPROBE2(<Name>)} ... {<RECPROBE8(<Name>)} }

[Statistics]

NmbOfSamples	Anzahl der aufgezeichneten Zeilen
LimitUsedBuffers	Anzahl der maximal verfügbaren Puffer Die Echtzeitdaten der Digitalisierung werden in Puffern zwischengespeichert, bevor sie in die Datei geschrieben werden.
MaxUsedBuffers	Max. erreichte Pufferdistanz Das Abspeichern in der Datei erfolgt je nach System- oder Netzwerkbelastung sehr viel langsamer als das Erfassen der Echtzeitdaten. Hierdurch entsteht eine Pufferdistanz zwischen dem momentan beschriebenen Puffer in der NC und dem gerade gespeicherten Puffer auf der Festplatte.
DownSlopesDetected	Anzahl "Überschreiten der kritischen Pufferdistanz" Überschreitet die o.g. Pufferdistanz einen maximalen Wert (=MP 8006 00002), so versucht die Steuerung in Abhängigkeit des Parameters "File-I/O" die Achsbewegung anzuhalten, um dann das Digitalisieren zu unterbrechen.
DownSlopesExecuted	Anzahl der durchgeführten Bremsvorgänge bis zum Stillstand (siehe AutoStop). Nach Überschreiten der kritischen Pufferdistanz wird ein Bremsvorgang der Achsen ausgelöst. Nach Erreichen des Stillstandes wird die Digitalisierung unterbrochen, damit die Echtzeitdaten in die Aufzeichnungsdatei geschrieben werden können. Das NC-Programm kann fortgesetzt werden, wenn wieder freie Puffer vorhanden sind. Die ausgegebene Anzahl ist ein Maß für die System- bzw. Netzauslastung.
TimeAtEnd	Zeitpunkt "Ende der Digitalisierung".
Zusatzinformationen	nur für interne Zwecke

Tab. 15-25: Detailbeschreibung des Dateiaufbaus

Beispiel:

Inhalt der Aufzeichnungsdatei "beispiel.dig"

```

[Settings]
[DATE]
6.11.2000 10:35
[PROGRAM]
Example.cnc
[RECFILE]
Name: /mnt/digit/beispiel.dig, Rewrite: 1, File-I/O: 0
[RECTIME]
RECTIME: 8, AutoStop: 0, Poti: 0
[RECPROBE1]
Axis: X, Name: I-ACTPOS, Factor: 1, Format: %d ,
[RECPROBE2]
Axis: Y, Name: I-ACTPOS, Factor: 1, Format: %d ,
[RECPROBE3]
Axis: Z, Name: I-ACTPOS, Factor: 1, Format: %d ,
[RECPROBE4]
Axis: Z, Name: S-0-53, Factor: 1, Format: %d ,
[RECSEPARATOR]
<Tab>
[DATA]
111.111 222.222 100.000 150.444
111.111 444.444 200.000 250.444
111.111 546.345 122.000 350.444
...
...
[Statistics]
NumOfSamples: 1000
LimitUsedBuffers: 50
MaxUsedBuffers: 30
DownslopesDetected: 0
DownslopesExecuted: 0
TimeAtEnd: 7.11.2000 11:50
Info

```

15.7 Höhenregelung für Digitalisieren**15.7.1 Beschreibung****Funktion**

Beim Digitalisieren einer 3D-Kontur müssen die Koordinaten der betreffenden Oberfläche ermittelt und abgespeichert werden.

Dazu verwendet man ein Teileprogramm, das

1. zu Beginn eine Messvorrichtung (z. B. Laser) über die Zustellachse (z. B. Z) auf einen vorbestimmten Abstand zur Kontur bringt. Dieser Abstand dient im weiteren Verlauf als Bezugswert.
2. die Funktion "Höhenregelung für Digitalisieren" einschaltet
3. den Digitalisierungsvorgang (siehe [Kap. 15.5 "Digitalisieren \(Aufnehmen von Messwerten\)"](#) auf Seite 620) startet und
4. anschließend das zu digitalisierende Objekt in derjenigen Ebene, die durch die beiden anderen Achsen aufgespannt wird (z. B. X/Y-Ebene), z. B. mäanderförmig abfährt.

Mittels der Messvorrichtung werden Abstandsänderungen zum Bezugswert permanent erfasst, der NC zugeführt und bei Bedarf in die Einheit mm umgerechnet (Normierung siehe MP 7050 00720). Hierfür ist es erforderlich, die Messdaten in den externen Geberanschluss eines Antriebs einzuspeisen (z. B. als Inkrementalgeber-Signal) und per Sercos Interface zyklisch an die NC zu übermitteln.

Die NC kann dann nach Aktivieren der Höhenregelung die Position der Zustellachse so regeln, dass der zum Aktivierungszeitpunkt aktuelle Abstand zwischen Messvorrichtung und Objektoberfläche konstant bleibt (Abweichung zum Bezugswert = 0). Dadurch ist sichergestellt, dass der verfügbare Arbeitsbereich der Messvorrichtung nicht verlassen wird.

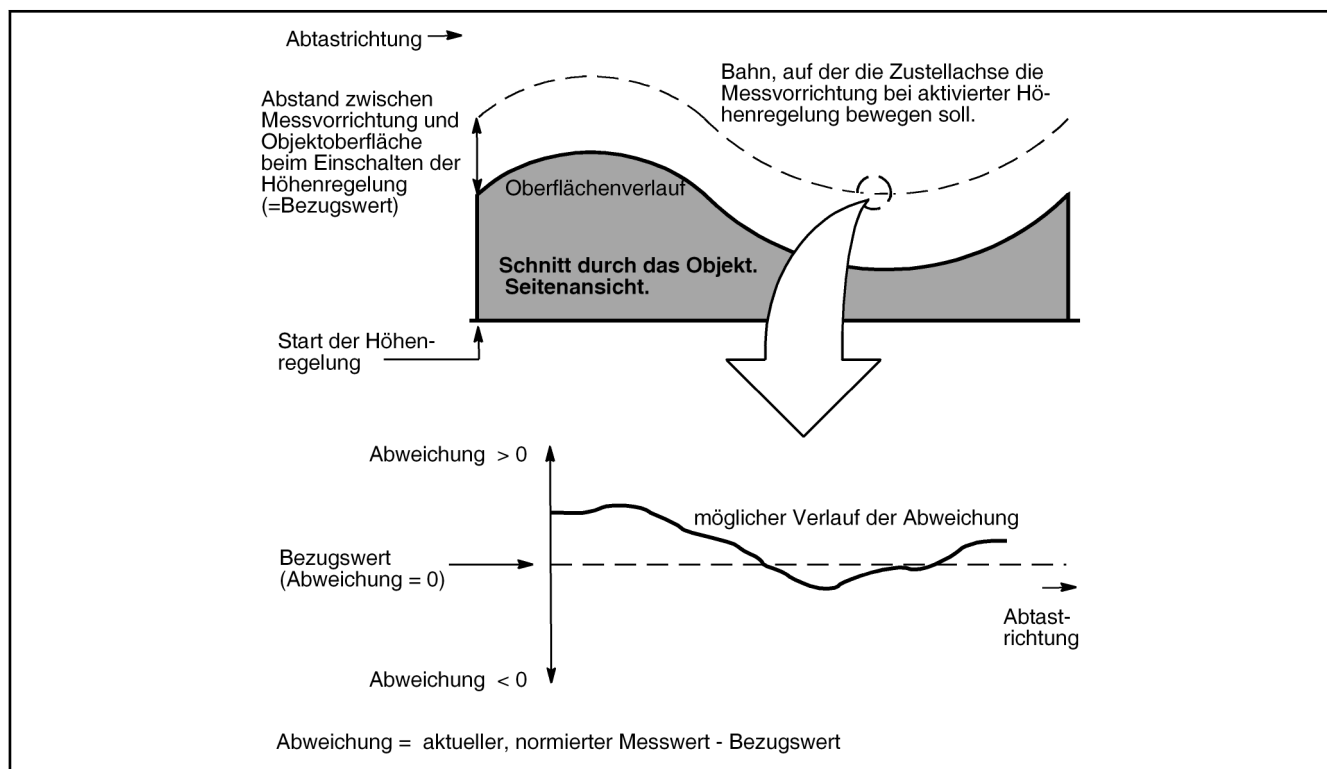


Abb. 15-5: Messvorrichtung

Im Rahmen der Funktion stehen Ihnen folgende Möglichkeiten zur Verfügung:

- programmgesteuertes Einschalten, Unterbrechen, Fortsetzen und Ausschalten der Höhenregelung. Siehe "Relevante NC-Funktionen".



Nach Steuerungshochlauf ist die Höhenregelung immer ausgeschaltet.

- programmgesteuertes Überblenden zugehöriger Maschinenparameter. Siehe "Relevante NC-Funktionen".



Überblendete Maschinenparameter-Konfigurationsdaten werden erst nach Programmabwahl, Kanal- oder Systemgrundstellung wieder wirksam!

- Überwachung der Abweichung per programmierbarem Toleranzband.
In diesem Zusammenhang ist eine Kollisions- und Locherkennung möglich (siehe [Kap. 15.7.2 "Abweichungen überwachen"](#) auf Seite 643).
- Begrenzung des aktuellen, normierten Messwertes bzgl. Geschwindigkeit und Beschleunigung.
Siehe MP 7050 00740 und MP 7050 00741.
- Glättung der eintreffenden Messwerte.
Unterdrückt "Messausreißer", siehe MP 7050 00730.

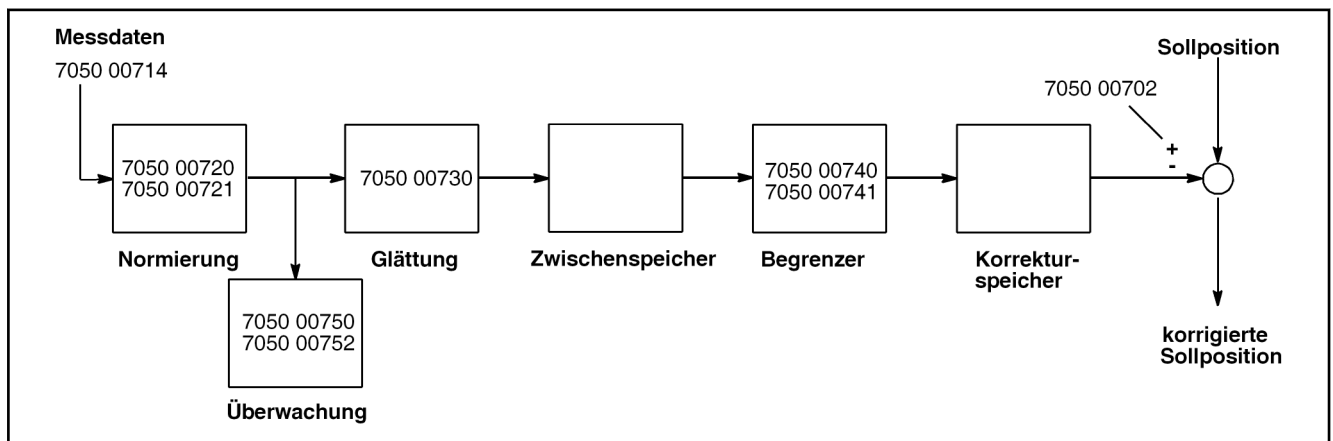


Abb. 15-6: Wirkungspunkte von Maschinenparametern in der Messdatenverarbeitungsstrecke

Restriktionen

- Bei eingeschalteter Höhenregelung darf die Zustellachse nicht aus dem Kanal entfernt werden (z. B. per Achsübergabe).
- Höhenregelung ist nur für solche Achsen möglich, die bei eingeschalteter Koordinatenprogrammierung nicht an einer Achstransformation beteiligt sind.
- Die Software-Endschalter der Zustellachse sind bei eingeschalteter Höhenregelung außer Funktion.
- Bei eingeschalteter Höhenregelung kann die Zustellachse nicht auf eine maschinenfeste Position gefahren werden.

Relevante NC-Funktionen

DistCtrl/DCR		Startet die Höhenregelung. Zusätzlich wird der aktuelle Abstand zwischen Messvorrichtung und Oberfläche als Bezugswert übernommen. Alleine programmiert, wirken die per Maschinenparameter definierten Konfigurationsdaten. Optional lassen sich über die Programmierung diverser Zusatzbefehle einige Konfigurationsdaten überblenden. Diese werden innerhalb einer von runden Klammern eingeschlossenen Parameterliste der Funktion "DistCtrl/DCR" programmiert:
DCAXIS(<Achse>,<Korr>)		
		Überblendet MP 7050 00702.
<Achse>		Name oder Nummer der Kanalachse, die höhen-geregelt werden soll.
<Korr>	+1	Korrekturwerte in positive Bewegungs-oder 1: richtung einrechnen
	-1:	Korrekturwerte in negative Bewegungs-richtung einrechnen
DCFILTER(<Zeit>)		
		Überblendet MP 7050 00730.
<Zeit>	0:	Filter aus
	>0:	Filter ein, Werte in ms
DCLIMIT([<Geschw>] [<Beschl>])		
		Überblendet MP 7050 00740 und 7050 00741.
<Geschw>		Überblendet MP 7050 00740. Eingabewert je nach aktivem Maßsystem(G71,G70) in der Einheit mm/min bzw. Inch/min
<Beschl>		Überblendet MP 7050 00741. Eingabewert je nach aktivem Maßsystem (G71,G70) in der Einheit m/s ² bzw. 1000 Inch/s ² .
DCMON([<Kollision>][<Loch>])		
		Überblendet MP 7050 00750 und 7050 00752.,
<Kollision>		Überblendet MP 7050 00740. Eingabewert je nach aktivem Maßsystem(G71,G70) in der Einheit mm/min bzw. Inch/min
<Loch>		Überblendet MP 7050 00741. Eingabewert je nach aktivem Maßsystem (G71,G70) in der Einheit m/s "2" bzw. 1000 Inch/s "2".

DcBreak/DCB	Unterbricht die Höhenregelung. Der aktuelle Korrekturwert bleibt aktiv.
DcCont/DCC	Setzt eine zuvor unterbrochene Höhenregelung wieder fort. Die NC regelt die Abweichung vom Bezugswert schnellstmöglich aus.
DistCtrl(0) oder DCR(0)	Beendet die Höhenregelung, übernimmt den aktuellen Korrekturwert und stoppt die Achsbewegung. Wird der Befehl zusammen mit einer Verfahrbewegung im gleichen Satz programmiert, schaltet die NC die Höhenregelung erst nach Ausführen der Bewegung ab.

Tab. 15-26: Relevante NC-Funktionen

Relevante CPL-Funktionen

PCS, WCS	Beide Funktionen berücksichtigen den Korrekturwert der Höhenregelung nicht!
MCS	Liefert die Position der Koordinate inklusive des Korrekturwertes.
SPOS	Liefert die Sollposition der Achse inklusive des Korrekturwertes.
APOS, PPOS, PCSPROBE, PROBE	Alle CPL-Funktionen, die einen auf der Istposition oder dem Messwert eines Antriebs basierenden Wert liefern, berücksichtigen den Korrekturwert der Höhenregelung.

Tab. 15-27: Relevante CPL-Funktionen

Relevante Sercos Parameter

S-0-1050.1.6	Konfigurationsliste AT Legt fest, welche Sercos Parameter an die NC zyklisch übertragen werden sollen. Hier muss der in MP 7050 00714 angegebene Sercos Parameter enthalten sein (z. B. S-0-0053 Lage-Istwert 2 inkrementell). Die Realisierung der zusätzlichen Auswertung des externen Gebers ist der jeweiligen Antriebsdokumentation zu entnehmen.

Tab. 15-28: Relevante Sercos Parameter

Relevante Kanal-IF-Signale

iAx_Custom1 bis iAx_Custom8 siehe "MP 7050 00708".

Relevante Maschinenparameter (MP)

7050 00700	Höhenregelung vorhanden 0: nein; 1: ja								
7050 00702	Höhenregelung: Nummer der Kanalachse und Korrekturrichtung. Legt fest, welche Achse des Kanals höhengeregelt werden soll. Eine Achsnummer • ohne oder mit positivem Vorzeichen bewirkt die Einrechnung der Korrekturwerte in positiver Bewegungsrichtung der höhengeregelten Achse. • mit negativem Vorzeichen bewirkt die Einrechnung der Korrekturwerte in negativer Bewegungsrichtung der höhengeregelten Achse.								
7050 00708	Höhenregelung: IF-Signal, Nummer des Kundenausgangs des Kanals Legt fest, an welchem Ausgang des betreffenden Kanal-Interface eine eingeschaltete Höhenregelung gemeldet werden soll (Ausgang gesetzt: Höhenregelung ist aktiv). <table border="1"> <tr> <td>-1:</td><td>keine Signalausgabe</td></tr> <tr> <td>0:</td><td>i Signal Ax_Custom1</td></tr> <tr> <td>:</td><td></td></tr> <tr> <td>7:</td><td>i Signal Ax_Custom8</td></tr> </table>	-1:	keine Signalausgabe	0:	i Signal Ax_Custom1	:		7:	i Signal Ax_Custom8
-1:	keine Signalausgabe								
0:	i Signal Ax_Custom1								
:									
7:	i Signal Ax_Custom8								
7050 00710	Höhenregelung: Sensorauswahl, Gerätegruppe								
7050 00712	Höhenregelung: Sensorauswahl, Nummer des Gerätes								
7050 00714	Höhenregelung: Sensorauswahl, Besondere Kennzeichnung (S-0-xxxx, P-0-xxxx, @xxxx) Gibt an, in welchem Antriebsparameter die von der Messvorrichtung kommenden Geberimpulse gezählt werden. Der hier eingetragene Sercos Parameter muss im Antriebstelegramm S-0-1050.1.6) ebenfalls enthalten sein.								
7050 00720	Höhenregelung: Sensorinkremente → mm - Faktor Dient (zusammen mit MP 7050 00721) zur Umrechnung von an die NC übermittelten Daten (z. B. Geberimpulsen) in die Einheit Millimeter.								
7050 00721	Höhenregelung: Sensorinkremente → mm - Divisor Siehe "MP 7050 00720".								
7050 00730	Höhenregelung: Filter - Glättungszeit in ms Parametriert ein Rechteckfilter Eingabe von 0 deaktiviert das Filter.								

7050 00740	Höhenregelung: Begrenzer - Geschwindigkeit in mm/min Maximale Geschwindigkeit, mit der sich der aktuelle, normierte Messwert ändern darf. Dadurch lässt sich z. B. beim Überfahren eines Loches verhindern, dass die Zustellachse mit ihrer maximal erlaubten Geschwindigkeit in das Loch eintaucht.
7050 00741	Höhenregelung: Begrenzer - Beschleunigung in m/s ² Maximale Beschleunigung, mit der sich der aktuelle, normierte Messwert ändern darf.
7050 00750	Höhenregelung: Überwachung - Kollisionserkennung in mm (0.000=Aus) Legt die Toleranzgrenze für die Kollisionserkennung fest (in mm). Überschreitet die Abweichung vom Bezugswert den hier angegebenen Wert, meldet die NC eine Kollision. Die Eingabe von 0.000 deaktiviert die Kollisionserkennung.
7050 00752	Höhenregelung: Überwachung - Locherkennung in mm (0.000=Aus) Legt die Toleranzgrenze für die Locherkennung fest (in mm). Überschreitet die Abweichung vom Bezugswert den hier angegebenen Wert, "friert" die NC die Höhenregelung ein. In diesem Zustand bleibt die Zustellachse an ihrer momentanen Position. Die Eingabe von 0.000 schaltet die Locherkennung aus.

Tab. 15-29: Relevante Maschinenparameter

15.7.2 Abweichungen überwachen

Die über Sercos Interface ankommenden Daten der Messvorrichtung rechnet die NC mittels MP 7050 00720 und MP 7050 00721 zunächst in die Einheit mm um.

Beachten Sie, dass zur korrekten Interpretation der Messdaten

- eine ordnungsgemäße Konfiguration der externen Geberschnittstelle im Antrieb erforderlich ist (S-0-0-0115, Bit 3 bzw. Bit 5),
- der Korrekturwert passend eingerechnet werden muss (per MP 7050 00702: Vorzeichen, Achsnummer).

Im Ergebnis müssen sich die gemeldeten Lagedaten:

- bei einer Abstandsvergrößerung zwischen Messvorrichtung und Oberfläche verkleinern, da die Oberflächenposition (bezogen auf die Messvorrichtung) absinkt.
- bei einer Abstandsverkleinerung zwischen Messvorrichtung und Oberfläche vergrößern, da die Oberflächenposition (bezogen auf die Messvorrichtung) ansteigt.

Der normierte Messwert wird jetzt auf die Einhaltung maximal zulässiger Toleranzgrenzen überprüft.

Grundlage für die Überprüfung ist der beim Einschalten der Höhenregelung aktive Abstand zwischen Messvorrichtung und Objektoberfläche. Dieser Bezugswert dient im Folgenden als Nulllinie eines Toleranzbandes.

Stimmt der aktuelle normierte Messwert nicht mit dem Bezugswert überein, bedeutet das eine positive bzw. negative Abweichung von der Nulllinie.

Das Überfahren eines Loches macht sich hierbei als sprunghafter Anstieg der Abweichung in negativer Richtung bemerkbar, da über den Geber

schlagartig eine niedrigere Position der abgetasteten Oberfläche gemeldet wird.

Im Gegenzug bewirkt eine unvermittelt ansteigende Kontur, dass sich die gemeldete Oberflächenposition abrupt erhöht, was einen sprunghaften Anstieg der Abweichung in positiver Richtung zur Folge hat.

Beispiel:

Abweichungen überwachen

- Beim Einschalten der Höhenregelung beträgt der Abstand zur Oberfläche 100 mm. Der Geber meldet also z. B. den normierten Wert 100 mm. Dieser Abstand wird ab jetzt als Bezugswert verwendet:

$$\text{Abweichung} = \text{aktueller, normierter Messwert} - \text{Bezugswert} = 100 \text{ mm} - 100 \text{ mm} = 0 \text{ mm}$$
- Die Oberflächenkontur sackt jetzt um 60 mm ab. Der Geber meldet die Position 40 mm:

$$\text{Abweichung} = \text{aktueller, normierter Messwert} - \text{Bezugswert} = 40 \text{ mm} - 100 \text{ mm} = -60 \text{ mm}$$

Die Abweichung wird sofort von der NC durch die Nachstellung der Zustellachse ausgeregelt. Dadurch baut sich die Abweichung auf 0 ab. Gleichzeitig vergrößert sich die vom Geber gemeldete normierte Position wieder auf 100 mm.
- Die Oberflächenkontur steigt jetzt abrupt von 40 mm auf 100 mm. Der Geber meldet die Position 160 mm:

$$\text{Abweichung} = \text{aktueller, normierter Messwert} - \text{Bezugswert} = 160 \text{ mm} - 100 \text{ mm} = +60 \text{ mm}$$

Die Abweichung wird sofort von der NC durch die Nachstellung der Zustellachse ausgeregelt. Dadurch baut sich die Abweichung auf 0 ab. Gleichzeitig verringert sich die vom Geber gemeldete normierte Position wieder auf 100 mm.

Der sprunghafte Anstieg der Abweichung in positive bzw. negative Richtung lässt sich zur **Kollisions- und Locherkennung** nutzen. Dazu wird die aktuelle Abweichung permanent mit MP 7050 00750 (Kollisionserkennung) und MP 7050 00752 (Locherkennung) verglichen.

Übersteigt die aktuelle Abweichung den jeweils parametrisierten Grenzwert, reagiert die NC wie folgt:

- nach dem Erkennen einer Kollision:
 die NC generiert eine Warnung und hält die Bewegung im Kanal per Vorschub-Halt an.
 Der Bediener kann, wenn keine Kollision aufgetreten ist, die Bewegung mit NC-Start fortsetzen (evtl. mit niedrigerer Vorschubpotistellung).
- nach dem Erkennen eines Loches:
 die NC unterbricht die Höhenregelung für die Dauer der Überschreitung (Wirkungsweise wie DcBreak/DCB). Das Objekt wird weiterhin abgetastet. Hierbei bleibt der aktuelle Korrekturwert aktiv, d. h. die Zustellachse ändert ihre Koordinate nicht.

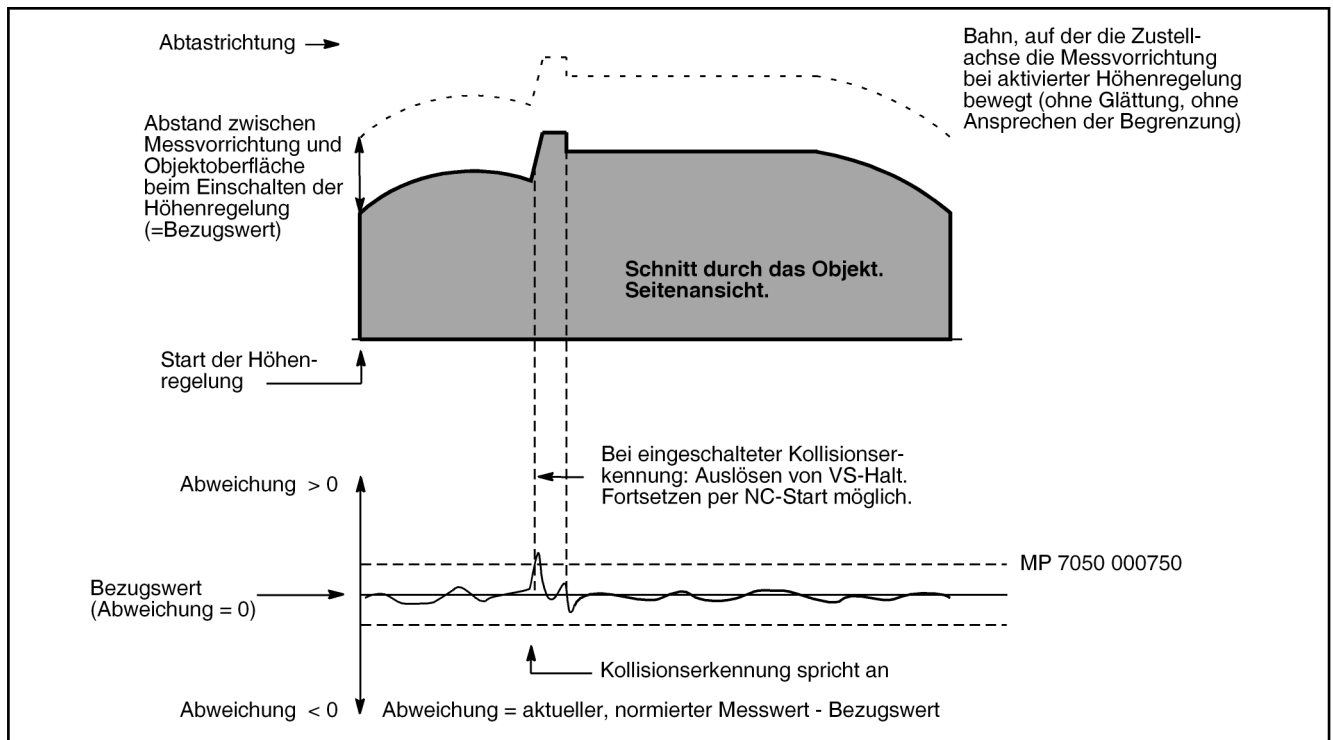


Abb. 15-7: Prinzip der Kollisionserkennung

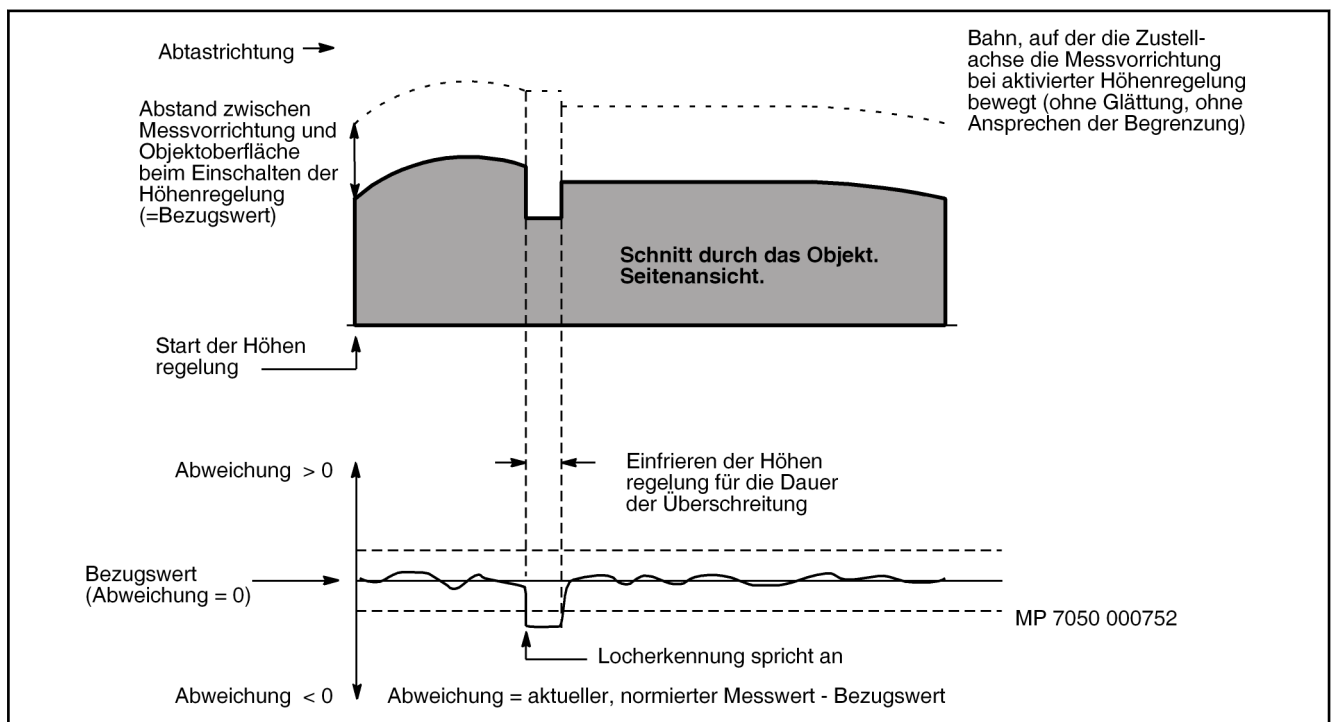


Abb. 15-8: Prinzip der Locherkennung

15.7.3 Handlungsanweisung: Höhenregelung für Digitalisieren

- Applizieren:** 1. Stellen Sie sicher, dass die Messdatensignale der Messvorrichtung für den externen Geberanschluss des Antriebs geeignet sind.

2. Relevante Sercos Parameter (siehe oben) im Antrieb korrekt parametrieren und ins Antriebs-EEPROM sichern oder im Zuge der Sercos Initialisierung von der Steuerung an den Antrieb übergeben.
Informationen zu Sercos siehe Dokumentation "MTX Funktionsbeschreibung Sonderfunktionen", Kap. "Funktionen zur Antriebsparametrierung".
3. Prüfen Sie, ob sich bei Abstandsänderungen zwischen Messvorrichtung und abgetasteter Oberfläche der externe Messwert korrekt ändert.
4. Nehmen Sie im Antriebstelegramm (S-0-1050.1.6) den Sercos Parameter auf, der den Lageistwert der Messvorrichtung enthält und zyklisch an die NC übertragen werden soll.
Prüfen Sie, ob in S-0-0015 das korrekte Telegramm konfiguriert ist. In S-0-1050.1.6 enthaltene Sercos Parameter werden nur dann an die NC übertragen, wenn in S-0-0015 das "frei konfigurierbare Telegramm" gewählt wird.
5. Konfigurieren Sie die unter "Relevante Maschinenparameter" angegebenen Parameter entsprechend Ihrer Applikation.
6. Steuerung und Antriebe neu hochlaufen lassen.

HINWEIS

Schäden an der Maschine oder am Werkstück möglich!

Stellen Sie sicher, dass die Korrekturrichtung (MP 7050 00702) richtig parametrier ist.



MP 7050 00702 und MP 7050 00730 bis 7050 752 können auch per NC-Funktion "DistCtrl" beeinflusst werden.

HINWEIS

Schäden an der Maschine oder am Werkstück möglich!

Während der Bearbeitung darf kein Steuerungs-Reset ausgelöst werden.

Ein-/Ausschalten:

Einschalten:

DistCtrl(...) bzw. DCR(...) (evtl. mit Zusatzparametern) programmieren.

Ausschalten:

DistCtrl(0) oder DCR(0) programmieren.

Die NC kann dann nach Aktivieren der Höhenregelung die Position der Zu-stellachse so regeln, dass der zum Aktivierungszeitpunkt aktuelle Abstand zwischen Messvorrichtung und Objektoberfläche konstant bleibt (Abweichung zum Bezugswert = 0). Dadurch ist sichergestellt, dass der verfügbare Arbeitsbereich der Messvorrichtung nicht verlassen wird.

IW-Operation: Messdatenverarbeitung vorbereiten

Die Messdaten sind in den externen Geberanschluss eines Antriebs einzuspeisen (z. B. als Inkrementalgeber-Signal) und per Sercos Interface zyklisch an die NC zu übermitteln.

IW-Operation / Programm: Sercos Datei der Phase 2 anpassen

- In den Sercos Parameter **S-0-0015** "Telegramm-Parameter" das "Frei konfigurierbare Telegramm" eintragen.
- In den Sercos Parameter **S-0-1050.1.6** "Konfigurationsliste Antriebs-Telegramm" den Sercos Parameter, der in dem Parameter **ScsParMcs-**

Corr "Sercos Parameter" (7050 00714) enthalten ist (z. B. S-0-0053 "La-geistwert 2 inkrementell"), eintragen.



Die Realisierung der zusätzlichen Auswertung des externen Ge-bers ist der jeweiligen Antriebsdokumentation zu entnehmen.

	Documentation	Instruction
Instruction:		Editieren von SCS-Parametern
Documentation:	Rexroth IndraDrive Sercos Parameterbeschreibung	Höhenregelung

IW-Engineering / Configuration: Parameter editieren

- Die Funktion Abstandsregelung muss über den Parameter **onl** "Abstandsregelung" im Setup (SUP) aktiviert werden.
- **EnaMcsCorr** "Aktiviere Abstandsregelung" (7050 00700)
- **ChAxMcsCorr** "Kanalachse" (7050 00702)
- **DirMcsCorr** "Korrekturrichtung" (7050 00702)
- **ChIfSigMcsCorr** "Interface-Signal" (7050 00708) legt fest, über welches Kunden-Kanal-Interfacesignal eine aktivierte Höhenregelung gemeldet wird.
- Rechte Maustaste auf den Knoten **Device** "Sensorauswahl"
- "Ändern in"
- Gewünschte Funktion wählen:
 - **Dr** "Antrieb" (7050 00710)
 - **CplVar** "Permanente CPL-Variable" (7050 00710)

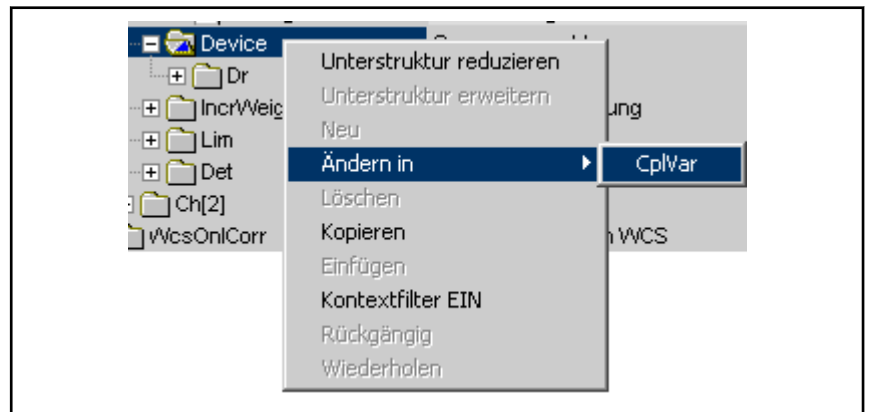


Abb. 15-9: Ändern der Sensorauswahl

- Je nach gewählter Funktion sind nur bestimmte der folgenden Parameter aktiv:
 - **DrIndMcsCorr**
"Antriebsnummer" (7050 00712)
 - **ScsParMcsCorr**
"Sercos Parameter" (7050 00714)
 - **CplVarMcsCorr**
"Variablenname" (7050 00714)
- **FactMcsCorr**
"Multiplikator" (7050 00720)

- **DivMcsCorr**
"Divisor" (7050 007210)
- **FilterMcsCorr**
"Glättung/Filter" (7050 00730)
- **MaxVelMcsCorr**
"Maximale Geschwindigkeit" (7050 00740)
- **MaxAccMcsCorr**
"Maximale Beschleunigung" (7050 00741)
- **CollDet**
"Kollisionserkennung" (7050 00750)
- **HoleDet**
"Locherkennung" (7050 00752)



Der Parameter **DirMcsCorr** "Korrekturrichtung" (7050 00702) und die Parameter **FilterMcsCorr** "Glättung/Filter" (7050 00730) bis **HoleDet** "Locherkennung" (7050 00752) können auch per NC-Funktion **"DistCtrl"** beeinflusst werden.

	Documentation	Instruction
Instruction:		Editieren von Maschinenparametern
Documentation:	MTX Maschinenparameter	Höhenregelung für Digitalisieren

IW-Operation / IndraLogic: Interface-Signale verknüpfen

Ist die Signalausgabe aktiviert, wird das Kanalinterface-Signal (**iAx_Custom1...iAx_Custom8**), das im Parameter **ChlfsigMcsCorr** "Interface-Signal" (7050 00708) deklariert ist, verknüpft.

	Documentation	Instruction
Instruction:		Editieren von SPS-Signalen
Documentation:	MTX SPS-Interface	Höhenregelung für Digitalisieren

15.7.4 Handlungsanweisung: Höhenregelung für Digitalisieren Parameter per CPL anpassen

Die NC kann nach Aktivieren der Höhenregelung die Position der Zustellachse so regeln, dass der zum Aktivierungszeitpunkt aktuelle Abstand zwischen Messvorrichtung und Objektoberfläche konstant bleibt (Abweichung zum Bezugswert = 0). Dadurch ist sichergestellt, dass der verfügbare Arbeitsbereich der Messvorrichtung nicht verlassen wird.

IW-Engineering / Configuration: Höhenregelung applizieren

		Instruction Kap. 15.7.3 "Handlungsanweisung: Höhenregelung für Digitalisieren" auf Seite 645
Instruction:		Höhenregelung applizieren

IW-Operation / NC-Programmierung: Werkzeuglängenkorrektur ändern

- **DistCtrl(<Fkt>)** bzw. **DCR(<Fkt>)** überblendet einige funktionspezifische Konfigurationsdaten in den Maschinenparametern.
 - mit -
<Fkt>
 - **DcAxis(<Achse>,<Korr>)** überblendet den Parameter **ChAxMcsCorr** "Kanalachse" (7050 00702) und den Parameter **DirMcsCorr** "Korrekturrichtung" (7050-00702)
Kurzform: **DCA(...)**
 - **<Achse>**
Name oder Nummer der Kanalachse, die höhengeregelt werden soll.
 - **<Korr>**
Bewegungsrichtung, in welche die Korrekturwerte eingerechnet werden sollen:
+1 oder 1: in positive Bewegungsrichtung
-1: in negative Bewegungsrichtung
 - **DcFilter(<Zeit>)** überblendet den Parameter **FilterMcsCorr** "Glättung/Filter" (7050 00730).
Kurzform: **DCF(...)**
 - **<Zeit>** Filterparametrierung zur Glättung der Sensorwerte.
0: Filter aus
>0: Filter ein; Glättungszeit in ms
 - **DcLimit({<Geschw>},{<Besch>})** überblendet den Parameter **MaxVelMcsCorr** "Max. Geschwindigkeit" (7050 00740) und den Parameter **MaxAccMcsCorr** "Maximale Beschleunigung" (7050 00741).
Kurzform: **DCL(...)**
 - **<Geschw>**
Maximale Änderungsgeschwindigkeit des Korrekturwertes. Eingabewert je nach aktivem Maßsystem (G71,G70) in der Einheit mm/min bzw. Inch/min.
 - **<Besch>**
Maximale Steilheit (Beschleunigung) des Korrekturwertes. Eingabewert je nach aktivem Maßsystem (G71,G70) in der Einheit m/s² bzw. 1000 Inch/s².
 - **DcMon({<Kollision>},{<Loch>})** überblendet den Parameter **CollDet** "Kollisionserkennung" (7050 00750) und den Parameter **HoleDet** "Locherkennung" (7050 00752). Kurzform: **DCM(...)**
 - **<Kollision>**
Toleranzband zur Kollisionserkennung. Eingabewert je nach aktivem Maßsystem (G71,G70) in der Einheit mm bzw. Inch.
0: Kollisionserkennung aus.
 - **<Loch>**
Toleranzband zur Locherkennung. Eingabewert je nach aktivem Maßsystem (G71,G70) in der Einheit mm bzw. Inch.
0: Locherkennung aus.
 - **DcBreak**

Unterbricht die Höhenregelung. Der aktuelle Korrekturwert bleibt aktiv. Kurzform: DCB

– **DcCont**

Setzt eine zuvor per DCB unterbrochene Höhenregelung wieder fort.

Die NC regelt die Abweichung vom Bezugswert schnellstmöglich aus.

Kurzform: **DCC**

	Documentation	
Documentation:	MTX Programmierhandbuch	Werkzeuglängenkorrektur

15.8 Messen am Festanschlag

15.8.1 Beschreibung

Während die Steuerung alle programmierten synchronen Achsen per Linearinterpolation im angegebenen Vorschub auf den programmierten Endpunkt fährt, wird bei einer ausgewählten Achse das aktuelle Drehmoment überwacht. Wird eine Drehmomentschwelle für diese Achse überschritten, werden die Bahnbewegungen auf v=0 abgebremst.

Die folgende Handlungsanweisung beschreibt das Messen am Festanschlag.

15.8.2 Handlungsanweisung: Messen am Festanschlag

IW-Engineering / Configuration:

- Die Funktion "Fahren auf Festanschlag" muss über den Parameter **MPS** "Fahren auf Festanschlag" im Setup (SUP) aktiviert werden
- **EnablMovePoStop** "Fahren gegen Festanschlag aktivierbar (1003 00030)"
- **MaxVelPoStop**
"Maximale Achsgeschwindigkeit (1005 00030)"
- **MaxTorqPoStop**
"Drehmomentgrenzwert (1003 00031)"

Instruction:		Editieren von Maschinenparameter
Documentation:	MTX Maschinenparameter	Messen am Festanschlag

IW-Operation / NC-Programmierung:

- **FsProbe(MfsAxis<i>)<Koordinaten><Vorschub>**
"Messen am Festanschlag"
- **FsProbe(MfsAxis(<k>, {<Schwelle>}))<Koordinaten><Vorschub>**
"Messen am Festanschlag" mit Drehmomentschwelle
 - <i>
Systemachsnummer der Messachse
 - <k>
Systemachsname oder -nummer der Messachse
 - <Schwelle>

Drehmomentschwelle Eingabewert in % vom Maximaldrehmoment. Ist <Schwelle> nicht programmiert, wirkt der Parameter **MaxTorqPoStop** "Drehmomentgrenzwert (1003 00031)"

– **<Koordinate>**

Position auf die während des Messvorgangs gefahren werden soll.

– **<Vorschub>**

Gewünschter Bahnvorschub. Wird begrenzt durch den Parameter **MaxVelPoStop** "Maximale Achsgeschwindigkeit (1005 00030)" und den Parameter **MaxVel** "Maximale Achsgeschwindigkeit (1005 00002)".



- Das Interface-Signal **iAx_FxStopAct** "Festanschlag aktiv" wird ausgegeben, solange die Verfahrbewegung zum Festanschlag aktiv ist.
- Im **FsProbe-Satz** sind folgende Funktionen nicht erlaubt:
 - **G75** (Messtaster)
 - **MOB**(Messen auf einem Satz)
 - **InitMeas/FlyMeas** (Fliegendes Messen)
 - **RedTorque** (Momentenreduzierung)
 - **FsMove / FsReset / FsTorque** (Fahren auf Festanschlag)

Documentation:	MTX Programmierhandbuch	Messen am Festanschlag

16 Synchronisationsfunktionen

16.1 Einleitung

In der MTX existieren verschiedene Instanzen die untereinander synchronisiert werden sollen bzw. müssen. Die wichtigsten Instanzen sind im folgenden Bild dargestellt.

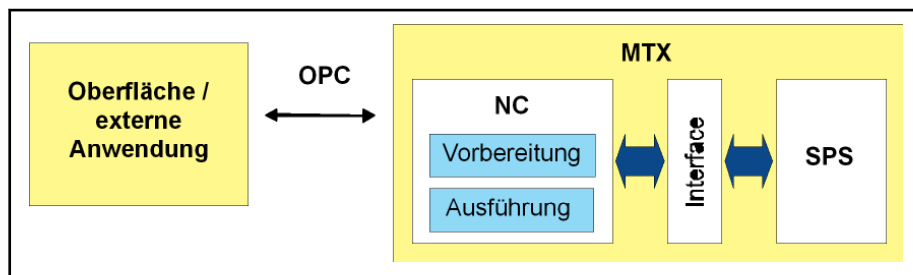


Abb. 16-1: Instanzen der MTX

Eine Synchronisation erfolgt immer zwischen zwei Teilnehmern. Ein Teilnehmer wartet dabei auf ein bestimmtes Ereignis, dass von dem anderen Teilnehmer in Abhängigkeit von seinem Zustand ausgelöst wird.

In diesem Kapitel werden Funktionen beschrieben, die zur Synchronisation der NC-Vorbereitung sowie der NC-Ausführung mit anderen Instanzen genutzt werden können. Dazu werden NC- oder CPL-Funktionen im Teileprogramm programmiert, die beim Abarbeiten Ereignisse auslösen oder die Abarbeitung anhalten und auf Ereignisse warten.

In Tabelle Tab. 16-1 "NC-Funktionen zum Anhalten der NC-Vorbereitung und der NC-Ausführung" auf Seite 653 sind die Funktionen aufgeführt, mit denen in Richtung NC-Vorbereitung bzw. NC-Ausführung synchronisiert werden kann. Das bedeutet, in einer Spalte stehen jeweils die Funktionen, mit denen die Vorbereitung oder die Ausführung auf ein Ereignis wartet.

		NC-Vorbereitung	NC-Ausführung
NC-Vorbereitung		-	PREP
NC-Ausführung	im Kanal	WAIT	-
	zwischen Kanälen	-	WEV, WPV
Interface		WAIT(BITIF(...))	WAITA/WAITO, Vor-schub halt
SPS		WAIT(BITIF(...))	WAITA/WAITO, Vor-schub halt
Positionen		-	ASTOPA/ASTOPO, BSTOPA/BSTOPO, WSTOPA/WSTOPO
Oberfläche (OPC)		-	-

Tab. 16-1: NC-Funktionen zum Anhalten der NC-Vorbereitung und der NC-Ausführung

Beispiel:

Wartefunktion in Tabelle finden

Die Ausführung von Sätzen soll an der Stelle "N1010" im Teileprogramm in Kanal 2 unterbrochen werden, bis die Achse X (die in Kanal 1 programmiert ist) die Achsposition 500 überschreitet.

Weil die NC-Ausführung warten soll, muss in der Tabelle in der Spalte "NC-Ausführung" gesucht werden. In der Zeile "Positionen" ist die NC-Funktion `ASTOPA` zu finden.

Die Aufgabe wird gelöst, indem im Teileprogramm von Kanal 2 die Zeile `N1010 ASTOPA(1,"X">500)` eingetragen wird. Damit hält die MTX die Ausführung des Kanals 2 in diesem Satz an, bis die Achsposition der X-Achse größer als 500mm ist.



Die meisten Funktionen, die während der NC-Ausführung warten (u.a. `WEV`, `WAITA`, `WAITO`, `WPV`, `WPVE` ...) bewirken implizit eine fallende Flanke am Ende des Satzes. Falsche gesetzte Synchronisationspunkte können zu Schäden an der Maschine führen.

Testen Sie den Programmablauf vor der eigentlichen Bearbeitung auf mögliche Synchronisationsprobleme im Programmablauf.

Die Funktionen, mit denen NC-Vorbereitung und NC-Ausführung an andere Instanzen ein Synchronisationsereignis senden können, sind in Tabelle [Tab. 16-2 "NC-Funktionen zum Senden von Signalen aus der NC-Vorbereitung und der NC-Ausführung"](#) auf Seite 654 aufgeführt.

Beispiel:

Sendefunktion in Tabelle finden

Sobald die NC-Vorbereitung den Satz "2020" erreicht, soll die Oberfläche ihre Anzeige ändern. Weil die NC-Vorbereitung das Ereignis auslösen soll, muss in der Tabelle in der Spalte "NC-Vorbereitung" gesucht werden. In der Zeile "Oberfläche (OPC)" ist der Eintrag `SD (CPL)` zu finden.

Die Aufgabe wird gelöst, indem

- ein Systemdatum "SignalVar" angelegt wird,
- die Oberfläche das OPC-Item "NC.SystemDataEvent" abonniert und "NC.SystemData_U" entsprechend auswertet (siehe Dokumentation "MTX OPC-Kommunikation", Kap. "Items") und
- im Teileprogramm die Zeile `2020 SD.SignalVar=10` eingetragen wird.

Damit wird das Systemdatum von der NC-Vorbereitung auf den Wert 10 gesetzt, wenn sie den Satz `2020` erreicht. Über den Systemdatenserver wird die Oberfläche in diesem Moment aktiviert und kann den Wert abfragen bzw. entsprechend reagieren.

		NC-Vorbereitung	NC-Ausführung
NC-Vorbereitung		-	-
NC-Ausführung	im Kanal	-	-
	zwischen Kanälen	perm. CPL-Variablen	SEV, SPV

	NC-Vorbereitung	NC-Ausführung
Interface	BITIF	verschiedene NC-Funktionen
SPS	BITIF, PLC, SD (CPL)	M-Funktionen, SSD
Oberfläche (OPC)	SD (CPL)	SSD, SSDQ

Tab. 16-2: NC-Funktionen zum Senden von Signalen aus der NC-Vorbereitung und der NC-Ausführung



Es gibt einzelne NC-Funktionen (z.B. G61 (IPS2)), die aufgrund ihres Zustandes Bits im Interface setzen. Auf diese Bits können dann verschiedene Instanzen reagieren.

16.2 Synchronisation mit der NC-Vorbereitung

16.2.1 NC-Vorbereitung und NC-Ausführung (WAIT)

Beschreibung

Funktion

Die Hauptaufgabe der NC-Steuerung besteht darin, Teileprogramme abzuarbeiten und die darin enthaltenen Befehle (NC-Sätze) in Bewegungen der Antriebe umzusetzen. Dazu müssen die einzelnen Sätze in einem ersten Schritt interpretiert und vorbereitet werden. Dieser Schritt wird als **NC-Vorbereitung** bezeichnet.

Aktuell vorbereitete Sätze werden in einem Puffer (dem Satzvorbereitungspuffer) zwischengespeichert. Die Größe dieses Puffers wird durch den Maschinenparameter `NCO/LookAh/Ch[k]/NofBlkPrep` (MP 7060 00110) vorgegeben. Verschiedene Funktionalitäten der MTX benötigen eine gewisse Größe dieses Puffers. Ein Beispiel dafür ist die Satzvorausschau.

Immer wenn ein Satz fertig abgearbeitet (bzw. gefahren) wurde, holt sich die **NC-Ausführung** den nächsten Satz aus dem Satzvorbereitungspuffer. Die NC-Ausführung erzeugt dann auf Basis der Vorbereitungsdaten Sollwerte für die Antriebe. Die Sollwerte werden schließlich mit dem Antriebsbus an die Antriebsregler geschickt. Das bedeutet, dass die programmierte Bewegung erst zum Ausführungszeitpunkt stattfindet.



Alle CPL-Sätze (Sätze die mit einer Zahl beginnen) sind Sätze der NC-Vorbereitung. Alle Befehle und Berechnungen dieser Sätze werden zur Vorbereitungszeit durchgeführt. Siehe auch Dokumentation "MTX Programmierhandbuch", Kap. "Grundsätzliches zur Standard- und CPL-Programmierung".

Die NC-Vorbereitung eines Satzes kann deutlich früher (bis zu mehreren Minuten) erfolgen als dessen Ausführung. Weil der zeitliche Abstand von vielen verschiedenen Faktoren abhängt, kann er für einen spezifischen NC-Satz nicht vorhergesagt werden. Er ist, im Allgemeinen, auch bei wiederholter Ausführung eines Teileprogramms nicht konstant für einzelne Sätze.

Aus diesem Grund muss eine Synchronisation zwischen Vorbereitung und Ausführung stattfinden, wenn auf aktuelle maschinen- oder prozessbezogene Zustände des Systems zugegriffen werden soll. Zu dieser Synchronisation wird der Befehl **WAIT** (ohne Parameter) genutzt.

Der WAIT-Befehl blockiert die Satzvorbereitung, bis der unmittelbar davor programmierte NC-Satz komplett abgearbeitet wurde. Die Satzvorbereitung

wird dadurch mit dem aktiven Zustand (Ausführungszeitpunkt) der Steuerung synchronisiert. Anschließend wird mit der Satzvorbereitung des nachfolgend programmierten Satzes fortgefahren. Zu diesem Zeitpunkt existieren keinerlei vorbereitete Sätze mehr, da diese zuvor alle abgearbeitet wurden.



Die Funktion `WAIT` (ohne Parameter) kann sowohl in Standard-NC-Sätzen (DIN), als auch in CPL-Sätzen programmiert werden. In einem CPL-Satz mit einer `WAIT`-Anweisung darf kein ":" programmiert werden. Nachfolgende CPL-Anweisungen müssen in einem neuen CPL-Satz geschrieben werden.

Relevante NC-Funktionen

N10 `WAIT` ; `WAIT` als NC-Funktion

Relevante CPL-Funktionen

10 `WAIT` : `REM WAIT` als CPL-Funktion

Relevante Maschinenparameter (MP)

Die Größe des Satzvorbereitungspuffers wird durch den Maschinenparameter `NCO/LookAh/Ch[k]/NofBlkPrep` (MP 7060 00110) vorgegeben.

Handlungsanweisung

Im folgenden Beispiel soll die aktuelle Werkstückposition der Achse X als Meldung ausgegeben werden.

Weil die CPL-Funktion `WCS` zum Vorbereitungszeitpunkt abgearbeitet wird, die Achse aber zum Ausführungszeitpunkt bewegt wird, ist eine Synchronisation zwischen beiden Instanzen notwendig.

```
N10 G1 F2500 X234      ; Verfahrbewegung für X
20 WAIT                ; Synchronisation (Vorbereitung=Ausführung)
30 XPOS = WCS("X")    ; Position zum Vorbereitungszeitpunkt ermitteln
40 PRN#(0, "X = ", XPOS); Position ausgeben
```

Ohne den Satz 20 würde die Position in Satz 30 zum Vorbereitungszeitpunkt ermittelt, während noch die Ausführung von N10 aktiv wäre. Das bedeutet, dass in diesem Fall nicht "X=234" sondern ein anderer Wert ausgegeben werden würde.

16.2.2 NC-Vorbereitung und Interface/SPS (WAIT(BITIF(...)))

Beschreibung

Funktion

Viele Prozesse und Abläufe werden in der Werkzeugmaschine von der SPS gesteuert. Deshalb ist es notwendig, NC und SPS zu synchronisieren. Die Schnittstelle zwischen beiden Instanzen ist das Interface. Eine detaillierte Beschreibung des Interface ist in der Dokumentation "MTX SPS-Interface" zu finden.

Wie bereits in [Kap. 16.2.1 "NC-Vorbereitung und NC-Ausführung \(WAIT\)" auf Seite 655](#) beschrieben wurde, existieren in der MTX zwei verschiedene Zeitpunkte für Vorbereitung und Ausführung eines Satzes. Beide Zeitpunkte können deutlich (mehrere Minuten) voneinander getrennt sein. Weil dieser Abstand von vielen verschiedenen Faktoren abhängt, kann nicht vorhergesagt werden, wie groß er in einem Teileprogramm ist.

Wenn auf aktuelle maschinen- oder prozessbezogene Zustände des Systems in der NC-Vorbereitung (z. B. mit CPL-Funktionen) zugegriffen werden

soll und diese von der SPS abhängen, muss eine Synchronisation zwischen NC-Vorbereitung und SPS erfolgen.

Dazu setzt die SPS ein Bit im Interface, um zu signalisieren, dass der entsprechende Systemzustand erreicht ist. Im Teileprogramm muss die CPL-Funktion `WAIT(BITIF(...))` programmiert werden, damit sie an diesem Satz auf das OK der SPS wartet.

Relevante CPL-Funktionen

Mit der CPL-Funktion `WAIT(BITIF(...))` wartet die NC-Vorbereitung auf ein SPS-Signal.

Relevante IF-Signale

Es kann auf jedes Bit im Interface gewartet werden.

Handlungsanweisung

Im folgenden Beispiel soll die Achsposition der asynchronen Achse XA ermittelt werden, wenn eine Laserschranke (die von der SPS verwaltet wird) ausgelöst hat. Die SPS schreibt dazu den Zustand der Laserschranke auf den 4. Kundenausgang des entsprechenden Kanals (`qCh_Custom4`).

SPS-Teil Im folgenden Programmausschnitt wird der Kundenausgang von der SPS beschrieben. Das Beispiel ist in "Strukturiertem Text (ST)" geschrieben. Der Baustein soll zyklisch ausgeführt werden.

```
VAR_INPUT                                // Deklarationsteil
LaserInput : BOOL;
END_VAR
```

```
qCh_Custom4 := LaserInput                // Schreiben des Ausgangs
```

Tab. 16-3: Programmbeispiel SPS-Programm, kopieren des Laserschrankeeingangs auf den Kundenausgang 4

NC-Teil Die Vorbereitung der NC muss nun angehalten werden, bis der Kundenausgang 4 im Interface von der SPS gesetzt wurde. Dazu wird der CPL-Befehl `WAIT(BITIF(...))` verwendet.

Falls ein Fehler in der Lichtschranke auftritt, wird ein Timeout von 2 Sekunden definiert. Kommt das Signal der SPS nicht innerhalb dieser Zeit bei der NC an, wird eine Fehlermeldung ausgegeben. Damit bleibt das System nicht in einem undefinierten Zustand stehen, sondern gibt eine Fehlermeldung an den Anwender weiter.

```
10 WAIT(BITIF("qCh_Custom4"), 2000, ERG%)           ; Warten auf Kundeneingang 4
20 IF ERG%=0 THEN                                   ; Kam das Signal?
30 XAPOS = ACS("XA")                                ; Position ermitteln
40 ENDIF
50 IF ERG%=2 THEN                                   ; Kam der Timeout?
60 SETERR("Lasersignal fehlt!")                     ; Fehler ausgeben
70 ENDIF
```

Tab. 16-4: Programmbeispiel CPL-Programm, Warten auf den Kundeneingang 4

16.2.3 Interface/SPS und NC-Vorbereitung (BITIF, PLC, SD)

Beschreibung

Funktion

Viele Prozesse und Abläufe werden in der Werkzeugmaschine von der SPS gesteuert. Deshalb ist es notwendig, NC und SPS zu synchronisieren. Die Schnittstelle zwischen beiden Instanzen ist das Interface. Eine detaillierte Be-

schreibung des Interface ist in der Dokumentation "MTX SPS-Interface" zu finden.

Im Gegensatz zu [Kap. 16.2.2 "NC-Vorbereitung und Interface/SPS \(WAIT\(BITIF\(...\)\)\)"](#) auf Seite 656 wird hier der Weg von der NC-Vorbereitung zur SPS beschrieben. Das bedeutet, die NC-Vorbereitung sendet ein Signal (via Interface), dass von der SPS empfangen wird. Diese kann ihren Ablauf dann entsprechend anpassen.

Wenn das Signal aus der NC-Vorbereitung kommen soll, gibt es drei mögliche Funktionen, die benutzt werden können:

- BITIF für Signale ins Interface,
- PLC für Signale in Merker und
- Systemdaten für komplexere Informationen.

Die Bits des Interface sind in der SPS als globale Variablen bekannt und müssen nicht extra deklariert werden. Die SPS fragt zyklisch den Zustand des Interface-Bits bzw. des Merkers ab und erkennt dadurch das Signal der NC-Vorbereitung.

Bei Verwendung von Systemdaten ist das Verhalten anders:

- keine Auswirkungen auf das Interface
- SPS kann Systemdaten (per Funktionsbaustein "MT_SD_RD", beschrieben in der Dokumentation "MTX SPS-Interface", Kap. "Systemdaten") direkt abfragen
- Systemdaten müssen in der NC extra angelegt werden
- Systemdaten können sowohl einzelne Werte (u.a. INTEGER, REAL) als auch komplexe Strukturen enthalten

Restriktionen

PLC kann nur dann zum Schreiben von Merkern genutzt werden, wenn der Maschinenparameter PLC/Plc auf IL (= IndraLogic) (MP 2060 00200 auf den Wert 5) eingestellt ist.

Relevante CPL-Funktionen

- BITIF für Signale ins Interface
- PLC für Signale in Merker
- CPL-Zuweisungen für Werte in Systemdaten

Relevante Maschinenparameter (MP)

Maschinenparameter PLC/Plc muss auf IL (= IndraLogic) (MP 2060 00200 auf den Wert 5) gestellt sein, damit PLC auf Merker schreiben darf.

Relevante IF-Signale

Mit BITIF können alle Interface-Eingänge geschrieben werden.

Handlungsanweisung Interface

In diesem Beispiel soll die NC der SPS mitteilen, wenn das Programm angefahren ist. Dazu soll die NC-Vorbereitung am Beginn des Programms den Kundeneingang 2 setzen. Die SPS wird dann eine LED anschalten.



Statt BITIF oder PLC zu nutzen, kann in diesem speziellen Beispiel auch der Eingang "iCh_ProgRun" im Interface von der SPS abgefragt werden, um zu ermitteln, ob ein Teileprogramm abgearbeitet wird.

NC-Teil Die Vorbereitung läuft an, sobald das Programm startet. Wird dann der `BITIF`-Befehl erreicht, setzt die NC-Vorbereitung das entsprechende Bit.

```
10 BITIF("iCh_Custom2") = TRUE ; Eingang setzen
```

Tab. 16-5: Programmbeispiel NC-Programm, setzen des Kundenausgangs 2 mittels BITIF

SPS-Teil Die SPS fragt in einem Funktionsblock zyklisch den Zustand des Interface ab. Sobald der Kundeneingang 2 aktiv ist, wird ein interner Zustand umgeschaltet und der Ausgang für die LED aktiviert.

```
VAR_OUTPUT ; Deklarationsteil
LEDOuput : BOOL;
END_VAR

IF iCh_Custom2 THEN ; Prüfen des Eingangs
LEDOuput := TRUE ; LED einschalten
InterState := 4 ; internen Status umsetzen
END_IF;
```

Tab. 16-6: Programmbeispiel SPS-Programm, kopieren des Laserschrankeeingangs auf den Kundenausgang 4

Handlungsanweisung Merker

Dieses Beispiel hat die gleiche Anforderung wie das vorangegangene, soll aber statt des Interfaces einen Merker nutzen. Der Merker (RunningState) wird als globale Variable vom Typ Byte angelegt. Die Merkeradresse soll Byte 23 sein. Weil dieser Merker auch für andere Signale benutzt werden soll, wird definiert, dass ein Wert von 7 bedeutet, dass das Programm ange laufen ist.

```
VAR_GLOBAL ; Deklarationsteil
RunningState AT %MB23 : BYTE;
END_VAR
```

Tab. 16-7: Programmbeispiel SPS-Programm, kopieren des Laserschrankeeingangs auf den Kundenausgang 4

NC-Teil Die Vorbereitung läuft an, sobald das Programm startet. Wird dann der `PLC`-Befehl erreicht, schreibt die NC-Vorbereitung den Wert auf den Merker. Wichtig ist dabei, dass die gleiche Adresse wie in der Deklaration benutzt wird.



Schreibender Zugriff auf Merker ist erlaubt, wenn der Maschinenparameter PLC/Plc auf IL (= IndraLogic) (MP 2060 00200 auf den Wert 5) eingestellt ist.

```
10 PLC(3, , 23, 1) = 7 ; Merker an Byte 32 auf 7 setzen
```

Tab. 16-8: Programmbeispiel NC-Programm, setzen des Merkers mittels PLC

SPS-Teil Die SPS fragt in einem Funktionsblock zyklisch den Zustand des Merkers ab. Sobald dieser den Wert 7 hat, wird ein interner Zustand umgeschaltet und der Ausgang für die LED aktiviert.

```
VAR_OUTPUT ; Deklarationsteil
LEDOuput : BOOL;
END_VAR

IF RunningState = 7 THEN ; Prüfen des Eingangs
LEDOuput := TRUE ; LED einschalten
```

```
InternState := 4           // internen Status umsetzen  
END_IF;
```

Tab. 16-9: Programmbeispiel SPS-Programm, Abfragen des Merkers

Handlungsanweisung Systemdaten

In diesem Beispiel liegt die gleiche Anforderung zugrunde wie im vorangegangenen. Zusätzlich zum Abarbeitungsstatus soll aber auch der aktuell aktive Vorschub sowie die aktive Spindeldrehrichtung zur SPS gemeldet werden. Aus diesem Grund bietet sich die Verwendung von Systemdaten an.

Als erstes wird eine anwenderspezifische Systemdatenstruktur namens "StartState_t" mit folgenden Elementen definiert:

- State : Byte_t
- CurFeed : Float_t
- SpindleDir : Int_t

Aufgrund der Hardwarearchitektur moderner Computer ist es sinnvoll, die Elemente, die 4 Bytes groß sind ("CurFeed" und "SpindleDir") am Beginn der Struktur zu definieren. Das führt dann zu geringeren Zugriffszeiten. Die fertige Strukturdefinition sieht in xsd-Syntax folgendermaßen aus:

Programm:

```
<xs:complexType name="StartState_t">
  <xs:sequence>

    <xs:element name="CurFeed" type="Float_t">
      <xs:annotation>
        <xs:documentation>current feed</xs:documentation>
      </xs:annotation>
    </xs:element>

    <xs:element name="SpindleDir" type="Int_t">
      <xs:annotation>
        <xs:documentation>current spindle orientation</xs:documentation>
      </xs:annotation>
    </xs:element>

    <xs:element name="State" type="Byte_t">
      <xs:annotation>
        <xs:documentation>NC execution state</xs:documentation>
      </xs:annotation>
    </xs:element>

  </xs:sequence>
</xs:complexType>
```

Nachdem die neue Struktur angelegt ist, kann eine SD-Variable "StartState" vom Typ "StartState_t" deklariert werden.

Programm:

```
<Variable Storage="volatile">
  <Name>StartState</Name>
  <Type>StartState_t</Type>
  <Comment>Control variable for NC after PLC</Comment>
</Variable>
```

NC-Teil Die Vorbereitung läuft an, sobald das Programm startet. Per CPL-Befehle werden die notwendigen Daten ermittelt und in die Systemdatenstruktur geschrieben. Um auf die Systemdatenvariable zuzugreifen, wird das CPL-Schlüsselwort SD. benutzt.

```
10 SD.StartState.CurFeed = SDR(5,1,2) ; prog. Vorschub
20 SD.StartState.SpindleDir = SD(12,1) ; Spindelrichtung
30 SD.StartState.State = 7 ; Zustand "läuft"
...
```

Tab. 16-10: Programmbeispiel NC-Programm, Zuweisung auf Systemdatum

SPS-Teil Die SPS fragt mit dem Funktionsbaustein "MT_SD_RD" in einem Funktionsblock zyklisch den Zustand des Systemdatum ab. Sobald das Element "State" den Wert 7 hat, wird ein interner Zustand umgeschaltet und der Aus-

Synchronisationsfunktionen

gang für die LED aktiviert. In diesem Moment sind dann auch die anderen beiden Elemente mit gültigen Werten befüllt.

Vorab soll eine Instanz des Funktionsbausteins "MT_SD_RD" mit dem Namen "SD_READ" sowie ein Merker namens "ReadState" (Typ Byte, Initialisierung 1) angelegt worden sein.

Programm:

```

VAR_OUTPUT
  LEDOutput : BOOL;                                // Declaration
END_VAR

IF ReadState = 1 THEN
  SD_READ(Start:=TRUE, XPath="/StartState/State",
    DestAdr:=ADR(InternState),
    Size:=SIZEOF(InternState));                    // Read SD
  ReadState := 2;                                  // Change status
END_IF;

IF (ReadState = 2) AND (NOT SD_READ.Active) THEN
  IF SD_READ.Error THEN
    ...                                              // Error handling
  ELSEIF InternState = 7 THEN                       // in State stand 7
    LEDOutput := TRUE;                             // switch on LED
    ReadState := 3;                                 // Query next value
  ELSE
    ReadState := 1;                                 // Read SD again
  END_IF;
  SD_READ(Start:=FALSE);                           // Reset FB
END_IF;

IF ReadState = 3 THEN
  SD_READ(Start:=TRUE, XPath="/StartState/CurFeed",
    DestAdr:=ADR(CurrentFeed),
    Size:=SIZEOF(CurrentFeed));                    // Read SD
  ReadState := 4;                                  // Change status
END_IF;

IF (ReadState = 4) AND (NOT SD_READ.Active) THEN
  ...

```



Die Syntax im Programmbeispiel entspricht der IndraLogic 1G. Bei IndraLogic 2G haben Ein- und Ausgänge des Funktionsbausteins zum Teil andere Bezeichnungen.

16.2.4 Oberfläche (OPC) und NC-Vorbereitung (SD)

Beschreibung

Funktion

Die Oberfläche (IndraWorks Operation und IndraWorks Engineering) kommunizieren mit der NC über OPC. Weiterhin können externe Anwendungen die OPC-Schnittstelle benutzen um den Status der NC abzufragen und Kommandos an die NC zu geben.

Damit eine Oberfläche den aktuellen Zustand der NC anzeigen kann, muss dieser von der NC übertragen werden. Neben einer Reihe von Statusinformationen die standardmäßig im OPC-Interface vorhanden sind (siehe Dokumentation "MTX OPC-Kommunikation"), können spezifische Informationen aus dem Teileprogramm übermittelt werden.

Das Hilfsmittel für eine Synchronisation zwischen Oberfläche und NC-Vorbereitung sind die Systemdaten (siehe Dokumentation "MTX Maschinenparameter", Kap. "Systemdaten"). Systemdaten sind sowohl in der NC als auch in der Oberfläche (per OPC) verfügbar.

Die NC-Vorbereitung kann Systemdaten einfach durch eine CPL-Zuweisung beschreiben. Das Schlüsselwort "SD." muss der Variable vorangestellt sein.

Eine Oberfläche kann das OPC-Item "NC.SystemDataEvent" abonnieren und "NC.SystemData_U" entsprechend auswerten (siehe Dokumentation "MTX OPC-Kommunikation", Kap. "Items"). Dadurch wird die Anwendung geweckt (z.B. ein Delegate gerufen), sie kann den Wert des Systemdatums ermitteln und ihren Status ändern.

Relevante CPL-Funktionen

Das Verändern von Systemdaten per CPL erfolgt als Wertzuweisung. Vor das Systemdatum muss das Schlüsselwort "SD." programmiert werden.

Relevante OPC-Items

NC.SystemData_U: Das OPC-Item `NC.SystemData_U, <XPath>` (siehe "MTX OPC-Kommunikation", Kapitel "Systemdaten, Item NC.SystemData_U") liefert den Inhalt der angegebenen Systemdatenstruktur als XML-Stream.

NC.SystemDataEvent: Mit dem OPC-Item `NC.SystemDataEvent, <XPath>` (siehe "MTX OPC-Kommunikation", Kapitel "Systemdaten, Item NC.SystemDataEvent") kann ein Änderungsereignis einer Systemdatenstruktur abonniert werden. Eine (CallBack-)Funktion wird immer dann von der OPC-Schicht gerufen, wenn sich das angegebene Systemdatum ändert.

Handlungsanweisung

Im folgenden Beispiel arbeitet die NC im Teileprogramm eine Schleife ab. Es wird wiederholt ein Unterprogramm aufgerufen, das die eigentliche Bearbeitung enthält.

Die NC-Vorbereitung soll nun der Oberfläche mitteilen, in welchem Durchgang sich die Bearbeitung gerade befindet. Dazu soll das Systemdatum "PPState" (Typ `Byte_t`) benutzt werden.

NC-Teil In einem ersten Schritt muss das Systemdatum angelegt werden (siehe Dokumentation "MTX Maschinenparameter", Kap. "Systemdaten"). Da ein Standardtyp verwendet wird, muss keine Schemadatei (*.xsd) angelegt werden. Es wird eine Datei "SDDefMTB.XML" im usrfep erzeugt, in der das Systemdatum deklariert wird.

```
...
<Variable Storage="volatile">
  <Name>PPState</Name>
  <Type>Byte_t</Type>
  <Comment>Synchronisation</Comment>
</Variable>
...
```

Tab. 16-11: Deklaration des Systemdatums PPState

Danach muss die MTX neu gestartet werden, damit das Systemdatum angelegt wird.

Im zweiten Schritt wird das Teileprogramm so modifiziert, dass es die Nummer des Durchlaufs in das Systemdatum schreibt.

```
...
100 FOR NR%=1 TO COUNT%           ; Schleife
110 WAIT                           ; Synchronisation mit Ausführung
120 SD.PPState = NR%               ; Durchlauf -> Systemdatum
N130 P SubProgram(NR%)             ; Unterprogrammaufruf
140 NEXT I%
...
```

Tab. 16-12: Teileprogramm mit Schleifendurchlauf nach Systemdatum

Sobald die NC-Vorbereitung Satz 120 erreicht, wird das Systemdatum geändert.



Das `WAIT` in Satz 110 ist notwendig, weil sonst möglicherweise bereits viele Schleifendurchgänge von der NC-Vorbereitung abgearbeitet wurden, noch bevor die NC-Ausführung den ersten Satz im Unterprogramm erreicht. Siehe dazu auch [Kap. 16.2.1 "NC-Vorbereitung und NC-Ausführung \(WAIT\)" auf Seite 655](#). Alternativ könnte für die Zuweisung in Satz 120 auch die NC-Funktion `SSD` benutzt werden. In diesem Fall könnte das `WAIT` entfallen.

Oberflächen-Teil

Die Schritte der Oberfläche sind an dieser Stelle nur beschrieben (und nicht als Programmcode gelistet), weil die spezifische Syntax von der eingesetzten Programmiersprache bzw. -umgebung abhängen.

Als erstes muss die Oberfläche das OPC-Item "NC.SystemDataEvent" abonnieren. Dazu wird eine Gruppe angelegt, die ein Änderungsevent (hier "NCStateChange") erhält. In dieser Gruppe ist das Item "NC.SystemDataEvent" enthalten.

```
...
// Initialisierung
Gruppe "NCState" anlegen
Änderungsevent "NCStateChange" für Gruppe "NCState"
Item "NC.SystemDataEvent" in Gruppe "NCState" anlegen
...
```

Tab. 16-13: Vorbereitung der SD-Abfrage mittels OPC

In der Eventfunktion wird dann das OPC-Item "NC.SystemData_U" abgefragt. Dort wird dann der Wert gelesen, der von der NC-Vorbereitung in Satz 120 geschrieben wurde.

```
...
// Änderungseventfunktion "NCStateChange"
Item "NC.SystemData_U,/PPState" abfragen
...
```

Tab. 16-14: SD-Abfrage mittels OPC



Es wird dringend davon abgeraten das OPC-Item "NC.SystemData_U,/PPState" zyklisch abzufragen. Das Item ist nicht dafür optimiert und würde somit das System sehr stark belasten.

16.3 Synchronisation mit der NC-Ausführung

16.3.1 NC-Ausführung und NC-Vorbereitung

Beschreibung

Funktion

Kommen im Teileprogramm eine kritische Passage mit (zeitlich) sehr kurzen Sätzen vor, soll ggf. sichergestellt werden, dass der Puffer der vorbereiteten Sätze weitestgehend gefüllt ist, bevor diese Passage abgearbeitet wird. Auf diese Weise kommt es nicht zu einem Abriss (Abarbeitung kann keinen neuen Satz holen, weil noch keiner komplett vorbereitet ist), wenn die Satzvorbereitung länger als die Satzabarbeitung dauert.

Genauso ungünstig kann es sein, dass die MTX die Ausführung eines Programms beginnt, sobald der erste Satz komplett vorbereitet ist. Der Vorbereitungspuffer ist in diesem Fall leer.

Schließlich bietet die MTX die Möglichkeit per gepufferter NC-Satzvorgabe ein Programm in die Steuerung zu übertragen und ausführen zu lassen. Die Kommunikation wird jedoch mit niedriger Priorität behandelt und kann fehleranfällig sein. Aus diesem Grund kann es vorkommen, dass die Übertragung einzelner Sätze deutlich länger dauert.

In allen drei Anwendungsszenarien kann vor der kritischen Passage gezielt die Ausführung angehalten werden, bis die Satzvorbereitung genügend Sätze komplett vorbereitet hat. Dazu wird die NC-Funktion PREP verwendet.

Restriktionen

- Der Satzvorbereitungspuffer muss mindesten so groß sein, wie die programmierte Anzahl an vorzubereitenden Sätzen. Er kann in den Maschinenparametern kanalweise konfiguriert werden.
- Die Funktion wartet nur in Handeingabe, gepufferter NC-Satzvorgabe und im Automatik-Modus. In allen anderen Betriebsarten (z.B. Einzelsatz) wird nicht gewartet.
- Wird während des Wartens ein WAIT oder das Programmende gefunden, werden die Sätze bis dorthin komplett vorbereitet und danach läuft die Ausführung weiter.
- Während einer Grundstellung wird grundsätzlich nicht gewartet.
- Der Befehl PREP darf nicht in einem asynchronen Unterprogramm eingesetzt werden. Es wird ein Laufzeitfehler erzeugt, falls er in einem asynchronen Unterprogramm programmiert wurde.
- Wird mit gepufferter NC-Satzvorgabe gearbeitet, darf der erste Satz einer gepufferten NC-Satznachricht entweder kein Autostart-Flag oder kein PREP enthalten.
- Am Ende eines Satzes, in dem PREP programmiert wurde, wird zwingend die Geschwindigkeit auf Null zurückgefahren.

Relevante NC-Funktionen

Die NC-Funktion PREP kann benutzt werden, um die Satzausführung so lange anzuhalten, bis die programmierte Anzahl von Sätzen vorbereitet wurde.

Relevante Maschinenparameter (MP)

Der Maschinenparameter NCO/LookAh/Ch[k]/NofBlkPrep "Gesamtzahl Sätze für Satzvorb.u.Interpolation" (MP 7060 00110) konfiguriert die Speichergröße des Satzvorbereitungspuffers. In diesem Speicher werden die fertig vorbereiteten Sätze abgelegt. Aus diesem Grund darf der programmierte Wert der PREP-Funktion niemals größer als der Maschinenparameter sein.

Handlungsanweisung

In diesem Beispiel soll in einem Teileprogramm eine kritische Passage mit 200 komplexen Splinesätzen stehen. Es ist für den Bearbeitungsprozess notwendig, dass diese Passage mit konstanter (programmierter) Bahngeschwindigkeit abgearbeitet wird. Ungünstigerweise muss kurz vor der Passage ein WAIT programmiert werden, weil dort ein Maschinenzustand abgefragt wird. Es ist also sehr wahrscheinlich, dass die Splinesätze noch nicht komplett vorbereitet wurde, wenn deren Ausführung beginnt.

Um sicherzustellen, dass alle 200 Sätze vorbereitet sind, bevor die Ausführung der Passage beginnt, wird ein PREP programmiert.

100 WAIT ; Synchronisation für Maschinenstatus
 ... Maschinenstatus abfragen ...

N150 PREP(200) ; Warten, bis Splinesätze vorbereitet sind
 N160 G6 F1234
 ... 200 Splinesätze ... ; kritische Passage

16.3.2 NC-Ausführung zwischen Kanälen (WEV/WREV/SEV, WPV/SPV)

Beschreibung

Funktion

Einer der häufigsten Fälle für Synchronisationen ist die Synchronisation zwischen Kanälen. Im Allgemeinen stellt jeder Kanal eine geschlossene Arbeitseinheit dar. Wenn sich die Arbeitsbereiche dieser Einheiten überdecken oder wenn die Arbeitsabläufe aufeinander abgestimmt werden müssen (z.B. Achsen müssen Kanal wechseln), wird eine Synchronisation notwendig.

Die MTX bietet zwei verschiedene Hilfsmittel für die Synchronisation zwischen Kanälen an:

- Bit-Ereignisse (WEV/WREV/SEV/REV) und
- permanente Variablen (WPV(E)/SPV(E) oder CPL)

Bit-Ereignisse

In der MTX sind 96 global systemweit gültige Bit-Ereignisse angelegt. Diese Ereignisse sind Pegel ("0" oder "1"), die mit den NC-Funktionen **REV** (ResetEvent), **SEV** (SetEvent), **WEV** (WaitForEvent) und **WREV** (WaitForAndResetEvent) modifiziert oder genutzt werden können.

Mit der Funktion **WEV** wird zur Laufzeit gewartet (bzw. die Satzweiserschaltung verhindert), bis ein bestimmtes Bit-Ereignis eintritt. Dazu wird geprüft, ob der entsprechende Pegel "1" ist. Wenn der Pegel bereits beim Erreichen von **WEV** "1" war, wird nicht gewartet. (Es wird jedoch in jedem Fall am Satzende auf Geschwindigkeit 0 gebremst.)

Genauso funktioniert die Funktion **WREV**. Hier werden aber nach dem Warten alle Bit-Events, die in der Wartebedingung programmiert wurde, gelöscht.

Mit der Funktion **SEV** wird zur Laufzeit ein Bit-Ereignis ausgelöst (gesetzt). Damit wird der Pegel des programmierten Ereignisses auf "1" gesetzt (egal ob er vorher "0" oder "1" war).

Mit der Funktion **REV** wird zur Laufzeit ein Bit-Ereignis gelöscht (auf Pegel "0" gesetzt).

Sind mehrere Funktionen in einem Satz programmiert sind, wirken sie in der Reihenfolge:

1. **REV**
2. **SEV**
3. **WEV/WREV**

Mit dem Satz **N10 REV(10) WEV(10)** kann somit auf eine Flanke des Ereignis 10 gewartet werden.

permanente Variablen

Unter der Bezeichnung @1 bis @100 lassen sich permanente Variablen vom Typ **INTEGER** ansprechen. Weiterhin können in der MTX weitere permanente Variablen definiert werden. Sie sind nicht automatisch Bestandteil der Systemsoftware, sondern müssen in den Dateien "wmhperm.dat" (für WMH-spezifische Daten) und "anwperm.dat" (für Endanwender-spezifische Daten) manuell per Eintrag deklariert werden (siehe "Rexroth IndraMotion MTX 14VRS Programmierhandbuch", Kap. Variablenprogrammierung).

Permanente Variablen sind systemweit gültige Variablen mit einfachen Typen (u.a. **INTEGER**, **REAL**, **DOUBLE**, **STRING**). Sie können zum Vorbereitungs-

zeitpunkt direkt per CPL-Zuweisung beschrieben werden oder zum Ausführungszeitpunkt mit `SPV` und `SPVE`.

```

...
100 @MYVAR=10                                ; Vorbereitungszeitpunkt
N110 SPV[@MYVAR=20]                          ; Ausführungszeitpunkt
...

```

Tab. 16-15: Zuweisen an permanente Variablen

Die Ausführung der NC-Sätze kann angehalten werden, indem die NC-Funktion `WPV` bzw. `WPVE` programmiert wird. In dieser Funktion kann eine Bedingung für eine permanente Variable angegeben werden. Die NC-Ausführung unterbricht so lange, bis die Bedingung erfüllt ist.

Restriktionen

Von der MTX wird nicht überwacht, ob es zu ungültigen Zuständen kommt. Das bedeutet, wenn zwei Kanäle wechselseitig auf ein Synchronisationssignal des anderen Kanals warten, steht die NC (deadlock).

Relevante NC-Funktionen

`WEV/WREV/SEV/REV` für Bit-Ereignisse

`WPV(E)/SPV(E)` für permanente Variablen

Relevante CPL-Funktionen

CPL-Zuweisungen für permanente Variablen

Handlungsanweisung Bit-Ereignisse

Im folgenden Beispiel sollen die Kanäle 1 und 2 synchronisiert werden. Die X-Achse von Kanal 1 passiert den Arbeitsraum von Kanal 2 wenn sie zum Werkzeugwechsel fährt. Deshalb soll der Kanal 1 warten, bis Kanal 2 signalisiert, dass die Bahn frei ist. Das Bit-Ereignis 42 soll für "Bahn frei" stehen.

Danach fährt Kanal 1 zum Werkzeugwechsel. In diesem Zustand darf Kanal 2 nicht wieder zurück fahren, weil es sonst zu einer Kollision kommen würde. Aus diesem Grund wird ein zweites Bit-Ereignis 44 für den Zustand "K2 zurück" benutzt.

```

...
N100 WEV(42)                                ; Warte auf "Bahn frei"
N110 REV(44)                                ; Lösche "K2 zurück"
N120 M6 T4                                  ; Werkzeugwechsel
                                           ; kritischer Bereich wurde verlassen
N130 SEV(44)                                ; Signalisiere "K2 zurück"
...

```

Tab. 16-16: Programm für Kanal 1

```

...
N100 G0 X -50 Y 200                          ; Fahre aus Kollisionszone heraus
N110 SEV(42)                                  ; Signalisiere "Bahn frei"
...                                           ; Bearbeitung außerhalb Kollisionszone

```

N180 WEV(44)	; darf K2 zurückfahren?
...	; Bearbeitung innerhalb Kollisionszone

Tab. 16-17: Programm für Kanal 2



Das angegebene Beispiel kann nicht ohne weiteres in einer Schleife eingesetzt werden. Bei einer ungünstigen Abfolge könnte Kanal 1 dann in N100 auf 42 warten während Kanal 2 in N180 auf 44 wartet.

Handlungsanweisung permanente Variablen

In diesem Fall soll die gleiche Situation wie in [Kap. "Handlungsanweisung Bit-Ereignisse" auf Seite 667](#) vorliegen. Nun soll eine permanente Variable @CH_SYNC zur Synchronisation genutzt werden. Der Wert 6 soll für "Bahn frei" stehen, der Wert 7 für "K2 zurück".

Im ersten Schritt muss die Variable angelegt werden. Dazu wird in die Datei "wmhperm.dat" (im USRFEP oder root) die folgende Zeile ergänzt:

```
DEF INT @CH_SYNC;
```

Die Programme für beide Kanälen sehen dann wie folgt aus:

...	
N100 WPV[@CH_SYNC=6]	; Warte auf "Bahn frei"
N120 M6 T4	; Werkzeugwechsel
	; kritischer Bereich wurde verlassen
N130 SPV[@CH_SYNC=7]	; Signalisiere "K2 zurück"
...	

Tab. 16-18: Programm für Kanal 1

...	
N100 G0 X -50 Y 200	; Fahre aus Kollisionszone heraus
N110 SPV[@CH_SYNC=6]	; Signalisiere "Bahn frei"
...	; Bearbeitung außerhalb Kollisionszone
N180 WPV[@CH_SYNC=7]	; darf K2 zurückfahren?
...	; Bearbeitung innerhalb Kollisionszone

Tab. 16-19: Programm für Kanal 2

16.3.3 NC-Ausführung und Interface/SPS (WAITA/WAITO)

Beschreibung

Funktion

Viele Prozesse und Abläufe werden in der Werkzeugmaschine von der SPS gesteuert. Deshalb ist es notwendig, NC und SPS zu synchronisieren. Die Schnittstelle zwischen beiden Instanzen ist das Interface. Eine detaillierte Beschreibung des Interface ist in der Dokumentation "MTX SPS-Interface" zu finden.

Wenn die Ausführung des Teileprogramms (und damit die Bewegung der Maschine) von der SPS abhängt, muss eine Synchronisation zwischen NC-Ausführung und SPS erfolgen.

Mit den NC-Funktionen `WAITA` und `WAITO` wird zur Laufzeit (aktiver Zeitpunkt, siehe auch [Kap. "Funktion" auf Seite 656](#)) gewartet, bis ein oder mehrere Interface-Signale von maximal 16 Signalen einen jeweils vorgegebenen Wert angenommen haben.

Je nach Verknüpfung bei mehreren Interface-Signale kann programmiert werden:

- `WAITA`: "Und-Verknüpfung" der einzelnen Signale
Warten, bis **alle** Interface-Signale den vorgegebenen Wert angenommen haben.
- `WAITO`: "Oder-Verknüpfung" der einzelnen Signale
Warten, bis **mindestens ein** Interface-Signal den vorgegebenen Wert angenommen hat.



Werden `WAITA` und `WAITO` in einem NC-Satz programmiert, wird die Satzausführung solange angehalten, bis beide Bedingungen erfüllt sind. Die `WAITO`-Bedingung wird dabei zuerst ausgewertet.



Die SPS kann auch mit dem Kanalinterface-Bit "qCh_FeedHold" die Ausführung des Teileprogramms anhalten. In diesem Fall ist jedoch nicht bekannt, an welcher Stelle im Programm das geschieht.

Relevante NC-Funktionen

`WAITA`: "Und-Verknüpfung" der einzelnen Signale

`WAITO`: "Oder-Verknüpfung" der einzelnen Signale

Relevante IF-Signale

Mittels der Funktion `BITIF` kann jedes Bit im Interface abgefragt werden. Aus diesem Grund können `WAITA` und `WAITO` auf beliebige Bits im Interface warten.

Handlungsanweisung

SPS-Teil

An einer Beispielmachine sollen zwei SPS-gesteuerte Spindeln existieren. Damit das Teileprogramm abgearbeitet werden kann, müssen beide Spindeln mit ihrer Solldrehzahl laufen.

Weil die Spindeln von der SPS gesteuert werden, muss diese erkennen, wann die Solldrehzahl erreicht ist. Sie signalisiert das (für jede Spindel separat) im Kanalinterface. Die Solldrehzahlen sollen in den globalen Variablen "SP1_SET" und "SP2_SET" abliegen.

Bei einem NC-Start wird automatisch das Programm gestartet und die Spindeln laufen an.

```

VAR_OUTPUT                                // Deklarationsteil
SP1_ON: BOOL;                             // Start-Bit von Spindel 1
SP2_ON: BOOL;                             // Start-Bit von Spindel 2
END_VAR

IF qCh_NCStart THEN                       // wird NC-Start gedrückt?
SP1_ON := TRUE                           // Starten von Spindel 1

```

```

SP2_ON := TRUE           // Starten von Spindel 2
qCh_Custom4 := 0         // Drehzahl nicht erreicht
qCh_Custom5 := 0         // Drehzahl nicht erreicht
END_IF;

```

Tab. 16-20: *Programmbeispiel SPS-Programm, Starten der Spindeln bei NC-Start*

Danach wird in einem Funktionsbaustein zyklisch geprüft, ob die Solldrehzahl erreicht ist und ggf. die Bits im Interface gesetzt.

```

VAR_INPUT                // Deklarationsteil
SP1_FEED: REAL            // aktuelle Drehzahl Sp 1
SP2_FEED: REAL            // aktuelle Drehzahl Sp 2
END_VAR

```

```

IF ABS(SP1_FEED-SP1_SET) < EPSILON THEN // Drehzahl prüfen
qCh_Custom4 := 1                 // Spindel 1 läuft
END_IF;

```

```

IF ABS(SP2_FEED-SP2_SET) < EPSILON THEN // Drehzahl prüfen
qCh_Custom5 := 1                 // Spindel 2 läuft
END_IF;

```

Tab. 16-21: *Programmbeispiel SPS-Programm, Abfragen der Spindeldrehzahl*

NC-Teil

Sobald die beiden Bits, die den Status der Spindeln anzeigen, aktiv sind, soll das Teileprogramm anlaufen. Dazu wird am Beginn des Programms die NC-Funktion WAITA programmiert, die beide Bits prüft. Nur wenn beide aktiv sind, wird die Ausführung des Teileprogramms fortgesetzt.

... ggf. CPL-Berechnungen ...

```

N10 WAITA[BITIF("qCh_Custom4"), BITIF("qCh_Custom5")] // Warte auf Bits

```

... Teileprogramm ...

Tab. 16-22: *Programmbeispiel NC-Programm, Warten auf Interface-Bits*

16.3.4 NC-Ausführung und Positionen (*STOPA/*STOPO)

Beschreibung

Funktion

Wenn sich die Arbeitsbereiche von Achsen überdecken oder wenn die Arbeitsabläufe aufeinander abgestimmt werden müssen, wird eine Synchronisation notwendig. In diesem Kapitel werden Synchronisationen der NC-Ausführung auf Basis von Achspositionen behandelt. Das bedeutet, mit den entsprechenden Funktionen kann die Ausführung angehalten werden, bis programmierte Positionen erreicht bzw. verlassen wurden.



Im weitesten Sinne handelt es sich dabei um eine Kanalsynchronisation wie sie in [Kap. 16.3.2 "NC-Ausführung zwischen Kanälen \(WEV/WREV/SEV, WPV/SPV\)" auf Seite 666](#) beschrieben ist. Im Gegensatz zu den dort aufgeführten Anwendungsfällen muss bei der Synchronisation mit Positionen die Stelle im Teileprogramm nicht bekannt sein.

Die grundsätzliche Funktionsweise dieser Synchronisationsbefehle ist folgendermaßen:

1. *STOPA/*STOPO wird in Kanal A programmiert.
2. Die dort angegebenen Bedingungen betreffen (synchrone) Kanalachsen von Kanal A.
3. In *STOPA/*STOPO wird ein Kanal B angegeben, der angehalten wird.
4. Kanal B darf dann weiterlaufen, wenn die Bedingungen von *STOPA/*STOPO erfüllt sind.

Im folgenden wird der Kanal A (zu dem die Achsen gehören) als "Steuerkanal" und Kanal B (der angehalten wird) als "zu steuernder Kanal" bezeichnet.

Die hier beschriebenen Funktionen unterscheiden sich in der Art des überwachten Koordinatensystems. Eine genaue Erläuterung der Koordinatensysteme der MTX ist in der Dokumentation "Rexroth IndraMotion MTX Funktionsbeschreibung Grundlagen", Kap. "Koordinatensysteme im Kanal" zu finden. Dem Anwender stehen die folgenden Funktionen zur Verfügung:

- ASTOPA/ASTOPO: Achskoordinatensystem ACS
- BSTOPA/BSTOPO: Basis-Werkstückkoordinatensystem BCS
- WSTOPA/WSTOPO: Werkstückkoordinatensystem WCS

Für jeden zu steuernden Kanal können eine einzelne oder mehrere Bedingungen zum Anhalten des Kanals vorgegeben werden:

- *STOPA: Solange **alle** Bedingungen erfüllt sind, wird die synchrone Bewegung des zu steuernden Kanals angehalten (logische UND-Verknüpfung).
- *STOPO: Solange **mindestens eine** Bedingung erfüllt ist, wird die synchrone Bewegung des zu steuernden Kanals angehalten (logische ODER-Verknüpfung).

Relevante NC-Funktionen

	Und-Verknüpfung	Oder-Verknüpfung
Achskoordinatensystem ACS	ASTOPA	ASTOPO
Basis-Werkstückkoordinatensystem BCS	BSTOPA	BSTOPO
Werkstückkoordinatensystem WCS	WSTOPA	WSTOPO

Tab. 16-23: Relevante NC-Funktionen

Handlungsanweisung

Im folgenden Beispiel soll eine Schleifmaschine Werkstücke auf einem Förderband bearbeiten. Das Band bewegt die Teile kontinuierlich in Y-Richtung durch die Schleifmaschine durch. Die rotierende Schleifscheibe wird von der Maschine in X-Richtung hin- und herbewegt.

Dort wo das Werkzeug am Eingriff ist, soll mit konstanter Bahngeschwindigkeit gefahren werden. Die X-Achse wurde deshalb so programmiert, dass sie in jede Richtung ein Stück über das Teil hinausfährt. In dieser Zeit (Werkzeug nicht im Eingriff) soll das Band das Werkstück weiterbewegen, damit die nächste Schleifbahn bearbeitet wird.

Synchronisationsfunktionen

Das Programm für die Bewegung des Förderbandes läuft in Kanal 1. Die Länge eines Werkstückes wird als Parameter in einem Systemdatum übergeben.

```

N10 G0 G90 Y0           ; auf Startposition
20 WAIT
30 LENGTH=SD.PartLength ; Teillänge aus Systemdatum
40 IF LENGTH = 0 THEN    ; kein neues Teil?
50 GOTO .EXIT            ; dann beenden
60 ENDIF
N70 P GetWorkpiece       ; Werkstück aufs Band
N80 Y100                 ; in die Maschine fahren
N90 G91 Y[LENGTH]        ; Schleifprozess
N100 Y100                ; aus Maschine raus fahren
N110 P RemoveWorkpiece   ; Werkstück vom Band nehmen
120 GOTO N10             ; Schleifenrücksprung

```

Tab. 16-24: *Programmbeispiel NC-Programm im Kanal 1, Bewegen des Förderbandes*

In Kanal 2 wird das Werkzeug bewegt. Dieses fährt immer entlang der X-Achse hin und her. Zur Synchronisation mit der Bewegung in Kanal 1 wird vorab die NC-Funktion `WSTOPA` programmiert. Immer dann, wenn die WCS-Position von X zwischen 200 und 800 mm liegt, befindet sich das Werkzeug am Eingriff. In dieser Zeit soll das Band stehen.

```

N10 G0 X0               ; auf Startposition
N20 WSTOPA(1,"X">200,"X"<800) ; Haltebedingung für Kanal 1
N30 G1 F800 X1000       ; fahre hin (schleifen)
N40 X0                  ; fahre zurück (schleifen)
N50 GOTO N30            ; und wieder von vorn

```

Tab. 16-25: *Programmbeispiel NC-Programm im Kanal 2, Schleifbewegung und Steuerung Kanal 1*

16.3.5 SPS und NC-Ausführung (Hilfsfkt., SSD)

Beschreibung

Funktion

Viele Prozesse und Abläufe werden in der Werkzeugmaschine von der SPS gesteuert. Deshalb ist es notwendig, NC und SPS zu synchronisieren.

In diesem Kapitel wird der entgegengesetzte Fall zu [Kap. 16.3.3 "NC-Ausführung und Interface/SPS \(WAITA/WAITO\)" auf Seite 668](#) behandelt. Hier soll die NC-Ausführung der SPS eine Nachricht senden, an welcher Stelle im Teileprogramm sie sich befindet. Die SPS kann darauf reagieren und ihren Ablauf entsprechend anpassen.

Die MTX stellt für diese Kommunikation zwei verschiedene Wege zur Verfügung:

- Hilfsfunktionen für einzelne Bit-Ereignisse
- Systemdaten für komplexere Statusinformationen

Das Applizieren und die Verwendung von Hilfsfunktionen sind ausführlich in der Dokumentation "MTX Funktionsbeschreibung - Grundlagen", Kap. "Hilfsfunktionen" beschrieben. Aus diesem Grund wird hier nicht näher darauf eingegangen. Dieses Kapitel erläutert die Nutzung von Systemdaten als Kommunikationsmittel zwischen SPS und NC-Ausführung.

Die Verwendung von Systemdaten erfolgt in diesen Schritten:

1. Systemdaten (ggf. inklusive Typ) müssen in der NC angelegt werden

2. NC-Funktionen SSD/SSDE weisen Systemdaten zur Laufzeit (aktiver Zeitpunkt) Werte zu
3. SPS kann Systemdaten (per Funktionsbaustein "MT_SD_RD", beschrieben in der Dokumentation "MTX SPS-Interface", Kapitel "Systemdaten") direkt abfragen

Systemdaten können sowohl einzelne Werte (u.a. INTEGER, REAL) als auch komplexe Strukturen enthalten.

Relevante NC-Funktionen

Die NC-Funktionen SSDE und SSD weisen Systemdaten zur Laufzeit (aktiver Zeitpunkt) Werte zu. Sie können somit zur Synchronisation mit der NC-Ausführung genutzt werden.

Handlungsanweisung Systemdaten

In diesem Beispiel arbeitet das Teileprogramm eine Schleife ab. Über das Systemdatum "CycleNr" vom Typ "Int_t" soll die NC-Ausführung der SPS mitteilen, in welchem Durchlauf sie sich gerade befindet.

Vorab muss die SD-Variable deklariert werden. Weil es sich um einen Standardtyp handelt, muss kein anwenderspezifischer Typ angelegt werden.

Programm:

```
<Variable Storage="volatile">
  <Name>CycleNr</Name>
  <Type>Int_t</Type>
  <Comment>Durchlaufnummer</Comment>
</Variable>
```

NC-Teil

Innerhalb der Schleife wird dem Systemdatum mit der NC-Funktion SSDE der Wert des aktuellen Durchlaufs beschrieben. Diese Funktion muss genutzt werden, damit diese Zuweisung erst zur Laufzeit (aktiver Zeitpunkt) geschieht. Damit unterscheidet sich dieser Fall von dem Beispiel in [Kap. "Handlungsanweisung Systemdaten" auf Seite 661](#).

```
...
100 FOR NR%=1 TO COUNT%                ; Schleife
N110 SSDE[SD.CycleNr = NR%]             ; Durchlauf -> Systemdatum
N120 P SubProgram(NR%)                  ; Unterprogrammaufruf
130 NEXT I%
...
```

Tab. 16-26: Teileprogramm mit Schleifendurchlauf nach Systemdatum

SPS-Teil

Die SPS fragt mit dem Funktionsbaustein "MT_SD_RD" in einem Funktionsblock zyklisch den Zustand des Systemdatum ab. Sie kann dann auf Basis von "CycleNr" verschiedene Reaktionen ausführen.

Vorab soll eine Instanz des Funktionsbausteins "MT_SD_RD" mit dem Namen "SD_READ" sowie ein Merker namens "ReadState" (Typ Byte, Initialisierung 1) angelegt worden sein.

Programm:

```
IF ReadState = 1 THEN
  SD_READ(Start:=TRUE, XPath="/CycleNr",
    DestAdr:=ADR(InternCycle),
    Size:=SIZEOF(InternCycle));           // Read SD
  ReadState := 2;                         // Change status
END_IF;

IF (ReadState = 2) AND (NOT SD_READ.Active) THEN
  IF SD_READ.ERROR THEN
    ...                                   // Error handling
  ELSE
    ...                                   // Work with InternCycle
  END_IF;
END_IF;
```

```

END_IF;
SD_READ(Start:=FALSE);           // Reset function block
ReadState := 1;                  // Read SD again

```



Die Syntax im Programmbeispiel entspricht der IndraLogic 1G. Bei IndraLogic 2G haben Ein- und Ausgänge des Funktionsbausteins zum Teil andere Bezeichnungen.

16.3.6 Oberfläche (OPC) und NC-Ausführung (SSD/SSDE/SSDQ)

Beschreibung

Funktion

Die Oberfläche (IndraWorks Operation und IndraWorks Engineering) kommunizieren mit der NC über OPC. Weiterhin können externe Anwendungen die OPC-Schnittstelle benutzen um den Status der NC abzufragen und Kommandos an die NC zu geben.

Damit eine Oberfläche den aktuellen Zustand der NC anzeigen kann, muss dieser von der NC übertragen werden. Neben einer Reihe von Statusinformationen die standardmäßig im OPC-Interface vorhanden sind (siehe Dokumentation "MTX OPC-Kommunikation"), können spezifische Informationen aus dem Teileprogramm übermittelt werden.

Das Hilfsmittel für eine Synchronisation zwischen Oberfläche und NC-Ausführung sind die Systemdaten (vergleiche Dokumentation "MTX Maschinenparameter", Kap. "Systemdaten"). Systemdaten sind sowohl in der NC als auch in der Oberfläche (per OPC) verfügbar.

SSD und SSDE

Im Gegensatz zu [Kap. 16.2.4 "Oberfläche \(OPC\) und NC-Vorbereitung \(SD\)" auf Seite 662](#) muss die NC-Ausführung das Systemdatum zur Laufzeit (aktiver Zeitpunkt) ändern. Dazu werden die NC-Funktionen `SSD` und `SSDE` verwendet.

Beide Funktionen schreiben einen CPL-Ausdruck zur Laufzeit in ein Systemdatum. Dabei akzeptiert `SSDE` beliebige Ausdrücke, die zur Vorbereitungszeit berechnet und zum aktiven Zeitpunkt zugewiesen werden. Die Funktion `SSD` kann nur einfache Ausdrücke zuweisen. Diese werden dann jedoch zum aktiven Zeitpunkt berechnet (siehe auch Dokumentation "MTX Programmierhandbuch", Kapitel "Schreiben eines Systemdatum zum Ausführungszeitpunkt").



Es wird empfohlen, möglichst `SSDE` zu nutzen. Weil die Berechnung bereits in der Vorbereitung geschieht, ist die Laufzeit des Befehls kürzer.

SSDQ

Die komplexere NC-Funktion `SSDQ` ("Schreiben in eine Systemdatenqueue") stellt mehr Komfort für die Nachrichtenübermittlung zur Verfügung. Wird `SSDQ` programmiert, schreibt es die vom Anwender vorgegebene Nachricht (eine Ganzzahl) zum **Ausführungszeitpunkt** in eine Queue in den Systemdaten (ein "Int_t"-Systemdatenfeld). Die Queue wird dabei als Ringspeicher benutzt. Das bedeutet, wenn das letzte Element des Systemdatenfelds beschrieben wird, trägt die Funktion beim nächsten Aufruf die programmierte Nachricht in das erste Element der Queue ein. In allen anderen Fällen wird das nachfolgende Element beschrieben.

Gleichzeitig modifiziert die Funktion `SSDQ` eine Steuerungsstruktur (vom Typ "SSDQCtrl_t"), die auch in den Systemdaten liegt. Der Anwender kann somit über einen Zugriff auf die Systemdaten (siehe Dokumentation "MTX OPC-Kommunikation", Kapitel "Systemdaten") ermitteln, welche Queueelemente

beschrieben wurden bzw. welche Nachrichten neu eingetroffen sind. Die Steuerungsstruktur selbst besteht aus den zwei "Int_t"- Elementen **LastFilled** und **LastGet**. Hat die Funktion eine programmierte Nachricht in die Queue geschrieben, wird das Element **LastFilled** auf den Index des modifizierten Eintrags innerhalb der Queue gesetzt. Dieser Index ist eins-basiert.

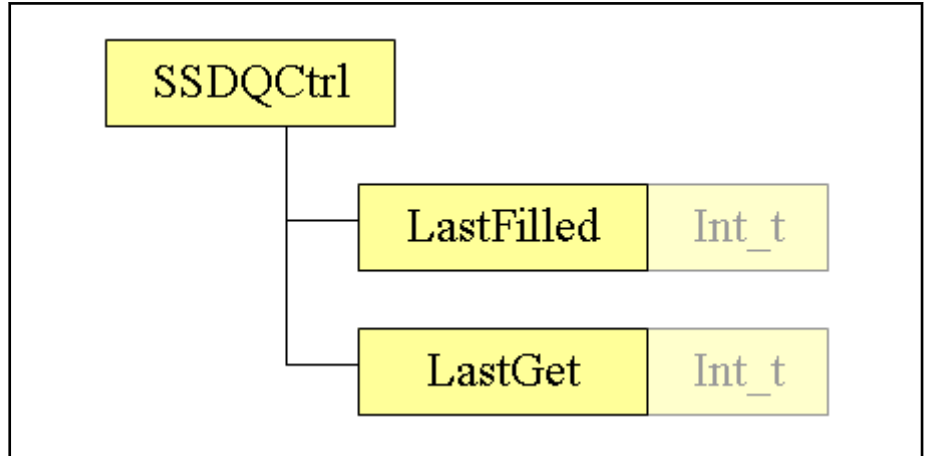


Abb. 16-2: Aufbau der Steuerungsstruktur von SSDQ

Im Falle einer Schreibaktion wird für die Steuerungsstruktur ein Änderungereignis ausgelöst. Die Queue selbst wird zwar modifiziert, ein Änderungereignis wird jedoch nicht ausgelöst. Der Anwender kann seine externe Applikation somit durch Abonnieren des Änderungereignis der Steuerungsstruktur aktivieren lassen. Dies kann beispielsweise mit dem OPC-Item NC.System-DataEvent (siehe Dokumentation "MTX OPC-Kommunikation", Kapitel "Systemdaten") realisiert werden.

Nachdem die Funktion SSDQ ausgeführt wurde, ist der Anwender dafür verantwortlich, die Nachricht aus der Queue zu lesen. Er muss dabei in der Steuerungsstruktur das Element **LastGet** auf den Index des von ihm zuletzt gelesenen Elements der Queue setzen.

Wenn der Anwender die Steuerungsstruktur liest und dabei die Werte von **LastFilled** und **LastGet** unterschiedlich sind, sind neue Nachrichten in der Queue eingetroffen.

Ist **LastFilled** größer als **LastGet**, stehen die neuen Nachrichten in den Einträgen der Queue von **LastGet+1** bis **LastFilled** (in der Reihenfolge, wie sie programmiert wurden).

Sollte **LastFilled** kleiner als **LastGet** sein, wirkt sich das Ringspeicherverhalten der Queue aus. In diesem Fall stehen die neuen Nachrichten in **LastGet+1** bis zum Enden der Queue sowie vom Beginn der Queue bis **LastFilled** (in der Reihenfolge, wie sie programmiert wurden).

Sollte es dem Anwender nicht gelingen rechtzeitig **LastGet** zu setzen, besteht die Gefahr, dass die Queue voll wird und Nachrichten verloren gehen. Aus diesem Grund prüft die Funktion SSDQ, ob noch freie Elemente in der Queue vorhanden sind (nächste Schreibposition ungleich **LastGet**). Sollte das nicht der Fall sein, wird ein Laufzeitfehler erzeugt.

Mit der Funktion **SSDQInit** kann eine Steuerungsstruktur initialisiert werden. Bei Programmierung von **SSDQInit** werden sowohl **LastGet** als auch **LastFilled** zum Ausführungszeitpunkt auf 1 gesetzt.

Restriktionen

SSD	• Es können nur einfache CPL-Ausdrücke zugewiesen werden.
SSDQ	• Die Queue muss mindestens 3 Elemente groß sein.

- Die Größe der Nachrichtenqueue ist vom freien Speicher des Targets abhängig.
- Wenn die Ausführung der programmierten Bewegung eines Satzes mit SSDQ weniger als einen IPO-Takt dauert kommt es zu einem "Ruckeln".
- Da die Kommunikation via NCS nach Windows nur niederprior ist, kann es beim Auslesen der Nachrichten zu Verzögerungen kommen.



Da eine einzelne Abfrage eines Systemdatums (beispielsweise per OPC) relativ viel Verwaltungsaufwand erfordert, wird empfohlen, ganze Strukturen bzw. ganze Felder mit einer Abfrage auf den PC zu laden.

Relevante NC-Funktionen

Die MTX stellt die folgenden drei NC-Funktionen für eine Kommunikation von der NC-Ausführung zu einer Oberfläche zur Verfügung:

- **SSD:** einfache CPL-Ausdrücke in Systemdaten zuweisen
- **SSDE:** beliebige CPL-Ausdrücke in Systemdaten zuweisen
- **SSDQ:** schreiben in eine Systemdatenqueue

SSDQ Mittels SSDQ (siehe "MTX Programmierhandbuch", Kapitel NC-Funktionen mit Hochsprachensyntax, Unterkapitel Schreiben in eine Systemdatenqueue SSDQ) wird eine programmierte Nachricht in eine Queue in den Systemdaten geschrieben. Zeitgleich wird eine Steuerungsstruktur so modifiziert, dass der Anwender daraus ermitteln kann, welche Nachrichten neu in der Queue eingetroffen sind. Sowohl Queue als auch Steuerungsstruktur können mittels eines XPath vorgegeben werden. Werden keine XPaths programmiert, dann versucht SSDQ das Systemdatum

"/SSDQueue" als Queue und

"/SSDQCtrl" als Steuerungsstruktur zu verwenden.

SSDQInit: Durch SSDQInit kann eine Steuerungsstruktur initialisiert werden. Beide Elemente der Struktur (LastFilled und LastGet) werden auf 1 gesetzt. Sinnvollerweise geschieht das vor der ersten Nutzung von SSDQ.

Relevante OPC-Items

NC.SystemData_U: Das OPC-Item `NC.SystemData_U,<XPath>` (siehe "MTX OPC-Kommunikation", Kapitel "Systemdaten, Item NC.SystemData_U") liefert den Inhalt der angegebenen Systemdatenstruktur als XML-Stream.

NC.SystemDataEvent: Mit dem OPC-Item `NC.SystemDataEvent,<XPath>` (siehe "MTX OPC-Kommunikation", Kapitel "Systemdaten, Item NC.SystemDataEvent") kann ein Änderungsereignis einer Systemdatenstruktur abonniert werden. Eine (CallBack-)Funktion wird immer dann von der OPC-Schicht gerufen, wenn sich das angegebene Systemdatum ändert.

Handlungsanweisung SSDE

Im folgenden Beispiel arbeitet die NC im Teileprogramm eine Schleife ab. Es wird wiederholt ein Unterprogramm aufgerufen, das die eigentliche Bearbeitung enthält.

Die NC-Vorbereitung soll nun der Oberfläche mitteilen, in welchem Durchgang sich die Bearbeitung gerade befindet. Dazu soll das Systemdatum "/PPState" (Typ Byte_t) benutzt werden.

NC-Teil In einem ersten Schritt muss das Systemdatum angelegt werden (siehe Dokumentation "MTX Maschinenparameter", Kap. "Systemdaten"). Da ein Stan-

Standardtyp verwendet wird, muss keine Schemadatei (*.xsd) angelegt werden. Es wird eine Datei "SDDefMTB.XML" im usrfep erzeugt, in der das Systemdatum deklariert wird.

```
...
<Variable Storage="volatile">
<Name>PPState</Name>
<Type>Byte_t</Type>
<Comment>Synchronisation</Comment>
</Variable>
...
```

Tab. 16-27: Deklaration des Systemdatums PPState

Danach muss die MTX neu gestartet werden, damit das Systemdatum angelegt wird.

Im zweiten Schritt wird das Teileprogramm so modifiziert, dass es die Nummer des Durchlaufs in das Systemdatum schreibt.

```
...
100 FOR NR%=1 TO COUNT%           ; Schleife
N110 SSDE[SD.PPState = NR%]      ; Durchlauf -> Systemdatum
N120 P SubProgram(NR%)             ; Unterprogrammaufruf
130 NEXT I%
...
```

Tab. 16-28: Teileprogramm mit Schleifendurchlauf nach Systemdatum

Sobald die NC-Ausführung Satz 110 erreicht, wird das Systemdatum geändert.

Oberflächen-Teil

Die Schritte der Oberfläche sind an dieser Stelle nur beschrieben (und nicht als Programmcode gelistet), weil die spezifische Syntax von der eingesetzten Programmiersprache bzw. -umgebung abhängen.

Als erstes muss die Oberfläche das OPC-Item "NC.SystemDataEvent" abonnieren. Dazu wird eine Gruppe angelegt, die ein Änderungsevent (hier "NCStateChange") erhält. In dieser Gruppe ist das Item "NC.SystemDataEvent" enthalten.

```
...
// Initialisierung
Gruppe "NCState" anlegen
Änderungsevent "NCStateChange" für Gruppe "NCState"
Item "NC.SystemDataEvent" in Gruppe "NCState" anlegen
...
```

Tab. 16-29: Vorbereitung der SD-Abfrage mittels OPC

In der Eventfunktion wird dann das OPC-Item "NC.SystemData_U" abgefragt. Dort wird dann der Wert gelesen, der von der NC-Vorbereitung in Satz 120 geschrieben wurde.

```
...
// Änderungseventfunktion "NCStateChange"
Item "NC.SystemData_U,/PPState" abfragen
...
```

Tab. 16-30: SD-Abfrage mittels OPC



Es wird dringend davon abgeraten das OPC-Item "NC.SystemData_U,/PPState" zyklisch abzufragen. Das Item ist nicht dafür optimiert und würde somit das System sehr stark belasten.

Handlungsanweisung SSDQ

Applizieren

MTX: In der MTX müssen die Systemdaten für die Queue und die Steuerungsstruktur angelegt werden. Die Queue ist ein eindimensionales Feld aus Int_t-Elementen, die Steuerungsstruktur hat den Aufbau wie in [Abb. 16-2 "Aufbau der Steuerungsstruktur von SSDQ" auf Seite 675](#).

Wo die Strukturen angelegt werden (als Root-Elemente oder Unterstrukturen, separat oder einzeln) ist dem Anwender freigestellt. Wie Systemdaten definiert werden, ist in der Dokumentation "MTX Maschinenparameter", Kap. "Systemdaten" beschrieben. Im folgenden Beispiel werden die Strukturen "/SSDQueue" als Queue und "/SSDQCtrl" als Steuerungsstruktur mit Kanalanzahl Dimensionen angelegt. Dies sind gleichzeitig die Vorgabewerte für SSDQ.

Beispiel:

Anlegen der Systemdatenstrukturen "/SSDQueue" und "/SSDQCtrl"

Im folgenden Auszug einer XML-Definitionsdatei werden die Systemdaten für eine Queue und eine Steuerungsstruktur angelegt.

Programm:

```
<SDDef>
  <Variable Storage="volatile" Dimension="Channel,30">
    <Name>SSDQueue</Name>
    <Type>Int_t</Type>
    <Comment>Single queue on default name</Comment>
  </Variable>

  <Variable Storage="volatile" Dimension="Channel">
    <Name>SSDQCtrl</Name>
    <Type>SSDQCtrl_t</Type>
    <Comment>Single management structure on default name</Comment>
  </Variable>
</SDDef>
```

Wenn die Definitionsdateien ergänzt wurden, muss ein Neustart des Systems erfolgen, damit die Systemdaten angelegt werden.

Windows-Anwendung: In dieser Beschreibung wird davon ausgegangen, dass die Applikation, die mit der Ausführung des Teileprogramms synchronisiert werden soll, unter Windows läuft und OPC zur Kommunikation mit der MTX nutzt.

Nachdem die Windowsapplikation die OPC-Kommunikation zur MTX etabliert hat, muss das Änderungsereignis auf die verwendete Steuerungsstruktur abonniert werden.

Dazu sollte das OPC-Item `NC.SystemDataEvent,<XPath>` verwendet werden.

Sobald das Ereignis auftritt, sollte die Applikation sowohl Steuerungsstruktur als auch Queue lesen. Wenn die Queue nicht zu groß ist (unter 4kB), sollte dies mit je einem Zugriff erfolgen.



Die Kommunikation wird auch schneller, wenn die Items zum Lesen der Systemdaten bereits vorher angelegt wurden.

Danach folgt die Auswertung der Daten. Dies ist im folgenden Pseudo-Code dargestellt.

Programm:

```

<SDDef>
  <Variable Storage="volatile" Dimension="Channel,30">
    <Name>SSDQueue</Name>
    <Type>Int_t</Type>
    <Comment>Single queue on default name</Comment>
  </Variable>

  <Variable Storage="volatile" Dimension="Channel">
    <Name>SSDQCtrl</Name>
    <Type>SSDQCtrl_t</Type>
    <Comment>Single management structure on default name</Comment>
  </Variable>
</SDDef>

```

Schließlich muss die Applikation den geänderten Wert von LastGet in die MTX zurückschreiben. Der Wert gibt somit den Index der letzten gelesenen Nachricht an.

Folgender Ablauf muss von der Windows-Applikation realisiert werden:

1. Anmelden an OPC, Verbindung mit MTX herstellen
2. das Änderungsereignis auf die verwendete Steuerungsstruktur abonnieren
3. wenn das Ereignis auftritt, Steuerungsstruktur und Queue lesen
4. Auswertung der Daten, neue Nachrichten lesen
5. geänderten Wert von LastGet in die MTX zurückschreiben

Aktivieren

Durch Programmieren von SSDQ wird eine Nachricht in die Queue geschrieben und die Windows-Applikation kann diese empfangen und auswerten.

Beispiel:

Auszug aus einem Teileprogramm

N10 SSDQInit

...

N310 X123.32 Y43.20 Z54.11 SSDQ(M27)

...

N410 X128.55 Y40.23 Z23.26 SSDQ(M28)

...

17 Service und Support

Für Ihre schnelle und optimale Unterstützung verfügen wir über ein dichtes weltweites Servicenetz. Unsere Experten stehen Ihnen mit Rat und Tat zur Seite. Sie erreichen uns täglich **rund um die Uhr – auch an Wochenenden und Feiertagen**.

Service Deutschland Unser technologieorientiertes Competence Center in Lohr deckt alle Belange rund um den Service für elektrische Antriebe und Steuerungen ab.

Sie erreichen unsere **Service-Hotline** und unseren **Service-Helpdesk** unter:

Telefon: **+49 9352 40 5060**
Fax: **+49 9352 18 4941**
E-Mail: service.svc@boschrexroth.de
Internet: <http://www.boschrexroth.com>

Auf unseren Internetseiten finden Sie ergänzende Hinweise zu Service, Reparatur (z. B. Anlieferadressen) und Training.

Service weltweit Außerhalb Deutschlands nehmen Sie bitte zuerst Kontakt mit Ihrem Ansprechpartner auf. Die Hotline-Rufnummern entnehmen Sie bitte den Vertriebsadressen im Internet.

Vorbereitung der Informationen Wir können Ihnen schnell und effizient helfen, wenn Sie folgende Informationen bereithalten:

- Eine detaillierte Beschreibung der Störung und der Umstände
- Angaben auf dem Typenschild der betreffenden Produkte, insbesondere Typenschlüssel und Seriennummern
- Ihre Kontaktdaten (Telefon-, Faxnummer und E-Mail-Adresse)

Glossar

ACS

Axis Coordinate System = Achskoordinatensystem

BCS

Basic workpiece Coordinate System = Basiskoordinatensystem

MCS

Machine Coordinate System = Maschinenkoordinatensystem

PCS

Program Coordinate System = Programmkoordinatensystem

TCP

Tool Center Point = Werkzeugmittelpunkt

TCS

Tool Coordinate System = Werkzeugkoordinatensystem

WCS

Workpiece Coordinate System = Werkstückkoordinatensystem

Index

0 ... 9

2-dimensionale Delta-Kinematik mit zusätzlicher Drehachse in der Spitze (RRR)		Mehrdeutige Achspositionen.....	293
Aktivieren.....	246	mit Schwenk-/Rundtisch.....	288
Deaktivieren.....	246	Nordpolsingularität.....	293
2/3-Achs-Transformation: Stirnseite		Referenzstellung.....	291
Aktivieren.....	388	Vektor-Ori.....	291
Applizieren.....	386	Werkzeugkorrektur.....	292
Deaktivieren.....	388	5-Achs-Transformation - Typ 4: Kardanischer Kopf	298
3-/4-Achs-Transformation: Scara		Äquatorsingularität.....	304
Aktivieren.....	214	Eigenschaften.....	300
Applizieren.....	212	Mehrdeutige Achspositionen.....	302
3-Achs-Transformation: LLR		Nordpolsingularität.....	303
Aktivieren.....	396	Referenzstellung.....	301
Applizieren.....	244, 394	Vorschub.....	304
Deaktivieren.....	397	Werkzeugkorrektur.....	302
3-Achs-Transformation: Scara		5-Achs-Transformation - Typ 4.1: Kardanischer Kopf mit linearer Orientierungsbewegung.....	308
Deaktivieren.....	214	Nordpol.....	309
3-Achs-Transformation: Zylindermantel		Schiefe Ebene.....	309
Aktivieren.....	361	5-Achs-Transformation - Typ 5: Minderbestückte RRLLL.....	310
Applizieren.....	358	Aktivieren.....	316
Deaktivieren.....	362	Applizieren.....	314
3-Achstransformation LLR.....	330	Daten der Referenzstellung ermitteln.....	314
Aktivieren.....	336	Mehrdeutigkeiten.....	312
Applizieren.....	333	Optionale Transformationsparameter.....	312
3/4 Achs-Transformation: Bipod		Referenzstellung.....	313
Aktivieren.....	206	Transformationszweig.....	312
Applizieren.....	202	Werkzeugkorrektur.....	313
Deaktivieren.....	209	Wertebereiche.....	313
4-Achs-Transformation - Typ 1: SCARA auf Linearachse.....	223	5-Achs-Transformation - Typ 5: Minderbestückte RRLLL (mit Schwenk-/Rundtisch).....	310
4-Achs-Zylindermantel-Transformation		5-Achs-Transformation - Typ 6: Kardanische RLLLR.....	320
Aktivieren.....	369	Äquatorsingularität.....	326
Applizieren.....	367	Eigenschaften.....	323
Deaktivieren.....	370	Linear-Ori.....	324
5-Achs-Transformation - Typ 1: LLLCB		Mehrdeutige Achspositionen.....	325
Achsverbindungsvektoren.....	264	Nordpolsingularität.....	325
mit linearer Bewegung der rotatorischen Achsen.....	263	Referenzstellung.....	325
mit linearer Orientierung.....	262	Vektor-Ori.....	324
mit Vektororientierung.....	261	Werkzeugkorrektur.....	325
Vorschub.....	267	5-Achs-Transformation RRLLL mit Schwenk-Rundtisch.....	288
Werkzeugkorrektur	267	5-Achs-Transformation: Kardanische LLLRR	
5-Achs-Transformation - Typ 1: LLLRR.....	260	Aktivieren.....	307
Nordpolsingularität.....	267	Applizieren.....	305
5-Achs-Transformation - Typ 2: RLLLR.....	278	Deaktivieren.....	308
Linear-Ori.....	281	5-Achs-Transformation: Kardanische RLLLR	
Mehrdeutige Achspositionen.....	284	Aktivieren.....	329
Nordpolsingularität.....	284	Applizieren.....	326
Vektor-Ori.....	281	Deaktivieren.....	330
Werkzeugkorrektur.....	283	5-Achs-Transformation: LLLAB	
5-Achs-Transformation - Typ 2: RLLLR (mit Rundtisch).....	278	Aktivieren.....	277
5-Achs-Transformation - Typ 3: RRLLL.....	288		
Linear-Ori.....	291		

Index

Applizieren.....	275	4-Achs-Transformation - Typ 2: LLLR.....	247
Deaktivieren.....	278	4-Achs-Transformation - Typ 3: RLLL.....	253
5-Achs-Transformation: LLLRR		5-Achs-Transformation - Typ 2: RLLLR	
Aktivieren.....	271	(mit Rundtisch).....	278
Applizieren.....	268	5-Achs-Transformation - Typ 3: RRLLL	
Deaktivieren.....	272	(mit Schwenk-/Rundtisch)	288
5-Achs-Transformation: RLLLR		5-Achs-Transformation - Typ 4: Kardani-	
Aktivieren.....	287	scher Kopf.....	298
Applizieren.....	285	5-Achs-Transformation - Typ 4.1: Kar-	
Deaktivieren.....	288	danischer Kopf mit linearer Orientie-	
5-Achs-Transformation: RRLLR		rungsbewegung.....	308
Aktivieren.....	297	5-Achs-Transformation - Typ 5: Minder-	
Applizieren.....	294	bestückte RRLLR (mit Schwenk-/Rund-	
Deaktivieren.....	298	tisch).....	310
5-Achs-Zylindermantel-Transformation		5-Achs-Transformation - Typ 6: Kardani-	
Aktivieren.....	378	sche RLLLR.....	320
Applizieren.....	376	5-Achs-Transformation -Typ 1: LLLRR.....	260
Deaktivieren.....	379	6-Achs-Transformation.....	336
6-Achs-Transformation.....	336	Achs-Kinematikkalibrierung.....	404
mit linearer Bewegung der Rotationsach-		Anzeige der aktiven Transformations-Da-	
sen.....	338	ten.....	194
mit Tensor-Orientierung.....	337	Ausgangsgrößen.....	190
Orientierung des TCS - x-Koordinate.....	344	Begriffe.....	188
Orientierung des TCS - y-Koordinate.....	345	Bipod-Achs-Transformation.....	200
Orientierung des TCS - z-Koordinate.....	345	Eingangsgrößen.....	190
Referenzstellung.....	342	Kennung für Achsanordnungen.....	190
Typ 3033001.....	338	Nichtorthogonale 2- und 3-Achs-Trans-	
Typ 333301.....	337	formation.....	214
Typ und Position der Rotationsachsen	339	Prinzip.....	187
Vorschub.....	345	SCARA-Achs-Transformation.....	209
Werkzeugkorrektur.....	346	Stirnseiten-Transformation.....	379
6-Achs-Transformation mit linearer Bewe-		tabellarische Übersicht.....	197
gung der Rotationsachsen.....	338	Typ-Kennung.....	188
6-Achs-Transformation mit Tensor-Orientie-		Wirkstellen.....	190
rung.....	337	Achs-Transformationen	
6-Achs-Transformation: LLLRRR		Ein- und Ausschalten.....	192
Aktivieren.....	352	Konfigurierung	191
Applizieren.....	348	Achse entfernen.....	466
Deaktivieren.....	353	Achse übernehmen.....	467
A		Achsen	
Abbruchsatz.....	43	Gantry-.....	485
Abkürzungen.....	9	Lineare Moduloachse.....	13
Abwählen einer Achs-Transformation.....	193	Achsen tippen	
Achs-Kinematikkalibrierung.....	404	In Werkstückkoordinaten.....	19
Auswahl der Achs-Transformation.....	404	Achskollisionsüberwachung.....	101
Direktes Messen.....	405	Aktivieren.....	104
Indirektes Messen.....	407	Applizieren.....	103
Kalibrierdatei.....	413	Deaktivieren.....	105
Kalibrierzyklus.....	418	Achskonfiguration im Kanal.....	465
Messverfahren.....	405	Achsübergabe.....	465
NC-Befehle.....	409	Achsnamen umbenennen.....	468
Optimierbare Konfigurationsparameter.....	404	Achspositionsüberwachung.....	106
Achs-Transformation.....	187	Aktivieren.....	107
3-Achs-Zylindermantel-Transformation.....	353	Applizieren.....	107
4-Achs-Transformation - Typ 1: SCARA		Achstransformation	
auf Linearachse.....	223	2(3)-Achs-Transformation	397
		3-Achstransformation LLR.....	330
		4-Achs-Zylindermantel-Transformation.....	362

5-Achs-Zylindermantel-Transformation.....	370
Achsübergabe.....	465
Achsanzeigen.....	469
Achse entfernen.....	466
Achse übernehmen.....	467
Achsgruppe.....	465
Achsnamen umbenennen.....	468
Achszuordnung.....	465
Aktivieren.....	473
Applizieren.....	471
Deaktivieren.....	483
Vergabe einer arbeitstechnischen Bedeutung.....	468
Wechseln von Achsen zwischen Achsgruppen.....	465
ActionBlock.....	54
Active.....	52
ActTool.K1.....	55
ActTool.K2.....	55
ADJ.....	56
Akima-Verfahren.....	128
Aktivieren der Achs-Transformation an AT1.....	192
Aktivieren der Achs-Transformation an AT2.....	192
Aktivieren der Achs-Transformation per Systemdatum.....	193
Allgemeines zum Satzvorlauf	
Abgrenzung zum Satzsuchlauf.....	42
Begriffsdefinition.....	40
Anregungen.....	9
ANSI Z535.6-2006.....	7
Antriebe (Achsen, Spindel).....	13
Antriebe (Achsen, Spindeln)	
Hirthachse.....	15
Lineare Moduloachse.....	13
Anwendungen zum Satzvorlauf	
Handeingabe.....	62
Init-Abgleichprogramm.....	63
Wiedereinstiegsmarken RCB-Marken.....	63
Anzeige der aktiven Transformations-Daten.....	194
SysAxTrafo_t.....	195
SysAxTrafo1.....	194
SysAxTrafo2.....	194
Approximierende B-Splines mit Toleranzangabe (Kompressor).....	146
Äquidistante Parametrisierung.....	129
Aufzeichnungsdatei	
Datei öffnen.....	634
Datei schließen.....	634
Dateiaufbau.....	635
Dateiname und Pfad.....	634
Digitalisieren.....	634
Struktur.....	634
Automatische Ecken- und Geradenerkennung.....	155

B

B-Splines

NURBS	122
Bahnbewegung.....	119
Approximierende B-Splines mit Toleranzangabe (Kompressor).....	146
Automatische Ecken- und Geradenerkennung.....	155
Automatische Linear-Zirkular-Spline-Wandlung.....	157
Kubische Hermite-Splines.....	133
Spline.....	119
Spline mit Kontrollpunktprogrammierung (B-Spline).....	140
Spline mit Stützpunktprogrammierung und krümmungsstetigen Übergängen.....	135
Spline programmieren.....	148
Bahnlängenparameterinterpolation.....	90
BAP.....	147
Bedienfunktionen.....	19
Handrad in Werkstückkoordinaten.....	23
Tippen in Werkstückkoordinaten.....	19
Beispiele	
Achskonfiguration und Koordinatenkonfiguration.....	70
Einzelachsenabgleich mit G77.....	75
Hilfsfunktionen.....	75
NC-Funktionen.....	75
Spindelabgleich.....	70
Spindelkopplung.....	73
Werkzeuge.....	74
Beschwerde.....	9
Bessel-Verfahren.....	128
Bestimmungsgemäßer Gebrauch	
Anwendungsbereiche.....	12
Einleitung.....	11
Einsatzfälle.....	12
Bipod-Achs-Transformation	
Programmierung der Spindel-C-Achsnachführung.....	207
Spindel Positionieren (M19).....	207
Bipod-Achs-Transformationen.....	200
3-Achs-Bipod.....	200
4-Achs-Bipod.....	200
Daten der Referenzstellung ermitteln.....	202

C

CalibDirect.cnc.....	419
CalibIndirect.cnc.....	422
Chordale Parametrisierung.....	130
CLD.....	156

D

Darstellung von Informationen	
Bezeichnungen und Abkürzungen.....	9
De-Boor-Randbedingung.....	136
Deaktivieren.....	94
Definitionsintervall.....	124
Digitalisieren.....	620

Index

Aktivieren.....	632	Istwert-Versatz.....	487
Applizieren.....	631	Maßbezug.....	486
Deaktivieren.....	634	Messsystem.....	486
Digitalisieren initialisieren.....	626	Momentenüberwachung.....	497
Digitalisieren steuern.....	629	Schräglagen.....	487
Höhenregelung.....	637	Schräglagenausgleich.....	489
Distanz vor Endpunkt.....	107	Sercos Einstellung.....	497
Applizieren.....	108	Sollwert-Kopplung.....	486
Deaktivieren.....	109	Stromüberwachung.....	497
Dokumentation		Überwachung des Referenzschalters.....	496
Änderungsverlauf.....	1	Überwachungsfunktionen	496
Drehmatrix	168	Gefahrenhinweise.....	7
Dynamik und Geschwindigkeitsführung.....	85	Genauigkeit.....	115
E		Genauigkeitsprogrammierung.....	115
EBC.....	129	Genauigkeitsprogrammierung.....	115
Eckenrunden		Geschwindigkeitsführung.....	85
mit Splines.....	445	Geschwindigkeitsvorgabe für asynchrone li-	
zirkular.....	440	neare Endlosachsen.....	13
Eckenrunden mit Splines.....	445	Gewindebohren.....	425
lokale (nichtmodale) NC-Funktion.....	447	Aktivieren.....	432
modale NC-Funktion.....	446	Applizieren.....	431
Wechselwirkung zwischen modaler und		Deaktivieren.....	433
lokaler NC-Funktion.....	447	Rückzug aus Gewindebohrung (ohne	
Eilgang, reduzierter.....	94	Ausgleichsfutter).....	426
Aktivieren.....	96	Gewindeschneiden.....	433
Applizieren.....	95	Schneller Rückzug.....	435
Deaktivieren.....	96	H	
Einrichtbetrieb		Handrad	
Tippen in Werkstückkoordinaten.....	19	In Werkstückkoordinaten.....	23
Endpunkt, Distanz vor.....	107	Handrad in Werkstückkoordinaten.....	23
Eulerwinkel.....	166	Aktivieren.....	29
Drehung um die jeweils neue Koordinate....	162	Applizieren.....	27
Drehung um raumfeste Koordinaten.....	163	Deaktivieren.....	30
Eulerwinkelorientierung - Varianten.....	162	Handrad-Anwahl.....	24
Gemischte Drehung um die jeweils neue		koordinatenanwahl im WCS.....	24
raumfeste Koordinate.....	164	Helpdesk.....	681
F		High-Speed-Signal.....	76
Fahren gegen Messtaster (G75).....	593	Hirrhachse.....	15
Inbetriebnahme.....	595	in Grad programmiert.....	16
Feedback.....	9	Platzprogrammierung.....	16
FeedMode.....	53	Höhenregelung für Digitalisieren.....	637
FeedVal.....	53	Abweichungen überwachen.....	643
feprom/schemas/sdaxtrf.xsd.....	193	Applizieren.....	645
Fliegendes Messen (FME).....	603	Ausschalten.....	646
Inbetriebnahme.....	605	Einschalten.....	646
G		Kollisionserkennung.....	644
G33.....	434	Locherkennung.....	644
Gantry-Achsen.....	485	Hotline.....	681
Aktivieren.....	506	I	
Applizieren.....	498	Interface-Signale, Satzvorlauf.....	68
Ausgleichsbewegung, Signal-Ablauf	490	K	
Deaktivieren.....	506	Kalibrierdatei.....	413
Ermittlung der korrekten Bezugsmaße.....	504	Kalibrieren von Achs-Kinematiken - NC-Be-	
IB-Überwachung.....	496	fehle.....	409

Berechnung optimaler Parametersatz.....	410
Lesen der aktuellen Parameter aus der NC.....	411
Rückwärtstransformation mit aktuellen Maschinenparametern.....	413
Schreiben der optimierten Parameter in die NC.....	412
Vorwärtstransformation mit vorgegebenen Parametern.....	412
Kalibrierzyklus.....	418
CalibDirect.cnc.....	419
CalibIndirect.cnc.....	422
direktes Messverfahren.....	418
indirektes Messverfahren.....	420
Kanalsteuerung.....	31
Kardanische 5-Achs-Transformation.....	298
mit linearer Orientierungsbewegung.....	308
Kardanische 5-Achs-Transformation RLLLR....	320
Kollision	
Achskollisionsüberwachung.....	101
Konfiguration von Skalierungsfaktoren für Rundachsgeschwindigkeiten.....	85
Koordinatenkopplung	
Selektiv additiv.....	517
Kopplungen.....	485
Gantry-Achsen.....	485
Kopplungstabellen.....	526
Selektive additive Koordinatenkopplung.....	517
Selektive additive Koordinatenkopplung mit Tabellen.....	521
Systemachskopplungen.....	530
Kopplungstabellen.....	526
Approximation von f(pm) zwischen den Stützpunkten.....	528
Spline-Tabellen erzeugen.....	528
Syntax.....	526
Kritik.....	9
Kubische Hermite-Splines.....	133
Kundenfeedback.....	9
Kurvenparameterinterpolation.....	90
L	
Laserbearbeitung.....	457
Aktivieren.....	460
Applizieren.....	460
Bahngeschwindigkeitsabhängige Laserleistungssteuerung.....	457
Deaktivieren.....	463
Laserleistungssteuerung, bahngeschwindigkeitsabhängig.....	457
Lineare Moduloachse.....	13
Lineare Orientierung.....	177
LockTargBlock.....	53
LockWarning.....	54
Look Ahead.....	68

M	
Manueller Betrieb.....	19
Messen auf eine Satz (MOB).....	610
Messen auf einer Kontur (MOC).....	612
Aktivieren.....	617
Applizieren.....	617
Deaktivieren.....	617
Messwerte per CPL-Funktionen ermitteln....	616
Messwerte per Systemdatum ermitteln.....	615
Reaktionen auf das Messergebnis.....	613
Messfunktionen.....	593
Aufzeichnungsdatei.....	634
Digitalisieren.....	620
Fahren gegen Messtaster (G75).....	593
Fliegendes Messen (FME).....	603
Höhenregelung für Digitalisieren.....	637
Messen auf einem Satz (MOB).....	610
Messen auf einer Kontur (MOC).....	612
Minderbestückte 5-Achs-Transformation mit Schwenk-/Rundtisch.....	310
MOB.....	610
MOC.....	612
Moduloachse	
lineare.....	13
Momenten-Reduktion.....	93, 94
Aktivieren.....	94
Applizieren.....	94
N	
NameOfAsup.....	54
NameOfInitAdjPrg.....	54
NC-Funktionen mit Hochsprachensyntax	
Abgleich modularer NC-Funktionen.....	56
NC.SystemData_U.....	663, 676
NC.SystemDataEvent.....	663, 676
Nenner-Polynom.....	125
Nibbeln.....	451
Nicht-bestimmungsgemäßer Gebrauch.....	12
Folgen, Haftungsausschluss.....	11
Nichtorthogonale 2- und 3-Achs-Transformation.....	214
Applizieren - nichtorthogonale 2-Achs-Transformation in XY-Ebene.....	220
Applizieren - nichtorthogonale 2-Achs-Transformation in XZ-Ebene.....	221
Nichtorthogonale 2/3-Achs-Transformation	
Aktivieren.....	222
Applizieren.....	218
Deaktivieren.....	223
O	
Oberflächenprofil aufnehmen.....	620
OPC-Items, relevante.....	663, 676
NC.SystemData_U.....	663, 676
NC.SystemDataEvent.....	663, 676
OptFeedVal.....	53
Orientierung.....	169

Index

Orientierungsbewegung.....	161	Rückwärtstransformation.....	187
Orientierungsbewegung, linear	177	Rückzugsvektor als Systemdatum.....	109
Orientierungstensor.....	168	Aktivieren.....	112
Bewegung.....	182	Deaktivieren.....	112
Drehgeschwindigkeit	183		
Orientierung mit dem 3x3-Orientierungs- tensor.....	168	S	
Singularitäten.....	184	Satzvorlauf.....	40
Orientierungstypen.....	169	Allgemeines zum Satzvorlauf.....	40
Lineare Orientierung.....	177	Anwendungen zum Satzvorlauf.....	62
Tensororientierung.....	180	Beispiele.....	70
Vektororientierung.....	170	Funktionale Einschränkungen.....	68
Orientierungsvektor.....	166	Interface-Signale.....	68
Drehgeschwindigkeit.....	173	Look Ahead.....	68
Orientierungsbewegung.....	171	Projektierung des Satzvorlaufs.....	48
Orientierungsbewegung durch Vorgabe des Endorientierungsvektor.....	172	Spindeln.....	64
Orientierungsbewegung mit program- mierter Drehachse.....	172	Systemdaten.....	65
Programmierung der Orientierungsbewe- gung.....	171	Umschaltung von modalen Zuständen im Abgleichprogramm.....	59
Programmierung mit Spline.....	171	Umschaltung von modalen Zuständen im Aktionssatz.....	59
Singularität.....	175	Vorgehensweise beim Satzvorlauf.....	44
P		Satzwechsel über High-Speed.....	76
periodische Randbedingung.....	136	Aktivieren.....	80
Polarkinematik		Applizieren.....	80
Aktivieren.....	403	fliegender Satzwechsel.....	79
Applizieren.....	400	Parameter HS.....	78
Deaktivieren.....	404	Parameter HSSTOP.....	78
Polynom-Kurve.....	90	Restweg löschen.....	79
Positionsüberwachung.....	106	SBC.....	129
PrecProg.....	116	SCARA auf Linearachse.....	223
Programmfortführung mit Satzvorlauf nach		SCARA-Achs-Transformation.....	209
Programmabbruch.....	83	Schiefe Ebene.....	166
Programmierung von Skalierungsfaktoren für Rundachsgeschwindigkeiten.....	88	Schreiben in eine Systemdatenqueue	
Projektierung des Satzvorlaufes		Applizieren.....	678
Abgleich modaler NC-Funktionen.....	56	Relevante OPC-Items.....	663, 676
Systemdaten.....	48	SSDQ.....	676
		SSDQInit.....	676
R		Sehnen-Verfahren.....	128
Randbedingungen und Abhängigkeiten		Selektive additive Koordinatenkopplung.....	517
Einstieg in Unterprogramme.....	69	Aktivieren.....	521
Vorschubprogrammierung (pro Umdr.).....	69	Applizieren.....	519
RCB.....	83	Deaktivieren.....	521
RECFILE.....	628	Selektive additive Koordinatenkopplung mit	
RecordBreak.....	630	Tabellen.....	521
RecordContinue.....	630	Aktivieren.....	525
RecordOff.....	630	Deaktivieren.....	525
RecordOn.....	630	Herstellen einer Kopplung.....	523
RecordSet.....	628	Spline-Tabellendatei erzeugen.....	524
RecordSetProbe.....	626	Service-Hotline.....	681
RECPROBE.....	626	Sicherheit / Überwachung	93
RECSEPARATOR.....	629	1-dimensionale Achskollisionsüberwa- chung.....	101
RECTIME.....	629	Achspositionsüberwachung.....	106
Rotationsachsen-Anordnung.....	340	Distanz vor Endpunkt.....	107
		Eilgang, reduzierter.....	94
		Momenten-Reduktion.....	93
		Rückzugsvektor als Systemdatum.....	109

Test-Eilgang.....	96	Punktefolgen.....	119
Test-Vorschub.....	96	rationale Monom-Splines.....	126
Sicherheitshinweise.....	7	Rationale Monom-Splines.....	125
Signalgrafik.....	7	Sehnen-Verfahren.....	128
Signalwörter.....	7	Spline mit Stützpunktprogrammierung und tangentialen Übergängen.....	127
Skalierung von Rundachsgeschwindigkeiten.....	85	Spline-Parameterlänge.....	124
Skalierung von Rundachsgeschwindigkei- ten, Konfiguration.....	85	Tangentenberechnung.....	128
Allgemeine Einstellung.....	86	Typen.....	120
Amerikanische Einstellung.....	85	Vektororientierung	130
Applizieren.....	87	Winkelbedingung, absolute	155
Europäische Einstellung.....	85	Winkelbedingung, relative	155
Skalierung von Rundachsgeschwindigkei- ten, Programmierung	88	Spline mit Stützpunktprogrammierung und krümmungsstetigen Übergängen.....	135
Handlungsanweisung.....	90	SSDQ.....	676
Spline		SSDQInit.....	676
Abstandsbedingung, absolut.....	155	Stanzen.....	451
Abstandsbedingung, relativ.....	156	Stanzen und Nibbeln.....	450
Akima-Verfahren.....	128	Aktivieren.....	453
Approximierende B-Splines mit Toleranz- angabe (Kompressor).....	122, 146	Applizieren.....	453
Automatische Ecken- und Geradenerken- nung.....	155	Deaktivieren.....	454
B-Spline.....	140	Stirnseiten-Transformation.....	379
B-Spline-Approximation.....	146	2-Achs-Transformation.....	379
B-Spline, APT-Format.....	145	3-Achs-Transformation.....	380
B-Spline, Beschreibung für Entwickler.....	142	Automatischer Zweigwechsel.....	384
B-Spline, Definition und Eigenschaften	142	Referenzstellung.....	381
B-Spline, Knotenvektor verändern.....	143	Transformationszweige.....	383
B-Spline, Kontrollpunkte	140	Werkzeuglängenkorrektur	382
B-Spline, MTX-Format.....	145	Werkzeugradiuskorrektur.....	383
B-Spline, Punktgewichte.....	141	Support.....	681
B-Spline, Punktgewichte verändern.....	145	Symbole.....	9
B-Spline, Umsetzung einer CAD/CAM- Nurbs-Sequenz in das NC-Programmfor- mat der MTX.....	145	SysAxTrafo_t.....	193
B-Splines mit Kontrollpunktprogrammie- rung.....	122	Systemachskopplung.....	530
Bessel-Verfahren.....	128	Abbrucharten aus	
Bewegung der Werkzeugorientierung	130	SysAxCoupleCmd[i].AbortAction.....	571
Bewegung der Werkzeugorientierung als C2-stetiger kubischer Spline	136	Abbruchzustände in SysAxCoupleS- ta[i].Abort.....	574
C1-stetige kubische Splines.....	127	Aktivieren.....	560
c1-stetige kubische Splines mit Stütz- punktprogrammierung.....	121	Antriebsgeführte Rückzugsbewegung.....	567
C2-stetige kubische.....	135	Antriebsgeführtes Stillsetzen.....	567
c2-stetige kubische Splines mit Stütz- punktprogrammierung.....	121	Applizieren.....	534
Integrale Monom-Splines mit Koeffizien- tenprogrammierung.....	123	Deaktivieren.....	560
Koeffizientenprogrammierung.....	123	Echtzeitschaltung.....	562
Kubische Hermite-Splines.....	133	Folgeachse, Synchronisiermode.....	556
Lineare Orientierungsbewegung als C1- stetiger kubischer Spline.....	131	Folgeachse, Synchronlaufüberwachung.....	559
mit Koeffizientenprogrammierung.....	120	Formelkopplung.....	543
mit Kontrollpunktprogrammierung (B-Spli- ne)	140	Koppelverhältnisse.....	539
programmieren.....	148	Masterkopplung mit Tabellen.....	540
		Spindelkopplung.....	565
		Spline-Tabellendatei erzeugen, laden und entfernen.....	542
		Steuerungsgeführte Reaktion.....	567
		Synchronisation.....	563
		Synchronisation, Synchronlaufüberwa- chung.....	564
		Zustände.....	538
		Systemdatenqueue aktivieren.....	679
		Systemdatenqueue applizieren.....	678

Index

MTX.....	678
Windows-Anwendungen.....	678

T

Tangentenberechnung.....	128
Tangentiale Werkzeugführung.....	184
Tangentiale Werkzeugorientierung.....	455
Aktivieren.....	456
Deaktivieren.....	456
TangTool.....	184
TangToolOri.....	184
TargDNr.....	55
TargEd.....	55
Technologie.....	425
Eckenrunden mit Splines.....	445
Gewindebohren.....	425
Gewindeschneiden.....	433
Laserbearbeitung.....	457
Stanzen und Nibbeln.....	450
Tangentiale Werkzeugorientierung.....	455
Zirkulares Eckenrunden.....	440
Tensororientierung.....	180
Test-Eilgang.....	96
Aktivieren.....	99
Deaktivieren.....	100
Test-Vorschub.....	96
Aktivieren.....	98, 99
Deaktivieren.....	99, 100
Timeout.....	54
Tippen	
Einrichtbetrieb in Werkstückkoordinaten.....	19
Tippen in Werkstückkoordinaten.....	19
Aktivieren.....	22
Applizieren.....	20
Deaktivieren.....	23
Inkrementelles Tippen.....	19
Kontinuierliches Tippen.....	19
Typ und Position der Rotationsachsen.....	339
Typ-Kennung	
Achss-Transformation.....	188

U

Über diese Dokumentation	
Gültigkeit der Dokumentation.....	1
Überwachungsfunktionen	
Gantry.....	496
Überwachungsposition.....	106
usrfep/SDDDefMTB.xml.....	193

V

Vektororientierung.....	170
Vorgehensweise beim Satzvorlauf	
Aktivieren des Satzvorlaufes.....	44
Vorgänge beim Rechenlauf.....	46

W

Warnhinweise.....	7

Wechseln von Achsen zwischen Achsgruppen.....	465
Weg- und Wiederanfahren.....	31
Aktivieren.....	38
Applizieren.....	37
Optionen zum Wiederanfahren.....	34
Wahl des Anfahrmodus.....	35
Wahl des Anfahrpunktes.....	34
Wegfahren von der Kontur per SPS.....	38
Wiederanfahren per SPS.....	38
Werkzeugführung, tangentiale.....	184
Werkzeugorientierung.....	161
Wirkstellen.....	190
AT1.....	190
AT2.....	191

Z

Zentripetale Parametrisierung.....	130
Zielgruppen.....	1
Zirkulares Eckenrunden.....	440
Aktivieren.....	442, 444
Applizieren.....	442, 443
Deaktivieren.....	443, 444
RoundEps (RND).....	441
RoundEps (RNE).....	441
Zu dieser Dokumentation.....	1
Darstellung von Informationen.....	7
Zylindermantel-Transformation.....	353
Abwicklung.....	353
Ebenenprojektion.....	355
Referenzstellung.....	357
Werkzeugkorrektur.....	357

Notizen

Bosch Rexroth AG

Postfach 13 57

97803 Lohr a.Main, Deutschland

Bgm.-Dr.-Nebel-Str. 2

97816 Lohr a.Main, Deutschland

Tel. +49 9352 18 0

Fax +49 9352 18 8400

www.boschrexroth.com/electrics



R911342627