



Steuersysteme

CNC8X9 - DUAL

Produktion und service

MEFI, s.r.o.
Peroutkova 37
150 00 PRAHA 5

tel: 251 045 113
fax: 251 045 112
e-mail: mefi@mefi.cz
[http: //www.mefi.cz](http://www.mefi.cz)



Programmieranleitung

Inhalt

1. Einführung in die Programmierung

1.1 Kodierung der Eingabeinformationen

2 Aufbau eines Part-Programms

2.1 Part-Programmwort

2.1.1. Wortaufbau

2.2 Part-Programmblock

2.2.1 Weitere Hinweise für Erstellung eines Programmblocks

2.3 Aufbau der Part-Programme, Makrozyklen und Festzyklen

2.3.1 Part-Programmanfang

2.3.2 Part-Programmende

2.3.3 Unterprogramm (PPRG)

2.3.4 Beispiele des Part-Programms mit Unterprogrammen

2.3.5 Makrozyklus (MRC)

2.3.6 Sprünge innerhalb eines Part-Programms und Makrozyklus

2.3.7 Part-Programmausdruck

3 KOORDINATENPROGRAMMIERUNG

3.1 Koordinatensystem - Koordinatenbezeichnung

3.2 Eigenschaften einer kontinuierlichen Koordinatensteuerung

3.3 Eingabe der Bahn

3.4 Nullpunkt der Maschine und des Programms

3.4.1 Definition der Maschinen-Nullpunkte

3.4.2 Referenzpunkt der Maschine

3.4.3 Programm-Nullpunkt

3.4.4 Nullpunktverschiebung (Gruppe G5)

3.4.5 Ausfüllung der Anfangverschiebungs-Tabelle

3.4.6 Definition der Verschiebung von Nullpunkten

3.4.7 Aufrufen der Nullpunktverschiebung

4. Bewegungstypen

4.1 Aufstellen von Koordinaten - Funktion G00

4.2 Linearinterpolation - Funktion G01

4.3 Kreisinterpolation - Funktion G02, G03

5 Gewindeschnieden mit Drehmeißel

5.1 Gewindeschneiden ohne Auslauf

5.2 Gewindeschneiden mit Auslauf

6. Vorschubeingabe

6.1 Konstante Schneidegeschwindigkeit

7. Werkzeugkorrektur

7.1 Eingabe der Korrekturtabelle aus dem Part-Programm

7.2 Werkzeugradius-Korrektur

7.2.1 Korrektur nach links - G41

7.2.2. Korrektur nach rechts G42

7.2.3 Werkzeugradius-Korrektur für Innenflächen

7.3 Werkzeuglänge-Korrektur

8 Programmierung von technologischen Funktionen

8.1 Verweilzeit

8.2 Spindeldrehzahl

8.2.1. Spindelstopp in einem Punkt mit Orientation und stellungbezogene Steuerung

8.3 Werkzeugkühlung

8.4 Unterbrechung und Beendigung des Part-Programms

- 8.5 Einspannung und Entspannung des Werkstückes
- 8.6 Austausch des Werkzeuges/Werkstückes
- 8.7 M-Hilfsfunktionen der Gruppe M14
- 8.8 M-Hilfsfunktionen der Gruppe M10, M11, M12, M13
- 8.9 Werkzeugnummer - T-Funktion
- 8.10 H-, P-Hilfsfunktionen

9 Parameterprogrammierung - Funktion R

- 9.1 Parameterfestlegung
- 9.2 Aufrufen von Parametern

10 Wendel

11 Parameterarithmetik

- 11.1 Allgemein
- 11.2 Notizblockspeicher
- 11.3 Struktur der Steuerparameter
- 11.4 Doppelte Genauigkeit

12 PROGRAMMIERUNG UND DEFINIERUNG VON FEESTZYKLEN

- 12.1 Erstellung von Festzyklen
- 12.2 Beispiele der Bohr-Festzyklen
- 12.3 Beispiele der Dreh-Festzyklen
 - 12.3.1 Längsschruppen - G81
 - 12.3.2 Querschruppen - G82
 - 12.3.3 Tiefe Bohrungen - G83
 - 12.3.4 Gewindeschneiden auf einer zylindrischen Oberfläche
 - 12.3.5 Gewindeschneiden auf einer zylindrischen Oberfläche mit Auslauf - G85

ANLAGEN

Anlage A - Werkzeugkorrektur in CNC836/846-Steuersystem

- Anlage A1 - Datei TAB0.KOR und gespeicherte Korrekturtabelle
- Anlage A2 - Radiuskorrektur mit Äquidistante
- Anlage A3 - Aktivierung der Radiuskorrektur
- Anlage A4 - Radiuskorrektur für Drehautomaten
- Anlage A5 - Anzeige an einem Drehautomaten bei Verwendung der Radiuskorrektur

Anlage B - Kontinuierliche Blockverknüpfung

- Anlage B1 - Nichtkontinuierliche Blockverknüpfung
- Anlage B2 - Enge Blockverknüpfung
- Anlage B3 - Manuelle Steuerung ohne kontinuierliche Blockverknüpfung
- Anlage B4 - Automatische Erkennung der kontinuierlichen Blockverknüpfung

1 Einführung in die Programmierung

Bei CNC-gesteuerten Maschinen wird für die Übertragung der werkstück- oder teilbezogenen Information kodierte Schreibweise, das sog. Part-Programm (Teil-Programm), verwendet.

Bei modernen Systemen wird diese Datenfolge in einem RAM mit Datensicherung gespeichert. Bei CNC8x6 und CNC 8x9 Steuersystem ist die Speicherkapazität für ca. 32Mbyte Partprogramme ausgelegt. Neben den Part-Programmen werden hier auch Korrekturtabellen, Startverschiebungswerte und Maschinenkonstanten gespeichert. Part-Programme (Unterprogramme, Makrozuklen, Festzyklen) werden in einer Form gespeichert, die das Steuersystem im Laufe des Echtzeit- Steuerungsprozesses leicht bearbeiten und in sofortige Befehle für die gesteuerte Maschine umsetzen kann.

Das auf einem geeigneten Medium (Lochstreifen, serieller Kommunikationsadapter, Master-HD, Diskette) gespeichertes Part-Programm wird dann über eine Peripherieanlage in den Computerspeicher übertragen. Ein Part-Programm kann entweder maschinell auf einem Computer nach Eingabe der wichtigsten Geometrie- und Technologiedaten sowie Parameter der Werkzeugmaschine (AUTOPROG, KOVOPROG, APT u.ä) oder von Hand, und zwar durch Berechnung der Endlagen aller Elementarbewegungen des Werkzeuges nach Ergänzung der erforderlichen technologischen Funktionen erstellt werden. Das fertige Part-Programm wird auf einem geeigneten Medium (Lochstreifen, Diskette) gespeichert oder direkt in den Systemspeicher übertragen (in direkter Master-System-Verbindung), oder kann in den Systemspeicher direkt auf der Betätigungstafel geschrieben werden.

Die kodierte Aufzeichnung der Teil-Geometrie und -Technologie, die ein Part-Programm bildet, stellt eine Kette von zulässigen ISO- oder EIA-Zeichen dar. Diese Zeichenkette muß das Bearbeitungsverfahren für jeweilige Werkzeugmaschine eindeutig beschreiben und als eine Gesamtheit eindeutig identifizierbar und in Form eines Ausdruckes/einer Bildschirmdarstellung leicht verständlich und übersichtlich sein.

Bei Erstellung eines Part-Programme ist aus folgenden Daten auszugehen:

- a) Maschinengeometrie (Koordinatensystem, Achsenorientierung, Nullpunkte)
- b) Werkstückgeometrie (Möglichkeit einer Kollision zwischen Werkstück und Werkzeug während der Bearbeitung, Werkstückpositionierung im Maschinen-Koordinatensystem)
- c) Werkzeuggeometrie (Abmessungen, Form, Korrektur der Werkzeugbahn nach der Werkstückform)
- d) Geometrie des Fertigteiles (nach der Zeichnung des Teiles)
- e) Technologie- und Schneidebedingungen (Schneidegeschwindigkeit, Spandicke usw.)
- f) übrige Bedingungen die für Betrieb einer Werkzeugmaschine wichtig sind (Kühlung, Vorschub, Drehzahlen, Werkzeug- und Korrekturnummer usw.).

Die Struktur eines Part-Programms muß die vorgeschriebene Schreibsyntax berücksichtigen, um Eindeutigkeit der Darstellung, ähnlich wie bei höheren Programmiersprachen, zu gewährleisten.

Ein Part-Programm setzt sich aus einzelnen Elementaroperationen, - sog. Blöcke- zusammen. Ein Block setzt sich aus einzelnen Daten, der sog. Datenwörter, zusammen. Jedes Datenwort (bis auf einige Ausnahmen) hat eine Adresse, die die Art der Information bestimmt, sowie einen Zahlwert, der ein Maß oder eine kodierte Zuordnung zu einer Funktion oder Operation angibt.

Beispiel eines einfachen Part-Programms:

N20 G1 X10.355 Z625.50 F300 S150 T12 M0

Das Wort: N20, G1, X10.355, Z625.50, F300, S150, T12, M0

Die Adresse: N, G, X, Z, F, S, T, M

Die Nummer: 20, 1, 10.355, 625.50, 300, 150, 12, 0

Block-Nr.
Art der Bewegung
Endpunkte der Bewegung
Vorschubgeschwindigkeit
Spindeldrehzahl
Werkzeug-Nr.
Hilfsfunktion
Zeichen "Zeilenende" (LINE FEED)

1.1 Kodierung der Eingabeinformationen

Das System akzeptiert Part-Programm-Eingaben in Textform.

2 Aufbau eines Part-Programms

2.1 Part-Programmwort

2.1.1. Wortstruktur

Der elementare Baustein eines Part-Programms ist das sog. Wort (Programmbefehl). Jedes Wort besteht aus einer Adressenbuchstabe und ein- oder mehrfacher Zahl des Adresseninhaltes, ggf. mit Vorzeichen und Dezimalpunkt.

Beispiel:

X 002500
Adresse
Einführungs-Nullen
Zahlenwert

2.1.2 Schreibweise des Zahleninhaltes

- Beim Schreiben des Wertes von Hilfsfunktionen können die führenden Nullen vor der ersten bedeutungstragenden Zahl entfallen. Z.B. F 1.4 S 35 uä.
- Beim Schreiben der Koordinaten ohne Dezimalpunkt können die führenden Nullen vor der ersten bedeutungstragenden Zahl entfallen, der Wert bedeutet dann die Bahn in mm. Z.B. X 3620 = 3620 mm.
- Beim Schreiben der Koordinaten mit Dezimalpunkt können die führenden Nullen vor der ersten gültigen Zahl vor dem Dezimalpunkt sowie nach der letztgültigen Zahl nach Dezimalpunkt entfallen.

Z.B. X 36.12 = 36.12 mm
 X 0.2 = 0.2 mm
 X 384. = 384 mm
 X .12 = 0.12 mm

- Das Zeichen "-" muß angeführt werden, "+" darf jedoch entfallen.
- Wenn ein Wort keine Nullinformation trägt, muß mindestens eine Null gesetzt werden.
- Zwischen der Adresse und dem Zahlenwert eines Wortes kann (muß aber nicht) ein oder mehrere Leerzeichen gesetzt werden. Ein Zahlenwert, einschl. Vorzeichen, darf jedoch nicht mit einem Leerzeichen unterbrochen werden. Beim Programmierung von Parametern wird der Zahlenwert durch jeweiligen Parameter ersetzt (s. Kapitel mit Beschreibung von Parameterprogrammierung).

Z.B. X36.12 richtig
 X 36.12 richtig
 X 36.12 richtig
 X 36. 12 falsch (Leerzeichen innerhalb eines Zahlenwertes)
 X- 36.12 falsch (Leerzeichen nach dem Vorzeichen)

2.1.3 Datenworttypen

Je nach Qualität der Information, die in einzelnen Datenwörtern eines Part-Programms enthalten ist, werden die Wörter und deren Adressen grundsätzlich in zwei Gruppen aufgeteilt, und zwar Datenwörter mit und ohne Maß.

- Datenwörter mit Maß - neben einem Wert geben sie auch ein Maß an (mit "+" und "-" Vorzeichen und mit/ohne Dezimalpunkt). Hierzu gehören folgende Adressen: A,B,C,X,Z, U, V W, I, J, K, F, R, Q (für Verweilzeit)
- Datenwörter ohne Maß - enthalten nur eine Wertangabe. Hierzu gehören die Adressen D, G, M, S, T, P, H, Q (für eine Anzahl von Wiederholungen).

Manche Part-Programm-Datenwörter sind im System dauernd gespeichert, dh. sie sind auch für alle nachfolgenden Blöcke wirksam, bis sie nicht mit einem anderen Wert derselben Adresse überschrieben werden. Die andere Gruppe sind Datenwörter, die nur innerhalb eines Blocks wirksam sind, in dem sie programmiert wurden.

2.2 Part-Programmblock

Einzelne Datenwörter werden in Programmblöcke zusammengefügt. Ein Programmblock stellt elementare Informationseinheit ist, die Angabe der Geometrie und Technologie enthält. Programmblöcke haben verschiedene Längen.

Maximaler Blockformat:

Anm.:

In Klammern steht die max. Zifferanzahl der jeweiligen Adresse. Die Ziffern vor der G- und M-Adresse geben die Anzahl der Gruppen an.

Bedeutung der Adressen und entsprechenden Werten sind in der nachfolgenden Tabelle (Systemwörterbuch) zusammengefaßt. Diese Liste gibt eine komplette Übersicht der Adressen an, auf die CNC836 reagiert, einschl. jeweilige Bereiche.

Adresse	Beschreibung
%	Partprogramm-Nr., Bereich 1 - 999999
N	Partprogrammblock-Nr., Bereich 1 - 999999
G	Vorbereitungsfunktion 00 bis 99, unterteilt in 11 Gruppen
M	Technologische Hilfsfunktionen, Bereich 00 -99, unterteilt in 14 Gruppen
X,Y,Z,U,V, A,B,C	Bezeichnung der Koordinaten im Bereich -69999,99mm bis +69999,999mm, höchstens sechs Koordinaten aus der angeführten Anzahl werden unter Maschinenkonstanten 00 bis 05 eingegeben.
I,J,K	Adresse für Eingabe des Kreis-Mittelpunktes, I auch für Eingabe der Winkelverschiebung für mehrgängige Gewinde, K für Drehautomaten mit X-X-Koordinatensystem vorgesehen
D	Korrekturnummer-Adresse, Bereich 1 - 99
R	Wert des Koordinaten- oder Funktionsparameters
Q	Anzahl der Makrozykluswiederholungen Anzahl der Sprünge im Part-Programm oder die Verweilzeit
F	Verfahren in mm/min oder mm/U im Bereich 0.001 . 24,999, beim Gewindeschneiden mittels G33 Gewindesteigung von max. 99.999
L	Makrozyklus-, Unterprogramm- oder Blocknummer, in den im Part-Programm gesprungen wird, Bereich 1 - 9999
P	Hilfsfunktion, Bereich 1 - 99
&	Programmierte Werkzeuglänge-Korrektur, eine Vier-Dekaden-Funktion, zulässige Ziffern 0, 1, 2
S	Spindeldrehzahlen, Bereich 1 - 9999U/min
T	Werkzeugnummer, Bereich 1 - 699999

Übersicht der verwendeten G-Funktionen je nach Gruppen

Gruppe	Funktion	Bedeutung
0	G00	Eilgang
	G01	Linearinterpolation
	G02	Kreisinterpolation CW
	G03	Kreisinterpolation CCW
	G30	Fahrt in den Bezugspunkt aus dem Programm
	G33	Gewindeschneiden
1	G17	Wahl der XY-Ebene (nur für die Werkzeugradius-Korrektur)
	G18	Wahl der ZX-Ebene
	G19	Wahl der YZ-Ebene
	G14	Wahl der Z4-Ebene
	G15	Wahl der Y4-Ebene
	G16	Wahl der 4X-Ebene
2	G23	Kontinuierliche Blockverknüpfung
	G24	Beschleunigte Blockverknüpfung
	G05	Kopieren nach Vorlage
	G06	Kopieren aus dem Speicher (reserviert)
	G07	Kopieren in den Speicher (reserviert)
	G08	Schleppen einer/zwei Achsen nach den Steuerachsen
	G09	Wendel (Spirale, Schleppen nach Winkel)
	G10	Winkeleingabe für die Schleppachse in der Wendel
	G98	Grundzustand der 2. Gruppe, Ausschaltung aller G-Funktionen
3	G40	Löschung der Werkzeugradius-Korrektur
	G41	Radiuskorrektur links
	G42	Radiuskorrektur rechts
4	G26	Parameterarithmetik gem. R5
	G27	Parameterarithmetik gem R5 u. R6
	G28	Parameterarithmetik gem. R5, R6 u. R7
	G29	Parameterarithmetik gem. R5, R6, R7 u. R8
5	G53	Nullpunktverschiebung 0
	G54	Nullpunktverschiebung 1
	G55	Nullpunktverschiebung 2
	G56	Nullpunktverschiebung 3
	G57	Nullpunktverschiebung 4
	G58	Nullpunktverschiebung 5
	G59	Nullpunktverschiebung 6
6	G94	Vorschub in mm/min ohne konstante Schneidegeschwindigkeit
	G95	Vorschub in mm/U ohne konstante Schneidegeschwindigkeit
	G96	Konstante Schneidegeschwindigkeit (KRR) mit Vorschub mm/min
	G97	Konstante Schneidegeschwindigkeit (KRR) mit Vorschub mm/U
7	G70	Makrozyklus- oder Programmende
	G71	Unterprogramm-Aufrufung
	G72	Unterprogramm-Aufrufung
	G73	Programmsprung
	G79	Eintritt ins Makrozyklus/Unterprogramm
8	G80	Löschung des Festzyklus
	G81-G89	Festzyklen gem. ISO-Norm oder vom Benutzer programmierte Festzyklen
9	G90	Absolutwertprogrammierung
	G91	Inkrementalwertprogrammierung (Schrittmaßprogrammierung)
10	G04	Verweilzeit nach Maßgabe der Adresse Q
	G92	Eingabe der Koordinatensystem-Anfangspunkte in die Tabelle
	G93	Eingabe der Koordinatensystem-Anfangspunkte in die Tab.

Übersicht der verwendeten M-Funktionen je nach Gruppen:

Gruppe	Funktion	Bedeutung
0	Reserve	
1	M00	Programmstopp
	M01	Wahlweiser Stopp
	M02	Partprogramm-Ende
	M30	Partprogramm-Ende
2	M03	Spindelstart CW
	M04	Spindelstart CCW
	M05	Spindelstopp
	M19	Spindelstopp im orientierten Punkt
		Anm.: In die 2. Gruppe kann man weitere 16 M-Funktionen eingeben, die in Maschinenkonstanten R160-R163 definiert sind (z.B. M13 = M3 + M07 ...SpindelSTART und Kühlung)
3	M41	Spindeldrehzahlen - Bereich 1
	M42	Spindeldrehzahlen - Bereich 2
	M43	Spindeldrehzahlen - Bereich 3
	M44	Spindeldrehzahlen - Bereich 4
4	M36	Direkte Vorschubprogrammierung
	M37	Vorschubprogrammierung 1:100
5	M07	Kühlung 2 ein
	M08	Kühlung 1 ein
	M09	Kühlung 1 u. 2 aus
	M17	Kühlung 1 und 2 ein
6	M50	Kühlung 3 ein
	M51	Kühlung 4 ein
	M52	Kühlung 4 ein
	M53	Kühlung 3 und 4 aus
7	M10	Werkstückeinspannung
	M11	Werkstücklockerung
8	M49	manuelle Überbrückung der FEED OVERRIDE-Funktion
	M48	Löschung der manuellen Überbrückung der FEED OVERRIDE-Funktion
9	M06	Werkzeugaustausch
	M60	Werkzeugaustausch
10	...	Funktionen der Maschinenkonstanten (R56)
11	...	Funktionen der Maschinenkonstanten (R57)
12	...	Funktionen der Maschinenkonstanten (R58)
13	...	Funktionen der Maschinenkonstanten (R59)
14	...	Übrige Hilfsfunktionen, die in den vorstehenden Gruppen nicht angeführt wurden, Relaisausgaben in BCD-Kode

Erklärung der Symbole in der Tabelle:

x - Hilfsfunktionen, wirksam nur im Block, wo sie programmiert wurden

+ - Nr. der Funktion, die die in den jeweiligen Gruppen angeführte Nummern aufheben und die einstweilig bei MOO- und MO1-Funktionen in M2-, M5-, und M6-Gruppen und dauernd auch bei MO2-, M30 und nach Systemnullung auftreten.

2.2.1 Weitere Hinweise für Erstellung eines Programmblocks

- Nach Adressen 4 oder 5 kann Buchstabe A,B,C,U,V,W (nach Absprache mit dem Maschinenhersteller) programmiert werden.
- Nach Adresse 6 kann Buchstabe A,B,C (nach Absprache mit dem Maschinenhersteller) programmiert werden.

- KOMMENTAR ist eine Kette von beliebigen Zeichen (außer Anführungszeichen), die zwischen den Anführungszeichen geschlossen ist. Falls nach dem Kommentar ein Zeilenende-Zeichen (CR,LF) steht, können Anführungszeichen entfallen.

Beispiel:

```
N10 X100 Y100 "DIES IST DER KOMMENTAR"
N20 X200 Y200 "DIES IST DER KOMMENTAR
" "
"
N30 X100 "DIES IST DER KOMMENTAR" Y200 Z300
N40 X200 "DIES IST DER KOMMENTAR Y200 Z300
```

Anm. Im N40-Block schließt der Kommentar auch Y200 Z300 in sich, da das zweite Anführungszeichen nicht gesetzt ist und das Zeilenende als Kommentarende verstanden wird!

- Jeder Programmblock muß mit einer N-Adresse (Block-Nr.) begonnen und mit N-Zeichen (Nr. des nachfolgenden Programmblocks) oder mit "*" - Zeichen beendet werden. Übrige Blockwörter können in beliebiger Reihenfolge programmiert werden. Es empfiehlt sich jedoch, Standard-Wortfolge zu benutzen (s. maximaler Blockformat). Falls ein Part-Programm vom System gelocht oder geschrieben wird, werden die Wörter in einem Block in diese Standardfolge angeordnet.
- Das System arbeitet mit einer variablen Blocklänge. Das bedeutet, daß jeder Block beliebige Anzahl der Wörter enthalten kann. (Auch ein leerer Block, in dem nur die Block-Nr. programmiert ist, kann gespeichert werden, was jedoch in der Praxis nicht vorkommt). Jede Adresse (Wort) kann in einem Block nur einmal auftreten, die Ausnahme davon bilden die Gruppenfunktionen (M,G), in denen Werte aus jeder Gruppe und R-Adresse stehen kann, die dann mehrmals in dem Block verwendet werden können (höchstens 32x).
- Ein Programmblock kann beliebige Anzahl von Zeichen enthalten (einschl. Trennzeichen u. Kommentar). Auf einer Zeile (zwischen CR u. LF) können jedoch max. 80 Zeichen stehen.
- Anstatt des Adressenwertes können in allen Adressen (außer N-Adressen) R-Parameter-Nr. geschrieben werden - s. Abschnitt über Parameterprogrammierung.
- Für Trennen einzelner Blockwörter kann beliebige Anzahl der Zwischenräume und Tabulatoren gesetzt werden. Die Zwischenräume und Tabulatoren können auch innerhalb eines Wortes gesetzt werden, jedoch nur zwischen der Adresse und der Nummer (nie innerhalb eines Zahlenwertes!).
- Auf beliebiger Stelle den Programmblockzeichen (zwischen einzelnen Wörtern auch innerhalb eines Wortes) kann ein beliebiges weggelassenes BL-, RS-, CR- oder DEL-Hilfszeichen gesetzt werden, das vom System nicht berücksichtigt wird. Nacheinander können höchstens 10 BL-Zeichen stehen.
- DEL-Zeichen bewirkt Nichtberücksichtigung eines Blocks von Anfang (Adresse N) bis zu diesem Zeichen. Nach ESC-Zeichen muß der geänderte oder nachstehende neue Block in die N-Adresse geschrieben werden.
- "/"-Zeichen (Schrägstrich) ist die einzige Adresse, wo kein Wert steht (die nachstehende Ziffer wird nicht ausgewertet). Bei Anwählen des "/"-Modus (Blockunterdrückung) wird der mit "/"-Zeichen angeführte Block bei Abarbeitung eines Part-Programmes nicht berücksichtigt.
- Wenn in einem Block eine Verweilzeit (GO4-Funktion) zusammen mit einer Funktion der G7-Gruppe vorgegeben ist, hat die Q-Funktion Priorität für die Verweilzeit. In diesem Fall werden die Unterprogramme, Makrozyklen, bzw. Sprünge nur einmal ausgeführt (gleich wie bei Q=1).

2.3 Aufbau der Part-Programme, Makrozyklen und Festzyklen

Das Part-Programm für ein Teil besteht in der Regel aus dem eigentlichen Part-Programm und aus einem Teil der sog. Unterprogramme. Die Unterprogramme beziehen sich lediglich auf das Part-Programm, und sie müssen unmittelbar nach seinem Ende geschrieben (gelocht) werden. Ein Part-Programm muß nicht unbedingt Unterprogramme enthalten.

2.3.1 Part-Programmanfang

Am Anfang eines Part-Programmes muss immer Part-Programm-Nr. stehen. Es wird mit %-Adresse und einer positiven Zahl (höchstens 6-Dekaden, anders als Null) ohne Dezimalpunkt eingeführt werden. Diese Zahl identifiziert das jeweilige Part-Programm innerhalb eines Systems.

Beispiel:

(1 ist identisch mit 000001)

Falls während des Part-Programmeinlesens aus einem Eingabegerät im READ-Modus kein %-Zeichen auftritt, werden die Zeichen nicht berücksichtigt, d.h. vor dem %-Zeichen kann ein beliebiges Zeichen außer Null stehen.

Wenn sich z.B. auf einem Lochstreifen mehrere Part-Programme befinden und man will nur eines davon einlesen, es genügt - bei Anwendung eines Lochstreifen-Lesegerätes - den Lochstreifen beliebig vor dem Anfang des gewünschten Part-Programms einzulegen.

Man kann die Nummer des einzulesenden Part-Programms im 1. Parameter der READ-Modus anwählen. In diesem Fall wird das gewünschte Programm auf dem Lochstreifen nach dessen Nummer gesucht. Wenn im Laufe des Einleseprozesses mehr als 10 Leerzeichen nach einem Zeichen (außer Null) eingelesen werden, wird Fehlermeldung "Streifenende" angezeigt. Dies bedeutet, daß das Streifen zu Ende gelaufen ist, ohne das Part-Programm gefunden zu haben. Mit Restart kann man das Einleseprozeß fortsetzen (z.B. wenn es sich mehr als 10 Leerzeichen zwischen einzelnen auf dem Lochstreifen befindlichen Programmen stehen). Wenn mehrere Part-Programme auf einmal eingelesen werden, muss jeder von ihnen mit "*" -Zeichen abgeschlossen werden oder es müssen mindestens 10 Leerzeichen nach dem nicht abgeschlossenen Programm stehen. In diesem Fall wird jedoch Fehlermeldung "Streifenende" angezeigt.

2.3.2 Part-Programmende

Das logische Ende eines Part-Programms wird mit Programmierung von M02- oder M30-Funktionen bestimmt, und zwar in dem letzten Part-Programmblock. Nach dem Block mit M02- oder M30-Funktion können in dem Part-Programm nur Blöcke einzelner Unterprogramme stehen, die aus der Struktur des Part-Programms "abgerufen werden".

Das Ende des gesamten Part-Programms, einschl. möglicher Unterprogramme, muss mit "*" -Zeichen (Stern) abgeschlossen werden.

2.3.3. Unterprogramm (PPRG)

Das Unterprogramm bedeutet eine Gruppe von Programmblöcken, die mit einem Standardblock beginnen und abgeschlossen sind. In logischer Hinsicht wird ein Unterprogramm nur zum jeweiligen Part-Programm zugeordnet und kann nur aus diesem Part-Programm abgerufen werden.

Jedes Unterprogramm muß unbedingt mit dem Block beginnen, in dem neben Block-Nr. nur G79-Funktion und L-Adresse mit dem Wert angeführt ist, der die Unterprogramm-Nr. festlegt.

Die Blocknummer hat für die Identifizierung eines Unterprogramms keine Bedeutung. Der Einführungsblock ist wie folgt strukturiert:

N.... G79 Lxxxx

Unterprogramm-Nr.

Unterprogrammanfang

Blocknummer

Das Unterprogramm muss mit dem Block abgeschlossen werden, in dem G-70-Funktion angeführt ist. In dem letzten Unterprogrammblock können auch andere Funktionen, bzw. Bewegungen programmiert werden. Der letzte Block ist wie folgt strukturiert:

N.... G70

Unterprogrammende

Beim Einlesen des Part-Programms müssen jeweilige Unterprogramme zusammen mit dem Part-Programm eingelesen werden, dessen Teil sie sind (unmittelbar nach dem Block, wo M02- und M30-Funktionen programmiert werden).

Insgesamt können 99 Unterprogramme programmiert werden. In dem jeweiligen Part-Programm, das mit "%" beginnt und mit "*" beendet wird (eigentliche Part-Programm mit eventuellen Unterprogrammen), darf nicht mehr als ein Programmblock mit derselben Blocknummer (Wert der N-Adresse) bezeichnet werden. Das Programm kann in jedem Block eines Part-Programms abgerufen werden. Struktur der Aufrufanweisung:

```
Nxxxx .... G71 Lxxxx Qxxxx
                        Anzahl der Unterprogramm-Wiederholungen
                        Nr. des abgerufenen Unterprogramms
                        Aufrufbefehl
```

Die Anzahl der in der Q-Funktion programmierten Unterprogramm-Wiederholungen muß nicht angeführt werden. In diesem Fall wird das Unterprogramm nur einmal abgearbeitet (gleich wie bei Q1-Programmierung). Nach Abarbeitung des Unterprogramms bleiben die Werte der Technologie- und Hilfsfunktionen in solchem Zustand, wie sie von dem Unterprogramm eingestellt wurden. Das Unterprogramm kann in demselben Umfang wie das eigentliche Part-Programm editiert werden. In einem Programmblock des Part-Programms, aus dem das Unterprogramm abgerufen wird, kann man auch andere Vorschub- und Technologiefunktionen programmieren. Der Sprung zum gewählten Unterprogramm erfolgt erst am Ende des Programmblocks, d.h. nach Durchführung anderer programmierter Operationen. Nach Abarbeitung des Unterprogramms kehrt die Steuerung zurück zum Part-Programmblock, der nach dem Block folgt, aus dem gesprungen wurde.

2.3.4 Beispiele des Part-Programms mit Unterprogrammen

```
% 35                " PART-PROGRAMM NR. 35-ANFANG "
N10 X0 Y0 Z0 G54    " ERSTER PART-PROGRAMMBLOCK "
N20 X200 G1 F300    " ZWEITER PART-PROGRAMMBLOCK "
....
....
N80 G71 L100 Q2      " AUFRUFEN DES PART-PROGRAMMS NR. 100"
....                " UNTERPROGRAMM WIRD ZWEIMAL ABGERUFEN "
....
....
N120 G71 L200        " AUFRUFEN DES PART-PROGRAMMS NR. 200"
....
....
N310 X0 Y0 Z0 G00    " GELÄUFIGER PART-PROGRAMMBLOCK "
N320 M30             " LOG. PART-PROGRAMMSENDE "
"
" ES FOLGEN ZWEI UNTERPROGRAMME
"
" UNTERPROGRAMM NR.100 "
"
N500 G79 L100        " ERSTER BLOCK DES UNTERPROGRAMMS NR. 100 "
N510 ...
N520 ...
....
....
N590 G70             " LETZTER BLOCK DES UNTERPROGRAMMS NR.100"
"
```

" UNTERPROGRAMM NR.200"

"

N600 G79 L200 " ERSTER BLOCK DES UNTERPROGRAMMES NR. 200"

N610 ...

N620 ...

....

....

N700 G70 " LETZTER BLOCK DES UNTERPROGRAMMES NR.200"

*

" STERN - PART-PROGRAMM-ENDE "

2.3.5 Makrozyklus (MRC)

Makrozyklus bezeichnet eine Gruppe von Programmblocken, die ein Part-Programm für ein typisches Drehteil oder für ein auf einer konkreten Maschine zu produzierendes Teil bilden.

Am Anfang und am Ende eines jeden Makrozyklus muss ein Block stehen, der die gleiche Form als der Anfangs- und Abschlußblock des Unterprogramms hat. Allgemein bezieht sich ein Makrozyklus zu allen gespeicherten Part-Programmen, aus denen er abgerufen werden kann (er kann aus Part-Programmen als auch aus deren Unterprogrammen abgerufen werden).

Die Makrozyklen gehören zur Gruppe der sog. Bibliothek-Part-programmen und werden in das System individuell eingelesen. In das System dürfen nicht zwei oder mehrere Makrozyklen mit derselben Nummer eingelesen werden. Die Makrozyklen werden oft für Aktionen verwendet, die allen Part-Programmen gemeinsam sind, wie z.B. Anfahren der Werkzeugaustausch-Position und der eigentliche Werkzeugaustausch. Es empfiehlt sich auch Programmierung von Parametern.

Ein Makrozyklus-Aufruf kann im beliebigen Unterprogrammblock erfolgen.

Der Aufruf hat folgendes Format:

Nxxxx G72 Lxxxx Qxxxx

Anzahl der Makrozyklus-Wiederholungen

Nr. des abgerufenen Makrozyklus

Aufrufbefehl

Der Makrozyklus kann auch aus beliebigem Block eines anderen Makrozyklus (sog. Makrozyklus-Einbettung) abgerufen werden. Nach Beendigung eines Makrozyklus bleiben die Werte der Technologie- und Hilfsfunktionen in dem Zustand erhalten, wie sie in dem Makrozyklus eingestellt wurden.

In einem Part-Programm- oder auch Makrozyklusblock (d.h. auch in dem Unterprogramm), aus dem ein anderes Makrozyklus abgerufen wird, können auch andere Vorschub- und Technologiefunktionen programmiert werden. Der Sprung zum gewählten Makrozyklus erfolgt - ähnlich wie beim Aufruf eines Unterprogramms - erst am Ende des Programmblocks. Jeder Makrozyklus muss auf die gleiche Weise wie ein Unterprogramm abgeschlossen werden:

Nxxxx G70

Makrozyklus-Ende

2.3.6 Sprünge innerhalb eines Part-Programms und Makrozyklus

Mit Hilfe der G73-Funktion kann man einen Sprung in anderen Programmblock bestimmen. Format des Blocks in dem ein Sprung programmiert ist:

Nxxxx G73 Lxxxx Qxxxx
Anzahl der Sprungwiederholungen
Nummer des Blocks, in den gesprungen wird
Sprungbefehl

Die Sprünge dürfen nur in die Blöcke innerhalb des sog. Programmkomplexes, d.h. im Rahmen eines Part-Programms, Unterprogramms oder Makrozyklus nach vorne und nach hinten (d.h. in Richtung zum Anfang oder Ende des Part-Programms, Unterprogramms oder Makrozyklus) programmiert werden. Es können also keine Sprünge aus dem Part-Programm ins Unterprogramm ausgeführt werden.

Der Wert der Q-Funktion gibt die Anzahl der Sprungwiederholungen an. Bei Q (n) wird der Programmabschnitt zwischen jeweiligen Programmblöcken (n+1)mal abgearbeitet.

Anmerkung:

L-Hilfsfunktionen (Nr. des aufgerufenen Unterprogramms, Makrozyklus sowie Block-Nr. in den der Sprung erfolgen soll) Q (Anzahl der Unterprogramm- oder Makrozyklus-Wiederholungen oder Anzahl der Sprünge bei SPRB) bezieht sich nur zum Block, in dem sie programmiert wurden.

Beispiel:

N25

N26

N27

.....

.....

N54 G73 L26 Q3

N55 „Fortsetzung auf N55

In dem Programmblock, wo ein Sprung programmiert wurde, können auch andere Vorschub- und Technologiefunktionen programmiert werden. Der Sprung in den definierten Programmblock wird erst am Ende des Blocks ausgeführt.

3 KOORDINATENPROGRAMMIERUNG

3.1 Koordinatensystem - Koordinatenbezeichnung

CNC8x6-Steuersystem kann max. 6 kontinuierlich gesteuerte Koordinaten steuern (X, Y, Z, 4., 5., 6.). In einem Block können höchstens 6 Koordinaten programmiert werden.

X-, Y- und Z-Koordinaten bilden das Koordinatensystem (Raum). Die Achsen in diesem räumlichen Rechtssystem verlaufen parallel mit den Führungsebenen der Maschine.

Die Achsenorientierung entsprechend der Definition gem. DIN 66217 und ISO/R841 lautet dann: " Die Z-Achse verläuft immer parallel/ist identisch mit der Achse des Arbeitsspindels. Die positive Richtung der Z-Achse verläuft von dem Werkstück zum Werkzeug. Die X-Achse liegt in der horizontalen Ebene, die vertikal zu Z-Achse ist, und parallel mit der Aufspannfläche des Maschinentisches verläuft. Es ist die Hauptachse für Bewegung in der Ebene. Im Falle eines rotierenden Werkzeuges mit Horizontalachse richtet sich die positive Bewegung der X-Achse nach rechts betrachtet von der Spindel zur Aufspanntisch-Ebene. Y-Achse ergänzt die Koordinaten auf normales (d.h. rechteckiges, rechtssteigendes) Koordinatensystem.. Ein Beispiel für Achsenorientierung der Horizontalfräsmaschine ist auf Bild. 4 dargestellt.

Die Koordinatenorientierung ist von dem relativen Drehsinn des Werkzeuges zum aufgespannten Werkstück abgeleitet. Die Werkzeugbewegungen werden mit Buchstaben ohne Strich gekennzeichnet(z.B. X). Wenn sich das Werkstück in einer Achse bewegt, wobei das Werkzeug unbeweglich aufgespannt ist, ist der Sinn der Werkstückbewegung umgekehrt und wird mit einer Buchstabe mit Strich (z.B. X) gekennzeichnet.

Den als 4. und 5. gekennzeichneten Koordinaten kann man eine Adresenbezeichnung U, V, W, A, B oder C zuordnen. Die Adressenzuordnung diesen Koordinaten erfolgt durch Einstellung jeweiliger Maschinenkonstanten.

Der 6.-Koordinate kann je nach Wunsch des Kunden A-, B- oder C- Adresenbezeichnung zugeordnet werden.

Bei Adressenzuordnung gilt die Regel, daß 2 Koordinaten nicht dieselbe Adresse haben dürfen.

Falls keine Spezifikation bezüglich der Adressenzuordnung in dem Angebot gegeben wurde, wird standardmäßig der 4. Koordinate die U-Adresse, der 5. Koordinate die V-Adresse und der 6. Koordinate die C-Adresse zugeordnet. In nachfolgenden Abschnitten werden diese Koordinaten nach dieser Systematik bezeichnet werden.

3.2 Eigenschaften einer kontinuierlichen Koordinatensteuerung

Alle X-, Y-, Z-, U- und V-Koordinaten werden in bei dem Arbeits- und Eilvorschub positionsabhängig gesteuert. Bei auf diese Weise gesteuerten Koordinaten besteht während des Vorschubes eine gewisse funktionale Abhängigkeit, die die Einhaltung der Bewegungsgenauigkeit auf der programmierten Bahn gewährleistet.

Alle Bewegungen (Arbeits- und Eilvorschub) werden mit kontinuierlicher linearen Erreichung der gewünschten Geschwindigkeit am Anfang der Bewegung und mit linearen Herabsetzung der gewünschten Geschwindigkeit auf Minimum am Ende der Bewegung ausgeführt. Diese Systemeigenschaft wird als An- und Auslauf bezeichnet und sie wird in alle Interpolationstypen gewährleistet (außer Gewindeschneiden) und bei Override und STOPP.

Die An- und Auslaufsteilheit (Erhöhung und Herabsetzung der Geschwindigkeit) ist für Arbeits- und Eilvorschub gleich. Diese Steilheit wird bei Einstellung von Bewegungs-Servomechanismen einzelner Koordinaten bestimmt und als eine Maschinenkonstante im System gespeichert.

Bei Bearbeitung von kontinuierlichen Kurvenlinien entfällt der An- und Auslauf zwischen einzelnen Programmblöcken. Die gewünschte Kontinuität der Bewegungsbahn zwischen einzelnen Programmblöcken wird vom System automatisch ausgewertet. Die C-Koordinate ist für Eingabe der Spindelposition für die positionsabhängige Steuerung vorbehalten.

Die Koordinatenwerte können mit oder ohne Dezimalpunkt eingegeben werden. Der Koordinatenwert ohne Dezimalpunkt bei Linearkoordinaten (X, Y, Z, U, V, W, I, J, K) bedeutet die Angabe in mm, bei Rotationskoordinaten (A, B, C) die Angabe in Grad (Dezimalpunkt muß nicht eingegeben werden).

Beispiel:

X 100	= 100 mm	= 100000 mikron
Y 100.35	= 100.35 mm	= 100350 mikron
X 0.135	= 0.135 mm	= 135 mikron
Y .002	= 0.002 mm	= 2 mikron
B 120	= 120°	
C 140.5	= 140.5°	

Der Koordinatenwert mit Eingabe des Dezimalpunktes bei Linearkoordinaten bedeutet die Angabe in mm und bei Rotationskoordinaten die Angabe in Grad.

3.4 Nullpunkt der Maschine und des Programms

3.4.1 Definition der Maschinen-Nullpunkte

Nullpunkt der Maschine (NBS) ist ein fester Ausgangspunkt des des Koordinatensystems einer Maschine. Die Identifizierung der Systm- und Maschinenkoordinaten erfolgt nach Anfahren der Ausgangs-(Referenz-) Stellung im REF-Modus. Der Abstand zwischen dem Referenz- und Nullpunkt der Maschine wird als die Maschinenkonstante in REK-Modus (s. Bedienungsanleitung CNC806) eingegeben.

Die Identifizierung des **Programm**-Koordinatensystems mit diesem Koordinatensystem (der Maschine und des Systems) erfolgt mittels Funktion G54, falls die Verschiebungswerte für G54 in der Tabelle der Anfangverschiebungen gleich Null sind.

Falls die Werte für Anfangverschiebung in G54 nicht gleich Null sind, erfolgt die Identifizierung des Programm-Koordinatensystems mittels G54 und des verschobenen Koordinatensystems des entsprechenden Programms mit Anfangspunkt NPB0.

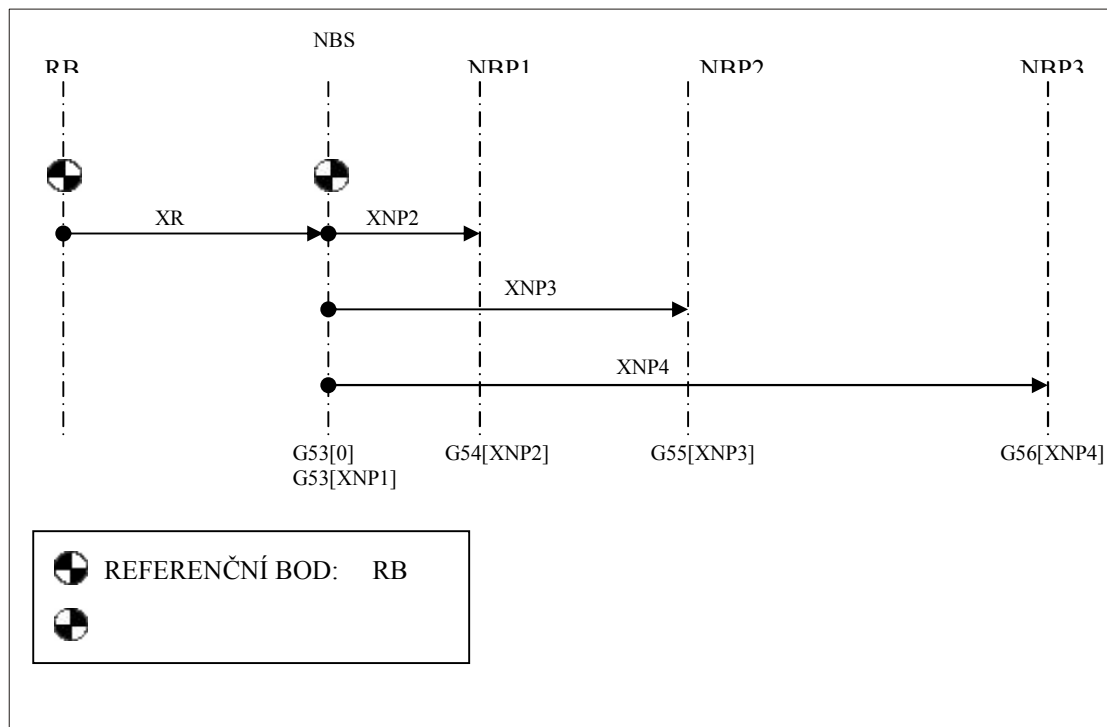
Die eingegebenen Verschiebungswerte einzelner in der Tabelle der Anfangverschiebungen angeführter Nullpunkte für Funktionen G53- bis G59 beziehen sich immer zum Nullpunkt der Maschine (NBS), abgesehen davon, ob für G54 ein anderer Verschiebungswert als Null eingegeben wurde.

Wir empfehlen, in der Tabelle der Anfangverschiebungen für Funktion G54 eine Null für Koordinatenverschiebung zu programmieren. Diese Funktion ist hauptsächlich für die Transformation des Programmkoordinaten in die Systemkoordinaten (der Maschine und des Systems) vorgesehen.

RB - Referenzpunkt der Maschine ist ein Festpunkt der Maschine, der durch Schließung des Endschalters in dem Nullpunkt des Wegmessungssystems bestimmt ist.

NBS - Nullpunkt der Maschine ist ein Festpunkt, dessen Absatand von dem Maschinen-Referenzpunkt (RB) dem Wert X_R gleich ist. Der X_R -Wert wird als Maschinenkonstante in die Tabelle der Maschinenkonstanten in REK-MOduS eingegeben und kann nicht frei geändert werden.

NBP0 - NBP6 - Programm-Nullpunkte, allgemeine Punkte, zu denen das Programm bezogen wird und deren Abstand von dem Maschinennullpunkt (NBS) dem Wert X_{NP} gleich ist (Verschiebungswert X_{NPO}) gilt für G53, Verschiebungswert X_{NP1} gilt für G54).



REFERENZPUNKT : RB
MASCHINEN-NULLPUNKT: NBS

VERSCHIEBUNG G53 HAT PRIORITÄT
EMPFOHLENER WERT $XNP1=0$

$NPBx$ = Programm-Nullpunkt
 $XNPx$ = Verschiebungswert
 XR = Abstand zwischen Referenzpunkt und Maschinennullpunkt

3.4.2 Referenzpunkt der Maschine

Jede Koordinatenachse hat einen Festpunkt an der Maschine, der als Maschinenreferenzpunkt RB bezeichnet ist. Beim Anfahren dieses Punktes in REF-Modus (Anfahren der Ausgangsstellung) wird in dem Computerspeicher für Koordinaten-Absolutwerte der X_R -Wert eingelesen, der als Maschinenkonstante eingegeben wurde; somit kommt es zur Identifizierung der Systemkoordinaten und Maschinenkoordinaten (X_R -Wert gilt für X, Y_R -Wert gilt für Y usw.)

3.4.3 Programm-Nullpunkt

Das System erlaubt Anfangverschiebung des Programm-Koordinatensystems mittels Funktionen G53 bis G59 in Punkten NBP0 bis NBP6 (Bild 3) innerhalb des Maschinenarbeitsraumes.

3.4.4 Nullpunktverschiebung (Gruppe G5)

Bei Verschiebung des Programm-Koordinatensystems (Nullpunkt) werden alle anderen Absolutmaße zum neuen Nullpunkt bezogen. Bei Inkrementalprogrammierung (Schrittmaßprogrammierung) hat die Nullpunktverschiebung für den Endpunkt der programmierten Bahn keine Bedeutung. Bei Absolutwertprogrammierung wird der programmierte Endpunkt der Bahn automatisch auf den neuen Endpunkt umgerechnet, der zum entsprechenden Nullpunkt bezogen wird. Die Anzeige der Absolutstellung (jeweils aktuelle Stellung) bezieht sich immer auf einen Maschinen- oder Programmpunkt (Referenz- Nullpunkt der

Maschine oder Programm-Nullpunkt). Der jeweilige Bezugspunkt wird mittels einer G-Funktion aus G5-Gruppe bestimmt.

Der Verschiebungswert von o.a. Punkte im Vergleich zum NBS (Maschinen-Nullpunkt) ist in der nachfolgenden Tabelle für Anfangverschiebungen angeführt:

FCE	VERSCHIEBUNGSWERTE IN ACHSEN						
Achsen:	1.	2.	3.	4.	5.	6.	
G53	1 _{NP0}	2 _{NP0}	3 _{NP0}	4 _{NP0}	5 _{NP0}	6 _{NP0}	Dauerhafte Priorität
G54	1 _{NP1}	2 _{NP1}	3 _{NP1}	4 _{NP1}	5 _{NP1}	6 _{NP1}	Dauerhafte Funktion
G55	1 _{NP2}	2 _{NP2}	3 _{NP2}	4 _{NP2}	5 _{NP2}	6 _{NP2}	Dauerhafte Funktion
G56	1 _{NP3}	2 _{NP3}	3 _{NP3}	4 _{NP3}	5 _{NP3}	6 _{NP3}	Dauerhafte Funktion
G57	1 _{NP4}	2 _{NP4}					Dauerhafte Funktion
G58	1 _{NP5}						Gültig nur innerhalb eines Programmblocks
G59	1 _{NP6}				5 _{NP6}	6 _{NP6}	Gültig nur innerhalb eines Programmblocks

Bei Programmierung der Nullpunktverschiebung innerhalb eines Programmblocks mittels Funktionen G53 bis G59 wird keine Koordinatenbewegung ausgeführt, wenn in diesem Programmblock eine andere Bewegungsdefinition programmiert wurde. Die Verwendung einer der Funktionen G53 - G59 in Programm setzt die Definierung von Verschiebungswerten in einzelnen Koordinatenwerten der Anfangverschiebungstabelle voraus. Die in dieser Tabelle definierten Werte bleiben gespeichert, bis sie mit neuen Werten in POČ-?odus überschrieben oder mittels G92 oder G93 geändert würden.

Funktionen G53 bis G57 bleiben dauernd wirksam, d.h. das Koordinatensystem, das mittels einer G-Funktion gewählt wurde, auch für alle nachfolgenden Programmblöcke gilt, bis sie mit einer anderen G-Funktion der 5. Gruppe überschrieben werden. Funktionen G58 und G59 sind nur in dem Block wirksam, in dem sie programmiert wurden, im nächsten Block kehrt das System wieder zurück, und zwar in das verschobene Koordinatensystem, das mit einer der Funktionen G53 bis G57 gewählt wurde.

3.4.5 Ausfüllung der Anfangverschiebungs-Tabelle

Die Ausfüllung der Tabelle erfolgt :

- aus Part-Programm mit Hilfe der G92-Funktion
- aus Part-Programm mit Hilfe der G93-Funktion
- auf der Betätigungstafel in ZERO-Modus

Die Ausfüllung der Anfangverschiebungs-Tabelle auf der Betätigungstafel ist in der Betriebsanleitung CNC8x6 beschrieben - s. ZERO-Modus. Im weiteren wird die Ausfüllung der Tabelle aus einem Part-Programm beschrieben.

3.4.6 Definition der Verschiebung von Nullpunkten

Unter der Definition der Verschiebung von Nullpunkten versteht man die Zuordnung von Zahlenwerten für Anfangverschiebung der koordinaten für einzelne G-Funktionen der 5. Gruppe. Diese Werte sind für einzelne G-Funktionen der 5. Gruppe in der Tabelle festgelegt. Die Eintragungen oder Änderungen der Anfangverschiebungs-Tabelle können auf folgende Weise durchgeführt werden:

- Einlesen der Werte aus einem Part-Programmblock mittels G92. Bei Eingabe und Abarbeitung des Programmblocks mit der G92-Funktion werden die Werte in die Tabelle für jeweilige Funktion des Bereiches G53 bis G59 eingetragen, die innerhalb dieses Blocks unter den Werten von jeweiligen Koordinaten programmiert wurden. Das Programmblock, in dem die Verschiebung vorgegeben wird, muss nachstehendes Format haben, z.B.:

Nxxxx G55 G92 X100.5 Y200. Z-300.650 U0 V10.

In einem solchen Block werden in die Tabelle Verschiebungswerte 100,5 für X-Achse, 200 für Y-Achse usw. geschrieben. Statt Funktion G55 kann auch eine andere G5-Funktion programmiert werden. Im Block werden die Werte in der Anfangverschiebung-Tabelle für fünf Koordinaten eingestellt, der Tabellenwert für die sechste Koordinate bleibt erhalten. Wenn im Block G92 programmiert wurde, dann bekommen die Angaben für einzelne Koordinaten die Bedeutung der Werte, die in die Tabelle eingetragen werden. In diesem Block wird keine Bewegung ausgeführt!

b) Automatische Einstellung der Tabellenwerte aus dem Programmblock mittels G93.

Nach Eingabe und Abarbeitung des Programmblocks mit G93-Funktion werden automatisch die Anfangverschiebungswerte für jeweilige Funktion im Bereich G53 - G59 eingestellt, bei der es zur Identifizierung des Programm-Nullpunktes für die jeweilige Funktion im Bereich G53 - G59 mit gewissem Punkt des Koordinatensystems kommt. Der Abstand zwischen diesem Punkt und der aktuellen Werkzeugstellung ist der programmierte Koordinatenwert in dem jeweiligen Block. Das Format des Programmblocks für diese Operation ist identisch mit dem Block, der unter a) dargestellt wurde. Der einzige Unterschied ist Programmierung der G93-Funktion anstatt G92.

Die Anwendung von G93-Funktion gem. b) wird anhand nachstehender Beispiele erklärt.

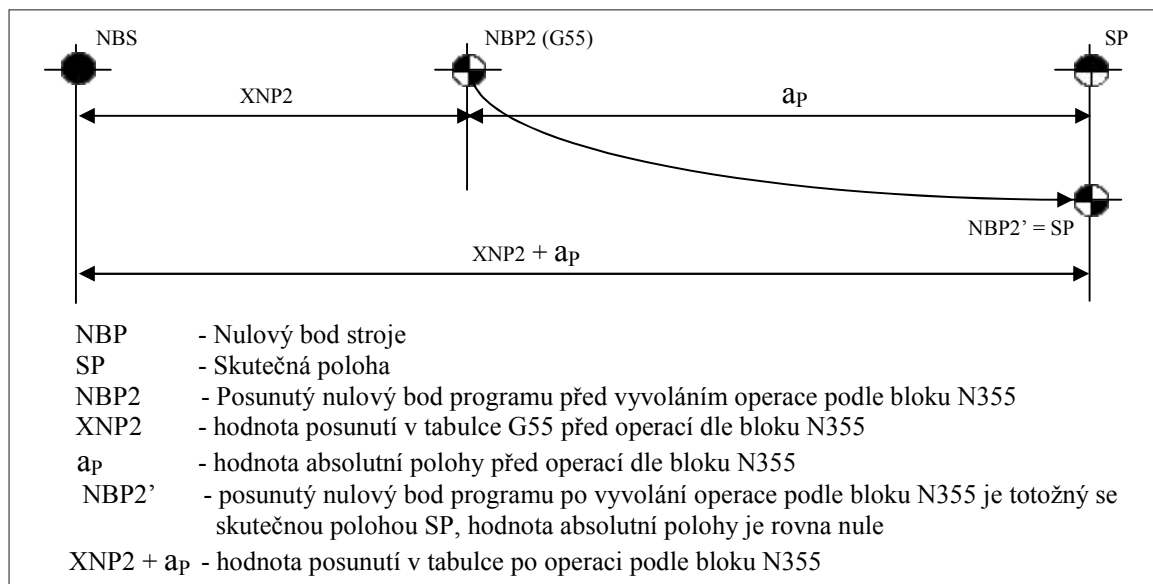
Der Wert der programmierten Koordinate ist gleich 0 (Bild 3).

In diesem Fall ist der Anfang des Programm-Koordinatensystems für eine der Funktionen G53 bis G59 mit dem Punkt identisch, in dem sich die Koordinate befindet. Für die Absolutstellung (aktuelle Stellung) wird in diesem Fall 0 angezeigt.

Programmblockbeispiel:

N355 G55 G93 X0

Bestimmung des neuen Verschiebungswertes ($X_{NP2} + a_p$):



- NBS - Nullpunkt der Maschine
 SP - Ist-Stellung
 NBP2 - Der Programm-Nullpunkt nach Verschiebung und vor Aufruf der Operation gem. Block N355
 XNP2 - G55-Verschiebungswert in der Tabelle vor Ausführung der Operation gem. N355-Block
 a_p - Wert der Stellung vor Ausführung der Operation gem. N355-Block
 NBP2' - Der verschobene Programm-Nullpunkt nach Aufruf der Operation gem. N355-Block ist identisch mit der Ist-Stellung SP, der Wert der Absolutstellung ist gleich 0
 $XNP2 + a_p$ - Verschiebungswert in der Tabelle nach Ausführung der Operation gem. N355-Block

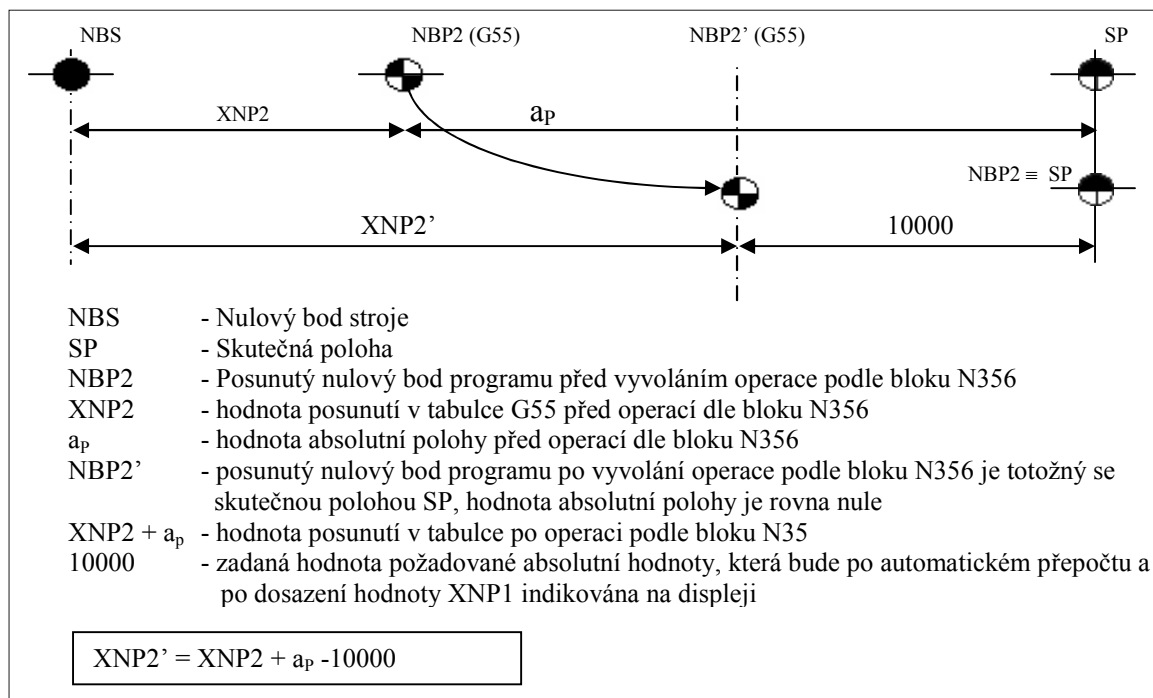
Der Wert der programmierten Koordinate ist nicht gleich 0 (Bild 4)

In diesem Fall ist der Anfang des Programm-Koordinatensystems für eine der Funktionen G54 bis G59 mit dem Punkt identisch, in dem sich die Koordinate befindet. Für die Absolutstellung (OP - aktuelle Stellung) werden in diesem Fall die programmierten Koordinatenwerte angezeigt.

Programmblockbeispiel:

N356 G55 G93 X10.000

Bestimmung des neuen Verschiebungswertes (X_{NP2}):



- NBS - Nullpunkt der Maschine
SP - Ist-Stellung
NBP2 - Der Programm-Nullpunkt nach Verschiebung und vor Aufruf der Operation gem. N356-Block
 X_{NP2} - G55-Verschiebungswert in der Tabelle vor Ausführung der Operation gem. N356-Block
 a_p - Wert der Absolutstellung vor Ausführung der Operation gem. N356-Block
NBP2' - Der Programm-Nullpunkt nach Verschiebung nach Aufruf der Operation gem. N355-Block ist identisch mit der Ist-Stellung SP, der Wert der Absolutstellung ist gleich 0
 $X_{NP2}+a_p$ - Verschiebungswert in der Tabelle nach Ausführung der Operation gem. N35-Block
10000 - Der eingegebene Soll-Absolutwert, der nach Durchführung der automatischen Umrechnung und Einsetzung des X_{NP1} -Wertes angezeigt wird.

Bild 4

3.4.7 Aufrufen der Nullpunktverschiebung

Die Verschiebung des Koordinatensystems wird während des Programmlaufes im Augenblick der Abarbeitung des Programmblocks mit G-Funktion der 5. Gruppe aufgerufen. In diesem Block, wie auch bei G-53 bis G57 in weiteren Blöcken (dauernde Gültigkeit), beginnt die Maschine sowohl die programmierten Koordinaten, als auch Werkzeugstellung im Vergleich mit dem Anfangverschiebung zu akzeptieren. Priorität unter der

Funktionen der 5. Gruppe hat G54, die beim Start eines neuen Programms automatisch als auch beim Erreichen des Programmendes automatisch angesteuert wird. Die Werte der Anfangverschiebung für Funktionen G53 bis G57 in der Tabelle bleiben solange erhalten, bis sie nicht überschrieben werden (am Anfang und am Ende eines Programms werden diese nicht beeinflusst). Den Nullpunkt-Verschiebungswert für eine der o.a. G-Funktionen können auch im Automatik-Modus definiert werden, und zwar beim Programmstopp und nach Übergang in TAB-Modus. Bei Ausfüllung der Verschiebungstabellen mittels G92 und G93 wird die Verschiebung erst ab nächstem Block wirksam.

4 Typen der Bewegung

4.1 Aufstellen von Koordinaten - Funktion G00

Unter dem Begriff "Aufstellen von Koordinaten" versteht man die Werkzeugverschiebung in dem (programmierten) Endpunkt im Eilgang. Die Endstellung wird in Absolut- oder Inkrementalwerten programmiert (gilt allgemein für alle Bewegungstypen). Die Eingabe der Werkzeugverschiebung erfolgt in G00-Block, der auch Informationen für Ausführung des Eilganges enthält. Die Eilganggeschwindigkeit ist als eine feste Maschinenkonstante einzugeben, im Block wird sie nicht vorgegeben. Beim Eilgang ist eine stufenlose An- und Ausfahrt gewährleistet. In einem Block können eine bis sechs Koordinaten programmiert werden.

4.2 Linearinterpolation - Funktion G01

Die Linearinterpolation wird mittels G01 angewählt. In einem Block kann die Interpolation zwischen einer bis sechs Koordinaten vorgegeben werden, und zwar durch Programmierung von Endpunkt-Koordinaten in den jeweiligen Achsen. Es kann also auch eine Kombination von Linear- und Kreiskoordinaten programmiert werden. Für die Interpolation muss man unter Adresse F die Vorschubgeschwindigkeit eingeben.

Beispiel des Bahnverlaufes bei Linearinterpolation in Ebene X-Y:

Absolutwertprogrammierung:
N10 G01 G90 X 90.000 Y 50.000

Inkrementalwertprogrammierung:
N11 G01 G91 X 40.000 Y 30.000

Wenn der Ausgangspunkt für eine Interpolation Punkt A (Bild 1) mit Koordinaten X=50, Y=20 ist und G54-Verschiebung gleich NBS sowie der Endpunkt B mit Koordinaten X=90 und Y=50 programmiert sind, dann wird in Blöcken N10 und N11 dieselbe Bahn gefahren.

4.3 Kreisinterpolation - Funktion G02, G03

Die Kreisinterpolation wird entweder mit G02 (Bewegung im Uhrzeigersinn) oder mit G03 (Bewegung gegen den Uhrzeigersinn) angewählt. Der Bewegungssinn der Kreisinterpolation (G02 oder G03) in beliebiger Ebene des (rechtssteigenden) Koordinatensystems wird beim Anblick der Kreisbahnebene bestimmt *Bild 5).

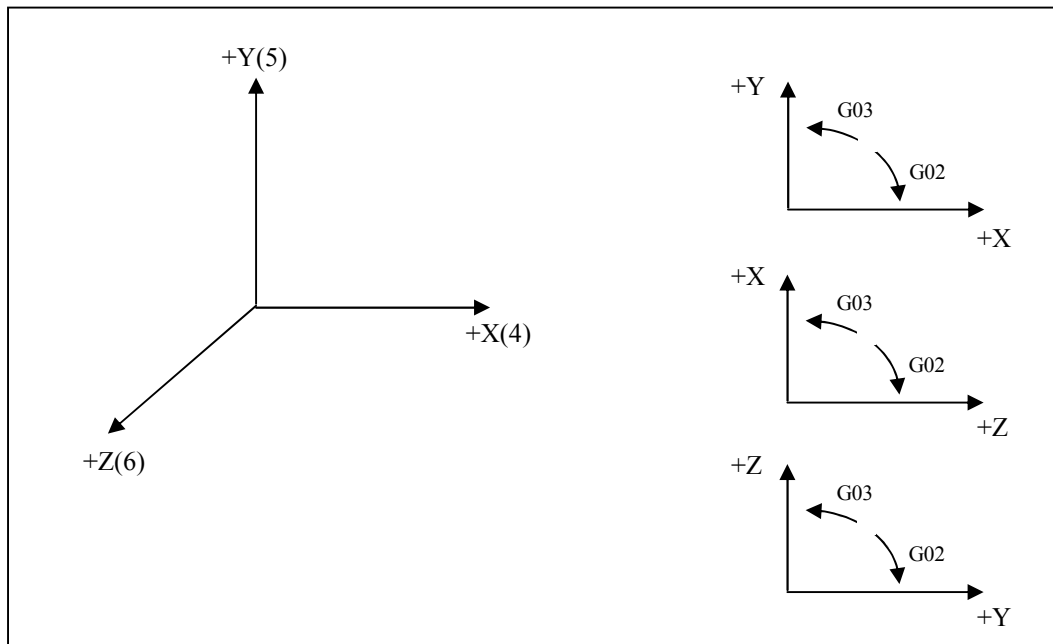


Bild 5

In Klammern stehen die Systemkoordinaten, die als Parallelen zu den ursprünglichen betrachtet werden. Die Kreisinterpolation ist nur in einer Ebene möglich. Für die Kreisinterpolation kann eine mit zwei von sechs

Koordinaten X-Y, Y-Z, Z-X, 4,-5, usw. wählen. Theoretisch kann eine Kreisinterpolation auch dann angewählt werden, wenn eine Kreiskoordinate bestimmt ist, die Programmierung ist jedoch ziemlich kompliziert.

Die Eingabe der Kreisinterpolation erfolgt durch Bestimmung der Kreis- oder Kreisbogen-Endpunktkoordinaten. Es müssen beide Endpunkt-Koordinaten eingegeben werden, auch wenn der Endpunkt identisch mit dem Anfangspunkt ist. Die Endpunkt-Koordinaten können in Absolut- oder Inkrementalwerten eingegeben werden.

Die Kreismittelpunkt-Koordinaten werden in I- und J-Adressen programmiert. Im Unterschied zum Endpunkt müssen die Kreismittelpunkt-Koordinaten ausschließlich inkremental zum Kreis-Ausgangspunkt (Ausgangspunkt der Kreisinterpolation) programmiert werden. Der Mittelpunkt angesichts der ersten Achse wird in der I-Adresse, der Mittelpunkt angesichts der zweiten Achse in der J-Adresse programmiert. Hier ist zu betonen, dass bei einer Kreisinterpolation in Z - X - Ebene wird als erste die Z-Achse und als zweite die X-Achse betrachtet (Bild 5), der Abstand zwischen dem Kreis-Mittelpunkt und -Ausgangspunkt wird demzufolge für Achse Z in der Adresse I und für Achse X in der Adresse J programmiert!

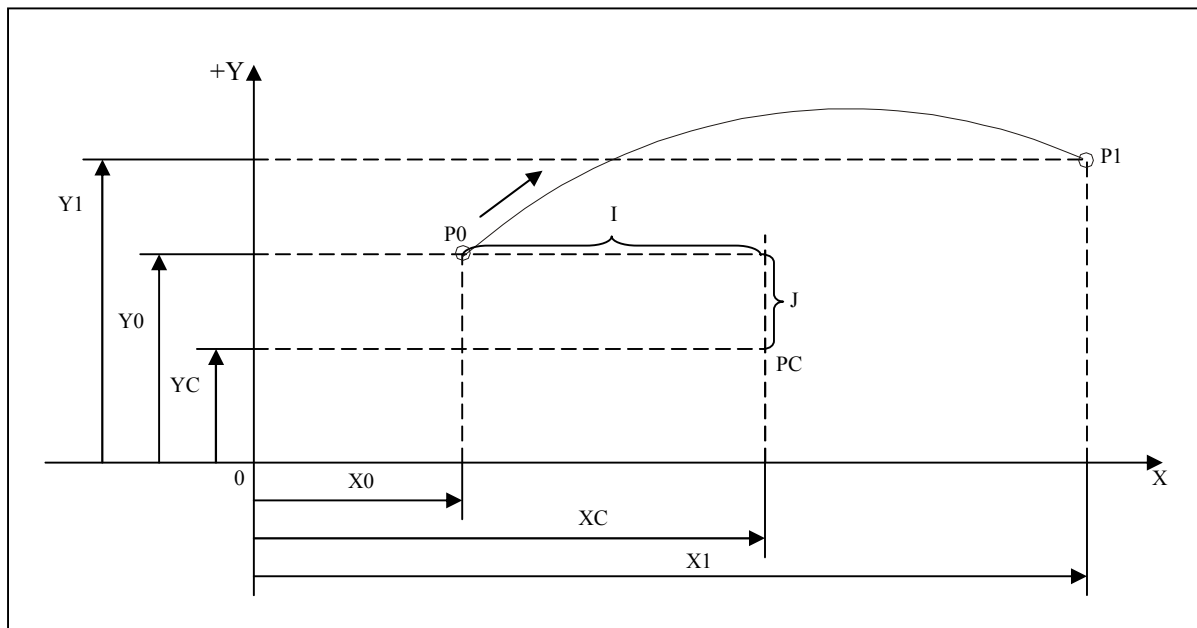


Bild 6

P0 - Anfangspunkt der Kreisinterpolation
X0, Y0 - Ausgangspunkt-Koordinaten
P1 - Endpunkt der Kreisinterpolation
X1, Y1 - Endpunkt-Koordinaten
DER FESTZYKLUS - Kreis (Kreisbogen-)Mittelpunkt
XC, YC - Mittelpunk-Koordinaten

Die auf Bild 6 dargestellte Kreisinterpolation wird mit G02 gesteuert (im Uhrzeigersinn).
Die Entfernung der Kreismittelpunkt-Koordinate für Achse X von dem Anfang ist gleich I.
 $I = XC - X0$
 $J = YC - Y0$

In Adresse J muß "-"-Vorzeichen stehen, weil sich der Kreismittelpunkt von dem Anfang in der "-"-Richtung entfernt. Wenn I- oder J-Adresse gleich 0 ist, brauchen nicht programmiert zu werden.

Programmbeispiel für einen vollen Kreis in Ebene X - Y (Bild 7a):

Absolut:

N19 G02 G90 X200.0 Y250.0 I200.0 F120

Inkremental:

N19 G02 G91 X0 Y0 I200.0 F120

Programmbeispiel für einen Halbkreis in Ebene Z - X (Bild 7b):

Absolut:

N20 G03 G90 X100.0 Z100.0 I-200.0 F100

Inkremental:

N20 G03 G91 X0 Z-400.0 I-200.0 F100

Programmbeispiel für einen Kreisbogen in Ebene X-Z (Bild 7c):

Absolut:

N21 G03 G90 X250.0 Z140.0 I-100.0 J-50.0 F100

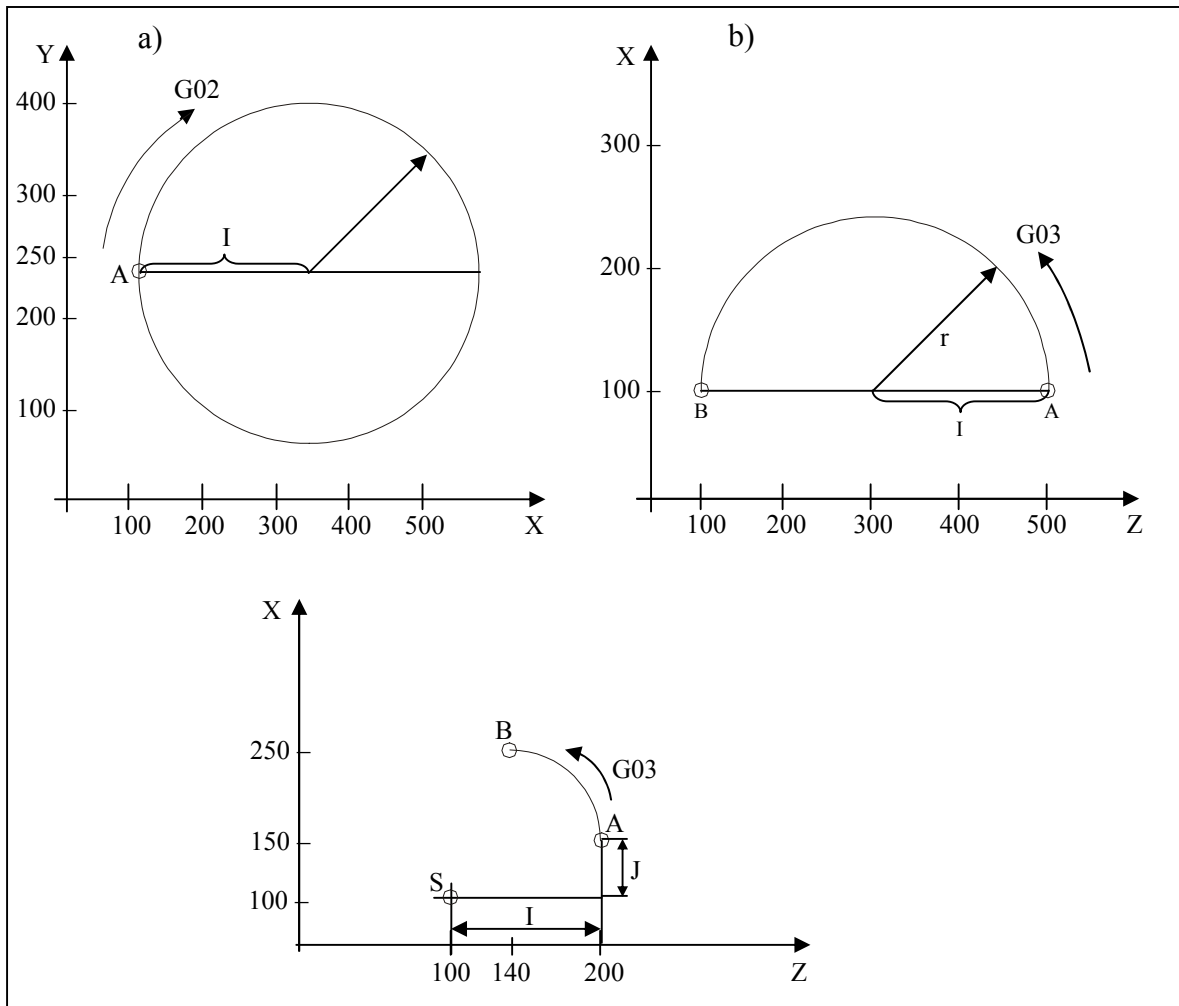
Inkremental:

N21 G03 G91 X100.0 Z-60.0 I-100.0 J-50.0 F100

Anm.

Die Kreisinterpolation ist nicht auf einen Quadrant begrenzt. Der programmierte Kreis kann mehrere Quadranten durchgehen.

Wenn bei der Kreisinterpolation Koordinaten eines Endpunktes programmiert werden, der nicht auf der Kreislinie liegt, wird Fehlermeldung 7.56 "ENDPUNKT LIEGT AUSSERHALB DER KREISLINIE" angezeigt. Die Endpunkte sowie Mittelpunkt-Koordinaten sind im Part-Programm in μm -Genauigkeit einzugeben!



5 Gewindeschnieden mit Drehmeißel

5.1 Gewindeschneiden ohne Auslauf

Bei Programmierung der Gewindeschneidoperation mit einem Drehmeißel mittels G33 wird von dem Steuersystem die Bewegung in Koordinate, für die Gewindesteigung eingegeben wurde, mit der Spindelbewegung verknüpft. Die Bewegung für andere Interpolationskoordinaten wird so ausgeführt, daß die resultierende Bahn je nach eingegebenen Koordinaten auf der vorprogrammierten Bahn verläuft.

Bei Programmierung des Gewindeschneidblocks werden folgende Adressen verwendet:

F	- Gewindesteigung in mm/U
X, Y, Z,...	- Koordinate des Gewindeendpunktes (kann absolut oder inkremental programmiert werden), nach deren Erreichung die Bewegung mit Spindeldrehung verknüpft wird
I	- Winkelverschiebung für den Anfang der verknüpften Bewegung von dem NULLIMPULS
M03, M04	- Spindeldrehsinn
S	- Spindeldrehzahl
G33	- Gewindeschneiden

Max. Gewindesteigung ist 99,99 mm/U.

Wenn I=0 oder gar nicht programmiert ist, beginnt die Verknüpfung der Impulse, die von dem Spindel-Drehzahlgeber kommen, und zwar ab dem Nullimpuls von dem Drehzahlgeber. Wenn unter der I-Adresse ein Wert programmiert wurde, gibt dieser die Winkelverschiebung des Anfangspunktes der verknüpften Bewegung von dem Nullimpuls an. Die Winkelverschiebung für Anfang der Gewindeschneidoperation ist meistens nur bei Programmierung von mehrgängigem Gewinde von Bedeutung.

Bei Programmierung der Gewindeschneidoperationen gilt die ähnliche Vorgangsweise, wie für Vorschubprogrammierung in mm/U mittels G95. Der einzige Unterschied liegt darin, dass bei G95 nicht auf den NULLIMPULS beim Vorschubstart gewartet wird.

Beispiel 1:

Gewindeschneiden in Achse Z, 1 mm-Gewindesteigung:

```
"GEWINDEAUFDREHUNG  
"GEWINDESCHNEIDEN
```

Anm.:Funktionen M03 und S20 können zusammen mit dem Gewindeschneidblock programmiert werden, d.h. die Blöcke N10 und N20 können in einen Block geschrieben werden.

Beispiel 2:

Gewindeschneiden in Achse Y, Anfang um 180° von dem Nullimpuls verschoben:

Beispiel 3:

Gewindeschneiden auf Kegeloberfläche:

5.2 Gewindeschneiden mit Auslauf

Die Programmierung ist im Prinzip gleich wie beim Gewindeschneiden ohne Auslauf. In den Block wird unter Adresse J nur die Auslauflänge (immer ein positiver Wert, der Drehsinn wird nicht berücksichtigt) eingegeben. Die Gewinde-Endstellung wird in diesem Fall einschl. der Auslauflänge programmiert. Nach dem für Gewindeschneiden-Programmblock muss ein leerer Block stehen. In nächstem Block müssen alle Koordinaten programmiert werden. Wenn keine Bewegung ausgeführt werden soll, müssen die Endstellungen erneut programmiert werden.

Beispiel 4:

Gewindeschneiden in Achse Z, Gewindesteigung 1 mm, Auslauf 2 mm:

"SPINDELAUFDREHUNG
"GEWINDESCHNEIDEN mit AUSLAUF 3 MM
"LEERBLOCK
"ALLE ACHSEN MÜSSEN PROGRAMMIERT WERDEN

In der Maschinenkonstante Nr. 8 (TAB0.REK) wird der Auslaufwinkel (höchstens 60 Grad) mit entspr. Vorzeichen eingestellt werden.

Einzelne Dekaden haben folgende Bedeutung: (zn=Vorzeichen):

In der 1. bis 5. Dekade wird der Auslaufswinkel mit Genauigkeit in Tausendstel-Größenordnung eingegeben. In unserem Beispiel ist ein Winkel von 12,345 Grad eingestellt.

In der 6. Dekade wird nur die Nummer der Achse eingegeben, in der der Auslauf aus dem Gewinde erfolgt. Bei Drehautomaten handelt es sich meistens um Achse X, d.h. Eingabe "1".(Eingabe "2" für Achse Y, Eingabe "3" für Achse Z usw.).

Die 7. und 8. Dekade haben keine Bedeutung.

Das Vorzeichen bestimmt positiven oder negativen Winkelwert (z.B. bei Innengewinde kann dieser Wert negativ sein).

Anm.

In Vorbereitung stehen Gewindeschneidezyklen, die den Programmierungsaufwand senken.

6 VORSCHUBEINGABE

Der Arbeitsablauf in der Bearbeitungskurve wird unter Adresse F programmiert. Der Vorschub wird wie gem. G6 programmiert, d.h. dass diese Funktion für den Wert in Adresse F maßgebend ist.

Bei Programmierung von G94 und G96 drückt der Adressenwert F folgende Vorschubwerte an:

a) bei Programmierung ohne Dezimalpunkt, Vorschub in mm/min.

Beispiel:

N1 G94 F1000 bedeutet Vorschub von 1000 mm/min

b) bei Programmierung mit Dezimalpunkt, Vorschub in m/min

Beispiel:

N2 G94 F0.5 bedeutet Vorschub von 500 mm/min
N3 G94 F1.2 bedeutet Vorschub von 1128 mm/min

Das System gestattet maximalen Vorschub von 24000 mm/min (24 m/min).

Bei Programmierung von G95 und G97 gibt der Adressenwert F folgende Werte an:

a) bei Programmierung ohne Dezimalpunkt Vorschub in m/U

Beispiel:

N4 G95 F1000 bedeutet Vorschub von 1000 mikron/U

b) bei Programmierung mit Dezimalpunkt Vorschub in mm/U

Beispiel :

N5 G95 F1.0 bedeutet Vorschub 1mm/U
N6 G95 F0.2 bedeutet Vorschub 0.2mm/U

Das System gestattet max. Vorschubeingabe pro Umdrehung von 99,999 mm/U (F 99999 oder F99.999). Für den Vorschub pro Umdrehung müssen die Spindelimpulse abgelesen werden, und deshalb ist es erforderlich, daß die Maschine mit System das Gewindeschneiden erlaubt. Vorschub pro Umdrehung kann auch bei manueller Steuerung angewendet werden.

Max. Programmgeschwindigkeit für das System + Maschine hängt von den Maschinenparametern ab. Bei Programmierung der Vorschubgeschwindigkeit pro Umdrehung hängt der Maximalwert des programmierbaren Vorschubs [mm/U] von der höchsten zulässigen Vorschubgeschwindigkeit V_{max} sowie von den eingegebenen Spindeldrehzahlen S ab.

Es gilt:

$$V = S \cdot \beta$$

wo:

V - Vorschubgeschwindigkeit in mm/U

S - Spindeldrehzahlen in U/min

β - Vorschub in mm/U (programmiert in Adresse F bei G95)

Den maximalen Vorschubwert für maximale Vorschubgeschwindigkeit V_{max} und eingegebene Spindeldrehzahlen S wird nach folgender Formel ermittelt:

$$\beta_{max} = \frac{V_{max}}{S}$$

und analogisch werden die maximalen Drehzahlen S_{max} für jeweils maximale Vorschubgeschwindigkeit V_{max} und für eingegebenen Vorschub nach folgender Formel ermittelt:

$$S_{\max} = \frac{V_{\max}}{\beta_{\max}}$$

Beispiel:

$$V_{\max} = 2000 \text{ mm/min}$$

$$S = 1000 \text{ U/min}$$

$$\beta_{\max} = ?$$

$$\beta_{\max} = \frac{V_{\max}}{S} = \frac{2000}{1000} = 2 \text{ mm/U}$$

Für die Kreiskoordinaten, wo die Geschwindigkeit in Tausenden von Impulsen des Drehzahlgebers pro Minute (Grad/min) eingegeben wird, wird die Geschwindigkeit in mm/min auf dem Radius R [mm] von der Drehachse berechnet, falls sich die Koordinate nach folgender Formel bewegt:

$$F_{sk} = \frac{F2\pi R}{\text{Anzahl der Tausend Impulse pro Umdrehung}}$$

Bei Programmierung von M36-Funktion (Gruppe M4) ist der Ist-Vorschubwert dem Sollwert (Programmwert) gleich. Bei Programmierung der M37-Funktion wird der Ist- gegenüber dem Sollwert im Verhältnis 1:100 herabgesetzt.

6.1 Konstante Schneidegeschwindigkeit

In REK-Modus wird bestimmt, welche Koordinate die Steuerung der konstanten Schneidegeschwindigkeit übernimmt. In den Programmblöcken mit konstanter Schneidegeschwindigkeit muss der Programm-Nullpunkt für die Steuerkoordinate in der Drehachse programmiert werden. Aufgrund der S-Adresse [Zehntel m/min] und des aktuellen Radius werden immer wieder die aktuellen Drehzahlen berechnet, die sich während der Bewegung stets ändern. Da im Laufe der Ausführung eines Programmblocks die vorgegebene Geschwindigkeit nicht umgeschaltet werden darf, ist es erforderlich der Wahl dieser Geschwindigkeit große Aufmerksamkeit zu schenken, bzw. mehrere Blöcke mit verschiedenen Schaltgängen anzuwenden. Bei G97 wird proportional auch die Vorschubgeschwindigkeit geändert. Wenn die konstante Schneidegeschwindigkeit beendet ist, ohne im nachfolgenden Programmblock den S-Wert zu programmieren, wird in die Adresse die letzterreichte Drehzahl eingesetzt. Beim Übergang von dem drehzahl- auf den minutenbezogenen Vorschub entfällt diese Programmierungsoperation für die F-Adresse. Bei manueller Steuerung findet die konstante Geschwindigkeit keine Anwendung.

Beispiel:

N10 G00 X300 Z100 M3 S100

N15 Z-1

N20 G96 G1 F200 S90 X50

N25 G00 Z100

7 WERKZEUGKORREKTUR

Die Werkzeugkorrektur erlaubt die Erstellung eines allgemeinen Part-Programms, das für verschiedene Durchmesser- bzw. Werkzeuglängen-Werte eingesetzt werden kann.

Man unterscheidet zwei Korrekturtypen:

- a) auf den Werkzeugradius - wird in G3 bestimmt
- b) auf die Werkzeuglänge - wird in Adresse &

Die Korrekturwerte werden in der Korrekturtabelle gespeichert, die insgesamt 99 Positionen enthält. Im System können demzufolge 99 verschiedene Werkzeugradius- und Werkzeuglänge-Korrekturen verwendet werden. Jede Position der Korrekturtabelle besteht aus fünf Angaben. Die erste von ihnen bestimmt den Wert der Werkzeugradius-Korrektur. Die Werte für einzelne Werkzeuglänge-Korrekturen werden in der Position als die zweite bis fünfte Angabe gespeichert. Jeweilige Tabellenposition wird mit einer D-Adresse angewählt. Die Eingabe der Korrekturwerte in die Tabelle erfolgt:

- a) von Hand auf der Bedienungstafel in RCOR-Modus
- b) mit Programmierung von G92 aus dem Part-Programm

Die Eingaben in die Korrekturtabelle können im TCOR-Modus gemacht oder geändert werden. Der Vorgang bei manueller Eingabe in die Tabelle ist in der Bedienungsanleitung der CNC806-Steuerung beschrieben. Es ist jedoch einfacher, die Eingabe in die Korrekturtabelle, bzw. in die einzelnen Positionen aus dem Part-Programm zu machen. Damit ist die Richtigkeit der Eingabewerte fürs jeweilige Part-Programm gesichert. In einem Part-Programmblock kann nur eine Position der Korrekturtabelle eingegeben werden. In der D-Adresse bestimmt man die Tabellenposition, wo die Eingabe gemacht werden soll. Im weiteren wird die Vorgangsweise bei Eingabe aus dem Part-Programm beschrieben, und zwar sowohl für die Werkzeugradius- als auch für die Werkzeuglänge-Korrektur.

7.1 Eingabe der Korrekturtabelle aus dem Part-Programm

Für die Ausfüllung der Korrekturtabelle verwendet man Part-Programme, deren Werte der jeweiligen Position der Korrekturtabelle zugeordnet werden, und zwar:

Parameter R0 ist für Werkzeugradius-Korrektur vorgesehen
Parameter R1 ist für Werkzeuglänge-Korrektur in der X-Achse vorgesehen
Parameter R2 ist für Werkzeuglänge-Korrektur in der Y-Achse vorgesehen
Parameter R3 ist für Werkzeuglänge-Korrektur in der Z-Achse vorgesehen
Parameter R4 ist für Werkzeuglänge-Korrektur in der 4-Achse vorgesehen

Wenn der jeweilige Parameter im Block programmiert werden soll, schreibt man dessen Wert in die jeweilige Position der Korrekturtabelle, die unter D-Adresse angeführt ist. Dies wird mittels G92 bestimmt.

Beispiel:

Folgende Werte sollen in Position 12 der Korrekturtabelle geschrieben werden:

Werkzeugradius-Korrektur -10,0mm, Werkzeuglänge-Korrektur von 25.5 mm in der Y-Achse und Werkzeuglänge-Korrektur von -5.0mm in der Y-Achse. Das Part-Programm wird dann folgendes Format haben:

Falls ein Parameter in dem Block nicht angeführt ist, wird der Wert in der jeweiligen Position der Korrekturtabelle nicht geändert. In unserem Beispiel wird der Wert der Werkzeuglänge-Korrektur in Achse X und 4 in Tabellenposition 12 nicht geändert, d.h. der ursprüngliche Wert bleibt erhalten.

Struktur der Korrekturtabelle:

D	I. Angabe	II. Angabe	III. Angabe	IV. Angabe	V. Angabe
1	Radius Kor.	Längen-X	Längen-Y	Längen-Z	Längen-4
2	Radius Kor.	Längen-X	Längen-Y	Längen-Z	Längen-4
3	Radius Kor.	Längen-X	Längen-Y	Längen-Z	Längen-4
usw. Bis 99					

7.2 Werkzeugradius-Korrektur

Werkzeugradius-Korrektur wird mit G-Funktion der Gruppe 3 angewählt.

G40 - Löschung der Radiuskorrektur
G41 - Radiuskorrektur nach links
G42 - Radiuskorrektur nach rechts

Die Wahl der Korrekturebene wird mit der G-Funktion der Gruppe 1 bestimmt:

G17 - Ebene X - Y
G18 - Ebene Z - X
G19 - Ebene Y - Z

Die Korrektur wird mittels G41 oder G42 sowie Adresse D gewählt. Die Werte für einzelne D-Adressen sind in der Korrekturtabelle vorgegeben. Der Wert der Radiuskorrektur kann höchstens ± 999.999 mm betragen. Die Korrekturebene wird mit G17, G18 und G19 gewählt.

Die Werkzeugradius-Korrektur kann als kontinuierlich (glatt) oder nichtkontinuierlich (nicht glatt) vorgegeben werden. Bei der Ausführung der Korrektur in der nichtkontinuierlichen Kurve wird an den Kurvenübergängen vom System automatisch erforderlicher Bogen (Bild 8) eingelegt.

Wenn die programmierte Form nur für den Eckradius gesichert ist, wobei dieser Radius größer als der Radius der verwendeten Fräsmaschine ist, ist bei Programmierung der nichtkontinuierlichen Kurve auf unentsprechende Bearbeitung der Innenecken zu achten.

Der in der Tabelle gespeicherte Korrekturwert bedeutet entweder den Unterschied zwischen dem Ist- und Soll(Programm)-Radius der Fräsmaschine bei Programmierung der technologischen Äquivalenten oder dem Ist-Radius der Fräsmaschine bei Programmierung der Werkzeugkontur.

A - B - C Werkzeugkontur
1 - 2 Einlegen eines Bogens bei einer unkontinuierlichen Bahn

Die Korrektur kann man nur in dem Block programmieren und löschen, d.h. einsetzen oder aufheben, wo die Linearinterpolation oder Eilgang vorgegeben wurden.

Das Werkzeug wird in dem nachfolgenden Programm auf der korrigierten Bahn solange gehalten, bis die Korrektur durch Programmierung von G40 nicht gelöscht oder ein neuer Wert eingesetzt wird.

Die Bewegung von dem alten zum neuem Korrekturpunkt erfolgt mittels Linearinterpolation. Der Übergang der Werkzeugradius-Korrektur aus einer Ebene in die andere ist nur dann möglich, wenn die Korrektur in der ersten Ebene gelöscht (aufgehoben) wurde.

Auf der folgenden Abbildung ist ein Beispiel für die Korrektur von rechts (G42) und von links (G41) unter Berücksichtigung des eingegebenen positiven oder negativen Korrekturwertes dargestellt.

7.2.1 Korrektur nach links - G41

Auf Bild 10 ist die Werkzeugbewegung auf einer korrigierten Bahn nach Eingabe eines positiven (D01) oder negativen (D02) Korrekturwertes sowie mögliche Korrekturverläufe, Übergang vom positiven zum negativen Wert oder vom positiven oder negativen Wert auf Null dargestellt.

Zwecks einer anschaulichen Darstellung der Funktionierung des Steuersystems bei Korrektur nach links G41 ist auf diesem Bild ein frei erfundenes Programmbeispiel dargestellt (in der Praxis kommt nicht vor). Die in diesem Beispiel verwendeten Werte der Werkzeugradius-Korrektur R_k werden aus Positionen D01 und D02 in der Korrekturtabelle genommen.

D01: Rk + 10.000 mm
D02: Rk -20.000 mm

Darstellung des Part-Programms findet man auf Bild 10
(Anm. Die Abbildung ist nicht in genauem Maßstab dargestellt)

Funktionierung des Steuersystems Bei Ausführung einzelner Programmblocke:

BLOCK BESCHREIBUNG DER FUNKTION

N1: Werkzeugradius-Korrektur erfolgt in der XY-Ebene (G17), und von links in Richtung der Bewegung (G41), Korrekturwert beträgt +10,000mm (Tabellenwert f. D01). In diesem Block erfolgt die Berechnung des neuen Werkzeug-Endpunktes unter Berücksichtigung der Korrektur und das Werkzeug fährt in diesen Punkt. Zugleich wird der Korrektur-Wert und -Richtung für weitere Programmblocke gespeichert.

N1-N4: Bewegung auf der Korrekturbahn mit dem positiven Korrekturwert wie unter N1 eingegeben.

N5: Programmierung der Korrektur nach links (G41). Der Korrekturwert beträgt -20,000 mm(D02). Das System berechnet den neuen Werkzeug-Endpunkte unter Berücksichtigung der neuen Korrektur und das Werkzeug fährt in diesen Punkt. Zugleich wird der Korrektur-Wert und -Richtung für weitere Programmblocke gespeichert.

N6-N8: Bewegung auf der Korrekturbahn mit dem negativen Korrekturwert wie in N5 eingegeben.

N9: Löschen der Korrektur G40. Der Korrekturwert beträgt 0,000 mm. Das System berechnet den neu eingegebenen Punkt mit Nullkorrektur und das Werkzeug fährt in diesen Punkt. In diesem Block wird die Korrektur gelöscht.

N10: Systemfunktion ist identisch wie die in N1.

N11: Bewegung auf der Korrekturbahn mit einem positiven Korrekturwert N12 wie unter N10 eingegeben.

N13: G40 - Löschen der Korrektur, die Fahrt in den neuen Endpunkt mit Nullkorrektur.

7.2.2. Korrektur nach rechts G42

Auf Bild 11 ist ein Beispiel für die Bewegung auf der Korrekturbahn bei Eingabe der Korrektur nach rechts (G42) und einem positiven (D01) oder negativen (D02) Korrekturwert dargestellt. Das Part-Programm für Bewegung auf der programmierten Bahn ist identisch mit dem für Korrektur nach links. Der einzige Unterschied ist Programmierung von G42 anstatt G41 in jeweiligen Programmblocken.

7.2.3 Werkzeugradius-Korrektur für Innenflächen

Bei Programmierung der Werkzeugradius-Korrektur ist das Risiko einer unrichtigen Bearbeitung von Innenecken zu berücksichtigen. Beim Einsatz eines Werkzeuges von größerem Radius als im Programm für die Bahn der Werkzeugmitte vorgesehen, kommt es zur Situation, die auf Bild 12a dargestellt ist.

7.3 Werkzeuglänge-Korrektur

Die jeweilige Werkzeuglänge-Korrektur wird mit Steuerfunktion & und Adresse D angewählt. Die einzelnen Korrekturwerte sind in der Korrekturabelle gespeichert und werden mit Adresse D angewählt. Die Zuordnung der mittels D-Adresse angewählten Korrekturwerte aus der jeweiligen Tabellenpositionen der einzelnen Koordinaten wird durch eine sog. "Steuerfunktion für Zuordnung der Werkzeuglänge-Korrekturen" gesteuert,

die unter der &-Adresse programmiert wurde. Die Funktion hat vier Dekaden (k1, k2, k3, k4), in die nur Zahlenwerte 0, 1, 2 eingesetzt werden können.

Die Bedeutung einzelner Dekaden bei Programmierung von &k1k2k3k4:

- k1=0 - in Koordinate X wird Werkzeuglänge-Korrekturwert 0 eingesetzt
- k1=1 - zur Koordinate X wird der Werkzeuglänge-Korrekturwert, d.h. II. Angabe der jeweiligen mit Adresse D angewählten Korrekturtabelle zugerechnet
- k1=2 - von der Koordinate X wird der Werkzeuglänge-Korrekturwert (II. Angabe der jeweiligen Position der Korrekturtabelle) abgerechnet
- k2=0 - in Koordinate Y wird Werkzeuglänge-Korrekturwert 0 eingesetzt
- k2=1 - zur Koordinate Y wird der Werkzeuglänge-Korrekturwert (III. Angabe der jeweiligen Position der Korrekturtabelle) zugerechnet
- k2=2 - von der Koordinate Y wird der Werkzeuglänge-Korrekturwert (III. Angabe der jeweiligen Position der Korrekturtabelle) abgerechnet
- k3=0 - in Koordinate Z wird Werkzeuglänge-Korrekturwert 0 eingesetzt
- k3=1 - zur Koordinate Z wird der Werkzeuglänge-Korrekturwert (IV. Angabe der jeweiligen Position der Korrekturtabelle) zugerechnet
- k3=2 - von der Koordinate Z wird der Werkzeuglänge-Korrekturwert (IV. Angabe der jeweiligen Position der Korrekturtabelle) abgerechnet
- k4=0 - in die 4. Koordinate wird Werkzeuglänge-Korrekturwert 0 eingesetzt
- k4=1 - zur 4. Koordinate wird der Werkzeuglänge-Korrekturwert (V. Angabe der jeweiligen Position der Korrekturtabelle) zugerechnet
- k4=2 - von der 4. Koordinate wird der Werkzeuglänge-Korrekturwert (V. Angabe der jeweiligen Position der Korrekturtabelle) abgerechnet

Die Steuerfunktion & ist nur für die Wahl der Werkzeuglänge-Korrekturwerten aus der jeweiligen Position der Korrekturtabelle wirksam. Nummer der Position ist von der Spezifikation der Adresse D abhängig. Die Steuerfunktion & und D haben dauernde Wirkung, d.h. solange sie nicht geändert werden.

Beispiel:

...zur Koordinate X wird der Wert aus der zweiten Position der Korrekturtabelle Nummer 12 zugerechnet, von der Koordinate Y wird der Wert aus der dritten Position derselben Tabelle abgerechnet, zur Koordinate Z wird kein Wert zugerechnet, zur vierten Koordinate wird der Wert aus der fünften Position der Tabelle Nr. 12 zugerechnet

Die Werkzeuglänge-Korrekturen beziehen sich ausschließlich auf den Raum XYZ und die vierte Koordinate und sind ausschließlich für simple Verschiebung in diesem Raum bestimmt, nicht jedoch für Werkzeugradius-Kompensation. Sie können maximal +/- 999,999 mm sein. (Anm. Version A des CNC806-Systems arbeitet mit Werkzeuglänge-Korrekturen für alle sechs Koordinaten).

Bei Programmierung der Werkzeuglänge-Korrektur wird die Bewegung auf der Bahn ausgeführt, die nur um das Korrekturmaß in einzelnen Achsen linear verschoben ist und deren Verlauf identisch mit der programmierten Form ist oder mit der Form, die durch Werkzeugradius-Korrektur (wenn programmiert) beeinflusst ist. Die Werkzeuglänge-Korrektur wird nur in dem Block programmiert, wo Linearinterpolation oder Eilgang eingegeben sind. Das Werkzeug wird in dem nachfolgenden Programm auf der Korrekturbahn solange gehalten, bis die Korrektur mittels Programmierung von &0-Funktion widerrufen oder ein neuer Werkzeuglänge-Korrekturwert programmiert ist.

- a) Werkzeuglänge-Korrekturwert positiv in Achse X 120mm
- b) Werkzeuglänge-Korrekturwert positiv in Achse Y 80mm
- c) Werkzeuglänge-Korrekturwert negativ in Achse X 80mm und Y 50 mm

Auf Bild 13 ist die Werkzeugbewegung bei Werkzeuglänge-Korrekturen dargestellt. Die Bahn des Part-Programms ohne Korrektur ist auf Bild 13 mit der dicken Linie dargestellt. Die dünne Linie zeigt den Verlauf eines Part-Programms der Korrektur. Das Part-Programm ist für alle Darstellungen gleich. Für den Verlauf der Korrektur gem. a) und b) werden folgende Daten in die Korrekturtabelle geschrieben:

Für den Verlauf der Korrektur gem. c) werden folgende Daten in die Korrekturtabelle geschrieben:

Weitere Part-Programmblöcke können z.B. folgendes Format haben (es wird angenommen, daß die X- und Y-Koordinaten auf Null stehen):

"Länge-Korrektur gelöscht"

Es soll bemerkt werden, daß die Richtung der Korrektur in diesem Fall durch die Steuerfunktion & bestimmt ist. Das gleiche Resultat bekommt man, wenn man bei Eingabe in die Tabelle (Block N1) negative Werte schreibt und unter die Steuerkonstante & 1 anstatt 2 und 2 anstatt 1 eingibt. Block N10 würde dann folgendes Format haben:

Bei Abarbeitung einzelner Blöcke dieses Programms werden folgende Operationen durch:

Block 1 Eingabe der angeführten Werte in die D1-Position der Korrekturtabelle

Block N10 Bei Bewegung auf Koordinaten x100.0 Y120.0 wird die Werkzeuglänge-Korrektur gem. Steuerfunktion & eingesetzt. In dem Fall wie unter a) beschrieben, erfolgt die Umrechnung des Endpunktes in Achse X auf $100.0 + 125.0 = 225.0$, Y bleibt ohne Änderung (Punkt A1 auf Bild a).

In dem Fall wie unter b) beschrieben, erfolgt die Umrechnung des Endpunktes in Achse Y auf $120.0 + 80.0 = 200.0$, X bleibt ohne Änderung (Punkt A2 auf Bild b).

In dem Fall wie unter c) beschrieben, erfolgt die Umrechnung von X und Y. Der umgerechnete Wert X ist $100.0 - 80.0 = 20.0$, Y ist $120.0 - 50.0 = 70.0$ (Punkt A3 auf Bild c).

Block N20 Ähnlich wird auch die Umrechnung der Endpunkte mittels Funktion & durchgeführt (Funktion & hat dauernde Wirkung)

Block N60 Löschen der Korrektur für beide Achsen. Der Endpunkt ist dann identisch mit dem im Block programmierten Wert.

Programmierung der Werkzeugradius- und Werkzeuglänge-Korrekturen in einem Part-Programmblock.

In einem Part-Programmblock kann gleichzeitig - und sogar in einer Ebene - Werkzeugradius- und Werkzeuglänge-Korrektur programmiert werden. Es müssen jedoch die bereits beschriebenen Programmierungsregeln berücksichtigt werden. Das System führt bei der Ausführung eines Programms die Werkzeugradius-Korrektur und ihre Aufteilung in einzelne Achsen separat von der Werkzeuglänge-Korrektur in einzelnen Achsen durch. Die Endpunkte werden folglich als die Summe der Werkzeugradius- und Werkzeuglänge-Korrektur bestimmt.

8 PROGRAMMIERUNG VON TECHNOLOGISCHEN FUNKTIONEN

8.1 Verweilzeit

Die Verweilzeit wird mittels G04-Funktion aus G10-Gruppe programmiert. Diese Funktion besitzt nur eingeschränkte Wirkung, und zwar für den Block, in dem sie programmiert wurde. Die Länge der Verweilzeit wird mit einem Wert der Q-Funktion definiert, die 4 Dekaden hat und kann demzufolge Werte im Bereich 0 - 9999 gewinnen. Eine Einheit bedeutet einen Zeitabschnitt von 10 ms. Man kann also eine Verweilzeit im Bereich 10 ms bis 99,99 s in 10 ms-Schritten programmieren.

Beispiel:

Q10		10 x 10 ms	= 100ms
Q100		100 x 10 ms	= 1000ms = 1 sec.
Q1000		1000 x 10 ms	= 10000ms = 10sec.

8.2 Spindeldrehzahl

Die Spindeldrehzahl wird unter Adresse S programmiert. Den Drehzahlwert kann man im Bereich 0 - 9999 U/min (d.h. 4 Dekaden) programmieren. Der Drehzahlbereich eines Schaltganges wird in Funktionen M41, M42, M43, M44 programmiert.

Drehzahlstart in CW-Sinn wird unter M03, Drehzahlstart in CCW-Sinn wird unter M04 programmiert. Nach dem Start mittels dieser Funktionen dauert der Drehbetrieb bis zum M05 (Spindel STOPP) oder M19 (Spindel STOPP in einem Punkt mit Orientierung).

Die S-Funktion kann das System an seinem Ausgang entweder in BCD-Kode senden, meistens sendet es jedoch am Ausgang die Analogspannungssignale im Bereich von +/- 10V für Ansteuerung des Spindelantrieb-Reglers.

8.2.1. Spindelstopp in einem Punkt mit Orientation und die stellungsbezogene Steuerung

Der Spindelstopp in einem Punkt mit Orientation wird mittels M19 programmiert. Nach dem Stopp in diesem Punkt an hält der Spindel in dem definierten Punkt und es kommt zur Schließung der Stellungsverbindung. Von jetzt an kann der Spindel stellungsbezogen gesteuert werden.

Die stellungsbezogene Spindelsteuerung dient zur Verdrehung der Spindel in beliebige Stellung mit Genauigkeit von 1/n Umdrehung, wo n die Anzahl der gesendeten Drehzahlgeber-Impulse pro 1 Spindelumdrehung ist. Diese Steuerung erfolgt mit Geschwindigkeit F, wo $F1 = F0,001$ 1000 Impulse/Minute bedeutet.

In einem Block kann man die Spindelverdrehung höchstens um 69 99 99 Impulse programmieren. Der Verdrehungswert wird in der Regel unter Adresse C oder auch A und B vorgegeben. Die Unterbrechung der stellungsbezogenen und Übergang zum geschwindigkeitsbezogener Steuerung erfolgt durch Programmierung von M03-, M04- oder M05-Funktionen.

8.3 Werkzeugkühlung

Für die Steuerung der Maschinenkühlung sind im System zwei Gruppen von M-Funktionen (M5, M6) vorgesehen, die 4 separate Kühlungskreise steuern. Die Nummern der M-Funktionen sind in der Gesamtübersicht der CNC806-Systemadressen angeführt.

8.4 Unterbrechung und Beendigung des Part-Programms

Im System können vier M-Funktionen (M00, M01, M02, M30) programmiert werden. Diese Funktionen unterbrechen oder beenden das Programm, der gerade abgearbeitet wird.

M00 - Unbedingter Stopp

Das laufende Programm wird nach Ausführung aller Operationen im Block unterbrochen, in dem diese Funktion programmiert wurde. Zugleich werden in Gruppen M2, M5 und M6 vorübergehende M-Funktionen auf die mit "+" bezeichnete Werte eingestellt (s. Tabelle der Adressenbedeutung) d.h. M05 (Spindelstopp), M00 (Kühlung 1 und 2 aus), M52 (Kühlung 2 und 4 aus). Nach erneutem Start des Part-Programms mit START-Taste kehren die M-Funktionen in Gruppen M2, M5 und M6 zu ihren ursprünglichen Werten, falls sie nicht neu eingegeben wurden.

M01 - Bedingter Stopp

Das System funktioniert gleich wie bei M00, falls es in dem modifizierten Aut-M01-MOduS arbeitet.

M02 - Part-Programmstopp

Ende des Part-Programms und Rückkehr zum Part-Programmanfang. Es werden die M05-, M09- und M02-Abschlussfunktionen gesendet.

M30 - Das System arbeitet gleich wie bei M02.

8.5 Einspannung und Entspannung des Werkstückes

Mit M10 wird ein Befehl zur Einspannung und mit M11 ein Befehl zur Entspannung des Werkstückes erteilt. Das Einspannungs- und Entspannungszyklus wird von einer programmierbaren System-Schnittstelle gesteuert. Funktionen M10 und M11 gehören der 7. Gruppe an.

8.6 Austausch des Werkzeuges/Werkstückes

Mit M06 wird ein Befehl zum Werkzeugaustausch gesendet. Der Austauschzyklus wird von einer programmierbaren System-Schnittstelle gesteuert.

Mit M60 wird ein Befehl zum Werkstückeaustausch gesendet. Der Austauschzyklus wird von einer programmierbaren System-Schnittstelle gesteuert. Funktionen M06 und M60 gehören der 8. Gruppe an.

8.7 M-Hilfsfunktionen der Gruppe M14

In dieser Gruppe kann man einen beliebigen Wert für die M-Funktion aus denjenigen Werten eingeben, die in anderen Gruppen (M1 bis M9) nicht angeführt wurden. Das System sendet diesen Wert in dem BCD-Kode.

8.8 M-Hilfsfunktionen der Gruppe M10, M11, M12, M13

Diese M-Funktionen werden von dem Benutzer definiert. In jeder Gruppe können vier M-Funktionen definiert werden. Der jeweilige Code für diese Funktionen wird als Maschinenkonstanten eingegeben.

8.9 Werkzeugnummer - T-Funktion

Mit Hilfe der T-Funktion, die 8 Dekaden hat, wird die Nummer des Werkzeuges eingegeben, das aufgrund der gesendeten M06-Funktion in entsprechende Position für Bearbeitung eingestellt wurde (die Maschine verfügt über einen automatischen Werkzeugaustausch).

8.10 H-, P-Hilfsfunktionen

Diese Funktionen haben zwei Dekaden und werden in BCD-Kode gesendet. Sie sind für Steuerung der Werkstück- und Werkzeugmanipulators oder für andere von dem Benutzer gewünschte Applikationen vorgesehen.

9 PARAMETERPROGRAMMIERUNG – FUNKTION R

Das System gestattet, anstatt eines konkreten Wertes jeder Adresse den entsprechenden R-Parameter zuzuweisen. Bei Bearbeitung einer solchen Adresse wird anstatt dieses Parameters das letztgültige Wert eingesetzt, der dem Parameter in einem der vorgegangenen Blöcken zugewiesen (festgelegt) wurde. Die Parameterprogrammierung ist vor allem bei Anwendung von Festzyklen, Makrozyklen und Unterprogrammen von Bedeutung, wo in einzelnen Part-Programmblöcken in der Regel die Werte einzelner Parameter festgelegt sind und aus dem Festzyklus, Unterprogramm oder Makrozyklus aufgerufen werden. Die Parameter können jedoch auch in anderen Part-Programmblöcken bestimmt und zugewiesen werden.

Die Arbeit mit Parametern schließt 2 Teiloperationen in sich:

- a) Parameterfestlegung
- b) Parameteraufrufung

9.1 Parameterfestlegung

Bei Festlegung von Parametern werden einzelnen Parametern konkrete Werte zugewiesen. Format der Parameterfestlegung:

N1 R0=aaaa R1=bbbb R2=cccc usw.
Festgelegter Parameterwert R [0]
Parameter-Nr.
Parameteradresse R

R-Parameter werden in dem Block zusammen mit anderen Adressen festgelegt.

Beispiel:

N620 G54 G0 X0 Y0 R1=864 R2=-0.864 R3=100000 R20=250 R31=1

Nach Ausführung dieses Blocks werden in der Parametertabelle die Parameter Nr. 1, 2, 3, 20 und 31 mit neuen Werten überschrieben. Die Parameterwerte haben keine Dimension. Der erste und der zweite Parameter werden in der Tabelle als 0.864 (der zweite mit Vorzeichen "-") und Wert 100000 als 100.000 eingegeben.

Insgesamt können 96 Parameter (0 - 95) zugewiesen werden. Der Parameterwert liegt im Bereich +69999.999 bis -69999.999. Der festgelegte Wert muss jedoch mit dem zulässigen Adressenwert übereinstimmen, in der er aufgerufen wird.

ACHTUNG: Beim Einlesen des Part-Programms kann man die Zulässigkeit von Werten einzelner Parameter nicht überprüfen und für richtige Festlegung der Parameterwerte ist einzig und allein der Programmierer verantwortlich.

9.2 Aufrufen von Parametern

Beim Aufrufen von Parametern wird anstatt es jeweiligen Parameters sein bereits früher festgelegter Wert eingesetzt. Im Part-Programm wird anstatt des jeweiligen Adressenwertes Rxx angegeben, wo "xx" die Nummer des Part-Programms bezeichnet.

Beispiel:

Der Block in dem die Parameter aufgerufen werden, die im vorigen Beispiel (Block N620) festgelegt wurden.

N630 GR31 XR1 YR2 M03 SR20 TR3

Wenn keine Parameter verwendet wären, hätte der Block folgenden Format:
N630 G01 X0.864 Y-0.864 M03 S250 T100000

Parametrisierbar sind alle Adressen außer Block-Nr. und G79-Funktion.

Beispiel:

Part-Programm mit programmierten Parametern:

```
.  
.  
N2 R30=350. R31=250. R29=1000  
N3 R1=100. R2 =200.  
N4 R3=15 M03 M42 M08  
N5 G90 G00 XR1 TR3  
N6 G82  
N7 G91 XR1 XR2  
N8 G90 G82  
N9 Z500. M05  
.  
.
```

In diesem Beispiel wurden einerseits R-Parameter in einzelnen Programmblocken (N02, N03) als auch im Block zusammen mit anderen programmierten Adressen (N04) festgelegt. In N5 und N7 werden die Parameter aufgerufen.

10 WENDEL

Die Steuerung CNC836 gestattet Programmierung von Schleppkoordinaten, d.h. Steuerung einer Koordinate in Abhängigkeit von der Bewegung einer anderen Koordinate. Diese Eigenschaft findet bei Bearbeitung von Wendeln seine Anwendung, was einer der Fälle der Schleppkoordinatensteuerung in Abhängigkeit von der Kreisbewegung ist.

Die Programmierung einer Wendel gestattet bedingte Bewegung der Kreis- und Linearinterpolation.

Die Wendel wird durch Eingabe der Steigung bei der Koordinate programmiert, die geschleppt wird (die Achse der Linearinterpolation) und der G10-Funktion. Die Steigung ist der Abstand, um den sich die Schleppkoordinate während einer vollen Kreisbewegung im Rahmen der Kreisinterpolation der Steuerkoordinaten verschieben werden soll. Diese Steigung hängt weder von dem Kreisinterpolationsradius noch von den Radiuskorrekturen ab. Der Ablenkungswinkel ändert sich jedoch in Abhängigkeit von dem Kreisinterpolationsradius.

Der Programmblock, in dem die Steigung programmiert ist, enthält keine Bewegung und wird in der Regel separat, und zwar in einem Block vor der Wendel programmiert. Die Steigung bleibt bis zur Eingabe eines anderen Wertes wirksam. Z.B. im Block N10 Z1 G10 ist Steigung 1.05 mm und Z-Koordinate als die Schleppachse vorgegeben.

Die Wendelaktivierung erfolgt mit G09 und Löschung mit G98, wobei G98 in einem eigenständigen Block programmiert werden muß. Bei Wendelprogrammierung muss die Anzahl der "Gewindegänge" eingegeben werden. Diese Zahl wird mit sog. Sprung mit Wiederholung auf denselben Block programmiert.

Beispiel:

```
N5 X0 Y0 Z0 G00
N10 Z1.05 G10                " Steigung 1.05 mm
....
....
....
N50 G09 G02 X0 Y0 I100 G73 L50 Q20 " Wendelanfang
N60 X200. Y0 I100
N70 G98                      " Wendelende
N80 X... Y... Z...          " Bewegung aller Koordinaten
```

In Block N50 ist ein Kreis mit Durchmesser von 100mm (Ausgangs- und Endpunkt auf Koordinaten 0,0) programmiert. Der Kreis soll insgesamt 21mal wiederholt werden (um einen Kreis mehr als unter Adresse Q vorgegeben). Mit G73 wird ein Sprung auf den unter L-Funktion programmierten Block vorgegeben. In Block N60 ist ein Kreisausschnitt (Halbkreis) und in Block N70 die Löschung der Wendelfunktion (G09) programmiert.

11 PARAMETERARITHMETIK

11.1 Allgemein

Die Parameterarithmetik wird mittels Funktionen **G26**, **G27**, **G28** und **G29** durchgeführt. In dem Block, wo eine dieser Funktionen programmiert ist, müssen auch jeweilige Steuerparameter R5 bis R8 bestimmt werden.

Funktion **G26** führt die Operation gem. **R5** durch.

Funktion **G27** löst nacheinander 2 Operationen gem. **R5** und **R6** (zuerst R5 und dann R6) aus.

Funktion **G28** löst nacheinander 3 Operationen gem. **R5**, **R6** und **R7** (zuerst R5, dann R6 und schließlich R7) aus.

Funktion **G29** löst nacheinander 4 Operationen gem. **R5**, **R6**, **R7** und **R8** (zuerst R5, dann R6, dann R7 und schließlich R8) aus.

11.2 Notizblockspeicher

Zur Speicherung von Zwischenresultate der Parameterarithmetik-Operationen ist der **Notizbuchspeicher** vorgesehen. Dieser Speicher stellt eigentlich weitere Parameter dar, und zwar 31 bis 49.

Parameter-Nr.: 0, 1, 2,,31

Notizbuchspeicher: 32, 33, 34,,49

In diesem Speicher können nur die Resultate der Parameteroperationen gespeichert und aus dem Speicher die Operanden für weitere Parameteroperationen genommen werden. Es ist keine direkte Eingabe von Werten in diesen Speicher möglich und im System kann man mit Hilfe vom Notizbuchspeicher keine Parameteradressen programmieren.

11.3 Struktur der Steuerparameter

+/-	8	7	6	5	4	3	2	1
Vorzeichen	Operationskode		Operand 1		Operand 2		Opererand3	

Nummer des Parameters, in dem das Resultat der Operation gespeichert werden soll (s. auch Doppelte Genauigkeit)

Nummer des Parameters, der als der zweite Operand einer Operation wirksam ist (Wenn für eine Operation kein zweiter Operand vorgegeben wurde, ist diese Eingabe nicht wichtig)
(s. auch Doppelte Genauigkeit)

Nummer des Parameters, der als der erste Operand einer Operation wirksam ist (s. auch Doppelte Genauigkeit)

Operationskode

Vorzeichen "+" zeigt an, daß alle Operanden sowie das Resultat als reale Zahlen mit Dezimalpunkt präsentiert werden. (Die Operationen werden mit einer Tausendstel-Genauigkeit ausgeführt)

Vorzeichen "-" zeigt an, daß alle Operandwn sowie das Resultat als Integralwerte ohne Dezimalpunkt präsentiert werden. (Die Operationen werden mit einer Tausendstel-Genauigkeit ausgeführt)

11.4 Doppelte Genauigkeit

Bei manchen Operationen kann man in ihre Operanden die Werte mit einer doppelten Genauigkeit - vier Wörter - einlesen. In diesem Fall werden in den Operanden die Werte aus zwei nacheinander gehenden Parametern **op1** und **op1+1**, oder **op2** und **op2+1** angeführt. In diesem Fall wird in dem Steuerparameter die um 50 vergrößerte Parameternummer angegeben.

Wenn die doppelte Genauigkeit bei Eingabe eines Resultates in den Parameter gem. Ziel und Ziel+1 ohne Berücksichtigung des Zahlenwertes gewünscht wird, wird die um 50 vergrößerte Parameternummer für das Resultat eingegeben.

Parameter-Nr.: 0, 1, 2, ..., 31 normale Genauigkeit 32 Bit
50, 52, ..., 81 doppelte Genauigkeit: 64 Bit

Notizbuchspeicher: 32, 33, 34, ..., 49 normale Genauigkeit 32 Bit
50, 52, ..., 81 doppelte Genauigkeit: 64 Bit

Operation	Kode	Beschreibung
Kopieren	0	Kopiert den Parameterwert gem. op1* in den Parameter gem. Ziel* . op1 -Parameterwert bleibt ohne Änderung. op2 - und Z -Werte ohne Einfluß
Addition → Ziel*	1	Addieren der Parameterwerte gem. op1* und op2* und Speicherung des Resultates in den Parameter gem. Ziel*
Subtrahieren → Ziel*	2	Subtrahieren des Parameterwertes gem. op2* von dem Parameterwert gem. op1* und Speicherung des Resultates in den Parameter gem. Ziel* .
Multiplikation → Ziel*	3	Multiplikation des Parameterwertes gem. op1* mit dem Parameterwert gem. op2* und Speicherung des Resultates in den Parameter gem. Ziel* .
Dividierung → Ziel*	4	Dividierung des Parameterwertes gem. op1* durch den Parameterwert gem. op2* und Speicherung des Resultates in den Parameter gem. Ziel* .
Quadratpotenz → Ziel*	5	Quadratpotenz des Parameterwertes gem. op1 und Speicherung des Resultates in den Parameter gem. Ziel* .
Quadratwurzel → Ziel*	6	Quadratwurzel des Parameters gem. op1* und Speicherung des Resultates in den Parameter gem. Ziel*.s
Absolutwert → Ziel*	7	Absolutwert des Parameters gem. op1* und Speicherung des Resultates in den Parameter gem. Ziel* .
Umkehrwert → Ziel*	8	Umkehrwert des Parameterwertes gem. op1* und Speicherung des Resultates in den Parameter gem. Ziel* .
Binärverschiebung → Ziel*	9	Binärverschiebung des Parameterwertes gem. op1 und Speicherung des Resultates in den Parameter gem. Ziel* . Die Verschiebung erfolgt gem. Parameter op2 .. Der op2 -Wert kann ± 32 sein (± 64 für doppelte Genauigkeit)
Kehrwert → Ziel*	10	Kehrwert des Parameters gem. op1* und Speicherung des Resultates in den Parameter gem. Ziel* .
Funktion 2^x → Ziel*	11	Funktion 2^x gem. op1 und Speicherung des Resultates in Ziel . Der Operand op1 kann nicht doppelte Genauigkeit haben und es muss eine reelle Zahl sein. Das Resultat ist eine reelle Zahl, die die doppelte Genauigkeit haben kann.
Funktion $\log_2 x$ → Ziel*	12	Funktion $\log_2 x$ gem. op1 und Speicherung des Resultates in Ziel . Der Operand op1 kann doppelte Genauigkeit haben und es muss eine reelle Zahl sein. Das Resultat ist eine reelle Zahl, die die doppelte Genauigkeit nicht haben kann.
Funktion x^y → Ziel*	13	Potentierung des Wertes gem. op1 zum Wert gem. op2 und Speicherung des Resultates in Ziel . Die Operanden op1 und op2 können nicht die doppelte Genauigkeit haben und es müssen reelle Zahlen sein. Das Resultat ist eine reelle Zahl und kann doppelte Genauigkeit haben.
Funktion ...	14	Funktion $\sqrt{(\text{op1} + \text{op2})}$. Kann für Real- sowie Integralwerte verwendet werden.
Größer als	15	Wenn der Parameterwert gem. op1 größer als der gem. Parameter op2 ist, wird im Parameter gem. Ziel* der Wert 73 , ansonsten 78 gespeichert.
Größer oder gleich	16	Wenn der Parameterwert gem. op1 größer als oder gleich dem Parameter gem. op2 ist, wird im Parameter gem. Ziel* der Wert 73 , ansonsten 78 gespeichert.
Kleiner	17	Wenn der Parameterwert gem. op1 kleiner als der gem. op2 ist, wird im Parameter gem. Ziel* der Wert 73 , ansonsten 78 gespeichert.
Kleiner oder gleich	18	Wenn der Parameterwert gem. op1 kleiner oder gleich dem Parameterwert

		gem. op2 ist, wird im Parameter gem Ziel* der Wert 73 , ansonsten 78 gespeichert.
Vergleich	19	Wenn der Parameterwert gem. op1 gleich dem Parameterwert gem. op2 ist, wird im Parameter gem Ziel* der Wert 73 , ansonsten 78 gespeichert
Bedingung "gleich" → wiederholen	20	Wenn der Parameterwert gem. op1 gleich dem Parameterwert gem. op2 ist, wird die Ausführung der folgenden Parameteroperation fortgesetzt.
Bedingung "nicht gleich" → wiederholen	21	Wenn der Parameterwert gem. op1 nicht gleich dem Parameterwert gem. op2 ist, wird die Ausführung der folgenden Parameteroperation fortgesetzt
Bedingung-Ende → wiederholen	22	Ende der Bedingungen für Ausführung von Parameteroperationen
Konversion int→real	23	Konversion der Integralwerte gem Parameter op1 auf reelle Zahlen gem. Ziel
Konversion real→int	24	Konversion der reellen Zahlen gem Parameter op1 auf Integralwerte gem. Ziel
Wegmessungverschiebung	25	Verschiebung der Wegmessung-Koordinaten gem. op1* (sowie op2*) und Speicherung im Parameter gem. Ziel* . In Operanden op1 (op2) wird die Ordnungs-Nr. der jeweiligen Koordinate (1, 2,...6) eingegeben.
Verschiebung der Koordinaten- Programmwerte	26	Verschiebung des Koordinaten-Programmwertes gem. Koordinatenwert op1* (sowie op2*) und Speicherung im Parameter gem. Ziel* . In Operanden op1 (op2) wird die Ordnungs-Nr. der jeweiligen Koordinate (1, 2,...6) eingegeben
Funktion tan(x)	27	Funktion tan(x) gem. op1 und Speicherung des Resultates in Ziel . Operand op1 kann nicht doppelte Genauigkeit haben und es muss ein Integralwert sein. Resultat ist eine reelle Zahl und kann doppelte Genauigkeit haben
Summe der geometrischen Progression	28	Summe der geometrischen Progression $1+k+k^2+\dots+k^{n-1} = (k^n-1)/(k-1)$ Quozient k ist mit Parameter op1 gegeben. Die Anzahl der n -Elemente ist mit Parameter op2 gegeben. Operanden op1 sowie op2 können keine doppelte Genauigkeit haben und es müssen reelle Zahlen sein. Das Resultat ist eine reelle Zahl, die eine doppelte Genauigkeit haben kann.
Bedingung "größer" → wiederholen	29	Wenn der Parameterwert gem. op1 größer als der Parameterwert gem. op2 ist, wird die Ausführung der folgenden Parameteroperation fortgesetzt
Bedingung "größer oder gleich"	30	Wenn der Parameterwert gem. op1 größer als oder gleich dem Parameterwert gem. op2 ist, wird die Ausführung der folgenden Parameteroperation fortgesetzt
Bedingung "kleiner"	31	Wenn der Parameterwert gem. op1 kleiner als der Parameterwert gem. op2 ist, wird die Ausführung der folgenden Parameteroperation fortgesetzt
Bedingung "kleiner oder gleich"	32	Wenn der Parameterwert gem. op1 kleiner als oder gleich dem Parameterwert gem. op2 ist, wird die Ausführung der folgenden Parameteroperation fortgesetzt
PROGRAMM Programmstart von op1 bis zum Ziel	33	Programmstart von dem Parameter op1 bis (einschl.) Parameter Ziel . Die Parameter können in demselben Block eingegeben werden. Nach Funktion 33 darf keine weitere Parameteroperation programmiert werden.
Parameterverschiebung Verschiebung von op1 bis op2 ins Ziel	34	Verschiebung der Parameter von (einschl.) op1 bis (einschl.) op2 ins Gebiet gem. Ziel
Parameteraustausch Austausch von op1 bis op2 ins Ziel	35	Austausch der Parameter von (einschl.) op1 bis (einschl.) op2 und des Gebietes gem. Ziel

12 PROGRAMMIERUNG UND DEFINIERUNG VON FESTZYKLEN

12.1 Erstellung von Festzyklen

Die Festzyklen sind in der Norm empfohlene Makrozyklen, die Bohr-, Fräs- oder Drehzyklen ausführen. Die Durchführung eines Festzyklus hängt leiglich von Einstellung der Parameter ab, die das Zyklus steuern.

Die Bewegungen innerhalb eines Festzyklus werden in dem Block nach Interpolationen durchgeführt, jedoch vor den Abschlußfunktionen M.

Festzyklen werden im CNC836-Systemspeicher als normale Part-Programme gespeichert und sie können programmiert und editiert werden. Es können auch mehrere Festzyklus-Dateien geschaffen werden und das System liest bei der Wahl des Programmes diejenige Datei ein, deren Name in dem 17. Parameter der Konfigurationsdatei CNC836.KNF angeführt ist. Weitere Bedingung für automatisches Einlesen der Festzyklen ist die Einstellung des Wertes "1" in der 3. Dekade der Maschinenkonstante 99 (s. Bedienungsanleitung, Abschnitt "Maschinenkonstanten" und "Konfiguration").

Die Erstellung und Editierung von Festzyklen kann im CNC836-System auf zweierlei Art und Weise durchgeführt werden. Man kann das herkömmliche Verfahren wie bei Erstellung eines Part-Programms anwenden, dabei muß jedoch während der Editierung von Festzyklen die Einstellung der 3. Dekade der Maschinenkonstante 99 aufgehoben werden. Nach Abschluß der Editierung erfolgt die Syntaxprüfung von Festzyklen, danach muss Maschinenkonstante 99 erneut eingestellt werden. Beim zweiten Verfahren der Editierung von Festzyklen wird der Systemeditor angewendet. In diesem Fall kann die 3. Dekade der Maschinenkonstante 99 auf "1" eingestellt bleiben, es wird jedoch keine Syntaxprüfung der editierten Festzyklen durchgeführt. Diese wird erst bei der Programmwahl erfolgen.

Die Datei mit programmierten Festzyklen beginnt mit dem Anfangswort "**\$DER FESTZYKLUS**" und wird mit "*" Zeichen beendet.

Bei Erstellung von Festzyklen wird höchstens die Parameterarithmetik benutzt. Für Speicherung von Zwischen-ergebnissen der Operationen kann man den Notizbuchspeicher benutzen, um Parameter zu sparen.

Die Anwendung von Parameterfunktion **33** (PROGRAM) erlaubt mehrere Operationen innerhalb eines Blocks, was den Programmablauf wesentlich verkürzt.

Man unterscheidet vier Typen von Parameteroperationen:

- **Verschiebung** (Verschiebungen zwischen einzelnen Parametern, zwischen einem Programmwert und den Parametern und zwischen Wegmessung und Parametern)
- **Operationen** (+, -, *, /, x^2 , sqrt, abs, $1/x$, 2^x , $\log(x)$, x^y , $\sqrt{x^2 + y^2}$, $\tan(x)$, geom.Progression)
- **Bedingter Sprung** (=, <, >, ≤, ≥)
- **Berechnungsverzweigung** (=, <, >, ≤, ≥)
- **Programm** Programmstart

Detaillierte Erklärung der Parameteroperationen ist in Abschnitt "Parameterarithmetik" angeführt. Hier sei nur die Wichtigkeit der bedingten Sprünge und der Verzweigung bei Parameteroperationen anhand von zwei Beispielen gezeigt:

Beispiel für Programmverzweigung:

Falls Parameterwert R30 negativ ist, wird die Inversion von R28 durchgeführt:

Beispiel für bedingte Sprünge:

Falls Parameterwert R11 größer als R30 ist, dann wird zu N20 gesprungen, sonst fortsetzen:

bedingter Sprung zu N20

12.2. Beispiele der Bohr-Festzyklen

Die Bohr-Festzyklen stellen einen speziellen Typ dar, da diese eingebaute Systemfunktionen benutzen. Diese Festzyklen wurden aus älteren Systemversionen übernommen, sie könnten jedoch auch einfacher mittels Parameterarithmetik programmiert werden.

Ein Festzyklus beginnt und endet in der Bezugsebene RA. Die Bohrtiefe reicht bis zur Ebene RB. Die Koordinaten von RA, RB und der Zwischenebene RC müssen in Absolutwerten vorgegeben werden. Das Tiefen-Inkrement q bei tiefen Bohrungen oder Sprünge in Y-Koordinate bei G87 muss als ein Schrittmaß eingegeben werden. Das Ausführungsprogramm von Festzyklen ist in der parametrischen Form gestaltet. In dem Part-Programm, aus dem ein Festzyklus aufgerufen wird, müssen die Ebenenkoordinaten, Tiefen-Inkrement für tiefe Bohrungen, Verweilzeit sowie Vorschub- und Drehzahlwert mit Hilfe von folgenden Parametern programmiert werden:

- R26 Festlegung der Vorschubgeschwindigkeit (F-Adresse) im Rahmen von DER FESTZYKLUS
- R27 Festlegung der Drehzahlen (S-Adresse) für den jeweiligen DER FESTZYKLUS in Übereinstimmung mit dem bereits eingegebenen Funktionen M41 - M44 (Schaltgang)
- R28 Festlegung des q-Inkrementes oder des Sprunges in Y-Koordinate bei G87
- R29 Festlegung der Verweilzeit (Q-Funktion) oder Koordinate der Zwischenebene RC für G87
- R30 Festlegung der Koordinate für RA-Ebene
- R31 Festlegung der Koordinate für RB-Ebene

Alle anderen Funktionen, außer denen, die mit o.a. Parametern eingestellt werden, sind nur für den Ausgangszustand des Festzyklus wirksam, und zwar so, wie sie in dem jeweiligen oder früheren Block eingegeben wurden. Nach Ausführung eines Festzyklus werden alle Funktionen, einschl. derer, die während des Festzyklus geändert wurden, in den ursprünglichen Zustand eingeführt, in dem sie sich am Anfang des Festzyklus befanden.

Anwendungsbeispiel für Festzyklen:

Anmerkung:

Die Festzyklus-Blöcke werden in Absolutwerten programmiert. Funktion G91 (Schrittmaßprogrammierung) ist nur in Blöcken N5 und N6 sie nur für den jeweiligen Block wirksam (nicht jedoch für den Festzyklus).

12.3 Beispiele der Dreh-Festzyklen

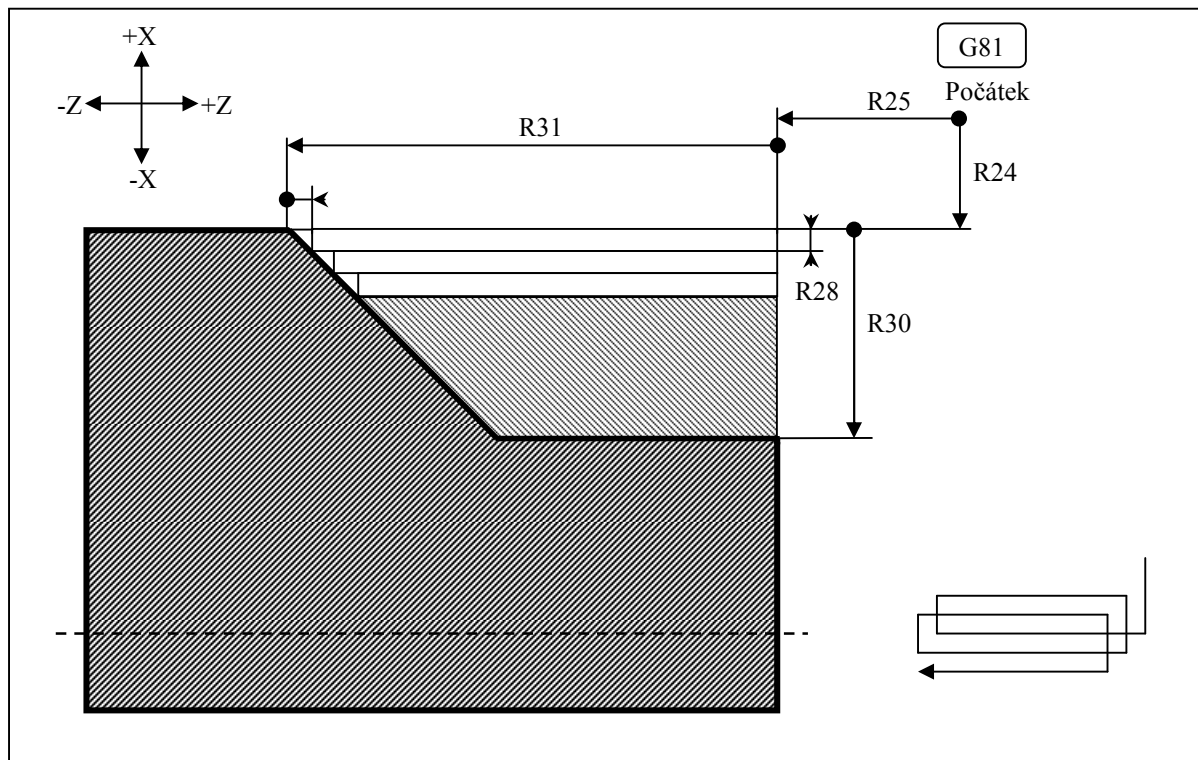
12.3.1 Längsschruppen - G81

Angewandte Parameter:

- R24 Wieviel mm vor dem Werkstück in X-Achse startet der Festzyklus, Vorzeichen bestimmt die Richtung
- R25 Wieviel mm vor dem Stirn in Z-Achse startet der Festzyklus, Vorzeichen bestimmt die Richtung
- R26 Vorschubgeschwindigkeit
- R27 Anzahl der Endbahnläufe (Endbearbeitung)
- R28 Spanstärkte (-Dicke) in mm
- R29 Verkürzung des Abstandes Z auf einen Span, Vorzeichen bestimmt die Richtung
- R30 Größe des Schrappgebietes in X-Achse, Vorzeichen bestimmt die Richtung
- R31 Größe des Schrappgebietes in Z-Achse, Vorzeichen bestimmt die Richtung

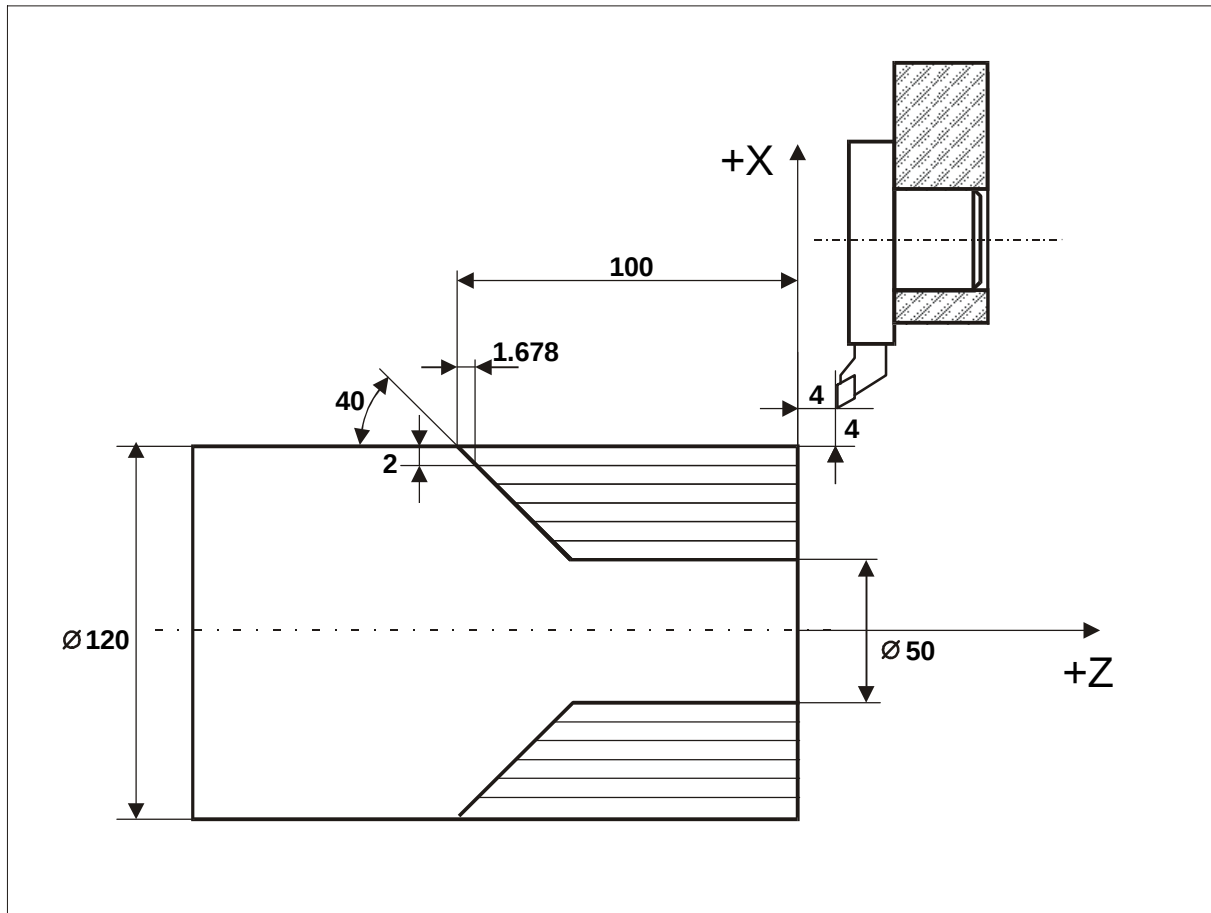
Das Schrappen kann in allen vier Spiegelebenen durchgeführt werden. Die Schrappebenen werden mit dem Vorzeichen in R30 und R31 vorbestimmt. Die Vorzeichen für R24, R25 und R29 sind auch von der jeweiligen Schrappebene abhängig. Man kann sagen, daß alle diesen Parameter (R24, R25, R26, R29, R30, R31) inkremental zum Anfangspunkt programmiert werden (ähnlich wie bei I- und J-Adressen bei Kreisprogrammierung).

Auf dem Bild ist ein Beispiel der Schrappoperation dargestellt, bei dem vorausgesetzt wird, daß R24, R25, R30 und R31 negative Werte haben.



Beispiel:

N10 G54 &1100 D1 T1 M3 S500	"Hlavní věta
N20 X0 Z300	
N30 X128 Z4	" Nájezd do výchozí polohy
N40 R24= -8.000	" Kolik mm před dílcem v X, (znaménko = směr)
R25= -4.000	" Kolik mm před dílcem v Z, (znaménko = směr)
R26=200	" Rychlost posuvu
R27=0.002	" Počet opakovaných přejezdů po konečné draze
R28=4.000	" Tloušťka třísky v mm
R29=1.678	" Zkrácení vzdálenosti Z na jednu třísku (znaménko = " směr)
R30= -70.000	" Velikost hrubování v ose X, (znaménko = směr)
R31= -100.000	" Velikost hrubování v ose Z, (znaménko = směr)
G81	
N50 G80	
N60 M30	



12.3.2 Querschruppen - G82

Angewandte
Parameter:

- R24 Wieviel mm vor dem Werkstück in X-Achse startet der Festzyklus, Vorzeichen bestimmt die Richtung
- R25 Wieviel mm vor dem Stirn in Z-Achse startet Vorzeichen bestimmt die Richtung
- R26 Vorschubgeschwindigkeit
- R27 Anzahl der Endbahnläufe (Endbearbeitung)
- R28 Spanstärke (-Dicke) in mm
- R29 Verkürzung des X-Abstandes auf einen Span, Vorzeichen bestimmt die Richtung
- R30 Größe des Schrappgebietes in X-Achse, Vorzeichen bestimmt die Richtung
- R31 Größe des Schrappgebietes in Z-Achse, Vorzeichen bestimmt die Richtung

Das Schrappen kann in allen vier Spiegelebenen durchgeführt werden. Die Schrappebenen werden mit dem Vorzeichen in R30 und R31 vorbestimmt. Die Vorzeichen für R24, R25 und R29 sind auch von der jeweiligen Schrappebene abhängig. Man kann sagen, daß alle diesen Parameter (R24, R25, R26, R29, R30, R31) inkremental zum Anfangspunkt programmiert werden (ähnlich wie bei I- und J-Adressen bei Kreisprogrammierung).

Technical drawing of a part with a G82 cycle. The part has a rectangular base with a sloped top surface. The drawing shows the part with a hatched area indicating the material to be removed. The G82 cycle is used to drill a hole through the part. The drawing includes a coordinate system with +X, -X, +Z, and -Z axes. The G82 cycle is defined by R25, R24, R31, R28, R30, and R29. The start point is labeled "Počátek".

Angewandte Parameter:

The diagram illustrates a mechanical assembly with a bolt and a nut. The bolt is labeled R31 (rovina RB) and the nut is labeled R30 (rovina RA). A force F is applied to the bolt head, and a distance $G00$ is indicated. A dimension of 1mm is shown for the nut's thickness. A coordinate system with $+X$ and $-Z$ axes is shown.

Auf dem Bild ist das Beispiel einer tiefen Bohrung dargestellt

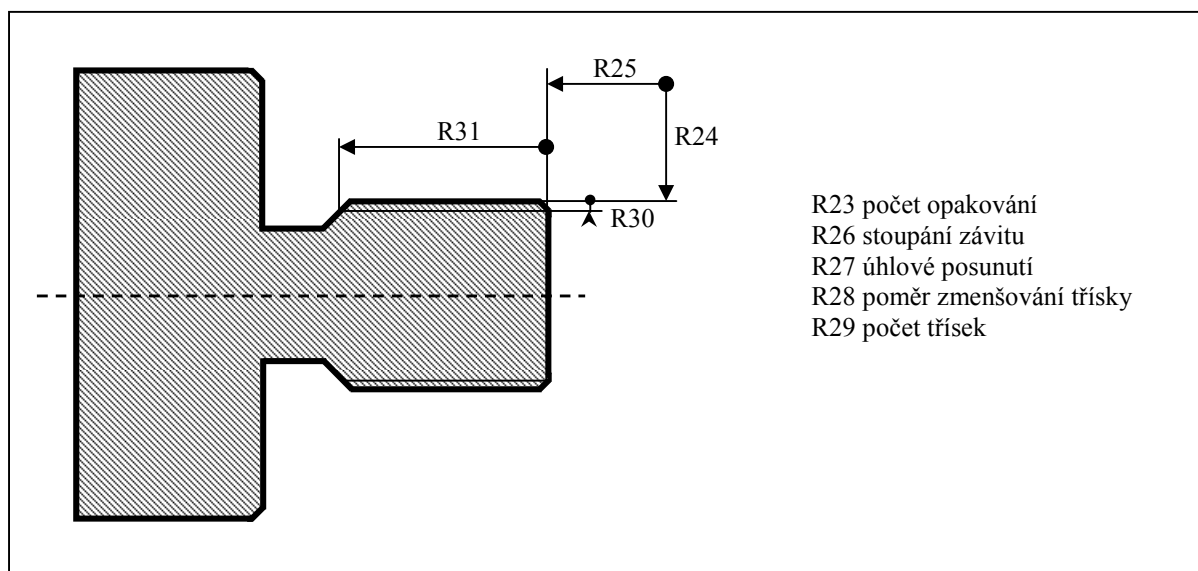
12.3.4 Gewindeschneiden auf einer zylindrischen Oberfläche G84

Angewandte

Parameter:

R23 Anzahl der Endbahnläufe

- R24 Wieviel mm vor dem Werkstück in X-Achse startet der Festzyklus, Vorzeichen bestimmt die Richtung
- R25 Wieviel mm vor dem Stirn in Z-Achse startet der Festzyklus, Vorzeichen bestimmt die Richtung
- R26 Gewindesteigung
- R27 Winkelverschiebung am Anfang der Gewindeschneidoperation
- R28 Abnahmeverhältnis zwischen dem gerade gedrehten und dem nächsten Span beim Gewindeschneiden (Quozient < 1)
- R29 Anzahl der Späne, mit denen das Gewinde geschnitten werden soll
- R30 Gewindetiefe in Achse X, das Vorzeichen bestimmt die Richtung
- R31 Gewindelänge in Achse Z, das Vorzeichen bestimmt die Richtung



Parameter R28 (Quozient) gibt das Verkleinerungsverhältnis zwischen dem gerade gedrehten und dem nächsten Span beim Gewindeschneiden an. Die Späne werden beim Gewindeschneiden immer kleiner (diese Verkleinerung bestimmt diese Konstante) und bilden geometrische Progression mit dem Quozienten kleiner als 1. Z.B. bei einer Spanstärke von 0,5 mm und dem Quozienten 0,8 bekommt man folgende Spanstärkenwerte:

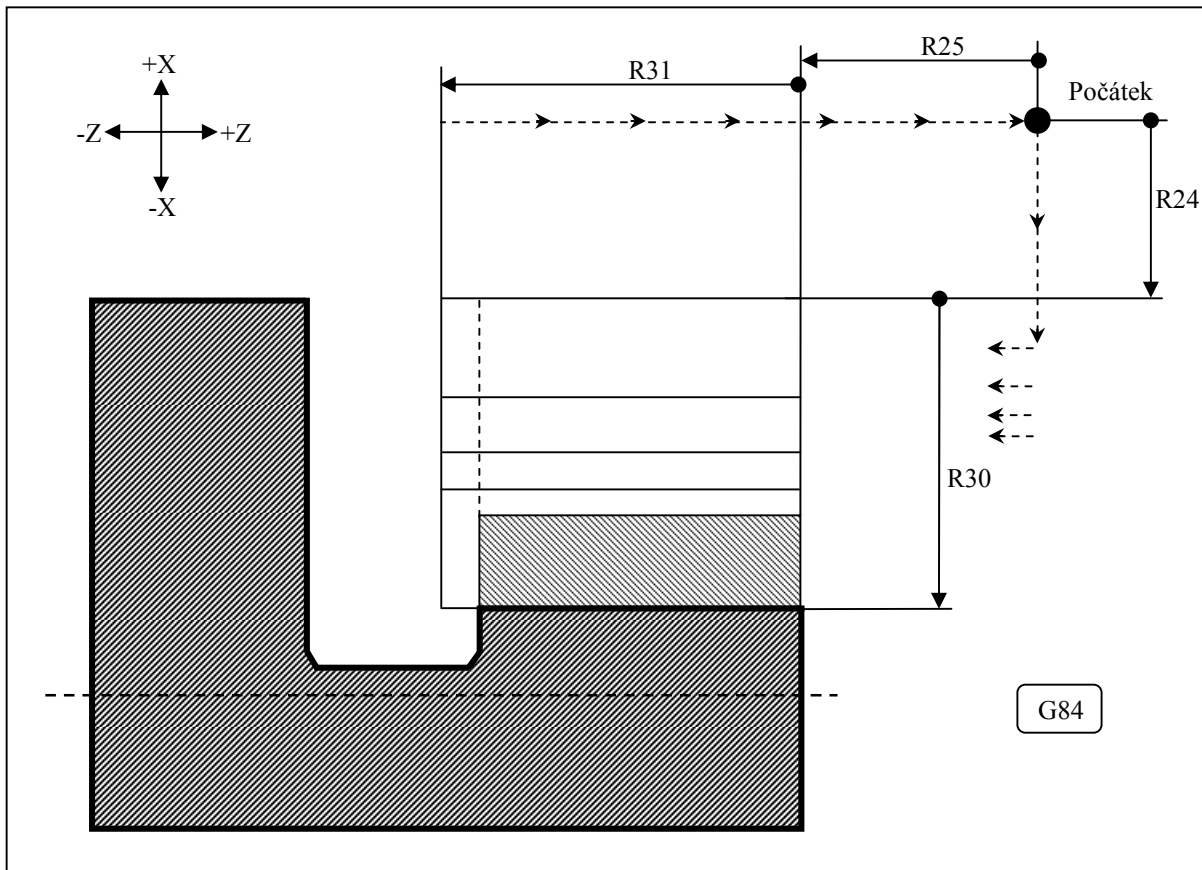
$$0,50 \cdot 0,8 = 0,4 \text{ mm}$$

$$0,40 \cdot 0,8 = 0,32 \text{ mm}$$

$$0,32 \cdot 0,8 = 0,256 \text{ mm}$$

Das Gewindeschneiden kann in allen vier Spiegelebenen durchgeführt werden. Die Gewindeschneiden-Ebenen werden mit dem Vorzeichen in R30 und R31 vorbestimmt. Die Vorzeichen für R24, R25 sind auch von der jeweiligen Ebene abhängig. Man kann sagen, daß alle diesen Parameter (R24, R25, R30, R31) inkremental zum Anfangspunkt programmiert sind (ähnlich wie bei I- und J-Adressen bei Kreisprogrammierung).

- R23 Anzahl der Wiederholungen
- R26 Gewindesteigung
- R27 Winkelverschiebung
- R28 Spanverkleinerungsverhältnis
- R29 Anzahl der Späne



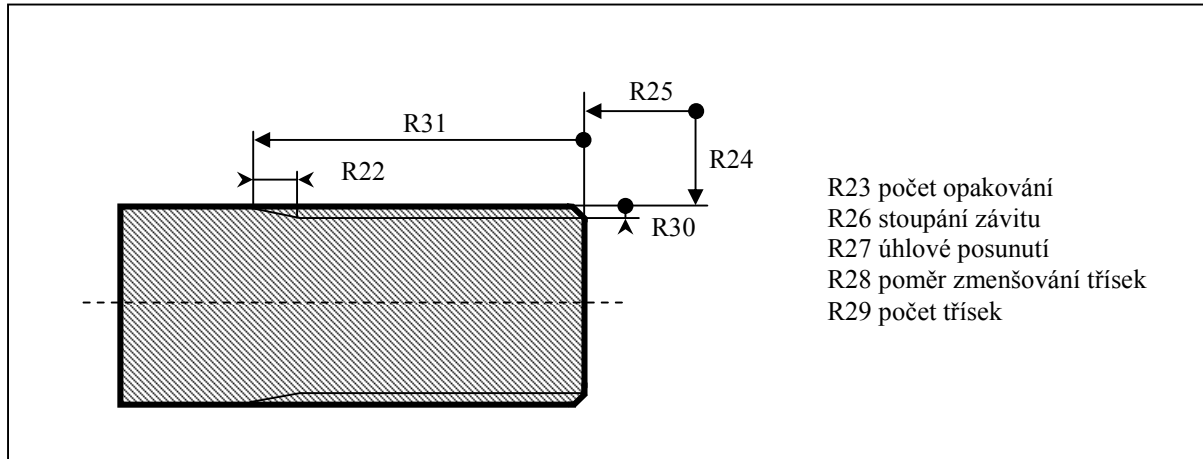
Auf dem Bild ist ein Beispiel für Gewindeschneiden dargestellt, bei dem vorausgesetzt wird, daß R24, R25, R30 und R31 negative Werte haben.

12.3.5 Gewindeschneiden auf einer zylindrischen Oberfläche mit Auslauf - G85

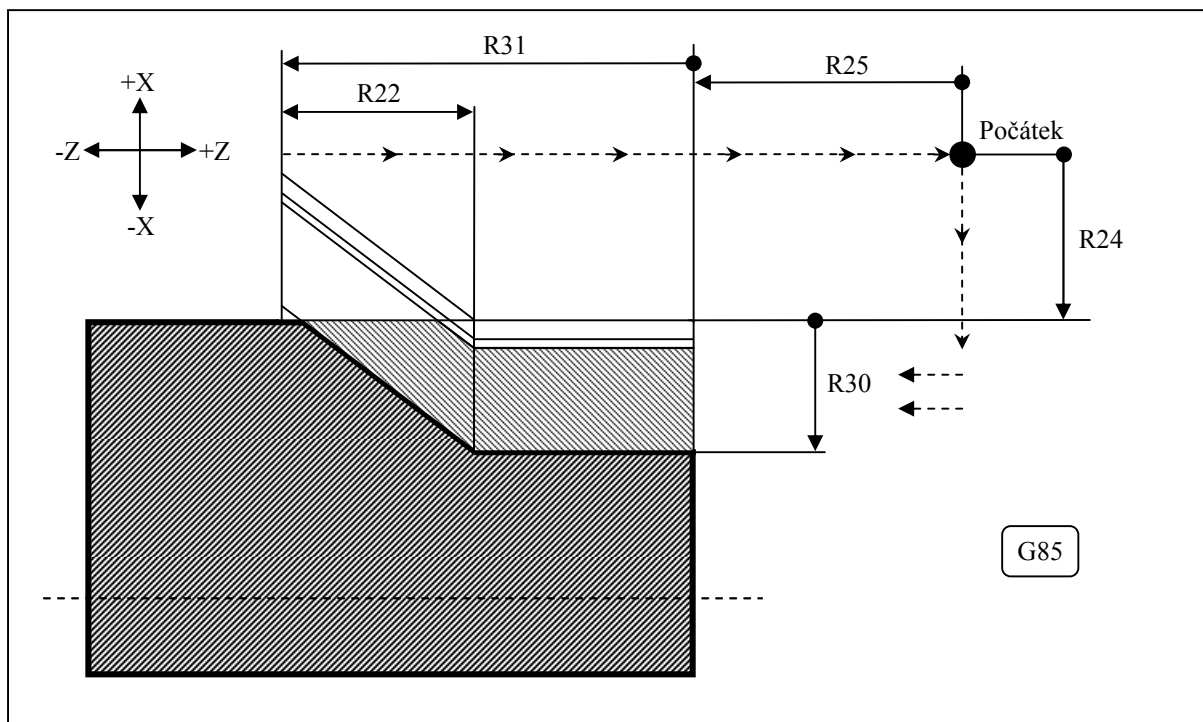
Angewandte Parameter:

- R22 Auslauflänge
- R23 Anzahl der Endbahnläufe
- R24 Wieviel mm vor dem Werkstück in X-Achse startet der Festzyklus, Vorzeichen bestimmt die Richtung
- R25 Wieviel mm vor dem Stirn in Z-Achse startet der Festzyklus, Vorzeichen bestimmt die Richtung
- R26 Gewindesteigung
- R27 Winkelverschiebung am Anfang der Gewindeschneidoperation
- R28 Abnahmeverhältnis zwischen dem gerade gedrehten und dem nächsten Span beim Gewindeschneiden
- R29 Anzahl der Späne, mit denen das Gewinde geschnitten werden soll
- R30 Gewindetiefe in Achse X, das Vorzeichen bestimmt die Richtung
- R31 Gewindelänge in Achse Z, das Vorzeichen bestimmt die Richtung

Parameter R28 (Quozient) gibt das Verkleinerungsverhältnis zwischen dem gerade gedrehten und dem nächsten Span beim Gewindeschneiden an. Beim Gewindeschneiden werden die Späne immer kleiner (diese Verkleinerung bestimmt diese Konstante) und bilden geometrische Progression mit dem Quozienten kleiner als 1 (ähnlich wie in G84).



Parameter R22 gibt die Auslauflänge (positiver Wert) an. Die Gewinde-Gesamtlänge ist in R33 (einschl. Auslauflänge) vorgegeben. Der Auslaufwinkel ist durch die Maschinenkonstante 08 des Rekonfigurators (einschl. Vorzeichen) vorbestimmt. Die Auslauflänge muss so bestimmt werden, daß beim letzten Lauf das Werkzeug über das Material ausläuft.



Das Gewindeschneiden kann in allen vier Spiegelebenen durchgeführt werden. Die Gewindeschneidebenen werden mit dem Vorzeichen in R30 und R31 vorbestimmt. Die Vorzeichen für R24, R25 sind auch von der jeweiligen Ebene abhängig. Man kann sagen, daß alle diesen Parameter (R24, R25, R30, R31) inkremental zum Anfangspunkt programmiert sind (ähnlich wie bei I- und J-Adressen bei Kreisprogrammierung).

....

R23 Anzahl der Wiederholungen
R26 Gewindesteigung
R27 Winkelverschiebung
R28 Spanverkleinerungsverhältnis
R29 Anzahl der Späne

Auf dem Bild ist ein Beispiel für Gewindeschneiden mit Auslauf dargestellt, bei dem vorausgesetzt wird, daß R24, R25, R30 und R31 negative Werte haben.

ANLAGEN

ANLAGE A - Werkzeugkorrektur in CNC836/846- Steuersystem

Anm. Gilt für Systemversionen nach 20.10.1997

Die Werkzeugkorrektur erlaubt die Erstellung eines allgemeinen Part-Programms, das für verschiedene Längen und Durchmesser von Werkzeugen (Fräsen, Drehmeißel usw.) anwendbar ist. Beim Einsatz eines Werkzeuges mit anderen Parametern wird nur die jeweilige Korrektur durchgeführt und das Part-Programm bleibt ungeändert.

Man unterscheidet zwei Korrekturtypen:

Werkzeugradius-Korrektur - wird durch G-Funktionen der 3. Gruppe (G41, G42 und G43) vorgegeben

Werkzeuglänge-Korrektur - wird durch &-Adresse vorgegeben

Anlage A1 - Datei TAB0.KOR und gespeicherte Korrekturtabelle

Anm. Dateiname, bzw. die Nummer nach TAB ist nicht verbindlich. Man kann mehrere Tabellen verwenden. In dieser Programmieranleitung wird die Bezeichnung TAB0.KOR verwendet.

Die Korrekturwerte (Länge- und Radiuskorrektur) werden in der Datei TAB0.KOR. gespeichert. Das Steuersystem arbeitet mit der Kopie dieser Datei, die sich in der Korrekturtabelle **im internen Systemspeicher befindet**. Die Datei TAB0.KOR wird in diese Tabelle automatisch nach jeder Systemaktivierung und auch nach jeder Editierung dieser Datei überschrieben, falls nach deren Abschluß die neue Dateiversion gespeichert wird (s. Bedienungsanleitung CNC836).

Die Struktur der Korrekturtablelle im internen Speicher:

Korrekturs-Nr. I.Angabe II.Angabe III.Angabe IV.Angabe V.Angabe

1	Radius-	Längenkorrr.	Längenkorrr.	Längenkorrr.	Längenkorrr.
		korrektur f. 1.Achse	f. 2.Achse	f. 3.Achse	f. 4.Achse

2	Radius-	Längen X	Längen Y	Längen Z	Längen 4.
	korrektur				

usw. bis 99

Die Korrekturabelle hat insgesamt 99 Positionen. Das heißt, daß im System 99 verschiedene Länge- und Radiuskorrekturen verwendet werden können. Jede Tabellenposition besteht aus insgesamt 5 Angaben. Die erste Angabe enthält den Radiuskorrekturwert. Weitere vier Angaben beziehen sich zur Längenkorrektur in der ersten, zweiten, dritten und vierten Achse. Die Längenkorrektur kann nicht für eventuelle fünfte und sechste Achse angewendet werden.

Die in TAB0.KOR gespeicherte Korrekturen haben dieselbe Struktur und sind in dieser Datei in nachstehend beschriebenem Format gespeichert (Anm.: Die Datei beginnt mit dem Schlüsselwort \$KOR, davor kann ein Kommentar nach Wunsch geschrieben werden).

Die Zeile beginnt mit einer zweistelligen Korrekturnummer (auf der ersten Stelle kann eine Null eingegeben werden) mit Doppelpunkt. Die Radiuskorrekturwerte werden mit "R=" eingeleitet, die Längenkorrekturwerte werden mit Koordinaten oder deren Ordnungs-Nr. eingeleitet.

Anm.

Die Bezeichnung von Koordinaten wird dann verwendet, wenn sie in der Reihenfolge 1. Koordinate - Achse X, 2. Koordinate - Achse Y, 3. Koordinate - Achse Z und 4. Koordinate - Achse U stehen. Die Reihenfolge und Bezeichnung der Koordinaten ist in Maschinenkonstanten Nr. 0 bis 5 in TAB0.REK vorgegeben. Bei Drehautomaten tragen in der Regel die Achsen folgende Bezeichnung: X die 1. Achse, Z die 2. Achse. In diesem Fall muß in der Korrekturdatei für die Z-Achse anstatt "Z" Ziffer "2" (zweite Koordinate) eingegeben werden. Für Achse X kann man X (oder auch 1) eingeben.

Im folgenden Beispiel einer TAB0.KOR-Datei werden diverse Schreibweisen angeführt. Die empfohlene Schreibweise in TAB0.KOR für Dreiachsen-Fräsmaschinen (X,Y,Z) ist bei Korrektur Nummer 1 angegeben, die empfohlene Schreibweise für Drehautomaten (X,Z) ist bei Korrektur Nummer 2 angegeben. Die empfohlene Schreibweise für Vierachsen-Maschinen (X,Y,Z,4) ist bei Korrektur Nummer 3 angeführt.

Bei Korrektur Nr. 25 ist keine Radiuskorrektur vorgegeben, deshalb steht in der Tabelle für diese Korrektur eine Null. Bei Korrektur Nr. 95 ist die Schreibweise mit Ordnungszahlen der einzelnen Koordinaten vorgesehen.

Falls keine Korrektur angeführt ist, steht in der Tabelle (im Systemspeicher) eine Null. In dem angeführten Beispiel werden Nullwerte für Radius- und Längenkorrektur in allen hier nicht angeführten Positionen stehen, d.h. alle Korrekturen mit Ausnahme von Korrekturen Nr. 1, 2, 3, 25 und 95. Die für einzelne Achsen nicht angeführten Korrekturen sind gleich Null.

Musterdatei TAB0.KOR für Drehautomaten enthält 99 Positionen mit Nullwert-Korrekturen im Format:

Anm. 1:

Falls bei einem Drehautomaten die Radiuskorrektur verwendet wird, kann in jeder Position noch der verwendete Werkzeugtyp in folgendem Format stehen:

Die Nummer aus P= muss im Bereich 1 bis 9 liegen. Andere Werte sind nicht zulässig. Die Anwendung von Werkzeugtypen bei Drehautomaten wird weiter im Text beschrieben.

Anm.2:

Wenn nicht alle 99 Korrekturen eingesetzt werden, könnten die Dateien wegen einer besseren Übersicht verkürzt werden. Die Bezeichnung TAB0.KOR ist nicht verbindlich, für Drehautomaten kann im System die Bezeichnung TAB2.KORR stehen. Wichtig ist, daß dieselbe Bezeichnung auch in der Konfigurationsdatei CNC836.KNF in Parameter 21 angeführt ist. Im System können zwar verschiedene Korrekturdateien verwendet werden können, wir empfehlen jedoch wegen Eindeutigkeit nur eine Bezeichnung zu verwenden (und zu speichern).

Eingabe in die Korrekturtabelle

Bei der Eingabe in die Korrekturtabelle muß man die Eingabe in TAB0.KOR-Datei und in die Tabellenkopie in dem internen Systemspeicher unterscheiden.

Die Eingabe in die Korrekturtabelle im internen Systemspeicher erfolgt durch:

- a) Editierung der TAB0.KOR-Datei und **nachfolgender Datenspeicherung** - die Editorbeschreibung s. Bedienungsanleitung. Die geänderten Korrekturwerte bleiben in der Datei dauernd gespeichert. Nach jeder Systemaktivierung werden diese Werte in den internen Systemspeicher eingelesen.
- b) Online-Eingabe im manuellen Modus durch Betätigung der R- bzw. D- oder &-Taste - die Beschreibung s. Bedienungsanleitung. Die online-eingegebenen Korrekturwerte werden in die TAB0.KOR-Datei und in den

internen Systemspeicher geschrieben. Nach jeder Systemaktivierung werden diese Werte in den internen Systemspeicher eingelesen.

c) Eingabe aus dem Part-Programm durch Anwahl von G92 - Beschreibung s. Programmieranleitung.

ACHTUNG: Auf diese Weise wird nur die Eingabe der Werte in die Tabelle im internen Systemspeicher durchgeführt, die Werte in TAB0.KOR werden nicht geändert. Das heißt, dass die in die Systemtabelle im Part-Programm mittels G92 geschriebenen Korrekturwerte nur solange gültig sind, bis die Speicherung der TAB0.KOR-Datei nach deren Editierung erfolgt oder das System deaktiviert wird. Nach neuer Aktivierung sind bereits die in TAB0.KOR angeführten Werte wirksam. Die Eingabe in die Korrekturtabelle wird direkt im Part-Programm durchgeführt. Damit ist gesichert, dass für jeweiliges Part-Programm immer die darin eingegebenen Korrekturwerte wirksam sind. Jegliche Änderungen dieser Werte müssen jedoch in diesem Fall im Part-Programm, nicht jedoch in der Datei TAB0.KOR durchgeführt werden!

Für die Eingabe in die Korrekturtabelle mittels G92 werden die Parameter verwendet, deren Werte der jeweiligen Position der Korrekturtabelle auf folgende Art und Weise zugeordnet werden:

R0 ist für die Eingabe der Radiuskorrektur vorgesehen

R1 ist für die Eingabe der Längenkorrektur in der 1. Achse (meistens X) vorgesehen

R2 ist für die Eingabe der Längenkorrektur in der 2. Achse (meistens Y oder Z bei Drehautomaten)) vorgesehen

R3 ist für die Eingabe der Längenkorrektur in der 3. Achse (meistens Z) vorgesehen

R4 ist für die Eingabe der Längenkorrektur in der 4. Achse vorgesehen

In G92 wird bestimmt, dass falls der jeweilige Parameter in einem Block programmiert ist, wird sein Wert in diejenige Position der Korrekturtabelle geschrieben, die unter der D-Adresse angeführt ist.

Beispiel:

Man will folgende Werte in Position Nr. 12 der Korrekturtabelle eingeben:

Radiuskorrektur -10.0mm, Längenkorrektur von 25.5mm in Achse Y und -5.0mm in Achse Z. Format des Part-Programmblöcke:

Wenn einer der Parameter im Block nicht angeführt ist, wird der Wert in der jeweiligen Tabellenposition nicht geändert. In unserem Fall wird die Längenkorrektur in Achse X und 4 in der Position 12 nicht geändert, d.h. der ursprüngliche Wert bleibt erhalten.

Anlage A2 - Radiuskorrektur mit Äquidistante

Wenn Maschinenkonstante Nr. 95, 8.Dekade auf 1 eingestellt ist kann man eine neue Lösung für Radiuskorrekturen anwenden. Man kann (mit Ausnahmen, die im weiteren beschrieben sind) die Werkstückkontur (Zeichnungswerte) programmieren. Die Korrekturbahn (die Bahn der Werkzeugmitte bei Fräsmaschinen oder die Bahn des Schneideradius bei Drehautomaten) liegt auf der Äquidistante. Wenn zwei nacheinander folgende Blöcke nicht verknüpft sind, fährt die Werkzeugmitte in den Äquidistanzen-Schnittpunkt.

Die Radiuskorrektur wird am Ende des Blocks wirksam, in dem sie mittels G41 oder G42 aufgerufen wurde, d.h. sie gilt ab nächstem Block. Die Aufrufung (Aktivierung von G41 und G42) und Widerrufung (Deaktivierung mittels G40) der Radiuskorrektur ist nur im Block mit Linearinterpolation möglich. Bei der Kreisinterpolation kann keine Radiuskorrektur aktiviert oder deaktiviert werden.

Für Programmierung der Radiuskorrektur G41 und G42 sind folgende Regeln zu berücksichtigen:

Die Korrektur kann nicht im Block ohne Bewegung in der Korrekturebene programmiert werden, d. h. in allen Blöcken, wo G41/G42 aktiviert ist, muss eine Bewegung wenigstens in einer Achse der Korrekturebene programmiert werden. Somit bleibt ist die Korrektur auf Abrufung von Unterprogrammen oder Festzyklen eingeschränkt, weil der Einführungsblock eines Unterprogramms oder Festzyklus ein festgelegtes Format (Nxxxx G79 Lyyy) hat, das für einen Block ohne Bewegung charakteristisch ist. Vor der Abrufung des Programms müßte also die Korrektur widerrufen werden und erst in dem Unterprogramm erneut programmiert werden. (Anm.: Diese Einschränkung wird ab der nächsten Programmversion beseitigt). Die Korrektur darf auch in keinem Block stehen, wo nur eine der M-Funktionen programmiert wurde, ohne gleichzeitig eine Bewegung zu programmieren.

Anlage A3 - Aktivierung der Radiuskorrektur

Bei Aktivierung der Radiuskorrektur ist folgende Faustregel streng zu beachten:

Die Bahn der Werkzeugmitte bewegt sich aus dem Anfangspunkt im Block N auf der Linie, die senkrecht zur Bahn des Blocks N+1 verläuft, d.h. Block N+1 wird bereits mit der aktivierten Korrektur ausgeführt. Auf Bild 1 ist die im Block N10 programmierte Radiuskorrektur G42 dargestellt, die Korrekturbahn ist mit einer Strichlinie von A zum B dargestellt. Anfahrspunkt B liegt auf der Linie, die senkrecht zur Bahn gem. N20 verläuft.

PROGRAMMIERUNG EINER RADIUSKORREKTUR IM BLOCK N10

PROGRAMMIERTE BAHN KORREKTURBAHN

Bild 1

Auf die gleiche Weise wird der Einsatzpunkt der Korrektur ermittelt, wenn ein Kreis folgt, was auf demselben Bild dargestellt ist. Punkt B liegt auch auf einer Linie, die senkrecht zur Tangente des programmierten Kreises verläuft. Bei Einschaltung der Korrektur ist eine richtige Anfahrrichtung zu wählen.

Auf Bild 2 ist ein falscher Anfahrwinkel dargestellt. Diese Richtung der Korrekturanfahrt würde fehlerhafte Eckbearbeitung zur Folge haben, weil die Werkzeugkontur das Werkstück schneidet, bevor das Werkzeug an die Senkrechte zum nächsten Block kommt. Auf Bild 2 rechts ist der maximale Anfahrwinkel dargestellt. Falls die Korrektur noch vor dem Werkstück eingesetzt wird, was in der Praxis ziemlich oft der Fall ist, hat der Anfahrwinkel keine Bedeutung.

FEHLERHAFTER KORREKTUR-ANFAHRWINKEL

Bild 2

Dieser Fall ist unter Block-Nr. N15 auf dem Bild links dargestellt.

Wenn die Korrektur programmiert wurde, bewegt sich die Werkzeugmitte auf den Äquidistanten. Auf Bildern 3 und 4 sind Beispiele einer Korrekturbahn für den Fall gerade Linie - gerade Linie, gerade Linie - Kreis und Kreis - Kreis dargestellt, und zwar sowohl für Links- als auch Rechtskorrektur.

BEWEGUNG IN KORREKTUREBENE BEWEGUNG IN KORREKTUREBENE GERADE LINIE - GERADE LINIE GERADE LINIE - BOGEN

Bild 3

BEWEGUNG IN KORREKTUREBENE BEWEGUNG IN KORREKTUREBENE BOGEN - GERADE LINIE BOGEN - BOGEN

DARSTELLUNG EINER KORREKTURBAHN DARSTELLUNG EINER KORREKTURBAHN
 NACH LINKS UND NACH RECHTS NACH LINKS UND NACH RECHTS
 (C42 C43) (C42 C41)

Bild 4

Bei der Bahnprogrammierung muss immer darauf geachtet werden, das die Äquidistanten immer einen Kreuzpunkt haben. Wenn die Äquidistanten von zwei nacheinander folgenden Blöcken keinen Kreuzpunkt haben, wird Fehlermeldung "ÄQUIDISTANTEN-KREUZPUNKT NICHT GEFUNDEN" angezeigt. Dies kommt bei einer unrichtigen Kombination des Korrekturwertes und Tangentenwinkels in dem Kreuzpunkt der Bahnen von zwei nacheinander folgenden Blöcken vor.
 In manchen Fällen ist zwar der Kreuzpunkt der Äquidistanten gefunden, er liegt jedoch in ziemlich großer Entfernung. Diese zwei Fälle

KEIN KREUZPUNKT BEWEGUNG IN KORREKTUREBENE
 DER ÄQUIDISTANTEN KREIS - KREIS

ÄQUIDISTANTEN-KREUZPUNKT EXISTIERT

ÄQUIDISTANTEN-KREUZPUNKT
 ENTFERNT

KEIN ÄQUIDISTANTEN-KREUZPUNKT

Bild 5

sind auf Bild 5 dargestellt. In der Regel handelt es sich um technologisch nicht geeignete Fälle, die in der Praxis nicht vorkommen. Sollte sie doch mal auftreten, muß man einen besseren Programmierungsprozeß wählen. Die Widerrufung der Radiuskorrektur erfolgt mit Hilfe von G40 und ist - genauso wie die Programmierung von G41 und G42 - nur bei einer Linearinterpolation möglich. Der Korrektur-Endpunkt des letzten Bocks vor G40 liegt auf der Linie, die senkrecht zur Tangentiale ist, die durch ?? verläuft. Die Widerrufung einer Korrektur ist auf Bild 6 dargestellt. Ähnlich wie beim

Radiuskorrektur-Widerrufung Radiuskorrektur-Widerrufung

FEHLERHAFTE WIDERRUFUNG FEHLERHAFTE WIDERRUFUNG
 EINER RADIUSKORREKTUR EINER RADIUSKORREKTUR

Bild 6

Einsatz einer Korrektur ist auf die Blockrichtung zu achten, in dem G40 programmiert wurde. Dies gilt nicht für eine Korrektur, die bereits außerhalb eines Produktes widerrufen wird.

Anlage A4 - Radiuskorrektur für Drehautomaten

Bei Anwendung der Radiuskorrektur kann die Werkstückkontur (Zeichnungswerte) programmiert werden. Bei Werkzeugänderung (Schneideradius) ändert sich nur der Tabellenwert der Radiuskorrektur.

P - THEORETISCHE MEIßELSPITZE

R - SCHNEIDERADIUS
S - SCHNEIDERADIUS-MITTE

STELLUNGEN FÜR ERMITTLUNG
DER LÄNGE-KORREKTUR
IN ACHSE X UND Z

REFERENZSTELLUNG
DER WERKZEUGNULL

Bild 7

Bei Eingabe der Werkzeuglänge-Korrektur wird dieser Korrekturwert zum theoretischen Punkt P der Schneide bezogen. Dieser Punkt liegt (in der Regel) nicht in der Werkzeugschneidemitte, sondern in dem Kreuzpunkt der Tangentialen und des Schneideradius, wie aus dem Bild 7 ersichtlich ist.

Um eine richtige Berechnung der Äquidistanten-Verfahrbahn zu gewährleisten, muss außer Schneideradius auch dessen Lage eingegeben werden. Für die Berechnung der Radiuskorrektur werden dann vom System außer dem Werkzeugradius (Schneideradius) auch die Information über Werkzeugeinspannung verwendet. Die in Frage kommenden Meißelstellungen sind auf Bild 8 dargestellt.

Bild 8

Anm.:

Beide Abbildungen sind identisch und nur um Achse Z spiegelartig gedreht. Für die Bestimmung der Werkzeugstellung wird die Darstellung gewählt, in der die Achsenrichtung X mit dem jeweiligen Drehautomaten übereinstimmt. Die Vorzeichen für einzelne Schneidepositionen sind in der Konfigurationsdatei CNC836.KNF geschrieben, und zwar unter \$54 (für die erste Werkzeugachse) und \$52 (für eventuelle zweite Werkzeugachse). Wenn der Drehautomat nur einen Werkzeugkopf besitzt, spielt der Wert in \$52 keine Rolle (es werden hier z.B. die gleichen Werte wie für den ersten Werkzeugkopf angegeben). Die Vorzeichen bestimmen die Richtung der additiven Verschiebung der theoretischen Schneidespitze im Vergleich mit der Schneideradius-Mitte.

Der Einfluss dieser additiven Verschiebung, die auch die angewendete Werkzeugstellung berücksichtigt, ist auf Bild 9 dargestellt. Auf Abb. 9A ist die Bahn ohne Radiuskorrektur dargestellt. Die theoretische Werkzeugspitze bewegt sich auf der von dem Technologen programmierten Bahn - in diesem Fall ist die Bahn auf die Werkzeugkante programmiert. Auf Abb. 9B ist die Radiuskorrektur nach links G41 dargestellt, in der Korrekturtabelle ist jedoch kein Werkzeugtyp eingegeben (keine P-Eingabe) oder P=9. Die theoretische Werkzeugspitze P9 liegt in der Mitte der Schneide,

OHNE RADIUSKORREKTUR

MIT RADIUSKORREKTUR

OHNE ADDITIVE VERSCHIEBUNG

MIT RADIUSKORREKTUR
MIT ADDITIVER VERSCHIEBUNG

Bild 9

deshalb ist die additive Verschiebung in beiden Richtungen gleich Null.

Wie aus dem Bild ersichtlich ist, wird zwar die Korrektur G41 gesetzt (die Spitze bewegt sich von V1 zum V3), da jedoch Endpunkt V3 nicht um die additive Verschiebung korrigiert ist, würde das Werkzeug außerhalb des Werkstückes laufen. Auf Abb. 9C ist die richtige Anwendung einer Radiuskorrektur dargestellt. In der

Korrekturtabelle muß bei der jeweiligen Radiuskorrektur auch das Werkzeugtyp angegeben werden (in unserem Fall P=3), das gem. Bild 8 negative additive Verschiebung in Achsen X und Z hat. Die theoretische Spitze V3 auf Abb. 9B bewegt sich in die Stellung V3 auf Abb. 9C, was eine richtige Schneidstellung zum Werkstück ist. Die Größe dieser Bewegung ist gleich dem Radiuswert (unter Berücksichtigung der Minus-Vorzeichen). Die Bahn der theoretischen Spitze ist aus der Abb. 9c ersichtlich und mit einer Strichlinie dargestellt.

Anlage A5 - Anzeige an einem Drehautomaten bei Verwendung der Radiuskorrektur

Die Anzeige der Stellung im System schließt additive Verschiebung in sich. Nachfolgendes Beispiel eines Part-Programmeiles zeigt, wie das System die Werte in einzelnen Blöcken anzeigt.

Der Inhalt der Datei TAB0.KOR mit Korrekturen wird folgendes Format haben:

In der Datei ist für die Korrektur Nummer D1 die Radiuskorrektur von 0,8mm eingegeben. Es wird angenommen, daß die Werkzeuglänge-Korrektur für Werkzeug Nr. T1 eingestellt ist, Schneidstellung gem. Bild 8 ist P=3.

Nach rechts Z +

Nach oben X +

Format des Part-Programmabschnittes (es wird die Durchmesserprogrammierung in Achse X angenommen):

WAHL DES WERKZEUGES UND DER KORREKTURTABELLE
FAHRT IN DIE AUSGANGSSTELLUNG
AKTIVIERUNG DER KORREKTUR

LETZTER BLOCK MIT KORREKTUR
DEAKTIVIERUNG DER KORREKTUR

Bild 10

Auf Bild 10 ist die Bahn der Meißelschneide gem. o.a. Part-Programm dargestellt (Um bessere Anschaulichkeit zu erreichen, ist der Schneideradius mehrmals vergrößert).

Anzeige am Ende einzelner Blöcke ist aus der Tabelle ersichtlich:

BLOCKENDE THEORETISCHE ANZEIGE X ANZEIGE Z
SPITZE IM PUNKT (DURCHM.)

Im Block N70 (der letzte vor der Korrekturdeaktivierung) liegt die Schneidemitte an der Linie, die senkrecht zum Endpunkt verläuft, die theoretische Spitze liegt um einen Radiuswert näher zur Achse. Wert 48.8 ist Durchmesser 50 minus 2x Schneideradius 0,8 (Durchmesseranzeige!).

Anm.:

Bei graphischer Darstellung der Bahn wird die theoretische Spitze der Meißelschneide gezeichnet. Mit Aufnahme von N20 und N50 (auf Bild 10 mit einer Strichlinie zwischen A-B und D-E gezeichnet), verläuft die Korrekturbahn identisch mit der Programmbahn.

Anlage B - kontinuierliche Blockverknüpfung

CNC836 mit Prozessor CPU04 in der Kassette (80486) gestattet eine kontinuierliche Bearbeitung von komplizierten Formen, die z.B. mit einer Anzahl von kleinen Blöcken programmiert wurden. Unter "Kontinuierlichkeit" versteht man Fähigkeit des Steuersystems, zum nachfolgenden Bewegungsblock zu übergehen, ohne die Bearbeitungsgeschwindigkeit dabei zu reduzieren. Für einen kontinuierlichen Übergang von einem Block zum anderen müssen diese nicht unbedingt aufeinander tangential angeknüpft sein (die Bewegungsrichtung beider Blöcke muss nicht identisch sein). Im weiteren werden alle Varianten dieser Problematik behandelt.

Anlage B1 - Nichtkontinuierliche Blockverknüpfung

Die Standardverknüpfung wird mit G98 angewählt. Es handelt sich um eine Default-Einstellung nach Aktivierung des Steuersystems. In diesem Fall ist die Bewegung mit einer Rampe eingeleitet und abgeschlossen. Die Steilheit der An- und Ausfahrrampe ist mit der Maschinenkonstante R52 vorgegeben. Zwischen der Kassette und der Betätigungstafel verläuft serielle Kommunikation mit Geschwindigkeit von 115200 Bd. Die Zeitverzögerung bei dieser Kommunikation von ca. 40 ms ist für die Übergabe aller notwendigen Daten für die Fahrt in den nächsten Block sowie für Synchronisierung erforderlich. Es ist nicht sinnvoll, die Standardverknüpfung der Blöcke bei Bearbeitung mit hohen Ansprüchen auf die Oberflächenqualität anzuwenden.

Anlage B2 - Enge Blockverknüpfung

Die enge Blockverknüpfung wird mittels der Kontinualfunktion G24 angewählt. In diesem Fall wird von der Betätigungstafel das nachfolgende Block in die Kassette bereits während Abarbeitung des vorgehenden Blocks gesendet. Die Bewegung in jedem Block wird mit einer Rampe eingeleitet und abgeschlossen und nach Ende der Ausfahrrampe wartet das System auf die Ausfahrt der Minimalabweichung gem. Maschinenkonstanten R06 - R07. Unmittelbar nach Auswertung der Ausfahrt dieser Minimalabweichung startet der nächste Block. Die Bedingung für enge Blockverknüpfung ist geringe Dauer eines Blocks. Wenn die Blockzeit größer oder gleich 30 ms und G42 programmiert ist, wird die Verknüpfung des nächsten Blocks durchgeführt. Für eine kontinuierliche Bewegung ist in diesem Fall die Dynamik des Servoantriebes sowie Steilheit der An- und Ausfahrrampe von großer Bedeutung.

Anlage B3 - Manuelle Steuerung ohne kontinuierliche Blockverknüpfung

Für die manuelle Steuerung der kontinuierlichen Blockverknüpfung muß die 1. Dekade der Maschinenkonstante auf 1 eingestellt werden. 1. Dekade der Maschinenkonstante R97:

- 0 Übergang zwischen Blöcken ohne Restzählung
- 1 Manuelle Steuerung der kontinuierlichen Blockverknüpfung mittels Funktionen G23 - G24
- 2 Automatische Erkennung der kontinuierlichen Blockfolge

Während der manuellen Steuerung wird die kontinuierliche Blockverknüpfung mit G23 und G24 gesteuert. Die Blöcke, in denen G23 programmiert wurde, werden kontinuierlich miteinander verknüpft. Dies bedeutet, daß in diesem Fall keine Rampe eingesetzt wird und auf die Ausfahrt der Minimalabweichung nicht gewartet wird. Außerdem wird der Rest der letzten Bahn im Blocktakt gleichmäßig in mehrere Überlauftakte umgerechnet,

sodaß beim dem Überlauf die Geschwindigkeit ungeändert bleibt, falls jedoch kein großer Unterschied in den Tangentialen (Richtlinien für Bewegung in beiden Blöcken) besteht. Wenn der Unterschied in den Tangentialen von nacheinander folgenden Blöcken größer ist, kommt es zu einer gewissen Änderung der Überlaufgeschwindigkeit, die jedoch gering und in diesem Fall gar nicht wichtig ist. Allgemein kann man sagen, daß je kleiner Tangentialenunterschied der beiden verknüpften Blöcke, desto kontinuierlicher Übergang.

Auswertung der kontinuierlichen Blockfolge in beiden nacheinander folgenden Blöcken.

Block N-1	Block n	Kontinuierlichkeits- zeichen	Beschreibung
			keine kont. Blockverknüpfung
			keine kont. Blockverknüpfung
			kont. Blockfverknüpfung
			kont. Blockverknüpfung

Beispiele der manuellen Steuerung einer kontinuierlichen Blockverknüpfung

plynule kontinuierlich
ramp. Rampe
všude überall

Anlage B4 - Automatische Erkennung der kontinuierlichen Blockverknüpfung

Für automatische Steuerung der kontinuierlichen Blockverknüpfung muss man die 1. Dekade der Maschinenkonstante R97 auf 2 einstellen. In der gegenwärtigen Software-Version kann man automatische Erkennung nur für die 1. und 2. Koordinate anwenden.

Die automatische Erkennung der kontinuierlichen Blockverknüpfung wird mittels G23 programmiert. Wenn die 1. Dekade der Maschinenkonstante R97 auf 2 eingestellt ist, hat G23 eine andere Bedeutung als bei manueller Steuerung. In diesem Fall erkennt das Steuersystem die Möglichkeit einer kontinuierlichen Blockverknüpfung und diese wird automatisch aktiviert und deaktiviert. In den meisten Fällen genügt es, wenn am Programmmanfang Funktion G23 vorgegeben wird, die bis zum Programmende wirksam ist. Im weiteren werden die Kriterien für die Erkennung behandelt, die im System angewendet werden.

Die automatische Erkennung der kontinuierlichen Blockverknüpfung wird mit G24 gelöscht. In diesem Fall wird die kontinuierliche Blockverknüpfung völlig (einschl. automatischer Erkennung) aufgehoben und das System steuert die Bewegung mit enger Blockverknüpfung einschl. Ein- und Ausfahrrampen. In diesem Fall gilt alles, was im Abschnitt "Enge Blockverknüpfung" beschrieben wurde. (G24 findet bei automatischer Erkennung nur selten Anwendung).

G23 ... Einschaltung der automatischen Erkennung der kontinuierlichen Blockverknüpfung hängt vom System ab, ob dieser den kontinuierlichen Blocküberlauf erkennt

G24 ... Ausschaltung der kontinuierlichen Blockverknüpfung, das System steuert die Bewegung in enger Blockverknüpfung mit Einschaltung der An- und Ausfahrrampen
(Diese Funktion findet bei automatischer Erkennung nur selten Anwendung).

Wenn die kontinuierliche Blockverknüpfung vom System erkannt und eingeschaltet wurde, werden die Blöcke kontinuierlich verknüpft (ähnlich wie bei G23 bei manueller Steuerung der kontinuierlichen Blockverknüpfung). Das System schaltet in diesem Fall keine Rampe ein und wartet auch nicht auf die Ausfahrt der Minimalabweichung. Außerdem wird der Rest der letzten Bahn im Blocktakt gleichmäßig in mehrere Überlauftakte berechnet, sodaß bei dem Überlauf die Geschwindigkeit ungeändert bleibt, falls jedoch kein großer Unterschied in den Tangentialen (Richtlinien für Bewegung in beiden Blöcken) besteht. Wenn der Unterschied in den Tangentialen von nacheinander gehenden Blöcken größer ist, kommt es zu einer gewissen Änderung der Überlaufgeschwindigkeit, die jedoch nur gering und in diesem Fall gar nicht wichtig ist. Allgemein kann man sagen, daß je kleiner Tangentialenunterschied der beiden verknüpften Blöcke ist, desto kontinuierlicher Übergang.

Toleranzwinkel für automatische Erkennung

In der Maschinenkonstante R39 wird der Toleranzwinkel für automatische Erkennung eingegeben. Diese Eingabe erfolgt mit Genauigkeit von Tausendstel-Grad, sodaß der Dezimpalpunkt gültig und die Eingabe in Grad ist. Das System berechnet den Winkel zwischen aktuellen Bewegungsrichtlinien im Punkt der Blockverknüpfung (Winkel zwischen Bewegungstangenten) und dieser Wert wird mit der Toleranzwinkel-Eingabe in R39 verglichen. Wenn der Winkel zwischen Bewegungsrichtlinien kleiner als der Toleranzwinkel ist, kann eine kontinuierliche Blockverknüpfung durchgeführt werden. Wenn Eilgang G0 vorgegeben wurde, verhindert das System kontinuierlichen Überlauf.

Notwendige Bedingungen für Erkennung der Einschaltung einer kontinuierlichen Blockverknüpfung

Programmierung von G23 und Eingabe einer "2" in der 1. Dekade der Maschinenkonstante R97.
Der Winkel zwischen Bewegungsrichtlinien beider Blöcke liegt im Toleranzbereich.
Im vorgehenden sowie aktuellen Block ist Bewegung und auch eine der Funktionen G1, G2, G3 programmiert.
Die Blockzeit ≥ 30 ms.

Toleranzwinkel
kontinuierlich
Rampe
Winkel zwischen Richtlinien

G23, G1 programmiert