

# SIEMENS

## SINUMERIK

### SINUMERIK 840D sl/828D 上級機能

機能マニュアル

適用

制御

SINUMERIK 840D sl / 840DE sl / 828D

CNC ソフトウェア バージョン 4.8 SP3

08/2018

6FC5397-1BP40-6TA2

まえがき

基本的な安全に関する指示事項

1

A4: SINUMERIK 840D sl 用デジタルとアナログ NC I/O

2

B3: 分散システム - 840D sl のみ

3

H1: 手動移動

4

K3: 補正

5

K5: 相互チャネルプログラム協調とチャネルごとの試し運転

6

K10: チャネル間の軸入れ替え

7

M1: キネマティックトランスフォーメーション

8

M5: 計測

9

N3: ポジションスイッチ、ポジションスイッチサイクル - 840D sl のみ

10

N4: 自チャネル - 840D sl のみ

11

P2: 位置決め軸

12

P5: 揺動

13

R2: 回転軸

14

S3: 主軸同期

15

S7: メモリ構成

16

T1: 割り出し軸

17

W4: 研削用工具オフセットと工具監視

18

Z2: NC/PLC インタフェース信号

19

付録

A

## 法律上の注意

### 警告事項

本書には、ユーザーの安全性を確保し製品の損傷を防止するうえ守るべき注意事項が記載されています。ユーザーの安全性に関する注意事項は、安全警告サインで強調表示されています。このサインは、物的損傷に関する注意事項には表示されません。以下に表示された注意事項は、危険度によって等級分けされています。

#### 危険

回避しなければ、直接的な死または重傷に至る危険状態を示します。

#### 警告

回避しなければ、死または重傷に至るおそれのある危険な状況を示します。

#### 注意

回避しなければ、軽度または中度の人身傷害を引き起こすおそれのある危険な状況を示します。

#### 通知

回避しなければ、物的損傷を引き起こすおそれのある危険な状況を示します。

複数の危険レベルに相当する場合は、通常、最も危険度の高い事項が表示されることになっています。安全警告サイン付きの人身傷害に関する注意事項があれば、物的損傷に関する警告が付加されます。

### 有資格者

本書が対象とする製品 / システムは必ず有資格者が取り扱うものとし、各操作内容に関連するドキュメント、特に安全上の注意及び警告が遵守されなければなりません。有資格者とは、訓練内容及び経験に基づきながら当該製品 / システムの取り扱いに伴う危険性を認識し、発生し得る危害を事前に回避できる者をいいます。

### シーメンス製品を正しくお使いいただくために

以下の事項に注意してください。

#### 警告

シーメンス製品は、カタログおよび付属の技術説明書の指示に従ってお使いください。他社の製品または部品との併用は、弊社の推奨もしくは許可がある場合に限りです。製品を正しく安全にご使用いただくには、適切な運搬、保管、組み立て、据え付け、配線、始動、操作、保守を行ってください。ご使用になる場所は、許容された範囲を必ず守ってください。付属の技術説明書に記述されている指示を遵守してください。

### 商標

®マークのついた称号はすべて **Siemens AG** の商標です。本書に記載するその他の称号は商標であり、第三者が自己の目的において使用した場合、所有者の権利を侵害することになります。

### 免責事項

本書のハードウェアおよびソフトウェアに関する記述と、実際の製品内容との一致については検証済みです。しかしなお、本書の記述が実際の製品内容と異なる可能性もあり、完全な一致が保証されているわけではありません。記載内容については定期的に検証し、訂正が必要な場合は次の版で更新いたします。

# まえがき

## SINUMERIK 取扱説明書

SINUMERIK 取扱説明書は以下のカテゴリに分類されます。

- 製品の取扱説明書/カタログ
- ユーザーマニュアル
- メーカー/サービスマニュアル

## 他の情報

次の項目に関する情報は、以下のアドレス (<https://support.industry.siemens.com/cs/de/en/view/108464614>)にあります:

- 取扱説明書の注文/取扱説明書の概要
- 説明書をダウンロードするその他のリンク
- オンラインでの説明書の利用(マニュアル/情報の検索)

ご提案や訂正など、本書に関するお問い合わせがございましたら、以下の電子メールアドレス (<mailto:docu.motioncontrol@siemens.com>)にご連絡ください。

## mySupport/ドキュメンテーション

以下のアドレス (<https://support.industry.siemens.com/My/ww/en/documentation>)では、シーメンスのコンテンツに基づいてお客さま自身の文書を作成し、お客さまの機械装置の取扱説明書にご利用いただく方法を説明しています。

## トレーニング

以下の "address (<http://www.siemens.com/sitrain>)" では、SITRAIN (製品、システム、およびオートメーションエンジニアリングソリューション用のシーメンスのトレーニング)に関する情報を提供しています。

## FAQ

[Service&Support]ページの[Product Support (<https://support.industry.siemens.com/cs/de/en/ps/faq>)]の[Frequently Asked Questions]を参照してください。

## SINUMERIK

SINUMERIK に関する情報は以下のアドレス (<http://www.siemens.com/sinumerik>)にあります。

## 対象

この文書は以下の方を対象にしています。

- プロジェクトエンジニア
- 工作機械メーカーの技術者
- セットアップエンジニア(システム/機械)
- プログラマ

## 本書の目的

本書は、対象者が機能を理解して選択できるようにするために、各機能について説明しています。対象者には、各機能の実行に必要な情報が提供されます。

## 標準仕様

この取扱説明書には、標準仕様の機能についてのみ記載されています。工作機械メーカーが実施した拡張または変更については、工作機械メーカー発行の説明書に記載されています。

その他本書で説明していない機能も、コントローラで実行できる場合があります。ただし、これは、そのような機能を新しいコントローラによって提供したり、サービス時に提供したりするというものではありません。

さらに、簡略化のため、本説明書では製品のすべてのタイプの詳細を記載していません。また、取り付け、運転および保守において想定されるすべてのケースを網羅したものではありません。

## テクニカルサポート

テクニカルサポートの国別電話番号については、インターネットの [Contact (連絡先)] の下のアドレス (<https://support.industry.siemens.com/sc/ww/en/sc/2090>)を参照してください。

## 構成と内容に関する情報

### 構成

この機能マニュアルは、以下のように構成されています。

- 機能マニュアルという表題、機能マニュアルのこの版が適用されるソフトウェアとバージョンおよび SINUMERIK コントローラ、ならびに個々の機能説明の一覧が記述された内部の見出しページ(3 ページ)
- アルファベット順(A2、A3、B1 などの略号順)の機能説明
- 付録:
  - 略語の一覧
  - 本書の概要
- 用語のインデックス

---

### 注記

データとアラームの詳細説明については、以下を参照してください。

- マシンデータとセッティングデータ  
マシンデータの詳細説明
  - NC/PLC インタフェース信号  
『NC 変数とインタフェース信号』 リストマニュアル
  - アラーム:  
『診断マニュアル』
-

## システムデータの表記法

本書では、システムデータに対して以下の表記法をが適用されます。

信号/データ	表記法	例
NC/PLC インタフェース信号	...NC/PLC インタフェース信号: <信号アドレス> (<信号名称>)	新しいギヤが入ると、以下の NC/PLC インタフェース信号が PLC プログラムによって設定されます。  DB31、... DBX16.0-2(実際のギヤ選択 A ~ C) DB31、... DBX16.3 (ギヤ切替済)
マシンデータ	...マシンデータ: <タイプ><番号> <識別子全体> (<意味>)	メイン主軸は、下記のマシンデータに設定された主軸です。  MD20090 \$MC_SPIND_DEF_MASTER_SPIND (チャネルのメイン主軸の初期設定)
セッティングデータ	...セッティングデータ: <タイプ><番号> <識別子全体> (<意味>)	論理メイン主軸は、下記のセッティングデータに含まれます。  SD42800 \$SC_SPIND_ASSIGN_TAB[0] (主軸再割り当て)

### 注記

#### 信号アドレス

機能説明書に NC/PLC インタフェース信号の<信号アドレス>として記載されているのは、SINUMERIK 840D sl に有効なアドレスだけです。SINUMERIK 828D の信号アドレスは、特定の機能マニュアルの最後にあるデータリスト「送信/受信信号」から入手してください。

## 構造の大きさ

NC/PLC インターフェースに関する説明は、以下のコンポーネントについて考えられる最大数に基づいています。

- モードグループ(DB11)
- チャネル(DB21 など)
- 軸/主軸(DB31 など)

## データタイプ

制御装置は、パートプログラムでのプログラミングに使用できる次のデータタイプを提供します。

タイプ	意味	値の範囲
INT	符号付き整数	-2,147,483,648 ... +2,147,483,647
REAL	小数点付き数値	$\approx \pm 5.0 \times 10^{-324} \dots \approx \pm 1.7 \times 10^{+308}$
BOOL	ブール値	TRUE (≠0)、FALSE (0)
CHAR	ASCII 文字およびバイト	0 ... 255 または -128 ... 127
STRING	文字列、NULL で終了	最大 400 文字の文字+ /0 (特殊文字は使用できません)
AXIS	軸名称	コントロールシステムに存在するすべての軸名称
FRAME	座標系の平行移動、座標回転、スケーリングおよびミラーリングのための幾何学的なパラメータ	---

### 配列

配列は、同種の基本データタイプからのみ形成できます。最大 3 次元の配列が使用可能です。

例:DEF INT ARRAY[2, 3, 4]

### 番号システム

次の番号システムが使用可能です。

- 10 進数: DEF INT number = 1234 or DEF REAL number = 1234.56
- 16 進数: DEF INT number = 'H123ABC'
- 2 進法: DEF INT number = 'B10001010010'

### REAL 変数の問い合わせ

NC プログラムとシンクロナイズドアクションでの REAL 変数または DOUBLE 変数の問い合わせを、制限値の評価としてプログラムすることを推奨します。

例:特定の値の軸の現在値の問い合わせ

プログラムコード	コメント
DEF REAL AXPOS = 123.456	
IF (\$VA_IM[<軸>] - 1ex-6) <= AXPOS <= (\$VA_IM[<軸>] + 1ex-6)	; 現在位置
...	== AXPOS
ELSE	

プログラムコード

コメント

...

<> AXPOS

ENDIF

# 目次

まえがき	3
<b>1 基本的な安全に関する指示事項</b>	<b>29</b>
1.1 一般的な安全に関する指示事項	29
1.2 アプリケーション例に対する保証と責任	29
1.3 産業セキュリティ	30
<b>2 A4:SINUMERIK 840D sl 用デジタルとアナログ NC I/O</b>	<b>33</b>
2.1 はじめに	33
2.2 PLC 経由の間接 I/O アクセス	35
2.2.1 概略説明	35
2.2.2 パラメータ設定	36
2.2.3 システム変数	39
2.2.4 コンパレータ入力	39
2.2.5 デジタル NC I/O	40
2.2.5.1 デジタル NC 入力	40
2.2.5.2 デジタル NC 出力	42
2.2.5.3 高速デジタル I/O の接続と論理演算	45
2.2.6 アナログ NC I/O	47
2.2.6.1 アナログ NC 入力	47
2.2.6.2 アナログ NC 出力	50
2.2.6.3 アナログ I/O 値の表現	54
2.2.7 コンパレータ入力	56
2.3 PLC 経由の直接 I/O アクセス	60
2.3.1 パラメータ設定	60
2.3.2 読み取り/書き込み:システム変数	62
2.3.3 必要条件	63
2.3.4 例	64
2.3.4.1 PLC-I/O への書き込み	64
2.3.4.2 PLC-I/O からの読み取り	65
2.4 PLC を使わない直接 I/O アクセス	66
2.4.1 概略説明	66
2.4.2 パラメータ設定	68
2.4.3 読み取り/書き込み	70
2.4.3.1 システム変数	70
2.4.4 必要条件	73
2.4.5 例	73
2.4.5.1 NC I/O への書き込み	73

2.4.5.2	NC I/O からの読み取り.....	75
2.4.5.3	状態確認による NC I/O の書き込み.....	77
2.5	データリスト.....	78
2.5.1	マシンデータ.....	78
2.5.1.1	一般マシンデータ.....	78
2.5.1.2	チャンネル別マシンデータ.....	79
2.5.2	セッティングデータ.....	80
2.5.2.1	一般セッティングデータ.....	80
2.5.3	システム変数.....	80
2.5.4	信号.....	81
2.5.4.1	NC への信号.....	81
2.5.4.2	NC からの信号.....	81
<b>3</b>	<b>B3 : 分散システム - 840D sl のみ.....</b>	<b>83</b>
3.1	概略説明.....	83
3.1.1	複数の NCU での複数の操作パネル(T:M:N).....	83
3.1.2	NCU リンク.....	86
3.1.2.1	リンク通信.....	86
3.1.2.2	リンク変数.....	87
3.1.2.3	リンク軸.....	88
3.1.2.4	マスターリンク軸.....	88
3.1.2.5	従属関係.....	89
3.1.2.6	用途例:ロータリインデックスマシン.....	90
3.2	NCU リンク.....	92
3.2.1	リンク通信.....	92
3.2.1.1	概要.....	92
3.2.1.2	リンクモジュール.....	97
3.2.1.3	パラメータ設定 NC システム周期.....	98
3.2.1.4	パラメータ設定 リンク通信.....	100
3.2.1.5	設定.....	101
3.2.1.6	NCU の配線.....	101
3.2.1.7	適用.....	102
3.2.2	リンク変数.....	102
3.2.2.1	リンク変数メモリの特性.....	103
3.2.2.2	リンク変数の特性.....	104
3.2.2.3	書き込み要素.....	105
3.2.2.4	書き込み時のダイナミック応答.....	105
3.2.2.5	システム変数.....	106
3.2.2.6	書き込み要求の同期.....	107
3.2.2.7	例: リンク変数メモリの構成.....	107
3.2.2.8	例: ドライブデータの読み取り.....	109
3.2.3	リンク軸.....	110
3.2.3.1	概要.....	110
3.2.3.2	リンク軸名称.....	112

3.2.3.3	パラメータ設定.....	112
3.2.3.4	主軸の補助機能出力.....	113
3.2.3.5	補足条件.....	114
3.2.4	軸コンテナ.....	117
3.2.4.1	概要.....	117
3.2.4.2	パラメータ設定.....	120
3.2.4.3	プログラミング.....	127
3.2.4.4	システム変数.....	129
3.2.4.5	軸コンテナによる加工(概略図).....	131
3.2.4.6	さまざまな動作状態の特性.....	131
3.2.4.7	軸コンテナ回転のイネーブルを取り消す場合の特性.....	132
3.2.4.8	補足条件.....	134
3.2.5	マスタリンク軸.....	137
3.2.5.1	概要.....	137
3.2.5.2	パラメータ設定.....	138
3.2.5.3	マスタ値を入力するためのシステム変数.....	140
3.2.5.4	補足条件.....	140
3.2.5.5	例.....	141
3.2.6	リンクグループ内の単位系.....	141
3.3	例.....	142
3.3.1	リンク軸.....	142
3.3.2	軸コンテナの協調.....	143
3.3.2.1	パートプログラムの待機なしの軸コンテナの回転.....	144
3.3.2.2	パートプログラムが自動的に待機する軸コンテナの回転.....	144
3.3.2.3	1チャンネルのみによる軸コンテナの回転(例: 電源投入時).....	144
3.3.3	軸コンテナシステム変数の使用.....	145
3.3.3.1	条件付き分岐.....	145
3.3.3.2	<b>\$AN_AXCTSWA</b> による静的なシンクロナイズドアクション.....	145
3.3.3.3	軸コンテナ回転の個別の完了を待機します。.....	146
3.3.4	複数主軸旋盤の構成.....	147
3.3.5	マスタリンク軸.....	158
3.3.5.1	構成.....	158
3.3.5.2	プログラミング.....	160
3.4	データリスト.....	161
3.4.1	マシンデータ.....	161
3.4.1.1	一般マシンデータ.....	161
3.4.1.2	チャンネル別マシンデータ.....	162
3.4.1.3	軸/主軸別マシンデータ.....	162
3.4.2	セッティングデータ.....	163
3.4.2.1	一般セッティングデータ.....	163
3.4.2.2	軸/主軸別セッティングデータ.....	163
3.4.3	信号.....	163
3.4.3.1	NC からの信号.....	163
3.4.3.2	HMI/PLC からの信号.....	164

3.4.3.3	一般オンラインインタフェース.....	165
3.4.3.4	軸/主軸からの信号.....	167
3.4.4	システム変数.....	167
<b>4</b>	<b>H1:手動移動.....</b>	<b>169</b>
4.1	概要.....	169
4.2	PLC インターフェースを使った制御.....	173
4.3	パラメータ割り付け(一般).....	175
4.4	連続手動移動.....	181
4.4.1	機能.....	181
4.4.2	パラメータ設定.....	185
4.4.3	必要条件.....	185
4.5	インクリメンタル手動移動.....	185
4.5.1	機能.....	185
4.5.2	パラメータ設定.....	189
4.5.3	必要条件.....	190
4.6	ハンドルを使用した手動移動.....	190
4.6.1	機能.....	190
4.6.2	パラメータ設定.....	198
4.6.3	移動要求.....	205
4.7	主軸の手動による移動.....	208
4.8	ジオメトリ軸/旋回軸の手動による移動.....	210
4.9	JOG モードの固有位置へのアプローチ.....	213
4.9.1	機能.....	213
4.9.2	パラメータ設定.....	217
4.9.3	プログラミング.....	220
4.9.4	補足条件.....	220
4.9.5	用途例.....	221
4.10	JOG での位置移動.....	222
4.10.1	機能.....	222
4.10.2	パラメータ設定.....	225
4.10.3	補足条件.....	226
4.10.4	用途例.....	227
4.11	JOG での円移動.....	228
4.11.1	機能.....	228
4.11.2	パラメータ設定.....	234
4.11.3	補足条件.....	237
4.11.4	用途例.....	238
4.12	工具方向の後退(ジョグ後退).....	240
4.12.1	概要.....	240

4.12.2	パラメータ設定.....	241
4.12.2.1	電源投入後のジョグ後退の自動選択.....	241
4.12.2.2	移動方向の有効化.....	242
4.12.2.3	検出器の状態.....	242
4.12.3	選択.....	243
4.12.4	工具後退.....	245
4.12.5	解除.....	246
4.12.6	再選択.....	247
4.12.7	加工の継続.....	248
4.12.8	状態図.....	249
4.12.9	システムデータ.....	250
4.12.10	必要条件.....	250
4.13	自動モードでのハンドルの使用.....	251
4.13.1	自動モードのハンドルオーバーライド.....	251
4.13.1.1	一般機能.....	251
4.13.1.2	プログラミングとハンドルオーバーライドの有効化.....	257
4.13.1.3	自動モードのハンドルオーバーライドの特記事項.....	259
4.13.2	ハンドルを使用した輪郭ハンドル/軌跡指令(オプション).....	260
4.13.3	DRF オフセット.....	264
4.13.3.1	DRF オフセット.....	264
4.13.3.2	セットアップ.....	267
4.13.3.3	プログラミング軸別の重畳の解除(CORROF).....	270
4.13.4	ハンドルの二重使用.....	273
4.14	監視機能.....	275
4.15	セットアップ:ハンドル.....	276
4.15.1	一般情報.....	276
4.15.2	PPU による接続 (828D のみ).....	278
4.15.3	PROFIBUS による接続 (828D).....	279
4.15.4	PROFIBUS による接続(840D sl).....	280
4.15.5	Ethernet を介した接続 - (840D sl のみ).....	284
4.16	データリスト.....	289
4.16.1	マシンデータ.....	289
4.16.1.1	一般マシンデータ.....	289
4.16.1.2	チャンネルマシンデータ.....	290
4.16.1.3	軸/主軸別マシンデータ.....	290
4.16.2	セッティングデータ.....	291
4.16.2.1	一般セッティングデータ.....	291
4.16.2.2	チャンネルセッティングデータ.....	292
4.16.2.3	軸/主軸セッティングデータ.....	292
4.16.3	信号.....	292
4.16.3.1	NC からの信号.....	292
4.16.3.2	HMI からの信号.....	293
4.16.3.3	モードグループの信号.....	293

4.16.3.4	モードグループからの信号.....	294
4.16.3.5	チャンネルへの信号.....	294
4.16.3.6	チャンネルからの信号.....	296
4.16.3.7	軸/主軸への信号.....	299
4.16.3.8	軸/主軸からの信号.....	299
4.16.4	システム変数.....	300
4.16.4.1	システム変数.....	300
4.16.5	OPI 変数.....	300
4.16.5.1	OPI 変数.....	300
<b>5</b>	<b>K3:補正.....</b>	<b>301</b>
5.1	はじめに.....	301
5.2	熱変位補正.....	302
5.2.1	機能.....	302
5.2.2	セットアップ.....	305
5.2.3	例.....	306
5.2.3.1	旋盤のZ軸の熱変位補正のセットアップ.....	306
5.3	バックラッシ補正.....	309
5.3.1	機械バックラッシ補正.....	309
5.3.1.1	機能.....	309
5.3.1.2	試運転:軸マシンデータ.....	311
5.3.2	ダイナミックバックラッシ補正.....	313
5.3.2.1	機能.....	313
5.3.2.2	試運転:軸マシンデータ.....	314
5.3.3	二重位置検出.....	314
5.3.3.1	試運転:軸マシンデータ.....	315
5.3.3.2	必要条件.....	316
5.4	ノディング補正.....	316
5.4.1	機能の説明.....	316
5.4.1.1	オプション.....	316
5.4.1.2	特徴.....	316
5.4.1.3	位置に依存しない整合係数.....	319
5.4.1.4	補正関係.....	320
5.4.2	試運転:マシンデータ.....	321
5.4.2.1	一覧.....	321
5.4.2.2	ノディング補正のスージングの時定数.....	322
5.4.2.3	加速軸.....	322
5.4.2.4	調整軸.....	323
5.4.2.5	調整特性の位置の数.....	324
5.4.2.6	調整特性の位置.....	325
5.4.2.7	整合係数.....	326
5.4.2.8	ファンクションダイアグラム.....	329
5.5	補間補正.....	330

5.5.1	一般特性.....	330
5.5.2	ピッチ誤差およびピッチ誤差補正.....	333
5.5.2.1	検出器誤差補正(MSEC: Measuring System Error Compensation).....	333
5.5.2.2	セットアップ.....	334
5.5.2.3	例.....	337
5.5.3	真直度および直角度補正.....	338
5.5.3.1	一般情報.....	338
5.5.3.2	試運転:マシンデータ.....	341
5.5.3.3	試運転:セッティングデータ.....	343
5.5.3.4	試運転:システム変数.....	343
5.5.3.5	試運転:基本手順.....	348
5.5.3.6	試運転:概要図.....	349
5.5.3.7	例 1:真直度補正.....	350
5.5.3.8	例 2:テーブルの乗算を使った補正.....	351
5.5.3.9	例 3:補正值の二次元配列.....	352
5.5.4	NCU リンクを使用した真直度補正の拡張 - 840D sl のみ.....	358
5.5.4.1	一般的なパラメータ割り付け.....	358
5.5.4.2	チャンネル名称を使ったパラメータ設定.....	359
5.5.4.3	機械軸名称を使ったパラメータ設定.....	359
5.5.4.4	軸コンテナ.....	360
5.5.4.5	構成例 1:固定結合の NCU リンク.....	361
5.5.4.6	構成例 2: 軸コンテナとの NCU リンク.....	363
5.5.5	方向に依存したピッチ誤差補正.....	367
5.5.5.1	機能説明.....	367
5.5.5.2	セットアップ.....	368
5.5.5.3	例.....	371
5.5.6	シリンダエラー補正.....	374
5.5.6.1	オプション.....	374
5.5.6.2	機能.....	374
5.5.6.3	セットアップ.....	375
5.5.6.4	例.....	380
5.5.7	必要条件.....	385
5.6	動的なフィードフォワード制御 (追従誤差補正) .....	387
5.6.1	一般特性.....	387
5.6.2	速度フィードフォワード制御.....	389
5.6.3	トルクフィードフォワード制御.....	391
5.6.4	ダイナミック応答調整.....	393
5.6.5	命令軸および PLC 軸のフィードフィード制御.....	394
5.6.6	二次条件.....	396
5.7	摩擦補正の概要.....	397
5.8	定補正值による摩擦補正.....	398
5.8.1	機能説明.....	398
5.8.2	セットアップ.....	398

5.8.2.1	真円度テスト.....	400
5.8.3	必要条件.....	403
5.9	調整特性付きの摩擦補正.....	404
5.9.1	機能説明.....	404
5.9.2	セットアップ.....	405
5.9.3	必要条件.....	407
5.10	調整特性付きの摩擦補正.....	408
5.10.1	機能の説明.....	408
5.10.2	セットアップ.....	409
5.10.2.1	機能の有効化.....	409
5.10.2.2	<b>SINUMERIK Operate</b> ユーザーインターフェースの機能のセットアップ機能.....	410
5.10.2.3	特性補間点での加速度のパラメータ化.....	413
5.10.2.4	速度セットポイントパルス.....	414
5.10.2.5	トルク指令値パルス.....	417
5.11	吊り下げ軸の補正機能.....	421
5.11.1	電子カウンタウェイト.....	421
5.11.2	特殊機能:再起動遅延.....	422
5.12	データリスト.....	425
5.12.1	マシンデータ.....	425
5.12.1.1	一般マシンデータ.....	425
5.12.1.2	チャンネル別マシンデータ.....	425
5.12.1.3	軸/主軸別マシンデータ.....	425
5.12.2	セッティングデータ.....	428
5.12.2.1	一般セッティングデータ.....	428
5.12.2.2	軸/主軸別セッティングデータ.....	428
5.12.3	信号.....	428
5.12.3.1	NC からの信号.....	428
5.12.3.2	モードグループからの信号.....	428
5.12.3.3	チャンネルからの信号.....	428
5.12.3.4	軸/主軸への信号.....	429
5.12.3.5	軸/主軸からの信号.....	429
<b>6</b>	<b>K5:相互チャンネルプログラム協調とチャンネルごとの試し運転.....</b>	<b>431</b>
6.1	相互チャンネルプログラム協調.....	431
6.1.1	協調指令(INIT、START、WAITM、WAITMC、WAITE、SETM、CLEARM).....	431
6.1.2	連続軌跡モードでの条件待機(WAITMC).....	437
6.2	チャンネル単位の試し運転.....	441
6.2.1	機能.....	441
6.2.2	処理.....	442
6.2.3	シングルチャンネル表示.....	443
6.2.4	マルチチャンネル表示.....	445
6.2.5	システム変数.....	448
6.2.6	必要条件.....	448

6.2.7	例.....	449
6.3	必要条件.....	451
6.3.1	MDI モード:経路制御モードと WAITMC.....	451
6.3.2	WAIT 指令後の移動動作の非同期開始.....	451
<b>7</b>	<b>K10:チャンネル間の軸入れ替え.....</b>	<b>455</b>
7.1	一覧.....	455
7.2	セットアップ.....	456
7.3	プログラミング軸の解放(RELEASE).....	457
7.4	プログラミング軸の確保(GET、GETD).....	458
7.5	自動軸入れ替え.....	460
7.6	PLC による軸入れ替え.....	461
7.7	軸コンテナ回転による軸入れ替え.....	464
7.8	先読み停止処理のある軸入れ替えと先読み停止処理のない軸入れ替え.....	465
7.9	PLC のみの制御軸.....	467
7.10	PLC に完全に固定された軸.....	468
7.11	回転フレームのジオメトリ軸と軸入れ替え.....	469
7.12	シンクロナイズドアクションを介した軸入れ替え.....	471
7.13	マスタ軸(ガントリ)の軸入れ替え.....	473
7.14	状態図.....	474
7.15	境界条件.....	475
7.15.1	PLC による軸入れ替え.....	476
7.15.2	計算ありのブロック検索.....	476
7.16	例.....	477
7.17	データリスト.....	479
7.17.1	マシンデータ.....	479
7.17.1.1	一般マシンデータ.....	479
7.17.1.2	チャンネル別マシンデータ.....	479
7.17.1.3	軸/主軸別マシンデータ.....	482
7.17.2	セッティングデータ.....	483
7.17.2.1	チャンネル別セッティングデータ.....	483
7.17.3	信号.....	483
7.17.3.1	チャンネルへの信号.....	483
7.17.3.2	チャンネルからの信号.....	483
7.17.3.3	軸/主軸への信号.....	483
7.17.3.4	軸/主軸からの信号.....	484

<b>8</b>	<b>M1:キネマティックトランスフォーメーション</b> .....	<b>485</b>
8.1	TRANSMIT 面端座標変換(オプション).....	485
8.1.1	機能.....	485
8.1.1.1	はじめに.....	485
8.1.1.2	加工選択.....	487
8.1.1.3	ワーキングエリアリミット.....	493
8.1.1.4	TRANSMIT による重畳動作.....	495
8.1.1.5	回転量が 360°を超える回転軸の監視.....	495
8.1.2	パラメータの割り付け.....	496
8.1.2.1	一覧.....	496
8.1.2.2	軸設定.....	497
8.1.2.3	個別設定.....	499
8.1.3	プログラミング.....	502
8.1.4	制約事項.....	503
8.1.5	例.....	504
8.2	TRACYL 円筒補間(オプション).....	508
8.2.1	機能.....	508
8.2.2	パラメータの割り付け.....	512
8.2.2.1	一覧.....	512
8.2.2.2	軸構成.....	514
8.2.2.3	個別設定.....	516
8.2.3	プログラミング.....	521
8.2.4	境界条件.....	524
8.2.5	例.....	527
8.2.5.1	X-Y-Z-C キネマティックスによる円筒面上の溝加工.....	527
8.2.5.2	X-Y-Z-A-C キネマティックスによる円筒面上の溝加工.....	532
8.3	傾斜角座標変換(傾斜軸) TRAANG (オプション).....	536
8.3.1	機能.....	536
8.3.2	パラメータの割り付け.....	537
8.3.2.1	一覧.....	537
8.3.2.2	軸構成.....	539
8.3.2.3	個別設定.....	541
8.3.3	プログラミング.....	542
8.3.3.1	角度がプログラム指令可能な傾斜角座標変換(TRAANG)の有効化(TRAANG).....	542
8.3.3.2	角度が固定の傾斜角座標変換の有効化(TRAANG).....	544
8.3.3.3	研削盤での傾斜プランジ切削(G5, G7).....	545
8.3.4	境界条件.....	546
8.3.5	例.....	549
8.4	座標変換重畳.....	552
8.4.1	機能.....	552
8.4.1.1	はじめに.....	552
8.4.1.2	システム変数.....	554
8.4.2	プログラミング.....	557

8.4.3	例.....	559
8.4.3.1	座標変換重畳の用途例.....	559
8.4.3.2	座標変換重畳の軸位置の特定.....	563
8.5	モーダル座標変換.....	566
8.6	直交 PTP 移動.....	570
8.6.1	機能.....	570
8.6.2	セットアップ.....	574
8.6.2.1	電源投入後の動作.....	574
8.6.2.2	リセット/パートプログラム終了後のレスポンス.....	574
8.6.2.3	PTP 移動中に SW 制限を考慮.....	575
8.6.2.4	STAT と TU の表示.....	576
8.6.3	プログラミング.....	576
8.6.3.1	直交 PTP 移動(PTP、PTPG0、PTPWOC、CP)の有効化/無効化.....	576
8.6.3.2	ジョイント(STAT)の位置を指定します.....	578
8.6.3.3	軸角度の符号を指定します(TU).....	582
8.6.3.4	例 1:ROBX 座標変換による 6 軸ロボットの PTP 移動.....	585
8.6.3.5	例 2:総合 5 軸座標変換の場合の PTP 移動.....	586
8.6.3.6	例 3:PTPG0 および TRANSMIT.....	587
8.6.4	補足条件.....	588
8.7	直交座標ジョグ移動(オプション).....	590
8.8	パートプログラム/ソフトキーによる座標変換マシンデータの起動.....	599
8.8.1	機能.....	599
8.8.2	制約事項.....	600
8.8.3	電源投入、モード変更、RESET、ブロック検索、REPOS へのコントロールシ テムの動作.....	602
8.8.4	対応するマシンデータのリスト.....	603
8.8.5	例.....	606
8.9	データリスト.....	607
8.9.1	マシンデータ.....	607
8.9.1.1	TRANSMIT.....	607
8.9.1.2	TRACYL.....	608
8.9.1.3	TRAANG.....	610
8.9.1.4	座標変換重畳.....	612
8.9.1.5	共通変換マシンデータ.....	612
8.9.2	信号.....	613
8.9.2.1	チャンネルへの信号.....	613
8.9.2.2	チャンネルからの信号.....	613
<b>9</b>	<b>M5:計測.....</b>	<b>615</b>
9.1	概略説明.....	615
9.2	ハードウェアの必要条件.....	616
9.2.1	使用可能なプローブ.....	616

9.3	チャンネル別測定.....	618
9.3.1	計測.....	618
9.3.2	計測結果.....	619
9.4	軸別の計測.....	620
9.4.1	計測.....	620
9.4.2	メッセージ選択.....	629
9.4.3	測定結果.....	630
9.5	原点の設定、ワーク測定、および工具測定.....	631
9.5.1	現在位置設定と接触計測.....	631
9.5.2	ワーク測定.....	633
9.5.2.1	入力値.....	633
9.5.2.2	測定の選択.....	643
9.5.2.3	出力値.....	644
9.5.2.4	計算方法.....	645
9.5.2.5	計測および計算の計測変数の単位系.....	648
9.5.2.6	診断.....	650
9.5.3	ワーク測定タイプ.....	651
9.5.3.1	エッジの計測(計測タイプ 1、2、3).....	651
9.5.3.2	角度計測(計測タイプ 4、5、6、7).....	656
9.5.3.3	穴計測(計測タイプ 8).....	660
9.5.3.4	シャフトの計測(計測タイプ 9).....	663
9.5.3.5	溝の計測(計測タイプ 12).....	665
9.5.3.6	ウェブの計測(計測タイプ 13).....	668
9.5.3.7	ジオメトリ軸と付加軸の計測(計測タイプ 14、15).....	669
9.5.3.8	傾斜エッジの計測(計測タイプ 16).....	672
9.5.3.9	平面での傾斜角度の計測(計測タイプ 17).....	674
9.5.3.10	WCS 基準フレーム回転計測の再定義(計測タイプ 18).....	678
9.5.3.11	1次元、2次元、および3次元の指令値選択の計測(計測タイプ 19、20、21).....	682
9.5.3.12	任意の座標系での計測点の計測(計測タイプ 24).....	687
9.5.3.13	長方形の計測(計測タイプ 25).....	690
9.5.3.14	データ管理フレームを保存するための計測(計測タイプ 26).....	692
9.5.3.15	バックアップされたデータ管理フレームを復元するための計測(計測タイプ 27).....	693
9.5.3.16	テーパ旋削の追加回転を定義するための計測(計測タイプ 28).....	694
9.5.4	工具測定.....	695
9.5.5	タイプ s of workpiece measurement.....	696
9.5.5.1	工具長の計測(計測タイプ 10).....	696
9.5.5.2	工具直径の測定(測定タイプ 11).....	699
9.5.5.3	ズームイン機能による工具長測定(測定タイプ 22).....	700
9.5.5.4	保存済みの位置または現在の位置をもつ工具長の測定(測定タイプ 23).....	701
9.5.5.5	次の向きをもつ2種類の工具長の測定:.....	703
9.6	測定精度および機能テスト.....	716
9.6.1	計測精度.....	716
9.6.2	プローブ機能テスト.....	717

9.7	計測シミュレーション.....	717
9.7.1	一般機能.....	717
9.7.2	位置に対応した切り替え要求.....	718
9.7.3	外部切り替え要求.....	720
9.7.4	システム変数.....	721
9.8	チャンネル - 840D sl のみ.....	722
9.8.1	計測モード 1.....	722
9.8.2	計測モード 2.....	723
9.8.3	連続計測.....	723
9.8.4	機能テストと繰り返し精度.....	724
9.9	データリスト.....	726
9.9.1	マシンデータ.....	726
9.9.1.1	一般マシンデータ.....	726
9.9.1.2	チャンネル別マシンデータ.....	727
9.9.2	システム変数.....	727
<b>10</b>	<b>N3: ポジションスイッチ、ポジションスイッチサイクル - 840D sl のみ.....</b>	<b>731</b>
10.1	概略説明.....	731
10.2	ポジションスイッチ信号および動作位置.....	732
10.2.1	個別出力のためのポジションスイッチ信号の生成.....	732
10.2.2	ゲート制出力をもつポジションスイッチ信号の生成.....	736
10.2.3	動作位置.....	740
10.2.4	リード時間/遅延時間(ダイナミックスイッチ).....	742
10.3	ポジションスイッチ信号の出力.....	744
10.3.1	有効化.....	744
10.3.2	PLC へのポジションスイッチ信号の出力.....	744
10.3.3	位置制御周期での NCK I/O へのポジションスイッチ信号の出力.....	745
10.3.4	割り込み型ポジションスイッチ信号出力.....	747
10.3.5	独立した、割り込み型ポジションスイッチ信号出力.....	749
10.4	位置タイマスイッチ.....	750
10.5	補足条件.....	752
10.6	データリスト.....	752
10.6.1	マシンデータ.....	752
10.6.1.1	一般マシンデータ.....	752
10.6.2	セッティングデータ.....	753
10.6.2.1	一般セッティングデータ.....	753
10.6.3	信号.....	754
10.6.3.1	軸/主軸への信号.....	754
10.6.3.2	軸/主軸からの信号.....	754
<b>11</b>	<b>N4: 自チャンネル - 840D sl のみ.....</b>	<b>755</b>
11.1	概略説明.....	755

11.2	ストローク制御.....	755
11.2.1	一般情報.....	755
11.2.2	高速信号.....	756
11.2.3	ストローク開始の条件.....	758
11.2.4	パンチング後の軸の開始.....	760
11.2.5	パンチングおよびニブリングに固有の PLC 信号.....	761
11.2.6	標準 PLC 信号への、パンチングおよびニブリング固有の応答.....	762
11.2.7	信号監視.....	763
11.3	適用と解除.....	763
11.3.1	言語命令.....	763
11.3.2	機能拡張.....	768
11.3.3	以前のシステムとの互換性の章.....	773
11.4	自動軌跡分割.....	775
11.4.1	一般情報.....	775
11.4.2	軌跡軸の動作特性.....	777
11.4.3	単独軸についての動作.....	781
11.5	回転工具.....	786
11.5.1	一般情報.....	786
11.5.2	パンチとダイの連結動作.....	787
11.5.3	法線方向制御.....	788
11.6	保護領域.....	792
11.7	補足条件.....	793
11.8	例.....	793
11.8.1	ニブリング運転開始の定義例.....	793
11.9	データリスト.....	799
11.9.1	マシンデータ.....	799
11.9.1.1	一般マシンデータ.....	799
11.9.1.2	チャンネル別マシンデータ.....	799
11.9.2	セッティングデータ.....	800
11.9.2.1	チャンネル別セッティングデータ.....	800
11.9.3	信号.....	800
11.9.3.1	チャンネルへの信号.....	800
11.9.3.2	チャンネルからの信号.....	800
11.9.4	言語命令.....	801
<b>12</b>	<b>P2 :位置決め軸.....</b>	<b>803</b>
12.1	製品の概要.....	803
12.2	自チャンネル、位置決め軸、または同時位置決め軸.....	806
12.2.1	自チャンネル.....	806
12.2.2	位置決め軸.....	807
12.2.3	同時位置決め軸.....	810

12.3	移動動作および補間機能.....	811
12.3.1	軌跡補間器および軸補間器.....	811
12.3.2	独立の単独軸運転.....	812
12.3.3	NC 制御 ESR による独立の単独軸機能.....	817
12.4	位置決め軸のダイナミック応答.....	819
12.5	プログラミング.....	822
12.5.1	概要.....	822
12.5.2	毎回転送り速度の外部プログラミング.....	825
12.6	ブロック切り替え.....	825
12.6.1	設定可能なブロック切り替えタイミング.....	827
12.6.2	ブロック検索の動作終了条件.....	833
12.7	PLC による制御.....	834
12.7.1	PLC からの同時位置決め軸の開始.....	836
12.7.2	PLC 制御軸.....	837
12.7.3	PLC 制御軸の応答制御.....	839
12.8	応用機能の動作.....	840
12.8.1	ドライラン(DRY RUN).....	840
12.8.2	シングルブロック.....	840
12.9	例.....	841
12.9.1	移動動作と補間機能.....	841
12.10	データリスト.....	842
12.10.1	マシンデータ.....	842
12.10.1.1	NC 別マシンデータ.....	842
12.10.1.2	チャンネル別マシンデータ.....	842
12.10.1.3	軸/主軸別マシンデータ.....	843
12.10.2	セッティングデータ.....	843
12.10.2.1	軸/主軸別セッティングデータ.....	843
12.10.3	信号.....	843
12.10.3.1	チャンネルへの信号.....	843
12.10.3.2	チャンネルからの信号.....	844
12.10.3.3	軸/主軸への信号.....	844
12.10.3.4	軸/主軸からの信号.....	844
<b>13</b>	<b>P5 :揺動.....</b>	<b>847</b>
13.1	概略説明.....	847
13.2	非同期揺動.....	848
13.2.1	非同期揺動の動作.....	849
13.2.2	PLC 制御による非同期揺動.....	856
13.2.3	非同期揺動中の特殊応答.....	857
13.3	シンクロナイズドアクションによる揺動制御.....	861
13.3.1	反転点 1 または 2 の切り込み.....	865

13.3.2	反転点範囲の切り込み.....	866
13.3.3	両方の反転点の切り込み.....	867
13.3.4	反転点で揺動移動を停止.....	868
13.3.5	揺動移動の再開.....	869
13.3.6	区間切り込みの早期開始の防止.....	870
13.3.7	揺動軸と切り込み軸の割り当て OSCILL.....	871
13.3.8	切り込みの定義 POSP.....	871
13.3.9	外部揺動反転.....	872
13.4	制約条件.....	874
13.5	例.....	874
13.5.1	非同期揺動例.....	874
13.5.2	シンクロナイズドアクションによる揺動の例 1.....	875
13.5.3	シンクロナイズドアクションによる揺動の例 2.....	878
13.5.4	開始位置の例.....	880
13.5.4.1	言語命令による開始位置の定義.....	880
13.5.4.2	セッティングデータによる揺動の開始.....	881
13.5.4.3	ノンモーダル揺動(開始位置 = 反転点 1).....	881
13.5.5	外部揺動反転例.....	884
13.5.5.1	「外部揺動反転」を使用してシンクロナイズドアクションによって反転位置を変更.....	884
13.6	データリスト.....	884
13.6.1	マシンデータ.....	884
13.6.1.1	一般マシンデータ.....	884
13.6.2	セッティングデータ.....	885
13.6.2.1	軸/主軸別セッティングデータ.....	885
13.6.3	信号.....	885
13.6.3.1	軸/主軸への信号.....	885
13.6.3.2	軸/主軸からの信号.....	886
13.6.4	システム変数.....	886
13.6.4.1	モーションシンクロナイズドアクションの実行プログラム変数.....	886
<b>14</b>	<b>R2 : 回転軸.....</b>	<b>891</b>
14.1	簡単な説明.....	891
14.2	モジュロ 360°.....	893
14.3	回転軸のプログラミング.....	899
14.3.1	一般情報.....	899
14.3.2	モジュロ変換ありの回転軸(回転軸の連続回転).....	900
14.3.3	モジュロ変換がない回転軸.....	903
14.3.4	回転軸に関連するその他のプログラミング機能.....	906
14.4	回転軸の有効化.....	906
14.5	回転軸の特記事項.....	908
14.6	例.....	909

14.7	データリスト.....	910
14.7.1	マシンデータ.....	910
14.7.1.1	一般マシンデータ.....	910
14.7.1.2	軸/主軸別マシンデータ.....	910
14.7.2	セッティングデータ.....	910
14.7.2.1	一般セッティングデータ.....	910
14.7.2.2	軸/主軸別セッティングデータ.....	911
14.7.3	信号.....	911
14.7.3.1	軸/主軸への信号.....	911
14.7.3.2	軸/主軸からの信号.....	911
<b>15</b>	<b>S3:主軸同期.....</b>	<b>913</b>
15.1	概略説明.....	913
15.1.1	機能.....	913
15.1.2	同期モード.....	915
15.1.3	同期モードの必要条件.....	922
15.1.4	パートプログラムの同期モードの選択.....	923
15.1.5	パートプログラムの同期モードの解除.....	925
15.1.6	PLC による主軸同期連結制御.....	927
15.1.7	同期運転の監視.....	930
15.2	プログラミング.....	932
15.2.1	定義 (COUPDEF).....	933
15.2.2	連結(COUPON、COUPONC、COUPOF)のオンとオフを切り換えます.....	937
15.2.3	主軸同期の軸のシステム変数.....	938
15.2.4	位置制御の自動選択と選択解除.....	939
15.3	設定.....	940
15.3.1	NC スタートの主軸同期連結の動作.....	942
15.3.2	リセットの主軸同期連結の特性.....	942
15.4	注意点.....	943
15.4.1	同期モードの一般的な特記事項.....	943
15.4.2	スレーブ主軸の再同期.....	946
15.4.3	同期モードと NC/PLC インタフェース信号.....	948
15.4.4	マスタ主軸とスレーブ主軸の速度差.....	953
15.4.5	同期補正中の同期信号の動作.....	958
15.4.6	同期補正削除と NC リセット.....	958
15.4.7	主軸同期連結の起動に関する注意事項.....	959
15.5	境界条件.....	965
15.6	例.....	965
15.7	データリスト.....	966
15.7.1	マシンデータ.....	966
15.7.1.1	NC 別マシンデータ.....	966
15.7.1.2	チャンネル別マシンデータ.....	966

15.7.1.3	軸/主軸別マシンデータ.....	966
15.7.2	セッティングデータ.....	967
15.7.2.1	チャンネル別セッティングデータ.....	967
15.7.3	信号.....	968
15.7.3.1	チャンネルへの信号.....	968
15.7.3.2	チャンネルからの信号.....	968
15.7.3.3	軸/主軸への信号.....	968
15.7.3.4	軸/主軸からの信号.....	969
15.7.4	システム変数.....	969
<b>16</b>	<b>S7:メモリ構成.....</b>	<b>971</b>
16.1	はじめに.....	971
16.2	アクティブファイルシステムとパッシブファイルシステム.....	972
16.3	セットアップ.....	973
16.3.1	設定.....	973
16.3.2	再構成.....	973
16.4	スタティックユーザーメモリの構成.....	975
16.4.1	スタティック NC メモリの区分.....	975
16.4.2	セットアップ.....	978
16.5	ダイナミックユーザーメモリの構成.....	979
16.5.1	ダイナミック NC メモリの区分.....	979
16.5.2	セットアップ.....	981
16.6	境界条件.....	982
16.6.1	チャンネルの数と工具ホルダによって異なります。.....	982
16.6.2	チャンネル/軸の数のその後の低減.....	983
16.7	データリスト.....	983
16.7.1	マシンデータ.....	983
16.7.1.1	一般マシンデータ.....	983
16.7.1.2	チャンネル別マシンデータ.....	988
16.7.1.3	軸/主軸別マシンデータ.....	990
<b>17</b>	<b>T1:割り出し軸.....</b>	<b>991</b>
17.1	概略説明.....	991
17.2	詳細説明.....	991
17.2.1	AUTOMATIC モードの割り出し軸の移動.....	991
17.2.2	JOG モードの割り出し軸の移動.....	992
17.2.3	PLC による割り出し軸の移動.....	994
17.3	セットアップ.....	994
17.3.1	割り出し軸のマシンデータ.....	994
17.3.2	等間隔割り出し区間のマシンデータ.....	998
17.3.2.1	機能.....	998
17.3.2.2	ハース軸.....	1000

17.3.3	軸固有のシステム変数.....	1000
17.3.4	NC/PLC インタフェース信号.....	1001
17.4	プログラミング.....	1002
17.5	必要条件.....	1005
17.6	例.....	1008
17.6.1	例 1:割り出し軸としての回転軸.....	1008
17.6.2	例 2:直線軸としての割り出し軸.....	1009
17.6.3	例 3 :等間隔の割り出し軸としての回転モジュロ軸.....	1010
17.6.4	例 4 :移動範囲制限付きの等間隔の割り出し軸としての回転軸.....	1011
17.6.5	例 5 :等間隔の割り出し軸としての直線軸.....	1012
17.6.6	例 6 :「ハース軸」 .....	1013
17.7	データリスト.....	1014
17.7.1	マシンデータ.....	1014
17.7.1.1	一般マシンデータ.....	1014
17.7.1.2	軸/主軸別マシンデータ.....	1014
17.7.2	セッティングデータ.....	1015
17.7.2.1	一般セッティングデータ.....	1015
17.7.3	信号.....	1015
17.7.3.1	軸/主軸からの信号.....	1015
17.7.4	システム変数.....	1015
<b>18</b>	<b>W4:研削用工具オフセットと工具監視.....</b>	<b>1017</b>
18.1	研削用工具データ.....	1017
18.1.1	工具データの構造.....	1017
18.1.2	刃先用パラメータ.....	1019
18.1.2.1	刃先用パラメータのリスト.....	1019
18.1.2.2	\$TC_DP1.....	1021
18.1.2.3	追加パラメータの定義\$TC_DPC1 ... 10.....	1022
18.1.3	工具用パラメータ.....	1022
18.1.3.1	工具用パラメータのリスト.....	1022
18.1.3.2	\$TC_TPG1.....	1023
18.1.3.3	\$TC_TPG2.....	1023
18.1.3.4	\$TC_TPG3, \$TC_TPG4.....	1025
18.1.3.5	\$TC_TPG5.....	1026
18.1.3.6	\$TC_TPG6 および\$TC_TPG7.....	1026
18.1.3.7	\$TC_TPG8.....	1026
18.1.3.8	\$TC_TPG9.....	1027
18.1.3.9	\$TC_TPG_DRSPATH および\$TC_TPG_DRSPROG.....	1027
18.1.3.10	追加パラメータの定義\$TC_TPC1 ... 10.....	1028
18.1.3.11	工具用パラメータへのアクセス.....	1028
18.1.4	平面と軸の割り当て.....	1029
18.1.5	例.....	1030
18.2	オンライン工具オフセット.....	1032

18.2.1	機能.....	1032
18.2.2	セットアップ.....	1034
18.2.3	プログラミング.....	1035
18.2.3.1	多項式機能(FCTDEF)の定義.....	1035
18.2.3.2	オンライン工具補正の連続書き込み(PUTFTOCF).....	1037
18.2.3.3	オンライン工具補正の書き込み、個別(PUTFTOC).....	1038
18.2.3.4	オンライン工具オフセットの起動/解除(FTOCON/FTOCOF).....	1039
18.2.4	補足条件.....	1040
18.2.5	例.....	1042
18.2.5.1	例:オンライン工具オフセットの連続書き込み.....	1042
18.3	オンライン工具径補正.....	1044
18.4	研削用工具監視.....	1045
18.4.1	機能.....	1045
18.4.2	セットアップ.....	1047
18.4.3	プログラミング.....	1047
18.4.3.1	研削用工具監視の有効化/無効化(TMON、TMOF).....	1047
18.5	砥石周速度一定(GWPS).....	1048
18.5.1	機能.....	1048
18.5.2	セットアップ.....	1050
18.5.3	プログラミング.....	1051
18.5.3.1	砥石周速度一定制御(GWPSON、GWPSOF)のスイッチオン/オフ.....	1051
18.5.4	例.....	1052
18.6	データリスト.....	1053
18.6.1	マシンデータ.....	1053
18.6.1.1	一般マシンデータ.....	1053
18.6.1.2	チャンネル別マシンデータ.....	1053
18.6.1.3	軸/主軸別マシンデータ.....	1054
18.6.2	信号.....	1054
18.6.2.1	軸/主軸からの信号.....	1054
<b>19</b>	<b>Z2:NC/PLC インタフェース信号.....</b>	<b>1055</b>
<b>A</b>	<b>付録.....</b>	<b>1057</b>
A.1	略語の一覧.....	1057
A.2	概要.....	1069
	用語.....	1071
	索引.....	1097

## 基本的な安全に関する指示事項

### 1.1 一般的な安全に関する指示事項

 <b>警告</b>
<b>安全に関する情報および残存危険性に注意しない場合の死亡の危険性</b> 関連するハードウェアの資料/文書にある安全に関する情報の遵守や存在する危険性に対する注視がなされていない場合、重大な傷害または死亡事故が発生する可能性があります。 <ul style="list-style-type: none"><li>● ハードウェアドキュメントに記載された安全に関する指示事項を遵守してください。</li><li>● リスク評価では残存危険性を考慮してください。</li></ul>

 <b>警告</b>
<b>不正なまたは変更されたパラメータ設定による機械の誤作動</b> 不正なまたは変更されたパラメータ設定により、傷害や死亡に至る機械の誤動作が発生する場合があります。 <ul style="list-style-type: none"><li>● 承認されないアクセスに対するパラメータ設定変更 (パラメータ割り付け) を保護してください。</li><li>● 適切な対策を講じることで、考えられる誤作動に対応します (例: 非常停止または非常電源遮断)。</li></ul>

### 1.2 アプリケーション例に対する保証と責任

アプリケーション例に拘束力はなく、設定、機器、または起こり得る不測の事態に関する完全性を主張するものではありません。アプリケーション例は、特定のカスタムソリューションを示したのではなく、代表的なタスクを支援することのみを目的としています。

ユーザー自身が責任を持って本製品の適切な運用を確実なものとしてください。アプリケーション例は、機器の使用、取り付け、操作、および保守を行うときの安全な取扱いに対する責任からお客様を解放するものではありません。

## 1.3 産業セキュリティ

---

### 注記

#### 産業セキュリティ

シーメンスでは、プラント、システム、機械装置およびネットワークの安全な運転をサポートする産業セキュリティ機能を備えた製品およびソリューションを提供しています。サイバー攻撃に対して、プラント、システム、機械装置およびネットワークを保護するために、総合的で最新の産業セキュリティコンセプトを実装し、継続的に維持することが必要です。当社の製品およびソリューションは、このようなコンセプトの一要素を構成するものです。

お客様には、プラント、システム、機械装置およびネットワークへの不正なアクセスを防止する責任があります。このようなシステム、機械装置およびコンポーネントは、このような接続が必要な場合にのみ、必要に応じて、十分なセキュリティ対策を講じた上で(例: ファイアウォールとネットワークの細分化)、企業ネットワークまたはインターネットに接続してください。

実装可能な産業セキュリティ対策に関する関連情報については、以下をご覧ください。

産業セキュリティ (<http://www.siemens.com/industrialsecurity>)

シーメンスの製品およびソリューションは、更にセキュリティレベルを高めるために、継続的な開発が行われています。当社では、製品の更新が利用できるようになったらすぐに適用すること、および最新の製品バージョンを使用することを強く推奨しています。サポートされていない製品バージョンの使用、最新版への更新適用失敗は、お客様へのサイバー攻撃の危険性を高めることがあります。

製品のアップデート情報を受け取るには、以下で **Siemens Industrial Security RSS Feed** を申し込んでください:

産業セキュリティ (<http://www.siemens.com/industrialsecurity>)

---

関連情報はインターネットから入手できます。

産業セキュリティ設定マニュアル (<https://support.industry.siemens.com/cs/ww/en/view/108862708>)

**ソフトウェアの誤動作による安全でない運転状態**

ソフトウェアの誤動作 (例: ウィルス, トロイの木馬, マルウェアまたはウォーム) は, 死亡, 重傷や物損に至る場合があるシステムにおける安全ではない運転状態の原因となる場合があります。

- 最新のソフトウェアを使用して下さい。
- オートメーションおよびドライブコンポーネントを, 据えつけられた機器または機械装置に対する総合的で最先端の産業セキュリティコンセプトに組み込んでください。
- 据えつけられたすべての製品を総合的な産業セキュリティコンセプトに確実に組み込むようにしてください。
- 適切な保護対策で, 例えば, ウィルススキャンで悪意のあるソフトウェアから交換可能な記憶媒体上に保存されたファイルを保護してください。
- 「ノウハウプロテクト」ドライブ機能を有効にすることで, 不正な変更からドライブを保護してください。

### 1.3 産業セキュリティ

# A4:SINUMERIK 840D sl 用デジタルとアナログ NC

## I/O

# 2

### 2.1 はじめに

#### 機能

I/O ブロックは、PROFIBUS または PROFINET で SINUMERIK 840D sl に接続できます。通常、PLC ユーザープログラムは、当該のデジタルまたはアナログ入出力をアクセスに使用します。「SINUMERIK 840D sl 用デジタルおよびアナログ NC I/O」機能によって、NC（パートプログラム、シンクロナイズドアクションおよびコンパイルサイクル）からの直接のコンパイルサイクルのシステム変数を使った I/O ブロックの入力/出力へのアクセスが可能になります。以降ではこの I/O を **NC I/O** と呼びます。

互換性の理由から、以下の 3 つの機能を使用できます。

1. PLC を使わない直接 I/O アクセス (ページ 66)  
PLC を避け、直接 NC から I/O ブロックの入力/出力の制御内部イメージにアクセスできます。  
これは、数量構成および応答時間において現時点で最高の特性を発揮する機能です。
2. PLC 経由の直接 I/O アクセス (ページ 60)  
I/O ブロックの入力/出力の読み取りおよび書き込み要求は、NC からインターフェース経由で PLC に書き込まれます。この場合、PLC でアラームがトリガされます。PLC からの要求は、アラーム処理の一部として処理されます。  
数量構成から、この機能は上記で説明したものと同等です。ただし、応答時間では劣ります。
3. PLC 経由の間接 I/O アクセス (ページ 35)  
I/O ブロックの入力/出力の読み取りおよび書き込み要求は、NC からインターフェース経由で PLC に書き込まれます。これらの要求は、OB1 サイクルで周期的に処理されます。  
この機能は、数量構成および応答時間において最低の特性となります。

#### 必要条件

- NC I/O の PROFIBUS/PROFINET I/O モジュールが接続され、使用準備が整っていること。
- SIMATIC S7 Manager または HW Config を使って NC I/O の PROFIBUS/PROFINET I/O モジュールの HW 構成が完了し、PLC に読み込み済みであること。

## 2.1 はじめに

### モニタリング

NC I/O に対して下記の監視機能が有効です。

- ランプアップ。
  - PLC で検出された I/O とマシンデータでパラメータが割り付けられた NC I/O の一貫性のチェック。
- サイクリック運転。
  - 補間周期でのライフサイン監視
  - 補間周期でのモジュール監視
  - 温度監視

エラーが発生した場合、DB10.DBX104.7「NC 準備完了」メッセージがリセットされ、メッセージが表示されます。

### 障害発生時の動作

NC I/O のデジタルとアナログ出力は、NCU のエラー発生時(例: NC Ready = 0)または停電時に、セーフ状態(OV)に切り替えられます。

### 用途

NC I/O は、たとえば以下の NC 機能が使用します。

- 1 ブロック内での複数の送り速度値または補助機能
- 最終位置からの高速退避
- 軸別に残移動距離削除
- プログラム指令分岐
- 高速 NC スタート
- アナログ キャリバ
- ポジションスイッチ信号
- パンチング/ニブリング機能
- アナログ値制御

## 2.2 PLC 経由の間接 I/O アクセス

### 2.2.1 概略説明

SINUMERIK 840D sl NCU には 3 つの I/O インターフェース(X122、X132 および X142)があります。

X142 インターフェースの 4 つのデジタル入/出力部をいわゆる高速 NC I/O として使用できます。これらは、最初のアドレスバイトおよび\$A\_IN[1...4]と\$A\_OUT[1...4]システム変数を使って読み取り/書き込みが可能です。

I/O モジュールはまた、PROFIBUS DP / MPI インタフェース X126 と X136 に接続できます。これによって、デジタルとアナログ NC 入力/出力の数はそれぞれ、**デジタル 32** および **アナログ 8** ずつ拡張できます。以下では、これらの NC 入力/出力を**外部 NC I/O**と呼びます。

表 2-1 デジタルとアナログ NC I/O の最大数

	オンボード	NC I/O	合計
デジタル入力	4	32	36
デジタル出力	4	32	36
アナログ入力	-	8	8
アナログ出力	-	8	8

#### 参照先

ハードウェアについての詳細は、以下を参照してください。

- SINUMERIK 840D sl NCU マニュアル
- SIMATIC ET 200S FC 取扱説明書

#### 下記も参照

PLC 経由の直接 I/O アクセス (ページ 60)

## 2.2.2 パラメータ設定

## マシンデータ

## 有効な NC I/O の数

有効または NC から使用可能なデジタル NC I/O のバイト数は、以下のマシンデータで設定します。

- デジタル NC の入力バイト数  
MD10350 \$MN\_FASTIO\_DIG\_NUM\_INPUTS = <数字>
- デジタル NC の出力バイト数  
MD10360 \$MN\_FASTIO\_DIG\_NUM\_OUTPUTS = <数字>

有効または NC から使用可能なアナログ I/O の数は、以下のマシンデータで設定します。

- アナログ NC の入力数  
MD10300 \$MN\_FASTIO\_ANA\_NUM\_INPUTS = <数字>
- アナログ NC の出力数  
MD10310 \$MN\_FASTIO\_ANA\_NUM\_OUTPUTS = <数字>

I/O は、実際にハードウェアに存在する必要はありません。この場合、信号状態またはバイナリのアナログ値は、NC 内部で定義された方法で「ゼロ」に設定されます。

I/O 入力の値は、NC が読み取る前に PLC ユーザープログラムから変更できます。

## スロットアドレス

デジタル I/O のアドレス指定。

- 外部デジタル入力の HW 割り当て  
MD10366 \$MN\_HW\_ASSIGN\_DIG\_FASTIN[ <n> ] = <アドレス<sub>H</sub>>
- 外部デジタル出力の HW 割り当て  
MD10368 \$MN\_HW\_ASSIGN\_DIG\_FASTOUT[ <n> ] = <アドレス<sub>H</sub>>

アナログ I/O のアドレス指定。

- 外部アナログ入力の HW 割り当て  
MD10362 \$MN\_HW\_ASSIGN\_ANA\_FASTIN[ <n> ] = <アドレス<sub>H</sub>>
- 外部アナログ出力の HW 割り当て  
MD10364 \$MN\_HW\_ASSIGN\_ANA\_FASTOUT[ <n> ] = <アドレス<sub>H</sub>>

<n>: 外部デジタル I/O バイト(0 ... 3)または外部アナログ I/O(0 ... 7)をアドレス指定するためのインデックス

<アドレス>: 05 00 xxxx の PROFIBUS/PROFINET モジュールのロットアドレス

05 PROFIBUS/PROFINET モジュールの識別子

00 固定値として割り当て

xxxx<sub>H</sub> スロットの 16 進法の論理先頭アドレス

0000 = 有効なスロットなし

PLC プロセスイメージ内の論理先頭アドレスでは、下記を考慮してください(参照先をご覧ください):

- 入力スロット:NC からの読み取りも可能
- 出力スロット。NC からの書き込み禁止 ⇒ NC 起動後にメッセージ

#### 参照先

NCU 7x0.3 PN, NCU 7x0.3B PN マニュアル、セクション「技術仕様」 > サブセクション「PLC」 > 「プロセスイメージのサイズ」  
詳細情報はアドレス (<http://support.automation.siemens.com/WW/view/de/54058408>)を参照してください。

#### アナログ入力/出力の加重係数

使用されているアナログ I/O モジュールの AD または DA コンバータに対して各アナログ NC を調整するために加重係数を使用できます。

- アナログ NC 入力の荷重係数(「アナログ NC 入力 (ページ 47)」を参照)。  
MD10320 \$MN\_FASTIO\_ANA\_INPUT\_WEIGHT[<n>]
- アナログ NC 出力の荷重係数(「アナログ NC 出力 (ページ 50)」を参照)。  
MD10330 \$MN\_FASTIO\_ANA\_OUTPUT\_WEIGHT[<n>]

<n>: 外部アナログ I/O のアドレス指定のインデックス(0 ... 7)

#### NC 機能への割り当て

複数の NC 機能が I/O を必要とします。これらの機能に使用される I/O の割り当ては、マシンデータによって機能別におこなわれます。たとえば、「1 ブロック複数送り速度」機能では以下のマシンデータが使用されます。

2.2 PLC 経由の間接 I/O アクセス

MD21220 \$MC\_MULTFEED\_ASSIGN\_FASTIN = <バイトアドレス>

<バイトアドレス>	デジタル NC I/O	
0	なし	
1	1 ... 4	オンボード入力/出力
	5 ... 8	ハードウェアなしの NC 出力
2	9 ... 16	外部 NC I/O
3	17 ... 24	外部 NC I/O
4	25 ... 32	外部 NC I/O
5	33 ... 40	外部 NC I/O
128	コンパレータバイト 1 の入力 1 ~ 8	
129	コンパレータバイト 2 の入力 9 ~ 16	

注記

重複した割り付け

入力の複数割り付けは、不正なパラメータ割り付けとは考慮されません。

出力の複数割り付けは、不正なパラメータ割り付けと考慮されます。これらは起動中にチェックされ、メッセージで示されます。

例:外部 I/O のハードウェア割り付け

PROFIBUS モジュールのデジタル I/O の読み取り/書き出し用に 2 入力バイトおよび 1 出力バイトがパラメータの割り付けされます。

有効な NC I/O のバイト数

MD10350 ; 2 デジタル入力バイト + 1 オンボードバ

\$MN\_FASTIO\_DIG\_NUM\_INPUTS = 2 + 1 = イト

3

MD10360 ; 1 デジタル出力バイト + 1 オンボードバ

\$MN\_FASTIO\_DIG\_NUM\_OUTPUTS = 1 + イト

1 = 2

## ハードウェア割り当て

MD10366 ; 1 番目の I/O 入力バイトからの \$A\_IN[ 9 ] ...  
 \$MN\_HW\_ASSIGN\_DIG\_FASTIN[ 0 ] = [ 16 ]  
 'H5000200'

MD10366 ; 2 番目の I/O 入力バイトからの \$A\_IN[ 17 ] ...  
 \$MN\_HW\_ASSIGN\_DIG\_FASTIN[ 1 ] = [ 24 ]  
 'H5000201'

MD10368 ; 1 番目の I/O 出力バイトの \$A\_OUT[ 9 ] ...  
 \$MN\_HW\_ASSIGN\_DIG\_FASTOUT[ 0 ] = [ 16 ]  
 'H5000200'

マシンデータに入力された 16 進法アドレス 200<sub>H</sub> および 201<sub>H</sub> は、SIMATIC S7 Manager での設定中に割り当てられた 10 進法論理先頭アドレス 512<sub>D</sub> および 513<sub>D</sub> に対応します。

## 2.2.3 システム変数

## 入力データ

システム変数	インデックスまたは入力する数字<n>
\$A_IN[ <n> ]	1 ... 4 および 9 ... 40、デジタル NC 入力 (ページ 40)を参照
\$A_INA [ <n>]	1 ... 8、アナログ NC 入力 (ページ 47)を参照

パートプログラムからの入力データの読み取り中に、チャンネルの先読み停止がトリガされます。

## 出力データ

システム変数	インデックスまたは出力する数字<n>
\$A_OUT [ <n>]	1 ... 4 および 9 ... 40、デジタル NC 出力 (ページ 42)を参照
\$A_OUTA [ <n>]	1 ... 8、アナログ NC 出力 (ページ 50)を参照

パートプログラムからの出力データの読み取り中に、チャンネルの先読み停止がトリガされます。

## 2.2.4 コンパレータ入力

I/O 入力に加え、16 個の制御用内部コンパレータ入力を使用できます。

## 2.2 PLC 経由の間接 I/O アクセス

コンパレータ入力の現在の信号状態は、アナログ I/O 入力とセッティングデータで指定された検出値との比較に基づきます。

「コンパレータ入力 (ページ 56)」を参照してください。

### 2.2.5 デジタル NC I/O

#### 2.2.5.1 デジタル NC 入力

##### 機能

システム変数 `$A_IN` を使用して、NC プログラムまたはシンクロナイズドアクションにデジタル NC 入力の値を読み込むことができます。読み込まれた値は、NC/PLC インタフェース信号によって影響を受ける場合があります。

##### 用途例

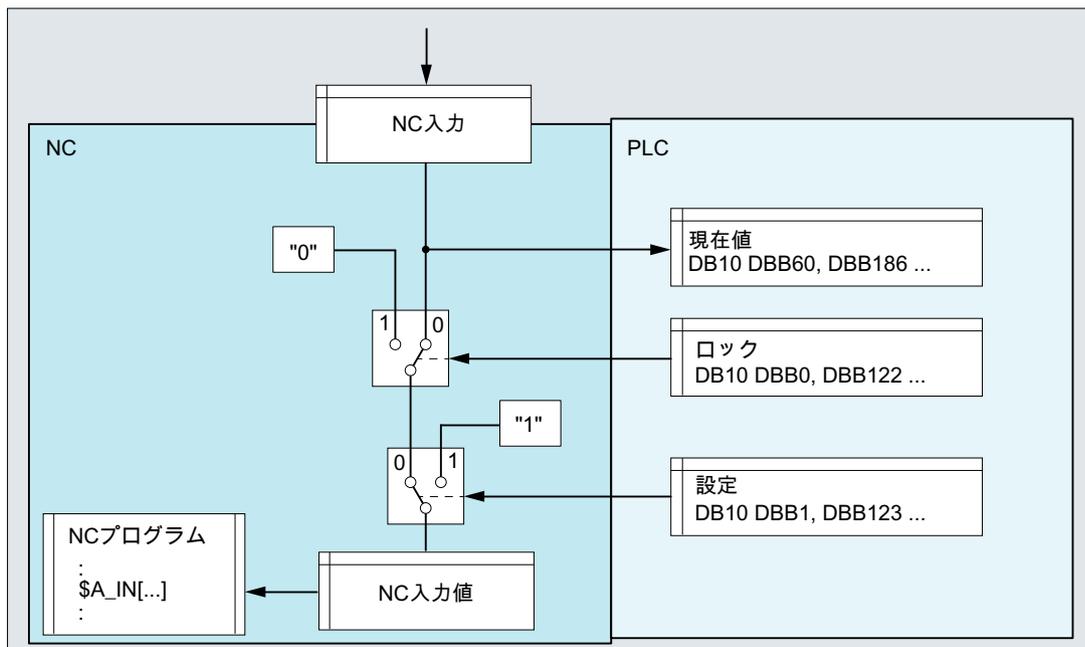
たとえば、次の NC 機能でデジタル NC 入力を使用できます。

- 位置決め軸の残移動距離削除
- ブロック終了時の高速プログラム指令分岐
- 読み込み禁止のプログラム指令
- 1 ブロック内で複数の送り速度

##### 参照先

『機能マニュアル、シンクロナイズドアクション』

## NC/PLC インタフェース信号



## 現在値

NC 入力の現在値は、現在値インタフェース経由で PLC ユーザープログラムに読み込むことができます。

## 注記

## 異なる値

「現在値」インタフェースの値は、考えられるその後の影響の結果として、システム変数 \$A\_IN を使って読み込まれる NC 入力で使用可能な値とは異なる値になる場合があります。

## ロック

インタフェースにビットを設定した場合は、対応する入力に対して値 0 が伝送されます。

## セット

インタフェースにビットを設定した場合は、対応する入力に対して値 1 が伝送されます。

## 一覧

ロック	セット	現在値	NC 入力
DBX0.0 - 7	DBX1.0 - 7	DBX60.0 - 7	1 - 8
DBX122.0 - 7	DBX123.0 - 7	DBX186.0 - 7	9 - 16

## 2.2 PLC 経由の間接 I/O アクセス

ロック	セット	現在値	NC 入力
DBX124.0 - 7	DBX125.0 - 7	DBX187.0 - 7	17 - 24
DBX126.0 - 7	DBX127.0 - 7	DBX188.0 - 7	25 - 32
DBX128.0 - 7	DBX129.0 - 7	DBX189.0 - 7	33 - 40

## 一般条件

ウォーム再起動およびチャネルリセット後の応答

ウォーム再起動およびチャネルリセット後も変更されずに、ハードウェア入力の値が伝送されます。

## 2.2.5.2 デジタル NC 出力

## 機能

システム変数 \$A\_OUT を使用して、NC プログラムまたはシンクロナイズドアクションにデジタル NC 出力の値を書き込むことができます。書き込まれた値は、NC/PLC インタフェース信号によって影響を受ける場合があります。

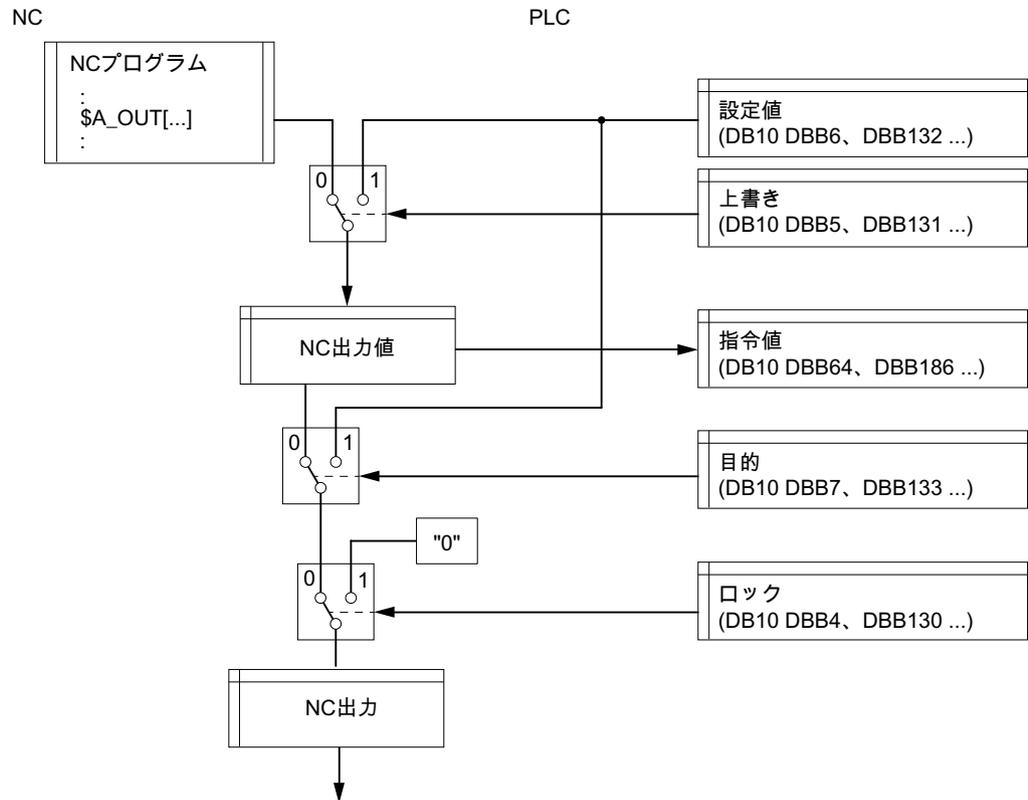
## 用途例

- ポジションスイッチ信号(「N3: ポジションスイッチ、ポジションスイッチサイクル - 840D sl のみ (ページ 731)」の章を参照してください)
- コンパレータ信号の出力

## 参照先

『機能マニュアル、シンクロナイズドアクション』

## NC/PLC インタフェース信号



## 上書き

インタフェースでビットを設定している場合は、システム変数\$A\_OUTによって書き込まれた値ではなく、PLC ユーザープログラムから指定された設定値が、対応する出力として使用されます。そして、システム変数\$A\_OUTによって書き込まれた値は失われます。

インタフェースでビットがリセットされた場合は、ハードウェア出力で現在値が保持されます。

## 設定値

設定値を使用して、PLC ユーザープログラムは、定義された出力値を指定できます。設定値を有効にするには、「上書き」または「ターゲット」のインタフェースを使用して有効化する必要があります。

2.2 PLC 経由の間接 I/O アクセス

**指令値**

実際の NC 出力値は、指令値を使用して PLC ユーザープログラムに読み込むことができます。

**注記**

**異なる値**

「指令値」インタフェースの値は、考えられるその後の影響の結果として、NC 出力で使用可能な値とは異なる場合があります。

**目的**

インタフェースでビットを設定している場合は、NC 出力値ではなく、PLC ユーザープログラムによって指定された設定値が、対応する出力として使用されます。実際の NC 出力値は保持されます。

インタフェースでビットがリセットされた場合は、最後の NC 出力値が再び有効になります。

**ロック**

インタフェースにビットを設定した場合は、対応する出力に対して値 0 が出力されます。

**一覧**

ロック	上書き	設定値	目的	指令値	NC 出力
DBX4.0 - 7	DBX5.0 - 7	DBX6.0 - 7	DBX7.0 - 7	DBX64.0 - 7	1 - 8
DBX130.0 - 7	DBX131.0 - 7	DBX132.0 - 7	DBX133.0 - 7	DBX190.0 - 7	9 - 16
DBX134.0 - 7	DBX135.0 - 7	DBX136.0 - 7	DBX137.0 - 7	DBX191.0 - 7	17 - 24
DBX138.0 - 7	DBX139.0 - 7	DBX140.0 - 7	DBX141.0 - 7	DBX192.0 - 7	25 - 32
DBX142.0 - 7	DBX143.0 - 7	DBX144.0 - 7	DBX145.0 - 7	DBX193.0 - 7	33 - 40

**一般条件**

**ハードウェアなしの NC 出力**

(MD10360 \$MN\_FASTIO\_DIG\_NUM\_OUTPUTS)と定義された NC 出力が書き込まれている場合は、アラームが表示されませんが、対応するハードウェアは使用できません。指令値は、PLC ユーザープログラムを使用して読み込むことができます。

**プログラム終了リセット/チャンネルリセットの動作**

すべての NC 出力は、PLC ユーザープログラムから、「ターゲット」または「ロック」のインタフェースを使用して、プログラム終了またはチャンネルリセットで特定のアプリケーションに応じて、定義された方法で設定されます。

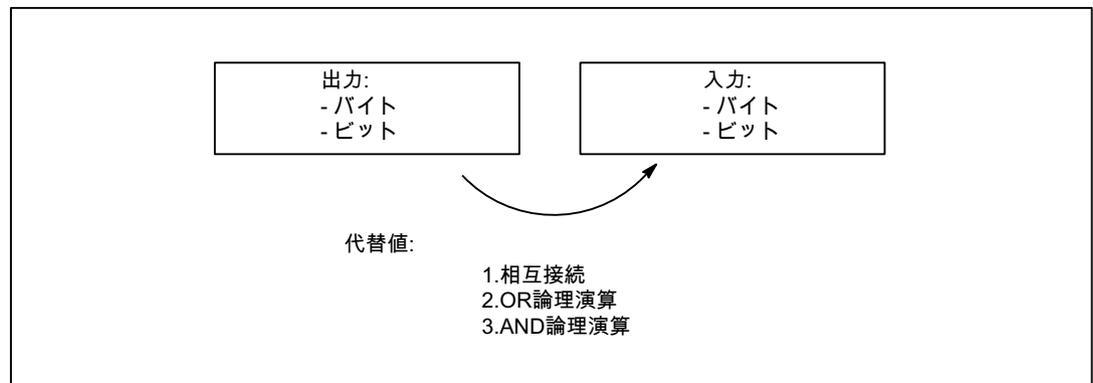
**ウォーム再起動の動作**

すべての NC 出力は、ウォーム再起動後に「0」に設定されます。PLC ユーザープログラムから上書きを使用するか、ビットパターンを設定して、すべての NC 出力を特定のアプリケーションに応じて設定できます。

**2.2.5.3 高速デジタル I/O の接続と論理演算****機能**

高速 NC I/O 入力は、高速出力信号状態に応じてソフトウェアで設定できます。

概要:

**接続**

NC I/O 高速入力は割り付けされた高速出力の信号状態に設定されます。

**OR 演算**

NC I/O 高速入力では、割り付けされた入力信号と出力信号の OR 演算結果の信号状態が適用されます。

### AND 演算

NC I/O 高速入力では、割り付けされた入力信号と出力信号の AND 演算結果の信号状態が適用されます。

### 特別な事例

- 複数の出力ビットを同じ入力ビットに割り付けた場合、最も大きな MD インデックス番号を持つビットが有効になります。
- 存在していないか、無効の入力または出力を指定した場合、アラームは発生せずに割り付けが無視されます。マシンデータの入力によって、NC I/O の有効バイトのチェックをおこないます。

```
MD10350 $MN_FASTIO_DIG_NUM_INPUTS
MD10360 $MN_FASTIO_DIG_NUM_OUTPUTS
```

### 割り付けの定義

この割り付けは、マシンデータによって指定します。

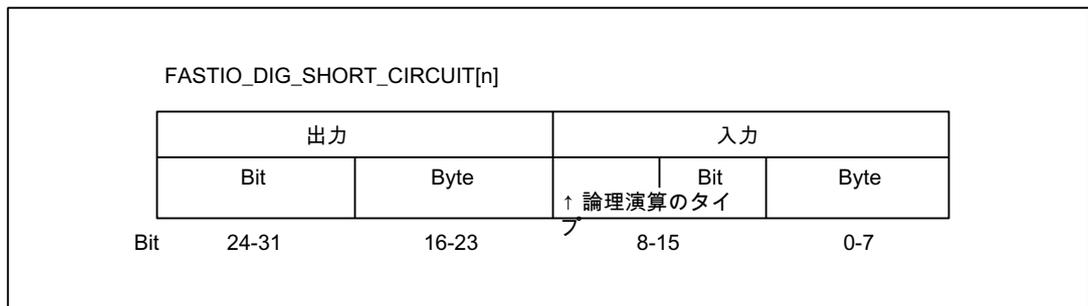
```
MD10361 $MN_FASTIO_DIG_SHORT_CIRCUIT[n]
```

n: 値 0 ~ 9 を指定できるため、最大で 10 個の割り付けを指定できます。

出力と入力のバイト、およびビットを指定するために、2 つの 16 進文字が設けられています。

入力ビット 12 ~ 15 に 0、A、および B を指定すると、次の論理演算の結果になります。

- 0 接続
- A AND 演算
- B OR 演算



**例****接続:**

MD10361 \$MN\_FASTIO\_DIG\_SHORT\_CIRCUIT = '04010302H'

出力 4、バイト 1 を下記に接続します

入力 3、バイト 2

**AND 演算:**

MD10361 \$MN\_FASTIO\_DIG\_SHORT\_CIRCUIT = '0705A201H'

出力 7、バイト 5 と下記を AND 演算します

入力 2、バイト 1

**OR 演算:**

MD10361 \$MN\_FASTIO\_DIG\_SHORT\_CIRCUIT = '0103B502H'

出力 1、バイト 3 と下記を OR 演算します

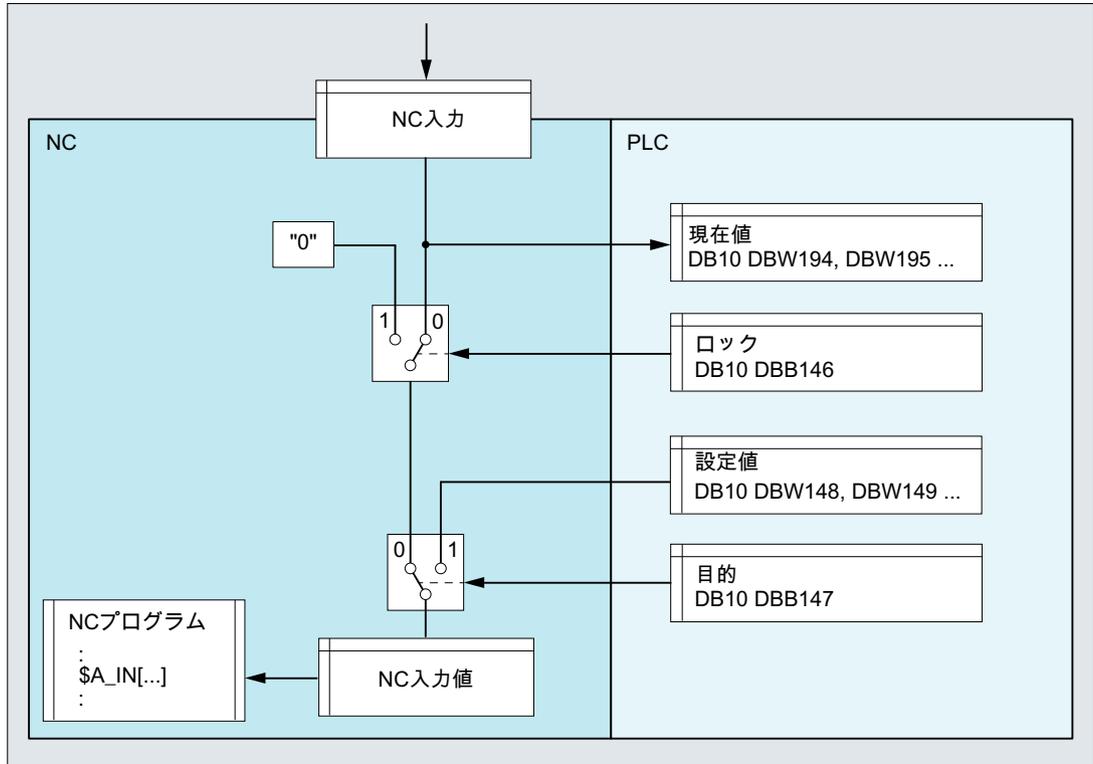
入力 5、バイト 2

**2.2.6 アナログ NC I/O****2.2.6.1 アナログ NC 入力****機能**

システム変数 \$A\_INA を使用して、NC プログラムまたはシンクロナイズドアクションにアナログ NC 入力の値を読み込むことができます。読み込まれた値は、NC/PLC インタフェース信号によって影響を受ける場合があります。

バイナリアナログ値の表示

NC/PLC インタフェース信号



現在値

NC 入力の現在値は、現在値インタフェース経由で PLC ユーザープログラムに読み込むことができます。

現在値は、2 の補数の固定小数点数(符号付き 16 ビット値)として表されます。

「アナログ I/O 値の表現 (ページ 54)」を参照してください。

注記

異なる値

「現在値」インタフェースの値は、考えられるその後の影響の結果として、システム変数 \$A\_INA を使って読み込まれる NC 入力で使用可能な値とは異なる値になる場合があります。

ロック

インタフェースにビットを設定した場合は、対応する入力に対して値 0 が伝送されます。

### 設定値

設定値を使用して、PLC ユーザープログラムは、定義された入力値を指定できます。設定値を有効にするには、「上書き」インタフェースを使用して有効化する必要があります。

設定値は、2 の補数の固定小数点数(符号付き 16 ビット値)として指定する必要があります。

「アナログ I/O 値の表現 (ページ 54)」を参照してください。

### 目的

インタフェースにビットを設定した場合は、対応する入力に対して、関連する「設定値」が伝送されます。

### 一覧

ロック	目的	設定値	現在値	NC 入力
DBX146.0 - 7	DBX147.0 - 7	DBW148 - 162	DBW194 - 209	1 - 8

## マシンデータ

### 加重係数

加重係数を使用して、使用している I/O モジュールの DA コンバータに適切なパラメータを設定し、各アナログ NC 入力用に調整することができます。

MD10320 \$MN\_FASTIO\_ANAINPUT\_WEIGHT[<出力>]

### 注記

#### ハードウェアを装備しないアナログ NC 入力

32767 の加重係数の適用で、NC プログラムと PLC ユーザープログラムのために数値化されたアナログ値は同じです。したがって、NC プログラムと PLC ユーザープログラムとの間の 1:1 の通信に NC 出力を使用できます。

## 必要条件

### ハードウェアを装備しないアナログ NC 入力

(MD10300 \$MN\_FASTIO\_ANA\_NUM\_INPUTS)と定義された NC 出力が書き込まれている場合は、アラームが表示されませんが、対応するハードウェアは使用できません。現在値は、PLC ユーザープログラムを使用して読み込むことができます。

## 2.2 PLC 経由の間接 I/O アクセス

### ウォーム再起動、プログラム終了リセットおよびチャネルリセットの動作

ウォーム再起動、プログラム終了リセットまたはチャネルリセットの後、すべての NC 入力に対して、存在するアナログ値が伝送されます。すべての NC 入力は、PLC ユーザープログラムから特定のアプリケーションに応じて設定できます。

### 2.2.6.2 アナログ NC 出力

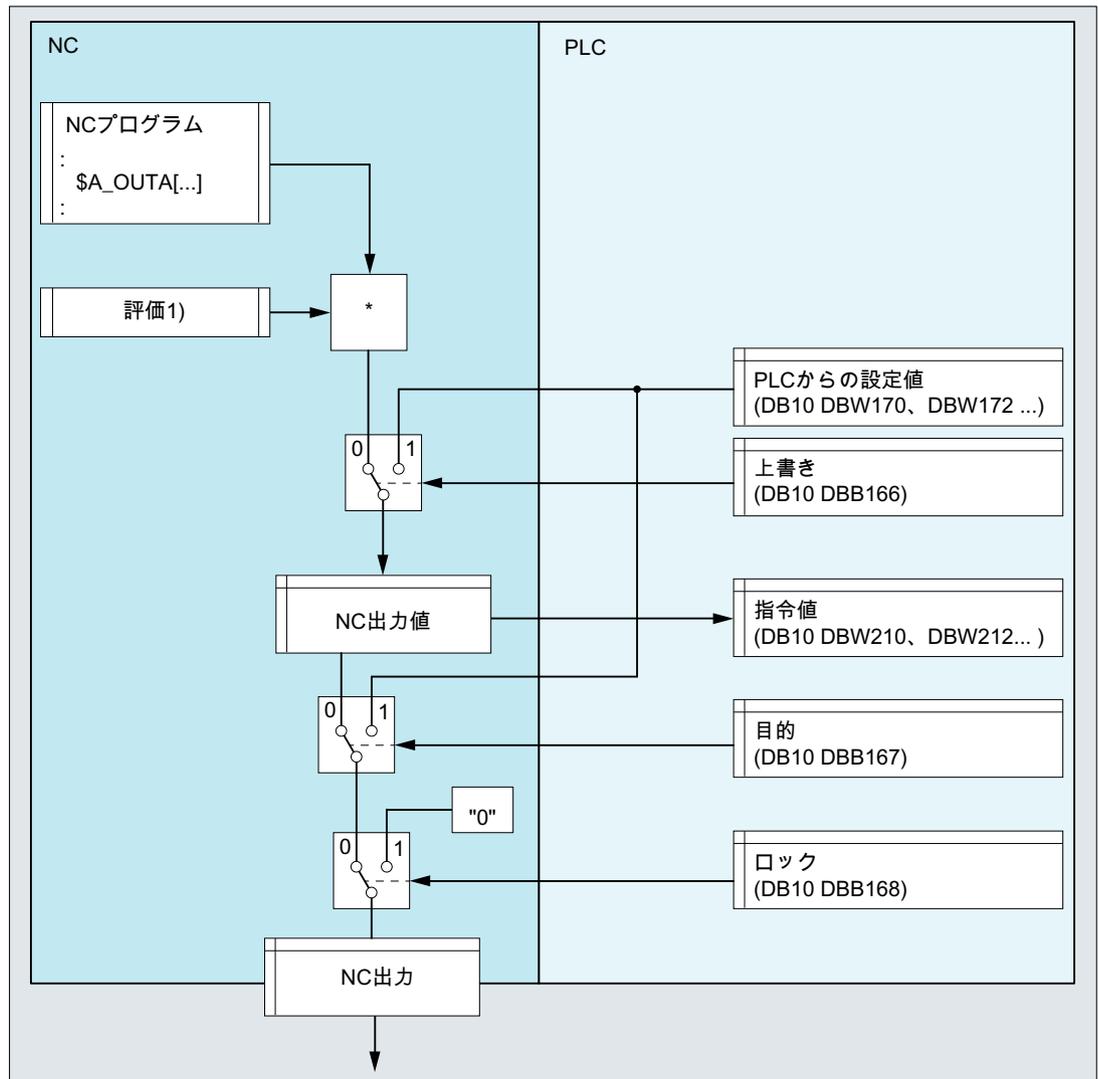
#### 機能

システム変数 `$A_OUTA` を使用して、NC プログラムまたはシンクロナイズドアクションにアナログ NC 出力の値を書き込むことができます。書き込まれた値は、マシンデータおよび NC/PLC インタフェース信号によって影響を受ける場合があります。

#### 参照先

『機能マニュアル、シンクロナイズドアクション』

## NC/PLC インタフェース信号



## 1) MD10330 \$MN\_FASTIO\_ANA\_OUTPUT\_WEIGHT[&lt;出力&gt;]

## 設定値

設定値を使用して、PLC ユーザープログラムは、定義された出力値を指定できます。設定値を有効にするには、「上書き」または「ターゲット」のインターフェースを使用して有効化する必要があります。

設定値は、2 の補数の固定小数点数(符号付き 16 ビット値)として指定する必要があります。

「アナログ I/O 値の表現 (ページ 54)」を参照してください。

2.2 PLC 経由の間接 I/O アクセス

**上書き**

ビットを設定している場合は、システム変数\$A\_OUTAによって書き込まれた値ではなく、PLC ユーザープログラムから指定された設定値が、対応する出力として使用されます。そして、システム変数\$A\_OUTAによって書き込まれた値は失われます。

ビットがリセットされた場合は、対応する出力に対して、ハードウェア出力の現在値が保持されます。

**目的**

ビットを設定している場合は、NC 出力値ではなく、PLC ユーザープログラムによって指定された設定値が、対応する出力として使用されます。実際の NC 出力値は保持されます。

ビットがリセットされた場合は、対応する出力に対して最後の NC 出力値が再び有効になります。

**ロック**

インタフェースにビットを設定した場合は、対応する NC 出力で値 0 が出力されます。

**指令値**

実際の NC 出力値は、指令値を使用して PLC ユーザープログラムに読み込むことができます。

指令値は、2 の補数の固定小数点数(符号付き 16 ビット値)として表されます。

「アナログ I/O 値の表現 (ページ 54)」を参照してください。

**注記**

**異なる値**

「指令値」インタフェースの値は、考えられるその後の影響の結果として、NC 出力で使用可能な値とは異なる場合があります。

**一覧**

ロック	上書き	設定値	目的	指令値	NC 出力
DBX168.0 - 7	DBX166.0 - 7	DBW170 - 184	DBX167.0 - 7	DBW210 - 225	1 - 8

## マシンデータ

## 加重係数

加重係数を使用して、使用している I/O モジュールの DA コンバータに適切なパラメータを設定し、各アナログ NC 出力用に調整することができます。

MD10330 \$MN\_FASTIO\_ANA\_OUTPUT\_WEIGHT[<出力>]

プログラムコード	コメント
; 前提条件:アナログ値の範囲 10 V	
; パラメータ設定:MD10330 \$MN_FASTIO_ANA_OUTPUT_WEIGHT[<出力>] = 10000	
\$A_OUTA[<出力>] = 9500	; NC 出力:9.5 V
\$A_OUTA[<出力>] = -4120	; NC 出力 -4.12 V

## 注記

## ハードウェアを装備しないアナログ NC 出力

32767 の加重係数の適用で、NC プログラムと PLC ユーザープログラムのために数値化されたアナログ値は同じです。したがって、NC プログラムと PLC ユーザープログラムの中の 1:1 の通信に NC 出力を使用できます。

## 必要条件

## ハードウェアを装備しないアナログ NC 出力

(MD10310 \$MN\_FASTIO\_ANA\_NUM\_OUTPUTS)と定義された NC 出力が書き込まれている場合は、アラームが表示されませんが、対応するハードウェアは使用できません。指令値は、PLC ユーザープログラムを使用して読み込むことができます。

## ウォーム再起動の動作

すべての NC 出力は、ウォーム再起動後に「0」に設定されます。すべての NC 入力、PLC ユーザープログラムから特定のアプリケーションに応じて設定できます。

## プログラム終了リセット/チャンネルリセットの動作

すべての NC 出力は、PLC ユーザープログラムから、「ターゲット」または「ロック」のインタフェースを使用して、プログラム終了またはチャンネルリセットで特定のアプリケーションに応じて、定義された方法で設定されます。

2.2.6.3 アナログ I/O 値の表現

数値化されたアナログ値は、NC/PLC インタフェースで、固定小数点数値(符号付き 16 ビット)として 2 の補数で表されます。

ビット番号	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
意味	S G	$2^{14}$	$2^{13}$	$2^{12}$	$2^{11}$	$2^{10}$	$2^9$	$2^8$	$2^7$	$2^6$	$2^5$	$2^4$	$2^3$	$2^2$	$2^1$	$2^0$

SG:符号

最小値	最大値
$-32768_D$	$32767_D$
$8000_H$	$7FFF_H$

ステップ値

分解能が 16 ビットで公称レンジが ±10 V の場合、ステップ値は次のようになります。

$$20 \text{ V} / 2^{16} = 20 \text{ V} / 65536 \approx 0.305 \text{ mV}$$

分解能 < 16 ビット

アナログモジュールの分解能が符号を含め 16 ビットより低い場合、デジタル化されたアナログ値は、ビット 14 から開始してインターフェースに入力されます。未使用の最下位ビットの桁には「0」が入力されます。

14 ビット分解能

分解能が符号付き 14 ビットで公称レンジが ±10 V の場合、ステップ値は次のようになります。

$$20 \text{ V} / 2^{14} = 20 \text{ V} / 16384 \approx 1.22 \text{ mV}$$

ビット 0 ~ 1 は常に「0」です。

12 ビット分解能

分解能が符号付き 12 ビットで公称レンジが ±10 V の場合、ステップ値は次のようになります。

$$20 \text{ V} / 2^{12} = 20 \text{ V} / 4096 \approx 4.88 \text{ mV}$$

ビット 0 ～ 3 は常に「0」です。

さまざまな分解能の最大値の表示

ビット番号	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
ビットの有効桁	S G	$2^{14}$	$2^{13}$	$2^{12}$	$2^{11}$	$2^{10}$	$2^9$	$2^8$	$2^7$	$2^6$	$2^5$	$2^4$	$2^3$	$2^2$	$2^1$	$2^0$
16 ビット分解能 $32767_D = 7FFF_H$	0	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1
14 ビット分解能 $8191_D = 1FFF_H$	0	1	1	1	1	1	1	1	1	1	1	1	1	1	0	0
12 ビット分解能 $2047_D = 7FF_H$	0	1	1	1	1	1	1	1	1	1	1	1	0	0	0	0

#### 注記

使用するアナログ入出力モジュールのデータ(分解能、公称レンジ)は、対象モジュールの取扱説明書から入手できます。

#### 例

分解能が符号付き 14 ビットで公称レンジが  $\pm 10$  V の場合のアナログ値のデジタル表示。

##### 例 1: アナログ値 = 9.5 V

数値化されたアナログ値(10 進数):  $9.5 \text{ V} / 20 \text{ V} * 16384 = 7782$

数値化されたアナログ値 14 ビット(2 進数): 01 1110 0110 0110

数値化されたアナログ値 16 ビット(2 進数): 0111 1001 1001 1000

数値化されたアナログ値 16 ビット(16 進数):  $7998_H$

##### 例 2: アナログ値 = -4.12 V

数値化されたアナログ値(10 進数):  $-4.12 \text{ V} / 20 \text{ V} * 16384 = -3375$

数値化されたアナログ値 14 ビット(2 進数): 11 0010 1101 0001

数値化されたアナログ値 16 ビット(2 進数): 1100 1011 0100 0100

数値化されたアナログ値 16 ビット(16 進数):  $CB44_H$

## 2.2.7 コンパレータ入力

### 機能

デジタルとアナログの NC 入力の他に、2つの内部コンパレータ入力バイト(それぞれ、8つのコンパレータ入力を持ちます)が使用できます。コンパレータ入力の信号状態は、高速アナログ入力されたアナログ値と、セッティングデータでパラメータ設定されたしきい値の比較に基づいて生成されます。

$\$A\_INCO[<n>]$ システム変数により、コンパレータ入力[ $<n>$ ]の信号状態(つまり比較の結果)をパートプログラムで直接スキャンできます。

インデックス $<n>$ の用途:

$<n> = 1 \dots 8$	コンパレータバイト 1
$<n> = 9 \dots 16$	コンパレータバイト 2

### 用語

この説明では、**コンパレータ入力**(インデックス $<n>$ 、 $<n>$ の数値の範囲:1 ~ 8 または 9 ~ 16)および**コンパレータ入力ビット**(インデックス $<b>$ 、 $<b>$ の数値の範囲:0 ~ 7)という用語を使用します。

これらの用語の関連は次のとおりです。

$<n> = 1 \dots 8:$	コンパレータ入力 $<n>$ は、コンパレータ入力ビット $<b> = <n> - 1$ に相当します。
$<n> = 9 \dots 16:$	コンパレータ入力 $<n>$ は、コンパレータ入力ビット $<b> = <n> - 9$ に相当します。

例:コンパレータ入力 1 は、コンパレータ入力ビット 0 に相当します。

### アナログ入力の割り付け

次のマシンデータは、コンパレータバイト 1 の入力ビット $<b>$ にアナログ入力を割り付けるために使用します。

MD10530 \$MN\_COMPAR\_ASSIGN\_ANA\_INPUT\_1[ $<b>$ ]

例:

MD10530 \$MN\_COMPAR\_ASSIGN\_ANA\_INPUT\_1[0] = 1

MD10530 \$MN\_COMPAR\_ASSIGN\_ANA\_INPUT\_1[1] = 1

MD10530 \$MN\_COMPAR\_ASSIGN\_ANA\_INPUT\_1[7] = 7

アナログ入力 1 は、コンパレータバイト 1 の入力ビット 0 と 1 を操作します。

アナログ入力 7 は、コンパレータバイト 1 の入力ビット 7 を操作します。

コンパレータバイト 2 のアナログ入力は、次のマシンデータに割り当てられます。

MD10531 \$MN\_COMPAR\_ASSIGN\_ANA\_INPUT\_2[<b>

### コンパレータの設定

コンパレータバイト 1 または 2 の個々のビット(0 ~ 7)の設定は、次のマシンデータによって決定します。

MD10540 \$MN\_COMPAR\_TYPE\_1 (コンパレータバイト 1 のパラメータ設定)

または

MD10541 \$MN\_COMPAR\_TYPE\_2 (コンパレータバイト 2 のパラメータ設定)

下記の設定が可能です。

- 比較形式のマスク(ビット 0 ~ 7)
  - 各コンパレータ入力ビットについて、比較条件タイプを定義します。
    - ビット = 1: 以下の場合、関連コンパレータ入力ビットは「1」に設定されます。  
アナログ値  $\geq$  検出値
    - ビット = 0: 以下の場合、関連コンパレータ入力ビットは「0」に設定されます。  
アナログ値  $<$  検出値
- デジタル NC 出力(ビット 16 ~ 23)によるコンパレータ入力バイトの出力
  - コンパレータビットは、バイトごとにデジタル NC 出力バイトによって直接出力できます。このためには、使用するデジタル NC 出力バイトのこのバイト(ビット 16 ~ 23)の指定が必要です。
    - バイト = 0: デジタル NC 出力による出力はありません。
    - バイト = 1: デジタルオンボード NC 出力 9 ~ 16 による出力
    - バイト = 2: 外部デジタルオンボード NC 出力 17 ~ 24 による出力
    - バイト = 3: 外部デジタルオンボード NC 出力 25 ~ 32 による出力
    - バイト = 4: 外部デジタルオンボード NC 出力 33 ~ 40 による出力

## 2.2 PLC 経由の間接 I/O アクセス

- コンパレータ入力バイト(ビット 24 ~ 31)を出力するための反転マスク  
各コンパレータ信号について、デジタル NC 出力で出力する信号状態を反転するかどうかを定義できます。

ビット= 1: 関連コンパレータ入力ビットは反転されません。

ビット= 0: 関連コンパレータ入力ビットは反転されます。

### しきい値

コンパレータバイト 1 または 2 の比較で使用されるしきい値は、セッティングデータとして保存してください。各コンパレータ入力ビット<b>( <b> = 0 ~ 7)に個別の検出値を入力する必要があります。

SD41600 \$SN\_COMPAR\_THRESHOLD\_1[<b>]

または

SD41601 \$SN\_COMPAR\_THRESHOLD\_2[<b>]

### デジタル NC 入力としてのコンパレータ信号

デジタル NC 入力に応じて処理されるすべての NC 機能もまた、コンパレータの信号状態によって制御できます。コンパレータバイト 1(HW バイト 128)または 2(HW バイト 129)のバイトアドレスは、NC 機能(「使用するハードウェアバイトの割り付け」)に関連するマシンデータに入力してください。

例:

「1 ブロックの複数の送り速度」 NC 機能

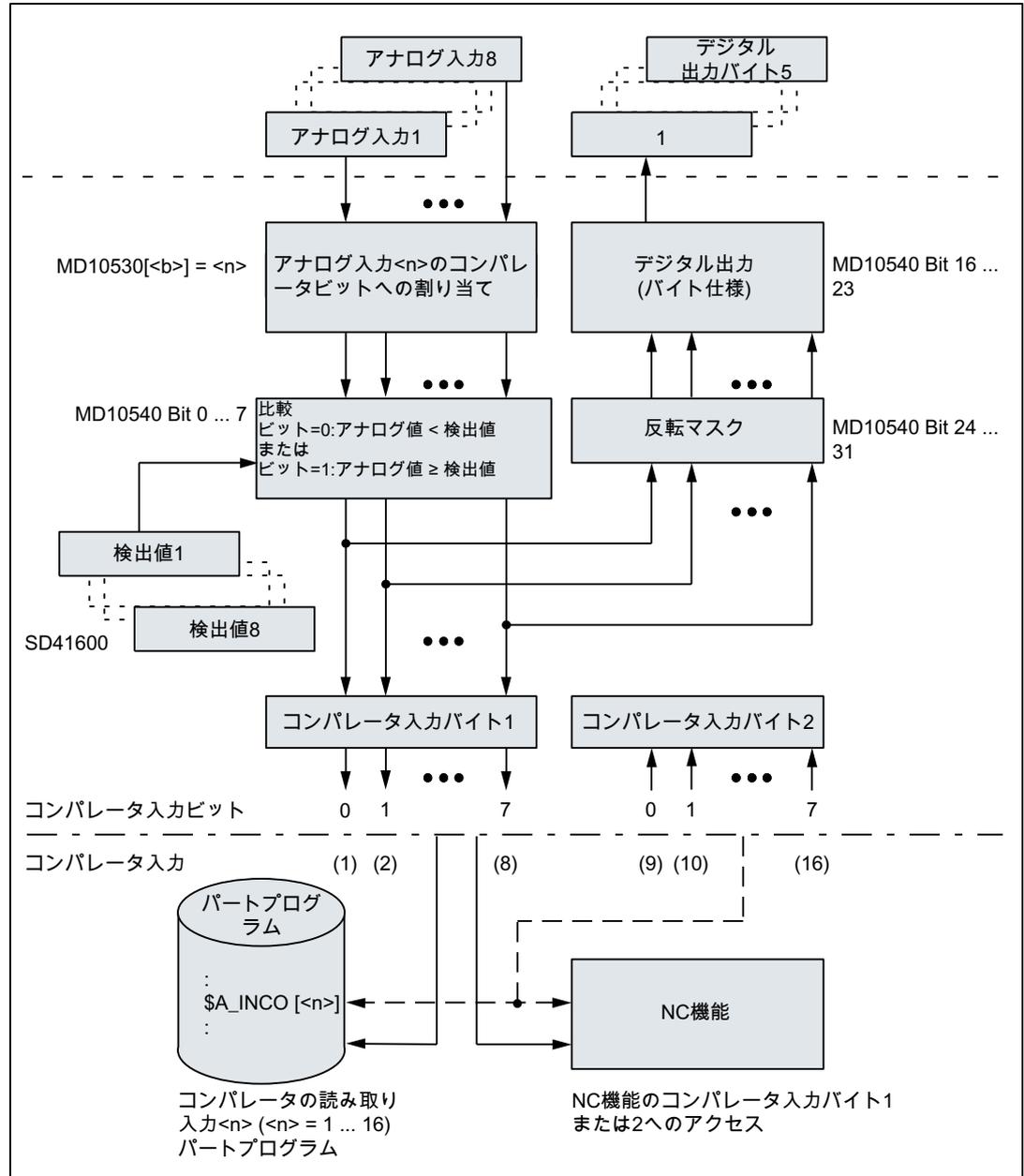
チャンネル別のマシンデータの入力:

MD21220 \$MC\_MULTFEED\_ASSIGN\_FASTIN = 129

これにより、コンパレータバイト 2 の状態に応じて、様々な送り速度値が有効になります。

機能の処理

以下の図に、コンパレータ入力バイト 1 の機能の概略の処理を示します。



## 2.3 PLC 経由の直接 I/O アクセス

### 2.3.1 パラメータ設定

#### マシンデータ

##### I/O 範囲の長さ

- NC で直接読み取られる PLC I/O 入力バイト数:  
MD10394 \$MN\_PLCIO\_NUM\_BYTES\_IN
- NC で直接書き込まれる PLC-I/O 出力バイト数:  
MD10396 \$MN\_PLCIO\_NUM\_BYTES\_OUT

##### 論理先頭アドレス

- PLC の入力 I/O からデータの読み込みが開始される論理先頭アドレス。以降のアドレス指定に使用されるオフセット(例: \$A\_PBB\_IN[<オフセット>])は、以下のマシンデータで指定された先頭アドレスを参照します。  
MD10395 \$MN\_PLCIO\_LOGIC\_ADDRESS\_IN
- PLC の入力 I/O からデータの書き込みが開始される論理先頭アドレス。以降のアドレス指定に使用されるオフセット(例: \$A\_PBB\_OUT[<オフセット>])は、以下のマシンデータで指定された先頭アドレスを参照します。  
MD10397 \$MN\_PLCIO\_LOGIC\_ADDRESS\_OUT

##### 更新時間

- \$A\_PBx\_IN によって読み取られるデータが更新される時間(以下の「伝送時間」のサブセクションを参照してください)。  
MD10398 \$MN\_PLCIO\_IN\_UPDATE\_TIME

##### 書式表示

- システム変数 \$A\_PBx\_OUT および \$A\_PBx\_IN の書式表示(以下の「保存書式の選択(リトル/ビッグエンディアン)」のサブセクションを参照してください)。  
MD10399 \$MN\_PLCIO\_TYPE\_REPRESENTATION

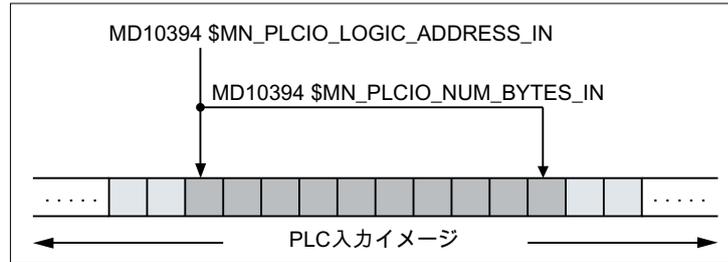
---

#### 注記

マシンデータに入力する論理 PLC I/O アドレスと伝送バイト数は、PLC ハードウェア構成に一致させてください。設定された領域では、PLC I/O の拡張構成に「アドレスの隙間」が存在することはできません。

---

### 入力領域での NC I/O のパラメータ設定の原理



### 伝送時間

- NC から出力モジュールへの出力データの伝送
  - 出力データは、現在の補間周期の最後に NC から出力モジュールに出力されます。
  - この伝送は、現在の補間周期で 1 つ以上の出力データ項目が書き込まれた場合のみ実行されます。
- 入力モジュールから NC への入力データの伝送
  - NC から PLC に入力データの更新(モジュールの入カイメージ → NC 入力データ)の周期要求が送信される時間は、以下のマシンデータで設定できます。
  - MD10398 \$MN\_PLCIO\_IN\_UPDATE\_TIME = <更新時間>
  - パラメータ設定された更新時間は、内部で補間周期の倍数に切り上げられます。更新時間= 0 の場合、要求はそれぞれの補間サイクルで PLC に伝送されます。
  - 入力データ更新の要求は、パラメータ設定された補間サイクルの最後に PLC に送信されます。
  - 更新された入力データは、早ければ次の補間サイクルで使用可能になります。

### 保存書式の選択(リトル/ビッグエンディアン)

NC と PLC 間のデータ転送において、全 NC チャンネルで合計 16 KB を使用できます。これらの領域はユーザーが管理してください(つまり、チャンネル境界を超えない場合でも、変数の重複が存在しないようにしてください)。

これらの領域での変数表示は、次のマシンデータの設定に応じて異なります。

2.3 PLC 経由の直接 I/O アクセス

MD10399 \$MN\_PLCIO\_TYPE\_REPRESENTATION = <値>

<値>	意味
0	リトルエンディアン書式(初期設定) システム変数は、リトルエンディアン書式で表示されます ⇒ 最下位アドレスに最下位バイト
1	ビッグエンディアン書式(PLC の標準書式、推奨) システム変数は、ビッグエンディアン書式で表示されます ⇒ 最下位アドレスに最上位バイト  注 ビッグエンディアン書式は、PLC および PLC I/O で通常使用される書式です。そのため、この書式を使用することをお勧めします。

2.3.2 読み取り/書き込み:システム変数

入力データ

システム変数	タイプ	オフセット<n>
\$A_PBB_IN[<n>]	バイト	0, 1, 2, ...(MD10394 \$MN_PLCIO_NUM_BYTES_IN - 1)
\$A_PBW_IN[<n>]	ワード	0, 2, 4, ...(MD10394 \$MN_PLCIO_NUM_BYTES_IN - 2)
\$A_PBD_IN[<n>]	ダブル	0, 4, 8, ...(MD10394 \$MN_PLCIO_NUM_BYTES_IN - 4)
\$A_PBR_IN[<n>]	実数	0, 4, 8, ...(MD10394 \$MN_PLCIO_NUM_BYTES_IN - 4)

パートプログラムからの読み取り中に、チャンネルの先読み停止がトリガされます。

出力データ

システム変数	タイプ	オフセット<n>
\$A_PBB_OUT[<n>]	バイト	0, 1, 2, ...(MD10396 \$MN_PLCIO_NUM_BYTES_OUT - 1)
\$A_PBW_OUT[<n>]	ワード	0, 2, 4, ...(MD10396 \$MN_PLCIO_NUM_BYTES_OUT - 2)
\$A_PBD_OUT[<n>]	ダブル	0, 4, 8, ...(MD10396 \$MN_PLCIO_NUM_BYTES_OUT - 4)
\$A_PBR_OUT[<n>]	実数	0, 4, 8, ...(MD10396 \$MN_PLCIO_NUM_BYTES_OUT - 4)

パートプログラムからの出力データの読み取り中に、チャンネルの先読み停止がトリガされます。

#### 出力データの数値の範囲

タイプ	数値の範囲	
バイト	符号付き:	-128 ... +127
	符号なし:	0 ... 255
ワード	符号付き:	-32768 ... +32767
	符号なし:	0 ... 65535
ダブル	符号付き:	-2147483648 ... +2147483647
	符号なし:	0 ... 4294967295
実数	-3.402823466*10 <sup>+38</sup> ... +3.402823466*10 <sup>+38</sup>	

### 2.3.3 必要条件

#### 複数のスロット

NC が直接使用する PLC I/O の入力または出力範囲が複数のスロットを構成する場合、スロットのアドレス範囲を途切れのない連続した範囲として設定してください。

#### NC と PLC の並列書き込み

PLC を介した NC による直接アクセスおよび PLC ユーザープログラムからの I/O 出力の並列書き込みを行うと、出力値のランダムな相互書き換えが発生します。よってこれは許可されませんが、コントローラでは防ぐことができません。

#### 応答時間

PLC I/O からデータが読み取られる時間と、パートプログラムでシステム変数によってデータが使用可能になる時間は同期しません。

#### データ伝送

- 各システム変数について出力データ項目が 1 つだけ書き込まれた場合でも、PLC I/O の出力データの出力は、常にすべてのパラメータ設定された出力データについて行われます。
- 複数のシステム変数に同時に値が割り当てられた場合、これらが同じ補間サイクル内に伝送される保証はありません。

2.3 PLC 経由の直接 I/O アクセス

2.3.4 例

2.3.4.1 PLC-I/O への書き込み

仕様

- PLC I/O で書き込まれる出力データの 10 進数の論理アドレス
  - 521:1 バイトの整数値
  - 522:2 バイトの整数値
- 出力データは、R 変数 R10 - R11 を使ってパートプログラムに書き込まれます。
- PLC ユーザープログラム (OB1)の実行の速度をわずかにだけ低下させるために、書き込みアクセスの更新サイクルは 12 ms の補間周期の 3 倍とします。
- データは、次の PLC I/O に直接出力されます。

パラメータ設定

マシンデータは次のように設定してください。

- NC I/O 出力データ領域の長さ:  $2 + 1 = 3$  バイト  
MD10396 \$MN\_PLCIO\_NUM\_BYTES\_OUT = 3
- 入力データ領域の論理先頭アドレス:521  
MD10397 \$MN\_PLCIO\_LOGIC\_ADDRESS\_OUT = 521
- 更新サイクル:I/O 入力データ → システム変数 =  $3 * 補間周期 = 3 * 12 \text{ ms} = 36 \text{ ms}$   
MD10398 \$MN\_PLCIO\_OUT\_UPDATE\_TIME =  $3 * 0.012 = 0.036$
- データ書式:ビッグエンディアン  
MD10399 \$MN\_PLCIO\_TYPE\_REPRESENTATION = 1

更新

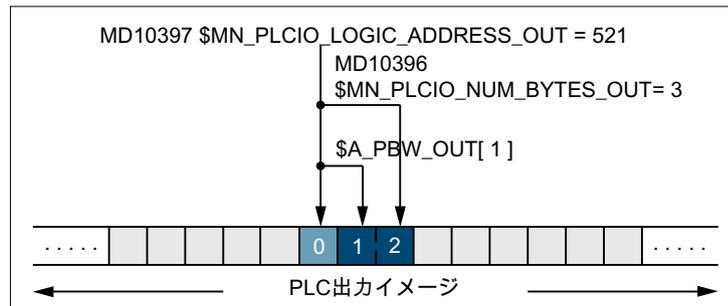
システムデータの PLC I/O への伝送は、コントローラ起動後に PLC がエラーなしで PROFIBUS/PROFINET I/O を検出するまで行われません。その後、3 回目の補間サイクル毎に伝送が行われます。

プログラミング

R 変数を使った NC I/O のシンクロナイズドアクションへの書き込み:

プログラムコード	コメント
R10=123	
R11='Habcd'	
ID=1 WHENEVER TRUE DO \$A_PBB_OUT[0]=R10	; 1 バイトの整数値
ID=2 WHEN \$AA_IW[x] >= 5 DO \$A_PBW_OUT[1]=R11	; 2 バイトの整数値

アドレス指定の例:  $\$PBW\_OUT[1] = R11$



### 2.3.4.2 PLC-I/O からの読み取り

#### 仕様

- PLC I/O で読み取られる入力データの 10 進数の論理アドレス
  - 420:2 バイトの整数値
  - 422:4 バイトの整数値
  - 426:4 バイトの REAL 値
  - 430:1 バイトの整数値
- 入力データは、パートプログラムの R 変数 R1 - R4 に格納します。
- PLC ユーザープログラム (OB1) の実行の速度をわずかに低下させるために、読み取りアクセスの更新サイクルは 12 ms の補間周期の 3 倍とします。

#### パラメータ設定

- NC I/O 入力データ領域の長さ:  $2 + 4 + 4 + 1 = 11$  バイト  
MD10394 \$MN\_PLCIO\_NUM\_BYTES\_IN = 11
- 入力データ領域の論理先頭アドレス: 420  
MD10395 \$MN\_PLCIO\_LOGIC\_ADRESS\_IN = 420
- 更新サイクル: I/O 入力データ → システム変数 =  $3 * \text{補間周期} = 3 * 12 \text{ ms} = 36 \text{ ms}$   
MD10398 \$MN\_PLCIO\_IN\_UPDATE\_TIME =  $3 * 0.012 = 0.036$
- データ書式: ビッグエンディアン  
MD10399 \$MN\_PLCIO\_TYPE\_REPRESENTATION = 1

#### 更新

システム変数は、NC および PLC の起動後、3 回目の補間サイクル毎に更新されます。

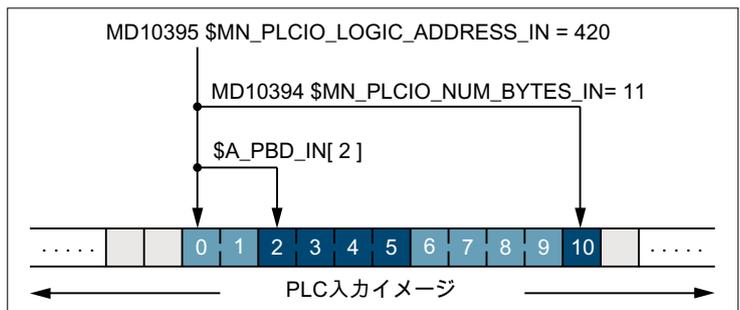
2.4 PLC を使わない直接 I/O アクセス

プログラミング

R 変数での NC I/O の読み取り:

プログラムコード	コメント
R1=\$A_PBW_IN[0]	; 2 バイトの整数値
R2=\$A_PBD_IN[2]	; 4 バイトの整数値
R3=\$A_PBR_IN[6]	; 4 バイトの REAL 値
R4=\$A_PBB_IN[10]	; 1 バイトの整数値

アドレス指定の例: R2 = \$PBD\_IN[ 2 ]



2.4 PLC を使わない直接 I/O アクセス

2.4.1 概略説明

同期クロックおよび非同期クロック PROFIBUS/PROFINET

同期クロックおよび非同期クロック PROFIBUS/PROFINET 設定によって、PROFIBUS/PROFINET の読み取り/書き込みが可能です。

I/O 範囲

NC I/O で使用される PROFIBUS/PROFINET スレーブ用に途切れなく昇順にスロットを構成している場合、以下では I/O 範囲と呼びます。

よって、I/O 範囲は以下の特徴があります。

- 論理先頭アドレス: PROFIBUS/PROFINET スレーブの最初のスロットの論理先頭アドレス
- 長さ: PROFIBUS/PROFINET スレーブ用に使用されるすべてのスロットの合計長さ

論理先頭アドレスおよび I/O 範囲のパラメータ設定は、NC でマシンデータを使っておこないません(「パラメータ設定 (ページ 68)」を参照してください)。

### 読み取り/書き込み

#### パートプログラム/シンクロナイズドアクション:システム変数

NC I/O の読み取り/書き込みは、NC で補間周期のシステム変数を使っておこなわれます。NC I/O の書き込みは、補間サイクル後におこなわれます。

データの完全性:補間周期(Interpolator cycle)

「システム変数 (ページ 70)」を参照してください。

### 並列読み取り/書き込み

#### 読み取り

同じ I/O 範囲の入力データのコンパイルサイクルとパートプログラム、またはシンクロナイズドアクションによる並列読み取りは**可能**です。NC I/O への書き込みアクセスは、異なる方法でおこなわれます。

- データの完全性が保証されます。
- 補間周期中の等価性は保証されません。

#### 書き込み

同じ I/O 範囲の出力データのコンパイルサイクルとパートプログラム、またはシンクロナイズドアクションによる並列書き込みは**不可**です。

## 2.4.2 パラメータ設定

### マシンデータ

#### I/O 範囲の論理先頭アドレス

使用する I/O 範囲の論理先頭アドレスは、以下のマシンデータを使って設定します。

- 入力範囲 1、2、... m の論理先頭アドレス:  
MD10500 \$MN\_DPIO\_LOGIC\_ADDRESS\_IN[ <n> ] = <論理先頭アドレス> ; <n> = 0、1、2、... (m - 1)
  - 出力範囲 1、2、... m の論理先頭アドレス:  
MD10510 \$MN\_DPIO\_LOGIC\_ADDRESS\_OUT[ <n> ] = <論理先頭アドレス> ; <n> = 0、1、2、... (m - 1)
- PLC プロセスイメージ内の論理先頭アドレスでは、下記を考慮してください(参照先をご覧ください):

- 入力スロット:NC からの読み取りも可能
- 出力スロット:NC からの書き込み禁止 ⇒ NC 起動後にメッセージ

#### 参照先

NCU 7x0.3 PN, NCU 7x0.3B PN マニュアル、セクション「技術仕様」 > サブセクション「PLC」 > 「プロセスイメージのサイズ」

詳細情報はアドレス (<http://support.automation.siemens.com/WW/view/de/54058408>) を参照してください。

#### 注記

PROFIBUS I/O の書き込みアクセス(MD10510)の I/O 範囲は、プロセスイメージの範囲であってはなりません(例:PLC 317-3、出力アドレス 0 ~ 1023)。

PROFINET I/O の場合は、これらの I/O 範囲をプロセスイメージで「TPA 2」に割り当てる必要があります。設定は、同期サイクル割り込み「NC」を使用しておこなわれます。

#### I/O 範囲の長さ

使用する I/O 範囲の長さは、以下のマシンデータを使って設定します。

- 入力範囲 1、2、... m の長さ:  
MD10501 \$MN\_DPIO\_RANGE\_LENGTH\_IN[ <n> ] = <長さ>; <n> = 0、1、2、... (m - 1)
- 出力範囲 1、2、... m の長さ:  
MD10511 \$MN\_DPIO\_RANGE\_LENGTH\_OUT[ <n> ] = <長さ>; <n> = 0、1、2、... (m - 1)

長さとして値 0 を入力した場合、指定された先頭アドレス(MD10500/MD10510)で見つかった最初のユーザーデータスロットの長さが、内部的に I/O 範囲の長さとして設定されます。

#### I/O 範囲の属性

- 入力範囲 1、2、... m の属性:

MD10502 \$MN\_DPIO\_RANGE\_ATTRIBUTE\_IN[ <n> ]; <n> = 0、1、2、... (m - 1)

ビット	規格値	意味
0		システム変数\$A_DP<x>_IN[<n>,<m>]の書式表示
	0	リトルエンディアン書式
	1	ビッグエンディアン書式
2		入力データの読み取り
	0	システム変数と CC バインディングによって読み取りできます
	1	CC バインディングのみ読み取りできます
3		スロットライフサインアラームの出力
	0	スロットライフサインアラームが発行されます
	1	スロットライフサインアラームがマスクされます

- 出力範囲 1、2、... m の属性:

MD10512 \$MN\_DPIO\_RANGE\_ATTRIBUTE\_OUT[ <n> ]; <n> = 0、1、2、... (m - 1)

ビット	規格値	意味
0		システム変数\$A_DP<x>_OUT[<n>,<m>]の書式表示
	0	リトルエンディアン書式
	1	ビッグエンディアン書式
1		出力データの書き込み
	0	システム変数によってのみ書き込み可能
	1	CC バインディングによってのみ書き込み可能
3		スロットライフサインアラームの出力
	0	スロットライフサインアラームが発行されます
	1	スロットライフサインアラームがマスクされます

2.4 PLC を使わない直接 I/O アクセス

2.4.3 読み取り/書き込み

2.4.3.1 システム変数

入力データ

システム変数	タイプ	意味
\$A_DPB_IN[<n>,<b>]	8 ビット符号なし	データバイト(8 ビット)を読み取る
\$A_DPW_IN[<n>,<b>]	16 ビット符号なし	データワード(16 ビット)を読み取る
\$A_DPSB_IN[<n>,<b>]	8 ビット符号付き	データバイト(8 ビット)を読み取る
\$A_DPSW_IN[<n>,<b>]	16 ビット符号付き	データワード(16 ビット)を読み取る
\$A_DPSD_IN[<n>,<b>]	32 ビット符号付き	データダブルワード(32 ビット)を読み取る
\$A_DPR_IN[<n>,<b>]	32 ビット REAL	入力データ(32 ビット REAL)を読み取る

<n> = 入力範囲 1、2、... m; <b> = 入力範囲:0、1、... (長さ- 1)のバイトインデックス

出力データ

システム変数	タイプ	意味
\$A_DPB_OUT[<n>,<b>]	8 ビット符号なし	データバイト(8 ビット)を書き込む
\$A_DPW_OUT[<n>,<b>]	16 ビット符号なし	データワード(16 ビット)を書き込む
\$A_DPSB_OUT[<n>,<b>]	8 ビット符号付き	データバイト(8 ビット)を書き込む
\$A_DPSW_OUT[<n>,<b>]	16 ビット符号付き	データワード(16 ビット)を書き込む
\$A_DPSD_OUT[<n>,<b>]	32 ビット符号付き	データダブルワード(32 ビット)を書き込む
\$A_DPR_OUT[<n>,<b>]	32 ビット REAL	出力データ(32 ビット REAL)を書き込む

<n> = 出力範囲 1、2、... m; <b> = 出力範囲:0、1、... (長さ- 1)のバイトインデックス

**パートプログラム/シンクロナイズドアクションの設定済みおよびパラメータ設定済みの I/O 範囲**

各システム変数は 32 ビットのビット配列です。各ビットは、1 つの I/O 範囲に割り当てられます。

ビット<n> ≙ マシンデータインデックス<n> ≙ I/O 範囲<n+1>

ビット<n> == 1 ⇒ I/O 範囲<n+1>がパラメータ設定されます。

システム変数	タイプ	意味
\$A_DP_IN_CONF	32 ビットのビット配列	すべての設定された入力範囲を読み取る
\$A_DP_OUT_CONF	32 ビットのビット配列	すべての設定された出力範囲を読み取る

**パートプログラム/シンクロナイズドアクションの有効な I/O 範囲**

各システム変数は 32 ビットのビット配列です。各ビットは、1 つの I/O 範囲に割り当てられます。

ビット<n> ≙ マシンデータインデックス<n> ≙ I/O 範囲<n+1>

ビット<n> == 1 ⇒ I/O 範囲<n+1>が有効です。パートプログラム/シンクロナイズドアクションを使った読み取り/書き込み

システム変数	タイプ	意味
\$A_DP_IN_VALID	32 ビットのビット配列	すべての有効な入力範囲を読み取り
\$A_DP_OUT_VALID	32 ビットのビット配列	すべての有効な出力範囲を読み取り

**I/O 範囲の状態**

I/O 範囲の状態は、以下のシステム変数で読み取ることができます。

システム変数	タイプ	意味
\$A_DP_IN_STATE[<n>]	INT	入力範囲の状態を読み取り
\$A_DP_OUT_STATE[<n>]		出力範囲の状態を読み取り

2.4 PLC を使わない直接 I/O アクセス

システム変数	タイプ	意味
<n> = I/O 範囲のインデックス		
状態		
0: I/O 範囲が未設定		
1: I/O 範囲が選択できなかった		
2: I/O 範囲が利用可能		
3: I/O 範囲が現在使用不可		

I/O 範囲長

I/O 範囲の設定した長さは、以下のシステム変数で読み取ることができます。

システム変数	意味
\$A_DP_IN_LENGTH[<n>]	入力データ範囲長の読み取り
\$A_DP_OUT_LENGTH[<n>]	出力データ範囲長の読み取り
<n> = I/O 範囲のインデックス	

必要条件

- パートプログラムからの読み取り/書き込み中に、チャンネルの先読み停止がトリガされます。
- パートプログラムとシンクロナイズドアクションからのプログラム実行中にデータの整合性を保証するために、各 IPO サイクルで整合性が維持されている PROFIBUS-I/O データへのアクセスがおこなわれます。
- 補間サイクル内で同じ PROFIBUS-I/O データの書き込みアクセスを数回おこなう場合 (たとえば、シンクロナイズドアクション、様々なチャンネルからのアクセスなど)、最後の書き込みアクセスのデータが有効になります。
- 書き込み PROFIBUS-I/O データは、対応する IPO サイクル後のタイミングでのみ PROFIBUS I/O へ出力されます。

- **<b>** (RangeOffset)は、データアクセスを開始する I/O 範囲内の位置(バイトオフセット)を示します。データタイプは、I/O 範囲内のどのバイトオフセットでも、読み取り/書き込みをおこなうことができます。各 I/O 範囲に設定された制限を超えた読み取り/書き込みアクセスは、アラーム(17030)を発行して拒否されます。
- マシンデータ MD10502 \$MN\_DPIO\_RANGE\_ATTRIBUTE\_IN または MD10512 \$MN\_DPIO\_RANGE\_ATTRIBUTE\_OUT によって(「パラメータ設定 (ページ 68)」を参照してください)、読み取り/書き込み指令について、および個々の I/O 範囲について、システム変数\$A\_DPx\_IN[<n>,<b>]または\$A\_DPx\_OUT[<n>,<b>]の表示書式(リトル/ビッグエンディアン)を定義できます。

## 2.4.4 必要条件

### NC と PLC の並列書き込み

NC による直接アクセスおよび PLC ユーザープログラムからの I/O 出力の並列書き込みを行うと、出力値のランダムな相互書き換えが発生します。よってこれは許可されませんが、コントローラでは防ぐことができません。

### ライフビートのモニタ

補間周期の開始時に、関連するスロットまたは I/O 範囲のライフサインが消失していないか、各 I/O 範囲のチェックがおこなわれます。消失している場合、メッセージ 9050 または 9052 が表示されます。

働き:

- パートプログラムの処理は停止しません。
- ライフサインが回復すると、メッセージは自動的にクリアされます。

## 2.4.5 例

### 2.4.5.1 NC I/O への書き込み

#### 必要条件

有効な設定が PLC に読み込み済みであること。

2.4 PLC を使わない直接 I/O アクセス

パートプログラム/シンクロナイズドアクションのパラメータ設定

仕様

- 6 番目のデータブロックのパラメータ設定: マシンデータ/システム変数インデックス = 5
- 設定ファイル:
  - 論理先頭アドレス = 334
  - スロット長さ = 8 バイト
- 表示: リトルエンディアン書式

マシンデータのパラメータ設定

- MD10510 \$MN\_DPIO\_LOGIC\_ADDRESS\_OUT[ 5 ] = 334 (論理先頭アドレス)
- MD10511 \$MN\_DPIO\_LENGTH\_OUT[ 5 ] = 8 (I/O 範囲の長さ(バイト))
- MD10512 \$MN\_DPIO\_ATTRIBUTE\_OUT[ 5 ]
  - Bit0 = 0 (リトルエンディアン書式)
  - Bit1 = 0 (システム変数のみによる書き込み)
  - Bit3 = 0 (スロットのライフサインメッセージの発行)

例

プログラムコード	コメント
\$A_DPB_OUT[5,6]=128	; バイト □ 8 ビット、インデックス=5、オフセット=6
\$A_DPW_OUT[5,5]='B0110'	; ワード □ 16 ビット、インデックス=5、オフセット=5
	; 注意: オフセット 6 のデータは上書きされます
\$A_DPSD_OUT[5,3]='H8F'	; ダブル □ 32 ビット、インデックス=5、オフセット=3
	; 注意: オフセット 4、5、6 のデータは上書きされます
\$AC_MARKER[0]=5	; インデックス=5
\$AC_MARKER[1]=3	; オフセット=3
\$A_DPSD_OUT[\$AC_MARKER[0], \$AC_MARKER[1]]='H8F'	; 間接アドレス指定
	; ダブル □ 32 ビット、オンインデックス=5、オフセット=3
R1 = \$A_DPB_OUT[5,6]	; R 変数への割り付け、バイト □ 8 ビット、インデックス=5、オフセット=6
	; 結果: R1 == 'HFF'
ID=1 WHENEVER TRUE DO \$A_DPB_OUT[5,0]=123	; IPO サイクル毎の周期的な書き込み
	; バイト □ 8 ビット、インデックス=5、オフセット=0
<b>; 誤ったプログラミング</b>	
\$A_DPB_OUT[5.255]=128	; □ メッセージ 17030: オフセット 255 > I/O 範囲
\$A_DPB_OUT[6.10]=128	; □ メッセージ 17020: コンパイルサイクル用に予約されたインデックス 6、以下を参照
\$A_DPB_OUT[7.10]=128	; □ メッセージ 17020: マシンデータでインデックス 7 が未定義
\$A_DPB_OUT[16.6]=128	; □ メッセージ 17020: インデックス 16 が数値の範囲外

## コンパイルサイクルによるプログラミングのための設定

## 仕様

- 7 番目のデータブロックのパラメータ設定:マシンデータ/システム変数インデックス = 6
- 設定ファイル:
  - 論理先頭アドレス = 444
  - スロット長さ = 10 バイト
- 表示:リトルエンディアン書式

## マシンデータのパラメータ設定

- MD10510 \$MN\_DPIO\_LOGIC\_ADDRESS\_OUT[ 6 ] = 444 (論理先頭アドレス)
- MD10511 \$MN\_DPIO\_LENGTH\_OUT[ 6 ] = 0 (最初のユーザーデータスロットのみ使用)
- MD10512 \$MN\_DPIO\_ATTRIBUTE\_OUT[ 6 ]
  - Bit0 = 0 (リトルエンディアン書式)
  - Bit1 = 1 (CC バインディングのみによる書き込み)
  - Bit3 = 1 (スロットのライフサインメッセージのマスク)

## 2.4.5.2 NC I/O からの読み取り

## 必要条件

有効な設定が PLC に読み込み済みであること。

## パートプログラム/シンクロナイズドアクションのパラメータ設定

## 仕様

- 1 番目のデータブロックのパラメータ設定:マシンデータ/システム変数インデックス = 0
- 設定ファイル:
  - 論理先頭アドレス = 456
  - スロット長さ = 32 バイト
- 表示:ビッグエンディアン書式

2.4 PLC を使わない直接 I/O アクセス

マシンデータのパラメータ設定

- MD10500 \$MN\_DPIO\_LOGIC\_ADDRESS\_IN[ 0 ] = 456 (論理先頭アドレス)
- MD10501 \$MN\_DPIO\_LENGTH\_IN[ 0 ] = 32 (I/O 範囲の長さ(バイト))
- MD105032 \$MN\_DPIO\_ATTRIBUTE\_IN[ 0 ]
  - Bit0 = 1 (ビッグエンディアン書式)
  - Bit2 = 0 (システム変数および CC バインディングによって読み取り可能)
  - Bit3 = 0 (スロットのライフサインメッセージの発行)

例

プログラムコード	コメント
\$AC_MARKER[0]= \$A_DPW_IN[0,0]	; バイト □ 8 ビット、インデックス=0、オフセット=0
\$AC_MARKER[1]= \$A_DPSD_IN[0,1]	; 符号付きダブル □ 32 ビット、インデックス=0、オフセット=1
\$AC_MARKER[1]= \$A_DPSD_IN[0.8]	; 符号付きダブル □ 32 ビット、インデックス=0、オフセット=8
\$AC_MARKER[2]=0	; インデックス=0
\$AC_MARKER[3]=8	; オフセット=8
\$AC_MARKER[1]=\$A_DPSD_IN[\$AC_MARKER[2],\$AC_MARKER[3]] ; 間接アドレス指定	; 符号付きダブル □ 32 ビット、インデックス=0、オフセット=8
ID=2 WHEN \$A_DPB_IN[0,11]>=5 DO \$AC_MARKER[2]=\$A_DPSD_IN[0,8]	; インデックス 0、オフセット 11 >= 5 の場合
	; 符号付きダブル □ 32 ビット、インデックス=0、オフセット=8
<b>; 誤ったプログラミング</b>	
R1=\$A_DPB_IN[0,255]	; □ メッセージ 17030:オフセット 255 > I/O 範囲
R1=\$A_DPB_IN[2.6]	; □ メッセージ 17020:マシンデータでインデックス 2 が未定義
R1=\$A_DPB_IN[1.10]	; □ メッセージ 17020:コンパイルサイクル用に予約されたインデックス 1、以下を参照
R1=\$A_DPB_IN[16.6]	; □ メッセージ 17020:インデックス 16 が数値の範囲外

CompileCycles によるプログラミングのためのパラメータ設定

仕様

- 2 番目のデータブロックのパラメータ設定:マシンデータ/システム変数インデックス = 1
- 設定ファイル:
  - 論理先頭アドレス = 312
  - スロット長さ = 32 バイト
- 表示:ビッグエンディアン書式

## マシンデータのパラメータ設定

- MD10500 \$MN\_DPIO\_LOGIC\_ADDRESS\_IN[ 01 ] = 312 (論理先頭アドレス)
- MD10501 \$MN\_DPIO\_LENGTH\_IN[ 1 ] = 32 (I/O 範囲の長さ(バイト))
- MD10502 \$MN\_DPIO\_ATTRIBUTE\_IN[ 1 ]
  - Bit0 = 1 (ビッグエンディアン書式)
  - Bit2 = 1 (CC バインディングのみ読み取り可能)
  - Bit3 = 1 (スロットのライフサインメッセージのマスク)

## 2.4.5.3 状態確認による NC I/O の書き込み

## 必要条件

有効な設定が PLC に読み込み済みであること。

## パートプログラム/シンクロナイズドアクションのパラメータ設定

## 仕様

- 6 番目のデータブロックのパラメータ設定:マシンデータ/システム変数インデックス = 5
- 設定ファイル:
  - 論理先頭アドレス = 1200
  - スロット長さ = 32 バイト
- 表示:ビッグエンディアン書式

## マシンデータのパラメータ設定

- MD10510 \$MN\_DPIO\_LOGIC\_ADDRESS\_OUT[ 5 ] = 1200 (論理先頭アドレス)
- MD10511 \$MN\_DPIO\_LENGTH\_OUT[ 5 ] = 0 (I/O 範囲の長さ(バイト))
- MD10512 \$MN\_DPIO\_ATTRIBUTE\_OUT[ 5 ]
  - Bit0 = 1 (リトルエンディアン書式)
  - Bit1 = 0 (システム変数のみによる書き込み)
  - Bit3 = 0 (スロットのライフサインメッセージの発行)

## 例

```

check:                                     ; マーカヘジャンプ
IF $A_DP_OUT_STATE[5]==2 GOTOF write      ; データ範囲有効; => N15 ヘジャンプ
GOTOB check                               ; check へ戻る

```

2.5 データリスト

```

write:                                     ; マーカヘジャンプ
$A_DPB_OUT[5,6]=128                       ; データバイト書き込み

check:                                     ; マーカヘジャンプ
IF $A_DP_OUT_CONF==$A_DP_OUT_VALID GOTOF write ; データ範囲有効
                                                ; => N15 ヘジャンプ
SETAL(61000)                               ; アラーム番号 61000 設定
write:                                     ; マーカヘジャンプ
$A_DPB_OUT[5,6]=128                       ; データバイト書き込み

check:                                     ; マーカヘジャンプ
IF $A_DP_OUT_VALID B_AND 'B100000' GOTOF write ; データ範囲有効
                                                ; => N15 ヘジャンプ
SETAL(61000)                               ; アラーム番号 61000 設定
write:                                     ; マーカヘジャンプ
$A_DPB_OUT[5,6]=128                       ; データバイト書き込み

R1=$A_DP_OUT_LENGTH[5]                    ; バイト単位の I/O 範囲長 (スロット)
                                                ; 結果: R1 = 32
    
```

2.5 データリスト

2.5.1 マシンデータ

2.5.1.1 一般マシンデータ

番号	識別子: \$MN_	説明
10300	FASTIO_ANA_NUM_INPUTS	有効アナログ NC 入力数
10310	FASTIO_ANA_NUM_OUTPUTS	有効アナログ NC 出力数
10320	FASTIO_ANA_INPUT_WEIGHT	アナログ NC 入力の加重係数
10330	FASTIO_ANA_OUTPUT_WEIGHT	アナログ NC 出力の加重係数
10350	FASTIO_DIG_NUM_INPUTS	有効デジタル NC 入力バイト数
10360	FASTIO_DIG_NUM_OUTPUTS	有効デジタル NC 出力バイト数
10362	HW_ASSIGN_ANA_FASTIN	外部アナログ NC 入力のハードウェア割り付け

番号	識別子: \$MN_	説明
10364	HW_ASSIGN_ANA_FASTOUT	外部アナログ NC 出力のハードウェア割り付け
10366	HW_ASSIGN_DIG_FASTIN	外部デジタル NC 入力のハードウェア割り付け
10368	HW_ASSIGN_DIG_FASTOUT	外部デジタル NC 出力のハードウェア割り付け
10394	PLCIO_NUM_BYTES_IN	PLC I/O の直接読み取り可能な入力バイト数
10395	PLCIO_LOGIC_ADDRESS_IN	PLC I/O の直接読み取り可能な入力バイトの開始アドレス
10396	PLCIO_NUM_BYTES_OUT	PLC I/O の直接書き込み可能な出力バイト数
10397	PLCIO_LOGIC_ADDRESS_OUT	PLC I/O の直接書き込み可能な出力バイトの開始アドレス
10398	PLCIO_IN_UPDATE_TIME	PLC I/O 入力周期の更新時間
10399	PLCIO_TYPE_REPRESENTATION	PLC I/O のリトル/ビッグエンディアン表示
10500	DPIO_LOGIC_ADDRESS_IN	PROFIBUS I/O の論理スロットアドレス
10501	DPIO_RANGE_LENGTH_IN	PROFIBUS I/O 範囲長
10502	DPIO_RANGE_ATTRIBUTE_IN	PROFIBUS I/O の属性
10510	DPIO_LOGIC_ADDRESS_OUT	PROFIBUS I/O の論理スロットアドレス
10511	DPIO_RANGE_LENGTH_OUT	PROFIBUS I/O 範囲長
10512	DPIO_RANGE_ATTRIBUTE_OUT	PROFIBUS I/O の属性
10530	COMPAR_ASSIGN_ANA_INPUT_1	コンパレータバイト 1 の NC アナログ入力のハードウェア割り付け
10531	COMPAR_ASSIGN_ANA_INPUT_2	コンパレータバイト 2 の NC アナログ入力のハードウェア割り付け
10540	COMPAR_TYPE_1	コンパレータバイト 1 のパラメータ設定
10541	COMPAR_TYPE_2	コンパレータバイト 2 のパラメータ設定

### 2.5.1.2 チャンネル別マシンデータ

番号	識別子: \$MC_	説明
21220	MULTFEED_ASSIGN_FASTIN	「1 ブロック内の複数の送り速度」が存在する場合の NC I/O の入力バイトの割り付け

## 2.5 データリスト

## 2.5.2 セッティングデータ

## 2.5.2.1 一般セッティングデータ

番号	識別子: \$SN_	説明
41600	COMPAR_THRESHOLD_1	コンパレータバイト 1 のしきい値
41601	COMPAR_THRESHOLD_2	コンパレータバイト 2 のしきい値

## 2.5.3 システム変数

識別子	説明
\$A_IN	デジタル NC 入力
\$A_OUT	デジタル NC 出力
\$A_INA	アナログ NC 入力
\$A_OUTA	アナログ NC 出力
\$A_PBB_IN	デジタル NC 入力データ、byte
\$A_PBW_IN	デジタル NC 入力データ、word
\$A_PBD_IN	デジタル NC 入力データ、double
\$A_PBR_IN	デジタル NC 入力データ、real
\$A_PBB_OUT	デジタル NC 出力データ、byte
\$A_PBW_OUT	デジタル NC 出力データ、word
\$A_PBD_OUT	デジタル NC 出力データ、double
\$A_PBR_OUT	デジタル NC 出力データ、real

## 2.5.4 信号

### 2.5.4.1 NC への信号

信号名称	SINUMERIK 840D sl	SINUMERIK 828D
デジタル NC 入力の無効化	DB10.DBB0/122/124/126/ 128	DB2800.DBB0/1000
PLC からのデジタル NC 入力の設定	DB10.DBB1/123/125/127/ 129	DB2800.DBB1/1001
デジタル NC 出力の無効化	DB10.DBB4/130/134/138/ 142	DB2800.DBB4/1008
デジタル NC 出力の上書きマスク	DB10.DBB5/131/135/139/ 143	DB2800.DBB5/1009
PLC からのデジタル NC 出力の設定値	DB10.DBB6/132/136/140/ 144	DB2800.DBB6/1010
デジタル NC 出力の入力マスク	DB10.DBB7/133/137/141/ 145	DB2800.DBB7/1011
アナログ NC 入力の無効化	DB10.DBB146	-
アナログ NC 入力の入力マスク	DB10.DBB147	-
アナログ NC 入力への PLC からの設定値	DB10.DBB148-163	-
アナログ NC 出力の上書きマスク	DB10.DBB166	-
アナログ NC 出力の入力マスク	DB10.DBB167	-
NC 出力の無効化	DB10.DBB168	-
アナログ NC 出力への PLC からの設定値	DB10.DBB170-185	-

### 2.5.4.2 NC からの信号

信号名称	SINUMERIK 840D sl	SINUMERIK 828D
デジタル NCK 入力の実際値	DB10.DBB60/186-189	DB2900.DBB0/1000
デジタル NCK 出力の指令値	DB10.DBB64/190-193	DB2900.DBB4/1004
アナログ NCK 入力の実際値	DB10.DBB194-209	-
アナログ NCK 出力の指令値	DB10.DBB210-225	-

## 2.5 データリスト

## B3 : 分散システム - 840D sl のみ

### 3.1 概略説明

#### 3.1.1 複数の NCU での複数の操作パネル(T:M:N)

特定の状況下で、単一のオペレータコントロールと監視ステーションでは複雑なシステムや機械に対して不十分な場合があります。したがって、システム全体の、柔軟性のある分散操作と監視を可能にする方法により、SINUMERIK システムネットワーク(Ethernet)で複数のオペレータコントロールと監視ステーションを PCU 経由で複数の数値制御(NCU)に接続できます。

以下の図は、オペレータコントロールと監視システム T:1:N を構成するためにシステムネットワークに現在接続できるコンポーネントの概要を示しています。意味:

- **T:**シンクライアントユニット(TCU)または HT8 ハンドヘルドユニット(PCU に接続)
- **M:**マンマシン制御(MMC)、例: PCU 50.x(SINUMERIK Operate 装備)
- **N:**数値制御ユニット(NCU)、例: NCU 7x0.3 PN

3.1 概略説明

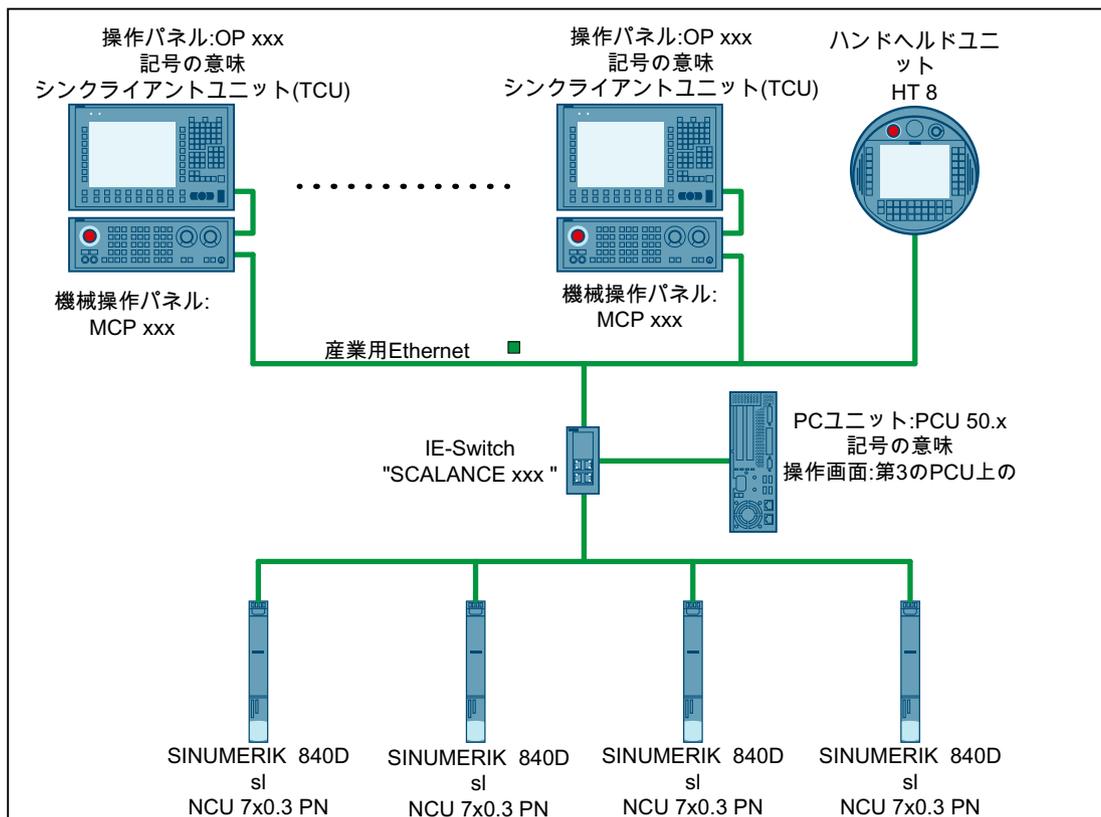


図 3-1 T:1:N ネットワークの例

**T:**シンククライアントユニット(TCU)または HT8 ハンドヘルドユニット

PCU の操作画面のグラフィック情報は、TCU によって操作パネル(OP)に伝送されて表示されます。TCU と操作パネルは併せて、操作ステーションを構成します。

- 最大数  
原則として、使用可能なアドレス空間量によってのみ制限される任意の数の操作ステーションをシステムネットワークで操作できます。
- オンライン  
PCU 毎に最大で 4 つの操作ステーションを同時にオンラインにできます。

- 運転フォーカス  
常に、オンラインの PCU 上の操作ステーションのいずれか 1 台のみが運転フォーカスを持つことができます。操作画面に関する入力、この操作ステーションによってのみおこなうことができます。他のオンライン操作ステーション上の操作画面は、運転フォーカスを持たないことを示すために暗くなります。オンラインではない操作ステーション上では、メッセージのみが表示されます。  
オンライン操作ステーション間で運転フォーカスを切り替えることができます。
- HT8 ハンドヘルドユニット  
HT8 ハンドヘルドユニットは、TCU、操作パネル、および機械操作パネルを一つの機器内に組み合わせたものです。

#### M:SINUMERIK Operate を備えた PCU

SINUMERIK Operate ユーザーインターフェースを備えた PCU には、複数の操作ステーションと複数の NCU を接続できます。チャンネルメニューの適切な設定によって、さまざまな NCU のチャンネルのデータを操作ステーションに表示できます。

#### N:SINUMERIK 840D sl

- 内部の SINUMERIK Operate ユーザーインターフェース  
システムネットワーク内の PCU 上の既存の外部 SINUMERIK Operate ユーザーインターフェースのために、NCU に組み込まれた内部 SINUMERIK Operate ユーザーインターフェースのスイッチをオフにしてください。工具管理機能の限られた機能範囲で使用する場合に限り、内部 SINUMERIK Operate ユーザーインターフェースのスイッチはオフにしないでください。詳細については、試運転マニュアル『オペレータコンポーネントとネットワーク設定(IM5)』、「ネットワーク設定」>「用途例」の章を参照してください。
- 機械操作パネル(MCP)  
機械操作パネルを柔軟に切り替えるために、1つの MCP のみを1つの NCU に接続できます。MCP の切り替えは、PLC ユーザープログラム(ファンクションブロック FB 9)によってユーザー別におこないます。

## 設置と接続

### 参照先

- TCU、MCP、PCU:  
『SINUMERIK 840D sl オペレータコンポーネントとネットワーク設定』マニュアル
- NCU:  
『SINUMERIK 840D sl NCU 7x0.3 PN マニュアル』

### 3.1 概略説明

- 機械操作パネル(MCP)  
FB 9:M 対 N コントロールユニット切り替え
- SCALANCE の切り替え:  
SIMATIC システムの通信マニュアル

#### 設定、セットアップ、およびパラメータ設定

##### 参照先

- システムネットワークの構成と TCU の試運転:  
『オペレータコンポーネントとネットワーク設定』試運転マニュアル(IM5)
- チャネルメニューの設定  
『SINUMERIK Operate』試運転マニュアル(IM9)、「チャネルメニュー」の章

#### プログラミング

##### 参照先

- MCP:  
『基本機能』機能マニュアル、「SINUMERIK 840D sl 用 P3 基本 PLC プログラム」>  
「ブロック概要」> 「FB 9:M 対 N コントロールユニット切り替え」の章

#### 境界条件

現在は、各 T:M:N ネットワークに接続できる PCU (M)は 1 台のみです(⇒ T:1:N)。

### 3.1.2 NCU リンク

#### 3.1.2.1 リンク通信

NCU リンク通信周期では、以下の用途で補間器クロック同期のクロス NCU データ交信が可能です。

- リンク変数\$A\_DLx によるクロス NCU データ交信
- クロス NCU によるリンク軸とコンテナリンク軸を使用した軸と主軸の移動
- マスタリンク軸を使用した軸と主軸のクロス NCU 軸間連動機能

### Safety Integrated

軸の Safety Integrated 機能がリンク軸に対して有効である場合、ローカル安全軸(NCUx 上のリンク軸)と、NCU リンクによってそれを移動しているチャンネル(NCUy)の間で、安全に関連しないデータが交信されます。またこれによって速度監視に関連する指令値の制限、上位停止機能(STOP D / E)など、リンク軸の Safety Integrated サポート機能が有効になります。

---

#### 注記

Safety Integrated 機能「セーフソフトウェアリミットスイッチ」(SE)と「セーフポジションスイッチ、セーフカムトラック」(SN)は、まだサポートされていません。

---

### リンクモジュール

リンク通信では、オプションのリンクモジュールが必要です。リンクモジュールは、Ethernet によるアイソクロナスリアルタイム通信(IRT)を可能にするオプションの PROFINET モジュールです。リンクモジュールは、NCU のオプションスロットを占有します。

### 数量構成

標準では、リンクグループを形成するために最大 3 つの NCU を相互接続できます。

---

#### 注記

特定のプロジェクト毎に、地域のシーメンス代理店にご要望いただくことにより、リンクグループを形成するためにさらに NCU を組み込むことができます。

---

#### 3.1.2.2 リンク変数

リンク変数は、設定されたリンク通信のためにネットワークの全 NCU によって使用可能なグローバルシステムユーザー変数です。リンク変数は、パートプログラム、サイクル、およびシンクロナイズドアクションで読み取りと書き込みをおこなうことができます。

リンク変数の読み取りと書き込みがメインランでおこなわれます。したがって、リンク変数についてチャンネル同期が自動的におこなわれ、全てのチャンネルが同じ補間周期内に同じ値を読み取ります。それに対して、グローバルユーザーデータ(GUD)もまた、リンク変数のようにクロスチャンネルで使用できます。ただし、GUD 変数は前処理されるた

### 3.1 概略説明

め、ここでは、自動チャンネル同期はおこなわれません。したがって、必要に応じてユーザーが独自のプログラミングにより GUD 変数を同期する必要があります。

---

#### 注記

NCU リンクのないシステムでは、リンク変数は標準グローバルユーザーデータ(GUD)と一緒に使用される追加 NCU グローバルユーザーデータとして使用することもできます。

---

#### 3.1.2.3 リンク軸

リンク軸により、機械軸が物理的に接続された NCU のチャンネルによって機械軸の指令値が作成されるのではなく、リンクグループの任意の NCU のチャンネルによって機械軸の指令値が作成される分散コントロールシステムが可能になります。

##### リンク軸の固定割り当て

どの NCU 上でどの機械軸の指令値を割り当てるかは、マシンデータ設定によりリンク軸に固定されます。

##### コンテナリンク軸の変数の割り当て

どの機械軸の指令値がどの NCU に対して作成されるかが割り当てられるのは、コンテナリンク軸の変数です。したがってコンテナリンク軸は、リンクグループの各 NCU によって代わりに移動できます。「軸コンテナ (ページ 117)」の章を参照してください。

前述の割り当てには関係なく、指令値を生成する NCU 上の複数のチャンネルのリンク軸に論理機械軸を知らせて、その後、たとえば位置/送りの軸間連動機能、NCU ローカル軸入れ替えなど、ローカル論理機械軸として機能的に処理できます。

#### 3.1.2.4 マスタリンク軸

汎用の連結機能と組み合わせて、全ての補間器(つまり指令値作成/処理チャンネル、マスタ軸とスレーブ軸/主軸)が同じ NCU 上にある場合、マスタリンク軸を使用する必要はありません。マスタ軸および(または)スレーブ軸/主軸の機械軸はまた、異なる NCU に接続するリンク軸になることができます。

##### 用途

連結グループの全ての補間器の1つでも同じ NCU 上にはない場合は、常にマスタリンク軸が必要です。マスタリンク軸は、別の NCU 上で、マスタ軸/主軸のイメージまたはローカル補間器のマスタ値を提供します。次に、このマスタ値から始めて、スレーブ軸/主軸の指令値がこの NCU 上の補間器で生成されます。

### その他の用途

連結のために、NCU 上で連結に含まれる機械軸が、絶対的または相対的に使用できる機械軸より多い場合、マスタリンク軸も使用されます。

### 参照先

『応用機能』機能マニュアル、「軸連結(M3)」>「汎用連結」の章

### 3.1.2.5 従属関係

補間関係で複数の軸を移動するには、移動チャンネルの補間器によって生成された指令値を同時に位置制御に伝送することが必要です。

例えば、リンク軸に対する NCU リンクによる指令値の伝送中に、ローカル軸の指令値について 1 補間周期の遅延が生じます。この遅延は、補間器と位置コントローラ間のバッファを適切に増やすことによって補正してください。

MD18720 \$MN\_MM\_SERVO\_FIFO\_SIZE = <値> = <初期値 + 補正值>

機能	補正值
リンク軸	リンク軸またはコンテナリンク軸により補間される軸を持つすべての NCU 上では、1 補間周期を補正してください。 補正值 = 1
コンテナリンク軸	
マスタリンク軸	マスタリンク軸の指令値を生成するマスタ軸の NCU 上では、2 補間周期を補正してください。 補正值 = 2

### 注記

マスタリンク軸とは違って、リンク軸とコンテナリンク軸ではもう 1 つの補正值が必要です(MD18720 \$MN\_MM\_SERVO\_FIFO\_SIZE)。リンクグループ内のリンク軸またはコンテナリンク軸とマスタリンク軸の同時使用により、遅延を補正することはできません。

3.1 概略説明

3.1.2.6 用途例:ロータリインデックスマシン

以下の例に、2 台の NCU を備えるロータリサイクルマシンに基づく「NCU リンク」機能の用途例を示します。ロータリインデックスマシンの主要ユニットは次のとおりです。

- 回転テーブル用の MTR 回転軸
- MS1 ~ MS4 主軸
- マシンステーション 1 と 2(それぞれに直線軸の X1 と Z1 を装備しています)
- ステーションのロードとアンロード

機械軸はすべて、それぞれ特定の NCU に完全に固定されたままです。同じ軸(X および Z)と主軸(S1)は、当該の NCU の加工プログラムで常にアドレス指定されます。

ワーク状態は、各加工ステップの後に概略図によって表わされます。

各加工ステップについて、回転テーブル(MTR)は 1 つの位置だけ左回りに進みます。これにより、各加工ステップについて別のステーションに主軸が割り当てられます。チャンネルでの機械軸と定義された主軸の関係の変更は、NCU 内の軸コンテナとして表されません。

主軸の機械軸が自身の NCU に含まれない場合、リンク通信によって指令値が関連する NCU に転送され、その機械軸に出力されます。

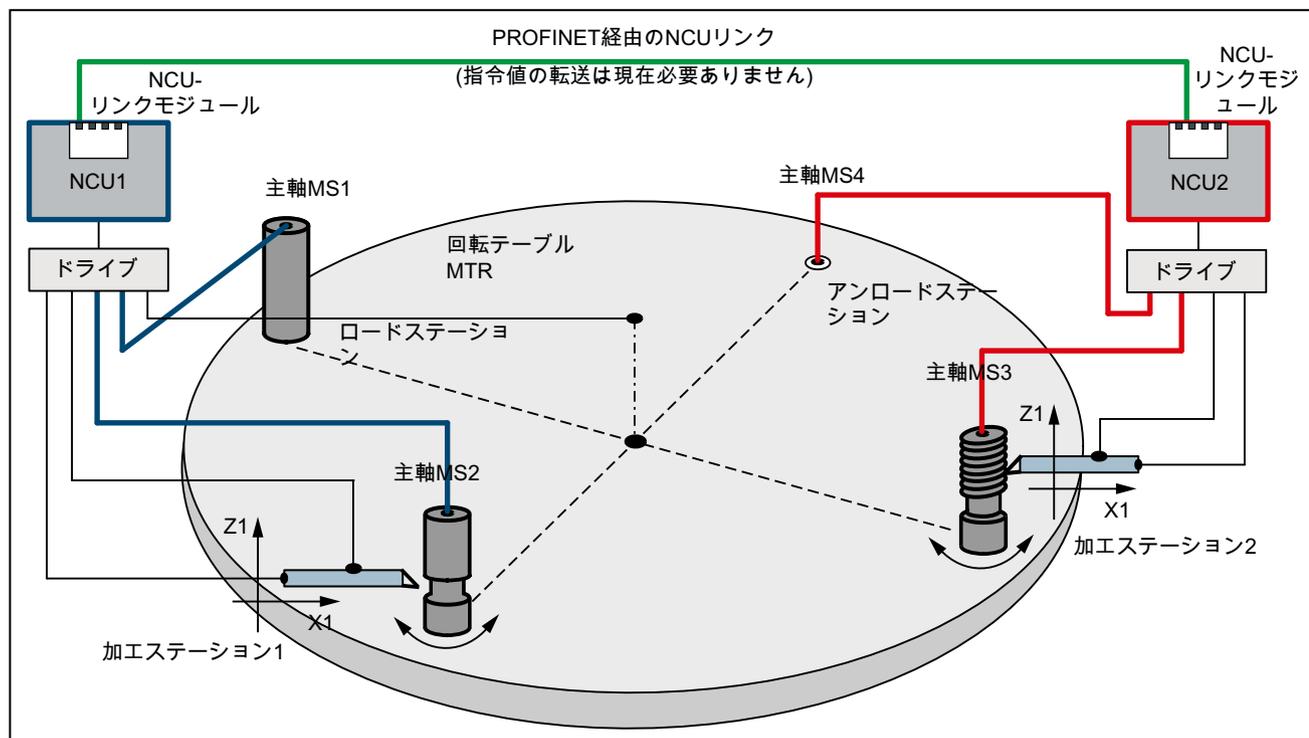


図 3-2 図 1 : 初期状態

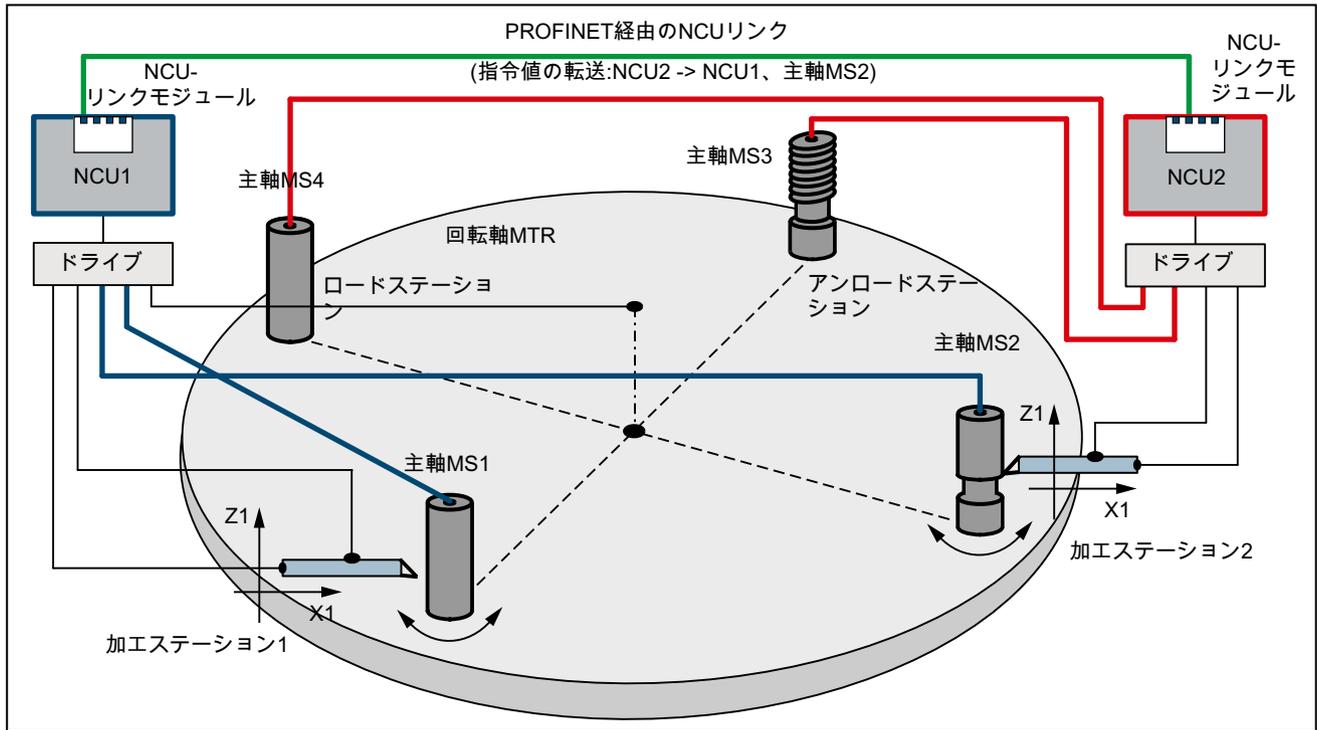


図 3-3 図 2 : 1位置の回転後のロケーション

パラメータ設定(概略)

概要	
両方の NCU のパートプログラムでプログラム指令されたチャネル軸: X、Z、S1	
NCU1	NCU で定義された機械軸
	ローカル: X1、Z1
	軸コンテナ: MS1、MS2
NCU2	NCU で定義された機械軸
	ローカル: X1、Z1
	軸コンテナ: MS3、MS4

3.2 NCU リンク

	初期設定 (図 1)	MTR(回転テーブル)の回転軸の 1 位置 毎の回転 (図 2)
NCU1	加工ステーション 1: X1、Z1、MS2	
	パートプログラムでプログラム指令されたチャンネル軸を示す図:	
	X → X1 と Z → Z1	X → X1 と Z → Z1
	S1 → MS2	S1 → MS1
NCU2	加工ステーション 2: X1、Z1、MS2	
	パートプログラムでプログラム指令されたチャンネル軸を示す図:	
	X → X1 と Z → Z1	X → X1 と Z → Z1
	S1 → MS3	S1 → MS2 (リンク軸)

3.2 NCU リンク

3.2.1 リンク通信

3.2.1.1 概要

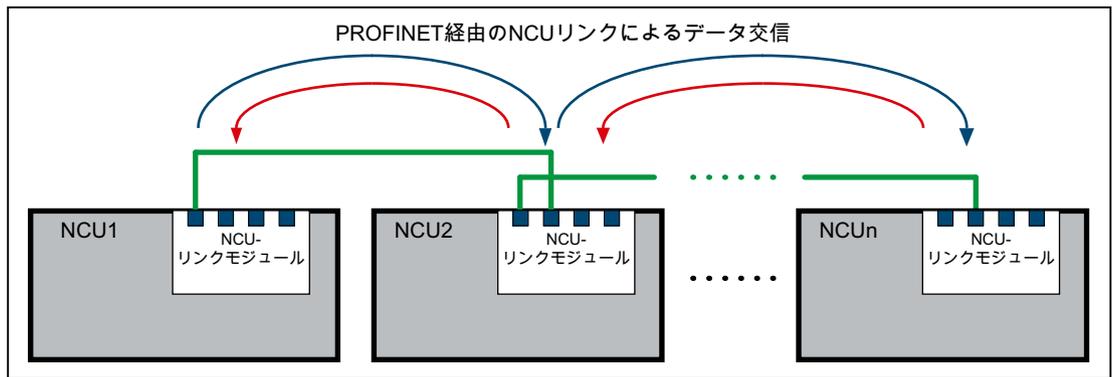


図 3-4 リンク通信(基本)

NCU リンク通信周期では、以下の用途で補間器クロック同期のクロス NCU データ交信が可能です。

- クロス NCU リンク変数\$A\_DLx  
NCU リンク通信に関連するすべての NCU では、リンクグループの NCU 間での NCU リンク補間周期の同期によってリンク変数が交信されるので、リンク変数に関する共通の表示が得られます。
- クロス NCU によるリンク軸とコンテナリンク軸を使用した軸と主軸の移動  
単独軸または補間軸の移動が NCU チャネルからおこなわれる場合、リンクグループの別の NCU に物理的に接続する軸に NCU リンク指令値も伝送できます。こうした軸はリンク軸として指定されます。
- マスタリンク軸を使用した軸と主軸のクロス NCU 軸間連動機能  
NCU1 の X1 軸(マスタ軸)を移動します。指令値は NCU リンク経由で NCU2 のリンク軸(マスタリンク軸)に転送されます。X2 軸の連結は、NCU2 のこのマスタリンク軸でおこなわれます。これは、軸 X2 が間接的に X1 のスレーブ軸であることを意味します。

## データ送受信

有効な機能に応じて、リンク通信に関連する NCU 間で、以下の周期的および非周期的データ伝送が発生します。

- 優先順位の高い周期的なデータ送受信
  - 軸の指令値、現在値、および状態信号
  - NCU 状態信号
- 優先順位の高い、非周期的で、置き換え不能なデータ送受信:
  - Safety Integrated 機能の一部として安全性に関連しないデータ
  - 軸コンテナ回転時のコンテナ軸の状態
- 優先順位の低い、非周期的で、置き換え可能なデータ送受信(優先順位の降順に表記):
  - リンク変数
  - ウォーム再起動要求
  - 軸コンテナ回転の起動
  - NCU グローバルマシンデータとセッティングデータの変更
  - リンク軸の軸マシンデータの適用
  - アラーム

### 3.2 NCU リンク

#### 置き換え

優先順位の低い、非周期的で置き換え可能なデータ伝送では、単一の補間周期内ですべての要求を送信できない場合、より高い優先順位の要求で低い優先順位の要求を置き換えます。より低い優先順位の要求は、その後の時点で送信されます。

#### 数量構成:NCU 7x0.3 PN

標準では、リンクグループを形成するために最大 3 つの NCU を相互接続できます。

---

#### 注記

特定のプロジェクトについて、3 台を超える NCU を含む NCU リンクグループをご希望の場合は、お近くの当社代理店にお問い合わせください。プロジェクト専用の条件がない場合、3 台を超える NCU についてアラーム 380020 により拒否されます。

---

## NCU リンクと Safety Integrated

以下の図は、2 台の NCU と 2 つの機械軸の組み合わせを示しています(そのうち NCU2 の MA2 機械軸は、NCU1 のリンク軸として移動します)。Safety Integrated 機能は、両軸の安全に関連する事項を監視します。

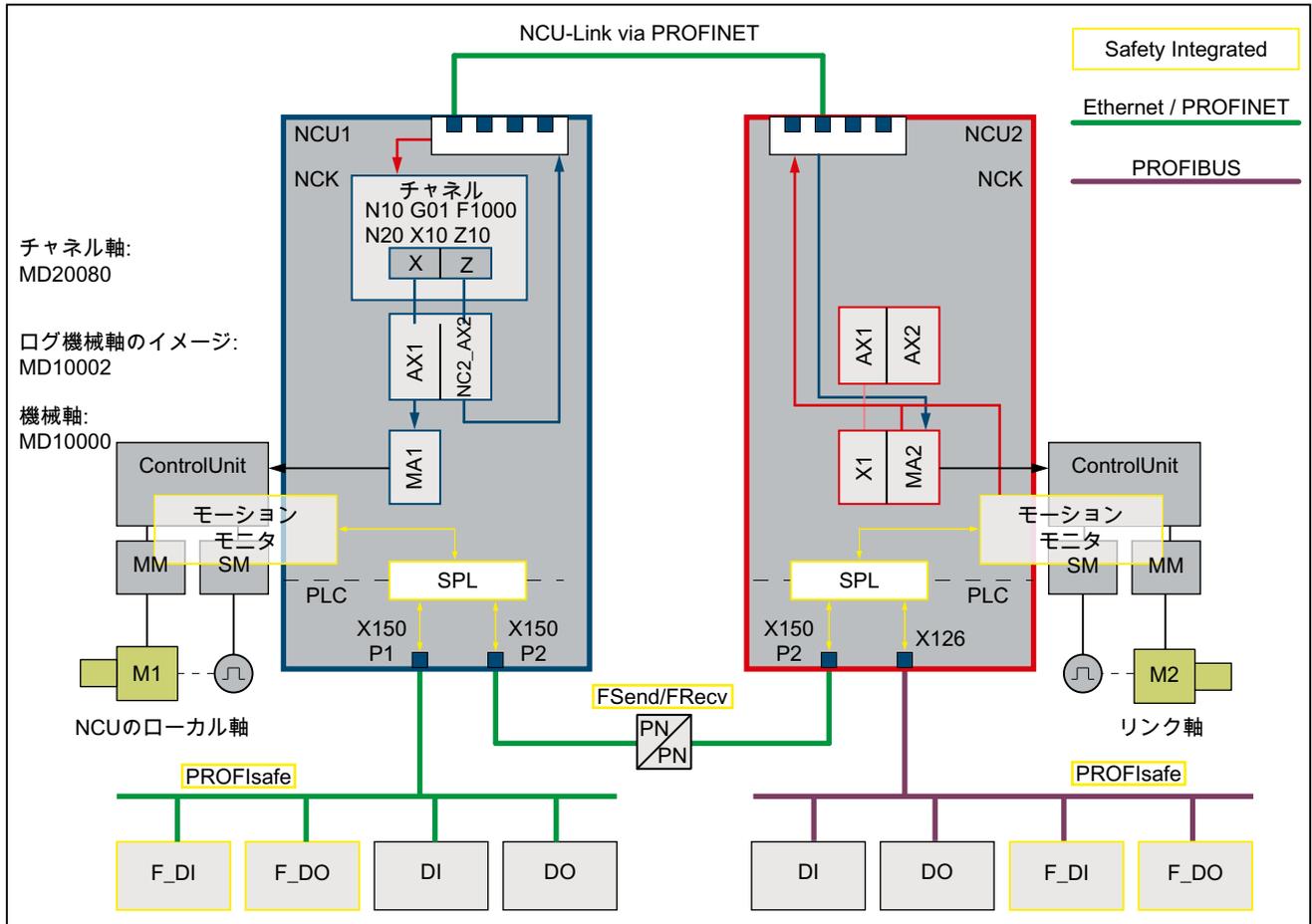


図 3-5 NCU リンクと Safety Integrated

両 NCU の安全に関連する入力信号(F\_DI)は、安全プログラマブルロジック(SPL)によって取得、およびリンクが可能で、安全に関連する CPU-CPU 通信(FSend/FRecv)を使用して交信できます。

Safety Integrated 機能の一部として、NCU リンク通信は、Safety Integrated サポート機能について、単一チャンネルに安全性に関連しないリンク軸データの伝送を可能にします。

## 3.2 NCU リンク

例:

- チャンネル内で、リンク軸を含めた、すべての移動軸の自動 Stop D 用の補間減速。
- 安全速度の変更のために、リンク軸を含めて、チャンネル内のすべての軸移動速度を減速。

### Safety Integrated 受け入れテストと NCU リンク

受け入れテストウィザード(ATW)は一般に、各 NCU について受け入れテストを個別におこなうために使用されます。リンク軸については、軸のホーム NCU 上のアラームのみが表示されます。ATW でリンク軸のアラームもチェックをするには、NCU リンク経由で接続された安全関連の NCU を ATW に指示してください。このためには、ATW が実行される PG/PCU で NETNAMES.INI ファイルを入力してください。

例:2 台の安全関連 NCU を含む NCU リンク用の NETNAMES.INI ファイルのエントリ

#### 設定ファイル:NETNAMES.INI (抜粋)

```
[own]
owner = HMI_1
[conn HMI_1]
conn_1= NCU_1
conn_2= NCU_2
[param HMI_1]
mmc_address = 1
[param network]
[param NCU_1]
nck_address= 192.168.214.2,LINE=10,NAME=/NC,SAP=030d,PROFILE=CLT1__CP_L4_INT
plc_address= 192.168.214.2,LINE=10,NAME=/PLC,SAP=0201,PROFILE=CLT1__CP_L4_INT
ADDRESS2= 192.168.214.2,LINE=10,NAME=/CP,SAP=0501,PROFILE=CLT2__CP_L4_INT
ADDRESS10= 192.168.214.2,LINE=10,NAME=/
DRIVE_00_000,SAP=0201,SUBNET=0000-00000000:000,PROFILE=CLT2__CP_L4_INT
ADDRESS11= 192.168.214.2,LINE=10,NAME=/DRIVE_03_003,SAP=0900,PROFILE=CLT2__CP_L4_INT
ADDRESS12= 192.168.214.2,LINE=10,NAME=/DRIVE_03_011,SAP=0B00,PROFILE=CLT2__CP_L4_INT
ADDRESS13= 192.168.214.2,LINE=10,NAME=/DRIVE_03_012,SAP=0C00,PROFILE=CLT2__CP_L4_INT
ADDRESS14= 192.168.214.2,LINE=10,NAME=/DRIVE_03_013,SAP=0D00,PROFILE=CLT2__CP_L4_INT
ADDRESS15= 192.168.214.2,LINE=10,NAME=/DRIVE_03_014,SAP=0E00,PROFILE=CLT2__CP_L4_INT
ADDRESS16= 192.168.214.2,LINE=10,NAME=/DRIVE_03_015,SAP=0F00,PROFILE=CLT2__CP_L4_INT
name=Machine_1
```

## 設定ファイル:NETNAMES.INI (抜粋)

```
[param NCU_2]
nck_address= 192.168.214.1,LINE=10,NAME=/NC,SAP=030d,PROFILE=CLT1__CP_L4_INT
plc_address= 192.168.214.1,LINE=10,NAME=/PLC,SAP=0201,PROFILE=CLT1__CP_L4_INT
ADDRESS2=    192.168.214.1,LINE=10,NAME=/CP,SAP=0501,PROFILE=CLT2__CP_L4_INT
ADDRESS10=   192.168.214.1,LINE=10,NAME=/
DRIVE_00_000,SAP=0201,SUBNET=0000-00000000:000,PROFILE=CLT2__CP_L4_INT
ADDRESS11=   192.168.214.1,LINE=10,NAME=/DRIVE_03_003,SAP=0900,PROFILE=CLT2__CP_L4_INT
ADDRESS12=   192.168.214.1,LINE=10,NAME=/DRIVE_03_011,SAP=0B00,PROFILE=CLT2__CP_L4_INT
ADDRESS13=   192.168.214.1,LINE=10,NAME=/DRIVE_03_012,SAP=0C00,PROFILE=CLT2__CP_L4_INT
ADDRESS14=   192.168.214.1,LINE=10,NAME=/DRIVE_03_013,SAP=0D00,PROFILE=CLT2__CP_L4_INT
ADDRESS15=   192.168.214.1,LINE=10,NAME=/DRIVE_03_014,SAP=0E00,PROFILE=CLT2__CP_L4_INT
ADDRESS16=   192.168.214.1,LINE=10,NAME=/DRIVE_03_015,SAP=0F00,PROFILE=CLT2__CP_L4_INT
name=Machine_2
```

## 参照先

「Safety Integrated」機能の詳細は、以下を参照してください。

『機能マニュアル、Safety Integrated』

## 3.2.1.2 リンクモジュール

NCU リンク通信は、リンクモジュール経由でおこなわれます。リンクモジュールは、Ethernet 経由のアイソクロナスリアルタイム通信(IRT)を可能にするオプションの PROFINET モジュールです。リンクモジュールは、NCU リンク通信のみに使用できません。リンクモジュールを一般の PROFINET 通信に使用することはできません。

リンクモジュールを使用するには、NCU モジュールにオプションスロットが必要です。

## 注記

NCU モジュールにはオプションスロットが1つしかありません。これにより、NCU リンクモジュールと他のオプションモジュールの並列使用が防止されます。

## リンクモジュールと NCU モジュール

「NCU710.3 PN」、「NCU720.3 PN」、および「NCU730.3 PN」 NCU モジュール用のリンクモジュールとして、通信ボード Ethernet CBE30-2 を利用できます。リンクモジュールを使用するには、NCU モジュールにオプションスロットが必要です。

## 参照先

マニュアル『NCU7x0.3 PN』、「接続可能なコンポーネント」>「CBE30-2」の章

## 3.2 NCU リンク

### 3.2.1.3 パラメータ設定 NC システム周期

リンク通信の基本的な必要条件として、NCU グループに含まれるすべての NCU で、後述のシステム周期が同じになるように設定してください。

- 基本システム周期
- 位置制御周期
- 補間周期

#### 基本システム周期

STEP 7 プロジェクトでアイソクロナス通信用に設定された DP 周期が基本システム周期として使用されます。現在の基本システム周期は、次のマシンデータで表示されます。

MD10050 \$MN\_SYSCLOCK\_CYCLE\_TIME

---

#### 注記

##### 複数の通信バス間での手動調整

複数のアイソクロナス通信バス(PROFIBUS 1... n、PROFINET 1... m)が NCU 上で設定されている場合、STEP 7 HW Config で各通信バスについて同じ DP サイクル時間を設定してください。

##### 位置制御周期により異なります

SINUMERIK 840D sl の場合、基本システム周期と位置制御周期間の比率は固定値 1:1 であり、変更できません。NCU リンクに対して特定の位置制御周期のみを設定できるので、基本システム周期または DP サイクル時間として、これらの位置制御周期のみを設定できます。次の「位置制御周期」の項を参照してください。

---

#### 位置制御周期

位置制御周期は、基本システム周期の比率として設定されます。SINUMERIK 840D sl の場合、比率は 1:1 に固定されており、変更できません。現在の位置制御周期は、次のマシンデータで表示されます。

## MD10061 \$MN\_POSCTRL\_CYCLE\_TIME

## 注記

## 許可された位置制御周期

NCU リンクの場合、リンクグループ内の NCU の数に応じて、以下の位置制御周期[ms]のみが設定されます。

- **2 NCU:**2.0, 2.5, 3.0, 3.5, 4.0
- **3 NCU:**3.0, 3.5, 4.0

「設定 (ページ 101)」の章を参照してください。

## 補間周期

補間周期は、基本システム周期の比率として設定されます。この設定は、次のマシンデータを使用しておこないます。

## MD10070 \$MN\_IPO\_SYSCLOCK\_TIME\_RATIO

実際の補間周期は、次のマシンデータで表示されます。

## MD10071 \$MN\_IPO\_CYCLE\_TIME

## 設定に関する注意

## 周期の設定内容

推奨される設定は次のとおりです。

- NC の CPU 時分割は、初期設定の **90%**を維持してください。

## MD10185 \$MN\_NCK\_PCOS\_TIME\_RATIO

- 補間器と位置コントローラによる**平均**システム負荷が通常のプログラム運転で **60%**を超えないようにシステム周期を設定してください。**最大値**として、**90%**を超えてはいけません。

## 注:

**特別な場合では、最大値 > 100%**と表示されることがあります。

## 最小通信周期負荷

リンク通信中の最小通信周期負荷では、補間周期と位置制御周期比は **1:1** になります。

1. 「ダイナミックサーボ制御(DSC)」ドライブ機能が起動している場合、これは一般的に最も有効な設定です。

制限事項：リンク通信ではメッセージの繰り返しは可能ではありません。

3.2 NCU リンク

**NC/PLC インタフェースの更新のタイミング**

以下の設定は、リンクグループに含まれるすべての NCU の起動時に有効になります。

**MD18000 \$MN\_VDI\_UPDATE\_IN\_ONE\_IPO\_CYCLE = 1**

これにより、1つの補間周期で NC/PLC インタフェースの読み書きが完了します。

**システム負荷**

システム負荷は、SINUMERIK Operate ユーザーインタフェースの次の場所に表示されます。「診断」操作エリア > 「ETC キー」 (「>」) > 「システム負荷」

**補間器クロック周期およびリンク変数**

システムクロック周期のパラメータ設定時には、「最悪の場合」のシナリオでの最大システム負荷の値に加えて、生産運転でリンク変数を伝送した場合に補間器レベルで予想される最大 CPU 処理時間ロードを考慮に入れる必要があります。

 <b>警告</b>
<p><b>補間器レベルでの CPU 処理時間オーバーフローによる開ループ制御障害</b></p> <p>NC のパラメータ設定時に補間器クロック周期が短く設定されていた場合は、リンク変数の伝送により CPU 処理時間ロードが増加するため、生産運転で CPU 処理時間オーバーフローが補間器レベルで発生する可能性があります。</p> <p>参照先: 『診断マニュアル』、「NC アラーム」の章 &gt; アラーム 4240 「IPO または位置コントローラレベルでの CPU 処理時間オーバーフロー」。</p>

3.2.1.4 パラメータ設定 リンク通信

**NC 別マシンデータ**

番号	識別子 \$MN_	意味
MD12510	NCU_LINKNO	<p>リンクグループ内の NCU の固有の数値識別子。識別子は、1 から昇順に連番で抜けないように割り当ててください。</p> <p>数値の範囲: 1, 2, ... 最大 NCU 番号</p> <p><b>注:</b></p> <p>NCU 識別子として値 1 が割り当てられる NCU は、リンクグループの<b>マスタ NCU</b>です。リンク軸と軸コンテナのパラメータ設定は、マスタ NCU のマシンデータとセッティングデータでのみ可能です。</p>
MD18780	MM_NCU_LINK_MASK.Bit 0	リンク通信の起動

番号	識別子 \$MN_	意味
MD18781	NCU_LINK_CONNECTIONS	内部リンク接続数  注: 初期値 0 を維持することを推奨します(この数値は NC によって決定されます)。
MD18782	MM_LINK_NUM_OF_MODULES	NCU リンク経由で相互接続される NCU の数。

### 3.2.1.5 設定

NC システムソフトウェアは、NCU 番号とリンクグループの位置制御周期のサポートされる各組み合わせについて、特定の設定を提供します(「パラメータ設定 NC システム周期 (ページ 98)」の章を参照ください)。

システム起動時に、当該の設定がロードされます(マシンデータでパラメータ設定された値による指定に従います)。

- MD18782 \$MN\_MM\_LINK\_NUM\_OF\_MODULES (リンクグループの NCU の数)
- MD10061 \$MN\_POSCTRL\_CYCLE\_TIME (位置制御周期)

#### 注記

用途によって、提供された標準設定が使用できない場合は、お近くの当社営業所の販売担当者にお問い合わせください。

### 3.2.1.6 NCU の配線

リンクグループ内の NCU の続き番号は、NCU 内で次のマシンデータを使用して定義されます。

MD12510 \$MN\_NCU\_LINKNO = <NCU 番号>、NCU 番号 = 1 ~ 最大 NCU 番号

#### ケーブル接続

NCU リンクモジュールは、NCU1 から開始して、後述の概略図に従って NCU 番号の順に配線してください。NCU(n), Port 0 → NCU(n+1), Port 1

3.2 NCU リンク

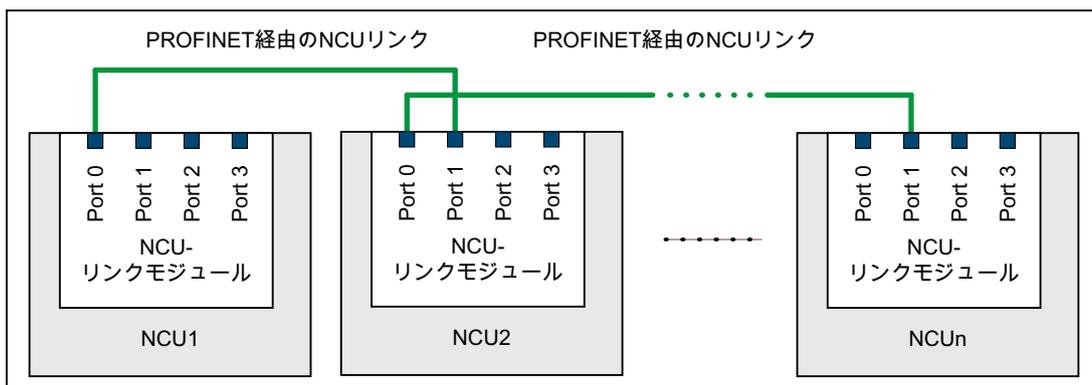


図 3-6 配線図、NCU リンク

3.2.1.7 適用

リンク通信は次のマシンデータを使用して有効にします。

MD18780 \$MN\_MM\_NCU\_LINK\_MASK, Bit 0 = 1

注記

適用タイミング

すべての関与する NCU で全機能の完全なセットアップが終わった後でのみ、リンク通信を起動することを推奨します。

3.2.2 リンク変数

複数の NCU を備える複雑なシステムでは、加工処理のシステム全体の協調のために、NCU 間でユーザー別のデータの周期的交信が必要です。データ交信は、リンク通信と特別なメモリ領域(各 NCU で利用可能なリンク変数メモリ)によっておこなわれます。

リンク変数メモリのサイズとデータ構造は、特定のユーザー毎に指定できます。リンク変数メモリに保存されたデータは、特別な \$A\_DLx リンク変数を使用してアドレス指定されます。

したがって、リンク通信を構成している場合、リンク変数は、リンクグループに関与するすべての NCU によりパートプログラムとサイクルで読み取りと書き込みが可能な、システムグローバルユーザー変数です。グローバルユーザー変数(GUD)と異なり、リンク変数はシンクロナイズドアクションで使用することもできます。

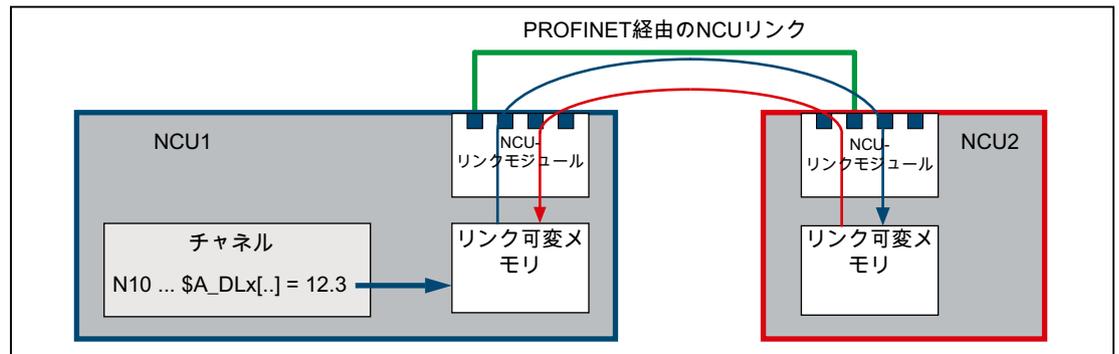


図 3-7 リンク変数

### NCU グローバルユーザー変数としてのリンク変数

#### 注記

NCU がリンクグループのノードでないか、または NCU リンクによる通信でリンク変数が必要ない場合、NCU グローバルユーザー変数としてリンク変数を使用できます。

#### 3.2.2.1

### リンク変数メモリの特性

#### メモリサイズのパラメータ設定

リンク変数メモリのサイズ(バイト)は、以下のマシンデータで設定します。

MD18700 \$MN\_MM\_SIZEOF\_LINKVAR\_DATA (リンク変数メモリのサイズ)

リンク変数メモリのサイズの設定は、リンクグループに関連するすべての NCU で同じにしてください。メモリサイズが異なっている場合は、割り当てられた最大値が適用されます。

#### 初期化

NCU の起動後、リンク変数メモリは 0 で初期化されます。

#### 構成

システムの視点から見ると、リンク変数メモリはリンク通信に使用できる、未構成のメモリ空間です。リンク変数メモリはユーザー/工作機械メーカーのみによって構成されます。データ構成がどのように定義されているかに応じて、データフォーマット別のリンク変数を使用してリンク変数メモリにアクセスします。

#### システム全体の整合

リンク変数メモリに書き込みがおこなわれると、データに加えられた変更はリンクグループ内の関連する他のすべての NCU のリンク変数メモリに伝送されます。リンク変数メモリは通常、2 補間周期以内にリンク通信によって更新されます。

## 3.2 NCU リンク

## 3.2.2.2 リンク変数の特性

リンク変数メモリには、次のデータフォーマット別リンク変数を使用してアクセスします。

データタイプ <sup>1)</sup>	説明	データフォーマット <sup>2)</sup>	バイト数 <sup>2)</sup>	インデックス $i$ <sup>3)</sup>	数値の範囲
UINT	$\$A\_DLB[i]$	BYTE	1	$i = n * 1$	0 ... 255
INT	$\$A\_DLW[i]$	WORD	2	$i = n * 2$	-32768 ... 32767
INT	$\$A\_DLD[i]$	DWORD	4	$i = n * 4$	-2147483648 ... 2147483647
REAL	$\$A\_DLR[i]$	REAL	8	$i = n * 8$	$\pm(2,2*10^{-308} \dots 1,8*10^{+308})$

1) パートプログラム/サイクルで使用される時のリンク変数のデータタイプ

2) リンク変数のデータフォーマット、リンク変数メモリ内でリンク変数によってアドレス指定されるバイト数

3) インデックス  $i$  に関して以下に注意してください。

- インデックス  $i$  は、リンク変数メモリの先頭からのバイトインデックスです。
- インデックスは、リンク変数メモリ内でアドレス指定されたバイトがデータフォーマット制限範囲になるように選択してください。⇒ インデックス  $i = n * \text{バイト数}$  (ここでは  $n = 0, 1, 2$ ) になります。
  - $\$A\_DLB[i]: i = 0, 1, 2, \dots$
  - $\$A\_DLW[i]: i = 0, 2, 4, \dots$
  - $\$A\_DLD[i]: i = 0, 4, 8, \dots$
  - $\$A\_DLR[i]: i = 0, 8, 16, \dots$

**書き込み**

リンク変数は実行プログラムに同期して書き込まれます。

**読み出し**

リンク変数の読み出し時に先読み停止がおこなわれます。

**確認**

リンク変数とリンク変数メモリに対して以下のチェックがおこなわれます。

- 数値範囲制限の監視
- フォーマット制限範囲へのアクセス
- リンク変数メモリ内の定義されたメモリ領域の監視

以下の誤りを防止する責任はユーザー/工作機械メーカーにあります。

- 不正なデータフォーマットによるアクセス
- 誤ったアドレス(インデックス *i*)へのアクセス
- 1つの NCU または複数の NCU の複数のチャネルによる同一データ項目の相互上書き
- 自身の NCU または別の NCU のチャネルによる更新前のデータ項目の読み出し

---

#### 注記

##### データの整合性

単独の NCU または複数の NCU の両方でリンク変数メモリ内でデータ適合性を維持する責任は、ユーザー/工作機械メーカーにあります。

---

### 3.2.2.3 書き込み要素

リンク変数メモリへの書き込みアクセス(たとえば\$A\_DLB[4] = 21)の場合、システム内で書き込みプロセスを管理するためにリンク変数書き込み要素が必要です。各補間周期で使用できる書き込み要素の最大数は、以下のマシンデータで設定します。

MD28160 \$MC\_MM\_NUM\_LINKVAR\_ELEMENTS

書き込み要素の最大数は各補間周期中に書き込むことができるリンク変数の個数を制限します。

### 3.2.2.4 書き込み時のダイナミック応答

リンク変数は実行プログラム同期制御で書き込まれます。新しい値は、次の補間周期の前に自身の NCU の他のチャネルによって読み出すことができます。この値は自身のチャネルの次のパートプログラムブロックで読み出すことができます。

リンクグループ内の他の NCU のチャネルは通常、2 補間周期後に新しい値を検出します。ただし、帯域幅に制限があると、書き込み要求をリンクグループ内の他の NCU への送信が遅れることがあります(通知遅延)。通知遅延の考えられる原因を以下に示します。

- 1 補間周期内に多数のリンク変数の書き込み
- 同一補間周期内でのリンク変数の書き込みと軸コンテナ回転の要求
- 同一補間周期内でのリンク変数の書き込みとアラームの通知

3.2 NCU リンク

3.2.2.5 システム変数

NC 別システム変数

識別子	意味
\$AN_LINK_TRANS_RATE_LAST	最後の補間周期でまだ未処理の書き込み要求の数。
\$AN_LINK_TRANS_RATE_LAST_SUM[<n>]	指定された NC(NCU 番号)への送信方向での、最後の補間周期でまだ未処理の書き込み要求の数。
\$AN_LINK_CONN_SIZE_LINKVAR	リンク変数の書き込み要求に対して伝送されるバイト数。
\$AN_LINK_CONN_SND[<n>]	補間周期毎に現在の NCU から指定された NCU に伝送できる最大バイト数。
\$AN_LINK_CONN_RCV[<n>]	補間周期毎に指定された NCU から現在の NCU に伝送できる最大バイト数。
<n>: 当該の NCU の MD12510 \$MN_NCU_LINKNO に従った NCU 番号	

注記: \$AN\_LINK\_CONN\_SIZE\_LINKVAR

リンク変数の書き込みにより、\$AN\_LINK\_CONN\_SIZE\_LINKVAR で指定されたバイト数が非周期的なリンク通信によって伝送されます。この数は、リンク変数のフォーマットに関係しません。

補間周期毎に指定された NCU に伝送できる最大書き込み要求数は、次のように計算されます。

$$\text{最大書き込み要求数} = \$AN\_LINK\_CONN\_SND[<n>] / \$AN\_LINK\_CONN\_SIZE\_LINKVAR$$

チャンネル別システム変数

識別子	意味
\$A_LINK_TRANS_RATE <sup>1)</sup>	現在の補間周期内でまだ伝送可能な書き込み要求の数。
1) 用途例、以下の章を参照してください。 "書き込み要求の同期 (ページ 107)"	

### 3.2.2.6 書き込み要求の同期

特定の用途で、リンク変数の新しい値をリンクグループ内の他の NCU に正確に 2 補間周期で伝送する必要がある場合、リンク変数への書き込みはシンクロナイズドアクションでおこなってください。シンクロナイズドアクションでは、リンク変数への書き込みは、実際の補間周期で書き込み要求をまだ実行できる場合にもおこなわれます。

**\$A\_LINK\_TRANS\_RATE** システム変数には、現在の補間周期でさらに伝送できる書き込み要求数を含みます。

後述の例では、リンク変数、データタイプ **WORD (2 バイト)** とリンク変数、データタイプ **DWORD (4 バイト)** が書き込まれます。

#### プログラム例

```
N120 WHEN $A_LINK_TRANS_RATE > 0 DO $A_DLW[0] = 9
N125 WHEN $A_LINK_TRANS_RATE > 0 DO $A_DLD[2] = 7
N130 G4 F1
```

N120 内のシンクロナイズドアクションは、同じ補間周期内でリンクグループの他の NCU に書き込み要求を伝送できる場合にもおこなわれます。この場合、同じ補間周期内で **\$A\_LINK\_TRANS\_RATE** システム変数も減算され、次の N125 ブロックのシンクロナイズドアクションで更新値が利用可能になります。

### 3.2.2.7 例: リンク変数メモリの構成

リンク通信に対して次のデータが定義されます。

データフォーマット	番号	データ当たりのバイト数	必要なバイト数
BYTE	2	1	2
WORD	1	2	2
DWORD	3	4	12
REAL	1	8	8
リンク変数メモリに必要なサイズ			<b>24</b>

#### メモリ構成

データはデータフォーマット制限を考慮して、以下の通りリンク変数メモリに配列されます。

3.2 NCU リンク

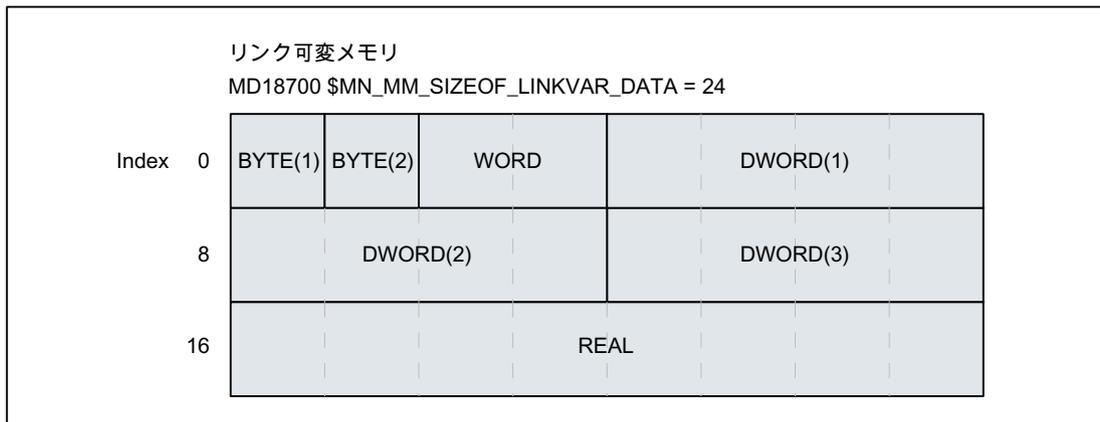


図 3-8 例:リンク変数メモリの構成

**注記**

**メモリ構成**

リンク変数メモリのデータは常にランダムに配列され、そのため異なるように見えることがあります(ただし、さらにデータフォーマット制限は考慮されます)。

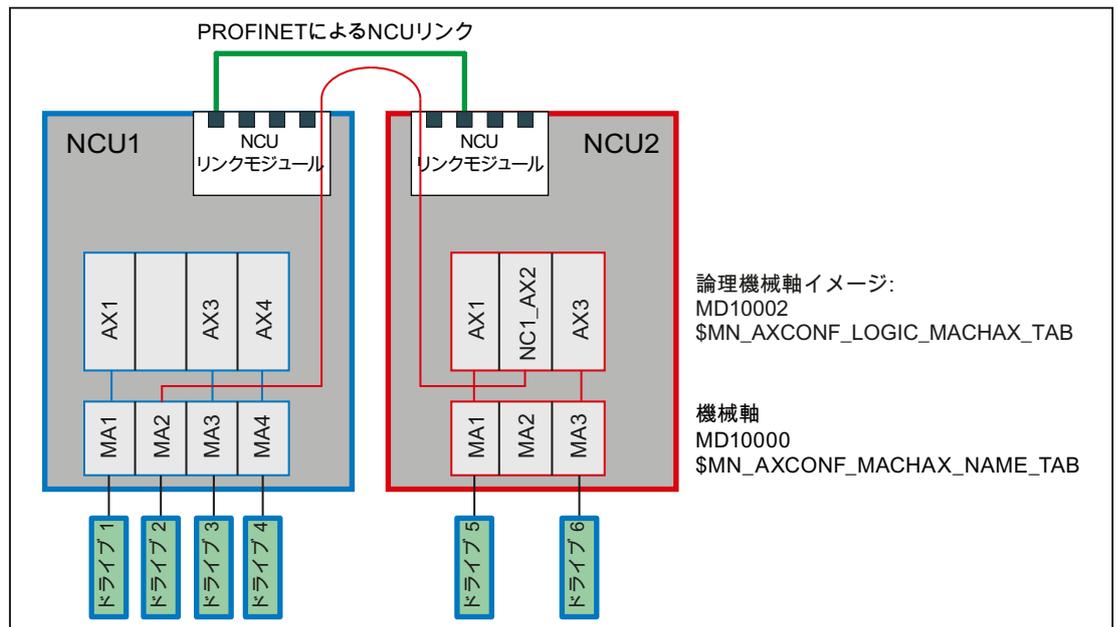
リンク変数によるアクセスは、定義されたメモリ構成に応じて以下の通りプログラム指令してください。

プログラムコード	説明
\$A_DLB[0]	; BYTE (1)
\$A_DLB[1]	; BYTE (2)
\$A_DLW[2]	; WORD
\$A_DLD[4]	; DWORD (1)
\$A_DLD[8]	; DWORD (2)
\$A_DLD[12]	; DWORD (3)
\$A_DLR[16]	; REAL

### 3.2.2.8 例: ドライブデータの読み取り

#### 処理

システムには、2基のNCU(NCU1/NCU2)が含まれています。このNCUは、NCUリンク経由で接続されています。NCU1(ドライブ2)のMA2機械軸は、NCU2のリンク軸として補間モードで移動します。ドライブ2の電流フィードバック値は、評価のためにNCU1からNCU2に伝送すべきです。次の図はシステムの基本構成を示しています。



#### 必要条件

ドライブ2(NCU1/MA2)の電流フィードバック値は、\$VA\_CURRシステム変数によって読み取ることができます。PROFIdriveベースのドライブの場合、この目的のために以下のマシンデータ項目を設定する必要があります。

MD36730 \$MA\_DRIVE\_SIGNAL\_TRACKING = 1 (ドライブの追加フィードバック値の取得)

マシンデータを設定することで、以下のドライブのフィードバック値を使用できるようになります。

- \$AA\_LOAD, \$VA\_LOAD (ドライブ容量の負荷率[%])
- \$AA\_POWER, \$VA\_POWER (ドライブの有効電力[W])

### 3.2 NCU リンク

- \$AA\_TORQUE, \$VA\_TORQUE (ドライブのトルク指令値[Nm])
- \$AA\_CURR, \$VA\_CURR (軸または主軸の電流フィードバック値[A])

## プログラミング

### NCU1

軸 2 の電流フィードバック値\$VA\_CURR(NCU1/MA2)をリンク変数メモリの最初の 8 バイトに、リンク変数\$A\_DLR[ 0 ] (REAL 値)を使用して補間周期内で周期的に書き込むには、静的シンクロナイズドアクションを使用します。

#### プログラムコード

```
IDS=1 WHENEVER TRUE DO $A_DLR[0]=$VA_CURR[MA2]
```

### NCU2

静的シンクロナイズドアクションにより、伝送された電流フィードバック値は、リンク変数\$A\_DLR[0]によって補間周期で周期的に読み取られます。電流フィードバック値が 23 A より大きい場合、アラーム 61000 が表示されます。

#### プログラムコード

```
IDS=1 WHEN $A_DLR[0] > 23.0 DO SETAL(61000)
```

## 3.2.3 リンク軸

### 3.2.3.1 概要

リンク軸は、機械軸が物理的に接続されている NCU とは異なる NCU によって指令値が生成される対象である機械軸として指定されます。軸コンテナ(「軸コンテナ (ページ 117)」の章を参照してください)と組み合わせたリンク軸は、複数の NCU を備えたロータリインデックスマシンなどの複雑なシステム内で、リンクグループの NCU の機械軸の代替使用ができます。

次の図は、NCU1 に接続する MA1 機械軸の例を示しています。MA2 機械軸は、NCU2 に接続されています。X チャネル軸と Z チャネル軸は、NCU1 のチャンネル内で実行するパートプログラムによって補間モードで移動します。この指令値は、NCU1 補間器で生成されます。機械軸 MA1 では、この指令値は NCU1 の位置制御に伝送されます。機械軸 MA2 では、この指令値は NCU リンクによって NCU2 の位置制御に伝送され、そこからドライブに出力されます。

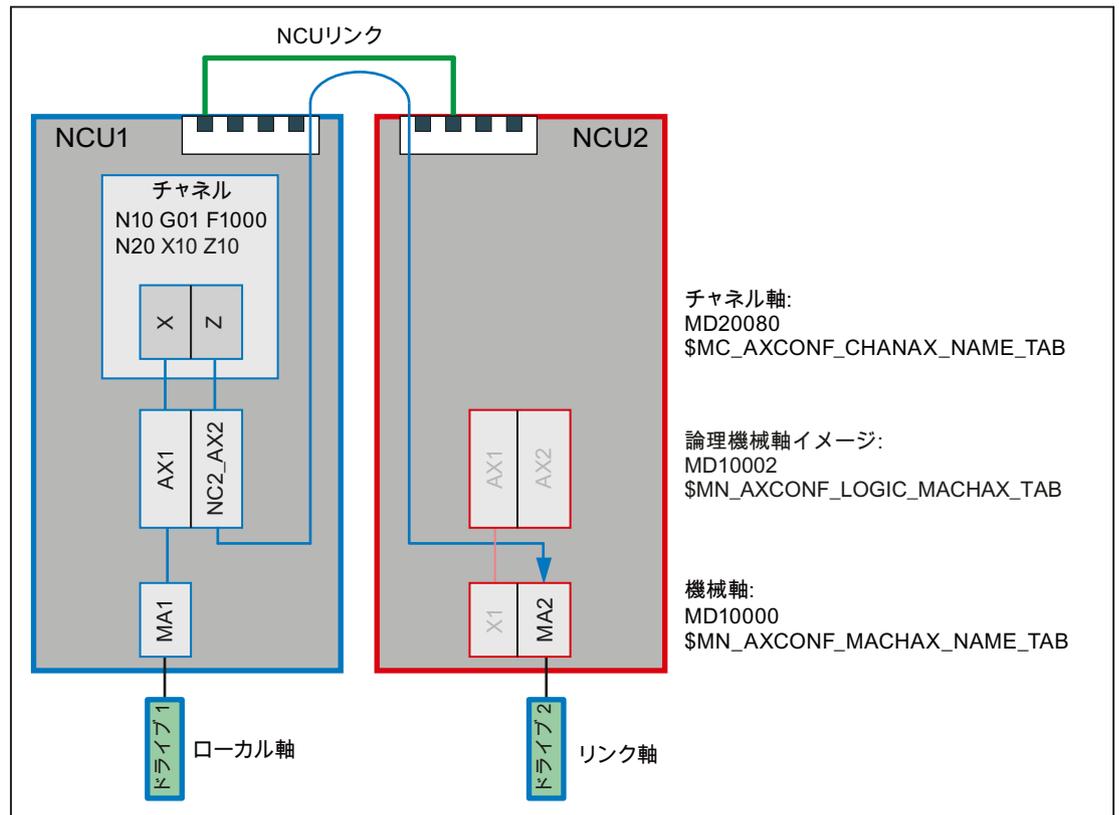


図 3-9 リンク軸

### 必要条件

リンク軸を使用するには、「リンク通信 (ページ 92)」の章に従って定義されたリンク通信が必要です。

### ホーム NCU

リンク軸のホーム NCU は、リンク軸が機械軸として物理的に接続された NCU です。リンク軸の軸 NC/PLC インタフェース信号の位置制御と交換は常に、ホーム NCU 上でおこなわれます。指令値の生成は原則として、リンクグループの任意の NCU 上でおこなわれます。

上記の図で

- NCU1: 機械軸 MA1 のホーム NCU
- NCU2: 機械軸 MA2 のホーム NCU

3.2 NCU リンク

3.2.3.2 リンク軸名称

リンク軸名称は、機械軸が物理的に接続されているホーム NCU の識別子と、一般機械軸名称 AXn から成ります。

NC<ID>\_<軸>

<ID>: MD12510 \$MN\_NCU\_LINKNO に従ったリンクグループの NCU の識別  
「パラメータ設定 リンク通信 (ページ 100)」の章を参照してください。

<軸>: 一般機械軸名称 AX1、AX2、AX3、...

3.2.3.3 パラメータ設定

割り当て: リンク軸に対するジオメトリ軸または付加軸

直接割り当て

ジオメトリ軸または付加軸は、以下のリンク軸名称の指定によって、論理機械軸イメージのリンク軸に直接割り当てることができます。

MD10002 \$MA\_AXCONF\_LOGIC\_MACHAX\_TAB[<軸>] = <リンク軸>

パラメータ	意味
<軸>:	機械軸インデックス: 0、1、2、... (機械軸の最大数 - 1)
<リンク軸>:	リンク軸名称:NCx_AXy、「リンク軸名称 (ページ 112)」の章を参照してください

間接的な割り当て

ジオメトリ軸または付加軸は、コンテナスロットの指定によって、論理機械軸イメージのリンク軸に間接的に割り当てることができます。この場合、前述のように、コンテナスロットにはリンク軸の実際の名称が含まれます。これは、コンテナリンク軸と呼ばれています。参照: "軸コンテナ (ページ 117)".

同期指令値出力の章

指令値を生成する NCU のリンク軸指令値をホーム NCU に伝送するときに、1 補間周期の遅延が生じます。ローカル軸とリンク軸の補間用の指令値が正確に同時にドライブに出力されるようにするには、この遅延を補正する必要があります。このために、指令値を生成する NCU で、補間器と位置コントローラ間のバッファのバッファ要素数は、ホーム NCU のバッファ要素数よりも 1 要素大きく設定してください。

MD18720 \$MN\_MM\_SERVO\_FIFO\_SIZE = 3

## 例

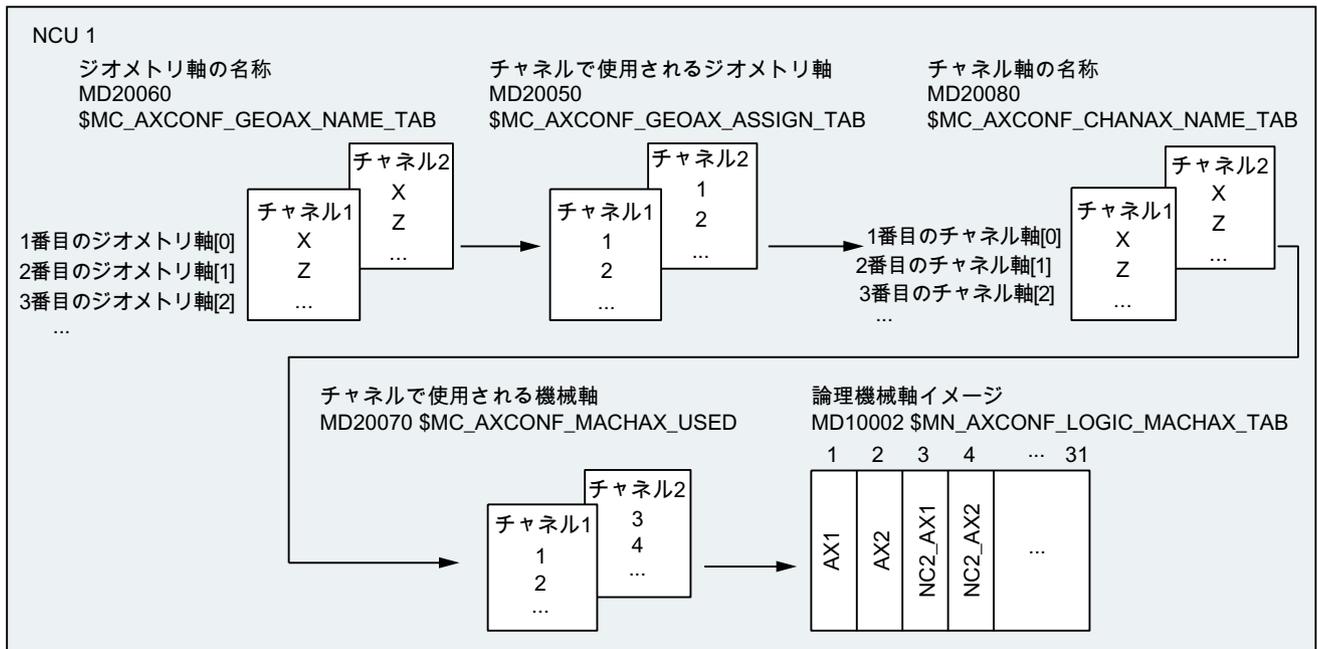


図 3-10 例: リンク軸のパラメータ設定

**チャンネル 1**

NCU1 のローカル AX1/AX2 機械軸は、X/Z ジオメトリ軸に割り当てられます。

**チャンネル 2**

NCU2 の NC2\_AX1/NC2\_AX2 リンク軸は、X/Z ジオメトリ軸に割り当てられます。

**3.2.3.4 主軸の補助機能出力**

プログラム実行中、および「プログラムテストによる検索」(SERUPRO)によるブロック検索後に、主軸がプログラム指令され、ホーム NCU では軸別となっている NCU で、事前定義補助機能 S、M3、M4、M5、M19、および M70 がチャンネル別に出力されます。

**チャンネル別補助機能の出力**

- DB21、... DBW68 (M 機能の拡張アドレス)
- DB21、... DBD70 (M 機能 1)
- DB21、... DBW98 (S 機能 1 の拡張アドレス)
- DB21、... DBD100 (S 機能 1)

### 3.2 NCU リンク

#### 軸別補助機能の出力

- DB31、... DBD78 (軸の F 機能)
- DB31、... DBW86 (主軸の M 機能)
- DB31、... DBD88 (主軸の S 機能)

#### 参照先

補助機能出力の詳細については、『機能マニュアル、基本機能』、「PLC(H2)に対する補助機能出力」の章を参照してください。

#### 3.2.3.5 補足条件

#### 機械軸の最大数

リンク軸の使用時にも前記と同様に、各 NCU タイプについて、機械軸に加えて同時使用可能なジオメトリ軸と付加軸の最大数がさらに使用できます。

#### 「マスタリンク軸」機能と「リンク軸」機能

「マスタリンク軸」機能と「リンク軸」機能ではマシンデータ MD18720 \$MN\_MM\_SERVO\_FIFO\_SIZE で異なる設定を必要とするために、リンクグループ内で同時に使用できません。

#### アラーム:一般的な動作

リンク軸のホーム NCU の位置制御レベルで異常が検出され、関連するアラームに応答として「NC 準備未完」が含まれない場合、アラームは NCU リンクによって指令値を生成する NCU と、その後出力に伝送されます。

#### アラーム:非常停止に対する応答

NC/PLC インタフェース経由で非常停止要求が NCU に対して有効になった場合、この NCU に物理的に接続するすべての軸が「フォローアップ」モードに切り替わります。これはまた、指令値が他の NCU によって現在生成されているリンク軸にも影響します。このとき、これらの NCU ではこれ以降の実加工運転が不可との前提で、関連するすべての軸の移動を停止する追加アラームが生成されます。

### アラーム応答

NC のリセットによって、この追加アラームの確認をおこなってください。このタイミングで元のアラームがまだ発生中である場合、追加アラームには応答できますが、元のアラームに応答されるまで、軸の移動またはプログラムの開始を防止する追加自己クリアアラームが生成されます。

### アラーム: 「NC 準備未完」アラーム応答に対する応答

リンク軸のホーム NCU の位置制御レベルで異常が検出され、関連するアラームに応答として「NC 準備未完」が含まれない場合、アラームは NCU リンクによって指令値を生成する NCU と、その後出力に伝送されます。また、ホーム NCU 上でアラームが出力されます。

このとき、これらの NCU ではこれ以降の実加工運転が、指令を生成した NCU では不可との前提で、関連するすべての軸の移動を停止する追加アラームが生成されます。

### アラーム応答

「アラーム:非常停止に対する応答」の「アラームの応答」を参照してください。

### アラーム: 「モードグループ準備未完」アラーム応答の動作

複数のチャンネルを含むモードグループ内で異常が検出された場合で、関連するアラームに応答として「モードグループ準備未完」が含まれる場合、これによってモードグループのすべてのチャンネルで移動動作が停止します。一般に移動動作が相互に独立している場合、下記のマシンデータ項目による応答は「チャンネル準備未完」に変更できます。

MD11412 \$MN\_ALARM\_REACTION\_CHAN\_NOREADY = TRUE

### 作用

NC/PLC インタフェースでは、信号 DB11 DBX26.3(モードグループ準備完了)の代わりに、信号 DB21、DBX36.5(チャンネル準備完了)がリセットされます。

### 長所

アラーム応答は引き続き、異常が検出されたチャンネルに限定されます。必要に応じて、PLC ユーザープログラムは追加応答を開始できます。

### 必要条件

「モードグループ準備未完」よりも高い優先順位のアラーム応答は存在しません。

### 3.2 NCU リンク

#### 補正

次の補正は使用できません。

- リンク軸:象限突起補償(QEC)
- コンテナリンク軸真直度補正(CEC)

#### リンクグループ内の NCU のスイッチオフ

リンクグループ内の NCU のスイッチをオフにした場合、リンクグループの他のすべての NCU 上で加工がキャンセルされ、アラームが発行されます。

#### NCU グループの起動

リンクグループ内の NCU 上で NC リセットが起動した場合、リンクグループ内の他の全 NCU にもそれが伝送されるので、リンクグループのすべての NCU でウォーム再起動がおこなわれます。

#### ニブリングとパンチングの用途

ニブリングとパンチングに必要な高速入/出力部は、パートプログラムを処理して、軸が補間される NCU に接続して、パラメータ設定してください。「高速ニブリングとパンチング」の命令(例: PONS と SONS)は、リンク軸で使用できません。

#### フレーム

リンク軸は、フレーム命令で、それがジオメトリ軸である場合に限り許可されます。このフレーム命令は、現在、リンク軸が割り当てられているチャンネルのジオメトリのみを変更します。

#### 速度/トルク連結、マスタスレーブ

マスタスレーブグループのすべての軸/主軸のドライブは、同じ NCU に接続してください。ただしマスタ軸は、別の NCU のチャンネルのリンク軸として移動できます。

## 3.2.4 軸コンテナ

### 3.2.4.1 概要

軸コンテナは、パラメータ設定された要素数を含む循環データ構造です。これらの要素は軸コンテナと一緒に、スロット(スロット1、スロット2、...スロットn)として指定されます。スロットにより、ジオメトリおよび(または)付加軸を機械軸に可変に割り当てることができます。スロットの入力では、NCU ローカル機械軸(コンテナ軸)またはリンク軸(コンテナリンク軸)を参照できます。

下記の図は、4つのスロットを備える軸コンテナを示しています。軸コンテナの現在の位置で、コンテナ軸は下記の機械軸を意味します。

コンテナ軸	機械軸
CT1_SL1	NCU 1: AX1
CT1_SL2	NCU 1: AX2
CT1_SL3	NCU 2: AX1
CT1_SL4	NCU 2: AX2

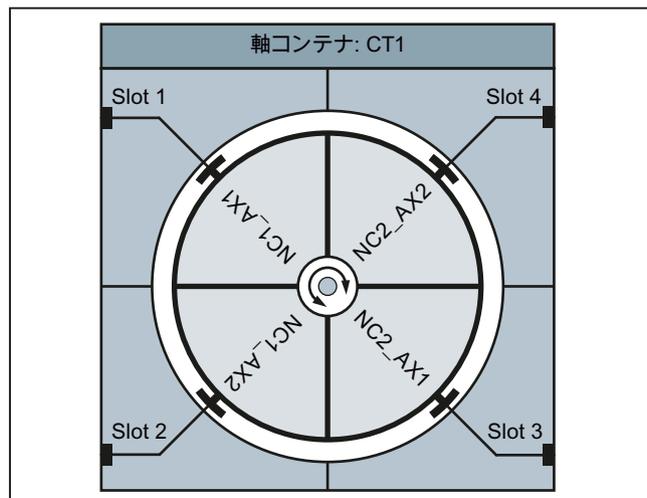


図 3-11 例: 4つのスロットを備える軸コンテナ CT1

#### 規則

軸コンテナに関して下記の規則に従ってください。

- 軸コンテナのすべての機械軸は、同時に1つのチャンネル軸にのみ割り当てることができます。
- 軸コンテナの複数のスロットが同じ機械軸を参照することはできません。

### 3.2 NCU リンク

- 同時に 1 つのチャンネルのみが機械軸に直接またはコンテナ軸経由で書き込むことが許可されます。
- チャンネルの複数のジオメトリ軸および(または)付加軸を軸コンテナのコンテナ軸に割り当てすることもできます。

#### 割り当て: ジオメトリ軸または付加軸 → コンテナ軸

論理機械軸イメージ MD10002 \$MN\_AXCONF\_LOGIC\_MACHAX\_TAB では、コンテナとスロットによって、コンテナ軸は機械軸としてジオメトリ軸または付加軸(たとえば、CONTAINER 「CT1」、スロット 「1」)に割り当てることができます。

```
MD10002 $MN_AXCONF_LOGIC_MACHAX_TAB[ n ] = CT1_SL1
```

ジオメトリ軸または付加軸の移動中に、このタイミングでスロット 1 に割り当てられた機械軸が移動します。

#### 軸コンテナの回転

軸コンテナ上の関連するすべてのチャンネルがプログラム命令によりイネーブルを許可している時に、常に軸コンテナの回転がおこなわれます。軸コンテナの回転後、他の機械軸がチャンネルのジオメトリ軸または付加軸に割り当てられます。

セッティングデータは、軸コンテナ回転の分割数を指定します。

「パラメータ設定 (ページ 120)」の章を参照してください。

#### 軸コンテナ名称

下記のプログラム命令は、軸コンテナ名称(<軸コンテナ>)によってアドレス指定できます。

- プログラム命令:
  - AXCTSWE (<軸コンテナ>)
  - AXCTSWED (<軸コンテナ>)
  - AXCTSWEC (<軸コンテナ>)

以下の名称が選択可能です。

- CT<コンテナ番号>: 軸コンテナ番号を、CT 文字と組合わせて記述します。  
例: CT3
- <コンテナ名称>: MD12750 \$MN\_AXCT\_NAME\_TAB で設定した個々の軸コンテナの名称です  
例: A\_CONT3
- <軸名称>: 関係するチャンネル内で既知のコンテナ軸の軸名称です。

## 自動待機

次の事象のいずれかが発生した場合、要求された軸コンテナ回転の完了まで自動的に待機します。

- このチャンネルのこの軸コンテナに割り当てられたコンテナ軸を移動させるパートプログラム言語命令
- 対応するコンテナ軸の GET (<チャンネル軸の名称>) 命令
- この軸コンテナの次の AXCTSWE (<軸コンテナ>) 命令

---

### 注記

IC (0) の場合でも、必要に応じて同期制御を含む待機が生じます(アブソリュート指令がグローバルに設定されている場合でも、インCREMENTAL指令に従ったアドレス指定でブロック毎の切り替えがおこなわれます)。

---

## 軸位置の同期制御

コンテナ回転後にチャンネルに割り当てられた新しいコンテナ軸が、以前の軸と同じアブソリュート機械位置を持たない場合、コンテナは新しい位置(内部 REORG)で原点同期されます。

---

### 注記

SD41700 \$SN\_AXCT\_SWWIDTH[<軸コンテナ>] は、新しい設定についてのみ更新されます。RTM/MS のインCREMENTAL回転後に初期位置の直前の繰り返し位置に達した場合、通常はコンテナが**前進方向**に回転することで、再度コンテナの初期位置に到達できます。ただしドラムまたは回転テーブルは、検出器ケーブルと電源ケーブルのねじれを避けるために、元の位置に**戻**してください。

---

3.2 NCU リンク

下記も参照

システム変数 (ページ 129)

プログラミング (ページ 127)

3.2.4.2 パラメータ設定

マシンデータ

NC 別マシンデータ

番号	識別子 \$MN_	意味
MD12750	AXCT_NAME_TAB	軸コンテナ名称
MD12760	AXCT_FUNCTION_MASK.Bit x	軸コンテナ別の機能
MD1270x	AXCT_AXCONF_ASSIGN_TA Bx	軸コンテナのスロットへの機械軸の割り当て
MD18720	MM_SERVO_FIFO_SIZE	IPO/SERVO データバッファのサイズ 注: クロス NCU 軸コンテナに関与するすべての NCU 上で値 3 を設定してください。

軸コンテナ名称

MD12750 \$MN\_AXCT\_NAME\_TAB[ <インデックス> ] = "<名称>"

パラメータ	意味
<インデックス>: >:	0, 1, ... 最大軸コンテナインデックス
<名称>:	軸コンテナ名称(例: CT1)

## 軸コンテナ別の機能

MD12760 \$MN\_AXCT\_FUNCTION\_MASK.Bit x = &lt;値&gt;

パラメータ	<値>	意味
ビット 0:	0	直接の軸コンテナ接続(ACTSWED)では、その他のすべてのチャンネルがリセット状態になる必要があります。
	1	直接軸コンテナ接続(ACTSWED)では、軸コンテナの軸に対して補間権限を持つ他のチャンネルのみがリセット状態にしてください。

このマシンデータは、軸コンテナ用の機能を有効にするために使用されます。

## 軸コンテナのロットへの機械軸の割り当て

MD1270x \$MN\_AXCT\_AXCONF\_ASSIGN\_TABx[&lt;インデックス&gt;] = &lt;軸&gt;

パラメータ	意味
x:	1 ... 軸コンテナの最大数
<インデックス>:	0、1、...最大ロットインデックス
<軸>:	ローカル機械軸の名称(例: AX1)
	リンク軸名称、「概要 (ページ 110)」の章を参照してください。

軸コンテナ内のロットは、ロットインデックス 0 から昇順に隙間なく割り当ててください。

## セッティングデータ

## 軸コンテナ回転の分割数

SD41700 \$SN\_AXCT\_SWWIDTH[&lt;インデックス&gt;]=&lt;分割数&gt;

パラメータ	意味
<インデックス>:	0, 1, ... 最大軸コンテナインデックス
<分割数>:	軸コンテナが回転するロットの数

3.2 NCU リンク

軸コンテナ回転の図

軸コンテナの回転は、プログラム命令によって有効になります。「プログラミング (ページ 127)」の章を参照してください。

「軸コンテナ回転、図 1」の図の左側で、リンク軸 NCU1\_AX1 は軸コンテナの初期設定で、スロット 1 に割り当てられています。

分割 1 による回転の後(右側)、NCU2\_AX2 リンク軸がスロット 1 に割り当てられます。

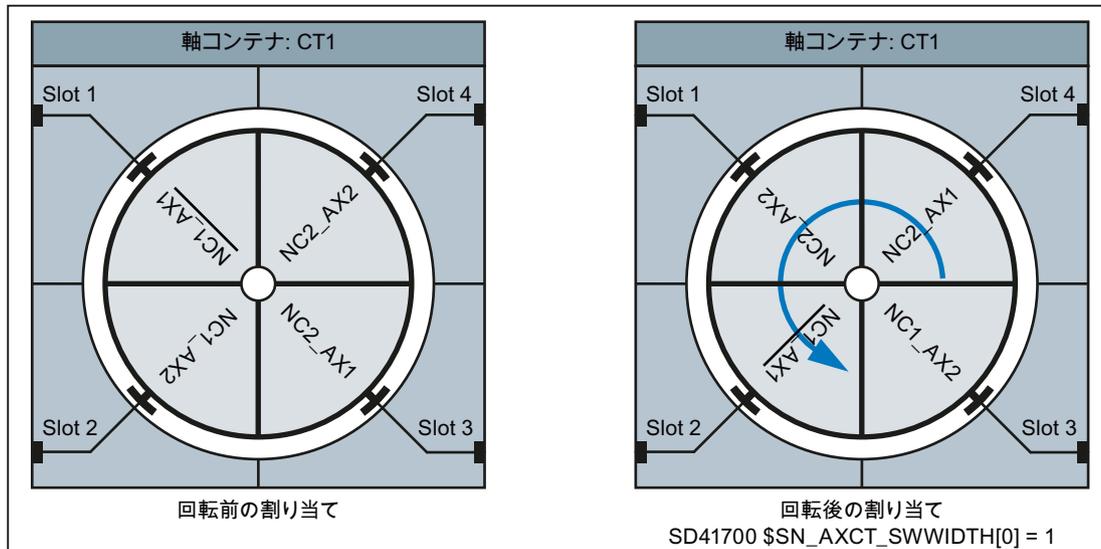


図 3-12 軸コンテナ回転、図 1

上記の初期設定に基づいて、2 のインCREMENTによる回転後、リンク軸 NCU2\_AX1 はスロット 1 に割り当てられます(図「軸コンテナ回転、図 2」の左側)。

上記の初期設定に基づいて、-1 のインCREMENTによる回転後、リンク軸 NCU1\_AX2 はスロット 1 に割り当てられます(図「軸コンテナ回転、図 2」の右側)。

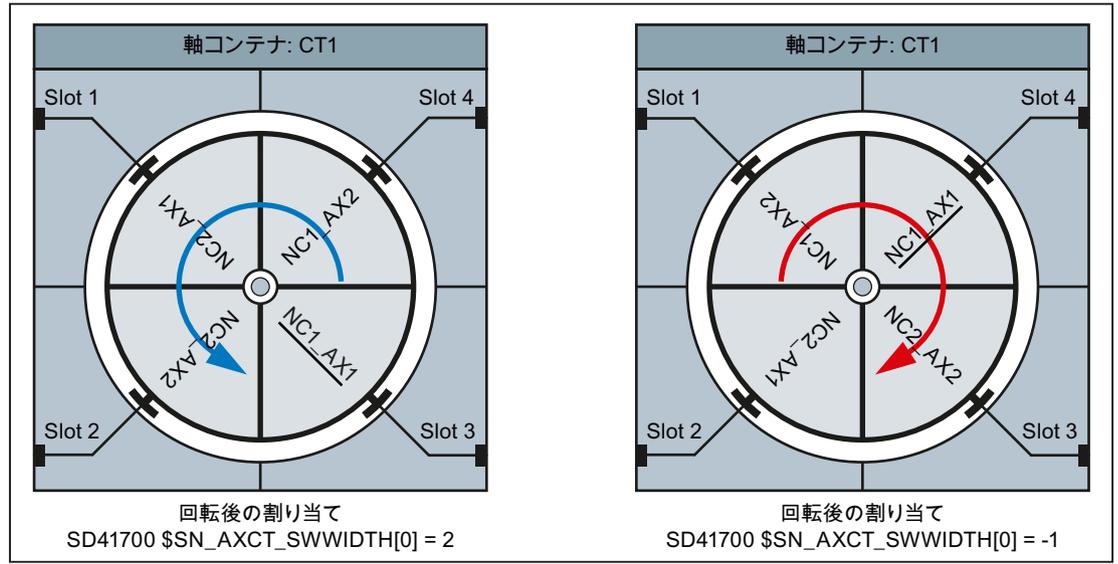


図 3-13 軸コンテナ回転、図 2

### コンテナリンク軸を備える軸コンテナ

リンクグループ(MD12510 \$MN\_NCU\_LINKNO == 1)のマスター NCU 上で、コンテナリンク軸を含む軸コンテナのパラメータ設定をおこなう必要があります。

### 軸マシンデータの整合

コンテナ軸では、「CTEQ」(コンテナ一致)属性によりマーキングされたすべての軸マシンデータがすべてのコンテナ軸について同じ値を持つ必要があります。任意の異なる値に自動的に揃えられます。

#### コントローラの起動

コントローラ起動時に、すべてのマシンデータが 1 番目のスロットのコンテナ軸の値に揃えられます。マシンデータ項目の値が変更された場合、下記の通知が表示されます。「軸コンテナ<n>の軸の軸マシンデータが設定されました」

#### マシンデータの有効化

コンテナ軸のマシンデータ項目が変更された場合、新しい値はまた、その他すべてのコンテナ軸ですぐに使用可能になります。下記の通知が表示されます。「注意:すべてのコンテナ軸についてこの MD が設定されます」。

3.2 NCU リンク

スロットの変更

軸コンテナのスロットに別の機械軸(MD127xx AXCT\_AXCONF\_ASSIGN\_TAB<x>)が割り当てられた場合、下記の通知が表示されます。「軸コンテナ<n>の軸のマシンデータは次の起動時に設定されます」。

注記

コンテナリンク軸

コンテナリンク軸では、軸コンテナ上の関連するリンクグループのすべての NCU について、マシンデータの整合がおこなわれます。

パラメータの例

前提条件

NCU	コンポーネント
NCU1:	チャンネル 1、X/Z ジオメトリ軸 → 1 番目/2 番目のチャンネル軸 チャンネル 2、X/Z ジオメトリ軸 → 1 番目/2 番目のチャンネル軸 機械軸:AX1、AX2 4つのスロットを備える CT1 軸コンテナ
NCU2:	機械軸:AX1、AX2

## パラメータ設定:NCU1

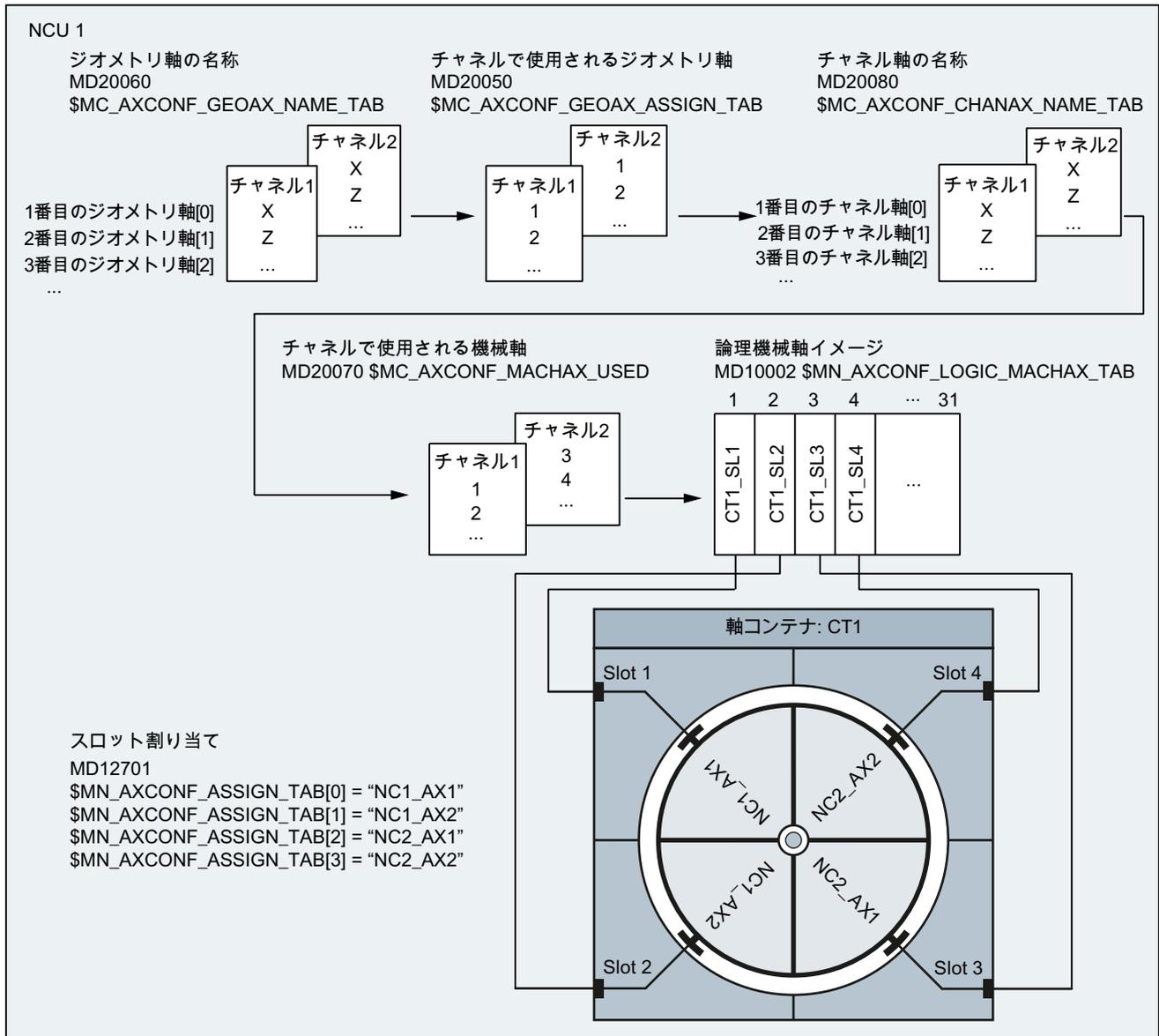


図 3-14 例:チャンネル軸と軸コンテナのパラメータ設定

## 働き

NCU1 の 1 番目と 2 番目のチャンネルにあるジオメトリ軸 X と Z をプログラミングすることによって、下記の軸がコンテナの現在の位置に移動します。

- NCU1 のローカル AX1 と AX2 機械軸。
- コンテナリンク軸として、NCU2 の AX1 と AX2 機械軸。

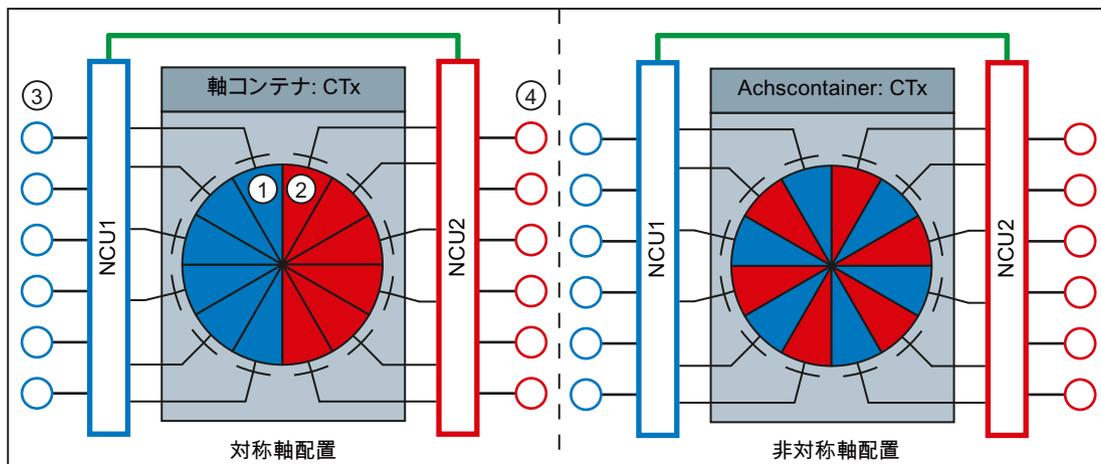
軸コンテナの回転については、「プログラミング (ページ 127)」の章を参照してください。

3.2 NCU リンク

パラメータ設定に関する注記

コンテナ軸の配置と通信の負荷率

軸コンテナと組み合わせて他の NCU(リンク軸)の軸と交互に移動する複数の NCU を備えるシステムの場合、軸コンテナ内でのリンク軸の配置方法と配置タイプにより、リンク通信の負荷率が決定します。



- ① 青いスロット:NCU1 に接続するドライブを意味します
- ② 赤いスロット:NCU2 に接続するドライブを意味します
- ③ ドライブは NCU1 に接続します
- ④ ドライブは NCU2 に接続します

図 3-15 軸配置

- 対称軸配置  
対称軸配置の場合は、各 NCU は最初にローカル軸のみを移動します。つまり、リンク通信は発生しません。すべての NCU がリンク軸のみを移動する場合、軸コンテナの切り替え毎に、リンク通信の負荷率が最大になります。
- 非対称軸配置  
非対称軸配置では、各 NCU は最初からローカル軸とリンク軸を移動します。対称軸配置とは異なり、リンク通信の負荷率は一定の平均になります。

### ドライブの配置と通信の負荷率

軸コンテナと組み合わせて別の NCU(リンク軸)の軸を交互に移動する複数の NCU を備えるシステムでは、NCU に接続するドライブの配置により、リンク通信の負荷率が決定します。

- 対称ドライブ配置

対称ドライブ配置の場合、軸コンテナによってアドレス指定されたドライブが両方の NCU に接続されます。この配置を考慮したうえで、可能なドライブの最大数はまだ、両方の NCU 上の論理機械軸イメージ(LAI)によってアドレス指定できます。

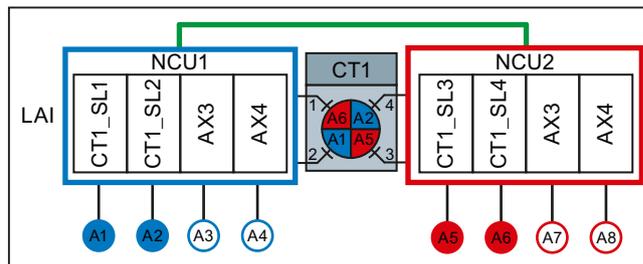


図 3-16 対称ドライブ配置

- 非対称ドライブ配置

非対称ドライブ配置の場合、軸コンテナによってアドレス指定されたドライブは NCU1 にのみ接続されます。この配置を考慮したうえで、可能なドライブの最大数は、NCU2 上の論理機械軸イメージ(LAI)によってのみアドレス指定できます。最大数から NCU2 によって使用されているドライブ数を減算した数のみ、NCU1 の LAI によってアドレス指定できます。また、NCU1 上で最大数のドライブを使用するには、それらを NCU2 に接続し、NCU リンクによって NCU1 からアドレス指定してください。これにより、周期的なリンク通信の負荷が高まります。

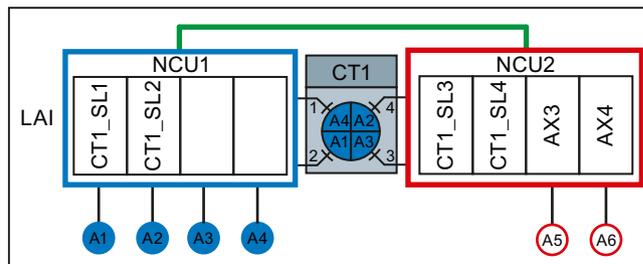


図 3-17 非対称ドライブ配置

### 3.2.4.3 プログラミング

#### 機能

AXCTSWED 命令または AXCTSWE 命令を使用して、指定した軸コンテナの回転を有効にします。

### 3.2 NCU リンク

AXCTSWEC 命令により、以前に設定された軸コンテナ回転のイネーブルはキャンセルされます。

#### 構文

AXCTSWE (<ID>)  
AXCTSWED (<ID>)  
AXCTSWEC (<ID>)

#### 意味

- AXCTSWE: 軸コンテナの回転を有効にします  
プログラムの処理は AXCTSWE で停止しません。  
軸コンテナ上で関連するすべてのチャンネルが有効になるとすぐに、回転がおこなわれます。  
軸コンテナ回転のステップ値は、SD41700 \$SN\_AXCT\_SWWIDTH セッティングデータによって設定されます(「パラメータ設定 (ページ 120)」の章を参照してください)。
- AXCTSWED: 軸コンテナに含まれる他のチャンネルを考慮しないで軸コンテナの回転を有効にします  
軸コンテナ回転のステップ値は、SD41700 \$SN\_AXCT\_SWWIDTH セッティングデータによって設定されます(「パラメータ設定 (ページ 120)」の章を参照してください)。
- 注**
- パートプログラムまたはシンクロナイズドアクションのセットアップを簡単にするための命令タイプ。
  - 軸コンテナに関連する他チャンネルに関する特性は、MD12760 \$MN\_AXCT\_FUNCTION\_MASK、ビット 0 によって指定できます。「パラメータ設定 (ページ 120)」の章を参照してください。
- AXCTSWEC: 軸コンテナの回転のイネーブルをキャンセルします
- 注**  
回転がまだ起動していない場合にのみ、軸コンテナの回転のイネーブルはキャンセルできます。
- \$AN\_AXCTSWA[<軸コンテナ>] == 0**  
「システム変数 (ページ 129)」の章を参照してください

<ID>:	軸コンテナまたはコンテナ軸の名称:
CT<番号>:	軸コンテナの初期設定の識別子: MD12750 \$MN_AXCT_NAME_TAB 例:CT1
<コンテナ>:	軸コンテナのユーザー別名称: MD12750 \$MN_AXCT_NAME_TAB 例:CONTAINER_1
<軸>:	チャンネル内の既知のコンテナ軸の名称

## 参照先

シンクロナイズドアクションでの AXCTSWEC 命令の使用について詳細は、以下を参照してください。

『シンクロナイズドアクション機能マニュアル』、「詳細説明」 > 「シンクロナイズドアクションの動作」 > 「軸コンテナ回転のイネーブルのキャンセル(AXCTSWEC)の章」

### 3.2.4.4

#### システム変数

##### コンテナ用システム変数

システム変数	説明
\$AC_AXCTSWA[<ID>]	軸コンテナ回転のチャンネル別の状態です。
\$AN_AXCTSWA[<ID>]	軸コンテナ回転の NCU 別の状態です。
\$AN_AXCTSWE[<ID>]	軸コンテナの回転のスロット別の状態です。
\$AN_AXCTAS[<ID>]	軸コンテナの切り替えに使用されるスロットの数です。
ID:	軸コンテナの名称またはコンテナ軸の名称

##### NC 別システム変数

システム変数	説明
\$AN_LAI_AX_IS_AXCTAX <sup>1)</sup>	状態: LAI 軸 == コンテナ軸 軸コンテナに関する論理機械軸イメージ (MD10002 \$MN_AXCONF_LOGIC_MACHAX_TAB) の機械軸の状態。
\$AN_LAI_AX_IS_LEADLINKAX <sup>1)</sup>	状態: LAI 軸 == マスタリンク軸

3.2 NCU リンク

システム変数	説明
\$AN_LAI_AX_IS_LINKAX <sup>1)</sup>	状態: LAI 軸 == リンク軸
\$AN_LAI_AX_TO_IPO_NC_CHANAX[<ID>]	チャンネルとチャンネル軸番号、または NCU-ID とグローバル軸番号
\$AN_LAI_AX_TO_MACHAX[<ID>]	機械軸の NCU-ID と軸数
1) ビットマスク:ビット $n \triangleq$ LAI 軸( $n+1$ )、MD10002 \$MN_AXCONF_LOGIC_MACHAX_TAB ID:LAI 軸番号 NCU-ID: MD12510 \$MN_NCU_LINKNO の値	

参照先

システム変数の詳細については、以下を参照してください。

『リストマニュアル』、「システム変数」

下記も参照

軸コンテナシステム変数の使用 (ページ 145)

## 3.2.4.5 軸コンテナによる加工(概略図)

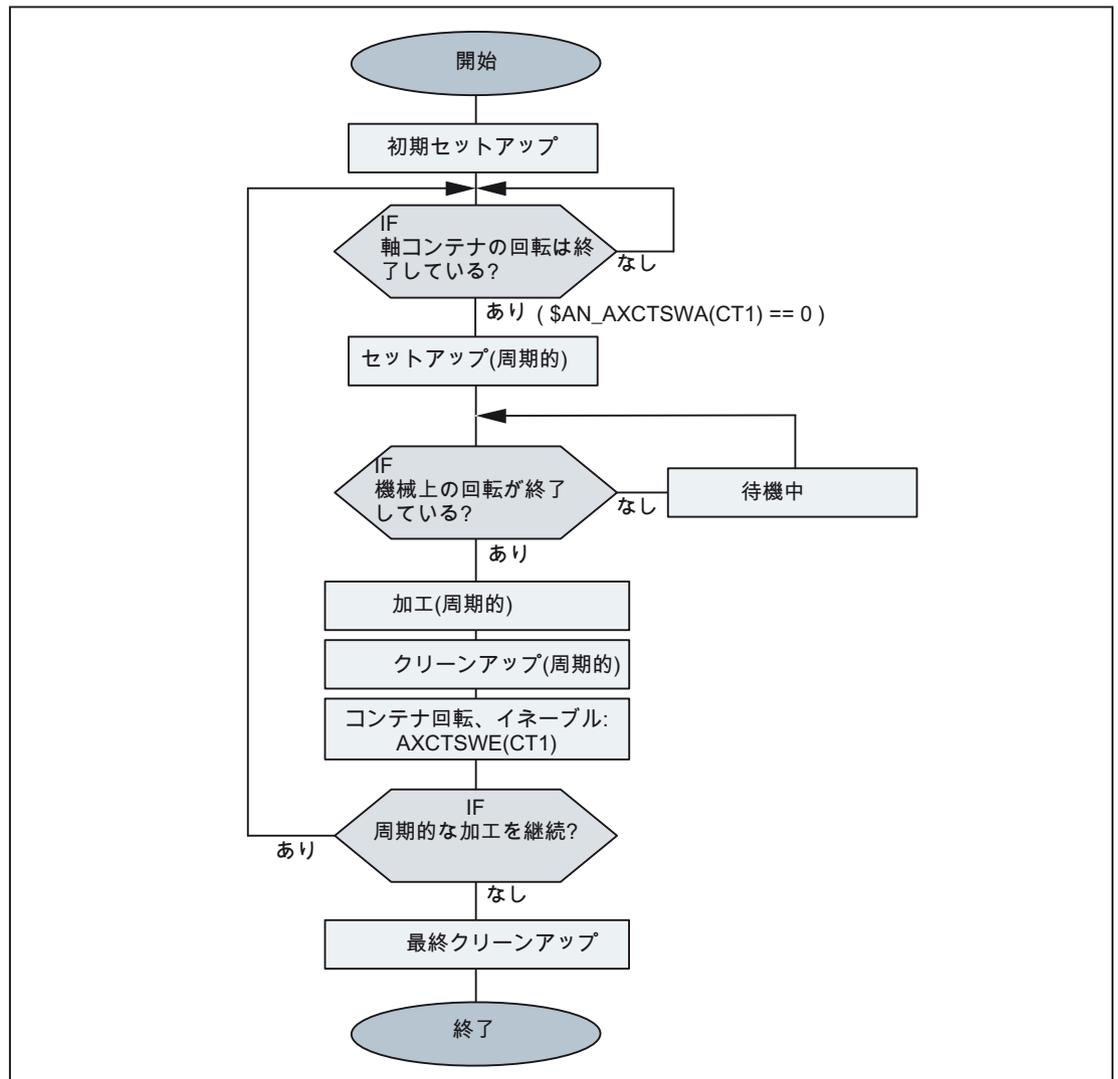


図 3-18 例: ロータリサイクルマシンのステーションの加工手順概略図

## 3.2.4.6 さまざまな動作状態の特性

## 起動(電源投入)

コントローラの起動では、スロットの割り当てに関して、マシンデータで定義された初期状態は、コントローラがオフされた軸コンテナの状態とは常に無関係とみなされます。

MD1270x \$MN\_AXCT\_AXCONF\_ASSIGN\_TABx

**通知**

**指令値と実際の状態の間での調整**

コントローラの起動後、軸コンテナの状態と機械の状態の差異を検出し、適切な軸コンテナの回転によりこれを補正することは、ユーザー/工作機械メーカーご自身の責任になります。

**モード変更**

自動モードでの軸コンテナの回転が有効であるコンテナ軸は、JOG モードに変更後は移動できません。

**チャンネル別リセット状態**

軸コンテナに関連するチャンネルがリセット状態になると直ちに、このチャンネルで軸コンテナ回転のイネーブルは不要になります。これで、残りの動作中のチャンネルを有効するためには十分です。

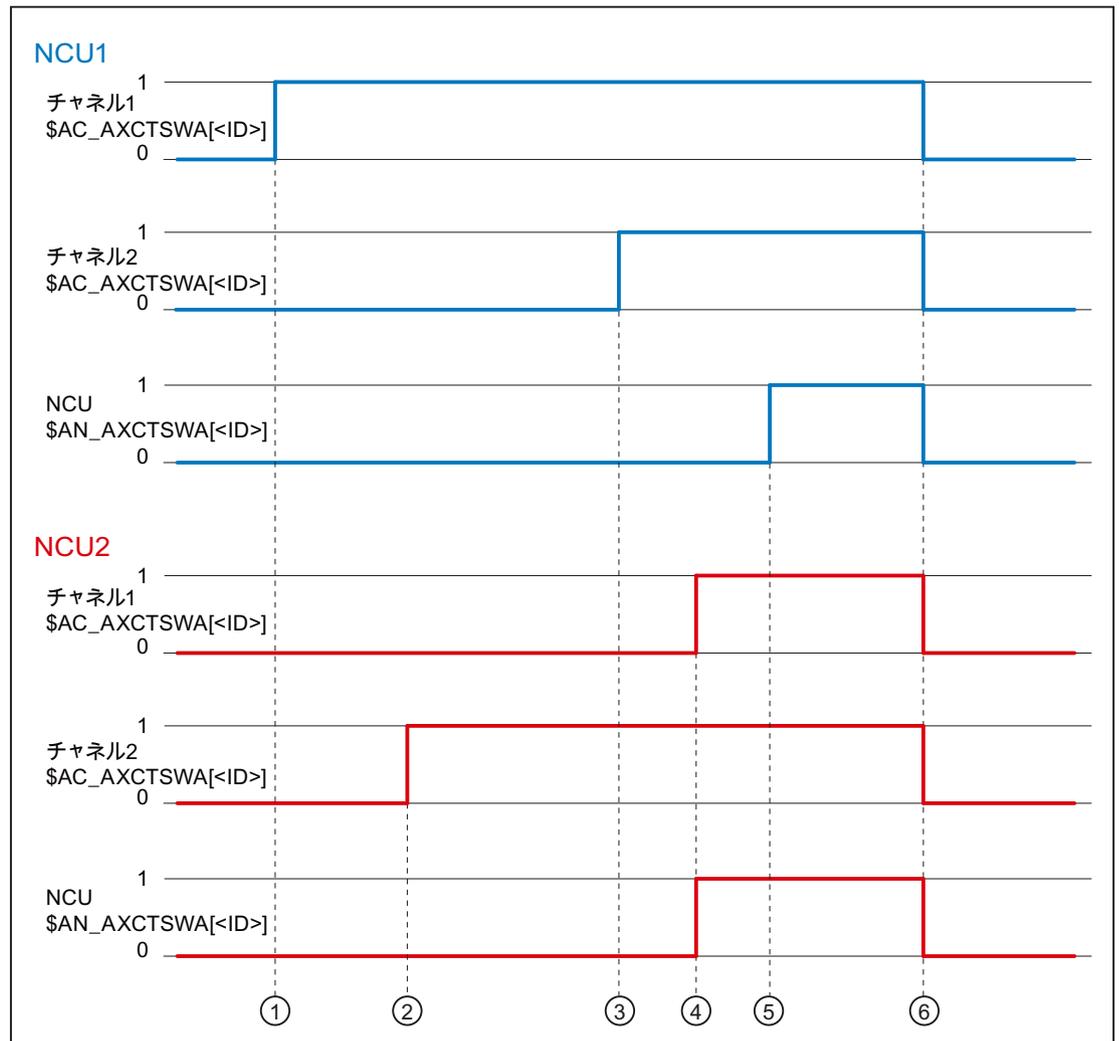
**ブロック検索**

軸コンテナ回転(AXCTSWE)は有効にできません。そして、1ブロックで有効になることはできません。命令の有効化と起動は、個別のアクションブロックでプログラム指令してください。つまり、他のチャンネルの状態に応じて、個別の各回転命令に応じて軸コンテナの状態が変化します。

**3.2.4.7 軸コンテナ回転のイネーブルを取り消す場合の特性**

AXCTSWE 命令は、軸コンテナのチャンネル別軸コンテナ回転を有効にします。AXCTSWE C 命令はイネーブルをキャンセルします。

次の図は、軸コンテナのシステム変数で表わされる軸コンテナ回転の手順の例を示しています。軸コンテナ上の 2 基の NCU の 2 つのチャンネルは常に関連します。



- ① NCU1、チャンネル 1: AXCTSWE 命令を使用して発行されたイネーブル
- ② NCU2、チャンネル 2: AXCTSWE 命令を使用して発行されたイネーブル
- ③ NCU1、チャンネル 2: AXCTSWE 命令によって発行されたイネーブル → 全チャンネルの全イネーブルが NCU1 にあります → NCU1 のすべてのイネーブル状態がリンク通信によって NCU2 に伝送されます
- ④ NCU2、チャンネル 1: AXCTSWE 命令によって発行されたイネーブル → 全チャンネルの全イネーブルが NCU2 にあります → NCU2 のすべてのイネーブル状態がリンク通信によって NCU1 に伝送されます  
全チャンネル(NCU2 と NCU1)の全イネーブル信号が現在、NCU2 にあります → 軸コンテナ回転が NCU2 でおこなわれます
- ⑤ NCU1: 全チャンネル(NCU1 と NCU2)の全イネーブル信号が現在、NCU1 にあります → 軸コンテナ回転が NCU1 でおこなわれます
- ⑥ NCU1/NCU2: 軸コンテナ回転が完了しました。

### 3.2 NCU リンク

図 3-19 クロス NCU イネーブルと軸コンテナ回転

以前に許可されたイネーブルをキャンセルするには、軸コンテナ上で関連するチャンネル (NCU1 または NCU2) の少なくとも 1 つに対するイネーブルがキャンセル時にまだ未処理であることが必要です。これは、タイミング ④ の前にキャンセルがおこなわれる必要があることを意味します。

全 NCU の全チャンネルからの全イネーブル信号が有効になると(タイミング ④)すぐに、取り消しはできなくなります。この場合、AXCTSWEC 命令は無効です。フィードバックはユーザーに送信されません。

#### 下記も参照

プログラミング (ページ 127)

#### 3.2.4.8 補足条件

##### 軸モード

軸モードで、または位置決め主軸(POSA、SPOSA)として、コンテナ軸が移動する場合、軸コンテナはプログラム指令された終了位置に到達した後でのみ回転します。

##### 主軸

- 主軸として有効なコンテナ軸は、軸コンテナの回転時に回転を続けます。
- 主軸のコントロールモード(速度/位置制御)は、関連する機械軸を参照します。軸コンテナの回転時には、設定コントロールモードでは機械軸を「移動」します。
- チャンネルのメイン主軸を参照する命令では、命令の実行時にチャンネルに対応する主軸番号を持つ機械軸が存在する必要があります。

MD35000 \$MA\_SPIND\_ASSIGN\_TO\_MACHAX[ <軸> ] == メイン主軸の番号

##### 注記

コンテナ軸としての主軸について、メイン主軸の軸コンテナ回転後に、チャンネル内に当該の機械軸がまだ存在することは、ユーザー/工作機械メーカーご自身の責任で確認してください。

## ゼロオフセット

---

### 注記

軸コンテナ回転後に、変更された機械軸の割り当てに対して、チャンネルの有効なゼロオフセットが調整されていることは、ユーザー/工作機械メーカーご自身の責任で確認してください。

---

## 連続軌跡モード

連続軌跡モードがチャンネル内で有効であり、軸コンテナの回転がおこなわれている場合、コンテナ軸の以降のプログラミングでは連続軌跡モードを中断します。この中断は、コンテナ軸が軌跡軸ではない場合でも発生します。

## PLC 軸

軸コンテナの回転が有効にされているコンテナ軸が PLC 軸になる場合、状態変更は軸コンテナ回転の完了後にのみ発生します。

## コマンド軸

軸コンテナの回転が有効にされているコンテナ軸がコマンド軸として移動する場合、軸コンテナ回転の完了後にのみ移動動作がおこなわれます。

## 揺動軸

軸コンテナの回転が有効にされているコンテナ軸が揺動軸になる場合、状態変更は軸コンテナ回転の完了後にのみ発生します。

## 外部ゼロオフセット

「外部ゼロオフセット」は、機械座標系(MCS)に基づきます。したがって、有効な「外部ゼロオフセット」について、いずれかのコンテナ軸によりアラーム 4022 で軸コンテナの回転が拒否されます。

### 3.2 NCU リンク

#### 軸のフレーム

コンテナ軸でもあるチャネル軸の軸のフレームは、軸コンテナの回転後に有効ではありません。軸コンテナ回転により、新しい機械軸がチャネル軸に割り当てられます。ただし、軸のフレームが機械軸を基準にしているため、その回転に応じて軸のフレームも変更されます。2つのフレームが一致しない場合、同期制御処理(内部 REORG)がおこなわれます。

---

#### 注記

軸コンテナの回転によって、チャネル軸と機械軸間の割り当てが変更されます。現在のフレームは、回転後も有効です。たとえば、基本フレームマスクのプログラミングによって回転後に正しいフレームが選択されていることをユーザご自身で確認してください。

---

#### 座標変換

コンテナ軸が主軸として座標変換に関与する場合、軸コンテナの回転を有効にする前に座標変換を解除してください。

#### 軸連結

コンテナ軸に対して軸連結が有効である場合、軸コンテナの回転を有効にする前に COUPOF による連結を解除してください。回転の完了後、COUPON により、すぐに連結を再選択できます。連結の新しい定義は必要ありません。

#### ガントリ軸

ガントリ軸をコンテナ軸にすることはできません。

#### 固定リミットへの移動

コンテナ軸がリミット停止の場合、軸コンテナ回転をおこなうことはできません。

#### ドライブのアラーム

コンテナ軸のドライブのアラームが未処置の場合、軸コンテナ回転をおこないません。

## 3.2.5 マスタリンク軸

### 3.2.5.1 概要

軸連結で、マスタ軸とスレーブ軸の機械軸が同じ NCU に接続されない場合、スレーブ軸の NCU のリンク軸を使用して連結を確立してください。この場合、リンク軸はマスタリンク軸として指定します。

マスタ軸の指令値は、NCU リンクによって補間周期内にマスタリンク軸に同期して伝送されます。同様に反対方向では、マスタリンク軸とスレーブ軸の現在値と状態データがマスタ軸に伝送されます。

マスタリンク軸は、スレーブ軸のローカルマスタ軸としてパラメータ設定します。

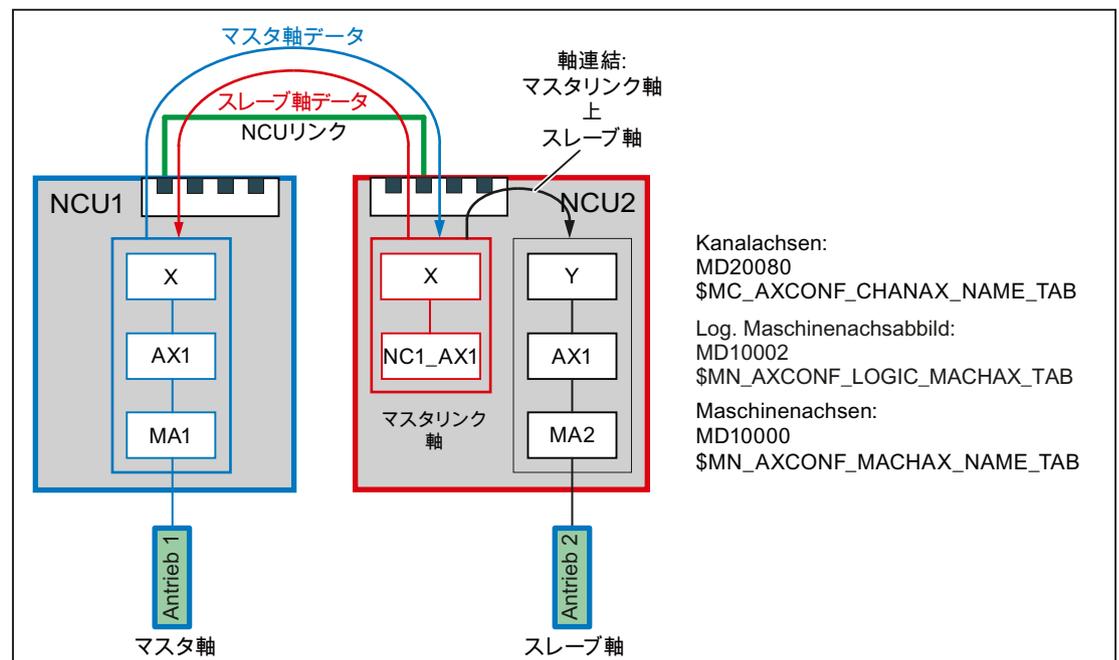


図 3-20 マスタリンク軸

#### 連結軸

マスタリンク軸は、以下の軸連結と組み合わせて使用できます。

- 軸間連動機能
- 連結動作
- 法線方向制御
- 電子ギア(ELG)
- 主軸同期

3.2 NCU リンク

必要条件

NCU は、NCU リンク経由で通信することが必要です。「リンク通信 (ページ 92)」の章を参照してください。

3.2.5.2 パラメータ設定

リンク通信

NC 別マシンデータ

番号	識別子 \$MN_	意味
MD12510	NCU_LINKNO	リンクグループ内の NCU の固有の数値識別子。識別子は、1 から昇順に連番で抜けのないように割り当ててください。 数値の範囲:1, 2, ... 最大 NCU 番号 <b>注:</b> NCU 識別子として値 <b>1</b> が割り当てられる NCU は、リンクグループの <b>マスタ NCU</b> です。リンク軸と軸コンテナのパラメータ設定は、マスタ NCU のマシンデータとセッティングデータでのみ可能です。
MD18780	MM_NCU_LINK_MASK.Bit 0	リンク通信の起動
MD18781	NCU_LINK_CONNECTIONS	内部リンク接続数 <b>注:</b> 初期値 <b>0</b> を維持することを推奨します(この数値は NC によって決定されます)。
MD18782	MM_LINK_NUM_OF_MODULES	NCU リンク経由で相互接続される NCU の数。

## 指令値同期

## NC 別マシンデータ

番号	識別子 \$MN_	意味
MD18720	MM_SERVO_FIFO_SIZE	<p>IPO/SERVO データバッファのサイズ</p> <p>マスタ軸の指令値を NCU リンクによりマスタリンク軸の NCU に伝送する場合、2 補間周期のむだ時間を生じます。マスタ軸の NCU とマスタリンク軸の NCU 上の IPO/SERVO データバッファサイズをパラメータ設定することにより、このむだ時間を補正します。</p> <ul style="list-style-type: none"> <li>● マスタ軸の NCU: 4</li> <li>● マスタリンク軸の NCU: 2</li> </ul>

## 通知

マスタリンク軸とクロス NCU 軸コンテナを同時に使用する場合、軸コンテナではマシンデータ MD18720 \$MN\_MM\_SERVO\_FIFO\_SIZE = 3 を設定してください。つまり、マスタ軸とスレーブ軸の指令値の同期出力は可能ではありません。この場合、オフセットのサイズは 1 補間周期です。

## マスタ軸、マスタリンク軸、およびスレーブ軸

## NC 別マシンデータ

番号	識別子 \$MN_	意味
MD10000	AXCONF_MACHAX_NAME_TA B	機械軸名称
MD10002	AXCONF_LOGIC_MACHAX_T AB	論理機械軸イメージ

## チャンネル別マシンデータ

番号	識別子 \$MC_	意味
MD20070	AXCONF_MACHAX_USED	使用機械軸

## 3.2 NCU リンク

## 軸マシンデータ

番号	識別子\$MA_	意味
MD30554	AXCONF_ASSIGN_MASTER_NCU	マスタ NCU 複数の NCU に対して機械軸が移動する場合は、1 基の NCU をマスタ NCU として定義してください。この NCU のコントローラの起動後に、この指令値が生成されます。

## 3.2.5.3 マスタ値を入力するためのシステム変数

マスタ値は、次のシステム変数を使用してマスタ軸の NCU で指定できます。

- 位置マスタ値: \$AA\_LEAD\_SP[<マスタ軸>]
- 速度マスタ値: \$AA\_LEAD\_SV[<マスタ軸>]

値を変更した場合、その値は NCU リンク毎にスレーブ軸の NCU にも伝送されます。

## 注記

これらのシステム変数は、リンク変数よりも低い伝送優先順位を持ちます。

## 3.2.5.4 補足条件

以下の補足条件を遵守してください。

- マスタ軸をリンク軸にすることはできません。
- マスタ軸をコンテナ軸にすることはできません。
- マスタ軸をガントリ軸にすることはできません。
- マスタ軸は、自身の NCU 内でのみ入れ換えることができます(「K10:チャンネル間の軸入れ替え (ページ 455)」の章を参照してください)。
- マスタリンク軸との連結は、カスケードにすることはできません。
- マスタリンク軸は、マスタ軸と独立して移動することはできません。

## 注記

## 「マスタリンク軸」機能と「リンク軸」機能

「マスタリンク軸」機能と「リンク軸」機能では、マシンデータ MD18720 \$MN\_MM\_SERVO\_FIFO\_SIZE 内で異なる設定が必要であるため、リンクグループ内で同時に使用することはできません。

### 3.2.5.5 例

マスタリンク軸を使用した軸連結のパラメータ設定とプログラミングの詳細例は: 「例」> 「マスタリンク軸 (ページ 158)」の章を参照してください。

### 3.2.6 リンクグループ内の単位系

クロス NCU 補間では、リンクグループのすべての NCU で同じ単位系を有効にしてください。

#### HMI による一般的な単位系切替

リンクグループの 1 つの NCU の HMI 操作画面からだけでなく、リンクグループの他のどの NCU でも単位系切替をおこなえるように、リンクグループのすべての NCU で次の条件を満たしてください。

- MD10260 \$MN\_CONVERT\_SCALING\_SYSTEM=1
- すべてのチャンネルに対して:  
MD20110 \$MC\_RESET\_MODE\_MASK、ビット 0 = 1
- すべてのチャンネルがリセット状態
- JOG モードか DRF モードで、あるいは PLC 経由で移動している軸がない。
- 「砥石周速度一定(制御) (GWPS)」機能が有効でない。

リンクグループの 1 つの NCU で指定した条件のいずれかが満たされていない場合、単位系切替はリンクグループのどの NCU でもおこなわれません。

#### 異なる単位系

異なる単位系は、クロス NCU 補間がおこなわれていない限り、有効なリンクグループに関係なく使用できます。パートプログラムまたはシンクロナイズドアクション内の特定の NCU に対して、G 命令(G70、G71、G700、G710)を使用して、単位系の設定をおこないます。

#### 参照先

『機能マニュアル、基本機能』; 「速度、指令値/フィードバック回路、閉ループ制御(G2)」

## 3.3 例

## 3.3 例

## 3.3.1 リンク軸

それぞれがリンク軸を備える 2 基の NCU のパラメータ例

## NCU1

マシンデータ	注
一般リンクデータ:	
\$MN_NCU_LINKNO = 1	マスタ NCU
\$MN_MM_NCU_LINK_MASK = 1	NCU リンクを有効に設定
\$MN_MM_SERVO_FIFO_SIZE = 3	補間と位置制御間のデータバッファのサイズ
\$MN_MM_LINK_NUM_OF_MODULES = 2	リンクモジュールの数
論理機械軸イメージ(LAI):	
\$MN_AXCONF_LOGIC_MACHAX_TAB[0] = 「AX1」	ローカル機械軸
\$MN_AXCONF_LOGIC_MACHAX_TAB[1] = 「AX2」	ローカル機械軸
\$MN_AXCONF_LOGIC_MACHAX_TAB[2] = 「NC2_AX3」	リンク軸
機械軸名称、NCU 識別子としてシステム全体で固有:	
\$MN_AXCONF_MACHAX_NAME_TAB[0] = 「NC1_A1」	
\$MN_AXCONF_MACHAX_NAME_TAB[1] = 「NC1_A2」	
\$MN_AXCONF_MACHAX_NAME_TAB[2] = 「NC1_A3」	
機械軸へのチャネル軸の割り当て:	
\$MC_AXCONF_MACHAX_USED[0]=1	1. LAI[0]の機械軸へのチャネル軸の割り当て
\$MC_AXCONF_MACHAX_USED[1]=2	2. LAI[1]の機械軸へのチャネル軸の割り当て
\$MC_AXCONF_MACHAX_USED[2]=3	3. LAI[2]の機械軸へのチャネル軸の割り当て

## NCU2

マシンデータ	注
一般リンクデータ:	
\$MN_NCU_LINKNO = 2	スレーブ NCU
\$MN_MM_NCU_LINK_MASK = 1	NCU リンクを有効に設定
\$MN_MM_SERVO_FIFO_SIZE = 3	補間と位置制御間のデータバッファのサイズ
\$MN_MM_LINK_NUM_OF_MODULES = 2	リンクモジュールの数
論理機械軸イメージ(LAI):	
\$MN_AXCONF_LOGIC_MACHAX_TAB[0] = 「AX1」	ローカル機械軸
\$MN_AXCONF_LOGIC_MACHAX_TAB[1] = 「AX2」	ローカル機械軸
\$MN_AXCONF_LOGIC_MACHAX_TAB[2] = 「NC1_AX3」	リンク軸
機械軸名称、NCU 識別子としてシステム全体で固有:	
\$MN_AXCONF_MACHAX_NAME_TAB[0] = 「NC2_A1」	
\$MN_AXCONF_MACHAX_NAME_TAB[1] = 「NC2_A2」	
\$MN_AXCONF_MACHAX_NAME_TAB[2] = 「NC2_A3」	
機械軸へのチャネル軸の割り当て:	
\$MC_AXCONF_MACHAX_USED[0]=1	1. LAI[0]の機械軸へのチャネル軸の割り当て
\$MC_AXCONF_MACHAX_USED[1]=2	2. LAI[1]の機械軸へのチャネル軸の割り当て
\$MC_AXCONF_MACHAX_USED[2]=3	3. LAI[2]の機械軸へのチャネル軸の割り当て

## 3.3.2 軸コンテナの協調

次の表で、上から下方向へ時間に応じた特性を示しています。データは、2つのチャネルのみがコンテナ軸を持つ条件として有効です。

3.3 例

3.3.2.1 パートプログラムの待機なしの軸コンテナの回転

チャンネル 1	チャンネル 2	コメント
AXCTWE(C1)	パートプログラム...	チャンネル 1 で、軸コンテナの回転を有効にします。
コンテナ軸の移動なしのパートプログラム	パートプログラム...	
	AXCTSWE(C1)	チャンネル 2 により、軸コンテナの回転が可能になります。両方のチャンネルで回転が有効であるため、コンテナが回転します。
コンテナ軸の移動ありのパートプログラム	コンテナ軸の移動ありのパートプログラム	待機なし

3.3.2.2 パートプログラムが自動的に待機する軸コンテナの回転

チャンネル 1	チャンネル 2	コメント
AXCTWE(C1)	パートプログラム...	チャンネル 1 で、軸コンテナの回転を有効にします。
コンテナ軸の移動ありのパートプログラム	パートプログラム...	チャンネル 1 は、軸コンテナの回転を自動的に待機します。
	AXCTSWE(C1)	チャンネル 2 により、軸コンテナが回転可能になります。回転が実行されます。 チャンネル 1 は継続します。

3.3.2.3 1チャンネルのみによる軸コンテナの回転(例: 電源投入時)

チャンネル 1	チャンネル 2	コメント
AXCTWED(C1)	RESET ステータス	瞬間的な回転

### 3.3.3 軸コンテナシステム変数の使用

#### 3.3.3.1 条件付き分岐

チャンネル 1	コメント
N100 AXCTWE(CT1)	チャンネル 1: 軸コンテナ CT1 の回転の有効化
MARKER1:	
N200 ...	コンテナ軸の移動なしのパートプログラム
IF \$AC_AXCTSWA[CT1] == 1 GOTOB MARKE1	軸コンテナ CT1 の回転がまだ有効な場合 MARKER1 により続行します そうでない場合(軸コンテナ CT1 の回転を完了します)
N300 ...	コンテナ軸の移動ありのパートプログラム

#### 3.3.3.2 \$AN\_AXCTSWA による静的なシンクロナイズドアクション

チャンネル 1	コメント
IDS =1 EVERY \$AN_AXCTSWA[CT1] == 1 DO M99	静的なシンクロナイズドアクション: 常に軸コンテナ回転の開始時に、補助機能 M99 を 出力します。
	参照先: 『シンクロナイズドアクション』機能マニ ュアル

3.3 例

3.3.3.3 軸コンテナ回転の個別の完了を待機します。

注記

プログラミングシステム変数\$AN\_AXCTSWA によって自動の先読み停止が開始されますが、以降の指令シーケンスを使用して、要求されたチャンネルでの軸コンテナ回転の完了を一義的に識別することはできません。

```
WHILE ($AN_AXCTSWA [<ID>] == TRUE) ; wait:軸コンテナ回転の有効化がキャンセルされるまで
```

```
ENDWHILE
```

指令シーケンスのチャンネルが、別のチャンネルから開始された軸コンテナ回転の完了を待機する間に、バックグラウンドで再編成がおこなわれる場合があります。たとえば、要求された軸を別のチャンネルに伝送することで再編成がおこなわれます。その結果、軸コンテナ回転が完了していない場合でも、システム変数によって値 FALSE が供給されません。

推奨される指令シーケンス:

```
RL = $AN_AXCTAS [<ID>] ; 読み取り:実際の軸コンテナ位置
```

```
AXCTSWE (<ID>) ; 有効化:軸コンテナ回転
```

```
WHILE (RL == $AN_AXCTAS [<ID>]) ; 待機:読み取りが実際の軸コンテナと等しくなくなるまで
```

```
; 位置
```

```
ENDWHILE
```

RL は、ランダムなユーザー変数です。

例 1:システム変数\$AN\_AXCTAS を使用した明示的な待機

プログラムコード	コメント
RL = \$AN_AXCTAS [<ID>]	; 読み取り:実際の軸コンテナ位置
AXCTSWE (<ID>)	; 有効化:軸コンテナ回転
WHILE (RL == \$AN_AXCTAS [<ID>])	; 待機:軸コンテナの変更
ENDWHILE	; 位置

例 2:WAIT マークを使用した明示的な待機

プログラムコード	コメント
CLEARM (9)	; 削除チャンネルのマーク (タグ) 9
AXCTSWE (<ID>)	; 有効化:軸コンテナ回転
; SynAct:軸コンテナ回転が完了するまで待機⇒設定:チャンネルのマーク (タグ) 9	
WHEN \$AN_AXCTSWA [<ID>] == TRUE DO SETM (9)	
WAITMC (9, 1)	; 待機:チャンネル 1 のマーク (タグ)

**例 3:コンテナ主軸を使用した自動待機**

プログラムコード	コメント
; コンテナ主軸を (再度) プログラム指令	
; 内部的に、コンテナ主軸の開始前にシステムが	
; 軸コンテナ回転の終了を待機	
M3 S100	; チャネルのメイン主軸

**例 4:コンテナ軸を使用した自動待機**

プログラムコード	コメント
; 軌跡を移動せずにコンテナ軸をインクリメンタルにプログラム指令	
; 内部的に、コンテナ軸の開始前にシステムが	
; 軸コンテナ回転の終了を待機	
X=IC (0)	; チャネル軸 X

**3.3.4 複数主軸旋盤の構成****はじめに**

下記では、次の使用例について説明します。

- NCU リンクグループの複数の NCU
- 軸コンテナのフレキシブル構成

### 3.3 例

#### 機械仕様

- 機械のドラム A の外周(正面機械加工)に沿って配置:
  - 4つの主軸、HS1 ~ HS4  
各主軸では、材料送り(バー、油圧バーフィード、軸:STN1-STN4)が可能です。
  - 4クロススライド
  - 各スライドは2基の軸を持ちます。
  - オプションで、電動工具 S1-S4 が各スライド上で動作可能です。
- 機械のドラム B の外周(背面機械加工)に沿って配置:
  - 4基の対向主軸 GS1 ~ GS4
  - 4クロススライド
  - 各スライドは2基の軸を持ちます。
  - オプションで、電動工具 S5-S8 が各スライド上で動作可能です。
  - 各対向主軸の位置は、たとえばドラム B の背面加工用の主軸から部品を移動するために、直線軸によって移動できます。(移送軸。軸:ZG1-ZG4)。
- カップリング:
  - ドラム A が回転している場合、このドラムのすべての主軸が異なるスライドのグループに属します。
  - ドラム B が回転している場合、このドラムのすべての対向主軸およびすべての移送軸は、異なるスライドのグループに属します。
  - ドラム A と B の回転は独立しています。
  - ドラム A と B の回転は、270度までに制限されています。  
(電源ケーブル仕様およびねじれのため)

#### 用語:位置

主軸 HS<sub>i</sub> と対向主軸 GS<sub>i</sub> 各スライド上の位置を表します。

#### NCU の割り付け

軸と主軸の(この例の)位置はそれぞれ、NCU に割り当てられます。こうした NCU の 1 つであるマスタ NCU はさらに、ドラム A と B の回転のために軸を制御します。その結果、4つの NCU の従動軸の最大数は以下ようになります。

#### 軸数

NCU<sub>i</sub> 毎に、下記の軸/主軸を設定してください。

スライド 1:X<sub>i</sub>1, Z<sub>i</sub>1

2:X<sub>2</sub>, Z<sub>2</sub>

主軸:HS<sub>i</sub>、GS<sub>i</sub>、電動工具:S1、S2

移送軸:ZG<sub>i</sub>

バーフィード:STN<sub>i</sub>。

マスタ NCU では、上記の軸にドラム A および B を回転するための 2 つの軸を追加します。リストは、NCU で合計 4 つの位置に軸番号を設定することができないことを示します。(制限 31 軸、4 + 10 + 2 軸が必要)。

### 軸コンテナ

ドラム A/B の回転で、HS<sub>i</sub>、GS<sub>i</sub>、ZG<sub>i</sub>、および STN<sub>i</sub> は別の NCU に割り当てる必要があるため、軸コンテナでリンク軸として設定してください。

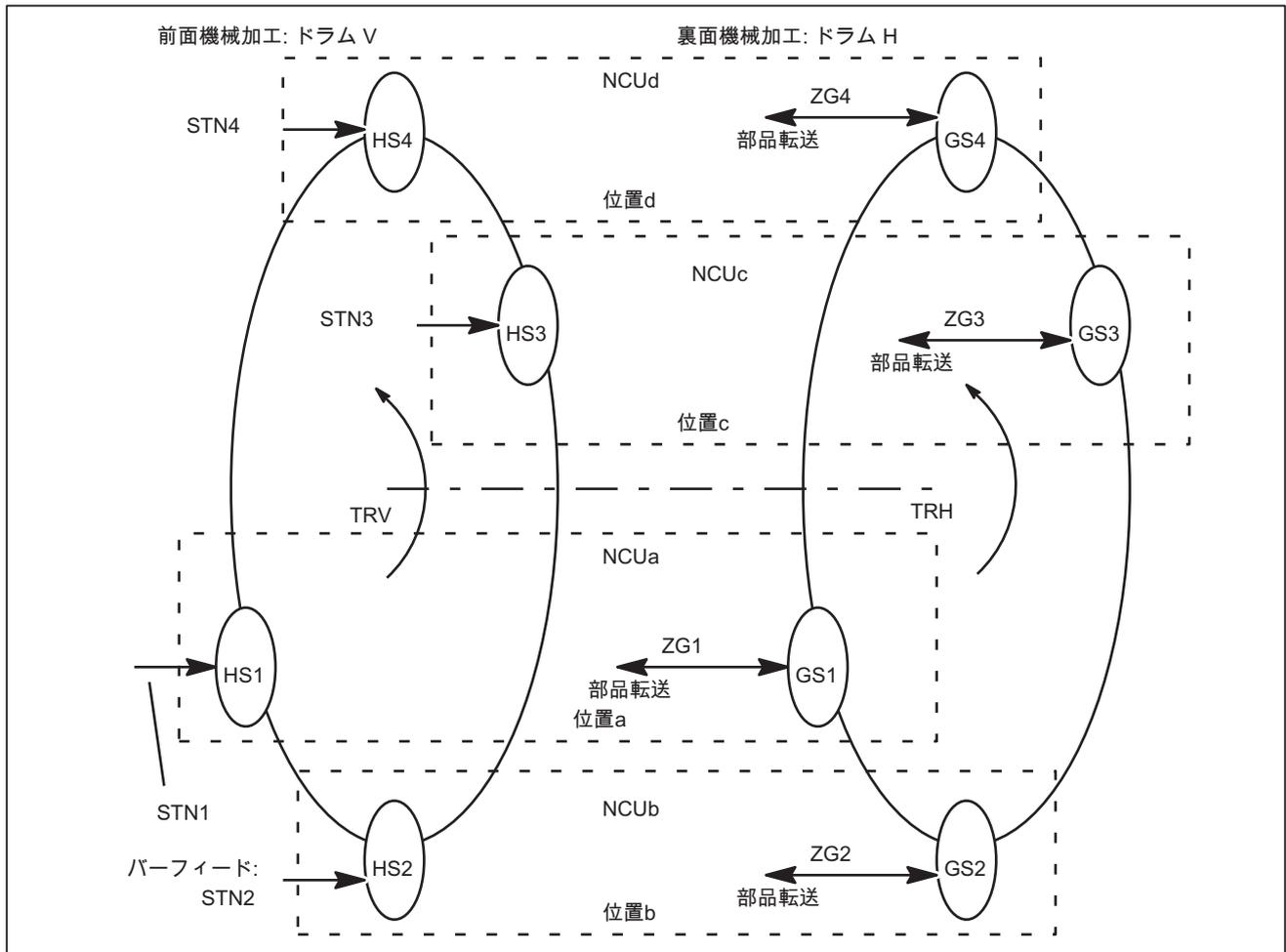


図 3-21 主軸 HS<sub>i</sub> の概略図、対向主軸 GS<sub>i</sub>、バーフィード軸 STN<sub>i</sub>、および移送軸 ZG<sub>i</sub>

3.3 例

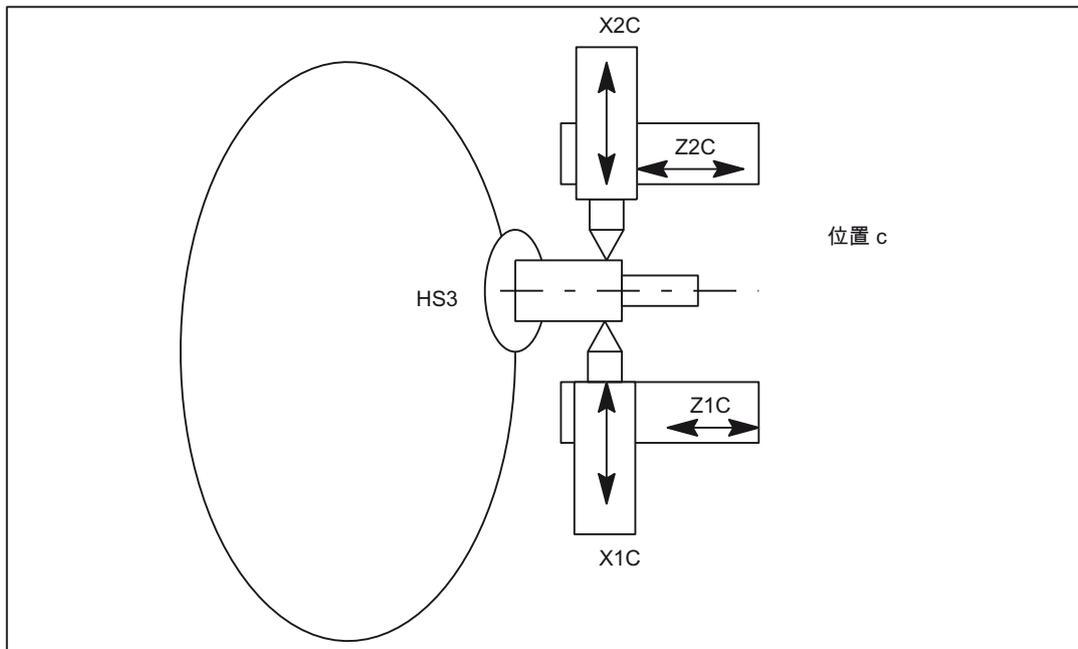


図 3-22 また、位置毎に 2 つのスライドが 1 基の主軸上で一緒に動作できます。

注記

軸には、スライドおよび位置への軸の割り付けを明確にするための次の名称が与えられています。

i スライド(1 と 2)、j 位置(A-D)の  $X_{ij}$

i スライド(1 と 2)、j 位置(A-D)の  $Z_{ij}$

位置およびそのスライドが 1 箇所にとどまるのに対し、主軸、対向主軸、バーフィード軸 STN、および移送軸 ZG はドラム V または H の回転の結果、新しい位置に移動します。

たとえば、スライドが考慮される時に NC 毎に管理される軸は、上記の図に示された構成については次のようになります。

## マスタ NCU の軸

表 3-1 マスタ NCU の軸:NCUa

共有軸	ローカル軸	備考
	TRV (ドラム V)	マスタ NCU のみ
	TRH (ドラム H)	マスタ NCU のみ
	X1A	スライド 1
	Z1A	スライド 1
	X2A	スライド 2
	Z2A	スライド 2
	S1	スライド 1
	S2	スライド 2
HS1		軸コンテナが必要
GS1		軸コンテナが必要
ZG1		軸コンテナが必要
STN1		軸コンテナが必要
4	8	

## NCUb から NCUd への軸

マスタ NCU 以外の NCU は、同じ軸を持ちます。ただし、ドラム TRV および TRH 用ドライブの軸の場合は例外です。位置を指定する文字は、NCU および軸の名称に応じて置換えてください(a、A → b、B ~ d、D)。

## コンフィギュレーションルール

下記の構成について次の規則が適用されます。

- 上記の図「主軸...」で示されているように、動作中にドラム回転によってそれぞれの NCU に割り当てられる主軸、対向主軸、および軸は、軸コンテナで構成してください。  
(HS<sub>i</sub>、GS<sub>i</sub>、ZG<sub>i</sub>、STN<sub>i</sub>)。
- ドラム A のすべての主軸が同じコンテナ(番号 1)に存在すること。
- ドラム A のすべてのバーフィード軸が同じコンテナ(番号 2)に存在すること。
- ドラム B のすべての対向主軸が同じコンテナ(番号 3)に存在すること。

### 3.3 例

- ドラム B のすべての移送軸が同じコンテナ(番号 4)に存在すること。
- 主軸  $HS_i$  およびその対向主軸  $GS_i$  に加えて、対向主軸  $ZG_i$  の移送軸および主軸のバーフィード軸  $STN_i$  は、NCU の均一な負荷分散の理由のために次のように割り当てること。  
NCUa HS1 ~ STN1,  
NCUb HS2 ~ STN2、... など。
- スライド軸  $X_{ij}$ 、 $Z_{ij}$  は、固定の NCU の割り付けを持つ単独のローカル軸であること。
- それぞれのケースで、スライドが NCU の専用チャンネルに割り当てられていること。  
したがって、スライドは独立に移動させることができます。

### 設定オプション

- 主軸または対向主軸は、スライドに適応して割り当ててください。
- 主軸および対向主軸の速度は、各位置で個別に定義できます。  
例外:  
ドラム V の前面加工からドラム H の背面加工に部品が移動する間、主軸および対向主軸の速度は同期する必要があります(主軸同期の連結)。  
スライド 1 を「サポート」するためにスライド 2 も前面加工に関与する場合、スライド 2 にも主軸速度が適用されます。同様に、スライド 1 が背面加工に関与する場合、スライド 1 にも対向主軸速度が適用されます。

### 少ない速度変化

フィードバック値の処理で発生する遅延は避けられないため、クロス NCU の加工動作中の急な速度変化は避けてください。軸データと信号を参照してください。

### NCU1 の構成

パートプログラムの同一チャンネル軸名称の使用:

S4:主軸

S3:対向主軸

X1:切り込み軸

Z1:長手軸

S1:回転工具

Z3:移送軸

TRV:主軸のドラム V

TRH:対向主軸のドラム H

STN:油圧バーフィード

太字で強調表示された軸は、軸入れ替えに関しては、現在のチャンネルがその軸のホームチャンネルであることを表します。

表 3-2 NCUa、位置:a、チャンネル:1、スライド:1

チャンネル軸の名称	..._MACH AX_USED	\$MN_ AXCONF_LOGIC_MA CHAX_TAB	コンテナ、スロット 入力値 (文字列)	機械軸名称
<b>S4</b>	1	AX1:CT1_SL1	1 1 NC1_AX1	HS1
S3	2	AX2:CT3_SL1	3 1 NC1_AX2	GS1
<b>X1</b>	3	AX3:		X1A
<b>Z1</b>	4	AX4:		Z1A
Z3	5	AX5:CT4_SL1	4 1 NC1_AX5	ZG1
<i>S1</i>	<i>6</i>	<i>AX6:</i>		<i>WZ1A</i>
<b>STN</b>	7	AX7:CT2_SL1	2 1 NC1_AX7	STN1
<b>TRV</b>	11	AX11:		TRV
<b>TRH</b>	12	AX12:		TRH
x2 *				
z2 *				

## 3.3 例

表 3-3 NCUa、位置:a、チャンネル:2、スライド:2

チャンネル軸の名称	..._MACH AX_USED	\$MN_ AXCONF_LOGIC_MA CHAX_TAB	コンテナ、スロット 入力値 (文字列)	機械軸名称
S4	1	AX1:CT1_SL1	1 1 NC1_AX1	HS1
S3	2	AX2:CT3_SL1	3 1 NC1_AX2	GS1
Z3	5	AX5:CT4_SL1	4 1 NC1_AX5	ZG1
STN	7	AX7:CT2_SL1	2 1 NC1_AX7	STN1
X2	8	AX8:		X2A
Z2	9	AX9:		Z2A
S1	10	AX10:		WZ2A
x1 *				
z1 *				

## 注記

\* 軸位置および 1 つの位置での 4 軸加工によるプログラム協調のために、軸コンテナの位置の入力では、書式: 「NC1\_AX..」が必要で、これは NC1 = NCU 1 を意味します。上記のテーブルで、NCUa は NC1\_..., NCUb、NC2\_...などにマッピングされます。

## その他の NCU

上記のリストの構成データは、NCUb ~ NCUd に合わせて指定してください。以下の条件を守る必要があります。

- 軸 TRA および TRB は、NCUa、チャンネル 1 のみで使用できます。
- 構成データは個々の軸について指定するため、コンテナ番号は他の NCU でも保持されます。

- スロット番号は、下記のとおりです。  
NCUb → 2  
NCUc → 3  
NCUd → 4
- 機械軸の名称は次のとおりです。  
NCUb → HS2、GS2、ZG2、STN2  
NCUc → HS3、GS3、ZG3、STN3  
NCUd → HS4、GS4、ZG4、STN4。

### 軸コンテナ

表 7-17 のコンテナに関連する情報と、同様に構成された NCU、NCUb ~ NCUd のコンテナ開始点をコンテナおよびスロットの順で次の表に示します。これらは、マシンデータで設定してください。

MD12701 \$MN\_AXCT\_AXCONF\_ASSIGN\_TAB1[スロット]

...

MD12716 \$MN\_AXCT\_AXCONF\_ASSIGN\_TAB16[スロット]

ここで、スロット:1 ~ 4 は、複数主軸旋盤の 4 つの位置に対して設定してください。

---

#### 注記

マシンデータ入力値について

\$MN\_AXC\_AXCONF\_ASSIGN\_TAB<sub>i</sub>[スロット]

上記の表の初期位置での値を入力してください(小数点および機械軸の名称は含めません)。

---

3.3 例

表 3-4 ドラム A の軸コンテナおよびその位置に依存する内容

コンテナ	スロット	初期の状態 (TRA 0°)	スイッチ 1 (TRA 90°)	スイッチ 2 (TRA 180°)	スイッチ 3 (TRA 270°)	スイッチ 4 = (TRA 0°)
1	1	NC1_AX1、 HS1	NC2_AX1、 HS2	NC3_AX1、 HS3	NC4_AX1、 HS4	NC1_AX1、 HS1
	2	NC2_AX1、 HS2	NC3_AX1、 HS3	NC4_AX1、 HS4	NC1_AX1、 HS1	NC2_AX1、 HS2
	3	NC3_AX1、 HS3	NC4_AX1、 HS4	NC1_AX1、 HS1	NC2_AX1、 HS2	NC3_AX1、 HS3
	4	NC4_AX1、 HS4	NC1_AX1、 HS1	NC2_AX1、 HS2	NC3_AX1、 HS3	NC4_AX1、 HS4
2	1	NC1_AX7、 STN1	NC2_AX7、 STN2	NC3_AX7、 STN3	NC4_AX7 STN4	NC1_AX7、 STN1
	2	NC2_AX7、 STN2	NC3_AX7、 STN3	NC4_AX7、 STN4	NC1_AX7、 STN1	NC2_AX7、 STN2
	3	NC3_AX7、 STN3	NC4_AX7、 STN4	NC1_AX7、 STN1	NC2_AX7、 STN2	NC3_AX7、 STN3
	4	NC4_AX7、 STN4	NC1_AX7、 STN1	NC2_AX7、 STN2	NC3_AX7、 STN3	NC4_AX7、 STN4
ドラムの移動		0°	+ 90°	+ 90°	+ 90°	- 270°

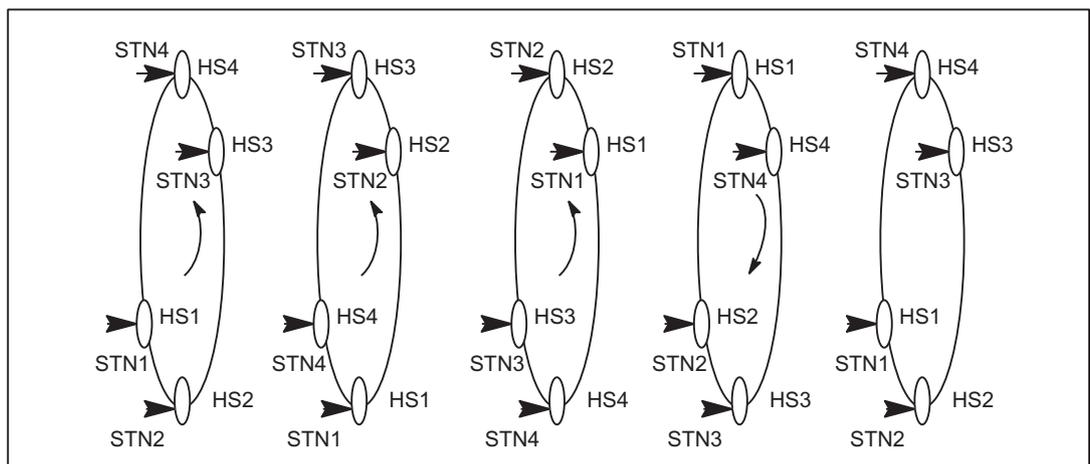


図 3-23 ドラム A の位置

表 3-5 ドラム B の軸コンテナおよびその位置に依存する内容

コンテナ	スロット	初期の状態 (TRB 0°)	スイッチ 1 (TRB 90°)	スイッチ 2 (TRB 180°)	スイッチ 3 (TRB 270°)	スイッチ 4 = (TRB 0°)
3	1	NC1_AX2、 GS1	NC2_AX2、 GS2	NC3_AX2、 GS3	NC4_AX2、 GS4	NC1_AX2、 GS1
	2	NC2_AX2、 GS2	NC3_AX2、 GS3	NC4_AX2、 GS4	NC1_AX2、 GS1	NC2_AX2、 GS2
	3	NC3_AX2、 GS3	NC4_AX2、 GS4	NC1_AX2、 GS1	NC2_AX2、 GS2	NC3_AX2、 GS3
	4	NC4_AX2、 GS4	NC1_AX2、 GS1	NC2_AX2、 GS2	NC3_AX2、 GS3	NC4_AX2、 GS4
4	1	NC1_AX5、 ZG1	NC2_AX5、 ZG2	NC3_AX5、 ZG3	NC4_AX5 ZG4	NC1_AX5、 ZG1
	2	NC2_AX5、 ZG2	NC3_AX5、 ZG3	NC4_AX5、 ZG4	NC1_AX5、 ZG1	NC2_AX5、 ZG2
	3	NC3_AX5、 ZG3	NC4_AX5、 ZG4	NC1_AX5、 ZG1	NC2_AX5、 ZG2	NC3_AX5、 ZG3
	4	NC4_AX5、 ZG4	NC1_AX5、 ZG1	NC2_AX5、 ZG2	NC3_AX5、 ZG3	NC4_AX5、 ZG4

3.3 例

3.3.5 マスタリンク軸

3.3.5.1 構成

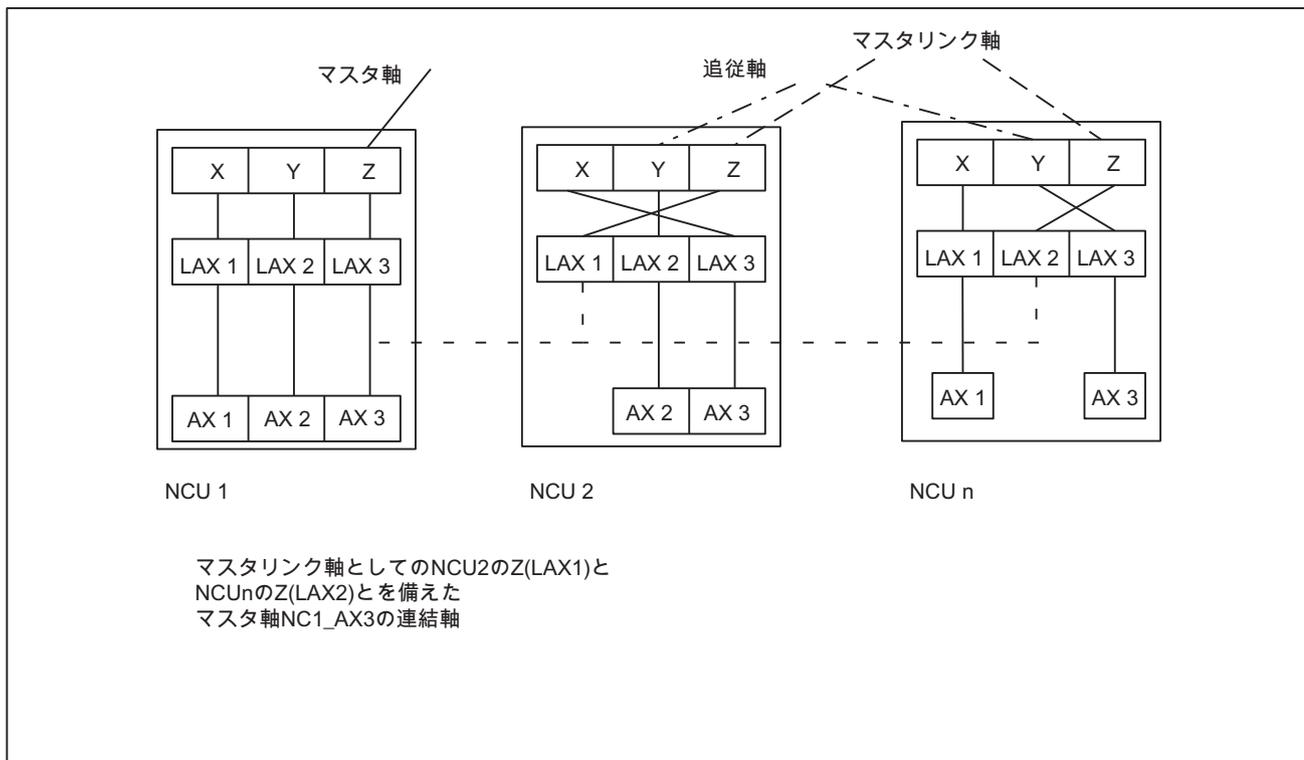


図 3-24 NCU2 ~ NCUn は、NCU1 (NCU1-AX3) 上の機械軸への連結を有効にするためにマスタリンク軸を使用します。

次の例では、NCU2 上のスレーブ軸としての Y 軸(LAX2、AX2)とマスタリンク軸としての Z 軸(LAX3、NC1\_AX3)の間の軸連結部を示しています。

マシンデータ

- マスタ値軸のマシンデータは、ホーム NCU にのみロードできます。関連するマシンデータは、この NCU からマスタリンク軸が定義されている他の NCU に配信されません。
- 各マスタリンク軸について、この NCU 上で移動可能な軸の最大数が 1 軸ずつ減少します。

## NCU1 のマシンデータ(マスタ軸)

マシンデータ	意味
\$MN_NCU_LINKNO = 1	1. 番目またはマスタ NCU
\$MN_MM_NCU_LINK_MASK = 1	NCU リンク有効
\$MN_MM_LINK_NUM_OF_MODULES = 2	リンクモジュールの数
\$MN_MM_SERVO_FIFO_SIZE = 4	補間と位置制御間のデータバッファのサイズは、4 に増えます。
\$MN_AXCONF_LOGIC_MACHAX_TAB[0] = 「AX1」	
\$MN_AXCONF_LOGIC_MACHAX_TAB[1] = 「AX2」	
\$MN_AXCONF_LOGIC_MACHAX_TAB[2] = 「AX3」	
\$MN_AXCONF_MACHAX_NAME_TAB[0] = 「XM1」	
\$MN_AXCONF_MACHAX_NAME_TAB[2] = 「YM1」	
\$MA_AXCONF_ASSIGN_MASTER_NCK[ AX3 ] = 1	
\$MC_AXCONF_MACHAX_USED[0]=1 ; X	
\$MC_AXCONF_MACHAX_USED[1]=2 ; Y	
\$MC_AXCONF_MACHAX_USED[2]=3 ; Z	

## NCU2 のマシンデータ(スレーブ軸)

マシンデータ	意味
\$MN_NCU_LINKNO = 2	2. 番目の NCU 番号
\$MN_MM_NCU_LINK_MASK = 1	リンクの起動
\$MN_MM_NUM_CURVE_TABS = 5	カーブテーブルの数
\$MN_MM_LINK_NUM_OF_MODULES = 2	リンクモジュールの数
\$MN_MM_NUM_CURVE_SEGMENTS = 50	
\$MN_MM_NUM_CURVE_POLYNOMS = 100	
\$MN_MM_SERVO_FIFO_SIZE = 2	補間と位置制御間のデータバッファのサイズ(標準)

3.3 例

マシンデータ	意味
\$MN_AXCONF_LOGIC_MACHAX_TAB[0] = 「NC1_AX3」	NCU1/AX3 上のマスタリンク
\$MN_AXCONF_LOGIC_MACHAX_TAB[1] = 「AX2」	
\$MN_AXCONF_LOGIC_MACHAX_TAB[2] = 「AX3」	
\$MC_AXCONF_MACHAX_USED[0]=3	AX3 は、1 番目のチャンネル軸の機械軸です。
\$MC_AXCONF_MACHAX_USED[1]=2	AX2 は、2 番目のチャンネル軸の機械軸です。
\$MC_AXCONF_MACHAX_USED[2]=1	NCU1 の AX3 は、3 番目のチャンネル軸の機械軸です。

3.3.5.2 プログラミング

NCU1 のプログラム(マスタ軸)

NCU1 はマスタ軸 Z を移動

NCU1 のマスタ軸が割り当てられた NCU2 の識別子: リンク変数\$A\_DLB[0] = 1

NCU1 のマスタ軸である NCU2 の識別子が解放されています: リンク変数\$A\_DLB[0] = 0

プログラムコード	コメント
N1000 R1 = 0	; ループカウンタの初期化
N1004 G1 Z0 F1000	; 軸 Z を開始位置に移動
N1005 \$A_DLB[0] = 1	; NCU2 の識別子: 軸 Z の割り当て
LOOP10:	;
N1005 R1=R1+1	; ループカウンタの加算
N1006 Z0.01 G91	; マスタ値軸 Z の移動
N1008 Z0.02	; マスタ値軸 Z の移動
N1010 Z0.03	; マスタ値軸 Z の移動
N1012 IF R1 < 10 GOTOB LOOP10	;
N1098 \$A_DLB[0] = 0	; NCU2 の識別子: 軸 Z の解放

## NCU2(スレーブ軸)のプログラム

このプログラムは、NCU1 上のマスタ軸の移動と、NCU2 上のスレーブ軸の移動の間の結合をカーブテーブルによって確立します。テーブルが定義されている場合、NCU1 がマスタ軸として軸 Z を割り当てるまで(N1005)、NCU2 は待機位置(N2006)に移行します。軸 Z がマスタ軸として割り当てられるとすぐに、連結が有効になります(N2010)。NCU1 がマスタ軸として軸 Z を解放するまで、この連結は維持されます。

プログラムコード	コメント
N2000 CTABDEL(1)	; テーブル 1 の初期化
N2001 G04 F.1	;
N2003 G0 Y0 Z0	; 軸 Y、Z を開始位置に移動
N2002 CTABDEF(Y, Z, 1, 0)	; テーブル定義のオン
N2003 G1 X0 Y0	; 中間点 1
N2004 G1 X100 Y200	; 中間点 2
N2005 CTABEND	; テーブル定義のオフ
LOOP20:	;
N2006 IF (\$A_DLB[0] == 0) GOTOB LOOP20	; NCU1 の待機
N2010 LEADON(Y,Z,1)	; => 連結の完了
LOOP25:	;
N2030 IF (\$A_DLB[0] > 0) GOTOB LOOP25	; NCU1 がマスタ値軸を移動しなくなるまで連結を維持
N2090 LEADOF(Y,Z)	; => 連結を開く

## 3.4 データリスト

### 3.4.1 マシンデータ

#### 3.4.1.1 一般マシンデータ

番号	識別子: \$MN_	説明
10002	AXCONF_LOGIC_MACHAX_TAB	論理 NCU 機械軸イメージ
10065	POSCTRL_DESVAL_DELAY	位置指令値遅延
10134	MM_NUM_MMC_UNITS	MMC 同時通信相手の数
12510	NCU_LINKNO	NCU グループの NCU 番号
12520	LINK_TERMINATION	バス終端抵抗が有効な NCU 番号
12530	LINK_NUM_OF_MODULES	リンクモジュールの数

## 3.4 データリスト

番号	識別子: \$MN_	説明
12701	AXCT_AXCONF_ASSIGN_TAB1	軸コンテナ軸のリスト
...	...	
12716	AXCT_AXCONF_ASSIGN_TAB16	
12750	AXCT_NAME_TAB	軸コンテナ名称のリスト
12760	AXCT_FUNCTION_MASK	軸コンテナの機能
18700	MM_SIZEOF_LINKVAR_DATA	リンク変数メモリのサイズ
18720	MM_SERVO_FIFO_SIZE	補間と位置コントローラ間のデータバッファのサイズ
18780	MM_NCU_LINK_MASK, bit 0	リンク通信の起動

## 3.4.1.2 チャンネル別マシンデータ

番号	識別子: \$MC_	説明
20000	CHAN_NAME	チャンネル名
20070	AXCONF_MACHAX_USED	チャンネルで有効な機械軸番号
28160	MM_NUM_LINKVAR_ELEMENTS	NCU リンク変数の書き込み要素数

## 3.4.1.3 軸/主軸別マシンデータ

番号	識別子: \$MA_	説明
30550	AXCONF_ASSIGN_MASTER_CHAN	軸とチャンネル間の割り付けの初期設定
30554	AXCONF_ASSIGN_MASTER_NCU	この軸の指令値を生成する NCU を定義する初期設定
30560	IS_LOCAL_LINK_AXIS	軸は、ローカルリンク軸です。
32990	POCTRL_DESVAL_DELAY_INFO	現在の位置指令値の遅延

### 3.4.2 セッティングデータ

#### 3.4.2.1 一般セッティングデータ

番号	識別子: \$SA	説明
41700	AXCT_SWWIDTH[コンテナ番号]	軸コンテナ回転の設定

#### 3.4.2.2 軸/主軸別セッティングデータ

番号	識別子: \$SA_	説明
43300	ASSIGN_FEED_PER_REV_SOURCE	位置決め軸/主軸の毎回転送り速度

### 3.4.3 信号

#### 3.4.3.1 NC からの信号

信号名称	SINUMERIK 840D sl
MCP1 準備完了	DB10.DBX104.0
MCP2 準備完了	DB10.DBX104.1
HHU 準備完了	DB10.DBX104.2
NCU リンク有効	DB10.DBX107.6
HMI2-CPU 準備完了(HMI は OPI または MPI に接続)	DB10.DBX108.1
MPI 準備完了での HMI1-CPU	DB10.DBX108.2
OPI 準備完了での HMI1-CPU(標準接続)	DB10.DBX108.3

3.4 データリスト

3.4.3.2 HMI/PLC からの信号

信号名称	SINUMERIK 840D sl
ONL_REQUEST HMI からのオンライン要求	DB19.DBB100
ONL_CONFIRM オンライン要求に対する PLC からの応答	DB19.DBB102
PAR_CLIENT_IDENT HMI は自身のクライアント識別子(バスタイプ、 HMI バスアドレス)を書き出します	DB19.DBB104
PAR_MMC_TYP NETNAMES.INI に応じた HMI タイプ: 主/副操作パ ネルアラームサーバ	DB19.DBB106
PAR_MSTT_ADR HMI は有効にするアドレスを MCP に書き出します	DB19.DBB107
PAR_STATUS PLC は HMI に対してオンライン有効を書き出しま す(接続状態)	DB19.DBB108
PAR_Z_INFO PLC は追加情報を接続状態に書き出します	DB19.DBB109
M_TO_N_ALIVE M ~ N ブロックを使用した PLC から HMI へのラ イフサイン	DB19.DBB110

## 3.4.3.3 一般オンラインインタフェース

信号名称	SINUMERIK 840D sl
MMC1_CLIENT_IDENT	DB19.DBB120
HMI がオンラインになると、PLC は PAR_CLIENT_IDENT を MMCx_CLIENT_IDENT に書き出します。	
MMC1_TYP	DB19.DBB122
HMI がオンラインになると、PLC は PAR_MMC_TYP を MMCx_TYP に書き出します。	
MMC1_MSTT_ADR	DB19.DBB123
HMI がオンラインになると、PLC は PAR_MSTT_ADR を MMCx_MSTT_ADR に書き出します。	
MMC1_STATUS	DB19.DBB124
接続状態、HMI と PLC は要求/応答を交互に書き出します。	
MMC1_Z_INFO	DB19.DBB125
追加情報、接続状態(肯定/否定応答、アラームメッセージなど)	
MMC1_SHIFT_LOCK	DB19.DBX126.0
HMI 切替ロック	
MMC1_MSTT_SHIFT_LOCK	DB19.DBX126.1
MCP 切替ロック	
MMC1_ACTIVE_REQ	DB19.DBX126.2
HMI は有効なオペレータモードを要求します	
MMC1_ACTIVE_PERM	DB19.DBX126.3
PLC からのオペレータモードの変更を有効にします	
MMC1_ACTIVE_CHANGED	DB19.DBX126.4
HMI がオペレータモードを変更しました	
MMC1_CHANGE_DENIED	DB19.DBX126.5
HMI のアクティブ/パッシブの切替が拒否されました	

## 3.4 データリスト

信号名称	SINUMERIK 840D sl
MMC2_CLIENT_IDENT	DB19.DBB130
HMI がオンラインになると、PLC は PAR_CLIENT_IDENT を MMCx_CLIENT_IDENT に書き出します。	
MMC2_TYP	DB19.DBB132
HMI がオンラインになると、PLC は PAR_MMC_TYP を MMCx_TYP に書き出します。	
MMC2_MSTT_ADR	DB19.DBB133
HMI がオンラインになると、PLC は PAR_MSTT_ADR を MMCx_MSTT_ADR に書き出します。	
MMC2_STATUS	DB19.DBB134
接続状態、HMI と PLC は要求/応答を交互に書き出します。	
MMC2_Z_INFO	DB19.DBB135
追加情報、接続状態(肯定/否定応答、アラームメッセージなど)	
MMC2_SHIFT_LOCK	DB19.DBX136.0
HMI 切替ロック	
MMC2_MSTT_SHIFT_LOCK	DB19.DBX136.1
MCP 切替ロック	
MMC2_ACTIVE_REQ	DB19.DBX136.2
HMI は有効なオペレータモードを要求します	
MMC2_ACTIVE_PERM	DB19.DBX136.3
PLC からのオペレータモードの変更を有効にします	
MMC2_ACTIVE_CHANGED	DB19.DBX136.4
HMI がオペレータモードを変更しました	
MMC2_CHANGE_DENIED	DB19.DBX136.5
HMI のアクティブ/パッシブの切替が拒否されました	

## 3.4.3.4 軸/主軸からの信号

信号名称	SINUMERIK 840D sl	SINUMERIK 828D
NCU リンク:軸が有効	DB31、...DBX60.1	-
軸別アラーム	DB31、...DBX61.1	DB390x, DBX1.1
軸準備完了	DB31、...DBX61.2	DB390x, DBX1.2
軸コンテナ:回転が有効	DB31、...DBX62.7	-

## 3.4.4 システム変数

システム変数	説明
\$AN_AXCTSWE[axis]	指定した軸の軸コンテナのスロットを提供し、これを次の軸コンテナの回転で有効にします
\$AN_LAI_AX_IS_AXCTAX	ビット領域として論理機械軸イメージのコンテナ軸を含めます
\$AN_LAI_AX_IS_LINKAX	ビット領域として論理機械軸イメージのリンク軸を含めます
\$AN_LAI_AX_IS_LEADLINKAX	ビット領域として論理機械軸イメージのマスタリンク軸を含めます
\$AN_LAI_AX_TO_MACHAX[axis]	論理機械軸イメージの指定した軸に対して、NCU-ID、と関連する機械軸の軸番号を提供します
\$AN_LAI_AX_TO_IPO_NC_CHANAX[axis]	論理機械軸イメージの指定した軸に対して、チャンネル番号とチャンネル軸番号か、NCU 番号とグローバル軸番号またはそのすべてを提供します
\$AN_IPO_CHANAX[global axis number]	指定したグローバル軸番号に対して、チャンネル番号とチャンネル軸番号を提供します
\$AA_MACHAX[axis]	指定した軸に対して、NCU-ID と機械軸番号を提供します
\$AA_IPO_NC_CHANAX[axis]	指定した軸に対して、チャンネル番号とチャンネル軸番号またはNCU-ID とグローバル軸番号を提供します
\$VA_IPO_NC_CHANAX[axis]	指定した機械軸に対して、チャンネル番号とチャンネル軸番号またはNCU-ID とグローバル軸番号を提供します

システム変数の詳細説明については、

参照先:

『システム変数』リストマニュアルを参照してください。

### 3.4 データリスト

## H1:手動移動

### 4.1 概要

#### 用途

最新の NC 制御工作機械であっても、ユーザが軸を手動で移動するための装置を設ける必要があります。

例:

- 機械のセットアップ  
特に、新規の加工プログラムを設定している場合、機械軸の手動の移動が必要です。座標オフセットまたは座標回転が選択されている場合は、座標変換後のワーク座標系で手動移動をおこなうこともできます。
- プログラム中止後の工具の後退  
停電またはリセットによるプログラム中止の後、オペレータは手動で現在の加工位置から工具を後退する必要があります。この操作中、加工で使用する座標変換と座標系は、そのまま有効にしてください。

#### JOG モード

軸の手動による移動は、JOG モードで行います。

ジョグモードが現在のモードグループで有効な場合、対応するインタフェース信号によって PLC に報告されます。

DB11 DBX6.2 (BAG1:ジョグ運転モード)

DB11 DBX26.2 (BAG2:ジョグ運転モード)

...

DB11 DBX186.2 (BAG10:ジョグ運転モード)

#### 参照先:

『機能マニュアル、基本機能』; 「モードグループ、チャンネル、プログラム運転、リセット応答 (K1)」

## 4.1 概要

### ディスプレイ

JOG モードが有効な場合、ユーザーインターフェースに基本 JOG 画面が表示されます。この主画面には、位置、送り速度、主軸、および工具に関連する値が含まれます。

### BCS および WCS での手動による移動

軸を移動する場合、基本座標系(BCS)またはワーク座標系(WCS)を使用するオプションがあります。

- BCS での手動による移動  
各軸は、手動で移動できます。
- WCS での手動による移動  
WCS では、ジオメトリ軸と宣言した軸のみを手動で移動できます(チャンネル別)。選択されている座標オフセットまたは座標回転は、そのまま有効です。

### 複数の軸の手動による移動

JOG モードで複数の軸を同時に手動で移動できます。ただし、補間関係はありません。

### PLC インターフェースを使った制御

手動による移動を PLC インターフェースで制御します。

「PLC インターフェースを使った制御 (ページ 173)」の章を参照してください。

機械軸(軸別)、ジオメトリ軸(チャンネル別)、および旋回軸(チャンネル別)それぞれに、別の PLC インターフェースが用意されています。

### パラメータ割り付け(一般)

軸の手動による移動のため、移動タイプおよび選択された JOG 機能に関係なく、すべての JOG 移動に有効なパラメータ設定可能な特性があります。これらについては、「パラメータ割り付け(一般) (ページ 175)」を参照してください。

## 移動タイプ

JOG モードでは、以下の手動移動タイプが可能です。

- JOG または連続モードで機械操作パネルの移動キーを使った連続移動  
「連続手動移動 (ページ 181)」の章を参照してください。
- JOG または連続モードで機械操作パネルの移動キーを使ったインクリメンタル移動  
「インクリメンタル手動移動 (ページ 185)」の章を参照してください。
- 手動パルス発生器による軸の移動  
「ハンドルを使用した手動移動 (ページ 190)」の章を参照してください。

## 主軸の手動による移動

主軸もまた、ジョグモードで手動移動ができます。基本的に、軸の手動移動と同じ条件が適用されます。主軸は、ジョグモードで移動キーを使用して連続的に、あるいはジョグまたは連続モードで、またはハンドルを使用してステップで移動できます。この機能は、機械軸の場合と同じ方法で、軸/主軸別 PLC インタフェースによって選択して、有効にします。軸別マシンデータはまた、主軸にも適用されます。

手動による主軸の移動中に遵守すべき特記事項については、「主軸の手動による移動 (ページ 208)」の章を参照してください。

## ジオメトリ軸の手動による移動

ジオメトリ軸の手動による移動は、座標変換および境界線が有効である必要のある移動に使用します。ジオメトリ軸は、最新の有効座標系上で移動します。

手動によるジオメトリ軸(および旋回軸)の移動中に遵守すべき特記事項については、「ジオメトリ軸/旋回軸の手動による移動 (ページ 210)」の章を参照してください。

## その他の機能

JOG モードでは、さまざまな移動タイプに加え、他の機能も選択できます。

- REF (レファレンス点復帰)
- REPOS (再位置決め)
- 設定された固定点への移動  
「JOG モードの固有位置へのアプローチ (ページ 213)」の章を参照してください。
- 定義可能位置への移動  
「JOG での位置移動 (ページ 222)」の章を参照してください。

## 4.1 概要

- 円弧に沿ったジオメトリ軸の移動  
「JOG での円移動 (ページ 228)」の章を参照してください。
- JOG 後退(停電またはリセットによるプログラム中止後の工具方向の後退移動)  
「工具方向の後退(ジョグ後退) (ページ 240)」の章を参照してください。

### 自動モードでのハンドルの使用

ハンドルは、自動モードでも使用できます。

- パス指定および(または)速度オーバーライドによる自動モードでのハンドルオーバーライド (ページ 251)
- ハンドルを使用した輪郭ハンドル/軌跡指令(オプション) (ページ 260)
- 手動パルス発生器(DRF)を使用した自動モード時の追加インクレメンタルゼロオフセットによる加工用工具摩耗の補正 (ページ 251)

### モニタリング

手動による移動では、制限が適用されます。これは、コントローラが監視します。

「監視機能 (ページ 275)」の章を参照してください。

### JOG モードから自動/MDI モードへの切り替え

チャンネルのすべての軸が「汎用イグザクトストップ」に到達している場合にのみ、運転モードを JOG から自動または MDI に切り替えることができます。

#### 参照先:

『機能マニュアル、基本機能』; 「モードグループ、チャンネル、プログラム運転、リセット応答 (K1)」

### セットアップ

手動パルス発生器のセットアップについては、「セットアップ:ハンドル (ページ 276)」の章を参照してください。

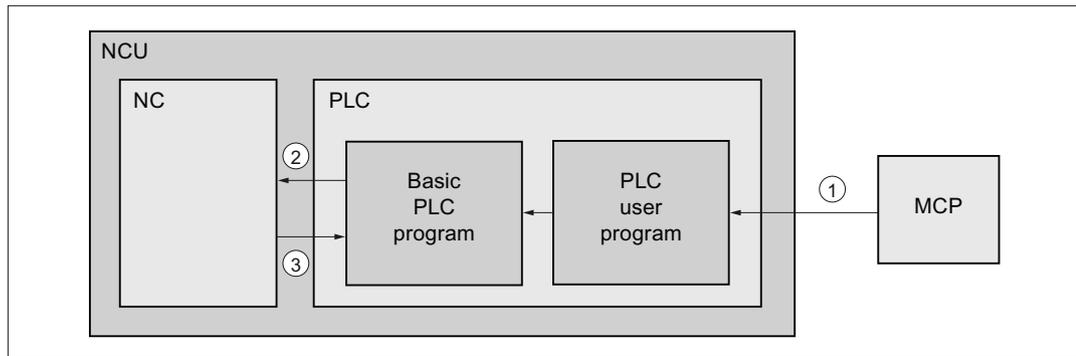
## 4.2 PLC インターフェースを使った制御

NC の JOG 機能は、機械操作パネル(MCP)から PLC インタフェースにより有効にします。このためには、PLC から機械操作パネルの入力信号を NC/PLC インタフェースでNC の入力インタフェースに伝送する必要があります。これにより工作機械メーカーは、PLC ユーザープログラムにより工作機械へ手動移動機能を簡単に設定できます。たとえば、機械操作パネルの方向キーと、機械軸とジオメトリ軸に関する NC への移動要求の間の割り当てを変更するために、この機能を使用できます。

### 基本手順

下記の図は、機械操作パネル(MCP)から NC への JOG 機能を選択するための基本的な手順を示しています。

4.2 PLC インターフェースを使った制御



- ① たとえばオペレータは、機械軸の機械操作パネルで「連続移動」機械機能を選択します。

MCP の入力信号は、MCP 入力インタフェースのデータブロックで、PLC 基本プログラムから周期的に伝送されます。

- ② PLC ユーザープログラムは、MCP の入力信号を読み取ります。入力信号は、現在の機械または加工の状態に従って、他の任意の信号とリンクできます。最後に PLC ユーザープログラムは、関連する軸別の NC/PLC インタフェースで、対応する要求信号を NC に書き込みます。PLC 基本プログラムは、内部入力インタフェースの要求信号を NC に伝送します。
- ③ 要求された機能の起動後、NC は内部出力インタフェースでフィードバックを PLC に書き込みます。基本 PLC プログラムは、周期的にそれぞれの軸別の NC/PLC インタフェースの出力信号を伝送します。

#### 参照先

PLC ユーザープログラムの機械操作パネルの設定と組み込みに関する詳細情報については、『基本機能マニュアル』を参照してください。

- SINUMERIK 840D sl:"P3:SINUMERIK 840D sl 用 PLC 基本プログラム
- SINUMERIK 828D:"P4:SINUMERIK 828D 用 PLC

## 4.3 パラメータ割り付け(一般)

### 送り速度の特性

JOG モードでは、メイン主軸の現在の速度に対して、指定の(G95 による)毎回転送り速度で軸/主軸を手動で移動することもできます。

この設定は、以下のセッティングデータを使用して行います。

SD41100 \$SN\_JOG\_REV\_IS\_ACTIVE (JOG:毎回転/毎分送り)

ビット	値	意味
0	0	軸/主軸の特性は、次のセッティングデータによって決まります。 SD43300 \$SA_ASSIGN_FEED_PER_REV_SOURCE (位置決め軸/主軸の毎回転送り速度) 回転が有効なフレームに対するジオメトリ軸または旋回軸の特性は、チャンネルセッティングデータによって決まります。 SD42600 \$SC_JOG_FEED_PER_REV_SOURCE(ジョグの毎回転送り速度の制御)。
	1	JOG の移動の場合、軸/主軸は、メイン主軸に応じて、毎回転送り速度で常に移動します。

## 4.3 パラメータ割り付け(一般)

ビット	値	意味
1	0	高速処理の場合も、軸/主軸、ジオメトリ軸または旋回軸が毎回転送り速度で移動します(選択については、ビット 0 を参照)。
	1	高速処理の場合、軸/主軸、ジオメトリ軸または旋回軸が必ず毎回転送り速度なしで移動します。
2	0	JOG モードでハンドルを使用した手動移動の場合は、軸/主軸、ジオメトリ軸または旋回軸が毎回転送り速度で移動します(選択については、ビット 0 を参照)。
	1	JOG モードでハンドルを使用した手動移動の場合は、軸/主軸、ジオメトリ軸または旋回軸が常に毎回転送り速度で移動します。
3	0	DRF モードでハンドルを使用した手動移動の場合は、軸/主軸が毎回転送り速度で移動します(選択については、ビット 0 を参照)。
	1	DRF モードでハンドルを使用した手動移動の場合は、軸/主軸が常に、毎回転送り速度以外で移動します。

## 注記

メイン主軸が定義されていない場合に、手動で軸を毎回転送り速度で移動しようとする  
と、アラーム 20055 が出力されます(ジオメトリ軸のアラームは 20065 です)。

## 速度

手動による移動の速度を決定するシステムデータは、有効な送り速度特性によって異なります。

**毎分送り(G94)が有効**

## ● 機械軸

速度は、以下のセッティングデータで決定します。

– 直線軸の場合:

SD41110 \$SN\_JOG\_SET\_VELO (ジョグの軸速度)

– 回転軸の場合:

SD41130 \$SN\_JOG\_ROT\_AX\_SET\_VELO (回転軸の JOG 速度)

SD41110/SD41130 に「0」が入力されている場合、軸マシンデータが有効になります。

MD32020 \$MA\_JOG\_VELO (手動軸速度)

送り速度の速度単位のデフォルト設定は、mm/min または回転軸の場合は rpm です。

## ● 主軸

主軸の手動による移動に適用される速度は、次のセッティングデータによって定義されます。

SD41200 \$SN\_JOG\_SPIND\_SET\_VELO (JOG モードでの主軸の速度)

SD41200 に「0」が入力されている場合、軸マシンデータが有効になります。

MD32020 \$ \_MA\_JOG\_VELO (手動による軸速度)

主軸を手動で移動する場合、有効なギヤ選択の最高速度が考慮されます。

**参照先:**

『機能マニュアル、基本機能』；「主軸(S1)」

## ● ジオメトリ軸

ジオメトリ軸の手動による移動に適用される速度は、次のチャンネル別マシンデータによって定義されます。

MD21165 \$MC\_JOG\_VELO\_GEO (ジオメトリ軸の手動速度)

## ● 旋回軸

旋回軸の手動による移動に適用される速度は、次のチャンネル別マシンデータによって定義されます。

MD21155 \$MC\_JOG\_VELO\_ORI (旋回軸の手動速度)

**毎回転送り速度(G95)が有効**

送り速度の値(mm/rev.)は、以下のセッティングデータで定義されます。

SD41120 \$SN\_JOG\_REV\_SET\_VELO (JOB モードの軸の毎回転送り)

SD41120 に「0」が入力されている場合、軸マシンデータが有効になります。

MD32050 \$MD\_JOG\_REV\_VELO (ジョグの毎回転送り)

### 4.3 パラメータ割り付け(一般)

#### 早送りオーバーライド

また、早送りオーバーライドキーと一緒にキー/ハンドルを作動させた場合、移動は設定された早送り速度で行われます。

- 機械軸
  - MD32010 \$MA\_JOG\_VELO\_RAPID (ジョグモードの早送り)
  - MD32040 \$MA\_JOG\_REV\_VELO\_RAPID (早送りオーバーライドによる JOG モードの毎回転送り速度)
- ジオメトリ軸
  - MD21160 \$MC\_JOG\_VELO\_RAPID\_GEO (ジオメトリ軸に対するジョグモードの早送り)
- 旋回軸
  - MD21150 \$MC\_JOG\_VELO\_RAPID\_ORI (旋回軸に対するジョグモードの早送り)

#### 送り速度オーバーライド

また、下記の NC/PLC インタフェース信号が指令されている場合に、軸別の送り速度オーバーライドスイッチを使用すると、手動による移動に適用された速度が影響を受けることがあります。

##### DB31、... DBX1.7 (軸別送り速度オーバーライド有効)

マシンデータによって個々の送り速度オーバーライド切り替え位置にパーセント値を割り当てます。0%の切り替え位置の場合、該当するマシンデータに 0 を入力すると、軸は移動しません。

インターフェース信号 DB31、... DBX1.7 (軸別の送り速度オーバーライド有効) は、スイッチ位置 0%では意味を持ちません。

パーセント値(0%～200%)は、送り速度オーバーライド切り替え位置(グレイコード)で設定する代わりに、オプションにより PLC で直接設定できます。ここでも、マシンデータによって選択をおこないます。

##### 参照先:

機能マニュアル、基本機能; 送り速度 (V1)

#### 主軸オーバーライド

主軸の手動による移動中の速度は、主軸オーバーライドスイッチで操作できます。

##### 参照先:

『機能マニュアル、基本機能』; 「主軸(S1)」

## 加速度と加々速度

軸固有の加速度および加々速度の制限値は、JOG モード時にも有効です。

また、ジオメトリ軸と旋回軸の手動での移動に対して、加速度と加々速度をチャンネル別に制限することも可能です。これにより、回転軸で全ての直交移動を生成するキネマティックの処理を設定できます。(ロボット)

### JOG 移動の最高軸別加速度

手動での軸の移動に対する最大加速度は、マシンデータで定義します。

MD32300 \$MA\_MAX\_AX\_ACCEL [0] (DYNORM ダイナミック応答モード DYNORM での軌跡移動に対する最大軸別加速度)

---

### 注記

JOG モードでは、常にダイナミック応答モード DYNORM のみが有効です。

---

### 軸別の加々速度制限の初期設定

加々速度制限が指定された加減速は、JOG モードの軸別基本設定として指定できます。

MD32420 \$MA\_JOG\_AND\_POS\_JERK\_ENABLE == TRUE

### 軸別の最大加々速度

加々速度制限が有効な JOG モードの場合、次に示すマシンデータの値が軸別の最大加々速度として有効になります。

MD32430 \$MA\_JOG\_AND\_POS\_MAX\_JERK (軸別の最大軸加々速度)

---

### 注記

MD32430 \$MA\_JOG\_AND\_POS\_MAX\_JERK は、JOB 移動の軸別加々速度の制限が、移動する機械軸について有効になっている場合にのみ機能します。

MD32420 \$MA\_JOG\_AND\_POS\_JERK\_ENABLE [<軸>] == TRUE

これは、パートプログラムで SOFTA (軸 1>, <軸 2>, ...) をプログラム指令することでも可能です。

---

### 主軸の手動移動時の最大加速度

主軸は速度制御モードと位置制御モードで複数のギヤ選択を使用することが多いので、現在のギヤ選択に関連する加速度は、常に JOG モードの主軸に適用されます。

### 参照先:

『機能マニュアル、基本機能』; 「主軸(S1)」

4.3 パラメータ割り付け(一般)

**ジオメトリ軸の手動移動時の最大加速度**

ジオメトリ軸を手動で移動する場合の最高加速度は、次のマシンデータを使用してチャンネル毎に指定することができます。

MD21166 \$MC\_JOG\_ACCEL\_GEO [<ジオメトリ軸>]

<ジオメトリ軸> = 0、1、2

---

**注記**

MD21166 \$MC\_JOG\_ACCEL\_GEO [<ジオメトリ軸>]の場合、MD32300 \$MA\_MAX\_AX\_ACCEL に対する直接の制限はありません。

---

**注記**

座標変換が有効の場合、MD32300 \$MA\_MAX\_AX\_ACCEL によって最高許容軸別加速度が決まります。

**ジオメトリ軸の手動移動時の最大加々速度**

SOFT 加減速モード(加々速度一定加減速)でジオメトリ軸を手動で移動する場合の最高加々速度は、次のマシンデータを使用してチャンネル毎に指定することができます。

MD21168 \$MC\_JOG\_JERK\_GEO [<ジオメトリ軸>]

<ジオメトリ軸> = 0、1、2

---

**注記**

MD21168 \$MC\_JOG\_JERK\_GEO は、JOG モードの軸別加々速度制限が基本機械軸に対して有効になっている場合にだけ作用します。

MD32420 \$MA\_JOG\_AND\_POS\_JERK\_ENABLE [<軸>] == TRUE

---

MD21168 = 0 の場合、チャンネル別の加速度制限の代わりに MD32430 \$MA\_JOG\_AND\_POS\_MAX\_JERK の軸別制限値が有効になります。

**旋回軸の手動移動時の最大加々速度**

次のマシンデータによって、各チャンネルについて、手動で旋回軸を移動時の最高加々速度を指定できます。

MD21158 \$MC\_JOG\_JERK\_ORI [<旋回軸>]

MD21158 を有効にするには、次のマシンデータによって、旋回軸の手動移動のチャンネル別加々速度制限を有効にしてください。

MD21159 \$MC\_JOG\_JERK\_ORI\_ENABLE == TRUE

下記も参照

ジオメトリ軸/旋回軸の手動による移動 (ページ 210)

主軸の手動による移動 (ページ 208)

## 4.4 連続手動移動

### 4.4.1 機能

連続手動移動の場合、プラス/マイナス移動キーを選択すると、当該軸が指定方向に移動します。両方の移動キーを同時に押すと、移動動作は行われません。また、軸が移動中である場合、移動を中止します。

#### 注記

制御の切り替え時、軸はまだ原点確立されていないので、機械のリミット位置に移動する場合があります。ハードウェアリミットスイッチが動作することがあります。ソフトリミットスイッチとワーキングエリアリミットは有効ではありません。

### ジョグモードでの連続移動

ジョグモードでは、移動キーを押している限り、軸が移動します。移動キーを離すと、軸は停止状態まで減速します。設定された軸のイグザクトストップ条件に達すると、移動が完了します。

移動キーを離す前に軸が移動範囲リミットに達した場合(作業領域リミット、ソフトウェアリミットスイッチなど)、軸はリミットで停止します。

### 連続モードの連続移動

連続モードでは、移動キーを押すことにより軸の移動が開始します。この移動動作は、移動キーを離しても継続します。



**警告**

#### 衝突の可能性

連続モードでは、対応する移動キーを押すことにより複数の軸を起動できます。インタロックは、ユーザー/工作機械メーカーが PLC ユーザープログラムによって組み込んでください。

4.4 連続手動移動

移動動作は、オペレータによっていつでも中断と続行、または中止できます。移動動作がキャンセルされる前に軸が移動範囲リミットに達した場合(作業領域リミット、ソフトウェアリミットスイッチなど)、軸はリミットで停止します。

**移動動作の中断**

オペレータは、下記の方法で機械操作パネル(MCP)のオペレータ制御により移動を中断できます。

- 送り速度オーバーライド = 0%
- 送り停止
- NC ストップまたは NC ストップの軸/主軸

中断の原因を解消すると、軸は移動を継続します。

**移動動作の中止**

オペレータは、下記の方法で機械操作パネル(MCP)のオペレータ制御により移動をキャンセルできます。

- 同じ移動キー再度押す
- 反対方向の移動キーを押す
- RESET
- 運転モードを AUTOMATIC または MDI に変更することによってジョグモードを解除する

コントローラによって移動動作がキャンセルされます。

- 有効な移動範囲リミットに達した場合(作業領域リミット、ソフトウェアリミットスイッチなど)

 <b>注意</b>
<b>移動範囲のリミットが無効の場合</b> ソフトウェアリミットスイッチと作業領域リミットは、軸のレファレンス点復帰動作の後でのみ有効です。

- 移動動作のキャンセルによりアラームが発生した場合

## インタフェース信号

### 選択

PLC インターフェースを使って、JOG モードでの連続移動を有効にしてください。

- 機械軸:
  - DB31、... DBX5.6 (機械機能「連続手動移動」の要求)
- ジオメトリ軸:
  - DB21、... DBX13.6 (ジオメトリ軸 1:機械機能「連続手動移動」の要求)
  - DB21、... DBX17.6 (ジオメトリ軸 2:機械機能「連続手動移動」の要求)
  - DB21、... DBX21.6 (ジオメトリ軸 3:機械機能「連続手動移動」の要求)
- 回転軸:
  - DB21、... DBX321.6 (回転軸 1:機械機能「連続手動移動」の要求)
  - DB21、... DBX325.6 (回転軸 2:機械機能「連続手動移動」の要求)
  - DB21、... DBX329.6 (回転軸 3:機械機能「連続手動移動」の要求)

### 予備充電の

連続手順が有効になると、PLC にフィードバックが送信されます。

- 機械軸:
  - DB31、... DBX65.6 (有効な機械機能「連続手動移動」)
- ジオメトリ軸:
  - DB21、... DBX41.6 (ジオメトリ軸 1:有効な機械機能「連続手動移動」)
  - DB21、... DBX47.6 (ジオメトリ軸 2:有効な機械機能「連続手動移動」)
  - DB21、... DBX53.6 (ジオメトリ軸 3:有効な機械機能「連続手動移動」)
- 回転軸:
  - DB21、... DBX333.6 (回転軸 1:有効な機械機能「連続手動移動」)
  - DB21、... DBX337.6 (回転軸 2:有効な機械機能「連続手動移動」)
  - DB21、... DBX341.6 (回転軸 3:有効な機械機能「連続手動移動」)

### 移動指令プラス/マイナス

軸の移動要求が保留されると、移動方向に応じてインターフェース信号「移動指令マイナス」または「移動指令プラス」が PLC に出力されます。

- 機械軸:
  - DB31、... DBX64.6 (移動指令「マイナス」)または
  - DB31、... DBX64.7 (移動指令「プラス」)
- ジオメトリ軸 1:
  - DB21、... DBX40.6 (ジオメトリ軸 1:移動指令「マイナス」)または
  - DB21、... DBX40.7 (ジオメトリ軸 1:移動指令「プラス」)
- ジオメトリ軸 2:
  - DB21、... DBX46.6 (ジオメトリ軸 2:移動指令「マイナス」)または
  - DB21、... DBX46.7 (ジオメトリ軸 2:移動指令「プラス」)
- ジオメトリ軸 3:
  - DB21、... DBX52.6 (ジオメトリ軸 3:移動指令「マイナス」)または
  - DB21、... DBX52.7 (ジオメトリ軸 3:移動指令「プラス」)
- 旋回軸 1;
  - DB21、... DBX332.6 (旋回軸 1:移動指令「マイナス」)または
  - DB21、... DBX332.7 (旋回軸 1:移動指令「プラス」)
- 旋回軸 2;
  - DB21、... DBX336.6 (旋回軸 2:移動指令「マイナス」)または
  - DB21、... DBX336.7 (旋回軸 2:移動指令「プラス」)
- 旋回軸 3;
  - DB21、... DBX340.6 (旋回軸 3:移動指令「マイナス」)または
  - DB21、... DBX340.7 (旋回軸 3:移動指令「プラス」)

---

### 注記

移動する軸のコントローライネーブルまたはパルスイネーブルが欠落している場合、JOG モードに設定された移動指令がキャンセルされます。したがって、軸の解放後に移動指令を再度設定してください。

---

## 4.4.2 パラメータ設定

### JOG または連続モード

連続作業のためのジョグモードまたは連続モードの選択は、下記のセッティングデータによって、すべての軸について NC 別に行われます。

SD41050 \$SN\_JOG\_CONT\_MODE\_LEVELTRIGGRD (ジョグ連続用ジョグ/連続モード)

ジョグモードは初期設定です。

## 4.4.3 必要条件

### 割り出し軸

割り出し軸は常に、ジョグモードと連続モードの両方で割り出し位置にて停止します。JOG モードで、たとえば移動キーを離しても、割り出し軸は移動方向の次の割り出し位置に移動します。

### 下記も参照

T1:割り出し軸 (ページ 991)

## 4.5 インクリメンタル手動移動

### 4.5.1 機能

ステップ移動では、オペレータは機械操作パネルで軸の移動インCREMENT数を指定します。

また、5つの固定ステップ量(初期設定:INC1、INC10、INC100、INC1000、およびINC10000)、セッティングデータによって設定できる可変ステップ量(INCvar)が使用可能です。

## 4.5 インクリメンタル手動移動

### ジョグモードでのステップ移動

目的の方向の移動キー(例: +)を押すと、設定されたステップ値で軸が移動を開始します。ステップ値を完全に移動する前に移動キーを離すと、移動が中断され、軸が停止します。同じ移動キーを再び押すと、軸は残移動距離がゼロになるまで移動します。この点までは、移動キーを離すことによって、まだ移動を中断できます。

ステップ値が完全に移動されるか、または移動が中止されるまで、反対方向の移動キーを押しても無効です。

#### 移動プロセスのキャンセル

ステップ量を終点まで移動させない場合は、次のようにして移動を中断できます。

- RESET
- DB31、... DBX2.2 (残移動距離削除)

### 連続モードのステップ移動

移動キーを押すと(第1の立ち上がりエッジ)、設定された全ステップ値を軸が移動します。軸がステップ値の移動を終える前に同じ移動キーを再度押した場合(第2の立ち上がりエッジ)、移動動作が中止されます(つまり、完了しません)。

#### 移動動作の中断

連続移動に関する動作(「連続手動移動 (ページ 181)」を参照)。

#### 移動プロセスのキャンセル

移動動作は、次の操作または監視機能によって停止および中止することができます。

- 同じ移動キー再度押す
- 反対方向の移動キーを押す
- RESET
- DB31、... DBX2.2 (残移動距離削除)
- 最初の有効なリミットに達した場合



注意

#### 移動範囲のリミットが無効の場合

ソフトウェアリミットスイッチおよびワーキングエリアリミットは、レファレンス点復帰後でなければ有効になりません。

- 現在のステップ値の選択解除または変更時(例: INC100 から INC10 への変更時)
- 障害発生時(例: サーボイネーブルのキャンセル時)

 <b>警告</b>
<b>干渉の可能性</b> 「連続」モードを選択した場合、当該の方向キーを押して離すと複数の軸を起動することができます。PLCによって何らかのインターロック機能を実施してください!

**注記**

軸の移動中、ジョグから **AUTOMATIC** または **MDI** へのモードの変更は制御装置内部で許可されません。

**インタフェース信号****選択**

PLC インタフェースを使って、**JOG** モードでのインクリメンタル移動を有効にしてください。

- 機械軸:
  - DB31、... DBX5.0 - 5 (機械機能「INC1」...「INCvar」の要求)
- ジオメトリ軸:
  - DB21、... DBX13.0 - 5 (ジオメトリ軸 1:機械機能「INC1」...「INCvar」の要求)
  - DB21、... DBX17.0 - 5 (ジオメトリ軸 2:機械機能「INC1」...「INCvar」の要求)
  - DB21、... DBX21.0 - 5 (ジオメトリ軸 3:機械機能「INC1」...「INCvar」の要求)
- 旋回軸:
  - DB21、... DBX321.0 - 5 (旋回軸 1:機械機能「INC1」...「INCvar」の要求)
  - DB21、... DBX325.0 - 5 (旋回軸 2:機械機能「INC1」...「INCvar」の要求)
  - DB21、... DBX329.0 - 5 (旋回軸 3:機械機能「INC1」...「INCvar」の要求)

## 4.5 インクリメンタル手動移動

## 予備充電の

インクリメンタル手順が有効になると、PLC にフィードバックが送信されます。

- 機械軸:
  - DB31、... DBX65.0 - 5 (有効な機械機能「INC1」...「INCvar」)
- ジオメトリ軸:
  - DB21、... DBX41.0 - 5 (ジオメトリ軸 1:有効な機械機能「INC1」...「INCvar」)
  - DB21、... DBX47.0 - 5 (ジオメトリ軸 2:有効な機械機能「INC1」...「INCvar」)
  - DB21、... DBX53.0 - 5 (ジオメトリ軸 3:有効な機械機能「INC1」...「INCvar」)
- 旋回軸:
  - DB21、... DBX333.0 - 5 (旋回軸 1:有効な機械機能「INC1」...「INCvar」)
  - DB21、... DBX337.0 - 5 (旋回軸 2:有効な機械機能「INC1」...「INCvar」)
  - DB21、... DBX341.0 - 5 (旋回軸 3:有効な機械機能「INC1」...「INCvar」)

## 移動指令プラス/マイナス

軸の移動要求が保留されると、移動方向に応じてインターフェース信号「移動指令マイナス」または「移動指令プラス」が PLC に出力されます。

- 機械軸:
  - DB31、... DBX64.6 (移動指令「マイナス」)または
  - DB31、... DBX64.7 (移動指令「プラス」)
- ジオメトリ軸 1:
  - DB21、... DBX40.6 (ジオメトリ軸 1:移動指令「マイナス」)または
  - DB21、... DBX40.7 (ジオメトリ軸 1:移動指令「プラス」)
- ジオメトリ軸 2:
  - DB21、... DBX46.6 (ジオメトリ軸 2:移動指令「マイナス」)または
  - DB21、... DBX46.7 (ジオメトリ軸 2:移動指令「プラス」)
- ジオメトリ軸 3:
  - DB21、... DBX52.6 (ジオメトリ軸 3:移動指令「マイナス」)または
  - DB21、... DBX52.7 (ジオメトリ軸 3:移動指令「プラス」)
- 旋回軸 1;
  - DB21、... DBX332.6 (旋回軸 1:移動指令「マイナス」)または
  - DB21、... DBX332.7 (旋回軸 1:移動指令「プラス」)

- 回転軸 2;
  - DB21、... DBX336.6 (回転軸 2:移動指令「マイナス」)または
  - DB21、... DBX336.7 (回転軸 2:移動指令「プラス」)
- 回転軸 3;
  - DB21、... DBX340.6 (回転軸 3:移動指令「マイナス」)または
  - DB21、... DBX340.7 (回転軸 3:移動指令「プラス」)

---

**注記**

移動する軸のコントローライネーブルまたはパルスイネーブルが欠落している場合、JOGモードに設定された移動指令がキャンセルされます。したがって、軸の解放後に移動指令を再度設定してください。

---

## 4.5.2 パラメータ設定

### 固定ステップ量

固定ステップ量のパラメータ設定は、NC 別の以下のマシンデータによっておこなわれます。

MD11330 \$MN\_JOG\_INCR\_SIZE\_TAB[1 ... 5] = <ステップ数 1 ... 5>

### 可変ステップ量

可変ステップ量のパラメータ設定は、NC 別の以下のマシンデータによっておこなわれます。

SD41010 \$SN\_JOG\_VAR\_INCR\_SIZE = <ステップ数>

### 1 ステップの距離の評価

固定と可変ステップ量の 1 ステップの距離評価は、以下の軸マシンデータによっておこなわれます。

## 4.6 ハンドルを使用した手動移動

MD31090 \$MA\_JOG\_INCR\_WEIGHT = <距離>

---

### 注記

#### 方向評価の反転

負の値を入力すると、移動キーまたは手動パルス発生器の回転方向の方向評価が反転します。

---

## JOG または連続モード

インクリメンタル手順のための JOG または連続モードの選択は、以下のマシンデータによって、すべての軸について NC 別に行われます。

MD11300 \$MN\_JOG\_INC\_MODE\_LEVELTRIGGRD (ジョグモードの INC および REF)

ジョグモードは初期設定です。

## 4.5.3 必要条件

### 割り出し軸

設定されたステップ値に関係なく、割り出し軸は移動キーを押した後に移動方向の次の割り出し位置に移動します。

### 下記も参照

T1:割り出し軸 (ページ 991)

## 4.6 ハンドルを使用した手動移動

### 4.6.1 機能

手動パルス発生器(付属品)を使用して、手動で選択した軸を同時に移動できます。ハンドル目盛の倍率は、ステップ量の倍率によって決まります。座標オフセットまたは座標回転が選択されている場合は、座標変換後のワーク座標系で手動移動をおこなうこともできます。

## セットアップ

手動パルス発生器のセットアップについては、「セットアップ:ハンドル (ページ 276)」の章を参照してください。

## 選択

ジョグモードを有効にしてください。ユーザーはまた、ハンドルを使用した手動移動に適用されるステップ値 **INC1**、**INC10** などを設定してください。ステップ移動の場合と同様に必要な機械機能は、機械機能に応じて **PLC** インタフェースで設定してください。

## 移動

手動パルス発生器を回転すると、その回転方向に応じて、関連する軸が正方向または負方向のいずれかに移動します。

---

### 注記

移動キーを使用してすでに軸が移動されている場合、ハンドルを使用した手動移動はできません。

---

## 移動距離

ハンドルを回転することによって生成される移動距離は、次の要素によって決まります。

- インタフェースで受信したハンドルパルス数
- 有効なステップ値(機械機能 **INC1**、**INC10**、**INC100**、... **INCvar**)
- 手動パルス発生器のパルス検出:  
**MD11320 \$MN\_HANDWH\_IMP\_PER\_LATCH** (1 クリック位置毎の手動パルス発生器のパルス)  
「パラメータ設定 (ページ 198)」の章を参照してください。
- ステップ値の距離:  
**MD31090 \$MA\_JOG\_INCR\_WEIGHT** (**INC**/手動パルス発生器のステップ値の検出)  
「パラメータ設定 (ページ 198)」の章を参照してください。

## ハンドル接続

最大で6基の手動パルス発生器を同時に接続できます。つまり、手動パルス発生器によって最大で**6**軸を同時に移動できます。

### NC/PLC インタフェース信号での手動パルス発生器番号の表現

MD11324 のパラメータ設定に応じて、NC/PLC インタフェース信号の手動パルス発生器の番号の表示はビットコード化(3 基の手動パルス発生器を表現できます)またはバイナリコード化(6 基の手動パルス発生器は表現できます)されます。HMI からはハンドル番号が、MD11324 の設定とは無関係に、常にビットコード化されて伝送されます。

「パラメータ設定 (ページ 198)」の章を参照してください。

### 手動パルス発生器の割り当て

ハンドルを回して移動する軸は、PLC ユーザーインターフェースまたはユーザーインターフェース(HMI)で設定できます。この割り当ては、NC/PLC ユーザプログラムによって PLC インタフェースにリンクされます。この方法では、複数の軸を 1 つのハンドルに同時に割り当てることができます。

#### PLC ユーザーインターフェースによる設定

次のいずれかのインタフェース信号を使用して割り当てをおこないます。

- 機械軸:
  - DB31、... DBX4.0 - 2 (ハンドルの起動)
- ジオメトリ軸:
  - DB21、... DBX12.0 - 2 (ジオメトリ軸 1:ハンドルの起動)
  - DB21、... DBX16.0 - 2 (ジオメトリ軸 2:ハンドルの起動)
  - DB21、... DBX20.0 - 2 (ジオメトリ軸 3:ハンドルの起動)
- 回転軸:
  - DB21、... DBX320.0 - 2 (回転軸 1:ハンドルの起動)
  - DB21、... DBX324.0 - 2 (回転軸 2:ハンドルの起動)
  - DB21、... DBX328.0 - 2 (回転軸 3:ハンドルの起動)

#### 操作画面(HMI)による設定。

ジョグモード基本メニューで「ハンドル」ソフトキーを押すと、「ハンドル」ウィンドウが表示されます。ここで、各ハンドルを軸に割り当て、ハンドルを有効または無効にできます。

---

#### 注記

3 基を超える接続手動パルス発生器に対して、および NC/PLC インタフェース信号での手動パルス発生器番号のバイナリコード表現で、ユーザーインターフェース(HMI)による手動パルス発生器の割り当ては**できません**。

---

## HMI によるハンドル選択

ユーザーインターフェースから手動パルス発生器を起動するための個別のユーザーインターフェースが HMI と PLC 間で提供されています。ハンドル 1、2、および 3 用に PLC 基本プログラムによって提供されるこのインターフェースには、次の情報が含まれています。

- ハンドルに割り当てられた**軸番号**(ハンドルの選択時に機械軸を選択した場合):
  - DB10 DBX100.0 - 4 (ハンドル 1 の軸番号)
  - DB10 DBX101.0 - 4 (ハンドル 2 の軸番号)
  - DB10 DBX102.0 - 4 (ハンドル 3 の軸番号)
- ハンドルに割り当てられた**チャンネル番号**(ハンドルの選択時にジオメトリ軸を選択した場合):
  - DB10 DBX97.0 - 3 (ジオメトリ軸ハンドル 1 のチャンネル番号)
  - DB10 DBX98.0 - 3 (ジオメトリ軸ハンドル 2 のチャンネル番号)
  - DB10 DBX99.0 - 3 (ジオメトリ軸ハンドル 3 のチャンネル番号)
- 機械軸またはジオメトリ軸に関する追加情報:
  - DB10 DBX100.7 (ハンドル 1 の機械軸)
  - DB10 DBX101.7 (ハンドル 2 の機械軸)
  - DB10 DBX102.7 (ハンドル 3 の機械軸)
- 手動パルス発生器の有効または無効の情報
  - DB10 DBX100.6 (手動パルス発生器 1 選択)
  - DB10 DBX101.6 (手動パルス発生器 2 選択)
  - DB10 DBX102.6 (手動パルス発生器 3 選択)

指定された軸について、PLC 基本プログラムは、関連するインターフェース信号を「0」(無効)または「1」(有効)に設定します。

- 機械軸:
  - DB31、... DBX4.0 - 2 (ハンドルの起動)
- ジオメトリ軸:
  - DB21、... DBX12.0 - 2 (ジオメトリ軸 1:ハンドルの起動)
  - DB21、... DBX16.0 - 2 (ジオメトリ軸 2:ハンドルの起動)
  - DB21、... DBX20.0 - 2 (ジオメトリ軸 3:ハンドルの起動)

---

**注記**

旋回軸は、以下の関連する PLC ユーザーインターフェース信号によってのみ有効にできません。

- DB21、... DBX320.0 - 2 (旋回軸 1:ハンドルの起動)
  - DB21、... DBX324.0 - 2 (旋回軸 2:ハンドルの起動)
  - DB21、... DBX328.0 - 2 (旋回軸 3:ハンドルの起動)
- 

**移動要求**

次の NC/PLC インタフェース信号は、軸が移動を要求しているか、または移動していることを PLC に通知します:

- 機械軸:
  - DB31、... DBX64.4 (移動要求「マイナス」)または
  - DB31、... DBX64.5 (移動要求「プラス」)
- ジオメトリ軸 1:
  - DB21、... DBX40.4 (ジオメトリ軸 1:移動要求「マイナス」)または
  - DB21、... DBX40.5 (ジオメトリ軸 1:移動要求「プラス」)
- ジオメトリ軸 2:
  - DB21、... DBX46.4 (ジオメトリ軸 2:移動要求「マイナス」)または
  - DB21、... DBX46.5 (ジオメトリ軸 2:移動要求「プラス」)
- ジオメトリ軸 3:
  - DB21、... DBX52.4 (ジオメトリ軸 3:移動要求「マイナス」)または
  - DB21、... DBX52.5 (ジオメトリ軸 3:移動要求「プラス」)
- 旋回軸 1;
  - DB21、... DBX332.4 (旋回軸 1:移動要求「マイナス」)または
  - DB21、... DBX332.5 (旋回軸 1:移動要求「プラス」)
- 旋回軸 2;
  - DB21、... DBX336.4 (旋回軸 2:移動要求「マイナス」)または
  - DB21、... DBX336.5 (旋回軸 2:移動要求「プラス」)
- 旋回軸 3;
  - DB21、... DBX340.4 (旋回軸 3:移動要求「マイナス」)または
  - DB21、... DBX340.5 (旋回軸 3:移動要求「プラス」)

---

**注記**

移動要求は、関連するすべての移動の合計になります。つまり、連結と訂正值の成分も考慮されます。

---

「移動要求」機能の操作の方法については、「移動要求 (ページ 205)」の章を参照してください。

**移動指令**

マシンデータ MD17900 \$MN\_VDI\_FUNCTION\_MASK (「パラメータ設定 (ページ 198)」の章を参照)の設定に応じて、移動要求が存在する場合には、すでに次のインタフェース信号が PC に出力されています。また、軸が移動するまではこの信号は出力されません。

- 機械軸:
  - DB31、... DBX64.6 (移動指令「マイナス」)または
  - DB31、... DBX64.7 (移動指令「プラス」)
- ジオメトリ軸 1:
  - DB21、... DBX40.6 (ジオメトリ軸 1:移動指令「マイナス」)または
  - DB21、... DBX40.7 (ジオメトリ軸 1:移動指令「プラス」)
- ジオメトリ軸 2:
  - DB21、... DBX46.6 (ジオメトリ軸 2:移動指令「マイナス」)または
  - DB21、... DBX46.7 (ジオメトリ軸 2:移動指令「プラス」)
- ジオメトリ軸 3:
  - DB21、... DBX52.6 (ジオメトリ軸 3:移動指令「マイナス」)または
  - DB21、... DBX52.7 (ジオメトリ軸 3:移動指令「プラス」)
- 旋回軸 1;
  - DB21、... DBX332.6 (旋回軸 1:移動指令「マイナス」)または
  - DB21、... DBX332.7 (旋回軸 1:移動指令「プラス」)
- 旋回軸 2;
  - DB21、... DBX336.6 (旋回軸 2:移動指令「マイナス」)または
  - DB21、... DBX336.7 (旋回軸 2:移動指令「プラス」)
- 旋回軸 3;
  - DB21、... DBX340.6 (旋回軸 3:移動指令「マイナス」)または
  - DB21、... DBX340.7 (旋回軸 3:移動指令「プラス」)

---

**注記**

移動する軸のコントローライネーブルまたはパルスイネーブルが欠落している場合、JOGモードに設定された移動指令がキャンセルされます。したがって、軸の解放後に移動指令を再度設定してください。

---

**手動パルス発生器の回転方向の反転**

手動パルス発生器の移動方向が想定される軸の移動方向に一致しない場合、手動パルス発生器の回転方向を反転できます。手動パルス発生器(HT2、HT8)を複数の軸に割り当てることができる場合、この補正は特に必要になります。

特定の MD の設定に加えて、手動パルス発生器の回転方向の反転は、特定の軸に対応する IS 「手動パルス発生器の回転方向の反転」 の設定により有効にできます。

- 機械軸:
  - DB31、... DBX7.0 (手動パルス発生器の回転方向の反転)
- ジオメトリ軸:
  - DB21、... DBX15.0 (ジオメトリ軸 1:手動パルス発生器の回転方向の反転)
  - DB21、... DBX19.0 (ジオメトリ軸 2:手動パルス発生器の回転方向の反転)
  - DB21、... DBX23.0 (ジオメトリ軸 3:手動パルス発生器の回転方向の反転)
- 旋回軸:
  - DB21、... DBX323.0 (旋回軸 1:手動パルス発生器の回転方向の反転)
  - DB21、... DBX327.0 (旋回軸 2:手動パルス発生器の回転方向の反転)
  - DB21、... DBX331.0 (旋回軸 3:手動パルス発生器の回転方向の反転)
- 輪郭ハンドル:
  - DB21、... .DBX31.5 (輪郭ハンドル:手動パルス発生器の回転方向の反転)

---

**注記**

反転信号は、PLC ユーザープログラム内で、手動パルス発生器の選択(IS 「手動パルス発生器の起動」 )と同時に設定してください。

---

手動パルス発生器の回転方向が NC によって反転された旨の通知は、以下のようにして各軸について IS「手動パルス発生器回転方向の反転が有効」を使用して設定されます。

- 機械軸:
  - DB31、... DBX67.0 (手動パルス発生器回転方向の反転が有効)
- ジオメトリ軸:
  - DB21、... DBX43.0 (ジオメトリ軸 1:手動パルス発生器回転方向の反転が有効)
  - DB21、... DBX49.0 (ジオメトリ軸 2:手動パルス発生器回転方向の反転が有効)
  - DB21、... DBX55.0 (ジオメトリ軸 3:手動パルス発生器回転方向の反転が有効)
- 旋回軸:
  - DB21、... DBX335.0 (旋回軸 1:手動パルス発生器回転方向の反転が有効)
  - DB21、... DBX339.0 (旋回軸 2:手動パルス発生器回転方向の反転が有効)
  - DB21、... DBX343.0 (旋回軸 3:手動パルス発生器回転方向の反転が有効)
- 輪郭ハンドル:
  - DB21、... DBX39.5 (輪郭ハンドル:手動パルス発生器回転方向の反転が有効)

#### 注記

反転信号の変更は、静止状態でのみ可能です。補間器によって移動指令値が出力されている間に変更が加えられた場合、当該の信号の変更は拒否され、さらにアラームが出力され、実加速度値を考慮して移動が停止します。

#### 移動動作の中止/中断

次の NC/PLC インタフェース信号は移動動作を中止します。指令値とフィードバック値の差が解消されます。

NC/PLC インタフェース信号	有効範囲	働き	
DB21、... DBX7.7 (NC リセット)	ジオメトリ軸/機械軸	中止	
DB21、... DBX12.0 - 2 (ジオメトリ軸 1:ハンドルの起動) DB21、... DBX16.0 - 2 (ジオメトリ軸 2:ハンドルの起動) DB21、... DBX20.0 - 2 (ジオメトリ軸 3:ハンドルの起動)	ジオメトリ軸	1 → 0	中止
DB21、... DBX6.2 (残移動距離削除)	ジオメトリ軸/機械軸	0 → 1	中止

## 4.6 ハンドルを使用した手動移動

NC/PLC インタフェース信号	有効範囲	働き	
DB31、... DBX2.2 (残移動距離削除/主軸リセット)	ジオメトリ軸/機械軸	0 → 1	中止
DB31、... DBX4.2 (ハンドルの起動)	機械軸	1 → 0	中止
DB31、... DBX1.3 (軸/主軸無効)	ジオメトリ軸/機械軸	0 → 1	中止
DB31、... DBX1.5 (位置検出器 1)	ジオメトリ軸/機械軸	0 → 1	中止
DB31、... DBX1.6 (位置検出器 2)	ジオメトリ軸/機械軸	0 → 1	中止

他の NC/PLC インタフェース信号(中断信号)について、ハンドルを使用した手動移動の影響(移動動作の中止または中断)を設定できます(「パラメータ設定 (ページ 198)」の章を参照してください)。

NC STOP は移動動作の中断のみを行います。指令値とフィードバック値の差はすべて維持されます。NC START を使用すると、それから残移動距離分を移動します。

## 下記も参照

監視機能 (ページ 275)

## 4.6.2 パラメータ設定

## 距離または速度の指定

手動パルス発生器によって、距離または速度のいずれかを入力できます。

- 距離の指定(初期設定)  
このハンドルで指令された距離を移動します。パルスが無くなることはありません。速度を最大許容値に制限すると、軸が行き過ぎる原因になります。
- 速度指定  
このハンドルは、移動速度のみを定義します。ハンドルを停止するとすぐに、軸も停止します。ハンドルから 1IPO サイクルでパルスが供給されない場合、動作がすぐに減速されるので、軸の行き過ぎが防止されます。ハンドルパルスにより、軌跡指令が出力されることはありません。

入力モードは、次のマシンデータで設定されます。

MD11346 \$MN\_HANDWH\_TRUE\_DISTANCE (手動パルス発生器の距離または速度の指定)

## 手動パルス発生器のパルス検出

各手動パルス発生器について 1 クリック位置毎に生成されるパルス数を指定してください。

MD11320 \$MN\_HANDWH\_IMP\_PER\_LATCH [<n>] (1 クリック位置毎の手動パルス発生器のパルス)

---

### 注記

負の値を入力すると、手動パルス発生器の回転方向が反転します。

---

## ステップ量

手動パルス発生器によるパルス毎の軸の移動ステップ量は、選択したステップ量によって決定されます。

また、5 つの固定ステップ量(初期設定:INC1、INC10、INC100、INC1000、および INC10000)、セッティング データによって設定できる可変ステップ量(INCvar)が使用可能です。

### 固定ステップ量

固定ステップ量のパラメータ設定は、NC 別の以下のマシンデータによっておこなわれます。

MD11330 \$MN\_JOG\_INCR\_SIZE\_TAB[1 ... 5] = <ステップ数 1 ... 5>

### 可変ステップ量

可変ステップ量のパラメータ設定は、NC 別の以下のマシンデータによっておこなわれます。

SD41010 \$SN\_JOG\_VAR\_INCR\_SIZE = <ステップ数>

## 1 ステップの距離の評価

固定と可変ステップ量の 1 ステップの距離評価は、以下の軸マシンデータによっておこなわれます。

MD31090 \$MA\_JOG\_INCR\_WEIGHT = <距離>

---

### 注記

負の値を入力すると、手動パルス発生器の回転方向が反転します。

---

4.6 ハンドルを使用した手動移動

ステップ量の制限

機械のオペレータは、以下のようにして選択したステップ量を制限できます。

- 機械軸で軸マシンデータを使用する場合:  
MD32080 \$MA\_HANDWH\_MAX\_INCR\_SIZE (選択したステップ値の制限)
- ジオメトリ軸でチャンネル別マシンデータを使用する場合:  
MD20620 \$MC\_HANDWH\_GEOAX\_MAX\_INCR\_SIZE (ジオメトリ軸に対する手動パルス発生器のステップ値の制限)
- 旋回軸でチャンネル別マシンデータを使用する場合:  
MD20621 \$MC\_HANDWH\_ORIAX\_MAX\_INCR\_SIZE (旋回軸に対する手動パルス発生器のステップ値の制限)

NC/PLC インタフェース信号での手動パルス発生器番号の表現

NC/PLC インタフェース信号の手動パルス発生器番号の表現は、以下のマシンデータを使用して定義します。

MD11324 \$MN\_HANDWH\_VDI\_REPRESENTATION

規格値	意味
0	ビットコード表現(初期設定) → 3 基の手動パルス発生器を表現できます。
1	バイナリコード表現 → 6 基の手動パルス発生器を表現できます。

NC/PLC インタフェース信号「移動指令プラス」 / 「移動指令マイナス」の出力

NC/PLC インタフェース信号「移動指令プラス」 / 「移動指令マイナス」の出力動作は、次のマシンデータにより指定します。

## MD17900 \$MN\_VDI\_FUNCTION\_MASK

規格値	意味
0	移動要求が有効である場合(初期設定)、NC/PLC インタフェース信号「移動指令プラス」/「移動指令マイナス」はすでに出力されています。
1	軸が実際に移動している場合(つまり、指令値がサーボで出力されている場合)にのみ、「移動指令プラス」/「移動指令マイナス」が NC/PLC インタフェース信号で出力されます。

## 速度

ジョグ時に有効な次の軸速度は、ハンドルを使用した手動移動にも使用されます。

- SD41110 \$SN\_JOG\_SET\_VELO (ジョグの軸速度)
- SD41130 \$SN\_JOG\_ROT\_AX\_SET\_VELO (ジョグモードの回転軸の軸速度)
- MD32020 \$MA\_JOG\_VELO (手動軸速度)

## 注記

送り速度が制限されているために、特に大きなパルス倍率に設定され、そのため、行き過ぎた場合には、ハンドル回転に同期して軸が追従できません。

## 加減速

手動移動の場合(「パラメータ割り付け(一般)(ページ 175)」を参照してください)。

## 反対方向への移動

(手動パルス発生器を反対方向に回転させることによる)移動方向の反転の動作は、次のマシンデータで設定できます。

## 4.6 ハンドルを使用した手動移動

MD11310 \$MN\_HANDWH\_REVERSE (ハンドル方向変更のしきい値)

規格値	意味
= 0	<p>ハンドルを反対方向に移動すると、その移動結果の距離が計算され、計算された終点に、できる限り高速でアプローチします。</p> <p>現在の移動方向で移動軸の減速点より前にこの終点がある場合、ユニットは減速した後、反対方向に移動して終点にアプローチします。それ以外の場合は、新たに計算された終点に直ぐにアプローチします。</p>
> 0	<p>少なくともマシンデータで指定されたパルス数分、反対方向にハンドルが移動する場合、軸はできる限り早く減速し、補間終了までの間に受け取ったすべてのパルスは無視されます。</p> <p>つまり、軸が停止状態(指令値側)に到達した後でなければ、反対方向の移動はおこなわれません。</p>

## ソフトウェアリミットスイッチの動作、ワーキングエリアリミット

軸がジョグモードで移動する場合、最初の有効リミット位置まで移動した後で、対応するアラームが出力されます。

下記のマシンデータの設定によります:

MD11310 \$MN\_HANDWH\_REVERSE (ハンドルの方向変更のしきい値)に応じて、応答は次のようになります(ただし軸が指令値側から終点にまだ到達していない場合)。

- ハンドルパルスから得られた距離により、以降の経路計算で使用される仮の終点が作成されます。  
たとえば、この仮の終点がリミット位置を **10 mm** 越えている場合、軸の再移動の前に、反対方向にこの **10 mm** の移動をおこなってください。リミット位置に達した直後に、反対方向への移動が直ちにおこなわれる場合、残移動距離削除またはハンドル割り当ての選択解除によって、仮の残移動距離を削除できます。
- リミット位置の背後にある終点までのハンドルパルスはすべて無視されます。ハンドルを反対方向に移動すると、直ちに反対方向(リミット位置とは逆方向)へ移動がおこなわれます。

## 送り速度の特性

ジョグモードでは、軸/主軸の送り速度の特性も次のセッティングデータによって決まります。

SD41100 \$SN\_JOG\_REV\_IS\_ACTIVE (JOG:毎回転/毎分送り)

SD41100 \$SN_JOG_REV_IS_ACTIVE	
有効	軸や主軸は、メイン主軸に応じて、毎回転送り速度 MD32050 \$MA_JOG_REV_VELO (ジョグの毎回転送り速度) または MD32040 \$MA_JOG_REV_VELO_RAPID (早送りオーバーライドによるジョグモードの毎回転送り速度) で常に移動されます。
無効	軸/主軸の特性は、次のセッティングデータによって決まります。 SD43300 \$SA_ASSIGN_FEED_PER_REV_SOURCE (位置決め軸/主軸の毎回転送り速度)
	回転が有効なフレームに対するジオメトリ軸の特性は、 チャンネルセッティングデータによって決まります。 SD42600 \$SC_JOG_FEED_PER_REV_SOURCE (ジョグの毎回転送り速度の制御)。

### NC/PLC インタフェース停止信号の動作

ハンドルを使用した手動移動での NC/PLC インタフェース停止信号の動作(移動動作の中止または中断)は、次のマシンデータによって設定できます。

MD20624 \$MC\_HANDWH\_CHAN\_STOP\_COND (ハンドル移動の特性の定義、チャンネル別)

MD32084 \$MA\_HANDWH\_STOP\_COND (ハンドルを使用した手動移動時の特性)

NC/PLC インタフェース 信号	有効範囲	MD20624 \$MC_HANDWH_CHAN_STOP_COND	
		Bit == 0	Bit == 1
DB11 DBX0.5 (モードグループ停止)	ジオメトリ軸/機械軸	NC スタートまで中断	中止
DB11 DBX0.6 (モードグループ停止、軸プラス主軸)	ジオメトリ軸/機械軸	NC スタートまで中断	中止
DB21、... DBX7.3 (NC ストップ)	ジオメトリ軸/機械軸	NC スタートまで中断	中止
DB21、... DBX7.4 (NC ストップ軸および主軸)	ジオメトリ軸/機械軸	NC スタートまで中断	中止

4.6 ハンドルを使用した手動移動

NC/PLC インタフェース 信号	有効範囲	MD20624 \$MC_HANDWH_CHAN_STOP_COND	
		Bit == 0	Bit == 1
DB21、... DBX6.0 (送り速度無効化)	ジオメトリ軸/機械軸	中断	中止
DB21、... DBB4 (送り速度オーバーライド) DB31、... DBX4.5 (早送りオーバーライド)	ジオメトリ軸	オーバーライド == 0:早送りオーバーライドが有効でない場合は中断します	オーバーライド == 0:早送りオーバーライドが有効でない場合は中止します
DB21、... DBB5 (早送りオーバーライド) DB31、... DBX4.5 (早送りオーバーライド)	ジオメトリ軸	早送りオーバーライド == 0:早送りオーバーライドが有効である場合は中断します	早送りオーバーライド == 0:早送りオーバーライドが有効な場合は中止します
DB21、... DBX12/16/20.3 (ジオメトリ軸 1/2/3:送り停止)	ジオメトリ軸	中断	中止

NC/PLC インタフェース 信号	有効範囲	MD32084 \$MC_HANDWH_CHAN_STOP_COND	
		Bit == 0	Bit == 1
DB31、... DBB0 (送り速度オーバーライド、軸)	主軸ではない機械軸	オーバーライド == 0:中断	オーバーライド == 0:中止
DB31、... DBB19 (主軸オーバーライド)	主軸	オーバーライド == 0:中断	オーバーライド == 0:中止
DB31、... DBX4.3 (送り停止/主軸停止)	ジオメトリ軸/機械軸	中断	中止
DB31、... DBX2.3 (クランプ中)	ジオメトリ軸/機械軸	動作なし	中止
DB31、... DBX2.1 (コントロールシステムイネーブル)	ジオメトリ軸/機械軸	中断	中止
DB31、... DBX21.7 (パルスイネーブル)	ジオメトリ軸/機械軸	中断	中止

**注記**

移動時に「コントローライネーブル」および「パルスイネーブル」を削除すると、軸の急停止およびリセットアラームにつながります。

**移動動作の中断**

停止命令が発行されると、残移動距離が保存され、手動パルス発生器のパルスが回収されます。停止状態がまだ適用されていない場合、結果距離が移動します。

**移動動作の中止**

停止命令が発行されると、残移動距離が削除され、停止状態が適用されなくなるまで、手動パルス発生器のパルスが無視されます(つまり、回収されません)。

**セーフオペレーティングストップ時の動作**

NC/PLC インターフェースのストップ信号と同様に、セーフオペレーティングストップ (SBH)を有効にしてストップが発生した場合の動作も設定できます。

ストップ条件	有効範囲	MD20624 \$MC_HANDWH_CHAN_STOP_COND	
		ビット 12 == 0	ビット 12 == 1 (初期設定)
有効なターゲット速度制限に達した場合のセーフオペレーティングストップ	機械軸	中断	中止

初期設定では、ビット 12 は「1」に設定されます。つまり、安全停止の有効化によって停止した場合、現在の移動はキャンセルされ、新規のハンドルパルスは無視されます。

補間停止の必要条件は、ターゲット速度制限が有効であることです(つまり、MD36933 \$MA\_SAFE\_DES\_VELO\_LIMIT > 0)。

**4.6.3 移動要求**

次の例は、「移動要求」NC/PLC インタフェース信号の操作方法を示しています。

4.6 ハンドルを使用した手動移動

例 1:距離入力による、ハンドルを使用した手動移動、停止状態が中止条件ではない

距離入力による、ハンドルを使用した手動移動  
 (MD11346 \$MN\_HANDWH\_TRUE\_DISTANCE == 1 または == 3)で、有効な停止状態が中止条件(MD32084 \$MA\_HANDWH\_STOP\_COND または MD20624 \$MC\_HANDWH\_CHAN\_STOP\_COND を参照してください)でない場合、NC/PLC インタフェース信号「移動要求」および「移動指令」の出力は、次の図に示す特性に対応します。

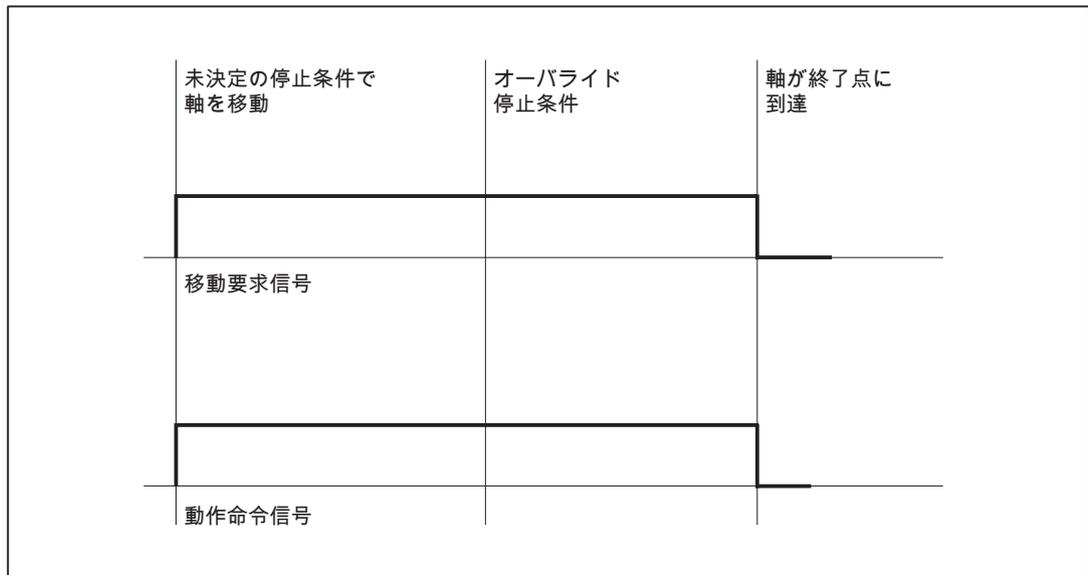


図 4-1 信号 - 時間の図:距離入力による、ハンドルを使用した手動移動で、停止状態が中止条件ではない場合 MD17900 \$MN\_VDI\_FUNCTION\_MASK ビット 0 = 0

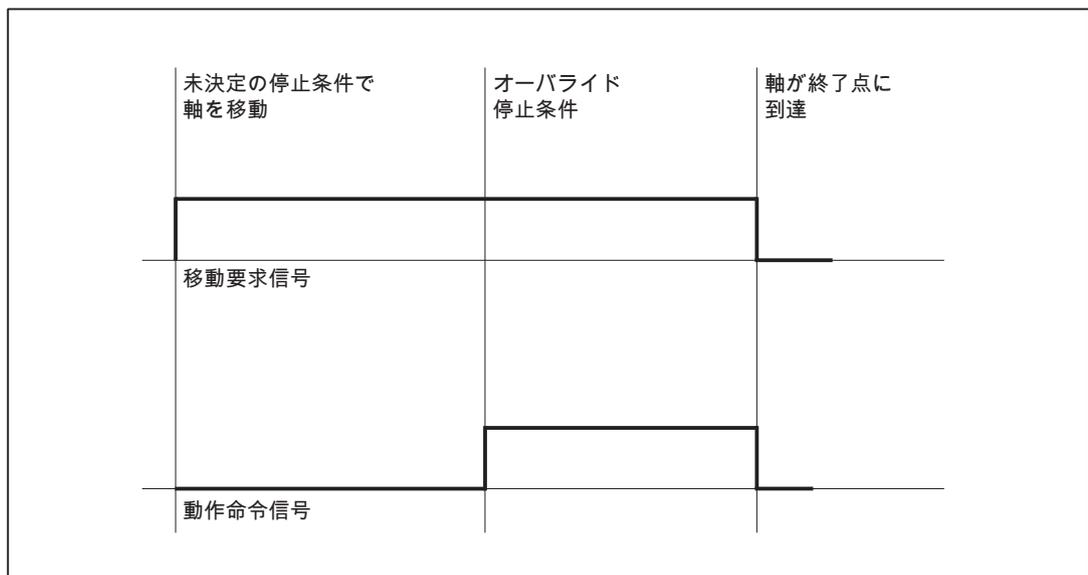


図 4-2 信号 - 時間の図:距離入力による、ハンドルを使用した手動移動で、停止状態が中止条件ではない場合 MD17900 \$MN\_VDI\_FUNCTION\_MASK ビット 0 = 1

### 例 2:ハンドルを使用した手動移動、停止状態が中止条件である

ハンドルを使用した手動移動中に、マシンデータ MD32084 \$MA\_HANDWH\_STOP\_COND または MD20624 \$MC\_HANDWH\_CHAN\_STOP\_COND により停止状態が中止条件として選択されている場合、移動指令は出力されませんが、対応する移動要求が出力されます。

停止状態が削除されている場合、現在中止しているため、対応する「移動要求」NC/PLC インタフェース信号がリセットされます。停止状態はもう有効ではありませんが、停止状態によって中止となったため、軸は移動できません。

また、いずれかの距離指定(MD11346 \$MN\_HANDWH\_TRUE\_DISTANCE == 1 または == 3)が有効になるか、またはハンドルが連続的に移動します。つまり、パルスが出力されます。

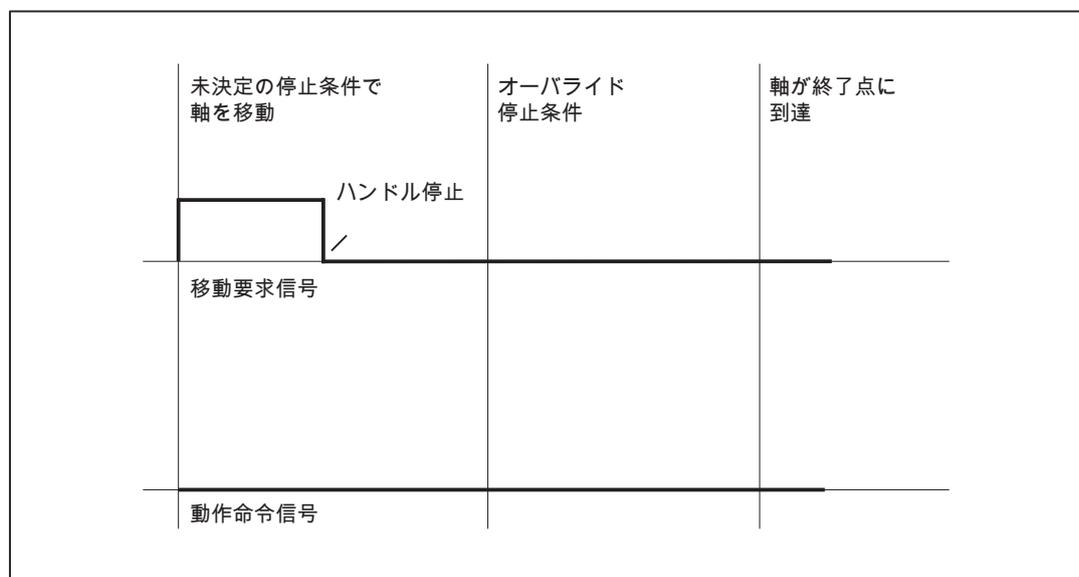


図 4-3 信号 - 時間の図:ハンドルを使用した手動移動、停止状態が中止条件である

ハンドル運転中に停止状態が有効になった場合、動作は中止され、「移動要求」および「移動指令」NC/PLC インタフェース信号がリセットされます。

### 例 3:速度指令値による、ハンドルを使用した手動移動、停止状態が中止条件である

ハンドルが速度指定(MD11346 \$MN\_HANDWH\_TRUE\_DISTANCE==0 または==2)によって移動なくなると、「移動要求」NC/PLC インタフェース信号がリセットされます。

4.7 主軸の手動による移動

ハンドルの選択が解除されると、「移動要求」PLC 信号もリセットされます。

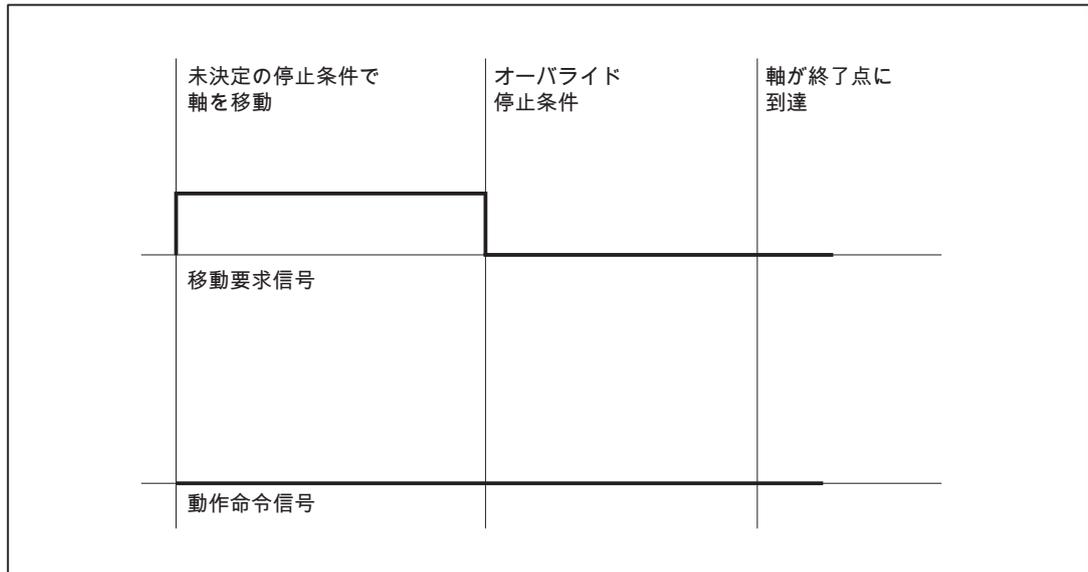


図 4-4 信号 - 時間の図:速度指令値による、ハンドルを使用した手動移動、停止状態が中止条件である

補足条件

NC ストップ

NC ストップ状態の場合は、そのために、移動指令と移動要求は出力されません。DRF 移動による例外が存在する場合:マシンデータ MD20624

\$MC\_HANDWH\_CHAN\_STOP\_COND (ビット 13==1)によって NC ストップステータスで DRF の移動が許可される場合、その特性は、ハンドルを使用した手動移動のステータスに対応します。

4.7 主軸の手動による移動

主軸モード

主軸の手動移動は、位置決めモード(主軸が位置制御中)、およびオープンループ制御モードの両方で使用可能です。

## 速度

主軸の手動による移動に適用される速度は、次の一般セッティングデータによって定義されます。

SD41200 \$SN\_JOG\_SPIND\_SET\_VELO (JOG モードでの主軸の速度)

SD41200 に「0」が入力されている場合、軸マシンデータが有効になります。

MD32020 \$\_MA\_JOG\_VELO (手動による軸速度)

主軸を手動で移動する場合、有効なギヤ選択の最高速度も考慮されます。

### 参照先:

『機能マニュアル、基本機能』; 「主軸(S1)」

## 主軸オーバーライド

主軸の手動による移動中の速度は、主軸オーバーライドスイッチで操作できます。

## 加速度

主軸は速度制御モードと位置制御モードで複数のギヤ選択を使用することが多いので、現在のギヤ選択に関連する加速度は常に、JOG モードの主軸に適用されます。

### 参照先:

『機能マニュアル、基本機能』; 「主軸(S1)」

## インタフェース信号

主軸を手動で移動する場合、NC と PLC 間のインタフェース信号は、機械軸の場合と同じ作用を持ちます。

主軸が位置制御中である場合、インタフェース信号 DB31、... DBX60.7 または DBX60.6 (汎用/精密イグザクトストップ範囲内の位置に到達)のみが設定されます。

### 主軸別のインターフェース信号

JOG モードでは、主軸に対して次のインタフェース信号は無効です。

- DB31、... DBX17.6 (反転 M3/M4)
- DB31、... DBX18.6 - 7 (揺動回転方向、右回り/左回り)
- DB31、... DBX18.5 (揺動イネーブル)
- DB31、... DBX16.7 (S 値削除)

## 4.8 ジオメトリ軸/旋回軸の手動による移動

JOG モードでは、下記の主軸からのインタフェース信号は設定されません。

- DB31、... DBX83.7 (右回りの実回転方向)
- DB31、... DBX83.5 (主軸の設定目標範囲)

## 4.8 ジオメトリ軸/旋回軸の手動による移動

### 注記

#### 座標変換パッケージ、処理

JOG モードでは、SINUMERIK 840D sl の「座標変換パッケージの処理」を使用して、複数の有効な基準座標系でジオメトリ軸の座標変換をそれぞれに個別に設定できます。

#### 参照先:

機能マニュアル、応用機能; TE4:座標変換パッケージ、処理

「直交座標ジョグ移動(オプション) (ページ 590)」の章も参照してください。

## 同時移動

移動キーを使用して、同時に 1 つのジオメトリ軸のみを連続的またはステップで移動できます。複数のジオメトリ軸を移動しようとする、アラーム 20062 「軸は既に動作中です」が出力されます。ただし、ハンドル 1 から 3 を使って、3 つのジオメトリ軸を同時に移動できます。

## 径方向軸

ジオメトリ軸を径方向軸として定義し(MD20100 \$MC\_DIAMETER\_AX\_DEF を参照)、半径指定を選択している場合、以下が適用されます。

- 連続移動:  
径方向軸が連続的に移動する場合、違いはありません。
- ステップ移動:  
選択したステップ量の半分の距離のみを移動します。たとえば INC10 では、移動キーを押している間、軸が 5 ステップ単位だけ移動します。
- ハンドルによる移動:  
ステップ移動では、ハンドルによるハンドルパルス毎に半分の距離だけ移動します。

#### 参照先:

『機能マニュアル、基本機能』; 「径方向軸(P1)」

## PLC インタフェース

ジオメトリ軸/旋回軸では、軸別 PLC インタフェースと同じ信号を含む個別の PLC インタフェースが存在します。

- ジオメトリ軸:  
DB21、... DBB12 - 23 および  
DB21、... DBB40 - 56
- 旋回軸:  
DB21、... DBB320 - 331 および  
DB21、... DBB332 - 343

## 速度

- ジオメトリ軸  
ジオメトリ軸の手動による移動に適用される速度は、次のチャンネル別マシンデータによって定義されます。  
MD21165 \$MC\_JOG\_VELO\_GEO (ジオメトリ軸の手動速度)
- 旋回軸  
旋回軸の手動による移動に適用される速度は、次のチャンネル別マシンデータによって定義されます。  
MD21155 \$MC\_JOG\_VELO\_ORI (旋回軸の手動速度)

## 早送りオーバーライド

また、早送りオーバーライドキーと一緒にキー/手動ホイールを作動させた場合、移動は設定された早送り速度で行われます。

- ジオメトリ軸
  - MD21160 \$MC\_JOG\_VELO\_RAPID\_GEO (ジオメトリ軸の手動早送り)
- 旋回軸
  - MD21150 \$MC\_JOG\_VELO\_RAPID\_ORI (旋回軸の手動早送り)

## 送り速度オーバーライド

ジオメトリ軸を手動で移動している場合は、チャンネル別の送り速度オーバーライドスイッチが有効です。

## 4.8 ジオメトリ軸/旋回軸の手動による移動

### 加速度と加々速度

ジオメトリ軸/旋回軸の手動移動では、指定のチャンネルについて加速度と加々速度を制限できます。これにより、回転軸によって完全な直交移動を生成するキネマティックスをより適切に処理できます(ロボット)。

#### ジオメトリ軸

次のマシンデータによって、各チャンネルについて、手動でジオメトリ軸を移動時の最大加速度を指定できます。

**MD21166 \$MC\_JOG\_ACCEL\_GEO [<ジオメトリ軸>]**

<ジオメトリ軸> = 0、1、2

次のマシンデータによって、各チャンネルについて、SOFT 加速度モード(加々速度一定加速度) で、手動でジオメトリ軸を移動する時の最大加々速度を指定できます。

**MD21168 \$MC\_JOG\_JERK\_GEO [<ジオメトリ軸>]**

<ジオメトリ軸> = 0、1、2

#### 旋回軸

次のマシンデータによって、各チャンネルについて、手動で旋回軸を移動時の最大加々速度を指定できます。

**MD21158 \$MC\_JOG\_JERK\_ORI [<旋回軸>]**

MD21158 を有効にするには、次のマシンデータによって、旋回軸の手動移動のチャンネル別加々速度制限を有効にしてください。

**MD21159 \$MC\_JOG\_JERK\_ORI\_ENABLE == TRUE**

### アラーム

#### アラーム 20060

ジオメトリ軸として軸が定義されていない場合に、JOG モードでジオメトリ軸として軸を手動で移動しようとする、アラーム 20060 「ジオメトリ軸として軸を移動できません」が出力されます。

### アラーム 20062

次の条件でジオメトリ軸/旋回軸の手動移動がおこなわれた場合、アラーム 20062, 「軸はすでに有効です」が発生します。

- この軸は既に、軸の PLC インタフェースによって JOG モードで移動中です。
- 回転座標系のフレームが既に有効で、この座標系の別のジオメトリ軸が移動キーによって JOG モードで移動中です。

## 4.9 JOG モードの固有位置へのアプローチ

### 4.9.1 機能

機械ユーザーは、機械操作パネルの移動キーの起動によって、マシンデータによって定義された軸位置にアプローチするために、「ジョグモードの固有位置へのアプローチ」機能を使用できます。移動軸は、定義された固有位置に達すると自動的に停止します。

---

#### 注記

##### G75 による固定点へのアプローチ

定義済み固定点へのアプローチは、G75 命令を使用して、パートプログラムから有効にできます。

G75 による固定点へのアプローチの詳細については、

##### 参照先:

:プログラミングマニュアル、基本編の次の章を参照してください。「追加命令」 > 「固定点へアプローチ(G75)」の章

---

### 用途

たとえば、代表的な用途は次のとおりです。

- NC プログラムの起動前の基本位置へのアプローチ。
- 工具交換位置、ロード位置、およびパレット変更位置へのアプローチ。

---

## 4.9 JOG モードの固有位置へのアプローチ

### 必要条件

- 「ジョグモードの固有位置へのアプローチ」は、ジョグモードだけ有効にできます。機械機能 JOG-REPOS または JOG-REF が有効で、自動の JOG である場合、この機能は有効にできません。
- 移動する軸を原点確立してください。
- キネマティックトランスフォーメーションを有効にすることはできません。
- 動作中の連結追従軸を移動軸にすることはできません。

### 処理

「JOG モードで固定点へアプローチ」の手順

1. JOG モードの選択
2. 「JOG モードで固定点へアプローチ」機能の設定
3. 移動キーまたはハンドルによる機械軸の移動

### 起動

「JOG モードで固定点へアプローチ」機能を選択した後 PLC は、アプローチする固定点の数をバイナリコードで次のビットを使用して NC に出力します。

DB31、... DBX13.0 - 2 (JOG モードで固定点へアプローチ)

NC は、この機能を有効にすると、すぐに次のインタフェース信号で起動結果を確認します。

DB31、... DBX75.0 - 2 (JOG モードで固定点へアプローチが有効)

---

### 注記

次の場合、この機能の適用はできません。

- NC リセット中
- 緊急に非常停止を行う場合
- ASUP の処理中

アラームメッセージは発生しません。終了後、または動作中の機能の確認応答後に、起動遅延が行われます。

---

### 手順

実移動は、移動キーまたはハンドルによって固定点へのアプローチ方向に開始されます。

選択した機械軸は、固定点で自動的に停止するまで移動します。

固定点に達すると、対応する NC/PLC インタフェース信号と「精密イグザクトストップ」:

#### DB31、... DBX75.3 - 5 (JOG モードで固定点へアプローチの到達)

またこの表示信号は、軸が指令値側で他の方法(例: NC プログラム、FC18 (840 sl の場合)、シンクロナイズドアクションなど)によって機械座標系の固定点位置に到達し、「精密イグザクトストップ」許容範囲(MD36010 \$MA\_STOP\_LIMIT\_FINE)内のフィードバック値側で停止した場合にも通知されます。

## 反対方向への移動

(アプローチ先の固定点に対して)反対方向への移動中の動作は、次のマシンデータのビット 2 の設定によって決まります。

#### MD10735 \$MN\_JOG\_MODE\_MASK (JOG モードの設定)

このビットを設定した場合にのみ、反対方向への移動が可能です。

このビットが設定されていない場合、反対方向への移動が禁止され、移動キーまたはハンドルによってアプローチ先の固定点に対して反対方向に移動しようとする、次のチャネル状態メッセージが出力されます。

「JOG:方向は禁止されました<軸>」

## 他の固定点へのアプローチ

固定点への移動中に別の固定点を選択した場合、軸移動がキャンセルされ、次のアラームが出力されます。

アラーム 17812 「JOG モードのチャネル%1 軸%2 の固定点アプローチ:固定点が変更されました」

メッセージ信号 DB31、... DBX75.0 - 2 (ジョグモードで固定点へアプローチが有効)では、新たに選択した固定点の番号が示されます。移動を継続するためにジョグ移動を再起動してください。

---

### 注記

このアラームメッセージを回避するには、機械ユーザは次の実施手順に従ってください。

1. 残移動距離の削除によって現在の移動動作をキャンセルします。
  2. 別の固定点に対して固定点へのアプローチを有効にし、軸の停止後に運転を開始します。
-

## 4.9 JOG モードの固有位置へのアプローチ

### 固定点からの引き戻し、または解除

固定点から引き戻すためには、「JOG モードで固定点へアプローチ」機能を無効にしてください。この作業は、起動信号を「0」にリセットすることによって行います。

DB31、... DBX13.0 - 2 = 0

メッセージ信号 DB31、... DBX75.0 - 2 (ジョグモードで固定点へアプローチが有効)および DB31、... DBX75.3 - 5 (ジョグモードで固定点へアプローチ位置到達)は、固定点から離れると解除されます。

### 特殊なケース:軸が既に、固定点上にあるとき。

固定点移動の開始時に、アプローチ先の固定点に軸が既にある場合、軸は移動できません。この場合、次のチャンネル状態メッセージによってその内容が表示されます。

「JOG:位置決め完了<軸>」

固定点から引き戻すには、「JOG モードで固定点へアプローチ」機能を無効にしてください。

### ステップ移動の特記事項

ステップ移動中にステップ値が完了する前に固定点に達した場合、ステップ値は完了したものとみなされます。これは、ステップ単位だけをすべて移動した場合にも当てはまります。

MD11346 \$MN\_HANDWH\_TRUE\_DISTANCE = 2 または 3

### モジュロ回転軸の特記事項

モジュロ回転軸では、両方向から固定点にアプローチできます。最短軌跡(DC)は、移動中に監視されません。

### 主軸の機能

主軸は、「JOG モードで固定点へアプローチ」機能を起動すると、位置決めモードに変わります。閉ループ位置制御が有効であり、軸が固定点に移動できます。

原点パルスが検知されない場合、軸運転で次のアラームメッセージが出力されます。

アラーム 17810 「チャンネル%1 軸%2 が原点確立されていません」

主軸はまた常にモジュロ回転軸であることが必要であるため、方向監視についてもモジュロ回転軸と同じ条件が適用されます(「モジュロ回転軸の機能」の節を参照してください)。

## 4.9.2 パラメータ設定

### 反対方向への移動

反対方向(つまり、アプローチ先の固定点の反対方向)への移動時の特性は、次の設定によって決まります。

MD10735 \$MN\_JOG\_MODE\_MASK、ビット 2 (ジョグモードの設定)

ビット	値	意味
2	0	反対方向へ移動 <b>できません</b> (基本設定)。
	1	反対方向へ移動 <b>できます</b> 。

### 固定点位置

1 つの軸に対して最大で 4 つの固定点位置を定義できます。

MD30600 \$MA\_FIX\_POINT\_POS[0...3] = <固定点位置 1...4>

### 有効な固定点位置の数

MD30600 \$MA\_FIX\_POINT\_POS に入力される実際に有効な固定点位置の数は、以下を使用して定義できます。

MD30610 \$MA\_NUM\_FIX\_POINT\_POS = <有効な固定点位置の数>

---

**注記**

**例外:G75**

互換性の理由から、G75 では次のパラメータ設定も可能です。

MD30610 \$MA\_NUM\_FIX\_POINT\_POS = 0 (有効な固定点位置なし)

MD30600 \$MA\_FIX\_POINT\_POS[0]と[1]は、2つの有効な固定点位置が存在することが前提になります。

固定点位置 1 と 2 は、NC/PLC インターフェース経由で有効にできます。ただし、そのアプローチは G75 によってのみ可能です。

---

## 軸の応答性

「ジョグモードで固定点へアプローチ」の軸別加速度と軸別加々速度は、次のマシンデータによって決定します。

- **移動キーまたは手動パルス発生器**により移動する場合:
  - MD32300 \$MA\_MAX\_AX\_ACCEL [0] (DYNORM ダイナミック応答モード DYNORM での軌跡移動に対する最大軸別加速度)
  - MD32430 \$MA\_JOG\_AND\_POS\_MAX\_JERK (軸別の最大軸加々速度)

**注記**

機械軸の移動については、単独軸移動の軸別加々速度の制限が有効である場合にのみ、MD32430 \$MA\_JOG\_AND\_POS\_MAX\_JERK は有効です。

MD32420 \$MA\_JOG\_AND\_POS\_JERK\_ENABLE [<軸>] == TRUE

- **G75 パートプログラム命令**により移動する場合:
  - MD32300 \$MA\_MAX\_AX\_ACCEL [0/1] (ダイナミック応答モード DYNORM/DYNPOS での軌跡移動に対する最大軸別加速度)

**注記**

位置決め軸の応答性(DYNORM または DYNPOS)のタイプは、次のマシンデータによって定義されます。

MD18960 \$MN\_POS\_DYN\_MODE = <モード>

- MD32431 \$MA\_MAX\_AX\_JERK [0] (ダイナミック応答モード DYNORM の軌跡移動の最大軸別加々速度)

**注記**

機械軸の移動については、単独軸移動の軸別加々速度の制限が有効である場合にのみ、MD32431 \$MA\_MAX\_AX\_JERK は有効です。

MD32420 \$MA\_JOG\_AND\_POS\_JERK\_ENABLE [<軸>] == TRUE

**参照:**

『機能マニュアル、基本機能』; 「加速度制御(B2)」

## 4.9 JOG モードの固有位置へのアプローチ

## 4.9.3 プログラミング

## システム変数

パートプログラムおよびシンクロナイズドアクションで、「固有位置へアプローチ」機能関連の次のシステム変数を読み取ることができます。

システム変数	説明
\$AA_FIX_POINT_SELECTED [<軸>]	アプローチする固有位置の番号
\$AA_FIX_POINT_ACT [<軸>]	現在、軸が位置している固有位置の番号

## 4.9.4 補足条件

## 軸は割り出し軸です

移動する軸が割り出し軸であり、アプローチ先の固有位置が割り出し位置に一致しない場合、軸が移動せず、アラームが出力されます。

## 有効フレーム

すべての有効フレームが無視されます。そして機械座標系で移動が実行されます。

## 有効オフセット値

有効オフセット値(DRF、外部ゼロオフセット、シンクロナイズドアクションオフセット \$AA\_OFF、オンライン工具オフセット)も移動します。固有位置は、機械座標システムの位置です。

JOG モードの固有位置アプローチ時にオフセット移動(DRF、外部ゼロオフセット、シンクロナイズドアクションオフセット \$AA\_OFF、オンライン工具オフセット)が行われた場合、アラームが発行されます。この場合、機械座標系で、アプローチ先の固有位置に到達しません。その代わりに、有効オフセットの移動なしの位置に到達しています。固有位置に対応する NC/PLC インタフェース信号 DB31、... DBX75.3-5 は出力されません。

## ワーキングエリアリミット

(BCS および WCS の)ワーキングエリアリミットが考慮され、軸移動がこのリミット位置に到達すると停止します。

### 4.9.5 用途例

#### 目的

「JOG モードの固有位置へのアプローチ」機能で、回転軸(機械軸 4[AX4])を固有位置 2(90 度)に移動します。

#### パラメータの割り付け

機械軸 4 の「固有位置へのアプローチ」機能のマシンデータは、次のようにパラメータ設定します。

MD30610 \$MA_NUM_FIX_POINT_POS[AX4] = 4	機械軸 4 に対して 4 つの固有位置を定義します。
MD30600 \$MA_FIX_POINT_POS[0,AX4] = 0	1. 番目の AX4 = 0 度の固有位置
MD30600 \$MA_FIX_POINT_POS[1,AX4] = 90	2. 番目の AX4 = 90 度の固有位置
MD30600 \$MA_FIX_POINT_POS[2,AX4] = 180	3. 番目の AX4 = 180 度の固有位置
MD30600 \$MA_FIX_POINT_POS[3,AX4] = 270	4. 番目の AX4 = 270 度の固有位置

#### 初期状態

機械軸 4 は原点が確立され、0 度の位置にあります。これは第 1 の固有位置に対応し、NC/PLC インタフェース信号によって出力されます。

DB31 DBX75.0 = 1(ビット 0 - 2 = 1)

#### 固有位置 2 へのアプローチ

制御をジョグモードに切り換えます。

次の NC/PLC インタフェース信号によって、「固有位置へのアプローチ」機能が固有位置 2 上で有効になります。

DB31 DBX13.1 = 1(ビット 0 - 2 = 2)

起動の結果は、次の NC/PLC インタフェース信号によって確認されます。

## 4.10 JOG での位置移動

DB31 DB75.1 = 1(ビット 0 - 2 = 2)

固有位置 2 へアプローチする連続的な移動を行うために、機械操作パネルのプラス移動キーを使用します。

機械軸 4 は、90 度の位置で停止します。これは、NC/PLC インタフェース信号によって通知されます。

DB31 DBX75.4 = 1(ビット 3 - 5 = 2)

## 4.10 JOG での位置移動

### 4.10.1 機能

「JOG での位置移動」機能によって、機械のオペレータがセッティングデータを使って、移動キーまたはハンドルを使って選択した機械軸を移動できる機械座標系の位置を指定できます。

### 用途

手動のみで操作する工作機械でこの機能を使って、たとえばドリル穴に精密なインクリメントで移動します。

### 必要条件

- 「JOG での位置移動」機能が、JOG モードのみで有効にできること。  
機械機能 JOG-REPOS または JOG-REF が有効で、自動の JOG である場合、この機能は有効にできません。
- 移動する軸を原点確立してください。
- キネマティックトランスフォーメーションが有効になっていない。
- 動作中の連結追従軸を移動軸にすることはできません。

### 処理

JOG での位置移動の手順:

1. JOG モードの選択
2. 移動先の位置の指定

3. 「JOG での位置移動」機能の有効化
4. 移動キーまたはハンドルによる機械軸の移動

### 移動先の位置の指定

移動先の位置は、以下のセッティングデータに入力します。

SD43320 \$SA\_JOG\_POSITION (JOG 位置)

### 機能の有効化

「JOG での位置移動」機能を選択後、PLC によりインタフェース信号が設定されます。

DB31、... DBX13.3 (JOG で位置に移動)

NC は、この機能を有効にするとすぐに、次のインタフェース信号で起動結果を確認します。

DB31、... DBX75.6 (JOG JOG 位置移動が有効)

---

#### 注記

次の場合、この機能の適用はできません。

- NC リセット中
- 緊急に非常停止を行う場合
- ASUP の処理中
- 「固定位置に JOG で移動」機能がすでに有効な場合

アラームメッセージは発生しません。終了後、または動作中の機能の確認応答後に、起動遅延が行われます。

---

### 手順

位置への移動は、移動キーまたはハンドルで開始します。

軸は、移動先の位置で自動停止するまで移動します。

「精密イグザクトストップ」で位置に達した場合、以下の NC/PLC インターフェース信号が出力されます。

DB31、... DBX75.7 (JOG 位置決め完了)

### 異なる位置への移動

位置への移動中にセッティングデータ SD43320 の内容を変更した場合、軸の移動はキャンセルされ、次のアラームが出力されます。

#### 4.10 JOG での位置移動

アラーム 17822 「チャンネル%1 軸%2 が JOG で位置に移動:位置が変更されました」  
新規の位置に移動するために JOG 移動を再起動してください。

##### 注記

このアラームメッセージを回避するには、機械ユーザは次の実施手順に従ってください。

1. 残移動距離の削除によって現在の移動動作をキャンセルします。
2. 軸の停止後に、SD43320 に新規の位置を入力します。
3. JOG 移動を再起動します。

#### 位置からの後退/無効化

位置から後退するには、「JOG での位置移動」機能を無効にしてください。この作業は、起動信号を「0」にリセットすることによって行います。

DB31、... DBX13.3 = 0

この無効化によって、メッセージ信号 DB31、... DBX75.6 および DB31、... DBX75.7 もリセットされます。

#### 反対方向への後退

位置からの反対方向への後退の動作は、マシンデータのビット 2 の設定に依存します。

MD10735 \$MN\_JOG\_MODE\_MASK (JOG モードの設定)

このビットを設定した場合にのみ、反対方向への後退が可能です。

このビットが設定されていない場合、反対方向への後退が禁止され、移動キーまたはハンドルによってアプローチ先の固定点に対して反対方向に移動しようとする、次のチャンネル状態メッセージが出力されます。

「JOG:方向は禁止されました<軸>」

#### 特殊なケース:軸がすでに位置決めされている場合

JOG 移動を開始する時点で、軸がすでに移動先の位置にある場合、軸は移動できません。この場合、次のチャンネル状態メッセージによってその内容が表示されます。

「JOG:位置決め完了<軸>」

後退するには、「JOG での位置移動」機能を無効にしてください。

### ステップ移動の特記事項

ステップ移動中にステップ値が完了する前に位置に達した場合、ステップ値は完了したものとみなされます。これは、ステップ単位だけをすべて移動した場合にも当てはまります。

MD11346 \$MN\_HANDWH\_TRUE\_DISTANCE = 2 または 3

### モジュロ回転軸の特記事項

モジュロ回転軸では、両方向から位置にアプローチできます。最短軌跡(DC)は、移動中に監視されません。

### 主軸の機能

主軸は、「JOG での位置移動」機能を起動すると、位置決めモードに変わります。閉ループ位置制御が有効であり、軸が位置に移動できます。

原点パルスが検知されない場合、軸運転で次のアラームメッセージが出力されます。

アラーム 17810 「チャンネル%1 軸%2 が原点確立されていません」

主軸はまた常にモジュロ回転軸であることが必要であるため、方向監視についてもモジュロ回転軸と同じ条件が適用されます(「モジュロ回転軸の機能」の節を参照してください)。

### 下記も参照

パラメータ設定 (ページ 225)

## 4.10.2 パラメータ設定

### 軸境界線と工具長補正の考慮

軸境界線、および軸がジオメトリ軸として設定されている場合はと工具長補正の考慮は、次のマシンデータのビット 1 の設定によって異なります。

4.10 JOG での位置移動

MD10735 \$MN\_JOG\_MODE\_MASK (JOG モードの設定)

ビット	値	意味
1	0	軸境界線と工具長補正は考慮されません。
	1	軸境界線と工具長補正が考慮されます。

反対方向への後退

位置からの反対方向への後退の動作は、マシンデータのビット 2 の設定に依存します。

MD10735 \$MN\_JOG\_MODE\_MASK (JOG モードの設定)

ビット	値	意味
2	0	反対方向へ後退できません(初期設定)。
	1	反対方向へ後退できます。

移動先の位置

移動先の位置は、以下のセッティングデータに入力します。

SD43320 \$SA\_JOG\_POSITION (JOG 位置)

4.10.3 補足条件

軸は割り出し軸です

移動する軸が割り出し軸であり、アプローチ先の位置が割り出し位置に一致しない場合、軸が移動せず、アラームが出力されます。

有効フレーム

すべての動作中のフレームが無視されます。そして機械座標系で移動が行われます。

## 訂正值が有効

有効訂正值(DRF、外部ゼロオフセット、シンクロナイズドアクションオフセット \$AA\_OFF、オンライン工具オフセット)も移動します。

JOG モードの可変位置アプローチ時にオフセット移動(DRF、外部ゼロオフセット、シンクロナイズドアクションオフセット \$AA\_OFF、オンライン工具オフセット)が行われた場合、アラームが出力されます。この場合、機械座標系ではアプローチ先の位置に到達しません。その代わりに、有効オフセットの移動なしの位置に達しています。NC/PLC インターフェース信号 DB31、... DBX75.7 は報告されません。

## 作業領域リミット

(BCS および WCS の)ワーキングエリアリミットが考慮され、軸移動がこのリミット位置に達すると停止します。

### 4.10.4 用途例

#### 目的

「JOG での位置移動」機能で、回転軸(機械軸 4[AX4])を 100 度位置に移動します。

#### パラメータの割り付け

移動先の位置は、ユーザーインターフェースで以下のように指定します。

```
SD43320 $SA_JOG_POSITION[AX4] = ;機械軸 4 の移動先の位置= 100  
100
```

#### 初期状態

軸 4 (AX4)は原点確立して、位置 0 度にあります。

#### 位置への移動

コントロールが JOG モードに切り替えられ、機械軸 4 が選択されます。

「JOG での位置移動」機能を有効にします。

```
DB34 DBX13.3 = 1
```

## 4.11 JOG での円移動

NC は、この機能を有効にすると、すぐに起動結果を確認します。

DB34 DBX75.6 = 1

連続移動および機械操作パネルの JOG 移動プラスキーで位置への移動が開始されます。

軸は、移動先の 100 度位置で自動停止するまで移動します。

「精密イグザクトストップ」で位置に達した場合、以下の NC/PLC インターフェース信号が設定されます。

DB34 DB75.7 = 1

## 4.11 JOG での円移動

### 4.11.1 機能

「JOG での円移動」機能を使って、機械のオペレータは移動キーまたはハンドルを使って、有効平面の 2 つのジオメトリ軸を同時に円弧に沿って移動できます。

#### 用途

この機能は、手動操作のみの工作機械に使用します。

#### 必要条件

- 「JOG での円移動」機能が、JOG モードのみで有効にできること。  
機械機能 JOG-REPOS および JOG-REF が有効である場合、この機能は有効にできません。
- 移動に関連する軸は、原点確立済みであることが必要です。
- 有効平面は傾斜してはなりません。

#### 処理

JOG での円移動の手順:

1. 平面の選択
2. JOG モードの選択
3. 円パラメータの指定

4. 「JOG での円移動」機能の有効化
5. 移動キーまたはハンドルによるジオメトリ軸の移動

## 平面の選択

有効な平面のジオメトリ軸は、以下の G 命令のいずれかをプログラミングして選択します。

G 命令	平面の 1 番目のジオメトリ軸	平面の 2 番目のジオメトリ軸
G17	X	Y
G18	Z	X
G19	Y	Z

## 円パラメータの指定

JOG での円移動には、少なくとも次のデータが必要です。

- 円の中心点の座標
- 内径機械加工の円の最大半径、または外径加工の円の最小半径
- 加工モード(内径または外径加工)

指定は、ユーザーインターフェースでチャンネルセッティングデータを使って行います。

SD42690 \$SC\_JOG\_CIRCLE\_CENTRE (円の中心点)

SD42691 \$SC\_JOG\_CIRCLE\_RADIUS (円の半径)

SD42692 \$SC\_JOG\_CIRCLE\_MODE (JOG 円モード)

詳細については、「パラメータ設定 (ページ 234)」章を参照してください。

## 機能の有効化

「JOG での円移動」機能を選択後、PLC によりインターフェース信号が設定されます。

DB21、... DBX30.6 (ジョグ円移動)

NC は、この機能を有効にすると、すぐに次のインターフェース信号で起動結果を確認します。

4.11 JOG での円移動

---

DB21、... DBX377.6 (ジョグ円移動が有効)

---

**注記**

次の場合、この機能の適用はできません。

- NC リセット中
- 緊急に非常停止を行う場合
- ASUP の処理中
- 1つ以上の関連する軸がすでに移動している場合
- PTP 移動が有効の場合
- 一般の座標変換を使って有効な直交を手動で移動している場合

アラームメッセージは発生しません。終了後、または動作中の機能の確認応答後に、起動遅延が行われます。

---

**手順**

機能を有効にした後、連続またはインクリメンタル JOG 移動により、移動キーまたはハンドルを使って工具を移動できます。

- 有効な平面の 1 番目のジオメトリ軸の JOG 移動要求では、現在の位置からの移動は、定義された円弧と同じ中心点を持つ同心円弧を使って実行されます。
- 有効な平面の 2 番目のジオメトリ軸の JOG 移動要求では、移動は定義された円弧の直交方向に行われます。その結果、円の半径が変わります。
- 有効な平面では、1 回に 1 つの移動のみが実行できます。これは、円弧上または円弧の直交方向に行われます。
- 有効な平面上の移動に関係なく、3 番目のジオメトリ軸は有効な平面に直交するよう移動できます。

---

**注記**

半径の縮小は、ゼロ(つまり、円の中心点)のみ可能です。その後、半径を再度拡大した場合、縮小が開始された点に戻る方向への移動が発生します。

---

**注記**

この機能が有効になった時点、または円の中心点に変更された後、現在位置が円の中心点である場合、以下が適用されます。

- 一周円の場合、2 番目のジオメトリ軸の方向に移動します。
  - 円のセグメントの場合、角の 2 等分線の方向に移動します。
-

## 内径または外径加工

セッティングデータ SD42692 \$SC\_JOG\_CIRCLE\_MODE での指定に応じて、内径または外径加工が行われます。

- 内径加工
  - 移動は、定義された円の**内側**のみで可能です。
  - SD42691 \$SA\_CIRCLE\_RADIUS の円の半径は、**最大半径**です。
- 外側加工
  - 移動は、定義された円の**外側**のみで可能です。
  - SD42691 \$SA\_CIRCLE\_RADIUS の円の半径は、**最小半径**です。

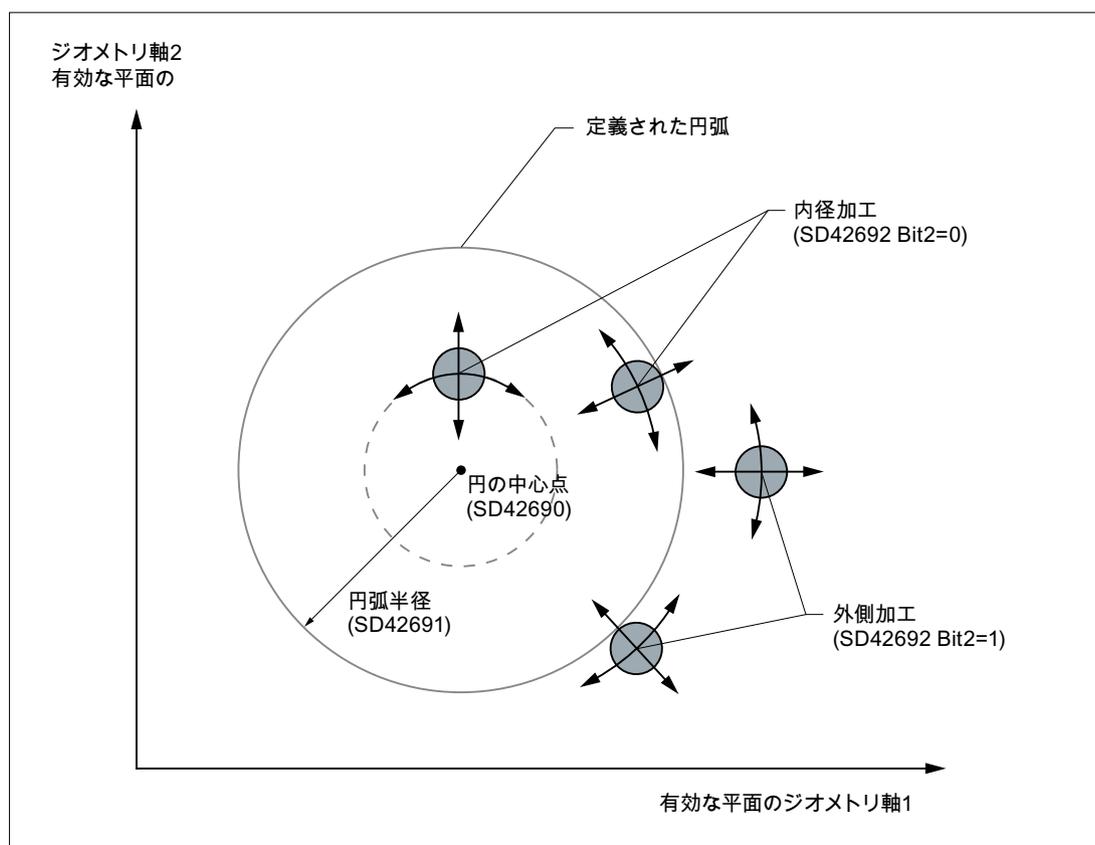


図 4-5 JOG での円移動:内径/外径加工

## 円セグメントの加工

開始および終了角度を指定することで、1 番目および 2 番目のジオメトリ軸の作業領域を円セグメントに制限することができます。

SD42693 \$SC\_JOG\_CIRCLE\_START\_ANGLE (円の開始角度)

4.11 JOG での円移動

SD42694 \$SC\_JOG\_CIRCLE\_END\_ANGLE (円の終了角度)

さらに、円セグメントの固有の定義ができるよう、開始から終了までの角度の方向を指定してください。

SD42692 \$SC\_JOG\_CIRCLE\_MODE、ビット 0

このビットは、円弧軌跡に沿った移動方向も指定します。

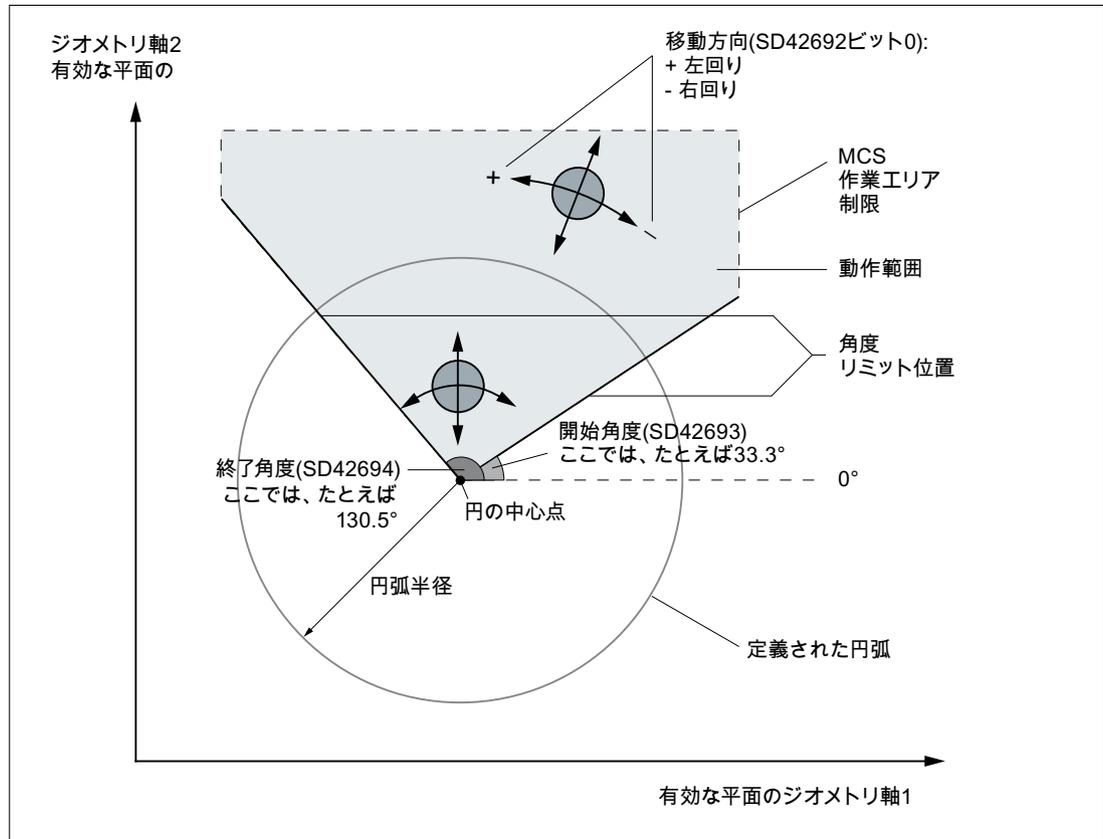


図 4-6 JOG での円移動:円セグメントの加工

## 工具半径の許容量

セッティングデータ SD42692 \$SC\_JOG\_CIRCLE\_MODE にビット 1 を設定した場合、動作領域制限の監視中に工具半径が考慮されます(円弧を定義した場合、開始および終了角のため円セグメント加工も制限されます)。

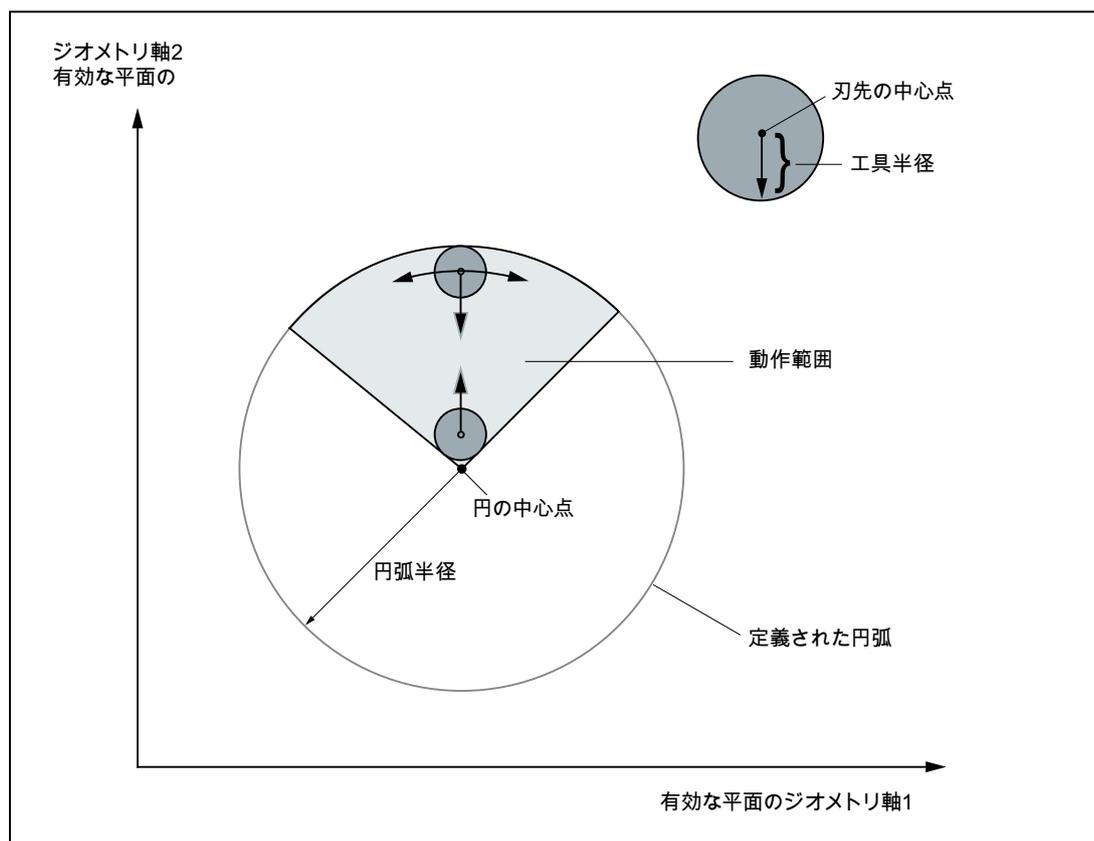


図 4-7 JOG での円移動:工具径補正付きの円セグメントの内径加工

### 注記

工具半径を考慮に入れ、円の中心周りの内径加工の最小可能半径も制限されます。

## 軸監視

この機能が有効で、締結した軸の現在の軸の位置が作業領域限界を超えている場合、軸は移動せず、アラームが出力されます。

作業領域限界を超える軸の位置は、以下のとおりです。

- 内径加工の場合:定義された円の外側の位置
- 外径加工の場合:定義された円の内側の位置

## 4.11 JOG での円移動

- 円セグメント加工の場合:開始および終了角を基本とした制限(セグメント境界線)の外側の位置
- 有効な工具径補正の場合:制限円に接近しすぎた位置
- 有効な工具径補正および内径加工の場合:円の中心に接近しすぎた位置
- 有効な工具径補正および円セグメント加工の場合:セグメント境界線に接近しすぎた位置

### セッティングデータの変更

有効な JOG 移動中に行った機能関連のセッティングデータ SD42690 ~ SD42694 への変更は、新規の JOG 移動をトリガして、有効な JOG 移動を終了した後のみに有効になります。

---

#### 注記

セッティングデータの変更がより短時間で有効になるようにするために、機械のオペレータは以下の手順を実施することができます。

1. 残移動距離の削除によって現在の移動動作をキャンセルします。
  2. セッティングデータを変更します。
  3. JOG 移動を再起動します。
- 

### キャンセル/解除

JOG 移動中に、「JOG での円移動」機能を解除した場合(DB21、... DBX30.6 = 0)、JOG 移動はキャンセルされ、アラームが出力されます。

---

#### 注記

このアラームメッセージを回避するには、機械ユーザは次の実施手順に従ってください。

1. 残移動距離の削除によって現在の移動動作をキャンセルします。
  2. 軸が停止状態になってから機能を解除します。
- 

## 4.11.2 パラメータ設定

### 2 番目のジオメトリ軸の移動方向

2 番目のジオメトリ軸の移動方向は、以下のマシンデータで設定できます。

## MD10735 \$MN\_JOG\_MODE\_MASK (JOG モードの設定)

ビット	値	意味
6	0	有効な平面の 2 番目の軸をプラス方向に移動すると半径が拡大し、マイナス方向に移動すると半径が縮小します。移動の動きは、内径または外径加工のいずれが有効になっているかには関係しません。
7	0	

ビット	値	意味
6	0	有効な平面の 2 番目のジオメトリ軸のプラス方向への移動は、必ず常に制限円の方向に向かいます。つまり、内径加工の半径は拡大し、外径加工の半径は縮小します。
7	1	

ビット	値	意味
6	1	有効な平面の 2 番目のジオメトリ軸のマイナス方向への移動は、必ず常に制限円の方向に向かいます。つまり、内径加工の半径は拡大し、外径加工の半径は縮小します。
7	1	

## 半径の値としての円の中心点の座標

初期設定では、軸の指定された座標は、直径指定が有効な場合は常に直径の値です。これは、以下のマシンデータを使って JOG での円移動の円の中心座標を指定することで変更できます。

## MD20360 \$MC\_TOOL\_PARAMETER\_DEF\_MASK (工具パラメータの定義)

ビット	値	意味
13	0	JOG での円移動で、円の中心点の座標は直径の値(初期設定)。
	1	JOG での円移動で、円の中心点の座標は常に半径の値。

4.11 JOG での円移動

円の中心点

移動する円の中心点は、次のセッティングデータで設定します。

SD42690 \$SSC\_JOG\_CIRCLE\_CENTRE (円の中心点)

座標データは WCS を基準とします。これは、直径の値または半径の値のいずれかとすることができます(MD20360 ビット 13 の設定に依存)。

有効な平面のジオメトリ軸の中心点座標のみが評価され、平面に直交しているジオメトリ軸の座標は評価されません。

円弧半径

内径加工の最大円の半径、または外径加工の最小円の半径は、次のセッティングデータで設定します。

SD42691 \$SSC\_JOG\_CIRCLE\_RADIUS (円の半径)

指定された半径は、WCS を基準とします。

内径または外径加工

加工モード(内径または外径加工)は、以下のセッティングデータで定義されます。

SD42692 \$SSC\_JOG\_CIRCLE\_MODE (JOG 円モード)

ビット	値	意味
2	0	内径加工が行われます。 SD42691 の円の半径が、最大可能半径です。
	1	外径加工が行われます。 SD42691 の円の半径が、最小可能半径です。

円セグメントの加工

作業領域を制限する円のセグメントを定義するための開始および終了角度は、以下のセッティングデータで設定します。

SD42693 \$SSC\_JOG\_CIRCLE\_START\_ANGLE (円の開始角度)

SD42694 \$SSC\_JOG\_CIRCLE\_END\_ANGLE (円の終了角度)

角度データは、有効な平面の横軸を基準にします。

開始から終了角度の方向は、以下のセッティングデータで指定します。

SD42692 \$SC\_JOG\_CIRCLE\_MODE (JOG 円モード)

ビット	値	意味
0	0	開始から終了角度の方向は左回りです。
	1	開始から終了角度の方向は右回りです。

#### 注記

開始から終了角度がゼロの場合、一周円を制限なく移動できます。

### 工具半径の許容量

工具半径を考慮するかどうかは、以下のセッティングデータで選択できます。

SD42692 \$SC\_JOG\_CIRCLE\_MODE (JOG 円モード)

ビット	値	意味
1	0	作業領域制限を監視している場合 (円弧を定義、円弧セグメント加工の場合は開始および終了角度から生じる制限も)、工具半径は考慮されません (初期設定)。
	1	作業領域制限を監視する場合、工具半径を考慮します。

### 4.11.3 補足条件

#### 直径指定が有効

有効な平面のジオメトリ軸のインクリメンタル移動およびハンドル移動中、2つの関連するジオメトリ軸の1つで直径指定が有効な場合でも、移動は必ず半径指定で行われます。

## 4.11 JOG での円移動

### 軸がジオメトリ軸でない場合

有効な平面の両方の軸をジオメトリ軸として定義してください。

- これは、MD20050 \$MC\_AXCONF\_GEOAX\_ASSIGN\_TAB を設定するか、
- パートプログラムで GEOAX をプログラミングして行います。

有効な平面の軸がジオメトリ軸として定義されていない場合、軸は移動されず、アラームが出力されます。

### 軸がハース軸の場合

円弧軌跡に関連する軸がハース軸である場合、軸は移動されず、アラームが出力されません。

### プロテクションゾーン

プロテクションゾーンは監視されません。

## 4.11.4 用途例

### 初期状態

- ジオメトリ軸は、以下のように設定します。

MD20050 \$MC\_AXCONF\_GEOAX\_ASSIGN\_TAB[0] = 1 1 番目のジオメトリ軸= 1 番目のチャンネル軸

MD20050 \$MC\_AXCONF\_GEOAX\_ASSIGN\_TAB[1] = 2 2 番目のジオメトリ軸= 2 番目のチャンネル軸

MD20050 \$MC\_AXCONF\_GEOAX\_ASSIGN\_TAB[2] = 3 3 番目のジオメトリ軸= 3 番目のチャンネル軸

- G17 が有効(初期設定):
  - X 軸 = 有効な平面の 1 番目のジオメトリ軸
  - Y 軸 = 有効な平面の 2 番目のジオメトリ軸
- 選択済み工具の半径訂正值は 0.02 mm。

- 現在の軸位置は、以下のとおりです。
  - X 軸:10
  - Y 軸:10
- 軸は原点確立済みです。

### パラメータの割り付け

SD42690 \$SC\_JOG\_CIRCLE\_CENTRE[AX1] = 円の X 軸の中心を 10 mm の位置にする  
10

SD42690 \$SC\_JOG\_CIRCLE\_CENTRE[AX2] = 円の Y 軸の中心を 20 mm の位置にする  
20

SD42691 \$SC\_JOG\_CIRCLE\_RADIUS = 20 円の半径を 20 mm にする

SD42692 \$SC\_JOG\_CIRCLE\_MODE = 2 プラスキーで移動:  
円を左回りに移動  
工具径補正の考慮  
内径加工

SD42693 \$SC\_JOG\_CIRCLE\_START\_ANGLE 有効な開始角度なし  
= 0

SD42694 \$SC\_JOG\_CIRCLE\_END\_ANGLE = 0 有効な終了角度なし

### 結果

内径の円加工が行われます。

プラスの移動キーを使って X 軸を連続移動することで、円の中心点(10, 20)周りに半径 10 mm の左回り(G2)の円の動きを開始できます。

Y 軸の移動を使って、円の半径を 0.02 ~ 19.98 の間で変更できます。

## 4.12 工具方向の後退(ジョグ後退)

### 4.12.1 概要

#### 機能

機能「ジョグ後退機械機能での工具方向の後退」(以降、「ジョグ後退」と言います)は、AUTOMATIC モードまたは MDI モードで、コントローラの電源オフまたはチャンネルリセットによるプログラム中止後に、ワーク座標系(WCS)での手動による工具後退をサポートします。

特に、下記の機能の特定の特性が考慮されます。

- フローティングチャックとエンコーダ付き速度制御主軸ありのタッピング(G33)
- フローティングチャックと位置制御主軸なしのタッピング(G331、G332)
- 旋回サイクル CYCLE800 または方向座標変換を備える工具オリエンテーションの加工

#### 適用

ジョグ後退は、チャンネルリセットまたは電源切断の結果、以下の機能のいずれかが割り込まれた場合のみ選択できます。

- G33/G331/G332 を使用したタッピング
- 工具タイプ 2xx (穴あけ工具)を使用した加工

#### 復元されるデータ

プログラム中止後に工具方向の後退を実行するために、プログラム中止前にチャンネル内で有効であった下記のデータが復元されます。

- 有効な工具オフセット
- 有効な加工平面
- 有効な工具ホルダ
- 座標変換パラメータを含む有効な座標変換データブロック
- G33 または G331/G332 を含むねじグループのデータ
- 座標変換に関連する軸の位置

このデータがプログラム中止後に全て使用できる場合、ジョグ後退機械機能が選択されると、データがチャンネルに復元されます。この場合、ワーク座標系(WCS)は、ジオメトリ軸の1つが工具軸の方向になるようにコントローラで割り出されます。次に、このジオメトリ軸を移動することにより、工具後退を手動でおこなうことができます。

#### 注記

##### データバックアップ

指定されたデータ自体はバックアップされません。このデータへの参照のみです。ジョグ後退の選択の前にデータが変更された場合、変更されたデータに基づいてこの機能がおこなわれます。

### 制限事項

割込信号によって起動可能なプログラムは、ジョグ後退では実行されません。

## 4.12.2 パラメータ設定

### 4.12.2.1 電源投入後のジョグ後退の自動選択

コントローラの起動(電源投入)後、標準で BAG チャンネルは設定された初期設定モードになります。

MD10720 \$MN\_OPERATING\_MODE\_DEFAULT[<モードグループ>] = <初期設定モード>

以下の NC 別マシンデータを使って、BAG チャンネルに後退データがある場合に、コントローラ起動後にどの動作状態が選択されるかを設定できます。

MD10721 \$MN\_OPERATING\_MODE\_EXTENDED[<モードグループ>] = <値>

規格値	意味
0	MD10720 \$MN_OPERATING_MODE_DEFAULT に対応する運転モードの選択
1	以下の場合にジョグモードを選択:DB21、... DBX377.5 == 1 (後退データが使用可能)

## 4.12 工具方向の後退(ジョグ後退)

## 4.12.2.2 移動方向の有効化

後退は、以下の NC 別マシンデータを使って移動の正方向に制限するか、または両移動方向で有効にできます。

MD10735 \$MN\_JOG\_MODE\_MASK, Bit 8 = <値>

規格値	意味
0	リリース:正の移動方向
1	リリース:正および負の移動方向

## 4.12.2.3 検出器の状態

検出器の現在の状態は、以下の軸別のマシンデータで表示するか、選択の目的の動作に設定してください。

MD34210 \$MA\_ENC\_REFP\_STATE[<軸>] = <値>

値 <sup>1)</sup>	意味
絶対値エンコーダ	
2	エンコーダ位置合わせ済み
インクリメンタルエンコーダ	
1	自動原点確立:有効、ただしエンコーダはまだ原点未確立
2	自動原点確立:エンコーダは原点確立済みであり、イグザクトストップでは、自動原点確立は次のエンコーダ起動時に有効になります
3	現在位置の復元:電源オフの前に記憶された最後の軸位置が復元されます(自動原点確立なし)
1)ジョグ後退に関連する値のみ記述	

### 4.12.3 選択

#### 機能

##### 必要条件

関連するチャンネルの有効な後退データが使用可能であり、チャンネルがジョグモードで、リセット状態にある場合にのみ、ジョグ後退の選択が可能です。

- DB21、... DBX377.5 == 1 (後退データが使用可能)
- DB11 DBX(n\*20+6).2 == 1 (アクティブモード:ジョグ、n=0、1、2...、モードグループ1、2、...)
- DB21、... DBX35.7 == 1 (チャンネル状態:リセット)

##### 軸と主軸

工具方向の後退に関連する機械軸の有効なすべての検出器は、コントローラを再度電源投入したときに(電源投入)、「原点確立済み」または「復元済み」の状態にある必要があります。

コントローラの次回の起動(電源投入)後に現在位置を自動復元する場合について詳細は、以下を参照してください。

参照先:

『基本機能』機能マニュアル、「R1 原点確立」>「機械レファレンス点の自動復元」の章

##### 軸入れ替え

選択時に後退に関連するすべての軸と主軸がチャンネルに割り当てられていない場合、割り当てなしの軸について自動的な軸の入れ替えがおこなわれます。

##### 座標系

ジョグ後退の選択により、チャンネル軸移動のワーク座標系(WCS)が設定されます。後退に関連する軸は、この WCS 内でのみ移動できます。後退に関与しないすべての軸もまた、この機械座標系(MCS)内で移動できます。

#### 選択オプション

ジョグ後退の選択は、電源投入後のジョグモードの自動選択でサポートできます。ジョグ後退の実際の選択は、ユーザーインターフェースによる手動選択または PLC ユーザープログラムで行われます。

## 4.12 工具方向の後退(ジョグ後退)

**ジョグモードの自動選択**

ジョグ後退の自動選択をパラメータ設定し(「電源投入後のジョグ後退の自動選択 (ページ 241)」の章を参照)、コントローラ起動(電源投入)後にチャンネルの後退データがある場合、モードグループにジョグモードが選択されます。

**操作画面による選択**

ジョグ後退の選択は、操作画面上で以下を使って行われます。

「機械操作エリア」 > 「ETC キー」 (「>」) > 「後退」

**注記**

ソフトキー「後退」は、有効な工具に後退データがある場合のみ表示されます。

**PLC ユーザープログラムによる選択**

PLC ユーザープログラムによってジョグ後退を選択するには、下記のアクションをおこなってください。

- 後退データが使用可能であるかどうかのチャンネル別の確認  
DB21、... DBX377.5 == 1 (後退データが使用可能)
- ジョグモードのモードグループ別の選択:  
DB11、... DBX(n\*20+6).2 == 1 (アクティブモード:ジョグ、n=0、1、2 ...、モードグループ 1、2、...)
- PI サービス「RETRAC」によるチャンネル別のジョグ後退の選択。  
A ファンクションブロック FB 4 による PI サービス「RETRAC」の起動について詳細は、以下を参照してください。

**参照先:**

『基本機能』機能マニュアル、「P3:SINUMERIK 840D sl の PLC 基本プログラム」の「構成部分の説明」の「PI サービス」「PI サービス:RETRAC」の章を参照してください。

- ジョグ後退の選択を確定した後(以下を参照してください)、ワーク座標系を選択してください。
  - DB19.DBX0.7 (現在の座標系:1 ⇒ WCS、0 ⇒ MCS)
  - DB19.DBX20.7 (要求:MCS/WCS の切り替え)

**選択の確定**

ジョグ後退を選択した後、NC/PLC インターフェース信号が設定されます。

DB21、... DBX377.4 = 1 (ジョグ後退有効)

### ねじ切り(G33 または G331/G332)

ねじ切り運転(G33 または G331/G332)中にプログラム実行が中止された場合、ジョグ後退の選択時に工具の軸の軸グループと主軸が復元されます。さらに関連する軸のコントローラパラメータとパラメータセットも、パートプログラムでプログラム指令されたねじ切りに従って設定されます。

## 4.12.4 工具後退

### 一般的な後退動作

ワーク座標系(WCS)で、ジョグ後退の選択時に指定された後退軸(ジオメトリ軸)を手動で移動することにより、工具が後退します。後退移動は、機械操作パネル(MCP)の移動キー、または手動パルス発生器によって行うことができます。後退は、移動範囲リミット(作業領域リミット、ソフトウェアリミットスイッチなど)内で可能です。

後退移動は、NC ストップ/NC スタートにより再度停止と開始できます。

工具後退に関連しない軸と主軸は、必要に応じて手動で移動できます。

座標系の切り替えが可能です(MCS ⇄ WCS)。

### 移動方向

初期設定に応じて、後退移動は正の移動方向にのみ有効です。負方向の移動も可能である場合は、下記でそれを明確に有効にしてください。

MD10735 \$MN\_JOG\_MODE\_MASK、ビット 8 = 1

後退動作は、処理タイプおよび工具タイプによって異なります。

加工タイプおよび有効な工具タイプに応じて、以下の後退動作が可能です。

- ねじ切り(G33)またはタッピング(G331, G332):  
工具タイプに関係なく、移動動作は後退軸(工具軸)のみで可能です。
- 工具タイプ 2xx (穴あけ工具):移動動作は後退軸(工具軸)のみで可能です。
- 工具タイプ 4xx (研削工具):後退動作は不可
- 工具タイプ 5xx (旋削工具):後退動作は不可

## 4.12 工具方向の後退(ジョグ後退)

### ねじ切り作業のための後退動作

ねじ切り(G33)またはタッピング(G331, G332)では、後退動作は後退軸または主軸のいずれかが移動した場合に実行されます。

### 手動パルス発生器を使用した後退

後退移動関連のいずれかの軸が移動キーによって移動された場合、後退移動関連の他の軸の手動パルス発生器のパルスは無視されます。

複数の軸に対して手動パルス発生器が選択されており、これらの軸が移動された場合、手動パルス発生器パルスは下記の順序で評価されます。

1. 後退軸
2. 主軸
3. 後退移動に関連しない軸/主軸

他の手動パルス発生器のパルスは無視され、先行する手動パルス発生器が停止状態でのみ評価されます。

### 禁止される機能

ジョグ後退時に、以下の機能はブロックされ、実行されません。

- DB31、...DBX30 による主軸起動
- ファンクションブロック FC18 による後退に関連する主軸または軸の移動
- 後退に関連する主軸または軸の PLC 制御軸への切り替え
- 後退に関連する主軸または軸の別のチャンネルへの変更
- メイン軸(コマンド軸、揺動軸、FC18/同時軸)としての後退関連の主軸または軸の使用

### 4.12.5 解除

ジョグ後退は、以下によってチャンネル別に解除されます。

- チャンネルリセット:DB21、... DBX7.7 = 1 (リセット)
- 操作画面:[戻る] ([<<]) ソフトキー

以下の初期設定が有効です:

- MD20110 \$MC\_RESET\_MODE\_MASK[<チャンネル>]
- MD20150 \$MC\_GCODE\_RESET\_VALUES[<チャンネル>]
- MD20151 \$MC\_GCODE\_RESET\_MODE[<チャンネル>]

解除後にこのチャンネルはジョグモードで、チャンネル状態は「リセット」です。

すべてのチャンネル軸を再度手動で移動できます。後退データは保持されます。したがって、ジョグ後退を再選択できます。

#### NC/PLC インタフェース信号

ジョグ後退を解除することにより、NC/PLC インタフェース信号はリセットされます。

DB21、... DBX377.4 = 0 (ジョグ後退有効)

### 4.12.6 再選択

後退データがあれば、ジョグ後退はいつでも再選択できます。

DB21、... DBX377.5 == 1 (後退データが使用可能)

再選択した場合、元の後退データが復元されます。

#### 軸位置の変更

ジョグ後退の解除中は、チャンネル軸はジョグモードで移動できます。ジョグ後退が再選択された場合、新しい軸位置に基づいて後退移動が実行されます。

## 4.12.7 加工の継続

## AUTOMATIC モード

中止されたパートプログラムが **AUTOMATIC** モードで **NC** スタートにより続行する前に、「復元済み」または「原点未確立」状態の有効な検出器付きのすべての機械軸について、原点確立をおこなってください。

## 通知

## 考えられる軸位置の移動

停電後にコントローラを起動した後、マシンデータの設定に対応して、インクリメンタル検出器の軸位置は原点同期済みになるか、または復元されます。工具がジョグ後退機械機能で後退した後は、位置が復元された軸の原点が確立されます。

## MDI モードとオーバーストア

MDI モードのオーバーストア機能では、軸を原点確立することなく、復元された位置で、加工をおこなうこともできます。このためには、復元された位置での **NC** スタートを、下記でチャンネル別に明確に有効にしてください。

**MD20700 \$MC\_REFP\_NC\_START\_LOCK = 2**

## 中断点でのブロック検索

パートプログラムの処理は、中断点でのブロック検索によって中断点から続行できます。中断前のメインプログラムレベルの最後のブロックは、中断点として使用可能です。

ブロック検索の機能と動作について詳細は、以下を参照してください。

## 参照先

『基本機能』機能マニュアル、「K1:モードグループ、チャンネル、プログラム運転、リセット応答」 > 「ブロック検索」または「ブロック検索タイプ 5 SERUPRO」の章

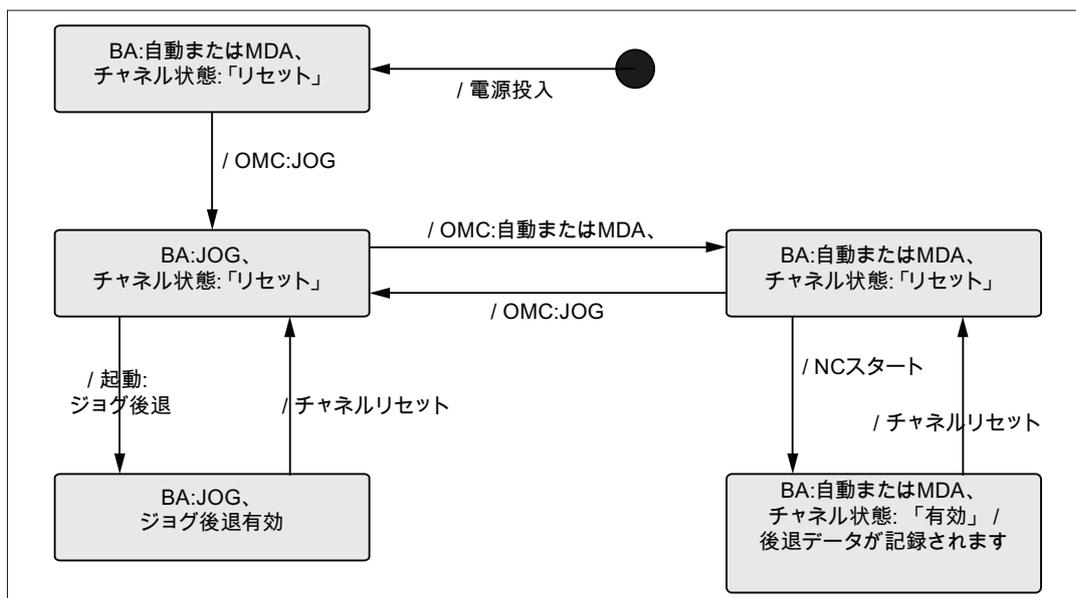
『旋削またはフライス削り』操作マニュアルの「ワークの加工」 > 「特定点での加工の開始」の章

## NC スタートでの継続

AUTOMATIC モードまたは MDI モードでの NC スタートにより、選択位置からパートプログラムの処理が継続します。ジョグ後退に関して、下記のアクションがおこなわれます。

- 後退データは削除されます。
- NC/PLC インタフェース信号がリセットされます。  
DB21、... DBX377.5 = 0 (後退データが使用可能)
- 現在のデータ環境が保存されます。

## 4.12.8 状態図



OM 運転モード

OM 運転モード変更

C

図 4-8 状態図: ジョグ後退

## 4.12 工具方向の後退(ジョグ後退)

## 4.12.9 システムデータ

ジョグ後退では、次のシステムデータが使用できます。

意味	システム変数\$VA_	NC/PLC インタフェース	OPI 変数
後退データが使用可能	-	DB21、...DBX377.5	retractState、ビット0
ジョグ後退有効	-	DB21、...DBX377.4	retractState、ビット1
後退軸	-	-	retractState、ビット2～3
復元された位置、1番目の検出器	\$AA_POSRES	DB31、...DBX71.4	aaPosRes
復元された位置、2番目の検出器	\$AA_POSRES	DB31、...DBX71.5	aaPosRes

## 参照先

## OPI とシステム変数

「リストブック 2」 リストマニュアル、「変数」の章。

## 4.12.10 必要条件

## インクレメンタル検出器

ユーザーは、コントローラで既知の保存された最後の位置の変更を防ぐために、停電時にインクレメンタル検出器付きの機械軸が十分な速度にクランプされることを確認してください。そうでない場合、コントローラの再起動時の想定位置が、機械軸の実位置と大きく異なります。これらの軸については、**ドライブの自律的後退**を起動しないでください。

コントローラの次回の起動(電源投入)後に現在位置を自動復元する場合について詳細は、以下を参照してください。

## 参照先:

『基本機能』機能マニュアル、「R1 原点確立」>「機械レファレンス点の自動復元」の章

## ドライブの自律的後退

「ドライブの自律的後退」機能は、後退移動関連の機械軸については**起動しない**でください。

### 軸連結

軸連結は、ジョグ後退の選択では復元されません。

### G63 を使用したタッピング

ジョグ後退は、フローティングチャック付きタッピングとエンコーダなしの速度制御主軸(G63)では可能ではありません。

### 座標変換

ジョグ後退を通じて座標変換が選択される場合、座標変換に関連するすべての機械軸の有効な検出器が「原点確立済み」または「復元済み」状態であることが必要です。

ヘキサポッドの平行キネマティックスなどの **OEM 座標変換**は、**原点確立済み**の検出器でのみ移動できます。復元された軸位置による移動は可能ではありません。

### 直接プログラム指令された旋回軸による工具オリエンテーション

NC 機能によってではなく、旋回軸の直接のプログラミングを通じて工具オリエンテーションがおこなわれる場合、ジョグ後退では工具方向に後退移動を生成できません。

### NCU リンク

またジョグ後退は、軸の NCU 全体の移動と組み合わせることができます(「NCU リンク (ページ 92)」の章を参照してください)。軸コンテナの状態は、ジョグ後退では変更されません。たとえば MDI モードでジョグ後退の移動前に、ユーザーが工具方向の後退に必要なあらゆる調整を行う必要があります。

## 4.13 自動モードでのハンドルの使用

### 4.13.1 自動モードのハンドルオーバーライド

#### 4.13.1.1 一般機能

#### 機能

この機能によって、自動モード(自動、MDI)でハンドル軸を使用した軸の移動、またはその速度の直接変更ができます。

ハンドルオーバーライドは、NC 言語命令 FD(軌跡軸の場合)と FDA(位置決め軸の場合)を使用して NC パートプログラムで有効にされ、**ノンモーダル**になります。

4.13 自動モードでのハンドルの使用

位置決め軸で、移動指令 POSA を使用して、ハンドルオーバーライドのモードを有効にできます。プログラム指令された目標位置に到達すると、ハンドルオーバーライドは再び無効になります。

付加軸は、同時に、または同じ NC ブロックの補間を使用して移動できます。

同時位置決め軸機能はまた、PLC ユーザプログラムで有効にできます。

ハンドルオーバーライドの種類

プログラム指令された送り速度に応じて、次の種類のハンドルオーバーライドがあります。

- 軌跡指令  
軸送り速度 = 0 (FDA = 0)
- 速度オーバーライド  
軸送り速度 > 0 (FD または FDA > 0)

下の表に、「自動モードのハンドルオーバーライド」機能で作用を受ける可能性がある軸のタイプを示します。

「自動モードのハンドルオーバーライド」機能で作用を受ける軸		
軸タイプ	速度オーバーライド	軌跡指令
位置決め軸	FDA[AXi] > 0、 軸で動作します	FDA[AXi] = 0
同時位置決め軸	パラメータ「ハンドルオーバーライドの有効化」 = 1 と 軸送り速度 > 0 (FC18 から)	パラメータ「ハンドルオーバーライドの有効化」 = 1 と 軸送り速度 = 0 (FC18 から)
軌跡軸	FD > 0 、軌跡速度で動作します	不可

軌跡指令

軸送り速度 = 0(例: FDA[AXi] = 0)の場合、プログラム指令された目標位置への位置決め軸の移動動作は、割り当てられたハンドルをユーザが回転することで完全に制御できます。

ハンドルの回転方向により、軸の移動方向が決定されます。ハンドルオーバーライド中に、プログラム指令された目標位置を超過することはできません。軸は、プログラム指令された目標位置に反対方向から移動できます。この反対方向の移動では、軸のリミット位置によってのみ制限されます。

ブロック切り替えは、次の場合におこなわれます。

- プログラム指令された目標位置にこの軸が到達している場合。

または

- 残移動距離が軸のインタフェース信号  
DB31、... DBX2.2 (残移動距離削除)で削除された場合。

この時点から、軌跡指令が無効になり、以降のハンドルパルスが無効になります。

以降、インCREMENTALのプログラム指令位置は、最後のプログラム指令位置ではなく、この中断点を示します。

## 速度オーバーライド

速度オーバーライドについては、軸送り速度と軌跡送り速度があります。

- **軸速度オーバーライド (FDA[AXi] > 0):**

位置決め軸は、プログラム指令された軸の送り速度で、目標位置に移動します。割り当てられたハンドルを使用して、その回転方向に応じて、軸速度を増加させたり、最小値の0に設定したりすることが可能です。得られる軸の送り速度は、最大速度で制限されます。ただし軸は、プログラム指令方向に対して反対方向に移動することはできません。

プログラム指令された目標位置に軸が達するとすぐに、ブロック切り替えがおこなわれます。これにより、速度オーバーライドが自動的に無効になり、以降のハンドルパルスが無効になります。

PLC で目標位置と速度が定義される同時位置決め軸の場合も同様です。

- **軌跡速度オーバーライド (FD > 0):**

NC ブロックでプログラム指令された軌跡軸は、プログラム指令された送り速度で目標位置に移動します。速度オーバーライドが有効な場合、プログラム指令された軌跡速度は、第1ジオメトリ軸のハンドルによって生成される速度でオーバーライドされます。プログラム指令された目標位置に到達するとすぐに、ブロック切り替えがおこなわれます。

軌跡速度は、ハンドルの回転方向に応じて、増加させたり、最小値の0に減少させることができます。ただし、ハンドルオーバーライドで移動方向を反転することはできません。

#### 4.13 自動モードでのハンドルの使用

##### 用途例

「自動モードのハンドルオーバーライド」機能は、一般的に研削盤で使用されます。たとえばユーザは、ハンドル(軌跡指令)を使用して、ワーク上に揺動研削砥石を位置決めできます。接触後に、移動動作は終了し、(DB31、... DBX2.2 (残移動距離削除)を有効にすることによって)ブロック切り替えが開始されます。

##### 必要条件

「自動モードのハンドルオーバーライド」を有効にするために、次の必要条件を満たしてください。

- ハンドルは、当該軸に割り当ててください。
- 割り当てられたハンドル用に、パルス倍率があります。

##### ハンドルの割り当て

接続されたハンドルを軸に割り当てることは、次のインタフェース信号によるユーザーインタフェースまたは PLC ユーザーインタフェース経由の「ハンドルを使用した手動移動 (ページ 190)」と類似しています。

- 機械軸:
  - DB31、... DBX4.0 - 2 (ハンドルの起動)
- ジオメトリ軸:
  - DB21、... DBX12.0 - 2 (ジオメトリ軸 1:ハンドルの起動)
  - DB21、... DBX16.0 - 2 (ジオメトリ軸 2:ハンドルの起動)
  - DB21、... DBX20.0 - 2 (ジオメトリ軸 3:ハンドルの起動)

ハンドルが割り当てられていない軸に対してハンドルオーバーライドがプログラム指令されている場合、次の事例に分かれます。

- **速度オーバーライドの場合:**  
軸は、プログラム指令速度で移動します。  
自己確認アラームが出力されます(応答なし)。
- **軌跡指令の場合:**  
速度が 0 であるため、移動動作はおこなわれません。  
自己確認アラームが出力されます(応答なし)。

---

##### 注記

軌跡軸の速度オーバーライドの場合、**第 1 ジオメトリ軸のハンドルのみ**が軌跡速度に適用されます。

---

## ハンドル倍率

1 クリック位置分ハンドルを回転させることによって生成される軸の移動軌跡は、次のような様々な要素によって決まります(「ハンドルを使用した手動移動 (ページ 190)」の章を参照してください)。

- 選択ステップ量:  
MD11330\$MN\_JOG\_INCR\_SIZE\_TAB[5](INC/ハンドルのステップ量)  
または  
SD41010\$SN\_JOG\_VAR\_INCR\_SIZE(ジョグの可変ステップ量)
- ステップ値の倍率:  
MD31090 \$MA\_JOG\_INCR\_WEIGHT
- 1 クリック位置毎のハンドルパルス数:  
MD11320 \$MN\_HANDWH\_IMP\_PER\_LATCH

たとえば、機械機能 INC1 と上記のマシンデータの初期設定が選択された場合、軸はハンドル 1 クリック位置毎に 0.001 だけ移動します。

速度オーバライドの場合、速度は、一定時間内のハンドルを使用しておこなわれた移動軌跡に対応します。

## 例

### 前提条件:

オペレータは 100 パルス/秒でハンドルを回転させます。

選択される機械機能は INC100 です。

ハンドル倍率について、上記のマシンデータが初期設定されています。

- ⇒ 1 秒毎のハンドル移動軌跡: 10 mm
- ⇒ 速度オーバライド: 0.6 m/min

## PLC インタフェース信号

ハンドルオーバライドが有効になるとすぐに、PLC に対して次のインタフェース信号が信号 1 に設定されます。

- 位置決め軸/同時位置決め軸/コマンド軸/揺動軸の場合:  
DB31、... DBX62.1(ハンドルオーバライド有効)
- 軌跡軸の場合:  
DB21、... DBX33.3(ハンドルオーバライド有効)

## 4.13 自動モードでのハンドルの使用

軌跡入力の場合、移動方向に応じて、当該のインタフェース信号が PLC に出力されません。

- 機械軸:
  - DB31、... DBX64.6 / 7 (移動指令「マイナス」/「プラス」)
- ジオメトリ軸:
  - DB21、... DBX40.6 - 7 (ジオメトリ軸 1:移動指令「マイナス」/「プラス」)
  - DB21、... DBX46.6 - 7 (ジオメトリ軸 2:移動指令「マイナス」/「プラス」)
  - DB21、... DBX52.6 - 7 (ジオメトリ軸 3:移動指令「マイナス」/「プラス」)

## リミット位置

軸のリミット位置(ソフトウェアリミットスイッチ、ハードウェアリミットスイッチ、ワーキングエリアリミット)は、ハンドルオーバーライドと組み合わせで有効になります。軌跡指令では、プログラム指令された移動方向にプログラム指令された目標位置まで、ハンドルで軸を移動できます。

得られた速度は、軸マシンデータで制限されます。

MD32000 \$MA\_MAX\_AX\_VELO(最大軸速度)

## NC ストップ/オーバーライド = 0

送り速度オーバーライドが 0%に設定されているか、ハンドルオーバーライドが有効な間にNC ストップが開始された場合、次のことが適用されます。

- 軌跡指令の場合:

途中で受信したハンドルパルスは、合計されて保存されます。NC スタートまたは送り速度オーバーライド > 0%の場合、保存されたハンドルパルスが有効になります(つまり、移動します)。

ただし、手動パルス発生器を最初に無効にした場合[IS DB21、... DBX12/16/20.0-2 経由(ジオメトリ軸 1/2/3:ハンドルの起動)]、設定されたハンドルのパルスが削除されません。
- 速度指定の場合:

途中で受信したハンドルパルスは合計されず、有効になりません。

### 4.13.1.2 プログラミングとハンドルオーバーライドの有効化

#### 概要

NC 言語命令 FD(軌跡軸の場合)と FDA(位置決め軸の場合)によってハンドルオーバーライドがプログラム指令されている場合、次のことを守ってください。

- FDA と FD 機能はノンモーダルです。  
位置決め軸の例外:移動命令 POSA がプログラム指令されている場合、該当する位置決め軸はブロック切り替えに影響しないので、ハンドルオーバーライドもまたモーダルに動作できます。
- ハンドルオーバーライドを FDA または FD によって有効にした場合、位置決め軸または軌跡軸の NC ブロックで目標位置をプログラム指令してください。プログラム指令された目標位置に到達すると、ハンドルオーバーライドは再び無効になります。
- 同じ NC ブロックで、FDA と FD、または FA と F をプログラム指令することはできません。
- 位置決め軸と割り出し軸は同一にできません。

#### 位置決め軸

ハンドルオーバーライドの構文:  $FDA[AXi] = [送り速度値]$

#### 例 1:

##### 速度オーバーライドの有効化

```
N10 POS[U]=10 FDA[U]=100 POSA[V]=20 FDA[V]=150 . . .
```

POS[U]=10	位置決め軸 U の目標位置
FDA[U]=100	位置決め軸 U の速度オーバーライドの有効化、軸速度 U=100 mm/min
POSA[V]=20	位置決め軸 V の目標位置(モーダル)
FDA[V]=150	位置決め軸 V の速度オーバーライドの有効化、軸速度 V=150 mm/min

#### 例 2:

##### 同じ NC ブロックの軌跡指令と速度オーバーライドの有効化

### 4.13 自動モードでのハンドルの使用

```
N20 POS[U]=100 FDA[U]= 0 POS[V]=200 FDA[V]=150 . . .
```

POS[U]=100      位置決め軸 **U** の目標位置  
FDA[U]= 0      位置決め軸 **U** の軌跡指令を有効にします。  
POS [V]=200      位置決め軸 **V** の目標位置  
FDA[V]=150      位置決め軸 **V** の速度オーバーライドの有効化、軸速度 **V=150 mm/min**

### 軌跡軸

ハンドルオーバーライドの構文:    FD = [送り速度値]

「自動モードのハンドルオーバーライド」を有効にするために、次の必要条件を満たしてください。

- グループ 1 の有効な移動命令:G01、G02、G03、CIP
- イグザクトストップが有効(G60)
- 毎分送り (mm/min または inch/min)が有効(G94)

上記の必要条件がコントローラで確認され、そのいずれかが満たされていない場合にアラームが出力されます。

### 例 3:

#### 速度オーバーライドの有効化

```
N10 G01 X10 Y100 Z200 FD=1500 . . .
```

X10 Y100 Z200      軌跡軸 **X**、**Y**、および **Z** の目標位置  
FD=1500            軌跡軸の速度オーバーライドの有効化、軌跡速度 = **1500mm/min**

### 同時位置決め軸

同時位置決め軸のハンドルオーバーライドは、当該のインタフェース信号を設定することにより PLC から **FC18** 経由で有効にされます。

**DB31**、... **DBX62.1**(ハンドルオーバーライド有効)

速度パラメータ(F\_Value)が値 0 で伝送された場合、有効になったハンドルオーバーライドは、距離入力として動作します。つまりこの場合、軸マシデータから送り値は得られません(「P2 :位置決め軸 (ページ 803)」の章を参照してください)。

MD32060 \$MA\_POS\_AX\_VELO (位置決め軸速度の初期設定)

参照先:

『機能マニュアル、基本機能』; 「PLC 基本プログラム(P3)」

#### 4.13.1.3 自動モードのハンドルオーバーライドの特記事項

##### 速度表示

ハンドルオーバーライドの速度表示では、次の値が表示されます。

- **設定速度**  
= プログラム指令速度
- **現在の速度**  
= ハンドルオーバーライドを含む速度

##### 径方向軸への作用

軸が径方向軸として定義され、DIAMON が有効である場合、ハンドルオーバーライドが有効な間、ハンドルパルスは直径値として解釈および伝送されます。

##### ドライラン運転送り速度

ドライラン運転

DB21、... DBX0.6(ドライラン運転送り速度の有効化) = 1 では、

ドライラン運転送り速度は常に有効です

SD42100 \$SC\_DRY\_RUN\_FEED。

この場合、軌跡指令(FDA[AXi] = 0)を含んだ有効なハンドルオーバーライドにかかわらず、ハンドルからの作用を受けないドライラン運転送り速度で、プログラム指令された目標位置に軸が移動します。つまり、軌跡指令は無効です。

##### DRF が有効

「自動モードのハンドルオーバーライド」が有効である場合、「DRF」機能が有効であるかどうかを確認することが重要です(DB21、... DBX0.3 = 1)。

#### 4.13 自動モードでのハンドルの使用

この場合、ハンドルパルスにより軸の DRF オフセットも発生します。したがって、ユーザはまず、DRF を無効にしてください。

#### 送り速度オーバーライド

送り速度オーバーライドは、ハンドルによる移動の速度に影響しません(例外: 0%). プログラム指令された送り速度にのみ影響します。

軌跡指令で高速なハンドル移動を行った場合、軸がハンドル回転に同期して追従できない場合があります(特に、ハンドルパルス倍率が大きい場合)、その結果軸が行き過ぎることがあります。

### 4.13.2 ハンドルを使用した輪郭ハンドル/軌跡指令(オプション)

#### 「輪郭ハンドル」機能

この機能が有効な場合、AUTOMATIC モードと MDI モードで、軌跡と同期軸をハンドルによって制御できます。

#### 必要条件

固定送り速度、ドライラン送り速度、ねじ切り、タッピングはいずれも選択できません。

#### 適用

SINUMERIK 840D si および SINUMERIK 828D では、「輪郭ハンドル」機能はライセンスを受けたオプションとして使用できます。

#### 入力モード(軌跡入力または速度入力)

手動パルス発生器によって、距離または速度のいずれかを入力できます。

- 軌跡指令

各 IPO クロック周期のハンドルパルスで定義された軌跡の結果として得られた移動速度は、コントロールシステムによって最大許容速度に制限されます。すぐには実行されないハンドルパルスは回収され、その後の時点で移動されます。その結果、軸は動作を継続し、パルスが失われません。

- 速度指定

各 IPO クロック周期のハンドルパルスに基づいて、軸の移動速度が指定されます。IPO クロック周期のハンドルからパルスが受信されなくなると、移動動作はすぐに停止します。軸は動作を継続しません。

入力モードは、次のマシンデータで設定されます。

MD11346 \$MN\_HANDWH\_TRUE\_DISTANCE

## 軌跡速度

軌跡速度(mm/min)は、以下に 応じて異なります。

- 各 IPO クロック周期のハンドルパルス
- マシンデータによるハンドルのパルス評価:  
MD11322 \$MN\_CONTOURHANDWH\_IMP\_PER\_LATCH
- 有効なステップ値(INC1、10、100、...)
- 有効な第 1 ジオメトリ軸のステップ値の距離倍率:  
MD31090 \$MA\_JOG\_INCR\_WEIGHT

軌跡速度は、以下に 依存しません。

- プログラム指令された送り速度モード(mm/min、mm/rev)
- プログラム指令された軌跡速度
- G0 ブロックの早送り速度
- オーバライド(0%オーバライドを除く)

## 移動方向

移動方向は、ハンドルの回転方向によって決まります。

- 右回り:プログラム指令された方向に軸を移動します。  
ブロック切り替え条件(IPO の終了)に達した場合、プログラムは次のブロックに進みます(G60 と同じ動作です)。
- 左回り:プログラム指令された方向とは逆方向に軸を移動します。  
この場合、移動できるのは、実際のブロックの開始までです。先頭へ移動完了後にハンドルが回転を続けている場合でも、パルスは受け取られません。

## 4.13 自動モードでのハンドルの使用

### 機能の適用

この機能は、インターフェース信号によって、または NC プログラムで有効にできます。

- インターフェース信号による適用

DB21、... DBX30.0 - 2

- NC プログラムによる適用

指令 FD=0。輪郭ハンドルは、ノンモーダルに有効になります。つまり、次のブロックでは、輪郭ハンドルが再度適用される前に輪郭ハンドルを有効にすると、ブロックからのプログラム指令された速度が F... になります。

---

#### 注記

以前の NC ブロックで送り速度がプログラム指令されていない場合は、アラームが表示されます。

---

### 「シミュレートされた輪郭ハンドル」機能

「輪郭ハンドル」機能が有効な場合に「シミュレートされた輪郭ハンドル」機能を使用すると、ハンドルまたはハンドルパルスがシミュレートされます。

#### 適用

この機能は、次のインターフェース信号により有効になります。

DB21、... DBX30.3 = 1

送り速度はハンドルパルスによって指定されなくなりますが、NC プログラムでプログラム指令された送り速度 F... になります。

#### 回転方向の指定

シミュレートされたハンドルの回転方向は、インターフェース信号によって指定されます。

DB21、... DBX30.4

#### オーバライド

「シミュレートされた輪郭ハンドル」機能が有効な場合は、NC プログラムの実行と同様にオーバライドが有効です。

#### 移動動作の停止

「シミュレートされた輪郭ハンドル」機能が選択されていない場合、または回転方向が切り替えられた場合は、実際の移動動作が減速カーブに沿って停止します。

## 必要条件

- **速度制限値**
  - 「輪郭ハンドル」機能をインタフェース信号(DB21、... DBX30.0 - 2)によって有効にした場合、ハンドルで入力できる軌跡速度は、次のように制限されます。  
最大軌跡速度 = (プログラム指令された軌跡速度 F...) \* (MD12030 \$MN\_OVR\_FACTOR\_FEEDRATE[ 30 ] + ダイナミック操作に対応した予備)
  - 「輪郭ハンドル」機能を NC プログラムで指令 FD=0 を使用して有効にした場合、ハンドルで入力できる軌跡速度は、マシンデータ(MD32000 \$MA\_MAX\_AX\_VELO)で定義された最大軸速度に制限されます。
- **加速度と速度の制限値**

「輪郭ハンドル」機能を使用して軸を移動している場合、マシンデータ(例: MD32300 \$MA\_MAX\_AX\_ACCEL など)で定義された加速度と(MD32000 \$MA\_MAX\_AX\_VELO)で定義された速度の軸別制限値は維持されます。
- **移動動作の中断**

「輪郭ハンドル」機能を使用して移動中に NC 停止が開始された(DB21、... DBX7.3 または 4)場合は、この機能が選択されたままになります。NC 停止中に受信したハンドルパルスは評価されません。必要条件:MD32084  
\$MA\_HANDWH\_CHAN\_STOP\_COND、ビット 2 == 1
- **DRF**

有効な DRF 機能(DB21、... DBX0.3)には、軌跡のオーバーライド動作もあります。
- **チャンネル別残移動距離削除**

チャンネル別残移動距離削除(DB21、... DBX6.2)によって、輪郭ハンドルによって開始された動作がキャンセルされます。軸は減速され、プログラムが次の NC ブロックで再起動します。
- **金型加工**

輪郭ハンドルを使用して金型加工プログラムを実行すると、適切な表面品質や一定の切断速度が生成されません。

## 4.13 自動モードでのハンドルの使用

### 4.13.3 DRF オフセット

#### 4.13.3.1 DRF オフセット

##### 機能

「DRF オフセット」機能(差動レゾルバ機能)を使用することで、電子手動パルス発生器によって、AUTOMATIC モードで、基本座標系のジオメトリ軸と補助軸へ追加のゼロオフセットシフトを設定できます。

移動対象のジオメトリ軸または補助軸へのハンドルの割り当て(つまり、DRF オフセットのステップ単位が出力されるハンドルの割り当て)は、当該の機械軸によっておこなってください。当該の機械軸は、ジオメトリ軸または補助軸が割り当てられた機械軸です。

DRF オフセットは、軸フィードバック値の画面で表示されません。

##### 用途

DRF オフセットはたとえば、次の用途で使用できます。

- NC ブロック内の工具の摩耗のオフセット  
NC ブロックの処理時間が非常に長い場合、NC ブロック内で工具摩耗のオフセットを手動で設定することが必要になります。(例: 大型正面フライス盤)
- 研削中の高精度オフセット
- 簡単な熱変位補正

---

##### 注記

DRF オフセットによって取り入れられたゼロオフセットは、常にすべてのモードで、RESET 後に有効になります。ただしこれは、パートプログラムでノンモーダルで抑制できます。

---

##### 減速

DRF の手動パルス発生器を使用して生成された速度は、ジョグ速度に関しては減速ができません。

MD32090 \$MA\_HANDWH\_VELO\_OVERLAY\_FACTOR (ハンドル速度に対するジョグ速度の比率(DRF))

## DRF が有効

ハンドル移動によって DRF オフセットを変更するには、DRF を有効にしてください。  
次の必要条件を満たしてください。

- AUTOMATIC モード
- DB21、... DBX0.3 (DRF の起動) = 1

HMI 操作画面上で「プログラム制御」機能を使用して、個別のチャンネルについて DRF オフセットを有効/無効にできます。

次に HMI ソフトウェアは、インタフェース信号:

DB21、... DBX24.3 (DRF の選択) = 1 を設定します。

対応する論理演算がおこなわれると、PLC プログラム(PLC 基本プログラムまたはユーザプログラム)はインタフェース信号

DB21、... DBX0.3(DRF の起動)に、このインタフェース信号を伝送します。

## DRF オフセットの制御

次のように、DRF オフセットの変更、削除、または読み取りをおこなうことができます。

ユーザー:	<ul style="list-style-type: none"> <li>• ハンドルによる移動</li> </ul>
パートプログラム:	<ul style="list-style-type: none"> <li>• 軸別システム変数\$AC_DRF[&lt;軸&gt;]による読み取り</li> <li>• チャンネルのすべての軸に対して、パートプログラム命令 (DRFOF)によって削除</li> <li>• パートプログラム命令(SUPA)によるノンモーダル抑制</li> </ul> <p>参照先: プログラミングマニュアル、基本編</p>
PLC ユーザプログラム:	<ul style="list-style-type: none"> <li>• DRF オフセット(軸別)の読み取り</li> </ul> <p>参照先: 『機能マニュアル、基本機能』; 「PLC 基本プログラム(P3)」</p>
HMI 操作画面:	<ul style="list-style-type: none"> <li>• DRF オフセット(軸別)の表示</li> </ul>

### 注記

DRF オフセットを削除した場合、軸は移動しません!

4.13 自動モードでのハンドルの使用

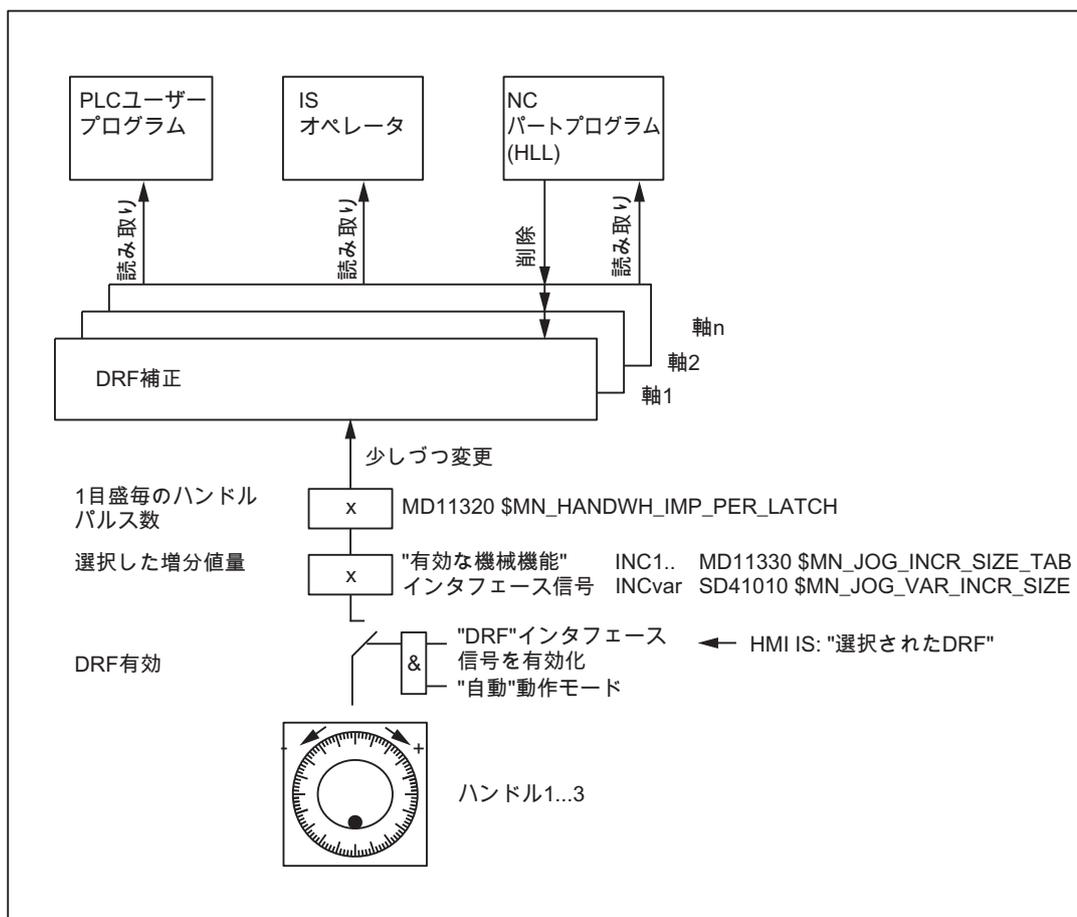


図 4-9 DRF オフセットの制御

表示

軸の現在位置の表示(ACTUAL POSITION)は、軸が DRF によってハンドルで移動している間、変更されません。現在の軸 DRF オフセットは、DRF ウィンドウで表示できます。

レファレンス点復帰

機械軸レファレンス点復帰のフェーズ 1 では、対応するジオメトリまたは補助軸の DRF オフセットが削除されます。

機械軸レファレンス点復帰の間、対応するジオメトリまたは補助軸の DRF オフセットを同時におこなうことはできません。

リセット動作

PowerOn-Reset: DRF オフセットが削除されます。

### 4.13.3.2 セットアップ

#### 機能

「DRF オフセット」機能(差動レゾルバ機能)を使用することで、電子手動パルス発生器によって、AUTOMATIC モードで、基本座標系のジオメトリ軸と補助軸へ追加のゼロオフセットシフトを設定できます。

移動対象のジオメトリ軸または補助軸へのハンドルの割り当て(つまり、DRF オフセットのステップ単位が出力されるハンドルの割り当て)は、当該の機械軸によっておこなってください。当該の機械軸は、ジオメトリ軸または補助軸が割り当てられた機械軸です。

DRF オフセットは、軸フィードバック値の画面で表示されません。

#### 用途

DRF オフセットはたとえば、次の用途で使用できます。

- NC ブロック内の工具の摩耗のオフセット  
NC ブロックの処理時間が非常に長い場合、NC ブロック内で工具摩耗のオフセットを手動で設定することが必要になります。(例: 大型正面フライス盤)
- 研削中の高精度オフセット
- 簡単な熱変位補正

---

#### 注記

DRF オフセットによって取り入れられたゼロオフセットは、常にすべてのモードで、RESET 後に有効になります。ただしこれは、パートプログラムでノンモーダルで抑制できます。

---

#### 減速

DRF の手動パルス発生器を使用して生成された速度は、ジョグ速度に関しては減速ができません。

MD32090 \$MA\_HANDWH\_VELO\_OVERLAY\_FACTOR (ハンドル速度に対するジョグ速度の比率(DRF))

4.13 自動モードでのハンドルの使用

**DRF が有効**

ハンドル移動によって DRF オフセットを変更するには、DRF を有効にしてください。  
次の必要条件を満たしてください。

- AUTOMATIC モード
- DB21、... DBX0.3 (DRF の起動) = 1

HMI 操作画面上で「プログラム制御」機能を使用して、個別のチャンネルについて DRF オフセットを有効/無効にできます。

次に HMI ソフトウェアは、インタフェース信号:  
DB21、... DBX24.3 (DRF の選択) = 1 を設定します。

対応する論理演算がおこなわれると、PLC プログラム(PLC 基本プログラムまたはユーザプログラム)はインタフェース信号  
DB21、... DBX0.3(DRF の起動)に、このインタフェース信号を伝送します。

**DRF オフセットの制御**

次のように、DRF オフセットの変更、削除、または読み取りをおこなうことができます。

<b>ユーザー:</b>	<ul style="list-style-type: none"> <li>● ハンドルによる移動</li> </ul>
<b>パートプログラム:</b>	<ul style="list-style-type: none"> <li>● 軸別システム変数\$AC_DRF[&lt;軸&gt;]による読み取り</li> <li>● チャンネルのすべての軸に対して、パートプログラム命令 (DRFOF)によって削除</li> <li>● パートプログラム命令(SUPA)によるノンモーダルの抑制</li> </ul> <p><b>参照先:</b> プログラミングマニュアル、基本編</p>
<b>PLC ユーザプログラム:</b>	<ul style="list-style-type: none"> <li>● DRF オフセット(軸別)の読み取り</li> </ul> <p><b>参照先:</b> 『機能マニュアル、基本機能』；「PLC 基本プログラム(P3)」</p>
<b>HMI 操作画面:</b>	<ul style="list-style-type: none"> <li>● DRF オフセット(軸別)の表示</li> </ul>

**注記**

DRF オフセットを削除した場合、軸は移動しません!

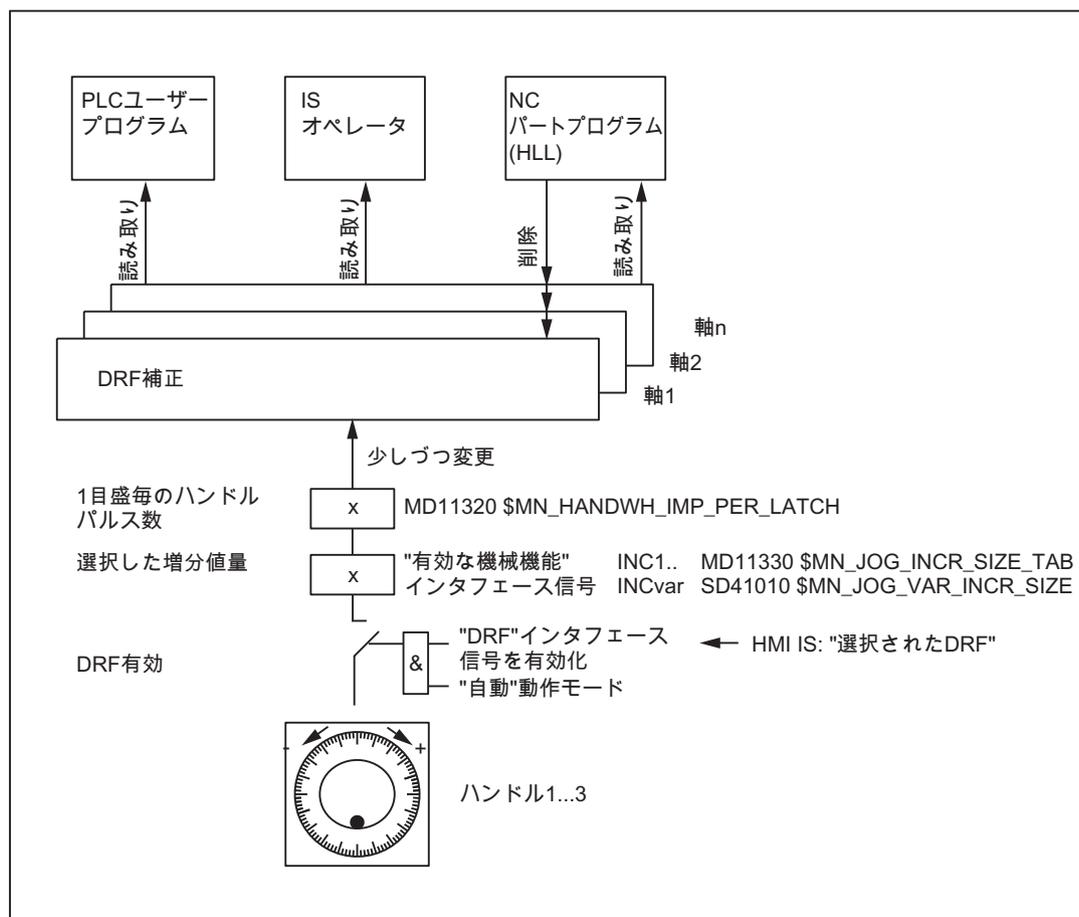


図 4-10 DRF オフセットの制御

## 表示

軸の現在位置の表示(ACTUAL POSITION)は、軸が DRF によってハンドルで移動している間、変更されません。現在の軸 DRF オフセットは、DRF ウィンドウで表示できます。

## リファレンス点復帰

機械軸リファレンス点復帰のフェーズ 1 では、対応するジオメトリまたは補助軸の DRF オフセットが削除されます。

機械軸リファレンス点復帰の間、対応するジオメトリまたは補助軸の DRF オフセットを同時におこなうことはできません。

## リセット動作

PowerOn-Reset: DRF オフセットが削除されます。

4.13 自動モードでのハンドルの使用

4.13.3.3 プログラミング軸別の重畳の解除(CORROF)

CORROF 手順により、以下の軸別重畳が解除されます:

- ハンドル移動により設定された追加ゼロオフセット(DRF オフセット)
  - \$AA\_OFF システム変数によってプログラムされた位置オフセット
  - \$AC\_OFF\_...システム変数によってプログラムされた工具オリエンテーションの重畳
- 先読み停止は、重畳値の解除によって開始され、解除された重畳動作の位置成分が基本座標系の位置に伝送されます。それによって、移動する軸はありません。

\$AA\_IM システム変数(軸の現在の MCS 指令)によって読み取ることができる位置データは、**機械**座標系で**変化しません**。

重畳動作の解除された成分が含まれるようになったために、\$AA\_IW システム変数(軸の現在の WCS 指令)によって読み取ることができる位置データは**ワーク**座標系では**変化**します。

注記

CORROF は、**NC プログラム**でプログラムできます。  
CORROF は、**シンクロナイズドアクション**ではプログラムしないでください。

構文

CORROF (<Axis>,"<String>"[,<Axis>,"<String>"])

意味

CORROF:	軸の以下のオフセットおよび重畳の解除の手順:	
	<ul style="list-style-type: none"> <li>● DRF オフセット</li> <li>● 位置オフセット(\$AA_OFF)</li> <li>● 工具オリエンテーションの重畳(\$AC_OFF_...)</li> </ul>	
	有効性:	モーダル
<軸>:	軸識別子(チャンネル、ジオメトリ軸、または機械軸の識別子)	
	データタイプ:	AXIS

<String>:	重畳タイプを定義する文字列	
データタイプ:	BOOL	
値	意味	
DRF	DRF オフセット	
AA_OFF	位置オフセット(\$AA_OFF)	
OFF_ORI	工具オリエンテーションの重畳(\$SAC_OFF...)	
	注 工具オリエンテーションの重畳の解除は、旋回軸の軸別オフセットを解除することにより実行されます。<Axis>パラメータとして、任意のチャンネル軸を指定できます。	

## 例

## 例 1:DRF オフセットの軸別解除(1)

DRF オフセットは、DRF 手動パルス発生器の移動により X 軸で生成されています。チャンネルの他の軸に対しては、DRF オフセットは機能しません。

プログラムコード	コメント
N10 CORROF(X,"DRF")	; ここでは CORROF は、DRFOF と同じ働きがあります。
...	

## 例 2:DRF オフセットの軸別解除(2)

DRF オフセットは、DRF 手動パルス発生器の移動により X 軸と Y 軸で生成されています。チャンネルの他の軸に対しては、DRF オフセットは機能しません。

プログラムコード	コメント
; X 軸の DRF オフセットのみが選択解除されます。Y 軸の DRF オフセットは保持されます	
; DRFOF では、両方のオフセットが解除されます。	
N10 CORROF(X,"DRF")	
...	

## 例 3:\$AA\_OFF 位置オフセットの軸別解除

プログラムコード	コメント
; 位置オフセット== 10 が X 軸に対して補間されます。	
N10 WHEN TRUE DO \$AA_OFF[X]=10 G4 F5	
...	

4.13 自動モードでのハンドルの使用

プログラムコード	コメント
<pre> ; X 軸の位置オフセットの選択解除: \$AA_OFF[X]=0 ; X 軸は移動しません。 ; 位置オフセットを、X 軸の現在位置に加算します。 N80 CORROF(X,"AA_OFF") ... </pre>	

**例 4 :DRF オフセットと\$AA\_OFF 位置オフセットの軸別選択解除 (1)**

DRF オフセットは、DRF 手動パルス発生器の移動により X 軸で生成されます。チャンネルの他の軸に対しては、DRF オフセットは機能しません。

プログラムコード	コメント
<pre> ; 位置オフセット 10 が X 軸に対して補間されます。 N10 WHEN TRUE DO \$AA_OFF[X]=10 G4 F5 ... ; X 軸の DRF オフセットと位置オフセットのみを選択解除します。 ; Y 軸の DRF オフセットは保持されます。 N70 CORROF(X,"DRF",X,"AA_OFF") ... </pre>	

**例 5 :DRF オフセットと\$AA\_OFF 位置オフセットの軸別選択解除 (2)**

DRF オフセットは、DRF 手動パルス発生器の移動により X 軸と Y 軸で生成されます。チャンネルの他の軸に対しては、DRF オフセットは機能しません。

プログラムコード	コメント
<pre> ; 位置オフセット == 10 が X 軸に対して補間されます。 N10 WHEN TRUE DO \$AA_OFF[X]=10 G4 F5 ... ; Y 軸の DRF オフセットと X 軸の位置オフセットのみを選択解除します。 ; X 軸の DRF オフセットは保持されます。 N70 CORROF(Y,"DRF",X,"AA_OFF") ... </pre>	

詳細情報

**\$AA\_OFF\_VAL**

位置オフセットを\$AA\_OFF で選択解除すると、対応する軸のシステム変数 \$AA\_OFF\_VAL (軸重畳の統合距離)はゼロになります。

#### JOG モードの\$AA\_OFF

JOG モードでも、\$AA\_OFF が変更されると、マシンデータ MD 36750 \$MA\_AA\_OFF\_MODE でこの機能を有効にしている場合は、位置オフセットは重畳移動として補間されます。

#### シンクロナイズドアクションの\$AA\_OFF

位置オフセットを CORROF (<軸>, "AA\_OFF") で選択解除時に、\$AA\_OFF (DO \$AA\_OFF [<軸>]=<値>)を直ちにリセットするシンクロナイズドアクションが有効である場合、\$AA\_OFF は選択解除されますが、リセットされません。また、アラーム 21660 が表示されます。ただし、CORROF の後のブロックなど、後の方のブロックでシンクロナイズドアクションが有効になる場合は、\$AA\_OFF が設定されたままとなり、位置オフセットは補間されます。

#### 自動チャネル軸入れ替え

別のチャネルで有効な軸が CORROF のプログラム指令対象である場合、その軸は、軸入れ替え時にそのチャネルに取り込まれ(条件:MD30552 \$MA\_AUTO\_GET\_TYPE > 0)、位置オフセットか DRF オフセットのいずれか、または両方が選択解除されます。

### 4.13.4 ハンドルの二重使用

#### アラーム 14320

ハンドルによって軸に及ぼす影響が異なる場合は、輪郭ハンドルを含めて、DRF と、速度または距離の重畳用ハンドルの二重使用は、自己クリアアラーム 14320(ハンドル%1 が二重(%2)にチャネル%3 の軸%4 で使用されています)を使用して表示され、抑制されます。

これは、移動に関与する基本座標系で(同じハンドルで起動した)DRF オフセットが軸に対して有効でない場合にのみ、重畳移動を実行できることを意味します。つまり、DRF の移動が終了している必要があります。

重畳移動を開始している場合は、同じハンドルによって与えられたどの関連軸に対しても DRF オフセットを開始できません。重畳を伴う移動がその終点に達した場合、または残移動距離削除または RESET によって中止された場合にのみ、このような DRF の移動が可能です。

ハンドルオーバーライドと DRF オフセットが同時に有効である場合、2 基の個別のハンドルを適用できます。

## 4.13 自動モードでのハンドルの使用

## 例: 軌跡オーバーライド

## 前提条件:

:チャンネル1とジオメトリ軸Xは機械軸3に対応し、ジオメトリ軸Yは機械軸5に対応し、第1ジオメトリ軸に対してハンドル2が選択されています。

ブロック X10 Y10 FD=0 が実行プログラムで処理される場合、機械軸3と機械軸5はともにハンドル2によってDRFでは移動できません。チャンネル別DRF信号が有効な間にハンドル2が機械軸3に割り当てられた場合、アラーム14320(チャンネル1のX軸でハンドル2が2回(8)使用されました)が通知されます。

第2ハンドルを使用して機械軸3または機械軸5をDRFで移動した場合、動作 X10 Y10 FD=0 は実行できず、アラーム14320(チャンネル1のX軸でハンドル2が2回(3)使用されました)またはアラーム14320(ハンドル2がチャンネル1のY軸で2回(3)使用されました)が通知されます。

## 例: 位置決め軸の速度オーバーライド

## 前提条件:

チャンネル1:チャンネル軸Aは機械軸4に対応し、ハンドル1がこの軸に割り当てられます。

実行プログラム POS[A]=100 FDA[A]=0 が処理される場合、機械軸4はDRFで移動できません。つまり、チャンネル別DRF信号が有効な場合、アラーム14320(チャンネル1のA軸で、ハンドル1が2回(6)使用されました)が通知されます。

DRFで機械軸4を移動する場合、DRF移動が実行されている間、POS[A]=100 FDA[A]=0の移動は実行できません。アラーム14320(チャンネル1のA軸でハンドル1が2回(1)使用されました)が通知されます。

## 例: PLC軸の距離の重畳(840D sl)

## 前提条件:

チャンネル1:ハンドル2は加工軸4に割り当てられています。

FC18で起動した第4機械軸の軌跡オーバーライドで軸移動が実行プログラムで処理される場合、第4機械軸はDRFでは移動できません。つまり、チャンネル別DRF信号が有効な場合、アラーム14320(チャンネル1のA軸で、ハンドル2が2回(9)使用されました)が通知されます。

機械軸4がDRFで移動する場合、DRF移動の実行中に、FC18で起動した軌跡オーバーライドで軸移動は実行できません。アラーム14320(チャンネル1のA軸でハンドル2が2回(4)使用されました)が通知されます。

## 4.14 監視機能

### リミット位置

手動移動および、ハンドルを使用した手動移動には、以下の制限が適用されます。

- ワーキングエリアリミット(軸を原点確立してください)。
- ソフトウェアリミットスイッチ 1 と 2(軸を原点確立してください)
- ハードウェアリミットスイッチ

この制御により、最初の有効リミット位置に到達するとすぐに移動動作が中止されます。速度制御により、リミット位置(例:ソフトウェアリミットスイッチ)で軸が正確に停止できるために、かなり早い段階で減速が開始されます。ハードウェアリミットスイッチが起動した場合にのみ、「急停止」によって瞬間的に軸停止がおこなわれます。

各種リミット位置に到達した場合にのみアラームが出力されます(アラーム 16016、16017、16020、16021)。この制御により、この方向への追加の移動は自動的に禁止されます。移動キーとハンドルは、この方向では無効です。

---

#### 注記

ソフトウェアリミットスイッチとワーキングエリアリミットは、軸が最初に原点確立されている場合にのみ有効です。

ハンドルによるゼロオフセット(DRF オフセット)が軸に有効である場合、ジョグモードでプログラム実行中にこうした軸のソフトウェアリミットスイッチが監視されます。つまり、ソフトウェアリミットスイッチへのアプローチ時には、加々速度制限は無効になります。MD32300 \$MA\_MAX\_AX\_ACCEL (最大軸加速度)に従った加速度の後、速度はソフトウェアリミットスイッチで減速します。

---

ワーキングエリアリミットとハードウェアとソフトウェアのリミットスイッチについて詳しくは、

#### 参照先:

『機能マニュアル、基本機能』、「軸監視と保護領域 (A3)」を参照してください。

## 4.15 セットアップ:ハンドル

### 軸の戻し

軸は、反対方向への移動によって、リミット位置から戻ることができます。

---

#### 注記

##### 工作機械メーカー

リミット位置にアプローチした軸を戻す機能は、工作機械メーカーに確認してください。工作機械メーカーの取扱説明書を参照してください。

---

### 最大速度と最大加速度

手動移動時に使用される速度と加速度は、セットアップエンジニアが指定軸に対してマシンデータを使用して定義します。この制御により最高速度と最大加速度指定を上限としている軸の動作値を制限します。

#### 参照先:

『機能マニュアル、基本機能』；「速度、指令値/フィードバック系、位置ループ制御 (G2)』  
『機能マニュアル、基本機能』；「加速度制御 (B2)」

## 4.15 セットアップ:ハンドル

### 4.15.1 一般情報

SINUMERIK コントロールシステムのハンドルを操作するには、NC マシンデータによってそのパラメータ設定をおこなってください。

手動パルス発生器をコントローラに直接接続しない場合、たとえば PROFIBUS- または Ethernet-MCP または手動パルス発生器モジュールによる接続 SIMATIC STEP 7、HW-Config によるモジュールの挿入と設定などの追加対策が必要です。

---

#### 注記

現在、SINUMERIK コントロールシステムで 6 基までのハンドルをパラメータ設定することができます。

---

### 接続オプション

#### SINUMERIK 840D sl

SINUMERIK 840D sl では、ハンドルは次のコンポーネントによって接続できます。

- PROFIBUS (ページ 280)モジュール
  - Ethernet (ページ 284)モジュール
- 

#### 注記

異なるコンポーネントによって接続された複数のハンドルを 1 つの SINUMERIK 840D sl に同時に接続できます。

---

### SINUMERIK 828D

SINUMERIK 828D では、ハンドルは次のコンポーネントによって接続できます。

- PPU (ページ 278)
  - PROFIBUS (ページ 279)による機械操作パネル(MCP)
- 

#### 注記

異なるコンポーネントによって接続された複数のハンドルを 1 つの SINUMERIK 828D に同時に接続できます。

---

### 4.15.2 PPU による接続 (828D のみ)

#### パラメータ設定

PPU の X143 端子に直接接続されたハンドルは、次の NC マシンデータを使用してパラメータ設定されます。

- MD11350 \$MN\_HANDWHEEL\_SEGMENT[<NCK のハンドル番号 - 1>] = 2  
PPU に直接接続する場合、手動パルス発生器セグメントとして常に 2(8xxD\_HW)を入力してください。
- MD11351 \$MN\_HANDWHEEL\_MODULE[NCK のハンドル番号 - 1>] = 1  
PPU に直接接続する場合、常に 1 を入力してください。
- MD11352 \$MN\_HANDWHEEL\_INPUT[< NCK のハンドル番号 - 1>] = <ハンドル接続>  
ハンドルの番号を入力します:1 または 2

---

#### 注記

2 基のハンドルを PPU(X143 端子)に接続できます。

---

#### 例

PPU の X143 端子に接続する 2 基の手動パルス発生器のパラメータ設定。

マシンデータ	値	意味
		第 1 ハンドル:
MD11350 \$MN_HANDWHEEL_SEGMENT[0]	2	ハードウェアセグメント:8xxD_HW
MD11351 \$MN_HANDWHEEL_MODULE[0]	1	---
MD11352 \$MN_HANDWHEEL_INPUT[0]	1	X143 の第 1 ハンドルの接続
		第 2 ハンドル:
MD11350 \$MN_HANDWHEEL_SEGMENT[1]	2	ハードウェアセグメント:8xxD_HW
MD11351 \$MN_HANDWHEEL_MODULE[1]	1	---
MD11352 \$MN_HANDWHEEL_INPUT[1]	2	X143 の第 2 ハンドルの接続

### 4.15.3 PROFIBUS による接続 (828D)

#### パラメータ設定

SINUMERIK 828D では、PPU インタフェース(X143 端子)への 2 基のハンドル接続に加えて、機械操作パネル(例: MCP 483C PN インタフェース X60)によって第 3 のハンドルを接続することもできます。

第 3 のハンドルのパラメータ設定は、次の NC マシンデータによって行います。

- MD11350 \$MN\_HANDWHEEL\_SEGMENT[ 2 ] = 5
- MD11351 \$MN\_HANDWHEEL\_MODULE[ 2 ] = 1
- MD11352 \$MN\_HANDWHEEL\_INPUT[ 2 ] = 1

#### 必要条件

初期設定データ(マシンデータ、STEP 7 の設定)による制御操作。

#### 例

PPU および「MCP 483C PN」機械操作パネルによって接続された 3 基のハンドルのパラメータ設定。

NC の ハンドル番号	マシンデータセット (インデックス)	接続
1	0	PPU, 1.番目のハンドルスロットのハンドル
2	1	PPU, 2.番目のハンドルスロットのハンドル
3	2	MCP, 1.番目のハンドルスロットのハンドル

NC マシンデータのパラメータ設定:

マシンデータ	規格 値	意味
		NC の第 1 ハンドル
MD11350 \$MN_HANDWHEEL_SEGMENT[0]	2	ハードウェアセグメント:8xxD_HW
MD11351 \$MN_HANDWHEEL_MODULE[0]	1	---
MD11352 \$MN_HANDWHEEL_INPUT[0]	1	ハンドルスロットの第 1 ハンドル

## 4.15 セットアップ:ハンドル

マシンデータ	規格値	意味
		NC の第 2 ハンドル
MD11350 \$MN_HANDWHEEL_SEGMENT[1]	2	ハードウェアセグメント:8xxD_HW
MD11351 \$MN_HANDWHEEL_MODULE[1]	1	---
MD11352 \$MN_HANDWHEEL_INPUT[1]	2	ハンドルスロットの第 2 ハンドル
		NC の第 3 ハンドル
MD11350 \$MN_HANDWHEEL_SEGMENT[2]	5	ハードウェアセグメント:PROFIBUS
MD11351 \$MN_HANDWHEEL_MODULE[2]	1	MCP のハンドルスロットの論理基準アドレス参照先
MD11352 \$MN_HANDWHEEL_INPUT[2]	1	ハンドルスロットの第 1 ハンドル

## 4.15.4 PROFIBUS による接続(840D sl)

## パラメータ設定

PROFIBUS モジュール(例: 「MCP 483」 機械操作パネル)によって接続されたハンドルのパラメータ設定は、次の NC マシンデータでおこないます。

- MD11350 \$MN\_HANDWHEEL\_SEGMENT[<NCK のハンドル番号 - 1>] = 5  
PROFIBUS モジュールによって接続する場合は、ハードウェアセグメントは常に 5(PROFIBUS)で入力してください。
- MD11351 \$MN\_HANDWHEEL\_MODULE[<NCK のハンドル番号 - 1>] = <インデックス + 1>  
MD11353 \$MN\_HANDWHEEL\_LOGIC\_ADDRESS[<インデックス>]の参照先 (ハンドルの論理基準アドレスを含みます)を入力してください。

- MD11352 \$MN\_HANDWHEEL\_INPUT[<NCK のハンドル番号 - 1>] = <ハンドルスロット番号>  
ハンドルスロットには、複数のハンドルを格納できます。ハンドルスロット内のハンドル番号を入力してください。1, 2, ...
- MD11353 \$MN\_HANDWHEEL\_LOGIC\_ADDRESS[<インデックス>] = <論理基準アドレス>  
SIMATIC STEP 7 の HW 構成設定で指定したハンドルスロットの論理基準アドレスを入力してください。

### ハンドルスロット

STEP 7 の NC マシンデータのハンドルのパラメータ設定に加えて、PROFIBUS モジュールを設定する必要があります。特に、ハンドルスロットの論理アドレスを指定します。ハンドルスロットは、次の PROFIBUS モジュールのスロットにあります。

PROFIBUS モジュール	スロット
機械操作パネル MCP 483	2
機械操作パネル MCP 310	2
ハンドル接続モジュール	1

### 例

4 つの機械操作パネル「MCP 483」によって接続された 5 基のハンドルのパラメータ設定。2 基のハンドルを 1 つの機械操作パネル「MCP 483」に接続できます。

NC の ハンドル番号	マシンデータセット (インデックス)	接続
1	0	第 1MCP、1.番目のハンドルスロットのハンドル
2	1	第 1MCP、2.番目のハンドルスロットのハンドル
3	2	第 2MCP、1.番目のハンドルスロットのハンドル
5	4	第 3MCP、1.番目のハンドルスロットのハンドル
6	5	第 4MCP、2.番目のハンドルスロットのハンドル

4.15 セットアップ:ハンドル

NC の第 4 ハンドルは使用されません(マシンデータの隙間部分)。

注記

NC マシンデータでハンドルのパラメータ設定を行う場合、マシンデータの隙間が許されます。

機械操作パネルは、SIMATIC STEP 7 の HW 構成設定で次のように設定されています。

	スロット	DP ID	品番/名称	Iアドレス	Oアドレス
第 1MCP	1	55	標準+ハンドル	0 ... 7	0 ... 7
	2	2AE	→ 標準+ハンドル	288 ... 291	
	3	1	→ 標準+ハンドル		
第 2MCP	1	55	標準+ハンドル	8 ... 15	8 ... 15
	2	2AE	→ 標準+ハンドル	304 ... 307	
	3	1	→ 標準+ハンドル		
第 3MCP	1	55	標準+ハンドル	16 ... 23	16 ... 23
	2	2AE	→ 標準+ハンドル	320 ... 323	
	3	1	→ 標準+ハンドル		
第 4MCP	1	55	標準+ハンドル	24 ... 29	24 ... 29
	2	2AE	→ 標準+ハンドル	330 ... 333	
	3	1	→ 標準+ハンドル		

NC マシンデータのパラメータ設定:

マシンデータ	規格値	意味
		NC の第 1 ハンドル
MD11350 \$MN_HANDWHEEL_SEGMENT[0]	5	ハードウェアセグメント:PROFIBUS
MD11351 \$MN_HANDWHEEL_MODULE[0]	1	第 1MCP のハンドルスロットの論理基準アドレス参照先
MD11352 \$MN_HANDWHEEL_INPUT[0]	1	ハンドルスロットの第 1 ハンドル

マシンデータ	規格値	意味
		NC の第 2 ハンドル
MD11350 \$MN_HANDWHEEL_SEGMENT[1]	5	ハードウェアセグメント:PROFIBUS
MD11351 \$MN_HANDWHEEL_MODULE[1]	1	第 1MCP のハンドルスロットの論理基準アドレス参照先
MD11352 \$MN_HANDWHEEL_INPUT[1]	2	ハンドルスロットの第 2 ハンドル
		NC の第 3 ハンドル
MD11350 \$MN_HANDWHEEL_SEGMENT[2]	5	ハードウェアセグメント:PROFIBUS
MD11351 \$MN_HANDWHEEL_MODULE[2]	2	第 2MCP のハンドルスロットの論理基準アドレス参照先
MD11352 \$MN_HANDWHEEL_INPUT[2]	1	ハンドルスロットの第 1 ハンドル
		NC の第 4 ハンドル
MD11350 \$MN_HANDWHEEL_SEGMENT[3]	0	ハンドルがパラメータ設定されていません
MD11351 \$MN_HANDWHEEL_MODULE[3]	0	ハンドルがパラメータ設定されていません
MD11352 \$MN_HANDWHEEL_INPUT[3]	0	ハンドルがパラメータ設定されていません
		NC の第 5 ハンドル
MD11350 \$MN_HANDWHEEL_SEGMENT[4]	5	ハードウェアセグメント:PROFIBUS
MD11351 \$MN_HANDWHEEL_MODULE[4]	6	第 3MCP のハンドルスロットの論理基準アドレス参照先
MD11352 \$MN_HANDWHEEL_INPUT[4]	1	ハンドルスロットの第 1 ハンドル
		NC の第 6 ハンドル
MD11350 \$MN_HANDWHEEL_SEGMENT[5]	5	ハードウェアセグメント:PROFIBUS

## 4.15 セットアップ:ハンドル

マシンデータ	規格値	意味
MD11351 \$MN_HANDWHEEL_MODULE[5]	5	第 4MCP のハンドルスロットの論理基準アドレス参照先
MD11352 \$MN_HANDWHEEL_INPUT[5]	2	ハンドルスロットの第 2 ハンドル

論理基準アドレス:

マシンデータ	規格値	意味
MD11353 \$MN_HANDWHEEL_LOGIC_ADDRESS [0]	288	第 1MCP のハンドルスロットの論理基準アドレス
MD11353 \$MN_HANDWHEEL_LOGIC_ADDRESS [1]	304	第 2MCP のハンドルスロットの論理基準アドレス
MD11353 \$MN_HANDWHEEL_LOGIC_ADDRESS [4]	330	第 4MCP のハンドルスロットの論理基準アドレス
MD11353 \$MN_HANDWHEEL_LOGIC_ADDRESS [5]	320	第 3MCP のハンドルスロットの論理基準アドレス

## 4.15.5 Ethernet を介した接続 - (840D sl のみ)

## パラメータの割り付け

Ethernet モジュール(たとえば機械操作パネル「MCP 483C IE」、「HT 8」または「HT 2」)を介して接続されるハンドルのパラメータは、以下の NC マシンデータに割り当てられます。

- MD11350 \$MN\_HANDWHEEL\_SEGMENT[< x - 1 >] = 7  
Ethernet モジュールを介して接続する場合、常に、セグメントとして 7 を入力してください(Ethernet)。
- MD11351 \$MN\_HANDWHEEL\_MODULE[< x - 1 >] = 1  
Ethernet モジュールを介して接続する場合、常に、モジュールとして 1 を入力してください。
- MD11352 \$MN\_HANDWHEEL\_INPUT[< x - 1 >] = y  
ここで、y = 1、2、3、... (Ethernet バスのハンドルインタフェース)

ここで、x = 1、2、3、... (NC 内のハンドル番号)

## Ethernet バスのハンドルインタフェース

Ethernet バスのハンドルインタフェースには、以下の定義に基づいて番号が付けられます。

- オペレータコンポーネントインタフェースの順序は次のとおりです。MCP1、MCP2、BHG
- 各オペレータコンポーネントインタフェースには2つのハンドルインタフェースが付いています。

- オペレータコンポーネント:MCP 483C IE

最大2つのハンドルを接続部 MCP 483C IE へ X60 と X61 を経由して接続することができます。オペレータコンポーネントインタフェースの接続部の割り当て:

- 接続部 X60:オペレータコンポーネントインタフェース MCP1 /MCP2 の第1ハンドル
- 接続部 X61:オペレータコンポーネントインタフェース MCP1 /MCP2 の第2ハンドル

- オペレータコンポーネント:HT 8

HT 8 のハンドルは、常に、オペレータコンポーネントインタフェース MCP1 /MCP2 の第1のハンドルに割り当てられます。

- オペレータコンポーネント:HT 2

HT 2 のハンドルは常に、オペレータコンポーネントインタフェース BHG の第1のハンドルに割り当てられます。

オペレータコンポーネントインタフェース ->	MCP1		MCP2		BHG	
ハンドルインタフェース <sup>1)</sup>	1	2	1	2	1	2
FB1 パラメータ <sup>2)</sup>	MCP1BusAdr		MCP2BusAdr		BHGRecGDNo	
ハンドルの割り当て <sup>3)</sup>						
MCP 483C IE	X60	X61	X60	X61	-	-
HT 8	x	-	x	-	-	-
HT 2	-	-	-	-	x	-

4.15 セットアップ:ハンドル

オペレータコンポーネントインタフェース ->	MCP1		MCP2		BHG	
Ethernet バスのハンドルインタフェース (y) 4) ->	1	2	3	4	5	6
1) オペレータコンポーネントインタフェースのハンドルインタフェースの番号付け 2) 対応する FB1 パラメータによる、オペレータコンポーネントのインタフェースへの割り当て 3) 各オペレータコンポーネントハンドルのハンドルインタフェースへの割り当て 4) Ethernet バスのハンドルインタフェースの番号付け -> MD11352 \$MN_HANDWHEEL_INPUT[ x - 1 > ] = y						

例

以下のオペレータコンポーネントを経由して接続された 3 つのハンドルのパラメータ設定

オペレータコンポーネントインタフェース ->	MCP1		MCP2		BHG	
オペレータコンポーネント	HT 8		MCP 483C		HT 2	
FB1 パラメータ	MCP1BusAdr := 39		MCP2BusAdr := 192		BHGRcGDNo := 40	
ハンドルインタフェース	x	-	-	X61	x	-
Ethernet バスのハンドルインタフェース ->	1	2	3	4	5	6

表 4-1 ハンドル割り当て用の NC マシンデータ

マシンデータ	規格値	説明
		HT 8:NC のハンドル番号= 1
MD11350 \$MN_HANDWHEEL_SEGMENT[0]	7	セグメント:Ethernet
MD11350 \$MN_HANDWHEEL_MODULE[ 0 ]	1	モジュール:Ethernet
MD11350 \$MN_HANDWHEEL_INPUT[ 0 ]	1	Ethernet バスのハンドルインタフェース
		MCP 483C IE:NC のハンドル番号= 2
MD11350 \$MN_HANDWHEEL_SEGMENT[1]	7	セグメント:Ethernet

マシンデータ	規格値	説明
MD11350 \$MN_HANDWHEEL_MODULE [ 1 ]	1	モジュール:Ethernet
MD11350 \$MN_HANDWHEEL_INPUT [ 1 ]	4	Ethernet バスのハンドルインタフェース
		HT 2:NC のハンドル番号= 3
MD11350 \$MN_HANDWHEEL_SEGMENT[2]	7	セグメント:Ethernet
MD11350 \$MN_HANDWHEEL_MODULE [ 2 ]	1	モジュール:Ethernet
MD11350 \$MN_HANDWHEEL_INPUT [ 2 ]	5	Ethernet バスのハンドルインタフェース

表 4-2 FB1 パラメータ(抜粋)

パラメータ	規格値	備考
MCPNum	:= 2	// 接続されている MCP 数 // MCP1 = HT 8
MCP1In	...	// MCP1 パラメータ...
...	...	
MCP1BusAdr	:= 39	// 接続機器のスイッチ S1 と S2 を使用 // 「IP アドレス」の設定 // MCP2 = MCP 483C IE
MCP2In	...	// MCP2 パラメータ...
...	...	
MCP2BusAdr	:= 192	// MCP 483C のスイッチ S2 を使用 // 「IP アドレス」の設定
MCPBusType	:= b#16#55	// バスタイプ:Ethernet // HHU = HT 2
HHU	:= 5	// バスタイプ:Ethernet
BHGIn	...	// HHU パラメータ ...
...	...	
BHGRecGDNo	:= 40	// 接続機器のスイッチ S1 と S2 を使用 // 「IP アドレス」の設定

4.15 セットアップ:ハンドル

フィルタ時間

Ethernet バスでは手動パルス発生器のパルスは確定されたものとして伝送されないため、高度のダイナミック特性を備えたドライブ装置では、手動パルス発生器のパルス伝送処理のフィルタリング(スムージング)が必要になることがあります。フィルタ時間は、次のマシンデータを使用してパラメータ設定します。

- MD11354 \$MN\_HANDWHEEL\_FILTER\_TIME[< x - 1 >] = <フィルタ時間>  
ここで、x = 1、2、3、... (NC のハンドル番号)とフィルタ時間 = 0.0 ... 2.0 s

フィルタ時間は、制御装置に送られる手動パルス発生器のパルスが、移動のために補間器に送られるのに要する時間を指定します。フィルタ時間が 0.0 s の場合、手動パルス発生器のパルスは 1 補間周期内に補間器に送られます。このために、軸の移動がぎこちなくなる場合があります。

推奨フィルタ時間:0.2 ~ 0.5 s

ゼロ速度検出

停止状態は手動パルス発生器が接続されている Ethernet モジュールによって検出されます。手動パルス発生器が設定時間内にパルスを転送しなかった場合、モジュールはこれを停止状態として検出し、これを NC/PLC インタフェースに転送します。

NC/PLC インタフェース信号	規格値	意味
DB10、DBX245.0	0	ハンドル 1 が操作されています。
	1	ハンドル 1 が停止しています
DB10 DBX245.1	0	ハンドル 2 が操作されています。
	1	ハンドル 2 が停止しています。
DB10 DBX245.2	0	ハンドル 3 が操作されています。
	1	ハンドル 3 が停止しています。
DB10 DBX245.3	0	ハンドル 4 が操作されています。
	1	ハンドル 4 が停止しています。
DB10 DBX245.4	0	ハンドル 5 が操作されています。
	1	ハンドル 5 が停止しています。
DB10 DBX245.5	0	ハンドル 6 が操作されています。
	1	ハンドル 6 が停止しています。

この信号を使用することで、制御装置は受け取っても、まだ移動用の補間器に伝送されない手動パルス発生器のパルスによって、手動パルス発生器による移動軸のオーバート

ラベルを低減することができます。このためには、停止状態が検出されたとき、対応する軸に対してまたはチャンネル内で、残移動距離の削除を実施してください。

- DB31、...DBX2.2 = 1 (軸の残移動距離削除)
- DB21、...DBX6.2 = 1(チャンネル別の残移動距離削除)

## 4.16 データリスト

### 4.16.1 マシンデータ

#### 4.16.1.1 一般マシンデータ

番号	識別子: \$MN_	説明
10000	AXCONF_MACHAX_NAME_TAB[n]	機械軸名称
10720	OPERATING_MODE_DEFAULT	電源投入後の運転モードの設定
10721	OPERATING_MODE_EXTENDED	電源投入後の運転モードの拡張設定
10735	JOG_MODE_MASK	ジョグモードの設定
11300	JOG_INC_MODE_LEVELTRIGGRD	ジョグモードの INC と REF
11310	HANDWH_REVERSE	反対方向への移動の定義
11320	HANDWH_IMP_PER_LATCH[n]	1目盛毎のハンドルパルス
11324	HANDWH_VDI_REPRESENTATION	ハンドル番号の割り当て(NCK/PLC インタフェース)
11330	JOG_INCR_SIZE_TAB[n]	INC/ハンドルのステップ量
11340	ENC_HANDWHEEL_SEGMENT_NR	第3ハンドル:バスセグメント
11342	ENC_HANDWHEEL_MODULE_NR	第3ハンドル:論理ドライブ番号
11344	ENC_HANDWHEEL_INPUT_NR	第3ハンドル:エンコーダインタフェース
11346	HANDWH_TRUE_DISTANCE	ハンドル軌跡または速度指定
11350	HANDWHEEL_SEGMENT[n]	ハンドルセグメント
11351	HANDWHEEL_MODULE[n]	ハンドルモジュール
11352	HANDWHEEL_INPUT[n]	ハンドル接続
11353	HANDWHEEL_LOGIC_ADDRESS[n]	論理ハンドルスロットアドレス(STEP 7)
17900	VDI_FUNCTION_MASK	VDI信号の機能マスク

## 4.16 データリスト

## 4.16.1.2 チャネルマシンデータ

番号	識別子: \$MC_	説明
20060	AXCONF_GEOAX_NAME_TAB	チャネルのジオメトリ軸名称
20100	DIAMETER_AX_DEF	ジオメトリ軸で径方向軸機能を使用
20110	RESET_MODE_MASK	リセット/プログラム終了後の初期設定
20150	GCODE_RESET_VALUES	Gグループのリセット設定
20151	GCODE_RESET_MODE	Gグループのリセット応答
20360	TOOL_PARAMETER_DEF_MASK	工具パラメータの定義
20620	HANDWH_GEOAX_MAX_INCR_SIZE	ジオメトリ軸のリミット位置
20622	HANDWH_GEOAX_MAX_INCR_VSIZE	軌跡速度オーバーライド
20624	HANDWH_CHAN_STOP_COND	ハンドルを使用した手動移動の特性の定義、チャネル別
20700	REFP_NC_START_LOCK	レファレンス点なしでNCスタート無効
21150	JOG_VELO_RAPID_ORI	旋回軸用手動早送り
21155	JOG_VELO_ORI	旋回軸の手動移動速度
21158	JOG_JERK_ORI	旋回軸の手動移動時の最大加々速度
21159	JOG_JERK_ORI_ENABLE	有効な旋回軸の手動移動の加々速度制限
21160	JOG_VELO_RAPID_GEO	ジオメトリ軸用手動早送り
21165	JOG_VELO_GEO	ジオメトリ軸用手動速度
21166	JOG_ACCEL_GEO	ジオメトリ軸の手動移動時の最大加速度
21168	JOG_JERK_GEO	ジオメトリ軸の手動移動時の最大加々速度

## 4.16.1.3 軸/主軸別マシンデータ

番号	識別子: \$MA_	説明
30450	IS_CONCURRENT_POS_AX	リセット時の初期設定:中立軸またはチャネル軸
30600	FIX_POINT_POS[n]	軸の固有位置
30610	NUM_FIX_POINT_POS	軸の固有位置の数
31090	JOG_INCR_WEIGHT	INC/ハンドルのステップ値の倍率
32000	MAX_AX_VELO	最大軸速度
32010	JOG_VELO_RAPID	ジョグモードの早送り

番号	識別子: \$MA_	説明
32020	JOG_VELO	ジョグモードでの軸速度
32040	JOG_REV_VELO_RAPID	早送りオーバライドによるジョグモードの毎回転送り速度
32050	JOG_REV_VELO	ジョグの毎回転送り速度
32060	POS_AX_VELO	位置決め軸速度のリセット設定
32080	HANDWH_MAX_INCR_SIZE	選択ステップ量の制限
32082	HANDWH_MAX_INCR_VELO_SIZE	速度オーバライドの選択ステップ値の制限
32084	HANDWH_STOP_COND	特性、ハンドル運転
32090	HANDWH_VELO_OVERLAY_FACTO R	(DRF による)ハンドル速度に対するジョグ速度の比率
32300	MAX_AX_ACCEL	最高軸加速度
32420	JOG_AND_POS_JERK_ENABLE	軸別の加々速度制限の初期設定
32430	JOG_AND_POS_MAX_JERK	単独軸移動の最大軸別加々速度
32431	MAX_AX_JERK	軌跡移動中の最高軸別加々速度
34210	ENC_REFP_STATE	検出器の状態
35130	GEAR_STEP_MAX_VELO_LIMIT[n]	ギヤ選択用の最大速度

## 4.16.2 セッティングデータ

### 4.16.2.1 一般セッティングデータ

番号	識別子: \$SN_	説明
41010	JOG_VAR_INCR_SIZE	ステップ手動移動、またはハンドルを使用した手動移動の可変ステップ量のサイズ
41050	JOG_CONT_MODE_LEVELTRIGGRD	連続手動移動のジョグ/連続モード
41100	JOG_REV_IS_ACTIVE	JOG:毎回転/毎分送り
41110	JOG_SET_VELO	ジョグモードでの軸速度
41120	JOG_REV_SET_VELO	ジョグモードの軸の毎回転送り速度
41130	JOG_ROT_AX_SET_VELO	ジョグモードの回転軸の軸速度
41200	JOG_SPIND_SET_VELO	ジョグモードの主軸の速度

## 4.16 データリスト

## 4.16.2.2 チャネルセッティングデータ

番号	識別子: \$SC_	説明
42690	JOG_CIRCLE_CENTRE	円の中心点
42691	JOG_CIRCLE_RADIUS	円弧半径
42692	JOG_CIRCLE_MODE	ジョグ円モード
42693	JOG_CIRCLE_START_ANGLE	ジョグでの円移動の開始角度
42694	JOG_CIRCLE_END_ANGLE	ジョグでの円移動の終了角度

## 4.16.2.3 軸/主軸セッティングデータ

番号	識別子: \$SA_	説明
43320	JOG_POSITION	ジョグ位置

## 4.16.3 信号

## 4.16.3.1 NC からの信号

信号名称	SINUMERIK 840D sl	SINUMERIK 828D
ハンドル 1 が操作されています	DB10.DBB68	DB2700.DBB12
ハンドル 2 が操作されています	DB10.DBB69	DB2700.DBB13
ハンドル 3 が操作されています	DB10.DBB70	DB2700.DBB14
ハンドル 4 が操作されています	DB10.DBB242	-
ハンドル 5 が操作されています	DB10.DBB243	-
ハンドル 6 が操作されています	DB10.DBB244	-
Ethernet 手動パルス発生器が停止しています	DB10.DBX245.0 - 5	-

## 4.16.3.2 HMI からの信号

信号名称	SINUMERIK 840D sl	SINUMERIK 828D
ジオメトリ軸のチャンネル番号、ハンドル 1	DB10.DBX97.0 - 3	
ジオメトリ軸のチャンネル番号、ハンドル 2	DB10.DBX98.0 - 3	
ジオメトリ軸のチャンネル番号、ハンドル 3	DB10.DBX99.0 - 3	
ハンドル 1 の軸番号	DB10.DBX100.0 - 4	
輪郭ハンドルとしてハンドル 1 を定義	DB10.DBX100.5	
ハンドル 1 が選択されました	DB10.DBX100.6	
ハンドル 1 の機械軸	DB10.DBX100.7	
ハンドル 2 の軸番号	DB10.DBX101.0 - 4	
輪郭ハンドルとしてハンドル 2 を定義	DB10.DBX101.5	
ハンドル 2 が選択されました	DB10.DBX101.6	
ハンドル 2 の機械軸	DB10.DBX101.7	
ハンドル 3 の軸番号	DB10.DBX102.0 - 4	
輪郭ハンドルとしてハンドル 3 を定義	DB10.DBX102.5	
ハンドル 3 が選択されました	DB10.DBX102.6	
ハンドル 3 の機械軸	DB10.DBX102.7	

## 4.16.3.3 モードグループの信号

信号名称	SINUMERIK 840D sl	SINUMERIK 828D
モードグループ 1: JOG モード	DB11.DBX0.2	DB3000.DBX0.2
モードグループ 2: JOG モード	DB11.DBX20.2	-
モードグループ 3: JOG モード	DB11.DBX40.2	-
モードグループ 4: JOG モード	DB11.DBX60.2	-
モードグループ 5: JOG モード	DB11.DBX80.2	-
モードグループ 6: JOG モード	DB11.DBX100.2	-
モードグループ 7: JOG モード	DB11.DBX120.2	-
モードグループ 8: JOG モード	DB11.DBX140.2	-
モードグループ 9: JOG モード	DB11.DBX160.2	-
モードグループ 10: JOG モード	DB11.DBX180.2	-

## 4.16 データリスト

## 4.16.3.4 モードグループからの信号

信号名称	SINUMERIK 840D sl	SINUMERIK 828D
モードグループ 1:アクティブモード JOG	DB11.DBX6.2	DB3100.DBX0.2
モードグループ 2:アクティブモード JOG	DB11.DBX26.2	-
モードグループ 3:アクティブモード JOG	DB11.DBX46.2	-
モードグループ 4:アクティブモード JOG	DB11.DBX66.2	-
モードグループ 5:アクティブモード JOG	DB11.DBX86.2	-
モードグループ 6:アクティブモード JOG	DB11.DBX106.2	-
モードグループ 7:アクティブモード JOG	DB11.DBX126.2	-
モードグループ 8:アクティブモード JOG	DB11.DBX146.2	-
モードグループ 9:アクティブモード JOG	DB11.DBX166.2	-
モードグループ 10:アクティブモード JOG	DB11.DBX186.2	-

## 4.16.3.5 チャネルへの信号

信号名称	SINUMERIK 840D sl	SINUMERIK 828D
DRF の起動	DB21、... .DBX0.3	DB320x.DBX0.3
ジオメトリ軸 1:		
ハンドルの起動	DB21、... .DBX12.0 - 2	DB320x.DBX1000.0 - 2
移動キーの禁止	DB21、... .DBX12.4	DB320x.DBX1000.4
早送りオーバライド	DB21、... .DBX12.5	DB320x.DBX1000.5
移動キー「マイナス」	DB21、... .DBX12.6	DB320x.DBX1000.6
移動キー「プラス」	DB21、... .DBX12.7	DB320x.DBX1000.7
要求の機械機能	DB21、... .DBX13.0 - 6	DB320x.DBX1001.0 - 6
手動パルス発生器の回転方向 の反転	DB21、... .DBX15.0	DB320x.DBX1003.0

信号名称		SINUMERIK 840D sl	SINUMERIK 828D
ジオメトリ軸 2:	ハンドルの起動	DB21、... .DBX16.0 - 2	DB320x.DBX1004.0 - 2
	移動キーの禁止	DB21、... .DBX16.4	DB320x.DBX1004.4
	早送りオーバーライド	DB21、... .DBX16.5	DB320x.DBX1004.5
	移動キー「マイナス」	DB21、... .DBX16.6	DB320x.DBX1004.6
	移動キー「プラス」	DB21、... .DBX16.7	DB320x.DBX1004.7
	要求の機械機能	DB21、... .DBX17.0 - 6	DB320x.DBX1005.0 - 6
	手動パルス発生器の回転方向 の反転	DB21、... .DBX19.0	DB320x.DBX1007.0
ジオメトリ軸 3:	ハンドルの起動	DB21、... .DBX20.0 - 2	DB320x.DBX1008.0 - 2
	移動キーの禁止	DB21、... .DBX20.4	DB320x.DBX1008.4
	早送りオーバーライド	DB21、... .DBX20.5	DB320x.DBX1008.5
	移動キー「マイナス」	DB21、... .DBX20.6	DB320x.DBX1008.6
	移動キー「プラス」	DB21、... .DBX20.7	DB320x.DBX1008.7
	要求の機械機能	DB21、... .DBX21.0 - 6	DB320x.DBX1009.0 - 6
	手動パルス発生器の回転方向 の反転	DB21、... .DBX23.0	DB320x.DBX1011.0
輪郭ハンドルの起動	DB21、... .DBX30.0 - 2	DB320x.DBX14.0 - 2	
シミュレーション輪郭ハンドル:起動	DB21、... .DBX30.3	DB320x.DBX14.3	
シミュレーション輪郭ハンドル:逆方向	DB21、... .DBX30.4	DB320x.DBX14.4	
ジョグ円移動	DB21、... .DBX30.6	DB320x.DBX14.6	
輪郭ハンドルの回転方向の反転	DB21、... .DBX31.5	DB320x.DBX15.5	
旋回軸 1:	ハンドルの起動	DB21、... .DBX320.0 - 2	-
	移動キーの禁止	DB21、... .DBX320.4	-
	早送りオーバーライド	DB21、... .DBX320.5	-
	移動キー「マイナス」	DB21、... .DBX320.6	-
	移動キー「プラス」	DB21、... .DBX320.7	-
	要求の機械機能	DB21、... .DBX321.0 - 6	-
	手動パルス発生器の回転方向 の反転	DB21、... .DBX323.0	-

## 4.16 データリスト

信号名称	SINUMERIK 840D sl	SINUMERIK 828D
旋回軸 2:		
ハンドルの起動	DB21、... .DBX324.0 - 2	-
移動キーの禁止	DB21、... .DBX324.4	-
早送りオーバーライド	DB21、... .DBX324.5	-
移動キー「マイナス」	DB21、... .DBX324.6	-
移動キー「プラス」	DB21、... .DBX324.7	-
要求の機械機能	DB21、... .DBX325.0 - 6	-
手動パルス発生器の回転方向 の反転	DB21、... .DBX327.0	-
旋回軸 3:		
ハンドルの起動	DB21、... .DBX328.0 - 2	-
移動キーの禁止	DB21、... .DBX328.4	-
早送りオーバーライド	DB21、... .DBX328.5	-
移動キー「マイナス」	DB21、... .DBX328.6	-
移動キー「プラス」	DB21、... .DBX328.7	-
要求の機械機能	DB21、... .DBX329.0 - 6	-
手動パルス発生器の回転方向 の反転	DB21、... .DBX331.0	-

## 4.16.3.6 チャンネルからの信号

信号名称	SINUMERIK 840D sl	SINUMERIK 828D
DRF 選択	DB21、... .DBX24.3	DB170x.DBX0.3
ハンドルオーバーライド有効(軌跡軸)	DB21、... .DBX33.3	DB330x.DBX1.3
輪郭ハンドル有効	DB21、... .DBX37.0-2	DB330x.DBX5.0-2
輪郭ハンドル:手動パルス発生器の回転方向の反転 が有効	DB21、... .DBX39.5	DB330x.DBX7.5

信号名称		SINUMERIK 840D sl	SINUMERIK 828D
ジオメトリ軸 1:	ハンドル有効	DB21、... .DBX40.0 - 2	DB330x.DBX1000.0 - 1
	移動要求「マイナス」	DB21、... .DBX40.4	DB330x.DBX1000.4
	移動要求「プラス」	DB21、... .DBX40.5	DB330x.DBX1000.5
	移動指令「マイナス」	DB21、... .DBX40.6	DB330x.DBX1000.6
	移動指令「プラス」	DB21、... .DBX40.7	DB330x.DBX1000.7
	有効機械機能	DB21、... .DBX41.0 - 6	DB330x.DBX1001.0 - 6
	手動パルス発生器の回転方向 の反転が有効	DB21、... .DBX43.0	DB330x.DBX1003.0
ジオメトリ軸 2:	ハンドル有効	DB21、... .DBX46.0 - 2	DB330x.DBX1004.0 - 1
	移動要求「マイナス」	DB21、... .DBX46.4	DB330x.DBX1004.4
	移動要求「プラス」	DB21、... .DBX46.5	DB330x.DBX1004.5
	移動指令「マイナス」	DB21、... .DBX46.6	DB330x.DBX1004.6
	移動指令「プラス」	DB21、... .DBX46.7	DB330x.DBX1004.7
	有効機械機能	DB21、... .DBX47.0 - 6	DB330x.DBX1005.0 - 6
	手動パルス発生器の回転方向 の反転が有効	DB21、... .DBX49.0	DB330x.DBX1007.0
ジオメトリ軸 3:	ハンドル有効	DB21、... .DBX52.0 - 2	DB330x.DBX1008.0 - 1
	移動要求「マイナス」	DB21、... .DBX52.4	DB330x.DBX1008.4
	移動要求「プラス」	DB21、... .DBX52.5	DB330x.DBX1008.5
	移動指令「マイナス」	DB21、... .DBX52.6	DB330x.DBX1008.6
	移動指令「プラス」	DB21、... .DBX52.7	DB330x.DBX1008.7
	有効機械機能	DB21、... .DBX53.0 - 6	DB330x.DBX1009.0 - 6
	手動パルス発生器の回転方向 の反転が有効	DB21、... .DBX55.0	DB330x.DBX1011.0

## 4.16 データリスト

信号名称	SINUMERIK 840D sl	SINUMERIK 828D	
旋回軸 1;	ハンドル有効	DB21、... .DBX332.0 - 2	-
	移動要求「マイナス」	DB21、... .DBX332.4	-
	移動要求「プラス」	DB21、... .DBX332.5	-
	移動指令「マイナス」	DB21、... .DBX332.6	-
	移動指令「プラス」	DB21、... .DBX332.7	-
	有効機械機能	DB21、... .DBX333.0 - 6	-
	手動パルス発生器の回転方向 の反転が有効	DB21、... .DBX335.0	-
旋回軸 2;	ハンドル有効	DB21、... .DBX336.0 - 2	-
	移動要求「マイナス」	DB21、... .DBX336.4	-
	移動要求「プラス」	DB21、... .DBX336.5	-
	移動指令「マイナス」	DB21、... .DBX336.6	-
	移動指令「プラス」	DB21、... .DBX336.7	-
	有効機械機能	DB21、... .DBX337.0 - 6	-
	手動パルス発生器の回転方向 の反転が有効	DB21、... .DBX339.0	-
旋回軸 3;	ハンドル有効	DB21、... .DBX340.0 - 2	-
	移動要求「マイナス」	DB21、... .DBX340.4	-
	移動要求「プラス」	DB21、... .DBX340.5	-
	移動指令「マイナス」	DB21、... .DBX340.6	-
	移動指令「プラス」	DB21、... .DBX340.7	-
	有効機械機能	DB21、... .DBX341.0 - 6	-
	手動パルス発生器の回転方向 の反転が有効	DB21、... .DBX343.0	-
ジョグ後退有効	DB21、... .DBX377.4	DB330x.DBX4005.4	
ジョグ後退の後退データが使用可能	DB21、... .DBX377.5	DB330x.DBX4005.5	
ジョグ円移動が有効	DB21、... .DBX377.6	DB330x.DBX4005.6	

## 4.16.3.7 軸/主軸への信号

信号名称	SINUMERIK 840D sl	SINUMERIK 828D
送り速度オーバーライド	DB31、... .DBB0	DB380x.DBB0
オーバーライド有効	DB31、... .DBX1.7	DB380x.DBX1.7
残移動距離削除/主軸リセット	DB31、... .DBX2.2	DB380x.DBX2.2
ハンドルの起動	DB31、... .DBX4.0 - 2	DB380x.DBX4.0 - 2
移動キーの禁止	DB31、... .DBX4.4	DB380x.DBX4.4
早送りオーバーライド	DB31、... .DBX4.5	DB380x.DBX4.5
移動キー「マイナス」	DB31、... .DBX4.6	DB380x.DBX4.6
移動キー「プラス」	DB31、... .DBX4.7	DB380x.DBX4.7
要求の機械機能	DB31、... .DBX5.0 - 6	DB380x.DBX5.0 - 6
手動パルス発生器の回転方向の反転	DB31、... .DBX7.0	DB380x.DBX7.0
ジョグ固定点アプローチ	DB31、... .DBX13.0 - 2	DB380x.DBX1001.0 - 2
位置へのジョグ移動	DB31、... .DBX13.3	DB380x.DBX1001.3

## 4.16.3.8 軸/主軸からの信号

信号名称	SINUMERIK 840D sl	SINUMERIK 828D
汎用/精密イグザクトストップ範囲内の位置に到達	DB31、... .DBX60.6 - 7	DB390x.DBX0.6 - 7
ハンドルオーバーライド有効	DB31、... .DBX62.1	DB390x.DBX2.1
ハンドル有効	DB31、... .DBX64.0 - 2	DB390x.DBX4.0 - 2
移動要求「マイナス」	DB31、... .DBX64.4	DB390x.DBX4.4
移動要求「プラス」	DB31、... .DBX64.5	DB390x.DBX4.5
移動指令「マイナス」	DB31、... .DBX64.6	DB390x.DBX4.6
移動指令「プラス」	DB31、... .DBX64.7	DB390x.DBX4.7
有効機械機能	DB31、... .DBX65.0 - 6	DB390x.DBX5.0 - 6
手動パルス発生器の回転方向の反転が有効	DB31、... .DBX67.0	DB390x.DBX7.0
復元された位置、検出器 1/2	DB31、... .DBX71.4 - 5	DB390x.DBX11.4 - 5
ジョグ固定点アプローチ有効	DB31、... .DBX75.0 - 2	DB390x.DBX1001.0 - 2
ジョグア固定点アプローチへの到達	DB31、... .DBX75.3 - 5	DB390x.DBX1001.3 - 5

## 4.16 データリスト

信号名称	SINUMERIK 840D sl	SINUMERIK 828D
位置へのジョグ移動有効	DB31, ... .DBX75.6	DB390x.DBX1001.6
ジョグ位置決め完了	DB31, ... .DBX75.7	DB390x.DBX1001.7

## 4.16.4 システム変数

## 4.16.4.1 システム変数

識別子	説明
\$AA_POSRES	軸状態「位置復元済み」

## 4.16.5 OPI 変数

## 4.16.5.1 OPI 変数

識別子	説明
retractState、ビット 0	後退データが使用可能
retractState、ビット 1	ジョグ後退有効
retractState、ビット 2 ~ 3	後退軸
aaPosRes	軸状態「位置復元済み」。

## K3:補正

### 5.1 はじめに

#### 精度誤差

工作機械の精度は、理想形状からの誤差、動力伝達誤差、および検出器誤差によって損なわれます。大型のワークの加工時に、温度差と機械的応力により、精度が大幅に低下します。

#### 補正機能

こうした誤差のいくつかは通常、セットアップ時に計測して、位置フィードバック値エンコーダ、その他の計測装置で読み取られた値に基づき、運転に備えて補正できます。最先端の CNC コントローラには、各軸上で有効な補正機能があります。

#### パラメータ設定

これらの補正機能は、軸別マシンデータで各機械について個別に設定できます。

#### 適用

入力データが使用可能になると直ちに、コントローラのすべての運転モードで補正が有効になります。軸がレファレンス点復帰するまで、位置フィードバック値を必要とする補正は有効になりません。

#### 位置表示

通常のフィードバック位置と指令位置の表示では、補正値が無視され、理想的な機械位置データが表示されます。この補正値は、[軸/主軸サービス]画面の「診断」操作エリアに出力されます。

## 5.2 熱変位補正

### 5.2.1 機能

#### 温度の作用による変形

ドライブ装置の発生熱、または(たとえば、日光、通気による)周囲温度の熱によって、機械のベッドと部品が膨張します。この膨張は、特に機械部品の温度と熱伝導度によって決まります。

#### 作用

機械の熱膨張のために、軸の実位置は温度に応じて、変化します。これはワークの加工精度に悪影響を及ぼします。

#### 熱変位補正

「熱変位補正」機能を有効にすることにより、温度の作用を軸毎に補正できるため、実位置が変化します。

#### センサ装置

有効な熱変位補正を得るために、既存のエンコーダからのフィードバック値データの他に、温度変化を取得するための多数の温度センサが必要です。

たとえば、温度による変化は比較的緩やかであるため、PLC は数分の周期で温度変化を取得し、前処理をおこなうことができます。

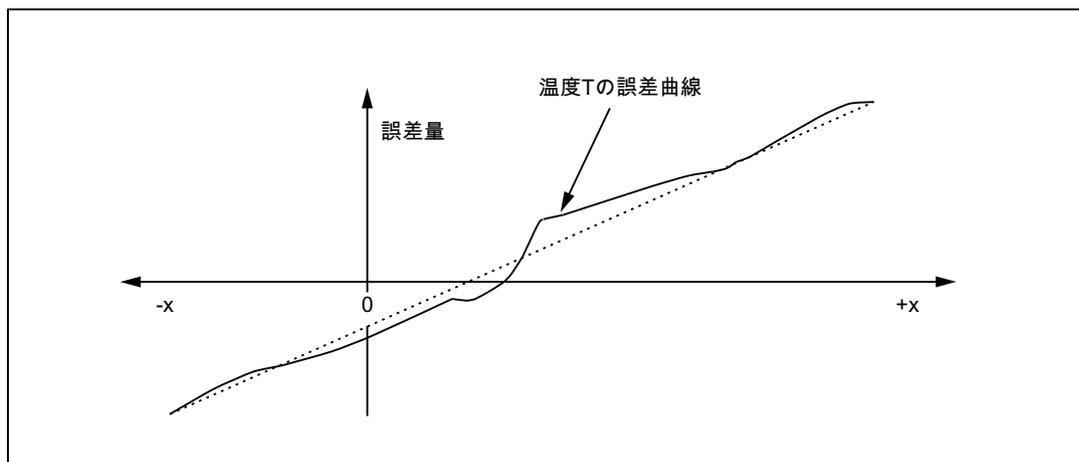
#### 誤差曲線

熱変位補正をおこなうために、所定の温度(T)で位置決め軸範囲のフィードバック値オフセットを計測し、プロットしてください。これにより、この温度値の誤差曲線が作成されます。それぞれの温度について誤差曲線を作成してください。

## 誤差曲線の特徴

軸に対して軸位置基準点  $P_0$  が選択されている場合、(熱変位補正の「位置に依存しない成分」に対応する)基準点のオフセットは、温度変化として監視され、そして、長さの変化によって、基準点への距離で増加するような、その他の位置決め点の追加オフセットは、(熱変位補正の「位置に依存した成分」に対応します)。

所定の温度  $T$  の誤差曲線は一般に、温度に依存した傾きと基準位置を持った直線によって十分な精度で表すことができます。



## 補正式

補正值  $\Delta K_x$  は、次の式に従って、この軸の現在の実位置  $P_x$  と温度  $T$  に基づいて計算されます。

$$\Delta K_x = K_0(T) + \tan\beta(T) * (P_x - P_0)$$

次に意味を示します。

- $\Delta K_x$ : 位置  $P_x$  の軸の熱変位補正值
- $K_0$ : 軸の位置に依存しない熱変位補正值
- $P_x$ : 軸の実位置
- $P_0$ : 軸の基準位置
- $\tan\beta$ : 位置に依存する熱変位補正の係数(概算誤差直線の傾きに対応します)

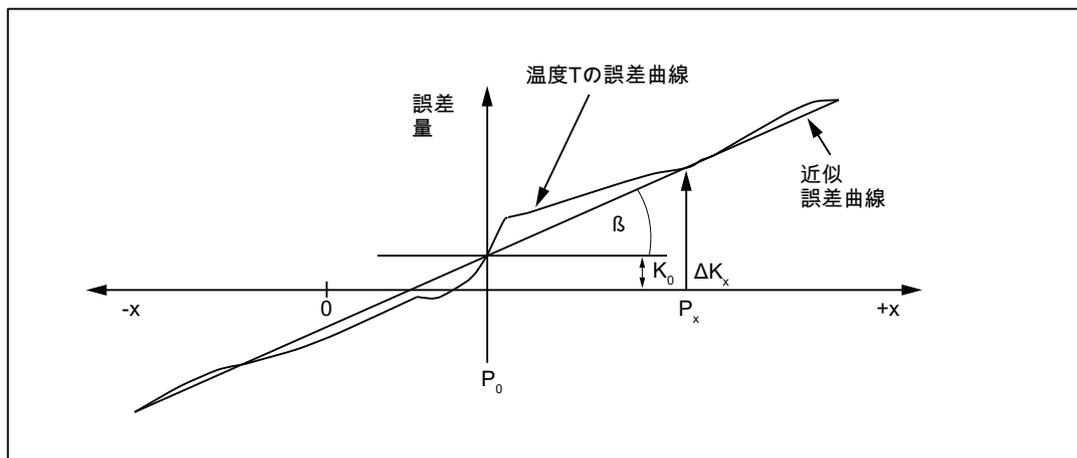


図 5-1 熱変位補正の概算誤差直線

## 適用

熱変位補正を有効にできるように、次の条件を満たしてください。

1. 補正タイプが選択されていること(MD32750、「セットアップ (ページ 305)」を参照してください)。
2. 補正タイプのパラメータが定義されていること(「セットアップ (ページ 305)」を参照してください)。
3. 軸が原点確立されていること。  
DB31、... DBX60.4 または 60.5 = 1 (原点確立済み/原点同期済み 1 または 2)

こうした条件が満たされると直ちに、すべてのモードで位置フィードバック値に対する熱変位補正值が指令値に追加され、機械軸がこの距離だけ移動します。補正值 $\Delta K_x$ が正である場合、軸は負方向に移動します。

たとえばエンコーダの周波数が超過してしまったために基準位置が失われた場合 (DB31、... DBX60.4 または 60.5 = 0)、補正処理が無効になります。

## 周期

補正值は補間器クロック周期で特定されます。

## 表示

実位置に対応する温度と真直度補正機能から計算された合計補正值は、[軸/主軸サービス]画面の「診断」操作エリアに表示されます。

## 温度変化に合わせたパラメータ補正

概算誤差直線は瞬間的な温度値にのみ適用されるので、温度が上下したときに、新たに生成される誤差直線のパラメータを NC に再送信する必要があります。こうした方法だけが、熱による膨張を常に正しく補正できます。

温度 T が変化すると、温度依存のパラメータ ( $K_0$ 、 $\tan\beta$  および  $P_0$ ) も変更されるため、常に PLC またはシンクロナイズドアクションによって上書きできます。

したがって工作機械メーカーは、PLC ユーザープログラムによって軸位置と温度値間の数学的および技術的な関係をエミュレートできます。また、熱変位補正のさまざまなパラメータを計算できます。温度パラメータは、変数サービス (FB2 (GET) 「読み取りデータ」と FB3 (PUT) 「書き込みデータ」) を使用して NC に伝送されます。

パラメータの取扱いと FB2 と FB3 の提供についての詳細は、以下を参照してください。

### 参照先:

機能マニュアル 基本機能、PLC 基本プログラム (P3)

## 補正值の平滑化

機械のオーバーロードまたは熱変位補正パラメータでステップ応答動作時に監視機能の異常検出を防止するために、次のマシンデータで各 IPO サイクルについて指定された最大補正值を超過すると、直ちに複数の IPO サイクルで内部制御機能によって補正值が分配されます (MD32760、「セットアップ (ページ 305)」を参照してください)。

## 5.2.2 セットアップ

### 温度依存のパラメータ

それぞれの温度の誤差曲線は、各軸に対して定義できます。各誤差曲線について、次のパラメータを決定して、セッティングデータに入力してください。

- 位置に依存しない熱変位補正值  $K_0$ :  
SD43900 \$SA\_TEMP\_COMP\_ABS\_VALUE
- 位置に依存する熱変位補正の基準位置  $P_0$ :  
SD43920 \$SA\_TEMP\_COMP\_REF\_POSITION
- 位置に依存する熱変位補正の傾き  $\tan\beta$ :  
SD43910 \$SA\_TEMP\_COMP\_SLOPE

### 熱変位補正タイプと適用

熱変位補正タイプを選択し、次の軸マシンデータを使用して熱変位補正を有効にします。

MD32750 \$MA\_TEMP\_COMP\_TYPE (熱変位補正タイプ)

ビット	規格値	意味	関連パラメータ
0		位置に依存しない熱変位補正	SD43900
	0	無効	
	1	有効	
1		位置に依存する熱変位補正	SD43920、SD43910
	0	無効	
	1	有効	
2		工具方向の熱変位補正	MD20390 \$MC_TOOL_TEMP_COMP_ON (工具長の熱変位補正の有効化)
	0	無効	
	1	有効	

### IPO サイクル毎の最大補正值

IPO サイクル毎の使用可能な最大補正值、つまり熱変位補正の結果として IPO サイクルで移動できる最大距離は、マシンデータを使用して制限されます。

MD32760 \$MA\_COMP\_ADD\_VELO\_FACTOR (補正の結果としての速度増加)

指定された値は係数として使用され、最大軸速度(MD32000 \$MA\_MAX\_AX\_VELO)で参照されます。

MD32760 は、熱変位補正の誤差直線( $\tan \beta$ )の最大傾きも制限します。

## 5.2.3 例

### 5.2.3.1 旋盤の Z 軸の熱変位補正のセットアップ

次に、下記の旋盤の Z 軸の例を使用して、熱変位補正のセットアップについて説明します。

## Z軸の誤差特性の特定

Z軸の温度に依存した誤差特性を特定するためには、次の実施手順に従います。

- Z軸の移動範囲全体を軸が移動することによる一定の温度上昇(この例では:500mm～1500mm)
- 100mmの分割で軸位置を計測
- 送りねじで実際の温度を計測
- 20分毎の移動計測サイクルの実施

熱変位補正に関する数学的および技術的な関係とその結果のパラメータが記録されたデータから得られます。下図には、個別の温度について計算された偏差誤差がグラフで表されています。これは、NCで表示されたZ軸の実位置を示します。

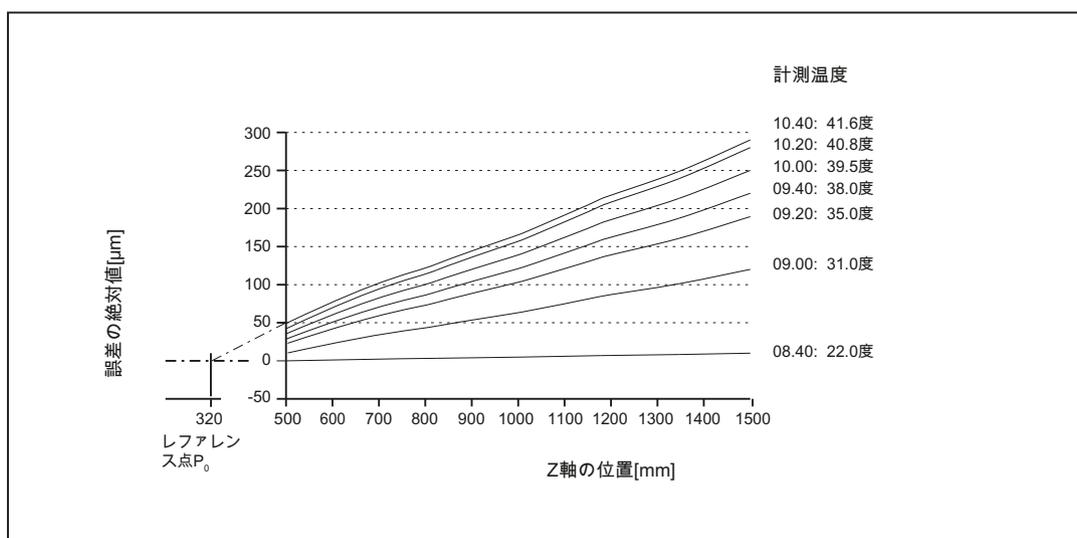


図 5-2 特定された Z 軸の誤差曲線

## パラメータの指定

ここで、計測結果に基づいて熱変位補正パラメータを特定してください(上図を参照してください)。

### 基準位置 $P_0$

上図に示したように、基準位置  $P_0$  をパラメータ設定するには、基本的に次の 2 つの方法があります。

1.  $P_0 = 0$  (位置に依存しない熱変位補正值  $K_0 \neq 0$ )
2.  $P_0 \neq 0$  (位置に依存しない熱変位補正值  $K_0 = 0$ )

## 5.2 熱変位補正

この場合は、方法 2 を選択します。つまり、位置に依存しない熱変位補正值は常に 0 です。したがって、熱変位補正值は、位置に依存した成分のみから構成されます。

次のパラメータが得られます。

- MD32750 \$MA\_TEMP\_COMP\_TYPE = 2  
(位置に依存した熱変位補正のみが有効)
- P<sub>0</sub> = 320 mm → SD43920 \$SA\_TEMP\_COMP\_REF\_POSITION = 320  
(位置に依存した熱変位補正の基準位置)

係数  $\tan\beta$  (T)

温度上で、位置に依存する熱変位補正係数  $\tan\beta$  の依存関係を特定するために、計測された温度に対して誤差曲線の傾きをプロットします。

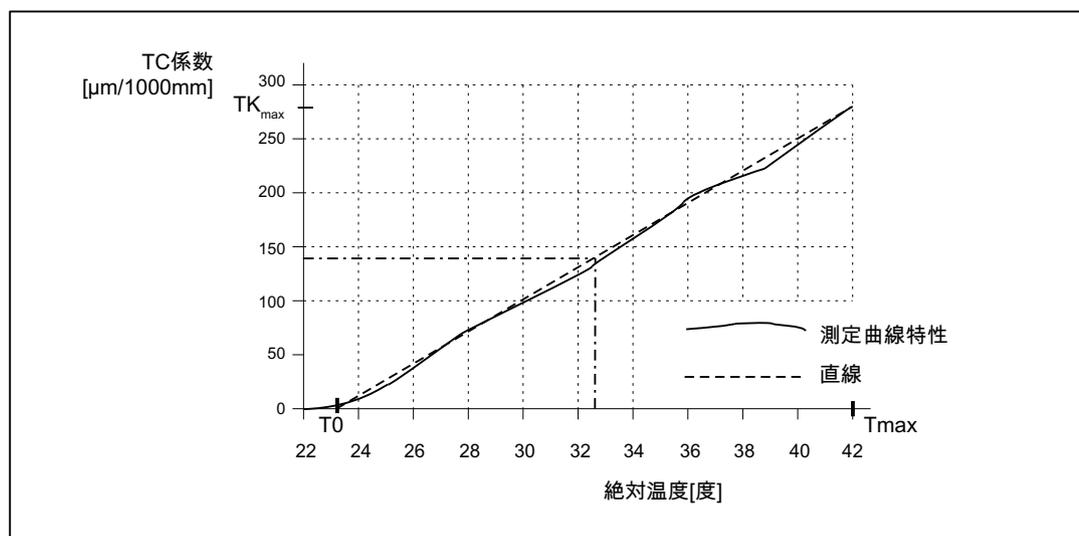


図 5-3 計測温度 T の関数としての係数  $\tan\beta$  の特性

適切な線形化により、係数  $\tan\beta$  は次のように T に依存します。

$$\tan\beta(T) = (T - T_0) * TK_{\max} * 10^{-6} / (T_{\max} - T_0)$$

記号の意味

T<sub>0</sub> = 位置に依存する誤差が 0 のときの温度 [°]

T<sub>max</sub> = 最大計測温度 [°]

TK<sub>max</sub> = T<sub>max</sub> の温度係数 [μm/1000 mm]

したがって、上記の図の値に基づいて:

$$T_0 = 23^\circ$$

$$T_{\max} = 42^\circ$$

$$TK_{\max} = 270 \mu\text{m}/1000 \text{ mm}$$

したがって  $\tan\beta$  (T)は:

$$\begin{aligned}\tan\beta(T) &= (T - 23) [^\circ] * 270 [\mu\text{m}/1000 \text{ mm}] * 10^{-6} / (42 - 23) [^\circ] \\ &= (T - 23) [^\circ] * 14.21 [\mu\text{m}/1000 \text{ mm}] * 10^{-6}\end{aligned}$$

例:

したがって、 $T = 32.3^\circ$ で: $\tan\beta = 0.000132$  です。

## PLC ユーザープログラム

計測温度に対応する係数  $\tan\beta(T)$ を計算するために PLC ユーザープログラムで上記の計算式を使用し、その結果を次の NC セッティングデータに書き込んでください。

SD43910 \$SA\_TEMP\_COMP\_SLOPE (位置に依存する熱変位補正の傾き)

前述の例では、次のようになります。

SD43910 \$SA\_TEMP\_COMP\_SLOPE = 0.000132

## 5.3 バックラッシ補正

### 5.3.1 機械バックラッシ補正

#### 5.3.1.1 機能

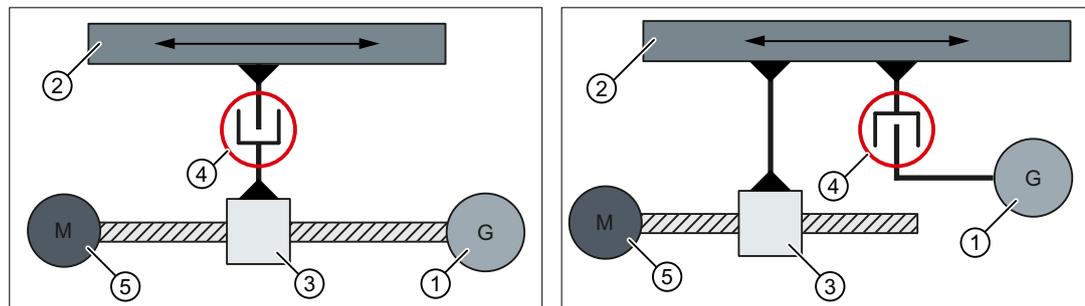
ボールねじや検出器接続部分など、動力伝達での機械構成部分(機械軸)の移動で機械的バックラッシが発生する可能性があります。

## 5.3 バックラッシュ補正

## 作用

間接検出器を備えた機械軸の場合、機械的バックラッシュによって、検出器を使って特定した NC の実際の位置と、機械構成部分の実際の位置に差異が生じます。方向を反転すると、機械軸はバックラッシュ分だけ、ずれた距離を移動します。

正のバックラッシュ⇒ エンコーダの現在値が機械の現在値より先行  
 負のバックラッシュ⇒ エンコーダの現在値が機械の現在値より遅行



テーブルの静止中にバックラッシュによりエンコーダがすでに移動しているにも関わらず、エンコーダの現在値がすでに変更されているため、バックラッシュによってエンコーダの現在値(先行)、テーブルが十分な距離を移動しませんでした(遅行)、テーブルの移動距離が大きすぎます。

- |         |                    |
|---------|--------------------|
| ① エンコーダ | ② 移動した機械構成部分(テーブル) |
| ③ ボールねじ | ④ 機械的バックラッシュ       |
| ⑤ 電動機   |                    |

## 補正

機械的バックラッシュを補正するために、セットアップ (ページ 311)中に設定された軸別補正值によって軸が反転するたびに機械軸の現在値を修正します。

## 適用

機械軸の機械的バックラッシュの補正は、すべての動作状態で有効です。前提条件:

- インCREMENTAL検出器:エンコーダ状態 == 「原点確立済み」
- 絶対値エンコーダ:エンコーダ状態 == 「原点同期済み」

## 補正值の表示

機械軸の実際の位置で有効な補正值は、個々の軸のユーザーインターフェースに表示されます。

SINUMERIK Operate:[「診断」操作エリア| ETC キー (「>」)| 「軸の診断」| 「サービス軸」|

- 「アブソリュート補正值検出器 1」]
- 「アブソリュート補正值検出器 2」]
- 「補正、真直度 + 熱変位補正」]

### 5.3.1.2 試運転:軸マシンデータ

#### 補正值

機械的バックラッシの補正值は、マシンデータに入力します。

MD32450 \$MA\_BACKLASH[ <有効な検出器> ] = <補正值>

#### 計測

- 機械軸や機械構成部分を計測位置に高速で移動します。
- 機械構成部分の実際の位置の計測
- バックラッシ補正值  $K_L$  の計算
  - $K_L =$  「機械軸の表示された実際の位置」 - 「機械構成部分の計測された実際の位置」
  - $K_L > 0$  (正のバックラッシ)  $\Rightarrow$  正の補正值
  - $K_L < 0$  (負のバックラッシ)  $\Rightarrow$  負の補正值

#### 確認

補正の効果を確認するため、機械的バックラッシの計測は、マシンデータを有効にした後に行うことを推奨します。

#### 真円度テスト

ユーザーインターフェースに内蔵された真円度テストを使用して、機械的バックラッシを視覚化できます。

SINUMERIK Operate:[「スタートアップ」操作エリア| 「オートチューニング/試験」| 「真円度テスト」]

#### 2番目の検出器

軸に2番目の検出器がある場合、この検出器の補正值も特定し、以下のマシンデータに入力してください。

MD32450 \$MA\_BACKLASH[ <検出器 2> ]

検出器の切り替えが行われる場合、関連する補正值は自動的に使用されます。

5.3 バックラッシュ補正

評価要因は、パラメータセットに依存します。

パラメータセットに依存する機械的バックラッシュに対して、次のマシンデータにパラメータセット固有の係数を入力し、これがバックラッシュの補正值(MD32450 \$MA\_BACKLASH)に適用されます。

MD32452 \$MA\_BACKLASH\_FACTOR[ <パラメータセット 1 ... n> ] = <係数>

有効な補正值

パラメータセットの有効な補正值  $K_p$  は、以下のように計算します。

$$K_p = ( MD32450 \$MA\_BACKLASH[ <有効な検出器> ] ) * ( MD32452 \$MA\_BACKLASH\_FACTOR[ <パラメータセット> ] )$$

バックラッシュ補正モード

機械軸の機械的バックラッシュ補正の動作は、次のマシンデータで設定します。

MD32454 \$MA\_BACKLASH\_MODE

ビット	意味
0	コントロールシステム起動後の、有効な最後のバックラッシュ補正を復元します。
	= 0    バックラッシュ補正值は <b>復元されません</b> (初期設定)。
	= 1    バックラッシュ補正值が復元されます。
	<b>必要条件</b> コントローラ電源投入後に有効な検出器が設定済みで原点同期済みであること。 <ul style="list-style-type: none"> <li>● MD34210 \$MA_ENC_REFP_STATE[&lt;有効な検出器&gt;] == 2</li> <li>● DB31、... .DBX60.4 - 5 == 1 (検出器 1 / 2:原点確立済み、原点同期済み)</li> </ul>
1	他の補正機能の効果(熱変位補正、真直度誤差または直角度誤差の補正)
	= 0    バックラッシュ補正が、他の補正動作に反応して動作します(初期設定)。
	= 1    バックラッシュ補正が、他の補正動作に反応して <b>動作しません</b> (初期設定)。

## 5.3.2 ダイナミックバックラッシュ補正

### 5.3.2.1 機能

#### ダイナミックバックラッシュ

ダイナミックバックラッシュは、スライドガイド付き機械タイプに対して実行可能です。終了位置にアプローチするために使用される軸ダイナミック応答(速度、加々速度など)に対応して、機械スライドはプログラム指令された終了位置に達するか、またはそれ以前に静止摩擦のために停止します。位置エラーの結果は、方向対称です。

#### 補正

ダイナミックバックラッシュを補正するには、符号付き補正值(MD32456、「試運転:軸マシンデータ (ページ 314)」を参照してください)の半分を軸の当該移動方向に従って適用します。補正值は、カーブとして適用されます。

#### 適用

ダイナミックバックラッシュ補正は、以下の必要な状態で PLC によってのみ起動されます。

DB31、... DBX25.0 (ダイナミックバックラッシュ補正の起動)

---

#### 注記

工作機械メーカーは、PLC ユーザープログラムでダイナミックバックラッシュ補正を起動するために「必要な」状態を指定します。このような状態は、G1 による JOG モードまたは手動パルス発生器での軸の移動の結果として生じます。

---

NC は次のインタフェースを使用して、PLC に必要な起動を返します。

DB31、... DBX102.0 (ダイナミックバックラッシュ補正の起動)

#### 必要条件

補正する軸は原点確立済みであることが必要です。

## 表示

現在の実位置に対応する補正值は、機械的およびダイナミックなバックラッシ補正である LEC から計算された合計補正值として、[軸/主軸サービス]ウィンドウの「アラーム診断」オペレータエリアに表示されます。

## 5.3.2.2 試運転:軸マシンデータ

## 補正值

ダイナミックなバックラッシ補正をセットアップする必要条件として、機械的バックラッシ補正がすでにセットアップ済みであることが必要です(「試運転:軸マシンデータ (ページ 311)」の章を参照してください)。ダイナミックなバックラッシ補正の補正值を決定するために、そこで説明されている計測を低い移動速度で実行します。このようにして特定した補正值に対応する検出器のマシンデータに入力します。

MD32456 \$MA\_BACKLASH\_DYN[ <有効な検出器> ] = <ダイナミックなバックラッシ補正の補正值>

## 補正值の変更

マシンデータを使って、補正值を移動する速度を最大軸速度 (MD32000 \$MA\_MAX\_AX\_VELO)のパーセントで設定します。

MD32457 \$MA\_BACKLASH\_DYN\_MAX\_VELO = <最大軸速度のパーセント>

## 5.3.3 二重位置検出

「二重位置検出フィードバック」機能では、従来の(機械的またはダイナミックな)バックラッシ補正とは対照的に、閉ループ位置制御に 2 つの検出器を使用します。これらの検出器は、バックラッシが発生しないギヤボックスで機械的に連結されています。直接検出器の長所が、間接検出器の長所と組み合わせられています。

- 直接検出器:直接、つまりエンコーダを負荷側に備えた閉ループ位置制御で、「自動的に」バックラッシを補正します。
- 間接検出器:間接、つまりエンコーダーをモーター側に備えた閉ループ位置制御で、不連続のバックラッシに対して「大まか」で「安定」しています。

セットポイントが変更されるたびに、最初に間接検出器からのエンコーダデータのみを使用して、大まかで振動のない制御を行います。パラメータ設定された遅延時間を経て、閉ループ制御はスムーズに直接検出器の監視に移行し、負荷側で必要な精度を達成しま

す。この時点では、制御操作は短距離のみに限ります。これは、その間に歯面が相互に接触し、バックラッシュが伝わるからです。

「二重位置検出フィードバック」機能を使用する場合、バックラッシュを計測し、数学的に補正する必要はありません。

## 必要条件

補正する軸で、以下の必要条件を満たす必要があります。

- 直接および間接検出器、機械的に連結:
  - MD30200 \$MA\_NUM\_ENCS = 2
  - MD31040 \$MA\_ENC\_IS\_DIRECT[ 0 ] = 0 または 1
  - MD31040 \$MA\_ENC\_IS\_DIRECT[ 1 ] = 1 または 0
- 検出器の位置合わせがリリースされている:
  - MD34102 \$MA\_REFP\_SYNC\_ENCS = 1
- 両方の検出器が原点確立済み:
  - DB31、... DBX60.4 (原点確立済み/原点同期済み 1 / 2) = 1
  - DB31、... DBX60.5 (原点確立済み/原点同期済み 1 / 2) = 1

### 5.3.3.1 試運転:軸マシンデータ

#### 遅延時間

閉ループ制御がスムーズに間接から直接検出器へ移行するための遅延時間は、以下のマシンデータに入力します。

MD32960 \$MA\_POSCTRL\_DUAL\_FEEDBACK\_TIME = <遅延時間>

遅延時間がゼロの場合、この機能は無効になり、有効な検出器(DB31、... DBX1.5 / .6 (位置検出器 1 / 2) )のみが位置制御で有効になります。

---

#### 注記

「二重位置検出フィードバック」機能を有効にした後、すべての既存の検出器補正および監視機能は変更されず、有効な状態を維持します。つまり、これらは必要に応じてユーザーが無効にします(たとえばバックラッシュ補正値を削除するなど)。

---

### 5.3.3.2 必要条件

#### 原点確立および浮動計測

「二重位置検出フィードバック」機能が有効な場合も、「原点確立」および「計測」機能は有効な検出器(DB31、...DBX1.5/.6 (位置検出器 1/2))を参照します。

#### 参照先

- 機能マニュアル 基本機能、章「R1 原点確立」
- 機能マニュアル 上級機能、章「M5 計測」

## 5.4 ノディング補正

### 5.4.1 機能の説明

#### 5.4.1.1 オプション

「ノディング補正」機能は、ライセンス対象のオプションです。次の種類が使用可能です。

- ノディング補正 **ECO:6FC5800-0AS20-0YB0**  
加速軸が 1 本の場合の 1 本の機械軸のノディング補正
- ノディング補正 **ADVANCED:6FC5800-0AS21-0YB0**  
任意の数の機械軸のノディング補正

#### 5.4.1.2 特徴

ノディング補正を使用して、機械軸または別の機械軸の加速によるダイナミック位置の誤差を補正します。位置の誤差は、機械の中での機械的整合性により発生します。

次の 3 軸移動列機械の概略図は、その例を示しています。機械の機械的整合性によって Y 軸が  $y$  で加速している場合、位置の誤差  $\Delta_z$  が Z 軸で発生します。

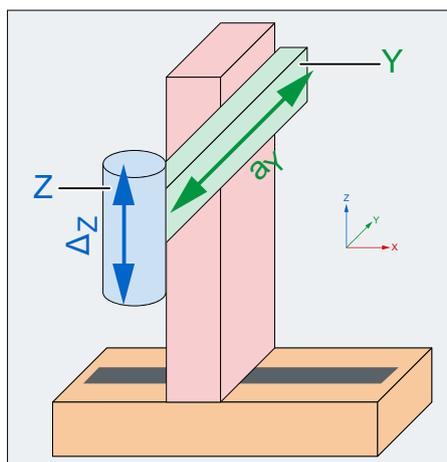


図 5-4 3 軸移動列機械

### 条件の定義

- ノディング補正軸  
ノディング補正に関連する機械軸は、次のように指定されます。
  - 補正軸  
他の機械軸が加速しているために位置の誤差が発生した機械軸。この誤差は、ノディング補正で補正できます。図では、補正軸⇒Z軸
  - 加速軸  
機械の整合性により加速している機械軸。この軸が、補正軸の位置の誤差を引き起こしています。図では、加速軸⇒Y軸
  - 調整軸  
加速軸の移動時のこの機械軸の位置が、補正軸の位置の誤差の大きさに影響します。図では、調整軸⇒X軸
- 整合係数  
整合係数  $N$  は、加速軸による加速度  $a$  での移動動作で生じた、補正軸の位置の誤差  $\Delta$  の大きさを定義します。整合係数  $N$  は、補正軸の位置の誤差  $\Delta$  を、実際に誤差を引き起こした加速軸の加速度  $a$  で除算した商です。 $N = \Delta / a$

---

### 注記

#### 参照先

整合係数の特定方法の詳細については、以下を参照してください。

- 試運転マニュアル、「CNCの試運転:NC、PLC、ドライブ;「NCの試運転」章>「ノディング補正」
- オンラインヘルプ  
ノディング補正のオンラインヘルプは、SINUMERIK Operate ユーザーインターフェースで以下を使用すると表示されます。「試運転」>「NC」>「ノディング補正」>操作パネル前面のヘルプボタン

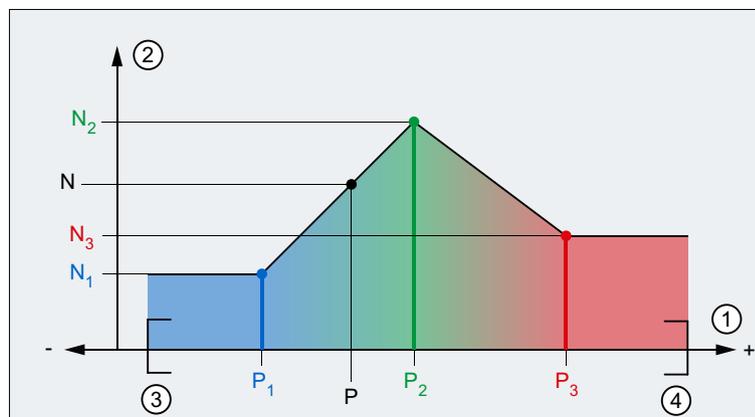
- 
- 補正值  
実際の補正值  $\Delta$  は、実際の整合係数  $N$  と、誤差を引き起こしている軸の実際の加速度の乗算により得られます。

### 必要条件

- 回転軸  
補正軸、加速軸、調整軸はいずれも回転軸であってはなりません。
- コンパイルサイクル「RMCC/NOCO ノディング補正」との相違点  
(6FC5800-0AN63-0YB0)
  - システム機能「ノディング補正」の機能は、コンパイルサイクル「RMCC/NOCO」とは対照的に、ノディング動作の補正のみに限られます。回転軸の位置および総合的な位置テーブルを考慮した、これ以外にコンパイルサイクルで使用できる機能(遠心力の補正など)は、システム機能では使用できません。
  - システム機能「ノディング補正」またはコンパイルサイクル「RMCC/NOCO」を使用することはできますが、両方を同時には使用できません。
  - コンパイルサイクル「RMCC/NOCO」のパラメータ設定データを使用して、「ノディング補正」システム機能のパラメータ設定をおこなうことはできません。

#### 5.4.1.3 位置に依存しない整合係数

機械の整合性が、他の機械軸(調整軸)の位置に応じて異なる場合は、調整軸の位置を最大3か所、調整特性の補間点として選択できます。この特性から、有効な整合係数を特定できます。位置(P1、P2、P3)と、各位置で有効な整合係数(N1、N2、N3)を、補正軸のマシデータに保存する必要があります。加速軸の移動時に、有効な整合係数は、補間点間で直線で補間することにより、調整軸の位置に応じて計算できます。



- ①  $P_1 < P_2 < P_3$  の場合の調整軸の移動範囲
- ② 補正軸の整合係数
- ③ 移動範囲の下限
- ④ 移動範囲の上限

図 5-5 補正特性

現在の位置:P	整合係数:N
P が $P_1$ より小さく、移動範囲の下限以上	$N = N_1$
P が $P_1$ と $P_2$ の間	$N = N_1$ と $N_2$ の間の直線補間
P が $P_2$ と等しい	$N = N_2$
P が $P_2$ と $P_3$ の間	$N = N_2$ と $N_3$ の間の直線補間
P が $P_3$ 以上で、移動範囲の上限以下	$N = N_3$

#### 5.4.1.4 補正関係

補正関係により、以下に対する補正値の依存が定義されます。

- 加速軸の加速
- 調整軸の位置
- 機械の整合性

3つまでの補正関係を、補正軸のマシンデータでパラメータ設定できます。各補正関係について、次のデータを設定できます。

- 加速軸
- 調整軸
- 調整軸の位置の数
- 調整軸の位置決め
- 位置での整合係数

##### 加速軸または調整軸

NC の直線機械軸はいずれも、加速軸または調整軸として使用できます。補正軸自体としても使用できます。

##### 補正値の合計

補正軸に作用する補正値の合計は、個々の補正関係の補正値の合計です。

## 5.4.2 試運転:マシンデータ

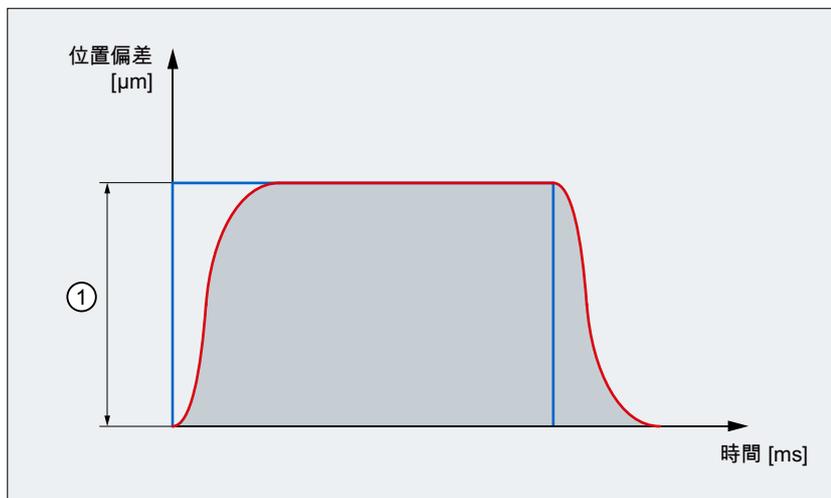
## 5.4.2.1 一覧

次の表は、ノディング補正のパラメータ設定に使用されるマシンデータの概要を示します。

番号	名称: \$MA_	意味
MD37302	NOCO_FILTER_TIME	ノディング補正のスムージングの時定数 (ページ 322)
<b>補正関係 1</b>		
MD37310	NOCO_INPUT_AX_1	ノディング動作を引き起こす機械軸 (ページ 322)
MD37312	NOCO_ADAPT_AX_1	その位置がノディング動作に影響を与える機械軸 (ページ 323)
MD37314	NOCO_ADAPT_NUM_1	位置の数、ノディング補正の調整特性 (ページ 324)
MD37316	NOCO_ADAPT_POS_1	位置、ノディング補正の調整特性 (ページ 325)
MD37318	NOCO_COMPLIANCE_1	ノディング補正の整合係数 (ページ 326)
<b>補正関係 2</b>		
MD37320	NOCO_INPUT_AX_2	MD37310 を参照
MD37322	NOCO_ADAPT_AX_2	MD37312 を参照
MD37324	NOCO_ADAPT_NUM_2	MD37314 を参照
MD37326	NOCO_ADAPT_POS_2	MD37316 を参照
MD37328	NOCO_COMPLIANCE_2	MD37318 を参照
<b>補正関係 3</b>		
MD37330	NOCO_INPUT_AX_3	MD37310 を参照
MD37332	NOCO_ADAPT_AX_3	MD37312 を参照
MD37334	NOCO_ADAPT_NUM_3	MD37314 を参照
MD37336	NOCO_ADAPT_POS_3	MD37316 を参照
MD37338	NOCO_COMPLIANCE_3	MD37318 を参照

## 5.4.2.2 ノディング補正のスムージングの時定数

マシンデータを使用して、入力時におこなわれるノディング補正値のスムージングに使用される時定数を設定します。



- ① 位置偏差
- 青:指令値
  - 赤:現在値

MD37302 \$MA\_NOCO\_FILTER\_TIME[<補正軸>] = <時定数>

## 注記

## スムージングの無効化

時定数を、位置コントローラのクロック周期の 10%未満に設定すると、スムージングは無効になります(または、有効でなくなります)。

時定数 <math>0,1 \* MD10061 \\$MN\\_POSCTRL\\_CYCLE\\_TIME</math>

## 5.4.2.3 加速軸

加速により補正軸の位置の誤差を引き起こす機械軸の番号をマシンデータに入力する必要があります。

加速軸の番号は、マシンデータ MD10002 \$MN\_AXCONF\_LOGIC\_MACHAX\_TAB でパラメータ設定される NC の機械軸を表します。この機械軸の数は、対応するマシンデータインデックスに 1 を加算した数です。

MD37310 \$MA\_NOCO\_INPUT\_AX\_1[<補正軸>] = <機械軸の番号>

## 例

機械の整合性の結果、Y 軸が加速すると、Z 軸の位置の誤差が発生します。

- 補正軸:AX3 (Z 軸)
- 加速軸:AX2 (Y 軸)

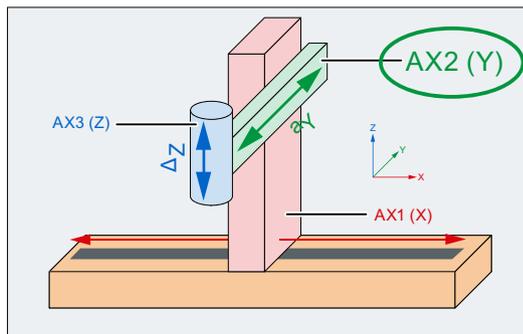


図 5-6 加速軸 AX2 (Y 軸)

```
MD10002 $MN_AXCONF_LOGIC_MACHAX_TAB[ 0 ] = "AX1" ; X 軸
MD10002 $MN_AXCONF_LOGIC_MACHAX_TAB[ 1 ] = "AX2" ; Y 軸
MD10002 $MN_AXCONF_LOGIC_MACHAX_TAB[ 2 ] = "AX3" ; Z 軸
```

```
MD37310 $MA_NOCO_INPUT_AX_1[ AX3 ] = 2 ; 加速軸= Y 軸
```

## 5.4.2.4

## 調整軸

その実際の位置が補正軸の位置の誤差に影響する機械軸の数をマシンデータに入力する必要があります。

機械軸の数は、マシンデータ MD10002 \$MN\_AXCONF\_LOGIC\_MACHAX\_TAB でパラメータ設定される NC の機械軸の数です。この機械軸の数は、対応するマシンデータインデックスに 1 を加算した数です。

```
MD37312 $MA_NOCO_ADAPT_AX_1[<補正軸>] = <機械軸の数>
```

## 注記

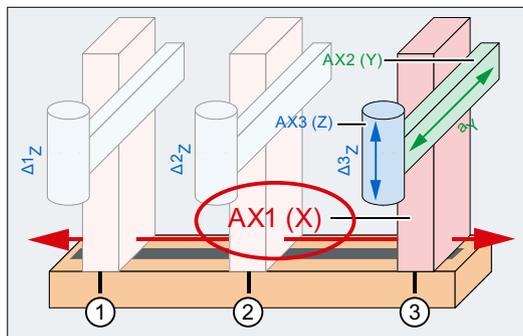
## 位置に依存する調整の有効化

位置に依存する調整は、1 本の調整軸と複数の位置(MD37314 \$MA\_NOCO\_ADAPT\_NUM\_1 (ページ 324))をパラメータ設定した場合にのみ有効になります。

例

機械の整合性の結果、Y 軸が加速すると、Z 軸の位置の誤差が発生します。さらに、位置の誤差は、X 軸の位置によって異なります。

- 補正軸:AX3 (Z 軸)
- 加速軸:AX2 (Y 軸)
- 調整軸:AX1 (X 軸)



①, ②, ③ 調整軸 AX1 (X 軸)の軸位置

図 5-7 調整軸 AX1 (X 軸)

```

MD10002 $MN_AXCONF_LOGIC_MACHAX_TAB[ 0 ] = "AX1" ; X 軸
MD10002 $MN_AXCONF_LOGIC_MACHAX_TAB[ 1 ] = "AX2" ; Y 軸
MD10002 $MN_AXCONF_LOGIC_MACHAX_TAB[ 2 ] = "AX3" ; Z 軸

MD37312 $MA_NOCO_ADAPT_AX_1[ AX3 ] = 1 ; 調整軸= X 軸
    
```

5.4.2.5 調整特性の位置の数

別の整合が有効になっている軸位置の数を、位置に依存する調整のマシンデータに入力する必要があります。

```

MD37314 $MA_NOCO_ADAPT_NUM_1[<補正軸>] = <数>
    
```

注記

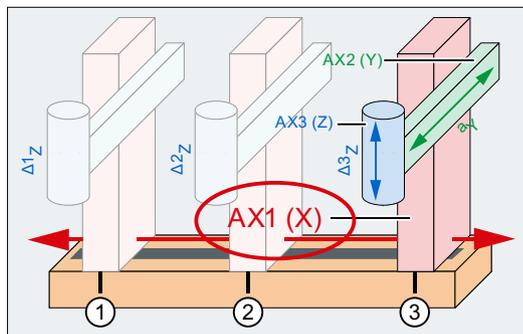
位置に依存する調整の有効化

位置に依存する調整は、複数の位置と 1 本の調整軸(MD37312 \$MA\_NOCO\_ADAPT\_AX\_1 (ページ 323))をパラメータ設定した場合にのみ有効になります。

## 例

調整特性の補間点を、調整軸(X 軸)の移動範囲に沿った 3 か所の位置でパラメータ設定します。

- 補正軸:AX3 (Z 軸)
- 加速軸:AX2 (Y 軸)
- 調整軸:AX1 (X 軸)



①, ②, ③ 調整軸 AX1 (X 軸)の軸位置

図 5-8 調整軸 AX1 (X 軸)の軸位置の数

```
MD10002 $MN_AXCONF_LOGIC_MACHAX_TAB[ 0 ] = "AX1" ; X 軸
MD10002 $MN_AXCONF_LOGIC_MACHAX_TAB[ 1 ] = "AX2" ; Y 軸
MD10002 $MN_AXCONF_LOGIC_MACHAX_TAB[ 2 ] = "AX3" ; Z 軸
```

```
MD37314 $MA_NOCO_ADAPT_NUM_1[ AX3 ] = 3 ; 軸位置の数
```

以下も参照ください:

- 軸位置:MD37316 \$MA\_NOCO\_ADAPT\_POS\_1 (ページ 325)
- 整合係数:MD37318 \$MA\_NOCO\_COMPLIANCE\_1 (ページ 326)

## 5.4.2.6

## 調整特性の位置

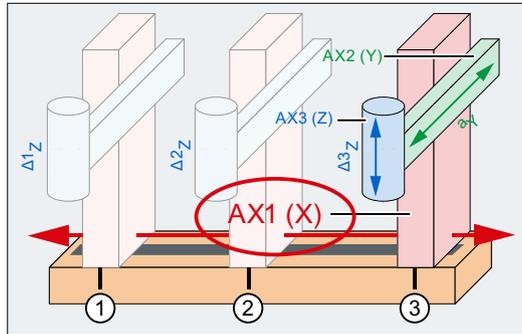
専用の整合係数のパラメータ設定がおこなわれる調整軸の位置を、調整特性の補間点のマシンデータに入力する必要があります。

```
MD37316 $MA_NOCO_ADAPT_POS_1[<インデックス>, <補正軸>] = <位置>
```

例

調整特性の調整軸 AX1 (X 軸)の位置、-100.0、0.0、および 120.0 で補間点をパラメータ設定します。

- 補正軸:AX3 (Z 軸)
- 加速軸:AX2 (Y 軸)
- 調整軸:AX1 (X 軸)



- ① 位置 1. -100.0
- ② 位置 2. 0.0
- ③ 位置 3. 120.0

図 5-9 調整軸 AX1 (X 軸)の軸位置

```
MD10002 $MN_AXCONF_LOGIC_MACHAX_TAB[ 0 ] = "AX1" ; X 軸
MD10002 $MN_AXCONF_LOGIC_MACHAX_TAB[ 1 ] = "AX2" ; Y 軸
MD10002 $MN_AXCONF_LOGIC_MACHAX_TAB[ 2 ] = "AX3" ; Z 軸
```

```
MD37316 $MA_NOCO_ADAPT_POS_1[ 0, AX3 ] = -100.0
MD37316 $MA_NOCO_ADAPT_POS_1[ 1, AX3 ] = 0.0
MD37316 $MA_NOCO_ADAPT_POS_1[ 2, AX3 ] = 120.0
```

以下も参照ください:

- 軸位置の数:MD37314 \$MA\_NOCO\_ADAPT\_NUM\_1 (ページ 324)
- 整合係数:MD37318 \$MA\_NOCO\_COMPLIANCE\_1 (ページ 326)

5.4.2.7 整合係数

調整特性の特定の位置(MD37316 \$MA\_NOCO\_ADAPT\_POS\_1 (ページ 325))に関連する整合係数をマシンデータに入力する必要があります。

MD37318 \$MA\_NOCO\_ADAPT\_COMPLIANCE\_1[<インデックス>, <補正軸>] = <整合係数>

#### 位置に依存する調整をおこなわない場合

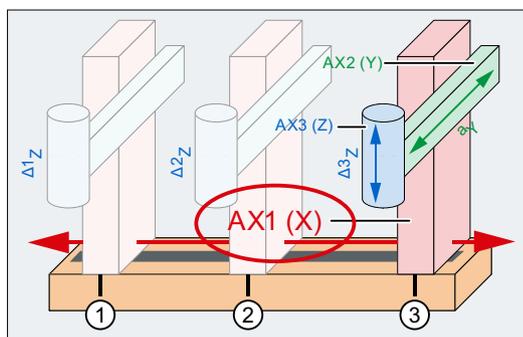
位置に依存する調整をパラメータ設定しない場合(MD37312 \$MA\_NOCO\_ADAPT\_AX\_1 (ページ 323) == 0)は、整合係数のインデックスに 0 を入力する必要があります。

MD37318 \$MA\_NOCO\_ADAPT\_COMPLIANCE\_1[0, <補正軸>] = <整合係数>

#### 例

調整軸(X軸)の移動範囲の位置、-100.0、0.0、および 120.0 で、補正軸 AX3 (Z軸)の整合係数として-12、-25、および-10  $\mu\text{m} / \text{m/s}^2$  が特定されます。

- 補正軸:AX3 (Z軸)
- 加速軸:AX2 (Y軸)
- 調整軸:AX1 (X軸)



- ① 位置 1 の整合係数: -12  $\mu\text{m} / \text{m/s}^2$
- ② 位置 2 の整合係数: -25  $\mu\text{m} / \text{m/s}^2$
- ③ 位置 3 の整合係数: -10  $\mu\text{m} / \text{m/s}^2$

図 5-10 調整軸:AX1 (X軸)

```
MD10002 $MN_AXCONF_LOGIC_MACHAX_TAB[ 0 ] = "AX1" ; X軸
MD10002 $MN_AXCONF_LOGIC_MACHAX_TAB[ 1 ] = "AX2" ; Y軸
MD10002 $MN_AXCONF_LOGIC_MACHAX_TAB[ 2 ] = "AX3" ; Z軸
```

```
MD37318 $MA_NOCO_COMPLIANCE_1[ 0, AX3 ] = -0.000012
MD37318 $MA_NOCO_COMPLIANCE_1[ 1, AX3 ] = -0.000025
MD37318 $MA_NOCO_COMPLIANCE_1[ 2, AX3 ] = -0.000010
```

5.4 ノディング補正

以下も参照ください:

- 軸位置の数:MD37314 \$MA\_NOCO\_ADAPT\_NUM\_1 (ページ 324)

5.4.2.8 ファンクションダイアグラム

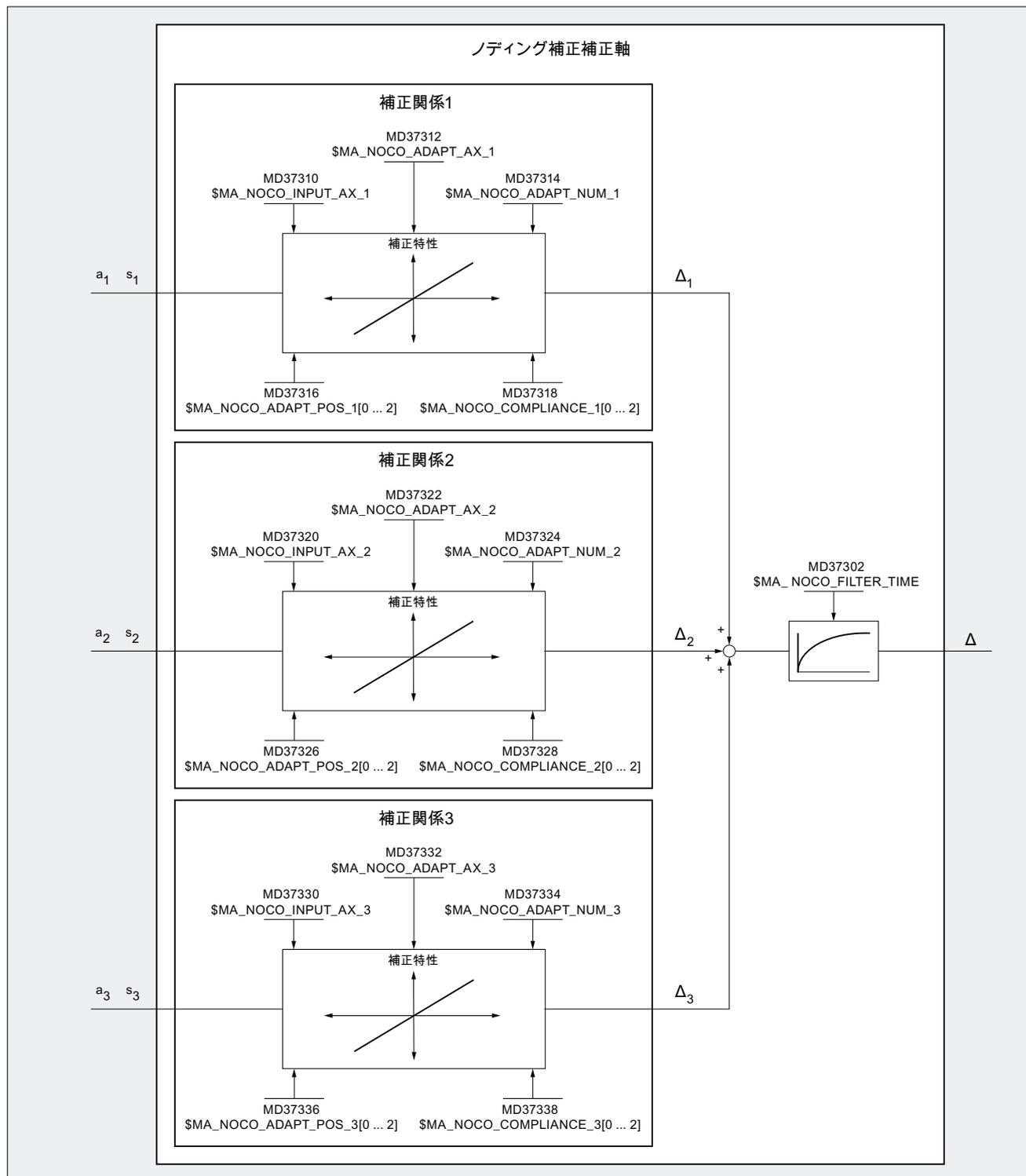


図 5-11 補正軸のファンクションダイアグラム

## 5.5 補間補正

### 5.5.1 一般特性

#### 機能

「補間補正」では、送りねじ、検出器、真直度および直角度誤差を使って、目的の軸の位置と実際の位置の誤差を補正できます。これは、計測によって誤差を特定し、対応する補正値を 1 つ以上の NC の補正テーブルに保存して行います。運転中、現在の設定位置に基づいて、関連する補正値が補正軸の補正テーブルから特定されます。補正テーブル補間点の間では、直線補間が実行されます。

「補間補正」内では、次の 2 つの補正方法が区別されます。

- ピッチ誤差と検出器誤差の補正
- 真直度誤差と直角度誤差の補正

#### 用語

- 補正値  
目的の軸の位置と実際の軸の位置の差がゼロになるように補正軸適用される補助設定値。
- 基本軸  
その設定位置と実際の位置から補正値を特定する軸。
- 補正軸  
設定位置または実際の位置に補正値が適用される軸。
- 補間点  
補正テーブルの値のペアで、基本軸の設定位置および実際の位置(補間点)と、関連する補正値(補間値)で構成されます。
- 補正テーブル  
基本軸の定義されたトラベル範囲に定義された数の補間点。
- 関連補正  
基本軸、補正軸および補正テーブルで構成される単位。

## 補正テーブルの入力

補正テーブルのサイズ(つまり補間点の数)は、マシンデータで最初に定義してください。次の電源投入後、NCによって補正テーブルが生成され、「0」の値が事前割り当てされます。

特別なシステム変数を使用して、補正值と追加テーブルのパラメータを補正テーブルに入力します。データは、次の異なる2つの方法で読み込むことができます。

- パラメータ値により NC プログラムを起動
- 外部コンピュータからコントローラへの補正テーブルの伝送

---

### 注記

関連する補正テーブルは、対応する補正機能が**すべての軸に有効でない**場合にのみロードできます。

- MD32700 \$MA\_ENC\_COMP\_ENABLE[<軸>] == 0
  - MD32710 \$MA\_CEC\_ENABLE[<軸>] == 0
- 

コントローラがオフになっても補正データは保持されます。

---

### 注記

マシンデータの変更時:

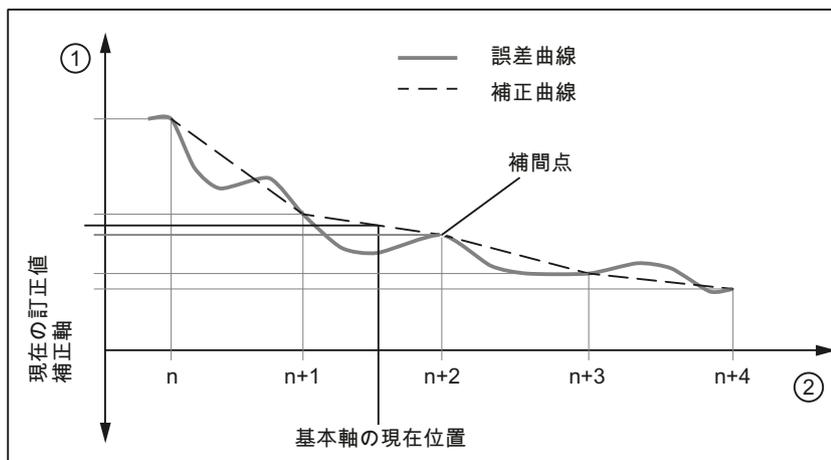
- MD18342 \$MN\_MM\_CEC\_MAX\_POINTS
- MD38000 \$MA\_MM\_ENC\_COMP\_MAX\_POINTS

静的ユーザーメモリが次のシステム起動時にフォーマットされます(「S7:メモリ構成(ページ 971)」の章を参照してください)。

---

暫定値の計算

補正を適用する開始および終了位置で定義された移動距離は、複数の同じサイズの区間に分けられます。区間の数は、誤差曲線と目的の精度に応じて定義します。こうした区間を限定する実位置を以降、「補間点」といいます。補間および補正值は、補間点に割り当ててください。2つの補間点間で、有効な補正值は**直線補間**で特定されます。



① 補正軸の補正值

② 基本軸の位置

図 5-12 直線補間による暫定値計算

必要条件

レファレンス点の補正值

補正テーブルは、軸のレファレンス点で補正值が「0」になるように構成することを推奨します。

## 5.5.2 ピッチ誤差およびピッチ誤差補正

### 5.5.2.1 検出器誤差補正(MSEC: Measuring System Error Compensation)

#### 送りねじと検出器誤差

NC 制御機械の、「間接計測」の計測原理は、ボールねじのリードが、移動範囲内のどの時点でも一定であり、その結果として軸の実位置をドライブメインシャフトの位置から得ることができること(理想事例)を前提条件としています。ただし、加工時の誤差は、メインシャフトの様々なレベルの問題に対して、寸法誤差が生じます(いわゆるピッチ誤差)。

これに加えて、検出器と、その検出器の機械への取り付け(いわゆる検出器取り付け誤差)、および機械側の誤差要因による寸法誤差(基準目盛りの違い)が存在します。

#### 補正

「検出器誤差補正」(以降、**MSEC** といいます)では、基本軸と補正軸は常に同じです。したがって、この**軸の補正**では、補正テーブルの基本軸と補正軸の定義が不要です。

---

#### 注記

ピッチ誤差補正(**LEC**)は、検出器誤差補正の一部です。

---

**MSEC** では原則的に、補間器クロックサイクルで、割り当てられた補正值で軸別位置フィードバック値を変更し、この値を機械軸に適用して即時に移動を行います。正の補正值により、対応する機械軸が負の方向に移動します。

補正值の大きさは制限されておらず、監視されません。補正に起因する許容されないほどの高速度と高加速度を避けるために、小さな補正值を選択してください。補正值が大きい場合、他の軸監視機能でアラーム(たとえば、輪郭監視、速度指令値の制限)が出力される場合があります。

補正対象の軸に第 2 の位置検出器が存在する場合、各検出器について個別の補正テーブルを作成し、有効にしてください。検出器間で切り替えをおこなう場合、自動的に適切なテーブルが使用されます。

## 必要条件/適用

MSEC は、次の必要条件が満たされる場合のみ有効になります。

- 補正值が不揮発性ユーザーメモリに設定され、(電源投入後に)有効になっている。
- 当該の機械軸に対して次の機能が有効である。

MD32700 \$MA\_ENC\_COMP\_ENABLE [<e>] = 1

意味: <e> = 位置検出器

<e> = 0 検出器 1

<e> = 1 検出器 2

- 軸が原点確立されている。  
DB31、... DBX60.4 または 60.5 = 1 (原点確立済み/原点同期済み 1 または 2)

上記の条件が満たされると直ちに、すべてのモードで軸別フィードバック値が補正值によって変更され、機械軸で直ちに移動されます。

たとえばエンコーダの周波数が超過したために原点が失われた場合(DB31、... DBX60.4 または 60.5 = 0)、補正処理が解除されます。

## 5.5.2.2 セットアップ

## マシンデータ

## 補正補間点の数

中国軸 - 各検出器について補正テーブルの補間点の数を記述してください。

MD38000 \$MA\_MM\_ENC\_COMP\_MAX\_POINTS[<e>,<AXi>]

- <e>:位置検出器
- <AXi>:軸

必要な補正テーブルの補間点の数は、添付のシステム変数(以下を参照)から計算します。

- 終了位置: \$AA\_ENC\_COMP\_MAX[...]
- 開始位置: \$AA\_ENC\_COMP\_MIN[...]
- 補間点間の距離: \$AA\_ENC\_COMP\_STEP[...]

補間点の数 = ((終了位置 - 開始位置) / 補間点間の距離) + 1

## システム変数

各機械軸および各検出器(検出器がもう 1 台ある場合)に対して、すべての補正関係についての位置関連の補正值および追加のテーブルパラメータを、システム変数の形式で保存する必要があります。

- **\$AA\_ENC\_COMP\_MIN[<e>,<AXi>] (初期位置)**

初期位置は、当該軸の補正テーブルが開始される軸位置です(△ 補間点 0)。

初期位置の補正值は\$AA\_ENC\_COMP[<e>,0,<AXi>]です。

補間点 0 の補正值は、初期位置より小さいすべての位置に使用されます(モジュロ機能のテーブルには適用されません)。

- **\$AA\_ENC\_COMP\_MAX[<e>,<AXi>] (終了位置)**

終了位置は、当該軸の補正テーブルが終了する軸位置(△ 補間点 <k>)です。

終了位置の補正值は\$AA\_ENC\_COMP[<e>,<k>,<AXi>]です。

補間点の補正值は、終了位置より大きいすべての位置に使用されます(モジュロ機能のテーブルは除きます)。

補間点<k>には次の補足条件が適用されます。

- k = MD38000 - 1 の場合:

補正テーブルはすべて利用されています!

- k < MD38000 - 1 の場合:

補正テーブルは完全には利用されていません。テーブルに入力された補正值が k より大きい場合は、無視されます。

- k > MD38000 - 1 の場合:

補正テーブルは、終了位置を減らす制御機能によって制限されます。k より大きい補正值は無視されます。

- **\$AA\_ENC\_COMP\_STEP[<e>,<AXi>] (補間点間の距離)**

補間点間の距離は、関連する補正テーブルの補正值間の距離を定義します。

- **\$AA\_ENC\_COMP[<e>,<N>,<AXi>]** (補正テーブルの補間点 N の訂正值)  
 <N> = 補間点(軸位置)  
 個々の各補間点について、テーブルに補正值を入力してください。  
 <N>は、次のように特定の補正テーブルの補間点の最大数によって制限されます  
 (MD38000 \$MA\_MM\_ENC\_COMP\_MAX\_POINTS)。  
 $0 \leq N \leq \text{MD38000} - 1$   
 補正值のサイズは制限されていません。

#### 注記

最初と最後の補正值は移動範囲全体に渡って有効なままです。つまり、補正テーブルで移動範囲全体を対象としない場合は、これらの値に「0」を設定してください。

- **\$AA\_ENC\_COMP\_IS\_MODULO[<e>,<AXi>]** (モジュロ機能による補正)  
 モジュロ機能による補正を有効化/無効化するシステム変数
  - **\$AA\_ENC\_COMP\_IS\_MODULO[<e>,<AXi>] = 0:**モジュロ機能を使用しない補正
  - **\$AA\_ENC\_COMP\_IS\_MODULO[<e>,<AXi>] = 1:**モジュロ機能を使用する補正
 モジュロ機能による補正が有効である場合、補正テーブルは周期的に繰り返されます。つまり、位置**\$AA\_ENC\_COMP\_MAX** (≒ 補間点 **\$AA\_ENC\_COMP[<e>,<k>,<AXi>]**)の補正值のすぐ後に、位置**\$AA\_ENC\_COMP\_MIN** (≒ 補間点 **\$AA\_ENC\_COMP[<e>,<0>,<AXi>]**)の補正值が続きます。  
 したがって、モジュロ 360°度を持つ回転軸の場合、初期位置として 0° (**\$AA\_ENC\_COMP\_MIN**)、終了位置として 360° (**\$AA\_ENC\_COMP\_MAX**)をプログラム指令することが適切です。  
 この 2 つの位置に入力する補正值は同じにしてください。そうでない場合、切り替え点で MAX から MIN に補正值がジャンプします(逆方向の場合も同様です)。

#### 注意

##### 誤った訂正值

訂正テーブルに訂正值を書き込む場合、パラメータ設定した範囲のすべての補間点に値が割り当てられていることを確認してください。説明されていない訂正值には、ランダムな値が含まれます。

**注記**

位置情報を含むテーブルパラメータは、単位系が変更されると自動的に変換されます (MD10240 \$MN\_SCALING\_SYSTEM\_IS\_METRIC から変更)。

位置情報は常に、現在の単位系で解釈されます。変換は外部でおこなってください。位置データの自動変換は、次のようにして設定できます。

MD10260 \$MN\_CONVERT\_SCALING\_SYSTEM=1

外部変換は、不要になります。

**参照先:**

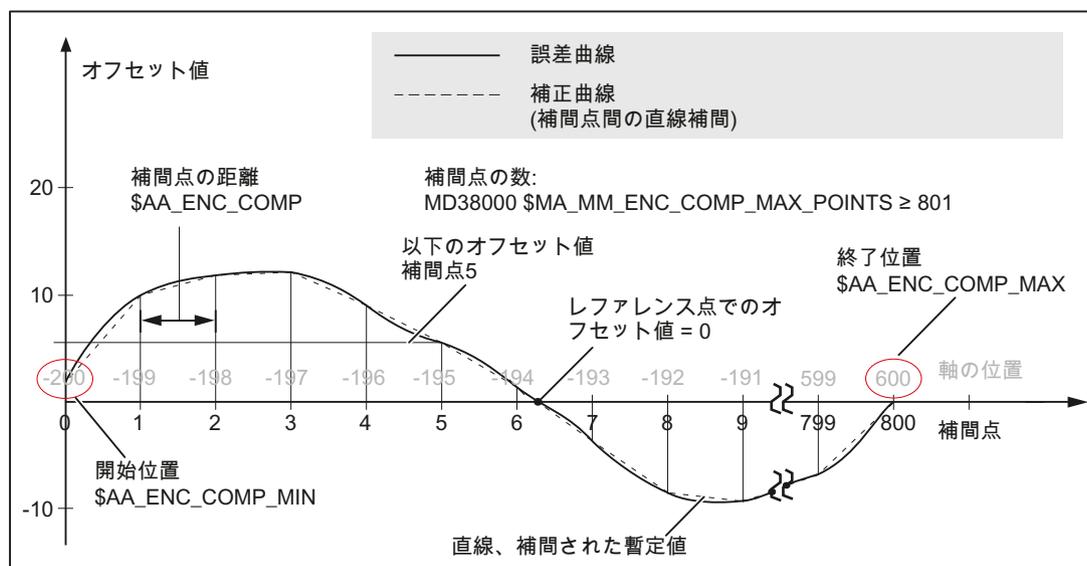
『機能マニュアル、基本機能』；「速度、指令値/フィードバック回路、閉ループ制御 (G2)」

**5.5.2.3 例**

補正テーブルのパラメータ設定の例:

- 機械軸:X1
- 検出器:1
- 開始位置: -200 mm
- 終了位置:600 mm
- 補間点間の距離:1 mm
- 補間点の数:MD38000 \$MA\_MM\_ENC\_COMP\_MAX\_POINTS = ((600 - -200) / 1) + 1 = 801

スタティックユーザーメモリのメモリ要件:801 \* 8 バイト = 6408 バイト



## システム変数の書き込み用プログラム

プログラムコード	コメント
%_N_AX_EEC_INI	
CHANDATA (1)	
\$AA_ENC_COMP[0,0,X1] = 0.003	第1補正值(補間点0): +3 $\mu$ m
\$AA_ENC_COMP[0,1,X1] = 0.01	第2補正值(補間点1): +10 $\mu$ m
\$AA_ENC_COMP[0,2,X1] = 0.012	第3補正值(補間点2): +12 $\mu$ m
...	
\$AA_ENC_COMP[0,800,X1] = -0.0	最後の補正值(補間点800):0 $\mu$ m
\$AA_ENC_COMP_STEP[0,X1] = 1.0	補間点間の距離1.0 mm
\$AA_ENC_COMP_MIN[0,X1] = -200.0	補正は-200.0 mm で開始します
\$AA_ENC_COMP_MAX[0,X1] = 600.0	補正は+600.0mm で終了します
\$AA_ENC_COMP_IS_MODULO[0,X1] = 0	モジュロ機能を使用しない補正
M17	

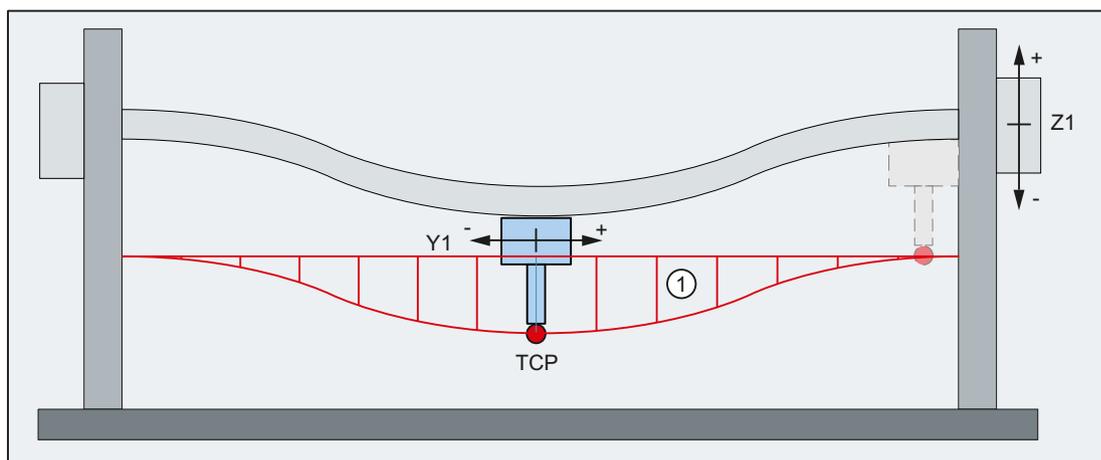
## 5.5.3 真直度および直角度補正

## 5.5.3.1 一般情報

## 機能

真直度誤差と直角度誤差の補正(相互誤差補正、CEC)は軸別補正です。訂正值や補正值が**補正軸**の指令位置に追加されます。補正值は、1本または数本の**基本軸**の現在の指令位置から得られます。

- 真直度誤差  
真直度誤差の場合、機械構成部分またはワークの重量が、工具中心点(TCP)の位置に依存する誤差につながります。
- 直角度誤差  
直角度誤差の場合、ジオメトリ軸の相互の理想的角度の誤差が、工具中心点(TCP)の位置に依存する誤差につながります。



① Y1 の位置に応じた Z1 の位置の誤差

図 5-13 例:真直度誤差

### 誤差補正

誤差補正では、基本軸(Y1)のさまざまな補間点(指令位置)で、特定の位置の誤差を判定し、補正テーブルに入力する必要があります。補間クロック周期での基本軸(Y1)の移動時に、コントロールシステムは、各点の間を直線で補間して、補正軸(Z1)の実際の補正值を計算します。この補正值が、補正軸(Z1)の指令値に加算されます。補正值が正の場合は、補正軸(Z1)が負の方向に移動します。

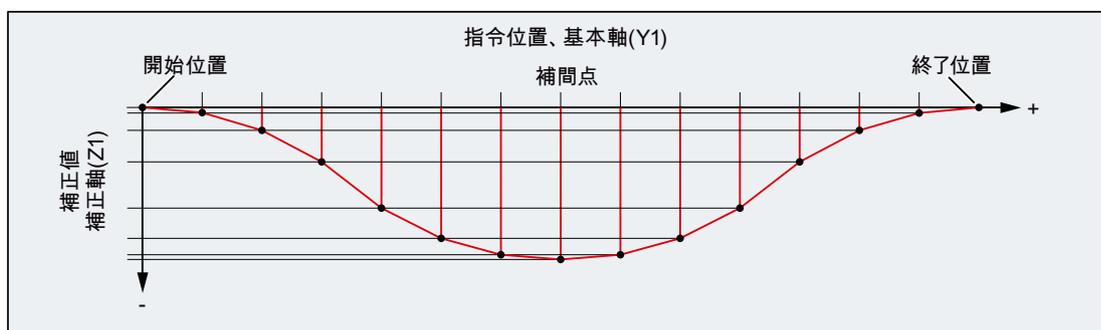


図 5-14 補正テーブル

### 合計補正值

複数の補正テーブルの補正值を 1 本の補正軸に適用できます。これにより得られる合計補正值は、個々の補正值すべての合計です。

### パラメータ設定オプション

- **1本**の軸を、**複数**の補正テーブルの基本軸として定義できます。
- **複数**の補正テーブルが、**1本**の補正軸に作用することができます。総(合計)補正值は、個々の補正テーブルの補正值の合計から得られます。
- 軸は常に、基本軸と補正軸の両方として動作できます。プログラム指令された位置指令値は、補正值の計算で使用されます。
- 補正の処理範囲(基本軸の開始位置と終了位置)と補間点間の距離を各補正テーブルに対して定義できます。
- 補正は、**方向に応じて**作用します。
- すべての補正テーブルには、周期的に使用するための**モジュロ機能**があります。
- 各補正テーブルでは、テーブル内の値の乗算に使用する**加重係数**を考慮してください。
- **テーブルの乗算**を使用して、補正テーブル **A** の実際の補正值  $K_A$  を、任意の補正テーブル **X** の実際の補正值  $K_x$  で乗算できます。テーブル **A** 自体をテーブル **X** として使用することもできます。テーブルの乗算の結果は、補正テーブル **A** の実際の補正值  $K_A$  に加算され、これが補正值の合計  $SK_A$  となり、補正軸に作用します。  

$$SK_A = K_A + K_A * K_x$$
- マシンデータによる、軸のすべての補正関係の**軸別**の有効化:  
MD32710 \$MA\_CEC\_ENABLE[<軸>]
- セットアップデータによる**テーブル別**の有効化:  
SD41300 \$SN\_CEC\_TABLE\_ENABLE[<テーブル>]  
用途例: 有効な補正テーブルをパートプログラム/シンクロナイズドアクションまたは PLC ユーザープログラムで切り替えることによる加工に依存した補正比の変更。

### 用途例

#### 波長の短い反復誤差の補正

軸での波長の短い反復誤差を補正するには、モジュロ機能のある反復誤差成分(波長の短い)用補正テーブルと共に、モジュロ機能のない、もう 1 つの非周期誤差成分用補正テーブルを、同じ軸に対してパラメータ設定します。

#### ユーザー別ピッチ誤差補正

ユーザー別ピッチ誤差を補正するには、同じ軸を持つ補正テーブルを、基本軸および補正軸としてパラメータ設定します。

ただし、この場合は、結果として標準機能とは逆に、検出器の切り替えが自動的に考慮されなくなります。

### 5.5.3.2 試運転:マシンデータ

番号	識別子	意味
NC 別マシンデータ		
MD1024 0	\$MN_SCALING_SYSTEM_IS_METRIC	メトリック基本単位系
MD1026 0	\$MN_CONVERT_SCALING_SYSTEM	基本単位系切り替え有効
MD1834 2	\$MN_MM_CEC_MAX_POIN TS	補正テーブルあたりの補間点の数
軸マシンデータ		
MD3271 0	\$MA_CEC_ENABLE	真直度補正の有効化
MD3271 1	\$MA_CEC_SCALING_SYSTEM_METRIC	真直度補正の単位系
MD3272 0	\$MA_CEC_MAX_SUM	真直度補正の最大補正值
MD3273 0	\$MA_CEC_MAX_VELO	CEC の最大速度の変更

#### 単位系(MD10240, MD10260, MD32711)

コントロールシステムで有効な単位系(メトリックまたはインチ)は、以下を使用して定義します。

MD10240 \$MN\_SCALING\_SYSTEM\_IS\_METRIC = <単位系>

#### 自動変換をおこなわない単位系切り替え

位置データを含むテーブルパラメータは、単位系が変更されると自動的に変換されます。位置データは常に、実際の単位系で解釈されます。変換は外部で行ってください。

#### 自動変換を行う単位系切り替え

位置データの自動変換は、以下を使用して有効にします。

MD10260 \$MN\_CONVERT\_SCALING\_SYSTEM = TRUE

その結果、以下のマシンデータで設定された単位系が、軸のすべての補正テーブルで有効になります。

MD32711 \$MA\_CEC\_SCALING\_SYSTEM\_METRIC = <単位系>

その結果、すべての位置データが、設定した単位系での合計補正值と共に解釈されます。単位系が変更されると、外部での位置データ変換は不要になります。

### 軸別の有効化(MD32710)

軸のすべての補正関係の軸別の有効化は、以下を使用して実行します。

MD32710 \$MA\_CEC\_ENABLE[<軸>] = TRUE

### 補正テーブルあたりの補間点の数(MD18342)

補正テーブルあたりの補間点の数は、以下を使用してパラメータ設定します。

MD18342 \$MN\_MM\_CEC\_MAX\_POINTS[<テーブルインデックス>] = <補間点の数>

補正テーブルに必要な補間点の数は、テーブルのシステム変数 (ページ 343) で設定された最大幅、最小幅、およびステップ(インクリメント)幅から計算します。

<補間点の数> = (\$AN\_CEC\_MAX - \$AN\_CEC\_MIN) / \$AN\_CEC\_STEP + 1

### 監視(MD32720, MD32730)

#### 絶対制限

補正軸の過度の補正動作を避けるために、次のマシンデータで指定した最大値を超えないように合計補正值を監視します。

MD32720 \$MA\_CEC\_MAX\_SUM[<補正軸>] = <最大値>

この値を超えると、合計補正值が最大値に制限され、アラーム 20124 「合計補正值が高すぎます」が表示されます。

#### 変更の制限

補正軸への過度のダイナミック負荷を避けるために、合計補正值の変更を監視して、次のマシンデータで指定した最大値を超えないようにします。

MD32730 \$MA\_CEC\_MAX\_VELO[<補正軸>] = <最大軸速度のパーセント(MD32000 \$MA\_MAX\_AX\_VELO)>

この値を超えると、合計補正值の変更が最大値に制限され、アラーム 20125 「合計補正值の変更が速すぎます」が表示されます。

補正值が制限されなくなるとすぐに、制限により短くなった移動距離が出力されます。

### 5.5.3.3 試運転:セッティングデータ

番号	識別子	意味
SD41300	\$SN_CEC_TABLE_ENABLE	補正テーブルのイネーブル
SD41310	\$SN_CEC_TABLE_WEIGHT	加重係数

#### 補正テーブルのイネーブル(SD41300)

補正テーブルの評価を、以下のセッティングデータを使用して有効にします。

SD41300 \$SN\_CEC\_TABLE\_ENABLE[<テーブルインデックス>] = TRUE

評価を有効にすると、補正テーブルを使用して特定した補正值は、補正軸の合計補正值にのみ加算されます。

#### 加重係数(SD41310)

加重係数をセッティングデータに入力し、補正テーブルから特定した補正值を、このセッティングデータで乗算します。

SD41310 \$SN\_CEC\_TABLE\_WEIGHT[<テーブルインデックス>] = <加重係数>

標準では、加重係数の値は 1.0 です。加重係数が 0.0 の場合、補正テーブルは無効になります。

### 5.5.3.4 試運転:システム変数

識別子	意味
NC 別システム変数	
\$AN_CEC	補正值
\$AN_CEC_DIRECTION	方向への依存
\$AN_CEC_INPUT_AXIS	基本軸
\$AN_CEC_INPUT_NCU	NCU 上の基本軸:
\$AN_CEC_IS_MODULO	モジュロ機能

識別子	意味
\$AN_CEC_MAX	終了位置
\$AN_CEC_MIN	開始位置
\$AN_CEC_MULT_BY_TABLE	乗算
\$AN_CEC_OUTPUT_AXIS	補正軸
\$AN_CEC_OUTPUT_NCU	NCU 上の補正軸
\$AN_CEC_STEP	補間点間の距離
\$AN_CEC_TYPE	表タイプ
軸固有のシステム変数	
\$VA_CEC_COMP_VAL	実際の補正值

### 補正值(\$AN\_CEC)

補正テーブルの補正值をシステム変数に入力する必要があります。

$\$AN\_CEC[\langle \text{テーブルインデックス} \rangle, \langle \text{補間点インデックス} \rangle] = \langle \text{補正值} \rangle$

$\langle \text{補間点インデックス} \rangle = 0 \leq x \leq (\text{MD18342}[\langle \text{テーブルインデックス} \rangle] \text{の値}) - 1$

#### 注記

システム変数\$AN\_CECに書き込む前に、すべての軸のすべての補正機能を無効にしてください。

- MD32700 \$MA\_ENC\_COMP\_ENABLE[<軸>] = 0
- MD32710 \$MA\_CEC\_ENABLE[<軸>] = 0

### 基本軸(\$AN\_CEC\_INPUT\_AXIS)

基本軸をシステム変数に入力する必要があります。つまり、補正テーブルの入力として位置指令値が使用される軸の名称を入力します。

$\$AN\_CEC\_INPUT\_AXIS[\langle \text{テーブルインデックス} \rangle] = \langle \text{チャンネル軸の名称} \rangle$ または" $\langle \text{機械軸の名称} \rangle$ "

### 補正軸(\$AN\_CEC\_OUTPUT\_AXIS)

補正軸をシステム変数に入力する必要があります。つまり、その指令値に補正值が加算される軸の名称を入力します。

---

$\$AN\_CEC\_OUTPUT\_AXIS[<テーブルインデックス>] = "<チャンネル軸の名称>"$ または" $<機械軸の名称>$ "

---

#### 注記

マルチチャンネルシステムでチャンネルおよび機械軸の名称が同じである場合、標準の軸名称 AX1、AX2、などを使用してください。

---

### 補間点間の距離(\$AN\_CEC\_STEP)

2つの補間点間の距離を、補正テーブルのシステム変数(基本軸の位置値)に入力する必要があります。

$\$AN\_CEC\_STEP[<テーブルインデックス>] = <補間点間の距離>$

補正テーブル内では、補間点間の距離は一定に保たれます。

### 開始位置(\$AN\_CEC\_MIN)

最初の補間点の基本軸または補正テーブルの開始の指令位置をシステム変数に入力する必要があります。

$\$AN\_CEC\_MIN[<テーブルインデックス>] = <開始位置>$

---

#### 注記

##### 開始位置より小さい指令位置

最初の補間点の補正值が、開始位置より小さいすべての指令位置で使用されます。

例外:モジュロ機能のある補正テーブル

---

### 終了位置(\$AN\_CEC\_MAX)

最後の補間点の基本軸または補正テーブルの終了の指令位置をシステム変数に入力する必要があります。

$\$AN\_CEC\_MAX[<テーブルインデックス>] = <終了位置>$

---

#### 注記

##### 終了位置より大きい指令位置

最後の補間点の補正值が、開始位置より大きいすべての指令位置で使用されます。

例外:モジュロ機能のある補正テーブル

---

**NCU 上の基本軸または補正軸(\$AN\_CEC\_INPUT\_NCU、\$AN\_CEC\_OUTPUT\_NCU)**

リンク軸による真直補正の場合、基本軸または補正軸として、軸に接続されている NCU の番号をシステム変数に入力する必要があります。

$\$AN\_CEC\_INPUT\_NCU[<テーブルインデックス>]$ または $\$AN\_CEC\_OUTPUT\_NCU[<テーブルインデックス>] = <NCU 番号>$

**方向に依存する補正(\$AN\_CEC\_DIRECTION)**

補正が適用される基本軸の移動方向をシステム変数に入力する必要があります。

$\$AN\_CEC\_DIRECTION[<テーブルインデックス>] = <方向>$

- 0:両方の移動方向
- 1:正の移動方向のみ
- -1:負の移動方向のみ

**テーブルの乗算(\$AN\_CEC\_MULT\_BY\_TABLE)**

テーブルの乗算を使用して、補正テーブルの実際の補正值  $K$  を、任意の補正テーブル  $X$  の実際の補正值  $K_x$  で乗算できます。最初の補正テーブル自体を補正テーブル  $X$  として使用することもできます。テーブルの乗算の結果は、補正テーブルの実際の補正值  $K$  に加算され、これが合計補正值  $SK$  となり、補正軸に作用します。

$$SK = K + K * K_x$$

その補正值を実際の補正值と乗算する補正テーブルの番号をシステム変数に入力する必要があります。

$\$AN\_CEC\_MULT\_BY\_TABLE[<テーブルインデックス>] = <テーブル番号 X>$

$<テーブル番号 X> = <テーブルインデックス X> + 1$

**モジュロ機能(\$AN\_CEC\_IS\_MODULO)**

モジュロ機能を補正テーブルに対して有効にしている場合は、入力値(基本軸の指令位置)が、補正テーブルの入力範囲によりモジュロが計算されます。これは、終了の補正值の後に再び開始位置の補正值が続く、または移動方向が逆の場合は、開始位置の補正值の後に再び終了位置の補正值が続くことを意味します。

$\$AN\_CEC\_IS\_MODULO[<テーブルインデックス>] = <値>$

- 0:モジュロ機能を使用しない補正
- 1:モジュロ機能による補正

**注記****等しい補正值**

モジュロ機能が有効な場合は、開始位置と終了位置の補正值を同じ値に設定することを推奨します。

**注記****モジュロ回転軸**

モジュロ回転軸には、補正のモジュロ機能を有効にしてください。

**例:モジュロ回転軸のパラメータ設定**

```

$MA_IS_ROT_AX[AX1] = 1      ; 回転軸
$MA_ROT_IS_MODULO[AX1] = 1 ; モジュロ 360°
$AN_CEC_INPUT_AXIS[0] = AX1
$AN_CEC_MIN[0]=0.0
$AN_CEC_MAX[0] = 360.0
$AN_CEC_STEP[0]=1.0
$AN_CEC_IS_MODULO[0] = 1
$MN_MM_CEC_MAX_POINTS = 361
$AN_CEC[0, 0] = $AN_CEC[0, 360] = 0.1

```

**テーブルタイプ(\$AN\_CEC\_TYPE)**

テーブルタイプ(補正タイプ)をシステム変数に入力する必要があります。

**\$AN\_CEC\_TYPE[<テーブルインデックス>] = <タイプ>**

- 0:一般的な補間補正
- 1:シリンダエラー補正

**実際の補正值(\$VA\_CEC\_COMP\_VAL)**

システム変数で、軸で現在有効な補正值が提供されます。

**<実際の補正值> = \$VA\_CEC\_COMP\_VAL[<軸>]**

## 5.5.3.5 試運転:基本手順

最初のセットアップステップで、補正テーブルを定義します。これを行うには、特定の補正テーブルに対して必要な補間点の数を設定します。コントロールシステム内に補正テーブルが作成され、次のウォーム再起動時に初期値が入力されます。

2番目のセットアップステップでは、システム変数を使用して補正データをパラメータ設定します。これは、次の異なる2つの方法で実行できます。

- システム変数の書き込み先となる NC プログラムを起動します。
- 外部コンピュータからコントローラへ補正テーブルを伝送します。

**注記**

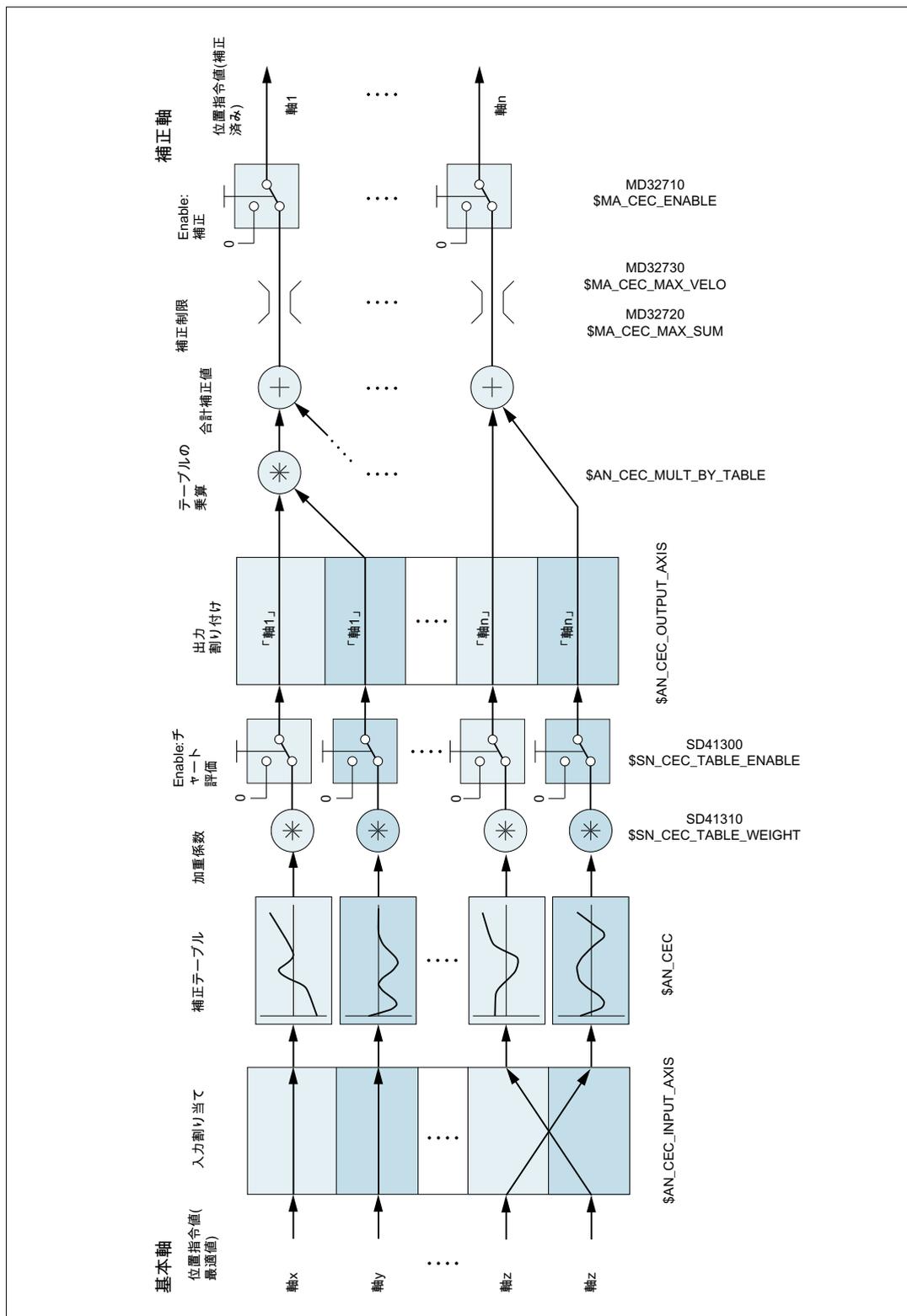
補正テーブルをロードする前に、すべての軸で補正機能をすべて無効にしてください。

- MD32700 \$MA\_ENC\_COMP\_ENABLE[<軸>] == 0
- MD32710 \$MA\_CEC\_ENABLE[<軸>] == 0

**手順**

1. 補正テーブルの補間点の数をパラメータ設定します。  
MD18342 \$MN\_MM\_CEC\_MAX\_POINTS
2. 監視機能のパラメータ設定:
  - 絶対制限:MD32720 \$MA\_CEC\_MAX\_SUM
  - 変更の制限:MD32730 \$MA\_CEC\_MAX\_VELO
3. 単位系切り替えのパラメータ設定
  - 自動変換なし:  
MD10240 \$MN\_SCALING\_SYSTEM\_IS\_METRIC
  - 自動変換あり:
    - MD10240 \$MN\_SCALING\_SYSTEM\_IS\_METRIC
    - MD10260 \$MN\_CONVERT\_SCALING\_SYSTEM
    - MD32711 \$MA\_CEC\_SCALING\_SYSTEM\_METRIC
4. NC のウォーム再起動を開始してマシンデータの変更を有効にします。
5. 補正テーブルの加重係数をパラメータ設定します。  
SD41310\$SN\_CEC\_TABLE\_WEIGHT
6. システム変数\$AN\_CEC\_...でテーブルパラメータを設定します。
7. 基本軸と補正軸の実際の検出器を確認または原点確立します。  
DB31、... DBX60.4 bzw.60.5 == 1 (原点確立済み/原点同期済み 1 または 2)
8. 補正の有効化:
  - テーブル別:SD41300 CEC\_TABLE\_ENABLE
  - 軸別:MD32710 \$MA\_CEC\_ENABLE

5.5.3.6 試運転:概要図



## 5.5.3.7 例 1:真直度補正

軸 Y1 の位置に応じて、軸 Z1 の設定位置に追加の補正値が適用されます。

使用する補正テーブル:テーブル 1 ⇒ インデックス= 0

## 補正パラメータ

- 開始位置: -400.0
- 終了位置:400.0
- 補間点間の距離:8.0

## 補間点の数

$MD18342 \$MN\_MM\_CEC\_MAX\_POINTS[0] = (400.0 - -400.0) / 8.0 + 1 = 101$

不揮発性ユーザーメモリに必要なメモリは最小 808 バイトです(補正値毎に 8 バイト)。

プログラムコード	コメント
%_N_NC_CEC_INI	; 補正データの書き込み
CHANDATA(1)	; 補正テーブル 1、インデックス 0
\$AN_CEC[0,0]=0	; 最初の補正値= 0μm
\$AN_CEC[0,1]=0.01	; 2 番目の補正値= 10μm
\$AN_CEC[0,2]=0.012	; 3 番目の補正値= 12μm
...	
\$AN_CEC[0,100]=0	; 101 番目の補正値= 0μm
\$AN_CEC_INPUT_AXIS[0]=AX2	; 基本軸 Y1 □ 機械軸の名称 AX2
\$AN_CEC_OUTPUT_AXIS[0]=AX3	; 補正軸 Z1 □ 機械軸の名称 AX3
\$AN_CEC_STEP[0]=8.0	; 補間点間の距離 8.0 mm
\$AN_CEC_MIN[0]=-400.0	; 開始位置:Y1 = -400mm
\$AN_CEC_MAX[0]=400.0	; 終了位置: Y1 = +400mm
\$AN_CEC_DIRECTION[0]=0	; Y1 の両方の移動方向で補正
\$AN_CEC_MULT_BY_TABLE[0] = 0	; テーブルの乗算なし
\$AN_CEC_IS_MODULO[0]=0	; モジュロ機能なし



**補正テーブル 1(テーブルインデックス:0)**

- 基本軸:X1
- 補正軸:X1
- 補正值:位置に依存する傾斜角  $\beta(X1)$  の正弦。この場合、 $\beta = f(X1)$  の位置)

**補正テーブル 2(テーブルインデックス:1)**

- 基本軸:Z1
- 補正軸:X1
- 補正值:軸 X1 の計測位置での軸 Z1 の位置の動作

**テーブルの乗算**

補正テーブル 1 の場合(テーブルインデックス:0)は、テーブルの乗算を補正関係 2 で設定します。

`$AN_CEC_MULT_BY_TABLE[ 0 ] = 2`

**5.5.3.9 例 3 :補正值の二次元配列**

フラットベッド型の機械の場合、Z 軸の真直度補正值が X 軸と Y 軸の軸位置に依存するこの使用事例が現実に頻繁に発生します。そのため、二次元配列で補正值を編成することには意味があります。

この例では、サイズが 4 x 5 (行 x 列) のグリッドを使用して真直度補正を実行する方法の一例を詳しく説明しています。計測範囲全体のサイズは、2000 x 900 mm<sup>2</sup> です。各補正值は、X 軸に沿って 500 mm、Y 軸に沿って 300 mm のインクリメントで計測されます。当該の補正值を含む補間点は、座標(X-Y 面)の交点上に配置されます。これらの補間点間の補正值は直線で補間されます。

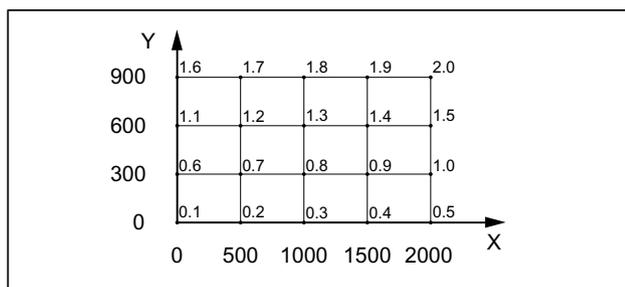


図 5-17 Z 軸の補正值

## 注記

## SINUMERIK Operate ユーザーインターフェース

真直度補正の結果、Z 軸の現在有効な補正值が、以下の SINUMERIK Operate ユーザーインターフェースに表示されます。

操作エリア: 「診断」 > 「軸の診断」 > 「サービス軸」 > 信号: 「補正、真直度 + 熱変位補正」

## 実行

4 行グリッドの各行について、補正テーブルが、5 か所の補間点で設定されます。第 1 の補正テーブルに補正值 0.1 ~ 0.5 が、補間点 1 から昇順で第 1 行に入力されます。同じ手順を使用して、第 1 の補正テーブルに補正值 0.6 ~ 1.0 が第 2 行に入力され、以降、同様に入力されます。これで、X 軸の位置に関連するこれらの補正テーブルは **f テーブル** として、テーブルの値は **f<sub>i</sub>(x)** として指定されます。ここで i はテーブル番号です。

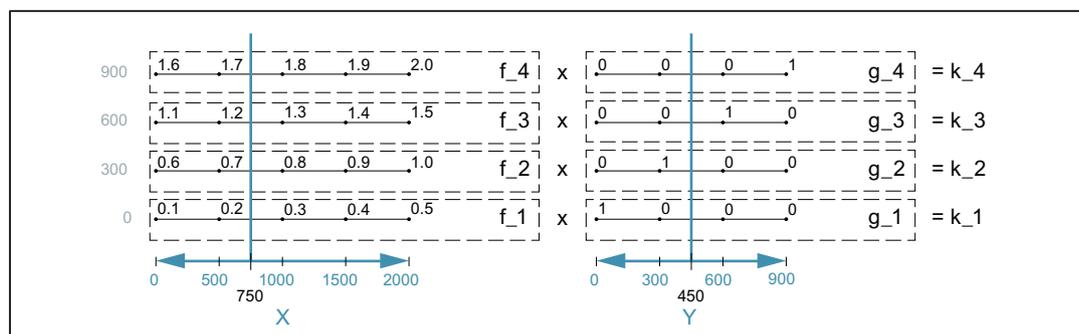
Y 軸の位置を考慮するために、追加の補正テーブルが必要です。これらの補正テーブルは **g テーブル** として、テーブルの値は **g<sub>i</sub>(y)** として指定されます。f テーブルと g テーブルの数は一致します。例では、

g テーブルでは、1 つの補正值が 1 に設定され、その他はすべて 0 に設定されます。テーブル内の補正值 1 の位置は、テーブル番号で決定されます。第 1 の g テーブルでは、補正值 1 は第 1 補間点、第 2 の g テーブルでは第 2 補間点に位置します。以降の g テーブルについても同様です。g テーブルに f テーブルを乗算することによって、各 f テーブルの適切な補正值に 1 が乗算され、その補正值が選択されます。すべての不適切な補正值は 0 で乗算してマスクされます。

次の式に従って、位置(x/y)で合計補正值  $D_z$  を計算します。

$$D_z(x/y) = f_1(x) * g_1(y) + f_2(x) * g_2(y) + f_3(x) * g_3(y) + f_4(x) * g_4(y) = k_1 + k_2 + k_3 + k_4$$

ロケーション  $D_z(750/450)$  で、Z 軸の合計補正值を計算し、次の値が得られます。



## 5.5 補間補正

$$\begin{aligned}
 Dz(750/450) &= f_1(750) * g_1(450) + f_2(750) * g_2(450) + \\
 &\quad f_3(750) * g_3(450) + f_4(750) * g_4(450) \\
 &= 0.25 * 0.0 + 0.75 * 0.5 + \\
 &\quad 1.25 * 0.5 + 1.75 * 0.0 \\
 &= 1.0
 \end{aligned}$$

## パラメータ設定

## NC プログラムを使用したマシンデータのパラメータ設定

プログラムコード	コメント
; 補正テーブルの補間点の数	
\$MN_MM_CEC_MAX_POINTS[0]=5	; 補正テーブル 1
\$MN_MM_CEC_MAX_POINTS[1]=5	; 補正テーブル 2
\$MN_MM_CEC_MAX_POINTS[2]=5	; 補正テーブル 3
\$MN_MM_CEC_MAX_POINTS[3]=5	; 補正テーブル 4
\$MN_MM_CEC_MAX_POINTS[4]=4	; 補正テーブル 5
\$MN_MM_CEC_MAX_POINTS[5]=4	; 補正テーブル 6
\$MN_MM_CEC_MAX_POINTS[6]=4	; 補正テーブル 7
\$MN_MM_CEC_MAX_POINTS[7]=4	; 補正テーブル 8
; 第 3 の機械軸に対応する軸 z1 の監視機能	
\$MA_CEC_MAX_SUM[AX3]=10.0	; 最大合計補正值
\$MA_CEC_MAX_VELO[AX3]=100.0	; 変更の最大値

## NC プログラムを使用したテーブルパラメータ(システム変数)の設定

プログラムコード	コメント
; 補正テーブルに書き込むには、	
; 最初に軸 z1 (補正軸) の補正を	
; 無効にしてください。	
\$MA_CEC_ENABLE[Z1] = FALSE	
NEWCONF	; \$MA_CEC_ENABLE を有効にします
; f テーブルの値 f_i(x) を定義します。	
; インデックス[0] のテーブルの関数値 f_1(x)	
\$AN_CEC[0,0]=0.1	
\$AN_CEC[0,1]=0.2	
\$AN_CEC[0,2]=0.3	
\$AN_CEC[0,3]=0.4	
\$AN_CEC[0,4]=0.5	

プログラムコード	コメント
<pre>; インデックス [1] のテーブルの関数値 f_2 (x) \$AN_CEC [1, 0]=0.6 \$AN_CEC [1, 1]=0.7 \$AN_CEC [1, 2]=0.8 \$AN_CEC [1, 3]=0.9 \$AN_CEC [1, 4]=1.0</pre>	
<pre>; インデックス [2] のテーブルの関数値 f_3 (x) \$AN_CEC [2, 0]=1.1 \$AN_CEC [2, 1]=1.2 \$AN_CEC [2, 2]=1.3 \$AN_CEC [2, 3]=1.4 \$AN_CEC [2, 4]=1.5</pre>	
<pre>; インデックス [3] のテーブルの関数値 f_4 (x) \$AN_CEC [3, 0]=1.6 \$AN_CEC [3, 1]=1.7 \$AN_CEC [3, 2]=1.8 \$AN_CEC [3, 3]=1.9 \$AN_CEC [3, 4]=2.0</pre>	
<pre>; 補正値を含む f テーブルの評価を有効にします \$SN_CEC_TABLE_ENABLE [0]=TRUE \$SN_CEC_TABLE_ENABLE [1]=TRUE \$SN_CEC_TABLE_ENABLE [2]=TRUE \$SN_CEC_TABLE_ENABLE [3]=TRUE</pre>	
<pre>; f テーブルの加重係数を定義します \$SN_CEC_TABLE_WEIGHT [0]=1.0 \$SN_CEC_TABLE_WEIGHT [1]=1.0 \$SN_CEC_TABLE_WEIGHT [2]=1.0 \$SN_CEC_TABLE_WEIGHT [3]=1.0</pre>	
<pre>; 次のテーブルパラメータを変更しても、 ; ウォーム再起動後に ; 基本軸 X1 を定義します \$AN_CEC_INPUT_AXIS [0]=(X1) \$AN_CEC_INPUT_AXIS [1]=(X1) \$AN_CEC_INPUT_AXIS [2]=(X1) \$AN_CEC_INPUT_AXIS [3]=(X1)</pre>	
<pre>; 補正軸 Z1 を定義します \$AN_CEC_OUTPUT_AXIS [0]=(Z1) \$AN_CEC_OUTPUT_AXIS [1]=(Z1)</pre>	

プログラムコード	コメント
\$AN_CEC_OUTPUT_AXIS[2]=(Z1)	
\$AN_CEC_OUTPUT_AXIS[3]=(Z1)	
;	f テーブルで補正值の補間点間の距離を定義します
\$AN_CEC_STEP[0]=500.0	
\$AN_CEC_STEP[1]=500.0	
\$AN_CEC_STEP[2]=500.0	
\$AN_CEC_STEP[3]=500.0	
;	補正は X1 = 0 で開始します
\$AN_CEC_MIN[0]=0.0	
\$AN_CEC_MIN[1]=0.0	
\$AN_CEC_MIN[2]=0.0	
\$AN_CEC_MIN[3]=0.0	
;	補正は X1 = 2000 で終了します
\$AN_CEC_MAX[0]=2000.0	
\$AN_CEC_MAX[1]=2000.0	
\$AN_CEC_MAX[2]=2000.0	
\$AN_CEC_MAX[3]=2000.0	
;	インデックス[t1]を持つ f テーブルの値と g テーブルの値を
;	値[t2]で乗算します
;	上で指定した式(アルゴリズム)に従います
\$AN_CEC_MULT_BY_TABLE[0] = 5	
\$AN_CEC_MULT_BY_TABLE[1] = 6	
\$AN_CEC_MULT_BY_TABLE[2] = 7	
\$AN_CEC_MULT_BY_TABLE[3] = 8	
;	g <sub>i</sub> (y) の g テーブル値を定義します:
;	インデックス[4]のテーブルの関数値 g <sub>1</sub> (x)
\$AN_CEC[4,0]=1.0	
\$AN_CEC[4,1]=0.0	
\$AN_CEC[4,2]=0.0	
\$AN_CEC[4,3]=0.0	
;	インデックス[5]のテーブルの関数値 g <sub>2</sub> (x)
\$AN_CEC[5,0]=0.0	
\$AN_CEC[5,1]=1.0	
\$AN_CEC[5,2]=0.0	
\$AN_CEC[5,3]=0.0	
;	インデックス[6]のテーブルの関数値 g <sub>3</sub> (x)
\$AN_CEC[6,0]=0.0	

プログラムコード	コメント
\$AN_CEC[6,1]=0.0	
\$AN_CEC[6,2]=1.0	
\$AN_CEC[6,3]=0.0	
; インデックス[7]のテーブルの関数値 g_4 (x)	
\$AN_CEC[7,0]=0.0	
\$AN_CEC[7,1]=0.0	
\$AN_CEC[7,2]=0.0	
\$AN_CEC[7,3]=1.0	
; 補正値を含む g テーブルの評価を有効にします	
\$SN_CEC_TABLE_ENABLE[4]=TRUE	
\$SN_CEC_TABLE_ENABLE[5]=TRUE	
\$SN_CEC_TABLE_ENABLE[6]=TRUE	
\$SN_CEC_TABLE_ENABLE[7]=TRUE	
; g テーブルの加重係数を定義します	
\$SN_CEC_TABLE_WEIGHT[4]=1.0	
\$SN_CEC_TABLE_WEIGHT[5]=1.0	
\$SN_CEC_TABLE_WEIGHT[6]=1.0	
\$SN_CEC_TABLE_WEIGHT[7]=1.0	
; 次のテーブルパラメータを変更しても、	
; 電源が投入されるまでは有効になりません	
; 基本軸 Y1 を定義します	
\$AN_CEC_INPUT_AXIS[4]=(Y1)	
\$AN_CEC_INPUT_AXIS[5]=(Y1)	
\$AN_CEC_INPUT_AXIS[6]=(Y1)	
\$AN_CEC_INPUT_AXIS[7]=(Y1)	
; 補正軸 Z1 を定義します	
\$AN_CEC_OUTPUT_AXIS[4]=(Z1)	
\$AN_CEC_OUTPUT_AXIS[5]=(Z1)	
\$AN_CEC_OUTPUT_AXIS[6]=(Z1)	
\$AN_CEC_OUTPUT_AXIS[7]=(Z1)	
; g テーブルの補正値の補間点間の距離を定義します	
\$AN_CEC_STEP[4]=300.0	
\$AN_CEC_STEP[5]=300.0	
\$AN_CEC_STEP[6]=300.0	
\$AN_CEC_STEP[7]=300.0	
; 補正は Y1 = 0 で開始します	
\$AN_CEC_MIN[4]=0.0	
\$AN_CEC_MIN[5]=0.0	

プログラムコード	コメント
\$AN_CEC_MIN[6]=0.0	
\$AN_CEC_MIN[7]=0.0	
;補正は Y1 = 900 で終了します	
\$AN_CEC_MAX[4]=900.0	
\$AN_CEC_MAX[5]=900.0	
\$AN_CEC_MAX[6]=900.0	
\$AN_CEC_MAX[7]=900.0	
; 補正を再度起動します	
\$MA_CEC_ENABLE[Z1]=TRUE	
NEWCONF	
; プログラムテストを実行して、補正が有効かどうかをチェックします	
G01 F1000 X0 X0 Z0 G90	
R1=0 R2=0	
LOOP_Y:	
LOOP_X:	
STOPRE	
X=R1 Y=R2	
M0	; CEC 値を確認するまで待機します
R1=R1+500	
IF R1 <=2000 GOTOB LOOP_X	
R1=0	
R2=R2+300	
IF R2<=900 GOTOB LOOP_Y	

## 5.5.4 NCU リンクを使用した真直度補正の拡張 - 840D sl のみ

### 5.5.4.1 一般的なパラメータ割り付け

#### 基本軸と補正軸

補正テーブルの基本軸(入力)と補正軸(出力)のパラメータ設定は、次のシステム変数を使用して、リンク軸 (ページ 92)と組み合わせて実行します。

- \$AN\_CEC\_INPUT\_NCU
- \$AN\_CEC\_INPUT\_AXIS
- \$AN\_CEC\_OUTPUT\_NCU
- \$AN\_CEC\_OUTPUT\_AXIS

次の必要条件を遵守してください。

- 真直度補正の基本軸と補正軸は、同じ NCU 上でチャンネル軸として保管する必要があります。
- 対応する機械軸が異なる NCU に接続できること。
- システム変数は、NCU のウォーム再起動後にのみ有効になります。
- データバックアップは、機械軸名称で行われます。

#### 真直度補正の一般的なセットアップ

真直度補正の一般的なセットアップについては、「試運転:基本手順 (ページ 348)」の章を参照してください。

### 5.5.4.2 チャンネル名称を使ったパラメータ設定

チャンネル軸名称を使った、基本軸と補正軸のパラメータ設定は、以下のシステム変数によっておこないます。

- 入力軸  
\$AN\_CEC\_INPUT\_AXIS[<CEC テーブル番号>] = <チャンネル軸名称>
- 出力軸  
\$AN\_CEC\_OUTPUT\_AXIS[<CEC テーブル番号>] = <チャンネル軸名称>

#### 例:軸「ZZ」(AX2、NCU 2)を「XR」(AX1、NCU 1)に連結

チャンネル 1 用に TP1、チャンネル 2 用に TP2 の 2 つのパートプログラムを作成してください。補正テーブルの基本軸と補正軸のシステム変数が、これらに書き込まれます。

TP1 からの抜粋、チャンネル 1: \$AN\_CEC\_INPUT\_AXIS[0] = "XR"

TP2 からの抜粋、チャンネル 2: \$AN\_CEC\_OUTPUT\_AXIS[0] = "ZZ"

### 5.5.4.3 機械軸名称を使ったパラメータ設定

機械軸名称を使った、基本軸と補正軸のパラメータ設定は、以下のシステム変数によって行います。

- 基本軸
  - \$AN\_CEC\_INPUT\_NCU[<CEC テーブル番号>] = <NCU 番号>
  - \$AN\_CEC\_INPUT\_AXIS[<CEC テーブル番号>] = <機械軸名称>
- 補正軸
  - \$AN\_CEC\_OUTPUT\_NCU[<CEC テーブル番号>] = <NCU 番号>
  - \$AN\_CEC\_OUTPUT\_AXIS[<CEC テーブル番号>] = <チャンネル軸名称>

**例:軸「ZZ」(AX2、NCU 2)を「XR」(AX1、NCU 1)に連結**

NCU 1 の任意のチャンネルで起動できるパートプログラムを作成してください。機械軸名称と NCU 番号を含む補正テーブルの基本軸と補正軸のシステム変数が、これらに書き込まれます。

パートプログラム:

```
$AN_CEC_INPUT_NCU[0] = 1
$AN_CEC_INPUT_AXIS[0] = "AX1"
$AN_CEC_OUTPUT_NCU[0] = 2
$AN_CEC_OUTPUT_AXIS[0] = "AX2"
```

**注記**

- NCU 番号は、NC プログラムに、軸名称より前に書き込んでください。
- NC1\_AX1 と NC2\_AX2 間では、真直度補正を実行できません。

**5.5.4.4****軸コンテナ**

進捗度補正は、コンテナ軸 (ページ 117) と組み合わせて使用することもできます。ただし、コンテナ軸を使用する場合は、各軸コンテナ回転後に、真直度補正に関連して他の機械軸が連結されます。軸コンテナ回転後であっても、各回転後に 2 つの特定の機械軸間の連結を維持するために、ユーザーが、基本軸と補正軸をパラメータ設定する適切なシステム変数を補正テーブルに新しく書き込む必要があります。

**必要条件**

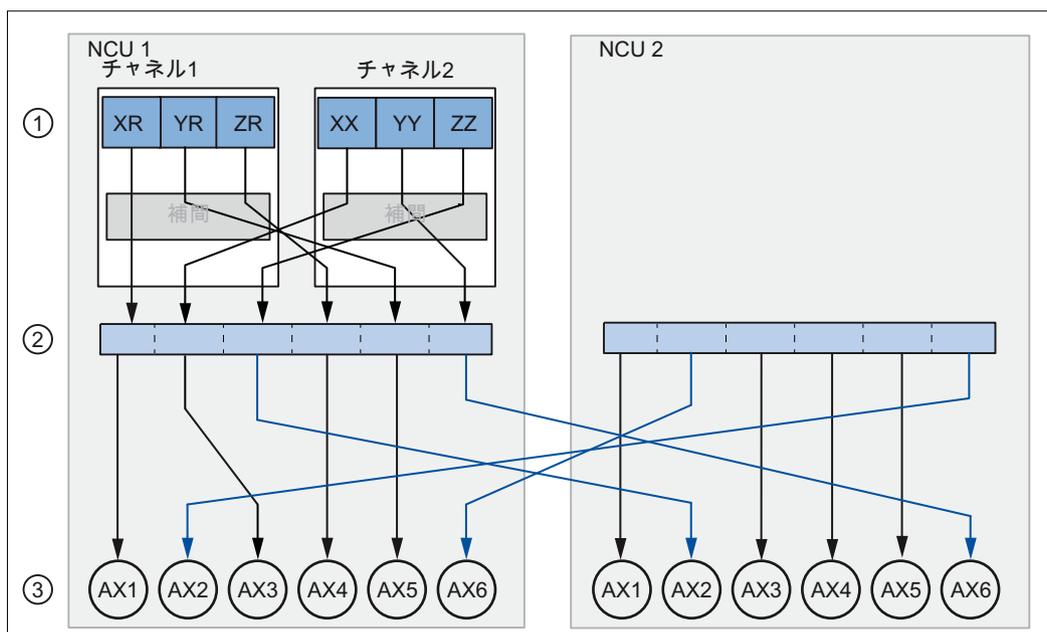
真直度補正が有効になったときに、軸コンテナが開始位置( $\$AN\_AXCTAS == 0$ )にあること。

**通知****機械軸の変更**

軸コンテナ回転後、同じチャンネル軸が連結を維持します。連結された機械軸を変更することができます。

### 5.5.4.5 構成例 1:固定結合の NCU リンク

構成例 1 では、NCU 1 の 2 つのチャンネル、およびマシンデータ MD20080 `$MC_AXCONF_CHANAX_NAME_TAB` で定義されたチャンネル軸名称を示します。NCU 2 の構成は示されません。



- ① 機械軸へのチャンネル軸の割り当て:  
MD20070 `$MC_AXCONF_MACHAX_USED[0 ... 2]`
- ② 実際の機械軸への論理機械軸の割り当て:  
MD10002 `$MN_AXCONF_LOGIC_MACHAX_TAB`
- ③ 機械軸名称:  
MD10000 `$MN_AXCONF_MACHAX_NAME_TAB`

図 5-18 構成 1:固定結合の NCU リンク

チャンネル軸 ZZ (NCU1、チャンネル 2)は、NCU 2 の機械軸 AX2 に接続されています。これは、NCU 2 の機械軸 AX2 が NCU 1 の第 2 チャンネルで移動するように、軸 ZZ が移動することを意味します。たとえば、NC プログラムで以下の指令を使用します。  
`POS[ZZ]=10 FA[ZZ]=1000`

#### パラメータ設定

両方の NCU のマシンデータを、次のリストにある NC プログラム NC\_P1 および NC\_P2 を使用してパラメータ設定します。

## NC プログラム NC\_P1:NCU 1 のマシンデータのパラメータ設定

## プログラムコード

```

; ##### NCU1 #####
; NC 別マシンデータ
$MN_NCU_LINKNO = 1
$MN_MM_NCU_LINK_MASK = 1
$MN_MM_LINK_NUM_OF_MODULES= 2
$MN_MM_SERVO_FIFO_SIZE = 3
$MN_ASSIGN_CHAN_TO_MODE_GROUP[1]=1
$MN_AXCONF_LOGIC_MACHAX_TAB[0] = 「NC1_AX1」
$MN_AXCONF_LOGIC_MACHAX_TAB[1] = 「NC1_AX3」
$MN_AXCONF_LOGIC_MACHAX_TAB[2] = 「NC2_AX2」
$MN_AXCONF_LOGIC_MACHAX_TAB[3] = 「NC1_AX4」
$MN_AXCONF_LOGIC_MACHAX_TAB[4] = 「NC1_AX5」
$MN_AXCONF_LOGIC_MACHAX_TAB[5] = 「NC2_AX6」
; チャネル別マシンデータ:チャネル 1
CHANDATA (1)
$MC_AXCONF_MACHAX_USED[0]=1
$MC_AXCONF_MACHAX_USED[1]=5
$MC_AXCONF_MACHAX_USED[2]=4
$MC_AXCONF_MACHAX_USED[3]=0
$MC_AXCONF_MACHAX_USED[4]=0
$MC_AXCONF_MACHAX_USED[5]=0
$MC_AXCONF_CHANAX_NAME_TAB[0] = 「XR」
$MC_AXCONF_CHANAX_NAME_TAB[1] = 「YR」
$MC_AXCONF_CHANAX_NAME_TAB[2] = 「ZR」
; チャネル別マシンデータ:チャネル 2
CHANDATA (2)
$MC_REFP_NC_START_LOCK=0
$MC_AXCONF_MACHAX_USED[0]=2
$MC_AXCONF_MACHAX_USED[1]=6
$MC_AXCONF_MACHAX_USED[2]=3
$MC_AXCONF_MACHAX_USED[3]=0
$MC_AXCONF_MACHAX_USED[4]=0
$MC_AXCONF_MACHAX_USED[5]=0
$MC_AXCONF_CHANAX_NAME_TAB[0] = 「XX」
$MC_AXCONF_CHANAX_NAME_TAB[1] = 「YY」
$MC_AXCONF_CHANAX_NAME_TAB[2] = 「ZZ」
M30

```

## NC プログラム NC\_P2:NCU 2 のマシンデータのパラメータ設定

```

プログラムコード
; ##### NCU-2 #####
; NC 別マシンデータ
$MN_NCU_LINKNO = 2
$MN_MM_NCU_LINK_MASK = 1
$MN_MM_LINK_NUM_OF_MODULES= 2
$MN_MM_SERVO_FIFO_SIZE = 3
$MN_AXCONF_LOGIC_MACHAX_TAB[0] = 「NC2_AX1」
$MN_AXCONF_LOGIC_MACHAX_TAB[1] = 「NC1_AX6」
$MN_AXCONF_LOGIC_MACHAX_TAB[2] = 「NC2_AX3」
$MN_AXCONF_LOGIC_MACHAX_TAB[3] = 「NC2_AX4」
$MN_AXCONF_LOGIC_MACHAX_TAB[4] = 「NC2_AX5」
$MN_AXCONF_LOGIC_MACHAX_TAB[5] = 「NC1_AX2」
; チャンネル別マシンデータ:チャンネル 1
CHANDATA(1)
$MC_AXCONF_MACHAX_USED[0]=1
$MC_AXCONF_MACHAX_USED[1]=2
$MC_AXCONF_MACHAX_USED[2]=3
$MC_AXCONF_MACHAX_USED[3]=4
$MC_AXCONF_MACHAX_USED[4]=5
$MC_AXCONF_MACHAX_USED[5]=6
$MC_AXCONF_MACHAX_USED[6]=0
M30

```

### 5.5.4.6 構成例 2: 軸コンテナとの NCU リンク

構成例 2 には、軸コンテナ (ページ 117)CT1 が含まれます。軸コンテナに含まれる軸のパラメータ設定は、MD12701 \$MN\_AXCT\_AXCONF\_ASSIGN\_TAB1 によっておこなわれます。軸コンテナは、すべての NCU に 1 つのみ存在するマルチ NCU オブジェクトです。

チャンネル軸 YR の論理機械軸はスロット CT1\_SL3 を、チャンネル軸 YY はスロット CT1\_SL4 を基準にしています。NCU2 のチャンネル軸は示していません。

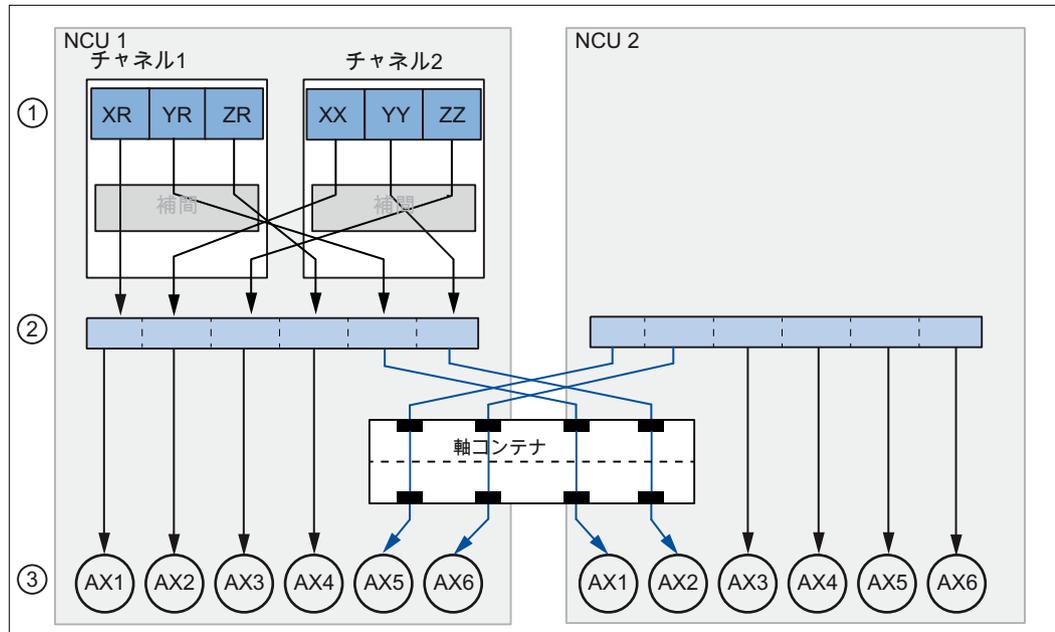
コンテナ軸は、以下の機械軸に割り当てられます。

- NC1\_AX5
- NC1\_AX6

- NC2\_AX1
- NC2\_AX2

軸コンテナの初期状態での軸の組み合わせ

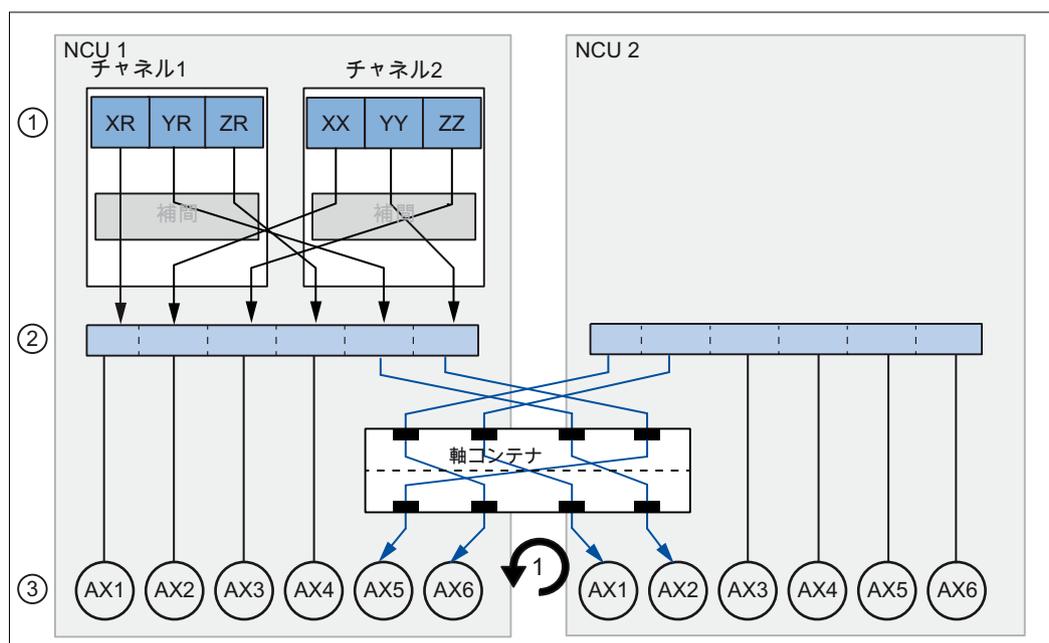
NCU の電源投入後の軸コンテナの初期状態では、チャンネル軸 YR が NC2\_AX1 に、チャンネル軸 YY が NC2\_AX2 に接続されています。



- ① MD20070 \$MC\_AXCONF\_MACHAX\_USED.
- ② MD10002 \$MN\_AXCONF\_LOGIC\_MACHAX\_TAB
- ③ MD10000 \$MN\_AXCONF\_MACHAX\_NAME\_TAB

軸コンテナの 1 ステップ後の軸の組み合わせ

軸コンテナが 1 ステップ(インクリメント)回転した後、チャンネル軸 YR は NC2\_AX2 に、チャンネル軸 YY は NC1\_AX5 に接続されます。



- ① MD20070 \$MC\_AXCONF\_MACHAX\_USED.
- ② MD10002 \$MN\_AXCONF\_LOGIC\_MACHAX\_TAB
- ③ MD10000 \$MN\_AXCONF\_MACHAX\_NAME\_TAB

## NC プログラム NC\_P1:NCU 1 のマシンデータのパラメータ設定

### プログラムコード

```

; ##### NCU1 #####
; NC 別マシンデータ
$MN_NCU_LINKNO = 1
$MN_MM_NCU_LINK_MASK = 1
$MN_MM_LINK_NUM_OF_MODULES= 2
$MN_MM_SERVO_FIFO_SIZE = 3
$MN_ASSIGN_CHAN_TO_MODE_GROUP[1]=1
$MN_AXCONF_LOGIC_MACHAX_TAB[0] = 「NC1_AX1」
$MN_AXCONF_LOGIC_MACHAX_TAB[1] = 「NC1_AX3」
$MN_AXCONF_LOGIC_MACHAX_TAB[2] = 「NC2_AX2」
$MN_AXCONF_LOGIC_MACHAX_TAB[3] = 「NC1_AX4」
$MN_AXCONF_LOGIC_MACHAX_TAB[4] = 「CT1_SL3」
$MN_AXCONF_LOGIC_MACHAX_TAB[5] = 「CT1_SL4」
$MN_AXCT_AXCONF_ASSIGN_TAB1[0] = 「NC1_AX5」
$MN_AXCT_AXCONF_ASSIGN_TAB1[1] = 「NC1_AX6」
$MN_AXCT_AXCONF_ASSIGN_TAB1[2] = 「NC2_AX1」
$MN_AXCT_AXCONF_ASSIGN_TAB1[3] = 「NC2_AX2」

```

## プログラムコード

```

$SN_AXCT_SWWIDTH[0] = 1
; チャンネル別マシンデータ:チャンネル 1
CHANDATA(1)
$MC_AXCONF_MACHAX_USED[0]=1
$MC_AXCONF_MACHAX_USED[1]=5
$MC_AXCONF_MACHAX_USED[2]=4
$MC_AXCONF_MACHAX_USED[3]=0
$MC_AXCONF_MACHAX_USED[4]=0
$MC_AXCONF_MACHAX_USED[5]=0
$MC_AXCONF_CHANAX_NAME_TAB[0] = 「XR」
$MC_AXCONF_CHANAX_NAME_TAB[1] = 「YR」
$MC_AXCONF_CHANAX_NAME_TAB[2] = 「ZR」
; チャンネル別マシンデータ:チャンネル 1
CHANDATA(2)
$MC_REFP_NC_START_LOCK=0
$MC_AXCONF_MACHAX_USED[0]=2
$MC_AXCONF_MACHAX_USED[1]=6
$MC_AXCONF_MACHAX_USED[2]=3
$MC_AXCONF_MACHAX_USED[3]=0
$MC_AXCONF_MACHAX_USED[4]=0
$MC_AXCONF_MACHAX_USED[5]=0
$MC_AXCONF_CHANAX_NAME_TAB[0] = 「XX」
$MC_AXCONF_CHANAX_NAME_TAB[1] = 「YY」
$MC_AXCONF_CHANAX_NAME_TAB[2] = 「ZZ」
M30

```

## NC プログラム NC\_P2:NCU 2 のマシンデータのパラメータ設定

## プログラムコード

```

; ##### NCU-2 #####
; NC 別マシンデータ
$MN_NCU_LINKNO = 2
$MN_MM_NCU_LINK_MASK = 1
$MN_MM_LINK_NUM_OF_MODULES= 2
$MN_MM_SERVO_FIFO_SIZE = 3
$MN_AXCONF_LOGIC_MACHAX_TAB[0] = 「CT1_SL1」
$MN_AXCONF_LOGIC_MACHAX_TAB[1] = 「CT1_SL2」
$MN_AXCONF_LOGIC_MACHAX_TAB[2] = 「NC2_AX3」
$MN_AXCONF_LOGIC_MACHAX_TAB[3] = 「NC2_AX4」
$MN_AXCONF_LOGIC_MACHAX_TAB[4] = 「NC2_AX5」

```

### プログラムコード

```
$MN_AXCONF_LOGIC_MACHAX_TAB[5] = 「NC2_AX6」  
CHANDATA(1)  
; チャンネル別マシンデータ:チャンネル1  
$MC_AXCONF_MACHAX_USED[0]=1  
$MC_AXCONF_MACHAX_USED[1]=2  
$MC_AXCONF_MACHAX_USED[2]=3  
$MC_AXCONF_MACHAX_USED[3]=4  
$MC_AXCONF_MACHAX_USED[4]=5  
$MC_AXCONF_MACHAX_USED[5]=6  
$MC_AXCONF_MACHAX_USED[6]=0  
M30
```

## 5.5.5 方向に依存したピッチ誤差補正

### 5.5.5.1 機能説明

補正点での方向に依存した差が非常に大きい場合、バックラッシの不整合や異常に高い精度要求のために、ピッチ誤差や検出器誤差(直接位置検出の場合)に対して方向に依存した補正が必要になることがあります

#### 方向に依存したピッチ誤差補正

「方向に依存したピッチ誤差補正」(「方向に依存した LEC」または「双方向 LEC」)では、各軸について 2 つのテーブルが使用されます。正の移動方向に対する補正テーブル 1 つと負の移動方向に対する補正テーブル 1 つ。指定の補正点の誤差は、理想的な指令値と計測された現在値の相違として補正テーブルに入力します。このコントローラは、直線補間を使用して、中間値の補正值を自動的に計算します。

#### 必要条件/適用

「方向に依存した LEC」は、SINUMERIK コントローラに「真直度補正」の特別なケースとして実行されます。このため、「真直度補正」の必要条件と条件が適用されます(「真直度および直角度補正 (ページ 338)」を参照してください)。

補正の適用は基準計測を使用して確認できます(たとえばレーザー干渉計や、最も単純なケースでは特定の軸のサービス画面を使用して)。

**通知**

「方向に依存した LEC」を真直度補正と直角度補正を並行して使用する場合、これらの機能の二次条件も合わせて考慮してください(たとえばテーブル<t>の特定の機能への割り当てなど)。

**5.5.5.2 セットアップ****誤差または補正值の計測**

「方向に依存した LEC」をセットアップする場合、「方向に依存した LEC」をセットアップする場合とまったく同様に、適切な計測機器(例: レーザー干渉計など)を使用して、各軸の方向に依存した誤差曲線が特定されます(「ピッチ誤差およびピッチ誤差補正 (ページ 333)」の章を参照してください)。この計測をおこなうには、計測点と待機時間を含むパートプログラムを生成してください(「例 (ページ 371)」の章:プログラム「BI\_SSFK\_MESS\_AX1\_X.MPF」を参照してください)。

SINUMERIK コントローラと組み合わせて、さまざまな計測機器が実際の実装について異なるサポートオプションを提供するので、次ではこの処理についてコントローラを基準として一般的な説明のみをおこないます。

**注記**

ピッチ誤差を特定するための計測は、マシンデータで機械座標系に対する軸の移動方向が正しく設定されているときに、初回セットアップでのみ実行してください。

## 試運転(基本)

1. 補正補間点の数を指定します(「ドループ補正と直角度誤差補正:セットアップ (ページ 348)」の章も参照してください)。  
方向ピッチ誤差補正では、正および負の移動方向の補正テーブルを各軸に割り当ててください。テーブルの補正補間点の数は、以下で定義します。

MD18342 \$MN\_MM\_CEC\_MAX\_POINTS[ <補正テーブルインデックス> ]

 注意
--

## データ消失の可能性

メモリを構成するマシンデータ MD18342 \$MN\_MM\_CEC\_MAX\_POINTS に変更を加えると、次回コントローラを起動すると、NC メモリが再構成 (ページ 973) されます。これにより、すべてのユーザー個別データが消失する可能性があります。

セットアップアーカイブの作成:

操作エリア:[「スタートアップ」|「ETC」キー|「セットアップアーカイブ」|「セットアップアーカイブの作成」|「OK」]選択:「NC データ」

## 例

- X 軸:正の移動方向、テーブル 1、11 個の補間点
  - X 軸:負の移動方向、テーブル 2、11 個の補間点
- マシンデータ:
- MD18342 \$MN\_MM\_CEC\_MAX\_POINTS[ 0 ] = 11
  - MD18342 \$MN\_MM\_CEC\_MAX\_POINTS[ 1 ] = 11
2. 作成したセットアップアーカイブの読み込み:  
操作エリア:[「スタートアップ」|「ETC」キー|「セットアップアーカイブ」|「セットアップアーカイブの読み込み」|「OK」]  
これで、補正テーブルが使用できるようになります。
  3. セットアップを簡略にするため、マシンデータおよびシステム変数に補正パラメータが書き込まれる NC プログラムを作成します(「例 (ページ 371)」の章を参照)。
  4. コントローラで NC プログラムを実行します。  
モード:[「自動」|プログラムを選択|NC スタート]
  5. 電源投入(ウォームリスタート)。
  6. これで、レーザー干渉計を使用して計測値比較を実行できるようになりました。
  7. 補正結果をさらに改善するために、プログラムで個々の補正値を修正することも可能です。これでテーブルを再読み込みするときに、電源投入する必要はなくなります。

---

**注記****SINUMERIK 828D での手順**

SINUMERIK 828D では、手順 **2** と **3** は除いてください。これは、「真直度補正、多次元」オプションが有効な場合は、それぞれ 200 個の補間点を持つ 12 個の補正用テーブルがすぐに使用可能になるためです。これは拡張できません!

**NC\_CEC.INI**

[セットアップ|システムデータ](「NC active data\sag angularity comp」フォルダから)によりコピーされるファイル「NC\_CEC.INI」には、取り決められたすべての真直度/直角度と方向に依存した LEC テーブルが含まれます。

**バックラッシュ**

バックラッシュを **0** に設定します。

- MD32450 \$MA\_BACKLASH [ <検出器> ] = 0
- 

**補正パラメータ**

補正パラメータは、以下のシステム変数で設定します。

- \$AN\_CEC[ <テーブル>, <補間点> ] (補正值)
  - \$AN\_CEC\_INPUT\_AXIS[ <テーブル> ] (基本軸)
  - \$AN\_CEC\_OUTPUT\_AXIS[ <テーブル> ] (補正軸)
- 

**注記**

「方向依存の LEC」の場合、基本軸および補正軸は、常に同じです。

---

- \$AN\_CEC\_STEP[ <テーブル> ] (補間点の距離)
  - \$AN\_CEC\_MIN[ <テーブル> ] (開始位置)
  - \$AN\_CEC\_MAX[ <テーブル> ] (終了位置)
  - \$AN\_CEC\_DIRECTION[ <テーブル> ] (方向)
- 

**注記**

セッティング \$AN\_CEC\_DIRECTION[ <t> ] = 0 (テーブルを基本軸の両方の移動方向に適用)は、「方向に依存した LEC」には該当しません。

---

- \$AN\_CEC\_IS\_MODULO[ <テーブル> ] (モジュロ機能による補正)

---

**注記**

これらのシステム変数の説明については、「真直度誤差と直角度誤差の補正:セットアップ (ページ 348)」の章を参照してください。

---

**単位系**

「ドループと直角度誤差に対する補正:セットアップ (ページ 348)」の章を参照してください。

**監視**

「ドループと直角度誤差に対する補正:セットアップ (ページ 348)」の章を参照してください。

**5.5.5.3 例**

次の例では、軸(機械軸 AX1)の方向補正テーブルのパラメータ設定を示します。補正テーブルのすべてのパラメータ値は、プログラムによって書き込まれます。

**補正パラメータ**

- 基本軸 = 補正軸 = 機械軸 AX1
- 補間点間の距離:58.0 mm
- 開始位置: -585.0 mm
- 終了位置: -5.0 mm

**テーブルの定義**

1 番目と 2 番目の補正テーブルは、方向補正テーブルとして機械軸 AX1 のそれぞれ 11 個の補正補間点で定義されます。

- MD18342 \$MN\_MM\_CEC\_MAX\_POINTS[ 0 ] = 11 (テーブル 1:正の移動方向)
- MD18342 \$MN\_MM\_CEC\_MAX\_POINTS[ 1 ] = 11 (テーブル 2:負の移動方向)

---

**注記**

SINUMERIK 828D では、補間点の数を定義する必要はありません。これは、「真直度補正、多次元」オプションが有効な場合は、それぞれ 200 個の補間点を持つ 12 個の補正用テーブルがすぐに使用可能になるためです。これは拡張できません!

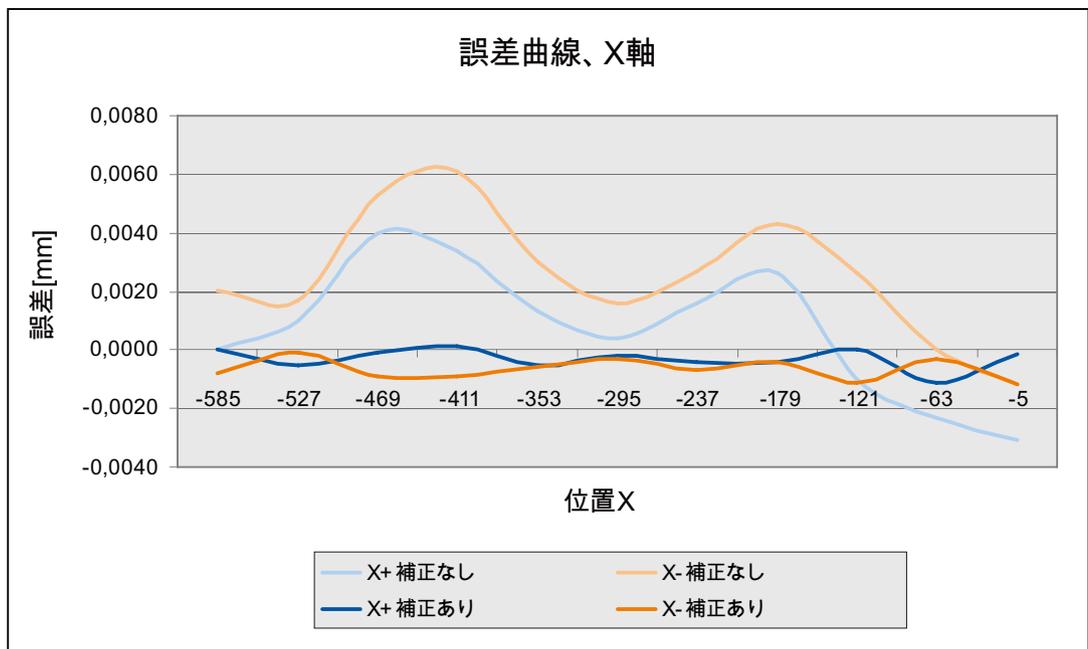
---

補間点および訂正值

補間点		誤差または訂正值		補正された誤差	
インデックス	位置 [mm]	正の移動方向 [mm]	負の移動方向 [mm]	正の移動方向 [mm]	負の移動方向 [mm]
0	-585 <sup>1)</sup>	0.0000	0.0020	0.0000	-0.0008
1	-527	0.0010	0.0017	-0.0005	-0.0001
2	-469	0.0040	0.0053	-0.0001	-0.0009
3	-411	0.0034	0.0061	0.0001	-0.0009
4	-353	0.0013	0.0030	-0.0005	-0.0006
5	-295	0.0004	0.0016	-0.0002	-0.0003
6	-237	0.0016	0.0027	-0.0004	-0.0007
7	-179	0.0026	0.0043	-0.0004	-0.0004
8	-121	-0.0010	0.0026	0.0000	-0.0011
9	-63	-0.0023	0.0000	-0.0011	-0.0003
10	-5 <sup>2)</sup>	-0.0031	-0.0012	-0.0001	-0.0012

1) 開始位置: \$AC\_CEC\_MIN[ <テーブル> ]

2) 終了位置: \$AC\_CEC\_MAX[ <テーブル> ]



## プログラミング

以下の操作は、プログラム「BI\_SSFK\_TAB\_AX1\_X.MPF」で実行されます。

- 補正の無効化
- 書き込む補正テーブルを無効にします(有効なテーブルには書き込みができません)。
- X 軸の正および負の移動方向の補正テーブルに補正値を書き込む
- 補正パラメータの書き込み

```

; 方向に依存した LEC
; 1 番目の軸 AX1
; テーブル 1 - 正の移動方向
; テーブル 2 - 負の移動方向
;--- Deaktivierung der Kompensation und der Tabellen
CHANDATA(1)
$MA_CEC_ENABLE[AX1]=0           ;補正オフ
$SN_CEC_TABLE_ENABLE[0]=0       ;テーブル 1 をロック
$SN_CEC_TABLE_ENABLE[1]=0       ; テーブル 2 をロック
NEWCONF
;--- 1. Kompensationstabelle, positive Verfahrriichtung
;----- Kompensationswerte
$AN_CEC[0,0]=0                   ; 訂正值(補間点 0)
$AN_CEC[0,1]=0.001               ; 訂正值(補間点 1)
$AN_CEC[0,2]=0.004               ; 訂正值(補間点 2)
$AN_CEC[0,3]=0.0034              ; 訂正值(補間点 3)
$AN_CEC[0,4]=0.0013              ; 訂正值(補間点 4)
$AN_CEC[0,5]=0.0004              ; 訂正值(補間点 5)
$AN_CEC[0,6]=0.0016              ; 訂正值(補間点 6)
$AN_CEC[0,7]=0.0026              ; 訂正值(補間点 7)
$AN_CEC[0,8]=-0.001              ; 訂正值(補間点 8)
$AN_CEC[0,9]=-0.0023             ; 訂正值(補間点 9)
$AN_CEC[0,10]=-0.0031            ; 訂正值(補間点 10)
; -----
Kompensationsparameter
$AN_CEC_INPUT_AXIS[0]=(AX1)      ;基本軸
$AN_CEC_OUTPUT_AXIS[0]=(AX1)    ;補正軸
$AN_CEC_STEP[0]=58.0             ; 補間点間の距離
$AN_CEC_MIN[0]=-585.0            ; 開始位置
$AN_CEC_MAX[0]=-5.0              ; 終了位置
$AN_CEC_DIRECTION[0]=1          ; テーブルを正の移動方向に適用
$AN_CEC_MULT_BY_TABLE[0]=0       ;乗算しない(ここでは該当しません)
$AN_CEC_IS_MODULO[0]=0           ;モジュロ機能を使用しない補正
;--- 2. Kompensationstabelle, negative Verfahrriichtung
;----- Kompensationswerte

```

## 5.5 補間補正

```

$AN_CEC[1,0]=0.002 ; 訂正值(補間点0)
$AN_CEC[1,1]=0.0017 ; 訂正值(補間点1)
$AN_CEC[1,2]=0.0053 ; 訂正值(補間点2)
$AN_CEC[1,3]=0.0061 ; 訂正值(補間点3)
$AN_CEC[1,4]=0.003 ; 訂正值(補間点4)
$AN_CEC[1,5]=0.0016 ; 訂正值(補間点5)
$AN_CEC[1,6]=0.0027 ; 訂正值(補間点6)
$AN_CEC[1,7]=0.0043 ; 訂正值(補間点7)
$AN_CEC[1,8]=0.0026 ; 訂正值(補間点8)
$AN_CEC[1,9]=0.000 ; 訂正值(補間点9)
$AN_CEC[1,10]=-0.0012 ; 訂正值(補間点10)
; -----
Kompensationsparameter
$AN_CEC_INPUT_AXIS[1]=(AX1) ; 基本軸
$AN_CEC_OUTPUT_AXIS[1]=(AX1) ; 補正軸
$AN_CEC_STEP[1]=58.0 ; 補間点間の距離
$AN_CEC_MIN[1]=-585.0 ; 開始位置
$AN_CEC_MAX[1]=-5.0 ; 終了位置
$AN_CEC_DIRECTION[1]==-1 ; テーブルを負の移動方向に適用
$AN_CEC_MULT_BY_TABLE[1]=0 ; 乗算しない(ここでは該当しません)
$AN_CEC_IS_MODULO[1]=0 ; モジュロ機能を使用しない補正(回転軸の場合のみ)
;--- Aktivierung der Kompensation und der Tabellen
$MA_CEC_ENABLE[AX1]=1 ; 補正オン
$$SN_CEC_TABLE_ENABLE[0]=1 ; テーブル1の有効化
$$SN_CEC_TABLE_ENABLE[1]=1 ; テーブル2の有効化
NEWCONF
M17 ; プログラム終了

```

## 5.5.6 シリンダエラー補正

## 5.5.6.1 オプション

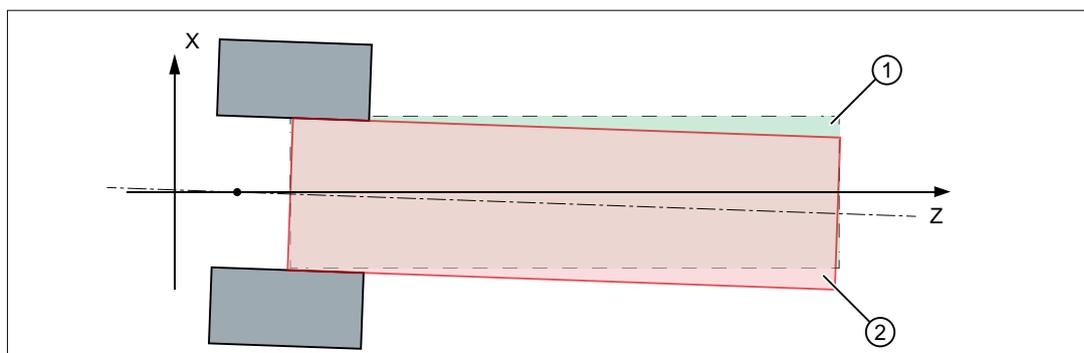
「シリンダエラー補正」機能を使用するには、次のオプションが必要です。

- SINUMERIK Grinding Advanced (6FC5800-0AS35-0YB0)

## 5.5.6.2 機能

シリンダエラー補正は、円筒研削のクランプ誤差を補正するために使用します。この場合、加工平面でのワークのシフトや回転が補正されます。

シリンダエラー補正の基礎として使用する真直度および直角度補正(ページ 338)とは対照的に、2つの補間点のみを使用して補正機能(直線)を定義します。



- ① シリンダのセット位置  
 ② シリンダの実際の位置

図 5-19 不正なクランプ固定によるシリンダエラー

### 5.5.6.3 セットアップ

番号	識別子	意味
SD41300	\$SN_CEC_TABLE_ENABLE	補正テーブルのイネーブル
SD41310	\$SN_CEC_TABLE_WEIGHT	加重係数
SD41320	\$SN_CEC_0	開始位置の補正值
SD41321	\$SN_CEC_1	終了位置の補正值
SD41330	\$SN_CEC_BAS_0	計測点 P1:基準値
SD41331	\$SN_CEC_BAS_1	計測点 P2:基準値
SD41335	\$SN_CEC_BAS_STORE_0	計測点 P1 のビットメモリ:基準値
SD41336	\$SN_CEC_BAS_STORE_1	計測点 P2 のビットメモリ:基準値
SD41340	\$SN_CEC_COMP_0	計測点 P1:シリンダエラー
SD41341	\$SN_CEC_COMP_1	計測点 P2:シリンダエラー
SD41350	\$SN_CEC_COMP_STORE_0	計測点 P1 のビットメモリ:シリンダエラー
SD41351	\$SN_CEC_COMP_STORE_1	計測点 P2 のビットメモリ:シリンダエラー

番号	識別子	意味
SD41355	\$SN_CEC_CALC	0/1 エッジで計算を開始
SD41356	\$SN_CEC_CALC_ADD	絶対または追加補正值

#### 注記

##### テーブルインデックス

以降で説明する同一インデックス<▶>を持つすべてのマシンデータ、セッティングデータ、およびシステム変数は、同じ補正テーブルに属します。

## 補正機能

シリンダエラー補正の補正機能は、以下のような直線の補正です。

$$\Delta X = m \cdot Z + b$$

$\Delta X$ : 補正軸のセット位置の補正值

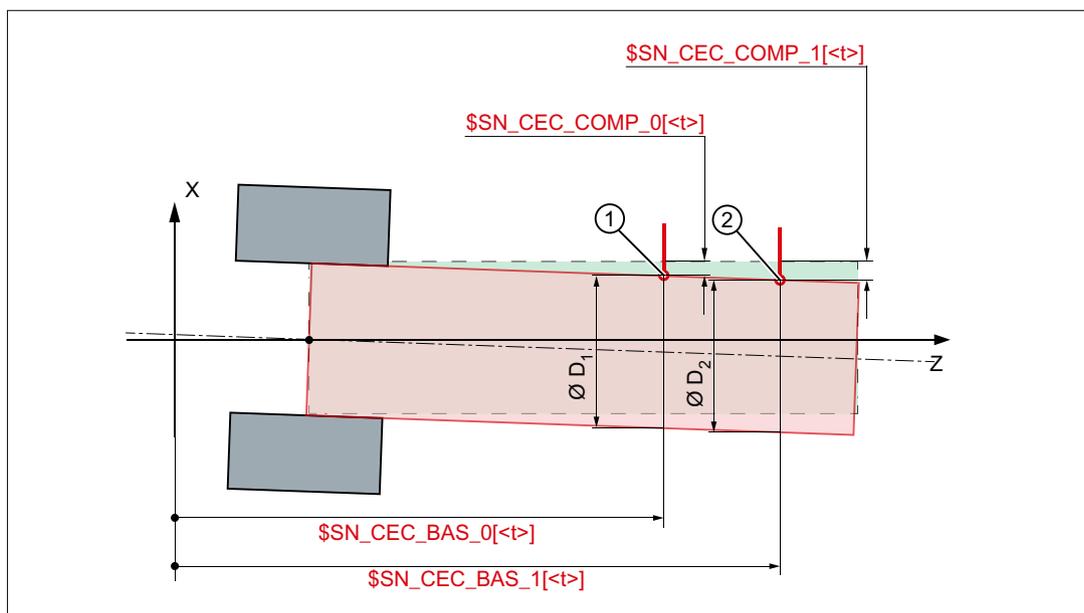
$m$ : 補正直線の傾斜

$z$ : 基本軸のセット位置

$b$ : 補正直線のオフセット

#### 計測点

コントローラが補正機能を計算できるようにするため、クランプした円筒形のワーク上の2点で補正軸の誤差を計測する必要があります。下図では、研削で標準の加工平面G18 (Z/X)に基づき、基本軸をZに指定し、補正軸をXとしています。



① 計測点 P1

② 計測点 P2

--- \$SN\_CEC\_BAS\_0/\_1 :基本軸(Z)の計測点の位置

--- \$SN\_CEC\_COMP\_0/\_1:補正軸(X)の計測点のシリンダエラー

---  $\varnothing D_1 == \varnothing D_2$

図 5-20 シリンダエラー補正の計測点

計測点は、以下の条件を満たす必要があります。

$$Z_{P1} < Z_{P2} \Rightarrow \$SN\_CEC\_BAS\_0 < \$SN\_CEC\_BAS\_1$$

## ベーシック試運転

シリンダエラー補正は、真直度補正(CEC) (ページ 338)に基づきます。その結果、シリンダエラー補正をセットアップする前に、まず真直度補正のシステム変数に以下の補正パラメータを設定してください。このために、NC プログラムのシステム変数に必要な値を書き込みます。

プログラムコード	コメント
$\$AN\_CEC\_INPUT\_AXIS[<t>] = \text{「<基本軸>」}$	; 基本軸の名称
$\$AN\_CEC\_OUTPUT\_AXIS[<t>] = \text{「<補正軸>」}$	; 補正軸の名称
$\$AN\_CEC\_MIN[<t>] = \text{<開始位置>}$	; 補正が有効な移動範囲の開始位置
$\$AN\_CEC\_MAX[<t>] = \text{<終了位置>}$	; 補正が有効な移動範囲の終了位置
$\$AN\_CEC\_STEP[<t>] = \text{<補間点間の距離>}$	; 補間点間の距離 = 終了位置 - 開始位置
$\$AN\_CEC\_DIRECTION[<t>] = 0$	; 方向に依存しない
$\$AN\_CEC\_IS\_MODULO[<t>] = FALSE$	; モジュロ機能なし

5.5 補間補正

プログラムコード	コメント
\$AN_CEC_MULT_BY_TABLE[<t>] = 0	; 他のテーブルを使った乗算なし
\$AN_CEC_TYPE = 1	; 補正タイプ = シリンダエラー補正

**補間点間の距離(\$AN\_CEC\_STEP[t])**

補正機能は直線であるため、シリンダエラー補正に必要なのは2つの補間点のみです。よって補間点間の距離は、終了点および開始点の差です。

$$\$AN\_CEC\_STEP[<t>] = \$AN\_CEC\_MAX[<t>] - \$AN\_CEC\_MIN[<t>]$$

**シリンダエラー補正の設定**

以下に、シリンダ補正の設定の基本的なアプローチを示します。

1. 実際の補正值を絶対または相対のいずれで実行するかの定義:

セッティングデータ		意味
SD4135 6	\$SN_CEC_CALC_ADD[<t>]	絶対または追加補正值(0:絶対、1:相対)

2. 補正值を加算する加重係数の定義:

セッティングデータ		意味
SD4131 0	\$SN_CEC_TABLE_WEIGHT[<t>]	加重係数

加重係数を使って特性曲線を垂直に移動することができます。

3. 2つの計測点で補正軸(X)方向の誤差を特定します。
4. 計測点(基本値および誤差または補正值)を補正データに転送します。

	基準値(Z)	補正值(X)
計測点 P1	SD41330 \$SN_CEC_BAS_0[<t>]	SD41340 \$SN_CEC_COMP_0[<t>]
計測点 P2	SD41331 \$SN_CEC_BAS_1[<t>]	SD41341 \$SN_CEC_COMP_1[<t>]

5. 補正機能の計算の開始時に信号立ち上がりエッジの変化(0→1)をトリガします。

セッティングデータ		意味
SD4135 5	\$SN_CEC_CALC[<t>]	0→1 エッジで計算を開始

6. コントローラが、補正直線の開始点および終了点の補正値をセッティングデータ SD41356 \$SN\_CEC\_CALC\_ADD[<t>]に応じて絶対または相対で計算します。

絶対:

- SD41320 \$SN\_CEC\_0[<t>] = <開始位置の計算された補正値>
- SD41321 \$SN\_CEC\_1[<t>] = <終了位置の計算された補正値>

相対:

- SD41320 \$SN\_CEC\_0[<t>] = SD41320 \$SN\_CEC\_0[<t>] + <開始位置の計算された補正値>
- SD41321 \$SN\_CEC\_1[<t>] = SD41321 \$SN\_CEC\_1[<t>] + <終了位置の計算された補正値>

セッティングデータ		意味
SD4132 0	\$SN_CEC_0[<t>]	開始位置の補正値
SD4132 1	\$SN_CEC_1[<t>]	終了位置の補正値

計測値は、演算後にビットメモリに書き込まれ、その後削除されます。

- ビットメモリ = 計測値
- 計測値 = 0.0

セッティングデータ		意味
ビットメモリ		
SD4133 5	\$SN_CEC_BAS_STORE_0[<t>]	計測点 P1 のビットメモリ:基本値
SD4135 0	\$SN_CEC_COMP_STORE_0[<t>]	計測点 P1 のビットメモリ:シリンダエラー
SD4133 6	\$SN_CEC_COMP_STORE_0[<t>]	計測点 P2 のビットメモリ:基本値
SD4135 1	\$SN_CEC_COMP_STORE_1[<t>]	計測点 P2 のビットメモリ:シリンダエラー
削除された計測値		
SD4133 0	\$SN_CEC_BAS_0[<t>]	計測点 P1:基本値
SD4134 0	\$SN_CEC_COMP_0[<t>]	計測点 P1:シリンダエラー

セッティングデータ		意味
SD4133 1	\$SN_CEC_BAS_1[<t>]	計測点 P2:基本値
SD4134 1	\$SN_CEC_COMP_1[<t>]	計測点 P2:シリンダエラー

7. 補正テーブルの使用のイネーブルを設定:

セッティングデータ		意味
SD4130 0	\$SN_CEC_TABLE_ENABLE[<t>]	補正テーブルのイネーブル

#### 注記

##### SINUMERIK ツールボックス

SINUMERIK ツールボックスでは、ファイル CNC\_TAB\_CONF.MPF も提供されます。このファイルには、NC プログラム内でシリンダエラー補正をパラメータ設定する方法の例が含まれています。

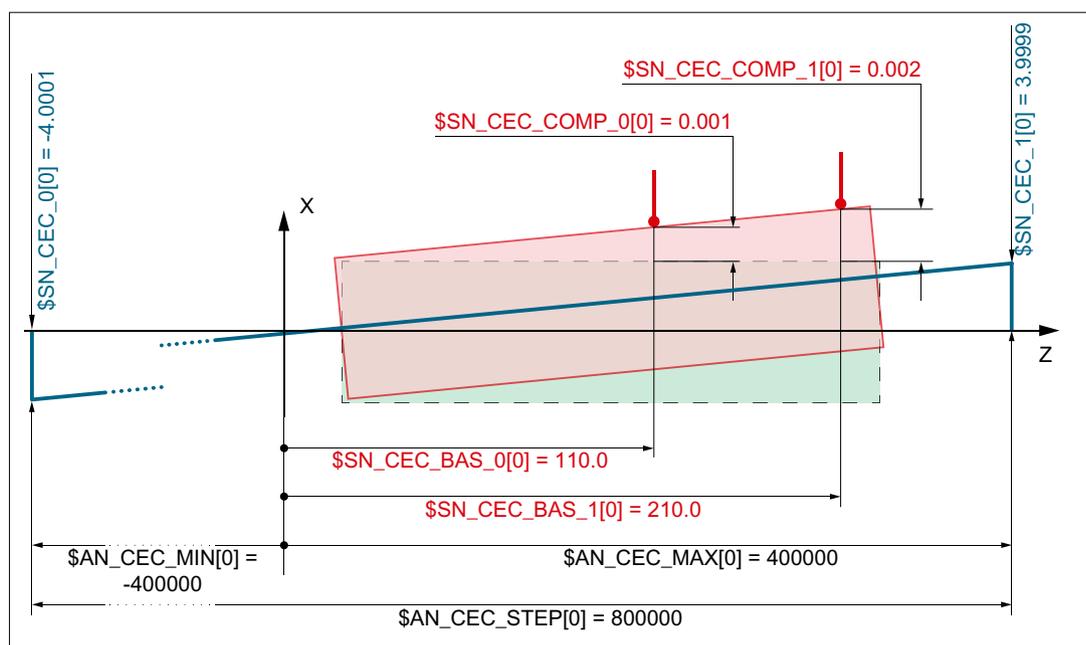
#### 5.5.6.4 例

以下のシリンダエラー補正の例では、最初の補正テーブル(インデックス 0)を使用します。

##### 例 1:絶対補正值を使用したシリンダエラー補正

###### 特性を決定する補正パラメータの概要

例では、下図に示す補正パラメータを使用します。



## 一般補正データ

セッティングデータ\$SN_...		値	意味
SD4130 0	CEC_TABLE_ENABLE [0]	FALSE	イネーブル
SD4131 0	CEC_TABLE_WEIGH T[0]	1	加重係数
SD4135 6	CEC_CALC_ADD[0]	FALSE	絶対補正值 <sup>1)</sup>

1) 以下で計算する補正值は、セッティングデータ SD41320 \$SN\_CEC\_0[0]および SD41321 \$SN\_CEC\_1[0]に書き込まれます。つまり、絶対値として適用されます。TRUE の場合、これらは既存の補正值に加算されます。

## 軸、補正範囲およびテーブル特性

システム変数\$AN_...	値	意味
CEC_INPUT_AXIS[0]	(Z)	基本軸
CEC_OUTPUT_AXIS[0]	(x)	補正軸
CEC_MIN[0]	-400,000 mm	開始位置
CEC_MAX[0]	400,000 mm	終了位置

システム変数\$AN_...	値	意味
CEC_STEP[0]	800,000 mm	補間点間の距離
CEC_DIRECTION[0]	0	補正テーブルは、基本軸の両方の移動方向に適用されます。
CEC_IS_MODULO[0]	FALSE	モジュロ機能無効
CEC_MULT_BY_TABLE[0]	0	テーブルの乗算が無効

## 計測点

セッティングデータ\$SN_...		値	意味
SD4133 0	CEC_BAS_0[0]	110.0 mm	計測点 P1:基本値
SD4133 1	CEC_BAS_1[0]	210.0 mm	計測点 P2:基本値
SD4134 0	CEC_COMP_0[0]	0.001 mm	計測点 P1:シリンダエラー
SD4134 1	CEC_COMP_1[0]	0.002 mm	計測点 P2:シリンダエラー

## 補正值の計算

補正值の計算は、セッティングデータ SD41355 \$SN\_CEC\_CALC[0]の信号立ち上がりエッジ(0→1)に基づいて、コントローラがおこないます。よって、以下に示す計算は、説明のためだけのものです。

- 補正機能  $\Delta X = f(Z)$ :

$$\Delta X = m * Z + b = \frac{X_{P2} - X_{P1}}{Z_{P2} - Z_{P1}} * Z + b \quad (1)$$

- 傾斜  $m$  の計算:

$$m = \frac{\text{SD41341 } \$SN\_CEC\_COMP\_1[0] - \text{SD41340 } \$SN\_CEC\_COMP\_0[0]}{\text{SD41331 } \$SN\_CEC\_BAS\_1[0] - \text{SD41330 } \$SN\_CEC\_BAS\_0[0]}$$

$$= \frac{0.002 - 0.001}{210.0 - 110.0} = 1 * 10^{-5} \quad (2)$$

- オフセット  $b$  の計算 → (1)で使用される計測点 P1(110.0, 0.001)および(2):

$$0.001 = 1 * 10^{-5} * 110.0 + b$$

$$b = -1 * 10^{-4} \quad (3)$$

- 得られる補正機能 → (1)で使用される(3)および(2):  

$$\Delta X = 1 \cdot 10^{-5} \cdot Z - 1 \cdot 10^{-4}$$
- 補正值の計算 → (4)で使用される開始または終了位置:  

$$\text{SD4132 } \$\text{SN\_CEC\_0}[0] = \Delta X_{\text{COMP\_0}} = 1 \cdot 10^{-5} \cdot -400000 - 1 \cdot 10^{-4} = -4.0001$$

$$\text{SD41321 } \$\text{SN\_CEC\_1}[0] = \Delta X_{\text{COMP\_1}} = 1 \cdot 10^{-5} \cdot 400000 - 1 \cdot 10^{-4} = 3.9999$$

セッティングデータ\$SN_...		値	意味
SD4132 0	CEC_0[0]	-4.0001 mm	開始位置の補正值
SD4132 1	CEC_1[0]	3.9999 mm	終了位置の補正值
SD4133 0	CEC_BAS_0[0]	0.0 mm	計測点 P1:基本値(削除)
SD4133 1	CEC_BAS_1[0]	0.0 mm	計測点 P2:基本値(削除)
SD4133 5	CEC_BAS_STORE_0[0]	110.0 mm	計測点 P1 のビットメモリ:基本値
SD4133 6	CEC_BAS_STORE_1[0]	210.0 mm	計測点 P2 のビットメモリ:基本値
SD4134 0	CEC_COMP_0[0]	0.0 mm	計測点 P1:シリンダエラー(削除済み)
SD4134 1	CEC_COMP_1[0]	0.0 mm	計測点 P2:シリンダエラー(削除済み)
SD4135 0	CEC_COMP_STORE_0[0]	0.001 mm	計測点 P1 のビットメモリ:シリンダエラー
SD4135 1	CEC_COMP_STORE_1[0]	0.002 mm	計測点 P2 のビットメモリ:シリンダエラー

## 例 2 :相対補正值を使用したシリンダエラー補正

例 2 は、例 1 の補正データに基づきます。この場合も、シリンダエラー補正は最初の補正テーブル(インデックス 0)を使っておこないます。

例 1 とは異なり、例 2 ではシリンダエラーは 2 つの計測点のみで特定します。補正值は、相対値で適用します。つまり、これらは例 1 の既存の補正值に**加算**されます。

## 一般補正データ

セッティングデータ\$SN_...		値	意味
SD4135 6	CEC_CALC_ADD[0]	TRUE	相対補正值 <sup>1)</sup>
1) 以下で計算した補正值は、セッティングデータ SD41320 \$SN_CEC_0[0]および SD41321 \$SN_CEC_1[0]に加算されます。			

## 計測点

セッティングデータ\$SN_...		値	意味
SD4133 0	CEC_BAS_0[0]	10.0 mm	計測点 P1:基本値
SD4133 1	CEC_BAS_1[0]	410.0 mm	計測点 P2:基本値
SD4134 0	CEC_COMP_0[0]	0.0 mm	計測点 P1:シリンダエラー
SD4134 1	CEC_COMP_1[0]	-0.003 mm	計測点 P2:シリンダエラー

## 補正值の計算

補正值の計算は、セッティングデータ SD41355 \$SN\_CEC\_CALC[0]の信号立ち上がりエッジ(0→1)に基づいて、コントローラがおこないます。

- 補正機能:  $\Delta X = m * Z + b = -7.5 * 10^{-6} * Z + 7.5 * 10^{-5}$
- 計算された補正值:
  - $\Delta X_{COMP\_0} = 3.000075 \text{ mm}$
  - $\Delta X_{COMP\_1} = -2.999925 \text{ mm}$
- 計算された補正值を実際の補正值に加算します。
  - $SD41320 \$SN\_CEC\_CEC\_0[0] = SD41320 \$SN\_CEC\_CEC\_0[0] + \Delta X_{COMP\_0} = -4.0001 \text{ mm} + 3.000075 = -1.000025 \text{ mm}$
  - $SD41321 \$SN\_CEC\_CEC\_1[0] = SD41321 \$SN\_CEC\_CEC\_1[0] + \Delta X_{COMP\_1} = 3.9999 \text{ mm} - 2.999925 \text{ mm} = 0.999975$

セッティングデータ\$SN_CEC_...		値	意味
SD4132 0	CEC_0[0]	-1.00002 5 mm	開始位置の補正值
SD4132 1	CEC_1[0]	0.99997 5 mm	終了位置の補正值
SD4133 0	CEC_BAS_0[0]	0.0 mm	計測点 P1:基本値(削除)
SD4133 1	CEC_BAS_1[0]	0.0 mm	計測点 P2:基本値(削除)
SD4133 5	CEC_BAS_STORE_0[0]	10.0 mm	計測点 P1 のビットメモリ:基本値
SD4133 6	CEC_BAS_STORE_1[0]	410.0 mm	計測点 P2 のビットメモリ:基本値
SD4134 0	CEC_COMP_0[0]	0.0 mm	計測点 P1:シリンダエラー(削除済み)
SD4134 1	CEC_COMP_1[0]	0.0 mm	計測点 P2:シリンダエラー(削除済み)
SD4135 0	CEC_COMP_STORE_0[0]	0.0 mm	計測点 P1 のビットメモリ:シリンダエラー
SD4135 1	CEC_COMP_STORE_1[0]	-0.003 mm	計測点 P2 のビットメモリ:シリンダエラー

### 5.5.7 必要条件

#### 補正された実際の位置

以下の機能は、補正された実際の位置に基づきます。

- 計測
- ティーチング
- ソフトウェアリミットスイッチ

### 実際の位置の表示

補正なしの軸(理想的な機械)の実際の位置は、実際の位置画面に機械座標系で表示されます。

補正した(MSEC およびバックラッシュ補正)軸の実際の位置は、「軸/主軸」サービス画面(「診断」操作エリア)に表示されます。

### 補正值の表示

次の補正值も、[軸/主軸]サービス画面([診断]操作エリア)に表示されます。

- 「絶対補正值検出器 1」または 2  
表示値は、MSEC の補正值で、基本軸および補正軸の実際の位置、およびバックラッシュ補正から取得されます。
- 補正、真直度 + 熱変位補正  
表示値は、基本軸および補正軸の実際の位置から取得された真直度および熱変位補正の補正值の合計です。

### 基本軸の原点確立済み状態の損失

基本軸の有効な検出器の原点確立済み状態が「原点確立済み/原点同期済み」から「原点未確立/原点非同期」(DB31、... DBX60.4 または5:1→0)に変わった場合、対応する軸(基本軸/補正軸)で MSEC や真直度補正は無効化されます。

その後、基本軸の有効な検出器の原点確立済み状態が「原点未確立/原点非同期」から「原点確立済み/原点同期済み」(DB31、... DBX60.4 または5:0→1)に変わった場合、対応する軸(基本軸/補正軸)で MSEC や真直度補正が再度有効になります。

### コントローライネーブル信号

補正関係が有効な場合、コントローライネーブル(DB31、... DBX2.1)が必ず基本軸/補正軸で同じに設定されます。

### 移動信号の出力

補正軸の移動信号は、補正信号が有効化/無効化されたとき、および有効な補正テーブルの数が変わった場合のみ出力されます。

基本軸の動作から生じる補正軸の移動動作によって、補正軸で移動信号が出力されることはありません。

## 5.6 動的なフィードフォワード制御 (追従誤差補正)

### 5.6.1 一般特性

#### 軸の追従誤差

機械軸を移動するときの位置コントローラに残るシステム誤差を、軸の追従誤差と呼びます。別の方法で表現すると、軸の追従誤差は、機械軸の指令位置と実位置の差です。

#### 作用

特に、輪郭コーナ(たとえば円弧、角など)の加速時に、この追従誤差により速度に依存した、不必要な輪郭誤差が生じます。

#### 補正

軸の追従誤差は、「ダイナミックフィードフォワード制御」により、ほとんどゼロにまで削減できます。したがって、この機能は「追従誤差補正」とも呼ばれています。

#### 方法

「ダイナミックフィードフォワード制御」には以下の2つの方法があります。

- 速度フィードフォワード制御(速度依存)
- トルクフィードフォワード制御(加速度依存)

#### 適用

フィードフォワード制御方法を選択し、マシンデータを使用して有効化します。

MD32620 \$MA\_FFW\_MODE (フィードフォワード制御モード)

規格値	意味
0	フィードフォワード制御はおこないません。
1	PT1 バランスによる速度フィードフォワード制御
2	PT1 バランスによるトルクフィードフォワード制御

## 5.6 動的なフィードフォワード制御 (追従誤差補正)

規格値	意味
3	Tt バランスによる速度フィードフォワード制御
4	Tt バランスによるトルクフィードフォワード制御

## パートプログラムでの起動/解除

次の軸別のマシンデータを使用して、対応する軸/主軸のフィードフォワード制御をパートプログラムで起動/解除ができるように定義することができます。

MD32630 \$MA\_FFW\_ACTIVATION\_MODE (プログラムからのフィードフォワード制御の起動)

規格値	意味
0	フィードフォワード制御をパートプログラムから起動と解除ができません。つまり、MD32620 \$MA_FFW_MODE を使用して指定された状態が軸/主軸に対して常に有効になります。
1	フィードフォワード制御はパートプログラムから起動と解除ができます。操作は直ちに有効になります。
2	フィードフォワード制御はパートプログラムから起動と解除ができます。操作は次に軸が停止したあとに有効になります。

フィードフォワード制御は次の操作を使用してパートプログラムから起動と解除をします。

FFWON: フィードフォワード制御「オン」

FFWOF: フィードフォワード制御「オフ」

初期設定(つまり M30 またはリセット後の場合)は、次のチャンネル別マシンデータを使用して入力します。

MD20150 \$MC\_GCODE\_RESET\_VALUES (G グループの初期設定)

FFWON/FFWOF は、軸モードですべての軸/主軸に対して有効です。

MD32630 \$MA\_FFW\_ACTIVATION\_MODE = 1 (または 2)

および

MD32620 \$MA\_FFW\_MODE = 1、2、3 または 4

MD32630 は、相互に補間する軸では同じ設定にしてください。

## 5.6 動的なフィードフォワード制御 (追従誤差補正)

急加速度を防止するため、フィードフォワード制御は軸モードで軸/主軸が停止している間にのみ有効または無効にしてください。したがって、ブロック先読み停止によって、次の停止状態まで切り替えが自動的に遅延します。

## 通知

先読み停止は、パートプログラム処理に非同期に移動するコマンド軸または PLC 軸には無効です。軸モードで次の停止時にのみ FFWON/FFWOF が軸/主軸に作用するためには、軸モードで各軸/主軸に対して MD32630 = 2 を明示的に設定してください(「命令軸および PLC 軸のフォワードフィード制御 (ページ 394)」も参照してください)。

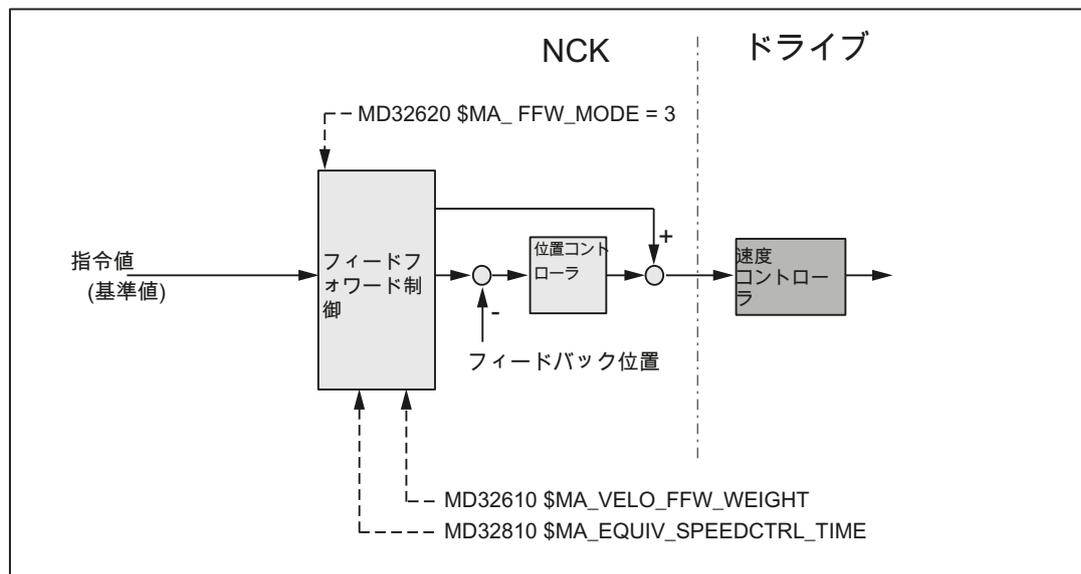
## 5.6.2 速度フィードフォワード制御

## 機能

速度フィードフォワード制御の場合、速度指令値も速度コントローラの入力に直接適用されます。この値によって、速度が一定の場合、追従偏差はほぼ 0(つまり、システム偏差が 0)に減少します。

## セットアップ

セットアップで、速度フィードフォワード制御に下記の軸別パラメータを定義してください。



## 5.6 動的なフィードフォワード制御 (追従誤差補正)

**速度制御ループの等価時定数(MD32810)**

速度制御ループの等価時定数は、速度フィードフォワード制御を正しく設定するため、(たとえば、速度指令値のステップ応答からグラフィックで)正確に特定して、次のマシンデータに入力してしてください。

**MD32810 \$MA\_EQUIV\_SPEEDCTRL\_TIME** (フィードフォワード制御の速度制御ループ等価時定数)

**速度フィードフォワード制御のフィードフォワード制御係数(MD32610)**

追加速度指令は係数を使用して重みを付加できます。

**MD32610 \$MA\_VELO\_FFW\_WEIGHT**

データ範囲: 0 ... 1

「0」: フィードフォワード制御は起こりません。標準では、係数の値は 1 です(△100%)。

係数の設定は 100%のままにしてください。この値は、軸/主軸に適切に設定された制御ループや、正確に特定された速度制御ループの等価時定数には最適な設定です。

**微調整**

特定の軸/主軸の速度フィードフォワード制御は、速度制御ループの等価時定数をわずかに変更(微調整)することで最適化できます(MD32810)。

この確認のために、軸/主軸は一定速度で移動して、サービス画面[軸/主軸]で、「システム偏差」を確認してください。

サービス画面で値を確認しやすいように、小さな加速度と大きな送り速度を選択してください。これにより、加速区間がかなり長くなり、制御偏差を読み取り易くなります。

例:

**MD32300 \$MA\_MAX\_AX\_ACCEL = 0.1** ; 最大軸加速度 = 0.1 m/s<sup>2</sup>

**MD32000 \$MA\_MAX\_AX\_VELO = 20000.0** ; 最大軸速度  
= 20000.0 mm/min

---

```
; 等価時定数を設定するためのパートプログラム
```

```
G1 F20000
FFWON
LOOP:
X1000
X0
GOTOB LOOP
```

M30

## 参照先

速度制御ループ(MD32810)の等価時定数の設定について詳細は、以下を参照してください。

- 機能マニュアル『基本機能』、速度、指令値/フィードバック回路、閉制御ループ(G2)、: 制御の調整の章

## 5.6.3 トルクフィードフォワード制御

### 機能

トルクフィードフォワード制御の場合は、トルクに比例する追加電流指令値が電流制御器の入力に直接適用されます。この値は、加速度と慣性モーメントから成ります。

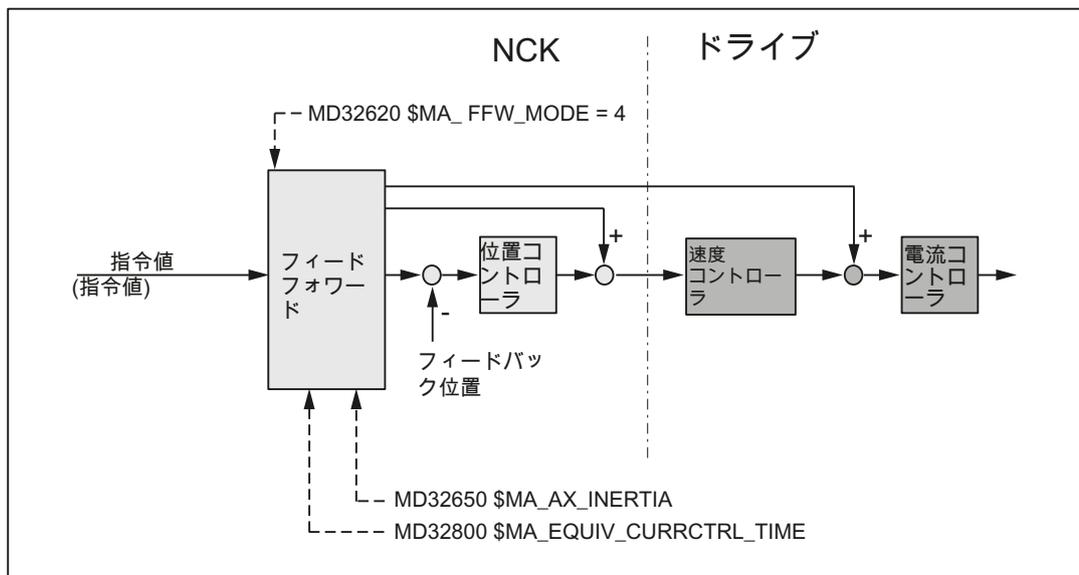
### 用途

トルクフィードフォワード制御は、高性能のダイナミック応答に対する要求がある場合に、高輪郭精度を実現するために必要です。この制御を正しく設定していれば、加速度が大きい場合でもほとんど完全に追従誤差を補正できます。

## 5.6 動的なフィードフォワード制御 (追従誤差補正)

## セットアップ

セットアップで、トルクフィードフォワード制御に関する下記の軸別パラメータを定義してください。

**電流制御ループの等価時定数(MD32800)**

電流制御ループの等価時定数は、トルクフィードフォワード制御を正しく設定するため、(たとえば、電流制御ループのステップ応答からグラフィックで)正確に特定して次のマシンデータに入力してしてください。

**MD32800 \$MA\_EQUIV\_CURRCTRL\_TIME** (フィードフォワード制御の電流制御ループ等価時定数)

**軸の合計慣性モーメント(MD32650)**

軸の合計慣性モーメント(ドライブの慣性モーメント + モータシャフト基準の負荷)を決定し、次のマシンデータに入力してください。

**MD32650 \$MA\_AX\_INERTIA** (トルクフィードフォワード制御の慣性)

**微調整**

特定の軸/主軸のトルクフィードフォワード制御は、MD32800 と MD32650 の値をわずかに変更(微調整)することで最適化できます

---

## 5.6 動的なフィードフォワード制御 (追従誤差補正)

確認のために、追従誤差をトレース機能に記録してください。一定速度での移動に加えて、特に軸/主軸が加速するときの追従誤差を監視してください。

---

### 注記

加速時の処理が非常に高速なため、トルクフィードフォワード制御をセットアップするときに、サービス画面を使用して微調整を確認できません。

---

## 参照先

電流制御ループ(MD32810)の等価時定数の設定について詳細は、以下を参照してください。

- 機能マニュアル『基本機能』、速度、指令値/フィードバック回路、閉制御ループ(G2)、: 制御の調整の章

## 5.6.4 ダイナミック応答調整

### 機能

相互に補間する軸であっても、それぞれの軸の制御ループ応答時間を持つ場合、制御品質を損なうことなく最適な輪郭精度を保証するために、すべての軸で同一時間で応答を実現するダイナミック応答調整を使用できます。

### セットアップ

#### ダイナミック応答調整の時定数(MD32910)

「最低速」または電流制御ループの等価時定数と特定の軸の等価時定数との差を、後述のマシンデータでダイナミック応答調整の時定数として入力してください。

MD32910 \$MA\_DYN\_MATCH\_TIME (ダイナミック応答調整の時定数)

例:

軸 1、2 および 3 の有効な速度フィードフォワード制御に対する速度制御ループの等価時定数(MD32810)

- 軸 1:2 ms
- 軸 2:4 ms (ダイナミックには最低速の軸)
- 軸 3:1 ms

## 5.6 動的なフィードフォワード制御 (追従誤差補正)

つまり、ダイナミック応答調整の時定数 MD32910 として、後述の値が得られます。

- 軸 1:2 ms
- 軸 2:0 ms
- 軸 3:3 ms

### 適用(MD32900)

ダイナミック応答調整は、後述のマシンデータが設定されている場合のみ有効になります。

MD32900 \$MA\_DYN\_MATCH\_ENABLE= 1

## 参照先

機能マニュアル 基本機能、速度、位置制御ループ(G2)「閉ループ制御の最適化」の章

## 5.6.5 命令軸および PLC 軸のフィードフォワード制御

### 機能

命令と PLC 軸に対して、後述のようにして高速時にフィードフォワード制御が有効/無効になるのを回避してください。

MD32630 \$MA\_FFW\_ACTIVATION\_MODE = 2

この設定により、FFWON/FFWOF 操作は、この特定の軸に設定された停止速度 (MD36060 \$MA\_STANDSTILL\_VELO\_TOL)以下でのみ有効になります。

切り替え命令が軸移動と同時に起こわれた場合、次回に軸が停止状態になってから必要な切り替えが実行されます。これにより、追従誤差が突然発生したり減少したりするのを避けられます。

---

### 注記

停止速度を非常に大きい値に設定すると、移動時にフィードフォワード制御の切り替えがおこなわれる可能性があります。制御は、既存の追従誤差に応じて、有効にできます。

---

## セットアップ

命令と PLC 軸に対するフィードフォワード制御を確認するには、後述の手順を推奨します。

1. MD36060 で停止速度を確認します。
2. 停止状態の軸について既存の追従誤差を確認します。
3. 切替条件の設定とその有効化  
MD32630 = 2
4. POSA 命令を使用してパートプログラムで軸を移動します。
5. 軸移動中に FFWON を実行します。
6. サービス画面[軸/主軸]に表示される  $K_v$  係数と追従誤差は、急変化しないようにしてください。
7. より大きい  $K_v$  係数でのより小さい追従誤差は、停止状態のあとの移動動作でのみ得られます。ただし、フィードフォワード制御は停止状態からのみ有効になります。

原則的には、フィードフォワード制御を無効にする場合には、有効にする場合と同様に後述の手順が適用されます。

1. POSA 命令を使用してパートプログラムで軸を移動します。
2. 軸移動中に FFWOF を実行します。
3. サービス画面[軸/主軸]に表示される  $K_v$  係数と追従誤差は、急変化しないようにしてください。
4. より小さい  $K_v$  係数でのより大きい追従誤差は、停止状態のあとの移動動作でのみ得られます。ただし、フィードフォワード制御は停止条件からのみ無効になります。

## 例

次のプログラム例では、軸 A は軌跡に対して非同期に移動します。移動中にチャンネルのフィードフォワード制御を有効にすることが試みられます。ジオメトリ軸 X、Y、および Z の場合とは異なり、A 軸に対してフィードフォワード制御は直ちに有効にはなりません。ここで、N60 の後の停止を待機します。次に A 軸は、N70 でフィードフォワード制御で移動します。

プログラムコード
N10 FFWOF
N20 POSA[A]=1000 FA[A]=10000
N30 G4 F1
N40 <b>FFWON</b>
N50 G0 X10 Y10 Z10
N60 WAITP (A)
N70 POSA[A]=1500 FA[A]=10000
N80 WAITP (A)
M30

## 5.6 動的なフィードフォワード制御 (追従誤差補正)

### 5.6.6 二次条件

#### 相互に補間する軸

また、相互に補間する軸には、**軸毎**に最適なフィードフォワード制御パラメータを設定してください。つまり、相互に補間する複数の軸にそれぞれ異なるフィードフォワード制御パラメータを設定できます。

#### 軌跡輪郭監視のチェック

2つの等価時定数として

- MD32810 \$MA\_EQUIV\_SPEEDCTRL\_TIME (フィードフォワード制御の速度制御ループ等価時定数)

および

- MD32800 \$MA\_EQUIV\_CURRCTRL\_TIME (フィードフォワード制御の電流制御ループ等価時定数)

これも軌跡誤差監視に影響を与えるため、続けてこれをチェックしてください。

#### 参照先:

『機能マニュアル、基本機能』、「軸監視と保護領域(A3)」

#### サーボゲイン係数( $K_V$ 係数)への影響

フィードフォワード制御が正しく設定されている場合、速度フィードフォワード制御の制御対象システムの指令値変更に対する応答は、速度制御ループの場合とダイナミックな特性は同じです。また、トルクフィードフォワード制御の制御対象システムの指令値変更に対する応答は、電流制御ループの場合とダイナミックな特性は同じです。つまり、MD32200 \$MA\_POS\_CTRLGAIN に入力されたサーボゲイン係数( $K_V$  係数)が制御動作(例: コーナ誤差、オーバーシュート、円弧/半径誤差)に影響を及ぼすことはほとんどありません。

また、フィードフォワード制御は外乱応答(同期制御)には影響を与えません。この場合、MD32200 に入力したサーボゲイン係数( $K_V$  係数)は、有効な要素です。

#### サービス画面「 $K_V$ 係数」

フィードフォワード制御が有効な場合、サービス画面[軸/主軸]に表示される軸のサーボゲイン係数( $K_V$  係数)(指令値変更に対する応答として有効なサーボゲイン係数( $K_V$  係数)に対応)は、非常に高くなります。

## 5.7 摩擦補正の概要

質量慣性と機械的な力の他に、機械のギヤと案内面の摩擦力が機械軸のダイナミックな特性に影響を及ぼします。特に、停止状態から機械軸を加速する際、静的摩擦から摺動摩擦への移行が悪影響を及ぼします。その結果として発生する摩擦力の突然の変化により、短期間、追従誤差が増大します。この結果として、補間軸により、大幅な経路干渉が生じます。円の場合、これらは特にいずれかの関与する軸が反転時に停止する象限移行時に発生します。

このような点で輪郭精度を改善するため、機械軸が停止状態から加速する際に、摩擦補正に対して補助設定値パルス(速度またはトルク指令値パルス)が適用されます。

### 摩擦補正機能

摩擦補正では、以下の機能を使用できます。

- 定補正值による摩擦補正 (ページ 398)  
機械軸の加速度に応じて、速度セットポイントに常に同じパルスが適用されます。速度セットポイントパルスの振幅と減衰時間が設定できます。
- 調整特性付きの摩擦補正 (ページ 404)  
機械軸の加速度に応じて、速度セットポイントパルスはパラメータ設定可能特性から決定されます。  
速度セットポイントパルスの最小/最大の振幅と減衰時間に加え、3つの異なる加速度値を設定できます。
- 調整特性付きの摩擦補正 (ページ 408) (ライセンスオプション:6FC5800-0AS06-0YB0)  
機械軸の加速度に応じて、速度セットポイントパルスは3つのパラメータ設定可能特性から決定されます。  
最大9つの異なる加速度値を設定できます。各加速度値では、速度セットポイントに振幅、有効時間および減衰時間の加重係数を定義できます。  
速度セットポイントパルスでは目的の結果を達成するのに不十分な場合、追加のトルク指令値パルスをパラメータ設定できます。

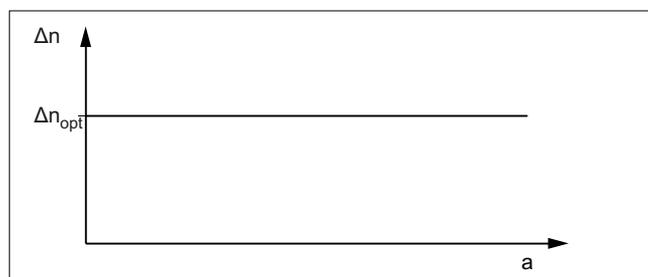
## 5.8 定補正值による摩擦補正

### 5.8.1 機能説明

以下で説明する定補正值による摩擦補正は、以下の一般的な条件が当てはまる用途を意図しています。

- 摩擦補正に必要な速度セットポイントパルスが、加速度に**依存しない**場合。
- 加速度に**依存しない**振幅および加速度に**依存しない**減衰時間が、速度セットポイントパルスのモデリングに十分な場合。
- 正および負の反転点での摩擦補正には、同一の速度セットポイントパルスで十分な場合。
- 反転点での輪郭精度の要求が比較的低い場合。

下図は、この機能で設定できる特性の例を示します。



$\Delta n_{op}$  速度セットポイントパルスの振幅

t

a 象限移行の加速度

### 5.8.2 セットアップ

軸別の補正值  $\Delta n_{opt}$  を決定するために、真円度テスト (ページ 400) を使って加速度値の各値の最適な速度セットポイントパルスの最適な振幅  $\Delta n_{opt,a}$  を求める必要があります。異なる加速度値がダイナミックな範囲全体をカバーするようにします。

### 象限移行での加速度の計算

円弧軌跡上で、象限移行時の反転時の機械軸の加速度は、円の半径  $r$  と軌跡速度  $v$  から計算されます。 $a = v^2 / r$

### 注記

軌跡速度、したがって軸別の加速度は、送り速度オーバーライドの切り替えによって簡単に変更できます。

### 決定された値ペア( $a$ 、 $\Delta n_{opt,a}$ )の評価

補正值  $\Delta n_{opt}$  を決定するため、加速度  $a$  と、それに関連する速度セットポイントパルスの最適な振幅  $\Delta n_{opt,a}$  で構成される真円度テストで特定された値のペアである  $\Delta n_{opt,a} = f(a)$  をグラフに適用することを推奨します。

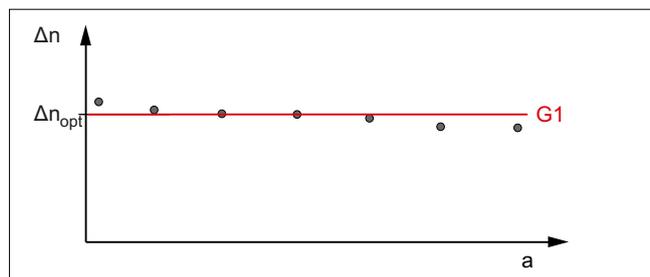


図 5-21 軸別の特性の決定

速度セットポイントパルスの最適な振幅  $\Delta n_{opt}$  は、直線 G1 を描画することで得られます。

## 軸マシンデータ

### 摩擦補正の有効化

摩擦補正は、以下を使って有効化します。

- MD32500 \$MA\_FRICT\_COMP\_ENABLE[ <軸> ] = TRUE (1)

### 定補正值の摩擦補正の有効化

定補正值の摩擦補正は、以下を使って有効化します。

- MD32490 \$MA\_FRICT\_COMP\_MODE[ <軸> ] = 1
- MD32510 \$MA\_FRICT\_COMP\_ADAPT\_ENABLE[ <軸> ] = FALSE (0)

### 速度セットポイントパルス:振幅

速度セットポイントパルスの振幅は、以下を使って設定します。

## 5.8 定補正值による摩擦補正

MD32520 \$MA\_FRICT\_COMP\_CONST\_MAX[ <軸> ] = <振幅>

速度セットポイントパルス:減衰時間

速度セットポイントパルスの減衰時間は、以下を使って設定します。

MD32540 \$MA\_FRICT\_COMP\_TIME[ <軸> ] = <時定数>

## 5.8.2.1 真円度テスト

## 真円度テストによるセットアップ

摩擦補正をセットアップする最も簡単な方法は、操作画面に組み込まれた真円度テストを使用することです。このために、関連する機械軸の実際の位置に基づいて円が移動する際に機械上で作成される円弧軌跡を取得します。特に象限移行で発生するプログラムされた理想の円弧軌跡からの誤差は、グラフで視覚化されます。

真円度テストは、下記の操作画面で利用できます。

- SINUMERIK Operate: 「操作エリア切替」 > [スタートアップ | 最適化 | 真円度テスト]

## 手順

機械軸の摩擦補正のセットアップ手順は、以下のステップに分けられます。

1. 軌跡速度  $v_n$  と円の半径  $r$  を使って加速度  $a_n$  を設定
2. 摩擦補正なしで真円度テストをおこないます
3. 振幅および減衰時間に小さな初期値に設定した摩擦補正を**使って**真円度テストを実行
4. パラメータ値を段階的に変更しながら摩擦補正を**使って**真円度テストを実行
5. 最適化されたパラメータ値をグラフに適用。例:振幅 =  $f(a_n)$
6. 次の加速度  $a_n$  を設定し、ポイント 2 で継続。

## 摩擦補正なしでの真円度テストの実行

象限移行での円弧軌跡の初期品質を特定するために、摩擦補正なしの真円度テストをおこなってください。このために、摩擦補正を一時的にオフにしてください。

MD32500 FRICT\_COMP\_ENABLE[ <軸> ] = 0

下記の図は、摩擦補正のない象限移行の代表的な例を示しています。

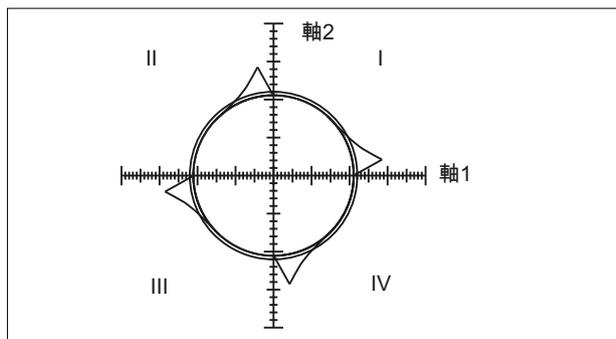


図 5-22 摩擦補正のない象限移行

次に、摩擦補正を再度有効化してください。

MD32500 FRICT\_COMP\_ENABLE[ <軸> ] = 1

### 摩擦補正ありの真円度テスト

速度セットポイントパルスには、以下の例のように少数の位置制御周期に小さな振幅値および減衰時間を初期値として設定することを推奨します。

- MD32520 \$MA\_FRICT\_COMP\_CONST\_MAX[ <軸> ] = 10 [mm/min]
- MD32540 \$FRICT\_COMP\_TIME[ <軸> ] = 0.008 [秒]

これらの値でおこなわれる真円度テストで、摩擦補正の初期評価が得られます。

#### 振幅が小さすぎる場合

振幅値(MD32520)が小さすぎる場合、真円度テストにおいて象限移行での輪郭誤差の補正が不十分になることにより明らかになります。

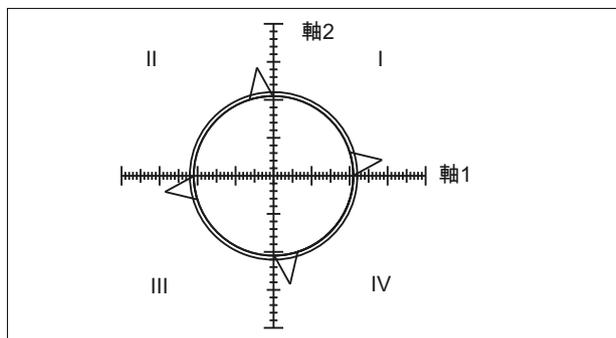


図 5-23 振幅が小さすぎる場合

#### 振幅が大きすぎる場合

振幅値(MD32520)が大きすぎる場合、真円度テストにおいて象限移行での輪郭誤差が過剰に補正されることにより明らかになります。

5.8 定補正值による摩擦補正

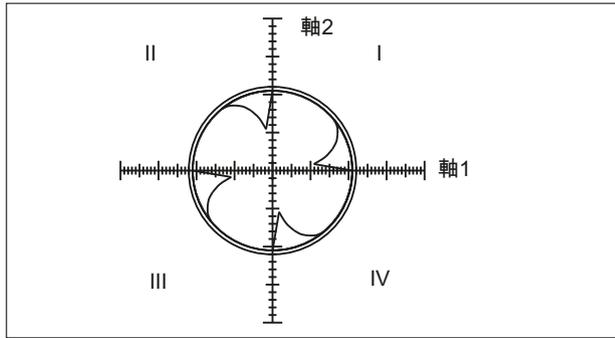


図 5-24 振幅が大きすぎる場合

**減衰時間が短すぎる場合**

減衰時間(MD32540)が短すぎる場合、真円度テストにおいて象限移行での輪郭誤差が短時間補正され、その直後に増大することにより明確になります。

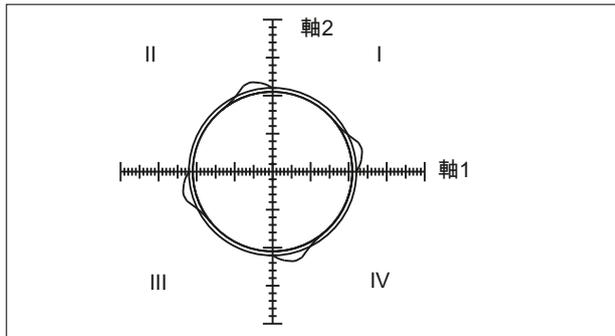


図 5-25 減衰時間が短すぎる場合

**減衰時間が長すぎる場合**

減衰時間(MD32540)が長すぎる場合、真円度テストにおいて象限移行での輪郭誤差が最初に補正されたときに明確になります。これは、最適な振幅値が設定済みであることを前提とします。ただし、減衰時間が長すぎる場合、補正の適用が長すぎ、その結果、次の円弧軌跡での過補正につながります。

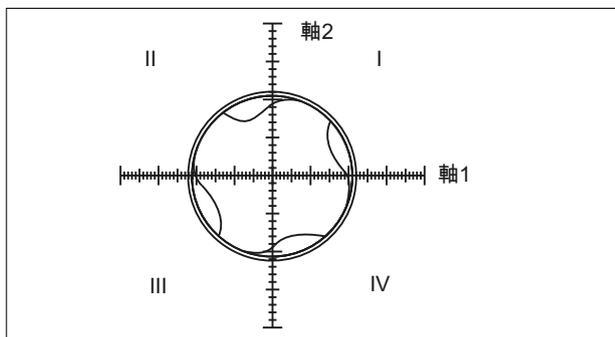


図 5-26 減衰時間が長すぎる場合

### 適切な摩擦補正設定

適切な摩擦補正を設定することにより、象限移行で経路干渉は検出されなくなります。

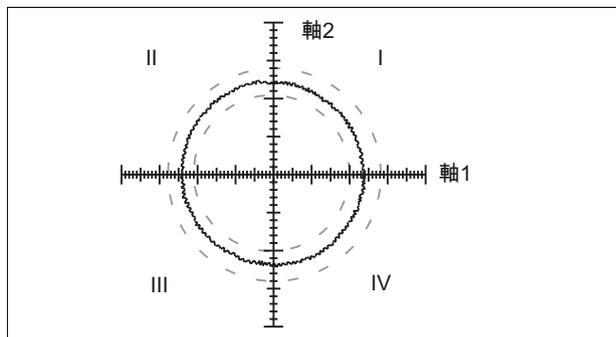


図 5-27 適切な摩擦補正設定

### 下記も参照

調整特性付きの摩擦補正 (ページ 404)

### 5.8.3 必要条件

#### セットポイントに関連した補正機能の応答

以下のセットポイント関連の補正機能は、位置指令に影響するため、真円度テストを実行する軸では無効化してください。

- 真直度と直角度の補正(CEC)  
MD32710 \$MA\_CEC\_ENABLE[ <軸> ] = 0
- 方向ピッチ誤差補正(LEC):  
MD32710 \$MA\_CEC\_ENABLE[ <軸> ] = 0
- 熱変位補正:  
MD32750 \$MA\_TEMP\_COMP\_TYPE[ <軸> ] = 0

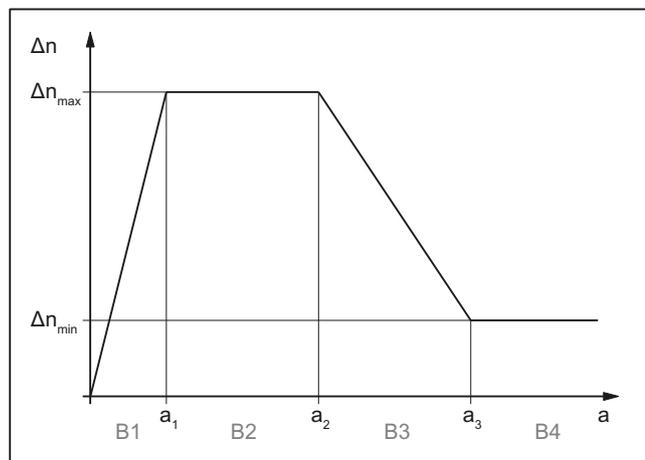
## 5.9 調整特性付きの摩擦補正

### 5.9.1 機能説明

調整特性付きの摩擦補正は、以下の要件を備えた用途に使用します。

- 摩擦補正に必要な速度セットポイントパルスが、加速度に依存する場合。
- 特性のモデリングには、3つの加速度補間点と速度セットポイントパルスの最小および最大振幅で十分な場合。
- 加速度に**依存する**振幅および加速度に**依存しない**減衰時間が、速度セットポイントパルスのモデリングに十分な場合。
- 正および負の反転点での摩擦補正には、同一の速度セットポイントパルスで十分な場合。
- より大きな加速度に、より小さな加速度に対するより小さな補正值が必要な場合。
- 反転点での輪郭精度の要求が高い場合。

下図は、この機能で設定できる特性の例を示します。



$\Delta n_{\max}$  速度セットポイントパルスの最大振幅

$\Delta n_{\min}$  速度セットポイントパルスの最大振幅

$a_1$ 、 加速度補間点 1、2、および 3

$a_2$ 、

$a_3$

B1 ... 加速度範囲 1 ... 4

B4

速度セットポイントパルス  $\Delta n$  の振幅は、B1 ~ B4 の対応する加速度範囲で、以下のよう  
に計算されます。

範囲	加速度 $a$	速度セットポイントパルス $\Delta n$ の振幅
B1	$a < a_1$	$\Delta n = \Delta n_{\max} * a / a_1$
B2	$a_1 \leq a \leq a_2$	$\Delta n = \Delta n_{\max}$
B3	$a_2 < a < a_3$	$\Delta n = \Delta n_{\max} + [(\Delta n_{\min} - \Delta n_{\max}) / (a_3 - a_2)] * (a - a_2)$
B4	$a \geq a_3$	$\Delta n = \Delta n_{\min}$

## 5.9.2 セットアップ

軸別の特性パラメータを決定するために、真円度テスト (ページ 400) を使って加速度値  
の各値の最適な速度セットポイントパルスの最適な振幅  $\Delta n_{\text{opt}_a}$  を求める必要がありま  
す。異なる加速度値がダイナミックな範囲全体をカバーするようにします。特に、低い  
加速度値と大きな円の半径では十分な数の計測をおこなう必要があります。

### 象限移行での加速度の計算

円弧軌跡上で、象限移行時の反転時の機械軸の加速度は、円の半径  $r$  と軌跡速度  $v$  から  
計算されます。  $a = v^2 / r$

### 注記

軌跡速度、したがって軸別の加速度は、送り速度オーバーライドの切り替えによって簡単  
に変更できます。

### 決定された値ペア ( $a$ , $\Delta n_{\text{opt}_a}$ ) の評価

加速度補間点  $a_1$ ,  $a_2$ , および  $a_3$  と、速度セットポイントパルス  $\Delta n_{\min}$  および  $\Delta n_{\max}$  の最  
小振幅を決定するため、加速度  $a$  と関連する速度セットポイントパルス  $\Delta n_{\text{opt}_a}$  の最適振  
幅で構成される真円度テストで決定した値のペアである  $\Delta n_{\text{opt}_a} = f(a)$  をグラフで適用する  
ことを推奨します。

## 5.9 調整特性付きの摩擦補正

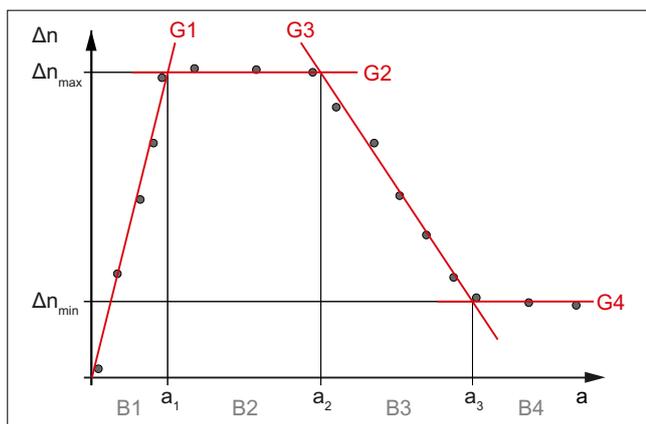


図 5-28 軸別の特性の決定

加速度補間点  $a_1$ 、 $a_2$ 、および  $a_3$  と、速度セットポイントパルス  $\Delta n_{\min}$  および  $\Delta n_{\max}$  の最小/最大振幅は、直線 G1 ...G4 を描画して得られます。

## 軸マシンデータ

## 摩擦補正の有効化

摩擦補正は、以下を使って有効化します。

MD32500 \$MA\_FRICT\_COMP\_ENABLE[ <軸> ] = TRUE (1)

## 調整特性付きの摩擦補正の有効化

調整特性付きの摩擦補正は、以下を使って有効化します。

- MD32490 \$MA\_FRICT\_COMP\_MODE[ <軸> ] = 1
- MD32510 \$MA\_FRICT\_COMP\_ADAPT\_ENABLE[ <軸> ] = TRUE (1)

特性パラメータ:加速度  $a_1$ 、 $a_2$ 、および  $a_3$ 

象限移行での反転時に円軌跡で発生する軸別の加速度は、半径  $r$  と軌跡速度  $v$  から、以下のように計算されます。 $a = v^2 / r$

加速度値  $a_1$ 、 $a_2$ 、および  $a_3$  は、マシンデータに一樣に昇順( $a_1 < a_2 < a_3$ )で入力されます。

- MD32550 \$MA\_FRICT\_COMP\_ACCEL1[ <軸> ] = < $a_1$ >
- MD32560 \$MA\_FRICT\_COMP\_ACCEL2[ <軸> ] = < $a_2$ >
- MD32570 \$MA\_FRICT\_COMP\_ACCEL3[ <軸> ] = < $a_3$ >

**特性パラメータ:振幅  $\Delta n_{\min}$  および  $\Delta n_{\max}$** 

速度セットポイントパルスの最大および最小の振幅( $\Delta n_{\max}$ 、 $\Delta n_{\min}$ )を以下のマシンデータに入力してください。

- MD32520 \$MA\_FRICT\_COMP\_CONST\_MAX[ <軸> ] = < $\Delta n_{\max}$ >
- MD32530 \$MA\_FRICT\_COMP\_CONST\_MIN[ <軸> ] = < $\Delta n_{\min}$ >

**注記**

非常に小さな軌跡速度に対して十分な結果が得られない場合は、計算分解能を高めることが必要になる場合があります。

- MD10200 \$MA\_INT\_INCR\_PER\_MM (直線位置の計算分解能)
- MD10210 \$MA\_INT\_INCR\_PER\_DEG (回転位置の計算分解能)

**速度セットポイントパルス:減衰時間**

速度セットポイントパルスの減衰時間は、以下を使って設定します。

MD32540 \$MA\_FRICT\_COMP\_TIME[ <軸> ] = <減衰時間>

**5.9.3 必要条件****セットポイントに関連した補正機能の応答**

以下のセットポイント関連の補正機能は、位置指令に影響するため、真円度テストを実行する軸では無効化してください。

- 真直度と直角度の補正(CEC)  
MD32710 \$MA\_CEC\_ENABLE[ <軸> ] = 0
- 方向ピッチ誤差補正(LEC):  
MD32710 \$MA\_CEC\_ENABLE[ <軸> ] = 0
- 熱変位補正:  
MD32750 \$MA\_TEMP\_COMP\_TYPE[ <軸> ] = 0

## 5.10 調整特性付きの摩擦補正

### 5.10.1 機能の説明

調整特性付きの摩擦補正には、以下の特性があります。

- 高速で単純な最適化のため、**SINUMERIK Operate** ユーザーインターフェースを使ってセットアップを実行できます。
- ガイド付きのセットアップでは、1つまたは2つの軸の試験プロセスがオプションで自動生成されます。
- 補正值は、垂直軸、各反転位置など、軸別に設定できます。
- セット位置に対して(補正モード (ページ 409) = 3)、補正值を早期に適用することにより、より良好な結果が得られます。
- ギヤレスドライブなどで、目的の結果を得るために速度セットポイントパルスが不十分な場合、パルスをトルク指令値に追加で適用できます。
- 最大 9 つの加速度補間点を使った 4 つの補正特性を定義できます。
- 速度セットポイントパルスに設定可能なパラメータは、以下のとおりです。
  - 遅延時間
  - 振幅 <sup>1)</sup>
  - アクション時間 <sup>1)</sup>
  - 立ち上がり時間
  - 減衰時間 <sup>1)</sup>
- トルク指令値パルスに設定可能なパラメータは、以下のとおりです。
  - 振幅 <sup>1)</sup>
  - 遅延時間
  - 立ち上がり時間
- 並列して有効な温度、ピッチ誤差や真直度補正は、摩擦補正には影響しません。

1):加速度に応じて調整できるパラメータ

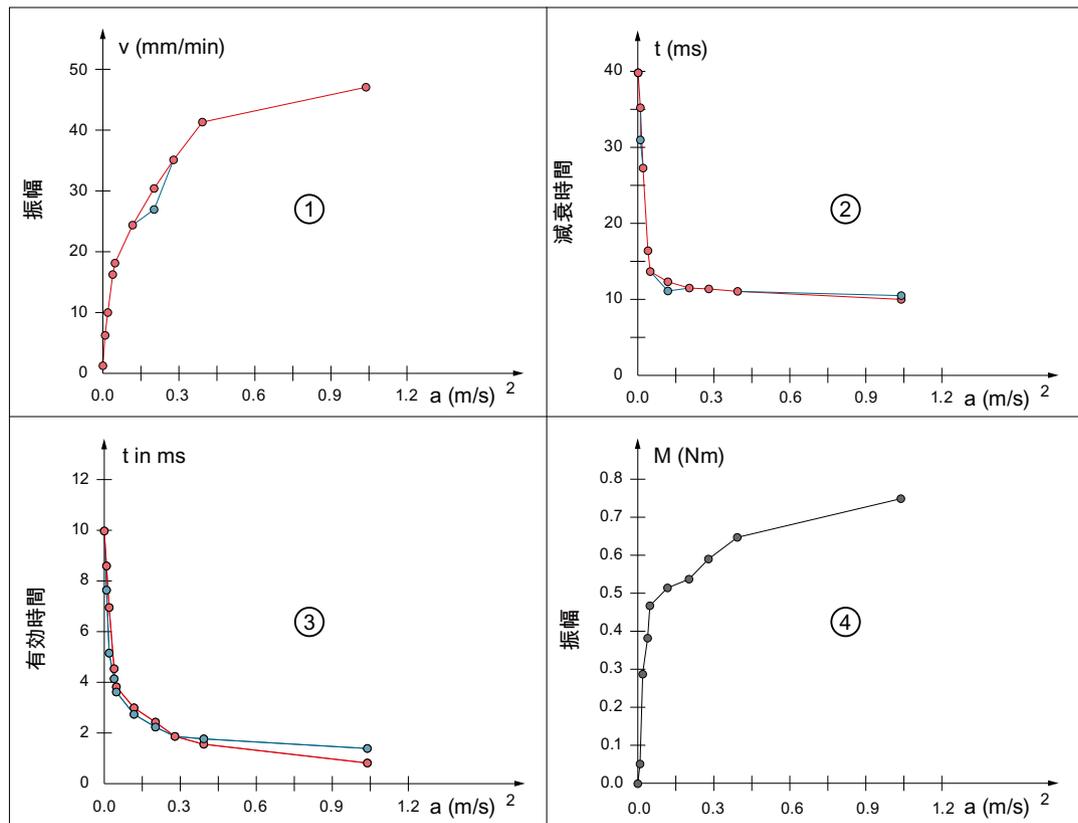
---

#### 注記

ほとんどの用途で、速度セットポイントパルスおよび振幅および減衰時間の補正特性による補正で十分です。満足できる結果が得られない場合のみ、その他のオプションを使用してください。

---

## 補正済みの特性の例



①②③ 速度セットポイントパルス:

- 赤色:下側の反転点
- 青色:上側の反転点

④ トルク指令値パルス

パラメータ設定された補間点間の加速度の補正值は、線形で補間されます。

## 5.10.2 セットアップ

### 5.10.2.1 機能の有効化

#### 摩擦補正の有効化

摩擦補正は、以下を使って軸別に有効化します。

```
MD32500 $MA_FRICT_COMP_ENABLE[<軸>] = TRUE
```

## 5.10 調整特性付きの摩擦補正

## 調整特性付きの摩擦補正の有効化

調整特性付きの摩擦補正は、軸のダイナミクスに応じて機械軸の設定位置または実際位置に適用できます。この機能は、以下を使って軸別に有効化します。

MD32490 \$MA\_FRICT\_COMP\_MODE = <モード>

<モード>	意味
3	ダイナミクスが大きく、フィードフォワードコントロールが有効の軸の場合 <sup>1)</sup> 。 機械軸の <b>セット</b> 位置がトリガ位置に到達すると、速度またはトルク指令値パルスが適用されます。
4	ダイナミクスが小さく、フィードフォワードコントロールが無効の軸の場合 <sup>1)</sup> 。 機械軸の <b>実際</b> の位置がトリガ位置に到達すると、速度またはトルク指令値パルスが適用されます。
1) フィードフォワード制御:MD32620 FFW_MODE (フィードフォワード制御モード)	

## 注記

## セットアップ

SINUMERIK Operate ユーザーインタフェース (ページ 410)のセットアップ機能を使ってガイド付きのセットアップを実行し、その後はマシンデータの生成された値を変更しないよう強く推奨します。

## 5.10.2.2 SINUMERIK Operate ユーザーインタフェースの機能のセットアップ機能

加速度調整付きの摩擦補正のセットアップは、「摩擦補正」ウィンドウで実行します。

操作エリア:[セットアップ|NC|摩擦補正]

## 注記

## ソフトキー[摩擦補正]

「摩擦補正」ソフトキーは、1つ以上の機械軸で調整特性付きの摩擦補正が有効 (ページ 409)な場合のみ表示されます。

軸別の速度およびトルク指令値パルスのパラメータが決定され、マシンデータに保持されるよう保存されます。通常は、速度セットポイントパルスを使った摩擦補正で十分です。

速度セットポイントパルスの特性パラメータの決定は、操作画面のガイドを使って実行します。

例外的な場合に必要となるトルク指令値パルスの特性パラメータは、手動で決定する必要があります。

#### 注記

##### 摩擦補正 - 軸選択ダイアログ

現在選択されている NCU 以外の NCU で軸を選択する場合、「摩擦補正」ダイアログを開くと切り替えはできなくなります。

切り替えのロックを消去するには、まず「摩擦補正」ダイアログを終了してください。

#### 参照先

SINUMERIK Operate ユーザーインタフェースを使った調整特性付きの摩擦補正の説明は、以下を参照してください。試運転マニュアル, CNC 試運転:NC、PLC、ドライブ; 「NC の試運転」章 > 「調整特性付きの摩擦補正」

## 機能

特性パラメータは、9つの加速度値 (ページ 413)まで真円度テストを使って決定します。コントローラによって自動的に生成された真円度テストプログラムで1つまたは2つの機械軸が交互に連続して移動され、結果は円の図に表示されます。

真円度プログラムの移動動作(半径、軌跡速度、および回転方向)は、以下のセッティングデータで設定済みです。必要に応じて、これらはデータリストを使って修正できます。以下の「その他のパラメータ」の説明を参照してください)。

パラメータ	軸タイプ	セッティングデータ
半径	直線軸	SD55820 \$SCS_FRICT_OPT_RADIUS
	回転軸	SD55821 \$SCS_FRICT_OPT_RADIUS_ROT
軌跡速度	直線軸	SD55822\$SCS_FRICT_OPT_FEED[ 0 ... 8 ]
	回転軸	SD55823 \$SCS_FRICT_OPT_FEED_ROT[ 0 ... 8 ]
回転方向 <sup>1)</sup>	---	SD55828 \$SCS_FRICT_OPT_DIR_MINUS
1) 2つの軸を使った真円度テストのみで有効		

#### 測定ステップ

真円度テストプログラムの開始後、一連の計測の9つまでの測定ステップのそれぞれで、軸はセッティングデータで定義された軌跡速度、または対応する加速度で移動します。実際の位置の理想的な軌跡からの誤差が取得され、セットとして表示されるか、実際の

## 5.10 調整特性付きの摩擦補正

円弧軌跡で表示されます。真円度テストは一定して連続的に実行しながら、速度セットポイントパルスのパラメータをセットアップエンジニアリングで変更できます。変更結果は、結果のグラフィックで自動的に更新されます。

真円度テストを1つの軸のみで実行した場合、結果は円としても表示されます。このために、軸の特定された輪郭誤差は、水平および垂直に分けて表示されます。真円度テストを2つの軸で実行する場合、両方の軸を同じチャンネルのジオメトリ軸としてください。

## 完了

一連の計測全体の補正パラメータの最適化が完了したら、コントローラによって最大値特性補間点および加重係数が計算され、以下のマシンデータに書き込まれます。

最大値	マシンデータ
振幅	MD32571 \$MA_FRICT_VELO_STEP
減衰時間	MD32574 \$MA_FRICT_V_PULSE_DECAY_TIME
有効時間	MD32573 \$MA_FRICT_V_PULSE_CONST_TIME

特性補間点	マシンデータ
摩擦補正特性の加速度	MD32581 \$MA_FRICT_ADAPT_TABLE_ACCEL[ 0 ... 9 ]

加重係数	マシンデータ
上側/下側の振幅	MD32582 \$MA_FRICT_ADAPT_V_STEP_PLUS[ 0 ... 9 ] MD32583 \$MA_FRICT_ADAPT_V_STEP_MINUS[ 0 ... 9 ]
上側/下側の減衰時間	MD32586 \$MA_FRICT_ADAPT_V_DECAY_PLUS[ 0 ... 9 ] MD32587 \$MA_FRICT_ADAPT_V_DECAY_MINUS[ 0 ... 9 ]
上側/下側のアクション時間	MD32584 \$MA_FRICT_ADAPT_V_CONST_PLUS[ 0 ... 9 ] MD32585 \$MA_FRICT_ADAPT_V_CONST_MINUS[ 0 ... 9 ]

### 追加パラメータ

垂直ソフトキー「データリスト」を使って、調整特性付きの摩擦補正のパラメータ設定をおこなうためのその他の機械およびセッティングデータが表示されます。例として、トルク指令値パルスのパラメータも表示されます。これらのパラメータの設定は、以下のセクションで説明します。

### 5.10.2.3 特性補間点での加速度のパラメータ化

真円度テストのチャンネルセッティングデータ(円の半径と移動速度)から、コントローラが特性補間点の加速度値を計算します。

### チャンネルセッティングデータ

円の半径および速度のチャンネルセッティングデータは、当該チャンネルのチャンネル軸となっている調整特性付きの摩擦補正が有効なすべての機械軸に適用されます。これらには、すでに標準値が割り当て済みです。必要に応じて、これらは機械別に調整できます。

#### 円弧半径

自動的に生成された真円度テストプログラムでは、円はパラメータ設定された半径で移動します。

- 直線軸  
SD55820 \$SCS\_FRICT\_OPT\_RADIUS = <半径>
- 回転軸  
SD55821 \$SCS\_FRICT\_OPT\_RADIUS\_ROT = <半径>

#### 移動速度

真円度テストでは、機械軸は最大 9 つの計測セクションを以下のセッティングデータでパラメータ設定された速度で移動します。

- 直線軸  
SD55822\$SCS\_FRICT\_OPT\_FEED[ 0 ... 8 ] = <速度 1 ... 9>
- 回転軸  
SD55823 \$SCS\_FRICT\_OPT\_FEED\_ROT[ 0 ... 8 ] = <速度 1 ... 9>

---

### 注記

- 9 つより少ない特性補間点( 1 ... n )が必要な場合は、不要な特性補間点( (n+1) ... 9 )に対応するそれぞれの配列エレメントに速度ゼロを入力します。
  - 配列エレメントで速度がゼロと等しい場合、それ以降の配列エレメントにゼロに等しい速度が含まれないようにしてください。
-

5.10 調整特性付きの摩擦補正

回転方向

2つの軸を使った真円度テストでは、以下で円の回転方向を定義します。

SD55828 \$SCS\_FRICT\_OPT\_DIR\_MINUS = <回転方向>

軸マシンデータ

特性補間点  $n$  での加速度  $a$  は、円半径  $r$  と軌跡速度  $v_{n-1}$  のセッティングデータから、コントローラによって次のように計算されます。 $a_n = v_{n-1}^2 / r$

$$\text{MD32581 } \$\text{MA\_FRICT\_ADAPT\_TABLE\_ACCEL}[n] = \frac{(\text{SD55822 } \$\text{SCS\_FRICT\_OPT\_FEED}[(n-1)])^2}{\text{SD55820 } \$\text{SCS\_FRICT\_OPT\_RADIUS}}$$

ここで、 $n = 1, 2, 3, \dots, 9$

特性補間点  $n = 0$  での加速度  $a$  の場合、常に値ゼロを入力します。

MD32581 \$MA\_FRICT\_ADAPT\_TABLE\_ACCEL[ 0 ] = 0

5.10.2.4 速度セットポイントパルス

速度セットポイントパルスの特性パラメータの特定と、対応するマシンデータの計算は、ユーザーインターフェース (ページ 410) で完全にサポートされています。よって、手動による特性パラメータの特定や、マシンデータの書き込みを推奨しません。

軸マシンデータ

速度セットポイントパルス

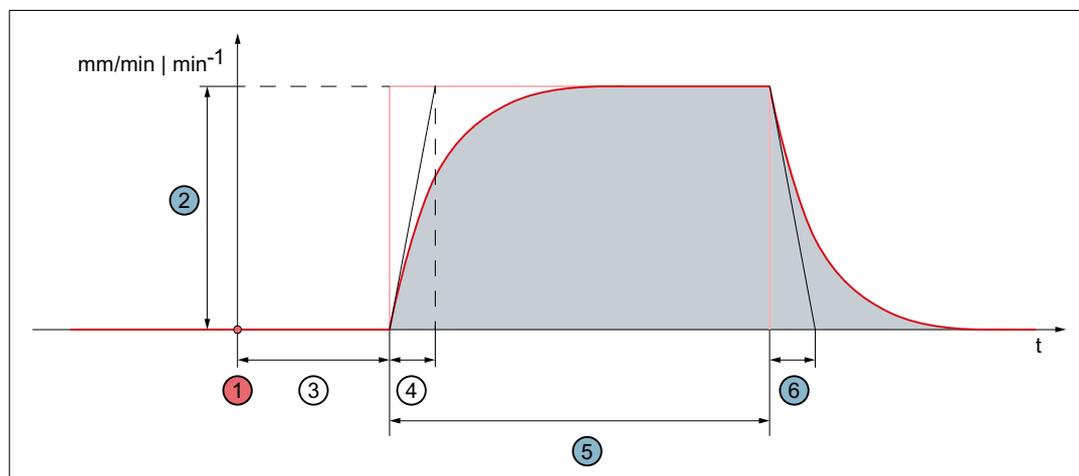


図 5-29 基本パルス形状

以下のテーブルの番号(①、②、...)は、上図の番号を示します。

#### 加速度に依存しないパラメータ

番号	マシンデータ	説明
①	-	停止状態からの軸のトリガ時間または加速度
③	MD32572 \$MA_FRICT_V_PULSE_DELAY_TIME	遅延時間
④	MD32575 \$MA_FRICT_V_PULSE_SMOOTH_TIME	立ち上がり時間の時定数 T、 5 * T の後 ⇒ 出力値 ≈ 入力値

#### 加速度依存の調整可能パラメータの最大値

番号	マシンデータ	説明
②	MD32571 \$MA_FRICT_VELO_STEP	振幅
⑤	MD32573 \$MA_FRICT_V_PULSE_CONST_TIME	有効時間
⑥	MD32574 \$MA_FRICT_V_PULSE_DECAY_TIME	減衰時間

## 5.10 調整特性付きの摩擦補正

## 最大値の加速度依存の調整の加重係数

表 5-1 下側の反転点

マシンデータ	説明
MD32582 \$MA_FRICT_ADAPT_V_STEP_PLUS[0 ... 9]	振幅の加重係数
MD32584 \$MA_FRICT_ADAPT_V_CONST_PLUS[0 ... 9]	アクション時間の加重係数
MD32586 \$MA_FRICT_ADAPT_V_DECAY_PLUS[0 ... 9]	減衰時間の加重係数

表 5-2 上側の反転点

マシンデータ	説明
MD32583 \$MA_FRICT_ADAPT_V_STEP_MINUS[0 .. 9]	振幅の加重係数
MD32585 \$MA_FRICT_ADAPT_V_CONST_MINUS[0 ... 9]	アクション時間の加重係数
MD32587 \$MA_FRICT_ADAPT_V_DECAY_MINUS[0 ... 9]	減衰時間の加重係数

## 特性パラメータの計算

セットアップ技術者は、真円度テスト (ページ 410)中に、異なる加速度値の速度セットポイントパルスの振幅、アクション時間、および減衰時間の最適な値を特定しなければなりません。真円度テストの完了時に、コントローラは得られる特性パラメータ(最大値、特性補間点、および加重係数)を計算し、それらに対応するマシンデータに書き込みます。

例:下側の反転点の振幅特性

- 最大値

$$\begin{aligned} & \text{MD32571} \\ & \$\text{MA\_FRICT\_VELO\_STEP} = \max\{\text{振幅1、振幅2 ... 振幅9}\} \end{aligned}$$

- 加重係数

$$\begin{aligned} & \text{MD32582} \\ & \$\text{MA\_FRICT\_ADAPT\_V\_STEP\_PLUS}[n] = \frac{\text{振幅}[n]}{\text{MD32571 } \$\text{MA\_FRICT\_VELO\_STEP}} \end{aligned}$$

- 有効な補間点

振幅特性の有効な補間点は、以下が適用される補間点です。

- MD32581  $\$MA\_FRICT\_ADAPT\_TABLE\_ACCEL[ <補間点> ] \neq 0$  (加速度)
- MD32582  $\$MA\_FRICT\_ADAPT\_V\_STEP\_PLUS[ <補間点> ] \neq 0$  (加重係数)

有効な補間点の速度セットポイントパルスの振幅は、以下のように計算されます。

$$\begin{aligned} \text{振幅}[n] &= \text{最大値} * \text{加重係数}[n] = \\ & \text{MD32571 } \$\text{MA\_FRICT\_VELO\_STEP} * \\ & \text{MD32582 } \$\text{MA\_FRICT\_ADAPT\_V\_STEP\_PLUS}[n] \end{aligned}$$

---

#### 注記

##### 特性の垂直オフセット

その後の特性の垂直変位は、マシンデータの対応する最大値を使って最も簡単におこなうことができます。

---

### 5.10.2.5 トルク指令値パルス

トルクセットポイントパルスの特性パラメータの特定と、対応するマシンデータの計算は、ユーザーインターフェースで直接はサポートされていません。よって、手動による特性パラメータの特定や、マシンデータの書き込みを推奨します。

軸マシンデータ

トルク指令値パルス

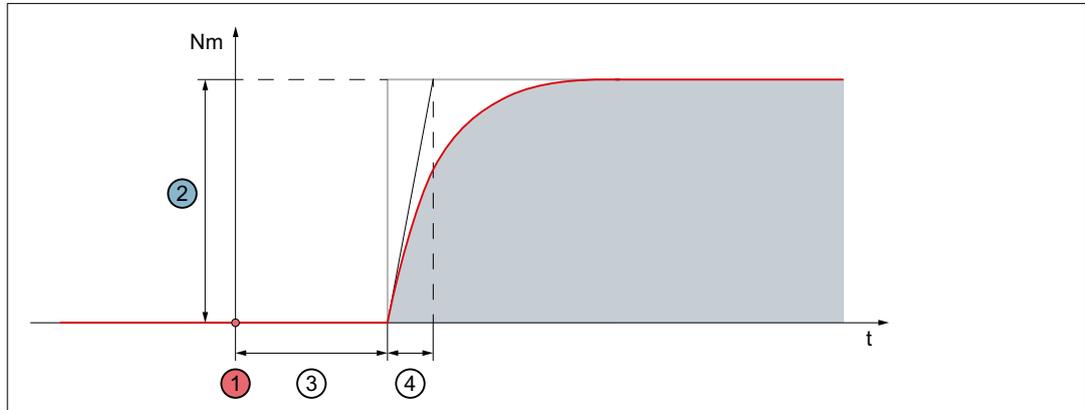


図 5-30 基本パルス形状

以下のテーブルの番号(①、②、...)は、上図の番号を示します。

加速度に依存しないパラメータ

番号。	マシンデータ	説明
①	-	停止状態からの軸のトリガ時間または加速度
③	MD32577 \$MA_FRICT_T_PULSE_DELAY_TIME	遅延時間
④	MD32578 \$MA_FRICT_T_PULSE_SMOOTH_TIME	立ち上がり時間

加速度依存の調整可能な振幅の最大値

No.	マシンデータ	説明
②	MD32576 \$MA_FRICT_TORQUE_STEP	振幅

## 最大値の加速度依存の調整の加重係数

マシンデータ	説明
MD32588 \$MA_FRICT_ADAPT_T_STEP[ 0 ... 9 ]	振幅の加重係数

## セットアップ(手動)

## 必要条件

特性パラメータを特定するための真円度テスト (ページ 410) が、問題なくすべて完了しているか、少なくとも現在の測定ステップについて完了していること。

## 注記

## 特性パラメータの特定

- トルク指令値パルスの特性パラメータは、ユーザーインターフェースの真円度テスト (ページ 410) で最も簡単に特定できます。
- 操作エリア:[セットアップ|NC|摩擦補正]
- マシンデータは、真円度テストのデータリストを使って最も簡単に設定できます。  
操作エリア:[セットアップ|NC|摩擦補正]、「データリスト」垂直ソフトキー

## 前提条件

真円度テストが開始され、測定ステップ 1 が実行中であること。

## 推奨手順

1. 加重係数の初期化:
  - MD32588 \$MA\_FRICT\_ADAPT\_T\_STEP[ 0 ] = 1
  - MD32588 \$MA\_FRICT\_ADAPT\_T\_STEP[ 1 ... 9 ] = 0
 注：すべての測定ステップで、インデックス 0 の加重係数を使用します。
2. 以下で、現在の測定ステップに振幅を設定します。
  - MD32576 \$MA\_FRICT\_TORQUE\_STEP = <振幅>
3. 変更された輪郭誤差を円の図で特定します。
4. MD32576 \$MA\_FRICT\_TORQUE\_STEP で変更し、円の図でチェックして振幅値を最適化します。
5. 後に測定ステップの最大値または加重係数を特定するために、この測定ステップの振幅値を記録します。
6. 次の測定ステップに切り替え、すべての測定ステップの振幅値が特定されるまでポイント 2 を継続します。

## 5.10 調整特性付きの摩擦補正

7. 記録した振幅値から最大値を特定し、それらをマシンデータに入力します。

$$\text{MD32576 } \$\text{MA\_FRICT\_TORQUE\_STEP} = \max\{\text{振幅1, 振幅2 ... 振幅9}\}$$

8. すべての有効な補間点の加重係数を記録した振幅値から計算し、それらをマシンデータに入力します。

$$\text{MD32588 } \$\text{MA\_FRICT\_ADAPT\_T\_STEP}[n] = \frac{\text{振幅}[n]}{\text{MD32576 } \$\text{MA\_FRICT\_TORQUE\_STEP}}$$

これで、トルク指令値パルスの振幅特性のセットアップが完了します。

## オプション

加速度と関係しないパラメータを設定します。

- MD32577 \$MA\_FRICT\_T\_PULSE\_DELAY\_TIME
- MD32578 \$MA\_FRICT\_T\_PULSE\_SMOOTH\_TIME

## 有効な補間点

振幅特性の有効な補間点は、以下が適用される補間点です。

- MD32581 \$MA\_FRICT\_ADAPT\_TABLE\_ACCEL[ <補間点> ] ≠ 0 (加速度)
- MD32583 \$MA\_FRICT\_ADAPT\_T\_STEP[ <補間点> ] ≠ 0 (加重係数)

有効な補間点のトルク指令値パルスの振幅は、以下のように計算されます。

$$\begin{aligned} \text{振幅}[n] &= \text{最大値} * \text{加重係数}[n] = \\ & \text{MD32576 } \$\text{MA\_FRICT\_TORQUE\_STEP} * \text{MD32588 } \$\text{MA\_FRICT\_ADAPT\_T\_STEP}[n] \end{aligned}$$

## 注記

## 特性の垂直オフセット

その後の振幅特性の垂直変位は、マシンデータの対応する最大値を使って最も簡単におこなうことができます。

## 5.11 吊り下げ軸の補正機能

### 5.11.1 電子カウンタウェイト

#### カウンタウェイトのない軸

カウンタウェイトのない重量負荷のある軸では、ブレーキの解放後、吊り下げ(懸架)軸が降下し、結果として後述の図のような動作が得られます。

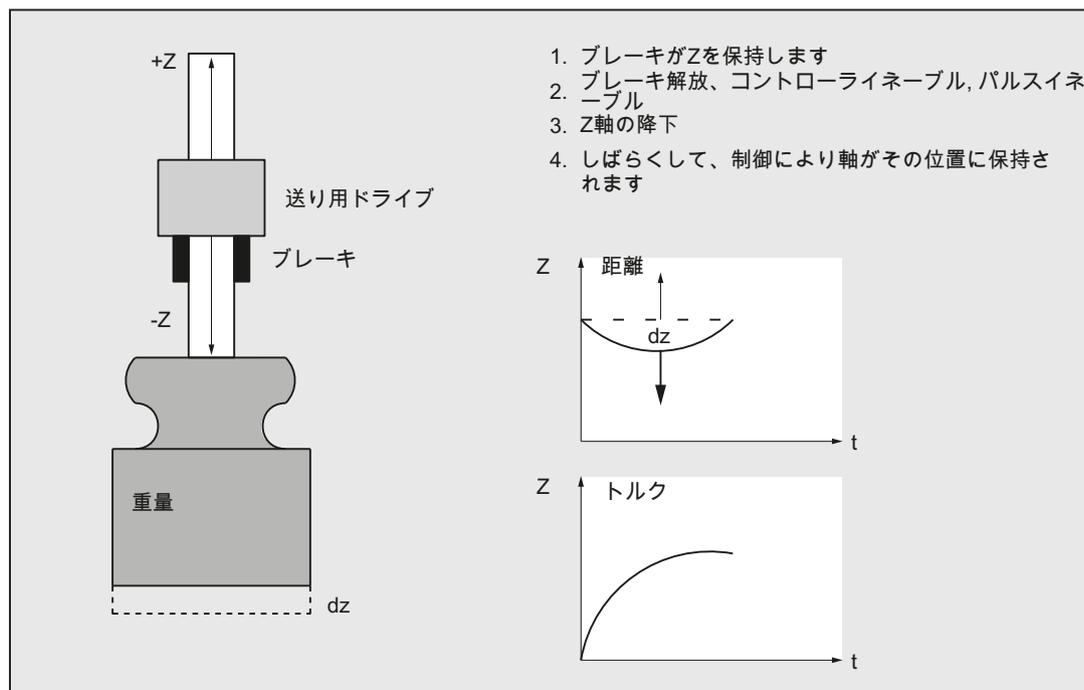


図 5-31 カウンタウェイトのない懸架軸の降下

#### 「電子カウンタウェイト」機能

「電子カウンタウェイト」機能を使用して、吊り下げ(懸架)軸の降下(落下)をほぼ完全に防ぐことができます。

電子カウンタウェイトは、閉ループ制御がスイッチオンされたときに、重量負荷のある軸の降下を防ぎます。ブレーキの解放後、一定のカウンタウェイトトルクにより垂直軸の位置が保持されます。

5.11 吊り下げ軸の補正機能

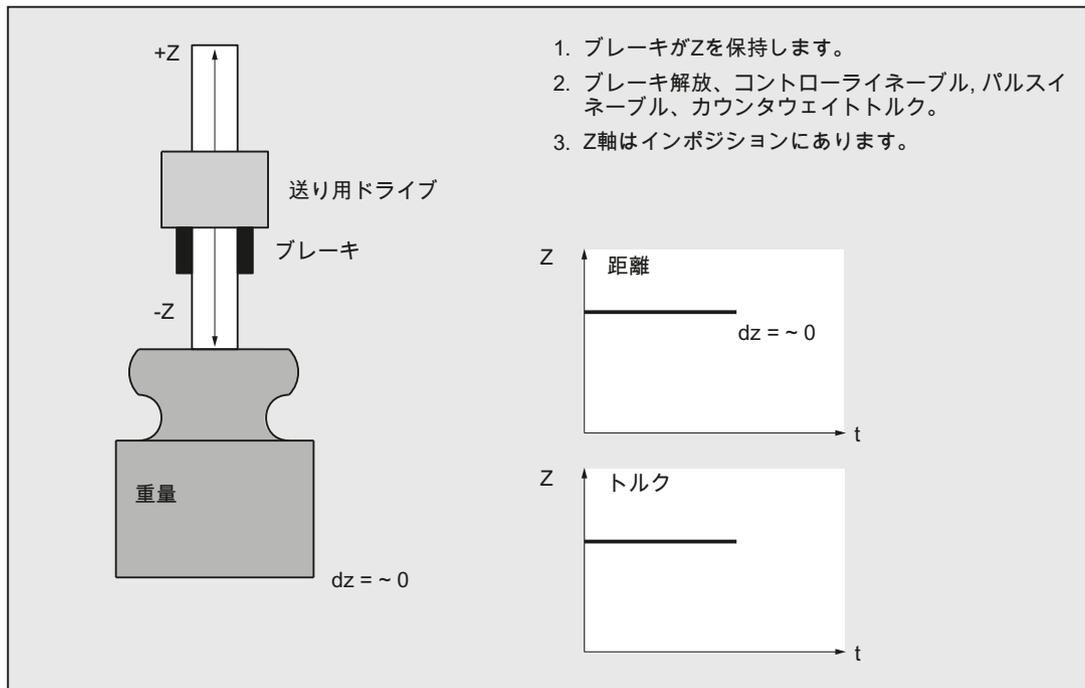


図 5-32 電子カウンタウエイト補正のある垂直軸の降下

セットアップ

注記

「電子カウンタウエイト」は、ドライブからセットアップされます!

参照先

追加情報については、以下を参照してください。

SINAMICS S120 機能マニュアル ドライブ機能

5.11.2 特殊機能:再起動遅延

機能

マシンデータの値などへの変更を有効にするには、NC を再起動してください。これは、たとえばユーザーインターフェースで NC リセットをトリガして実行できます。機械に吊り下げ軸がある場合、コントローラの起動中に閉ループ制御に障害が発生すると、軸が高さを維持しなくなります。

「再起動遅延」機能で、NC の再起動要求(NC リセット)が以前と同様に操作画面から NC に送信されます。軸の閉ループ制御が無効化される再起動は、パラメータ設定可能な時間だけ NC で遅延されます。この時間、吊り下げ軸の保持ブレーキを作動するなど、ユーザーが指定したアクションが実行されます。

#### 注記

再起動遅延は、ユーザーインターフェースで NC の再起動(NC リセット)を要求した場合のみ有効です。

コントローラをオフにしてから再びオンにする電源投入リセット、NCU の前面のリセットボタンを押した場合、または電源故障の場合、パラメータの設定した再起動遅延時間は無効です。

#### アラーム 2900 「遅延後に再起動」

再起動要求が検出された場合、アラーム 2900 「遅延後に再起動」がトリガされます。

#### アラーム応答

アラーム 2900 で、以下の応答がトリガされます。

- NC/PLC インタフェース信号は以下でリセットされます。
  - DB11 DBX6.3 = 0 (モードグループ準備完了); すべてのモードグループ
  - DB21、... DBX36.5 = 0 (チャンネル準備完了)、すべてのチャンネル
  - DB31、... DBX61.2 = 0 (軸準備完了)、すべてのチャンネル
- 現在の制限で軸/主軸を減速します。  
詳細については、次のマシンデータを参照してください。
  - MD36610 \$MA\_AX\_EMERGENCY\_STOP\_TIME (エラー発生時の減速時間)
  - MD36620 \$MA\_SERVO\_DISABLE\_DELAY\_TIME (コントローライネーブルの遮断遅延)

NC/PLC インタフェース信号「NC 準備完了」を設定したままにします。

DB10 DBX108.7 == 1

#### アラームのマスク

次のマシンデータを使って、ユーザーインターフェースでのアラーム 2900 「遅延後に再起動」の表示をマスクします。

MD11410 \$MN\_SUPPRESS\_ALARM\_MASKBit 20 = 1

これは、アラーム応答には影響しません。

### 保持ブレーキの制御

PLC の再起動動作中、0 として定義された PLC 出力はリセットされます。よって、保持ブレーキの制御は、ユーザー側で動作信号 == 0 でブレーキが作動または作動を維持し、動作信号 == 1 で解除または解除を維持するよう接続してください。

### パラメータの割り付け

再起動遅延時間は、次のマシンデータで設定されます。

MD10088 \$MN\_REBOOT\_DELAY\_TIME = <再起動遅延時間>

パラメータ設定した再起動遅延時間が 0.0 の場合、機能は無効になります。

### システム変数

NC が再起動を完了するまでの残り時間は、次のシステム変数で読み込まれます。

\$AN\_REBOOT\_DELAY\_TIME

ユーザーインタフェースから NC の再起動の要求(NC リセット)がトリガされていない場合、システム変数の値は 0.0 となります。

0.0 より大きな値は、ユーザーインタフェースから再起動要求(NC リセット)がトリガされたことを示し、NC または PLC にも再起動までの残り時間が表示されます。

### 用途例

静的シンクロナイズドアクションのシステム変数の評価

条件構成部分:0.0 より大きな値をチェックします。その場合、NC の再起動の要求(NC リセット)がユーザーインタフェースからトリガされたことを示すためです。

操作構成部分:例: 「Safety Integrated」機能の一部の「安全停止」。

## 5.12 データリスト

### 5.12.1 マシンデータ

#### 5.12.1.1 一般マシンデータ

番号	識別子: \$MN_	説明
10050	SYSCLOCK_CYCLE_TIME	基本システムのクロック周期
10070	IPO_SYSCLOCK_TIME_RATIO	補間クロック周期の係数
10082	CTRLOUT_LEAD_TIME	指令値転送時間のシフト
10083	CTRLOUT_LEAD_TIME_MAX	指令値転送時間のシフトの最大許容設定値
10088	REBOOT_DELAY_TIME	再起動遅延
18342	MM_CEC_MAX_POINTS[t]	真直度補正の補間点の最大数

#### 5.12.1.2 チャンネル別マシンデータ

番号	識別子: \$MC_	説明
20150	GCODE_RESET_VALUES	G グループのリセット

#### 5.12.1.3 軸/主軸別マシンデータ

番号	識別子: \$MA_	説明
32450	BACKLASH	バックラッシ
32452	BACKLASH_FACTOR	バックラッシの加重係数
32454	BACKLASH_MODE	バックラッシ補正モード
32456	BACKLASH_DYN	ダイナミックバックラッシ補正の補正值
32457	BACKLASH_DYN_MAX_VELO	ダイナミックバックラッシ補正值の変更の制限
32490	FRICT_COMP_MODE	摩擦補正のタイプ
32500	FRICT_COMP_ENABLE	摩擦補正有効
32510	FRICT_COMP_ADAPT_ENABLE	摩擦補正の調整は有効

## 5.12 データリスト

番号	識別子: \$MA_	説明
32520	FRICT_COMP_CONST_MAX	最大摩擦補正值
32530	FRICT_COMP_CONST_MIN	最小摩擦補正值
32540	FRICT_COMP_TIME	摩擦誤差補正の時定数
32550	FRICT_COMP_ACCEL1	調整加速度値 1
32560	FRICT_COMP_ACCEL2	調整加速度値 2
32570	FRICT_COMP_ACCEL3	調整加速度値 3
32610	VELO_FFW_WEIGHT	速度フィードフォワード制御のフィードフォワード制御係数
32620	FFW_MODE	フィードフォワード制御モード
32630	FFW_ACTIVATION_MODE	プログラムからのフィードフォワード制御の起動
32650	AX_INERTIA	トルクフィードフォワード制御の慣性
32700	ENC_COMP_ENABLE	補間補正
32710	CEC_ENABLE	真直度補正の設定
32711	CEC_SCALING_SYSTEM_METRIC	真直度補正の単位系
32720	CEC_MAX_SUM	真直度補正の最大補正值
32730	CEC_MAX_VELO	真直度補正時の速度変更
32750	TEMP_COMP_TYPE	熱変位補正タイプ
32760	COMP_ADD_VELO_FACTOR	補正の結果としての速度増加
32711	CEC_SCALING_SYSTEM_METRIC	真直度補正の単位系
32800	EQUIV_CURRCTRL_TIME	フィードフォワード制御の電流制御ループ等価時定数
32810	EQUIV_SPEEDCTRL_TIME	フィードフォワード制御の速度制御ループ等価時定数
32910	DYN_MATCH_TIME	ダイナミック応答調整の時定数
36500	ENC_CHANGE_TOL	位置フィードバック値切り替えの最大許容範囲
37302	NOCO_FILTER_TIME	ノディング補正のスムージングの時定数
37310	NOCO_INPUT_AX_1	補正関係 1: ノディング動作を引き起こす機械軸
37312	NOCO_ADAPT_AX_1	補正関係 1: その位置がノディング動作に影響を与える機械軸

番号	識別子: \$MA_	説明
37314	NOCO_ADAPT_NUM_1	補正関係 1: 位置の数、ノディング補正の調整特性
37316	NOCO_ADAPT_POS_1	補正関係 1: 位置、ノディング補正の調整特性
37318	NOCO_COMPLIANCE_1	補正関係 1: ノディング補正の整合係数
37320	NOCO_INPUT_AX_2	補正関係 2: ノディング動作を引き起こす機械軸
37322	NOCO_ADAPT_AX_2	補正関係 2: その位置がノディング動作に影響を与える機械軸
37324	NOCO_ADAPT_NUM_2	補正関係 2: 位置の数、ノディング補正の調整特性
37326	NOCO_ADAPT_POS_2	補正関係 2: 位置、ノディング補正の調整特性
37328	NOCO_COMPLIANCE_2	補正関係 2: ノディング補正の整合係数
37330	NOCO_INPUT_AX_3	補正関係 3: ノディング動作を引き起こす機械軸
37332	NOCO_ADAPT_AX_3	補正関係 3: その位置がノディング動作に影響を与える機械軸
37334	NOCO_ADAPT_NUM_3	補正関係 3: 位置の数、ノディング補正の調整特性
37336	NOCO_ADAPT_POS_3	補正関係 3: 位置、ノディング補正の調整特性
37338	NOCO_COMPLIANCE_3	補正関係 3: ノディング補正の整合係数
38000	MM_ENC_COMP_MAX_POINTS	補間補正の補間点の数

## 5.12 データリスト

## 5.12.2 セッティングデータ

## 5.12.2.1 一般セッティングデータ

番号	識別子: \$SN_	説明
41300	CEC_TABLE_ENABLE[t]	真直度補正テーブルの評価の有効化
41310	CEC_TABLE_WEIGHT[t]	真直度補正テーブルの加重係数

## 5.12.2.2 軸/主軸別セッティングデータ

番号	識別子: \$SA_	説明
43900	TEMP_COMP_ABS_VALUE	位置に依存しない熱変位補正值
43910	TEMP_COMP_SLOPE	位置に依存した熱変位補正の傾き
43920	TEMP_COMP_REF_POSITION	位置に依存した熱変位補正の基準位置

## 5.12.3 信号

## 5.12.3.1 NC からの信号

信号名称	SINUMERIK 840D sl	SINUMERIK 828D
NC 準備完了	DB10.DBX108.7	DB2700.DBX2.7

## 5.12.3.2 モードグループからの信号

信号名称	SINUMERIK 840D sl	SINUMERIK 828D
モードグループ準備完了	DB11.DBX6.3	DB3100.DBX0.3

## 5.12.3.3 チャンネルからの信号

信号名称	SINUMERIK 840D sl	SINUMERIK 828D
チャンネル準備完了	DB21, ... .DBX36.5	DB330x.DBX4.5

## 5.12.3.4 軸/主軸への信号

信号名称	SINUMERIK 840D sl	SINUMERIK 828D
ダイナミックバックラッシュ補正の有効化	DB31、... .DBX25.0	DB380x.DBX5001.0

## 5.12.3.5 軸/主軸からの信号

信号名称	SINUMERIK 840D sl	SINUMERIK 828D
原点確立済み/原点同期済み 1	DB31、...DBX60.4	DB390x.DBX0.4
原点確立済み/原点同期済み 2	DB31、...DBX60.5	DB390x.DBX0.5
軸準備完了	DB31、...DBX61.2	DB390x.DBX1.2
ダイナミックバックラッシュ補正が有効	DB31、... .DBX102.0	DB390x.DBX5006.0



# K5:相互チャネルプログラム協調とチャネルごとの試し運転

# 6

## 6.1 相互チャネルプログラム協調

### 6.1.1 協調指令(INIT、START、WAITM、WAITMC、WAITE、SETM、CLEARM)

原則として、NC のチャネルは、その中で開始されたプログラムを、そのモードグループ内の他のチャネルとは無関係に処理できます。ただし、モードグループの複数のチャネルにある複数のプログラムがワークの加工に関与している場合は、異なるチャネルのプログラム処理を以下の協調コマンドによって協調させる必要があります。

#### 必要条件

プログラム協調に使用するすべてのチャネルは、同じモードグループに属します。

```
MD10010 $MC_ASSIGN_CHAN_TO_MODE_GROUP[<チャネル>] = <モードグループ番号>
```

#### チャネル番号ではなくチャネル名

チャネル番号の代わりに MD20000 \$MC\_CHAN\_NAME[<チャネルインデックス>]に入力されたチャネル名は、プログラム協調のための予約処理のパラメータとして使用することもできます。まず NC プログラムでチャネル名の使用を有効にする必要があります。

```
MD10280 $MN_PROG_FUNCTION_MASK, bit 1 = TRUE
```

#### 構文

```
INIT(<ChanNr>, <Prog>, <AckMode>)  
START(<ChanNr>, <ChanNr>, ...)  
WAITM(<MarkNr>, <ChanNr>, <ChanNr>, ...)  
WAITE(<ChanNr>, <ChanNr>, ...)  
WAITMC(<MarkNr>, <ChanNr>, <ChanNr>, ...)  
SETM(<MarkNr>, <MarkNr>, ...)  
CLEARM(<MarkNr>, <MarkNr>, ...)
```

#### 意味

INIT():	指定したチャネルで処理する NC プログラムを選択する予約処理
START():	それぞれのチャネルで選択したプログラムを開始するための予約機能

6.1 相互チャンネルプログラム協調

WAITM():	<p>指定したチャンネルで待機マークに到達するまで待機する予約処理</p> <p>指定した待機マークは同じチャンネルの WAITM で設定します。その前のブロックをイグザクトストップで終了します。待機マークは、同期制御が終わると解除されます。</p> <p>チャンネルごとに <b>10</b> 個のマークを同時に設定できます。</p>		
WAITE():	<p>複数の他のチャンネルのプログラムの終了まで待機する予約処理</p>		
WAITMC(): 1)	<p>指定したチャンネルで待機マークに到達するまで待機する予約処理</p> <p>WAITM とは異なり、イグザクトストップの減速は他のチャンネルが待機マークに到達していない場合のみ開始します。</p>		
SETM(): <sup>1)</sup>	<p>チャンネル協調用の複数の待機マークを設定する予約処理</p> <p>これはそのチャンネル内の実行には影響しません</p> <p>SETM はチャンネルリセットと <b>NC</b> スタート後も有効です。</p>		
CLEARM(): 1)	<p>チャンネル協調用の複数の待機マークを削除する予約処理</p> <p>これはそのチャンネル内の実行には影響しません</p> <p>CLEARM () はチャンネル内のすべての待機マークを削除します。</p> <p>CLEARM (0) は待機マーク「0」のみを削除します。</p> <p>CLEARM はチャンネルリセットと <b>NC</b> スタート後も有効です。</p>		
<ChanNr>:	<p>チャンネル番号</p> <p>自身のチャンネルの番号を指定する必要はありません。</p>		
	<table border="1"> <tr> <td>タイプ:</td> <td>INT</td> </tr> </table>	タイプ:	INT
タイプ:	INT		
<Prog>:	<p>絶対または相対パス指定(オプション)+プログラム名称</p>		
	<table border="1"> <tr> <td>タイプ:</td> <td>STRING</td> </tr> </table>	タイプ:	STRING
	タイプ:	STRING	
<p>パス指定については、以下を参照してください。</p> <p><b>参照先</b></p> <p>『プログラミングマニュアル、加工スケジュール』、「ファイルとプログラムの管理」&gt;「プログラムメモリ」&gt;「プログラムメモリのファイルのアドレス指定」のセクション</p>			

<AckMode> >:	応答モード(オプション)				
	タイプ:	CHAR			
	規格値:	<table border="1"> <tr> <td>"n"</td> <td>                     応答なし                      命令の送信後にプログラムの実行を続行します。命令を正常に実行できない場合でも、送信側には通知されません。                 </td> </tr> <tr> <td>"S"</td> <td>                     同期確認応答                      プログラムの実行は、受信コンポーネントが命令を応答するまで中止します。確認の応答があった場合は、次の命令を実行します。確認の応答がない場合は、故障メッセージを出力します。                 </td> </tr> </table>	"n"	応答なし 命令の送信後にプログラムの実行を続行します。命令を正常に実行できない場合でも、送信側には通知されません。	"S"
"n"	応答なし 命令の送信後にプログラムの実行を続行します。命令を正常に実行できない場合でも、送信側には通知されません。				
"S"	同期確認応答 プログラムの実行は、受信コンポーネントが命令を応答するまで中止します。確認の応答があった場合は、次の命令を実行します。確認の応答がない場合は、故障メッセージを出力します。				

<MarkNr>:	待機マークの番号  <b>注</b> マルチチャネルシステムでは、最大 100 個の待機マーク(待機マーク 0 ~ 99)を使用できます。 シングルチャネルシステムで使用できるのは、待機マーク 0 のみです。
-----------	--

1) ユーザー別の通信/チャネル協調では、SETM / CLEARM を使用して待機マークを配置できます。また、条件付き待機指令 WAITMC を使用する必要もありません。チャネルリセットおよび NC スタートの後も、待機マークの値は保持されます。

例

**MD20000 からチャネル名の使用を開始する**

- パラメータ設定
 

```
MD10280 $MN_PROG_FUNCTION_MASK, bit 1 = TRUE
$MC_CHAN_NAME[ 0 ] = "MACHINING" ; チャネル 1 の名前
$MC_CHAN_NAME[ 1 ] = "INFEED" ; チャネル 2 の名前
```
- プログラミング

プログラムコード	コメント
START (MACHINING)	; チャネル 1 の開始
START (INFEED)	; チャネル 2 の開始

**START でローカルの「チャネル名称」と「ユーザー変数」を使用**

プログラムコード	コメント
DEF INT MACHINE = 1	; チャネル 1 のユーザー変数の定義
DEF INT LOADER = 2	; チャネル 2 のユーザー変数の定義

6.1 相互チャンネルプログラム協調

プログラムコード	コメント
...	
START (MACHINE)	; チャンネル 1 の開始
START (LOADER)	; チャンネル 2 の開始

**START** でローカルの「チャンネル名称」、ユーザー変数およびパラメータ設定したチャンネル名称を使用

プログラムコード	コメント
DEF INT chanNo1	; チャンネル 1 のユーザー変数の定義
DEF INT chanNo2	; チャンネル 2 のユーザー変数の定義
chanNo1 = CHAN_1	; パラメータ設定したチャンネル名称チャンネル 1 の割り当て
chanNo2 = CHAN_2	; パラメータ設定したチャンネル名称チャンネル 2 の割り当て
...	
START (chanNo1)	; チャンネル 1 の開始
START (chanNo2)	; チャンネル 2 の開始

**絶対パス指定による INIT 命令**

チャンネル 2 でのプログラム/\_N\_MPF\_DIR/\_N\_ABSPAN1\_MPF の選択

**プログラムコード**

```
INIT (2, "/_N_WCS_DIR/_N_SHAFT1_WPD/_N_CUT1_MPF")
```

**プログラム名称を使用した INIT 命令**

名称"MYPROG"のプログラムの選択検索パスを使用してプログラムが検索されます。

**プログラムコード**

```
INIT (2, "MYPROG")
```

## WAITM によるプログラム協調

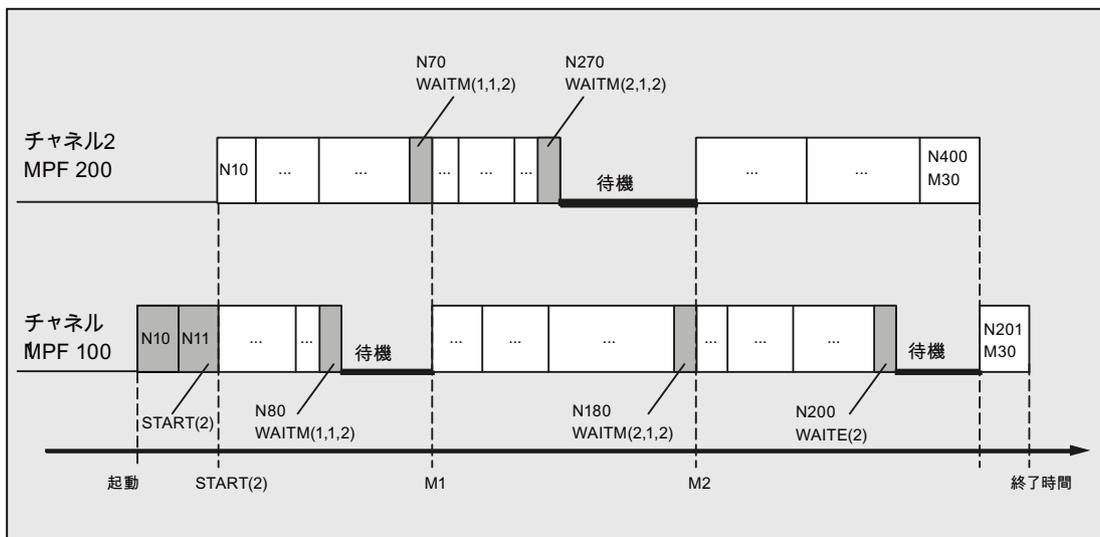
- チャンネル 1:プログラム/\_N\_MPF\_DIR/\_N\_MPF100\_MPF はすでに選択され、開始されています。

プログラムコード	コメント
	; プログラム MPF100
N10 INIT(2, "MPF200", "N")	; プログラム MPF200 の選択、チャンネル 2
N11 START(2)	; チャンネル 2 の開始
...	
N80 WAITM(1,1,2)	; チャンネル 1 と 2 で WAIT マーク 1 を待機
N81 ...	; チャンネル 1、N81 とチャンネル 2、N71 は ; 同時に開始されます
...	
N180 WAITM(2,1,2)	; チャンネル 1 と 2 で WAIT マーク 2 を待機
N181 ...	; チャンネル 1、N181 とチャンネル 2、N271 は ; 同時に開始されます
...	
N200 WAITE(2)	;チャンネル 2 のプログラムの終了を待機
N201 ...	; N201 はプログラム終了後にのみ開始されます ; チャンネル 2 の MPF200
N201 M30	; チャンネル 1 のプログラム終了

- チャンネル 2:チャンネル 1 で、ブロック N10 と N20 を使用してチャンネル 2 のためにプログラム MPF200\_MPF が選択され、開始されます。

プログラムコード	コメント
;\$PATH=/_N_MPF_DIR	; プログラム MPF200
...	
N70 WAITM(1,1,2)	チャンネル 1 と 2 で WAIT マーク 1 を待機
N71 ...	; チャンネル 1、N81 とチャンネル 2、N71 は ; 同時に開始されます
...	
N270 WAITM(2,1,2)	チャンネル 1 と 2 で WAIT マーク 2 を待機
N271 ...	; チャンネル 1、N181 とチャンネル 2、N271 は ; 同時に開始されます
...	
N400 M30	チャンネル 2 のプログラム終了

6.1 相互チャンネルプログラム協調



必要条件

待機マーク後の後続ブロック実行の非同期開始

待機マークを使用したチャンネル協調の場合は、次のブロックの実行時に非同期開始が発生する可能性があります。この動作は、共通の待機マークに到達する直前に、チャンネルのいずれかで操作がトリガされた場合に発生します。その結果、この残移動距離削除で自動的に再位置決め(REPOSA)が行われます。

前提条件:チャンネル 1 と 2 での現在の軸割り当て

- チャンネル 1:軸 X1 と軸 U
- チャンネル 2:軸 X2

表 6-1 チャンネル 1 と 2 の処理の流れ

チャンネル 1	チャンネル 2	説明
...	...	チャンネル 1 と 2 で任意の処理
N100 WAITM(20,1,2)		チャンネル 1:WAIT マークに到達し、チャンネル 2 との同期のために待機

チャンネル 1	チャンネル 2	説明
<b>GETD(U)処理の開始:</b> <ul style="list-style-type: none"> <li>● 軸入れ替え</li> <li>● 残移動距離削除</li> <li>● REPOSA</li> </ul> <b>終了</b>	N200 GETD (U)	チャンネル 2:チャンネル 1 に軸 U を要求 チャンネル 1:バックグラウンドで GET (U) の処理
	N210 WAITM(20,1,2)	チャンネル 2:WAIT マークに到達 ⇒ これによってチャンネル 1 と 2 の同期が完了
	N220 G0	チャンネル 2:N220 の処理を開始
N110 G0 X1=100	X2=100	チャンネル 1:N110 の処理をずらして開始

下記も参照

連続軌跡モードでの条件待機(WAITMC) (ページ 437)

### 6.1.2 連続軌跡モードでの条件待機(WAITMC)

チャンネルでの WAITMC の処理中に、他のチャンネルの必要な待機マークがすでにすべて受信されている場合は、このチャンネルで移動動作が減速されず、追加プログラム実行が停止されませんが、以下のブロックの処理が続行されます。

#### 必要条件

このチャンネルでは、以下の機能を有効にしてください。

- 連続軌跡モード(G64、G641、G642、G643、G644、または G645)
- 「先読み」機能

#### 減速動作

##### 減速なし

WAITMC 呼び出しの前に移動ブロックを使用して起動すると、同期される他のチャンネルの待機マークがチェックされます。これらがすでに使用可能な場合は、減速が行われません。

6.1 相互チャネルプログラム協調

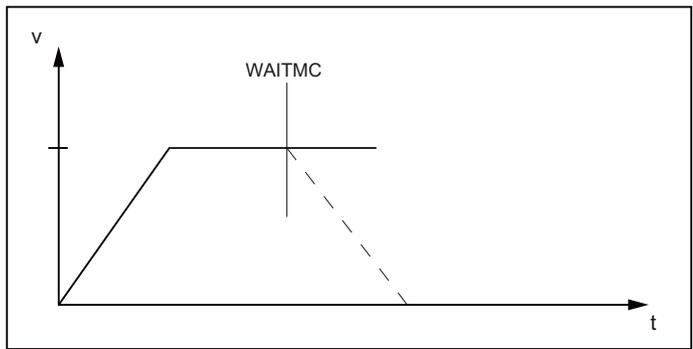


図 6-1 WAITMC による条件待機のための軌跡速度変更:すでに有効なすべてのチャネルに対する待機マーク

短時間の減速

同期するチャネルの待機マークがない場合、減速が開始されます。減速中に、補間器クロック周期ごとに、まだ受信していない待機マークを受信したかどうかチェックされます。受信した場合は、軸が加速され、プログラム指令された速度に戻ります。

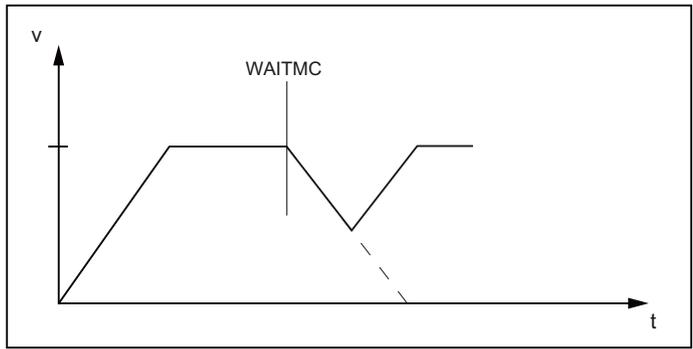


図 6-2 WAITMC による条件待機のための軌跡速度変更:最後の待機マークが減速中に到達

停止状態までの減速

減速中に待機マークを受信しない場合は、軸が停止するまで減速され、システムが待機状態になります。待機マークを受信すると、軸が加速され、プログラム指令された速度に戻ります。

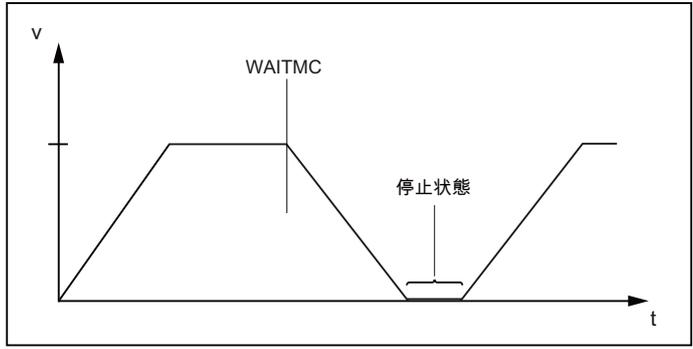


図 6-3 WAITCM による条件待機のための軌跡速度変更:減速後に最後の待機マークに到達します

### 減速中のブロック切り替え(IPOBRKA)

IPOBRKA (ブロック切り替え条件「減速カーブ」)動作終了条件が有効な場合は、まだ受信していなかった最後の待機マークを受信すると、次のブロックに切り替わり、軸が加速されます。

### 例 1:軌跡制御運転の条件待機

この例は概略図であり、同期制御処理に関連する命令のみを示しています。

#### チャンネル 1

プログラムコード	コメント
N10 INIT (2, 「_N_200_MPF」, 「n」)	; 相手のプログラムチャンネル 2 を選択。
N11 INIT (3, 「_N_300_MPF」, 「n」)	; 相手のプログラムチャンネル 3 を選択。
N15 START (2,3)	; チャンネル 2、3 でプログラムを起動。
...	; チャンネル 1 の加工。
N20 WAITCM (7,2,3)	; チャンネル 2 と 3 からのマーク 7 の条件待機。
...	; チャンネル 1 の追加加工。
N40 WAITCM (8,2)	; チャンネル 2 からのマーク 8 の条件待機。
...	; チャンネル 1 の追加加工。
N70 M30	; チャンネル 1 の終了。

#### チャンネル 2

プログラムコード	コメント
N200 ...	; チャンネル 2 の加工。
N210 SETM (7)	; チャンネル 2 が待機マーク 7 を設定。
...	; チャンネル 2 の追加加工。
N250 SETM (8)	; チャンネル 2 が待機マーク 8 を設定。
N260 M30	; チャンネル 2 の終了。

#### チャンネル 3

プログラムコード	コメント
N300 ...	; チャンネル 3 の加工。
...	
N350 WHEN <条件> DO SETM (7)	; シンクロナイズドアクションで待機マークを設定。
...	; チャンネル 3 の追加加工。
N360 M30	; チャンネル 3 の終了。

6.1 相互チャネルプログラム協調

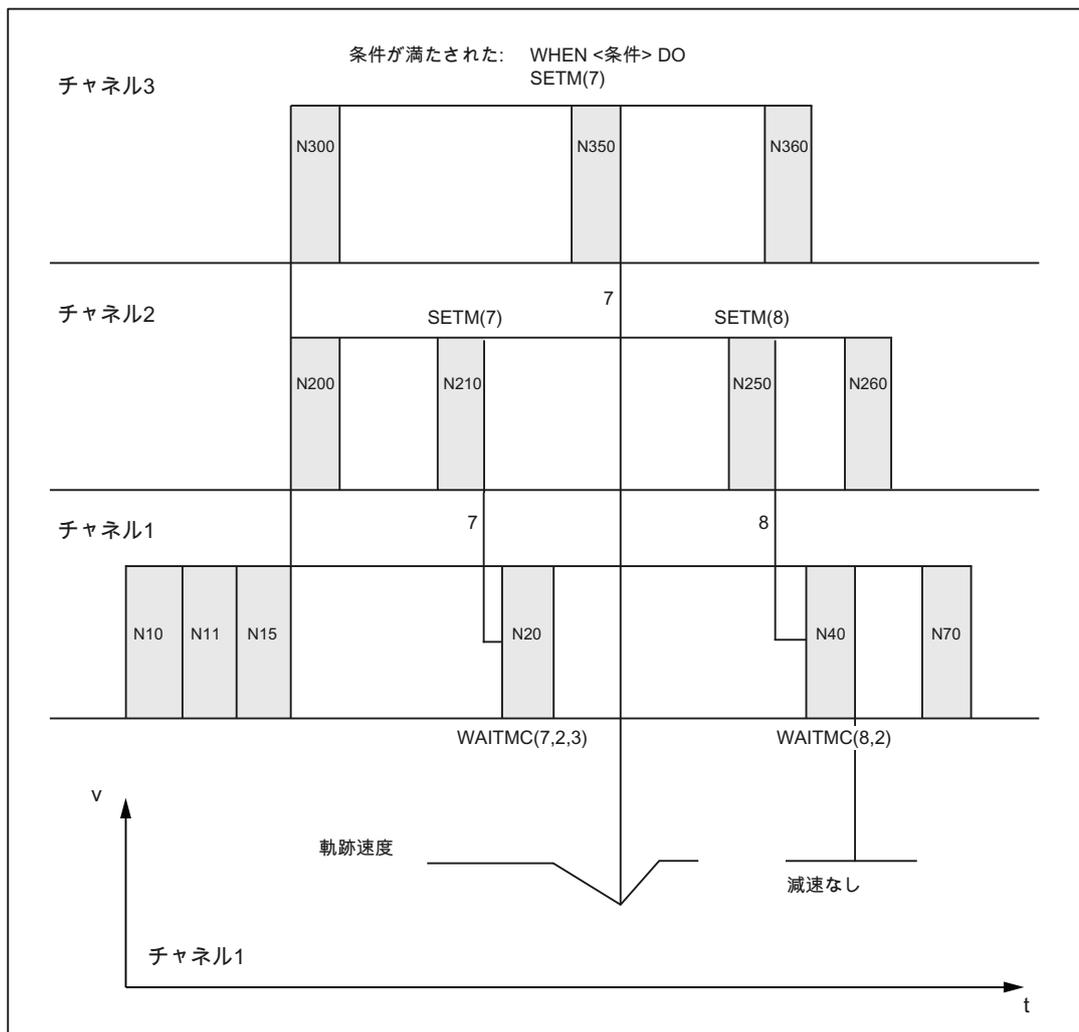


図 6-4 3つの関連チャネルにおける軌跡制御運転の条件待機(概略図)

例 2 :WAITMC と読み込み停止

補助機能 M555 は、軸が移動中に読み込み禁止を生成する間にチャンネル 3 で出力されま  
す。WAITMC がブロック N312 に割り当てられるため、待機マークが設定され、チャネ  
ル 2 が進行します。読み込み停止により、チャンネル 3 のプログラム処理が停止します。

注記

有効な G64 では、WAITMC が専用ブロックを作成しませんが、代わりに前のブロックに  
追加されます。連続軌跡モードが有効な場合は、速度の低下を防ぐ必要があります。し  
たがって、先行するブロックが読み込み禁止などで停止された場合でも、WAITMC が実  
行されます。

## チャネル 2

プログラムコード	コメント
N112 G18 G64 X200 Z200 F567	; チャネル 2 の処理
N120 WAITMC(1,2,3)	; 待機マーク 1 を条件付きで待機
	; チャネル 2 および 3 から
...	; チャネル 2 の追加処理、このとき
	; WAITMC をブロック N312 へ付加
...	; チャネル 2 の追加処理
N170 M30	; チャネル 2 の終了

## チャネル 3

	コメント
	; 移動中の読み込み禁止 M555
N300	; チャネル 3 の処理
N312 G18 G64 D1 X180 Z300 M555	
N320 WAITMC(1,2,3)	; 読み込み無効により待機

## 6.2 チャネル単位の試し運転

## 6.2.1 機能

「チャネル単位の試し運転」機能を使って、相互に同期された複数のチャネルの NC プログラムを試験または試し運転します。「通常運転」状態のチャネルのチャネル軸/主軸のプログラム指令された移動動作により、機械軸の移動が発生します。「プログラムテスト」状態のチャネルには、以下が適用されます。

- 初期設定では、通常運転の各軸に対して、同じ指令値が生成されますが、これらは機械軸では出力されません。
- 初期設定では、主軸の指令値が機械軸で出力されます。
- 無効な軸/主軸の現在値が指令値から内部的に生成され、表示されます。
- チャネル同期の指令は正常に処理されます。
- NC/PLC インタフェース信号は正常に処理されます。
- プログラムの処理時間は、通常動作の場合と同じです。

必要な場合は一時的に、「プログラムテスト」状態を軸ごとに終了できます。その場合、機械軸にセットポイントが再度出力され、機械で実際に移動します。

## 6.2 チャンネル単位の試し運転

下記も参照

K10:チャンネル間の軸入れ替え (ページ 455)

### 6.2.2 処理

通常、チャンネルは工具を作業領域で移動します。同一の作業領域で複数のチャンネルがそれぞれ工具を移動する場合、工具の移動を同期してください。以下の同期が可能です。

- プログラム協調指令 WAITM、WAITMC、WAITE、START によるチャンネル同期。
- PLC ユーザープログラムおよび NC/PLC インタフェース信号によるチャンネル同期。  
たとえば、チャンネルから PLC に M 機能を出力し、PLC からそのチャンネルに読み込み停止を出力します。
- 軸の入れ替え:チャンネルは、他のチャンネルが軸を解放するまで待機します。
- NC プログラムのグローバル変数を使った同期。
- 相互チャンネル連結
- 軸コンテナの回転
- 実行プログラムでの並列のシンクロナイズドアクションを含むプログラムのテスト、およびシンクロナイズドアクションのチャンネルとの同期化

これらの一般的な条件では、1つのチャンネルだけを開始するのはほとんど不可能です。この条件では最初の同期位置に停止したままになります。

「チャンネル単位の試し運転」機能を使用して、必要なすべてのチャンネルのみを起動して移動イネーブル信号を受信するようにし、そのプログラム指令された移動動作の相互作用をテストすることができます。他のチャンネルは「プログラムテスト」状態になります。

これをおこなうには、チャンネルを起動する前に、機械のオペレータが、「プログラムテスト」状態で移動するチャンネルを定義する必要があります。これは[プログラム制御]メニューの操作画面でおこないます。

## 6.2.3 シングルチャネル表示

### 選択

「機械」基本表示に表示されるチャネルの「プログラムテスト」(PRT)は、ユーザーインタフェースで選択します。以下は、SINUMERIK Operate の場合の手順です。

1. ソフトキー:操作エリア「機械」>「プログラム制御」
2. 「プログラム制御」メニュー:「プログラムテスト(PRT)」チェックボックスを選択します。

### NC/PLC インタフェース信号

プログラムテスト(PRT)を選択すると、以下の NC/PLC インタフェース信号が設定されます(下の注「インタフェース信号の自動伝送」を参照してください)。

- チャネル
  - DB21、... DBX25.7 == 1 (HMI から:プログラムテストを選択)
  - DB21、... DBX1.7 == 1 (PLC から:プログラムテスト起動)
  - DB21、... DBX33.7 == 1 (NC から:プログラムテスト有効)
- 軸
  - DB31、... DBX128.0 == 0 (HMI から:プログラムテストのマスク)
  - DB31、... DBX128.1 == 0 (HMI からプログラムテストを起動)
  - DB31、... DBX14.0 == 0 (PLC から:プログラムテストのマスク)
  - DB31、... DBX14.1 == 0 (PLC から:プログラムテスト起動)
- 主軸
  - DB31、... DBX128.0 == 1 (HMI から:プログラムテストのマスク)
  - DB31、... DBX128.1 == 0 (HMI から:プログラムテスト起動)
  - DB31、... DBX14.0 == 1 (PLC から:プログラムテストのマスク)
  - DB31、... DBX14.1 == 0 (PLC から:プログラムテスト起動)

---

### 注記

#### 移動動作の初期状態

チャネルで「プログラムテスト」を選択した後の移動動作に関する初期状態は、次のとおりです。

- 軸:無効
  - 主軸:有効
-

6.2 チャネル単位の試し運転

注記

インタフェース信号の自動伝送

HMI 要求信号 DB21、... DBX128.0 / .1 は、FB1 パラメータ **MMCToIF** を **TRUE** に設定した場合に、基本 PLC プログラムから PLC 要求信号 DB21、... DBX14.0 / .1 にのみ伝送されます。このパラメータを設定しない場合は、PLC 要求信号を PLC ユーザープログラムで設定する必要があります。

主軸の移動動作の無効化

「プログラムテスト」中に主軸を**移動しない**場合は、PLC ユーザープログラムで明示的に主軸を無効化/ロックしてください。

- DB31、... DBX128.0 = 0 (プログラムテストのマスク)
- DB31、... DBX128.1 = 1 (プログラムテストの有効化)
- DB31、... DBX14.0 == 0 (PLC から:プログラムテストのマスク)
- DB31、... DBX14.1 == 1 (PLC から:プログラムテスト起動)

<b>通知</b>
<p><b>主軸の移動動作</b></p> <p>チャネルの主軸の移動動作は、初期設定として「プログラムテスト」で有効になっています。</p>

軸の移動動作の有効化

「プログラムテスト」中にチャネル軸を移動する場合は、PLC ユーザープログラムで明示的にチャネル軸を有効化してください。

- DB31、... DBX14.0 == 1 (PLC から:プログラムテストのマスク)
- DB31、... DBX14.1 == 0 (PLC から:プログラムテスト起動)

切り替えが可能なタイミング

- チャネル  
チャネル別に「プログラムテスト」状態を無効/有効にするインタフェース信号 (DB21、... DBX25.7 または DBX1.7)は、「リセット」または「中断」チャネル状態でのみ切り替えることができます。
- 軸/主軸  
軸別に「プログラムテスト」状態を無効/有効にするインタフェース信号(DB31、... DBX128.0 / .1 または DBX14.0 / .1)は、いつでも切り替えることができます。

## 6.2.4 マルチチャンネル表示

### 前提条件

「プログラム制御」メニューのマルチチャンネル表示でチャンネルと主軸が表示されるようにするには、以下の必要条件を満たす必要があります。

- チャンネル
  - 複数のチャンネルがパラメータ設定されている(MD10010 \$MN\_ASSIGN\_CHAN\_TO\_MODE\_GROUP[<チャンネル>] ≠ 0)
  - オプション「programSYNC」(6FC5800-0AP05-0YB0)が設定/ライセンス供与されている
- 主軸
  - チャンネルでパラメータ設定されている主軸の意味が定義されている(MD52206 \$MC\_AXIS\_USAGE[<主軸>] ≠ 0)

### 選択

制御でパラメータ設定された1つまたは複数のチャンネルの「プログラムテスト」(PRT)を、ユーザーインターフェースで選択します。たとえば、SINUMERIK Operate の場合は、次のように操作します。

1. ソフトキー:操作エリア「機械」>「プログラム制御」
2. ソフトキー:「試し運転」
3. 「プログラム制御」メニュー:グループ「試し運転」の選択を解除>「チャンネル x」チェックボックス。

### チャンネル/主軸のチェックボックス

「プログラム制御」メニューで、チェックボックス「チャンネル x」または「主軸 x」を使用して、以下の動作を「試し運転」グループに対して設定できます。

チェックボックス	選択済み	選択解除
「チャンネル x」	(通常運転)になっています。	プログラムが計算されるのみ (チャンネルはテスト運転)
「主軸 x」	(通常運転)になっています。	主軸 x での加工なし (主軸はテスト運転)

## 6.2 チャネル単位の試し運転

## NC/PLC インタフェース信号

チェックボックス「チャネル x」を選択解除した後、チャネルは「プログラムテスト (PRT)」状態になります。以下の NC/PLC インタフェース信号が設定されます(下の注「インタフェース信号の自動伝送」を参照してください)。

- チャネル
  - DB21、... DBX25.7 == 1 (HMI から:プログラムテストを選択)
  - DB21、... DBX1.7 == 1 (PLC から:プログラムテスト起動)
  - DB21、... DBX33.7 == 1 (NC から:プログラムテスト有効)
- 軸
  - DB31、... DBX128.0 == 0 (HMI から:プログラムテストのマスク)
  - DB31、... DBX128.1 == 0 (HMI からプログラムテストを起動)
  - DB31、... DBX14.0 == 0 (PLC から:プログラムテストのマスク)
  - DB31、... DBX14.1 == 0 (PLC から:プログラムテスト起動)

初期設定では、主軸のチェックボックスが選択されています。つまり、主軸は**通常運転**をおこないます。

- 主軸
  - DB31、... DBX128.0 == 1 (HMI から:プログラムテストのマスク)
  - DB31、... DBX128.1 == 0 (HMI から:プログラムテスト起動)
  - DB31、... DBX14.0 == 1 (PLC から:プログラムテストのマスク)
  - DB31、... DBX14.1 == 0 (PLC から:プログラムテスト起動)

## 注記

## 移動動作の初期状態

チャネルで「プログラムテスト」を選択した後の移動動作に関する初期状態は、次のとおりです。

- 軸:無効
- 主軸:有効

## 注記

## インタフェース信号の自動伝送

HMI 要求信号 DB21、... DBX128.0 / .1 は、FB1 パラメータ **MMCToIF** を **TRUE** に設定した場合に、基本 PLC プログラムから PLC 要求信号 DB21、... DBX14.0 / .1 にのみ伝送されます。このパラメータを設定しない場合は、PLC 要求信号を PLC ユーザープログラムで設定する必要があります。

**主軸の移動動作の無効化**

「プログラムテスト」中にチャネルの主軸を**移動しない**場合は、PLC ユーザープログラムで明示的に主軸を無効化/ロックしてください。

- DB31、... DBX128.0 = 0 (プログラムテストのマスク)
- DB31、... DBX128.1 = 1 (プログラムテストの有効化)
- DB31、... DBX14.0 == 0 (PLC から:プログラムテストのマスク)
- DB31、... DBX14.1 == 1 (PLC から:プログラムテスト起動)

**通知****主軸の移動動作**

チャネルの主軸の移動動作は、初期設定として「プログラムテスト」で有効になっています。

**軸の移動動作の有効化**

「プログラムテスト」中にチャネル軸を移動する場合は、PLC ユーザープログラムで明示的にチャネル軸を有効化してください。

- DB31、... DBX14.0 == 1 (PLC から:プログラムテストのマスク)
- DB31、... DBX14.1 == 0 (PLC から:プログラムテスト起動)

**切り替えが可能なタイミング**

- チャネル  
チャネル別に「プログラムテスト」状態を無効/有効にするインタフェース信号 (DB21、... DBX25.7 または DBX1.7)は、「リセット」または「中断」チャネル状態でのみ切り替えることができます。
- 軸/主軸  
軸別に「プログラムテスト」状態を無効/有効にするインタフェース信号(DB31、... DBX128.0 / .1 または DBX14.0 / .1)は、いつでも切り替えることができます。

6.2 チャンネル単位の試し運転

6.2.5 システム変数

「プログラムテスト」状態は、システム変数を使用して確認できます。

- 操作画面での表示、シンクロナイズドアクション、またはパートプログラムでの先読み停止処理の場合は、以下のシステム変数を使用します。

\$AC_ISTEST	チャンネルの「プログラムテスト」状態 チャンネルの「プログラムテスト」状態が有効な場合は、TRUE (1)を返します。
\$AA_ISTEST[<n>]	軸<n>の「プログラムテスト」状態 軸<n>の「プログラムテスト」状態が有効な場合は、TRUE (1)を返します。

- 次のシステム変数は、パートプログラムでは先読み停止をおこないません。

\$P_ISTEST	チャンネルの「プログラムテスト」状態が有効な場合は、TRUE (1)を返します。
------------	--

例

チャンネルは「プログラムテスト」で動作しており、軸「C」は「プログラムテストのマスク」を使用して取り消されています。システム変数を使用した確認に対して、次の結果が返されます。

- \$AC\_ISTEST == TRUE
- \$P\_ISTEST == TRUE
- \$AA\_ISTEST[C] == FALSE

6.2.6 必要条件

軸入れ替え

「軸入れ替え」機能により軸/主軸を複数のチャンネルで認識し、軸/主軸を入れ替えてプログラム指令することができます(「K10:チャンネル間の軸入れ替え (ページ 455)」の章を参照してください)。

「プログラムテスト」機能と「チャンネル単位の試し運転」機能の組み合わせでは、軸入れ替えで以下の条件を守ってください。

- チャンネルのいずれか1つが「プログラムテスト」状態である場合は、このチャンネルから入れ替える軸を選択し、「プログラムテスト」状態でないチャンネルに挿入します。軸停止を有効にして入れ替えられた軸では、「プログラムテスト」状態のチャンネルとそうでないチャンネルを使用した変更の場合、軸自体の状態は変更されません(例3を参照してください)。
- プログラムテストの場合、パートプログラムの終了/リセットでは、補間されないすべての軸/主軸に対して、再原点同期が実際の位置で行われます。結果として、プログラムの終了後初めておこなわれる軸入れ替えでは、軸はプログラムの終了時にのみチャンネルを終了するため、シミュレートされた到達位置は、受信チャンネルに伝送されません。

---

#### 注記

また、プログラムの最後に WAIT タグを含めて、プログラムが同時に終了するようにしてください。

---

## 6.2.7 例

**例 1:チャンネル2を3チャンネルのシステムでテストします。**

試験オプション 1:SERUPRO を使用しないプログラムテスト

1. ユーザーは実際に物理的に移動する軸/主軸を決定します。これらの軸に「プログラムテストのマスク」を設定します。
2. チャンネル1と3に対して「プログラムテスト」状態を選択します。
3. PLCを介してチャンネル1、2および3を開始します。
4. 「プログラムテスト」は、プログラムの終了後に再度選択できます。
5. 「プログラムテストのマスク」の実際の設定が他の状況(チャンネル1またはチャンネル3のテスト)でも有用な場合は、この信号を設定したままにすることができます。確かに、これは多くの場合に実用的です。

試験オプション 2:SERUPRO を使用したプログラムテスト

1. ユーザーは実際に物理的に移動する軸/主軸を決定します。これらの軸に「プログラムテストのマスク」を設定します。
2. チャンネル1と3に対して「プログラムテスト」状態を選択します。
3. PLCを介してチャンネル1、2および3を開始します。
4. 異常またはアラームが発生し、ユーザーが RESET で中断します。
5. 3つのチャンネルすべての中断位置で SERUPRO を実行します。
6. 3つのチャンネルすべてで検索対象に達しました。

6.2 チャンネル単位の試し運転

- 7. 3つのチャンネルすべてを開始します。
- 8. これでチャンネル1と3は再度「プログラムテスト」状態になり、「チャンネル単位の試し運転」が実行されます。

**例2: 「プログラムテストのマスク」の有効化**

チャンネルはプログラムテスト状態です。運転時、軸「Y」に対して「プログラムテストのマスク」を開始します(ブロック N1010 で)。

プログラムコード	コメント
N1000 G0 Y1000	
N1010 G4 F10	
N1020 G0 G91 Y=10	; インクリメンタル移動。
N1030 M30	

この手順では、プログラムは 1010 の位置に移動します。つまり、この軸の「1000」のシミュレートされた成分を「プログラムテストのマスク」を有効にした後で移動します。

**例3: プログラムテストと軸入れ替え**

チャンネル1の軸 X1 とチャンネル2の軸 X2 が、NC の最初の機械軸 AX1 に割り当てられます。

「プログラムテスト」を使用するチャンネル1	「プログラムテスト」を使用しないチャンネル2
N10010 G0 G90 X0	
N10020 X1=100	
N10030 WAITM(91,1,2)	
N10040 WAITM(92,1,2)	
N10050 M0	
N10060 M30	
	N20010 WAITM(91,1,2)
	N20020 G91 G0 X2=10
	N20030 WAITM(92,1,2)
	N20040 M0
	N20050 M30

ブロック N20040 で、機械軸 AX1 がチャンネル2に入れ替えられ、チャンネル1の軸の最後の位置が引き継がれ、軸が位置 110 に移動されます。

## 6.3 必要条件

### 6.3.1 MDI モード:経路制御モードと WAITMC

#### MDI モード:連続軌跡モードと WAITMC

MDI モードでは、MDI ブロックバッファの実行の開始時点で、経路制御モード(G64/G604)との組み合わせで WAITMC を MDI ブロックバッファの最後のブロックに配置することは許可されません。これを守らない場合、最後の 1 つ前の移動ブロックでプログラムが停止し、リセット以外の方法ではキャンセルできなくなります。

#### 例

NC スタート前は、次のブロックが MDI ブロックバッファに含まれています。

プログラムコード	コメント
N10 G64 G1 G94 F5000 X100	; MDI ブロックバッファの最初のブロックでの ; 連続軌跡モード
N20 X200	
N30 X300	; 最後から 1 つ前の移動ブロック
N40 X400	; プログラムの実行がここで停止
N50 WAITMC (...)	; 理由は N50 の WAITMC ; リセットによるキャンセルが必要

### 6.3.2 WAIT 指令後の移動動作の非同期開始

WAIT 指令によるチャネル協調の場合は、次のブロックの実行時に非同期開始が発生する可能性があります。この動作は、共通の同期位置に到達する直前に、同期されるチャネルのいずれかで操作がトリガされた場合に発生します。その結果、この残移動距離削除で自動的に再位置決め(REPOSA)が行われます。

前提条件:チャネル 1 と 2 での現在の軸割り当て

- チャネル 1:軸 Z1 および W
- チャネル 2:軸 Z2

6.3 必要条件

例 1:時間オフセットのある移動動作の開始

チャンネル 1 および 2 で次の指令シーケンスを実行すると、同期位置 N110 / N210 の後に、時間オフセットのある移動動作が開始されて問題が発生する可能性があります。

表 6-2 チャンネル 1 と 2 のシーケンス(時間)

チャンネル 1	チャンネル 2	説明
...	...	チャンネル 1 と 2 で任意の処理
N100 W100		チャンネル 1:移動軸 W
...	...	チャンネル 1 と 2 で任意の処理
N110 WAITM(2,1,2)	N210 WAITM(2,1,2)	チャンネル 1 および 2:同期を待機
「AUTOGET(W)」の 開始: ● 軸入れ替え ● 残移動距離削除 ● REPOSA 終了 N120 Z1=100	「AUTOGET(W)」、 およびチャンネル 1 か らの軸 W を待機 N220 Z2=100 W100	チャンネル 2:時間オフセットのある移動動作 の開始  チャンネル 1:移動動作の開始

例 2:移動動作の同時開始

例 1 の、時間オフセット(遅延)のある移動動作の開始は、軸 W の入れ替えと追加の同期位置の挿入を明示的に有効にして要求すると、回避できます。チャンネル 1 および 2 で次の指令シーケンスを実行すると、同期位置 N120 / N220 の後に移動動作が同時に開始されます。

表 6-3 チャンネル 1 と 2 のシーケンス(時間)

チャンネル 1	チャンネル 2	説明
...	...	チャンネル 1 と 2 で任意の処理
N100 W100		チャンネル 1:移動軸 W
...	...	チャンネル 1 と 2 で任意の処理
RELEASE (W)		チャンネル 1:軸 W の有効化⇒チャンネル 2 での軸の入れ替えはチャンネル 1 に影響しなくなります

チャンネル1	チャンネル2	説明
N110 WAITM(2,1,2)	N210 WAITM(2,1,2)	チャンネル1 および 2:同期を待機
	N215 GET(W)	チャンネル2:軸Wを要求します。軸Wがすでに有効になっているので、軸の入れ替えに最適なタイミングです。
N120 WAITM(9,1,2)	N220 WAITM(9,1,2)	チャンネル1 および 2:同期を待機
N130 Z1=100	N230 Z2=100 W100	チャンネル1 および 2:移動動作の同時開始

### 6.3 必要条件

## K10:チャンネル間の軸入れ替え

### 7.1 一覧

---

#### 注記

#### 主軸

軸の「軸入れ替え」機能に関連する以下のステートメントおよび機能は、**主軸**にも適用されます。

---

コントローラのセットアップ中に、それぞれの軸をチャンネルに割り当ててください。軸は、たとえば軸が割り当てられたチャンネルのパートプログラムやシンクロナイズドアクションのみで移動できます。「軸入れ替え」機能を使って、軸を有効化して、これを別のチャンネルに割り当てる、つまり軸を置き換えることができます。これが、他のチャンネルから軸を移動できる唯一の方法です。

#### 軸の状態

「軸入れ替え」機能の一部として、軸は以下の状態を持つことができます。

- 「チャンネル軸」  
チャンネル軸は、チャンネルに割り当てられた軸です。これは、パートプログラムまたは手動で移動できます。
- 「PLC 軸」  
PLC 軸は、PLC に割り当てられた軸です。これは、PLC ユーザープログラムまたはファンクションブロック FC18 のみで移動できます。
- 「中立軸」  
中立軸は、現在チャンネルまたは PLC に割り当てられていない軸です。移動する前に、まずチャンネルまたは PLC の要求を受ける必要があります。
- 「別チャンネル軸」  
チャンネルが軸を要求すると、軸がこの状態になります。ただし、別のチャンネルに占められているため、割り当てはできていません。

## 7.2 セットアップ

### パラメータの割り付け

#### NC 別マシンデータ

- 軸入れ替え動作の一般パラメータ設定  
MD10722 \$MN\_AXCHANGE\_MASK

#### チャンネルマシンデータ

- どの軸がチャンネルに属するか、またはチャンネル軸かのパラメータ設定  
MD20070 \$MC\_AXCONF\_MACHAX\_USED[<チャンネル軸>] = <機械軸>  
注：「軸入れ替え」機能にかかわらず、どのような場合でも、NC で使用されるすべての軸を1つまたは複数のチャンネルのチャンネル軸として割り付けてください。

#### 軸マシンデータ

- 1つの軸が複数のチャンネルのチャンネル軸である場合、コントローラの起動(電源投入リセット)後にその軸がどのチャンネルに割り当てられるかを以下のマシンデータで定義します。  
MD30550 \$MA\_AXCONF\_ASSIGN\_MASTER\_CHAN[<チャンネル>] = <チャンネル番号>  
パラメータ設定されたチャンネルが、軸のマスタチャンネルです。
- 軸がパートプログラムでプログラムされ、軸が現在チャンネルに割り当てられていない場合の動作のパラメータ設定:
  - アラームを表示し、軸を移動しません。
  - GET (ページ 458)命令で自動的に軸に要求します。
  - GETD (ページ 458)命令で自動的に軸に要求します。
 MD30552 \$MA\_AUTO\_GET\_TYPE[<軸>] = <動作>

### システム変数

#### 軸別のシステム変数

- 軸入れ替えに関連する軸のタイプ:  
\$AA\_AXCHANGE\_TYP[<軸>]
- 軸の軸入れ替え状態:  
\$AA\_AXCHANGE\_STAT[<軸>]

## 7.3 プログラミング軸の解放(RELEASE)

### 機能

現在のチャンネルに割り当てられた軸は、予約された RELEASE () 手順による軸の入れ替えで解除され、このため「中立軸」状態になります。

### 構文

RELEASE (<軸 1>[, 軸 2 ... 軸 15])

### 意味

RELEASE:	軸の入れ替えのために軸を解除	
	先読み停止:	あり
	単独ブロック 指令:	あり
<軸>:	軸:解除する軸のチャンネル軸名称 主軸:解除する主軸またはチャンネル軸名称、または SPI (<主軸番号>) によるチャンネル軸名称の主軸番号の変換。	
	タイプ:	AXIS

### 必要条件

#### イネーブル不可

- 軸が変換に関係している。
- 軸が軸グループに属している。

#### ガントリグループのガイド軸

ガントリグループのガイド軸が解放されると、すべての同期軸も解放されます。

## 7.4 プログラミング軸の確保(GET、GETD)

### 軸の入れ替え要求(GET):

#### 機能

予約された GET () 手順を使って、同じチャンネルに軸を要求します。

RELEASE () を使った軸の入れ替えのために、軸が現在割り当てられているそのチャンネルで、パートプログラムまたはシンクロナイズドアクションで軸を解除する必要があります。

軸の入れ替え後、軸は「チャンネル軸」の状態になります。

#### 構文

GET (<軸 1>[, 軸 2 ... 軸 15])

#### 意味

GET:	現在のチャンネルに軸を要求	
	先読み停止:	あり
	単独ブロック 指令:	あり
<軸>:	軸:要求された軸のチャンネル軸名称 主軸:要求された主軸またはチャンネル軸名称、または SPI (<主軸番号>) によるチャンネル軸名称の主軸番号の変換。	
	タイプ:	AXIS

#### 必要条件

以下の状況で軸の入れ替えが遅延されること。

- 軸が割り当てられたチャンネルによって **RELEASE** で解放されていないこと。
- 検出器の変更が完了していないこと。
- コントローライネーブルの状態の変更が完了していないこと(フォローアップ/ストップの規則への移行およびその逆)。
- NC/PLC インタフェース信号「軸または主軸の無効化」が未処置であること(DB31、... DBX1.3 == 1)。
- 軸の現在の移動動作(補間)が完了していないこと。

## 軸の直接確保(GETD)

### 機能

パートプログラムの後続の加工セクションには、現在のチャンネルに割り当てられていない軸が必要です。予約手順 GETD () を使って、軸が割り当てられるチャンネルから直接確保されます。このために、RELEASE () を使って、このチャンネルで軸が解放される必要はありません。

軸の入れ替え後、軸は「チャンネル軸」の状態になります。

解放するチャンネルの軸の状態に応じて、このチャンネルで先読み停止がトリガされます (STOPRE)。

- 「チャンネル軸」状態 ⇒ 先読み停止
- 「中立軸」状態 ⇒ 先読み停止なし

GETD () を使ってチャンネル間で軸を座標転送するために、要求するチャンネルと解放するチャンネル間でチャンネル同期 (ページ 431) を使用することを推奨します。

<b>通知</b>
<b>解放するチャンネルの先読み停止</b> 解放するチャンネルの軸が「チャンネル軸」状態である場合、このチャンネルで先読み停止がトリガされます (STOPRE)。

### 構文

GETD (<軸>)

### 意味

GETD:	現在のチャンネルで軸を直接確保する
	先読み停止: あり
	単独ブロック 指令: あり
<軸>:	軸:要求された軸のチャンネル軸名称 主軸:要求された主軸またはチャンネル軸名称、または SPI (<主軸番号>) によるチャンネル軸名称の主軸番号の変換。

### 必要条件

軸が解放するチャンネルで「PLC 軸」状態である場合、軸の入れ替えのため PLC ユーザープログラムを使って軸を解除してください。

7.5 自動軸入れ替え

必要条件

チャンネルリセット

- 軸を要求したチャンネルでチャンネルリセットがトリガされた場合、軸の入れ替えはキャンセルされます。
- チャンネルリセット後も、入れ替え済みの軸は最後にそれを要求したチャンネルに割り当てられたままになります。

## 7.5 自動軸入れ替え

機能

軸がパートプログラムまたはシンクロナイズドアクションでプログラムされている場合、自動の軸入れ替えまたは現在のチャンネルでの軸の自動確保は実行されますが、この時点ではチャンネルに割り当てられません。

必要条件

GET () または GETD () で自動の軸入れ替えがパラメータ設定されていること。

MD30552 \$MA\_AUTO\_GET\_TYPE (ページ 456)

必要条件

GETD () の説明 (ページ 458)を参照してください。

例

例 1

プログラムコード	コメント
N1 M3 S1000	; 主軸を移動
N2 RELEASE (SPI (1))	; 中立状態に解除
N3 S3000	; 主軸をプログラム => 自動確保
	; MD30552 \$MA_AUTO_GET_TYPE に対応する動作:
	; 0 => アラーム「軸のタイプが不正」
	; 1 => 自動的な GET (SPI (1))
	; 2 => 自動的な GETD (SPI (1))

## 例 2

プログラムコード	コメント
	; 機械軸 AX1 □ チャンネル軸 X
N1 RELEASE (AX1)	; 中立状態に解除
N2 G04 F2	; 休止時間
N3 G0 X100 Y100	; 軸 X を軌跡軸としてプログラム
	; MD30552 \$MA_AUTO_GET_TYPE に対応する動作:
	; 0 => アラーム「軸のタイプが不正」
	; 1 => 自動的な GET (AX1)
	; 2 => 自動的な GETD (AX1)

## 例 3

プログラムコード	コメント
	; 機械軸 AX1 □ チャンネル軸 X
N1 RELEASE (AX1)	; 中立状態に解除
N2 G04 F2	; 休止時間
N3 POS (X) = 100	; 軸 X を位置決め軸としてプログラム
	; MD30552 \$MA_AUTO_GET_TYPE に対応する動作:
	; 0 => アラーム「軸のタイプが不正」
	; 1 => 自動的な GET (AX1) *)
	; 2 => 自動的な GETD (AX1) *)

\*) 軸の同期が確立されていない場合、GET () または GETD () を使って軸の自動確保のための別のブロックが生成されます。

## 7.6 PLC による軸入れ替え

### 機能

軸入れ替えは、PLC ユーザープログラムを使って NC/PLC インタフェース経由で要求できます。

- NC チャンネルから PLC へ
- PLC から NC チャンネルへ
- NC チャンネルから別の NC チャンネルへ

7.6 PLC による軸入れ替え

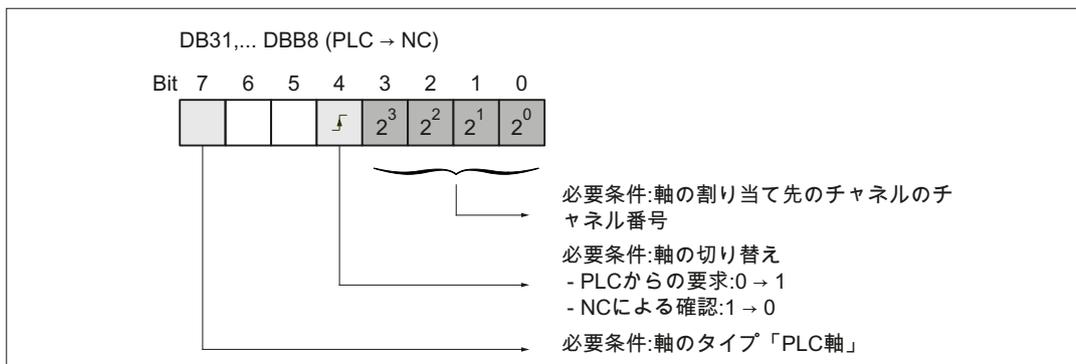


図 7-1 軸入れ替え要求:DB31、... DBB8 (PLC → NC)

軸入れ替え状態

軸入れ替えに関連した軸の現在の状態は、PLC ユーザープログラムを使って NC/PLC インタフェース経由で読み取れます。

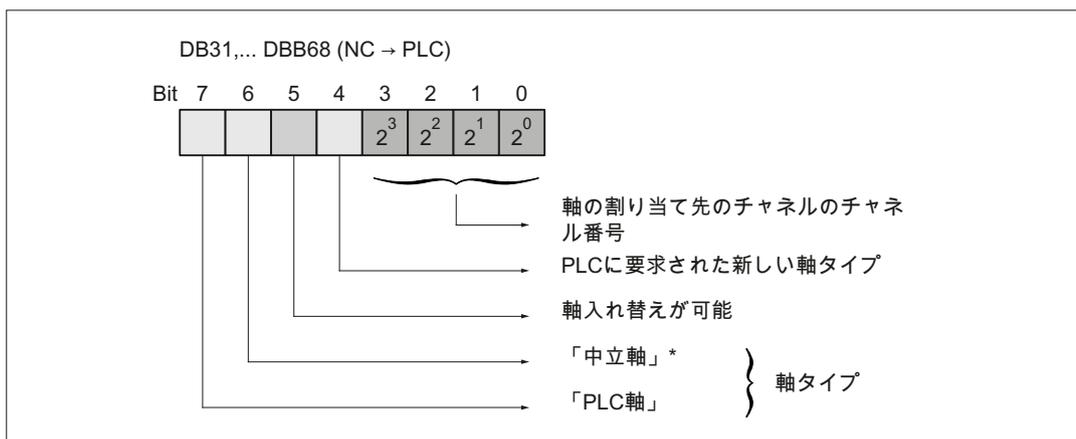


図 7-2 軸入れ替え状態:DB31、... DBB68 (NC → PLC)

例

例 1

各チャンネルで実行されたパートプログラムで RELEASE () および GET () を使った軸のチャンネル 1 からチャンネル 2 への軸入れ替え:

- チャンネル 1:RELEASE (<軸>)
- チャンネル 2:GET (<軸>)

	DB31,... DBB68	DB31,... DBB8	
電源投入後: K1の「チャンネル軸」	0 0 1 0 0 0 0 1	0 0 0 0 0 0 0 0	↓ 調整
K1: RELEASE(AX1)	0 1 1 0 0 0 0 1	0 0 0 0 0 0 0 0	
	0 1 0 0 0 0 0 1	0 0 0 0 0 0 0 0	
K2: GET(AX1)	0 0 1 0 0 0 1 0	0 0 0 0 0 0 0 0	
	0 0 0 0 0 0 1 0	0 0 0 0 0 0 0 0	
	0 0 0 0 0 0 1 0	0 0 0 0 0 0 0 0	

## 例 2

PLC ユーザープログラムによるチャンネルに割り当てられた軸の「NC 軸」から「PLC 軸」への状態の変更。

	DB31,... DBB68	DB31,... DBB8		
電源投入後: K1の「チャンネル軸」	0 0 1 0 0 0 0 1	0 0 0 0 0 0 0 0	↓ 調整	
	0 0 1 0 0 0 0 1	1 0 0 0 0 0 0 0		PLCが要求: 新規のTYP「PLC軸」
要求を検出 軸入れ替え可能	0 0 1 1 0 0 0 1	1 1 0 0 0 0 0 0		NCリセットによる要求
	0 0 1 1 0 0 0 1	1 0 0 0 0 0 0 0		
K1の「PLC軸」	1 0 0 0 0 0 0 1	1 0 0 0 0 0 0 0		
	1 0 0 0 0 0 0 1	1 0 0 0 0 0 0 0		

## 例 3

PLC ユーザープログラムによるチャンネルに割り当てられた軸の「NC 軸」から「PLC 軸」を経て「中立軸」への状態の変更。

7.7 軸コンテナ回転による軸入れ替え

	DB31,... DBB68	DB31,... DBB8	
電源投入後: K1の「チャンネル軸」	0 0 0 0 0 0 0 0 1	0 0 0 0 0 0 0 0	↓ 問題
	0 0 0 0 0 0 0 0 1	1 0 0 0 0 0 0 0	
要求を検出 軸入れ替え可能	0 0 1 1 0 0 0 0 1	1 1 0 0 0 0 0 0	
	0 0 1 1 0 0 0 0 1	1 0 0 0 0 0 0 0	NCリセットによる要求
K1の「PLC軸」	1 0 0 0 0 0 0 0 1	1 0 0 0 0 0 0 0	
	1 0 0 0 0 0 0 0 1	0 1 0 0 0 0 0 0	PLCが要求: 新規のTYP「中立軸」
要求を検出 軸入れ替え可能	1 0 1 1 0 0 0 0 1	0 1 0 0 0 0 0 0	
「中立軸」、 K1	0 1 0 0 0 0 0 0 1	0 0 0 0 0 0 0 0	NCリセットによる要求

## 7.7 軸コンテナ回転による軸入れ替え

### 軸コンテナの回転の有効化

軸コンテナの回転が有効である場合、チャンネルに割り当てることができるすべてのコンテナ軸は、自動生成された GET または GETD を使用してこのチャンネルに割り当てられます。軸は、コンテナ回転の後、たとえば別のチャンネルに受け渡すことだけができます。

#### 注記

軸が「メインラン軸」状態である場合(例: PLC 軸である場合など)、チャンネルへの軸の自動的な割り当ては**できません**。軸コンテナの回転に含めるには、最初に軸がこの状態を終了する必要があります。

コンテナ軸の軸入れ替えの詳細については、「B3：分散システム - 840D sl のみ (ページ 83)」の章を参照してください。

## 例: 自動の GET または GETD で軸コンテナ回転

チャンネル 1 動作

AXCTSWE (CT 1)

チャンネル 2 動作

SPOS = 180 へ位置決め完了

;チャンネル 1 の主軸を取得

;さらに、軸コンテナ回転を許可

## 前提条件:

主軸が両方のチャンネルで使用されており、さらに軸コンテナ CT 1 の軸です。

## 適用

軸コンテナ回転と自動 GET/GETD を使用した軸入れ替えは、マシンデータ MD10722 \$MN\_AXCHANGE\_MASK、ビット 1=1 を使用して起動されます。

## 7.8 先読み停止処理のある軸入れ替えと先読み停止処理のない軸入れ替え

### 先読み停止処理のない軸入れ替え拡張機能

先読み停止処理のある GET ブロックの代わりに、中間ブロックのみが、この GET 要求により生成されます。実行プログラムでこのブロックが実行されると、システムにより、このブロックの各軸のステータスが現在の軸ステータスに一致するかどうかをチェックします。一致しない場合は、強制的に再解析が起動されることがあります。

次の軸または位置決めされた主軸のステータスがチェックされます。

- 軸または位置決めされた主軸のモード
- 指令位置

速度モードの主軸について次のステータスがチェックされます。

- 主軸モード: 速度モード
- 主軸速度 S
- 回転の方向 M3、M4
- ギア選択 M40、M41、M42、M43、M44、M45
- メイン主軸が周速一定制御状態

7.8 先読み停止処理のある軸入れ替えと先読み停止処理のない軸入れ替え

場合によっては、強制的に再解析が行われるかもしれません。追従軸の再解析は、いずれの場合も強制的に行われます。

適用

先読み処理のない入れ替えおよび現在のステータスのチェックは、マシンデータ MD10722 \$MN\_AXCHANGE\_MASK、ビット 2=1 で有効です。

例

先読み停止処理のない軸入れ替えの有効化

表 7-1

```
N010 M4 S1000
N011 G4 F2
N020 M5
N021 SPOS=0
N022 POS[B]=1
N023 WAITP (B) ; 軸 b が中立軸になります
N030 X1 F10
N031 X100 F500
N032 X200
N040 M3 S500
N041 G4 F2
N050 M5
N099 M30
```

主軸(軸 B)がブロック N023 の直後に、PLC 軸として 180°に移動され、その後 1°に戻され、再び中立軸に戻された場合は、ブロック N040 では、先読み停止処理と再解析のいずれも起動されません。

特別な事例: 先読み停止処理のある軸入れ替え

実行プログラムで事前に GET、GETD のどちらの命令も使用せずに、主軸または軸を、RELEASE (軸)または WAITP (軸)などで再び使用可能にすることができます。その後に GET を使用すると、先読み停止を伴う GET になります。

## 7.9 PLC のみの制御軸

### 機能

コントローラの起動後、軸は「中立軸」状態です。PLC がそれを制御します。軸を競合位置決め軸として移動するには (PLC からファンクションブロック FC18 により)、最初にその軸を PLC から指定して要求してください。

---

#### 注記

次のマシンデータによって、PLC による軸入れ替えを PLC 制御軸のみに制限できます。MD10722 \$MN\_AXCHANGE\_MASK、ビット 3 = 1

---

軸は NC パートプログラムから移動できません。

### パラメータ設定

軸を PLC のみの制御軸としてパラメータ設定するには、次の軸マシンデータを使用します。

MD30460 \$MA\_BASE\_FUNCTION\_MASK、ビット 4 = 1

### PLC による制御

PLC のみの制御軸の移動動作は、軸の NC/PLC インタフェース信号にのみ影響されません。

- DB31、... DBX28.1 (リセット)
- DB31、... DBX28.2 (継続)
- DB31、... DBX28.6 (減速カーブに沿って停止)

### 使用可能な移動機能

次の移動機能は、PLC のみの制御軸に対して使用できます。

1. 移動キーとハンドルを使用した JOG モードでの移動
2. 軸の原点確立
3. 内部的なシンクロナイズドアクションによる コマンド軸としての移動
4. 非同期の揺動軸としての移動
5. FC18 による、PLC から競合位置決め軸としての移動

## 7.10 PLC に完全に固定された軸

移動機能 1 ～ 4 の完了後、軸は自動的に「中立軸」状態に戻ります。PLC からの移動機能 5 の完了後、軸は「PLC 軸」状態のままです。軸は、PLC によって指定して解放された後にのみ「中立軸」状態に変更されます。

## 7.10 PLC に完全に固定された軸

### 機能

コントローラの起動後、軸は「中立軸」状態になり、NC チャンネルから制御されます。軸を競合位置決め軸として移動するには(PLC からファンクションブロック FC18 により)、軸を PLC から指定して要求する**必要はありません**。PLC による軸入れ替えは、FC18 により移動要求を使用して自動的におこなわれます。FC18 により要求された移動動作の完了後、軸は再度「中立軸」状態に自動的に変更されます。

軸の入れ替えを完了して、PLC から要求後に、軸を PLC から制御できるようになります。「PLC 軸」状態

### 注記

次のマシンデータによって、PLC による軸入れ替えを PLC に完全に固定された軸のみに制限できます。MD10722 \$MN\_AXCHANGE\_MASK、ビット 3 = 1

### パラメータ設定

軸を PLC に完全に固定された軸としてパラメータ設定するには、次の軸マシンデータを使用します。

MD30460 \$MA\_BASE\_FUNCTION\_MASK、ビット 5 = 1

### PLC または NC チャンネルによる制御

PLC に完全に固定された軸の移動動作は、NC チャンネルまたは PLC のいずれかの影響を受ける可能性があります。

NC チャンネル: チャンネル別 NC/PLC インタフェース信号(選択)

- DB21、... DBXDBX7.1 (NC スタート)
- DB21、... DBXDBX7.3 (NC ストップ)
- DB21、... DBXDBX7.7 (リセット)

PLC: 軸の NC/PLC インタフェース信号

- DB31、... DBX28.1 (リセット)
- DB31、... DBX28.2 (継続)
- DB31、... DBX28.6 (減速カーブに沿って停止)

### 使用可能な移動機能

次の移動機能は、PLC に完全に固定された軸に使用できます

1. 移動キーとハンドルを使用した JOG モードでの移動
2. 軸の原点確立
3. FC18 による、PLC から競合位置決め軸としての移動

移動機能 1～3 の完了後、軸は自動的に「中立軸」状態に戻ります。

## 7.11 回転フレームのジオメトリ軸と軸入れ替え

### 座標回転フレームによる入れ替え拡張

JOG モードでは、静的シンクロナイズドアクションによって、回転 WCS のジオメトリ軸を PLC 軸またはコマンド軸として移動できます。これを行うために、次のマシンデータを設定してください。

MD32074 \$MA\_FRAME\_OR\_CORRPOS\_NOTALLOWED、ビット 10 == 1

---

#### 注記

##### ジョグモードでの運転モードの変更

運転モードをジョグモードから別のモードに変更する前に、回転 WCS のジオメトリ軸としてリンクされているすべての PLC 軸とコマンド軸のすべての移動動作が終了している必要があります。これらの軸は、少なくとも中立軸に戻してください。そうでない場合は、運転モードを変更すると、アラーム 16908 が出力されます。このアラームは、回転座標系で、単独のジオメトリ軸だけが PLC 軸またはコマンド軸として移動されたときにも表示されます。

このような軸は、このチャンネル内の PLC 軸またはコマンド軸にのみ、なることができます。別のチャンネルでの軸入れ替えは許可されません。

---

**JOG モードから AUTOMATIC モードへの変更の必要条件**

「プログラム中断」状態でジョグモードから自動モードに変更すると、次のマシンデータが設定されている場合にのみ、ジオメトリ軸の終点が受け入れられます。

MD 32074:FRAME\_OR\_CORRPOS\_NOTALLOWED、ビット 11 == 1

これにより、PLC 軸またはコマンド軸はフレームの回転を考慮して位置決めされます。

回転フレームの作用を受ける**すべての軸**は、ジオメトリ軸グループとみなされ、まとめて処理されます。このようにして、

- NC プログラムに割り当てられる。
- すべての軸が中立軸となる。
- すべての軸が、実行プログラムの軸(PLC 軸、コマンド軸、または揺動軸)として有効になる。

たとえば、ある軸が WAITP でプログラム指令されており、ジオメトリ軸グループの他のすべての軸を待機し、これらの軸がまとめて中立軸となることができるとします。これらの軸の 1 つが実行プログラムで PLC 軸となると、このグループの他のすべての軸が中立軸となります。

**補足条件**

MD32074 \$MA\_FRAME\_OR\_CORRPOS\_NOTALLOWED ビット 10 == 0 および NC プログラムで ROT Z45 がプログラム指令されている場合は、**X 軸と Y 軸で軸入れ替えをおこなうことはできません**。これはまた、このタイプのプログラミングによりブロックが中断された場合に、たとえば ROT X45 または ROT Y45 などの Z 軸に対して、および JOG 運転モードでも同様に有効です。この場合は、NC/PLC インタフェース信号「軸入れ替え可能」(DB31、...DBX68.5)が X 軸と Y 軸に対して設定されていても、それがリセットされます。

MD32074 \$MA\_FRAME\_OR\_CORRPOS\_NOTALLOWED、ビット 10 == 1 であり、このプログラミングを含むブロックが現在移動中でない場合にのみ、JOG モードでこれらのタイプの軸を入れ替えることができます。

## 7.12 シンクロナイズドアクションを介した軸入れ替え

### 機能

シンクロナイズドアクションの操作として、軸をチャンネル用に要求(GET (<軸>))するか、軸入れ替えのために解放(RELEASE (<軸>))することができます。

### 注記

軸は、マシンデータによって、チャンネル軸として割り当ててください。

軸は、シンクロナイズドアクションによって、またはパートプログラムで、チャンネルから特定のチャンネルに指令 AXTOCHAN で直接転送できます。この軸は同じチャンネル内にある必要はなく、このチャンネルが、その軸が現在、補間権限を持っている必要もありません。

### 軸の現在のステータスと補間権限

どの軸タイプと補間権限で軸入れ替えを実行できるかは、次のシステム変数から読み取ることができます。

<値> = \$AA\_AXCHANGE\_TYP[<軸>]

<値>	意味
0	軸は NC プログラムに割り当てられます。
1	軸は PLC に割り当てられているか、コマンド軸または揺動軸として有効です。
2	別のチャンネルが補間権限を持っています。
3	軸は中立軸です。
4	中立軸は PLC で制御されます。
5	別のチャンネルが補間権限を持ち、軸は NC プログラムのために要求されます。
6	別のチャンネルが補間権限を持ち、軸は中立軸として要求されます。
7	軸は PLC 軸であるか、コマンド軸/揺動軸として有効です。この軸は NC プログラム用に要求されます。
8	軸は PLC 軸であるか、コマンド軸/揺動軸として有効です。この軸は中立軸として要求されます。
9	中立軸状態の固定割り当て PLC 軸です。
10	PLC で制御された中立軸状態の固定割り当て PLC 軸です。

7.12 シンクロナイズドアクションを介した軸入れ替え

「中立軸状態の固定割り当て PLC 軸です」(9)と「PLC で制御された中立軸状態の固定割り当て PLC 軸です」(10)は、PLC に固定で割り当てられた PLC に固定値として割り当てられた GET と RELEASE には依存していません。軸入れ替えも可能かどうかは、システム変数\$AA\_AXCHANGE\_STAT[軸]に表示されます。

シンクロナイズドアクションからの状態遷移 GET、RELEASE、および GET の完了時

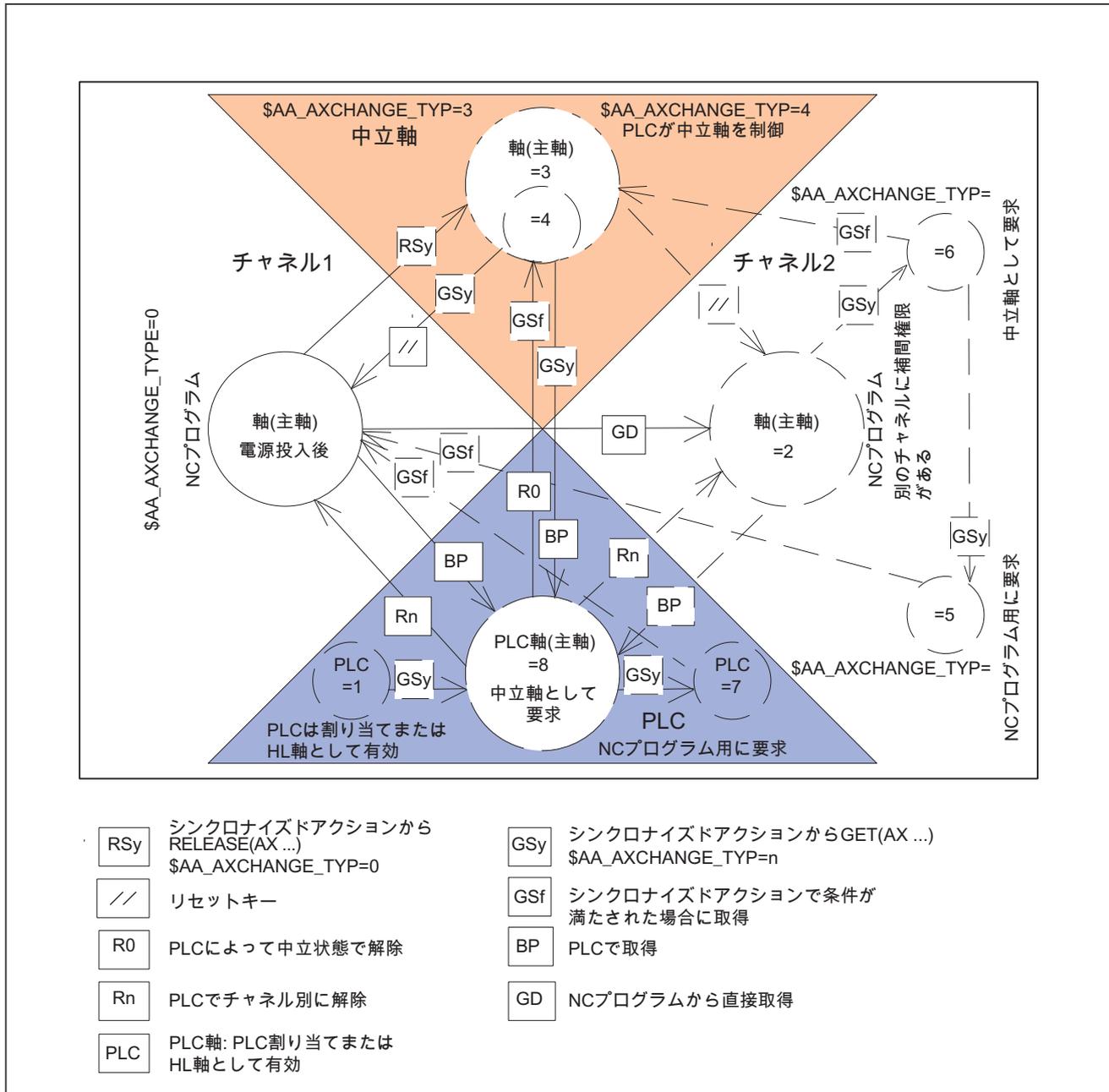


図 7-3 シンクロナイズドアクションからの遷移

### 参照先

『機能マニュアル、シンクロナイズドアクション』::シンクロナイズドアクションの動作の章

## 7.13 マスタ軸(ガントリ)の軸入れ替え

### 機能

閉鎖型ガントリグループは、その軸に関して常に軸入れ替えに関するユニットとして扱われます。このため、マスタ軸の軸入れ替えに対して、ガントリグループのすべての同期軸で同時に軸入れ替えがおこなわれます。前の章で説明したマスタ軸の必要条件に加えて、ガントリグループのすべての同期軸に対して該当する必要条件も満たすようにしてください。

### 軸マシンデータ

軸入れ替えでは、閉鎖型ガントリグループのすべての軸に対して、後述の軸マシンデータに同じ値を設定してください。

- MD30460 \$MA\_BASE\_FUNCTION\_MASK、ビット 4 (制御を実行するコンポーネント)
- MD30460 \$MA\_BASE\_FUNCTION\_MASK、ビット 5 (コンポーネントへの割り当て)

### 軸の NC/PLC インタフェース信号

軸入れ替え機能の範囲内では、後述の軸の NC/PLC インタフェース信号は、閉鎖型ガントリグループのすべての軸で常に同じ値になります。

- DB31、... DBX63.0 (リセット実行)
- DB31、... DBX63.1 (PLC 制御軸)
- DB31、... DBX63.2 (軸停止有効)

## 7.14 状態図

### 軸のシステム変数

軸入れ替え機能の範囲内では、後述の軸のシステム変数は閉鎖型ガントリグループのすべての軸で常に同じ値になります。

- \$AA\_AXCHANGE\_TYP (軸入れ替えに関する軸タイプ)
- \$AA\_AXCHANGE\_STAT (軸入れ替えに関する軸状態)
- \$AA\_SNGLAX\_STAT (個々の軸の軸タイプ)

## 7.14 状態図

下図は、「軸入れ替え」機能に関連したチャンネル 1 の軸の状態、イベント、操作および状態移行を示します。

理解しやすくするため、チャンネル 2 の下位状態「中立軸」および「PLC 軸」は示していません。

前提条件:軸は、初期設定でチャンネル 1 に割り当てられています。

MD30550 \$MA\_AXCONF\_ASSIGN\_MASTER\_CHAN[<軸>] = 1



7.15 境界条件

7.15.1 PLCによる軸入れ替え

PLCに軸入れ替え(PLC→チャンネル、またはチャンネル→PLC)が要求された時点で、チャンネルのパートプログラムが以下のいずれかのセクションにある場合、軸入れ替えは、必ずこの加工セクションが終了した後に実行されます。

- 連続軌跡モード(G64/G640)
- ねじ切り/タッピング (G33/G331/G332)

7.15.2 計算ありのブロック検索

計算ありのブロック検索では、相互にキャンセルにしない命令 GET、GETD、および RELEASE のみがアクションブロックに出力されます。

例

ターゲットブロック N700 の計算ありのブロック検索:

プログラムコード	コメント
...	
N100 RELEASE (AX1)	; RELEASE (AX1)を収集
N110 GET (AX2)	; GET (AX2)を収集
...	
N400 GET (AX1)	; GET (AX1)が RELEASE (AX1)をキャンセル => ; RELEASE (AX1)および GET (AX1)は保存されない
...	
N700 ...	; ターゲットブロック □ 出力:GET (AX2)
N710 RELEASE (AX2)	
...	

参照先

『基本機能』機能マニュアル、「K1:モードグループ、チャンネル、プログラム運転、リセット応答」>「ブロック検索」

## 7.16 例

### 前提条件

- チャンネル 1:以下の軸はチャンネル軸です。1、2、3、4
- チャンネル 2:以下の軸はチャンネル軸です。4、5、6
- 初期設定の割り付け:初期設定で軸 4 (AX4)がチャンネル 1 に割り付けられています。

### パラメータの割り付け

#### チャンネル 1

チャンネルの軸名称:MD20080

- \$MC\_AXCONF\_CHANAX\_NAME\_TAB[<チャンネル 1>][ 0 ] = 「X」 ; 1 番目のチャンネル軸
- \$MC\_AXCONF\_CHANAX\_NAME\_TAB[<チャンネル 1>][ 1 ] = 「Y」 ; 1 番目のチャンネル軸
- \$MC\_AXCONF\_CHANAX\_NAME\_TAB[<チャンネル 1>][ 2 ] = 「Z」 ; 1 番目のチャンネル軸
- \$MC\_AXCONF\_CHANAX\_NAME\_TAB[<チャンネル 1>][ 3 ] = 「U」 ; 1 番目のチャンネル軸

使用する機械軸:MD20070

- \$MC\_AXCONF\_MACHAX\_USED[<チャンネル 1>][ 0 ] = 1 ; 1 番目のチャンネル軸 → 軸 1
- \$MC\_AXCONF\_MACHAX\_USED[<チャンネル 1>][ 1 ] = 2 ; 2 番目のチャンネル軸 → 軸 2
- \$MC\_AXCONF\_MACHAX\_USED[<チャンネル 1>][ 2 ] = 3 ; 3 番目のチャンネル軸 → 軸 3
- \$MC\_AXCONF\_MACHAX\_USED[<チャンネル 1>][ 3 ] = 4 ; 4 番目のチャンネル軸 → 軸 4

#### チャンネル 2

チャンネルの軸名称:MD20080

- \$MC\_AXCONF\_CHANAX\_NAME\_TAB[<チャンネル 2>][ 0 ] = 「X」 ; 1 番目のチャンネル軸
- \$MC\_AXCONF\_CHANAX\_NAME\_TAB[<チャンネル 2>][ 1 ] = 「Y」 ; 1 番目のチャンネル軸
- \$MC\_AXCONF\_CHANAX\_NAME\_TAB[<チャンネル 2>][ 2 ] = 「U」 ; 1 番目のチャンネル軸

7.16 例

使用する機械軸:MD20070

- \$MC\_AXCONF\_MACHAX\_USED[<チャンネル 2>][ 0 ] = 5 ; 1 番目のチャンネル軸 → 軸 5
- \$MC\_AXCONF\_MACHAX\_USED[<チャンネル 2>][ 1 ] = 6 ; 2 番目のチャンネル軸 → 軸 6
- \$MC\_AXCONF\_MACHAX\_USED[<チャンネル 2>][ 2 ] = 4 ; 3 番目のチャンネル軸 → 軸 4

初期設定の割り付け

軸 4 (AX4) のマスタチャンネル → チャンネル 1

4 番目の機械軸の AX4 は、MD10000 \$MN\_AXCONF\_MACHAX\_NAME\_TAB[ 3 ] に基づく標準名称です。

- MD30550 \$MA\_AXCONF\_ASSIGN\_MASTER\_CHAN[ AX4 ] = 1

プログラム例

チャンネル 1 のプログラム	チャンネル 2 のプログラム
...	...
; 移動軸 4 (AX4)	; チャンネル 1 との同期ポイント
G01 F1000 U100	; WAITM(1,1,2)
; AX4 の解放	; AX4 の要求
RELEASE (AX4)	GET (AX4)
; チャンネル 2 でプログラム TAUSH2 を選択:	; 移動軸 4 (AX4)
INIT (2, "_N_MPF_DIR\_N_TAUSH2_MPF", "S")	G0 U0
; チャンネル 2 でプログラム TAUSH2 を起動	...
START (2)	
; チャンネル 2 との同期ポイント	; AX4 の解放
WAITM(1,1,2)	RELEASE (AX4)
...	...
M30	M30

## 7.17 データリスト

### 7.17.1 マシンデータ

#### 7.17.1.1 一般マシンデータ

番号	識別子: \$MN_	説明
10010	ASSIGN_CHAN_TO_MODE_GROUP[n]	モードグループで有効なチャンネル[チャンネル番号]: 0, 1
10722	AXCHANGE_MASK	軸入れ替え動作のパラメータ設定

#### 7.17.1.2 チャンネル別マシンデータ

##### チャンネルの基本マシンデータ

番号	識別子: \$MC_	説明
20000	CHAN_NAME	チャンネル名称
20050	AXCONF_GEOAX_ASSIGN_TAB[n]	ジオメトリ軸のチャンネル軸への割り当て [ジオメトリ軸番号]:0...2
20060	AXCONF_GEOAX_NAME_TAB[n]	チャンネルのジオメトリ軸名称 [ジオメトリ軸番号]:0...2
20070	AXCONF_MACHAX_USED[n]	チャンネルで有効な機械軸番号 [チャンネル軸番号]:0...7
20080	AXCONF_CHANAX_NAME_TAB[n]	チャンネル内の軸名称 [チャンネル軸番号]:0...7
20090	SPIND_DEF_MASTER_SPIND	チャンネル内のメイン主軸の初期設定
20100	DIAMETER_AX_DEF	径方向軸機能が指定されたジオメトリ軸
20110	RESET_MODE_MASK	リセット/TP 終了後の基本制御設定内容の決定
20112	START_MODE_MASK	NC スタートの基本制御設定内容の決定
20150	GCODE_RESET_VALUES[n]	G グループの初期設定 [G グループ番号]:0...59

番号	識別子: \$MC_	説明
20160	CUBIC_SPLINE_BLOCKS	C スプライン用ブロック数
20170	COMPRESS_BLOCK_PATH_LIMIT	圧縮のための NC ブロックの最大移動距離
20200	CHFRND_MAXNUM_DUMMY_BLOCKS	面取り/半径のあるダミーブロック
20210	CUTCOM_CORNER_LIMIT	工具半径補正の補正ブロックの最大角度
20220	CUTCOM_MAX_DISC	DISC の最大値
20230	CUTCOM_CURVE_INSERT_LIMIT	工具径補正の交点演算の最大角度
20240	CUTCOM_MAXNUM_CHECK_BLOCKS	工具径補正の予測輪郭演算のブロック
20250	CUTCOM_MAXNUM_DUMMY_BLOCKS	工具半径補正に対する移動動作なしのブロック数
20270	CUTTING_EDGE_DEFAULT	プログラム指令しないときの工具刃先の初期状態
20400	LOOKAH_USE_VELO_NEXT_BLOCK	プログラム指令された次のブロック速度の先読み
20430	LOOKAH_NUM_OVR_POINTS	先読みのオーバーライドスイッチコーナ値の数
20440	LOOKAH_OVR_POINTS[n]	先読みのオーバーライドスイッチコーナ値 [コーナ値番号]:0...1
20500	CONST_VELO_MIN_TIME	定速移動の最短時間
20600	MAX_PATH_JERK	軌跡関連の最高加々速度
20610	ADD_MOVE_ACCEL_RESERVE	重畳動作に対応した加速度予備
20650	THREAD_START_IS_HARD	ねじ切り時の軸の加速動作
20700	REFP_NC_START_LOCK	レファレンス点なしで NC スタート無効
20750	ALLOW_GO_IN_G96	G96 内の G0 ロジック
20800	SPF_END_TO_VDI	PLC へのサブプログラム終了
21000	CIRCLE_ERROR_CONST	円弧終点監視定数
21010	CIRCLE_ERROR_FACTOR	円弧終点監視係数
21100	ORIENTATION_IS_EULER	旋回プログラミングに対する角度定義
21110	X_AXIS_IN_OLD_X_Z_PLANE	自動フレーム定義用座標系
21200	LIFTFAST_DIST	輪郭からの高速リトラクトの移動距離
21250	START_INDEX_R_PARAM	最初のチャンネル別 R 変数の数

## チャンネルの補助機能セッティング

番号	識別子: \$MC_	説明
22000	AUXFU_ASSIGN_GROUP[n]	補助機能グループ [チャンネルの補助機能番号]:0...49
22010	AUXFU_ASSIGN_TYPE[n]	補助機能のタイプ [チャンネルの補助機能番号]:0...49
22020	AUXFU_ASSIGN_EXTENSION[n]	補助機能拡張 [チャンネルの補助機能番号]:0...49
22030	AUXFU_ASSIGN_VALUE[n]	補助機能値 [チャンネルの補助機能番号]:0...49
22200	AUXFU_M_SYNC_TYPE	M 機能の出力タイミング
22210	AUXFU_S_SYNC_TYPE	S 機能の出力タイミング
22220	AUXFU_T_SYNC_TYPE	T 機能の出力タイミング
22230	AUXFU_H_SYNC_TYPE	H 機能の出力タイミング
22240	AUXFU_F_SYNC_TYPE	F 機能の出力タイミング
22250	AUXFU_D_SYNC_TYPE	D 機能の出力タイミング
22260	AUXFU_E_SYNC_TYPE (すぐ使用可能)	E 機能の出力タイミング
22400	S_VALUES_ACTIVE_AFTER_RESET	リセット後に S 機能有効
22410	F_VALUES_ACTIVE_AFTER_RESET	リセット後に F 機能有効
22500	GCODE_OUTPUT_TO_PLC	PLC への G 命令
22550	TOOL_CHANGE_MODE	M 機能の新しい工具オフセット
22560	TOOL_CHANGE_M_CODE	工具交換の M 機能

## チャンネル別メモリのセッティング

番号	識別子: \$MC_	説明
25000	REORG_LOG_LIMIT	ログファイルを有効にするための IPO バッファの割合 (%)
28000	MM_REORG_LOG_FILE_MEM	REORG のメモリサイズ(DRAM)

## 7.17 データリスト

番号	識別子: \$MC_	説明
28010	MM_NUM_REORG_LUD_MODULES	REORG のローカルユーザ変数のブロック数(DRAM)
28020	MM_NUM_LUD_NAMES_TOTAL	ローカルユーザ変数の数(DRAM)
28030	MM_NUM_LUD_NAMES_PER_PRO G	プログラム毎のローカルユーザ変数の数(DRAM)
28040	MM_LUD_VALUES_MEM	ローカルユーザ変数のメモリサイズ(DRAM)
28050	MM_NUM_R_PARAM	チャンネル別 R 変数の数(SRAM)
28060	MM_IPO_BUFFER_SIZE	IPO バッファの NC ブロックの数(DRAM)
28070	MM_NUM_BLOCKS_IN_PREP	ブロック解析のためのブロック数(DRAM)
28080	MM_NUM_USER_FRAMES	設定可能フレームの数(SRAM)
28090	MM_NUM_CC_BLOCK_ELEMENTS	コンパイルサイクルのブロック要素数(DRAM)
28100	MM_NUM_CC_BLOCK_USER_MEM	コンパイルサイクルのブロックメモリのサイズ(DRAM)
28500	MM_PREP_TASK_STACK_SIZE	解析処理のスタックサイズ(DRAM)
28510	MM_IPO_TASK_STACK_SIZE	IPO 処理のスタックサイズ(DRAM)

## 7.17.1.3 軸/主軸別マシンデータ

番号	識別子: \$MA_	説明
30460	BASE_FUNCTION_MASK	軸機能
30550	AXCONF_ASSIGN_MASTER_CHAN	軸変更のためのチャンネルの初期設定
30552	AUTO_GET_TYPE	自動 GET の定義
30600	FIX_POINT_POS	G75 の軸固定値の位置
32074	FRAME_OR_CORRPOS_NOTALLOWED	フレームも HL オフセットも許可されません
33100	COMPRESS_POS_TOL	補正の最大誤差

## 7.17.2 セッティングデータ

### 7.17.2.1 チャンネル別セッティングデータ

番号	識別子: \$SC_	説明
42000	THREAD_START_ANGLE	ネジ山の開始角度
42100	DRY_RUN_FEED	ドライラン送り速度

## 7.17.3 信号

### 7.17.3.1 チャンネルへの信号

信号名称	SINUMERIK 840D sl	SINUMERIK 828D
プログラムテスト起動	DB21、... .DBX1.7	DB320x.DBX1.7
NC スタート	DB21、... .DBX7.1	DB320x.DBX7.1
NC ストップ	DB21、... .DBX7.3	DB320x.DBX7.3
リセット	DB21、... .DBX7.7	DB330x.DBX3.7
プログラムテストのマスク	DB21、... .DBX14.0	-

### 7.17.3.2 チャンネルからの信号

信号名称	SINUMERIK 840D sl	SINUMERIK 828D
プログラムテスト選択中	DB21、... .DBX25.7	DB170x.DBX1.7
プログラムテスト有効	DB21、... .DBX33.7	DB330x.DBX1.7

### 7.17.3.3 軸/主軸への信号

信号名称	SINUMERIK 840D sl	SINUMERIK 828D
軸または主軸が無効	DB31、... .DBX1.3	DB380x.DBX1.3
ターゲットチャンネルのチャンネル番号	DB31、... .DBX8.0 -3	-
チャンネル割り付けの変更	DB31、... .DBX8.4	-

## 7.17 データリスト

信号名称	SINUMERIK 840D sl	SINUMERIK 828D
PLC 軸/主軸を要求	DB31、... .DBX8.7	-
プログラムテストのマスク	DB31、... .DBX14.0	DB3800.DBX1002.0
リセット	DB31、... .DBX28.1	-
継続	DB31、... .DBX28.2	-
減速カーブに沿って停止	DB31、... .DBX28.6	-

## 7.17.3.4 軸/主軸からの信号

信号名称	SINUMERIK 840D sl	SINUMERIK 828D
リセット実行	DB31、... .DBX63.0	-
PLC 制御軸	DB31、... .DBX63.1	-
軸停止有効	DB31、... .DBX63.2	DB390x.DBX3.2
実際のチャンネルの番号	DB31、... .DBX68.0 -3	-
PLC から要求された新しいタイプ	DB31、... .DBX68.4	-
軸入れ替え可能	DB31、... .DBX68.5	-
中立軸/主軸	DB31、... .DBX68.6	-
PLC 軸/主軸	DB31、... .DBX68.7	-

# M1:キネマティックトランスフォーメーション

## 8.1 TRANSMIT 面端座標変換(オプション)

### 8.1.1 機能

#### 8.1.1.1 はじめに

---

##### 注記

「円筒補間の終了(TRANSMIT)」機能には、ライセンスを受けた「TRANSMIT と円筒補間」オプションが必要です。

---

TRANSMIT 座標変換により、旋盤上の面加工(穴空け、輪郭)が可能になります。

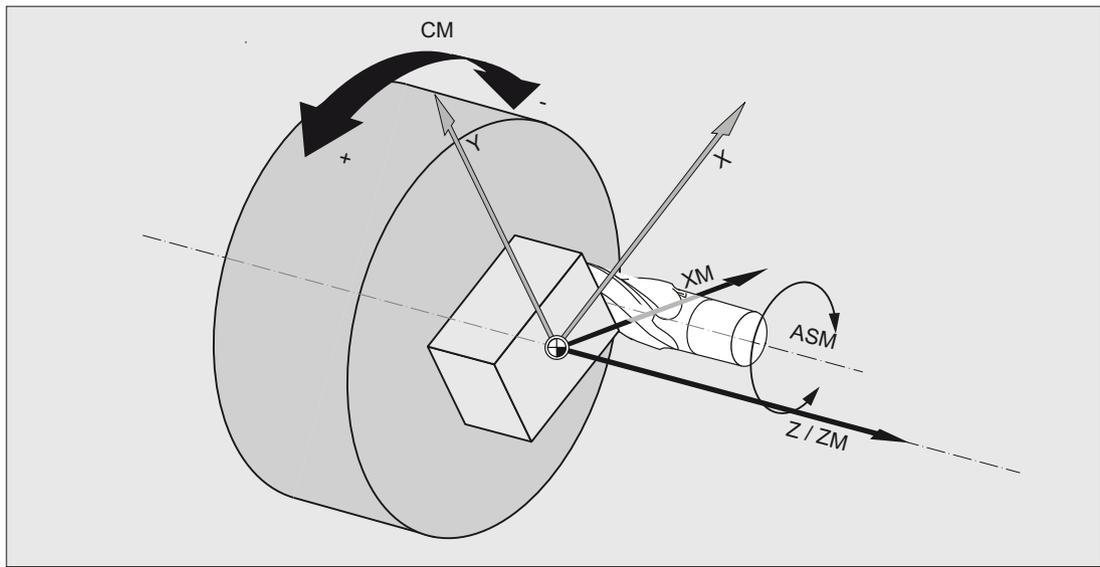
直交座標系を使用して、これらの加工運転のプログラム指令ができます。

コントローラは、プログラム指令した直交座標系の移動動作を、実際の機械軸の移動動作に変換します。

標準ケース:

- 回転軸
- 切り込み軸、回転軸に直交
- 長手軸、回転軸に平行
- これらの直線軸は相互に直交しています。

8.1 TRANSMIT 面端座標変換(オプション)



- X、Y、 ジオメトリ軸
- Z
- CM 機械軸:回転軸
- XM 機械軸:直線軸、回転軸に直交
- ZM 機械軸:直線軸、回転軸に平行
- ASM 機械軸:主軸

他のオプション:

- 回転中心に対して工具中心オフセットを設定できます。
- 工具中心点の軌跡は、回転軸の回転中心点を通過できます。
- 回転軸がモジュロ軸である必要はありません。

注記

動作中の座標変換では、関連する機械、チャンネル、およびジオメトリ軸の名称は異なります。

- MD10000 \$MN\_AXCONF\_MACHAX\_NAME\_TAB (機械軸名称)
- MD20080 \$MC\_AXCONF\_CHANAX\_NAME\_TAB (チャンネル軸名称)
- MD20060 \$MC\_AXCONF\_GEOAX\_NAME\_TAB (ジオメトリ軸名称)

### 8.1.1.2 加工選択

TRANSMIT 座標変換の極は、TRANSMIT 平面の原点にあります。極は、半径方向直線軸と回転軸の交点上にあります。極付近では通常、ジオメトリ軸の位置をわずかに変更しただけで、結果的に機械回転軸の位置が大きく変わります。唯一の例外は、極の内部への、または極を通過する直線移動です。

工具中心点軌跡が極を通過しても、パートプログラムは中止されません。指令可能な移動命令または有効な工具径補正に関しては、制限事項はありません。ただし、極付近のワーク加工操作は推奨しません。これは、回転軸が過負荷にならないよう、急激に送り速度を低下させる場合があるためです。

### 新機能

極は、工具中心点で描かれる線が回転軸の回転中心と交差する場合に存在します。

次の場合について説明します。

- 極を移動できる条件と方法
- 極付近の動作
- 作業領域リミットに関する動作
- 回転量が 360°を超える回転軸の監視

### 極の通過

極は、2つの方法で移動できます。

- 直線軸に沿った移動
- 極回転軸の回転で極へ移動

8.1 TRANSMIT 面端座標変換(オプション)

直線軸に沿った移動

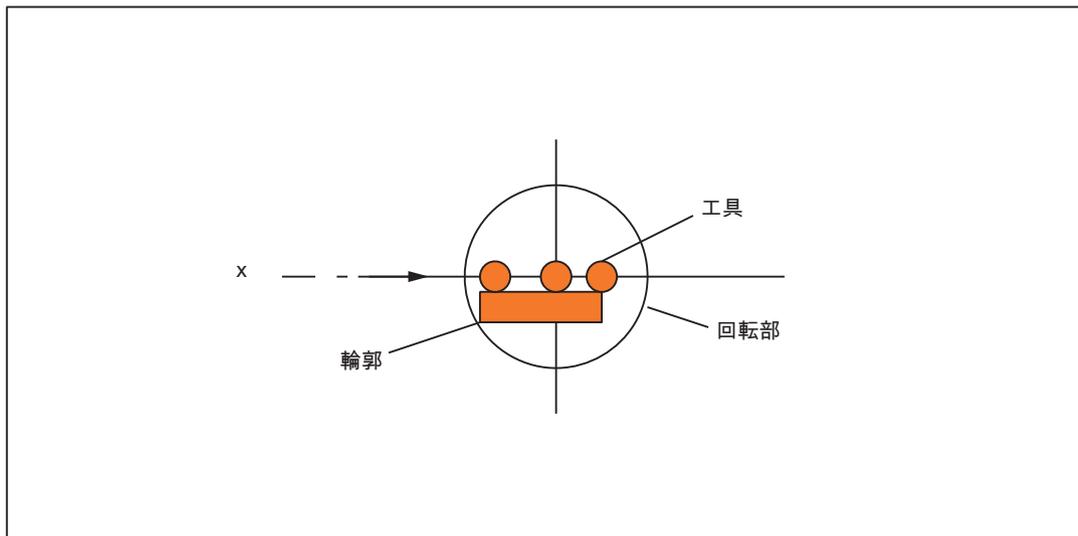


図 8-1 極を通過する x 軸の移動

極の回転

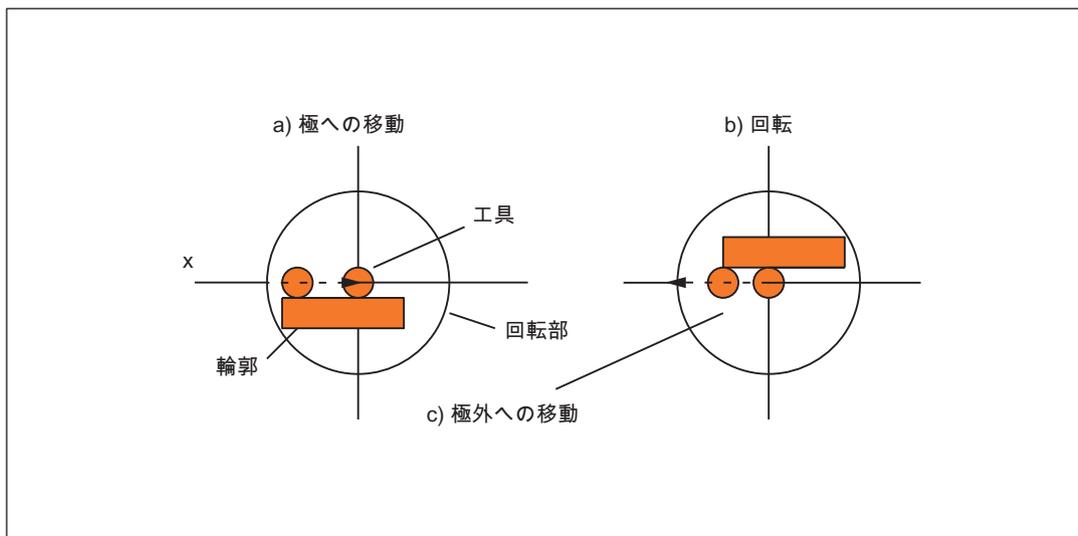


図 8-2 x 軸の極内部への移動(a)、回転(b)、極からの移動(c)

方法の選択

機械の能力および加工される部品の必要条件に従って選択してください。方法は、次のマシンデータで選択されます。

MD24911 \$MC\_TRANSMIT\_POLE\_SIDE\_FIX\_1

## MD24951 \$MC\_TRANSMIT\_POLE\_SIDE\_FIX\_2

1 番目の MD がチャンネルの 1 番目の TRANSMIT 変換に適用され、2 番目の MD が同様に、2 番目の TRANSMIT 変換に適用されます。

値	意味
0	極の通過 工具中心点軌跡(直線軸)は、連続軌跡で極を移動する必要があります。
1	極を中心とした回転。 工具中心点軌跡は、(回転中心の前の)直線軸の正の移動範囲に制限されます。
2	極を中心とした回転。 工具中心点軌跡は、(回転中心の後ろの)直線軸の負の移動範囲に制限されます。

## 極の通過に関する特記事項

直線軸に沿った極の通過方法は、AUTOMATIC と JOG モードに適用される場合があります。

8.1 TRANSMIT 面端座標変換(オプション)

システム動作:

表 8-1 直線軸に沿った極の通過

モード	状態	動作
AUTOMATIC	すべての変換関連軸は、同期して移動します。TRANSMIT が有効な場合	高速度で極を通過
	変換関連軸の中には、同期移動をおこなわない軸があります(位置決め軸など)。TRANSMIT が有効でない	低速度の極の通過
	適用されている DRF (外部ゼロオフセット)は運転を妨げません。極付近では、DRF の用途で使用時にサーボエラーが発生する場合があります。	加工の中止、アラーム
JOG	-	低速度の極の通過

極の回転に関する特記事項

必要条件:この方法は、AUTOMATIC モードでのみ有効です。

MD24911 \$MC\_TRANSMIT\_POLE\_SIDE\_FIX\_1 = 1 または 2

MD24951 \$MC\_TRANSMIT\_POLE\_SIDE\_FIX\_2 = 1 または 2

値:1 つの直線軸が正の移動範囲内に残ります。

値:2 つの直線軸が負の移動範囲内に残ります。

工具中心点軌跡に沿った極の移動が必要な輪郭の場合は、次の 3 つのステップにより、回転中心を超える範囲への直線軸の移動が防止されます。

ステップ	動作
1	直線軸が極内部へ移動します。
2	回転軸が 180°を超えて回転し、もう一方の変換関連軸は停止した状態です。
3	残りのブロックが実行されます。直線軸が、再び極の外へ移動します。

JOG モードでは、移動が極で停止します。このモードでは、極にアプローチするときにとどった接線軌跡に沿ってのみ、軸は極の外に移動できます。他のすべての動作命令では、回転軸位置の切り替えが必要となり、また最小限の移動命令の場合に対しては大きな機械移動が必要となります。このような命令はアラーム 21619 で拒否されます。

## 極付近の通過

工具中心点が極を通過して移動する場合は、コントロールシステムが自動的に送り速度と軌跡加速度を下げて、機械軸(MD 32000 \$MA\_MAX\_AX\_VELO[AX\*]および MD32300 \$MA\_MAX\_AX\_ACCEL[AX\*])の設定値を超えないようにします。軌跡が極に近いほど、送り速度は大幅に低下します。

## 極内部にコーナーのある工具中心点軌跡

極内部にコーナーを持つ工具中心点軌跡は、軸速度の切り替えだけでなく、回転軸位置の切り替えが発生します。これは、減速しても無くすことはできません。

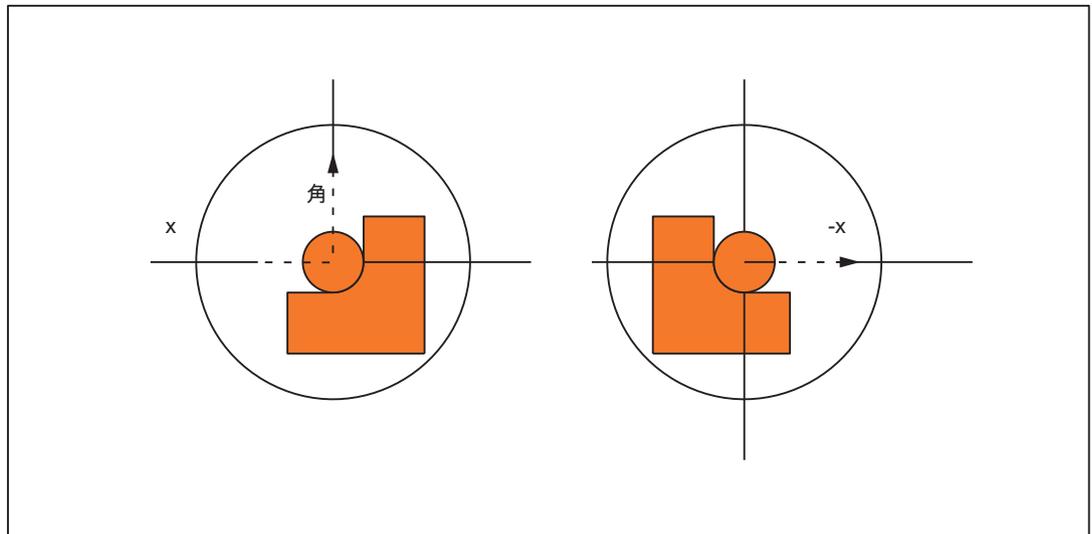


図 8-3 極の通過

必要条件:

AUTOMATIC モード、

MD24911 \$MC\_TRANSMIT\_POLE\_SIDE\_FIX\_1 = 0

または

MD24951 \$MC\_TRANSMIT\_POLE\_SIDE\_FIX\_2 = 0

## 8.1 TRANSMIT 面端座標変換(オプション)

コントロールシステムは、切り替え点で移動ブロックを挿入します。このブロックにより、可能な限り小さい回転が生成され、輪郭の加工を継続できます。

## 極の通過のないコーナー

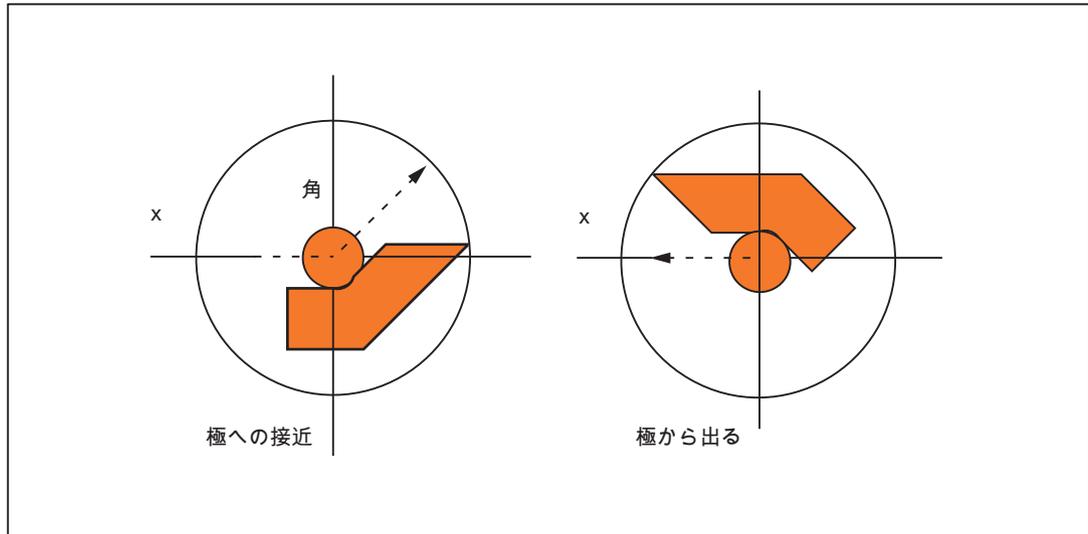


図 8-4 極の片側での加工

必要条件:

AUTOMATIC モード、

MD24911 \$MC\_TRANSMIT\_POLE\_SIDE\_FIX\_1 = 1 または 2

または

MD24951 \$MC\_TRANSMIT\_POLE\_SIDE\_FIX\_2 = 1 または 2

コントロールシステムは、切り替え点で移動ブロックを挿入します。このブロックにより、必要な回転が生成されるため、極の同じ側で輪郭の加工を継続できます。

## 極での変換の選択

加工運転を、動作中の変換の極にある工具中心軌跡の位置から継続する必要がある場合は、新しい変換のために、極からの退出が指定されます。

もし

MD24911 \$MC\_TRANSMIT\_POLE\_SIDE\_FIX\_1 = 0

または

MD24951 \$MC\_TRANSMIT\_POLE\_SIDE\_FIX\_2 = 0

上のいずれかが設定されている場合(極変換)は、極を基点とするブロックの開始時に、可能な限り小さい回転が生成されます。この回転に応じて、軸が回転中心の前または後ろ側に移動します。

MD24911 \$MC\_TRANSMIT\_POLE\_SIDE\_FIX\_1 = 1

または

MD24951 \$MC\_TRANSMIT\_POLE\_SIDE\_FIX\_2 = 1

上記のいずれかの場合は、加工が、回転中心点(直線軸の正の移動範囲)の**前側**で行われます。

MD24911 \$MC\_TRANSMIT\_POLE\_SIDE\_FIX\_1 = 2

または

MD24951 \$MC\_TRANSMIT\_POLE\_SIDE\_FIX\_2 = 2

上記のいずれかの場合は、回転中心点(負の移動範囲の直線軸)の**後ろ側**で行われます。

### 極外での変換の選択

コントロールシステムは、座標変換に関連する軸をマシンデータ MD24911 \$MC\_TRANSMIT\_POLE\_SIDE\_FIX\_<t>を評価せずに移動します。この場合、<t> = 1 はチャンネルの 1 番目、<t> = 2 はチャンネルの 2 番目の TRANSMIT 座標変換を表します。

### 8.1.1.3 ワーキングエリアリミット

#### 初期状態

TRANSMIT は有効ですが、工具中心点の位置が、変換に関連する回転軸の回転中心に位置できない場合は、作業領域リミットによって極が置き換えられます。これは、回転軸に垂直な(工具オフセットのための)軸が、その回転軸と同じ半径方向平面上に位置していない場合、または両方の軸が互いに直角に位置していない場合です。2 軸間の距離により、工具がそこに位置決めできないような、BCS の円筒形の空間が定義されます。

機械軸の移動範囲には影響を及ぼさないため、不適切な範囲はソフトウェアリミットスイッチ監視機能で保護できません。

## 8.1 TRANSMIT 面端座標変換(オプション)

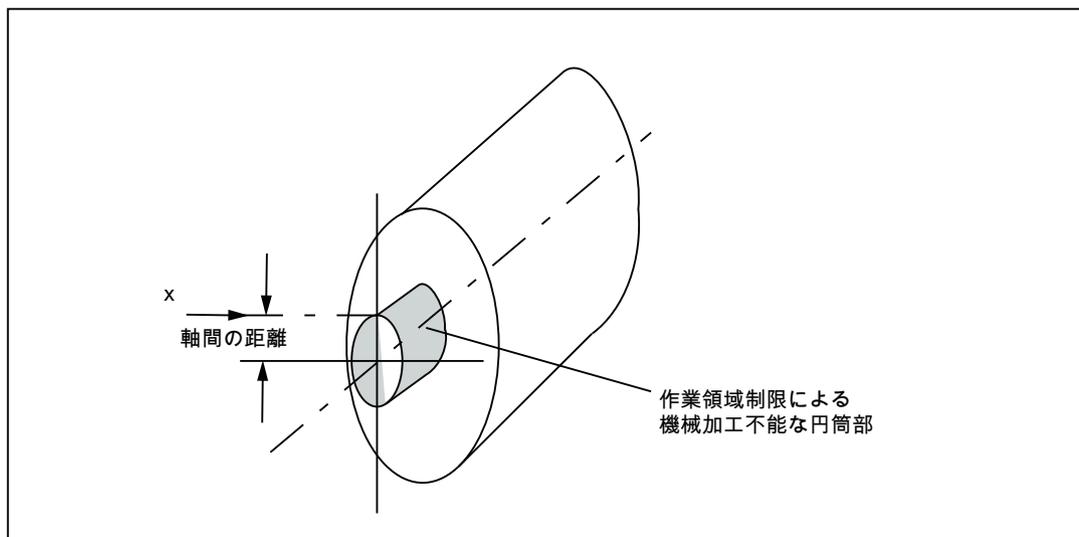


図 8-5 直線軸オフセットに基づいた作業領域リミット

## 作業領域リミットへの移動

作業領域リミットに抵触するあらゆる移動は拒否され、アラーム **21619** が出力されます。対応するパートプログラムブロックは処理されません。コントロールシステムは、その前のブロックで処理を中止します。

移動が事前に予測できず(JOG モード、位置決め軸)、その前のブロックで中止できない場合は、制御を作業領域リミットの端で中止します。

## 作業領域リミット付近の動作

工具中心点が禁止範囲を通過して移動する場合は、コントロールシステムが自動的に送り速度と軌跡加速度を下げて、機械軸(MD 32000 \$MA\_MAX\_AX\_VELO[AX\*]および MD32300 \$MA\_MAX\_AX\_ACCEL[AX\*])の設定値を超えないようにします。軌跡が作業領域リミットに近いほど、送り速度が大幅に低下します。

#### 8.1.1.4 TRANSMIT による重畳動作

コントロールシステムは、重畳動作を予測できません。しかし、この重畳動作が極(または作業領域リミット)からの現在の距離と比較して非常に小さい場合(精密工具オフセットなど)は、これにより機能が妨げられることはありません。座標変換に関連する軸では、座標変換は重畳動作を監視し、臨界量をアラーム 21618 で示します。このアラームは、ブロック単位の速度プランニング機能が、機械の実際の状態と十分に一致しなくなったことを示します。このため、アラームが出力されると、従来の最適化なしのオンライン速度監視が有効になります。先読みルーチンは、制御装置内部で生成された REORG によって、実行プログラムで再同期します。

アラーム 21618 は、軸の過負荷を引き起こし、その結果、パートプログラム処理を中止する可能性をもった状態を示しているため、できる限り発生しないようにしてください。

#### 8.1.1.5 回転量が 360°を超える回転軸の監視

回転軸の位置は、回転数ではわかりません。コントローラは、極を中心とした複数の回転を含むブロックをサブブロックに分割します。

この分割により、最初のサブブロックの終点以外のプログラム指令ブロックの終点は、同期制御とは関連しなくなり、並列運転(補助機能の出力、ブロック同期位置決め軸の移動など)に関しては注意してください。

##### 参照先:

『機能マニュアル、基本機能』; 「H2:PLC に対する補助機能出力」

『機能マニュアル、シンクロナイズドアクション』

シングルブロック運転では、コントロールシステムは、指定によって個々のブロックを加工します。それ以外の場合は、サブブロックが、シングルブロックとまったく同様に先読みをおこないながら移動します。回転軸設定仕様のリミット位置は、ソフトウェアリミットスイッチ監視機能で監視されます。

## 8.1 TRANSMIT 面端座標変換(オプション)

### 8.1.2 パラメータの割り付け

#### 8.1.2.1 一覧

##### マシンデータ:一般の座標変換データ

チャンネルの座標変換データセットを定義するために、次のマシンデータを使用します。

- MD2xxxx \$MC\_TRAFO\_TYPE\_<n> (チャンネルの<n>番目の座標変換の定義)
- MD2xxxx \$MC\_TRAFO\_AXES\_IN\_<n> (チャンネルの<n>番目の座標変換の軸割り当て)
- MD2xxxx \$MC\_TRAFO\_GEOAX\_ASSIGN\_TAB\_<n> (座標変換<n>のためのジオメトリ軸のチャンネル軸への割り当て)
- MD2xxxx \$MC\_TRAFO\_INCLUDES\_TOOL\_<n> (有効な n 番目の座標変換による工具処理)

ここで、<n> = 1、2、3、... 座標変換データセットの最大数

TRANSMIT (タイプ 256 または 257)の場合、1つのチャンネルにパラメータ設定できる座標変換データセットは最大2つです。

- MD2xxxx \$MC\_TRAFO\_TYPE\_<x> = <TRANSMIT タイプ>
- MD2xxxx \$MC\_TRAFO\_TYPE\_<y> = <TRANSMIT タイプ>

##### マシンデータ:TRANSMIT 変換

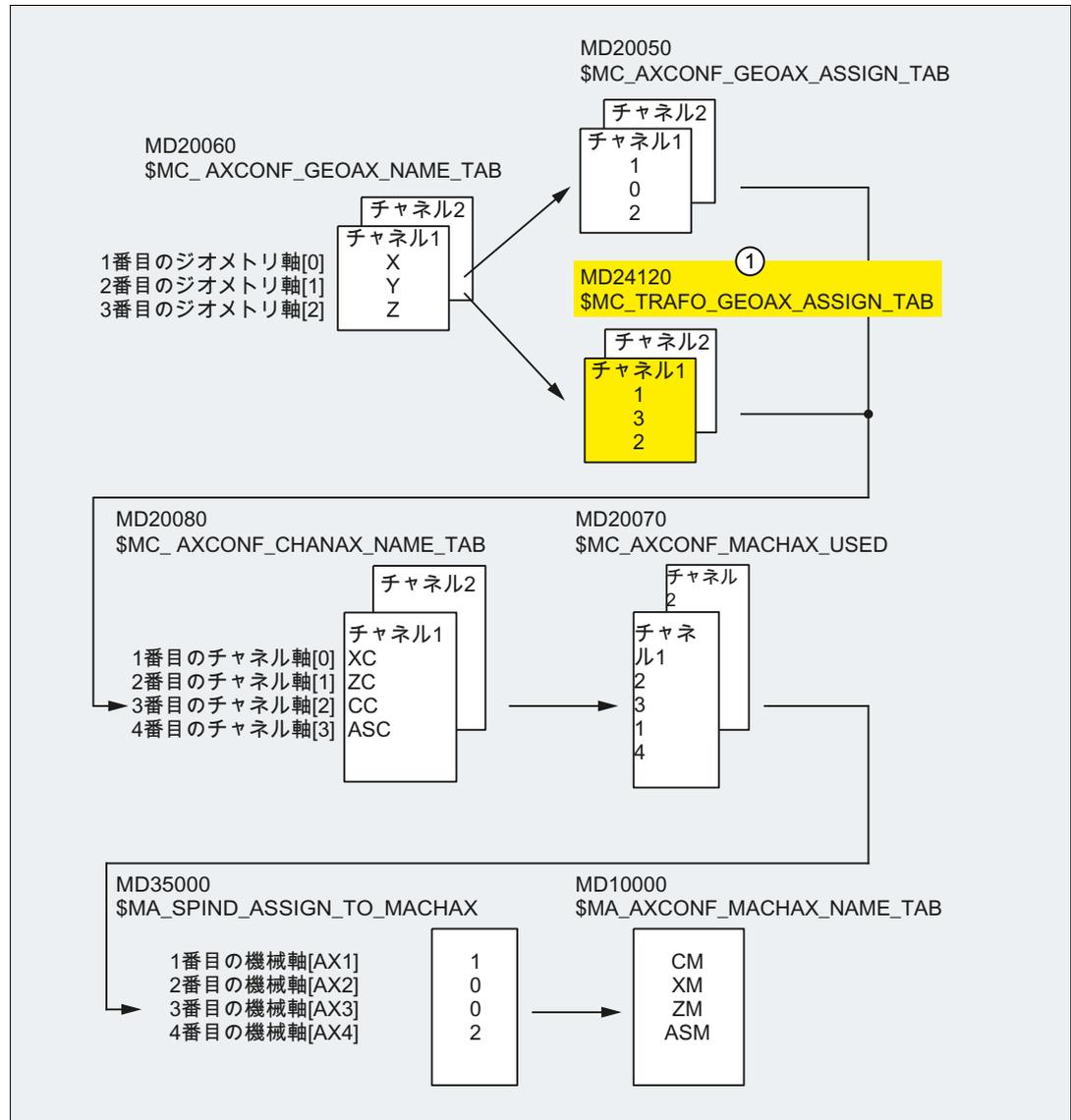
TRANSMIT 座標変換は、次のマシンデータを使用してパラメータ設定します。

- MD2xxxx \$MC\_TRANSMIT\_ROT\_AX\_OFFSET\_<n> (回転軸のオフセット)
- MD2xxxx \$MC\_TRANSMIT\_ROT\_AX\_FRAME\_<n> (回転軸のオフセット)
- MD2xxxx \$MC\_TRANSMIT\_ROT\_SIGN\_IS\_PLUS\_<n> (回転軸の符号)
- MD2xxxx \$MC\_TRANSMIT\_BASE\_TOOL\_<n> (基本工具のベクトル)
- MD2xxxx \$MC\_TRANSMIT\_POLE\_SIDE\_FIX\_<n> (極の前/後の作業領域の制限)

ここで、<n> = 1、2 (TRANSMIT データセット番号)

## 8.1.2.2 軸設定

以下に、TRANSMIT で一般的な軸設定を示します。



① TRANSMIT が有効な場合に有効

## 機械軸名称

- MD10000 \$MN\_AXCONF\_MACHAX\_NAME\_TAB[ 0 ] = 「CM」
- MD10000 \$MN\_AXCONF\_MACHAX\_NAME\_TAB[ 1 ] = 「XM」
- MD10000 \$MN\_AXCONF\_MACHAX\_NAME\_TAB[ 2 ] = 「ZM」
- MD10000 \$MN\_AXCONF\_MACHAX\_NAME\_TAB[ 3 ] = 「ASM」

8.1 TRANSMIT 面端座標変換(オプション)

ジオメトリ軸名称

- MD20060 \$MC\_AXCONF\_GEOAX\_NAME\_TAB[ 0 ] = 「X」 (1 番目のジオメトリ軸の名称)
- MD20060 \$MC\_AXCONF\_GEOAX\_NAME\_TAB[ 1 ] = 「Y」 (2 番目のジオメトリ軸の名称)
- MD20060 \$MC\_AXCONF\_GEOAX\_NAME\_TAB[ 2 ] = 「Z」 (3 番目のジオメトリ軸の名称)

チャンネル軸名称

- MD20080 \$MC\_AXCONF\_CHANAX\_NAME\_TAB[ 0 ] = 「XC」
- MD20080 \$MC\_AXCONF\_CHANAX\_NAME\_TAB[ 1 ] = 「ZC」
- MD20080 \$MC\_AXCONF\_CHANAX\_NAME\_TAB[ 2 ] = 「CC」
- MD20080 \$MC\_AXCONF\_CHANAX\_NAME\_TAB[ 3 ] = 「ASC」

チャンネル軸へのジオメトリ軸の割り当て

TRANSMIT が有効でない

- MD20050 \$MC\_AXCONF\_GEOAX\_ASSIGN\_TAB[ 0 ] = 1 (1 番目のジオメトリ軸 → 1 番目のチャンネル軸 XC)
- MD20050 \$MC\_AXCONF\_GEOAX\_ASSIGN\_TAB[ 1 ] = 0 (-)
- MD20050 \$MC\_AXCONF\_GEOAX\_ASSIGN\_TAB[ 2 ] = 2 (3 番目のジオメトリ軸 → 2 番目のチャンネル軸 ZC)

TRANSMIT が有効な場合

- MD24120 \$MC\_TRAFO\_GEOAX\_ASSIGN\_TAB\_1[ 0 ] = 1 (1 番目の座標変換ジオメトリ軸 → 1 番目のチャンネル軸 XC)
- MD24120 \$MC\_TRAFO\_GEOAX\_ASSIGN\_TAB\_1[ 1 ] = 3 (2 番目の座標変換ジオメトリ軸 → 3 番目のチャンネル軸 CC)
- MD24120 \$MC\_TRAFO\_GEOAX\_ASSIGN\_TAB\_1[ 2 ] = 2 (3 番目の座標変換ジオメトリ軸 → 2 番目のチャンネル軸 ZC)

機械軸へのチャンネル軸の割り当て

- MD20070 \$MC\_AXCONF\_MACHAX\_USED[ 0 ] = 2 (1 番目のチャンネル軸 → 2 番目の機械軸 XM)
- MD20070 \$MC\_AXCONF\_MACHAX\_USED[ 1 ] = 3 (2 番目のチャンネル軸 → 3 番目の機械軸 ZM)

- MD20070 \$MC\_AXCONF\_MACHAX\_USED[ 2 ] = 1 (3 番目のチャンネル軸 → 1 番目の機械軸 CM)
- MD20070 \$MC\_AXCONF\_MACHAX\_USED[ 3 ] = 4 (4 番目のチャンネル軸 → 4 番目の機械軸 ASM)

#### 主軸の定義

- MD35000 \$MA\_SPIND\_ASSIGN\_TO\_MACHAX[ 0 ] = 1 (主軸)
- MD35000 \$MA\_SPIND\_ASSIGN\_TO\_MACHAX[ 1 ] = 0 (軸)
- MD35000 \$MA\_SPIND\_ASSIGN\_TO\_MACHAX[ 2 ] = 0 (軸)
- MD35000 \$MA\_SPIND\_ASSIGN\_TO\_MACHAX[ 3 ] = 2 (主軸)

### 8.1.2.3 個別設定

#### 1つの回転軸と1つの直線軸:TRAFO\_TYPE = 256

1つの回転軸および1つの直線軸を含む TRANSMIT には、座標変換タイプ 256 を設定してください:

$\$MC\_TRAFO\_TYPE\_<n> = 256$

、ここで<n> = 1, 2, ... 座標変換の最大数

座標変換入力軸:  $\$MC\_TRAFO\_AXES\_IN\_<n>$

こうしたチャンネル軸は、座標変換直交座標系の軸を割り当てるマシンデータで指定します:

$\$MC\_TRAFO\_AXES\_IN\_<n>[ <index> ] = <チャンネル軸番号>$

、ここで<n> = 1, 2, ... 座標変換の最大数

<インデックス>	意味
0	直線軸、回転軸に直交
1	回転軸
2	直線軸、回転軸に平行

チャンネル軸番号は $\$MC\_TRAFO\_GEOAX\_ASSIGN\_TAB\_<n>$ により定義された軸の順序を示します。

8.1 TRANSMIT 面端座標変換(オプション)

1つの回転軸と2つの直線軸:TRAFO\_TYPE = 257

1つの回転軸および2つの直線軸を含む TRANSMIT には、座標変換タイプ 257 を設定してください:

**\$MC\_TRAFO\_TYPE\_<n> = 257**

、ここで<n> = 1、2、... 座標変換の最大数

2番目の直線軸は、回転軸および直線軸によって固定された平面に対して直交するように回転させてください。TRANSMIT の2番目の直線軸は、工具計算でのみ使用されません。

**座標変換入力軸: \$MC\_TRAFO\_AXES\_IN\_<n>**

こうしたチャンネル軸は、座標変換直交座標系の軸を割り当てるマシンデータで指定します:

**\$MC\_TRAFO\_AXES\_IN\_<n>[ <index> ] = <チャンネル軸番号>**

、ここで<n> = 1, 2, ... 座標変換の最大数

<インデックス>	意味
0	直線軸、回転軸に直交
1	回転軸
2	直線軸、回転軸に平行
3	インデックス 0 と 1 の軸に垂直な直線軸

チャンネル軸番号は\$MC\_TRAFO\_GEOAX\_ASSIGN\_TAB\_<n>により定義された軸の順序を示します。

回転軸オフセット:TRANSMIT\_ROT\_AX\_OFFSET

TRANSMIT 座標変換が有効である場合に、回転軸原点が回転軸のゼロ位置に一致しないときは、次のマシンデータにオフセットとして角度差を入力してください:

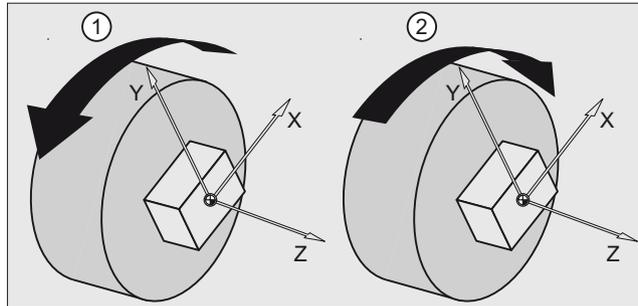
**\$MC\_TRANSMIT\_ROT\_AX\_OFFSET\_<t> = <角度差>**、

ここで<t> = 1、2

### 回転の方向:TRANSMIT\_ROT\_SIGN\_IS\_PLUS

TRANSMIT には、次のマシンデータにより回転軸の回転方向を指定します:

- X/Y 平面を基準にして Z 軸に向かって回転軸が左回りに回転する場合、正方向に移動するときは、TRANSMIT について回転軸の回転方向は正になります。
- 右回りに回転する場合、正の方向に移動するときは、TRANSMIT について回転軸の回転方向は負になります。



① 回転の正方向

② 回転の負方向

図 8-6 回転軸の回転方向

\$MC\_TRANSMIT\_ROT\_SIGN\_IS\_PLUS\_<t> = <回転方向>、  
ここで<t> = 1, 2

<回転方向>	意味
0	回転軸の負の回転方向⇒内部符号反転
1	回転軸の正の回転方向⇒内部符号反転なし

### 工具のゼロ位置:TRANSMIT\_BASE\_TOOL

工具原点位置は、TRANSMIT の有効な直交座標系の原点に対して指定します:

- MD24920 \$MC\_TRANSMIT\_BASE\_TOOL\_<t>[ 0 ] = <X 軸のオフセット>
- MD24920 \$MC\_TRANSMIT\_BASE\_TOOL\_<t>[ 1 ] = <Y 軸のオフセット>
- MD24920 \$MC\_TRANSMIT\_BASE\_TOOL\_<t>[ 2 ] = <Z 軸のオフセット>

ここで<t> = 1, 2

8.1 TRANSMIT 面端座標変換(オプション)

置換可能なジオメトリ軸

GEOAX () ジオメトリ軸を切り替えた場合、パラメータ設定された M 機能が NC/PLC インタフェースに出力されます:

- MD22534 \$MC\_TRAFO\_CHANGE\_M\_CODE = <M 機能>

注記

値 0 ~ 6、17、および 30 は出力されません。

参照先:

『機能マニュアル、基本機能』; 「軸と座標系とフレーム」 (K2)

8.1.3 プログラミング

前面座標変換(TRANSMIT)は、TRANSMIT 命令を使用してパートプログラムまたはシンクロナイズドアクションで有効になります。

構文

TRANSMIT  
TRANSMIT (<n>)

意味

TRANSMIT:	1 番目の TRANSMIT データセットで TRANSMIT を有効にします。
TRANSMIT (n):	n 番目の TRANSMIT データセットで TRANSMIT を有効にします。

注記

チャンネルで TRANSMIT 座標変換が有効になり、次のことが適用されます:

- 座標変換の解除:TRAFOOF
- 別の座標変換の適用:例: TRACYL、TRAANG、TRAORI

## 8.1.4 制約事項

### 先読み

先読みが必要なすべての機能(極を通過する移動、先読み)は、当該軸の移動が、あらかじめ正確に計算可能である場合にのみ、十分に機能します。TRANSMIT では、これが回転軸および、それに垂直な直線軸に適用されます。これらの軸の1つが位置決め軸である場合は、先読み機能がアラーム 10912 で解除され、通常オンライン速度チェックが代わりに有効になります。

### 方法の選択

ユーザは、「極を通過する移動」または「極を中心とした回転」のうち、最適な方を選択してください。

### 複数回の極通過

1つのブロックは、何度でも極を通過できます(複数の回転を伴う螺旋のプログラミングなど)。パートプログラムブロックは、対応するサブブロック数に分割されます。同様に、極を中心として複数回、回転するブロックも、サブブロックに分割されます。

### モジュロ軸としての回転軸

回転軸はモジュロ回転軸となることができます。ただし、これは、SW 2 および 3 の場合のような絶対必要条件ではありません。SW 2 および 3 で適用されている関連の制限事項は、なくなっています。

### 主軸としての回転軸

変換を伴わない回転軸が主軸として使用される場合は、変換を選択する前に、SPOS で位置制御モードに切り替えてください。

### 補助直線軸による TRANSMIT

TRANSMIT が有効な場合、パートプログラムではジオメトリ軸のように、posBCS[ax[3]] のチャンネル名称に別の名称が必要です。posBCS[ax[3]] が、TRANSMIT 変換指令なしのときだけ書き込まれる場合は、軸がジオメトリ軸に割り当てられていれば、この制限は適用されません。TRANSMIT が有効な場合は、ax[3]によって処理される輪郭情報はありません。

8.1 TRANSMIT 面端座標変換(オプション)

REPOS

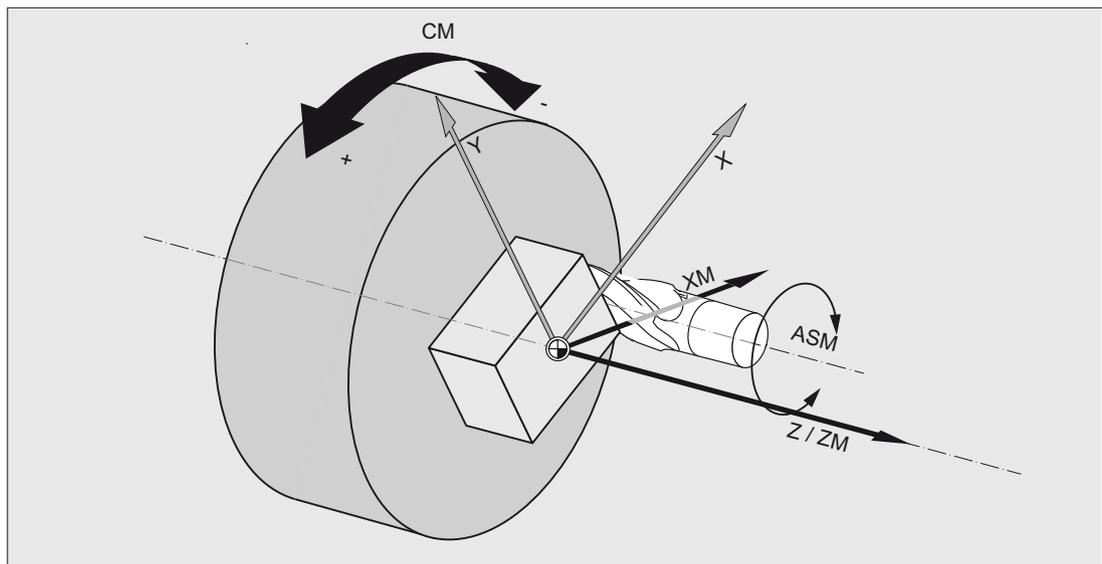
SW 4 では、拡張 TRANSMIT 機能によって作成されたサブブロックで再位置決めが可能です。この場合、制御装置は、BCS の再位置決め点に最も近くにある最初のサブブロックを使用します。

ブロック検索

計算有りブロック検索の場合は、SW 4 の拡張機能により中間ブロックが生成されていれば、(最後のサブブロックの)ブロック終点にアプローチします。

8.1.5 例

例は、下図の軸設定を示します。



- X、Y、 ジオメトリ軸
- Z
- CM     1 番目の機械軸:回転軸
- XM     2 番目の機械軸:直線軸、回転軸に直交
- ZM     3 番目の機械軸:直線軸、回転軸に平行
- ASM    4 番目の機械軸:主軸

## パラメータの割り付け

## 機械軸名称

- MD10000 \$MN\_AXCONF\_MACHAX\_NAME\_TAB[ 0 ] = 「CM」 (1 番目の機械軸:主軸/回転軸)
- MD10000 \$MN\_AXCONF\_MACHAX\_NAME\_TAB[ 1 ] = 「XM」 (2 番目の機械軸:直線軸)
- MD10000 \$MN\_AXCONF\_MACHAX\_NAME\_TAB[ 2 ] = 「ZM」 (3 番目の機械軸:直線軸)
- MD10000 \$MN\_AXCONF\_MACHAX\_NAME\_TAB[ 3 ] = 「ASM」 (4 番目の機械軸:主軸)

## ジオメトリ軸名称

- MD20060 \$MC\_AXCONF\_GEOAX\_NAME\_TAB[ 0 ] = 「X」 (1 番目のジオメトリ軸の名称)
- MD20060 \$MC\_AXCONF\_GEOAX\_NAME\_TAB[ 1 ] = 「Y」 (2 番目のジオメトリ軸の名称)
- MD20060 \$MC\_AXCONF\_GEOAX\_NAME\_TAB[ 2 ] = 「Z」 (3 番目のジオメトリ軸の名称)

## チャンネル軸名称

- MD20080 \$MC\_AXCONF\_CHANAX\_NAME\_TAB[ 0 ] = 「XC」 (直線軸)
- MD20080 \$MC\_AXCONF\_CHANAX\_NAME\_TAB[ 1 ] = 「ZC」 (直線軸)
- MD20080 \$MC\_AXCONF\_CHANAX\_NAME\_TAB[ 2 ] = 「CC」 (主軸/回転軸)
- MD20080 \$MC\_AXCONF\_CHANAX\_NAME\_TAB[ 3 ] = 「ASC」 (主軸)

## チャンネル軸へのジオメトリ軸の割り当て

TRANSMIT が有効でない:

- MD20050 \$MC\_AXCONF\_GEOAX\_ASSIGN\_TAB[ 0 ] = 1 (1 番目のジオメトリ軸 → 1 番目のチャンネル軸 XC)
- MD20050 \$MC\_AXCONF\_GEOAX\_ASSIGN\_TAB[ 1 ] = 0 (-)
- MD20050 \$MC\_AXCONF\_GEOAX\_ASSIGN\_TAB[ 2 ] = 2 (3 番目のジオメトリ軸 → 2 番目のチャンネル軸 ZC)

8.1 TRANSMIT 面端座標変換(オプション)

TRANSMIT が有効な場合:

- MD24120 \$MC\_TRAFO\_GEOAX\_ASSIGN\_TAB\_1[ 0 ] = 1 (1 番目の TrafoGeoAxis → 1 番目のチャンネル軸 XC)
- MD24120 \$MC\_TRAFO\_GEOAX\_ASSIGN\_TAB\_1[ 1 ] = 3 (2 番目の TrafoGeoAxis → 3 番目のチャンネル軸 CC)
- MD24120 \$MC\_TRAFO\_GEOAX\_ASSIGN\_TAB\_1[ 2 ] = 2 (3 番目の TrafoGeoAxis → 2 番目のチャンネル軸 ZC)

機械軸へのチャンネル軸の割り当て

- MD20070 \$MC\_AXCONF\_MACHAX\_USED[ 0 ] = 2 (1 番目のチャンネル軸 → 2 番目の機械軸 XM)
- MD20070 \$MC\_AXCONF\_MACHAX\_USED[ 1 ] = 3 (2 番目のチャンネル軸 → 3 番目の機械軸 ZM)
- MD20070 \$MC\_AXCONF\_MACHAX\_USED[ 2 ] = 1 (3 番目のチャンネル軸 → 1 番目の機械軸 CM)
- MD20070 \$MC\_AXCONF\_MACHAX\_USED[ 3 ] = 4 (4 番目のチャンネル軸 → 4 番目の機械軸 ASM)

主軸の定義

- MD35000 \$MA\_SPIND\_ASSIGN\_TO\_MACHAX[ 0 ] = 1 (1 番目の機械軸:主軸)
- MD35000 \$MA\_SPIND\_ASSIGN\_TO\_MACHAX[ 1 ] = 0 (2 番目の機械軸:軸)
- MD35000 \$MA\_SPIND\_ASSIGN\_TO\_MACHAX[ 2 ] = 0 (3 番目の機械軸:軸)
- MD35000 \$MA\_SPIND\_ASSIGN\_TO\_MACHAX[ 3 ] = 2 (4 番目の機械軸:主軸)

座標変換タイプ

- MD24100 \$MC\_TRAFO\_TYPE\_1 = 256 (回転軸または直線軸での座標変換 TRANSMIT)

回転軸のゼロ位置に対するオフセット

- MD24900 \$MC\_TRANSMIT\_ROT\_AX\_OFFSET\_1 = 0

回転軸の符号

- MD24910 \$MC\_TRANSMIT\_ROT\_SIGN\_IS\_PLUS\_1 = 0

**TRANSMIT が有効な場合の工具のゼロのジオメトリ軸に対する基本オフセット**

- MD24920 \$MC\_TRANSMIT\_BASE\_TOOL\_1[0]=0.0 (1 番目の TrafoGeoAxis に対するオフセット)
- MD24920 \$MC\_TRANSMIT\_BASE\_TOOL\_1[1]=0.0 (2 番目の TrafoGeoAxis に対するオフセット)
- MD24920 \$MC\_TRANSMIT\_BASE\_TOOL\_1[2]=0.0 (3 番目の TrafoGeoAxis に対するオフセット)

**TRANSMIT 入力軸**

- MD24110 \$MC\_TRAFO\_AXES\_IN\_1[0]=1 (1 番目のチャンネル軸 XC、回転軸に直交)
- MD24110 \$MC\_TRAFO\_AXES\_IN\_1[1]=3 (3 番目のチャンネル軸 CC、回転軸)
- MD24110 \$MC\_TRAFO\_AXES\_IN\_1[2]=2 (2 番目のチャンネル軸 ZC、回転軸に平行)

**回転軸のモジュロ変換**

- MD30300 \$MA\_ROT\_IS\_MODULO[0]=0 (1 番目の機械軸:主軸/回転軸)
- MD30300 \$MA\_ROT\_IS\_MODULO[1]=0 (2 番目の機械軸:直線軸)
- MD30300 \$MA\_ROT\_IS\_MODULO[2]=0 (3 番目の機械軸:直線軸)
- MD30300 \$MA\_ROT\_IS\_MODULO[3]=1 (4 番目の機械軸:主軸)

**プログラミング例**

プログラムコード	コメント
N10 T1 D1 G54 G17 G90 F5000 G94	; 工具選択 ; 1 番目の設定可能なゼロオフセット ; XY 平面 ; アブソリュート指令 ; 毎分送り F (mm/min)
N20 G0 X20 Z10 SPOS=45	; 開始位置へアプローチします ; 主軸位置決め 45
N30 TRANSMIT	; TRANSMIT、最初の ; MD24100 ON のデータブロックによる
N40 ROT RPL=-45	; フレーム:XY 平面での Z による回転
N50 ATRANS X-2 Y10	; フレーム:追加オフセット
N60 G1 X10 Y-10 G41 OFFN=1	; 四角形荒削り; 許容範囲 1 mm (OFFN)
N70 X-10	
N80 Y10	
N90 X10	
N100 Y-10	

## 8.2 TRACYL 円筒補間(オプション)

プログラムコード	コメント
N110 G0 Z20 G40 OFFN=0	; 工具径補正 OFF、 ; 許容量 0 mm
N120 T2 D1 X15 Y-15	; 工具交換
N130 Z10 G41	; 輪郭左側の工具径補正
N140 G1 X10 Y-10	; 4 面部品仕上げ
N150 X-10	
N160 Y10	
N170 X10	
N180 Y-10	
N190 Z20 G40	
N200 TRANS	; フレームの解除
N210 TRAFOOF	; TRANSMIT OFF
N220 G0 X20 Z10 SPOS=45	; 開始位置へアプローチします
N230 M30	

## 8.2 TRACYL 円筒補間(オプション)

## 8.2.1 機能

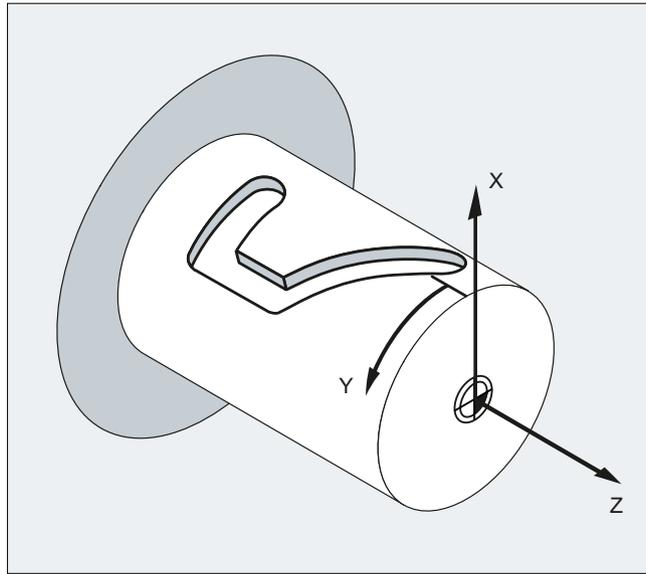
## 注記

「円筒補間 (TRACYL)」機能には、ライセンスを受けた「TRANSMIT と円筒補間」オプションが必要です。

TRACYL 座標変換により、旋盤上でシリンダジャケットカーブ(溝)の加工が可能になります。

溝の軌跡は、平面に展開された円筒面を基準にしてプログラム指令されます。

コントローラは、プログラム指令した円筒座標系の移動動作を、実際の機械軸の移動動作に変換します。



機械のキネマティクスは、円筒座標系に対応します:

- 1つ、2つ、または3つの直線軸と1つの回転軸
- 直線軸は、相互に直交するように回転させてください
- 回転軸は、いずれかの直線軸に平行になるように回転してください。

#### 注記

動作中の座標変換では、関連する機械、チャネル、およびジオメトリ軸の名称は異なります。

- MD10000 \$MN\_AXCONF\_MACHAX\_NAME\_TAB (機械軸名称)
- MD20080 \$MC\_AXCONF\_CHANAX\_NAME\_TAB (チャネル軸名称)
- MD20060 \$MC\_AXCONF\_GEOAX\_NAME\_TAB (ジオメトリ軸名称)

#### 座標変換タイプ

TRACYL 座標変換には、次の3つのタイプがあります。

- 溝側補正なし(座標変換タイプ 512):
- 溝側補正あり(座標変換タイプ 513):
- 溝側補正の有無の変更可(座標変換タイプ 514)

## 溝壁補正なしの TRACYL

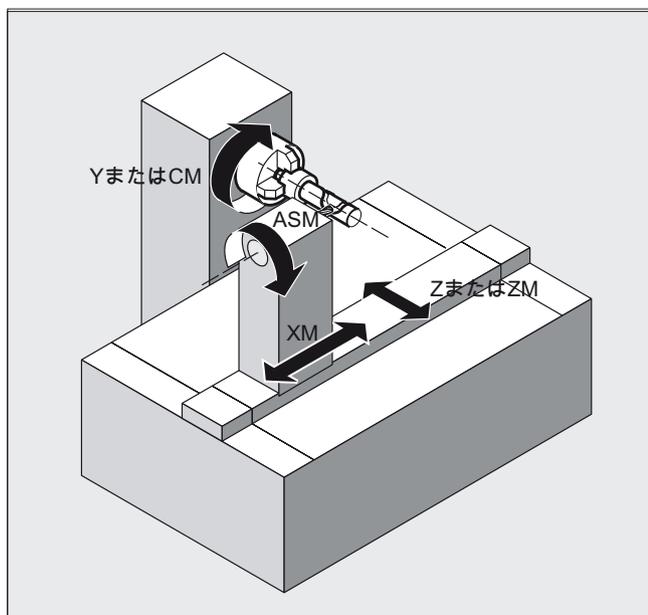
溝側補正なしの円筒補間は、1つまたは2つの直線軸(軸設定 1)を持つ機械のキネマティクスで使します。

## 1つの直線軸

1つの直線軸(X)のみが存在する機械キネマティックでは、円筒の周囲に平行な溝のみを生成できます(切り込み溝)。

## 2つの直線軸

2つの直線軸(X および Z)の機械キネマティックについて、円筒上で任意の形式の溝を生成できます。



XM 回転中心に対して直交している切り込み軸

ZM 回転中心に対して平行な直線軸

Y/CM 座標変換 Y 軸/回転軸

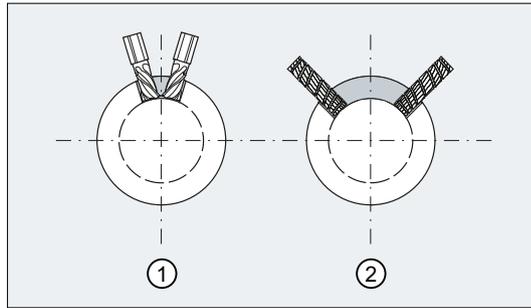
ASM 主軸

図 8-7 2つの直線軸を持つ機械のキネマティクス

## 溝の端面

溝壁補正なしの円筒補間の場合、回転軸に長手方向の溝(直線溝)の端面は、溝幅が工具直径と同じ場合にだけ平行になります。工具直径より広い溝幅の場合、溝の端面は相互に角度が付き(①を参照)。

外周に平行な溝の端面(径方向切り込み溝)は、相互に平行になりません。平行でないのは、溝端面の始点および終点のみです。

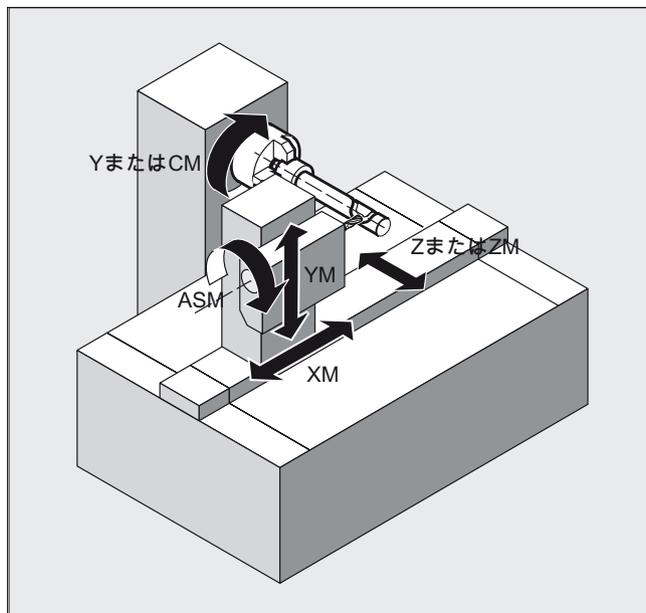


- ① 直線溝
- ② 径方向の溝

図 8-8 溝側補正なしの TRACYL を使った溝の端面

### 溝壁補正ありの TRACYL

溝壁補正ありの円筒補間は、3つの直線軸(X、Y、およびZ)(軸設定2)を持つ機械のキネマティクスで使します。



- XM 回転中心に対して直交している切り込み軸
- YM X-Z 平面に直交する補助軸
- ZM 回転中心に対して平行な直線軸
- Y/CM 座標変換 Y 軸/回転軸
- ASM 主軸

図 8-9 3つの直線軸を持つ機械のキネマティクス

## 8.2 TRACYL 円筒補間(オプション)

3つの直線軸(XおよびZ)の機械キネマティックについて、円筒上で任意の形式の溝を生成できます。

## 溝の端面

Y軸が回転中心に対して直角に配置されているため、工具直径より広い溝幅でも、ほとんど平行の溝端面を生成できます。

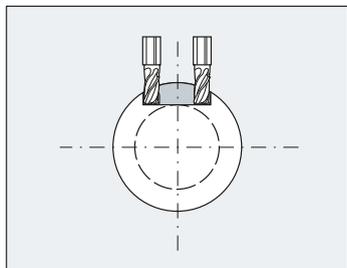


図 8-10 TRACYL を使った平行に制限された直線溝 (溝側補正あり)

## 8.2.2 パラメータの割り付け

## 8.2.2.1 一覧

## マシンデータ:一般の座標変換データ

チャンネルの座標変換データセットを定義するために、次のマシンデータを使用します。

- MD2xxxx \$MC\_TRAFO\_TYPE\_<n> (チャンネルの<n>番目の座標変換の定義)
- MD2xxxx \$MC\_TRAFO\_AXES\_IN\_<n> (チャンネルの<n>番目の座標変換の軸割り当て)
- MD2xxxx \$MC\_TRAFO\_GEOAX\_ASSIGN\_TAB\_<n> (座標変換<n>のためのジオメトリ軸のチャンネル軸への割り当て)
- MD2xxxx \$MC\_TRAFO\_INCLUDES\_TOOL\_<n> (有効な<n>番目の座標変換による工具処理)

ここで、<n> = 1、2、3、... 座標変換データセットの最大数

TRACYL (タイプ 512、513、または 514)の場合、1つのチャンネルにパラメータ設定できる座標変換データセットは最大2つです。

- MD2xxxx \$MC\_TRAFO\_TYPE\_<x> = <TRACYL タイプ>
- MD2xxxx \$MC\_TRAFO\_TYPE\_<y> = <TRACYL タイプ>

### マシンデータ:TRACYL 座標変換

TRACYL 座標変換は、次のマシンデータを使用してパラメータ設定します:

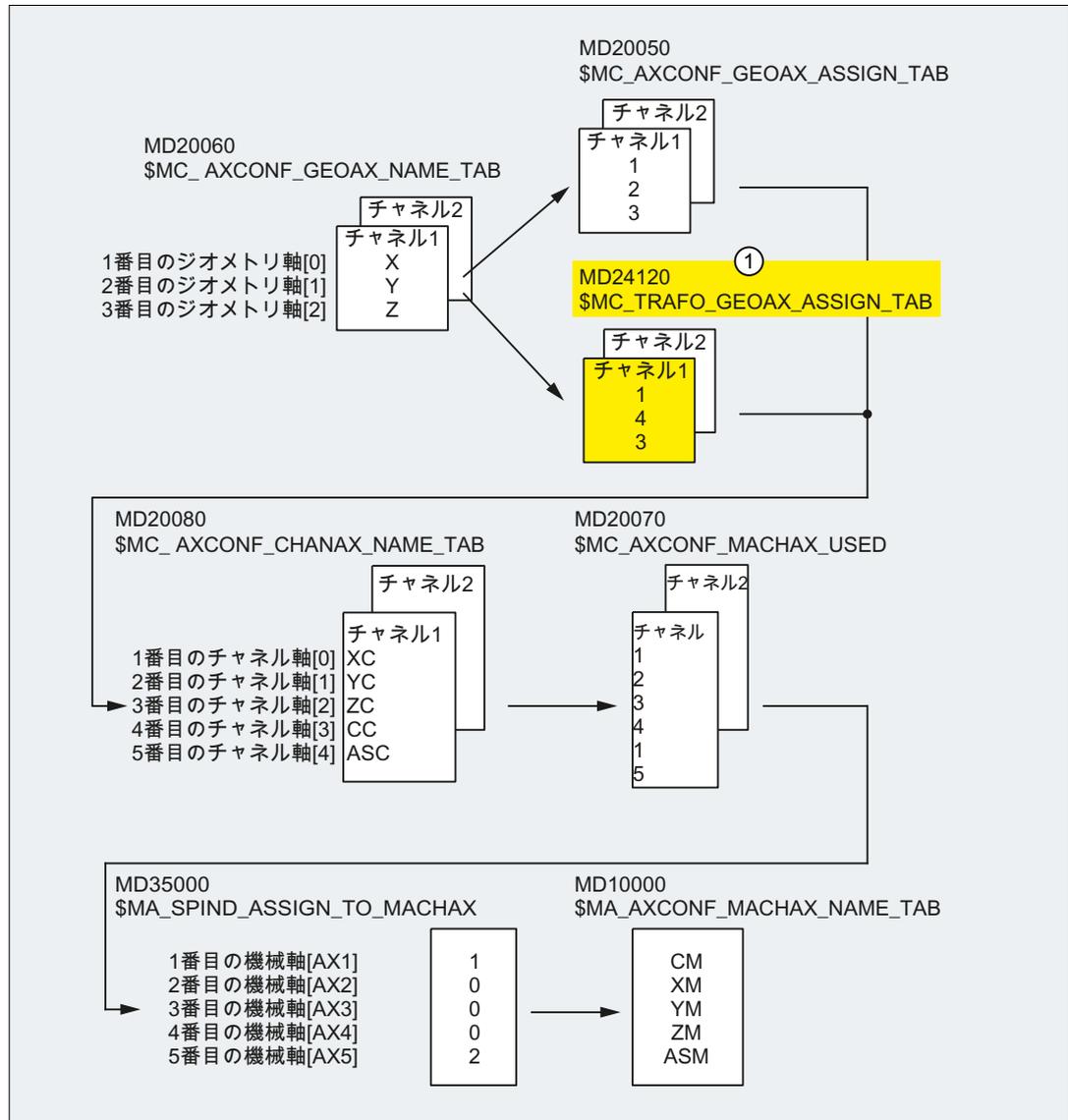
- MD2xxxx \$MC\_TRACYL\_ROT\_AX\_OFFSET\_<n> (回転軸のオフセット)
- MD2xxxx \$MC\_TRACYL\_ROT\_AX\_FRAME\_<n> (回転軸オフセット)
- MD2xxxx \$MC\_TRACYL\_DEFAULT\_MODE\_<n> (TRACYL モードの選択)
- MD2xxxx \$MC\_TRACYL\_ROT\_SIGN\_IS\_PLUS\_<n> (回転軸の符号)
- MD2xxxx \$MC\_TRACYL\_BASE\_TOOL\_<n> (基本工具のベクトル)

where <n> = 1, 2 (TRACYL データセット番号)

8.2 TRACYL 円筒補間(オプション)

8.2.2.2 軸構成

以下に、TRACYL で一般的な軸設定を示します。



① TRACYL が有効な場合に有効。

機械軸名称

- MD10000 \$MN\_AXCONF\_MACHAX\_NAME\_TAB[ 0 ] = 「CM」
- MD10000 \$MN\_AXCONF\_MACHAX\_NAME\_TAB[ 1 ] = 「XM」
- MD10000 \$MN\_AXCONF\_MACHAX\_NAME\_TAB[ 2 ] = 「YM」

- MD10000 \$MN\_AXCONF\_MACHAX\_NAME\_TAB[ 3 ] = 「ZM」
- MD10000 \$MN\_AXCONF\_MACHAX\_NAME\_TAB[ 4 ] = 「ASM」

## ジオメトリ軸名称

- MD20060 \$MC\_AXCONF\_GEOAX\_NAME\_TAB[ 0 ] = 「X」 (1 番目のジオメトリ軸の名称)
- MD20060 \$MC\_AXCONF\_GEOAX\_NAME\_TAB[ 1 ] = 「Y」 (2 番目のジオメトリ軸の名称)
- MD20060 \$MC\_AXCONF\_GEOAX\_NAME\_TAB[ 2 ] = 「Z」 (3 番目のジオメトリ軸の名称)

## チャンネル軸名称

- MD20080 \$MC\_AXCONF\_CHANAX\_NAME\_TAB[ 0 ] = 「XC」
- MD20080 \$MC\_AXCONF\_CHANAX\_NAME\_TAB[ 1 ] = 「YC」
- MD20080 \$MC\_AXCONF\_CHANAX\_NAME\_TAB[ 2 ] = 「ZC」
- MD20080 \$MC\_AXCONF\_CHANAX\_NAME\_TAB[ 3 ] = 「CC」
- MD20080 \$MC\_AXCONF\_CHANAX\_NAME\_TAB[ 4 ] = 「ASC」

## チャンネル軸へのジオメトリ軸の割り当て

TRACYL が有効でない

- MD20050 \$MC\_AXCONF\_GEOAX\_ASSIGN\_TAB[ 0 ] = 1 (1 番目のジオメトリ軸 → 1 番目のチャンネル軸 XC)
- MD20050 \$MC\_AXCONF\_GEOAX\_ASSIGN\_TAB[ 1 ] = 2 (2 番目のジオメトリ軸 → 2 番目のチャンネル軸 YC)
- MD20050 \$MC\_AXCONF\_GEOAX\_ASSIGN\_TAB[ 2 ] = 3 (3 番目のジオメトリ軸 → 3 番目のチャンネル軸 ZC)

TRACYL が有効:

- MD24120 \$MC\_TRAFO\_GEOAX\_ASSIGN\_TAB\_1[ 0 ] = 1 (1 番目の座標変換ジオメトリ軸 → 1 番目のチャンネル軸 XC)
- MD24120 \$MC\_TRAFO\_GEOAX\_ASSIGN\_TAB\_1[ 1 ] = 4 (2 番目の座標変換ジオメトリ軸 → 4 番目のチャンネル軸 CC)
- MD24120 \$MC\_TRAFO\_GEOAX\_ASSIGN\_TAB\_1[ 2 ] = 3 (3 番目の座標変換ジオメトリ軸 → 3 番目のチャンネル軸 ZC)

8.2 TRACYL 円筒補間(オプション)

機械軸へのチャネル軸の割り当て

- MD20070 \$MC\_AXCONF\_MACHAX\_USED[ 0 ] = 2 (1 番目のチャネル軸 → 2 番目の機械軸 XM)
- MD20070 \$MC\_AXCONF\_MACHAX\_USED[ 1 ] = 3 (2 番目のチャネル軸 → 3 番目の機械軸 YM)
- MD20070 \$MC\_AXCONF\_MACHAX\_USED[ 2 ] = 4 (3 番目のチャネル軸 → 4 番目の機械軸 ZM)
- MD20070 \$MC\_AXCONF\_MACHAX\_USED[ 3 ] = 1 (4 番目のチャネル軸 → 1 番目の機械軸 CM)
- MD20070 \$MC\_AXCONF\_MACHAX\_USED[ 4 ] = 5 (5 番目のチャネル軸 → 5 番目の機械軸 ASM)

主軸の定義

- MD35000 \$MA\_SPIND\_ASSIGN\_TO\_MACHAX[ 0 ] = 1 (主軸)
- MD35000 \$MA\_SPIND\_ASSIGN\_TO\_MACHAX[ 1 ] = 0 (軸)
- MD35000 \$MA\_SPIND\_ASSIGN\_TO\_MACHAX[ 2 ] = 0 (軸)
- MD35000 \$MA\_SPIND\_ASSIGN\_TO\_MACHAX[ 3 ] = 0 (軸)
- MD35000 \$MA\_SPIND\_ASSIGN\_TO\_MACHAX[ 4 ] = 2 (主軸)

8.2.2.3 個別設定

座標変換タイプの設定

座標変換タイプは、次のように座標変換データセット別に設定します。

\$MC\_TRAFO\_TYPE\_<n> = <座標変換タイプ>

ここで、<n> = 1、2、... 座標変換の最大数

軸構成	<座標変換タイプ>
1	512
2	513 または 514

プログラムされた溝側補正(タイプ 514)付きの TRACYL 座標変換は、溝側補正付きおよび溝側補正なしで有効にできます(「プログラミング (ページ 521)」を参照してください)。

### 溝壁補正なしの座標変換タイプ 514

機械に、回転軸と最初の直線軸の両方に垂直な別の直線軸がある場合は、座標変換タイプ 514 を使用して、実 Y 軸による工具オフセットを適用できます。この場合は、第 2 直線軸の作業領域が小さく、パートプログラムの実行には使用されないことを前提としています。

MD10000 \$MC\_TRAFO\_GEOAX\_ASSSIGN\_TAB\_<n>の既存の設定が適用されます。

### 溝壁オフセットのある溝

溝側補正ありの TRACYL 座標変換の場合は、予め工具オフセットが含まれていることが考慮されています。

### 軸イメージ

次の節では、座標変換軸イメージの指定方法を説明します。

### 座標変換ジオメトリ軸

座標変換データセット<n>では、TRACYL に 3 つ(または 4 つ)のチャネル軸番号を指定する必要があります。

- MD24110 \$MC\_TRAFO\_AXES\_IN\_1[0]=回転軸に対して半径方向軸のチャネル軸番号
- MD24110 \$MC\_TRAFO\_AXES\_IN\_1[1]=回転軸のチャネル軸番号
- MD24110 \$MC\_TRAFO\_AXES\_IN\_1[2]=回転軸に平行な軸のチャネル軸番号
- MD24110 \$MC\_TRAFO\_AXES\_IN\_1[3]=円筒面に平行で、回転軸に垂直な補助軸のチャネル軸番号(2 軸構成の場合)

軸番号は、次のマシンデータで定義されたチャネル軸の順序を基準にしてください。

MD24120 \$MC\_TRAFO\_GEOAX\_ASSIGN\_TAB\_<n>

### 溝壁補正のない溝

座標変換タイプ 514 の場合は、次のインデックスが MD24110 \$MC\_TRAFO\_AXES\_IN\_<n>[ ]に適用されます。

基本座標系(BCS)に関連するインデックスの意味は、次のとおりです。

- [0]:回転軸の半径方向直交軸(設定されている場合)
- [1]:回転軸に垂直な円筒面の軸

8.2 TRACYL 円筒補間(オプション)

- [2]:回転軸に平行な直交軸
- [3]:機械の初期位置で、インデックス 2 に平行な直線軸

機械座標系(MCS)に関連するインデックスの意味:

- [0]:回転軸の半径方向直線軸(設定されている場合)
- [1]:回転軸
- [2]:直線軸、回転軸に平行
- [3]:直線軸、インデックス[0]と[1]の軸に垂直

回転位置

回転軸に垂直な円筒面の軸の回転位置は、次のように定義してください。

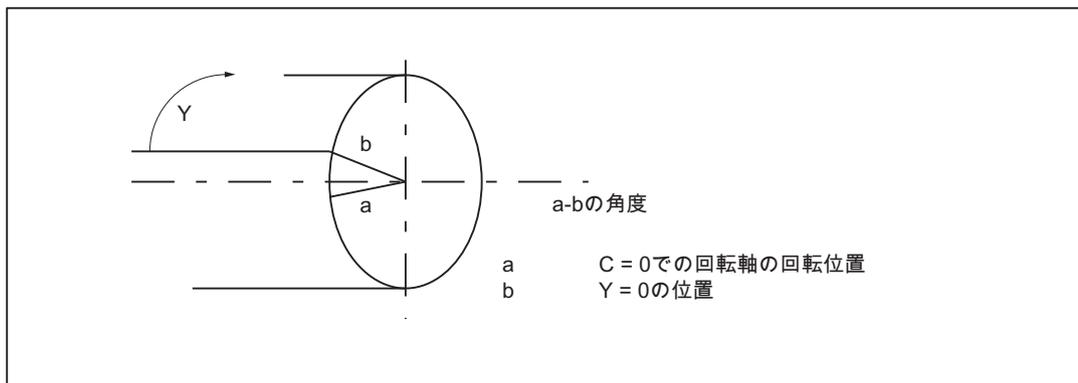


図 8-11 円筒面の軸回転の中心

MD24800 TRACYL\_ROT\_AX\_OFFSET\_<t>

回転軸の定義された原点位置を基準とする円筒面の回転位置は、以下によって指定されます。

MD24800 \$MC\_TRACYL\_ROT\_AX\_OFFSET\_<t> = ...°

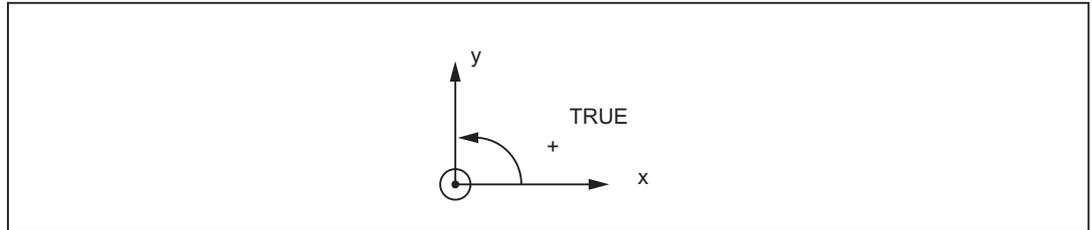
この場合、<t>は、座標変換データブロックで宣言された TRACYL 座標変換の数に置き換えられます(<t>は 2 以下でなければなりません)。

回転方向

回転軸の回転方向は、次の節で説明するマシンデータで指定されます。

**TRACYL\_ROT\_SIGN\_IS\_PLUS\_<t>**

X-Y 平面上の回転軸の回転方向が、Z 軸の周りに反時計りである場合は、マシンデータを TRUE に設定してください。それ以外の場合は、FALSE に設定してください。



MD24810 \$MC\_TRACYL\_ROT\_SIGN\_IS\_PLUS\_<t>=TRUE

この場合、「t」は、座標変換データブロックで宣言された TRACYL 座標変換の数に置き換えられます(t は 2 以下でなければなりません)。

**置換可能なジオメトリ軸**

ジオメトリ軸が、GEOAX()を使用して置換されると、マシンデータで設定できる M コードのオプション出力によって PLC に通知されます。

- MD22534 \$MC\_TRAFO\_CHANGE\_M\_CODE  
座標変換切り替えの場合に VDI インタフェースで出力される M コード番号

**注記**

このマシンデータが 0 ~ 6、17、30 のいずれかの値に設定されている場合は、M コードは出力されません。

**参照先:**

『機能マニュアル、基本機能』; 「軸と座標系とフレーム」(K2)

**工具原点位置**

直交座標系の基点を基準とした工具原点位置は、次の節に記載されているマシンデータで指定されます。

MD24820 TRACYL\_BASE\_TOOL\_<t>

このマシンデータを使用して、TRACYL に宣言された円筒座標系の基点を基準とした工具原点位置をコントローラへ通知します。マシンデータには、機械座標系の軸 X、Y、Z のための 3 種類の成分があります。

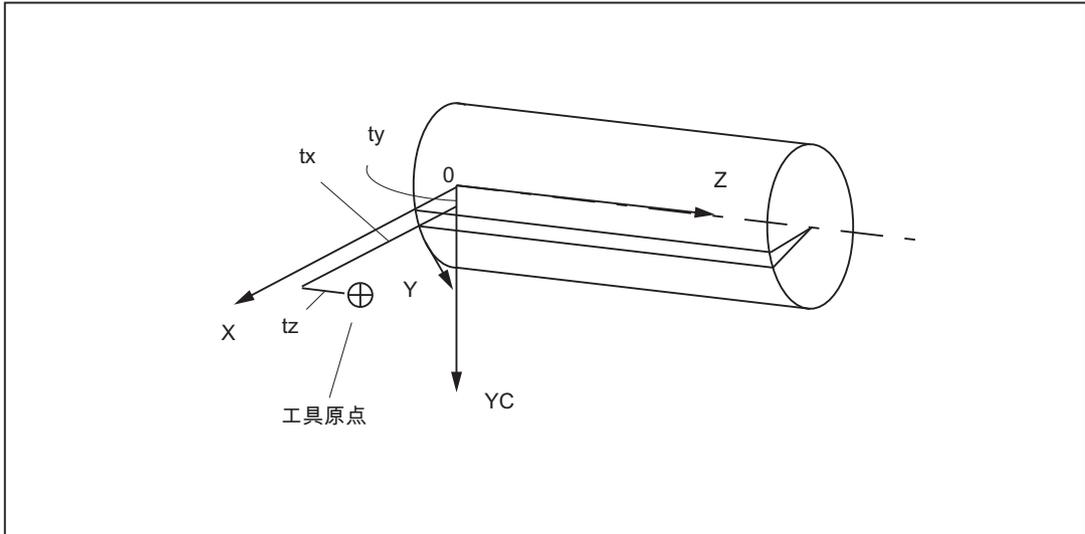


図 8-12 機械原点を基準とした工具原点位置

例

MD24820 \$MC\_TRACYL\_BASE\_TOOL\_<t>[ 0 ] = tx

MD24820 \$MC\_TRACYL\_BASE\_TOOL\_<t>[ 1 ] = ty

MD24820 \$MC\_TRACYL\_BASE\_TOOL\_<t>[ 2 ] = tz

<t> = 座標変換データセットで定義された TRACYL 座標変換の数

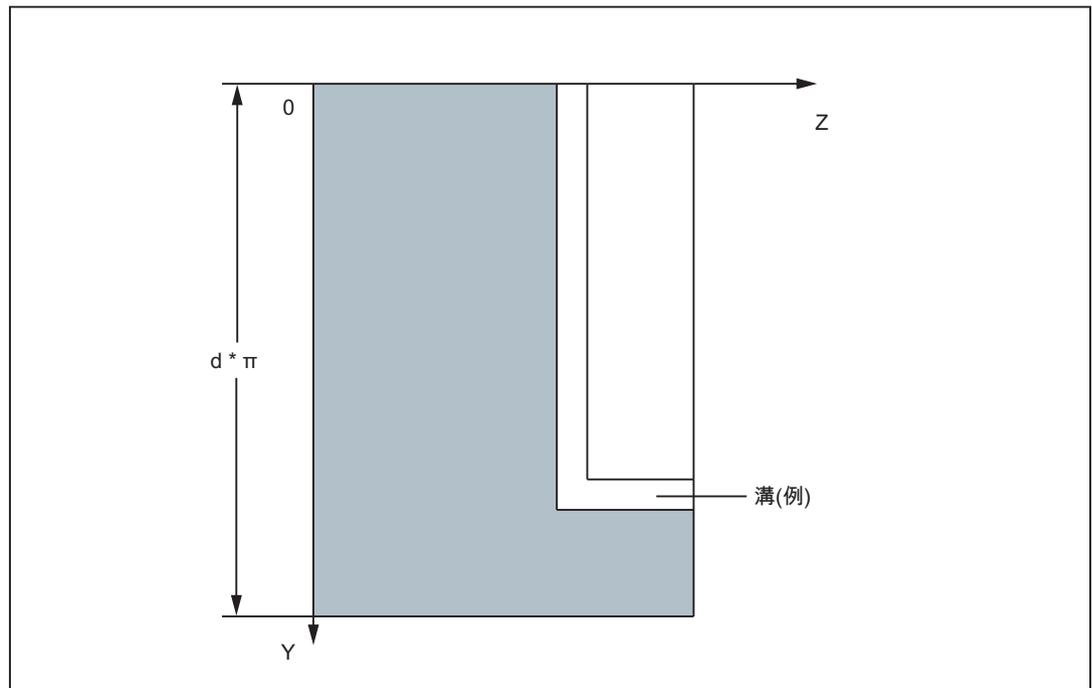


図 8-13 円筒座標系

### 8.2.3 プログラミング

円筒座標変換(TRACYL)は、TRACYL 命令を使用してパートプログラムまたはシンクロナイズドアクションで有効になります。

#### 構文

TRACYL (<d>)

TRACYL (<d>, <n>)

TRACYL (<d>, <n>, <k>)

#### 意味

TRACYL (<d>):	1 番目の TRACYL データセットと作業直径<d>で TRACYL を有効にします。
TRACYL (<d>, <n>):	<n>番目の TRACYL データセットと作業直径<d>で TRACYL を有効にします。

8.2 TRACYL 円筒補間(オプション)

<d>:	基準または作業直径 この値は 1 より大きくします。	
<n>:	TRACYL データセット番号(任意選択)	
	値の範囲:	1, 2
<k>:	パラメータ<k>は座標変換タイプ 514 にのみ関連します。	
	k = 0:	溝壁補正なし
	k = 1:	溝壁補正あり
	パラメータが指定されていない場合、次のパラメータ設定された基本位置が適用されます: \$MC_TRACYL_DEFAULT_MODE_<n> <n> = TRACYL データセット番号	

注記

チャンネル内で有効な TRACYL 座標変換は、以下によってオフに切り替えることができます。

- 座標変換の解除:TRAFOOF
- 別の座標変換の有効化:TRAANG、TRANSMIT、TRAORI など

例

プログラムコード	コメント
...	
N40 TRACYL(40.)	; 1 番目の TRACYL データセットと作業直径 40 mm で TRACYL を有効にします。
...	

詳細情報

プログラム構成

TRACYL 座標変換 513 (溝壁オフセットありの TRACYL)を使用して溝を加工するパートプログラムは通常、以下の手順から成ります。

1. 工具の選択
2. TRACYL の選択
3. 適切な座標オフセット(フレーム)の選択
4. 位置決め
5. OFFN のプログラム指令

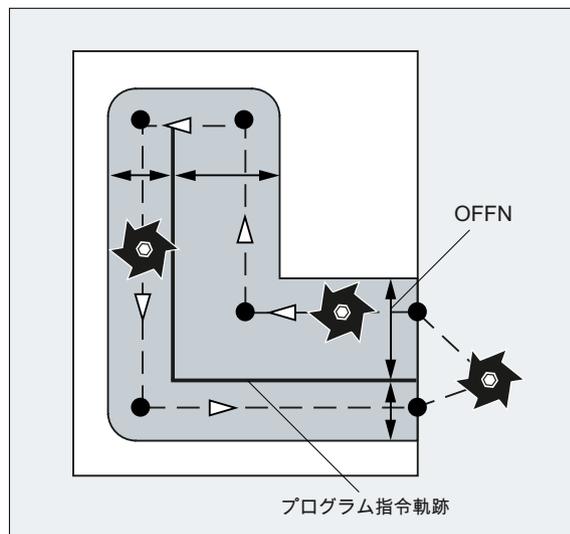
6. TRC の選択
7. 移動ブロック(TRC を位置決め後に溝壁へアプローチする)
8. 溝の中心線の輪郭
9. TRC の選択解除
- 10.後退ブロック(TRC を後退後に溝壁から離れる)
- 11.位置決め
- 12.TRAFOOF
- 13.オリジナルの座標シフト(フレーム)の再選択

### 輪郭オフセット(OFFN)

TRACYL 座標変換 513 を使用して溝を加工するには、パートプログラムで、溝の中心線と溝幅の半分を OFFN アドレスを使用してプログラム指令します。

溝壁の損傷を避けるために、OFFN は工具径補正が有効なときにのみ機能します。

パートプログラム内で OFFN を変更できます。これにより、溝の中心線を中心からオフセットすることができます。



### 注記

溝壁の反対側の損傷を避けるために、OFFN には工具の半径以上の値を指定してください。

### 注記

OFFN は、TRACYL と併用した場合と、TRACYL と併用しない場合では、動作が異なります。TRACYL と併用していない場合でも、TRC が有効なときは OFFN が含まれるため、OFFN は、TRAFOOF の後でゼロにリセットしてください。

## 8.2 TRACYL 円筒補間(オプション)

**通知**

**OFFN の動作は、座標変換タイプによって異なります。**

TRACYL 座標変換 513 (溝壁オフセットありの TRACYL)の場合、溝幅の半分を OFFN にプログラム指令します。

TRACYL 座標変換 512 (溝壁オフセットありの TRACYL)の場合、OFFN 値は TRC の許容範囲として機能します。

**工具径補正(TRC)**

TRACYL 座標変換 513 の場合、TRC は溝壁ではなく、プログラム指令された溝の中心を基準にして考慮されます。工具を溝壁の左に移動させるには、命令 **G41** の代わりに **G42** をプログラム指令するか、**OFFN** の値を負の符号付きで指定してください。

**工具直径**

TRACYL と直径が溝幅より小さい工具を使用した場合、直径が溝幅と同じ工具を使用した場合と同じ溝壁の形状は生成されません。精度を上げるには、工具の直径が溝幅より少しだけ小さくなるよう選択することをお勧めします。

**軸の使用****注記**

以下の軸は、位置決め軸と揺動軸のいずれにも使用できません。

- 円筒面の外周面方向のジオメトリ軸(Y 軸)
- 溝壁補正用の追加の直線軸(Z 軸)

## 8.2.4 境界条件

**選択/選択解除**

- 中間移動ブロックは挿入されません(面取り/半径)。
- 一連のスプラインブロックは終了させてください。
- 工具径補正は選択解除してください。
- TRACYL の前に有効であったフレームは、コントロールシステム(アナログ G500)によって解除されます。
- 座標変換(アナログ WALIMOF)の影響を受ける軸について、有効な作業領域リミットが解除されます。

- 連続軌跡の制御と丸み付けは中断されます。
- DRF オフセットは、オペレータが削除してください。
- 補正に使用される Y 軸は、選択および有効な溝壁補正のために原点に設定する必要があります。

## 工具交換

工具は、工具径補正機能が選択解除されているときにのみ交換できます。

## フレーム

G91(インクリメンタル指令)によるフレーム変更は特に、有効な座標変換では処理されません。移動するパスは、どのフレームが前のブロックで有効であったかにかかわらず、新しいフレームのワーク座標系で使用されます。

たとえば、ワークの傾斜した位置を補正することにより入力された回転軸オフセットは、フレームを使用して、または回転軸のオフセットとして考慮できます。

座標変換内で作用する回転軸の全軸フレームについて次の設定が必要です:

```
$MC_TRACYL_ROT_AX_FRAME_<t> = 1
```

---

### 注記

軸割り付けの変更は、座標変換が選択または選択解除されるたびに変換されます。

### 参照先:

『機能マニュアル、基本機能』; 「軸と座標系とフレーム (K2)」。

---

## 座標変換軸の機能制限

円筒補間に関連する軸は、次の機能では使用できません:

- 位置決め軸
- 揺動軸
- 事前設定軸
- 固定点アプローチ(G75)
- レファレンス点復帰(G74)

## 8.2 TRACYL 円筒補間(オプション)

## ジョグの手動移動

溝壁補正のある円筒補間(\$MC\_TRAFO\_TYPE = 513)がジョグモードで有効な場合は、軸が、その前の AUTOMATIC モードの状態に応じて移動することに注意してください。したがって有効な溝壁補正では、補正が解除された場合とは軸の移動が異なります。したがって、パートプログラムは、中断後も継続して動作(REPOS)できます。

## パートプログラムの中断

- **AUTOMATIC からジョグへのモード変更**  
有効な座標変換に対してパートプログラム加工が中断され、ジョグ運転モードで手動移動する場合、AUTOMATIC 運転モードでのパートプログラムの継続性のために、現在位置から中断位置への再起動ブロックで座標変換が既に有効であることを確認してください。

**警告****衝突の可能性**

干渉の監視はおこなわれません。

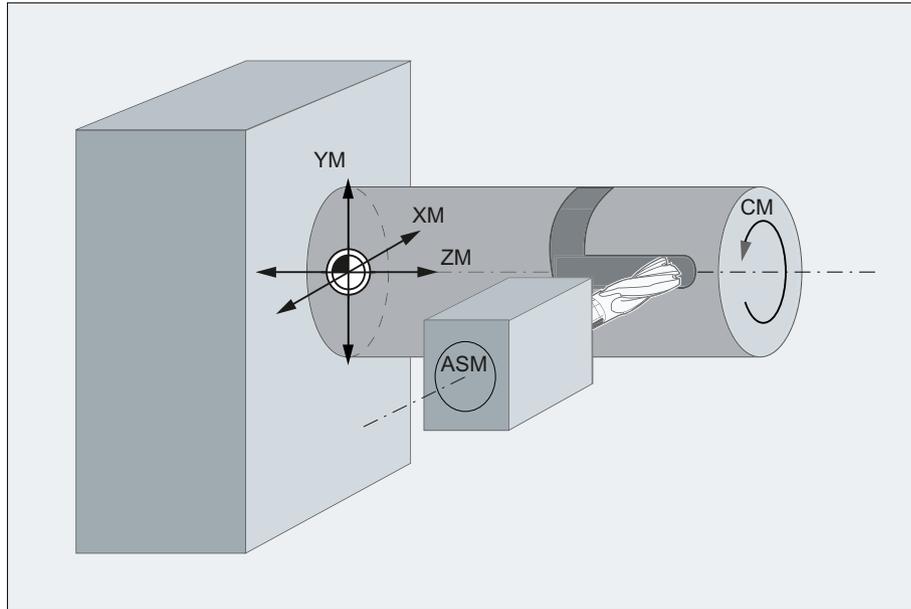
オペレータは、工具位置へ問題なく再位置決めできるように確認してください。

- **RESET 後の START**  
パートプログラム加工が RESET により中断され、START により再起動された場合、パートプログラムで定義された位置への直線ブロック(G0 または G1)によりすべての軸移動が開始し、以降のパートプログラムが再現可能な方法で移動することを確認してください。RESET 時に有効であった工具は、コントローラでは考慮されない場合があります(マシンデータによって設定可能)。

## 8.2.5 例

## 8.2.5.1 X-Y-Z-C キネマティクスによる円筒面上の溝加工

例は、下図の追加の Y 軸を描画した旋盤を示します。



XM 切り込み軸、回転軸に直交

YM 付加軸

ZM 軸は回転軸に平行

CM 回転軸

AS 主軸

M

## パラメータの割り付け

## 機械軸名称

- MD10000 \$MN\_AXCONF\_MACHAX\_NAME\_TAB[ 0 ] = 「CM」
- MD10000 \$MN\_AXCONF\_MACHAX\_NAME\_TAB[ 1 ] = 「XM」
- MD10000 \$MN\_AXCONF\_MACHAX\_NAME\_TAB[ 2 ] = 「YM」
- MD10000 \$MN\_AXCONF\_MACHAX\_NAME\_TAB[ 3 ] = 「ZM」
- MD10000 \$MN\_AXCONF\_MACHAX\_NAME\_TAB[ 4 ] = 「ASM」

8.2 TRACYL 円筒補間(オプション)

ジオメトリ軸名称

- MD20060 \$MC\_AXCONF\_GEOAX\_NAME\_TAB[ 0 ] = 「X」 (1 番目のジオメトリ軸の名称)
- MD20060 \$MC\_AXCONF\_GEOAX\_NAME\_TAB[ 1 ] = 「Y」 (2 番目のジオメトリ軸の名称)
- MD20060 \$MC\_AXCONF\_GEOAX\_NAME\_TAB[ 2 ] = 「Z」 (3 番目のジオメトリ軸の名称)

チャンネル軸名称

- MD20080 \$MC\_AXCONF\_CHANAX\_NAME\_TAB[ 0 ] = 「XC」
- MD20080 \$MC\_AXCONF\_CHANAX\_NAME\_TAB[ 1 ] = 「YC」
- MD20080 \$MC\_AXCONF\_CHANAX\_NAME\_TAB[ 2 ] = 「ZC」
- MD20080 \$MC\_AXCONF\_CHANAX\_NAME\_TAB[ 3 ] = 「CC」
- MD20080 \$MC\_AXCONF\_CHANAX\_NAME\_TAB[ 4 ] = 「ASC」

チャンネル軸へのジオメトリ軸の割り当て

TRACYL が有効でない

- MD20050 \$MC\_AXCONF\_GEOAX\_ASSIGN\_TAB[ 0 ] = 1 (1 番目のジオメトリ軸 → 1 番目のチャンネル軸 XC)
- MD20050 \$MC\_AXCONF\_GEOAX\_ASSIGN\_TAB[ 1 ] = 2 (2 番目のジオメトリ軸 → 2 番目のチャンネル軸 YC)
- MD20050 \$MC\_AXCONF\_GEOAX\_ASSIGN\_TAB[ 2 ] = 3 (3 番目のジオメトリ軸 → 3 番目のチャンネル軸 ZC)

TRACYL が有効:

- MD24120 \$MC\_TRAFO\_GEOAX\_ASSIGN\_TAB\_1[ 0 ] = 1 (1 番目の座標変換ジオメトリ軸 → 1 番目のチャンネル軸 XC)
- MD24120 \$MC\_TRAFO\_GEOAX\_ASSIGN\_TAB\_1[ 1 ] = 4 (2 番目の座標変換ジオメトリ軸 → 4 番目のチャンネル軸 CC)
- MD24120 \$MC\_TRAFO\_GEOAX\_ASSIGN\_TAB\_1[ 2 ] = 3 (3 番目の座標変換ジオメトリ軸 → 3 番目のチャンネル軸 ZC)

**機械軸へのチャネル軸の割り当て**

- MD20070 \$MC\_AXCONF\_MACHAX\_USED[ 0 ] = 2 (1 番目のチャネル軸 → 2 番目の機械軸 XM)
- MD20070 \$MC\_AXCONF\_MACHAX\_USED[ 1 ] = 3 (2 番目のチャネル軸 → 3 番目の機械軸 YM)
- MD20070 \$MC\_AXCONF\_MACHAX\_USED[ 2 ] = 4 (3 番目のチャネル軸 → 4 番目の機械軸 ZM)
- MD20070 \$MC\_AXCONF\_MACHAX\_USED[ 3 ] = 1 (4 番目のチャネル軸 → 1 番目の機械軸 CM)
- MD20070 \$MC\_AXCONF\_MACHAX\_USED[ 4 ] = 5 (5 番目のチャネル軸 → 5 番目の機械軸 ASM)

**主軸の定義**

- MD35000 \$MA\_SPIND\_ASSIGN\_TO\_MACHAX[ 0 ] = 1 (主軸)
- MD35000 \$MA\_SPIND\_ASSIGN\_TO\_MACHAX[ 1 ] = 0 (軸)
- MD35000 \$MA\_SPIND\_ASSIGN\_TO\_MACHAX[ 2 ] = 0 (軸)
- MD35000 \$MA\_SPIND\_ASSIGN\_TO\_MACHAX[ 3 ] = 0 (軸)
- MD35000 \$MA\_SPIND\_ASSIGN\_TO\_MACHAX[ 4 ] = 2 (主軸)

**座標変換タイプ**

- MD24100 \$MC\_TRAFO\_TYPE\_1 = 513 (溝側補正ありの TRACYL)

**回転軸のゼロ位置に対するオフセット**

- MD24800 \$MC\_TRACYL\_ROT\_AX\_OFFSET\_1 = 0

**回転軸の符号**

- MD24810 \$MC\_TRACYL\_ROT\_SIGN\_IS\_PLUS\_1 = FALSE

**TRACYL が有効な場合の工具のゼロのジオメトリ軸に対する基本オフセット**

- MD24820 \$MC\_TRACYL\_BASE\_TOOL\_1 [ 0 ] = 0.0 (1 番目の座標変換ジオメトリ軸に対するオフセット)
- MD24820 \$MC\_TRACYL\_BASE\_TOOL\_1 [ 1 ] = 0.0 (2 番目の座標変換ジオメトリ軸に対するオフセット)
- MD24820 \$MC\_TRACYL\_BASE\_TOOL\_1 [ 2 ] = 0.0 (3 番目の座標変換ジオメトリ軸に対するオフセット)

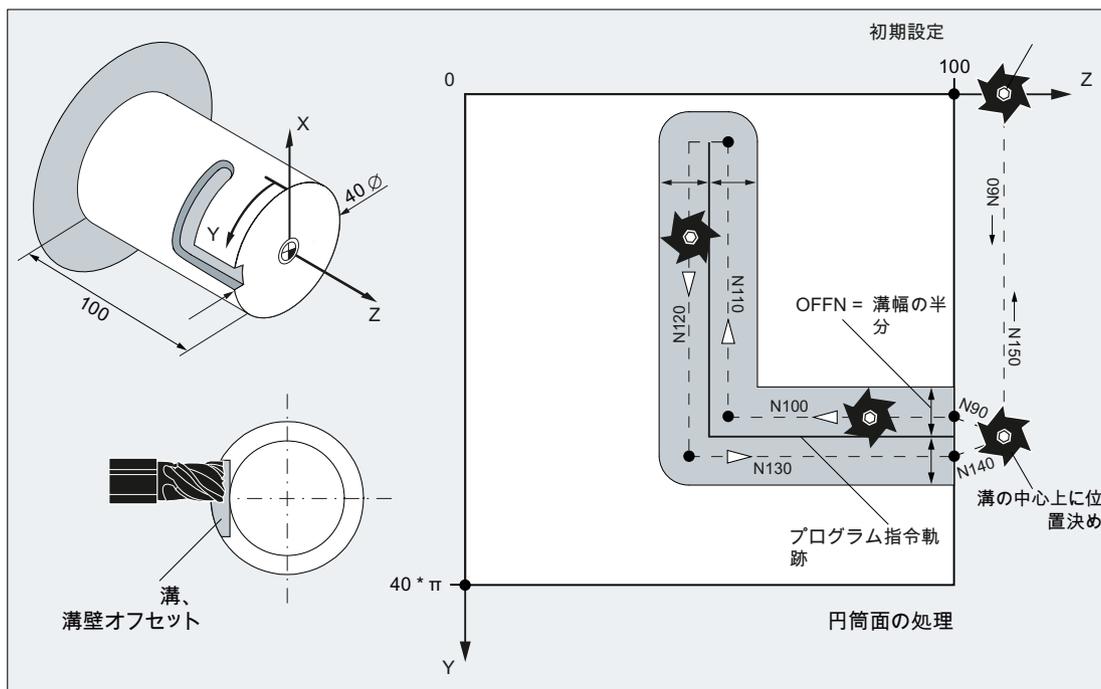
8.2 TRACYL 円筒補間(オプション)

TRACYL 入力軸

- MD24110 \$MC\_TRAFO\_AXES\_IN\_1[0] = 1 (1 番目のチャンネル軸 XC、回転軸に直交)
- MD24110 \$MC\_TRAFO\_AXES\_IN\_1[1] = 4 (4 番目のチャンネル軸 CC、回転軸)
- MD24110 \$MC\_TRAFO\_AXES\_IN\_1[2] = 3 (3 番目のチャンネル軸 ZC、回転軸に平行)
- MD24110 \$MC\_TRAFO\_AXES\_IN\_1[3] = 2 (2 番目のチャンネル軸 YC、付加軸)

プログラミング

溝側補正(TRACYL 座標変換タイプ 513)を使ったフック型の溝の作成



工具の定義

プログラムコード	コメント
; 工具パラメータ	
\$TC_DP1[1,1] = 120	; 工具タイプ:フライス工具
\$TC_DP2[1,1]=0	; 刃先位置:旋盤のみ
プログラムコード	コメント
; ジオメトリ:長さ補正	
\$TC_DP3[1,1]=8.	; 長補正ベクトル:タイプに基づく計算
\$TC_DP4[1,1]=9.	; 長補正ベクトル:平面に基づく計算
\$TC_DP5[1,1]=7.	

プログラムコード	コメント
; ジオメトリ:半径	
\$TC_DP6[1,1]=6.	; 半径
\$TC_DP7[1,1]=0	; メタルソーの溝幅 b、フライス工具の丸み付け半径
\$TC_DP8[1,1]=0	; 突起 k:メタルソーの場合のみ
\$TC_DP9[1,1]=0	
\$TC_DP10[1,1]=0	
\$TC_DP11[1,1]=0	; テーパーフライス工具の角度

プログラムコード	コメント
; 摩耗:長さと径補正	
\$TC_DP12[1,1] = 0	; その他のパラメータは\$TC_DP24 まで =0 (基本寸法/アダプタ)

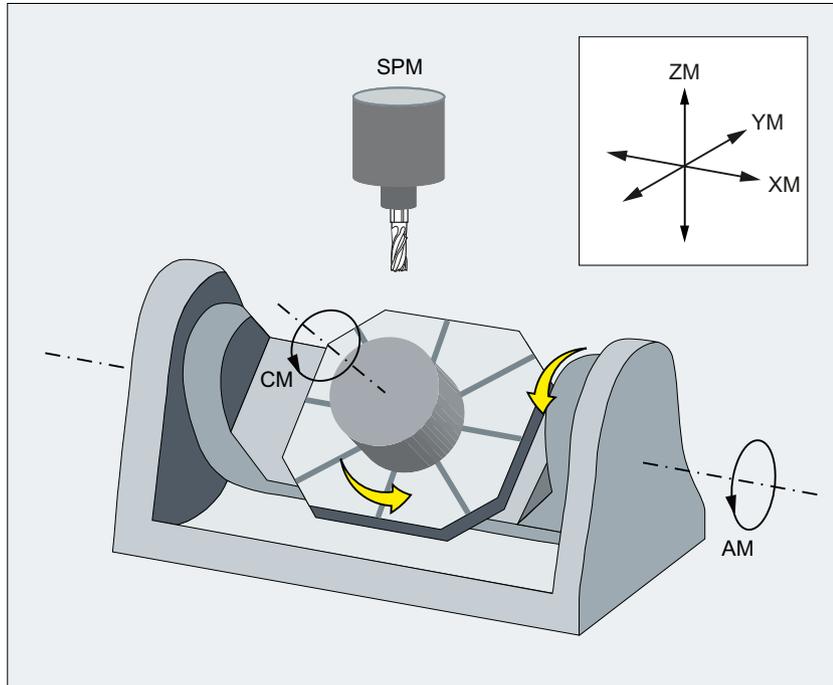
### 円筒面の溝加工

プログラムコード	コメント
N10 T1 D1 G54 G90 F5000 G94	; 工具の選択、クランプ補正
N20 SPOS=0	; 主軸の回転軸運転への切り替え、0度で位置決め
N30 G0 X25 Z110	; 開始位置へアプローチします
N40 TRACYL(40.)	; 1番目の TRACYL データセットと作業直径 40 mm で TRACYL を有効にします
N50 G19	; 加工平面 Y/Z (円筒面)
N60 Y70	; 溝の中心上に位置決め
N70 G1 X15	; 溝の底部への工具の切り込み
N80 OFFN=10	; 溝の中心線に対して 10 mm の溝壁間隔を定義します
N90 Z100 G42	; TRC 選択および右側の溝壁へのアプローチ ; G42:プログラム可能な輪郭(□ 溝の中心)の右側の TRC
N100 Z50	; 軌跡 I:円筒軸に平行な溝セクションの作成
N110 Y10	; 軌跡 I:外周に平行な溝セクションの作成
N120 Y70	; 軌跡 II:外周に平行な溝セクションの作成
N130 Z100	; 軌跡 II:円筒軸に平行な溝セクションの作成
N140 Z110 G40	; 溝壁から離れ、TRC を解除
N150 G0 X25 Y0	; 初期位置に戻る
N170 TRAF00F	; 座標変換の解除
N180 M30	; プログラム終了

8.2 TRACYL 円筒補間(オプション)

8.2.5.2 X-Y-Z-A-C キネマティクスによる円筒面上の溝加工

例は、下図の A および C 軸を備えた 5 軸フライス盤を示します。



- XM 1. 加工平面の軸
- YM 2. 加工平面の軸
- ZM 切り込み軸
- AM 回転軸
- CM 回転軸
- SP 主軸
- M

パラメータの割り付け

機械軸名称

- MD10000 \$MN\_AXCONF\_MACHAX\_NAME\_TAB[ 0 ] = 「XM」
- MD10000 \$MN\_AXCONF\_MACHAX\_NAME\_TAB[ 1 ] = 「YM」
- MD10000 \$MN\_AXCONF\_MACHAX\_NAME\_TAB[ 2 ] = 「ZM」
- MD10000 \$MN\_AXCONF\_MACHAX\_NAME\_TAB[ 3 ] = 「SPM」

- MD10000 \$MN\_AXCONF\_MACHAX\_NAME\_TAB[ 4 ] = 「AM」
- MD10000 \$MN\_AXCONF\_MACHAX\_NAME\_TAB[ 5 ] = 「CM」

#### ジオメトリ軸名称

- MD20060 \$MC\_AXCONF\_GEOAX\_NAME\_TAB[ 0 ] = 「X」 (1 番目のジオメトリ軸の名称)
- MD20060 \$MC\_AXCONF\_GEOAX\_NAME\_TAB[ 1 ] = 「Y」 (2 番目のジオメトリ軸の名称)
- MD20060 \$MC\_AXCONF\_GEOAX\_NAME\_TAB[ 2 ] = 「Z」 (3 番目のジオメトリ軸の名称)

#### チャンネル軸名称

- MD20080 \$MC\_AXCONF\_CHANAX\_NAME\_TAB[ 0 ] = 「XC」
- MD20080 \$MC\_AXCONF\_CHANAX\_NAME\_TAB[ 1 ] = 「YC」
- MD20080 \$MC\_AXCONF\_CHANAX\_NAME\_TAB[ 2 ] = 「ZC」
- MD20080 \$MC\_AXCONF\_CHANAX\_NAME\_TAB[ 3 ] = 「SPC」
- MD20080 \$MC\_AXCONF\_CHANAX\_NAME\_TAB[ 4 ] = 「AC」
- MD20080 \$MC\_AXCONF\_CHANAX\_NAME\_TAB[ 5 ] = 「CC」

#### チャンネル軸へのジオメトリ軸の割り当て

TRACYL が有効でない

- MD20050 \$MC\_AXCONF\_GEOAX\_ASSIGN\_TAB[ 0 ] = 1 (1 番目のジオメトリ軸 → 1 番目のチャンネル軸 XC)
- MD20050 \$MC\_AXCONF\_GEOAX\_ASSIGN\_TAB[ 1 ] = 2 (2 番目のジオメトリ軸 → 2 番目のチャンネル軸 YC)
- MD20050 \$MC\_AXCONF\_GEOAX\_ASSIGN\_TAB[ 2 ] = 3 (3 番目のジオメトリ軸 → 3 番目のチャンネル軸 ZC)

TRACYL が有効:

- MD24220 \$MC\_TRAFO\_GEOAX\_ASSIGN\_TAB\_2[ 0 ] = 6 (1 番目の座標変換ジオメトリ軸 → 6 番目のチャンネル軸 CC)
- MD24220 \$MC\_TRAFO\_GEOAX\_ASSIGN\_TAB\_2[ 1 ] = 2 (2 番目の座標変換ジオメトリ軸 → 2 番目のチャンネル軸 YC)
- MD24220 \$MC\_TRAFO\_GEOAX\_ASSIGN\_TAB\_2[ 2 ] = 3 (3 番目の座標変換ジオメトリ軸 → 3 番目のチャンネル軸 ZC)

8.2 TRACYL 円筒補間(オプション)

機械軸へのチャネル軸の割り当て

- MD20070 \$MC\_AXCONF\_MACHAX\_USED[ 0 ] = 1 (1 番目のチャネル軸 → 1 番目の機械軸 XM)
- MD20070 \$MC\_AXCONF\_MACHAX\_USED[ 1 ] = 2 (2 番目のチャネル軸 → 2 番目の機械軸 YM)
- MD20070 \$MC\_AXCONF\_MACHAX\_USED[ 2 ] = 3 (3 番目のチャネル軸 → 3 番目の機械軸 ZM)
- MD20070 \$MC\_AXCONF\_MACHAX\_USED[ 3 ] = 4 (4 番目のチャネル軸 → 4 番目の機械軸 SPM)
- MD20070 \$MC\_AXCONF\_MACHAX\_USED[ 4 ] = 5 (5 番目のチャネル軸 → 5 番目の機械軸 AM)
- MD20070 \$MC\_AXCONF\_MACHAX\_USED[ 5 ] = 6 (6 番目のチャネル軸 → 6 番目の機械軸 CM)

主軸の定義

- MD35000 \$MA\_SPIND\_ASSIGN\_TO\_MACHAX[ 0 ] = 0 (軸)
- MD35000 \$MA\_SPIND\_ASSIGN\_TO\_MACHAX[ 1 ] = 0 (軸)
- MD35000 \$MA\_SPIND\_ASSIGN\_TO\_MACHAX[ 2 ] = 0 (軸)
- MD35000 \$MA\_SPIND\_ASSIGN\_TO\_MACHAX[ 3 ] = 1 (主軸)
- MD35000 \$MA\_SPIND\_ASSIGN\_TO\_MACHAX[ 4 ] = 0 (軸)
- MD35000 \$MA\_SPIND\_ASSIGN\_TO\_MACHAX[ 4 ] = 0 (軸)

座標変換タイプ

- MD24200 \$MC\_TRAFO\_TYPE\_2 = 513 (溝側補正ありの TRACYL)

回転軸のゼロ位置に対するオフセット

- MD24850 \$MC\_TRACYL\_ROT\_AX\_OFFSET\_2 = 0

回転軸の符号

- MD24860 \$MC\_TRACYL\_ROT\_SIGN\_IS\_PLUS\_2 = FALSE

**TRACYL が有効な場合の工具のゼロのジオメトリ軸に対する基本オフセット**

- MD24870 \$MC\_TRACYL\_BASE\_TOOL\_2 [ 0 ] = 0.0 (1 番目の座標変換ジオメトリ軸に対するオフセット)
- MD24870 \$MC\_TRACYL\_BASE\_TOOL\_2 [ 1 ] = 0.0 (2 番目の座標変換ジオメトリ軸に対するオフセット)
- MD24870 \$MC\_TRACYL\_BASE\_TOOL\_2 [ 2 ] = 0.0 (3 番目の座標変換ジオメトリ軸に対するオフセット)

**TRACYL 入力軸**

- MD24210 \$MC\_TRAFO\_AXES\_IN\_2 [ 0 ] = 3 (3 番目のチャンネル軸 ZC、回転軸に直交)
- MD24210 \$MC\_TRAFO\_AXES\_IN\_2 [ 1 ] = 6 (6 番目のチャンネル軸 CC、回転軸)
- MD24210 \$MC\_TRAFO\_AXES\_IN\_2 [ 2 ] = 2 (2 番目のチャンネル軸 YC、回転軸に平行)
- MD24210 \$MC\_TRAFO\_AXES\_IN\_2 [ 3 ] = 1 (1 番目のチャンネル軸 XC、付加軸)

**プログラミング**

プログラムコード	コメント
N10 WORKPIECE ( , " , "CYLINDER" , 0 , 0 , -180 , -80 , 179 )	; ブランク定義
N20 M3 S2000	; 主軸速度の設定
N30 T="NUTFRAESER" M6 D1	; 工具選択
N40 G0 G54 X0 Y-20 Z105	; 位置決め
N50 CYCLE800 ( 0 , "TABLE" , 100000 , 57 , 0 , 0 , 0 , -90 , 0 , 0 , 0 , 0 , -1 , 100 , 1 )	; 旋回サイクルで A 軸を回転
N60 G17 G90	; 加工平面の設定
N70 G0 Y-10 Z100 G40	; 位置決め
N80 TRACYL ( 179 , 2 )	; 溝壁補正ありの Tracyl データセット 2 を選択
N90 OFFN=20	; オフセットの設定 (溝幅の半分)
N100 G1 F500 X0 Z75 G42	; 起点の設定および TRC の選択
N110 Y30	; 溝の中心軌跡
N120 X-60	; 溝の中心軌跡
N130 X0	; 溝の中心軌跡
N140 Y-10	; 溝の中心軌跡
N150 Z105 G40	; TRC の後退および解除
N160 TRAFOOF	; 座標変換の解除
N170 G0 X0 Y-20 Z115	; 後退位置決め動作
N180 M5	; 主軸停止
N190 CYCLE800 ( 0 , "TABLE" , 100000 , 57 , 0 , 0 , 0 , 0 , 0 , 0 , 0 , -1 , 100 , 1 )	; A 軸を戻す
N200 M30	; プログラム終了

## 8.3 傾斜角座標変換(傾斜軸) TRAANG (オプション)

### 8.3.1 機能

**注記**

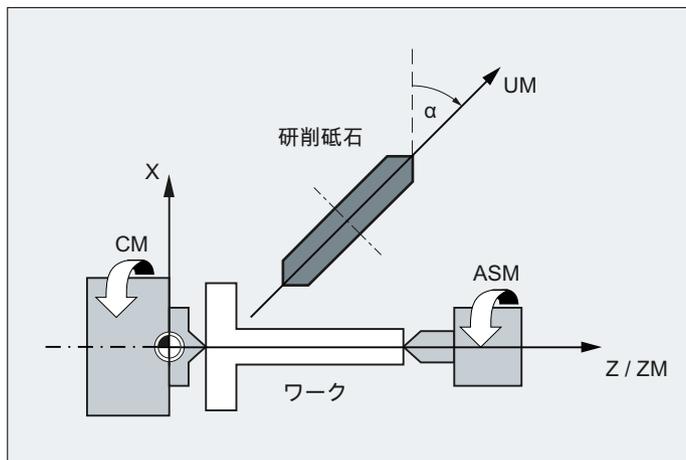
**オプション**

「角度がプログラム指令可能な傾斜角座標変換(TRAANG)」機能には、オプション「6FC5800-0AM28-0YB0」が必要です。このオプションにはライセンスが必要です。

「角度が固定の傾斜角座標変換(TRAANG)」機能には、オプション「6FC5800-0AS54-0YB0」が必要です。このオプションにはライセンスが必要です。

傾斜角座標変換により、研削機械で一般的な軸構成である傾斜機械軸を持つ機械上の直交ワーク座標系(WCS)でプログラミングが可能になります。

コントローラは、プログラム指令した直交座標系の移動動作を、実際の機械軸の移動動作に変換します。



- X ジオメトリ軸
- Z ジオメトリ軸
- ZM 機械軸
- UM 機械軸
- α 傾斜軸の角度
- CM 機械軸(主軸)
- AS 機械軸(心押し軸)
- M

図 8-14 傾斜切り込み軸による研削

## バージョン

- 角度が**プログラム指令可能な傾斜角座標変換(TRAANG)**  
角度がプログラム指令可能な傾斜角座標変換(TRAANG)では、変換を有効にするときに角度を指定できます。  
「角度がプログラム指令可能な傾斜角座標変換(TRAANG)の有効化(TRAANG) (ページ 542)」の章を参照
- 角度が**固定の傾斜角座標変換(TRAANG)**  
角度が固定の傾斜角座標変換(TRAANG)では、変換を有効にするときに角度を指定できません。  
「角度が固定の傾斜角座標変換の有効化(TRAANG) (ページ 544)」の章を参照

## 注記

動作中の座標変換では、関連する機械、チャンネル、およびジオメトリ軸の名称は異なります。

- MD10000 \$MN\_AXCONF\_MACHAX\_NAME\_TAB (機械軸名称)
- MD20080 \$MC\_AXCONF\_CHANAX\_NAME\_TAB (チャンネル軸名称)
- MD20060 \$MC\_AXCONF\_GEOAX\_NAME\_TAB (ジオメトリ軸名称)

## 8.3.2 パラメータの割り付け

## 8.3.2.1 一覧

## マシンデータ:一般の座標変換データ

チャンネルの座標変換データセットを定義するために、次のマシンデータを使用します。

- MD2xxxx \$MC\_TRAFO\_TYPE\_<n> (チャンネルの<n>番目の座標変換の定義)
- MD2xxxx \$MC\_TRAFO\_AXES\_IN\_<n> (チャンネルの<n>番目の座標変換の軸割り当て)
- MD2xxxx \$MC\_TRAFO\_GEOAX\_ASSIGN\_TAB\_<n> (座標変換<n>のためのジオメトリ軸のチャンネル軸への割り当て)
- MD2xxxx \$MC\_TRAFO\_INCLUDES\_TOOL\_<n> (有効な<n>番目の座標変換による工具処理)

ここで、<n> = 1、2、3、... 座標変換データセットの最大数

### 8.3 傾斜角座標変換(傾斜軸) TRAANG (オプション)

TRAANG (タイプ 1024)の場合、1つのチャンネルにパラメータ設定できる座標変換データセットは最大2つです。

- MD2xxxx \$MC\_TRAFO\_TYPE\_<x> = <TRAANG タイプ>
- MD2xxxx \$MC\_TRAFO\_TYPE\_<y> = <TRAANG タイプ>

#### マシンデータ:座標変換 TRAANG

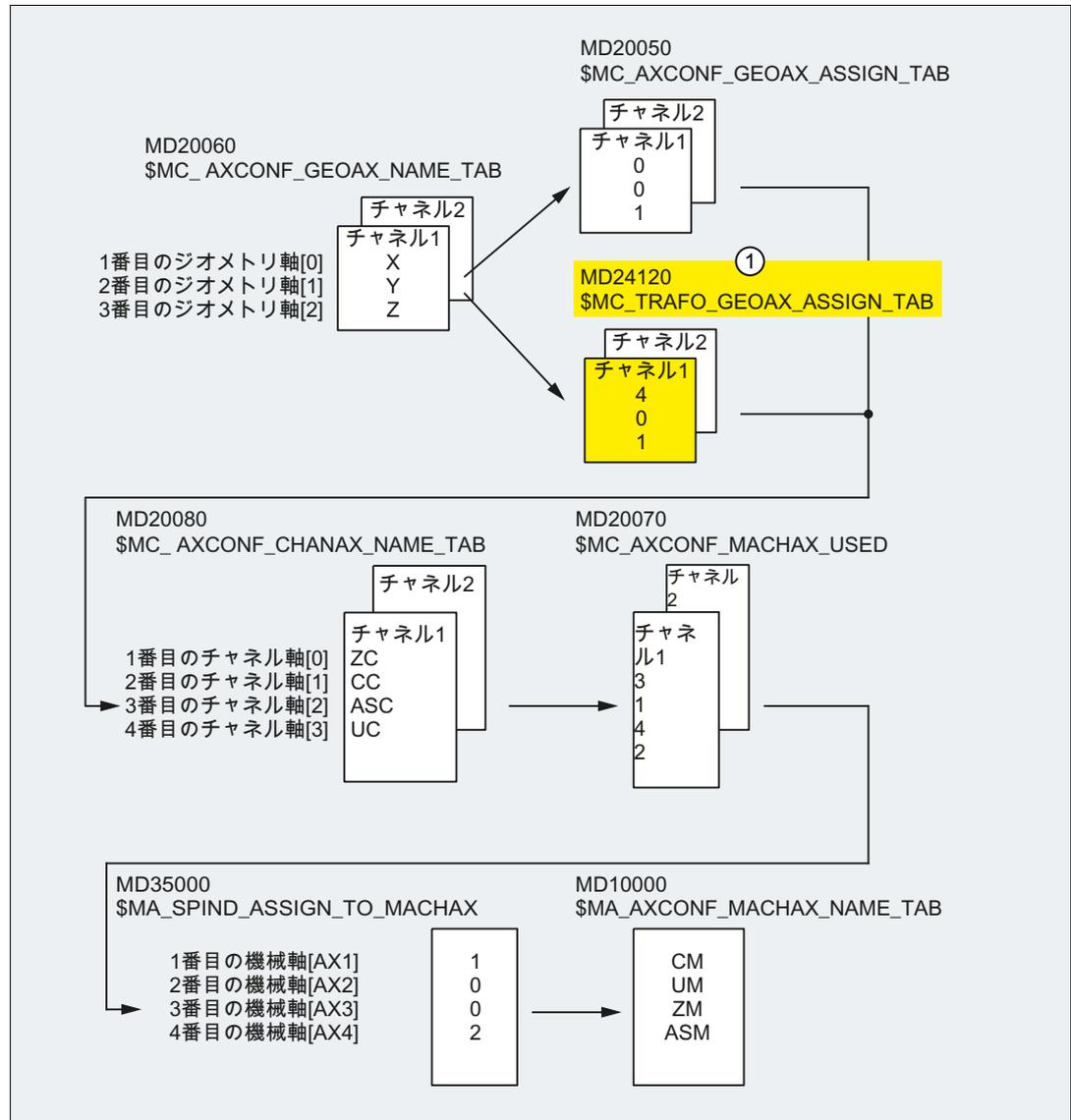
TRAANG 座標変換は、次のマシンデータを使用してパラメータ設定します。

- MD2xxxx \$MC\_TRAANG\_ANGLE\_<n> (手軸と傾斜軸間の角度)
- MD2xxxx \$MC\_TRAANG\_BASE\_TOOL\_<n> (工具原点の基本オフセット)
- MD2xxxx \$MC\_TRAANG\_PARALLEL\_VELO\_RES\_<n> (長手軸の補正動作の速度マージン  
)
- MD2xxxx \$MC\_TRAANG\_PARALLEL\_ACCEL\_RES\_<n> (長手軸の補正動作の加速マージン  
)

ここで、<n> = 1、2 (TRAANG データセット番号)

### 8.3.2.2 軸構成

以下に、TRAANG で一般的な軸設定を示します。



① TRAANG が有効な場合に有効。

#### 機械軸名称

- MD10000 \$MN\_AXCONF\_MACHAX\_NAME\_TAB[ 0 ] = 「CM」
- MD10000 \$MN\_AXCONF\_MACHAX\_NAME\_TAB[ 1 ] = 「UM」
- MD10000 \$MN\_AXCONF\_MACHAX\_NAME\_TAB[ 2 ] = 「ZM」
- MD10000 \$MN\_AXCONF\_MACHAX\_NAME\_TAB[ 3 ] = 「ASM」

8.3 傾斜角座標変換(傾斜軸) TRAANG (オプション)

ジオメトリ軸名称

- MD20060 \$MC\_AXCONF\_GEOAX\_NAME\_TAB[ 0 ] = 「X」 (1 番目のジオメトリ軸の名称)
- MD20060 \$MC\_AXCONF\_GEOAX\_NAME\_TAB[ 1 ] = 「Y」 (2 番目のジオメトリ軸の名称)
- MD20060 \$MC\_AXCONF\_GEOAX\_NAME\_TAB[ 2 ] = 「Z」 (3 番目のジオメトリ軸の名称)

チャンネル軸名称

- MD20080 \$MC\_AXCONF\_CHANAX\_NAME\_TAB[ 0 ] = 「ZC」
- MD20080 \$MC\_AXCONF\_CHANAX\_NAME\_TAB[ 1 ] = 「CC」
- MD20080 \$MC\_AXCONF\_CHANAX\_NAME\_TAB[ 2 ] = 「ASC」
- MD20080 \$MC\_AXCONF\_CHANAX\_NAME\_TAB[ 3 ] = 「UC」

チャンネル軸へのジオメトリ軸の割り当て

TRAANG が有効でない場合:

- MD20050 \$MC\_AXCONF\_GEOAX\_ASSIGN\_TAB[ 0 ] = 0 (-)
- MD20050 \$MC\_AXCONF\_GEOAX\_ASSIGN\_TAB[ 1 ] = 0 (-)
- MD20050 \$MC\_AXCONF\_GEOAX\_ASSIGN\_TAB[ 2 ] = 1 (3 番目のジオメトリ軸 → 1 番目のチャンネル軸 ZC)

TRAANG が有効の場合:

- MD24120 \$MC\_TRAFO\_GEOAX\_ASSIGN\_TAB\_1[ 0 ] = 4 (1 番目の座標変換ジオメトリ軸 → 4 番目のチャンネル軸 UC)
- MD24120 \$MC\_TRAFO\_GEOAX\_ASSIGN\_TAB\_1[ 1 ] = 0 (-)
- MD24120 \$MC\_TRAFO\_GEOAX\_ASSIGN\_TAB\_1[ 2 ] = 1 (3 番目の座標変換ジオメトリ軸 → 1 番目のチャンネル軸 ZC)

機械軸へのチャンネル軸の割り当て

- MD20070 \$MC\_AXCONF\_MACHAX\_USED[ 0 ] = 3 (1 番目のチャンネル軸 → 3 番目の機械軸 ZM)
- MD20070 \$MC\_AXCONF\_MACHAX\_USED[ 1 ] = 1 (2 番目のチャンネル軸 → 1 番目の機械軸 CM)

- MD20070 \$MC\_AXCONF\_MACHAX\_USED[ 2 ] = 4 (3 番目のチャンネル軸 → 4 番目の機械軸 ASM)
- MD20070 \$MC\_AXCONF\_MACHAX\_USED[ 3 ] = 2 (4 番目のチャンネル軸 → 2 番目の機械軸 UM)

#### 主軸の定義

- MD35000 \$MA\_SPIND\_ASSIGN\_TO\_MACHAX[ 0 ] = 1 (主軸)
- MD35000 \$MA\_SPIND\_ASSIGN\_TO\_MACHAX[ 1 ] = 0 (軸)
- MD35000 \$MA\_SPIND\_ASSIGN\_TO\_MACHAX[ 2 ] = 0 (軸)
- MD35000 \$MA\_SPIND\_ASSIGN\_TO\_MACHAX[ 3 ] = 2 (主軸)

### 8.3.2.3 個別設定

#### 長手軸と傾斜軸間の角度

- MD24700 \$MC\_TRAANG\_ANGLE\_<n> = <角度>

-90° < 角度 < 90°、0°を除く

角度は、X から開始して右回りの方向に正の角度で計測されます(セクション「機能 (ページ 536):角度 α」を参照してください)。

#### 工具原点位置の基本オフセット

有効な座標変換の有効なジオメトリ軸について、工具原点位置のベースオフセットを指定します。ベースオフセットでは、工具長補正を選択する場合と選択しない場合があります。プログラムされた工具長補正が、ベース工具に追加されます。

- \$MC\_TRAANG\_BASE\_TOOL\_<n>[ k ] = <ベースオフセット>

with k = 0、1、2 (第 1 ~ 第 3 のジオメトリ軸)

---

#### 注記

角度が変更された場合に、工具の原点位置は変換されません。

---

#### 速度制御の最適化

次のマシンデータを使用して、ジョグモード、位置決めモード、および揺動モードで速度制御が最適化されます:

8.3 傾斜角座標変換(傾斜軸) TRAANG (オプション)

速度マージン

次のマシンデータでは、長手軸の補正移動の速度マージンを設定します:

MD24720 \$MC\_TRAANG\_PARALLEL\_VELO\_RES\_<n> = <値>

<値>	意味
0.0	速度マージンは、同じ速度制限によって長手軸および関連する直交(仮想)軸の方向になるように、傾斜軸の角度、傾斜軸および長手軸の速度機能に応じて、NCによって決定されます。
> 0.0	速度マージン = <値> * (長手軸の MD32000 \$MA_MAX_AX_VELO)

加速マージン:

次のマシンデータでは、長手軸の補正移動の加減速マージンを設定します:

\$MC\_TRAANG\_PARALLEL\_ACCEL\_RES\_<n> = <値>

<値>	意味
0.0	加減速マージンは、同じ加減速制限によって長手軸および関連する直交(仮想)軸の方向になるように、傾斜軸の角度、傾斜軸および長手軸の加減速機能に応じて、NCによって決定されます。
> 0.0	加減速マージン = <値> * (長手軸の MD32300 \$MA_MAX_AX_ACCEL)

座標変換切り替え用 M コード

座標変換の切り換えのために、指定された M コードが出力されます:

- MD22534 \$MC\_TRAFO\_CHANGE\_M\_CODE = <値>

 <b>警告</b>
M コードは出力されません。 値 0 ~ 6、17 および 30 は、PLC に出力されません。

8.3.3 プログラミング

8.3.3.1 角度がプログラム指令可能な傾斜角座標変換(TRAANG)の有効化(TRAANG)

角度がプログラム指令可能な傾斜角座標変換(TRAANG)は、TRAANG 命令を使用して、パートプログラムまたはシンクロナイズドアクションで有効にします。

## 構文

```

TRAANG
TRAANG ()
TRAANG (, <n>)
TRAANG (<α>)
TRAANG (<α>, <n>)

```

## 意味

TRAANG: TRAANG ():	1 番目の TRAANG データセットと最後の有効な角度<α>で TRAANG を有効にします。	
TRAANG (, <n>):	<n>番目の TRAANG データセットと最後の有効な角度<α>で TRAANG を有効にします。	
TRAANG (<α>):	1 番目の TRAANG データセットと角度<α>で TRAANG を有効にします。	
TRAANG (<α>, <n>):	<n>番目の TRAANG データセットと角度<α>で TRAANG を有効にします。	
<α>:	傾斜軸の角度(任意選択)	
	値の範囲:	$-90^\circ < \alpha < +90^\circ$
	角度を指定しない場合は、次のマシンデータでパラメータ設定した初期状態が有効になります。 MD2xxxx \$MC_TRAANG_ANGLE_<n>	
<n>:	TRAANG データセット番号(任意選択)	
	値の範囲:	1, 2

## 注記

チャンネルで有効な傾斜角座標変換 TRAANG は、以下を行って無効にします。

- 座標変換の解除:TRAF00F
- 別の座標変換の適用:例: TRACYL、TRANSMIT、TRAORI

## 例

プログラムコード	コメント
N20 TRAANG (45)	; 1 番目の TRAANG データセットと角度 45°で TRAANG を有効にします。

8.3 傾斜角座標変換(傾斜軸) TRAANG (オプション)

8.3.3.2 角度が固定の傾斜角座標変換の有効化(TRAANG)

角度が固定の傾斜角座標変換(TRAANG)は、TRAANG 命令を使用して、パートプログラムまたはシンクロナイズドアクションで有効にします。有効にすると、角度を指定できなくなります。特定の TRAANG データセットのマシンデータでパラメータ設定した角度が有効になります。

構文

```
TRAANG
TRAANG ()
TRAANG (, <n>)
```

意味

TRAANG: TRAANG ():	1 番目の TRAANG データセット、およびマシンデータ MD24700 \$MC_TRAANG_ANGLE_1 の角度< $\alpha$ >で TRAANG を有効にします。	
TRAANG (, <n>):	<n>番目の TRAANG データセット、およびマシンデータ MD2xxxx \$MC_TRAANG_ANGLE_<n>の角度< $\alpha$ >で TRAANG を有効にします。	
<n>:	TRAANG データセット番号(任意選択)	
	値の範囲:	1, 2

注記

チャンネルで有効な傾斜角座標変換 TRAANG は、以下を行って無効にします。

- 座標変換の解除:TRAFOOF
- 別の座標変換の適用:例: TRACYL、TRANSMIT、TRAORI

例

プログラムコード	コメント
N20 TRAANG (, 2)	; 2 番目の TRAANG データセットおよび ; MD24750 \$MC_TRAANG_ANGLE_2 の角度で TRAANG を有効にします。

### 8.3.3.3 研削盤での傾斜プランジ切削(G5, G7)

G 命令の G7 と G5 は、「傾斜軸」 (TRAANG)による研削盤での傾斜プランジ切削のプログラム指令を簡素化するのに使用されます。その結果、プランジ切削時に傾斜軸だけが移動します。

プランジ切削動作の必要な終了位置だけを X と Z にプログラム指令します。G7 では、X 軸の現在位置から開始して、NC が傾斜軸のプログラム指令された終了位置と角度  $\alpha$  を計算し、アプローチします。

開始位置は、2 つの直線の交点から計算されます。

- X 軸の現在値から少し離れた Z 軸に平行な直線
- プログラム指令された終了位置を通る傾斜軸に平行な直線

この後の G5 で、傾斜軸はプログラム指令された終了位置に移動します。

#### 構文

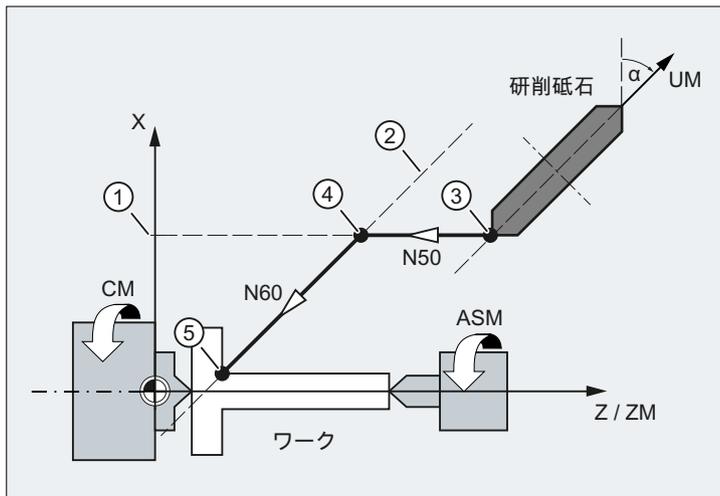
```
G7 <終了位置_X> <終了位置_Z>
G5 <終了位置_X>
```

#### 意味

G7:	傾斜プランジ切削の開始地点を計算し、アプローチします。
G5:	傾斜軸をプログラム指令された終了位置に移動します。
<終了位置_X>:	X 軸の終了位置
<終了位置_Z>:	Z 軸の終了位置

8.3 傾斜角座標変換(傾斜軸) TRAANG (オプション)

例



- ① X 軸の現在値から少し離れて Z 軸に平行
  - ② プログラム指令された終了位置を通過して傾斜軸に平行
  - ③ 開始位置
  - ④ プランジ切削:移動
  - ⑤ プランジ切削:終了位置
- X ジオメトリ軸  
 Z ジオメトリ軸  
 ZM 機械軸  
 UM 機械軸

プログラムコード	コメント
N...G18	; XZ 平面を選択します
N40 TRAANG (45.0)	; TRAANG 座標変換、角度 = 45°を有効にします。
N50 G7 X40 Z70 F4000	; 開始位置を計算してアプローチします。
N60 G5 X40 F100	; 傾斜軸を終了位置に移動します。
N70 ...	

8.3.4 境界条件

座標変換は、パートプログラムまたは MDI によって選択と選択解除できます。

## 選択と選択解除

- 中間移動ブロックは挿入されません(面取り/半径)。
- 一連のスプラインブロックは終了させてください。
- 工具径補正は選択解除してください。
- 現在のフレームはコントロールシステムで選択解除されます。プログラム指令 G500 に対応します。
- 座標変換が影響を及ぼす軸の有効なワーキングエリアリミットは、コントローラで選択解除されます(プログラム指令 WALIMOF に対応します)。
- 有効な工具長補正が、コントローラによって座標変換に含められます。
- 連続軌跡の制御と丸み付けは中断されます。
- DRF オフセットは、オペレータが削除してください。
- マシンデータ MD24110 \$MC\_TRAFO\_AXES\_IN\_n で指定されたすべての軸は、ブロック単位の基本指令で同期させてください(移動命令なしの POSA...、など)。

## 工具交換

工具は、工具径補正機能が選択解除されているときにのみ交換できます。

## フレーム

ワーク座標系を基準とする命令はすべて、許容されます(FRAME、工具径補正)。ただし、座標変換が機能しないときの手順とは異なり、G91(インクリメンタル指令)によるフレームの変更は特別扱いされません。移動するインクリメンタル値は、どのフレームが前のブロックで有効であったかにかかわらず、新しいフレームのワーク座標系で使用されます。

### 拡張機能

傾斜角座標変換(TRAANG)を選択したり、選択解除したりすると、ジオメトリ軸とチャネル軸の間の割り当てが変更される場合があります。ユーザーは、これらの軸のフレームに対する形状輪郭区間を、傾斜切り込み軸関連の x と z 平面に関して、平行移動、回転、スケーリング、およびミラーリングとして適用できます。

### 参照先

座標変換のためのフレーム補正の詳細については、以下を参照してください。

『機能マニュアル、基本機能』; 「軸、座標系、フレーム (K2)」

### 8.3 傾斜角座標変換(傾斜軸) TRAANG (オプション)

#### 使用できない機能

後述の機能は、座標変換に使用するチャンネル軸では使用できません。

- フィードバック値の設定(PRESETON)
- 突き当て点停止(FXS)
- レファレンス点復帰(G74 または手動のレファレンス点復帰)

#### 速度制御

先読み中に、傾斜角座標変換(TRANSMIT)の速度監視機能が初期設定条件で実行されま  
す。速度監視機能とメインランでの制限は、後述の動作状態で実行されます。

- **AUTOMATIC** モード:座標変換に使用する位置決めまたは揺動軸のプログラミング
- **JOG** モード:JOB モードへの動作モードの変更後

動作モードを **JOG** モードから **AUTOMATIC** モードに変更するなど、メインランへの先  
読みの再原点同期後に、速度監視と制限が先読みで再び実行されます。

---

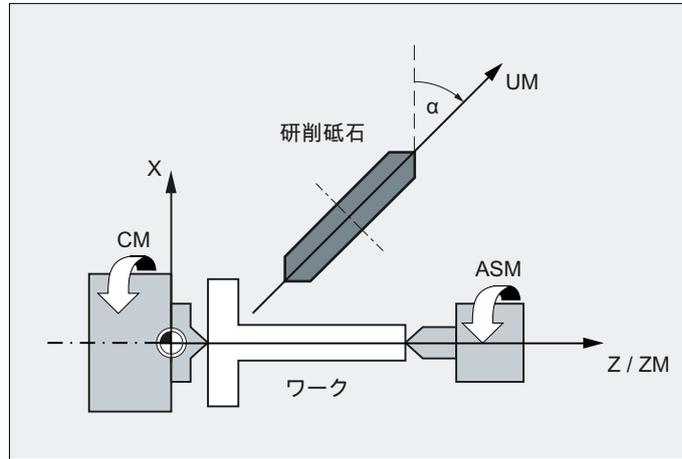
#### 注記

##### プログラム運転:軸固有の送り速度停止の動作

- メインランでの速度監視  
プログラムの実行は、(DBX31, ... DBX4.3 == 1)で軸固有の送り速度停止の座標変換に使用される軸のいずれかに関して、メインランで動作中の速度監視によって停止されます。
  - 先読みでの速度監視  
NCプログラムの実行は、(DBX31, ... DBX4.3 == 1)で軸固有の送り速度停止の座標変換に使用される軸のいずれかに関して、先読みで動作中の速度監視が継続されます。このための必要条件は、関連する軸に移動動作がプログラム指令されていないことです。
-

### 8.3.5 例

例は、下図の軸設定を示します。



- X ジオメトリ軸
- Z ジオメトリ軸
- ZM 機械軸
- UM 機械軸
- $\alpha$  傾斜軸の角度

#### パラメータの割り付け

##### 機械軸名称

- MD10000 \$MN\_AXCONF\_MACHAX\_NAME\_TAB[ 0 ] = 「CM」
- MD10000 \$MN\_AXCONF\_MACHAX\_NAME\_TAB[ 1 ] = 「UM」
- MD10000 \$MN\_AXCONF\_MACHAX\_NAME\_TAB[ 2 ] = 「ZM」
- MD10000 \$MN\_AXCONF\_MACHAX\_NAME\_TAB[ 3 ] = 「ASM」

##### ジオメトリ軸名称

- MD20060 \$MC\_AXCONF\_GEOAX\_NAME\_TAB[ 0 ] = 「X」 (1番目のジオメトリ軸の名称)
- MD20060 \$MC\_AXCONF\_GEOAX\_NAME\_TAB[ 1 ] = 「Y」 (2番目のジオメトリ軸の名称)
- MD20060 \$MC\_AXCONF\_GEOAX\_NAME\_TAB[ 2 ] = 「Z」 (3番目のジオメトリ軸の名称)

8.3 傾斜角座標変換(傾斜軸) TRAANG (オプション)

チャンネル軸名称

- MD20080 \$MC\_AXCONF\_CHANAX\_NAME\_TAB[ 0 ] = 「ZC」
- MD20080 \$MC\_AXCONF\_CHANAX\_NAME\_TAB[ 1 ] = 「CC」
- MD20080 \$MC\_AXCONF\_CHANAX\_NAME\_TAB[ 2 ] = 「ASC」
- MD20080 \$MC\_AXCONF\_CHANAX\_NAME\_TAB[ 3 ] = 「UC」

チャンネル軸へのジオメトリ軸の割り当て

TRAANG が有効でない場合:

- MD20050 \$MC\_AXCONF\_GEOAX\_ASSIGN\_TAB[ 0 ] = 0 (-)
- MD20050 \$MC\_AXCONF\_GEOAX\_ASSIGN\_TAB[ 1 ] = 0 (-)
- MD20050 \$MC\_AXCONF\_GEOAX\_ASSIGN\_TAB[ 2 ] = 1 (3番目のジオメトリ軸 → 1番目のチャンネル軸 ZC)

TRAANG が有効の場合:

- MD24120 \$MC\_TRAFO\_GEOAX\_ASSIGN\_TAB\_1[ 0 ] = 4 (1番目の座標変換ジオメトリ軸 → 4番目のチャンネル軸 UC)
- MD24120 \$MC\_TRAFO\_GEOAX\_ASSIGN\_TAB\_1[ 1 ] = 0 (-)
- MD24120 \$MC\_TRAFO\_GEOAX\_ASSIGN\_TAB\_1[ 2 ] = 1 (3番目の座標変換ジオメトリ軸 → 1番目のチャンネル軸 ZC)

機械軸へのチャンネル軸の割り当て

- MD20070 \$MC\_AXCONF\_MACHAX\_USED[ 0 ] = 3 (1番目のチャンネル軸 → 3番目の機械軸 ZM)
- MD20070 \$MC\_AXCONF\_MACHAX\_USED[ 1 ] = 1 (2番目のチャンネル軸 → 1番目の機械軸 CM)
- MD20070 \$MC\_AXCONF\_MACHAX\_USED[ 2 ] = 4 (3番目のチャンネル軸 → 4番目の機械軸 ASM)
- MD20070 \$MC\_AXCONF\_MACHAX\_USED[ 3 ] = 2 (4番目のチャンネル軸 → 2番目の機械軸 UM)

主軸の定義

- MD35000 \$MA\_SPIND\_ASSIGN\_TO\_MACHAX[ 0 ] = 1 (主軸)
- MD35000 \$MA\_SPIND\_ASSIGN\_TO\_MACHAX[ 1 ] = 0 (軸)
- MD35000 \$MA\_SPIND\_ASSIGN\_TO\_MACHAX[ 2 ] = 0 (軸)
- MD35000 \$MA\_SPIND\_ASSIGN\_TO\_MACHAX[ 3 ] = 2 (主軸)

#### 座標変換タイプ

- MD24100 \$MC\_TRAFO\_TYPE\_1 = 1024 (TRAANG)

#### 直交軸とリアル軸(傾斜)間の角度

- MD24700 \$MC\_TRAANG\_ANGLE\_1 = 45。

#### TRAANG が有効な場合の工具のゼロのジオメトリ軸に対する基本オフセット

- MD24710 \$MC\_TRAANG\_BASE\_TOOL\_1 [0] = 0.0 (1 番目の座標変換ジオメトリ軸に対するオフセット)
- MD24710 \$MC\_TRAANG\_BASE\_TOOL\_1 [1] = 0.0 (2 番目の座標変換ジオメトリ軸に対するオフセット)
- MD24710 \$MC\_TRAANG\_BASE\_TOOL\_1 [2] = 0.0 (3 番目の座標変換ジオメトリ軸に対するオフセット)

#### TRAANG 入力軸

- MD24110 \$MC\_TRAFO\_AXES\_IN\_1[0] = 4 (4 番目のチャンネル軸 UC、傾斜軸)
- MD24110 \$MC\_TRAFO\_AXES\_IN\_1[1] = 1 (1 番目のチャンネル軸 ZC、Z 軸に平行)
- MD24110 \$MC\_TRAFO\_AXES\_IN\_1[2] = 0 (-)

#### プログラミング例

プログラムコード	コメント
N10 G0 G90 Z0 UM=10 G54 F5000 G18 G64 T1 D1	; XZ 平面、工具選択、クランプ補正
N20 TRAANG (45)	; 1 番目の TRAANG データセットと角度 45°で TRAANG を有効にします
N30 G0 Z10 X5	; 開始位置へアプローチします
N40 WAITP (Z)	; Z 軸の移動終了まで待機
N50 OSP1[Z]=10 OSP2[Z]=5 OST1[Z]=-2 OST2[Z]=-2 FA[Z]=5000	; 揺動動作 ; OSP1/OSP2:左右の反転点 ; OST1/OST2:左右の反転点の停止点
N60 OS[Z]=1	; 揺動の起動
N70 POS[X]=4.5 FA[X]=50	; X 軸を平行に位置決め
N80 OS[Z]=0	; 揺動の解除
N90 WAITP (Z)	; Z 軸の移動終了まで待機
N100 TRAFOOF	; 座標変換の解除
N110 G0 Z10 UM=10	; 後退
N120 M30	

## 8.4 座標変換重畳

## 8.4 座標変換重畳

## 8.4.1 機能

## 8.4.1.1 はじめに

座標変換重畳では、**2つの**座標変換を連続して接続(重畳)できます。その結果、**1番目**の変換から軸の移動成分を選択し、**2番目**の変換の入力データとして使用できます。これで、**2番目**の座標変換の軸の移動成分が機械軸に作用します。

- **1番目**の変換には、以下を使用できます。
    - ユニバーサルフライスヘッドを含む TRAORI 方向座標変換
- 参照先:  
機能マニュアル 応用機能; 多軸座標変換(F2)
- 面端座標変換 TRANSMIT
  - 円筒補間 TRACYL
- **2番目**の変換として使用できるのは、以下のみです。
    - 傾斜軸 TRAANG

## 注記

## セットアップ時の例外

座標変換重畳は、通常、連続して重畳された**2つ**の変換から成っています。試験の目的のため、セットアップ時などに、重畳リストに**1つ**の座標変換のみを入力することができます。

## 注記

## 異なる軸名称

動作中の座標変換では、関連する機械、チャンネル、およびジオメトリ軸の名称は異なります。

- MD10000 \$MN\_AXCONF\_MACHAX\_NAME\_TAB (機械軸名称)
- MD20080 \$MC\_AXCONF\_CHANAX\_NAME\_TAB (チャンネル軸名称)
- MD20060 \$MC\_AXCONF\_GEOAX\_NAME\_TAB (ジオメトリ軸名称)

## 用途例

- 工具研削など、傾斜研削砥石を使用して円筒側面上の線としてプログラム指令された研削輪郭(TRACYL)。
- 傾斜研削砥石を使用して TRANSMIT で生成された、円形以外の輪郭の仕上げ研削。

## 軸構成

座標変換重畳には、次の構成方法が必要です。

- ジオメトリ軸への名称の割り当て
- チャネル軸への名称の割り当て
- チャネル軸へのジオメトリ軸の割り当て
  - 一般的な状態(座標変換は有効でない)
- 機械軸番号へのチャネル軸の割り当て
- 軸の主軸、回転、モジュロの定義
- 機械軸名称の割り当て
- 座標変換毎の設定(個々の変換用と**重畳された**座標変換用)
  - 座標変換タイプ
  - 座標変換軸
  - 有効な座標変換中のチャネル軸へのジオメトリ軸の割り当て
  - 座標変換、座標系の回転位置、元の座標系基準の工具原点、傾斜軸の角度などに応じて、設定が異なります。

---

### 注記

#### 二次的条件と特別な事例

個々の座標変換の説明で指定された補足条件と特別な事例は、座標変換重畳で使用される場合も守ってください。

---

## 境界条件

### 各チャネルの座標変換データブロック

各チャネルには、複数の座標変換データブロックを定義できます。座標変換の対応するマシンデータの名前は、\$MC\_TRAFO... and end with...\_n で始まります。ここで、n = 1, 2, 3, ...です。

### 座標変換重畳の数

チャネルの座標変換のうち、最大で**2つの重畳された**座標変換を定義できます。

### 重畳方向

座標変換重畳の1番目の座標変換には、入力として基本座標系(BCS)があります。

座標変換重畳の2番目の座標変換には、出力として機械座標系(MCS)があります。

## 8.4 座標変換重畳

## 工具データ

工具は常に、座標変換重畳の **1 番目**の座標変換に割り当てられます。2 番目の座標変換は、有効な工具長がゼロとして動作します。

**1 番目**の座標変換重畳で有効なのは、マシンデータ(`_BASE_TOOL_`)で設定された基本工具長のみです。

## 8.4.1.2 システム変数

次の変換別システム変数を使用できます。

システム変数	意味
<code>\$AA_ITR</code>	n 番目の座標変換出力の現在の指令値
<code>\$AA_IBC</code>	直交軸の現在の指令値
<code>\$VA_ITR</code>	n 番目の座標変換出力の現在のフィードバック値
<code>\$VA_IBC</code>	軸の現在の直交座標 <code>BCS</code> のエンコーダ位置
<code>\$VA_IW</code>	軸の現在の <code>WCS</code> フィードバック値
<code>\$VA_IB</code>	軸の現在の <code>BCS</code> エンコーダ位置

## 境界条件

- ウォーム再起動  
エンコーダ位置では、原点確立未完軸の値は **0** です。`$VA` 変数では、エンコーダフィードバック値が適切に逆変換されます。
- チャネルリセット  
チャネルのリセット後に、有効な座標変換が変わる場合があります。これは、システム変数の値に直接、影響を及ぼす場合があります。  
チャネルのリセット後に再度有効になる座標変換は、短時間の間、無効になった後、再び有効になります。これは、位置の値に影響を及ぼします。変数の値は変わる場合があります。  
次の変数:  
**`$AC_STAT == 0`**  
によって、シンクロナイズドアクションでこの状態を参照できます。

### \$AA\_ITR:n 番目の座標変換出力の現在の指令値

システム変数\$AA\_ITR[<軸>, <座標変換レイヤ>]により、n 番目の座標変換重畳の出力での軸の指令位置が決定されます。

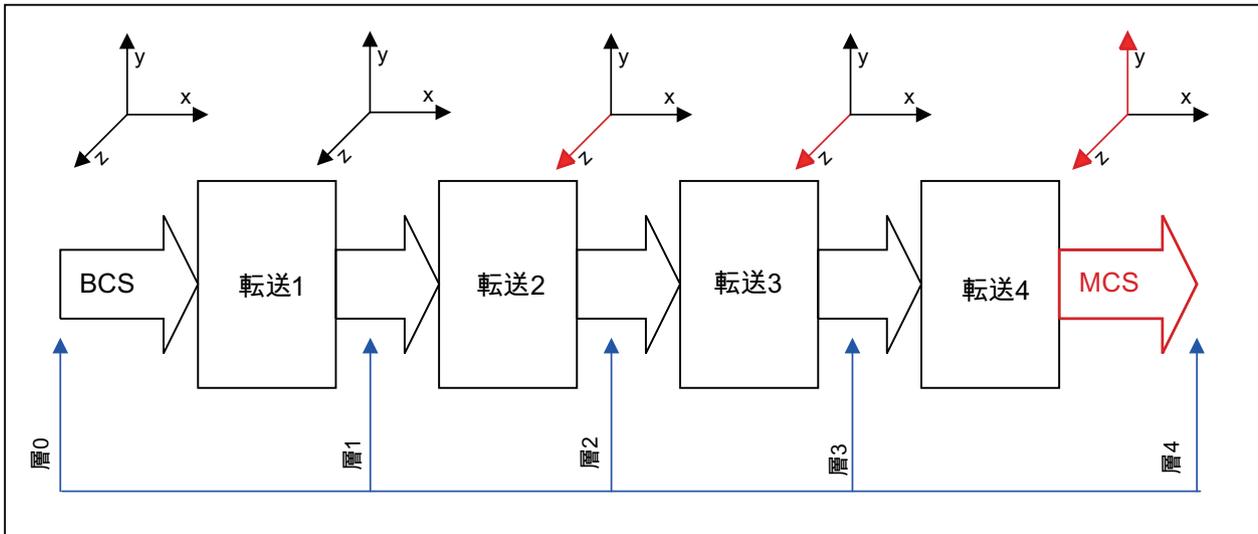


図 8-15 座標変換レイヤ

#### 軸

変数の 1 番目のインデックスとして、ジオメトリ軸、チャンネル軸、または機械軸の名称を使用できます。ジオメトリ軸名称のプログラミング中に各座標変換レイヤでは、座標変換レイヤ 0 に対応するジオメトリ軸へのチャンネル軸の割り当てが有効です。つまり、BCS 側でのジオメトリ軸の割り当ては、すべての平面で有効です。

ジオメトリ軸名称を使用する意味があるのは、ジオメトリ軸が切り替えられていない場合のみです。それ以外の場合は、常にチャンネル軸名称を使用した方が効果的です。

#### 座標変換レイヤ

システム変数の 2 番目のインデックスは、位置が取り出される座標変換レイヤに対応します。

- 座標変換レイヤ 0:位置が BCS 位置に対応、 $\$AA\_ITR[x,0] == \$AA\_IB[x]$
- 座標変換レイヤ 1:1 番目の座標変換出力の指令位置
- 座標変換レイヤ 2:2 番目の座標変換出力の指令位置
- 座標変換レイヤ 3:3 番目の座標変換出力の指令位置
- 座標変換レイヤ 4:4 番目の座標変換出力の指令位置、 $\$AA\_ITR[x,4] == \$AA\_IM[x]$

## 8.4 座標変換重畳

一連の座標変換重畳で1つ以上の座標変換が欠けている場合は、最上位層が同じ値を提供し続けます。たとえば、座標変換3と座標変換4が欠けている場合は、以下が適用されます。

$$\$AA\_ITR[x,2] = \$AA\_ITR[x,3] = \$AA\_ITR[x,4] = \$AA\_IM[x]$$

座標変換が、TRAFOOFまたはチャンネルのリセットにより無効になった場合は、レイヤ0～4が結合します。そして、システム変数がBCS値(レイヤ0)を提供します。

**\$AA\_IBC:直交軸の現在の指令値**

システム変数\$AA\_IBC[<軸>]により、BCSとMCSの間にある直交軸の指令位置が決定されます。n番目の座標変換出力で軸が直交軸の場合は、この出力値が提供されます。すべての座標変換出力で、対応する軸に直交軸でないものがある場合は、軸のすべてのBCSオフセットを含むBCS値が決められます。

TRACONが直交軸として軸に応答した場合は、そのMCS値が提供されます。使用された軸名称は、ジオメトリ、チャンネル、または機械軸の名称です。

**\$VA\_ITR:n番目の座標変換出力の現在値**

システム変数\$VA\_ITR[<軸>, <座標変換レイヤ>]により、n番目の座標変換重畳の出力での軸の指令位置が決定されます。

**\$VA\_IBC:軸の現在の直交座標BCSのエンコーダ位置**

システム変数\$VA\_IBC[<軸>]により、BCSとMCSの間にある直交軸のエンコーダ位置が決定されます。使用された軸名称は、ジオメトリ、チャンネル、または機械軸の名称です。

n番目の座標変換出力で軸が直交軸の場合は、この出力値が提供されます。すべての座標変換出力の対応軸に直交軸でないものがある場合は、軸のBCS値になります。

**\$VA\_IW:軸の現在のWCSフィードバック値**

システム変数\$VA\_IW[<軸>]により、WCSの軸の逆変換されたエンコーダ位置が決定されます。WCS値には、すべての軸関連の重複部分(DRF、AA\_OFF、外部ゼロオフセットなど)とオフセット値(CECなど)が含まれます。

### \$VA\_IB:軸の現在の BCS エンコーダ位置

システム変数\$VA\_IB[<軸>]により、BCS の軸の逆変換されたエンコーダ位置が決定されます。BCS 値には、すべての軸関連の重複部分(DRF、AA\_OFF、外部ゼロオフセットなど)とオフセット値(CEC など)が含まれます。

---

#### 注記

##### \$VA\_ITR\$, VA\_IBC, \$VA\_IW, \$VA\_IB

変数の値は、IPO サイクル内での変数の読み取り中には変更されませんが、フィードバック値は変更されている場合があります。

動作中の座標変換では、IPO サイクルでフィードバック値を BCS へ座標変換により、多くの時間を消費する可能性があることを考慮してください。この場合は、適切な IPO サイクルを設定してください。

---

## 8.4.2 プログラミング

TRAANG 座標変換は、TRAANG 命令を使用してパートプログラムまたはシンクロナイズドアクションで有効になります。

### 構文

```
TRACON (<座標変換番号>,<パラメータ_1>,<パラメータ_2>,...)  
...  
TRAFOOF
```

8.4 座標変換重畳

意味

TRACON:	座標変換重畳を有効にします 別の座標変換が以前に有効であった場合、その変換は TRACON () によって自動的に解除されます。			
<座標変換番号 >:	座標変換重畳の番号:			
	タイプ:	INT		
	データ範囲:	0 ... 2		
	値:	0, 1	1 番目/座標変換重畳のみ	
		2	2 番目の座標変換重畳	
指定なし		0 または 1 と同じ意味		
注:		0、1、2 以外の値を指定すると、エラーメッセージが生成されます。		
<パラメータ _1>, <パラメータ _2, ...:	座標変換重畳のパラメータ(例: 傾斜軸の角度) パラメータが設定されていない場合は、初期設定または最後に使用されたパラメータが有効となります。 以前のパラメータで初期設定が有効な場合は、指定したパラメータが確実に目的の順序で使用されるよう、コンマを使用してください。特に、<座標変換番号>の指定が不要な場合でも、1 つ以上のパラメータの前にはコンマが必要です。例: TRACON ( , 3.7) <b>TRANSMIT</b> または <b>TRAORI</b> による <b>TRACON</b> (5 番目の軸の加工) では、2 番目の パラメータ <b>Par_2</b> は、傾斜軸の角度として作用しません。 <b>TRACYL</b> による <b>TRACON</b> (円筒面の加工) では、 <b>Par</b> がユニットの直径として使用されます。			
TRAFOOF:	最後に有効にした座標変換(重畳)を解除します。			

例

プログラムコード	コメント
...	
N230 TRACON(1,45.)	; 1 番目の座標変換重畳を有効にします。 ; 以前に有効であった座標変換は自動的に選択解除されます ; 傾斜軸のパラメータは 45°です。
...	

プログラムコード	コメント
N330 TRACON (2, 40.)	; 2 番目の座標変換重畳を有効にします。 ; 傾斜軸のパラメータは 40°です。
...	
N380 TRAF00F	; 2 番目の座標変換重畳を解除します。
...	

### 8.4.3 例

#### 8.4.3.1 座標変換重畳の用途例

以下の例は、次のものを示します。

- 一般的なチャンネル設定
- 単独の座標変換
- 以前に定義された単独の座標変換から成る座標変換重畳
- 単独の座標変換の適用
- 座標変換重畳の適用

以下の座標変換は、チャンネルで定義します。

データセット	座標変換
1	TRAORI 回転工具と軸シーケンス AB による 5 軸座標変換(座標変換タイプ 16)
2	TRANSMIT 円筒補間の終了(座標変換タイプ 256)
3	TRAANG 角度座標変換(座標変換タイプ 1024)
4	TRAORI + TRAANG 1 番目の座標変換重畳(座標変換タイプ 8192)
5	TRANSMIT + TRAANG 2 番目の座標変換重畳(座標変換タイプ 8192)

## 8.4 座標変換重畳

## パラメータの割り付け

## 一般的なチャネル設定

```

CHANDATA (1) ;チャネル1のチャネルデータ
MD20070 $MC_AXCONF_MACHAX_USED[0]=1
MD20070 $MC_AXCONF_MACHAX_USED[1]=2
MD20070 $MC_AXCONF_MACHAX_USED[2] = 3
MD20070 $MC_AXCONF_MACHAX_USED[3] = 4
MD20070 $MC_AXCONF_MACHAX_USED[4]=5
MD20070 $MC_AXCONF_MACHAX_USED[5]=6
MD20070 $MC_AXCONF_MACHAX_USED[6]=7
MD20070 $MC_AXCONF_MACHAX_USED[7] = 0
MD20080 $MC_AXCONF_CHANAX_NAME_TAB[3]=「A」
MD20080 $MC_AXCONF_CHANAX_NAME_TAB[4]=「B」
MD20080 $MC_AXCONF_CHANAX_NAME_TAB[5] = 「C」
MD36902 $MA_IS_ROT_AX[ AX4 ] = TRUE
MD36902 $MA_IS_ROT_AX[ AX5 ] = TRUE
MD36902 $MA_IS_ROT_AX[ AX6 ] = TRUE
MD36902 $MA_IS_ROT_AX[ AX7 ] = TRUE
MD35000 $MA_SPIND_ASSIGN_TO_MACHAX[AX5]= 0
MD35000 $MA_SPIND_ASSIGN_TO_MACHAX[AX7] = 1
MD35000 $MA_ROT_IS_MODULO[AX7] = TRUE

```

## 単独の座標変換

```

; 1番目のTRAORI
MD24470 $MC_TRAFO_TYPE_1= 16 ; TRAORI:A-Bキネマティックス
MD24410 $MC_TRAFO_AXES_IN_1[0]=1
MD24410 $MC_TRAFO_AXES_IN_1[1]=2
MD24410 $MC_TRAFO_AXES_IN_1[2]=3
MD24410 $MC_TRAFO_AXES_IN_1[3]=4
MD24410 $MC_TRAFO_AXES_IN_1[4]=5
MD24410 $MC_TRAFO_AXES_IN_1[5]=0
MD24120$MC_TRAFO_GEOAX_ASSIGN_TAB_1[0]=1
MD24120$MC_TRAFO_GEOAX_ASSIGN_TAB_1[1]=2
MD24120$MC_TRAFO_GEOAX_ASSIGN_TAB_1[2]=3
MD24550$MC_TRAFO5_BASE_TOOL_1[0]=0
MD24550$MC_TRAFO5_BASE_TOOL_1[1]=0
MD24550$MC_TRAFO5_BASE_TOOL_1[2]=0

; 2番目のTRANSMIT
MD24200 $MC_TRAFO_TYPE_2 = 256 ; TRANSMIT
MD24210 $MC_TRAFO_AXES_IN_2[0] = 1
MD24210 $MC_TRAFO_AXES_IN_2[1] = 6

```

```

MD24210 $MC_TRAFO_AXES_IN_2[2]=3
MD24210 $MC_TRAFO_AXES_IN_2[3] = 0
MD24210 $MC_TRAFO_AXES_IN_2[4] = 0
MD24210 $MC_TRAFO_AXES_IN_2[5] = 0
MD24210 $MC_TRAFO_AXES_IN_2[6]=0
MD24220 $MC_TRAFO_GEOAX_ASSIGN_TAB_2[0] =1
MD24220 $MC_TRAFO_GEOAX_ASSIGN_TAB_2[1] =6
MD24220 $MC_TRAFO_GEOAX_ASSIGN_TAB_2[2] =3

```

```

; 3番目の TRAANG
MD24300 $MC_TRAFO_TYPE_3 = 1024 ; TRAANG
MD24310 $MC_TRAFO_AXES_IN_3[0] = 1
MD24310 $MC_TRAFO_AXES_IN_3[1] = 3
MD24310 $MC_TRAFO_AXES_IN_3[2] = 2
MD24310 $MC_TRAFO_AXES_IN_3[3] = 0
MD24310 $MC_TRAFO_AXES_IN_3[4] = 0
MD24320 $MC_TRAFO_GEOAX_ASSIGN_TAB_3[0] =1
MD24320 $MC_TRAFO_GEOAX_ASSIGN_TAB_3[1] =3
MD24320 $MC_TRAFO_GEOAX_ASSIGN_TAB_3[2] =2
MD24700 $MC_TRAANG_ANGLE_1 = 45。
MD24720 $MC_TRAANG_PARALLEL_VELO_RES_1 = 0.2
MD24721 $MC_TRAANG_PARALLEL_ACCEL_RES_1 = 0.2
MD24710 $MC_TRAANG_BASE_TOOL_1 [0] = 0.0
MD24710 $MC_TRAANG_BASE_TOOL_1 [1] = 0.0
MD24710 $MC_TRAANG_BASE_TOOL_1 [2] = 0.0

```

#### 座標変換重畳

```

; 4番目の TRACON (TRAORI と TRAANG の重畳)
MD24400 $MC_TRAFO_TYPE_4 = 8192
MD24420 $MC_TRAFO_GEOAX_ASSIGN_TAB_4[0] =2
MD24420 $MC_TRAFO_GEOAX_ASSIGN_TAB_4[1] =1
MD24420 $MC_TRAFO_GEOAX_ASSIGN_TAB_4[2] =3
MD24995 $MC_TRACON_CHAIN_1[0] = 1
MD24995 $MC_TRACON_CHAIN_1[1] = 3
MD24995 $MC_TRACON_CHAIN_1[2] = 0

```

```

; 5番目の TRACON (TRANSMIT と TRAANG の重畳)
MD24430 $MC_TRAFO_TYPE_5 = 8192
MD24434 $MC_TRAFO_GEOAX_ASSIGN_TAB_5[0] =1
MD24434 $MC_TRAFO_GEOAX_ASSIGN_TAB_5[1] =6
MD24434 $MC_TRAFO_GEOAX_ASSIGN_TAB_5[2] =3

```

8.4 座標変換重畳

```
MD24996 $MC_TRACON_CHAIN_2[0] = 2
MD24996 $MC_TRACON_CHAIN_2[1] = 3
MD24996 $MC_TRACON_CHAIN_2[2]=0
```

プログラミング例

前提条件

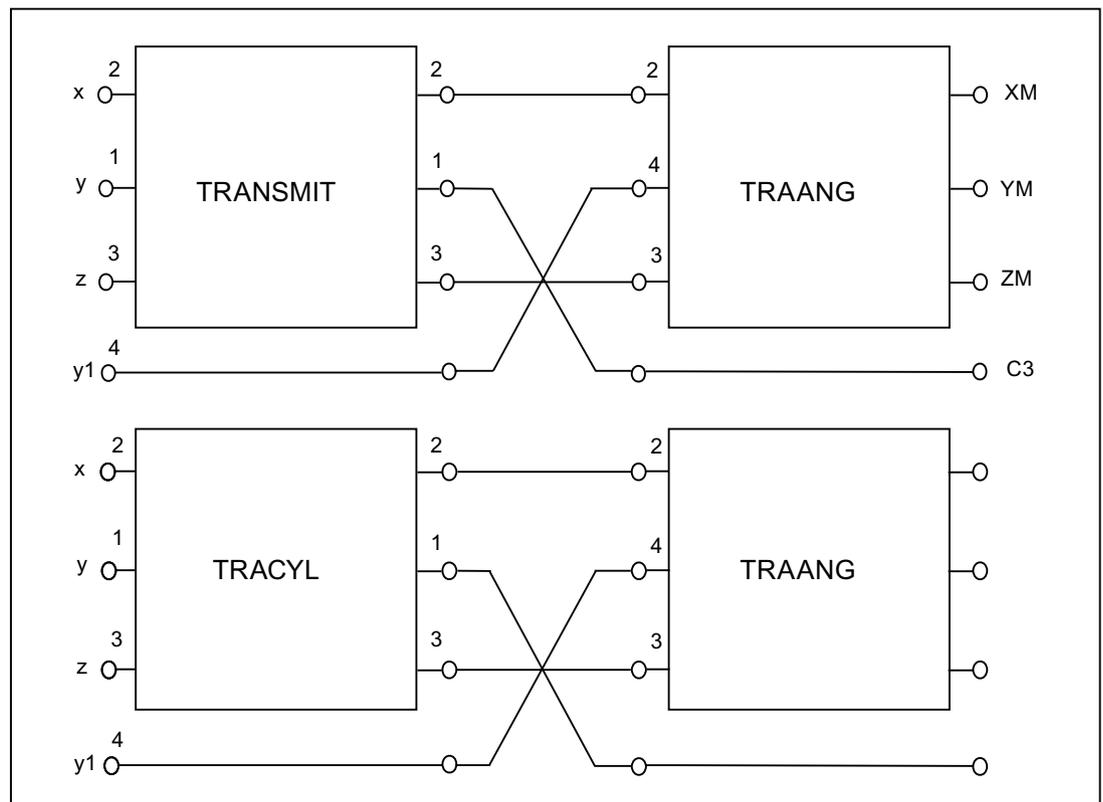
以下のプログラミング例では、機械で「傾斜軸」の角度を設定することができ、単独の座標変換が有効な場合は、0°に設定されていることを前提にしています。

プログラムコード	コメント
<b>; 工具指定</b>	
\$TC_DP1[1,1]=120	; 工具タイプ
\$TC_DP3[1,1]=10	; 工具長
N2 X0 Y0 Z0 A0 B0 F20000 T1 D1	
N4 X20	
<b>; 単独の座標変換を呼び出し</b>	
N30 TRANSMIT	; TRANSMIT を有効化
N40 X0 Y20	
N50 X-20 Y0	
N60 X0 Y-20	
N70 X20 Y0	
N80 TRAFOOF	; TRANSMIT の解除
N130 TRAANG(45.)	; 傾斜軸座標変換を起動します。 ; パラメータ:角度 45°
N140 X0 Y0 Z20	
N150 X-20 Z0	
N160 X0 Z-20	
N170 X20 Z0	
...	
<b>; 座標変換重畳の起動</b>	
N230 TRACON(1,45.)	; 1 番目の座標変換重畳 ; 起動(TRAORI + TRAANG)。 ; 以前に有効であった座標変換は ; 自動的に選択解除されます ; 傾斜軸のパラメータは 45°です。
N240 X10 Y0 Z0 A3=-1 C3=1 ORIWKS	
N250 X10 Y20 B3=1 C3=1	
...	
N330 TRACON(2,40.)	; 2 番目の座標変換重畳 ; (TRANSMIT + TRAANG)を有効にします。 ; 傾斜軸のパラメータは ; 40°です。
N335 X20 Y0 Z0	

プログラムコード	コメント
N340 X0 Y20 Z10	
N350 X-20 Y0 Z0	
N360 X0 Y-20 Z0	
N370 X20 Y0 Z0	
N380 TRAFOOF	; 2番目の座標変換重畳を ; 無効にします。
...	
N1000 M30	

### 8.4.3.2 座標変換重畳の軸位置の特定

次の例では2個の座標変換重畳が設定され、シンクロナイズドアクションのなかで軸位置を決めるために、システム変数が、パートプログラムで繰り返し読み込まれます。



### パラメータの割り付け

CHANDATA (1)

## 8.4 座標変換重畳

```
; TRANSMIT
MD24100 $MC_TRAFO_TYPE_1=256
MD24110 $MC_TRAFO_AXES_IN_1[0] = 2
MD24110 $MC_TRAFO_AXES_IN_1[1] = 1
MD24110 $MC_TRAFO_AXES_IN_1[2] = 3
MD24120 $MC_TRAFO_GEOAX_ASSIGN_TAB_1 [0] = 2
MD24120 $MC_TRAFO_GEOAX_ASSIGN_TAB_1 [1] = 1
MD24120 $MC_TRAFO_GEOAX_ASSIGN_TAB_1 [2] = 3

; TRACYL
MD24200 $MC_TRAFO_TYPE_2=512
MD24210 $MC_TRAFO_AXES_IN_2[0]=2
MD24210 $MC_TRAFO_AXES_IN_2[1]=1
MD24210 $MC_TRAFO_AXES_IN_2[2]=3
MD24220 $MC_TRAFO_GEOAX_ASSIGN_TAB_2[0] =2
MD24220 $MC_TRAFO_GEOAX_ASSIGN_TAB_2[1] =1
MD24220 $MC_TRAFO_GEOAX_ASSIGN_TAB_2[2] =3

; TRAANG
MD24300 $MC_TRAFO_TYPE_3=1024
MD24310 $MC_TRAFO_AXES_IN_3[0] = 2
MD24310 $MC_TRAFO_AXES_IN_3[1] = 4
MD24310 $MC_TRAFO_AXES_IN_3[2] = 3
MD24320 $MC_TRAFO_GEOAX_ASSIGN_TAB_3[0] =2
MD24320 $MC_TRAFO_GEOAX_ASSIGN_TAB_3[1] =4
MD24320 $MC_TRAFO_GEOAX_ASSIGN_TAB_3[2] =3
MD24700 $MC_TRAANG_ANGLE_1 = 45。
MD24720 $MC_TRAANG_PARALLEL_VELO_RES_1 = 0.2
MD24721 $MC_TRAANG_PARALLEL_ACCEL_RES_1 = 0.2
MD24710 $MC_TRAANG_BASE_TOOL_1 [0] = 0.0
MD24710 $MC_TRAANG_BASE_TOOL_1 [1] = 0.0
MD24710 $MC_TRAANG_BASE_TOOL_1 [2] = 0.0

; 1 番目の TRANSMIT / TRAANG 重畳
MD24400 $MC_TRAFO_TYPE_4=8192 ; TRACON (1)
MD24995 $MC_TRACON_CHAIN_1[0] = 1
MD24995 $MC_TRACON_CHAIN_1[1] = 3
MD24995 $MC_TRACON_CHAIN_1[2] = 0
MD24995 $MC_TRACON_CHAIN_1[3] = 0
MD24410 $MC_TRAFO_AXES_IN_4[0]=1
MD24410 $MC_TRAFO_AXES_IN_4[1]=2
MD24410 $MC_TRAFO_AXES_IN_4[2]=3
MD24420 $MC_TRAFO_GEOAX_ASSIGN_TAB_4[0] =2
MD24420 $MC_TRAFO_GEOAX_ASSIGN_TAB_4[1] =1
MD24420 $MC_TRAFO_GEOAX_ASSIGN_TAB_4[2] =3
```

```

; 2 番目の TRACYL / TRAANG 重畳
MD24430 $MC_TRAFO_TYPE_5=8192 ; TRACON (2)
MD24996 $MC_TRACON_CHAIN_2[0] = 2
MD24996 $MC_TRACON_CHAIN_2[1]=3
MD24996 $MC_TRACON_CHAIN_2[2]=0
MD24996 $MC_TRACON_CHAIN_2[3]=0
MD24432 $MC_TRAFO_AXES_IN_5[0]=1
MD24432 $MC_TRAFO_AXES_IN_5[1]=2
MD24432 $MC_TRAFO_AXES_IN_5[2]=3
MD24434 $MC_TRAFO_GEOAX_ASSIGN_TAB_5[0] =2
MD24434 $MC_TRAFO_GEOAX_ASSIGN_TAB_5[1] =1
MD24434 $MC_TRAFO_GEOAX_ASSIGN_TAB_5[2] =3
M17

```

### プログラミング例

プログラムコード	コメント
N10 \$TC_DP1[1,1]=120	
N20 \$TC_DP3[1,1]= 20	
N30 \$TC_DP4[1,1]=0	
N40 \$TC_DP5[1,1]=0	
N50	
N60 X0 Y0 Z0 F20000 T1 D1	
N70	
N80	; シンクロナイズドアクションで繰り返し読み込み
N90 ID=1 WHENEVER TRUE DO \$R0=\$AA_ITR[X,0] \$R1=\$AA_ITR[X,1] \$R2=\$AA_ITR[X,2]	
N100 ID=2 WHENEVER TRUE DO \$R3=\$AA_IBC[X] \$R4=\$AA_IBC[Y] \$R5=\$AA_IBC[Z]	
N110 ID=3 WHENEVER TRUE DO \$R6=\$VA_IW[X]-\$AA_IW[X]	
N120 ID=4 WHENEVER TRUE DO \$R7=\$VA_IB[X]-\$AA_IB[X]	
N130 ID=5 WHENEVER TRUE DO \$R8=\$VA_IBC[X]-\$AA_IBC[X]	
N140 ID=6 WHENEVER TRUE DO \$R9=\$VA_ITR[X,1]-\$AA_ITR[X,1]	
N150	
N160	
N170 TRACON(1,)	; 1 番目の TRANSMIT / TRAANG 重畳
N180 X20 Y0 Z0	
N190 X0 Y20 Z10	
N200 X-20 Y0 Z0	
N210 X0 Y-20 Z0	
N220 X20 Y0 Z0	
N230 TRAFOOF	
N240	
N250	
N260 TRACON(2, 40.)	; 2 番目の TRACYL / TRAANG 重畳
N270 X20 Y0 Z0	
N280 X0 Y20 Z10	

## 8.5 モーダル座標変換

プログラムコード	コメント
N290 X-20 Y0 Z0	
N300 X0 Y-20 Z0	
N310 X20 Y0 Z0	
N320 TRAF00F	
N330	
N340 M30	

## 8.5 モーダル座標変換

## 機能

モーダル座標変換は常に有効であり、指定によって選択された他の座標変換に対して関連的作用があります。

選択された座標変換は、モーダル座標変換関連の 1 番目の座標変換重畳として計算されます。

モーダル座標変換に対して選択される TRANSMIT などの座標変換は、TRACON を使用してモーダル座標変換とともに NC プログラムで重畳されます。その後、TRANSMIT などの座標変換重畳が NC プログラムでプログラム指令されます。

## 参照先:

『プログラミングマニュアル、上級編』、「座標変換」>「座標変換重畳の起動(TRAANG)」

## 選択と選択解除

モーダル座標変換は、次のマシンデータによって選択されます。

- MD20144 \$MC\_TRAFO\_MODE\_MASK、ビット 0 = 1
- MD20144 \$MC\_TRAFO\_RESET\_VALUE は、モーダル座標変換を定義します。
- MD20140 \$MC\_TRAFO\_RESET\_VALUE= モーダル座標変換の座標変換データ設定数

#### その他のマシンデータ

- MD20110 \$MC\_RESET\_MODE\_MASK
  - ビット 0 = 1 (ビット 7 が評価されます)
  - ビット 7 = 0 (MD20140 \$MC\_TRAFO\_RESET\_VALUE によって座標変換データ設定が決定されます)
- MD20112 \$MC\_START\_MODE\_MASK (MD20140 \$MC\_TRAFO\_RESET\_VALUE)
- MD20118 \$MC\_GEOAX\_CHANGE\_RESET= TRUE (すなわち、ジオメトリ軸がリセットされます)

TRAFOOF の場合は、動作中の座標変換重畳 TRACON が選択解除され、モーダル座標変換が自動的に選択されます。

## HMI 操作への影響

モーダル座標変換では座標変換が常に有効であるため、座標変換の選択と選択解除に従って、HMI 操作画面が適応します。

#### 座標変換重畳 TRACON

HMI 操作画面は、TRACON ではなく、TRACON の 1 番目の座標変換重畳(TRANSMIT など)を表示します。これに従い、1 番目の座標変換重畳の座標変換タイプがシステム変数、つまり、\$P\_TRAFO または \$AC\_TRAFO によって返されます。そして、TRANSMIT が使用する周期を直接使用できるようになります。

#### TRAFOOF 座標変換の無効化

TRAFOOF プログラミング命令に従い、ユーザーインタフェース上の G 命令リストには、座標変換は表示されません。システム変数 \$P\_TRAFO および \$AC\_TRAFO が値 0 を提供します。モーダル座標変換が有効であり、BCS および MCS 座標系は異なります。表示された MCS 位置は、常に実際の機械軸を示します。

## システム変数

次のシステム変数は、有効な座標変換または座標変換重畳の座標変換タイプを読み込みます。

- \$AC\_TRAFO (有効な座標変換)
- \$AC\_TRAFO\_CHAIN[0] (有効な座標変換重畳)
- \$P\_TRAFO (プログラム指令された座標変換)
- \$P\_TRAFO\_CHAIN[0] (プログラム指令された座標変換重畳)

## 8.5 モーダル座標変換

## フレーム

TRACON の選択と選択解除のフレーム調整は、1 番目の座標変換重畳だけが存在するとみなして実行されます。実軸上の座標変換は、TRAANG が選択されると、無効になります。

## JOG

モーダル座標変換は、JOG の移動時は、そのまま有効です。

## 補足条件

モーダル座標変換では、NC の基本の運転順序は変更されません。有効な座標変換に適用されるすべての制限も、引き続き適用されます。

- リセットの場合、既存の座標変換が完全に選択解除され、モーダル座標変換が再び選択されます。
- 原点確立時には、座標変換が選択解除され、その後、チャンネルリセットまたは NC スタートでモーダル座標変換を再選択する必要があります。

## 例

追加の傾斜 Y 軸を持つ旋盤の場合は、傾斜軸の座標変換は機械構成の要素に含まれません。したがって、プログラマはこの座標変換を考慮する必要はありません。TRACYL または TRANSMIT の場合は、座標変換が選択されているため、TRAANG を含めてください。

プログラム指令された座標変換を無効にするときは、TRAANG が自動的に再び有効になります。TRACYL または TRANSMIT は、ユーザーインターフェースで対応して表示されません。

機械の組み合わせ:X1 に対して傾斜し、Z1 に垂直な Y1 軸を持つ旋盤

## マシンデータのパラメータ設定

## プログラムコード

```
CANDATA (1)
; 座標変換のないキネマティクス
MD20080 $MC_AXCONF_CHANAX_NAME_TAB[1]=「Y2」
MD20050 $MC_AXCONF_GEOAX_ASSIGN_TAB[0] = 1
MD20050 $MC_AXCONF_GEOAX_ASSIGN_TAB[1] = 0
MD20050 $MC_AXCONF_GEOAX_ASSIGN_TAB[2] = 3
```

## プログラムコード

```

; TRAANG のデータ
MD24100 $MC_TRAFO_TYP_1 = 1024; TRAANG、X1 に対して傾斜し、Z1 に垂直な Y1 軸
MD24110 $MC_TRAFO_AXES_IN_1[0] = 2
MD24110 $MC_TRAFO_AXES_IN_1[1] = 1
MD24110 $MC_TRAFO_AXES_IN_1[2] = 3
MD24110 $MC_TRAFO_AXES_IN_1[3] = 0
MD24110 $MC_TRAFO_AXES_IN_1[4] = 0
MD24120 $MC_TRAFO_GEOAX_ASSIGN_TAB_1[0] =1

; モーダル座標変換の定義
MD20144 $MC_TRAFO_MODE_MASK = 1
MD20140 $MC_TRAFO_RESET_VALVUE= 1
MD20110 $MC_RESET_MODE_MASK = 'H01'
MD20112 $MC_START_MODE_MASK = 'H80'
MD20140 $MC_TRAFO_RESET_VALUE
MD20118 $MC_GEOAX_CHANGE_RESET= TRUE

; TRANSMIT、TRACYL のデータ
MD24911 $MC_TRANSMIT_POLE_SIDE_FIX_1 = 1 ; 同じく 2 の場合 (アラーム 21617 が発生
します)
MD24200 $MC_TRAFO_TYP_2 = 257 ; TRANSMIT
MD24210 $MC_TRAFO_AXES_IN_2[0] = 1
MD24210 $MC_TRAFO_AXES_IN_2[1] = 4
MD24210 $MC_TRAFO_AXES_IN_2[2]=3
MD24210 $MC_TRAFO_AXES_IN_2[3] = 0
MD24210 $MC_TRAFO_AXES_IN_2[4] = 0
MD24220 $MC_TRAFO_GEOAX_ASSIGN_TAB_2[0] =1
MD24220 $MC_TRAFO_GEOAX_ASSIGN_TAB_2[1] =4
MD24220 $MC_TRAFO_GEOAX_ASSIGN_TAB_2[2] =3

MD24300 $MC_TRAFO_TYP_3 = 514 ; TRACYL
MD24310 $MC_TRAFO_AXES_IN_3[0] = 1
MD24310 $MC_TRAFO_AXES_IN_3[1]=4
MD24310 $MC_TRAFO_AXES_IN_3[2] = 3
MD24310 $MC_TRAFO_AXES_IN_3[3] = 0
MD24310 $MC_TRAFO_AXES_IN_3[4] = 0
MD24320 $MC_TRAFO_GEOAX_ASSIGN_TAB_3[0] =1
MD24320 $MC_TRAFO_GEOAX_ASSIGN_TAB_3[1] =4
MD24320 $MC_TRAFO_GEOAX_ASSIGN_TAB_3[2] =3

; TRACON のデータ
; TRACON は TRANSMIT 257/TRAANG を重畳 (X1 に対して傾斜した Y1 軸)
MD24400 $MC_TRAFO_TYP_4 = 8192 ; TRACON
MD24995 $MC_TRACON_CHAIN_1[0] = 3
MD24995 $MC_TRACON_CHAIN_1[1] = 1
MD24995 $MC_TRACON_CHAIN_1[2] = 0
MD24420 $MC_TRAFO_GEOAX_ASSIGN_TAB_4[0] =1
MD24420 $MC_TRAFO_GEOAX_ASSIGN_TAB_4[1] =4
MD24420 $MC_TRAFO_GEOAX_ASSIGN_TAB_4[2] =3

```

## 8.6 直交 PTP 移動

## プログラムコード

```

; TRACON は TRANSMIT 257/TRAANG を重畳 (X1 に対して傾斜した Y1 軸)
MD24430 $MC_TRAFO_TYP_5 = 8192 ; TRACON
MD24996 $MC-TRACON_CHAIN_2[0] = 2
MD24996 $MC-TRACON_CHAIN_2[1] = 1
MD24996 $MC-TRACON_CHAIN_2[2]=0
MD24434 $MC_TRAFO_GEOAX_ASSIGN_TAB_5[0] =1
MD24434 $MC_TRAFO_GEOAX_ASSIGN_TAB_5[1] =4
MD24434 $MC_TRAFO_GEOAX_ASSIGN_TAB_5[2] =3
M17

```

## NC プログラム

## プログラムコード

```

$TC_DP1[1,1]=120 ; 工具タイプ
$TC_DP2[1,1] = 0
$TC_DP3[1,1]=3 ; 長補正ベクトル
$TC_DP4[1,1]=25
$TC_DP5[1.1] =5
$TC_DP6[1,1]= 2; 半径; 工具半径
; 座標変換の切り替え
N1000 G0 X0 Y=0 Z0 A80 G603 SOFT G64
N1010 X10 Y20 Z30 ; 自動的に選択されるため、 TRAANG(,1) 指令は不要
N1110 TRANSMIT(1)
N1120 X10 Y20 Z30
N1130 Y2=0 ; 自動的に選択されるため、 TRACON(2) 指令は不要
N1210 TRAFOOF ; 自動的に変換されるため、 TRAANG(,1) 指令は不要
N1220 X10 Y20 Z30
M30

```

## 8.6 直交 PTP 移動

## 8.6.1 機能

## 前提条件

直交のポイントツーポイント、または PTP 移動が、以下の座標変換が有効な場合に有効化できること。

- TRAORI (方向座標変換)
- TRANSMIT (表面座標変換)

- TRACON (座標変換重畳)  
前提条件:1 番目の座標変換重畳が PTP 移動をサポートしていること。
- RCTRA (2 ~ 4 軸処理/ロボットキネマティックスの座標変換)  
前提条件:コンパイルサイクル「RMCC/RCTRA 座標変換処理」がロードされ、有効であること。
- RCBX (4 ~ 6 軸処理/ロボットキネマティックスの座標変換)  
前提条件:コンパイルサイクル「RMCC/RCBX 座標変換拡張ロボティックス」がロードされ、有効であること。

有効な座標変換がない場合、アラーム 14146「座標変換なしの CP または PTP 動作は許可されていません」が出力されます。

---

#### 注記

座標変換の軸は、同時に位置決め軸とすることはできません。

---

## 機能

直交の PTP 移動の場合、G0 および G1 のセットで、同期軸の移動を使って直交の終点としてプログラムされている点にアプローチすることができます。

移動動作では、直交の PTP 移動は、有効な座標変換がないように作用します。パートプログラムの位置データは、引き続き直交のワーク座標系を使用します。プログラムされたフレーム(座標移動および回転)は、引き続き有効です。表示も直交のワーク座標系になります。

直交の PTP 移動の動作は、軸の始点および終点で説明されます。個々の軸の開始から終了位置で描画される移動の軌跡は、コントローラでは軌跡補間点を使って指定されないため、予測できません。軸は、単に指定された目標位置に移動するだけです。軸の動きの相互作用は、時間に関連してのみ存在します。つまり、すべての軸が同時に開始し、目標位置に到達すると同時に停止します。最も遅い軸が合計移動時間を決定します。

ひとつの実用的なオプションは、特異点を使用することです。CAD/CAM システムが提供する直交位置を利用できる場合、たとえば、それらを機械軸の値に変換する必要がないという利点があります。さらに、機械軸の直接の移動は、有効な座標変換とプログラムされた送りによる直交移動と比較して時間が最適化されます。この結果、特異点を移

## 8.6 直交 PTP 移動

動する場合の対応する送り速度の抑制(先読みの速度プランニング)、または先読みで速度プランニングが実行されない場合は軸の過負荷のいずれかにつながります。

## 注記

直交の PTP 移動は、補間タイプ G0 および G1 との組み合わせのみで使用できます。

## バージョン

直交の PTP 移動には、次の 3 つのバージョンがあります。

- **PTP**  
G0 および G1 ブロック内のプログラムされた直交位置には、同期軸移動でアプローチします。
- **PTPG0**  
同期軸移動でアプローチされるプログラムされた直交位置は、G0 ブロック内にのみ存在します。G1 ブロックでは、移動タイプ CP (CP = 連続軌跡)に切り替えられ、直交の軌跡移動で位置にアプローチします。
- **PTPWOC (方向座標変換のみ)**  
ただし、PTP と同様に、回転軸および旋回軸によって生じる補正移動はありません。方向変更中に工具先端(TCP)の移動を補正する座標変換が有効でないため、座標変換に直線軸がプログラムされていない場合、一般的にワーク座標系の TCP の位置も変更する必要があります。  
プログラムされた直線軸の移動は、ブロックの最初に適用され、該当する場合は同時にプログラムされた回転軸の移動を重ねるワーク座標系で実行されます。よって、ワーク座標系で到達するブロック終点は、一般的に、プログラムされた終点とは同じではありません。

次の表は、座標変換と、それに実際に使用できるバージョンの一覧です。

変換	PTP	PTPG0	PTPWOC	CP
TRAORI	+	+	+	+
TRANSMIT	+	+	-	+
RCTRA	+	+	-	+
ROBX	+	+	-	+

<p><b>通知</b></p> <p><b>干渉の可能性</b></p> <p>PTP 移動では、CP より大幅に異なる工具の移動が発生する場合がありますことに注意してください!</p> <p>有効な座標変換とは独立してサブプログラムを作成することができるため、特にこれは PTPG0 で考慮してください。ただし、有効なフロントエンド座標変換(TRANSMIT)の場合は、方向座標変換 (TRAORI)とはまったく異なる干渉のリスクが発生する場合があります。</p> <p>有効なフロントエンド座標変換(TRANSMIT)付きの PTP 移動では、機械軸は基本的に最短軌跡で移動します。ブロック終点のわずかな変位が、回転軸が+ -179.99°ではなく +179.99°回転する原因となる場合があります。これによって、ブロック終点がほとんど変更されていない場合でも、まったく異なる移動がおこなわれます。</p>
--

### NC プログラムでの PTP/CP の切り替え

NC プログラムでの直交の軌跡移動(CP)と直交の PTP 移動の切り替えは、G グループ 49 の命令でおこなわれます(「直交 PTP 移動(PTP、PTPG0、PTPWOC、CP)の有効化/無効化 (ページ 576)」を参照してください)。

### JOG モードの PTP/CP 切り替え

直交の軌跡移動(CP)と直交の PTP 移動の切り替えは、JOG モードでもおこなえます。切り替えは、次の NC/PLC インタフェース信号を使って実行します。

DB21、... DBX29.4 (PTP 移動の起動)

#### 注記

DB21、... DBX29.4 は、座標変換が有効な場合に JOG モードでのみ有効になります。

移動の有効なタイプは、次の NC/PLC インタフェース信号で出力されます。

DB21、... DBX317.6 (PTP 移動有効)

## 8.6 直交 PTP 移動

## 8.6.2 セットアップ

## 8.6.2.1 電源投入後の動作

電源投入後、座標変換を伴った軸移動のために移動モード **CP** が自動的に有効になります。

次のマシンデータを使って、デフォルトの設定を直交の PTP 移動に切り替えることができます。

MD20150 \$MC\_GCODE\_RESET\_VALUES[48] (G グループ 49 の初期設定)

値	意味
1	CP (初期設定)
2	PTP
3	PTPG0
4	PTPWOC

## 8.6.2.2 リセット/パートプログラム終了後のレスポンス

リセット/パートプログラム終了後に G グループ 49 のどの命令が有効になるかは、次のマシンデータの設定によって異なります。

MD20152 \$MC\_GCODE\_RESET\_MODE[48] (G グループ 49 のリセット応答)

値	意味
0	MD20150 \$MC_GCODE_RESET_VALUES[48]のデフォルトの命令が有効になります(初期設定)。
1	リセット/パートプログラム終了の前に有効だった命令が有効になります。

### 8.6.2.3 PTP 移動中に SW 制限を考慮

#### 機能

直交の PTP 移動では、TU (ページ 582)命令で位置情報がプログラムされていない場合、回転軸は通常は「最短軌跡」に基づいて移動します。目標位置がソフトウェアのリミットスイッチの外側にある場合に、ソフトウェアのリミットスイッチを通過して移動しないようにするために、「PTP 移動中に SW 制限を考慮」機能を使用することができます。これを使用した場合、ソフトウェアのリミットスイッチを通過した場合に、回転軸は「最短軌跡」ではなく「長い軌跡」で、つまり「最短軌跡」と反対の移動方向に移動します。

#### 必要条件

関連する軸が原点確立済みの場合のみこの機能を使用できること。

#### 適用

「PTP 移動中に SW 制限を考慮」機能は、各軸で個別に有効化できます。

この設定は、軸マシンデータのビット 14 を使っておこないます。

MD30455 \$MA\_MISC\_FUNCTION\_MASK

ビット 14	= 0	直交の PTP 移動の場合、「最短軌跡」は回転軸がソフトウェアのリミットスイッチを通過しても維持されます(初期設定)。
	= 1	直交の PTP 移動中に「最短軌跡」の回転軸がソフトウェアのリミットスイッチを通過すると「長い軌跡」が使用されます。

#### 補足条件

##### モジュロ回転軸

モジュロ回転軸の場合、「長い軌跡」は関連する軸でソフトウェアのリミットスイッチ監視が有効化されている場合のみ使用できます。

DB31、... DBX12.4 (モジュロ回転軸:トラベル範囲制限を有効化) = 1

#### 例

軸 6 が +150°~+240°まで移動します。ソフトウェアのリミットスイッチは +200°にあります。

## 8.6 直交 PTP 移動

軸 6 に対して MD30455 \$MA\_MISC\_FUNCTION\_MASK のビット 14 が「1」に設定されている場合、軸 6 は-120°の位置に移動します。

## 8.6.2.4 STAT と TU の表示

SINUMERIK Operate ユーザーインターフェースの「機械」操作エリアで、STAT (ページ 578)と TU (ページ 582)の現在値を追跡できます。

Y	997.974514
Z	880.502383
A	-51.356473 °
B	89.879675 °
C	81.356404 °
STAT	00000010
TU	00100011

前提条件:STAT と TU の表示が、SINUMERIK Operate のセットアップ時に有効になっていること。

## 参照先:

試運転マニュアル、SINUMERIK Operate、「機械」操作エリアの調整」の章>「STAT と TU の表示」

## 8.6.3 プログラミング

## 8.6.3.1 直交 PTP 移動(PTP、PTPG0、PTPWOC、CP)の有効化/無効化

直交ポイントツーポイント、つまり PTP 移動は G グループ 49 命令を使用して NC プログラムで有効化/無効化されます。

これらの命令はモーダルです。デフォルト設定は、直交軌跡移動(CP)での移動です。

CP とは対照的に、有効な PTP 移動の場合、直交ターゲット点のみが変換され、機械軸は同期制御で移動されます。

直交ターゲット点を機械軸値に一義的に変換するには、位置データと角度データに加えて、軸位置を識別するための情報も必要です。このデータは、設定可能なアドレス STAT (ページ 578)および TU (ページ 582)から取り出されます。

## 前提条件

座標変換 TRAORI、TRANSMIT、RCTRA、または ROBX が有効であること。

## 構文

```
PTP / PTPG0 / PTPWOC
...
CP
```

## 意味

PTP:	ポイントツーポイント移動 PTP の起動 G0 および G1 ブロック内のプログラムされた直交位置には、同期軸移動でアプローチします。
PTPG0:	ポイントツーポイント移動 PTPG0 の起動 同期軸移動でアプローチされるプログラムされた直交位置は、G0 ブロック内にのみ存在します。G1 ブロックでは、CP 軌跡移動に切り替えられます。
PTPWOC:	ポイントツーポイント移動 PTPWOC を有効化 (方向座標変換が有効な場合のみ可能) ただし、PTP と同様に、回転軸および旋回軸によって生じる補正移動はありません。
CP:	ポイントツーポイント移動の無効化と軌跡移動 CP の有効化 直交軌跡移動は CP で実行されます。

## 注記

## PTPWOC

PTPWOC を座標変換 RCTRA または ROBX と併用することは無意味です。

## 例

参照:

- 例 1:ROBX 座標変換による 6 軸ロボットの PTP 移動 (ページ 585)
- 例 2:総合 5 軸座標変換の場合の PTP 移動 (ページ 586)
- 例 3:PTPG0 および TRANSMIT (ページ 587)

## 8.6 直交 PTP 移動

## 8.6.3.2 ジョイント(STAT)の位置を指定します

直交座標付きの位置データと工具オリエンテーションの指定だけでは、機械の位置を一義的に識別するには不十分です。同じ工具オリエンテーションに対して、複数のジョイント位置が考えられるからです。関連するキネマティックに応じて、8 つもの異なるジョイント位置が考えられます。これらの異なるジョイント位置は、座標変換に固有です。

不確定性を回避するために、ジョイント位置は **STAT** アドレスで指定されます。

**注記**

制御装置は PTP 移動に対してプログラムされた **STAT** 値のみを考慮します。有効な座標変換で移動する場合、通常は位置変更が不可能であるため、**CP** 移動は無視されます。有効な **CP** で移動する場合、ターゲット点の位置は起点から取られます。

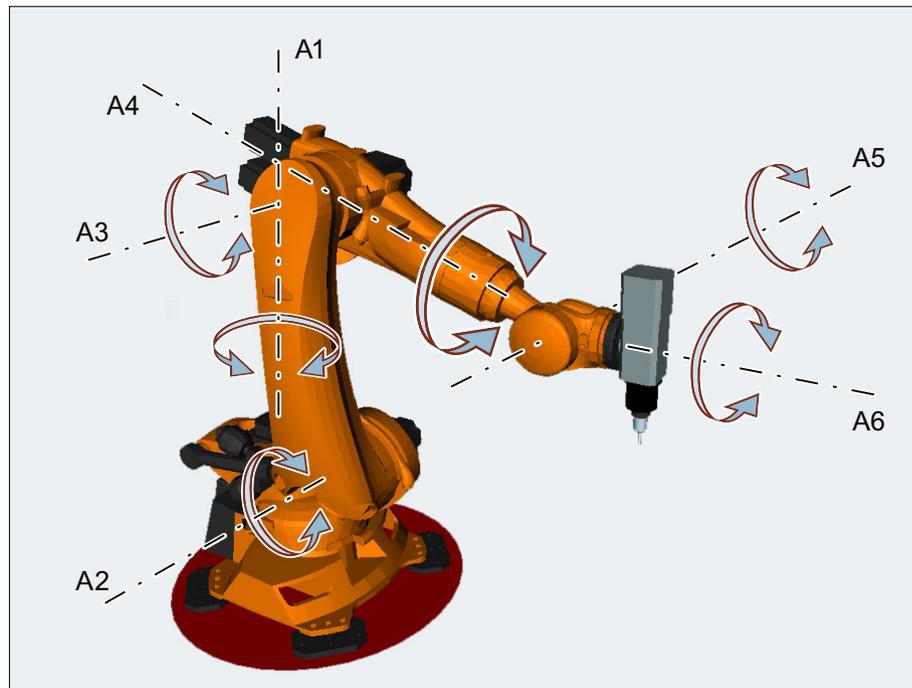
## 構文

STAT=<値>

## 意味

STAT:	ジョイント位置を指定するための設定可能なアドレス
<値>:	2 進数または 10 進数値 考えられる位置ごとに 1 ビットが含まれます。ビットの有効桁は特定の座標変換によって定義されます。

**STAT** の使用は、フライス主軸を備えた 6 軸の多関節ロボットで説明されます。**ROBX** ロボット座標変換を使用してキネティックトランスフォーメーションが実行されます(前提条件:コンパイルサイクル「**RMCC/RCBX** 座標変換拡張ロボティクス」がロードされ、有効であること)。

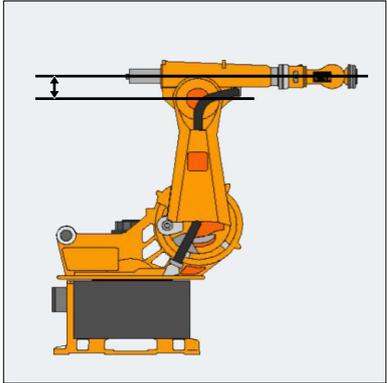


軸 A1、A2、および A3 は、多関節ロボットのメイン軸です。メイン軸とともに、ヘッド軸またはハンド/リスト軸としても指定される軸 A4、A5、および A6 が加工スペースに配置されます。ハンド/リスト軸がさらに移動した結果、特定の加工作業での必要に応じてフライス主軸を空間で旋回できます。同じ工具オリエンテーションを実現するために、さまざまな関節継手位置を使用できます。

加工に必要な関節継手位置は、調節可能な STAT アドレスのビット 0 ~ 2 のプログラミングで選択します。

ビット 0	手動軸の交点の位置(A4, A5, A6)	
= 0	基本範囲(右ショルダー)	
= 1	オーバーヘッド範囲(左ショルダー)	
		<p>例:ハンド/リスト軸の交点が基本範囲内にあります。</p>

8.6 直交 PTP 移動

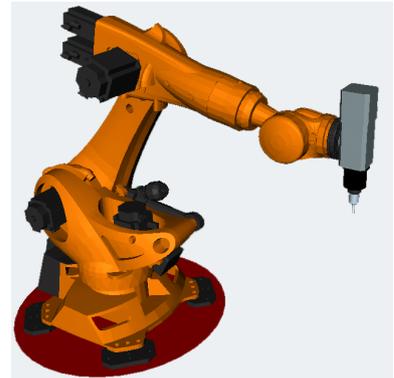
ビット 1	軸 3 の位置	
	ビット 1 の値が変化する角度は、特定のロボットタイプによって異なります。軸 3 および 4 が交差するロボットには以下が適用されます。	
	= 0	$A3 < 0^\circ$ (下部エルボー)
	= 1	$A3 \geq 0^\circ$ (上部エルボー)
	注記:	 <p>A3 と A4 の間のオフセット</p>
軸 3 と 4 の間にオフセットのあるロボットの 場合、ビット 1 の値が変化する角度はこのオ フセットの大きさとは無関係です。		
ビット 2	軸 5 の位置	
	= 0	$A5 \leq 0^\circ$ (ハンドフリップなし)
	= 1	$A5 > 0^\circ$ (ハンドフリップ)

プログラム例:

プログラムコード	コメント
...	
N14 T="T8MILLD20" D1	; \$TC_DP3[1,1]=132.95
N16 ORIMKS	
N17 G1 PTP X1665.67 Y0 Z1377.405 A=0 B=0 C=0 <b>STAT=...</b> F2000	; STAT 値は関節継手位置を定義 します (以下を参照)
...	

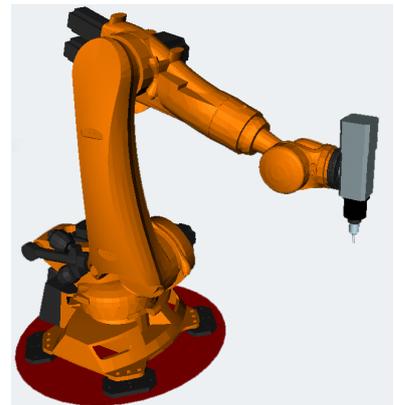
STAT=1 ('B001')

- 左ショルダー
- 下部エルボー
- ハンドフリップなし



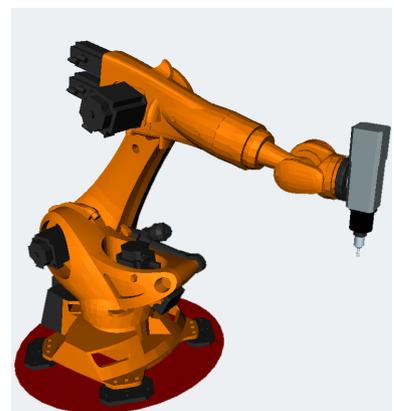
STAT=2 ('B010')

- 右ショルダー
- 上部エルボー
- ハンドフリップなし



STAT=5 ('B101')

- 左ショルダー
- 下部エルボー
- ハンドフリップ





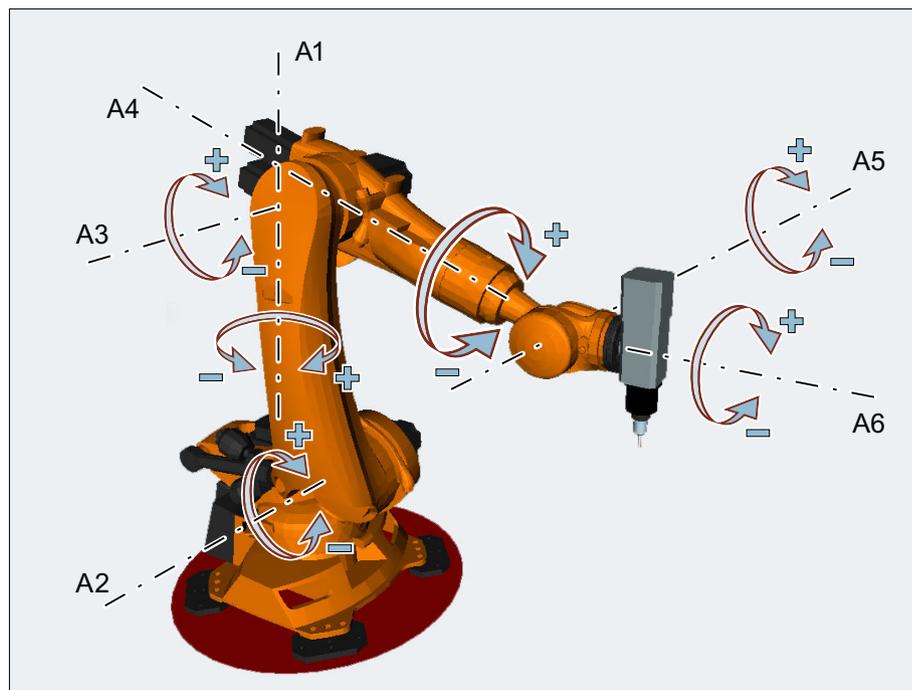
## 構文

TU=&lt;値&gt;

## 意味

TU:	軸角度の符号を指定するための設定可能アドレス		
<値>:	2進数または10進数値 座標変換に関する軸ごとに、軸角度( $\theta$ )の符号、すなわち移動方向を示すビットが存在します。		
ビット	= 0	軸角度の符号: +	軸角度の範囲: $0^\circ \leq \theta < 360^\circ$
	= 1	軸角度の符号: -	軸角度の範囲: $-360^\circ < \theta < 0^\circ$

例:6軸の多関節ロボット



ビット	意味	値	軸角度の符号	軸角度
ビット 0 <sup>1)</sup>	A1の軸角度の符号	= 0	+	$\geq 0^\circ$
		= 1	-	$< 0^\circ$
ビット 1 <sup>1)</sup>	A2の軸角度の符号	= 0	+	$\geq 0^\circ$
		= 1	-	$< 0^\circ$

8.6 直交 PTP 移動

ビット	意味	値	軸角度の符号	軸角度
ビット 2 <sup>1)</sup>	A3 の軸角度の符号	= 0	+	$\geq 0^\circ$
		= 1	-	$< 0^\circ$
ビット 3 <sup>1)</sup>	A4 の軸角度の符号	= 0	+	$\geq 0^\circ$
		= 1	-	$< 0^\circ$
ビット 4 <sup>1)</sup>	A5 の軸角度の符号	= 0	+	$\geq 0^\circ$
		= 1	-	$< 0^\circ$
ビット 5 <sup>1)</sup>	A6 の軸角度の符号	= 0	+	$\geq 0^\circ$
		= 1	-	$< 0^\circ$

1) ロボット軸のチャンネル軸番号から得られた現在の TU ビット番号例では、ロボット軸(A1 ~ A6)がチャンネルの最初の 6 本の軸であるため、TU ビット 0 ~ 5 が使用されます。ロボット軸の別のチャンネル軸の割り当てでは、ロボット軸の TU ビット番号が、それに対応して変わります(例:ロボット軸は 3 ~ 8 番目のチャンネル軸です。つまり、TU ビット 2 ~ 7 がロボット軸に使用されます)。

従って、TU=19 (TU='B010011 に対応)は以下を意味します。

ビット	値		軸角度
0	= 1	①	$\theta_{A1} < 0^\circ$
1	= 1	①	$\theta_{A2} < 0^\circ$
2	= 0	①	$\theta_{A3} \geq 0^\circ$
3	= 0	①	$\theta_{A4} \geq 0^\circ$
4	= 1	①	$\theta_{A5} < 0^\circ$
5	= 0	①	$\theta_{A6} \geq 0^\circ$

注記

移動範囲が±360°を超える軸の場合は、軸が常に最短の軌跡に沿って移動します。これは、軸位置を TU 情報で固有の指定をすることができないためです。

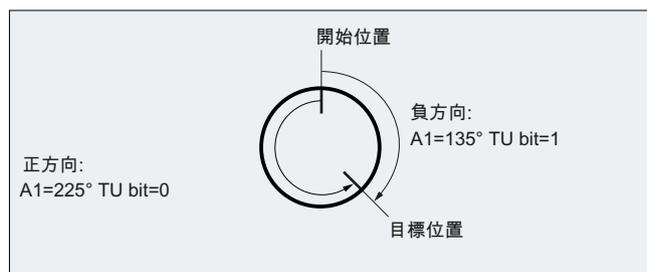
位置に対して TU がプログラム指令されていない場合は、MD30455 \$MA\_MISC\_FUNCTION\_MASK に応じて、より短い、または長い軌跡を移動します(上級機能の機能マニュアルの「PTP 移動のソフトウェア制限の考慮 (ページ 575)」の章を参照してください)。

**TRANSMIT**

TRANSMIT が有効の PTP 移動の場合、TU のアドレスは意味がありません!

**例**

次の図に示す回転軸の位置は、負または正方向へアプローチ可能です。回転位置は、アドレス A1 でプログラム指令されます。移動方向は、TU が指定されている場合にのみ絶対的にクリアです。

**8.6.3.4 例 1:ROBX 座標変換による 6 軸ロボットの PTP 移動**

以下の使用例では、直交 PTP 移動および関連する NC 命令を例として示しています。

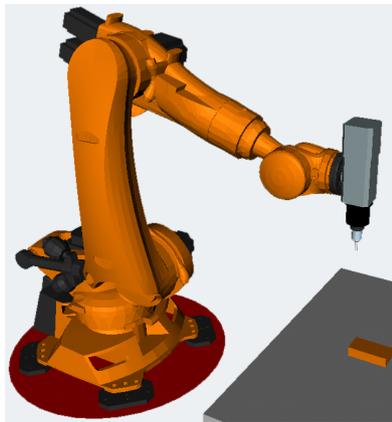


図 8-16 フライス主軸を備えた 6 軸の多関節ロボット

```

N1 G90
N2 T="T8MILLD20" D1 M6
N3 TRAORI
; $P_UIFR[1]=CTRANS(X,1500,Y,0,Z,400):CROT(X,0,Y,0,Z,-90)
N4 G54

```

## 8.6 直交 PTP 移動

```

N5 M3 S20000
N6 ORIWKS
N7 ORIVIRT1
N8 CYCLE832(0.01, _FINISH, 1)
;HOME
N9 TRAF00F
N10 G0 RA1=0.0000 RA2=-90.0000 RA3=90.0000 A=0.0000 B=90.0000 C=0.0000
N11 TRAORI
N12 G54
N13 G0 PTP X1369.2426 Y956.7528 Z502.5517 A=135.5761 B=-33.2223 C=161.1435
STAT='B010' TU='B001011'
N14 G0 X1355.1242 Y1014.9394 Z424.9695 A=135.8491 B=-33.1439 C=160.9941
STAT='B010' TU='B001011'
N15 G1 CP X1354.8361 Y1016.1269 Z423.3862 A=136.0635 B=-33.0819 C=160.8770
F1000
N16 G1 X1336.4283 Y1016.1269 Z426.6311 A=136.0484 B=-32.2151 C=160.9643
F2000
N17 G1 X1317.9831 Y1016.1269 Z429.6730 A=136.0175 B=-31.3394 C=161.0655
;HOME
N18 TRAF00F
N19 G0 RA1=0.0000 RA2=-90.0000 RA3=90.0000 A=0.0000 B=90.0000 C=0.0000
N20 M30

```

## 8.6.3.5 例 2:総合 5 軸座標変換の場合の PTP 移動

前提条件:直角の CA キネマティックがベースとして使用されていること。

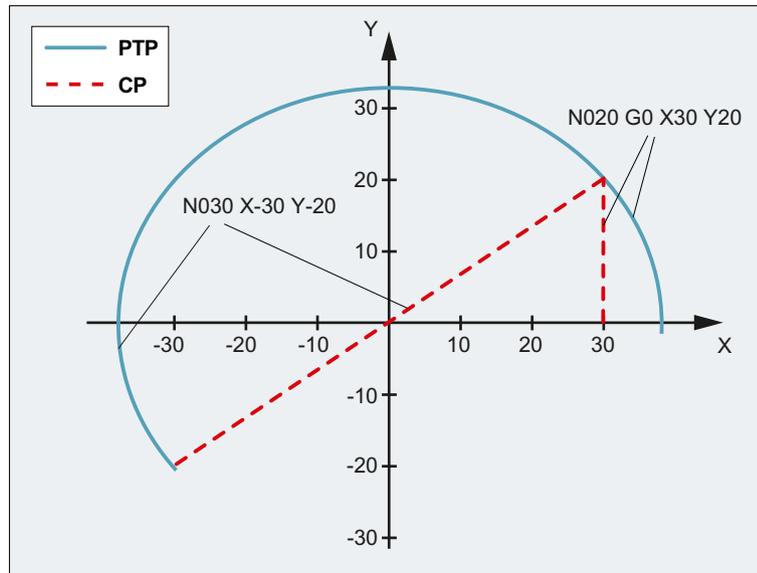
プログラムコード	コメント
<b>TRAORI</b>	; 座標変換 CA キネマティックスをオンにします
<b>PTP</b>	; PTP 移動を起動します
N10 A3=0 B3=0 C3=1	; 回転軸位置 C=0 A=0
N20 A3=1 B3=0 C3=1	; 回転軸位置 C=90 A=45
N30 A3=1 B3=0 C3=0	; 回転軸位置 C=90 A=90
N40 A3=1 B3=0 C3=1 <b>STAT=1</b>	; 回転軸位置 C=270 A=-45

回転軸位置の明確なアプローチ位置は、次のように選択します。

ブロック N40 で、回転軸は - **STAT=1** のプログラミングの結果 - 開始点(C=90, A=90)から終点(C=270, A=-45)までの距離より長い距離を移動します。一方、**STAT=0** では、回転軸は終点(C=90, A=45)までの最短距離に沿って移動します。

## 8.6.3.6 例 3:PTPG0 および TRANSMIT

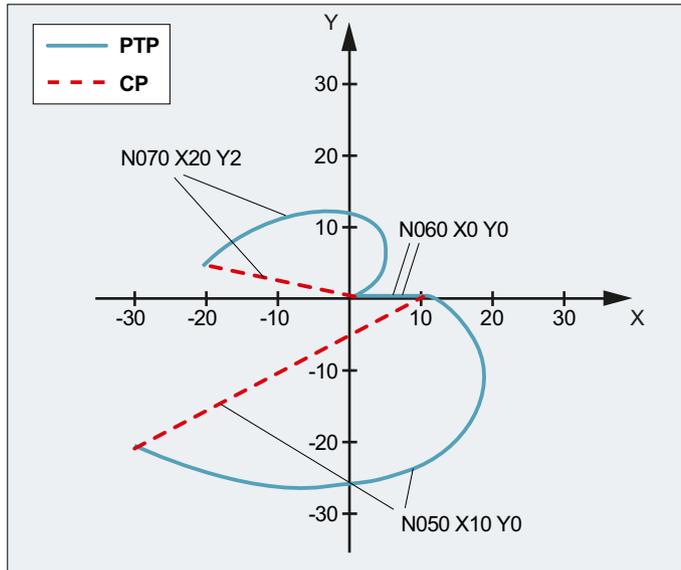
## PTPG0 および TRANSMIT での極を中心とした移動



プログラムコード	コメント
N001 G0 X30 Z0 F10000 T1 D1 G90	; 初期設定、アブソリュート指令
N002 SPOS=0	
N003 <b>TRANSMIT</b>	; TRANSMIT 座標変換
N010 <b>PTPG0</b>	; G0 ブロックごとに、自動的に PTP - その後、再び CP。
N020 G0 X30 Y20	
N030 X-30 Y-20	
N120 G1 X30 Y20	
N110 X30 Y0	
M30	

8.6 直交 PTP 移動

PTPG0 および TRANSMIT での極からの移動



プログラミング	コメント
N001 G0 X90 Z0 F10000 T1 D1 G90	; 初期設定
N002 SPOS=0	
N003 <b>TRANSMIT</b>	; TRANSMIT 座標変換
N010 <b>PTPG0</b>	; G0 ブロックごとに、自動的に PTP - その後、再び CP。
N020 G0 X90 Y60	
N030 X-90 Y-60	
N040 X-30 Y-20	
N050 X10 Y0	
N060 X0 Y0	
N070 X-20 Y2	
N170 G1 X0 Y0	
N160 X10 Y0	
N150 X-30 Y-20	
M30	

8.6.4 補足条件

工具径補正(TRC)

PTP/PTPWOC は TRC で計算された以外の輪郭を作成するため、PTP/PTPWOC では工具径補正(TRC)を有効化できません。

PTPG0 の動作は異なります。この場合、有効な TRC を使って、TRC が適切に実行されるよう CP への内部切り替えがおこなわれます。

### 滑らかなアプローチおよび後退(SAR)

PTP/PTPWOC が有効な場合、SAR は使用できません。これは、SAR がアプローチと後退を設計するため、および接線方向にアプローチまたは後退するために輪郭を必要とするからです。

一方、PTPG0 の場合、SAR に必要なブロックが CP と一緒に移動するため SAR が可能です。実際のアプローチ輪郭までの G0 ブロックは、PTP の移動を使用して、すぐに実行されます。同じことが、ブロックの後退にも適用されます。

### ブロックサイクル

有効な PTP/PTPWOC の場合、ブロックが切削分割を構成するために輪郭を必要とするため、CONTPRON や CONTDCON などのブロックサイクルは適用できません。

一方、PTPG0 の場合、ブロックサイクルに必要なブロックが CP と一緒に移動するため、ブロックサイクルが可能です。

### 面取りおよび S 字

PTP 移動中は面取りおよび S 字は無視されます。

### 軸の重畳

補間に重畳する軸は、PTP シーケンス中に変更できません。これは、LIFTFAST、精密工具オフセット、連結動作 TRAILON、および法線方向制御 TANGON なども当てはまります。

### NC ブロック圧縮

PTP ブロックでは、有効なコンプレッサ機能(COMPON、COMPCURV、COMPCAD または COMPSURF)は自動的に解除されます。

### 丸み付け

PTP 移動では G643 を使った滑らかな指令は不可能です。G643 が有効な PTP ブロックでは、G642 への自動切り替えがおこなわれます。

## 8.7 直交座標ジョグ移動(オプション)

また、以下の特徴に注意してください。

- 滑らかな指令は、常に機械座標系でおこなわれ、よって、時間の面でのパフォーマンスは高くなります。G642 の条件は、常に機械座標系で解釈され、通常の直交の基本座標系ではおこなわれません。
- スムーズ化 G641 は、機械軸座標から計算された仮想軌跡に応じて決定されます。

### 軌跡送り速度

G1 の F 値の入力は、機械軸座標から計算された仮想軌跡を示します。

### モード変更

直交の PTP 移動は、自動および MDI モードのみで意味を持ちます。

JOG モードへの変更時には、現在の設定が保持されます。

- PTP 移動を設定すると、軸は MCS で移動します。
- CP を設定すると、軸は WCS で移動します。

自動または MDI モードに戻った後、JOG モードへの切り替え前に有効だった状態が復元されます。

### ブロック検索

面端座標変換(TRANSMIT)が有効な場合、ブロック検索でプログラムのセクションを実行すると、同じ直交位置に対して異なる機械軸位置となる場合があります。

### REPOS

直交 PTP 移動の設定内容は、再位置決め中には変更されません。PTP が中断ブロックで設定された場合は、再位置決めも PTP でおこなわれます。

## 8.7 直交座標ジョグ移動(オプション)

---

### 注記

「直交座標ジョグ移動」機能には、「マテハン装置用座標変換」オプションが必要です。

---

## 機能

JOG モードの基準系としての「直交座標ジョグ移動」機能により、次の直交座標系で、軸を相互に個別設定できるようになります。

- 基本座標系(BCS)
- ワーク座標系(WCS)
- 工具座標系(TCS)

次のマシンデータを使用して、補正と適用がおこなわれます。

MD21106 \$MC\_CART\_JOG\_SYSTEM (直交 JOG の座標系)

ビット	意味
0	基本座標系
1	ワーク座標系
2	工具座標系

## 注記

ワーク座標系は、フレームによって、基本座標系に対してシフトして回転されています。

## 参照先:

機能マニュアル 基本機能、軸、座標系、フレーム(K2)

座標系での基準点の表示:

	ワーク原点		工具ホルダレファレンス点
---	-------	---	--------------

## 基準系の選択

ジョグ移動では、3つの基準系のいずれかについて、下記のセッティングデータによって、ジオメトリ軸による **平行移動** (荒削りオフセット)だけでなく、旋回軸による **オリエンテーション** についても個別に指定できます。

SD42650 \$SC\_CART\_JOG\_MODE

平行移動または旋回の基準系に対して1ビット以上が設定されている場合、または解除していない基準系を MD21106 \$MC\_CART\_JOG\_SYSTEM で設定しようとしたときは、アラーム 14148 「直交座標ジョグ移動の基準系は許可されません」が発生します。

8.7 直交座標ジョグ移動(オプション)

平行移動

平行移動動作を使用して、基準系の軸に平行に、または立体的に工具中心点(TCP)を移動できます。移動動作は、ジオメトリ軸のVDI信号経由で実行されます。

ジオメトリ軸を割り当てるには、マシンデータ MD24120

\$MC\_TRAFO\_GEOAX\_ASSIGN\_TAB\_x[n]を使用します。複数方向へ同時に移動することにより、基準系の方向に平行に、移動を実行できます。

BCS での平行移動

基本座標系(BCS)は、機械の直交座標原点を表します。

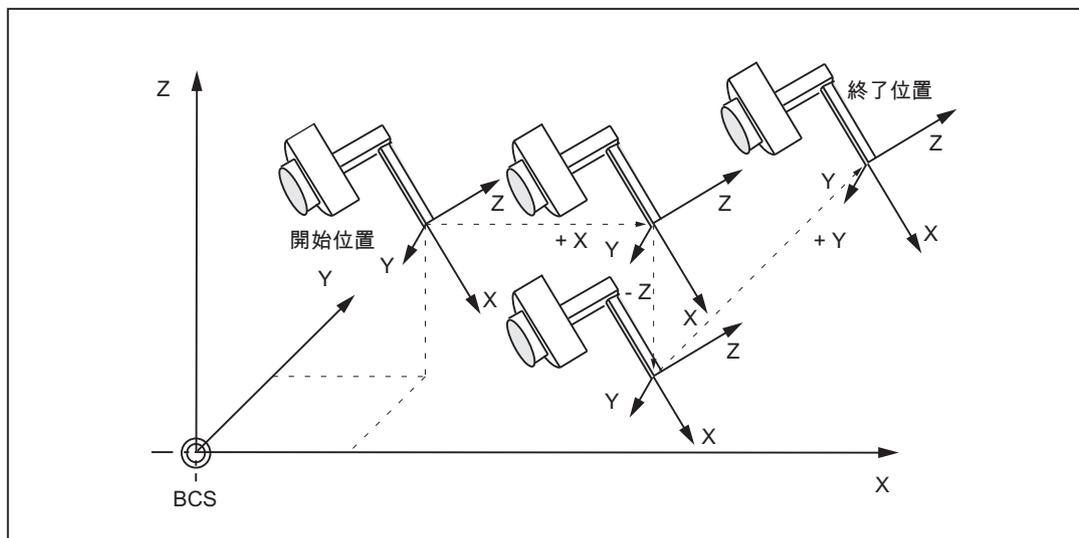


図 8-17 基本座標系での直交座標ジョグ移動 (平行移動)

### WCS での平行移動

ワーク座標系(WCS)はワーク原点にあります。ワーク座標系は、フレームによって、基本座標系を基準にしてシフトして回転できます。フレーム回転が有効である場合は、移動動作が、基本座標系での平行移動の移動に対応します。

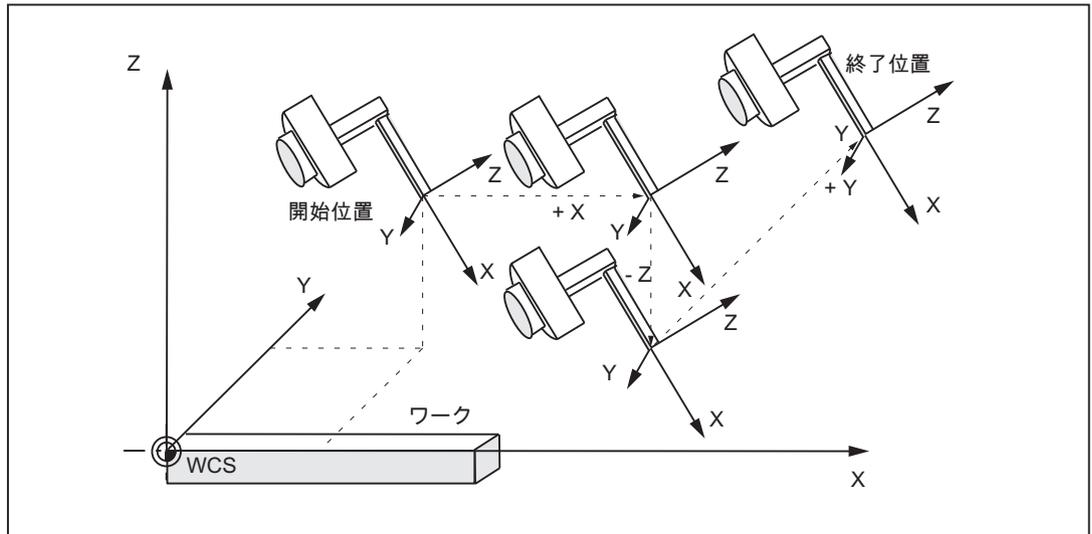


図 8-18 ワーク座標系での直交座標ジョグ移動(平行移動)

### TCS での平行移動

工具座標系(TCS)は工具の中心点にあります。その方向は、機械の現在の設定によって決まります。これは、工具座標系が移動中に動くためです。

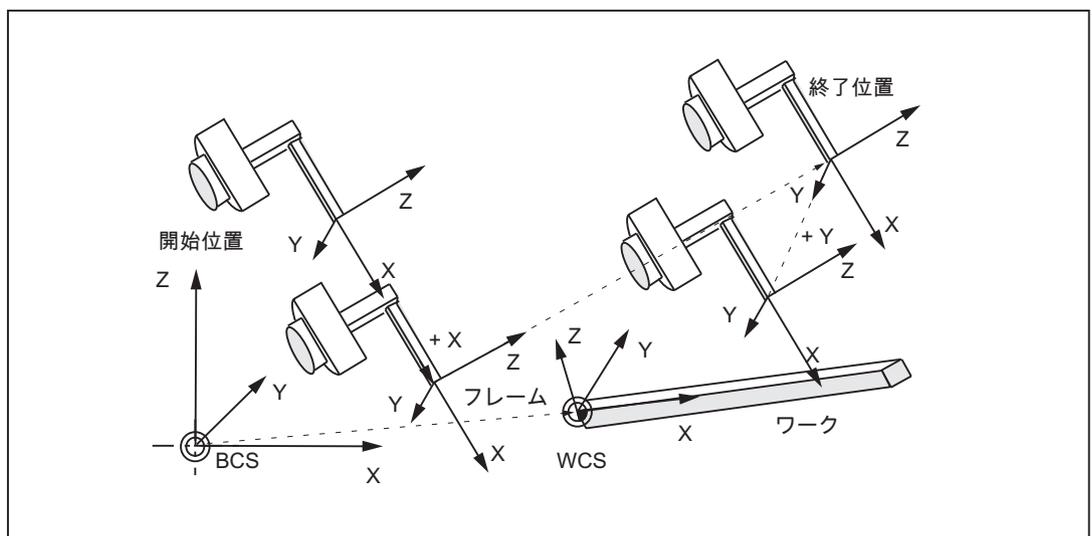


図 8-19 工具座標系での直交座標ジョグ移動(平行移動)

## TCS での同時平行移動と同時旋回

平行移動と旋回移動が同時に実行された場合、平行移動は常に、現在の工具の向きに対応して移動します。これにより、工具方向に直接実行される切り込み移動、または工具方向に対して垂直な移動が可能となります。

## 旋回

工具は、旋回移動によって、部品の表面に位置合わせされます。旋回移動は、旋回軸のVDI信号 (DB21、... DBB321)によって PLC から制御されます。

複数の旋回軸を同時に移動できます。実旋回軸は、当該の基準系の固定軸を中心とした回転を実行します。

この **回転** は、RPY 角度に従って識別されます。

- A 角度:Z 軸を中心とした回転
- B 角度:Y 軸を中心とした回転
- C 角度:X 軸を中心とした回転

### 回転のプログラミング:

ユーザーは、グループ 50 の現在の G 命令を旋回の定義として使用し、回転の実行方法を定義できます。

ORIEULER、ORIRPY、ORIVIRT1、および ORIVIRT2 の指定

ORIVIRT1 では、MD21120 \$MC\_ORIAX\_TURN\_TAB\_1 に従って回転が実行されます。

旋回軸が、機械軸経由でチャンネル軸に割り当てられます。MD24585

\$MC\_TRAFO5\_ORIAX\_ASSIGN\_TAB\_1

**回転方向** は、「右手の法則」に従って決定されます。親指が回転軸方向を指します。これが、回転の正方向を規定します。

## WCS での旋回

回転は、ワーク座標系の定義方向を中心として実行されます。座標回転フレームが有効な場合は、移動が基本座標系での回転に対応します。

## BCS での旋回

回転は、基本座標系の定義方向を中心として実行されます。

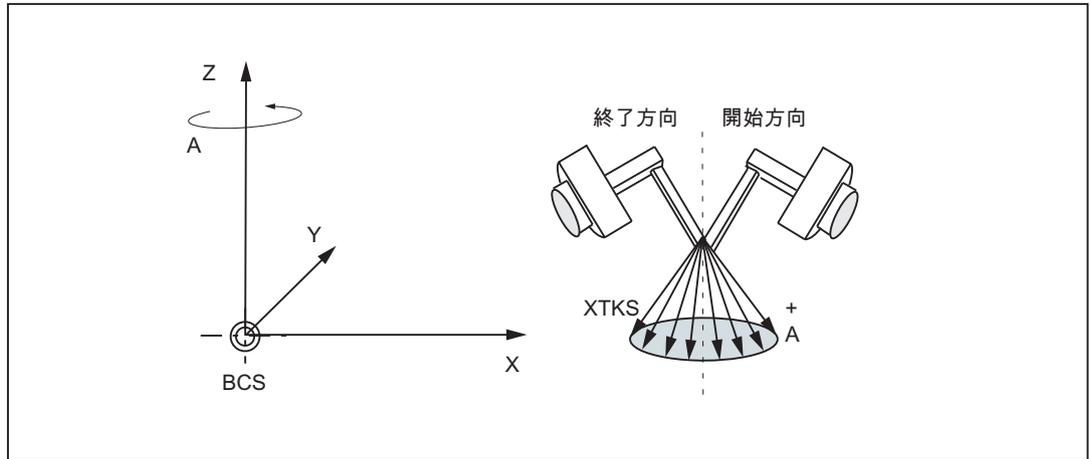


図 8-20 基本座標系での直交座標ジョグ移動、旋回角度 A

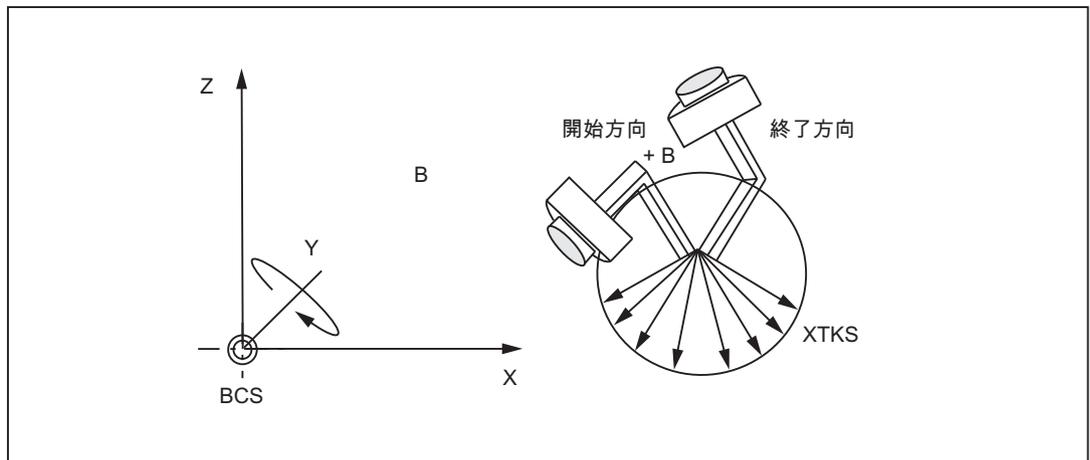


図 8-21 基本座標系での直交座標ジョグ移動、旋回角度 B

8.7 直交座標ジョグ移動(オプション)

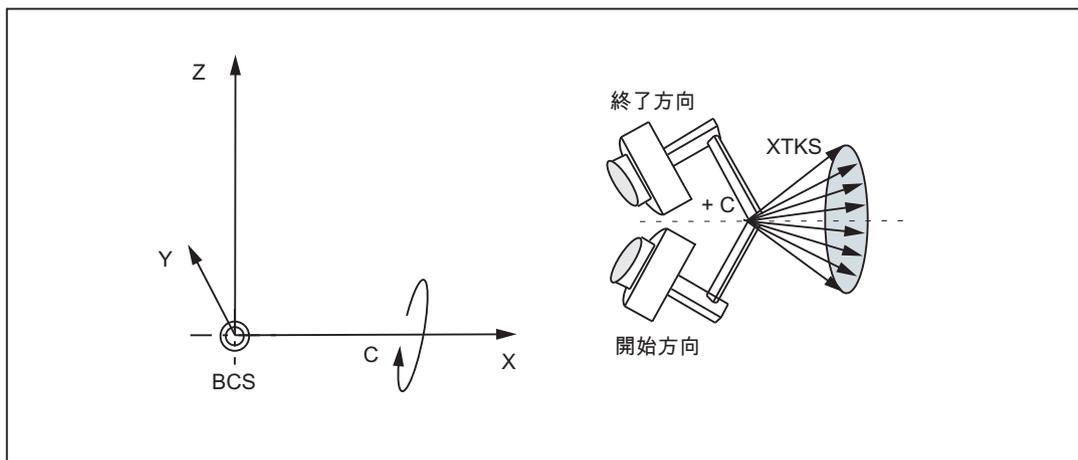


図 8-22 基本座標系での直交座標ジョグ移動、旋回角度 C

TCS での旋回

回転は、工具座標系での移動方向を中心とします。現在の工具のレファレンス点復帰方向が常に、回転軸として使用されます。

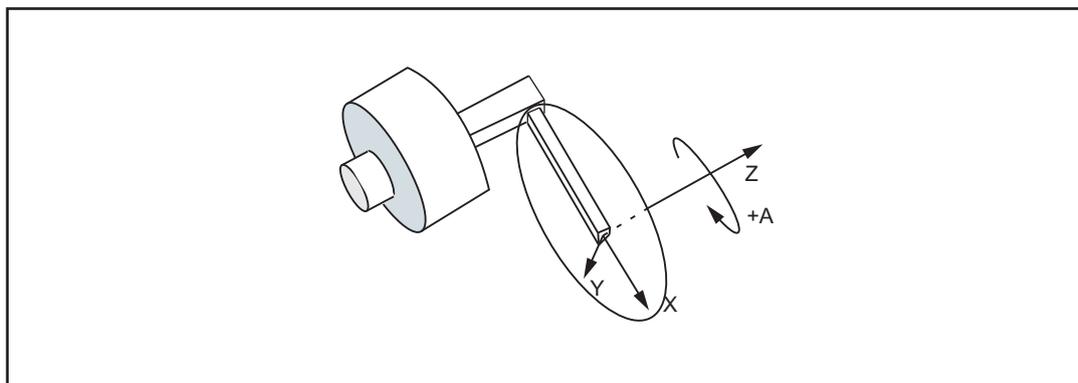


図 8-23 工具座標系での直交座標ジョグ移動、旋回角度 A

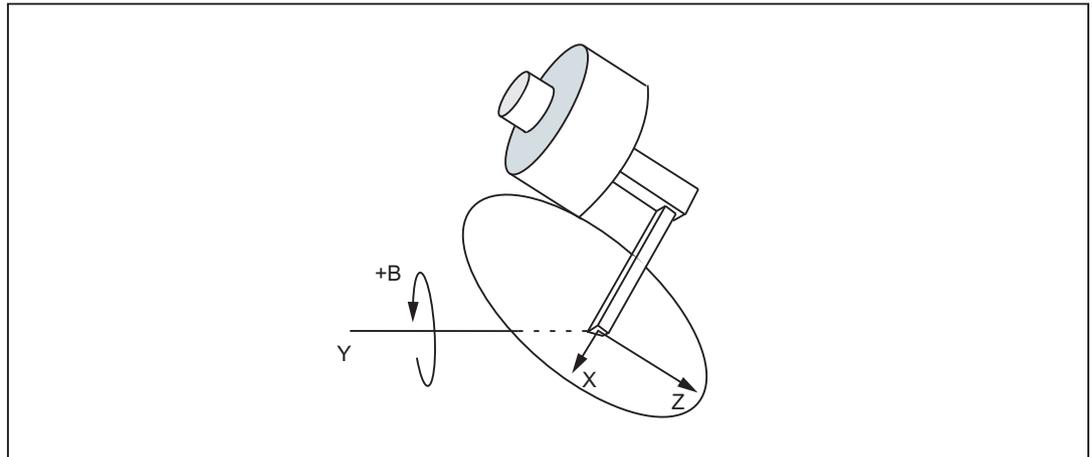


図 8-24 工具座標系での直交座標ジョグ移動、旋回角度 B

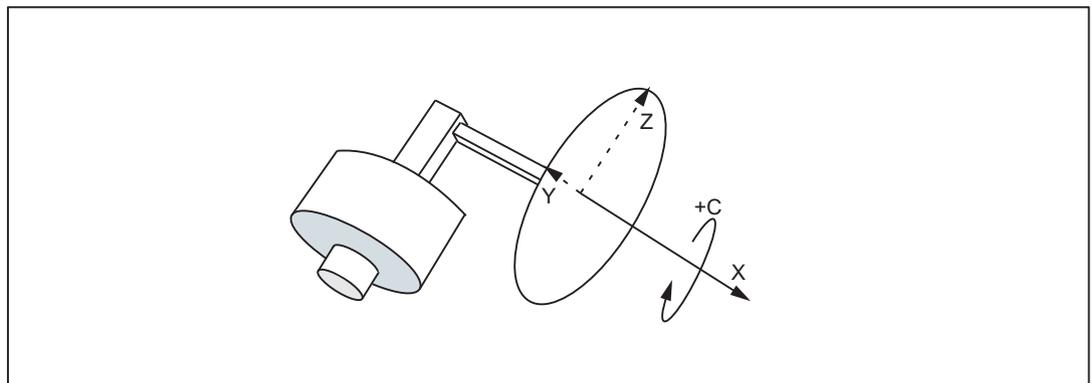


図 8-25 工具座標系での直交座標ジョグ移動、旋回角度 C

## 必要条件

「直交座標ジョグ移動」機能は、NC で座標変換が有効である場合にのみ実行できます。  
DB21、... DBX33.6 == 1 ("座標変換有効")

下記の必要条件を遵守する必要があります。

- 5 軸または 6 軸の座標変換を含む「マテハン装置用座標変換パッケージ」オプションが設定されている
- 実旋回軸は、次のマシンデータによって定義してください。  
MD24585 \$MC\_TRAFO5\_ORIAX\_ASSIGN\_TAB\_1[n]
- DB21、... DBX29.4 == 0 (PTP 移動の起動)
- MD21106 \$MC\_CART\_JOG\_SYSTEM > 0

8.7 直交座標ジョグ移動(オプション)

表 8-2 直交座標ジョグ移動の条件

プログラムでの座標変換が有効 (TRAORI..)	プログラム指令移動タイプ	DB21、...DBX29.4「PTP 移動の起動」	DB21、...DBX33.6「座標変換有効」
FALSE	無効	無効	0
TRUE	CP	0	1
TRUE	CP	1	0
TRUE	PTP	0	1
TRUE	PTP	1	0

プログラムで現在動作中の G 命令 PTP/CP は、直交座標ジョグ移動に影響しません。NC/PLC インターフェース信号は、ジオメトリ軸と回転軸のチャンネル DB で解釈されま

す。

適用

直交座標ジョグ移動の基準系は、次のように設定されます。

- 直交座標ジョグ移動機能は、次のマシンデータで有効になります。  
MD21106 \$MC\_CART\_JOG\_SYSTEM > 0  
BCS、WCS、または TCS 基準系は、MD21106 \$MC\_CART\_JOG\_SYSTEM によって有効になります。
- SD42650 SC\_CART\_JOG\_MODE による JOG 移動動作  
従来の標準動作の場合:ビット 0 ~ 2 = 0、ビット 8 ~ 10 = 0  
ビット 0 ~ 2 によって平行移動の基準系を、ビット 8 ~ 10 によって旋回の基準系を設定します。  
0 に設定されていないビットがある場合は、この新機能が使用されます。平行移動と旋回の基準系は、個別に設定することもできます。

SD42650 \$SC\_CART\_JOG\_MODE (1 ビットのみ設定):

SD42650 \$SC_CART_JOG_MODE							
ビット 11 ~ 15	ビット 10	ビット 9	ビット 8	ビット 7 ~ 3	ビット 2	ビット 1	ビット 0
予約済み	TCS での旋回	WCS での旋回	BCS での旋回	予約済み	TCS での平行移動	WCS での平行移動	BCS での平行移動

## 8.8 パートプログラム/ソフトキーによる座標変換マシンデータの起動

## 基準系の組み合わせ

次の表は、基準系のすべてのオプションの組み合わせを示しています。

SD42650 \$SC_CART_JOG_MODE						基準系の組み合わせ	
ビット 10	ビット 9	ビット 8	ビット 2	ビット 1	ビット 0	旋回	平行移動
0	0	0	0/1	0/1	0/1	標準	標準
標準	標準	標準	0	0	0	標準	標準
0	0	1	0	0	1	BCS	BCS
0	0	1	0	1	0	BCS	WCS
0	0	1	1	0	0	BCS	TCS
0	1	0	0	0	1	WCS	BCS
0	1	0	0	1	0	WCS	WCS
0	1	0	1	0	0	WCS	TCS
1	0	0	0	0	1	TCS	BCS
1	0	0	0	1	0	TCS	WCS
1	0	0	1	0	0	TCS	TCS

## 8.8 パートプログラム/ソフトキーによる座標変換マシンデータの起動

## 8.8.1 機能

変換 MD は、プログラム命令ソフトキーを使用して有効にすることができます。つまり、パートプログラムなどから書くことができ、このために、変換の設定が完全に変更されます。

10 個までの各種の変換をコントロールシステムで設定できます。変換タイプは、次のマシンデータで設定されます。

MD24100 \$MC\_TRAFO\_TYPE\_1

から

MD24460 \$MC\_TRAFO\_TYPE\_10 まで。

## 8.8 パートプログラム/ソフトキーによる座標変換マシンデータの起動

### 特性

変換 MD が、「マシンデータを有効に設定」を使用して有効化されます。

保護レベルは現在、7/7 (KEYSWITCH\_0)です。これは、このデータが特別な権限を必要とせずに NC プログラムから変更可能であることを意味します。

マシンデータが有効になっている(NEWCONF NC プログラム命令または HMI による発行、または、その後のリセットまたはプログラム終了による自動発行のいずれの場合でも)場合は、上のリストのマシンデータを制限なしで変更し、有効にすることができます。

特に重要なことは、新しい変換を設定するか、または、既存の変換を異なるタイプのいずれかに置換したり、削除したりできます。これは、変更オプションが、既存の変換の再パラメータ設定を制限していないためです。

## 8.8.2 制約事項

### マシンデータの変更

動作中の変換に関与するマシンデータは変更できません。変更しようとする、アラームが生成されます。

これらのマシンデータは通常、関連する変換データグループによって変換に割り当てられたすべてのマシンデータです。有効な変換のグループに含まれるが、使用されていないマシンデータは変更できます(ただし、意味のある操作とは考えられません)。たとえば、マシンデータ MD24564 \$MC\_TRAFO5\_NUTATOR\_AX\_ANGLE\_n は MD24100 \$MC\_\$MC\_TRAFO\_TYPE = 16 (回転工具および互いに垂直な 2 つの回転軸 A と B を持つ 5 軸座標変換)で動作中の変換に対して変更することは可能です。これは、この特定のマシンデータが変換には関与していないためです。

マシンデータ MD21110 \$MC\_X\_AXIS\_IN\_OLD\_X\_Z\_PLANE は、動作中のオリエンテーション変換によって、変更できない場合があることに注意してください。

---

### 注記

プログラムを中断(再位置決め、残移動距離削除、ASUP など)して再起動する場合は、すでに実行済みの複数のブロックが制御装置に必要です。有効な変換のマシンデータの変更が禁止されていることは、これらのブロックにも該当します。

---

例:

2つのオリエンテーション変換が、マシンデータ MD24100 \$MC\_TRAFO\_TYPE\_1 = 16、MD24200 \$MC\_TRAFO\_TYPE\_2 = 18 などによって設定されています。

「マシンデータの有効化」の実行時には、2番目の変換が有効であることを前提としています。この場合、次のような、1番目の変換のみに関連するマシンデータはすべて変更できます。

MD24500 \$MC\_TRAFO5\_PART\_OFFSET\_1

ただし、次のようなデータは変更できません。

MD24650 \$MC\_TRAFO5\_BASE\_TOOL\_2

または

MD21110 \$MC\_X\_AXIS\_IN\_OLD\_X\_Z\_PLANE

さらに、別の変換(TRANSMIT)を MD24300 \$MC\_TRAFO\_TYPE\_3 = 256 など設定でき、追加のマシンデータでパラメータ設定できます。

### ジオメトリ軸の定義

ジオメトリ軸は、コントロールシステムを起動する前に、次のマシンデータで定義してください。

MD24120 \$MC\_TRAFO\_GEOAX\_ASSIGN\_TAB\_X[n]

または

MD20050 \$MC\_AXCONF\_GEOAX\_ASSIGN\_TAB[n]

### 割り当ての変更

変換データセットの変換への割り当ては、MD24100 \$MC\_TRAFO\_TYPE\_X への一連の入力値で決定されます。表の1番目の入力値は最初の変換データセットに、これに従って、2番目の入力値は2番目のデータセットに、という順に割り当てられます。有効な変換のこの割り当てを変更することは許可されません(不可能です)。

例:

3個の変換が次のように設定されています。2個はオリエンテーション変換で、1個は極座標補間です。

MD24100 \$MC\_TRAFO\_TYPE\_1 = 16

;オリエンテーション座標変換 1:オリエンテーション座標変換データセット

MD24200 \$MC\_TRAFO\_TYPE\_2 = 256 :極座標補間

## 8.8 パートプログラム/ソフトキーによる座標変換マシンデータの起動

MD24300 \$MC\_TRAFO\_TYPE\_3 = 18

;オリエンテーション座標変換 2:オリエンテーション座標変換データセット

オリエンテーション変換の 1 番目のデータセットは、1 番目の変換(1 番目のオリエンテーション変換に相当)に割り当てられ、2 番目の変換データセットは、3 番目の変換(2 番目のオリエンテーション変換に相当)に割り当てられます。

「マシンデータの有効化」機能の実行時に 3 番目の変換が有効である場合は、1 番目の変換を別のグループの変換(TRACYL など)に変更できません。変更した場合は、3 番目の変換が、2 番目ではなく、1 番目のオリエンテーション変換になるためです。

ただし、上の例では、2 番目の変換が、MD24200 \$MC\_TRAFO\_TYPE\_2 = 48 などと同時にオリエンテーション変換に変更される場合は、別のオリエンテーション変換を 1 番目の変換に設定する(MD24100 \$MC\_TRAFO\_TYPE\_1 = 32 を使用する、など)か、別のグループの変換を 1 番目の変換として設定する(\$MD24100 \$MC\_TRAFO\_TYPE\_1 = 1024、TRAANG を使用する、など)ことができます。

### 8.8.3 電源投入、モード変更、RESET、ブロック検索、REPOS へのコントロールシステムの動作

次のマシンデータによって、RESET 時(プログラム終了時にも)の応答動作および(または)プログラムの起動時に、自動的に座標変換を選択できます。

MD20110 \$MC\_RESET\_MODE\_MASK

MD20112 \$MC\_START\_MODE\_MASK

および

MD20140 \$MC\_TRAFO\_RESET\_VALUE

動作中の座標変換のマシンデータが変更されていた場合は、これにより、プログラムの終了または起動時などにアラームが発生する場合があります。

この問題を回避するには、NC プログラムによる座標変換の再設定時に、NC プログラムを次のように作成することをお勧めします。

プログラムコード	コメント
N10 TRAFOOF()	; まだ動作中の座標変換を選択します
N20\$MC_TRAFO5_BASE_TOOL_1[0]=0	; マシンデータの入力
N30\$MC_TRAFO5_BASE_TOOL_1[0]=3	
N40\$MC_TRAFO5_BASE_TOOL_1[0]=200	
N130 NEWCONF	; 新たに入力されたマシンデータ
	; 転送
N140 M30	

#### 8.8.4 対応するマシンデータのリスト

有効化できるマシンデータは、次のとおりです。

##### すべての変換

すべての変換に関連するマシンデータ:

- MD24100 \$MC\_TRAFO\_TYPE\_1 ~ MD24480 \$MC\_TRAFO\_TYPE\_10
- MD24110 \$MC\_TRAFO\_AXES\_IN\_1 ~ MD24482 \$MC\_TRAFO\_AXES\_IN\_10
- MD24120 \$MC\_TRAFO\_GEOAX\_ASSIGN\_TAB\_1 to  
MD24484 \$MC\_TRAFO\_GEOAX\_ASSIGN\_TAB\_10

##### 方向座標変換

オリエンテーション変換に関連するマシンデータ:

- MD24550 \$MC\_TRAFO5\_BASE\_TOOL\_1 および  
MD24650 \$MC\_TRAFO5\_BASE\_TOOL\_2
- MD24558 \$MC\_TRAFO5\_JOINT\_OFFSET\_1 および  
MD24658 \$MC\_TRAFO5\_JOINT\_OFFSET\_2
- MD24500 \$MC\_TRAFO5\_PART\_OFFSET\_1 および  
MD24600 \$MC\_TRAFO5\_PART\_OFFSET\_2
- MD24510 \$MC\_TRAFO5\_ROT\_AX\_OFFSET\_1 および  
MD24610 \$MC\_TRAFO5\_ROT\_AX\_OFFSET\_2
- MD24520 \$MC\_TRAFO5\_ROT\_SIGN\_IS\_PLUS\_1 および  
MD24620 \$MC\_TRAFO5\_ROT\_SIGN\_IS\_PLUS\_2
- MD 24530:TRAF05\_NON\_POLE\_LIMIT\_1 および  
MD24630 \$MC\_TRAFO5\_NON\_POLE\_LIMIT\_2
- MD24540 \$MC\_TRAFO5\_POLE\_LIMIT\_1 および  
MD24640 \$MC\_TRAFO5\_POLE\_LIMIT\_2
- MD24570 \$MC\_TRAFO5\_AXIS1\_1 および  
MD24670 \$MC\_TRAFO5\_AXIS1\_2
- MD24572 \$MC\_RAFO5\_AXIS2\_1 および  
MD24672 \$MC\_TRAFO5\_AXIS2\_2
- MD24574 \$MC\_TRAFO5\_BASE\_ORIENT\_1 および  
MD24674 \$MC\_TRAFO5\_BASE\_ORIENT\_2

### 8.8 パートプログラム/ソフトキーによる座標変換マシンデータの起動

- MD24562 \$MC\_TRAFO5\_TOOL\_ROT\_AX\_OFFSET\_1 および  
MD24662 \$MC\_TRAFO5\_TOOL\_ROT\_AX\_OFFSET\_2
- MD24564 \$MC\_TRAFO5\_NUTATOR\_AX\_ANGLE\_1 および  
MD24664 \$MC\_TRAFO5\_NUTATOR\_AX\_ANGLE\_2
- MD24566 \$MC\_TRAFO5\_NUTATOR\_VIRT\_ORIAX\_1 および  
MD24666 \$MC\_TRAFO5\_NUTATOR\_VIRT\_ORIAX\_2

### 極座標補間

極座標補間に関連するマシンデータ:

- MD24920 \$MC\_TRANSMIT\_BASE\_TOOL\_1 および  
MD24970 \$MC\_TRANSMIT\_BASE\_TOOL\_2
- MD24900 \$MC\_TRANSMIT\_ROT\_AX\_OFFSET\_1 および  
MD24950 \$MC\_TRANSMIT\_ROT\_AX\_OFFSET\_2
- MD24910 \$MC\_TRANSMIT\_ROT\_SIGN\_IS\_PLUS\_1 および  
MD24960 \$MC\_TRANSMIT\_ROT\_SIGN\_IS\_PLUS\_2
- MD24911 \$MC\_TRANSMIT\_POLE\_SIDE\_FIX\_1 および  
MD24961 \$MC\_TRANSMIT\_POLE\_SIDE\_FIX\_2

### TRACYL 変換

TRACYL 変換に関連するマシンデータ:

- MD24820 \$MC\_TRACYL\_BASE\_TOOL\_1 および  
MD24870 \$MC\_TRACYL\_BASE\_TOOL\_2
- MD24800 \$MC\_TRACYL\_ROT\_AX\_OFFSET\_1 および  
MD24850 \$MC\_TRACYL\_ROT\_AX\_OFFSET\_2
- MD24810 \$MC\_TRACYL\_ROT\_SIGN\_IS\_PLUS\_1 および  
MD24870 \$MC\_TRACYL\_ROT\_SIGN\_IS\_PLUS\_2
- MD24808 \$MC\_TRACYL\_DEFAULT\_MODE\_1 および  
MD24858 \$MC\_TRACYL\_DEFAULT\_MODE\_2

## 傾斜軸変換

傾斜軸方向性変換に関連するマシンデータ:

- MD24710 \$MC\_TRAANG\_BASE\_TOOL\_1 および  
MD24760 \$MC\_TRAANG\_BASE\_TOOL\_2
- MD24700 \$MC\_TRAANG\_ANGLE\_1 および  
MD24750 \$MC\_TRAANG\_ANGLE\_2
- MD24720 \$MC\_TRAANG\_PARALLEL\_VELO\_RES\_1 および  
MD24770 \$MC\_TRAANG\_PARALLEL\_VELO\_RES\_2
- MD24721 \$MC\_TRAANG\_PARALLEL\_ACCEL\_RES\_1 および  
MD24771 \$MC\_TRAANG\_PARALLEL\_ACCEL\_RES\_2

## 座標変換重畳

座標変換重畳に関連するマシンデータ:

- MD24995 \$MC\_TRACON\_CHAIN\_1 および  
MD24996 \$MC\_TRACON\_CHAIN\_2
- MD24997 \$MC\_TRACON\_CHAIN\_3 および  
MD24998 \$MC\_TRACON\_CHAIN\_4

## モーダル座標変換

モーダル座標変換に関連するマシンデータ:

- MD20144 \$MC\_TRAFO\_MODE\_MASK
- MD20140 \$MC\_TRAFO\_RESET\_VALUE
- MD20110 \$MC\_RESET\_MODE\_MASK  
MD20112 \$MC\_START\_MODE\_MASK

## 共通変換マシンデータ

変換別でないマシンデータこれらは、変換データセット別には割り当てられず、有効な変換の外部では意味を持ちません。

- MD21110 \$MC\_X\_AXIS\_IN\_OLD\_X\_Z\_PLANE
- MD21090 \$MC\_MAX\_LEAD\_ANGLE

## 8.8 パートプログラム/ソフトキーによる座標変換マシンデータの起動

- MD21092 \$MC\_MAX\_TILT\_ANGLE
- MD21100 \$MC\_ORIENTATION\_IS\_EULER

## 8.8.5 例

次の例では、ブロック N90 の 2 番目の座標変換(MD24650 \$MC\_TRAFO5\_BASE\_TOOL\_2[2])に影響するマシンデータの再構成(書くこと)ができます。これは、マシンデータは、書くだけでは有効化されないためです。しかし、もしプログラムが変更されずそのままの場合は、その後に動作中の座標変換の変更が試みられるため、ブロック N130 でアラームが発生します。

## プログラム例:

プログラムコード	コメント
N40 TRAORI(2)	; 2 番目の方向座標変換の選択
N50 X0 Y0 Z0 F20000 T1 T1	
N60 A50 B50	
N70 A0 B0	
N80 X10	
N90 \$MC_TRAFO5_BASE_TOOL_1[2] = 50	; 1 番目の方向座標変換のマシンデータ項目の上書き
N100 A20	
N110 X20	
N120 X0	
N130 NEWCONF	; 新しいマシンデータの受け入れ
N140 TRAORI(1)	; 1 番目の方向座標変換 MD の選択が有効です
N150 G19 X0 Y0 Z0	
N160 A50 B50	
N170 A0 B0	
N180 TRAFOOF	
N190 M30	

## 8.9 データリスト

### 8.9.1 マシンデータ

#### 8.9.1.1 TRANSMIT

チャンネル別マシンデータ

番号	識別子: \$MC_	説明
20110	RESET_MODE_MASK	立ち上げ後および RESET/パートプログラム終了後の制御基本設定の定義
20140	TRAFO_RESET_VALUE	基本変換位置
22534	TRAFO_CHANGE_M_CODE	変換切り替えの M コード
24100	TRAFO_TYPE_1	チャンネルの 1 番目の変換の定義
24110	TRAFO_AXES_IN_1	1 番目の変換の軸割り付け
24120	TRAFO_GEOAX_ASSIGN_TAB_1	1 番目の変換のジオメトリ軸割り付け
24200	TRAFO_TYPE_2	チャンネルの 2 番目の変換の定義
24210	TRAFO_AXES_IN_2	2 番目の変換の軸割り付け
24220	TRAFO_GEOAX_ASSIGN_TAB_2	2 番目の変換のジオメトリ軸割り付け
24300	TRAFO_TYPE_3	チャンネルの 3 番目の変換の定義
24310	TRAFO_AXES_IN_3	3 番目の変換の軸割り付け
24320	TRAFO_GEOAX_ASSIGN_TAB_3	3 番目の変換のジオメトリ軸割り付け
24400	TRAFO_TYPE_4	チャンネルの 4 番目の変換の定義
24410	TRAFO_AXES_IN_4	4 番目の変換の軸割り付け
24420	TRAFO_GEOAX_ASSIGN_TAB_4	4 番目の変換のジオメトリ軸割り付け
24430	TRAFO_TYPE_5	チャンネルの 5 番目の変換の定義
24432	TRAFO_AXES_IN_5	5 番目の変換の軸割り付け
24434	TRAFO_GEOAX_ASSIGN_TAB_5	5 番目の変換のジオメトリ軸割り付け
24440	TRAFO_TYPE_6	チャンネルの 6 番目の変換の定義
24442	TRAFO_AXES_IN_6	6 番目の変換の軸割り付け
24444	TRAFO_GEOAX_ASSIGN_TAB_6	6 番目の変換のジオメトリ軸割り付け

## 8.9 データリスト

番号	識別子: \$MC_	説明
24450	TRAFO_TYPE_7	チャンネルの 7 番目の変換の定義
24452	TRAFO_AXES_IN_7	7 番目の変換の軸割り付け
24454	TRAFO_GEOAX_ASSIGN_TAB_7	7 番目の変換のジオメトリ軸割り付け
24460	TRAFO_TYPE_8	チャンネルの 8 番目の変換の定義
24462	TRAFO_AXES_IN_8	8 番目の変換の軸割り付け
24464	TRAFO_GEOAX_ASSIGN_TAB_8	8 番目の変換のジオメトリ軸割り付け
24900	TRANSMIT_ROT_AX_OFFSET_1	原点位置からの回転軸偏差角度(1 番目の TRANSMIT)(単位 度)
24910	TRANSMIT_ROT_SIGN_IS_PLUS_1	TRANSMIT の回転軸の符号(1 番目の TRANSMIT)
24911	TRANSMIT_POLE_SIDE_FIX_1	極の前側/後側のワーキングエリアリミット、1 番目の変換
24920	TRANSMIT_BASE_TOOL_1	ジオメトリ軸の基点からの工具原点の距離(1 番目の TRANSMIT)
24950	TRANSMIT_ROT_AX_OFFSET_2	原点位置からの回転軸偏差角度(2 番目の TRANSMIT)(単位 度)
24960	TRANSMIT_ROT_SIGN_IS_PLUS_2	TRANSMIT の回転軸の符号(2 番目の TRANSMIT)
24961	TRANSMIT_POLE_SIDE_FIX_2	極の前側/後側のワーキングエリアリミット、2 番目の変換
24970	TRANSMIT_BASE_TOOL_2	ジオメトリ軸の基点からの工具原点の距離(2 番目の TRANSMIT)

## 8.9.1.2 TRACYL

## チャンネル別マシンデータ

番号	識別子: \$MC_	説明
20110	RESET_MODE_MASK	立ち上げ後および RESET/パートプログラム終了後の制御基本設定の定義
20140	TRAFO_RESET_VALUE	基本変換位置
20144	TRAFO_MODE_MASK	キネマティック変換機能の選択
24100	TRAFO_TYPE_1	チャンネルの 1 番目の変換の定義
24110	TRAFO_AXES_IN_1	1 番目の変換の軸割り付け

番号	識別子: \$MC_	説明
24120	TRAFO_GEOAX_ASSIGN_TAB_1	1 番目の変換のジオメトリ軸割り付け
24130	TRAFO_INCLUDES_TOOL_1	動作中の変換 1 での工具の取扱い
24200	TRAFO_TYPE_2	チャンネルの 2 番目の変換の定義
24210	TRAFO_AXES_IN_2	2 番目の変換の軸割り付け
24220	TRAFO_GEOAX_ASSIGN_TAB_2	2 番目の変換のジオメトリ軸割り付け
24230	TRAFO_INCLUDES_TOOL_2	動作中の変換 2 での工具の取扱い
24300	TRAFO_TYPE_3	チャンネルの 3 番目の変換の定義
24310	TRAFO_AXES_IN_3	3 番目の変換の軸割り付け
24320	TRAFO_GEOAX_ASSIGN_TAB_3	3 番目の変換のジオメトリ軸割り付け
24330	TRAFO_INCLUDES_TOOL_3	動作中の変換 3 での工具の取扱い
24400	TRAFO_TYPE_4	チャンネルの 4 番目の変換の定義
24410	TRAFO_AXES_IN_4	4 番目の変換の軸割り付け
24420	TRAFO_GEOAX_ASSIGN_TAB_4	4 番目の変換のジオメトリ軸割り付け
24426	TRAFO_INCLUDES_TOOL_4	動作中の変換 4 での工具の取扱い
24430	TRAFO_TYPE_5	チャンネルの 5 番目の変換の定義
24432	TRAFO_AXES_IN_5	5 番目の変換の軸割り付け
24434	TRAFO_GEOAX_ASSIGN_TAB_5	5 番目の変換のジオメトリ軸割り付け
24436	TRAFO_INCLUDES_TOOL_5	動作中の変換 5 での工具の取扱い
24440	TRAFO_TYPE_6	チャンネルの 6 番目の変換の定義
24442	TRAFO_AXES_IN_6	6 番目の変換の軸割り付け
24444	TRAFO_GEOAX_ASSIGN_TAB_6	6 番目の変換のジオメトリ軸割り付け
24446	TRAFO_INCLUDES_TOOL_6	動作中の変換 6 での工具の取扱い
24450	TRAFO_TYPE_7	チャンネルの 7 番目の変換の定義
24452	TRAFO_AXES_IN_7	7 番目の変換の軸割り付け
24454	TRAFO_GEOAX_ASSIGN_TAB_7	7 番目の変換のジオメトリ軸割り付け
24456	TRAFO_INCLUDES_TOOL_7	動作中の変換 7 での工具の取扱い
24460	TRAFO_TYPE_8	チャンネルの 8 番目の変換の定義
24462	TRAFO_AXES_IN_8	8 番目の変換の軸割り付け
24464	TRAFO_GEOAX_ASSIGN_TAB_8	8 番目の変換のジオメトリ軸割り付け
24466	TRAFO_INCLUDES_TOOL_8	動作中の変換 8 での工具の取扱い

## 8.9 データリスト

番号	識別子: \$MC_	説明
24470	TRAFO_TYPE_9	チャンネルの 9 番目の変換の定義
24472	TRAFO_AXES_IN_9	9 番目の変換の軸割り付け
24474	TRAFO_GEOAX_ASSIGN_TAB_9	9 番目の変換のジオメトリ軸割り付け
24476	TRAFO_INCLUDES_TOOL_9	動作中の変換 9 での工具の取扱い
24480	TRAFO_TYPE_10	チャンネルの 10 番目の変換の定義
24482	TRAFO_AXES_IN_10	10 番目の変換の軸割り付け
24484	TRAFO_GEOAX_ASSIGN_TAB_10	10 番目の変換のジオメトリ軸割り付け
24486	TRAFO_INCLUDES_TOOL_10	動作中の変換 10 での工具の取扱い
24800	TRACYL_ROT_AX_OFFSET_1	原点位置からの回転軸偏差角度(1 番目の TRACYL) (単位 度)
24808	TRACYL_DEFAULT_MODE_1	TRACYL モードの選択(1 番目の TRACYL)
24810	TRACYL_ROT_SIGN_IS_PLUS_1	TRACYL の回転軸の符号(1 番目の TRACYL)
24820	TRACYL_BASE_TOOL_1	ジオメトリ軸の基点からの工具原点の距離(1 番目の TRACYL)
24850	TRACYL_ROT_AX_OFFSET_2	原点位置からの回転軸偏差角度(2 番目の TRACYL) (単位 度)
24858	TRACYL_DEFAULT_MODE_2	TRACYL モードの選択(2 番目の TRACYL)
24860	TRACYL_ROT_SIGN_IS_PLUS_2	TRACYL の回転軸の符号(2 番目の TRACYL)
24870	TRACYL_BASE_TOOL_2	ジオメトリ軸の基点からの工具原点の距離(2 番目の TRACYL)
22534	TRAFO_CHANGE_M_CODE	変換切り替えの M コード

## 8.9.1.3 TRAANG

## チャンネル別マシンデータ

番号	識別子: \$MC_	説明
20110	RESET_MODE_MASK	立ち上げ後および RESET/パートプログラム終了後の制御基本設定の定義
20140	TRAFO_RESET_VALUE	基本変換位置
20144	RAFO_MODE_MASK	キネマティック変換機能の選択
20534	TRAFO_CHANGE_M_CODE	変換切り替えの M コード

番号	識別子: \$MC_	説明
24100	TRAFO_TYPE_1	チャンネルの 1 番目の変換の定義
24110	TRAFO_AXES_IN_1	1 番目の変換の軸割り付け
24120	TRAFO_GEOAX_ASSIGN_TAB_1	1 番目の変換のジオメトリ軸割り付け
24200	TRAFO_TYPE_2	チャンネルの 2 番目の変換の定義
24210	TRAFO_AXES_IN_2	2 番目の変換の軸割り付け
24220	TRAFO_GEOAX_ASSIGN_TAB_2	2 番目の変換のジオメトリ軸割り付け
24300	TRAFO_TYPE_3	チャンネルの 3 番目の変換の定義
24310	TRAFO_AXES_IN_3	3 番目の変換の軸割り付け
24320	TRAFO_GEOAX_ASSIGN_TAB_3	3 番目の変換のジオメトリ軸割り付け
24400	TRAFO_TYPE_4	チャンネルの 4 番目の変換の定義
24410	TRAFO_AXES_IN_4	4 番目の変換の軸割り付け
24420	TRAFO_GEOAX_ASSIGN_TAB_4	4 番目の変換のジオメトリ軸割り付け
24430	TRAFO_TYPE_5	チャンネルの 5 番目の変換の定義
24432	TRAFO_AXES_IN_5	5 番目の変換の軸割り付け
24434	TRAFO_GEOAX_ASSIGN_TAB_5	5 番目の変換のジオメトリ軸割り付け
24440	TRAFO_TYPE_6	チャンネルの 6 番目の変換の定義
24442	TRAFO_AXES_IN_6	6 番目の変換の軸割り付け
24444	TRAFO_GEOAX_ASSIGN_TAB_6	6 番目の変換のジオメトリ軸割り付け
24450	TRAFO_TYPE_7	チャンネルの 7 番目の変換の定義
24452	TRAFO_AXES_IN_7	7 番目の変換の軸割り付け
24454	TRAFO_GEOAX_ASSIGN_TAB_7	7 番目の変換のジオメトリ軸割り付け
24460	TRAFO_TYPE_8	チャンネルの 8 番目の変換の定義
24462	TRAFO_AXES_IN_8	8 番目の変換の軸割り付け
24464	TRAFO_GEOAX_ASSIGN_TAB_8	8 番目の変換のジオメトリ軸割り付け
24700	TRAANG_ANGLE_1	傾斜軸の角度(1 番目の TRAANG)(単位 度)
24710	TRAANG_BASE_TOOL_1	ジオメトリ軸の基点からの工具原点の距離(1 番目の TRAANG)
24720	TRAANG_PARALLEL_VELO_RES_1	補正移動の平行軸の速度制限(1 番目の TRAANG)
24721	TRAANG_PARALLEL_VELO_RES_2	補正移動の平行軸の速度制限(2 番目の TRAANG)
24750	TRAANG_ANGLE_2	傾斜軸の角度(2 番目の TRAANG)(単位 度)

## 8.9 データリスト

番号	識別子: \$MC_	説明
24760	TRAANG_BASE_TOOL_2	ジオメトリ軸の基点からの工具原点の距離(2番目のTRAANG)
24770	TRAANG_PARALLEL_ACCEL_RES_1	補正移動の平行軸の軸加速度制限(1番目のTRAANG)
24771	TRAANG_PARALLEL_ACCEL_RES_2	補正移動の平行軸の軸加速度制限(2番目のTRAANG)

## 8.9.1.4 座標変換重畳

## チャンネル別マシンデータ

番号	識別子: \$MC_	説明
24995	TRACON_CHAIN_1	1番目の座標変換重畳の座標変換重畳
24996	TRACON_CHAIN_2	2番目の座標変換重畳の座標変換重畳
24997	TRACON_CHAIN_3	3番目の座標変換重畳の座標変換重畳
24998	TRACON_CHAIN_4	4番目の座標変換重畳の座標変換重畳

## 8.9.1.5 共通変換マシンデータ

## チャンネル別マシンデータ

番号	識別子: \$MC_	説明
21110	X_AXIS_IN_OLD_X_Z_PLANE	自動フレーム定義の座標系
21090	MAX_LEAD_ANGLE	旋回プログラミングの最大許容リード角
21092	MAX_TILT_ANGLE	旋回プログラミングの最大許容サイド角
21100	ORIENTATION_IS_EULER	旋回プログラミングの角度定義

## 8.9.2 信号

### 8.9.2.1 チャンネルへの信号

信号名称	SINUMERIK 840D sl	SINUMERIK 828D
PTP 移動を起動します	DB21、...DBX29.4	-

### 8.9.2.2 チャンネルからの信号

信号名称	SINUMERIK 840D sl	SINUMERIK 828D
座標変換有効	DB21, ... .DBX33.6	DB330x.DBX1.6
PTP 移動が有効	DB21、...DBX317.6	-



## M5:計測

### 9.1 概略説明

#### チャンネル別計測

パートプログラムブロックのチャンネル別計測に対してプログラミング指令されたトリガ事象は、計測処理を開始し、計測に使用される計測方法を指定します。この命令は、この特定のブロックにプログラム指令されたすべての軸に適用されます。

#### 軸の計測

軸の計測の場合には、パートプログラムとシンクロナイズドアクションからも計測をおこなうことができます。計測方法、エンコーダ、およびトリガ事象がプログラム指令されます。これによってトリガ事象は、プローブ(1と2)とトリガ条件(信号の立ち上り/立ち下がり)から形成されます。計測処理に対応して、計測とトリガ事象毎に複数の計測値を記録できます。

#### 現在位置設定と接触計測

現在位置設定は、HMI オペレータ操作によって起動されます。計算されたフレームは、システムフレーム\$P\_SETFRAME に書き込み可能です。現在位置設定がされている場合は、WCS の軸の指令位置を変更できます。

PI サービスが、次のいずれかによって有効になると、NC で計算がおこなわれます。

- HMI オペレータ操作 または
- 計測サイクルからのパートプログラム命令

接触計測という用語は、ワーク計測と工具計測の両方を意味します。計測は、次のいずれかによって開始できます。

- HMI オペレータ操作 または
- 計測サイクル

NC との通信が、予約システム変数によっておこなわれます。

## 9.2 ハードウェアの必要条件

### ワーク計測と工具計測

エッジ、コーナー、または穴に関連するワークの位置を計測できます。

ワークの原点位置(ワーク原点 **W**)または穴を決定するために、ワーク座標系で計測された位置に指令位置を追加できます。その結果のオフセットを、選択したフレームに入力できます。

工具計測の場合は、コントローラが、ユーザが指定した工具長から、工具先端と工具キヤリヤレファレンス点 **T** との間の距離を計算します。

### 計測サイクル

計測サイクルの取扱いの方法の説明は、次の資料にあります。

#### 参照先

『プログラミングマニュアル』、「計測サイクルの説明」

## 9.2 ハードウェアの必要条件

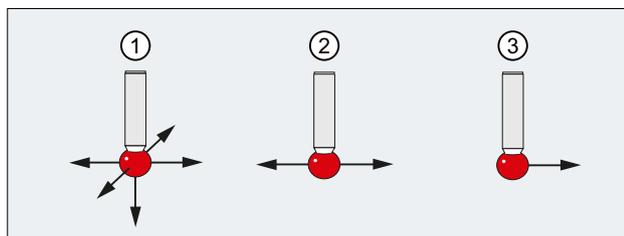
### 9.2.1 使用可能なプローブ

#### 概要

SINUMERIK 制御装置の「計測」機能には、変位に関する一定のバウンスフリー信号を提供する切り替えプローブを使用してください。変位時に単一のパルスを提供するプローブには対応しません。

#### プローブタイプ

以下のプローブタイプは、考えられる変位の方向の数に応じて異なります。



## ① 多方向プローブ(3次元)

多方向プローブは、工具およびワーク寸法の計測に無条件に使用できます。

## ③ 双方向プローブ

双方向プローブは、フライス盤およびマシニングセンタでワーク計測のために一方向プローブのように扱われます。ワーク計測用の双方向プローブは、旋盤に使用できます。

## ③ 一方向プローブ

一方向プローブは、制限が少ないフライス盤やマシニングセンタでのワーク計測にのみ使用できます。

## 主軸位置と一方向プローブ

フライス盤およびマシニングセンタで一方向プローブを使用するには、SPOS 機能、および 360°にわたって伝送されるプローブの切替え信号により、主軸の位置決めを行う必要があります。

0°の主軸位置で、下記の方向の計測を可能にするために、主軸内でプローブの機械的な割り出しを行う必要があります:

平面	計測方向
G17 (X-Y)	正の X 方向
G18 (Z-X)	正の Z 方向
G19 (Y-Z)	正の Y 方向

## 注記

全計測には時間がかかります。これは、計測サイクル中に主軸を複数回、SPOS で位置決めする必要があるためです。

## 9.3 チャンネル別測定

### 9.3.1 計測

#### 適用

計測は、パートプログラムから起動します。トリガ事象と計測方法は、プログラム指令します。

以下の異なる 2 つの計測方法があります。

- MEAS: 残移動距離を削除する計測

例:

```
N10 G01 F300 X300 Z200 MEAS=-2
```

トリガ事象は、2 番目のプローブ(2)の立下り(-)です。

- MEAW: 残移動距離を削除しない計測

例:

```
N20 G01 F300 X300 Y100 MEAW=1
```

トリガ事象は、最初のプローブ(1)の立ち上がりです。

計測ジョブは、リセット時またはプログラムが新しいブロックに進んだときに中止されます。

---

#### 注記

ジオメトリ軸が計測ブロック内でプログラム指令された場合、現在のすべてのジオメトリ軸について計測値が保存されます。

座標変換に関与する軸が計測ブロックにプログラム指令されている場合は、この座標変換に関与するすべての軸の計測値が記録されます。

---

#### プローブ状態

パートプログラムとシンクロナイズドアクションでは、プローブ状態を直接スキャンできません。

`$A_PROBE[n]`、`n`=プローブ番号

`$A_PROBE[n]==1`: プローブのスイッチがオン

`$A_PROBE[n]==0`: プローブのスイッチがオフ

## 9.3.2 計測結果

### 計測結果の読み取り

計測命令の結果は、NC のシステムデータに保存され、パートプログラムのシステム変数によって読み取ることができます。

- **システム変数\$AC\_MEA[番号]**  
計測ジョブ状態信号を確認します。  
[番号]はプローブ(1 または 2)を表します。  
変数は、計測の開始時に削除されます。変数は、プローブが起動条件(立ち上がりまたは立下り)を満たすと同時に設定されます。したがって、計測ジョブの実行をパートプログラムでチェックできます。
- **システム変数\$AA\_MM[軸]**  
機械座標系(MCS)の計測値にアクセスします。  
パートプログラムとシンクロナイズドアクションで読み取ります。  
[軸]は計測軸(X、Y、...)の名称を表します。
- **システム変数\$AA\_MW[軸]**  
ワーク座標系の計測値にアクセスします。  
パートプログラムとシンクロナイズドアクションで読み取ります。  
[軸]は計測軸(X、Y、...)の名称を表します。

### PLC サービス表示

プローブの機能テストは、NC プログラムを使用しておこなわれます。

計測信号は、診断メニュー[ PLC 状態]のなかで、プログラム終了時点でチェックできます。

表 9-1 計測信号の状態表示

	状態表示
プローブ 1 信号スイッチオン	DB10、...DBX107.0
プローブ 2 信号スイッチオン	DB10、...DBX107.1

軸の現在の計測状態は、インタフェース信号 DB31、... DBX62.3 によって表示されます。

ビット 3 = 1:計測が  
有効 ビット 3=0:計測が無効

## 9.4 軸別の計測

この信号は、すべての計測機能に対して表示され、次のシステム変数で、シンクロナイズドアクションでも読み込むことができます。

- システム変数\$AA\_MEA<sub>ACT</sub>[軸]

参照先:

『機能マニュアル、シンクロナイズドアクション』

## 9.4 軸別の計測

### 9.4.1 計測

軸別の計測は、NC プログラムまたはシンクロナイズドアクションで命令 MEAC を使用して有効にします。残移動距離を削除する計測と削除しない計測を実行できます。得られる計測値は、機械座標系(MCS)を基準としています。計測値は、計測を有効にしたときに指定した FIFO 原理に従ってリングバッファに保存されます。計測に 2 つのプローブを設定した場合は、2 番目のプローブの計測値が自動的に次のリングバッファに書き込まれます。

軸別の計測では、4 つまでのトリガ事象と計測モード、つまり、時間設定またはプログラム指令されたトリガ事象シーケンスを指定できます。これを使用して計測が実行されます。連続計測の場合は、計測値の保存先となるリングバッファ(FIFO)の番号も指定してください。

以下の計測方法を使用できます。

- 残移動距離を削除する計測、MEASA
- 残移動距離を削除しない計測、MEAWA
- 残移動距離を削除しない連続計測、MEAC

### 計測モードと検出器

計測モードにより、複数のトリガ事象を並行して有効にするか、昇順で順番に有効にするかを指定されて、受け付ける計測数が定義されます。

- 以下の計測モードを使用できます。
  - 計測ジョブのキャンセル
  - 4種類までのトリガ事象を同時に有効化できます。
  - 4種類までのトリガ事象を連続して有効化できます。1番目のトリガ事象がすでに有効である場合はエラーが発生します。
  - 4種類までのトリガ事象を連続して有効化できます。1番目のトリガ事象がすでに有効であってもエラーは発生しません。
- 計測は、以下の検出器を使用して実行できます。
  - 有効な検出器
  - 1番目の検出器
  - 2番目の検出器
  - 1番目と2番目の検出器

### トリガ事象

計測器が1つ保存されている、以下のトリガ事象も可能です。

- プローブ1の立ち上がり
- プローブ1の立ち下がり
- プローブ2の立ち上がり
- プローブ2の立ち下がり

2つの検出器により計測処理をおこなう場合、最大で2つのトリガ事象をプログラム指令できます(立ち上りまたは立ち下がり)。両方のプローブの計測値が、2つのトリガ事象の両方について取得されます。PROFIBUS 通信メッセージ 391 の初期設定の場合は、各トリガ事象および位置コントローラクロック周期について取得できる計測値は1つです。

MEAC の場合、PROFIBUS メッセージ 395 の使用時には、トリガ事象あたりの計測値数を、各トリガ事象および位置コントローラクロック周期について立ち上がりに8つ、立ち下がりに8つまで増やすことができます。

- 1つのプローブ:立ち上がりに8つ、立ち下がりに8つの計測値
- 2つのプローブ:立ち上がりに4つ、立ち下がりに4つの計測値

つまり、PROFIBUS メッセージ 395 の使用時には、より速い送り速度や速度を使用できます。

メッセージの選択について詳しくは、「メッセージ選択 (ページ 629)」の章を参照してください。

## システム変数:FIFO 変数(\$AC\_FIFO&lt;n&gt;)

FIFO 変数\$AC\_FIFO<n>を R 変数内で使用して軸別計測値を保存すると、NC が管理する、特別なデータ構造が提供されます。これらは、FIFO 原理(先入れ先出し)に従って機能するリングバッファとして編成されています。

## マシンデータ

- **Start:**R 変数内のリングバッファの開始は、以下を使用して定義します。  
MD28262 \$MC\_START\_AC\_FIFO = <開始 R 変数番号>

## 注記

## 使用可能な R 変数の制限

開始 R 変数より小さい番号の R 変数はすべて、NC プログラムとシンクロナイズドアクションで書き込むことができます。

開始 R 変数より大きい番号の R 変数はすべて、NC プログラムとシンクロナイズドアクションで書き込むことができなくなります。

- **Number:**リングバッファの番号は、以下を使用して定義します。  
MD28260 \$MC\_NUM\_AC\_FIFO
- **Length:**リングバッファの長さは、以下を使用して定義します。  
MD28264 LEN\_AC\_FIFO  
すべてのリングバッファは同じ長さです。
- リングバッファの R 変数を考慮して、必要な R 変数の数は、以下を使用して定義します。  
MD28050 \$MC\_MM\_NUM\_R\_PARAM = MD28262 \$MC\_START\_AC\_FIFO +  
MD28260 \$MC\_NUM\_AC\_FIFO \* (MD28264 \$MC\_LEN\_AC\_FIFO + 6)

## 計測値の読み取り

NC プログラムとシンクロナイズドアクションで、FIFO 変数\$AC\_FIFO<n>を使用して計測値を読み取ることができます。

## 計測の終了

次の事象のいずれかが発生すると、計測が終了します。

- NC プログラムまたはシンクロナイズドアクション内のプログラミング:MEAC [<軸>]  
= 0
- リングバッファがいっぱいです
- チャンネルリセット
- プログラム終了 M02/M30

**参照先**

『機能マニュアル、シンクロナイズドアクション』の章:「詳細説明」>「シンクロナイズドアクションの変数」>「パラメータ\$AC\_FIFO」

**システム変数:プローブ状態(\$A\_PROBE)**

NC プログラムまたはシンクロナイズドアクションで、システム変数\$A\_PROBE を使用してプローブ状態を読み取ることができます。

状態= \$A\_PROBE[<n>]; ここで、n =プローブ番号

状態	意味
1	プローブのスイッチがオン
0	プローブのスイッチがオフ

**システム変数:プローブ制限(\$A\_PROBE\_LIMITED)**

NC プログラムまたはシンクロナイズドアクションで、PROFIBUS メッセージ 395 の使用時にシステム変数\$A\_PROBE\_LIMITED を使用してプローブ制限状態を読み取ることができます。

状態= \$A\_PROBE\_LIMITED[<n>]; ここで、n =プローブ番号

状態	意味
1	プローブの制限の有効化
0	プローブの制限の無効化/リセット

**プログラミング**

この機能のプログラミングには、キーワード MEASA、MEAWA、および MEAC を使用できます。

プログラム指令軸に MEASA または MEAWA を使用する場合は、各計測で最高 4 つの計測値が取得され、トリガ事象に対応したシステム変数に保存されます。

連続計測ジョブは MEAC で実行できます。この場合は、計測結果が FIFO 変数 \$AC\_FIFO<n>に保存されます。

**構文**

MEASA [<軸>] = (<モード>, <TE1>, ..., <TE4>)

MEAWA [<軸>] = (<モード>, <TE1>, ..., <TE4>)

MEAC [<軸>] = (<モード>, <計測メモリ>, <TE1>, ..., <TE4>)

**注記**

MEASA と MEAWA はノンモーダルで、1ブロックと一緒にプログラム指令できます。

MEASA/MEAWA および MEAS/MEAW を一緒に1ブロックでプログラム指令することは許可されません。

**意味**

MEASA:	残移動距離を削除する軸別計測	
	<b>注</b> シンクロナイズドアクションではプログラム指令できません。	
	有効性:	ノンモーダル
MEAWA:	残移動距離を削除しない軸別計測	
	有効性:	ノンモーダル
MEAC:	残移動距離を削除しない軸別連続計測	
	有効性:	ノンモーダル
<軸>:	計測に使用するチャンネル軸の名称	

<モード>:	計測モードと検出器を指定する 2 桁の数	
	1 の位:計測モード	
	0	計測ジョブをキャンセルします。
	1	4 種類までのトリガ事象を同時に有効化できます。
	2	4 種類までのトリガ事象を連続して有効化できます。
	3	4 種類までのトリガ事象を連続して有効にできますが、起動時にトリガ事象 1 は監視されません(アラーム 21700/21703 はマスクされます)。 注: MEAC は、このモードをサポートしていません。
	10 の位:検出器	
	0	有効な検出器
	1	検出器 1
	2	検出器 2 注 2 番目の検出器を使用できない場合は、1 番目の検出器が使用されます。アラームは表示されません。
3	両方の検出器	
<TE>:	計測を開始するトリガ事象	
	タイプ:	INT
	値の範囲:	-2, -1, 1, 2
	意味	
	(+)1	プローブ 1 の立ち上がり
	-1	プローブ 1 の立ち下がり
	(+)2	プローブ 2 の立ち上がり
	-2	プローブ 2 の立ち下がり
<計測メモリ>:	FIFO (リングバッファ)の数	

## 例

- 残移動距離を削除する計測、MEASA
  - 計測モード:1、4種類までのトリガ事象を同時に有効化できます。
  - 検出器:0、有効な検出器
  - トリガ事象 1:1、X=100 への移動軌跡上の1番目のプローブ(1)の立ち上がり
  - トリガ事象 2:-1、X=100 への移動軌跡上の1番目のプローブ(1)の立ち下がり

MEASA[X]=(1,1,-1) G01 X100 F100
- 残移動距離を削除しない計測、MEAWA
  - 計測モード:1、4種類までのトリガ事象を同時に有効化できます。
  - 検出器:0、有効な検出器
  - トリガ事象 1:-1、X=100 への移動軌跡上の1番目のプローブ(1)の立ち下がり
  - トリガ事象 2:1、X=100 への移動軌跡上の1番目のプローブ(1)の立ち上がり
  - トリガ事象 3:-2、X=100 への移動軌跡上の1番目のプローブ(2)の立ち下がり
  - トリガ事象 4:2、X=100 への移動軌跡上の1番目のプローブ(2)の立ち上がり

MEAWA[X]=(1,-1,1,-2,2) G01 X100 F100:
- 残移動距離を削除しない連続計測、MEAC
  - 計測モード:1、4種類までのトリガ事象を同時に有効化できます。
  - 検出器:0、有効な検出器
  - 測定値メモリ:リングバッファ 1、\$AC\_FIFO1
  - トリガ事象 1:-2、X=100 への移動軌跡上の1番目のプローブ(2)の立ち下がり
  - トリガ事象 2:2、X=100 への移動軌跡上の1番目のプローブ(2)の立ち上がり

MEAC[X]=(1,1,-2,2) G01 X100 F100

## 例 1:モード1の残移動距離を削除する軸別計測

例での評価は、時間の順に実行されます。

表 9-2 1つの検出器

プログラムコード	コメント
N100 MEASA[X]=(1,1,-1) G01 X100 F100	; 有効な検出器によるモード1の計測。プローブ1からの立ち上がり/立ち下がり、x=100への移動軌跡の計測信号を待機します。
N110 STOPRE	; 先読み停止
N120 IF \$AC_MEA[1]==FALSE GOTOF ENDE	; 計測が正常におこなわれたことを確認します。
N130 R10=\$AA_MM1[X]	; 1番目にプログラム指令のトリガ事象(立ち上がり)で得られた計測値を保存します。
N140 R11=\$AA_MM2[X]	; 2番目にプログラム指令のトリガ事象(立ち下がり)で得られた計測値を保存します。
N150 END:	

表 9-3 2つの検出器

プログラムコード	コメント
N200 MEASA[X]=(31,1,-1) G01 X100 F100	; 両方の検出器によるモード1の計測。プローブ1からの立ち上がり/立ち下がり、X=100への移動軌跡の計測信号を待機します。
N210 STOPRE	; 先読み停止
N220 IF \$AC_MEA[1]==FALSE GOTOF END	; 計測が正常におこなわれたことを確認します。
N230 R10=\$AA_MM1[X]	; 立ち上がりで検出器1の計測値を保存します。
N240 R11=\$AA_MM2[X]	; 立ち上がりで検出器2の計測値を保存します。
N250 R12=\$AA_MM3[X]	; 立ち下がり検出器1の計測値を保存します。
N260 R13=\$AA_MM4[X]	; 立ち下がり検出器2の計測値を保存します。
N270 END:	

### 例 2:モード2の残移動距離を削除する軸別計測

例での評価は、プログラミングされた順に実行されます。

プログラムコード	コメント
N100 MEASA[X]=(2,1,-1,2,-2) G01 X100 F100	; 有効な検出器によるモード2の計測。X=100への移動軌跡の間、立ち上がりプローブ1、立ち下がりプローブ1、立ち上がりプローブ2、立ち下がりプローブ2という順に計測信号を待機します。
N110 STOPRE	; 先読み停止
N120 IF \$AC_MEA[1]==FALSE GOTOF PROBE2	; プローブ1による計測が正常におこなわれたことを確認します。
N130 R10=\$AA_MM1[X]	; 1番目にプログラム指令のトリガ事象(プローブ1の立ち上がり)で得られた計測値を保存します。
N140 R11=\$AA_MM2[X]	; 2番目にプログラム指令のトリガ事象(プローブ1の立ち上がり)で得られた計測値を保存します。
N150 PROBE2:	
N160 IF \$AC_MEA[2]==FALSE GOTOF END	; プローブ2による計測が正常におこなわれたことを確認します。
N170 R12=\$AA_MM3[X]	; 3番目にプログラム指令のトリガ事象(プローブ2の立ち上がり)で得られた計測値を保存します。
N180 R13=\$AA_MM4[X]	; 4番目にプログラム指令のトリガ事象(プローブ2の立ち上がり)で得られた計測値を保存します。
N190 END:	

### 例 3:モード1の軸別の連続計測

例での評価は、時間の順に実行されます。

表 9-4 100個までの計測値の計測

プログラムコード	コメント
N110 DEF REAL MEASVALUE[100]	

## 9.4 軸別の計測

プログラムコード	コメント
N120 DEF INT loop=0	
N130 MEAC[X]=(1,1,-1) G01 X1000 F100	; 有効な検出器によるモードの計測。\$AC_FIFO1 に計測値を保存します。X=1000 への移動軌跡の間に、検出器のプローブ 1 の立ち下がりを待機します。
N135 STOPRE	
N140 MEAC[X]=(0)	; 軸位置に到達すると、計測を終了します。
N150 R1=\$AC_FIFO1[4]	; パラメータ R1 に計測値の累計個数を保存します。
N160 FOR loop=0 TO R1-1	
N170 MEASURED VALUE[loop]=\$AC_FIFO1[0]	; \$AC_FIFO1 から計測値を読み出し、保存します。
N180 ENDFOR	

表 9-5 10 個の計測値を得た後の、残移動距離を削除する計測

プログラムコード	コメント
N10 WHEN \$AC_FIFO1[4]>=10 DO MEAC[x]=(0) DELDTG(x)	; 残移動距離削除
N20 MEAC[x]=(1,1,1,-1) G01 X100 F500	
N30 MEAC[X]=(0)	
N40 R1=\$AC_FIFO1[4]	; 計測値の数
...	

表 9-6 2つのプローブによる立ち下がり/立ち上がり歯面の計測

プログラムコード	コメント
N110 DEF REAL MEASURED VALUE[16]	
N120 DEF INT loop=0	
N130 MEAC[X]=(1,1,-1,2) G01 X100 F100	; 有効な検出器によるモード 1 の計測。\$AC_FIFO1 により計測値を保存します。X=100 への移動軌跡の間に、プローブ 1 の立ち下がり、プローブ 2 の立ち上がりの順に計測信号を待機します。
N140 STOPRE	; 先読み停止
N150 MEAC[X]=(0)	; 軸位置に到達すると、計測を終了します。
N160 R1=\$AC_FIFO1[4]	; パラメータ R1 に計測値の累計個数を保存します。
N170 FOR loop=0 TO R1-1	
N180 MEASURED VALUE[loop]=\$AC_FIFO1[0]	; \$AC_FIFO1 から計測値を読み出し、保存します。
N190 ENDFOR	

## 9.4.2 メッセージ選択

### MEAC による軸計測のためのメッセージ選択

初期設定では、PROFIBUS メッセージ 391 により軸の計測が実施されます。PROFIBUS メッセージ 395 は、トリガ事象毎、および位置制御周期毎の複数の計測値で計測をおこなうために使用します。

メッセージ選択は、STEP 7 HW Config の[DP スレーブ特性|設定]対話画面でおこないます。

次の設定も必要です。

- ドライブパラメータ  
CU: p0922 = 395; メッセージ選択の設定  
CU: p0684 = 16; 計測手順の設定  
CU: p0680; セントラルプローブ端子の設定
- PROFIBUS 接続:  
MD13211 \$MN\_MEAS\_CENTRAL\_SOURCE = 2 (応答確認のない組み込みメッセージ)

### 9.4.3 測定結果

#### MEASA、MEAWA の計測結果の読み取り

計測命令の結果は、NC のシステムデータに保存され、パートプログラムのシステム変数によって読み取ることができます。

- システム変数\$SAC\_MEA[番号]  
計測ジョブ状態信号を確認します。  
[番号]はプローブ(1 または 2)を表します。  
変数は、計測の開始時に削除されます。変数は、プローブが起動条件(立ち上がりまたは立下り)を満たすと同時に設定されます。したがって、計測ジョブの実行をパートプログラムでチェックできます。
- システム変数\$AA\_MM1[軸]～\$AA\_MM4[軸]  
機械座標系でのトリガ信号の計測値へのアクセス。パートプログラムとシンクロナイズドアクションで読み取ります。  
[軸]は計測軸(X、Y、...)の名称を表します。
- システム変数\$AA\_MW1[軸]～\$AA\_MW4[軸]  
ワーク座標系でのトリガ信号の計測値へのアクセス。パートプログラムとシンクロナイズドアクションで読み取ります。  
[軸]は計測軸(X、Y、...)の名称を表します。

#### 2つの検出器

2つの検出器により計測ジョブを行う場合、最大で2つのトリガ事象をプログラム指令できます。両方のプローブの計測値は、2つのプローブのそれぞれについて取得されません。

##### 1つのトリガ事象

\$AA\_MM1[軸] = トリガ事象 1、エンコーダ 1 からの計測値

\$AA\_MM2[軸] = トリガ事象 1、エンコーダ 2 からの計測値

##### 2つのトリガ事象

\$AA\_MM1[軸] = トリガ事象 1、エンコーダ 1 からの計測値

\$AA\_MM2[軸] = トリガ事象 1、エンコーダ 2 からの計測値

\$AA\_MM3[軸] = トリガ事象 2、エンコーダ 1 からの計測値

\$AA\_MM4[軸] = トリガ事象 2、エンコーダ 2 からの計測値

## PLC サービス表示

プローブの機能テストの詳細については、「計測結果 (ページ 619)」の章を参照してください。

## MEAC の計測結果の読み取り

すべての計測は、予約 FIFO 変数に書き込まれます。可能な計測値の数は、マシンデータを使用して定義されます(「計測 (ページ 620)」の章を参照してください)。

- IPO サイクル/位置制御周期の比率 $\leq 8$ :の PROFIBUS メッセージ 391 に対して、正しい処理モードが高い信頼性で実行されます。1.
- IPO サイクル/位置制御周期の比率 $\leq 4$ :の PROFIBUS メッセージ 395 に対して、正しい処理モードが高い信頼性で実行されます。1.
- FIFO メモリの内容が読み込まれるのは 1 回だけです。計測結果が 2 回以上使用されるときは、読み取られた値をユーザデータに一時的に記憶してください。

## エンドレス計測

エンドレス計測を実行するには、FIFO 値をパートプログラムから繰り返し読み取ってください。FIFO メモリからの計測値の読み取りと処理に使用される頻度は、NC の書き込み頻度に合わせる必要があります。

有効入力値の数は FIFO 変数で読み取ることができます。

計測値が定義された数に達した後で計測が終了する場合、プログラム内で計測ジョブを明確に解除してください。

## 9.5 原点の設定、ワーク測定、および工具測定

### 9.5.1 現在位置設定と接触計測

#### 現在位置設定

現在位置設定は、HMI オペレータ操作、または計測サイクルによって開始されます。計算されたフレームは、システムフレーム \$P\_SETFRAME に書き込み可能です。現在位置設定がされている場合は、WCS の軸の指令位置を変更できます。

PI サービスが、次のいずれかによって有効になると、NC で計算がおこなわれます。

- HMI オペレータ操作 または
- 計測サイクルからのパートプログラム命令

工具と平面を、計算の基準として選択できます。計算されたフレームは、フレーム結果に入力されます。

## 接触計測

**接触計測**という用語は、ワーク計測と工具計測の両方を意味します。エッジ、コーナー、または穴に関連するワークの位置を計測できます。ワークの原点位置または穴を決定するために、WCS で計測された位置に指令位置を追加できます。その結果のオフセットを、選択したフレームに入力できます。工具計測では、工具の長さまたは半径を、計測基準部品を使用して計測できます。

計測は、次のいずれかによって開始できます。

- HMI オペレータ操作 または
- 計測サイクル

NC との通信が、予約システム変数によっておこなわれます。PI サービスが、次のいずれかによって有効になると、NC で計算が行われます。

- HMI オペレータ操作
- または、計測サイクルからのパートプログラム命令

工具と平面を、計算の基準として選択できます。計算されたフレームは、フレーム結果に入力されます。

チャンネル別システムフレームの詳細については、次の資料を参照してください。

『パラメータマニュアル、システム変数』；「システム変数リスト」：フレームの章

『機能マニュアル、基本機能』；「軸、座標系、フレーム (K2)」  
：フレーム結合のフレームの章

### 関連参照先:

『プログラミングマニュアル、上級編』；「テクノロジーサイクル」  
：「旋回-CYCLE800」の章

## 9.5.2 ワーク測定

### ワーク計測

ワーク計測の場合は、プローブが工具と同様に、クランプされたワークまで移動します。使用可能なそれぞれの計測タイプのために、一般的な計測作業のほとんどは、ターニングセンタまたはフライス盤で簡単かつ容易におこなうことができます。

エッジ、コーナー、または穴に関連するワークの位置を計測できます。

ワークの原点位置(ワーク原点 **W**)または穴を決定するために、**WCS** で計測された位置に指令位置を追加できます。その結果のオフセットを、選択したフレームに入力できます。

### インタフェース変数

インタフェース変数は複数のシステム変数から構成されます。

これらの変数は、次のいずれかに分類されます。

- 入力値
- 出力値

#### 参照先:

『パラメータマニュアル、システム変数』

入力値は、HMI またはサイクルで書き込んでください。計算結果から出力値が得られません。

#### 参照先:

『プログラミングマニュアル、計測サイクル』

### 9.5.2.1 入力値

#### 計測タイプの有効ビット

現在の計測に有効なシステム変数を定義するには、最初に各計測処理で、すべての変数を無効として宣言してください。これは、**\$AC\_MEAS\_VALID = 0** で実行されます。

## 9.5 原点の設定、ワーク測定、および工具測定

各入力変数は、インタフェースへの書き込み時に、対応する\$AC\_MEAS\_VALIDのビットを自動的に設定します。有効ビットがリセットされていない場合、値は次の計算でも、そのまま有効です。

## 注記

インタフェースは、機械操作盤リセットのとき、またはM30(プログラム終了時のリセット)の後は、リセットされません。

表 9-7 変数\$AC\_MEAS\_VALIDの入力値の有効ビット

ビット	入力値	意味
0	\$AA_MEAS_POINT1[軸]	すべてのチャンネル軸の1番目の計測点
1	\$AA_MEAS_POINT2[軸]	すべてのチャンネル軸の2番目の計測点
2	\$AA_MEAS_POINT3[軸]	すべてのチャンネル軸の3番目の計測点
3	\$AA_MEAS_POINT4[軸]	すべてのチャンネル軸の4番目の計測点
4	\$AA_MEAS_SETPOINT[軸]	エッジ、コーナー、穴の指令位置
5	\$AC_MEAS_WP_SETANGLE	ワーク位置指令角度 $\alpha$ ; $-90 < \alpha < 180$
6	\$AC_MEAS_CORNER_SETANGLE	コーナーの交点 $\phi$ の指令角度 $0 < \phi < 180$
7	\$AC_MEAS_T_NUMBER	選択工具
8	\$AC_MEAS_D_NUMBER	選択刃先
9	\$AC_MEAS_DIR_APPROCH	アプローチ方向はエッジ、溝、ウェブ、および工具計測のみ
10	\$AC_MEAS_ACT_PLANE	作業面と切り込み方向の設定
11	\$AC_MEAS_FRAME_SELECT	指定フレームの計算フレーム
12	\$AC_MEAS_TYPE	ワーク計測タイプ
13	\$AC_MEAS_FINE_TRANS	平行移動オフセットを入力
14	\$AA_MEAS_SETANGEL[軸]	軸の指令角度
15	\$AA_MEAS_SCALEUNIT	入力値と出力値の単位系
16	\$AA_MEAS_TOOL_MASK	工具設定内容
17	\$AA_MEAS_P1_COORD	1番目の計測点の座標系
18	\$AA_MEAS_P2_COORD	2番目の計測点の座標系
19	\$AA_MEAS_P3_COORD	3番目の計測点の座標系
20	\$AA_MEAS_P4_COORD	4番目の計測点の座標系

ビット	入力値	意味
21	\$AA_MEAS_SET_COORD	指令値の座標系
22	\$AA_MEAS_CHSFR	システムフレームマスク
23	\$AA_MEAS_NCBFR	グローバル基本フレームのマスク
24	\$AA_MEAS_CHBFR	チャンネル基本フレームのマスク
25	\$AA_MEAS_UIFR	データ管理の設定可能フレーム
26	\$AA_MEAS_PFRAME	プログラマブルフレームの計算禁止
27	\$AC_MEAS_INPUT[n]	長さ n の配列の計測入力パラメータ

#### 注記

当該の計測点のすべての軸の現在位置は、次の変数で無効になります。

\$AC\_MEAS\_LATCH = 0

## 計測点

すべての計測用チャンネル軸では、4 個までの計測点を使用できます。

タイプ	入力変数	意味
REAL	\$AA_MEAS_POINT1[軸]	すべてのチャンネル軸の 1 番目の計測点
REAL	\$AA_MEAS_POINT2[軸]	すべてのチャンネル軸の 2 番目の計測点
REAL	\$AA_MEAS_POINT3[軸]	すべてのチャンネル軸の 3 番目の計測点
REAL	\$AA_MEAS_POINT4[軸]	すべてのチャンネル軸の 4 番目の計測点

計測された位置は通常、WCS の現在位置(= 指令値)として使用できます。計測点は、軸位置データ値が記述されると同時に有効となります。個々の計測点はそれぞれ、書き込みまたは読み取りが可能です。

## 9.5 原点の設定、ワーク測定、および工具測定

2、3の計測タイプは、異なる座標系(BCS、MCS)にある計測点もサポートしています。対応する計測点の座標系計測の入力は、次の変数によって実行できます。

タイプ	入力変数	意味	値
INT	\$AA_MEAS_P1_COORD	1番目の計測点の座標系	0:WCS が標準設定です。 1:BCS 2:MCS 3:ENS 4:WCS_REL 5:ENS_REL
INT	\$AA_MEAS_P2_COORD	2番目の計測点の座標系	
INT	\$AA_MEAS_P3_COORD	3番目の計測点の座標系	
INT	\$AA_MEAS_P4_COORD	4番目の計測点の座標系	
INT	\$AA_MEAS_SET_COOR D	指令値の座標系	

## 現在位置

計測点は、現在の軸の現在位置を持つすべての軸に対して記述できます。位置は、選択した座標系に関して取得されます。座標系が指定されていない場合は、WCSの位置になります。このために、次の変数

が使用されます。**\$AC\_MEAS\_LATCH[0..3]**

インデックスは、1番目から4番目までの計測点に対応して0から3のいずれかとなります。変数に値0を割り当てると、対応する計測点のすべての軸の実際値が無効となります。値1を割り当てると、対応する計測点のすべての軸の実際値が取得されます。変数は書き込み専用の変数です。

計測点の個々の軸の現在位置は、次の変数で記述できます。

タイプ	システム変数	意味	値
REAL	\$AA_MEAS_P1_VALID[ax ]	1番目の軸の計測点を取得	0:軸の計測点が無効 1:軸の計測点を決定
REAL	\$AA_MEAS_P2_VALID[ax ]	2番目の軸の計測点を取得	
REAL	\$AA_MEAS_P3_VALID[ax ]	3番目の軸の計測点を取得	
REAL	\$AA_MEAS_P4_VALID[ax ]	4番目の軸の計測点を取得	

変数**\$AC\_MEAS\_LATCH[0..3]**と**\$AA\_MEAS\_P[1..4]\_VALID**は、相互に作用しながら使用できます。許容値は直径/半径指定の直径指定に従います。

## 指令値

結果フレームは、ユーザで指定された指令値に計測値が適合するように計算されます。

表 9-8 ユーザ指令値の入力値

タイプ	システム変数	意味
REAL	\$AA_MEAS_SETPOINT[ax]	軸の指令位置
REAL	\$AA_MEAS_SETANGLE[ax]	軸の指令角度
INT	\$AA_MEAS_SP_VALID[ax]	1:軸の指令位置が有効/ 0:無効
REAL	\$AC_MEAS_WP_SETANGLE	ワーク位置指令角度 $\alpha$ $-90 < \alpha < 180$
REAL	\$AC_MEAS_CORNER_SETANGLE	コーナーの指令切削角度 $\varphi$ $0 < \varphi < 180$
INT	\$AC_MEAS_DIR_APPROACH *)	アプローチの方向: 0: +x、1: -x、2: +y、3: -y、4: +z、5: -z

\*) アプローチ方向はエッジ、溝、ウェブ、および工具計測にのみ必要です。

次の計測点は無効のため、参照されません。

- ワーク位置指令角度  $\alpha$  の入力時の 2 番目の計測点。
- 交点の指令角度  $\varphi$  の入力時の 4 番目の計測点。

## 平面選択

工具の向きを定義するための平面選択。平面が指定されていない場合は、すべての計算に対して動作中の平面が使用されます。

タイプ	システム変数	値
INT	\$AC_MEAS_ACT_PLANE	0:G17 作業平面 x/y 切り込み方向 z 1:G18 作業平面 z/x 切り込み方向 y 2:G19 作業平面 y/z 切り込み方向 x

### 平行移動オフセット

ワーク計測時に、選択したフレームの精密オフセット成分に平行移動オフセットを入力できます。このためには、変数\$SAC\_MEAS\_FINE\_TRANS が使用されます。

タイプ	システム変数	値
INT	\$SAC_MEAS_FINE_TRANS	0:平行移動補正が汎用オフセットに入力されます。 1:平行移動補正が精密オフセットに入力されます。

変数\$SAC\_MEAS\_FINE\_TRANS が記述されていない場合は、下記が適用されます。

- 補正值は、汎用オフセットに入力され、目的のフレームに変換されます。座標変換によって、平行移動補正が精密オフセットである場合もあります。
- 次のマシンデータが事前に 1 に設定されていない場合:  
MD18600 \$MN\_MM\_FRAME\_FINE\_TRANS  
補正は常に、汎用オフセットに入力されます。

### 計算フレーム

ワークが計測されると、計算フレームが指定フレームに入力されます。

表 9-9

タイプ	システム変数	意味
INT	\$SAC_MEAS_FRAME_SELECT	工具計測中のフレーム選択

変数\$SAC\_MEAS\_FRAME\_SELECT は、次の値を受け入れることができます。

規格値		意味
0	\$P_SETFRAME	動作中のシステムフレーム
1	\$P_PARTFRAME	動作中のシステムフレーム
2	\$P_EXTFRAME	動作中のシステムフレーム
10 ... 25	\$P_CHBFRAME[0..15]	動作中のチャンネル別基本フレーム
50 ... 65	\$P_NCBFRAME[0..15]	動作中の NCU グローバル基本フレーム

## 9.5 原点の設定、ワーク測定、および工具測定

規格値		意味
100 ... 199	\$P_IFRAME	対応するフレームが選択されている場合、動作中の設定可能フレームを使用して計算がおこなわれます。選択されたフレームが有効でない場合は、対応するデータ管理フレームが計算に含まれます。
500	\$P_TOOLFRAME	動作中のシステムフレーム
501	\$P_WPFRAME	動作中のシステムフレーム
502	\$P_TRAFRAME	動作中のシステムフレーム
503	\$P_PFRAME	動作中の、現在のプログラマブルフレーム
504	\$P_CYCFRAME	動作中のシステムフレーム
505	\$P_RELFRAME (ワーク座標系)	動作中のシステムフレーム
506	\$P_RELFRAME (SZS)	動作中のシステムフレーム
1010 ... 1025	\$P_CHBFRAME[0..15]	G500 が有効な、動作中のチャンネル指定基本フレーム
1050 ... 1065	\$P_NCBFRAME[0..15]	G500 が有効な、動作中の NCU グローバル基本フレーム
2000	\$P_SETFR	データ管理のシステムフレーム
2001	\$P_PARTFR	データ管理のシステムフレーム
2002	\$P_EXTFR	データ管理のシステムフレーム
2010 ... 2025	\$P_CHBFR[0..15]	データ管理のチャンネル別基本フレーム
2050 ... 2065	\$P_NCBFR[0..15]	データ管理の NCU グローバル基本フレーム
2100 ... 2199	\$P_UIFR[0..99]	データ管理の設定可能フレーム
2500	\$P_TOOLFR	データ管理のシステムフレーム
2501	\$P_WPFR	データ管理のシステムフレーム
2502	\$P_TRAFR	データ管理のシステムフレーム
2504	\$P_CYCFR	データ管理のシステムフレーム
2505	\$P_RELFR (ワーク座標系)	データ管理のシステムフレーム
2506	\$P_RELFR (SZS)	データ管理のシステムフレーム

## 9.5 原点の設定、ワーク測定、および工具測定

規格値		意味
3010 ... 3025	\$P_CHBFR[0..15]	チャンネル別 G500 が有効な、データ管理の基本フレーム。
3050 ... 3065	\$P_NCBFR[0..15]	G500 が有効な、データ管理の NCU グローバル基本フレーム。

MEASURE()機能は、指定されたフレームに従って、フレーム\$AC\_MEAS\_FRAME を計算します。

値が

**0 ~ 1065** の場合は、動作中のフレームを使用して、計算がおこなわれます。

**2000 ~ 3065** の場合は、データ管理で選択されたフレームで、計算がおこなわれます。データ管理のフレームの選択は、計測タイプ **14** と **15** ではサポートされていません。フレームが有効でない場合でも、データ管理で選択できます。この場合、計算は、フレーム結合が有効であるかのようにおこなわれます。

計測点は、選択された系で変換され、選択フレームは、それ自体を含むフレーム全体を使用して決定されます。現在位置設定は、補正とフレームの適用後にのみ有効になります。

値が

**G500** および (1010..1025、1050..1065、3010..3025、3050..3065)が有効な場合は、指令位置が計算できるようにフレームが選択された後に **G500** が有効になり、その結果、対象フレームが計算されます。

## 別の座標系への変換

位置が、別の座標系の位置に変換される場合は、次の変数を使用して、目的のフレーム結合の構成を指定できます。

タイプ	システム変数	意味	値
INT	\$AC_MEAS_CHSFR	システムフレームの選択	MD28082 \$MC_MM_SYSTEM_FRAME_MASK に対応するビットマスク
INT	\$AC_MEAS_NCBFR	グローバル基本フレームの選択	ビットマスク(0 ... FFFF)
INT	\$AC_MEAS_CHBFR	チャンネル基本フレームの選択	ビットマスク(0 ... FFFF)

タイプ	システム変数	意味	値
INT	\$AC_MEAS_UIFR	設定可能フレームの選択	0 ... 99
INT	\$AC_MEAS_PFRAM E	プログラマブルフレーム	0:含まれる 1:含まれない

データ管理フレームが読み込まれ、新しいフレームが、変数の対応する値に対して設定されます。

#### 注記

変数が設定されていない場合は、動作中のフレームが保持されます。

値は、書き込む変数のデータ管理フレームが、新しいフレーム結合に含まれる場合にのみ、書き込んでください。基本フレームの場合は、**すべての**フレームが入れ替えられます。特定のフレームのみを入れ替えることはできません。**\$P\_NCBFRMASK** と **\$P\_CHBFRMASK** による動作中の変更は無視されます。

#### ワーク計測と工具計測の配列変数

次の長さ  $n$  の配列変数は、様々な計測タイプで使用される詳細な入力パラメータに使用されます。

タイプ	システム変数	意味	値
REAL	\$AC_MEAS_INPUT[n]	計測入力パラメータ	$n = 0 \dots 9$

計測入力パラメータの制御動作が、計測方法と共に記述されます。

#### 工具または刃先の選択

工具および動作中の工具の刃先番号は、選択工具に対応する必要があります。T0 のときは、D0 が選択され、動作中の工具が計算されます。動作中の工具がない場合は、T、D で選択された工具が計算されます。選択工具以外は、有効にすることはできません。

タイプ	システム変数	意味
INT	\$AC_MEAS_T_NUMBER	選択工具
INT	\$AC_MEAS_D_NUMBER	選択刃先

### 3次元プローブによる計測

3次元プローブで計測する場合は、工具半径がすでに計測点に関して補正されているため、様々な計測操作の計算時に半径を含める必要はありません。この機能は、次の変数を使用して定義できます。

タイプ	システム変数	意味
INT	\$AC_MEAS_TOOL_MASK	工具位置

変数\$AC\_MEAS\_TOOL\_SCREEN は、次の値を取ることができます。

規格値	意味
0x0	すべての工具長が考慮されます(初期設定)。
0x1	工具半径は計算に含まれません。
0x2	x 方向の工具位置(G19)。
0x4	y 方向の工具位置(G18)。
0x8	y 方向の工具位置(G17)。
0x10	工具長は計算に含まれません。
0x20	動作中の工具長が位置の座標変換に含まれます。
0x40	x 方向の工具位置。
0x80	y 方向の工具位置。
0x100	z 方向の工具位置。
0x200	工具長の差分値が負の値で含まれます。

フライス工具の半径が計算に含まれるかどうかは、工具位置とアプローチ方向で決定できます。アプローチ方向が明確に指定されていない場合は、選択された平面で決定されます。

平面	アプローチ方向
G17	-z 方向
G18	-y 方向:
G19	-x 方向

## 9.5.2.2 測定の選択

測定は、次の変数を使用して選択されます。

タイプ	システム変数	説明
INT	\$AC_MEAS_TYPE	測定タイプの選択

変数\$AC\_MEAS\_TYPE は、次の値を取ることができます。

値		説明
0		初期設定
1	Edge_x	x エッジの測定
2	Edge_y	y エッジの測定
3	Edge_z	z エッジの測定
4	Corner_1	コーナ 1 の測定
5	Corner_2	コーナ 2 の測定
6	Corner_3	コーナ 3 の測定
7	Corner_4	コーナ 4 の測定
8	穴	穴測定
9	Stud	シャフトの測定
10 *	工具長	工具長の測定
11 *	ToolDiameter	工具直径の測定
12	スロット	溝の測定
13	Plate	ウェブの測定
14	Set_Pos	ジオメトリ軸および付加軸の現在位置設定
15	Set_AuxPos	付加軸のみの現在位置設定
16	Edge_2P	傾斜エッジの測定
17	Plane_Angles	平面の角度
18	Plane_Normal	指令値入力 of 平面の角度
19	Dimension_1	1 次元の指令値
20	Dimension_2	2 次元の指令値
21	Dimension_3	3 次元の指令値
22 *	ToolMagnifier	ShopTurn: ズームイン機能による工具長の測定
23 *	ToolMarkedPos	ShopTurn: マーキング位置による工具長の測定
24	座標変換	位置の座標変換

## 9.5 原点の設定、ワーク測定、および工具測定

値		説明
25	長方形	長方形の測定
26	Save	データ管理フレームの保存
27	復元	データ管理フレームの復元
28	テーパ旋削	平面の付加回転

\* ワーク測定タイプ

個々の方法は、「ワーク測定タイプ」または「工具測定タイプ」にあるリストに記載されています、そして当該のプログラミング例を使用して詳しく説明されています。

## 9.5.2.3 出力値

## 計算結果

指令位置が指定されている場合は、結果のフレームがフレーム結果\$AC\_MEAS\_FRAMEに入力されます。このフレームは、パートプログラムで読み込みと書き込みが可能です。フレーム結果は、選択されたフレームに従って計算されます。

フレームが選択されていない場合は、フレーム結果により、WCSの最終の平行移動と回転が決定されます。このフレームは、PI サービス\_N\_SETUDT およびパラメータタイプ7番を使用して、選択されたフレームに入力できます。一度入力が終わると、フレーム結果は削除されます。

表 9-10 計算結果の出力値

タイプ	システム変数	説明
FRAME	\$AC_MEAS_FRAME	フレーム結果
REAL	\$AC_MEAS_WP_ANGLE	ワーク位置計算角度 $\alpha$
REAL	\$AC_MEAS_CORNER_ANGLE E	計算交点角度 $\phi$
REAL	\$AC_MEAS_DIAMETER	計算直径
REAL	\$AC_MEAS_TOOL_LENGTH	計算工具長
REAL	\$AC_MEAS_RESULTS[10]	計算結果(\$AC_MEAS_TYPEにより異なります)

### 9.5.2.4 計算方法

#### 計算の有効化

計算は、HMI オペレータ操作で PI サービス\_N\_SETUDT で有効になります。この PI サービスには、次のパラメータタイプのいずれかを指定できます。

タイプ	意味
1	有効な工具オフセット
2	有効な基本フレーム
3	有効な設定可能フレーム
4	グローバル基本フレーム
5	グローバルな設定可能フレーム
6	ワーク原点または工具長の計算
7	ワーク原点の有効化(接触計測の書き込み)
8	外部ワークオフセットの有効化
9	動作中の工具キャリヤ、TCOABS、および PAROT の有効化

変更は、リセットステータスで直ちに反映されます。停止ステータスでは、フレームは次の起動時に後退します。

#### 注記

PI サービスは、リセットおよび停止ステータスでのみ実行できます。ワーク測定の場合は、計算されたフレームが、タイプ 7 番で直ちに有効になります。工具測定では、原点を有効にする必要がないため、タイプ 7 番で PI サービスを送信しないでください。

#### 停止ステータスでの適用

新しい WCS 位置は、停止ステータスで更新されます。パートプログラムの継続起動時に、中断されたブロックの残移動距離が削除されます。現在の位置から次ブロックの終点への移動が行われます。

したがって停止ステータスにおいて、MDAP モードまたはパートプログラムで主軸を起動した後、M0 で現在値を設定および接触計測することもできます。別の計測を行うこともできます。

## 計測サイクル

計測サイクルでの計算は、次の予約機能で行われます。

## INT MEASURE()

MEASURE()によりフレーム結果が提供され、それを\$AC\_MEAS\_FRAMEによって読むことができます。

- この結果は、選択されたフレームで再計算された指令値からの平行移動と回転です。
- このフレーム結果は、次のように計算されます。  
結合後の合計フレームは、計算された平行移動と回転による (測定前の)合計フレームと同じです。

## 通知

## 事前処理の制限はありません

MEASURE()により、自動的にブロック先読み停止が起動されることはありません。MEASURE()は事前処理ブロックのフレームを処理するので、ユーザーは演算前に事前処理の停止が必要であるかどうかを決定してください。

## 注記

フレームが選択されていない場合は、結合されたフレームは変換されません。つまり、平行移動と回転は、指定された指令値および計算されたエッジ、コーナー、溝などの位置に基づいて決定されます。機能が複数回使用されると、その結果が常にフレーム結果に追加されます。

結果フレームを事前に削除することが必要になる場合があります。

## セマフォ変数

測定変数は、チャンネル毎に 1 回だけ使用されます。測定操作は、停止およびリセットのステータスでオペレータの入力によって開始できます。この操作は、停止ステータスの計測サイクルと重複する可能性があります。次の変数は、相互上書きに対する保護のために役立ちます:

## \$AC\_MEAS\_SEMA (測定インタフェースのセマフォ)

セマフォ変数\$AC\_MEAS\_SEMA は、次のように機能します。

- サイクル開始時に 1 に設定
- サイクル終了時に 0 に戻してリセット

変数の値が 1 の場合、HMI は測定インタフェースを使用しません。

## エラーメッセージ

クライアントがログオンしない場合は、常にグループエラー番号 0xD003 が発生します。DIAGN:errCodeSetNrGent または DIAGN:errCodeSetNrPi によってログオンした場合は、PI\_SETUDT によって、次の構文に対応するエラーコードが生成されます。

EX\_ERR\_PI\_REJ\_<戻り値>、例: EX\_ERR\_PI\_REJ\_MEASNOTYPE

次の戻り値は、予約 MEASURE()機能によって出力されます。

表 9-11 予約エラーメッセージ

番号	戻り値	意味
0	MEAS_OK	正常計算
1	MEAS_NO_TYPE	タイプが指定されていません
2	MEAS_TOOL_ERROR	工具決定エラー
3	MEAS_NO_POINT1	測定点 1 が存在しません
4	MEAS_NO_POINT2	測定点 2 が存在しません
5	MEAS_NO_POINT3	測定点 3 が存在しません
6	MEAS_NO_POINT4	測定点 4 が存在しません
7	MEAS_NO_SPECPOINT	レファレンス点がありません
8	MEAS_NO_DIR	アプローチ方向がありません
9	MEAS_EQUAL_POINTS	測定点が同じです
10	MEAS_WRONG_ALPHA	アルファ $\alpha$ が誤っています
11	MEAS_WRONG_PHI	ファイ $\phi$ が誤っています
12	MEAS_WRONG_DIR	アプローチ方向が誤っています
13	MEAS_NO_CROSSING	線が交差しません
14	MEAS_NO_PLANE	平面が存在しません
15	MEAS_WRONG_FRAME	選択されたフレームが無い、または誤っています
16	MEAS_NO_MEMORY	使用可能なメモリが不足しています
17	MEAS_INTERNAL_ERROR	内部エラー

## 工具測定エラー

エラーコード MEAS\_TOOL\_ERROR または EX\_ERR\_PI\_REJ\_MEASTOOLERROR の場合は、システムにより、出力変数 \$AC\_MEAS\_TOOL\_LENGTH の下記の値で、詳細なエラー内容が保存されます。

表 9-12 MEAS\_TOOL\_ERROR の予約エラーメッセージ

番号	戻り値	意味
1	TOOL_NO_BLOCK	工具計算に使用できるブロックがありません
2	TOOL_WRONG_T_NUMBER	T 番号が誤っています
3	TOOL_WRONG_D_NUMBER	D 番号が誤っています
4	TOOL_EVAL_WRONG_TYPE	工具がありません
5	TOOL_NO_TOOLCORR_BODY	メモリに問題があります
6	TOOL_DATA_READ_ERROR	工具データの読み込みエラーです
7	TOOL_NO_TOOL_WITH_TRAFO	動作中の変換で選択された工具がありません

## 9.5.2.5 計測および計算の計測変数の単位系

## INCH または METRIC の単位系

次の入力変数と出力変数は、インチまたはメトリックの単位系で使用されます。

\$AA_MEAS_POINT1[軸]	1 番目の計測点の入力変数
\$AA_MEAS_POINT2[軸]	2 番目の計測点の入力変数
\$AA_MEAS_POINT3[軸]	3 番目の計測点の入力変数
\$AA_MEAS_POINT4[軸]	4 番目の計測点の入力変数
\$AA_MEAS_SETPPOINT[軸]	指令位置の入力変数
\$AC_MEAS_DIAMETER	計算直径の出力変数
\$AC_MEAS_TOOL_LENGTH	計算工具長の出力変数
\$AC_MEAS_RESULTS[n]	計算結果の出力変数

入力値と出力値の読み書きに使用される単位系は、入力変数によって設定できます。

<b>INT \$AC_MEAS_SCALEUNIT</b>	入力変数と出力変数の単位系
<b>0:</b> 下記に関連した単位系	<b>INCH</b> 単位の動作中の G 命令 <b>G70/G700</b> <b>METRIC</b> 単位の動作中の G 命令 <b>G71/G701</b>
<b>1:</b> 次のものに対応する単位系	設定; 単位系は <b>OPI</b> (標準設定) によって設定可能

変数が書き込まれない場合は、常に値 **1** が標準設定として扱われます。

#### 例:

基本単位系はメトリックです。

プログラムコード	コメント
G70	
\$AC_MEAS_POINT1[x] = \$AA_IW[x]	; \$AA_IW[x] が基本単位系を提供
\$AC_MEAS_POINT1[x] = 10	; 10 mm
G71	
\$AC_MEAS_POINT1[x] = \$AA_IW[x]	; \$AA_IW[x] が基本単位系を提供
\$AC_MEAS_POINT1[x] = 10	; 10 mm
G700	
\$AC_MEAS_POINT1[x] = \$AA_IW[x]	; \$AA_IW[x] がインチ値を提供
\$AC_MEAS_POINT1[x] = 10	; 10 インチ
G710	
\$AC_MEAS_POINT1[x] = \$AA_IW[x]	; \$AA_IW[x] がメトリック値を提供
\$AC_MEAS_POINT1[x] = 10	; 10 mm

## 直径指定

直径指定は、次のマシンデータによって設定されます。

```
MD20100 $MC_DIAMETER_AX_DEF = 「X」 ;径方向軸 x
MD20150 $MC_GCODE_RESET_VALUES[28] = ; DIAMON
2
MD20360 ; 工具長、フレーム、および
$MC_TOOL_PARAMETER_DEF_MASK = ; 直径の現在位置
'B1001010'
```

## 9.5 原点の設定、ワーク測定、および工具測定

次の事柄を考慮してください。

- MCS の軸位置は、直径値にはなりません。
- 計算工具長とフレーム成分は、動作中の G 命令 DIAMON と DIAMOF によって変わることはありません。
- 計測点と指令位置は、DIAMON に応じて、読み込まれ、書き込まれます。
- フレームでの平行移動は、径方向軸の直径として計算されます。

### 演算と表示精度

mm、インチ、または度単位の位置データは、小数点 6 位まで正確に計算され、表示されます。

#### 9.5.2.6 診断

測定インタフェースには、次の診断オプションがあります。

- `/_N_MPF_DIR/_N_MEAS_DUMP_MPF` ファイルが使用可能な場合は、このファイルにログが書き込まれ、それにより問題の再現が可能となります。
- ログの書き込みは、`_N_MEAS_DUMP_MPF` というファイル名の空のファイルを `/_N_MPF_DIR` ディレクトリに作成すると、開始されます。
- ファイルの内容は、`$AC_MEAS_VALID = 0` で削除されるまで保存されます。

運転時間に影響しないように、トレースは、問題が検出された場合にのみ有効にしてください。

### 9.5.3 ワーク測定タイプ

#### 9.5.3.1 エッジの計測(計測タイプ 1、2、3)

##### x エッジの計測(\$AC\_MEAS\_TYPE = 1)

クランプされたワークのエッジは、既知の工具でこのエッジにアプローチして計測します。

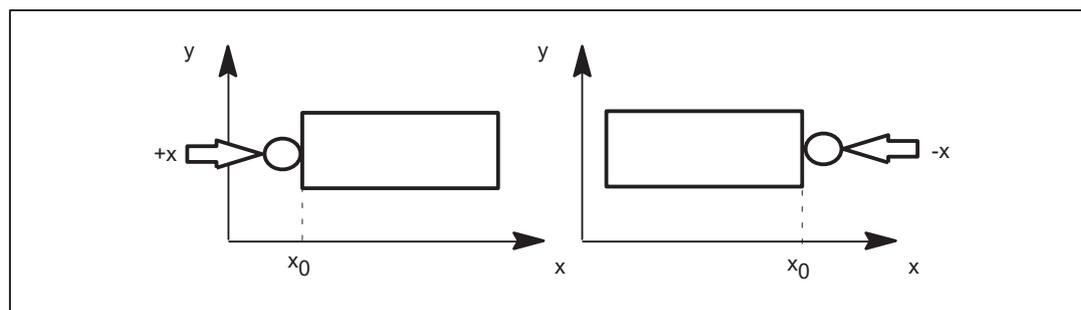


図 9-1 x エッジ

次の変数の値が計測タイプ 1 のために使用されます。

入力変数	意味
\$AC_MEAS_VALID	入力変数の有効ビット
\$AA_MEAS_POINT1[軸]	すべてのチャンネル軸の計測点 1
\$AA_MEAS_SETPOINT[軸]	x エッジの指令位置*
\$AC_MEAS_DIR_APPROACH	0: +x、1: -x
\$AC_MEAS_ACT_PLANE	指定されていない場合は動作中の平面で計算がおこなわれ、 工具半径は G17 と G18 のみで使用*
\$AC_MEAS_FINE_TRANS	0: 汎用オフセット、1: 仕上げオフセット
\$AC_MEAS_FRAME_SELECT	特に指定されていない限り、追加フレームに従って計算*
\$AC_MEAS_T_NUMBER	特に指定されていない限り、動作中の T に従って計算(T0) *

9.5 原点の設定、ワーク測定、および工具測定

入力変数	意味
\$AC_MEAS_D_NUMBER	特に指定されていない限り、動作中の D に従って計算(D0) *
\$AC_MEAS_TYPE	1

\* オプションです

次の出力変数は、計測タイプ 1 のために書き込まれます。

出力変数	意味
\$AC_MEAS_FRAME	平行移動のあるフレーム結果
\$AC_MEAS_RESULTS[0]	計測されたエッジの位置

例

x エッジの計測値

プログラムコード	コメント
DEF INT RETVAL	
DEF FRAME TMP	
\$TC_DP1[1,1]=120	; タイプ
\$TC_DP2[1,1]=20	; 0
\$TC_DP3[1,1]=10	; (z) 長補正ベクトル
\$TC_DP4[1,1]=0	; (y)
\$TC_DP5[1,1]=0	; (x)
\$TC_DP6[1,1]=2	; 半径
T1 D1	
g0 x0 y0 z0 f10000	
G54	
	; x エッジを計測
\$AC_MEAS_VALID = 0	; すべての入力値を無効に設定
g1 x-1 y-3	; 1. 番目の計測点へアプローチ
\$AA_MEAS_POINT1[x] = \$AA_IW[x]	
\$AA_MEAS_POINT1[y] = \$AA_IW[y]	
\$AA_MEAS_POINT1[z] = \$AA_IW[z]	
\$AC_MEAS_DIR_APPROACH = 0	; アプローチ方向+x を設定

プログラムコード	コメント
\$AA_MEAS_SETPOINT[x] = 0	; エッジの指令位置を設定
\$AA_MEAS_SETPOINT[y] = 0	
\$AA_MEAS_SETPOINT[z] = 0	
\$AC_MEAS_ACT_PLANE = 0	; 計測面は G17
\$AC_MEAS_FRAME_SELECT = 101	; フレーム- IFRAME を選択
\$AC_MEAS_T_NUMBER = 1	; 工具選択
\$AC_MEAS_D_NUMBER = 1	
\$AC_MEAS_TYPE = 1	; x エッジの計測タイプを設定
RETVL = MEASURE()	; 計測処理を開始
if RETVAL <> 0	
setal(61000 + RETVAL)	
endif	
\$P_IFRAME = \$AC_MEAS_FRAME	
\$P_UIFR[1] = \$P_IFRAME	; データ管理のシステムフレームを記述
g1 x0 y0	; エッジヘアプローチ
m30	

### y エッジの計測(\$AC\_MEAS\_TYPE = 2)

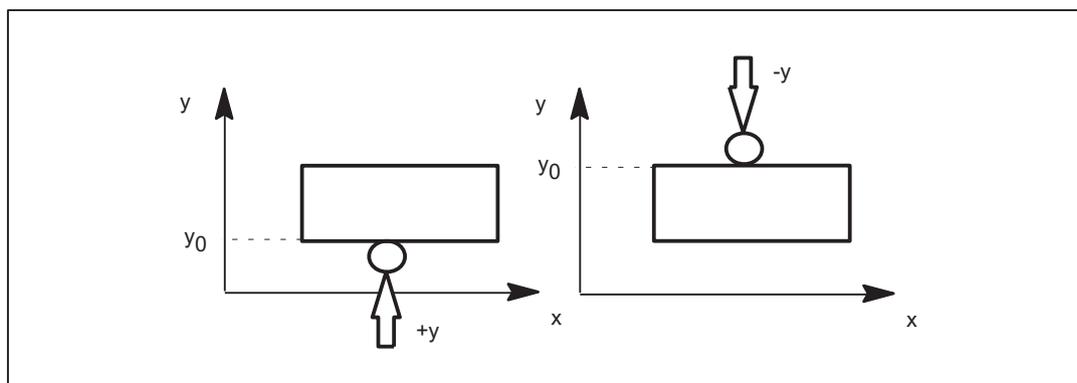


図 9-2 y エッジ

## 9.5 原点の設定、ワーク測定、および工具測定

次の変数値が計測タイプ 2 のために使用されます。

入力変数	意味
\$AC_MEAS_VALID	入力変数の有効ビット
\$AA_MEAS_POINT1[軸]	すべてのチャンネル軸の計測点 1
\$AA_MEAS_SETPOINT[軸]	y エッジの指令位置*
\$AC_MEAS_DIR_APPROACH	2: +y、3: -y
\$AC_MEAS_ACT_PLANE	指定されていない場合は動作中の平面で計算がおこなわれ、 工具半径は G17 と G19 のみで使用*
\$AC_MEAS_FINE_TRANS	0: 汎用オフセット、1: 精密オフセット*
\$AC_MEAS_FRAME_SELECT	特に指定されていない限り、追加フレームに従って計算*
\$AC_MEAS_T_NUMBER	特に指定されていない限り、動作中の T に従って計算(T0) *
\$AC_MEAS_D_NUMBER	特に指定されていない限り、動作中の D に従って計算(D0) *
\$AC_MEAS_TYPE	2

\* オプションです

次の出力変数は、計測タイプ 2 のために書き込まれます。

出力変数	意味
\$AC_MEAS_FRAME	平行移動のあるフレーム結果
\$AC_MEAS_RESULTS[0]	計測されたエッジの位置

## z エッジの計測(\$AC\_MEAS\_TYPE = 3)

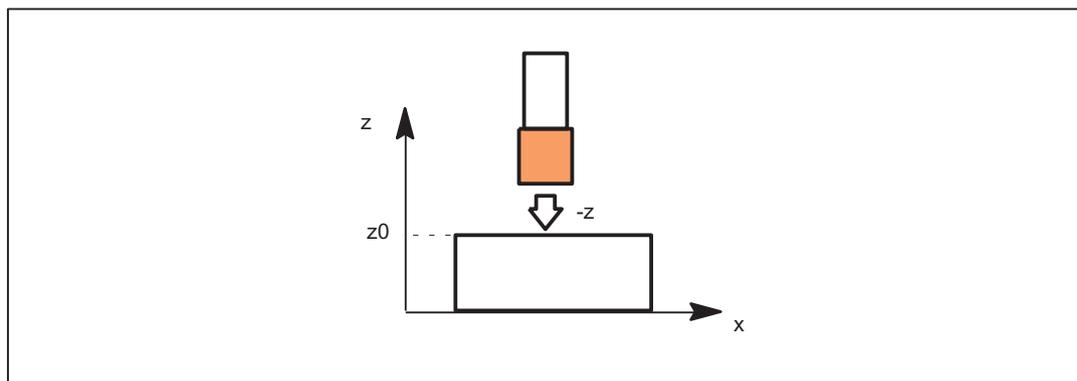


図 9-3 z エッジ

次の変数値は計測タイプ 3 のために使用されます。

入力変数	意味
\$AC_MEAS_VALID	入力変数の有効ビット
\$AA_MEAS_POINT1[軸]	すべてのチャネル軸の計測点 1
\$AA_MEAS_SETPOINT[軸]	z エッジの指令位置*
\$AC_MEAS_DIR_APPROACH	4: +z、5: -z
\$AC_MEAS_ACT_PLANE	指定されていない場合は動作中の平面で計算がおこなわれ、 工具半径は G18 と G19 のみで使用*
\$AC_MEAS_FINE_TRANS	0: 汎用オフセット、1: 精密オフセット*
\$AC_MEAS_FRAME_SELECT	特に指定されていない限り、追加フレームに従って計算*
\$AC_MEAS_T_NUMBER	特に指定されていない限り、動作中の T に従って計算(T0) *
\$AC_MEAS_D_NUMBER	特に指定されていない限り、動作中の D に従って計算(D0) *
\$AC_MEAS_TYPE	3

\* オプションです

## 9.5 原点の設定、ワーク測定、および工具測定

次の出力変数は、計測タイプ 3 のために書き込まれます。

出力変数	意味
\$AC_MEAS_FRAME	平行移動のあるフレーム結果
\$AC_MEAS_RESULTS[0]	計測されたエッジの位置

## 9.5.3.2 角度計測(計測タイプ 4、5、6、7)

## コーナーの計測 C1 - C4 (\$AC\_MEAS\_TYPE = 4, 5, 6, 7)

コーナーは、P1 ~ P4 という 4 つの計測点へのアプローチによって、固有に定義されます。交点  $\phi$  の既知の角度に対しては 3 つの計測点で十分です。

交点の角度  $\phi$  とワーク位置角度  $\alpha$  がすでにわかっている場合は、2 つの計測点 P1 と P3 でも十分です。

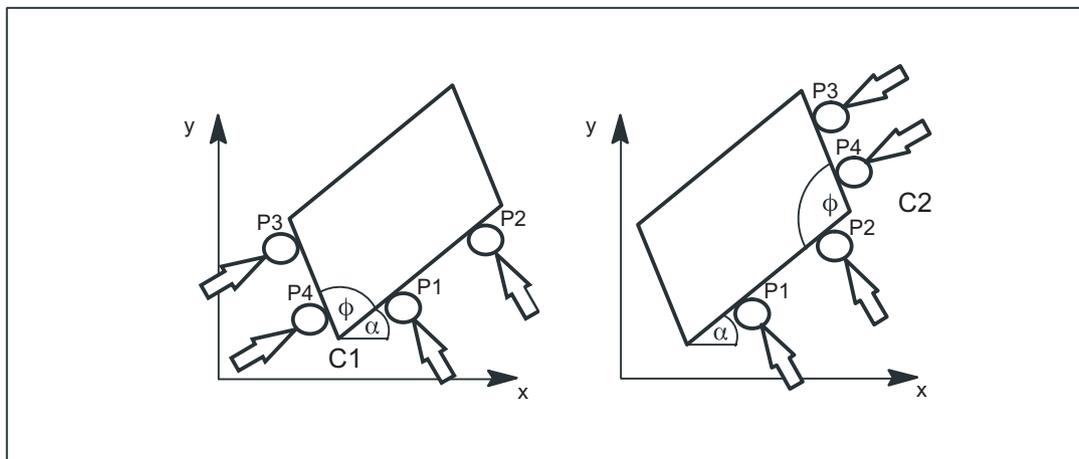


図 9-4 コーナー C1、コーナー C2

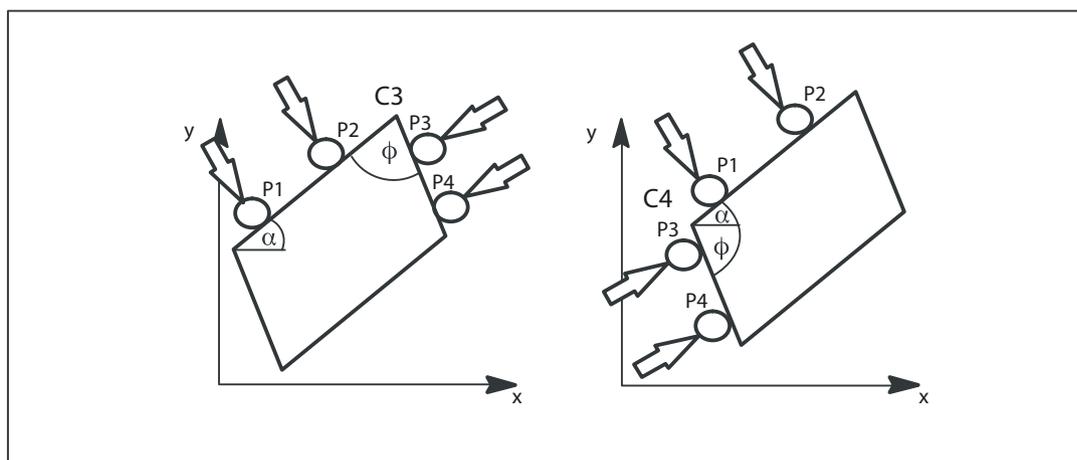


図 9-5 コーナー C3、コーナー C4

次の変数の値が計測タイプ 4～7 のために使用されます。

入力変数	意味
\$AC_MEAS_VALID	入力変数の有効ビット
\$AA_MEAS_POINT1[軸]	計測点 1
\$AA_MEAS_POINT2[軸]	\$AC_MEAS_WP_SETANGLE には無関係な計測点 2
\$AA_MEAS_POINT3[軸]	計測点 3
\$AA_MEAS_POINT4[軸]	\$AC_MEAS_CORNER_SETANGLE には無関係な計測点 4
\$AA_MEAS_WP_SETANGLE	ワーク位置指令角度*
\$AA_MEAS_CORNER_SETANGLE	交点の指令角度*
\$AA_MEAS_SETPOINT[軸]	コーナーの指令位置*
\$AC_MEAS_ACT_PLANE	特に指定されていない限り、動作中の平面に従って計算*
\$AC_MEAS_FINE_TRANS	0: 汎用オフセット、1: 精密オフセット*
\$AC_MEAS_FRAME_SELECT	特に指定されていない限り、追加フレームに従って計算*
\$AC_MEAS_T_NUMBER	特に指定されていない限り、動作中の T に従って計算(T0) *
\$AC_MEAS_D_NUMBER	特に指定されていない限り、動作中の D に従って計算(D0) *

## 9.5 原点の設定、ワーク測定、および工具測定

入力変数	意味
\$AC_MEAS_INPUT[0]	外部コーナーの指定なし* =0: 外部コーナーの計測 =1: 内部コーナーの計測
\$AC_MEAS_TYPE	4, 5, 6, 7

\* オプションです

次の変数は、計測タイプ 4～7 のために書き込まれます。

出力変数	意味
\$AC_MEAS_FRAME	平行移動と回転のあるフレーム結果
\$AC_MEAS_WP_ANGLE	ワーク位置計算角度
\$AC_MEAS_CORNER_ANGLE	計算交点角度
\$AC_MEAS_RESULTS[0]	計算された頂点の横座標
\$AC_MEAS_RESULTS[1]	計算された頂点の縦座標
\$AC_MEAS_RESULTS[2]	計算された頂点の垂直座標(輪郭に垂直な軸の座標)

## 例

コーナー計測 C1: 3つの計測点(P1、P3、およびP4)、既知の交点角度  $\phi(90^\circ)$ 、および未知のワーク位置角度  $\alpha$  を持つコーナー。

プログラムコード	コメント
DEF INT RETVAL	
DEF FRAME TMP	
\$TC_DP1[1,1]=120	; タイプ
\$TC_DP2[1,1]=20	; 0
\$TC_DP3[1,1]=10	; (z)長補正ベクトル
\$TC_DP4[1,1]=0	; (y)
\$TC_DP5[1,1]=0	; (x)
\$TC_DP6[1,1]=2	; 半径
T1 D1	
g0 x0 y0 z0 f10000	
G54	
\$P_CHBFRAME[0] = crot(z,45)	

## 9.5 原点の設定、ワーク測定、および工具測定

プログラムコード	コメント
\$P_IFRAME[x,tr] = -sin(45)	
\$P_IFRAME[y,tr] = -sin(45)	
\$P_PFRAME[z,tr] = -45	
	; 3つの計測点を持つコーナーを計測
\$AC_MEAS_VALID = 0	; すべての入力値を無効に設定
g1 x-1 y-3	; 1. 番目の計測点へアプローチ
\$AC_MEAS_LATCH[0] = 1	; 計測点 P1 を取得
g1 x-4 y4	; 3. 番目の計測点へアプローチ
\$AC_MEAS_LATCH[2] = 1	; 計測点 P3 を取得
g1 x-4 y1	; 4. 番目の計測点へアプローチ
\$AC_MEAS_LATCH[3] = 1	; 計測点 P4 を取得
\$AA_MEAS_SETPOINT[x] = 0	; コーナーの位置指令値を設定 (0, 0, 0)
\$AA_MEAS_SETPOINT[y] = 0	
\$AA_MEAS_SETPOINT[z] = 0	
\$AC_MEAS_CORNER_SETANGLE = 90	; 交点の指令角度 $\phi$ を定義
\$AC_MEAS_ACT_PLANE = 0	; 計測面は G17
\$AC_MEAS_FRAME_SELECT = 0	; フレーム- SETFRAME を選択
\$AC_MEAS_T_NUMBER = 1	; 工具選択
\$AC_MEAS_D_NUMBER = 1	
\$AC_MEAS_TYPE = 4	; コーナー 1 に計測タイプを設定
RETVAL = MEASURE()	; 計測処理を開始
if RETVAL <> 0	
setal(61000 + RETVAL)	
endif	
if \$AC_MEAS_CORNER_ANGLE <> 90	; 交点の既知の指令角度 $\phi$ を確認します
setal(61000 + \$AC_MEAS_CORNER_ANGLE)	
endif	
\$P_SETFRAME = \$AC_MEAS_FRAME	
\$P_SETFR = \$P_SETFRAME	; データ管理のシステムフレームを記述
g1 x0 y0	; コーナーへアプローチ

プログラムコード	コメント
g1 x10	; 長方形にアプローチ
y10	
x0	
y0	
m30	

### 9.5.3.3 穴計測(計測タイプ 8)

#### 穴を決定するための計測点(\$AC\_MEAS\_TYPE = 8)

中心点と直径を決定するには、3つの計測点が必要です。この3つの位置はそれぞれ、別の位置にしてください。4つの位置を指定すると、最小二乗法方式に従って円弧が調整されます。円弧は、円弧までの4つの位置の距離の二乗の合計が最小になるように決定されます。この計算の結果は、読み取ることができます。

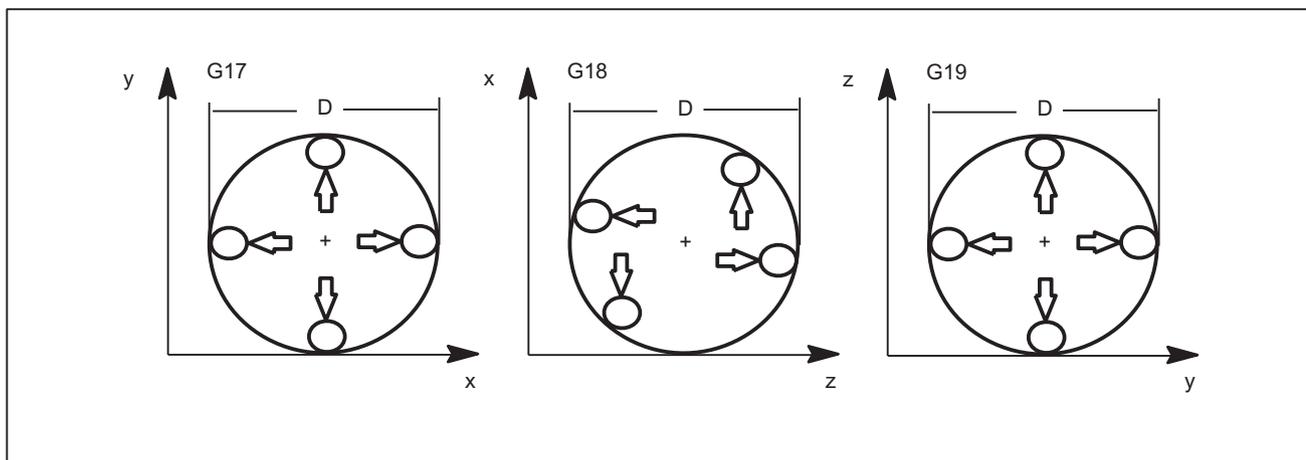


図 9-6 穴

次の変数の値が計測タイプ 8 のために使用されます。

入力変数	意味
\$AC_MEAS_VALID	入力変数の有効ビット
\$AA_MEAS_POINT1[軸]	計測点 1
\$AA_MEAS_POINT2[軸]	計測点 2
\$AA_MEAS_POINT3[軸]	計測点 3

入力変数	意味
\$AA_MEAS_POINT4[軸]	指定した場合は、4つの位置から中心が決定されます*
\$AA_MEAS_SETPOINT[軸]	穴の中心の指令位置*
\$AC_MEAS_ACT_PLANE	特に指定されていない限り、動作中の平面に従って計算*
\$AC_MEAS_FINE_TRANS	0:汎用オフセット、1:精密オフセット*
\$AC_MEAS_FRAME_SELECT	特に指定されていない限り、追加フレームに従って計算*
\$AC_MEAS_T_NUMBER	特に指定されていない限り、動作中のTに従って計算(T0)*
\$AC_MEAS_D_NUMBER	特に指定されていない限り、動作中のDに従って計算(D0)*
\$AC_MEAS_TYPE	8

\* オプションです

次の出力変数は、計測タイプ 8 のために書き込まれます。

出力変数	意味
\$AC_MEAS_FRAME	平行移動のあるフレーム結果
\$AC_MEAS_DIAMETER	穴直径
\$AC_MEAS_RESULTS[0]	横座標計算中心点
\$AC_MEAS_RESULTS[1]	縦座標計算中心点
\$AC_MEAS_RESULTS[2]	垂直座標計算中心点(輪郭に垂直な軸の座標)
\$AC_MEAS_RESULTS[3]	円弧計算結果: 距離の二乗の合計

## 例

### 穴計測

プログラムコード	コメント
DEF INT RETVAL	
DEF FRAME TMP	
\$TC_DP1[1,1]=120	; タイプ
\$TC_DP2[1,1]=20	; 0
\$TC_DP3[1,1]=10	; (z)長補正ベクトル

## 9.5 原点の設定、ワーク測定、および工具測定

プログラムコード	コメント
\$TC_DP4[1,1]=0	; (y)
\$TC_DP5[1,1]=0	; (x)
\$TC_DP6[1,1]=2	; 半径
T1 D1	
g0 x0 y0 z0 f10000	
G54	
	<b>: 穴計測</b>
\$AC_MEAS_VALID = 0	; すべての入力値を無効に設定
g1 x-3 y0	; 1. 番目の計測点へアプローチ
\$AA_MEAS_POINT1[x] = \$AA_IW[x]	
\$AA_MEAS_POINT1[y] = \$AA_IW[y]	
\$AA_MEAS_POINT1[z] = \$AA_IW[z]	
g1 x0 y3	; 2. 番目の計測点へアプローチ
\$AA_MEAS_POINT2[x] = \$AA_IW[x]	
\$AA_MEAS_POINT2[y] = \$AA_IW[y]	
\$AA_MEAS_POINT2[z] = \$AA_IW[z]	
g1 x3 y0	; 3. 番目の計測点へアプローチ
\$AA_MEAS_POINT3[x] = \$AA_IW[x]	
\$AA_MEAS_POINT3[y] = \$AA_IW[y]	
\$AA_MEAS_POINT3[z] = \$AA_IW[z]	
\$AA_MEAS_SETPOINT[x] = 0	; 中心の指令位置を設定
\$AA_MEAS_SETPOINT[y] = 0	
\$AA_MEAS_SETPOINT[z] = 0	
\$AC_MEAS_ACT_PLANE = 0	; 計測面は G17
\$AC_MEAS_FRAME_SELECT = 0	; フレーム- SETFRAME を選択
\$AC_MEAS_T_NUMBER = 1	; 工具選択
\$AC_MEAS_D_NUMBER = 1	
\$AC_MEAS_TYPE = 8	; 穴に計測タイプを設定
RETVAl = MEASURE()	; 計測処理を開始
if RETVAL <> 0	
setal(61000 + RETVAL)	
endif	
if \$AC_MEAS_DIAMETER <> 10	; 既知の直径を確認します

プログラムコード	コメント
setal(61000 + \$AC_MEAS_WP_ANGLE)	
endif	
\$P_SETFRAME = \$AC_MEAS_FRAME	
\$P_SETFR = \$P_SETFRAME	; データ管理のシステムフレームを記述
g1 x-3 y0	; P1 へアプローチ
g2 I = \$AC_MEAS_DIAMETER / 2	; 円弧中心を基準にして穴へアプローチ
m30	

### 9.5.3.4 シャフトの計測(計測タイプ 9)

#### シャフトを決定するための計測点(\$AC\_MEAS\_TYPE = 9)

中心点と直径を決定するには、3つの計測点が必要です。この3つの位置はそれぞれ、別の位置にしてください。4つの位置を指定すると、最小二乗法方式に従って円弧が調整されます。円弧は、円弧までの4つの位置の距離の二乗の合計が最小になるように決定されます。この計算の結果は、読み取ることができます。

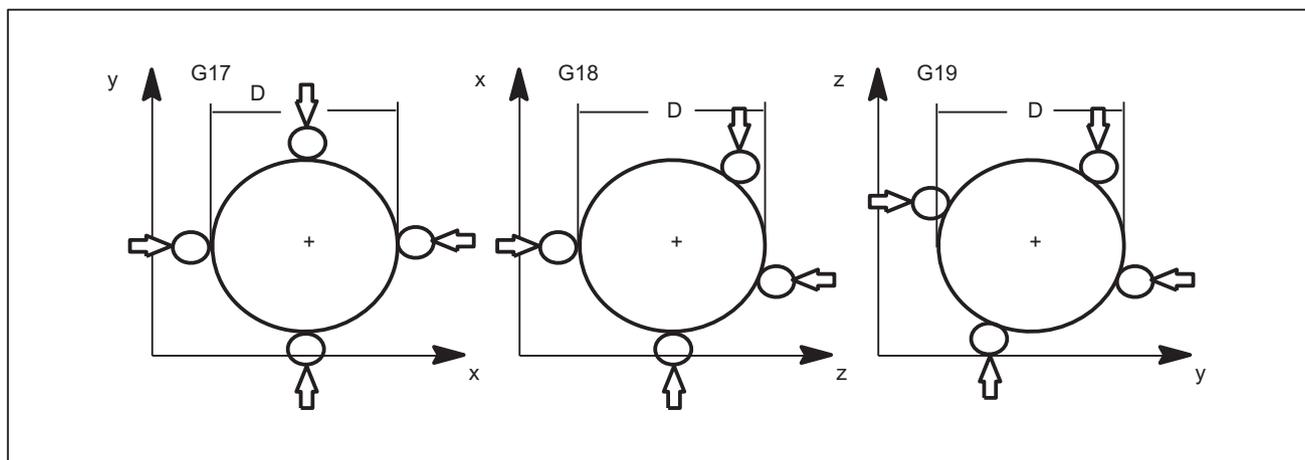


図 9-7 シャフト

次の変数の値が計測タイプ 9 のために使用されます。

入力変数	意味
\$AC_MEAS_VALID	入力変数の有効ビット
\$AA_MEAS_POINT1[軸]	計測点 1

## 9.5 原点の設定、ワーク測定、および工具測定

入力変数	意味
\$AA_MEAS_POINT2[軸]	計測点 2
\$AA_MEAS_POINT3[軸]	計測点 3
\$AA_MEAS_POINT4[軸]	指定した場合は、4 つの位置から中心が決定されます*
\$AA_MEAS_SETPPOINT[軸]	シャフト中心点の指令位置*
\$AC_MEAS_ACT_PLANE	特に指定されていない限り、動作中の平面に従って計算*
\$AC_MEAS_FINE_TRANS	0: 汎用オフセット、1: 精密オフセット*
\$AC_MEAS_FRAME_SELECT	特に指定されていない限り、追加フレームに従って計算*
\$AC_MEAS_T_NUMBER	特に指定されていない限り、動作中の T に従って計算(T0) *
\$AC_MEAS_D_NUMBER	特に指定されていない限り、動作中の D に従って計算(D0) *
\$AC_MEAS_TYPE	9

\* オプションです

次の出力変数は、計測タイプ 9 のために書き込まれます。

出力変数	意味
\$AC_MEAS_FRAME	平行移動のあるフレーム結果
\$AC_MEAS_DIAMETER	穴直径
\$AC_MEAS_RESULTS[0]	横座標計算中心点
\$AC_MEAS_RESULTS[1]	縦座標計算中心点
\$AC_MEAS_RESULTS[2]	垂直座標計算中心点(輪郭に垂直な軸の座標)
\$AC_MEAS_RESULTS[3]	円弧計算結果:距離の二乗の合計

## 9.5.3.5 溝の計測(計測タイプ 12)

## 溝位置を決定するための計測点(\$AC\_MEAS\_TYPE = 12)

溝は、2つの外部コーナーまたは内部エッジへのアプローチによって計測されます。溝の中心は、指令位置に設定できます。アプローチ方向の成分により、溝位置が決定されます。

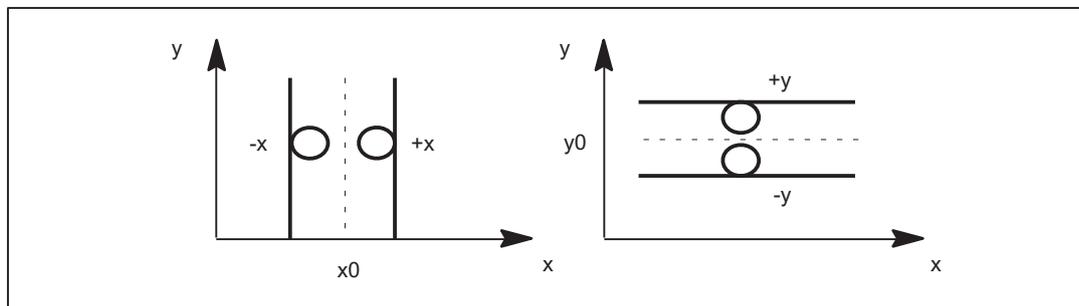


図 9-8 溝

次の変数の値が計測タイプ 12 のために使用されます。

入力変数	意味
\$AC_MEAS_VALID	入力変数の有効ビット
\$AA_MEAS_POINT1[軸]	計測点 1
\$AA_MEAS_POINT2[軸]	計測点 2
\$AA_MEAS_SETPOINT[軸]	溝の中心の指令位置*
\$AC_MEAS_DIR_APPROACH	0: +x、1: -x、2: +y、3: -y、4: +z、5: -z
\$AC_MEAS_ACT_PLANE	特に指定されていない限り、動作中の平面に従って計算*
\$AC_MEAS_FINE_TRANS	0: 汎用オフセット、1: 精密オフセット*
\$AC_MEAS_FRAME_SELECT	特に指定されていない限り、追加フレームに従って計算*
\$AC_MEAS_T_NUMBER	特に指定されていない限り、動作中の T に従って計算(T0) *
\$AC_MEAS_D_NUMBER	特に指定されていない限り、動作中の D に従って計算(D0) *

## 9.5 原点の設定、ワーク測定、および工具測定

入力変数	意味
\$AC_MEAS_INPUT[0]	凹部計測の 2 番目の計測点のアプローチ方向 1 番目の位置のアプローチ方向と同じ座標にしてください。* 0: +x、1:-x、2:+y、3:-y、4:+z、5: -z
\$AC_MEAS_TYPE	12

\* オプションです

次の出力変数は、計測タイプ 12 のために書き込まれます。

出力変数	意味
\$AC_MEAS_FRAME	平行移動のあるフレーム結果
\$AC_MEAS_RESULTS[0]	計算溝中心位置(x0、y0 または z0)
\$AC_MEAS_RESULTS[1]	アプローチ方向の溝幅

## 例

アプローチ方向が x の溝の計測

プログラムコード	コメント
DEF INT RETVAL	
DEF FRAME TMP	
\$TC_DP1[1,1]=120	; タイプ
\$TC_DP2[1,1]=20	; 0
\$TC_DP3[1,1]=10	; (z)長補正ベクトル
\$TC_DP4[1,1]=0	; (y)
\$TC_DP5[1,1]=0	; (x)
\$TC_DP6[1,1]=2	; 半径
T1 D1	
G0 x0 y0 z0 f10000	
G54	
\$P_CHBFRAME[0] = crot(z,45)	
\$P_IFRAME[x,tr] = -sin(45)	
\$P_IFRAME[y,tr] = -sin(45)	
\$P_PFRAME[z,rt] = -45	
	; 溝計測
\$AC_MEAS_VALID = 0	; すべての入力値を無効に設定

## 9.5 原点の設定、ワーク測定、および工具測定

プログラムコード	コメント
g1 x-2 \$AA_MEAS_POINT1[x] = \$AA_IW[x] \$AA_MEAS_POINT1[y] = \$AA_IW[y] \$AA_MEAS_POINT1[z] = \$AA_IW[z]	; 1. 番目の計測点へアプローチ
g1 x4 \$AA_MEAS_POINT2[x] = \$AA_IW[x] \$AA_MEAS_POINT2[y] = \$AA_IW[y] \$AA_MEAS_POINT2[z] = \$AA_IW[z]	; 2. 番目の計測点へアプローチ
\$AA_MEAS_SETPOINT[x] = 0 \$AA_MEAS_SETPOINT[y] = 0 \$AA_MEAS_SETPOINT[z] = 0	; 溝の中心の指令位置を設定
\$AC_MEAS_DIR_APPROACH = 0 \$AC_MEAS_ACT_PLANE = 0 \$AC_MEAS_FRAME_SELECT = 0	; アプローチ方向+xを設定 ; 計測面はG17 ; フレーム- SETFRAMEを選択
\$AC_MEAS_T_NUMBER = 1 \$AC_MEAS_D_NUMBER = 1	; 工具選択
\$AC_MEAS_TYPE = 12	; 溝の計測タイプを設定
RETVAL = MEASURE()	; 計測処理を開始
if RETVAL <> 0 setal(61000 + RETVAL) endif	
\$P_SETFRAME = \$AC_MEAS_FRAME \$P_SETFR = \$P_SETFRAME	; データ管理のシステムフレームを記述
g1 x0 y0	; 溝の中心へアプローチ
m30	

## 9.5.3.6 ウェブの計測(計測タイプ 13)

## ウェブの位置を決定するための計測点(\$AC\_MEAS\_TYPE = 13)

ウェブは、2つの外部コーナーまたは内部エッジへのアプローチによって計測されます。ウェブの中心は、指令位置に設定できます。アプローチ方向の成分により、ウェブの位置が決定されます。

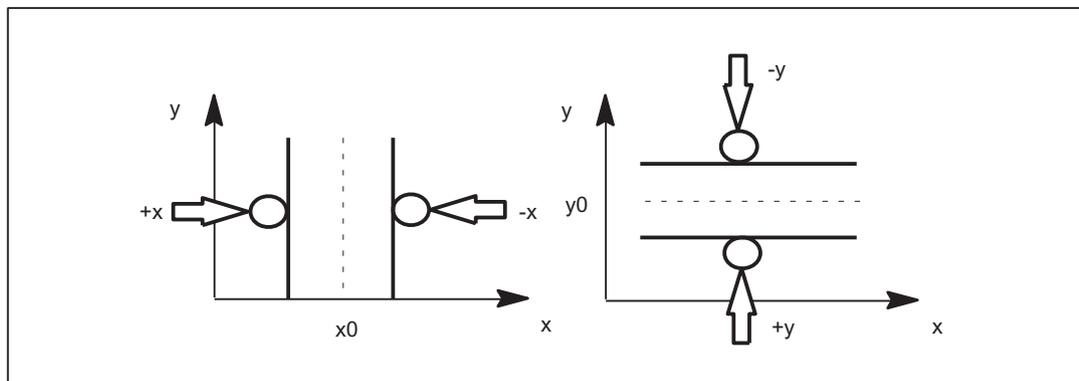


図 9-9 ウェブ

次の変数の値が計測タイプ 13 のために使用されます。

入力変数	意味
\$AC_MEAS_VALID	入力変数の有効ビット
\$AA_MEAS_POINT1[軸]	計測点 1
\$AA_MEAS_POINT2[軸]	計測点 2
\$AA_MEAS_SETPOINT[軸]	ウェブ中心の指令位置*
\$AC_MEAS_DIR_APPROACH	0: +x, 1: -x, 2: +y, 3: -y, 4: +z, 5: -z
\$AC_MEAS_ACT_PLANE	特に指定されていない限り、動作中の平面に従って計算*
\$AC_MEAS_FINE_TRANS	0: 汎用オフセット、1: 精密オフセット*
\$AC_MEAS_FRAME_SELECT	特に指定されていない限り、追加フレームに従って計算*
\$AC_MEAS_T_NUMBER	特に指定されていない限り、動作中の T に従って計算(T0) *
\$AC_MEAS_D_NUMBER	特に指定されていない限り、動作中の D に従って計算(D0) *

入力変数	意味
\$AC_MEAS_INPUT[0]	凹部計測の 2 番目の計測点のアプローチ方向 1 番目の位置のアプローチ方向と同じ座標にしてください。 * 0: +x、1:-x、2:+y、3:-y、4:+z、5: -z
\$AC_MEAS_TYPE	13

\* オプションです

次の出力変数は、計測タイプ 13 のために書き込まれます。

出力変数	意味
\$AC_MEAS_FRAME	平行移動のあるフレーム結果
\$AC_MEAS_RESULTS[0]	ウェブ中心計算位置(x0、y0 または z0)
\$AC_MEAS_RESULTS[1]	アプローチ方向のウェブ幅

### 9.5.3.7 ジオメトリ軸と付加軸の計測(計測タイプ 14、15)

#### ジオメトリ軸と付加軸の現在位置設定(\$AC MEAS TYPE = 14)

この計測タイプは、HMI 操作画面で使用されます。

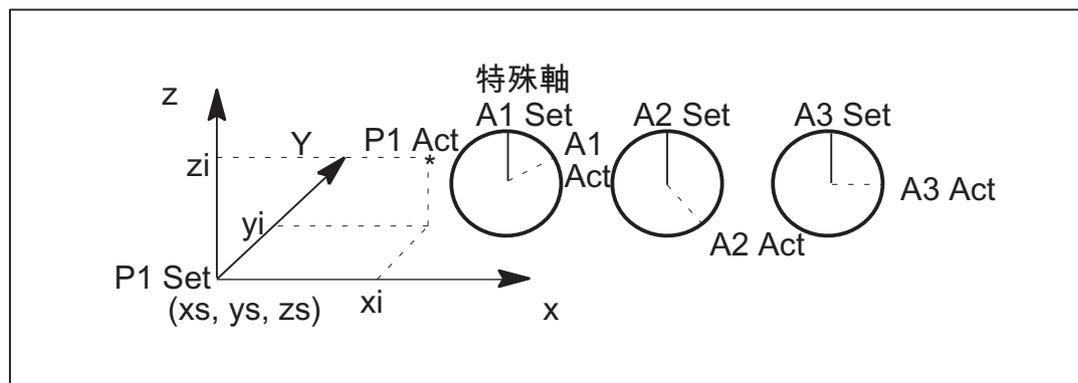


図 9-10 現在位置設定

次の変数の値が計測タイプ 14 のために使用されます。

入力変数	意味
\$AC_MEAS_VALID	入力変数の有効ビット
\$AA_MEAS_POINT1[軸]	軸の現在位置

## 9.5 原点の設定、ワーク測定、および工具測定

入力変数	意味
\$AA_MEAS_SETPOINT[軸]	個々の軸の指令位置*
\$AC_MEAS_ACT_PLANE	特に指定されていない限り、動作中の平面に従って計算*
\$AC_MEAS_FINE_TRANS	0: 汎用オフセット、1: 精密オフセット*
\$AC_MEAS_FRAME_SELECT	特に指定されていない限り、追加フレームに従って計算*
\$AC_MEAS_T_NUMBER	特に指定されていない限り、動作中の T に従って計算(T0) *
\$AC_MEAS_D_NUMBER	特に指定されていない限り、動作中の D に従って計算(D0) *
\$AC_MEAS_TYPE	14

\* オプションです

次の出力変数は、計測タイプ 14 のために書き込まれます。

出力変数	意味
\$AC_MEAS_FRAME	平行移動のあるフレーム結果

## 例

相対座標系でのレファレンス点設定

プログラムコード	コメント
DEF INT RETVAL	
T1 D1	; プローブを有効化
G54	; すべてのフレームと G54 の有効化
TRANS x=10	; WCS と ENS の間のオフセット
G0 x0 f10000	; WCS(x) = 0; ENS(x) = 10
\$AC_MEAS_VALID = 0	; すべての入力変数を無効に設定
\$AC_MEAS_TYPE = 14	; <b>現在位置設定の計測タイプ</b>
\$AC_MEAS_ACT_PLANE = 0	; 計測面は G17
\$AC_MEAS_P1_COORD = 5	; 1 番目の計測点の ENS_REL
\$AC_MEAS_LATCH[0] = 1	; すべての軸の位置を取得
\$AC_MEAS_SET_COORD = 5	; 指令位置が ENS 基準
\$AA_MEAS_SETPOINT[x] = 0	; 相対 ENS 座標系での指令位置

プログラムコード	コメント
\$AC_MEAS_FRAME_SELECT = 2505	; \$P_RELFR
RETVAL = MEASURE()	; \$P_RELFR; PI SETUDT(6)の計算
IF RETVAL <> 0 GOTOF ERROR	
ENDIF \$ P_RELFR = \$AC_MEAS_FRAME	; 適用; PI SETUDT(7)

### 付加軸のみの現在位置設定(\$AC MEAS TYPE = 15)

この計測タイプは、HMI 操作画面で使用されます。

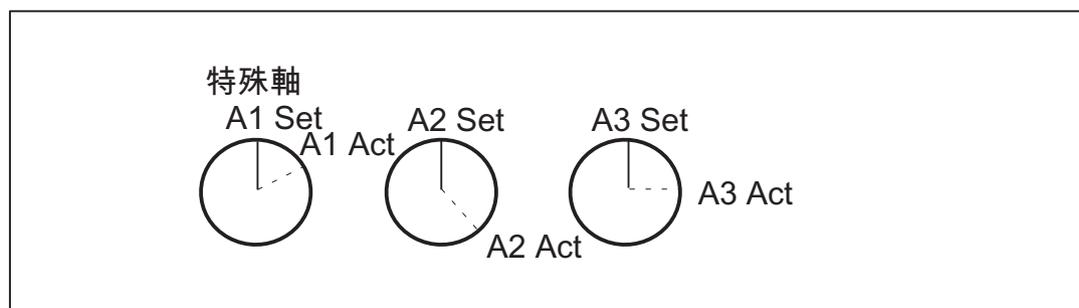


図 9-11 付加軸のみの現在位置設定

次の変数の値が計測タイプ 15 のために使用されます。

入力変数	意味
\$AC_MEAS_VALID	入力変数の有効ビット
\$AA_MEAS_POINT1[軸]	軸の現在位置
\$AA_MEAS_SETPOINT[軸]	個々の軸の指令位置*
\$AC_MEAS_FINE_TRANS	0: 汎用オフセット、1: 精密オフセット*
\$AC_MEAS_FRAME_SELECT	特に指定されていない限り、追加フレームに従って計算*
\$AC_MEAS_TYPE	15

\* オプションです

次の出力変数は、計測タイプ 15 に従って書き込まれます。

出力変数	意味
\$AC_MEAS_FRAME	平行移動のあるフレーム結果

## 9.5.3.8 傾斜エッジの計測(計測タイプ 16)

## 傾斜エッジの計測(\$AC\_MEAS\_TYPE = 16)

この計測により、ワークの位置角度が決定され、フレームに入力されます。入力できるのは、 $\pm 90^\circ$ の範囲の指令角度です。これは、動作中の WCS のフレーム結果が有効になった後のワークの回転結果と見なすことができます。

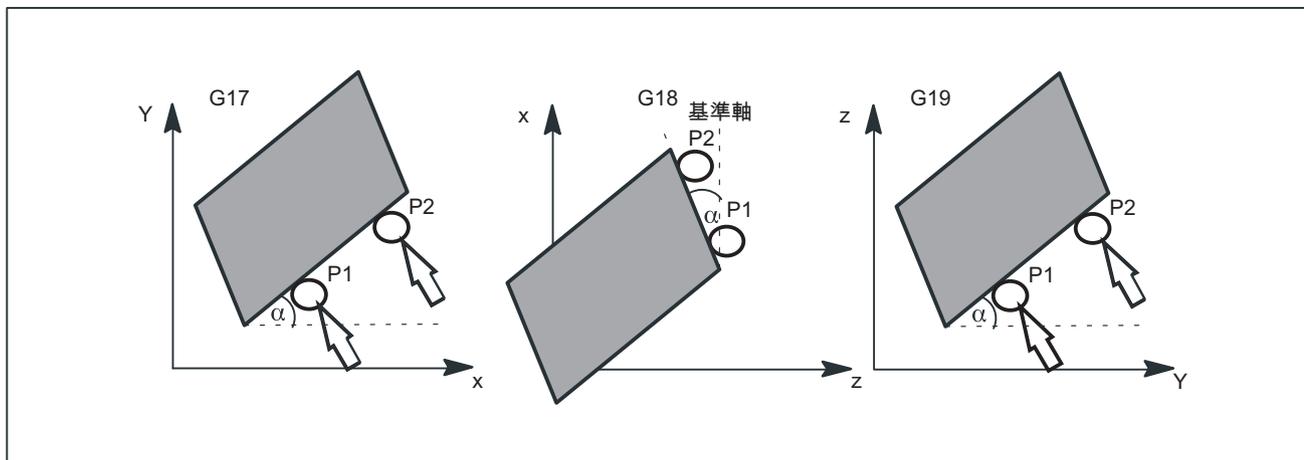


図 9-12 平面 G17、G18、および G19 の傾斜エッジ

次の変数の値が計測タイプ 16 のために使用されます。

入力変数	意味
\$AC_MEAS_VALID	入力変数の有効ビット
\$AA_MEAS_POINT1[軸]	計測点 1
\$AA_MEAS_POINT2[軸]	計測点 2
\$AA_MEAS_SETANGLE	指令角度*
\$AC_MEAS_ACT_PLANE	特に指定されていない限り、動作中の平面に従って計算*
\$AC_MEAS_FINE_TRANS	0: 汎用オフセット、1: 精密オフセット*
\$AC_MEAS_FRAME_SELEC T	特に指定されていない限り、追加フレームに従って計算*
\$AC_MEAS_T_NUMBER	特に指定されていない限り、動作中の T に従って計算 (T0) *
\$AC_MEAS_D_NUMBER	特に指定されていない限り、動作中の D に従って計算 (D0) *

## 9.5 原点の設定、ワーク測定、および工具測定

入力変数	意味
\$AC_MEAS_INPUT[0]	特に指定されていない限り、ワークの位置合わせの基準座標は常に、選択された平面の横座標です。 * =0: 基準座標は横座標です =1: 基準座標は縦座標です
\$AC_MEAS_INPUT[1]	特に指定されていない限り、ワーク位置角度はフレームに、座標回転として入力されます。それ以外の場合は、回転軸の座標系設定が、現在の回転軸位置+計算回転量で設定されたその回転軸に対して、チャンネル軸インデックスを指定できます。その後、ワークは回転軸の位置=0に位置合わせされます。現在の回転軸の値は、\$AA_MEAS_POINT[軸]に設定してください。 *
\$AC_MEAS_TYPE	16

\* オプションです

次の出力変数は、計測タイプ 16 のために書き込まれます。

出力変数	意味
\$AC_MEAS_FRAME	座標回転フレーム結果
\$AC_MEAS_WP_ANGLE	ワーク位置計算角度

## 9.5.3.9 平面での傾斜角度の計測(計測タイプ 17)

## 傾斜面の角度計測(\$AC\_MEAS\_TYPE = 17)

傾斜面は、3つの計測点 P1、P2、および P3 を使用して決定されます。

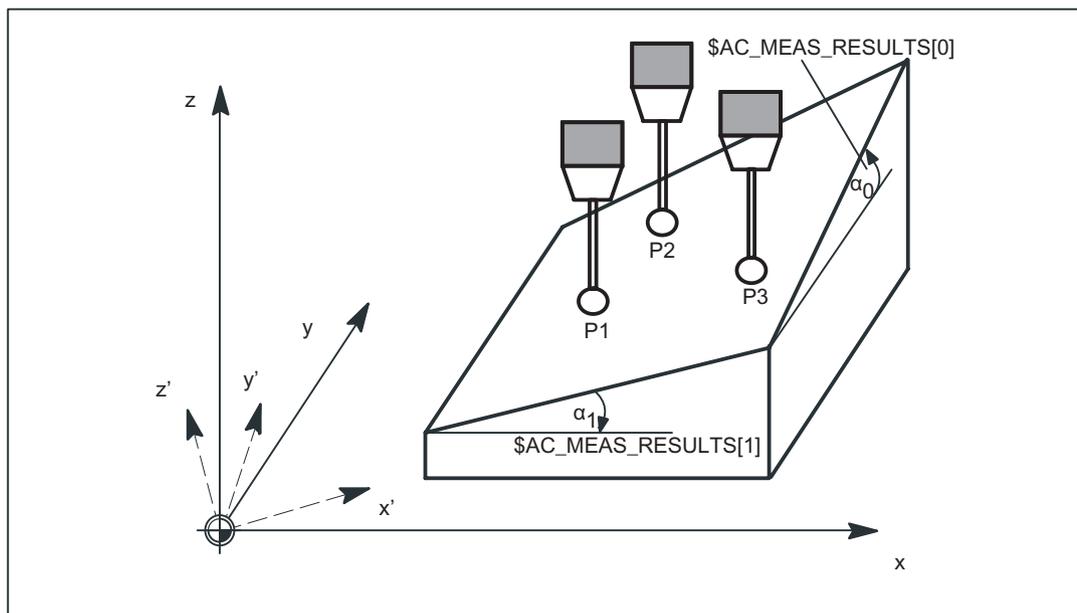


図 9-13 G17 の傾斜面

\$AC\_MEAS\_TYPE = 17 により、2つの角度  $\alpha_0$  と  $\alpha_1$  の結果が平面の傾きとして定義されます。これらは、次のように \$AC\_MEAS\_RESULTS[0..1] に入力されます。

- \$AC\_MEAS\_RESULTS[0] → 横座標の回転
- \$AC\_MEAS\_RESULTS[1] → 縦座標の回転

これらの角度は、3つの計測点 P1、P2、および P3 によって計算されます。このタイプの計測では、垂直座標(輪郭に垂直な軸の座標)の角度(\$AC\_MEAS\_RESULTS[2])には常に、事前に 0 が入力されます。

フレーム結果に入力された回転指令値は、横座標および(または)縦座標として入力できます。指令値で1つの角度のみが指定されている場合は、2番目の角度が、3つの計測点が指令角度の同一傾斜面上に置かれるように計算されます。フレーム結果には座標回転のみが入力され、WCS レファレンス点は保持されます。WCS は、 $z'$  が傾斜面に垂直になるように回転されます。

次の平面の設定は、計測タイプ 17 のために定義されます。

軸識別子	G17	G18	G19
横座標	x 軸	z 軸	y 軸
縦座標	y 軸	x 軸	z 軸
垂直座標(切り込み軸)	z 軸	y 軸	x 軸

次の変数の値が計測タイプ 17 のために使用されます。

入力変数	意味
\$AC_MEAS_VALID	入力変数の有効ビット
\$AA_MEAS_POINT1[軸]	計測点 1
\$AA_MEAS_POINT2[軸]	計測点 2
\$AA_MEAS_POINT3[軸]	計測点 3
\$AA_MEAS_SETANGLE[軸]	横座標と縦座標を中心とした回転座標指令値*
\$AC_MEAS_ACT_PLANE	特に指定されていない限り、動作中の平面に従って計算 *
\$AC_MEAS_FRAME_SELECT	特に指定されていない限り、追加フレームに従って計算 *
\$AC_MEAS_T_NUMBER	特に指定されていない限り、動作中の T に従って計算 (T0) *
\$AC_MEAS_D_NUMBER	特に指定されていない限り、動作中の D に従って計算 (D0) *
\$AC_MEAS_INPUT[0]	特に指定されていない限り、 各位置は平面上に投影されません* 0: 各位置は平面上に投影されません 1: 各位置は、動作中の平面または選択された平面上に 投影されます
\$AC_MEAS_TYPE	17

\* オプションです

9.5 原点の設定、ワーク測定、および工具測定

次の出力変数は、計測タイプ 17 のために書き込まれます。

出力変数	意味
\$AC_MEAS_FRAME	フレーム結果
\$AC_MEAS_RESULTS[0]	3つの計測点の計算に使用される、横座標を中心とした角度
\$AC_MEAS_RESULTS[1]	3つの計測点の計算に使用される、縦座標を中心とした角度
\$AC_MEAS_RESULTS[2]	3つの計測点の計算に使用される、垂直座標を中心とした角度
\$AC_MEAS_RESULTS[3]	フレーム結果に入力される、横座標を中心とした角度
\$AC_MEAS_RESULTS[4]	フレーム結果に入力される、縦座標を中心とした角度
\$AC_MEAS_RESULTS[5]	フレーム結果に入力される、垂直座標を中心とした角度

例

平面角度の計測

プログラムコード	コメント
DEF INT RETVAL DEF AXIS _XX, _YY, _ZZ	
T1 D1 G54	; プローブを有効化 ; すべてのフレームと G54 の有効化
\$AC_MEAS_VALID = 0	; すべての入力値を無効に設定
\$AC_MEAS_TYPE = 17 \$AC_MEAS_ACT_PLANE = 0	; 傾斜面の計測タイプを設定 ; 計測面は G17
_XX=\$P_AXN1 _YY=\$P_AXN2 _ZZ=\$P_AXN3	; 平面に従って軸を定義
G17 G1 _XX=10 _YY=10 F1000 MEAS = 1 _ZZ=...	; 1. 番目の計測点へアプローチ
\$AA_MEAS_POINT1[_xx] = \$AA_MW[_xx] \$AA_MEAS_POINT1[_yy] = \$AA_MW[_yy] \$AA_MEAS_POINT1[_zz] = \$AA_MW[_zz]	; 計測値を横座標に割り当て ; 計測値を縦座標に割り当て ; 計測値を垂直座標に割り当て
G1 _XX=20 _YY=10 F1000	; 2. 番目の計測点へアプローチ

## 9.5 原点の設定、ワーク測定、および工具測定

プログラムコード	コメント
MEAS = 1 _ZZ=...	
\$AA_MEAS_POINT2[_xx] = \$AA_MW[_xx]	; 計測値を横座標に割り当て
\$AA_MEAS_POINT2[_yy] = \$AA_MW[_yy]	; 計測値を縦座標に割り当て
\$AA_MEAS_POINT2[_zz] = \$AA_MW[_zz]	; 計測値を垂直座標に割り当て
G1 _XX=20 _YY=20 F1000	; 3. 番目の計測点へアプローチ
MEAS = 1 _ZZ=...	
\$AA_MEAS_POINT3[_xx] = \$AA_MW[_xx]	; 計測値を横座標に割り当て
\$AA_MEAS_POINT3[_yy] = \$AA_MW[_yy]	; 計測値を縦座標に割り当て
\$AA_MEAS_POINT3[_zz] = \$AA_MW[_zz]	; 計測値を垂直座標に割り当て
	; 角度の指令値を定義
\$AA_MEAS_SETANGLE[_xx] = 12	; 横座標を中心とした回転
\$AA_MEAS_SETANGLE[_yy] = 4	; 縦座標を中心とした回転
\$AC_MEAS_FRAME_SELECT = 102	; 対象フレーム- G55 を選択
\$AC_MEAS_T_NUMBER = 1	; 工具選択
\$AC_MEAS_D_NUMBER = 1	
RETVL = MEASURE()	; 計測計算の開始
if RETVAL <> 0	
setal(61000 + RETVAL)	
endif	
if \$AC_MEAS_RESULTS[0] <> 12	
setal(61000 + \$AC_MEAS_RESULTS[0])	
endif	
if \$AC_MEAS_RESULTS[1] <> 4	
setal(61000 + \$AC_MEAS_RESULTS[1])	
endif	
\$P_UIFR[2] = \$AC_MEAS_FRAME	; 計測フレームをデータ管理に書き込み (G55)
G55 G0 AX[_xx]=10 AX[_yy]=10	; フレームを有効化して移動
m30	

## 9.5.3.10 WCS 基準フレーム回転計測の再定義(計測タイプ 18)

## WCS 座標系の再定義(\$AC\_MEAS\_TYPE = 18)

新しい WCS の原点は、計測点 P1 によって、傾斜面上の平面に垂直に決定されます。

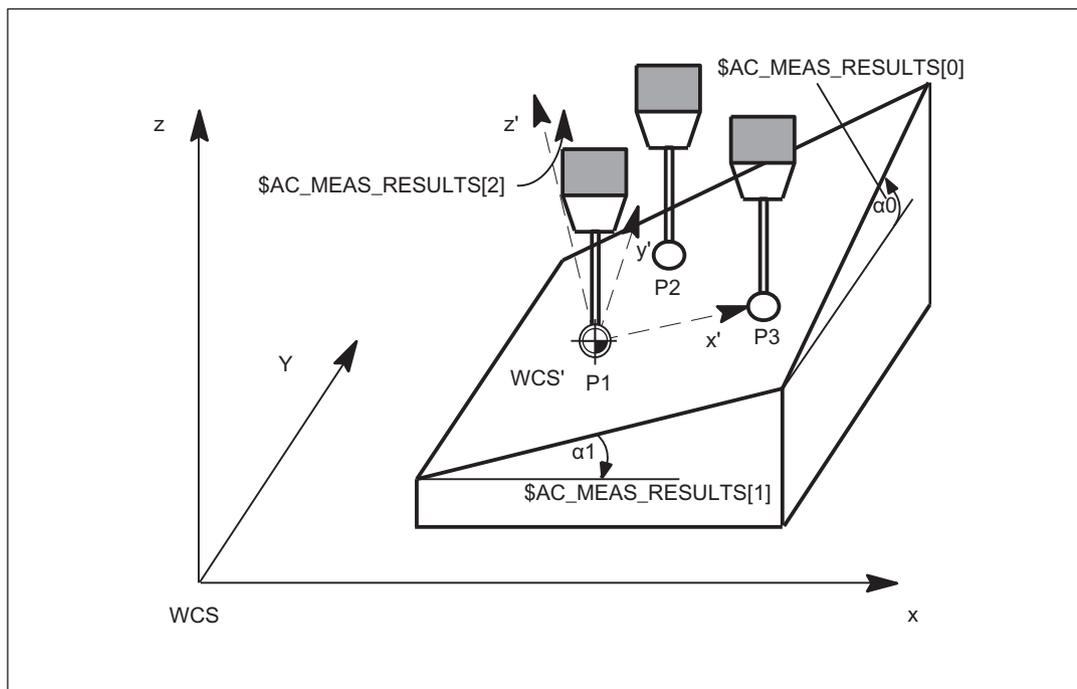


図 9-14 G17 の傾斜面

## 平面の計測

平面は、1つの計測サイクルで計測されます。このサイクルは3つの計測点を記録し、必要な値を変数インタフェースに渡します。

MEASURE() 機能は、入力値に基づいて、新しい WCS' の立体角と平行移動オフセットを計算します。

## 計測フレームの座標変換

計算結果、つまり、立体角と平行移動は、計測フレーム\$SAC\_MEAS\_FRAME に入力されます。計測フレームは、フレーム結合で選択されたフレームに従って、変換されます。立体角も、出力値\$SAC\_MEAS\_RESULTS[0..2]に保存されます。In

- 古い WCS の横座標を中心とした角度は\$SAC\_MEAS\_RESULTS[0]に保存されます。
- 縦座標を中心とした角度は\$SAC\_MEAS\_RESULTS[1]に保存されます。
- 垂直座標を中心とした角度は\$SAC\_MEAS\_RESULTS[2]に保存されます。

## 新しい WCS'原点の定義

計算をおこなった後に、計測サイクルは計測フレームを使って、フレーム結合で選択されたフレームに書き込み、有効にすることができます。有効にした後に、新しい WCS は、計測点 P1 をその原点として、傾斜面上の平面に垂直に位置決めされます。

プログラム指令位置は、傾斜面を基準としています。

## 用途

CAD システムでは、この平面上に P1、P2、および P3 の 3 点を指定して、傾斜面を定義することがよくあります。この場合は、次のようになります。

- 1. 番目の計測点 P1 は、新しい WCS'レファレンス点として適用されます。
- 2. 番目の計測点 P2 により、新しく回転された WCS'座標系の横座標 x'の方向が定義されます。
- 3. 番目の計測点 P3 は、立体角の決定に使用されます。

次の変数の値が計測タイプ 18 のために使用されます。

入力変数	意味
\$SAC_MEAS_VALID	入力変数の有効ビット
\$AA_MEAS_POINT1[軸]	計測点 1
\$AA_MEAS_POINT2[軸]	計測点 2
\$AA_MEAS_POINT3[軸]	計測点 3
\$AA_MEAS_SETPOINT[軸]	P1 の指令位置*
\$SAC_MEAS_ACT_PLANE	特に指定されていない限り、動作中の平面に従って計算 *

## 9.5 原点の設定、ワーク測定、および工具測定

入力変数	意味
\$AC_MEAS_FRAME_SELECT	特に指定されていない限り、追加フレームに従って計算*
\$AC_MEAS_T_NUMBER	特に指定されていない限り、動作中の T に従って計算 (T0) *
\$AC_MEAS_D_NUMBER	特に指定されていない限り、動作中の D に従って計算 (D0) *
\$AC_MEAS_INPUT[0]	特に指定されていない限り、 各位置は平面上に投影されません* 0:各位置は平面上に投影されません 1:各位置は、動作中の平面または選択された平面上に投影されます
\$AC_MEAS_TYPE	18

\* オプションです

次の出力変数は、計測タイプ 18 のために書き込まれます。

出力変数	意味
\$AC_MEAS_FRAME	座標回転と座標変換のフレーム結果
\$AC_MEAS_RESULTS[0]	計算された横座標を中心とした角度
\$AC_MEAS_RESULTS[1]	計算された縦座標を中心とした角度
\$AC_MEAS_RESULTS[2]	計算された垂直座標を中心とした角度

## 例

傾斜面上のワーク座標系

プログラムコード	コメント
DEF INT RETVAL	
DEF AXIS _XX, _YY, _ZZ	
T1 D1	; プローブを有効化
G54	; すべてのフレームと G54 の有効化
\$AC_MEAS_VALID = 0	; すべての入力値を無効に設定
\$AC_MEAS_TYPE = 18	; 傾斜面の計測タイプを設定
\$AC_MEAS_ACT_PLANE = 0	; 計測面は G17
_XX=\$P_AXN1	; 平面に従って軸を定義

## 9.5 原点の設定、ワーク測定、および工具測定

プログラムコード	コメント
_YY=\$P_AXN2 _ZZ=\$P_AXN3	
G17 G1 _XX=10 _YY=10 F1000 MEAS = 1 _ZZ=...	; 1. 番目の計測点へアプローチ
\$AA_MEAS_POINT1[_xx] = \$AA_MW[_xx] \$AA_MEAS_POINT1[_yy] = \$AA_MW[_yy] \$AA_MEAS_POINT1[_zz] = \$AA_MW[_zz]	; 計測値を横座標に割り当て ; 計測値を縦座標に割り当て ; 計測値を垂直座標に割り当て
G1 _XX=20 _YY=10 F1000 MEAS = 1 _ZZ=...	; 2. 番目の計測点へアプローチ
\$AA_MEAS_POINT2[_xx] = \$AA_MW[_xx] \$AA_MEAS_POINT2[_yy] = \$AA_MW[_yy] \$AA_MEAS_POINT2[_zz] = \$AA_MW[_zz]	; 計測値を横座標に割り当て ; 計測値を縦座標に割り当て ; 計測値を垂直座標に割り当て
G1 _XX=20 _YY=20 F1000 MEAS = 1 _ZZ=...	; 3. 番目の計測点へアプローチ
\$AA_MEAS_POINT3[_xx] = \$AA_MW[_xx] \$AA_MEAS_POINT3[_yy] = \$AA_MW[_yy] \$AA_MEAS_POINT3[_zz] = \$AA_MW[_zz]	; 計測値を横座標に割り当て ; 計測値を縦座標に割り当て ; 計測値を垂直座標に割り当て
\$AA_MEAS_SETPPOINT[_xx] = 10 \$AA_MEAS_SETPPOINT[_yy] = 10 \$AA_MEAS_SETPPOINT[_zz] = 10	; P1 の指令値を定義
\$AC_MEAS_FRAME_SELECT = 102 \$AC_MEAS_T_NUMBER = 1 \$AC_MEAS_D_NUMBER = 1	; 対象フレーム- G55 を選択 ; 工具選択
RETVL = MEASURE() if RETVAL <> 0 setal(61000 + RETVAL) endif	; 計測計算の開始
R0 = \$AC_MEAS_RESULTS[0] R1 = \$AC_MEAS_RESULTS[1] R2 = \$AC_MEAS_RESULTS[2]	; 立体角の計算結果 ; 古い WCS の横座標を ; 中心とした角度 ; 縦座標 ; 垂直座標
\$P_UIFR[2] = \$AC_MEAS_FRAME	; 計測フレームをデータ管理に書き込み(G55)

## 9.5 原点の設定、ワーク測定、および工具測定

プログラムコード	コメント
G55 G0 AX[_xx]=10 AX[_yy]=10 m30	; フレームを有効化して移動

## 9.5.3.11 1次元、2次元、および3次元の指令値選択の計測(計測タイプ 19、20、21)

## 1次元指令値(\$AC\_MEAS\_TYPE = 19)

この計測方法では、横座標、縦座標、および垂直座標に1つだけ指令値を定義できます。2個または3個の指令値を定義した場合は、最初の指令値のみが、横座標、縦座標、垂直座標の順に受け入れられます。工具は考慮されません。

これは純粋に、横座標、縦座標、または垂直軸のための現在位置設定です。

次の変数の値が計測タイプ 19 のために使用されます。

入力変数	意味
\$AC_MEAS_VALID	入力変数の有効ビット
\$AA_MEAS_POINT1[軸]	横座標の計測点 1
\$AA_MEAS_POINT1[軸]	縦座標の計測点 1
\$AA_MEAS_POINT1[軸]	垂直座標の計測点 1
\$AA_MEAS_SETPPOINT[軸]	横座標、縦座標または垂直座標の指令位置
\$AC_MEAS_ACT_PLANE	特に指定されていない限り、動作中の平面に従って計算 *
\$AC_MEAS_FRAME_SELECT	特に指定されていない限り、追加フレームに従って計算 *
\$AC_MEAS_FINE_TRANS	特に指定されていない限り、フレームは汎用の平行移動 に書き込まれます*
\$AC_MEAS_TYPE	19

\* オプションです

次の出力変数は、計測タイプ 19 のために書き込まれます。

出力変数	意味
\$AC_MEAS_FRAME	座標回転と平行移動のフレーム結果

## 例

## 1 次元の指令値選択

プログラムコード	コメント
DEF INT RETVAL	
DEF REAL _CORMW_XX,	
_CORMW_YY,	
_CORMW_ZZ	
DEF AXIS _XX, _YY, _ZZ	
T1 D1	; プローブを有効化
G54	; すべてのフレームと G54 の有効化
\$AC_MEAS_VALID = 0	; すべての入力値を無効に設定
\$AC_MEAS_TYPE = 19	; <b>1次元指令値選択の計測タイプを設定</b>
\$AC_MEAS_ACT_PLANE = 0	; 計測面は G17
_XX=\$P_AXN1	; 平面に従って軸を定義
_YY=\$P_AXN2	
_ZZ=\$P_AXN3	
	; 計測値を割り当て
\$AA_MEAS_POINT1[_xx] = \$AA_MW[_xx]	; 計測値を横座標に割り当て
\$AA_MEAS_POINT1[_yy] = \$AA_MW[_yy]	; 計測値を縦座標に割り当て
\$AA_MEAS_POINT1[_zz] = \$AA_MW[_zz]	; 計測値を垂直座標に割り当て
\$AA_MEAS_SETPPOINT[_xx] = 10	; 横座標の指令値を定義
\$AC_MEAS_FRAME_SELECT = 102	; 対象フレーム- G55 を選択
RETVAL = MEASURE()	; 計測計算の開始
if RETVAL <> 0	
setal(61000 + RETVAL)	
endif	
\$P_UIFR[2] = \$AC_MEAS_FRAME	; 計測フレームをデータ管理に書き込み(G55)
G55 G0 AX[_xx]=10 AX[_yy]=10	; フレームを有効化して移動
m30	

## 2 次元指令値(\$AC\_MEAS\_TYPE = 20)

2次元の指令値は、この計測方法を使用して定義できます。3つの軸のうち、任意の2つの組み合わせが許容されます。3個の指令値が指定されている場合は、横座標と縦座標の値のみが受け入れられます。工具は考慮されません。

これは純粋に、現在位置設定です。

## 9.5 原点の設定、ワーク測定、および工具測定

次の変数の値が計測タイプ 20 のために使用されます。

入力変数	意味
\$AC_MEAS_VALID	入力変数の有効ビット
\$AA_MEAS_POINT1[軸]	横座標の計測点 1
\$AA_MEAS_POINT1[軸]	縦座標の計測点 1
\$AA_MEAS_POINT1[軸]	垂直座標の計測点 1
\$AA_MEAS_SETPPOINT[軸]	1 番目の次元の指令位置
\$AA_MEAS_SETPPOINT[軸]	2 次元の指令位置
\$AC_MEAS_ACT_PLANE	特に指定されていない限り、動作中の平面に従って計算 *
\$AC_MEAS_FRAME_SELECT	特に指定されていない限り、追加フレームに従って計算 *
\$AC_MEAS_FINE_TRANS	特に指定されていない限り、フレームは汎用の平行移動 に書き込まれます*
\$AC_MEAS_TYPE	20

\* オプションです

次の出力変数は、計測タイプ 20 のために書き込まれます。

出力変数	意味
\$AC_MEAS_FRAME	座標回転と平行移動のフレーム結果

## 例

## 2 次元の指令値選択

プログラムコード	コメント
DEF INT RETVAL	
DEF REAL _CORMW_XX, _CORMW_YY, _CORMW_ZZ	
DEF AXIS _XX, _YY, _ZZ	
T1 D1	; プローブを有効化
G54	; すべてのフレームと G54 の有効化
\$AC_MEAS_VALID = 0	; すべての入力値を無効に設定
\$AC_MEAS_TYPE = 20	; 2 次元指令値選択の計測タイプを設定
\$AC_MEAS_ACT_PLANE = 0	; 計測面は G17

プログラムコード	コメント
_XX=\$P_AXN1	; 平面に従って軸を定義
_YY=\$P_AXN2	
_ZZ=\$P_AXN3	
	; 計測値を割り当て
\$AA_MEAS_POINT1[_xx] = \$AA_MW[_xx]	; 計測値を横座標に割り当て
\$AA_MEAS_POINT1[_yy] = \$AA_MW[_yy]	; 計測値を縦座標に割り当て
\$AA_MEAS_POINT1[_zz] = \$AA_MW[_zz]	; 計測値を垂直座標に割り当て
\$AA_MEAS_SETPOINT[_xx] = 10	; 横座標と縦座標の指令値を定義
\$AA_MEAS_SETPOINT[_yy] = 10	
\$AC_MEAS_FRAME_SELECT = 102	; 対象フレーム- G55 を選択
RETVAl = MEASURE()	; 計測計算の開始
if RETVAL <> 0	
setal(61000 + RETVAL)	
endif	
\$P_UIFR[2] = \$AC_MEAS_FRAME	; 計測フレームをデータ管理に書き込み (G55)
G55 G0 AX[_xx]=10 AX[_yy]=10	; フレームを有効化して移動
m30	

### 3次元指令値(\$AC\_MEAS\_TYPE = 21)

この計測方法を使用すると、横座標、縦座標、および垂直座標に1つの指令値を定義できます。工具は考慮されません。

これは純粋に、横座標、縦座標、および垂直座標に対する現在位置設定です。

次の変数の値が計測タイプ 21 のために使用されます。

入力変数	意味
\$AC_MEAS_VALID	入力変数の有効ビット
\$AA_MEAS_POINT1[軸]	横座標の計測点 1
\$AA_MEAS_POINT1[軸]	縦座標の計測点 1
\$AA_MEAS_POINT1[軸]	垂直座標の計測点 1
\$AA_MEAS_SETPOINT[軸]	横座標の指令位置
\$AA_MEAS_SETPOINT[軸]	縦座標の指令位置
\$AA_MEAS_SETPOINT[軸]	垂直座標の指令位置
\$AC_MEAS_ACT_PLANE	特に指定されていない限り、動作中の平面に従って計算 *
\$AC_MEAS_FRAME_SELECT	特に指定されていない限り、追加フレームに従って計算 *

9.5 原点の設定、ワーク測定、および工具測定

入力変数	意味
\$AC_MEAS_FINE_TRANS	特に指定されていない限り、フレームは汎用の平行移動に書き込まれます*
\$AC_MEAS_TYPE	21

\* オプションです

次の出力変数は、計測タイプ 21 のために書き込まれます。

出力変数	意味
\$AC_MEAS_FRAME	座標回転と平行移動のフレーム結果

例

3次元の指令値選択

プログラムコード	コメント
DEF INT RETVAL	
DEF REAL _CORMW_XX,	
_CORMW_YY,	
_CORMW_ZZ	
DEF AXIS _XX, _YY, _ZZ	
T1 D1	; プローブを有効化
G54	; すべてのフレームと G54 の有効化
\$AC_MEAS_VALID = 0	; すべての入力値を無効に設定
\$AC_MEAS_TYPE = 21	; 3次元指令値選択の計測タイプを設定
\$AC_MEAS_ACT_PLANE = 0	; 計測面は G17
_XX=\$P_AXN1	; 平面に従って軸を定義
_YY=\$P_AXN2	
_ZZ=\$P_AXN3	
	; 計測値を割り当て
\$AA_MEAS_POINT1[_xx] = \$AA_MW[_xx]	; 計測値を横座標に割り当て
\$AA_MEAS_POINT1[_yy] = \$AA_MW[_yy]	; 計測値を縦座標に割り当て
\$AA_MEAS_POINT1[_zz] = \$AA_MW[_zz]	; 計測値を垂直座標に割り当て
\$AA_MEAS_SETPOINT[_xx] = 10	; 横座標、縦座標、および垂直座標の指令値を定義
\$AA_MEAS_SETPOINT[_yy] = 10	; 定義
\$AA_MEAS_SETPOINT[_zz] = 10	
\$AC_MEAS_FRAME_SELECT = 102	; 対象フレーム- G55 を選択
\$AA_MEAS_SETPOINT[_yy] = 10	
RETVAl = MEASURE()	; 計測計算の開始
if RETVAL <> 0	
setal(61000 + RETVAL)	
endif	

プログラムコード	コメント
\$P_UIFR[2] = \$AC_MEAS_FRAME	; 計測フレームをデータ管理に書き込み (G55)
G55 G0 AX[_xx]=10 AX[_yy]=10	; フレームを有効化して移動
m30	

### 9.5.3.12 任意の座標系での計測点の計測(計測タイプ 24)

#### 任意の座標系での計測点変換の計測方法(\$AC\_MEAS\_TYPE = 24)

この計測方法では座標変換によって、任意の座標系(WCS、BCS、MCS)の計測点を、新しい座標系を基準に変換できます。

新しい座標系は、目的のフレーム結合を指定すると、生成できます。

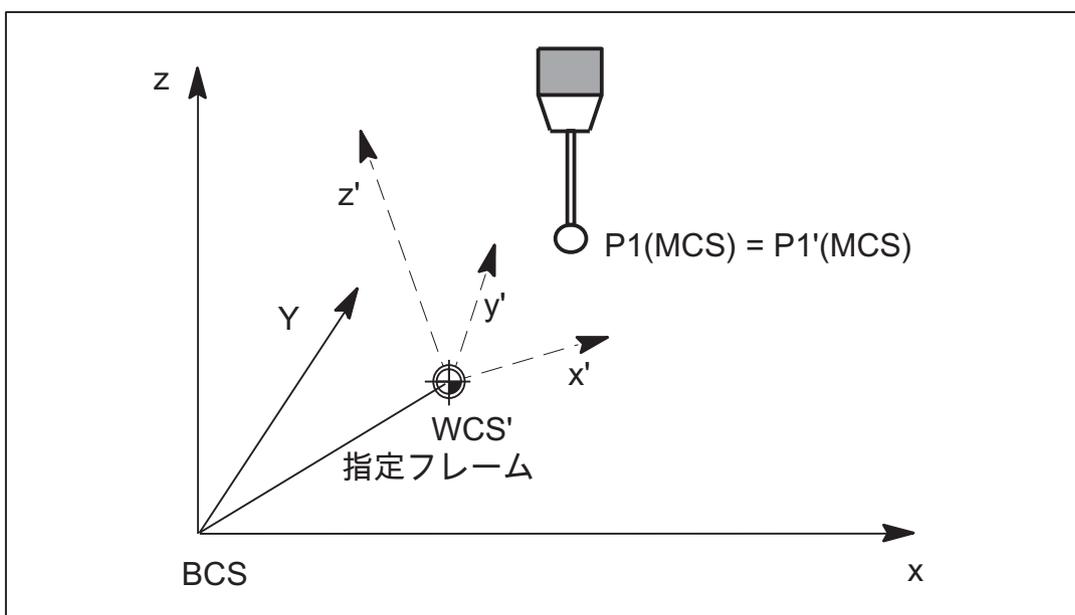


図 9-15 位置の座標変換

次の変数の値が計測タイプ 24 のために使用されます。

入力変数	意味
\$AC_MEAS_VALID	入力変数の有効ビット
\$AA_MEAS_POINT1[軸]	変換される位置
\$AC_MEAS_P1:COORD	初期設定は、0: WCS、1: BCS、2: MCS *
\$AC_MEAS_P2:COORD	対象座標系*
\$AC_MEAS_TOOL_MASK	0x20; 動作中の工具長は位置の座標変換に含まれます*
\$AC_MEAS_CHSFR	データ管理のシステムフレーム*

9.5 原点の設定、ワーク測定、および工具測定

入力変数	意味
\$AC_MEAS_NCBFR	データ管理のグローバル基本フレーム*
\$AC_MEAS_CHBFR	データ管理のチャンネル基本フレーム*
\$AC_MEAS_UIFR	データ管理の設定可能フレーム*
\$AC_MEAS_PFRAME	1: プログラマブルフレームは計算に含まれません*
\$AC_MEAS_TYPE	24

\* オプションです

次の出力変数は、計測タイプ 24 のために書き込まれます。

出力変数	意味
\$AC_MEAS_POINT2[軸]	変換後の軸位置

例

計測点の WCS 座標変換

プログラムコード	コメント
DEF INT RETVAL	
DEF INT LAUF	
DEF REAL_CORMW_xx, _CORMW_yy, _CORMW_zz	
DEF AXIS _XX, _YY, _ZZ	
\$TC_DP1[1,1]=120	; 工具タイプはエンドミル
\$TC_DP2[1,1]=20	
\$TC_DP3[1,1]=0	; (z)長補正ベクトル
\$TC_DP4[1,1]=0	; (y)長補正ベクトル
\$TC_DP5[1,1]=0	; (x)長補正ベクトル
\$TC_DP6[1,1]=2	; 半径
T1 D1	; プローブを有効化
G17	; 傾斜面は G17
_xx=\$P_AXN1 _yy=\$P_AXN2 _zz=\$P_AXN3	; 平面に従って軸を定義
	; すべてのフレームが結果的に CTRANS になります (_xx, 10, _yy, -1, _zz, 5, A, 6, B, 7)
\$P_CHBFR[0]=CTRANS(_zz, 5, A, 6) : CROT(_zz, 45)	
\$P_UIFR[1]=CTRANS( )	
\$P_UIFR[1, _xx, TR]=--SIN(45)	
\$P_UIFR[1, _yy, TR]=--SIN(45)	
\$P_UIFR[2]=CTRANS( )	

## 9.5 原点の設定、ワーク測定、および工具測定

プログラムコード	コメント
\$P_PFRAME=CROT(_zz,-45)	
\$P_CYCFR=CTRANS(_xx,10,B,7)	
G54	; すべてのフレームと G54 の有効化
G0 X0 Y0 Z0 A0 B0 F1000	
\$AC_MEAS_VALID = 0	; すべての入力値を無効に設定
\$AC_MEAS_TYPE = 24	; 座標変換の計測タイプを設定
\$AC_MEAS_ACT_PLANE = 0	; 計測面は G17
	; 計測値を割り当て
\$AA_MEAS_POINT1[_xx] = \$AA_IW[_xx]	; 計測値を横座標に割り当て
\$AA_MEAS_POINT1[_yy] = \$AA_IW[_yy]	; 計測値を縦座標に割り当て
\$AA_MEAS_POINT1[_zz] = \$AA_IW[_zz]	; 計測値を垂直座標に割り当て
\$AA_MEAS_POINT1[A] = \$AA_IW[A]	
\$AA_MEAS_POINT1[B] = \$AA_IW[B]	
\$AC_MEAS_P1_COORD=0	; WCS の位置を WCS' に変換
\$AC_MEAS_P2_COORD=0	
	; <b>WCS を設定</b>
	; すべてのフレームが結果的に CTRANS になります (_xx, 0,_yy,0,_zz,5,A,6,B,0)
	; サイクルのフレームを停止
\$AC_MEAS_CHSER=\$MC_MM_SYSTEM_FRAME_MASK B_AND 'B1011111'	
\$AC_MEAS__NCBFR='B0'	; グローバル基本フレームを停止
\$AC_MEAS__CHBFR='B1'	; データ管理のチャネル基本フレーム 1
\$AC_MEAS__UIFR=2	; データ管理の設定可能フレーム G55*
\$AA_MEAS_PFRAME=1	; プログラマブルフレームは計算に含めないでください
RETVAL = MEASURE()	; 計測計算の開始
if RETVAL <> 0	
setal(61000 + RETVAL)	
endif	
if \$AA_MEAS_PIONT2[_xx] <> 10	
setal(61000)	
M0	
stopre	
endif	
if \$AA_MEAS_PIONT2[_yy] <> -1	
setal(61000)	
M0	
stopre	
if \$AA_MEAS_PIONT2[_zz] <> 0	
setal(61000)	

9.5 原点の設定、ワーク測定、および工具測定

プログラムコード	コメント
M0	
stopre	
if \$AA_MEAS_PIONT2[A] <> 0	
setal(61000)	
M0	
stopre	
if \$AA_MEAS_PIONT2[B] <> 7	
setal(61000)	
M0	
stopre	
m30	

9.5.3.13 長方形の計測(計測タイプ 25)

長方形を決定するための計測点(\$AC\_MEAS\_TYPE = 25)

長方形を決定するには、次の作業平面で工具寸法が必要です。

- G17 作業平面 x/y 切り込み方向 z
- G18 作業平面 z/x 切り込み方向 y
- G19 作業平面 y/z 切り込み方向 x

長方形毎に 4 つの計測点が必要です。

計測点は、どの順序でも指定可能です。長い方の縦座標距離計測点が、位置 P3 と P4 に対応します。

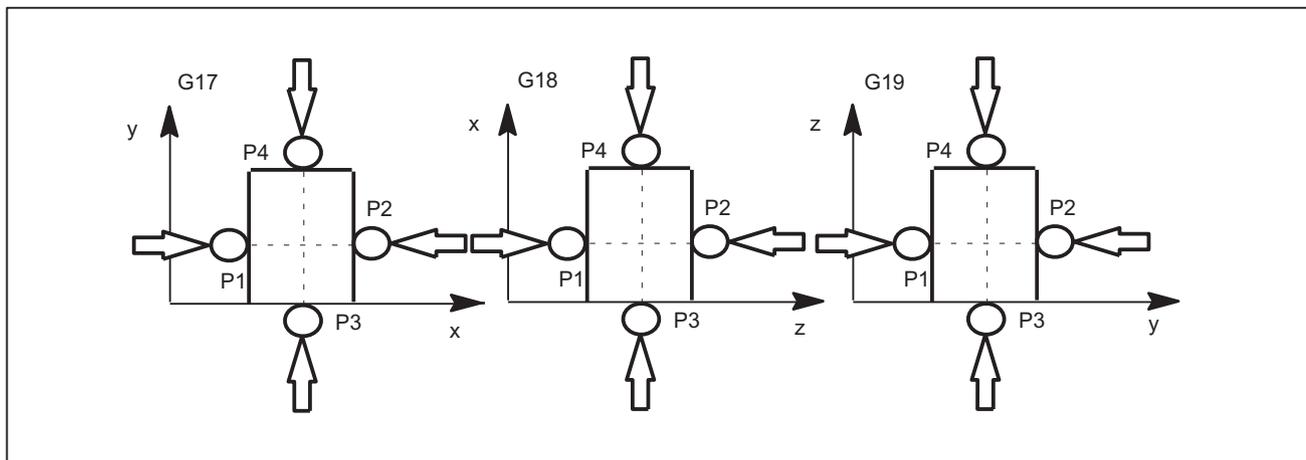


図 9-16 作業平面 G17、G18、および G19 で切り込みをもつ長方形の計測

次の変数の値が計測タイプ 25 のために使用されます。

入力変数	意味
\$AC_MEAS_VALID	入力変数の有効ビット
\$AA_MEAS_POINT1[軸]	計測点 1
\$AA_MEAS_POINT2[軸]	計測点 2
\$AA_MEAS_POINT3[軸]	計測点 3
\$AA_MEAS_POINT4[軸]	計測点 4
\$AA_MEAS_SETPOINT[軸]	ウェブ中心の指令位置*
\$AC_MEAS_ACT_PLANE	特に指定されていない限り、動作中の平面に従って計算 *
\$AC_MEAS_FINE_TRANS	0: 汎用オフセット、1: 精密オフセット*
\$AC_MEAS_FRAME_SELECT	特に指定されていない限り、追加フレームに従って計算 *
\$AC_MEAS_T_NUMBER	特に指定されていない限り、動作中の T に従って計算 (T0) *
\$AC_MEAS_D_NUMBER	特に指定されていない限り、動作中の D に従って計算 (D0) *
\$AC_MEAS_INPUT[0]	外部コーナーの指定なし* =0: 外部コーナーの計測 =1: 内部コーナーの計測
\$AC_MEAS_TYPE	25

\* オプションです

次の出力変数は、計測タイプ 25 のために書き込まれます。

出力変数	意味
\$AC_MEAS_FRAME	平行移動のあるフレーム結果
\$AC_MEAS_RESULTS[0]	横座標計算中心点
\$AC_MEAS_RESULTS[1]	縦座標計算中心点
\$AC_MEAS_RESULTS[2]	垂直座標計算中心点(輪郭に垂直な軸の座標)
\$AC_MEAS_RESULTS[3]	長方形の幅 P1/P2
\$AC_MEAS_RESULTS[4]	長方形の長さ P3/P4

## 9.5.3.14 データ管理フレームを保存するための計測(計測タイプ 26)

## データ管理フレームの保存(\$AC\_MEAS\_TYPE = 26)

この計測タイプでは、フレームの現在の割り当て値を含むデータ管理フレームの全部または一部をファイルに保存することができます。計測は、命令の実行によって、またはパートプログラムによって開始できます。この機能は、別のチャンネルから有効化することもできます。ファイルはディレクトリ `_N_SYF_DIR` に設定されます。

復元操作をおこなうと、バックアップデータが削除されます。また、新規に保存操作をおこなうと、既存のバックアップデータに上書きされます。さらに、最後に保存されたデータは、

- \$AC\_MEAS\_CHSFR = 0 システムフレーム;
- \$AC\_MEAS\_NCBFR = 0 グローバル基本フレーム;
- \$AC\_MEAS\_CHBFR = 0 チャンネル基本フレーム;
- \$AC\_MEAS\_UIFR = 0 設定可能フレームの数

2番目の保存操作を使って、データ管理システムからの上記の指令でも削除できます。

## 注記

すべてのデータ管理フレームのバックアップを作成する場合は、保存するフレーム毎に 1 KB のメモリが必要なため、注意してください。使用可能なメモリが不足していると、エラーメッセージ `MEAS_NO_MEMORY` で処理が中止されます。スタティックメモリのサイズを変更するために、次のマシンデータ項目を使用できます。

MD18351 \$MM\_DRAM\_FILE\_MEM\_SIZE

次の変数の値が計測タイプ 26 のために使用されます。

入力変数	意味
\$AC_MEAS_VALID	入力変数の有効ビット
\$AC_MEAS_CHSFR	データ管理のシステムフレームのビットマスク。* この変数が書かれていない場合は、すべてのシステムフレームがバックアップされます。
\$AC_MEAS_NCBFR	データ管理のグローバル基本フレームのビットマスク。* この変数が書かれていない場合は、すべてのグローバル基本フレームがバックアップされます。

入力変数	意味
\$AC_MEAS_CHBFR	データ管理のチャンネル基本フレームのビットマスク。 * この変数が書かれていない場合は、すべてのチャンネル基本フレームがバックアップされます。
\$AC_MEAS_UIFR	データ管理の設定可能フレームの数。 * 0..100: 1: G500 2: G500、G54 この変数が書かれていない場合は、すべての設定可能フレームがバックアップされます。
\$AC_MEAS_TYPE	26

\* オプションです

### 9.5.3.15 バックアップされたデータ管理フレームを復元するための計測(計測タイプ 27)

#### 最後にバックアップされたデータ管理フレームの復元(\$AC\_MEAS\_TYPE = 27)

この計測タイプでは、計測タイプ 26 でバックアップされたデータ管理フレームを SRAM に復元することができます。

最後にバックアップされたこれらのフレームの一部または全部を復元できます。バックアップされていないフレームを選択すると、その選択が無視されます。処理は中止されません。

次の変数の値が計測タイプ 27 のために使用されます。

入力変数	意味
\$AC_MEAS_VALID	入力変数の有効ビット
\$AC_MEAS_CHSFR	データ管理のシステムフレームのビットマスク。 * この変数が書かれていない場合は、すべてのシステムフレームが復元されます。
\$AC_MEAS_NCBFR	データ管理のグローバル基本フレームのビットマスク。 * この変数が書かれていない場合は、すべてのグローバル基本フレームが復元されます。

入力変数	意味
\$AC_MEAS_CHBFR	データ管理のチャンネル基本フレームのビットマスク。* この変数が書かれていない場合は、すべてのチャンネル基本フレームが復元されます。
\$AC_MEAS_UIFR	データ管理の設定可能フレームの数。* 範囲: 1: G54 ~ G99: G599。 この変数が書かれていない場合は、すべての設定可能フレームが復元されます。
\$AC_MEAS_TYPE	27

\* オプションです

### 9.5.3.16 テーパー旋削の追加回転を定義するための計測(計測タイプ 28)

#### テーパー旋削のための平面追加回転(\$AC\_MEAS\_TYPE = 28)

計測タイプ 28 を使用して、有効な平面または特定の平面の追加回転を  $\alpha = +/- 90^\circ$  の範囲の角度で指定できます。回転は、平面に垂直な座標軸上で発生します。

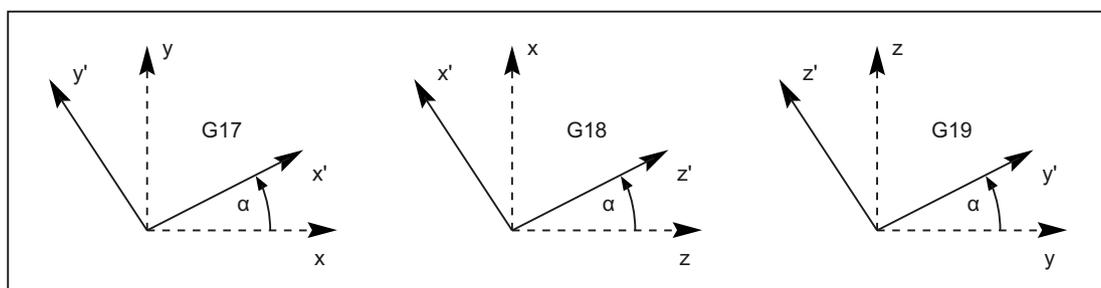


図 9-17 平面 G17、G18、および G19 の角度  $\alpha = +30^\circ$  の回転

#### 用途

テーパー旋削では、動作中の平面がテーパー角度で回転し、それにより、この座標回転が動作中のサイクルのフレームに書き込まれます。RESET を使用すると、サイクルのフレームは削除されます。再度有効にしなければならない場合があります。サイクルのフレームは、SZS 位置表示に応じて選択されます。動作中の平面 G18 などの座標回転を起動した後で、z' の方向に移動がおこなわれた場合は、対応する軸の実位置が、同時に x と z について変更されます。

動作中の平面 G17 と G19 の座標回転は同様に、上の図に示すように動作します。次の変数の値が計測タイプ 28 のために使用されます。

入力変数	意味
\$AC_MEAS_VALID	入力変数の有効ビット
\$AC_MEAS_WP_SETANGLE	指令角度
\$AC_MEAS_ACT_PLANE	特に指定されていない限り、動作中の平面に従って回転*
\$AC_MEAS_FRAME_SELECT	特に指定されていない限り、追加フレームに従って計算*
\$AC_MEAS_INPUT[0]	1: テーパー旋削が有効*
\$AC_MEAS_TYPE	28

\* オプションです

次の出力変数は、計測タイプ 28 のために書き込まれます。

出力変数	意味
\$AC_MEAS_FRAME	回転結果

## 9.5.4 工具測定

制御装置は、工具先端と工具キャリヤレファレンス点 T との間の距離を、ユーザが指定した工具長から計算します。

次の測定タイプは、ターニングセンタまたはフライス盤に搭載された工具の測定に使用できます。

測定タイプ;ソクテイタイプ	工具測定
\$AC_MEAS_TYPE = 10	すでに測定済みの基準部品上の工具長
\$AC_MEAS_TYPE = 11	すでに測定済みの基準部品上の工具直径

## 9.5 原点の設定、ワーク測定、および工具測定

測定タイプ;ソクテイタイプ	工具測定
\$AC_MEAS_TYPE = 22	ズームイン機能搭載の機械の工具直径(ShopTurn)
\$AC_MEAS_TYPE = 23	<p>保存された位置または現在の位置をもつ工具長(ShopTurn)</p> <p>次の向きをもつ 2 種類の工具長の測定:</p> <p>下記を備えた 2 つの旋盤工具:            工具の向きがアプローチ方向の工具の、それぞれの工具のレファレンス点。            アプローチ方向および工具の向きが反対側の工具位置の、1 つのレファレンス点。</p> <p>下記を備えた 2 つのフライス工具:            工具の向きが-y の工具のそれぞれのレファレンス点。            工具の向きが-y およびアプローチ方向と反対側の工具位置の 1 つのレファレンス点。</p> <p>下記を備え、90°回転した 2 つのフライス工具:            工具の向きがアプローチ方向の工具の、それぞれのレファレンス点。            アプローチ方向および工具の向きと反対側の工具位置の、1 つのレファレンス点。</p>

## 9.5.5 タイプ s of workpiece measurement

## 9.5.5.1 工具長の計測(計測タイプ 10)

## 計測済みの基準部品上の工具長の計測(\$AC\_MEAS\_TYPE = 10)

工具長は、事前に計測された基準部品上で計測できます。

平面選択は、工具の位置によって決まります。

- z 方向の工具位置の G17
- y 方向の工具位置の G18
- x 方向の工具位置の G19

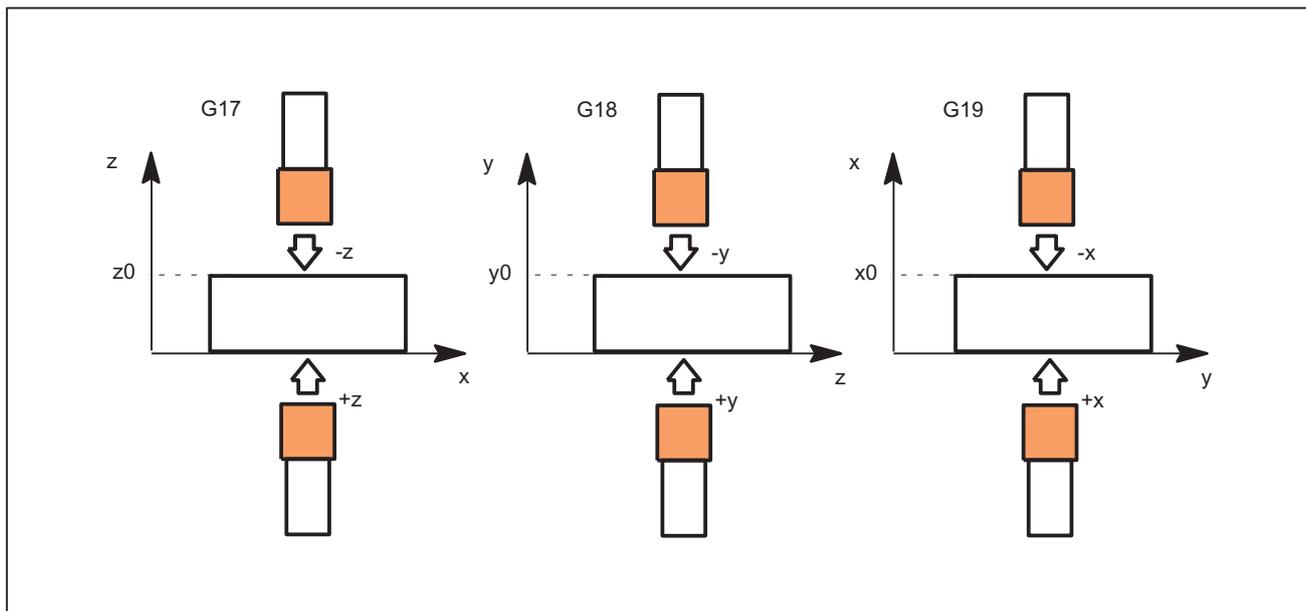


図 9-18 選択した G17、G18、および G19 の工具長の計測

次の変数の値が計測タイプ 10 のために使用されます。

入力変数	意味
\$AC_MEAS_VALID	入力変数の有効ビット
\$AA_MEAS_POINT1[軸]	計測点 1
\$AC_MEAS_P1_COORD	計測点の座標系*
\$AA_MEAS_SETPOINT[軸]	位置 z0 を設定
\$AC_MEAS_SET_COORD	指令値の座標系*
\$AC_MEAS_DIR_APPROACH	0: +x, 1: -x, 2: +y, 3: -y, 4: +z, 5: -z
\$AC_MEAS_ACT_PLANE	特に指定されていない限り、動作中の平面に従って計算*
\$AC_MEAS_TYPE	10

\* オプションです

9.5 原点の設定、ワーク測定、および工具測定

次の出力変数は、計測タイプ 10 のために書き込まれます。

出力変数	意味
\$AC_MEAS_TOOL_LENGTH	工具長
\$AC_MEAS_RESULTS[0]	x の工具長
\$AC_MEAS_RESULTS[1]	y の工具長
\$AC_MEAS_RESULTS[2]	z の工具長
\$AC_MEAS_RESULTS[3]	工具長 L1
\$AC_MEAS_RESULTS[4]	工具長 L2
\$AC_MEAS_RESULTS[5]	工具長 L3

例

工具長の計測

プログラムコード	コメント
DEF INT RETVAL	
T0 D0	
g0 x0 y0 z0 f10000	
	; 工具長計測
\$AC_MEAS_VALID = 0	; すべての入力値を無効に設定
g1 z10	; 工具を基準部品へと移動
\$AC_MEAS_LATCH[0] = 1	; 計測点 1 を取得
\$AC_MEAS_DIR_APPROACH = 5	; アプローチ方向-zを設定
\$AA_MEAS_SETPOINT[x] = 0	; 基準位置を設定
\$AA_MEAS_SETPOINT[y] = 0	
\$AA_MEAS_SETPOINT[z] = 0	
\$AC_MEAS_ACT_PLANE = 0	; 計測面は G17
\$AC_MEAS_T_NUMBER = 0	; 工具は未選択
\$AC_MEAS_D_NUMBER = 0	
\$AC_MEAS_TYPE = 10	; 工具長に計測タイプを設定
RETVAL = MEASURE()	; 計測処理を開始
if RETVAL <> 0 setal(61000 + RETVAL)	
endif	
if \$AC_MEAS_TOOL_LENGTH <> 10	; 既知の工具長を確認します
setal(61000 + \$AC_MEAS_TOOL_LENGTH)	
endif	
m30	

## 9.5.5.2 工具直径の測定(測定タイプ 11)

## 基準部品上の工具直径の測定(\$AC\_MEAS\_TYPE = 11)

工具直径は、測定済みの基準部品上で測定できます。工具位置に応じて、z 方向の工具位置の平面 G17、y 方向の工具位置の平面 G18、および x 方向の工具位置の平面 G19 を選択できます。

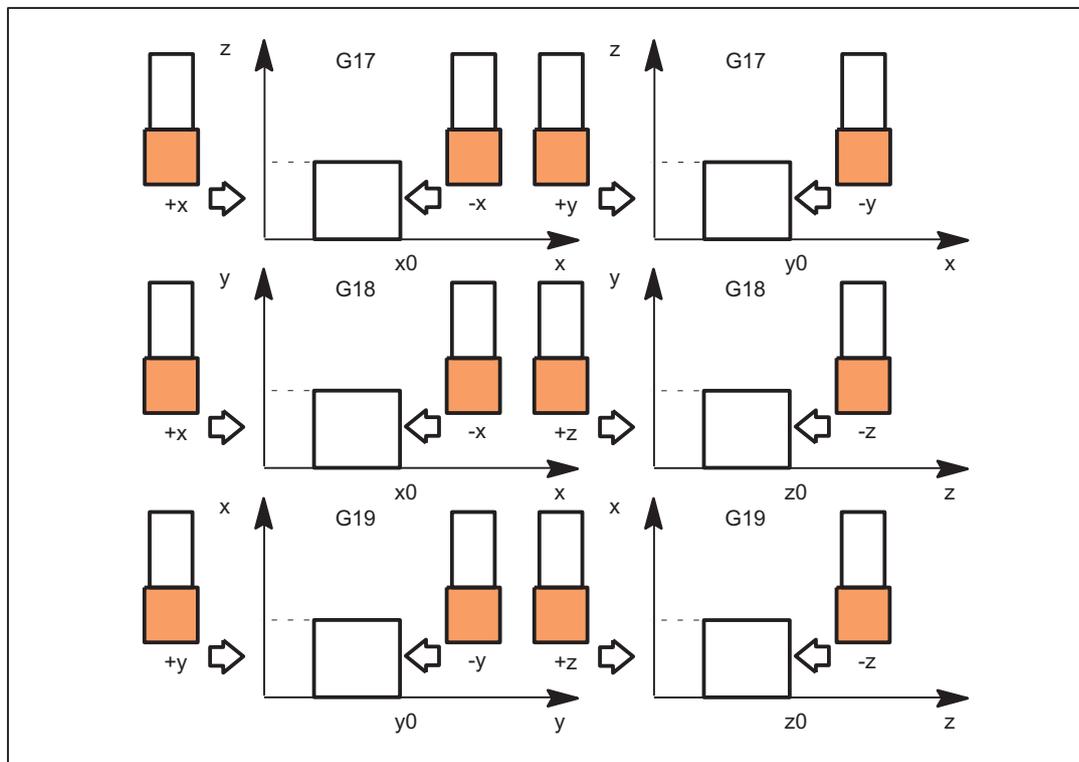


図 9-19 選択平面 G17、G18、および G19 の工具直径

次の変数の値が測定タイプ 11 のために使用されます。

入力変数	意味
\$AC_MEAS_VALID	入力変数の有効ビット
\$AA_MEAS_POINT1[軸]	測定点 1
\$AA_MEAS_SETPPOINT[軸]	位置 x0 を設定
\$AC_MEAS_DIR_APPROACH H	0: +x、1: -x、2: +y、3: -y、4: +z、5: -z

## 9.5 原点の設定、ワーク測定、および工具測定

入力変数	意味
\$AC_MEAS_ACT_PLANE	特に指定されていない限り、動作中の平面に従って計算*
\$AC_MEAS_TYPE	11

\* オプションです

次の出力変数は、測定タイプ 11 のために書き込まれます。

出力変数	意味
\$AC_MEAS_TOOL_DIAMETE R	工具直径

## 9.5.5.3 ズームイン機能による工具長測定(測定タイプ 22)

## ズームイン機能による工具長

## ズームイン機能による工具長測定(\$AC\_MEAS\_TYPE = 22)

機械でズームイン機能を使用できる場合は、この機能を使って工具寸法を決定できます。

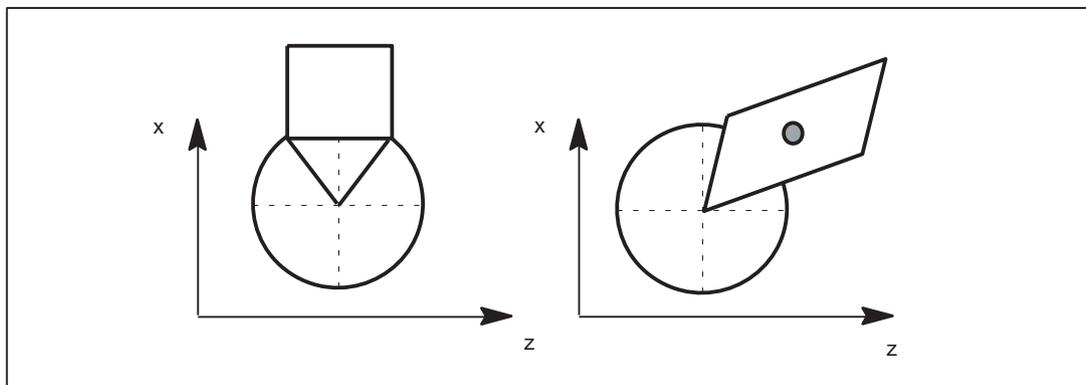


図 9-20 ズームイン機能による工具長の測定

次の変数の値が測定タイプ 22 のために使用されます。

入力変数	説明
\$AC_MEAS_VALID	入力変数の有効ビット
\$AA_MEAS_POINT1[軸]	すべてのチャネル軸の測定点 1
\$AC_MEAS_P1_COORD	測定点の座標系*

入力変数	説明
\$AA_MEAS_SETPOINT[軸]	ズーム位置 x および z を指定
\$AC_MEAS_SET_COORD	指令値の座標系*
\$AC_MEAS_ACT_PLANE	特に指定されていない限り、動作中の平面に従って計算 *
\$AC_MEAS_FRAME_SELECT	特に指定されていない限り、追加フレームに従って計算 *
\$AC_MEAS_T_NUMBER	特に指定されていない限り、動作中の T に従って計算 (T0) *
\$AC_MEAS_D_NUMBER	特に指定されていない限り、動作中の D に従って計算 (D0) *
\$AC_MEAS_TYPE	22

\* オプションです

次の出力変数は、測定タイプ 22 のために書き込まれます。

出力変数	説明
\$AC_MEAS_RESULT[0]	x の工具長
\$AC_MEAS_RESULT[1]	y の工具長
\$AC_MEAS_RESULT[2]	z の工具長
\$AC_MEAS_RESULT[3]	工具長 L1
\$AC_MEAS_RESULT[4]	工具長 L2
\$AC_MEAS_RESULT[5]	工具長 L3

#### 9.5.5.4 保存済みの位置または現在の位置をもつ工具長の測定(測定タイプ 23)

##### 保存済みの位置/実位置をもつ工具長

##### 保存済みの位置または実位置をもつ工具長の測定(\$AC\_MEAS\_TYPE = 23)

手動測定の場合は、工具の寸法を X および Z の方向で測定できます。次の既知の位置を使用します。

- 工具キャリヤレファレンス点
- ワーク寸法

上記から ShopTurn が工具オフセットデータを計算します。

9.5 原点の設定、ワーク測定、および工具測定

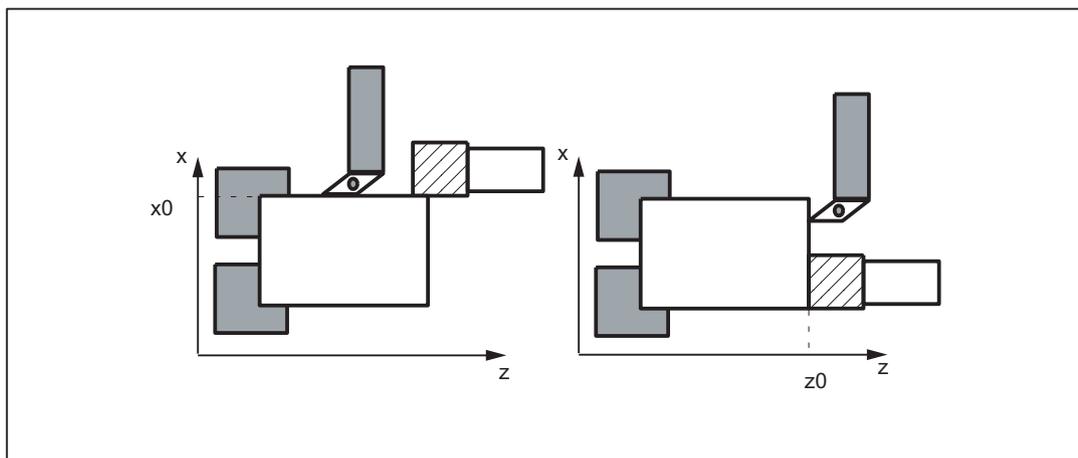


図 9-21 保存済み位置または実位置をもつ工具長の測定

次の入力変数の値が測定タイプ 23 のために使用されます。

入力変数	説明
\$AC_MEAS_VALID	入力変数の有効ビット
\$AA_MEAS_POINT1[軸]	現在の位置またはマーキングされた位置
\$AC_MEAS_P1_COORD	測定点の座標系*
\$AA_MEAS_SETPOINT[軸]	指令位置(1 軸以上のジオメトリ軸を指定してください)
\$AC_MEAS_SET_COORD	指令値の座標系*
\$AC_MEAS_ACT_PLANE	特に指定されていない限り、動作中の平面に従って計算 *
\$AC_MEAS_T_NUMBER	特に指定されていない限り、動作中の T に従って計算 (T0) *
\$AC_MEAS_D_NUMBER	特に指定されていない限り、動作中の D に従って計算 (D0) *
\$AC_MEAS_TOOL_MASK	工具位置、半径*
\$AC_MEAS_DIR_APPROACH	アプローチ方向*
\$AC_MEAS_INPUT[0] = 1	計算工具長をデータ管理に書き込み*
\$AC_MEAS_TYPE	23

\* オプションです

次の出力変数は、測定タイプ 23 のために書き込まれます。

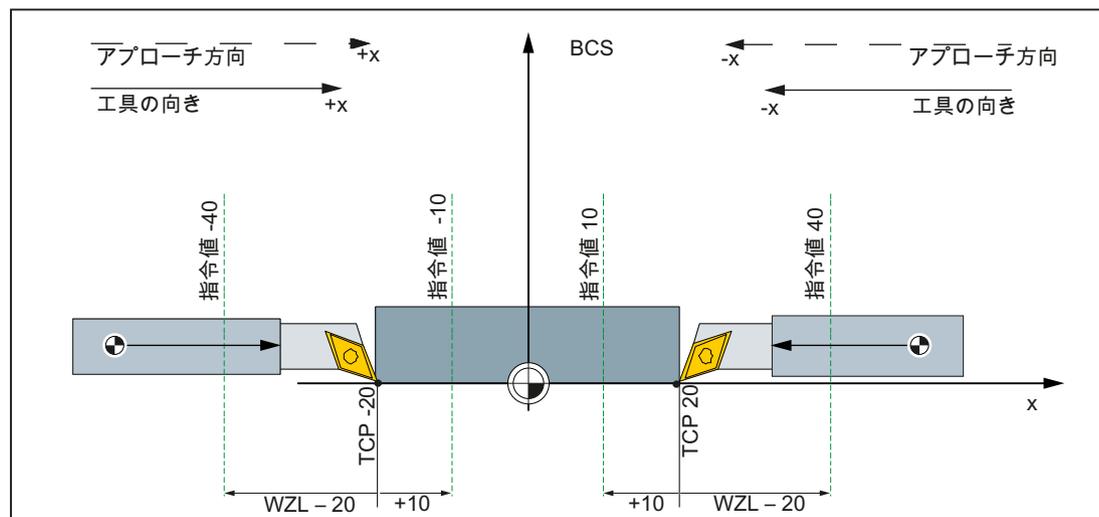
出力変数	説明
\$AC_MEAS_RESULT[0]	x の工具長
\$AC_MEAS_RESULT[1]	y の工具長
\$AC_MEAS_RESULT[2]	z の工具長
\$AC_MEAS_RESULT[3]	工具長 L1
\$AC_MEAS_RESULT[4]	工具長 L2
\$AC_MEAS_RESULT[5]	工具長 L3

### 9.5.5.5 次の向きをもつ 2 種類の工具長の測定:

#### 工具の向き

工具に対して、工具ホルダの向きをシステム変数\$AC\_MEAS\_TOOL\_MASK、ビット 9=1(0x200)を設定してください。これにより、計算工具長が負の値として含まれます。

アプローチ方向の工具の向きで、レファレンス点をそれぞれに持つ 2 つの旋盤工具



9.5 原点の設定、ワーク測定、および工具測定

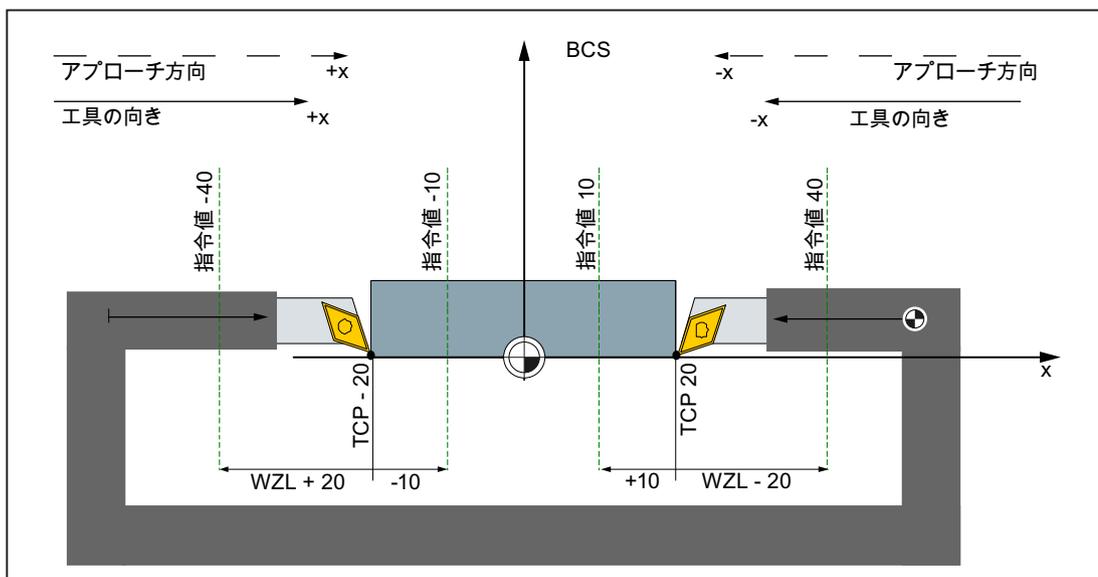
システムデータの設定:

左片刃工具: アプローチ方向と工具の向き+x	
システム変数	意味
\$AC_MEAS_TOOL_MASK = 0x2	x 方向の工具位置(G19)。
\$AC_MEAS_DIR_APPROACH = 0	アプローチ方向+x

右片刃工具: アプローチ方向と工具の向き-x	
システム変数	意味
\$AC_MEAS_TOOL_MASK = 0x40	-x 方向の工具位置。
\$AC_MEAS_DIR_APPROACH = 1	アプローチ方向-x

両工具について	
システム変数	意味
\$AC_MEAS_Px_COORD = 1	x 番目の計測点の座標系 = BCS
\$AC_MEAS_SET_COORD = 1	指令値の座標系 = BCS

アプローチ方向の工具の対向方向に、レファレンス点をそれぞれに持つ2つの旋盤工具

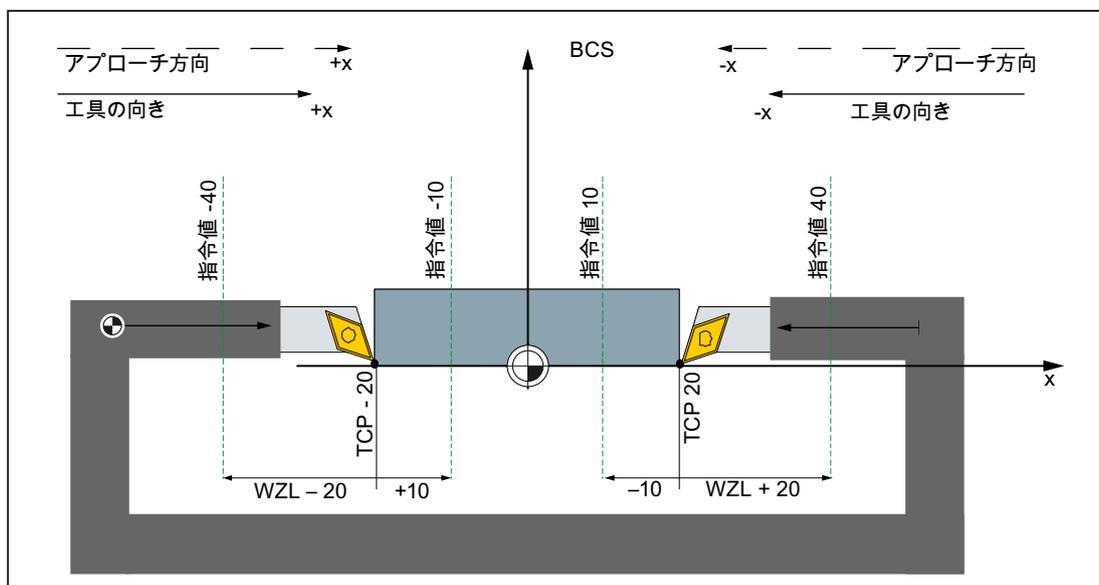


システムデータの設定:

左片刃工具:アプローチ方向と工具の向き+x	
システム変数	意味
$\$AC\_MEAS\_TOOL\_MASK = 0x2 + 0x200$	x 方向の工具位置(G19) + 工具長の差分値に負の値が含まれます。
$\$AC\_MEAS\_DIR\_APPROACH = 0$	アプローチ方向+x

右片刃工具:アプローチ方向と工具の向き-x	
システム変数	意味
$\$AC\_MEAS\_TOOL\_MASK = 0x40$	-x 方向の工具位置。
$\$AC\_MEAS\_DIR\_APPROACH = 1$	アプローチ方向-x

両工具について	
システム変数	意味
$\$AC\_MEAS\_Px\_COORD = 1$	x 番目の計測点の座標系 = BCS
$\$AC\_MEAS\_SET\_COORD = 1$	指令値の座標系 = BCS



9.5 原点の設定、ワーク測定、および工具測定

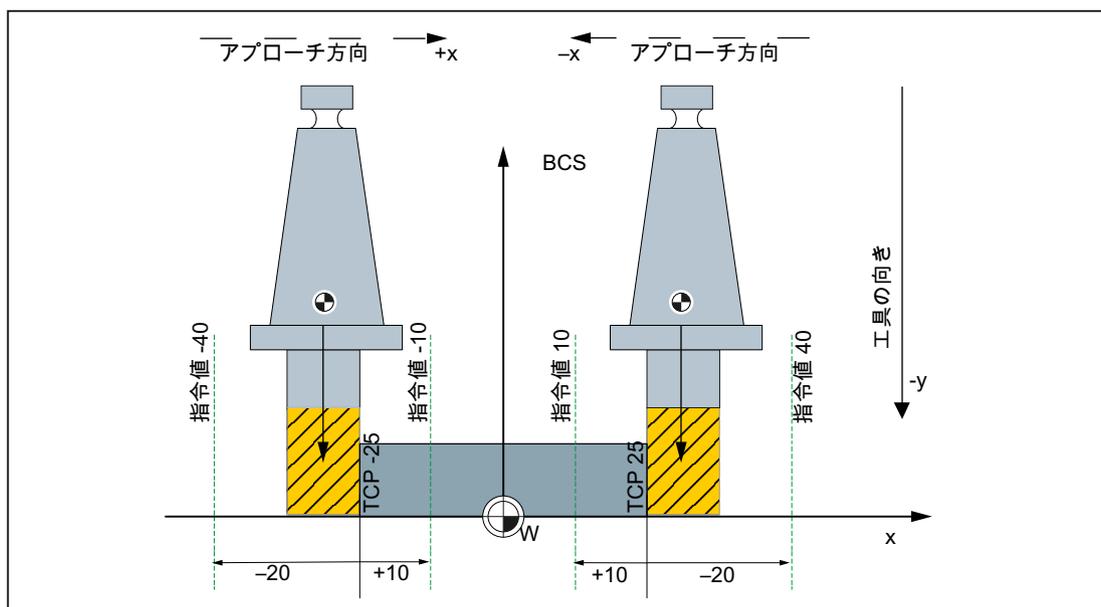
システムデータの設定:

左片刃工具:アプローチ方向と工具の向き+x	
システム変数	意味
\$AC_MEAS_TOOL_MASK = 0x2	x 方向の工具位置(G19)。
\$AC_MEAS_DIR_APPROACH = 0	アプローチ方向+x

右片刃工具:アプローチ方向と工具の向き-x	
\$AC_MEAS_TOOL_MASK = 0x40 + 0x200	x 方向の工具位置 + 工具長、差分値に負の値が含まれます。
\$AC_MEAS_DIR_APPROACH = 1	アプローチ方向-x

両工具について	
\$AC_MEAS_Px_COORD = 1	x 番目の計測点の座標系 = BCS
\$AC_MEAS_SET_COORD = 1	指令値の座標系 = BCS

それぞれが自身のレファレンス点を持ち、工具の向きがアプローチ方向に直交する 2 つのフライス工具



システムデータの設定:

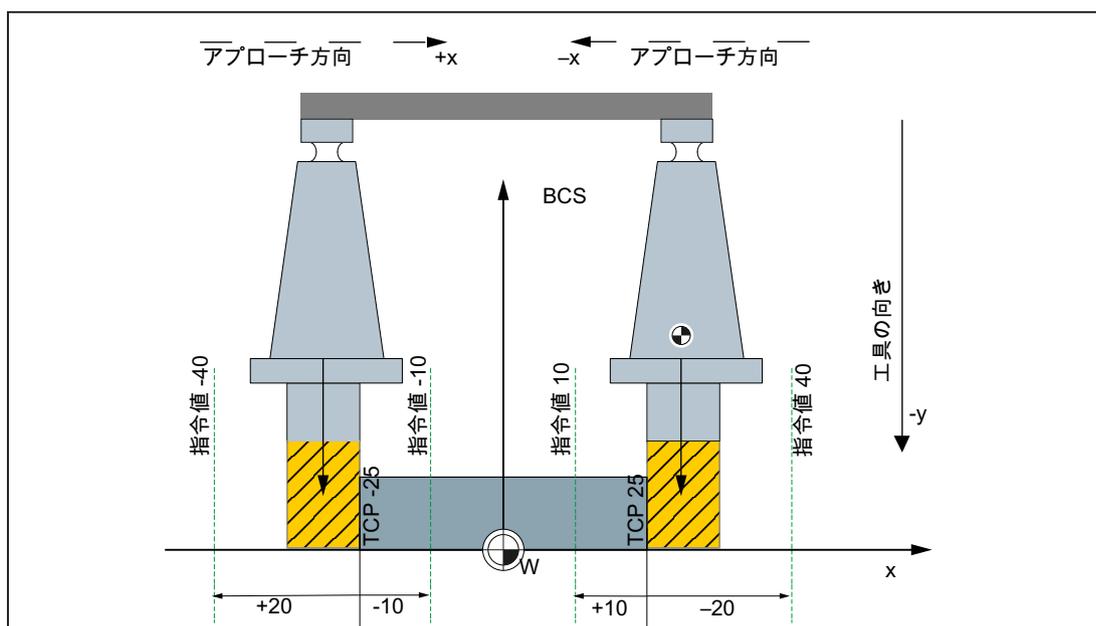
左側工具: アプローチ方向+x と工具の向き-y	
システム変数	意味
\$SAC_MEAS_TOOL_MASK = 0x80	-y 方向の工具位置
\$SAC_MEAS_DIR_APPROACH = 0	アプローチ方向+x

右側工具: アプローチ方向-x と工具の向き-y	
システム変数	意味
\$SAC_MEAS_TOOL_MASK = 0x80	-y 方向の工具位置
\$SAC_MEAS_DIR_APPROACH = 1	アプローチ方向-x

両工具について	
システム変数	意味
\$SAC_MEAS_Px_COORD = 1	x 番目の計測点の座標系 = BCS
\$SAC_MEAS_SET_COORD = 1	指令値の座標系 = BCS

それぞれが 1 つのレファレンス点を持ち、工具の向きがアプローチ方向に直交する 2 つのフライス工具

-y の工具の向きに 1 つのレファレンス点を持つ 2 つのフライス工具



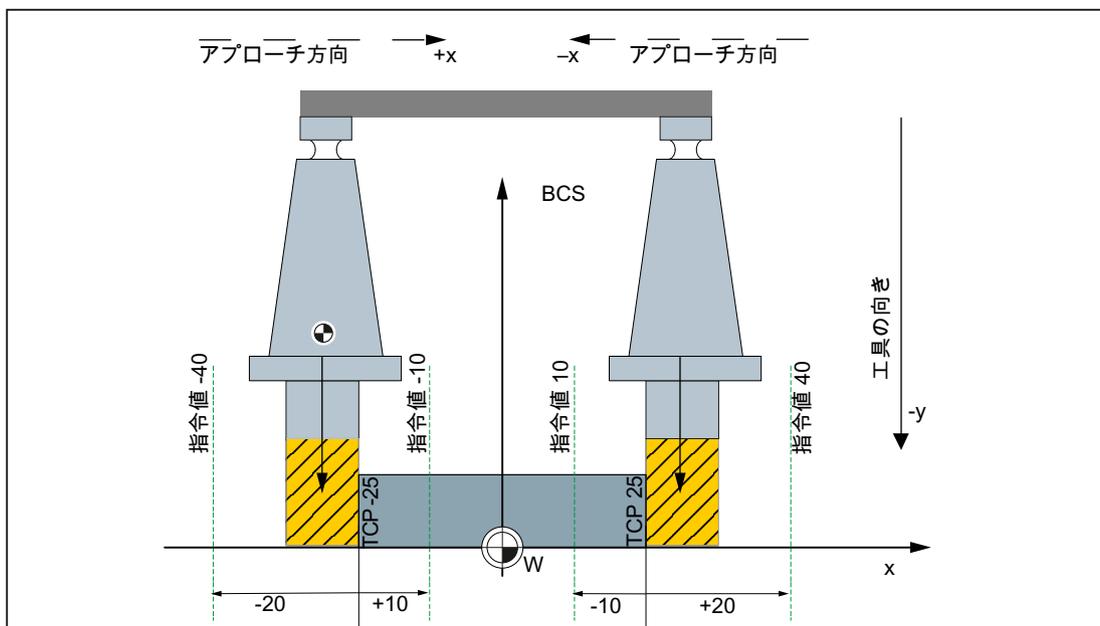
9.5 原点の設定、ワーク測定、および工具測定

現在のレイアウトで、工具位置\$AC\_MEAS\_TOOL\_MASK とワーク \$AC\_MEAS\_DIR\_APPROACH へのアプローチ方向は、次のように設定してください。

左側工具: アプローチ方向+x と工具の向き-y	
システム変数	意味
\$AC_MEAS_TOOL_MASK = 0x80 + 0x200	-y 方向の工具位置 + 工具長の差分値に負の値が含まれます。
\$AC_MEAS_DIR_APPROACH = 0	アプローチ方向+x

右側工具: アプローチ方向-x と工具の向き-y	
システム変数	意味
\$AC_MEAS_TOOL_MASK = 0x80	-y 方向の工具位置
\$AC_MEAS_DIR_APPROACH = 1	アプローチ方向-x

両工具について	
システム変数	意味
\$AC_MEAS_Px_COORD = 1	x 番目の計測点の座標系 = BCS
\$AC_MEAS_SET_COORD = 1	指令値の座標系 = BCS



現在のレイアウトで、工具位置\$AC\_MEAS\_TOOL\_MASK とワーク  
\$AC\_MEAS\_DIR\_APPROACH へのアプローチ方向は、次のように設定してください。

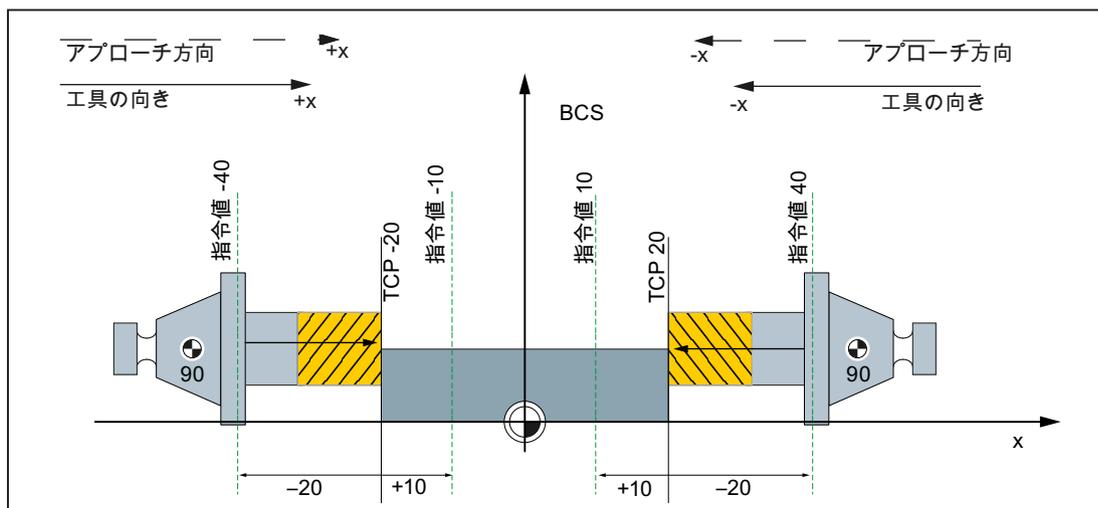
左側工具: アプローチ方向+x と工具の向き-y	
システム変数	意味
\$AC_MEAS_TOOL_MASK = 0x80	-y 方向の工具位置
\$AC_MEAS_DIR_APPROACH = 0	アプローチ方向+x

右側工具: アプローチ方向-x と工具の向き-y	
\$AC_MEAS_TOOL_MASK = 0x80 + 0x200	-y 方向の工具位置 + 工具長の差分値に負の 値が含まれます。
\$AC_MEAS_DIR_APPROACH = 1	アプローチ方向-x

両工具について	
\$AC_MEAS_Px_COORD = 1	x 番目の計測点の座標系 = BCS
\$AC_MEAS_SET_COORD = 1	指令値の座標系 = BCS

アプローチ方向の工具の対向方向にレファレンス点をそれぞれに持つ2つのフライス工具

アプローチ方向の工具の向きにレファレンス点をそれぞれに持つ2つのフライス工具



9.5 原点の設定、ワーク測定、および工具測定

現在のレイアウトで、工具位置\$AC\_MEAS\_TOOL\_MASK とワーク \$AC\_MEAS\_DIR\_APPROACH へのアプローチ方向は、次のように設定してください。

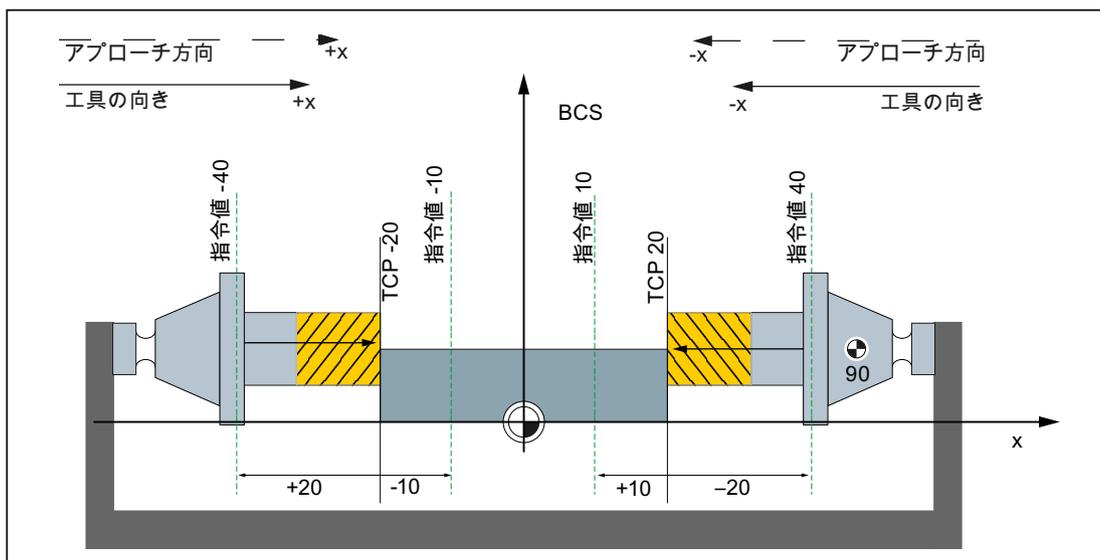
左側工具:アプローチ方向と工具の向き+x	
システム変数	意味
\$AC_MEAS_TOOL_MASK = 0x2	x 方向の工具位置(G19)。
\$AC_MEAS_DIR_APPROACH = 0	アプローチ方向+x

右側工具:アプローチ方向と工具の向き-x	
システム変数	意味
\$AC_MEAS_TOOL_MASK = 0x40	-x 方向の工具位置。
\$AC_MEAS_DIR_APPROACH = 1	アプローチ方向-x

両工具について	
システム変数	意味
\$AC_MEAS_Px_COORD = 1	x 番目の計測点の座標系 = BCS
\$AC_MEAS_SET_COORD = 1	指令値の座標系 = BCS

アプローチ方向の工具の対向方向にレファレンス点をそれぞれに持つ 2 つのフライス工具

工具向きと反対側の工具位置に 1 つのレファレンス点を持つ 2 つのフライス工具

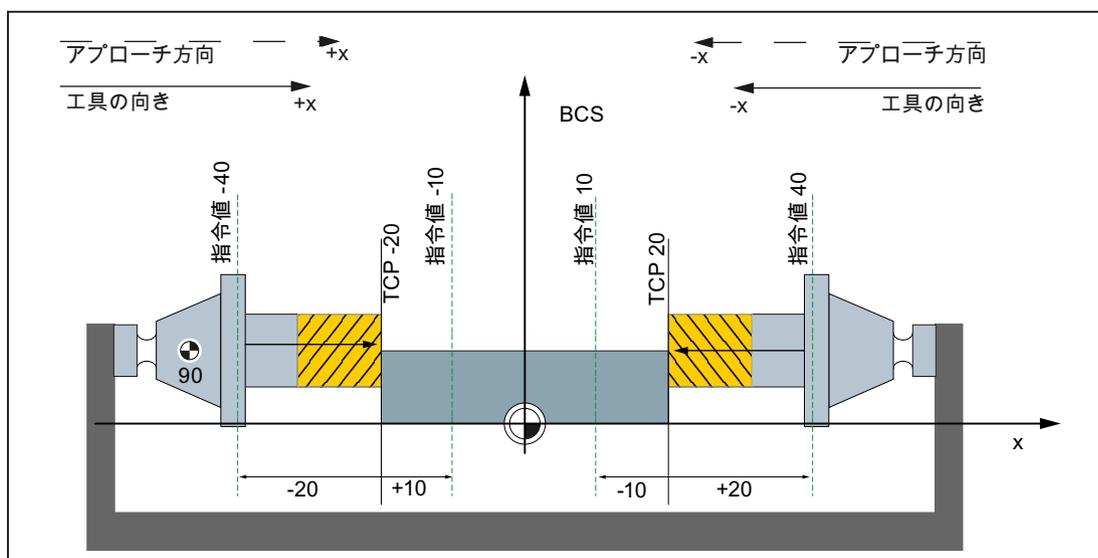


現在のレイアウトで、工具位置\$AC\_MEAS\_TOOL\_MASK とワーク  
\$AC\_MEAS\_DIR\_APPROACH へのアプローチ方向は、次のように設定してください。

左側工具:アプローチ方向と工具の向き+x	
システム変数	意味
\$AC_MEAS_TOOL_MASK = 0x2 + 0x200	x 方向の工具位置(G19) + 工具長の差分値に負の値が含まれます。
\$AC_MEAS_DIR_APPROACH = 0	アプローチ方向+x

右側工具:アプローチ方向と工具の向き-x	
\$AC_MEAS_TOOL_MASK = 0x40	-x 方向の工具位置。
\$AC_MEAS_DIR_APPROACH = 1	アプローチ方向-x

両工具について	
\$AC_MEAS_Px_COORD = 1	x 番目の計測点の座標系 = BCS
\$AC_MEAS_SET_COORD = 1	指令値の座標系 = BCS



9.5 原点の設定、ワーク測定、および工具測定

現在のレイアウトで、工具位置\$AC\_MEAS\_TOOL\_MASK とワーク \$AC\_MEAS\_DIR\_APPROACH へのアプローチ方向は、次のように設定してください。

左側工具:アプローチ方向と工具の向き+x	
システム変数	意味
\$AC_MEAS_TOOL_MASK = 0x2	x 方向の工具位置(G19)。
\$AC_MEAS_DIR_APPROACH = 0	アプローチ方向+x

右側工具:アプローチ方向と工具の向き-x	
\$AC_MEAS_TOOL_MASK = 0x40 + 0x200	-x 方向の工具位置 + 工具長の差分値に負の値が含まれます。
\$AC_MEAS_DIR_APPROACH = 1	アプローチ方向-x

両工具について	
\$AC_MEAS_Px_COORD = 1	x 番目の計測点の座標系 = BCS
\$AC_MEAS_SET_COORD = 1	指令値の座標系 = BCS

WCS のさまざまな工具

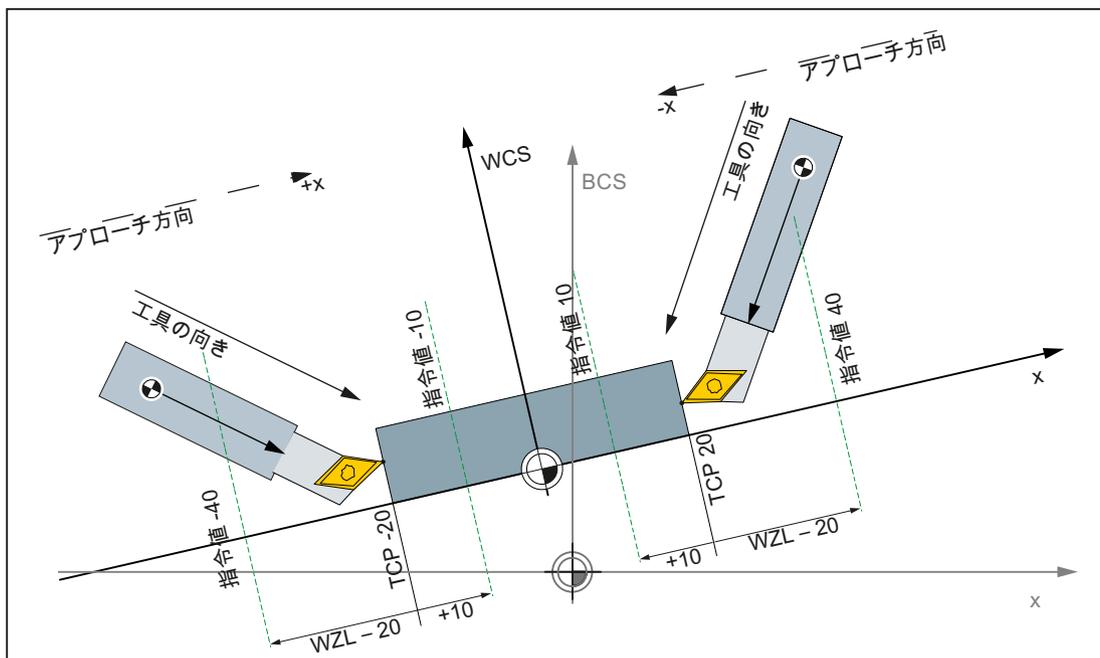


図 9-22 それぞれにリファレンス点を持つ 2 つの旋盤工具

システムデータの設定:

左片刃工具: アプローチ方向+x と工具の向き-y	
システム変数	意味
\$AC_MEAS_TOOL_MASK = 0x0	すべての工具長が考慮されます (初期設定)。
\$AC_MEAS_DIR_APPROACH = 0	アプローチ方向+x

右片刃工具: アプローチ方向-x と工具の向き-y	
\$AC_MEAS_TOOL_MASK = 0x0	すべての工具長が考慮されます (初期設定)。
\$AC_MEAS_DIR_APPROACH = 1	アプローチ方向-x

両工具について	
\$AC_MEAS_Px_COORD = 0	x 番目の計測点の座標系 = WCS (初期設定)
\$AC_MEAS_SET_COORD = 0	指令値の座標系 = WCS (初期設定)

9.5 原点の設定、ワーク測定、および工具測定

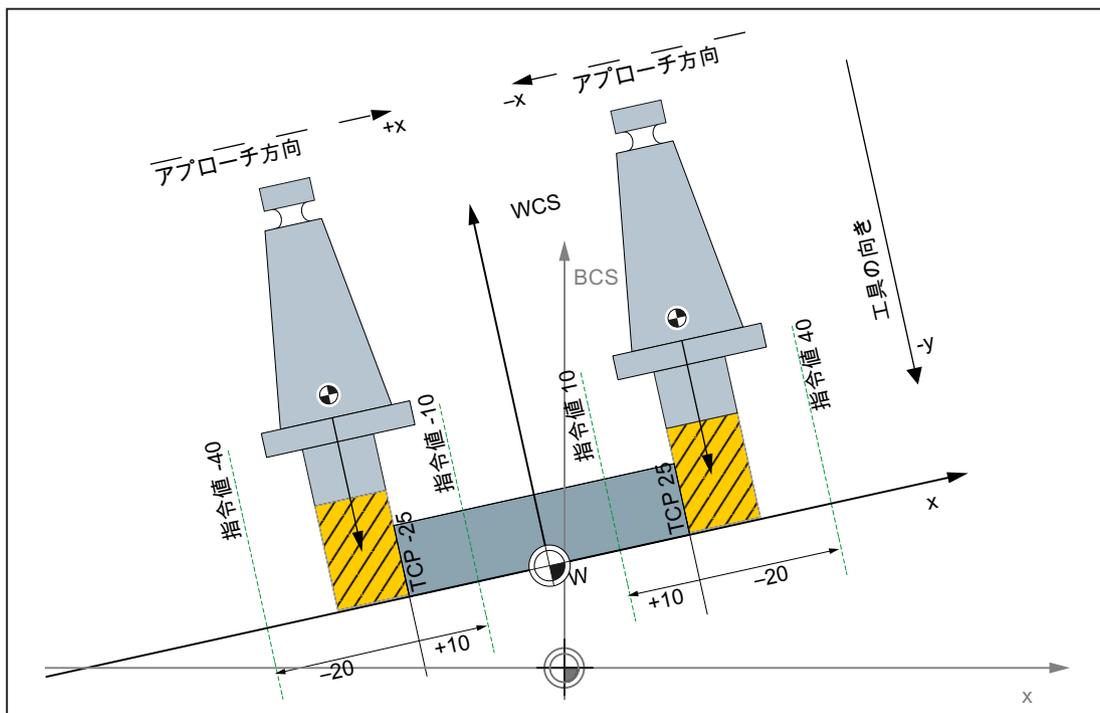


図 9-23 それぞれにレファレンス点を持つ 2つのフライス工具

システムデータの設定:

左側工具: アプローチ方向+x と工具の向き-y	
システム変数	意味
\$AC_MEAS_TOOL_MASK = 0x80	-y 方向の工具位置
\$AC_MEAS_DIR_APPROACH = 0	アプローチ方向+x

右側工具: アプローチ方向-x と工具の向き-y	
システム変数	意味
\$AC_MEAS_TOOL_MASK = 0x80	-y 方向の工具位置
\$AC_MEAS_DIR_APPROACH = 1	アプローチ方向-x

両工具について	
システム変数	意味
\$AC_MEAS_Px_COORD = 0	x 番目の計測点の座標系 = WCS (初期設定)
\$AC_MEAS_SET_COORD = 0	指令値の座標系 = WCS (初期設定)

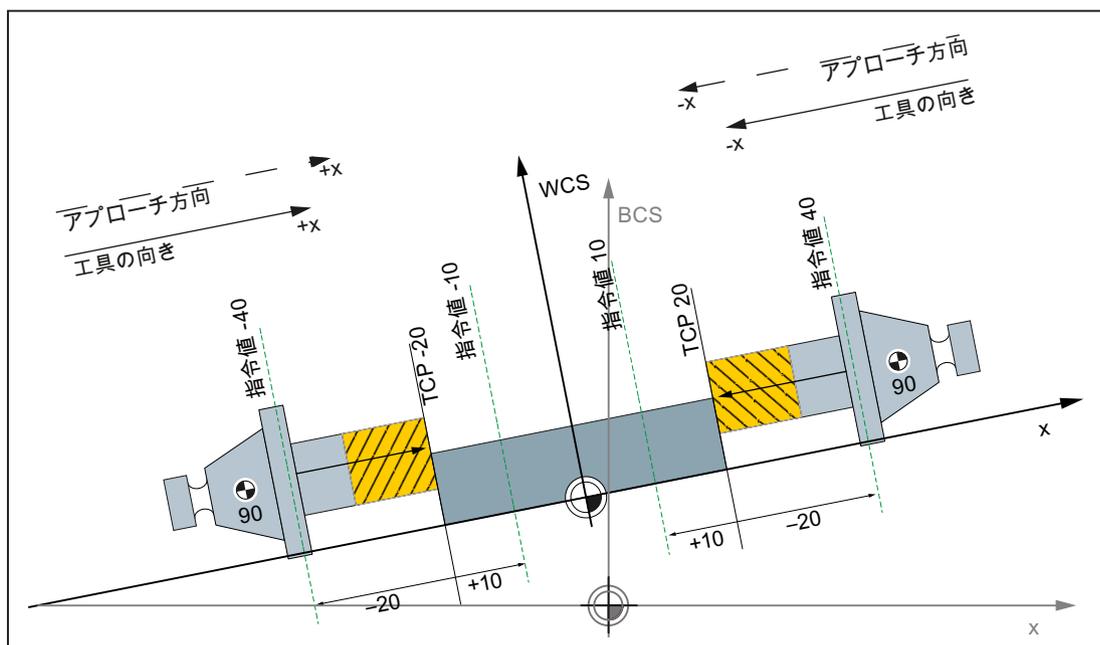


図 9-24 それぞれにレファレンス点を持つ、90°回転された 2 つのフライス工具  
システムデータの設定:

左側工具: アプローチ方向+x と工具の向き-y	
システム変数	意味
\$AC_MEAS_TOOL_MASK = 0x2	x 方向の工具位置(G19)。
\$AC_MEAS_DIR_APPROACH = 0	アプローチ方向+x

右側工具: アプローチ方向-x と工具の向き-y	
システム変数	意味
\$AC_MEAS_TOOL_MASK = 0x40	-x 方向の工具位置。
\$AC_MEAS_DIR_APPROACH = 1	アプローチ方向-x

両工具について	
システム変数	意味
\$AC_MEAS_Px_COORD = 0	x 番目の計測点の座標系 = WCS (初期設定)
\$AC_MEAS_SET_COORD = 0	指令値の座標系 = WCS (初期設定)

## 9.6 測定精度および機能テスト

### 9.6.1 計測精度

計測精度は次のパラメータによって影響を受けます:

- 計測信号( $T_{\text{Delay}}$ )の遅延時間
- 計測時の移動速度 ( $v_M$ )

#### 計測信号( $T_{\text{Delay}}$ )の遅延時間補正

計測信号の遅延時間(つまり、プローブの応答時間および制御ハードウェアの信号ランタイムに応じて、制御装置で測定された値が保存されるまでのプローブの起動以降の時間)。この制御装置は、計測時の遅延時間を補正します。このためには、次のマシンデータで遅延時間を特定および入力してください:

MD13220 \$MN\_MEAS\_PROBE\_DELAY\_TIME = <特定された遅延時間>

---

#### 注記

##### 最大補正遅延時間 $T_{\text{MaxDelay}}$

$T_{\text{MaxDelay}} = 31 * \text{位置コントローラまたは DP サイクル}$

遅延時間の補正  $> T_{\text{MaxDelay}}$  とすることは、制御装置の観点からお勧めしません。つまり、より大きな値は  $T_{\text{MaxDelay}}$  に制限されます。

---

##### 計測時の最大移動速度 ( $v_M$ )

計測で許容される最大移動速度は、プログラム指令された計測端面の数、およびパラメータ設定された位置コントローラまたは DP サイクルに依存します。

正しい結果を得るために、**2つの**各位置コントローラまたは DP サイクルで次の条件が満たされるように、計測時の移動速度を選択してください:

- 最大で**1つの**同一トリガ信号(つまり**1つの**プローブの**1つの**立ち上がりエッジまたは**1つの**立ち下がりエッジ)
- 最大で**4つの異なる**同一トリガ信号(つまり**2つの**プローブの**1つの**立ち上がりエッジおよび**1つの**立ち下がりエッジ)

## 9.6.2 プローブ機能テスト

### 機能テスト例

表 9-13

プログラムコード	コメント
%_N_PRUEF_MESSTASTER_MPF	
;\$PATH=/_N_MPF_DIR	
;プローブ接続テストのプログラム	
N05 DEF INT MTSIGNAL	;トリガ状態のフラグ
N10 DEF INT ME_NR=1	;計測入力番号
N20 DEF REAL MESSWERT_IN_X	
N30 G17 T1 D1	;事前に選択されたプローブのための
	;工具補正
N40 _ANF: G0 G90 X0 F150	;開始位置および
	;計測速度
N50 MEAS=ME_NR G1 X100	; X 軸の計測入力 =1 で
	;計測
N60 STOPRE	
N70 MTSIGNAL=\$AC_MEA[1]	;1 番目の計測入力でソフトウェア切り替え信号の
	;読み込み
N80 IF MTSIGNAL == 0 GOTOF _FEHL1	;信号の評価
N90 MESSWERT_IN_X=\$AA_MW[X]	;ワーク座標の計測値の
	;読み込み
N95 M0	
N100 M02	
N110 _FEHL1: MSG (「Probe not switching!」)	
N120 M0	
N130 M02	

## 9.7 計測シミュレーション

### 9.7.1 一般機能

#### 概略説明

実際の機械で計測をおこなうには、特定の位置で切り替え信号を送信するプローブを接続します。シミュレーション環境で計測をおこなう場合、プローブは使用されません。この場合、切り替え位置は別の方法で指定されます。

シミュレーション計測での切り替え位置の入力では、後述の2つの方法がサポートされています。

- 位置に対応した切り替え要求 切り替え位置は、計測ブロックでプログラム指令された軸の終了位置から取得されます。
- 外部からの切り替え要求 切り替え位置は、デジタル出力の制御により定義されません。

## 必要条件

計測シミュレーションでは、システム内の機械軸すべてをシミュレーションされる軸として次のようにパラメータ設定してください。

- MD30130 \$MA\_CTRLOUT\_TYPE[軸] = 0 (シミュレーションされる指令値)
- MD30240 \$MA\_ENC\_TYPE[軸] = 0 (シミュレーションされるエンコーダ)

### 9.7.2 位置に対応した切り替え要求

#### 機能

「位置に対応した切り替え要求」は、後述の NC 用のマシンデータを使用して選択します。

- MD13230 \$MN\_MEAS\_PROBE\_SOURCE = 0
- MD13231 \$MN\_MEAS\_PROBE\_OFFSET = <位置オフセット>

軸の切り替え位置は、計測ブロックでプログラム指令された軸の終了位置とパラメータ設定された位置オフセットから計算されます。

切り替え位置[軸] = 終了位置[軸] - 位置オフセット

ブロックの計測中に、軸の切り替え位置に到達したかどうか周期的にチェックされます。

指令位置[軸] ≥ 切り替え位置[軸]

切り替え位置に到達すると、切り替え信号の立ち上がりエッジが、プローブ 1 および 2 に対して生成されます。1 回の位置制御周期の後、次のエッジが生成されます。

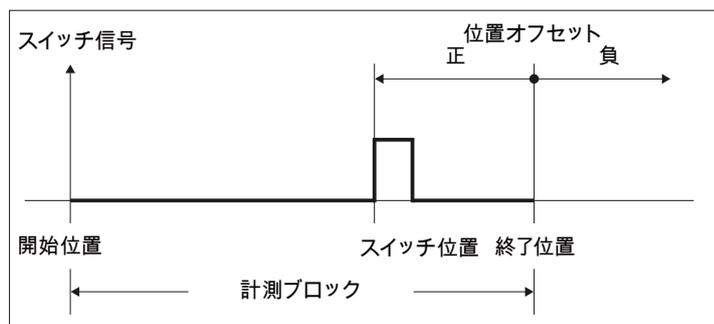


図 9-25 位置に依存する切り替え要求

計測値は、計測ブロックでプログラム指令された切り替え信号が発生した時点での軸の現在値です(立ち上がりエッジ/立ち下がりエッジ)。

計測ブロックで複数の軸がプログラム指令されている場合は、軸方向に考慮される位置オフセットによって、各軸に固有の切り替え位置が得られます。プローブ信号は、最初の軸切り替え位置に達したときに生成されます。

#### 注記

##### プローブ信号

信号はプローブ 1 と 2 に対して常に同時に生成されます。

##### 負のオフセット値

位置オフセットに負の値を入力すると、切り替え位置は終了位置の外側にシフトされません。この場合、プローブ信号は生成されません。

## 例

次の場合、位置オフセットは 0.1 mm に設定されます。MD13231

`$MN_MEAS_PROBE_OFFSET = 0.1`

### 例 1:2 つの軸でのチャンネル別計測

プログラムコード	コメント
N10 G01 G90	
N20 MEAS=1 X100 Y10 F100	; 立ち上がり、プローブ 1 ; 切り替え位置[X] = 99.9 ; 切り替え位置[Y] = 9.9

## 例 2:シンクロナイズドアクションを使用した軸計測

プログラムコード	コメント
N10 G01 G90	
N15 WHEN TRUE DO MEASA[X]=(1,1)	; 立ち上がり、プローブ 1
N20 X10 F100	; 切り替え位置[X] = 9.9

## 9.7.3 外部切り替え要求

## 機能

「外部切り替え要求」は、使用されるデジタル出力の数(1 ~ 8)を入力することにより、NC 用マシンデータを使用して選択されます。

- MD13230 \$MN\_MEAS\_PROBE\_SOURCE = <デジタル出力の数>

プローブ信号は、設定されたデジタル出力の制御によりトリガされます。デジタル出力を計測入力に固定する必要はありません。

プローブ 1 と 2 に対する切り替え信号の立ち上がりエッジは、デジタル出力の設定により生成されます。立ち下がりエッジは、デジタル出力のリセットにより生成されません。

計測値は、計測ブロックでプログラム指令された切り替え信号が発生した時点での軸の現在値です(立ち上がりエッジ/立ち下がりエッジ)。

## デジタル出力:設定

計測シミュレーションでデジタル出力を使用できるようにするには、次のマシンデータを設定してください。

- MD10360 \$MN\_FASTIO\_DIG\_NUM\_OUTPUTS = 1 (有効桁数。NC 出力バイト)
- MD13120 \$MN\_CONTROL\_UNIT\_LOGIC\_ADDRESS = 0 (論理アドレス、SINAMICS-CU)

## デジタル出力:設定

デジタル出力設定は、シンクロナイズドアクションで設定できます。

```
WHEN <条件> DO $A_OUT[<number of digital output>] = 1
```

## 例

使用されるデジタル出力:MD13230 \$MN\_MEAS\_PROBE\_SOURCE = 1

## 例 1:2つの軸でのチャンネル別計測

プログラムコード	コメント
N10 G01 G90 \$A_OUT[1]=0	; デジタル出力1の事前割り当て
N15 WHEN \$AC_DETW<=10 DO \$A_OUT[1]=1	; 軌跡の残移動距離<= 10 =>デジタル出力1 = 1
N20 MEAS=1 X100 Y10 F100	; 立ち上がり、プローブ1

## 例 2:軸の計測

プログラムコード	コメント
N10 G01 G90 \$A_OUT[1]=0	; デジタル出力1の事前割り当て
N15 WHEN \$AA_IW[X]>=80 DO \$A_OUT[1]=1	; 軸の指令値>= 80 =>デジタル出力1 = 1
N20 MEASA[X]=(1,1) X100 F100	; 立ち上がり、プローブ1

## 9.7.4 システム変数

計測シミュレーションでの後述のシステム変数は、実際の計測の場合と同じ機能を持ちます。

- \$AC\_MEA (プローブ検出済み)
- \$AA\_MEA ACT (軸計測が有効)
- \$AA\_MM (取得されたプローブ位置(MCS))
- \$AA\_MM1...4 (プローブ位置1番目 - 4番目のトリガ(MCS))
- \$AA\_MW (取得されたプローブ位置(WCS))
- \$AA\_MW1...4 (プローブ位置1番目のトリガ(WCS))

次のシステム変数は適切な値を提供しません。

- \$A\_PROBE (プローブ状態)

## 9.8 チャンネル - 840D sl のみ

### 9.8.1 計測モード 1

補足条件

- 1回計測
- 1つのプローブ
- トリガ信号は立ち上がりと立ち下りです

#### 1つのエンコーダによる計測 - 現在のエンコーダの現在値

プログラムコード

```
MEASA[X] = (1, 1, -1) G01 X100 F100
STOPRE
IF $AC_MEA[1]==FALSE GOTOF ENDE
R10=$AA_MM1[X]
R11=$AA_MM2[X]
END
```

#### 2つのエンコーダによる計測 - 2台のエンコーダの現在値

プログラムコード

```
MEASA[X] = (31, 1, -1) G01 X100 F100
STOPRE
IF $AC_MEA[1]==FALSE GOTOF ENDE
R10=$AA_MM1[X]
R11=$AA_MM2[X]
R12=$AA_MM3[X]
R13=$AA_MM4[X]
END
```

## 9.8.2 計測モード 2

補足条件

- 2つのプローブ
- トリガ信号は立ち上がりとしち下りです
- 現在のエンコーダからのフィードバック値

プログラムコード

```
MEASA[X] = (2, 1, -1, 2, -2) G01 X100 F100
STOPRE
IF $AC_MEA[1]==FALSE GOTOF MESSTASTER2
R10=$AA_MM1[X]
R11=$AA_MM2[X]
PROBE2
IF $AC_MEA[2]==FALSE GOTOF ENDE
R12=$AA_MM3[X]
R13=$AA_MM4[X]
ENDE
```

## 9.8.3 連続計測

必要条件

- 計測は計測モード 1 で実行:
- 100 個の値の計測
- 1つのプローブ
- 信号のトリガ:立ち下りエッジ
- 現在のエンコーダからのフィードバック値

プログラム指令された移動動作完了時の連続計測

プログラムコード	コメント
DEF REAL MESSWERT[100]	
DEF INT INDEX=0	
MEAC[x]=(1, 1, -1) G01 X1000 F100	
MEAC[X]=(0)	; 中止
R1=\$AC_FIFO1[4]	;計測値の数
FOR INDEX=0 TO R1 - 1	

## 9.8 チャンネル - 840D sl のみ

プログラムコード	コメント
MESSWERT[INDEX]=\$AC_FIFO1[0] ENDFOR	;計測値の読み取り

## 残移動距離を削除する連続計測

最後の計測後に残移動距離を削除

プログラムコード	コメント
DEF INT ANZAHL=100 DEF REAL MESSWERT[ANZAHL] DEF INT INDEX=0 WHEN \$AC_FIFO1[4]==ANZAHL DO DELDTG (X) MEAC[X] =(0) MEAC[X]=(1, 1, -1) G01 X1000 F100 R1=\$AC_FIFO1[4] FOR INDEX=0 TO R1 - 1 MESSWERT[INDEX]=\$AC_FIFO1[0] ENDFOR	;計測の開始 ;計測値の数 ;計測値の読み取り

## 複数のブロックにわたる連続モダルの計測

プログラムコード	コメント
DEF INT ANZAHL=100 DEF REAL MESSWERT[ANZAHL] DEF INT INDEX=0 ID=1 MEAC[X]=(1, 1, -1) ID=2 WHEN \$AC_FIFO1[4]==ANZAHL DO MEAC[X]=(0) CANCEL(2) G01 X1000 Y100 NX100 Y100 R1=\$AC_FIFO1[4] FOR INDEX=0 TO R1 - 1 MESSWERT[INDEX]=\$AC_FIFO1[0] ENDFOR	;計測の開始 ;計測値の数 ;計測値の読み取り

## 9.8.4 機能テストと繰り返し精度

## 機能テスト

プログラムコード	コメント
%_N_PRUEF_MESSTASTER_MPF ;\$PATH=/_N_MPF_DIR	

プログラムコード	コメント
;プローブ接続テストのプログラム	
DEF INT MTSIGNAL	;トリガ状態のフラグ
DEF INT ME_NR=1	;計測入力番号
DEF REAL MESSWERT_IN_X	
G17 T1 D1	;事前に選択されたプローブのための
	;工具補正
_ANF:G0 G90 X0 F150	;開始位置および
	;計測速度
MEAS=ME_NR G1 X100	; X 軸の計測入力 =1 で
	;計測
STOPRE	
MTSIGNAL=\$AC_MEA[1]	;1 番目の計測入力ソフトウェア切り替え信号の
	;読み込み
IF MTSIGNAL == 0 GOTOF _FEHL1	;信号の評価
MESSWERT_IN_X=\$AA_MW[X]	;ワーク座標の計測値の
	;読み込み
M0	
M02	
_FEHL1:MSG (「Probe not switching!」)	
M0	
M02	

## 繰り返し精度

このプログラムでは、検出器全体(機械-プローブ-信号の NC への送信)の計測のばらつき(繰り返し精度)を計算できます。

この例では、10 件の計測が X 軸でおこなわれ、計測値がワーク座標に記録されます。

したがって、どの傾向にも左右されないランダムな寸法誤差を計測できます。

プログラムコード	コメント
%_N_PRUEF_GENAU_MPF;	
\$PATH=/_N_MPF_DIR	
DEF INT SIGNAL, II	;変数の定義
DEF REAL MESSWERT_IN_X[10]	
G17 T1 D1	;初期状態、
	;事前に選択されたプローブのための
	;工具補正
_ANF:G0 X0 F150	;1 番目の計測入力時の
MEAS=+1 G1 X100	;計測される軸の事前位置決め
	; (切り替え信号がスイッチオンしないとき、
	;および X 軸でスイッチオンするとき)

## 9.9 データリスト

プログラムコード	コメント
STOPRE	;その後におこなわれる結果の評価の後で ;この先読みを ;停止します
SIGNAL= \$AC_MEA[1]	;1番目の計測入力でソフトウェア切り替え信号の ;読み取り
IF SIGNAL == 0 GOTOF_FEHL1	;切り替え信号をチェック
MESSWERT_IN_X[II]=\$AA_MW[X]	;ワーク座標で計測値を読み込み
II=II+1	
IF II<10 GOTOB_ANF	;10回繰り返し
M0	
M02	
_FEHL1:MSG (「Probe not switching」)	
M0	
M02	

パラメータ表示(カスタマイズ変数)を選択した後、プログラム実行が有効であるかぎり、測定結果は MESSWERT\_IN\_X[10] フィールドで読み込むことができます。

## 9.9 データリスト

## 9.9.1 マシンデータ

## 9.9.1.1 一般マシンデータ

番号	識別子: \$MN_	意味
13200	MEAS_PROBE_LOW_ACTIVE	プローブの切り替えの特性
13201	MEAS_PROBE_SOURCE	デジタル出力による計測パルスシミュレーション
13210	MEAS_TYPE	PROFIBUS DP ドライブの計測タイプ
13211	MEAS_CENTRAL_SOURCE	PROFIBUS DP ドライブ装置によるセントラル計測データソース

## 9.9.1.2 チャンネル別マシンデータ

番号	識別子: \$MC_	意味
20360	TOOL_PARAMETER_DEF_MASK	工具パラメータの定義
28264	LEN_AC_FIFO	\$AC_FIFO ... FIFO 変数の長さ

## 9.9.2 システム変数

すべての入力値の表

識別子	意味
\$AC_FIFO1...10	FIFO 変数 1 ~ 10
\$AC_MEAS_SEMA	インタフェースの割り当て
\$AC_MEAS_VALID	入力値の有効ビット
\$AA_MEAS_POINT1	1. 番目のすべてのチャンネル軸の計測点
\$AA_MEAS_POINT2	2. 番目のすべてのチャンネル軸の計測点
\$AA_MEAS_POINT3	3. 番目のすべてのチャンネル軸の計測点
\$AA_MEAS_POINT4	4. 番目のすべてのチャンネル軸の計測点
\$AA_MEAS_SETPOINT	すべてのチャンネル軸の指令位置
\$AA_MEAS_SETANGLE	すべてのチャンネル軸の指令角度
\$AC_MEAS_P1_COORD	1 番目の計測点の座標系
\$AC_MEAS_P2_COORD	2 番目の計測点の座標系
\$AC_MEAS_P3_COORD	3 番目の計測点の座標系
\$AC_MEAS_P4_COORD	4 番目の計測点の座標系
\$AC_MEAS_SET_COORD	指令値の座標系
\$AC_MEAS_LATCH[0...3]	WCS で計測点を取得
\$AA_MEAS_P1_VALID	1. 番目の計測点を WCS で取得
\$AA_MEAS_P2_VALID	2. 番目の計測点を WCS で取得
\$AA_MEAS_P3_VALID	3. 番目の計測点を WCS で取得
\$AA_MEAS_P4_VALID	4. 番目の計測点を WCS で取得
\$AA_MEAS_SP_VALID	有効軸の指令位置を設定

識別子	意味
\$AC_MEAS_WP_SETANGLE	ワーク位置指令角度
\$AC_MEAS_CORNER_SETANGLE	コーナーの指令切削角度
\$AC_MEAS_DIR_APPROACH	アプローチ方向
\$AC_MEAS_ACT_PLANE	ワークの作業平面
\$AC_MEAS_SCALEUNIT	単位系 INCH/METRIC
\$AC_MEAS_FINE_TRANS	精密位置誤差補正
\$AC_MEAS_FRAME_SELECT	ワーク計測のフレーム選択
\$AC_MEAS_CHSFR	フレーム結合の設定: システムフレーム
\$AC_MEAS_NCBFR	フレーム結合の設定: グローバル基本フレーム
\$AC_MEAS_CHBFR	フレーム結合の設定: チャンネル基本フレーム
\$AC_MEAS_UIFR	フレーム結合の設定: 設定可能フレーム
\$AC_MEAS_PFRAME	フレーム結合の設定: プログラムフレーム
\$AC_MEAS_T_NUMBER	工具選択
\$AC_MEAS_D_NUMBER	刃先選択
\$AC_MEAS_TOOL_MASK	工具設定内容
\$AC_MEAS_TYPE	計測タイプ
\$AC_MEAS_INPUT	計測入力パラメータ

## すべての出力値の表

識別子	意味
\$A_PROBE[1,2]	プローブ状態
\$A_PROBE_LIMITED[1,2]	計測速度超過
\$AC_MEA[1,2]	プローブ検出済み
\$AA_MM	取得されたプローブ位置(MCS)
\$AA_MM1...4	プローブ位置 1～4 番目のトリガ事象(MCS)
\$AA_MW	取得されたプローブ位置(WCS)
\$AA_MW1...4	プローブ位置 1～4 番目のトリガ事象(WCS)
\$AC_MEAS_FRAME	フレーム結果
\$AC_MEAS_WP_ANGLE	ワーク位置計算角度

識別子	意味
\$AC_MEAS_CORNER_ANGLE	計算交点角度
\$AC_MEAS_DIAMETER	計算直径
\$AC_MEAS_TOOL_LENGTH	計算工具長
\$AC_MEAS_RESULTS	計測結果(計測タイプにより異なります)



# N3: ポジションスイッチ、ポジションスイッチサイクル - 840D sl のみ

# 10

## 10.1 概略説明

### 機能

「ポジションスイッチ」機能により、位置フィードバック値(機械軸)を供給する軸とシミュレーション軸に対して、位置に依存する切り替え信号が生成されます。これらのスイッチ信号は、PLC だけでなく、NC I/O にも出力できます。

信号が出力される動作位置は、セッティングデータによって定義と変更ができます。セッティングデータは、HMI、PLC、およびパートプログラムによって、読み書きが可能です。

### 適用

「ポジションスイッチ」機能は、すべての運転モードで有効にして使用できます。この機能は、リセットまたは非常停止の場合でも、そのまま有効です。

### 応用分野

出力スイッチ信号を使用可能です。以下に例を示します。

- 保護領域を有効にする場合
- 位置決めに応じて、追加移動をおこなう場合
- 油圧制御の揺動軸のための反転信号となる場合

### 軸タイプ

ポジションスイッチは、機械軸として定義された直線軸とモジュール回転軸上で使用できます。

### 信号出力領域/スイッチペア

スイッチは常に、ペアで軸に割り当ててください。ペアは、プラス側とマイナス側のスイッチで構成されます。32 組のスイッチペアを使用できます。

## 10.2 ポジションスイッチ信号および動作位置

プラス側とマイナス側のスイッチはそれぞれ、軸が動作位置に達したときに、定義位置(動作位置)で指定アプローチ方向に作動する機械式スイッチをシミュレーションします。信号出力領域は、プラス側とマイナス側のスイッチへ次のように割り当てられます。

- 信号出力プラス領域: すべての位置 $\geq$ プラス側スイッチ
- 出力信号マイナス領域: すべての位置 $\leq$ マイナス側スイッチ

## 10.2 ポジションスイッチ信号および動作位置

### 10.2.1 個別出力のためのポジションスイッチ信号の生成

プラス側とマイナス側のポジションスイッチ信号を個別に出力すると、軸がプラス側とマイナス側のスイッチ領域の内部にあるか、外部にあるかを検出しやすくなります。

## 直線軸

ポジションスイッチ信号の切り替えエッジは、次のように軸の移動方向に応じて生成されます。

- マイナス側スイッチ信号は、軸がマイナス側スイッチを正の軸方向へ移動すると、1 から 0 に切り替わります。
- プラス側スイッチ信号は、軸がプラス側スイッチを正の軸方向へ移動すると、0 から 1 に切り替わります。

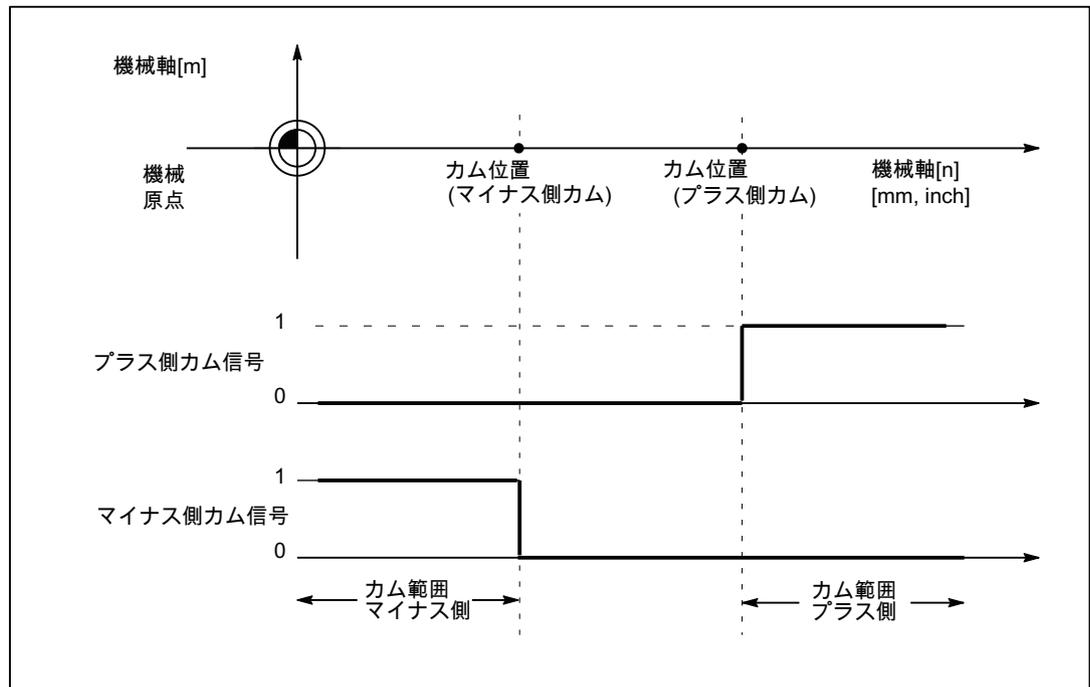


図 10-1 直線軸のポジションスイッチ(マイナス側スイッチ < プラス側スイッチ)

## 注記

軸が出力スイッチ位置(プラスまたはマイナス)の真上にある場合は、定義された出力が点滅します。軸がさらに 1 ステップ単位移動すると、出力は明確に 0 または 1 となります。

実位置が揺らぐと、信号が同様に点滅します。  
実位置が評価されます。

10.2 ポジションスイッチ信号および動作位置

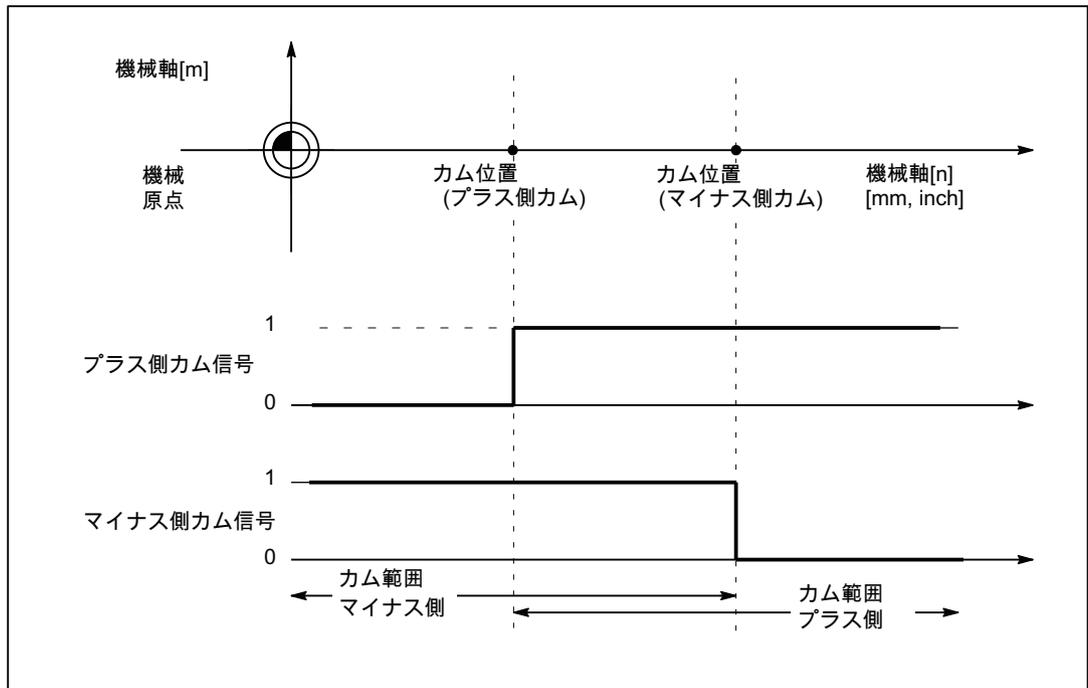


図 10-2 直線軸のポジションスイッチ(プラス側スイッチ < マイナス側スイッチ)

## モジュロ回転軸

ポジションスイッチ信号の切り替えエッジは、次のように回転軸の移動方向に応じて生成されます。

- プラス側スイッチ信号は、軸がマイナス側スイッチを正の軸方向に横切ると、0 から 1 に切り替わり、プラス側スイッチを横切ると、再度 1 から 0 に切り替わります。
- マイナス側スイッチ信号は、プラス側スイッチ信号の各立ち上がりエッジに応じて、レベルを変更します。

## 注記

説明したプラス側スイッチの動作は、次の条件が満たされる場合に適用されます。  
 プラス側スイッチ - マイナス側スイッチ < 180°

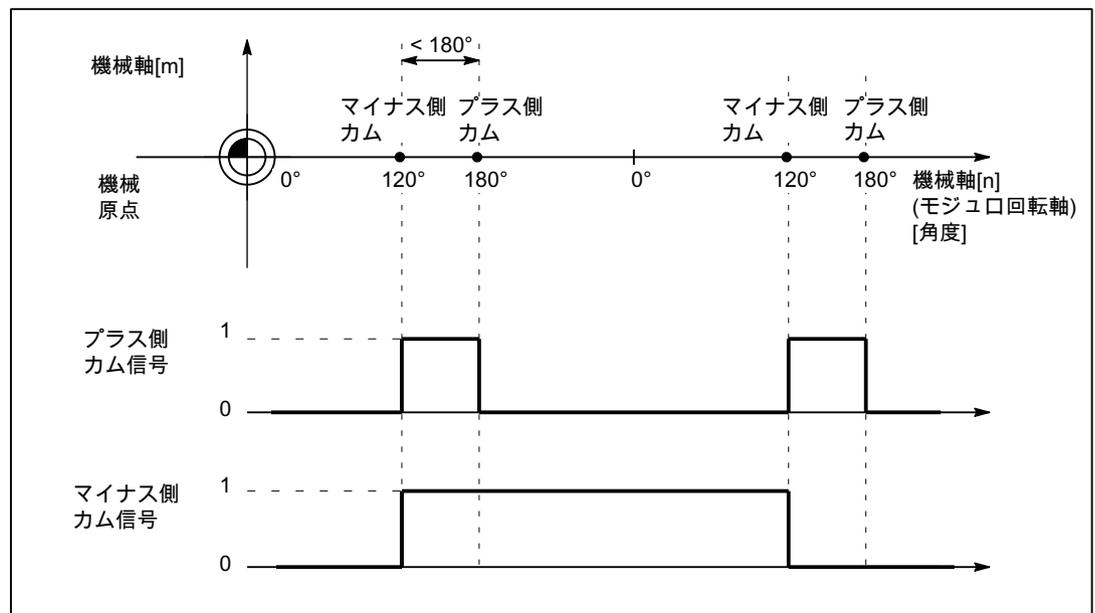


図 10-3 モジュロ回転軸のポジションスイッチ(プラス側スイッチ - マイナス側スイッチ < 180°)

マイナス側スイッチの信号の変化は、PLC が確実に検出できないほど信号出力領域が小さく設定されている場合でも、スイッチの移動を検出できます。

ポジションスイッチ信号は両方とも、PLC と NC I/O に出力できます。プラス側とマイナス側のポジションスイッチ信号を個別に出力すると、軸がプラス側とマイナス側のスイッチ領域の内部にあるか、外部にあるかを検出しやすくなります。

この条件(プラス側スイッチ - マイナス側スイッチ < 180°)が満たされない、またはマイナス側スイッチがプラス側スイッチより大きい値に設定されている場合は、プラス側スイッチ信号の動作が反転します。マイナス側スイッチ信号応答は変わりません。

10.2 ポジションスイッチ信号および動作位置

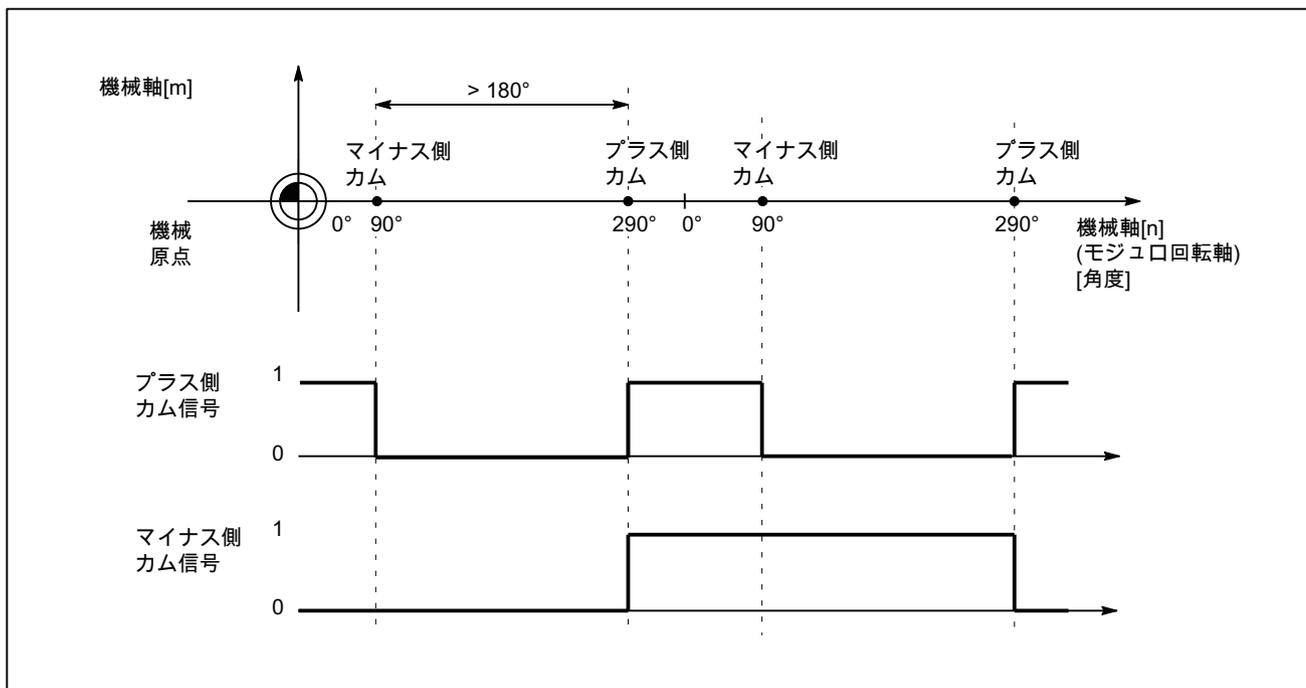


図 10-4 モジュロ回転軸のポジションスイッチ(プラス側スイッチ - マイナス側スイッチ > 180°)

10.2.2 ゲート制出力をもつポジションスイッチ信号の生成

プラス側とマイナス側のポジションスイッチ出力信号は、次の場合にゲート制御されます。

- NCU 上の 4 つのオンボード出力部への割り込み型ポジションスイッチ信号出力
- 次のマシンデータの 2 番目のバイトを指定しなかった場合(= "0")、NCK I/O への出力  
 MD10470 SW\_CAM\_ASSIGN\_FASTOUT\_2  
 ...  
 MD10473 SW\_CAM\_ASSIGN\_FASTOUT\_4

直線軸

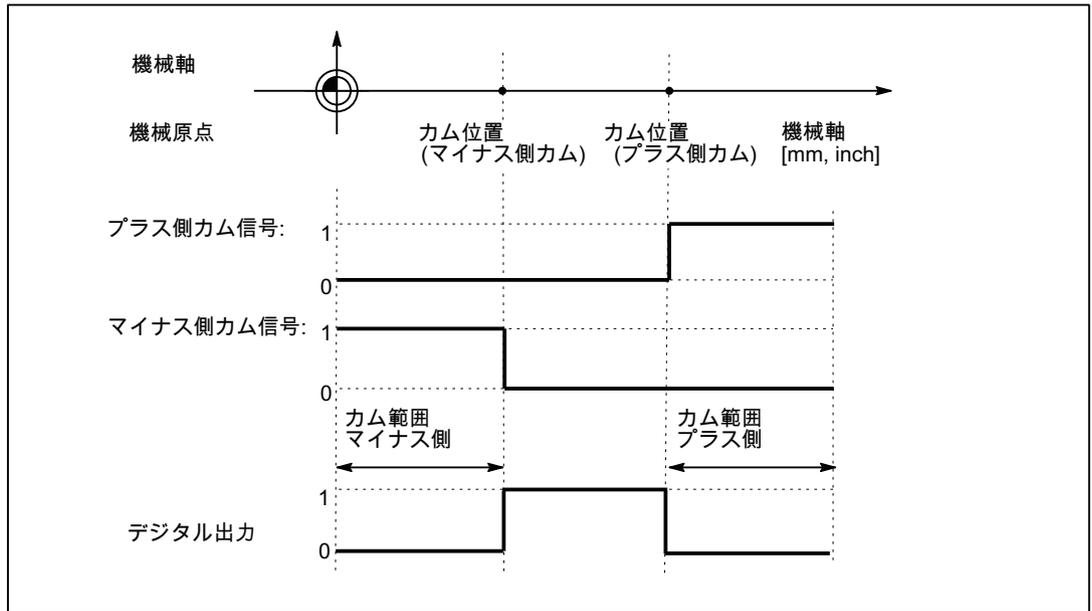


図 10-5 直線軸のポジションスイッチ信号(マイナス側スイッチ < プラス側スイッチ)

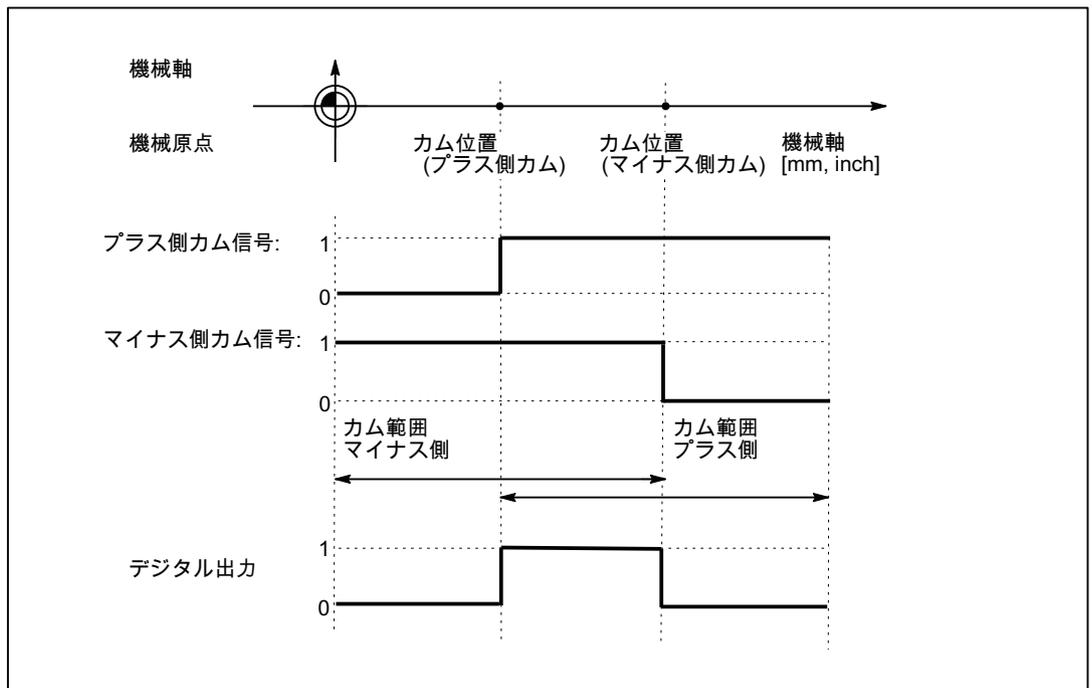


図 10-6 直線軸のポジションスイッチ信号(プラス側スイッチ < マイナス側スイッチ)

10.2 ポジションスイッチ信号および動作位置

モジュール回転軸

モジュール回転軸の初期設定の信号応答は、スイッチ幅によって決まります。

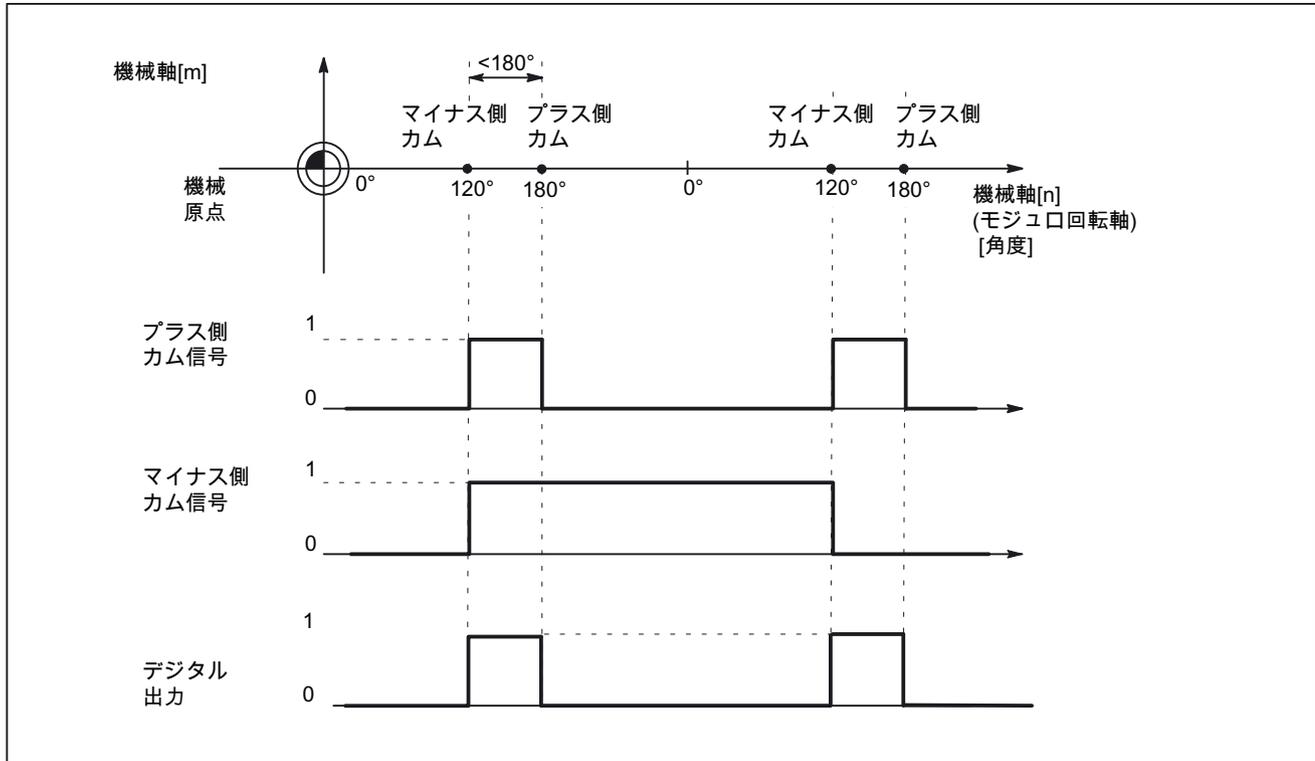


図 10-7 モジュール回転軸のポジションスイッチ(プラス側スイッチ - マイナス側スイッチ  $< 180^\circ$ )

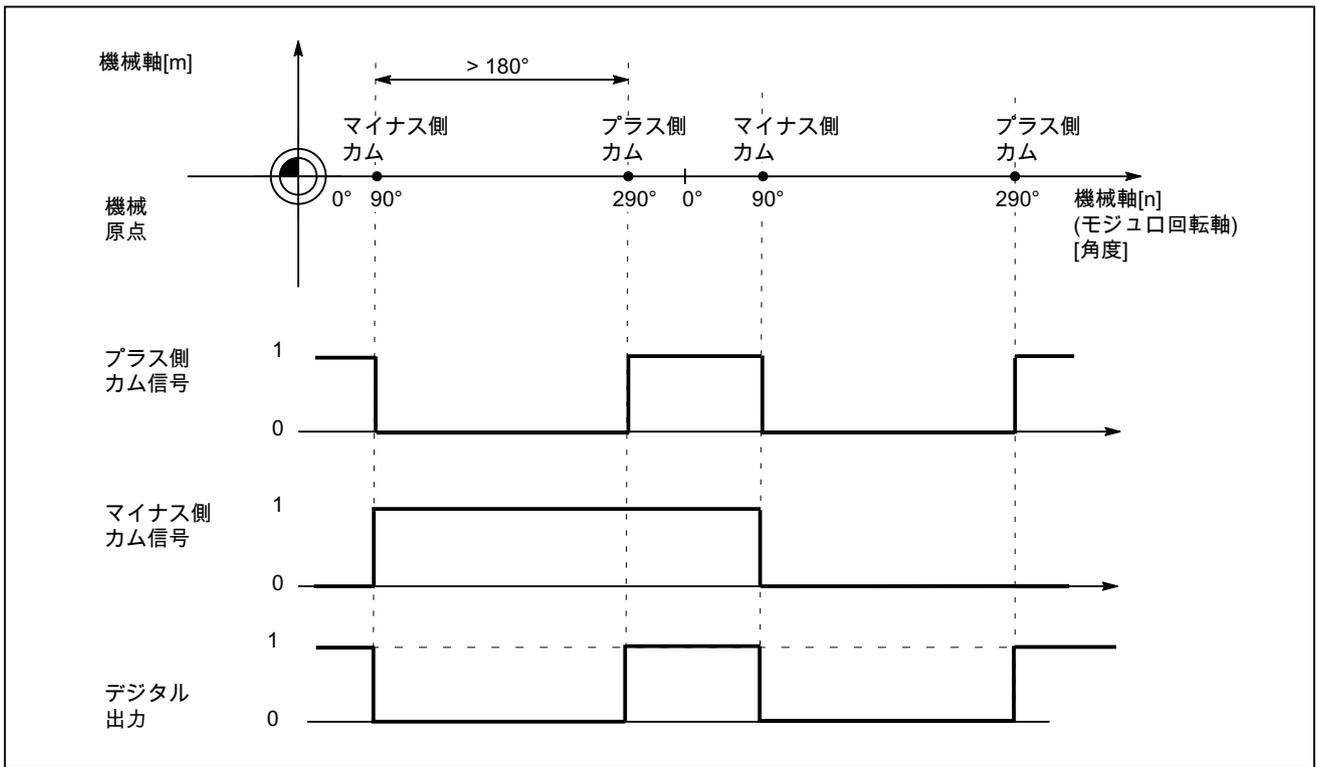


図 10-8 モジュロ回転軸のポジションスイッチ(プラス側スイッチ - マイナス側スイッチ > 180°)

### 信号反転の抑制

次の設定では、「  
 プラス側ポジションスイッチ - マイナス側ポジションスイッチ > 180°」の信号反転の選  
 択を抑制できます。

MD10485 SW\_CAM\_MODE ビット 1=1

10.2 ポジションスイッチ信号および動作位置

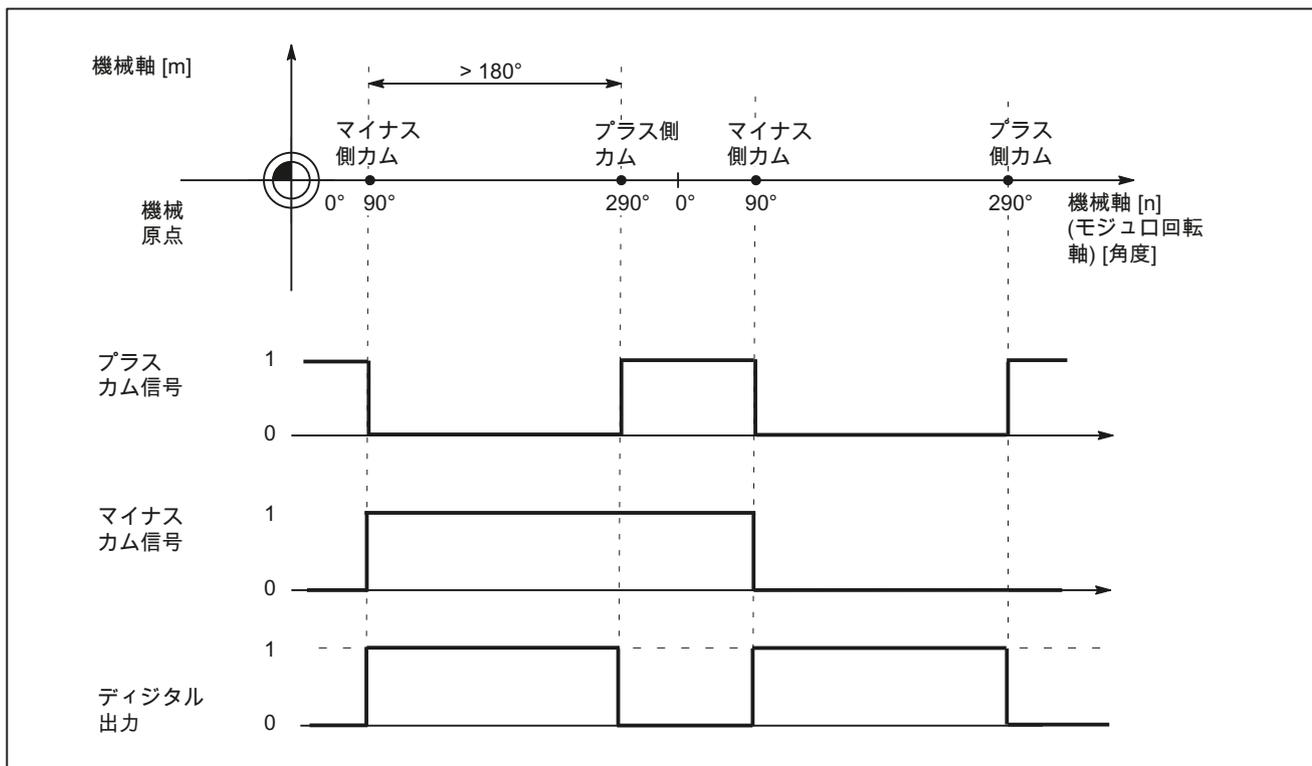


図 10-9 モジュロ回転軸のポジションスイッチ(プラス側スイッチ - マイナス側スイッチ > 180°)と信号反転の抑制

10.2.3 動作位置

動作位置の設定

プラス側ポジションスイッチとマイナス側ポジションスイッチの位置は、次の一般セッティングデータによって定義されます。

- SD41500 SW\_CAM\_MINUS\_POS\_TAB\_1[n]      マイナス側スイッチの位置 1 ~ 8
- SD41501 SW\_CAM\_PLUS\_POS\_TAB\_1[n]      プラス側スイッチの位置 1 ~ 8
- SD41502 SW\_CAM\_MINUS\_POS\_TAB\_2[n]      マイナス側スイッチの位置 9 ~ 16
- SD41503 SW\_CAM\_PLUS\_POS\_TAB\_2[n]      プラス側スイッチの位置 9 ~ 16
- SD41504 SW\_CAM\_MINUS\_POS\_TAB\_3[n]      マイナス側スイッチの位置 17 ~ 24
- SD41505 SW\_CAM\_PLUS\_POS\_TAB\_3[n]      プラス側スイッチの位置 17 ~ 24
- SD41506 SW\_CAM\_MINUS\_POS\_TAB\_4[n]      マイナス側スイッチの位置 25 ~ 32
- SD41507 SW\_CAM\_PLUS\_POS\_TAB\_4[n]      プラス側スイッチの位置 25 ~ 32

**注記**

スイッチのペア(各グループに 8 組)のグループを作成して、異なるアクセス権レベル(機械関連およびワーク関連の動作位置のアクセス権レベルなど)を割り当てることができます。スイッチの位置は、機械座標系で入力されます。最大移動範囲に関してはチェックされません。

**単位系、メトリック/インチ****MD10260 CONVERT\_SCALING\_SYSTEM = 1**

の設定では動作位置は、設定された基本単位系ではなく、次のマシンデータで設定された単位系を参照するようになります。

**MD10270 POS\_TAB\_SCALING\_SYSTEM (位置テーブルの検出器)**

規格値	意味
0	メトリック
1	インチ

したがって MD10270 は、セッティングデータ SD41500 ... SD41507 からの位置データの単位系を定義します。

G70/G71 または G700/G710 による切り替えは動作しません。

**動作位置の検知**

ポジションスイッチ信号を設定するために、各軸の実位置が動作位置と比較されます。

**動作位置の書き込み/読み込み**

セッティングデータは、HMI、PLC、およびパートプログラムによって、読み書きが可能です。

パートプログラムからのアクセスは、加工と同期しません。

同期制御は、プログラム指令ブロック先読み停止処理(STOPRE 命令)を使用した場合にのみ実行されます。

PLC ユーザプログラムの FB2 と FB3 で、動作位置の読み込みと書き込みを実行できます。

## 10.2 ポジションスイッチ信号および動作位置

### 軸/スイッチの割り当て

スイッチペアは、次の一般マシンデータを使用して機械軸に割り当てられます。

MD10450 SW\_CAM\_ASSIGN\_TAB[n] (機械軸へのポジションスイッチの割り当て)

---

#### 注記

軸の割り当ての変更は、次の NC の電源投入後に有効となります。

軸が割り当てられていないスイッチペアは有効になりません。

スイッチペアは、一度に 1 つの機械軸にのみ割り当てることができます。

1 つの機械軸に複数のスイッチペアを定義できます。

---

## 10.2.4 リード時間/遅延時間(ダイナミックスイッチ)

### 機能

発生した遅延を補正するために、ポジションスイッチ信号出力のプラス側とマイナス側のスイッチのそれぞれに追加操作をおこなって、2 つのリード時間または遅延時間を割り当てることができます。

2 つのリード時間、または遅延時間は、マシンデータとセッティングデータに入力します。

---

#### 注記

負の時間値を入力すると、ポジションスイッチ信号の出力が遅延します。

---

### マシンデータのリード時間または遅延時間

1 番目のリード時間、または遅延時間は、次の一般マシンデータに入力します。

- MD10460 SW\_CAM\_MINUS\_LEAD\_TIME[n] (マイナス側ポジションスイッチのリード時間または遅延時間)
- MD10461 SW\_CAM\_PLUS\_LEAD\_TIME[n] (プラス側ポジションスイッチのリード時間または遅延時間)

たとえば、次のデータをこのマシンデータに入力できます。

- 現在値の検知とポジションスイッチ信号出力との間の一定の内部遅延時間(オシロスコープで計測されたものなど)
- 一定の外部遅延時間

### セッティングデータのリード時間または遅延時間

2 番目のリード時間、または遅延時間は、次の一般セッティングデータに入力します。

- SD41520 SW\_CAM\_MINUS\_TIME\_TAB\_1[n] (マイナス側ポジションスイッチ1～8のリード時間または遅延時間)
- SD41521 SW\_CAM\_PLUS\_TIME\_TAB\_1[n] (プラス側ポジションスイッチ1～8のリード時間または遅延時間)
- SD41522 SW\_CAM\_MINUS\_TIME\_TAB\_2[n] (マイナス側ポジションスイッチ9～16のリード時間または遅延時間)
- SD41523 SW\_CAM\_PLUS\_TIME\_TAB\_2[n] (プラス側ポジションスイッチ9～16のリード時間または遅延時間)
- SD41524 SW\_CAM\_MINUS\_TIME\_TAB\_3[n] (マイナス側ポジションスイッチ17～24のリード時間または遅延時間)
- SD41525 SW\_CAM\_PLUS\_TIME\_TAB\_3[n] (プラス側スイッチのリード時間または遅延時間 17～24)
- SD41526 SW\_CAM\_MINUS\_TIME\_TAB\_4[n] (マイナス側ポジションスイッチ25～32のリード時間または遅延時間)
- SD41527 SW\_CAM\_PLUS\_TIME\_TAB\_4[n] (プラス側ポジションスイッチ25～32のリード時間または遅延時間)

加工中に変わる場合のある遅延時間などは、このセッティングデータに入力してください。

### 10.3 ポジションスイッチ信号の出力

## 10.3 ポジションスイッチ信号の出力

### 10.3.1 有効化

スイッチの状態(ポジションスイッチ信号)は、PLC だけでなく、NC I/O にも出力可能です。

#### ポジションスイッチ信号出力の適用

軸のポジションスイッチ信号の出力は、NC/PLC インタフェース信号によって有効になります。

DB31、... DBX2.0 (スイッチの起動)

#### PLC へのアンサーバック信号

軸の全スイッチの起動が成功すると、次の NC/PLC インタフェース信号を使用して PLC に返信されます。

DB31、... DBX62.0 (有効なスイッチ)

---

#### 注記

PLC ユーザは、この適用を他の条件(原点確立軸、RESET 有効など)とリンクすることができます。

---

### 10.3.2 PLC へのポジションスイッチ信号の出力

#### PLC への出力

動作中のポジションスイッチを備えたすべての機械軸のポジションスイッチ信号状態は、PLC に出力されます。

状態は IPO サイクルで出力され、非同期で PLC に伝送されます。

### マイナス側スイッチ信号

マイナス側ポジションスイッチ信号の状態は、次の NC/PLC インタフェース信号に入力されます。

DB10 DBX110.0 ~ 113.7 (マイナス側スイッチ信号 1 ~ 32)

### プラス側スイッチ信号

プラス側ポジションスイッチ信号の状態は、次の NC/PLC インタフェース信号に入力されます。

DB10 DBX114.0 ~ 117.7 (プラス側スイッチ信号 1 ~ 32)

---

#### 注記

検出器が選択されていない、または NC/PLC インタフェース信号 DB31、... DBX2.0 (スイッチの起動)が 0 に設定されている場合は、次の NC/PLC インタフェース信号も 0 に設定されます。

- DB10 DBX110.0 ~ 113.7 (マイナス側スイッチ信号 1 ~ 32)
  - DB10 DBX114.0 ~ 117.7 (プラス側スイッチ信号 1 ~ 32)
  - DB31、... DBX62.0 (有効なスイッチ)
- 

## 10.3.3 位置制御周期での NCK I/O へのポジションスイッチ信号の出力

### 位置制御周期での信号出力

マシンデータ MD10470 ~ MD10473 によって HW バイトに割り当てられたスイッチについては(「ハードウェアの割り当て」の章を参照してください)、信号が位置制御周期で出力されます。

NCU 上の 4 つのオンボード出力部および合計 32 個の外部 NC 出力部(オプション)が、NC I/O のデジタル出力部として使用可能です(「A4:SINUMERIK 840D sl 用デジタルとアナログ NC I/O (ページ 33)」の章を参照してください)。

### 10.3 ポジションスイッチ信号の出力

#### ハードウェア割り当て

使用されるハードウェアバイトへの割り当ては、8組のスイッチペア毎に、次の一般マシンデータでおこなわれます。

- MD10470 SW\_CAM\_ASSIGN\_FASTOUT\_1      スイッチ 1～8 の出力の NC I/O へのハードウェア割り当て
- MD10471 SW\_CAM\_ASSIGN\_FASTOUT\_2      スイッチ 9～16 の出力の NC I/O へのハードウェア割り当て
- MD10472 SW\_CAM\_ASSIGN\_FASTOUT\_3      スイッチ 17～24 の出力の NC I/O へのハードウェア割り当て
- MD10473 SW\_CAM\_ASSIGN\_FASTOUT\_4      スイッチ 25～32 の出力の NC I/O へのハードウェア割り当て

---

#### 注記

各マシンデータで、8個のマイナス側スイッチ信号の出力に対して1個のハードウェアバイトを、8個のプラス側スイッチ信号の出力に対して1個のハードウェアバイトを定義できます。

また、2つのマシンデータで、ポジションスイッチ信号の出力を反転できます。

2番目のバイトが指定されていない(=「0」)場合は、8個のスイッチが、マイナス側とプラス側のポジションスイッチ信号の論理的な操作として、1番目のハードウェアバイトを使用して1番目の反転マスクによって出力されます。

---

#### パートプログラムで状態確認

ハードウェア出力の状態は、パートプログラムで実行プログラム変数\$A\_OUT[n] (n = 出力ビット数)を使って読み込むことができます。

#### 切り替え精度

信号は、位置制御周期で NC I/O またはオンボードの出力部に出力されます。位置制御周期のタイムスケールにより、ポジションスイッチ信号の切り替え精度は速度に応じて制限されます。

これは下記のように適用されます。

デルタ位置 =  $V_{act} \times$  位置制御周期

意味: デルタ位置 = 切り替え精度(位置制御周期により異なります)

$V_{act}$  = 現在の軸速度

例:

$V_{act}$  = 20 m/min、PC 周期 = 4 ms

デルタ位置 = 1.33 mm

$V_{act}$  = 2000 rpm、PC 周期 = 2 ms

デルタ位置 = 24°

### 10.3.4 割り込み型ポジションスイッチ信号出力

#### 割り込み制御の出力

位置制御周期とは別に、タイマ割り込みを使用してポジションスイッチ信号を出力すると、精度を大幅に向上させることができます。

4つのスイッチペアの4つのNCUオンボード出力に対する割り込み制御出力を選択するために、次のマシンデータを使用できます。

MD10480 SW\_CAM\_TIMER\_FASTOUT\_MASK (NCU上のタイマー割り込みによるポジションスイッチ信号の出力のビットパターン)

この場合、スイッチペアのマイナスとプラスの信号が、1つの信号として論理的に組み合わせられます。

#### 信号の生成

上記では、論理的に組み合わせられた信号を生成する方法を指定する必要がありました。これは、次のマシンデータのビット1を使用して設定します。

10.3 ポジションスイッチ信号の出力

MD10485 SW\_CAM\_MODE (ポジションスイッチの特性)

ビット	規格値	信号の生成
1	0	プラス側スイッチ - マイナス側スイッチ $\geq 180^\circ$ の場合に、プラス側スイッチの信号特性を反転する。
	1	以下の場合には、プラス側ポジションスイッチの信号の特性は反転されません。 プラス側スイッチ - マイナス側スイッチ $\geq 180^\circ$

注記

この機能は、MD10470 ... MD10473 で選択された HW 割り当てとは無関係に動作します。

オンボードバイトは、複数回使用することはできません。

制限事項

次の制限事項が、相互の動作位置に適用されます。

補間周期毎に **1 つ** の割り込み型出力のみが発生します。

複数組のスイッチペアの信号が同じ補間周期で有効な場合は、出力が次のように優先されます。

最も小さい番号(1...32)のスイッチペアで、**すべての**動作中の信号の出力時間が決まります。つまり、その他のスイッチペアの信号が同時に変化します。

PLC インタフェース

オンボード出力とプラス側およびマイナス側スイッチの状態の **NC** イメージが PLC インタフェースに表示されます。

ただし、次の節で説明するように、これらの信号は**割り込み型**のポジションスイッチ出力タイプと無関係、またはこのために不正確です。プラス側とマイナス側のスイッチの信号は、補間器クロック周期と同期して(**1 度だけ**)生成され、一緒に PLC に伝送されません。

このため、**1** 補間器クロック周期より短いパルスは PLC 上では表示されません。

オンボード出力部は、割り込み毎に非同期で補間器クロック周期に設定およびリセット

されます。オンボード出力部の状態が検出され、PLC インタフェースの更新時間と同期して PLC に伝送されます。

PLC インタフェースが更新されたときの現在の状態に応じて、1 補間器クロック周期より短いパルスは表示されないか、複数の IPO サイクルに拡大されて表示されます。

### その他の設定内容

ここで記述された特性を有効にする場合、次のビットを「0」に設定してください。

MD10485 SW\_CAM\_MODE ビット 0=0

## 10.3.5 独立した、割り込み型ポジションスイッチ信号出力

### 独立した、割り込み型スイッチ出力

各切り替えエッジは、割り込み制御によって、(補間器クロック周期から)独立したポジションスイッチ出力により、割り込み毎に個別に出力されます。

以下の結果として、ポジションスイッチ信号の相互の影響は適用されなくなります。

- 補間器クロック周期毎の単独の出力である。
- 最も優先順位の高いスイッチペア(最も番号の小さいスイッチペア)で決定される出力時間制御である。

**補間器クロック周期毎に合計 8 つの割り込み型のポジションスイッチ出力を、4 つのオンボード出力部の設定/リセットに対して設定できます。** タイマ制御タイプでは、PLC インタフェースで、標準でプラスおよびマイナス側スイッチの信号状態が利用できます。ただし、これらの信号は妥当ではなく、よってタイマ制御出力では不正確です。

### 信号の生成

上記では、論理的に組み合わされた信号を生成する方法を指定する必要がありました。これは、次のマシンデータのビット 1 を使用して設定します。

10.4 位置タイマスイッチ

MD10485 SW\_CAM\_MODE (ポジションスイッチの特性)

ビット	規格値	信号の生成
1	0	プラス側スイッチ - マイナス側スイッチ $\geq 180^\circ$ の場合に、プラス側スイッチの信号特性を反転する。
	1	以下の場合には、プラス側ポジションスイッチの信号の特性は反転されません。 プラス側スイッチ - マイナス側スイッチ $\geq 180^\circ$

設定

スイッチペアは、次のマシンデータを使用して、オンボード出力部に割り当てられます。

MD10480 SW\_CAM\_TIMER\_FASTOUT\_MASK (NCU 上のタイマー割り込みによるポジションスイッチ信号の出力のビットパターン)

また、このタイプの処理は明示的に選択してください。

MD10485 SW\_CAM\_MODE ビット 0=1

注記

この機能は、MD10470 ... MD10473 で選択された HW 割り当てとは無関係に動作します。

オンボードバイトは、複数回使用することはできません。

10.4 位置タイマスイッチ

位置タイマスイッチ

「位置タイマスイッチ」という用語は、定義された軸位置で特定の持続時間のパルスを出力できるポジションスイッチのペアを意味します。

ソリューション

位置は、ポジションスイッチのペアで定義されます。

パルス継続時間は、プラス側スイッチのリード時間/遅延時間で定義されます。

マシンデータを使用して、「マイナス側スイッチ位置=プラス側スイッチ位置」のスイッチペアが位置タイマスイッチとして処理されるよう指定できます。

### 位置タイマスイッチの機能

- パルス継続時間は、軸速度と移動方向反転とは関係ありません。
- パルス継続時間は、軸位置(事前設定値)を変更しても関係ありません。
- 起動(信号立ち上がり)は、動作位置を横切るときのみです。  
軸位置の移動(PRESET など)で、起動されることはありません。
- リード時間/遅延時間はマイナス側スイッチに対して有効となり、パルスの時間のずれが起きます。
- 起動(ON の立ち上がり)とパルス継続時間は、移動方向とは無関係です。
- スイッチが有効なときに動作位置を再度横切って(方向反転)も、スイッチは無効になりません。
- スイッチ時間(パルス幅)は中断されず、動作位置が再度横切ったときに再開されません。  
この動作は、特にモジュロ軸に関して意味があります。つまり、スイッチ時間がモジュロ範囲を横切る時間より長い場合、スイッチは回転毎に切り替わりません。

### 設定

位置タイマスイッチをプログラミングするには、次のように設定してください。

- **位置**  
位置は、マイナス側スイッチの位置がプラス側スイッチの位置と同じスイッチペアで定義してください。  
これは、次のセッティングデータを使用して定義されます。  
SD41500 SW\_CAM\_MINUS\_POS\_TAB\_1  
...  
SD41507 SW\_CAM\_PLUS\_POS\_TAB\_4.
- **パルス継続時間**  
パルス継続時間は、次のマシンデータでスイッチペアの関連入力値を合計して計算されます。  
MD10461 SW\_CAM\_PLUS\_LEAD\_TIME[n]  
SD41521 SW\_CAM\_PLUS\_TIME\_TAB\_1[n]...  
SD41527 SW\_CAM\_PLUS\_TIME\_TAB\_4[n]

10.6 データリスト

- オフセット  
位置タイマスイッチの時間のずれは、次のマシンデータでスイッチペアの関連入力値を合計して計算されます。  
MD10460 SW\_CAM\_MINUS\_LEAD\_TIME[n]  
SD41520 SW\_CAM\_MINUS\_TIME\_TAB\_1[n]...  
SD41526 SW\_CAM\_MINUS\_TIME\_TAB\_4[n]
- モード  
マシンデータで MD10485 SW\_CAM\_MODE  
ビット 2 = 1 を設定して、マイナス側スイッチとプラス側スイッチの位置に対して同じ値をもつ全てのスイッチペアが、確実に位置タイマスイッチとして取り扱われるようにしてください。

10.5 補足条件

「ポジションスイッチ、ポジションスイッチ信号」機能の適用

この機能はオプションです(「ポジションスイッチ信号/スイッチコントローラ」)。ライセンス管理を通じてハードウェアに割り当ててください。

10.6 データリスト

10.6.1 マシンデータ

10.6.1.1 一般マシンデータ

番号	識別子: \$MN_	説明
10260	CONVERT_SCALING_SYSTEM	基本単位系切り替え有効
10270	POS_TAB_SCALING_SYSTEM	位置決めテーブルの単位系
10450	SW_CAM_ASSIGN_TAB[n]	機械軸へのポジションスイッチの割り当て
10460	SW_CAM_MINUS_LEAD_TIME[n]	マイナス側ポジションスイッチ 1 ~ 16 のリード時間または遅延時間

番号	識別子: \$MN_	説明
10461	SW_CAM_PLUS_LEAD_TIME[n]	プラス側ポジションスイッチ 1 ~ 16 のリード時間または遅延時間
10470	SW_CAM_ASSIGN_FASTOUT_1	スイッチ 1 ~ 8 の出力の NC I/O へのハードウェア割り当て
10471	SW_CAM_ASSIGN_FASTOUT_2	スイッチ 9 ~ 16 の出力の NC I/O へのハードウェア割り当て
10472	SW_CAM_ASSIGN_FASTOUT_3	スイッチ 17 ~ 24 の出力の NC I/O へのハードウェア割り当て
10473	SW_CAM_ASSIGN_FASTOUT_4	スイッチ 25 ~ 32 の出力の NC I/O へのハードウェア割り当て
10480	SW_CAM_TIMER_FASTOUT_MASK	NCU でのタイマ割り込みによるポジションスイッチ信号出力のビットパターン
10485	SW_CAM_MODE	ポジションスイッチの特性

## 10.6.2 セッティングデータ

### 10.6.2.1 一般セッティングデータ

番号	識別子: \$SN_	説明
41500	SW_CAM_MINUS_POS_TAB_1[n]	マイナス側スイッチの位置 1 ~ 8
41501	SW_CAM_PLUS_POS_TAB_1[n]	プラス側スイッチの位置 1 ~ 8
41502	SW_CAM_MINUS_POS_TAB_2[n]	マイナス側スイッチの位置 9 ~ 16
41503	SW_CAM_PLUS_POS_TAB_2[n]	プラス側スイッチの位置 9 ~ 16
41504	SW_CAM_MINUS_POS_TAB_3[n]	マイナス側スイッチの位置 17 ~ 24
41505	SW_CAM_PLUS_POS_TAB_3[n]	プラス側スイッチの位置 17 ~ 24
41506	SW_CAM_MINUS_POS_TAB_4[n]	マイナス側スイッチの位置 25 ~ 32
41507	SW_CAM_PLUS_POS_TAB_4[n]	プラス側スイッチの位置 25 ~ 32

10.6 データリスト

番号	識別子: \$SN_	説明
41520	SW_CAM_MINUS_TIME_TAB_1[n]	マイナス側スイッチのリード時間または遅延時間 1 ~ 8
41521	SW_CAM_PLUS_TIME_TAB_1[n]	プラス側スイッチのリード時間または遅延時間 1 ~ 8
41522	SW_CAM_MINUS_TIME_TAB_2[n]	マイナス側スイッチのリード時間または遅延時間 9 ~ 16
41523	SW_CAM_PLUS_TIME_TAB_2[n]	プラス側スイッチのリード時間または遅延時間 9 ~ 16
41524	SW_CAM_MINUS_TIME_TAB_3[n]	マイナス側スイッチのリード時間または遅延時間 17 ~ 24
41525	SW_CAM_PLUS_TIME_TAB_3[n]	プラス側スイッチのリード時間または遅延時間 17 ~ 24
41526	SW_CAM_MINUS_TIME_TAB_4[n]	マイナス側スイッチのリード時間または遅延時間 25 ~ 32
41527	SW_CAM_PLUS_TIME_TAB_4[n]	プラス側スイッチのリード時間または遅延時間 25 ~ 32

10.6.3 信号

10.6.3.1 軸/主軸への信号

信号名称	SINUMERIK 840D sl	SINUMERIK 828D
スイッチの起動	DB31、... .DBX2.0	-

10.6.3.2 軸/主軸からの信号

信号名称	SINUMERIK 840D sl	SINUMERIK 828D
スイッチの有効化	DB31、... .DBX62.0	-

## N4: 自チャネル - 840D sl のみ

### 11.1 概略説明

#### サブファンクション

パンチングおよびニブリングの操作に固有の機能は、下記で構成されます。

- ストローク制御
- 自動軌跡分割
- 回転パンチとダイ
- クランプ保護

これらは、言語命令によって有効および無効にします。

### 11.2 ストローク制御

#### 11.2.1 一般情報

##### 機能

ストローク制御は、実際のワークの加工で使用されます。パンチは、パンチング位置に達したときに、NC 出力信号によって有効になります。パンチングユニットは、NC への入力信号でパンチング動作を確認します。この期間は、どの軸も動かすことはできません。パンチング運転の後に再位置決めが行われます。

##### 高速信号

「高速信号」は、NC とパンチングユニットの間の直接通信に使用されます。パンチング位置決め時間は加工遅れと解釈されるため、「高速信号」をパンチと組み合わせると、毎分のパンチング加工穴数を、大幅に増やすことができます。

11.2 ストローク制御

PLC 信号

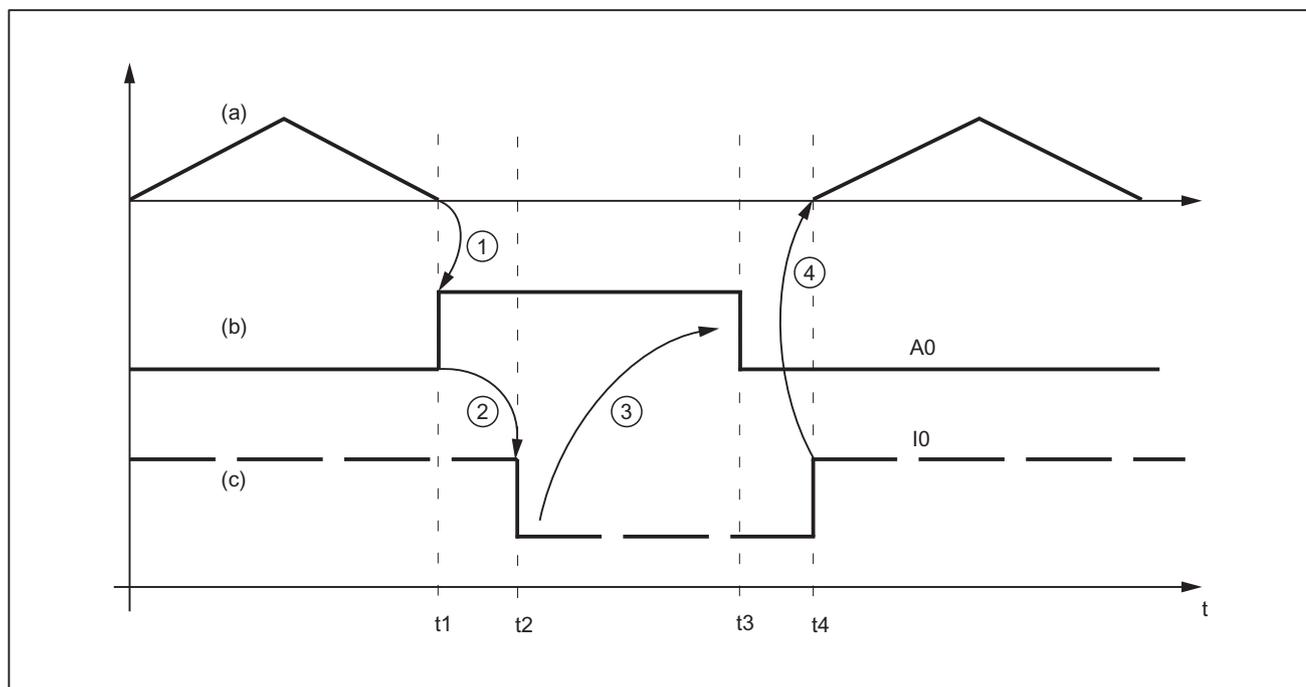
PLC インタフェース信号は、設定や監視などのタイムクリティカルでない機能に使用されます。

11.2.2 高速信号

機能

高速信号は、NC とパンチングユニットの同期に使用されます。一方では、金属板が静止するまでパンチのストロークが開始されないように、高速出力によって制御されます。その上、パンチ実行中は金属板が静止しているように、高速入力によって制御されます。パンチングユニットの駆動には、コントローラの高速度デジタル入出力部が使用されます。

次の信号チャートは、信号シーケンスを示しています。



- (a) 関数  $v(t)$  としての機械の軸移動
- (b) [ストローク開始]信号
- (c) [ストローク有効]信号

図 11-1 信号チャート

## 注記

[ストローク有効]信号は、開回路監視に関連するためハイアクティブです。

パンチングとニブリングの事象の時系列シーケンスは、 $A_0$  と  $E_0$  という 2 つの信号で制御されます。

$A_0$	NC で設定され、ストローク開始と同じです。
$E_0$	パンチングユニットの状態を定義し、[ストローク有効]信号と同じです。

信号ステータスは、以下のように  $t_1 \sim t_4$  の時間を特徴づけて、定義します。

$t_1$	パンチングに関連するワーク(金属板)の移動は、 $t_1$ の時点で完了します。 ストローク開始で定義された条件(「ストローク開始の条件 (ページ 758)」の章を参照してください)に応じて、高速出力 $O_0$ がパンチング開始として設定されます①。
$t_2$	パンチングユニットは高速入力 $E_0$ によって、 $t_2$ の時点でパンチング動作の信号を送ります。 これは、信号 $A_0$ で出力されます②。 安全上の理由により、信号 $E_0$ はハイアクティブです(開回路の場合は[ストローク有効]が常に設定され、軸は動きません)。 [ストローク有効]信号は、工具が金属板から離れる( $t_4$ )までは再度リセットされることはありません。
$t_3$	NC は、[ストローク開始]信号をキャンセルして、 $t_3$ の時点で[ストローク有効]信号に応答します③。これ以降は、NC が待機ステータスになります。NC は[ストローク有効]信号のキャンセルを待ち、キャンセルされると、次の軸移動を開始できます。 次のストロークは、信号 $A_0$ がオフした後にのみ開始できます。
$t_4$	パンチング運転は、 $t_4$ の時点で完了します。つまり、パンチが再度、金属板から離れます。 NC は、軸移動を開始して、信号 $E_0$ の信号切り替えに応答します④。 信号エッジ変化④へのNCの応答については、「パンチング後の軸の開始」の章で説明しています。

11.2 ストローク制御

注記

ストローク時間は、 $\Delta t_h = t_4 - t_1$  の期間で決定されます。

$E_0$  の信号切り替えと軸移動の開始の間で  $t_4$  の時点の応答時間も追加してください。

11.2.3 ストローク開始の条件

ストロークの開始

ストローク開始は、早くとも、軸が確実に停止状態に達したときに設定してください。これにより、パンチングの時点で、加工面のパンチと金属板の間の相対的な移動を確実になくすことができます。

次の図は、ストローク開始に適用可能な、様々な条件を示しています。

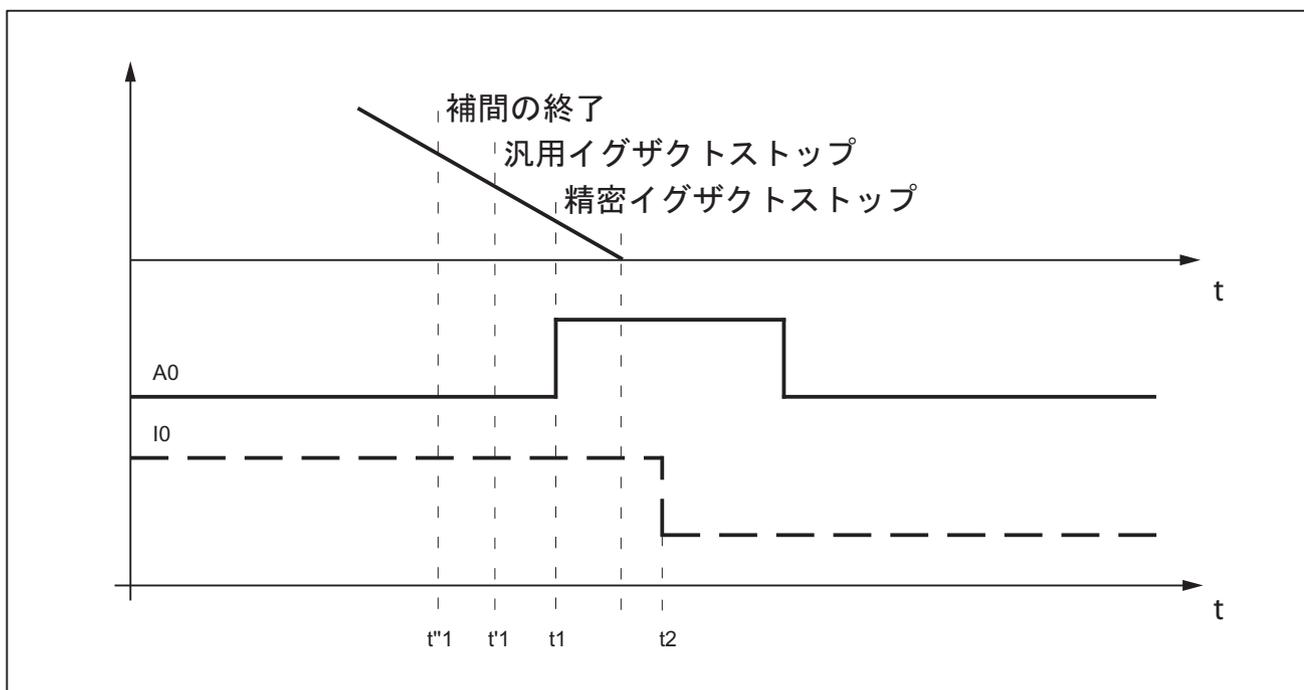


図 11-2 信号チャート: ストローク開始の条件

$t_1$  と  $t_2$  の区間は、出力  $A_0$  の設定に対するパンチングユニットの応答で決定されます。これは変更できませんが、むだ時間を最小にするためのリード時間として利用できます。

上の図は、初期設定の「精密イグザクトストップ範囲」に達したときの出力(G601; Gグループ 12 の初期設定)を示しています。パンチング開始時間  $t''_1$  および  $t'_1$  は、G602 および G603 を使用してプログラム指令されます(下の表を参照してください)。

プログラミング	適用	説明
G603	補間を停止	補間がブロック終点に到達します。この場合、軸は行き過ぎるまで移動し続けます。つまり、軸がゼロ速度に到達する前に、信号が検出可能な区間で出力されます( $t''_1$ を参照)。
G602	汎用インポジション範囲に到達	軸が一度汎用インポジション範囲に到達すると、信号が出力されます。この条件をストローク開始出力に選択すると、ストローク開始点を、補間範囲のサイズによって変更することができます( $t'_1$ を参照)。
G601	精密インポジション範囲に到達	この場合、軸データが適切に設定されていれば、パンチングの時点には必ず、機械が停止状態に達していることを保証できます。ただし、この方式では、むだ時間が最大になるという結果にもなります( $t_1$ を参照)。

#### 注記

G601、G602、および G603 (Gグループ 12)を含む Gグループの初期設定は、次のマシンデータによって定義されます。

**MD20150 \$MC\_GCODE\_RESET\_VALUES[11]**

初期設定は G601 です。

## G603

速度と機械の動的な関係に応じて、補間終点で約 3 ~ 5 補間周期が処理された後、軸がゼロ速度に達します。

マシンデータ **MD26018 \$MC\_NIBBLE\_PRE\_START\_TIME**

と組み合わせると、補間終点への到達と[ストローク ON]の高速出力の設定の区間を遅延させることができ、そのため、このタイミングを最適化できます。

次のセッティングデータは、MD26018 に付加して使用できます。

11.2 ストローク制御

SD42402 \$SC\_NIBPUNCH\_PRE\_START\_TIME

SD42402 はパートプログラムから変更できます。そのため、実行パートプログラムの進行状況に応じて、パンチング処理に適応させることができます。

遅延時間は下記のようになります。

MD26018 = 0 → SD42402 が有効

MD26018 ≠ 0 → MD26018 が有効

「ドウェル時間、PDELAYON のパンチング」が有効な場合は、この機能についてプログラム指令されたドウェル時間が有効となります。この場合は、MD26018 と SD42402 のいずれも無効です。

11.2.4 パンチング後の軸の開始

入力信号[strook ON]

strook開始後の軸移動の開始は、入力信号[strook ON]によって制御されます。

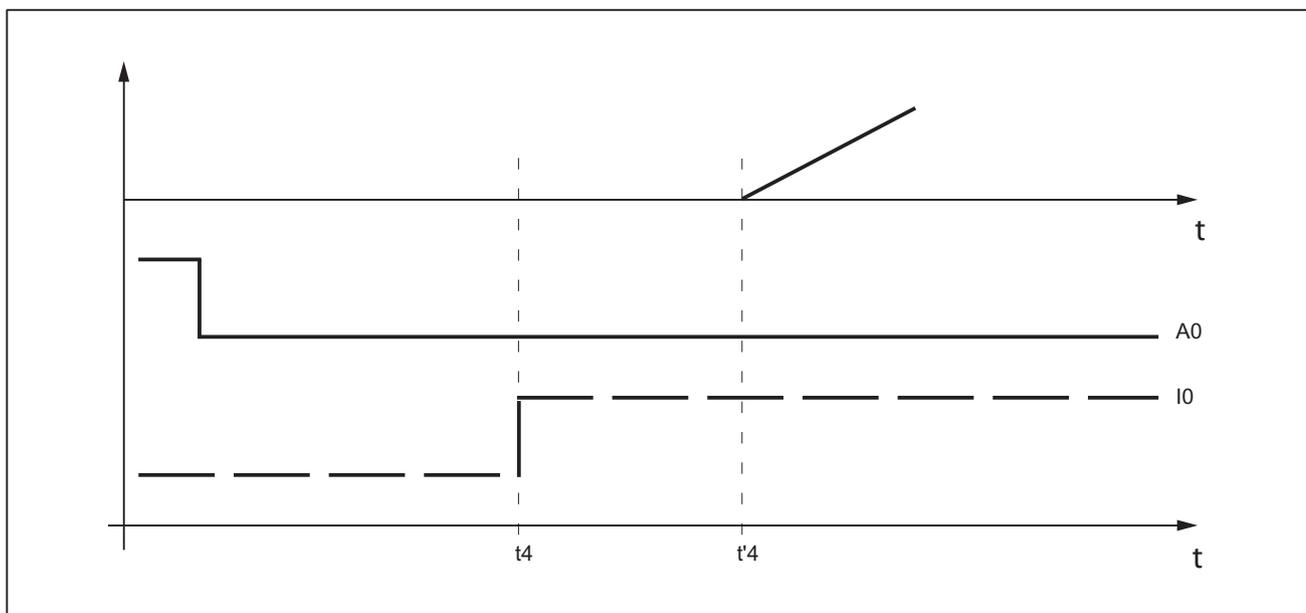


図 11-3 信号チャート:パンチング後の軸の開始

この場合、 $t_4$  と  $t'_4$  の区間は、切り替え時間に依存した応答時間として機能します。これは、補間サンプリング時間およびプログラム指令されたパンチング/ニブリングのモードで決定されます。

## PON/SON

パンチングユニットが PON/SON によって制御されていると、最大遅延時間は次のように計算されます。

$$|t'_4 - t_4| = 3 \times \text{補間器クロック周期}$$

## PONS/SONS

パンチが PONS/SONS で制御されている場合は、遅延時間が次の式で決定されます。

$$|t'_4 - t_4| \leq 3 \times \text{位置制御周期}$$

前提条件ストローク時間( $t_4 - t_2$ ) > 4 補間器クロック周期

### 11.2.5 パンチングおよびニブリングに固有の PLC 信号

#### 機能

直接のストローク制御に使用される信号の他に、チャネル別 PLC インタフェース信号にも使用できます。この信号は、パンチング処理の制御と運転中のステータスの表示の両方に使用されます。

#### 信号

信号	適用
DB21、... DBX3.0 (ストローク無効)	NC がパンチング運転を開始しないようにします。 NC は、有効信号が使用可能になるまで待った後に、パートプログラムを続行します。
DB21、... DBX3.2 (ストローク抑制)	パンチング運転を開始せずに、パートプログラムが処理されるようにします(ドライラン)。 動作中の軌跡分割によって、「停止と進行」モードで軸が移動します。

11.2 ストローク制御

信号	適用
DB21、... DBX3.4 (ストローク遅延)	PDELAYON で許可された場合は、ストローク出力の遅延を有効にします。
DB21、... DBX3.1 (手動ストローク抑制)	オペレータが、パートプログラムを実行せずに、(PLC によって制御された)パンチング運転を開始できるようにします。 手動ストローク開始が、次の信号で確認されます。 DB21、... DBX38.1 (手動ストローク開始の確認応答)

11.2.6 標準 PLC 信号への、パンチングおよびニブリング固有の応答

DB21、... DBX12.3 (送り停止)

インタフェース信号

DB21、... DBX12.3 (送り停止)

が通知されると、NC はストローク制御に関して、次のように応答します。

信号が、 $t_1$ の時点より前に検出された場合:	ストロークは抑制されます。 次のストロークは、次の開始まで、または[送り停止]信号がキャンセルされるまでは開始されません。 加工は、まったく中断されずに続行されます。
信号が、 $t_1$ の時点で検出された場合:	現在のストロークが実行され、完了します。 NC は、 $t_4$ の特性を持つステータスになります。 NC がこのように応答できるように、[ストローク有効]および[ストローク開始]信号のタイマ監視は行われません。

## 11.2.7 信号監視

### 揺動信号

パンチング部の油圧部の経年変化とともに、パンチング部のオーバーシュートにより「ストローク有効」信号がストローク終了時にふらつく場合があります。

この場合、マシンデータ:

MD26020 \$MC\_NIBBLE\_SIGNAL\_CHECK に応じて、アラームが生成される可能性があります(アラーム 22054 「未定義のパンチング信号」)

。

### リセット応答

NC リセットの場合は、高速入力によって、インタフェース信号:

DB21、... DBX38.0 (ストローク開始が有効)

が確認応答なしで直ちにキャンセルされます。

現在動作中のストロークは抑制できません。

## 11.3 適用と解除

### 11.3.1 言語命令

パンチングとニブリングの機能は、設定可能な言語命令によって有効および無効にします。この言語命令は、以前のシステムの特殊 M 機能に代わるものです。

参照先:

『プログラミングマニュアル、上級編』

### グループ

言語命令は、次のグループに分類されます。

グループ 35	
次の言語命令を使用して実際のパンチングとニブリング固有の機能が有効および無効になります。	
PON	= パンチング ON
SON	= ニブリング ON

11.3 適用と解除

グループ 35	
PONS	= パンチング ON、位置コントローラで有効
SONS	= ニブリング ON、位置コントローラで有効
SPOF	= パンチング/ニブリング OFF

グループ 36	
このグループには、準備機能のみを持ち、パンチング機能の性質を決定する命令が含まれます。	
PDELAYON	= 遅延のあるパンチング ON
PDELAYOF	= 遅延のあるパンチング OFF
これらの準備機能に関しては、通常は PLC がいくつかの予備の処理をおこなう必要があるため、予備の処理は、命令の起動前にプログラム指令されます。	

グループ 38	
このグループには、2 番目のパンチインタフェースに切り替えるための命令が含まれます。このグループは、2 番目のパンチングユニットまたはハンマーシャワーのセットなどに使用できます。パンチング機能に使用できる 2 番目の I/O ペアはマシンデータによって定義されます。	
SPIF1	= 1 番目のインタフェースが有効
SPIF2	= 2 番目のインタフェースが有効

注記

G グループのうち、同時に有効にできるのは 1 つの機能のみです(G0、G1、G2、G3 など、相互に排他的な様々な補間モードと同様です)。

SPOF

パンチングとニブリング OFF

SPOF 機能により、すべてのパンチングとニブリング機能が終了します。このステータスの NC は、「ストローク有効」信号にも、パンチングおよびニブリング機能に固有の PLC 信号にも応答しません。

SPOF が 1 ブロック(および、パンチング/ニブリングを SON または PON で有効にしてない場合は、その後のすべてのブロック)で移動指令と共にプログラム指令されている場合

は、機械が、パンチング運転を開始せずに、プログラム指令位置にアプローチします。SPOF は、SON、SONS、PON、および PONS を選択解除し、リセット条件に対応します。

プログラミング例:

プログラムコード	コメント
:	
N20 G90 X100 SON	; ニブリングの起動
N25 X50 SPOF	; ニブリングを無効化、 ; ストロークを開始せずに位置決め
:	

## SON

### ニブリングのオン

SON はニブリング機能を有効にし、G グループ 35 のその他の機能(PON など)を選択解除します。

パンチングとは対照的に、最初のストロークが、起動命令によってブロックの開始点で(つまり、最初の機械動作の前に)実行されます。

SON はモーダル動作をおこないません。つまり、SPOF と PON のいずれかがプログラム指令されるまで、またはプログラムの終点に達するまでは、そのまま有効です。

パンチング軸またはニブリング軸として指定された軸(一般的に動作中の平面の軸)に関する移動情報のないブロックでは、ストロークの開始が抑制されます。それでもストロークの開始が必要な場合は、パンチング/ニブリング軸の 1 軸を、移動軌跡なしでプログラム指令してください。SON を使用した最初のブロックが、上記のタイプの移動情報のないブロックである場合は、このブロックで 1 ストロークのみが実行されます。これは、開始点と終了点が同じであるためです。

プログラミング例:

プログラムコード	コメント
:	
:	
N70 X50 SPOF	; パンチングを開始せずに位置決め
N80 X100 SON	; ニブリングを起動し、移動前 (x=50) および ; プログラムされた移動 (x=100) の最後に ; ストロークを開始
:	
:	

11.3 適用と解除

**SONS**

**ニブリング ON (位置制御周期で)**

SONS は、SON と同じように動作をします。この機能は、位置制御周期で有効にされるため、最適なタイミングのストローク開始と、毎分のパンチング速度の向上が可能となります。

**PON**

**パンチング ON**

PON はパンチング機能を有効にし、SON を無効にします。

PON は、SON と同様にモーダル動作をおこないます。

ただし、SON とは対照的に、ストロークはブロックの終点まで(自動軌跡分割の場合は、分割軌跡の終点まで)は実行されません。PON は、移動情報のないブロックについては、SON と同様に動作します。

プログラミング例:

プログラムコード	コメント
:	
N100 Y30 SPOF	; パンチングを開始せずに位置決め
N110 Y100 PON	; パンチングを有効化し、
	; 位置決め運転 (Y=100) の終点でパンチング開始
:	

**PONS**

**パンチング ON (位置制御周期で)**

PONS は、PON と同じように動作します。説明については、SONS を参照してください。

**PDELAYON**

**遅延のあるパンチング ON**

PDELAYON は準備機能です。これは、PDELAYON が通常は、PON の前にプログラム指令されるという意味です。パンチング部のストロークは、プログラム指令終了位置に達すると、遅延されて出力されます。

遅延時間の秒数は、次のセッティングデータにプログラム指令します。

SD42400 \$SC\_PUNCH\_DWELLTIME

定義された値が、整数の補間器クロック周期に分割できない場合は、分割可能なその次の整数値に丸められます。

この機能はモーダル動作をおこないます。

## PDELAYOF

### 遅延のあるパンチング OFF

PDELAYOF は、遅延のあるパンチング機能を無効にします。つまり、パンチング処理が通常どおりに続行されます。PDELAYON と PDELAYOF は、G グループを形成します。

プログラミング例:

SPIF2 は、2 番目のパンチング部インタフェースを有効にします。つまり、ストロークが、高速 I/O の 2 番目のペアによって制御されます(MD26004 と MD26006、「"チャンネル別マシンデータ (ページ 799)"」の章を参照してください)。

プログラムコード	コメント
:	
N170 PDELAYON X100 SPOF	; パンチングを開始せずに位置決めし、遅延したパンチング開始を起動
:	
:	
N180 X800 PON	; パンチングを起動します。; パンチのストロークは、終了位置に達すると、遅延されて出力されます。
:	
:	
N190 PDELAYOF X700	; 遅延のあるパンチングを無効化し、通常のパンチング開始 ON。プログラム移動指令の終了
:	

## SPIF1

### 1 番目のパンチング部インタフェースの適用

SPIF1 は、1 番目のパンチング部インタフェースを有効にします。つまり、ストロークが、高速 I/O の 1 番目のペアによって制御されます(MD26004 と MD26006、「"チャンネル別マシンデータ (ページ 799)"」の章を参照してください)。

1 番目のパンチング部インタフェースは常に、リセットまたはコントロールシステムの電源投入の後に有効になります。1 つのインタフェースのみが使用される場合は、プログラム指令は不要です。

11.3 適用と解除

SPIF2

2 番目のパンチング部インタフェースの適用

SPIF2 は、2 番目のパンチング部インタフェースを有効にします。つまり、ストロークが、高速 I/O の 2 番目のペアによって制御されます(MD26004 と MD26006、「"チャンネル別マシンデータ (ページ 799)"」の章を参照してください)。

プログラミング例:

プログラムコード	コメント
:	
N170 SPIF1 X100 PON	; ブロックの終点で、1 番目の高速出力でストロークを開始します。[ストローク有効]信号が、1 番目の入力で監視されます。
:	
:	
N180 X800 SPIF2	; 2 番目のストロークを、2 番目の高速出力で開始します。[ストローク有効]信号が、2 番目の入力で監視されます。
:	
:	
N190 SPIF1 X700	; その後のストロークはすべて、1 番目のインタフェースで制御されます。
:	

11.3.2 機能拡張

補助インタフェース

2 番目のパンチングユニット、または代わりにそれと同等の手段を使用する機械は、2 番目の I/O ペアに切り替えることができます。

2 番目の I/O ペアは、次のマシンデータによって定義できます。

MD26004 \$MC\_NIBBLE\_PUNCH\_OUTMASK

MD26006 \$MC\_NIBBLE\_PUNCH\_INMASK

インタフェースは、命令 SPIF1 または SPIF2 で切り替えます。

いずれのインタフェースでも、パンチング/ニブリング機能全体を使用できます。

例:

ストローク制御のためのハードウェア割り当て

CPU では、各々の場合で、高速バイトを高速パンチング部インタフェースとして定義します。

MD26000 \$MC\_PUNCHNIB\_ASSIGN\_FASTIN = 'H00030001' → バイト 1

MD26002 \$MC\_PUNCHNIB\_ASSIGN\_FASTOUT = 'H00000001'

備考:

1 番目と 2 番目のビットは反転されます。

高速入力ビットと高速出力ビットのビットパターン:

MD26004 \$MC_NIBBLE_PUNCH_OUTMASK[0]	= 1	1 番目のインタフェース出力ビット →ビット 1 SPIF1
MD26004 \$MC_NIBBLE_PUNCH_OUTMASK[1]	= 2	2 番目のインタフェース出力ビット →ビット 2 SPIF2
MD26006 \$MC_NIBBLE_PUNCH_INMASK[0]	= 1	1 番目のインタフェース入力ビット →ビット 1 SPIF1
MD26006 \$MC_NIBBLE_PUNCH_INMASK[1]	= 2	2 番目のインタフェース入力ビット →ビット 2 SPIF2

### 自動的に有効になる事前開始のタイミング

軸の補間範囲に達する前にストロークを開始できれば、パンチングユニットの応答時間によるむだ時間を最小限にできます。このための基準時間は、補間の終点です。ストロークは、G603 で自動的に開始され、その補間終点に達する時間が、設定された値だけ遅延されます。

ストローク開始の遅延時間は、次のマシンデータで調整できます。

MD26018 \$MC\_NIBBLE\_PRE\_START\_TIME

例:

5 ms の IPO サイクルの場合は、ストロークが補間終点に達した 2 周期後に解除されません。

⇒ MD26018 \$MC\_NIBBLE\_PRE\_START\_TIME = 0.01 [s]

事前開始のタイミングは、セッティングデータにプログラム指令することもできます。

SD42402 \$SC\_NIBPUNCH\_PRE\_START\_TIME

### 11.3 適用と解除

この設定は、MD26018 = 0 の設定時にのみ有効となります。

#### 入力信号の監視

プランジャのオーバーシュートなどのために[ストローク有効]信号がストローク間で変動する場合は、メッセージ[未定義のパンチング信号]も、補間の停止時に出力される場合があります。

出力されるメッセージは、次のマシンデータの設定によって決まります。

#### MD26020 \$MC\_NIBBLE\_SIGNAL\_CHECK

MD26020 = 0	アラームなし
MD26020 = 1	アラーム

#### 2 ストローク間の最短区間

連続する 2 ストローク間の最短時間間隔は、次のセッティングデータにプログラム指令できます。

#### SD42404 \$SC\_MINTIME\_BETWEEN\_STROKES

例:

2 つのストローク開始間隔は、物理的距離に関係なく、少なくとも 1.3 秒の最小遅延時間が必要です。

⇒ SD42404 \$SC\_MINTIME\_BETWEEN\_STROKES = 1.3 [s]

パンチングドウェル時間(PDELAYON)も一緒にプログラム指令されている場合は、この 2 倍の時間が加算して適用されます。

G603 で事前開始のタイミングがプログラム指令されている場合は、SD 42404 で設定された時間以前に補間終点に達した場合にのみ有効となります。

プログラム指令時間は、直ちに有効となります。ブロックバッファのサイズに応じて、すでにプログラム指令されているストロークが、この最短時間間隔で実行される可能性があります。これは、次のプログラミングによる処置(例)で防ぐことができます。

#### プログラムコード

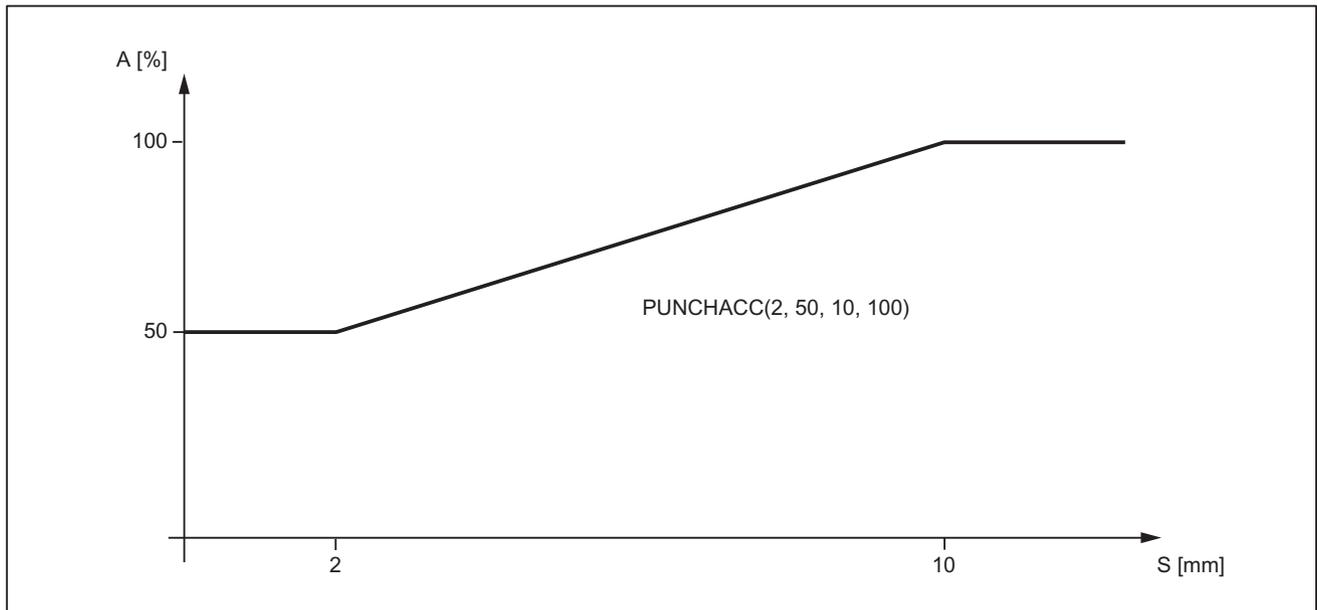
```
N...
N100 STOPRE
N110 $SC_MINTIME_BETWEEN_STROKES = 1.3
```

この機能は、SD42404 = 0 のときは有効になりません。

### 移動に依存した加速度

加速特性は、PUNCHACC (Smin、Amin、Smax、Amax) で定義できます。この命令を使用して、穴間の距離に応じてそれぞれの加速度を定義できます。

#### 例 1:

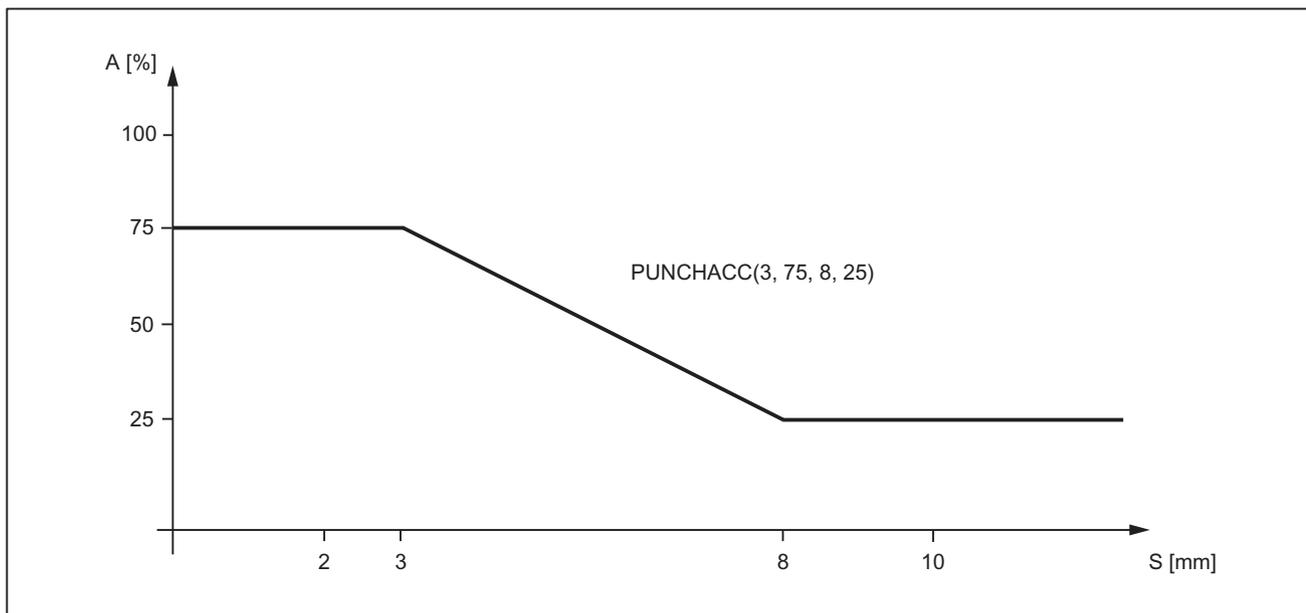


この特性で、次の加速度が定義されます。

穴間の距離	加速度
< 2 mm	軸は、最大加速度の 50 % のレートで加速します。
2 ~ 10 mm	加速度は間隔に比例して、100 % に増加します。
> 10 mm	軸は、最大レート(100%)で加速します。

#### 例 2:

11.3 適用と解除



この特性で、次の加速度が定義されます。

穴間の距離	加速度
< 3 mm	軸は、最大加速度の 75 % のレートで加速します。
3 ~ 8 mm	加速度は間隔に比例して、25 % に減少します。
> 10 mm	軸は、最大レート(25%)で加速します。

加速度でい減レートがすでに ACC でプログラム指令されている場合、PUNCHACC によって定義された加速制限は、加速度でい減レートを示します。

この機能は、次の式で選択解除されます。

$$S_{\min} = S_{\max} = 0$$

事前に ACC でプログラム指令された加速度は、そのまま有効です。

ブロック検索

ニブリング機能を含むブロックを検索する場合は、パンチング部のストロークがブロックの始点で実行されるか、抑制されるかをプログラミングできます。

この設定は、次のマシンデータでプログラム指令します。

## MD11450 \$MN\_SEARCH\_RUN\_MODE

ビット	規格値	意味
5	0	ブロックの始点のパンチング部のストロークが抑制されます。
	1	ブロックの始点のパンチング部のストロークが実行されます。

## 参照先:

『機能マニュアル、基本機能』; 「モードグループ、チャネル、プログラム運転、リセット応答 (K1)」:

### 11.3.3 以前のシステムとの互換性の章

#### M 機能の使用

以前のバージョンのように、マクロ指令により、言語命令の代わりに特殊な M 機能を使用できます(互換性)。

以前のシステムで使用されていた M 機能と同等の言語命令は、次のとおりです。

M20、M23	△	SPOF
M22	△	SON
M25	△	PON
M26	△	PDELAYON

#### 注記

M 機能は、次のマシンデータで設定できます。

M 機能が言語命令に割り当てられている場合、M 機能は補助機能グループにまとめられることに注意してください。

11.3 適用と解除

例

```

DEFINE M20 AS SPOF                パンチング/ニブリング OFF
または
DEFINE M20 AS SPOF M=20          補助機能出力によるパンチング

DEFINE M20 AS SPOF PDELAYOF       パンチング/ニブリング OFF と遅延のあるパンチン
                                  グ OFF

DEFINE M22 AS SON                ニブリング ON
または
DEFINE M22 AS SON M=22          補助機能出力によるニブリング ON

DEFINE M25 AS PON                パンチング ON
または
DEFINE M25 AS PON M=25          補助機能出力によるパンチング ON

DEFINE M26 AS PDELAYON           遅延のあるパンチング ON
または
DEFINE M26 AS PDELAYON M=26     パンチングと補助機能出力
    
```

プログラミング例:

プログラムコード	コメント
:	
:	
N100 X100 M20	; パンチングを開始せずに位置決め
N110 X120 M22	; 移動ストローク開始の前後に ; ニブリングを起動
:	
N120 X150 Y150 M25	; パンチングを有効にし、移動終了後に ; ストロークを開始
:	
:	

## 11.4 自動軌跡分割

### 11.4.1 一般情報

#### 機能

次の2つ方法のいずれかを使用して、プログラム指令された移動軌跡を自動的に分割できます。

- 言語命令 SPP によってプログラム指令された最大分割で軌跡分割します。
- 言語命令 SPN によってプログラム指令された分割数で軌跡分割します。

いずれの機能でも、サブブロックが独立して生成されます。

以前のシステムでは、次のようになっていました。

- SPP<番号>は E<番号>に対応
- SPN<番号>は H<番号>に対応

現在では、アドレス E および H が補助機能を表すので、重複を避けるために言語命令 SPP および SPN が使用されます。したがって、新しい手順は、以前のシステムで使用されていた手順とは互換性がありません。両方の言語命令(SPP および SPN)とも、設定が可能です。

---

#### 注記

SPP のプログラム指令値は、初期設定に応じて、mm または inch による設定です(軸に類似しています)。

自動軌跡分割機能により、軌跡は、直線および円弧補間を、確実に等距離の区間に分割します。

プログラムが中断され、自動軌跡分割が有効な場合(SPP/SPN)に、輪郭を再入力できるのは、分割ブロックの始点のみです。最初のパンチング部のストロークは、このサブブロックの終点で実行されます。

SPP および SPN は、ジオメトリ軸が設定されている場合にのみ有効にすることができます。

---

#### SPP

自動軌跡分割機能 SPP はプログラム指令された移動軌跡を、分割指定に従って、同一サイズの区間に分割します。

## 11.4 自動軌跡分割

次の条件が適用されます。

- 軌跡分割は、SON または PON が有効なときにのみ有効になります。  
(例外: MD26014 \$MC\_PUNCH\_PATH\_SPLITTING = 1)
- SPP はモーダルで有効です。つまり、プログラム指令された分割軌跡は、再度プログラム指令されるまでは有効ですが、ブロック毎(ノンモーダル)に SPN を使用して分割軌跡を抑制できます。
- 分割後の軌跡は、プログラム指令距離の合計が分割後の軌跡区間の整数倍になるように、必要に応じて、コントロールシステムで四捨五入されます。
- 分割後の軌跡の単位は mm/ストロークまたは inch/ストロークです(軸の設定により異なります)。
- プログラム指令 SPP 値が移動距離より大きい場合は、軸の位置が、軌跡分割なしのプログラム指令された終了位置に位置決めされます。
- SPP = 0、リセット、またはプログラムの終了により、プログラム指令 SPP 値は削除されます。SPP 値は、パンチング/ニブリングが無効の場合は削除されません。

### SPN

自動軌跡分割機能 SPN は移動軌跡を、プログラム指令された分割軌跡数に分割します。

次の条件が適用されます。

- 軌跡分割は、SON または PON が有効なときにのみ有効になります。  
(例外: MD26014 \$MC\_PUNCH\_PATH\_SPLITTING = 1)
- SPN はノンモーダル動作を行います。
- 以前にプログラム指令 SPP 値はどれも、SPN を含むブロックでは抑制されますが、次のブロックでは再び有効になります。

### 補足条件

- 軌跡分割機能は、直線補間および円弧補間で有効です。  
補間モードは変更されません。つまり、円弧補間が選択されると、円弧で移動します。
- ブロックに SPN (ストローク数)と SPP (ストローク軌跡)の両方が含まれる場合は、現在のブロックでブロック数が有効になり、それに続くすべてのブロックではストローク軌跡が有効になります。

- 軌跡分割は、パンチングまたはニブリング機能と組み合わせた場合にのみ有効になります。  
(例外: MD26014 \$MC\_PUNCH\_PATH\_SPLITTING = 1)
- プログラム指令された補助機能はどれも、最初のサブブロックの前または間、または最後のサブブロックの後で出力されます。
- 移動情報のないブロックの場合は、SON および PON のプログラミングを規定する規則が、SPP および SPN にも適用されます。言い換えると、ストロークは、軸移動がプログラム指令されている場合にのみ開始されます。

## 11.4.2 軌跡軸の動作特性

### MD26010

マシンデータ

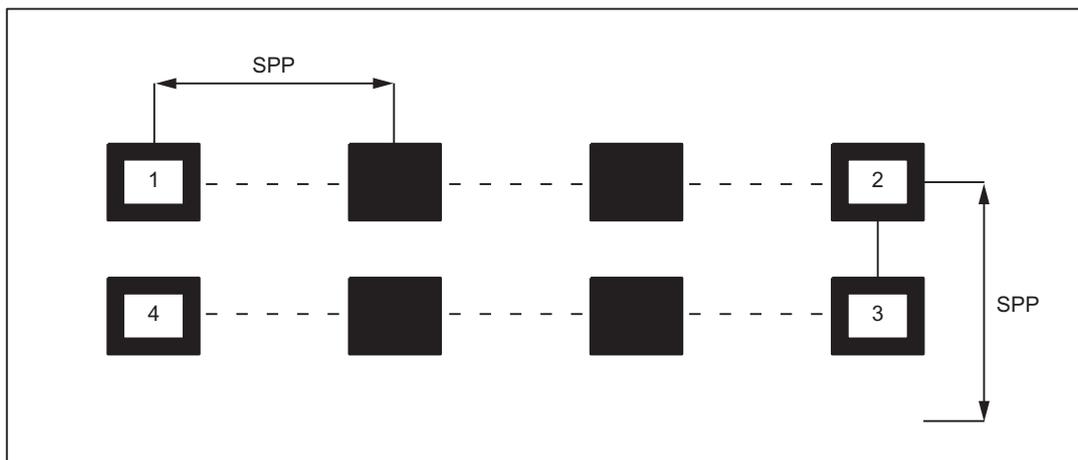
:MD26010 \$MC\_PUNCHNIB\_AXIS\_MASK

によって定義され、プログラム指令されたすべての軸は、SPP と SPN で、同じサイズの軌跡区間に沿って、プログラム指令終点に達するまで移動します。プログラム指令されている場合は、回転工具軸にもこれが適用されます。単独軸に対しては、動作を調整できます。

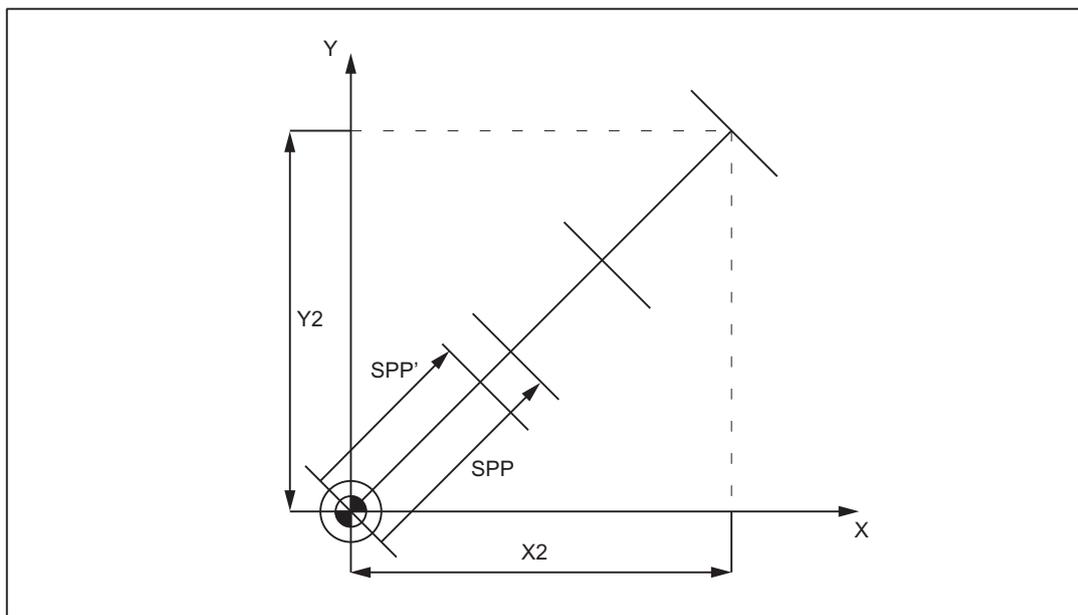
### SPP の例

プログラムコード	コメント
N1 G01 X0 Y0 SPOF	; パンチングを開始しない位置決め
N2 X75 SPP=25 SON	; 送り値 25mm を含むニブル、1 番目の移動の前と ; 各分割軌跡の後に ; パンチングを開始。
:	
:	
N3 Y10	; 移動距離が SPP 値のため、
:	; SPP 値を減らして位置決めし、
:	; 移動後にパンチングを開始。
:	
:	
N4 X0	各軌跡区間の後で パンチングを開始して再位置決め。
:	

11.4 自動軌跡分割



プログラム指令された軌跡分割が軌跡の合計の整数倍でない場合は、送り軌跡が削減されます。



- X2/Y2: プログラム指令された移動距離
- SPP: プログラム指令 SPP 値
- SPP': 自動的に四捨五入されたオフセット距離

図 11-4 軌跡分割

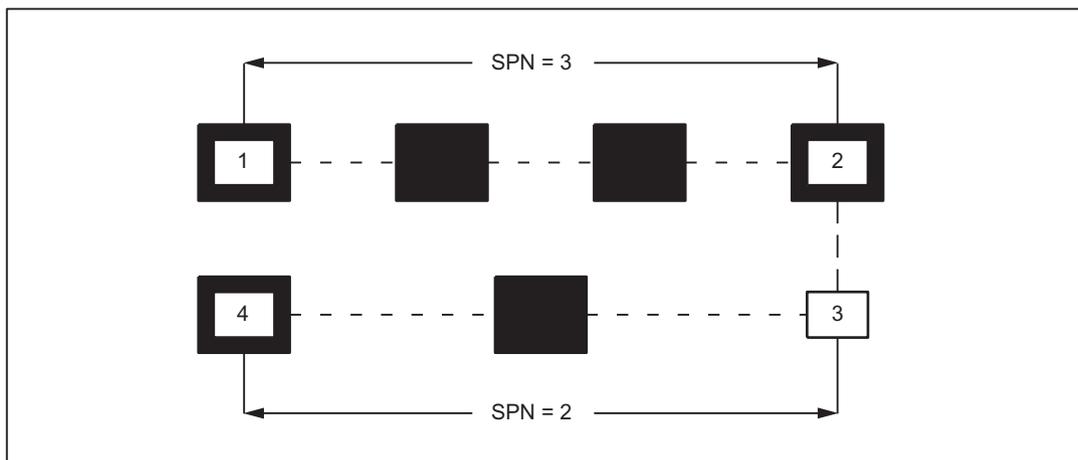
## SPN の例

ブロック毎の分割軌跡の数は、SPN によってプログラム指令されます。

SPN によるプログラム指令値は、パンチングとニブリングの両方の用途にノンモーダルで有効となります。2つのモードが異なっているのは、最初のストロークに関してのみです。通常は、ニブリング運転では、最初のストロークが最初の分割軌跡の始点で実行され、パンチング運転では終点で実行されます。これは、 $n$  個の分割軌跡がプログラム指令されていると、パンチング運転では  $n$  個のストロークが実行されるが、ニブリング運転では  $n+1$  個のストロークが実行されるという意味です。さらに、有効な移動情報がない場合は、複数のストロークがプログラム指令されていても、実行されるのは 1つのストロークのみです。複数のストロークを 1つの位置で生成する必要がある場合は、移動情報のない、対応する数のブロックをプログラム指令してください。

プログラムコード	コメント
N1 G01 X0 Y0 SPOF	; パンチングを開始しない位置決め
N2 X75 SPN=3 SON	; ニブリングの起動。軌跡全体が 3 個の分割軌跡に ; 分割されます。1 番目の移動の前、および各セグメントの ; 終点で、ストロークを ; 開始します。
:	
:	
:	
:	
N3 Y10 SPOF	; パンチングを開始しない位置決め
N4 X0 SPN=2 PON	; パンチングを起動します。軌跡全体が 2 個の分割軌跡 ; に分割されます。パンチングが有効なため、 ; 1 番目のストロークが 1 番目の分割軌跡の終点で ; 開始されます。
:	
:	
:	
:	

11.4 自動軌跡分割



例

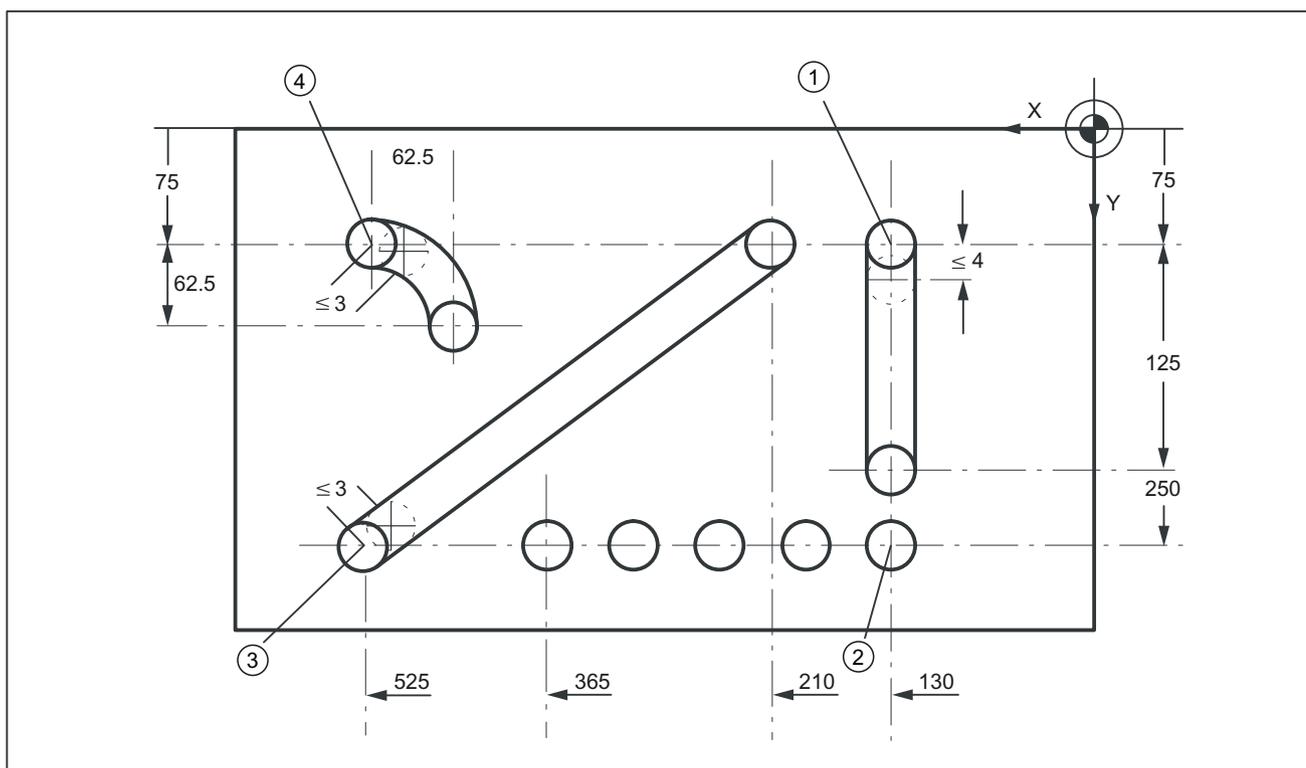


図 11-5 ワーク

プログラムからの抜粋

プログラムコード	コメント
N100 G90 X130 Y75 F60 SPOF	; 縦方向ニブリングの分割軌跡の起点 (1)

プログラムコード	コメント
N110 G91 Y125 SPP=4 SON	; への位置決め ; 終点座標 (インクリメンタル) ; 分割軌跡: 4 mm、ニブリングを起動
N120 G90 Y250 SPOF	; アブソリュート指令、 ; 横方向ニブリング分割軌跡の起点 (2) ; への位置決め
N130 X365 SPN=4 SON	; 終点座標、4 つの分割軌跡、 ; ニブリングの起動
N140 X525 SPOF	; 傾斜ニブリング軌跡の起点 (3) ; への位置決め
N150 X210 Y75 SPP=3 SON	; 終点座標、分割軌跡: 3 mm, ; ニブリングの起動
N160 X525 SPOF	; ピッチ円上のニブリング軌跡の起点 (4) ; への位置決め
N170 G02 G91 X-62.5 Y62.5 I0 J62.5 SON	; インクリメンタル円弧補間 ; パラメータの有効化、 ; ニブリング
N180 G00 G90 Y300 SPOF	; 位置決め

### 11.4.3 単独軸についての動作

#### MD26016

軌跡軸に加えてプログラム指令された単独軸の軌跡は、標準では、生成された各中間ブロックに均等に配分されます。

次の例では、付加回転軸 **C** が同期軸として定義されています。

この軸が「パンチング-ニブリング軸」

**MD26010 \$MC\_PUNCHNIB\_AXIS\_MASK = 1,**

として、追加でプログラム指令されている場合は、同期軸の動作がマシンデータ

**MD26016 \$MC\_PUNCH\_PARTITION\_TYPE**

に応じて、変化する場合があります。

**プログラミング例:**

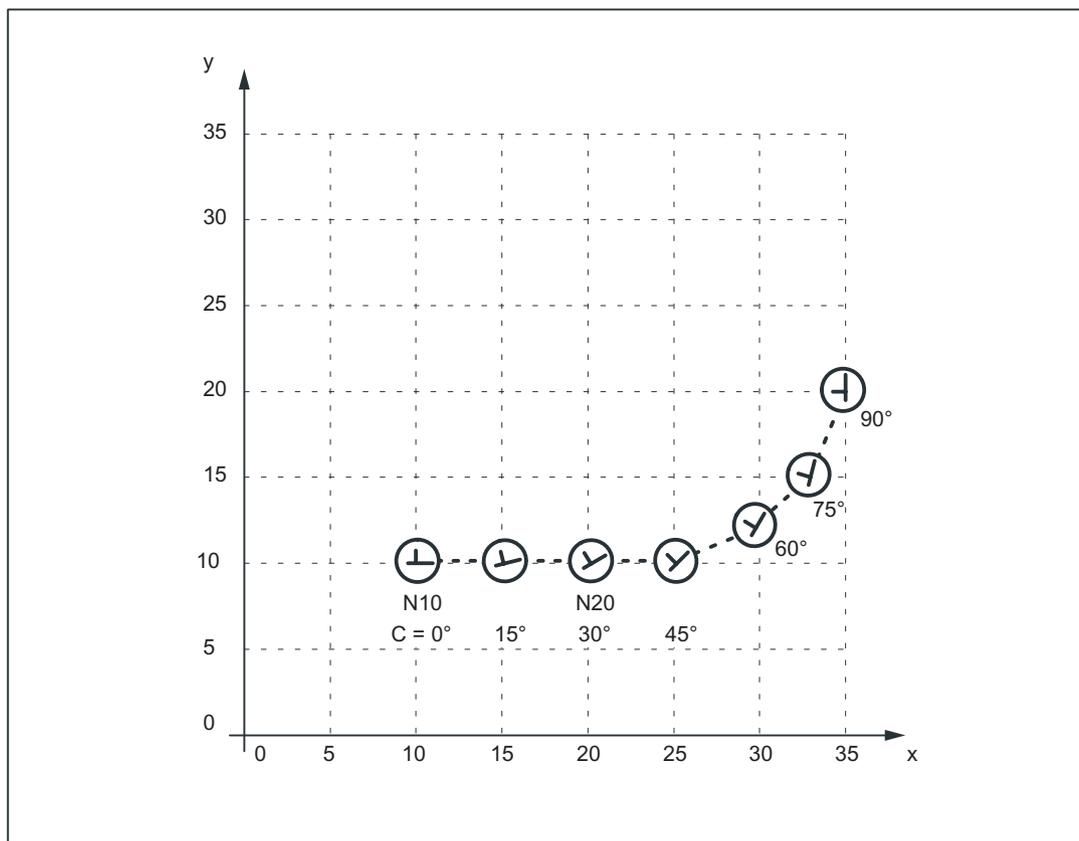
プログラムコード
N10 G90 G1 PON X10 Y10 C0 F10000
N20 SPP=5 X25 C45
N30 G3 SPN=3 X35 Y20 I0 J10 C90

11.4 自動軌跡分割

**MD26016 \$MC\_PUNCH\_PARTITION\_TYPE=0 (初期設定)**

この設定では、軸は標準の動作をします。つまり、プログラム指令された特別な軸移動は、すべての補間モードで動作中の軌跡分割機能の、生成された各中間ブロックに配分されます。

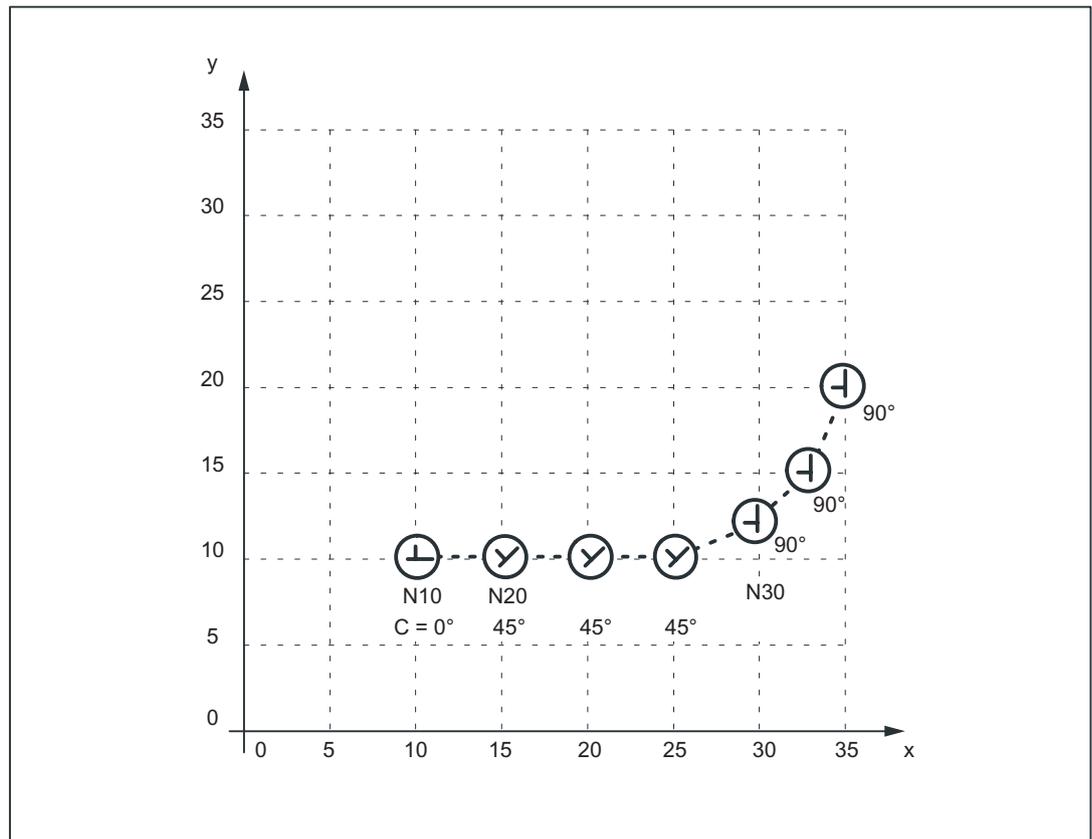
ブロック N20 では、3 個の中間ブロックのそれぞれで C 軸が 15°回転します。ブロック N30 の軸動作は、円弧補間の場合は同じ(3 個のサブブロックのそれぞれで 15°の軸回転)です。



**MD26016 \$MC\_PUNCH\_PARTITION\_TYPE=1**

上記の動作とは対称的に、ここでは同期軸が、選択した軌跡分割機能の最初のサブブロックで、プログラム指令された回転軌跡全体を移動します。

例に適用すると、C 軸は、X 位置が X=15 のときにはすでに、プログラム指令された終点位置 C=45 に到達しています。C 軸は、下の円弧補間ブロックでも同様に動作します。

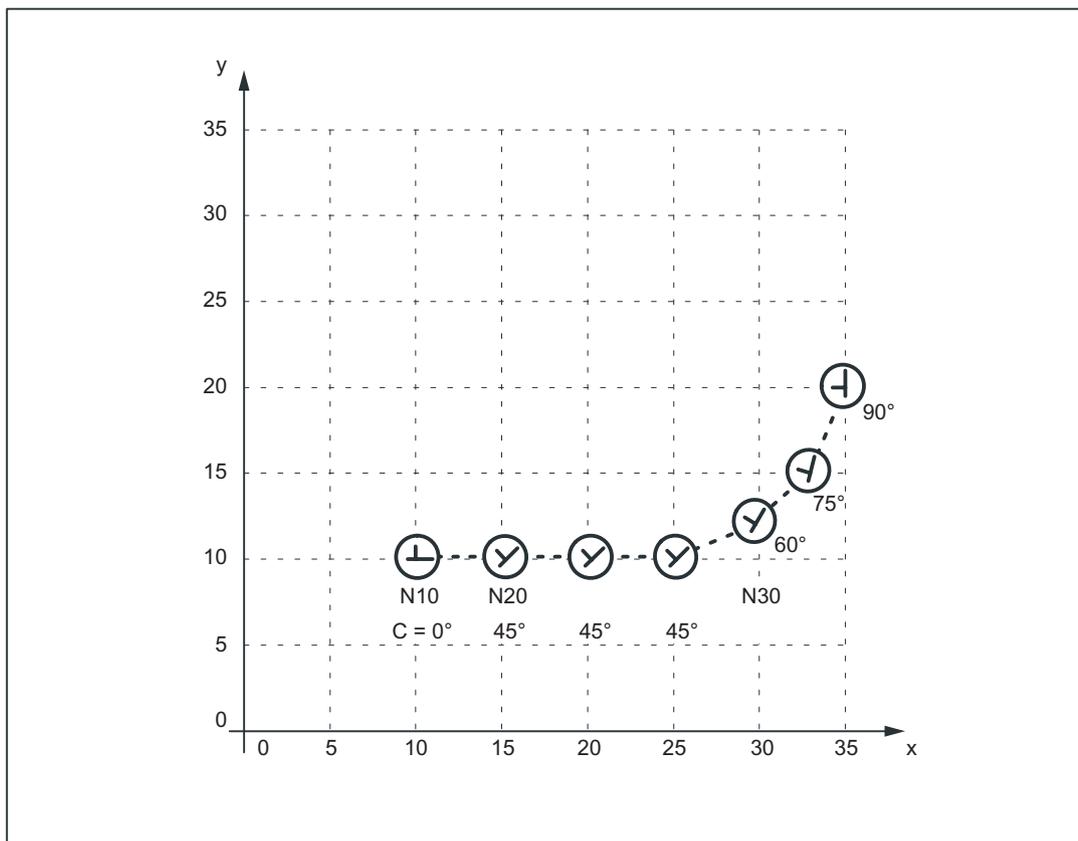


### MD26016 \$MC\_PUNCH\_PARTITION\_TYPE=2

軸が直線補間モードで、上で説明したように動作しなければならないが、円弧補間モードでは初期設定に従って動作しなければならない場合は、**MD26016=2** が設定されています(1番目の事例を参照してください)。

この場合、例にある軸の動作は次のようになります。ブロック N20 では、C 軸が、最初のサブブロックで **C=45°** まで回転します。次の円弧補間ブロックでは、C 軸が各サブブロックで **15°** 回転します。

11.4 自動軌跡分割



上の図が示す軸動作は、回転工具の軸を、輪郭に関して定義した方向(接線方向など)に工具の配置する場合に使用され、そのとき法線方向制御機能を適用できない場合に、特に便利です。ただし、軸動作は、法線方向制御機能の代わりには使用できません。これは、回転軸の開始位置と終了位置が常にプログラム指令されている必要があるためです。

**注記**

付加軸(この場合は回転軸 C)の追加のオフセット移動は、ゼロオフセットによって実行されます。

## 補足条件

- C 軸が「パンチング-ニブリング軸」として定義されていない場合は、上の例のブロック N30 で C 軸の移動軌跡が分割されず、ストロークがブロック終点で開始されません。
- 上で説明した機能が、ニブリングの用途に固有でない、付加軸の位置合わせの方式で実行される場合は、ストローク開始を次の PLC インタフェース信号で抑制できます。  
DB 21, 22 DBX3.2 (ストローク抑制)  
(用途: 溶接中の電子ビームの位置合わせなど)  
次のマシンデータの設定内容で、類似動作をプログラム指令できます。  
MD26014 \$MC\_PUNCH\_PATH\_SPLITTING=1  
この場合は、軌跡が、パンチングまたはニブリング機能に関係なく分割されます。

## 11.5 回転工具

### 11.5.1 一般情報

#### 機能概要

回転パンチとダイを備えたニブリング/パンチング機械のための、次の 2 つの機能が設けられています。

- 連結動作  
パンチとダイの同期回転で実行
- 法線方向制御  
パンチ用回転軸の、ワークに対する通常的位置合わせ

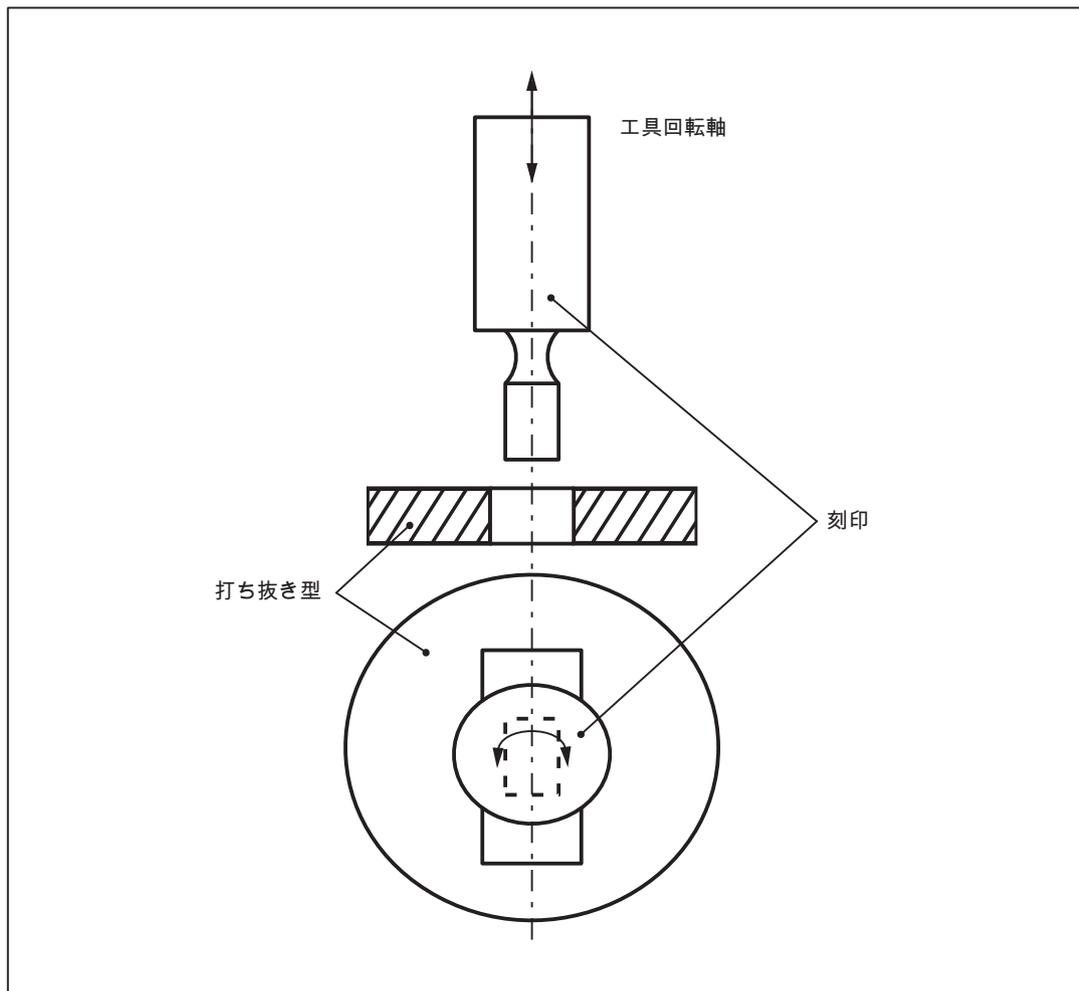


図 11-6 回転工具軸の図

## 11.5.2 パンチとダイの連結動作

### 機能

標準機能の「連結動作」を使用して、ダイの軸を、パンチの回転軸に連結動作軸として割り当てることができます。

### 適用

「連結動作」機能を有効および無効にするにはそれぞれ、言語命令 TRAILON と TRAILOF を使用します。

#### 参照先:

『機能マニュアル、応用機能』; 「軸間連動と ESR (M3)」

### 例

C がパンチ軸で C1 がダイ軸の、回転パンチを備えた一般的なニブリング機械の例:

プログラムコード	コメント
:	
:	
TRAILON (C1, C, 1)	; 連結動作グループを設定
G01 X100 Y100 C0 PON	; C 軸によるストロークの開始/ ; C1 軸位置 C=0=C1
X150 C45	; C 軸によるストロークの開始/ ; C1 軸位置 C=45=C1
:	
:	
M30	

### 基本位置

電源投入後は、有効な連結動作グループはありません。2つの工具軸がレファレンス点にアプローチした後は、通常は連結動作グループは解消されません。

これをおこなう方法は次のとおりです。

- 連結動作グループの単独起動をプログラム指令します(上の例を参照してください)。
- プログラム MD を次のように設定します。

MD20110 \$MC\_RESET\_MODE\_MASK、ビット 8=1

## 11.5 回転工具

こうすると、連結動作グループが、RESET/パートプログラムの起動と終了後も、そのまま有効になります。

### 11.5.3 法線方向制御

#### 機能

「法線方向制御」機能は、プログラムされた軌跡の法線となるようパンチング/ニブリング工具を位置決めするために使われる回転軸を割り出します。法線方向制御に関連して、この軸は法線軸と呼ばれます。加工平面の直線軸は、法線連結の誘導軸です。法線軸は、誘導軸の軌跡の接線方向に割り出された従動軸です。法線軸には、接線角度の追加オフセット角度もプログラムすることができます。

パンチング/ニブリング工具(ダイおよびダイプレート)は、2つの個別の回転軸で移動され、2番目の回転軸は、「連結モーション」で法線軸に結合できます。

---

#### 注記

##### スイッチオンのシーケンス

NC プログラムで、まず法線方向制御を定義し、次に連結モーションを有効化してください。

---

#### 参照先

法線方向制御の詳細については、以下を参照してください。

『機能マニュアル、応用機能』「T3:法線方向制御」の章

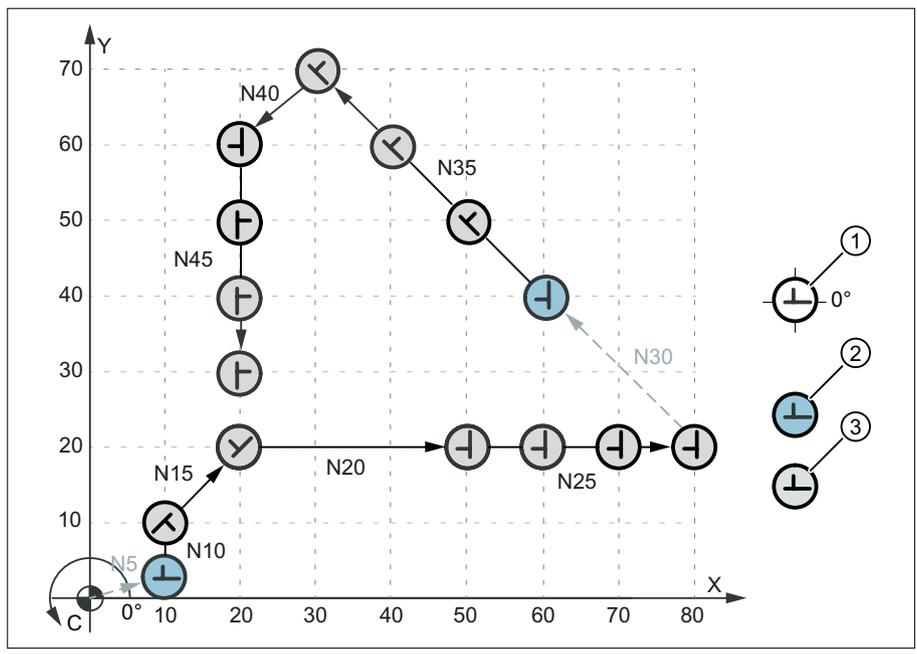
#### 例:直線補間

##### 前提条件

- パンチング/ニブリング工具は、下部および上部の工具に区分されます。
- 下部工具および上部工具は、異なる回転軸(C/C1)で移動します。
- 直線軸 X および Y が誘導軸です(加工平面 G17)

プログラムコード	コメント
N2 TANG (C, X, Y, 1, "B")	; 法線方向制御の定義 ; 誘導軸 X、Y ; 従動軸:C ; 連結係数:1 ; 基本座標系
N5 G0 X10 Y5	; 開始位置
N8 TRAILON (C1, C, 1)	; 連結モーション ON
N10 Y10 C225 PON F60	; C/C1 軸 → 225° ; ニブリング開始 ON、 → ストローク
N15 X20 Y20 C45	; C/C1 軸 → 45° → ストローク
N20 X50 Y20 C90 SPOF	; C/C1 軸 → 90° ; ストローク開始 OFF
N25 X80 Y20 SPP=10 SON	; 軌跡分割:ON ; 分割軌跡:10 mm → 4つのリフト ; パンチ開始:ON
N30 X60 Y40 SPOF	; 位置決め ; ストローク開始:OFF
N32 TANGON (C, 180)	; 法線方向制御:ON ; オフセット角度:180°
N35 X30 Y70 SPN=3 PON	; 軌跡分割:3つの起動セグメント → リフト ; ニブリング開始:ON
N40 G91 C45 X-10 Y-10	; C/C1:90° (-135°+180°+45°) → リフト
N42 TANGON (C, 0)	; 法線方向制御:ON ; オフセット角度:0°
N45 G90 Y30 SPN=3 SON	; 軌跡分割 → 3つのリフト ; パンチ開始:ON
N50 SPOF TANGOF	; ストローク開始 OFF ; 法線方向制御:OFF
N55 TRAILOF (C1, C)	; 連結モーション:OFF
N60 M2	

11.5 回転工具



- ① 挿入ポイント 0°
- ② 位置決め
- ③ ストローク

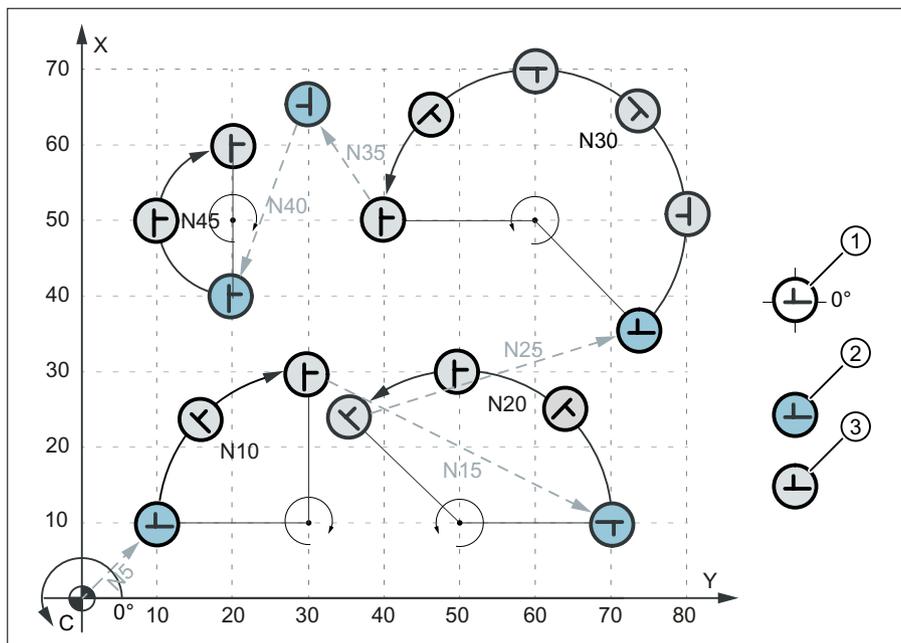
例:円弧補間

円弧補間モードでは、とくに軌跡分割が有効なときに、各サブブロックでプログラム指令された軌跡軸に対して接線方向に位置合わせされた軌跡に沿って、工具軸が回転します。

プログラムコード	コメント
N2 TANG (C, X, Y, 1, "B")	; 法線方向制御の定義 ; 誘導軸 X、Y ; 従動軸:C ; 連結係数:1 ; 基本座標系
N5 G0 F60 X10 Y10	; 開始位置
N8 TRAILON (C1, C, 1)	; 連結モーション:ON
N9 TANGON (C, -90)	; 法線方向制御:ON ; オフセット角度: -90° または 270°
N10 G02 X30 Y30 I20 J0 SPN=2 PON	; 円弧補間 ; 軌跡分割 → 2つのリフト
N15 G0 X70 Y10 SPOF	; 位置決め ; ストローク開始 OFF

プログラムコード	コメント
N17 TANGON (C, 90)	; 法線方向制御:ON ; オフセット角度:90°
N20 G03 X35,86 Y24,14 CR=20 SPP=16 SON	; 円弧補間 ; 軌跡分割 → 4つのリフト ; パンチ開始:ON
N25 G0 X74.14 Y35.86 C0 PON	; 位置決め ; C/C1 軸:0° ; ニブリング開始:ON
N27 TANGON (C, 0)	; 法線方向制御:ON ; オフセット角度:0°
N30 G03 X40 Y50 I-14,14 J14,14 SPN=5 SON	; 円弧補間 ; 軌跡分割 → 5つのリフト ; パンチ開始:ON
N35 G0 X30 Y65 C90 SPOF	; 位置決め ; ストローク開始 OFF
N40 G91 X-10 Y-25 C180	; 位置決め、インクリメンタル ; C/C1 軸: +180° ; □ 270° 絶対
N43 TANGOF	; 法線方向制御:OFF
N45 G90 G02 Y60 I0 J10 SPP=2 PON	; 円弧補間 ; 軌跡分割 → 2つのリフト
N50 SPOF	; ストローク開始 OFF
N55 TRAILOF (C1, C)	; 連結モーション:OFF
N60 M2	

11.6 保護領域



- ① 挿入ポイント 0°
- ② 位置決め
- ③ ストローク

## 11.6 保護領域

### クランプ保護領域

「クランプ保護領域」機能は、「保護領域」機能のサブセットとして含まれます。その目的は、単純にクランプと工具が相互にリスクとなるかどうかを監視することです。

#### 注記

クランプ保護違反の場合には、バイパス方式はまったく実行されません。

#### 参照先:

『機能マニュアル、基本機能』、「軸監視と保護領域(A3)」

## 11.7 補足条件

### 「パンチングとニブリング」機能の適用

この機能はオプションです(「パンチングとニブリング機能」)。ライセンス管理を通じてハードウェアに割り当ててください。

## 11.8 例

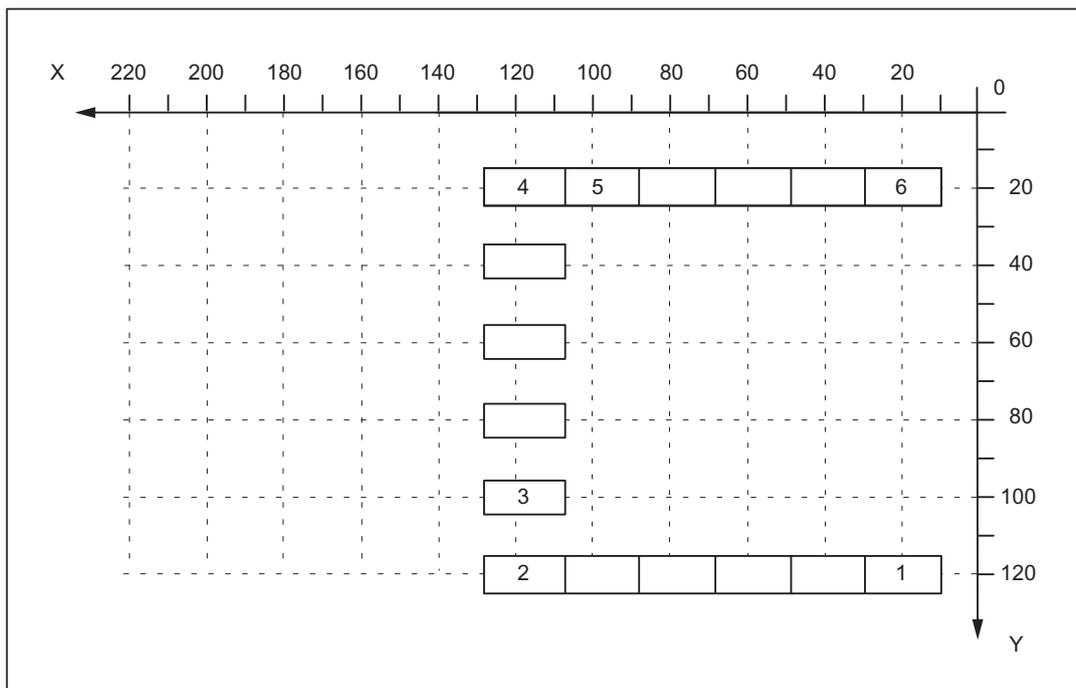
### 11.8.1 ニブリング運転開始の定義例

#### 例 1

ニブリング運転開始の定義例

プログラムコード	コメント
:	
:	
N10 G0 X20 Y120 SPP= 20	; 位置 1 にアプローチ
N20 X120 SON	; ニブリング開始の定義、「1」で1番目のストローク、 ; 「2」で最後のストローク
N30 Y20	; ニブリング開始の定義、「3」で1番目のストローク、 ; 「4」で最後のストローク
N40 X20	; ニブリング開始の定義、「5」で1番目のストローク、 ; 「6」で最後のストローク
N50 SPOF	
N60 M2	

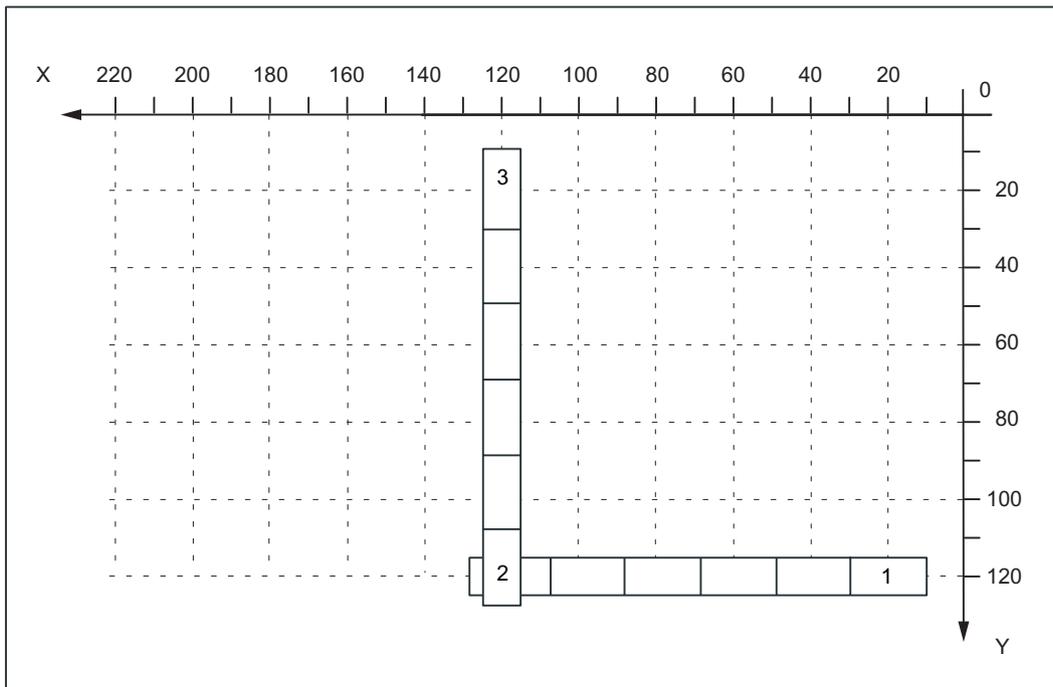
11.8 例



例 2

この例は、「法線方向制御」機能を利用しています。接線軸の名称として、Z が選択されています。

プログラムコード	コメント
:	
:	
N5 TANG (Z, X, Y, 1, 「B」)	; 接線軸を定義
N8 TANGON (Z, 0)	; 法線方向制御を選択
N10 GO X20 Y120	; 位置 1 にアプローチ
N20 X120 SPP=20 SON	; ニブリング開始の定義、 ; 法線方向制御選択、 ; 「1」で 1 番目のストローク、「2」で最後のストローク
N30 SPOF TANGOF	; ニブリングモードを選択解除し、 ; 法線方向制御を選択解除
N38 TANGON (Z, 90)	; 法線方向制御を選択
N40 Y20 SON	; ニブリング開始の定義、 ; 法線方向制御選択、 ; 「2」で 1 番目のストローク、ブロック N20 に対して 90 度回転、 ; 「3」で最後のストローク
N50 SPOF TANGOF	; ニブリングモードを選択解除し、 ; 法線方向制御を選択解除
N60 M2	



ニブリング開始の定義の例 3 と 4

例 3: SPP のプログラミング

プログラムコード	コメント
:	
:	
N5 G0 X10 Y10	; 位置決め
N10 X90 SPP=20 SON	; ニブリング開始の定義、 ; 5 回のパンチングを開始
N20 X10 Y30 SPP=0	; 1 回のパンチングを軌跡の終点で開始
N30 X90 SPP=20	; 4 回のパンチングを 20 mm の間隔で開始
N40 SPOF	
N50 M2	

11.8 例

例 4 SPN のプログラミング

プログラムコード	コメント
:	
:	
N5 G0 X10 Y10	; 位置決め
N10 X90 SPN=4 SON	; ニブリング開始の定義、 ; 5 回のパンチングを開始
N20 X10 Y30 PON	; 1 回のパンチングを軌跡の終点で開始
N30 X90 SPN=4	; 4 回のパンチングを開始
N40 SPOF	
N50 M2	

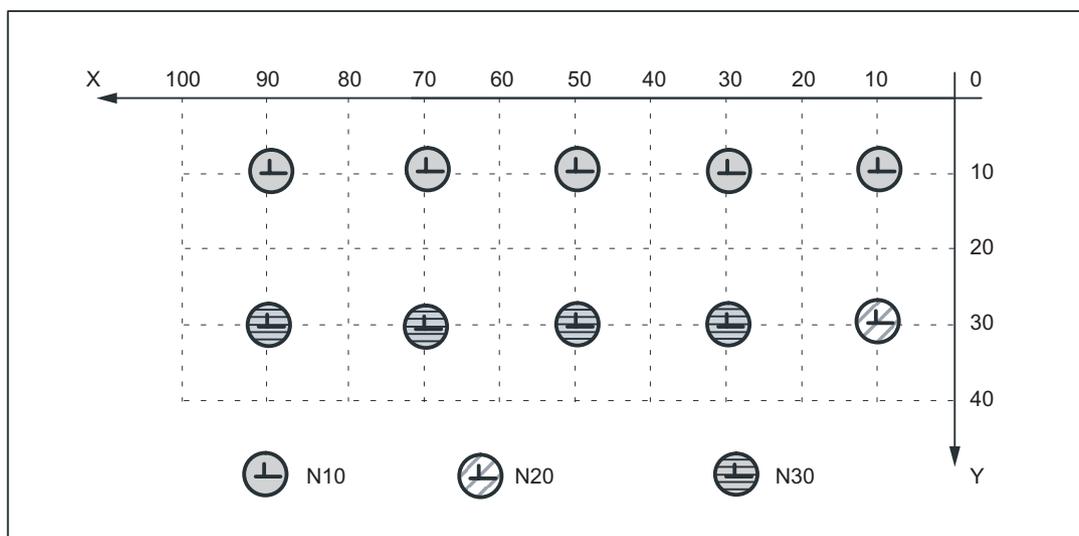


図 11-7 ニブリング開始の定義の例 3 と 4

ニブリング開始の定義無しの例 5 と例 6

例 5 SPP のプログラミング

プログラムコード	コメント
:	
:	
N5 G0 X10 Y30	; 位置決め
N10 X90 SPP=20 PON	; ニブリング開始の定義なし、 ; 4 回のパンチングを開始

プログラムコード	コメント
N15 Y10	; 1 回のパンチングを軌跡の終点で開始
N20 X10 SPP=20	; 4 回のパンチングを 20 mm の間隔で開始
N25 SPOF	
N30 M2	

## 例 6 SPN のプログラミング

プログラムコード	コメント
:	
:	
N5 G0 X10 Y30	; 位置決め
N10 X90 SPN=4 PON	; ニプリング開始の定義なし、 ; 4 回のパンチングを開始
N15 Y10	; 1 回のパンチングを軌跡の終点で開始
N20 X10 SPN=4	; 4 回のパンチングを開始
N25 SPOF	
N30 M2	

11.8 例

例 7 SPP プログラミングの用途例

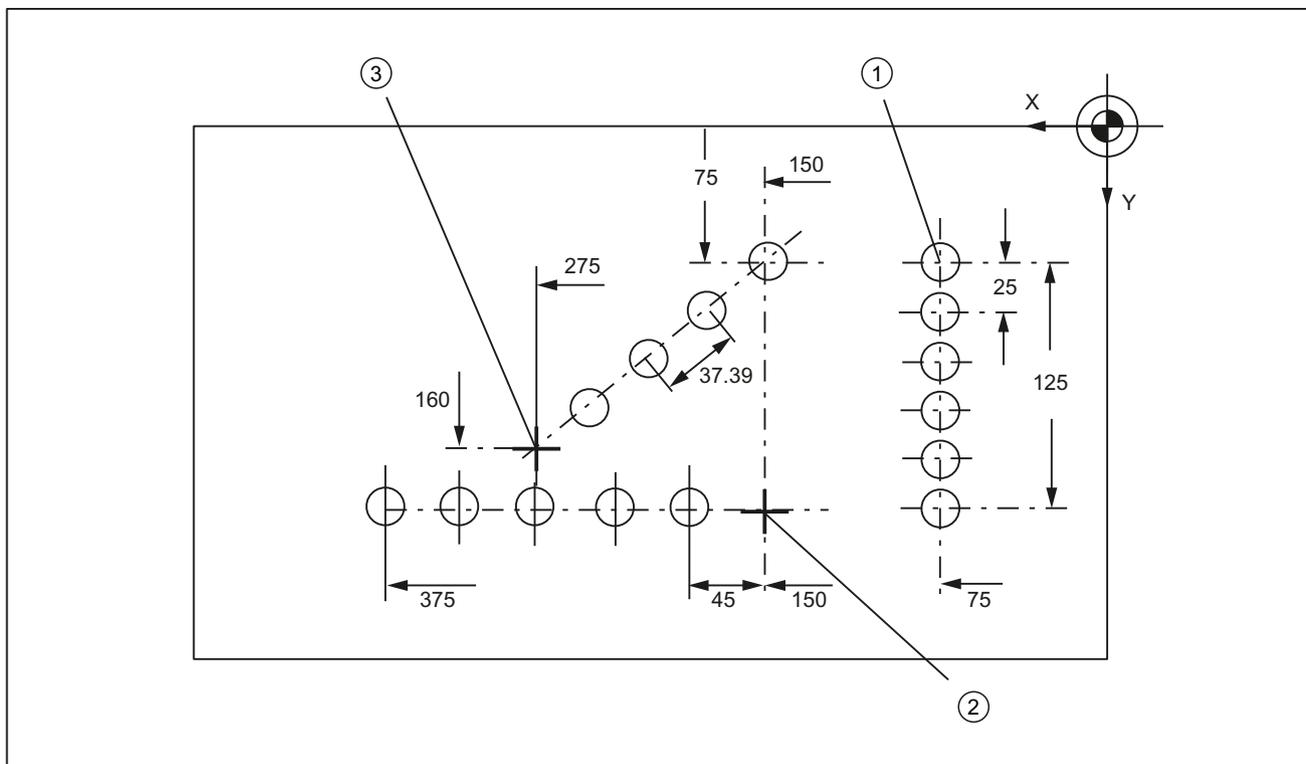


図 11-8 ワーク

プログラムからの抜粋:

プログラムコード	コメント
N100 G90 X75 Y75 F60 PON	; 穴の垂直線の起点(1)への ; 位置決め、単独穴のパンチング
N110 G91 Y125 SPP=25 PON	; 終点座標(インクリメンタル) ; 分割軌跡: 25 mm、パンチングを起動
N120 G90 X150 SPOF	; アブソリュート指令、 ; 穴横線の起点(2)での位置決め
N130 X375 SPP=45 PON	; 終点座標、分割軌跡: 45 mm
N140 X275 Y160 SPOF	; 穴の傾斜線の起点(3)への ; 位置決め
N150 X150 Y75 SPP=40 PON	; 終点座標、プログラム指令された ; 分割軌跡: 40 mm、計算値
	; 分割軌跡: 37.39 mm
N160 G00 Y300 SPOF	; 位置決め

## 11.9 データリスト

### 11.9.1 マシンデータ

#### 11.9.1.1 一般マシンデータ

番号	識別子: \$MN_	説明
11450	SEARCH_RUN_MODE	ブロック検索パラメータの設定内容

#### 11.9.1.2 チャンネル別マシンデータ

番号	識別子: \$MC_	説明
20150	GCODE_RESET_VALUES[n]	G グループのリセット
26000	PUNCHNIB_ASSIGN_FASTIN	ストローク制御による入力バイトのハードウェア割り当て
26002	PUNCHNIB_ASSIGN_FASTOUT	ストローク制御による出力バイトのハードウェア割り当て
26004	NIBBLE_PUNCH_OUTMASK[n]	高速出力ビットのマスク
26006	NIBBLE_PUNCH_INMASK[n]	高速入力ビットのマスク
26008	NIBBLE_PUNCH_CODE[n]	M 機能の決定
26010	PUNCHNIB_AXIS_MASK	パンチング軸およびニブリング軸の定義
26012	PUNCHNIB_ACTIVATION	パンチング機能およびニブリング機能の適用
26014	PUNCH_PATH_SPLITTING	自動軌跡分割の適用
26016	PUNCH_PARTITION_TYPE	動作中の自動軌跡分割による単独軸の動作
26018	NIBBLE_PRE_START_TIME	自動的に有効になる事前開始のタイミング
26020	NIBBLE_SIGNAL_CHECK	入力信号の監視

11.9 データリスト

11.9.2 セッティングデータ

11.9.2.1 チャンネル別セッティングデータ

番号	識別子: \$SC_	説明
42400	PUNCH_DWELL_TIME	ドウェル時間
42402	NIBPUNCH_PRE_START_TIME	事前開始時間
42404	MINTIME_BETWEEN_STROKES	連続する 2 ストローク間の最短時間間隔

11.9.3 信号

11.9.3.1 チャンネルへの信号

信号名称	SINUMERIK 840D sl	SINUMERIK 828D
ストローク無効	DB21、... .DBX3.0	-
手動ストローク開始	DB21、... .DBX3.1	-
ストローク抑制	DB21、... .DBX3.2	-
ストローク動作無効	DB21、... .DBX3.3	-
ストローク遅延	DB21、... .DBX3.4	-
手動ストローク開始	DB21、... .DBX3.5	-

11.9.3.2 チャンネルからの信号

信号名称	SINUMERIK 840D sl	SINUMERIK 828D
ストローク開始が有効	DB21、... .DBX38.0	-
手動ストローク開始の確認応答	DB21、... .DBX38.1	-

## 11.9.4 言語命令

G グループ	言語命令;ゲン ゴメイレイ	意味	
35	SPOF	ストローク/パンチング OFF	パンチングおよびニブリング OFF
35	SON	ストローク ON	ニブリング ON
35	SONS	ストローク ON	ニブリング ON (位置コントローラ)
35	PON	パンチング ON	パンチング ON
35	PONS	パンチング ON	パンチング ON (位置コントローラ)
36	PDELAYON	遅延のあるパンチング ON	遅延のあるパンチング ON
36	PDELAYOF	遅延のあるパンチング OFF	遅延のあるパンチング OFF
軌跡分割;キセキブンカツ			
	SPP		ストローク毎の軌跡、モーダル動作
	SPN		ブロック毎のストローク数、ノンモーダル動作

## 11.9 データリスト

## P2 :位置決め軸

### 12.1 製品の概要

#### 補助動作軸

ワーク加工用の軸の他に、最新の工作機械には、次のような補助動作軸も搭載できます。

- 工具マガジン用の軸
- 工具タレット用の軸
- ワーク運搬用の軸
- パレット運搬用の軸
- ロータ用の軸(複数軸も)
- 工具交換装置用の軸
- スリーブセット/端面支持用の軸

ワーク加工用の軸は、軌跡軸と呼ばれます。チャンネル内では、これらの軸が補間器で指令されて、同時に開始し、加速し、終点に達して、停止します。

補助動作軸は、個別の軸別送り速度で、軌跡軸とは無関係に移動します。以前は、これらの軸の多くが油圧で移動されて、パートプログラムの補助機能で開始されていました。

NC で実行される閉ループ軸制御によって、パートプログラム中の名称と画面に表示される実位置で軸を特定できます。

---

### 注記

#### 「位置決め軸/補助主軸」オプション

補助動作軸は補間(「全値」)NC 軸であってはけません。補助動作は付加軸を使用して実行することもできます。付加軸を使用するには、「位置決め軸/補助主軸」オプションを選択してください。

#### 機能上の制限

オプションの位置決め軸/補助主軸にはわずかな機能しかありません。以下の機能は使用できません。

- 軌跡軸、ジオメトリ軸、または特殊軌跡軸としての軸の使用
- ジオメトリ軸グループ(GEOAX)への軸の統合
- 精密なねじ切りとタッピング

#### セットアップ

標準では、軸は補間軸として定義されています。

#### MD30460 \$MA\_BASE\_FUNCTION\_MASK bit 8 = 0

軸を、機能が制限された位置決め軸/補助主軸として動作させるには、ビット 8 の値を「1」に設定してください。

#### MD30460 \$MA\_BASE\_FUNCTION\_MASK bit 8 = 1

---

## 機能

「位置決め軸」機能により、コントロールシステムへの補助動作軸の組み込みが容易になります。

- プログラミング中:  
軸は、貴重な加工時間を犠牲にすることなく、同じパートプログラムでワーク加工用の軸と一緒にプログラム指令されます。  
特別な(POS、POSA)移動命令があります。
- プログラムのテスト中/起動中:  
プログラムのテストと起動は、すべての軸で同時におこなわれます。
- 運転中:  
加工処理の運転と監視は、すべての軸で同時に開始します。

- PLC 設定中/セットアップ中:  
PLC でも外部コンピュータ(PC)でも、加工用の軸と補助動作軸の間の同期制御は考慮されません。
- システム設定中:  
第2チャンネルは必要ありません。

## 移動と補間

各チャンネルには、次の補間機能を備えた1個の軌跡補間器と1個以上の軸補間器があります。

- 軌跡補間器の場合:  
直線補間(G1)、円弧補間(G2 / G3)、スプライン補間、など。
- 軸補間器の場合:  
位置決め軸がプログラム指令されている場合は、軸補間器はコントローラで(直線補間 G1 で)起動します。
- 動作終了条件:  
位置決め軸がプログラム指令終了位置に到達したら、動作終了条件 FINEA、COARSA、または IPOENDA が満たされます。
- 早送り移動動作の軌跡軸:  
軌跡軸は、直線補間または非直線補間モードの早送り移動動作(G0)で移動できます。  
参照先:  
機能マニュアル 基本機能; 章「B1:連続軌跡モード、イグザクトストップ、先読み」
- 独立の単独軸運転:  
単独 PLC 軸、静的なシンクロナイズドアクションによって開始されるコマンド軸、または非同期揺動軸は、NC とは無関係に補間可能です。  
したがって、実行プログラムで補間される軸/主軸は、NC プログラムと無関係に応答します。プログラムの実行で出力されるチャンネル応答は分離されて、特定の軸/主軸の制御が PLC に伝送されます。
- PLC による制御:  
すべてのチャンネル別信号は、通常は軌跡軸と位置決め軸について同じように機能します。  
位置決め軸は、追加の軸別信号によって制御できます。  
PLC 軸は PLC で、基本プログラムの特殊機能ブロックによって移動されて、その移動は、その他のすべての軸と非同期になることがあります。移動動作は、軌跡とシンクロナイズドアクションは別におこなわれます。

## 12.2 自チャネル、位置決め軸、または同時位置決め軸

軸が、工作機械の補助動作用に設けられると、必要な機能は、軸が次のどれに該当するかで決定されます。

- 内部のパートプログラムでプログラム指令される(「自チャネル (ページ 806)」の章を参照してください)。
- 加工運転として同一パートプログラムでプログラム指令される(「位置決め軸 (ページ 807)」の章を参照してください)。
- 加工中に PLC で独自に開始される (「同時位置決め軸 (ページ 810)」の章を参照してください)。

### 12.2.1 自チャネル

チャネルは独立の NC を表しており、パートプログラムの助けによって、他のチャネルとは別個に軸、主軸、およびマシン機能の動作の制御に使用できます。

#### チャネル間の独立性

チャネル間の独立性は、次の条件によって保証されます。

- チャネル毎の有効なパートプログラム
- 次のチャネル別インタフェース信号
  - DB21、... DBX7.1 (NC スタート)
  - DB21、... DBX7.3 (NC ストップ)
  - DB21、... DBX7.7 (リセット)
- チャネルごとに 1 つの送り速度オーバーライド
- チャネル毎に 1 つの早送りオーバーライド
- チャネル別評価およびアラーム表示

- 次のチャンネル別表示
  - 実軸位置
  - 有効な G 命令
  - 動作中の補助機能
  - 現在のプログラムブロック
- チャンネル別テストおよび次のチャンネル別プログラム制御:
  - (Single BLock)
  - ドライラン(DRY RUN)
  - ブロック検索
  - プログラムテスト

## 参照先

チャンネルの機能の詳細については、次の資料を参照してください:  
機能マニュアル、基本機能; BAG、チャンネル、プログラム運転、リセット応答 (K1)

### 12.2.2 位置決め軸

位置決め軸は軌跡軸、つまり、ワークの加工を実行する軸と一緒にプログラム指令されます。

位置決め軸と軌跡軸の命令はいずれも、同じ NC ブロックに含めることができます。これらの軸は同じ NC ブロックにプログラム指令されますが、軌跡軸と位置決め軸は一緒には補間されず、それぞれの終点に同時に達するわけでもありません(直接の時間的な相関関係はありません。「移動動作および補間機能 (ページ 811)」の章も参照してください)。

### 位置決め軸のタイプとブロック切り替え

ブロック切り替えタイミングは、プログラム指令された位置決め軸のタイプによって決まります(「ブロック切り替え (ページ 825)」の章も参照してください)。

タイプ	説明
1	すべての軌跡軸と位置決め軸が、プログラム指令されたそれぞれの終点に達すると、ブロック切り替えが発生します。
2	すべての軌跡軸が、プログラム指令されたそれぞれの終点に達すると、ブロック切り替えが発生します。 位置決め軸タイプ 2 の場合は、複数のブロック境界にわたってプログラム指令された、その終点にアプローチできます。
3	軌跡補間のための移動終了とブロック切り替えの条件が満たされている場合は、単独軸補間の減速カーブ内のブロック切り替えを設定できます。

### 移動の同期制御

位置決め軸では、同じ加工プログラムから移動を有効にすることができます。また、この移動を、WAITP 命令(タイプ 2)を使用して、ブロック境界(タイプ 1)または指定された点で同期させることができます。

### 減速カーブでブロック切り替えのための動作終了条件

単独軸補間では、減速カーブでブロック切り替えのために別の動作終了条件を設定することもできます。

### G0 の位置決め軸としての移動軌跡軸

各軌跡軸は、位置決め軸として早送り移動動作で移動動作できます(G0)。このため、すべての軸は独立してその終点に移動します。

この方法で、順番にプログラム指令された 2 つの X 軸と Z 軸は、G0 と組み合わせて位置決め軸として扱われます。Z 軸へのブロック切り替えは、減速時間設定(100-0%)に応じて、X 軸で開始できます。Z 軸は、X 軸が移動している間に、移動を開始します。両方の軸は互いに独立して、それぞれの終点にアプローチします。

### 軸タイプ

直線軸と回転軸も、位置決め軸にすることができます。

位置決め軸は、割り出し軸として設定することもできます。

### 位置決め軸と軌跡軸の独立性

位置決め軸と軌跡軸の相互の独立性は、次の方法で保証されています。

- 補間を共有しない
- 各位置決め軸には専用の軸補間器がある
- 各位置決め軸専用の送り速度オーバライド
- 指令可能な専用の送り速度
- 専用の「軸別残移動距離削除」インタフェース信号

### 依存関係

位置決め軸には、次のような依存関係があります。

- 共有パートプログラム
- パートプログラムのブロック境界のみで軸の位置決めを開始
- 早送り移動動作 G0 では、軌跡軸が 2 種類のモードのいずれかで位置決め軸として移動します
- 早送りオーバライドなし
- 次のインタフェース信号はチャネル全体で機能します。したがって、位置決め軸にも機能します。
  - DB21、... DBX7.1 (NC スタート)
  - DB21、... DBX7.3 (NC ストップ)
  - DB21、... DBX7.7 (リセット)
  - DB21、... DBX6.1 (読み込み禁止)
- プログラムとチャネルに固有のアラームも位置決め軸を無効にします。
- プログラム制御(ドライラン送り、プログラムテスト、DRF など)も、位置決め軸に機能します。
- ブロック検索とシングルブロックも位置決め軸で機能します。
- 検索実行で処理された動作終了条件がプログラム指令されている最後のブロックは、すべての軸を設定するためのコンテナの役割を果たします。
- G 命令(G0、G1、G2、...)のグループ 1 (モーダル動作命令)は、位置決め軸には適用されません。

**参照先:**

『プログラミングマニュアル、基本編』

**用途**

下記は、位置決め軸の一般的な用途です。

- 単独軸ローダ
- 補間をおこなわない複数軸ローダ(PTP → ポイントツーポイント移動)
- ワークの送りと運搬

次のような、その他の用途も可能です。

- G0 では、ワーク搬出とワーク運搬は互いに無関係に、その終点に移動できます。
- 複数の加工処理が順番におこなわれる機械では、単独軸補間の減速カーブのブロック切り替えによって、個々の加工処理が大幅に減少します。

---

**注記**

位置決め軸は、複数軸間の補間が必要な複数軸ローダ(軌跡補間器)には対応していません。

---

**12.2.3 同時位置決め軸**

同時位置決め軸は、次の機能を持つ位置決め軸です。

- PLC からの起動は、ブロック境界でおこなう必要はありません、いつでもどの運転モードでも(パートプログラムがすでにチャネルで処理中のときでも)実行できます。
- 電源投入直後にパートプログラムで同時位置決め軸を移動するには、プログラム命令 WAITP が必要です。
- PLC で定義された位置に同時位置決め軸が達していなくても、パートプログラムは制約されずに動作し続けます。
- マシンデータ MD30552 \$MA\_AUTO\_GET\_TYPE の設定に応じて、自動の軸入れ替えが可能です。
- プログラミング命令では、次のようになります。
  - GET(<axis>) または WAITP(<axis>) は再度チャネル軸の同時位置決め軸になります。
  - 「RELEASE (axis)」 または WAITP(<axis>) は、PLC 制御では同時軸になるチャネル軸です。

## PLC からの起動

SINUMERIK 840D sl の場合、同時位置決め軸は、PLC から FC 18 によって有効になります。

- 送り速度  
送り速度=0 の場合は、送り速度が次のマシンデータで決定されます。  
MD32060 \$MA\_POS\_AX\_VELO (位置決め軸速度の初期設定)
- アブソリュート指令(G90)、インクリメンタル指令(G91)  
回転軸のために最短軌跡に沿ったアブソリュート指令(<回転軸名称>=DC (<value>))

次の機能が定義されます。

- 直線補間(G1)
- mm/min 単位または度/min 単位の送り速度 (G94)
- イグザクトストップ(G9)
- 現在選択されている設定可能ゼロオフセットが有効

## 用途

同時位置決め軸の一般的な用途は次のとおりです。

- 加工中に手動でロードとアンロードをおこなう工具マガジン
- 加工中に工具準備をおこなう工具マガジン

## 12.3 移動動作および補間機能

### 12.3.1 軌跡補間器および軸補間器

#### 軌跡補間

すべてのチャンネルには、直線補間(G1)、円弧補間(G2/G3)、スプライン補間など、広範囲の補間モードのための軌跡補間器があります。

#### 軸補間器

各チャンネルには、軌跡補間器の他に軸補間器もあります。その最大数は、既存の最大チャンネル軸数に対応しています。

### 12.3 移動動作および補間機能

位置決め軸がプログラム指令されている場合は、軸補間器は制御装置で直線補間 G1 で起動します。この軸補間器は、位置決め軸のプログラム指令終了位置に達するまで、軌跡補間器とは無関係に動作します。

軌跡補間器と軸補間器の間および各軸補間器間のいずれにも時間的な相関関係はありません。

軌跡コントロールモード(G64)は、位置決め軸には使用できません。

位置決め軸がプログラム指令終了位置に到達したら、動作終了条件 FINEA、COARSA、または IPOENDA が満たされます。

#### 12.3.2 独立の単独軸運転

##### 機能

単独 PLC 軸、静的なシンクロナイズドアクションによって開始されるコマンド軸、または非同期揺動軸は、NC とは無関係に補間可能です。実行プログラムで補間される軸/主軸は、次の操作については、NC プログラムとは無関係に応答します。

- NC ストップ
- アラームの取扱い
- プログラムの制御
- プログラムの終了
- RESET

##### 境界条件

現在、NC プログラムに従って動作中の軸/主軸は、PLC では制御されません。

コマンド軸動作は、PLC 制御の軸/主軸に対して、ノンモーダルまたはモーダルのシンクロナイズドアクションによっては開始できません。アラーム 20143 が通知されます。

##### PLC への軸制御の伝送

###### シーケンスの説明

1. PLC → NC:軸  
DB31、... DBX28.7 = 1 (PLC 制御軸)の制御要求
2. NC:軸がメイン軸または中立軸のいずれであるかをチェックします。

3. NC:PLC から付加軸を制御可能どうかをチェックします。
4. NC が伝送を確認します。
  - DB31、... DBX63.1 = 1(PLC が軸を制御します)
  - システム変数\$AA\_SINGLAX\_STAT = 1

結果:PLC が軸/主軸を制御します。

#### 選択肢

初期状態:軸は PLC で制御されます。チャンネル停止の結果、チャンネルは「中断」状態です。

- 軸状態「無効」 ⇒
  - 停止状態がキャンセルされます。
  - 軸が起動した場合、これは直接、軸移動につながります。
- 軸状態「有効」 ⇒
  - 停止状態はキャンセルされません。
  - 使用例 2「軸の停止」に従って軸状態を生成し、
  - 使用例 3「軸移動の再開」に従って、軸移動を再開します。
- リセットがチャンネルでおこなわれます ⇒
 

この処理は、PLC による制御の受け入れとは非同期です。上記の 2 つの切り替えをおこなわなかった場合は、軸は、チャンネルに割り当てられ、リセットされます。

#### 境界条件

NC プログラムによって移動する軸/主軸は、PLC に伝送できません。静的シンクロナイズドアクションによって、または揺動軸、中立軸、同時位置決め軸、またはコマンド軸として移動する軸/主軸は、伝送できます。

### PLC による軸制御の解除

#### シーケンスの説明:

1. PLC → NC:PLC が、軸制御を NC に返します。  
DB31、... DBX28.7 = 0 (PLC 制御軸)
2. NC:軸のアラームが発生しているかどうかをチェックします。
3. NC:まだ完了していない移動が有効であるかどうかをチェックします。これが有効である場合、使用例 2「軸/主軸の停止」に従った軸停止により移動が停止します。
4. NC:使用例 4「軸/主軸のリセット」に対応して軸のリセットを実行します。
5. NC が受け入れを確認します。  
DB31、... DBX63.0 = 0 (リセット実行)  
DB31、... DBX63.1 = 0(PLC が軸を制御します)  
DB31、... DBX63.2 = 0 (軸停止有効)  
システム変数\$AA\_SINGLAX\_STAT = 0

### 12.3 移動動作および補間機能

結果:これで、NC が軸/主軸の制御を獲得します。

#### 選択肢

次の場合、NC は伝送を確認しますが、内部的に軸/主軸の「停止」チャンネル状態を設定します。

- チャンネルが「中断」状態である
- 停止アラームがこのチャンネルに発生する
- 停止アラームがこのモードグループに発生する

#### 境界条件

軸/主軸は、PLC 制御で動作中の必要があります。

NC は、軸のアラームが発生中でない場合にかぎり、軸/主軸の受け入れを確認します。

### 使用例に基づくシーケンスの説明

#### 必要条件

軸/主軸が PLC で制御されている

#### 関連する NC/PLC インタフェース信号

PLC により制御される軸/主軸のいずれかが、NC プログラムから独立した次の NC/PLC インタフェース信号によって影響を受ける場合があります。

- DB21、... DBX6.2 (残移動距離削除)
- DB31、... DBX28.1 (リセット)
- DB31、... DBX28.2 (継続)
- DB31、... DBX28.6 (減速カーブに沿って停止)

自律単独運転時の NC/PLC インタフェースでの NC と PLC 間の信号フローの場合(「PLC による制御 (ページ 834)」の章を参照してください)。

### 使用例 1:軸/主軸のキャンセル

軸/主軸機能をキャンセルした場合の動作は「残移動距離削除」と同じです。

DB21、... DBX6.2 = 1 (残移動距離削除)

## 使用例 2:軸/主軸の停止

メインランから制御されている軸/主軸の次の移動動作が停止します。

- PLC 軸
- 非同期の揺動軸
- 静的シンクロナイズドアクションによるコマンド軸
- 重畳動作\$AA\_OFF、DRF ハンドル移動、オンライン工具オフセット、および外部ゼロオフセット。

軸/主軸のスレーブ軸の移動は停止されません。

シーケンスの説明:

- PLC → NC:軸/主軸の停止要求  
DB31、... DBX28.6 = 1(減速カーブに沿った停止)
- NC:減速カーブに沿って軸を減速します。
- NC が実行を確認します。
  - DB31、... DBX60.6 = 1 (汎用イグザクトストップ)
  - DB31、... DBX60.7 = 1 (精密イグザクトストップ)
  - DB31、... DBX63.2 = 0 (軸停止有効)
  - DB31、... DBX64.6 / 7 = 0 (移動指令マイナス / プラス)
  - システム変数\$AA\_SINGLAX\_STAT == 3 で軸状態が中断されます。

結果:軸/主軸が停止されます。

---

### 注記

#### スレーブ軸の移動

スレーブ軸の移動は、マスタ軸が停止している場合にのみ抑制できます。

#### 後退移動

「停止延長と退避」機能によって起動された退避移動は停止できません。

#### 参照先

『機能マニュアル、応用機能』; 「停止延長と退避(R3)」

---

## 使用例 3:軸/主軸の継続

使用例 2 「軸の停止」の後に中断された移動動作が継続します。

## シーケンスの説明:

- PLC → NC:軸の継続  
DB31、... DBX28.2 = 1 (継続)
- NC:軸/主軸の軸アラームに解除条件「CANCELCLEAR」または「NCSTARTCLEAR」が発生しているかどうかをチェックします。これが発生している場合は、解除されます。
- NC:軸移動を再開できるかどうかをチェックします。再開できる場合、軸/主軸は「有効」状態に遷移します。
- NC が実行を確認します。
  - DB31、... DBX60.6 = 0 (汎用イグザクトストップ)
  - DB31、... DBX60.7 = 0 (精密イグザクトストップ)
  - DB31、... DBX63.2 = 0 (軸停止有効)
  - DB31、... DBX64.6 / 7 = 1 (移動指令マイナス/プラス)
  - システム変数\$AA\_SINGLAX\_STAT == 4 で軸状態が有効になります。

結果:軸/主軸の移動動作は継続します。

## 境界条件

次の場合、継続要求は無視されます。

- 軸/主軸が PLC により制御されていない。
- 軸/主軸が停止状態になっていない。
- 軸/主軸のアラームが未処置である。

## 使用例 4:軸/主軸のリセット(リセット)

## シーケンスの説明:

- PLC → NC:この軸/主軸に対するリセット要求  
DB31、... DBX28.1 = 1(リセット)
- NC:軸/主軸の「停止」状態への遷移。
- NC:PLC に対して、停止されたシーケンスを中断して PLC に中断を通知します。基本的に、「残移動距離削除」と同じです。
- NC:軸/主軸の内部ステータスがリセットされます。

- NC:リセット時に有効になる軸別マシンデータが有効になります。

---

#### 注記

チャンネルリセットと同時に、PLC から制御される軸に対して有効な軸別マシンデータはありません。

---

- NC が実行を確認します。
  - DB31、... DBX63.0 = 1 (リセット実行)
  - DB31、... DBX63.2 = 0 (軸停止有効)
  - システム変数\$AA\_SINGLAX\_STAT = 1
- NC がこの処理を終了します。

### 12.3.3 NC 制御 ESR による独立の単独軸機能

#### NC 制御による停止延長

NC 制御による停止延長と後退機能は、単独軸にも使用可能で、次の軸マシンデータで設定できます。

ESR 単独軸の遅延時間の場合

MD37510 \$MA\_AX\_ESR\_DELAY\_TIME1

単独軸の補間減速の ESR 時間の場合

MD37511 \$MA\_AX\_ESR\_DELAY\_TIME2

ただし、これらの軸マシンデータの値は、軸/主軸が単独軸の場合にのみ有効です。

NC 制御停止延長と退避は、軸のトリガ\$AA\_ESR\_TRIGGER[軸]で有効になります。これは、\$AC\_ESR\_TRIGGER と同様に機能し、単独軸のみ適用されます。

#### 参照先:

『機能マニュアル、応用機能』; 「連結軸と ESR(M3)」

## NC 制御による退避

単独軸の退避には、値が POLFA(軸、タイプ、値)によってプログラム指令されており、かつ次の条件を満たしてください。

- 起動時に軸は単独軸である
- \$AA\_ESR\_ENABLE[軸]=1
- POLFA(軸、タイプ、値)でタイプ=1 のみ、または POLFA(軸、値、軸、タイプ、軸タイプ)でタイプ=2 のみである

## 注記

## 単独軸の NC 制御停止延長の場合:

起動時に軸が単独軸である場合にのみ有効です。それ例外の場合は、起動が無視され、この軸の軸停止は実行されません。

## 単独軸の NC 制御退避の場合:

チャンネル別 NC 制御の退避機能は、単独軸には無効です。トリガ \$SAC\_ESR\_TRIGGER が実行時に、すべての単独軸は、チャンネル別の退避が無視されます。

これは、次のような退避パラメータが、すべて設定されているときも適用されます。

MD37500 \$MA\_ESR\_REACTION

軸の \$AA\_ESR\_ENABLE、など。

## 例

単独軸の停止延長の場合:

MD37500 \$MA\_ESR\_REACTION[AX1]=22

MD37510 \$MA\_AX\_ESR\_DELAY\_TIME1[AX1]=0.3

MD37511 \$MA\_AX\_ESR\_DELAY\_TIME2[AX1]=0.06

...

\$AA\_ESR\_ENABLE[AX1] = 1

\$AA\_ESR\_TRIGGER[AX1]=1 ; 軸はここで停止処理を開始します

単独軸の退避の場合:

MD37500 \$MA\_ESR\_REACTION[AX1]=21

...

\$AA\_ESR\_ENABLE[AX1] = 1

POLFA(AX1, 1, 20.0); AX1 が軸の退避位置 20.0 ; (アブソリュート値)に割り当てられます

\$AA\_ESR\_TRIGGER[AX1]=1 ; AX1 はここで退避を開始します

POLFA(軸、タイプ): プログラミングの省略は可能です

POLFA(軸、0/1/2) ; 高速の解除/適用

 <b>警告</b>
<p><b>事前処理の制限はありません</b></p> <p>記号の省略を使用し、タイプのみを変更した場合は、使用した退避位置または退避軌跡の値は有効値であることを確認してください。</p> <p>記号の省略は、例外的な場合のみ使用してください。</p> <p>特に次の操作の後には、例外的な場合に該当します。</p> <p>電源投入。この場合は、退避軌跡または退避位置をリセットしてください。</p> <p>POLFA(軸、1、\$AA_POLFA[軸]) ; 先読み停止処理をおこないません。</p> <p>POLFA(軸、1); 先読み停止処理をおこないません。</p>

## 12.4 位置決め軸のダイナミック応答

### 速度

位置決め軸は、その軸にプログラム指令された、軸別送り速度で移動します。「移動動作および補間機能 (ページ 811)」で説明されているように、この送り速度は軌跡軸による影響を受けません。

送り速度は、軸別速度として、mm/min、inch/min、または° /min の単位でプログラム指令されます。

軸別送り速度は常に、軸名称によって、固定で位置決め軸に割り当てられます。

位置決め軸に対して送り速度がプログラム指令されていない場合は、コントロールシステムが自動的に、次の軸別マシンデータに設定された送り速度を適用します。

MD32060 \$MA\_POS\_AX\_VELO (位置決め軸速度の初期設定)

プログラム指令された軸別送り速度は、プログラム終了まで有効です。

## 送り速度オーバーライド

軌跡軸と位置決め軸には、個別の送り速度オーバーライドがあります。各位置決め軸は、それ自体の軸別送り速度オーバーライドの作用を受けます。

## 早送りオーバーライド

早送りオーバーライドは、軌跡軸にのみ適用されます。位置決め軸では、早送り補間はおこなわれません(軸の直線補間 G01 のみ)。したがって、早送りオーバーライドは適用されません。

## 毎回転送り速度

ジョグモードでは、軸/主軸の動作は、SD41100 JOG\_REV\_IS\_ACTIVE (ジョグ有効時の毎回転送り速度)の設定内容によって決まります。

- このセッティングデータが有効な場合は、軸/主軸が常に、メイン主軸に応じて、毎回転送り速度 MD32050 \$MA\_JOG\_REV\_VELO (ジョグモードの回転送り速度)または MD32040 \$MA\_JOG\_REV\_VELO\_RAPID (早送りオーバーライドによるジョグモードの毎回転送り速度)で移動します。
- セッティングデータが有効でない場合、軸/主軸の動作は SD43300 \$SA\_ASSIG\_FEED\_PER\_REV\_SOURCE (位置決め軸/主軸の毎回転送り速度)によって決まります。
- セッティングデータが有効でない場合、座標回転フレームが有効なジオメトリ軸の動作は、チャンネル別セッティングデータ SD42600 \$SC\_JOG\_FEED\_PER\_REV\_SOURCE (ジョグ運転モードの、座標回転フレームが有効なジオメトリ軸の毎回転送り速度によって決まります)。

## 最大軸加速度

位置決め軸の移動では、設定された位置決め軸のダイナミック応答モードに応じて、次の2つの最大値のいずれかが有効になります。

- MD32300 \$MA\_MAX\_AX\_ACCEL [0] (DYNORM ダイナミック応答モード DYNORM での軌跡移動に対する最大軸加速度)
- MD32300 \$MA\_MAX\_AX\_ACCEL [1] (DYNPOS ダイナミック応答モード DYNORM での軌跡移動に対する最大軸加速度)

位置決め軸ダイナミック応答モードは、NC 別マシンデータで設定されます。

MD18960 \$MN\_POS\_DYN\_MODE = <モード>

<モード>	意味
0	有効な最大軸加速度:MD32300 \$MA_MAX_AX_ACCEL[ 0 ]
1	有効な最大軸加速度:MD32300 \$MA_MAX_AX_ACCEL[ 1 ]

### 最大軸加々速度

加々速度制限が有効な位置決め軸の移動時に、次に示すマシンデータのいずれかの値が最大軸加々速度として有効になります。

- MD32430 \$MA\_JOG\_AND\_POS\_MAX\_JERK (位置決め軸の移動の最大軸加々速度)
- MD32431 \$MA\_MAX\_AX\_JERK [0] (ダイナミック応答モード DYNORM の軌跡移動の最大軸加々速度)
- MD32431 \$MA\_MAX\_AX\_JERK [1] (ダイナミック応答モード DYNPOS の軌跡移動の最大軸加々速度)

使用されるマシンデータは、設定された位置決め軸のダイナミック応答モードによって決定されます。

MD18960 \$MN\_POS\_DYN\_MODE = <モード>

<モード>	意味
	位置決め軸の移動の最大軸加々速度として、以下が有効になります。
0	MD32430 \$MA_JOG_AND_POS_MAX_JERK G75 が有効の場合(固定点アプローチ):MD32431 \$MA_MAX_AX_JERK[0]
1	MD32431 \$MA_MAX_AX_JERK[1]

## 12.5 プログラミング

### 12.5.1 概要

---

#### 注記

位置決め軸のプログラミングについては、

プログラミングマニュアル 基本編の「送り速度制御」と「主軸動作」の章を参照してください。

---

#### 注記

1つのブロックでプログラムできる位置決め軸の最大軸数は、使用可能なチャンネル軸の最大軸数に制限されます。

---

### 定義

位置決め軸は次のパラメータを使用して定義します。

- 軸タイプ: 位置決め軸タイプ 1、タイプ 2、タイプ 3
- 終点座標(アブソリュート指令またはインクリメンタル指令)
- [mm/min]単位の直線軸の送り速度。回転軸の場合は[度/min]単位。

例: 位置決め軸タイプ 1

プログラムコード	コメント
POS[Q1]=200 FA[Q1]=1000	; 軸 Q1 の送り速度は 1000mm/min、位置は 200。

例: 位置決め軸タイプ 2

プログラムコード	コメント
POSA[Q2]=300 FA[Q2]=1500	; 軸 Q2 の送り速度は 1,500mm/min、位置は 300。

---

#### 注記

パートプログラム内では、軸を軌跡軸または位置決め軸にすることができます。しかし移動ブロック内では、固有の軸タイプをそれぞれの軸に割り当ててください。

---

## シンクロナイズドアクションのプログラミング

軸は、パートプログラムと完全に非同期にシンクロナイズドアクションから位置決めできます。

例:

プログラムコード	コメント
ID=1 WHENEVER \$R==1 DO POS[Q4]=10 FA[Q3]=990	; 軸の送り速度をモーダルで指定。

**参照先:**

『機能マニュアル、シンクロナイズドアクション』

## ブロック切り替え

ブロック切り替えは、位置決め軸タイプ 1 とタイプ 2 で下記のように選択できます。

FINEA=<軸名称> or  
FINEA [<軸名称>]

COARSEA=<軸名称> or  
COARSEA [<軸名称>]

IPOENDA=<軸名称> または  
IPOENDA [<軸名称>]

タイプ 3 の位置決め軸では、単独補間の減速カーブ内のブロック切り替えを下記のように設定できます。

IPOBRKA=<軸名称> または  
IPOBRKA (<軸名称> [, <タイミング\*>])

\* ブロック切り替え点のタイミング、減速カーブを%として表します

## アブソリュート指令/インクレメンタル指令

終点座標のプログラミングは、アブソリュート指令(G90)またはインクレメンタル指令(G91)でおこないます。

例	意味
	終点座標のプログラミング
G90 POS [Q1]=200	アブソリュート指令
G91 POS [Q1]=AC (200)	アブソリュート指令
G91 POS [Q1]=200	インクレメンタル指令
G90 POS [Q1]=IC (200)	インクレメンタル指令

## タイプ 2 位置決め軸の再プログラム

タイプ 2 位置決め軸(ブロック境界を超えた移動)では、位置決め軸が終了位置に到達したかどうかをパートプログラムで検出できる必要があります。検出できる場合にかぎり、この位置決め軸を再プログラムできます。(検出できない場合はアラームが発行されます)。

POSA をプログラムし、IPOBRKA (減速カーブ内のブロック切り替え)を指定した POSA を再プログラムした場合、アラームは発行されません。詳細については、「設定可能なブロック切り替えタイミング」の章の NC 命令 IPOBKA を参照してください。

## 協調(WAITP)

協調命令 WAITP では、前の NC ブロックにおいて POSA でプログラムした軸が終了位置に到達するまでプログラムが待機する位置を NC プログラムで指定できます。

WAITP は内部ブロックに存在します。

プログラムが待機する軸に対して参照を指定してください。

例:

プログラムコード	コメント
N10 G01 G90 X200 F1000 POSA[Q1]=200 FA[Q1]=500	
N15 X400	
N20 WAITP(Q1)	; プログラムの処理は、Q1 がそのインポジションになるまでに自動的に停止します。
N25 X600 POS[Q1]=300	; Q1 はタイプ 1 の位置決め軸です (ブロック N10 の送り速度 FA[Q1])。
N30 X800 Q1=500	; Q1 は軌跡軸です (ブロック N10 の軌跡送り F1000)。

## 工具オフセット

位置決め軸の工具長補正は、軸のゼロオフセットを使用して実行できます。たとえばローダの位置決め軌跡を変更できます。工具長補正の代わりに軸のゼロオフセットを使用する例として、様々な寸法の工具を備えたローダが障害物をバイパスすることを例に挙げることができます。

## プログラムの終了

プログラムの終了(選択されたプログラムの状態)は、すべての軸(軌跡軸と位置決め軸)がプログラムされている終点に達するまで遅延されます。

## 12.5.2 毎回転送り速度の外部プログラミング

次の2つのセッティングデータを使用し、位置決め軸の毎回転送り速度を別の回転軸/主軸から得ることを指定できます。

SD43300 \$SA\_ASSIGN\_FEED\_PER\_REV\_SOURCE(位置決め軸/主軸の毎回転送り速度)

SD42600 JOG\_FEED\_PER\_REV\_SOURCE (JOG の毎回転送り速度の制御)

下記の設定が可能です。

値	説明
0	毎回転送り速度が選択されていない。
>0	ここに指定した機械軸インデックスで回転軸/主軸から毎回転送り速度が得られます。
-1	軸/主軸が有効なチャンネルのメイン主軸から毎回転送り速度が得られます。
-2	機械軸インデックス 0 の回転軸/主軸から毎回転送り速度が得られます。
-3	軸/主軸が有効なチャンネルのメイン主軸から毎回転送り速度が得られます。メイン主軸が停止状態である場合、毎回転送り速度は選択されません。

## 12.6 ブロック切り替え

軌跡軸と位置決め軸が個別に補間されるので、異なるタイミングでプログラム指令された終了位置に到達します。軌跡軸と位置決め軸をブロック内でまとめてプログラム指令する場合、ブロック切り替え動作は指令できる位置決め軸のタイプに応じて異なります。

### タイプ 1: ブロック単位の位置決め軸

機能:

- **すべての軌跡軸と位置決め軸**が対応するプログラム指令された動作終了条件に達するとすぐに、ブロック切り替えがおこなわれます。
  - 軌跡軸: G601、G602、G603
  - 位置決め軸: FINEA、COARSA、IPOENDA
- 位置決め軸のプログラミング: POS [<軸>]

12.6 ブロック切り替え

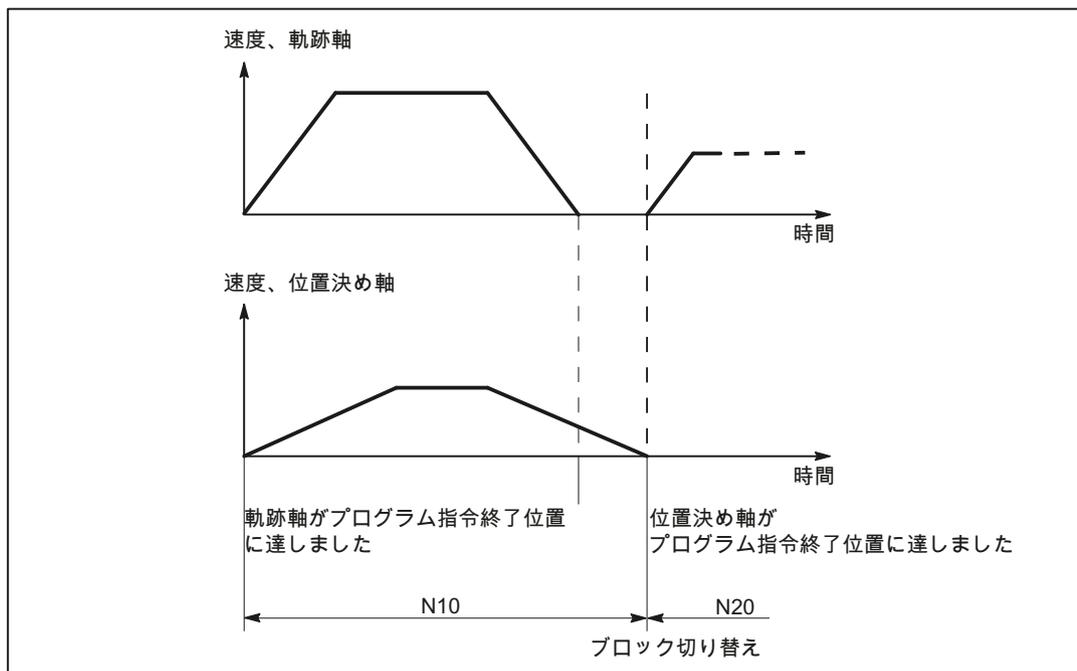


図 12-1 軌跡軸と位置決め軸のブロック切り替え タイプ 1

注記

連続軌跡モード

位置決め軸が軌跡軸の前に動作終了条件に達した場合にのみ、ブロック境界(G64)を超えた連続軌跡モードが可能です(上記の図では、これは該当しません)。

タイプ 2: モーダル位置決め軸(ブロック間)

機能:

- すべての軌跡軸がプログラム指令された動作終了条件に達するとすぐに、ブロック切り替えがおこなわれます(G601、G602、G603)。
- 位置決め軸のプログラミング: POSA [<軸>]
- 位置決め軸は、ブロック境界を超えてプログラム指令された終了位置に移動します。動作終了条件に達する前に、位置決め軸を再度プログラム指令することはできません。

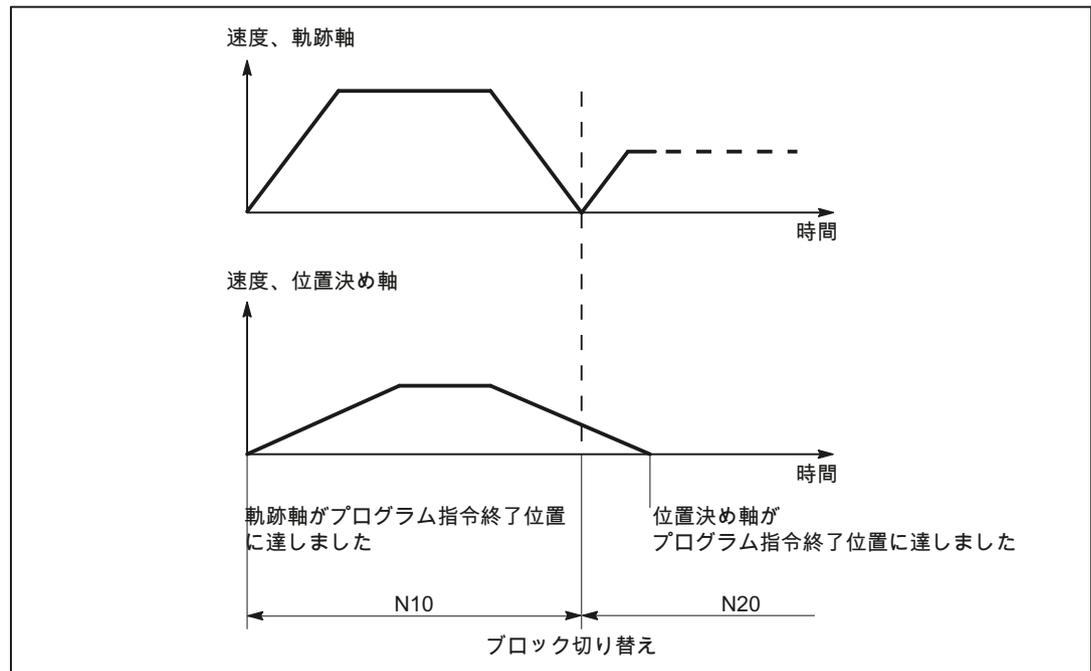


図 12-2 軌跡軸と位置決め軸のブロック切り替え タイプ 2

### 12.6.1 設定可能なブロック切り替えタイミング

#### タイプ 3: 条件付きブロック単位位置決め軸

機能:

- すべての軌跡軸と位置決め軸が対応するプログラム指令された動作終了条件に達するとすぐに、ブロック切り替えがおこなわれます。
  - 軌跡軸: G601、G602、G603
  - 位置決め軸: IPOBRKA
- 位置決め軸のプログラミング:
  - N(x) IPOBRK (<軸> [, <タイミング>]) ; 自身のブロック
  - N(x+1) POS [<軸>]

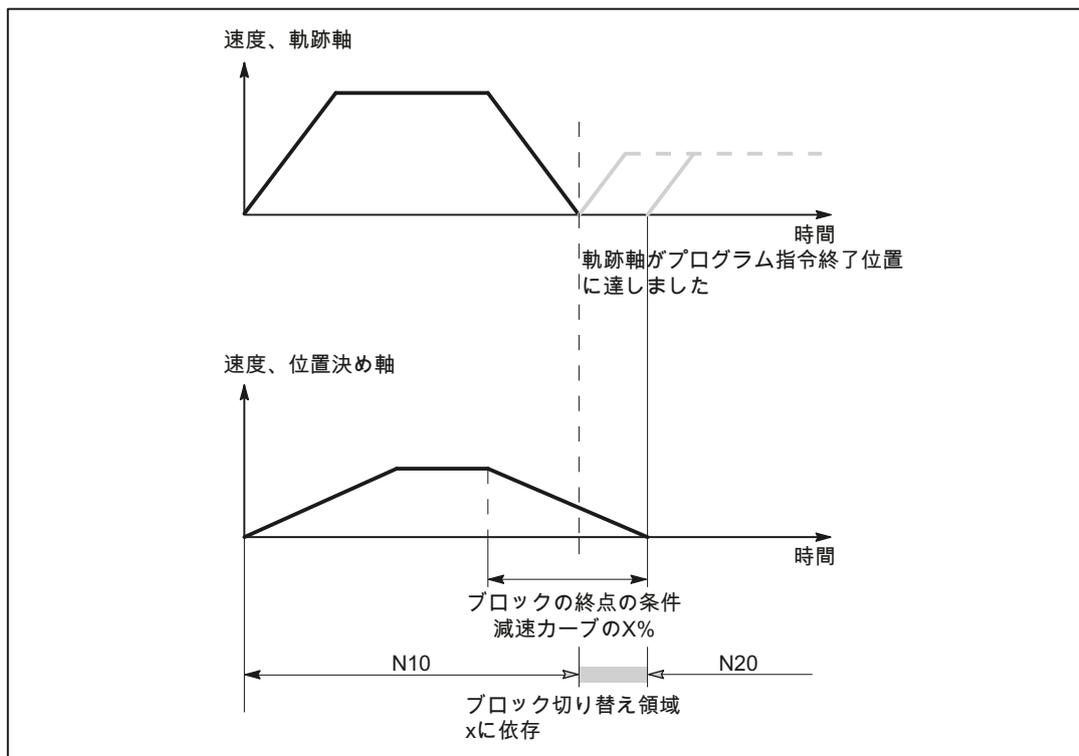


図 12-3 軌跡軸と位置決め軸のブロック切り替え タイプ 3

### ブロック切り替え条件: 「減速カーブ」 (IPOBRKA)

ブロック切り替え条件「減速カーブ」を有効にするときに、任意選択のパラメータ<タイミング>の値がプログラム指令されている場合、この値は次の位置決め動作に対して有効になり、メインランに同期してセッティングデータに書き込まれます。ブロック切り替え点のタイミングの値が指定されていない場合、セッティングデータの現在値が有効になります。

#### SD43600 \$SA\_IPOBRAKE\_BLOCK\_EXCHANGE

ブロック切り替えを実行できるタイミングを減速カーブのパーセント値で指定します。

- 100% = 減速カーブの開始
- 0% = 減速カーブの終了、ブロック切り替え条件 IPOENDA と同じ意味です。

#### プログラミング

IPOBRKA (<軸> [, <タイミング>])

IPOBRKA:                   ブロック切り替え条件: 減速カーブ  
効果:                        モーダル

<軸>:	チャンネル軸名称(X、Y、...)
<タイミング>:	ブロック切り替えのタイミング、減速カーブを基準にして% で表わします。 <ul style="list-style-type: none"> <li>● 100% = 減速カーブの開始</li> <li>● 0% = 減速カーブの終了、IPOENDA と同じ意味です。</li> </ul>
タイプ:	REAL

IPOBRKA は、軸の動作終了条件(FINEA、COARSEA、IPOENDA)がその次に軸へプログラム指令されると、対応するアクセスに対して無効になります。

### 追加のブロック切り替え条件: 「許容範囲」 (ADISPOSA)

ブロック終点周囲の許容範囲(現在位置または指令位置)を追加のブロック切り替え条件として定義できます。ブロック切り替えに対して以下の2つの条件を満たしてください。

- ブロック切り替え条件: 「減速カーブ」
- ブロック切り替え条件: 「許容範囲」

#### プログラミング

ADISPOSA (<軸> [, <モード>, <範囲のサイズ>])

ADISPOSA:	動作終了条件の許容範囲
効果:	モーダル
<軸>:	チャンネル軸名称(X、Y、...)
<モード>:	許容範囲の参照
タイプ:	INT
数値の範囲:	0 許容範囲は無効 1 指令位置に関する許容範囲 2 実位置の許容範囲
<範囲のサイズ>:	許容範囲のサイズ
タイプ:	REAL

### 動作終了条件用のシステム変数

有効な動作終了条件を、システム変数\$AA\_MOTENDを使用して読み取ることができます。

参照先:『パラメータマニュアル、第2巻』

---

**注記**

その他の指令可能な動作終了条件 FINEA、COARSA、IPOENDA に関する情報については、以下を参照してください。

参照先:『機能マニュアル、基本機能』

- 主軸(S1)、「主軸モード」の章
  - 送り速度(V1)、単独軸区間の指令可能なダイナミック応答
- 

**補足条件**

**早期のブロック変更**

次の場合には、早期のブロック変更はできません。

- 揺動軸  
区間切り込みがある軸が終了位置に達するまで、ブロック毎の揺動動作は、そのまま有効でなければいけません。
- 手動パルス発生器  
手動パルス発生器の入力では、最後に設定した動作終了条件が適用されます。

**軸状態の変更**

ブロック切り替えが減速カーブ内で発生する軸は、同じ軸ステータスの下記のブロックのみでプログラム指令できます。

軸状態が変更された場合(たとえば、POS、次に SPOS に変更された場合)、最後にプログラム指令された動作終了条件 FINEA、COARSEA、IPOENDA が有効になります。これはまた、次の場合に該当します。

- 位置決め軸は軌跡軸になります
  - 位置決め移動の終了までプログラムが待機する場合: WAITP、M30、テクノロジーサイクル終了、先読み停止
  - 速度オーバーライドが無効または有効になります
- 

**注記**

位置決め軸のプログラミングについて詳しくは、下記を参照してください。

**参照先:**

『プログラミングマニュアル、基本編』、「送り速度制御と主軸動作」の章  
『プログラミングマニュアル、上級編』、「特殊動作命令」の章

---

## 例

## ブロック切り替え条件、パートプログラムでの「減速カーブ」

プログラムコード	コメント
	; 初期設定が有効です。
N10 POS [X]=100	; X から位置 100 への位置決め動作です。
	; ブロック切り替え条件: 「精密イグザクトストップ」
N20 IPOBRKA (X, 100)	; ブロック切り替え条件: 「減速カーブ」、100% = 減速カーブの開始。
N30 POS [X]=200	; X 軸が減速を始めると同時にブロックが変更されます。
N40 POS [X]=250	; X 軸が位置 200 で停止せず、さらに位置 250 へと移動します。X 軸が減速を始めると同時にブロックが変更されます。
N50 POS [X]=0	; X 軸が減速して位置 0 に戻り 、位置 0 と「精密イグザクトストップ」でブロックが変更されます。
N60 X10 F100	; X 軸が軌跡軸として位置 10 に移動します。
N70 M30	

## ブロック切り替え条件、シンクロナイズドアクションの「減速カーブ」

テクノロジーサイクル:

プログラムコード	コメント
FINEA	; 動作終了条件: 「精密イグザクトストップ」
N10 POS [X]=100	; X 軸が位置 100 に到達し、精密イグザクトストップに達した場合にテクノロジーサイクルブロックの変更が実行されます。
N20 IPOBRKA (X, 100)	; ブロック変更条件、「減速カーブ」の有効化、 100% = 減速カーブの開始。
N30 POS [X]=200	; X 軸が減速を始めると同時にテクノロジーサイクルブロックが変更されます。
N40 POS [X]=250	; X 軸が位置 200 で減速せず、さらに位置 250 へと移動します。X 軸が減速を始めると同時に、テクノロジーサイクルのブロックの変更が実行されます。
N50 POS [X]=0	; X 軸が減速して位置 0 に戻り 、位置 0 と「精密イグザクトストップ」でブロックが変更されます。
N60 M17	

## パートプログラムでのブロック変更条件「減速カーブ」と「許容範囲」

プログラムコード	コメント
	; 初期設定が有効です。
N10 POS [X]=100	; X から位置 100 への位置決め動作です。
	; ブロック切り替え条件: 「精密イグザクトストップ」
N20 IPOBRKA (X, 100)	; ブロック切り替え条件: 「減速カーブ」、100% = 減速カーブの開始。
N21 ADISPOSA (X, 1, 0.5)	; 許容範囲: 1 = 指令位置、許容範囲 = 0.5
N30 POS [X]=200	; X 軸が減速を始めると同時にブロックが変更されます。

## 12.6 ブロック切り替え

プログラムコード	コメント
N40 POS[X]=250	; X軸が位置 200 で停止せず、さらに位置 250 へと移動します。X軸が減速を始めると同時にブロックが変更されます。
N50 POS[X]=0	; X軸が減速して位置 0 に戻り、位置 0 と「精密イグザクトストップ」でブロックが変更されます。
N60 X10 F100	
N70 M30	

## ブロック切り替え条件、シンクロナイズドアクションの「減速カーブ」と「許容範囲」

テクノロジーサイクル:

プログラムコード	コメント
FINEA	; 動作終了条件:「精密イグザクトストップ」
N10 POS[X]=100	; X軸が位置 100 に到達し、精密イグザクトストップに達した場合にテクノロジーサイクルブロックの変更が実行されます。
N20 IPOBRKA(X,100)	; ブロック変更条件、「減速カーブ」の有効化、100% = 減速カーブの開始。
N21 ADISPOSA(X,2,0.3)	; 許容範囲: 2 = 実位置、許容範囲 = 0.3
N30 POS[X]=200	; X軸が減速を始めて、軸の実位置 $\geq 199.7$ になると同時に、テクノロジーサイクルブロックの変更が実行されます。
N40 POS[X]=250	; X軸が位置 200 で減速せず、さらに位置 250 へと移動します。X軸が減速を始めて、X軸 $\geq 249.7$ になると同時にテクノロジーサイクルのブロックの変更が実行されます。
N50 POS[X]=0	; X軸が減速して位置 0 に戻り、位置 0 と「精密イグザクトストップ」でブロックが変更されます。
N60 M17	

対応する値がすでにセッティングデータに入力されている場合、タイミングを指定しなくても、N20 ブロックに IPOBRKA(X) を書き込むことができます。

SD43600 \$SA\_IPOBRAKE\_BLOCK\_EXCHANGE[X] == 100

下記も参照

PLC による制御 (ページ 834)

## 12.6.2 ブロック検索の動作終了条件

最終ブロックはコンテナとして使用されます。

軸にプログラム指令する最後の動作終了条件を収集し、アクションブロックで出力します。検索実行で処理された動作終了条件がプログラム指令されている最後のブロックは、すべての軸を設定するためのコンテナの役割を果たします。

### 例

3 軸の動作終了条件をもつ 2 つの処理ブロックの場合:

プログラムコード	コメント
N01 G01 POS[X]=20 POS[Y]=30 IPOENDA[X]	; 最後にプログラム指令された IPOENDA の動作終了条件
N02 IPOBRKA(Y, 50)	; Y 軸の 2 番目のアクションブロック IPOENDA
N03 POS[Z]=55 FINEA[Z]	; Z 軸の 2 番目のアクションブロック FINEA
N04 \$A_OUT[1]=1	; デジタル出力として出力する 1 番目のアクションブロック
N05 POS[X]=100	;
N06 IPOBRKA(X, 100)	; X 軸の 2 番目のアクションブロック IPOBRKA
...	; すべてのプログラム指令された動作終了条件について
TARGET:	; ブロック検索目標

1 番目の処理ブロックには、次のデジタル出力が含まれます。

$\$A\_OUT[1] = 1$ 。

2 番目の処理ブロックには、次の動作終了条件の設定内容が含まれます。

X 軸用 IPOBRKA /  $\$SA\_IPOBRAKE\_BLOCK\_EXCHANGE[AX1]=100$

Y 軸用 IPOBRKA /  $\$SA\_IPOBRAKE\_BLOCK\_EXCHANGE[AX2]=50$

Z 軸用 FINEA。最後にプログラム指令した動作終了条件 IPOENDA も、X 軸用に保存されます。

## 12.7 PLC による制御

### PLC 軸

PLC 軸は PLC から移動されて、他のすべての軸と同期しないで移動できます。移動動作は、軌跡とシンクロナイズドアクションは別におこなわれます。

#### 参照先:

『機能マニュアル、基本機能』、「SINUMERIK 840D sl の PLC 基本プログラム」(P3)または「SINUMERIK 828D の PLC(P4)」

### 同時位置決め軸

SINUMERIK 840D sl では、ファンクションブロック FC18 を使用して PLC から同時位置決め軸を開始できます。

### チャンネル別信号

すべてのチャンネル別信号は、軌跡軸と位置決め軸について同じように動作します。

後述の信号のみは例外です。

- NST DB21、... DBB4 (「送り速度オーバーライド」)
- IS DB21、... DBX6.2 (「残移動距離削除」)

### 軸別の信号

位置決め軸には次の追加信号があります。

- IS DB31、... DBX76.5 (「位置決め軸」)
- 位置決め軸/主軸の送り速度(FA)
- NST DB31、... DBB0 (「送り速度オーバーライド」) - 軸別
- IS DB31、... DBX2.2 (「残移動距離削除/主軸リセット」) 軸別の残移動距離削除

## PLC 制御軸の単独軸機能

個々の PLC 軸の動作は、マシンデータを使用して次の方法で操作できます。

MD30460 \$MA\_BASE\_FUNCTION\_MASK

:

- ビット 4 = 1  
軸は PLC のみで制御されます。
- ビット 5 = 1  
軸は PLC に完全に固定された軸です。
- ビット 6 = 1  
チャンネル別 NC/PLC インタフェース信号:  
これが PLC 制御軸でない場合、DB21、... DBX6.0 (「送り無効」)  
は軸に対して有効になりません。
- ビット 7 = 1  
チャンネル別 NC/PLC インタフェース信号:  
この軸が PLC 制御軸の場合、軸に関係なく DB21、... DBX36.3 (「すべての軸が停止」)  
が設定されます。

PLC 制御軸の場合:

- チャンネル別 NC/PLC インタフェース信号 DB21、... DBX6.0 (「送り無効」)は、マシンデータ MD30460 \$MA\_BASE\_FUNCTION\_MASK でビット 6 = 0 の場合に有効です。
- チャンネル別 NC/PLC インタフェース信号 DB21、... DBX36.3 (「すべての軸が停止」)は、マシンデータ MD30460 \$MA\_BASE\_FUNCTION\_MASK でビット 7 = 0 の場合のみ設定されます。

専用 PLC 制御軸を NC プログラムに割り当てようとした場合、または NC プログラムの軸を要求した場合は、アラーム 26075 が発生して拒否されます。アラーム 26076 は、固定割り当て PLC 軸の場合と同様に発生します。

PLC に完全に固定された軸は、電源投入時に「中立軸」になります。NC/PLC インタフェース経由の移動要求の場合、同時位置決め軸は、事前入れ替えなしで自動的に PLC 軸に変更されます。

## PLC による軸入れ替え

軸入れ替え用の軸のタイプは、軸のインタフェースバイト NCK→PLC IS DB31、... DBB68 で PLC に転送されます(「K5:相互チャネルプログラム協調とチャネルごとの試し運転(ページ 431)」の章も参照してください)。

- IS DB31、... DBX68.0-68.3 (「チャネルの NC 軸/主軸」) チャネル 1 ~ 10
- IS DB31、... DBX68.4 (「PLC 要求の新タイプ」)
- IS DB31、... DBX68.5 (「軸入れ替え可能」)
- IS DB31、... DBX68.6 (「中立軸/主軸」)
- IS DB31、... DBX68.7 (「PLC 軸/主軸」)

IS DB31、... DBX68.5 (「軸入れ替え可能」) = 1 である場合は、軸入れ替え要求を PLC から発行できます。

### 12.7.1 PLC からの同時位置決め軸の開始

PLC ユーザープログラムから競合する位置決め軸として NC の軸を移動するには、ブロック FC18 を使用します。

次の機能が定義されます。

- 直線補間(G01)
- mm/min 単位または度/min 単位の送り速度(G94)
- イグザクトストップ(G09)
- 有効な設定できるゼロオフセットが考慮されます

それぞれの軸は正確に 1 チャネルに割り当てられるので、制御装置は軸名称や軸番号から正しいチャネルを選択し、そのチャネルで同時位置決め軸を起動できます。

#### 参照先

『機能マニュアル 基本機能』(P3)「SINUMERIK 840D si の PLC 基本プログラム」> 「ブロックの説明」> 「FC18:SpinCtrl 主軸制御」

## 12.7.2 PLC 制御軸

### PLC 処理

以下の表では、以下の PLC 処理を機械軸に対する NC 応答と対比しています。

- FC18 によって PLC 軸として機械軸を起動する
- NC スタートまたは NC ストップを開始する
- 軸別の STOP、RESUME または RESET
- NC-RESET を起動する
- 機械軸でコントローライネーブルをキャンセル、または設定する
- 機械軸の制御を NC に渡す

### NC 応答の例

次の表は、NC の応答としての PLC 処理を示します。

PLC 処理	NC 動作
機械軸 AX1 はチャンネル 1 のチャンネル軸、FC18 を使って PLC 軸として起動	
「NC ストップ軸と主軸」の起動 DB21、... DBX7.4 = 1 (「NC ストップ軸と主軸」)	AX1 が停止されます。
DB21、... DBX7.1 = 1 開始(NC スタート)	AX1 は移動を継続します。
PLC が AX1 の制御を要求 DB31、... DBX28.7 = 1 (PLC が軸を制御)	PLC に対して AX1 が解除されていることを確認します。DB31、... DBX63.1 = 1(PLC が軸を制御します)
「軸と主軸の NC ストップ」の起動 DB21、... DBX7.4 = 1 (「NC ストップ軸と主軸」)	AX1 は停止されません。
軸停止の開始 DB31、... DBX28.6 = 1 (減速カーブに沿って停止)	AX1 が停止されます。 DB31、... DBX63.2 == 1 (軸停止有効)
軸継続の開始 DB31、... DBX28.2 = 1 (継続)	AX1 がさらに移動 DB31、... DBX63.2 == 0 (軸停止有効)

PLC 処理	NC 動作
NC RESET を開始 DB21、... DBX7.7 = 1 開始(リセット)	AX1 に対して無効
軸リセットを開始 DB31、... DBX28.1 = 1 (リセット)	AX1 が停止され、移動が中断されます: <ul style="list-style-type: none"> <li>● DB31、... DBX63.2 = 0 (軸停止有効)</li> <li>● 軸マシンデータの読み取り</li> <li>● DB31、... DBX63.0 = 1 (リセット実行)</li> <li>● DB31、... DBX63.2 = 0 (軸停止有効)</li> </ul>
機械軸 AX1 を PLC 軸として FC 18 によって起動	DB31、... DBX63.0 = 0 (リセット実行)
AX1 のコントローライネーブルを解除: DB31、... DBX2.1 = 0 (コントローライネーブル)	アラーム 21612 「軸 %1 検出器変更」を表示。
軸継続の開始 DB31、... DBX28.2 = 1 (継続)	<ul style="list-style-type: none"> <li>● アラーム 21612 「軸 %1 検出器変更」を解除。</li> <li>● DB21、... DBX40.7 = 1(移動指令プラス)</li> <li>● AX1 は、コントローライネーブル信号がないために移動しません。</li> </ul>
AX1 のコントローライネーブルを設定: DB31、... DBX2.1 = 1 (コントローライネーブル)	AX1 は、プログラム指令された終点に移動します。
軸リセットを開始 DB31、... DBX28.1 = 1 (リセット)	<ul style="list-style-type: none"> <li>● ストップ AX1</li> <li>● 軸マシンデータの読み取り</li> <li>● DB31、... DBX63.0 = 0 (リセット)</li> </ul>
PLC は、DB31、... DBX28.7 = 0 から NC に、AX1 の制御を渡します	<ul style="list-style-type: none"> <li>● NC は、機械軸の制御を受け入れます</li> <li>● DB31、... DBX63.1 = 0(PLC が軸を制御します)</li> <li>● DB31、... DBX63.0 = 0 (リセット)</li> </ul>

### 12.7.3 PLC 制御軸の応答制御

チャンネルリセットに対する応答、マシンデータ、ブロック検索、および MD30460 の有効化

表 12-1

応答制御	PLC 制御軸
モード変更と NC プログラム制御	軸とは無関係に動作する。
チャンネル RESET	軸マシンデータは無効で、移動動作は中止されない。
NEWCONF	軸マシンデータは無効。
ブロックサーチ タイプ 5 SERUPRO	SERUPRO 中に処理され、正常動作をシミュレートする。たとえば PLC は、PLC 経由でも移動するこの軸の制御を受け取るか解除する。
タイプ 1、2、4、および 5 の方式のすべてのブロックサーチ	PLC はアプローチブロックの前に軸の制御を受け取り、この軸の位置決めをおこなう。
\$AC_ESR_TRIGGER で起動された NC 後退制御	動作は無効で、指定 PLC 制御軸のみで動作する。
マシンデータ: MD30460 \$MA_BASE_FUNKTION_MASK	PLC のみで制御しない
ビット 4 = 0	軸入れ替え命令 GET (軸) または AXTOCHAN(軸、チャンネル)で、NC プログラムの制御軸へ直接変更できない。* 軸入れ替えに関する注を参照してください。
ビット 4 = 1	NC プログラムに要求できない。NC プログラムかシンクロナイズドアクションからの GET や AXTOCHAN、または NC プログラムの軸のプログラミングは、アラーム 26075 で拒否される。
MD30460 \$MA_BASE_FUNKTION_MASK	PLC 制御軸の場合

応答制御	PLC 制御軸
ビット 6 = 1	チャンネル別 IS DB 21、... DBX6.0 (「送り無効」)は無効。送り無効が有効になっても、PLC 制御軸は停止せずに引き続き移動する。
ビット 7 = 1	IS DB 21、... DBX36.3 (「全軸停止」)が発生しても、この軸は考慮されない。そのチャンネルのその他すべての軸が停止し、PLC 制御軸のみが動作している場合でも、この信号は 1 を出力する。

**\* 軸入れ替えに関する注意**

「NC プログラム軸」による、この「中立軸」の入れ替えは、PLC が使用事例「軸の制御の解除」に従って軸の制御を実際に解除するまで実行されません。この軸入れ替えの待機は、HMI 操作パネルに表示されます。

## 12.8 応用機能の動作

### 12.8.1 ドライラン(DRY RUN)

プログラム指令した送り速度がドライラン送り速度より大きくない場合は、ドライラン送り速度も位置決め軸に有効です。

SD42100 \$SA\_DRY\_RUN\_FEED に入ったドライラン送りの適用は、SD42101 \$SA\_DRY\_RUN\_FEED\_MODE で制御できます。参照

参照先:

『機能マニュアル、基本機能』; 「送り速度(V1)」

### 12.8.2 シングルブロック

#### 位置決め軸タイプ 1

シングルブロックモードは、タイプ 1 の位置決め軸で有効です。

## 位置決め軸タイプ 2

タイプ 2 の位置決め軸は、シングルブロックモードでブロック境界を越えて継続します。

## 位置決め軸タイプ 3

タイプ 3 の位置決め軸も、シングルブロックモードでブロック境界を越えて継続します。

## 12.9 例

### 12.9.1 移動動作と補間機能

次の例では、2 つの位置決め軸 Q1 と Q2 が 2 つの個別の移動ユニットを表します。2 つの軸間に補間関係はありません。この例では、位置決め軸はタイプ 1 (N20 など) とタイプ 2 (N40 など) としてプログラム指令されています。

### プログラム例

プログラムコード
N10 G90 G01 G40 T0 D0 M3 S1000
N20 X100 F1000 POS[Q1]=200 POS[Q2]=50 FA[Q1]=500 FA[Q2]=2000
N30 POS[Q2]=80
N40 X200 POSA[Q1] = 300 POSA[Q2]=200] FA[Q1]=1500
N45 WAITP[Q2]
N50 X300 POSA[Q2]=300
N55 WAITP[Q1]
N60 POS[Q1]=350
N70 X400
N75 WAITP[Q1, Q2]
N80 G91 X100 POS[Q1]=150 POS[Q2]=80
N90 M30

12.10 データリスト

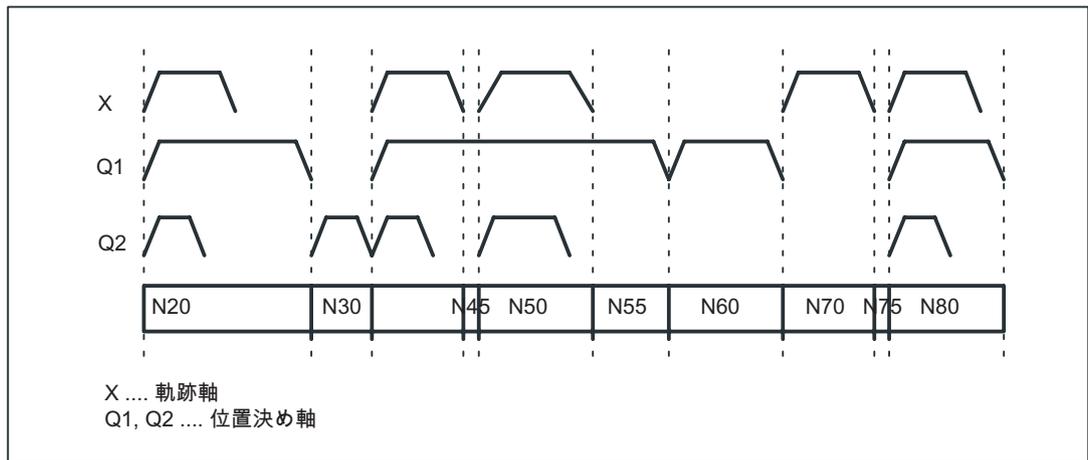


図 12-4 軌跡軸と位置決め軸のタイミング

## 12.10 データリスト

### 12.10.1 マシンデータ

#### 12.10.1.1 NC 別マシンデータ

番号	識別子:\$MN_	説明
18960	POS_DYN_MODE	位置決め軸のダイナミック応答のタイプ

#### 12.10.1.2 チャンネル別マシンデータ

番号	識別子:\$MC_	説明
20730	G0_LINEAR_MODE	G0 の補間動作
20732	EXTERN_G0_LINEAR_MODE	G00 の補間動作
22240	AUXFU_F_SYNC_TYPE	F 機能の出力タイミング

### 12.10.1.3 軸/主軸別マシンデータ

番号	識別子: \$MA_	説明
30450	IS_CONCURRENT_POS_AX	同時位置決め軸
30460	BASE_FUNCTION_MASK	軸機能
32060	POS_AX_VELO	位置決め軸の送り速度
32300	MAX_AX_ACCEL	最大軸加速度
32430	JOG_AND_POS_MAX_JERK	位置決め軸移動の最大軸加々速度
32431	MAX_AX_JERK	軌跡移動の最大軸加々速度
37510	AX_ESR_DELAY_TIME1	ESR 単独軸の遅延時間
37511	AX_ESR_DELAY_TIME2	単独軸の補間減速の ESR 時間

### 12.10.2 セッティングデータ

#### 12.10.2.1 軸/主軸別セッティングデータ

番号	識別子: \$SA_	説明
43600	IPOBRAKE_BLOCK_EXCHANGE	減速カーブのブロック切り替え条件
43610	ADISPOSA_VALUE	減速カーブの許容範囲

### 12.10.3 信号

#### 12.10.3.1 チャネルへの信号

信号名称	SINUMERIK 840D sl	SINUMERIK 828D
送り無効	DB21, ... .DBX6.0	DB320x.DBX6.0
NC スタート	DB21, ... .DBX7.1	DB320x.DBX7.1
軸と主軸の NC ストップ	DB21, ... .DBX7.4	DB320x.DBX7.4
リセット	DB21, ... .DBX7.7	DB320x.DBX7.7

## 12.10 データリスト

## 12.10.3.2 チャンネルからの信号

信号名称	SINUMERIK 840D sl	SINUMERIK 828D
全軸停止	DB21, ... .DBX36.3	DB330x.DBX4.3
移動指令マイナス	DB21, ... .DBX40.6	DB330x.DBX1000.6
移動指令プラス	DB21, ... .DBX40.7	DB330x.DBX1000.7

## 12.10.3.3 軸/主軸への信号

信号名称	SINUMERIK 840D sl	SINUMERIK 828D
送り速度オーバーライド、軸別	DB31, ... .DBB0	DB380x.DBB0
コントローライネーブル	DB31, ... .DBX2.1	DB380x.DBX2.1
指定軸の残移動距離削除、または指定軸の主軸リセット	DB31, ... .DBX2.2	DB380x.DBX2.2
リセット	DB31, ... .DBX28.1	DB380x.DBX5004.1
継続	DB31, ... .DBX28.2	DB380x.DBX5004.2
減速カーブに沿って停止	DB31, ... .DBX28.6	DB380x.DBX5004.6
PLC 制御軸	DB31, ... .DBX28.7	DB380x.DBX5004.7

## 12.10.3.4 軸/主軸からの信号

信号名称	SINUMERIK 840D sl	SINUMERIK 828D
汎用イグザクトストップ	DB31, ... .DBX60.6	DB390x.DBX0.6
精密イグザクトストップ	DB31, ... .DBX60.7	DB390x.DBX0.7
軸のアラーム	DB31, ... .DBX61.1	DB390x.DBX1.1
軸準備完了(Axis Ready)	DB31, ... .DBX61.2	DB390x.DBX1.2
軸コンテナ回転有効	DB31, ... .DBX62.7	-
リセット実行	DB31, ... .DBX63.0	DB390x.DBX3.0
PLC 制御軸	DB31, ... .DBX63.1	DB390x.DBX3.1
軸停止有効	DB31, ... .DBX63.2	DB390x.DBX3.2
軸/主軸無効化が有効	DB31, ... .DBX63.3	DB390x.DBX3.3
移動指令マイナス	DB31, ... .DBX64.6	DB390x.DBX4.6

信号名称	SINUMERIK 840D sl	SINUMERIK 828D
移動指令プラス	DB31, ... .DBX64.7	DB390x.DBX4.7
位置決め軸	DB31, ... .DBX76.5	DB390x.DBX1002.5
位置決め軸の F 機能(送り速度)	DB31, ... .DBB78-81	-
緊急リトラクション有効	DB31, ... .DBX98.7	DB390x.DBX5002.7



## P5 :揺動

### 13.1 概略説明

#### 定義

揺動機能を選択すると、揺動軸は、2つの反転点の間で、プログラム指令された送り速度、または得られた送り速度(毎回転送り速度)で、前後に揺動します。複数の揺動軸を同時に有効にすることができます。

#### 揺動方式

揺動機能は、反転点の軸動作と切り込みに従って分類できます。

- ブロック境界を越えた非同期揺動  
往復動作中に、その他の軸が任意に補間できます。揺動軸は、ダイナミック座標変換の入力軸として、またはガントリー軸、または連結動作軸のマスター軸として動作できます。揺動は自動的にAUTOMATICモードへ接続しません。
- 連続切り込みの揺動  
複数の軸で同時に切り込むことができます。しかし、切り込み移動と揺動移動の間に補間関係はありません。
- 両方の反転点の切り込み、または左側か右側のみの反転点の切り込みの揺動 反転点からの距離を指令して切り込みを開始できます。
- 揺動後のスパークアウトストローク
- 定義位置の揺動の開始と終了

#### 反転点の動作

方向の変更は次のように開始されます。

- イグザクトストップ範囲に達していない状態(精密イグザクトストップまたは汎用イグザクトストップ)で開始
- プログラム指令位置へ到達後に開始 または
- プログラム指令位置へ到達し、ドウェル終了後に開始
- 外部信号(PLC から)による開始

## 制御方法

揺動移動は、次のように様々な方法で制御できます。

- 揺動移動と(または)切り込みは残移動距離削除によって中断できます。
- 反転点は、NC プログラム、PLC、HMI、ハンドル、方向キーのいずれかによって変更できます。
- 揺動軸の送り速度速度は、NC プログラム、PLC、HMI の値入力、またはオーバーライドによって変更できます。送り速度は、メイン主軸、回転軸、主軸のいずれかによってプログラム指令できます(毎回転送り速度)。

### 参照先:

『機能マニュアル、基本機能』; 「送り速度(V1)」

- 揺動移動はすべて PLC で制御できます。

## 揺動制御の方法

揺動には次の 2 つのモードがあります。

1. 非同期揺動:  
ブロック境界を越えて有効であり、PLC/HMI から開始することもできます。
2. シンクロナイズドアクション移動による揺動:  
非同期揺動と切り込み移動がシンクロナイズドアクションによって組み合わせが可能です。この方法では、反転点で切り込みとともに揺動をプログラム指令して、ノンモードベースで有効にすることができます。

## 13.2 非同期揺動

### 特徴

非同期揺動の特徴は次のとおりです。

- 揺動軸は、揺動移動が無効になるか、必要条件に適切な応答があるまで、指定送り速度で反転点の間を前後に揺動します。揺動軸は、移動の開始時に反転点 1 で位置決めされていない場合、最初にこの点まで移動します。
- 揺動軸では、プログラムで有効な G 命令に関係なく、直線補間 G01 が有効です。代わりに、毎回転送り速度 G95 を有効にすることもできます。
- 非同期揺動は、軸毎にブロック境界を超えて有効です。
- 複数の揺動軸(位置決め軸の最大軸数など)を同時に有効にすることができます。

- 揺動移動中は、揺動軸以外の軸を自由に補間できます。連続切り込みは、軌跡移動または位置決め軸によっておこなうことができます。この場合、揺動移動と切り込み移動の間に補間関係はありません。
- PLC が軸を制御していない場合、非同期揺動中に、軸は通常の位置決め軸のように扱われます。PLC 制御の場合、PLC プログラムは、NC/PLC インタフェースの当該のストップビットによって、適切な方法で、その軸を NC/PLC インタフェース信号へ応答してください。この NC/PLC インタフェース信号には、エンドオブプログラム、運転モード変更、およびシングルブロックも含まれます。
- 揺動軸は座標変換用の入力軸(傾斜軸など。「M1:キネマティックトランスフォーメーション (ページ 485)」の章を参照してください)として動作できます。
- 揺動軸は、ガントリ軸と連結動作軸のマスタ軸として動作できます。  
**参照先:**  
『機能マニュアル、応用機能』; 「ガントリ軸(G1)」
- 加々速度制限(SOFT)および(または)膝形加速特性(位置決め軸に関して)。
- この他に、揺動移動はパートプログラムのブロックとの同期制御によって有効にすることができます。
- 揺動移動は、PLC/HMI からも同じように開始、動作、停止することができます。
- 補間揺動(傾斜揺動など)はできません。

### 13.2.1 非同期揺動の動作

#### セッティングデータ

揺動に必要なセッティングデータは、NC パートプログラムの特殊言語命令によって HMI および(または)PLC で設定できます。

## 送り速度

揺動軸の送り速度は、次のように選択するかプログラム指令します。

- 位置決め軸として軸に定義した速度を送り速度として使用します。この値はFA[軸]によってプログラム指令ができ、モーダル動作になります。速度をプログラム指令しない場合は、マシンデータ POS\_AX\_VELO に保存されている値が使用されます(「P2:位置決め軸 (ページ 803)」の章を参照してください)。
- 揺動移動の進行時に、揺動軸の送り速度をセッティングデータによって変更できます。これは、変更速度をすぐに有効にするか、次の反転点で有効にするかは、パートプログラムとセッティングデータによって指定できます。
- 送り速度は、オーバライド(軸の NC/PLC インタフェース信号およびプログラム指令可能)によって作用を受けます。
- ドライランが有効であり、現在プログラム指令されている速度より速い場合は、ドライラン速度設定が適用されます。

SD42100 \$SC\_DRY\_RUN\_FEED に入力されたドライラン送りの適用は、SD42101 \$SC\_DRY\_RUN\_FEED\_MODE で制御できます。

### 参照先:

『機能マニュアル、基本機能』; 「送り速度(V1)」

- 速度重畳/軌跡重畳はハンドルの作用を受けます(次の表と「H1:手動移動 (ページ 169)」の章を参照してください)。
- 揺動軸は反転送りで移動できます。

## 反転送り

揺動軸には反転送りも使用できます。

## 反転点

揺動移動を開始する前、または揺動移動の進行中に、反転点の位置をセッティングデータによって入力できます。

- 反転点の位置は、揺動移動が中断されたかどうかに関係なく、揺動移動の前、またはその進行中に手動移動(ハンドル、ジョグキー)によって入力できます。

下記は、反転点の位置の変更に適用されます。揺動移動がすでに進行中である場合は、反転点の位置を変更しても、この点に再びアプローチするまで、変更は有効になりませ

ん。軸がその位置にすでにアプローチしている場合、変更内容は次の揺動ストロークで有効になります。

#### 注記

反転点が NC/PLC インタフェース信号 DB21、... DBX0.3(「DRF の起動」)の設定と同時に変更された場合、ハンドル信号は DRF オフセットと反転点オフセットの両方に適用されます。すなわち反転点は、距離の 2 倍に相当する距離だけアブソリュートでシフトします。

## 停止時間

停止時間は、反転点毎にセッティングデータによってプログラム指令できます。

設定は、NC プログラムの次のブロックで変更できます。その後に、次の反転点からブロックの同期が有効になります。

停止時間はセッティングデータによって非同期に変更できます。その後、当該の反転点が次に移動された時点から有効になります。

次の表は、停止時間入力に応じた、イグザクトストップ範囲または反転点の動作を示しています。

表 13-1 停止時間の動作

停止時間設定	動作
-2	イグザクトストップを待機せずに補間が継続
-1	汎用イグザクトストップを反転点で待機
0	精密イグザクトストップを反転点で待機
> 0	精密イグザクトストップを反転点で待機してから停止時間を待機

## 揺動の解除

揺動モードを無効にするときは、次のうちいずれかのオプションを揺動移動の終了に設定できます。

- 次の反転点で揺動移動を終了
- 反転点 1 で揺動移動を終了
- 反転点 2 で揺動移動を終了

この終了処理に続いてスパークアウトストロークが処理され、プログラム指令されている場合は終了位置にアプローチします。

## 13.2 非同期揺動

非同期揺動からスパークアウトへの切り替え時およびスパークアウト中には、イグザクトストップに関する反転点の動作は、当該の反転点でプログラム指令されている停止時間で決まる動作に対応します。スパークアウトストロークは、別の反転点に向かう動作、および別の反転点に戻る動作です(以下の表を参照してください)。

## 注記

モーションシンクロナイズドアクションと停止時間「OST1/OST2」を含む揺動設定停止時間を過ぎると、揺動中に内部ブロック切り替えが実行されます(軸の新しい残移動距離によって示されます)。ブロック切り替え時には、解除機能がチェックされません。解除機能は、動作シーケンス「OSCTRL」の制御設定に従って定義されます。このダイナミック応答は送りオーバーライドの作用を受けます。スパークアウトストロークが開始されるか、終了位置にアプローチ前に、揺動ストロークが実行されることがあります。解除動作が変更されたようにみえますが、そうではありません。

表 13-2 揺動解除の動作シーケンス

機能	入力	説明
定義された反転点で解除	スパークアウトストローク数は0、 終了位置は無効	揺動移動は当該の反転点で停止
指定スパークアウトストローク数を解除	スパークアウトストローク数は0ではなく、 終了位置は無効	当該の反転点へ到達後、命令に指定されているスパークアウトストローク数が処理される。
スパークアウトストロークと定義された終了位置を解除(オプション)	スパークアウトストロークの数は0ではなく、 終了位置は有効	当該の反転点へ到達後、命令に指定されているスパークアウトストローク数が処理され、指定の終了位置にアプローチする。
スパークアウトストロークを含まないが、定義された終了位置で解除(オプション)	スパークアウトストローク数は0、 終了位置は有効	当該の反転点に到達後、指定の終了位置へ軸が移動する。

## NC 言語

NC プログラミング言語では、非同期揺動をパートプログラムから制御できます。次の機能では、NC プログラム実行に応じて、非同期揺動の有効化と制御ができます。

---

### 注記

セッティングデータをパートプログラムに直接書き込むと、データ変更はパートプログラムの処理によって早く有効になります(先読み)。先読み停止(STOPRE)を使用して、パートプログラムと揺動機能命令を再同期できます。

---

### 参照先:

『プログラミングマニュアル』

#### 1) 揺動の起動と解除:

- OS[揺動軸] = 1; 揺動軸の揺動を起動
- OS[揺動軸] = 0; 揺動軸の揺動を解除

---

### 注記

すべての軸を揺動軸として使用できます。

---

#### 2) 揺動の終了:

- WAITP(揺動軸)  
位置決め軸命令は – 揺動軸がきちんと停止するまでブロックを停止し、先読みと実行プログラムを同期化します。揺動軸は位置決め軸として再び入力され、通常どおりに使用できるようになります。  
軸を揺動に使用する場合は、WAITP(軸)命令で軸を事前に解放してください。  
揺動を PLC/HMI から開始する場合にも、これは当てはまります。この場合は、事前に、軸を NC プログラムによってプログラム指令すると、WAITP(軸)の呼び出し指令も必要になります。SW バージョン 3.2 の場合は、WAITP()をプログラミングでおこなうか自動的におこなうかをマシンデータ\$MA\_AUTO\_GET\_TYPEによって選択できます。

---

### 注記

WAITP では、揺動移動が実行されるまで、効果的に時間遅延が実行されます。移動の終了は、たとえば NC プログラムでプログラム指令した解除命令、または残移動距離削除によって、PLC、または HMI で開始されます。

---

#### 3) 反転点の設定:

- OSP1[揺動軸] = 反転点 1 の位置
- OSP2[揺動軸] = 反転点 2 の位置

## 13.2 非同期揺動

位置は実行プログラムのブロックとの同期制御で、当該のセッティングデータに入力されるので、次にセッティングデータが変更されるまで、そのまま有効です。

インCREMENTAL移動が有効の場合、位置は、NC プログラムでプログラム指令された最後の当該の反転点からの増分で計算されます。

## 4) 反転点の停止時間:

- OST1[揺動軸] = 反転点 1 の停止時間([s]単位)
- OST2[揺動軸] = 反転点 2 の停止時間([s]単位)

停止時間は実行プログラムのブロックとの同期制御で当該のセッティングデータに入力されるので、次にセッティングデータが変更されるまで、そのまま有効です。

停止時間の単位は、G04 でプログラム指令された停止時間の選択単位と同じです。

## 5) 送り速度の設定:

- FA[軸] = FValue  
位置決め軸の切り込み。

送り速度は、実行プログラムのブロックとの同期制御で当該のセッティングデータに伝送されます。揺動軸が反転送りで移動する場合は、機能 V1 の説明で述べたように、対応する従属機能を指定してください。

## 6) 動作シーケンス制御設定の設定内容:

- OSCTRL[揺動軸] = (設定オプション、リセットオプション)  
設定オプションは次のように定義されます(リセットオプションは設定内容を選択解除)。

表 13-3 設定/リセットオプション

オプション値	意味
0	揺動移動の解除で次の反転点で停止(初期設定)。オプション値 1 と値 2 のリセットのみで達成。
1	揺動移動の解除で反転点 1 で停止
2	揺動移動の解除で反転点 2 で停止
3	揺動移動の解除では、スパークアウトストロークがプログラムされていない場合、反転点へアプローチしない
4	スパークアウト処理後に終了位置へアプローチ
8	揺動移動が残移動距離削除によって中止された場合、スパークアウトストロークを実行して終了位置へアプローチする(プログラム指令されている場合)

オプション値	意味
16	揺動移動が残移動距離削除によって終了した場合、揺動移動の解除でプログラム指令された反転点へアプローチする
32	変更した送り速度は、次の反転点から有効。
64	送り速度設定が 0 である場合は、軌跡重畳が有効。0 でない場合は、速度重畳が有効。
128	回転軸 DC 用(最短軌跡)
256	単独ストロークとしてのスパークアウトストローク
512	最初に開始位置へアプローチ

### 注記

オプション値 0-3 では、電源断で反転点の動作が内部コード化されます。切り替え値 0 ～ 3 のいずれかを選択できます。その他の設定は、個々の必要条件に従って、選択した切り替え値と組み合わせることができます。文字+を挿入すると、オプション列を作成できます。

**例:**Z 軸の揺動移動は、解除時に反転点 1 で停止します。その後、終了位置にアプローチし、新しくプログラムした送り速度がすぐに有効になります。軸は、残移動距離削除直後に停止します。

**OSCTRL[Z] = (1+4, 16+32+64)**

設定/リセットオプションは実行プログラムのブロックとの同期制御で当該のセッティングデータに入力されるので、次にセッティングデータが変更されるまで、そのまま有効です。

### 注記

コントローラではリセットオプションが参照されてから設定オプションが参照されます。

#### 7) スパークアウトストローク:

- OSNSC[揺動軸] = スパークアウトストローク数

スパークアウトストローク数は実行プログラムのブロックとの同期制御で当該のセッティングデータに入力されるので、次にセッティングデータが変更されるまで、そのまま有効です。

#### 8) 揺動の解除後にアプローチする終了位置:

- OSE[揺動軸] = 揺動軸の終了位置

## 13.2 非同期揺動

終了位置は実行プログラムのブロックとの同期制御で当該のセッティングデータに入力されるので、次にセッティングデータが変更されるまで、そのまま有効です。オプション値 4 が自動的に設定され、設定された終了位置にアプローチします。

## 9) 揺動の起動の前にアプローチする開始位置:

- OSB [揺動軸] = 揺動軸の開始位置

開始位置は実行プログラムのブロックとの同期制御で当該のセッティングデータ SD43790 \$SA\_OSCILL\_START\_POS に入力されるので、次にセッティングデータが変更されるまで、そのまま有効です。セッティングデータ SD43770 \$SA\_OSCILL\_CTRL\_MASK のビット 9 は、開始位置へのアプローチを開始するように設定してください。反転点 1 の前に開始位置にアプローチします。開始位置が反転点 1 と同じである場合は、次に反転点 2 にアプローチします。

プログラミング命令 OSB の代わりに、セッティングデータ SD43790 \$SA\_OSCILL\_START\_POS に開始位置を直接入力することもできます。

セッティングデータとシステム変数のすべての位置情報は、基本座標系(BCS)を基準とします。OSB、OSE の位置データは、ワーク座標系(WCS)を基準とします。

開始位置に到達したとき、この位置が反転点 1 と同一であっても停止時間は適用されず、軸は精密イグザクトストップ信号を待機します。設定されているすべてのイグザクトストップ条件が満たされます。

開始位置が反転点 1 と同一で、ノンモーダル揺動処理が切り込み移動を必要としない場合は、このオプションを、別のシンクロナイズドアクションで設定できます(「ノンモーダル揺動(開始位置 = 反転点 1) (ページ 881)」の章を参照してください)。

## プログラミング例

非同期揺動のすべての重要要素を含む例については、「非同期揺動例 (ページ 874)」の章を参照してください。

## 13.2.2 PLC 制御による非同期揺動

## 適用

MDI Ref と JOG Ref を除くすべての運転モードで、以下のセッティングデータを使用して PLC から機能を選択できます。

SD43780 OSCILL\_IS\_ACTIVE(揺動動作オン)

## 設定

次の条件をセッティングデータによって PLC から制御できます。揺動移動の起動と解除、反転点の位置、反転点の停止時間、送り速度、反転点のオプション、スパークアウトストローク数、解除後の終了位置。また、HMI 操作画面で直接、または NC プログラムによって、これらの値をセッティングデータとして事前に設定することもできます。この設定は電源投入後に有効となり、PLC はさらにセッティングデータ **OSCILL\_IS\_ACTIVE** (変数サービス)によって、このようにして直接設定された揺動移動を開始することができます。

## 補足条件

PLC によって開始され、揺動移動を実行する軸として動作する主軸が、位置決め軸として移動するためには、必要条件を満たす必要があります。例えば、主軸が、事前に位置制御(SPOS)へ切り替えてください。

この軸は常に、PLC から制御されているかどうかにかかわらず、次の 2 つの停止ビットに応答します。

- DB31、... DBX28.5 (次の反転点で停止)
- DB31、... DBX28.6 (減速カーブに沿って停止)

### 13.2.3 非同期揺動中の特殊応答

#### PLC で制御する場合

PLC プログラムは、NC/PLC 信号によって揺動軸の制御を取ることができます。この NC/PLC インタフェース信号には、エンドオブプログラム、運転モード変更、およびシングルブロックも含まれます。

NCU システムソフトウェアを使用している場合、メインランによって補間された非同期揺動軸は、NCSTOP、アラーム処理、エンドオブプログラム、プログラム制御、および RESET に応答します。

軸/主軸は、IS DB31、... DBX28.7 (「PLC 制御軸」) = 1 によって、軸の NC/PLC インタフェース(PLC→NC)を経由して制御されます。

PLC が制御する軸について詳しくは、「P2:位置決め軸 (ページ 803)」の章を参照してください。

### PLC で制御しない場合

PLC が軸を制御していない場合、軸は非同期揺動中に通常の位置決め軸(POSA)のように取り扱われます。

### 残移動距離削除

チャンネル別の残移動距離削除は無視されます。

軸の残移動距離削除

- PLC で制御しない場合;減速カーブに沿って停止
- PLC で制御する場合;停止しない(PLC から開始する必要があります)

下記は**両方**の事例に適用されます。軸の停止後、必要に応じて、当該の反転点にアプローチし、残移動距離が削除されます。次に、スパークアウトストロークを実行して、終了位置へアプローチします。これが **OSCILL\_CTRL\_MASK** で設定されている場合揺動移動が完了します。

---

#### 注記

研削中は、軸の移動距離削除によってキャリパを動作させることができます。

---

### 非常停止

非常停止により揺動移動は完了し、再起動する必要があります。

### リセット

揺動移動は、減速カーブによって中断して解除されます。その後の選択オプションは処理されません(スパークアウトストローク、終了点へのアプローチ)。

### ワーキングエリアリミット、リミットスイッチ

揺動移動処理中に有効なリミット位置に違反することが検出された場合は、アラームが出力されて揺動移動は開始されません。

揺動移動が有効である間に、途中で有効になったリミット位置を揺動軸が超える(第2ソフトウェアリミットスイッチなど)場合は、軸が減速カーブで減速し、アラームが出力されます。

 <b>注意</b>
<b>プロテクションゾーン</b>
揺動移動ではプロテクションゾーンは作動しません。

### フォローアップモード

位置決め軸と同じです。

### プログラムの終了

PLC が軸を制御していない場合、揺動移動が終了するまでプログラムは終了しません (POSA の応答:

ブロック境界を越えた位置決め)。

PLC が軸を制御している場合、プログラムが終了しても軸は揺動を続けます。

### モード変更

次の表は、揺動を実行できる運転モードを示しています。揺動可能な運転モードへ切り替えは、揺動移動に影響しません。許容できない運転モードに切り替えようとする、アラームが発生して拒否されます。制御命令の適用中に、NC プログラムから、またはオペレータ入力(ジョグ)によって、これと同時に揺動モードの軸を移動することはできません。これを実行しようとする、アラームが出力されます。これは下記のように適用されます。最初に開始した移動のタイプが優先されます。

表 13-4 揺動を許可する運転モード

運転モード	揺動の可否
AUTO	あり
MDI	あり
MDI Repos	あり
MDI Teachin	あり
MDI Ref	なし
JOG	あり

## 13.2 非同期揺動

運転モード	揺動の可否
JOG Ref	なし
JOG Repos	あり

## シングルブロック処理

PLC が軸を制御していない場合、軸は位置決め軸(POSA)と同じ方法でシングルブロックに応答します。軸は移動し続けます。

## オーバーライド

オーバーライドは下記によって指定されます。

**NC/PLC インタフェース**

軸のオーバーライドは揺動軸で動作します。

**プログラミング**

オーバーライドは、位置決め軸と同じ方法で揺動軸で動作します。

## ブロック検索

ブロック検索の場合、最後に有効だった揺動機能が登録され、NC スタート後に直接(ブロック検索後に開始位置にアプローチしているとき)、またはブロック検索後に開始位置に到達したあとで、マシンデータ **OSCILL\_MODE\_MASK** が、それに応じて有効になります(初期設定)。

**OSCILL\_MODE\_MASK ビット 0:**

0:揺動は開始位置へ到達後に開始されます。

1:揺動は NC スタート直後に開始されます。

**REORG**

常に先に反転点 1 へアプローチしてから揺動が継続します。

**ASUB**

ASUB(非同期サブプログラム)の実行中は揺動移動を継続します。

## 13.3 シンクロナイズドアクションによる揺動制御

### 手順の概要

シンクロナイズドアクションによって、非同期揺動移動を切り込み移動と組み合わせ、それによって制御します。

ここでは、揺動機能に関連するモーションシンクロナイズドアクションのみに絞って説明します。

### 参照先

『機能マニュアル、シンクロナイズドアクション』

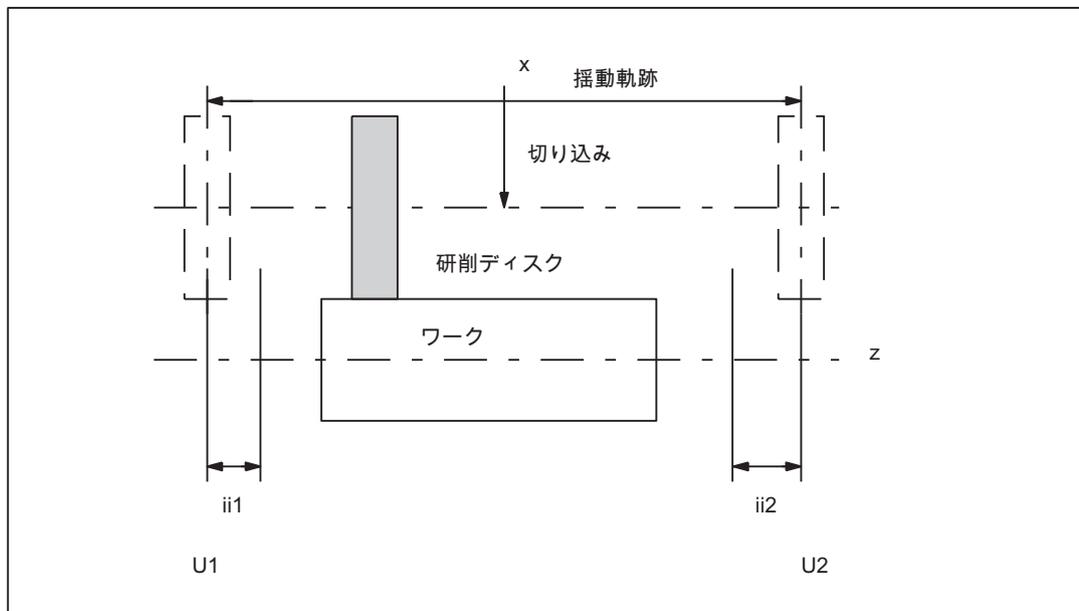
### 機能

以下に説明するコマンドで、次の機能を実行できます。

1. 反転点の切り込み(「反転点 1 または 2 の切り込み (ページ 865)」の章を参照してください)。
2. 反転範囲の切り込み(「反転点範囲の切り込み (ページ 866)」の章を参照してください)。
3. 両反転点の切り込み(「両方の反転点の切り込み (ページ 867)」の章を参照してください)。
4. 切り込みが終了するまで、反転点での揺動移動を停止します(「反転点で揺動移動を停止 (ページ 868)」の章を参照してください)。

13.3 シンクロナイズドアクションによる揺動制御

5. 揺動移動を有効にします(「揺動移動の再開 (ページ 869)」の章を参照してください)。
6. 早すぎる区間切り込みの起動を防止します(「区間切り込みの早期開始の防止 (ページ 870)」の章を参照してください)。



U1 反転点 1

U2 反転点 2

ii1 反転範囲 1

ii2 反転範囲 2

図 13-1 揺動軸の構成、切り込み軸

プログラミング

切り込み軸と揺動軸間の割り当て(「揺動軸と切り込み軸の割り当て OSCILL (ページ 871)」の章を参照してください)、切り込みの定義(POSP)、およびモーションシンクロナサクションを含む移動ブロックの前に、最初に揺動のパラメータを定義する必要があります(「非同期揺動の動作 (ページ 849)」の章を参照してください)。

WAITP [揺動軸] (MD30552 \$MA\_AUTO\_GET\_TYPE を参照してください)を使用して、揺動に使用する軸を有効にします。これで、揺動パラメータが同時にセッティングデー

タに伝送されます。**\$SA\_REVERSE\_POS1**などのシンボリック名称を使用してモーションシンクロナイズドアクションをプログラム指令できます。

---

#### 注記

**\$SA\_REVERSE\_POS**の値によるモーションシンクロナイズドアクションでは、**解釈時の比較値**が有効です。セッティングデータが後で変更されても、これは影響されません。

**\$AA\_REVERSE\_POS**の値によるモーションシンクロナイズドアクションでは、**補間内の比較値**が有効です。これにより、変更された反転位置の応答が確保されます。

---

- **モーションシンクロナイズドアクションの条件(周波数 WHEN / WHENEVER)**
- 移動ブロックによる起動
  - 揺動軸と切り込み軸をそれぞれ **OSCILL** に割り当てます。
  - 切り込み動作 **POSP** を指定します。

---

#### 注記

モーションシンクロナイズドアクションのアクション成分の軸のオーバーライドをゼロに設定する(**DO \$AA\_OVR[<軸>]=0**)と、関連する条件(周波数 **WHEN / WHENEVER**)を満たしている場合は、揺動/切り込み軸を停止できます。

オーバーライドが **100%**で、条件を満たさなくなった場合は、自動的に有効になります。

---

## 実行プログラムの評価

実行プログラム補間器クロック周期の同期条件を現在のフィードバック値(比較条件の右の**\$\$変数**)と比較できます。通常のシステム変数比較では、最初の実行で演算式が評価されます。

シンクロナイズドアクションの詳しい説明については、以下を参照してください。

#### 参照先

『機能マニュアル、シンクロナイズドアクション』

## 13.3 シンクロナイズドアクションによる揺動制御

## 例

## 例 1:変更されていない反転位置

モーションシンクロナイズドアクションでは、反転位置 $\$SA\_OSCILL\_REVERSE\_POS$ を先読みレベルで使用します。対応するセッティングデータを変更しても、変更された値はプログラムで有効になりません。

プログラムコード	コメント
...	
$\$SA\_OSCILL\_REVERSE\_POS1[Z]=-10$	; 先読み変数:違うとき
$\$SA\_OSCILL\_REVERSE\_POS2[Z]=10$	; 先読み変数:大きい、または等しいとき
G0 X0 Z0	
WAITP(Z)	
	; シンクロナイズドアクション 1:反転点 1 を下回った □
	; オーバーライド 0%で切り込み軸を停止
	; <b>先読み変数</b> 反転点 1 を使用
ID=1 WHENEVER $\$AA\_IM[Z] < \$SA\_OSCILL\_REVERSE\_POS1[Z]$ DO $\$AA\_OVR[X]=0$	
	; シンクロナイズドアクション 2:反転点 2 を超えた □
	; オーバーライド 0%で切り込み軸を停止
	; <b>先読み変数</b> 反転点 2 を使用
ID=2 WHENEVER $\$AA\_IM[Z] > \$SA\_OSCILL\_REVERSE\_POS2[Z]$ DO $\$AA\_OVR[X]=0$	
OS[Z]=1 FA[X]=1000 POS[X]=40	; 揺動の起動
OS[Z]=0	; 揺動の解除
M30	

## 例 2:反転位置の変更

モーションシンクロナイズドアクションでは、反転点として実行プログラム変数 $\$AA\_OSCILL\_REVERSE\_POSx$ を使用します。対応するセッティングデータを変更すると、変更された値がプログラムで有効になります。

プログラムコード	コメント
...	
$\$SA\_OSCILL\_REVERSE\_POS1[Z]=-10$	; 先読み変数:違うとき
$\$SA\_OSCILL\_REVERSE\_POS2[Z]=10$	; 先読み変数:大きい、または等しいとき
G0 X0 Z0	
WAITP(Z)	
	; シンクロナイズドアクション 1:反転点 1 を下回った □
	; オーバーライド 0%で切り込み軸を停止
	; <b>実行プログラム変数</b> 反転点 1 を使用
ID=1 WHENEVER $\$AA\_IM[Z] < \$\$SA\_OSCILL\_REVERSE\_POS1[Z]$ DO $\$AA\_OVR[X]=0$	
	; シンクロナイズドアクション 2:反転点 2 を超えた □
	; オーバーライド 0%で切り込み軸を停止
	; <b>実行プログラム変数</b> 反転点 2 を使用
ID=2 WHENEVER $\$AA\_IM[Z] > \$\$SA\_OSCILL\_REVERSE\_POS2[Z]$ DO $\$AA\_OVR[X]=0$	

プログラムコード	コメント
OS[Z]=1 FA[X]=1000 POS[X]=40	; 揺動の起動
OS[Z]=0	; 揺動の解除
M30	

### 13.3.1 反転点 1 または 2 の切り込み

#### 機能

揺動軸が反転点に達していない限り、切り込み軸は移動しません。

#### 用途

反転点の直接切り込み

#### プログラミング

##### 反転点 1

```
WHENEVER $AA_IM[Z] <> $SA_OSCILL_REVERSE_POS1[Z]
DO $AA_OVR[X] = 0 $AA_OVR[Z] = 100
```

##### 反転点 2

```
WHENEVER $AA_IM[Z] <> $SA_OSCILL_REVERSE_POS2[Z]
DO $AA_OVR[X] = 0 $AA_OVR[Z] = 100
```

##### システム変数の説明

- \$AA\_IM[ Z ]:MCS の揺動軸 Z の現在位置
- \$SA\_OSCILL\_REVERSE\_POS1[ Z ]:揺動軸の反転点 1 の位置
- \$AA\_OVR[ X ]:切り込み軸の軸オーバーライド
- \$AA\_OVR[ Z ]:揺動軸の軸オーバーライド

#### 切り込み

アブソリュート切り込み値は、POSP 命令によって定義されます(「切り込みの定義 POSP (ページ 871)」の章を参照してください)。

#### 割り当て

揺動軸と切り込み軸の間の割り当ては、OSCILL 命令によって定義されます(「揺動軸と切り込み軸の割り当て OSCILL (ページ 871)」の章を参照してください)。

### 13.3.2 反転点範囲の切り込み

#### 反転範囲 1

##### 機能

揺動軸が反転範囲(反転点 1 の位置+変数 ii1 の内容)に達していない場合、切り込みはおこなわれません。これは、反転点 1 が反転点 2 より低い値に設定されているという条件で適用されます。この条件に当てはまっていない場合、条件が合うように変更してください。

##### 用途

このシンクロナイズドアクションの目的は、揺動移動が反転範囲 1 に達するまで切り込み移動の開始を防止することです(「図 13-1 揺動軸の構成、切り込み軸 (ページ 862)」を参照してください)。

##### プログラミング

```
WHENEVER $AA_IM[Z] > $SA_OSCILL_REVERSE_POS1[Z] + ii1
DO $AA_OVR[X] = 0
```

システム変数の説明:

- \$AA\_IM[ Z ]:揺動軸 Z の現在位置
- \$SA\_OSCILL\_REVERSE\_POS1[ Z ]:揺動軸の反転点 1 の位置
- \$AA\_OVR[ X ]:切り込み軸の軸オーバーライド
- ii1:反転範囲の幅(ユーザー変数)

#### 反転範囲 2

##### 機能

揺動軸の現在位置(値)が、反転点 2 の位置と変数 ii2 の内容の差より小さくなるまで、切り込み軸は停止します。これは、反転点位置 2 の設定が、反転点位置 1 より高く設定されているという条件で適用されます。この条件に当てはまっていない場合、条件が合うように変更してください。

##### 用途

このシンクロナイズドアクションの目的は、揺動移動が反転範囲 2 に達するまで切り込み移動の開始を防止することです(「図 13-1 揺動軸の構成、切り込み軸 (ページ 862)」を参照してください)。

##### プログラミング

```
WHENEVER $AA_IM[Z] < $SA_OSCILL_REVERSE_POS2[Z] - ii2
```

```
DO $AA_OVR[X] = 0
```

説明:

- \$AA\_IM[ Z ]:揺動軸 Z の現在位置
- \$SA\_OSCILL\_REVERSE\_POS2[ Z ]:揺動軸の反転点 2 の位置
- \$AA\_OVR[ X ]:切り込み軸の軸オーバーライド
- ii2:反転範囲 2 の幅(ユーザー変数)

## 切り込み

アブソリュート切り込み値は、POSP 命令によって定義されます(「切り込みの定義 POSP (ページ 871)」の章を参照してください)。

## 割り当て

揺動軸と切り込み軸の間の割り当ては、OSCILL 命令によって定義されます(「揺動軸と切り込み軸の割り当て OSCILL (ページ 871)」の章を参照してください)。

## 下記も参照

シンクロナイズドアクションによる揺動制御 (ページ 861)

### 13.3.3 両方の反転点の切り込み

#### 手順の概要

これまでに説明した、反転点と反転範囲の切り込みの機能は、自由に組み合わせることができます。

#### 組み合わせ

##### 2面での切り込み

- 反転点 1 - 反転点 2
- 反転点 1 - 反転範囲 2
- 反転範囲 1 - 反転点 2
- 反転範囲 1 - 反転範囲 2

### 13.3 シンクロナイズドアクションによる揺動制御

#### 片側切り込み

- 反転点 1
- 反転点 2
- 反転範囲 1
- 反転範囲 2

(「反転点 1 または 2 の切り込み (ページ 865)」と「反転点範囲の切り込み (ページ 866)」の章を参照してください)。

### 13.3.4 反転点で揺動移動を停止

#### 機能

##### 反転点 1:

揺動軸は、反転点 1 に達するたびに、オーバーライドによって停止し、切り込み移動が始まります。

#### 用途

区間切り込みを実行するまで揺動軸を停止させるには、シンクロナイズドアクションを使用します。区間切り込みが実行されるまで揺動軸が反転点 1 で待機する必要がない場合、このシンクロナイズドアクションは省略できます。まだ動作中の以前のシンクロナイズドアクションによって停止している場合は、同時に、このシンクロナイズドアクションを使用して切り込み移動を開始できます。

#### プログラミング

```
WHENEVER $AA_IM[揺動軸] == $SA_OSCILL_REVERSE_POS1[揺動軸]
DO $AA_OVR[揺動軸] = 0    $AA_OVR[切り込み軸] = 100
```

##### システム変数の説明

\$AA\_IM[ 揺動軸 ]:揺動軸の現在位置

\$SA\_OSCILL\_REVERSE\_POS1[ 揺動軸 ]:揺動軸の反転点 1

\$AA\_OVR[ 揺動軸]:揺動軸の軸オーバーライド

\$AA\_OVR[切り込み軸]:切り込み軸の軸オーバーライド

## 機能

### 反転点 2:

揺動軸は、反転点 2 に達するたびに、オーバーライド 0 によって停止し、切り込み移動が始まります。

## 用途

区間切り込みを実行するまで揺動軸を停止させるには、シンクロナイズドアクションを使用します。区間切り込みが実行されるまで揺動軸が反転点 2 で待機する必要がない場合、このシンクロナイズドアクションは省略できます。まだ動作中の以前のシンクロナイズドアクションによって停止している場合は、同時に、このシンクロナイズドアクションを使用して切り込み移動を開始できます。

## プログラミング

```
WHENEVER $AA_IM[揺動軸] == $SA_OSCILL_REVERSE_POS2[揺動軸]  
DO $AA_OVR[揺動軸] = 0    $AA_OVR[切り込み軸] = 100
```

### システム変数の説明

- \$AA\_IM[ 揺動軸 ]:揺動軸の現在位置
- \$SA\_OSCILL\_REVERSE\_POS2[ 揺動軸 ]:揺動軸の反転点 2
- \$AA\_OVR[ 揺動軸 ]:揺動軸の軸オーバーライド
- \$AA\_OVR[ 切り込み軸 ]:切り込み軸の軸オーバーライド

## 13.3.5 揺動移動の再開

## 機能

切り込み軸の現在の移動軌跡区間の残移動距離が 0 になるたびに、つまり区間切り込みが実行されるたびに、揺動軸をオーバーライドによって開始します。

## 用途

このシンクロナイズドアクションの目的は、区間切り込み移動の完了時に揺動軸の移動を続けることです。揺動軸が区間切り込みの完了を待機する必要がない場合、反転点で揺動軸を停止するモーションシンクロナイズドアクションは省略してください。

## 13.3 シンクロナイズドアクションによる揺動制御

## プログラミング

```
WHENEVER $AA_DTEPW[切り込み軸] == 0
DO $AA_OVR[揺動軸] =100
```

## システム変数の説明

- \$AA\_DTEPW[ 切り込み軸 ]:切り込み軸の軸残移動距離(ワーク座標系 (WCS)):切り込み軸の軌跡距離
- \$AA\_OVR[ 揺動軸 ]:揺動軸の軸オーバーライド

## 13.3.6 区間切り込みの早期開始の防止

## 機能

これまでに説明した機能では、反転点外、または反転範囲外の切り込み移動は防止されます。しかし切り込みの完了時には、次の区間切り込み移動の再開を防止する必要があります。

## 用途

チャンネル別フラグをこのために使用します。このフラグは、区間切り込みの終了時(区間残移動距離が0のとき)に設定し、軸が反転範囲を離れたときに削除します。次の切り込み移動はシンクロナイズドアクションで防止します。

## プログラミング

```
WHENEVER $AA_DTEPW[切り込み軸]==0 DO $AC_MARKER[インデックス]=1
```

## 反転点 1

```
WHENEVER $AA_IM[Z] <> $SA_OSCILL_REVERSE_POS1[Z] DO $AC_MARKER[インデックス]=0
```

```
WHENEVER $AC_MARKER[インデックス]==1 DO $AA_OVR[切り込み軸]=0
```

## システム変数の説明

- \$AA\_DTEPW[ 切り込み軸 ]:切り込み軸の軸残移動距離(ワーク座標系 (WCS)):切り込み軸の軌跡距離
- \$AC\_MARKER[ インデックス ]:チャンネル別のインデックス付きマーカー
- \$AA\_IM[ 揺動軸 ]:揺動軸の現在位置
- \$SA\_OSCILL\_REVERSE\_POS1[ 揺動軸 ]:揺動軸の反転点 1
- \$AA\_OVR[ 切り込み軸 ]:切り込み軸の軸オーバーライド

### 13.3.7 揺動軸と切り込み軸の割り当て OSCILL

#### 機能

命令 OSCILL では、複数の切り込み軸を揺動軸に割り当てます。揺動動作が始まります。

PLC は、NC/PLC インタフェース経由で、どの軸が割り当てられたかを通知されます。PLC は、揺動軸を制御している場合、さらに、切り込み軸を監視し、切り込み軸の信号を使用して、インタフェースの 2 つのストップビットによって揺動軸の応答を生成します。

#### 用途

同期条件によって動作がすでに定義されている軸は、揺動モードの起動のためにそれぞれに割り当てられます。揺動移動が始まります。

#### プログラミング

OSCILL[揺動軸] = (切り込み軸 1、切り込み軸 2、切り込み軸 3)

括弧内の切り込み軸 2 と切り込み軸 3 およびその区切り文字は、不要の場合は省略できます。

### 13.3.8 切り込みの定義 POSP

#### 機能

制御装置は、切り込み軸の次のデータを受け取ります。

- 合計切り込み
- 反転点や反転点範囲の区間切り込み
- 終了時の区間切り込み動作

#### 用途

この命令は、OSCILL で揺動を起動した後で実行し、反転点、または反転点範囲の必要切り込み値をコントローラに通知します。

### 13.3 シンクロナイズドアクションによる揺動制御

#### プログラミング

POSP[切り込み軸] = (終了位置、区間、モード)

終了位置: すべての区間切り込み移動後の切り込み軸の終了位置

区間: 反転点や反転点範囲の区間切り込み

モード 0: 最後の 2 つの区間ステップの目標点までの残軌跡は、大きな等しい意 2 つの残りステップに分割されます(初期設定)。

モード 1: 計算したすべての区間長の合計が目標点までの軌跡に正確に対応するように、区間長が調整されます。

#### 13.3.9 外部揺動反転

たとえば PLC のキーを使用し、揺動領域を変更したり、揺動方向を直ちに反転したりすることができます。

エッジトリガの PLC 入力信号 DB31、... DBX28.0 (揺動反転)は、現在の揺動動作を減速した後、反対方向に移動するために使用されます。減速状態は、PLC 出力信号 DB31、... DBX100.2(揺動反転有効)によって通知されます。

軸の減速位置は、**新しい反転位置**として PLC 信号 DB31、... DBX28.4 (反転位置変更)によって受け付けられます。

PLC 入力信号 DB31、... DBX28.3 (反転点選択)は無視されますが、この変更は最後に開始された外部揺動反転命令に影響します。

ハンドルか JOG キーによって適用した反転点の変更は、当該軸で有効にならないことがあります。ハンドルか JOG キーの変更が現在有効である場合は、アラーム 20081(ハンドルが有効であるために減速位置を反転点として受け付けできない)が発生します。干渉が解消されると、アラームは自動的にリセットされます。

#### 停止時間

外部揺動反転のために、停止時間は方向転換に適用されません。軸は精密イグザクトストップ信号を待機します。設定されているすべてのイグザクトストップ条件が満たされます。

## 切り込み移動

ノンモーダル揺動の場合、外部揺動反転のために、方向転換のための切り込み移動はおこなわれません。これは反転点に到達しないために、当該のシンクロナイズドアクションが満たされないためです。

## システム変数

反転点 1 へのアプローチが中止された場合は、システム変数\$AA\_OSCILL\_BREAK\_POS1 によって、減速位置がスキャンできます。または

反転点 2 へのアプローチが中止された場合は、\$AA\_OSCILL\_BREAK\_POS2 によって、減速位置がスキャンできます。

当該の反転点に再びアプローチした場合、反転点の位置は\$AA\_OSCILL\_BREAK\_POS1 または\$AA\_OSCILL\_BREAK\_POS2 でスキャンできます。

言い換えると、外部揺動反転命令の後では、\$AA\_OSCILL\_BREAK\_POS1 と \$AA\_OSCILL\_REVERSE\_POS1 の値、および\$AA\_OSCILL\_BREAK\_POS2 と \$AA\_OSCILL\_REVERSE\_POS2 の値に違いがあります。

外部揺動反転は、シンクロナイズドアクションによって検出できます(「例 (ページ 874)」の章を参照してください)。

## 特別な事例

軸が開始位置にアプローチしているときに PLC 入力信号「揺動反転」が有効になると、アプローチ移動は中止されて軸は中断位置 1 にアプローチし続けます。

停止期間中に PLC 入力信号「揺動反転」を設定すると、停止タイマは無効になります。精密イグザクトストップにまだ達していない場合、軸は移動を続ける前に精密イグザクトストップ信号を待機します。

軸が終了位置にアプローチしているときに PLC 入力信号「揺動反転」を有効にすると、アプローチ移動は中止されて揺動は終了します。

外部揺動反転命令の例については、「外部揺動反転」を使用してシンクロナイズドアクションによって反転位置を変更 (ページ 884)」の章を参照してください。

## 13.4 制約条件

### 「揺動」機能の適用

この機能はオプションです(「揺動機能」)。ライセンス管理を通じてハードウェアに割り当ててください。

## 13.5 例

### 必要条件

下記の例では、次の章で指定する NC 言語のコンポーネントが必要となります。

- 非同期揺動
- および
- シンクロナイズドアクションによる揺動制御

### 13.5.1 非同期揺動例

#### 処理

揺動軸 Z は-10 から 10 の間で揺動します。反転点 1 には汎用イグザクトストップで、反転点 2 にはイグザクトストップなしでアプローチします。揺動軸の送り速度は 5000 です。加工運転の終了時には 3 つのスパークアウトストロークを実行し、揺動軸は終了位置 30 にアプローチします。切り込み軸の送り速度は 1000 で、X 方向の切り込みの終了は位置 15 です。

#### プログラム区間

プログラムコード	コメント
OSP1[Z]=-10	; 反転点 1
OSP2[Z]=10	; 反転点 2
OST1[Z]=-1	; 反転点 1 の停止時間汎用イグザクトストップ
OST2[Z]=-2	; 反転点 2 の停止時間 イグザクトストップなし
FA[Z]=5000	; 揺動軸の送り速度

プログラムコード	コメント
OSNSC[Z]=3	; 3つのスパークアウトストローク
OSE[Z]=-3	; 終了位置
OS1 F500 X15	; 揺動開始、X軸切り込み
	; 切り込み速度 500、送り目標 15

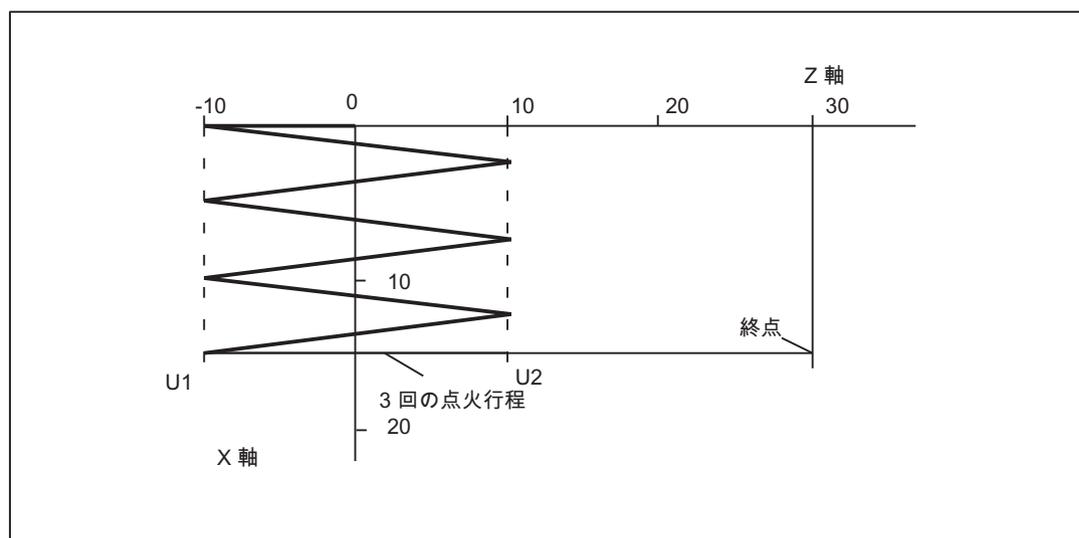


図 13-2 揺動移動と切り込みのシーケンス例 1

## 13.5.2 シンクロナイズドアクションによる揺動の例 1

### 処理

直接切り込みは反転点 1 で実行します。揺動軸は区間切り込みの実行を待機後に移動を継続します。反転点 2 では、反転点 2 の-6 の距離で切り込みを実行します。揺動軸は、区間切り込みが実行されるまで、この反転点で待機しません。Z 軸が揺動軸で、X 軸が切り込み軸です(「シンクロナイズドアクションによる揺動制御 (ページ 861)」の章を参照してください)

### 注記

セッティングデータ OSCILL\_REVERSE\_POS\_1/2 は機械座標系の値です。このため、\$AA\_IM[n]のみで比較を実行できます。

13.5 例

プログラム区間

プログラムコード	コメント
; 例 1: シンクロナイズドアクションによる揺動	
OSP1[Z]=10 OSP2[Z]=60	; 反転点 1 と反転点 2 を定義
OST1[Z]=-2 OST2[Z]=0	; 反転点 1: イグザクトストップなし ; 反転点 2: 精密イグザクトストップ
FA[Z]=5000 FA[X]=250	; 揺動軸の速度、送り速度、切り込み軸速度
OSCTRL[Z]=(1+8+16,0)	; DTG スパークアウトの解除および終了位置への移動後に ; 反転点 1 で揺動動作のオフ ; DTG の後、当該の反転位置にアプローチ ;
OSNSC[Z]=3	; 3 つのスパークアウトストローク
OSE[Z]=0	; 終了位置 = 0
WAITP(Z)	; Z 軸の揺動を有効化
;	
; モーションシンクロナイズドアクション:	
;	
; このときはいつも	機械座標系 (Machine Coordinate System) の揺動軸の
	現在位置が
; 反転位置 1 と	違うとき
; その後	インデックス 1 のマーカーを値 0 に設定
	(マーカー 1 をリセット)
WHENEVER \$AA_IM[Z]<>\$SA_OSCILL_REVERSE_POS1[Z] DO \$AC_MARKER[1]=0	
;	
; このときはいつも	機械座標系 (Machine Coordinate System) の揺動軸の
	現在位置が
; 反転範囲 2 の開始より	少ないとき
	(ここでは: 反転点 2 -6)
; その後	切り込み軸の軸オーバーライドを 0% に設定
; および	インデックス 2 のマーカーを値 0 に設定
	(マーカー 2 をリセット)
WHENEVER \$AA_IM[Z]<\$SA_OSCILL_REVERSE_POS2[Z]-6 DO \$AA_OVR[X]=0 \$AC_MARKER[2]=0	
;	
; このときはいつも	機械座標系 (Machine Coordinate System) の揺動軸の
	現在位置が
; 反転点 1 に	等しいとき
; その後	揺動軸の軸オーバーライドを 0% に設定
;	
; および	切り込み軸の軸オーバーライドを
	100% に設定 (これによって、以前のシンクロナイズドアクション
	がキャンセルされる!)
WHENEVER \$AA_IM[Z]==\$SA_OSCILL_REVERSE_POS1[Z] DO \$AA_OVR[Z]=0 \$AA_OVR[X]=100	
;	
; このときはいつも	区間切り込みの残移動距離

プログラムコード	コメント
; が	0,
; のとき、その後	インデックス 2 のマーカーを値 1 に設定
; および	インデックス 1 のマーカーを値 1 に設定
WHENEVER \$AA_DTEPW[X]==0 DO \$AC_MARKER[2]=1 \$AC_MARKER[1]=1	
;	
; このときはいつも	インデックス 2 のフラグ
; が	1,
; のとき、その後	切り込み軸の軸オーバーライドを
;	0%に設定、これによって早期の切り込みを防止
;	(揺動軸は
;	反転点 1 を離れていない)。
;	
WHENEVER \$AC_MARKER[2]==1 DO \$AA_OVR[X]=0	
;	
; このときはいつも	インデックス 1 のフラグ
; が	1,
; のとき、その後	切り込み軸の軸オーバーライドを
;	0%に設定、これによって早期の切り込みを防止
;	(揺動軸は
;	反転点範囲 2 を離れていない)。
; および	切り込み軸の軸オーバーライドを
;	100%に設定(揺動の「起動」)
WHENEVER \$AC_MARKER[1]==1 DO \$AA_OVR[X]=0 \$AA_OVR[Z]=100	
;	
;MCS の揺動軸の現在位置が	
;反転点 1 に	等しいとき
; その後	切り込み軸の軸オーバーライドを
;	100%
; および	切り込み軸の軸オーバーライドを
;	0%に設定(これによって、第 2 のシンクロナイズドアクション
;	が一度キャンセルされます!)
WHEN \$AA_IM[Z]==\$SA_OSCILL_REVERSE_POS1[Z] DO \$AA_OVR[Z]=100 \$AA_OVR[X]=0	
;	
;	-----
OSCILL[Z]=(X)	; 揺動軸 Z に切り込み軸として軸 X を割り当てます。
POSP[X]=(5,1,1)	
	; これにより、1 からの手順で、終了位置 5 への
	; 切り込みがおこなわれ、すべての部分の長さの
	; 合計が終了位置を正確に
	; 示すことが必要です。
M30	; プログラムの終了

13.5 例

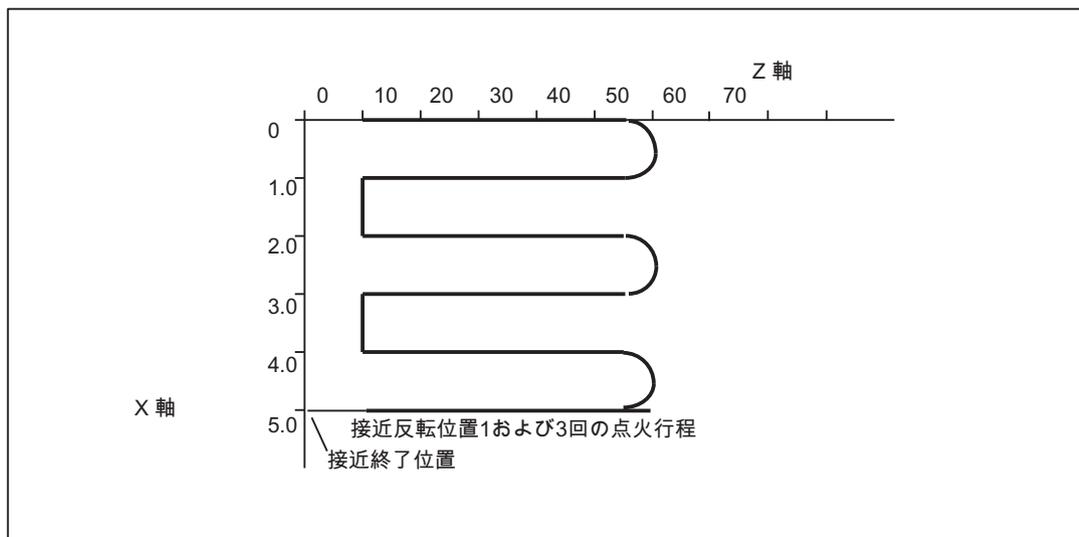


図 13-3 揺動移動と切り込みのシーケンス例 1

13.5.3 シンクロナイズドアクションによる揺動の例 2

処理

反転点 1 で切り込みを実行しません。反転点 2 では、反転点 2 から距離 ii2 で切り込みをおこないます。この反転点で、揺動軸は区間切り込みが実行されるまで待機します。Z 軸が揺動軸で、X 軸が切り込み軸です

プログラム区間

例 2:シンクロナイズドアクションによる揺動

プログラムコード	コメント
DEF INT ii2	; 反転領域 2 の変数を定義
;	
OSP1[Z]=10 OSP2[Z]=60	; 反転点 1 と反転点 2 を定義
OST1[Z]=0 OST2[Z]=0	; 反転点 1:精密イグザクトストップ
;	反転点 2:精密イグザクトストップ
FA[Z]=5000 FA[X]=100	; 揺動軸の送り、切り込み軸の送り
OSCTRL[Z]=(2+8+16,1)	; 残移動距離削除、スパークアウト、および終了位置後に
;	反転点 2 の揺動動作オフ
;	残移動距離削除の後に適切にアプローチ
;	反転位置へアプローチ
OSNSC[Z]=3	; 3 つのスパークアウトストローク
OSE[Z]=70	; 終了位置 = 70

プログラムコード	コメント
ii2=2	; 反転点範囲を設定する
WAITP (Z)	; Z 軸の揺動を許可
;	
;	; モーションシンクロナイズドアクション:
;	; このときはいつも 機械座標系 (Machine Coordinate System) の揺動軸の
;	; 現在位置が
;	; 反転範囲 2 の開始より 少ないとき
;	; その後 送り軸の軸オーバーライドを
;	; 0%に
;	; および インデックス 0 のマーカーを値 0 に設定
WHENEVER \$AA_IM[Z]<\$SA_OSCILL_REVERSE_POS2[Z]-ii2 DO \$AA_OVR[X]=0 \$AC_MARKER[0]=0	
;	
;	; このときはいつも 機械座標系 (Machine Coordinate System) の揺動軸の
;	; 現在位置が
;	; 反転位置 2 より 大きい、または等しいとき
;	; その後 揺動軸の軸オーバーライドを
;	; 0%に
WHENEVER \$AA_IM[Z]>=\$SA_OSCILL_REVERSE_POS2[Z] DO \$AA_OVR[Z]=0	
;	
;	; このときはいつも 区間切り込みの残移動距離
;	; が 0,
;	; のとき、その後 インデックス 0 のマーカーを値 1 に設定
WHENEVER \$AA_DTEPW[X] == 0 DO \$AC_MARKER[0]=1	
;	
;	; このときはいつも インデックス 0 のフラグ
;	; が 1,
;	; のとき、その後 新しい早期の切り込みを防止するために、
;	; 切り込み軸の軸オーバーライドを 0%に設定
;	; (揺動軸は反転点範囲 2 をまだ離れていないが、
;	; 切り込み軸では新しい切り込みの準備が
;	; 完了している)
;	; および 揺動軸の軸オーバーライドを 100%に設定
;	; (これは、第 2 のシンクロナイズドアクションをキャンセルする)
;	
WHENEVER \$AC_MARKER[0]==1 DO \$AA_OVR[X]=0 \$AA_OVR[Z]=100	
;	
OSCILL[Z]=(X)	; 軸を起動する
POSP[X]=(5,1,1)	
;	; 揺動軸 Z を切り込み軸として X 軸に
;	; 割り当てます
;	; 軸 X は、ステップ値 1 で終了位置 5 に
;	; 移動する必要があります
;	
M30	

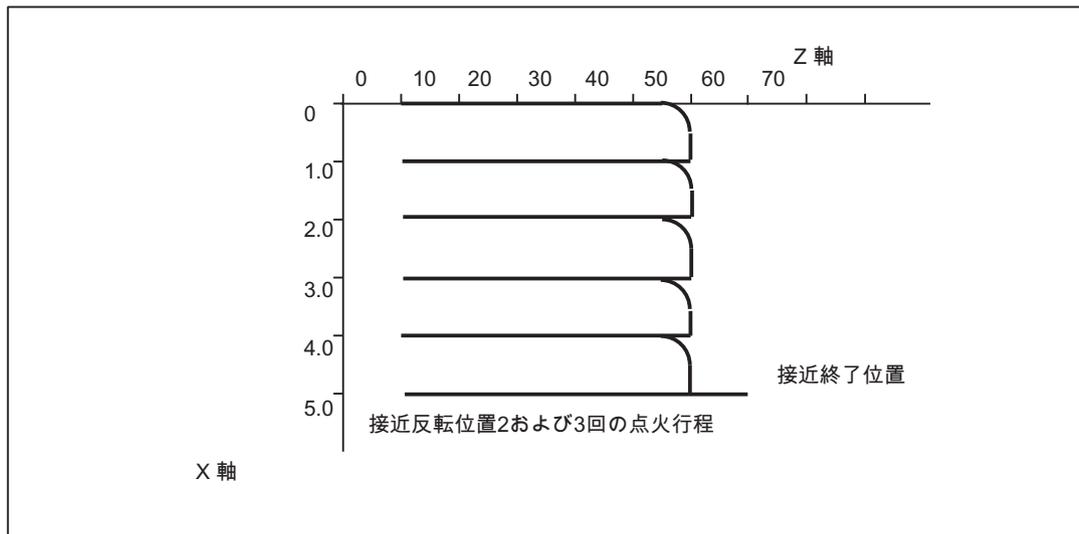


図 13-4 揺動移動と切り込みのシーケンス例 2

### 13.5.4 開始位置の例

#### 13.5.4.1 言語命令による開始位置の定義

プログラムコード	コメント
WAITP(Z)	; Z 軸の揺動を有効化
OSP1[Z]=10 OSP2[Z]=60	; 反転点 1 と反転点 2 を定義
OST1[Z]=-2 OST2[Z]=0	; 反転点 1: イグザクトストップなし ; 反転点 2: 精密イグザクトストップ
FA[Z]=5000 FA[X]=2000	; 揺動軸の送り速度、 ; 切り込み軸の送り速度
OSCTRL[Z]=(1+8+16,0)	; DTG スパークアウトの解除および終了位置への移動後に ; 反転点 1 で揺動動作のオフ ; DTG の後、当該の反転位置にアプローチ ;
OSNSC[Z]=3	; 3 つのスパークアウトストローク
OSE[Z]=0	; 終了位置 = 0
OSB[Z]=0	; 開始位置 = 0
OS[Z]=1 X15 F500	; 揺動開始、連続切り込み
OS[Z]=0	; 揺動の解除
WAITP(Z)	; 揺動動作の完了を待機
M30	

## 説明

Z 軸は、揺動を始めるとき、開始位置(この例の位置 = 0)にアプローチしてから、反転点 10 と 60 の間で揺動動作を始めます。X 軸が終了位置 15 に達すると、3 つのスパークアウトストロークで揺動は終了し、終了位置 0 にアプローチします。

## 13.5.4.2 セッティングデータによる揺動の開始

プログラムコード	コメント
WAITP (Z)	
STOPRE	
\$SA_OSCILL_REVERSE_POS1[Z]=-10	; 反転位置 1 = -10
\$SA_OSCILL_REVERSE_POS2[Z]=30	; 反転位置 2 = 30
\$SA_OSCILL_START_POS[Z] = -50	; 開始位置 = -50
\$SA_OSCILL_CTRL_MASK[Z] = 512	; 開始位置へアプローチ、 ; スイッチオフの場合は次の ; 反転点で停止 ; 終了位置へアプローチしない ; DTG のスパークアウトストロークなし
\$SA_OSCILL_VELO[ Z ] = 5000	; 揺動軸の送り速度
\$SA_OSCILL_IS_ACTIVE[ Z ] = 1	; 起動
\$SA_OSCILL_DWELL_TIME1[ Z ] = -2	; イグザクトストップを待機しない
\$SA_OSCILL_DWELL_TIME2[ Z ] = 0	; 精密イグザクトストップを待機
STOPRE	
X30 F100	
\$SA_OSCILL_IS_ACTIVE[ Z ] = 0	; ストップ
WAITP (Z)	
M30	

## 説明

Z 軸は、揺動を始めるとき、開始位置(この例の位置 = -50)にアプローチしてから、反転点 -10 と 30 の間で揺動動作を始めます。X 軸が終了位置 30 に達すると、揺動は次にアプローチした反転点で終了します。

## 13.5.4.3 ノンモーダル揺動(開始位置 = 反転点 1)

## シンクロナイズドアクションによる揺動

プログラムコード	コメント
N701	; シンクロナイズドアクションによる揺動、

## 13.5 例

プログラムコード	コメント
	; 開始位置 == 反転点 1
;	
N702 OSP1[Z]=10 OSP2[Z]=60	; 反転点 1 と反転点 2 を定義
N703 OST1[Z]=0 OST2[Z]=0	; 反転点 1:汎用イグザクトストップ
	; 反転点 2:精密イグザクトストップ
N704 FA[Z]=5000 FA[X]=2000	; 揺動軸の送り速度、
	; 切り込み軸の送り速度
N705 OSCCTRL[Z]=(1+8+16.0)	; DTG スパークアウトおよび終了位置への移動の後、
	; 反転点 1 で
	; 揺動動作をオフ
	; DTG の後、
	; 当該の反転位置にアプローチ
	;
N706 OSNSC[Z]=3	; 3つのスパークアウトストローク
N707 OSE[Z]=0	; 終了位置 = 0
N708 OSB[Z]=10	; 開始位置 = 10
N709 WAITP(Z)	; Z軸の揺動を有効化
;	
; モーションシンクロナイズドアクション:	
; インデックス 2 のマーカーを 1 で設定 (初期化)	
WHEN TRUE DO \$AC_MARKER[2]=1	
;	
; このときはいつも	インデックス 2 のマーカーが 0 に等しいとき
;	また、揺動軸の現在位置が
;	反転位置 1 に一致しないとき
; その後	インデックス 1 のマーカーを 0 に設定
;	
WHENEVER (\$AC_MARKER[2] == 0) AND \$AA_IW[Z]>\$SA_OSCILL_REVERSE_POS1[Z]	
DO \$AC_MARKER[1]=0	
; このときはいつも	揺動軸の現在位置が
;	反転範囲 2 の開始位置未満であるとき
;	
; その後	送り軸の軸オーバーライドを
;	0 に設定し、インデックス 0 のマーカーを下記に設定
;	0
WHENEVER \$AA_IW[Z]<\$SA_OSCILL_REVERSE_POS2[Z]-6 DO \$AA_OVR[X]=0 \$AC_MARKER[0]=0	
;	
; このときはいつも	揺動軸の現在位置が
;	反転位置 1 に等しいとき
; その後	揺動軸の軸オーバーライドを 0 に設定し、
;	切り込み軸の軸オーバーライドを
;	100%に設定する (それによって、
;	以前のシンクロナイズドアクションがキャンセルされる!)
WHENEVER \$AA_IW[Z]==\$SA_OSCILL_REVERSE_POS1[Z] DO \$AA_OVR[Z]=0 \$AA_OVR[X]=100	

プログラムコード	コメント
;	
; このときはいつも	区間切り込みの残移動距離が 0 に等しいとき
;	
; その後	インデックス 0 のマーカを 1 に設定、そして
;	インデックス 1 のマーカを 1 に設定、そして
WHENEVER \$AA_DTEPW[X]==0 DO \$AC_MARKER[0]=1 \$AC_MARKER[1]=1	
;	
; このときはいつも	インデックス 0 のマーカが 1 に等しいとき
; その後	新たな早期切り込みを防止するために
;	送り軸の軸オーバーライドを 0
;	に設定する!
WHENEVER \$AC_MARKER[0]==1 DO \$AA_OVR[X]=0	
;	
; このときはいつも	インデックス 1 のマーカが 1 に等しいとき
; その後	(新たな早期切り込みを防止するために!)
;	送り軸の軸オーバーライドを 0
;	に設定し、
;	(以前のシンクロナイズドアクションをキャンセルするために!)
;	揺動軸の軸オーバーライドを
;	100%に設定する
WHENEVER \$AC_MARKER[1]==1 DO \$AA_OVR[X]=0 \$AA_OVR[Z]=100	
;	
; このとき;	揺動軸の現在位置が
;	反転位置 1 に等しいとき
; その後	インデックス 2 のマーカをリセットし、
;	1 番目のシンクロナイズドアクションを
;	解除する (開始位置 == 反転位置 1 に達したときに
;	切り込みなし)
WHEN \$AA_IW[Z]==\$SA_OSCILL_REVERSE_POS1[Z] DO \$AC_MARKER[2]=0	
;	
;	-----
N750 OSCILL[Z]=(X) POSP[X]=(5,1,1)	
;	切り込み軸として揺動軸 Z に軸 X を割り当てる。
;	これは、1 のサブステップで終了位置 5 まで切り込みをおこない、
;	すべてのサブステップ長さの合計が正確に終了位置に
;	一致する必要があります。
;	
N780 WAITP(Z)	; Z 軸を解放
;	
N790 X0 Z0	
N799 M30	; プログラムの終了

13.6 データリスト

説明

開始位置は反転点 1 に一致しています。シンクロナイズドアクション WHEN ... (上記参照してください)は、開始位置に達したときの切り込みを防止します。

13.5.5 外部揺動反転例

13.5.5.1 「外部揺動反転」を使用してシンクロナイズドアクションによって反転位置を変更

プログラムコード	コメント
DEFINE BREAKPZ AS \$AA_OSCILL_BREAK_POS1[Z]	
DEFINE REVPZ AS \$SA_OSCILL_REVERSE_POS1[Z]	
WAITP(Z)	; Z 軸の揺動を有効化
OSP1[Z]=10 OSP2[Z]=60	; 反転点 1 と反転点 2 を定義
OSE[Z]=0	; 終了位置 = 0
OSB[Z]=0	; 開始位置 = 0
	; 揺動反転点 1 の揺動の外部反転では、
	; 以下を設定する
WHENEVER BREAKPZ <> REVPZ DO \$\$SA_OSCILL_REVERSE_POS1 = BREAKPZ	
OS[Z]=1 X150 F500	; 揺動開始、連続切り込み
OS[Z]=0	; 揺動の解除
WAITP(Z)	; 揺動動作の完了を待機
M30	

13.6 データリスト

13.6.1 マシンデータ

13.6.1.1 一般マシンデータ

番号	識別子: \$MN_	説明
10710	PROG_SD_RESET_SAVE_TAB	指定された SD 番号の関連揺動データを保存
11460	OSCILL_MODE_MASK	非同期揺動の制御ビットパターン

## 13.6.2 セッティングデータ

### 13.6.2.1 軸/主軸別セッティングデータ

番号	識別子: \$SA_	説明
43700	OSCILL_REVERSE_POS1	反転点 1 の位置
43710	OSCILL_REVERSE_POS2	反転点 2 の位置
43720	OSCILL_DWELL_TIME1	反転点 1 の停止時間
43730	OSCILL_DWELL_TIME2	反転点 2 の停止時間
43740	OSCILL_VELO	揺動軸の送り速度
43750	OSCILL_NUM_SPARK_CYCLES	退避ストローク回数
43760	OSCILL_END_POS	退避ストローク後または揺動移動終了時の位置
43770	OSCILL_CTRL_MASK	揺動の制御ビットパターン
43780	OSCILL_IS_ACTIVE	揺動移動のオンとオフの切り替え
43790	OSCILL_START_POS	SD43770 で有効にした場合、揺動後で反転点 1 到達前のアプローチ位置:

## 13.6.3 信号

### 13.6.3.1 軸/主軸への信号

信号名称	SINUMERIK 840D sl	SINUMERIK 828D
外部揺動反転	DB31, ... .DBX28.0	DB380x.DBX5004.0
反転点の設定	DB31, ... .DBX28.3	DB380x.DBX5004.3
反転点の変更	DB31, ... .DBX28.4	DB380x.DBX5004.4
次の反転点で停止	DB31, ... .DBX28.5	DB380x.DBX5004.5
減速カーブに沿って停止	DB31, ... .DBX28.6	DB380x.DBX5004.6
PLC 制御軸	DB31, ... .DBX28.7	DB380x.DBX5004.7

## 13.6 データリスト

## 13.6.3.2 軸/主軸からの信号

信号名称	SINUMERIK 840D sl	SINUMERIK 828D
揺動反転が有効	DB31, ... .DBX100.2	DB390x.DBX5004.2
揺動開始不可	DB31, ... .DBX100.3	DB390x.DBX5004.3
揺動移動中のエラー	DB31, ... .DBX100.4	DB390x.DBX5004.4
退避が有効	DB31, ... .DBX100.5	DB390x.DBX5004.5
揺動移動が有効	DB31, ... .DBX100.6	DB390x.DBX5004.6
揺動が有効	DB31, ... .DBX100.7	DB390x.DBX5004.7

## 13.6.4 システム変数

## 13.6.4.1 モーションシンクロナイズドアクションの実行プログラム変数

## 実行プログラム変数読み出し(variable\_read)

実行プログラム変数読み出し(variable\_read)には次の変数があります。

\$A_IN[<演算式>]	デジタル入力(論理型)
\$A_OUT[<演算式>]	デジタル出力(論理型)
\$A_INA[<演算式>]	アナログ入力(論理型)
\$A_OUTA[<演算式>]	アナログ出力(論理型)
\$A_INCO[<演算式>]	コンパレータ入力(論理型)
\$AA_IW[<軸の形式>]	PCS 軸の実位置(実数)
\$AA_IB[<軸の形式>]	BCS 軸の実位置(実数)
\$AA_IM[<軸の形式>]	MCS 軸の実位置(IPO 指令値) (実数) \$AA_IM[S1]では主軸の指令値を参照できます。マシンデータ\$MA_ROT_IS_MODULOと\$MA_DISPLAY_IS_MODULOに応じて、モジュロ演算が主軸と回転軸に使用されません。
\$AA_OSCILL_BREAK_POS1	反転点 1 へアプローチ時の外部揺動反転後の減速位置

\$AA_OSCILL_BREAK_POS2	反転点 2 へアプローチ時の外部揺動反転後の減速位置
\$AC_TIME	ブロック開始からの秒単位の時間(実数) (内部で生成された中間ブロックの時間を含む)
\$AC_TIMES	ブロックの開始からの秒単位の時間(実数) (内部で生成された中間ブロックの時間を含まない)
\$AC_TIMEC	ブロックの開始からの IPO ステップ単位の時間(実数) (内部で生成された中間ブロックのステップを含む)
\$AC_TIMESC	ブロックの開始からの IPO ステップ単位の時間(実数) (内部で生成された中間ブロックのステップを含まない)
\$AC_DTBB	BCS のブロック先頭からの距離 (先頭までの距離、基本座標系) (実数)
\$AC_DTBW	PCS のブロック先頭からの距離 (先頭までの距離、ワーク座標系) (実数)
\$AA_DTBB[<軸の形式>]	BCS のブロック先頭からの軸の距離 (先頭までの距離、基本座標系) (実数)
\$AA_DTBW[<軸の形式>]	PCS のブロック先頭からの軸の距離 (先頭までの距離、ワーク座標系) (実数)
\$AC_DTEB	BCS のブロック終了までの距離(終了までの距離) (終了までの距離、基本座標系) (実数)
\$AC_DTEW	PCS のブロック終了までの距離 (終了までの距離、ワーク座標系) (実数)
\$AA_DTEB[<軸の形式>]	BCS の移動終了までの軸の距離 (終了までの距離、基本座標系) (実数)
\$AA_DTEW[<軸の形式>]	PCS の移動終了までの軸の距離 (終了までの距離、ワーク座標系) (実数)
\$AC_PLTBB	BCS のブロック先頭からの距離 (先頭からの軌跡長、基本座標系) (実数)
\$AC_PLTEB	BCS のブロック終了までの距離(終了までの距離) (終了までの軌跡長、基本座標系) (実数)

## 13.6 データリスト

\$AC_VACTB	BCS の軌跡速度 (実速度、基本座標系) (実数)
\$AC_VACTW	PCS の軌跡速度 (実速度、ワーク座標系) (実数)
\$AA_VACTB[<軸の形式>]	BCS の軸速度 (実速度、基本座標系) (実数)
\$AA_VACTW[<軸の形式>]	PCS の軸速度 (実速度、ワーク座標系) (実数)
\$AA_DTEPB[<軸の形式>]	BCS の揺動切り込み軸の残移動距離 (終点までの距離、揺動値、基本座標系) (実数)
\$AA_DTEPW[<軸の形式>]	PCS での揺動切り込みの軸の残移動距離 (終点までの距離、揺動値、ワーク座標系) (実数)
\$AC_DTEPB	BCS での揺動切り込みの軌跡残移動距離 (P2 ではない) (終点までの距離、揺動値、基本座標系) (実数)
\$AC_DTEPW	PCS の揺動切り込みの軌跡残移動距離 (P2 ではない) (終点までの距離、揺動値、ワーク座標系) (実数)
\$AC_PATHN	(正規化軌跡パラメータ) (実数) 正規化軌跡パラメータ: ブロック先頭の 0 からブロック終了の 1 まで
\$AA_LOAD[<軸の形式>]	ドライブ負荷率
\$AA_POWER[<軸の形式>]	ドライブ効率(W)
\$AA_TORQUE[<軸の形式>]	ドライブトルク指令値[Nm]
\$AA_CURR[<軸の形式>]	軸の電流フィードバック値

<p><b>\$AC_MARKER[&lt;演算式&gt;]</b> (整数)</p>	<p>フラグ変数: シンクロナイズドアクションで複雑な条件を構築するために使用できます。<b>8</b> 個のマーカ (インデックス <b>0 - 7</b>) を使用できます。</p> <p>リセットではマーカが <b>0</b> に設定されます。</p> <p>例:</p> <pre>WHEN .....DO \$AC_MARKER[0]=2   WHEN .....DO \$AC_MARKER[0]=3   WHEN \$AC_MARKER[0]==3 DO     \$AC_OVR=50</pre> <p>パートプログラムでシンクロナイズドアクションとは無関係に読み書きができます。</p> <pre>IF \$AC_MARKER == 4 GOTOF SPRUNG</pre>
<p><b>\$AC_PARAM[&lt;演算式&gt;]</b> (実数)</p>	<p>シンクロナイズドアクションの浮動小数点パラメータ シンクロナイズドアクションの中間保存および評価として動作します。</p> <p><b>50</b> 個のパラメータ (インデックス <b>0 - 49</b>) を使用できます。</p>
<p><b>\$AA_OSCILL_REVERSE_POS1</b> [&lt;軸の形式&gt;] (実数)</p>	
<p><b>\$AA_OSCILL_REVERSE_POS2</b> [&lt;軸の形式&gt;] (実数)</p>	<p>現在の揺動反転点 <b>1</b> と反転点 <b>2</b>:</p> <p><b>\$SA_OSCILL_REVERSE_POS1</b> または <b>\$SA_OSCILL_REVERSE_POS2</b> から現在のセッティングデータが読み取られます。</p> <p>動作中のシンクロナイズドアクション中などで揺動動作中に、反転位置でセッティングデータを変更できるようになります。</p>

## 条件

モーションシンクロナイズドアクションの条件は定式化されています。

実行プログラム変数の比較演算式

詳細については、以下を参照してください。

**参照先:**

『機能マニュアル、シンクロナイズドアクション』

### 13.6 データリスト

## R2 : 回転軸

### 14.1 簡単な説明

下記は、位置決め軸の一般的な用途です。

- 5 軸加工(動作範囲制限付き、または動作範囲制限なし)
- 偏心加工用回転軸(動作範囲制限なし)
- 円筒または成形研削の回転軸(動作範囲制限なし)
- TRANSMIT を使用した C 軸(動作範囲制限なし)
- 巻線機械の回転軸(動作範囲制限なし)
- ホブ盤の回転ワーク軸(C) (動作範囲制限なし)
- 円筒工具マガジンと工具タレット(動作範囲制限なし)
- 円筒補間用回転軸(動作範囲制限付き)
- グリッパ用旋回軸(動作範囲 360°)
- 旋回用回転軸(動作範囲 < 360°、60°など)
- ホブ盤のフライス旋回軸(A) (動作範囲、90°など)

### 軸アドレス

NC 制御工作機械の直線軸と移動方向は、以下の通り DIN 66025 に従って指定します。

- 直線軸:X、Y、Z
- 回転軸:A、B、C

14.1 簡単な説明

- 割り当て: X を中心とした A 回転、Y を中心とした B 回転、Z を中心とした C 回転。
- 回転の方向: 回転の正の回転軸方向は、対応する直線軸の正の軸方向を見て時計回り。

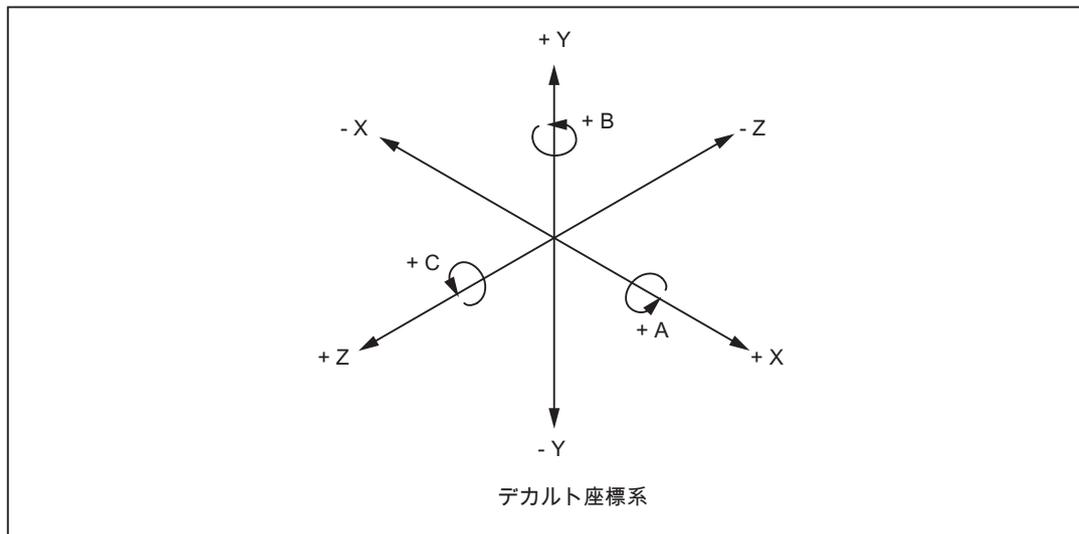


図 14-1 回転軸の軸識別子と回転方向

単位系

標準では、次の単位系が回転軸の入出力に適用されます。

回転軸の単位系	
物理量	単位
回転位置	度
プログラム指令角速度	度/min
回転速度	1) rev/min
回転加減速度	1) rev/s <sup>2</sup>
回転加々速度制限	1) rev/sec <sup>3</sup>
1) 回転軸の軸マシンデータの単位。	

参照先:

『機能マニュアル、基本機能』；「速度、指令値/フィードバック回路、閉ループ制御(G2)」

送り速度

回転軸の場合、プログラム指令送り速度 F は角速度(度/min)に対応します。

G94 または G95 で回転軸と直線軸が共通軌跡を移動する場合、送り速度は直線軸の単位系(mm/min、inch/min)で解釈されます。

回転軸の接線速度は、直径  $D_{\text{unit}}$  (単位直径  $D_{\text{unit}}=360/\pi$ )になります。単位直径  $D=D_{\text{unit}}$  の場合、度/min のプログラム指令角速度と mm/min 単位の接線速度は数値的に同じです。

一般的に、接線速度には下記が適用されます。

$$F = F_{\text{angle}} * D/D_{\text{unit}}$$

$F$  = 接線速度[mm/min]  
 $F_{\text{angle}}$  = 角速度[度/min]  
 $D$  = F に作用する直径[mm]  
 $D_{\text{unit}} = 360/\pi$  では  
 $DE$  = 直径単位[mm]  
 $\pi$  = 円弧定数 Pi

### JOG モード:毎回転送り速度

JOG モードでは、回転軸の動作も次のセッティングデータによって決まります。

SD41100 \$SN\_JOG\_REV\_IS\_ACTIVE (JOG が有効なときの毎回転送り速度)

## 14.2 モジュール 360°

360°モジュール回転軸は、位置が 0°～ 359.999°の範囲に割り当てられている回転軸です。したがって、360°モジュール回転軸の移動範囲は、原則として無限です。360°モジュール回転軸に対する移動範囲リミット(ソフトウェアリミットスイッチおよびワーキングエリアリミット)は無効です。

次の図では、回転の正方向への回転軸のアブソリュート位置を螺旋で表しています。螺旋を指すポインタは、回転軸のアブソリュート位置(420°)を示しています。ポインタがモジュール円と交差する点は、対応する 360°モジュール位置(60°)を示します。

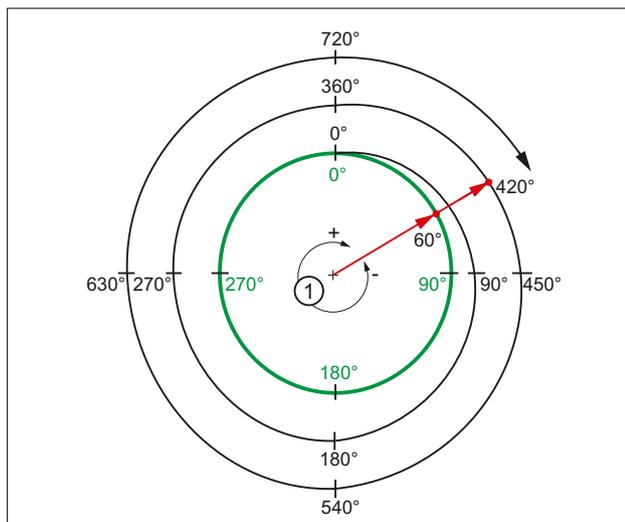


図 14-2 モジュロ割り当て

## パラメータの割り付け

### モジュロ機能

回転軸のモジュロ機能は、次の軸別マシンデータを使用して有効にします。

```
MD30310 $MA_ROT_IS_MODULO = 1
```

### 注記

360°モジュロ回転軸の場合は、位置表示もユーザーインターフェースで 360°に設定することを推奨します。

### Modulo range

360°モジュロ回転軸に対しては**必ず**、次の軸別マシンデータを使用してモジュロ範囲を 360°に設定してください。

```
MD30330 $MA_MODULO_RANGE = 360.0
```

### 360°モジュロ範囲の開始位置

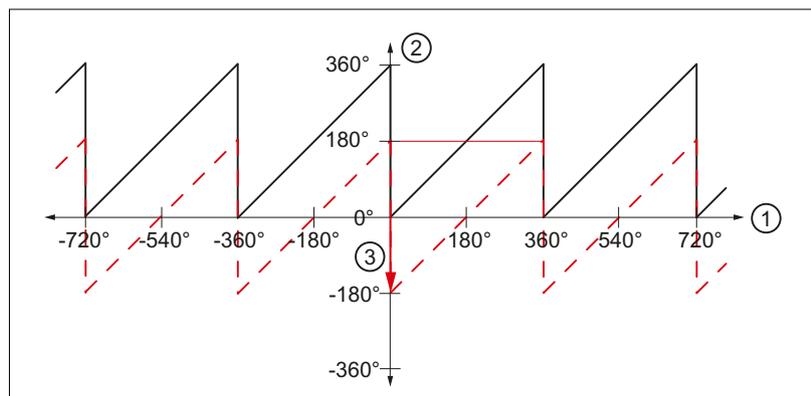
360°モジュロ範囲の開始位置は、次の軸別マシンデータを使用して設定できます。

```
MD30340 $MA_MODULO_RANGE_START = <開始位置>
```

したがって、パラメータ設定した開始位置は、垂直軸のモジュロ機能のオフセットを表します。モジュロ範囲は 360°のままです。

例

- 標準開始位置:  $0^\circ$   $\Rightarrow$  モジュロ範囲:  $0^\circ \sim 360^\circ$
- シフトした開始位置:  $-180^\circ$   $\Rightarrow$  モジュロ範囲:  $-180^\circ \sim +180^\circ$



- ① アブソリュート位置
- ② モジュロ位置
- ③ シフトした開始位置( $-180^\circ$ )

図 14-3 標準開始位置およびシフトした開始位置(赤)

## 注記

### モジュロパーティション軸

次の2つのマシンデータを調整して、360°モジュロ割り出し軸の割り出し位置を、360°モジュロ範囲と同じ方法で実行できます。

- MD30503 \$MA\_INDEX\_AX\_OFFSET
- MD30340 \$MA\_MODULO\_RANGE\_START

「割り出し軸」の章の「例3:等間隔の割り出し軸としての回転モジュロ軸(ページ1010)」および以降の各章の例を参照してください。

### 360°モジュロ回転軸:G90の位置決め特性(モーダルアブソリュート指令)

G90の標準特性は、次のマシンデータを使用して設定します。

MD30455 \$MA\_MISC\_FUNCTION\_MASK、ビット 2 = <値ステータス>

<値>	意味
0	G90 プログラミングでは、360°モジュロ回転軸の位置が標準として AC に対応。 <ul style="list-style-type: none"> <li>● 目標位置 &gt; 現在位置: 正の移動方向に移動</li> <li>● 目標位置 &lt; 現在位置: 負の移動方向に移動</li> </ul>
1	G90 プログラミングでは、360°モジュロ回転軸の位置が標準として DC に対応。つまり、最短軌跡で移動。

### 360°モジュロ表現での位置表示

ユーザーインターフェースで 360°モジュロ表現の位置表示を有効にするには、次の軸別マシデータを使用します。

MD30320 \$MA\_DISPLAY\_IS\_MODULO = 1

## NC/PLC インタフェース信号

### 移動範囲リミットの有効化

原則として、360°モジュロ回転軸の移動範囲は無限です。軸別移動範囲リミットが、ソフトウェアリミットスイッチやワーキングエリアリミットのように、標準としては適用されないのは、このためです。360°モジュロ回転軸の移動範囲リミットは、次のアブソリュート軸別 NC/PLC インタフェース信号を使用して有効にすることができます。この信号は、NC プログラムの M/H 機能で開始される場合があります。

DB31、... DBX12.4 = 1 (360°モジュロ回転軸: トラベル範囲リミット有効)

必要条件:

- 360°モジュロ回転軸が原点確立されていること。
- ソフトウェアリミットスイッチ  
監視対象のソフトウェアリミットスイッチ 1 または 2 を有効にしてください。
  - DB31、... .DBX12.2 / 3 (有効化)
  - MD36100 \$MA\_POS\_LIMIT\_MINUS / MD36110 \$MA\_POS\_LIMIT\_PLUS (ソフトウェアリミットスイッチ 1)
  - MD36120 \$MA\_POS\_LIMIT\_MINUS2 / MD36130 \$MA\_POS\_LIMIT\_PLUS2 (ソフトウェアリミットスイッチ 2)

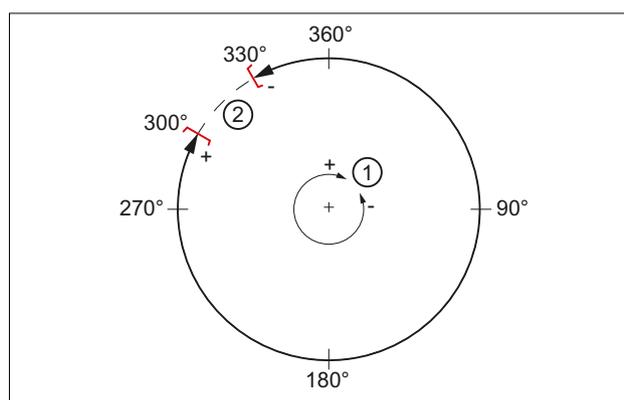
- 作業領域リミット
  - 両方のリミットを、G26/G25 またはセッティングデータを使用して有効にしてください。
  - SD43400 \$SA\_WORKAREA\_PLUS\_ENABLE / SD43410 \$SA\_WORKAREA\_MINUS\_ENABLE (有効化)
  - SD43420 \$SA\_WORKAREA\_LIMIT\_PLUS / SD43430 \$SA\_WORKAREA\_LIMIT\_MINUS (制限)
- STOPRE との同期
 

PLC での M/H 命令を実行した後など、実行中の NC プログラムの NC/PLC インタフェース信号によって移動範囲リミットが有効である場合は、次のブロックで先読み停止をプログラム指令する必要があります(STOPRE)。これにより、移動範囲リミットの有効化が以降の移動ブロックで確実に考慮されるようになります。

### 通知

#### User:有効化と効果

360°モジュール回転軸の移動範囲リミットが、実行プログラムのブロックで NC/PLC インタフェース信号(DB31、... DBX12.4)によって有効になった(M/H 機能を使用した、など)場合に、特定の状況では、以降の移動ブロックの先読みがすでに実行されていたために、移動範囲リミットが考慮されていません。このような場合は、移動範囲リミットを有効にした後に次のブロックで STOPRE をプログラム指令するなど、適切な対策をユーザーが講じる必要があります。これで、NC プログラムで有効にした時点から、以降のすべてのブロックでも移動範囲リミットが有効になります。



- ① 回転方向+/-
- ② ソフトウェアリミットスイッチ+/-

図 14-4 ソフトウェアリミットスイッチのある 360°モジュール回転軸

## フィードバック信号、移動範囲リミットの有効化

360°モジュロ回転軸の移動範囲リミットが NC で有効なフィードバック信号は、軸別 NC/ PLC インタフェース信号を使用して実行します。

DB31、 ... DBX74.4 == 1 (モジュロ回転軸:トラベル範囲リミット有効)

## 例:トラベル範囲リミットの変更

複数のワークをクランプしたパレット 1 を 360°モジュロ回転軸で加工するとします。その場合、パレット 1 を付加軸のパレット 2 と交換します。ここで、トラベル範囲を監視する必要があります。

パラメータ設定:

- MD30300 \$MA\_IS\_ROT\_AX[ AX5 ] = 1
- MD30310 \$MA\_ROT\_IS\_MODULO[ AX5 ] = 1
- MD36110 \$MA\_POS\_LIMIT\_PLUS[ AX5 ] = 340
- MD36100 \$MA\_POS\_LIMIT\_MINUS[ AX5 ] = 350

パートプログラムからの抜粋:

プログラムコード	コメント
M123	; 4つのクランプを搭載したパレット 1 を実装 ; PLC:B 軸 => DB35、DBX12.4=0 の移動範囲リミットを無効化 ;
STOPRE	; 先読み停止処理を起動
S1000 M3	
G4 F2	
G1 X0 Y300 Z500 B0 F5000	
CYCLE84(500,400,0,350,0,1,4,10,,0,500,1000)	; ドリルサイクル
...	
G0 Z540 B0	
M124	; 付加軸のパレット 2 を実装 ; PLC:B 軸=> DB35、DBX12.4=1 の移動範囲リミットを有効化 ;
STOPRE	; 先読み停止処理を起動
B270	

## 14.3 回転軸のプログラミング

### 14.3.1 一般情報

---

#### 注記

プログラミングの概要については、以下を参照してください。

#### 参照先:

『プログラミングマニュアル 基本編』

---

### MD30310

プログラミング中と位置決め中に回転軸が直線軸と同じような動作とするかどうか、または回転軸の特記事項を考慮するかどうか、を定義するには、軸別マシンデータ

**MD30310 ROT\_IS\_MODULO (回転軸のモジュロ変換)**

を使用します。

この機能と違い(おもにアブソリュートプログラミングとの違い)については、次のページで説明します。

### 回転軸と主軸のモジュロ変換

MD30310 「回転軸と主軸のモジュロ変換」

MD30320 「回転軸と主軸のモジュロ表示」

MD30310 と MD30320 を「1」に設定した場合は、現在値表示が、正の回転方向に  $0^{\circ}$ ～ $359.999^{\circ}$ の範囲で変化するか、 $359.999^{\circ}$ を超えるか、または負の回転方向に  $0^{\circ}$ を超えると、**Operate** でのモジュロ表示が直ちに変更されます。

内部の指令値と現在値については、回転軸と主軸の動作が異なります。

- 主軸では、主軸のプログラミング(M3、M4、および M5)ごとにモジュロ変換が実行されます。
- 回転軸では、新しい移動指令によってのみモジュロ変換が実行されます。

指令値と現在値が読み取られた場合、これらは、選択した軸の「サービス概要」-「サービス軸」の表示値と同じです。回転主軸または回転軸では、これらの値が、 $0^{\circ}$ ～ $359,999^{\circ}$ のモジュール範囲を下回る場合もあります。

### 14.3.2 モジュロ変換ありの回転軸(回転軸の連続回転)

#### アブソリュートプログラミング(AC、ACP、ACN、G90)

位置決め軸としてのモジュロ回転軸のアブソリュート指令の例:

POS [<軸名称>] = ACP (<値>)

- <値>:0°～359.999°のモジュロ回転軸の目標位置  
また、次のマシンデータにより範囲オフセットが設定されている場合、負の値も可能です。
  - MD30340 \$MA\_MODULO\_RANGE\_START
  - MD30330 \$MA\_MODULO\_RANGE
- ACP:正の回転方向(左回り)の目標位置のアブソリュート指令。
- ACN:負の回転方向(右回り)の目標位置のアブソリュート指令。
- AC / G90:ブロック/目標位置のモーダルで有効なアブソリュート指令による。  
回転方向は、モジュロ回転軸の現在の位置によって決まります。
  - 目標位置 > 現在位置:正の方向に移動。
  - 目標位置 < 現在位置:負の方向に移動。

---

#### 注記

G90 の位置決め特性は、MD30455 \$MA\_MISC\_FUNCTION\_MASK、ビット 2 で推定できます。「モジュロ 360° (ページ 893)」の章を参照してください。

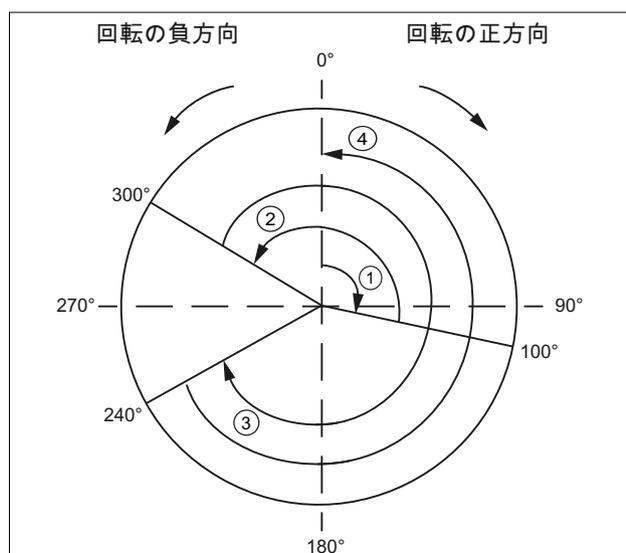
---

#### 用途例

ACP / ACN:回転中に衝突を避けるため非対称ワークに定義された回転方向。

#### プログラミング例

モジュロ回転軸の開始位置 C = 0° (下図を参照してください)。



- ① POS [C] = ACP (100) ; 正の回転方向で 100°位置に移動
- ② POS [C] = ACN (300) ; 負の回転方向で 300°位置に移動
- ③ POS [C] = ACP (240) ; 正の回転方向で 240°位置に移動
- ④ POS [C] = AC (0) ; 負の回転方向で 0°位置に移動

図 14-5 モジュロ回転軸のアブソリュート指令

### 最短軌跡のアブソリュートプログラミング(DC)

位置決め軸としてのモジュロ回転軸の最短軌跡のアブソリュート指令の例:

POS [<軸名称>] = ACP (<値>)

- <値>:0°～ 359.999°のモジュロ回転軸の目標位置  
また、次のマシンデータにより範囲オフセットが設定されている場合、負の値も可能です。
  - MD30340 \$MA\_MODULO\_RANGE\_START
  - MD30330 \$MA\_MODULO\_RANGE
- DC:最短の回転方向による移動の目標位置のアブソリュート指令

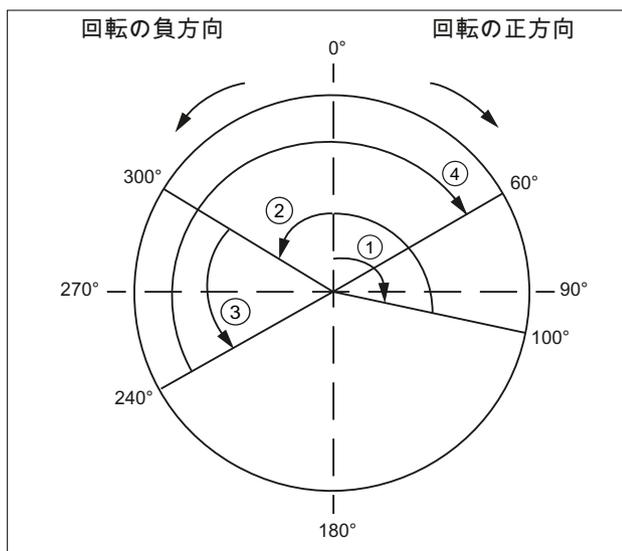
#### 注記

両方向で移動距離が同じ場合(180°)は、正の回転方向となります。

#### プログラミング例

モジュロ回転軸の開始位置 C = 0° (下図を参照してください)。

14.3 回転軸のプログラミング



- ① POS [C] = DC (100) ; 100°位置への最短軌跡による移動
  - ② POS [C] = DC (300) ; 300°位置への最短軌跡による移動
  - ③ POS [C] = DC (240) ; 240°位置への最短軌跡による移動
  - ④ POS [C] = DC (60) ; 60°位置への最短軌跡による移動
- ; 正の回転方向

図 14-6 モジュロ回転軸の最短軌跡の絶対指令

ブロック検索動作

計算ありのブロック検索の後、収集された検索実行位置をモジュロで読み取り、システム変数\$AC\_RETPOINT[<軸>]を使って変換できます。

計算ありブロック検索後の ASUB の補足条件:

この例では、相互チャンネルブロック検索ツール SERUPRO と同じように、ブロック検索でシミュレートされたモジュロ変換をパートプログラムでおこなってください。

インクレメンタルプログラミング(IC、G91)

位置決め軸の例:POS[軸名称] = IC(+/-値)

- 値は回転軸の移動距離を指定します。値は負および+/-360°以上にすることができません。
- 値の符号は、回転軸の移動方向を明確に定義します。
- 用途例:複数回転による螺旋溝のフライス加工

## 例

プログラミング	作用
POS[C] = IC(720)	C 軸は正方向でインクリメンタルに 720°まで移動 (2 回転)
POS[C] = IC(-180)	C は負方向でインクリメンタルに 180°まで移動

## プログラム例:移動制限のない回転軸としてのモジュロ回転軸

## プログラムコード

LOOP:

POS[C] = IC(720) ; 位置決め軸として 270°移動

GOTOB LOOP ; ラベル LOOP に戻る

## 14.3.3 モジュロ変換がない回転軸

## モジュロ変換の無効化

→ MD30310 \$MA\_ROT\_IS\_MODULO = 0 に設定します。

## アブソリュートプログラミング(AC、G90)

位置決め軸の例: POS[軸名称] = AC (+/-値)

- 値とその符号は回転軸の目標位置を固有に指定します。値は+/-360°以上にすることができます。位置データはソフトウェアリミットスイッチ位置によって制限されます。
- 移動方向は、符号付きの回転軸の実位置に従って制御装置で定義されます。
- ACP または ACN をプログラム指令すると、アラーム 16810「ACP 移動命令は使用不可」またはアラーム 16820「ACN 移動命令は使用不可」が出力されます。
- 用途例:  
直線移動(カムギア)は回転軸と連結されているため、特定の終了位置を超えることはできません。

例:

プログラミング	効果
POS[C] = AC (-100)	回転軸 C は位置-100°まで移動。 移動方向は開始位置によって決定。
POS[C] = AC (1500)	回転軸 C は位置 1500°まで移動

### 最短軌跡のアブソリュートプログラミング(DC)

#### POS[軸名称] = DC(値)

回転軸をモジュロ軸として定義していなくても、軸を DC (直接制御)で位置決めできます。動作はモジュロ軸と同じです。

- 値は **0 ~ 359.999°** の範囲で **回転軸の目標位置を指定します(モジュロ 360°)**。負の記号または **360°以上**を指定した値には、アラーム 16830「不正モジュロ位置のプログラム指令」が出力されます。
- DC (直接制御)を使用した場合、回転軸は、**最短軌跡(最大移動± 180°)**によって 1 回転以内で、プログラム指令されたアブソリュート位置にアプローチします。
- 制御装置は現在の実位置に従って、回転方向と移動軌跡を計算します(モジュロ 360°として計算)。移動軌跡が両方向で同じである(180°)場合は、正方向の回転が優先されます。
- DC の使用例:回転テーブルは最短時間(最短軌跡経由)で切り替え位置にアプローチするように要求されます。
- 直線軸に DC をプログラム指令すると、アラーム 16800「DC 移動命令を使用不可」が出力されます。

例:

プログラミング	効果
POS[C] = AC (7200)	回転軸 C は位置 7200°まで移動。 移動方向は開始位置によって決定。
POS[C] = DC (300)	回転軸 C は「モジュロ」位置 300° に最短軌跡でアプローチ  C は負の回転方向に 60°まで移動し、アブソリュート位置 7140°で停止
POS[C] = AC (7000)	回転軸 C はアブソリュート位置 7000°に移動。 負の回転方向で 140°まで移動

---

**注記**

この例では、モジュロ 360°とする表示を有効にしてください(MD30320 \$MA\_DISPLAY\_IS\_MODULO = 1)。

---

### インクリメンタルプログラミング(IC、G91)

位置決め軸の例:POS[軸名称] = IC(+/-値)

インクリメンタル指令でプログラミングすると、回転軸はモジュロ軸と同じ軌跡を移動します。しかしこの場合、移動範囲はソフトウェアリミットスイッチによって制限されます。

- 値は回転軸の移動距離を指定します。  
値は負および+/-360°以上にすることができます。
- 値の符号は、回転軸の移動方向を定義します。

### 移動範囲制限付き

移動範囲は、直線軸と同様に制限されます。範囲制限は、「プラス」および「マイナス」のソフトウェアリミットスイッチで定義します。

### 14.3.4 回転軸に関連するその他のプログラミング機能

#### オフセット

TRANS (アブソリュート)と ATRANS (インクレメンタル)を回転軸に適用できます。

#### スケーリング

SCALE と ASCALE は回転軸に対応していません。コントローラでは、モジュロ計算が 360° の完全な一周円で常におこなわれるからです。

#### 現在位置設定

PRESETON が可能です。

#### 割り出し軸

「T1:割り出し軸 (ページ 991)」の章を参照してください。

## 14.4 回転軸の有効化

### 手順

回転軸を有効にする手順は直線軸を有効にする手順と同じですが、例外が少しあります。軸を回転軸として定義すると(MD30300 \$MA\_IS\_ROT\_AX = 1)、コントローラでは軸別マシンデータの単位とセッティングデータの単位は次のように解釈されますので、注意してください。

位置	「度」
速度	「rev/min」
加速度	「rev/sec <sup>2</sup> 」
加々速度制限	「rev/sec <sup>3</sup> 」

## 特別マシンデータ

用途に応じて、回転軸の特別マシンデータも入力できます。

MD30310 \$MA_ROT_IS_MODULO	位置決めとプログラミング用のモジュロ変換
MD30320 \$MA_DISPLAY_IS_MODULO	位置表示用のモジュロ変換
MD10210 \$MN_INT_INCR_PER_DEG	回転位置の計算分解能

次表は、可能な回転軸のマシンデータの組み合わせを示しています。

可能な回転軸マシンデータの組み合わせ				
MD30300	MD30310	MD30320	アプリケーションの許可	コメント
0	0	0	あり	軸は直線軸です(初期設定)。
1	0	0	あり	軸は回転軸です。位置決めにモジュロ変換を使用しません。ソフトウェアリミットスイッチは有効です。位置表示はアブソリュートです。
1	0	1	あり	軸は回転軸です。位置決めにモジュロ変換を使用しません。ソフトウェアリミットスイッチは有効です。位置表示はモジュロです。 用途:たとえば、動作範囲が $\pm 1000^\circ$ の軸など
1	1	1	あり	軸は回転軸です。位置決めはモジュロ変換でおこなわれます。ソフトウェアリミットスイッチは無効です。動作範囲は制限なしです。位置表示はモジュロです(一般的に回転軸に使用される設定)。ワーキングエリアリミット有り、または無しの軸を使用できます。

## 14.5 回転軸の特記事項

可能な回転軸マシンデータの組み合わせ				
1	1	0	あり	軸は回転軸です。位置決めはモジュロ変換でおこなわれます。ソフトウェアリミットスイッチは無効です。動作範囲は制限なしです。位置表示はアブソリュートです。ワーキングエリアリミット有り、または無しの軸を使用できます。
0	0 または 1	0 または 1	推奨なし	軸は回転軸ではありません。このため、その他の MD は使用されません。

## 回転軸の JOG 速度

SD41130 \$SN\_JOG\_ROT\_AX\_SET\_VELO (回転軸の JOG 速度)

上のセッティングデータを使用して、すべての回転軸に有効な JOG 速度を定義できます (「H1:手動移動 (ページ 169)」の章を参照してください)。

セッティングデータに値 0 を入力すると、次の軸マシンデータが回転軸の JOG 速度として使用されます。

MD21150 \$MC\_JOG\_VELO (手動軸速度)

## 14.5 回転軸の特記事項

## ソフトウェアリミットスイッチ

旋回軸では、ソフトウェアリミットスイッチとワーキングエリアリミットが有効であり、それが必要です。しかし連続回転する回転軸の場合は(MD30310 \$MA\_ROT\_IS\_MODULO=1)、ソフトウェアリミットスイッチとワーキングエリアリミットを各軸毎に無効にすることができます。

ワーキングエリアリミット有り、または無しのモジュロ回転軸を使用できます。

## 参照先:

『機能マニュアル、基本機能』、「軸監視と保護領域(A3)」

## 回転軸のミラーリング

MIRROR (C) または AMIRROR (C) をプログラム指令して、回転軸にミラーリングを実行できます。

## レファレンス点復帰

参照先:

『機能マニュアル、基本機能』; 「レファレンス点復帰(R1)」

## 回転軸としての主軸

回転軸としての主軸(C 軸運転)の使用に関連する注意事項は、下記を参照してください。

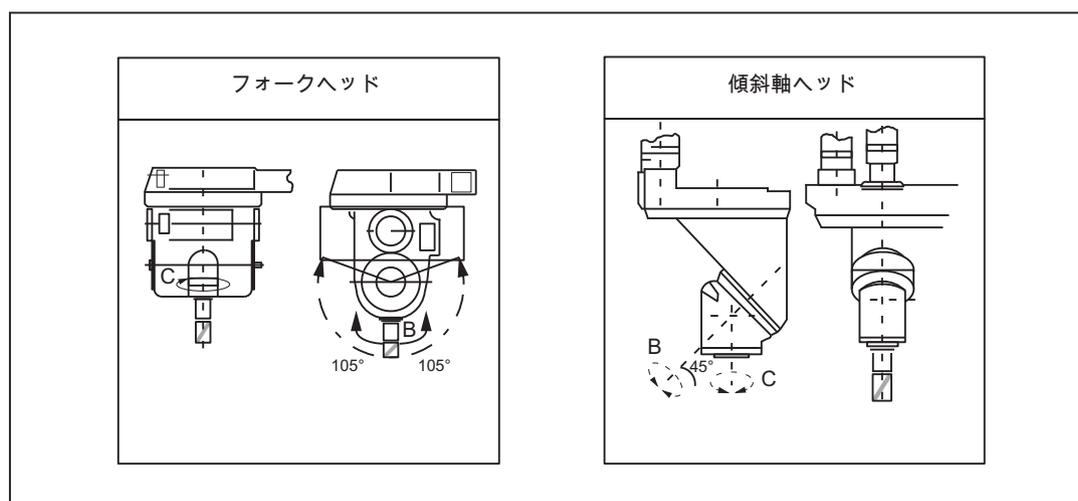
参照先:

『機能マニュアル、基本機能』; 「主軸(S1)」

## 14.6 例

### フォークヘッド、傾斜軸ヘッド

回転軸は一般的に 5 軸フライス盤に使用し、工具軸を旋回したりワークを回転したりします。この機械ではワークの任意の点に工具先端を位置決めして、工具軸の任意の位置を取り入れることができます。用途に応じて、様々なフライスヘッドが必要となります。次の図は、回転軸のアレンジ例として、フォークヘッドと傾斜軸ヘッドを示しています。



## 14.7 データリスト

図 14-7 フォークヘッド、傾斜軸ヘッド

## 14.7 データリスト

## 14.7.1 マシンデータ

## 14.7.1.1 一般マシンデータ

番号	識別子: \$MN_	説明
10210	INT_INCR_PER_DEG	回転位置の計算分解能

## 14.7.1.2 軸/主軸別マシンデータ

番号	識別子: \$MA_	説明
30300	IS_ROT_AX	軸は回転軸です
30310	ROT_IS_MODULO	回転軸のモジュロ変換
30320	DISPLAY_IS_MODULO	フィードバック値のモジュロ表示
30330	MODULO_RANGE	モジュロ範囲のサイズ
30340	MODULO_RANGE_START	モジュロ範囲の開始位置
30455	MISC_FUNCTION_MASK	軸機能
36100	POS_LIMIT_MINUS	マイナスソフトウェアリミットスイッチ
36110	POS_LIMIT_PLUS	プラスソフトウェアリミットスイッチ

## 14.7.2 セッティングデータ

## 14.7.2.1 一般セッティングデータ

番号	識別子: \$SN_	説明
41130	JOG_ROT_AX_SET_VELO	回転軸の JOG 速度

## 14.7.2.2 軸/主軸別セッティングデータ

番号	識別子: \$SA_	説明
43420	WORKAREA_LIMIT_PLUS	プラスワーキングエリアリミット
43430	WORKAREA_LIMIT_MINUS	マイナスワーキングエリアリミット

## 14.7.3 信号

## 14.7.3.1 軸/主軸への信号

信号名称	SINUMERIK 840D sl	SINUMERIK 828D
モジュロ軸の移動範囲リミット位置	DB31, ... .DBX12.4	DB380x.DBX1000.4

## 14.7.3.2 軸/主軸からの信号

信号名称	SINUMERIK 840D sl	SINUMERIK 828D
モジュロ軸のソフトウェアリミットスイッチ監視 状態	DB31, ... .DBX74.4	DB390x.DBX1000.4



## S3:主軸同期

### 15.1 概略説明

#### 15.1.1 機能

「主軸同期」機能を使用すると、位置同期か速度同期で2つの主軸を連結できます。片方の主軸をマスタ主軸(LS)として定義し、残りの主軸はスレーブ主軸(FS)になります。

$$\begin{array}{lll} \text{速度同期制御:} & n_{FS} = k_U * n_{LS} & k_U = 1, 2, 3, \dots \\ \text{位置同期制御:} & \varphi_{FS} = \varphi_{LS} + \Delta\varphi & 0^\circ \leq \Delta\varphi < 360^\circ \end{array}$$

#### 適用可能な用途

##### 背面加工

たとえば使用例として、停止するまで減速せずに同期位置 LS から FS(またはその反対)にワークを高速で移動する2主軸旋盤の背面加工を挙げることができます。

##### 多角形加工(ポリゴン加工)

「主軸同期」機能は、LS と FS の間の整数ギア比  $k_U$  の指定によって、多角形加工(ポリゴン加工)の基礎になります。

#### FS の数

LS と同期的に動作できる FS の数は、使用する NC の性能によってのみ制限されます。原則的には、NC の任意のチャンネルで、任意の数の FS を1つの LS に同時に連結できます。

それぞれの NC チャンネルでは、2つのペアの主軸同期を同時に有効にすることができます。

#### 定義

主軸同期の FS と LS のペアの割り当ては、マシンデータによってチャンネル別にパラメータ設定したり、パートプログラム命令によって柔軟に定義することができます。

## 選択と選択解除

主軸同期のペアの同期運転を選択/選択解除するには、パートプログラムの命令を使用します。

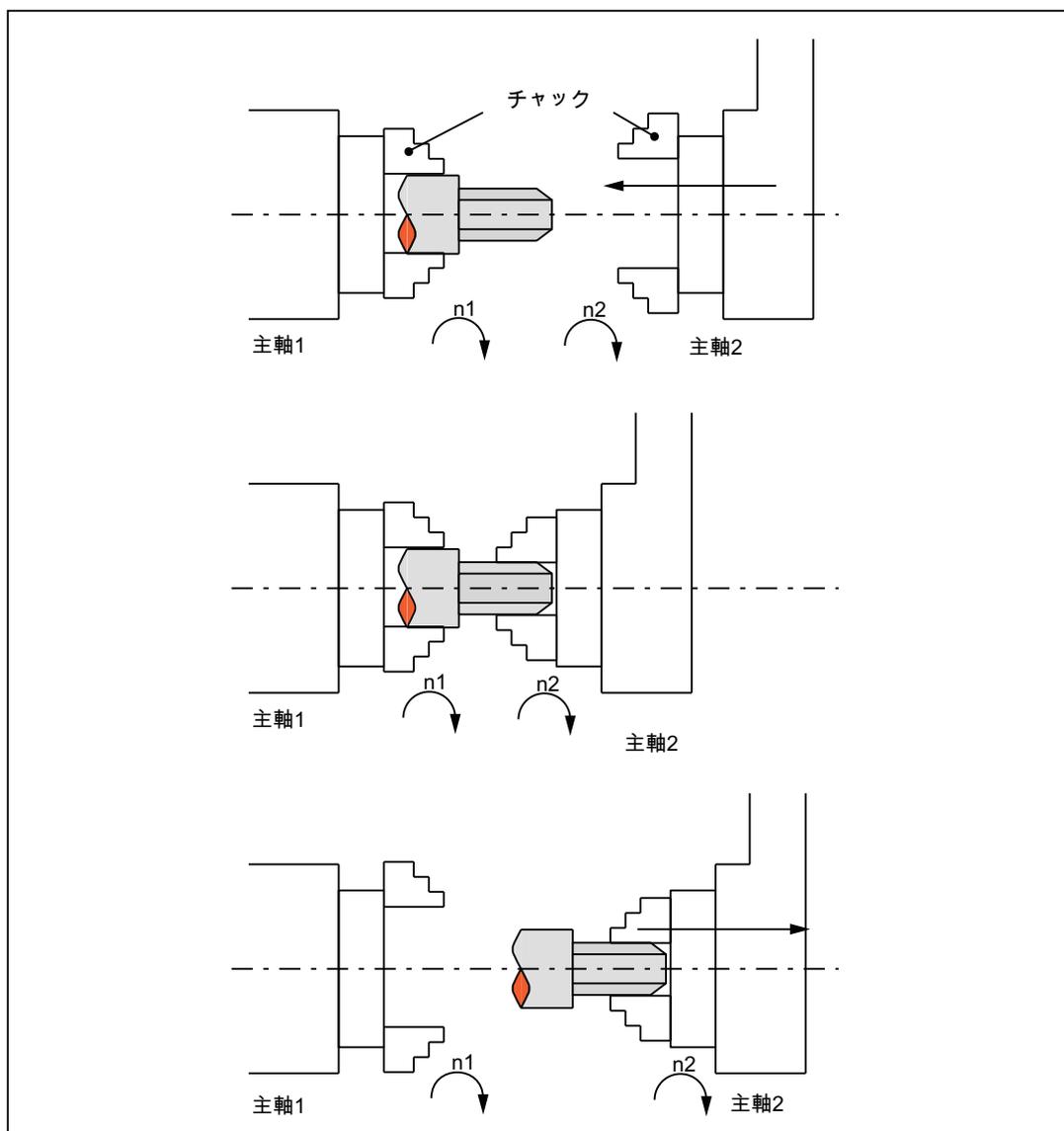


図 15-1 同期運転:主軸 1 から主軸 2 への高速ワーク移動

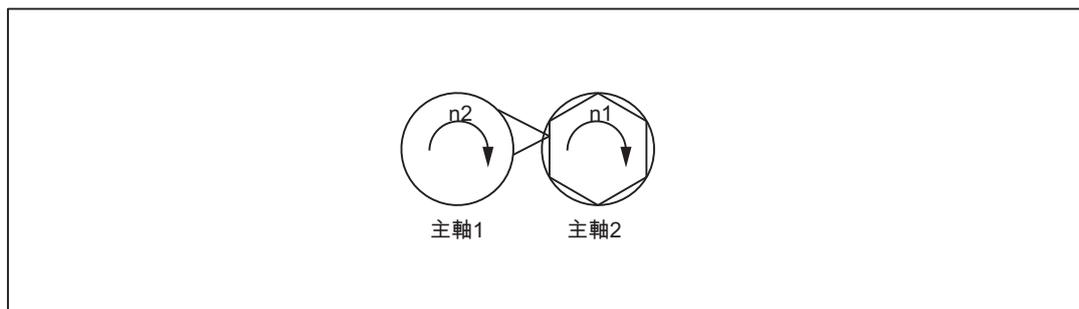


図 15-2 同期運転:ポリゴン加工

## 15.1.2 同期モード

### 説明

<軸の形式>:	次のうちいずれか。 - 軸名称 - 主軸名称
<軸名称>:	C (軸運転の主軸の名称が「C」である場合)
<主軸名称>:	Sn、SPI(n)。n = 主軸番号
<主軸番号>:	1、2、... MD35000 \$MA_SPIND_ASSIGN_TO_MACHAX で定義する主軸番号
(FS、LS、オフセット):	LS = マスタ主軸、FS = スレーブ主軸、オフセット = システム変数を使用したスレーブ主軸のプログラマブルオフセットの読み取り
\$P_COUP_OFFS[Sn]	主軸同期のプログラム指令位置オフセット

### 主軸同期ペア

同期運転には、スレーブ主軸(FS)とマスタ主軸(LS)が関連し、これは**主軸同期ペア**と呼ばれます。連結が有効であるとき(同期運転)、定義されている機能の相互関係に従って、スレーブ主軸はマスタ主軸の移動をコピーします。

## 同期モード

同期モード(「主軸同期運転」とも呼ばれる)は、もう一つの主軸運転モードです。同期モードを有効にする前に、追従(スレーブ)主軸を位置制御に切り替えてください。同期運転は、連結を有効にすると、スレーブ主軸に対して有効になります。連結を無効にすると、スレーブ主軸は開ループ制御モードに戻ります。

スレーブ主軸に同期運転を有効にすると、次のインタフェース信号を PLC に出力します。

IS「同期モード」(DB31、... DBX84.4) = 1

## 同期主軸数

複数のスレーブ主軸を1つのマスタ主軸に連結できます。このマスタ主軸のスレーブ主軸の数は、各ソフトウェアバージョンの関連バージョンによって決まります。

1台のNCU また別のNCUの任意のチャンネルで任意の数のスレーブ主軸をこのマスタ主軸に連結できます。

1つの主軸が常にマスターになり、連結数は軸数からマスターの数を差し引いた値になります。

## 同期モードのオプション

同期モードには次の機能を使用できます。

- FS と LS を同一速度で回転  
( $n_{FS} = n_{LS}$ ; 速度比  $k_U = 1$ )
- LS と FS の間で同一方向または反対方向で回転  
(座標変換比  $k_U$  を使用して正か負で定義できます)
- スレーブ主軸とマスタ主軸を別々の速度で回転  
( $n_{FS} = k_U \cdot n_{LS}$ ; 速度比  $k_U \neq 1$ )  
用途:ポリゴン加工
- FS と LS の間で回転位置を設定可能( $\varphi_{FS} = \varphi_{LS} + \Delta\varphi$ )  
主軸は、FS と LS の間で定義されている角度オフセットで、同期速度で動作します(位置同期連結)。  
用途:成形ワーク
- 主軸が移動中または停止中に、LS と FS の間で同期運転を起動できます。
- 開ループモードと位置制御モードのすべての機能は、マスタ主軸に使用できます。

- 同期モードが有効でないとき、FS と LS はすべて、その他の主軸モードで運転できません。
- 座標変換比も、主軸が同期モードで移動中に変更できます。
- 主軸同期の連結をオンに切り替えた状態で、LS に対する FS のオフセット(重畳移動)を変更できます。

## 連結オプション

主軸同期の連結は次のように定義できます。

- チャネル別マシンデータによる固定設定  
(以後「**固定連結設定**」と表します)
- パートプログラムで言語命令(COUP...)を使用して自由に定義  
(以後「**ユーザー定義連結**」と表します)

.次の方式があります。

1. 固定連結設定をマシンデータによってプログラム指令できます。また、2 番目の連結もパートプログラムによって自由に定義できます。
2. 連結はマシンデータによって設定されません。この場合、連結をユーザー定義にしてパートプログラムによってパラメータ設定できます。

## 個別のスレーブ主軸補間器

個別の**スレーブ主軸補間器**では、異なるチャネルまたは別の NCU の多くのスレーブ主軸を、ユーザー定義によって単独のマスタ主軸に連結できます。スレーブ主軸補間器は次のとおりです。

- COUPON または COUPONC が有効
- COUPOF または COUPOFS が無効

スレーブ主軸補間器は、COUPON、COUPONC 命令はスレーブ主軸用にプログラム指令されているチャネルに常に位置します。動作中のスレーブ主軸が別のチャネルでプログラム指令されている場合、COUPON/COUPONC によって軸入れ替えが開始されて、自身のチャネルに主軸を呼び出します。

特定の主軸同期機能は、連結用の軸の VDI インタフェース信号を使用して PLC から制御できます。この信号は、スレーブ主軸のみに作用し、マスタ主軸には影響しません(「PLC による主軸同期連結制御 (ページ 927)」の章を参照してください)。

## 主軸同期の定義

同期運転を有効にする前に、連結する主軸(FS、LS)を定義してください。

これは、この用途に応じて次の2つの方法で実行できます。

1. 固定連結設定:  
スレーブ主軸(FS)とマスタ主軸(LS)として機能する機械軸を、チャンネル別 MD 21300 \$MC\_COUPLE\_AXIS\_1[n]で定義します。  
この連結設定で LS と FS としてプログラム指令された機械軸は、NC パートプログラムで変更できません。  
必要に応じて、連結パラメータを NC パートプログラムで変更できます。
2. ユーザー定義連結:  
言語命令「COUPDEF(FS、LS、...)」を使用し、連結の作成と変更を NC パートプログラムで実行できます。新しい連結関係を定義する場合は、既存のユーザー定義連結を事前に削除しなければならないことがあります(言語命令 COUPDEL(FS、LS)を使用)。  
スレーブ主軸とマスタ主軸の軸名称(Sn、SPI(n))を言語命令 COUP...毎に FS と LS でプログラム指令してください、それで主軸同期連結は明確に定義されます。  
次の機械軸の軸別マシンデータに有効な主軸番号を割り当ててください。  
MD35000 \$MA\_SPIND\_ASSIGN\_TO\_MACHAX

IS 「スレーブ主軸有効」 (DB31、... DBX99.1)と IS 「マスタ主軸有効」 (DB31、... DBX99.0)は、軸がマスタ主軸として有効であるかスレーブ主軸として有効であるかを機械軸毎に PLC に示します。

LS は、パートプログラム、PLC またはシンクロナイズドアクションを使用してプログラム指令できます。

## 伝達比

座標変換比は、分子と分母(座標変換比パラメータ)に個別の数値を使用してプログラム指令します。このため、有理数でも非常に正確に座標変換比を指定できます。

一般的に:

$$k_{\ddot{u}} = \text{座標変換比パラメータの分子} / \text{座標変換比パラメータの分母} = \ddot{u}_{\text{numerator}} : \ddot{u}_{\text{denominator}}$$

座標変換比パラメータ( $\ddot{u}_{\text{numerator}}$ 、 $\ddot{u}_{\text{denominator}}$ )の数値の範囲には、コントローラでは事実上制限がありません。

マシンデータによって設定する連結の座標変換比パラメータは、チャンネル別 SD42300:COUPLE\_RATIO\_1[n]で定義できます。また、座標変換比は、言語命令 COUPDEF(FS、LS、 $\ddot{u}_{\text{numerator}}$ 、 $\ddot{u}_{\text{denominator}}$ 、...)で変更することができます。セッティングデータに入力した値は、この場合上書きされません(初期設定)。

NC パートプログラムによって定義した連結の座標変換比は、言語命令 COUPDEF (...)のみで入力できます。

新しい座標変換比パラメータは、COUPDEF 命令が処理されるとすぐに有効になります。

主軸同期連結のプログラミング命令について詳しくは、「主軸同期連結のプログラミング」の章を参照してください。

## 連結特性

主軸同期連結毎に、次の特性を定義できます。

### ● ブロック切り替え動作

連結を有効にしているときは、同期運転の適用、または座標変換比か、定義角度オフセットの速度の変更で、ブロック切り替えのために満たすべき条件を定義できます。

- すぐに実行されるブロック切り替え
- 「精密同期制御」に応じたブロック切り替え
- 「汎用同期制御」に応じたブロック切り替え
- IPOSTOP のブロック切り替え(指令値終点同期制御後)
- WAITC で任意の瞬間の同期制御条件の確認

### ● FS と LS 間の連結タイプ

マスタ主軸の位置指令値または位置フィードバック値は、スレーブ主軸の基準値として使用できます。次の連結タイプを選択できます。

#### - 指令値連結(DV)

位置制御運転で使用します。両方の主軸の制御ダイナミック応答は、できるだけ一致させてください。できるだけ指令値連結を使用してください。

#### - フィードバック値連結(AV)

LS の位置制御が不可能である場合、または FS と LS の間で制御特性の偏差が大きい場合に使用します。FS の指令値は、LS のフィードバック値から得られます。主軸速度にばらつきがある場合は、指令値連結よりも同期制御の質が低下します。

#### - 速度連結(VV)

速度連結は、内部的に指令値連結と同じです。FS と LS の必要条件是少なくなります。位置制御と検出器は、FS と LS には不要です。

FS と LS の間の位置オフセットは定義されません。

設定連結に対する当該の連結特性は、マシンデータ(「設定 (ページ 940)」の章を参照してください)を使用して選択され、ユーザー定義連結に対する連結特性は、COUPDEF 言語命令(「定義 (COUPDEF) (ページ 933)」の章を参照してください)を使用して選択されます。

また、言語命令 COUPDEF を使用して、連結特性タイプとブロック切り替え動作が、固定連結設定のために変更できます。

### 参照先:

『プログラミングマニュアル、上級編』、「主軸同期」。

## 連結特性の変更保護

連結設定パラメータ、座標変換比、連結タイプ、ブロック切り替え動作を NC パートプログラムで変更できるかどうかを定義するには、チャンネル別 MD21340 \$MC\_COUPLE\_IS\_WRITE\_PROT\_1 を使用します。

0:COUPDEF 命令によって連結パラメータを NC パートプログラムで変更できます。

1:連結パラメータを NC パートプログラムで変更できません。変更しようとしても、アラームメッセージで拒否されます。

## 重畳動作

同期運転の場合、主軸同期は、プログラム指令されている座標変換比に従ってマスタ主軸の移動をコピーします。

同時に主軸同期は重畳で移動することもできるので、LS と FS は相互に個別の回転位置で運転できます。

FS の重畳移動は、次のように様々な方法で開始できます。

- **AUTOMATIC** と **MDI** の **FS** のプログラマブル位置オフセット:
  - COUPON と SPOS の言語命令は、有効な同期運転に対して、FS と LS 間で基準位置を変更するために使用できます(「パートプログラムの同期モードの選択 (ページ 923)」の章を参照してください)。
- **FS** の手動位置オフセット:
  - ジョグ (ジョグ連続またはジョグステップ)モード  
同期運転が有効時に、ハンドル、またはプラスかマイナスの移動キーを使用したFSの重畳。
  - **AUTOMATIC** モードと **MDI** モード  
DRF オフセットを使用して、ハンドルによる FS の重畳

FS が重畳移動を実行するとすぐに、IS 「重畳移動」 (DB31、... DBX98.4)は 1 の信号が設定されます。

重畳移動は、COUPON によって最大可能 FS 速度で最適なタイミングで実行されます。SPOS を使用してオフセットを変更する場合、位置決め速度を FA[Sn]で指定し、オーバ

ライドで操作できます(IS 「主軸の送り速度オーバーライド有効」 DB31、... DBX17.0 によって選択可能です)。

---

#### 注記

FA[Sn]による位置決め速度の指定については、下記を参照してください。

#### 参照先:

『機能マニュアル 基本機能』; 「主軸(S1)」、 「主軸モード、位置決め運転」の章

---

### 指令値補正

システム変数\$AA\_COUP\_CORR[Sn]の指令値補正は、位置オフセットと同じ方法で、その後のすべてのスレーブ主軸プログラミングに強く影響し、MCS の DRF オフセットに対応します。

### 例:補正值の確立

連結オフセット 7°を COUPON(....,77)を使用してプログラム指令して、ワーク支持固定具を閉じた結果として機械的なオフセットが 81°になった場合は、補正值 4°が計算されます。

システム変数はスレーブ主軸に次の値を返します。

\$P\_COUP\_OFFS[S2]; プログラム指令位置オフセット = 77°

\$AA\_COUP\_OFFS[S2]; 指令値位置オフセット = 77°

\$VA\_COUP\_OFFS[S2]; フィードバック値位置オフセットは約 77°

\$AA\_COUP\_CORR[S2]; 補正值 = 4°

### 15.1.3 同期モードの必要条件

#### 同期モードを選択する場合の条件

次の条件を満たさないと、主軸同期連結を有効にすることができません。アラームメッセージが発生します。

- 主軸同期連結を事前に定義してください(マシンデータによる固定設定、または COUPDEF を使用したパートプログラムによるユーザー定義)。
- 連結を有効にする NC チャンネルで、連結する主軸を定義してください。  
チャンネル別 MD20070 \$MC\_AXCONF\_MACHAX\_USED  
軸指定 MD35000 \$MA\_SPIND\_ASSIGN\_TO\_MACHAX
- 連結を有効にする NC チャンネルに、スレーブ主軸を割り当ててください。  
軸別 MD30550 AXCONF\_ASSIGN\_MASTER\_CHAN による初期設定
- 指令値連結とフィードバック値連結(DV、AV)には下記が適用されます。  
FS と LS には少なくとも位置記録用に位置検出器を搭載して、位置制御を開始してください。

---

#### 注記

位置制御を有効にすると、LS の最大指令速度は最大速度の 90%(制御制限)に自動的に制限されます。この制限は IS 「指令速度制限」 (DB31、... DBX83.1)によって通知されます。

---

#### 参照先:

『機能マニュアル、基本機能』; 「主軸(S1)」

- 指令値連結(DV)には下記が適用されます。  
さらに正確な同期特性を確保するには、LS を位置制御モードにしてから(言語命令 SPCON)連結を有効にします。
- 同期モードを選択する前に、FS と LS に必要なギア選択をおこなってください。同期モードでは、ギア選択切り替えと揺動モードが FS と LS で不可になります。これを要求すると、アラームメッセージが発生します。

## 相互チャンネル連結

LS は任意のチャンネルに配置できます。

- LS は「軸入れ替え」によってチャンネル間で入れ替えできます。
- 複数のスレーブ主軸を 1 つのマスタ主軸に連結すると、連結のダイナミック応答は連結係数に応じて、最も弱い応答で決まります。加速度と最大速度は、連結されているスレーブ主軸が過負荷にならない程度に、マスタ主軸で小さくします。
- スレーブ主軸は常に、COUPON または COUPONC を使用して連結を有効にした、そのチャンネルに位置します。

### 15.1.4 パートプログラムの同期モードの選択

#### 連結 COUPON、COUPONC の起動

言語命令 COUPON では、最後に有効だったパラメータでプログラム指令された主軸間の連結が、パートプログラムで有効になります。その結果、同期モードも有効になります。この連結は固定設定またはユーザー定義にすることができます。起動時に、マスタ主軸と(または)スレーブ主軸は停止するか、移動することができます。

同期運転を有効にする前に、特定の条件を満たす必要があります(「同期モードの必要条件 (ページ 922)」の章を参照してください)。

COUPONC 命令では、スレーブ主軸とマスタ主軸に以前プログラム指令されていた主軸回転方向と主軸速度がパートプログラムで適用されます。角度オフセットを指定することはできません。

#### COUPON による起動方式

同期モードは次の 2 種類の方法で起動することができます。

1. マスタ主軸とスレーブ主軸の間で**任意の基準角度**を使用する、高速連結の起動。  
COUPON(FS, LS)
2. マスタ主軸とスレーブ主軸の間で**角度オフセット**  $POS_{FS}$  が定義されている連結の起動。  
この方法では、選択時に角度オフセットをプログラム指令してください。  
COUPON(FS, LS,  $POS_{FS}$ )

#### ブロック切り替え動作

同期運転を選択する前に、同期モードの有効時のブロック切り替えをおこなう条件を決めてください(「定義 (COUPDEF) (ページ 933)」の章を参照してください)。

## 現在の連結状態の確認

NC パートプログラムで\$AA\_COUP\_ACT[軸の形式>]軸のシステム変数を使用すると、指定の軸/主軸で現在の連結状態を指定できます(「主軸同期の軸のシステム変数 (ページ 938)」の章を参照してください)。スレーブ主軸で主軸同期連結を有効にするとすぐに、ビット 2 は読み取り時に「1」になります。

## 定義されている角度オフセットの変更

言語命令 COUPON と SPOS では、同期モードが有効時に、定義された角度オフセットを変更できます。スレーブ主軸は、 $POS_{FS}$  でプログラム指令された角度オフセットで重畳移動として位置決めされます。この間は IS 「重畳移動」 (DB31、... DBX98.4) が設定されます。

## 角度オフセット $POS_{FS}$

マスタ主軸の原点位置角度を基準とした場合の正回転方向によるアブソリュート位置として、定義される角度オフセット  $POS_{FS}$  を指定してください。

位置制御主軸の「0°位置」は次のように計算されます。

- 検出器のゼロマークまたは近接センサ信号から計算、および
- 軸別マシンデータを使用して保存された基準値から計算。  
MD34100 \$MA\_REFP\_SET\_POS、レファレンス点位置  
インターバルコードシステムでは意味がありません。  
MD34080 \$MA\_REFP\_MOVE\_DIST インターバルコードシステムによる  
レファレンス点距離または目標点  
MD34090 \$MA\_REFP\_MOVE\_DIST\_CORR レファレンス点オフセット/アブソリュートオフセット(インターバルコードによる)

$POS_{FS}$  の範囲:0 ... 359.999°

### 参照先:

『機能マニュアル、基本機能』; 「レファレンス点復帰(R1)」

## 現在の角度オフセットの読み取り

軸のシステム変数を使用すると、NC パートプログラムで FS と LS 間の現在の位置オフセットを読み取ることができます。以下の区別がされます。

- FS と LS 間の現在の指令値位置オフセット  
\$AA\_COUP\_OFFS [<FS の軸名称>]
- FS と LS 間のフィードバック値の現在の位置オフセット  
\$VA\_COUP\_OFFS [<FS の軸名称>]

(<軸名称>の詳細については、「同期モード (ページ 915)」の章を参照してください)

## 電源投入後の起動

同期モードは、原点確立未完/原点同期未完 FS または LS でも有効にすることができます (IS 「原点確立済み/原点同期済み 1 または 2」 DB31、... DBX60.4 または DBX60.5 = 0)。この場合は警告メッセージが表示されます。

例:

LS と FS は、電源投入後に既にワークによって、摩擦ロックで連結されています。

### 15.1.5 パートプログラムの同期モードの解除

#### 連結解除 (COUPOF、COUPOFS)

指定主軸間の同期モードは、パートプログラムの命令 COUPOF でキャンセルします。3 つの方式があります。

COUPOF を使用して指定主軸間で同期モードをキャンセルする場合、この連結が固定設定されているかユーザー定義であるかは関係ありません。同期運転を無効にするとき、マスタ主軸とスレーブ主軸は停止するか、移動することができます。

COUPOF で同期モードをオフに切り替えると、スレーブ主軸は**コントロールモード**になります。最初からプログラム指令していた S ワードは FS で無効になり、スレーブ主軸はその他の通常の主軸のように操作できます。

COUPOF で連結を無効にすると、一般的には制御装置内部でブロック先読み停止 STOPRE が開始されます。

位置データなしの停止、またはプログラム指令された位置での停止で、できるだけ速く連結を無効にするために、COUPOFS 命令が使用できます。

## COUPOF の種類

COUPOF で同期モードを選択解除するには、次の 3 種類の方法を使用できます。

1. できるだけ高速の連結解除  
ブロック切り替えがすぐに有効になります。  
COUPOF(FS、LS)
2. プログラム指令された解除位置  $POS_{FS}$  をスレーブ主軸が超えるまで、連結は選択解除されません。  
それからブロック切り替えが有効になります。  
COUPOF(FS、LS、 $POS_{FS}$ )
3. プログラム指令されている解除位置の  $POS_{FS}$  と  $POS_{LS}$  をスレーブ主軸とマスタ主軸が超えるまで、連結は選択解除されません。  
それからブロック切り替えが有効になります。  
COUPOF(FS、LS、 $POS_{FS}$ 、 $POS_{LS}$ )

## $POS_{FS}$ 、 $POS_{LS}$

解除位置  $POS_{FS}$  および  $POS_{LS}$  は、定義したレファレンス点の位置データを基準にして、それぞれに FS と LS の実位置と一致します。

$POS_{FS}$ 、 $POS_{LS}$  の範囲: 0 ... 359,999°.

参照先:

機能マニュアル 基本機能、レファレンス点復帰(R1)

## 移動中の COUPOF

COUPOF を使用して主軸の移動中に同期モードを選択解除すると、スレーブ主軸は現在の速度( $n_{FS}$ )で回転し続けます。現在の速度は、NC パートプログラムのシステム変数  $\$AA\_S$  で読み取ることができます。

スレーブ主軸は、M05、SPOS、SPOSA でパートプログラムから、または当該のインタフェース信号で PLC から停止できます。

## スレーブ主軸停止の COUPOFS

主軸同期連結を無効化は、スレーブ主軸の停止により拡張されます。

- できるだけ高速に連結を解除し、できるだけ高速に停止します。  
それからブロック切り替えが有効になります。  
COUPOFS(FS、LS)
- スレーブ主軸のプログラム指令位置での停止により連結を無効にします。それからブロック切り替えが有効になります。

**条件:**

連結が有効である場合、COUPOFS(FS、LS)と COUPOFS(FS、LS、POS<sub>FS</sub>)には意味がありません。

## 15.1.6 PLC による主軸同期連結制御

### PLC によるスレーブ主軸制御

連結用の軸の VDI インタフェース信号を使用すると、PLC プログラムからスレーブ主軸の同期動作を制御できます。これにより、オフセットプログラミングによって、指定したスレーブ主軸の同期動作を無効にしたり、抑制したり、再起動したりするために PLC を利用するオプションが提供されます。

この信号はマスタ主軸には無効です。次の連結用 VDI 信号(PLC → NC)を使用できます。

#### IS 「同期無効」 (DB31、... DBX31.5)

#### 「同期無効」

スレーブ主軸の同期動作は、軸の信号 IS 「同期無効」 (DB31、... DBX31.5)を使用して抑制します。

パートプログラム命令 COUPON (FS、LS、offset)を含むブロックまでメインランが進むと、スレーブ主軸で次のインタフェース信号が使用されます。

#### IS 「同期無効」 (DB31、... DBX31.5)

- IS 「同期無効」 (DB31、... DBX31.5) = 0 の場合、位置オフセットが以前のように移動されます。
- IS 「同期無効」 (DB31、... DBX31.5) = 1 の場合、連続運転速度同期制御のみが確立します。スレーブ主軸は追加動作を実行しません。  
連結は、プログラム指令 COUPON (FS、LS)と同じように応答します。

#### 特記事項

次のように生成されたスレーブ主軸の IS 「同期無効」 (DB31、... DBX31.5)オフセット動作は、制御できません。

- SPOS、POS
- シンクロナイズドアクション

## 15.1 概略説明

- FC18 (840D sl の場合)
- JOG

この機能は、VDI 信号 IS 「送り速度停止または主軸停止」 (DB31、... DBX4.3)で制御します。

## 同期状態への到達

同期状態に到達すると、同期が無効であったかどうかに関係なく、次の 2 つの VDI 信号が設定されます。

IS 「汎用同期」 (DB31、... DBX98.1)

IS 「精密同期」 (DB31、... DBX98.0)

COUPON の後のブロック切り替えは、同期の抑制によって防ぐことができません。

## 例

COUPON 後のブロック切り替え動作

プログラムコード	コメント
N51 SPOS=10 SPOS[2]=10	; IS 「同期無効」 ; S2 には (DB31, ... DBX31.5) = 1 を設定 ; 位置は 0° のオフセットに ; 対応
N52 COUPDEF(S2,S1,1,1,「FINE」,「DV」)	
N53 COUPON(S2,S1,77)	; 実オフセットの 0° を維持 ; スレーブ主軸の移動なし ; VDI 信号 ; IS 「汎用同期」 ; (DB31、... DBX98.1) および ; IS 「精密同期」 ; (DB31、... DBX98.0) ; を設定し、ブロック切り替え ; を有効化
N54 M0	
N57 COUPOF(S2,S1)	
N99 M30	

## リセットと回復

IS 「同期無効」 (DB31、... DBX31.5)をリセットしても、スレーブ主軸オフセットは影響されません。スレーブ主軸のオフセット移動を VDI インタフェース信号で抑制している場合、VDI 信号をリセットしても、オフセットは自動的に適用されません。

同期は次のように回復します。

- パートプログラムの命令 COUPON (FS、LS、offset)を IS 「同期無効」 (DB31、... DBX31.5) = 0 で繰り返します。  
COUPON (FS、LS、offset)を ASUB など書き込むことができます。
- IS 「再同期」 (DB31、... DBX31.4) = 1 を設定します。

## オフセットの読み取り

次のシステム変数を使用すると、パートプログラムとシンクロナイズドアクションからスレーブ主軸の 3 種類の位置オフセット値を読み取ることができます。変数 \$P\_COUP\_OFFS[Sn]はパートプログラムのみで使用できます。

説明	NC 変数
主軸同期のプログラム指令位置オフセット	\$P_COUP_OFFS[Sn]
主軸同期の位置オフセット、指令値終了	\$AA_COUP_OFFS[Sn]
主軸同期の位置オフセット、フィードバック値終点	\$VA_COUP_OFFS[Sn]

## 「送り速度停止/主軸停止」

MD30455 MISC\_FUNCTION\_MASK のビット 4 の設定で、軸 IS 「送り速度停止/主軸停止」 (DB31、... DBX4.3)の動作がスレーブ主軸用に定義されます。

ビット 4 = 0 互換性対応:

スレーブ主軸の送り有効をキャンセルすると、連結部が減速します。

ビット 4 = 1:

送り速度有効は補間成分(SPOS,..)のみに作用し、連結には影響しません。

### 注記

MD30455 \$MA\_MISC\_FUNCTION\_MASK を使用した軸機能のその他の設定オプション:

参照先:

『機能マニュアル、基本機能』、「回転軸(R2)」: 「回転軸のプログラミング」の章

### 15.1.7 同期運転の監視

#### 精密/汎用同期

通常主軸監視運転の他に、同期モードでは、FS と LS 間の同期運転も監視できます。

IS「精密同期」(DB31、... DBX98.0)またはIS「汎用同期」(DB31、... DBX98.1)が PLC に伝送され、スレーブ主軸の現在位置(AV、DV)や実速度(VV)が指定の許容範囲内に収まっているかどうかを示します。

連結をオンに切り替えると、指令値同期に到達したとき、信号「汎用同期」と「精密同期」が更新されます。

許容範囲は FS のマシンデータで設定します。同期の到達には次の要素が作用します。

- AV、DV: FS と LS 間の位置の差
- VV: FS と LS 間の速度差

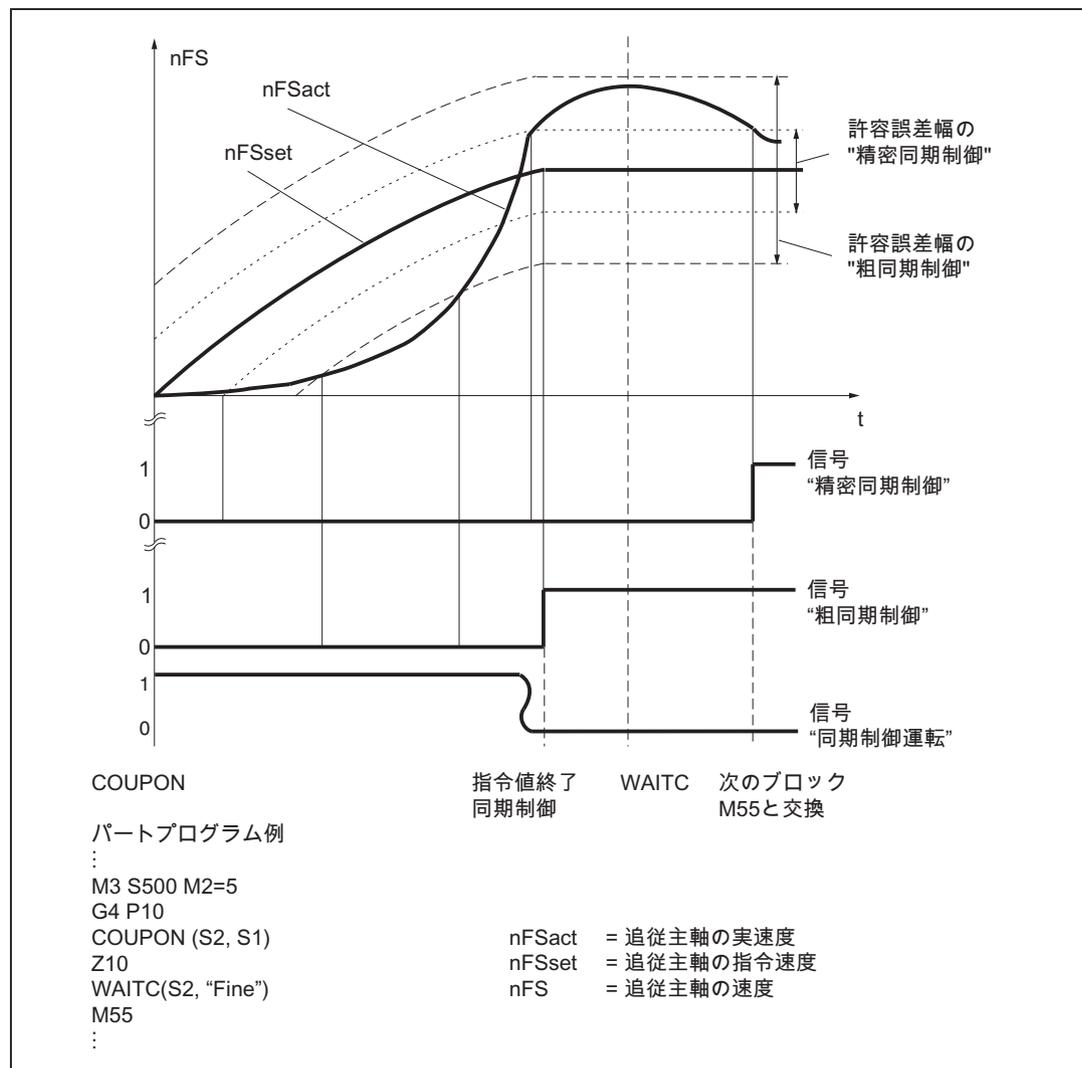


図 15-3 COUPON による同期監視と回転マスタ主軸の同期制御による同期テストマーク  
WAITC

## しきい値

マスタ主軸を基準にしたスレーブ主軸の当該の位置または速度許容範囲は、度または 1 rev/min で指定してください。

- 「汎用同期」のしきい値  
軸指定 MD37200: AV、DV: COUPLE\_POS\_TOL\_COARSE  
MD37220: VV: COUPLE\_VELO\_TOL\_COARSE
- 「精密同期」のしきい値  
軸指定 MD37210: AV、DV: COUPLE\_POS\_TOL\_FINE  
MD37230: VV: COUPLE\_VELO\_TOL\_FINE

## 速度/加速度制限

同期モードの場合、マスタ主軸の速度と加速度の制限値は、スレーブ主軸がその移動をコピーする方法でコントローラ内部で調整され、それぞれの制限値に違反せずに、現在選択しているギア選択と有効速度比にできます。

たとえば LS は自動的に減速し、FS が主軸間の同期を保つために最大速度を超えないようにします。

## 15.2 プログラミング

## マスタ主軸プログラミングによる連結命令

命令	機能
COUPDEF (FS, LS, ...)	連結の定義または設定された連結の変更を行います
COUPON (FS, LS, POSFS)	連結の有効化
COUPONC (FS, LS)	スレーブ主軸の現在有効な速度の伝送により連結を有効にします
COUPOF (FS, LS, POSFS, POSLS)	連結をオフにします
COUPOFS (FS, LS, POSFS)	スレーブ主軸を停止することにより連結を無効にします
COUPDEL (FS, LS)	連結の解除
COUPRES (FS, LS)	設定済みの連結データの再有効化

## マスタ主軸プログラミングによらない連結命令

命令	機能
COUPOF (FS, POSFS, POSLS)	連結をオフにします
COUPOFS (FS, POSFS)	スレーブ主軸を停止することにより連結を無効にします
COUPDEL (FS)	連結の解除
COUPRES (FS)	設定済みの連結データの再有効化

## 下記も参照

定義 (COUPDEF) (ページ 933)

連結(COUPON、COUPONC、COUPOF)のオンとオフを切り換えます (ページ 937)

## 15.2.1 定義 (COUPDEF)

## プログラム指令可能な連結

使用できる軸数に応じて、プログラム指令できる連結数が決まります。この数は、軸や主軸の数からマスター軸の1を差し引いた数になります。さらに前バージョンのSWと同じように、1つの連結をマシンデータによって設定することもできます。

## 固定連結設定

固定設定した主軸同期連結の連結特性と座標変換比は、書き込み保護されていない場合、NCパートプログラムで変更できます。LSとFSの機械軸を変更することはできません。

## 新しい連結の定義

新しい主軸同期連結(ユーザー定義)を作成し、既存の連結のパラメータを変更するには、言語命令「COUPDEF」を使用できます。

連結パラメータを完全に指定するときは、下記が適用されます。

**COUPDEF(FS, LS, T<sub>分子</sub>, T<sub>分母</sub>, ブロック切り替え動作, 連結タイプ)**

主軸同期連結はFSとLSで明確に定義します。

その他の連結パラメータは、変更が必要なときにプログラム指令してください。最後の有効状態が、未指定パラメータに適用可能となります。

個々の連結パラメータについて下記に説明します。

- **FS、LS:** スレーブ主軸とマスタ主軸の主軸名称  
例: S1、SPI(1)、S2、SPI(2)  
有効な主軸番号は、機械軸の軸別 MD35000 \$MA\_SPIND\_ASSIGN\_TO\_MACHAX で割り当ててください。
- **T<sub>分子</sub>, T<sub>分母</sub>:** 座標変換比パラメータの分子と分母  
座標変換比は、分子と分母の数値という形式で指定します(「同期モード(ページ 915)」の章を参照してください)。  
分子は常にプログラム指令してください。分母を指定しない場合は、その値として常に「1.0」が想定されます。
- **ブロック切り替え動作**  
このパラメータでは、同期運転を選択したときにブロック切り替えをおこなうタイミングを選択できます。

**NOC:** ブロック切り替えはすぐに有効

**FINE:** 「精密同期制御」に応じたブロック切り替え

**COARSE:** 「汎用同期制御」に応じたブロック切り替え

**IPOSTOP:** IPOSTOP のブロック切り替え(指令値終点同期制御後)

ブロック切り替え動作を文字列として指定します(引用符を使用)。

ブロック切り替え動作は、単に太字で文字を書き込んで指定できます。残りの文字は、パートプログラムを読みやすくするために入力できますが、意味はありません。

ブロック切り替え動作を指定しない場合は、現在選択している動作が引き続き適用されます。

指令可能な同期テストマーカー **WAITC** を使用すると、パラメータ指定の同期が達成されるまで、新しいブロックへの切り替えは遅れます。

- **連結タイプ**

**DV** (目標値): FS と LS の間の指令値連結

**AV** (フィードバック値): FS と LS 間の現在値連結

**VV** (速度値): FS と LS の間の速度連結

連結タイプを指定しない場合は、現在選択しているタイプが引き続き適用されます。

---

**注記**

連結タイプは、同期運転を解除したときに限って変更できます。

---

**例**

COUPDEF (SPI(2), SPI(1), 1.0, 1.0, 「FINE」, 「DV」 )

COUPDEF (S2, S1, 1.0, 4.0)

COUPDEF (S2, SPI(1), 1.0)

**初期設定**

次の初期設定がユーザー定義連結に適用されます。

- $\ddot{U}_{\text{Numerator}}=1.0$
- $\ddot{U}_{\text{Denominator}}=1.0$
- ブロック切り替え動作 = **IPOSTOP** (指令値同期制御でブロック切り替えを有効化)
- 連結タイプ = **DV** (指令値連結)

**連結削除**

ユーザー定義連結を削除するには、言語命令「COUPDEL」を使用します。

**COUPDEL (FS, LS)**

---

**注記**

COUPDEL は有効な連結に影響して、それを解除して連結データを削除します。このため、アラーム 16797 には意味がありません。

スレーブ主軸は最後の速度を適用します。これは、COUPOF(FS、LS)の関連動作と同じです。

---

**元の連結パラメータの有効化**

言語命令「COUPRES」を使用すると、連結設定パラメータを再び有効にすることができます。

**COUPRES (FS, LS)**

COUPDEF を使用して変更したパラメータ(座標変換比も含む)は、その後に削除されます。言語命令「COUPRES」では、機械とセッティングデータに保存されているパラメータ(連結設定)が有効になり、初期設定(ユーザー定義連結)が有効になります。

### プログラム可能なブロック切り替え

言語命令「WAITC」を使用して、NC プログラムの 1 点をマークできます。システムは、指定されている FS の同期条件の成立をこの点で待機し、指定されている同期条件が達成されるまで新しいブロックへの変更を遅らせます(「図 15-3 COUPON による同期監視と回転マスタ主軸の同期制御による同期テストマーク WAITC (ページ 931)」を参照してください)。

#### WAITC (FS)

特長: 技術上の面から見ると、同期連結を有効にしてから同期が達成されるまでの時間を有効に使用できます。

---

#### 注記

基本的には、常に WAITC を書き込むことは可能です。指定の主軸が FS として有効でない場合、この主軸命令は動作しません。

---

同期条件を指定していない場合は、それぞれの連結でプログラム指令されている、または設定されている同期条件は、少なくとも指令値同期で常にチェックされます。

#### 例:

```
WAITC(S2)、  
:  
WAITC(S2, 「Fine」 )、  
:  
WAITC(S2, S4, 「Fine」 )
```

### 停止とブロック切り替え

マスタ主軸かスレーブ主軸の有効軸のキャンセル期間に「停止」を有効にすると、サーボドライブからの有効軸の設定で、再び最後の指令位置へアプローチします。

命令 COUPON および WAITC は、ブロック変更の特性に作用する場合があります。ブロック切り替え条件は、COUPDEF または MD21320 \$MC\_COUPLE\_BLOCK\_CHANGE\_CTRL\_1 を使用して定義します。

## 15.2.2 連結(COUPON、COUPONC、COUPOF)のオンとオフを切り換えます

### 同期モードの起動

連結と同期モードを起動するには、言語命令 COUPON を使用します。

同期運転を起動にする場合は、次の 2 つの方法を使用できます。

1. **COUPON(FS, LS)**  
リード主軸と追従主軸の間の基準角度で、同期運転を可能な限り最高速で起動します。
2. **COUPON(FS, LS, POS<sub>FS</sub>)**  
リード主軸と追従主軸の間で角度オフセット POS<sub>FS</sub> を定義した同期運転の起動します。このオフセットは、リード主軸の原点位置角度を基準にして、正回転方向で行われます。ブロック切り替えは、定義設定に従って有効になります。POS<sub>FS</sub> の範囲: 0 ... 359.999 度。
3. **COUPONC(FS, LS)**  
COUPONC によって起動するとき、以前のプログラミングの M3 S... または M4 S... が適用されます。速度差はすぐに伝送されます。オフセット位置はプログラム指令できません。

同期運転がすでに有効であるときに COUPON(FS、LS、POS<sub>FS</sub>) または SPOS をプログラミングすると、LS と FS の間の角度オフセットを変更できます。

### 同期モードの解除

同期モードは次の 3 種類の方法で解除することができます。

1. **COUPOF(FS, LS)**  
可能な限り高速で同期運転を解除します。ブロック切り替えがすぐに有効になります。
2. **COUPOF(FS, LS, POS<sub>FS</sub>)**  
解除位置 POS<sub>FS</sub> を超えた後で、同期運転を選択解除します。この位置を超えるまで、ブロック切り替えは有効になりません。
3. **COUPOF(FS, LS, POS<sub>FS</sub>, POS<sub>LS</sub>)**  
2 つの解除位置 POS<sub>FS</sub> と POS<sub>LS</sub> を超えた後で、同期運転を選択解除します。両方のプログラム指令位置を超えるまで、ブロック切り替えは有効になりません。  
POS<sub>FS</sub>、POS<sub>LS</sub> の範囲: 0 ... 359,999°。

連続軌跡制御(G64)をプログラム指令した場合は、制御装置内部でノンモーダル停止が生成されます。

例:

```
COUPDEF (S2, S1, 1.0, 1.0, 「FINE」, 「DV」)
:
COUPON (S2, S1, 150)
:
COUPOF (S2, S1, 0)
```

:

COUPDEL (S2, S1)

## 1. COUPOFS(FS, LS)

スレーブ主軸の停止で連結を解除します。即時のブロック切り替えにより、高速でブロック切り替えをおこないます。

2. COUPOFS(FS, LS, POS<sub>FS</sub>)

機械座標系基準にプログラム指令した解除位置を超えた後に、解除位置 POSFS を超えるまで、ブロック切り替えは有効になりません。

数値の範囲: 0 ... 359.999°.

## 15.2.3 主軸同期の軸のシステム変数

## 現在の連結状態の確認

スレーブ主軸の現在の連結状態は、次の軸のシステム変数を使用して NC パートプログラムで読み取ることができます。

**\$AA\_COUP\_ACT[<軸の形式>]**

(<軸の形式>の詳細については、「同期モード (ページ 915)」の章を参照してください)

例:

**\$AA\_COUP\_ACT[S2]**

読み取った値には、スレーブ主軸について次の意味があります。

バイト = 0: 連結は無効

ビット 2 = 1: 主軸同期連結が有効

ビット 2 = 0: 主軸同期連結が有効になっていない

## 現在の角度オフセットの読み取り

FS と LS 間の現在の位置オフセットは、次の軸のシステム変数を使用して NC パートプログラムで読み取ることができます。

- FS と LS 間の指令値ベースの位置オフセット:

**\$AA\_COUP\_OFFS[<軸の形式>]**

- FS と LS 間のフィードバック値ベースの位置オフセット:

**\$VA\_COUP\_OFFS[<軸の形式>]**

例:

**\$AA\_COUP\_OFFS[S2]**

COUPON で角度オフセットをプログラム指令している場合、これは指令値同期の解釈後に、読み取った値と一致します。

**プログラム指令した角度オフセットの読み取り**

FS と LS の間で最後にプログラムした位置オフセットは、次の軸のシステム変数を使用して NC パートプログラムで読み取ることができます。

**\$P\_COUP\_OFFS[<軸の形式>]****注記**

同期運転とフォローアップモードが有効であるときのサーボイネーブル信号のキャンセル後、コントローラが再び有効になったときに適用される位置オフセットは、最初のプログラム指令値と異なります。

**\$P\_COUP\_OFFS** では、最初のプログラム指令値のみが返されます。**\$AA\_COUP\_OFFS** と **\$VA\_COUP\_OFFS** は現在の値を返します。プログラム指令したオフセットは、IS DB31, ... DBX31.4 で再度作成できます(再同期)。

**15.2.4 位置制御の自動選択と選択解除****速度制御モードの動作**

DV 連結モードでは、プログラム命令 COUPON、COUPONC、および COUPOF、COUPOFS を使用して、必要に応じてリード主軸の位置制御の有効化および(または)無効化を行います。リード主軸上に複数の追従主軸があるとき、速度制御モードでは、SPCON のプログラム指令がない場合、最初の DV がリード主軸の連結位置制御を有効にして最後の DV 連結がリード主軸の連結位置制御を無効にします。

リード主軸を追従主軸と同じチャンネルに配置する必要はありません。

### COUPON と COUPONC による自動選択

連結タイプに応じて、同期運転の位置制御の COUPON と COUPONC の動作は次のとおりです。

連結タイプ	DV	AV	VV
追従主軸 FS	位置制御オン	位置制御オン	動作なし
リード主軸 LS	位置制御オン <sup>1</sup>	動作なし	動作なし

<sup>1</sup> 最低 1 つの追従主軸がリード主軸に連結タイプ DV で連結している場合は、COUPON 命令と COUPONC 命令で位置制御を有効にします。

### COUPOF と COUPOFS による自動選択解除

連結タイプに応じた、位置制御の COUPOF と COUPOFS の動作は次のとおりです。

連結タイプ	DV	AV	VV
追従主軸 FS	位置制御オフ <sup>2</sup>	位置制御オフ <sup>2</sup>	動作なし <sup>2</sup>
リード主軸 LS	位置制御オフ <sup>3</sup>	動作なし	動作なし

<sup>2</sup> 位置指定無しの COUPOF と COUPOFS

速度制御モードは追従主軸で有効です。位置決めモードは、停止位置を指定した COUPFS で有効です。SPCON を使用して追従主軸を位置制御主軸モードにした、または位置指定で COUPFS をプログラム指令した場合、位置制御は無効になりません。

<sup>3</sup> COUPOF では、このリード主軸に、他に DV 連結タイプの連結がない場合、位置制御は無効になります。リード主軸が位置決めモードか軸のモードになっている場合、または SPCON を使用して位置制御主軸モードになっている場合、位置制御は維持されます。

## 15.3 設定

#### 注記

各チャンネルについて、1 つの主軸同期連結を設定できます。

表 15-1 マシンデータ

番号	名称:\$MC_	機能
MD21300	COUPLE_AXIS_1[<n>]	<p>主軸同期連結の機械軸:</p> <ul style="list-style-type: none"> <li>● &lt;n&gt; = 0: スレーブ主軸の機械軸番号</li> <li>● &lt;n&gt; = 1: マスタ主軸の機械軸番号</li> </ul> <p>機械軸番号は次に準拠: MD20070 \$MC_AXCONF_MACHAX_USED (チャンネルの機械軸)</p> <p>機械軸番号 == 0: 連結は設定されていません。この場合、以降のシステムデータは関連しません。</p> <p><b>注:</b> 主軸同期連結に設定された機械軸は、プログラム命令を使用して変更できません。</p>
MD21320	COUPLE_BLOCK_CHANGE_CTRL_1	<p>同期運転の有効化後のブロック変更のリリース<sup>1)</sup>:</p> <ul style="list-style-type: none"> <li>● 即時に実行する</li> <li>● 「精密同期」へ到達時に実行する</li> <li>● 「汎用同期」へ到達時に実行する</li> <li>● 「同期制御指令値」に到達時に実行する</li> </ul> <p><b>注:</b> 変更保護なし<sup>1)</sup>、ブロック変更リリースは、COUPDEF 命令により変更できます。</p>
MD21310	COUPLING_MODE_1	<p>連結タイプ<sup>1)</sup>:</p> <ul style="list-style-type: none"> <li>● フィードバック値連結</li> <li>● 指令値連結</li> <li>● 速度連結</li> </ul> <p><b>注:</b> 変更保護なし<sup>1)</sup>、連結タイプは、COUPDEF 命令により、無効化された連結に対応して変更できます。</p>
MD21330	COUPLE_RESET_MODE_1	NC スタート、NC ストップ、およびリセットについての主軸同期連結の特性。
MD21340	COUPLE_IS_WRITE_PROT_1	連結の変更保護
1) MD21340 \$MC_COUPLE_IS_WRITE_PROT_1 を参照してください		

## 15.3 設定

表 15-2 セッティングデータ

番号	名称:\$SSC_	機能
SD42300	COUPLE_RATIO_1[<n>]	速度伝達比: FS / LS = 分子/分母 <sup>1)</sup> : <ul style="list-style-type: none"> <li>● &lt;n&gt; = 0: 分子(FS)</li> <li>● &lt;n&gt; = 1: 分母(LS)</li> </ul> 注: 変更保護なし <sup>1)</sup> 、速度伝達比は、COUPDEF 命令により変更できません。
1) MD21340 \$MC_COUPLE_IS_WRITE_PROT_1 を参照してください		

## 15.3.1 NC スタートの主軸同期連結の動作

NC スタート時の主軸同期連結の特性は、次のマシンデータの設定に依存します:

## 主軸同期連結の設定

動作	MD21330 \$MC_COUPLE_RESET_MODE_1
連結の維持	ビット 0 = 0
連結の選択解除	ビット 0 = 1
設定データの有効化	ビット 5 = 1
連結の有効化	ビット 9 = 1

## プログラム指令された主軸同期の連結

動作	MD20112 \$MC_START_MODE_MASK
連結の維持	ビット 10 = 0
連結の選択解除	ビット 10 = 1

## 15.3.2 リセットの主軸同期連結の特性

リセットおよびプログラム終了時の同期運転の特性は、次のマシンデータの設定に依存します。

## 主軸同期連結の設定

動作	MD21330 \$MC_COUPLE_RESET_ MODE_1	MD20110 \$MC_RESET_MODE_MAS K
連結の維持	ビット 1 = 0	ビット 0 = 1
連結の選択解除	ビット 1 = 1	ビット 0 = 1
設定データの有効化	ビット 6 = 1	ビット 0 = 1

## プログラム指令された主軸同期の連結

	MD20110 \$MC_RESET_MODE_MASK
連結の維持	ビット 0 = 1、ビット 10 = 1
連結の選択解除	ビット 0 = 1、ビット 10 = 0

## 15.4 注意点

### 15.4.1 同期モードの一般的な特記事項

#### ダイナミックス制御

指令値連結を使用すると、FS と LS の位置制御パラメータ( $K_v$  係数など)は相互に一致する必要があります。必要に応じて、速度制御と同期モード用に、それぞれのパラメータセットを有効にしてください。連結していない場合と同様に、スレーブ主軸の制御パラメータは、MD30455 \$MA\_MISC\_FUNCTION\_MASK を使用して、位置制御、フィードフォワード制御、およびパラメータセットと異なる設定が可能です(「主軸同期連結の起動に関する注意事項 (ページ 959)」の章を参照してください)。

#### フィードフォワード

コントロールシステムダイナミック応答が改善されたために、同期モードのスレーブ主軸とマスタ主軸のフィードフォワード制御は常に有効です。

しかし、それは軸別 MD32620 \$MA\_FFW\_MODE を使用して FS と LS で選択解除できます。MD32620 \$MA\_FFW\_MODE をゼロに設定すると機能が制限されます。位置制御は、SPCON の移動中に有効にできなくなります。このため、SPOS、M19、SPOSA は

## 15.4 注意点

指令不可になります。NC パートプログラムでは、FFWOF を使用して LS と FS のフィードフォワード制御を無効にすることができません。

フィードフォワード制御モード(速度またはトルクのフィードフォワード制御)は、軸別 MD32620 \$MA\_FFW\_MODE で定義します(「K3:補正 (ページ 301)」の章を参照してください)。

## 速度と加速度の制限

同期モードで動作している主軸の速度制限と加速度制限は、主軸同期ペアで「最も弱い」主軸で決まります。現在のギア選択、プログラム指令加速度、およびマスタ主軸の場合、このために、効率的な位置制御状態(オン/オフ)が考慮されます。

マスタ主軸の最大速度は、スレーブ主軸の速度比と主軸制限を考慮してコントローラ内部で計算されます。

## 複数連結

同期モードを有効にしたとき、連結はすでに FS と LS に対して有効であることをシステムが検出した場合、起動処理は無視されてアラームメッセージが発生します。

複数連結例:

- 主軸が複数の LS の FS として動作している
- カスケード連結(FS は追加連結の LS です)

## チャンネル毎に構成可能な主軸数

チャンネルの全軸を主軸として設定できます。チャンネル毎の軸数は制御タイプによって決まります。

## 相互チャンネル指令連結

- 相互チャンネル主軸同期連結は、DV、AV、VV に別の制限をかけずに実行できます。
- NCU の任意のチャンネルにおいて、主軸総数から 1 つのマスタ主軸を差し引いた数のスレーブ主軸を 1 つのマスタ主軸に連結できます。

## ASUB を使用した同期モードの開始

PLC が AUTOMATIC モードまたは MDI モードで ASUB を起動する場合、同期運転のオンとオフの切り替えまたは終了が可能です。

**参照先:**

『機能マニュアル、基本機能』; 「モードグループ、チャンネル、プログラム運転、リセット応答(K1)」

**アラームへの応答**

同期運転中に発生し、コントローラでアラーム応答「制御イネーブルの取り消し」および「フォローアップモードの起動」として動作するアラームの場合、進行中の制御動作は NC/PLC インタフェース信号による動作と同じです。

- DB31、... DBX2.1 = 0 (コントローライネーブル)
- DB31、... DBX1.4 = 1 (フォローアップモード)

(「同期モードと NC/PLC インタフェース信号 (ページ 948)」の章を参照してください)

以下の NC/PLC インタフェース信号による再原点同期:

DB31、... DBX31.4 = 0 → 1 (再原点同期)

プログラム指令したオフセットを復元する場合(「スレーブ主軸の再同期 (ページ 946)」の章を参照してください)。

**同期運転が有効である場合のブロック検索****注記**

ブロック検索で同期運転が有効である場合は、ブロック検索タイプ 5 の「プログラムテストによるブロック検索」(SERUPRO)のみを使用することを推奨します。

**15.4.2 スレーブ主軸の再同期****位置オフセットの原因**

ドライバイネーブル信号のキャンセル後に連結を再び有効にしたとき、フォローアップモードを有効にすると、マスタ主軸とスレーブ主軸の間で位置オフセットが発生することがあります。位置オフセットの原因は次のとおりです。

- 部品がクランプされたか、両方の主軸を手動で回転した(加工エリアがオープンで、ドライブが電源からオフされている)。
- 主軸イネーブル信号のキャンセル後、2つの主軸が、機械的に連結されていない場合、惰性運転して別々の速度で停止する。

## 15.4 注意点

- ドライブアラームが発生する(内部フォローアップモード)。  
DB31、... DBX61.3 (フォローアップモード有効) = 1  
アラームがクリアされたとき、NC で同期動作を起動しないでください。
- スレーブ主軸の同期禁止のために、同期が実行されない。  
DB31、... DBX29.5 (同期無効化)

## 基本手順

スレーブ主軸とマスタ主軸が同期から外れるか、まったく同期されない場合は、次の方法によってスレーブ主軸とマスタ主軸の間で同期を再起動できます。

1. 軸イネーブル信号を設定し、同期無効信号が設定されている場合はそれをキャンセルします。
2. NC/PLC インタフェース信号を使用し、スレーブ主軸の再同期を開始します。  
DB31、... DBX31.4 (再同期)  
再原点同期処理の完了後に限り、指令値終点同期制御を完全に再起動できます。
3. 連結した主軸が同期されるまで待機します。

## 再同期の有効化

有効信号を設定すると、現在の実位置で連結が完了します。次の2つのNC/PLC インタフェース信号が設定されます。

DB31、... DBX98.1 (汎用同期)

DB31、... DBX98.0 (精密同期)

次の**必要条件**が成立しないと、再同期は動作しません。

- 軸有効信号をスレーブ主軸が設定されている。
- PLC でスレーブ主軸の同期無効が設定されていない。  
DB31、... DBX31.5 (同期無効)

## スレーブ主軸の再同期

次のインタフェース信号の立ち上がりエッジが検出されるとすぐに、当該のスレーブ主軸で再同期が開始されます。

DB31、... DBX31.4 (再同期)

NC は、NC/PLC インタフェース信号を出力によってエッジの検出を確認します。

DB31、... DBX99.4 (同期運転)

インタフェース信号「同期運転」は、次の場合にリセットされます。

- スレーブ主軸の同期が、指令値終点の同期段階まで完了した場合
- IS DB31、... DBX31.4 (再同期)がリセットされた場合

### スレーブ主軸の追加移動中の同期信号の応答

重畳成分が計算され、同期信号が確立されます。

### 例

プログラムコード	コメント
N51 SPOS=0 SPOS[2]=90	
N52 OUPDEF(S2,S1,1,1,「FINE」,「DV」)	
N53 COUPON(S2,S1,77)	
N54 M0	; Offset=77°、「汎用」、「精密」の同期運転信号があります。
N55 SPOS[2]=0 FA[S2]=3600	; 速度差、同期信号「汎用」、「精密」が出力されます。
N56 M0	; (許容範囲を記録して、上記を参照してください) ; Offset=0°、「汎用」、「精密」の同期運転信号があります。
N60 M2=3 S2=500	; 速度差、同期信号「汎用」、「精密」が出力されます。 ; オフセット不確定、同期信号「汎用」、「精密」が出力されます。
N65 M0	; (許容範囲を記録して、上記を参照してください)

### 注記

軸イネーブル信号をキャンセルし、スレーブ主軸の重畳移動を中断できます(SPOS)。IS「NC/PLC インタフェース信号」DB31、... DBX31.4 (再同期)はこの移動成分に対応していませんが、移動成分は REPOS 運転によって再起動されます。

### 必要条件

IS DB31、... DBX31.4(再同期)はスレーブ主軸とマスタ主軸に定義オフセット位置定義オフセット位置がある場合のみ有効です。

これは、連結が完了しているスレーブ主軸の COUPON(...,77) または SPOS、SPOSA、M19 などのオフセット位置をもつ COUPON 指令後の事例です。

## 15.4.3 同期モードと NC/PLC インタフェース信号

## 注記

同期運転時に、リード主軸(LS)またはスレーブ主軸(FS)について、関連するインタフェース信号の連結に対する影響を常に考慮してください。

## 主軸オーバーライド(DB31、... DBB19)

LS の主軸オーバーライド値のみが同期運転で有効です。

## 主軸無効(DB31、... DBX1.3)

LS	FS	連結	動作
0	0	オフ	指令値が出力されます
0	1	オフ	FS の指令値は出力されません
1	0	オフ	LS の指令値は出力されません
1	1	オフ	LS および FS の指令値は出力されません
0	0	ON	指令値が出力されます
0	1	ON	FS に対して主軸の無効化が有効ではありません
1	0	ON	指令値は出力されません。FS に対する主軸の禁止も有効です。
1	1	ON	LS および FS の指令値は出力されません

## コントローライネーブル(DB31、... DBX2.1)

## LS:同期運転中の「コントローライネーブル」のリセット

有効な指令値連結の同期運転中に LS のコントローライネーブルがリセットされた場合、コントローラ内部で現在値連結への切り替えが行われます。LS の移動時にコントローライネーブルがリセットされた場合、LS が停止し、アラームが発行されます。同期運転は、そのまま有効です。

## LS および FS:コントローライネーブルを含まない同期運転の選択

選択した同期運転に対して LS および FS の「コントローライネーブル」が設定されていない場合は、有効化されます。ただし、LS および FS のコントローライネーブルが許可されるまで、LS および FS は移動されません。

**LS および FS: 「コントローライネーブル」 の設定**

「コントローライネーブル」 が設定されている場合に主軸により想定される位置は、DB31、... DBX1.4 == <値>(フォローアップ運転)に依存します:

<値>	想定される主軸位置
0	コントローライネーブルのキャンセル時の位置
1	現在位置

**注記**

FS および LS について、DB31、... DBX2.1 インタフェース信号(コントローライネーブル)を常にまとめて書き込むために、ブロック検索時の主軸同期が推奨されます。これを行わない場合、たとえば LS のコントローライネーブルが未処置ではないので、FS 加工後にブロック検索が停止します。

**注記****同期制御誤差**

連結を事前に解除せずに主軸停止した後で、FS の DB31、...DBX2.1 インタフェース信号(コントローライネーブル)をキャンセルすると、「コントローライネーブル」を再び有効にしたときに、外部干渉に起因する同期制御誤差は補正されません。これによって、FS と LS 間でプログラム指令された角度の関係が失われます。

角度の関係は、再原点同期により復元できます:DB31、... DBX31.4 = 1 (再原点同期)

**フォローアップモード (DB31、... DBX1.4)**

フォローアップ運転では、指令位置が定期的に現在の実位置に設定されます。

DB31、... DBX1.4 == 1 (フォローアップ運転) および DB31、... DBX2.1 == 0 (コントローライネーブル)

⇒ サイクリック:指令位置 = 実位置

**注記**

DB31、... DBX1.4 (フォローアップ運転)は、DB31、... DBX2.1 == 0 の場合にのみ関連します(コントローライネーブル)

### 位置検出器 1/2 (DB31、... DBX1.5、および 1.6)

同期運転中に、FS および LS の位置検出器の切り替えが可能です。連結は維持されます。

---

#### 注記

同期解除運転でのみ、FS および LS の位置検出器を切り替えることを推奨します。

---

### 残移動距離削除/ 主軸リセット(DB31、... DBX2.2)

#### LS:同期運転中の主軸リセットの設定

主軸リセットを設定すると、LS はパラメータ設定された加減速で停止状態まで減速されます。同期運転は、そのまま有効です。

重畳移動は、COUPON/COUPONC と組み合わせた場合を除き、可能な限り迅速に完了します。

### 主軸停止(送り停止) (DB31、... DBX4.3)

#### LS および FS:同期運転中の主軸停止の設定

FS または LS の「主軸停止」を設定すると、両主軸は同期して停止状態まで減速されます。同期運転は、そのまま有効です。

#### 主軸停止のリセット

両主軸で「主軸停止」がリセットされると、最後の有効な速度指令値にまで再加減速が行われます。

#### 適用例

同期運転中に保護ドアが開いたときに、FS および LS を停止します。

LS および FS の信号特性:

1. 停止:DB31、... DBX4.3 = 1 (主軸停止)
2. 停止状態まで待機:DB31、... DBX61.4 == 1
3. 停止中:DB31、... DBX2.1 = 0 (コントローライネーブル)

### S 値の削除(DB31、... DBX16.7)

#### LS:同期運転中の S 値の解除

「S 値の解除」が設定された場合、LS は減速カーブに沿って停止状態まで減速されます。同期運転は、そのまま有効です。

**FS:同期運転中の S 値の解除**

制御インタフェース信号には、同期運転中に FS に対するいかなる機能も含まれません。

**主軸の再同期 1/2 (DB31、... DBX16.4、および 16.5)****LS:同期運転中の位置検出器の再原点同期**

---

**注記**

同期解除運転でのみ、LS の位置検出器を再原点同期することを推奨します。

---

**再原点同期(DB31、... DBX31.4)****FS:プログラム指令した角度オフセットの復元**

FS と LS 間の同期制御が失われたか、または行われていない場合に、プログラム指令された角度オフセットを復元できます。

- 必要条件:DB31、... DBX31.4 = 1 (再原点同期)
- 応答 : DB31、... DBX99.4 == 1 (同期運転)

**ジョグの移動キー(DB31、... DBX4.6、および 4.7)**

ジョグの「プラスとマイナスの移動キー」は、同期運転時の FS の制御で無効になりません。いずれかのキーを押すと、FS は重畳動作を実行します。

---

**注記**

重畳移動を除外する場合は、PLC ユーザープログラムで処置をおこなって重畳移動を禁止してください。

---

**軸と主軸の NC ストップ(DB21、... DBX7.4)**

同期運転時の「軸と主軸の NC ストップ」では、選択したダイナミック応答に従って、連結した主軸が減速します。連結した主軸は、同期モードで引き続き動作します。

## NC スタート(DB21、... DBX7.1)

(「NC スタートの主軸同期連結の動作 (ページ 942)」の章を参照してください)。

## 注記

NC ストップ後の NC スタートでは、同期運転が選択解除されません。

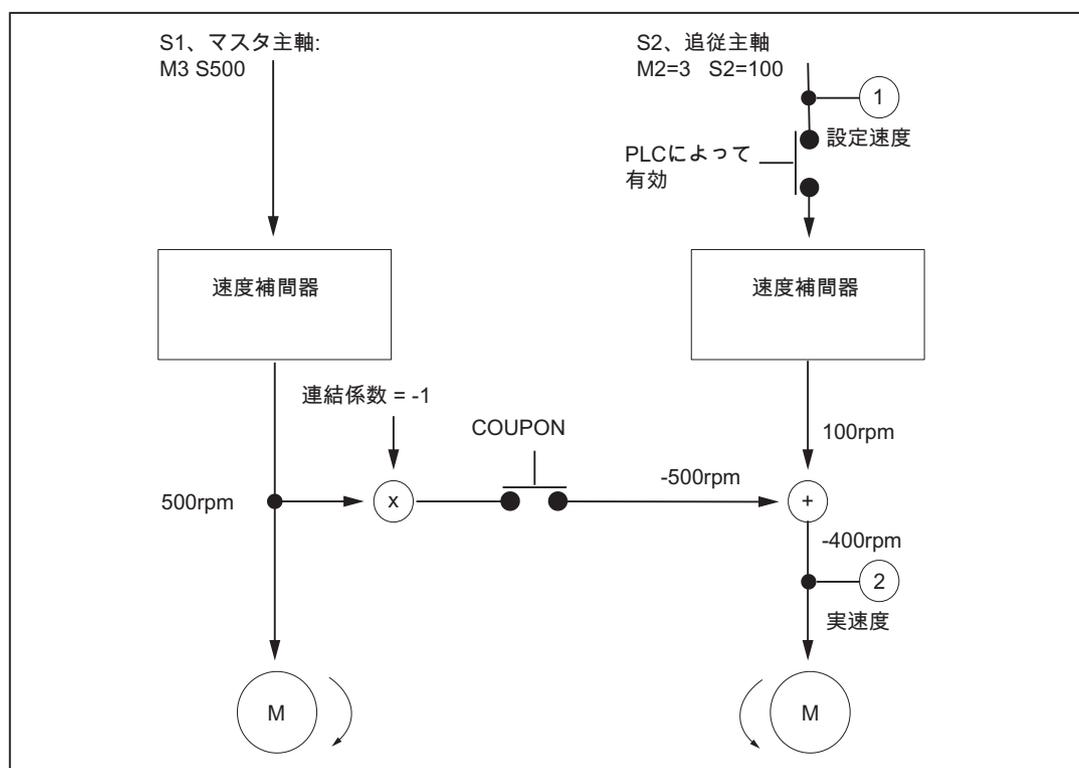
## 15.4.4 マスタ主軸とスレーブ主軸の速度差

## 速度差の発生

旋盤の用途などでは、2つの主軸が対向しているときに速度差が発生します。2つの速度源ソースを符号付きで追加すると、速度成分は連結係数によって、マスタ主軸から得られます。これに加え、スレーブ主軸では速度(S...)および回転方向(M3/M4)をプログラムできます。

一般的に、同期操作をおこなうには、「-1」の値の連結係数を使用します。この符号の反転により、追加のプログラム指令速度と比較してスレーブ主軸の速度差が発生します。

NC に関連する一般的な動作について、次の図で説明します。



## 例

プログラムコード	コメント
N01 M3 S500	; S1 が正の方向に 500 rpm で回転 ; メイン主軸は主軸 1
N02 M2=3 S2=300	; S2 が正の方向に 300 rpm で回転
N05 G4 F1	
N10 COUPDEF(S2,S1,-1)	; 連結係数-1:1
N11 COUPON(S2,S1)	; 連結を有効化.スレーブ主軸 S2 の速度が、マスタ主軸 S1 の速度と連結係数から得られます。
N26 M2=3 S2=100	; 速度差のプログラミング、S2 がスレーブ主軸。

## 用途

マスタ主軸を位置決めして工具を回転させる生産加工運転では、スレーブ主軸として機能する対向主軸との間で、正確な同期が必要です。スレーブ主軸を中心にして回転するタレットでは、様々な工具タイプで部品を加工できます。

次の図は、主軸と工具が平行に配置された用途を示しています。

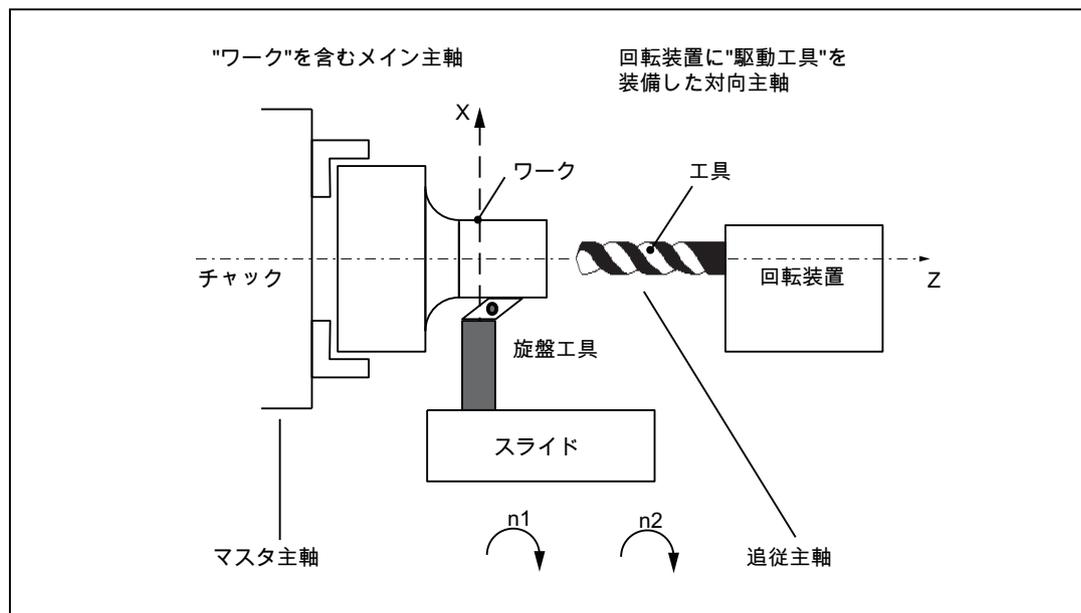


図 15-4 Z 軸周りのリボルバーを備えたシングル滑り旋盤

## 必要条件

速度差プログラミングの基本的な必要条件:

- 主軸同期機能が必要です。
- スレーブ主軸のダイナミック応答は、最低でもマスタ主軸のダイナミック応答と同じ水準にしてください。同水準でない場合、フローティングチャック G331/G332 無しのリジッドタッピングで、システム品質が低下することがあります。
- スレーブ主軸が設定されたチャンネルでも、速度差をプログラム指令してください。マスタ主軸は別のチャンネルでプログラム指令できます。
- PLC の IS 「重畳移動有効」(DB31、... DBX26.4)によって、スレーブ主軸の速度差を有効にしてください。有効信号を設定していない場合は、アラーム 16771 「チャンネル% 追従軸% 重畳移動が無効です」が出力されます。IS 「重畳移動有効」(DB31、... DBX26.4)を設定するか、連結を解除すると、このアラームはクリアされます。

---

### 注記

SPOS、SPOSA、M19 などの位置決めモーションおよび軸の動きはを有効化する必要はありません。

---

### 注記

このため、速度差は連結処理に影響しません。

連結が有効である間、スレーブ主軸やマスタ主軸ではギア選択を変更できません。

---

## COUPONC による連結の起動

連結を有効にすると、連結係数の適用によって、前と同じように、マスタ主軸の速度までスレーブ主軸が加速されます。連結を有効にしたときに、スレーブ主軸がすでに回転している場合(M3、M4)、連結後にこの動作がおこなわれます。

## 連結の解除

連結を解除すると、スレーブ主軸は、両方の速度成分の合計速度で回転し続けます。主軸は、別の主軸から伝送された速度と方向でプログラム指令されたように、動作します。解除するとき、以前の動作との違いはありません。

## 速度差

速度差は、スレーブ主軸の**指令更新**プログラミング(例の S2=... )、または主軸同期連結が動作中の速度制御モードの M2=3、M2=4、または COUPONC による、スレーブ主軸速度の適用によって発生します。

**条件:**

速度 S...は、回転 M3 または M4 の方向とともに再プログラム指令してください。このようにしないと、アラーム 16111 「チャンネル% ブロック% 主軸% プログラム指令速度がありません」が表示されます。

**スレーブ主軸のオフセットの読み取り**

速度差をプログラム指令すると、現在のオフセットは常時変化します。現在のオフセットは、指令値終了時に\$AA\_COUP\_OFFS[Sn]によって、フィードバック値終点で\$VA\_COUP\_OFFS[Sn]によって読み取ることができます。

変数\$P\_COUP\_OFFS[Sn]は、プログラム指令された最後のオフセットを返します。

**速度差の表示**

プログラム指令された速度差成分は、プログラム指令速度差の速度指令値として表示されます(この例の場合は 100 rpm です)。

実速度はモータ速度を示します。この例の実速度は  $500 \text{ rev/min} * (-1) + 100 \text{ rpm} = -400 \text{ rev/min}$  です。

**IS、NC から PLC へ****速度制御運転のスレーブ主軸**

プログラム指令した速度差(上の例の場合、N26 の M2=3 S2=100)に達すると、NC はスレーブ主軸に IS 「主軸は指令範囲内」(DB31、... DBX83.5)を設定します。速度差をプログラム指令しても、PLC で有効になっていない場合、この VDI インタフェース信号は設定されません。

速度差をプログラム指令しても、連結で速度差が要求される場合、スレーブ主軸は位置制御状態のままです。

**注記**

速度差プログラミングによって、指令値連結に加えて、別の指令値が作成されるときに、軸の VDI インタフェース信号 NC → PLC IS 「重畳動作」が設定されます(DB31、... DBX98.4)。

**右回りの実回転方向(DB31、... DBX83.7)**

IS 「右回りの実回転方向」(DB31、... DBX83.7)は、モータ方向を示します。

## IS、PLC から NC へ

### PLC インタフェースによるスレーブ主軸への作用

速度制御モードで、軸の VDI インタフェース信号が、速度差をもつスレーブ主軸に与える作用について説明します。

### 残移動距離削除/主軸リセット(DB31、... DBX2.2)

プログラム指令された速度差と方向は、IS「残移動距離削除/主軸リセット」(DB31、... DBX2.2)によって終了できます。プログラム指令速度のみを削除するには、IS「S 値を削除」(DB31、... DBX16.7)を設定できます。

### 主軸の再同期 1/2 (DB31、... DBX16.4、および 16.5)

この IS「主軸の再同期 1/2」(DB31、... DBX16.4/16.5)は禁止されません。連結しても、位置オフセットが自動的に補正されることはありません。

### 反転 M3/M4 (DB31、... DBX17.6)

IS「反転 M3/M4」(DB31、... DBX17.6)は、スレーブ主軸用に追加でプログラム指令された速度成分の反転のみをおこないます。

主軸同期連結によって生成される移動成分には影響しません。

### 主軸オーバーライド(DB31、... DBB19)

「主軸オーバーライド」VDI インタフェース(DB31、... DBB19)は、スレーブ主軸用に追加でプログラム指令された速度成分のみに影響します。主軸オーバーライドスイッチがすべての軸の入力に伝送されると、主軸オーバーライド値の変化はスレーブ主軸に 2 回適用されます。

- マスタ主軸の速度変化によって間接的に 1 回適用および
- スレーブ主軸のプログラム指令成分で 1 回適用

オフセット値は PLC プログラムで、上記に従って調整できます。

### 連結の選択解除

連結を解除すると、スレーブ主軸は、両方の速度成分の合計速度で回転し続けます。連結選択解除時の動作切り替えは、連続運転速度でおこなわれます。

COUPOF を使用すると、主軸は、別の主軸から伝送された速度と方向でプログラム指令されたように動作します。この例の場合は **M4 S400** です。

COUPOFS をプログラム指令すると、スレーブ主軸は現在の速度から減速して停止します。

### 追加機能の有効化

スレーブ主軸もメイン主軸にすることができます。この場合は追加の機能が可能になります。

- **G95、G96、G97** による毎回転送り速度。**G96 S2=...**では、「周速一定制御」をスレーブ主軸用に有効にすることができます。  
径方向軸の位置によって決まる速度は、主軸 2 の速度補間器の指令速度であるため、**S2** の合計速度に含まれます。
- **G331、G332** のフローティングチャックなしのリジッドタッピング。

## 15.4.5 同期補正中の同期信号の動作

### 同期補正の動作

フィードバック値と補正指令値を比較して、新しい同期信号が生成されます。補正処理が実行されると、再度同期信号が出力されます。

## 15.4.6 同期補正削除と NC リセット

変数 **\$AA\_COUP\_CORR[Sn]** は、それぞれの同期補正の削除状態で、値ゼロを返します。

- **COUPON(..)/COUPONC(..)** によって、当該の追従主軸の主軸同期連結を有効にすると、既存の同期補正は指令位置で適用されます。
- パートプログラムの終了時に有効でなく、**NC** リセット中に有効な同期補正は、指令位置で適用されます。これは同期信号に影響しません。
- **M30** では、既存の同期補正が保持されます。
- ユーザー側では、変数 **\$AA\_COUP\_CORR** に**値ゼロ**を入力して早い段階で修正値を削除することもできます。同期補正はすぐに削除されますが、大きい値が含まれる場合は、加減速でい減レートの減速カーブを使用して削除されます。

### 15.4.7 主軸同期連結の起動に関する注意事項

#### 主軸の起動

マスタ主軸とスレーブ主軸を通常の主軸と同じように最初に起動してください。当該の手順については、次を参照してください。

参照先:

『CNC 試運転マニュアル』:NC、PLC、ドライブ『機能マニュアル、基本機能』;「主軸(S1)」

#### 必要条件

主軸同期ペアに次のパラメータが設定されていること。

- マスタ主軸とスレーブ主軸の機械番号  
(チャンネル別マシンデータ MD21300 \$MC\_COUPLE\_AXIS\_1[n]で固定連結設定の場合)
- 必要な連結モード(指令値、フィードバック値、速度のいずれかの連結)  
(チャンネル別マシンデータ MD21310 \$MC\_COUPLING\_MODE\_1[n]で固定連結設定の場合)
- 同期運転の FS と LS のギア選択
- 固定設定する主軸同期連結には、次の連結機能も適用できます。
  - 主軸同期運転のブロック切り替え動作:  
MD21320 \$MC\_COUPLE\_BLOCK\_CHANGE\_CTRL\_1
  - 連結キャンセル動作:  
MD21330 \$MC\_COUPLE\_RESET\_MODE\_1
  - 連結パラメータの書き込み保護:  
MD21340 \$MC\_COUPLE\_IS\_WRITE\_PROT\_1
  - 主軸同期連結の座標変換パラメータ:  
SD42300 \$SC\_COUPLE\_RATIO\_1[n]

#### 指令値連結の FS と LS の命令応答

指令値連結で最適に同期するために、FS と LS のダイナミック応答を指令値変更に対する応答と同じにしてください。軸の制御ループ(位置、速度と電流コントローラ)はそれぞれ最適な値に設定し、できるだけ早く効率的に変動を抑えてください。

制御の質を低下させずに、軸毎の異なるダイナミック応答を調整するため、指令回路にダイナミック応答調整機能が設けられています(「K3:補正 (ページ 301)」の章を参照してください)。次の制御パラメータをそれぞれ FS と LS で最適に設定してください。

- $K_v$  係数(MD32200 \$MA\_POSCTRL\_GAIN)
- フィードフォワード制御パラメータ
  - MD32620 \$MA\_FFW\_MODE
  - MD32610 \$MA\_VELO\_FFW\_WEIGHT
  - MD32650 \$MA\_AX\_INERTIA
  - MD32800 \$MA\_EQUIV\_CURRCTRL\_TIME
  - MD32810 \$MA\_EQUIV\_SPEEDCTRL\_TIME

同期ずれ中の動作:

- 軸別 MD32620 \$MA\_FFW\_MODE

Tt バランス MD32620 = 3 で速度フィードフォワード制御に対して、スレーブ軸のフィードフォワード制御モードを設定することを推奨します。

バランス処理をさらに安全にするため、軸別マシンデータ変更によって、このフィードフォワード制御モードをさらに最適化できます。

マシンデータ	意味
MD32810 EQUIV_SPEEDCTRL_TIME	フィードフォワード制御の速度制御ループ等価時定数
MD37200 COUPLE_POS_TOL_COURSE	「汎用同期」のしきい値
MD37210 COUPLE_POS_TOL_FINE	「精密同期」のしきい値
MD37220 COUPLE_VELO_TOL_COURSE	速度許容範囲「汎用」
MD37220 COUPLE_VELO_TOL_FINE	速度許容範囲「精密」

このような場合、同期信号のしきい値を高く、位置および(または)速度の許容範囲を大きくすると、結果はさらに安定します。

## ダイナミック応答調整

適切な制御特性を得るために、FS および LS のダイナミック応答を同じにしてください。FS および LS の追従誤差は、任意の指定の速度で一致するようにしてください。ダイナミックに異なる主軸では、指令分岐でダイナミック応答調整による調整を実現できます。ダイナミック特性が「最も弱い」主軸と、関連する他の主軸間の等価時定数の差は、ダイナミック応答調整の時定数として入力してください。

### 例

速度フィードフォワード制御が有効であるときは、主に「最も遅い」速度制御ループ等価時定数によってダイナミック応答が決まります。

- 等価時定数 LS:MD32810 \$MA\_EQUIV\_SPEEDCTRL\_TIME[<LS>] = 5 ms
- 等価時定数 FS:MD32810 \$MA\_EQUIV\_SPEEDCTRL\_TIME[<FS>] = 3 ms
- FS のダイナミック応答調整の時定数:MD32910 \$MA\_DYN\_MATCH\_TIME[<FS>] = 5 ms - 3 ms = 2 ms
- FS のダイナミック応答調整の起動:MD32900 \$MA\_DYN\_MATCH\_ENABLE[<FS>] = 1

正しく指令されたダイナミック応答調整では、FS および LS の追従誤差は一致します:操作エリア[診断|サービス軸]

最適化のために、サーボゲイン<sub>係</sub>数またはフィードフォワード制御パラメータをわずかに調整することが必要になる場合があります。

## 位置制御パラメータセット

主軸の場合、ギヤ選択毎に位置制御パラメータセットが割り当てられます。これらのパラメータセットは、同期運転での LS と FS のダイナミック応答を調整するために使用できます。このためには、速度および位置決め運転、同期運転で異なるギヤ選択を使用してください。関連する運転モードを有効にする前に、関連するギヤ選択を切り替える必要があります。

### 制御パラメータ

次の制御パラメータは、FS と LS で同一に設定してください。

- MD33000 \$MA\_FIPO\_TYPE (ファインインタポレーションのタイプ)
- MD32400 \$MA\_AX\_JERK\_ENABLE (軸の加々速度制限)
- MD32402 \$MA\_AX\_JERK\_MODE (軸の加々速度制限用フィルタの種類)
- MD32410 \$MA\_AX\_JERK\_TIME (軸のジャークフィルタの時定数)

- MD32412 \$MA\_AX\_JERK\_FREQ (軸の加々速度フィルタのブロック周波数)
- MD32414 \$MA\_AX\_JERK\_DAMP (軸の加々速度フィルタの減衰)
- MD32420 \$MA\_JOG\_AND\_POS\_JERK\_ENABLE (リリース加々速度制限)
- MD32430 \$MA\_JOG\_AND\_POS\_MAX\_JERK (軸加々速度)

#### FS:制御パラメータの自動パラメータ設定

スレーブ主軸の制御パラメータは、このマシンデータを使用して次のように設定できます。

#### MD30455 \$MA\_MISC\_FUNCTION\_MASK

ビット 5=0:主軸同期連結、スレーブ主軸:

位置制御、フィードフォワード制御、パラメータセットは、スレーブ主軸用に設定されます。

ビット 5=1:主軸同期連結:

スレーブ主軸の制御パラメータは、連結していない場合と同じように設定されます。

#### 参照先:

『機能マニュアル、基本機能』; 「速度、指令値/フィードバック回路、閉ループ制御(G2)」

### 同期運転の LS パラメータの自動伝送

FS および LS の自動ダイナミック応答調整では、同期運転のために、FS の位置制御、フィードフォワード制御、およびパラメータセットのパラメータ値を LS から伝送できます:

MD30455 \$MA\_MISC\_FUNCTION\_MASK、ビット 5

### 主軸と軸の運転の個別のダイナミック応答

主軸と軸の運転では、ダイナミックプログラミング FA、OVRA、ACC、VELOLIMA を次の MD でそれぞれに個別に設定できます。

#### MD30455 \$MA\_MISK\_FUNCTION\_MASK ビット 6=0

プログラム指令軸または主軸の名称によって割り当てを行います。たとえば主軸運転では VELOLIMA[S1]=50 によって最大速度を 50%まで下げて、軸運転では VELOLIMA[C]=50 によって最大速度を 50%に下げます。

たとえば VELOLIMA[S1]=50 と VELOLIMA[C]=50 が、次のマシンデータで以前と同じ動作の場合、FA、OVRA、ACC、VELOLIM のプログラミングは、プログラム指令名称と関係なく動作します。

**MD30455 \$MA\_MISK\_FUNCTION\_MASK ビット 6=1****膝形加速特性**

マスタ主軸の場合、膝形加速特性がスレーブ主軸に与える作用は、次の軸別マシンデータで識別されます。

- MD35220 \$MA\_ACCEL\_REDUCTION\_SPEED\_POINT (加速度てい減の速度)
- MD35230 \$MA\_ACCEL\_REDUCTION\_FACTOR (加速度てい減)

MD35242 \$MA\_ACCEL\_REDUCTION\_TYPE が存在する場合は、加速度てい減タイプの設定にも使用できます。存在しない場合は、加速度の急激な低下が想定されます。

連結係数を考慮するとき、スレーブ主軸のダイナミック応答がマスタ主軸のダイナミック応答より小さい場合、マスタ主軸のダイナミック応答は、連結が有効である間に必要なレベルまで下げられます。

加速度は、スレーブ主軸の速度範囲全体で一定である必要があります。ただし、スレーブ主軸の上記のマシンデータにも膝形加速特性が格納されている場合、これは主軸が連結されている場合のみ考慮されます。指定された膝形加速特性のスレーブ主軸のセットポイントが適用されます。

**参照先:**

『機能マニュアル、基本機能』；「加速度制御(B2)」  
:膝形加速特性の章

**フィードバック値連結**

フィードバック値連結(AV)では、FS のドライブはマスタ主軸ドライブよりさらにダイナミックになります。フィードバック値連結の個々のドライブも、ダイナミック応答に従って最適に設定されます。

フィードバック値連結は例外的な場合に限って使用してください。

**速度連結**

速度連結(VV)は指令値連結(DV)に内部的には同じですが、FS と LS のダイナミックな必要条件はより低くなります。FS および(または)LS に位置制御サーボループは必要ありません。検出器は必要ありません。

## 汎用/精密同期のしきい値

速度コントローラオートチューニングとフィードフォワード制御の設定後、汎用同期と精密同期のしきい値を **FS** に入力してください。

- 「汎用同期」のしきい値  
軸別 MD7200;AV、DV:COUPLE\_POS\_TOL\_COARSE  
MD37220;VV:COUPLE\_VELO\_TOL\_COARSE
- 「精密同期」のしきい値  
軸別 MD37210;AV、DV:COUPLE\_POS\_TOL\_FINE  
MD37230;VV:COUPLE\_VELO\_TOL\_FINE

工作機械メーカーの精度要求に従って **FS** の値を計算し、サービス表示によって **PLC** インタフェースをチェックしてください。

## 角度オフセット **LS/FS**

同期運転が有効であるときなど、**FS** と **LS** の間に角度オフセットの定義が必要な場合は、**FS** と **LS** の「原点位置角度」を相互に調整してください。これは次のマシンデータで実行できます。

- MD34100 \$MA\_REFP\_SET\_POS
- MD34080 \$MA\_REFP\_MOVE\_DIST
- MD34090 \$MA\_REFP\_MOVE\_DIST\_CORR

参照先:

『機能マニュアル、基本機能』; 「レファレンス点復帰(R1)」

## **FS** のサービス表示

「アラーム診断」操作エリアで、同期モードでのセットアップ時に、スレーブ主軸に対して次の値が表示されます。

- **FS** と **LS** の指令値間の実偏差  
表示値:マスタ主軸関連の位置オフセット(指令値)  
(パートプログラムの軸変数\$AA\_COUP\_OFFS で読み取り可能な、**FS** と **LS** 間の角度オフセットに対応する値)
- **FS** と **LS** のフィードバック値間の実偏差  
表示値:マスタ主軸関連の位置オフセット(フィードバック値)

参照先:

『操作マニュアル』

## 15.5 境界条件

### 「主軸同期」機能の適用

この機能は、オプションです(「主軸同期/ポリゴン加工」または汎用連結の対応するオプションタイプ)。ライセンス管理によってハードウェアに割り当ててください。

#### 注記

様々なタイプの汎用連結については、下記を参照してください。

#### 参照先:

『機能マニュアル、応用機能』、「連結軸(M3)」

## 15.6 例

### プログラミング例

プログラムコード	コメント
	; マスタ主軸=メイン主軸=主軸 1
	; スレーブ主軸=主軸 2
N05 M3 S3000 M2=4 S2=500	; メイン主軸は 3000 rpm で回転 ; FS:500rpm
N10 COUPDEF (S2, S1, 1, 1, 「No」, 「Dv」)	; 連結の定義、これは ; 設定も可能
N70 SPCON	; マスタ主軸を位置制御に含める ; (指令値連結)
N75 SPCON(2)	; スレーブ主軸を閉ループ位置制御にします
N80 COUPON (S2, S1, 45)	; オフセット位置への高速連結= 45°
N200 FA [S2] = 100	; 位置決め速度= 100deg/min
N205 SPOS[2] = IC(-90)	; 負方向に 90°重畳して移動
N210 WAITC(S2、「Fine」)	; 「精密」同期を待機
N212 G1 X... Y... F...	; 加工タイプ
N215 SPOS[2] = IC(180)	; 正方向に 180°重畳して移動
N220 G4 S50	; メイン主軸の ; ドウエル時間 = 50 回転
N225 FA [S2] = 0	; 設定速度を有効にします(マシンデータ)
N230 SPOS[2] = IC (-7200)	; 負方向に ; 設定速度で 20 回転
N350 COUPOF (S2, S1)	; 高速連結解除、S = S2 = 3000
N355 SPOSA[2] = 0	; 0°で FS を停止します

プログラムコード	コメント
N360 G0 X0 Y0	
N365 WAITS (2)	; 主軸 2 を待機
N370 M5	; FS を停止
N375 M30	

## 15.7 データリスト

### 15.7.1 マシンデータ

#### 15.7.1.1 NC 別マシンデータ

番号	識別子: \$MN_	説明
10000	AXCONF_MACHAX_NAME_TAB	機械軸名称

#### 15.7.1.2 チャンネル別マシンデータ

番号	識別子: \$MC_	説明
20070	AXCONF_MACHAX_USED	チャンネルで有効な機械軸番号
21300	COUPLE_AXIS_1	主軸同期ペアの定義
21310	COUPLING_MODE_1	主軸同期モードの連結のタイプ
21320	COUPLE_BLOCK_CHANGE_CTRL_1	主軸同期運転のブロック切り替え動作
21330	COUPLE_RESET_MODE_1	連結中止動作
21340	COUPLE_IS_WRITE_PROT_1	連結パラメータの書き込み保護

#### 15.7.1.3 軸/主軸別マシンデータ

番号	識別子: \$MA_	説明
30455	MISK_FUNCTION_MASK	軸機能
30550	AXCONF_ASSIGN_MASTER_CHAN	軸入れ替え用のチャンネルのリセット設定
32200	POSCTRL_GAIN	サーボゲイン係数( $K_v$ 係数)

## 15.7 データリスト

番号	識別子: \$MA_	説明
32400	AX_JERK_ENABLE	軸加々速度制限
32410	AX_JERK_TIME	軸加々速度フィルタの時定数
32420	JOG_AND_POS_JERK_ENABLE	軸の加々速度制限の初期設定
32430	JOG_AND_POS_MAX_JERK	軸加々速度
32610	VELO_FFW_WEIGHT	速度フィードフォワード制御のフィードフォワード制御係数
32620	FFW_MODE	フィードフォワード制御モード
32650	AX_INERTIA	トルクフィードフォワード制御の慣性モーメント
32800	EQUIV_CURRCTRL_TIME	フィードフォワード制御のための電流制御ループ等価時定数
32810	EQUIV_SPEEDCTRL_TIME	フィードフォワード制御の速度制御ループ等価時定数
34080	REFP_MOVE_DIST	レファレンス点復帰距離
34090	REFP_MOVE_DIST_CORR	基準点オフセット
34100	REFP_SET_POS	レファレンス点位置
35000	SPIND_ASSIGN_TO_MACHAX	機械軸への主軸の割り当て
37200	COUPLE_POS_TOL_COARSE	「汎用同期」のしきい値
37210	COUPLE_POS_TOL_FINE	「精密同期」のしきい値
37220	COUPLE_VELO_TOL_COARSE	リード主軸と追従主軸の間の「汎用」速度許容差
37230	COUPLE_VELO_TOL_FINE	リード主軸と追従主軸の間の「精密」速度許容差

## 15.7.2 セットアップデータ

## 15.7.2.1 チャンネル別セットアップデータ

番号	識別子: \$SC_	説明
42300	COUPLE_RATIO_1	主軸同期運転の速度比パラメータ

## 15.7.3 信号

### 15.7.3.1 チャンネルへの信号

信号名称	SINUMERIK 840D sl	SINUMERIK 828D
NC スタート	DB21, ... .DBX7.1	DB320x.DBX7.1
NC ストップ、軸と主軸	DB21, ... .DBX7.4	DB320x.DBX7.4

### 15.7.3.2 チャンネルからの信号

信号名称	SINUMERIK 840D sl	SINUMERIK 828D
ドライラン送り速度選択	DB21, ... .DBX24.6	DB170x.DBX0.6
早送り速度オーバーライド選択	DB21, ... .DBX25.3	DB170x.DBX1.3

### 15.7.3.3 軸/主軸への信号

信号名称	SINUMERIK 840D sl	SINUMERIK 828D
軸/主軸無効	DB31, ... .DBX1.3	DB380x.DBX1.3
フォローアップモード	DB31, ... .DBX1.4	DB380x.DBX1.4
位置検出器 1、位置検出器 2	DB31, ... .DBX1.5/6	DB380x.DBX1.5/6
コントローライネーブル	DB31, ... .DBX2.1	DB380x.DBX2.1
残移動距離/主軸リセット	DB31, ... .DBX2.2	DB380x.DBX2.2
主軸停止/送り停止	DB31, ... .DBX4.3	DB380x.DBX4.3
JOG の移動キー	DB31, ... .DBX4.6/7	DB380x.DBX4.6/7
主軸再同期 1、主軸再同期 2	DB31, ... .DBX16.4/5	DB380x.DBX2000.4/5
S 値を削除	DB31, ... .DBX16.7	DB380x.DBX2000.7
送り速度オーバーライド有効	DB31, ... .DBX17.0	DB380x.DBX2001.0
M3/M4 反転	DB31, ... .DBX17.6	DB380x.DBX2001.6
主軸オーバーライド	DB31, ... .DBB19	DB380x.DBB2003
再同期	DB31, ... .DBX31.4	-
同期の無効化	DB31, ... .DBX31.5	-

## 15.7.3.4 軸/主軸からの信号

信号名称	SINUMERIK 840D sl	SINUMERIK 828D
原点確立済み/原点同期済み 1、原点確立済み/原点同期済み 2	DB31, ... .DBX60.4/5	DB390x.DBX0.4/5
同期モード	DB31, ... .DBX84.4	DB390x.DBX2002.4
精密同期	DB31, ... .DBX98.0	DB390x.DBX5002.0
汎用同期	DB31, ... .DBX98.1	DB390x.DBX5002.1
フィードバック値連結	DB31, ... .DBX98.2	DB390x.DBX5002.2
重畳動作	DB31, ... .DBX98.4	DB390x.DBX5002.4
マスタ主軸 LS/LA が有効	DB31, ... .DBX99.0	DB390x.DBX5003.0
スレーブ主軸 FS/FA が有効	DB31, ... .DBX99.1	DB390x.DBX5003.1

## 15.7.4 システム変数

システム変数	説明
\$P_COUP_OFFS[スレーブ主軸]	主軸同期のプログラム指令オフセット
\$AA_COUP_OFFS[スレーブ主軸]	主軸同期の位置オフセット(指令値)
\$VA_COUP_OFFS[スレーブ主軸]	主軸同期の位置オフセット(フィードバック値)

システム変数の詳細については、以下を参照してください。

**参照先:**

リストマニュアル『システム変数』

## S7:メモリ構成

### 16.1 はじめに

#### メモリ領域

NC のローカルの保持型および非保持型データを保存/管理するため、NCU の CF カードは 2 つのメモリ領域を備えます。

- **スタティック NC メモリ**  
スタティック NC メモリには、アクティブファイルシステムとパッシブファイルシステム (ページ 972) の **保持型 NC** データが含まれます。
- **ダイナミック NC メモリ**  
ダイナミック NC メモリには、ダイナミックに NC がダイナミックに生成したマクロ、ローカルユーザーデータ、バッファメモリなど、**非保持型 NC** データが含まれます。

#### 確実に厳格な動作

すべての加工状況で NC の特性が予め決められているようにするため、ローカルのスタティックおよびダイナミック NC メモリのすべてのメモリ領域は設定可能ではあるものの恒久的な容量が定められています。

#### メモリ構成

ローカルのスタティックおよびダイナミック NC メモリのメモリ領域は、メモリを設定するマシンデータの標準設定に基づいてコントローラが最初に起動されると設定されます。これらの設定は、ほとんどの場合、妥当です。

#### 再構成;

必要に応じて、ユーザー独自の必要条件に合わせてメモリ構成を設定できます。メモリの必要な再構成によって、スタティック NC メモリのユーザーデータが失われる場合があります。よって、メモリ構成マシンデータへの変更後、メッセージ 4400:「MD を変更すると記憶保持メモリの再解析がおこなわれます(データ消失!)」が表示されます。マシンデータの変更を有効化する前、または NC リセットによるメモリの再構成をおこなう前に、セットアップアーカイブを作成してください。

メモリ構成は、次回 NC を起動すると変更されます。その後、作成したセットアップアーカイブを読み込み、ユーザーデータを復元できます。

## 16.2 アクティブファイルシステムとパッシブファイルシステム

NC のユーザー独自のデータは、ローカルのスタティック NC メモリに格納されています。このメモリ領域には、アクティブおよびパッシブファイルシステムのデータが含まれます。

### アクティブファイルシステム

アクティブファイルシステムには、NC をパラメータ設定するためのシステムデータが含まれます。基本的に、以下が含まれます。

- マシンデータ
- セッティングデータ
- オプションデータ
- グローバルユーザーデータ(GUD)
- 工具オフセット/マガジンデータ
- プロテクションゾーン
- R 変数
- ワークオフセット/FRAME
- 真直度補正
- 象限突起補償
- ピッチ誤差補正(Leadscrew error compensation)

アクティブファイルシステムのデータは、NC の現在の作業データです。

アクティブファイルシステムの表示はデータに対応しています。

### パッシブファイルシステム

パッシブファイルシステムには、NC にロードされたすべてのローカルファイルが含まれます。

- メインプログラム
- サブプログラム
- ワーク
- グローバルユーザーデータとマクロ定義ファイル(\*.DEF)
- 標準サイクル
- ユーザーサイクル

- コメント
  - バイナリファイル(例:ピクチャ、PDF 文書)
- パッシブファイルシステムの表示はファイルに対応しています。

## 16.3 セットアップ

### 16.3.1 設定

ローカルのスタティックおよびダイナミック NC メモリの構成は、以下のマシンデータで設定され、影響を受けます。

- メモリを構成するマシンデータ:
  - \$MN\_MM...(NC 別、メモリ構成マシンデータ)
  - \$MC\_MM... (チャンネル別メモリ構成マシンデータ)
  - \$MA\_MM... (軸別メモリ構成マシンデータ)
- パラメータ設定されたチャンネル数:
  - MD10010 \$MN\_ASSIGN\_CHAN\_TO\_MODE\_GROUP(モードグループで有効なチャンネル)

### 16.3.2 再構成

コントローラを最初に起動すると、メモリ構成マシンデータのデフォルト設定に基づいてローカル NC メモリのメモリ領域の容量が設定されます。一般的に、これらの設定はコントローラの操作に十分です。特別なユーザー独自の必要条件のためローカル NC メモリのメモリ領域を設定した場合、ローカル NC メモリが再構成されます。スタティック NC メモリのアクティブやパッシブファイルシステムが影響された場合、メッセージ 4400 が表示されます。その場合、セットアップアーカイブを作成することで関連するデータを外部記憶媒体にバックアップします。

メッセージ 4400 は、以下の場合に表示されます。

- パッシブファイルシステムのメモリ構成マシンデータが変更された場合
- アクティブファイルシステムのメモリ構成マシンデータが変更され、「自動メモリ再設定」(AMR)機能が有効でない場合
- アクティブファイルシステムのメモリ構成マシンデータが変更され、AMR 機能が有効であるが、アクティブファイルシステムのデータをローカルでバッファできない場合。

#### 通知

##### 再構成によるデータ消失

ローカルのスタティック NC メモリを再構成すると、アクティブやパッシブファイルシステムのユーザーデータが消失する可能性があります。よって、メッセージ 4400 が表示された場合、NC リセットによって変更されたメモリ構成を有効化する前に、セットアップアーカイブを作成することで、すべての関連するデータをバックアップすることを強く推奨します。

### 「自動メモリ再構成」機能(AMR)

#### アクティブファイルシステム

The AMR 機能によって、ユーザーデータの消失を防ぐため、セットアップアーカイブを作成した後リードバックする必要なく、アクティブファイルシステム (ページ 970) のメモリ領域を再構成することができます。

機能が有効であり、アクティブファイルシステムに影響するメモリ構成マシンデータを修正した場合は、まず、アクティブファイルシステムのすべてのデータをローカルでバッファできるかがチェックされます。可能な場合、アクティブファイルシステムのデータは、ウォーム再起動のトリガによって、修正したメモリ構成を有効化すると、最初にローカルにバッファされます。その後、アクティブファイルシステムのメモリが再構成されます。最後に、バッファされたデータが、新しく構成されたアクティブファイルシステムに再度読み込まれます。

アクティブファイルシステムのデータをバッファできない場合、メッセージ 4400 が表示されます。その場合、マシンデータ変更を有効化する前に、関連するユーザーデータを含むセットアップアーカイブを作成します。以下の「参照先」の節を参照してください。

#### パッシブファイルシステム

パッシブファイルシステムの再構成を必要とするマシンデータが変更された場合、AMR 機能とは関係なく必ずメッセージ 4400 が表示されます。その場合、マシンデータ変更

を有効化する前に、関連するユーザーデータを含むセットアップアーカイブを作成します。

アクティブファイルシステムの再構成のみを必要とするマシンデータが変更された場合、パッシブファイルシステムのデータは保持されます。

#### 起動

「自動メモリ再構成」機能(AMR)は、次のマシンデータで有効化されます。

```
MD17950 $MN_IS_AUTOMATIC_MEM_RECONFIG = TRUE
```

### 参照先

セットアップアーカイブの作成

試運転マニュアル:CNC 試運転:NC、PLC、ドライブ; 「データの保存と管理」章 > 「セットアップアーカイブの作成」

## 16.4 スタティックユーザーメモリの構成

### 16.4.1 スタティック NC メモリの区分

スタティック NC メモリは、システムとユーザーが共同で使用します。ユーザーが使用できる領域はスタティックユーザーメモリとして定義されます。これには、アクティブファイルシステムとパッシブファイルシステムからのデータが含まれます。

下図は、SINUMERIK 840D sl のスタティック NC メモリの基本的な区分を示しています。

16.4 スタティックユーザーメモリの構成

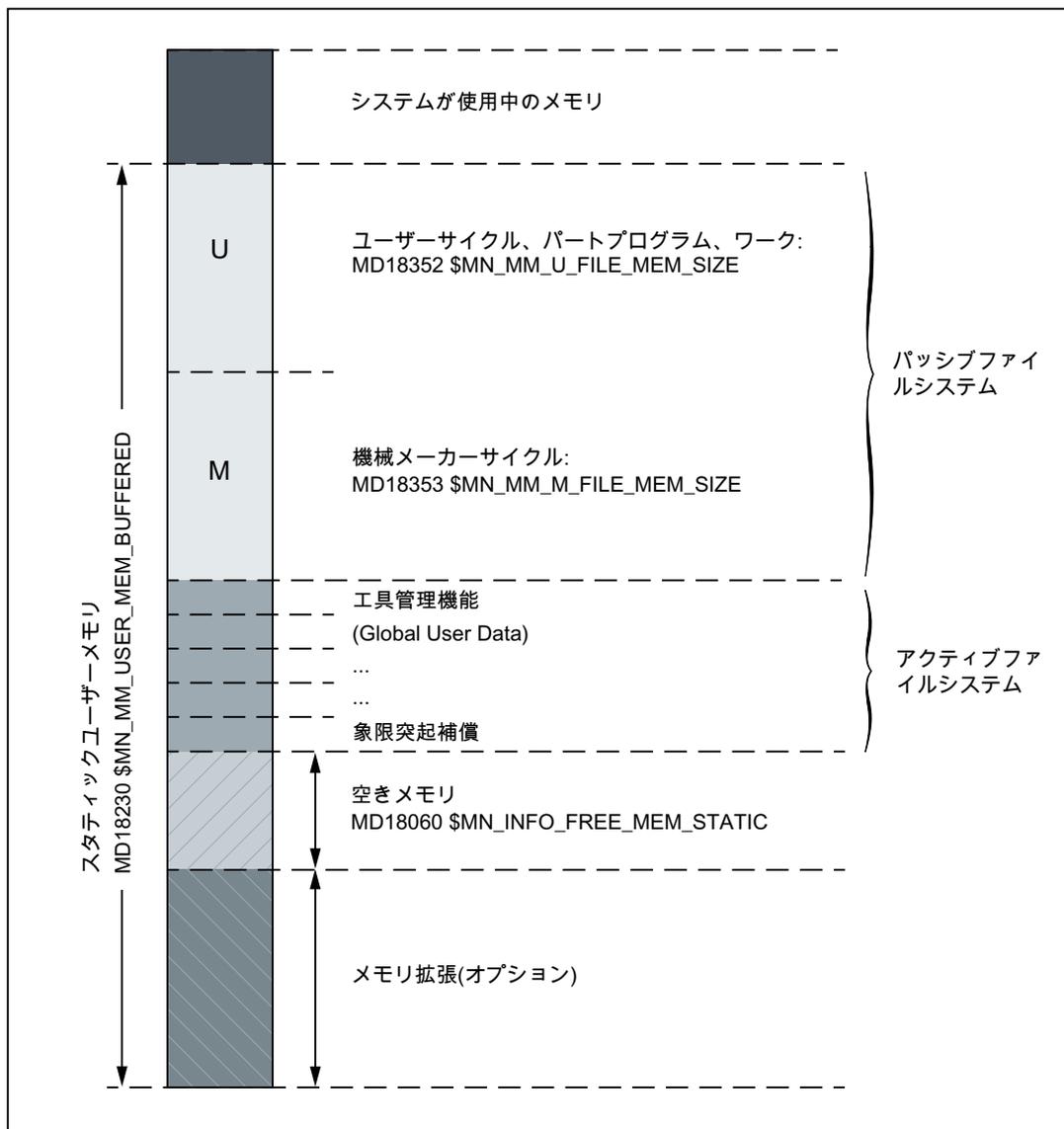


図 16-1 SINUMERIK 840D sl のスタティック NC メモリ

スタティックユーザーメモリのサイズ

スタティックユーザーメモリのサイズは次のマシンデータで表示されます。

MD18230 \$MN\_MM\_USER\_MEM\_BUFFERED

## スタティックユーザーメモリの構成部分

### パッシブファイルシステム

スタティックメモリには、パッシブファイルシステムの以下のパーティションが存在します。

パーティション	保存対象:
U (User、ユーザー)	<ul style="list-style-type: none"> <li>● <code>_N_CUS_DIR</code> ディレクトリからのファイル(ユーザーサイクル)</li> <li>● パートプログラムパートプログラム</li> <li>● ワーク</li> </ul>
M (Manufacturer = 工作機械メーカー)	<code>_N_CMA_DIR</code> ディレクトリからのファイル(工作機械メーカーサイクル)

MD18060 \$MN\_INFO\_FREE\_MEM\_STATIC に表示されるパッシブファイルシステムの空きメモリは、**U** および **M** のパーティションに区分できます。この設定は、次のマシンデータでおこないます。

- MD18352 \$MN\_MM\_U\_FILE\_MEM\_SIZE = <ユーザーデータのメモリ容量>
- MD18353 \$MN\_MM\_M\_FILE\_MEM\_SIZE = <工作機械メーカーデータのメモリ容量>

### 注記

#### S パーティション

パッシブファイルシステムの **S** パーティション(**Siemens** = コントローラメーカー)は、ダイナミックメモリ (ページ 979)にあります。

### アクティブファイルシステム

アクティブファイルシステムのメモリは、異なる領域(工具管理機能、グローバルユーザーデータなど)に区分されます。メモリを構成するマシンデータ(...MM...)で、各領域に容量を設定できます。

### 空き容量

メモリの空き容量は次のマシンデータで表示されます。

## MD18060 \$MN\_INFO\_FREE\_MEM\_STATIC

**注記**

メモリ領域の拡張に必要なメモリは、操作画面の[スタートアップ]領域に表示されません。システムセットアップエンジニアは、この情報により、計画しているメモリ拡張に必要な実メモリを見積もることができます。

**メモリ拡張(オプション)**

追加のスタティックユーザーメモリが必要な場合、オプションとして増設メモリを購入できます。

パーティション U や M を拡張するか、アクティブファイルシステムのメモリ領域を拡張するには、必要に応じて追加メモリを使用できます。

**16.4.2 セットアップ**

ユーザー毎に個々のメモリ領域を増減して、初期設定のメモリ区分を調整できます。

基本手順:

1. 標準マシンデータをロードします。  
**参照先:**  
CNC のセットアップ:NC、PLC、Drive; 「セットアップの必要条件」セクション > 「電源投入と起動」
2. スタティックユーザーメモリの最大容量の特定(オプションのメモリ増設を含む):  
MD18230 \$MN\_MM\_USER\_MEM\_BUFFERED
3. オプション:スタティックユーザーメモリの容量の変更:
  - MD19250 \$ON\_USER\_MEM\_BUFFERED
  - 電源投入リセットを実行。
4. オプション:容量パーティション U および M の設定:
  - MD18352 \$MN\_MM\_U\_FILE\_MEM\_SIZE
  - MD18353 \$MN\_MM\_M\_FILE\_MEM\_SIZE
5. オプション:コントローラの追加チャンネルの設定:  
MD10010 \$MC\_ASSIGN\_CHAN\_TO\_MODE\_GROUP

6. オプション:アクティブファイルシステムのメモリ領域の容量の設定(工具管理機能、グローバルユーザーデータなど):
  - スタティックユーザーメモリの空き容量の特定:  
MD18060 \$MN\_INFO\_FREE\_MEM\_STATIC
  - メモリを構成するマシンデータ(...MM...)のアクティブファイルシステムのメモリ領域の容量の設定:
7. 電源投入リセットを実行。  
メモリは、次のコントローラ起動で再構成されます。

**参照先:**

マシンデータの詳細については、以下を参照してください。

- マシンデータおよびパラメータのパラメータマニュアル

## 16.5 ダイナミックユーザーメモリの構成

### 16.5.1 ダイナミック NC メモリの区分

ダイナミック NC メモリは、システムとユーザーが共同で使用します。ユーザーが使用できる領域はダイナミックユーザーメモリとして定義されます。

下図は、ダイナミック NC メモリの基本的区分を示しています。

16.5 ダイナミックユーザーメモリの構成

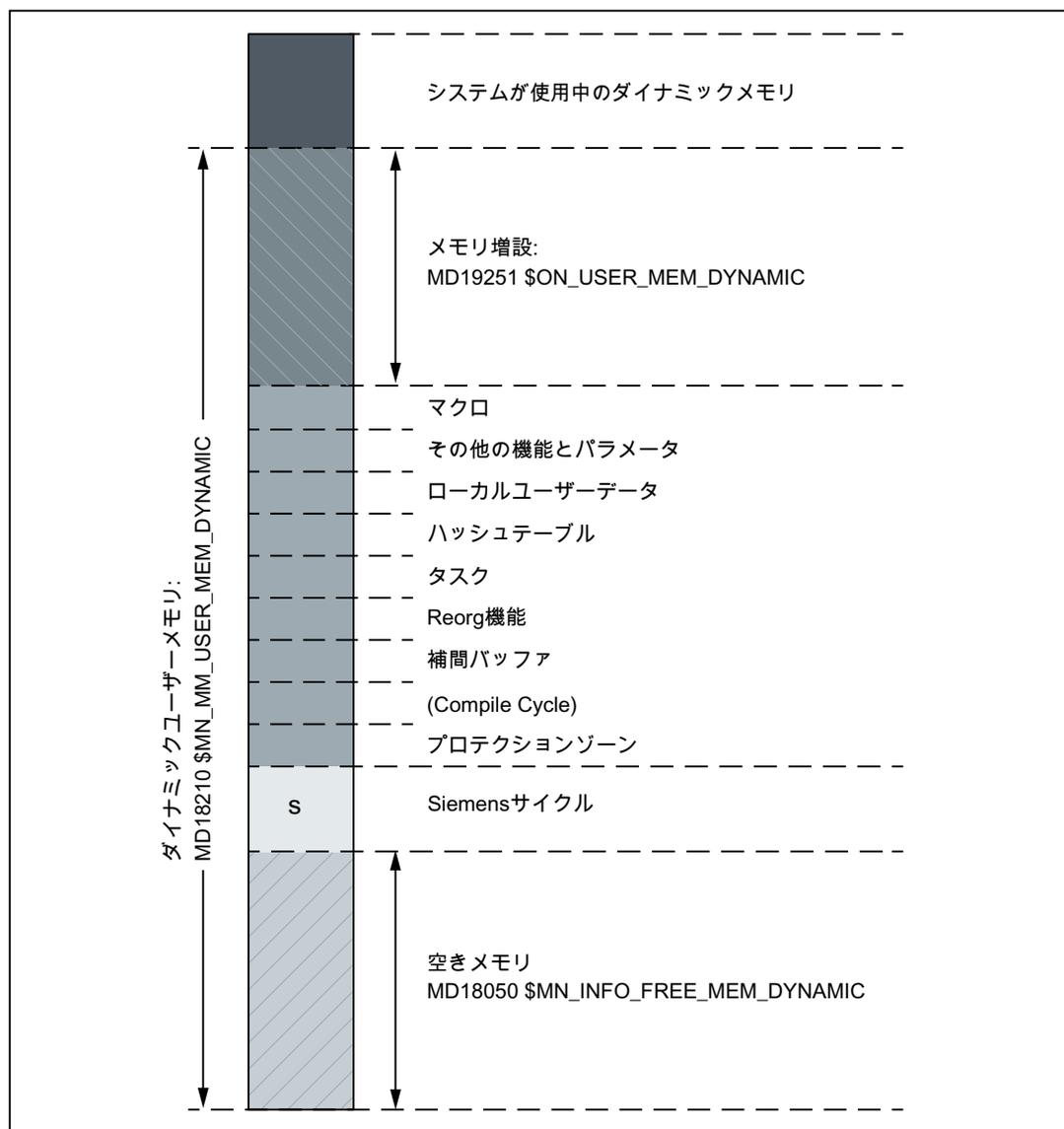


図 16-2 ダイナミック NC メモリ

ダイナミックユーザーメモリのサイズ

ダイナミックユーザーメモリのサイズは次のマシンデータで表示されます。

MD18210 \$MN\_MM\_USER\_MEM\_DYNAMIC

空きユーザーメモリ

使用可能ユーザーメモリは次のマシンデータで示されます。

MD18050 \$MN\_INFO\_FREE\_MEM\_DYNAMIC

このマシンデータの値は、チャンネル毎にユーザーデータ領域(ローカルユーザーデータ、IPO バッファなど)を拡張するために使用できるメモリ量を示しています。

### ダイナミックユーザーメモリの構成部分

#### パッシブファイルシステム

パッシブファイルシステムの S パーティションは、ダイナミックユーザーメモリにあります。

パーティション	保存対象:
S (Siemens = コントローラメーカー)	_N_CST_DIR ディレクトリからのファイル (Siemens サイクル)

**S** パーティションの容量はプリセットされており、変更できません。

### メモリ拡張

2 GB 以上のユーザーメモリを備えた NCU を使用する場合、ユーザーメモリの一部を使用してダイナミックユーザーメモリを拡張できます。

ダイナミックユーザーメモリの拡張は、次のマシンデータの 4 MB ステップで定義されます。

OD19240 \$ON\_USER\_MEM\_DYNAMIC

#### ダイナミックユーザーメモリの合計サイズ

「ダイナミックユーザーメモリの合計サイズ」 = 基本拡張 + OD19240  
\$ON\_USER\_MEM\_DYNAMIC \* 4MB

#### 表示

使用可能なダイナミックユーザーメモリは次のマシンデータで示されます。

MD18210 \$MN\_MM\_USER\_MEM\_DYNAMIC

## 16.5.2 セットアップ

ユーザー毎に個々のメモリ領域を増減して、初期設定のメモリ区分を調整できます。

基本手順:

1. ダイナミックユーザーメモリの空き容量の特定:  
MD18050 \$MN\_INFO\_FREE\_MEM\_DYNAMIC
2. オプション:ダイナミックユーザーメモリの拡張:
  - MD18210 \$MN\_MM\_USER\_MEM\_DYNAMIC
  - 電源投入リセットを実行。
3. オプション:メモリを構成するマシンデータ(...MM...)のダイナミックユーザーメモリのメモリ領域の容量の設定:
4. 電源投入リセットを実行。  
メモリは、次のコントローラ起動で再構成されます。

参照先:

マシンデータの詳細については、以下を参照してください。

- マシンデータおよびパラメータのパラメータマニュアル

## 16.6 境界条件

### 16.6.1 チャンネルの数と工具ホルダによって異なります。

チャンネル数が増えた場合は、工具ホルダの数を増やす必要がある可能性があります。

制御装置内の工具キャリアの数は、次の規則に従って設定する必要があります。

MD18088 \$MN\_NUM\_TOOL\_CARRIER =

<1つのホルダでの工具ホルダの最大数> \*

<パラメータ設定されたチャンネル数>

工具ホルダは、制御装置内で利用可能なチャンネル間で均等に分散されます。

例

NC では3つのチャンネルがパラメータ設定されます。チャンネルに必要な工具ホルダの数は次の通りです。

1. チャンネル:3 ⇒ 最大値
2. チャンネル:2
3. チャンネル:1

MD18088 = <1つのチャンネルでの工具ホルダの最大数> \* <チャンネル数> = 3 \* 3 = 9

したがって、3つの工具ホルダが各チャンネルに割り当てられます

## 16.6.2 チャンネル/軸の数のその後の低減

ウォーム再起動によって制御装置内でチャンネル/軸の数の増加が一度有効になると、この制御装置にセットアップアーカイブを作成することになっても、元に戻すことはできません。

こうしたセットアップアーカイブを制御装置に読み込むと、アラームが発生して処理は中断します。

### NC のダイナミックメモリと静的メモリの違い

- **ダイナミックメモリ:**チャンネルと軸の数が増えると、内部メモリの要求は増加します。
- **静的メモリ:**内部メモリの要求は、チャンネルの数が増えた場合のみ増加します。

## 16.7 データリスト

### 16.7.1 マシンデータ

#### 16.7.1.1 一般マシンデータ

番号	識別子: \$MN_	説明
10134	MM_NUM_MMC_UNITS	同時 HMI 通信パートナーの数
10850	MM_EXTERN_MAXNUM_OEM_GCODES	機械メーカー-G 命令の最大数
10880	MM_EXTERN_CNC_SYSTEM	最適コントロールシステムの定義
10881	MM_EXTERN_GCODE_SYSTEM	ISO_3 モード:GCodeSystem
18050	INFO_FREE_MEM_DYNAMIC	空きダイナミックメモリの表示データ
18060	INFO_FREE_MEM_STATIC	空きスタティックメモリの表示データ
18070	INFO_FREE_MEM_DPR	デュアルポート RAM の空きメモリの表示データ
18072	INFO_FREE_MEM_CC_MD	CC-MD メモリの使用可能メモリを表示
18078	MM_MAX_NUM_OF_HIERARCHIES	定義可能なマガジンロケーションタイプの階層の最大数
18079	MM_MAX_HIERARCHY_ENTRIES	マガジンロケーションタイプの階層項目の最大許容数

## 16.7 データリスト

番号	識別子: \$MN_	説明
18080	MM_TOOL_MANAGEMENT_MASK	工具管理用のメモリを予約するためのマスク
18082	MM_NUM_TOOL	NC で管理する工具数
18084	MM_NUM_MAGAZINE	NC で管理するマガジン数
18086	MM_NUM_MAGAZINE_LOCATION	マガジンロケーション数
18088	MM_NUM_TOOL_CARRIER	定義可能工具ホルダの最大数
18090	MM_NUM_CC_MAGAZINE_PARAM	コンパイルサイクルの工具管理: マガジンデータ数
18092	MM_NUM_CC_MAGLOC_PARAM	コンパイルサイクルの工具管理: マガジンロケーションデータ数
18094	MM_NUM_CC_TDA_PARAM	コンパイルサイクルの工具管理:TDA のデータ 数
18096	MM_NUM_CC_TOA_PARAM	コンパイルサイクルの工具管理:TOA のデータ 数
18098	MM_NUM_CC_MON_PARAM	コンパイルサイクルの工具管理: 監視データ数
18100	MM_NUM_CUTTING_EDGES_IN_TOA	NC の工具オフセット数
18102	MM_TYPE_OF_CUTTING_EDGE	D 番号プログラミングのタイプ
18104	MM_NUM_TOOL_ADAPTER	TO 領域の工具アダプタ
18105	MM_MAX_CUTTING_EDGE_NO	D 番号の最大値
18106	MM_MAX_CUTTING_EDGE_PERTOOL	工具毎の D 番号の最大数
18108	MM_NUM_SUMCORR	TO 操作エリアの追加オフセット数
18110	MM_MAX_SUMCORR_PER_CUTTEDGE	刃先毎の合計オフセットの最大数
18112	MM_KIND_OF_SUMCORR	TO 領域の追加オフセットの機能
18114	MM_ENABLE_TOOL_ORIENT	刃先への向きの割り当て
18116	MM_NUM_TOOL_ENV	TO 領域の工具環境数
18118	MM_NUM_GUD_MODULES	GUD モジュールの数
18120	MM_NUM_GUD_NAMES_NCK	グローバルユーザー変数の数
18130	MM_NUM_GUD_NAMES_CHAN	チャンネル別ユーザー変数の数
18140	MM_NUM_GUD_NAMES_AXIS	軸別ユーザー変数の数
18150	MM_GUD_VALUES_MEM	グローバルユーザー変数のメモリ空間
18160	MM_NUM_USER_MACROS	マクロの数

番号	識別子: \$MN_	説明
18170	MM_NUM_MAX_FUNC_NAMES	様々な関数の数
18180	MM_NUM_MAX_FUNC_PARAM	追加パラメータ数
18190	MM_NUM_PROTECT_AREA_NCK	NC の保護領域数
18200	MM_NUM_CCS_MAGAZINE_PARAM	Siemens 機械メーカーのマガジンデータ数
18201	MM_TYPE_CCS_MAGAZINE_PARAM	Siemens 機械メーカーのマガジンデータタイプ
18202	MM_NUM_CCS_MAGLOC_PARAM	Siemens 機械メーカーのマガジンロケーションデータ数
18203	MM_TYPE_CCS_MAGLOC_PARAM	Siemens 機械メーカーのマガジンロケーションデータタイプ
18204	MM_NUM_CCS_TDA_PARAM	Siemens 機械メーカーの工具データ数
18205	MM_TYPE_CCS_TDA_PARAM	Siemens 機械メーカーの工具データタイプ
18206	MM_NUM_CCS_TOA_PARAM	刃先毎の Siemens 機械メーカーのデータ数
18207	MM_TYPE_CCS_TOA_PARAM	刃先毎の Siemens 機械メーカーのデータタイプ
18208	MM_NUM_CCS_MON_PARAM	Siemens 機械メーカーの監視データ数
18209	MM_TYPE_CCS_MON_PARAM	Siemens 機械メーカーの監視データタイプ
18210	MM_USER_MEM_DYNAMIC	非保持データ用のユーザーメモリ
18220	MM_USER_MEM_DPR	デュアルポート RAM のユーザーメモリ
18230	MM_USER_MEM_BUFFERED	保持データ用のユーザーメモリ
18231	MM_USER_MEM_BUFFERED_TYPEOF	データバッファテクノロジー
18232	MM_ACTFILESYS_LOG_FILE_MEM	システム:ログファイルのサイズ
18238	MM_CC_MD_MEM_SIZE	コンパイルサイクルマシンデータ用のメモリ容量
18240	MM_LUD_HASH_TABLE_SIZE	ユーザー変数のハッシュテーブルのサイズ
18242	MM_MAX_SIZE_OF_LUD_VALUE	LUD 変数配列の最大サイズ
18250	MM_CHAN_HASH_TABLE_SIZE	チャンネル別データのハッシュテーブルのサイズ
18260	MM_NCK_HASH_TABLE_SIZE	グローバルデータのハッシュテーブルのサイズ
18270	MM_NUM_SUBDIR_PER_DIR	サブディレクトリ数
18280	MM_NUM_FILES_PER_DIR	ディレクトリ毎のファイル数
18290	MM_FILE_HASH_TABLE_SIZE	ディレクトリのファイルのハッシュテーブルのサイズ
18300	MM_DIR_HASH_TABLE_SIZE	サブディレクトリのハッシュテーブルのサイズ

## 16.7 データリスト

番号	識別子: \$MN_	説明
18310	MM_NUM_DIR_IN_FILESYSTEM	パッシブファイルシステムのディレクトリ数
18320	MM_NUM_FILES_IN_FILESYSTEM	パッシブファイルシステムのファイル数
18332	MM_FLASH_FILE_SYSTEM_SIZE	PCNC のフラッシュファイルシステムのサイズ
18342	MM_CEC_MAX_POINTS	真直度補正の最大テーブルサイズ
18350	MM_USER_FILE_MEM_MINIMUM	最小パートプログラムメモリサイズ
18352	MM_U_FILE_MEM_SIZE	パートプログラム/サイクル/ファイルのエンドユーザーメモリサイズ
18353	MM_M_FILE_MEM_SIZE	工作機械メーカーのサイクル/ファイルのメモリサイズ
18354	MM_S_FILE_MEM_SIZE	NC 工作機械メーカーのサイクル/ファイルのメモリサイズ
18355	MM_T_FILE_MEM_SIZE	一時ファイルのメモリサイズ
18356	MM_E_FILE_MEM_SIZE	外部ファイルのメモリサイズ
18360	MM_EXT_PROG_BUFFER_SIZE	外部メモリからの実行用 FIFO バッファ容量
18362	MM_EXT_PROG_NUM	外部から同時に処理できるプログラムレベルの数
18370	MM_PROTOD_NUM_FILES	ログファイルの最大数
18371	MM_PROTOD_NUM_ETPD_STD_LIST	標準 ETPD データリスト数
18372	MM_PROTOD_NUM_ETPD_OEM_LIST	ETPD 機械メーカーのデータリスト数
18373	MM_PROTOD_NUM_SERVO_DATA	ログのサーボデータ数
18374	MM_PROTOD_FILE_BUFFER_SIZE	ログファイルのバッファサイズ
18375	MM_PROTOD_SESS_ENAB_USER	セッション管理に対応可能なユーザーを設定
18390	MM_COM_COMPRESS_METHOD	サポートされている圧縮方法
18400	MM_NUM_CURVE_TABS	カーブテーブルの数(保持型)
18402	MM_NUM_CURVE_SEGMENTS	カーブセグメントの数(保持型)
18403	MM_NUM_CURVE_SEG_LIN	線形カーブセグメントの数(保持型)
18404	MM_NUM_CURVE_POLYNOMS	カーブテーブル多項式の数(保持型)
18406	MM_NUM_CURVE_TABS_DRAM	カーブテーブルの数(非保持型)
18408	MM_NUM_CURVE_SEGMENTS_DRAM	カーブセグメントの数(非保持型)
18409	MM_NUM_CURVE_SEG_LIN_DRAM	線形カーブセグメントの数(非保持型)
18410	MM_NUM_CURVE_POLYNOMS_DRAM	カーブテーブル多項式の数(非保持型)

番号	識別子: \$MN_	説明
18450	MM_NUM_CP_MODULES	CP モジュールの最大数
18452	MM_NUM_CP_MODUL_LEAD	CP 連結モジュール毎のマスタ値の最大数
18500	MM_EXTCOM_TASK_STACK_SIZE	外部通信処理のスタックサイズ(非保持型)
18502	MM_COM_TASK_STACK_SIZE	外部通信処理のスタックサイズ(k バイト)(非保持型)
18510	MM_SERVO_TASK_STACK_SIZE	サーボ処理のスタックサイズ(非保持型)
18512	MM_IPO_TASK_STACK_SIZE	IPO 処理のスタックサイズ(非保持型)
18520	MM_DRIVE_TASK_STACK_SIZE	ドライブ処理のスタックサイズ(非保持型)
18540	MM_PLC_TASK_STACK_SIZE	PLC 処理のスタックサイズ(非保持型)
18600	MM_FRAME_FINE_TRANS	FRAME の精密オフセット(保持型)
18601	MM_NUM_GLOBAL_USER_FRAMES	グローバルにあらかじめ定義されたユーザーフレームの番号(保持型)
18602	MM_NUM_GLOBAL_BASE_FRAMES	グローバル基本フレームの番号(保持型)
18660	MM_NUM_SYNACT_GUD_REAL	設定可能な実数タイプの GUD 変数の数
18661	MM_NUM_SYNACT_GUD_INT	設定可能な整数タイプの GUD 変数の数
18662	MM_NUM_SYNACT_GUD_BOOL	設定可能な論理タイプの GUD 変数の数
18663	MM_NUM_SYNACT_GUD_AXIS	設定可能な軸タイプの GUD 変数の数
18664	MM_NUM_SYNACT_GUD_CHAR	設定可能な文字タイプの GUD 変数
18665	MM_NUM_SYNACT_GUD_STRING	設定可能な文字列タイプの GUD 変数
18700	MM_SIZEOF_LINKVAR_DATA	NCU リンク可変メモリのサイズ
18710	MM_NUM_AN_TIMER	シンクロイズドアクションのグローバル時間変数の数
18720	MM_SERVO_FIFO_SIZE	IPO と閉ループ位置制御の間のバッファサイズの指令値
18780	MM_NCU_LINK_MASK	NCU リンク通信の適用
18781	NCU_LINK_CONNECTIONS	内部リンク接続数
18782	MM_LINK_NUM_OF_MODULES	NCU リンクモジュールの数
18790	MM_MAX_TRACE_LINK_POINTS	NCU リンクのトレースデータバッファのサイズ
18792	MM_TRACE_LINK_DATA_FUNCTION	NCU リンクファイルの内容を指定
18794	MM_TRACE_VDI_SIGNAL	VDI 信号のトレース指定

## 16.7 データリスト

番号	識別子: \$MN_	説明
18800	MM_EXTERN_LANGUAGE	コントロールシステムに外部言語を適用
18860	MM_MAINTENANCE_MON	保守データ記録の有効化
18870	MM_MAXNUM_KIN_CHAINS	最大キネマティック連結数
18880	MM_MAXNUM_KIN_CHAIN_ELEM	キネマティック連結の最大数
18890	MM_MAXNUM_3D_PROT_AREAS	3次元保護領域の最大要素数
18892	MM_MAXNUM_3D_PROT_AREA_ELEM	保護領域要素の最大数
18894	MM_MAXNUM_3D_PROT_GROUPS	保護領域グループの最大数
18896	MM_MAXNUM_3D_COLLISION	干渉チェック用一時メモリロケーションの最大数

## 16.7.1.2 チャンネル別マシンデータ

番号	識別子: \$MC_	説明
20096	T_M_ADDRESS_EXIT_SPINO	アドレス拡張機能としての主軸番号
27900	REORG_LOG_LIMIT	有効ログファイルの IPO バッファのパーセンテージ
28000	MM_REORG_LOG_FILE_MEM	REORG のメモリサイズ
28010	MM_NUM_REORG_LUD_MODULES	REORG のローカルユーザー変数のモジュール数
28020	MM_NUM_LUD_NAMES_TOTAL	ローカルユーザー変数の数
28040	MM_LUD_VALUES_MEM	ローカルユーザー変数のメモリサイズ
28050	MM_NUM_R_PARAM	チャンネル別 R 変数の数
28060	MM_IPO_BUFFER_SIZE	IPO バッファの NC ブロック数
28070	MM_NUM_BLOCKS_IN_PREP	ブロック解析のブロック数
28080	MM_NUM_USER_FRAMES	設定可能フレーム数
28081	MM_NUM_BASE_FRAMES	基本フレームの数(保持型)
28082	MM_SYSTEM_FRAME_MASK	システムフレーム(保持型)
28083	MM_SYSTEM_DATAFRAME_MASK	システムフレーム(保持型)
28085	MM_LINK_TOA_UNIT	チャンネルへの TO ユニットの割り当て
28090	MM_NUM_CC_BLOCK_ELEMENTS	コンパイルサイクルのブロック要素数
28100	MM_NUM_CC_BLOCK_USER_MEM	コンパイルサイクルのブロックメモリのサイズ

番号	識別子: \$MC_	説明
28105	MM_NUM_CC_HEAP_MEM	コンパイルサイクル用途のヒープメモリサイズ (KB)(非保持型)
28150	MM_NUM_VDIVAR_ELEMENTS	PLC 変数を書き込むための要素の数
28160	MM_NUM_LINKVAR_ELEMENTS	NCU リンク変数の書き込み要素数
28180	MM_MAX_TRACE_DATAPOINTS	トレースデータバッファのサイズ
28200	MM_NUM_PROTECT_AREA_CHAN	チャンネル別保護領域のモジュール数
28210	MM_NUM_PROTECT_AREA_ACTIVE	同時に有効になる保護領域の数
28212	MM_NUM_PROTECT_AREA_CONTOUR	有効なプロテクションゾーンの要素(非保持型)
28250	MM_NUM_SYNC_ELEMENTS	シンクロナイズドアクションの演算式の要素数
28252	MM_NUM_FCTDEF_ELEMENTS	FCTDEF 要素数
28254	MM_NUM_AC_PARAM	\$AC_PARAM の次元
28255	MM_BUFFERED_AC_PARAM	\$AC_PARAM[...] (保持型)
28256	MM_NUM_AC_MARKER	\$AC_MARKER の次元
28257	MM_BUFFERED_AC_MARKER	\$AC_MARKER[...] (保持型)
28258	MM_NUM_AC_TIMER	\$AC_TIMER 時間変数の数(非保持型)
28274	MM_NUM_AC_SYSTEM_PARAM	モーションシンクロナイズドアクションの \$AC_SYSTEM_PARAM の数
28276	MM_NUM_AC_SYSTEM_MARKER	モーションシンクロナイズドアクションの \$AC_SYSTEM_MARKER の数
28290	MM_SHAPED_TOOLS_ENABLE	輪郭工具の工具径補正を有効化
28300	MM_PROTOC_USER_ACTIVE	ユーザーのログの有効化
28301	MM_PROTOC_NUM_ETP_OEM_TYP	ETP 機械メーカーのイベントタイプの数
28302	MM_PROTOC_NUM_ETP_STD_TYP	ETP 標準イベントタイプ数
28400	MM_ABSBLOCK	絶対値によるブロック表示の有効化
28402	MM_ABSBLOCK_BUFFER_CONF	アップロードバッファの寸法サイズ
28450	MM_TOOL_DATA_CHG_BUFF_SIZE	工具データ変更用バッファサイズ(非保持型)
28500	MM_PREP_TASK_STACK_SIZE	準備処理のスタックサイズ
28520	MM_MAX_AXISPOLY_PER_BLOCK	ブロック毎の軸多項式の最大数
28530	MM_PATH_VELO_SEGMENTS	工具軌跡速度を制限するためのメモリユニット数
28535	MM_FEED_PROFILE_SEGMENTS	送りプロファイルのメモリユニット数

## 16.7 データリスト

番号	識別子: \$MC_	説明
28540	MM_ARCLENGTH_SEGMENTS	円弧長機能を表示するためのメモリユニット数
28560	MM_SEARCH_RUN_RESTORE_MODE	シミュレーション後のデータ復元
28580	MM_ORIPATH_CONFIG	軌跡基準の ORIPATH 工具方向の設定

## 16.7.1.3 軸/主軸別マシンデータ

番号	識別子: \$MA_	説明
38000	MM_ENC_COMP_MAX_POINTS	補間補正の中間点の数
38010	MM_QEC_MAX_POINTS	象限突起補正值の数

## T1:割り出し軸

### 17.1 概略説明

機械軸が特定数の固定位置を行き来する場合、これらの位置は割り出し位置としてパラメータ設定できます。その場合、NCプログラムではこれらの機械軸は割り出し軸となり、特殊なコマンドを使って割り出し位置を基本として移動できます。

割り出し軸の一般的な用途は、たとえば工具マガジンです(工具リボルバ、工具チェーンマガジンや工具カートリッジマガジンなど)。工具のマガジンロケーションは、割り出し位置としてパラメータ設定されます。

以下の動作状態で割り出し軸を移動することができます。

- AUTOMATIC
- MDI
- JOG
- JOG / INC1、INC10、... INCvar

### 17.2 詳細説明

#### 17.2.1 AUTOMATIC モードの割り出し軸の移動

##### 選択位置への移動

一般の移動指令(G90、G91、AC、IC、...)を使って、割り出し軸を機械軸の移動範囲内の任意の位置に移動できます。

##### 割り出し位置への移動

割り出し軸で特定の移動指令を使って、確実にパラメータ設定された割り出し位置のみに移動することができます。「プログラミング (ページ 1002)」の章を参照してください。

#### 17.2.2 JOG モードの割り出し軸の移動

JOG モードでは、割り出し位置は、割り出し軸の有効な検出器を原点確立/原点同期した後にものみ考慮されます。

### 参照先

機能マニュアル 基本機能、章「R1:原点確立」

## 連続移動(JOG CONT)

### ジョグモード

JOG モード(SD41050 \$SN\_JOG\_CONT\_MODE\_LEVELTRIGGRD = 1)では、割り出し軸は移動キーの操作後に選択した方向に移動します。割り出し軸は、移動キーを放した後、次の割り出し位置で停止します。軸が停止する割り出し位置は、以下に応じて異なります。

- 割り出し位置までの距離
- 移動速度
- 軸のダイナミック応答

### 連続運転

連続作動時(SD41050 \$SN\_JOG\_CONT\_MODE\_LEVELTRIGGRD = 0)には、移動キー操作した後(最初の信号立ち上がりエッジ)、割り出し軸が通常通り移動します。割り出し軸は、移動キーが再度操作されると(2番目の信号立ち上がりエッジ)、次の割り出し位置で停止します。軸が停止する割り出し位置は、以下に応じて異なります。

- 割り出し位置までの距離
- 移動速度
- 軸のダイナミック応答

一般的に、割り出し軸は JOG モードで移動します。

### 方向の切り替え

割り出し位置に到達する前にオペレータが逆方向の移動キーを操作して方向を切り替えた場合、割り出し軸は引き続き次の割り出し位置まで移動して、その後停止します。続いて、移動キーを再度操作して反対方向の移動動作を開始させる必要があります。

### 参照先

JOG モードでの軸の移動の詳細については、以下を参照してください。

機能マニュアル 上級機能、「H1:手動移動 (ページ 169)」の章

## インクリメンタル移動(INC)

有効な用途別機能(INC1、INC10、...、INCvar)とは関係なく、割り出し軸は移動キーが押された後、必ず次の割り出し位置まで選択した方向に移動します。

### ジョグモード

JOG モード(SD41050 \$SN\_JOG\_CONT\_MODE\_LEVELTRIGGRD = 1)では、移動キーを放すとすぐに割り出し軸は停止します。従って特定の状況下では、割り出し軸は割り出し位置で停止しません。割り出し位置で移動キーを再度操作すると、移動動作が継続します。

### 連続運転

連続作動時(SD41050 \$SN\_JOG\_CONT\_MODE\_LEVELTRIGGRD = 0)には、移動キー操作した後(最初の信号立ち上がりエッジ)、割り出し軸が通常通り移動します。移動キーが再度操作されると(2番目の信号立ち上がりエッジ)、割り出し軸はすぐに停止します。従って特定の状況下では、割り出し軸は割り出し位置で停止しません。割り出し位置で移動キーを再度操作すると、移動動作が継続します。

## 毎回転送り速度

JOG モードで割り出し軸が移動する毎回転送り速度は、以下に依存します。

SD41100 \$SN\_JOG\_REV\_IS\_ACTIVE、ビット 0 = <値>

<値>	意味
0	<p>移動モードに応じて、毎回転送り速度は以下のセッティングデータで定義されます。</p> <ul style="list-style-type: none"> <li>● 割り出し軸が<b>位置決め軸</b>として移動: SD43300 \$SA_ASSIGN_FEED_PER_REV_SOURCE (位置決め軸の毎回転送り速度)</li> <li>● 割り出し軸が<b>ジオメトリ軸</b>として移動、これに<b>回転するフレーム</b>が作用: SD42600 \$SC_JOG_FEED_PER_REV_SOURCE (JOG モードの毎回転送り速度)</li> </ul>
1	<p>割り出し軸が、メイン主軸から求められた毎回転送り速度で移動。 対応するマシンデータ:</p> <ul style="list-style-type: none"> <li>● MD32050 \$MA_JOG_REV_VELO (JOG モードの毎回転送り速度)</li> <li>● MD32040 \$MA_JOG_REV_VELO_RAPID (早送りオーバーライドによる JOG モードの毎回転送り速度)</li> </ul>

### 参照先

- 機能マニュアル 基本機能; 章「V1:送り速度」
- 機能マニュアル 上級機能、章「H1:JOG 運転とハンドル運転」
- プログラミングマニュアル 基礎編; 「送り速度制御」の章

### 17.2.3 PLCによる割り出し軸の移動

PLC ユーザープログラムから割り出し軸を移動することができます。

- 競合する位置決め軸、「P2 :位置決め軸 (ページ 803)」の章を参照してください。
- 非同期サブプログラム(ASUB)

**参照先:**

機能マニュアル 基本機能; K1:モードグループ、チャンネル、プログラム運転、リセット応答(K1)

## 17.3 セットアップ

### 17.3.1 割り出し軸のマシンデータ

#### 割り出し軸

軸は、以下の軸固有のマシンデータを使って割り出し位置テーブルをこの軸に割り当てて割り出し軸として定義します。

MD30500 \$MA\_INDEX\_AX\_ASSIGN\_POS\_TAB = <値>

<値>	意味
1	軸に割り出し位置テーブル 1 が割り当てられている。
2	軸に割り出し位置テーブル 2 が割り当てられている。
3	等間隔割り出し区間 (ページ 998)

#### 等距離の位置にある割り出し軸の最初の割り出し位置

等距離の位置にある割り出し軸の場合は、最初の割り出し位置のアブソリュート位置を、次の軸別マシンデータを使用して設定します。

MD30503 \$MA\_INDEX\_AX\_OFFSET

#### 割り出し位置テーブル

割り出し位置テーブルには、割り出し位置が mm、インチまたは度で保存されています。現在は、2 つの割り出し位置テーブルが使用可能です。

MD10910 \$MN\_INDEX\_AX\_POS\_TAB\_1 [ <n> ] (割り出し位置テーブル 1)

MD10930 \$MN\_INDEX\_AX\_POS\_TAB\_2 [ <n> ] (割り出し位置テーブル 2)

<n> = 0 ... (最大数- 1)

#### 複数テーブルの割り付け

割り出し位置テーブルには複数の軸を割り当てることができます。

前提条件：軸が同じタイプであること(直線軸、回転軸、モジュロ 360°機能)

#### 一般的な必要条件

- 割り出し位置がギャップなく昇順に割り出し位置テーブルにリストされていること。
- 連続する割り出し位置が同じでないこと。
- 割り出し位置が基本座標系で指定されていること。

#### モジュロ回転軸の追加条件

- 許容範囲: $0^{\circ} \leq$  割り出し位置  $< 360^{\circ}$
- 正の回転方向で次の割り出し位置に移動する際、割り出し軸が割り出し位置テーブルの最後の割り出し位置である場合、最初の割り出し位置に移動します。
- 負の回転方向で次の割り出し位置に移動する際、割り出し軸が割り出し位置テーブルの最初の割り出し位置である場合、最後の割り出し位置に移動します。

## 割り出し位置数

割り出し位置テーブルには、最大 60 個の割り出し位置を入力できます。割り出し位置の実数、または以下のマシンデータでプログラム可能な数を定義します。

MD10900 \$MN\_INDEX\_AX\_LENGTH\_POS\_TAB\_1 (割り出し位置テーブル 1 の割り出し位置の位置数)

MD10920 \$MN\_INDEX\_AX\_LENGTH\_POS\_TAB\_2 (割り出し位置テーブル 2 の割り出し位置の位置数)

---

#### 注記

割り出し位置の数のパラメータ設定を超えて割り出し位置テーブルの入力しても、それは無効です。

---

## 割り出し位置の単位系

割り出し位置の単位系は、NC 別に定義します。

MD10270 \$MN\_POS\_TAB\_SCALING\_SYSTEM = <値>

<値>	単位系
0	メトリック
1	インチ

---

**注記**

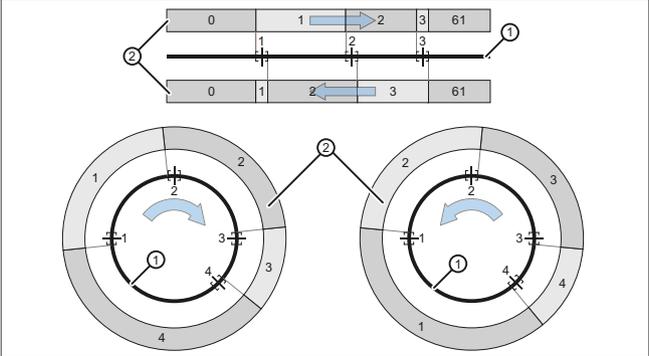
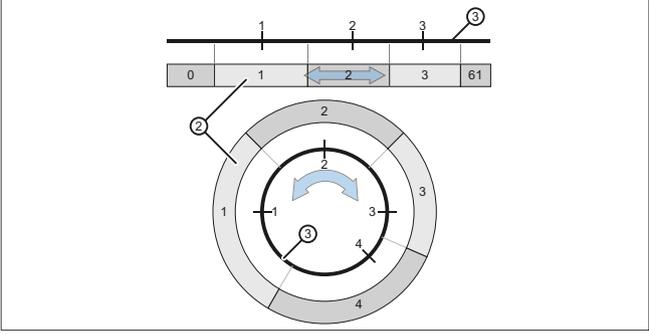
マシンデータは、以下が適用される場合のみ評価されます。MD10260  
\$MN\_CONVERT\_SCALING\_SYSTEM == 1

---

**実際の割り出し位置の表示**

割り出し軸を移動するとき、どの実際の位置で実際の割り出し位置の表示を変更するかをシステム変数\$AA\_ACT\_INDEX\_AX\_POS\_NO で定義するためにマシンデータが使用されます。

MD10940 \$MN\_INDEX\_AX\_MODE, bit 0 = &lt;値&gt;

ビット	<値>	意味
0	0	<p>システム変数\$AA_ACT_INDEX_AX_POS_NO で示される実際の割り出し位置は、移動方向の次の割り出し位置の精密イグザクトストップ範囲 (MD36010 \$MA_STOP_LIMIT_FINE)に到達またはそれを通過した場合に変更されます。</p> 
	1	<p>システム変数\$AA_ACT_INDEX_AX_POS_NO で示される実際の割り出し位置は、移動方向の次の割り出し位置までの移動距離の半分に到達またはそれを通過したときに変更されます。</p> 
①		割り出し軸(直線軸または回転モジュロ軸)の移動範囲と割り出し位置および精密イグザクトストップ範囲
②		システム変数\$AA_ACT_INDEX_AX_POS_NO で示される移動方向依存の割り出し位置
③		割り出し軸(直線軸または回転モジュロ軸)の移動範囲と割り出し位置

## 17.3 セットアップ

## 例

前提条件:

- MD10940 \$MN\_INDEX\_AX\_MODE、ビット 0 == 1
- 軸が、「精密イグザクトストップ」(DB31, ... .DBX60.7 == 0)許容範囲外の割り出し位置の前にあること

システム変数\$AA\_ACT\_INDEX\_AX\_POS\_NO が割り出し位置 2 を示しても、「次の位置へ移動」命令では割り出し位置 3 には**移動せず**、まず、最初に割り出し位置 2 に正確に移動します。軸が正確に(DB31, ... .DBX60.7 == 1)にあるか、表示された割り出し位置の後にある場合にのみ、「次の位置へ移動」命令を使用して、次の割り出し位置(例では、割り出し位置 3)に移動します。つまり、**移動方向**に最も近い割り出し位置に移動します。よって、特定の状況では、「次の位置に移動」命令を 2 回発行し、現在表示されている割り出し位置から次の割り出し位置に移動する必要があります。

## 17.3.2 等間隔割り出し区間のマシンデータ

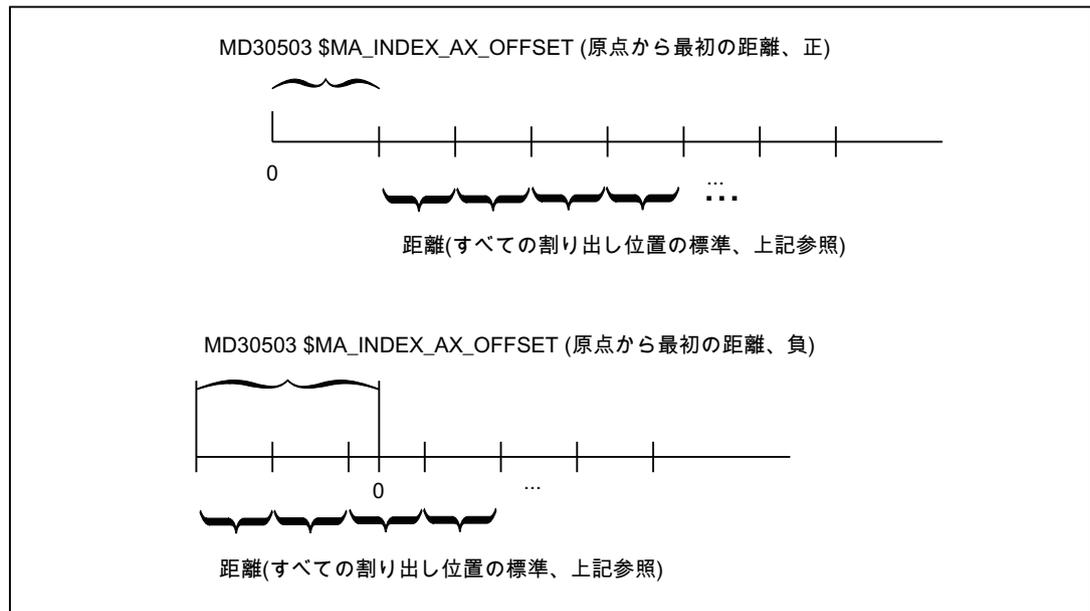
## 17.3.2.1 機能

## 割り出し間の距離

等間隔割り出し区間の割り出し距離は、次式に従って決定されます。

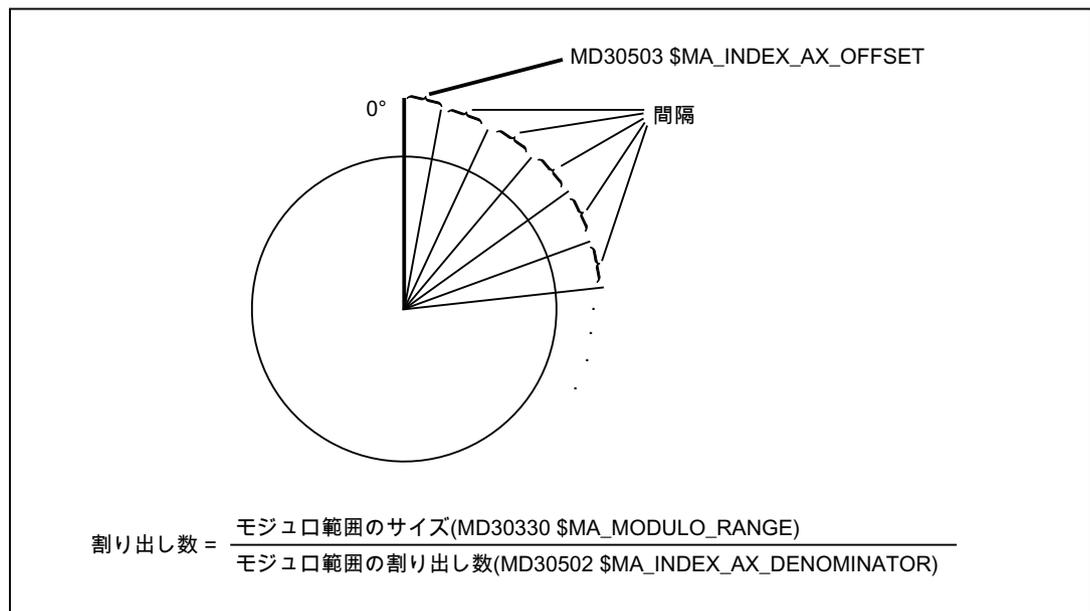
$$\text{距離} = \frac{\text{分子(MD30501 \$MA_INDEX_AX_NUMERATOR)}}{\text{分母(MD30502 \$MA_INDEX_AX_DENOMINATOR)}}$$

## 直線軸



## モジュロ回転軸

$$\text{割り出し数} = \frac{\text{分子(MD30330 \$MA_MODULO_RANGE)}}{\text{分母(MD30502 \$MA_INDEX_AX_DENOMINATOR)}}$$



### 17.3.2.2 ハース軸

#### 機能

「ハース軸」の場合、割り出し位置に到達すると、特別なギヤ式同期動作(ハースギヤ式同期動作)を使って、回転軸がインターロックされます。この場合、直線軸を使ってロックボルトまたはギヤが締結されます。機械の機械システムの損傷を防ぐため、インターロックは割り出し位置のみで締結できます。ハース軸では、すべての割り出し位置の間隔が等距離であることが必要です(等間隔の割り出し位置)。

#### 必要条件

- 「ハース軸」は回転軸であること。
- 「ハース軸」は割り出し軸であること。「セットアップ (ページ 992)」の章を参照してください。
- 割り出し軸の有効な検出器が同期している、または原点確立済みである。  
DB31、... DBX60.4/1.5 == 1 (原点確立済み/原点同期済み、エンコーダ 1/2)  
参照先:機能マニュアル 基本機能; レファレンス点復帰(R1)

#### セットアップ

##### マシンデータ

- MD30300 \$MA\_IS\_ROT\_AX = 1 (回転軸)
- MD30505 \$MA\_HIRTH\_IS\_ACTIVE[ <軸> ] = 1 (ハースギヤ付きの割り出し軸)
- MD30500 \$MA\_INDEX\_AX\_ASSIGN\_POS\_TAB[ <軸> ] = 3 (等間隔の割り出し位置を持つ割り出し軸)

### 17.3.3 軸固有のシステム変数

#### \$AA\_PROG\_INDEX\_AX\_POS\_NO

##### 機能

システム変数には、割り出し軸にプログラムされた割り出し位置の番号が含まれます。

##### 構文

\$AA\_PROG\_INDEX\_AX\_POS\_NO[ <軸> ]

意味

\$AA_PROG_INDEX_AX_POS_NO	
0	軸は割り出し軸でないか、現在、割り出し軸は割り出し位置に移動していません。
> 0	プログラム指令割り出し位置番号

**\$AA\_ACT\_INDEX\_AX\_POS\_NO**

機能

システム変数には、割り出し軸が最後に到達した割り出し位置、または割り出し軸が通過した割り出し位置の番号が含まれます。

構文

\$AA\_ACT\_INDEX\_AX\_POS\_NO [<軸>]

意味

\$AA_ACT_INDEX_AX_POS_NO	
0	軸は割り出し軸ではありません。
> 0	最後に到達または通過した割り出し位置の番号

**17.3.4 NC/PLC インタフェース信号**

軸別のインターフェース信号

割り出し軸が割り出し位置±精密イグザクトストップ範囲(MD36010

\$MA\_STOP\_LIMIT\_FINE)で停止する場合、次の軸別のインターフェース信号が設定されます。

DB31、... DBX76.6 = 1 (割り出し軸がインポジション)

## 17.4 プログラミング

### 符号化位置

NC パートプログラムから割り出し軸を位置決めするために特別な命令が用意されており、mm や度による軸の位置決めの代わりに、この命令を使用して割り出し番号(位置番号など)をプログラム指令します。特別な命令を適用できるかどうかは、軸のタイプ(直線軸または回転軸)によって決まります。

命令	作用	適用
CAC (i)	アブソリュート値で符号化位置を移動	直線軸、回転軸
CACP (i)	正方向にアブソリュート値で符号化位置に移動	回転軸
CACN (i)	負方向にアブソリュート値で符号化位置に移動	回転軸
CDC (i)	直接(最短)軌跡で符号化位置に移動	回転軸
CIC (i)	符号化位置に 1 単位毎に移動	直線軸、回転軸

i: 符号化位置(割り出し位置)

i の数値の範囲: 0 ... 59; 整数(CIC では正と負の値が可能)

### 例

プログラムコード	コメント
POS[B]=CAC(20)	割り出し軸 B は符号化位置(インデックス)20 にアブソリュートモードでアプローチします。移動方向は現在の実位置によって決まります。
POS[B]=CACP(10)	割り出し軸 B は、 <b>正方向の回転によってアブソリュートモード</b> で符号化位置(割り出し位置)10 にアプローチします(回転軸のみで可能)。
POS[B]=CACN(10)	割り出し軸 B は、 <b>負方向の回転によってアブソリュートモード</b> で符号化位置(割り出し位置)10 にアプローチします(回転軸のみで可能)。
POS[B]=CDC(50)	割り出し軸 B は、 <b>最短軌跡にそって直接</b> 割り出し位置 50 にアプローチします(回転軸のみで可能)。
POS[B]=CIC(-4)	割り出し軸 B は、 <b>負の方向</b> に現在位置から 4 つの割り出し位置へ <b>インクリメンタル</b> で移動します。
POS[B]=CIC(35)	割り出し軸 B は、 <b>正方向</b> に現在位置から 35 だけ先の割り出し位置へ <b>インクリメンタル</b> で移動します。

## 特記事項

- 割り出し軸としてのモジュロ回転軸  
モジュロ回転軸では、割り出し位置を係数  $360^\circ$  で割って直接アプローチします。
- 2つの割り出し位置の間にある割り出し軸  
指定された位置命令は、AUTOMATIC モードで次のように動作します。

POS [B]=CIC (1)      次に高い割り出し位置にアプローチします。

POS [B]=CIC (-1)    次に低い割り出し位置にアプローチします。

POS [B]=CIC (0)      割り出し軸は移動しません。

## システム変数

最後にプログラム指令した割り出し位置番号は、次のシステム変数で読み取ることができます。

**\$AA\_PROG\_INDEX\_AX\_POS\_NO**

最後に移動した割り出し位置番号は、次のシステム変数で表示できます。

**\$AA\_ACT\_INDEX\_AX\_POS\_NO**

表示は次のマシンデータの設定によって決まります。

MD10940 \$MN\_INDEX\_AX\_MODE (割り出し位置の設定)

ビット	規格値	意味
0	0	割り出し位置に達すると(「精密イグザクトストップ」の範囲)、割り出し位置が更新されます、そして次の割り出し位置に達するまでは更新されません。割り出し領域は、ある割り出し位置から始まり、その次の割り出し位置の手前で終了します。
	1	割り出し位置は、割り出し位置の半分に到達すると更新します。 準対称割り出し領域は、割り出し位置について適用されます(等距離割り出しの直線軸、または割り出し領域がモジュロ範囲(MD30330 \$MA_MODULO_RANGE)の整数倍であるモジュロ回転軸のみでは対称。その他の場合は割り出し位置間の距離に比例)。 <b>モジュロ回転軸</b> では、最終割り出し位置と先頭割り出し位置の間の領域が、先頭割り出し領域と最終割り出し領域の長さを基に <b>比例的に分割</b> されます。

次の図は、ビット0=0とビット0=1の違いを示しています。

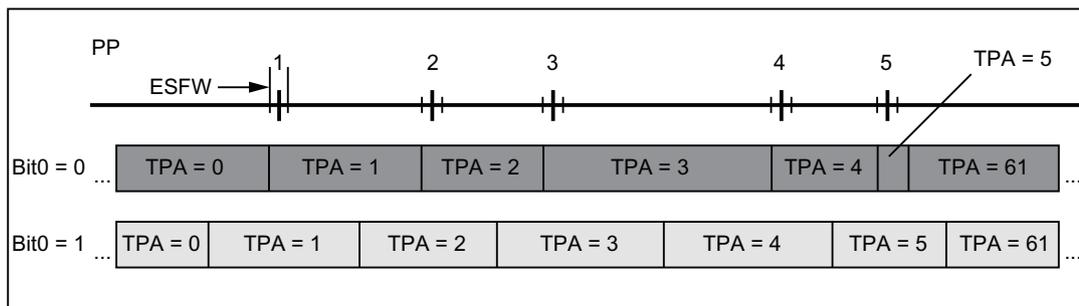
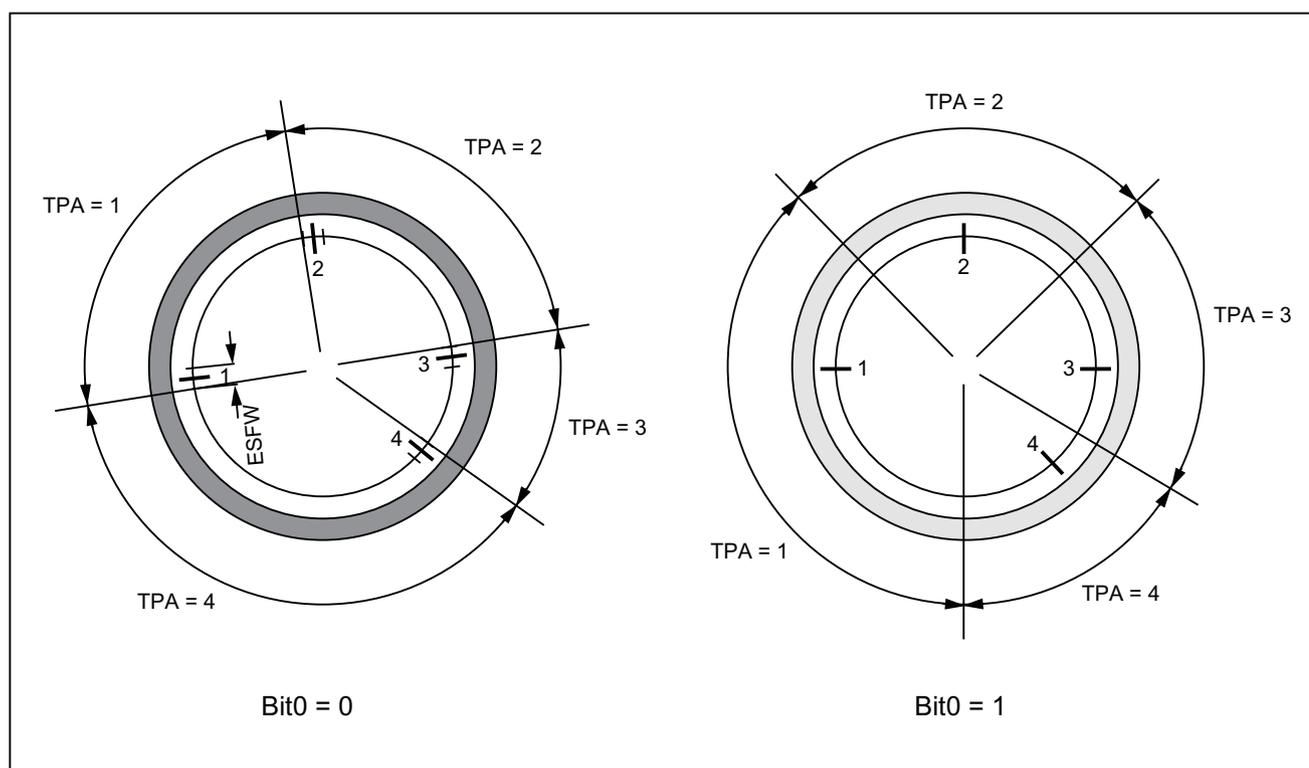


図 17-1 割り出し位置表示:直線軸



- TP 割り出し位置
- TPA 表示割り出し位置
- ESFW 「精密イグザクトストップ」範囲

図 17-2 割り出し位置表示:モジュロ回転軸

## システム変数\$AA\_ACT\_INDEX\_AX\_POS\_NO

\$AA\_ACT\_INDEX\_AX\_POS\_NO:最後に到達または通過した割り出し位置の番号

割り出し位置			
軸のタイプ	値	説明	
モジュロ回転軸	1, ... n	n = 割り出し位置数 (ページ 992)	
直線軸	0, 1, 2, 3, ... 59, 60, 61	0	実際の位置が最初の割り出し位置より <b>前</b> にある
		61	実際の位置が最後の割り出し位置より <b>後</b> にある

等距離割り出し位置			
軸のタイプ	値	説明	
モジュロ回転軸	1 ... m	m = 分母(カウンタ)	
直線軸	... -3, -2, -1, 0, 1, 2, 3, ...		

## 17.5 必要条件

### 軸割り出し

さまざまなチャンネルおよび軸別の NC/PLC インターフェース信号

割り出し軸の移動中に、以下の信号が 1 つ以上発生した場合、軸はすぐに停止します。割り出し位置は考慮されません。

- チャンネル別信号
  - DB21、... DBB4 == 0 (送り速度オーバーライド)
  - DB21、... DBX6.2 == 1 (残移動距離削除)
  - DB21、... DBX7.3 == 1 (NC ストップ)
  - DB21、... DBX7.4 == 1 (NC ストップ軸と主軸)
  - DB21、... DBX7.7 == 1 (リセット)
- 軸別の信号
  - DB31、... DBX4.3 == 1 (送り速度ストップ)

**シンクロナイズドアクション**

- ストップ(MOV=0)  
MOV=0 を使用してシンクロナイズドアクションで割り出し軸を停止した場合、軸は移動方向の次の割り出し位置で停止します。
- 残移動距離の削除 (DELDTG)  
DELDTG を使ってシンクロナイズドアクションで割り出し軸の残移動距離が削除された場合、軸は割り出し位置で停止せず、すぐに停止します。

参照先:機能マニュアル シンクロナイズドアクション

**フレームおよびゼロオフセット**

割り出し軸との組み合わせでは、一般的にフレームは不要です。その結果、割り出し軸のフレームおよびゼロオフセットを NC プログラムで抑制することを推奨します。

**ソフトウェアリミットスイッチ**

JOG または JOG/INC 操作モードで手動で割り出し軸を移動する場合、割り出し軸は(最大)ソフトウェアのリミットスイッチ手前の最後の割り出し位置まで移動します。

**ハンドル(DRF)を使用した手動移動**

ハンドルを使って割り出し軸を移動する場合、割り出し位置は考慮されません。これは、2つの割り出し位置の間でも、手動パルスの最後に割り出し軸がすぐに停止することを意味します。

**「ハース軸」**

上述の二次的条件に加え、「ハース軸」では、以下の追加の二次的条件を遵守してください。

**さまざまなチャネルおよび軸別の NC/PLC インターフェース信号**

「ハース軸」の移動中に、以下の信号が 1 つ以上発生した場合、軸は移動方向の次の割り出し位置ですぐに停止します。

- チャネル別信号
  - DB21、... DBB4 == 0 (送り速度オーバーライド)
  - DB21、... DBX6.2 == 1 (残移動距離削除)
  - DB21、... DBX7.3 == 1 (NC ストップ)
  - DB21、... DBX7.4 == 1 (NC ストップ軸と主軸)
  - DB21、... DBX7.7 == 1 (リセット)
- 軸別の信号
  - DB31、... DBX4.3 == 1 (送り速度ストップ)

**非常停止**

「ハース軸」の移動中に非常停止が開始された場合、軸はすぐに停止します。

DB10 DBX56.1 == 1 (非常停止)

参照先:機能マニュアル 基本機能; 章「N2:非常停止」

**通知****割り出し位置への移動**

非常停止後、機械ハースギヤのインターロックが再度締結する前に、まず PLC ユーザープログラムまたはオペレータ(手動)が JOG モードで「ハース軸」を割り出し位置に移動する必要があります。

**シンクロナイズドアクション**

## ● 残移動距離の削除 (DELDTG)

「ハース軸」で、DELDTG を使ってシンクロナイズドアクションで残移動距離を削除した場合、軸は移動方向の次の割り出し位置で停止します。

参照先:機能マニュアル シンクロナイズドアクション、「詳細説明」章 > 「シンクロナイズドアクションの操作」 > 「残移動距離の削除(DELDTG)」

**さまざまな軸別の機能**

「ハース軸」は、以下の機能と組み合わせて使用できません。

## ● キネマティックトランスフォーメーション

参照先:機能マニュアル 上級機能、章「M1:キネマティックトランスフォーメーション」

## ● 現在値の設定によるゼロオフセット:PRESETON

参照先:プログラミングマニュアル 上級編、「座標変換(フレーム)」章

## ● 毎回転送り速度(G95、FPRAON)

## ● ハンドルを使用した手動移動、およびハンドル(DFR)を使用した距離または速度加算

参照先:機能マニュアル 上級機能、章「H1:手動移動およびハンドルを使用した手動移動」

## ● フレーム

参照先:機能マニュアル 基本機能、章「K2:軸、座標系、フレーム」 > 「フレーム」

## 17.6 例

- 補間訂正、たとえば、外部ゼロオフセット、DRF オフセット
- 連結
 

「ハース軸」が、以下のいずれかの連動機能の従動軸となることは許可されません。

  - 軸間連動機能を使用した従属軸  
参照先:機能マニュアル 応用機能、章「M3:軸連結」 > 「軸間連動機能」または「一般の連動機能」
  - 連結モーション軸  
参照先:機能マニュアル 応用機能、章「M3:軸連結」 > 「連結モーション」または「一般の連動機能」
  - ガントリスレーブ軸  
参照先:機能マニュアル 応用機能、章「G1:ガントリ軸」

## 17.6 例

## 17.6.1 例 1:割り出し軸としての回転軸

機械軸 AX5 は、8 つのリボルバロケーションをもつ工具リボルバです(移動制限のない回転の回転軸)。リボルバロケーション間の距離は一定です(等間隔の割り出し位置)。

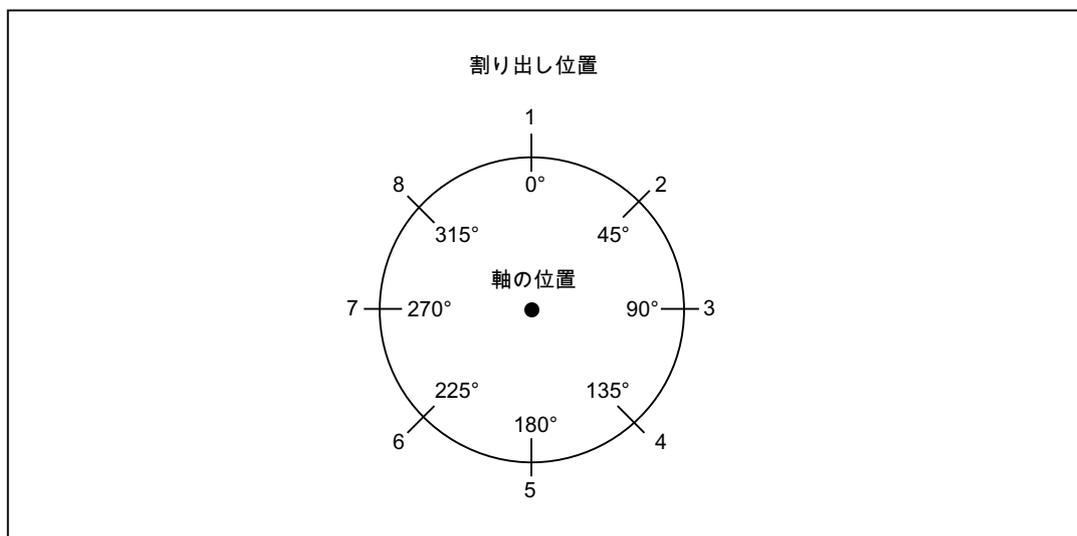


図 17-3 8 個のロケーションがある工具タレット

## 割り出し位置テーブル 1

MD10910 \$MN_INDEX_AX_POS_TAB_1[ 0 ] = 0	1 番目の割り出し位置 = 0°
MD10910 \$MN_INDEX_AX_POS_TAB_1[ 1 ] = 45	2 番目の割り出し位置 = 45°

MD10910 \$MN_INDEX_AX_POS_TAB_1[ 2 ] = 90	3 番目の割り出し位置= 90°
MD10910 \$MN_INDEX_AX_POS_TAB_1[ 3 ] = 135	4 番目の割り出し位置= 135°
MD10910 \$MN_INDEX_AX_POS_TAB_1[ 4 ] = 180	5 番目の割り出し位置= 180°
MD10910 \$MN_INDEX_AX_POS_TAB_1[ 5 ] = 225	6 番目の割り出し位置= 225°
MD10910 \$MN_INDEX_AX_POS_TAB_1[ 6 ] = 270	7 番目の割り出し位置= 270°
MD10910 \$MN_INDEX_AX_POS_TAB_1[ 7 ] = 315	8 番目の割り出し位置= 315°

割り出し位置テーブル(1)の長さ

MD10900 \$MN_INDEX_AX_LENGTH_POS_TAB_1= 8	8 個の割り出し位置がテーブル 1 にある
--	--------------------------

機械軸 AX5 を回転モジュロ軸および割り出し軸として定義

MD30300 \$MA_IS_ROT_AX[ AX5 ] = 1	回転軸
MD30310 \$MA_ROT_IS_MODULO [AX5] = 1	モジュロ回転軸
MD30500 \$MA_INDEX_AX_ASSIGN_POS_TAB [AX5] = 1	割り出し軸、割り出し位置テーブ ル 1

## 17.6.2 例 2:直線軸としての割り出し軸

機械軸 AX6 は 10 個のロケーションがあるワークホルダです。10 個のロケーション間の距離は異なります。第 1 ロケーションは-100mm です。機械軸は AX6 です。

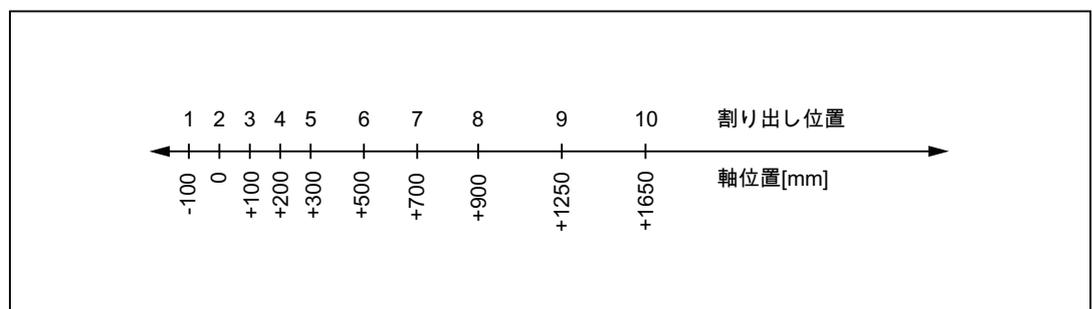


図 17-4 ワークホルダの割り出し位置

割り出し位置テーブル 2

MD10930 \$MN_INDEX_AX_POS_TAB_2[ 0 ] = -100	1 番目の割り出し位置= -100 mm
MD10930 \$MN_INDEX_AX_POS_TAB_2[ 1 ] = 0	2 番目の割り出し位置= 0 mm
MD10930 \$MN_INDEX_AX_POS_TAB_2[ 2 ] = 100	3 番目の割り出し位置= 100 mm
MD10930 \$MN_INDEX_AX_POS_TAB_2[ 3 ] = 200	4 番目の割り出し位置= 200 mm
MD10930 \$MN_INDEX_AX_POS_TAB_2[ 4 ] = 300	5 番目の割り出し位置= 300 mm
MD10930 \$MN_INDEX_AX_POS_TAB_2[ 5 ] = 500	6 番目の割り出し位置= 500 mm
MD10930 \$MN_INDEX_AX_POS_TAB_2[ 6 ] = 700	7 番目の割り出し位置= 700 mm
MD10930 \$MN_INDEX_AX_POS_TAB_2[ 7 ] = 900	8 番目の割り出し位置= 900 mm
MD10930 \$MN_INDEX_AX_POS_TAB_2[ 8 ] = 1250	9 番目の割り出し位置= 1250 mm
MD10930 \$MN_INDEX_AX_POS_TAB_2[ 9 ] = 1650	10 番目の。割り出し位置 = 1650 mm

割り出し位置テーブル(1)の長さ

MD10920 \$MN_INDEX_AX_LENGTH_POS_TAB_2 = 10	10 個の割り出し位置がテーブル 2 にある
--	---------------------------

機械軸 AX5 を割り出し軸として定義

MD30500 \$MA_INDEX_AX_ASSIGN_POS_TAB [AX6] = 2	割り出し軸、割り出し位置テーブル 2
---	--------------------

### 17.6.3 例 3 :等間隔の割り出し軸としての回転モジュロ軸

機械軸 AX4 は等間隔の割り出し位置を備えた回転モジュロ軸です。割り出し位置は 5° から始まります。割り出し位置の間隔は 20° です。

以下の割り出し位置が得られます。

5、25、45、65、85、105、125、145、165、185、205、225、245、265、285、305、325、245 [度]

## 割り出しデータ

MD30501 \$MA_INDEX_AX_NUMERATOR[AX4] ⇒ MD30330 \$MA_MODULO_RANGE[AX4] = 360.0	この値は回転モジュロ軸では無視され、代わりに MD30330 \$MA_MODULO_RANGE (初期設定:360°)が使用されます。
MD30502 \$MA_INDEX_AX_DENOMINATOR[AX4] = 18	割り出し位置数 = $360^\circ / 20^\circ = 18$
MD30503 \$MA_INDEX_AX_OFFSET[AX4] = 5.0	最初の割り出し位置 = $5.0^\circ$
MD30505 \$MA_HIRTH_IS_ACTIVE[AX4] = 0	ハースギヤなし

機械軸 AX4 を回転モジュロ軸および等間隔の割り出し軸として定義

MD30300 \$MA_IS_ROT_AX[AX4] = 1	回転軸
MD30310 \$MA_ROT_IS_MODULO[AX4] = 1	モジュロ機能有効
MD30500 \$MA_INDEX_AX_ASSIGN_POS_TAB[AX4] = 3	等間隔の割り出し軸

#### 17.6.4 例 4 :移動範囲制限付きの等間隔の割り出し軸としての回転軸

機械軸 AX4 は等間隔の割り出し位置を備えた回転軸です。等間隔の割り出し位置は  $100^\circ$  から開始し、 $20^\circ$  間隔です。 $100^\circ$  未満および  $260^\circ$  超の位置には移動しません。

以下の割り出し位置が得られます。

100、120、140、160、180、200、220、240、260 [度]

## 割り出しデータ

MD30501 \$MA_INDEX_AX_NUMERATOR[AX4] = 360.0	分子 = $360^\circ$
MD30502 \$MA_INDEX_AX_DENOMINATOR[AX4] = 18	割り出し位置数 = $360^\circ / 20^\circ = 18$
MD30503 \$MA_INDEX_AX_OFFSET[AX4] = 100.0	最初の割り出し位置 = $100.0^\circ$
MD30505 \$MA_HIRTH_IS_ACTIVE[AX4] = 0	ハースギヤなし

機械軸 AX4 を回転モジュロ軸および等間隔の割り出し軸として定義

MD30300 \$MA_IS_ROT_AX[ AX4 ] = 1	回転軸
MD30310 \$MA_ROT_IS_MODULO[ AX4 ] = 1	モジュロ機能有効
MD30500 \$MA_INDEX_AX_ASSIGN_POS_TAB[ AX4 ] = 3	等間隔の割り出し軸
MD36100 \$MA_POS_LIMIT_MINUS[ AX4 ] = 100.0	1 番目のソフトウェアリミットスイッチマイナス = 100.0°
MD36110 \$MA_POS_LIMIT_PLUS[ AX4 ] = 260.0	1 番目のソフトウェアリミットスイッチプラス = 260.0°

### 17.6.5 例 5 :等間隔の割り出し軸としての直線軸

機械軸 AX4 は等間隔の割り出し位置を備えた直線軸です。等間隔の割り出し位置は-200 mm から開始し、10 mm 間隔です。

以下の割り出し位置が得られます。

-200、-190、-180、... 180、190、200 [mm]

割り出しデータ

MD30501 \$MA_INDEX_AX_NUMERATOR[ AX4 ] = 10.0	分子 = 10
MD30502 \$MA_INDEX_AX_DENOMINATOR[ AX4 ] = 1	分母 = 1
MD30503 \$MA_INDEX_AX_OFFSET[ AX4 ] = -200.0	最初の割り出し位置 = -200.0°
MD30505 \$MA_HIRTH_IS_ACTIVE[ AX4 ] = 0	ハースギヤなし

機械軸 AX4 を直線軸および等間隔の割り出し軸として定義

MD30300 \$MA_IS_ROT_AX[ AX4 ] = 0	直線軸
MD30310 \$MA_ROT_IS_MODULO[ AX4 ] = 0	モジュロ機能無効
MD30500 \$MA_INDEX_AX_ASSIGN_POS_TAB[ AX4 ] = 3	等間隔の割り出し軸

MD36100 \$MA_POS_LIMIT_MINUS[ AX4 ] = -200.0	1 番目のソフトウェアリミットスイッチマイナス= -200.0 mm
MD36110 \$MA_POS_LIMIT_PLUS[ AX4 ] = 200.0	1 番目のソフトウェアリミットスイッチプラス= 200.0 mm

### 17.6.6 例 6: 「ハース軸」

機械軸 AX4 は「ハース軸」(等間隔のインデックス位置およびハースギヤを備えた回転モジュロ軸)です。等間隔の割り出し位置は 0°から開始し、1°間隔です。

#### 割り出しデータ

MD30501 \$MA_INDEX_AX_NUMERATOR[ AX4 ] ⇒ MD30330 \$MA_MODULO_RANGE[ AX4 ] = 360.0	この値は回転モジュロ軸では無視され、代わりに MD30330 \$MA_MODULO_RANGE (初期設定:360°)が使用されます。
MD30502 \$MA_INDEX_AX_DENOMINATOR[ AX4 ] = 360	割り出し位置数 = $360^\circ / 1^\circ = 360$
MD30503 \$MA_INDEX_AX_OFFSET[ AX4 ] = 0.0	最初の割り出し位置 = 0.0°
MD30505 \$MA_HIRTH_IS_ACTIVE[ AX4 ] = 1	ハースギヤ

機械軸 AX4 を直線軸および等間隔の割り出し軸として定義

MD30300 \$MA_IS_ROT_AX[ AX4 ] = 1	回転軸
MD30310 \$MA_ROT_IS_MODULO[ AX4 ] = 1	モジュロ機能有効
MD30500 \$MA_INDEX_AX_ASSIGN_POS_TAB[ AX4 ] = 3	等間隔の割り出し軸
MD36100 \$MA_POS_LIMIT_MINUS[ AX4 ] = -200.0	1 番目のソフトウェアリミットスイッチマイナス= -200.0 mm
MD36110 \$MA_POS_LIMIT_PLUS[ AX4 ] = 200.0	1 番目のソフトウェアリミットスイッチプラス= 200.0 mm

## 17.7 データリスト

### 17.7.1 マシンデータ

#### 17.7.1.1 一般マシンデータ

番号	識別子: \$MN_	説明
10260	CONVERT_SCALING_SYSTEM	基本単位系切り替え有効
10270	POS_TAB_SCALING_SYSTEM	位置テーブルの単位系
10900	INDEX_AX_LENGTH_POS_TAB_1	割り出し軸テーブル 1 の位置の数
10910	INDEX_AX_POS_TAB_1[n]	割り出し位置テーブル 1
10920	INDEX_AX_LENGTH_POS_TAB_2	割り出し軸テーブル 2 の位置の数
10930	INDEX_AX_POS_TAB_2[n]	割り出し位置テーブル 2
10940	INDEX_AX_MODE	割り出し位置のオプション

#### 17.7.1.2 軸/主軸別マシンデータ

番号	識別子: \$MA_	説明
30300	IS_ROT_AX	回転軸
30310	ROT_IS_MODULO	回転軸のモジュロ変換
30320	DISPLAY_IS_MODULO	位置表示はモジュロ 360°
30500	INDEX_AX_ASSIGN_POS_TAB	軸は割り出し軸です
30501	INDEX_AX_NUMERATOR	等距離の位置にある割り出し軸の分子
30502	INDEX_AX_DENOMINATOR	等距離の位置にある割り出し軸の分母
30503	INDEX_AX_OFFSET	等距離の位置にある割り出し軸の割り出し位置
30505	HIRTH_IS_ACTIVE	Hirth tooth システムが有効

## 17.7.2 セッティングデータ

### 17.7.2.1 一般セッティングデータ

番号	識別子: \$SN_	説明
41050	JOG_CONT_MODE_LEVELTRIGGRD	ジョグモードで連続 JOG

## 17.7.3 信号

### 17.7.3.1 軸/主軸からの信号

信号名称	SINUMERIK 840D sl	SINUMERIK 828D
原点確立済み/原点同期済み 1、原点確立済み/原点同期済み 2	DB31, ... .DBX60.4/5	DB390x.DBX0.4/5
割り出し軸はインポジション	DB31, ... .DBX76.6	DB390x.DBX1002.6

## 17.7.4 システム変数

識別子	説明
\$AA_ACT_INDEX_AX_POS_NO[軸]	最後に到達したか通過した割り出し位置の番号
\$AA_PROG_INDEX_AX_POS_NO[軸]	プログラム指令割り出し位置番号



## W4:研削用工具オフセットと工具監視

### 18.1 研削用工具データ

#### 18.1.1 工具データの構造

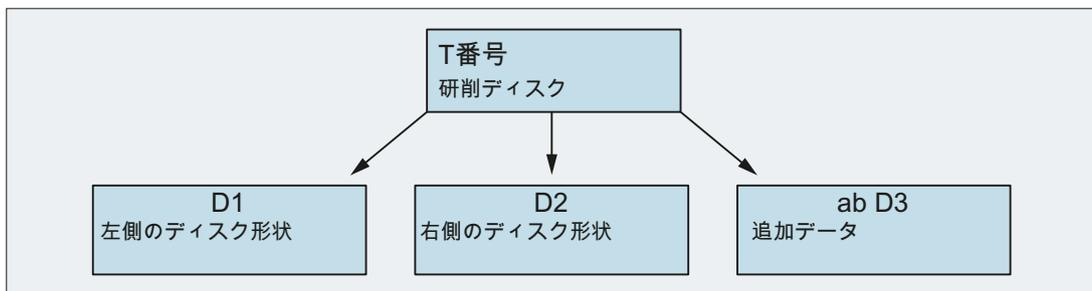
研削工具(工具タイプ:400 ~ 499)には、刃先データの他に、工具と目立て工具の個別のデータが含まれます。

左右のといし形状に関するといし別データは、工具刃先 D1 と D2 の T 番号に保存できます。

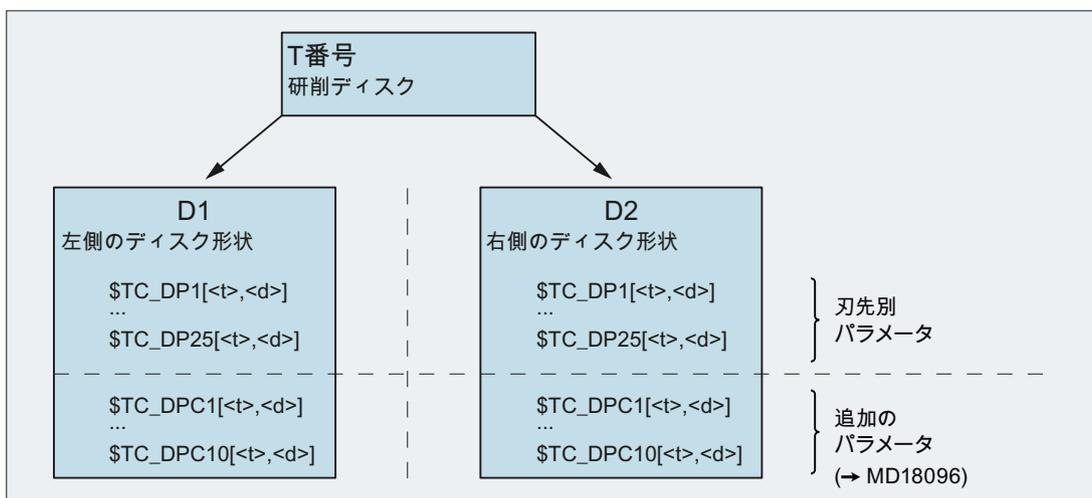
18.1 研削用工具データ

目立て形状のデータが必要な場合、以下のように保存できます。

- T 番号の D3 からの工具刃先に保存



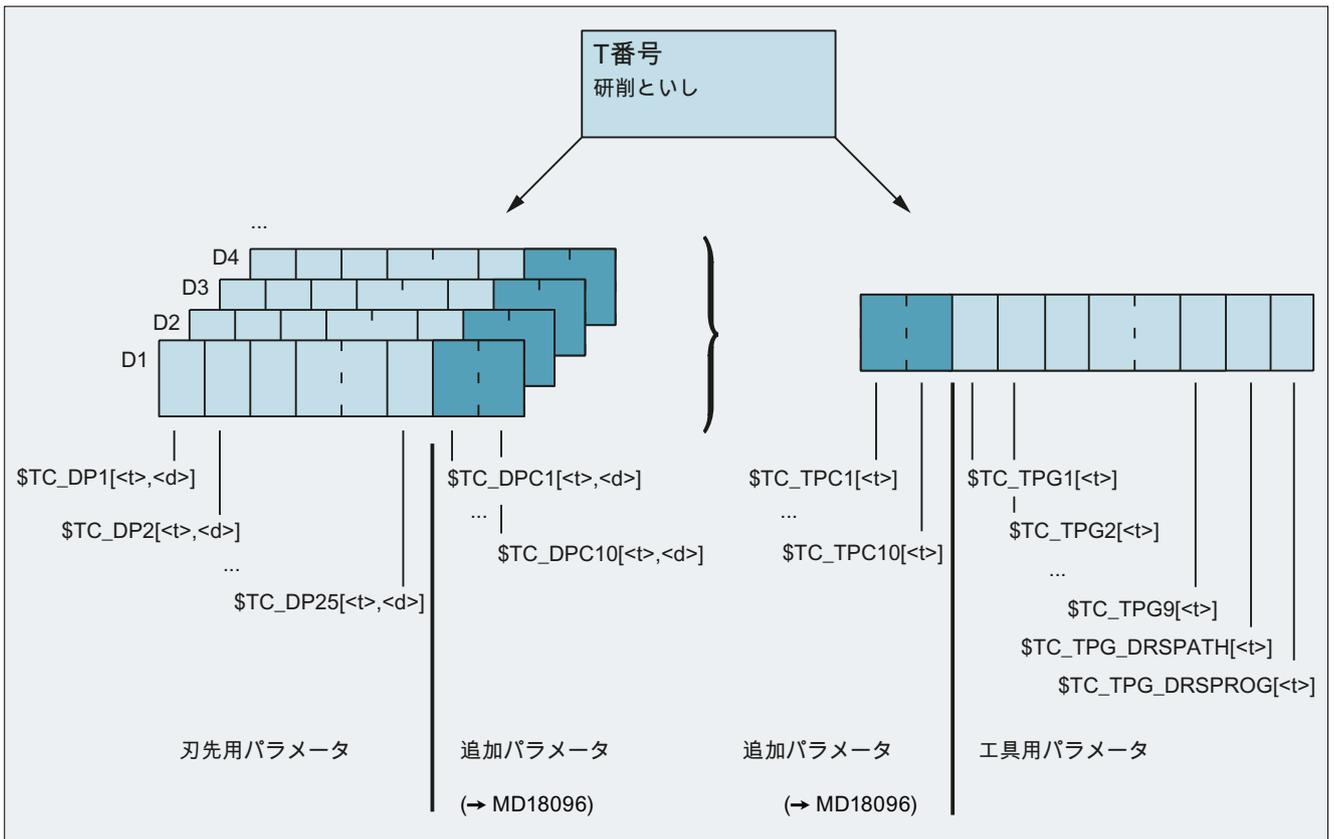
- 追加の刃先用パラメータに保存( MD18096 \$MN\_MM\_NUM\_CC\_TOA\_PARAM を参照)



<t>: T 番号

<d>: D 番号

以下の図に、研削工具の工具データの保存構造(例)の概要を示します。



<t>: T 番号

<d>: D 番号

図 18-1 研削工具の工具データの保存構造

## 18.1.2 刃先用パラメータ

### 18.1.2.1 刃先用パラメータのリスト

研削工具用の工具パラメータは、旋盤とフライス盤のパラメータと同じ意味があります。

工具パラメータ	意味	コメント	
\$TC_DP1	工具タイプ		
\$TC_DP2	刃先位置	旋盤工具と研削工具の	み
形状 - 工具長補正			

## 18.1 研削用工具データ

工具パラメータ	意味	コメント	
\$TC_DP3	長さ 1		
\$TC_DP4	長さ 2		
\$TC_DP5	長さ 3		
形状 - 工具径補正			
\$TC_DP6	半径 1		
\$TC_DP7			当社予約
\$TC_DP8			当社予約
\$TC_DP9			当社予約
\$TC_DP10			当社予約
\$TC_DP11			当社予約
摩耗 - 工具長補正			
\$TC_DP12	長さ 1		
\$TC_DP13	長さ 2		
\$TC_DP14	長さ 3		
摩耗 - 工具径補正			
\$TC_DP15	半径 1		
\$TC_DP16			当社予約
\$TC_DP17			当社予約
\$TC_DP18			当社予約
\$TC_DP19			当社予約
\$TC_DP20			当社予約
ベース寸法/アダプタ寸法 - 工具長補正			
\$TC_DP21	基本長さ 1		
\$TC_DP22	基本長さ 2		
\$TC_DP23	基本長さ 3		
テクノロジー			

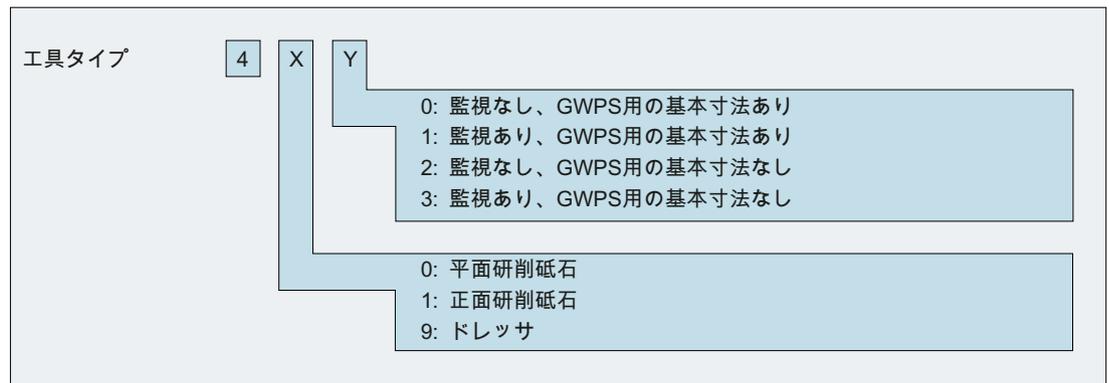
工具パラメータ	意味	コメント	
\$TC_DP24	アンダーカット角度	旋盤のみ	
\$TC_DP25			当社予約

### 注記

選択した研削工具の D1 と D2 の刃先データは連動できます。たとえば D1 か D2 のパラメータを変更すると、D1 か D2 の同一パラメータが新しい値で自動的に上書きされます (\$TC\_TPG2 (ページ 1023)を参照してください)。

### 18.1.2.2 \$TC\_DP1

パラメータ\$TC\_DP1 には、研削工具タイプの 3 桁の数字が含まれます。



番号	研削工具タイプ
400	平面研削といし
401	GWPS の工具ベース寸法による監視付き平面研削といし
402	GWPS の工具ベース寸法によらない監視なし平面研削といし
403	GWPS の工具ベース寸法によらない監視付き平面研削といし
410	正面研削といし
411	GWPS の工具ベース寸法による監視付き正面研削といし
412	GWPS の工具ベース寸法によらない監視なし正面研削といし
413	GWPS の工具ベース寸法によらない監視付き正面研削といし
490	といし目立て工具

18.1 研削用工具データ

18.1.2.3 追加パラメータの定義\$TC\_DPC1 ... 10

ユーザー別刃先データでは、工具タイプとは無関係に、次のメモリ構成マシンデータを使用して追加パラメータ\$TC\_DPC1 ~\$TC\_DPC10 を設定できます。

MD18096 \$MN\_MM\_NUM\_CC\_TOA\_PARAM

<b>通知</b>
<p><b>再構成によるデータ消失</b></p> <p>ローカルのスタティック NC メモリを再構成すると、アクティブファイルシステムとパッシブファイルシステムのデータが消失します。よって、修正したメモリ構成を有効にする前に、セットアップアーカイブを作成してすべての関連するデータをアーカイブするかバックアップすることを強く推奨します。</p>

18.1.3 工具用パラメータ

18.1.3.1 工具用パラメータのリスト

工具用パラメータは、すべての新規研削工具(工具タイプ:400 ~ 499)で自動的に設定されます。

**注記**

工具用パラメータには、刃先と同じ特徴があります。

刃先数を指定するときは、これを考慮してください。

MD18100 \$MN\_MM\_NUM\_CUTTING\_EDGES\_IN\_TOA

工具のすべての刃先を削除すると、関連する工具用パラメータも同時に削除されます。

**パラメータ**

パラメータは次のように割り当てます。

パラメータ	意味	データ形式
\$TC_TPG1 (ページ 1023)	主軸番号	INT
\$TC_TPG2 (ページ 1023)	連動規則	INT
\$TC_TPG3 (ページ 1025)	最小といし半径	REAL
\$TC_TPG4 (ページ 1025)	最小といし幅	REAL

パラメータ	意味	データ形式
\$TC_TPG5 (ページ 1026)	現在のといし幅	REAL
\$TC_TPG6 (ページ 1026)	最高速度	REAL
\$TC_TPG7 (ページ 1026)	最大周速度	REAL
\$TC_TPG8 (ページ 1026)	傾斜といしの角度	REAL
\$TC_TPG9 (ページ 1027)	半径計算用パラメータ番号	INT
\$TC_TPG_DRSPATH (ページ 1027)	目立てプログラムへのディレクトリパス	STRING
\$TC_TPG_DRSPROG (ページ 1027)	目立てプログラム名称	STRING

### 注記

データ監視は、研削といしの左右の刃先に適用されます。

工具用パラメータは、以下の場合に有効になります。

- 研削用工具監視 (ページ 1045)を有効にした場合
- といし周速度一定制御(GWPS) (ページ 1048)を有効にした場合

変更した期日を有効にするには、これらの機能を再度有効にしてください。

長補正は、直交座標の工具キャリアレファレンス点と工具先端の間の距離を常に指定します(傾斜研削といしでは注意が必要です)。

### 18.1.3.2 \$TC\_TPG1

主軸番号をパラメータ\$TC\_TPG1に入力します。これで、この主軸の砥石周速度が監視されます。

### 18.1.3.3 \$TC\_TPG2

2つの刃先 D1 および D2 の工具パラメータをパラメータ\$TC\_TPG2に入力します。これらの値は、常に同じでなければなりません。これらの刃先のパラメータのいずれかが変更されると、もう一方の対応するパラメータが自動的に、同じ値に設定されます。

工具パラメータのコードは、パラメータ\$TC\_TPG2に入力します。

工具パラメータ	意味	コード化		
		ビット	16進数	10進数
\$TC_DP1	工具タイプ	0	0001	1
\$TC_DP2	刃先長	1	0002	2

## 18.1 研削用工具データ

工具パラメータ	意味	コード化		
		ビット	16進数	10進数
形状 - 工具長補正				
\$TC_DP3	長さ 1	2	0004	4
\$TC_DP4	長さ 2	3	0008	8
\$TC_DP5	長さ 3	4	0010	16
\$TC_DP6	半径	5	0020	32
\$TC_DP7	当社予約	6	0040	64
\$TC_DP8	長さ 4	7	0080	128
\$TC_DP9	長さ 5	8	0100	256
\$TC_DP10	角度 1 など	9	0200	512
\$TC_DP11	当社予約	10	0400	1 024
摩耗 - 工具長補正				
\$TC_DP12	長さ 1	11	0800	2 048
\$TC_DP13	長さ 2	12	1000	4 096
\$TC_DP14	長さ 3	13	2000	8 192
\$TC_DP15	半径	14	4000	16 384
\$TC_DP16	当社予約	15	8000	32 768
\$TC_DP17	磨耗長さ 4	16	1 0000	65 536
\$TC_DP18	磨耗長さ 5	17	2 0000	131 072
\$TC_DP19	摩耗角度 1 など	18	4 0000	262 144
\$TC_DP20	当社予約	19	8 0000	524 288
ベース寸法/アダプタ寸法 - 工具長補正				
\$TC_DP21	基本長さ 1	20	10 0000	1 048 576
\$TC_DP22	基本長さ 2	21	20 0000	2 097 152
\$TC_DP23	基本長さ 3	22	40 0000	4 194 304
テクノロジー				
\$TC_DP24	当社予約	23	80 0000	8 388 608
\$TC_DP25	当社予約	24	100 0000	16 777 216

**変更が有効になるタイミング**

パラメータ\$TC\_TPG2 が、後の時点で有効になる場合、刃先の工具パラメータの値は、すぐに自動的に調整されません。調整されるのは、両方の刃先の工具パラメータの次回の変更時です。

**例**

検索工具(T1)では、刃先 D1 と D2 で、次のパラメータを同じ値のままにしてください。

工具パラメータ	意味	コード表示:16 進数	コード表示:10 進
\$TC_DP1	工具タイプ	0001	1
形状 - 工具長補正			
\$TC_DP3	長さ 1	0004	4
\$TC_DP4	長さ 2	0008	8
\$TC_DP5	長さ 3	0010	16
摩耗 - 工具長補正			
\$TC_DP12	長さ 1	0800	2 048
\$TC_DP13	長さ 2	1000	4 096
\$TC_DP14	長さ 3	2000	8 192
ベース寸法/アダプタ寸法 - 工具長補正			
\$TC_DP21	基本長さ 1	10 0000	1 048 576
\$TC_DP22	基本長さ 2	20 0000	2 097 152
\$TC_DP23	基本長さ 3	40 0000	4 194 304

すべてのパラメータコードの合計をパラメータ\$TC\_TPG2 に入力します。

\$TC_TPG2 =	'H70381D'	'D7354397'
-------------	-----------	------------

**18.1.3.4 \$TC\_TPG3, \$TC\_TPG4**

研削用工具監視 (ページ 1045)をおこなうには、砥石の下限値をパラメータに入力します。

- パラメータ\$TC\_TPG3:半径[mm]または[inch]
- パラメータ\$TC\_TPG4:幅[mm]または[inch]

注記

傾斜研削砥石では、最小半径を直交座標に入力します。

18.1.3.5 \$TC\_TPG5

砥石の実際の幅(目立ての後など)をパラメータ\$TC\_TPG5に入力します。

18.1.3.6 \$TC\_TPG6 および\$TC\_TPG7

研削用工具監視 (ページ 1045)を行うには、砥石の上限値をパラメータに入力します。

- パラメータ\$TC\_TPG6:速度[rpm]
- パラメータ\$TC\_TPG7:周速度[m\*s<sup>-1</sup>]または[インチ\*s<sup>-1</sup>]

18.1.3.7 \$TC\_TPG8

傾斜研削砥石の「砥石周速度一定(GWPS) (ページ 1048)」機能では、実際の有効な平面(作業平面 G17、G18、G19)を基準とした傾斜角度  $\alpha$  をパラメータ\$TC\_TPG8に入力します。

傾斜角度  $\alpha$  の許容範囲は、 $-90^\circ \leq \alpha < +90^\circ$ です。

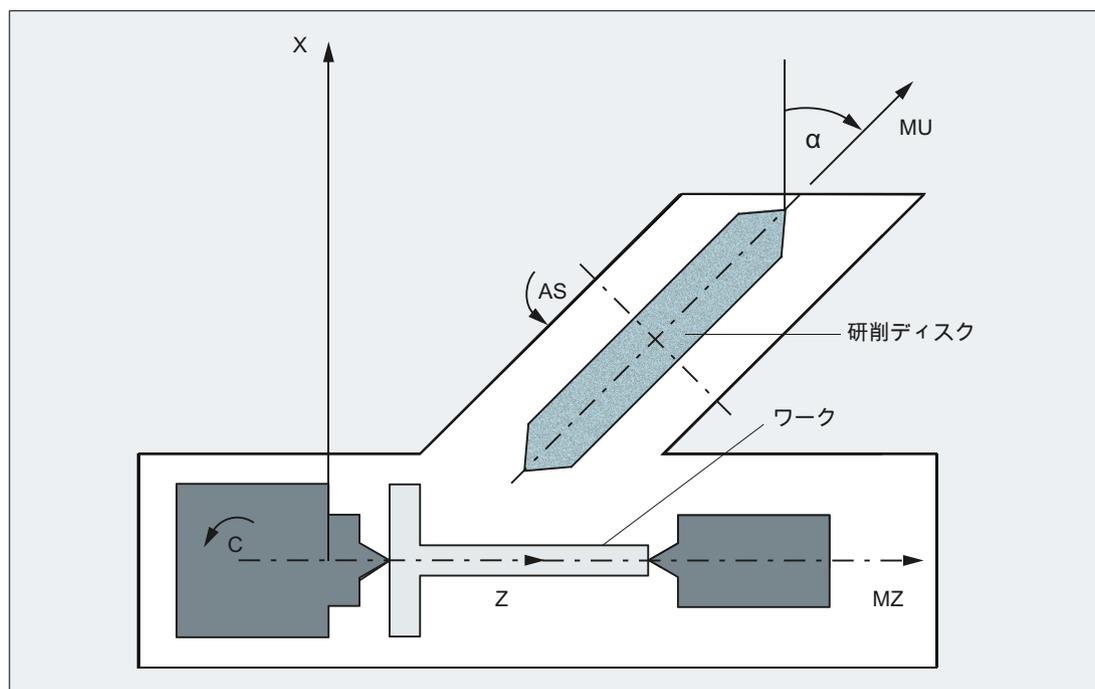


図 18-2 傾斜研削砥石

**注記**

工具長は、パラメータ\$TC\_TPG8の角度が変更されたときに自動的に補正されません。

**注記**

傾斜軸を含む機械では、傾斜軸と傾斜砥石に、同じ角度を入力してください。

**18.1.3.8 \$TC\_TPG9**

パラメータ\$TC\_TPG9を使用して、次の機能が参照する工具パラメータを定義します。

- 研削用工具監視 (ページ 1045)
- 砥石周速度一定(GWPS) (ページ 1048)

例:

\$TC\_TPG9 = <値>

<値>	基準変数
3	長さ 1 (工具タイプに応じた、形状 + 磨耗 + 基本)
4	長さ 2 (工具タイプに応じた、形状 + 磨耗 + 基本)
5	長さ 3 (工具タイプに応じた、形状 + 磨耗 + 基本)
6	半径

**18.1.3.9 \$TC\_TPG\_DRSPATH および\$TC\_TPG\_DRSPROG**

パラメータ\$TC\_TPG\_DRSPATH (目立てプログラムへのディレクトリパス)および\$TC\_TPG\_DRSPROG (目立てプログラム)を使用すると、目立てプログラムを砥石に割り当てることができます。これにより、一般的に有効な目立て手順を作成し、この手順で、プログラムの間接呼び出しを使用して目立てプログラムを実行することができます。

例:

プログラムコード	コメント
...	
	; アブソリュート目立て量を\$TC_DP12に入力:
\$TC_DP12[GETT("GRINDING_WHEEL",0),1]= \$TC_DP12[GETT("GRINDING_WHEEL",0),1] - 0.01	

18.1 研削用工具データ

プログラムコード	コメント
G500	
CYCLE435 ("GRINDING_WHEEL", 1, "DRESSER", 1)	; 目立て工具位置の計算
T="DRESSER" D1	; 目立て工具の有効化。
	; 目立てプログラムの間接呼び出し:
CALL "/"<<\$TC_TPG_DRSPATH[GETT("GRINDING_WHEEL", 0)]<<"/"<<\$ TC_TPG_DRSPROG[GETT("GRINDING_WHEEL", 0)]	
M17	

18.1.3.10 追加パラメータの定義\$TC\_TPC1 ... 10

ユーザー別刃先データでは、工具タイプとは無関係に、次のメモリ構成マシンデータを使用して、追加パラメータ\$TC\_TPC1 ~\$TC\_TPC10 を設定できます。

MD18096 \$MN\_MM\_NUM\_CC\_TDA\_PARAM

<b>通知</b>
<b>再構成によるデータ消失</b>
ローカルのスタティック NC メモリを再構成すると、アクティブファイルシステムとパッシブファイルシステムのデータが消失します。よって、修正したメモリ構成を有効にする前に、セットアップアーカイブを作成してすべての関連するデータをアーカイブするかバックアップすることを強く推奨します。

18.1.3.11 工具用パラメータへのアクセス

工具用パラメータの読み取りと書き込みをパートプログラムから実行できます。

例:

プログラムコード	コメント
R10=\$TC_TPG5[2]	; 工具 2 の現在の幅を読み取り、R10 に保存
\$TC_TPG6[3] = 2000	; 工具 3 の最大速度に値 2000 を書き込む

以下のシステム変数で、**現在の**工具の工具用パラメータをアドレス指定します。

\$P\_ATPG[<m>]

<m> = パラメータ番号(データタイプ:REAL)

例:

パラメータ 3 (\$TPG3[<t>])

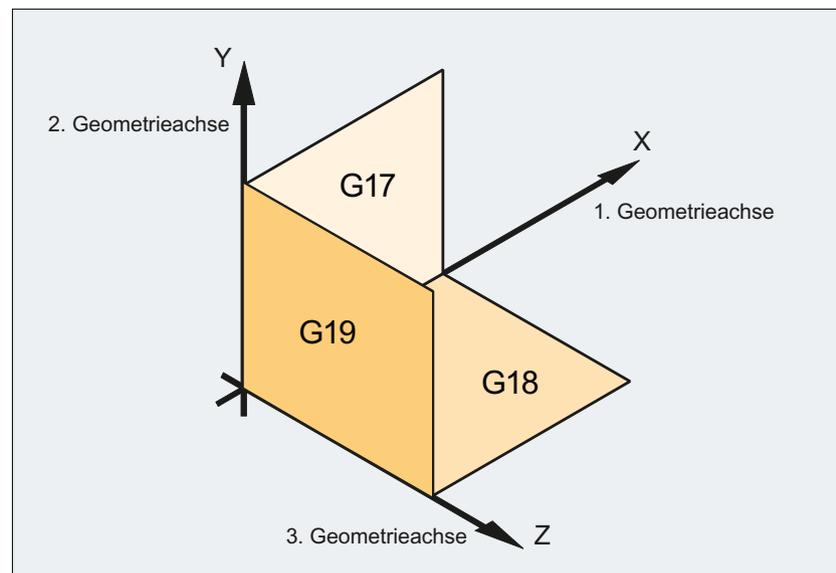
プログラムコード	コメント
\$P_ATPG[3]=R10	; 現在の工具の最小といし半径を読み取り、R10 に保存します。

### 18.1.4 平面と軸の割り当て

平面のジオメトリ軸の工具長補正または半径補正は、現在の加工平面を基準にして割り当てます。

次の平面と軸の割り当てが可能です(第 1、第 2、第 3 ジオメトリ軸の横座標、縦座標、垂直座標)。

命令	平面 (横座標/縦座標)	平面に直交する軸 (垂直座標)
G17	X / Y	Z
G18	Z / X	Y
G19	Y / Z	X



18.1 研削用工具データ

18.1.5 例

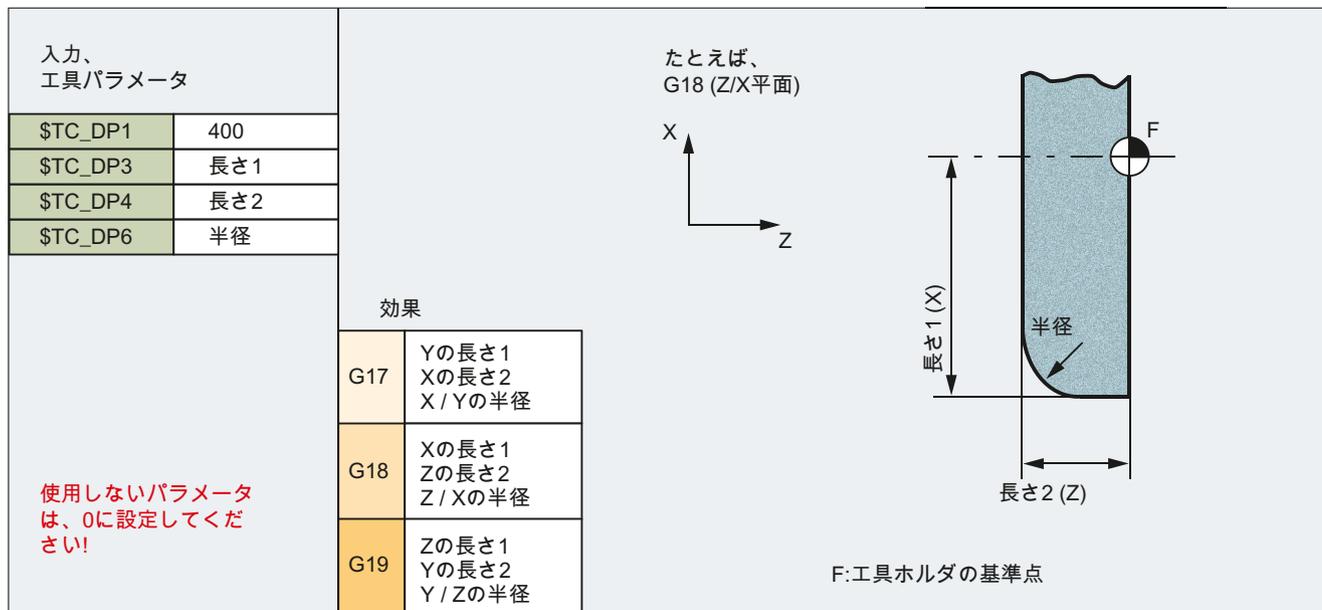


図 18-3 平面研削といしの必要オフセットデータ

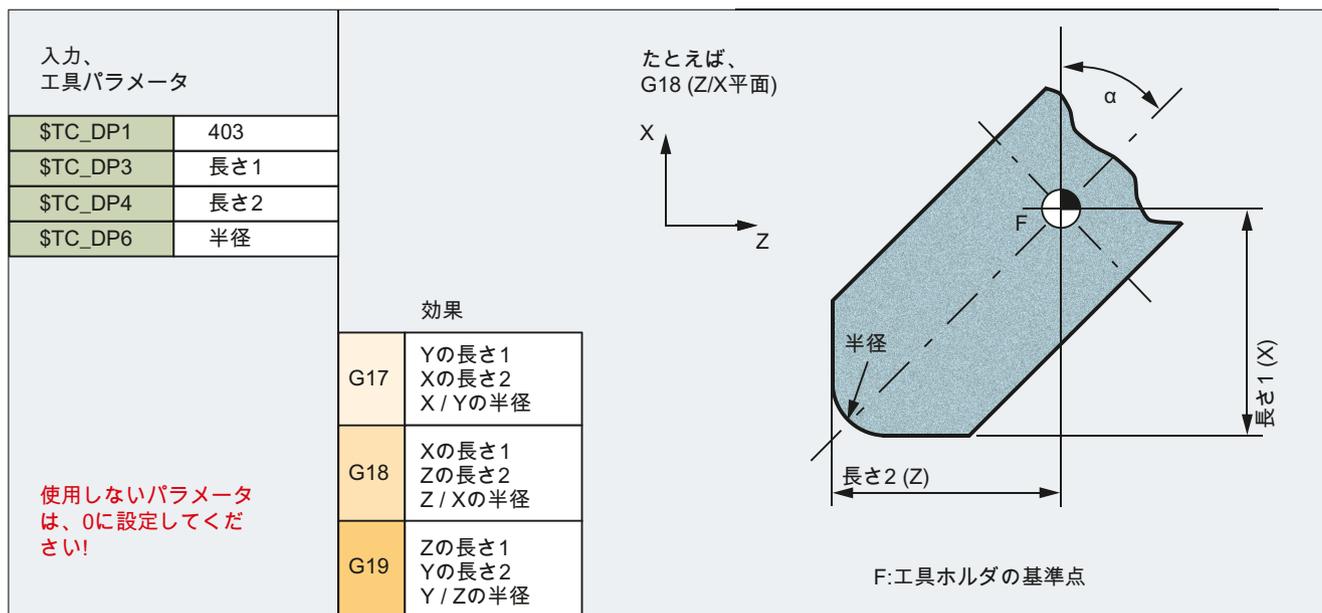


図 18-4 自動監視を選択し、GWPS 用の基本寸法なしの傾斜研削といしの必要オフセットデータ

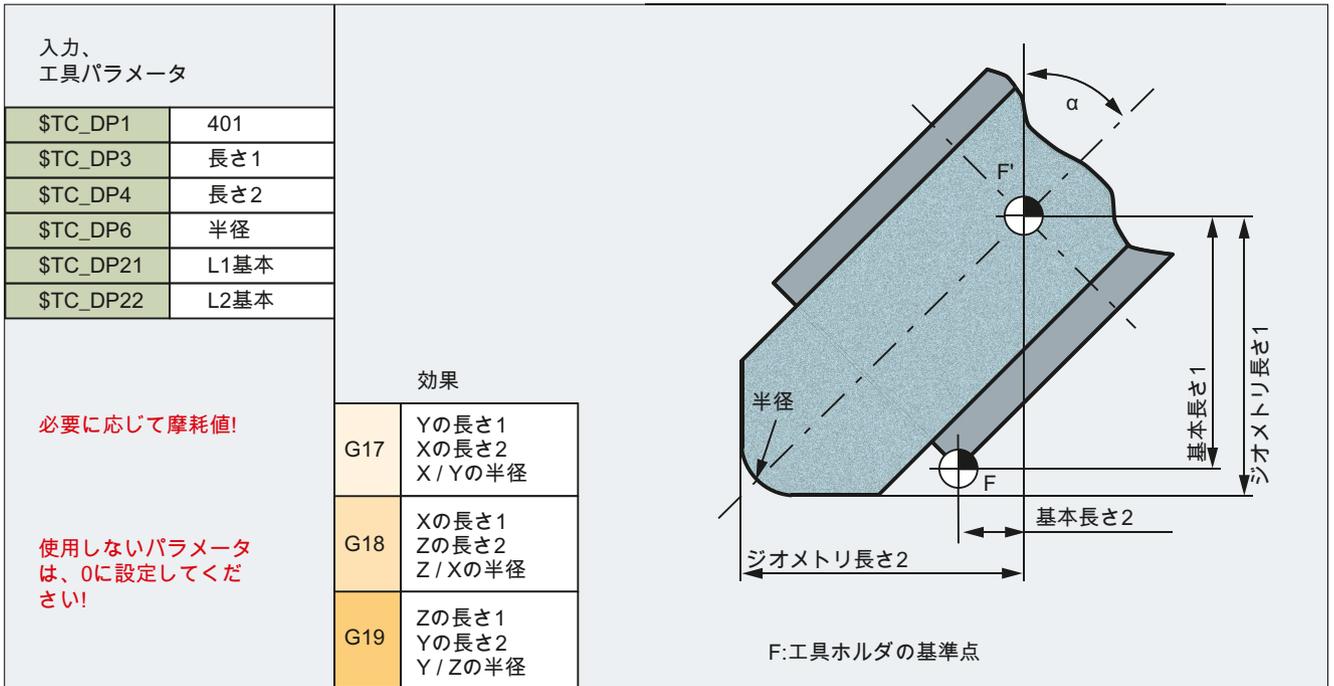


図 18-5 自動監視を選択し、GWPS 用の基本寸法ありの傾斜研削といしの必要オフセットデータ

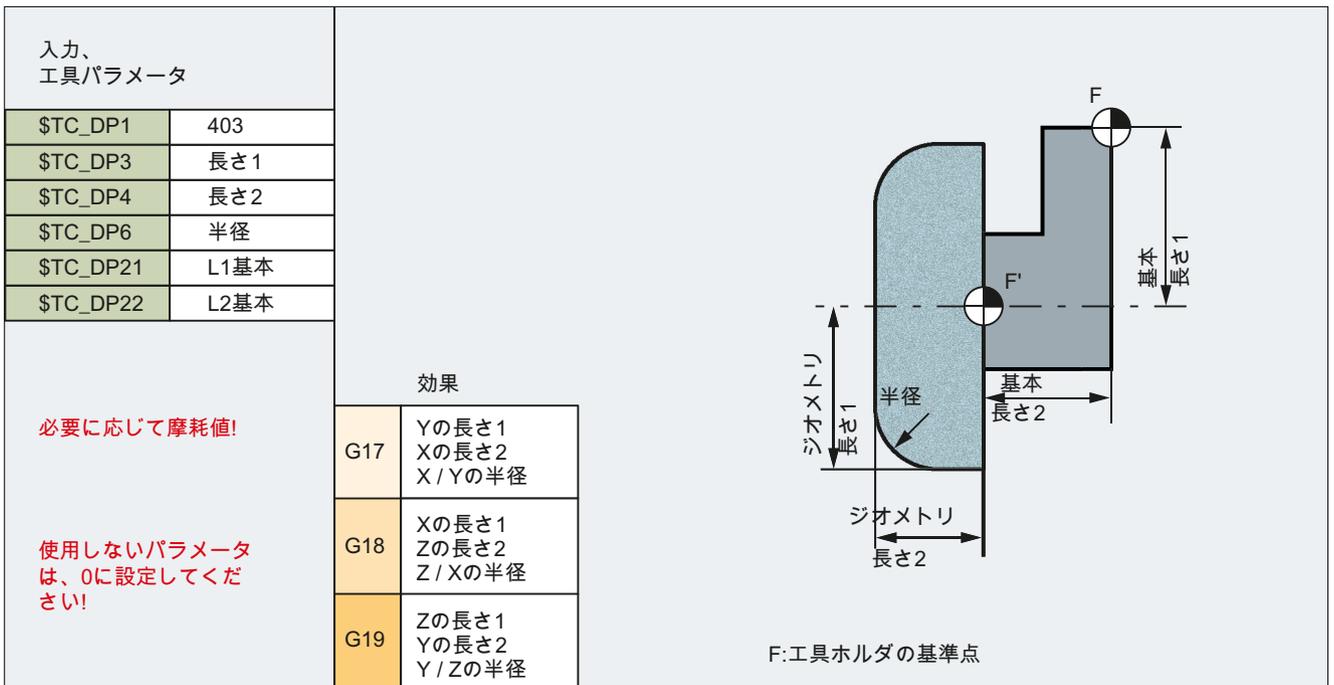


図 18-6 GWPS 用の基本寸法なしの平面研削といしの必要オフセットデータ

18.2 オンライン工具オフセット

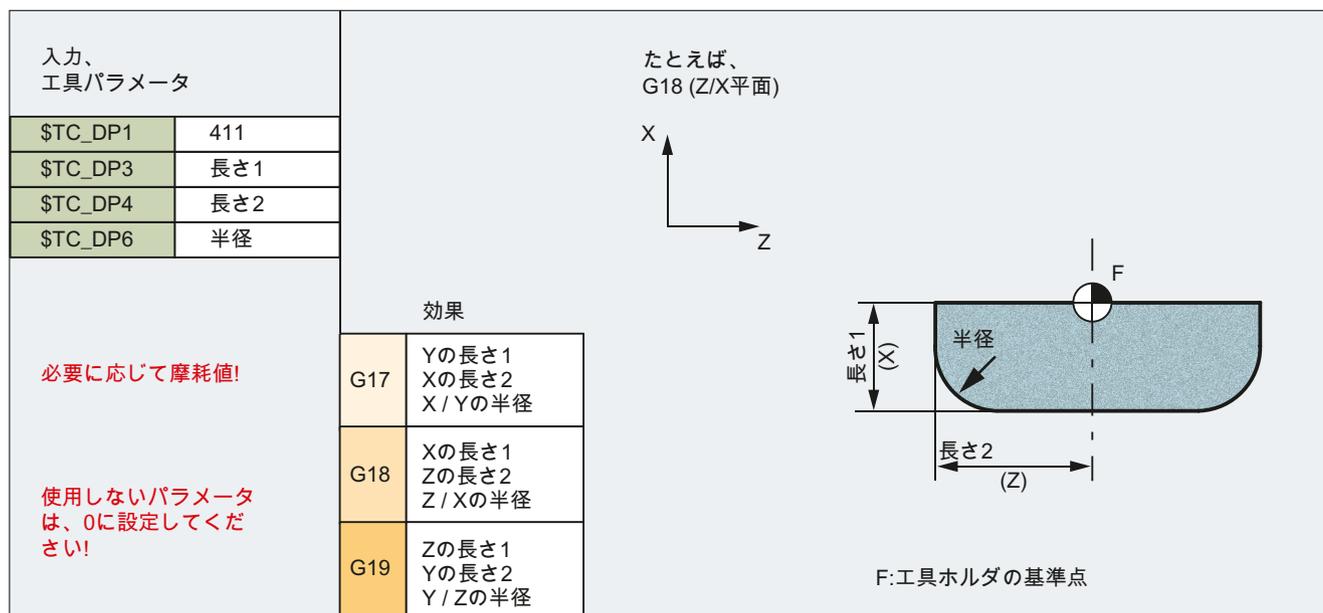


図 18-7 パラメータ監視をおこなう正面研削の必要オフセットデータ

## 18.2 オンライン工具オフセット

### 18.2.1 機能

研削運転には、ワークの加工と研削砥石の目立てがあります。この処理は、同一チャンネルでも別のチャンネルでも実行できます。

研削砥石の目立て中に加工を継続できるようにするには、目立てによる研削砥石のサイズ縮小を工具オフセットとして加工チャンネルの現在の工具に伝送して、すぐに適用してください。これは、「オンライン工具オフセット」機能を使用して実行できます。

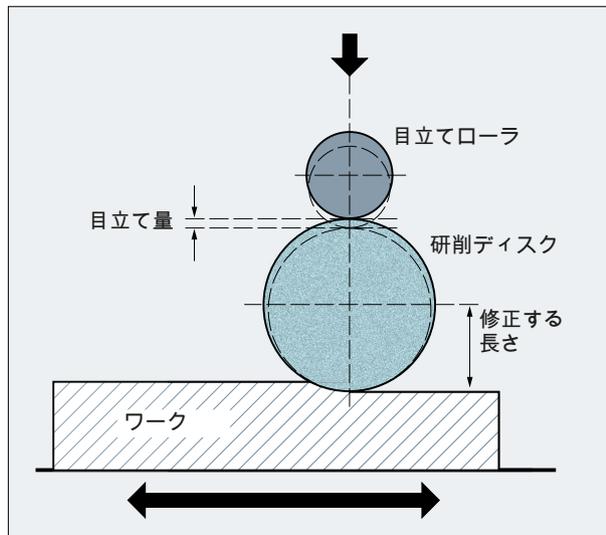


図 18-8 目立てローラを使用した加工中の目立て

この機能は、工具オフセットの書き込みの以下の方法をサポートします。

- 連続書き込み、ノンモーダル(PUTFTOCF (ページ 1037))  
PUTFTOCF を使用すると、目立てを加工と同時に実行します。  
工具補正は、多項式関数の 1 番目、2 番目、または 3 番目の順序に従って、加工中のチャンネルで連続的に変更されます。この機能は、事前に FCTDEF (ページ 1035) で定義しておいてください。  
PUTFTOCF は常にノンモーダルです。つまり、次の移動ブロックで有効になります。
- 連続書き込み、モーダル:ID=1 DO FTOC (詳細情報については、『ファンクションマニュアル』の「シンクロナイズドアクション」を参照)
- 個別書き込み(PUTFTOC (ページ 1038))  
PUTFTOC を使用すると、目立ては並列チャンネルの加工と同時に実行されません。  
PUTFTOC で指定した訂正值は、対象チャンネルですぐに有効になります。

#### 起動と解除

パートプログラムのオンライン工具オフセットは、G 命令 FTOCON/FTOCOF (ページ 1039) で有効化/無効化します。

## 18.2 オンライン工具オフセット

### その他プロパティ

- オンライン工具オフセットは、すべてのチャンネルのあらゆる研削工具(工具タイプ:400 ~ 499)で有効にすることができます。
- オンライン工具オフセットは、一般的には長補正として適用します。長さは、形状と磨耗データと同じように、工具タイプに応じて、現在の平面上のジオメトリ軸に割り当てます。
- オンライン工具オフセットを選択したとき、研削主軸監視機能は、そのまま有効です。

---

### 注記

オンライン工具オフセットは研削工具のみに使用できます。

このオンライン工具オフセットでは、選択した長さの磨耗パラメータが常に修正されません。長さ補正が複数の刃先で同一である場合は、連動指定を使用して第2刃先の値を自動的に修正してください。

オンライン工具オフセットが加工チャンネルで有効である場合は、このチャンネルで有効な工具の磨耗値を加工プログラムまたはオペレータ入力によって変更できません。

半径磨耗(P15)の変更は、工具を再選択するまで有効ではありません。

オンライン工具オフセットは、といし周速度一定制御(GWPS)にも適用されます。たとえば主軸速度は、対応する値で修正されます。

---

## 18.2.2 セットアップ

### 定義可能な工具ホルダ

「オンライン工具オフセット」機能は、「旋回可能な工具ホルダ」と同時に使用することはできません。

「オンライン工具オフセット」機能の使用時には、TO 領域で定義できる工具ホルダの最大数を「0」に設定してください。

MD18075 \$MN\_MM\_NUM\_TOOLHOLDERS = 0

---

### 注記

SINUMERIK 828D では、「研削」テクノロジーでのみ、MD18075 の値を設定できます。

---

## オンライン工具径補正を有効化

さらに、オンライン工具径補正 (ページ 1044) も、次のマシンデータで有効にしてください。

MD20254 \$MC\_ONLINE\_CUTCOM\_ENABLE = 1

## 18.2.3 プログラミング

### 18.2.3.1 多項式機能(FCTDEF)の定義

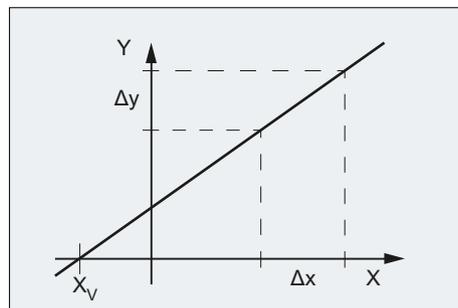
特定の目立て方法(目立てローラなど)には、目立てローラを送り込むと研削砥石の半径が連続して(直線的に)縮小するという特徴があります。この方法では、目立てローラの切り込みとそれぞれの長さの磨耗値の書き込みの間に一次関数が必要となります。直線機能は、予約手順 FCTDEF(...)を使用して三次多項式機能まで定義されます。

#### 直線方程式

$$y = f(x) = a_0 + a_1 \cdot x_1$$

$a_1$ :直線の傾き、 $a_1 = \Delta x / \Delta y$

$a_0$ : $a_0 = -a_1 \cdot X_v$  の場合の X 軸に沿った直線の移動



#### 構文

FCTDEF (<Func>, <LLimit>, <ULimit>, <a0>, <a1>, <a2>, <a3>)

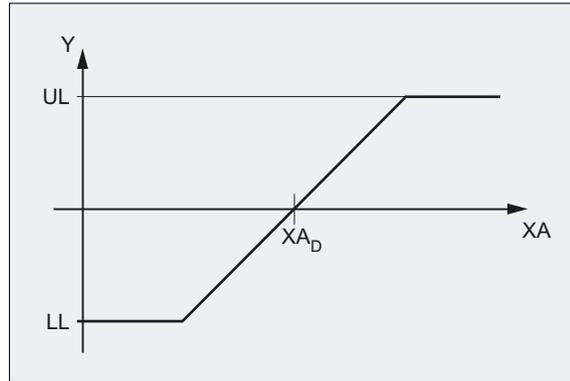
## 意味

FCTDEF (...):	PUTFTOCF(...)の多項式機能の定義: $y = f(x) = a_0 + a_1 \cdot x + a_2 \cdot x^2 + a_3 \cdot x^3$	
<Func>:	ファンクション番号	
	データタイプ:	INT
	値の範囲:	1, 2, 3
<LLimit>:	下限値	
	データタイプ:	REAL
<ULimit>:	上限値	
	データタイプ:	REAL
<a0>, <a1>, <a2>, <a3>:	多項式関数の係数です。	
	データタイプ:	REAL

## 例

## 定義

- ファンクション番号:1
- 下限値および上限値: -100、100
- 特性の変化率: $a_1 = 1$
- 動作点は特性の中心に位置する必要があります。機能が NC プログラムで定義される場合の WCS の軸 XA のセットポイント位置に基づいて、特性を負の Y 方向に移動してください。 $a_0 = -a_1 \cdot XA_D = -1 \cdot \$AA\_IW$
- $a_2 = a_3 = 0$

**Characteristic**

UL 上限値

LL 下限値

XA<sub>D</sub> 機能が NC プログラムで定義されている場合の軸 XA のセットポイント**プログラミング**

プログラムコード	コメント
FCTDEF (1, -100, 100, -\$AA_IW[XA], 1)	; 機能の定義

**18.2.3.2 オンライン工具補正の連続書き込み(PUTFTOCF)**

予約手順 PUTFTOCF(...)を使用して、オンライン工具補正が FCTDEF(...) (ページ 1033) で以前に定義した多項式機能に基づいて実行されます。

**注記**

オンライン工具オフセットをシンクロナイズドアクションを使用して実現することもできます。

詳細情報については、『ファンクションマニュアル』の「シンクロナイズドアクション」を参照。

**構文**

```
PUTFTOCF (<Func>, <RefVal>, <ToolPar>, <Chan>, <Sp>)
```

18.2 オンライン工具オフセット

意味

PUTFTOCF(...):	FCTDEF(...)で定義した多項式機能を使用して、オンライン工具補正をブロックごとに連続書き込み	
<Func>:	FCTDEF(...)による機能定義で定義された機能番号	
	データタイプ:	INT
	値の範囲:	1, 2, 3
<RefVal>:	オフセットの算出基準となる指令値です(軸のセットポイントなど)。	
	データタイプ:	VAR REAL
<ToolPar>:	オフセット値がインクルードされる摩耗パラメータ(長さ 1、2、または 3)の番号です。	
	データタイプ:	INT
<Chan>:	オンライン工具補正が有効となるチャンネルの番号です。 <b>注記:</b> オフセットが有効なチャンネルで無効な場合にのみ必要です。	
	データタイプ:	INT
<Sp>:	オンライン工具補正が有効となる主軸の番号です。 <b>注記:</b> オフセットが現在使用されている有効な工具ではなく、無効などいしに適用される場合にのみ必要です。	
	データタイプ:	INT

18.2.3.3 オンライン工具補正の書き込み、個別(PUTFTOC)

機能

予約手順 PUTFTOC(...)を使用して、オンライン工具補正が固定のオフセット値に基づいて実行されます。

構文

PUTFTOC (<CorrVal>, <ToolPar>, <Chan>, <Sp>)

## 意味

PUTFTOC (...):	オンライン工具補正の書き込み	
<CorrVal>:	摩耗パラメータに追加されるオフセット値。	
	データタイプ:	REAL
<ToolPar>:	オフセット値がインクルードされる摩耗パラメータ(長さ 1、2、または 3)の番号です。	
	データタイプ:	INT
<Chan>:	オンライン工具補正が有効となるチャンネルの番号です。 <b>注記:</b> オフセットが有効なチャンネルで無効な場合にのみ必要です。	
	データタイプ:	INT
<Sp>:	オンライン工具補正が有効となる主軸の番号です。 <b>注記:</b> オフセットが現在使用されている有効な工具ではなく、無効などいしに適用される場合にのみ必要です。	
	データタイプ:	INT

## 18.2.3.4 オンライン工具オフセットの起動/解除(FTOCON/FTOCOF)

オンライン工具補正は、G 命令 FTOCON および FTOCOF を使用して有効または無効にできます。

## 構文

```
FTOCON
...
FTOCOF
```

## 18.2 オンライン工具オフセット

## 意味

FTOCON:	オンライン工具補正を起動します 命令は、オンライン工具補正が有効になるチャンネルでプログラミングしてください。
FTOCOF:	オンライン工具長補正の解除 命令は、オンライン工具補正が無効になるチャンネルでプログラミングしてください。 <b>注記:</b> FTOCOF では、軸は工具オフセットを超えて移動しません。ただし、PUTFTOC/PUTFTOCF で計算された値は切削固有オフセットデータに残ります。 オンライン工具補正を最終的に無効にするには、工具(T...)を FTOCOF 後にもう一度選択/選択解除してください。

## 18.2.4 補足条件

**DRF オフセット**

定義された制限値(速度)で、プログラム指令の軸移動にオンライン工具オフセットを重畳します。

DRF オフセットとオンラインオフセットが軸で同時に有効である場合は、先に DRF オフセットが有効になります。

**加速マージン:**

有効なオンライン工具オフセットは、JOG 速度で移動します。この場合、最大加速度が許容されます。

FTOCON では次のチャンネル別マシンデータが有効です。

**MD20610 \$MC\_ADD\_MOVE\_ACCEL\_RESERVE**

加速度限界は重畳移動をすぐに実行できるような、この移動のためにあります。

**レファレンス点復帰**

有効なオンラインオフセットは、G74 によるレファレンス点復帰で削除されます。

**工具交換**

前回の工具交換または刃先変更から FTOCON が有効である場合、工具を交換すると、コントローラでは再同期による先読み停止処理が開始されます。

刃先の変更は先読み停止処理なしで実行できます。

---

#### 注記

工具交換は、T 番号の選択によりオンライン工具オフセットと組み合わせて実行できません。

M6 による工具交換は、オンライン工具オフセット機能と組み合わせて実行できません。

---

#### 加工平面と座標変換

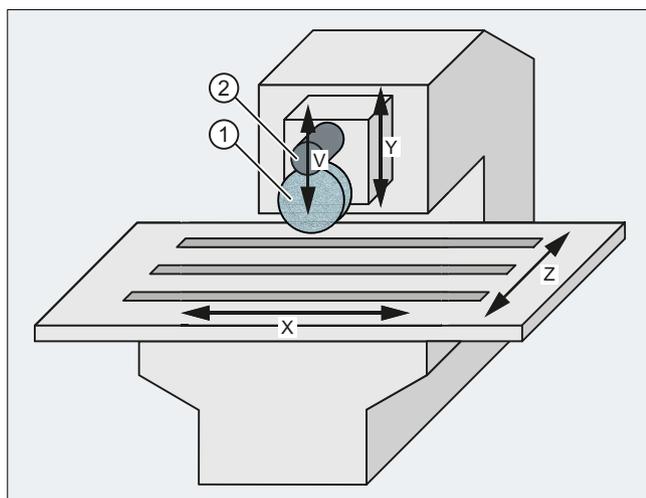
- FTCON は「傾斜軸」座標変換と組み合わせないと使用できません。
- FTCOF 状態である場合を除いて、FTCON が有効であるときに座標変換や平面(G17 から G18 など)を変更することはできません。

#### リセットと運転モードの変更

- オンラインオフセットが有効であるとき、NC-STOP および M2/M30 によるプログラム終了は、補正量の移動後に実行されます。
- オンライン工具オフセットは、NC-RESET に応じてすぐに選択解除されます。
- オンライン工具オフセットは、AUTOMATIC モードで、プログラムが有効であるときに有効にできます。

## 18.2.5 例

### 18.2.5.1 例:オンライン工具オフセットの連続書き込み



- ① 研削ディスク
- ② 目立てローラー
- X: 揺動軸
- Y: 切り込み軸:研削ディスク
- Z: テーブル軸
- V: 切り込み軸:目立てローラー

図 18-9 正面研削盤

#### 仕様

- (Tool Offset)
  - 加工平面:Y/Z 平面(G19)
  - 工具タイプ:401 (長さ1はZで、長さ2はYで動作)
- チャンネル 1:加工チャンネル、軸 X、Y、Z
- チャンネル 2:目立てチャンネル、軸 V

研削運転が Y100 で始まった後で、研削砥石を 0.05 で目立てしてください(V 方向)。目立て量は、オンラインオフセットにより連続で補正してください。

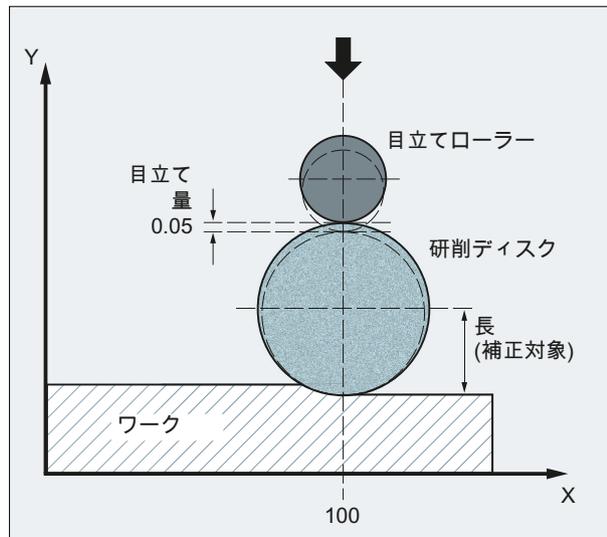


図 18-10 工具オフセット

## チャンネル 1 のプログラム(セクション):加工チャンネル

プログラムコード	コメント
...	
G1 G18 F10 G90	; 初期設定
T1 D1	; 実際の工具を選択します
S100 M3 X100	; スイッチオン主軸、x 軸を開始位置に移動
INIT (2, 「/_N_MPF_DIR/_N_ABRICHT_MPF」, 「S」)	; 選択:チャンネル 2 で「DRESS」をプログラム:
START (2)	; 開始:チャンネル 2
X200	; x 軸が目標位置
FTOCON	; オンライン補正を有効にします
...	

## チャンネル 2 のプログラム:目立てチャンネル

V 軸が Y 軸に平行して移動します。つまり、工具オフセットの長さ 3 が Y 軸方向に作用するため、補正が必要です。

プログラムコード	コメント
PROC ABRICHT	
FCTDEF (1, -1000, 1000, -\$AA_IW[V], 1)	; 機能の定義
PUTFTOCF(1, \$AA_IW[V], 3, 1)	; オンライン工具オフセットの連続書き込み: ; - 機能番号:1 ; - 指令値:v 軸の WCS のセットポイント位置 ; - 工具パラメータ:研削ディスクの長さ 3 ; - チャンネル:1 ; - 主軸:有効な研削ディスク
V-0.05 G1 F0.01 G91	; 目立ての v 軸の切り込み動作

## 18.3 オンライン工具径補正

プログラムコード	コメント
M30	

## 18.3 オンライン工具径補正

## 機能

工具の縦軸と輪郭が相互に直角である場合は、3つのジオメトリ軸のうちいずれかへオフセットを長補正として適用できます(オンライン工具長補正)。

この条件が満たされない場合は、実径補正值としてオフセット量を入力できます(オンライン工具径補正)。

## イネーブル

オンライン工具径補正は、次のマシンデータで有効にしてください。

```
MD20254 $MC_ONLINE_CUTCOM_ENABLE = 1
```

## 起動と解除

オンライン工具径補正は、G 命令 **FTOCON** および **FTOCOF** (ページ 1037)を使用して起動および解除します(オンライン工具長補正と同じ方法で)。

## パラメータ設定

オンライン工具オフセットは、**PUTFTOCF** (ページ 1035)または **PUTFTOC** (ページ 1036)を使用してパラメータ設定します。

以下のように、オンライン工具径補正にパラメータ<ToolPar> (訂正值を計算する摩耗パラメータ)を設定してください。

```
<ToolPar> = 4
```

## 境界条件

- 工具径補正およびオンライン工具径補正は、選択工具の半径が「0」以外のときのみ有効にすることができます。つまり、工具径補正のみで加工操作を実行することはできません。
- オンライン訂正值は元の半径より小さくして、オフセットが軸移動に重畳されたとき、許可されているダイナミック許容範囲を超えないようにしてください。
- 研削工具と回転工具(タイプ 400 ~ 599)では、訂正值は刃先位置に応じて適用されません。つまり刃先位置で指定された軸では、工具径補正が有効であるときは径補正として動作し、工具径補正が無効であるときは長さ補正として動作します。  
その他のすべての工具タイプの場合は、工具径補正を G41 または G42 で有効にしたときのみ訂正值が適用されます。工具径補正を G40 で無効にすると、訂正值はキャンセルされます。

## 18.4 研削用工具監視

### 18.4.1 機能

研削用工具監視は、形状および速度監視の組み合わせです。これは、すべてのチャンネルのあらゆる研削工具(工具タイプ:400 ~ 499)で有効にすることができます。

---

#### 注記

2つの工具が別々の主軸を基準にしている場合は、片方の工具で監視機能を選択していると、他方の工具の監視は選択解除されません。

主軸毎に1つの工具と1つの工具監視をいつでも有効にすることができます。

有効になった監視はリセット後も有効です。

---

## 形状監視

実際のといし半径と実際のといし幅を監視します。

## 18.4 研削用工具監視

パラメータ\$TC\_TPG9で参照される研削工具の最初の刃先(D1)補正值が、現在のといし半径として使用されます。監視が有効な場合、この値はパラメータ\$TC\_TPG3(最小といし半径)の値と比較されます。

---

**注記**

といし半径の監視機能は、オンライン工具オフセットの選択時に有効になります。

- 監視機能が有効であるとき
- 現在の半径(オンライン工具オフセット、磨耗パラメータ)または現在の幅(\$TC\_TPG5)が変更されたとき

---

現在のといし幅は通常は目立てサイクルで計算され、研削工具のパラメータ\$TC\_TPG5に入力できます。監視が有効な場合、この入力された値はパラメータ\$TC\_TPG4(最小といし幅)の値と比較されます。

現在のといし半径がパラメータ\$TC\_TPG3で指定された値を下回った場合、または現在のといし幅(\$TC\_TPG5)がパラメータ\$TC\_TPG4で定義された値を下回った場合、コントローラが以下のNC/PLCインターフェース信号を「1」に設定します。

DB31、... DBX83.3 (ジオメトリ監視) == 1

---

**注記**

コントローラ内部エラー応答なし。

---

**速度監視**

速度監視では、といし周速度(m/s) (パラメータ\$TC\_TPG7)および最大主軸速度(rpm) (パラメータ\$TC\_TPG6)が確認されます。

速度指令値が速度制限に対して主軸オーバライドの許容量を使用して周期的に監視されます。

速度制限は、最大といし周速度と現在のといし半径から算出した速度を、最大主軸速度と比較した場合の小さい方の値です。

速度制限値は次の場合に再計算されます。

- 監視が有効な場合
- オンライン訂正值(磨耗パラメータ)が変更された場合

速度制限値に達すると、制限速度に制限され、以下のNC/PLCインターフェース信号が「1」に設定されます。

---

DB31、 ... DBX83.6 (速度監視) == 1

---

#### 注記

コントローラ内部エラー応答なし。

---

### 起動と解除

研削用工具監視の起動と解除は、パートプログラムで TMON(...)/TMOF(...) (ページ 1047) のプログラミングによって行います。

## 18.4.2 セットアップ

### 自動起動

奇数の工具タイプ番号の研削工具の工具長補正を選択する際に、研削用工具監視を自動的に起動したい場合、以下のマシンデータを「1」に設定してください。

MD20350 \$MC\_TOOL\_GRIND\_AUTO\_TMON (工具監視の起動) = 1

## 18.4.3 プログラミング

### 18.4.3.1 研削用工具監視の有効化/無効化(TMON、TMOF)

予約手順 TMON(...)および TMOF(...)を使用して、研削用工具監視を有効化または無効化します(ジオメトリおよび速度監視)。

### 必要条件

工具固有のパラメータ\$TC\_TPG1 ~\$TC\_TPG9 を設定してください。

### 構文

```
TMON (<TNo>)  
...  
TMOF (<TNo>)
```

## 18.5 砥石周速度一定(GWPS)

## 意味

TMON (...) :	研削用工具監視の有効化 命令は、研削用工具監視が有効になるチャンネルでプログラミングしてください。
TMOF (...) :	研削用工具監視の無効化 命令は、研削用工具監視が無効になるチャンネルでプログラミングしてください。
<TNo>:	T 番号 <b>注記:</b> 監視を現在使用されている有効なといしではなく、無効なといしに対してオンまたはオフにする場合にのみ必要です。
TMOF (0):	すべての工具の監視を解除します

## 18.5 砥石周速度一定(GWPS)

## 18.5.1 機能

といしでは、一般的に主軸速度ではなくといし周速度を使用します。設定する値は、加工種別のプロセスパラメータ (例:といし特性、材料の組合せ)で決定します。

といし周速度をパートプログラムで指定できるようにするためには、工具用パラメータ \$TC\_TPG1 (ページ 1021)、\$TC\_TPG8 (ページ 1024)および\$TC\_TPG9 (ページ 1025)を設定し、機能「といし周速度一定制御 (GWPS)」を有効化してください。

**注記**

GWPS を有効化しても、工具長補正または研削用工具監視は自動的に有効化されません！

この機能が有効になるとすぐに、\$TC\_TPG1 でといしに割り当てられた主軸の以降のすべての S 値はといし周速度として解釈されます。といし速度は、現在のといし半径を考慮してといし周速度が一定に保たれるよう設定されます。

速度計算の公式は次のとおりです。

$$n[\text{min}^{-1}] = \frac{\text{SUG} [\text{m} \cdot \text{s}^{-1}] \cdot 60}{2\pi \cdot R[\text{m}]}$$

計算に使用する訂正值は、工具用パラメータ\$TC\_TPG9で定義されます。関連する磨耗と基本寸法は、工具タイプに応じてこの修正値に追加されます。合計は、パラメータ\$TC\_TPG8(傾斜といしの角度)の値が正であるとき「cos」(\$TC\_TPG8)で割られ、値が負であるとき「sin」(\$TC\_TPG8)で割られます。

速度は次の事象に応じて再計算されます。

- 新規のといし周速度の指定、たとえば次によります。
  - パートプログラムまたはオーバストアのプログラミング
  - MDI のアドレス「S」への割り付け
  - PLC (FC18)による主軸速度制御
- オンライン訂正值(磨耗パラメータ)の変更

## 起動と解除

GWPS の起動と解除は、パートプログラムで GWPSON(...)/GWPSOF(...)をプログラミングして行います(「プログラミング (ページ 1051)」を参照)。

## ステータス

次のインタフェース信号を使用して GWPS が有効であるかどうかを特定できます。

DB31、... DBX84.1 (GWPS 起動)

## 新しい工具

GWPS がすでに有効な主軸に新規の工具を装着した GWPS を選択する場合、まず動作中の GWPS を選択解除してください(選択解除しないとメッセージが出力されます)。

## リセット/パートプログラム終了に対するレスポンス

初期設定では、GWPS はリセットまたはパートプログラムの終了で無効化され、主軸が停止します。必要に応じてこの動作を制御するマシンデータ設定を変更することができます(「セットアップ (ページ 1050)」を参照)。

## 18.5 砥石周速度一定(GWPS)

### その他プロパティ

- GWPS は、すべてのチャンネルのあらゆる研削工具(工具タイプ:400 ~ 499)で有効にすることができます。
- GWPS は、同一チャンネルの複数の主軸で同時に有効にすることができ、それぞれの主軸は別々の研削工具を設置できます。

### 18.5.2 セットアップ

#### 自動起動

電源投入後、主軸にといし周速度一定制御(GWPS)を自動的に選択し、モード切り替え、リセットまたはパートプログラムの終了中も選択状態を維持するには、以下のマシンデータビットを「1」に設定してください。

MD35032 \$MA\_SPIND\_FUNC\_RESET\_MODE ビット 0 = 1

---

#### 注記

必要条件是、工具が研削工具(工具タイプ 400 ~ 499)であり、工具用パラメータ \$TC\_TPG1、\$TC\_TPG8 および\$TC\_TPG9 が設定され、\$TC\_TPG1 が有効な主軸を基準としていることです。

---

#### リセット/パートプログラム終了後の主軸の動作

リセットまたはパートプログラムの終了後に主軸を停止せずに現在の速度で回転を続けるには、以下のマシンデータを「1」に設定してください。

MD35040 \$MA\_SPIND\_ACTIVE\_AFTER\_RESET = 1

---

#### 注記

MD35040 は、主軸モードの開ループ制御操作のみで有効です。

---

### 18.5.3 プログラミング

#### 18.5.3.1 砥石周速度一定制御(GWPSON、GWPSOF)のスイッチオン/オフ:

事前提起された手順 GWPSON(...)および GWPSOF(...)により、研削工具(工具タイプ: 400 ~ 499)の砥石周速度一定(制御) (GWPS)がスイッチオン/オフになります。

#### 構文

```
GWPSON (<TNo>)
S<n>=... :
...
GWPSOF (<TNo>)
```

#### 意味

GWPSON (...):	砥石周速度一定(制御)のスイッチオン
GWPSOF (...):	砥石周速度一定(制御)のスイッチオフ
<TNo>:	T 番号 <b>注記:</b> 砥石周速度一定(制御)を現在使用されている有効なといしではなく、無効なといしに対してオンまたはオフにする場合にのみ必要です。
S<n>=...:	主軸<n>のといし周速度(m/s または ft/s 単位)
S0=...または S...:	メイン主軸のといし周速度

#### 状態確認

このシステム変数を使用すると、特定の主軸で砥石周速度一定(制御)が有効であるかどうかをパートプログラムから確認できます。

\$P\_GWPS[<n>]; ここで、<n> = 主軸番号

値	意味
0 (= FALSE)	GWPS が無効です。
1 (= TRUE)	GWPS が有効です。

18.5 砥石周速度一定(GWPS)

18.5.4 例

といし周速度一定制御が研削工具 T1 と T5 に使用されます。動作中の工具は T1 です。

工具 T1 のデータ(平面研削砥石)

\$TC\_DP1[1,1] = 403 ;工具タイプ  
 \$TC\_DP3[1,1] = 300 ;長さ 1  
 \$TC\_DP4[1,1] = 50 ;長さ 2  
 \$TC\_DP12[1,1] = 0 ;磨耗長さ 1  
 \$TC\_DP13[1,1] = 0 ;磨耗長さ 2  
 \$TC\_DP21[1,1] = 300 ;基本長さ 1  
 \$TC\_DP22[1,1] = 400 ;基本長さ 2  
 \$TC\_TPG1[1] = 1 ;主軸番号  
 \$TC\_TPG8[1] = 0 ;傾斜といしの角度  
 \$TC\_TPG9[1] = 3 ;半径計算のパラメータ番号

工具 T5 のデータ(傾斜研削砥石)

\$TC\_DP1[5,1] = 401 ;工具タイプ  
 \$TC\_DP3[5,1] = 120 ;長さ 1  
 \$TC\_DP4[5,1] = 30 ;長さ 2  
 \$TC\_DP12[5,1] = 0 ;磨耗長さ 1  
 \$TC\_DP13[5,1] = 0 ;磨耗長さ 2  
 \$TC\_DP21[5,1] = 100 ;基本長さ 1  
 \$TC\_DP22[5,1] = 150 ;基本長さ 2  
 \$TC\_TPG1[5] = 2 ;主軸番号  
 \$TC\_TPG8[5] = 45 ;傾斜といしの角度  
 \$TC\_TPG9[5] = 3 ;半径計算のパラメータ番号

プログラミング

プログラムコード	コメント
N20 T1 D1	; T1 および D1 を選択
N25 S1=1000 M1=3	; 主軸 1 は 1000 rpm
N30 S2=1500 M2=3	; 主軸 2 は 1500 rpm
...	
N40 GWPS ON	; 動作中の工具 T1 に対して GWPS を有効化

プログラムコード	コメント
N45 S[\$P_ATPG[1]]=60	; 動作中の工具のGWPSを60 m/sに設定 → n=1909.85 rpm
...	
N50 GWPSON(5)	; 工具5のGWPSを有効化(主軸2)。
N55 S[\$TC_TPG1[5]]=40	; 主軸2のGWPSを40 m/sに設定 n=1909.85 rpm
...	
N60 GWPSON	; 動作中の工具のGWPSを選択解除
N65 GWPSON(5)	; 工具5のGWPSを選択解除(主軸2)
...	

下記も参照

P5:揺動(ページ 847)

## 18.6 データリスト

### 18.6.1 マシンデータ

#### 18.6.1.1 一般マシンデータ

番号	識別子: \$MN_	説明
18075	MM_NUM_TOOLHOLDERS	定義可能な工具ホルダの最大数
18094	MM_NUM_CC_TDA_PARAM	TDA のデータ数
18096	MM_NUM_CC_TOA_PARAM	TOA のデータ数
18100	MM_NUM_CUTTING_EDGES_IN_TOA	TOA ごとの工具オフセット

#### 18.6.1.2 チャンネル別マシンデータ

番号	識別子: \$MC_	説明
20254	ONLINE_CUTCOM_ENABLE	オンライン工具径補正を有効化
20350	TOOL_GRIND_AUTO_TMON	自動工具監視
20610	ADD_MOVE_ACCEL_RESERVE	重畳移動の加速度制限

18.6 データリスト

18.6.1.3 軸/主軸別マシンデータ

番号	識別子: \$MA_	説明
32020	JOG_VELO	JOG 軸速度
35032	SPIND_FUNC_RESET_MODE	個別の主軸機能のリセット応答
35040	SPIND_ACTIVE_AFTER_RESET	独立主軸リセット

18.6.2 信号

18.6.2.1 軸/主軸からの信号

信号名称	SINUMERIK 840D sl	SINUMERIK 828D
形状監視	DB31, ... .DBX83.3	DB390x.DBX2001.3
速度監視	DB31, ... .DBX83.6	DB390x.DBX2001.6
GWPS 有効	DB31, ... .DBX84.1	DB390x.DBX2002.1

## Z2:NC/PLC インタフェース信号

エディション 05/2017 以降、NC/PLC インタフェース信号の詳細については、『NC 変数  
およびインタフェース信号用パラメータマニュアル』を参照してください。



# 付録

# A

## A.1 略語の一覧

A	
O	出力(Output)
ADI4	4 軸アナログドライブインタフェース(Analog Drive Interface for 4 axes)
AC	適応制御(Adaptive Control)
ALM	アクティブラインモジュール(Active Line Module)
ARM	回転型誘導電動機
AS	オートメーションシステム(Automation System)
ASCII	American Standard Code for Information Interchange:情報交換のための米国標準コード
ASIC	Application-Specific Integrated Circuit:ユーザー専用の特定の用途のための集積回路
ASUB	非同期サブプログラム(Asynchronous SUBprogram)
AUXFU	補助機能(AUXiliary FUnction)
STL	ステートメントリスト(Statement List)
UP	ユーザープログラム(User Program)

B	
OP	動作モード
BAG	モードグループ
BCD	2 進化 10 進数(Binary Coded Decimals):バイナリコードで表現された 10 進数
BERO	近接スイッチの商標
BI	バイネクタ入力(Binector Input)
BICO	バイネクタコネクタ(BInector COnnector)
BIN	バイナリファイル(BINary files)
BIOS	基本的な入出力をおこなうシステム(Basic Input Output System)
BCS	基本座標系(Basic Coordinate System)

## A.1 略語の一覧

<b>B</b>	
BO	バイネクタ出力(Binector Output)
OPI	操作パネルインタフェース(Operator Panel Interface)

<b>C</b>	
CAD	コンピュータ支援設計(Computer-Aided Design)
CAM	コンピュータ支援製造(Computer-Aided Manufacturing)
CC	コンパイルサイクル(Compile Cycle)
CEC	交差誤差補正(Cross Error Compensation)
CI	コネクタ入力(Connector Input)
CF カード	コンパクトフラッシュカード(Compact Flash Card)
CNC	コンピュータによる数値制御装置(Computerized Numerical Control)
CO	コネクタ出力(Connector Output)
CoL	ライセンス証明書(Certificate of License)
COM	通信(COMmunication)
CPA	コンパイラプロジェクトデータ(Compiler Projecting Data):コンパイラの設定データ
CRT	ブラウン管(Cathode Ray Tube):受像管
CSB	Central Service Board:PLC モジュール
CU	コントロールユニット(Control Unit)
CP	通信プロセッサ(Communication Processor)
CPU	中央演算処理装置(Central Processing Unit)
CR	キャリッジリターン(Carriage Return)
CTS	Clear To Send:シリアルデータインタフェースへの送信準備完了信号
CUTCOM	工具径補正(CUTter radius COMpensation)

<b>D</b>	
DAC	デジタル/アナログコンバータ(Digital-to-Analog Converter)
DB	データブロック(Data Block)(PLC)
DBB	データブロックバイト(Data Block Byte)(PLC)
DBD	データブロックダブルワード(Data Block Double word)(PLC)
DBW	データブロックワード(Data Block Word)(PLC)

D	
DBX	データブロックビット(Data block bit)(PLC)
DDE	ダイナミックデータ交信(Dynamic Data Exchange)
DDS	ドライブデータセット(Drive Data Set)
DIN	ドイツ工業規格(Deutsche Industrie Norm)
DIO	データ入力/出力(Data Input/Output):データ送受信表示
DIR	ディレクトリ(DIRectory)
DLL	ダイナミックリンクライブラリ(Dynamic Link Library)
DO	ドライブオブジェクト(Drive Object)
DPM	デュアルポートメモリ(Dual Port Memory)
DPR	デュアルポート RAM(Dual Port RAM)
DRAM	ダイナミックメモリ(非保持)
DRF	差動レゾルバ機能(Differential Resolver Function)(手動パルス発生器)
DRIVE-CLiQ	IQ によるドライブコンポーネントリンク(Drive Component Link with IQ)
DRY	ドライラン(DRY run):ドライラン送り速度
DSB	デコードシングルブロック(Decoding Single Block)
DSC	ダイナミックサーボ制御(Dynamic Servo Control) / ダイナミックステイフネスコントロール
DW	データワード(Data Word)
DWORD	ダブルワード(Double Word)(現在は 32 ビット)

E	
I	入力(Input)
EES	外部記憶からの実行(Execution from External Storage)
I/O	入力/出力(Input/Output)
ENC	エンコーダ(ENCorder):フィードバックエンコーダ
EFP	小型 I/O モジュール(PLC I/O モジュール)
ESD	静電気により破損するおそれのある部品(Electrostatic Sensitive Devices)
EMC	電磁両立性(ElectroMagnetic Compatibility)
EN	欧州統一規格
ENC	エンコーダ(ENCorder):フィードバックエンコーダ
EnDat	エンコーダインタフェース(Encoder interface)

## A.1 略語の一覧

E	
EPROM	消去可能なプログラマブル ROM(Erasable Programmable Read Only Memory)
ePS Network Services	インターネットベースのリモート機械保守のサービス
EQN	1回転当たり 2048 のサイン信号を出力する絶対値エンコーダの名称
ES	エンジニアリングシステム(Engineering System)
ESR	停止延長と退避(Extended Stop and Retract)
ETC	ETC キー「>」: 同じメニュー内でのソフトキーバーの拡張

F	
FB	ファンクションブロック (Function Block)(PLC)
FC	ファンクションコール (Function Call): ファンクションブロック (PLC)
FEPROM	フラッシュ EPROM(Flash EPROM): 読み取りと書き込みメモリ
FIFO	First In First Out: アドレス指定なしで動作し、データが格納された順序で読み取られるメモリ
FIPO	ファインインタポレーションの補間器(Fine InterPOLator)
FPU	浮動小数点演算ユニット (Floating Point Unit)
CRC	工具径補正 (Cutter Radius Compensation)
FST	送り停止 (Feed STop)
FBD	ファンクションブロック図 (Function Block Diagram )(PLC プログラミング方式)
FW	ファームウェア (FirmWare)

G	
GC	グローバル制御 (Global Control) (PROFIBUS: ブロードキャストメッセージ)
GDIR	グローバルパートプログラムメモリ
GEO	ジオメトリ (GEOMETRY)、例: ジオメトリ軸
GIA	ギヤ補間データ (Gear Interpolation dAta)
GND	基準電位
BP	基本プログラム (Basic Program )(PLC)
GS	ギヤ選択 (Gear Stage)

G	
GSD	PROFIBUS スレーブを記述するためのデバイスマスタファイル
GSDML	GSD ファイルを作成するための XML ベース記述言語(Generic Station Description Markup Language)
GUD	グローバルユーザーデータ(Global User Data)

H	
HEX	16 進数を示す略語
AuxF	補助機能(Auxiliary Function)
HLA	油圧リニアドライブ(Hydraulic Linear Drive)
HMI	マンマシンインタフェース(Human Machine Interface):SINUMERIK 操作画面
MSD	メイン主軸ドライブ(Main Spindle Drive)
HW	ハードウェア(HardWare)

I	
IBN	セットアップ
ICA	補間型補正(Interpolatory Compensation)
IM	インタフェースモジュール: 内部接続モジュール(Interface Module)
IMR	データ受信用のインタフェースモジュール(Interface Module Receive)
IMS	データ送信用のインタフェースモジュール(Interface Module Send)
INC	インCREMENT(INCrement)
INI	初期化データ(INItializing Data)
IPO	補間器(InterPOLator)
ISA	コンピュータのバス規格(Industry Standard Architecture)
ISO	国際標準化機構(International Standardization Organization)

J	
JOG	ジョグ(JOGging):セットアップモード

## A.1 略語の一覧

K	
$K_v$	制御ループのゲイン係数
$K_p$	比例ゲイン
$K_U$	伝達比
LAD	ラダー図(Ladder Diagram)(PLC プログラミング方式)

L	
LAI	論理機械軸イメージ(Logic Machine Axis Image)
LAN	ローカルエリアネットワーク(Local Area Network)
LCD	液晶ディスプレイ(Liquid Crystal Display)
LED	発光ダイオード(Light Emitting Diode)
LF	ラインフィード(Line Feed)
PMS	位置検出器(Position Measuring System)
LR	位置コントローラ
LSB	最下位ビット(Least Significant Bit)
LUD	ローカルユーザーデータ(Local User Data)

M	
MAC	メディアアクセス制御(Media Access Control)
MAIN	メインプログラム(MAIN program)(OB1、PLC)
MB	メガバイト(MegaByte)
MCI	モーションコントロールインタフェース(Motion Control Interface)
MCIS	モーションコントロール情報システム(Motion Control Information System)
MCP	機械操作パネル(Machine Control Panel)
MD	マシンデータ(Machine Data)
MDI	手動データ入力(Manual Data Input)
MDS	モータデータセット(Motor Data Set)
MSGW	メッセージワード(MeSsaGe Word)
MCS	機械座標系(Machine Coordinate System)
MM	モータモジュール(Motor Module)

M	
MPF	メインプログラムファイル(Main Program File):メインプログラム(NC)
MCP	機械操作パネル(Machine Control Panel)

N	
NC	数値制御移動範囲などのブロック解析をおこなう数値制御
NCU	数値制御装置(Numerical Control Unit):NC ハードウェアユニット
NRK	NC のオペレーティングシステムの名称
IS	インタフェース信号(Interface Signal)
NURBS	非一様有理 B スプライン(Non-Uniform Rational B-Spline)
WO	ゼロオフセット(Work Offset)
NX	数値拡張(Numerical eXtension):軸の拡張基板

O	
OB	PLC のオーガニゼーションブロック (Organization Block in the PLC)
OEM	独自製品製造業者(Original Equipment Manufacturer)
OP	操作パネル(Operator Panel)
OPI	Operator Panel Interface:操作パネルへの接続用インタフェース
OPT	オプション(OPTION)
OLP	光リンクプラグ(Optical Link Plug):光ファイババスコネクタ
OSI	開放型システム間相互接続(Open Systems Interconnection):コンピュータ通信の標準規格

P	
PIQ	プロセスイメージ出力
PII	プロセスイメージ入力(Process Image Input)
PC	パーソナルコンピュータ(Personal Computer)
PCIN	制御装置とのデータ交信用ソフトウェアの名称
PCMCIA	パーソナルコンピュータメモ리카ード国際協会(Personal Computer Memory Card International Association): プラグインメモ리카ードの標準規格
PCU	PC ユニット(PC Unit):コンピュータ装置

## A.1 略語の一覧

P	
PG	プログラミング装置(Programming device)
PKE	パラメータ識別(Parameter identification):PIV の一部
PIV	パラメータ識別子(Parameter Identification Value):PPO のパラメータ設定部分
PLC	プログラマブル論理制御(Programmable Logic Control):カスタマイズ制御
PN	PROFINET
PNO	PROFIBUS 協会
PO	電源投入(POWER ON)
POU	プログラム構成単位(Program Organization Unit)
POS	位置/位置決め(Position/positioning)
POSMO A	POSitioning MOtor Actuator:位置決めモータ
POSMO CA	POSitioning MOtor Compact AC:位置決めユニットとプログラムメモリだけでなく、内蔵の電源および制御モジュール、つまり AC 電源装置を備えた一体型ドライブユニット
POSMO CD	POSitioning MOtor Compact DC:POSMO CA と似ているが DC 電源装置を搭載
POSMO SI	POSitioning MOtor Servo Integrated:位置決めモータ、DC 電源装置
PPO	パラメータプロセスデータオブジェクト(Parameter Process data Object):PROFIBUS DP 伝送用のサイクリックデータメッセージと「可変速ドライブ」プロファイル
PPU	パネルプロセスユニット(Panel Processing Unit): パネルベースの CNC のセントラルハードウェア、例: SINUMERIK 828D
PROFIBUS	PROcess Field BUS:シリアルデータバス
PRT	プログラムテスト(PROgram Test)
PSW	プログラムコントロールワード
PTP	ポイントツーポイント(Point-To-Point)
PUD	プログラムグローバルユーザーデータ(Program global User Data):プログラムグローバルのユーザー変数
PZD	プロセスデータ(Process data):PPO のプロセスデータ部分

Q	
QEC	象限突起補償(Quadrant Error Compensation)

R	
RAM	ランダムアクセスメモリ (Random Access Memory):読み取り/書き込みメモリ
REF	レファレンス点復帰機能(REFerence point approach function)
REPOS	再位置決め機能(REPOSition function)
RISC	縮小命令セットコンピュータ(Reduced Instruction Set Computer):小規模命令セットと高速で命令を処理する能力を備えたタイプのプロセッサ
ROV	早送りオーバライド(Rapid OVerride):入力補正
RP	R 変数(R Parameter)、算術変数、予約ユーザー変数
RPA	R 変数用の NCK 上のメモリ領域 R 変数番号用の NC 上のメモリ領域
RPY	ロールピッチヨー(Roll Pitch Yaw):座標系の回転タイプ
RTL	早送り動作中の直線補間(Rapid Traverse Linear Interpolation)
RTS	Request To Send:シリアルデータインタフェースの制御信号
RTCP	リアルタイム制御プロトコル(Real Time Control Protocol)

S	
SA	シンクロナイズドアクション(Synchronized Action)
SBC	安全ブレーキ制御(Safe Brake Control)
SBL	シングルブロック(Single BLock)
SBR	サブプログラム(Subroutine)(PLC)
SD	セッティングデータ(Setting Data)
SDB	システムデータブロック(System Data Block)
SEA	Setting Data Active:セッティングデータの識別子(ファイルタイプ)
SERUPRO	プログラムテストによる検索実行ブロック検索、プログラムテスト
SFB	システムファンクションブロック(System Function Block)
SFC	システムファンクションコール(System Function Call)
SGE	安全関連入力
SGA	安全関連出力
SH	安全停止
SIM	シングルインラインモジュール(Single Inline Module)
SK	ソフトキー(SoftKey)

## A.1 略語の一覧

S	
SKP	スキップ(Skip):パートプログラムブロックをスキップするための機能
SLM	同期リニアモータ(Synchronous Linear Motor)
SM	ステッピングモータ(Stepper Motor)
SMC	取り付けセンサモジュールキャビネット(Sensor Module Cabinet Mounted)
SME	外部取り付けセンサモジュール外部(Sensor Module Externally Mounted)
SMI	内蔵センサモジュール(Sensor Module Integrated)
SPF	Sub Routine File:サブプログラム(NC)
PLC	プログラマブルロジックコントローラ(Programmable Logic Controller)
SRAM	スタテック RAM(Static RAM) (不揮発性)
TNRC	ノーズ R 補正(Tool Nose Radius Compensation)
SRM	回転型同期モータ(Synchronous Rotary Motor)
LEC	ピッチ誤差補正(Leadscrew Error Compensation)
SSI	同期シリアルインタフェース(Serial Synchronous Interface)
SSL	ブロック検索
STW	コントロールワード
GWPS	砥石周速制御(Grinding Wheel Peripheral Speed)
SW	ソフトウェア(SoftWare)
SYF	システムファイル(System Files)
SYNACT	シンクロナイズドアクション(SYNchronized ACTion)

T	
TB	端子基板(Terminal Board) (SINAMICS)
TCP	工具中心点(Tool Center Point):工具先端
TCP/IP	伝送制御プロトコル(Transport Control Protocol) /インターネットプロトコル(Internet Protocol)
TCU	シンクライアントユニット(Thin Client Unit)
TEA	Testing Data Active:マシンデータの識別子
TIA	統合オートメーション(Totally Integrated Automation)
TM	ターミナルモジュール(Terminal Module) (SINAMICS)
TO	工具オフセット(Tool Offset)
TOA	Tool Offset Active:工具オフセットの識別子(ファイルタイプ)

T	
TRANSMIT	TRANSform Milling Into Turning:施盤でのフライス削り運転用の座標変換
TTL	トランジスタ-トランジスタロジック(Transistor-Transistor Logic)(インタフェースタイプ)
TZ	テクノロジーサイクル

U	
UFR	ユーザフレーム(User FRame):ゼロオフセット
SR	サブプログラム
USB	ユニバーサルシリアルバス(Universal Serial Bus)
UPS	無停電電源装置(Uninterruptible Power Supply)

V	
VDI	NC と PLC の間の内部通信インタフェース
VDI	ドイツエンジニア協会(Verein Deutscher Ingenieure)
VDE	ドイツ電気技術者協会(Verband Deutscher Elektrotechniker)
VI	電圧入力(Voltage Input)
VO	電圧出力(Voltage Output)
FDD	送り用ドライブ(Feed Drive)

W	
SAR	滑らかなアプローチと後退(Smooth Approach and Retraction)
WCS	ワーク座標系(Workpiece Coordinate System)
T	工具(Tool)
TLC	工具長補正(Tool Lenght Compensation)
WOP	現場指向プログラミング(Workshop-Oriented Programming)
WPD	ワークディレクトリ(WorkPiece Directory)
TRC	工具径補正(Tool Radius Compensation)
T	工具(Tool)
TO	工具オフセット(Tool Offset)

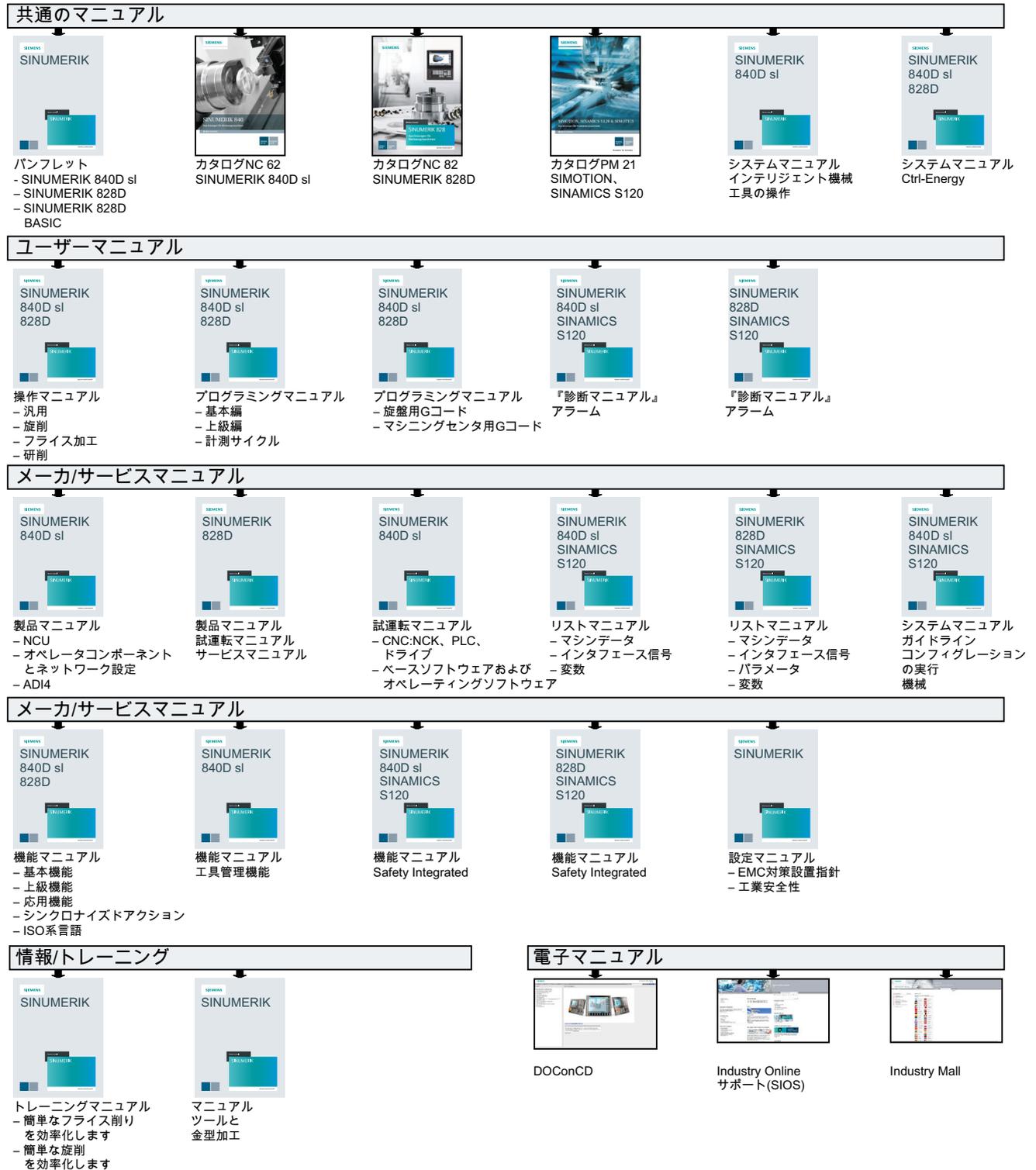
A.1 略語の一覧

W	
TM	工具管理(Tool Management)
TC	工具交換(Tool change)

X	
XML	拡張マークアップ言語(eXtensible Markup Language)

Z	
WOA	ゼロオフセットの識別子(Work Offset Active)
ZSW	ステータスワード(ドライブの)

## A.2 概要





# 用語

## CNC

→ NC を参照してください。

コンピュータによる数値制御(Computerized Numerical Control):→ NC、→ PLC、HMI、  
→ COM のコンポーネントが含まれます。

## CNC

→ NC を参照してください。

コンピュータによる数値制御(Computerized Numerical Control):→ NC、→ PLC、HMI、  
→ COM のコンポーネントが含まれます。

## COM

通信の実行と調整用の NC コンポーネントです。

## CPU

中央処理装置。 → PLC を参照してください。

## CU

伝達比

## C スプライン

C スプラインは最もよく知られていて広く使用されているスプラインです。補間点での遷移は、接線と曲率が連続的に変化するようにおこなわれます。3 次の多項式が使用されます。

## C 軸

工具主軸の制御された回転移動および位置決め移動の中心となる軸。

## DRF

差動レゾルバ機能(Differential Resolver Function):自動モードで手動パルス発生器と組み合わせてインクレメンタルゼロオフセットを生成する NC 機能です。

## HIGHSTEP

AS300/AS400 システムの → PLC のプログラミングオプションの要約です。

## HW Config

S7 プロジェクト内のハードウェアコンポーネントの構成とパラメータ設定用 SIMATIC S7 ツール

## JOG

コントローラの運転モード(セットアップモード):機械は JOG モードで段取りができません。個々の軸と主軸を、方向キーによって JOG モードで移動できます。JOG モードには次の追加機能があります: → レファレンス点復帰、→ 再位置決め、および → プリセット(現在値の設定)。

## KV

制御ループ内のサーボゲイン係数、制御変数です。

## MDI

コントローラの運転モードです。手動データ入力。MDI モードでは、メインプログラムやサブプログラムと関係しない個別のプログラムブロックまたはブロック列を入力でき、その後すぐに NC スタートキーの作動により実行できます。

## NC

→ パートプログラムを実行し、工作機械の動作を制御する → CNC の数値制御(Numerical Control)コンポーネントです。

## NRK

数値ロボットカーネル( → NC のオペレーティングシステム)

## NURBS

コントローラ内のモーションコントロールと軌跡補間は、**NURBS (Non Uniform Rational B-Splines: 非一様有理 B スプライン)**に基づいておこなわれます。これは、すべての内部補間に同じ処理を提供します。

## OEM

個々のソリューション(**OEM アプリケーション**)を実現するための適用範囲は、独自のユーザーインタフェースの作成やテクノロジー固有機能のコントローラへの統合をおこないたい工作機械メーカーに合わせて提供されています。

## PCIN データ送受信プログラム

PCIN は、CNC ユーザーデータ(たとえば、パートプログラム、工具オフセットなど)をシリアルインタフェースを介して送受信するためのユーティリティプログラムです。PCIN プログラムは、標準の工業用 PC 上の MS-DOS で実行できます。

## PLC

**Programmable Logic Controller (プログラマブルコントローラ):** →プログラマブルロジックコントローラ。 **NC のコンポーネント:** 工作機械の論理制御を処理するためのプログラマブルコントローラ。

## PLC プログラミング

**STEP 7** ソフトウェアを使用して、PLC をプログラム指令します。STEP 7 プログラミングソフトウェアは、**WINDOWS** 標準オペレーティングシステムの下で事項され、改良された拡張機能とともに **STEP 5** プログラミング機能を含んでいます。

## PLC プログラムメモリ

**SINUMERIK 840D sl:** PLC ユーザープログラム、ユーザーデータおよび PLC 基本プログラムは、まとめて PLC ユーザーメモリに格納されています。

## RS-232-C

データ入出力用のシリアルインタフェース。加工プログラムとメーカーデータ、ユーザーデータは、このインタフェースを介してロード、保存できます。

## R 変数

プログラムにおける任意の目的のためにパートプログラムプログラマが設定または確認することができる算術変数。

## TOA ユニット

各 TOA 領域は複数の TOA ユニットを持つことができます。可能な TOA ユニットの数は、有効なチャンネルの最大数によって制限されます。TOA ユニットには、正確に 1 つのデータブロックと 1 つのマガジンデータブロックが含まれます。また、TOA ユニットには工具ホルダデータブロック(オプション)を含めることもできます。

## TOA 領域

TOA 領域には、すべての工具とマガジンのデータが含まれます。初期設定で、データのアクセスに関してこの領域は→チャンネル領域と一致します。ただし、マシンデータを使用して、複数のチャンネルが 1 つの TOA ユニットの共有するように指定して、共通工具管理データをこれらのチャンネルで使用可能にすることができます。

## WinSCP

WinSCP は、ファイルを転送するために無料で使用できる Windows 用オープンソースプログラムです。

## アーカイビング(Archiving)

外部メモリ機器にファイルとディレクトリから読み取ります。

## アドレス

アドレスは、入力や出力などの特定のオペランドまたはオペランド範囲の識別子です。

## アブソリュート指令

軸移動の移動先は、現在有効な座標系の原点を基準とする寸法によって定義されます。  
→インクレメンタル指令を参照してください。

## アラーム

すべての→メッセージとアラームは、日付と時刻および解除条件に対応するシンボルとともに、操作パネルにプレーンテキストで表示されます。アラームとメッセージは別々に表示されます。

1. パートプログラムのアラームとメッセージ:  
アラームとメッセージは、パートプログラムから直接プレーンテキストで表示できます。
2. PLC からのアラームとメッセージ  
機械のアラームとメッセージは、PLC プログラムからプレーンテキストで表示できます。このために追加のファンクションブロックパッケージは必要ありません。

## イグザクトストップ

イグザクトストップ命令がプログラム指令されている場合、ブロックに指定された位置は正確に、また必要であれば低速で位置決めされます。移動時間を短縮するため、早送りと送りに→イグザクトストップ範囲が定義されています。

## イグザクトストップ範囲

すべての軌跡軸がイグザクトストップ範囲に達すると、制御装置は正確な終点に到達したものと動作します。→パートプログラムは次のブロックに進行します。

## インクレメンタル指令

インクレメンタル指令:軸移動の目標は、移動する距離とすでに到達した点を基準とする方向によって定義されます。→アブソリュート指令を参照してください。

## インCREMENT

INCREMENT数に基づく移動軌跡長さの指定。INCREMENT数は、セッティングデータとして格納するか、または適切にラベル付けされたキー(つまり、10、100、1000、10000)によって選択できます。

## インチ単位系

インチおよびインチの小数で距離を定義する単位系。

## インバースタイム送り

また、軸動作に対して、送り速度ではなく、ブロックの軌跡が移動するのに必要な時間をプログラム指令できます(G93)。

## エディタ

エディタを使用すると、プログラム/テキスト/プログラムブロックを作成、編集、拡張、結合およびインポートできます。

## オーバーライド

特定のワークや材料に対して、プログラム指令送り速度や回転数を最適化するために、ユーザーによるオーバーライドを可能にする手動またはプログラム指令可能な介入機能。

## オフセットメモリ

工具オフセットデータが格納されるコントローラ内のデータ領域です。

## キースイッチ

→機械操作パネルのスイッチには、コントローラのオペレーティングシステムによって機能が割り当てられた4つの位置があります。キースイッチには、3つの異なる色のキーがあり、指定した位置で取り外すことができます。

## キーワード

パートプログラムに対してプログラミング言語で定義された意味を持つ特定表記によるワード。

## サイクル

→ワークに対して繰り返しおこなわれる加工運転を実行するための保護されたサブプログラム。

## サブプログラム

「サブプログラム」という用語は、パートプログラムがメインプログラムとサブプログラムに厳密に分類されたときに生まれました。今日の **SINUMERIK NC** 言語では、このように厳密には分類されていません。原則として、すべてのパートプログラムまたはすべてのサイクルは、別のパートプログラム内でサブプログラムとして呼び出すことができます。呼び出された後、次のプログラムレベル(x+1) (サブプログラムレベル (x+1))で実行されます。

## サブブロック

たとえば番号などの順序に関する情報を含む「N」が頭に付くブロックです。

## ジオメトリ

→ ワーク座標系での → ワークの記述です。

## ジオメトリ軸

ジオメトリ軸は、→ パートプログラムで、ワークのジオメトリがプログラム指令される 2次元または 3次元の → ワーク座標系を形成します。

## システムメモリ

システムメモリは、次のデータが格納されている CPU 内のメモリです。

- オペレーティングシステムに必要なデータ
- 演算タイマ、カウンタ、マーカ

## システム変数

パートプログラムのプログラマによる入力は何もなくとも存在する変数。これは、\$の文字が頭に付いた、データタイプと変数名称によって定義されます。参照→ユーザー定義変数。

## シンクロナイズドアクション

1. 補助機能出力  
ワーク加工中、技術的な機能(補助機能)を CNC プログラムから PLC に出力できます。たとえば、次の補助機能を使用して、クイル、グラブ、クランピングチャックなどの工作機械の追加機器が制御されます。
2. 高速補助機能出力  
高速動作が必要なスイッチ機能のために、補助機能の応答時間を短縮して、加工処理での不要な待機時間を回避することができます。

## スケーリング

軸別の尺度を変更するフレームの成分。

## スプライン補間

スプライン補間を使用すると、コントローラは、設定輪郭のごくわずかな指定補間点から滑らかな曲線特性を生成できます。

## セッティングデータ

工作機械の特性を NC に伝える、システムソフトウェアを通じて定義されたデータ。

## ゼロオフセット

既存の原点と → フレームを基準として座標系の新しい基準点を指定します。

1. 設定可能  
任意の数の設定可能ゼロオフセットを CNC 軸毎に使用できます。オフセット - G 命令で選択された - が交互に有効になります。
2. 外部  
ワークの位置を定義するすべてのオフセットだけでなく、外部ゼロオフセットをハンドル (DRF オフセット) によって、または PLC から重畳することができます。
3. プログラム指令可能  
ゼロオフセットは、TRANS 命令を使用して、すべての軌跡軸と位置決め軸に対してプログラム指令できます。

## ソフトウェアリミットスイッチ

ソフトウェアリミットスイッチは、軸の移動範囲を制限して、ハードウェアリミットスイッチでのスライドの急停止を防止します。2つの値のペアを軸毎に指定して、PLC によって別々に起動できます。

## ソフトキー

その名称が画面の操作エリアに表示されるキー。表示されるソフトキーの選択は、運転状況に随時、適用されます。自由に割り当てられるファンクションキー(ソフトキー)には、ソフトウェアで定義された機能が割り当てられます。

## チャンネル

チャンネルは、→ パートプログラムを他のチャンネルと関係なく処理できるという特徴もっています。チャンネルは、そこに割り当てられた軸と主軸のみを制御します。別のチャンネルで実行されるパートプログラムとは、→ 同期によって協調できます。

## データブロック

1. → HIGHSTEP プログラムがアクセスできる → PLC のデータユニットです。
2. → NC のデータユニット:データブロックには、グローバルユーザーデータのデータ定義が含まれます。このデータは定義後そのまま初期化することができます。

## データワード

→ データブロック内の 2 バイトデータユニットです。

## テキストエディタ

参照→エディタ

## トータルリセット

全体リセットのとき、CPU の次のメモリが削除されます。

- →作業メモリ
- ロードメモリの読み取り/書き込み領域
- →システムメモリ
- →バックアップメモリ

## ドライブ

ドライブは、NC の設定に基づいて速度とトルクの制御を実行する CNC のユニットです。

## ネットワーク

ネットワークとは、→接続ケーブルを介した、複数の S7-300 および、その他の端末機器(プログラミング機器など)のつながりのことをいいます。データ通信は、接続された機器の間でネットワークを通じておこなわれます。

## ノーズ R 補正

輪郭プログラミングでは、工具が位置決めされるものと想定します。実際にはそうならないため、使用される工具の曲率半径をコントローラに伝えて考慮に入れる必要があります。曲率中心は、曲率半径によってオフセットされて、輪郭に等距離で維持されません。

## パートプログラム

特定のワークを作成するために一体となって動作する **NC** に対する一連の命令。同様に、この用語は特定の素材に対する特定の加工運転の実行にも適用されます。

## パートプログラムブロック

ラインフィードによって仕切られるパートプログラムの部分。次の 2 つのタイプがあります。→メインブロックと→サブブロック

## パートプログラム管理

パートプログラム管理はワークを基準に構成できます。ユーザーメモリのサイズは、プログラムの数と、管理可能なデータ量を決定します。各ファイル(プログラムとデータ)には、最大 24 文字の英数字からなる名称を指定できます。

## バックアップバッテリー

バックアップバッテリーによって、CPU のユーザープログラム格納されて停電から確実に防護され、指定されたデータ領域とビットメモリ、タイマおよびカウンタが確実に保持されます。

## バックラッシ補正

機構上の機械バックラッシ、たとえばボールネジの反転時のバックラッシなどを補正します。バックラッシ補正は軸毎に別々に入力できます。

## ピッチ誤差補正

送りに関与する送りねじの機械的な誤差に対する補正。コントローラは、保存された誤差値を補正に使用します。

## ブート

電源投入後のシステムプログラムのロードです。

## フレーム

フレームは、1 つの直交座標系を別の座標系に変換する演算規則です。フレームには次の成分が含まれます: →ゼロオフセット、→座標回転、→スケーリング、→ミラーリング。

### フローティングチェックなしのタッピング

この機能を使用すると、フローティングチェックなしでねじをタッピングできます。回転軸と穴あけ軸としての主軸の補間方法を使用することによって、たとえば止まり穴ねじのように、ねじを最終穴あけ深さまで正確に切削できます(必要条件:主軸の軸運転)。

### プログラマブルコントローラ(Programmable logic controller)

プログラマブルコントローラ(PLC)は電子コントローラであり、その機能はコントロールユニットにプログラムとして格納されています。つまり、機器のレイアウトと配線は、コントローラの機能に依存しません。プログラマブルコントローラは、コンピュータと同じ構成です。これは、メモリを備えたCPU(中央モジュール)、入出力モジュールおよび内部バスシステムからなります。周辺機器とプログラミング言語は、コントローラの機能の要求事項と一致します。

### プログラマブルフレーム

プログラマブルフレームを使用すると、パートプログラムの実行中、新しい座標系出力点の動的定義が可能になります。新しいフレームを使用する絶対定義と、既存の起点を基準とする追加定義が区別されます。

### プログラマブル作業領域リミット

工具の移動空間を、プログラム指令範囲によって定義された空間に制限します。

### プログラミングキー

→パートプログラム用プログラミング言語で定義された意味を持つ文字と文字列。

### プログラムブロック

プログラムブロックには、→パートプログラムのメインプログラムとサブプログラムが含まれています。

### プログラムレベル

チャンネル内で開始されたパートプログラムは、プログラムレベル0(メインプログラムレベル)でメインプログラムとして実行されます。メインプログラムで呼び出されたすべてのパートプログラムは、それ自体のプログラムレベル1...nでサブプログラムとして実行されます。

## ブロック

「ブロック」は、プログラムの作成と処理に必要なすべてのファイルを示す用語です。

## ブロック検索

デバッグ目的で、またはプログラム中止に続いて、「ブロック検索」機能を使用して、プログラムを開始または再開するパートプログラム内の任意の位置を選択できます。

## プロテクションゾーン

工具先端を通過させてはならない作業領域内の 3 次元領域。

## ヘリカル補間

ヘリカル補間機能は、成形フライスカッタを使用する、おねじ、および、めねじの加工と、潤滑溝のフライス加工に適しています。

ヘリカルは 2 つの移動で構成されます。

- 1 つの平面での円移動
- この平面に垂直な直線移動

## ボーレート

データ送受信速度(bps)。

## マクロ機能

1 つの識別子による一連の命令のグループです。この識別子は、プログラムのなかで一連の統合された命令を表します。

## マスタ軸

マスタ軸は → ガントリ軸で、オペレータとプログラムの視点からみると存在し、そのため標準的な NC 軸のように影響を受けます。

## ミラーリング

ミラーリングは、軸に関して輪郭の座標値の符号を反転します。一度に複数の軸に関してミラーリングをおこなうことができます。

## メインプログラム

「メインプログラム」という用語は、パートプログラムがメインプログラムと→サブプログラムに厳密に分類されたときに生まれました。今日の SINUMERIK NC 言語では、このように厳密には分類されていません。原則として、チャンネル内のすべてのパートプログラムを選択して開始できます。パートプログラムは→プログラムレベル 0 (メインプログラムレベル) で実行されます。また、パートプログラムまたはサブプログラムである→サイクルをメインプログラムで呼び出すことができます。

## メインブロック

→パートプログラム内の動作シーケンスを開始するためのすべての情報が含まれた、先頭に「:」の付いているブロック。

## メッセージ

パートプログラムでプログラム指令されたすべてのメッセージおよびシステムによって検出されたすべての→アラームは、日付と時刻および解除条件に対応するシンボルとともに、操作パネルにプレーンテキストで表示されます。アラームとメッセージは別々に表示されます。

## メトリック単位系

基本単位系:長さの場合は、たとえば mm(ミリメートル)、m(メートル)です。

## モード

SINUMERIK コントロールシステムの運転仕様です。次のモードが定義されています。  
→ジョグ → MDI、→ 自動。

## モードグループ

技術的に関連する軸と主軸は、1つのモードグループに組み合わせることができます。モードグループの軸/主軸は、1つまたは複数の→チャンネルによって制御できます。同じ→モードタイプが、常に、そのモードグループのチャンネルに割り当てられます。

## ユーザーインターフェース

ユーザーインターフェース(UI)は、画面による CNC 用の表示媒体です。これは、水平と垂直ソフトキーが特徴です。

## ユーザープログラム

**S7-300** オートメーションシステムのユーザープログラムは、プログラミング言語 **STEP 7** を使用して作成されます。ユーザープログラムはモジュール構造で、個々のブロックから構成されます。

基本ブロックタイプは次のとおりです。

- コードブロック  
これらのブロックには、**STEP 7** の命令が含まれます。
- データブロック  
これらのブロックには、**STEP 7** プログラムの定数と変数が含まれます。

## ユーザーメモリ

パートプログラム、サブプログラム、コメント、工具オフセット、ゼロオフセット/フレーム、さらにチャンネル、およびプログラムユーザーデータなどのすべてのプログラムとデータを共有 **CNC** ユーザーメモリに格納できます。

## ユーザー定義変数

ユーザーは、→パートプログラムまたはデータブロック(グローバルユーザーデータ)で任意の目的で独自の変数を宣言できます。定義には、データタイプ指定と変数名称が含まれます。→システム変数を参照してください。

## レファレンス点

機械軸の検出器が基準とする工作機械の位置。

## ロードメモリ

ロードメモリは、→PLC の→CPU 314 の→作業メモリと同じです。

## ワーキングエリアリミット

ワーキングエリアリミットによって、軸の移動範囲をリミットスイッチに加えてさらに制限することができます。軸あたりに 1 つの値ペアを使用して、保護する作業領域を記述することができます。

## ワーク

工作機械によって作成/加工される部品。

## ワーク原点

ワーク原点は、ワーク座標系の起点です。これは、機械原点との距離に関連して定義されます。

## ワーク座標系

ワーク座標系は、ワーク原点に、その起点があります。ワーク座標系でプログラム指令された加工運転で、寸法と方向はこの座標系を基準とします。

## ワーク輪郭

作成または加工されるワークの指令輪郭。

## 安全機能(Safety functions)

コントローラは、→CNC、→PLC および機械の故障を適切に検出する常に有効な監視機能を備えているため、ワーク、工具または機械の損傷の大部分は回避されます。故障の際は、加工運転は中断されてドライブは停止します。誤動作の原因が記録されて、アラームとして出力されます。同時に、PLC には CNC アラームがトリガされたことが通知されます。

## 位置タイマスイッチ

「位置タイマスイッチ」という用語は、定義された軸位置で特定の持続時間のパルスを出力できるポジションスイッチのペアを意味します。

## 位置決め軸

工作機械での補助移動(工具マガジン、パレット搬送など)をおこなう軸。位置決め軸とは、軌跡軸と補間しない軸のことです。

## 移動範囲

直線軸の最大許容移動範囲は、±9 桁です。アブソリュート値は、選択された最小入力単位と位置制御分解能および基本単位(インチまたはメートル)によって決まります。

## 円弧補間

→ 工具は輪郭上の指定された点の間の円の上を指定された送り速度で移動し、それによってワークが加工されます。

## 加々速度一定加減速

機構部を保護しながら同時に機械の加減速応答を最適化するために、パートプログラムで不連続な加減速と連続的な加減速(加々速度一定)を切り替えることができます。

## 加工チャンネル

チャンネル構造は、たとえばローディングクレーンを加工と同時に移動するなどの並列動作処理によってアイドル時間を短縮するために使用できます。ここで、**CNC** チャンネルは、解説、ブロック解析および補間において、別の **CNC** コントロールシステムとみなす必要があります。

## 回転軸

回転軸は、定義された角度へのワークまたは工具の回転を実現します。

## 外部ゼロオフセット

→ PLC によって指定されたゼロオフセットです。

## 割り込みルーチン

割り込みルーチンは加工処理中にイベント(外部信号)によって起動できる特殊なサブプログラムです。現在動作中のパートプログラムブロックが中断されて、中断点での軸の位置が自動的に保存されます。

## 割り出し軸

割り出し軸は、インデックスグリッドに対応する角度までワークまたは工具を回転させます。グリッドに到達すると、割り出し軸は「インポジション」になります。

## 完成品の輪郭

完成ワークの輪郭です。→ 素材を参照してください。

## 基本座標系

座標変換によって機械座標系へ投影される直交座標系です。

プログラマは → パートプログラムで基本座標系の軸名称を使用します。基本座標系は、  
→ 座標変換が有効ではない場合 → 機械座標系に平行して存在します。 → 軸名称に違いがあります。

## 基本軸

その指令値または現在位置が、補正值の計算の基礎を形成する軸です。

## 機械原点

(得られた)すべての検出位置へ復帰できる工作機械の固定点です。

## 機械固定点

工作機械によって一義的に定義される点、たとえば機械レファレンス点です。

## 機械座標系

工作機械の軸に関連する座標系です。

## 機械軸

工作機械に物理的に存在する軸です。

## 機械操作パネル

キー、ロータリスイッチなどの操作部品と LED などの単純な表示器を備えた工作機械上の操作パネルです。これを使用して、PLC を介して工作機械を直接操作します。

## 軌跡誤差監視

追従誤差は、輪郭精度の尺度として定義可能な許容誤差範囲内で監視されます。許容できないほど大きな追従誤差があると、たとえばドライブが過負荷になる場合があります。このような場合はアラームが出力されて、軸が停止します。

## 軌跡軸

軌跡軸には、→補間器によって起動、加速、停止および終点への到達が同時におこなわれるような方法で制御される→チャンネルのすべての機械軸が含まれます。

## 軌跡送り速度

軌跡送り速度は→軌跡軸に作用します。関連する→ジオメトリ軸の送り速度のジオメトリック合計を表しています。

## 軌跡速度

プログラム指令可能な最大軌跡速度は、最小入力単位によって決まります。たとえば、最小単位 0.1mm の場合、プログラム指令可能な最大軌跡速度は 1000m/min になります。

## 曲率

輪郭の曲率  $k$  は輪郭点に沿った円弧半径  $r$  の逆数( $k = 1/r$ )です。

## 極座標

平面上の点の位置を、基点からの距離と、定義済み軸での半径ベクトルによって形成された角度によって定義する座標系。

## 傾斜面加工

「傾斜面加工」機能を使用して、機械の座標平面にないワーク面に対する穴あけ加工とフライス加工運転を簡単におこなうことができます。

## 固定点アプローチ

工作機械は、工具交換位置、ロードポイント、パレット交換位置などの固定点に、定義された方法で移動できます。これらの点の座標はコントローラ内に格納されています。コントローラは、可能な場合はいつも、関連する軸を→早送りで移動します。

## 工具

加工を実行する工作機械の作用部分(旋削工具、フライス工具、ドリル、レーザービームなど)。

## 工具径補正

必要なワーク輪郭を直接プログラム指令するには、コントローラは、使用される工具の半径を考慮に入れて、プログラム指令輪郭に一定の距離を置く軌跡を移動させる必要があります(G41/G42)。

## 工具補正

軌跡計算における工具寸法の考慮。

## 高級 CNC 言語

高級言語は、**NC** プログラム、→シンクロナイズドアクション、および→サイクルを書き込むために使用されます。これは次のものを提供します:制御構造 →ユーザー定義変数、→システム変数、→マクロプログラミング。

## 高速デジタル入/出力

デジタル入力部を使用して、たとえば高速 **CNC** プログラムルーチン(割り込みルーチン)を起動できます。高速のプログラム駆動スイッチ機能を、デジタル **CNC** 出力で起動できます。

## 座標回転

ある角度での座標系の回転を定義するフレームの成分。

## 座標系

→機械座標系と →ワーク座標系を参照してください。

## 座標変換

軸の追加またはアブソリュートゼロオフセット。

## 作業メモリ

作業メモリは、アプリケーションプログラムを処理するときにプロセッサがアクセスする → CPU の RAM です。

## 作業領域

工作機械の物理的な構成を考慮して、工具先端が移動できる 3 次元の領域です。→ プロテクションゾーンを参照してください。

## 事前一致

軌跡残距離が終了位置の指定可能な範囲に等しい量に近づくと、ブロック切り替えがおこなわれます。

## 自動

コントローラの運転モード(DIN に準拠したブロックシーケンス運転):→ パートプログラムが選択されて連続的に実行される NC システムの運転モードです。

## 識別子

DIN 66025 に従って、ワードは、変数(算術変数、システム変数、ユーザー変数)、サブプログラム、キーワードおよび複数のアドレス文字を持つワードの識別子(名称)を使用して補完されます。これらの補完は、ブロック形式に関してワードと同じ意味を持ちます。識別子は一義的にしてください。異なる複数の対象には同じ識別子を使用できません。

## 軸

対象機能に従って、CNC 軸は次のように分割されます。

- 軸:軌跡軸の補間
- 補機軸:軸別の送り速度による非補間の軸送りと軸の位置決め。たとえば、工具供給や工具マガジンなど、補機軸は実際の加工にはかかりません。

## 軸アドレス

→ 軸名称を参照してください。

## 軸名称

明確な識別を確実にこなうために、すべてのチャンネルとコントロールシステムの→ 機械軸をチャンネルとコントロールシステム内で一義的な名称で指定してください。→ ジオメトリ軸は X、Y、Z と呼ばれます。ジオメトリ軸 → を中心に回転する回転軸は、A、B、C と呼ばれます。

## 主軸オリエンテーション

指定された角度でワーク主軸を停止します。たとえば、特定の位置で追加の加工を実行するためです。

## 周辺機器

I/O モジュールは、CPU とプロセスの間の接続をおこないます。

I/O モジュールは次のとおりです。

- →デジタル入/出力モジュール
- →アナログ入/出力モジュール
- →シミュレータモジュール

## 象限突起補償

象限切り替え時の輪郭誤差は、案内面での摩擦条件の変化の結果として生じますが、象限突起補償によって、ほとんどすべて取り除けます。突起誤差補正のパラメータ設定は、真円度テストによっておこなわれます。

## 診断

1. コントローラの操作エリアです。
2. コントローラには、サービスのためのテスト機能と自己診断プログラムの両方があります。状態、アラームとサービス表示

## 寸法指定、メトリックとインチ

位置値とピッチ値は、加工プログラムでインチ単位でプログラム指令できます。プログラム指令寸法(G70/G71)に関係なく、コントローラは基本単位系に設定されます。

## 制限速度

最大/最小(主軸)速度:主軸の最大速度は、マシンデータ、→PLC、または→セッティングデータを指定することによって制限できます。

## 接地

接地は、誤動作時でも危険な接触電圧によって活線状態にならない、機器のリンクされた非活性部分の総体としてとられます。

## 先読み

**先読み**機能を使用して、割り当て可能な数の移動ブロックを先読みすることによって、最適な加工速度を実現します。

## 素材

加工される前のワーク。

## 早送り

軸の最大移動速度。たとえば、早送りは、工具が停止位置から→ワーク輪郭にアプローチするとき、または工具がワーク輪郭から後退するときに使用されます。早送り速度は、マシンデータ単位を使用して機械ごとに設定されます。

## 送り速度オーバーライド

プログラム指令速度は、機械操作パネルを介して、または **PLC** からおこなわれた、現在速度設定によってオーバーライドされます(**0 ~ 200%**)。送り速度は、加工プログラムでプログラム可能なパーセンテージ係数(**1 ~ 200%**)によって修正できます。

## 速度制御

ブロックあたりの移動量が非常にわずかな場合に、許容できる移動速度を実現するために、複数のブロックわたる先行解析(→先読み)を指定できます。

## 多項式補間

多項式補間を使用すると、さまざまな曲線特性を生成できます。たとえば、**直線**、**放物線**、**指数関数**などです(SINUMERIK 840D sl)。

## 中間ブロック

選択した工具補正(G41/G42)による移動は、一定の中間ブロック(補正平面で軸移動のないブロック)によって中断できます。これにより、工具補正を引き続き正しく実施できます。コントローラが先読みする中間ブロックの許容数は、システムパラメータで設定できます。

## 直線軸

回転軸とは異なり、直線軸は直線を描きます。

## 直線補間

工具は、ワークを加工する間、終点に向かって直線に沿って移動します。

## 動的フィードフォワード制御

追従誤差による輪郭の不正確さは、動的加速度依存フィードフォワード制御を使用して現実的に、取り除くことができます。この結果、高い軌跡速度でも優れた加工精度を達成できます。フィードフォワード制御は、パートプログラムによって軸毎に選択と解除できます。

## 同期

特定の加工時点で異なるチャネルの順序調整をするためのパートプログラム内の命令。

## 同期軸

同期軸は→ガントリ軸で、その指令位置が→マスタ軸の移動から連続的に得られるため、マスタ軸と同期して移動します。プログラマとオペレータの観点から見ると、同期軸は「存在しません」。

## 同期軸

同期軸は、その軌跡を移動するために、ジオメトリ軸の軌跡を移動するのと同じ時間が掛かります。

## 非同期サブプログラム(Asynchronous SUBprogram)

割り込み信号(たとえば、「高速 NC 入力」信号)を使用して、現在のプログラム状態と非同期で(無関係に)起動できるパートプログラムです。

## 標準サイクル

標準サイクルは、下記の用途で頻繁に繰り返される加工運転のために設けられています。

- 穴あけ/フライス加工用
- 旋削加工用

使用可能なサイクルは、[プログラム]操作エリアの[サイクルサポート]メニューに表示されています。必要な加工サイクルを選択すると、値の割り当てに必要なパラメータがブレンテキストで表示されます。

## 変数定義

変数定義には、データタイプと変数名称指定が含まれます。変数名称を使用して、変数の値にアクセスできます。

## 補間器

パートプログラムで指定された最終位置の情報に基づいて、個々の軸で実行される移動の中間値を定義する → NC の論理ユニットです。

## 補間型補正

機械の機械的な誤差を、→ピッチ誤差、真直度補正、直角度補正、熱変位補正などの補間型補正機能により補正します。

## 補助機能

補助機能を使用すると、→パートプログラムは→パラメータを→PLC に伝送でき、それによって、工作機械メーカーによって定義された動作を起動します。

## 補正テーブル

補間点を含むテーブルです。これは、基準軸上の指定された位置に対する補正軸の補正値を提供します。

## 補正軸

補正値によって補正された指令値または現在値を持つ軸です。

## 補正值

エンコーダによって測定された軸位置と、プログラム指令された目標の軸位置との差です。

## 輪郭

→ワークの輪郭

## 輪郭からの高速リトラクト

割り込みが発生すると、CNC 加工プログラムによって動作を開始して、現在加工中のワーク輪郭から工具を素早く退避させることができます。退避角度と退避距離はパラメータ設定することもできます。割り込みルーチンは以下の高速リトラクトでも実行することができます。

## 連続軌跡モード

連続軌跡モードの目的は、パートプログラムブロック境界での → 軌跡軸の実質的な減速を回避して、可能な限り同じ軌跡速度に近い速度で次のブロックに移ることにあります。



# 索引

## \$

\$A\_DP\_IN\_CONF, 71  
\$A\_DP\_IN\_STATE, 71  
\$A\_DP\_IN\_VALID, 71  
\$A\_DP\_OUT\_CONF, 71  
\$A\_DP\_OUT\_STATE, 71  
\$A\_DP\_OUT\_VALID, 71  
\$A\_IN, 39, 40  
\$A\_INA, 39, 47  
\$A\_INCO, 56  
\$A\_OUT, 39, 42  
\$A\_OUT[n], 746  
\$A\_OUTA, 39, 50  
\$AA\_ACT\_INDEX\_AX\_POS\_NO, 1001  
\$AA\_COUP\_ACT, 938  
\$AA\_COUP\_OFFS, 938  
\$AA\_ENC\_COMP, 336  
\$AA\_ENC\_COMP\_IS\_MODULO, 336  
\$AA\_ENC\_COMP\_MAX, 335  
\$AA\_ENC\_COMP\_MIN, 335  
\$AA\_ENC\_COMP\_STEP, 335  
\$AA\_FIX\_POINT\_ACT; \$AA\_FIX\_POINT  
\_ACT, 220  
\$AA\_FIX\_POINT\_SELECTED; \$AA\_FIX\_PO  
INT\_SELECTED, 220  
\$AA\_ISTEST, 448  
\$AA\_MOTEND, 829  
\$AA\_PROG\_INDEX\_AX\_POS\_NO, 1001  
\$AC\_AXCTSWA, 129  
\$AC\_AXCTSWE, 129  
\$AC\_FIFO, 622  
\$AC\_ISTEST, 448  
\$AC\_RETPOINT, 902  
\$AN\_AXCTAS, 129  
\$AN\_AXCTSWA, 129  
\$AN\_CEC, 344  
\$AN\_CEC\_DIRECTION, 346  
\$AN\_CEC\_INPUT\_AXIS, 344  
\$AN\_CEC\_INPUT\_NCU, 346  
\$AN\_CEC\_IS\_MODULO, 346  
\$AN\_CEC\_MAX, 345  
\$AN\_CEC\_MIN, 345  
\$AN\_CEC\_MULT\_BY\_TABLE, 346  
\$AN\_CEC\_OUTPUT\_AXIS, 345  
\$AN\_CEC\_STEP, 345  
\$AN\_LAI\_AX\_IS\_AXCTAX, 129  
\$AN\_LAI\_AX\_IS\_LEADLINKAX, 129

\$AN\_LAI\_AX\_IS\_LINKAX, 130  
\$AN\_LAI\_AX\_TO\_IPO\_NC\_CHANAX, 130  
\$AN\_LAI\_AX\_TO\_MACHAX, 130  
\$AN\_REBOOT\_DELAY\_TIME, 424  
\$P\_COUP\_OFFS, 939  
\$P\_GWPS, 1049  
\$P\_ISTEST, 448  
\$TC\_DP1, 1019  
\$TC\_DPC1...10, 1020  
\$TC\_TPC1...10, 1026  
\$TC\_TPG\_DRSPATH, 1025  
\$TC\_TPG\_DRSPROG, 1025  
\$TC\_TPG3, 1023  
\$TC\_TPG4, 1023  
\$TC\_TPG5, 1024  
\$TC\_TPG8, 1024  
\$TC\_TPG9, 1025  
\$VA\_COUP\_OFFS, 938

## 1

1次元  
指令値選択(\$SAC\_MEAS\_TYPE = 19), 682

## 2

2次元  
指令値選択(\$SAC\_MEAS\_TYPE = 20), 683

## 3

3次元  
指令値選択(\$SAC\_MEAS\_TYPE = 21), 685  
3次元プローブ, 642

## 6

6FC5800-0AM28-0YB0, 536

## A

ASCALE, 906  
ATRANS, 906  
AUTOMATIC モードのハンドルオーバーライド  
軌跡指令, 252  
速度オーバーライド, 253  
AXCTSWE, 127

AXCTSWEC, 127  
AXCTSWED, 127  
AXTOCHAN, 471

## C

CLEARM, 431  
COARSEA, 823  
CORROF, 270  
CP, 577  
CT, 127

## D

### DB10

DBX100.0 - 4, 193  
DBX100.6, 193  
DBX100.7, 193  
DBX101.0 - 4, 193  
DBX101.6, 193  
DBX101.7, 193  
DBX102.0 - 4, 193  
DBX102.6, 193  
DBX102.7, 193  
DBX110.0 ~ 113.7, 745  
DBX114.0 ~ 117.7, 745  
DBX97.0 - 3, 193  
DBX98.0 - 3, 193  
DBX99.0 - 3, 193

DB10 DBX108.7, 423

### DB10, ...

DBX107.0, 619  
DBX107.1, 619

### DB11

DB6.2, 169  
DBX(n\*20+6).2, 243  
DBX186.2, 169  
DBX26.2, 169  
DBX6.3, 423

### DB19

DBX0.7, 244  
DBX20.7, 244

### DB21、...

DBX0.3, 265, 268, 851  
DBX15.0, 196  
DBX19.0, 196  
DBX23.0, 196  
DBX24.3, 265, 268  
DBX29.4, 573  
DBX30.0 - 2, 262  
DBX30.4, 262

DBX30.6, 229  
DBX31.5, 196  
DBX317.6, 573  
DBX323.0, 196  
DBX327.0, 196  
DBX33.3, 255  
DBX331.0, 196  
DBX335.0, 197  
DBX339.0, 197  
DBX343.0, 197  
DBX377.4, 244, 247  
DBX377.5, 244, 247  
DBX377.6, 230  
DBX38.0, 763  
DBX39.5, 197  
DBX40.5, 194  
DBX40.7, 184, 188, 195  
DBX43.0, 197  
DBX46.5, 194  
DBX46.7, 184, 188, 195  
DBX49.0, 197  
DBX52.5, 194  
DBX52.7, 184, 188, 195  
DBX55.0, 197  
DBX7.1, 952  
DBX7.4, 951

DB21、... DBX0.3, 263

DB21、... DBX30.0 - 2, 263

DB21、... DBX30.3, 262

DB21、... DBX7.3, 263

DB21、... DBX7.4, 263

DB21、...; D B 2 1、...

DBX0.3; D B X 0 . 3, 260

DBX0.6; D B X 0 . 6, 259

DBX12.3; D B X 1 2 . 3, 762

DBX3.0; D B X 3 . 0, 761

DBX3.1; D B X 3 . 1, 762

DBX3.2; D B X 3 . 2, 761

DBX3.4; D B X 3 . 4, 762

DBX38.1; D B X 3 8 . 1, 762

DB21、... DBX6.2, 263

DB21、...

DBX12.0 - 2, 192, 193, 254

DBX13.0 - 5, 187

DBX13.6, 183

DBX16.0 - 2, 254

DBX16.0 - 2, 192, 193

DBX17.0 - 5, 187

DBX17.6, 183

DBX20.0 - 2, 192, 193, 254

DBX21.0 - 5, 187

DBX21.6, 183

DBX320.0 - 2, 192, 194  
 DBX321.0 - 5, 187  
 DBX321.6, 183  
 DBX324.0 - 2, 192, 194  
 DBX325.0 - 5, 187  
 DBX325.6, 183  
 DBX328.0 - 2, 192, 194  
 DBX329.0 - 5, 187  
 DBX329.6, 183  
 DBX332.4, 194  
 DBX332.5, 194  
 DBX332.6, 184, 188, 195  
 DBX332.7, 184, 188, 195  
 DBX333.0 - 5, 188  
 DBX333.6, 183  
 DBX336.4, 194  
 DBX336.5, 194  
 DBX336.6, 184, 189, 195  
 DBX336.7, 184, 189, 195  
 DBX337.0 - 5, 188  
 DBX337.6, 183  
 DBX340.4, 194  
 DBX340.5, 194  
 DBX340.6, 184, 189, 195  
 DBX340.7, 184, 189, 195  
 DBX341.0 - 5, 188  
 DBX341.6, 183  
 DBX35.7, 243  
 DBX36.5, 423  
 DBX377.5, 243  
 DBX40.4, 194  
 DBX40.6, 184, 188, 195  
 DBX41.0 - 5, 188  
 DBX41.6, 183  
 DBX46.4, 194  
 DBX46.6, 184, 188, 195  
 DBX47.0 - 5, 188  
 DBX47.6, 183  
 DBX52.4, 194  
 DBX52.6, 184, 188, 195  
 DBX53.0 - 5, 188  
 DBX53.6, 183  
 DB31、... .DBX60.4 - 5, 312  
 DB31、...";"DBX2.2, 186  
 DB31、...  
   DBB19, 948, 956  
   DBX1.3, 948  
   DBX1.4, 945, 949  
   DBX1.5, 950  
   DBX1.6, 950  
   DBX1.7, 178  
   DBX100.2, 872  
   DBX102.0, 313  
   DBX12.4, 896  
   DBX13.0 - 2, 214  
   DBX13.0-2, 216  
   DBX13.3, 223, 224  
   DBX16.4, 951, 956  
   DBX16.5, 951, 956  
   DBX16.7, 950, 956  
   DBX17.6, 956  
   DBX2.0, 744  
   DBX2.1, 945, 948  
   DBX2.2, 253, 254, 950, 956  
   DBX25.0, 313  
   DBX26.4, 954  
   DBX28.0, 872  
   DBX28.1, 814  
   DBX28.2, 814  
   DBX28.3, 872  
   DBX28.4, 872  
   DBX28.5, 857  
   DBX28.6, 814, 857  
   DBX28.7, 857  
   DBX29.5, 946  
   DBX31.4, 929, 939, 945, 946, 949  
   DBX31.5, 927, 946  
   DBX4.3, 928, 950  
   DBX4.6, 951  
   DBX4.7, 951  
   DBX6.2, 814  
   DBX60.4, 925  
   DBX60.4/5, 304  
   DBX60.5, 925  
   DBX60.6, 815  
   DBX60.7, 815  
   DBX61.3, 946  
   DBX62.0, 744  
   DBX62.1, 255, 258  
   DBX62.3, 619  
   DBX63.0, 813, 817  
   DBX63.1, 813  
   DBX63.2, 813, 815, 817  
   DBX64.5, 194  
   DBX64.7, 184, 188, 195, 815  
   DBX65.0 - 5, 188  
   DBX65.6;DBX65.6, 183  
   DBX67.0, 197  
   DBX7.0, 196  
   DBX74.4, 898  
   DBX75.0 - 2, 214  
   DBX75.3 - 5, 215  
   DBX75.6, 223  
   DBX75.7, 223

- DBX76.6, 999  
DBX83.1, 922  
DBX83.3, 1044  
DBX83.5, 955  
DBX83.6, 1045  
DBX83.7, 955  
DBX84.1, 1047  
DBX84.4, 916  
DBX98.0, 928, 930, 946  
DBX98.1, 928, 930, 946  
DBX98.4, 920, 924, 955  
DBX99.0, 918  
DBX99.1, 918  
DBX99.4, 946
- DB31、…  
DBX4.0 - 2, 192, 193, 254  
DBX5.0 - 5, 187  
DBX5.6, 183  
DBX61.2, 423  
DBX64.4, 194  
DBX64.6, 184, 188, 195
- DRF, 264, 267
- F**
- FC18, 834  
FCTDEF, 1033  
FINEA, 823  
FTOCOF, 1037  
FTOCON, 1037
- G**
- G5, 545  
G7, 545  
G75, 213  
GET, 458  
GETD, 459  
GWPS, 1046, 1049  
GWPSOF, 1049  
GWPSON, 1049
- H**
- HMI 操作への影響, 567
- I**
- I/O 範囲, 66  
INCH または METRIC の単位系, 648
- INIT, 431  
IPOBRKA, 823  
IPOENDA, 823  
IS 送り速度停止/主軸停止, 929
- J**
- JOG, 568
- L**
- LEC, 333
- M**
- MD10000, 537  
MD10010, 431  
MD10050, 98  
MD10061, 99  
MD10070, 99  
MD10071:, 99  
MD10088, 424  
MD10185, 99  
MD10200, 407  
MD10210, 407, 907  
MD10240, 341  
MD10260, 141, 337, 341, 741  
MD10270, 741, 994  
MD10280, 431  
MD10300, 36, 49  
MD10310, 36, 53  
MD10320, 37, 49  
MD10330, 37, 53  
MD10350, 36, 46  
MD10360, 36, 44, 46  
MD10361, 46  
MD10362, 36  
MD10364, 36  
MD10366, 36  
MD10368, 36  
MD10398, 61  
MD10399, 62  
MD10450, 742  
MD10460, 742, 752  
MD10461, 742, 751  
MD10470, 746  
MD10471, 746  
MD10472, 746  
MD10473, 746  
MD1048, 749  
MD10480, 747, 750

MD10485, 739, 748, 750  
MD10500, 68  
MD10501, 68  
MD10502, 69  
MD10510, 68  
MD10511, 68  
MD10512, 69  
MD10530, 56  
MD10531, 57  
MD10540, 57  
MD10541, 57  
MD10720, 241  
MD10721, 241  
MD10722, 465, 836  
MD10722;MD 1 0 7 2 2, 466  
MD10735, 217, 224, 226, 235, 242, 245  
MD10900, 993  
MD10910, 992  
MD10920, 993  
MD10930, 993  
MD10940, 995, 1001  
MD11300, 190  
MD11310, 202  
MD11320, 191, 199  
MD11322, 261  
MD11324, 200  
MD11330, 189, 199, 255  
MD11346, 198, 216, 225, 261  
MD11350, 278, 279, 280  
MD11351, 278, 279, 280  
MD11352, 278, 279, 281  
MD11353, 281  
MD11410, 423  
MD11450, 773  
MD12030, 263  
MD12701, 124, 155  
MD12702, 124, 155  
MD12703, 124, 155  
MD12704, 124, 155  
MD12705, 124, 155  
MD12706, 124, 155  
MD12707, 124, 155  
MD12708, 124, 155  
MD12709, 124, 155  
MD12710, 124, 155  
MD12711, 124, 155  
MD12712, 124, 155  
MD12713, 124, 155  
MD12714, 124, 155  
MD12715, 124, 155  
MD12716, 124, 155  
MD12717, 124  
MD12750, 119  
MD13211, 629  
MD17900, 201  
MD17950, 973  
MD18000, 100  
MD18050, 978, 980  
MD18060, 976, 977  
MD18075, 1032  
MD18096, 1020, 1026  
MD18100, 1020  
MD18210;MD 1 8 2 1 0, 978, 980  
MD18230, 974, 976  
MD18351, 692  
MD18352, 975, 976  
MD18353, 975, 976  
MD18600, 638  
MD18720, 89  
MD18960, 821  
MD19250, 976  
MD19251, 979  
MD20060, 537  
MD20070, 922  
MD20080, 537  
MD20100, 649  
MD20110, 141, 247, 787  
MD20150, 247, 649  
MD20150;MD 2 0 1 5 0, 759  
MD20151, 247  
MD20254, 1033, 1042  
MD20350, 1045  
MD20360, 235, 649  
MD20390, 306  
MD20610, 1038  
MD20620, 200  
MD20621, 200  
MD20624, 203, 208  
MD20700, 248  
MD21106, 591  
MD21150, 908  
MD21155, 177, 211  
MD21158, 180, 212  
MD21159, 180, 212  
MD21160, 178, 211  
MD21165, 177, 211  
MD21166, 180, 212  
MD21168, 180, 212  
MD21220, 38, 58  
MD21300, 918, 958  
MD21310, 958  
MD21320, 936, 958  
MD21330, 958  
MD21340, 920, 958

MD24120, 592  
MD26000, 769  
MD26002, 769  
MD26004, 768  
MD26006, 768  
MD26010, 777, 781  
MD26014, 785  
MD26014;MD 2 6 0 1 4, 776, 777  
MD26016, 781  
MD26018, 769  
MD26018;MD 2 6 0 1 8, 759  
MD26020, 763, 770  
MD28050, 622  
MD28260, 622  
MD28264, 622  
MD30300, 898, 906  
MD30310, 894, 898, 899, 907  
MD30320, 896, 907  
MD30330, 900, 901, 1001  
MD30340, 894, 895, 900, 901  
MD30455, 575, 929, 943, 961  
MD30460, 835, 839  
MD30500, 992, 998  
MD30503, 895  
MD30505, 998  
MD30550, 922  
MD30552, 460, 810  
MD30600, 217  
MD31090, 190, 191, 199, 255, 261  
MD32000, 256, 263  
MD32010, 178  
MD32020, 177, 201, 209  
MD32040, 178, 203, 820, 991  
MD32050, 177, 203, 820, 991  
MD32060, 259, 811, 819  
MD32074, 469  
MD32080, 200  
MD32084, 203, 263  
MD32090, 264, 267  
MD32200, 959  
MD32300, 179, 263, 275, 820  
MD32400, 960  
MD32402, 960  
MD32410, 960  
MD32420, 179, 961  
MD32430, 179, 821, 961  
MD32431, 821  
MD32450, 311  
MD32452, 312  
MD32454, 312  
MD32456, 314  
MD32457, 314  
MD32490, 399, 406  
MD32500, 399, 401, 406  
MD32510, 399, 406  
MD32520, 401, 407  
MD32530, 407  
MD32540, 401  
MD32550, 406  
MD32560, 406  
MD32570, 406  
MD32610, 390, 959  
MD32620, 387, 944, 959  
MD32630, 388  
MD32650, 392, 959  
MD32700, 331, 334, 344, 348  
MD32710, 331, 340, 344  
MD32711, 341, 342  
MD32720, 342  
MD32730, 342  
MD32750, 306, 308, 403, 407  
MD32760, 306  
MD32800, 392, 959  
MD32810, 390, 959  
MD32900, 394  
MD32910, 393, 960  
MD32960, 315  
MD34080, 924, 963  
MD34090, 924, 963  
MD34100, 924, 963  
MD34210, 242, 312  
MD35000, 918, 922  
MD35032, 1048  
MD35040, 1048  
MD35220, 962  
MD35230, 962  
MD35242, 962  
MD36100, 896, 898  
MD36110, 896, 898  
MD36120, 896  
MD36130, 896  
MD36610, 423  
MD36620, 423  
MD36933, 205  
MD37200, 932, 959  
MD37210, 932, 959  
MD37220, 932, 959  
MD37230, 932, 963  
MD37500, 818  
MD37510, 817, 818  
MD37511, 817, 818  
MD38000, 334  
MD7200, 963  
MSEC, 333

**O**

OS, 853  
 OSB, 856  
 OSCILL, 863  
 OSCTRL, 854, 855  
 OSE, 855  
 OSNSC, 855  
 OSP, 853  
 OST, 854

**P**

p0680, 629  
 p0684, 629  
 p0922, 629  
 PLC サービス表示, 619, 631  
 PLC による軸入れ替え, 836  
 PLC 軸, 834  
   FC18 による起動, 837  
   PLC のみの制御軸, 835  
   固定割り当て PLC 軸, 835  
 PLC 制御軸  
   MD30460 ビット 6 とビット 7 の制御応答, 839  
   コントロールシステム応答, 839  
 POSP, 863  
 PTP, 572, 577  
 PTPG0, 572, 577  
 PTPWOC, 572, 577  
 PUNCHACC, 771  
 PUTFTOC, 1036  
 PUTFTOCF, 1035

**R**

RangeOffset, 73  
 RELEASE, 457

**S**

SD41010, 189, 199, 255  
 SD41050, 185, 990  
 SD41100, 175, 203, 820, 893, 991  
 SD41110, 177, 201  
 SD41120, 177  
 SD41130, 177, 201, 908  
 SD41200, 177, 209  
 SD41300, 340  
 SD41500, 740

SD41501, 740  
 SD41502, 740  
 SD41503, 740  
 SD41504, 740  
 SD41505, 740  
 SD41506, 740  
 SD41507, 740  
 SD41520, 743, 752  
 SD41521, 743, 751  
 SD41522, 743  
 SD41523, 743  
 SD41524, 743  
 SD41525, 743  
 SD41526, 743, 752  
 SD41527, 743, 751  
 SD41600, 58  
 SD41601, 58  
 SD42100, 840, 850  
 SD42100; S D 4 2 1 0 0, 259  
 SD42101, 840, 850  
 SD42300, 918, 958  
 SD42400, 766  
 SD42402, 769  
 SD42402; S D 4 2 4 0 2, 760  
 SD42404, 770  
 SD42600, 203, 820, 991  
 SD42600; S D 4 2 6 0 0, 825  
 SD42650, 591  
 SD42690, 229, 236  
 SD42691, 229, 236  
 SD42692, 229, 232, 233, 236, 237  
 SD42693, 231, 236  
 SD42694, 232, 236  
 SD43300, 203, 820, 991  
 SD43300; S D 4 3 3 0 0, 825  
 SD43320, 223, 226  
 SD43400, 897  
 SD43410, 897  
 SD43420, 897  
 SD43430, 897  
 SD43600, 832  
 SD43770, 856  
 SD43790, 856  
 SD43900, 305  
 SD43910, 305  
 SD43920, 305, 308  
 SETM, 431, 439  
 START, 431  
 STAT, 578

## T

TMOF, 1045  
TMON, 1045  
TRAANG  
    角度がプログラム指令可能な, 536, 542  
    角度が固定の, 536, 544  
    制限事項, 547  
TRACON, 557  
TRACYL, 521  
TRACYL\_BAE\_TOOL\_t, 520  
TRACYL\_ROT\_AX\_OFFSET\_t, 518  
TRACYL\_Rot\_Sign\_IS\_PLUS\_t, 519  
TRACYL 変換, 604  
TRANS, 906  
TRANSMIT, 502  
TRANSMIT\_ROT\_AX\_OFFSET\_t, 500  
TU, 582

## W

WAITE, 431  
WAITM, 431  
WAITMC, 431, 437  
WAITP, 824  
    揺動軸, 863

## X

x エッジ(\$AC\_MEAS\_TYPE = 1), 651

## Y

y エッジ(\$AC\_MEAS\_TYPE = 2), 653

## Z

z エッジ(\$AC\_MEAS\_TYPE = 3), 655

## ア

アクティブファイルシステム, 970  
アラーム  
    検出器, 333  
    送りねじ, 333

## イ

インクリメンタル手動運転, 185  
インタフェース変数, 633

## オ

オフセットの読み取り, 929  
オンライン工具オフセット, 1030  
オンライン工具径補正:, 1042

## カ

ガイド軸の解放, 457

## ク

クランプ保護領域, 792

## コ

コーナー C1 - C4 (\$AC\_MEAS\_TYPE = 4, 5, 6, 7), 656  
コーナー計測 C1, 658  
コンパレータ入力, 56

## シ

ジオメトリ軸グループのすべての軸は、次のいずれかとなります。;ジオメトリジクグループノスベテノジクハ、ツギノイズレカトナリマス。 , 470  
ジオメトリ軸の定義, 601  
システム変数, 567, 619, 630  
ジョグ  
    固有点へのアプローチ, 213  
ジョグの特記事項, 526  
ジョグモード, 181, 186  
ジョグ後退, 240  
シングルブロック;シングルブロック  
    位置決め軸タイプ 1;イチキメジクタイプ 1, 840  
    位置決め軸タイプ 2;イチキメジクタイプ 2, 841  
    位置決め軸タイプ 3;イチキメジクタイプ 3, 841

## ス

スタティック NC メモリ, 969  
ステップ移動, 189  
スパークアウトストローク, 847

スレーブ主軸  
再同期, 946  
スレーブ主軸補間器, 917

## す

すべての変換, 603

## セ

セットアップアーカイブ, 972

## ソ

ソフトウェア  
-出力スイッチ, 731

## タ

ダイナミック NC メモリ, 969  
ダイナミックバックラッシ, 313  
ダイナミック応答  
-調整, 393

## チ

チャンネル, 806

## テ

テーブル  
補正, 330

## ト

トラバース範囲リミット, 896

## と

といし周速制御(Grinding Wheel Peripheral Speed), 1046

## ハ

ハースギヤ, 998  
バックラッシ, 311  
ダイナミック, 313  
-補正、ダイナミック, 313

上級機能

機能マニュアル, 08/2018, 6FC5397-1BP40-6TA2

パッシブファイルシステム, 970

## ハンドル

ジョグの移動, 190  
軌跡指令, 260  
速度指定, 260  
ハンドルを使った位置指定, 260  
ハンドル接続  
Ethernet, 284

## フ

フィードフォワード制御, 387  
トルク, 391  
速度, 389  
プラス  
-出力スイッチ, 731  
フレーム, 568  
プローブ機能テスト, 717  
プローブ機能テスト例, 717  
ブロック検索, 772  
ブロック切り替え  
位置決め軸タイプ 1, 826  
位置決め軸タイプ 2, 827

## ホ

ポイントツーポイント移動, 576  
ホーム NCU, 111  
ポジションスイッチ信号, 731  
ハードウェア割り当て, 746  
プラス, 745  
マイナス, 745  
リード/遅延時間, 742  
リンク出力, 736  
割り込み制御の出力, 747  
個別出力, 732

## マ

マイナス  
-出力スイッチ, 731

## メ

メインエントリ, 536  
メッセージ選択, 629  
メモリの拡張, 976

## モ

モジュロ 360, 893

## ユ

ユーザー定義連結, 917

## レ

レファレンス点復帰, 266, 269

## ワ

ワーク計測, 633

## 位

位置オフセット

主軸同期, 945

位置タイマスイッチ, 750

位置決め軸, 804

タイプ, 808

チャンネル別信号, 834

ドライラン送り速度, 840

工具オフセット, 824

最大軸数, 822

軸タイプ, 808

軸別の信号, 834

同時, 810, 834

位置決め軸のダイナミック応答, 820

## 移

移動キー, 181

移動指令, 195

移動動作, 841

移動要求, 194

## 一

一方向プローブ, 617

## 円

円セグメントの加工, 231

円の中心点

JOG での円移動用, 236

円移動

ジョグモードの, 228

円弧半径

JOG での円移動用, 236

円筒座標系, 509

円筒補間, 508

## 温

温度

-作用, 302

-補正, 302

## 加

加速軸, 318

加速特性, 771

## 回

回転軸

アブソリュートプログラミング, 900

アブソリュート指令プログラミング, 903

インクレメンタルプログラミング, 902, 905

セットアップ, 906

ソフトウェアリミットスイッチ, 908

ミラーリング, 909

モジュロ 360, 893

モジュロ変換, 903

軸アドレス, 891

送り速度, 892

単位系, 892

## 拡

拡張機能, 547

## 角

角度オフセット POSFS, 924

## 割

割り出し位置

番号, 993

割り出し位置テーブル, 992

- 割り出し軸  
 プログラミング, 1000  
 単位系, 993  
 符号化位置, 1000  
 割り当ての変更, 601
- 起**  
 起動方法, 923
- 軌**  
 軌跡;キセキ  
 -補間器;一ホカンキ, 811  
 軌跡分割;キセキブンカツ, 775
- 共**  
 共通変換マシンデータ, 605
- 極**  
 極座標補間, 604
- 傾**  
 傾斜プランジ研削, 545  
 傾斜角座標変換(TRAANG)  
 角度がプログラム指令可能な, 536, 542  
 角度が固定の, 536, 544  
 傾斜軸変換, 605  
 傾斜面上の WCS の再定義(\$AC\_MEAS\_TYPE = 18), 678
- 計**  
 計算フレーム, 638  
 計算方法, 645  
 計測  
 ウェブの(\$AC\_MEAS\_TYPE = 13), 668  
 シャフトの(\$AC\_MEAS\_TYPE = 9), 663  
 傾斜エッジの(\$AC\_MEAS\_TYPE = 16), 672  
 計測時の移動速度, 716  
 計測信号の遅延時間, 716  
 計測精度, 716  
 穴の(\$AC\_MEAS\_TYPE = 8), 660  
 溝の(\$AC\_MEAS\_TYPE = 12), 665  
 最高移動速度, 716  
 遅延時間の補正, 716  
 平面の角度の(\$AC\_MEAS\_TYPE = 17), 674  
 計測インタフェース  
 入力値, 634  
 計測サイクル, 646  
 計測プローブ  
 -タイプ, 616  
 計測結果  
 MEAC, 631  
 MEAS, MEAW, 619  
 MEASA, MEAWA, 630  
 計測処理, 716  
 計測入力パラメータ, 641  
 計測方法  
 データ管理フレームの割り当て値復元の  
 (\$AC\_MEAS\_TYPE = 27), 693  
 位置の座標変換の(\$AC\_MEAS\_TYPE = 24), 687  
 現在の割り当て値を含むデータ管理フレームのフ  
 ァイルへの保存(\$AC\_MEAS\_TYPE = 26), 692  
 三角形を決定するための(\$AC\_MEAS\_TYPE =  
 25), 690  
 動作中の平面または選択平面の追加回転の定義の  
 (\$AC\_MEAS\_TYPE = 28), 694
- 現**  
 現在位置設定, 631  
 ジオメトリ軸と付加軸の(\$AC MEAS TYPE =  
 14), 669  
 付加軸の(\$AC MEAS TYPE = 15), 671
- 言**  
 言語命令, 763  
 SPN; S P N, 779  
 SPP, 777  
 言語命令;ゲンゴメイレイ  
 SPN; S P N, 776  
 SPP; S P P, 775
- 個**  
 個別のスレーブ主軸補間器, 917
- 固**  
 固定点アプローチ  
 G75 を使用, 213  
 固定点位置, 217  
 固定連結設定, 917

固有点へのアプローチ  
ジヨグモードの, 213

## 誤

誤差  
-熱変位補正曲線, 302

## 工

工具タイプ, 1019  
工具または刃先の選択, 641  
工具監視  
研削用, 1043  
工具計測  
それぞれにレファレンス点を持つ 2 つの旋盤工  
具, 703  
工具測定; コウグソクテイ, 695  
工具長  
(\$AC\_MEAS\_TYPE = 10), 697  
工具長; コウグナガ  
ズームイン機能による測定(\$AC\_MEAS\_TYPE =  
22); ズームインキノウニヨルソクテイ (\$AC\_  
MEAS\_TYPE = 22), 700  
保存済みの位置または実位置による測定  
(\$AC\_MEAS\_TYPE = 23); ホゾンズミノイチマタ  
ハジツイチニヨルソクテイ (\$AC\_MEAS\_  
TYPE = 23), 702  
工具直径の測定(\$AC\_MEAS\_TYPE = 11); コウグチョ  
ッケイノソクテイ (\$AC\_MEAS\_TYPE =  
11), 699

## 溝

溝側補正, 512

## 座

座標変換  
結合, 557  
座標変換重畳, 552  
モーダル座標変換, 566  
座標変換重畳、指令位置, 554

## 再

再同期, 946

## 指

指令値, 637  
指令値変更に対する応答, 958

## 時

時定数  
ダイナミック応答調整, 393

## 自

自動軸入れ替え, 460  
自動的に有効になる事前開始のタイミング, 769

## 軸

軸  
基本, 330  
補助動作, 803  
補正, 330  
軸; ジク  
-補間器; -ホカンキ, 811  
軸コンテナ  
-識別子, 118  
軸タイプ  
位置決め軸の, 808  
軸入れ替え  
シンクロナイズドアクションによる軸入れ替え; シ  
ンクロナイズドアクションニヨルジクイレカ  
エ, 471  
回転フレームのジオメトリ軸, 469  
軸コンテナ回転の解除, 464  
軸入れ替え; ジクイレカエ  
先読み停止処理のない; サキヨミテイシショリノ  
イ, 465

## 主

主軸/軸運転のダイナミックプログラミング, 961  
主軸同期  
位置オフセット, 945

## 手

手動パルス発生器  
HMI の選択, 193  
割り当て, 192

距離の指定, 198  
速度指定, 198  
手動パルス発生器接続 (828D)  
PPU, 278  
PROFIBUS, 279

## 出

出力スイッチ  
-ペア, 731  
-位置, 740  
-信号, 731  
-範囲, 731  
出力値, 633

## 乗

乗算  
テーブル, 340

## 整

整合係数, 318

## 切

切り込み, 847  
切り替え精度  
ポジションスイッチ信号, 746

## 接

接触計測, 632

## 先

先読み停止処理のない軸入れ替えの有効化;サキヨミ  
テイシシヨリノナイジクイレカエノユウコウカ, 466

## 選

選択と選択解除, 547, 566

## 双

双方向プローブ, 617

## 送

送りオーバーライド, 833

## 測

測定インタフェース;ソクテイインタフェース  
出力値;シュツリョクチ, 644  
診断;シンダン, 650

## 速

速度差, 954  
速度制御, 548  
速度制御の最適化, 541

## 多

多方向プローブ(3次元), 617

## 単

単独軸  
NC制御による退避, 818  
NC制御による停止延長, 817  
PLCによる軸制御, 812  
用途, 814

## 遅

遅延時間, 716

## 調

調整軸, 318

## 直

直交 PTP 移動, 571, 576  
直交座標ジョグ移動, 591

## 追

追従誤差, 387

**動**

- 動作終了条件
  - ブロック検索, 833
- 動作領域制限
  - JOG での円移動用, 233

**同**

- 同期モード, 916
  - 解除, 937
  - 膝形加速特性, 962
- 同期状態への到達, 928
- 同時位置決め軸
  - PLC からの起動, 836

**独**

- 独立の単独軸運転
  - NCK 応答, 837
  - PLC 処理, 837

**入**

- 入れ替え可能なジオメトリ軸, 502
- 入力信号の監視, 770
- 入力値, 633
  - 計測タイプ, 634
  - 指令値, 637, 638

**熱**

- 熱変位補正
  - 係数  $\tan(\alpha)$ , 308

**反**

- 反転点, 847

**平**

- 平行移動, 592
- 平行移動オフセット, 638
- 平面選択, 637

**別**

- 別の座標系への変換, 640

**変**

- 変形
  - 温度の作用による, 302

**補**

- 補間器;ホカンキ
  - シャフト;シャフト, 811
  - 軌跡;キセキ, 811
- 補間機能, 841
- 補間点, 330
- 補助インタフェース, 768
- 補正
  - ピッチ誤差, 333
  - 検出器誤差, 333
  - 追従誤差, 387
  - 補間, 330
- 補正軸, 318
- 補正值, 318
- 補足条件, 568

**方**

- 方向座標変換, 603

**命**

- 命令 MEAS、MEAW, 618

**目**

- 目立て, 1030

**揺**

- 揺動, 847
  - シンクロナイズドアクション, 875
  - 非同期, 847
  - 連続切り込み, 847
- 揺動軸, 847

## 輪

輪郭ハンドル, 260

## 例

例, 568

例外, 548

## 連

連結

固定設定, 933

新規の定義の, 933

連結オプション, 917

連結の起動, 923

連続する 2 ストローク間の最短時間間隔, 770

連続運転, 181

連続使用, 186

連続手動運転, 181

