

## 2.4 JOG 運転とハンドル (H1)

### 2.4.1 概略説明

#### マシンのセットアップ

オペレータが軸を手動移動できる設備が必要です。これは、新しい加工プログラムのセットアップ中に、マシン制御パネルのトラバースキーまたは電子ハンドルからマシン軸を移動しなければならない時に、特に必要です。座標のオフセットまたは回転が選択されると、変換されたワーク座標系でハンドホイールの JOG 送りも実行できます。

#### ツールの後退

たとえば、NC-STOP,RESET 操作または停電などの理由でプログラムが途中停止した場合に、マシンオペレータは、現在の加工位置からツールを手動で後退させる必要があります。これは、通常、JOG モードでトラバースキーを操作して行います。加工用の変換座標およびワーク座標系は有効でなければなりません。

#### 内容

次のファンクション説明では、JOG 移動に関連するオプションおよび特性を図示します。

- (JOG での) 連続検出または一時検出モードの連続 JOG 送り
- (JOG での) 連続検出または一時検出モードの相対 JOG 送り (INC)
- (JOG での) 電子ハンドル (付属品) による軸トラバース
- AUTOMATIC でのハンドルオーバーライド (パス設定および速度オーバーライド)

#### DRF

差動レゾルバファンクションを使用すると、AUTOMATIC モードで、電子ハンドルを介して追加相対ゼロオフセットが発生します。このファンクションによって、たとえば、プログラムされたブロック内のツール磨耗の補正が可能になります。

### 2.4.2 詳細説明

#### ■ 一般事項

#### ■ JOG の手動移動の一般特性

(選択したタイプとは無関係に) 通常、JOG モードに適用する特性は、次のとおりです。

#### JOG モード

通常、軸を手動で移動する操作 (以下、「手動移動」と呼びます) が必要な場合には、JOG モードが有効になっていなければなりません。運転モードがアクティブになると、PLC は、インタフェース信号 "Active mode: JOG" (有効モード) (DB11, DBX4.2) を受取ります。

---

参照： 1.10「運転モードの種類とプログラム運転モード (K1)」

## マシンファンクション

JOG モードには、次のように、いくつかの動作（いわゆる「マシンファンクション」）があります。

- 連続 JOG 送り (JOG CONT)
- 相対 JOG 送り (JOG INC)
- ハンドルによる JOG 送り

## ハンドル操作

次のファンクションでも、ハンドル操作が可能です。

- ジオメトリおよびマシン軸の移動のための運転モード JOG-REPOS
- DRF のずれムーブアウトのための運転モード AUTOMATIC
- パスオーバーライド
- 振動バックラッシュポイントの移動

マシンファンクションは、PLC インタフェースを介して選択されます。マシン軸（軸別）およびジオメトリ軸（チャンネル別）には、それぞれ、別々の PLC インタフェースが設けられています。

## 同時移動

JOG で、すべての軸が同時に移動できます。同時に複数の軸を移動した場合は、相互に補間せず各軸独立に移動します。

## 速度

JOG 移動の速度は、次のように、フィードモードに応じた設定値によって決定します。

- 直線フィードレート (G94) が有効の時：  
(SD: JOG\_REV\_IS\_ACTIVE (JOG の回転フィードレート) = 0) :
  - 一般 SD: JOG\_SET\_VELO (G94 での JOG 速度) で、または、回転軸の場合：  
一般 SD: JOG\_ROT\_AX\_SET\_VELO (回転軸の JOG 速度)
  - または、SD: JOG\_SET\_VELO = 0 の場合のみ、軸別 MD: JOG\_VELO (JOG での回転フィードレート)
- 回転フィードレート (G95) が有効の時：  
(SD: JOG\_REV\_IS\_ACTIVE (JOG での回転フィードレート) = 1) :
  - 一般 SD: JOG\_REV\_SET\_VELO (JOG 速度 G95) で
  - または、SD: JOG\_REV\_SET\_VELO = 0 の場合のみ、軸別 MD: JOG\_REV\_VELO (JOG での回転フィードレート)

フィードレート速度の初期設定単位は、mm/min です。また、回転フィードレートの初期設定単位は  $\text{min}^{-1}$  です。

## 早送りオーバーライド

早送りオーバーライドキーがトラバースキーと同時に押された場合、軸別 MD: JOG\_VELO\_RAPID (早送りオーバーライドでの JOG モード軸速度。または、MD: JOG\_REV\_VELO\_RAPID でセットされた回転フィードレートの場合) でセットされた早送り速度で移動します。

## フィードレートオーバーライド

インタフェース信号 "Axial feedrate override active" (有効な軸フィードレートオーバーライド) (DB31, ... DBX1.7) がセットされる場合に、JOG の軸移動速度は、軸フィードレートオーバーライドスイッチからの影響を受けることもあります。

フィードレートオーバーライドスイッチの各設定の割合の割当ては、マシンデータを介してプログラムされます。

スイッチ位置 0% の時、関連マシンデータに 0 を入れると、軸は移動しません。スイッチ位置 0% の時、IS "Axial feedrate override active" (有効な軸フィードレートオーバーライド) は、有効になりません。

フィードレートオーバーライドスイッチ (Gray コード) の位置の代わりに、% 単位の値 (0 ~ 200%) は、PLC からでもセットできます。マシンデータを介して再び選択されます。

参照： 1.18 「送り機能 (V1)」

## 加速

手動移動モードの加減速は、プログラムされた加減速特性に従って行われます。JOG モードで単一軸に適用可能な加速特性は、MD: AX\_JERK\_DEFAULT で定義されます（軸ジャーク制限の初期設定）。

参照： 1.4 「加減速制御 (B2)」

## 表示

JOG モードが選択されると、JOG 基本表示が画面に現れます。この基本表示には、位置、フィードレート、スピンドルおよびツールの値などが表示されています。表示された値の詳細については次を参照してください。

参照： YS840DI ユーザーズマニュアル・操作編

## 座標系

オペレータは、JOG モードの様々な座標系の中から軸を選択して移動することができます。座標系は、次のようなものが使用できます。

- 基本座標系：各軸の手動移動が可能です。
- ワーク座標系：（チャンネル別）ジオメトリ軸の移動のみが可能です。

## ジオメトリ軸

JOG 移動モードで、軸は、マシン軸（軸別）またはジオメトリ軸（チャンネル別）として移動できます。マシン軸の特性は、次の説明で扱います。JOG モードの移動ジオメトリ軸の特殊機能の詳細は、@セクション 2.8.1 @で説明します。

## JOG でのスピンドルの手動移動

スピンドルは、JOG モードで手動移動できます。特に、軸の手動移動に対して、同じ条件が適用されます。連続検出または一時検出モードでトラバースキーを連続的に（または、インクリメントに）使用するかハンドルを操作すれば、スピンドルを JOG モードで移動することができます。軸の場合、モードは、軸／スピンドル別 PLC インタフェースを介して選択して起動します。軸別マシン軸は、スピンドルにも適用します。スピンドルの手動移動の特殊機能の詳細は、2.4.2 「■主軸 JOG 送りの特殊機能」で説明します。

## ■ PLC インタフェースを介して行われる手動移動ファンクション制御

### MMC/NCK/PLC インタフェース

JOG での手動移動に必要な各ファンクションは、通常、PLC ユーザインタフェースを介して起動します。機械メーカーでは、PLC ユーザプログラムを介して、NC システム構成に応じて、JOG モードファンクションを工作機械に適用することができます。



## マシン制御パネル

マシン制御および PLC/NCK の各インタフェースデータブロックの間の信号は、PLC ユーザプログラムによって、マシン別に伝送されます。PLC ユーザプログラムは、軸／スピンドル（マシン軸、ジオメトリ）トラバースキーに対してマシン制御パネルの方向キーがどのように割当てられているかを定義します。

特に手動移動の場合、次のマシン制御パネル信号が重要です。

- JOG モード（選択）
- マシンファンクション INC1 ...
- 方向キー
- フィードレートオーバーライドおよびスピンドル速度オーバーライド

マシン制御パネルから送られる信号の詳細は次を参照してください。

参照： 1.15 「基本的 PLC プログラム (P3)」

## マシンファンクションの選択

JOG モードで有効なマシンファンクションは、次の手段で選択できます。

- マシン制御パネルから                      （たとえば、ユーザ DB インタフェース）
- PLC ユーザプログラムから              PLC/NCK インタフェース

PLC ユーザプログラムは、マシン制御インタフェースのマシンファンクション要求を関連 PLC/NCK インタフェースに伝送します。軸別 NCK/PLC インタフェース（DB31, ... 信号は、セクション 5.3 参照）は、マシン軸／スピンドル用チャンネル別 NCK/PLC インタフェース（DB21, ... 信号は、2.4.5 「■チャンネル別信号」参照）はジオメトリ軸用です。

## ■ 電源オン, モード変更, リセット, ブロックサーチ, 再位置決めに対する応答

リセットすると, ハンドル操作による移動は減速して中断します。

### MCP の選択

次の例では, マシン制御パネルのマシン軸の「連続」マシンファンクションの選択のための運転シーケンスを示します。

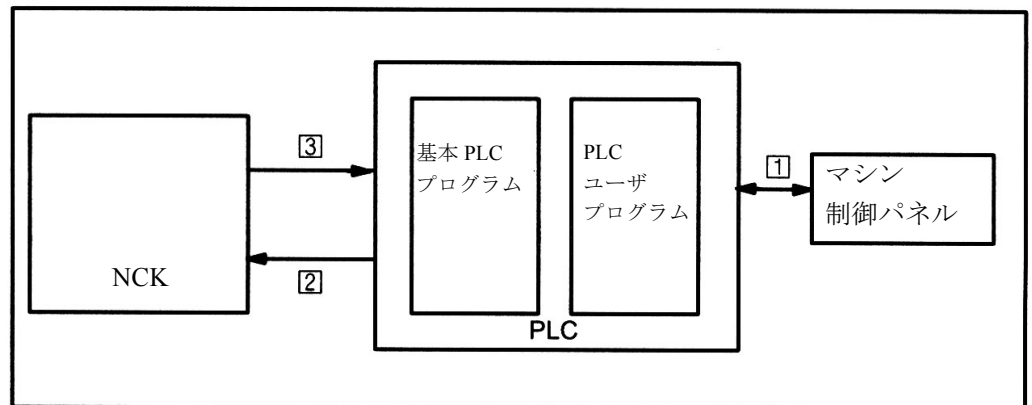


図 2.18 マシン制御パネルからのマシンファンクション選択のための運転シーケンス

#### 運転シーケンス

- (1) オペレータは, マシン制御パネルから, マシン軸のマシンファンクション "Continuous JOGGING" (連続 JOG 送り) を選択します。
- (2) IS "Machine function" (マシンファンクション)  
PLC プログラム (基本プログラムおよびユーザプログラム) は, この IS を論理的に組合わせて, リクエスト "Machine function continuous" (連続マシンファンクション) (DB31, ... DBX5.6) を NCK インタフェースに伝送します。  
これが発生する前, PLC ユーザプログラムは現在のマシンステータスに対し, このリクエストが許容できるかどうかを最初に確認します。
- (3) IS "Active machine function" (有効なマシンファンクション)  
制御内部で, マシンファンクションが選択されます。マシンファンクション "JOG continuous" (連続 JOG) (DB31, ... DBX65.6) が有効になると, NCK から PLC へ信号が戻されます。

マシン制御パネルおよび PLC の間の信号伝送の詳細は次を参照してください。

参照: 1.15 「基本的 PLC プログラム (P3)」

## ■ 連続 JOG 送り

### 選択

JOG モードで, 連続モードは, PLC インタフェースを介して選択されます (IS "Machine function: Continuous" (マシンファンクション: 連続) (DB21-28 DBX13.6, ff))。連続モードが有効になると, IS "Active machine function: Continuous" (有効な

---

マシンファンクション:連続) (DB21-28 DBX41.6, ff) で、信号が PLC に戻されます。

### トラバースキー +/-

「プラス」と「マイナス」のトラバースキーを選択すると、関連軸が選択された方向に移動します。

両方のキーが同時に押されると、移動しないか、移動軸が停止します。

#### 重要

制御がオンになっても、軸がまだ原点復帰されていないため、マシンのリミットまで軸が移動する場合があります。その結果、非常用リミットスイッチが動作します。

ソフトウェアリミットスイッチおよび作業エリア制限は、まだ有効になっていません。

### 移動コマンド +/-

(たとえば、トラバースキーの選択の後) 軸の移動リクエストが有効になると、IS "Travel command+" または "Travel command-" (移動コマンド) (DB21, ... DBX40.7 または DBX40.6) が (選択した移動方向に応じて) PLC に送られます。

## ■ 連続検出モードおよび一時検出モードの区別

### 選択

JOG モードでは、連続検出モードおよび一時検出モードの移動を区別しています。一般 SD:JOG\_CONT\_MODE\_LEVELTRIGGRD (連続検出モードでの連続 JOG) で選択されて、どの軸に対しても有効になります。

### 初期設定

初期設定は、連続検出モードです。

### 連続検出モードの連続移動

#### ファンクション

連続検出モードでは、最初から移動リミットに到達していなければ、トラバースキーを押している間、軸が移動します。トラバースキーを放すと、軸がゼロ速度まで減速して移動を停止します。

## 一時検出モードの連続移動

### ファンクション

(最初の立上がりエッジで) トラバースキーを押してすぐに放すと、軸が、希望の方向に設定速度で移動し始めます。トラバースキーを放しても、移動し続けます。オペレータが停止命令を出すか、(ソフトウェアリミットスイッチに到達するなどの) 制御上の理由が発生すると、軸が停止します。

#### 警告

「一時検出」モードを選択してから、関連方向キーを押してすぐに放すと、複数の軸を同時に起動できます。インタロックは、PLC を介して実行しなければなりません。

### 移動動作の中断

オペレータは、次の方法で移動動作を中断することができます。

- フィードレートオーバーライドを 0% にセットする。
- 軸フィードディスエーブル (PLC インタフェース信号)
- NC STOP または NC STOP 軸／スピンドル

中断の原因が除去されると、軸は移動を続行します。

### 移動動作の強制終了

次のオペレータ入力または監視ファンクションによって、移動動作を停止し、強制終了することができます。

- 同じトラバースキーが再び押された時 (第 2 の立上がりエッジ)
- 逆方向のトラバースキーが押された時
- RESET
- 連続 JOG 送りが選択解除された時
- 最初の有効リミットに到達した時

#### 注意

ソフトウェアリミットスイッチおよび作業エリア制限は、原点復帰が完了した後にのみ起動します。

- 障害発生時

#### (注)

軸の移動中、JOG から AUT または MDI にモードを変更しても内部でディスエーブルされます。

## ■ 連続 JOG 送りの特殊機能

### インデックス軸

インデックス軸として宣言された軸が連続 JOG モードで移動すると、必ず、インデックス位置まで移動します。たとえば、連続検出モードでキーを放しても、軸は、次のインデックス位置まで移動方向に移動します。

参照： 2.15 「割出し軸 (T1)」

## ■ 相対 JOG 送り (INC)

### プログラミングインクリメント

軸毎の移動距離は、いわゆるインクリメント量（別名「相対寸法」）で定義されます。マシンオペレータは、軸を JOG 送りさせる前に、必要なインクリメントをセットしなければなりません。セットは、たとえば、マシン制御パネルなどから行われます。セットされたインクリメント量に関連する IS "Machine function: INC1 to INCvar"（マシンファンクション）(DB31, ... DBB5 ff) は、適切な論理運転の後、PLC ユーザプログラムからセットされます。

### 設定可能なインクリメント

オペレータは、最高 6 種類までのインクリメントサイズを設定できます。これらは、更に、次の 2 つに細分されます。

- 5 件の固定インクリメント：インクリメントサイズが、一般 MD: JOG\_INCR\_SIZE\_TAB（インクリメントサイズ INC / ハンドル）で、すべての軸に対し定義されます。初期設定は、INC1, INC10, INC100, INC1000, INC10000 です。
- 可変インクリメント (INCvar): 可変インクリメントのインクリメント設定も、すべての軸に対し適用し、SD:JOG\_VAR\_INCR\_SIZE（INC / ハンドルの可変インクリメントのサイズ）で実行されます。

### インクリメントの重み

軸 MD: JOG\_INCR\_WEIGHT（INC / ハンドル用のマシン軸インクリメントの重み）は、1 つの JOG インクリメントのパスの重みを定義します。

## ■ 連続検出モードおよび一時検出モードの区別

### 選択

相対 JOG 送りでは、連続検出モードおよび一時検出モードの移動を区別します。選択は、一般 MD: JOG\_INC\_MODE\_LEVELTRIGGRD（INC と REF の連続検出モード）で行います。初期設定は、連続検出モードです。

---

## 連続検出モードでの相対 JOG 送り

### ファンクション

必要な方向トラバースキー（たとえば、+）を押すと、軸はセットしたインクリメントで移動を開始します。インクリメントの移動完了の前に、トラバースキーを放すと、移動が中断されて軸が停止します。同じキーを再び押すと、軸はゼロになるまで残りの距離を移動します。残りの距離がゼロに到達する前に、トラバースキーを放すと、移動が再び中断されます。

インクリメントが完全に移動する前に、逆方向のキーを押しても有効になりません。

### 移動動作の強制終了

全インクリメントまで移動したくない場合、RESET または IS "Delete axial distance-to-go"（軸移動距離の削除）(DB31, ... DBX2.2) で移動動作を強制終了することができます。

## 一時検出モードでの相対 JOG 送り

### ファンクション

（最初の立上がりエッジで）トラバースキーを押すと、軸は、設定インクリメントの限度まで移動します。軸がインクリメントを移動完了する前に、同じトラバースキーを再び押すと、移動動作が中断します（完了しません）。

### 移動動作の中断

連続 JOG 送りの場合と同じ。

### 移動動作の強制終了

次のオペレータ処置または監視ファンクションによって、移動動作が停止して強制終了します。

- （第 2 の立上がりエッジで）同じトラバースキーが同時に押された時
- 逆方向のトラバースキーが押された時
- RESET
- 軸移動距離の削除（PLC インタフェース信号）
- 最初の有効リミットに到達した時

#### 注意

ソフトウェアリミットスイッチおよび作業エリア制限は、最初の原点復帰完了後のみ有効です。

- 現在のインクリメントの選択解除または変更（たとえば、INC100 から INC10 への変更）
- 障害発生時（たとえば、サーボイネーブルのキャンセル）

---

(注)

軸の移動中、モードを JOG から AUT または MDI にモードを変更しても内部でディスエーブルします。

警告

「一時検出」モードを選択して関連方向キーを押してすぐに放すと、複数の軸が同時に起動できます。インタロックは、PLC を介して実行しなければなりません。

## ■ 相対 JOG 送りの特殊機能

### インデックス軸

現在の相対設定値とは関係なく、インデックス軸として宣言された軸 (MD: INDEX\_AX\_ASSIGN\_POS\_TAB) は、トラバースキー "+" が押された後、+ 方向で直近のインデックス位置まで移動します。同様に、トラバースキー "-" が押されると、軸は - 方向で直近のインデックス位置まで移動します。

参照： 2.15 「割出し軸 (T1)」

## ■ JOG でのハンドル移動

### 選択

JOG モードは、有効になっていなければなりません。ハンドルで JOG 送りする時に適用するインクリメント INC1, INC10, ... もセットしてください。相対 JOG 送りの場合、必要なマシンファンクションは、状況に応じて、PLC インタフェースでセットしてください。

### 移動

ハンドルを回すと、関連マシン軸が、回転方向に応じてプラスまたはマイナスの方向に移動します。

### 移動量

ハンドル回転で生じる移動量は、次の係数によって異なります。

- インタフェースで受取ったハンドルパルス数
- 有効なインクリメント (マシンファンクション INC1, INC10, INC100, ... INCvar)
- 一般 MD: HANDWH\_IMP\_PER\_LATCH によるハンドルパルス補正 (1 つの移動止め位置におけるハンドルパルス)
- INE / ハンドルによるインクリメント算出 (軸別 MD: JOG\_INCR\_WEIGHT)

### 移動コマンド +/-

軸の移動中、IS "Travel command+" または "Travel command-" (移動コマンド) (DB31, ... DBX64.7 または DBX64.6) が移動方向に応じて出力されます。

方向キーで軸を移動した後は、ハンドルは使用できません。アラーム 20051 "Jogging with the handwheel not possible" (ハンドルからの JOG 送り禁止) が出力されます。

### ハンドル接続

2 つのハンドルを、同時に接続することができます。この方法で、最高 2 つまでの軸を、ハンドルで同時に移動することができます。

例外:

1 つのハンドルに複数の軸が割当てられている場合には、3 個以上の軸が、ハンドルで移動できます。

### ハンドル割当て

ハンドルおよびジオメトリ (またはマシン軸) の割当てのために、別々の軸別 VDI インタフェース信号が使用されます。

ハンドル 1 または 2 の回転によって移動する軸は、以下のいずれかでセットされます。

- IS "Activate handwheel" (ハンドル起動) (DB31-48, DBX4.0 ~ DBX4.2) で、PLC ユーザインタフェースを介して  
(ジオメトリ軸で: DB21, ... DBX12.0 - 12.2 ff)



- ・メニューガイド操作 (MMC) によって

JOG モード基本メニューでソフトキー Handwheel (ハンドル) を操作すると、ウィンドウ "Handwheel" (ハンドル) が表示されます。ここで、各ハンドルに軸を割り当てることができ、ハンドルをイネーブルまたはディスエーブルにすることができます。

割り当ては、PLC ユーザプログラム論理を介して、PLC インタフェースの論理を介して伝えられます。この方法で、1 つのハンドルに複数の軸を同時に割り当てることができます。

## ファンクション

ハンドル (付属品) を使用すると、選択した軸を手動で同時に移動できます。ハンドル区画に割り当てられた補正は、インクリメントサイズ補正によって定義されます。座標オフセットまたは回転が選択されると、ハンドルは、変換されたワーク座標系でも JOG 送りすることができます。

## MMC からのハンドル選択

オペレータパネルからハンドルを起動できるように、MMC および PLC の間で別々のユーザインタフェースが設けられています。基本 PLC プログラムが供給するハンドル 1 と 2 用のこのインタフェースは、次の情報を含んでいます。

- ・ハンドルに割り当てられた軸番号。IS "Axis number handwheel n" (軸番号ハンドル n) (DB10, DBB100 ff)
- ・マシンまたはジオメトリ軸についての追加情報。IS "Machine axis" (マシン軸) (DB10, DBX100.7 ff)
- ・ハンドル選択でジオメトリ軸が選択された場合は、ハンドルに割り当てられたチャンネル番号。  
IS "Channel number geometry axis handwheel n" (チャンネル番号ジオメトリ軸ハンドル n) (DB10, DBX97 ff)
- ・ハンドルがイネーブルまたはディスエーブルされる情報。IS "Handwheel deselected" (ハンドル選択解除) (DB10, DBX100.6 ff)

IS "Activate handwheel" (ハンドル起動) は、定義された軸に合わせて、基本 PLC プログラムで、"0" (ディスエーブル) または "1" (イネーブル) にセットされます。

## 入力周波数

ハンドル接続は、最高 100 kHz までの入力周波数でハンドルパルスを受取ることができます。

## 速度

JOG で有効な軸速度設定 SD: JOG\_SET\_VELO (G94 の JOG 速度), MD: JOG\_VELO (JOG 送り軸速度) および SD: JOG\_ROT\_AX\_SET\_VELO (回転軸の JOG 速度) も、ハンドルによる JOG 送りに使用されます。

フィードレートが制限されているため、軸は、特に、大きなパルス補正ハンドル回転に同期できずに、遅延します。

---

## 移動動作の強制終了

移動動作は、RESET または IS "Axial deletion of distance-to-go" (軸移動距離の削除) (DB31, ... DBX2.2) で強制終了します。ハンドルで指令された位置はクリアされ、実位置が指令位置となります。STOP は、移動動作への割込みしか行いません。指令位置と実位置の差は保たれます。移動距離は、START で移動します。

## インクリメント値制限

オペレータは、チャンネル別マシンデータ (MD: HANDWH\_GEOAX\_MAX\_INCR\_SIZE) を介して選択したジオメトリ軸インクリメントのサイズを指定することができます。

選択したマシン軸インクリメントのサイズは、軸マシンデータ (MD: HANDW\_MAX\_INCR\_SIZE) で指定されます。

ハンドルで定義された移動動作は以下によって定義されます。

ジオメトリ軸の場合

- 移動量
- 可変インクリメントのサイズ (SD: JOG\_VAR\_INCR\_SIZE)
- ジオメトリ軸割当て  
(MD: HANDWH\_GEOAX\_MAX\_INCR\_SIZE)

マシン軸の場合

- 移動量
- 可変インクリメントのサイズ (SD: JOG\_VAR\_INCR\_SIZE)
- マシン軸割当て  
(MD: HANDWH\_MAX\_INCR\_SIZE)

## 逆方向の移動

移動方向の変更挙動は、マシンデータ \$MN\_HANDWH\_REVERSE に応じて次のようなものがあります。

- ハンドルを逆方向に動かすと、それを考慮した停止位置が計算されて、その位置に移動します。移動軸が現在の移動方向で減速できる位置より手前に停止位置がある場合、ユニットを減速して逆に移動すると、エンドポイントに近づきます。
- マシンデータに指示されたパルス数の分だけハンドルが逆方向に移動すると、軸は、できるだけ速く減速します。その次の移動指令は減速停止が完了したあとに受け付けられますこの機能は、ソフトウェアバージョン 3.2 以降から使用できます。

## 加減速

ハンドル移動の加速レートは、MD: AX\_JERK\_DEFAULT (軸ジャーク制限の初期設定) でプログラムされた加速特性によって決定します。

参照： 1.4 「加減速制御 (B2)」

## ソフトウェアリミットスイッチでの応答，作業エリア制限

JOG モードで軸を移動する場合，適切なアラームが出力されるまでは，最初の有効制限までしか移動できません。マシンデータ \$MN\_HANDWH\_REVERSE に応じて，（セットポイント側の軸がまだエンドポイントに到達していない場合）次のような挙動になります。

- ハンドルパルスによって生じた距離が，以後の計算に使用する仮想エンドポイントを生じます。この仮想エンドポイントが，たとえば，制限から 10 mm 後方に配置されている場合，これらの 10 mm は，軸が再び移動する前に逆方向に移動しなければなりません。制限の直後に逆方向に移動した場合，仮想移動距離は，移動距離削除またはハンドル割当てによって削除することができます。
- 制限の後方のエンドポイントまで続くハンドルパルスは，すべて無視されます。逆方向に移動すると，直ちに，ハンドルが逆方向に（すなわち，制限から離れて）移動します。この機能は，ソフトウェアバージョン 3.2 以降から使用できます。

## 制限

ハンドルで JOG 送りする時も，制限が有効です。詳細は，2.4.2 「■監視ファンクション」を参照してください。

## 回転フィードレート

JOG モードでは，軸／スピンドルの運転特性も，設定データの設定値 JOG \_REV\_IS\_ACTIVE （JOG の有効回転フィードレート）によって異なります。

- この設定データが有効の場合，軸／スピンドルが，必ず，マスタスピンドルに応じて，回転フィードレート MD JOG \_REV\_VELO （JOG による回転フィードレート）または，MD JOG \_REV\_VELO\_RAPID （早送りオーバーレイでの JOG の回転フィードレートで移動します。
- 設定データが有効でない場合，軸／スピンドルの挙動は，設定データ ASSIGN\_FEED\_PER\_REV\_SOURCE （位置決め軸／スピンドルの回転フィードレート）によって異なります。
- 設定データが有効でない場合，回転でフレームが有効なジオメトリ軸の挙動は，チャンネル別設定データ ASSIGN\_FEED\_PER\_REV\_SOURCE によって異なります（運転モード JOG では，回転でフレームが有効なジオメトリ軸の回転フィードレート）。

## ■ AUTOMATIC モードでのハンドルオーバーライド

### ■ 一般機能性

#### ファンクション

このファンクションを使用すると、Automatic モード (Automatic, MDI) で、ハンドルから直接、JOG 送り軸の移動または速度の変更を行うことができます。ハンドルオーバーライドは、NC 言語要素 FD（パス軸用）および FDA（位置決め軸用）を使用した NC パートプログラムで有効になり、ノンモーダルです。位置決め軸で、移動命令 POSA によって、ハンドルオーバーライドを有効にできます。プログラムされた目標位置に到達すると、ハンドルオーバーライドが再び解除されます。その他の軸が同じ NC ブロックで、同時に補間または移動できます。

平行位置決め軸ファンクションは、PLC ユーザプログラムからでも起動できます。

#### 区別

プログラムされたフィードレートに応じて、次のように区別されています。

- ・ パス定義                      軸フィードレート = 0 (FDA = 0)
- ・ 速度オーバーライド      軸フィードレート > 0 (FD または FDA > 0)

表 2-1 では、ファンクション "handwheel override in Auto-matic mode" (AUTOMATIC モードのハンドルオーバーライド) からどの軸タイプが影響を受けるかを示します。

表 2.4 ファンクション "handwheel override in Auto-matic mode" (AUTOMATIC モードのハンドルオーバーライド) から影響を受ける軸

軸タイプ	速度オーバーライド	パス定義
位置決め軸	FDA[AXi] > 0; 軸	FDA[AXi] = 0
平行位置決め軸	パラメータ "handwheel override active" (有効なハンドルオーバーライド) = 1 軸フィードレート > 0 (FC 15 から)	パラメータ "handwheel override active" (有効なハンドルオーバーライド) = 1 軸フィードレート = 0 (FC 15 から)
パス軸	FD > 0; パス速度に適用。	不可能

#### パス定義

軸フィードレート設定 = 0 の場合（たとえば、FDA[AXi] が 0 としてプログラムされた場合）、プログラムされた目標に向かう位置決め軸は、オペレータが該当するハンドルを回すことで移動します。

ハンドル回転方向が、軸の移動方向を決定します。ハンドルオーバーライドの間、プログラムされた目標位置を越えることはできません。軸は、プログラムされた位置と逆方向に移動させることもできます。逆方向の移動は、軸位置範囲からのみ制約されます。

ブロック遷移は次の時に発生します。

- プログラムされた目標位置に軸が到達した時
- 軸 IS "Delete distance-to-go" (移動距離の削除) (DB31, ... DBX2.2) によって、移動距離が削除された時

この瞬間から、パス定義が解除されるため、以後のハンドルパルスは無効です。

この後、相対プログラム位置は、最後にプログラムされた位置ではなく、中断点を指します。

## 速度オーバーライド

速度オーバーライドに関し、軸フィードおよびパスフィードは次のように区別されます。

- 軸速度のオーバーライド ( $FDA[AXi] > 0$ ):

位置決め軸は、プログラムされた軸フィードレートで、目標位置まで移動します。割当てハンドルによって、ハンドル回転方向に応じて、軸速度を増加したりゼロまで減少したりすることができます。その結果、軸フィードレートは、最高速度によって制限されます。軸は、目標位置から逆方向に移動できません。プログラムされた目標に軸が到達すると、ブロック遷移が直ちに発生します。

この方法で、速度オーバーライドが自動的に再解除されて、以後のハンドルパルスが無効になります。

これは、平行位置決め軸にも適用しますが、この軸の場合には、PLC によって目標位置および速度がセットされます。

- パス速度のオーバーライド ( $FD > 0$ ):

NC ブロックでプログラムされたパス軸は、プログラムされたパスフィードレートで、目標位置まで移動します。速度オーバーライドが有効の場合、最初のジオメトリ軸ハンドルによって発生した速度で、プログラムされたパス速度がオーバーライドされます。

プログラムされた目標に到達すると、ブロック遷移が発生します。

ハンドルの回転方向に応じて、パス速度が増加するか、ゼロまで減少します。

ハンドルオーバーライドで、移動方向を反転できません。

## アプリケーションの例

"Handwheel override in AUTOMATIC mode" ファンクション (AUTOMATIC モードのハンドルオーバーライド) は、頻繁に、研削盤に使用されます。たとえば、オペレータは、ハンドル (パス定義) で、往復砥石をワークに配置することができます。接触させたあと、トランズバース移動を終了させて、"Delete distance-to-go" (移動距離の削除) を起動すると、ブロック遷移が開始します。

## 要求事項

"Handwheel override in AUTOMATIC mode" (AUTOMATIC モードのハンドルオーバーライド) を起動するには、次の条件が満たされていなければなりません。

- 対象の軸にハンドルを割当てて必要があること。
- 割当てハンドルに合わせて、パルス補正がプログラムされていること。

## ハンドル割当て

接続されたハンドルが軸に割当ててる作業は、セクション 2.4 に説明した手順（すなわち、IS "Activate handwheel"（ハンドル起動）の PLC ユーザインタフェース (DB31, ... DBX4.0 ~ DBX4.2)) と同様に行われるか、または、オペレータがメニューの指示に従って入力して行われます。

ハンドルオーバーライドに合わせて軸が割当てられた場合でも、ハンドルが割当てられていなければ、次アラームが発生することがあります。

- 速度オーバーライドで  
軸は、プログラムとおりの速度で移動します。（応答なしで）自己確認応答アラームが出力されます。
- パス定義で  
速度がゼロのため、移動動作が行われません。（応答なしで）自己確認応答アラームが出力されます。

（注）

速度オーバーライドがパス軸にかけられていると、最初のジオメトリ軸のハンドルだけがパス速度で動作します。

## ハンドル補正

1 クリック分だけハンドルを前進した結果発生した軸の移動量は、いくつかの係数によって異なります（2.4.2 「■ JOG でのハンドル移動」参照）。

- 選択したインクリメントサイズ（一般 MD: JOG\_INCR\_SIZE\_TAB[5] または軸 SD: JOG\_VAR\_INCR\_SIZE）
- インクリメント係数（軸 MD: JOG\_INCR\_WEIGHT）
- 1 クリック毎のパルス倍率  
（一般 MD: HANDWH\_IMP\_PER\_LATCH）

たとえば、マシンファンクション INC1 および上記マシンデータの標準設定が選択された場合、ハンドルを 1 クリック回すたびに、軸が、0.001 mm 移動します。

速度オーバーライドで、一定期間、ハンドルを使用していると、対象のパスによって速度が発生します。

## 例

オペレータが、100 パルス／秒で、ハンドルを回転させて、  
選択したマシンファンクションは、INC100 であり、  
初期設定によるハンドル補正の上記ハンドルマシンデータであれば：

⇒ ハンドル移動量／秒： 10 mm

⇒ 速度オーバーライド： 0.6 m/min

## PLC インタフェース信号

ハンドルオーバーライドが有効になると、次のような PLC への信号が "1" にセットされます。

- 位置決め軸で：  
IS "Handwheel override active"（有効なハンドルオーバーライド）(DB31, ...

DBX62.1)

- 平行位置決め軸で：

IS "Handwheel override active" (有効なハンドルオーバライド) (DB31, ... DBX62.1)

- パス軸で：

IS "Handwheel override active" (有効なハンドルオーバライド) (DB21, ... DBX33.3)

パス定義で、IS "Traverse commands +/-" (移動コマンド) (DB31, ... DBX64.6 および 64.7) は、移動方向に応じて PLC に出力されます。

## 制限

軸制限 (SW リミットスイッチ, HW リミットスイッチ, 作業エリア制限) は、ハンドルオーバライドと組合わせた場合に有効です。パス定義では、プログラムされた目標位置までしか、軸が、ハンドルによって移動方向に移動できません。

その結果生じた速度は、軸 MD: MAX\_AX\_VELO (最高軸速度) によって制限されます。

## NC STOP / オーバライド = 0

フィードレートオーバライドが 0% にセットされるか、ハンドルオーバライドが有効の時に NC STOP が開始した場合には、次のものが適用します。

- パス定義で

その間に到達したハンドルパルスは、積算されて内部にストアされます。

NC 開始またはフィードレートオーバライドが 0% より大きい時、ストアされたハンドルパルスは起動します (すなわち、移動します)。

ただし、(IS "Activate handwheel n" (ハンドル n の起動) (DB21, ... DBX12/16/20) で) ハンドルがはじめて停止すると、ストアされたハンドルパルスがキャンセルされます。

- 速度指定で

その間に到達したハンドルパルスは、積算されずに、有効になりません。

## ■ ハンドルオーバライドのプログラミングと開始

### 一般注記

ハンドルオーバライドが NC 言語要素 FD (パス軸) および FDA (位置決め軸) でプログラムされる場合には、次の点に注意してください。

- FDA および FD は、ノンモーダルです。

位置決め軸については、例外があります。移動命令 POSA がプログラムされる場合、ブロック遷移が位置決め軸の影響を受けないため、ブロック範囲を超えても有効にできます。

- ハンドルオーバライドが FDA または FD で起動した場合、NC ブロックにおいて、位置決め軸またはパス軸の位置がプログラムされなければなりません。プログラムされた目標位置に到達すると、ハンドルオーバライドは、再び解除されます。
- 同一の NC ブロック内の FDA, FD (FA) および F をプログラムすることはできません。

- ・ 位置決め軸が、インデックス軸であってはいけません。

## 位置決め軸

ハンドルオーバーライド構文： **FDA[AXi] = [ フィード値 ]**

### 例 1

速度オーバーライドを設定します。

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を起動します。  
U の軸速度は、100 mm/min です。

POSA[V]=20          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を起動します。  
V の軸速度は、150 mm/min です。

### 例 2

同じ NC ブロック内のパス定義および速度オーバーライドを起動します。

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 のパス定義を起動します。

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を起動します。  
V の軸速度は、150 mm/min です。

## パス軸

ハンドルオーバーライド構文： **FD = [ フィード値 ]**

パス軸の "handwheel override in Automatic mode" (Automatic モードでのハンドルオーバーライド) をプログラムするには、次の条件が満たされていなければなりません。

- ・ グループ 1 からのコマンドが有効であること : G01,G02,G03,CIP
- ・ イグザクトストップが有効であること (G60)
- ・ mm/min or inch/min 単位の直線フィード (G94)

これらの条件が満たされているかどうかは制御で確認されて、1 つでも満たされていない場合には、アラームが出力されます。

### 例 3

速度オーバーライドを起動します。

N10 G01 X10 Y100 Z200 FD=1500...

X10 Y100 Z200      パス軸 X,Y および Z の目標位置

FD=1500            パス軸の速度オーバーライドを起動します。  
パス速度は、1500 mm/min です。



## 平行位置決め軸

平行位置決め軸のハンドルオーバーライドは、FC15 を介して、PLC から該当パラメータ "Handwheel override on" (ハンドルオーバーライドオン) をセットして起動します。

パラメータ速度 (F\_ 値) に 0 が割当てられている場合、起動したハンドルオーバーライドは、パス定義として動作します (すなわち、フィードレートは、軸 MD: POS\_AX\_VELO (位置決め軸速度の初期設定) からは得られません)。

参照： 2.10 「補正位置決め軸 (P2)」

1.15 「基本的 PLC プログラム (P3)」

## ■ Automatic モードのハンドルオーバーライドの特殊機能

### 速度表示

ハンドルオーバーライドの速度表示は、次の値を示します。

- ・ 設定速度: プログラムされた速度
- ・ 実際速度: 結果として生じた速度 (ハンドルオーバーライドも含めて)

### トランズバース軸への影響

軸がトランズバース軸として定義されていて、しかも DIAMON が有効の場合、ハンドルパルスは、直径値であると解釈されて、ハンドルオーバーライドが有効の時の直径値として移動します。

### ドライランフィードレート

(IS "Activate dry run feedrate" (ドライランフィードレートの起動) (DB21, ... DBX0.6=1) で) ドライランが有効の場合、ドライランフィードレートが、必ず適用されます (SD: DRY\_RUN\_FEED)。

この方法で、軸は、パス定義 (FDA[AXi]=0) でハンドルオーバーライドが有効になっています (すなわち、パス定義が有効ではありません) が、ハンドルがプログラム目標位置に影響を及ぼさずに、ドライランフィードレートで、軸が移動します。

### DRF 有効

"Handwheel override in Automatic mode" (Automatic モードのハンドルオーバーライド) を起動する場合には、ファンクション DRF (IS "Activate DRF" (DRF の起動) (DB21, ...DBX0.3=1)) が有効であるかどうか確認しなければなりません。この場合、ハンドルパルスによって、軸の DRF オフセットが生じます。したがって、オペレータは、最初に DRF を解除しなければなりません (2.4.2 「■ DRF」参照)。

### フィードレートオーバーライド

フィードレートオーバーライドは、ハンドルによって生じた移動に影響を及ぼしません (例外: 0%)。それは、プログラムされたフィードレートにのみ作用します。ハンドルによるパス定義および急速 JOG 送りで、軸は、軸が遅延しないようにするための、(特に、大きなハンドルパルス補正に合うような) ハンドル回転の同期

を行えない場合もあります。

## ■ エンコーダ入力を利用した第 3 のハンドル (840D,810D)

### ファンクション

今まで：

マシンデータで有効にできるドライブのエンコーダ入力を介して、第 3 のハンドルを接続することもできます。

第 3 のハンドルは、たとえば、輪郭ハンドルとして使用できます。

### 3 つのハンドルの比較

3 つのハンドルは、運転手順および機能性が共通しています。

第 3 のハンドルが他の 2 つと異なる点は、接続方法だけです。

### 実際値入力へのハンドルの接続

ハンドルからの信号は、次の配線（トラック A,\*A,B,\*B,5V,0V）で、実際値入力に送られなければなりません。

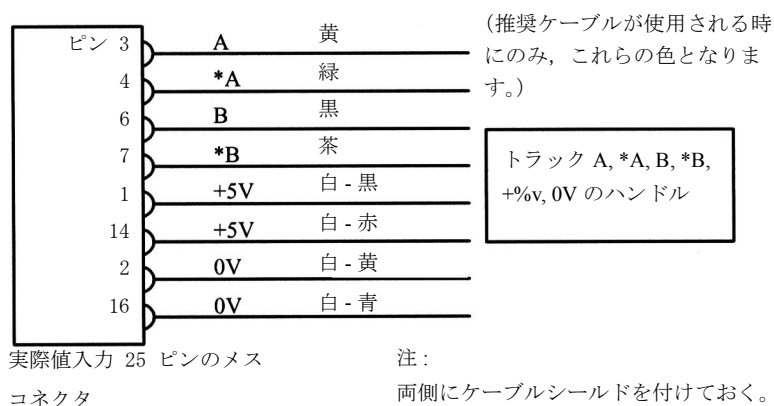


図 2.19 実際値入力へのハンドルの接続

推奨事項：

- "Actual-value cable for encoder with voltage signals"（電圧信号によるエンコーダの実際値ケーブル）(6FX2002-2CG00- ...) を使用してください。
- 円形コネクタでケーブルを離し、外側シールドを取除いて、グラウンドに接続してください。
- 図 2-2 に示すように、ハンドル信号を送ってください。

### 起動, マシンデータおよびインタフェース信号

第 3 のハンドルの起動には、次のマシンデータおよびインタフェース信号が必要です。

- マシンデータ

MD 11340: \$MN\_ENC\_HANDWHEEL\_SEGMENT\_NR

MD 11342: \$MN\_ENC\_HANDWHEEL\_MODULE\_NR

MD 11344: \$MN\_ENC\_HANDWHEEL\_INPUT\_NR

- インタフェース信号

IS "Channel number for handwheel 3" (ハンドル 3 のチャンネル番号) (DB10, DBX99.0, 99.1, 99.2)

IS "Axis number for handwheel 3" (ハンドル 3 の軸番号) (DB10, DBX102.0 ~ 2.4)

IS "Define handwheel 3 as contour handwheel" (輪郭ハンドルとしてのハンドル 3 の定義) (DB10, DBX102.5)

IS "Handwheel 3 selected" (選択したハンドル 3) (DB10, DBX102.6)

IS "Machine axis" (マシン軸) (DB10, DBX102.7)

IS "Activate handwheel 3" (ハンドル 3 の起動) (DB21, ..., DBX12.2, 16.2, 20.2)

IS "Handwheel 3 active" (有効なハンドル 3) (DB21, ..., DBX40.2, 46.2, 52.2)

IS "Activate handwheel 3" (ハンドル 3 の起動) (DB31, ..., DBX4.2)

IS "Handwheel 3 active" (有効なハンドル 3) (DB31, ..., DBX64.2)

## 補足条件

- 測定回路接続のパラメータ化が不正確であるか、電源オンで、ハードウェア構成成品が欠けていることを通知するため、アラーム "Handwheel %1 configuration incorrect or inactive" (ハンドル %1 構成異常) が出力されます。
- 実際値エンコーダとは異なり、ハードウェア接続時にエンコーダモニタが設けられていません。ハードウェア障害またはケーブル切断時に、ハンドルパルスが消えます。
- 実際値入力の割当ての重複を阻止するインタロックがありません。すなわち、理論的には、位置または速度検出用の実際値エンコーダまたは「第 3」のハンドルに、入力を割当てることができます。この場合、「ハンドル」パルスは、1 回転当たりのエンコーダパルス数（粗インクリメント）に従って評価されます。
- ドライブの電源オンまで、第 3 のハンドルは操作できません。

## マシンデータ

11340 MD 番号	\$MN_ENC_HANDWHEEL_SEGMENT_NR 第 3 のハンドル: バスセグメント	
デフォルト値: 1	最小入力リミット: 1	最大入力リミット: 1
変更が有効になるための条件: 電源オン	保護レベル: 0/0	単位: -
データタイプ: BYTE	適用開始 SW バージョン: 840D SW4.1 810D SW2.1	
意味:	第 3 のハンドルのアドレッシングのためのバスセグメント番号 (エンコーダ接続): 1: 611D ドライブバス 0,2,3: 未使用	
関連性	\$MN_ENC_HANDWHEEL_MODULE_NR \$MN_ENC_HANDWHEEL_INPUT_NR	

11342 MD 番号	\$MN_ENC_HANDWHEEL_MODULE_NR 第 3 のハンドル: ドライブ番号/測定回路番号	
デフォルト値: 0	最小入力リミット: 0	最大入力リミット: NCU 572: 15
変更が有効になるための条件: 電源オン	保護レベル: 7/2	単位: -
データタイプ: BYTE	適用開始 SW バージョン: 840D SW4.1 810D SW2.1	
意味:	第 3 のハンドルのアドレッシングのためのセグメント内のモジュール番号 (\$MN_ENC_HANDWHEEL_SEGMENT_NR)。 論理ドライブ番号 (MD 13010: DRIVE_LOGIC_NR 参照) は, 611D でここに入力されて, モジュール番号は, (左から右へ数えて) ローカルバスに入力される。	
例外, エラー	= 0: 第 3 のハンドル構成が解除される。この場合, \$MN_ENC_HANDWHEEL_SEGMENT_NR and \$MN_ENC_HANDWHEEL_INPUT_NR での設定が無意味である。	
関連性	MD 13010: DRIVE_LOGIC_NR \$MN_ENC_HANDWHEEL_SEGMENT_NR \$MN_ENC_HANDWHEEL_INPUT_NR	

11344 MD 番号	\$MN_ENC_HANDWHEEL_INPUT_NR 第 3 のハンドル: モジュール/測定回路カードの入力	
デフォルト値: 1	最小入力リミット: 1	最大入力リミット: 2
変更が有効になるための条件: 電源オン	保護レベル: 7/2	単位: -
データタイプ: BYTE	適用開始 SW バージョン: 840D SW4.1 810D SW2.1	
意味:	第 3 のハンドルのアドレッシングのためのモジュールにおける入力番号。 840D: 1/2 = 上/下実際値入力 810D: 必ず, 1	
関連性	\$MN_ENC_HANDWHEEL_SEGMENT_NR \$MN_ENC_HANDWHEEL_MODULE_NR	

## ■ 輪郭ハンドル／ハンドルによるパス定義

### ファンクション

ファンクションが有効の場合、同期軸およびパスのフィードレートは、AUTO および MDI モードで、ハンドルから制御することができます。

### ファンクションの運転特性

輪郭ハンドルと組合わせた時の運転特性は、MD \$MN  
\_HANDWH\_TRUE\_DISTANCE で以下の 2 種類を選択することができます。

- ・ パス定義

軸速度が最大可能値に制限されている場合には、軸がオーバシュートします。  
ハンドルで指定された移動量だけ移動し、パルスは失われません。

- ・ 速度指定

ハンドルは、軸の移動速度のみを指定します。ハンドルが停止すると、軸も停止します。1 つの IPO サイクルの中で、ハンドルからパルスが受取れなくなった場合、直ちに減速します。その結果、軸がオーグシューエしません。ハンドルパルスは、移動量を定義しません。

### フィードレート

mm/min 単位のフィードレートは、次のものによって異なります。

- ・ 一定の期間内に選択されたハンドルのパルス数
- ・ MD\$MN\_HANDWH\_IMP\_PER\_LATCH でセットされるハンドルパルス評価方法
- ・ 起動インクリメント (INC1, 10, 100, ...)
- ・ インクリメントのパス補正（最初の有効ジオメトリ軸の \$MA\_JOG\_INCR\_WEIGHT）

フィードレートは、次のものによって異なります。

- ・ プログラムされたフィードモード (mm/min, mm/rev)
- ・ プログラムされたフィードレート（より高い合成速度が出る場合もあります。）
- ・ G0 ブロックに対する早送り速度
- ・ オーバライド（有効設定 0%, すなわちゼロ速度）

### 移動方向

移動方向は、次のものによって異なります。

- ・ 時計方向の回転：軸は、プログラムどおりの方向に移動します。  
ブロック変更基準（IPO 端）に到達すると、次のブロックが挿入されます（G60 と同じ応答）。
- ・ 反時計方向の回転：軸は、プログラムどおりの方向に移動します。  
この場合、軸は、次のブロックの開始位置までしか移動できません。ハンドルが回転し続けても、パルスは受付けられません。

## ファンクションの起動

ファンクションは、インタフェース信号または NC プログラムによって起動できます。

- インタフェース信号 "Activate handwheel x as contour handwheel"（輪郭ハンドルとしてのハンドル x の起動）からの起動

ファンクションは、次のインタフェース信号を介して起動／停止します。

IS "Activate handwheel x as contour handwheel"（輪郭ハンドルとしてのハンドル x の起動）(DB21, ... DBX30.0, 30.1, 30.2)

- NC プログラムからの起動

輪郭ハンドルは、NC プログラムで、FD=0 によってノンモーダルに起動します。すなわち、ハンドルの前のブロックからの速度 F は、再プログラミングする必要がないように、次のブロックに対し自動的に適用します。

(注)

前の NC ブロックにフィードレートが含まれていない場合には、該当するアラームが出力されます。

1 つの NC ブロックの中の FD および F は、互いに受け合いません（たとえば、アラームを出力します）。

## 輪郭ハンドルのシミュレーション

輪郭ハンドルは、起動時にシミュレートすることができます。インタフェース信号を介して起動すると、ハンドルからはフィードレートを決定しません。代わりに、プログラムされたフィードレート値が適用されます。移動方向は、インタフェース信号を介して指定されます。

IS "Simulation of contour handwheel"（輪郭ハンドルのシミュレーション）(DB21, ... DBX30.3)

IS "Negative direction simulation of contour handwheel"（マイナス方向への輪郭ハンドルのシミュレーション）(DB21, ... DBX30.4)

条件	結果
シミュレーションが選択解除される。	制動ランプに沿って現在の移動が減速する。
移動方向が反転される。	
注意：NC プログラム実行の場合、オーバライドが働く。	

## 補足条件

- 前提条件  
固定フィードレート，ドライランフィードレートはねじ切りまたは同期タップには，選択できません。
- リミット値  
マシンデータに定義された関連軸の加速と減速のリミット値が適用されます。
- 移動動作の割込み  
ファンクションは，NC STOP の後でも選択されたままです。ただし，(MD \$MC\_HANDWH\_CHAN\_STOP\_COND ビット 2 = 1 の場合) ハンドルパルスは，積算されず，無効です。
- DRF  
DRF ファンクションは，選択されると，追加パスオーバーライド効果が生じます。
- 移動距離のチャンネル別削除  
ハンドルから開始した移動が中断します。軸が減速し，次の NC ブロックでプログラムが再起動します。  
輪郭ハンドルは，再び動作します。

## マシンデータ

11346 MD 番号	\$MN_HANDWH_TRUE_DISTANCE ハンドルによるパス定義または速度指定	
デフォルト値: 1	最小入力リミット: 0	最大入力リミット: 1
変更が有効になるための条件: 電源オン	保護レベル: 7/2	単位: -
データタイプ: BYTE	適用開始 SW バージョン: SW4.1	
意味:	<p>ハンドル，輪郭ハンドルまたは FDA=0 による送り運転特性の設定:</p> <p><b>値 = 1:</b> ハンドルパルスは，移動量を定義する。パルスは失われない。速度を最大許容値に制限した結果，軸がオーグシューエする。</p> <p><b>値 = 0:</b> ハンドルパルスは，軸移動速度を指定する。ハンドルが停止すると，軸も直ちに停止する。1 つの IPO サイクルの中で，ハンドルからパルスを受取らなくなった場合，直ちに減速する。その結果，軸はオーグシューエしない。ハンドルパルスは，移動量を定義しない。</p>	

# インタフェース信号

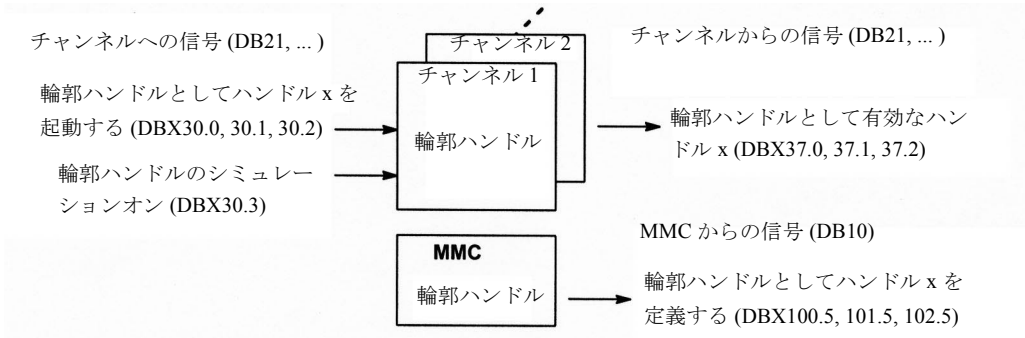


図 2.20 輪郭ハンドルのためのインタフェース信号の概要

DB 21, 22, ... DBX30.0 DBX30.1 DBX30.2 データブロック	輪郭ハンドルとしてハンドル 1 を起動する。 輪郭ハンドルとしてハンドル 2 を起動する。 輪郭ハンドルとしてハンドル 3 を起動する。 チャンネルへの信号 (PLC → NCK)	
エッジ評価：なし	信号更新：周期的	信号が有効となる最初の SW バージョン： SW4.1
説明	これらの信号によって、3 つのハンドルの中の 1 つが、輪郭ハンドルとして選択／選択解除できる。 信号 = 1     ハンドル x が輪郭ハンドルとして選択される。 信号 = 0     ハンドル x が輪郭ハンドルとして選択解除される。 輪郭ハンドルは、ブロックの途中で選択／選択解除することができる。 ハンドルが起動すると、ハンドルからの指示に従い、軸移動が減速してから動作する。 ハンドルが停止する場合には、NC プログラム実行が継続する前に減速する。新しい NC-START の後まで NC プログラムの実行継続が認められていない場合には、輪郭ハンドル停止が、PLC ユーザプログラムで NC-STOP にリンクされていることが必要である。	
例外，エラー	NC-RESET の後でも、信号設定は有効のままである。	
関連性	IS "Handwheel x active as contour handwheel" (DB21, 22, ... , DBX37.0, 37.1, 37.2) (輪郭ハンドルとして有効なハンドル x)	



DB 21, 22, ... DBX30.3 DBX30.4 データブロック	輪郭ハンドルのシミュレーションオン マイナス方向への輪郭ハンドルシミュレーション チャンネルへの信号 (PLC → NCK)																
エッジ評価: なし	信号更新: 周期的	信号が有効となる最初の SW バージョン: SW4.1															
説明	<p>輪郭ハンドルのシミュレーションの起動/停止および移動方向のセットを行う場合、これらの信号は、次のようにセットしなければならない。</p> <table border="1"> <thead> <tr> <th>ビット 3</th> <th>ビット 4</th> <th>意味</th> </tr> </thead> <tbody> <tr> <td>0</td> <td>0</td> <td>シミュレーションオフ</td> </tr> <tr> <td>0</td> <td>1</td> <td>シミュレーションオフ</td> </tr> <tr> <td>1</td> <td>0</td> <td>シミュレーションオン (プログラムされた方向へ)</td> </tr> <tr> <td>1</td> <td>1</td> <td>シミュレーションオン (プログラムされた方向とは逆に)</td> </tr> </tbody> </table> <p>シミュレーションの間、輪郭ハンドルによってフィードレートが決定されずに、プログラムされたフィードレートで輪郭に沿って軸が移動する。 ファンクションが選択解除されると、現在の軸移動が制動ランプに沿って減速する。 移動方向が反転すると、現在の軸移動が制動ランプに沿って減速し、逆方向に軸が移動する。</p>		ビット 3	ビット 4	意味	0	0	シミュレーションオフ	0	1	シミュレーションオフ	1	0	シミュレーションオン (プログラムされた方向へ)	1	1	シミュレーションオン (プログラムされた方向とは逆に)
ビット 3	ビット 4	意味															
0	0	シミュレーションオフ															
0	1	シミュレーションオフ															
1	0	シミュレーションオン (プログラムされた方向へ)															
1	1	シミュレーションオン (プログラムされた方向とは逆に)															
例外, エラー	シミュレーションファンクションは、AUTOMATIC モードの時のみ有効であり、輪郭ハンドルが既に有効の時にのみ起動できる。																

DB 21, 22, ... DBX37.0 DBX37.1 DBX37.2 データブロック	輪郭ハンドルとして有効なハンドル 1 輪郭ハンドルとして有効なハンドル 2 輪郭ハンドルとして有効なハンドル 3 チャンネルからの信号 (NCK → PLC)	
エッジ評価: なし	信号更新: 周期的	信号が有効となる最初の SW バージョン: SW4.1
説明	<p>これらの信号は、どの信号が輪郭ハンドルとして選択されたかを指示する。 信号 = 1    ハンドル x が輪郭ハンドルとして選択される。 信号 = 0    ハンドル x が輪郭ハンドルとして選択解除される。</p>	
例外, エラー	NC-RESET の後でも、信号設定は有効のままである。	
関連性	IS "Activate handwheel x as contour handwheel" (輪郭ハンドルとしてハンドル x を起動する) (DB21, 22, ..., DBX30.0, 30.1, 30.2)	

DB10 DBX100.5 DBX101.5 DBX102.5 データブロック	輪郭ハンドルとしてハンドル 1 を定義する 輪郭ハンドルとしてハンドル 2 を定義する 輪郭ハンドルとしてハンドル 3 を定義する MMC からの信号 (MMC → PLC)	
エッジ評価：なし	信号更新：周期的	信号が有効となる最初の SW バージョン： SW4.1
説明	これらの信号は、どの信号が MMC を介して輪郭ハンドルとして選択されたかを指示する。 信号 = 1      ハンドル x が MMC を介して輪郭ハンドルとして定義される。 信号 = 0      ハンドル x が輪郭ハンドルとして定義されない。 MMC を介して定義されたハンドルが輪郭ハンドルとして動作できるように、該当する信号が、IS "Activate handwheel x as contour handwheel"（輪郭ハンドルとしてハンドル x を起動する）(DB21, 22, ..., DBX30.0, 30.1, 30.2) でゲート制御される必要がある。	
例外，エラー	基本 PLC プログラムの FB1 でのパラメータ HWheelMMC の設定に応じて、これらの信号は、基本プログラムによって送られているか、または、PLC ユーザプログラムによって送られる必要がある。	
関連性	IS "Activate handwheel x as contour handwheel"（輪郭ハンドルとしてハンドル x を起動する）(DB21, 22, ..., DBX30.0, 30.1, 30.2) FB1 パラメータ HWheelMMC	

## ■ JOG モードの特殊機能

## ■ JOG モードのジオメトリ軸

### JOG モードの座標系

JOG モードでは、オペレータは、ワーク座標系において、ジオメトリ軸として宣言された軸を JOG 送りすることもできます。選択された座標オフセットまたは回転は、有効のままです。

JOG 送りジオメトリ軸の特殊機能は、次のとおりです。

ジオメトリ軸は、最後に起動した座標系で移動します。

### 同時 JOG 送り

トラバースキーによって、1 つのジオメトリ軸だけ、連続 JOG 送りまたは相対 JOG 送りが可能になります。複数のジオメトリ軸を JOG 送りしようとすると、アラーム 20062 "Axis already active" (既に有効な軸) が出力されます。軸がジオメトリ軸として定義されていない場合には、アラーム 20060 "Axis cannot be traversed as geometry axis" (軸がジオメトリ軸として移動不可能) が出力されます。

ただし、ハンドル 1,2 および 3 を使用すると、3 つのジオメトリ軸を同時に JOG 送りすることができます。

### PLC インタフェース

ジオメトリ軸には、固有の PLC インタフェース (DB21 ~ 28, DBB12 ~ 23 および DBB40 ~ 56) があり、そこには、軸別 PLC インタフェースと同じ信号が含まれています。

### フィードレート／早送りオーバーライド

チャンネル別フィードレートオーバーライドスイッチおよび早送りオーバーライドは、ジオメトリ軸の JOG 送りの時に有効となります。

### アラーム

ジオメトリ軸が次の条件の下で JOG 送りされている時、アラーム 20062 "Axis already active" (既に有効な軸) が検出されます。

- ・ 軸が、JOG モードで、軸 PLC インタフェースを介して既に移動を開始しています。
- ・ 回転させた座標系のフレームが既に有効であり、トラバースキーによって、この座標系における他のジオメトリ軸が JOG モードで移動しています。

軸がジオメトリ軸として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移動しようとすると、アラーム 20060 となります。

### 使用

変換およびフレームの JOG 送り移動は、有効になっていなければなりません。

## ■ 主軸 JOG 送りの特殊機能

### JOG モードのスピンドル移動

スピンドルも、JOG モードで手動移動できます。特に、軸の手動移動に対し、同じ条件が適用されます。連続または一時検出モードでトラバースキーを連続または相対使用するか、ハンドルを使用することによって、JOG モードでスピンドルが移動できます。ファンクションは、マシン軸の場合と同様に、軸／スピンドル別 PLC インタフェースを介して選択され起動します。軸別マシンも、スピンドルに適用します。

### スピンドルモード

スピンドルは、位置決めモード（スピンドルが位置制御されている場合）またはオープンループ制御モードで JOG 送りすることができます。

### JOG 速度

スピンドル JOG 送り速度は、

- 全スピンドルに対し起動する一般 SD: JOG\_SPIND\_SET\_VELO（スピンドルの JOG 速度）で定義できるか、
- マシンデータ JOG\_VELO（JOG 軸速度）で定義できます。ただし、この MD は、SD: JOG\_SET\_VELO（G94 の JOG 速度）= 0 の場合のみ有効です。

JOG モードでスピンドルが移動する場合には、ギヤステージの最高速度も適用されます。

参照： 1.17 「主軸機能 (S1)」

### 速度オーバーライド

スピンドル速度オーバーライドスイッチを使用すると、JOG モードのスピンドル移動速度を修正することができます。

### JOG 加速

速度制御および位置制御モードでスピンドルが多くのギヤステージを頻繁に使用するため、現在のギヤステージに合わせてプログラムされた加速が、必ず、スピンドル JOG モードで適用されます。

参照： 1.17 「主軸機能 (S1)」

## PLC インタフェース信号

スピンドルが手動移動する場合、NCK および PLC の間で交換される PLC インタフェース信号は、マシン軸と同じ効果を出します。IS "Position reached with exact stop fine or coarse" (正確停止 (微または粗) - 位置到達) (DB31, ... DBX60.7 または DBX60.6) は、スピンドルが位置制御モードにある場合のみセットされます。

スピンドル別インタフェース信号に限って、JOG でスピンドルを移動する場合は次の点に注意してください。

- 次のような、スピンドルへの PLC インタフェース信号は、有効になりません。
  - IS "Invert M3/M4" (M3/M4 反転) (DB31, ... DBX17.6)
  - IS "Set direction of rotation ccw" (反時計方向のセット) または "Set direction of rotation cw" (時計方向のセット) (DB31, ... DBX18.7 または DBX18.6)
  - IS "Oscillation speed" (振動速度) (DB31, ... DBX18.5)
  - IS "Spindle RESET" (スピンドルリセット) (DB31, ... DBX16.7)
- 次のような、スピンドルからの PLC インタフェース信号は、セットされません。
  - IS "Actual speed cw" (時計方向の実際速度) (DB31, ... DBX83.7)
  - IS "Spindle in set range" (設定範囲内のスピンドル) (DB31, ... DBX83.5)

## ■ 監視ファンクション

### 制限

JOG モードでは、次の制限が有効です。

- 作業エリア制限 (軸の基準化が必要です。)
- ソフトウェアリミットスイッチ 1 および 2 (軸の基準化が必要です。)
- ハードウェアリミットスイッチ

最初の有効制限に到達した直後、制御は、移動動作が中断されているか確認します。速度制御は、軸が制限位置 (たとえば、ソフトウェアリミットスイッチ) に正確に停止できるような早いタイミングで減速を開始したか確認します。ハードウェアリミットスイッチが検出されると、"Rapid stop" (急停止) で軸が急停止します。

制限値に到達すると、アラームが検出されます (アラーム 16016, 16017, 16020, 16021)。制御が、これ以上この方向へ移動しないようにします。トラバースキーおよびハンドルをこの方向で使用しても、反応がありません。

#### 重要

ソフトウェアリミットスイッチおよび作業エリア制限は、最初の軸基準化の時のみ有効です。

作業エリア制限およびソフトウェアとハードウェアのリミットスイッチの詳細は次を参照してください。

参照： 1.2「送り軸監視機能及びプロテクションゾーン (S3)」

## 軸後退

逆方向に移動すると、制限位置から軸を後退することができます。

### 機械メーカー

制限位置に到達した軸を後退するファンクションは、機械メーカーによって異なります。機械メーカーの説明書を参照してください。

## 最高速度と加速度

JOG モードで適用される速度と加速度の値は、試運転時に、マシンデータを介して、特定の軸に合うようにプログラムされています。軸の有効値が所定の最高速度、最高加速度以内になるように制御されます。

参照： 1.8「位置指令単位と検出単位 (G2)」

1.4「加減速制御 (B2)」

## ■ 各種機能

### 切換えモード JOG -> AUT または JOG -> MDI

チャンネルの全軸が "Exact stop coarse"（正確停止（粗））で到達した場合のみ、JOG から AUT または MDI に運転モードを切換えることができます。

参照： 1.10「運転モードの種類とプログラム運転モード (K1)」

### JOG の有効回転フィードレート

JOG モードでは、現在のマスタスピンドル速度を基準に（G95 に類似した）回転フィードレートで軸を移動することもできます。ファンクションは、SD:

JOG\_REV\_IS\_ACTIVE（回転フィードレートの JOG）で起動します。

使用するフィードレート値（mm/rev 単位）は、次のように定義することができます。

- フィードレート値 = 0 でない場合、一般 SD: JOG\_REV\_SET\_VELO（G95 の JOG 速度）で。
- SD: JOG\_REV\_SET\_VELO で 0 がセットされている場合、回転フィードレートが、軸マシンデータ JOG\_REV\_VELO（JOG の回転フィードレート）によって決定しますが、早送りオーバーライドの場合にのみ、JOG\_REV\_VELO\_RAPID によって決定します。

マスタスピンドルが定義されていない時、回転フィードレートの JOG で軸を移動すると、アラーム 20055 が出力されます。また、ジオメトリ軸の場合は、アラーム 20065 が出力されます。

---

## トランスバース軸

ジオメトリ軸がトランスバース軸として定義されて、半径プログラミング (MD: DIAMETER\_AX\_DEF (トランスバース軸によるジオメトリ軸)) が選択される場合、JOG の移動には次の点に注意してください。

- 連続 JOG 送り  
連続モードのトランスバース軸の移動には、差がありません。
- 相対 JOG 送り  
選択したインクリメントサイズの距離の半分だけが移動します。たとえば、INC10 では、トラバースキーが押されると、軸が、INC 5 だけ移動します。
- ハンドルによる JOG 送り  
相対 JOG 送りの場合、ハンドルパルスごとに、パスの半分だけが移動します。  
参照： 1.14 「直径指定 / 半径指定 (P1)」

## ■ DRF

### ファンクション DRF

ファンクション DRF（差動レゾルバファンクション）の場合，加工中に，自動モードで，電子ハンドルから追加相対ゼロオフセットを起動することができます。JOG モードでのハンドル割当て，パルス補正などに対して，ハンドルによる手動移動と同一の条件が適用されます（2.4.2「■ JOG でのハンドル移動」参照）。更に，DRF のハンドルによって生じた速度は，軸 MD: HAND\_VELO\_OVERLAY\_FACTOR（ハンドル速度に対する JOG 速度の比率）で，JOG 速度から低下させることができます。

### DRF オフセット

DRF オフセットは，DRF（すなわち，自動モードでハンドルを使った移動動作）によって生じた軸ゼロオフセットです。DRF オフセットは，基本座標系で有効です。

### 注意

DRF によるゼロオフセットは，必ず（すなわち，全モードおよびリセット後）有効です。ただし，パートプログラムで，ノンモーダルに除去できます。

参照：YS840DI ユーザーズマニュアル・プログラミング編

### 使用

DRF は，次のアプリケーションに使用されます。

- NC ブロック内のツール磨耗の補正。NC ブロックの加工時間が長い場合，NC ブロック内（たとえば，大型表面フライス盤）でツール磨耗を手動で補正する必要があります。
- 研削中に，非常に精密な補正をするため
- 非常に単純な温度補正
- 実際値表示に含まれていないオフセット

### 有効 DRF

ハンドルから移動で DRF オフセットが修正できるように，DRF が有効でなければなりません。次の条件が満たされていなければなりません。

- AUTOMATIC モードが選択されています（リセット／割込み／有効のチャンネル）。
- IS "Activate DRF"（DRF 起動）（DB21, ... DBX0.3）= 1

オペレータは，プログラム制御ファンクションで，特定チャンネルに合わせて，DRF オフセットをオフに切換えることができます。これは，IS "DRF selected"（DRF 選択）（DB21, ... DBX24.3）で，MMC によって PLC に通知されます。PLC プログラム（基本 PLC プログラムまたはユーザプログラム）は，論理組合わせの後，このインタフェース信号を IS "Activate DRF"（DRF 起動）として伝送します。



## DRF オフセットの制御

DRF オフセットの修正，削除または読取りは，次の方法で行うことができます（図 2.21 参照）。

- オペレータがハンドルで JOG 送りするか，
- （高級言語で書かれた）NC パートプログラムによって
  - DRF オフセット（軸別）を読取ります。
  - チャンネルの全軸の DRF オフセットを削除します（コマンド "DRFOF"）

参照： 1.15 「基本的 PLC プログラム (P3)」

- PLC ユーザプログラムから
  - DRF オフセット（軸別）を読取ります。

参照： /FB/, P3,

1.15 「基本的 PLC プログラム (P3)」

- オペレータが MMC から
  - DRF オフセット（軸別）を読取ります。

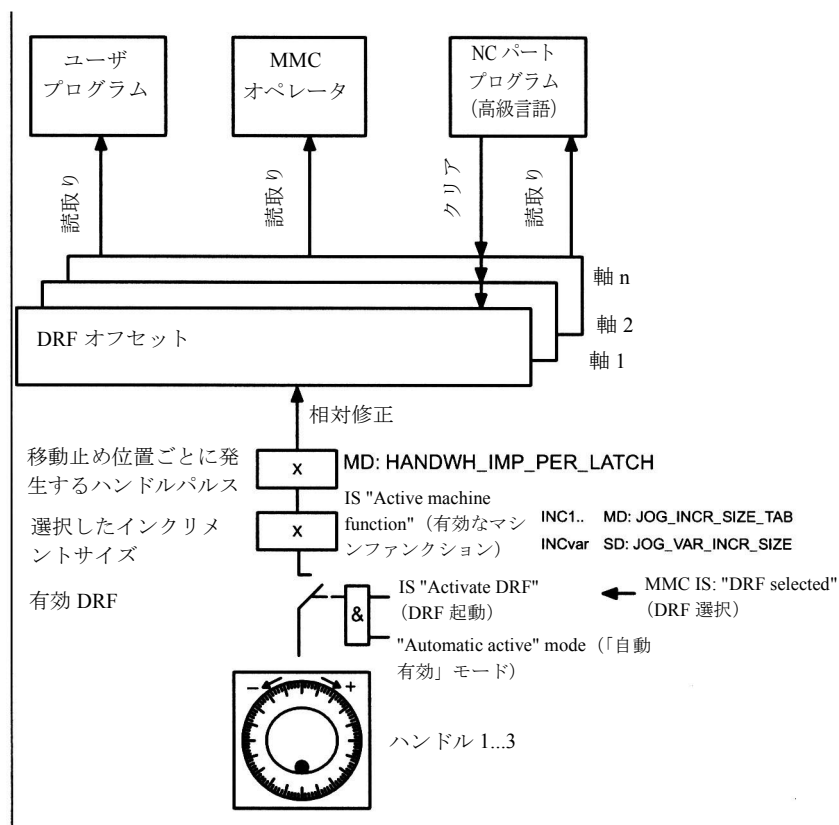


図 2.21 : DRF オフセットの制御

---

(注)

DRF オフセットがクリアされると、軸は移動しません。

## 電源オン

電源オンで、DRF オフセットが削除されます。

## 基準点への接近

DRF オフセット軸が基準化される場合、オフセットは、基準化運転の第 1 段階で削除されます。

(たとえば、G74 による) 軸の基準化の途中に、ハンドルで DRF オフセットを指定することはできません。この時、アラーム 20053 "DRF not possible" (DRF 不可能) が検出されます。

## 表示

DRF でハンドルから軸が移動している間、軸位置表示 (ACTUAL POSITION) は変わりません。現在の DRF オフセットは、DRF ウィンドウで表示できます。

## ■ インストールとスタートアップ

(注)

インストールを開始できる状態にする前に、いくつかの条件が満たされていなければなりません。詳細は、次のものを参照してください。

参照：YS840DI 保守説明書

### マシン／設定データ

マシン、ジオメトリまたは両方の軸は、特定のマシン／設定データがプリセットされた時にのみ、手動で移動することができます。特に、手動移動に対し適用するマシンおよび設定データは、次のリストに記載されています。これらのデータは、2.4.4「■データ説明 (MD, SD)」で、その初期設定とともに説明しています。

### 連続検出モードの連続 JOG

一般 SD: JOG\_CONT\_MODE\_LEVELTRIGGRD

(連続検出モードの連続 JOG)

### 連続検出モードの INC および REF

一般 MD: JOG\_INC\_MODE\_LEVELTRIGGRD

(連続検出モードの INC および REF)

### 速度

軸 MD: JOG\_VELO (JOG 軸速度)

軸 MD: JOG\_VELO\_RAPID (JOG 早送り)

一般 SD: JOG\_SET\_VELO (G94 の JOG 速度)

一般 SD: JOG\_ROT\_AX\_SET\_VELO (回転軸の JOG 速度)

### 回転フィードレート

一般 SD: JOG\_REV\_IS\_ACTIVE

(JOG で有効な回転フィードレート)

軸 MD: JOG\_REV\_VELO (JOG の回転フィードレート)

軸 MD: JOG\_REV\_VELO\_RAPID

(早送りオーバライドによる JOG 回転フィードレート)

一般 SD: JOG\_REV\_SET\_VELO (G95 の JOG 速度)

### 加速

軸 MD: AX\_JERK\_DEFAULT (軸ジャーク制限の初期設定)

---

## 相対／ハンドル

軸 MD: JOG\_INCR\_WEIGHT (INC/ ハンドルのインクリメント補正)

一般 SD: JOG\_VAR\_INCR\_SIZE (INC/ ハンドルの可変インクリメントサイズ)

軸 MD: HANDWH\_VELO\_OVERPLAY\_FACTOR (ハンドル速度 (DRF) に対する JOG 速度の比率)

一般 MD: JOG\_INCR\_SIZE\_TAB[n] (INC/ ハンドル可変インクリメントサイズ)

一般 MD: HANDWH\_IMP\_PER\_LATCH[n] (移動止め位置ごとに発生するハンドルパルス [ ハンドル番号 ])

## スピンドル

一般 SD: JOG\_SPIND\_SET\_VELO (JOG/ ハンドル可変インクリメントサイズ)

### 2.4.3 補足条件

#### ■ ファンクション "Handwheel override in Automatic mode" (Automatic モードのハンドルオーバーライド) の可用性

ファンクションは、次のバージョンから有効になります。

- YS840DI ソフトウェアバージョン 2 以降

### 2.4.4 データ説明 (MD, SD)

#### ■ 一般マシンデータ

11300 DBX37.2	JOG_INC_MODE_LEVELTRIGGERD 連続検出モードの INC および REF	
デフォルト値: 1	最小入力リミット: 0	最大入力リミット: 1
変更が有効になるための条件: 電源オン	保護レベル: 2	単位: -
データタイプ: BOOLEAN	適用開始 SW バージョン: 1.1	
意味:	<p>1: JOG-INC の連続検出モードおよび基準点への接近。 JOG-INC: 必要な方向 (たとえば, +) のトラバースキーを押すと, 軸は, 設定インクリメントで移動を開始する。インクリメントが完全に移動する前にキーを放すと, 移動が割込まれて軸が停止する。同じキーを再び押すと, 0 に到達するまで, 軸は残りの移動距離を移動して終了する。</p> <p>0: JOG-INC の一時検出モードおよび基準点への接近 JOG-INC: (最初の立上がりエッジの) トラバースキーを押すと, 軸は, 全設定インクリメントを移動する。軸がインクリメントを完全に移動する前に (第 2 の立上がりエッジの) 同じキーを再び押すと, 移動が中断する (すなわち, 終了しない)。</p> <p>JOG-INC における連続検出および一時検出モードの間の軸移動特性の差の詳細は, @セクション 2.3 @で説明する。 基準点への接近の移動挙動の詳細は, 次を参照。 参照: 1.16 「原点復帰 (R1)」</p>	
この MD は, 次の場合には無意味	連続 JOG 送り (連続 JOG)	

11310 MD 番号	\$MN_HANDWH_REVERSE ハンドル方向反転スレッシュホールド	
デフォルト値: 2	最小入力リミット: 0	最大入力リミット: -
変更が有効になるための条件: 電源オン	保護レベル: 2/7	単位: -
データタイプ: BYTE	適用開始 SW バージョン: 3.2	
意味:	0: 逆方向で、直ちに移動停止。 >0: 逆方向にハンドルが回されると、少なくとも指令パルス数の分だけ、直ちに逆方向に移動する。	

11320 MD 番号	HANDWH_IMP_PER_LATCH [n] 軸停止位置を検出するたびのハンドルパルス [ハンドル番号]: 0 ... 2	
デフォルト値: 1	最小入力リミット: ***	最大入力リミット: ***
変更が有効になるための条件: 電源オン	保護レベル: 2	単位: -
データタイプ: DOUBLE	適用開始 SW バージョン: 1.1	
意味:	接続されたハンドルは、MD: HANDW_IMP_PER_LATCH で、制御に適合する。 ハンドル移動止め位置のたびにハンドルによって発生するパルス数が入力される。ハンドルパルス補正は、接続されたハンドル (1 ~ 3) ごとに、個別に定義されなければならない。 制御に適合させた時、相対 JOG 送りモードでトラバースキーを押すたびに、ハンドル軸停止位置が同じ効果を出す。 マイナスの値が入力された場合、逆方向でハンドルが有効である。	
関連性	MD: JOG_INCR_WEIGHT (INC/ 手動のマシン軸インクリメント補正)	

11330 MD 番号	JOG_INCR_SIZE_TAB [n] INC/ ハンドルのインクリメントサイズ [インクリメントインデックス]: 0 ... 4	
デフォルト値: 1; 10; 100; 1000; 10000	最小入力リミット: 0	最大入力リミット: +
変更が有効になるための条件: 電源オン	保護レベル: 2	単位: 直線軸: mm 回転軸: 度
データタイプ: DOUBLE	適用開始 SW バージョン: 1.1	
意味:	ハンドル JOG 送りの相対 JOG 送りでは、オペレータが、(たとえば、オペレータパネルを介して) 軸で移動するインクリメント数を定義することができる。可変インクリメントサイズ (INCvar) の他に、5 個の固定インクリメントサイズ (INC...) もセット可能である。 JOG_INCR_SIZE_TAB [n] に値を入力すると、これらの 5 個件の固定インクリメントの各インクリメントサイズがすべての軸に対し定義される。初期設定は、INC1, INC10, INC100, INC1000 および INC10000 である。 入力されたインクリメントサイズは、DRF に対し有効である。 可変インクリメントのサイズは、SD: JOG_VAR_INCR_SIZE で定義される。	
関連性	MD: JOG_INCR_WEIGHT (INC/ 手動のインクリメント補正) IS "Active machine function: INC1; ...; INC10000" (有効なマシンファンクション) (DB21 ~ 28, DBB41 ff) IS "Active machine function: INC1; ...; INC10000" (有効なマシンファンクション) (DB31 ~ 48, DBB69)	

## ■ チャンネル別マシンデータ

20620 MD 番号	\$MC_HANDWH_GEOAX_MAX_INCR_SIZE ジオメトリ軸のハンドルインクリメント制限	
デフォルト値: 0	最小入力リミット: 0	最大入力リミット: -
変更が有効になるための条件: 電源オン	保護レベル: 7/2	単位: -
データタイプ: DOUBLE	適用開始 SW バージョン: 3.2	
意味:	>0: 選択インクリメントサイズの制限 それぞれ, \$MN_JOG_INCR_SIZE [<インクリメント / VDI 信号>] ジオメトリ軸: \$SN_JOG_VAR_INCR_SIZE 0: ジオメトリ軸の場合, 制限なし	

20622 MD 番号	\$HANDWH_GEOAX_MAX_INCR_VSIZE パス速度オーバライド	
デフォルト値: 500	最小入力リミット: 0	最大入力リミット: +
変更が有効になるための条件: 電源オン	保護レベル: 2 / 7	単位: mm/min
データタイプ: DOUBLE	適用開始 SW バージョン: 3.2	
意味:	パス速度オーバライドに関し: > 0: 選択インクリメントサイズの制限 (\$MN_JOG_INCR_SIZE [<インクリメント / VDI 信号>] または \$SN_JOG_VAR_INCR_SIZE) / 1000 * IPO サンプルング時間 =0: 制限なし	

20624 MD 番号	\$MC_HANDWH_CHAN_STOP_COND ハンドルを使用した JOG 送りの運転特性の定義	
デフォルト値 : 0x3FF, 0x3FF, 0x3FF, ...	最小入力リミット : 0	最大入力リミット : 0xFFF
変更が有効になるための条件 : 電源オン	保護レベル : 2/7	単位 : -
データタイプ : DWORD	適用開始 SW バージョン : 3.2	
意味 :	<p>チャンネル別 VDI インタフェース信号で、ハンドルを使用した JOG 送り挙動の定義 :</p> <p>ビット ==0 ハンドルを介してプリセットされた距離の割込みまたは収集          ビット ==1: 移動動作中断または収集なし</p> <p>ビット割当て</p> <p>ビット 0: モードグループ停止          ビット 1: モードグループ停止 軸 + スピンドル          ビット 2: NC 停止          ビット 3: NC 停止 軸 + スピンドル          ビット 4: フィードレートディスエーブル          ビット 5: フィードレートオーバライド          ビット 6: 早送りオーバライド          ビット 7: フィード停止 ジオメトリ軸              ビット 7=0: 割込み / 収集              ビット 7=1: 移動動作中断 / 収集なし</p> <p>ジオメトリ軸設定</p> <p>ビット 8 = 0 ハンドルを使用した JOG の場合、最大許容速度は、該当するマシン軸 / 軸に合わせて、MD 32020: JOG_AX_VELO でセットされたフィードレートに対応する。              = 1 ハンドルを使用した JOG の場合、最大許容速度は、該当するマシン軸 / 軸に合わせて、MD 32000: MAX_AX_VELO でセットされたフィードレートに対応する。</p> <p>ビット 9 = 0 オーバライドは、ハンドルを使用した JOG モードで有効である。              = 1 オーバライドスイッチのセット方法とは無関係に、ハンドルを使用した JOG モードの場合、オーバライドは、必ず、100% であると見なされる。              例外 : オーバライド 0% は、必ず有効である。</p> <p>チャンネルの全軸の DRF 設定</p> <p>ビット 10 = 0 DRF では、MD 11310: HANDWH_REVERSE が有効でない。すなわち、MD 11310 = 0 と同じ挙動である。              = 1 DRF では、MD 11310: HANDWH_REVERSE が有効である。輪郭ハンドル設定。</p> <p>ビット 11 = 0 輪郭ハンドルを停止すると、プログラム実行が自動的に継続する。</p> <p>ビット 11 = 1 輪郭ハンドルを停止すると、NC 停止が自動的に起動する。プログラム実行を継続できるのは、NC-START の後だけである。</p>	



## ■ 軸／スピンドル別マシンデータ

31090 MD 番号	JOG_INCR_WEIGHT INC / ハンドルのインクリメント算出	
デフォルト値 : 0.001	最小入力リミット : ***	最大入力リミット : ***
変更が有効になるための条件 : 電源オン	保護レベル : 2	単位 : 直線軸 : mm 回転軸 : 度
データタイプ : DOUBLE	適用開始 SW バージョン : 1.1	
意味 :	<p>相対モードの JOG キーで軸を移動したり、ハンドルがこの MD で定義したりする場合に適用するインクリメントのパス。</p> <p>方向キーが押されるか、ハンドル位置に到達するたびに各インクリメントで軸がカバーするパスは、次のパラメータによって定義される。</p> <p>± MD: JOG_INCR_WEIGHT (INC / ハンドルのマシン軸インクリメント補正)</p> <p>± 選択したインクリメントサイズ (INC1, ..., INCvar)</p> <p>MD: JOG_INCR_SIZE_TAB [n] および SD: JOG_VAR_INCR_SIZE で、全軸に対し可能なインクリメントステージがグローバルに定義される。</p> <p>負の値を入力すると、トラバースキー方向およびハンドル回転方向が反転する。</p> <p>ソフトウェアバージョン 5 以降 :</p> <p>初期設定 :</p> <p>JOG_INCR_WEIGHT[0]=0.001 mm (メートル法で有効)</p> <p>JOG_INCR_WEIGHT[1]=0.00254 mm (ヤード・ポンド法で有効。0.0001 インチに相当する)</p>	
この MD は次の場合は無意味	AUTOMATIC および MDI の運転モード	
関連性	MD: JOG_INCR_SIZE_TAB SD: JOG_VAR_INCR_SIZE	

32010 MD 番号	JOG_VELO_RAPID JOG モードの早送り	
デフォルト値 : 10000	最小入力リミット : 0	最大入力リミット : +
変更が有効になるための条件 : 電源オン	保護レベル : 2	単位 : 直線軸 : mm/min 回転軸 : mm/rev
データタイプ : DOUBLE	適用開始 SW バージョン : 1.1	
意味 :	<p>JOG モードで早送りオーバーライドキーを操作したり、軸フィードレートオーバーライドスイッチを 100% にセットする場合には、ここに入力される軸速度が適用する。</p> <p>入力値は、最大許容軸速度 (マシンデータ MAX_AX_VELO) を超えてはならない。</p> <p>このマシンデータは、プログラムされた早送り G00 に使用される。</p>	
この MD は次の場合は無意味	AUTOMATIC および MDI の運転モード	
関連性	MD: MAX_AX_VELO (最高軸速度) MD: JOG_REV_VELO_RAPID (早送りオーバーライドによる JOG 回転フィードレート) IS "Rapid traverse override" (早送りオーバーライド) (DB21 ~ 28, DBX12.5 ff) IS "Feedrate override" (フィードレートオーバーライド) (DB21 ~ 28, DBB4)	

32020 MD 番号	JOG_VELO JOG モードの軸速度	
デフォルト値: 2000	最小入力リミット: 0	最大入力リミット: +
変更が有効になるための条件: 電源オン	保護レベル: 2	単位: 直線軸: mm/min 回転軸: mm/rev
データタイプ: DOUBLE	適用開始 SW バージョン: 1.1	
意味:	<p>軸フィードレートオーバーライドスイッチが位置 100% の時, 入力値は, JOG モードで移動する速度である。</p> <p>一般設定データ JOG_SET_VELO = 0 が直線軸および直線フィードレート (MD: JOG_REV_IS_ACTIVE = 0) または SD: JOG_ROT_AX_SET_VELO = 0 が回転軸に合わせてセットされている場合にのみ, この速度が使用される。これに該当する場合, 次の JOG 送りに対する軸速度が有効である。</p> <ul style="list-style-type: none"> <li>- 連続 JOG 送り</li> <li>- 相対 JOG 送り (INC1, ... INCvar)</li> <li>- ハンドル JOG 送り</li> </ul> <p>入力値は, 最大許容軸速度 (マシンデータ MAX_AX_VELO) を超えてはならない。</p> <p>DRF が有効の時, MD: HANDWH_VELO_OVERLAY_FACTOR で, JOG 軸速度を低下させる必要がある。</p> <p>JOG モードのス핀ドル</p> <p>(SD: JOG_SPIND_SET_VELO = 0 の時) このマシンデータを使用すると, 特定のスピンドルの JOG モード速度が定義できる。ただし, 速度は, スピンドル速度オーバーライドスイッチで修正できる。</p>	
アプリケーション	JOG モードで移動する軸/スピンドルごとに, 別々の速度をセットする必要がある場合には, この MD で, 特定の軸に合わせてセットできる。SD: JOG_SET_VELO は, 0 にセットしなければならない。	
関連性	MD: MAX_AX_VELO (最高軸速度) MD: JOG_REV_VELO (JOG の回転フィードレート) MD: HANDWH_VELO_OVERLAY_FACTOR (ハンドル速度 (DRF) に対する JOG 速度の比率) SD: JOG_SET_VELO (G94 の JOG 速度) SD: JOG_ROT_AX_SET_VELO (回転軸の JOG 速度) IS "Feedrate override" (フィードレートオーバーライド) (DB21 ~ 28, DBB4)	

32040 MD 番号	JOG_REV_VELO_RAPID 早送りオーバーライドによる JOG モードの回転フィードレート	
デフォルト値: 2,5	最小入力リミット: 0	最大入力リミット: +
変更が有効になるための条件: 電源オン	保護レベル: 2	単位: mm/rev
データタイプ: DOUBLE	適用開始 SW バージョン: 1.1	
意味:	<p>この MD への入力値は, マスタスピンドル回転を基準として早送りオーバーライドで, JOG モードで軸回転フィードレートを定義する。</p> <p>SD: JOG_REV_IS_ACTIVE = 1 (JOG で有効な回転フィードレート) の時, このフィードレートは有効である。</p>	
この MD は次の場合は無意味	SD: JOG_REV_IS_ACTIVE = "0"	
関連性	SD: JOG_REV_IS_ACTIVE (有効 JOG 回転フィードレート) MD: JOG_REV_VELO (JOG 回転フィードレート)	

32050 MD 番号	JOG_REV_VELO JOG モードの回転フィードレート	
デフォルト値: 0,5	最小入力リミット: 0	最大入力リミット: +
変更が有効になるための条件: 電源オン	保護レベル: 2	単位: mm/rev
データタイプ: DOUBLE	適用開始 SW バージョン: 1.1	
意味:	この MD への入力値は, JOG モードで, マスタスピンドル回転を基準として, 軸の回転フィードレートを定義する。 SD: JOG モードで, 回転フィードレートが有効 (JOG_REV_IS_ACTIVE = 1) の時, このフィードレートが有効である。	
関連性	SD: JOG_REV_IS_ACTIVE (有効 JOG 回転フィードレート) MD: JOG_REV_VELO_RAPID (早送りによる JOG 回転フィードレート)	

32080 MD 番号	\$MA_HANDWH_MAX_INCR_SIZE 選択インクリメントの制限	
デフォルト値: 0	最小入力リミット: 0	最大入力リミット: -
変更が有効になるための条件: リセット	保護レベル: 2/7	単位:
データタイプ: DOUBLE	適用開始 SW バージョン: 3.2	
意味:	>0: 選択インクリメントサイズの制限 \$MN_JOG_INCR_SIZE [<インクリメント / VDI 信号>] または \$SN_JOG_VAR_INCR_SIZE (関連マシン軸) 0: 制限なし	

32082 MD 番号	\$MA_HANDWH_MAX_INCR_VELO_SIZE 速度オーバーライドの選択インクリメント制限	
デフォルト値: 500	最小入力リミット: 0	最大入力リミット: +
変更が有効になるための条件: リセット	保護レベル: 2/7	単位: mm/min
データタイプ: DOUBLE	適用開始 SW バージョン: 3.2	
意味:	位置決め軸速度オーバーライド: >0: 選択インクリメントサイズの制限 \$MN_JOG_INCR_SIZE [<インクリメント / VDI 信号>] または \$SN_JOG_VAR_INCR_SIZE (関連マシン軸) 0: 制限なし	

32084 MD 番号	\$MA_HANDWH_STOP_COND ハンドルに対する VDI 信号制御	
デフォルト値: 0xFF	最小入力リミット: 0	最大入力リミット: 0xFF
変更が有効になるための条件: リセット	保護レベル: 2/7	単位: -
データタイプ: DWORD	適用開始 SW バージョン: 3.2	
意味:	<p>軸別 VDI インタフェース信号で、ハンドルを使用した JOG 送り挙動の定義:</p> <p>ビット ==0 ハンドルを介してプリセットされた距離の割込みまたは収集</p> <p>ビット ==1: 移動動作中断または収集なし</p> <p>ビット割当て</p> <p>ビット 0: フィードレートオーバライド</p> <p>ビット 1: スピンドル速度オーバライド</p> <p>ビット 2: フィードレート停止/スピンドル停止</p> <p>ビット 3: クランピング手順実行 (==0 無効)</p> <p>ビット 4: コントローライネーブル</p> <p>ビット 5: パルスイネーブル</p> <p>マシン軸:</p> <p>ビット 6 =0 ハンドルを使用した JOG の場合、該当するマシン軸の最大許容速度は、MD 32020: JOG_VELO でセットされたフィードレートに対応する。</p> <p>=1 ハンドルを使用した JOG の場合、該当するマシン軸の最大許容速度は、MD 32000: MAX_AX_VELO でセットされたフィードレートに対応する。</p> <p>ビット 7 =0 オーバライドは、ハンドルを使用した JOG モードで有効である。</p> <p>=1 オーバライドは、オーバライドスイッチのセット方法とは無関係に、ハンドルによる JOG モードに対し 100% であると見なされる。</p> <p>例外: オーバライド 0% は、必ず有効である。</p>	

32090 MD 番号	HANDWH_VELO_OVERLAY_FACTOR ハンドル速度 (DRF) に対する JOG 速度の比率	
デフォルト値 : 0,5	最小入力リミット : 0	最大入力リミット : +
変更が有効になるための条件 : NEW_CONF	保護レベル : 2	単位 : -
データタイプ : DOUBLE	適用開始 SW バージョン : 1.1	
意味 :	<p>DRF で、ハンドルで有効な速度は、このマシンデータで JOG 速度と比較して低下させることができる。</p> <p>DRF で有効な速度に対して、次のものが適用される。</p> $V_{DRF} = SD:JOG\_SET\_VELO * MD:HANDWH\_VELO\_OVERLAY\_FACTOR$ <p>または、SD:JOG_SET_VELO = 0 の時 :</p> $V_{DRF} = MD:JOG\_VELO * MD:HANDWH\_VELO\_OVERLAY\_FACTOR$ <p>回転軸 DRF に対して、SD: JOG_SET_VELO の値の代わりに、SD: JOG_ROT_AX_SET_VELO の速度設定が適用される。</p>	
この MD は次の場合は無意味	JOG ハンドル	
関連性	MD: JOG_VELO (JOG 軸速度) SD: JOG_SET_VELO (G94 の JOG 速度) SD: JOG_AX_SET_VELO (回転軸の JOG 速度)	

## ■ 一般設定データ

41010 SD 番号	JOG_VAR_INCR_SIZE INC / ハンドル可変インタフェースのサイズ	
デフォルト値 : 0	最小入力リミット : ***	最大入力リミット : ***
変更が有効になるための条件 : 即時	保護レベル : MMC-MD 9220	単位 : mm または度
データタイプ : DOUBLE	適用開始 SW バージョン : 1.1	
意味 :	<p>この設定データで、可変インクリメント (INCvar) の選択時のインクリメント数が定義される。トラバースキーを押すか、次の移動止め位置までハンドルを回して、可変インクリメントを選択した場合、このインクリメントサイズは、JOG モードで軸によって移動する (マシン軸またはジオメトリ軸の PLC インタフェース信号 "Active machine function: INC variable" (有効なマシンファンクション: INC 変数) が 1 にセットされる)。</p> <p>定義されたインクリメントサイズも、DRF に対し適用する。</p> <p>注 : 相対 JOG 送りおよびハンドル JOG 送りに対しインクリメントサイズが有効である点に注意する。大きなインクリメント値を入力してハンドルを回すと、軸は、(MD: JOG_INCR_WEIGHT の設定に応じて) 大きい距離をカバーする。</p>	
この SD は次の場合は無意味	連続 JOG	
関連性	<p>IS "Active machine function: INC variable" (有効なマシンファンクション: INC 変数) (DB21 — 28, DBX41.5 ff)</p> <p>または IS "Active machine function; INC variable" (有効なマシンファンクション: INC 変数) (DB31 — 48, DBX 69.5)</p> <p>MD: JOG_INCR_WEIGHT (INC / ハンドルのインクリメント補正)</p>	

41015 SD 番号	JOG_CONT_MODE_LEVELTRIGGRD 連続検出モードの連続 JOG	
デフォルト値: 1	最小入力リミット: 0	最大入力リミット: 1
変更が有効になるための条件: 即時	保護レベル: MMC-MD 9220	単位: -
データタイプ: BOOLEAN	適用開始 SW バージョン: 1.1	
意味:	1: 連続 JOG の連続検出モード 連続検出モード (初期設定) では、軸制限に到達していない時に、トラバースキーを押し続けていると、軸が移動する。キーを放すと、軸がゼロ速度まで減速して、移動終了したものと見なされる。 0: 連続 JOG の一時検出モード 一時検出モードでは、トラバースキーの最初の立上がりエッジで移動動作が開始して、キーを放しても移動が継続する。(第 2 の立上がりエッジで) トラバースキーを再び押すと、軸を再停止できる。	
この SD は次の場合は無意味	相対 JOG 送り (JOG INC) 基準点への接近 (JOG REF)	

41100 SD 番号	JOG_REV_IS_ACTIVE 有効 JOG 回転フィードレート	
デフォルト値: 1	最小入力リミット: 0	最大入力リミット: 1
変更が有効になるための条件: 即時	保護レベル: MMC-MD 9220	単位: -
データタイプ: BOOLEAN	適用開始 SW バージョン: 1.1	
意味:	1: 軸 (マシン軸またはジオメトリ軸) は、主スピンドルの回転を基準として、JOG モードの回転フィードレート (G95) で移動する。 回転フィードレートは、次のようにセットできる。 - グローバル SD: JOG_REV_SET_VELO で (SD = 0 でない時にのみ有効になる) - 軸 MD: JOG_REV_VELO で - 軸 MD: JOG_REV_VELO_RAPID による早送りオーバーライドで 0: 軸は、直線フィードレート (G94) で、JOG モードで移動する。 回転フィードレートは、次のようにセットできる - グローバル SD: JOG_SET_VELO で (SD = 0 でない時にのみ有効になる) - 軸 MD: JOG_VELO で - 軸 MD: JOG_VELO_RAPID による早送りオーバーライドで	
この SD は次の場合は無意味	AUTOMATIC および MDI の運転モード	
関連性	SD: JOG_REV_SET_VELO (G95 の JOG 速度) MD: JOG_REV_VELO (JOG 回転フィードレート) MD: JOG_REV_VELO_RAPID (早送りによる JOG 回転フィードレート) SD: JOG_SET_VELO (G94 の JOG 速度) MD: JOG_VELO (JOG 軸速度) MD: JOG_VELO_RAPID (JOG 早送り)	

41110 SD 番号	JOG_SET_VELO 直線軸の JOG 速度 (G94)	
デフォルト値: 0	最小入力リミット: 0	最小入力リミット: +
変更が有効になるための条件: 即時	保護レベル: MMC-MD 9220	単位: mm/min または mm/rev
データタイプ: DOUBLE	適用開始 SW バージョン: 1.1	
意味:	<p>値 = 0 でない時:</p> <p>速度の入力値は、直線フィードレート (G94) が、関連軸に対し有効の時 (MD: JOG_REV_IS_ACTIVE = 0), JOG モードで移動する直線軸に対し適用する。</p> <p>軸速度は、次の JOG 送りに対し有効である。</p> <ul style="list-style-type: none"> <li>- 連続 JOG 送り</li> <li>- 相対 JOG 送り (INC1, ... INCvar)</li> <li>- ハンドル JOG 送り</li> </ul> <p>入力値は、どの直線軸に対しても有効であるが、最大許容軸速度 (MD: MAX_AX_VELO) を超えてはならない。</p> <p>DRF で: DRF が有効の時, SD: JOG_SET_VELO でセットされた速度を, MD: HANDWH_VELO_OVERLAY_FACTOR で減速する必要がある。</p> <p>値 = 0 の時:</p> <p>0 が設定データに入力された場合, JOG モードの有効直線フィードレートは, MD: JOG_VELO "JOG axis velocity" (JOG 軸速度) である。各軸には, この MD (軸 MD) で, それぞれ, 固有の JOG 速度が割当てられている。</p>	
この SD は次の場合は無意味	<p>- SD: JOG_REV_IS_ACTIVE = 1 の場合, 直線軸に対して</p> <p>- 回転軸に対して (SD: JOG_ROT_AX_SET_VELO がここで有効になる)</p>	
関連性	<p>SD: JOG_REV_IS_ACTIVE (有効 JOG 回転フィードレート)</p> <p>軸 MD: JOG_VELO (JOG 軸速度)</p> <p>軸 MD: MAX_AX_VELO (最高軸速度)</p> <p>軸 MD: HANDWH_VELO_OVERLAY_FACTOR (ハンドル速度 (DRF) に対する JOG 速度の比率)</p> <p>SD: JOG_ROT_AX_SET_VELO (回転軸の JOG 速度)</p>	



41120 MD 番号	JOG_REV_SET_VELO JOG 速度 (G95)	
デフォルト値: 0	最小入力リミット: 0	最大入力リミット: +
変更は直ちに有効。	保護レベル: MMC-MD 9220	単位: mm/rev
データタイプ: DOUBLE	適用開始 SW バージョン: 1.1	
意味:	<p>値 = 0 の時:</p> <p>速度の入力値は、回転フィードレート (G95) が、関連軸に対し有効の時 (MD: JOG_REV_IS_ACTIVE = 1), JOG モードで移動する直線軸に対し適用する。</p> <p>軸速度は、次の JOG 送りに対し有効である。</p> <ul style="list-style-type: none"> <li>- 連続 JOG 送り</li> <li>- 相対 JOG 送り (INC1, ... INCvar)</li> <li>- ハンドル JOG 送り</li> </ul> <p>入力値は、どの直線軸に対しても有効であるが、最大許容軸速度 (MD: MAX_AX_VELO) を超えてはならない。</p> <p>値 = 0 の時:</p> <p>0 が設定データに入力された場合、JOG モードの有効回転フィードレートは、MD: JOG_REV_VELO "revolutional feedrate with JOG" (JOG 回転フィードレート) である。</p> <p>各軸には、この MD (軸 MD) で、それぞれ、固有の回転フィードレートが割当てられている。</p>	
この SD は次の場合は無意味	- SD: JOG_REV_IS_ACTIVE = 0 の場合、軸に対して。	
アプリケーション	オペレータは、特定のアプリケーションに合わせて、JOG 速度を定義することができる。	
関連性	<p>軸 SD: JOG_REV_IS_ACTIVE (有効 JOG 回転フィードレート)</p> <p>軸 MD: JOG_VELO (JOG による回転フィードレート)</p> <p>軸 MD: MAX_AX_VELO (最高軸速度)</p>	

41130 MD 番号	JJOG_ROT_AX_SET_VELO 回転軸の JOG 速度	
デフォルト値 : 0	最小入力リミット : 0	最大入力リミット : +
変更は直ちに有効。	保護レベル : MMC-MD 9220	単位 : mm/rev
データタイプ : DOUBLE	適用開始 SW バージョン : 1.1	
意味 :	<p>値 = 0 でない時 :</p> <p>入力速度は , JOG モード (連続モード , 相対モード , ハンドルを使用した JOG 送り) の回転軸に対し適用する。その入力値は , すべての回転軸に共通した値であり , 最大許容軸速度を超えてはならない (MD: MAX_AX_VELO)。</p> <p>DRF で , SD: JOG_ROT_AX_SET_VELO でセットされた速度は , MD: HANDWH_VELO_OVERLAY_FACTOR で低下させなければならない。</p> <p>値 = 0 の時 :</p> <p>0 が設定データに入力される場合 , JOG モードで回転軸に適用する速度は , 軸 MD: JOG_VELO (JOG 送り軸速度) である。このように , 各軸に JOG 速度を個別に定義することができる。</p>	
アプリケーション	オペレータは , 特定のアプリケーションに合わせて JOG 速度を定義することができる。	
関連性	MD: JOG_VELO (JOG 軸速度) MD: MAX_AX_VELO (最高軸速度) MD: HANDWH_VELO_OVERLAY_FACTOR (ハンドル速度 (DRF) に対する JOG 速度の比率)	

41200 MD 番号	JOG_SPIND_SET_VELO スピンドルの JOG 速度	
デフォルト値 : 0	最小入力リミット : 0	最大入力リミット : +
変更は直ちに有効。	保護レベル : MMC-MD 9220	単位 : mm/rev
データタイプ : DOUBLE	適用開始 SW バージョン : 1.1	
意味 :	<p>値 = 0 でない時 :</p> <p>「プラスおよびマイナスのトラバースキー」で手動移動する場合 , 入力速度は , JOG モードのスピンドルに対し適用する。</p> <p>速度は , 次の JOG 送りに対し有効である。</p> <ul style="list-style-type: none"> <li>- 連続 JOG 送り</li> <li>- 相対 JOG 送り (INC1, ... INCvar)</li> <li>- ハンドル JOG 送り</li> </ul> <p>入力値は , すべての回転軸に有効な値であり , 最大許容軸速度を超えてはならない (MD: MAX_AX_VELO)。</p> <p>値 = 0 の時 :</p> <p>0 が設定データに入力される場合 , 有効な JOG 速度は , 軸 MD: JOG_VELO (通常の軸速度) であり , 回転軸の JOG 速度として作用する。各軸には , この MD で , 固有の JOG 速度を個別に与えることができる。</p> <p>スピンドルが JOG モードで移動する場合には , 有効なギヤステージの最高速度 (MD: GEAR_STEP_VELO_LIMIT) が考慮される。</p>	
この SD は次の場合は無意味	軸	
アプリケーション	オペレータは , 特定のアプリケーションに合わせて JOG 速度を定義することができる。	
関連性	軸 MD: JOG_VELO (JOG 軸速度) MD: GEAR_STEP_MAX_VELO_LIMIT (ギヤステージの最高速度)	
参照	1.17 「主軸機能 (S1)」	

## 2.4.5 信号の説明

### ■ 一般信号

### ■ NC からの信号

DB10 DBB97, 98, 99 データブロック	ハンドル 1, 2, 3 のジオメトリ軸のチャンネル番号 NC からの信号 (MMC → PLC)																																					
エッジ評価 : なし	信号更新 : 周期的	信号が有効となる最初の SW バージョン : 2.1																																				
信号の意味	<p>オペレータは、オペレータパネルのハンドル (1, 2, 3) に軸を直接割当てることができる。ジオメトリ軸 (IS: "Machine axis" (マシン軸) = 0) の場合には、対象のハンドルに割当てられたチャンネル番号が PLC に送られる。</p> <p>この方法で、オペレータがセットしたステータス (IS "Handwheel selected" (選択ハンドル)) に従い、選択したジオメトリ軸に合わせて、IS "Activate handwheel" (ハンドル起動) がセットされる。</p> <p>チャンネル番号に対し、次のコードが適用される：</p> <table><thead><tr><th>ビット 7</th><th>6</th><th>5</th><th>4</th><th>3</th><th>2</th><th>1</th><th>ビット 0</th><th>チャンネル番号</th></tr></thead><tbody><tr><td>0</td><td>0</td><td>0</td><td>0</td><td>0</td><td>0</td><td>0</td><td>0</td><td>-</td></tr><tr><td>0</td><td>0</td><td>0</td><td>0</td><td>0</td><td>0</td><td>0</td><td>1</td><td>1</td></tr><tr><td>0</td><td>0</td><td>0</td><td>0</td><td>0</td><td>0</td><td>1</td><td>0</td><td>2</td></tr></tbody></table> <p>マシン軸 (IS "Machine axis" = 1) で、IS "Channel number geometry axis for handwheel 1, 2, 3" (ハンドル 1, 2, 3 のチャンネル番号ジオメトリ軸) は無効である。</p> <p>詳細は、IS "Axis number for handwheel 1, 2, 3" (ハンドル 1, 2, 3 の軸番号) を参照。</p>		ビット 7	6	5	4	3	2	1	ビット 0	チャンネル番号	0	0	0	0	0	0	0	0	-	0	0	0	0	0	0	0	1	1	0	0	0	0	0	0	1	0	2
ビット 7	6	5	4	3	2	1	ビット 0	チャンネル番号																														
0	0	0	0	0	0	0	0	-																														
0	0	0	0	0	0	0	1	1																														
0	0	0	0	0	0	1	0	2																														
関連性	IS "Axis number of handwheel 1, 2, 3" (ハンドル 1, 2, 3 の軸番号) (DB10, DBB100 ff) IS "Handwheel selected" (ハンドル選択) (DB10, DBX100.6 ff) IS "Machine axis" (マシン軸) (DB10, DBX100.7 ff) IS "Activate handwheel" (ハンドル起動) (DB21, ... DBX12.0 - 12.2 ff)																																					
アプリケーション	DB10, DBB97 = 2 の時、ハンドル 1 がチャンネル 2 に割当てられる。																																					

DB10 DBB 100; 101; 102, ビット 0 – 4 データブロック	ハンドル 1, 2 または 3 の軸番号 NC からの信号 (MMC → PLC)																																																																
エッジ評価: なし	信号更新: 周期的		信号が有効となる最初の SW バージョン: 1.1																																																														
信号の意味	<p>オペレータは、オペレータパネルを介して、各ハンドルに軸を直接割り当てることができる。割り当てするには、必要な軸 (たとえば, X) を定義する。</p> <p>基本 PLC プログラムでは、軸番号の他に、「マシン軸またはジオメトリ軸」の情報 (IS "machine axis" マシン軸) も、MMC インタフェース信号として与える。基本 PLC プログラムは、定義された軸に合わせて、インタフェース信号 "Activate handwheel" (ハンドル起動) をセットする。</p> <p>MMC インタフェース信号 "machine axis" (マシン軸) の設定に応じて、ジオメトリ軸またはマシン軸のインタフェースが使用される。</p> <p>軸番号に軸名称を割り当てる場合には、次の点に注意する。</p> <ul style="list-style-type: none"><li>IS "Machine axis" (マシン軸) = 1; すなわち、マシン軸: 割り当ては、MD: AXCONF_MACHAX_NAME_TAB [n] (マシン軸名称) を介して行われる。</li><li>IS "Machine axis" (マシン軸) = 0; すなわち、ジオメトリ軸: MD: AXCONF_GEOAX_NAME_TAB [n] (ジオメトリ軸) 割り当ては、MD: AXCONF_MACHAX_NAME_TAB [n] (チャンネルのジオメトリ軸名称) を介して行われる。IS "Channel number geometry axis handwheel n" (チャンネル番号ジオメトリ軸のハンドル n) は、ハンドルに割り当てられたチャンネルを定義する。</li></ul> <p>軸番号に対し、次のコードが適用される。</p> <table><tr><th>ビット 4</th><th>ビット 3</th><th>ビット 2</th><th>ビット 1</th><th>ビット 0</th><th>軸番号</th></tr><tr><td>0</td><td>0</td><td>0</td><td>0</td><td>0</td><td>--</td></tr><tr><td>0</td><td>0</td><td>0</td><td>0</td><td>1</td><td>1</td></tr><tr><td>0</td><td>0</td><td>0</td><td>1</td><td>0</td><td>2</td></tr><tr><td>0</td><td>0</td><td>0</td><td>1</td><td>1</td><td>3</td></tr><tr><td>0</td><td>0</td><td>1</td><td>0</td><td>1</td><td>4</td></tr><tr><td>0</td><td>0</td><td>1</td><td>0</td><td>1</td><td>5</td></tr><tr><td>0</td><td>0</td><td>1</td><td>1</td><td>0</td><td>6</td></tr><tr><td>0</td><td>0</td><td>1</td><td>1</td><td>1</td><td>7</td></tr><tr><td>0</td><td>1</td><td>0</td><td>0</td><td>0</td><td>8</td></tr></table>					ビット 4	ビット 3	ビット 2	ビット 1	ビット 0	軸番号	0	0	0	0	0	--	0	0	0	0	1	1	0	0	0	1	0	2	0	0	0	1	1	3	0	0	1	0	1	4	0	0	1	0	1	5	0	0	1	1	0	6	0	0	1	1	1	7	0	1	0	0	0	8
ビット 4	ビット 3	ビット 2	ビット 1	ビット 0	軸番号																																																												
0	0	0	0	0	--																																																												
0	0	0	0	1	1																																																												
0	0	0	1	0	2																																																												
0	0	0	1	1	3																																																												
0	0	1	0	1	4																																																												
0	0	1	0	1	5																																																												
0	0	1	1	0	6																																																												
0	0	1	1	1	7																																																												
0	1	0	0	0	8																																																												
関連性	IS "Channel number geometry axis handwheel n" (DB10, DBX97 ff) IS "Handwheel selected" (DB10, DBX100.6 ff) IS "Machine axis" (DB10, DBX100.7 ff) IS "Activate handwheel" (DB21, DBX12.0 – DBX12.2 ff) IS "Activate handwheel" (DB31, DBX4.0 – DBX4.2) MD: AXCONF_MACHAX_NAME_TAB [n] (マシン軸名称) MD: AXCONF_GEOAX_NAME_TAB [n] (チャンネルのジオメトリ軸名称)																																																																

DB10 DBX100.6; 101.6; 102.6 データブロック	(ハンドル 1, 2 または 3 の) ハンドル選択 NC からの信号 (MMC → PLC)	
エッジ評価: なし	信号更新: 周期的	信号が有効となる最初の SW バージョン: 1.1
信号ステータス 1 あるいは信号 遷移 0 → 1	オペレータは、オペレータパネルを介して、定義軸に合ったハンドルを選択する（すなわち、起動する）。この情報は、MMC インタフェースで、基本 PLC プログラムによって有効な状態にされる。 したがって、定義軸のインタフェース信号 "Activate handwheel"（ハンドル起動）は、基本 PLC プログラムによって、'1' にセットされる。 対象の軸の信号も MMC インタフェースで送られる (IS: "Axis number"（軸番号）および IS "Machineaxis"（マシン軸））。 ハンドルが有効になると、ハンドルを使用した JOG モードで、軸は、移動できる (IS "Handwheel active"（有効なハンドル）)。	
信号ステータス 0 あるいは信号 遷移 1 → 0	オペレータは、オペレータパネルを介して、定義軸に合ったハンドルをディセーブルした。この情報は、MMC インタフェースで、基本 PLC プログラムによって有効な状態にされる。 その時、定義軸のインタフェース信号 "Activate handwheel"（ハンドル起動）を基本 PLC プログラムによってリセットすることができる。	
関連性	IS "Axis number" (DB10, DBB100 ff) IS "Machine axis" (DB10, DBX100.7 ff) IS "Activate handwheel" (DB21, DBX12.0 ~ DBX12.2 ff) IS "Handwheel active" (DB21, DBX40.0 ~ DBX40.2 ff) IS "Activate handwheel" (DB31, DBX4.0 ~ DBX4.2) IS "Channel number geometry axis for handwheel 1, 2 or 3" (DB10, DBB97 ff)	

DB10 DBX100.7; 101.	(ハンドル 1, 2 または 3 の) マシン軸 NC からの信号 (MMC → PLC)	
エッジ評価: なし	信号更新: 周期的	信号が有効となる最初の SW バージョン: 1.1
信号ステータス 1 あるいは信号 遷移 0 → 1	オペレータは、オペレータパネルのハンドル (1, 2, 3) に軸を直接割当てた。この軸は、マシン軸である。 詳細は、IS "Axis number"（軸番号）を参照。	
信号ステータス 0 あるいは信号 遷移 1 → 0	オペレータは、オペレータパネルのハンドル (1, 2, 3) に軸を直接割当てた。この軸は、ジオメトリ軸である。 詳細は、IS "Axis number"（軸番号）を参照。	
関連性	IS "Axis number" (DB10, DBB100 ff) IS "Handwheel selected" (DB10, DBX100.6 ff) NST "Channel number geometry axis for handwheel 1, 2 or 3" (DB10, DBB97 ff)	

## ■ チャンネル別信号

### ■ チャンネルへの信号の概要

DB 21, ...	チャンネルへの信号							
DBB	ビット 7	ビット 6	ビット 5	ビット 4	ビット 3	ビット 2	ビット 1	ビット 0
0		ドライラン フィード レートの 起動	M01 起動	単一 ブロック 起動	DRF 起動	トラバース フォワード 起動	トラバース バック ワード 起動	外部からの 実行
12	ジオメトリ軸 1							
	トラバースキー +	-	早送りオー バライド	トラバース キーディス エーブル	フィード ホールド	3	2	1
13	ジオメトリ軸 1 マシンファンクション							
		連続	可変 INC	10000 INC	1000 INC	100 INC	10 INC	1 INC
16	ジオメトリ軸 2							
	トラバースキー +	-	早送りオー バライド	トラバース キーディス エーブル	フィード ホールド	3	2	1
17	ジオメトリ軸 2 マシンファンクション							
		連続	可変 INC	10000 INC	1000 INC	100 INC	10 INC	1 INC
20	ジオメトリ軸 3							
	トラバースキー +	-	早送りオー バライド	トラバース キーディス エーブル	フィード ホールド	3	2	1
21	ジオメトリ軸 3 マシンファンクション							
		連続	可変 INC	10000 INC	1000 INC	100 INC	10 INC	1 INC

## ■ チャンネルへの信号の説明

DB21, ... DBX0.3 データブロック	DRF 起動 チャンネルへの信号 (PLC → NCK)	
エッジ評価: なし	信号更新: 周期的	信号が有効となる最初の SW バージョン: 1.1
信号ステータス 1 あるいは信号遷移 0 → 1	ファンクション DRF が選択される。 ファンクションは、直接、PLC ユーザプログラムまたはオペレータパネルから、MMC インタフェース信号 "DRF selected" (DRF 選択) を介して選択される。この MMC インタフェース信号は、基本 PLC プログラムまたは PLC ユーザプログラムによって、インタフェース信号 "Activate DRF" (DRF 起動) に変換される。 ファンクション DRF が有効になると、DRF オフセットは、AUTOMATIC または MDI の運転モードで修正することができる。	
信号ステータス 0 あるいは信号遷移 1 → 0	ファンクションが選択されない。	
この信号は次の場合は無意味	JOG モード	
アプリケーション	DRF ファンクションは、IS "Activate DRF" (DRF 起動) で、特に PLC ユーザプログラムによってイネーブルできる。	
関連性	IS "DRF selected" (DB21, ... DBX24.3)	

DB21, ... DBB12; 16; 20 ビット 0 – 2 データブロック	ジオメトリ軸 (1, 2, 3) のハンドル (1 ~ 3) 起動 チャンネルへの信号 (PLC → NCK)	
エッジ評価: なし	信号更新: 周期的	信号が有効となる最初の SW バージョン: 1.1
信号ステータス 1 あるいは信号遷移 0 → 1	これらのマシンデータは、1, 2 または 3 にこのジオメトリ軸が割当てられているかどうか、または、どのハンドルにも割当てられていないかを判定する。 軸には、一度に 1 件のハンドルしか割当てることができない。 複数のインタフェース信号 "Activate handwheel" (ハンドル起動) がセットされた場合には、"Handwheel 1" (ハンドル 1), "Handwheel 2" (ハンドル 2), "Handwheel 3" (ハンドル 3) の順番に適用される。 注: 3 つのジオメトリ軸は、ハンドル 1 ~ 3 を使用して、同時にトラバースすることができる。	
信号ステータス 0 あるいは信号遷移 1 → 0	このジオメトリ軸には、ハンドル 1, 2 および 3 を割当てることができない。	
アプリケーション	PLC ユーザプログラムは、このインタフェース信号で、ジオメトリ軸のハンドルの回転による影響をディスエーブルできる。	
関連性	ジオメトリ軸 -- IS "Handwheel active" (DB21, DBX40.7 or DBX40.6 ff)	

DB21, ... DBX12.4; 16.4; 20.4 データブロック	ジオメトリ軸のトラバースキーディスエーブル (1,2,3) チャンネルへの信号 (PLC → NCK)	
エッジ評価：なし	信号更新：周期的	信号が有効となる最初の SW バージョン：1.1
信号ステータス 1 あるいは信号 遷移 0 → 1	プラスおよびマイナスのトラバースキーは、対象のジオメトリ軸に対して無効である。したがって、マシン制御パネルのトラバースキーを使用して、JOG でジオメトリ軸をトラバースすることはできない。 トラバース動作中に、トラバースディスエーブルが起動すると、ジオメトリ軸が停止する。	
信号ステータス 0	プラスおよびマイナスのトラバースキーがイネーブルされる。	
アプリケーション	したがって、運転モードに応じて、PLC ユーザプログラムから、トラバースキーで、JOG モードでジオメトリ軸の手動トラバースをディスエーブルすることができる。	
関連性	ジオメトリ軸 -- IS "Traverse key plus" および "Traverse key minus" (DB21, DBX12.7 or DBX12.6 ff)	

DB21, ... DBX12.5; 16.5; 20.5 データブロック	ジオメトリ軸 (1,2,3) の早送りオーバーライド チャンネルへの信号 (PLC → NCK)	
エッジ評価：なし	信号更新：周期的	信号が有効となる最初の SW バージョン：1.1
信号ステータス 1 あるいは信号 遷移 0 → 1	インタフェース信号 "Rapid traverse override"（早送りオーバーライド）が "Traverse key plus"（プラスのトラバースキー）および "Traverse key minus"（マイナスのトラバースキー）とともにセットされる場合には、対象のジオメトリキーは、早送りでトラバースする。 早送りフィードレートは、マシンデータ JOG_VELO_RAPID で定義される。 次の JOG モードで、トラバースフィードレートが有効である。 - 連続 JOG 送り - 相対 JOG 送り 早送りオーバーライドが有効の場合には、速度は、早送りオーバーライドスイッチで修正することができる。	
信号ステータス 0 あるいは信号 遷移 1 → 0	ジオメトリ軸は、定義された JOG 速度 (SD: JOG_SET_VELO or MD: JOG_VELO) でトラバースする。	
この信号は次の場合は無意味	- AUTOMATIC および MDI の運転モード - 基準点への接近 (JOG モード)	
関連性	ジオメトリ軸 --- IS "Traverse key plus" および "Traverse key minus" (DB21, DBX12.7 or DBX12.6 ff)	
参照	1.18 「送り機能 (V1)」	



DB21, ... DBB12; 16; 20 ビット 7; 6 データブロック	ジオメトリ軸 (1,2,3) のトラバースキー (プラスおよびマイナス) チャンネルへの信号 (PLC → NCK)	
エッジ評価: あり	信号更新: 周期的	信号が有効となる最初の SW バージョン: 1.1
信号ステータス 1 あるいは信号遷移 0 → 1	<p>選択したジオメトリ軸は、プラスおよびマイナスのトラバースキーを使って、JOG モードで、両方の方向へトラバースできる。</p> <p>有効なマシンファンクションおよび設定 "Continuous-trigger or momentary-trigger mode" (連続 JOG 送り用の SD: JOG_CONT_MODE_LEVELTRIGGRD および相対 JOG 送り用の MD: JOG_INC_MODE_LEVELTRIGGRD) に応じて、信号遷移は、次のように様々な反応を引き起こす。</p> <p>ケース 1: 連続検出モードの連続 JOG 送り インタフェース信号が 1 にセットされている限り (更に、軸位置が起動制限に到達していない間)、ジオメトリ軸は、関連する方向にトラバースする。</p> <p>ケース 2: 一時検出モードの連続 JOG 送り 信号ステータスが最初に 0 ± 1 から変化した時、ジオメトリ軸は、関連方向へのトラバースを開始する。このトラバース動作は、信号ステータスが 1 0 に変化しても継続する。0 1 の新しい信号遷移 (同じ移動方向で) が、トラバース動作を停止する。</p> <p>ケース 3: 連続検出モードの相対 JOG 送り 信号 1 で、ジオメトリは、設定インクリメントでトラバースを開始する。インクリメントがトラバースされる前に、信号が 0 ステータスに変化すると、トラバース動作が中断される。信号ステータスが再び 1 に変化すると、移動が再開する。 ジオメトリ軸は、インクリメントが完全にトラバースされるまで、上記のように停止と再開を数回繰返すことができる。</p> <p>ケース 4: 一時検出モードの相対 JOG 送り 信号ステータスが最初に 0 1 に変化した時、ジオメトリ軸は、設定インクリメントのトラバースを開始する。ジオメトリ軸がインクリメントを移動する前に、同じトラバース信号の信号ステータスが 0 1 に再び変化した時、トラバース動作が強制終了させられる。インクリメントは、終端までトラバースされない。</p> <p>両方のトラバース信号 (プラスおよびマイナス) が同時にセットされると、移動しないか、現在の移動が強制終了する。</p> <p>PLC インタフェース信号 "Traverse key disable" (トラバースキーディスエーブル) によって、各ジオメトリ軸に対するトラバースキーの影響が個別にディスエーブルされる。</p> <p>注意 マシン軸とは異なり、トラバースキーでは、一度に 1 件の軸しか検出することができない。トラバースキーで複数の軸をトラバースしようとすると、アラーム 20062 が検出される。</p>	
信号ステータス 0 あるいは信号遷移 1 → 0	上記のケース 1 ~ 4 を参照。	
この信号は次の場合は無意味	AUTOMATIC および MDI の運転モード	
例外, エラー	<p>ジオメトリ軸は、次のような時、JOG モードでトラバースできない。</p> <ul style="list-style-type: none"> <li>- (マシン軸として) 軸 PLC インタフェースを介して既にトラバースを開始している時。</li> <li>- トラバースキーを使用して、他のジオメトリ軸のトラバースを既に開始している時。</li> </ul> <p>この時、アラーム 20062 "Axis already active" (既に有効な軸) が出力される。</p>	
関連性	<p>マシン軸 --- IS "Traverse keys plus or minus" (DB31, ... BX8.7 または DBX8.6)</p> <p>IS "Traverse key disable for geometry axes" (DB21, ... DBX12.4 ff)</p>	

DB21, ... DBB13; 17; 21 ビット 0 – 5 データブロック	ジオメトリ軸 (1,2,3) のマシンファンクション INC1, INC10, INC100, INC1000, INC10000, INCvar チャンネルへの信号 (PLC → NCK)	
エッジ評価: なし	信号更新: 周期的	信号が有効となる最初の SW バージョン: 1.1
信号ステータス 1 あるいは信号 遷移 0 → 1	<p>このインタフェース信号は、トラバースキーが押されるか、ハンドルが次の移動止め位置まで回された時にジオメトリ軸がトラバースするインクリメント数を定義する。JOG モードは、これに対し有効でなければならない（例外: DRF で）。</p> <p>インクリメントサイズは、次のように、インタフェース信号に割当てられている。</p> <ul style="list-style-type: none"> <li>- INC1 ~ INC10000: 一般マシンデータ JOG_INCR_SIZE_TAB で</li> <li>- INCvar: 一般設定データ JOG_VAR_INCR_SIZE で</li> </ul> <p>選択したマシンファンクションが有効になると、この信号は、PLC インタフェース (IS "Active machine function INC1;..." 有効なマシンファンクション) に送られる。</p> <p>複数のマシンファンクション信号 (INC1, INC... または "Continuous jogging" (連続 JOG 送り)) がインタフェースで同時に選択された場合、制御装置はマシンファンクションを起動することができない。</p>	
信号ステータス 0 あるいは信号 遷移 1 → 0	<p>対象のマシンファンクションが選択されない。</p> <p>軸がインクリメントをトラバースしている時に、このマシンファンクションが選択解除されるか切換えられると、この移動も強制終了させられる。</p>	
関連性 ....	<p>ジオメトリ軸 -- IS "Active machine function INC1,..." (DB21, ... DBB41 ff)</p> <p>ジオメトリ軸 -- IS "Machine function continuous" (DB21, ... DBX13.6 ff)</p>	

DB21, ... DBX13.6; 17.6; 21.6 データブロック	ジオメトリ軸 (1, 2, 3) の「連続」マシンファンクション チャンネルへの信号 (PLC → NCK)	
エッジ評価: なし	信号更新: 周期的	信号が有効となる最初の SW バージョン: 1.1
信号ステータス 1 あるいは信号 遷移 0 → 1	<p>マシンファンクション "Continuous jogging" (連続 JOG 送り) が選択される。</p> <p>関連ジオメトリ軸は、JOG モードで、プラスとマイナスのトラバースキーでトラバースできる。</p>	
信号ステータス 0 あるいは信号 遷移 1 → 0	マシンファンクション "Continuous jogging" (連続 JOG 送り) が選択されない。	
関連性	<p>IS "Active machine function INC 1,..., continuous" (DB21, ... DBB41 ff)</p> <p>IS "Machine function INC1,...,INC10000" (DB21, ... DBB13 ff)</p>	

■ チャンネルからの信号の概要

DB 21 ~ 28	チャンネルへの信号							
DBB	ビット 7	ビット 6	ビット 5	ビット 4	ビット 3	ビット 2	ビット 1	ビット 0
24 (MMC → PLC)		ドライラ ンフィー ドレート 選択	M01 選択	単一 ブロック 選択	DRF 選択			
33 (MMC → PLC)					ハンドル オーバ ライド 有効			
37	SBL 除去 で、ブ ロック終 端で 停止する。	読取り イネーブ ルが無視 される。	CLC 停止 上限 /TE1/	CLC 停止 下限 /TE1/	CLC 有効 / TE1/	輪郭ハンドル 有効		
						ハンドル 1	ハンドル 2	ハンドル 3
40	ジオメトリ軸 1							
	移動コマンド プラス	マイナス				ハンドルが有効		
						3	2	1
41	ジオメトリ軸 1							
	有効なマシンファンクション							
		連続	可変 INC	10000 INC	1000 INC	100 INC	10 INC	1 I NC
46	ジオメトリ軸 2							
	移動コマンド プラス	マイナス				ハンドルが有効		
						3	2	1
47	ジオメトリ軸 2							
	有効なマシンファンクション							
		連続	可変 INC	10000 INC	1000 INC	100 INC	10 INC	1 I NC
52	ジオメトリ軸 3							
	移動コマンド プラス	マイナス				ハンドルが有効		
						3	2	1
53	ジオメトリ軸 3							
	有効なマシンファンクション							
		連続	可変 INC	10000 INC	1000 INC	100 INC	10 INC	1 I NC

## ■ チャンネルからの信号の説明

DB21, ... DBX24.3 データブロック	DRF selected DRF 選択 チャンネルからの信号 (MMC → PLC)	
エッジ評価：なし	信号更新：周期的	信号が有効となる最初の SW バージョン：1.1
信号ステータス 1 あるいは信号遷移 0 → 1	オペレータは、オペレータパネル上で DRF を選択した。論理組合わせの後、PLC プログラム（基本 PLC プログラムまたはユーザプログラム）は、IS "Activate DRF"（DRF 起動）として、この MMC インタフェース信号を送る。DRF が有効になった時に、AUTOMATIC または MDI モードで、軸に割当てたハンドルを使用すると、DRF オフセットを変更することができる。	
信号ステータス 0 あるいは信号遷移 1 → 0	オペレータは、オペレータパネル上で DRF を選択していない。	
この信号は次の場合は無意味	JOG モード	
関連性	IS: "Activate DRF" (DB21, ... DBX0.3)	

DB21, ... DBB37 ビット 0 – 2 データブロック	有効な輪郭ハンドル (1 ~ 3) 軸／スピンドルからの信号 (NCK → PLC)	
エッジ評価：なし	信号更新：周期的	信号が有効となる最初の SW バージョン：4.3
信号ステータス 1 あるいは信号遷移 0 → 1	これらの PLC インタフェース信号は、輪郭ハンドル 1, 2 または 3 にこのジオメトリ軸が割当てられているかどうか、または、どの輪郭ハンドルにも割当てられていないかを通知する。 軸には、一度に 1 つの輪郭ハンドルしか割当てることができない。 複数のインタフェース信号 "Contour handwheel active"（有効な輪郭ハンドル）がセットされた場合には、'Contour handwheel 1'（輪郭ハンドル 1）、'Contour handwheel 2'（輪郭ハンドル 2）、'Contour handwheel 3'（輪郭ハンドル 3）の順番に適用される。 割当てが有効の時には、輪郭ハンドルを使用した JOG モードでのジオメトリ軸のトラバース、または、AUTOMATIC または MDI モードでの DRF オフセットの発生が可能になる。	
信号ステータス 0 あるいは信号遷移 1 → 0	輪郭ハンドル 1, 2 および 3 は、このジオメトリ軸に割当てられていない。	
関連性		

DB21, ... DBB40; 46; 52 ビット 0 – 2 データブロック	ジオメトリ軸の有効ハンドル (1 ~ 3) 軸／スピンドルからの信号 (NCK → PLC)	
エッジ評価: なし	信号更新: 周期的	信号が有効となる最初の SW バージョン: 1.1
信号ステータス 1 あるいは信号 遷移 0 → 1	これらの PLC インタフェース信号は、ハンドル 1, 2 または 3 にこのジオメトリ軸が割当てられているかどうか、または、どの輪郭ハンドルにも割当てられていないかを通知する。 軸には、一度に 1 つの輪郭ハンドルしか割当てることができない。 複数のインタフェース信号 "Contour handwheel active" (有効な輪郭ハンドル) がセットされた場合には、'Contour handwheel 1' (輪郭ハンドル 1), 'Contour handwheel 2' (輪郭ハンドル 2), 'Contour handwheel 3' (輪郭ハンドル 3) の順番に適用される。 割当てが有効の時には、ハンドルを使用した JOG モードでのジオメトリ軸のトラバース、または、AUTOMATIC または MDI モードでの DRF オフセットの発生が可能になる。	
信号ステータス 0 あるいは信号 遷移 1 → 0	輪郭ハンドル 1, 2 および 3 は、このジオメトリ軸に割当てられていない。	
関連性	IS "Activate handwheel" (DB21, DBX12.0 ~ DBX12.2 ff)	

DB21, ... DBB40; 46; 52 ビット 6;7 データブロック	プラスおよびマイナス方向への (ジオメトリ軸) 移動コマンド 軸／スピンドルからの信号 (NCK → PLC)	
エッジ評価: なし	信号更新: 周期的	信号が有効となる最初の SW バージョン: 1.1
信号ステータス 1 あるいは信号 遷移 0 → 1	軸は、どちらかの方向へトラバース動作されなければならない。コマンドは、選択モードに応じて、次のように検出される。 - JOG モード: プラスまたはマイナスのトラバースキーで。 - REF モード: トラバースキーによって、基準点まで軸を移動する。 - AUT/MDI モード: 対象の軸の座標値を含んでいるプログラムブロックの実行。	
信号ステータス 0 あるいは信号 遷移 1 → 0	関連軸の方向への移動コマンドが出されていないか、トラバース動作が既に終了している。 ・ JOG モード: - 移動コマンドは、現在の設定 "continuous-trigger or momentary-trigger mode" (連続検出または一時検出モード) (インタフェース信号 "Traverse keys plus and minus" (プラスおよびマイナスのトラバースキー)) に応じてリセットされる。 - ハンドルを使用したトラバース中に。 ・ REF モード: 基準点に到達した時。 ・ AUT/MDI モード: - プログラムブロックが実行された (更に、次のブロックには、対象の軸の座標値が含まれていない)。 - "RESET" による強制終了など。 - IS "Axis disable" (軸ディセーブル) が有効である。	
アプリケーション	(たとえば、回転テーブル上の) クランピングで軸クランピングを放すこと。 注: 移動コマンドが出されるまでクランピングが放されない場合には、これらの軸の連続パス運転は実行できない。	
関連性	ジオメトリ軸 -- IS: "Traverse key plus" および "Traverse key minus" (DB21, DBX12.7 または DBX12.6 ff)	

DB21, ... DBB41; 47; 53 ビット 0 – 6 データブロック	ジオメトリ軸 (1, 2, 3) 用の有効マシンファンクション INC1, ..., continuous jogging 連続 JOG 送り 軸/スピンドルからの信号 (NCK → PLC)	
エッジ評価: なし	信号更新: 周期的	信号が有効となる最初の SW バージョン: 1.1
信号ステータス 1 あるいは信号 遷移 0 → 1	PLC インタフェースは、ジオメトリ軸に対し JOG モードマシンファンクションが有効であることを伝える信号を受取る。 トラバースキーの作動またはハンドル回転への反応は、マシンファンクションが有効であるかどうかによって異なる (2.4.2 「■連続 JOG 送り」と「■相対 JOG 送り (INC)」を参照)。	
信号ステータス 0 あるいは信号 遷移 1 → 0	対象のマシンファンクションが有効でない。	
関連性 ....	ジオメトリ軸 -- IS "Machine function INC1,..., continuous jogging" (DB21, ... DBB13 ff)	

DB21, ... DBX33.3 データブロック	ハンドルオーバーライドが有効 軸/スピンドルからの信号 (NCK → PLC)	
エッジ評価: なし	信号更新: 周期的	信号が有効となる最初の SW バージョン: 2.1
信号ステータス 1 あるいは信号 遷移 0 → 1	ファンクション "Handwheel override in AUTOMATIC mode" (AUTOMATIC モードのハンドルオーバーライド) が、プログラムされたパス軸に対し有効である。 最初のジオメトリ軸のハンドルパルスは、プログラムされたパスフィードレートの範囲内で、速度オーバーライドとして働く。	
信号ステータス 0 あるいは信号 遷移 1 → 0	ファンクション "Handwheel override in AUTOMATIC mode" (AUTOMATIC モードのハンドルオーバーライド) が、プログラムされたパス軸に対し有効でない。 次の場合、有効ハンドルオーバーライドが有効でない。 <ul style="list-style-type: none"> <li>パス軸が目標位置に到達した。</li> <li>チャンネル別 IS "Delete distance-to-go" (DB21, ... DBX6.2) によって移動距離が削除される。</li> <li>リセットが実行される。</li> </ul>	

## ■ 軸／スピンドル別信号

### ■ 軸／スピンドルへの信号の概要

DB 31, ...	軸／スピンドルへの信号							
DBB	ビット 7	ビット 6	ビット 5	ビット 4	ビット 3	ビット 2	ビット 1	ビット 0
37	トラバースキー プラス	トラバースキー マイナス	早送り オーバー ライド	トラバース スキー ディス エーブル	フィード ホールド スピンド ルホールド	3	2	1
40	マシンファンクション							
		連続	可変 INC	10000 INC	1000 INC	100 INC	10 INC	1 INC

### ■ 軸／スピンドルへの信号の説明

DB31, ... DBX4.0; 4.1; 4.2 データブロック	ハンドル起動 (1 ~ 3) 軸／スピンドルへの信号 (PLC → NCK)	
エッジ評価: なし	信号更新: 周期的	信号が有効となる最初の SW バージョン: 1.1
信号ステータス 1 あるいは信号 遷移 0 → 1	<p>この PLC インタフェース信号は、ハンドル 1, 2 または 3 にこのマシン軸が割当てられるかどうか、または、どのハンドルにも割当てられないかを定義する。</p> <p>軸には、一度に 1 つのハンドルしか割当てることができない。</p> <p>複数のインタフェース信号 "Activate handwheel" (ハンドル起動) がセットされた場合には、"Handwheel 1" (ハンドル 1), "Handwheel 2" (ハンドル 2), "Handwheel 3" (ハンドル 3) の順番に適用される。</p> <p>割当てが有効の時には、ハンドルを使用した JOG モードでのマシン軸のトラバース、または、AUTOMATIC もしくは MDI モードでの DRF オフセットの発生が可能になる。</p>	
信号ステータス 0 あるいは信号 遷移 1 → 0	ハンドル 1, 2 および 3 が、このジオメトリ軸に割当てられていない。	
アプリケーション	PLC ユーザプログラムは、このインタフェース信号で、軸のハンドルを回す影響をディスエーブルできる。	
関連性	ジオメトリ軸 -- IS "Handwheel active" (DB31, ... DBX64.0 ~ DBX64.2)	

DB31, ... DBX4.4 データブロック	トラバースキーディスエーブル 軸／スピンドルへの信号 (PLC → NCK)	
エッジ評価：なし	信号更新：周期的	信号が有効となる最初の SW バージョン：1.1
信号ステータス 1 あるいは信号遷移 0 → 1	プラスおよびマイナスのトラバースキーは、対象のマシン軸に対し影響を与えない。したがって、マシン制御パネル上でトラバースキーを使用して、JOG でマシン軸をトラバースすることができない。 トラバース動作中に、トラバースキーディスエーブルが起動すると、マシン軸は停止する。	
信号ステータス 0 あるいは信号遷移 1 → 0	プラスおよびマイナスのトラバースキーがイネーブルされる。	
アプリケーション	運転モードに応じて、PLC ユーザプログラムから、JOG モードのトラバースキーでマシン軸の手動移動をディスエーブルすることができる。	
関連性	IS "Traverse key plus" および "Traverse key minus" (DB31, ... DBX4.7 or DBX4.6)	

DB31, ... DBX4.5 データブロック	Rapid traverse override 早送りオーバライド 軸／スピンドルへの信号 (PLC → NCK)	
エッジ評価：なし	信号更新：周期的	信号が有効となる最初の SW バージョン：1.1
信号ステータス 1 あるいは信号遷移 0 → 1	インタフェース信号 "Rapid traverse override"（早送りオーバライド）が "Traverse key plus"（プラスのトラバースキー）および "Traverse key minus"（マイナスのトラバースキー）とともにセットされた場合には、対象のマシン軸は、早送りでトラバースする。 早送りフィードレートは、マシンデータ JOG_VELO_RAPID で定義される。 早送りオーバライドは、次の JOG モードで有効である。 <ul style="list-style-type: none"> <li>- 連続 JOG 送り</li> <li>- 相対 JOG 送り</li> </ul> 早送りオーバライドが有効の時、速度は早送りオーバライドスイッチで修正できる。	
信号ステータス 0 あるいは信号遷移 1 → 0	マシン軸は、定義された JOG 速度 (SD: JOG_SET_VELO または MD: JOG_VELO) でトラバースする。	
この信号は次の場合は無意味	<ul style="list-style-type: none"> <li>- AUTOMATIC および MDI の運転モード</li> <li>- 基準点への接近 (JOG モード)</li> </ul>	
関連性	IS "Traverse key plus" および "Traverse key minus" (DB31, ... DBX4.7 or DBX4.6) IS "Axial feedrate/spindle speed override" (DB31, ... DBB0)	



DB31, ... DBB4.7, 4.6 データブロック	プラスおよびマイナスのトラバースキー 軸／スピンドルへの信号 (PLC → NCK)	
エッジ評価：なし	信号更新：周期的	信号が有効となる最初の SW バージョン: 1.1
信号ステータス 1 あるいは信号 遷移 0 → 1	<p>選択したマシン軸は、プラスとマイナスのトラバースキーを使って、JOG モードで、両方の方向へトラバースできる。</p> <p>有効なマシンファンクションおよび設定 "Continuous-trigger or momentary-trigger mode" (連続 JOG 送り用の SD: JOG_CONT_MODE_LEVELTRIGGRD, 相対 JOG 送り用の MD: JOG_INC_MODE_LEVELTRIGGRD) に応じて、信号遷移は、次のように様々な反応を引き起こす。</p> <p>ケース 1: 連続検出モードの連続 JOG 送り</p> <p>インタフェース信号が 1 にセットされている限り (軸位置が起動制限に到達していない間)、トラバース軸は、関連する方向にトラバースする。</p> <p>ケース 2: 一時検出モードの連続 JOG 送り</p> <p>信号ステータスが最初に 0 → 1 に変化した時、マシン軸は、関連方向へのトラバースを開始する。このトラバース動作は、信号ステータスが 1 → 0 に変化しても継続する。0 → 1 の新しい信号ステータス遷移 (同じ移動方向で) が生じると、トラバース動作が再び停止する。</p> <p>ケース 3: 連続検出モードの相対 JOG 送り</p> <p>信号 1 で、マシン軸は、設定インクリメントでトラバースを開始する。インクリメントがトラバースされる前に、信号が 0 ステータスに変化する場合、トラバース動作が中断される。信号ステータスが再び 1 に変化すると、移動が継続する。</p> <p>インクリメントが完全にトラバースされるまで、マシンは、上記のように停止と継続を数回繰返すことができる。</p> <p>ケース 4: 一時検出モードの相対 JOG 送り</p> <p>信号ステータスが最初に 0 → 1 に変化した時、マシン軸は、設定インクリメントのトラバースを開始する。軸がインクリメントを移動する前に、同じトラバース信号の信号ステータスが 0 → 1 に再び変化した時、トラバース動作が強制終了する。</p> <p>インクリメントは、終端までトラバースされない。</p> <p>両方のトラバース信号 (プラスおよびマイナス) が同時にセットされると、移動しないか、現在の移動が強制終了する。</p> <p>PLC インタフェース信号 "Traverse key disable" (トラバースキーディスエーブル) によって、各マシン軸に対するトラバースキーの影響が個別にディスエーブルできる。</p>	
信号ステータス 0 あるいは信号 遷移 1 → 0	上記のケース 1 ～ 4 を参照。	
この信号は次の場合は無意味	AUTOMATIC および MDI の運転モード	
アプリケーション	(ジオメトリ軸として) チャンネル別 PLC インタフェースを介して既にトラバースを開始している場合、マシン軸は、JOG モードでトラバースすることができない。 アラーム 20062 が検出される。	
例外、エラー	インデックス軸	
関連性	IS "Traverse keys plus and minus for geometry axes" (DB21, ... DBX12.7 and 12.6 ff) IS "Traverse key disable" (DB31, ... DBX4.4)	

DB31, ... DBX5.6 データブロック	連続マシンファンクション 軸／スピンドルへの信号 (PLC → NCK)	
エッジ評価：なし	信号更新：周期的	信号が有効となる最初の SW バージョン：1.1
信号ステータス 1 あるいは信号 遷移 0 → 1	マシンファンクション "Continuous jogging"（連続 JOG 送り）が選択される。 JOG モードで、プラスおよびマイナスのトラバースキーを使用すると、関連 マシン軸がトラバースできる。	
信号ステータス 0 あるいは信号 遷移 1 → 0	マシンファンクション "Continuous jogging"（連続 JOG 送り）が選択されない。	
関連性	IS "Active machine function INC 1,..., continuous" (DB31, ... DBB65) IS "Machine function INC1,...,INC10000" (DB31, ... DBB5)	

DB31, ... DBB5 ビット 0 – 5 データブロック	マシンファンクション INC1, INC10, INC100, INC1000, INC10000, INCvar 軸／スピンドルへの信号 (PLC → NCK)	
エッジ評価：なし	信号更新：周期的	信号が有効となる最初の SW バージョン：1.1
信号ステータス 1 あるいは信号 遷移 0 → 1	このインタフェース信号は、トラバースキーが押されるか、ハンドルが次の移動止め位置まで回される時に、マシン軸がトラバースするインクリメント回数を定義する。JOG モードは、これに対し有効でなければならない（例外：DRF の場合）。 インクリメントサイズは、次のように、インタフェース信号に割当てられる。 <ul style="list-style-type: none"> <li>- INC1 ~ INC10000: 一般マシンデータで JOG_INCR_SIZE_TAB</li> <li>- INCvar: 一般設定データ JOG_VAR_INCR_SIZE で</li> </ul> 選択したマシンファンクションが有効になると、この情報の信号が、PLC インタフェース (IS "Active machine function INC1;...") に送られる。 インタフェースで、複数のマシンファンクション信号 (INC1, INC...or "Continuous jogging") が同時に選択された場合、どのマシンファンクションも、制御装置は起動できない。	
信号ステータス 0 あるいは信号 遷移 1 → 0	対象のマシンファンクションが選択されない。 軸がインクリメントをトラバースしている間、このマシンファンクションが選択解除されるか切換えられると、この移動も強制終了する。	
関連性	ジオメトリ軸 -- IS "Active machine function INC1,..." (DB31, ... DBB65) IS "Machine function continuous" (DB31, ... DBX5.6)	

## ■ 軸／スピンドルからの信号の概要

DB 31, ...	軸／スピンドルへの信号							
	ビット 7	ビット 6	ビット 5	ビット 4	ビット 3	ビット 2	ビット 1	ビット 0
64	トラバースキー プラス	マイナス				3	2	1
65	マシンファンクション							
		連続	可変 INC	10000 INC	1000 INC	100 INC	10 INC	1 INC

## ■ 軸／スピンドルからの信号の説明

DB31, ... DBX62.1 データブロック	ハンドルオーバーライドが有効 軸／スピンドルからの信号 (NCK → PLC)	
エッジ評価: なし	信号更新: 周期的	信号が有効となる最初の SW バージョン: 2.1
信号ステータス 1 あるいは信号 遷移 0 → 1	<p>ファンクション "Handwheel override in AUTOMATIC mode" (AUTOMATIC モードのハンドルオーバーライド) は, プログラムされた位置決め軸 (FDA[AXi]) に対して有効である。この軸のハンドルパルスは, プログラムされた軸フィードレートの範囲内で, (FDA=0 の場合) パス設定として作用し, (FDA&gt;0 の場合) 速度オーバーライドとして作用する。</p> <p>"Handwheel override in AUTOMATIC mode" (AUTOMATIC モードのハンドルオーバーライド) が平行位置決め軸 (FC15 で) に対し有効の時, インタフェース信号もセットされる。</p>	
信号ステータス 0 あるいは信号 遷移 1 → 0	<p>プログラムされた位置決め軸 (または平行位置決め軸) に対し, ファンクション "Handwheel override in AUTOMATIC mode" (AUTOMATIC モードのハンドルオーバーライド) が有効にならない。</p> <p>有効なハンドルオーバーライドは, 次の時に有効にならない。</p> <ul style="list-style-type: none"> <li>位置決め軸が目標位置に到達した。</li> <li>軸別 IS "Delete distance-to-go" (DB31, ... DBX2.2) によって, 移動距離が削除される。</li> <li>リセットが実行された。</li> </ul>	

DB31, ... DBX64.0; 64.1; 64.2 データブロック	有効なハンドル (1 ~ 3) 軸／スピンドルからの信号 (NCK → PLC)	
エッジ評価：なし	信号更新：周期的	信号が有効となる最初の SW バージョン：1.1
信号ステータス 1 あるいは信号 遷移 0 → 1	これらの PLC インタフェース信号は、ハンドル 1, 2 または 3 にマシン軸が割 当てられるか、または、どのハンドルにも割当てられないかどうかをフィード バックする。 軸には、一度に 1 つのハンドルしか割当てることができない。 複数のインタフェース信号 "Activate handwheel" (ハンドル起動) がセットさ れた場合、"Handwheel 1" (ハンドル 1), "Handwheel 2" (ハンドル 2), "Handwheel 3" (ハンドル 3) の順番に適用される。 割当てが有効の時には、ハンドルを使用した JOG モードでのマシン軸のトラ バースや、AUTOMATIC または MDI モードでの DRF オフセットの発生が可能 になる。	
信号ステータス 0 あるいは信号 遷移 1 → 0	ハンドル 1, 2 および 3 は、このジオメトリ軸に割当てられていない。	
関連性	IS "Activate handwheel" (DB31, DBX4.0 ~ DBX4.2) IS "Handwheel selected" (DB10, DBB100.6 ff)	

DB31, ... DBX64.7, 64.6 データブロック	Plus and minus traverse keys プラスおよびマイナスのトラバースキー 軸／スピンドルからの信号 (NCK → PLC)	
エッジ評価：なし	信号更新：周期的	信号が有効となる最初の SW バージョン：1.1
信号ステータス 1 あるいは信号 遷移 0 → 1	軸は、どちらかの方向へトラバース動作されなければならない。 コマンドは、選択したモードに応じて、次のように検出される。 <ul style="list-style-type: none"> <li>- JOG モード：プラスまたはマイナスのトラバースキーで。</li> <li>- REF モード：トラバースキーで、軸を基準点に移動させる。</li> <li>- AUT/MDI モード：対象の軸の座標値が含まれているプログラムブロック の実行。</li> </ul>	
信号ステータス 0 あるいは信号 遷移 1 → 0	関連軸への方向の移動コマンドが出されていないか、トラバース動作が既に終 了している。 <ul style="list-style-type: none"> <li>• JOG モード： <ul style="list-style-type: none"> <li>- 移動コマンドは、現在の設定 "continuous-trigger or momentarytrigger mode" (連続検出または一時検出モード) に応じてリセットされる (インタ フェース信号 "Traverse keys plus and minus" を参照)。</li> <li>- ハンドルを使用したトラバース中に。</li> </ul> </li> <li>• REF モード： <ul style="list-style-type: none"> <li>- 基準点に到達した時。</li> </ul> </li> <li>• AUT/MDI モード： <ul style="list-style-type: none"> <li>- プログラムブロックが既に実行されている (更に、対象の軸の座標値が 次のブロックに含まれている)。</li> <li>- "RESET" による強制終了など。</li> <li>- IS "Axis disable" (軸ディスエーブル) が有効である。</li> </ul> </li> </ul>	
アプリケーション	(たとえば、回転テーブル上で) クランピングにより軸のクランピングを放す こと。 注：移動コマンドが出されるまでクランプが放されていない場合、これらの軸 は、連続パス制御の下で運転させることができない。	
関連性	IS "Traverse key plus" および "Traverse key minus" (DB31, ... DBX4.7 or DBX4.6)	

DB31, ... DBB65 ビット 0 – 6 データブロック	有効なマシンファンクション INC1, ..., continuous jogging 連続 JOG 送り 軸／スピンドルからの信号 (NCK → PLC)	
エッジ評価：なし	信号更新：周期的	信号が有効となる最初の SW バージョン: 1.1
信号ステータス 1 あるいは信号 遷移 0 → 1	PLC インタフェースは、JOG モードマシンファンクションがマシン軸に対し有効であることを通知する信号を受取る。 トラバースキーが押されるかハンドルが回された時、その結果は、有効なマシンファンクションによって異なる（2.4.2「 <b>■</b> 連続 JOG 送り」と「 <b>■</b> 相対 JOG 送り (INC)」を参照）。	
信号ステータス 0 あるいは信号 遷移 1 → 0	対象のマシンファンクションが有効でない。	
関連性	IS "Machine function INC1,..., continuous jogging" (DB31, ... DBB5)	

## 2.4.6 例

なし

## 2.4.7 データフィールド, リスト

### ■ インタフェース信号

DB 番号	ビット, バイト	名称	参照
NC への信号および NC からの信号			
10	97, 98, 99	Channel number for geometry axis handwheel 1, 2, 3 (ジオメトリ軸ハンドル 1, 2, 3 のチャンネル番号)	
10	100, 101, 102	Axis number for handwheel 1, 2, 3, handwheel selected and machine axis (ハンドル 1, 2, 3, 選択ハンドルおよびマシン軸の軸番号)	
モードグループ別			
11, ...	0.2	JOG mode (JOG モード)	1.10 (K1)
11, ...	4.2	Active JOG mode K1 (有効な JOG モード)	1.10 (K1)
チャンネル別			
21, ...	0.3	Activate DRF (DRF 起動)	
21, ...	12.2, 12.1, 12.0 16.2, 16.1, 16.0 20.2, 20.1, 20.0	Activate handwheel 1, 2, 3 (ハンドル 1, 2, 3 の起動)	
21, ...	12.4, 16.4, 20.4	Traverse key disable (トラバースキーディスエーブル)	
21, ...	12.5, 16.5, 20.5	Rapid traverse override (早送りオーバライド)	
21, ...	12.7, 12.6, 16.7, 16.6, 20.7, 20.6	Traverse keys plus and traverse keys minus (プラスのトラバースキーおよびマイナスのトラバースキー)	
21, ...	13, 17, 21	Geometry axis machine function INC1 ... continuous jogging (ジオメトリ軸マシンファンクション INC1 ... 連続 JOG 送り)	
21, ...	24.3	DRF selected (DRF 選択)	
21, ...	40.2, 40.1, 40.0, 46.2, 46.1, 46.0, 52.2, 52.1, 52.0	Handwheel 1, 2, 3 active (有効なハンドル 1, 2, 3)	
21, ...	40.7, 40.6, 46.7, 46.6, 52.7, 52.6	Travel command plus and travel command minus (プラスへの移動コマンドおよびマイナスへの移動コマンド)	

チャンネル別			
21, ...	41, 47, 53	Geometry axis active machine function INC1 ... continuous (ジオメトリ軸有効マシンファンクション INC1 ... 連続)	
21, ...	33.3	Handwheel override active for path axes (SW2 and higher) (パス軸の有効ハンドルオーバーライド (SW2 以降))	
21, ...	30.0 30.1 30.2	Activate handwheel 1 as contour handwheel (輪郭ハンドルとしてハンドル 1 を起動する) Activate handwheel 2 as contour handwheel (輪郭ハンドルとしてハンドル 2 を起動する) Activate handwheel 3 as contour handwheel (輪郭ハンドルとしてハンドル 3 を起動する)	
21, ...	30.3	Simulation of contour handwheel ON (輪郭ハンドルシミュレーションオン)	
21, ...	30.4	Negative direction simulation of contour handwheel (負の方向への輪郭ハンドルのシミュレーション)	
21, ...	37.0 37.1 37.2	Handwheel 1 active as contour handwheel (輪郭ハンドルとして有効なハンドル 1) Handwheel 2 active as contour handwheel (輪郭ハンドルとして有効なハンドル 2) Handwheel 3 active as contour handwheel (輪郭ハンドルとして有効なハンドル 3)	
21, ...	100.5 101.5 102.5	Handwheel 1 active as contour handwheel (輪郭ハンドルとして有効なハンドル 1) Handwheel 2 active as contour handwheel (輪郭ハンドルとして有効なハンドル 2) Handwheel 3 active as contour handwheel (輪郭ハンドルとして有効なハンドル 3)	
軸／スピンドル別			
31, ...	0	Feedrate/spindle speed override (フィードレート／スピンドル速度オーバーライド)	1.18 (V1)
31, ...	1.7	Override active (オーバーライド有効)	1.18 (V1)
31, ...	2.2	Axial delete distance-to-go (軸移動距離の削除)	
31, ...	4.2, 4.1, 4.0	Activate handwheel 1, 2, 3 (ハンドル 1, 2, 3 の起動)	
31, ...	4.4	Traverse key disable (トラバースキーディセーブル)	
31, ...	4.5	Rapid traverse override (早送りオーバーライド)	
31, ...	4.7, 4.6	Traverse keys plus and traverse keys minus (プラスのトラバースキーおよびマイナスのトラバースキー)	
31, ...	5.6	Machine function continuous jogging (マシンファンクション連続 JOG 送り)	
31, ...	5.6, 5.5, 5.4, 5.3, 5.2, 5.1, 5.0	Machine function (マシンファンクション) continuous, var. INC, 10000 INC, 連続 1000 INC, 100 INC, 10 INC, 1 INC	
31, ...	60.7, 60.6	Position reached with exact stop coarse/fine (正確停止 (粗／微) で位置到達)	1.3 (B1)
31, ...	64.2, 64.1, 64.0	Handwheel 1, 2, 3 active (有効なハンドル 1, 2, 3)	
31, ...	64.7, 64.6	Travel command plus and travel command minus (プラスの移動コマンドおよびマイナスの移動コマンド)	
31, ...	65	Active machine function INC1 ... continuous (有効なマシンファンクション INC1 ... 連続)	

31, ...	62.1	Handwheel override active, for positioning axes and concurrent positioning axes (SW2 and higher) (位置決め軸および平行位置決め軸の有効ハンドルオーバーライド (SW2 以降))	
---------	------	---	--

## ■ マシンデータ

DB 番号	ビット, バイト	名称	参照
<b>一般 (\$MN_ ... )</b>			
10000	AXCONF_MACHAX_NAME_TAB [n]	マシン軸名称 [n = 軸番号]	1.11 (K2)
11300	JOG_INC_MODE_LEVELTRIGGRD	続検出モードの INC および REF	
11310	HANDWH_REVERSE	逆方向への移動の定義	
11320	HANDWH_IMP_PER_LATCH [n]	移動止め位置のたびのハンドルパルス [= ハンドルパルス番号: 0 ~ 2]	
11330	JOG_INCR_SIZE_TAB [n]	インクリメントサイズ INC / ハンドル (n = インクリメントインデックス: 0 ~ 4)	
11340	ENC_HANDWHEEL_SEGMENT_NR	第 3 のハンドル: バスセグメント	FBMA
11342	ENC_HANDWHEEL_MODULE_NR	第 3 のハンドル: ドライブ番号 / 測定回路番号	FBMA
11344	ENC_HANDWHEEL_INPUT_NR	第 3 のハンドル: モジュール / 測定回路カード上の入力	
11346	HANDWH_TRUE_DISTANCE	ハンドルによるパス定義または速度指定	FBMA
<b>チャンネル別 (\$MC_ ... )</b>			
20060	AXCONF_GEOAX_NAME_TAB [n]	チャンネルのジオメトリ軸 [n = ジオメトリ軸番号]	1.11 (K2)
20100	DIAMETER_AX_DEF	トランズバース軸ファンクションによるジオメトリ軸	1.14 (P1)
20620	HANDWH_GEOAX_MAX_INCREMENT_SIZE	ジオメトリ軸の範囲指定	
20622	HANDWH_GEOAX_MAX_INCREMENT_VSIZE	パス速度オーバーライド	
20624	HANDWH_CHAN_STOP_CONDITION	チャンネル別 VDI インタフェース信号への応答	
<b>軸 / スピンドル別 (\$MA_ ... )</b>			
30450	IS_CONCURRENT_POS_AX	リセット時の初期設定: 中立軸またはチャンネル軸	1.10 (P2)
31090	JOG_INCR_WEIGHT	INC / ハンドルのインクリメント補正	
32000	MAX_AX_VELO	最高軸速度	1.8 (G2)
32010	JOG_VELO_RAPID	JOG 早送り	
32020	JOG_VELO	通常の軸速度	
32040	JOG_REV_VELO_RAPID	早送りオーバーライドによる JOG モードの回転フィードレート	
32050	JOG_REV_VELO	JOG モードの回転フィードレート	
32060	POS_AX_VELO	位置決め軸速度の初期設定	1.10 (P2)
32080	HANDWH_MAX_INCREMENT_SIZE	選択インクリメントサイズの範囲指定	
32082	HANDWH_MAX_INCREMENT_VELO_SIZE	速度オーバーライドの選択インクリメントの制限	



32084	HANDWH_STOP_COND	軸別 VDI インタフェース信号への応答 ビット 0...5	
32090	HANDWH_VELO_OVERLAY_FACTOR	ハンドル速度 (DRF で) に対する JOG 速度の比率	
35130	GEAR_STEP_MAX_VELO_LIMIT[n]	ギヤステージの最高速度	1.17 (S1)

## ■ 設定データ

DB 番号	ビット, バイト	名称	参照
一般 (\$MN_...)			
41010	JOG_VAR_INCR_SIZE	可変インクリメント INC / ハンドルのサイズ	
41050	JOG_CONT_MODE_LEVEL_TRIGGRD	JOG 連続モード	
41100	JOG_REV_IS_ACTIVE	有効な JOG での回転フィードレート	
41110	JOG_SET_VELO	直線軸の JOG 速度 (G94)	
41120	JOG_REV_SET_VELO JOG	JOG 速度 (G95)	
41130	JOG_ROT_AX_SET_VELO	回転軸の JOG 速度	
41200	JOG_SPIND_SET_VELO	スピンドルの JOG 速度	

## ■ アラーム

アラームの詳細説明については次を参照してください。

参照： アラーム一覧

あるいは MMC 101/102/103 を有するシステムのオンラインヘルプ

---

---

## 2.5 補正機能 (K3)

### 2.5.1 概略説明

#### 補正の目的

理論的座標と、実際の機械軸とのずれ、パワー伝達障害、測定系誤差が生じると、工作機械の精度が損なわれます。大型のワークが加工されるときには温度差やパワーの伝達効率により精度が大きく減少することがあります。

このようなずれの中には、機械のセットアップ時測定した値と、その後に位置実際値エンコーダなどのセンサ装置によって読取られた値を基準にして運転時に補正されるものもあります。

#### 補正

CNC には工作機械の精度を高めるために、誤差の本質的な原因を補正する機能があります。

- 温度補正
- バックラッシュ補正
- 補間補正
  - LEC  
(リードスクリュウ誤差および測定系誤差補正)
  - 直角度補正  
(直角度および角度誤差の補正)
- 追従誤差補正 (動的なフィードフォワード制御)
- 象現突起補正
- ドライブの電子釣合いおもり

これらの補正機能は軸別マシンデータを使って各々のマシンに別個に設定できます。

---

## 補間補正

"Interpolatory compensation"（補間補正）機能を使えば、位置に関連するずれ（例えば、リードスクリュウ誤差、測定系誤差、垂れなど）を修正できます。

補正值は、インストール時に測定され位置関連値としてテーブルに保存されます。運転時に、軸は線補間時の補間点間で補正されます。

## 摩擦補正

円形輪郭加工時の輪郭精度を著しく改善するうえで、"friction compensation"（摩擦補正）機能が特に効果的です。軸の回転方向を変更すると、摩擦条件が変化するため、速度がゼロになると（クォッドラント遷移点で）輪郭誤差が生じます。「摩擦補正」は、輪郭がはじめて加工されるとき確実にこの誤差を補正します。

YS840DI システムは、摩擦、バックラッシュ、ねじれを補正する自己学習プロセスに最適パラメータを採用しています。このシステムにより、いつでも簡単かつ自動的に再最適化を行うことができます。

摩擦補正システムは円試験を使って非常に簡単にインストールできます。円輪郭を加工してみて、プログラムされた半径から機械位置がどのくらいずれているか（クォッドラント遷移点で最も著しい）を測定し、グラフィックスで示します。円試験は「インストールツール」機能です。

## 起動

入力データが利用できるようになるとすぐに、制御装置の全ての運転モードで補正がアクティブになります。位置実際値が必要な補正は、軸が基準点に達するまでアクティブにはなりません。

## 位置表示

実位置および指令位置表示は補正值を考慮せず、理論的な位置を示します。補正值は、「診断」オペレーティングエリアの "Service axes"（サービス軸）画面に表示されます。

## 2.5.2 詳細説明

### ■ 温度補正

### ■ 一般事項

#### 温度効果による変形

ドライブ装置が発する熱や高い室温（例えば、日光、通風が原因で）は、マシンベースやマシン部品の膨張の原因となります。膨張の程度は、温度やマシン部品の熱伝導率によって異なります。

#### 効果

マシンの熱膨張により、軸の実際の位置が温度によって変わります。この現象が加工対象のワークピースの精度に悪影響を及ぼすため、実際値位置のこのような温度に関連する変化を補正（温度補正）する必要があります。

#### センサ

モータエンコーダの実際値とは別に、温度補正機能には一般に温度特性を得るために別に多くの温度センサが必要となります。

温度に関連する変化は効果が現れるまでに比較的長い時間かかるため、温度プロファイルの入手と処理を PLC により 1 分 (one-minute) サイクルで行います。

#### 誤差曲線

温度補正を実施するために、軸の位置決め範囲にわたる実際値オフセットを任意の温度 (T) で測定してプロットする必要があります。これによってその温度値での誤差曲線が得られます。誤差曲線は様々な温度で作成する必要があります。

#### 誤差曲線特性

上記の方法で作成された誤差曲線は通常は図 2-22 に示すような特性を示します。軸について位置基準点 P(0) を選択し、温度が変化するのに合わせて、基準点でのオフセット（温度補正の「位置に依存しない要素」に対応）を測定し、さらに長さが変化するので基準点からの距離に比例して大きくなる他の位置点でのオフセット（温度補正の「位置に依存する要素」に対応）も測定します。

任意の温度  $T$  において，通常，誤差曲線はほぼ直線となります。この直線の傾きと基準点は温度によって変わります（図 2.22 を参照）。

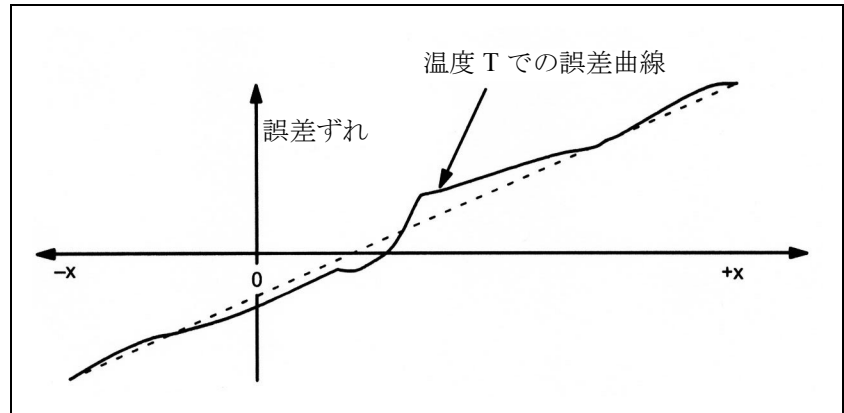


図 2.22 熱膨張の誤差曲線の例

### 補正の式

補正值  $\Delta K_x$  は，この軸の現在の機械位置  $P_x$  と温度  $T$  に基づいて，次の式から計算されます。

$$\Delta K_x = K_0(T) + \tan \beta(T) * (P_x - P_0)$$

ここでは（図 2.23 を参照のこと）：

$\Delta K_x$  は，位置  $P_x$  での軸の温度補正值。

$K_0$  は，軸の位置に依存しない温度補正值。

$P_x$  は，軸の機械位置。

$P_0$  は，軸の基準位置。

$\tan \beta$  は，位置に依存する温度補正係数（近似誤差直線の傾きに対応）。

補正值は補間サイクル中に計算されます。補正值  $\Delta K_x$  が正の場合，軸は負の方向に動きます。

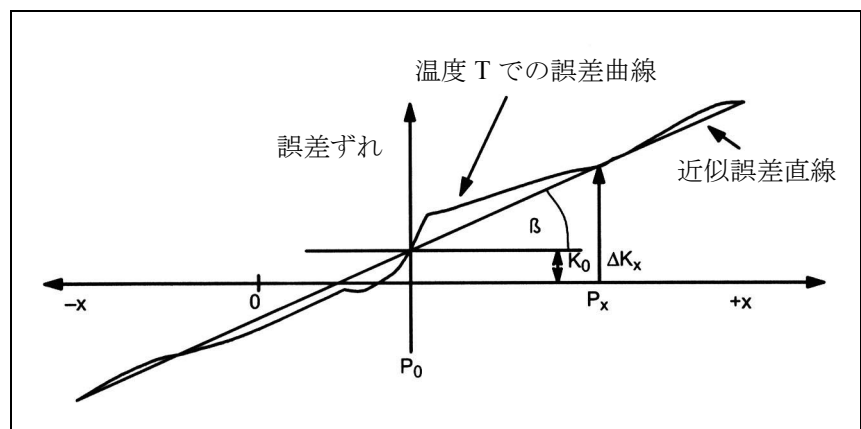


図 2.23 温度補正の近似誤差直線

---

## 温度変動

近似誤差直線は瞬時温度にしか対応しないので、温度が上下したときに新規に作成した誤差直線パラメータをその都度 NCK に送る必要があります。こうすることによってのみ、熱による膨張を効率的に補正できます。

### ■ 温度補正パラメータ

#### 温度に依存するパラメータ

様々な温度の誤差曲線を、図 2.23 のように各軸に定義することができます。それぞれの誤差曲線について、以下のパラメータを求めて設定データに入力する必要があります。

- 位置に依存する温度補正值  $K_0$   
SD 43900: TEMP\_COMP\_ABS\_VALUE
- 位置に依存する温度補正の基準位置  $P_0$   
SD 43920: TEMP\_COMP\_REF\_POSITION
- 位置に依存する温度補正の傾斜  $\tan \beta$   
SD 43910: TEMP\_COMP\_SLOPE

## 温度補正をアクティブにする

軸の MD 32750: TEMP\_COMP\_TYPE を使ってそれぞれの軸について温度補正をアクティブにすることができます。適用する温度補正タイプも選択できます。

表 2.5 MD 32750: TEMP\_COMP\_TYPE

MD 32750: TEMP_COMP_TYPE	意味	関連パラメータ
0	温度補正停止	-----
1	位置に依存しない温度補正アクティブ	SD 43900: TEMP_COMP_ABS_VALUE
2	位置に依存する温度補正アクティブ	SD 43920: TEMP_COMP_REF_POSITION SD 43910: TEMP_COMP_SLOPE
3	位置に依存しない温度補正および位置に依存する温度補正アクティブ	SD 43900: TEMP_COMP_ABS_VALUE SD 43920: TEMP_COMP_REF_POSITION SD 43910: TEMP_COMP_SLOPE

## 起動

温度補正を適用するには次の条件を満たす必要があります。

1. このオプションがイネーブルされていること。
2. 補正タイプが選択されていること (MD 32750: TEMP\_COMP\_TYPE)。
3. その補正タイプについてパラメータが定義されていること。
4. 軸が基準化されていること (IS "Referenced/synchronized 1 or 2" DB31 から 48, および DBX60.4 または 60.5 = '1')。

これらの条件が満たされると、現在位置の実際値の温度補正値が直ちに全モードで指令値に加算され、マシン軸が移動されます。

エンコーダ周波数が超過した (IS "Referenced/Synchronized 1 or 2" = 0) などの理由でその後に基準位置が失われた場合には、補正処理ルーチンは強制終了されます。

## パラメータの変更

温度 T が変化すると、温度に依存するパラメータ、すなわち ( $K_0$ ,  $\tan \beta$  および  $P_0$ ) も変化し、そのために常に PLC によって上書きされます。

工作機械製造業者は PLC ユーザプログラムを通じて軸位置と温度値との間の数学的技術的な関係を表すことができ、それによって温度補正用に様々なパラメータを計算することができます。温度パラメータは様々なサービス (FB2 (GET) "Read data" および FB3 (PUT) "Write data") を使って NCK に転送されます。

FB2 および FB3 の取扱いとパラメータ化の詳細については次を参照してください：

参照： 1.15 基本的 PLC プログラム (P3)



---

## 監視

MD 32760: COMP\_ADD\_VELO\_FACTOR（補正による速度違反）を設定することによって、各 IPO クロックサイクルで特定の速度値に加算される最大補正値を制限することができます。

このマシンデータは誤差曲線の最大の傾きを制限します。この最大の傾きを超えた場合、制御装置で補正値が制限されます。

## 平滑補正値

上記のパラメータ設定の急激な変化によって、マシンの過負荷や監視機能のエラー検出が発生しないようにするため、補正値が各サイクルについて指定された最大補正値 (MD 32760: COMP\_ADD\_VELO\_FACTOR) を超えると、直ちに内部制御装置により補正値が複数の IPO クロックサイクルに分散されます。

## 位置表示

実際の値と指令値位置表示は補正値を無視し、理想マシンの位置の値を表示します。

## 補正値の表示

現在の機械位置に属する温度および垂れ補正機能から算出された補正値合計は、"Diagnosis" オペレーティングエリアの "Service axes" 画面に出力されます。

## 例

温度補正のインストールを 1 軸の例を用いて説明します。

## 誤差曲線の計算

1 軸の温度に依存する誤差特性を次の手順で計算します。

- 1 軸をそのトラバース範囲全域にわたって（この例では 500 mm から 1500 mm まで）移動させることによって一定発熱させます。
- 100 mm の距離にある軸位置を測定します。
- リードスクリューの実際の温度を測定します。
- 20 分ごとに移動測定サイクルを実行します。

数学的技術的關係とそれに基づく温度補正パラメータを、記録データから導出します。NC によって測定された軸位置からの測定誤差ずれを図 2.24 にプロットします。

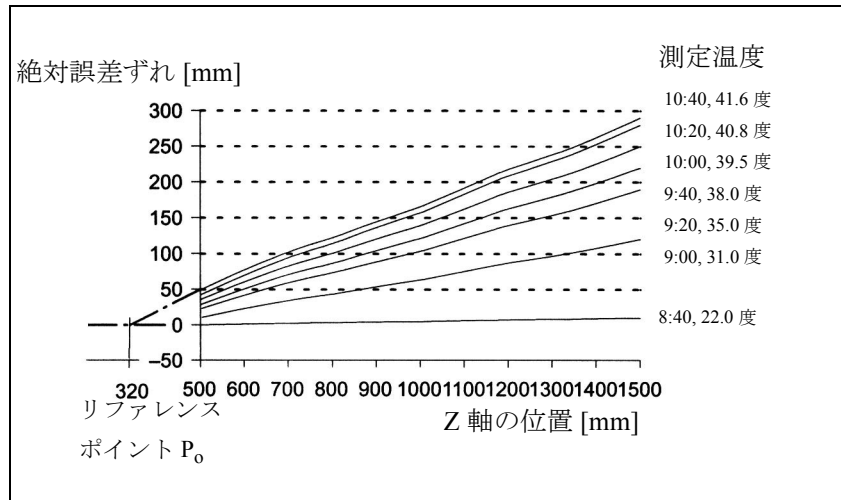


図 2.24 1 軸について求めた誤差曲線

温度補正のパラメータをここで上の測定結果に基づき設定する必要があります (図 2.24 を参照)。

基準位置  $P_0$

図 2.24 に示されているように、基準位置  $P_0$  をパラメータ化するには 2 つの方法があります。

1.  $P_0 = 0$  (位置に依存しない温度補正值  $K_0 = 0$  のとき)
2.  $P_0 \neq 0$  (位置に依存しない温度補正值  $K_0 = 0$  のとき)

この例では 2 の方法 (位置に依存しない温度補正值が常に 0) を選択します。したがって温度補正值には位置に依存する要素のみが含まれます。その結果が次のパラメータです：

- MD 32750: TEMP\_COMP\_TYPE = 2 (位置に依存する温度補正のみがアクティブ)
- $P_0 = 320 \text{ mm} \rightarrow$  SD 43920: TEMP\_COMP\_REF\_POSITION = 320 (パラメータ割当て)

### 係数 $\tan \beta$ (T)

位置に依存する温度補正係数  $\tan \beta$  の依存性を求めるために、誤差曲線の傾きを測定温度上にプロットします (図 2.25 を参照)。

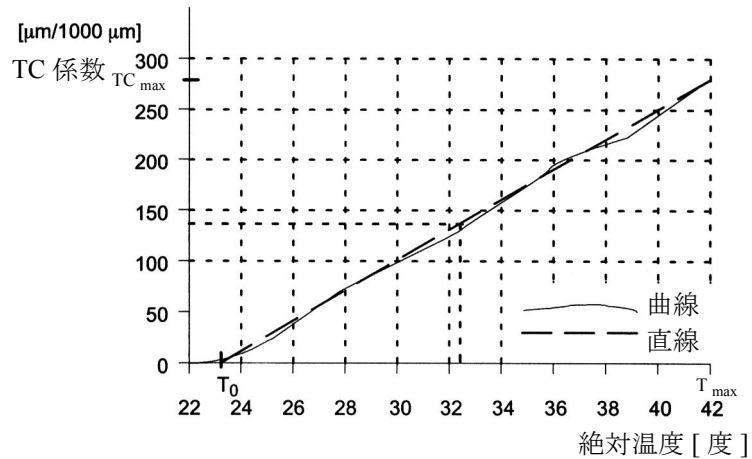


図 2.25 測定温度 T の関数としての係数  $\tan \beta$  の特性

求められた直線によって、係数  $\tan \beta$  については次のような T への依存性があります。

$$\tan \beta (T) = (T - T_0) * \frac{TC_{\max}}{T_{\max} - T_0}$$

仮定

$T_0$  = 温度 (位置に依存する誤差 = 0 の場合)

$T_{\max}$  = 最大測定温度

$TC_{\max}$  =  $T_{\max}$  についての温度係数

図 2.25 では

$T_0 = 23$  度

$T_{\max} = 42$  度

$TC_{\max} = 270 \mu m/1000 mm$

したがって：

$$\tan \beta (T) = (T - 23 \text{ 度}) * 14.21 [\mu m/1000 mm]$$

例：  $T = 32.3$  度  $\rightarrow \tan \beta = 132 \mu m/1000 mm$

PLC で上の式を使って全ての測定温度 T について係数  $\tan \beta (T)$  を簡単に計算できます。この係数は SD 43910: TEMP\_COMP\_SLOPE として NCK に転送されます。

## ■ バックラッシュ補正

### 機械的バックラッシュ

機械構成要素に全くバックラッシュがないとマシンの摩耗が過大になるので、マシンの可動部分とそのドライブ（例えば、リードスクリュー）との間のパワートレインにはわずかにバックラッシュを設けるのが普通です。このため、マシン部分と測定系との間にはバックラッシュが生じます。

### 影響

機械的バックラッシュにより、間接測定系では送り軸／主軸のトラバースパスに誤差が生じます。例えば、進行方向が逆になると、バックラッシュの程度に応じた距離だけ、軸の移動距離は長すぎたり短すぎたりします（図 2.26 および図 2.27 を参照）。

### 補正

バックラッシュを補正するには、軸別実際値を送り軸／主軸が方向を変えるたびにバックラッシュ分だけ補正します。

この量は、すべての送り軸／主軸のインストール時に MD: 32450 BACKLASH に入力することができます。2 次測定系がある場合、それぞれの測定系のバックラッシュを入力してください。

### 起動

基準点アプローチ後、全てのオペレーティングシステムで必ず、バックラッシュ補正がアクティブになります。

### 位置表示

通常は実際値と指令値位置表示は補正值を無視し、「理想マシン」の位置値を表示します。

### 補正值の表示

現在の実値に適用する補正值が、"Diagnosis" オペレーティングエリアの "Service axes" 画面に、"LEC" と 「バックラッシュ補正」 から算出された総補正として出力されます。

### 正のバックラッシュ

エンコーダが示す位置は、テーブルなどの機械構成部品の位置より進んでいます。取得した実際の位置はテーブルの真の位置より先になるため、テーブルの移動距離は短くなります（図 2.26 参照）。バックラッシュ補正值は、この場合（通常の場合）正の値として入力してください。

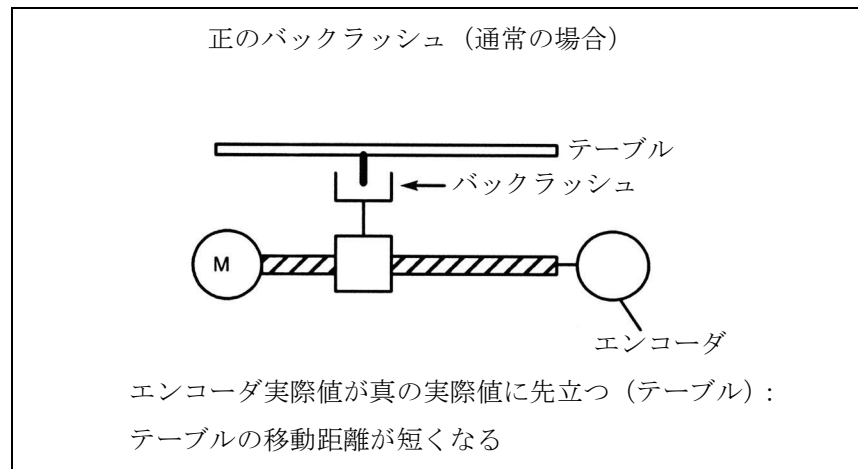


図 2.26 正のバックラッシュ（通常の場合）

## 負のバックラッシュ

エンコーダが示す位置は，テーブルなどの機械構成部品より遅れます。テーブルの移動は長くなります（図 2.27 を参照）。負の補正值を入力します。

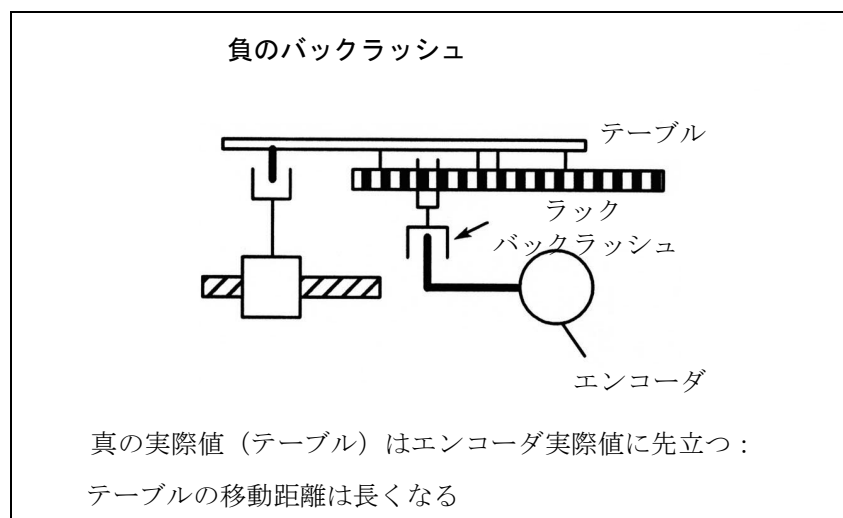


図 2.27 負のバックラッシュ

## 2 次測定系

送り軸／主軸に 2 次測定系がある場合，そこにもバックラッシュ補正を入力してください。1 次測定系とは異なる方法で 2 次測定系が取り付けられている場合，バックラッシュは 1 次測定系とは異なる場合があります。

測定系が切換えられると，関連する補正值が必ずアクティブになります。

（注）2 つのエンコーダの実際値の差異は，軸指定 MD 36500  
ENC\_CHANGE\_TOL（位置実際値切換えの最大許容量）に格納されている値を超えてはなりません。

## ■ 補間補正

## ■ 一般事項

### 補正方法

"interpolatory compensation" (補間補正) を実施するのに次の補正方法が適用されます。

1. "Leadscrew error compensation" または "Measuring system error compensation", 以後リードスクリュー誤差補正 LEC といいいます。
  2. 直角度補正または輪郭誤差補正 (SW 2 以降), 以後直角度補正といいいます。
- これら 2 つの補正方法の特性の多くは共通しています。これらは次のセクション「一般事項」で説明します。

### 用語

「補間補正」の説明には次のような用語が使用されています。

- 補正值  
実際値エンコーダが測定した軸位置と加工プログラムで要求された軸位置 (= 理論マシンの軸位置) との差異。補正值は「補正」値ともいいます。
- 基本軸  
その指令値または実際の位置が補正值の計算の基準となる軸。
- 補正軸  
その指令値または実際の位置が補正值によって修正される軸。
- 補間点  
基本軸の位置と補正軸の対応する補正值。
- 補正表  
補間点の表
- 補正関係  
基本軸と対応する補正軸の割当ならびに対応する補正テーブルの基準

### リードスクリューおよび測定系誤差

NC 制御マシンの「間接測定」の測定原理は、実際の軸位置を駆動スピンドルから導き出せるように、リードスクリューピッチは移動範囲内のどの点でも一定であるという仮定に基づいています (理想的な場合)。

しかし、製造公差のためにスピンドルには様々な程度の寸法上のずれがあります (リードスクリュー誤差)。

これらのずれの他にも、使用されている測定系が原因の (分解能の違いによる) ずれや、測定系がマシンに構築されている方法 (測定系誤差) が原因のずれ、さらにはマシンに依存する誤差要因がさらに加わります。

## 直角度による誤差

ガイドレールなどのマシン部分は負荷がかかると垂れるので、可動部分は重量の影響で位置によって「ずれ」たり「傾いたり」することがあります（図 2.30 参照）。

シリンダのような大きなワークもまた、自身の重みで垂れます。

## 角度誤差

可動軸同士が要求された角度で正確に（例えば直角に）位置決めできなければ、ゼロ点からのずれがますます大きくなって位置決め誤差が生じます。

## 補正テーブル

前述のような現象から生じる寸法上のずれは、ワークピースの加工精度に直接影響するため、位置に依存する補正值を使って適切に補正する必要があります。補正值は測定した誤差曲線から導き出し、インストール時に補正テーブルの形で制御装置に入力します。補正テーブルは補正関係ごとにそれぞれ作成しなければなりません。

補正值および追加テーブルパラメータは、特別システム変数を使って補正テーブルに入力します。

（注）補正テーブルは、MD 32700: ENC\_COMP\_ENABLE（補間補正）= 0 および／または MD 32710: CEC\_ENABLE（直角度補正イネーブル）がゼロに設定されていなければロードできません。

## 補正テーブルの入力

補正テーブルの大きさ（補間点の数）をまず最初にマシンデータに定義します。その後に電源を入れます。

補正テーブルをバックアップされた NC ユーザメモリにロードする方法には次の 2 つがあります。

- 補正テーブルを持つ NC プログラムがスタートしたときに、補正值をロードする。
- MMC のシリアルインタフェースを介して PC から補正テーブルを転送することによって補正值をロードする。

（注）補正テーブルの大きさをマシンデータに定義すると、NC は次に電源を入れたときにテーブルを作成します。これらのテーブルの初期設定は "0" です。  
補正テーブルは MMC のシリアルインタフェースを介して "Services" オペレーティングエリアから出力され、編集後にロードバックされます。

これらの補正值は、不揮発性のユーザメモリに保存されるので、制御装置の電源スイッチを切っても失われることはありません。必要に応じて更新することもできます（例えば、マシンの経年変化によるずれの変化を再調整するとき）。



#### 注意

MD 18342: MM\_CEC\_MAX\_POINTS[t]（直角度補正の補間点の最大数, SRAM）または MD 38000: MM\_ENC\_COMP\_MAX\_POINTS（補間補正の補間点の最大数, SRAM）が変更された場合、システムの電源が入ったときに、バックアップされていた NC ユーザメモリが再初期化されます。バッテリーでバックアップされていたユーザメモリのすべてのユーザデータ（例えば、ドライブと MMC マシンデータ、ツールオフセット、パートプログラム、補正テーブルなど）が削除されます。

参照： 2.14 メモリ構成（S7）

## アーカイブ

補正テーブルは、シリーズセットアップファイルを使って保存することはできません。

補正テーブルをアーカイブするには、MMC のシリアルインタフェースを介して出力する必要があります。オペレーティングエリア "Services", "Data OUT" にアーカイブする場合、次の補正タイプを選択できます。

- LEC / 測定系誤差補正 (%\_N\_AX\_EEC\_INI)
- 直角度 / 角度補正 (%\_N\_AX\_CEC\_INI)
- 象現突起補正 (%\_N\_AX\_QEC\_INI)

補正テーブルは、MMC 102/103 を使ってアーカイブファイルとして保存することもできます。

## 補間点間の直線補間

補正する移動パスを開始位置と終了位置で定義し、同じ大きさの複数の（正確な数は誤差曲線によって決定してください）小区間に分割します（図 2.28 参照）。これらの小区間を限定する実際の位置を「補間点」といいます。補正值はインストール時にそれぞれの補間点（実際の位置）ごとに入力する必要があります。2 つの補間点間に適用される補正值は、隣接する補間点の補正值を使って直線補間で作成されます（つまり、隣接する補間点同士は直線で結ばれます）。



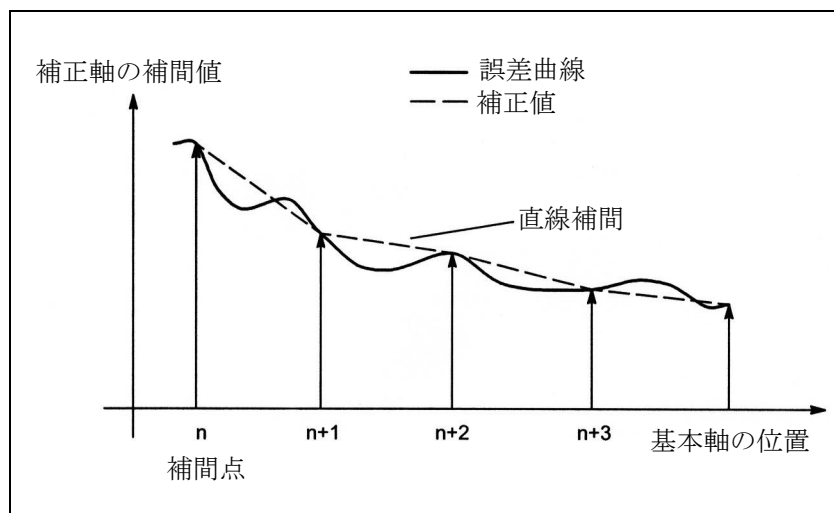


図 2.28 補間点同士の直線補間

## 基準点の補正值

補正テーブルは基準点の補正值が「ゼロ」になるように構成してください。

## ■ 測定系誤差補正 (MSEC)

### 機能

リードスクリュウ誤差補正機能は測定系誤差補正システムの一部です。

「測定系誤差補正」(以降 MSEC といいます)では、基本軸と補正軸は必ず同じになります。したがって、リードスクリュウ誤差補正機能は、補正テーブルで基本軸と補正軸を定義する必要のない軸補正です。

MSEC は、補間サイクル中に、軸別の位置実際値を補正值分だけ修正し、この修正後の値をマシン軸に直ちに適用します。位置補正值が正であれば、対応するマシン軸は負の方向に動きます。

補正值の大きさには制限はなく、監視ありません。補正によって速度や加速がオーバーすることのないように、小さな補正值を選択してください。補正值が大きいと、他の軸監視機能が警告を出すことがあります(例えば、輪郭監視、速度指令値制限)。

補正する軸が 2 次位置測定系を持つ場合、別個のテーブルを作成し、それぞれの測定系をアクティブにしてください。測定系を切換えると自動的に適正なテーブルが使用されます。

## 起動

"MSEC" は次の条件を満たして初めてアクティブになります。

- 補正値が NC ユーザメモリに保存されアクティブになっていること（電源オン後）。
- この機能が該当するマシン軸に対してアクティブになっていること (MD 32700: ENC\_COMP\_ENABLE [e] = 1)。2 次測定系に補正が必要な場合、上のマシンデータを使ってそれもイネーブルにする必要があります (e = 0: 1. measuring system; e = 1: 2. measuring system)。
- 軸が基準化されていること (IS: "Referenced/synchronized 1 or 2" DB31, ... DBX60.4 or 60.5 = '1')

これらの条件が満たされると、軸別の実値は、全モードで補正値分だけ修正され、マシン軸によって直ちに移動されます。

エンコーダ周波数が超過したなどの理由 (IS "Reference/synchronized 1 or 2"='0') でリファレンスが失われた場合には補正処理は停止します。

## 補正補間点

すべてのマシン軸およびすべての測定系（2 次測定系がインストールされている場合）に関して、補正テーブルの予約補間点数を定義し、必要メモリを MD 38000: MM\_ENC\_COMP\_MAX\_POINTS に予約する必要があります。

MD 38000: \$MA\_MM\_ENC\_COMP\_MAX\_POINTS [e,AXi]

ここで: AXi = 軸名 (X1, Y1, Z1 など)

$$\text{MM\_ENC\_COMP\_MAX\_POINTS}[e, \text{AXi}] = \frac{\$AA\_ENC\_COMP\_MAX[e, \text{AXi}] - \$AA\_ENC\_COMP\_MIN[e, \text{AXi}]}{\$AA\_ENC\_COMP\_STEP[e, \text{AXi}]} + 1$$

e = 測定系 (e = 0: 1 次測定系 ; e = 1: 2 次測定系)

## 補正テーブル

位置に関連する補正値は、補正テーブルに関連軸のシステム変数の形で保存されます。

次のエンコーダ別パラメータをテーブルに定義してください (図 2.29 参照)。

- 補正テーブルの補間点 N の補正値  
(\$AA\_ENC\_COMP [e,N,AXi])

個々の補間点（軸位置）について補正値を表に入力してください。

補間点 N は、関連の補正テーブル中に許容される補間点数によって制限されます (MD 38000: MM\_ENC\_COMP\_MAX\_POINTS)。

補正値の大きさの制限はありません。

N の許容限界 :  $0 \leq N < \text{MM\_ENC\_COMP\_MAX\_POINTS} - 1$

(注) 最初と最期の補正値は移動範囲全体にわたってアクティブになったままです。すなわち、補正テーブルが移動範囲全体をカバーしていない場合は、これらの値を "0" に設定すべきです。

- 補間点間の距離 (\$AA\_ENC\_COMP\_STEP[e,AXi])  
補間点間の距離は、関連補正テーブルの補正值間の距離に対応しています (e および AXi の意味については上記を参照してください)。
- 初期位置 (\$AA\_ENC\_COMP\_MIN[e,AXi])  
初期位置は、関連軸の補正テーブルが開始する軸位置です (= 補間点 0)。  
初期位置の補正值は \$AA\_ENC\_COMP\_STEP[e,0,AXi] です。  
初期位置より小さな位置にはすべて補間点ゼロの補正值を使用します (例外 : モジュロのあるテーブル)。
- エンド位置 (\$AA\_ENC\_COMP\_MAX[e,AXi])  
エンド位置は、関連軸の補正テーブルが終了する軸位置です (= 補間点 k)。  
エンド位置の補正值は \$AA\_ENC\_COMP\_STEP[e,k,AXi] です。  
エンド位置より大きな位置にはすべて補間点 k の補正值を使用します (例外 : モジュロのあるテーブル)。

必要となる補間点の数は次のように計算します :

$$k = \frac{\$AA\_ENC\_COMP\_MAX - \$AA\_ENC\_COMP\_MIN}{\$AA\_ENC\_COMP}$$

ここで,  $0 \leq k < MD\ 38000: MM\_ENC\_COMP\_MAX\_POINTS$

次の条件が補間点 k に適用されます :

- $k = MD\ 38000: MM\_ENC\_COMP\_MAX\_POINTS - 1$   
⇒補正テーブルが完全に利用されます。
- $k = MD\ 38000: MM\_ENC\_COMP\_MAX\_POINTS - 1$   
⇒補正テーブルが完全に利用されていません ; テーブルに入力された k より大きな値の補正值は無効です。
- $k = MD\ 38000: MM\_ENC\_COMP\_MAX\_POINTS - 1$   
⇒補正テーブルは、エンド位置を低減することによって内部的に制限されます ; k より大きな補正值は使用されません。
- モジュロ機能を持つ補正  
(\$AA\_ENC\_COMP\_IS\_MODULO[e,AXi])  
補正をモジュロ機能を使ってアクティブにする場合、補正テーブルが周期的に繰り返され、位置 \$AA\_ENC\_COMP\_MAX (= 補間点 \$AA\_ENC\_COMP[e,k,AXi]) の補正值の直後に、位置 \$AA\_ENC\_COMP\_MIN (= 補間点 \$AA\_ENC\_COMP[e,0,AXi]) の補正值がきます。  
モジュロ 360° を持つ回転軸では、初期位置 (\$AA\_ENC\_COMP\_MIN) としては 0° を、エンド点 (\$AA\_ENC\_COMP\_MAX) としては 360° をプログラムするのが適当です。

これら 2 つの位置用に入力した補正值は、同一であるか、あるいは補正值が変換点で MAX から MIN に、あるいはその逆にジャンプすべきです。

- \$AA\_ENC\_COMP\_IS\_MODULO[e,AXi] = 0:

モジュロ機能を持たない補正

- \$AA\_ENC\_COMP\_IS\_MODULO[e,AXi] = 1:

モジュロ機能を持つ補正



#### 注意

補正值を入力するときには、すべての補間点について、定義された範囲内の位置値を割当てする必要があります（ギャップがないように）。  
あるいは、これらの補間点に以前の有効な位置値を使用してください。

#### (注)

位置情報を持つテーブルパラメータは、ソフトウェアバージョン 4 かそれ以前では、測定系切換え (MD 10240 : SCALING\_SYSTEM\_IS\_METRIC) のときに位置情報を持つテーブルパラメータは自動変換されません。位置情報は常に現在の測定系で解釈されます。変換は外部で行う必要があります。

ソフトウェアバージョン 5 以降では、MD 10260: CONVERT\_SCALING\_SYSTEM=1 を設定することで行えます。外部で変換を行う必要はありません。

参照： 1.8 位置指令単位と検出単位 (G2)

## 例

マシン軸 X1 の補正值入力の例を示します。

%\_N\_AX\_EEC\_INI

CHANDATA (1)

\$AA\_ENC\_COMP[0,0,X1] = 0.0 ; 第 1 補正值

( = interpolation point 0) +0  $\mu$  m

\$AA\_ENC\_COMP[0,1,X1] = 0.01 ; 第 2 補正值

( = interpolation point 1) +10  $\mu$  m

\$AA\_ENC\_COMP[0,2,X1] = 0.012 ; 第 3 補正值

( = interpolation point 2) +12  $\mu$  m

:

\$AA\_ENC\_COMP[0,800,X1] = -0.0 ; 最終補正值

( = interpolation point 800)

\$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 ; -200.0 mm で補正の開始

\$AA\_ENC\_COMP\_MAX[0,X1] = 600.0 ; +600.0 mm で補正終了

\$AA\_ENC\_COMP\_IS\_MODULO[0,X1] = 0; モジュロ機能を持たない補正

M17

この例では、MD 38000: MM\_ENC\_COMP\_MAX\_POINTS  $\geq$  に設定された補正補間点数は 801 でなければなりません、そうでないとアラーム 12400 "Element does not exist" が出力されます。

この例の補正テーブルでは、最低 6.4 KB の不揮発性 NC ユーザメモリが必要です (補正值 1 つに 8 バイト必要)。

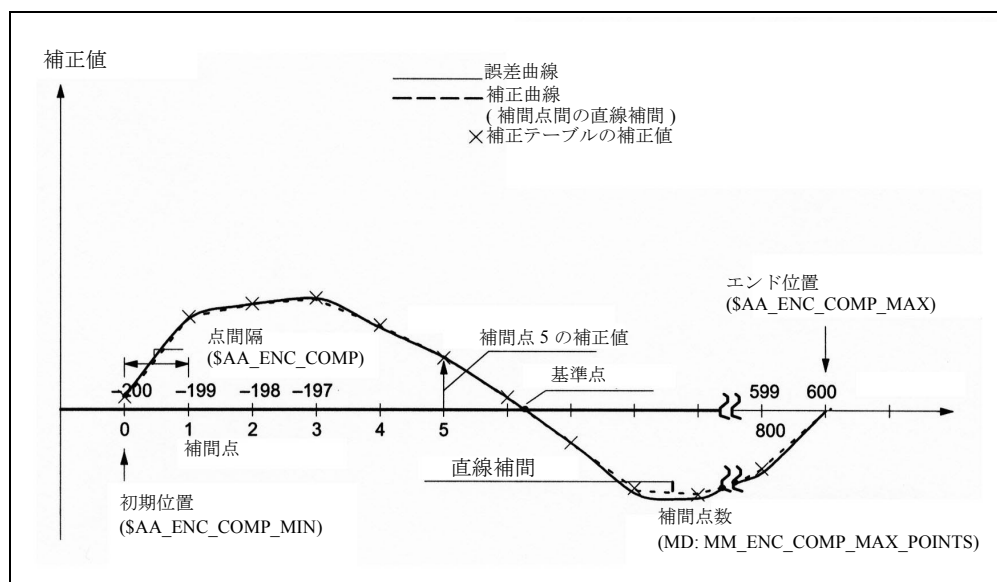


図 2.29 補正テーブルパラメータ (MSEC のシステム変数)

## ■ 直角度補正および角度誤差補正

### 機能

MSEC とは異なり、基本軸と補正軸は、直角度補正および角度誤差補正について同一である必要はありませんが、すべての補正テーブルについて軸割当てが必要です。

自重による軸（基本軸）の垂れを補正するには、別の軸（補正軸）の絶対位置が影響を受けていなければなりません。このため、直角度補正は軸内補正となります。

図 2.30 に示したように、マシンヘッドが負の Y1 方向に移動すればするほど、ビーム軸はますます負の Z1 軸方向に垂れます。

Y1 軸のすべての実際値位置について、Z1 軸についての補正值を含む補正テーブルとして誤差を記録する必要があります。補間点について補正值を入力するだけで十分です。

Y1 軸が移動すると、制御装置は補間点間の直線補間を実行する補間サイクルで Z1 軸の対応する補正值を計算します。この補正は追加指令値として位置制御ループに送られます。補正值が正の場合、対応するマシン軸は負の方向に移動します。

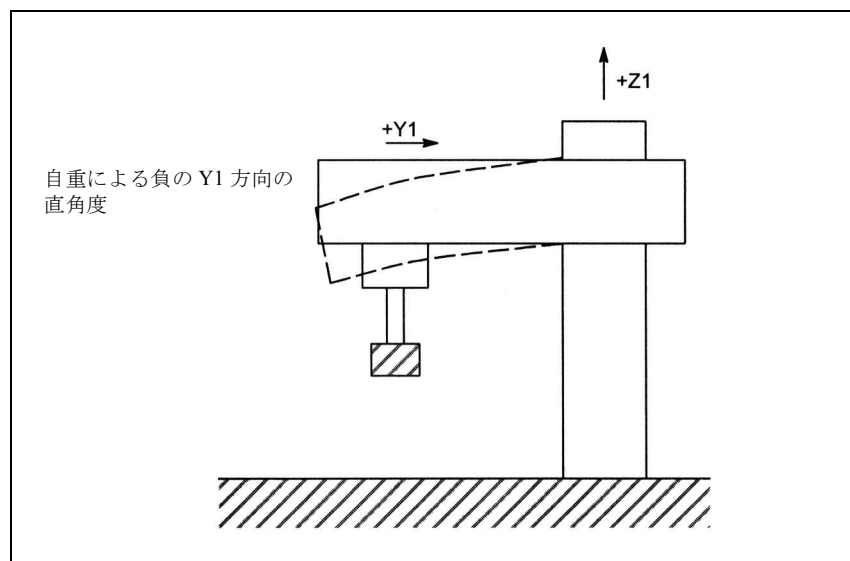


図 2.30 自重による直角度の例

要求事項によっては、いくつかの補正関係を 1 つの軸に定義することができます。総補正值はこの軸の補正值の合計です。

## 設定オプション

次に示すように、直角度補正の補正值を作成するため、および補正值に影響を与えるための様々なオプションが用意されています（図 2.10 参照）。

1. 軸は、複数の補正テーブル（システム変数を使って設定可能）について入力変数（基本軸）として定義できます。
2. 軸は、複数の補正テーブル（システム変数を使って設定可能）について出力変数（補正軸）として定義できます。総補正值はそれぞれの補正值の合計です。

次の定義は許される最大数の補正テーブルに適用されます：

- すべての軸に利用できる最大テーブル数（合計）

2 \* システムの軸の最大数

- 1 つの補正軸に影響与える可能性がある最大テーブル数

1 \* システムの最大軸数

3. 軸はいつでも基本軸と補正軸の両方であることができます。

補正值の計算には常にプログラムされた（要求された）位置指令値が使用されます。

4. 補正の影響の範囲（基本軸の開始位置と終了位置）および補間点間の距離をそれぞれの補正テーブル（システム変数を介して設定可能）について定義します。
5. 補正は方向に依存するようにできます（システム変数を介して設定可能）。
6. それぞれの補正テーブルには周期評価用にモジュロ機能がついています（システム変数を介して設定可能）。
7. テーブル値に掛ける加重係数（パートプログラム、PLC またはユーザが自由に変更できる設定データとして定義可能）をそれぞれの補正テーブルに導入することができます。
8. 補正テーブル同士を掛けることができます（システム変数を介して設定可能）。その積は補正軸の総補正值に加算されます。
9. 補正をアクティブにするには次の方法があります：

- MD 32710: CEC\_ENABLE [AXi] を使って、全補正関係の合計をマシン軸 AX についてイネーブルする。

- SD 41300: \$SN\_CEC\_TABLE\_ENABLE[t] を使って、補正テーブルの評価 [t] をイネーブルする。

例えば、加工要件に合わせて、パートプログラムあるいは PLC ユーザプログラムのいずれからも補正関係を変更することができます（例えば、テーブルの切換え）。

---

10. ソフトウェアバージョン 5 以降の場合、

MD 10260: CONVERT\_SCALING\_SYSTEM=1 をセットすると、軸の MD 32711 : CEC\_SCALING\_SYSTEM\_METRIC が有効になります。この軸について有効なすべてのテーブルの測定系がこのマシンデータにセットされます。これにより、すべての位置エントリは、構成された測定系において、計算された総補正值と共に解釈されます。測定系が切替わっても、位置情報を外部変換する必要はありません。

(注) 基本軸および補正軸が基準化されて初めて、補正テーブルはアクティブになります。

## 監視

直角度補正を適用した場合にもマシン軸の速度や加速率がオーバーすることのないように、総補正值が監視され最大値が制限されます。最大補正值は、軸別に、軸の MD 32720: CEC\_MAX\_SUM にセットされます。

その結果総補正值が最大値より大きくなる場合は、アラーム 20124 "Sum of compensation values too high" が出力されます。プログラム処理は中断されません。追加指令値としての補正值出力は、最大値に制限されます。

総補正值の変更もまた軸別に制限されます。制限値 MD 32730: CEC\_MAX\_VELO を超過すると、アラーム 20125 "Compensation value changed too quickly" が表示されます。プログラム処理は続行します。制限のためにカバーされていなかったパスは、補正值が制限を受けなくなると直ちにカバーされます。



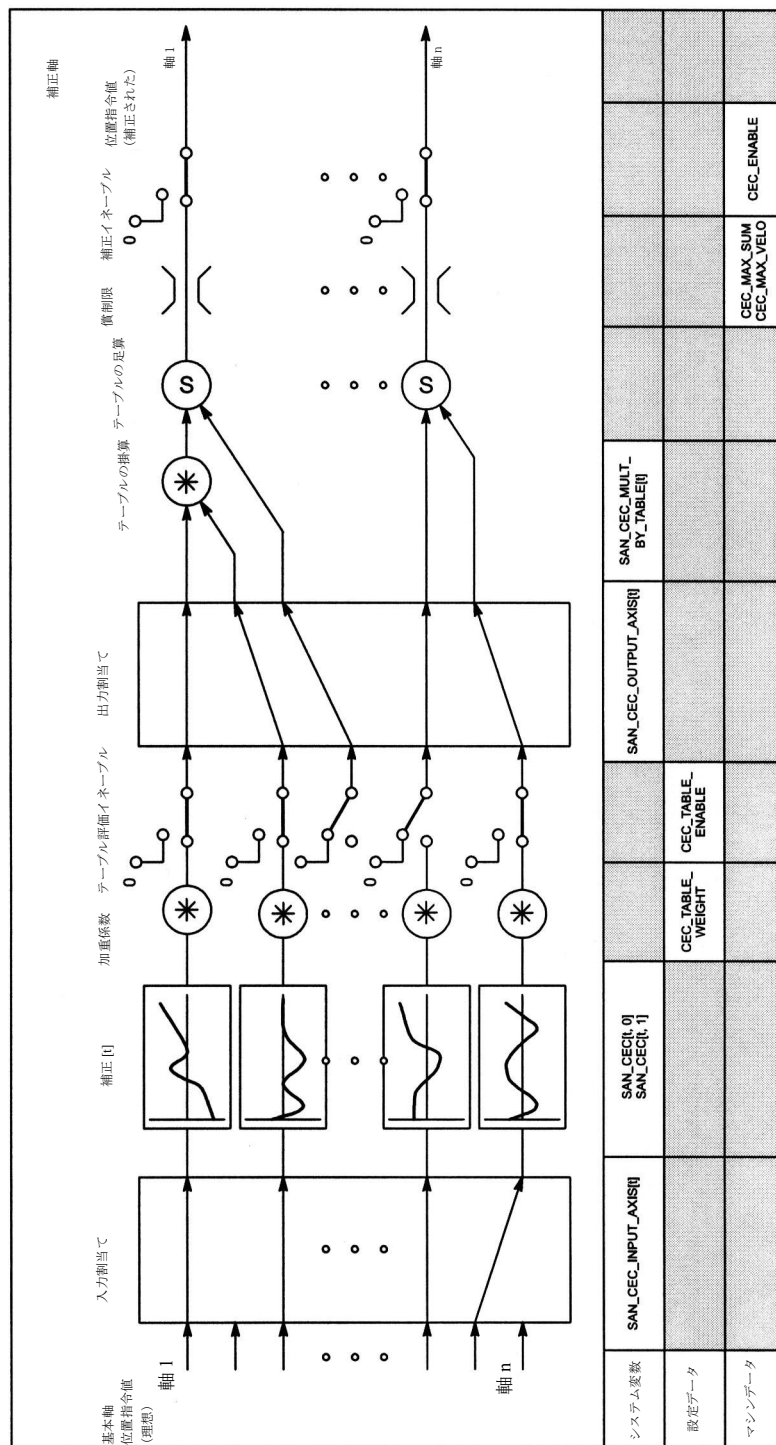


図 2.31 直角度補正の補正値の作成

---

## 複合補正

軸の位置を複数のテーブルに対する入力量（基本軸）として使用し、複数の補正関係（テーブル）から軸の総補正値を導き出し、テーブル同士を掛算することができるので、高度で複雑な直角度補正および角度誤差補正システムを構築することができます。

この機能を使えば、様々な誤差ソースを効率的に処理することができます。例えば、規則的に反復発生する誤差成分用のモジュロ機能を持つテーブルを、同じ軸の不規則的な誤差要素用のモジュロ機能を持たない別のテーブルに組み合わせることができます。

また、この機能を使ってリードスクリュウ誤差を補正することができます。具体的には、基本軸と補正軸について同じ軸をパラメータ化することによってリードスクリュウ誤差を補正できます。しかし、MSECとは異なって、この場合、測定系切換えは自動登録されません。

## 起動

直角度補正機能は、以下の条件が満たされて初めてアクティブになります。

- オプション "Interpolatory compensation"（補間補正）がイネーブルになっていること。
- 関連マシン軸（補正軸）についてこの機能がアクティブになっていること。  
(MD 32710: CEC\_ENABLE [AXi] = 1)
- 補正値が、不揮発性 NC ユーザメモリに保存されていて、アクティブになっていること（電源オン後）。
- 関連補正テーブルの評価がイネーブルになっていること  
(SD 41300: CEC\_TABLE\_ENABLE [t] = 1)
- 基本軸および補正軸の現在の測定系が基準化されていること (IS: "Referenced/Synchronized 1 or 2" DB31, ... DBX60.4 or 60.5 = '1')

これらの条件が満たされ次第、補正軸の指令値位置は基本軸の指令値位置を基準にしてすべてのモードで変更され、マシン軸によって直ちにトラバースされます。エンコーダ周波数が超過したため (IS "Referenced/Synchronized 1 or 2" = '0') などの理由でリファレンスが失われた場合には、補正処理は停止します。

---

## 補正補間点

補正テーブル中の必要となる補間点の数を，それぞれの補正関係について定義しなければならず，必要となるメモリスペースを一般事項 MD 18342:

MM\_CEC\_MAX\_POINTS に予約しなければなりません。

MD 18342: \$MN\_MM\_CEC\_MAX\_POINTS[t]

ここで: [t] = 補正テーブルのインデックス

( $0 \leq t < 2 * \text{軸の最大数}$ )

t = 0: 第 1 補正テーブル

t = 1: 第 2 補正テーブル

$$\text{MM\_CEC\_MAX\_POINTS}[t] = \frac{\text{\$AN\_CEC\_MAX}[t] - \text{\$AN\_CEC\_MIN}[t]}{\text{\$AN\_CEC\_STEP}[t]} + 1$$

## テーブルパラメータ

関連する補正関係についての位置に関連する補正值は、補正テーブル中にシステム変数として保存されます。

次のパラメータをそのテーブルについて定義してください：

- 補正テーブル [t] 中の補間点 N についての補正值 (\$AN\_CEC [t, N])  
補正軸の補正值を、それぞれの補間点ごとにテーブルに入力する必要があります（基本軸の位置）。  
補間点 N は、関連する補正テーブル中に許される補間点数によって限定されます (MD 18342: MM\_CEC\_MAX\_POINTS)。  
N の許容範囲 :  $0 \leq N < \text{MD 18342: MM\_CEC\_MAX\_POINTS}$
- 基本軸 (\$AN\_CEC\_INPUT\_AXIS[t])  
指令値が補正テーブル [t] に対する入力として使用されるマシン軸の名称
- 補正軸 (\$AN\_CEC\_OUTPUT\_AXIS[t])  
補正テーブル [t] の出力が適用されるマシン軸の名称  
  
(注) マルチチャンネルシステムでは、マシン軸およびチャンネル軸の識別子が同一である場合、"general axis identifiers" AX1... をプリセットする必要があります。
- 補間点間の距離 (\$AN\_CEC\_STEP[t])  
補間点間の距離で、補正テーブル [t] の入力値間の距離が決まります。
- 初期位置 (\$AN\_CEC\_MIN[t])  
初期位置は、補正テーブル [t] が開始する基本軸の位置です (= 補間点 0)  
初期位置の補正值は \$AN\_CEC [t, 0] です。  
補間点 0 の補正值は初期位置より小さい位置にすべて使用されます（例外：モジュール機能を持つテーブル）。
- 終了位置 (\$AN\_CEC\_MAX[t])

終了位置は補正テーブル [t] が終了する基本軸の位置です (= 補間点 k)。

終了位置の補正值は \$AN\_CEC [t, k] です。

補間点 k の補正值は終了位置より大きな位置すべてに使用されます（例外：モジュール機能を持つテーブル）。

必要となる補間点数は次のように計算します：

ここで、 $0 \leq k < \text{MD 18342: MM\_CEC\_MAX\_POINTS}$

次の条件が補間点 k に適用されます：

$$k = \frac{\$AN\_CEC\_MAX[t] - \$AN\_CEC\_MIN[t]}{\$AN\_CEC\_STEP[t]}$$

- k = MD 18342 の場合 : MM\_CEC\_MAX\_POINTS - 1

⇒ 補正テーブルが完全に利用されます。

- k < MD 18342: MM\_CEC\_MAX\_POINTS - 1

⇒補正テーブルが完全に利用されていません；入力された k より大きな補正値は無効です。

- k > MD 18342: MM\_CEC\_MAX\_POINTS - 1

⇒終了位置を減少することで制御装置の補正テーブルを制限します；k より大きな補正値は使用しません。

- 方向に依存する補正 (\$AN\_CEC\_DIRECTION[t])

このシステム変数を使うと、補正テーブル [t] を基本軸の両方の移動方向に適用するのか、あるいは正または負のいずれか一方の方向にだけ適用するのかを指定できます。

0: テーブルは基本軸の両方の移動方向に影響を与えます。

1: テーブルは基本軸の正の移動方向にのみ影響を与えます。

-1: テーブルは基本軸の負の移動方向にのみ影響を与えます。

考えられるアプリケーション：2つのテーブルを使って位置に依存するバックラッシュ補正を構築できます、1つは正の方向に影響を与え、もう1つが負の方向への移動に影響を与えます。

- テーブル同士の掛算 (\$AN\_CEC\_MULT\_BY\_TABLE[t])

このオプションを使うと、テーブルの補正値を別のテーブル（または同じテーブル）の補正値に掛けることができます。この積は補正テーブルの総補正値に加算されます。

構文: \$AN\_CEC\_MULT\_BY\_TABLE[t<sub>1</sub>] = t<sub>2</sub>

t<sub>1</sub> = 補正軸のテーブル 1 のインデックス

t<sub>2</sub> = 補正軸のテーブル 2 の番号

同じテーブルの番号とインデックスが同じであってはなりません。

通常はテーブルの番号 = テーブルのインデックス + 1 とします。

- モジュロ機能を使用した補正 (\$AN\_CEC\_IS\_MODULO[t])

モジュロ機能を使用した補正がアクティブになっている場合、補正テーブルは周期的に繰返されます。つまり、ロケーション \$AN\_CEC\_MAX[t] (補間点 \$AN\_CEC[t,k]) の補正値の直後にロケーション \$AN\_CEC\_MIN[t] (補間点 \$AN\_CEC[t,0]) の補正値が続きます。

この2つの補正値は同一であるべきです。そうでないと補正値が変異点で MAX から MIN にあるいはその逆にジャンプします。

\$AN\_CEC\_IS\_MODULO[t] = 0: モジュロ機能を持たない補正

\$AN\_CEC\_IS\_MODULO[t] = 1: モジュロ機能を持つ補正

基本軸としてモジュロ回転軸を使用してモジュロ補正が実施される場合、使用される補正テーブルもモジュロ計算される必要があります。

例:

MD 30300: IS\_ROT\_AX[AX1] = 1: 回転軸

MD 30310: 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\_IS\_MODULO[0]=1

(注) ソフトウェアバージョン 4 かそれ以前では、位置情報を含む  
テーブルパラメータは測定系切換え (MD 10240:  
SCALING\_SYSTEM\_IS\_METRIC の変更) 時に自動変換されま  
せん。位置情報は常に現在の測定系で解釈されます。変換は  
外部で行う必要があります。  
ソフトウェアバージョン 5 以降では、MD 10260:  
CONVERT\_SCALING\_SYSTEM=1 をセットすることで MD  
32711: CEC\_SCALING\_SYSTEM を介して測定系を構成するこ  
とができます。測定系を切換えても位置情報を外部で変換す  
る必要はありません。

参照： 1.8 位置指令単位と検出単位 (G2)

## テーブルの例

以下に、Y1 軸の直角度補正用の補正テーブルの例を示します。Y1 軸の位置に  
よっては補正值が Z1 軸に適用されます。この目的で第 1 補正テーブル (t=0) が使  
用されます。

%\_N\_NC\_CEC\_INI

CHANDATA (1)

\$AN_CEC [0,0]	= 0	; 第 1 補正值 ( = 補間点 0 ) (Z1: $\pm 0 \mu\text{m}$ 用)
\$AN_CEC [0,1]	= 0.01	; 第 2 補正值 ( = 補間点 1 ) (Z1: $+10 \mu\text{m}$ 用)
\$AN_CEC [0,2]	= 0.012	; 第 3 補正值 ( = 補間点 2 ) (Z1: $+12 \mu\text{m}$ 用)
:		
\$AN_CEC [0,100]	= 0	; 最終補正值 ( = 補間点 101 ) (Z1: $\pm 0 \mu\text{m}$ 用)
\$AN_CEC_INPUT_AXIS[0]	= (AX2)	; 基本軸 Y1
\$AN_CEC_OUTPUT_AXIS[0]	= (AX3)	; 補正軸 Z1
\$AN_CEC_STEP[0]	= 8	; 補間点間の距離 8.0 mm
\$AN_CEC_MIN[0]	= -400.0	; -400 mm で補正が開始
\$AN_CEC_MAX[0]	= 400.0	; Y1 = +400 mm で補正が終了
\$AN_CEC_DIRECTION[0]	= 0	; テーブルは Y1 の両方の移動方向 に適用される
\$AN_CEC_MULT_BY_TABLE[0]	=	;
\$AN_CEC_IS_MODULO[0]	= 0	; モジユロ機能を持たない補正

M17

この例では、MD 18342: MM\_CEC\_MAX\_POINTS [0] に設定された補正補間点の数  
は  $\geq 101$  でなければならず、それ以外の場合はアラーム 12400 が出力されます。

この例の補正テーブルは、最低 808 バイトの不揮発性 NC ユーザメモリを必要とし

ます。

## テーブル同士の掛算

テーブル同士の掛算機能を使うと、テーブルを他の（あるいは同じ）テーブルに掛けることができます。掛算リンクは、前述のシステム変数を使って設定します。以下に、マシン基礎の直角度補正の例を使って、テーブル同士の掛算アプリケーションを説明します。

大型マシンでは、マシン基礎の直角度がマシン全体の傾きの原因となります。例えば、図 2.33 に示したボーリング盤では、X1 軸の位置（これにより傾斜角  $\beta$  が決定されます）とドリルの高さ（すなわち Z1 軸の位置）の両方を基にして X1 軸の補正を設定しています。

補正を適用するには、X1 軸と Z1 軸の補正值が次のように掛算されなければなりません（図 2.32 を参照）。

$$\Delta X1 = Z1 * \sin \beta (X1) \quad Z1 * \beta (X1) \\ \approx$$

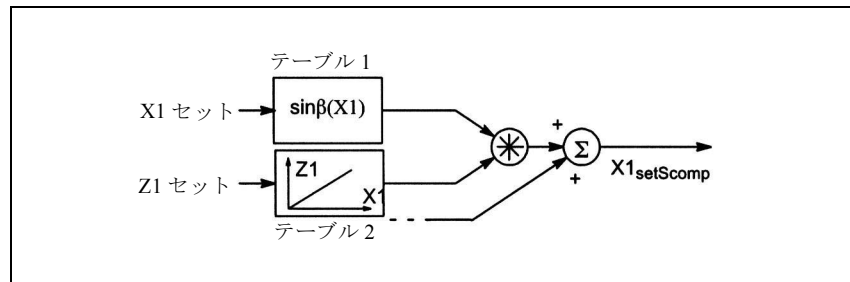


図 2.32 テーブル同士の掛算

補正テーブル 1（テーブルインデックス = 0）が、軸 X1 に対する軸 X1 のリアクションを記述しています（位置に依存する傾斜角  $\beta (X1)$  の符号）。

補正テーブル 2（テーブルインデックス = 1）が、軸 X1 に対する軸 Z1 のリアクションを記述しています（線形）。

テーブル 1（インデックス = 0）では、テーブル 1 とテーブル 2 との積を選択します：

$$\$AN\_CEC\_MULT\_BY\_TABLE[0] = 2$$

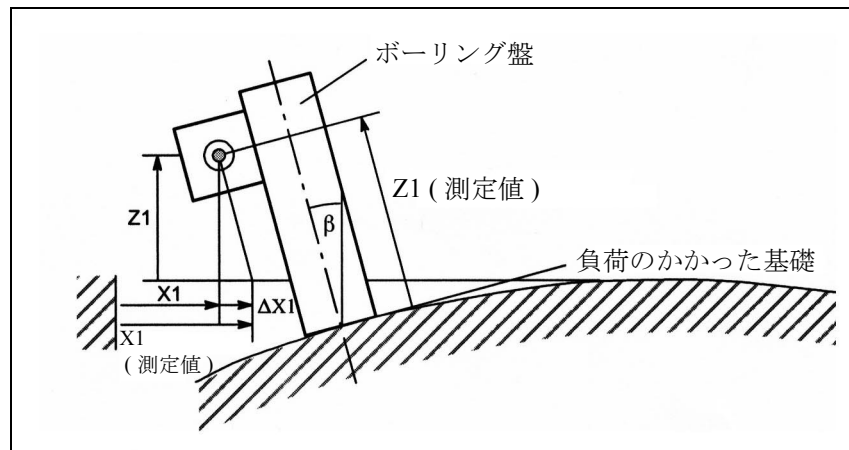


図 2.33 マシン基礎の直角度補正



## ■ 補間補正の特別機能

### 測定

"Measurement"（測定）機能により、マシンのオペレータやプログラマが必要とする補正済みの機械位置（理想マシン）が得られます。

### ティーチイン

"TEACH IN"（ティーチイン）機能も、保存する機械位置を求めるのに、補正された位置を使用します。

### ソフトウェアリミットスイッチ

理想位置（つまり MSEC およびバックラッシュ補正機能によって修正された位置）もソフトウェアリミットスイッチによって監視されます。

### 位置表示

機械座標系の機械位置画面は、軸（理想マシン）の理想の（プログラムされている）機械位置を表示します。

送り軸／主軸サービス画面（オペレーティングエリア診断）には、測定系によって測定された位置ならびにバックラッシュ補正とリードスクリュウ誤差補正の合計が表示されます（＝機械位置測定系 1/2）。

### 補正值表示

次の補正值もまた、「軸」サービス画面（診断オペレーティングエリア）に表示されます：

軸サービス画面	意味
絶対補正值測定系 1 または 2	表示値は、軸の現在の機械位置（測定系 1 または 2）についての "MSEC" および "バックラッシュ補正" から算出した補正值の合計に対応しています。
直角度／温度の補正值	表示値は、軸の現在の機械位置の「直角度補正」および「温度補正」から算出した補正值の合計です。

参照： 1.5 診断機能（D1）

### 基準点喪失

基本軸の基準点が失われた場合 (IS: "Referenced/Synchronized 1 or 2" DB31, ... DBX60.4 or 60.5 = '0'), 影響を受けた軸の MSEC またはバックラッシュ補正機能が停止します。基準点に達すると、これらの補正は再び自動的に起動します。

---

## 補正テーブルのアクセス保護

補正テーブルに対するアクセス保護は現在のところありません。

## コントローライネーブル

補正関係の結果、基本軸の移動動作により補正軸も移動してしまう場合があります、これらの軸についてコントローライネーブル信号をセットする必要があります (PLC ユーザプログラム)。セットしないと補正の効果が限定されます。

## PLC に対する移動コマンド

補正軸が基本軸の移動動作の結果としてのみ動作するように設定されている場合、インターフェース信号 "Travel command +" または "Travel command -" は補正されません。

## ■ 追従誤差補正（フィードフォワード制御）

### ■ 一般事項

#### 軸追従誤差

フィードフォワード制御を使うと、軸追従誤差は、ほぼゼロに低減することができます。このため、フィードフォワード制御は、"following error compensation"（追従誤差補正）とも呼ばれます。

追従誤差によって、特に輪郭曲線、例えば円弧やコーナーでの加速中に、速度に依存する輪郭誤差が発生します。

#### フィードフォワード制御方法

追従フィードフォワード制御方法を使用して、「追従誤差補正」を行うことができます。

- 速度フィードフォワード制御
- トルクフィードフォワード制御（オプション）  
MD 32620: FFW\_MODE（フィードフォワード制御方法）で選択します。

#### パートプログラムでの起動／停止

以下のパートプログラム的高级言語を使って、フィードフォワード制御を起動したり停止したりできます。

FFWON    フィードフォワード制御 ON

FFWOF    フィードフォワード制御 OFF

デフォルトの設定（すなわち、リセット後の M30）は、チャンネル別の MD 20150: GCODE\_RESET\_VALUES（G グループの初期設定）で入力します。

MD 32630: FFW\_ACTIVATION\_MODE は、フィードフォワード制御を FFWON でオンにし、FFWOF でオフにするかどうかを各軸に対して設定します。

FFWON や FFWOF を使用すると、（MD 32620: FFW\_MODE = 1 または 2 だけでなく）MD 32630: FFW\_ACTIVATION\_MODE = 1 と設定されているチャンネルのすべての送り軸／主軸のフィードフォワード制御を個別に起動したり停止することができます。

したがって、MD 32630: FFW\_ACTIVATION\_MODE は、互いに補間し合う軸に対しては同一設定となります。

ジャーク防止のため、フィードフォワード制御のオンやオフは、送り軸／主軸が停止しているときにしか行わないでください。プログラマが責任を持って行ってください。

#### 条件

フィードフォワード制御を適用する前に、下記のことを確認してください。

- マシン動作が正確であること
- マシンの動的応答についての正しい知識があること
- 位置指令値および速度指令値に急激な変化がないこと

## 制御ループの最適化

フィードフォワード制御は、送り軸／主軸別に設定されます。まず最初に、送り軸／主軸に対して電流制御ループ、速度制御ループ、位置制御ループを最適に設定しなければなりません。

参照： YS840DI API 取扱説明書

## 起動

まず MD 32620: FFW\_MODE を設定し、目的のフィードフォワード制御モードを選択する必要があります。

0 = フィードフォワード制御なし

1 = 速度フィードフォワード制御（初期設定）

2 = トルクフィードフォワード制御

トルクフィードフォワード制御を選択する前に、選択項目を有効にしてください。

## パラメータの割当て

フィードフォワード制御パラメータを、関連する送り軸／主軸に割当て、それからマシンデータに入力してください。

### ■ 速度フィードフォワード制御

速度フィードフォワード制御では、速度コントローラへの入力として、速度指令値も注入されます（図 2.34 を参照）。この追加の指令値は、通常、約 "1" に等しい係数で加重することができます。

速度フィードフォワード制御を正しく設定するために、速度制御ループの等価時定数を正確に求め、マシンデータとして入力する必要があります。

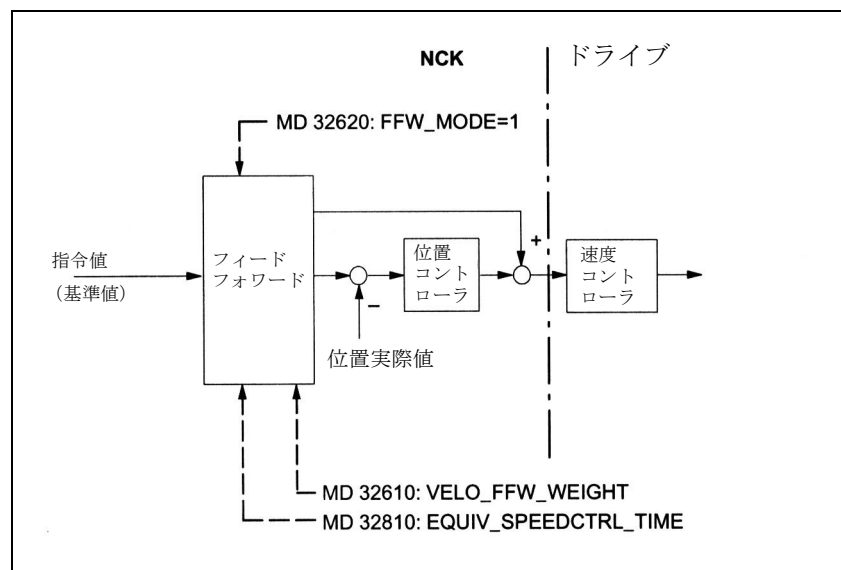


図 2.34 速度フィードフォワード制御

## パラメータ

インストール時に、速度フィードフォワード制御に対して、次のような軸別パラメータを設定する必要があります。

- MD 32610: VELO\_FFW\_WEIGHT  
フィードフォワード制御係数
- MD 32810: EQUIV\_SPEEDCTRL\_TIME  
閉じた速度制御ループの等価時定数

## 速度フィードフォワード制御用のパラメータ

### MD 32810: EQUIV\_SPEEDCTRL\_TIME

閉じた速度制御ループの等価時定数

閉じた速度制御ループの等価時定数は、速度制御ループのステップ応答を測定して求めます。インストールツールを使用して決定プロセスを表示することができます。

参照： YS840DI API 取扱説明書

速度制御ループの等価時定数は、位置制御サイクル（= 基本系サイクル × 位置制御サイクルに対する係数）に、速度指令値フィルタ（ドライブマシンデータ 1500 ~ 1521）を加算して求めることもできます。

### MD 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

送り軸／主軸に対する制御ループの設定が最適で、かつ等価時定数が正確に求められた場合、フィードフォワード制御係数は、約 1 になります。よって、マシンデータに入力される初期値は 1（= 標準初期設定）となります。

この値の場合、速度が一定であれば、追従誤差はほぼゼロになります（つまり、制御のずれがなくなります）。位置決め動作を行い、その結果サービス画面に表示される実際の制御のずれによって、チェックすることができます。

参照： 1.5 診断機能（D1）

## 微調整

MD 32610:VELO\_FFW\_WEIGHT および MD 32810:EQUIV\_SPEEDCTRL\_TIME で設定した値を微調整することにより、関連する送り軸／主軸に適した応答を設定することができます。

送り軸／主軸を一定の速度で移動し、マシンデータに加えた変更による影響をサービス画面 "Control deviation" でチェックするだけです。速度フィードフォワード制御の調整基準は、"Control deviation" = 0 です。

事例 1: 軸が正の方向に移動しているとき、"Control deviation" に正の値が表示されます。

⇒速度制御ループの等価時定数またはフィードフォワード係数が小さすぎます。

事例 2: 軸が正の方向に移動しているとき、"Control deviation" に負の値が表示されます。

⇒速度制御ループの等価時定数またはフィードフォワード係数が大きすぎます。

サービス画面上の数値を読み取りやすくするため、加速を小さくし、フィードレートを大きくしてください。こうすることにより、加速時間が非常に長くなり、制御のずれが読取りやすくなります。

X 軸の例：

```
MD 32300: MAX_AX_ACCEL = 0,1 ; m/s2
MD 32000: MAX_AX_VELO = 20000,0 ; mm/min
; 等価時定数設定用のパートプログラム
G1 F20000
FFWON
LOOP:
X1000
X0
GOTOB LOOP
M30
```

軸 1，軸 2，軸 3 のアクティブ速度フィードフォワード制御の例  
速度制御ループの等価時定数

(MD 32810: EQUIV\_SPEEDCTRL\_TIME) の内容：

- 軸 1 : 2 ms
- 軸 2 : 4 ms (動きが一番遅い軸)
- 軸 3 : 1 ms

動的応答適応の時定数に対する値

(MD 32910: DYN\_MATCH\_TIME) の内容：

- 軸 1 : 2 ms
- 軸 2 : 0 ms
- 軸 3 : 3 ms

参照： 1.8 位置指令単位と検出単位 (G2)

---

## 速度指令値のリードタイム

速度指令値のドライブへの転送は、マシンデータ MD 10082 および MD 10083 でセットできます。

(注) デジタル 611D ドライブを使用しないと、速度指令値出力用のリードタイムは調整できません。

速度指令値出力用のリードタイムは、MD 10082: CTRLOUT\_LEAD\_TIME で設定します。入力する値が大きいほど、ドライブが速度指令値を転送するまでの時間が短くなります。

つまり、下記のようになります。

- 0 % : 指令値は、次の位置制御サイクルの初めに転送されます。
- 50 % : 現在の位置制御サイクルを半分実行した時点で、指令値が転送されます。

適切なリードタイムは、最大位置制御計算時間を測定しないと決定できません。

MD 10083: CTRLOUT\_LEAD\_TIME\_MAX では、制御によって測定された値が示されます。この値は正味の値なので、例えば 5 % 程度の安全余裕を設けたほうがよいでしょう。

(注) 入力されたリードタイムの値が高すぎると、ドライブアラーム 300506 が出力されます。

入力された値は、ドライブで、次に低い速度コントローラパルスレートに丸められます。各ドライブの速度コントローラパルスレートの設定値が異なる場合、設定値を変更しても、設定されているドライブすべてに対して同程度のコントローラの改善が行われるとは限りません。

## ■ トルクフィードフォワード制御

トルクフィードフォワード制御を使用すると、トルクに比例する、追加電流指令値を電流コントローラの入力に直接注入できます（図 2.35 を参照）。この値は、加速および慣性モーメントを使用して設定します。

トルクフィードフォワード制御を正確に設定するためには、等価時定数を正確に求めてマシンデータに入力する必要があります。

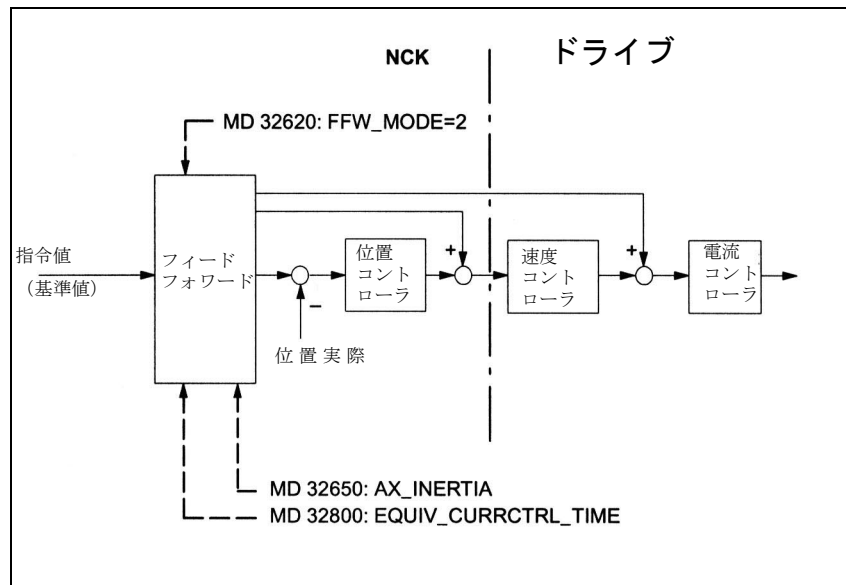


図 2.35 トルクフィードフォワード制御

### アプリケーション

力学上の要求が厳しい場合、輪郭精度を上げるため、トルクフィードフォワード制御が必要となります。正しく設定すれば、加速度が大きい場合でも、追従誤差をほぼ完全に補正することができます。

### パラメータ

トルクフィードフォワード制御を行うためには、インストール時に次の軸別パラメータを設定する必要があります。

- MD 32650: AX\_INERTIA  
トルクフィードフォワード制御用の、軸の慣性モーメント（ドライブの観点から見た）
- MD 32800: EQUIV\_CURRCTRL\_TIME  
電流制御ループの等価時定数
- ドライブ - MD 1004: CTRL\_CONFIG  
構成  
セットビット 0 = "1"（トルクフィードフォワード制御アクティブ）

## トルクフィードフォワード制御用のパラメータ



---

ドライブ MD 1004: CTRL\_CONFIG 構成

トルクフィードフォワード制御は、ビット 0 = "1" になっている ドライブ でアクティブになります。

MD 32800: EQUIV\_CURRCTRL\_TIME

閉じた電流制御ループの等価時定数

閉じた電流制御ループの等価時定数は、電流制御ループのステップ応答を測定して求めます。インストールツールを使用して決定プロセスを表示することができます。

さらに、モジュールの DA 変換器の各モジュールの第 1 ドライブの電流指令値が出力されるので、オシロスコープで確認できます。

参照： YS840DI API 取扱説明書

等価時定数は、できるだけ正確に求める必要があります。

MD 32650: AX\_INERTIA

軸の総慣性モーメント

軸の総慣性モーメント（ドライブの慣性モーメント + モータシャフトに関する負荷）を求めて、トルクフィードフォワード制御用のマシンデータに入力する必要があります。

ドライブ -MD 1117: MOTOR\_INERTIA（モータの慣性モーメント）を 1 ～ 2 倍したものが、MD 32650: AX\_INERTIA の推奨初期設定値です。

微調整

MD 32800: EQUIV\_CURRCTRL\_TIME および MD 32650: AX\_INERTIA で設定された数値を微調整することにより、関連する送り軸／主軸に適した応答を設定することができます。

加速が非常に速いので、サービス画面でパラメータを微調整することはできません。マシンデータに加えられた変更は、アナログ指令値出力の追従誤差を記録することによりチェックします（インストールツールが必要です）。

送り軸／主軸の加速中も、一定の移動に対する追従誤差を確認する必要があります。トルクフィードフォワード制御の調整基準は、追従誤差 0 です。

事例 1: 軸が正の方向に移動しているとき、追従誤差が正の値になる。

⇒電流制御ループの等価時定数または軸の慣性モーメントに対して入力された値が小さすぎます。

事例 2: 軸が正の方向に移動しているとき、追従誤差が負の値になる。

⇒電流制御ループの等価時定数または軸の慣性モーメントに対して入力された値が大きすぎます

---

## 補間軸の設定

補間軸についても、各軸に最適なフィードフォワード制御パラメータを設定する必要があります。各軸毎に、異なるフィードフォワード制御パラメータを設定することができます。

## 輪郭監視の確認

2つの等価時定数（MD 32810: EQUIV\_SPEEDCTRL\_TIME および MD 32800: EQUIV\_CURRCTRL\_TIME）は輪郭監視に影響を与えるので、この後すぐ引き続き確認してください。

参照： 1.2 送り軸監視機能及びプロテクションゾーン（A3）

## サーボゲイン係数に対する影響

フィードフォワード制御が正しく設定されていれば、速度フィードフォワード制御下のシステムにおける指令値の変化に対する応答は、速度制御ループの指令値の変化に対する応答と同じくらい動的になります。またトルクフィードフォワード制御下では、電流制御ループの指令値の変化に対する応答と同じくらい動的になります。つまり、MD 32200: POS\_CTRLGAIN で設定されたサーボゲイン係数は、（コーナの誤差、オーバシュート、円や半径の誤差などの）指令値の変化に対する応答にほとんど影響を与えません。

フィードフォワード制御は外乱特性（同期性）に影響を与えません。この場合、MD 32200: POS\_CTRLGAIN で設定された係数がアクティブ係数となります。

## サービス画面 "Servo gain factor"

フィードフォワード制御がアクティブであると、サービス画面に表示される（指令値の変化に対する応答に適用されるサーボゲイン係数に対応する）軸のサーボゲインは非常に高くなります。

## 動的応答調整

互いに補間し合うが、軸制御ループ応答時間が異なる軸に対して、動的応答調整を行なってすべての軸の時間応答を同じにし、制御の質を損なわずに、最適な輪郭精度を保証することができます。

フィードフォワード制御がアクティブになっている場合、関連する軸に対する最低速度または電流制御ループの等価時定数の差を、動的応答適用の時定数として入力する必要があります (MD 32910: DYN\_MATCH\_TIME)。

## ■ 摩擦補正（象現突起補正）

### ■ 一般事項

#### 機能

摩擦は主にギアおよびガイドウェイで発生します。マシン軸での静摩擦は特に注意する必要があります。定常運転時よりも加速時の方がはるかに大きな力が必要なため、加速時により大きな誤差が生じます。

静摩擦が摩擦力にジャンプを引き起こす方向変換でも同様の現象が生じます。例えば、軸が負から正の速度に変化する場合、摩擦状態が変化するために速度がゼロを通過するわずかな時間軸は停止します。補間軸では、摩擦状態の変化によって輪郭誤差が生じることがあります。

#### 象現突起

ある軸がクォッドラント遷移で停止しているときに別の軸が最大パス速度で移動する場合、輪郭に誤差（象現突起）がよく発生します。摩擦補正を行なうことによって、象現突起をほぼ完全に取り除くことができます。

#### 原理

静摩擦によって生じた輪郭誤差が正しい符号と振幅を持つ追加指令値パルスを入力することで効率的に補正されることがマシン測定で確認されています。

#### 摩擦補正方法

YS840DI (MD 32490: FRICT\_COMP\_MODE "Type of friction compensation") では、2種類の摩擦補正方法のうちから1つを選択できます。

- 従来の摩擦補正 (MD 32490: FRICT\_COMP\_MODE = 1)

これは、加速の機能としての特性に基づき補正パルスの度合いを設定するものです。この特性は、真円度試験を行なってスタートアップ時に確定しパラメータ化できます。この手順はかなり複雑で、経験を必要とします。

- ニューラルネットワークを使つての象現突起補正

( オプション ) (MD 32490: \$MA\_FRICT\_COMP\_MODE = 2)

スタートアップを簡略化するために、スタートアップエンジニアが補正特性を手作業で入力しなくてもよいようになりました。補正特性はトレーニング時に自動的に計算されて不揮発性ユーザメモリに保存されます。

ニューラルネットワークは、はるかに優れた質と精度を持つ補正曲線を生出すことができます。

また、この機能を使うと、再最適化をマシンに直接か、簡単に行えます。

## 真円試験

摩擦補正機能（従来の補正とニューラル補正）は、真円試験で非常に簡単にセットアップすることができます。円の輪郭をたどり、機械位置を測定し、プログラムされている半径とのずれ（特にクォドラント遷移ポイントにおいて）を図に表示します。この測定結果は受動ファイルシステムに保存されている "Trace" を使って記録されます。

この真円試験は、"Installation tool" 機能です。MMC101 や MMC102/103 では、診断エリアで直接この機能を選択できます。

真円試験の詳細については、2.5.2 「■真円試験」を参照してください。

## ■ 従来の摩擦補正

### 摩擦補正方法

従来の摩擦補正を選択する場合は、MD 32490: FRICT\_COMP\_MODE に値 1 を入力します（摩擦補正タイプ）。

### 振幅調整

多くの場合、注入した摩擦補正值の振幅は、加速範囲全体で常に一定になるとは限りません。例えば、高加速での最適化補正の場合、低加速の場合よりも小さな補正值を注入する必要があります。このため振幅調整付きの摩擦補正をアクティブにすることで精度要求に答えることができます（図 2.36 参照）。この機能は、MD 32510: FRICT\_COMP\_ADAPT\_ENABLE = 1（摩擦補正適応アクティブ）で軸別にアクティブにします。

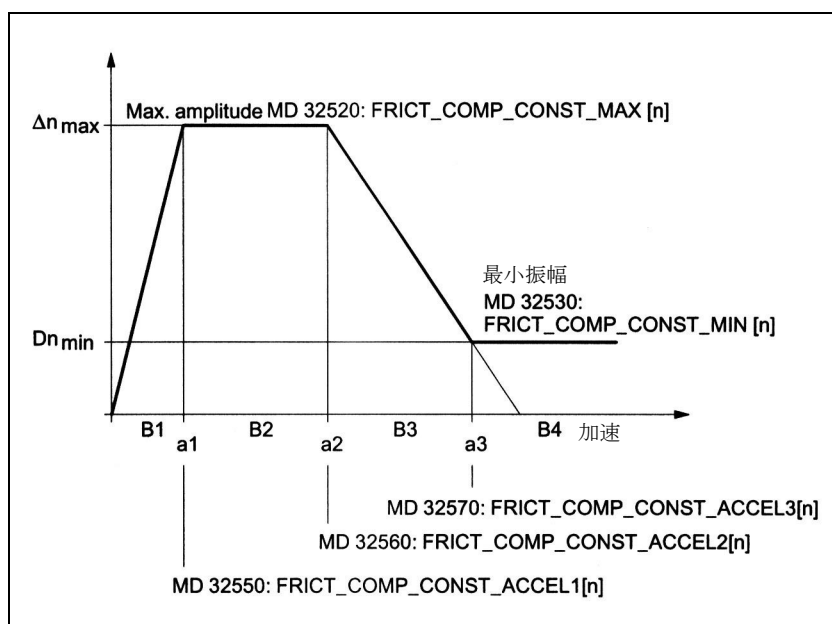


図 2.36 振幅調整付き摩擦補正の一般的な曲線

適応特性は 4 つに分類することができます（異なる注入振幅  $\Delta n$  を各範囲に適用します）：

B1:	$a < a_1$ の場合	$\Delta n = \Delta n_{\max} * a / a_1$
B2:	$a_1 \leq a \leq a_2$ の場合	$\Delta n = \Delta n_{\max}$
B3:	$a_2 < a < a_3$ の場合	$\Delta n = \Delta n_{\max} * (1 - (a - a_2) / (a_3 - a_2))$
B4:	$a \geq a_3$ の場合	$\Delta n = \Delta n_{\min}$

## 特性のパラメータ

図 2.36 に示した適応特性のパラメータは、マシンデータとして指定の軸に入力する必要があります。

$\Delta n$  = 摩擦補正值の注入振幅

$\Delta n_{\max}$  = 最大摩擦補正值

MD 32520: FRICT\_COMP\_CONST\_MAX [n]

$\Delta n_{\min}$  = 最小摩擦補正值

MD 32530: FRICT\_COMP\_CONST\_MIN [n]

$a_1$  = 摩擦補正に対する適応加速値 1

MD 32550: FRICT\_COMP\_ACCEL1 [n]

$a_2$  = 摩擦補正に対する適応加速値 2

MD 32560: FRICT\_COMP\_ACCEL2 [n]

$a_3$  = 摩擦補正に対する適応加速値 3

MD 32570: FRICT\_COMP\_ACCEL3 [n]

## 特性の形についての注記

特殊な場合においては、計算された特性が図 2.36 に示した一般的な形とは異なることがあります。

場合によっては、 $\Delta n_{\min}$  (MD 32530: FRICT\_COMP\_CONST\_MIN) の値であっても  $\Delta n_{\max}$  (MD 32520: FRICT\_COMP\_CONST\_MAX) より大きくなることもあります。

## ■ 従来の摩擦補正のスタートアップ

### 真円試験

摩擦補正機能は真円試験を使って簡単に起動することができます。

### ステップバイステップのスタートアップ

まず従来の摩擦補正機能を選択する必要があります。

(MD 32490: FRICT\_COMP\_MODE=1)

摩擦補正值は主にマシンの構成に左右されます。

- ステップ 1：適応特性のない摩擦補正值の計算
- ステップ 2：適応特性の計算（摩擦補正が加速に依存する場合、ステップ 1 の結果は使用できません）。

## インストールステップ 1：適応特性のない摩擦補正

### 1. 摩擦補正を使用しない真円試験

最初に摩擦補正を使用しない真円試験 (MD 32500: FRICT\_COMP\_ENABLE = 0) を実行する必要があります。

2.5.2 「■真円試験」に真円試験の手順の説明があります。

摩擦補正を使用しないで取得されたクォッドラント遷移の一般的な結果を図 2.37 に示します。

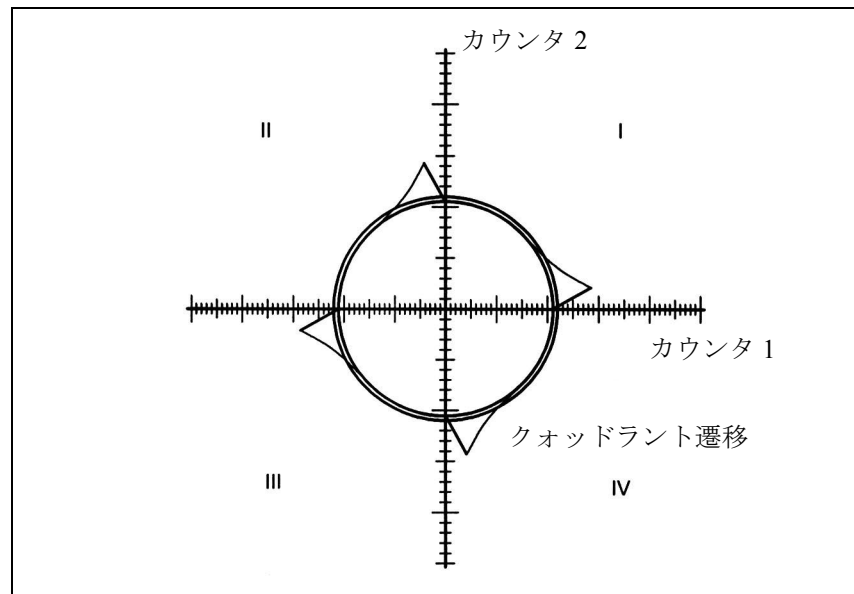


図 2.37 クォッドラント遷移時の未補正半径ずれ

## 2. 摩擦補正を有効にする

次に、問題の送り軸／主軸に対して摩擦補正を有効にしてください。

摩擦補正を有効にする

⇒ MD 32500: FRICT\_COMP\_ENABLE[n] = 1

## 3. 適応を停止する

適応を使用せずに摩擦補正を起動する場合は適応を停止する必要があります。

適応を停止する

⇒ MD 32510: FRICT\_COMP\_ADAPT\_ENABLE[n] = 0

## 4. 補正パラメータの確定

適応特性を持たない摩擦補正を、次のパラメータを使って定義します。

1. MD 32520: FRICT\_COMP\_CONST\_MAX [n]

[mm/min] で表した摩擦補正值（振幅）

2. MD 32540: FRICT\_COMP\_CONST\_TIME [n]

[s] で表した摩擦補正時定数

これら 2 つのパラメータは、クォッドラント遷移時に、プログラムされている半径からのずれが真円試験で最小となるかあるいはゼロになるまで変更されつづけます（図 2.38 ～ 2.41 を参照）。

試験を行う際には、さまざまな半径と速度を使ってください（マシンのアプリケーションの一般値）。

### スタート値

比較的小さな注入振幅に加えて、複数のコントローラサイクルの時定数を、測定開始時にスタート値として入力する必要があります。

例：

MD 32520: FRICT\_COMP\_CONST\_MAX [n] = 10 (mm/min)

MD 32540: FRICT\_COMP\_CONST\_TIME [n] = 0.008 (8 ms)

パラメータ変更の効果は、実際に測定して円をプロットすることによってチェックしてください。

### 平均化

様々な半径や軸について共通した一つの補正時定数を決めることができない場合は、計算された時定数を平均する必要があります。

### 正しく設定された摩擦補正

摩擦補正機能の設定が正しく行われたならば、クォッドラント遷移はほとんどゼロになります（図 2.38 を参照）。

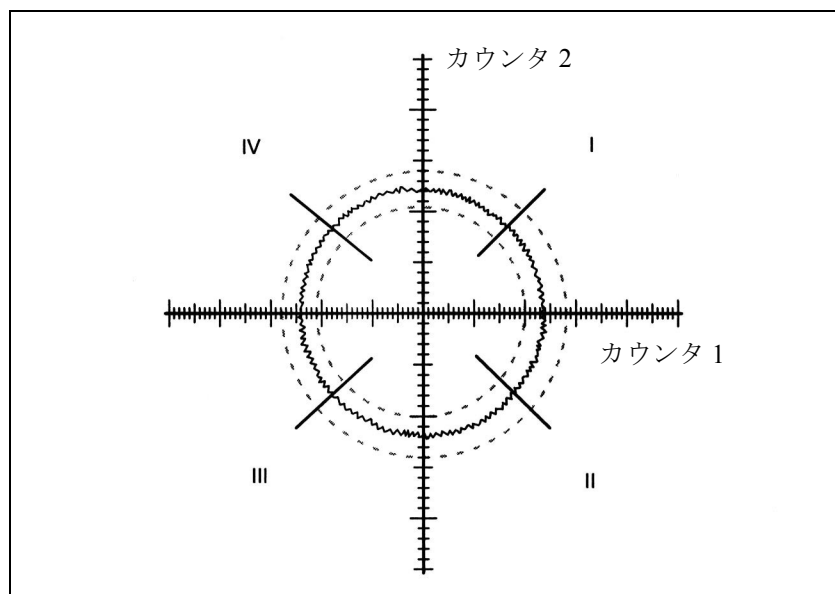


図 2.38 摩擦補正が正しく設定された場合のクォッドラント遷移

### 小さすぎる振幅

クォッドラント遷移時にプログラム半径からの半径ずれが十分に補正できていないことが真円試験で判明した場合には、注入されている振幅が小さすぎます（図 2.39 を参照）。

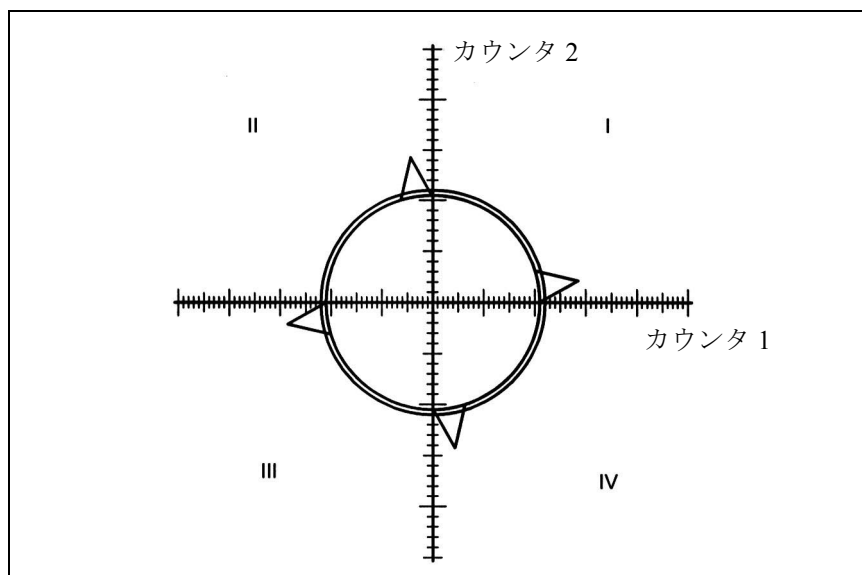


図 2.39 小さすぎる振幅



## 大きすぎる振幅

クォッドラント遷移時にプログラム半径からの半径ずれが過補正されていることが真円試験で判明した場合には、注入されている振幅が大きすぎます（図 2.40 を参照）。

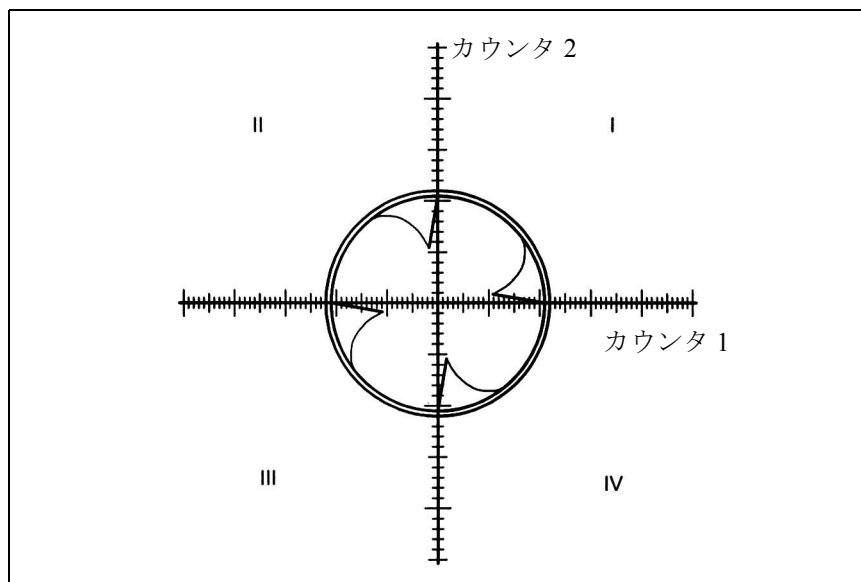


図 2.40 大きすぎる振幅

## 小さすぎる時定数

クォッドラント遷移時にプログラムされた半径からの半径ずれが一時的に補正されているものの、遷移点を過ぎると半径ずれが大きくなっていることが真円試験で判明した場合には、時定数設定が小さすぎます（図 2.41 を参照）。

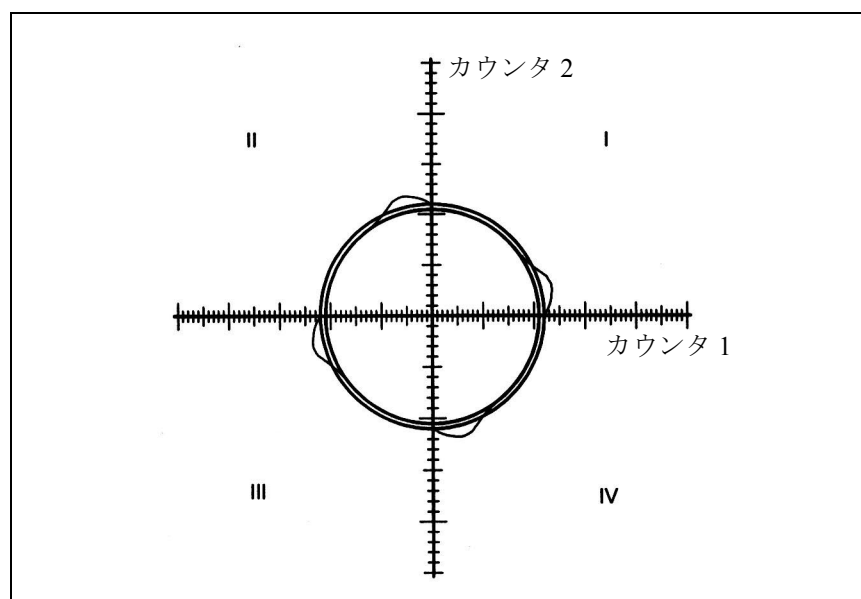


図 2.41 小さすぎる補正時定数

## 大きすぎる時定数

クォッドラント遷移時にプログラムされた半径からの半径ずれが補正されているものの（注入される振幅が最適化されていると仮定）、遷移点を過ぎると半径が明らかに短くなることが真円試験で判明した場合には、時定数設定が大きすぎます（図 2.42 を参照）。

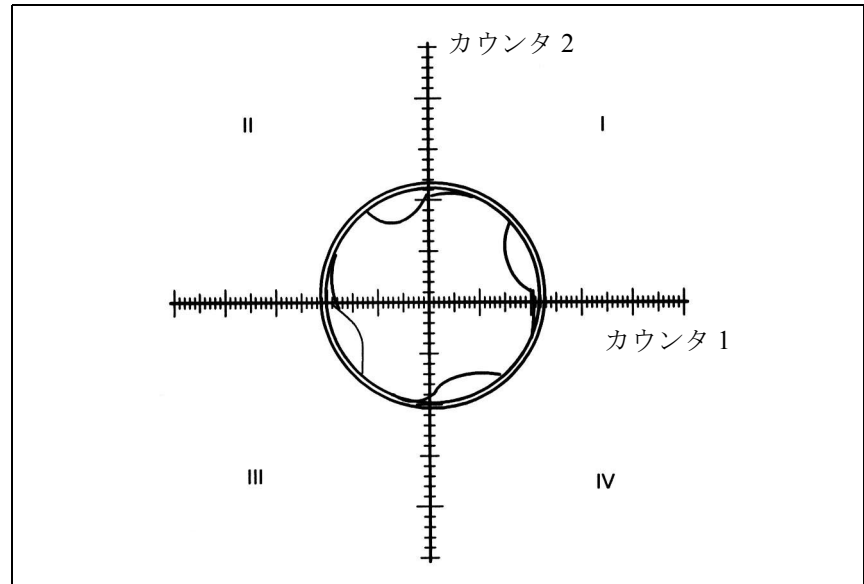


図 2.42 大きすぎる補正時定数

## 適応 yes/no?

時定数と注入振幅が正しく設定されると、作業エリア全域にわたって（様々な半径と速度について）真円試験と位置決め両方で満足な結果が得られます。この場合には、曲線適応は不要となります。

しかし、摩擦補正が加速に依存していると判明した場合、ステップ 2 で適応特性を計算する必要があります（ステップ 2：適応特性を用いた摩擦補正を参照）。

## インストールステップ 2：適応特性を用いた摩擦補正

### アプリケーション

摩擦補正が加速に依存し、かつ必要な結果を一定の注入振幅を使って達成することができない場合には適応特性を使用する必要があります。

非常に高い精度が求められる摩擦フィードフォワード制御の作動範囲全域で最適補正を得るには、補正值の加速依存性を計算する必要があります。そのため、加速ゼロと最大加速との間の作動範囲内のさまざまなポイントで依存性を測定する必要があります。測定結果から導き出した適応特性を、マシンデータに軸別に入力します。

## 1. 適応特性の確定

さまざまな軸および速度に関して

1. ... 必要となる注入振幅を確定する必要がある
2. ... 真円試験を用いて注入振幅の補正効果をチェックする必要がある
3. ... 最適振幅をログする必要がある

適応特性（図 2.36 の例を参照してください）は、2.5.2「**■**従来の摩擦補正」に記載されているパラメータを計算して求めます。しかし、検査を目的とする場合にはさらに多くの測定値を取得する必要があります。高速時の小さな半径に対しては、必ず相当多数の補間点をとってください。曲線のサイズをプロットすることによって求める必要があります。

## 2. 加速値の確定

円運動の場合、軸加速値を半径  $r$  を使って求め、移動速度を次の式で求めます。

$$a = v^2 / r$$

フィードレートオーバーライドスイッチを使うと、速度および軸加速値  $a$  を簡単に変更できます。

適応特性の加速値  $a_1, a_2$  および  $a_3$  を、条件  $a_1 < a_2 < a_3$  に応じて、MD 32550:

FRICT\_COMP\_ACCEL1 to MD 32570: FRICT\_COMP\_ACCEL3 に入力してください。

曲線が間違っていてパラメータ化されている場合、アラーム 26001 "Parameterization error for friction compensation" が出力されます。

## 特性設定の例

### 1. 加速の計算

円軌跡においての速度ゼロを通る軸加速は、公式  $a = v^2/r$  を使って求めます。

半径  $r = 10 \text{ mm}$ 、円速度  $v = 1 \text{ m/min}$  ( $=16.7 \text{ mm/s}$ ) の場合は、加速は  $a = 27.8 \text{ mm/s}^2$  となります。

### 2. 曲線編曲点の入力

次の加速を計算して曲線編曲点とします：

$$a_1 = 1.1 \text{ mm/s}^2 ; a_2 = 27.8 \text{ mm/s}^2 ; a_3 = 695 \text{ mm/s}^2$$

従って、次の値をこの順番でマシンデータに入力します：

MD 32550: FRICT\_COMP\_ACCEL1 [n] = 0.0011 [m/s<sup>2</sup>]

MD 32560: FRICT\_COMP\_ACCEL2 [n] = 0.0278 [m/s<sup>2</sup>]

MD 32570: FRICT\_COMP\_ACCEL3 [n] = 0.695 [m/s<sup>2</sup>]

例えば、注入振幅用に次の値を計算します：

MD 32520: FRICT\_COMP\_CONST\_MAX [n] = 30 [mm/min]

MD 32530: FRICT\_COMP\_CONST\_MIN [n] = 10 [mm/min]

（注）低速での結果が思わしくない場合は、直線位置 MD 10200：

INT\_INCR\_PER\_MM あるいは角度位置 MD 10210:

INT\_INCR\_PER\_DEG の計算分解能を増加する必要があります。

MD 32580: FRICT\_COMP\_INC\_FACTOR も参照してください。

（短い移動動作の摩擦補正值の加重係数）

## ■ ニューラル現象突起補正 (QEC)

### ■ 基本事項

QEC の原理

2.5.2「■摩擦補正（クォッドラント誤差補正）」に説明したように、現象突起（QEC）は、ドリフト、バックラッシュ、歪みが原因で逆転時に発生する輪郭誤差を減らすことを目的としています。追加の速度指令値を注入して補正を行ないます（図 2.22 参照）。

従来の QEC では、補正パルスの強度は加速の関数としての特性に基づいて設定できます。この特性は、真円試験を使ってスタート時に特定しパラメータ化する必要があります（図 2.37 を参照）。その手順はどちらかと言えば複雑で経験を必要とします。

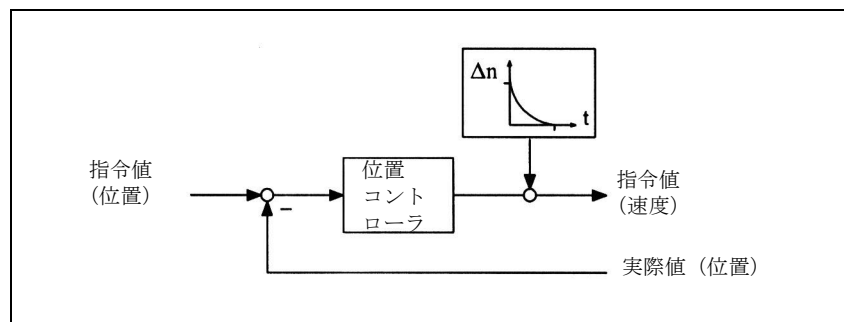


図 2.43 追加速度指令値パルスの注入

### ニューラルネットワークを持つ QEC の長所

これまで手作業でパラメータ化していた特性ブロックをニューラルネットワークと置換えることができます。ニューラルネットワークと置換えることで次の利点が得られます。

- 補正特性はスタートアップエンジニアが手作業で設定する必要がなく、学習フェーズで自動的に設定されるため、スタートアップが簡単になります。
- 摩擦補正を手作業でパラメータ化する場合、補正特性を 4 本の直線を持つ多角形定義によって近似します（図 2.36 を参照）。ニューラルネットワークを使うとはるかに優れた精度の高い本当の曲線を得ることができます。  
特性カーブの分解能を精度要求事項に合わせ、補正振幅の指向性を考慮することができます。  
補正振幅の他に、特別に崩壊時間を加速に適用することもできます。
- このシステムを使用すると、自動最適化をいつでも現場で簡単に行うことができます。

## ニューラル QEC 要件

ニューラルネットワークを持つ QEC を構築する際の必須条件は、クォッドラント遷移でワークピースに発生する誤差が測定系で検出できることです。この条件を満たすには、直接測定系を使用するか、あるいは、モータに対する負荷応答が高い（剛性の高いマシン構造、少ないバックラッシュ）間接測定系を使用するか、あるいは適切な補正を行なう必要があります。直接測定系を用いる場合、発生する可能性があるバックラッシュはバックラッシュ補正で補正する必要があります。

## 学習／作業フェーズ

ニューラルネットワークを持つ QEC には次の 2 つのフェーズがあります：

- 学習フェーズ

学習フェーズ中は、任意のパターンの行動がニューラルネットワークに記憶されます。そして、入出力信号の関係を学習します。学習結果は、不揮発性ユーザメモリに保存される学習補正特性となります。学習プロセスの起動と停止は、特別な高級言語コマンドを使って NC パートプログラムにプログラミングされます。

- 作業フェーズ

作業フェーズ中は、学習特性にしたがって追加速度指令値が注入されます。このフェーズ中は保存された特性を変更することはできません。

学習フェーズは複数（最大 4 つまで）の軸で同時に実行することができます。

ニューラルネットワークのトレーニングの詳細については 2.5.2 「**■ニューラルネットワークの学習**」を参照してください。

学習および作業フェーズならびにその結果としてのニューラル QEC は軸にのみ関連します。軸同士に相関関係はありません。

## 特性値の保存

学習フェーズが完了するとすぐに、ネットワークパラメータ（QEC システム変数）などの計算された補正データ（ユーザメモリの特性値）をオペレータが選択するファイルに保存する必要があります。これらのファイルは通常は "AXn\_QEC.INI" と呼ばれます。

## 特性値のロード

これらの保存され学習されたデータは、パートプログラムと同じ方法でユーザメモリに直接ロードすることができます。

テーブルを含むパートプログラムがロードされた場合、補正值は NC ユーザメモリに転送されます。特性値は、補正值がイネーブルになって初めて有効になります。補正機能がアクティブになっている場合、特性値を書込むことはできません（MD 32500: FRICT\_COMP\_ENABLE を 0 に設定し、アクティブにする必要があります）。

QEC の場合：

MD 32500: FRICT\_COMP\_ENABLE = 1 (QEC アクティブ) をセットして QEC をイネーブル（そしてアクティブに）してください：

## 推奨されるスタートアップ方法

上述したように、制御装置に統合されているニューラルネットワークは学習フェーズ中に最適補正データを自動的に取得します。

関連する軸は、セクションごとに加速値を一定にしてリバースを実行しなければなりません。学習フェーズをアクティブにする前に、ニューラルネットワーク (QEC システム変数) を要求事項にしたがって事前に割当てておいてください。

できる限りスタートアップを簡略化するために、参照例として NC プログラムが提供されています。

2.5.2 「■ニューラル QEC のスタートアップ」でも説明されているように、スタートアップエンジニアはまず、この参照例を使って軸の特性と推奨 QEC パラメータ値を学習し、次に真円試験を使って得られた輪郭精度を検査する必要があります (「■真円試験」を参照)。その結果が要求事項を満たさない場合、再最適化を実行してパラメータを適切に変更 (再学習) しなければなりません (2.5.2 「■ニューラル QEC のパラメータ化」、 「■ニューラルネットワークの学習」 および 「■さらなる最適化および介入オプション」 を参照)。

## ■ ニューラル QEC のパラメータ化

### マシンデータ

ニューラル QEC の基本構成データはマシンデータとして保存されます。

- MD 32490: FRICT\_COMP\_MODE  
摩擦補正方法 (2 = ニューラル QEC)
- MD 32500: FRICT\_COMP\_ENABLE  
摩擦補正アクティブ
- MD 32580: FRICT\_COMP\_INC\_FACTOR  
短い移動ブロックの摩擦補正值の加重係数
- MD 38010: MM\_QEC\_MAX\_POINTS  
ニューラルネットワークを持つ QEC の補正值の最大数

これらのマシンデータを使って、メモリスペースが不揮発性ユーザメモリに保存されるとすぐにニューラル QEC がアクティブになります。手順および割当てについては、2.5.2 「■ニューラル QEC のスタートアップ」または 2.5.4 「データ説明 (MD, SD)」を参照してください。

システム変数を使って他のすべてのデータを設定します。

---

## QEC システム変数

ニューラルネットワークをパラメータするためのデータは、NC プログラムによって読書きできるシステム変数であると定義されます。次のシステム変数を使って、ニューラルネットワークをパラメータ化します：

- **\$AA\_QEC\_COARSE\_STEPS** 「特性の量子化（粗）」

このパラメータが入力信号の量子化（粗）を定義します（したがって特性の分解能となります）。選択された値が大きければ大きいほど、メモリ要求事項は高度になり、トレーニングフェーズに必要な時間が長くなります。詳細については、本セクションの最後を参照してください。

値の範囲：1 ～ 1024; 推奨値：49

- **\$AA\_QEC\_FINE\_STEPS** 「特性の量子化（微）」

このパラメータが入力信号の量子化（微）を定義します（したがって特性の分解能となります）。選択された値が大きければ大きいほど、メモリ要求事項は高度になります。

値の範囲：1 ～ 16; 推奨値：8

- **\$AA\_QEC\_DIRECTIONAL** 「指向性」

このパラメータが、補正を方向を考慮して注入するかどうか定義します。アクティブにすると、個別の特性が確定され、それぞれの加速方向ごとに保存されます。2つの特性が用いられるため、2倍のメモリスペースを不揮発性ユーザーメモリに予約しておく必要があります。

値の範囲：TRUE/FALSE; 推奨値：FALSE

- **\$AA\_QEC\_LEARNING\_RATE** 「アクティブ学習フェーズの学習レート」

学習レートを用いると、ニューラル QEC のアクティブ学習フェーズにどれくらいの速度で最適特性を学習させるかを決定することができます。この値は、どの程度まで変数が注入振幅に影響を与えるかを定義するときに使用する加重係数です。この値が 100% より大きい場合、特性はより迅速に学習されますが、学習速度値（加重係数）を大きくしすぎると不安定性の原因となります（2段階応答）。

正常運転（<50%）時の再学習には、小さな学習速度を推奨します。学習速度を大きくすると、速度がゼロを通過するときに小さな外乱が発生するたびに特性が変化することになります。

値の範囲：> 0%; ≤ 500%; 推奨値：50%

- **\$AA\_QEC\_ACCEL\_1 / \_2 / \_3** 「特性範囲 1 / 2 / 3 についての加速リミット値」

加速特性を 3 つの範囲に分割します。それぞれの範囲では量子化の異なった加速段階を適用します。低加速範囲では、様々な補正值を得るために特に高い分解能が要求されます。そのため、入力信号がより細かく量子化され、加速はより小さくなります。

推奨値

- \$AA\_QEC\_ACCEL\_1      20 mm/s<sup>2</sup>      (= \$AA\_QEC\_ACCEL\_3 の 2%)
- \$AA\_QEC\_ACCEL\_2      600 mm/s<sup>2</sup>      (= \$AA\_QEC\_ACCEL\_3 の 60%)
- \$AA\_QEC\_ACCEL\_3      1000 mm/s<sup>2</sup>      ( 作業範囲内での最大加速 )

パラメータ \$AA\_QEC\_ACCEL\_3 の値は、要求事項に適する値を入力する必要があります。すなわち、ニューラルネットワークは、この範囲内では最適に作業したり学習したりすることができません。パラメータ化された作業エリアよりも大きい加速が検出された場合、作業エリアの最大加速時に確定された注入振幅が使用されます。高速加速時には、注入値はほぼ一定です。

加速範囲内で補正指令値パルスが不十分である場合にのみ、推奨値を変更する必要があります。詳細については 2.5.2 「■さらなる最適化および介入オプション」を参照してください。

- \$AA\_QEC\_TIME\_1 "      「ニューラル QEC 崩壊時間の時定数」  
崩壊時間の適応特性が使用されていない場合、この機能を使って補正指令値パルスの崩壊時間を設定します。

中間加速範囲での作業ポイントで真円試験を実施することによって、最適崩壊時間を手動で確認しなければなりません。詳しい手順は、摩擦補正を扱ったセクションで述べられています (2.5.2 「■従来の摩擦補正」) (MD 32540: FRICT\_COMP\_TIME と同様)。

推奨値 (15 ms) を使うと、良い結果を得ることができます。

値の範囲 :  $\geq 0$ ; 推奨値 : 0.015s

崩壊時間適応がアクティブであると、\$AA\_QEC\_TIME\_1 がオペレーティング範囲の中心でフィルタ時定数を測定します (0.5 \* AA\_QEC\_ACCEL\_3 を使用して)。

- \$AA\_QEC\_TIME\_2      「補正値の崩壊時間を調整するための補正時定数」  
値がゼロかもしくは \$AA\_QEC\_TIME\_1 以下の場合、適応は実行されません。  
崩壊時間は通常は作業エリアの全域にわたって一定です。しかし特殊な場合は、非常に小さな加速範囲において崩壊時間を増やすとよい場合、あるいは高加速時に崩壊時間を減らすとよい場合があります。詳細については 2.5.2 「■さらなる最適化および介入オプション」を参照してください。  
値の範囲 :  $\geq 0$ ; 推奨値 : 0.015 s (A\_QEC\_TIME\_1 と同一)

- \$AA\_QEC\_MEAS\_TIME\_1 / \_2 / \_3  
「加速範囲 1 / 2 / 3 でエラー基準を決定するための測定期間」  
補正値注入の基準が満たされると直ちに測定時間が開始します (すなわち、設定されている速度の符号が変わります)。測定終了は設定パラメータ値によって定義されます。  
特性範囲ごとに測定時間を変える必要があります。



推奨値

- \$AA\_QEC\_MEAS\_TIME\_1: 0.090 s (= 6 \* \$AA\_QEC\_TIME\_1)  
- \$AA\_QEC\_MEAS\_TIME\_2: 0.045 s (= 3 \* \$AA\_QEC\_TIME\_1)  
- \$AA\_QEC\_MEAS\_TIME\_3: 0.030 s (= 2 \* \$AA\_QEC\_TIME\_1)

これらの測定範囲で補正が十分に行われない場合、または \$AA\_QEC\_TIME\_1 が変更された場合にのみ、補正值を変更することができます。詳細については 2.5.2 「■さらなる最適化および介入オプション」を参照してください。

## パラメータの転送

NC プログラムがスタートした後に、QEC システム変数が不揮発性ユーザメモリに保存されます。QEC システム変数は、メモリが消去あるいは再フォーマットされるか、あるいは新規の学習あるいは再学習プロセスが開始するか、あるいは NC プログラムによって上書きされるまでは変化しません。

学習サイクルをコールする前に、すべてのシステム変数に学習プロセス用の有効値を割当てする必要があります。例えば、このことはサブプログラム中で行なうことができます。この NC プログラムが走っていて、リセットが実行されていれば、QEC データはアクティブです。

## 特性データ

学習プロセスで確定された特性データは、予約されているユーザメモリにシステム変数として保存されます。

フォーマット: \$AA\_QEC[n] n の範囲: 0 ~ 1024

これらの値が学習した特性を内部フォーマットに書込みます。したがって、絶対に変更しないでください。

## 特性の量子化

特性の量子化、従って分解能は、量子化（微）(\$AA\_QEC\_FINE\_STEPS) と量子化（粗）(\$AA\_QEC\_COARSE\_STEPS) の 2 つの量で定義します。分解能が細くなればなるほど、メモリ要求は大きくなり、学習フェーズに必要となる時間は長くなります。

要求されるメモリロケーションの数および量子化インタバルの総数は、次の式を使って求めます：

メモリロケーションの数 = \$AA\_QEC\_FINE\_STEPS \*  
(\$AA\_QEC\_COARSE\_STEPS+1)

1 つの軸に最大で 1025 のメモリロケーションを予約できます。このように、精度要求が高くても十分に満足 of いく高分解能を得ることができます。

量子化（粗および微）の特性値の意味、ならびにパラメータ "Detailed learning y/n" の機能としての学習フェーズに及ぼす効果については、図 2.44 ~ 2.46 で説明します。理解しやすいように 3 つの例を用います。

事例 1: 量子化（粗）> 1; 量子化（微）= 1 (通常, 量子化（微）は 8 以下です):  
 特性の補間点は量子化（粗）によって定義されます (図 2.44 を参照)

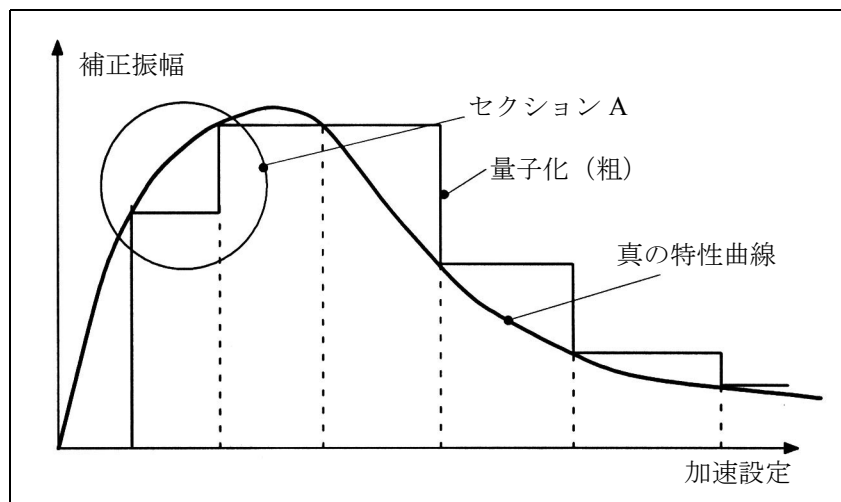


図 2.44 特性の量子化（粗）

事例 2: 量子化（粗）> 1; 量子化（微）> 1; "Detailed learning"（詳細学習）が停止する（この設定はデフォルト）  
 ±の事例の場合, 量子化（粗）の補間点間の量子化（微）には, ディスクリート直線補間を使用します。  
 量子化（粗）の補間点でのみ学習が行われるため, 学習フェーズは 1 と同一です。  
 図 2.45 は, 量子化（微）が量子化（粗）内の特性セクションに与える影響について示しています (図 2.44 のセクション A を参照)。

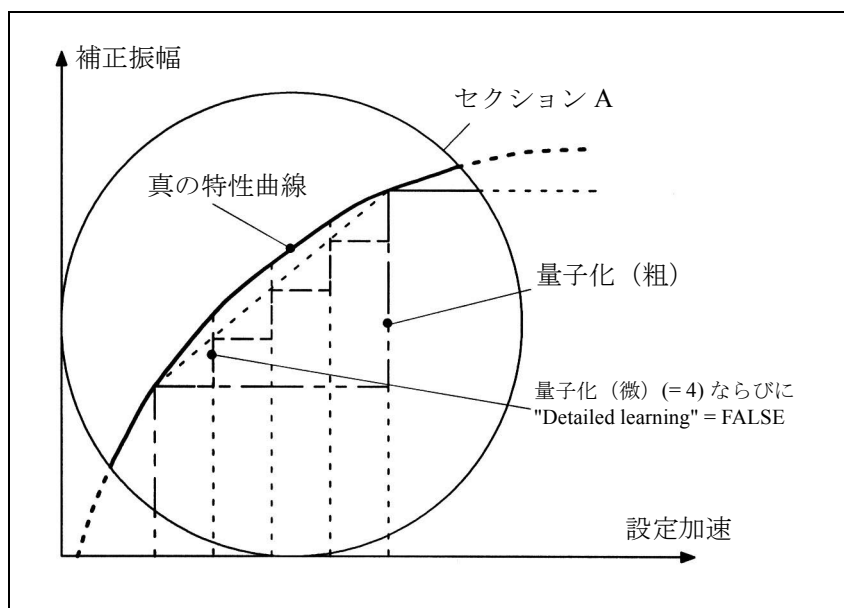


図 2.45 "Detailed learning"（詳細学習）停止時の量子化（微）の影響

事例 3: 量子化（粗） $> 1$ ; 量子化（微） $> 1$ ; "Detailed learning"（詳細学習）がアクティブになっています（非常に高度な精度要求時にのみ使用することを推奨します）。

"Detailed learning" を使うと、量子化（粗）と量子化（微）の両方の補間点で学習が実行されます。

したがって、学習時間が非常に長くなります。

図 2.46 は、大きく揺らいだ特性を使って、"Detailed learning" の選択時と非選択時の効果をわかりやすくしています。

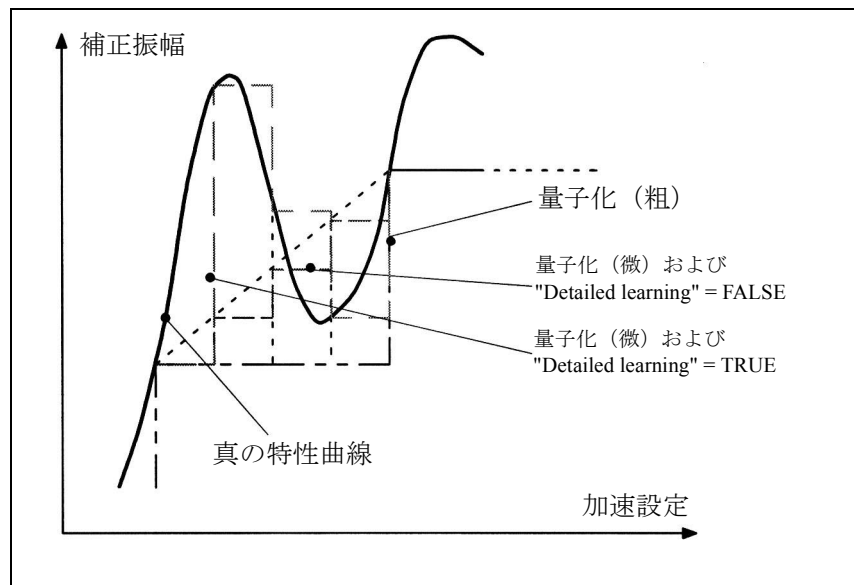


図 2.46 "Detailed learning" アクティブ時の量子化（微）の効果

## ■ ニューラルネットワークの学習

### 学習プロセスシーケンス

学習フェーズ中にある種の応答タイプがニューラルネットワーク上に刻印されます。入力信号と出力信号の関係が学習されます。

学習プロセスは、NC プログラムによって完全に制御され、次のようなエリアに分割されます。

1. 学習プロセス用の QEC システム変数をプリセットする。
2. (NC プログラムをスタートすることによって) QEC システム変数をアクティブにする
3. 学習サイクルをパラメータ化する
4. 学習サイクルをスタートする

結果は、学習済み補正特性として不揮発性ユーザメモリに保存されます。

真円試験を使って、この結果を検査する必要があります (2.5.2 「■真円試験」を参照)。

### 参照用の NC プログラム

スタートアップエンジニアがニューラルネットワークを持つ QEC を簡単にスタートアップできるように、学習動作ならびに QEC システム変数 (推奨値) 割当ての見本ルーチンを持つ NC プログラムを使用できます。

次のような参照用 NC プログラムがあります。

- |                |  |
|----------------|--|
| • QECLRNP.SPF  | 基準を使う学習 (オプション "POLY" が必要です)               |
| • QECLRNC.SPF  | 円を使う学習                                     |
| • QECDAT.MPF   | システム変数の割当ておよび学習サイクルのパラメータ化のための参照用 NC プログラム |
| • QECSTART.MPF | 学習サイクルをコールする参照用 NC プログラム                   |

これらの NC プログラムは、YS840DI の基本 PLC プログラムのディスクセットに入っています。

NC プログラムだけを使って学習プロセスを構築する場合、次のような利点があります：

- 学習は完全に自動的に行われ、オペレーターは介入する必要はありません。マシンタイプについて最適学習パラメータが見出されていて、そのパラメータだけを使ってそれぞれのマシンの特性を個別に確定したり、再教育したりする場合は、この方法は連続スタートアップに適しています。
- 同時に複数 (最大 4 まで) の軸で学習を実行できます。よって、マシンの学習フェーズをかなり低減できます。
- 移動動作を特別要求事項に簡単に適合することができます。
- 単純な MMC (例えば、MMC 100) が使用されている場合でも、ニューラル QEC のスタートアップが可能です (例外：MMC での真円試験は、MMC 101 から 103 で可能です；その他では、インストールツールを使用してください)。

## ■ 学習動作

指定の応答を学習するために実行する必要がある軸移動動作は、NC プログラムによって作成されます。基準学習サイクルの学習動作は放物線状動作を持つ NC ブロックグループで構成されています（その結果、動作は速度指令値のゼロを通過した後可能な限り一定となります。参照、図 2.47）、そしてこのブロックで、軸は一定の加速率でセクションごとに前後に移動します。加速はグループを追うごとに減少します。図 2.47 に、このようなグループを構成する NC ブロック 2 から 3、5 から 6、および 8 から 9 を示しています；ブロック 1、4、7 および 10 ではより小さな加速値への遷移はありません。

（注）学習パラメータをプリセットとして使用するためには、学習フェーズ中にフィードレートオーバーライドスイッチを 100% に設定する必要があります。

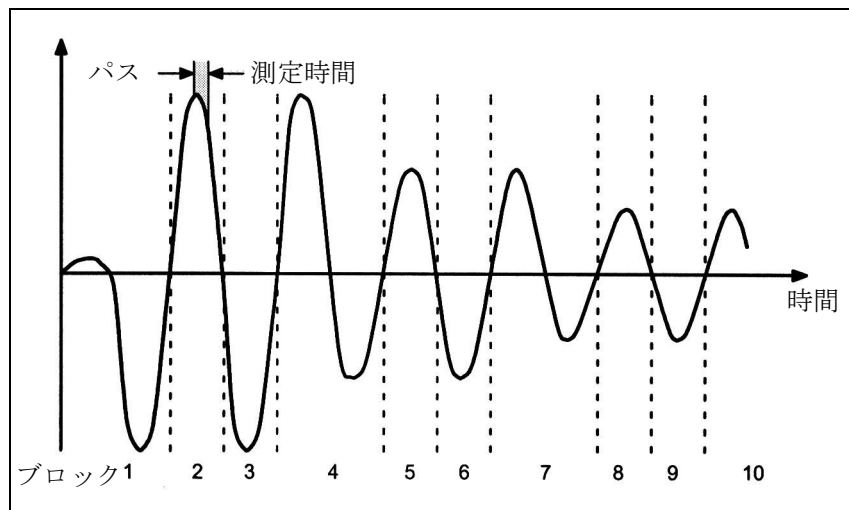


図 2.47 QEC 特性を学習する際の軸の一般的な移動動作

## システム変数の割当て

学習サイクルがコールされる前に、すべての QEC システム変数を学習プロセスに必要な値に設定してください。基準 NC プログラムに推奨された値は、必要であれば、検査して変更してください（2.5.2 「■ニューラル QEC のパラメータ化」を参照）。

## 学習 ON / OFF

ニューラルネットワークの実際の学習プロセスが、基準 NC プログラムでアクティブになります。次のような高レベル言語コマンドを使って行うことができます：

QECLRNON(axis name 軸名 1, ... 4)（指定軸の）学習をアクティブにする

この期間内でのみ、特性を変更できます。

要求された軸の学習動作の完了後、すべての軸の学習プロセスが停止します。高レベル言語コマンドを使って実行することができます：

QECLRNOF（同時にすべての軸の）学習を停止する

また、パワーオンのリセット、プログラム (M02/M30) の終了、またはオペレータパネルリセットの後、学習は停止します。

---

現在の "Learning on / off" 状態は, "QEC learning active" を使ってサービス画面 "Axes" に表示されます (1 = active アクティブ ; 0 = inactive 停止 )。

## 学習サイクルコール

学習がアクティブになっている場合, 基準 NC プログラムが次のような入力パラメータを介して学習サイクルをコールします :

- 学習が適用される軸の数 (最大数 4)

要求事項 : 2 以上の軸の学習が同時に行われる場合, 関連する軸の QEC システム変数はすべて同じ値です。これらの値は監視され, 同一でない場合にはエラーメッセージが表示されます。

- 学習軸の名称
- 初期番号 (すべての軸で同一)  
値は常に 0 (指令値ブランチ)
- 学習モード (initial learning (初期学習) = 0; relearning (再学習) = 1)

0: 初期学習アクティブ。学習の前にネットワークのすべての値をリセットします。

1: 再学習アクティブ。指定のステップ幅ですでに学習した値で学習は継続します。

- 詳細学習アクティブ yes/no (TRUE/FALSE)

FALSE: "Detailed learning" がアクティブになっていません。したがって, 特性は加速の量子化 (粗) のステップ幅で学習されます。

TRUE: "Detailed learning" がアクティブになっています。したがって, 特性は加速の量子化 (微) のステップ幅で学習されます。すなわち, 粗ステップごとに 10 ステップの量子化 (微) を使うため, 特性の確定には時間がかかります。よって, このパラメータはきわめて高い精度の要求事項の場合にのみ使用してください。

(注) "Detailed learning" を選択する場合, 学習パスの数は学習フェーズを減少するために削減できますし, また削減する必要があります (推奨範囲 : 1 から 5)。

- 学習パスの数

デフォルト値 = 15; 範囲 > 0

このパラメータの影響は, "Detailed learning active" が設定されているかどうかによって異なります。

- a) 詳細学習がアクティブでない (= FALSE):

試験動作（前後）の数を各加速ステップに定義します。その数が大きくなると, 学習の精度が高くないますが, 時間は長くなりません。

方向補正 (\$AA\_QEC\_DIRECTION = TRUE) を使って, 各軸の試験動作数のパラメータを作成します。

- b) 詳細学習がアクティブ (= TRUE):

この場合, 最大加速から最小加速あるいはその逆の完全パスの数が, 細かいステップ幅を使用して起動されます。

言い換えると, 値が 1 の場合, 加速ステップは最大値を使ってスタートすることで実行されます。すべての加速ステップで, 方向補正がない

(\$AA\_QEC\_DIRECTION = FALSE) 場合は 2 つの試験動作が行われるか, あるいは各加速ステップごとに 4 つの試験動作が行われます。

マシンタイプのデータブロックがすでに存在していて (シリーズマシン), それらがさらなる最適化の基礎として使用されている場合, "Number of learning passes" を低減することができます。

- セクション別学習アクティブ yes/no (TRUE/FALSE)

特定の加速範囲で "Section-by-section learning" が特に "Detailed learning" に意味を持つ場合, 例えば, マシンの技術的に重要な範囲。その範囲を適切に定義することによって, 学習フェーズを削減することができます。

デフォルト = FALSE

- "Section-by-section learning" の範囲限界 (最小加速, 最大加速); "Section-by-section learning active" にのみ関連します。

デフォルト値 = 0; フォーマット : mm/s<sup>2</sup>

- 試験動作 1 回にかかる時間 (前後動)

デフォルト値 = 0.5; フォーマット : s (秒)

(周波数 2 Hz に対応)

## 要求事項

学習フェーズには, ニューラル QEC に速度フィードフォワード制御 (MD 32620:

FFW\_MODE=1; FFWON) が必要ですが, ジャーク制限は必要ありません

(BRISK)。したがって, フィードフォワード制御を正しくパラメータ化し最適化する必要があります。学習プロセスがスタートすると, 速度フィードフォワード制御がアクティブになっているかチェックされます。チェックされない場合, 学習プロセスはキャンセルされ, エラーメッセージが表示されます。

## ■ ニューラル QEC のスタートアップ

### 一般事項

ここでは、ニューラルネットワークを持つ QEC のスタートアップについて説明します。前述したように、学習フェーズ中の補正特性は自動的に決定されます。

関連軸は、セクションごとに一定の加速値で反転しなければなりません。また、ニューラルネットワークのパラメータ化のシステム変数を、それに合うようにリセットする必要があります。

できるだけスタートアップを簡素化するため、基準例として NC プログラムを備付しています (2.5.2 「■ニューラルネットワークの学習」を参照)。

学習プロセスでは、"initial learning" (特に最初のスタートアップ) と "relearning" (特にすでに学習済みの特性の再最適化) を区別します。"initial learning" と "relearning" の手順は後で説明します。

マシンの補正特性を最初に学習する場合、2.5.2 「■ニューラルネットワークの学習」に指定した基準 NC プログラムの使用を推奨します。

### "Initial learning" プロセス

"Initial learning" → サイクルパラメータ "Learning mode" = 0

- 1.a) 要求された軸に対してニューラルネットワークを持つ QEC をアクティブにする：

MD 32490: FRICT\_COMP\_MODE = 2

注：ニューラルネットワークを持つ QEC はオプションです。

- b) 補正ポイント用にメモリスペースを予約する

MD 38010: MM\_QEC\_MAX\_POINTS

必要な容量が不明の場合、最初は十分なメモリを予約しておいてください (項目 11 も参照してください)。

- c) 速度フォワード制御をパラメータ化し、最適化します (学習フェーズに必要)。

- d) ハードウェアのリセットを行います (不揮発性ユーザメモリの再割当てのため)。

- 2.QEC システム変数をアクティブにする：

基準 NC プログラム QECDAT.MPF を用いて、関連するすべての軸に QEC システム変数を割当て (必要であれば、推奨値を使用します)、NC プログラムをスタートします。エラーメッセージが出た場合、値を修正し、NC プログラムを再起動します。

3. マシン軸を要求された位置に移動させる基準 NC プログラムを作成し、基準学習サイクル QECLRN.SPF をコールします (参照、例プログラム QECSTART.MPF)。パラメータがデフォルトにしたがって効果を発揮できるように、フィードレートオーバーライドスイッチを学習フェーズ 100% に設定してください。

4. NC プログラムをスタートして学習フェーズをアクティブにする

パラメータ化したすべての軸の補正特性を同時に学習します。

学習フェーズは指定された学習パラメータによって変化します。

デフォルト値を使用する場合、数分ほど時間がかかります。

"QEC learning active" 画面のサービス画面 "axis" で、関連する軸の状態を観察することができます。



5. 要求された軸の補正值注入をアクティブにする：

MD 32500: FRICT\_COMP\_ENABLE = 1 .

6. メニュー "Circularity test measurement" の真円試験のトレースのパラメータ化 (MMC101-103 またはインストールツールを使用する場合)。

基準 NC プログラムのパラメータ値

半径 [mm]:

フィードレート [mm/min].

ここで、垂直ソフトキー "start" を使って測定機能をイネーブルにします。

7. テスト動作 (円) を使って NC プログラムをスタートします。

円動作中に、位置の実際値が記録され、受動ファイルシステムに保存されます。データ記録が終了すると、記録された輪郭がダイアグラムで表示されます。

8. 記録された輪郭のクオッドラント遷移をチェックします。

9. その結果によって、必要があれば、項目 2, 4, 7, 8 を繰り返します。最初に、システム変数を変更しなければならないこともあります (2.5.2 「**ニューラルネットワークの学習**」も参照してください)。

10. 輪郭の精度が要求事項を満たしたら、補正特性を保存します (2.5.2 「**ニューラルネットワークの学習**」も参照してください)。

11. 必要であれば、以前に補正值に予約したメモリエリアを実際に必要なメモリに変更することができます。

注意 : MD 38010: MM\_QEC\_MAX\_POINTS の設定を変更する場合、不揮発性ユーザメモリは自動的にシステムパワーオンに再割当てされます。不揮発性ユーザメモリのユーザデータはすべて失われます。したがって、最初にユーザデータのバックアップをとってください。

## "Relearning" (再学習) プロセスのシーケンス

"Relearning" → サイクルパラメータ "Learning mode" = 1

"Relearning" 機能を使うと、すでに学習した特性を、簡単かつ自動的に再最適化できます。すでにユーザメモリに保存されている値を基本にします。

"relearning" (再学習) の学習フェーズには、(例えば, "initial learning" から) マシンに採用している基準 NC プログラムを使用してください。一般に、QEC システム変数の以前の値がそのまま使用できます。学習サイクルがコールされる前に、パラメータ "learning mode" を 1 ("relearning" の意味) に設定する必要があります。これを使用すると、"number of training passes" (訓練パスの数) が少なくなることがあります。

## "Relearning" (再学習) オペレーションのシーケンス

次に, "Relearning" プロセスに関連するオペレーションのシーケンスを説明します。

1. 特性値がまだユーザメモリ (RAM) に保存されていない場合 (例えば, シリーズマシンのスタートアップ), 事前に最適化したデータブロックをロードする必要があります (2.5.2 「**■**ニューラルクォッドラント誤差補正 基本事項」を参照)。
2. NC プログラムを使用して, マシン軸を要求された位置に移動させて, 学習サイクルをコールします。学習サイクルのパラメータ (例えば, QECLRN.SPF) が "relearning" 用に変更されます。
  - "learning mode" = 1 に設定します
  - 必要であれば, "number of learning passes" (訓練パスの数) を少なくします (例えば, 5 に)。
  - 必要であれば, "section-by-section learning" (セクション別の学習) をアクティブにし, 関連範囲限界を定義します。
3. この NC プログラムスタートすることによって学習フェーズをアクティブにします。パラメータ化した軸すべての補正特性が同時に学習されます。
4. メニュー "Circularity test measurement" (真円試験測定) の真円試験のトレースをパラメータ化します (MMC101-103 またはインストールツールを使用する場合)。そして, 垂直ソフトキーを使って測定機能をイネーブルにします。
5. 真円試験の試験動作を使って NC プログラムをアクティブにします。  
円動作中に位置実際値が記録されてパッシブファイルに保存されます。データ記録が終了すると, 記録された輪郭が MMC に表示されます。
6. 記録された輪郭のクォッドラント遷移をチェックします。
7. その結果によって, 必要があれば, 項目 3, 4, 5, 6 を繰り返します。最初に, 特定の QEC システム変数を変更しなければならないこともあります (2.5.2 「**■**さらなる最適化および介入オプション」も参照してください)。
8. 輪郭精度が要求事項を満たしたら, 補正特性を保存します (2.5.2 「**■**ニューラルクォッドラント誤差補正 基本事項」も参照してください)。

## ■ さらなる最適化および介入オプション

### 最適化オプション

真円試験の結果が要求されている精度基準に満たない場合、QEC システム変数に選択可能な変更を使ってさらに改善することができます。ニューラル QEC を最適化する方法をいくつか説明します。

### 量子化（粗）および（微）の変更

前述したように、入力変数は量子化（粗）および（微）の値によって量子化されます。

量子化（微）の値が高いと、入力信号の隣接インタバルで "similar" 信号が出力されます。このため、例えば、一定の加速でのみ発生する測定誤差を平均化することができます（図 2.44 参照）。

量子化（微）の値を低くすると、特性の変動を抑えることができます。

ニューラル摩擦補正では、量子化（微）を高く設定することによって最大誤差公差を使用する必要があります（\$AA\_QEC\_FINE\_STEPS を 5 ～ 10 の範囲に設定）。

### 方向補正

補正值が非方向基準で注入されていない場合、補正が反対のクォドラントでは同等の硬化を持たないような状況では、方向摩擦補正を適応する必要があります（図 2.48 参照）。

方向注入はシステム変数 \$AA\_QEC\_DIRECTIONAL = TRUE を介して起動されます。

ここで、次のことを守る必要があります：

- 加速の両方の特性が学習され保存されるので、不揮発性ユーザメモリのメモリスペースは 2 倍必要となります。  
それにしたがって、MD 38010: MM\_QEC\_MAX\_POINTS を設定する必要があります。
- パスは同一ロケーションでは 1 つおきにしか発生しないので、学習パスの数を増加する必要があります。
- 特性ソリューションが同じであれば、スタートアップはより時間がかかります。

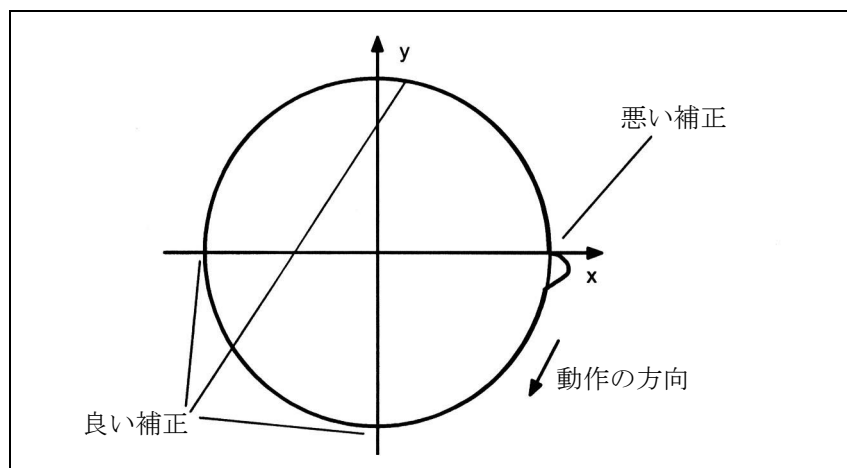


図 2.48 方向摩擦補正の例（真円試験）

## 特性範囲の修正

加速特性は3つの範囲にさらに分割されます。低加速範囲では、広範に渡るさまざまな補正値を再生するために、特に高いソリューションが必要となります。そのため、入力信号の量子化が微となればなるほど、加速はより低となります（図 2.49 参照）。

高加速範囲では、補正値にほんの小さな変更を加えるだけで、小さなリソリューションが完全に要求を満たすものとなります。

2.5.2 「■ニューラル QEC のパラメータ化」で、\$AA\_QEC\_ACCEL\_1 (2% of \$AA\_QEC\_ACCEL\_3) および \$AA\_QEC\_ACCEL\_2 (60% of \$AA\_QEC\_ACCEL\_3) に推奨したパーセント値は、約  $1 \text{ m/s}^2$  までの最大加速（作業範囲）を持つマシンで過去に得た経験値です。

作業範囲が顕著に減少した場合、 $a_1$  および  $a_2$  の制限値を  $a_3$  のパーセントとしていくぶん高めに設定する必要があります。しかし、\$AA\_QEC\_ACCEL\_1 は最大加速の約 5 % の範囲を超えないようにしてください。\$AA\_QEC\_ACCEL\_2 の有効限界は、ほぼ最大加速の 40 % ~ 75 % の値です。

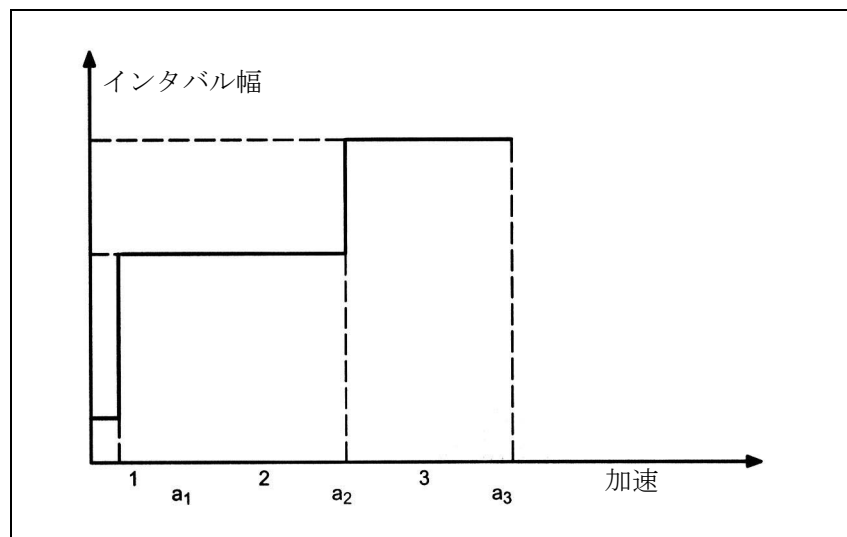


図 2.49 加速範囲のインタバル幅

## 崩壊時間の適応

特別な場合、補正振幅の他に補正指令値パルスの崩壊時間を適応することができます。

例えば、低加速範囲 ( $a_1$ ) において、クォッドラント遷移が良い補正結果を生じるにもかかわらず、その直後に再び半径ずれが生じることが真円試験によって判明した場合、崩壊時間を適応することによって改善することができます。

適応を持たない時定数 (\$AA\_QEC\_TIME\_1) は、中間加速範囲 (50%) でのみ有効です。

図 2.50 に示した特性に準じた補正指令値パルスの崩壊時間の適応を、システム変数 \$AA\_QEC\_TIME\_2 (加速 = 0 の場合) を使ってパラメータ化します。これら 2 つの点を通る e-x 機能にしたがって適応が行われます (図 2.50 参照)。

適用は次の条件で実行されます：

$$\$AA\_QEC\_TIME\_2 > \$AA\_QEC\_TIME\_1$$

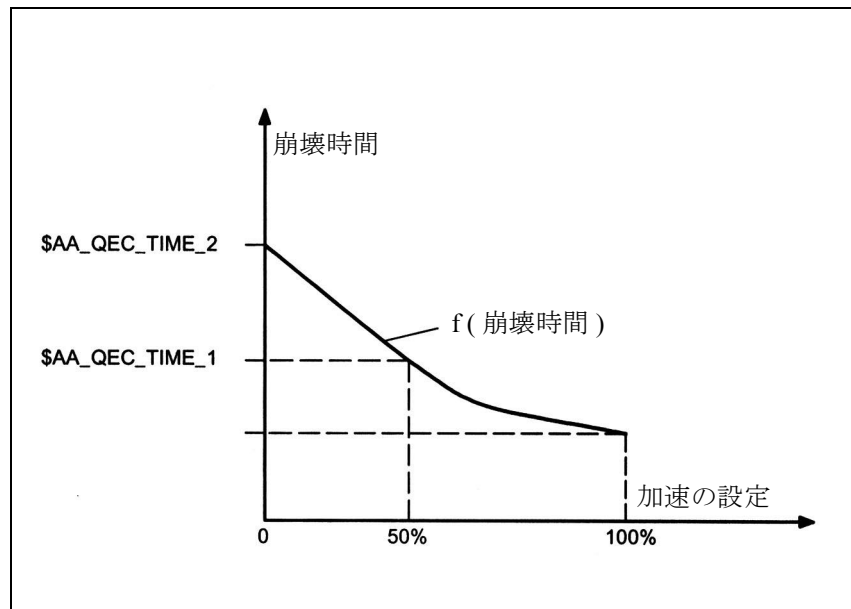


図 2.50 崩壊時間の適応

## 誤差測定時間の変更

ニューラルネットワークの学習フェーズでは、誤差測定時間がゼロ速度パス後に輪郭誤差が監視される時間ウィンドウを決定します。

経験からすると、平均加速レートに使用する誤差測定時間 (約  $2 \sim 50 \text{ mm/s}^2$ ) は、崩壊時間の値の 3 倍となります ( $\$AA\_QEC\_MEAS\_TIME\_2 = 3 * \$AA\_QEC\_TIME\_1$ )。

非常に低いまたは非常に高い加速範囲では、誤差測定時間を適応する必要があります。これは、図 2.51 に示した特性を使って自動的に行われます。

小さな加速の誤差測定期間は、崩壊時間値の 6 倍に設定します

( $\$AA\_QEC\_MEAS\_TIME\_1 = 6 * \$AA\_QEC\_TIME\_1$ ) ; 崩壊時間

( $\$AA\_QEC\_MEAS\_TIME\_3 = 2 * \$AA\_QEC\_TIME\_1$ ) をより大きな加速の誤差測定誤差とする場合には 2 倍にします。

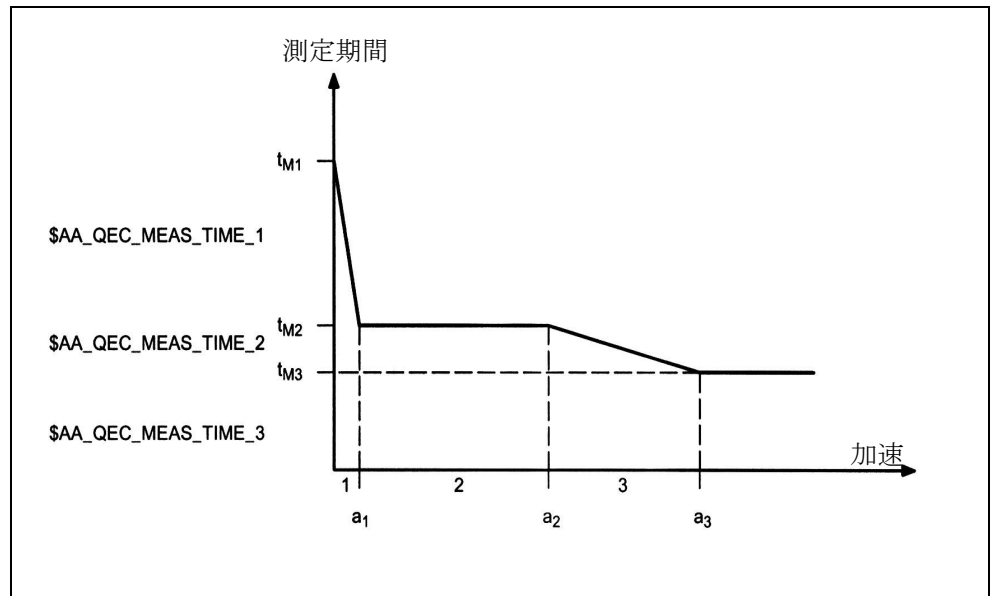


図 2.51 誤差測定時間の加速レートへの依存性

特別な場合、誤差測定時間を再パラメータ化する必要があります。

- QEC の補正時定数に非常に極端な値を設定する場合  
経験から誤差測定時間を 10 ms 未満あるいは 200 ms 以上に設定しても無意味であることが判明しています。
- 補正值の崩壊時間の適応を使用した誤差測定時間のパラメータ化  
補正值の崩壊時間の適応がアクティブになっている場合（上記を参照してください）、加速範囲 1 の誤差測定時間のパラメータ化には  
 $\$AA\_QEC\_MEAS\_TIME\_1 = 3 * \$AA\_QEC\_TIME\_2$  の式が適用されます。  
例：  
崩壊時間 ( $\$AA\_QEC\_TIME\_1$ ) = 10ms  
崩壊時間の適応 ( $\$AA\_QEC\_TIME\_2$ ) = 30ms  
上記の式を使用する場合の加速範囲の誤差測定時間：  
 $\$AA\_QEC\_MEAS\_TIME\_1 = 3 * 30ms = 90ms$   
崩壊時間の適応を使用しない場合：  
 $\$AA\_QEC\_MEAS\_TIME\_1 = 6 * 10ms = 60ms$ .

## 短いトラバース動作での過剰補正

経験より、短い軸位置決め動作を実行する場合、真円試験から算出した最高摩擦補正值は関連軸を過剰補正することが分かっています（例えば、 $\mu\text{m}$  範囲のインフィードの場合）。

このような場合、短い移動動作の補正振幅を低減すると、精度を向上させることができます。

これは、MD 32580 にプログラムされている加重係数です：

摩擦補正がアクティブな場合、\$MA\_FRICT\_COMP\_INC\_FACTOR が自動的に影響を与え（従来の QEC またはニューラルネットワークを持つ QEC）、また制御装置の補間サイクル内で創られた全ての位置決め動作上で作動します。

入力範囲は計算した補正值の 0 ～ 100% です。

## 学習プロセス期間の制御

これまでのセクションで述べたように、学習プロセスの期間はいくつかのパラメータに依存しています。主なものとして、次のような値があります：

- 量子化（粗）(\$AA\_QEC\_COARSE\_STEPS)
- 誤差基準を確定する測定時間  
(\$AA\_QEC\_MEAS\_TIME\_1 up to \$AA\_QEC\_MEAS\_TIME\_3)
- 学習パス数
- 詳細学習アクティブ [yes/no]?
- 量子化（微）(\$AA\_QEC\_FINE\_STEPS) ("detailed learning = yes" が選択されている場合のみ)
- 方向補正アクティブ [yes/no]?  
(\$AA\_QEC\_DIRECTIONAL)
- リバース動作の期間

設定 "Detailed learning active = yes" により学習に必要な時間が増加します。したがって、精度要求事項が高い場合にのみ使用してください。

これらの要求事項が特定の加速範囲に限って適用されるかチェックする必要があります。その場合、詳細学習をセクションごとに実行する必要があります ("Section-by-section learning y/n?" を参照)。いずれの場合でも、学習フェーズの数を低減してください。

上に述べた基準 NC プログラムが推奨パラメータ値と共に使用されている場合、学習プロセス時間には次の時間が決定されています。

- |                  |         |
|------------------|---------|
| • 詳細学習がアクティブでない： | 約 6.5 分 |
| • 詳細学習がアクティブ：    | 約 13 分  |

## ■ クイックスタートアップ

### 学習準備

1. 従来の摩擦補正を使って摩擦補正時定数 (MD 32540 FRICT\_COMP\_TIME) を求めます。
2. POWER ON せずに、次のマシンデータを入力します。

マシンデータ	デフォルト	変更	意味
MD 19330 NC-CODE_CONF_NAME_TAB[8]	0		オプション "IPO_FUNKTION_MASK" をアクティブにする学習プログラム "Polynomial" だけを使うこと ビット 4 = 1
MD 19300 COMP_MASK	0		セットオプション
MD 32490 FRIC_COMP_MODE	1	2	"Type of friction compensation" (摩擦補正のタイプ) ニューラル QEC
MD 32500 FRIC_COMP_ENABLE	0	0	学習のための "Friction compensation active" (摩擦補正アクティブ) を "OFF"
MD 32580 FRIC_COMP_INC_FACTOR	0	0	「短い移動動作の摩擦補正值の加重係数」 (mm 単位のインクリメント)
MD 38010 MM_QEC_MAX_POINTS	0	400	"Selection of values for QEC" =\$AA_QEC_FINE_STEPS *(\$AA_QEC_COARSE_STEPS + 1)
MD 32620 FFW_MODE	1	1	速度フィードフォワード制御
MD 32610 VELO_FFW_WEIGHT	1	1	注入 100%
MD 32630 FFW_ACTIVATION_MODE	1	0	フィードフォワード制御を連続 ON
MD 32810 EQUIV_SPEEDCTRL_TIME	0,004	初期値 $t_{pos} + n_{setSm} \cdot$	等価時定数 n 制御ループを調整

\*)  $t_{pos}$  ... 位置制御サイクル (= 基本システムサイクル \* 位置制御サイクルの係数),  
 $n_{setSm}$  ... 速度指令値平滑化

smoothing (MD 1500 ~ 1521)

3. メモリの再割当てにより (MD 38010), MMC100 を使ってマシンデータを保存します :

「サービス」, 「データオフ」, 「スタートアップデータ, NCK データ」, ならびにプログラムされている場合は "LEC, measuring system error, sag and angularity error compensation tables" (LEC, 測定系誤差, 垂れおよび角度誤差補正テーブル) を PCIN を介して保存します。

パワーオンリセットを実行し, PCIN と "Data IN" (= シリーズスタートアップ) を使って保存データを読み込みます。

MMC102:

"Series startup" (シリーズスタートアップ) ならびにプログラムされている場合は "LEC, measuring system error, sag and angularity error compensation tables" (LEC, 測定系誤差, 垂れおよび角度誤差補正テーブル) を保存します。

パワーオンリセットを実行し, "Start-up" アーカイブに読みこみます (保存データが再びロードされます)。



4. ディスケット "MMC100 TOOLBOX" に入っているプログラムを NC にコピーします（アーカイブを使用）。

QECDAT.MPF

QECSTART.MPF

QECLRN.PSF ("Polynomial" 学習プログラム) または QECLRNC.PSF ("Circle" 学習プログラム) は、NC 上に QECLRN.PSF として保存されています。

ジオメトリ軸の場合、できれば Circle 学習プログラムを使用してください；その他の軸には、Polynomial 学習プログラムを使用してください。

5. 次のプログラムを使用します：

- パートプログラム QECDAT の場合

必要であれば、摩擦補正時定数を使用します（ステップ 1 を参照）

N1340 \$AA\_QEC\_TIME\_1[outNo,axNo] = 0.0xx

N1040 def int numAxes = ..... 学習する軸数を入力します。

N1150 axisName[0] = ..... 第 1 軸の軸名を入力します。

N1160 axisName[1] = ..... 第 2 軸の軸名を入力します。

N1170 axisName[2] = ..... 第 3 軸の軸名を入力します。

N1180 axisName[3] = ..... 第 4 軸の軸名を入力します。

(AX1 から AX8 あるいはマシン軸名またはチャンネル軸名を、"Circle" 学習プログラムの軸名に使用することができます。しかし、"Polynomial" 学習プログラムにはチャンネル軸名以外は使用できません。)

- パートプログラム QECSTART の場合

N1080 def int numAxes = ..... 学習する軸数を入力します。

N1310 axisName[0] = ..... 第 1 軸の軸名を入力します。

N1320 axisName[1] = ..... 第 2 軸の軸名を入力します。

N1330 axisName[2] = ..... 第 3 軸の軸名を入力します。

N1340 axisName[3] = ..... 第 4 軸の軸名を入力します。

(AX1 から AX8 あるいはマシン軸名またはチャンネル軸名を、"Circle" 学習プログラムの軸名に使用することができます。しかし、"Polynomial" 学習プログラムにはチャンネル軸名しか使用できません。)

## "Learning" プロセスを実行します

次のプログラムをスタートします

- QECDAT.QECDAT を選択し、スタートします。  
システム変数が割当てられます。
- QECSTART を選択し、100%オーバライドし、スタートします。  
学習プログラムが約 15 分間走り、約 30 cm 移動動作を行います。メッセージ "REORG not possible" が表示された場合、無視します。このメッセージは約 10 秒間表示されます。メッセージが消えたら、移動動作で学習プロセスが継続します。

## QEC をアクティブにします

マシンデータ：	デフォルト	次に変更	意味
MD 32500 FRIC_COMP_ENABLE	0	1	"Friction compensation active"（摩擦補正アクティブ）をオンにします。

---

## "Circularity test"（真円試験）

"Circularity test"（真円試験）を使って結果をチェックしてください。

### 補正データの保存

補正データを保存します（QEC データを "SERIES START-UP" を使って保存することはできません。ソフトウェアバージョン 4 以降で選択可能）：

MMC100:

PCIN を使用して SERVICES Data\Circle error compensation\All に保存します。

MMC102:

ファイルを SERVICES NCK\NC Active Data\Quadrant Error Co\Quadrant error comp-complete.ini. に保存します。このファイルには全ての補正值が入っています。

注：SERVICES "System set-tings" "for display" で "displayed name length" を "20" に変更し、全ての名称が表示されるようにします。

## ■ 真円試験

### 機能

真円試験の目的は、1 つには摩擦補正機能によって得た輪郭精度のチェックがあります (従来の QEC またはニューラルネットワーク QEC)。真円試験では、円運動中の実位置を測定し、プログラム半径とのずれを図形に表示します (特にクォッドラント遷移時)。

### 手順

NC プログラムによって関連軸の円輪郭が指定されます。スタートアップエンジニアができる限り簡単に真円試験を行えるよう、NC プログラムには真円試験動作の基準例が入っています (基本 PLC プログラムの入ったディスクットのファイル QECTEST.MPF)。スタートアップエンジニアはアプリケーションとしてこの NC プログラムを使用する必要があります。

さまざまな加速値を使って真円試験測定を行い、学習補正特性 (ニューラル QEC の場合) あるいは定義された補正值 (従来の QEC の場合) が要求事項を満たしているか確認します。

円輪郭を変えずに送り速度オーバライドを使って送り速度を変更すると、さまざまな加速を使って簡単に円動作を創ることができます。入力フィールド "feedrate" における測定には実際の送り速度を考慮する必要があります。

選択した円半径は、機械の工作動作の典型的なものでなければなりません (例えば、範囲 10 ~ 200 mm 内の半径)。

円動作の期間中、軸の位置実際値は記録され、パッシブファイルシステムの "trace" に保存されます。従って、真円試験は純粋な測定機能です。

### 真円試験のパラメータ化

円が移動されたり、位置の実際値を記録しなければならない軸名や軸数を、このメニューで選択します。選択した軸が NC パートプログラムにプログラムされている軸に対応しているか確認する必要はありません。

入力フィールド "Radius" および "Feed" でのパラメータ設定は、軸の円動作を制御するパートプログラムから得た値に対応している必要がありますので、送りオーバライドスイッチ設定を確認してください。パートプログラム (送りオーバライドを含む) の値と入力値が一致しているか確認する必要はありません。

"Measuring time" 表示フィールドに、真円動作中の位置実際値を記録する "Radius" および "Feed" 値から算出した測定時間が表示されます。

円が部分的に表示されない (すなわち、測定時間が短すぎる) 場合、送り値を下げてメニューの測定時間を増加することができます。また、真円試験が停止位置から開始される場合にも適用できます。

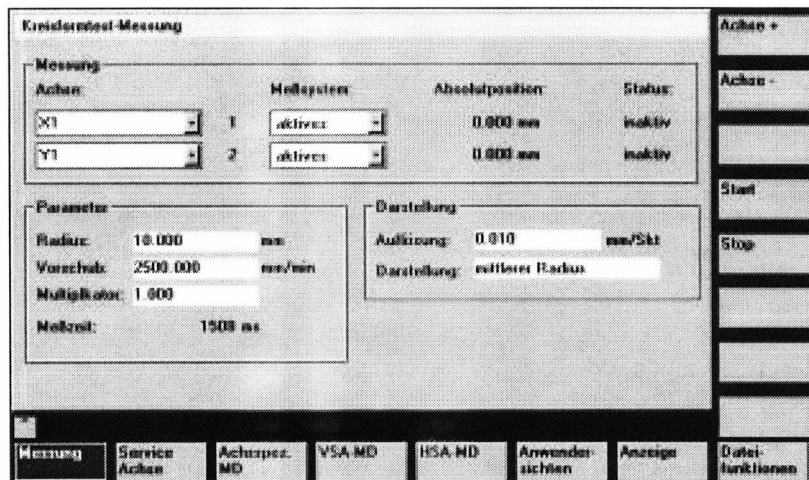


図 2.52 真円試験測定メニュー

## 表示のモード

測定結果の表示モードをプログラミングする際に、次のパラメータ設定も同時に行うことができます：

- 平均半径に基づく表示
- プログラム半径に基づく表示
- 図軸のスケールリング

算出した測定時間がトレースバッファ（最大測定時間 = 位置制御サイクル周波数 \* 2048）で表示可能な時間範囲を超える場合、記録（ $n \times$  位置制御サイクル周波数）に輪郭サンプリングレートを使用すると、完全な円が表示されます。

## 測定のスタート

オペレータはパートプログラムを開始するときには必ず、選択した軸の円動作が保存されている NC Start を使用してください（AUTOMATIC または MDA オペレーティングモード）。

垂直ソフトキーを使って、測定機能をスタートすることもできます。

目的によって、アプリケーションに適したオペレーティングシーケンスを選択できます（パートプログラムの NC スタートおよび測定のスタート）。

指定した軸に対して真円試験がアクティブになるとすぐ、"Status" 表示フィールドにメッセージ "Active" が出力されます。

## 測定の中止

Stop ソフトキーを使っていつでも測定を中止することができます。

Display ソフトキーを使うと、その時点でまだ完全に記録されていない測定が可能な限り全円に近く表示されます。これを確認する監視機能はありません。

必要となる制御パラメータに直接アクセスする場合、ソフトキー Axis-specific MD, FDD-MD および MSD-MD が利用できます。垂直ソフトキー Axis+ および Axis- を使って、必要となる軸を選択することができます。

Service Axis ソフトキー軸を押すと、"Service axis" が表示されます。

ここでは、摩擦補正のスタートアップに関する次のサービスデータが折り返し表示されます。

- QEC 学習アクティブ yes/no?
- 現在の位置および速度実際値

## 表示

Display ソフトキーを選択した場合、画面は記録済み円図の図形表示に切替わります。

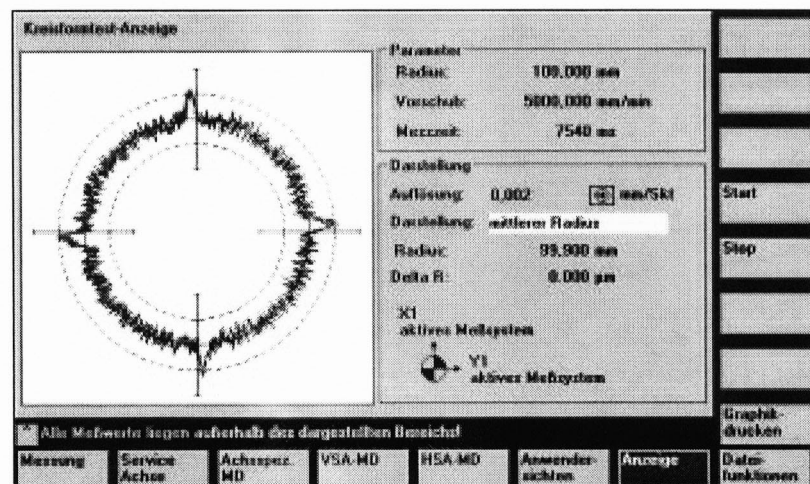


図 2.53 真円試験表示メニュー

この画面では、定められた分解能を使って両方の位置実際値の測定進行状態を円の形で表示します。

文書化のため（測定済み円特性をファイルに順次保存するため）に、プログラム半径、プログラム送り速度、および演算結果測定時間も表示されます。

ユーザは図軸のより微細な設定を入力フィールド分解能に入力し、クォッドラント遷移を強調することができます。分解能を変更した後の図を表示するには、Display ソフトキーを押してください。

## ファイル機能

表示された測定結果ならびにパラメータ設定は、File functions ソフトキーを選択してファイルとして MMC に保存することができます。

## プリンタの設定

MMC \ Printer selection ソフトキーを使い、プリンタ選択の基本画面（図 2.54）を呼び出すことができます。

Print graph ソフトキーを選択し、次にトグルキーを使い、表示された図を直接プリンタに転送するか、あるいはビットマップファイルに転送するかを選択します。

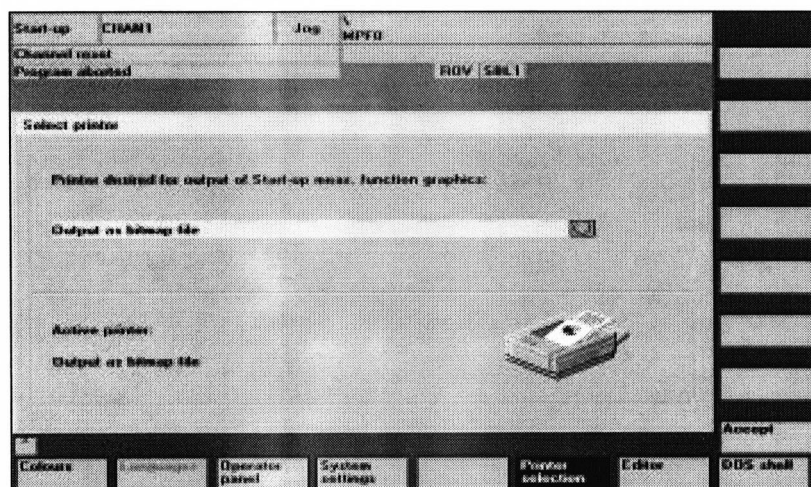


図 2.54 プリンタ選択の基本画面

## プリンタへの直接出力

プリンタは MS-WINDOWS の下で設定されている必要があります。

"Output on printer" が選択フィールドに設定されます。

Print graph のラベルのついたソフトキーを選択すると、画面表示の図が接続されているプリンタに出力されます。

## ビットマップファイルでの出力

図は、ビットマップファイル (\*.bmp) に保存されています。

プリンタ設定選択フィールドでは "Output as bit map file" が選択されます。

Print graph のラベルのついたソフトキーを選択すると、ファイル名割当て画面が "Circularity test display" に現れます。新規のファイル名を入力することができますが、既存のファイル名を選択しドロップダウンリストに上書きすることもできます。

ファイルを保存する場合は、Ok ソフトキーを選択します。

現在の図画面に戻る場合は、Abort ソフトキーを選択します。

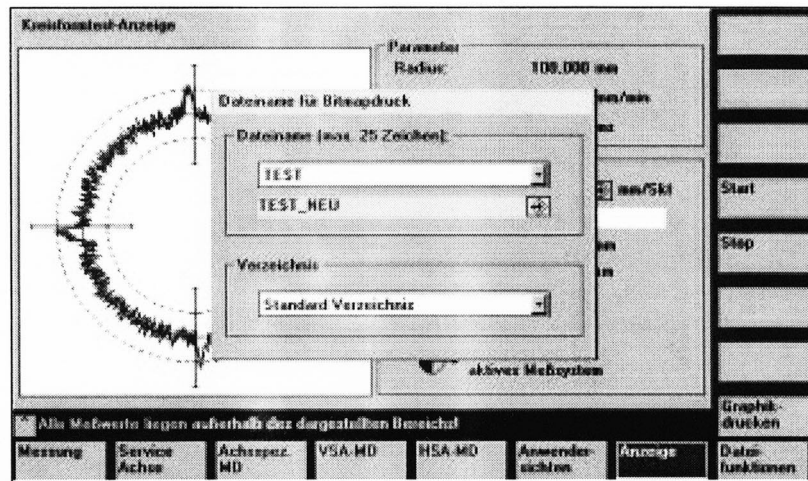


図 2.55 ビットマップファイルとして出力する場合のファイル名の割当て

## ■ ニューラル象現突起補正, クイックスタートアップ

MMC102/103/MMC100 上で放物線／円動作を使ったクイックスタートアップ  
"Neural Quadrant Error Compensation"

### 学習の準備

まず最初に、従来の摩擦補正を使って、摩擦補正時定数 (MD 32540  
FRICT\_COMP\_TIME) を計算します。

表 2.6 パワーオンしないで次のマシンデータを入力

マシンデータ	デフォルト	次に変更	意味
MD 19330 IPO_FUNCTION_MASK	0	8	"Polynomial interpolation" オプションを起動。多項式の場合のみ Bit 4 = 1
MD 19300 COMP_MASK	0	8	オプション "Neural QEC" の場合, bit 4 = 1
MD 32490 FRIC_COMP_MODE	1	2	"Type of friction compensation" ニューラル QEC
MD 32500 FRIC_COMP_ENABLE	0	0	学習用の "Friction compensation active" を "OFF"
MD 32580 FRIC_COMP_INC_FACTOR	0	0	"Weighting factor of friction compensation value for short traverse motions" (短い移動動作についての摩擦補正值の加重係数) ( $\mu$ m インクリメント)
MD 38010 MM_QEC_MAX_POINTS	0	400	"Selection of values for QEC" = \$AA_QERC_FINE_STEPSA * (\$AA_QEC_COARSE_STEPS + 1)
MD 32620 FFW_MODE	1	1	速度フィードフォワード制御
MD 32610 VELO_FFW_WEIGHT	1	1	注入 100%
MD 32630 FFW_ACTIVATION_MODE	1	0	フィードフォワード制御連続 ON

マシンデータ	デフォルト	次に変更	意味
MD 32810 EQUIV_SPEEDCTRL_TIME	0,004	t_pos.+ n_setSm. *)	等価時定数の調整, n 制御ループ

\*) t\_pos. ... 位置制御サイクル (= 位置制御サイクルの基本系サイクル \* factor),  
n-setSm. ... 速度セットポイントフィルタ (MD 1500 ... 1521)

メモリ (MD 38010) の再割当てを使用すると、次のマシンデータを保存することができます：

MMC100:

"Services", "Data OUT", "Start-up data, NCK data" ならびに、プログラムしてある場合には、測定系誤差およびビームの垂れ／角度補正テーブルを、PCIN を介して保存し、Power On-Reset を実行して、PCIN および "Data IN" を使って保存済みデータに読み込みます (= シリーズマシンスタートアップ)。

MMC102/103:

"SERIES START-UP" ならびに、プログラムしてある場合は、測定系誤差およびビームの垂れ／角度補正テーブルを保存し、Power On-Reset を実行して、  
"Start-up" アーカイブに読み込みます (保存済みデータは再ロードされます)。

ディスク "MMC100 TOOLBOX" に入っているプログラムを NC にコピーします (アーカイブを使ってください)。

QECDAT.MPF

QECSTART.MPF

QECLERNP.SPF ("Polynomial" 学習プログラム) あるいは QECLRNC.SPF ("Circle" 学習プログラム) が NC 上に QECLRNC.SPF として格納されます。

次のプログラムを使用します：

- パートプログラム QECDAT 中では

N1040 def int numAxes=... 学習する軸数を入力します。

N1150 axisName[0] 第 1 軸の軸名を入力します。

N1160 axisName[1] 第 2 軸の軸名を入力します。

N1170 axisName[2] 第 3 軸の軸名を入力します。

N1180 axisName[3] 第 4 軸の軸の名前を入力します。

("Circle" 学習プログラムの軸名には、AX1 .. AX8 あるいはマシン軸またはチャンネル軸名を私用することができます。しかし、"Polynomial" 学習プログラムには、チャンネル軸名しか使用できません。)

- パートプログラム QECSTRAT では

N1080 def int numAxes=... 学習する軸数を入力します。

N1310 axisName[0] 第 1 軸の軸名を入力します。

N1320 axisName[1] 第 2 軸の軸名を入力します。

N1330 axisName[2] 第 3 軸の軸名を入力します。

N1340 axisName[3] 第 4 軸の軸名を入力します。

("Circle" 学習プログラムの軸名には、AX1 ～ AX8 あるいはマシン軸または



チャンネル軸名を使用することができます。しかし、"Polynomial" 学習プログラムには、チャンネル軸名しか使用できません。）

## LEARN プロセスの実行

QECDAT を選択し、実行します

システム変数が割当てられます。

QECSTART を選択します：オーバライド 100% で、スタートします。

学習プログラムが約 15 分間走り、約 30 cm 移動動作を行います。メッセージ "REORG not possible" が表示された場合には、無視してください。このメッセージは約 10 秒間表示されます。メッセージが消えると、学習プロセスの移動動作が続行されます。

## QEC をアクティブにする

マシンデータ：	デフォルト	次に変更	意味
MD 32500 FRIC_COMP_ENABLE	0	1	"Friction compensation active"（摩擦補正アクティブ）をオンにします。

"Circularity test" を使って結果を確認してください。

補正データを保存します（"SERIES START-UP" を使用した場合は、QEC データはバックアップされません）：

MMC100: SERVICES/Data/Circle error compensation/All に PCIN を使って保存します。

MMC102/103: SERVICES\NCK\NC Active Data/Quadrant Error Co/Quadrant error comp-complete.ini. にファイルを保存します。このファイルに補正がすべて格納されます。

（注）SERVICES "System settings" "for display" で、"displayed name length" を "20" に変更し、名前全体が見えていることを確認します。

## ■ 重力補正

### 前提条件

#### 重力補正のない場合の応答

重力補正がプログラミングされていない、負荷のかかる軸の場合、システムは次のような応答をします：

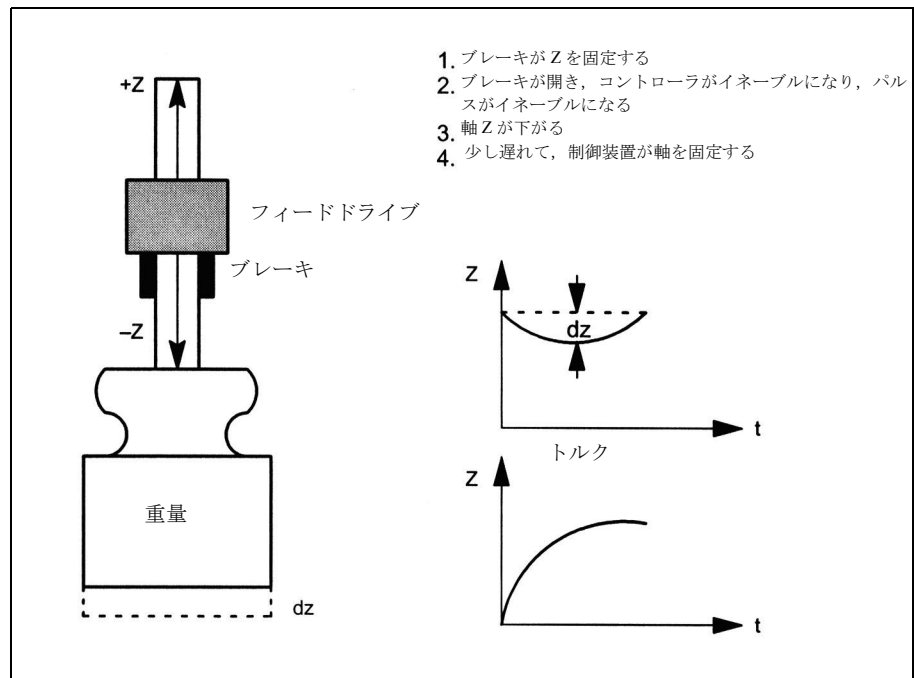


図 2.56 重力補正のない場合の垂直軸の降下

ブレーキが開いたときに軸 ( $Z$ ) が降下することは望ましいことではありません。SIMODRIVE 611D-MD 1409: SPEEDCTRL\_INTEGRATOR\_TIME\_1 を使って設定したリセットタイムが大きいほど、軸の降下量は増えます。重力補正機能をアクティブにすることで、軸の降下を最小限にとどめることができます。

#### 重力補正がある場合の応答

重力補正機能を使うと、制御装置の電源を入れたときに負荷のかかった垂直軸が降下することを防止できます。この機能の働きを図 2.57 に示します。

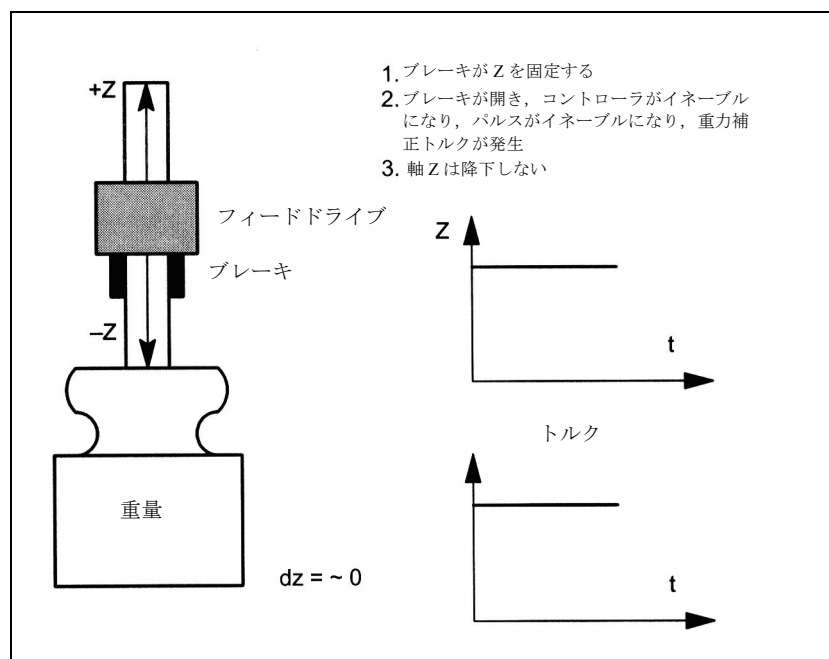


図 2.57 重力補正のある垂直軸の降下

## 固定停止点に対する移動との相互作用

"Electronic weight compensation" (重力補正) 機能と "Traverse against fixed stop" (固定停止点に対する移動) 機能は併用することができますが、次の点に注意する必要があります：

重力補正を使用して、固定停止点トルクまたは固定停止点力についてゼロ点をオフセットすることはできません。重力補正はこの目的に適していません。

- 例えば 40% 固定停止点トルクがプログラミングされていて、同じ方向において軸に 30% の重力補正が必要な場合、固定停止点にかかる軸圧力の実際トルクは定格トルクの 10% にしかありません。
- 40% 固定停止点トルクが上記と同じ状況で逆の方向（重力補正方向とは反対の方向、すなわち軸が降下する方向）にプログラミングされている場合、固定停止点にかかる軸圧力の実際トルクは定格トルクの 70% になります。
- 例えば、軸が 30% の重力補正を必要とする場合、30% 未満の停止点トルクがプログラミングされている場合は固定停止点にアプローチすることはできません。この場合、ドライブトルクが制限されてしまうので、制御装置は軸を固定することができず、軸が降下してしまいます。

垂直軸を使用した固定停止点移動機能のこれらの特性は、ドライブのトルク制限オプションを利用して決定することができます。これらの特性は、重力補正機能によって改善されることもなければ悪影響を受けることもありません。

## 起動

この機能は、軸別の MMD 32460 TORQUE\_OFFSET をゼロ以外の値にセットするとアクティブになりますが、次の RESET または POWER ON が実行されるか、あるいは "Set MD active" ソフトキーが選択されて初めて動作可能になります。

---

## 停止

この機能は、軸別の axis-specific MD 32460: TORQUE\_OFFSET をゼロにセットすると選択解除されますが、次の RESET または POWER ON が実行されるか、あるいは "Set MD active" ソフトキーが選択されて初めて選択が解除されます。

---

## 2.5.3 補足条件

### ■ 実用性

補正のタイプ：

- バックラッシュ補正
- リードスクリュー誤差および測定系補正
- 多次元直角度補正
- 手動象現突起補正
- 自動象現突起補正（ニューラルネットワーク）
- 温度補正
- アナログ速度セットポイントの自動ドリフト補正
- 重力補正

#### "Backlash compensation"（バックラッシュ補正）機能

この機能は標準で利用できます。

#### "Leadscrew error and measuring system error compensation"（リードスクリュー誤差および測定系補正）機能

この機能は標準で利用できます。

#### "Multi-dimensional beam sag compensation"（多次元直角度補正）機能

この機能はオプションです。

#### "Quadrant error compensation by operator input"（オペレータ入力による象現突起補正）機能

この機能は標準で利用できます。

#### "Automatic quadrant error compensation"（自動象現突起補正）機能

この機能は標準で利用できます。

#### "Temperature compensation"（温度補正）機能

この機能はオプションです。

#### "Electronic weight compensation"（重力補正）機能

この機能は標準で利用できます。

---

## 2.5.4 データの説明 (MD, SD)

### ■ マシンデータの説明

#### ■ 一般マシンデータ

10082 MD 番号	CTRLOUT_LEAD_TIME セットポイント転送時間のシフト	
デフォルト値 : 0.0	入力の下限 : 0.0	入力の上限 : 100.0
電源オン後, 変更は有効	保護レベル : 2 / 7	単位 : %
データタイプ : DOUBLE	適用ソフトウェアバージョン : 2 以降	
意味 :	<p>速度セットポイントの出力リードタイム入力された値が大きいほど, ドライブの速度セットポイント転送速度が早くなる。</p> <ul style="list-style-type: none"> <li>・ 0% セットポイントが, 次の位置制御サイクルの開始時に転送される。</li> <li>・ 50% セットポイントが, 位置制御サイクルの 1/2 が実行された時点ですでに転送されている。</li> </ul> <p>妥当なリードタイムは, 最大位置制御計算時間を測定して求めること。 MD 10083: CTRLOUT_LEAD_TIME_MAX は, 制御が測定した値を示す。これが正味値であるため, ユーザは安全公差を設ける必要がある (例えば 5%)。大きすぎるリードタイムが入力された場合, ドライブアラーム 300506 が出力される。入力値は, ドライブの次の速度コントローラパルスレートに丸められる。ドライブの速度コントローラパルスレートが異なる場合, その値を変更しても設定したドライブ全てに対してコントローラを同程度に改善できるとは限らない。</p> <p>注記: この MD はデジタルドライブを持つ軸に関してのみ重要。</p>	
関連性	MD 10083: CTRLOUT_LEAD_TIME_MAX	

10083 MD 番号	CTRLOUT_LEAD_TIME_MAX セットポイント転送時間のシフトに関する最大許容設定	
デフォルト値 : 100.0	入力の下限 : 0.0	入力の上限 : 100.0
NEW_CONF 後, 変更は有効	保護レベル : 2 / 7	単位 : %
データタイプ : DOUBLE	適用ソフトウェアバージョン : 4 以降	
意味 :	<p>速度セットポイント出力の最大許容リードタイム MD 10083 は MD 10082 の設定補助となる。表示された値は, 安全公差を考慮し, 直接 MD 10082 に転送される。</p> <p>許容リードタイムは, 位置コントローラに必要な最大測定計算時間から求める。位置コントローラの計算時間要求事項が増加すると許容リードタイムは減少する。</p> <p>MD 10060 または 10050 を介して位置制御サイクル低減することにより, 許容リードタイムを低減することができる。</p> <p>リードタイムは運転寿命期間中すべて測定される。</p> <p>表示値は手動入力で増すことができる。</p> <p>指定したリードタイムが許容リードタイム (例えば, 100%) より大きい場合, 改めて自動的に確定されます。</p> <p>注記: この MD はデジタルドライブを持つ軸に関してのみ重要。</p>	
関連性	MD 10050: SYSCLOCK_CYCLE_TIME      基本システムクロック周波数 MD 10060: POSCTRL_SYSCLOCK_TIME_RATIO      位置制御サイクルの係数 MD 10082: CTRLOUT_LEAD_TIME	

18342 MD 番号	MM_CEC_MAX_POINTS[t] 直角度補正 [table t] の補間点の最大数	
デフォルト値: 0	入力の下限: 0	入力の上限: 2000
電源オン後, 変更は有効	保護レベル: 2 / 4	単位: -
データタイプ: DWORD	適用ソフトウェアバージョン: 2.1 以降	
意味:	<p>直角度補正では, それぞれの補正テーブル [t] に対して必要となる補間点の数を定義する必要がある。  [t] = 補正テーブルのインデックス (<math>0 \leq t \leq 2 * \text{最大軸数}</math>) つまり,  t = 0: 1. 補正テーブル  t = 1: 2. 補正テーブル  など次の定義パラメータを使って, 必要数を求める  (■「直角度補正および角度誤差補正」を参照)</p> $\text{MM\_CEC\_MAX\_POINTS}[t] = \frac{\text{\$AN\_CEC\_MAX}[t] - \text{\$AN\_CEC\_MIN}[t]}{\text{\$AN\_CEC\_STEP}[t]} + 1$ <p> \$AN_CEC_MIN [t] 初期位置 (システム変数)  \$AN_CEC_MAX [t] 終了位置 (システム変数)  \$AN_CEC_STEP [t] 補間点間の距離 (システム変数)  補間点の数およびその間の距離を選択する場合, 補正テーブルの大きさの演算結果および非揮発性ユーザメモリの容量の演算結果に注意すること。各補正值 (補間点) に 8 バイト必要である。  値 0 が入力された場合, テーブルにメモリは予約されない; すなわち, テーブルは存在せず, したがって機能は起動できない。 </p>	
例外, エラー	<p>注意  !MD 18342: MM_CEC_MAX_POINTS[t] が変更されると, パワーオンで非揮発性 NC ユーザメモリが自動的に再割り当てされる。非揮発性ユーザメモリのユーザデータ (例えば, ドライブおよび MMC マシンデータ, ツールオフセット, パートプログラムなど) は全て削除される。</p>	
関連性	SD 41300: CEC_TABLE_ENABLE[t] 直角度補正テーブル [t] の評価をイネーブルにする	
参照	2.14 メモリ構成 (S7)	



## ■ 軸別マシンデータ

32450 MD 番号	BACKLASH[n] バックラッシュ	
デフォルト値: 0	入力の下限: ***	入力の上限: ***
NEW_CONF 後, 変更は有効	保護レベル: 2	単位: mm または度
データタイプ: DOUBLE	適用ソフトウェアバージョン: 1.1 以降	
意味:	移動の正方向と負方向のバックラッシュ補正值入力は ・エンコーダがマシンパートの前にある場合 (通常の場合), 正の数 ・エンコーダがマシンパートの後にある場合, 負の数, ゼロを入力すると, 補正は停止する。全てのモードで, 基準点ア プローチ後必ずバックラッシュ補正はアクティブになる。 インデックス [n] には次のコーディングが含まれる: [encoder no.]: 0 または 1	
例外, エラー .....	2 次測定系がある場合, その測定系にも別個のバックラッシュ値を入力する必 要がある。	
関連性 ....	MD: NUM_ENC                      測定系の数 MD: ENC_CHANGE_TOL    位置実際値スイッチオーバーの最大公差	

32460 MD 番号	TORQUE_OFFSET 電子重量補正用の追加トルク	
デフォルト値: 0	入力の下限: -100	入力の上限: 100
NEW_CONF 後, 変更は有効	保護レベル: 2 / 7	単位: %
データタイプ: DOUBLE	適用ソフトウェアバージョン: 3.1 以降	
意味:	電子重量補正の追加トルクは, 静的トルク (MD1113 x MD1118 で求める) の % ブロックに入力される。電流コントローラがアクティブになると直に有効 となる。したがってコントローライネーブル信号がセットされたときに垂直 軸が降下しない (特に速度コントローラリセット時間設定が高い場合)。 100% が軸ドライブの定格トルクに相当する。 速度コントローラが停止した場合, 正の値でドライブは正の方向へ移動する (詳細については, MD 32100: AX_MOTION_DIR を参照のこと)。 したがって, 正の移動方向が上向き (軸は上昇) の場合, 加重補正には正の 値を入力すること。それとは逆に, 正の移動方向が下向きの場合, 負の値が 必要となる。	
例外, エラー	"Traverse against fixed stop" (固定停止点に対する移動) 機能との相互作用を参 照のこと。	
関連性		

32490 MD 番号	FRICT_COMP_MODE 摩擦補正モード	
デフォルト値: 0	入力の下限: 0	入力の上限: 2
NEW_CONF 後, 変更は有効	保護レベル: 2/4	単位: -
データタイプ: BOOLEAN	適用ソフトウェアバージョン: 2.1 以降	
意味:	0: 摩擦補正なし 1: 一定フィードフォワード値または適用可能特性を持つ摩擦補正 2: ニューラルネットワークを介して学習した特性を持つ摩擦補正	
関連性		

32500 MD 番号	FRICT_COMP_ENABLE 摩擦補正アクティブ	
デフォルト値: 0	入力の下限: 0	入力の上限: 1
NEW_CONF 後, 変更は有効	保護レベル: 2 / 4	単位: -
データタイプ: BOOLEAN	適用ソフトウェアバージョン: 1.1 以降	
意味:	<p>1: 軸は, "friction compensation" (摩擦補正值) がイネーブルされ, 摩擦補正值が注入される。</p> <p>円輪郭の象現突起は, "friction compensation" で補正することができる。</p> <p>軸マシンデータ MD 32490: FRICT_COMP_MODE "friction compensation type" (摩擦補正のタイプ) で, "friction compensation with constant injected value" (一定注入値を使用した摩擦補正) または "quadrant error compensation with neural networks" (ニューラルネットワークを使用した象現突起補正) のいずれを選択するべきか決定される。</p> <p>ニューラルネットワークの場合, 有効特性がすでに学習されている場合, マシンデータは最初 "1" に設定する必要がある。学習フェーズ中に, このマシンデータの定数の補正值が別個に注入される。</p> <p>0: この軸に対して "Friction compensation" はイネーブルされていない。摩擦補償値は注入されない。</p>	
関連性	MD 32490: FRICT_COMP_MODE                      摩擦補正タイプ MD 32510: FRICT_COMP_ADAPT_ENABLE          摩擦補正適応アクティブ MD 32520: FRICT_COMP_CONST_MAX            最大摩擦補正值 MD 32540: FRICT_COMP_TIME                  摩擦補正時定数 MD 38010: MM_QEC_MAX_POINTS              ニューラルネットを使用したワーク 象現突起補正の補間点の数	

32510 MD 番号	FRICT_COMP_ADAPT_ENABLE [n] 摩擦補正適応アクティブ [セットポイントブランチ]: 0	
デフォルト値: 0	入力の下限: 0	入力の上限: 1
NEW_CONF 後, 変更は有効	保護レベル: 2	単位: -
データタイプ: BOOLEAN	適用ソフトウェアバージョン:	
意味:	<p>1: この軸に対して振幅適応を使用した摩擦補正はイネーブルでない。摩擦補償を使い, 円輪郭の現象突起を補正することができる。多くの場合, 摩擦補正值の注入振幅は振幅範囲全域に対して一定ではない。その場合, 最高の摩擦補正を得るために, 高加速には低加速よりも小さな補正值を注入する必要がある。適応曲線のパラメータ (図 2.35 参照) を求め, マシンデータに入力すること。</p> <p>0: この軸に対して, 振幅適応を使用した摩擦補正はイネーブルでなければならない</p>	
この MD は次の場合は無効	MD 32500: FRICT_COMP_ENABLE = 0 MD 32490: FRICT_COMP_MODE = 2 (ニューラル QEC)	
関連性	MD 32500: FRICT_COMP_ENABLE                      摩擦補正アクティブ MD 32520: FRICT_COMP_CONST_MAX            最大摩擦補正值 MD 32530: FRICT_COMP_CONST_MIN            最小摩擦補正值 MD 32550: FRICT_COMP_ACCEL1                適応加速値 1 MD 32560: FRICT_COMP_ACCEL2                適応加速値 2 MD 32570: FRICT_COMP_ACCEL3                適応加速値 3 MD 32540: FRICT_COMP_TIME                  摩擦補正時定数	

32520 MD 番号	FRICT_COMP_CONST_MAX [n] 最大摩擦補正值 [ セットポイントブランチ ]: 0	
デフォルト値: 0	入力の下限: 0	入力の上限: プラス
NEW_CONF 後, 変更は有効	保護レベル: 2	単位: mm/min
データタイプ: DOUBLE	適用ソフトウェアバージョン:	
意味:	マシンデータ MD 32520: FRICT_COMP_CONST_MAX を使って, 摩擦補正值の (最大) 注入振幅の大きさを求める。適応を使用しない摩擦補正の全加速範囲に, この値を注入する。 適応を使用した摩擦補正では, この値は適応曲線のセクション B2 にのみ注入する (図 2.35 参照)。	
この MD は次の場合は無効	MD 32500: FRICT_COMP_ENABLE = 0 MD 32490: FRICT_COMP_MODE = 2 ( ニューラル QEC)	
関連性	MD 32500: FRICT_COMP_ENABLE            摩擦補正 MD 32510: FRICT_COMP_ADAPT_ENABLE    摩擦補正 MD 32530: FRICT_COMP_CONST_MIN        最小摩擦補正值 MD 32550: FRICT_COMP_ACCEL1          適応加速値 1 MD 32560: FRICT_COMP_ACCEL2          適応加速値 2 MD 32570: FRICT_COMP_ACCEL3          適応加速値 3 MD 32540: FRICT_COMP_TIME            摩擦補正時定数	

32530 MD 番号	FRICT_COMP_CONST_MIN [n] 最小摩擦補正值 [ セットポイントブランチ ]: 0	
デフォルト値: 0	入力の下限: 0	入力の上限: プラス
NEW_CONF 後, 変更は有効	保護レベル: 2	単位: mm/min
データタイプ: DOUBLE	適用ソフトウェアバージョン: 1.1 以降	
意味:	"Friction compensation with adaptation" (摩擦補正適応) がアクティブになっている (MD 32510: FRICT_COMP_ADAPT_ENABLE = 1) 場合にのみ, 最小摩擦補正值が必要となる。 FRICT_COMP_CONST_MIN に設定されている摩擦補正值振幅が, 適応特性の加速範囲 B4 ( $a \geq a_3$ ) に注入される (図 2.35 参照)。	
この MD は次の場合は無効	MD 32510: FRICT_COMP_ADAPT_ENABLE = 0 MD 32490: FRICT_COMP_MODE = 2 ( ニューラル QEC)	
例外	例外として, FRICT_COMP_CONST_MIN にプログラミングされた値は, MD 32520: FRICT_COMP_CONST_MAX に設定された値より大きくてもよい。	
関連性	MD 32500: FRICT_COMP_ENABLE            摩擦補正アクティブ MD 32510: FRICT_COMP_ADAPT_ENABLE    摩擦補正適応アクティブ MD 32520: FRICT_COMP_CONST_MAX        最大摩擦補正值 MD 32550: FRICT_COMP_ACCEL1          適応加速値 1 MD 32560: FRICT_COMP_ACCEL2          適応加速値 2 MD 32570: FRICT_COMP_ACCEL3          適応加速値 3 MD 32540: FRICT_COMP_TIME            摩擦補正時定数	

32540 MD 番号	FRICT_COMP_TIME [n] 摩擦補正時定数 [ セットポイントブランチ ]: 0	
デフォルト値: 0,015	入力の下限: 0	入力の上限: プラス
NEW_CONF 後, 変更は有効	保護レベル: 2	単位: s
データタイプ: DOUBLE	適用ソフトウェアバージョン: 1.1 以降	
意味:	摩擦補正値が注入される時定数 (補正セットポイントパルスの崩壊時間) 象現突起遷移時のずれは注入振幅の影響を受けるだけでなく, 摩擦補正時定数の変更によっても影響を受ける (2.5.2 「 <span style="background-color: black; color: black;">■</span> 従来の摩擦補正のスタートアップ」)。	
この MD は次の場合は無効	MD 32500: FRICT_COMP_ENABLE = 0	
関連性	MD 32500: FRICT_COMP_ENABLE      摩擦補正アクティブ MD 32520: FRICT_COMP_CONST_MAX      最大摩擦補正値	

32550 MD 番号	FRICT_COMP_ACCEL1 [n] 適応加速値 1 [ セットポイントブランチ ]: 0	
デフォルト値: 0	入力の下限: 0	入力の上限: プラス
NEW_CONF 後, 変更は有効	保護レベル: 2	単位: m/s <sup>2</sup>
データタイプ: DOUBLE	適用ソフトウェアバージョン: 1.1 以降	
意味:	"Friction compensation with adaptation" (摩擦補正適応) がアクティブの場合のみ, 適応加速値が必要である。 適応加速値 1 ~ 3 が, 適応曲線を決める補間点である。適応曲線は, 異なる摩擦補正値が適用される 4 つのセクションに更に分割される。 FRICT_COMP_ACCEL1 (a <sub>1</sub> ) はセクション B1 を定義する (図 2.36 参照)。 セクション B1 内の注入振幅には, 次が適用される: $\Delta n = \Delta n_{\max} * a / a_1 \text{ for } a < a_1$	
この MD は次の場合は無効	MD 32510: FRICT_COMP_ADAPT_ENABLE = 0 MD 32490: FRICT_COMP_MODE = 2 (ニューラル QEC)	
関連性	MD 32500: FRICT_COMP_ENABLE      摩擦補正アクティブ MD 32510: FRICT_COMP_ADAPT_ENABLE      摩擦補正適応アクティブ MD 32520: FRICT_COMP_CONST_MAX      最大摩擦補正 MD 32530: FRICT_COMP_CONST_MIN      最小摩擦補正 MD 32560: FRICT_COMP_ACCEL2      適応加速値 2 MD 32570: FRICT_COMP_ACCEL3      適応加速値 3 MD 32540: FRICT_COMP_TIME      摩擦補正時定数	

32560 MD 番号	FRICT_COMP_ACCEL2 [n] 適応加速値 2 [セットポイントブランチ]: 0	
デフォルト値: 0	入力の下限: 0	入力の上限: プラス
NEW_CONF 後, 変更は有効	保護レベル: 2	単位: m/s <sup>2</sup>
データタイプ: DOUBLE	適用ソフトウェアバージョン: 1.1 以降	
意味:	<p>"Friction compensation with adaptation" (摩擦補正適応) がアクティブの場合にのみ, 適応加速値が必要である。</p> <p>適応加速値 1 ~ 3 が, 適応曲線を定める補間点である。適応曲線は, 異なる摩擦補正値が適用される 4 つのセクションに更に分割される。</p> <p>セクション B2 は, MD 32550: FRICT_COMP_ACCEL1 (a<sub>1</sub>) および FRICT_COMP_ACCEL2 (a<sub>2</sub>) によって定義される (図 2.36 参照)。</p> <p>セクション B2 内の注入振幅には, 次が適用される:</p> $\Delta n = \Delta n_{\max} \text{ for } a_1 \leq a \leq a_2$	
この MD は次の場合は無効	MD 32510: FRICT_COMP_ADAPT_ENABLE = 0 MD 32490: FRICT_COMP_MODE = 2 (ニューラル QEC)	
関連性	MD 32500: FRICT_COMP_ENABLE      摩擦補正アクティブ MD 32510: FRICT_COMP_ADAPT_ENABLE      摩擦補正適応アクティブ MD 32520: FRICT_COMP_CONST_MAX      最大摩擦補正 MD 32530: FRICT_COMP_CONST_MIN      最小摩擦補正 MD 32550: FRICT_COMP_ACCEL1      適応加速値 2 MD 32570: FRICT_COMP_ACCEL3      適応加速値 3 MD 32540: FRICT_COMP_TIME      摩擦補正時定数	

32570 MD 番号	FRICT_COMP_ACCEL3 [n] 適応加速値 3 [セットポイントブランチ]: 0	
デフォルト値: 0	入力の下限: 0	入力の上限: プラス
NEW_CONF 後, 変更は有効	保護レベル: 2	単位: m/s <sup>2</sup>
データタイプ: DOUBLE	適用ソフトウェアバージョン: 1.1 以降	
意味:	<p>"Friction compensation with adaptation" (摩擦補正適応) がアクティブの場合にのみ, 適応加速値が必要である。</p> <p>適応加速値 1 ~ 3 が, 適応曲線を定める補間点である。適応曲線は, 異なる摩擦補正値が適用される 4 つのセクションに更に分割される。</p> <p>セクション B3 は, MD 32560: FRICT_COMP_ACCEL2 (a<sub>2</sub>) および FRICT_COMP_ACCEL3 (a<sub>3</sub>) によって定義される (図 2.36 参照)。</p> <p>セクション B2 内の注入振幅には, 次が適用される:</p> $\Delta n = \Delta n_{\max} * (1 - (a - a_2) / (a_3 - a_2))$ <p>ここで a<sub>2</sub> &lt; a &lt; a<sub>3</sub> 加速値が &gt; a<sub>3</sub> の場合はセクション B4 が適用される。</p> <p>セクション B3 内の注入振幅には, 次が適用される:</p> $\Delta n = \Delta n_{\min} \text{ for } a \geq a_3$	
この MD は次の場合は無効	MD 32510: FRICT_COMP_ADAPT_ENABLE = 0 MD 32490: FRICT_COMP_MODE = 2 (ニューラル QEC)	
関連性	MD 32500: FRICT_COMP_ENABLE      摩擦補正アクティブ MD 32510: FRICT_COMP_ADAPT_ENABLE      摩擦補正適応アクティブ MD 32520: FRICT_COMP_CONST_MAX      最大摩擦補正 MD 32530: FRICT_COMP_CONST_MIN      最小摩擦補正 MD 32550: FRICT_COMP_ACCEL1      適応加速値 2 MD 32560: FRICT_COMP_ACCEL2      適応加速値 3 MD 32540: FRICT_COMP_TIME      摩擦補正時定数	

32580 MD 番号	FRICT_COMP_INC_FACTOR ショートトラバース動作を使用した摩擦補正値の加重係数	
デフォルト値 : 1	入力の下限 : 0	入力の上限 : 100
電源オン後, 変更は有効	保護レベル : 2 / 4	単位 : %
データタイプ : DOUBLE	適用ソフトウェアバージョン : 2.1 以降	
意味 :	<p>補正が起動されていて軸位置決め動作が短い場合, 真円試験によって求められた最適摩擦補正値がこの軸の過剰補正を引き起こすことがある。</p> <p>このような場合, 摩擦補正値 (従来の誤差補正またはニューラルネットワークを使用した象現突起補正) の振幅の削減, ならびに制御の補間サイクル内に作成された位置決め動作を全て削減することにより, 設定を改善することができる。</p> <p>入力する必要がある係数は経験的に求めることができるが, またさまざまな摩擦条件のために軸ごとに異なることがある。入力範囲は真円試験で求めた値の 0 ~ 100% である。</p> <p>初期設定 0; 短い移動動作に対する補正は行われない。</p>	
関連性	MD 32500: FRICT_COMP_ENABLE      摩擦補正アクティブ	

32610 MD 番号	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	
デフォルト値 : 1	入力の下限 : 0	入力の上限 : 正の数
NEW_CONF 後, 変更は有効	保護レベル : 2	単位 : 係数
データタイプ : DOUBLE	適用ソフトウェアバージョン : 1.1 以降	
意味 :	<p>速度フィードフォワード制御では, 速度コントローラの入力時に, 速度セットポイントが直接注入される (図 2.34 参照)。この追加セットポイントは, 係数 (フィードフォワード制御係数と呼ばれる) を使って加重することができる。</p> <p>確実に速度フィードフォワード制御を適切に設定するために, 速度制御ループの等価時定数を正確に求め, MD 32810: EQUIV_SPEEDCTRL_TIME に入力すること。</p> <p>速度制御ループの等価時定数が正確に確定された場合, フィードフォワード制御係数の値は約 1 になる。この場合, システムのずれは, ほぼゼロになる (操作エリア <b>Diagnosis</b> のサービス画面で確認すること)。</p> <p>フィードフォワード制御係数 0 が入力された場合, フィードフォワード制御は停止する。しかし, とにかく計算は実行されるので, MD: FFW_MODE = 0 を使用してフィードフォワード制御を停止する必要がある。</p>	
この MD は次の場合は無効	MD 32620: FFW_MODE = 0 または 2	
関連性	MD 32620: FFW_MODE MD 32630: FFW_ACTIVATION_MODE MD 32810: EQUIV_SPEEDCTRL_TIME	

32620 MD 番号	FFW_MODE フィードフォワード制御モード	
デフォルト値 : 1	入力の下限 : 0	入力の上限 : 2
RESET 後, 変更は有効	保護レベル : 0/0	単位 : -
データタイプ : Byte	適用ソフトウェアバージョン : 4.3 以降	
意味 :	<p>FFW_MODE は, 軸別に適用するフィードフォワード制御モードを決定する。  0 = フィードフォワード制御なし  1 = 速度フィードフォワード制御  2 = トルクフィードフォワード制御  FFWON や FFWOF を使用し, すべての軸の特定のチャンネルに対するフィードフォワード制御を起動する, または停止する。  フィードフォワード制御が個々の軸に関するこれらの命令による影響を受けないように, マシンデータ FFW_ACTIVATION_MODE に, 常に起動, または常に停止, を設定することができる (セクション FFW_ACTIVATION_MODE も参照のこと)。  グローバルオプションデータ \$ON_FFW_MODE_MASK は, 入力を 1 または 2 に限定する。  フィードフォワード制御モード (速度フィードフォワード制御またはトルクフィードフォワード制御) が選択されている場合, パートプログラムでフィードフォワード制御の起動や停止を行うかどうかともまた, MD 32630: FFW_ACTIVATION_MODE でプログラミングすることができる。  オプションのトルクフィードフォワード制御をイネーブルにする必要がある。</p>	
アプリケーション	高速のパスで高い機械加工精度を得るため, フィードフォワード制御を使用して, 追従誤差による輪郭誤差をなくすることができる。	
関連性	MD 32630: FFW_ACTIVATION_MODE MD 32610: VELO_FFW_WEIGHT MD 32650: AX_INERTIA	

32630 MD 番号	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	
デフォルト値 : 1	入力の下限 : 0	入力の上限 : 1
RESET 後, 変更は有効	保護レベル : 0/0	単位 : -
データタイプ : Byte	適用ソフトウェアバージョン : 4.3 以降	
意味 :	<p>FFW_ACTIVATION_MODE を設定して, この軸/スピンドルに対するフィードフォワード制御のオンやオフをパートプログラムで行うかどうかを決定することができる。  0 = ハイレベル言語エレメント FFWON や FFWOF を使ってフィードフォワード制御のスイッチオンやオフを行うことはできない。MD: FFW_MODE で設定された条件は軸/スピンドルについて常にアクティブ。  1 = FFWON や FFWOF を使用してフィードフォワード制御のスイッチオン/オフを行うことができる。デフォルト設定は, チャンネル別データの MD 20150: GCODE_RESET_VALUES にプログラミングされている。この設定は, 最初の NC ブロックが実行される前でも有効である。  アクティブになる最終条件は, Reset 後でもアクティブなままである (したがって, JOG は使用する)。FFWON や FFWOF により, チャンネルのすべての軸に対するフィードフォワード制御のオン/オフを行うため, MD:FFW_ACTIVATION_MODE は, 互いに補間し合う軸に対しては同一設定値となるはずである。</p>	
関連性	MD 32620: FFW_MODE MD 20150: GCODE_RESET_VALUES	
参照	YS840DI ユーザーズマニュアル・プログラミング編	

32640 MD 番号	STIFFNESS_CONTROL_ENABLE 動的剛性制御の起動	
デフォルト値 : 0	入力の下限 : 0	入力の上限 : 1
NEW_CONF 後, 変更は有効	保護レベル : 2 / 7	単位 : -
データタイプ :	適用ソフトウェアバージョン : 4.1 以降	
意味 :	<p>ビットがセットされている場合、動的剛性制御を起動する。動的剛性制御がアクティブになると、より大きなサーボゲイン係数を使うことができる (MD 32200: POSCTRL_GAIN)。</p> <p>ドライブにおける計算負荷がさらに高くなるため、ドライブのスキャンサイクル (電流／ドライブモジュロサイクル) の設定値を適応しなければならない場合がある。</p> <p>1 軸ドライブモジュロに対しては、デフォルト値 (電流コントローラサイクル 125 <math>\mu</math>s, 速度コントローラサイクル 125 <math>\mu</math>s) で十分であるが、2 軸モジュロの場合、速度コントローラサイクルを (250 <math>\mu</math>s まで) 増さなければならない場合がある。</p> <p>注記 : 現在のところ、ドライブのインプリメンテーションにより、動的剛性制御はモータ測定系を使用した場合にのみ可能である。</p>	
関連性 ....		
参照		

32650 MD 番号	AX_INERTIA トルクフィードフォワード制御用の慣性モーメント	
デフォルト値 :	入力の下限 : 0	入力の上限 : プラス
NEW_CONF 後, 変更は有効	保護レベル : 2	単位 : kgm <sup>2</sup>
データタイプ : DOUBLE	適用ソフトウェアバージョン : 1.1 以降	
意味 :	<p>トルクフィードフォワード制御を使用すると、トルクに比例して追加電流セットポイントが電流コントローラの入力時に直接注入される (図 2-14 参照)。この値は、加速および慣性モーメントを使用して作成される。このため、電流制御ループの等価時定数を設定し、MD 32800: EQUIV_CURRCTRL_TIME に入力する必要がある。</p> <p>また、軸の総慣性モーメント (ドライブ + 負荷) を、AX_INERTIA (マシンのメーカから提供されたデータに基づくモータシャフトに関する総慣性モーメント) に入力すること。</p> <p>AX_INERTIA および MD 32800: EQUIV_CURRCTRL_TIME が正しく設定されると、加速中でも追従誤差はほぼゼロになる (サービス画面の "following error" で確認)。</p> <p>AX_INERTIA が 0 に設定されている場合、トルクフィードフォワード制御は停止するが、とにかく計算は実行されるので、MD 32620 FFW_MODE = 0 または 1 を使ってトルクフィードフォワード制御を絶対に停止しなければならない。</p>	
この MD は次の場合は無効	MD 32620: FFW_MODE = 0 または 1	
アプリケーション	動力学上の要求が厳しいときに高い輪郭精度を得るために、トルクフィードフォワード制御が必要となる。	
関連性	MD 32620: FFW_MODE MD 32630: FFW_ACTIVATION_MODE MD 32800: EQUIV_CURRCTRL_TIME	



32652 MD 番号	AX_MASS トルクフィードフォワード制御用の軸の質量	
デフォルト値: 0	入力の下限: 0	入力の上限: プラス
NEW_CONF 後, 変更は有効	保護レベル: 2 / 7	単位: kg
データタイプ: DOUBLE	適用ソフトウェアバージョン: 4.1 以降	
意味:	トルクフィードフォワード制御用の軸の質量。この MD は、直線ドライブ (DRIVE_TYPE=3) で、AX_INERTIA の代わりに使用できる	
関連性		
参照		

32700 MD 番号	ENC_COMP_ENABLE[n] LEC アクティブ [n]	
デフォルト値: 0	入力の下限: 0	入力の上限: 1
NEW_CONF 後, 変更は有効	保護レベル: 2 / 7	単位: -
データタイプ: BOOLEAN	適用ソフトウェアバージョン: 1.1 以降	
意味:	<p>1: LEC は、軸／測定系に対して起動される。LEC を使用すると、リードスクリュウ誤差や測定系誤差を補正することができる。この機能は、測定系が基準化されている (IS: "Referenced/ synchronized = 1") 場合にのみ使用できる。書き込み保護機能 (補正值) がアクティブ</p> <p>0: 補間補正は、軸／測定系に対してアクティブになっていない</p> <p>インデックス [n] には次のコーディングが含まれる:</p> <p>[encoder no.]: 0 または 1</p>	
関連性	MD: MM_ENC_COMP_MAX_POINTS 補間補正用の補間ポイント数 IS "Referenced/synchronized 1" IS "Referenced/synchronized 2"	

32710 MD 番号	CEC_ENABLE 直角度補正イネーブル	
デフォルト値: 0	入力の下限: 0	入力の上限: 1
NEW_CONF 後, 変更は有効	保護レベル: 2 / 4	単位: -
データタイプ: BOOLEAN	適用ソフトウェアバージョン: 2.0 以降	
意味:	<p>1: 軸に対してイネーブルになっている場合は、"Beam sag compensation" (直角度補正) 。"Beam sag compensation" を使うと、軸内部幾何学的誤差 (例えば、直角度および角度誤差) を補正することができる。この機能は、次の条件が満たされて初めて、制御装置でイネーブルになる:</p> <ul style="list-style-type: none"> <li>・オプション "Interpolatory compensation" がセットされている。</li> <li>・関連の補正テーブルが利用できる</li> <li>・必要な補正テーブルの評価がイネーブルである (SD: CEC_TABLE_ENABLE[t] = 1)</li> <li>・必要な位置測定系が基準化されている (IS "Referenced/synchronized" = 1)</li> <li>・書き込み保護機能 (補正值) がアクティブ</li> </ul> <p>0: "Beam sag compensation" が補正軸に対してイネーブルになっていない。</p>	
関連性	MD: MM_CEC_MAX_POINTS[t] 直角度補間点数 SD: CEC_TABLE_ENABLE[t] 直角度補正テーブル t の評価 IS "Referenced/synchronized 1 or 2" DB31-48, DBX60.4 または 60.5	

32711 MD 番号	CEC_SCALING_SYSTEM_METRIC 直角度補正の測定系	
デフォルト値 : 1	入力の下限 : 0	入力の上限 : 1
RESET 後, 変更は有効	保護レベル : 2 / 7	単位 : -
データタイプ : BOOLEAN	適用ソフトウェアバージョン : 5 以降	
意味 :	補正データの保存先 : ・ MD 32711=0: インチ系 ・ MD 32711=1: メートル系 同一軸に影響を与える直角度補正テーブル全てに測定系を構築することができる。 これによって, 全ての位置入力が構築された測定系の全軸補正計算値を使用して一度に補間される。 測定系の切り換え後は, 外部テーブル変換の必要はなくなる。 総軸補正值が明らかに位置のみに依存していて, 相互の関係で計算された個別のテーブルコンテンツには依存していないので, 測定系の軸設定が必要です。 注記 : MD 10260: CONVERT_SCALING_SYSTEM=1 の場合にのみ有効 (1.8 位置指令単位と検出単位 (G2) 参照)	
関連性	MD 10260: CONVERT_SCALING_SYSTEM 基本システム切換えアクティブ	

32720 MD 番号	CEC_MAX_SUM 直角度補正用の最大補正值	
デフォルト値 : 1	入力の下限 : 0	入力の上限 : 10
NEW_CONF 後, 変更は有効	保護レベル : 2 / 4	単位 : 直線軸 : mm 回転軸 : 度
データタイプ : DOUBLE	適用ソフトウェアバージョン : 2.1 以降	
意味 :	直角度補正では, 総補正值の絶対的な大きさ (全ての補正関係の補正值の合計) を, マシンデータ値 CEC_MAX_SUM を使って軸別に監視する。求めた総補正值が最大値より大きい場合, アラーム 20124 が発せられる。	
この MD は次の場合は無効	リードスクリュー誤差補正 バックラッシュ補正 温度補正	
関連性	MD: CEC_ENABLE 直角度補正イネーブル SD: CEC_TABLE_ENABLE[t] 直角度補正テーブル t の評価 IS "Referenced/synchronized 1 or 2" DB31-48, DBX60.4 or 60.5	

32730 MD 番号	CEC_MAX_VELO 直角度補正用の最大許容変化値	
デフォルト値: 10	入力の下限: 0	入力の上限: 100
NEW_CONF 後, 変更は有効	保護レベル: 2 / 4	単位: %
データタイプ: DOUBLE	適用ソフトウェアバージョン: 2.1 以降	
意味:	直角度補正では, 総補正值の修正 (全てのアクティブな補正関係の補正值の合計) が軸別に制限される。最大変更値は, MD 32000: MAX_AX_VELO (最大軸速度) のパーセントとしてこのマシンデータに設定される。 総補正值の変更が最大値よりも大きい場合, アラーム 20125 が出力されるが, プログラムは続行される。制限のためにカバーされなくなったパスは, 補正值が制限を受けなくなると直ちに構築される。	
この MD は次の場合は無効	リードスクリュエラー補正 バックラッシュ補正 温度補正	
関連性	MD: CEC_ENABLE                      直角度補正イネーブル MD: MAX_AX_VELO                    最大軸速度 SD: CEC_TABLE_ENABLE[t]          直角度補正テーブル t の評価 IS "Referenced/synchronized 1 or 2"      DB31-48, DBX60.4 or 60.5	

32750 MD 番号	TEMP_COMP_TYPE 温度補正タイプ	
デフォルト値: 0	入力の下限: 0	入力の上限: 3
電源オン後, 変更は有効	保護レベル: 2	単位: 16 進数
データタイプ: BYTE	適用ソフトウェアバージョン: 1.1 以降	
意味:	MD: TEMP_COMP_TYPE で, マシン軸に対して有効な温度補正タイプを起動する。次のタイプを区別すること: 0: 温度補正がアクティブでない 1: 位置依存補正アクティブ (SD: TEMP_COMP_ABS_VALUE を使用した補正值) 2: 位置依存補正アクティブ (SD: TEMP_COMP_SLOPE および SD: TEMP_COMP_REF_POSITION を使用した補正值) 3: 位置依存温度補正および位置独立温度補正アクティブ (タイプ 1 およびタイプ 2 に従った SD での補正值) オプションの温度補正がイネーブルになっていなければならない。	
関連性	SD: TEMP_COMP_ABS_VALUE          位置依存温度補正值 SD: TEMP_COMP_REF_POSITION      位置依存温度補正用の基準点 SD: TEMP_COMP_SLOPE              位置依存温度補正用の傾き MD: COMP_ADD_VELO_FACTOR        補正による速度違反	

32760 MD 番号	COMP_ADD_VELO_FACTOR 補正による速度違反	
デフォルト値 : 0.01	入力の下限 : 0	入力の上限 : 0.1
電源オン後, 変更は有効	保護レベル : 2	単位 : 係数
データタイプ : DOUBLE	適用ソフトウェアバージョン : 1.1 以降	
意味 :	<p>軸の MD: COMP_ADD_VELO_FACTOR を使用して, いずれかの IPO サイクル内の温度補正のために移動可能な最大距離が制限される。</p> <p>演算結果温度補正值が上に述べた最大値である場合, この値は複数の IPO サイクル上に移動される。ここでは, アラームはない。IPO サイクルごとの最大補正值が, 最大軸速度 (MD: MAX_AX_VELO) に関する係数となる。このマシンのデータを使用して, 温度補正の最大傾き <math>\tan \beta_{\max}</math> が制限される。最大傾き <math>\tan \beta_{\max}</math> の計算例 :</p> <p>1. 補間器閉ループの計算 (Description of Functions Velocities, Setpoint/Actual-Value Systems, Closed-Loop Control (G2) を参照のこと)。</p> <p>補間器閉ループ制御 = 基本システムクロックレート * 補間サイクルの係数  補間器閉ループ制御 = MD: SYSCLOCK_CYCLE_TIME *  MD: IPO_SYSCLOCK_TIME_RATIO</p> <p>例 : MD: SYSCLOCK_CYCLE_TIME = 0.004 [s]  MD: IPO_SYSCLOCK_TIME_RATIO = 3  ⇒ 補間器閉ループ制御 = 0.004 * 3 = 0.012 [s]</p> <p>2. 温度補正パラメータ <math>\Delta v_{T\max}</math> に加えられた変更により増加する最大速度の計算</p> <p><math>\Delta v_{T\max} = \text{MD: MAX\_AX\_VELO} * \text{MD: COMP\_ADD\_VELO\_FACTOR}</math></p> <p>例 : MD: MAX_AX_VELO = 10 000 [mm/min]  MD: COMP_ADD_VELO_FACTOR = 0.01  ⇒ <math>\Delta v_{T\max} = 10\,000 * 0.01 = 100</math> [mm/min]</p> <p>3. 補間サイクルごとの移動距離の計算</p> $S_I(\text{at } v_{\max}) = 10\,000 \times \frac{0.012}{60} = 2.0 \text{ [mm]}$ $S_T(\text{at } \Delta v_{T\max}) = 100 \times \frac{0.012}{60} = 0.02 \text{ [mm]}$ <p>4. <math>\tan \beta_{\max}</math> の計算</p> $\tan \beta_{\max} = \frac{S_T}{S_I} = \frac{0.02}{2} = 0.01$ <p>( COMP_ADD_VELO_FACTOR の値に対応する )  ⇒ <math>\beta_{\max} = \arctan 0.01 = 0.57</math> 度</p> <p>SD: TEMP_COMP_SLOPE の大きい方の値を用いて, 位置依存温度補正值の最大傾き (ここでは 0.57 度) が内部的に使用される。アラームはでない。</p> <p>注記 : 温度補正によって生じた付加的な速度違反を考慮して, 速度監視用の制限値を決める必要がある (MD: AX_VELO_LIMIT)。</p>	
この MD は次の場合は無効	TEMP_COMP_TYPE = 0, 直角度, LEC, バックラッシュ補正	

32760 MD 番号	COMP_ADD_VELO_FACTOR 補正による速度違反	
関連性	MD: TEMP_COMP_TYPE                    温度補正タイプ SD: TEMP_COMP_ABS_VALUE            位置依存温度補正 SD: TEMP_COMP_SLOPE                位置依存温度補正の傾き MD: MAX_AX_VELO                    最大軸速度 MD: AX_VELO_LIMIT                  速度監視用の制限値 MD: IPO_SYSCLOCK_TIME_RATIO      IPO サイクルに対する基本システムクロックレート MD: SYSCLOCK_CYCLE_TIME          基本システムクロックレート	

32800 MD 番号	EQUIV_CURRCTRL_TIME[n] 電流制御ループの等価時定数	
デフォルト値 : 0.0005	入力の下限 : 0	入力の上限 : プラス
NEW_CONF 後, 変更は有効	保護レベル : 2	単位 : s
データタイプ : DOUBLE	適用ソフトウェアバージョン : 1.1 以降	
意味 :	この時定数は閉ループ電流制御の等価時定数と等しくなければならない。 この値は, トルクフィードフォワード制御のパラメータ化ならびに動的追従誤差モデル (輪郭監視) の計算に使用される。 トルクフィードフォワード制御を適切に設定するために, 電流制御ループのステップ応答を使用して電流制御ループの等価時定数を正確に求めること。 インストールツールを使って設定プロセスを表示することができる。 インデックス [n] には次のコーディングが含まれる : [ 制御パラメータブロック数 ]: 0 ~ 5 ( 参照 :1.8 位置指令単位と検出単位 (G2) )	
関連性	MD: FFW_MODE                        フィードフォワード制御タイプ MD: AX_INERTION                    速度フィードフォワード制御の慣性モーメント MD: CONTOUR_TOL                   公差範囲輪郭監視	
参照	YS840DI API 取扱説明書	

32810 MD 番号	EQUIV_SPEEDCTRL_TIME[n] 速度制御ループの等価時定数	
デフォルト値 : 0.004	入力の下限 : 0	入力の上限 : プラス
NEW_CONF 後, 変更は有効	保護レベル : 2	単位 : s
データタイプ : DOUBLE	適用ソフトウェアバージョン : 1.1 以降	
意味 :	<p>この時定数は閉ループ電流制御の等価時定数と等しくなければならない。  この値は, トルクフィードフォワード制御のパラメータ化ならびに動的追従誤差モデル (輪郭監視) の計算に使用される。トルクフィードフォワード制御を適切に設定するために, 電流制御ループのステップ応答を使用して電流制御ループの等価時定数を正確に求めること。  インストールツールを使って設定プロセスを表示することができる。  インデックス [n] には次のコーディングが含まれる :  [ 制御パラメータブロック数 ] : 0 ~ 5  ( 参照 : 1.8 位置指令単位と検出単位 (G2))</p>	
関連性 ....	MD: FFW_MODE                      フィードフォワード制御タイプ MD: VELO_FFW_WEIGHT            速度フィードフォワード制御の慣性モーメント MD: CONTOUR_TOL                公差範囲輪郭監視	
参照	YS840DI API 取扱説明書	

38000 MD 番号	MM_ENC_COMP_MAX_POINTS[n] LEC (SRAM) 用の補間点数	
デフォルト値 : 0	入力の下限 : 0	入力の上限 : 5000
電源オン後, 変更は有効	保護レベル : 2	単位 : -
データタイプ : DWORD	適用ソフトウェアバージョン : 1.1 以降	
意味 :	<p>リードスクリーン誤差補正では, 測定系ごとに必要となる補間点数を設定する必要がある。</p> <p>この必要数は, 設定されたパラメータを使って次のように計算することができる。(2.5.2「<span style="background-color: black; color: black;">■</span>測定誤差補正 (MSEC)」を参照)</p> $\text{MD: MM\_ENC\_COMP\_MAX\_POINTS} = \frac{\$AA\_ENC\_COMP\_MAX - \$AA\_ENC\_COMP\_MIN}{\$AA\_ENC\_COMP\_STEP} + 1$ <p> \$AA_ENC_COMP_MIN 初期位置 (システム変数)  \$AA_ENC_COMP_MAX 終了位置 (システム変数)  \$AA_ENC_COMP_STEP 補間点同士の距離 (システム変数) </p> <p>補間点数および補間点同士の距離を選択する場合, 補正テーブル [table t] の演算結果の大きさならびにバックアップ NC ユーザメモリ (SRAM) 内の必要スペースに注意を要する。補正值 (補間点) 1 つにつき 8 バイト必要である。</p> <p>インデックス [n] には次のコーディングが含まれる : [encoder no.]: 0 または 1</p>	
例外, エラー	<p>注意 :</p> <p>MD: MM_ENC_COMP_MAX_POINTS に何らかの変更があった場合, バックアップ NC ユーザメモリは電源が入ると自動的に再割り当てされる。</p> <p>バックアップ NC ユーザメモリ内のデータが全て喪失される (例えば, パートプログラム, ツールオフセットなど)。アラーム 6020 "Machine data changed - memory reallocated" が出力される。</p> <p>利用可能な総メモリ容量が足りないため, NC ユーザメモリの再割り当てが行われなかった場合, アラーム 6000 "Memory allocation made with standard machine data" が出力される。</p> <p>この場合, NC ユーザメモリは標準マシンデータのデフォルト値を使用して再割り当てされる。</p> <p>参照 : 2.14 メモリ構成 (S7), アラーム一覧</p>	
関連性 ....	MD: ENC_COMP_ENABLE[n]      補間補正アクティブ	
参照	2.14 メモリ構成 (S7)	

38010 MD 番号	MM_QEC_MAX_POINTS ニューラルネットワークを使用した QEC 用の最大補正值数	
デフォルト値 : 0	入力の下限 : 0	入力の上限 : 1024
電源オン後, 変更は有効	保護レベル : 2 / 4	単位 : -
データタイプ : DWORD	適用ソフトウェアバージョン : 2.1 以降	
意味 :	<p>ニューラルネットワーク (QEC) を使用した象現突起補正では, 必要補正值数を補正するそれぞれの軸に対して入力する必要がある。</p> <p>設定されたパラメータを使用して次のように必要数を計算する (2.5.2 「<b>■</b> ニューラル QEC のパラメータ化」を参照) :</p> $\$MA\_MM\_QEC\_MAX\_POINTS \geq (\$AA\_QEC\_COARSE\_STEPS + 1) * \$AA\_QEC\_FINE\_STEPS$ <p><math>\\$AA\_QEC\_COARSE\_STEPS</math>    特性の粗量子化 (システム変数)</p> <p><math>\\$AA\_QEC\_FINE\_STEPS</math>      特性の微量子化 (システム変数)</p> <p>「位置依存」補正では, 値はこの計算結果の値より大きいかまたは 2 倍でなければならない。</p> <p>粗量子化か微量子化のいずれかを選択した場合, 補正テーブルの演算結果の大きさならびに非揮発性ユーザメモリ内に必要となるメモリに注意する必要がある。補正值 1 つに 4 バイト必要である。値 0 が入力された場合, テーブルにはメモリは予約されない ; すなわち, テーブルは存在せず, したがって機能は起動しない。</p>	
例外, エラー .....	<p>注意</p> <p>!MD: MM_QEC_MAX_POINTS が出力された場合, 非揮発性ユーザメモリはシステムのパワーオンで自動的に再割り当てされる。その際, 非揮発性ユーザメモリ内のユーザデータは全て喪失する (例えば, ドライブ, MMC マシンデータ, ツールオフセット, パートプログラムなど)。</p> <p>注記 :</p> <p>必要となる正確な補間点数は機能の最初のインストール時には正確にはわからないため, 初期は大きめの数を選択すること。特性の記録, 保存が行われると, 必要数まで低減することができる。再度パワーオンすると, 保存された特性を再ロードすることができる。</p>	
参照	2.14 メモリ構成 (S7)	



## ■ 設定データの説明

41300 MD 番号	CEC_TABLE_ENABLE 直角度補正テーブル [t] の評価をイネーブルにする	
デフォルト値 : 0	入力の下限 : 0	入力の上限 : 1
変更は直ちに有効	保護レベル : 7	単位 : -
データタイプ : BOOLEAN	適用ソフトウェアバージョン : 2.1 以降	
意味 :	<p>1: 補正テーブル [t] の評価がイネーブルである。補正テーブルが [t] = 補正テーブルのインデックスを使用して、例えば補正関係を設定する (MD: MM_CEC_MAX_POINTS 参照)。</p> <p>"beam sag compensation" (直角度補正) では、補正軸は複数のテーブルの影響を受ける。SD: CEC_TABLE_ENABLE[t] は、NC パートプログラムまたは PLC ユーザプログラムによって変更し、機械加工アプリケーションの総補正値を適用することができる (例えば、テーブルの切換え)。</p> <p>制御装置では、追従条件が満たされるまで補正はイネーブルにならない :</p> <ul style="list-style-type: none"> <li>・ オプションの "Interpolatory compensation" がセットされている</li> <li>・ 割当てられた補正テーブルが存在する</li> <li>・ 補正軸用の直角度補正 が起動している (MD: \$MA_CEC_ENABLE=1)</li> <li>・ 必要な位置測定系が基準化されている (IS "Referenced/synchronized" = 1)。</li> </ul> <p>0: 直角度補正テーブル [t] の評価がイネーブルでない。</p>	
関連性	<p>MD: MM_CEC_MAX_POINTS[t] 直角度補正用の補間点数</p> <p>SD: CEC_TABLE_ENABLE[t] 直角度補正テーブル t の評価がイネーブルになっている。</p> <p>IS "Referenced/synchronized 1" DB31-48, DBX60.4</p> <p>IS "Referenced/synchronized 2" DB31-48, DBX60.5</p>	

41310 MD 番号	CEC_TABLE_WEIGHT 直角度補正テーブル [t] 用の加重係数	
デフォルト値 : 1,0	入力の下限 : ***	入力の上限 : ***
変更は直ちに有効	保護レベル : 7	単位 : 係数
データタイプ : DOUBLE	適用ソフトウェアバージョン : 2.1 以降	
意味 :	<p>テーブル [t] に格納されている補正値に重量係数を掛ける。重量係数を選択する際には、求めた補正値が最大値 (MD: CEC_MAX_SUM) を超えないことを確認すること。</p> <p>[t] = 補正テーブルのインデックス (MD: MM_CEC_MAX_POINTS を参照)</p> <p>例えば、マシンのツールまたは工作されるワークの重量は大きく異なり、振幅の変更により誤差曲線に影響を与える場合、加重係数を変更し修正することができる。直角度の補正では、PLC ユーザプログラムまたは NC プログラムを使用して設定データに上書きをし、テーブルの加重係数を、特定のツールやワークに関して変更することができる。しかし、特性曲線のプログレッションが加重の異なるのために大きく変更する場合、さまざまな補正テーブルを使用する必要がある。</p>	
関連性	<p>SD: CEC_TABLE_ENABLE[t] 直角度補正テーブル t の評価をイネーブル</p> <p>MD: \$MA_CEC_MAX_SUM 直角度補正用の最大補正値</p>	

43900 SD 番号	TEMP_COMP_ABS_VALUE 位置に依存しない温度補正	
デフォルト値 : 0	入力の下限 : ***	入力の上限 : ***
変更は直ちに有効	保護レベル : MMC-MD 9220	単位 : mm または度
データタイプ : DOUBLE	適用ソフトウェアバージョン : 1.1 以降	
意味 :	SD: TEMP_COMP_ABS_VALUE を使用して位置に依存しない温度補正値が設定される (図 2.23 を参照)。 この値は、PLC (ユーザプログラム) からの現在温度によって変化する。 位置に依存しない温度補正が起動される (MD: TEMP_COMP_TYPE = 1 または 3) と、この追加補正値は直ちにマシン軸によってトラバースされる。	
この SD は次の場合は無効	MD: TEMP_COMP_TYPE = 0 または 2	
関連性	MD: TEMP_COMP_TYPE MD: COMP_ADD_VELO_FACTOR	温度補正タイプ 補正により生じる速度違反

43910 SD 番号	TEMP_COMP_SLOPE 位置に依存する温度補正用の傾き	
デフォルト値 : 0	入力の下限 : ***	入力の上限 : ***
変更は直ちに有効	保護レベル : MMC-MD 9220	単位 : mm または度
データタイプ : DOUBLE	適用ソフトウェアバージョン : 1.1 以降	
意味 :	位置依存温度補正を使用すると、温度依存実際値ずれの誤差曲線がほぼ直線に近づく場合が多くある。この直線は、基準点 $P_0$ と傾き $\tan \beta$ で求める (図 2.23 を参照)。 SD: TEMP_COMP_SLOPE を使用して、傾き $\tan \beta$ を求める。この傾きは、現在の温度のファンクションとしての PLC ユーザプログラムを使用して変更することができる。 位置依存温度補正がアクティブになる (MD: TEMP_COMP_TYPE = 2 または 3) と、軸は直ちに現在の実際値に対して計算された補正値をトラバースする。 In MD: COMP_ADD_VELO_FACTOR では誤差曲線の最大傾き $\tan \beta_{\max}$ を制限する。この最大傾きを超えることはできない。	
この SD は次の場合は無効	MD: TEMP_COMP_TYPE = 0 または 1	
例外, エラー	TEMP_COMP_SLOPE が $\tan \beta_{\max}$ より大きい場合、傾き $\tan \beta_{\max}$ を使用して位置依存温度補正値を内部に求める。アラームの出力はない。	
関連性	MD: TEMP_COMP_TYPE SD: TEMP_COMP_REF_POSITION MD: COMP_ADD_VELO_FACTOR	温度補正タイプ 位置依存温度補正用の基準位置 補正により生じる速度違反

43920 SD 番号	TEMP_COMP_REF_POSITION 位置に依存する温度補正の基準位置	
デフォルト値 : 0	入力の下限 : ***	入力の上限 : ***
変更は直ちに有効	保護レベル : MMC-MD 9220	単位 : mm あるいは度
データタイプ : DOUBLE	適用ソフトウェアバージョン : 1.1 以降	
意味 :	<p>位置依存温度補正を使用すると、温度依存実際値ずれの誤差曲線がほぼ直線に近づく場合が多くある。この直線は、基準点 <math>P_0</math> と傾き <math>\tan \beta</math> で求める (図 2.23 を参照)。</p> <p>SD: TEMP_COMP_SLOPE を使用して、傾き <math>\tan \beta</math> を求める。この傾きは、現在の温度のファンクションとしての PLC ユーザプログラムを使用して変更することができる。</p> <p>位置依存温度補正がアクティブになる (MD: TEMP_COMP_TYPE = 2 or 3) と、軸は直ちに現在の実際値に対して計算された補正値をトラバースする。</p>	
この SD は次の場合は無効	MD: TEMP_COMP_TYPE = 0 または 1	
関連性	MD: TEMP_COMP_TYPE	温度補正タイプ
	SD: TEMP_COMP_SLOPE	位置依存温度補正用の傾き

## 2.5.5 信号の説明

補正用の個別信号はありません。

## 2.5.6 例

- なし -

## 2.5.7 データフィールド，リスト

### ■ インタフェース信号

DB 番号	ビット，バイト	名称	参照
軸／スピンドル別			
31-48	60.4	基準化／同期化 1	1.16 (R1)
31-48	60.5	基準化／同期化 2	1.16 (R1)

## ■ マシンデータ

番号	識別子	名称	参照
一般事項 (\$MN_...)			
10050	SYSCLOCK_CYCLE_TIME	基本システムサイクル	1.8 (G2)
10070	IPO_SYSCLOCK_TIME_RATIO	補間器サイクル用の係数	1.8 (G2)
10082	CTRL_OUT_LEAD_TIME	セットポイント転送時間のシフト	
10083	CTRL_OUT_LEAD_TIME_MAX	セットポイント転送時間のシフト用の最大許容可能設定	
18342	\$MN_MM_CEC_MAX_POINTS[t]	直角度補正用の最大補間点数	
チャンネル別 (\$MA_...)			
20150	GCODE_RESET_VALUES	G グループの初期設定	1.10 (K1)
軸別 (\$MC_...)			
32000	MAX_AX_VELO	最大軸速度	1.8 (G2)
32200	POSCTRL_GAIN	サーボゲイン係数	1.8 (G2)
32450	BACKLASH[n]	バックラッシュ	
32460	TORQUE_OFFSET	電子重量補正用の追加トルク	
32490	FRICT_COMP_MODE	摩擦補正のタイプ	
32500	FRICT_COMP_ENABLE	摩擦補正アクティブ	
32510	FRICT_COMP_ADAPT_ENABLE [n]	摩擦補正適用アクティブ	
32520	FRICT_COMP_CONST_MAX [n]	最大摩擦補正值	
32530	FRICT_COMP_CONST_MIN [n]	最小摩擦補正值	
32540	FRICT_COMP_TIME [n]	摩擦補正時定数	
32550	FRICT_COMP_ACCEL1 [n]	適応加速値 1	
32560	FRICT_COMP_ACCEL2 [n]	適応加速値 2	
32570	FRICT_COMP_ACCEL3 [n]	適応加速値 3	
32580	FRICT_COMP_INC_FACTOR	短いトラバース動作の摩擦補正の加重係数	
32610	VELO_FFW_WEIGHT	速度フィードフォワード制御用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のタイプ	
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	
32640	STIFFNESS_CONTROL_ENABLE	動的剛性制御	
32650	AX_INERTIA	トルクフィードフォワード制御用の慣性モーメント	
32652	AX_MASS	トルクフィードフォワード制御用の軸の質量	
32700	ENC_COMP_ENABLE	補間補正アクティブ	
32710	CEC_ENABLE	直角度の補正イネーブル	
32711	CEC_SCALING_SYSTEM_METRIC	直角度補正の測定系	
32720	CEC_MAX_SUM	直角度補正の最大補正值	
32730	CEC_MAX_VELO	直角度補正用の最大変更値	
32750	TEMP_COMP_TYPE	温度補正	
32760	COMP_ADD_VELO_FACTOR	補正を使用したことによる速度違反	
32800	EQUIV_CURRCTRL_TIME[n]	現在の制御ループの等価時定数	
32810	EQUIV_SPEEDCTRL_TIME[n]	速度制御ループの等価時定数	
36200	AX_VELO_LIMIT	速度監視用の制限値	1.2 (A3)

番号	識別子	名称	参照
36400	CONTOUR_TOL	公差範囲輪郭監視	1.2 (A3)
36500	ENC_CHANGE_TOL	位置実際値切換え用の最大公差	1.8 (G2)
38000	MM_ENC_COMP_MAX_POINTS[n]	補間補正用の補間点数	
38010	MM_QEC_MAX_POINTS	ニューラルネットワークを使用する QEC 用のオフセット値の最大数	
SIMODRIVE 611D マシンデータ (\$MD_...)			
1004	CTRL_CONFIG	構成	IAD
1117	MOTOR_INERTIA	モータの慣性モーメント	IAD

## ■ 設定データ

番号	識別子	名称	参照
一般事項 (\$MN_ ... )			
41300	CEC_TABLE_ENABLE	直角度補正テーブルの評価イネーブル	
41310	CEC_TABLE_WEIGHT	直角度補正テーブル用の加重係数	
軸別 (\$SA_...)			
43900	TEMP_COMP_ABS_VALUE	位置に依存しない温度補正值	
43910	TEMP_COMP_SLOPE	位置に依存する温度補正用のリード角	
43920	TEMP_COMP_REF_POSITION	位置に依存する温度補正用の基準位置	

---

## ■ アラーム

アラームの詳細説明については次を参照してください。

参照： アラーム一覧

あるいは MMC 101/102/103 を有するシステムのオンラインヘルプ

---



---

## 2.6 チャンネル間の協調運転と制御軸入替え (K5)

### 2.6.1 概略説明

#### モードグループ

モードグループは、マシンの軸、スピンドル、およびチャンネルのユニットまたはグループで構成されています。モードグループは、原理としては、NC 制御（複数のチャンネルを使用する）に類似しています。モードグループとは、常に同一モードで動作する必要がある全てのチャンネルが集まったものです。

(注)

標準の場合、モードグループは存在します。詳細については次を参照してください。

参照： 1.10 運転モードの種類とプログラム運転モード (K1)

#### チャンネル

各チャンネルは、プログラムデコーディング、ブロック準備、および補間用にそれぞれの役割を果たしています。チャンネルは単独で、パートプログラムを処理することができます。

(注)

標準の場合、チャンネルは存在します。詳細については次を参照してください。

参照： 1.10 運転モードの種類とプログラム運転モード (K1)

モードグループの複数のチャンネルの処理と、パートプログラムの処理は同期化することができます。

#### 軸／スピンドル置換

制御システムの起動後、軸／スピンドルは特定のチャンネルにを割当てられ、その割当てられたチャンネル内でのみ使用できます。

"Axis/spindle replacement" (軸／スピンドル置換) ファンクションを使用すると、任意の軸／スピンドルをイネーブルにし、それを別のチャンネルに割当てることができます。すなわち、軸／スピンドルを置換することができます。

ソフトウェアバージョン 3 以降では、パートプログラムと PLC プログラムのいずれを使用しても、軸／スピンドル置換を起動できます。

## 2.6.2 詳細説明

### ■ モードグループ

#### モードグループ

モードグループは、集合して1台のマシンユニットを構成している、NC チャンネル、軸、およびスピンドルが連携したものです。

モードグループには、必ず同一モードで操作しなければならないチャンネルが全て含まれます。任意のモードグループの任意のチャンネルに任意の軸をプログラムすることができます。したがって、モードグループは、独立した、マルチチャンネル NC に対応しています。

#### 例

規模の大きな工作機械（マシニングセンタ）では、場合によっては、マシンのあるパートでパートプログラムを処理する一方で、別のパートで新しいワークピースをクランプしセットアップする必要があります。このようなタスクを行うには、普通はそれぞれ独立した2つのNC制御が必要となります。

モードグループファンクションを使用すると、異なるモードをそれぞれのモードグループにセットすることができるため、この2つのタスクを2つのモードグループを持つ1つのNC制御に設定することができます（プログラム処理用にはAUTOMATICモード、ワークピースのセットアップ用にはJOG）。

#### モードグループの割当て

モードグループの構成が、構成に含めるチャンネル、ジオメトリ軸、マシン軸、およびスピンドルを決定します。

モードグループは1つまたは複数のチャンネルから構成されます。これらのチャンネルを他のモードグループに割当ててはなりません。マシン軸、ジオメトリ軸、および特別軸がこれらのチャンネルに割当てられます。マシン軸は、同じモードグループのチャンネルにしか割り当てることができず、そのモードグループ内でしかトラバースすることはできません。

モードグループには、次のデータが設定されています：

- チャンネル別 MD 10010: ASSIGN\_CHAN\_TO\_MODE\_GROUP  
(モードグループで有効なチャンネル)
- チャンネルの構成データ

(注)

第1モードグループの詳細については、以下を参照してください。

参照： 1.10 運転モードの種類とプログラム運転モード (K1)

## ■ チャンネル

(注)

第1チャンネル用の Channel (チャンネル), Channel Configuration (チャンネル構成), Channel States (チャンネルステータス), Effects of Commands/Signals (コマンド/信号の効果) などの用語については、以下の参照に説明があります。

参照： 1.10 運転モードの種類とプログラム運転モード (K1)

他のチャンネルについても、この情報が同様に適用されます。

## ■ チャンネル同期化 (プログラム協調)

### 一般事項

#### 定義

例を挙げて説明します。2つのチャンネルの処理を同期化できて初めて、ダブルスライド加工運転や実時間処理を実行できます。

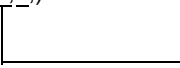
関連するチャンネルは、同期した特定の処理プロセスを実行することになります。同期処理を行うためには、関連のチャンネルを連結し同期グループを構成する必要があります (モードグループ)。

チャンネル同期化は、NC 言語を使用して設定します。関連するチャンネルは同一のモードグループに割当てする必要があります。

#### 協調

複数のチャンネルがワークピースの加工にかかわっている場合、個々のチャンネルのプログラムランを同期化する必要があります。このプログラム協調には特別なインストラクション (コマンド) があります。それぞれの場合に応じて、ブロックごとに表にまとめました。

表 2.5 プログラム協調コマンド

コマンド	意味
ソフトウェアバージョン 3	
INIT(n,"identifier","q") 	特定のチャンネルで処理するためのプログラム選択 確認モード：n (同期なし) または s (同期) パス指定のあるプログラム名 チャンネル名：1 ～ 4 の値が使用可
CLEAR (identifier)	プログラム識別子を指示するプログラムの削除
START (n,n,n) 	他のチャンネル中で選択されたプログラムのスタート チャンネル番号の指定：1 ～ 4 の値が使用可
WAITM (Mnr, n, n, n, n)	指定したチャンネル n (オプションで、使用チャンネルを指示することができる) でプログラム同期化用のマーク番号 Mnr を待機する。 チャンネルのマーク番号は全て同一であること。 1 ～ 9 の値が使用可
WAITE (n,n,n)	指示されたチャンネルのプログラムエンドを待機 (プログラム協調チャンネルを指定しないこと)
ソフトウェアバージョン 4	
SETM(Mnr1, Mnr2, ...Mnri)	SETM() が発行されているチャンネルに対して、WAITMC() を使用して条件待機用の待機マーク Mnr1, Mnr2, ...Mnri をセットする。したがって、そのチャンネルはパートナーチャンネルに対して、待機特性がまだ (もはや) 満足されていないことを知らせる。このコマンドは同期化アクションで起動できる。1 つのコマンドを使って 10 マーク (0 ～ 9) まで選択することができる。
CLEARM(Mnr1, Mnr2, ...Mnri)	SETM() が発行されているチャンネルに対して、WAITMC() を使用して条件待機用の待機マーク Mnr1, Mnr2, ...Mnri を削除する。したがって、そのチャンネルはパートナーチャンネルに対して、待機特性が満足されていることを知らせる。このコマンドは同期化アクションで起動できる。1 つのコマンドを使って 10 マーク (0 ～ 9) まで選択することができる。
WAITMC(Mnr, n1, n2, ...)	指定されたチャンネル n1, n2, ... nk からの、指定された待機特性 Mnr についての連続パスモードでの条件待機。オプションのプログラム協調チャンネルを指示することもできる。グループ内の他のチャンネルからの待機マークが到着してもまだ処理が継続する場合、これらのチャンネルの待機マークは削除される。

---

## ソフトウェアバージョン 3

### 挙動

WAITM() コールが到着すると、現在のチャンネルの軸は宣言を受け、そしてコールに指定されたマーク番号が同期化される他のチャンネルから到着するまで待機します。他のチャンネルもまたそれぞれの WAITM() コマンドに達して減速したとき、このグループは同期化されます。そして、同期化されたチャンネルが操作を継続します。

### プログラム協調の例

チャンネル 1:

%100

N10 INIT(2,"\_N\_200\_MPF","n")

N11 START(2)

. ; チャンネル 1 での処理

N80 WAITM(1,1,2) ; チャンネル 1 およびチャンネル 2 で WAIT マークを待つ

. ; チャンネル 1 で処理を続行する

N180 WAITM(2,1,2) ; チャンネル 1 およびチャンネル 2 で WAIT マークを待つ

. ; チャンネル 1 で処理を続行する

N200 WAITE(2) ; チャンネル 2 のプログラムエンドを待つ

N201 M30 ; プログラムエンド (チャンネル 1), すべての最後

.

.

チャンネル 2:

%200

. ; チャンネル 2 で処理をする

N70 WAITM(1,1,2) ; チャンネル 1 およびチャンネル 2 で WAIT マークを待つ

. ; チャンネル 2 で処理を続行する

N270 WAITM(2,1,2) ; チャンネル 1 およびチャンネル 2 で WAIT マーク 2 を待つ

. ; チャンネル 2 で処理を続行する

N400 M30 ; チャンネル 2 のプログラムエンド

.

.

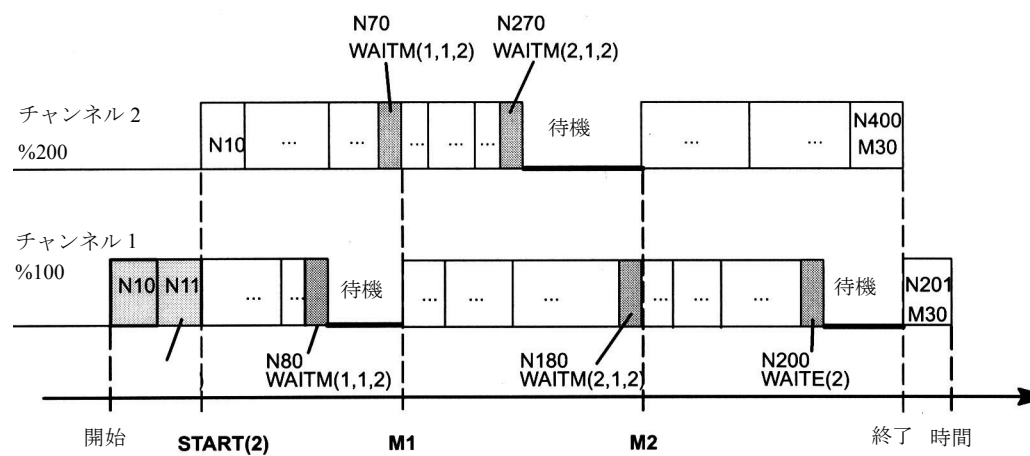


図 2.58 WAITM()（無条件待機）との強調例を示すプログラムラン

参照：YS840DI ユーザーズマニュアル・プログラミング編

## ソフトウェアバージョン 4

### 目的

協調させたい全てのチャンネルの中に同期化を目的としたマーク番号がセットされているのではない場合に限って、減速および待機が実行されなければなりません（条件待機）。

待機マーク作成の時間と条件待機コールは切離されています。

内部チャンネル通信のために、待機と減速を全く意図していない場合でもマークをセットすることができます。WAITMC() コマンドはありません。この場合、RESET と NC Start が実行されてもチャンネルマーク設定は有効です。

### 条件待機の前提条件

WAITMC() を使用した条件待機を利用し待機時間を低減するには、次の条件が満たされていなければなりません：

- 連続パスモード G64 がセットされていること
- ルックアヘッドファンクションがアクティブであること
- 正確停止 (G60, G09) がセットされていないこと

正確停止が選択されている場合、WAITMC() を使用した待機は SW レベル 3 から WAITM() を使用する待機に対応しています。

### 応答

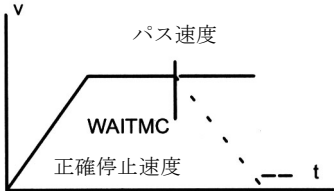
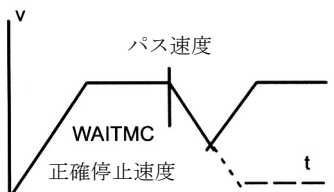
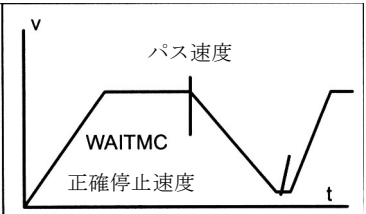
A) WAITMC() コールに先立ち動作ブロックを使用してスタートすると、同期化される他のチャンネルの待機マークがチェックされます。チェックが全て終了すると、チャンネルは連続パスモードで減速せずに操作を継続します。

B) 同期化されるチャンネルのうちの 1 つから出た待機マークが 1 つでも喪失した場合、軸はパス速度から正確停止速度へと減速し始めます。ここで同期化されるチャンネルから喪失した待機マークが到着していないか、全ての補間サイクルで検査が行われます。到着していた場合、軸はサイドパス速度を加速して正常操作を継続します。

C) 同期化されたチャンネルに付けられたマークが正確速度に達するまでに到達しない場合、工作操作は行方がわからないマークが発見されるまで停止されます。必要であるマークの最後の 1 つが現れると、軸は静止状態からパス速度へと加速されます。

次の表に、A) ～ C) のイベントの過程をまとめています。

表 2.6 WAITMC() を使用した条件待機に応じた減速反応

WAITMC を使用する	応答	速度曲線
A) 全チャンネルの待機マークにすでに到着している	減速せず運転を続行	 <p>速度曲線: 速度 (v) 対 時間 (t) のグラフ。パス速度で走行中、WAITMC が発生しても速度は減速せず、そのまま継続する。</p>
B) パス速度から正確停止速度への減速中に待機マークに全て到着する	最後の待機マークに到着し次第、減速は止まる。軸はパス速度に戻るため加速される。	 <p>速度曲線: 速度 (v) 対 時間 (t) のグラフ。減速中に WAITMC が発生し、減速が止まる。その後、軸は加速され、再びパス速度に戻る。</p>
C) 最後の待機マークに到着せずに正確停止速度に達した。	ブレーキがかかり正確停止速度になる。必要である最後の1つのマークが現れると、軸は正確停止速度からパス速度へと加速される。	 <p>速度曲線: 速度 (v) 対 時間 (t) のグラフ。減速中に WAITMC が発生し、速度が正確停止速度まで低下する。その後、マークが現れると、軸は加速され、再びパス速度に戻る。</p>



---

## 連続パスモードでの条件待機の例

例は、概要図の形で同期化プロセスに関連するコマンドだけを示しています。

チャンネル 1:

%100

N10 INIT(2, "\_N\_200\_MPF", "n") ; パートナプログラムチャンネル 2 を選択する

N11 INIT(3, "\_N\_300\_MPF", "n") ; 3 パートナプログラムチャンネル 3 を選択する

N15 START(2, 3) ; チャンネル 2, 3 中のプログラムをスタートする

... ; チャンネル 1 で処理する

N20 WAITMC(7, 2, 3) ; チャンネル 2 および 3 からマーク 7 を条件付き  
で待つ

... ; チャンネル 1 で処理する

N40 WAITMC(8, 2) ; チャンネル 2 からマーク 8 を条件付で待つ

... ; チャンネル 1 で処理が継続する

N70 M30 ; チャンネル 1 の終了

Channel チャンネル 2:

%200

N200 ; チャンネル 2 で処理する

N210 SETM(7) ; チャンネル 2 が待機マーク 7 をセットする

... ; チャンネル 2 で処理が継続する

N250 SETM(8) ; チャンネル 2 が待機マーク 8 をセットする

N260 M30 ; チャンネル 2 の終了

チャンネル 3:  
 %300  
 N300 ; チャンネル 3 で処理する  
 ...  
 N350 WHEN <condition> DO SETM(7)  
 ; 同期化アクションの中に待機マークをセットする  
 ... ; チャンネル 3 で処理が続行する  
 N360 M30 ; チャンネル 3 の終了

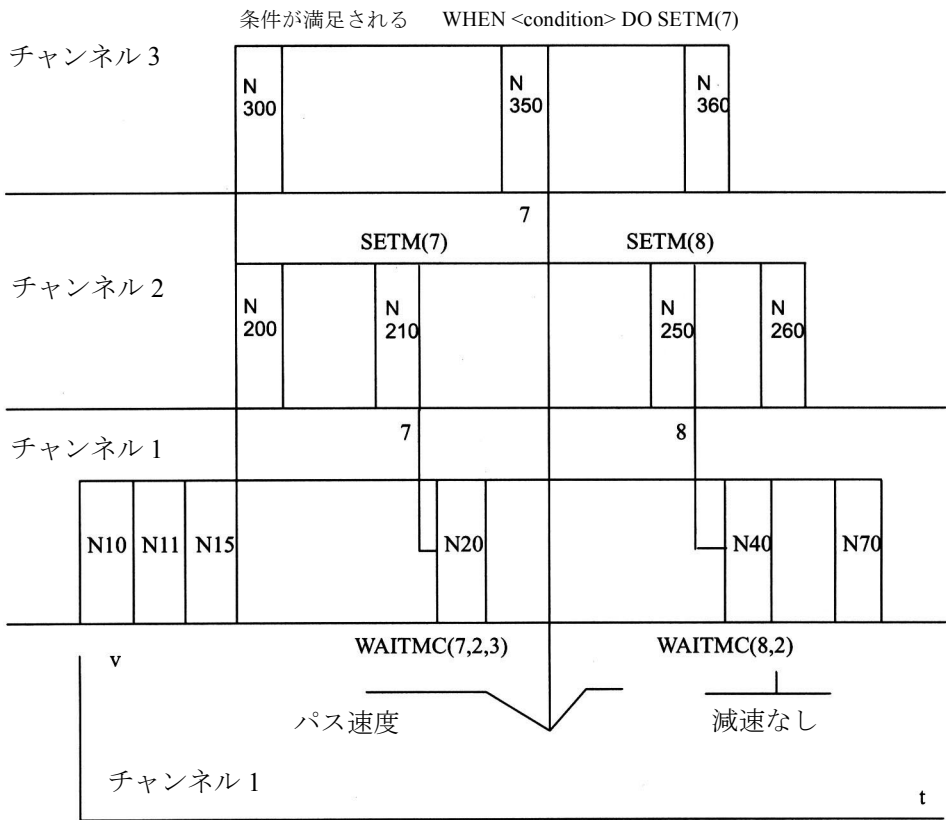


図 2.59 チャンネルが 3 つの場合の条件待機（図示）

## ■ 軸／スピンドル置換

### ■ はじめに

#### 一般事項

軸／スピンドルはマシンデータを使ってチャンネルに確実に割当てます。

軸／スピンドルは割当てたチャンネルでしか使用できません。

#### 定義

“軸／スピンドル置換” ファンクションを使うと、軸やスピンドルをイネーブルにし、別のチャンネルに割り当てる（置換する）ことができます。

スピンドルファンクションは軸ファンクションの中に含まれるため、以後は“軸置換”と呼びます。

#### 軸のタイプ

チャンネルに基づき、軸を 4 つに分類します。軸変更時の定義は MD 30552: `AUTO_GET_TYPE` の設定によって異なります。

##### チャンネル軸

チャンネル軸は、パートプログラムで設定し、全モードで移動できます。

##### PLC 軸

PLC 軸は PLC を介して位置決めします。

PLC 軸がパートプログラムに設定されている場合、

`MD AUTO_GET_TYPE = 0` のとき、アラームが出力されます。

`MD AUTO_GET_TYPE = 1` のとき、自動 GET が生成されます。

`MD AUTO_GET_TYPE = 2` のとき、自動 GETD が生成されます。

##### ニュートラル軸

ニュートラル軸がパートプログラムに設定されている場合、

`MD AUTO_GET_TYPE = 0` のとき、アラームが出力されます。

`MD AUTO_GET_TYPE = 1` のとき、自動 GET が生成されます。

`MD AUTO_GET_TYPE = 2` のとき、自動 GETD が生成されます。

##### 別のチャンネル中の軸

これは実際には軸のタイプではなく、置換可能な軸の内部状態です。別のチャンネルで軸がアクティブになった場合（チャンネル、PLC、またはニュートラル軸として）。

軸がパートプログラムの別のチャンネルに設定されている場合、

`MD AUTO_GET_TYPE = 0` のとき、アラームが出力されます。

`MD AUTO_GET_TYPE = 1` のとき、自動 GET が生成されます。

`MD AUTO_GET_TYPE = 2` のとき、自動 GETD が生成されます。

(注)

MD 20110: RESET\_MODE\_MASK および MD 20112: START\_MODE\_MASK が、リセット、起動、パートプログラムスタートにおける軸割当ての動作を制御します。軸変更が見込まれるチャンネルの設定は、MD 30552 (アラーム) と適合するように選択する必要があります。

参照： 1.11 軸構成及び座標系 (K2)

## 画面

現在の軸タイプおよびこの軸の現在のチャンネルが、軸 PLC インタフェース表示部に表示されます。「■ PLC を介しての軸置換」を使って軸変更を参照してください。

## 前提条件

軸置換を行うために、軸別 MD 20070: AXCONF\_MACHAX\_USED を使って、チャンネルで有効なマシン軸番号を定義する必要があります。

また、

軸指定 MD 30550: AXCONF\_ASSIGN\_MASTER\_CHAN を使って、軸置換用チャンネルの初期設定を定義してください。

- 1) どのチャンネルを使用して、軸を置換するか。
- 2) どのチャンネルに POWER ON 時の軸を割当てるか。

## 例

6 軸 2 チャンネルで、チャンネル 1 に第 1, 第 2, 第 3, 第 4 軸, チャンネル 2 に第 5, 第 6 軸を使用する。第 1 軸を置換可能とし、POWER ON 後にチャンネル 2 に割当てて場合。

以下を使用して、チャンネル別 MD を割当ててください：

CHANDATA(1)

AXCONF\_MACHAX\_USED=(1, 2, 3, 4, 0, 0, 0, 0)

CHANDATA(2)

AXCONF\_MACHAX\_USED=(5, 6, 1, 0, 0, 0, 0, 0)

以下を使用して、軸別 MD を割当ててください：

AXCONF\_ASSIGN\_MASTER\_CHAN[AX1]=2

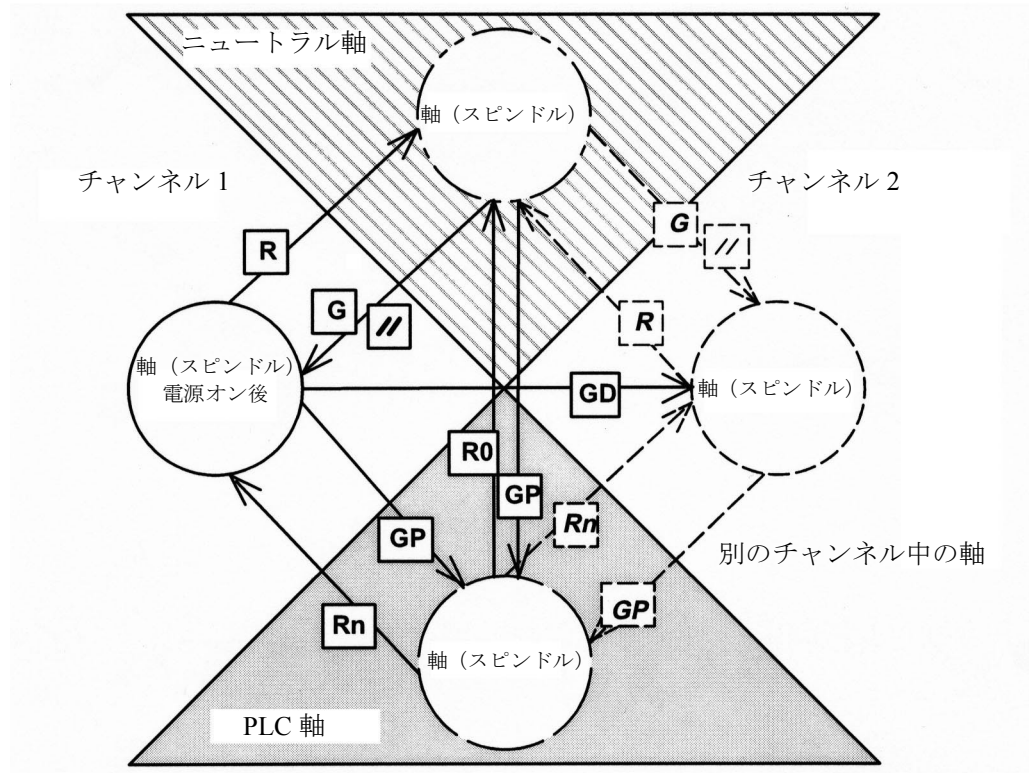
(注)

選択したチャンネルで軸が有効でない場合、MMC の軸名が反転して表示されます。

## ■ 説明

### 遷移の可能性

次の図は、軸置換の可能性を示したものです。



<b>R</b>	RELEASE(AX ...) NC プログラムから	<b>G</b>	GET (AX ...) NC プログラムから
<b>//</b>	リセットキー		
<b>R0</b>	PLC からニュートラルステータスへリリース	<b>GP</b>	GET (PLC を使用して)
<b>Rn</b>	PLC から指定チャンネルへリリース	<b>GD</b>	GET (直接 NC プログラムから)

## ■ ニュートラルステータスへの軸転送（リリース）

### RELEASE

パートプログラム中での表記：

RELEASE ( 軸名, 軸名, SPI (スピンドル番号) , ....)

(注)

軸名は系の軸の割当てに対応していて、次のいずれかとなります

- AX1, AX2, AX3, ... または
- MD 10000: AXCONF\_MACHAX\_NAME\_TAB に割当てられた名前

RELEASE ( 軸名 , ...) を使用すると、常に専用 NC ブロックが生成されます。

例外：軸がすでにニュートラルステータスにある場合。

次のような場合、RELEASE コマンドは中断されます

- 軸置換の前提条件が満たされていない場合  
(MD 20070: AXCONF\_MACHAX\_USED および  
MD 30550: AXCONF\_ASSIGN\_MASTER\_CHAN)
- 軸が変換に関与している場合
- 軸が軸のネットワーク内にある場合

(注)

RELEASE コマンドがガントリーマスタ軸に適用されている場合、次の軸も全てリリースされます。

条件	経過	結果
軸がリリースされるが、まだ GET を使用して転送されていない ...	... 操作パネルを介してリセットが発生する ...	... 軸は再び最後のチャンネルに割当てられる

■ 軸テークオーバー

1a) パートプログラム中のコマンド (GET コマンド)

GET ( 軸名 , 軸名 , SPI (スピンドル番号) , ...)

次の場合、軸のテークオーバーが待たされる

- 軸が測定系を変更中である
- 軸のサーボディスエーブル処理中である (追従／停止での制御からの遷移、ならびにその逆)
- 軸／スピンドルディスエーブルがセットされている
- 軸がまだ RELEASE を使用した他のチャンネルによってイネーブルにされていない
- 軸の補間がまだ完了していない (ただし、速度制御スピンドルは除く)

GET ( 軸名 , ...) を使用すると、サーチ停止を使用する個別 NC ブロックが常に生成されます。

例外：

- 軸がすでにチャンネルにある場合、ブロックは生成されません
- 軸が同期である場合、(すなわち、軸はその間に別のチャンネルにスワップされていない、あるいは PLC からの信号を受け取っていないので)、予備のブロックも生成されません。

1b)GETD コマンド

GETD (GET Directly) を使用して、軸を別のチャンネルから直接持ってくるができます。すなわち、別のチャンネル中で、この GETD 用に RELEASE をプログラムしてはいけません。さらに、供給側のチャンネルが GETD1 を使用すると中断されるため、別のチャンネル通信を確立しなければなりません (例えば、待機マーク)。軸が PLC 軸である場合、置換は PLC が軸をイネーブルするまで待たされます。

### 注意

この指令により，指定された軸があるプログラムの実行が中断されます (REORG)。

例外：軸はこの時点ではニュートラルステータスです。

(注)

条件	経過	結果
GET コマンドが設定されていて，転送が待たされる ...	... ..チャンネルでリセットが生じる	... チャンネルは軸のテークオーバを中断する

GET を使用して割当てた軸は，キー リセット またはプログラム リセット の後でも，このチャンネルに割当てられたままです。RELEASE または GET をプログラミングしてこの軸を再度置換することができます。そうしなければ，電源オン後にこの軸は MD 30550: AXCONF\_ASSIGN\_MASTER\_CHAN で定義されたチャンネルに割当てられたままです。

## 2) 軸名のプログラミングを通して自動的にコマンドを生成

### 例 1

MD 30552: AUTO\_GET\_TYPE に設定した値により，ニュートラル軸が再びプログラミングされたときに GET コマンドまたは GETD コマンドが自動的に生成されます。

N1 M3 S1000

N2 RELEASE (SPI(1)) ;=> ニュートラルステータスに遷移

N3 S3000 ; リリースされた軸の新規速度

; MD AUTO\_GET\_TYPE =

; 0 => アラーム "Wrong axis type" が出力されます

; 1 => GET (SPI(1)) が生成されます

; 2 => GETD (SPI(1)) が生成されます

### 例 2

; ( 軸 1 = X)

N1 RELEASE (AX1) ;=> ニュートラルステータスへの遷移

N2 G04 F2

N3 G0 X100 Y100: ; リリースされた軸の動作

; MD AUTO\_GET\_TYPE =

; 0 => アラーム "Wrong axis type" が出力されます

; 1 => GET (AX1) が生成されます

; 2 => GETD (AX1) が生成されます

### 例 3

```
                                ; (軸 1 = X)
N1 RELEASE (AX1)                ;=> ニュートラルステータスへの遷移
N2 G04 F2
N3 POS (X) = 100:                ; リリースされた軸の位置決め :
                                ; MD AUTO_GET_TYPE =
                                ; 0 => アラーム "Wrong axis type" が出力されます
                                ; 1 => GET (AX1) が生成されます *)
                                ; 2 => GETD (AX1) が生成されます *)
```

\*) 軸がまだ同期化している場合、専用ブロックは作成されません。

(注)

自動 GETD がセットされている場合、次のことに注意しなければなりません：

1. チャンネルが相互に影響し合うことがある（軸が取り去られた場合は REORG）。
2. 1 つの軸へ複数のチャンネルが同時にアクセスした場合、最終的にどのチャンネルがその軸にアクセスできるか分からない。

## ■ 軸置換の例

### 仮定

6 つの軸と 2 つのチャンネルを使用し、第 1、第 2、第 3、および第 4 軸はチャンネル 1 で、第 5 と第 6 軸はチャンネル 2 で使用する場合。チャンネル間同士で第 2 軸を置換できるようにし、電源オン後にチャンネル 1 に割当てることとします。

### タスク

タスクとしては次の 2 つがあります：

- 軸置換の必須条件が与えられるようなマシンデータ割当て
- チャンネル 1 と 2 との間での軸置換のプログラミング



## 前提条件の設定

チャンネル別 MD 20070 の割当て：

AXCONF\_MACHAX\_USED[1]=(1, 2, 3, 4, 0, 0, 0, 0)

AXCONF\_MACHAX\_USED[2]=(5, 6, 2, 0, 0, 0, 0, 0)

軸別 MD 30550:AXCONF\_ASSIGN\_MASTER\_CHAN[AX2] の割当て

チャンネル 1 でのプログラム

...

RELEASE (AX2)

；軸 AX2 のリリース

INIT (2, "\_N\_MPF\_DIR\\_N\_TAUSH2\_MPF", "S")

；チャンネル 2 でのプログラム TAUSH2 の選択

START (2)

；チャンネル 2 でのプログラムのスタート

WAITM (1,1,2)

；チャンネル 1 および 2 で待機マーク 1 を待つ

... ；

... ；軸置換後の運転シーケンス

... ；

... ；

M30

チャンネル 2 でのプログラム TAUSH2

...

WAITM (1,1,2)

；チャンネル 1 および 2 で待機マーク 1 を待つ

GET (AX2)

；軸 AX2 を指定

... ；

... ；軸置換後の運転シーケンス

... ；

RELEASE (AX2)

；更なる軸置換用のリリース

... ；

M30

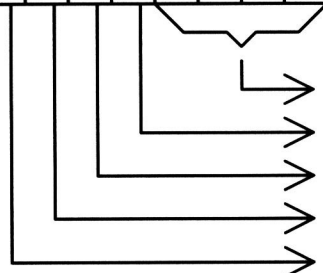
## ■ PLC を介しての軸置換

- インタフェースタイプを使用していつでも軸のタイプを指定することができます  
(PLC 軸, チャンネル軸, ニュートラル軸)

TYPE 画面

**NCK => PLC (DB31- ..., DBB68)**

0 0 0 0 0 0 0 0



チャンネル 1 ~ 10 の NC 軸  
PLC から要請された新規タイプ  
要請された新規タイプ  
ニュートラル軸  
PLC 軸

例 1: チャンネル軸 K2

0 0 0 0 0 0 1 0

例 2: PLC 軸。K1 の補間処理およびインタフェース処理

1 0 0 0 0 0 0 1

例 3: ニュートラル軸。K3 のインタフェース処理。

0 1 0 0 0 0 1 1

	NCK=>PLC, DBB68	PLC=>NCK, DBB8
電源オン後	0 0 0 0 0 0 0 1	0 0 0 0 0 0 0 0
RELEASE (K1)	0 1 1 0 0 0 0 1	
	0 1 0 0 0 0 0 1	
GET (K2)	0 0 1 0 0 0 1 0	
	0 0 0 0 0 0 1 0	

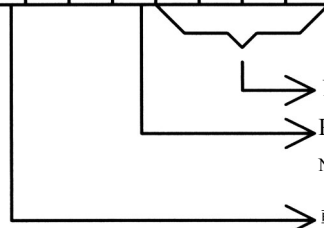
↓  
時間

図 2.60 K1 から K2 への軸置換

- PLC は、いつでもどのような運転モードでも、軸を要請し移動することができます。

TYPE の指定 **PLC => NCK (DB31- ..., DBB8)**

0 0 0 0 0 0 0 0



NC 軸はチャンネル 1 ~ 10 で移動します  
PLC: 新規タイプの要請  
NC は立上がりエッジをとらえます。

軸は PLC 軸となります。

原則として、PLC は信号 "Request new type" をセットしなければなりません。また、変更後は削除されます。これは、GET および RELEASE を使用したチャンネル

ル変更にも適用できます。

- EPLC は、1 つの軸をチャンネルから別のチャンネルへ変更することができます。

PLC 軸および PLC スピンドルは、基本 PLC プログラムの特別ファンクションモジュールを介して移動されます。

FC15: POS\_AX 直線軸および回転軸の位置決め

FC16: PART\_AX インデックス軸の位置決め

FC18: SpinCtrl スピンドル制御

## 例

次の図には、NC 軸の PLC 軸への変更ならびに PLC を使用した NC 軸からニュートラル軸への転送の IS 信号シーケンスを示しています。

	NCK=>PLC, DBB68	PLC=>NCK, DBB8
電源オン後	0 0 0 0 0 0 0 1	0 0 0 0 0 0 0 0
新規 TYPE (PLC)		1 0 0 1 0 0 0 0
	0 0 1 1 0 0 0 1	1 0 0 1 0 0 0 0
	0 0 1 1 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

⇓  
時間

図 2.61 NC 軸から PLC 軸への変更

	NCK=>PLC, DBB68	PLC=>NCK, DBB8
電源オン後	0 0 0 0 0 0 0 1	0 0 0 0 0 0 0 0
新規 TYPE (PLC)		1 0 0 1 0 0 0 0
	0 0 1 1 0 0 0 1	1 0 0 1 0 0 0 0
		1 0 0 0 0 0 0 0
	1 0 0 0 0 0 0 1	1 0 0 0 0 0 0 0
新規 TYPE (PLC)		0 0 0 1 0 0 0 0
	1 0 1 1 0 0 0 1	0 0 0 1 0 0 0 0
	0 1 0 0 0 0 0 1	0 0 0 0 0 0 0 0

⇓  
時間

図 2.62 PLC を使用した NC 軸からニュートラル軸への変更

---

## 2.6.3 補足条件

### 「モードグループ」 ファンクション

本 NC にはモードグループがあります。

### チャンネル数

本 NC には 2 チャンネル（1 チャンネル）あります。

### チャンネル軸への変更

軸を、PLC 軸、ニュートラル軸、または別のチャンネルの軸からその軸タイプのチャンネル軸へ変更する場合、同期化を実行する必要があります。

この同期化を行うと：

- 現在の位置が確定される。
- スピンドルについて現在の速度とギアの状態が確定される。

したがって、アクティブパス動作を中断するフィード停止を実行する必要があります。

GET を使用して軸が転送される場合、この遷移はパートプログラムによって定義します。

軸が PLC によって割当てられる場合、変更が行われるプログラムの場所は明確には予測できません。

（ただし、個別のユーザー別 NC <-> PLC ロジックによる場合は除く）

このため、チャンネル軸への変更は次の場合延期されます。

- パスモードがアクティブ (G64+ プログラムされている軸)
- ねじ切り／タッピングがアクティブ (G33/G331/G332)

### チャンネル軸からの変更

アクティブパスオペレーション中は、チャンネル軸をニュートラル軸または PLC 軸へ変更することはできません。

その理由として、RELEASE を使用すると、RELEASE を個別の NC ブロックに割当てなければならないからです。

PLC が軸を変更する場合、REORG が内部的に起動されます。したがって、一覧に挙げたプログラムを使用する変更が待たされます。

---

## ブロックサーチ

計算を使用するブロックサーチ時には、GET、GETD、または RELEASE ブロックは全て、次の NC スタート後に保存され、出力されます。

例外：

相互に除外されるブロックは削除されます。

例：

N10      RELEASE (AX1)      ブロックは削除されます。

N40      GET (AX1)      "

N70

## 2.6.4 データの説明 (MD, SD)

### ■ 軸／スピンドル別マシンデータ

30550 MD 番号	AXCONF_ASSIGN_MASTER_CHAN 軸変更用のチャンネルの初期設定	
デフォルト値 : 0	入力下限 : 0	入力上限 : 2
電源オン後, 変更は有効	保護レベル : 2	単位 : -
データタイプ : BYTE	適用ソフトウェアバージョン : 2 以降	
意味 :	電源オン後に軸が割当てられるチャンネルの定義。	
アプリケーション	ファンクション “軸／スピンドル置換” を使用して, 電源オン後に, マシン軸がチャンネルに割当てられなければならない。 AXCONF_ASSIGN_MASTER_CHAN[AX2]=1 ⇒ 軸 AX2 が電源オン後にチャンネル 1 に割当てられる。	
関連性 ....	MD: AXCONF_MACHAX_USED	

30552 MD 番号	AUTO_GET_TYPE 自動 GET の定義	
デフォルト値 : 1	入力下限 : 0	入力上限 : 1
電源オン後, 変更は有効	保護レベル : 2	単位 :
データタイプ : BYTE	適用ソフトウェアバージョン : 3 以降	
意味 :	0 = GET が自動的に生成されない ⇒ 不正なプログラミングにアラームが応答する 1 = GET が出力される (GET が自動的に生成された場合) 2 = GETD が出力される (GET が自動的に生成された場合)	
関連性 ...		

## 2.6.5 信号の説明

DB31, ... DBB8 データブロック	軸／スピンドル置換 チャンネルへの信号 (PLC ---> NCK)	
エッジ評価：あり	信号更新：周期的	信号が有効となる最初の SW バージョン：1.1
信号ステータス 1 または 信号遷移 0 ---> 1	現在の軸タイプとこの軸に対して現在アクティブなチャンネルを指定する必要がある。 Bit 0: A チャンネルに NC 軸／スピンドルを割当て Bit 1: B ... Bit 2: C ... Bit 3: D チャンネルに NC 軸／スピンドルを割当て Bit 4: 起動，正の信号エッジによる割当て Bit 5: - Bit 6: - Bit 7: PLC 軸／スピンドル要請	
信号ステータス 0 または 信号遷移 1 ---> 0		
関連性 ...	IS DB31, ... DBB68, "Axis/spindle replacement" MD 20070: AXCONF_ASSIGN_MASTER_USED (チャンネルで有効なマシン軸番号) MD 30550: AXCONF_ASSIGN_MASTER_CHAN (軸置換用のチャンネルの初期設定)	
例外，エラー .....		

DB31, ... DBB68 データブロック	軸／スピンドルの置換 チャンネルへの信号 (PLC ---> NCK)	
エッジ評価：あり	信号更新：周期的	信号が有効となる最初の SW バージョン：1.1
信号ステータス 1 または 信号遷移 0 ---> 1	現在の軸タイプとこの軸に対して現在アクティブなチャンネルが表示される。 Bit 0: A チャンネルの NC 軸／スピンドル Bit 1: B ... Bit 2: C ... Bit 3: D チャンネルの NC 軸／スピンドル Bit 4: PLC に要請された新規タイプ Bit 5: 要請された新規タイプ (GETD) Bit 6: ニュートラル軸／スピンドル Bit 7: PLC 軸／スピンドル	
信号ステータス 0 または 信号遷移 1 ---> 0		
関連性 ...	IS DB31, ... DBB68, “軸／スピンドル置換” MD 20070: AXCONF_ASSIGN_MASTER_USED (チャンネルで有効なマシン軸番号) MD 30550: AXCONF_ASSIGN_MASTER_CHAN (軸置換用のチャンネルの初期設定)	
例外，エラー .....		

### 2.6.6 例

なし

### 2.6.7 データフィールド、リスト

#### データの参照

このファンクション説明に取り上げられていないデータについては関連参照を参照してください（例えば、/K/とあれば、ファンクション説明 (K1) を参照してください）。

#### ■ 一般マシンデータ

番号	識別子	名称	参照
一般事項 (\$MN_ ... )			
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	K1

#### ■ チャンネルマシンデータ

#### ■ 基本チャンネルマシンデータ

番号	識別子	名称	参照
チャンネル別 (\$MC_ ... )			
20000	CHAN_NAME	チャンネル名	K1
20050	AXCONF_GEOAX_ASSIGN_TAB[n]	ジオメトリ的軸のチャンネルへの割当て [GEO 軸番号]: 0...2	K2
20060	AXCONF_GEOAX_NAME_TAB[n]	チャンネルのジオメトリ的軸名 [GEO 軸番号]: 0...2	K2



番号	識別子	名称	参照
チャンネル別 (\$MC_ ... )			
20070	AXCONF_MACHAX_USED[n]	チャンネルで有効なマシン軸番号 [チャンネル軸番号]: 0...7	K2
20080	AXCONF_CHANAX_NAME_TAB[n]	チャンネルのチャンネル軸番号 [チャンネル軸番号]: 0...7	K2
20090	SPIND_DEF_MASTER_SPIND	チャンネルのマスタスピンドルの 初期設定	S1
20100	DIAMETER_AX_DEF	フェーシング軸ファンクションを 使用するジオメトリ軸	P1
20150	GCODE_RESET_VALUES[n]	G グループの初期設定 [G グループ番号]: 0...59	K1
20160	CUBIC_SPLINE_BLOCKS	C スプラインを使用するブロックの番号	K1
20170	COMPRESS_BLOCK_PATH_LIMIT	圧縮を使用する NC ブロックの 最大移動長さ	K1
20200	CHFRND_MAXNUM_DUMMY_BLOCK	フェーズ／半径を使用する エントリブロック	K1
20210	CUTCOM_CORNER_LIMIT	TRC を使用する補償用の最大角度	W1
20220	CUTCOM_MAX_DISC	DISC 用の最大値	W1
20230	CUTCOM_CURVE_INSERT_LIMIT	TRC を使用するインタセクション 計算用の最大角度	W1
20240	CUTCOM_MAXNUM_CHECK_BLOCKS	TRC を使用する輪郭計算用のブロック	W1
20250	CUTCOM_MAXNUM_DUMMY_BLOCKS	TRC を使用するトラバース動作を 持たないブロック番号	W1
20270	CUTTING_EDGE_DEFAULT	プログラミングのないツール 切削エッジの基本設定	W1
20400	LOOKAH_USE_VELO_NEXT_BLOCK	プログラムされている追従ブロック速度 でのルックアヘッド	B1
20430	LOOKAH_NUM_OVR_POINTS	準備されたオーバーライド速度特性の番号 ルックアヘッドを使用する	B1
20440	LOOKAH_OVR_POINTS[n]	ルックアヘッドを使用する準備された オーバーライド速度特性 [特性番号]: 0...1	B1
20500	CONST_VELO_MIN_TIME	一定速度を使用する最小時間	B2
20600	MAX_PATH_JERK	パス関連の最大ジャーク	B2
20610	ADD_MOVE_ACCEL_RESERVE	オーバレイされた動作用の加速予約	K1
20650	THREAD_START_IS_HARD	スレッド切削を使用する軸の加速挙動	K1
20700	REFP_NC_START_LOCK	基準点を使用しない NC スタート ディスエーブル	R1
20750	ALLOW_GO_IN_G96	G96 を使用する GO ロジック	V1
20800	SPF_END_TO_VDI	PLC へのサブプログラムエンド	H2
21000	CIRCLE_ERROR_CONST	サークルエンド監視定数	K1
21010	CIRCLE_ERROR_FACTOR	サークルエンド監視係数	K1
21100	ORIENTATION_IS_EULER	方向プログラミング用の角度定義	F2
21110	X_AXIS_IN_OLD_X_Z_PLANE	自動フレーム定義を使用する寸法系	K2
21200	LIFTFAST_DIST	輪郭からファストリフティングを使用して トラバースされた距離	K1

番号	識別子	名称	参照
21250	START_INDEX_R_PARAM	最初のチャンネル別 R パラメータの番号	S7

### ■ チャンネルの補助ファンクション設定

番号	識別子	名称	参照
チャンネル別 (\$MC_ ... )			
22000	AUXFU_ASSIGN_GROUP[n]	補助ファンクショングループ [ チャンネルの補助ファンクション 番号 ]: 0...49	H2
22010	AUXFU_ASSIGN_TYPE[n]	補助ファンクションタイプ [ チャンネルの補助ファンクション 番号 ]: 0...49	H2
22020	AUXFU_ASSIGN_EXTENSION[n]	補助ファンクションエクステンション [ チャンネルの補助ファンクション 番号 ]: 0...49	H2
22030	AUXFU_ASSIGN_VALUE[n]	補助ファンクション値 [ チャンネルの補助ファンクション 番号 ]: 0...49	H2
22200	AUXFU_M_SYNC_TYPE	M ファンクションの出力時間	H2
22210	AUXFU_S_SYNC_TYPE	S ファンクションの出力時間	H2
22220	AUXFU_T_SYNC_TYPE	T ファンクションの出力時間	H2
22230	AUXFU_H_SYNC_TYPE	H ファンクションの出力時間	H2
22240	AUXFU_F_SYNC_TYPE	F ファンクションの出力時間	H2
22250	AUXFU_D_SYNC_TYPE	D ファンクションの出力時間	H2
22260	AUXFU_E_SYNC_TYPE ( 間もなく利用可能 )	E ファンクションの出力時間	-
22300	AUXFU_AT_BLOCK_SEARCH_END	ブロックサーチ実行後の補助ファンク ション出力	H2
22400	S_VALUES_ACTIVE_AFTER_RESET	RESET 後でも有効な S ファンクション	S1
22410	F_VALUES_ACTIVE_AFTER_RESET	RESET 後でも有効な F ファンクション	V1
22500	GCODE_OUTPUT_TO_PLC	PLC への G ファンクション	K1
22550	TOOL_CHANGE_MODE	M ファンクションを使用する新規ツール オフセット	W1
22560	TOOL_CHANGE_M_CODE	ツール変更用の M ファンクション	W1

## ■ チャンネル中での変換定義

番号	識別子	名称	参照
チャンネル別 (\$MC_... )			
24100	TRAFO_TYPE_1	チャンネルの変換 1 の定義	F2
24110	TRAFO_AXES_IN_1[n]	変換のための軸割当て [ 軸インデックス ]: 0...7	F2
24120	TRAFO_GEOAX_ASSIGN_TAB_1[n]	変換 1 のためのチャンネル軸への GEO 軸の割当て [GEO 軸番号 ]: 0...2	F2
24200	TRAFO_TYPE_2	チャンネルの変換 2 の定義	F2
24210	TRAFO_AXES_IN_2[n]	変換 2 用の軸割当て [ 軸インデックス ]: 0...7	F2
24220	TRAFO_GEOAX_ASSIGN_TAB_2[n]	変換 2 のためのチャンネル軸への GEO 軸の割当て [GEO 軸番号 ]: 0...2	F2
24300	TRAFO_TYPE_3	チャンネルの変換 3 の定義	F2
24310	TRAFO_AXES_IN_3[n]	変換 3 用の軸の割当て [ 軸インデックス ]: 0...7	F2
24320	TRAFO_GEOAX_ASSIGN_TAB_3[n]	変換 3 のためのチャンネル軸への GEO 軸の割当て [GEO 軸番号 ]: 0...2	F2
24400	TRAFO_TYPE_4	チャンネルの変換 4 の定義	F2
24410	TRAFO_AXES_IN_4[n]	変換 4 用の軸の割当て [ 軸インデックス ]: 0...7	F2
24420	TRAFO_GEOAX_ASSIGN_TAB_4[n]	変換 4 のためのチャンネル軸への GEO 軸の割当て [GEO 軸番号 ]: 0...2	F2
24430	TRAFO_TYPE_5	チャンネルの変換 5 の定義	F2
24432	TRAFO_AXES_IN_5[n]	変換 5 用の軸割当て [ 軸インデックス ]: 0...7	F2
24434	TRAFO_GEOAX_ASSIGN_TAB_5[n]	変換 5 のためのチャンネル軸への GEO 軸の割当て [GEO 軸番号 ]: 0...2	F2, M1
24440	TRAFO_TYPE_6	チャンネルの変換 6 の定義	F2
24442	TRAFO_AXES_IN_6[n]	変換 6 用の軸の割当て [ 軸インデックス ]: 0...7	F2
24444	TRAFO_GEOAX_ASSIGN_TAB_6[n]	変換 6 のためのチャンネル軸への GEO 軸の割当て [GEO 軸番号 ]: 0...2	F2, M1
24450	TRAFO_TYPE_7	チャンネルの変換 7 の定義	F2
24452	TRAFO_AXES_IN_7[n]	変換 7 用の軸の割当て [ 軸インデックス ]: 0...7	F2
24454	TRAFO_GEOAX_ASSIGN_TAB_7[n]	変換 7 のためのチャンネル軸への GEO 軸の割当て [GEO 軸番号 ]: 0...2	F2, M1
24460	TRAFO_TYPE_8	チャンネルの変換 8 の定義	F2
24462	TRAFO_AXES_IN_8[n]	変換 8 用の軸の割当て [ 軸インデックス ]: 0...7	F2
24464	TRAFO_GEOAX_ASSIGN_TAB_8[n]	変換 8 のためのチャンネル軸への GEO 軸の割当て [GEO 軸番号 ]: 0...2	F2, M1
24500	TRAFO5_PART_OFFSET_1[n]	5 軸変換 1 用のオフセットベクトル [ 軸番号 ]: 0...2	F2
24510	TRAFO5_ROT_AX_OFFSET_1[n]	5 軸変換 1 用の用の回転軸 1/2 の 位置オフセット [ 軸番号 ]: 0...1	F2

番号	識別子	名称	参照
24520	TRAFO5_ROT_SIGN_IS_PLUS_1[n]	5 軸変換 1 用の用の回転軸 1/2 の符号 [ 軸番号 ]: 0...1	F2
24530	TRAFO5_NON_POLE_LIMIT_1	5 軸変換 1 用の極値範囲の定義	F2
24540	TRAFO5_POLE_LIMIT_1	5 軸変換 1 用の極補間を使用する制限角度公差	F2
24550	TRAFO5_BASE_TOOL_1[n]	5 軸変換 1 用の起動の基本ツールのベクトル [ 軸番号 ]: 0...2	F2
24560	TRAFO5_JOINT_OFFSET_1[n]	5 軸変換 1 用の運動オフセットのベクトル [ 軸番号 ]: 0...2	F2
24600	TRAFO5_PART_OFFSET_2[n]	5 軸変換 2 用のオフセットベクトル [ 軸番号 ]: 0...2	F2
24610	TRAFO5_ROT_AX_OFFSET_2[n]	5 軸変換 2 用の回転軸 1/2 の位置オフセット [ 軸番号 ]: 0...1	F2
24620	TRAFO5_ROT_SIGN_IS_PLUS_2[n]	5 軸変換 2 用の回転軸 1/2 の符号 [ 軸番号 ]: 0...1	F2
24630	TRAFO5_NON_POLE_LIMIT_2	5 軸変換 2 用の極値範囲の定義	F2
24640	TRAFO5_POLE_LIMIT_2	5 軸変換 2 用の極値による補間を使用する制限角度公差	F2
24650	TRAFO5_BASE_TOOL_2[n]	5 軸変換 2 用の起動の基本ツールのベクトル [ 軸番号 ]: 0...2	F2
24660	TRAFO5_JOINT_OFFSET_2[n]	5 軸変換 2 用の運動オフセットのベクトル [ 軸番号 ]: 0...2	F2

## ■ チャンネル別メモリ設定

番号	識別子	名称	参照
チャンネル別 (\$MC_ ... )			
25000	REORG_LOG_LIMIT	ログファイルイネーブル用の IPO バッファのパーセント	S7
28000	MM_REORG_LOG_FILE_MEM	REORG (DRAM) 用のメモリサイズ	S7
28010	MM_NUM_REORG_LUD_MODULES	REORG (DRAM) を使用するローカルユーザ変数用のモジュール番号	S7
28020	MM_NUM_LUD_NAMES_TOTAL	ローカルユーザ変数の番号 (DRAM)	S7
28030	MM_NUM_LUD_NAMES_PER_PROG	プログラムごとのローカルユーザ変数の番号 (DRAM)	S7
28040	MM_LUD_VALUES_MEM	ローカルユーザ変数のメモリサイズ (DRAM)	S7
28050	MM_NUM_R_PARAM	軸別 R パラメータの番号 (SRAM)	S7
28060	MM_IPO_BUFFER_SIZE	IPO バッファの NC ブロックの番号 (DRAM)	S7
28070	MM_NUM_BLOCKS_IN_PREP	ブロック準備用のブロックの数 (DRAM)	S7
28080	MM_NUM_USER_FRAMES	セットするフレームの数 (SRAM)	S7
28090	MM_NUM_CC_BLOCK_ELEMENTS	コンパイルサイクル用のブロックエレメントの数 (DRAM)	S7
28100	MM_NUM_CC_BLOCK_USER_MEM	コンパイルサイクル用のブロックメモリのサイズ (DRAM)	S7
28500	MM_PREP_TASK_STACK_SIZE	準備タスクのスタックサイズ (DRAM)	S7
28510	MM_IPO_TASK_STACK_SIZE	IPO タスクのスタックサイズ (DRAM)	S7

## ■ 軸／スピンドル別マシンデータ

番号	識別子	名称	参照
軸／スピンドル別 (\$MA_ ... )			
30550	AXCONF_ASSIGN_MASTER_CHAN	軸変更用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 を使用する軸の固定値位置	K1
33100	COMPRESS_POS_TOL	補償を使用する最小ずれ	K1

## ■ チャンネル別設定データ

番号	識別子	名称	参照
チャンネル別 (\$SC_ ... )			
42000	THREAD_START_ANGLE	スレッドを使用するスタート角度	K1
42100	DRY_RUN_FEED	テストランフィード	V1

---

## ■ インタフェース信号

### ■ モードグループ信号

#### インタフェース信号の説明

PLC → NCK から送られたモードグループ信号および NCK → PLC から送られたモードグループ信号は、第 1 モードグループ用のデータブロック 11 に保存されます。この信号は次の参照に表示と説明があります。

参照： 1.10 運転モードの種類とプログラム運転モード (K1)

### ■ チャンネル信号

#### インタフェース信号の説明

PLC → NCK から送られたモードグループ信号および NCK → PLC から送られたモードグループ信号は、第 1, 第 2 ... チャンネル用のデータブロック 21, 22 に保存されます。この信号は次の参照に表示と説明があります。

参照： 1.10 運転モードの種類とプログラム運転モード (K1)

### ■ アラーム

発生する可能性があるアラームの詳細については、次の参照あるいは、オンラインヘルプを参照してください。

参照： アラーム一覧

## 2.7 座標変換機能 (M1)

### 2.7.1 概略説明

#### ■ TRANSMIT

TRANSMIT が提供する機能範囲は以下のとおりです：

- 回転クランプ内の回転パートに対するフェースエンド加工
  - 穴
  - 輪郭
- これらの加工オペレーションをプログラムするために直交座標系を使用することができます。
- 制御装置は直交座標系のプログラムによるトラバース動作を実際のマシン軸のトラバース動作に変換します（標準アプリケーション）：
  - 回転軸 (1)
  - 回転軸に対して垂直なインフィード軸 (2)
  - 回転軸に平行な縦軸 (3)直線軸 (2) および (3) は互いに直行しています。
- 回転中心を基準にしてツール中心のオフセットを設定できます。
- 速度制御は回転運動に関して規定されている制限値を考慮に入れています。
- 直交座標系内のパスは回転中心点を通ってはなりません（本制限事項はソフトウェアのバージョン 2 および 3 に適用されます）。

#### ソフトウェアバージョン 4 以降に追加された利点

- ツール中心点パスは回転軸の回転中心点を通ることができます。
- 回転軸はモジュロ軸である必要はありません。

## ■ TRACYL

TRACYL（円柱表面カーブ変換）の機能範囲は以下のとおりです：

- 円柱形物体上の縦溝の加工
- 円柱形物体上の横溝の加工
- 円柱形物体上の任意の溝パターンの加工

溝加工パスは展開された平面円柱生成表面に合わせてプログラムされています。

加工目的のために、当ファンクションは以下を有する旋盤をサポートしています：

- X-C-Z キネマティックス
- X-Y-Z-C キネマティックス。
- 制御装置によって、円柱座標系のプログラムされたトラバース動作が実際のマシン軸のトラバース動作に変換されます（標準アプリケーション X-C-Z キネマティックス）：
  - 回転軸 (1)
  - 回転軸に対して垂直なインフィード軸 (2)
  - 回転軸に対して平行な縦軸 (3)

（注）

直線軸 (2) および (3) は互いに直行しています。インフィード軸 (2) は回転軸と交差します。この配置では溝側補償はできません。

- 溝側補償には以下の軸による X-Y-Z-C キネマティックスが必要です：
  - 回転軸 (1)
  - 回転軸に垂直なインフィード軸 (2)
  - 回転軸に平行な縦軸 (3)
  - 右手直交座標系を得るために (2) および (3) を補う縦軸 (4)。

（注）

直線軸 (2), (3) および (4) は互いに直行しています。この配置では溝壁補正が可能です。

- 速度制御は回転運動に関して規定されている制限値を考慮に入れています。

## ■ TRAANG（傾斜軸）

「傾斜軸」ファンクションはグラインディングアプリケーション用です。その機能は以下のとおりです：

- 傾斜インフィード軸を使用した加工。
- プログラミングには直交座標系を用いることができます。
- この制御装置は直交座標系でプログラムされた動作を実際のマシン軸のトラバース動作に変換します（標準アプリケーション）：傾斜インフィード軸。



---

## ■ チェーニング変換

### 概要

ソフトウェアバージョン 5 以降では、二つの変換機能を、第一の変換機能からの軸動作パートが、連結された第二の変換機能の入力データになるように連結（チェーニング）できます。第二の変換機能による動作パートがマシン軸に作用します。

### チェーニングが使用できる場合

- ソフトウェアバージョン 5 では、チェーニングは二つの変換機能を含むことが可能です。
- 第二の変換機能は「傾斜軸」(TRAANG) でなければなりません。
- 第一の変換機能は次のいずれかでなければなりません：
  - 向き変換 (TRAORI), 万能フライスヘッドを含む
  - TRANSMIT
  - TRACYL
  - TRAANG

チェーニング変換についての詳細は 2.7.2 「■チェーニング変換」を、その他の変換機能に関する詳細は次を参照してください。

参照： 3.1 3 ～ 5 軸刃先方向制御 (F2)

## 2.7.2 詳細説明

### ■ TRANSMIT

(注)

以下に説明する TRANSMIT 変換では、この変換機能が有効な場合、マシン軸、チャンネル、およびジオメトリ軸にそれぞれ名前を付ける必要があります。

次を比較してください：

MD 10000: AXCONF\_MACHAX\_NAME\_TAB,

MD 20080: AXCONF\_CHANAX\_NAME\_TAB,

MD 20060: AXCONF\_GEOAX\_NAME\_TAB.

こうすることで明確な割当てが行えます。

### タスクの定義

完全な加工（図を参照）

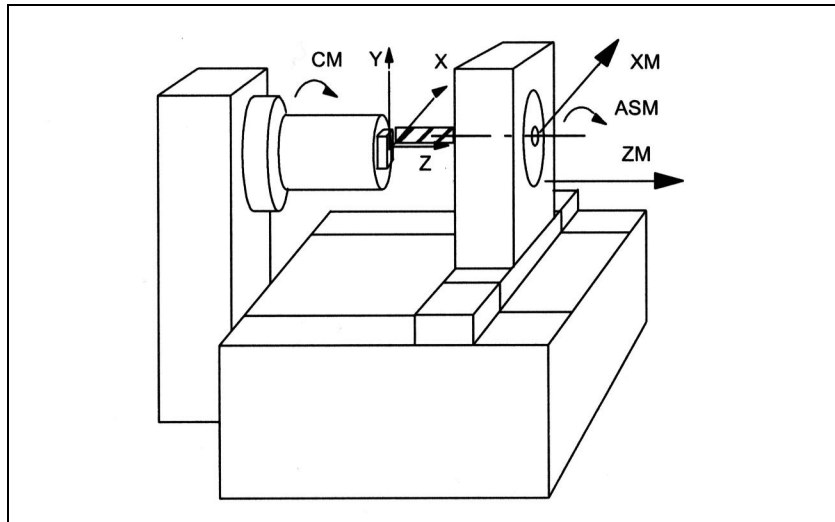


図 2.63 回転部分のエンドフェース加工

記号の意味：

CM	回転軸（主スピンドル）
ASM	加工スピンドル（フライスカッタ、ドリル）
X, Y, Z	フェースエンド加工オペレーションプログラミング用直交座標系 (フェースエンドの回転中心点における原点)
ZM	マシン軸（直線）
XM	マシン軸（直線）

## ■ TRANSMIT の前提条件

### 軸構成

(図 2.63 の X, Y, Z に従って) ユーザが直交座標系でプログラムするためには、制御装置にこの座標系と実際に存在するマシン軸 (CM, XM, ZM, ASM) との関係を設定する必要があります:

- ジオメトリ軸への名前の割当て
- チャンネル軸へのジオメトリ軸の割当て
  - 一般状態 (TRANSMIT は非アクティブ)
  - TRANSMIT アクティブ
- マシン軸番号へのチャンネル軸の割当て
- スピンドルの識別
- マシン軸名の割当て

「- TRANSMIT アクティブ」事項を除き、この手順は通常の軸構成手順と同じです。すでに一般的なステップをよくご存知の場合は、下記のステップリストの「チャンネル軸へのジオメトリ軸の割当て」ステップのみをお読みください。

参照: 1.11 軸構成及び座標系 (K2)

### 変換数

最大 4 個 (ソフトウェアバージョン 4 以降の場合は 8 個) の変換データブロックをシステム内の各チャンネルに対して定義できます。これらの変換マシンデータ名は \$MC\_TRAFO .. で始まり, ...\_n で終わります ("n" は 1 から 8 の数字を表します)。以下のセクションでこれらのデータについて説明します:

\$MC\_TRAFO\_TYPE\_n  
\$MC\_TRAFO\_GEOAX\_ASSIGN\_TAB\_n  
\$MC\_TRAFO\_AXES\_IN\_n.

### TRANSMIT 構造の数

チャンネル内の変換機能用の 8 つの許容データ構造の内の 2 つを TRANSMIT ファンクションに割当てることができます。\$MC\_TRAFO\_TYPE\_n で割当てられる値は "256" となります。

これら 2 つの TRANSMIT 変換では、以下のマシンデータを規定の方法で設定しなければなりません:

\$MC\_TRANSMIT\_ROT\_AX\_OFFSET\_t  
\$MC\_TRANSMIT\_ROT\_SIGN\_IS\_PLUS\_t  
\$MC\_TRANSMIT\_BASE\_TOOL\_t  
\$MC\_TRANSMIT\_POLE\_SIDE\_FIX\_t (ソフトウェアバージョン 4 以降)

この場合、"t" には宣言される TRANSMIT 変換の数を指定します (最大 2)。

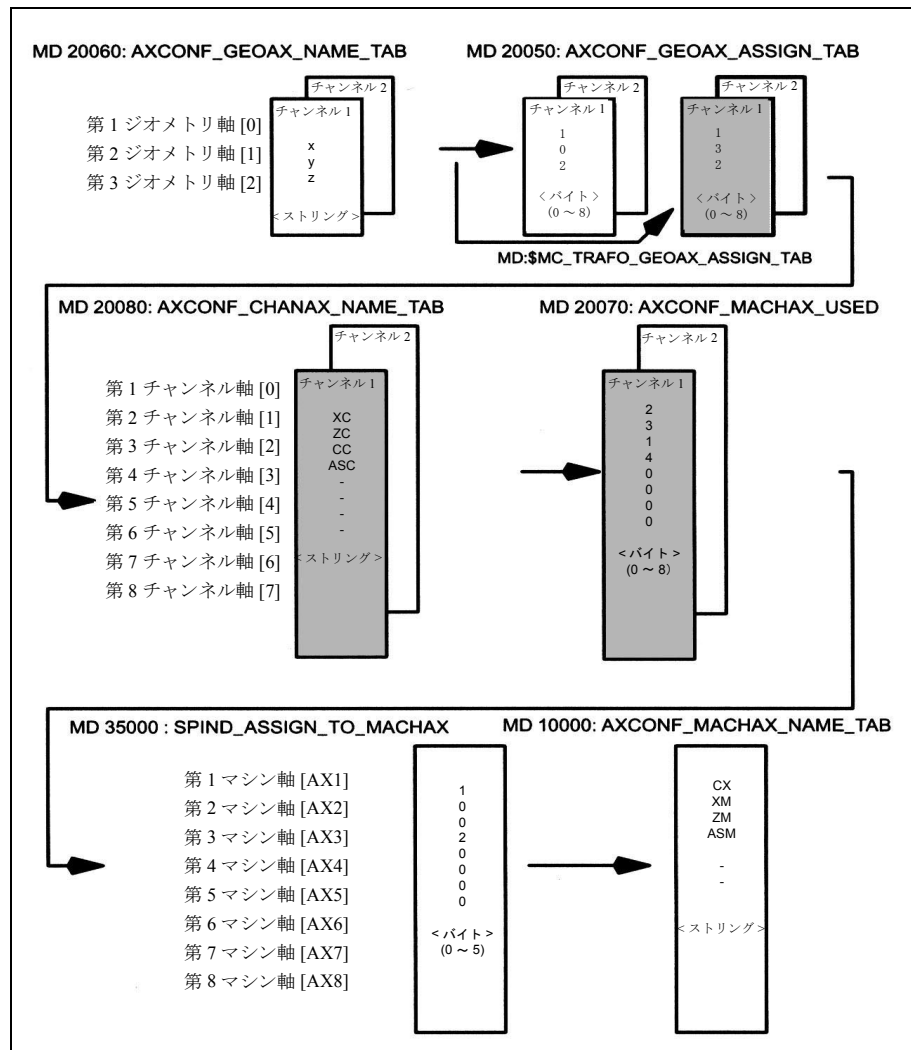


図 2.64 図 2.63 の例についての軸構成

図 2.64 に強調表示されている構成は TRANSMIT がアクティブの場合に適用されます。

## ジオメトリ軸の命名

上記の軸構成の概要に従って、TRANSMIT 動作に必要なジオメトリ軸を以下を使用して定義しなければなりません：

```
$MC_AXCONF_GEOAX_NAME_TAB[0]="X"
                        "_TAB[1]="Y"
                        "_TAB[2]="Z"
```

(図 2.64 による名前選択はデフォルト設定にも対応しています)。

## チャンネル軸へのジオメトリ軸の割当て

これらの割当ては TRANSMIT がアクティブであるかないかに応じて行われます：

### - TRANSMIT 非アクティブ

Y 軸は使用できません。

```
$MC_AXCONF_GEOAX_ASSIGN_TAB[0]=1  
                                "_TAB[1]=0  
                                "_TAB[2]=2
```

### - TRANSMIT アクティブ

Y 軸はパートプログラムで指定可能です。

```
$MC_TRAFO_GEOAX_ASSIGN_TAB_1[0]=1  
                                "_TAB_1[1]=3  
                                "_TAB_1[2]=2
```

Y 軸はチャンネル軸の第 3 エントリです。

## チャンネル軸の入力

直交座標系に属さない軸を入力します。

```
$MC_AXCONF_CHANAX_NAME_TAB[0]="XC"  
                                "[1]="ZC"  
                                "[2]="CC"  
                                "[3]="ASC"
```

## マシン軸へのチャンネル軸の割当て

チャンネル軸の cd を基準として、チャンネル軸の割当てられているマシン軸番号が制御装置に転送されます。

```
$MC_AXCONF_MACHAX_USED[0]=2  
                                "[1]=3  
                                "[2]=1  
                                "[3]=4  
                                "[4]=0
```

(図 2.64 に対応するエントリ)

## スピンドルの識別

ユーザは各マシン軸がスピンドル (値 > 0: スピンドル番号) かあるいは送り軸 (値 0) かを定義します。

```
$MA_SPIND_ASSIGN_TO_MACHAX[0]=1  
                                "[1]=0  
                                "[2]=0  
                                "[3]=2
```

## マシン軸への名前の割当て

マシン軸の cd を基準として、マシン軸名が制御装置に転送されます。

```
$MN_AXCONF_MACHAX_NAME_TAB[0]="CM"  
                                "[1]="XM"  
                                "[2]="ZM"  
                                "[3]="ASM"
```

## ■ TRANSMIT に固有の設定値

### 変換タイプ

以下に、変換タイプがどのように指定されるかを説明します。

### TRAFO\_TYPE\_n

ユーザは変換データブロックの変換タイプを指定しなければなりません（最大 n = 8）。TRANSMIT の場合は、256 という値を設定しなければなりません。

例 : \$MC\_TRAFO\_TYPE\_1=256

この設定を行い電源 ON でこの値を起動してから、TRANSMIT あるいは TRANSMIT(t) を呼び出さなければなりません。"t" は宣言された TRANSMIT 変換の番号です。

### 軸イメージ

以下のパラグラフでは変換軸イメージがどのように指定されるかを説明します。

### TRAFO\_AXES\_IN\_n

3 つのチャンネル軸番号を変換データブロック n に指定しなければなりません：

```
$MC_TRAFO_AXES_IN_1[0]=  回転軸に垂直な軸のチャンネル軸番号  
$MC_TRAFO_AXES_IN_1[1]=  回転軸のチャンネル軸番号  
$MC_TRAFO_AXES_IN_1[2]=  回転軸に平行な軸のチャンネル軸番号
```

図 2.64 に示す構成の例：

```
$MC_TRAFO_AXES_IN_1[0]=1  
$MC_TRAFO_AXES_IN_1[1]=3  
$MC_TRAFO_AXES_IN_1[2]=2
```

この設定は TRANSMIT あるいは TRANSMIT(t) をコールする前に行わなければなりません。軸番号は \$MC\_TRAFO\_GEOAX\_ASSIGN\_TAB\_n で定義されているチャンネル軸のシーケンスに対応していなければなりません。

### 回転の中心

直交座標系の回転の中心は下記に記述されているマシンデータで指定されます。

## TRANSMIT\_ROT\_AX\_OFFSET\_t

回転軸の定義済み原点位置を基準として、直交座標系 x-y 平面の回転の中心が下記で指定されます：

\$MC\_TRANSMIT\_ROT\_AX\_OFFSET\_t=... ; 度

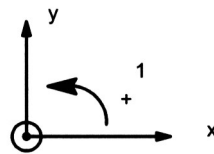
この場合、"t" は変換データブロックで宣言されている TRANSMIT 変換の数で置き換えられます (t は最大 2)。

## 回転方向

回転軸の回転方向は以下に記述されているマシンデータで指定されます。

## TRANSMIT\_ROT\_SIGN\_IS\_PLUS\_t

z 軸方向から見て回転軸が x-y 平面上を反時計方向に回転する場合は、マシン軸は "1" に設定しなければならず、そうでない場合は "0" に設定しなければなりません。



\$MC\_TRANSMIT\_ROT\_SIGN\_IS\_PLUS\_t=1

この場合、"t" は変換データブロックで宣言済みの TRANSMIT 変換の数で置き換えられます (t は最大 2)。

## ツール原点の位置

ツール原点の位置は以下のパラグラフに記述されているマシンデータで指定されます。

## TRANSMIT\_BASE\_TOOL\_t

マシンデータ \$MC\_TRANSMIT\_BASE\_TOOL\_t は、TRANSMIT に対して宣言されている座標系の原点を基準としてツール原点の位置を制御装置に知らせるために用いられます。このマシンデータは直交座標系の 3 つの軸用に 3 つの構成要素を有しています。

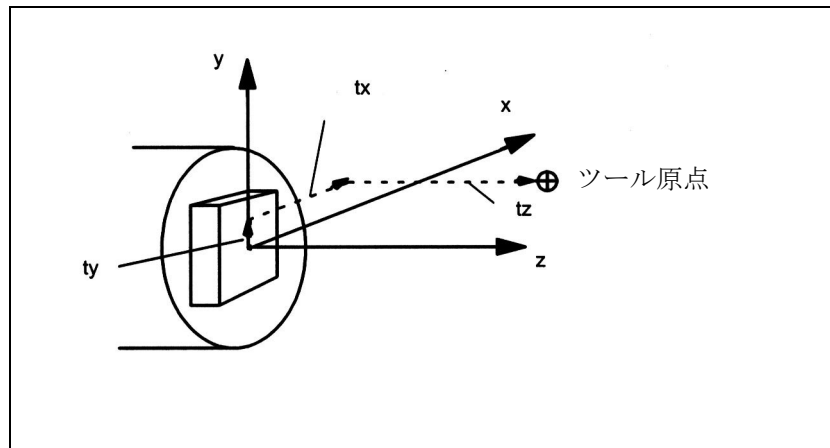


図 2.65 直交座標軸の原点を基準としたツール原点の位置（図 2.63 参照）

```
$MC_TRANSMIT_BASE_TOOL_t[0]=tx  
    " [1]=ty  
    " [2]=tz
```

この場合、インデックス値 [ ] の前の t は変換データブロックで宣言されている TRANSMIT 変換の数で置き換えられます。(t は最大 2)。

### 置き換え可能なジオメトリ軸

ジオメトリ軸が GEOAX() を使用して置き換えられた場合、M コード出力（マシンデータで設定可能）を通してそのことが PLC に通知されます。

- MD 22534: TRAFO\_CHANGE\_M\_CODE

変換切換えの場合に VDI インタフェースに出力される M コードの番号。

(注)

このマシンデータが 0 ～ 6, 17 もしくは 30 のうちのいずれかに設定された場合は、M コードは出力されません。

参照： 1.11 軸構成及び座標系 (K2)



## ■ TRANSMIT の起動

### TRANSMIT

前述の設定が実行されたなら、TRANSMIT ファンクションを起動することができます：

TRANSMIT あるいは

TRANSMIT(t)

1 番目に宣言された TRANSMIT ファンクションは TRANSMIT で起動されます。

TRANSMIT(t) は t 番目に宣言された TRANSMIT ファンクションを起動します。t は最大 2 です。

このファンクションが起動されると、2.7.2「■ソフトウェアバージョン 4 の TRANSMIT によるオーバーレイ動作」で説明されているような付近の移動、極などの特別な手順も同様にソフトウェアバージョン 4 以降に関して利用可能となります。以下に説明するファンクションの起動と停止の間で、直交座標系の軸のトラバース動作をプログラムすることができます。

## ■ TRANSMIT ファンクションの停止

### TRAFOOF

コード TRAFOOF はアクティブ状態の変換機能を停止します。変換機能が停止されると、基本座標系は再びマシン座標系と一致します。アクティブ状態の TRANSMIT 変換機能は、他の変換機能（例えば TRACYL, TRAANG, TRAORI など）の一つが適切なチャンネル内で起動された場合、同様に停止されます。

参照： 3.1 3 ～ 5 軸刃先方向制御 (F2)

## ■ TRANSMIT に関する特別なシステム応答

この変換はパートプログラムあるいは MDA によって選択あるいは選択解除可能です。

## 選択に関する注意事項

- 中間動作ブロックは挿入されません（フェーズ／半径）。
- スプラインブロックシーケンスは終了させなければなりません。
- ツール半径補償を解除しなければなりません。
- TRANSMIT より前にアクティブ状態にあったフレームは制御装置により解除されます（"Reset programmed frame" G500 に対応）。
- この変換機能が作用を及ぼす軸に関しては、使用中の作業エリア制限は制御装置によって解除されます（プログラムされた WALIMOF に対応）。
- 連続的なパス制御や丸め処理は中断されます。
- 変換された軸の DRF オフセットはオペレータによって削除されていなければなりません。

## 選択解除に関する注意事項

- 中間動作ブロックは挿入されません（フェーズ／半径）。
- スプラインブロックシーケンスは終了させなければなりません。
- ツール半径補償は解除しなければなりません。
- TRANSMIT より前にアクティブ状態にあったフレームは制御装置により解除されます。（"Reset programmed frame" G500 に対応）。
- 起動されたツール長補償は制御装置によりこの変換動作に組み込まれます。
- 連続的なパス制御や丸め処理は中断されます。
- 変換された軸の DRF オフセットはオペレータが削除していなければなりません。
- 仮想軸（図 2.63 の Y 軸）におけるツール長補償は実行されません。

## アクティブな TRANSMIT によって課せられる制限事項

起動された TRANSMIT ファンクションによって課せられる下記の制限事項に注意してください。

### ツール変更

ツールは、ツール半径補償ファンクションが解除された場合のみ変更可能です。

### フレーム

基本座標系のみに関係する命令はすべて許可されます（FRAME、ツール半径補償）。ただし非アクティブな変換機能に関する手順と異なり、G91 によるフレーム変更（相対寸法）は特別扱いされません。トラバースされる相対距離は、前のブロックでどのフレームが有効だったかにかかわらず、新しいフレームのワーク座標系で評価されます。

### 回転軸

回転軸はジオメトリ軸に占有されているのでプログラムすることができず、したがってチャンネル軸のように直接プログラムすることはできません。

## 極

ソフトウェアバージョン 3.x 以前：

極（直交座標系の原点）を通る動作は無効です、即ち極をトラバースする動作は極で停止し、アラームが出力されます。カッタ中心オフセットの場合、この動作は接近不能領域の端で終了します。

ソフトウェアバージョン 4 以降：

極トラバースおよび極近くでの加工オペレーションに関するオプションは 2.7.2 「■ソフトウェアバージョン 4 の TRANSMIT によるオーバーレイ動作」以降で説明します。

## 例外事項

この変換機能が作用を及ぼす軸は以下の手段や目的のために使用することはできません：

- ・ 事前設定軸として（アラーム）
- ・ 固定点アプローチ用に（アラーム）
- ・ 基準用に（アラーム）

（注）

TRANSMIT はツールの半径補償や長さ補償が解除される前に解除しておかなければなりません。

## 速度制御

TRANSMIT に関する速度の監視は前処理段階において標準機能として実行されます。メインランにおける監視および制限機能は以下の状態において有効です：

- ・ AUTOMATIC モードにおいて  
ただしマシンデータ \$MC\_TRAFO\_AXES\_IN\_n インデックス 0 または 1 によってこの変換機能に含まれる位置決め軸あるいは発振軸がプログラムされている場合。
- ・ JOG モードへの切換え時  
監視ファンクションは、この変換プロセスに関係する軸がパス軸としてオペレーションされる場合、メインランから前処理ルーチンに逆転送されます。前処理における速度監視ファンクションは、メインランにおける監視ファンクションよりも上手にマシンを活用します。またメインラン監視ファンクションはブロック先読みを停止します。

## パートプログラムの中断

パートプログラム処理が JOG のために中断される場合は、以下のことに注意しなければなりません：

## JOG

JOG に切換えられると、ソフトウェアバージョン 4 以降の場合、2.7.2 「■ソフトウェアバージョン 4 の TRANSMIT によるオーバーレイ動作」で説明されている最適化速度チェック機能に代わって従来のオンライン速度チェック機能が有効となります。

## AUTOMATIC から JOG へ

この変換機能が有効の時にパートプログラム処理が中断されて JOG モードでのトラバース動作が行われる場合は、AUTOMATIC の再度選択に際しては以下のことに注意しなければなりません：

- この変換機能は、現在位置から中断地点までアプローチブロックでアクティブ状態にあります。衝突に関する監視は全く行われません。



### 警告

オペレータはツールが再配置できるようにする責任があります。

## AUTOMATIC モードにおいて

速度最適化計画ファンクション（ソフトウェアバージョン 4）は、この変換動作に関係する軸がパス軸として相互同期でトラバースされる限り、アクティブ状態にあります。この変換動作に関係する軸が位置決め軸としてトラバースされる場合は、この変換動作が停止されるか、あるいはこの変換動作に関係する全ての軸が再度パス軸として作動するまで、このオンライン速度チェック機能はアクティブ状態にあります。2.7.2「■ソフトウェアバージョン 4 の TRANSMIT によるオーバーレイ動作」で説明しているように速度最適化動作への復帰がなされると自動的に STOPRE および非循環ブロック処理と補間処理の同期化機能が起動されます。

## リセット後のスタートから

パートプログラム処理が RESET で中止されて START で再始動された場合は、以下のことに注意しなければなりません：

- 全ての軸がパートプログラムの開始点にある直線ブロック（G0 または G1）によって規定の位置にトラバースさせられた場合のみ、残りのパートプログラムは再現可能なようにトラバースされます。RESET 時点でアクティブだったツールはもはや制御装置によって考慮されないようにすることもできます（マシンデータで設定可能）。

## 電源オンリセット

電源オン後のシステム応答は MD 20110: RESET\_MODE\_MASK および MD 20140: TRAFO\_RESET\_VALUE に保存されている設定値によって決まります。

参照： 1.11 軸構成及び座標系 (K2)

## 基準点アプローチ

変換機能がアクティブの時は軸を基準化することはできません。基準化動作中はアクティブ状態のいかなる変換機能も制御装置によって解除されます。

## ■ ソフトウェアバージョン 4.x 以降の TRANSMIT による加工

### 概要

TRANSMIT 変換の極は TRANSMIT 平面の原点にあります（図を参照：2-1,  $x=0$ ,  $y=0$ ）。この極はラジアル直線軸と回転軸（X と CM）との交点上に位置しています。極付近では、通常はジオメトリ軸のわずかな位置変更がマシン回転軸の大きな位置変更をもたらします。唯一の例外は極内への、あるいは極を通る直線動作です。

ソフトウェアバージョン 4 以降の場合、極を通るツール中心点パスがパートプログラムの中止を引き起こすことはありません。プログラム可能なトラバース動作コマンドやアクティブ状態のツール半径補償機能に関しては何らの制限事項もありません。しかし、極付近でのワーク加工動作はお勧めできません。これは、回転軸の過負荷を防止するためにフィードレートの急激な低下を必要とする場合があるからです。

### 新機能

定義：

ツール中心点によって描かれる線が回転軸の回転中心と交差する場合は極が存在します。

下記の事項について調べます：

- 極をトラバースすることができる場合の条件および方法
- 極付近での応答
- 作業エリア制限に関する応答
- $360^\circ$  を超える回転軸の回転の監視

### 極トラバース

極は二つの方法でトラバースすることができます：

- 直線軸に沿ったトラバース
- 極内へのトラバース（回転軸が極内で回転する）

### 直線軸に沿ったトラバース

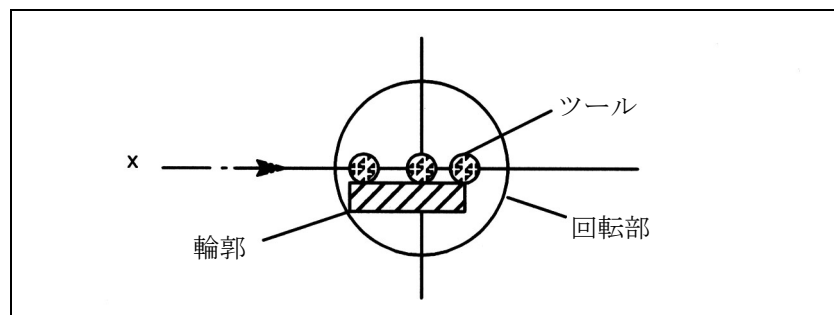


図 2.66 極を通る x 軸のトラバース

## 極内での回転

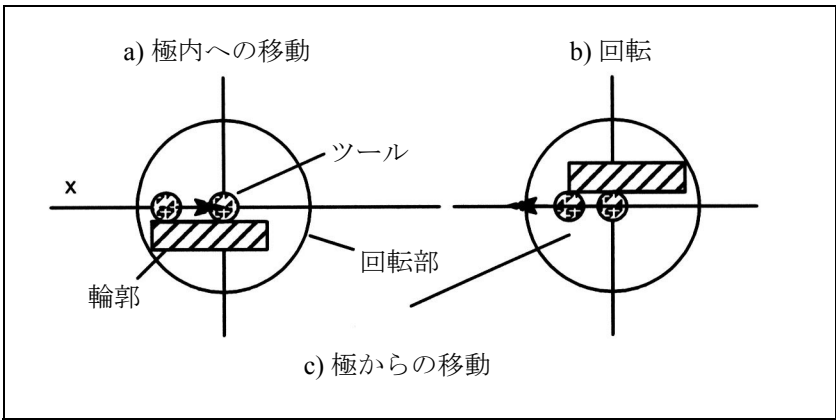


図 2.67 極内への x 軸のトラバース (a), 回転 (b), 極からの退出 (c)

## 方法の選択

方法はマシンの能力と被加工部品の要求条件に応じて選択しなければなりません。  
方法はマシンデータで選択します：

MD 24911: TRANSMIT\_POLE\_SIDE\_FIX\_1

MD 24951: TRANSMIT\_POLE\_SIDE\_FIX\_2

最初の MD はチャンネル内の 最初の TRANSMIT 変換に適用され、次の MD は同様に次の TRANSMIT 変換に適用されます。

表 2.7

値	意味
0	極トラバースツール中心点パス（直線軸）は連続するパス上で極をトラバースしなければならない。
1	極周りの回転。ツール中心点パスは直線軸の正のトラバース動作範囲（回転中心の前部）に限定しなければならない。
2	極周りの回転。ツール中心点パスは直線軸の負のトラバース 動作範囲（回転中心後部）に限定しなければならない。

## 極トラバースに関する特筆事項

直線軸に沿っての極トラバースの方法は AUTOMATIC および JOG モードにおいて適用可能です。

システム応答：下表のとおり。

表 2.8 直線軸に沿っての極のトラバース

動作モード	状態	反応
AUTOMATIC	変換に関係する全ての軸が同時に動かされる。TRANSMIT はアクティブ。	高速極トラバース
	変換に関係する軸がすべて同時にトラバースされるわけではない（例えば位置決め軸など）。TRANSMIT は非アクティブ。	徐行速度での極のトラバース動作
	適用されている DRF（外部原点オフセット）はこの動作を妨害しない。DRF の適用中は極近くでサーボエラーが発生する可能性がある。	加工動作の中止，アラーム
JOG	-	徐行速度での極のトラバース

### 極内での回転に関する注意事項

前提条件：この方法は AUTOMATIC モードにおいてのみ有効です。

MD 24911: TRANSMIT\_POLE\_SIDE\_FIX\_1 = 1 または 2

MD 24951: TRANSMIT\_POLE\_SIDE\_FIX\_2 = 1 または 2

値：1 直線軸は正のトラバース動作範囲内にとどまっている

値：2 直線軸は負のトラバース動作範囲内にとどまっている

極をツール中心点パスに沿ってトラバースする必要がある輪郭の場合は、直線軸が回転中心を超える範囲でトラバースするのを防止するために、次の 3 つのステップが実行されます：

ステップ	動作
1	直線軸は極内へトラバースする
2	回転軸は 180° 回転し、変換に関係するその他の軸は静止したまま。
3	残りのブロックの実行。直線軸はここで再び極を出る。

JOG モードでは、この動作は極で停止します。このモードの場合、この軸はそれが極に接近するパス接線に沿ってのみ極から出ることが可能です。その他の動作命令はすべて回転軸位置に関するステップ変更を必要とするか、あるいは最小動作命令の場合はより大きなマシン動作を必要とするかもしれません。それらはアラーム 21619 で拒絶されます。

### 極に接近したトラバース

ツール中心点が極の先でトラバースする場合は、マシン軸の設定値 (MD 32000: MAX\_AX\_VELO[AX\*] および MD 32300: MAX\_AX\_ACCEL[AX\*]) を超えないような方法で制御装置が自動的にフィードレートとパス加速レートを落とします。パスが極に近ければ近いほど、フィードレートの低下率は大きくなります。

## コーナが極内にある状態でのツール中心点パス

極内にコーナがあるツール中心点パスは軸速度にステップ変更を引き起こすだけでなく、回転軸位置にもステップ変更を引き起こします。これらは減速によって変化させることはできません。

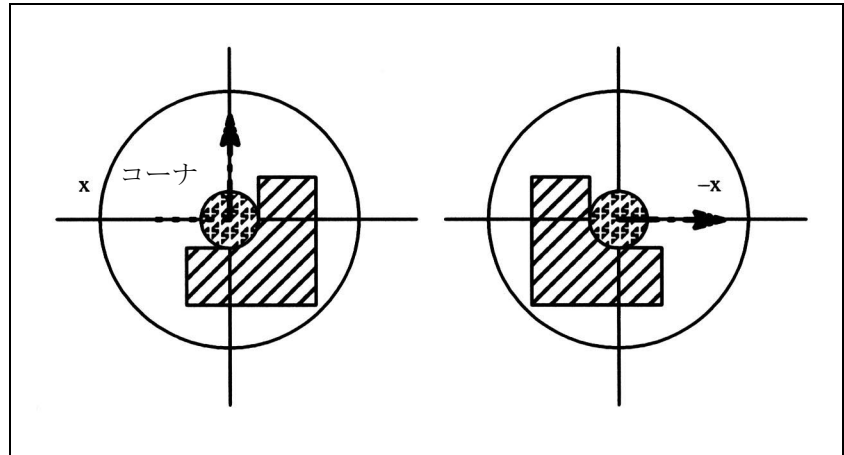


図 2.68 極トラバース

前提条件：

AUTOMATIC モード, MD 24911: TRANSMIT\_POLE\_SIDE\_FIX\_1 = 0 または  
MD 24951: TRANSMIT\_POLE\_SIDE\_FIX\_2 = 0

制御装置はステップ変更点でトラバース動作ブロックを挿入します。このブロックにより最小の回転が行われて輪郭の加工を続けることができます。

## 極トラバースのないコーナ

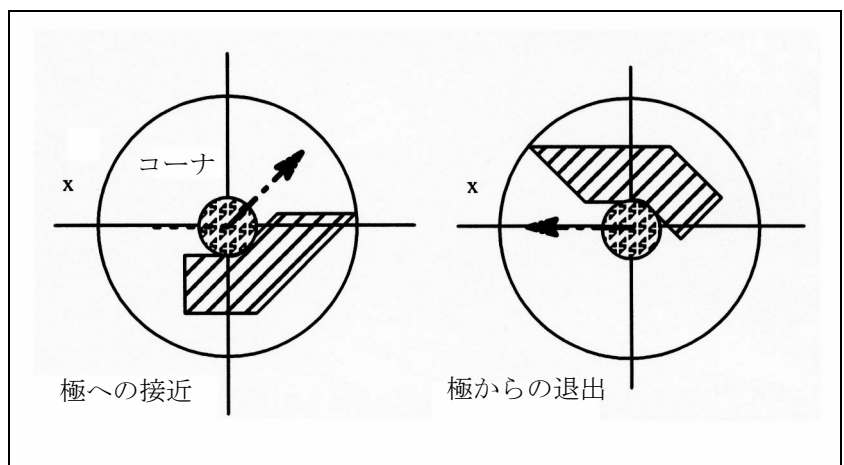


図 2.69 一方の極側での加工

前提条件：

AUTOMATIC モード,

MD 24911: TRANSMIT\_POLE\_SIDE\_FIX\_1 = 1 または 2 あるいは  
MD 24951: TRANSMIT\_POLE\_SIDE\_FIX\_2 = 1 または 2



---

制御装置はステップ変更点でトラバース動作ブロックを挿入します。このブロックにより、輪郭の加工が極の同じ側で続行できるように必要な回転が行われます。

### 極内での変換選択

加工動作が、起動された変換動作の極に対応するツール中心パス上の位置から継続しなければならない場合は、新たな変換動作のために極からの出口が指定されます。

MD 24911: TRANSMIT\_POLE\_SIDE\_FIX\_1=0 または

MD 24951: TRANSMIT\_POLE\_SIDE\_FIX\_2=0 が設定されている

(極トラバース) 場合は、

極からの退去を実行するブロックの開始で最小の回転が行われます。この回転に応じて、軸は次に回転中心の前あるいは後ろのどちらかでトラバースします。

MD 24911: TRANSMIT\_POLE\_SIDE\_FIX\_1=1 または

MD 24951: TRANSMIT\_POLE\_SIDE\_FIX\_2=1 の時は、加工動作は回転中心の前で続けられ (直線軸は正の制御範囲内) ;

MD 24911: TRANSMIT\_POLE\_SIDE\_FIX\_1=2 または

MD 24951: TRANSMIT\_POLE\_SIDE\_FIX\_2=2 の時は、加工動作は回転中心の後ろで続けられます (直線軸は負の制御範囲内)。

### 極外での変換選択

制御装置はマシンデータ \$MC\_TRANSMIT\_POLE\_SIDE\_FIX\_t の評価を行うことなく変換に関係する軸を移動させます。この場合、t=1 はチャンネル内の 1 番目の TRANSMIT 変換を、t=2 は 2 番目の TRANSMIT 変換を表します。

## ■ 作業エリア制限

### 初期状態

TRANSMIT がアクティブの時，この変換に関する回転軸の回転中心にツール中心点を配置することができない場合は，極は作業エリア制限と置き換えられます。これは回転軸に垂直な軸（ツールオフセットを考慮して）が回転軸と同じラジアル平面上に位置していない場合か，あるいは両方の軸が互いに斜角をなしている場合です。この2本の軸の間の距離がツールを位置づけることのできない BCS 内の円柱状のスペースを限定します。

マシン軸のトラバース動作範囲は影響されないで、不正な範囲をソフトウェアリミットスイッチ監視ファンクションで保護することはできません。

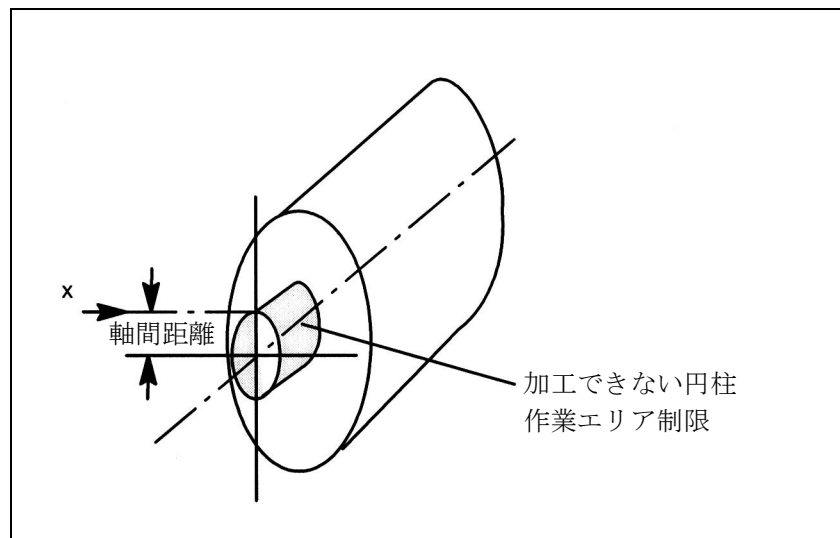


図 2.70 オフセット直線軸に基づく作業エリア制限

### 作業エリア制限内へのトラバース

作業エリア制限内へのいかなる動作もアラーム 21619 で拒絶されます。対応するパートプログラムブロックは処理されません。制御装置は先行するブロックの終わりで処理を停止します。

この動作が即座に予見できない場合は（JOG モード，位置決め軸），制御装置は作業エリア制限の端で停止します。

### 作業エリア制限近くでの応答

ツール中心点パスがこの不正範囲を通過する場合は，制御装置は自動的にフィードレートとパス加速レートを落として，マシンデータ軸の設定値 (MD 32000: MAX\_AX\_VELO[AX\*] および MD 32300: MAX\_AX\_ACCEL[AX\*]) が超えられないようにします。パスが作業エリア制限に近づけば近づくほど，フィードレートの低下率は大きくなります。

## ■ ソフトウェアバージョン 4 の TRANSMIT によるオーバレイ動作

制御装置はオーバレイ動作を予見することはできません。しかしながら、オーバレイ動作は、それらが極（あるいは作業エリア制限）からの現在の距離に比較して非常に小さい場合（例えばわずかなツールオフセットなど）は、ファンクションを妨害することはありません。変換に関係する軸に関しては、変換機能はオーバレイ動作を監視して、その量が危険な場合はすべてアラーム 21618 で知らせます。

このアラームは、ブロック関係速度計画ファンクションがもはやマシンの実条件に十分に対応していないことを示すものです。このアラームが出力されると、従来の、最適化されていないオンライン速度モニタが起動します。前処理ルーチンは制御装置内部で生成された REORG によるメインランと再同期化されます。

アラーム 21618 は、軸の過負荷に至り、したがってパートプログラムの中止に至る可能性があることを示しているため、可能な限りその発生を防止しなければなりません。

## ■ 360° を超える回転軸の回転の監視

### 回転軸位置のあいまいさ

回転軸の位置は回転数に関してはあいまいです。制御装置は極周りの複数の回転を含むブロックをサブブロックに再分割します。

プログラムされているブロックエンドはもはや同期化とは関係せず、代わりに最初のサブブロックの終わりが関係するので、並列アクション（例えば補助ファンクションの出力、ブロック同期位置決め軸動作など）に関してはこの再分割に注意しなければなりません。下記を参照してください：

参照： 1.9 「補助機能出力」 (H2)

単一ブロックモードでは、制御装置は個々のブロックを明示的に処理します。他のモードの場合、サブブロックはルックアヘッドで単一ブロックのようにトラバースされます。回転軸設定範囲の制限はソフトウェアリミットスイッチ監視ファンクションによって監視されます。

## ■ 補足条件

### ルックアヘッド

ルックアヘッド（極を通るトラバース動作、Look Ahead）を必要とするファンクションは、関係軸動作が前もって正確に計算され得る場合のみ満足に機能します。TRANSMIT に関しては、このことは回転軸とそれに垂直な直線軸に適用されます。もしこれらの軸の一方が位置決め軸の場合は、ルックアヘッドファンクションはアラーム 10912 によって停止され、代わりに従来のオンライン速度チェックファンクションが起動します。

### 方法の選択

ユーザは「極を通るトラバース動作」か「極周りの回転」かのいずれを選択する必要があります。ソフトウェアバージョン 2 および 3 で実行される極を通る軸のトラバースの自発的防止はソフトウェアバージョン 4 では削除されています。

---

## 複数回の極トラバース動作

ブロックは何度でも極をトラバースすることができます (例えば複数の折り返しを有する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数のサブブロックに再分割されます。同様に、複数回極を回転するブロックはサブブロックに分割されます。ソフトウェアバージョン 2 および 3 で適用されている関係制限事項はソフトウェアバージョン 4 では削除されています。

## モジュロ軸としての回転軸

回転軸はモジュール回転軸とすることができます。ただし、これはソフトウェアバージョン 2 および 3 とは異なって必須要件ではありません。ソフトウェアバージョン 2 および 3 で適用された関係制限事項はソフトウェア 4 では削除されています。

## スピンドルとしての回転軸

変換動作無しの回転軸がスピンドルとして使用される場合は、変換機能が選択される前に SPOS で位置制御モードに切り換えなければなりません。

## REPOS

ソフトウェアバージョン 4 の拡張 TRANSMIT ファンクションの結果として生成されたサブブロック上に再配置することができます。この場合、制御装置は BCS 内の再配置点に最も近い最初のサブブロックを用います。

## ブロックサーチ

計算によるブロックサーチの場合は、ソフトウェアバージョン 4 における拡張機能性の結果として中間ブロックが生成されているならば、(最後のブロックの) ブロック終点がアプローチされます。

## ■ TRACYL

(注)

以下で説明する TRACYL 変換は、この変換機能がアクティブの時、マシン軸、チャンネル、およびジオメトリ軸に固有の名前が割当てられている必要があります。以下を比較してください：

MD 10000: AXCONF\_MACHAX\_NAME\_TAB,

MD 20080: AXCONF\_CHANAX\_NAME\_TAB,

MD 20060: AXCONF\_GEOAX\_NAME\_TAB.

この方法でのみ明確な割当てを行うことができます。

### タスク定義

溝加工, 図を参照してください。

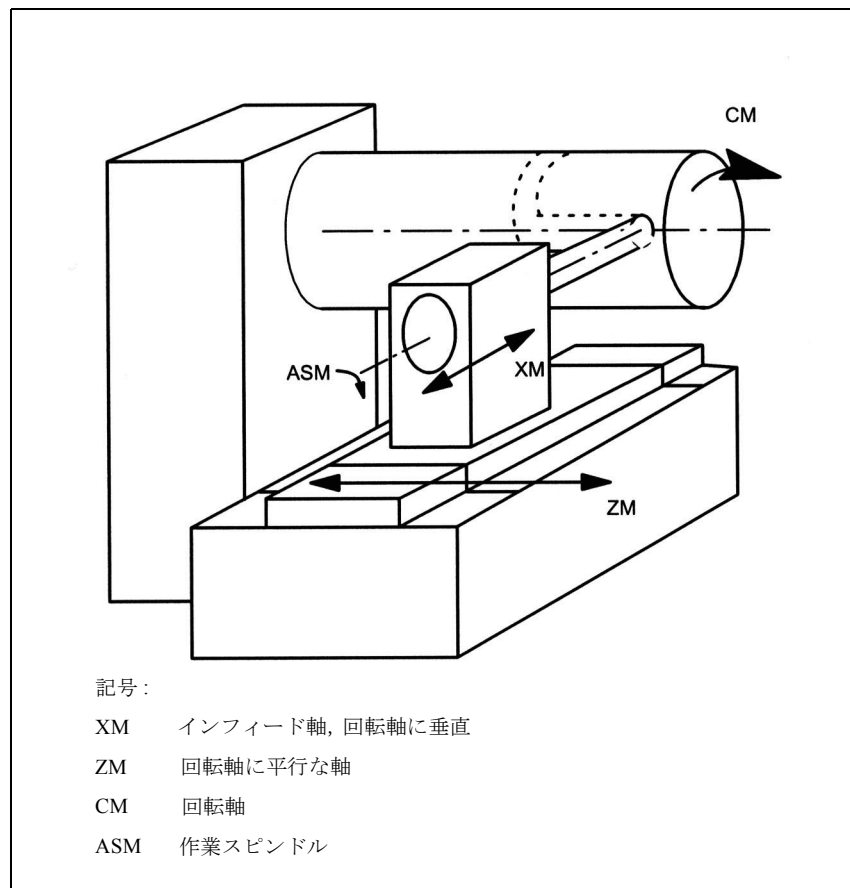


図 2.71 X-C-Z キネマティックスによる円柱面上の溝加工

## 軸構成 1

生成された円柱面カーブ変換機能により、円柱座標系の生成面を基準にしてトラバースパスを指定することができます。マシンのキネマティクスはこの円柱座標系に対応していなければなりません。これには1本ないし2本の直線軸、および回転軸が必要です。2本の直線軸は互いに垂直でなければなりません。回転軸は直線軸の一方と平行で、もう一方の直線軸と交差していなければなりません。さらに回転軸は円柱座標系と共通の軸を有していなければなりません。

直線軸が1本(X)のみの場合は、円柱の外周に平行な溝のみを生成することができます。直線軸が2本(X, Y)の場合、円柱上の溝パターンはオプションになります。図 2.72 を参照してください。

## 軸構成 2

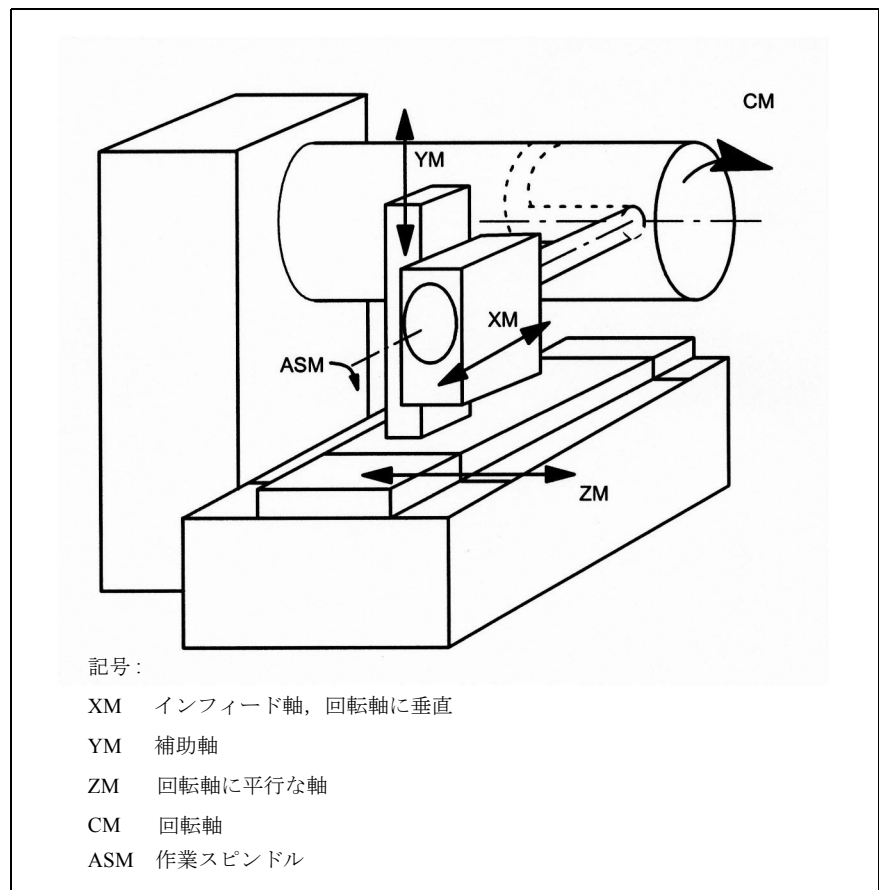


図 2.72 X-Y-Z-C キネマティクスによる円柱面上の溝加工

第3の直線軸が利用可能で(図 2.72)、他の2本の直線軸(軸構成1)と共に右手直交座標系を形成できる場合は、ツール半径補償ファンクションの助けを借りてプログラムされたパスに平行にツールを配置するのに利用できます。この場合、直角にトラバースする部分を有する溝が生成可能となります。

## 機能性

変換動作（両方の軸構成）の間、NC プログラムからの処理と JOG モードの処理の両方について制御装置の全機能が有効です（2.7.2 「■ JOG」 参照）。

## 溝断面

軸構成 1 の場合、回転軸に沿った縦溝は、溝幅がツール半径に正確に一致する場合のみ平行リミットが適用されます。円周に平行な溝（横溝）は始点と終点では平行になりません。

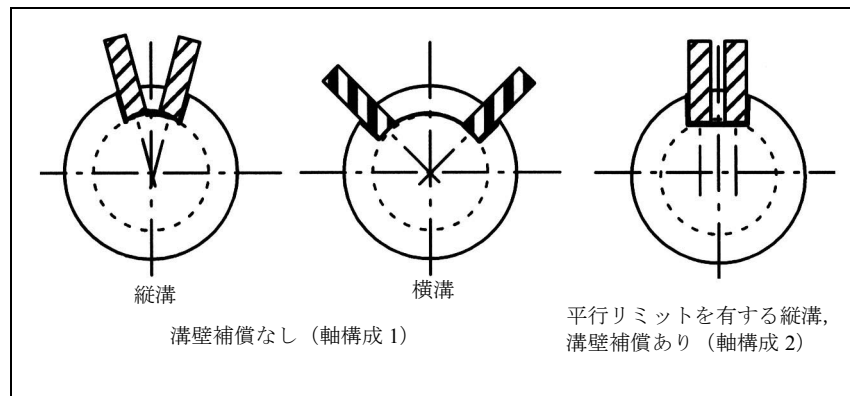


図 2.73 溝壁補償有りおよび無しの溝

## ■ TRACYL の前提条件

### 変換数

このシステムでは最大 8 個の変換データブロックを定義できます。これらの変換のマシンデータ名は \$MC\_TRAFO..<sub>n</sub> で始まり、...<sub>n</sub> で終わりますが、ここで "n" は 1 ～ 8 の数字のいずれかを表します。最初のマシンデータは TRANSMIT に関して説明したのと同じ意味を持っています：

\$MC\_TRAFO\_GEOAX\_ASSIGN\_TAB\_<sub>n</sub>

\$MC\_TRAFO\_TYPE\_<sub>n</sub>

\$MC\_TRAFO\_AXES\_IN\_<sub>n</sub>.

以下に説明する特別な設定値は、生成された円柱面変換 (TRACYL) に関して \$MC\_TRAFO\_TYPE\_<sub>n</sub> および \$MC\_TRAFO\_AXES\_IN\_<sub>n</sub> に適用されます。

### TRACYL 構造の数

変換用に許されている 8 つのデータ構造の内の 2 つを TRACYL ファンクションに割当てることが可能です。それらには \$MC\_TRAFO\_TYPE\_<sub>n</sub> で割当てられた値が 512 か 513 であるという特徴があります。

これらの TRACYL 変換（最大 2）の場合、以下のマシンデータを規定の方法で設定しなければなりません：

\$MC\_TRACYL\_ROT\_AX\_OFFSET\_<sub>t</sub>

\$MC\_TRACYL\_ROT\_SIGN\_IS\_PLUS\_<sub>t</sub>

\$MC\_TRACYL\_BASE\_TOOL\_t

この場合、tには宣言された TRACYL 変換の番号を指定します（最大 2）。

軸構成

図 2.72 に示されているマシンの軸関係とその関係軸データの概要を次に示します。

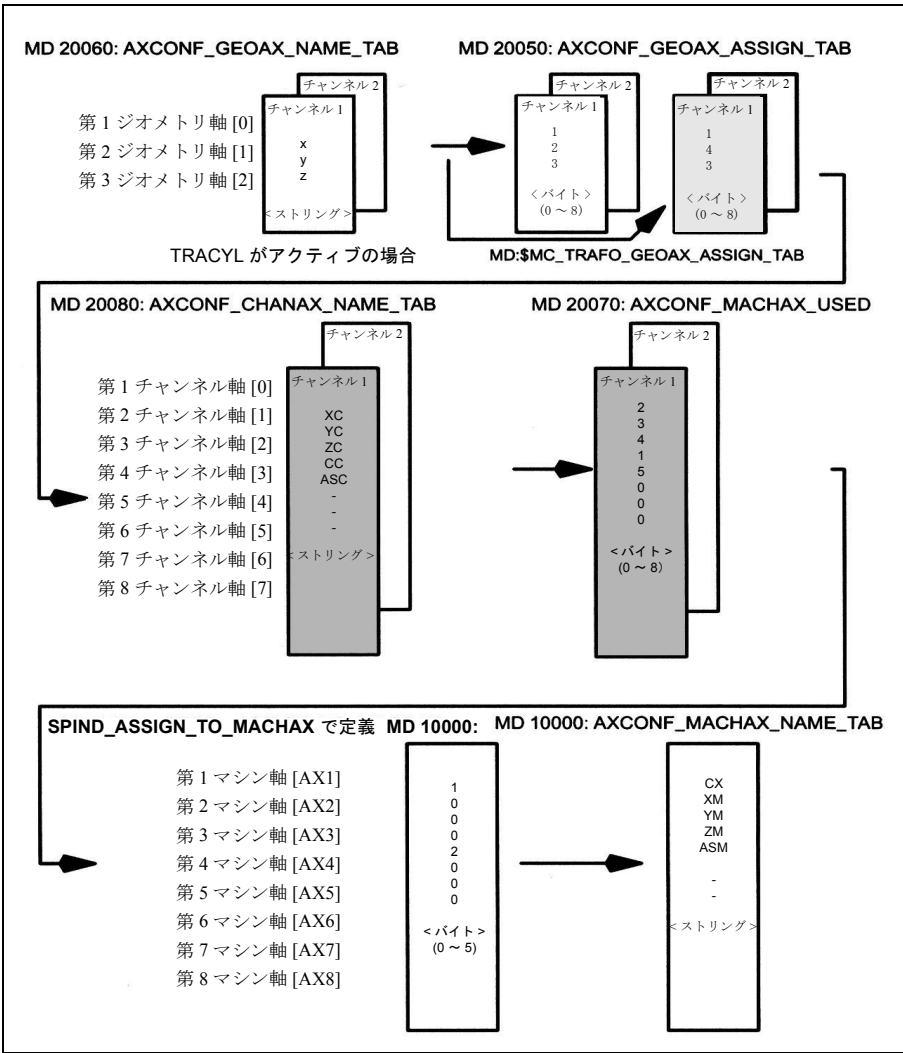


図 2.74 図 2-64 の例についての軸構成

図 2.74 で強調されている配列は TRACYL がアクティブの時に適用されます。

ジオメトリ軸の命名

上記の軸構成概要に従って、TRACYL 動作に関するジオメトリ軸を以下で定義しなければなりません：

```
$MC_AXCONF_GEOAX_NAME_TAB[0]="X"  
" _TAB[1]="Y"  
" _TAB[2]="Z"
```

(図 2.74 に従った名前選択は初期設定にも対応)



## チャンネル軸へのジオメトリ軸の割当て

これらの割当て方法は TRACYL がアクティブであるかないかによって異なります。

### - TRACYL 非アクティブ

Y 軸の通常のトラバース

```
$MC_AXCONF_GEOAX_ASSIGN_TAB[0]=1
" _TAB[1]=2
" _TAB[2]=3
```

### - TRACYL アクティブ

Y 軸は、周囲面が生成される円柱座標系方向の軸です。

```
$MC_TRAFO_GEOAX_ASSIGN_TAB[0]=1
" _TAB[1]=4
" _TAB[2]=3
```

## チャンネル軸の入力

直交座標軸に属さない軸が加えられます。

```
$MC_AXCONF_CHANAX_NAME_TAB[0]="XC"
" _TAB[1]="YC"
" _TAB[2]="ZC"
" _TAB[3]="CC"
" _TAB[4]="ASC"
```

## マシン軸へのチャンネル軸の割当て

チャンネル軸の cd を基準にして、チャンネル軸の割当てられているマシン軸の番号が制御装置に転送されます。

```
$MC_AXCONF_MACHAX_USED[0]=2
" [1]=3
" [2]=4
" [3]=1
" [4]=5
```

(図 2.74 に対応する入力データ)

## スピンドルの識別

ユーザは各マシン軸がスピンドル (値 > 0: スピンドル番号) かパス軸 (値 0) かを定義します。

```
$MA_SPIND_ASSIGN_TO_MACHAX[0]=1
" [1]=0
" [2]=0
" [3]=0
" [4]=2
```

## マシン軸への名前の割当て

マシン軸の cd を基準にして、マシン軸名が制御装置に転送されます。

```
$MN_AXCONF_MACHAX_NAME_TAB[0]="CM"  
" [1]="XM"  
" [2]="YM"  
" [3]="ZM"  
" [4]="ASM"
```

## ■ TRACYL に固有の設定値

### 変換タイプ

以下のパラグラフでは変換タイプがどのように指定されるかを説明します。

### TRAFO\_TYPE\_n

ユーザは変換データブロックに対して変換タイプを指定しなければなりません (最大 n=8)。TRACYL の場合は、値 512 を 軸構成 1 に、値 513 を 軸構成 2 に対して設定しなければなりません。

例 : MD 24100: TRAFO\_TYPE\_1=512

この設定を行い電源オンで起動してから、TRACYL(d,t) をコールしなければなりません。"t" は宣言された TRANSMIT 変換の番号です。

### 軸イメージ

以下のパラグラフでは変換軸イメージがどのように指定されるかを説明します。

### TRAFO\_AXES\_IN\_n

3 つ (または 4 つ) のチャンネル軸番号を変換データブロック n に指定しなければなりません :

\$MC_TRAFO_AXES_IN_1[0]=	回転軸に対してラジアルな軸のチャンネル軸番号
\$MC_TRAFO_AXES_IN_1[1]=	回転軸のチャンネル軸番号
\$MC_TRAFO_AXES_IN_1[2]=	回転軸に平行な軸のチャンネル軸番号
\$MC_TRAFO_AXES_IN_1[3]=	生成円柱面に平行で回転軸に垂直な追加軸のチャンネル軸番号 (軸構成 2 が選択されている場合)

図 2.74 に従った例 :

```
$MC_TRAFO_AXES_IN_1[0]=1  
$MC_TRAFO_AXES_IN_1[1]=4  
$MC_TRAFO_AXES_IN_1[2]=3  
$MC_TRAFO_AXES_IN_1[3]=2
```

この設定を行って電源オンで起動してから TRACYL(d) あるいは TRACYL(d,t) をコールしなければなりません。軸番号は \$MC\_TRAFO\_ASSIGN\_TAB\_n で定義されているチャンネル軸のシーケンスに対応していなければなりません。

## 回転の中心

回転軸に垂直な、生成された円柱面上の軸の回転の中心は以下のように定義しなければなりません：

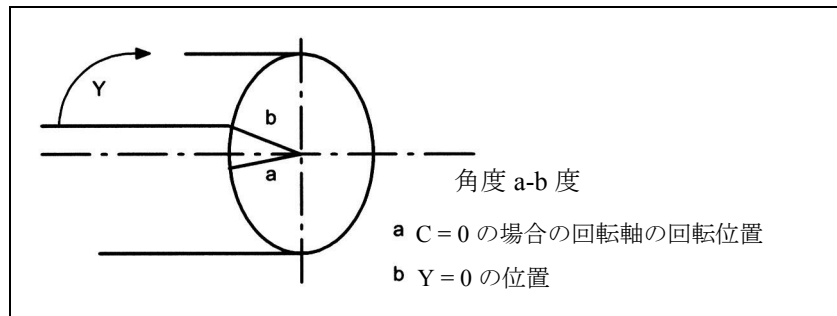


図 2.75 生成円柱面上の軸の回転の中心

## TRACYL\_ROT\_AX\_OFFSET\_t

回転軸の定義済み原点位置を基準として周囲面の回転位置が以下によって指定されます：

\$MC\_TRACYL\_ROT\_AX\_OFFSET\_t=... ;度

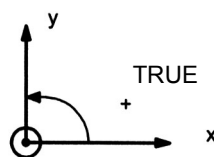
この場合, "t" には変換データブロックで宣言されている TRACYL 変換の数を指定します (tは最大 2)。

## 回転方向

回転軸の回転方向は以下のパラグラフに記述されているマシンデータで指定します。

## TRACYL\_ROT\_SIGN\_IS\_PLUS\_t

z 軸に沿って見て回転軸が x-y 平面上を反時計方向に回転する場合は、マシン軸は TRUE に設定しなければならず、そうでない場合は FALSE に設定しなければなりません。



\$MC\_TRACYL\_ROT\_SIGN\_IS\_PLUS\_t=TRUE

この場合, "t" には変換データブロックで宣言されている TRACYL 変換の数を指定します (tは最大 2)。

## 置き換え可能なジオメトリ軸

ジオメトリ軸が GEOAX() を使用して置き換えられた場合、M コード出力（マシンデータで設定可能）を通してそのことが PLC に通知されます。

- MD 22534: TRAFO\_CHANGE\_M\_CODE

変換切換えの場合に VDI インタフェースに出力される M コードの番号。

（注）

\* このマシンデータが 0 ～ 6, 17 もしくは 30 のうちのいずれかに設定された場合は、M コードは出力されません。

参照： 1.11 軸構成及び座標系 (K2)

## ツール原点の位置

直交座標系の原点を基準とするツール原点の位置は以下のパラグラフに記述されているマシンデータで指定されます。

### TRACYL\_BASE\_TOOL\_t

マシンデータ \$MC\_TRACYL\_BASE\_TOOL\_t は、TRACYL に対して宣言されている円柱座標系の原点を基準としてツール原点の位置を制御装置に知らせるために用いられます。このマシンデータはマシン座標系の 3 つの軸 x, y, z 用に 3 つの構成要素を有しています。

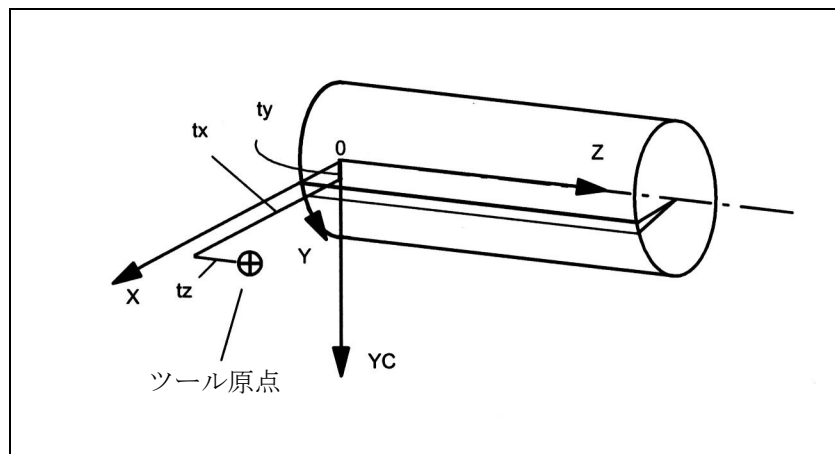


図 2.76 マシンの原点と比較したツール原点の位置（図 2.75 参照）

例：

```
$MC_TRACYL_BASE_TOOL_t[0]=tx  
" [1]=ty  
" [2]=tz
```

この場合, "t" には変換データブロックで宣言されている TRACYL 変換の数を指定します (tは最大 2)。

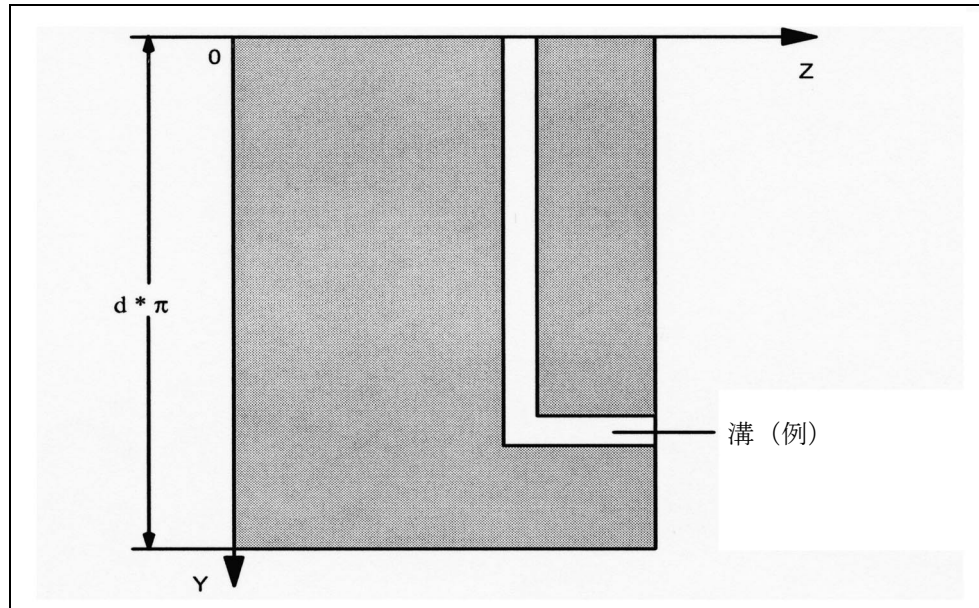


図 2.77 図 2.76 の円柱座標系

## ■ TRACYL の起動

### TRACYL

前述の設定が実行されたなら, TRACYL ファンクションを起動することができます:

TRACYL あるいは

TRACYL(d, t)

1 番目に宣言された TRACYL ファンクションは TRACYL(d) で起動されます。

TRACYL(d, t) は t 番目に宣言された TRACYL ファンクションを起動します。t は 2 以下とします。値 "d" は加工される円柱の現在の直径を意味します。

以下に説明するファンクションの起動と停止の間で, 円柱座標系の軸のトラバース動作をプログラムすることができます。

## ■ TRACYL ファンクションの停止

### TRAFOOF

キーワード TRAFOOF はアクティブ状態の変換動作を停止します。変換動作が停止されると、基本座標系は再びマシン座標系と一致します。

アクティブ状態の TRACYL 変換動作は、他の変換機能（例えば TRANSMIT, TRAANG, TRAORI など）の一つが適切なチャンネル内で起動かされた場合、同様に停止されます。

参照： 3.13 ～ 5 軸刃先方向制御 (F2)

## ■ TRACYL に関する特別なシステム応答

この変換動作はパートプログラムあるいは MDI によって選択あるいは選択解除可能です。

### 選択に関する注意点

- ・ 中間動作ブロックは挿入されません (phases/radii)。
- ・ スプラインブロックシーケンスは終了させなければなりません。
- ・ ツール半径補償は解除しなければなりません。
- ・ TRACYL より前にアクティブ状態にあったフレームは制御装置により解除されます ("Reset programmed frame" G500 に対応)。
- ・ この変換機能が作用を及ぼす軸に関しては、使用中の作業エリア制限は制御装置によって解除されます (プログラムされた WALIMOF に対応)。
- ・ 連続的なパス制御や丸め処理は中断されます。
- ・ DRF オフセットはオペレータが削除しておかなければなりません。
- ・ 溝壁補償付きの円柱面カーブ変換の場合 (軸構成 2, TRAFO\_TYPE\_n=513), 溝がプログラムされた溝中心線の中心で加工されるように補正に使用される軸 (TRAFO\_AXES\_IN\_n[3]) はゼロに設定しなければなりません (y=0)。

### 選択解除に関する注意事項

選択の場合と同様の事項が適用されます。

### アクティブ状態の TRACYL によって課せられる制限事項

起動された TRACYL ファンクションによって課せられる下記の制限事項に注意しなければなりません。

### ツール変更

ツールは、ツール半径補償ファンクションが解除された場合のみ変更可能です。

## フレーム

基本座標系のみに関係する命令はすべて許可されます (FRAME, ツール半径補償)。ただし非アクティブ状態の変換動作に関する手順とは異なり, G91 によるフレーム変更 (相対寸法) は特別扱いされません。トラバースされる相対移動量は, 前のブロックでどのフレームが有効だったかにかかわらず, 新しいフレームのワークピース座標系で評価されます。

## 回転軸

回転軸はジオメトリ軸に占有されているのでプログラムすることができず, したがってチャンネル軸のように直接プログラムすることはできません。

## 軸の利用

回転軸 (y) および追加軸 (YC) に垂直な, 生成された円柱面内の軸は位置決め軸や発振軸として使用してはなりません。

## 例外事項

この変換機能が作用を及ぼす軸は以下の手段や目的のために使用することはできません:

- 事前設定軸として (アラーム)
- 固定点へのアプローチ用に (アラーム)
- 基準化用に (アラーム)

(注)

TRACYL はツール半径の補償や長さ補償が解除される前に解除されなければなりません。

## パートプログラムの中断

TRACYL に関連するパートプログラム処理の中断については下記の事項に注意しなければなりません:

## JOG の後の AUTOMATIC

変換機能がアクティブの時にパートプログラム処理が中断され JOG モードが選択された場合は, AUTOMATIC の再選択に際しては以下のことに注意しなければなりません:

- この変換機能は, 現在位置から中断地点までアプローチブロックでアクティブ状態にあります。衝突に関する監視は全く行われません。



### 警告

オペレータはツールを再配置できるようにする責任があります。

---

## RESET 後 START

パートプログラム処理が RESET で中断されて START で再開した場合は、以下のことに注意しなければなりません：

- 全ての軸がパートプログラムの開始点にある直線ブロック（G0 または G1）によって規定の位置に移動させられた場合のみ、残りのパートプログラムは再生可能状態でトラバースされます。制御装置は、RESET 時点でアクティブだったツールをもう考慮する必要はありません（マシンデータで設定可能）。

### ■ JOG

#### JOG に関する注意事項

溝壁補償を有する、生成された円柱面の変換機能 (\$MC\_TRAFO\_TYPE = 513) が JOG モードでアクティブ状態にある場合は、軸のトラバース動作は AUTOMATIC における前の状態に依存することに注意しなければなりません。溝壁補償ファンクションがアクティブの場合、軸はこの補正ファンクションが選択解除された時の状況とは異なった動作をします。そのためパートプログラムは中断後に再開が可能です (REPOS)。



## ■ TRAANG（傾斜軸）

（注）

以下で説明する TRAANG 変換は、この変換機能がアクティブの時、マシン軸、チャンネル、およびジオメトリ軸に固有の名前が割当てられている必要があります。以下を比較してください：

MD 10000: AXCONF\_MACHAX\_NAME\_TAB,

MD 20080: AXCONF\_CHANAX\_NAME\_TAB,

MD 20060: AXCONF\_GEOAX\_NAME\_TAB.

この方法でのみ明確な割当てを行うことができます。

## タスク定義

グライインディング動作

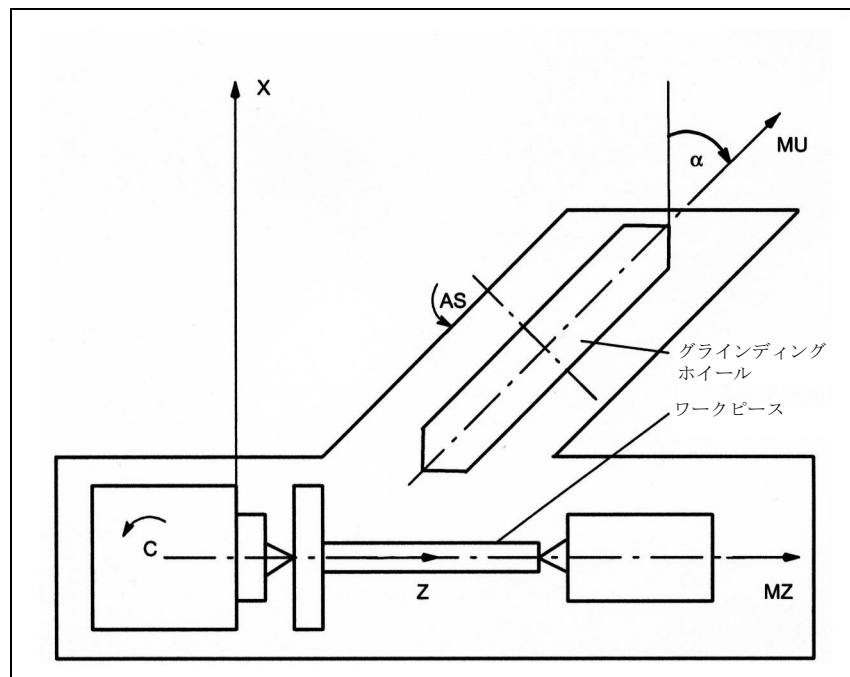


図 2.78 インフィード軸付きマシン

記号の説明：

X, Z	プログラミング用直交座標系
C	回転軸
AS	作業スピンドル
MZ	マシン軸（直線）
MU	傾斜軸

加工動作有効範囲は以下のとおりです：

1. 縦グラインディング
2. 面グラインディング
3. 特殊な輪郭のグラインディング
4. 斜めプランジカットグラインディング

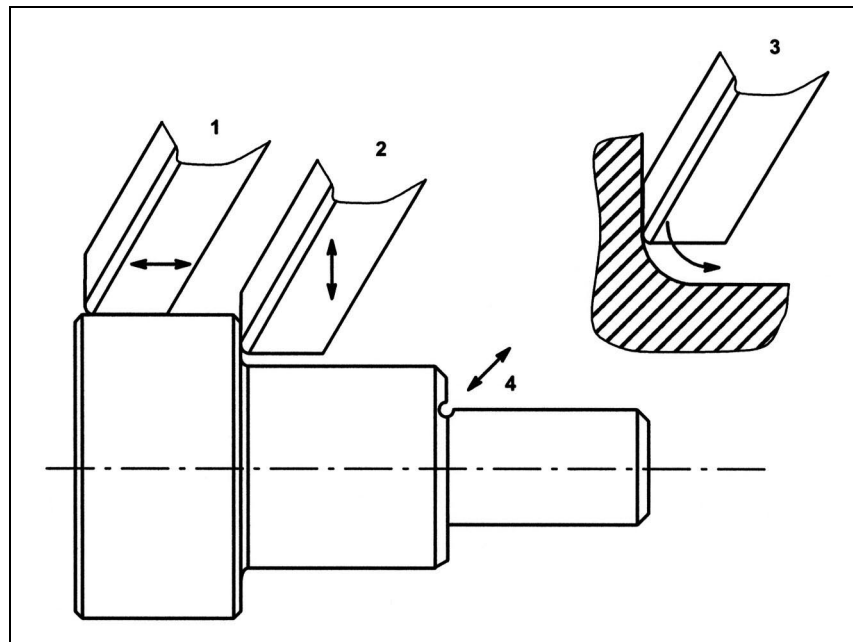


図 2.79 可能なグラインディング動作

## ■ TRAANG（傾斜軸）前提条件

### 軸構成

直交座標系でプログラムできるように、制御装置にこの座標系と実際に存在するマシン軸 (MU, MZ) との関係を知らせる必要があります：

- ジオメトリ軸への名前の割当て
- チャンネル軸へのジオメトリ軸の割当て
  - 一般状態（傾斜軸が非アクティブ）
  - 傾斜軸がアクティブ
- マシン軸番号へのチャンネル軸の割当て
- スピンドルの識別
- マシン軸名の割当て

「\_ 傾斜軸アクティブ」事項を除き、この手順は通常の軸構成用と同じです。

参照： 1.11 軸構成及び座標系 (K2)

---

## 変換数

このシステムでは最大 8 個の変換データブロックを定義できます。これらの変換のマシンデータ名は `$MC_TRAFO_..` で始まり, `..._n` で終わりますが, ここで "n" は 1 ~ 8 の数字のいずれかを表します。以下の項にこれらのデータに関する説明が記載されています:

`$MC_TRAFO_GEOAX_ASSIGN_TAB_n`

`$MC_TRAFO_TYPE_n`

`$MC_TRAFO_AXES_IN_n.`

## 傾斜軸の数

変換用に許されている 8 つデータ構造の内の 2 つをこの傾斜軸ファンクションに割当てることができます。これらは, `$MC_TRAFO_TYPE_n` で割当てられる値が "1024" であるという特徴を有しています。

## 軸構成

図 2.79 に示されているこのグライディングマシンの軸を以下のようにマシンデータに入力しなければなりません:

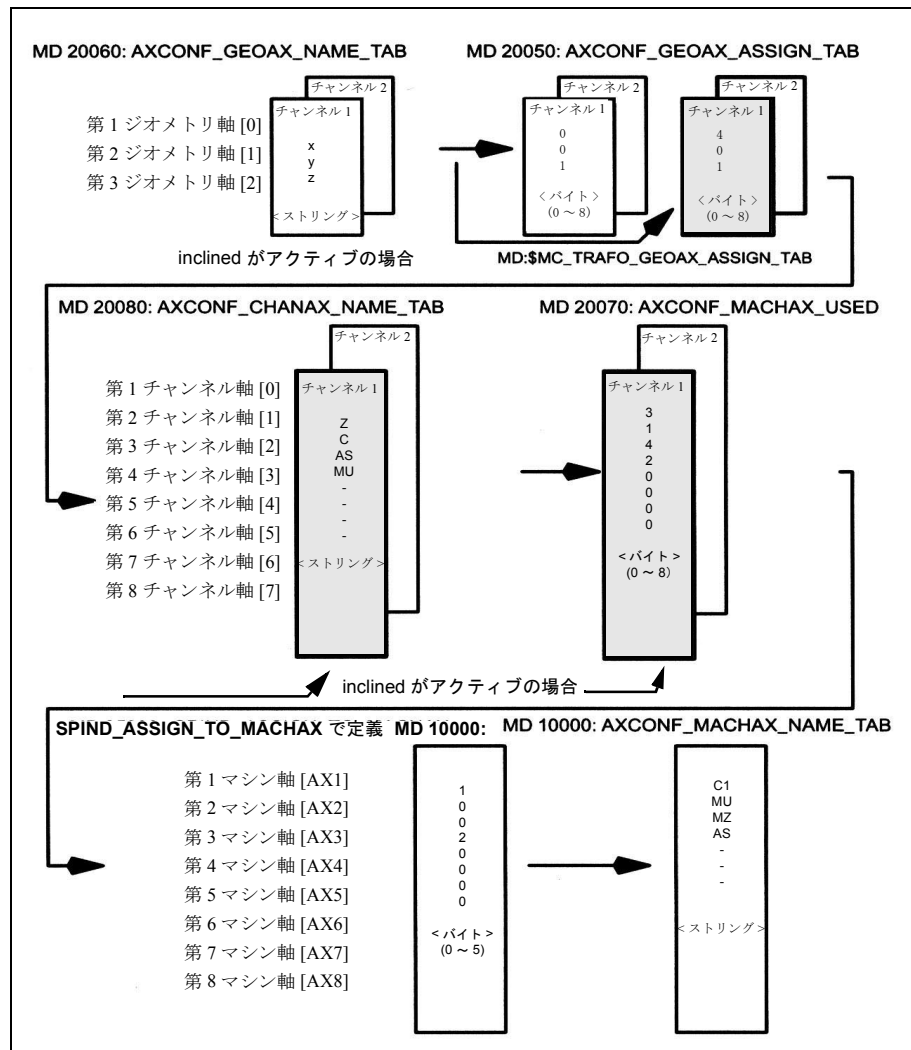


図 2.80 図 2.79 の例の場合の軸構成

図 2.80 で強調表示されている配列は TRAANG がアクティブの時に適用されます。

## ■ TRAANG に固有の設定値

### 変換タイプ

#### TRAFO\_TYPE\_n

ユーザはマシンデータ \$MC\_TRAFO\_TYPE\_n で変換ブロックデータに対して変換タイプを指定しなければなりません（最大 n = 8）。

傾斜軸に関する値は 1024 です：

MD 24100: TRAFO\_TYPE\_1=1024

（注）

マシンデータに対する変更は電源オン後に起動されます。

### 軸イメージ

#### TRAFO\_AXES\_IN\_n

変換データブロック n について 2 つの軸番号を指定しなければなりません：

\$MC\_TRAFO\_AXES\_IN\_1[0]=4      傾斜軸のチャンネル軸番号

\$MC\_TRAFO\_AXES\_IN\_1[1]=1      Z に平行な軸のチャンネル軸番号

\$MC\_TRAFO\_AXES\_IN\_1[2]=0      チャンネル軸番号は非アクティブ

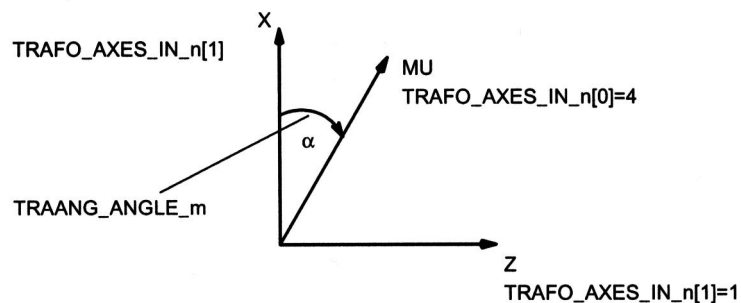


図 2.81 パラメータ TRAANG\_ANGLE\_m

（注）

マシンデータに対する変更は電源オン後に起動されます。

## 傾斜軸の角度

### TRAANG\_ANGLE<sub>m</sub>

マシン軸と傾斜軸の間の角度を制御装置に知らせるためにマシンデータ \$MC\_TRAANG\_ANGLE<sub>m</sub> が使用されます。

\$MC\_TRAANG\_ANGLE<sub>m</sub> = 直交軸とその対応するマシン傾斜軸間の角度  
(単位: 度)。この角度は時計方向にプラス計算  
されます (図 2.78, 角度  $\alpha$  を参照)。

この場合,  $m$  は変換データブロックで宣言されている TRAANG 変換の番号で代用  
されます。 $m$  は最大 2 とします。

## 許容角度範囲

許容角度範囲は:

$-90^\circ < \text{TRAANG\_ANGLE}_m < 0^\circ$

$0^\circ < \text{TRAANG\_ANGLE}_m < 90^\circ$

$0^\circ$  の場合は変換は必要ありません。

$\pm 90^\circ$  の場合は, 傾斜軸は 2 番目の直線軸に平行に配置されます。

## ツール原点の位置

### TRAANG\_BASE\_TOOL<sub>m</sub>

マシンデータ \$MC\_TRAANG\_BASE\_TOOL<sub>m</sub> は, 傾斜軸変換機能に対して宣言さ  
れている座標系の原点を基準としてツール原点の位置を制御装置に知らせるた  
めに用いられます。このマシンデータは直交座標系の 3 つの軸用に 3 つの構成要素を  
有しています。

デフォルト値としてゼロが入力されています。

角度が変更された場合補正值は換算されません。

## 速度制御の最適化

ジョグモード, 位置決めモード, および発振モードで速度制御を最適化するた  
めに下記のマシンデータが用いられます:

### TRAANG\_PARALLEL\_VELO\_RES<sub>m</sub>

マシンデータ \$MC\_TRAANG\_PARALLEL\_VELO\_RES<sub>m</sub> は補償動作用に平行軸  
(\$MC\_TRAFO\_AXES\_IN<sub>n</sub>[1] 参照) に対して作動可能状態に維持される速度リ  
ザーブを設定するのに用いられます。

値範囲: 0 ... 1

0 値 0 が設定されると, 制御装置が自動的にリザーブを決定します:

各軸は同じ優先順位 (デフォルト設定) で制限されます。

(2/M1/2-46)

>0 0 を超える値が設定されると、このリザーブは以下の値に固定されます：

\$MC_TRAANG_PARALLEL _VELO_RES_m	平行軸の許容マシン軸速度値 *
-------------------------------------	--------------------

垂直軸の速度特性はこのリザーブに基づいて制御装置によって決定されます。

## TRAANG\_PARALLEL\_ACCEL\_RES\_m

マシンデータ \$MC\_TRAANG\_PARALLEL\_ACCEL\_RES\_m は補償動作用に平行軸 (\$MC\_TRAFO\_AXES\_IN\_n[1] 参照) 上で作動可能状態に維持される軸加速レートリザーブを設定するのに用いられます。

値範囲 : 0 ... 1

0 値 0 が設定されると、制御装置が自動的にこのリザーブを決定します：  
各軸は同等の優先順位（デフォルト設定）で加速されます。

>0 0 を超える値が設定されると、このリザーブは以下に決められます：

\$MC_TRAANG_PARALLEL _ACCEL_RES_m	平行軸の許容マシン軸速度値 *
--------------------------------------	--------------------

垂直軸の速度特性はこのリザーブに基づいて制御装置によって決定されます。

## 置き換え可能なジオメトリ軸

ジオメトリ軸が GEOAX() を使用して置き換えられた場合、M コード出力（マシンデータで設定可能）を通してそのことが PLC に通知されます。

- MD 22534: TRAFO\_CHANGE\_M\_CODE

変換切換えの場合に VDI インタフェースに出力される M コードの番号。

(注)

このマシンデータが 0 ~ 6, 17 もしくは 30 のうちのいずれかに設定される場合は、M コードは出力されません。

参照： 1.11 「軸構成及び座標系」(K2)

## ■ TRAANG の起動

### TRAANG(a)

前述の設定が実行されたなら、TRAANG ファンクションを起動することができます：

TRAANG(a) あるいは

TRAANG(a, n)

1 番目に宣言された「傾斜軸」変換機能は TRAANG(a) で起動されます。

傾斜軸の角度は "a" で指定することができます。

"a" が省略されるかゼロが入力されると、この変換機能は前の選択に関するパラメータ設定値で起動されます。

初めての選択の場合は、マシンデータに応じた事前設定値が適用されます。

TRAANG ( a , n) は n 番目に宣言された「傾斜軸」変換機能を起動します。

この形式はチャンネル内で複数の変換機能が起動される場合のみ必要です。

n は最大 2 とします

### プログラミング

TRAANG(a,1)==TRAANG(a,0)==TRAANG(a,)==TRAANG(a)

以下に説明するファンクションの起動と停止の間で直交座標系軸のトラバース動作をプログラムしなければなりません。

## ■ TRAANG の停止

### TRAFOOF

キーワード TRAFOOF はアクティブ状態の変換機能を停止します。変換機能が停止されると、基本座標系は再びマシン座標系と一致します。

アクティブ状態の TRAANG 変換動作は、他の変換機能（例えば TRACYL, TRANSMIT, TRAORI など）の一つが適切なチャンネル内で起動された場合、同様に停止されます。

参照： 3.13 ～ 5 軸刃先方向制御 (F2)



## ■ TRAANG に関する特別なシステム応答

この変換機能はパートプログラムあるいは MDI  
によって選択あるいは選択解除可能です。

### 選択および選択解除

- 中間動作ブロックは挿入されません (phases/radii)。
- スプラインブロックシーケンスは終了させなければなりません。
- ツール半径補償は解除しなければなりません。
- 現在のフレームは制御装置により解除されます (プログラム済み G500 に対応)。
- この変換機能が作用を及ぼす軸に関しては、アクティブ状態の作業エリア制限は制御装置によって解除されます (プログラムされた WALIMOF に対応)。
- 起動されているツール長補償は制御装置によってこの変換に組み込まれます。
- 連続的なパス制御や丸め処理は中断されます。
- DRF オフセットはオペレータによって削除されていなければなりません。
- マシンデータ \$MC\_TRAFO\_AXES\_IN\_n で指定された軸はすべてブロック関係基準に基づいて同期化される必要があります (例えば POSA... によるトラバース動作命令無し)。

### 制限事項

#### ツール変更

ツールは、ツール半径補償ファンクションが解除された場合のみ変更可能です。

#### フレーム

基本座標系のみに関係する命令はすべて許可されます (FRAME, ツール半径補償)。ただし非アクティブの変換機能に関する手順と異なり、G91 によるフレーム変更 (相対寸法) は特別扱いされません。トラバースされる相対移動量は、前のブロックでどのフレームが有効だったかにかかわらず、新しいフレームのワークピース座標系で評価されます。

### 例外事項

この変換機能が作用を及ぼす軸は以下の手段や目的のために使用することはできません：

- 事前設定軸として (アラーム)
- 固定点へのアプローチ用に (アラーム)
- 基準化用に (アラーム)

(注)

TRAANG はツール半径の補償や長さ補償が解除される前に解除しておかなければなりません。

---

## 速度制御

TRAANG に関する速度監視ファンクションは前処理段階において標準機能として実行されます。

メインランにおける監視および制限機能は以下の状態において起動されます：

- AUTOMATIC モードにおいて  
ただしこの変換機能に含まれる位置決め軸あるいは発振軸がプログラムされている場合。
- JOG モードへの切換え時点において  
監視ファンクションは、前処理がメインランと再同期化される場合（例えば JOG から AUTOMATIC に切換えられた時点で）、メインランからブロック前処理ルーチンに再び逆転送されます。  
前処理行程における速度監視ファンクションはメインランにおける監視ファンクションよりも上手にマシンの動的制限値を活用します。

## ■ チェーニング変換

### 概要

ソフトウェアバージョン 5 以降では、「傾斜軸」型の別の変換機能と以下に記載するキネマティック変換機能をチェーニングすることが可能です：

- TRANSMIT
- TRACYL
- TRAANG（傾斜軸）

以下に記載されている機能：

参照： 3.13 ～ 5 軸刃先方向制御 (F2)

- 向き（オリエンテーション）変換
- 万能フライスヘッド（「傾斜軸」変換を使用）

### アプリケーション

以下は可能なチェーニング変換の例です：

- 傾斜したグラインディングホイールを使っての円柱 (TRACYL) のサイドラインとしてプログラムされた輪郭のグラインディング（ツールグラインディングなど）。
- 傾斜したグラインディングホイールを使って TRANSMIT で生成された、丸くない輪郭の仕上げカッティング加工

（注）

以下で説明する変換機能は、この変換機能がアクティブの時、マシン軸、チャンネル、およびジオメトリ軸に固有の名前が割当てられている必要があります。以下を比較してください：

MD 10000: AXCONF\_MACHAX\_NAME\_TAB,

MD 20080: AXCONF\_CHANAX\_NAME\_TAB,

MD 20060: AXCONF\_GEOAX\_NAME\_TAB.

この方法でのみ明確な割当てを行うことができます。

---

## 軸構成

チェーニング変換には以下の構成が必要です：

- ジオメトリ軸への名前の割当て
- チャンネル軸への名前の割当て
- チャンネル軸へのジオメトリ軸の割当て
  - 一般的な場合（変換機能は非アクティブ）
- マシン軸番号へのチャンネル軸の割当て
- スピンドル軸，回転軸，モジュロ軸の識別
- マシン軸名の割当て
- 変換機能別設定値（各単一変換および各チェーニング変換に関する）
  - 変換タイプ
  - 変換に関係する軸
  - アクティブ状態の変換機能によるチャンネル軸へのジオメトリ軸の割当て
  - 変換機能に応じて，同様に座標系の回転位置，  
回転方向，  
ツール原点あるいは本来の座標系  
傾斜軸の角度，その他。

## 変換数

各チャンネルに対して最大 4 個（ソフトウェアバージョン 4 以降の場合は 8 個）の変換データブロックを定義することができます。これらの変換機能のマシンデータ名は \$MC\_TRAFO..  
で始まり，...<sub>n</sub> で終わりますがここで "n" は 1 ～ 8 までの数字の一つを表します。

## チェーニング変換の数

チャンネルに対して最大 8 個の変換機能の内 2 個までをチェーニング変換機能として定義することができます。

## 変換機能シーケンス

マシンデータを構成する際は，チェーニング変換に関するデータに先立って（チェーニング変換の一部になる可能性のある）単一変換に関するデータを指定しなければなりません。

## チェーニングシーケンス

チェーニング変換の場合は，2 番目の変換機能は「傾斜軸」(TRAANG) でなければなりません。

## チェーニング方向

BCS は連結される変換機能のうち 1 番目のものに対する入力です；MCS は 2 番目のものに対する出力です。

## 補足条件

個々の変換機能説明に示されている補足条件および特殊なケースはチェーニング変換においても適用されます。

### ■ チェーニング変換機能の起動

#### TRACON

チェーニング変換機能は下記のもによって起動されます：

TRACON(trf, par)

ここで：

trf

チェーニング変換機能の番号：

1 番目の単一チェーニング変換機能の場合は 0 または 1。

ここで何もプログラムされていなければ、それは 0 または 1 を指定するのと同じ意味、すなわち 1 番目の単一変換機能が起動されます。

2 番目のチェーニング変換機能の場合は 2。

(0 ～ 2 以外の値の場合はエラーアラームが出力されます)。

par

パラメータが必要なチェーニング内の変換機能に対するカンマで区切られた 1 つ以上のパラメータ。

例えば傾斜軸の角度など。パラメータが設定されていない場合は、デフォルト設定値か最後に使用されたパラメータ値が適用できます。カンマを使用するのは、デフォルト設定値が前のパラメータに対して有効である場合に、指定されたパラメータの数値を要求されているシーケンスで確実に求めるためです。特に、trf を指定する必要がないとしても、少なくとも 1 個のパラメータの前にカンマが必要です。例えば：

TRACON(, 3.7).

もし別の変換機能が以前に起動されていた場合は、TRACON() によって暗黙の内に無効とされます。

### ■ チェーニング変換機能の選択解除

#### TRAFOOF

チェーニング変換機能は他の変換機能と同様 TRAFOOF で解除されます。

---

## ■ チェーニング変換機能の特殊特性

### ツールデータ

ツールは常にチェーニング内の 1 番目の変換機能に対して割当てられます。次の変換機能はしたがってアクティブ状態のツール長がゼロであるかのようにふるまいます。マシンデータ (`_BASE_TOOL_`) に設定されている基本ツール長だけがチェーニング内の最初の変換機能に対して有効です。

### 例：

単一変換機能およびそれらから形成されるチェーニング変換機能に関する構成例は 2.7.6 「■チェーニング変換」にあります。

---

## 2.7.3 補足条件

### ■ TRANSMIT

#### 可用性

"TRANSMIT" ファンクションはオプションです。

極トラバース動作および極付近における制御装置最適化反応動作はソフトウェアバージョン 4.1 以降で利用可能です。

### ■ TRACYL

#### 可用性

"TRACYL" ファンクションはオプションです。

### ■ TRAANG

#### 可用性

"TRAANG" (傾斜軸) ファンクションはオプションです。

### ■ チェーニング変換 (SW 5)

#### ソフトウェアバージョン 5 以降

2 つの変換機能をチェーニングさせることが可能です。

ただし、どの変換機能でもチェーニングできるというわけではありません。

ソフトウェアバージョン 5 では、次の制限事項が適用されます：

- チェーニング内の最初の変換機能は下記のものでなければなりません：
  - 向き (オリエンテーション) の変換 (3, 4, 5- 軸変換, 万能フライスヘッド)
  - 送信
  - サイドライン変換または
  - 傾斜軸
- 2 番目の変換機能は傾斜軸変換機能でなければなりません。
- チェーニングできるのは 2 個の変換機能だけです。  
(例えば試験動作のために) チェーニングリストに変換機能を 1 つだけ入力することも可能です。

## 2.7.4 データの説明 (MD, SD)

### ■ 変換機能別マシンデータ

22534 MD 番号	TRAFO_CHANGE_M_CODE ジオメトリ軸に関する変換機能切換え用 M コード	
デフォルト値 : 0	最小入力リミット : 0	最大入力リミット : 99999999
変更が有効になるための条件 :	保護レベル : 2/7	単位 : -
データタイプ : DWORD	適用ソフトウェアバージョン : 4.1 以降	
意味 :	ジオメトリ軸に関する変換機能の切換えの場合に VDI インターフェースに出力される M コードの番号。 この MD が 0 ~ 6, 17 あるいは 30 の内のいずれかの値に設定されている場合は M コード無し。この MD 設定値は、他のファンクションと矛盾することがないかを確認するために監視されるということはない。	



24100 MD 番号	TRAFO_TYPE_1 1 番目の変換機能のタイプ	
デフォルト値 : 0	最小入力リミット : 0	最大入力リミット : 2048
変更が有効になるための条件 : 電源オン	保護レベル : 2/4	単位 :
データタイプ : DWORD	適用ソフトウェアバージョン : 2.0 以降	
意味 :	<div>この MD は各チャンネルに対しどの変換機能が 1 番目のものとして利用できるかを指定する。</div> <div><div><div>430</div><div>変換タイプ</div><div>軸シーケンス</div></div></div> <div>5 軸変換の場合の軸シーケンスおよび許容変換機能のそれぞれに関する変換タイプを指定するための識別子</div> <div>変換タイプ :</div> <div><div>0</div><div>変換無し</div></div> <div><div>16</div><div>回転ツール有り 5 軸変換</div></div> <div><div>32</div><div>回転ツール有り 5 軸変換</div></div> <div><div>48</div><div>回転ツールとワークピース有り軸変換</div></div> <div><div></div><div>変換タイプ 16 ~ 48 の軸シーケンス</div></div> <div><div></div><div>0 軸シーケンス AB</div></div> <div><div></div><div>1 軸シーケンス AC</div></div> <div><div></div><div>2 軸シーケンス BY</div></div> <div><div></div><div>3 軸シーケンス BC</div></div> <div><div></div><div>4 軸シーケンス CA</div></div> <div><div></div><div>5 軸シーケンス CB</div></div> <div><div>256</div><div>TRANSMIT 変換</div></div> <div><div>512</div><div>TRACYL 変換</div></div> <div><div>513</div><div>TRACYL 変換, X-Y-Z-C キネマティックス有り</div></div> <div><div>1024</div><div>TRAANG 変換</div></div> <div><div>2048</div><div>中心無し変換</div></div> <div><div>8192</div><div>チェーニング変換</div></div> <div>変換タイプ 16 ~ 48 に数値が割当てられると, 対応軸シーケンスが追加される。 変換タイプ 256 ~ 2048 の軸シーケンスは意味がない (エラーメッセージ無し)。</div>	
この MD は次の場合は無意味	変換無し	
アプリケーション	\$MC_TRAFO_TYP_1=20 ; (16+4)	
関連性	TRAFO_TYPE_2, TRAFO_TYPE_3, ... TRAFO_TYPE_8	
参照	3.1 3 ~ 5 軸刃先方向制御 (F2)	

24110 MD 番号	TRAFO_AXES_IN_1[i] 変換 1 [インデックス]: 0 ... 7 に関する軸割当て	
デフォルト値: 1,2,3,4,5,0,0,0	最小入力リミット: 1	最大入力リミット: 8
変更が有効になるための条件: 電源オン	保護レベル: 2/4 単位:	
データタイプ: バイト	適用ソフトウェアバージョン: 2.0 以降	
意味:	1 番目の変換入力時点で軸割当てインデックス i は TRANSMIT に対して値 0, 1, 2 を想定。TRANSMIT に関する割当ては: \$MC_TRAFO_AXES_IN_1[0]= 回転軸に垂直な軸のチャンネル軸番号 \$MC_TRAFO_AXES_IN_1[1]= 回転軸のチャンネル軸番号 \$MC_TRAFO_AXES_IN_1[2]= 回転軸に平行な軸のチャンネル軸番号 n 番目の位置に入力されたインデックスはこの変換機能によってどの軸が 軸 n に内部的にマップされるかを指定する。	
この MD は次の場合は無意味	変換無し	
アプリケーション	\$MC_TRAFO_AXES_IN_1[0]=1	
関連性	TRAFO_AXES_IN_2, TRAFO_AXES_IN_3, ... TRAFO_AXES_IN_8	
参照	3.1 3 ～ 5 軸刃先方向制御 (F2)	

24120 MD 番号	TRAFO_GEOAX_ASSIGN_TAB_1[i] 変換 1 [ジオメトリ軸番号]: 0 ... 2 についてのチャンネル軸へのジオメトリ軸の割当て	
デフォルト値: 0,0,0	最小入力リミット: 0	最大入力リミット: 8
変更が有効になるための条件: 電源オン	保護レベル: 2/4	単位:
データタイプ: バイト	適用ソフトウェアバージョン: 2.0 以降	
意味:	この MD は、変換 1 がアクティブの時に直交座標系の軸がマップされるチャンネル軸を指定する。インデックス i は TRANSMIT で値 0, 1, 2 を想定する。それは 1 番目, 2 番目, 3 番目のジオメトリ軸に適用される。	
この MD は次の場合は無意味	変換無し	
アプリケーション	\$MC_TRAFO_GEOAX_ASSIGN_TAB_1[0]= チャンネル軸番号	
関連性	\$MC_AXCONF_GEOAX_ASSIGN_TAB, どの変換機能も非アクティブ状態の場合。	
参照	3.1 3 ～ 5 軸刃先方向制御 (F2)	

24200 / 24300 / 24400/24430/24440/ 24450/24460 MD 番号	TRAFO_TYPE_2 / _3 / _4 / _5/ _6/ _7/ _8  変換のタイプ	
デフォルト値 : 0	最小入力リミット : 0	最大入力リミット : 2048
変更が有効になるための条件 : 電源オン	保護レベル : 2/4	単位 :
データタイプ : DWORD	適用ソフトウェアバージョン : 2.0 以降	
意味 :	TRAFO_TYPE_1 に関しては, チャンネル内の 2 番目から 8 番目の変換機能として利用できる変換機能に適用される。	

24210 / 24310 / 24410/24432/24442/ 24452/24462 MD 番号	TRAFO_AXES_IN_2[i] / _3[i] / _4[i] / _5[i] / _6[i] / _7[i] / _8[i]  変換 2/3/4/5/6/7/8 [ 軸インデックス ]: 0 ... 7 に関する軸割当て	
デフォルト値 : 1,2,3,4,5,0,0,0	最小入力リミット : 1	最大入力リミット : 8
変更が有効になるための条件 : 電源オン	保護レベル : 2/4	単位 : -
データタイプ : バイト	適用ソフトウェアバージョン : 2.0 以降	
意味 :	2 番目から 8 番目の変換機能の入力で軸割当て。TRAFO_AXES_IN_1 と同じ意味。	

24220 / 24320 / 24420/24434/24444/ 24454/24464 MD 番号	TRAFO_GEOAX_ASSIGN_TAB_2[i] / _3[i] / _4[i] / _5[i] / _6[i] / _7[i] / _8[i] 変換 2/3/4/5/6/7/8 [ ジオメトリ軸番号 ]: 0 ... 2 に関するチャンネル軸への ジオメトリ軸の割当て	
デフォルト値 : 0,0,0	最小入力リミット : 0	最大入力リミット : 8
変更が有効になるための条件 : 電源オン	保護レベル : 2/4	単位 : -
データタイプ : バイト	適用ソフトウェアバージョン : 2.0	
意味 :	変換機能 2 から 8 がアクティブの時に直交座標系の軸が割当てられるべきチャンネル軸がこの MD に設定される。このファンクションは他の点では TRAFO_GEOAX_ASSIGN_TAB_1 に一致する。	

## ■ ファンクション別マシンデータ

### ■ TRANSMIT

24900 MD 番号	TRANSMIT_ROT_AX_OFFSET_1 回転軸の位置オフセット	
デフォルト値: 0	最小入力リミット: 0	最大入力リミット: 360
変更が有効になるための条件: 電源オン	保護レベル: 2/4	単位: 度
データタイプ: ダブル	適用ソフトウェアバージョン: 2.0 以降	
意味:	TRANSMIT ファンクションがアクティブの時, 各チャンネルに 1 番目に宣言されている TRANSMIT 変換に対して原点を基準にして回転軸のオフセットを度単位で指定する。	
この MD は次の場合は無意味	TRANSMIT 非アクティブ	
アプリケーション	\$MC_TRANSMIT_ROT_AX_OFFSET_1=15.0	
関連性	TRANSMIT_ROT_AX_OFFSET_2	

24910 MD 番号	TRANSMIT_ROT_SIGN_IS_PLUS_1 回転軸の符号 1/2	
デフォルト値: 1	最小入力リミット: 0	最大入力リミット: 1
変更が有効になるための条件: 電源オン	保護レベル: 2/4	単位: -
データタイプ: BOOLEAN	適用ソフトウェアバージョン: 2.0 以降	
意味:	TRANSMIT がアクティブの場合, 各チャンネルに 1 番目に宣言されている TRANSMIT 変換に対し回転軸に適用される符号を指定する。	
この MD は次の場合は無意味	TRANSMIT 変換無し	
アプリケーション	\$MC_TRANSMIT_ROT_SIGN_IS_PLUS_1=TRUE	
関連性	TRANSMIT_ROT_SIGN_IS_PLUS_2	

24911 MD 番号	TRANSMIT_POLE_SIDE_FIX_1 極前後の作業範囲の制限, 1 番目の変換	
デフォルト値: 0	最小入力リミット: 0	最大入力リミット: 2
変更が有効になるための条件: 電源オン	保護レベル: 2/4	単位:
データタイプ: バイト	適用ソフトウェアバージョン: 4.1 以降	
意味:	<p>極前後の作業範囲に関する制限, あるいは制限無し, すなわち極を通るトラバース動作セット点は以下の意味を有する:</p> <p>1: 位置 <math>\geq 0</math> の場合, 直線軸の作業範囲 (直線軸に平行なツール長さ補償が 0 の場合)</p> <p>2: 位置 <math>\leq 0</math> の場合, 直線軸の作業範囲 (直線軸に平行なツール長さ補償が 0 の場合)</p> <p>0: 作業範囲の制限無し。極を通るトラバース動作</p>	

24920 MD 番号	TRANSMIT_BASE_TOOL_1[i] 変換機能の起動時に基本ツールのベクトル	
デフォルト値 : 0	最小入力リミット : 0	最大入力リミット :
変更が有効になるための条件 : 電源オン	保護レベル : 2/4	単位 : mm
データタイプ : DOUBLE	適用ソフトウェアバージョン : 2.0	
意味 :	MD はツール長さ補償機能非選択状態でアクティブ状態にある TRAANG 用の適 当なジオメトリ軸に関するツール原点の距離を指定する。距離は各チャンネ ルの 1 番目の TRANSMIT 変換に対して指定される。プログラムされている長さ 補償は基本ツールに加えられる。インデックス i は 1 番目から 3 番目のジオメト リ軸に対して値 0, 1, 2 を想定する。	
この MD は次の場合は無意味	TRANSMIT 変換無し	
アプリケーション	\$MC_TRANSMIT_BASE_TOOL_1[0]=20.0	
関連性	\$MC_TRANSMIT_BASE_TOOL_2	

24950 MD 番号	TRANSMIT_ROT_AX_OFFSET_2 回転軸の位置オフセット	
デフォルト値 : 0	最小入力リミット : 0	最大入力リミット : 360
変更が有効になるための条件 : 電源オン	保護レベル : 2/4	単位 : 度
データタイプ : DOUBLE	適用ソフトウェアバージョン : 2.0 以降	
意味 :	TRANSMIT ファンクションがアクティブの時、各チャンネルに 2 番目に宣言さ れている TRANSMIT 変換に対して原点を基準にして回転軸のオフセットを度単 位で指定する。	
この MD は次の場合は無意味	TRANSMIT 非アクティブ	
関連性	TRANSMIT_ROT_AX_OFFSET_1	

24960 MD 番号	TRANSMIT_ROT_SIGN_IS_PLUS_2 回転軸の符号 1/2	
デフォルト値 : 1	最小入力リミット : 0	最大入力リミット : 1
変更が有効になるための条件 : 電源オン	保護レベル : 2/4	単位 : -
データタイプ : BOOLEAN	適用ソフトウェアバージョン : 2.0 以降	
意味 :	TRANSMIT がアクティブの場合、各チャンネルに 2 番目に宣言されている TRANSMIT 変換に対し回転軸に適用される符号を指定する。	
この MD は次の場合は無意味	TRANSMIT 変換無し	
アプリケーション	\$MC_TRANSMIT_ROT_SIGN_IS_PLUS_1=TRUE	
関連性	TRANSMIT_ROT_SIGN_IS_PLUS_1	

24961 MD 番号	TRANSMIT_POLE_SIDE_FIX_2 極前後の作業範囲の制限, 2 番目の変換機能	
デフォルト値: 0	最小入力リミット: 0	最大入力リミット: 2
変更が有効になるための条件: 電源オン	保護レベル: 2/4	単位:
データタイプ: BYTE	適用ソフトウェアバージョン: 4.1 以降	
意味:	<p>極前後の作業範囲に関する制限, あるいは制限無し, すなわち極を通るトラバース動作セット点は以下の意味を有する:</p> <p>1: 位置 <math>\geq 0</math> の場合, 直線軸の作業範囲 (直線軸に平行なツール長補償が 0 の場合)</p> <p>2: 位置 <math>\leq 0</math> の場合, 直線軸の作業範囲 (直線軸に平行なツール長補償が 0 の場合)</p> <p>0: 作業範囲の制限無し。極を通るトラバース動作</p>	

24970 MD 番号	TRANSMIT_BASE_TOOL_2[i] 変換機能起動時に基本ツールのベクトル	
デフォルト値: 0	最小入力リミット: 0	最大入力リミット:
変更が有効になるための条件: 電源オン	保護レベル: 2/4	単位: mm
データタイプ: DOUBLE	適用ソフトウェアバージョン: 2.0 以降	
意味:	<p>ツール長さ補償非選択状態でアクティブな TRANSMIT に有効なジオメトリ軸に関するツール原点を各チャンネルの 2 番目の TRANSMIT に対して指定する。プログラムされている長さ補償は基本ツールに加えられる。インデックス i は 1 番目から 3 番目のジオメトリ軸に対して値 0, 1, 2 を想定する。</p>	
この MD は次の場合は無意味	TRANSMIT 変換無し	
アプリケーション	\$MC_TRANSMIT_BASE_TOOL_2[0]=tx	
関連性	\$MC_TRANSMIT_BASE_TOOL_1	

## ■ TRACYL

24800 MD 番号	TRACYL_ROT_AX_OFFSET_1 1 番目の TRACYL 変換機能に関する回転軸のオフセット	
デフォルト値: 0	最小入力リミット: ***	最大入力リミット: ***
変更が有効になるための条件: 電源オン	保護レベル: 2/4	単位: 度
データタイプ: DOUBLE	適用ソフトウェアバージョン: 2.0 以降	
意味:	TRACYL ファンクションがアクティブの時, 各チャンネルに 1 番目に宣言されている TRACYL 変換に対して原点を基準にして回転軸のオフセットを度単位で指定する。	
この MD は次の場合は無意味	TRACYL 非アクティブ	
アプリケーション	\$MC_TRACYL_ROT_AX_OFFSET_1=15.0	
関連性	TRACYL_ROT_AX_OFFSET_2	

24810 MD 番号	TRACYL_ROT_SIGN_IS_PLUS_1 1 番目の TRACYL 変換に関する回転軸の符号	
デフォルト値: 1	最小入力リミット: 0	最大入力リミット: 1
変更が有効になるための条件: 電源オン	保護レベル: 2/4	単位: -
データタイプ: BOOLEAN	適用ソフトウェアバージョン: 2.0 以降	
意味:	TRACYL がアクティブの場合, 各チャンネルに 1 番目に宣言されている TRACYL 変換に対し回転軸に適用される符号を指定する。	
この MD は次の場合は無意味	TRACYL 変換無し	
アプリケーション	\$MC_TRACYL_ROT_SIGN_IS_PLUS_1=TRUE	
関連性	TRACYL_ROT_SIGN_IS_PLUS_2	

24820 MD 番号	TRACYL_BASE_TOOL_1[i] 1 番目の TRACYL 変換に関する基本ツールのベクトル [軸番号]: 0 ... 2	
デフォルト値: 0	最小入力リミット: 0	最大入力リミット:
変更が有効になるための条件:	保護レベル: 2/4	単位: mm
データタイプ: DOUBLE	適用ソフトウェアバージョン:	
意味:	MD はツール長さ補償ファンクション非選択状態でアクティブ状態にある TRACYL 用の適当なジオメトリ軸に関係するツール原点の距離を指定する。距離は各チャンネルの 1 番目の TRACYL 変換に対して指定される。プログラムされている長さ補償は基本ツールに加えられる。インデックス i は 1 番目から 3 番目のジオメトリ軸に対して値 0, 1, 2 を想定する。	
この MD は次の場合は無意味	TRACYL 変換無し	
アプリケーション	\$MC_TRACYL_BASE_TOOL_1[0]=tx	
関連性	\$MC_TRACYL_BASE_TOOL_2	

24850 MD 番号	TRACYL_ROT_AX_OFFSET_2 2 番目の TRACYL 変換に関する回転軸のオフセット	
デフォルト値: 0	最小入力リミット: ***	最大入力リミット: ***
変更が有効になるための条件: 電源オン	保護レベル: 2/4	単位: 度
データタイプ: DOUBLE	適用ソフトウェアバージョン: 2.0 以降	
意味:	TRACYL ファンクションがアクティブの時, 各チャンネルに 2 番目に宣言されている TRACYL 変換に対して原点を基準にして回転軸のオフセットを度単位で指定する。	
この MD は次の場合は無意味	TRACYL 非アクティブ	
アプリケーション	\$MC_TRACYL_ROT_AX_OFFSET_2=15.0	
関連性	TRACYL_ROT_AX_OFFSET_1	

24860 MD 番号	TRACYL_ROT_SIGN_IS_PLUS_2 2 番目の TRACYL 変換に関する回転軸の符号	
デフォルト値: 1	最小入力リミット: 0	最大入力リミット: 1
変更が有効になるための条件: 電源オン	保護レベル: 2/4	単位: -
データタイプ: BOOLEAN	適用ソフトウェアバージョン: 2.0	
意味:	TRACYL がアクティブの場合, 各チャンネルに 2 番目に宣言されている TRACYL 変換に対し回転軸に適用される符号を指定する。	
この MD は次の場合は無意味	TRACYL 変換無し	
アプリケーション	\$MC_TRACYL_ROT_SIGN_IS_PLUS_2=TRUE	
関連性	TRACYL_ROT_SIGN_IS_PLUS_1	

24870 MD 番号	TRACYL_BASE_TOOL_2[i] 2 番目の TRACYL 変換用の基本ツールのベクトル [ 軸番号 ]: 0 ... 2	
デフォルト値: 0	最小入力リミット: 0	最大入力リミット:
変更が有効になるための条件:	保護レベル: 2/4	単位: mm
データタイプ: DOUBLE	適用ソフトウェアバージョン: 2.0 以降	
意味:	ツール長補償非選択状態でアクティブ状態の TRACYL に有効なジオメトリ軸に 関係するツール原点を各チャンネルの 2 番目の TRACYL に対して指定する。 プログラムされている長さ補償は基本ツールに加えられる。インデックス i は 1 番目から 3 番目のジオメトリ軸に対して値 0, 1, 2 を想定する。	
この MD は次の場合は無意味	TRACYL 変換無し	
アプリケーション	\$MC_TRACYL_BASE_TOOL_2[0]=tx	
関連性	\$MC_TRACYL_BASE_TOOL_1	



## ■ TRAANG

24700 MD 番号	TRAANG_ANGLE_1 1 番目の TRAANG 変換に関する直交座標軸と実際の（傾斜）軸の間の角度	
デフォルト値：0	最小入力リミット：-90	最大入力リミット：90
変更が有効になるための条件： 電源オン	保護レベル：2/4	単位：度
データタイプ：DOUBLE	適用ソフトウェアバージョン：2.0 以降	
意味：	1 番目に宣言された TRAANG 変換に対して，TRAANG がアクティブのとき第 1 マシン軸と第 1 基本軸の間の傾斜軸の角度を度単位で指定する。角度は時計方向にプラス計算される。	
この MD は次の場合は無意味	TRAANG 非アクティブ	
アプリケーション	\$MC_TRAANG_ANGLE_1=15.0	
関連性	TRAANG_ANGLE_2	

24710 MD 番号	TRAANG_BASE_TOOL_1[i] 1 番目の TRAANG 変換 [ 軸番号 ]: 0 ... 2 に関する基本ツールのベクトル	
デフォルト値：0	最小入力リミット：0	最大入力リミット：2
変更が有効になるための条件： 電源オン	保護レベル：2/4	単位：
データタイプ：DOUBLE	適用ソフトウェアバージョン：2.0 以降	
意味：	MD はツール長さ補償非選択状態でアクティブにある TRAANG 用の適当なジオメトリ軸に関係するツール原点の距離を指定する。距離は各チャンネルの 1 番目の TRAANG 変換に対して指定される。プログラムされている長さ補償は基本ツールに加えられる。インデックス i は 1 番目から 3 番目のジオメトリ軸に対して値 0, 1, 2 を想定する。	
この MD は次の場合は無意味		
アプリケーション	\$MC_TRAANG_BASE_TOOL_1[0]=tx	
関連性	\$MC_TRAANG_BASE_TOOL_2	

24720 MD 番号	TRAANG_PARALLEL_VELO_RES_1	
デフォルト値：0	最小入力リミット：0	最大入力リミット：1
変更が有効になるための条件： 電源オン	保護レベル：2/4	単位：-
データタイプ：DOUBLE	適用ソフトウェアバージョン：2.0	
意味：	ジョグ動作，位置決め動作，発振動作の場合，平行軸 (\$MC_TRAFO_AXES_IN_n[1] 参照) に対して待機状態に維持される補償動作の速度リザーブを指定する；MD 設定値は各チャンネルの 1 番目の TRAANG 変換に適用される。	
この MD は次の場合は無意味	TRAANG 非アクティブ	
アプリケーション	\$MC_TRAANG_PARALLEL_VELO_RES_1=0	
関連性	TRAANG_PARALLEL_VELO_RES_2	

24721 MD 番号	TRAANG_PARALLEL_VELO_RES_2	
デフォルト値 : 0	最小入力リミット : 0	最大入力リミット : 1
変更が有効になるための条件 : 電源オン	保護レベル : 2/4	単位 :
データタイプ : DOUBLE	適用ソフトウェアバージョン : 2.0 以降	
意味 :	ジョグ動作, 位置決め動作, 発振動作の場合, 平行軸 (\$MC_TRAFO_AXES_IN_n[1] 参照) に対して待機状態に維持される補償動作の速度リザーブを指定する ; MD 設定値は各チャンネルの 2 番目の TRAANG 変換に適用される。	
アプリケーション	\$MC_TRAANG_PARALLEL_VELO_RES_2=0	
関連性	\$MC_TRAANG_PARALLEL_VELO_RES_1	

24750 MD 番号	TRAANG_ANGLE_2 2 番目の TRAANG 変換に関する直交座標軸と実際の (斜) 軸の間の角度	
デフォルト値 : 0	最小入力リミット : -90	最大入力リミット : 90
変更が有効になるための条件 : 電源オン	保護レベル : 2/4	単位 : 度
データタイプ : DOUBLE	適用ソフトウェアバージョン : 2.0 以降	
意味 :	2 番目に宣言された TRAANG 変換に対して, TRAANG がアクティブの場合第 1 マシン軸と第 1 基本軸の間の傾斜軸の角度を度単位で指定する。角度は時計方向にプラス計算される。	
この MD は次の場合は無意味	TRAANG 非アクティブ	
アプリケーション	\$MC_TRAANG_ANGLE_1=15.0	
関連性	TRAANG_ANGLE_1	

24760 MD 番号	TRAANG_BASE_TOOL_2[i] 2 番目の TRAANG 変換 [ 軸番号 ]: 0 ... 2 に関する基本ツールのベクトル	
デフォルト値 : 0	最小入力リミット : 0	最大入力リミット : 2
変更が有効になるための条件 : 電源オン	保護レベル : 2/4	単位 :
データタイプ : DOUBLE	適用ソフトウェアバージョン : 2.0 以降	
意味 :	ツール長さ補償非選択状態でアクティブ状態にある TRAANG 用の適当なジオメトリ軸に関するツール原点の距離を指定する。距離は各チャンネルの 2 番目の TRAANG 変換に対して指定される。 プログラムされている長さ補償は基本ツールに加えられる。インデックス i は 1 番目から 3 番目のジオメトリ軸に対して値 0, 1, 2 を想定する。	
この MD は次の場合は無意味	TRAANG 変換無し	
アプリケーション	\$MC_TRAANG_BASE_TOOL_2[0]=tx	
関連性	\$MC_TRAANG_BASE_TOOL_1	

24770 MD 番号	TRAANG_PARALLEL_ACCEL_RES_1	
デフォルト値: 0	最小入力リミット: 0	最大入力リミット: 1
変更が有効になるための条件: 電源オン	保護レベル: 2/4	単位:
データタイプ: DOUBLE	適用ソフトウェアバージョン: 2.0 以降	
意味:	ジョグ動作, 位置決め動作, 発振動作の場合, 平行軸 (\$MC_TRAFO_AXES_IN_n[1] 参照) に対して待機状態に維持される補償動作の軸加速レートリザーブを指定する; MD 設定値は各チャンネルの 1 番目の TRAANG 変換に適用される。	
この MD は次の場合は無意味	TRAANG 非アクティブ	
アプリケーション	\$MC_TRAANG_PARALLEL_ACCEL_RES_1=0	
関連性	TRAANG_PARALLEL_ACCEL_RES_2	

24771 MD 番号	TRAANG_PARALLEL_ACCEL_RES_2	
デフォルト値: 0	最小入力リミット: 0	最大入力リミット: 1
変更が有効になるための条件: 電源オン	保護レベル: 2/4	単位: -
データタイプ: DOUBLE	適用ソフトウェアバージョン: 2.0 以降	
意味:	ジョグ動作, 位置決め動作, 発振動作の場合, 平行軸 (\$MC_TRAFO_AXES_IN_n[1] 参照) に対して待機状態に維持される補償動作の軸加速レートリザーブを指定する; MD 設定値は各チャンネルの 2 番目の TRAANG 変換に適用される。	
この MD は次の場合は無意味	TRAANG 非アクティブ	
アプリケーション	\$MC_TRAANG_PARALLEL_RES_2=0	
関連性	SMC_TRAANG_PARALLEL_RES_1	

## ■ チェーニング変換用の MD

24995 MD 番号	TRACON_CHAIN_1[n] 1 番目のチェーニング変換の変換チェーニング	
デフォルト値: 0	最小入力リミット: 1	最大入力リミット: 8
変更が有効になるための条件: 電源オン	保護レベル: 2/7	単位: -
データタイプ: DWORD	適用ソフトウェアバージョン: 5 以降	
意味:	<p>この MD は表として内部にセーブされる。表には BCS から MCS まで変換機能が実行されるのと同じシーケンスで連鎖される変換機能の番号が記入される。n は MD の入力項目のインデックスを表す。</p> <p>例:</p> <p>マシンは 5 軸マシンとしてあるいは TRANSMIT マシンとして任意にオペレーションできる。直線軸はその他の直線軸に対し垂直ではない (傾斜軸)。5 つの変換機能をマシンデータによって設定しなければならない, 例えば</p> <p>TRAFO_TYPE_1 = 16      (5 軸変換), 1 番目の変換機能          TRAFO_TYPE_2 = 256    (TRANSMIT), 2 番目の変換機能          TRAFO_TYPE_3 = 1024   (傾斜軸), 3 番目の変換機能          TRAFO_TYPE_4 = 8192   1 番目のチェーニング変換, 4 番目の変換機能          TRAFO_TYPE_5 = 8192   2 番目のチェーニング変換, 5 番目の変換機能</p> <p>4 番目の変換機能を「5 軸変換 / 傾斜軸」チェーニングとし, 5 番目の変換機能を「TRANSMIT / 傾斜軸」チェーニングとする場合は, 1 番目のテーブルに TRACON_CHAIN_1 (1, 3, 0, 0) を, 2 番目のテーブルに TRACON_CHAIN_2 (2, 3, 0, 0) を入力する。詳しい記述法については 2.7.6「例」の例で示されている。</p> <p>入力 0 は変換無しを意味する (3 番目および 4 番目の変換機能はソフトウェアバージョン 5 では連鎖することができない)。</p> <p>変換機能は (TRAFO_TYPE_1 から TRAFO_TYPE_8) どんなシーケンスでも割当てることができる。チェーニング変換が最後になる必要はない。ただし, それらは一つの変換チェーニングの中で行われる全ての変換機能の後ろに存在しなければならない。これは先の例でいうなら, 例えば 3 番目と 4 番目の変換機能の位置は交換してはならないということを意味する。それでも, チェーニング変換に含まないのであれば, 6 番目の変換機能を定義することが可能。</p> <p>ただし, どんな変換機能も連結できるというわけではない。</p> <p>ソフトウェアバージョン 5 では, 以下の制限事項が適用される:</p> <ul style="list-style-type: none"> <li>• チェーニング内で 1 番目の変換機能でなければならないのは:             <ul style="list-style-type: none"> <li>- 向きの変換 (3, 4, 5 軸変換, 万能フライスヘッド),</li> <li>- TRANSMIT または,</li> <li>- サイドライン情報 または,</li> <li>- 傾斜軸 2 番目の変換機能は傾斜軸変換機能でなければならない。</li> </ul> </li> <li>• 連鎖できるのは変換機能 2 つだけである。</li> <li>• (例えばテストのために) 1 つだけ変換機能をこのリストに入力することも可能である。</li> </ul>	
この MD は次の場合は無意味		
アプリケーション	2.7.6「例」	
例外, エラー	チェーニング内に 3 つ以上の変換機能あり, 2 番目の変換機能が TRAANG でない	
関連性	MD 24100: TRAFO_TYPE	
参照	3.1 3 ~ 5 軸刃先方向制御 (F2)	

24996 MD 番号	TRACON_CHAIN_2[n] 2 番目のチェーニング変換の変換チェーニング	
デフォルト値:	最小入力リミット:1	最大入力リミット:8
変更が有効になるための条件: 電源オン	保護レベル:2/7	単位:-
データタイプ:DWORD	適用ソフトウェアバージョン:5 以降	
意味:	TRACON_CHAIN_1 と類似しているが、チャンネル内の 2 番目のチェーニング変換用	
この MD は次の場合は無意味	TRAFOOF	
アプリケーション	2.7.6「例」	
例外, エラー	チェーニング内に 3 つ以上の変換機能があり、2 番目の変換機能が TRAANG でない	
関連性	MD 24100: TRAFO_TYPE	
参照	3.1 3 ～ 5 軸刃先方向制御 (F2)	

## 2.7.5 信号の説明

### ■ TRANSMIT

DB21, ... DBX 33.6 データブロック	変換機能がアクティブ NCK チャンネルからの信号 (NCK->PLC)	
エッジ評価:無し	信号更新:周期的	信号有効ソフトウェアバージョン: 1.2 以降
信号ステータス 1 または 信号 変化 0 ---> 1	NC コマンドの TRANSMIT, TRACYL, TRAANG あるいは TRAORI がパートプログラムにプログラムされている。関係ブロックが NC によって処理済みで現在変換機能がアクティブ状態にある。	
信号ステータス 0 または 信号 変化 1 ---> 0	変換機能非アクティブ。	
参照	YS840DI ユーザーズマニュアル・プログラミング編 3.1 3 ～ 5 軸刃先方向制御 (F2)	

### ■ TRACYL

2.7.5 「■ TRANSMIT」を参照

### ■ TRAANG

2.7.5 「■ TRANSMIT」を参照

## 2.7.6 例

### ■ TRANSMIT

TRANSMIT の起動点までの軸構成用メインステップのプログラム例を、図 2.78 の構成に関して示します。

; 回転動作のための一般的な軸構成

```
$MC_AXCONF_GEOAX_NAME_TAB[0] = "X"      ; ジオメトリ軸
$MC_AXCONF_GEOAX_NAME_TAB[1] = "Y"      ; ジオメトリ軸
$MC_AXCONF_GEOAX_NAME_TAB[2] = "Z"      ; ジオメトリ軸
$MC_AXCONF_GEOAX_ASSIGN_TAB[0] = 1      ; チャンネル軸 1 として X
$MC_AXCONF_GEOAX_ASSIGN_TAB[1] = 0      ; Y は チャンネル軸ではない
$MC_AXCONF_GEOAX_ASSIGN_TAB[2] = 2      ; チャンネル軸 2 として Z
$MC_AXCONF_CHANAX_NAME_TAB[0] = "XC";
$MC_AXCONF_CHANAX_NAME_TAB[1] = "ZC";
$MC_AXCONF_CHANAX_NAME_TAB[2] = "CC";
$MC_AXCONF_CHANAX_NAME_TAB[3] = "ASC";
$MC_AXCONF_CHANAX_NAME_TAB[4] = "";
$MC_AXCONF_MACHAX_USED[0] = 2            ; マシン軸 2 として XC
$MC_AXCONF_MACHAX_USED[1] = 3            ; マシン軸 3 として ZC
$MC_AXCONF_MACHAX_USED[2] = 1            ; マシン軸 1 として CC
$MC_AXCONF_MACHAX_USED[3] = 4            ; マシン軸 4 として ASC
$MC_AXCONF_MACHAX_USED[4] = 0            ; 空
$MA_SPIND_ASSIGN_TO_MACHAX[AX1] = 1      ; C はスピンドル 1
$MA_SPIND_ASSIGN_TO_MACHAX[AX2] = 0      ; X はスピンドルではない
$MA_SPIND_ASSIGN_TO_MACHAX[AX3] = 0      ; Z はスピンドルではない
$MA_SPIND_ASSIGN_TO_MACHAX[AX4] = 2      ; AS はスピンドル 2
$MN_AXCONF_MACHAX_NAME_TAB[0] = "CM"     ; 第 1 マシン軸
$MN_AXCONF_MACHAX_NAME_TAB[1] = "XM"     ; 第 2 マシン軸
$MN_AXCONF_MACHAX_NAME_TAB[2] = "ZM"     ; 第 3 マシン軸
$MN_AXCONF_MACHAX_NAME_TAB[3] = "ASM"    ; 第 4 マシン軸
```

; TRANSMIT の準備 (1 番目で唯一の変換機能として)

\$MA\_ROT\_IS\_MODULO[3] = TRUE ; モジュロ軸として c

\$MC\_TRAFO\_TYPE\_1 = 256 ; TRANSMIT 変換

\$MC\_TRAFO\_AXES\_IN\_1[0] = 1 ; 回転軸に垂直なチャンネル軸

\$MC\_TRAFO\_AXES\_IN\_1[1] = 3 ; チャンネル軸は回転軸

\$MC\_TRAFO\_AXES\_IN\_1[2] = 2 ; 回転軸に平行なチャンネル軸

\$MC\_TRAFO\_GEOAX\_ASSIGN\_TAB\_1 [0] = 1 ; X は第 1 チャンネル軸

\$MC\_TRAFO\_GEOAX\_ASSIGN\_TAB\_1 [1] = 3 ; Z は第 2 チャンネル軸

\$MC\_TRAFO\_GEOAX\_ASSIGN\_TAB\_1 [2] = 2 ; Y は第 3 チャンネル軸

\$MC\_TRANSMIT\_ROT\_AX\_OFFSET\_1 = 0. ; 回転軸の原点を基準とする

X-Y 面の回転位置

\$MC\_TRANSMIT\_ROT\_SIGN\_IS\_PLUS\_1 = FALSE ; 回転軸はマイナス回転

\$MC\_TRANSMIT\_BASE\_TOOL\_1 [0] = 0.0 ; X に関するツール距離

\$MC\_TRANSMIT\_BASE\_TOOL\_1 [1] = 0.0 ; Y に関するツール距離

\$MC\_TRANSMIT\_BASE\_TOOL\_1 [2] = 0.0 ; Z に関するツール距離

; TRANSMIT の起動

; X, Y, Z に関するプログラミング

; 回転動作への復帰

TRAFOOF

## ■ TRACYL

TRACYL の起動点までの軸構成用メインステップのプログラム例を、図 2.72 の構成に関して示します。

; 回転動作のための一般的な軸構成

\$MC\_AXCONF\_GEOAX\_NAME\_TAB[0] = "X" ; ジオメトリ軸

\$MC\_AXCONF\_GEOAX\_NAME\_TAB[1] = "Y" ; ジオメトリ軸

\$MC\_AXCONF\_GEOAX\_NAME\_TAB[2] = "Z" ; ジオメトリ軸

\$MC\_AXCONF\_GEOAX\_ASSIGN\_TAB[0] = 1 ; チャンネル軸 1 として X

\$MC\_AXCONF\_GEOAX\_ASSIGN\_TAB[1] = 2 ; Y はチャンネル軸ではない

\$MC\_AXCONF\_GEOAX\_ASSIGN\_TAB[2] = 3 ; チャンネル軸 2 として Z

\$MC\_AXCONF\_CHANAX\_NAME\_TAB[0] = "XC";

\$MC\_AXCONF\_CHANAX\_NAME\_TAB[1] = "YC";

\$MC\_AXCONF\_CHANAX\_NAME\_TAB[2] = "ZC";

\$MC\_AXCONF\_CHANAX\_NAME\_TAB[3] = "CC";

\$MC\_AXCONF\_CHANAX\_NAME\_TAB[4] = "ASC";

\$MC\_AXCONF\_MACHAX\_USED[0] = 2 ; マシン軸 2 として X

\$MC\_AXCONF\_MACHAX\_USED[1] = 3 ; マシン軸 3 として Y

---

\$MC_AXCONF_MACHAX_USED[2] = 4	; マシン軸 4 として Z
\$MC_AXCONF_MACHAX_USED[3] = 1	; マシン軸 1 として C
\$MC_AXCONF_MACHAX_USED[4] = 5	; マシン軸 5 として AS
\$MA_SPIND_ASSIGN_TO_MACHAX[AX1]= 1	; C はスピンドル 1
\$MA_SPIND_ASSIGN_TO_MACHAX[AX2]= 0	; X はスピンドルではない
\$MA_SPIND_ASSIGN_TO_MACHAX[AX3]= 0	; Y はスピンドルではない
\$MA_SPIND_ASSIGN_TO_MACHAX[AX4]= 0	; Z はスピンドルではない
\$MA_SPIND_ASSIGN_TO_MACHAX[AX5]= 2	; AS はスピンドル 2
\$MN_AXCONF_MACHAX_NAME_TAB[0]= "CM"	; 第 1 マシン軸
\$MN_AXCONF_MACHAX_NAME_TAB[1]= "XM"	; 第 2 マシン軸
\$MN_AXCONF_MACHAX_NAME_TAB[2]= "YM"	; 第 3 マシン軸
\$MN_AXCONF_MACHAX_NAME_TAB[3]= "ZM"	; 第 4 マシン軸
\$MN_AXCONF_MACHAX_NAME_TAB[4]= "ASM"	; 第 5 マシン軸
; TRACYL の準備 (1 番目で唯一の変換機能)	
\$MC_TRAFO_TYPE_1 = 513	; TRACYL 変換, 溝壁補償有り
\$MC_TRAFO_AXES_IN_1[0] = 1	; 回転軸に対して半径方向の チャンネル軸
\$MC_TRAFO_AXES_IN_1[1] = 4	; 回転軸に垂直な円柱生成面 のチャンネル軸
\$MC_TRAFO_AXES_IN_1[2] = 3	; 回転軸に平行なチャンネル軸
\$MC_TRAFO_AXES_IN_1[3] = 2	; チャンネル軸 はインデックス [0] 用の追加軸
\$MC_TRAFO_GEOAX_ASSIGN_TAB_1 [0] = 1	; X は第 1 チャンネル軸
\$MC_TRAFO_GEOAX_ASSIGN_TAB_1 [1] = 4	; Y は第 2 チャンネル軸
\$MC_TRAFO_GEOAX_ASSIGN_TAB_1 [2] = 3	; Z は第 3 チャンネル軸
\$MC_TRACYL_ROT_AX_OFFSET_1 = 0.	; 回転軸の原点を基準とする X-Y 平面の回転位置
\$MC_TRACYL_ROT_SIGN_IS_PLUS_1 = FALSE	; 回転軸はマイナス方向に回転
\$MC_TRACYL_BASE_TOOL_1 [0] = 0.0	; X に関するツール距離
\$MC_TRACYL_BASE_TOOL_1 [1] = 0.0	; Y に関するツール距離
\$MC_TRACYL_BASE_TOOL_1 [2] = 0.0	; Z に関するツール距離
; TRACYL(40.0) の起動	
; Y および Z に関するプログラミングは下記参照	
; 回転動 TRAFOOF に復帰	



## 溝壁補償に関するプログラミング

(TRAFO\_TYPE\_n=513)

### 輪郭

プログラムされている参照輪郭およびその参照輪郭からの溝側壁の距離を基準にして補償方向 (G41, G42) をプログラムするためのアドレス OFFN を用いることによってツールより広い溝を形成することが可能です (図 2.82 参照)。

### ツール半径

溝側壁に関してツール半径は自動的に考慮されます (図 2.82)。平面ツール半径補償機能の機能が十分に発揮されます (外側および内側コーナでの安定した推移とボトルネックとなっている問題の解決)。

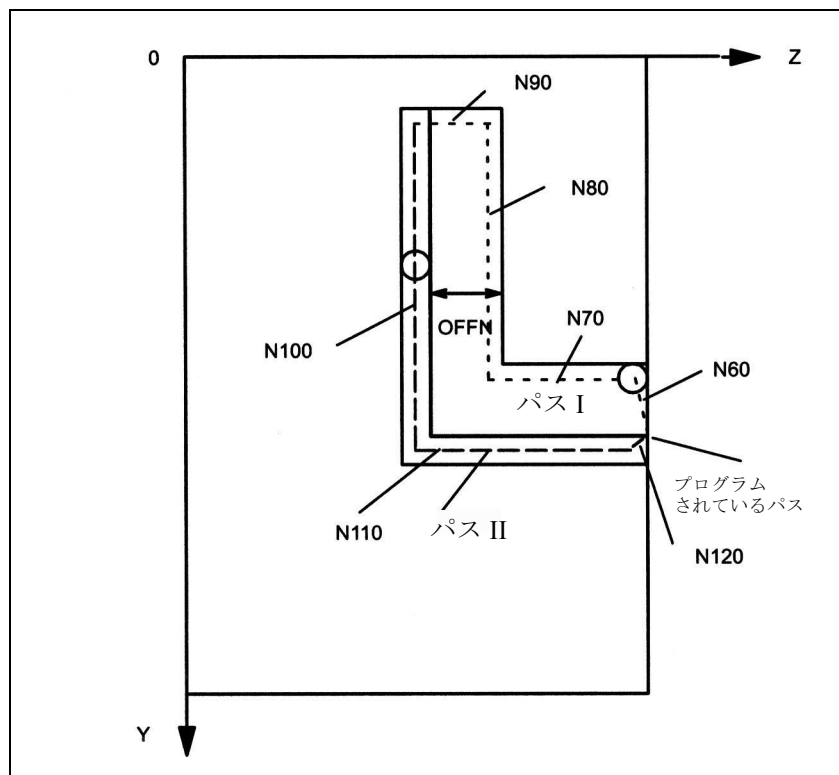


図 2.82 溝 (壁補償有り), 円柱座標 (概略図)

; 変換機能選択後, ツールをパス I づたいからパス II を横切って初期の位置へと誘導するプログラム例 (マシンデータについては 2.7.4 「データの説明 (MD, SD)」, 例 X-Y-Z-C キネマティックス参照) :

---

N1 SPOS=0;	スピンドルを回転軸モードに変更
N5 G0 X25 Y0 Z105 CC=200 F5000 G64;	
	マシンを溝中心上に配置
N10 TRACYL(40.);	変換機能選択, 基準直径 40 mm
N20 G19 G90;	工面は生成円柱面
N30 T1 D1;	ツール選択, 同様に TRACYL (..) の前に配置可能
N40 G1 X20;	溝ベースにツールインフィード
N50 OFFN=12. ;	溝壁距離の定義, 独立した行にあってはならない
; 溝壁に接近	
N60 G1 Z100 G42;	TRC 選択して溝壁に接近
; 溝部パス I の加工	
N70 G1 Z50;	円柱平面に平行な溝部
N80 G1 Y10;	円柱外周に平行な溝部
	; パス II の溝壁に接近
N90 OFFN=4 G42	; 溝壁距離の定義および TRC 選択して溝壁に接近
; 溝部パス II 加工	
N100 G1 Y70;	CC=200 度に一致
N110 G1 Z100;	初期値に戻る
; 溝壁から後退	
N120 G1 Z105 G40;	溝壁から後退するために TRC 解除
N130 G0 X25;	溝壁から後退
N140 TRAFOOF;	
N150 G0 X25 Y0 Z105 CC=200 D0;	初期位置に復帰し工具補正解除
N160 M30	

## ■ TRAANG（傾斜軸）

TRAANG の起動点までの軸構成用メインステップのプログラム例を、図 2.785 の構成に関して示します。

; 研削動作ための一般的な軸構成

```
$MC_AXCONF_GEOAX_NAME_TAB[0] = "X"           ; ジオメトリ軸
$MC_AXCONF_GEOAX_NAME_TAB[1] = ""            ; ジオメトリ軸
$MC_AXCONF_GEOAX_NAME_TAB[2] = "Z"           ; ジオメトリ軸
$MC_AXCONF_GEOAX_ASSIGN_TAB[0] = 0           ; X はチャンネル軸ではない
$MC_AXCONF_GEOAX_ASSIGN_TAB[1] = 0           ; Y はチャンネル軸ではない
$MC_AXCONF_GEOAX_ASSIGN_TAB[2] = 1           ; チャンネル軸 1 として Z
$MC_AXCONF_CHANAX_NAME_TAB[0] = "Z";
$MC_AXCONF_CHANAX_NAME_TAB[1] = "C";
$MC_AXCONF_CHANAX_NAME_TAB[2] = "AS";
$MC_AXCONF_CHANAX_NAME_TAB[3] = "MU"

$MC_AXCONF_MACHAX_USED[0] = 3                 ; マシン軸 3 として Z
$MC_AXCONF_MACHAX_USED[1] = 1                 ; マシン軸 1 として C
$MC_AXCONF_MACHAX_USED[2] = 4                 ; マシン軸 4 として AS
$MC_AXCONF_MACHAX_USED[3] = 2                 ; マシン軸 2 として MU
$MC_AXCONF_MACHAX_USED[3] = 0                 ; 空
$MC_AXCONF_MACHAX_USED[3] = 0                 ; 空
$MC_SPIND_ASSIGN_TO_MACHAX[AX1] = 1           ; C はスピンドル 1
$MC_SPIND_ASSIGN_TO_MACHAX[AX2] = 0           ; X はスピンドルではない
$MC_SPIND_ASSIGN_TO_MACHAX[AX3] = 0           ; Z はスピンドルではない
$MC_SPIND_ASSIGN_TO_MACHAX[AX4] = 2           ; AS はスピンドル 2
$MC_AXCONF_MACHAX_NAME_TAB[0] = "C1"          ; 第 1 マシン軸
$MC_AXCONF_MACHAX_NAME_TAB[1] = "MU"         ; 第 2 マシン軸
$MC_AXCONF_MACHAX_NAME_TAB[2] = "MZ"         ; 第 3 マシン軸
$MC_AXCONF_MACHAX_NAME_TAB[3] = "AS"         ; 第 4 マシン軸

; TRAANG の準備（1 番目で唯一の変換機能）
$MC_TRAFO_TYPE_1 = 1024                      ; TRAANG 変換機能
$MC_TRAFO_AXES_IN_1[0] = 4                  ; チャンネル軸 は「傾斜軸」
$MC_TRAFO_AXES_IN_1[1] = 1                  ; 軸 Z に平行なチャンネル軸
$MC_TRAFO_AXES_IN_1[2] = 0                  ; チャンネル軸 は非アクティブ
$MC_TRAFO_GEOAX_ASSIGN_TAB_1 [0] = 4        ; X は第 1 チャンネル軸
$MC_TRAFO_GEOAX_ASSIGN_TAB_1 [1] = 0        ; Y は第 2 チャンネル軸
$MC_TRAFO_GEOAX_ASSIGN_TAB_1 [2] = 1        ; Z は第 3 チャンネル軸
$MC_TRAANG_ANGLE_1 = 30.                    ; 傾斜軸の角度
```

```

$MC_TRAANG_BASE_TOOL_1 [0] = 0          ; X に関するツール距離
$MC_TRAANG_BASE_TOOL_1 [1] = 0          ; Y に関するツール距離
$MC_TRAANG_BASE_TOOL_1 [2] = 0          ; Z に関するツール距離
TRAANG          ; 起動
                ; X, Y, Z に関するプログラミング
TRAFOOF         ; 回転動作モードに復帰

```

## ■ チェーニング変換

### 例

下記の要素を次の章で決定します：

- 一般的なチャンネル構成
- 単一変換機能
- 先に定義された単一変換機能からなるチェーニング変換
- 単一変換機能の起動
- チェーニング変換機能の起動

このプログラム例には以下の変換機能が含まれます：

1. 回転ツールおよび軸シーケンス AB を有する 5 軸変換機能（変換タイプ 16）
2. TRANSMIT（変換タイプ 256）
3. 傾斜軸（変換タイプ 1024）
4. 1 番目と 3 番目の変換機能のチェーニング（変換タイプ 8192）
5. 2 番目と 3 番目の変換機能のチェーニング（変換タイプ 8192）

### 一般的なチャンネル構成

CHANDATA (1) ; チャンネル 1 のチャンネルデータ

```

$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] = 7
$MC_AXCONF_MACHAX_USED[7] = 0

$MC_AXCONF_CHANAX_NAME_TAB[3]="A"
$MC_AXCONF_CHANAX_NAME_TAB[4]="B"
$MC_AXCONF_CHANAX_NAME_TAB[5]="C"

$MA_IS_ROT_AX[ AX4 ] = TRUE
$MA_IS_ROT_AX[ AX5 ] = TRUE
$MA_IS_ROT_AX[ AX6 ] = TRUE

```

---

```
$MA_IS_ROT_AX[ AX7 ] = TRUE
$MA_SPIND_ASSIGN_TO_MACHAX[ AX5 ] = 0
$MA_SPIND_ASSIGN_TO_MACHAX[AX7] = 1
$MA_ROT_IS_MODULO[AX7] = TRUE
```

### 単一変換

```
; 1 番目 TRAORI
```

```
$MC_TRAFO_TYPE_1= 16 ; TRAORI: A-B キネマティックス
$MC_TRAFO_AXES_IN_1[0]=1
$MC_TRAFO_AXES_IN_1[1]=2
$MC_TRAFO_AXES_IN_1[2]=3
$MC_TRAFO_AXES_IN_1[3]=4
$MC_TRAFO_AXES_IN_1[4]=5
$MC_TRAFO_AXES_IN_1[5]=0
$MC_TRAFO_GEOAX_ASSIGN_TAB_1[0]=1
$MC_TRAFO_GEOAX_ASSIGN_TAB_1[1]=2
$MC_TRAFO_GEOAX_ASSIGN_TAB_1[2]=3
$MC_TRAFO5_BASE_TOOL_1[0]=0
$MC_TRAFO5_BASE_TOOL_1[1]=0
$MC_TRAFO5_BASE_TOOL_1[2]=0
```

```
; 2 番目 TRANSMIT
```

```
$MC_TRAFO_TYPE_2 = 256 ;TRANSMIT
$MC_TRAFO_AXES_IN_2[0]=1
$MC_TRAFO_AXES_IN_2[1]=6
$MC_TRAFO_AXES_IN_2[2]=3
$MC_TRAFO_AXES_IN_2[3]=0
$MC_TRAFO_AXES_IN_2[4]=0
$MC_TRAFO_AXES_IN_2[5]=0
$MC_TRAFO_AXES_IN_2[6]=0
$MC_TRAFO_GEOAX_ASSIGN_TAB_2[0]=1
$MC_TRAFO_GEOAX_ASSIGN_TAB_2[1]=6
$MC_TRAFO_GEOAX_ASSIGN_TAB_2[2]=3
```

```
; 3 番目 TRAANG
```

```
$MC_TRAFO_TYPE_3 = 1024 ;TRAANG
$MC_TRAFO_AXES_IN_3[0] = 1
$MC_TRAFO_AXES_IN_3[1] = 3
$MC_TRAFO_AXES_IN_3[2] = 2
$MC_TRAFO_AXES_IN_3[3] = 0
$MC_TRAFO_AXES_IN_3[4] = 0
```

---

\$MC\_TRAFO\_GEOAX\_ASSIGN\_TAB\_3[0] = 1  
\$MC\_TRAFO\_GEOAX\_ASSIGN\_TAB\_3[1] = 3  
\$MC\_TRAFO\_GEOAX\_ASSIGN\_TAB\_3[2] = 2  
\$MC\_TRAANG\_ANGLE\_1 = 45.  
\$MC\_TRAANG\_PARALLEL\_VELO\_RES\_1 = 0.2  
\$MC\_TRAANG\_PARALLEL\_ACCEL\_RES\_1 = 0.2  
\$MC\_TRAANG\_BASE\_TOOL\_1[0] = 0.0  
\$MC\_TRAANG\_BASE\_TOOL\_1[1] = 0.0  
\$MC\_TRAANG\_BASE\_TOOL\_1[2] = 0.0

### チェーニング変換

; 4 番目 TRACON (TRAORI / TRAANG チェーニング)

\$MC\_TRAFO\_TYPE\_4 = 8192  
\$MC\_TRACON\_CHAIN\_1[0] = 1  
\$MC\_TRACON\_CHAIN\_1[1] = 3  
\$MC\_TRACON\_CHAIN\_1[2] = 0  
\$MC\_TRAFO\_GEOAX\_ASSIGN\_TAB\_4[0] = 2  
\$MC\_TRAFO\_GEOAX\_ASSIGN\_TAB\_4[1] = 1  
\$MC\_TRAFO\_GEOAX\_ASSIGN\_TAB\_4[2] = 3

; 5 番目 TRACON (TRANSMIT / TRAANG チェーニング)

\$MC\_TRAFO\_TYPE\_5 = 8192  
\$MC\_TRACON\_CHAIN\_2[0] = 2  
\$MC\_TRACON\_CHAIN\_2[1] = 3  
\$MC\_TRACON\_CHAIN\_2[2] = 0  
\$MC\_TRAFO\_GEOAX\_ASSIGN\_TAB\_5[0] = 1  
\$MC\_TRAFO\_GEOAX\_ASSIGN\_TAB\_5[1] = 6  
\$MC\_TRAFO\_GEOAX\_ASSIGN\_TAB\_5[2] = 3

M30

## パートプログラム

設定変換機能を用いる NC プログラムの例：

```
                                ; 単一変換機能の呼出
                                ; ツール定義
$TC_DP1[1, 1]=120 ; ツールタイプ
$TC_DP3[1, 1] = 10 ; 工具長
n2 x0 y0 z0 a0 b0 f20000 t1 d1
n4 x20

n30 TRANSMIT                    ; TRANSMIT オン
n40 x0 y20
n50 x-20 y0
n60 x0 y-20
n70 x20 y0
n80 TRAFOOF                    ; TRANSMIT オフ

n130 TRAANG(45.) ; 傾斜軸変換機能 オン, パラメータ : 角度 45°
n140 x0 y0 z20
n150 x-20 z0
n160 x0 z-20
n170 x20 z0
```

(注)

上記の例は、傾斜軸の角度はマシン上で設定可能であり、この単一変換機能が起動された時に 0 度に設定されるということを想定しています。

```
                                ; 1 番目のチェーニング変換機能 TRAORI +
                                TRAANG オン
n230 TRACON(1, 45.) ; 2 つのチェーニング変換機能の内 1 番目のチェーニング変
                        換機能をオン
                        ; 先に有効状態の変換機能 TRAANG は自動的に解除
                        ; 傾斜軸のパラメータは 45°
n240 x10 y0 z0 a3=-1 C3 =1 oriwks
n250 x10 y20 b3 = 1 c3 = 1
...

                                ; 2 番目のチェーニング変換機能 TRANSMIT +
                                TRAANG をオフ
n330 TRACON(2, 40.) ; 2 番目のチェーニング変換機能をオフ
                        ; 傾斜軸のパラメータは 40°
n335 x20 y0 z0
n340 x0 y20 z10
```

---

n350 x-20 y0 z0

n360 x0 y-20 z0

n370 x20 y0 z0

n380 TRAFOOF ; 2 番目のチェーニング変換機能をオフ

...

n1000 M30



## 2.7.7 データフィールド

### ■ TRANSMIT

### ■ インタフェース信号

DB 番号	ビット, バイト	名称	参照
チャンネル別			
21, ...	33.6	変換機能アクティブ	F2

### ■ マシンデータ

番号	一般事項	名前	参照
チャンネル別 (\$MC_ ... )			
20110	RESET_MODE_MASK	電源オンおよび RESET / パートプログラム終了後基本制御位置の定義	K2
20140	TRAFO_RESET_VALUE	基本変換位置	K2
22534	TRAFO_CHANGE_M_CODE	ジオメトリ軸に関する変換切換え用 M コード	
24100	TRAFO_TYP_1	1 番目の変換機能のタイプ, 必要に応じて, 軸シーケンスを含む	F2
24110	TRAFO_AXES_IN_1	1 番目の変換機能入力時点で軸を割当てる	F2
24120	TRAFO_GOEAX_ASSIGN_TAB_1	1 番目の変換機能に関するジオメトリ軸を割当てる	F2
24200	TRAFO_TYP_2	2 番目の変換機能のタイプ, 必要に応じて, 軸シーケンスを含む	F2
24210	TRAFO_AXES_IN_2	2 番目の変換機能入力時点で軸を割当てる	F2
24220	TRAFO_GOEAX_ASSIGN_TAB_2	2 番目の変換機能に関するジオメトリ軸を割当てる	F2
24300	TRAFO_TYP_3	3 番目の変換機能のタイプ, 必要に応じて, 軸シーケンスを含む	F2
24310	TRAFO_AXES_IN_3	3 番目の変換機能入力時点で軸を割当てる	F2
24320	TRAFO_GOEAX_ASSIGN_TAB_3	3 番目の変換機能に関するジオメトリ軸を割当てる	F2
24400	TRAFO_TYP_4	4 番目の変換機能のタイプ, 必要に応じて, 軸シーケンスを含む	F2
24410	TRAFO_AXES_IN_4	4 番目の変換機能入力時点で軸を割当てる	F2
24420	TRAFO_GOEAX_ASSIGN_TAB_4	4 番目の変換機能に関するジオメトリ軸を割当てる	F2
24430	TRAFO_TYP_5	5 番目の変換機能のタイプ, 必要に応じて, 軸シーケンスを含む	F2
24432	TRAFO_AXES_IN_5	5 番目の変換機能入力時点で軸を割当てる	F2
24434	TRAFO_GOEAX_ASSIGN_TAB_5	5 番目の変換機能に関するジオメトリ軸を割当てる	F2
24440	TRAFO_TYP_6	6 番目の変換機能のタイプ, 必要に応じて, 軸シーケンスを含む	F2
24442	TRAFO_AXES_IN_6	6 番目の変換機能入力時点で軸を割当てる	F2
24444	TRAFO_GOEAX_ASSIGN_TAB_6	6 番目の変換機能に関するジオメトリ軸を割当てる	F2
24450	TRAFO_TYP_7	7 番目の変換機能のタイプ, 必要に応じて, 軸シーケンスを含む	F2

番号	一般事項	名前	参照
24452	TRAFO_AXES_IN_7	7 番目の変換機能入力時点で軸を割当てる	F2
24454	TRAFO_GOEAX_ASSIGN_TAB_7	7 番目の変換機能入力時点で軸を割当てる	F2
24460	TRAFO_TYP_8	8 番目の変換機能のタイプ，必要に応じて，軸シーケンスを含む	F2
24462	TRAFO_AXES_IN_8	8 番目の変換機能入力時点で軸を割当てる	F2
24464	TRAFO_GOEAX_ASSIGN_TAB_8	8 番目の変換機能に関するジオメトリ軸を割当てる	F2
24900	TRANSMIT_ROT_AX_OFFSET_1	回転軸の原点からのずれを度単位で設定 (1 番目の TRANSMIT)	
24910	TRANSMIT_ROT_SIGN_IS_PLUS_1	TRANSMIT に関する回転軸の符号 (1 番目の TRANSMIT)	
24911	TRANSMIT_POL_SIDE_FIX_1	極前後の作業範囲の制限，1 番目の変換機能	
24920	TRANSMIT_BASE_TOOL_1	ジオメトリ軸の原点からのツール原点の距離 (1 番目の TRANSMIT)	
24950	TRANSMIT_ROT_AX_OFFSET_2	回転軸の原点からのずれを度単位で設定 (2 番目の TRANSMIT)	
24960	TRANSMIT_ROT_SIGN_IS_PLUS_2	TRANSMIT に関する回転軸の符号 (2 番目の TRANSMIT)	
24961	TRANSMIT_POL_SIDE_FIX_2	極前後の作業範囲の制限，2 番目の変換機能	
24970	TRANSMIT_BASE_TOOL_2	ジオメトリ軸の原点からのツール原点の距離 (2 番目の TRANSMIT)	

## ■ アラーム

TRACYL 変換時に発生する可能性のあるアラームは，Diagnostic Guide（診断ガイド）および MMC 101/102/103 システムのオンラインヘルプで説明されています。

参照： アラーム一覧

## ■ TRACYL

### ■ インタフェース信号

DB 番号	ビット, バイト	名前	参照
チャンネル別			
21, ...	33.6	変換機能アクティブ	F2

### ■ マシンデータ

番号	一般事項	名前	参照
チャンネル別 (\$MC_ ... )			
20110	RESET_MODE_MASK	電源および RESET / パートプログラム終了後基本制御位置の定義	K2
20140	TRAFO_RESET_VALUE	基本変換位置	K2
24100	TRAFO_TYP_1	1 番目の変換機能のタイプ, 必要に応じて, 軸シーケンスを含む	F2
24110	TRAFO_AXES_IN_1	1 番目の変換機能入力時点で軸を割当てる	F2
24120	TRAFO_GOEAX_ASSIGN_TAB_1	1 番目の変換機能に関するジオメトリ軸を割当てる	F2
24200	TRAFO_TYP_2	2 番目の変換機能のタイプ, 必要に応じて, 軸シーケンスを含む	F2
24210	TRAFO_AXES_IN_2	2 番目の変換機能入力時点で軸を割当てる	F2
24220	TRAFO_GOEAX_ASSIGN_TAB_2	2 番目の変換機能に関するジオメトリ軸を割当てる	F2
24300	TRAFO_TYP_3	3 番目の変換機能のタイプ, 必要に応じて, 軸シーケンスを含む	F2
24310	TRAFO_AXES_IN_3	3 番目の変換機能入力時点で軸を割当てる	F2
24320	TRAFO_GOEAX_ASSIGN_TAB_3	3 番目の変換機能に関するジオメトリ軸を割当てる	F2
24400	TRAFO_TYP_4	4 番目の変換機能のタイプ, 必要に応じて, 軸シーケンスを含む	F2
24410	TRAFO_AXES_IN_4	4 番目の変換機能入力時点で軸を割当てる	F2
24420	TRAFO_GOEAX_ASSIGN_TAB_4	4 番目の変換機能に関するジオメトリ軸を割当てる	F2
24430	TRAFO_TYP_5	5 番目の変換機能のタイプ, 必要に応じて, 軸シーケンスを含む	F2
24432	TRAFO_AXES_IN_5	5 番目の変換機能入力時点で軸を割当てる	F2
24434	TRAFO_GOEAX_ASSIGN_TAB_5	5 番目の変換機能に関するジオメトリ軸を割当てる	F2
24440	TRAFO_TYP_6	6 番目の変換機能のタイプ, 必要に応じて, 軸シーケンスを含む	F2
24442	TRAFO_AXES_IN_6	6 番目の変換機能入力時点で軸を割当てる	F2
24444	TRAFO_GOEAX_ASSIGN_TAB_6	6 番目の変換機能に関するジオメトリ軸を割当てる	F2
24450	TRAFO_TYP_7	7 番目の変換機能のタイプ, 必要に応じて, 軸シーケンスを含む	F2
24452	TRAFO_AXES_IN_7	7 番目の変換機能入力時点で軸を割当てる	F2
24454	TRAFO_GOEAX_ASSIGN_TAB_7	7 番目の変換機能入力時点で軸を割当てる	F2
24460	TRAFO_TYP_8	8 番目の変換機能のタイプ, 必要に応じて, 軸シーケンスを含む	F2
24462	TRAFO_AXES_IN_8	8 番目の変換機能入力時点で軸を割当てる	F2
24464	TRAFO_GOEAX_ASSIGN_TAB_8	8 番目の変換機能に関するジオメトリ軸を割当てる	F2
24800	TRACYL_ROT_AX_OFFSET_1	回転軸の原点からのずれを度単位で設定 (1 番目の TRACYL)	

番号	一般事項	名前	参照
24810	TRACYL_ROT_SIGN_IS_PLUS_1	TRACYL に関する回転軸の符号 (1 番目の TRACYL)	
24820	TRACYL_BASE_TOOL_1	ジオメトリ軸の原点からのツール原点の距離 (1 番目の TRACYL)	
24850	TRACYL_ROT_AX_OFFSET_2	回転軸の原点からのずれを度単位で設定 (2 番目の TRACYL)	
24860	TRACYL_ROT_SIGN_IS_PLUS_2	TRANSMIT に関する回転軸の符号 (2 番目の TRACYL)	
24870	TRACYL_BASE_TOOL_2	ジオメトリ軸の原点からのツール原点の距離 (2 番目の TRACYL)	
22534	TRAFO_CHANGE_M_CODE	ジオメトリ軸に関する変換切換え用 M コード	

## ■ アラーム

TRACYL 変換時に発生する可能性のあるアラームは、「アラーム一覧 (」 診断ガイド) および MMC 101/102/103 システムのオンラインヘルプで説明されています。

参照： アラーム一覧

## ■ TRAANG（傾斜軸）

### ■ インタフェース信号

DB 番号	ビット, バイト	名前	参照
チャンネル別			
21, ...	33.6	変換機能アクティブ	F2

### ■ マシンデータ

番号	一般事項	名前	参照
チャンネル別 (\$MC_ ... )			
20110	RESET_MODE_MASK	電源オンおよび RESET / パートプログラム終了後基本制御位置の定義	K2
20140	TRAFO_RESET_VALUE	基本変換位置	K2
22534	TRAFO_CHANGE_M_CODE	ジオメトリ軸に関する変換切換え用 M コード	
24100	TRAFO_TYP_1	1 番目の変換機能のタイプ, 必要に応じて, 軸シーケンスを含む	F2
24110	TRAFO_AXES_IN_1	1 番目の変換機能入力時点で軸を割当てる	F2
24120	TRAFO_GOEAX_ASSIGN_TAB_1	1 番目の変換機能に関するジオメトリ軸を割当てる	F2
24200	TRAFO_TYP_2	2 番目の変換機能のタイプ, 必要に応じて, 軸シーケンスを含む	F2
24210	TRAFO_AXES_IN_2	2 番目の変換機能入力時点で軸を割当てる	F2
24220	TRAFO_GOEAX_ASSIGN_TAB_2	2 番目の変換機能に関するジオメトリ軸を割当てる	F2
24300	TRAFO_TYP_3	3 番目の変換機能のタイプ, 必要に応じて, 軸シーケンスを含む	F2
24310	TRAFO_AXES_IN_3	3 番目の変換機能入力時点で軸を割当てる	F2
24320	TRAFO_GOEAX_ASSIGN_TAB_3	3 番目の変換機能に関するジオメトリ軸を割当てる	F2
24400	TRAFO_TYP_4	4 番目の変換機能のタイプ, 必要に応じて, 軸シーケンスを含む	F2
24410	TRAFO_AXES_IN_4	4 番目の変換機能入力時点で軸を割当てる	F2
24420	TRAFO_GOEAX_ASSIGN_TAB_4	4 番目の変換機能に関するジオメトリ軸を割当てる	F2
24430	TRAFO_TYP_5	5 番目の変換機能のタイプ, 必要に応じて, 軸シーケンスを含む	F2
24432	TRAFO_AXES_IN_5	5 番目の変換機能入力時点で軸を割当てる	F2
24434	TRAFO_GOEAX_ASSIGN_TAB_5	5 番目の変換機能に関するジオメトリ軸を割当てる	F2
24440	TRAFO_TYP_6	6 番目の変換機能のタイプ, 必要に応じて, 軸シーケンスを含む	F2
24442	TRAFO_AXES_IN_6	6 番目の変換機能入力時点で軸を割当てる	F2
24444	TRAFO_GOEAX_ASSIGN_TAB_6	6 番目の変換機能に関するジオメトリ軸を割当てる	F2
24450	TRAFO_TYP_7	7 番目の変換機能のタイプ, 必要に応じて, 軸シーケンスを含む	F2
24452	TRAFO_AXES_IN_7	7 番目の変換機能入力時点で軸を割当てる	F2
24454	TRAFO_GOEAX_ASSIGN_TAB_7	7 番目の変換機能入力時点で軸を割当てる	F2
24460	TRAFO_TYP_8	8 番目の変換機能のタイプ, 必要に応じて, 軸シーケンスを含む	F2
24462	TRAFO_AXES_IN_8	8 番目の変換機能入力時点で軸を割当てる	F2
24464	TRAFO_GOEAX_ASSIGN_TAB_8	8 番目の変換機能に関するジオメトリ軸を割当てる	F2
24700	TRAANG_ANGLE_1	傾斜軸の角度を度単位で (1 番目の TRAANG)	

番号	一般事項	名前	参照
24710	TRAANG_BASE_TOOL_1	ジオメトリ軸の原点からのツール原点の距離 (1 番目の TRAANG)	
24720	TRAANG_PARALLEL_VELO_RES_1	補償動作用平行軸の速度リザーブ (1 番目の TRAANG)	
24721	TRAANG_PARALLEL_VELO_RES_2	補償動作用平行軸の速度リザーブ (2 番目の TRAANG)	
24750	TRAANG_ANGLE_2	傾斜軸の角度を度単位で (2 番目の TRAANG)	
24760	TRAANG_BASE_TOOL_2	ジオメトリ軸の原点からのツール原点の距離 (2 番目の TRAANG)	
24770	TRAANG_PARALLEL_ACCEL_RES_1	補償動作用平行軸の軸加速レートリザーブ (1 番目の TRAANG)	
24771	TRAANG_PARALLEL_ACCEL_RES_2	補償動作用平行軸の軸加速レートリザーブ (2 番目の TRAANG)	
24995	TRACON_CHAIN_1	1 番目のチェーニング変換機能の変換動作チェーニング	
24996	TRACON_CHAIN_2	2 番目のチェーニング変換機能の変換動作チェーニング	

## ■ アラーム

TRAANG 変換時に発生する可能性のあるアラームは、「アラーム一覧」(診断ガイド) および MMC 101/102/103 システムのオンラインヘルプで説明されています。

参照： アラーム一覧

---

## 2.8 計測機能 (M5)

### 2.8.1 詳細説明

#### チャンネル別測定方法

チャンネル別測定方法は、ソフトウェアバージョン 1 以降で使用できます。

測定運転を起動し、該当する測定モデルを定義するトリガイベントは、パートプログラムにプログラムされます。命令は、この特殊ブロックでプログラムされるすべての軸に与えられます。

#### 軸の測定方法

軸の測定は、ソフトウェアバージョン 4.1 以降で使用できます。

測定運転を起動し、該当する測定モデルを定義するトリガイベントは、パートプログラムにプログラムされます。測定のための測定モデルは、計測しなければならない軸と共に定義されます。

#### 測定サイクル

測定サイクルの扱い方についての説明は、下記に記載してあります。

### 2.8.2 詳細説明

#### ■ ハードウェア要件

#### ■ 適正なプローブ

##### 一般

ツールとワークの寸法を読取るには、変位すると定常信号（パルスではない）を出力するタッチプローブが必要です。

センサは、ほとんどバウンスせずに作動しなくてはなりません。ほとんどのセンサがこのように作動するよう機械的に調整できます。

異なるタイプのプローブが、様々なメーカーによって販売されています。プローブが変位する方向の数によって 3 種類に分けられます。

(図 2.83)

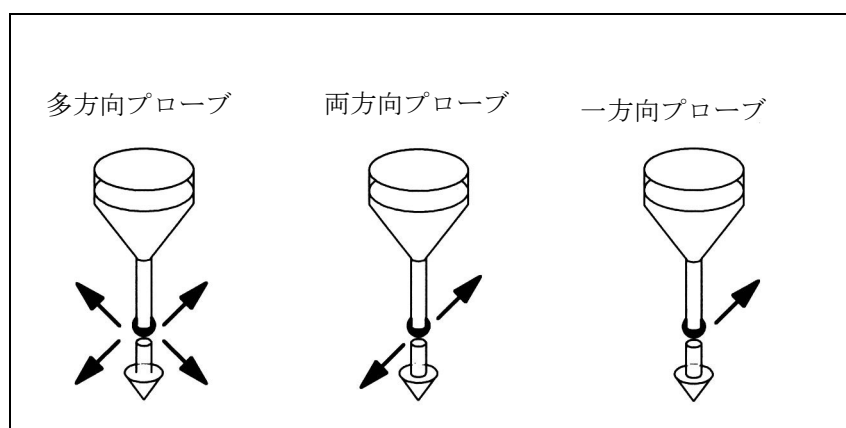


図 2.83 プローブのタイプ

表 2.9 プローブのタイプと用途との関係

プローブのタイプ	旋盤		フライスマシンおよびマシニングセンタ
	ツール測定値	ワークピース測定値	ワークピース測定値
多方向	×	×	×
両方向	-	×	×
一方向	-	-	×

両方向プローブは、ワーク測定用として、旋盤で使用しなければなりません。一方で、一方向プローブは、ワーク測定用としては、フライスマシンでもマシニングセンタでも使用することができます。

### 多方向プローブ (3D)

このタイプのプローブは、ツールとワークの寸法測定用に無条件で使用できます。

### 両方向プローブ

このタイプのプローブは、フライスマシンおよびマシニングセンタの一方向プローブと同様にして使用します。両方向プローブは、旋盤でワークピースの測定を行なうのに使用することもできます。

### 一方向プローブ

このタイプのプローブは、いくつかの制限事項に従って使用し、フライスマシンとマシニングセンタでワークピースを測定することができます。

### 一方向プローブを使用したスピンドル位置

フライスマシンとマシニングセンタで、このタイプのプローブを使用するには、NC ファンクションの SPOS でスピンドルを位置決めし、360° に渡るプローブからの切換え信号をレシーバステーション（マシンステータ）に転送できる必要があります。



---

プローブは、スピンドルが 0 度に位置決めされた状態で、下記の方  
向で測定を行なうことができるように、スピンドルに機械的に位置  
合わせしなければなりません。

表 2.10 プローブの調整のためのスピンドル位置

	0 度スピンドル位置における測定値
X-Y プレーン G17	正の X 方向
Z-X プレーン G18	正の Z 方向
Y-Z プレーン G19	正の Y 方向

一方向プローブを使用する場合、測定サイクルにおいて SPOS でスピ  
ンドルを数回位置決めする必要があるため、測定時間が長くなり  
ます。

## ■ プローブの接続

## YS840DI への接続

プローブは、NCU モジュールの正面パネル上の I/O デバイスインタフェース X121 を介して接続されます。

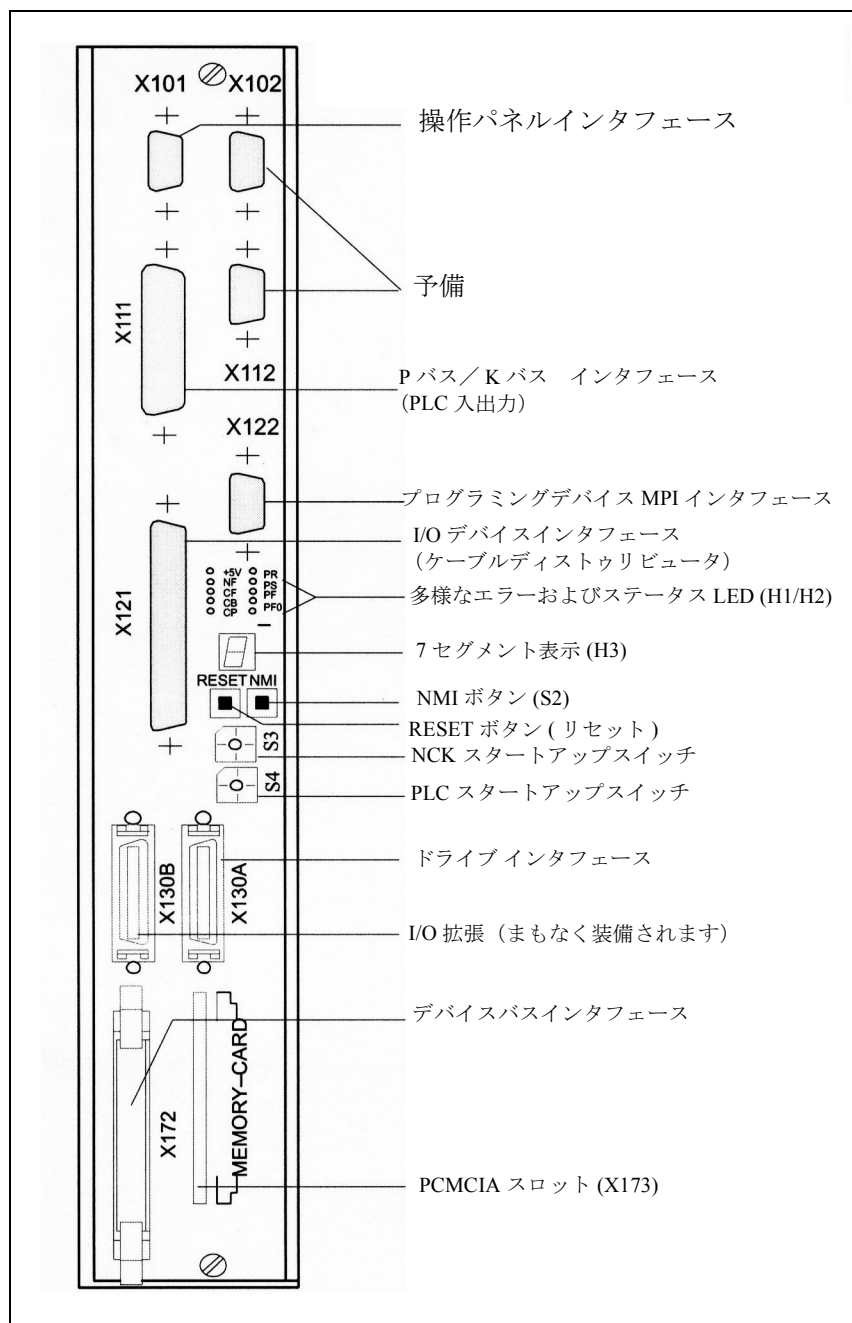


図 2.84 NCU モジュールのインタフェース，操作部分および表示部分

## インタフェース

プローブのインタフェース接続は、下記を介して行ないます。

- I/O デバイスインタフェース

37 ピン D-sub プラグコネクタ (X121), 最高 2 つのプローブが接続可能です。

24 V の負荷電力供給が X121 にも接続されます。

表 2.11 フロントコネクタ X121 のための PIN 割当て表からの抜粋

	PIN	指定	
x121	1	M24EXT	外部電力供給 外部接地
	2	M24EXT	外部接地
	...	...	...
	9	MEPUS 0	プローブ 1 の接続 パ ルス信号入力 of 測定
	10	MEPUC 0	パルス共通入力 of 測定
	...	...	...
	20	P24EXT	外部電力供給 P 24 V 外部
	21	P24EXT	P 24 V 外部
	...	...	...
	28	MEPUS 1	プローブ 2 の接続 パルス信号入力 of 測定
	29	MEPUC 1	パルス信号入力 of 測定
	...	...	...

インタフェース（例えば、ピン割当て）は下記に詳しく説明してあります。

参照： /PFD/, Hardware Planning Guide

## ■ チャンネル別測定値

## ■ ソフトウェア要件

### NC ソフトウェアバージョン

チャンネル別測定の機能は、ソフトウェアバージョン 1 以降で使用できます。

### MMC ソフトウェアバージョン

"Measurement result display" (測定結果表示) および "Parameter assignment via input dialog" (入力ダイアログからのパラメータ割当て) 機能には、MMC ソフトウェアバージョン 3.2 以降が必要です。

## ■ 測定モデル

### MEAS 測定コマンドと MEAW 測定コマンド

測定オペレーションは、パートプログラムから起動します。トリガイイベントと測定モデルがプログラムされます。

2 種類の測定モデルが使用できます。

- MEAS: 移動距離削除ありで測定

例:

N10 G01 F300 X300 Z200 MEAS=-2

トリガイイベントは、第 2 プローブ (2) の立下りエッジ (ー) です。

- MEAW: 移動距離削除なしで測定

例:

N20 G01 F300 X300 Z100 MEAW=1

トリガイイベントは、第 1 プローブ (1) の立上りエッジです。

測定ジョブは、RESET あるいは、プログラムが新しいブロックに進んだときに、強制終了されます。

(注)

GEO 軸が測定ブロックにプログラムされている場合、測定値は、すべて現行の GEO 軸について保存されます。

変形に関わっている軸が測定ブロックにプログラムされている場合は、この変形に関わっている軸のすべての測定値が保存されます。

### プローブのステータス

ソフトウェアバージョン 4 では、パートプログラムと同期するアクションで、プローブのステータスを直接調べます。

\$A\_PROBE[n] n = プローブ

\$A\_PROBE[n]==1: ふれのあるプローブ

\$A\_PROBE[n]==0: ふれのないプローブ

## ■ 測定結果

### PP での測定結果の読取り

測定コマンドの結果は、NCK のシステムデータに保存されており、パートプログラムのシステム変数を介して読取ることができます。

- システム変数 \$AC\_MEA[<No>]

測定ジョブのステータス信号を調べる。

<No.> は、プローブ（1 または 2）を示す。

この変数は、測定開始時に削除されます。プローブが起動基準（立上りエッジまたは立下りエッジ）を満たすと、直ちに変数がセットされます。測定ジョブの実行は、パートプログラムでチェックすることができます。

- システム変数 \$AA\_MM[<Axis>]

マシン座標系による測定結果にアクセスする。

パートプログラムおよび同期するアクションにて読取り。

<Axis> は測定軸 (X,Y,...) の名称を示す。

- システム変数 \$AA\_MM[axis]

ワークピース座標系による測定結果にアクセスする。

パートプログラムおよび同期するアクションにて読取り。

<Axis> は測定軸 (X,Y,...) の名称を示す。

参照：YS840DI ユーザーズマニュアル・プログラミング編

### PLC サービス表示

NC プログラムによって、プローブは機能的に試験されます。

測定信号は、診断メニュー "PLC status" のプログラムエンドでチェックすることができます。

表 2.12 測定信号のステータス表示

	ステータス表示
ふれのあるプローブ 1	DB10 DB B107.0
ふれのあるプローブ 2	DB10 DB B107.1

軸の現行測定ステータスは、インタフェース信号 DB(31-48) DBX62.3 によって表示されます。

ビット 3 = 1: 測定アクティブ

ビット 3 = 0: 測定イナクティブ

(注)

この信号は、すべての測定ファンクションについて表示でき、\$AA\_MEAACT["axis"] によって同期化されたアクションでも読取ることができます。

## ■ 軸測定（オプション）

測定オペレーションは、パートプログラムからでも同期するアクションからでも開始することができます。測定モデル、エンコーダ、および最高4つのトリガイイベントがプログラムされており、トリガイイベントはプローブ番号（1または2）および（立上り／立下り信号エッジ）で構成されます。

測定値がエンコーダ1およびエンコーダ2から各トリガイイベントについて保存される場合は、2つのトリガイイベントだけがプログラム可能です。

## ■ ソフトウェア要件

### NC ソフトウェアバージョン

軸測定の機能は、ソフトウェアバージョン4以降のオプションとして利用できません。

### MMC ソフトウェアバージョン

"Measurement result display"（測定結果表示）ファンクションおよび"Parameter assignment via input dialog"（入力ダイアログからのパラメータ割当て）機能には、MMC ソフトウェアバージョン3.2以降が必要です。

## ■ 補足条件

### 運転モード切換え

パートプログラムからの測定ジョブ

パートプログラムで起動される測定ジョブは、運転モードの切り替えの影響を受けません。しかし、パートプログラムが新しいブロックに進むと直ちに削除されます。

RESET で測定ジョブを強制終了します。

同期するアクションからの測定ジョブ

モーダル同期化アクションによって起動された測定ジョブは、運転モードの切換えの影響を受けません。このジョブは、ブロックリミットを超えても、モーダルについてはアクティブです。

### ブロックサーチ

パートプログラムからの測定ジョブ

このジョブは開始されません。測定チェックバック信号が用意されていません。

同期するアクションからの測定ジョブ

モーダル測定ジョブは、プログラムされた条件が満たされるまで、起動されません。

## Repos

再位置決め機能

パートプログラムからの測定ジョブ

測定ジョブが現時点で実行されている場合は、Repos ブロック後に再び強制終了されてから再始動します。このジョブがすでに完了してしまっている場合は、再始動しません。

同期化されたアクションからの測定ジョブ

起動された測定ジョブは、影響を受けません。

### ■ 測定モデル

測定モデルによって、トリガイイベントが並行、あるいは昇順で続いて起動する必要があるかどうかを指定し、さらに計測の数を定義します。

#### 測定モデル 1

ユーザは、同じ位置制御サイクルで異なった最高 4 とおりのトリガイイベントをプログラムすることができます。

測定信号エッジは、時間順に評価されます。

- 2つの測定信号エッジをもった最高 2つのプローブが、各測定ジョブごとにプログラムできます。2つのエンコーダを使用する場合、プログラムされたトリガイイベントの数は 2 等分されます。
- 6つの軸モジュールがインストールされている場合は、測定モデル 1 が制御装置の内部では測定モデル 2 で現れます。
- 移動速度は、各位置制御クロックサイクルの、2つの同一トリガイイベントの間の最短距離以下でなければなりません。

(注)

このモードの場合、最新の測定信号エッジが受け取られたときに出された補正值が、すべての測定値について計算されます。

#### 測定モデル 2

ユーザは、シーケンス内において最高 4 種類のトリガイイベントを次から次へとプログラムすることができます。測定信号エッジの評価は、一度に 1つのトリガイイベントを起動し、プログラムされたシーケンスにおいて起こります。

- トリガイイベントは、プログラムされたシーケンスにおいてのみ発生します。
- 移動速度は、各位置制御サイクルにおける 2つのトリガイイベントの間の最短距離以下でなければなりません。

(注)

測定は、シミュレーション中の軸では作動しません。

### プローブのステータス

ソフトウェアバージョン 4 では、パートプログラムおよび同期するアクションで、プローブのステータスを直接調べることができます。

\$A\_PROBE[n] n = プローブ

\$A\_PROBE[n]=1: ふれのあるプローブ

\$A\_PROBE[n]==0: ふれのないプローブ

## ■ プログラミング

### プログラミング

軸測定は、移動距離削除のあるなしに関わらず、プログラムが可能です。

MEASA 移動距離削除あり

MEAWA 移動距離削除なし

MEASA[axis] = (モード, トリガイイベント 1, トリガイイベント 2, トリガイイベント 3, トリガイイベント 4)

パラメータの説明 :

- 軸 :

チャンネル軸名 (X,Y,...)

- モード :

1 の位

0 = 測定ジョブを強制終了する (例えば, 同期するアクション)

1 = 同時に起動できる最高 4 つまでのトリガイイベント

2 = 順に続けて起動できる最高 4 つまでのトリガイイベント

10 の位 (= エンコーダの選択)

0/ セットしない = アクティブな測定系を使う

1 = 第 1 測定系

2 = 第 2 測定系 (インストールされている場合)

インストールされていなければ, 第 1 測定系を使用し, アラームは出力されない)

3 = 第 1 測定系および第 2 測定系

2 種類の測定系によって測定が行なわれる場合, 最大 2 つのトリガイイベントがプログラムできます。どちらのエンコーダの測定値も, 2 つのトリガイイベントのそれぞれについて記録されます。

- トリガイイベント

1 = プローブ 1 の立上りエッジ

-1 = プローブ 1 の立下りエッジ

2 = プローブ 2 の立上りエッジ

-2 = プローブ 2 の立下りエッジ

(注)

MEASA および MEAWA は, 1 つのブロックでプログラムできます。

MEASA がすでにプログラムされている軸は, すべてのプログラムされたトリガイイベントが出るまで, 減速されません。

パートプログラムから始動した測定ジョブは, RESET により, またはプログラムが新しいブロックに進むときに強制終了されます。

アラーム 21701 は, MEASA/MEAWA が MEAS/MEAW と同じブロックにプログラムされています。



アラームは、下記のオンラインヘルプに記載されています。  
参照： アラーム一覧

## ■ 測定結果

### 測定結果

測定コマンドの結果は、NCK システムデータに保存されており、システム変数を用いてパートプログラムで読み取ることができます。

- システム変数 \$AC\_MEA[No.]

測定ジョブのステータス信号を調べる。

<No.> は、プローブ（1 または 2）を示す。

この変数は、測定開始時に削除されます。プローブが起動基準（立上りエッジまたは立下りエッジ）を満たすと、直ちに変数がセットされます。測定ジョブの実行は、パートプログラムでチェックすることができます。

- システム変数 \$AA\_MM1[axis] から \$AA\_MM4[axis]

機械座標系によるトリガ信号の測定結果にアクセスする。

パートプログラムおよび同期するアクションで読取る。

<Axis> は測定軸 (X,Y,...) の名称を示す。

- システム変数 \$AA\_MW1[axis] から \$AA\_MW4[axis]

機械座標系のトリガ信号の測定結果にアクセスする。

パートプログラムおよび同期するアクションで読取る。

<Axis> は測定軸 (X,Y,...) の名称を示す。

### プログラミング

2 種類の測定系が測定に使用される場合、最高 2 つのトリガイイベントがプログラムできます。どちらのエンコーダの測定値も、2 つのトリガイイベントのそれぞれについて記録されます。

1 つのトリガイイベント

\$AA\_MM1[axis] = トリガイイベント 1, エンコーダ 1 からの測定値

\$AA\_MM2[axis] = トリガイイベント 1, エンコーダ 2 からの測定値

2 つのトリガイイベント

\$AA\_MM1[axis] = トリガイイベント 1, エンコーダ 1 からの測定値

\$AA\_MM2[axis] = トリガイイベント 1, エンコーダ 2 からの測定値

\$AA\_MM3[axis] = トリガイイベント 2, エンコーダ 1 からの測定値

\$AA\_MM4[axis] = トリガイイベント 2, エンコーダ 2 からの測定値

PLC サービス表示

NC プログラムによって、プローブは機能的に試験されます。

測定信号は、診断メニュー "PLC status" のプログラムのエンドでチェックすることができます。

表 2.13 測定信号のステータス表示

	ステータス表示
プローブ 1 が変位	DB10 DB B107.0
プローブ 2 が変位	DB10 DB B107.1

参照： YS840DI ユーザーズマニュアル・プログラミング編

■ 連続測定（周期測定）

すべての測定は、あらかじめ定義された FIFO 変数に書き込まれます。測定値の数は、機械データで定義されます。

- ファンクションの正確なオペレーションは、± 8:1 の IPO/ 位置制御サイクル比でのみ保証されます。
- FIFO メモリの内容は、一度だけ読むことができます。測定結果が、2 回目以降使用されるときは、読み出し値をユーザデータにバッファしなければなりません。

MEAC

移動距離削除なしの連続、軸測定

MEAC[ 軸 ] = (モード, 測定メモリ, トリガイイベント 1, トリガイイベント 2, トリガイイベント 3, トリガイイベント 4)

パラメータの説明：

- 軸：  
チャンネル軸名 (X,Y,...)
- モード：  
1 の位  
0 = 測定ジョブを強制終了する（例えば、同期化されたアクション）  
1 = 同時に起動できる最高 4 つのトリガイイベント  
2 = 順に続けて起動できる最高 4 つのトリガイイベント  
最初のトリガ信号がすでにアクティブである場合、エラーが出力される  
3 = 順に続いて起動される最高 4 つのトリガイイベント  
最初のトリガ信号がすでにアクティブの場合、エラーは出力されない  
10 の位 (= エンコーダの選択 )  
0/ イネーブルされない = アクティブな測定系  
1 = 第 1 測定系  
2 = 第 2 測定系（インストールされている場合。そうでない場合は、第 1 測定系を使用し、アラームは出力されない）

3 = 第 1 測定系および第 2 測定系

2 種類の測定系によって測定が行なわれる場合、最大 2 つのトリガイイベントがプログラムできます。

- 測定メモリ：

FIFO の数

- トリガイイベント

1 = プローブ 1 の立上りエッジ

-1 = プローブ 1 の立下りエッジ

2 = プローブ 2 の立上りエッジ

-2 = プローブ 2 の立下りエッジ

軸測定値はマシン座標系で使用でき、たとえば、\$AC\_FIFO1 のように、ユーザによって定義された FIFO 変数に書き込まれます。2 つのプローブが測定できるように構成されるとき、2 番目のプローブからの測定値は、次の FIFO に別に保存されます。

測定値の数は、MD 28264:LEN\_AC\_FIFO によって制限されます。したがって、変数 \$AC\_MEA と \$AA\_MM は関連ありません。

測定値は、パートプログラムの FIFO からでも、同期化されたアクションからでも読取ることができます。

測定は、下記の状態になるまで、アクティブです。

- MEAC["axis"] = (0) がプログラムされるまで
- FIFO が満杯になる
- RESET を押すか、あるいはプログラムエンド M02/M30 が見つかるまで

## エンドレス測定

エンドレス測定を実施するには、FIFO 値がパートプログラムから周期的に読まれなければなりません。測定値が FIFO メモリから読み取られ、処理される頻度は NC の書き込みレートに一致しなければなりません。有効なエントリの数は、FIFO 変数に読みこむことができます。

測定値の定義された数に達するには、測定ファンクションは、プログラムによって暗示的に選択解除されなければなりません。

## FIFO 変数

FIFO 変数の定義については、下記を参照のこと。

参照： /FB2/S5, Synchronized Actions

## ■ 測定精度および機能試験

### ■ 測定精度

#### 精度

測定信号の伝達時間は、使用されるハードウェアで決まります。

ドライブのインストレーションの遅延時間は、プローブの反応時間に 3.625 ～ 9.625 ミリ秒を加算した時間です。

測定の不確実性は下記のように計算されます。

測定の不確実性 = 測定信号伝達時間 × 移動速度

許容可能な移動速度は、プログラムされた測定信号エッジ数および IPO クロックサイクルと位置制御クロックサイクルとの間の比によって決まります。

測定結果の正確さは、1 以上の同一トリガ信号と、4 種類の異なるトリガ信号が各位置制御サイクルに達する移動速度の場合にのみ保証できます。

### ■ プローブの機能試験法

#### 機能試験の例

%\_N\_TEST\_PROBE\_MPF

;\$PATH=/\_N\_MPF\_DIR

;プローブインタフェース用試験プログラム

N05 DEF INT MTSIGNAL ;トリガステータスのフラグ

N10 DEF INT ME\_NR=1 ;入力数の測定

N20 DEF REAL MEASVALUE\_IN\_X

N30 G17 T1 D1 ;ツールの前選択

;プローブのオフセット

N40 \_ANF:G0 G90 X0 F150 ;開始位置および

;測定速度

N50 MEAS=ME NR G1 X 100 ;測定時の測定値

;X 軸に 1 を入力

N60 STOPRE

N70 MTSIGNAL=\$AC\_MEA[1] ;ソフトウェア切換え信号を読み込む

;第 1 測定入力時

N80 IF MTSIGNAL ==0GOTO FEHL1; 信号の評価

N90 MEASVALUE\_IN\_X=\$AA\_MW[X]; 測定値を

;ワークピース寸法に読み込む

N95 M0

N100 M02

N110 \_FEHL1:MSG (「プローブは切替っていない」)

N120 M0

N130 M02

### 2.8.3 補足条件

軸測定の可能な機能は、SW パッケージ 4 で使用できます。

### 2.8.4 データの説明 (MD,SD)

13200 MD 番号	MEAS_PROBE_LOW_ACTIVE プローブの切換え特性	
初期設定：偽 測定サイクルがある場合：偽	最小入力リミット：偽	最大入力リミット：真
変更が有効になるための条件： 電源オン	保護レベル：2/7	単位：-
データタイプ：BOOLEAN	適用開始 SW バージョン：2.2 以降	
意味：	値 0:( 初期設定 ) ふれのないステータス 0 V ふれのあるステータス 24 V 値 1 ふれのないステータス 24 V ふれのあるステータス 0 V	

28264 MD 番号	LEN_AC_FIFOFIFO 変数の長さ \$AC_FIFO...	
初期設定：0	最小入力リミット：0	最大入力リミット：10000
変更が有効になるための条件： 電源オン	保護レベル：2/7	単位：-
データタイプ：DWORD	適用開始 SW バージョン：4.1 以降	
意味：	FIFO 変数の \$AC_FIFO1 から \$AC_FIFO10 までの長さすべての FIFO 変数の長さは同じである。	

## 2.8.5 信号の説明

DB31, ... DBX62.3 データブロック	測定ステータス 軸／スピンドルからの信号 (NCK ± PLC)	
エッジ評価：なし	信号更新：周期的	信号が有効となる最初の SW バージョン：4
信号ステータス 1 あるいは 信号遷移 0 ---> 1	「測定」ファンクションがアクティブである。 この信号は測定中に使用され、軸の現時点での測定ステータスを表示する。	
信号ステータス 0 あるいは 信号遷移 1 ---> 0	「測定」ファンクションがアクティブではない。	

DB10, ... DBX107.0 および 107.1 データブロック	プローブの始動 軸／スピンドルからの信号（ドライブ -> PLC）	
エッジ評価：なし	信号更新：周期的	信号が有効となる最初の SW バージョ ン：1.1
信号ステータス 1 あるいは 信号遷移 0 ---> 1	プローブ 1 または 2 がスタートする。	
信号ステータス 0 あるいは 信号遷移 1 ---> 0	プローブ 1 または 2 がスタートしない。	
参照：	/PHD/,"NCU 571 - 573 Manual"/PHF/,"NCU 570 Manual"	
注：	SW バージョン 3.2 以前の場合、信号は 測定オペレーションを含む NC ブロッ ク が処理されている間のみ、信号はアクティブである。	

## 2.8.6 例

### ■ 測定モデル 1

1 エンコーダによる測定

- 単一測定

- 1 プローブ

- トリガ信号は、立上りエッジと立下りエッジである

- 現行のエンコーダからの実際値

N2 MEASA[X] = (1, 1, -1) G01 X100 F100

N3 STOPRE

N4 IF \$AC\_MEA[1]==FALSE gotof END

N5 R10=\$AA\_MM1[X]

N6 R11=\$AA\_MM2[X]

N7 END:

2 つのエンコーダによる測定

- 単一測定

- 1 プローブ

- トリガ信号は、立上りエッジと立下りエッジである

- 2 つのエンコーダからの実際値

N2 MEASA[X]=(31, 1, -1) G01 X100 F100

N3 STOPRE

N4 IF \$AC\_MEA[1]==FALSE gotof END

N5 R10=\$AA\_MM1[X]

N6 R11=\$AA\_MM2[X]

N7 R12=\$AA\_MM3[X]

N8 R13=\$AA\_MM4[X]

N9 END:

### ■ 測定モード 2

- 2 つのプローブ

- トリガ信号は、立上りエッジと立下りエッジである

- 現行のエンコーダからの実際値

N2 MEASA[X] = (2, 1, -1, 2, -2) G01 X100 F100

N3 STOPRE

N4 IF \$AC\_MEA[1]==FALSE gotof PROBE2

N5 R10=\$AA\_MM1[X]

N6 R11=\$AA\_MM2[X]

N7 PROBE2

N8 IF \$AC\_MEA[2]==FALSE gotof END

---

```
N9 R12=$AA_MM3[X]
N10 R13=$AA_MM4[X]
N11 END:
```

## ■ 連続測定

### ■ プログラムされたトラバース動作の完了時の連続測定

- 測定モデル 1 で、測定値を取る
- 100 の値で測定
- 1 つのプロープ
- トリガ信号は立上りエッジである
- 現行のエンコーダからの実際値

```
N1 DEF REAL MEASVALUE[100]
N2 DEF INT INDEX=0
N3 MEAC[X]=(1, 1, -1) G01 X1000 F100
N4 MEAC[X]=(0) ; 強制終了
N5 R1=$AC_FIFO1[4] ; 測定値の数
N6 FOR INDEX=0 TO R1
N7 MEASVALUE[INDEX]=$AC_FIFO1[0] ; 測定値読み出し
N8 ENDFOR:
```

### ■ 移動距離削除で連続測定

- 最後の測定後移動距離削除
- 測定は測定モデル 1 で行なう
- 100 の値で測定
- 1 つのプロープ
- トリガ信号は立下りエッジ
- 現行エンコーダからの実際値

```
N1 DEF INT NUMBER=100
N2 DEF REAL MEASVALUE[NUMBER]
N3 DEF INT INDEX=0
N4 WHEN $AC_FIFO1[4]==NUMBER DO DELDTG (X) MEAC[X]=(0)
N5 MEAC[X]=(1, 1, -1) G01 X1000 F100 ; 測定開始
N6 R1=$AC_FIFO1[4] ; 測定値の数
N7 FOR INDEX=0 TO R1
N8 MEASVALUE[INDEX]=$AC_FIFO1[0] ; 測定値の読み出し
N9 ENDFOR:
```



## ■ いくつかのブロックをモデルで連続測定

- 測定モデル 1 で測定を行なう
- 100 の値で測定
- 1 プローブ
- トリガ信号は立下りエッジである
- 現行エッジからの実際値

```
N1 DEF INT NUMBER=100
N2 DEF REAL MEASVALUE[NUMBER]
N3 DEF INT INDEX=0
N4 ID=1 MEAC[X]=(1, 1, -1) ; 測定の開始
N5 ID=2 WHEN $AC_FIFO1[4]==NUMBER DO MEAC[X]=(0) CANCEL(2)
N6 G01 X1000 Y100
N7 X100 Y100
N8 R1=$AC_FIFO1[4] ; 測定値の数
N9 FOR INDEX=0 TO R1
N10 MEASVALUE[INDEX]=$AC_FIFO1[0] ; 測定値の読み出し
N11 ENDFOR:
```

## ■ 機能試験と繰り返し精度

### 機能試験

```
%_N_TEST_PROBE_MPF
; $PATH=/_N_MPF_DIR
; Test program for probe interface
N05 DEF INT MTSIGNAL ; トリガステータスのフラグ
N10 DEF INT ME_NR=1 ; 入力番号の測定
N20 DEF REAL MEASVALUE_IN_X
N30 G17 T1 D1 ; ツールの前選択
; プローブのオフセット
N40 _ANF: G0 G90 X0 F150 ; 始動位置および
; 測定速度
N50 MEAS=ME_NR G1 X100 ; 測定時の測定
; X 軸に 1 を入力
N60 STOPRE
N70 MTSIGNAL=$AC_MEA[1] ; ソフトウェア切換え信号を読む
; 最初の測定入力時
N80 IF MTSIGNAL == 0 GOTO _FEHL1 ; 信号を評価する
N90 MEASVALUE_IN_X=$AA_MW[X] ; ワーク寸法に測定値を
; 読み込む
```

N95 M0

N100 M02

N110 \_FEHL1: MSG (「プローブが切換わらない」)

N120 M0

N130 M02

## 繰り返し精度

このプログラムによって、測定系全体（マシンプローブ信号を NC 伝達）の測定のパラツキ（繰り返し精度）を計算できます。

例では、X 軸で 10 個測定が行われ、測定値は、ワーク座標系で記録されます。

したがって、傾向とに左右されないいわゆる「ランダム」寸法ずれを測定することができます。

%\_N\_TEST\_GENAU\_MPF;

\$PATH=/\_N\_MPF\_DIR

N05 DEF INT SIGNAL, II ; 変数定義

N10 DEF REAL MEASVALUE\_IN\_X[10]

N15 G17 T1 D1 ; 初期条件,  
; プローブ用にツールオフセットを  
; 前もって選択する

N20 \_ANF: G0 X0 F150 ← ; 測定軸にアプローチ

N25 MEAS=+1 G1 X100 ← ; 切換え信号が  
; X 軸方向に変位する場合と  
; 変位しない場合の  
; 第 1 測定入力での測定

N30 STOPRE ← ; 次の結果の  
; デコードを停止する

N35 SIGNAL= \$AC\_MEA[1] ; 第 1 測定入力で  
; ソフトウェア切換え信号を読取る

N37 IF SIGNAL == 0 GOTO \_FEHL1 ; 切換え信号をチェックする

N40 MEASVALUE\_IN\_X[II]=\$AA\_MW[X] ; 測定値を  
; ワーク座標系に読込む

N50 II=II+1

N60 IF II<10 GOTOB \_ANF ; 10 回繰り返す

N65 M0

N70 M02

N80 \_FEHL1: MSG (" プローブが切換らない ")

N90 M0

N95 M02

パラメータ表示（ユーザで定義された変数）を選択後、プログラムがまだ処理中であれば、測定結果をフィールド MEASVALUE\_IN\_X[10] に読み込むことができ

ます。

## 2.8.7 データフィールド, リスト

番号	識別子	名称	参照
一般事項 (\$MN_ ... )			
13200	MEAS_PROBE_LOW_ACTIVE	プローブの特性の切換え	
チャンネル別 (\$MC_ ... )			
28264	MM_LEN_AC_FIFO	FIFO 変数 \$AC_FIFO ... の長さ	

---

---

## 2.9 ポジションスイッチ (N3)

### 2.9.1 概略説明

"Software cams" (ソフトウェアカム) ファンクションを使用すれば、(位置制御サイクルで) 位置依存カム信号を、NCK I/O および PLC に出力できます。

信号出力がセットされる位置の値の定義と変更は、設定データを介し行えます。

32 組のカムペアが使用できます。これらは、たとえば次の用途に使用できます。

- 油圧制御発振軸の反転信号として。
- リミットスイッチとして

(注)

ソフトウェアカムは、直線軸およびモジュロ回転軸に適用できます。

## 2.9.2 詳細説明

### ■ 一般事項 , アプリケーション

#### 一般事項

"Software cams" (ソフトウェアカム) ファンクション (2.9.3「補足条件」を参照) を使用すれば、位置の実際値を送る軸 (マシン軸) およびシミュレート軸のための位置依存切換え信号が発生します。カム信号は、NCK I/O および PLC に出力できます。

信号出力がセットされるカム位置の定義と変更は、設定データを介して行うことができます。設定データの読取り／書込みは、MMC, PLC およびパートプログラムを介して行えます。

#### 起動

"Software cams" (ソフトウェアカム) ファンクションは、どの運転モードでも起動して使用できます。ファンクションは、リセットまたは非常停止の時でもアクティブのままです。

#### アプリケーション

カム信号アプリケーションの中には、次のものがあります。

- 位置ゾーンの起動。
- 追加された移動を位置ファンクションとして開始すること。
- 油圧制御発振軸の反転信号として。

#### 軸タイプ

ソフトウェアカムは、マシン軸として定義された直線軸およびモジュロ回転軸に使用できます。

#### カムレンジ／カムペア

カムは、必ずペアの単位に組合わせて軸に割当てられます。1組のペアは、1つのプラスカムと1つのマイナスカムから構成されます。32組のカムペアが使用できます。

プラスおよびマイナスのカムは、それぞれ軸がカム位置に到達した時、特定の接近方向へ、定義された点 (カム位置) で作動させる機械カムをシミュレートします。

カムレンジは、次のように、プラスおよびマイナスのカムに割当てられます。

カムレンジプラス                      全ての位置  $\geq$  プラスカム

カムレンジマイナス                    全ての位置  $\leq$  マイナスカム

## ■ カム信号およびカム位置

### ■ カム信号の発生

#### 直線軸

カム信号の切換えエッジは、軸トラバース方向のファンクションとして発生されます。

- 正の軸方向へ軸がマイナスカムをトラバースした時、マイナスカム信号は、1 から 0 に切換わります。
- 正の方向へ軸がプラスカムをトラバースした時、プラスカム信号は、0 から 1 に切換わります。

両方のカム信号は、PLC および NCK I/O に出力させることができます。プラスおよびマイナスのカム信号の出力によって、プラスまたはマイナスのカムレンジ内に軸が入っているかどうかを個別に検出することが容易になります。

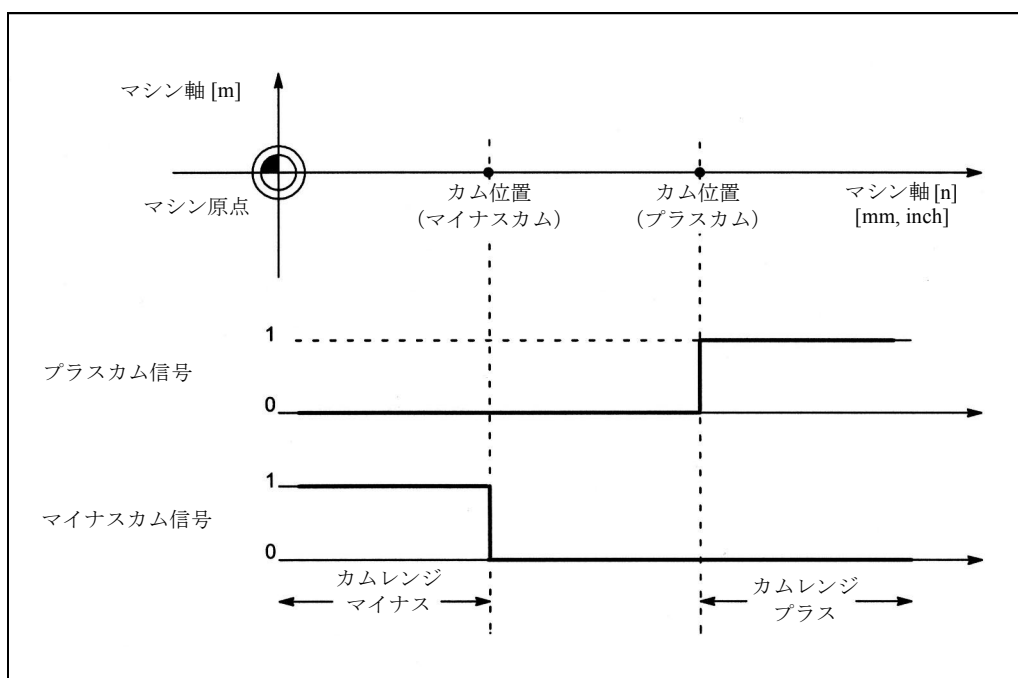


図 2.85 直線軸のソフトウェアカム（マイナスカム < プラスカム）

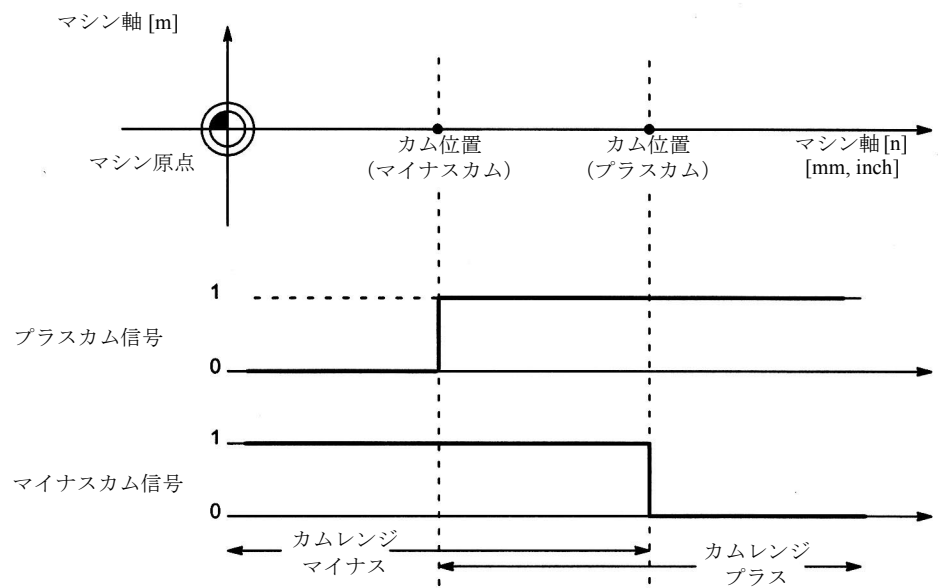


図 2.86 直線軸のソフトウェアカム（プラスカム＜マイナスカム）

## モジュロ回転軸

カム信号の切換えエッジは、回転軸トラバース方向のファンクションとして発生します。

- プラスカム信号は、正の軸方向へ軸がマイナスカムをトラバースすると 0 から 1 に切り替わり、プラスカムをトラバースすると 1 から 0 に復帰します。
- プラスカム信号の正のエッジのたびに、マイナスカム信号はレベルを変更します。

(注)

プラスカム応答は、次の条件の下で適用します。

プラスカム - マイナスカム < 180 度

この条件が満たされていない場合や、プラスカムより高い値にマイナスカムがセットされた場合には、プラスカム信号応答が反転させられます。マイナスカム信号応答は、変化しません。

PLC が確実に検出できないほど小さい値にカムレンジがセットされた場合でも、マイナスカムの信号変化によって、カムの移動を検出できます。

両方のカム信号が、PLC および NCK I/O に出力できます。プラスおよびマイナスのカム信号の出力によって、プラスまたはマイナスのカムレンジ内に軸が入っているかどうかを個別に検出することが容易になります。



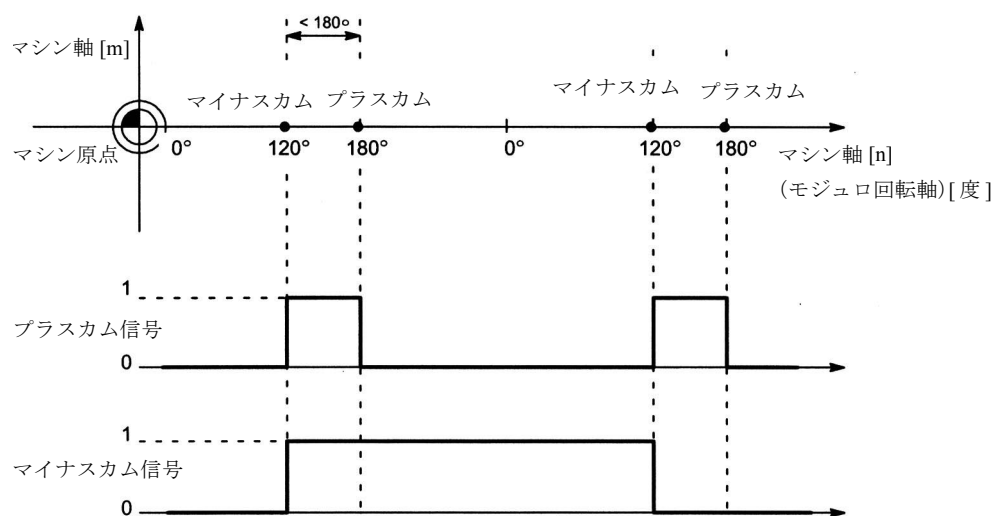


図 2.87 モジュロ回転軸のソフトウェアカム（プラスカム - マイナスカム < 180 度）

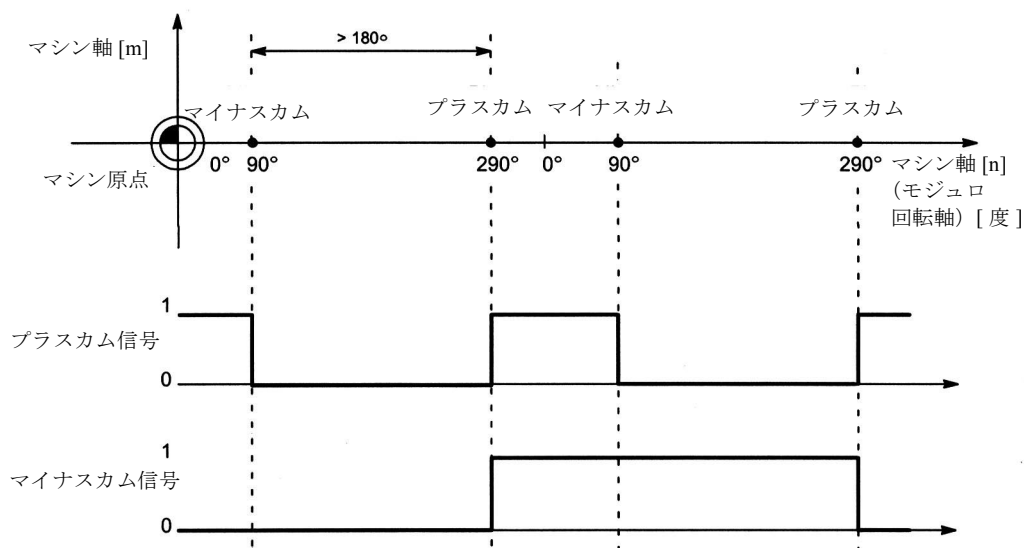


図 2.88 モジュロ回転軸のソフトウェアカム（プラスカム - マイナスカム > 180 度）

## ■ カム位置

### カム位置設定

プラスおよびマイナスのカムのカム位置は、次の一般設定データを介して定義されます。

SD 41500: SW\_CAM\_MINUS\_POS\_TAB\_1[n]      マイナスカム 1 ～ 8 の位置

SD 41501: SW\_CAM\_PLUS\_POS\_TAB\_1[n]      プラスカム 1 ～ 8 の位置

SD 41502: SW\_CAM\_MINUS\_POS\_TAB\_2[n]      マイナスカム 9 ～ 16 の位置

SD 41503: SW\_CAM\_PLUS\_POS\_TAB\_2[n]      プラスカム 9 ～ 16 の位置

更に、ソフトウェアバージョン 4.1 から：

SD 41504: SW\_CAM\_MINUS\_POS\_TAB\_3[n]      マイナスカム 17 ～ 24 の位置

SD 41505: SW\_CAM\_PLUS\_POS\_TAB\_3[n]      プラスカム 17 ～ 24 の位置

SD 41506: SW\_CAM\_MINUS\_POS\_TAB\_4[n]      マイナスカム 25 ～ 32 の位置

SD 41507: SW\_CAM\_PLUS\_POS\_TAB\_4[n]      プラスカム 25 ～ 32 の位置

(注)

カムペアのグループ分け（1つのグループに8組ずつ）によって、（たとえば、マシン関連カム位置およびワークピース関連カム位置ごとに異なった）アクセス認証レベルを割当てることができます。

位置は、マシン座標系に入力されます。最大トラバースレンジを基準にした確認は行われません。

### 測定系 メートル／インチ

ソフトウェアバージョン 5 および MD 10260: CONVERT\_SCALING\_SYSTEM=1 (/G2/ を参照) 以降から、カム位置は、セットされた基本系ではなく、MD 10270: POS\_TAB\_SCALING\_SYSTEM でセットされた測定系を指します。

MD 10270: POS\_TAB\_SCALING\_SYSTEM=0: メートル

MD 10270: POS\_TAB\_SCALING\_SYSTEM=1: インチ

従って、MD 10270 は、SD 41500 ～ SD 41507 の位置指定に対し、測定系を定義します。

G70/G71 または G700/G710 で切換えても無効になります。

### カム位置の検出

カム信号をセットするために、実際の軸位置がカム位置と照合されます。

### カム位置の書込み／読取り

MMC, PLC およびパートプログラムを介して書込み／読取りを実行するため、設定データにアクセスできます。

パートプログラムからのデータアクセスは、ブロック処理と同期させられません。プログラムされた前処理停止（STOPRE コマンド）によって同期させることができます。

PLC ユーザプログラムで、FB2 と FB3 でのカム位置の書込み／読取りを行うことができます。

## 軸／カム割当て

カムペアおよびマシン軸の間の割当ては、一般 MD: SW\_CAM\_ASSIGN\_TAB[n]  
(マシン軸へのソフトウェアカムの割当て) を介して行われます。

(注)

次の NCK 電源オンの後、軸割当てへの変更が有効になります。

軸が割当てられないカムペアは、アクティブとはなりません。

カムペアには、一度に 1 つのマシン軸しか割当てることができません。

1 つのマシン軸に、複数のカムペアを定義できます。

## ■ リード／遅延時間（動的カム）

### 時間

遅延を補償するため、追加処置によって、2 つのリードまたは遅延時間を、カム信号出力のマイナスとプラスのカムにそれぞれ割当てることができます。

2 つのリードまたは遅延時間は、マシンデータおよび設定データに入力されます。

(注)

負の時間の値の入力によって、カム信号出力が遅延します。

### マシンデータへの入力

第一のリードまたは遅延時間は、次の一般マシンデータに入力されます。

MD 10460: SW\_CAM\_MINUS\_LEAD\_TIME[n]

マイナスカムのリードまたは遅延時間

MD 10461: SW\_CAM\_PLUS\_LEAD\_TIME[n]

プラスカムのリードまたは遅延時間

たとえば、次のようなものが、これらのマシンデータに入力できます。

- （たとえば、オシロスコープによって判定される）実際値検出およびカム信号出力の間の一定の内部遅延時間。
- 一定の外部遅延時間

---

## 設定データへの入力

第二のリードまたは遅延時間は、次の一般設定データに入力されます。

SD 41520: SW\_CAM\_MINUS\_TIME\_TAB\_1[n]

マイナスカム 1 ～ 8 のリードまたは遅延時間

SD 41521: SW\_CAM\_PLUS\_TIME\_TAB\_1[n]

プラスカム 1 ～ 8 のリードまたは遅延時間

SD 41522: SW\_CAM\_MINUS\_TIME\_TAB\_2[n]

マイナスカム 9 ～ 16 のリードまたは遅延時間

SD 41523: SW\_CAM\_PLUS\_TIME\_TAB\_2[n]

プラスカム 9 ～ 16 のリードまたは遅延時間

SD 41524: SW\_CAM\_MINUS\_TIME\_TAB\_3[n]

マイナスカム 17 ～ 24 のリードまたは遅延時間

SD 41525: SW\_CAM\_PLUS\_TIME\_TAB\_3[n]

プラスカム 17 ～ 24 のリードまたは遅延時間

SD 41526: SW\_CAM\_MINUS\_TIME\_TAB\_4[n]

マイナスカム 25 ～ 32 のリードまたは遅延時間

SD 41527: SW\_CAM\_PLUS\_TIME\_TAB\_4[n]

プラスカム 25 ～ 32 のリードまたは遅延時間

たとえば、加工中に変化する恐れがある遅延時間は、これらの設定データに入力される必要があります。

## ■ カム信号の出力

カムステータス、即ちカム信号は、NCK I/O および PLC に出力できます。

### カム信号出力の起動

軸別 IS "Cam activation" (カム起動) (DB31 ～ 62, DBX2.0) は、軸のカム信号出力の起動に使用されます。

### PLC へのチェックバック信号

軸別 IS "Cams active" (カムアクティブ) (DB31 ～ 62, DBX62.0) は、PLC に送られて、軸のカムがすべて起動できたことを通知します。

(注)

PLC ユーザは、その他の条件（たとえば、軸基準化リセット有効）に起動をリンクすることもできます。

## ■ PLC へのカム信号の出力

ソフトウェアカムが起動した全マシン軸のカム信号のステータスは、PLC に出力されます。

ステータスは、IPO サイクルで出力されて PLC に非同期転送されます。

### マイナスカム信号

マイナスカム信号ステータスは、一般 IS "Minus cam signals 1 to 32" (マイナスカム信号 1 ～ 32) (DB10, DBX110.0 ～ 113.7) に入力されます。

### プラスカム信号

プラスカム信号ステータスは、一般 IS "Plus cam signals 1 to 32" (プラスカム信号 1 ～ 32) (DB10, DBX114.0 ～ 117.7) に入力されます。

(注)

... ならば	... が起こる
測定系が選択されていないか、または IS "Cam activation" (カム起動) (DB31 ～ 62, DBX2.0) = "0"	次の IS が "0" にセットされる。 - Minus cam signals 1-32 (マイナスカム信号 1 ～ 32) (DB10, DBX110.0 ～ 113.7) - Plus cam signals 1-32 (プラスカム信号 1 ～ 32) (DB10, DBX114.0 ～ 117.7) - Cam active (カムアクティブ) (DB31 ～ 62, DBX62.0)

## ■ NCK I/O へのカム信号の出力

カム信号は、位置制御サイクルで、NCK I/O に出力されます。

NCU の 4 つの実装出力および合計 32 の外部 NCK 出力（オプション）は、NCK I/O のデジタル出力として使用できます。

参照： 2.1 NCK ダイレクト入出力信号 (A4)

## ハードウェア割当て

ケースごとに、8 組のカムペアが、2 つの一般マシンデータに使用するハードウェアバイトに割当てられます。

MD 10470: SW\_CAM\_ASSIGN\_FASTOUT\_1

NCK I/O へのカム 1 ～ 8 の出力のハードウェア割当て

MD 10471: SW\_CAM\_ASSIGN\_FASTOUT\_2

NCK I/O へのカム 9 ～ 16 の出力のハードウェア割当て

MD 10472: SW\_CAM\_ASSIGN\_FASTOUT\_3

NCK I/O へのカム 17 ～ 24 の出力のハードウェア割当て

MD 10473: SW\_CAM\_ASSIGN\_FASTOUT\_4

NCK I/O へのカム 25 ～ 32 の出力のハードウェア割当て

(注)

各マシンデータにおいて、8 つのマイナスカム信号出力および 8 つのプラスカム信号出力に対し、1 HW バイトずつ定義できます。

更に、カム信号出力は、2 つのマシンデータで反転させることができます。

第二バイトが指定されない場合 (= "0") には、8 つのカムが、第一反転画面フォームで第一 HW バイトを介し、マイナスおよびプラスのカム信号の論理積演算として出力されます。

## パートプログラムでのステータス照会

HW 出力ステータスは、メインラン変数 \$A\_OUT[n]（n = 出力ビット番号）で、パートプログラムによって読み取ることができます。

## ■ 高精度のカム信号出力

### 切換え精度

カム信号は、位置制御サイクルで NCK I/O に出力されます。位置制御サイクルのタイムグリッドによって、カム信号の切換え精度は速度のファンクションとして制限されます。

この場合 :  $\Delta pos = V_{act} * \text{位置制御サイクル}$

### パラメータ

$\Delta pos$  : (位置制御サイクルによって調整される) 切換え精度

$V_{act}$  : 現在の軸速度

### 例

$V_{act} = 20 \text{ m/min}$  , 位置制御サイクル = 4 ms       $\Delta pos = 1.33 \text{ mm}$

$V_{act} = 2000 \text{ rev/min}$  , 位置制御サイクル = 2 ms       $\Delta pos = 24 \text{ 度}$

### タイマ制御出力

タイマ割込みを使用して、位置制御サイクルとは無関係にカム信号を出力すれば、かなり高い精度が得られます。

一般 MD : SW\_CAM\_TIMER\_FASTOUT\_MASK (タイマ割込みを介して NCU に出力されるカム信号の画面フォーム) で、4 つのカムペアに合わせて 4 つの NCU 実装出力が選択されます。

この場合、カムペアのマイナスおよびプラス信号は、1 つの信号として出力するために、排他的論理和演算を行います。

(注)

このファンクションは、次のマシンデータでセットした割当てから独立して働きます。

MD: SW\_CAM\_ASSIGN\_FASTOUT\_1 または

MD 10471: SW\_CAM\_ASSIGN\_FASTOUT\_2 または

MD 10472: SW\_CAM\_ASSIGN\_FASTOUT\_3 または

MD 10473: SW\_CAM\_ASSIGN\_FASTOUT\_4

実装バイトは、一度に 1 回しか使用できません。

### 制限事項

カムの相互の位置関係に対しては、次のように適用されます。

IPO の 1 サイクル当たり、タイマ制御方式で 1 つの信号が出力されます。IPO サイクルにおいて、2 組以上のカムペアの信号が変化した場合、信号は優先順に出力されます。

最低番号のカムペア (1 ... 32) は、全保留信号が出力される瞬間、即ち、同じ時にその他のカムペアの信号変化が発生する瞬間を判定します。

### 2.9.3 補足条件

#### ファンクション "Software cams, position switching signals" (ソフトウェアカム, 位置切換え信号) の可用性

このファンクションは、オプションであり、ソフトウェアバージョン 2 以降から有効です。

### 2.9.4 データの説明 (MD , SD)

#### ■ 一般マシンデータ

10450 MD 番号	SW_CAM_ASSIGN_TAB[n] マシン軸へのソフトウェアカムの割当て	
デフォルト値 : 0	最小入力リミット : 0	最大入力リミット : 8 または 31
変更が有効になるための条件 : 電源オン	保護レベル : 2/4	単位 : -
データタイプ : BYTE	適用開始 SW バージョン : 2.1 または 4.1	
意味 :	このマシンデータによって、(1 つのマイナスクムと 1 つのプラスカムから構成された) 16 とおりのカムペアに、マシン軸を 1 つずつ割当てることができる。 "0" が入力されると、該当するカムが処理されない。 カム信号出力は、軸 IS "Cam activation" (カム起動) (DB31 ~ 48, DBX2.0) を介して起動する。 マシンデータのインデックス [n] は、次のようにカムペアをアドレッシングする。n = 0, 1, ..., 31 は、カムペア 1, 2, ..., 32 に対応する。	
アプリケーション	カムペア 1 は、マシン軸 2 に割当てなければならない。カムペア 3 は、マシン軸 4 に割当てなければならない。 カムペア 2 は、どのマシン軸にも割当てられない。 -> MD: SW_CAM_ASSIGN_TAB[0]= 3 MD: SW_CAM_ASSIGN_TAB[1]= 0 MD: SW_CAM_ASSIGN_TAB[2]= 4	
関連性	IS "Cam activation" (カム起動) (DB31 ~ 48, DBX2.0)	



10460 MD 番号	SW_CAM_MINUS_LEAD_TIME[n] マイナスカム 1 ～ 16 のリードまたは遅延時間	
デフォルト値 : 0.0	最小入力リミット : ***	最大入力リミット : ***
変更が有効になるための条件 : 電源オン	保護レベル : 2/4	単位 : s
データタイプ : DOUBLE	適用開始 SW バージョン : 2.1	
意味 :	<p>リードまたは遅延時間は、遅延時間の補償のため、このマシンデータのマイナスカム 1 ～ 16 に割当てることができる。</p> <p>関連カム信号の切換えエッジは、入力した時間の値の分だけ進ませるか、または遅らせる。</p> <p>正の値 --&gt; リード時間</p> <p>負の値 --&gt; 遅延時間</p> <p>マシンデータのインデックス [n] が、次のようにカムペアをアドレッシングする。n = 0, 1, ..., 15 が、カムペア 1, 2, ..., 16 に対応する。</p> <p>このマシンデータが設定データに追加される :</p> <p>SW_CAM_MINUS_TIME_TAB_1[n] および SW_CAM_MINUS_TIME_TAB_2[n]</p>	
関連性	<p>SD: SW_CAM_MINUS_TIME_TAB_1[n] (マイナスカム 1 ～ 8 のリードまたは遅延時間)</p> <p>SD: SW_CAM_MINUS_TIME_TAB_2[n] (マイナスカム 9 ～ 16 のリードまたは遅延時間)</p>	

10461 MD 番号	SW_CAM_PLUS_LEAD_TIME[n] プラスカム 1 ～ 16 のリードまたは遅延時間	
デフォルト値 : 0.0	最小入力リミット : ***	最大入力リミット : ***
変更が有効になるための条件 : 電源オン	保護レベル : 2/4	単位 : s
データタイプ : DOUBLE	適用開始 SW バージョン : 2.1	
意味 :	<p>リードまたは遅延時間は、遅延時間の補償のため、このマシンデータでプラスカム 1 ～ 16 に割当てることができる。</p> <p>関連カム信号の切換えエッジは、入力した時間の値の分だけ進ませるか、または遅らせる。</p> <p>正の値 --&gt; リード時間</p> <p>負の値 --&gt; 遅延時間</p> <p>マシンデータのインデックス [n] が、次のようにカムペアをアドレッシングする。n = 0, 1, ..., 15 が、カムペア 1, 2, ..., 16 に対応する。</p> <p>このマシンデータが、次の設定データに追加される。SW_CAM_PLUS_TIME_TAB_1[n] および SW_CAM_PLUS_TIME_TAB_2[n]</p>	
関連性	<p>SD: SW_CAM_PLUS_TIME_TAB_1[n] (プラスカム 1 ～ 8 のリードまたは遅延時間)</p> <p>SD: SW_CAM_PLUS_TIME_TAB_2[n] (プラスカム 9 ～ 16 のリードまたは遅延時間)</p>	

10470 MD 番号	SW_CAM_ASSIGN_FASTOUT_1 NCK I/O へのカム 1 ～ 8 の出力のハードウェア割当て	
デフォルト値: 0	最小入力リミット: ***	最大入力リミット: ***
変更が有効になるための条件: 電源オン	保護レベル: 2/4	単位: 16 進値
データタイプ: DWORD	適用開始 SW バージョン: 2.1	
意味:	<p>カム信号ステータスは、PLC および NCK I/O に出力できる。</p> <p>カムペア 1 ～ 8 に使用する、デジタル出力バイトへのマイナスおよびプラスカム信号のハードウェア割当ては、このマシンデータで実行できる。</p> <p>更に、割当て出力信号は、このマシンデータで反転させることができる。</p> <p>MD は、次のように符号化される。</p> <p>ビット 0 ～ 7: デジタル出力で使用する第一 HW バイト番号。</p> <p>ビット 8 ～ 15: デジタル出力で使用する第二 HW バイト番号。</p> <p>ビット 16 ～ 23: 使用する第一 HW バイトの書込みの反転画面フォーム</p> <p>ビット 24 ～ 31: 使用する第二 HW バイトの書込みの反転画面フォーム</p> <p>ビット =0: 反転しない。</p> <p>ビット =1: 反転する。</p> <p>両方の HW バイトが指定された場合、第一バイトにはマイナスカム信号が含まれ、第二バイトにはプラスカム信号が含まれている。</p> <p>第二バイトが指定されていない場合 (= "0") には、8 つのカムが、第一反転画面フォームで第一 HW バイトを介し、マイナスおよびプラスカム信号の論理積演算として出力される。「プラスカム - マイナスカム &lt; 180 度」での、直線軸および回転軸の非反転出力信号のステータスは、次のとおりである。</p> <p>"1": マイナスカムおよびプラスカムの間で。</p> <p>"0": このレンジ外で。</p> <p>「プラスカム - マイナスカム =&gt; 180 度」での、回転軸の非反転出力信号のステータスは次のとおりである。</p> <p>"1": マイナスカムおよびプラスカムの間で。</p> <p>"0": このレンジ外で。</p> <p>デジタル出力のバイトアドレスは、次のように指定しなければならない。</p> <p>1: 実装バイト</p> <p>2 ～ 5: 外部バイト</p>	
アプリケーション:	<p>マイナスカム信号は、実装バイトを介して出力される必要がある。</p> <p>プラスカム信号は、NCU 端子台のバイトアドレス 3 を介して出力される必要がある。</p> <p>次の信号は、反転させなければならない。</p> <p>マイナスカム信号 2, 4, 5</p> <p>(第一 HW バイトのビット 1, 3, 4 に対応する)</p> <p>プラスカム信号 1, 3, 4</p> <p>(第二 HW バイトのビット 0, 2, 3 に対応する) -</p> <p>-&gt; MD: SW_CAM_ASSIGN_FASTOUT_1='H0D1A0301'</p>	

10471 MD 番号	SW_CAM_ASSIGN_FASTOUT_2 NCK I/O へのカム 9 ～ 16 の出力のハードウェア割当て	
デフォルト値 : 0	最小入力リミット : ***	最大入力リミット : ***
変更が有効になるための条件 : 電源オン	保護レベル : 2/4	単位 : 16 進値
データタイプ : DWORD	適用開始 SW バージョン : 2.1	
意味 :	<p>カム信号ステータスは, PLC および NCK I/O に出力できる。</p> <p>カムペア 9 ～ 16 に使用する, デジタル出力バイトへのマイナスおよびプラスカム信号のハードウェア割当ては, このマシンデータで実行できる。</p> <p>更に, 割当て出力信号は, このマシンデータで反転させることができる。</p> <p>MD は, 次のように符号化される。</p> <p>ビット 0 ～ 7: デジタル出力で使用する第一 HW バイト番号。</p> <p>ビット 8 ～ 15: デジタル出力で使用する第二 HW バイト番号。</p> <p>ビット 16 ～ 23: 使用する第一 HW バイトの書込みの反転画面フォーム。</p> <p>ビット 24 ～ 31: 使用する第二 HW バイトの書込みの反転画面フォーム。</p> <p>ビット = 0: 反転しない。</p> <p>ビット = 1: 反転する。</p> <p>両方の HW バイトが指定された場合, 第一バイトにはマイナスカム信号が含まれ, 第二バイトにはプラスカム信号が含まれている。</p> <p>第二バイトが指定されていない場合 (= "0") には, 8 つのカムが, 第一反転画面フォームで第一 HW バイトを介し, マイナスおよびプラスカム信号の論理積演算として出力される。「プラスカム - マイナスカム &lt; 180 度」での, 直線軸および回転軸の非反転出力信号のステータスは, 次のとおりである。</p> <p>"1": マイナスカムおよびプラスカムの間。</p> <p>"0": このレンジ外。</p> <p>「プラスカム - マイナスカム =&gt; 180 度」での, 回転軸の非反転出力信号のステータスは, 次のとおりである。</p> <p>"1": マイナスカムおよびプラスカムの間。</p> <p>"0": このレンジ外。</p> <p>デジタル出力のバイトアドレスとして, 次のように指定しなければならない。</p> <p>1: 実装バイト</p> <p>2 ～ 5: 外部バイト</p>	
アプリケーション :	SW_CAM_ASSIGN_FASTOUT_1 を参照。	

10472 MD 番号	SW_CAM_ASSIGN_FASTOUT_3 NCK I/O へのカム 17 ~ 24 の出力のハードウェア割当て	
デフォルト値 : 0	最小入力リミット : ***	最大入力リミット : ***
変更が有効になるための条件 : 電源オン	保護レベル : 2/7	単位 : 16 進値
データタイプ : DWORD	適用開始 SW バージョン : 4.1	
意味 :	<p>カム信号ステータスは、PLC および NCK I/O に出力できる。</p> <p>カムペア 17 ~ 24 に使用する、デジタル出力バイトへのマイナスおよびプラスカム信号のハードウェア割当ては、このマシンデータで実行できる。</p> <p>更に、割当て出力信号は、このマシンデータで反転させることができる。</p> <p>MD は、次のように符号化される。</p> <p>ビット 0 ~ 7 : デジタル出力で使用する第一 HW バイト番号。</p> <p>ビット 8 ~ 15 : デジタル出力で使用する第二 HW バイト番号。</p> <p>ビット 16 ~ 23 : 使用する第一 HW バイトの書込みの反転画面フォーム。</p> <p>ビット 24 ~ 31 : 使用する第二 HW バイトの書込みの反転画面フォーム。</p> <p>ビット = 0 : 反転しない。</p> <p>ビット = 1 : 反転する。</p> <p>両方の HW バイトが指定された場合、第一バイトにはマイナスカム信号が含まれ、第二バイトにはプラスカム信号が含まれている。</p> <p>第二バイトが指定されていない場合 (= "0") には、8 つのカムが、第一反転画面フォームで第一 HW バイトを介し、マイナスおよびプラスカム信号の論理積演算として出力される。「プラスカム - マイナスカム &lt; 180 度」での、直線軸および回転軸の非反転出力信号のステータスは、次のとおりである。</p> <p>"1": マイナスカムおよびプラスカムの間。</p> <p>"0": このレンジ外。</p> <p>「プラスカム - マイナスカム =&gt; 180 度」での、回転軸の非反転出力信号のステータスは、次のとおりである。</p> <p>"1": マイナスカムおよびプラスカムの間で。</p> <p>"0": このレンジ外で。</p> <p>デジタル出力のバイトアドレスとして、次のように指定しなければならない。</p> <p>1 : 実装バイト 2 ~ 5 : 外部バイト</p>	
アプリケーション :	SW_CAM_ASSIGN_FASTOUT_1 を参照。	

10473 MD 番号	SW_CAM_ASSIGN_FASTOUT_4 NCK I/O へのカム 25 ~ 32 の出力のハードウェア割当て	
デフォルト値 : 0	最小入力リミット : ***	最大入力リミット : ***
変更が有効になるための条件 : 電源オン	保護レベル : 2/7	単位 : 16 進値
データタイプ : DWORD	適用開始 SW バージョン : 4.1	
意味 :	<p>カム信号ステータスは, PLC および NCK I/O に出力できる。</p> <p>カムペア 25 ~ 32 に使用する, デジタル出力バイトへのマイナスおよびプラスカム信号のハードウェア割当ては, このマシンデータで実行できる。</p> <p>更に, 割当て出力信号は, このマシンデータで反転させることができる。</p> <p>MD は, 次のように符号化される。</p> <p>ビット 0 ~ 7 : デジタル出力で使用する第一 HW バイト番号。</p> <p>ビット 8 ~ 15 : デジタル出力で使用する第二 HW バイト番号。</p> <p>ビット 16 ~ 23 : 使用する第一 HW バイトの書込みの反転画面フォーム。</p> <p>ビット 24 ~ 31 : 使用する第二 HW バイトの書込みの反転画面フォーム。</p> <p>ビット = 0 : 反転しない。</p> <p>ビット = 1 : 反転する。</p> <p>両方の HW バイトが指定された場合, 第一バイトにはマイナスカム信号が含まれ, 第二バイトにはプラスカム信号が含まれている。</p> <p>第二バイトが指定されていない場合 (= "0") には, 8 つのカムが, 第一反転画面フォームで第一 HW バイトを介し, マイナスおよびプラスカム信号の論理積演算として出力される。「プラスカム - マイナスカム &lt; 180 度」での, 直線軸および回転軸の非反転出力信号のステータスは, 次のとおりである。</p> <p>"1": マイナスカムおよびプラスカムの間。</p> <p>"0": このレンジ外。</p> <p>「プラスカム - マイナスカム =&gt; 180 度」での, 回転軸の非反転出力信号のステータスは, 次のとおりである。</p> <p>"0": マイナスカムおよびプラスカムの間。</p> <p>"1": このレンジ外。</p> <p>デジタル出力のバイトアドレスとして, 次のように指定しなければならない。</p> <p>1 : 実装バイト</p> <p>2 ~ 5 : 外部バイト</p>	
アプリケーション	MD: SW_CAM_ASSIGN_FASTOUT_1 を参照。	

10480 MD 番号	SW_CAM_TIMER_FASTOUT_MASK タイマ割込みを介して NCU に出力されるカム信号の画面フォーム	
デフォルト値 : 0	最小入力リミット : ***	最大入力リミット : ***
変更が有効になるための条件 : 電源オン	保護レベル : 2/4	単位 : 16 進値
データタイプ : DWORD	適用開始 SW バージョン : 2.1	
意味 :	<p>NCK I/O の 4 つの実装出力へのタイマ制御出力は、このマシンデータで、4 組のカムペアに合わせて選択できる。</p> <p>この場合、カムペアのマイナスおよびプラス信号は 1 つの信号として出力するため、排他的論理和演算を行う。</p> <p>セットビットの意味 : 関連カム（マイナスおよびプラスカム信号の排他的論理和）は、NCU の 4 つの実装出力の 1 つで、タイマ割込みを介して出力される。</p> <p>この場合、カムペア信号は、実装出力 1 ～ 4 に昇順で割当てられる。この機能は、次のマシンデータにセットされた割当てから独立して働く。 MD: SW_CAM_ASSIGN_FASTOUT_1 または MD: SW_CAM_ASSIGN_FASTOUT_2</p> <p>注 : 実装バイトは、一度に 1 回しか使用できない。MD で指定されたカムペアで、IPO サイクルにおいて複数回の信号変化が生じた場合、最低番号のカムペアは、出力の瞬間を判定する。同時に、その他の信号変化も生じる。</p>	
アプリケーション	<p>カムペア 2, 5 および 7 の信号は、次のように、タイマ制御方式で出力される必要がある。</p> <p>--&gt; MD: SW_CAM_TIMER_FASTOUT_MASK= 'H52'</p> <p>--&gt; NCK の実装出力 1 へのカムペア 2 の信号 NCK の実装出力 2 へのカムペア 5 の信号 NCK の実装出力 3 へのカムペア 7 の信号</p>	

## ■ 一般設定データ

41500 SD 番号	SW_CAM_MINUS_POS_TAB_1[n] マイナスカム 1 ～ 8 の位置	
デフォルト値 : 0	最小入力リミット : ***	最大入力リミット : ***
変更が有効になるための条件 : 即時	保護レベル : 7/7	単位 : mm, 度
データタイプ : DOUBLE	適用開始 SW バージョン : 2.1	
意味 :	マイナスカム 1 ～ 8 のカム位置は、この設定データに入力される。 位置は、マシン座標系に入力される。 直線軸およびモジュロ回転軸がカム位置をオーバトラベルした時の応答は、 2.9.2 「■カム信号の発生」に説明されている。 設定データのインデックス [n] が、次のようにカムペアをアドレッシングする。 n = 0, 1, ..., 7 は、カムペア 1, 2, ..., 8 に対応する。	

41501 SD 番号	SW_CAM_PLUS_POS_TAB_1[n] プラスカム 1 ～ 8 の位置	
デフォルト値 : 0	最小入力リミット : ***	最大入力リミット : ***
変更が有効になるための条件 : 即時	保護レベル : 7/7	単位 : mm, 度
データタイプ : DOUBLE	適用開始 SW バージョン : 2.1	
意味 :	プラスカム 1 ～ 8 のカム位置は、この設定データに入力される。 位置は、マシン座標系に入力される。 直線軸およびモジュロ回転軸がカム位置をオーバトラベルした時の応答は、 2.9.2 「■カム信号の発生」で説明している。 設定データのインデックス [n] が、次のようにカムペアをアドレッシングする。 n = 0, 1, ..., 7 は、カムペア 1, 2, ..., 8 に対応する。	

41502 SD 番号	SW_CAM_MINUS_POS_TAB_2[n] マイナスカム 9 ～ 16 の位置	
デフォルト値 : 0	最小入力リミット : ***	最大入力リミット : ***
変更が有効になるための条件 : 即時	保護レベル : 7/7	単位 : mm, 度
データタイプ : DOUBLE	適用開始 SW バージョン : 2.1	
意味 :	マイナスカム 9 ～ 16 のカム位置は、この設定データに入力される。 位置は、マシン座標系に入力される。 直線軸およびモジュロ回転軸がカム軸をオーバトラベルした時の応答は、 2.9.2 「■カム信号の発生」で説明している。 設定データのインデックス [n] が、次のようにカムペアをアドレッシングする。 n = 8, 9, ..., 15 は、カムペア 9, 10, ..., 16 に対応する。	

41503 SD 番号	SW_CAM_PLUS_POS_TAB_2[n] プラスカム 9 ～ 16 の位置	
デフォルト値 : 0	最小入力リミット : ***	最大入力リミット : ***
変更が有効になるための条件 : 即時	保護レベル : 7/7	単位 : mm, 度
データタイプ : DOUBLE	適用開始 SW バージョン : 2.1	
意味 :	<p>プラスカム 9 ～ 16 のカム位置は、この設定データに入力される。 位置は、マシン座標系に入力される。 直線軸およびモジュール回転軸がカム位置をオーバトラベルした時の応答は、 2.9.2 「■カム信号の発生」で説明している。 設定データのインデックス [n] が、次のようにカムペアをアドレッシングする。 n = 8, 9, ..., 15 は、カムペア 9, 10, ..., 16 に対応する。</p>	

41504 SD 番号	SW_CAM_MUNUS_POS_TAB_3[n] マイナスカム 17 ～ 24 の位置	
デフォルト値 : 0	最小入力リミット : ***	最大入力リミット : ***
変更が有効になるための条件 : 即時	保護レベル : 7/7	単位 : mm, 度
データタイプ : DOUBLE	適用開始 SW バージョン : 4.1	
意味 :	<p>マイナスカム 17 ～ 24 のカム位置は、この設定データに入力される。 位置は、マシン座標系に入力される。 直線軸およびモジュール回転軸がカム位置をオーバトラベルした時の応答は、 2.9.2 「■カム信号の発生」で説明している。 設定データのインデックス [n] が、次のようにカムペアをアドレッシングする。 n = 0, 1, ..., 7 は、カムペア 17, 18, ..., 24 に対応する。</p>	

41505 SD 番号	SW_CAM_PLUS_POS_TAB_3[n] プラスカム 17 ～ 24 の位置	
デフォルト値 : 0	最小入力リミット : ***	最大入力リミット : ***
変更が有効になるための条件 : 即時	保護レベル : 7/7	単位 : mm, 度
データタイプ : DOUBLE	適用開始 SW バージョン : 4.1	
意味 :	<p>プラスカム 17 ～ 24 のカム位置は、この設定データに入力される。 位置は、マシン座標系に入力される。直線軸およびモジュール回転軸がカム軸をオーバトラベルした時の応答は、2.9.2 「■カム信号の発生」に説明されている。 設定データのインデックス [n] が、次のようにカムペアをアドレッシングする。 n = 0, 1, ..., 7 は、カムペア 17, 18, ..., 24 に対応する。</p>	



41506 SD 番号	SW_CAM_MINUS_POS_TAB_4[n] マイナスカム 25 ～ 32 の位置	
デフォルト値 : 0	最小入力リミット : ***	最大入力リミット : ***
変更が有効になるための条件 : 即時	保護レベル : 7/7	単位 : mm, 度
データタイプ : DOUBLE	適用開始 SW バージョン : 4.1	
意味 :	<p>プラスカム 25 ～ 32 のカム位置は、この設定データに入力される。</p> <p>位置は、マシン座標系に入力される。</p> <p>直線軸およびモジュロ回転軸がカム軸をオーバトラベルした時の応答は、@ 2.9.2 「<b>■カム信号の発生</b>」で説明している。</p> <p>設定データのインデックス [n] が、次のようにカムペアをアドレッシングする。</p> <p>n = 8, 9, ..., 15 は、カムペア 25, 26, ..., 32 に対応する。</p>	

41507 SD 番号	SW_CAM_PLUS_POS_TAB_4[n] プラスカム 25 ～ 32 の位置	
デフォルト値 : 0	最小入力リミット : ***	最大入力リミット : ***
変更が有効になるための条件 : 即時	保護レベル : 7/7	単位 : mm, 度
データタイプ : DOUBLE	適用開始 SW バージョン : 4.1	
意味 :	<p>プラスカム 25 ～ 32 のカム位置は、この設定データに入力される。</p> <p>位置は、マシン座標系に入力される。</p> <p>直線軸およびモジュロ回転軸がカム軸をオーバトラベルした時の応答は、2.9.2 「<b>■カム信号の発生</b>」で説明している。</p> <p>設定データのインデックス [n] が、次のようにカムペアをアドレッシングする。</p> <p>n = 8, 9, ..., 15 は、カムペア 25, 26, ..., 32 に対応する。</p>	

41520 SD 番号	SW_CAM_MINUS_TIME_TAB_1[n] マイナスカム 1 ～ 8 のリードまたは遅延時間	
デフォルト値 : 0	最小入力リミット : ***	最大入力リミット : ***
変更が有効になるための条件 : 即時。	保護レベル : 7/7	単位 : s
データタイプ : DOUBLE	適用開始 SW バージョン : 2.1	
意味 :	<p>リードまたは遅延時間は、遅延時間の補償のため、この設定データでマイナスカム 1 ～ 8 に割当てることができる。</p> <p>関連カム信号の切換えエッジは、入力した時間の値の分だけ進ませるか、または遅らせる。</p> <p>正の値 : リード時間</p> <p>負の値 : 遅延時間</p> <p>設定データのインデックス [n] が、次のようにカムペアをアドレッシングする。</p> <p>n = 0, 1, ..., 7 は、カムペア 1, 2, ..., 8 に対応する。</p> <p>この設定データは、MD: SW_CAM_MINUS_LEAD_TIME[n] に追加される。</p>	
関連性	MD: SW_CAM_MINUS_LEAD_TIME[n] (マイナスカム 1 ～ 16 のリードまたは遅延時間)	

41521 SD 番号	SW_CAM_PLUS_TIME_TAB_1[n] プラスカム 1 ～ 8 のリードまたは遅延時間	
デフォルト値: 0	最小入力リミット: ***	最大入力リミット: ***
変更が有効になるための条件: 即時。	保護レベル: 7/7	単位: s
データタイプ: DOUBLE	適用開始 SW バージョン: 2.1	
意味:	<p>リードまたは遅延時間は、リードまたは遅延時間の補償のため、この設定データでプラスカム 1 ～ 8 に割当てることができる。</p> <p>関連カム信号の切換えエッジは、入力した時間の値の分だけ進ませるか、または遅らせる。</p> <p>正の値: リード時間</p> <p>負の値: 遅延時間</p> <p>設定データのインデックス [n] が、次のようにカムペアをアドレッシングする。</p> <p>n = 0, 1, ..., 7 が、カムペア 1, 2, ..., 8 に対応する。</p> <p>この設定データが、MD: SW_CAM_PLUS_LEAD_TIME[n] に追加される。</p>	
関連性	MD: SW_CAM_PLUS_LEAD_TIME[n] (プラスカム 1 ～ 16 のリードまたは遅延時間)	

41522 SD 番号	SW_CAM_MINUS_TIME_TAB_2[n] マイナスカム 9 ～ 16 のリードまたは遅延時間	
デフォルト値: 0	最小入力リミット: ***	最大入力リミット: ***
変更が有効になるための条件: 即時	保護レベル: 7/7	単位: s
データタイプ: DOUBLE	適用開始 SW バージョン: 2.1	
意味:	<p>リードまたは遅延時間は、遅延時間の補償のため、この設定データでマイナスカム 9 ～ 16 に割当てることができる。</p> <p>関連カム信号の切換えエッジは、入力した時間の値の分だけ進ませるか、または遅らせる。</p> <p>正の値: リード時間</p> <p>負の値: 遅延時間</p> <p>設定データのインデックス [n] が、次のようにカムペアをアドレッシングする。</p> <p>n = 8, 9, ..., 15 が、カムペア 9, 10, ..., 16 に対応する。</p> <p>この設定データが、MD: SW_CAM_MINUS_LEAD_TIME[n+8] に追加される。</p>	
関連性	MD: SW_CAM_MINUS_LEAD_TIME[n] (マイナスカム 1 ～ 16 のリードまたは遅延時間)	

41523 SD 番号	SW_CAM_PLUS_TIME_TAB_2[n] プラスカム 9 ～ 16 のリードまたは遅延時間	
デフォルト値 : 0	最小入力リミット : ***	最大入力リミット : ***
変更が有効になるための条件 : 即時。	保護レベル : 7/7	単位 : s
データタイプ : DOUBLE	適用開始 SW バージョン : 2.1	
意味 :	<p>リードまたは遅延時間は、遅延時間の補償のため、この設定データでプラスカム 9 ～ 16 に割当てることができる。</p> <p>関連カム信号の切換えエッジは、入力した時間の値の分だけ進ませるか、または遅らせる。</p> <p>正の値 : リード時間</p> <p>負の値 : 遅延時間</p> <p>設定データのインデックス [n] が、次のようにカムペアをアドレッシングする。</p> <p>n = 8, 9, ..., 15 が、カムペア 9, 10, ..., 16 に対応する。</p> <p>この設定データが、MD: SW_CAM_PLUS_LEAD_TIME[n+8] に追加される。</p>	
関連性	MD: SW_CAM_PLUS_LEAD_TIME[n] (プラスカム 1 ～ 16 のリードまたは遅延時間)	

41524 SD 番号	SW_CAM_MINUS_TIME_TAB_3[n] マイナスカム 17 ～ 24 のリードまたは遅延時間	
デフォルト値 : 0	最小入力リミット : ***	最大入力リミット : ***
変更が有効になるための条件 : 即時	保護レベル : 7/7	単位 : s
データタイプ : DOUBLE	適用開始 SW バージョン : 4.1	
意味 :	<p>リードまたは遅延時間は、遅延時間の補償のため、この設定データでマイナスカム 1 ～ 8 に割当てることができる。</p> <p>関連カム信号の切換えエッジは、入力した時間の値の分だけ進ませるか、または遅らせる。</p> <p>正の値 : リード時間</p> <p>負の値 : 遅延時間</p> <p>設定データのインデックス [n] が、次のようにカムペアをアドレッシングする。</p> <p>n = 0, 1, ..., 7 が、カムペア 17, 18, ..., 24 に対応する。</p> <p>この設定データが、MD: SW_CAM_MUNUS_LEAD_TIME[n] に追加される。</p>	
関連性	MD: SW_CAM_MINUS_LEAD_TIME[n] (マイナスカム 1 ～ 16 のリードまたは遅延時間)	

41525 SD 番号	SW_CAM_PLUS_TIME_TAB_3[n] プラスカム 17 ~ 24 のリードまたは遅延時間	
デフォルト値 : 0	最小入力リミット : ***	最大入力リミット : ***
変更が有効になるための条件 : 即時。	保護レベル : 7/7	単位 : s
データタイプ : DOUBLE	適用開始 SW バージョン : 4.1	
意味 :	<p>リードまたは遅延時間は、遅延時間の補償のため、この設定データでプラスカム 17 ~ 24 に割当てることができる。</p> <p>関連カム信号の切換えエッジは、入力した時間の値の分だけ進ませるか、または遅らせる。</p> <p>正の値 : リード時間</p> <p>負の値 : 遅延時間</p> <p>設定データのインデックス [n] が、次のようにカムペアをアドレッシングする。 n = 0, 1, ..., 7 が、カムペア 17, 18, ..., 24 に対応する。</p> <p>この設定データが、MD: SW_CAM_PLUS_LEAD_TIME[n] に追加される。</p>	
関連性	MD: SW_CAM_PLUS_LEAD_TIME[n] (プラスカム 1 ~ 16 のリードまたは遅延時間)	

41526 SD 番号	SW_CAM_MINUS_TIME_TAB_4[n] マイナスカム 25 ~ 32 のリードまたは遅延時間	
デフォルト値 : 0	最小入力リミット : ***	最大入力リミット : ***
変更が有効になるための条件 : 即時	保護レベル : 7/7	単位 : s
データタイプ : DOUBLE	適用開始 SW バージョン : 4.1	
意味 :	<p>リードまたは遅延時間は、遅延時間の補償のため、この設定データでマイナスカム 9 ~ 16 に割当てることができる。</p> <p>関連カム信号の切換えエッジは、入力した時間の値の分だけ進ませるか、または遅らせる。</p> <p>正の値 : リード時間</p> <p>負の値 : 遅延時間</p> <p>設定データのインデックス [n] が、次のようにカムペアをアドレッシングする。 n = 8, 9, ..., 15 が、カムペア 25, 26, ..., 32 に対応する。</p> <p>この設定データが、MD: SW_CAM_MUNUS_LEAD_TIME[n+8] に追加される。</p>	
関連性	MD: SW_CAM_MINUS_LEAD_TIME[n] (マイナスカム 1 ~ 16 のリードまたは遅延時間)	

41527 SD 番号	SW_CAM_PLUS_TIME_TAB_4[n] プラスカム 25 ～ 32 のリードまたは遅延時間	
デフォルト値 : 0	最小入力リミット : ***	最大入力リミット : ***
変更が有効になるための条件 : 即時	保護レベル : 7/7	単位 : s
データタイプ : DOUBLE	適用開始 SW バージョン : 4.1	
意味 :	<p>リードまたは遅延時間は、リードまたは遅延時間の補償のため、この設定データでプラスカム 25 ～ 32 に割当てることができる。</p> <p>関連カム信号の切換えエッジは、入力した時間の値の分だけ進ませるか、または遅らせる。</p> <p>正の値 : リード時間</p> <p>負の値 : 遅延時間</p> <p>設定データのインデックス [n] が、次のようにカムペアをアドレッシングする。</p> <p>n = 8, 9, ..., 15 が、カムペア 25, 26, ..., 32 に対応する。</p> <p>この設定データが、MD: SW_CAM_PLUS_LEAD_TIME[n+8] に追加される。</p>	
関連性	MD: SW_CAM_PLUS_LEAD_TIME[n] (プラスカム 1 ～ 16 のリードまたは遅延時間)	

## 2.9.5 信号の説明

### ■ 信号の概要

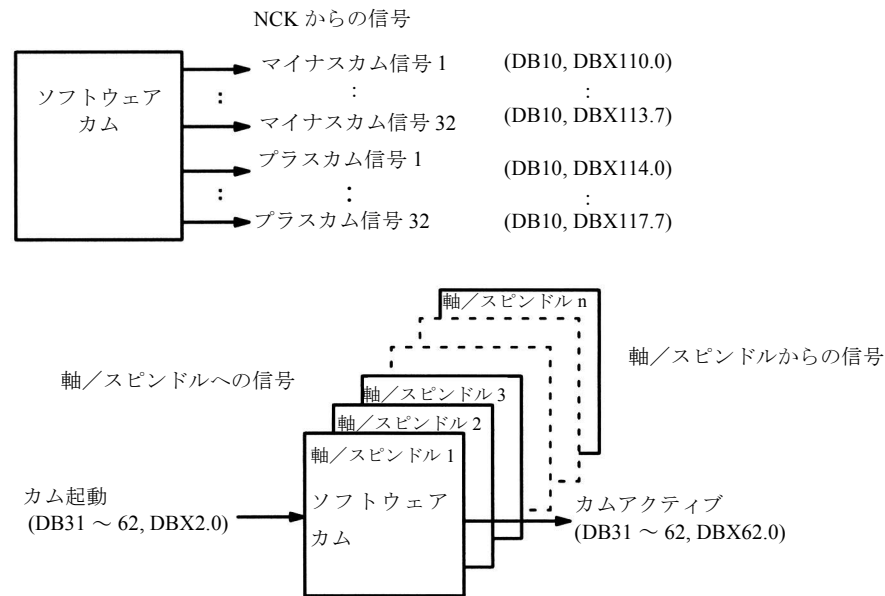


図 2.89 ソフトウェアカム，位置切換え信号の PLC インタフェース信号

## ■ 一般信号

### ■ NCK からの信号

DB10 DBX110.0 ~ 113.7 データブロック	Minus cam signals 1-32 (マイナスカム信号 1 ~ 32) NCK からの信号 (NCK +/- PLC)	
エッジ評価: なし	信号更新: 周期的	信号が有効となる最初の SW バージョン: 2.1
信号ステータス 1 あるいは信号遷移 0 ---> 1	マイナスカム信号 1 ~ 32 の切換えエッジは, (回転) 軸トラバース方向のファンクションとして発生し, IPO サイクルで PLC インタフェースに転送される。 直線軸: - 負の軸方向に軸がマイナスカムをオーバトラベルすると, マイナスカム信号は 0 から 1 に切換わる。 モジュール回転軸: - プラスカム信号の正のエッジのたびに, マイナスカム信号はレベルを変更する。	
信号ステータス 0 あるいは信号遷移 1 ---> 0	直線軸: - 正の軸方向に軸がマイナスカムをオーバトラベルすると, マイナスカム信号は 1 から 0 に切換わる。 モジュール回転軸: - プラスカム信号の正のエッジのたびに, マイナスカム信号はレベルを変更する。	

DB10 DBX114.0-117.7 データブロック	Plus cam signals 1-32 (プラスカム信号 1 ~ 32) NCK からの信号 (NCK +/- PLC)	
エッジ評価: なし	信号更新: 周期的	信号が有効となる最初の SW バージョン: 2.1
信号ステータス 1 あるいは信号遷移 0 ---> 1	プラスカム信号 1 ~ 32 の切換えエッジは, (回転) 軸トラバース方向のファンクションとして発生し, IPO サイクルで PLC インタフェースに転送される。 直線軸: - 正の軸方向に軸がプラスカムをオーバトラベルすると, プラスカム信号は 0 から 1 に切換わる。 モジュール回転軸: - 正の軸方向にマイナスカムをオーバトラベルすると, プラスカム信号は 0 から 1 に切換わる。 上記のプラスカム応答は, 次の条件の下で適用される。 プラスカム - マイナスカム < 180 度この条件が満たされていない場合や, プラスカムより高い値にマイナスカムがセットされた場合には, プラスカム信号応答が反転される。マイナスカム信号応答は, 変化しない。	
信号ステータス 0 または信号遷移 1 ---> 0	直線軸: - 負の方向に軸がプラスカムをオーバトラベルすると, プラスカム信号は 1 から 0 に切換わる。 モジュール回転軸: - 正の軸方向にプラスカムをオーバトラベルすると, プラスカム信号は 1 から 0 に復帰する。 上記のプラスカム応答は, 次の条件の下で適用される。 プラスカム - マイナスカム < 180 度この条件が満たされていない場合や, プラスカムより高い値にマイナスカムがセットされた場合には, プラスカム信号応答が反転される。マイナスカム信号応答は, 変化しない。	

## ■ 軸／スピンドル別信号

### ■ 軸／スピンドルへの信号

DB 31-62 DBX2.0 データブロック	カム起動 軸／スピンドルへの信号 (PLC +/- NCK)	
エッジ評価：なし	信号更新：周期的	信号が有効となる最初の SW バージョン：2.1
信号ステータス 1 あるいは信号遷移 0 ---> 1	一般 PLC インタフェースへの、 マイナスおよびプラスの軸カム信号の出力が起動する。NCK で IS "Cam activation"（カム起動）の処理の直後に、 起動が有効になる。	
信号ステータス 0 あるいは信号遷移 1 ---> 0	マイナスおよびプラスの軸カム信号の出力は、 一般 PLC インタフェースに出力されない。	
関連性	IS "Minus cam signal 1-32" (マイナスカム信号 1 ～ 32) (DB10, DBX110.0 - 113.7) IS "Plus cam signal 1-32" (プラスカム信号 1 ～ 32) (DB10, DBX114.0 - 117.7)	

### ■ 軸／スピンドルからの信号

DB 31-62 DBX62.0 データブロック	カムアクティブ 軸／スピンドルからの信号 (NCK +/- PLC)	
エッジ評価：なし	信号更新：周期的	信号が有効となる最初の SW バージョン：2.1
信号ステータス 1 あるいは信号遷移 0 ---> 1	IS "Cam activation"（カム起動）(DB31-48, DBX2.0) を介して選択された軸の全カムが、 正常に起動した。	
信号ステータス 0 あるいは信号遷移 1 ---> 0	軸のカムが起動していない。	
関連性	IS "Cam activation"（カム起動） (DB31 ～ 62, DBX2.0) IS "Minus cam signal 1-32"（マイナスカム信号 1 ～ 32） (DB10, DBX110.0 - 113.7) IS "Plus cam signal 1-32"（プラスカム信号 1 ～ 32） (DB10, DBX114.0 - 117.7)	



## 2.9.6 例

なし

## 2.9.7 データフィールド, リスト

### ■ インタフェース信号

DB 番号	ビット, バイト	名称	参照
一般事項			
10	110.0 . 110.7	Minus cam signal 1...8 (マイナスカム信号 1 ~ 8)	
10	111.0 . 111.7	Minus cam signal 9...16 (マイナスカム信号 9 ~ 16)	
10	112.0 . 112.7	Minus cam signal 17...24 (マイナスカム信号 17 ~ 24)	
10	113.0 .... 113.7	Minus cam signal 25...32 (マイナスカム信号 25 ~ 32)	
10	114.0 ... 114.7	Plus cam signal 1...8 (プラスカム信号 1 ~ 8)	
10	115.0 ... 115.7	Plus cam signal 9...16 (プラスカム信号 9 ~ 16)	
10	116.0 ... 116.7	Minus cam signal 17...24 (マイナスカム信号 17 ~ 24)	
10	117.0 ... 117.7	Minus cam signal 25...32 (マイナスカム信号 25 ~ 32)	
軸別			
31 ~ 62	2.0	Cam activation (カム起動)	
31 ~ 62	62.0	Cam active (カムアクティブ)	

### ■ マシンデータ

番号	識別子	名称	参照
一般 (\$MN _ .)			
10260	CONVERT_SCALING_SYSTEM	基本系切換えアクティブ	G2
10270	POS_TAB_SCALING_SYSTEM	位置テーブルの測定系	T1
一般 (\$MN _ .)			
10450	SW_CAM_ASSIGN_TAB[n]	マシン軸へのソフトウェアカムの割当て	
10460	SW_CAM_MINUS_LEAD_TIME[n]	マイナスカム 1 ~ 16 のリードまたは遅延時間	
10461	SW_CAM_PLUS_LEAD_TIME[n]	プラスカム 1 ~ 16 のリードまたは遅延時間	
10470	SW_CAM_ASSIGN_FASTOUT_1	NCK I/O へのカム 1 ~ 8 の出力のハードウェア割当て	
10471	SW_CAM_ASSIGN_FASTOUT_2	NCK I/O へのカム 9 ~ 16 の出力のハードウェア割当て	
10472	SW_CAM_ASSIGN_FASTOUT_3	NCK I/O へのカム 17 ~ 24 の出力のハードウェア割当て	
10473	SW_CAM_ASSIGN_FASTOUT_4	NCK I/O へのカム 25 ~ 32 の出力のハードウェア割当て	
10480	SW_CAM_TIMER_FASTOUT_MASK	NCU で, タイマ割込みを介して出力されるカム信号の画面フォーム	

## ■ 設定データ

番号	識別子	名称	参照
一般 (\$MN_ .)			
41500	SW_CAM_MINUS_POS_TAB_1[n]	マイナスカム 1 ～ 8 の位置	
41501	SW_CAM_PLUS_POS_TAB_1[n]	プラスカム 1 ～ 8 の位置	
41502	SW_CAM_MINUS_POS_TAB_2[n]	マイナスカム 9 ～ 16 の位置	
41503	SW_CAM_PLUS_POS_TAB_2[n]	プラスカム 9 ～ 16 の位置	
41504	SW_CAM_MINUS_POS_TAB_3[n]	マイナスカム 17 ～ 24 の位置	
41505	SW_CAM_PLUS_POS_TAB_3[n]	プラスカム 17 ～ 24 の位置	
41506	SW_CAM_MINUS_POS_TAB_4[n]	マイナスカム 25 ～ 32 の位置	
41507	SW_CAM_PLUS_POS_TAB_4[n]	プラスカム 25 ～ 32 の位置	
41520	SW_CAM_MINUS_TIME_TAB_1[n]	マイナスカム 1 ～ 8 のリードまたは遅延時間	
41521	SW_CAM_PLUS_TIME_TAB_1[n]	プラスカム 1 ～ 8 のリードまたは遅延時間	
41522	SW_CAM_MINUS_TIME_TAB_2[n]	マイナスカム 9 ～ 16 のリードまたは遅延時間	
41523	SW_CAM_PLUS_TIME_TAB_2[n]	プラスカム 9 ～ 16 のリードまたは遅延時間	
41524	SW_CAM_MINUS_TIME_TAB_3[n]	マイナスカム 17 ～ 24 のリードまたは遅延時間	
41525	SW_CAM_PLUS_TIME_TAB_3[n]	プラスカム 17 ～ 24 のリードまたは遅延時間	
41526	SW_CAM_MINUS_TIME_TAB_4[n]	マイナスカム 25 ～ 32 のリードまたは遅延時間	
41527	SW_CAM_PLUS_TIME_TAB_4[n]	プラスカム 25 ～ 32 のリードまたは遅延時間	

## ■ アラーム

アラームの詳細説明については次を参照してください。

参照：アラーム一覧

あるいは MMC 101/102/103 を有するシステムのオンラインヘルプ

---

## 2.10 補助位置決め軸 (P2)

### 2.10.1 概要説明

最新の工作機械には、加工用軸だけでなく、次のような補助動作用軸も装備することができます：

- ツールマガジン用軸。
- ツールタレット用軸。
- ワーク移動用軸。
- パレット移動用軸。
- ロータ用軸（また、マルチ軸）。
- ツールチェンジャ用軸。
- クイル／スリーブあるいはステディ用の軸。

#### 位置決め軸

「位置決め軸」機能によって、補助動作用軸の制御装置への統合がより容易になります。

下記の状態のとき、位置決め軸を統合することが簡単になります。

- プログラミング中：  
同じパートプログラム内で、ワーク加工のための軸と共に、プログラミングを実行します。貴重な加工時間をむだにすることがありません。
- プログラムテスト／スタートアップ中：  
全ての軸のプログラムテストおよびスタートアップを同時に行ないます。
- 運転中：  
全ての軸の加工処理運転と監視をすべての軸に同時に開始します。
- PLC 構成／スタートアップ中：  
PLC または外部コンピュータ (PCs) において、加工用軸と補助動作用軸との間の同期化を見込んでおく必要はありません。
- システム構成中：  
2 つ目のチャンネルは必要ありません。

## 2.10.2 詳細説明

複合的な最新の工作機械には、加工に必要な軸だけでなく、補助動作用の他の軸も装備することができます。ワーク加工用軸は補間として知られており、チャンネル内で同時に始動、加速し、一緒に最終ポイントに到達するよう補間器によって導かれます。

補助軸には次のものがあります：

- ツールマガジン用軸
- ワークピース移動用軸
- パレット移動用軸
- ロード用軸（また、マルチ軸）
- ツールチェンジャ用軸
- クイル／スリーブあるいはステディ用の軸

これらの軸の多くは、以前は、油圧式で操作し、補助ファンクションを使ってパートプログラムによりトリガしていました。

NC で軸を制御すれば、軸はパートプログラムでの名称で呼ぶことができ、実際位置をスクリーンに表示することができます。

位置決め軸は、それ自身の専用軸別フィードレートで、補間とは独立してトラバースされます。

同期軸およびジオメトリ軸は、位置決め軸として、モーダルではなくトラバースすることができます。

位置決め軸には特殊なトラバース命令が与えられます。すなわち、POS[...]、POSA[...] です。

### ■ 位置決め軸の選択

工作機械の補助動作用に軸が用意されると、その軸の扱いは、軸のプロパティによって決まります。

- 別のパートプログラムでプログラムされる。  
→ ■独立したチャンネルを参照のこと。
- 加工プロセスと同じパートプログラムにプログラムされる。  
→ ■位置決め軸を参照のこと。
- 加工中 PLC から独占的にトリガされる。  
→ ■同時位置決め軸を参照のこと。

## ■ 独立したチャンネル

チャンネルは NC に組み込まれており、パートプログラムの補助として使用し、他のチャンネルとは無関係に、軸、スピンドル、加工ファンクションの動作を制御することができます。

チャンネル間の独立は、次の規定項目によって保証されます：

- チャンネルごとに 1 つのアクティブなパートプログラム。
- 次の例のようなチャンネル別信号。
  - IS "NC Start" (DB21,...DBX7.1)
  - IS "NC Stop" (DB21,...DBX7.3)
  - IS "Reset" (DB21,...DBX7.7)
- チャンネルごとに 1 フィードレートオーバーライド。
- チャンネルごとに 1 高速トラバースオーバーライド。
- チャンネル別評価およびアラーム表示。
- チャンネル別表示。たとえば、次のとおり：
  - 実際軸位置
  - アクティブな G ファンクション
  - アクティブなヘルプファンクション
  - 現行のプログラムブロック
- チャンネル別テストおよびチャンネル別プログラム修正：
  - 単一ブロック
  - ドライラン (DRY RUN)
  - ブロックサーチ
  - プログラムテスト

可能なチャンネル機能についての詳しい情報は、下記を参照してください。

参照： 1.10 運転モードの種類とプログラム運転モード (K1)

## ■ 位置決め軸

位置決め軸は、補間すなわち、ワーク加工を担う軸と共にプログラムされます。

位置決め軸と補間のいずれの命令も、同じ NC ブロックに入れることができます。

補間も位置決め軸も、同じ NC プログラムにプログラムすることができますが、互いに補間されていないので、それぞれの最終ポイントに同時に到達しません（直接の時間関係はありません。■運動動作も参照のこと）。ブロックチェンジは、プログラムされた位置決め軸のタイプによって決まります（■ブロックチェンジ参照のこと）。

- タイプ 1 すべてのパス軸と位置決め軸が、それぞれのプログラムされた最終ポイントに到達したとき、ブロックチェンジが起こります。
- タイプ 2 すべての補間が、それぞれのプログラムされた最終ポイントに到達したとき、ブロックチェンジが起こります。

タイプ 2 の軸を位置決めすると、いくつかのブロック境界線を越えて、プログラムされた最終位置にアプローチできます。

位置決め軸によって、同じ加工プログラムから起動でき、その動作はブロックリミット（タイプ 1）または WAITP コマンドを使って、明確にされたポイント（タイプ 2）で同期化されます。

## 補間および位置決め軸の従属性

補間と位置決め軸は、下記の方法によって互いに独立します。

- 共有する補間がない。
- 各位置決め軸は、専用軸補間器を備えている。
- 各位置決め軸には専用フィードオーバーライドがある。
- 専用プログラム可能フィードレート。
- 専用「軸別移動距離削除」インターフェース信号。

## 位置決め軸の従属性

位置決め軸は、下記の条件では互いに作用します：

- 共有されたパートプログラム。
- パートプログラムのブロック境界線でのみ、位置決め軸が始動する。
- 急速トラバースオーバーライドなし。
- 下記の信号は、チャンネル全体と、それにしたがって位置決め軸に作用する。
  - IS "NC Start" (DB21,...DBX7.1) NC 始動
  - IS "NC Stop" (DB21,...DBX7.3) NC 停止
  - IS "Reset" (DB21,...DBX7.7) リセット
  - IS "Read-in disable" (DB21,...DBX6.1) 読み込みディスエーブル
- プログラム別アラームおよびチャンネル別アラームもまた、位置決め軸を停止する。
- プログラム制御（ドライランフィード、プログラムテスト、DRF、... 等）もまた、位置決め軸に作用する。
- ブロックサーチおよび単一ブロックもまた、位置決め軸に作用する。
- G ファンクション（G0,G1,G2 等）のグループ 1（モーダル効果のある動作コマンド）は、位置決め軸に適用されない。

参照： YS840DI ユーザーズマニュアル・プログラミング編

## アプリケーション

下記は、位置決め軸の一般的なアプリケーションです：

- 単一軸ローダ。
- 補間なしのマルチ軸ローダー（PTP → ポイントトゥポイントトラバース）。
- ワークのフィードおよび移動。

位置決め軸は、軸間の補間が必要なマルチ軸ローダには合いません（補間器関数発生器）。

## ■ 同時位置決め軸

同時位置決め軸は、次の属性をもった位置決め軸です：

- PLC からの起動は、ブロックリミットで行なわれる必要はなく、どんな操作モードでも、いつでも実行できる（パートプログラムがすでにチャンネルで処理中であっても）。
- プログラムコマンド "WAITP" は、電源オン直後、パートプログラムからの同時位置決め軸を移動させるのに必要。
- 同時位置決め軸が PLC によって定義された位置に到達していなくても、パートプログラムは制約なく動作をつづける。
- ソフトウェアバージョン 4.3 以降。
- マシンデータ AUTO\_GET\_TYPE によって、プログラミングコマンドを使用する。
  - パートプログラムからの "GET(<axis>)" または "WAITP(<axis>)" で、同時位置決め軸をチャンネル軸として再度トラバースできます。あるいは、
  - "RELEASE(<axis>)" または "WAITP(<axis>)" を使って、チャンネル軸を同時位置決め軸としてトラバースすることができます。

## PLC からの起動

PLC からの FC 15 または FC 16 を介して、同時位置決め軸が起動します。

- フィードレート（フィードレートを 0 に設定して、MD 32060:POS\_AX\_VELO（位置決め軸速度にはリセット）にセットされたフィード）が適用される。
- 絶対寸法 (G90)、相対寸法 (G91)、回転軸用最短パスに沿った絶対寸法（回転軸名 = DC（値））。

次のファンクションが定義されます：

- 直線補間 (G01)。
- mm/min または度 /min (G94) でのフィードレート。
- イグザクトストップ (G09)。
- 現時点で選択された設定可能なゼロオフセットが有効。

## アプリケーション

同時位置決め軸には、一般的にアプリケーションが含まれています：

- 加工中、手動でロードとアンロードを行なうことのできるツールマガジン。
- 加工中、ツール準備を行なうことのできるツールマガジン。

## ■ 動作

### 補間関数発生器

各チャンネルは、直線補間 (G01)、円形補間 (G02/G03)、スプライン補間など、広範囲の補間モードのための補間関数発生器を備えています。

### 軸補間器

補間関数発生器の他にも、各チャンネルは5つの軸補間器を備えています。位置決め軸がプログラムされると、軸補間器が制御されて始動します（直線補間 G01で）。

軸補間器は、位置決め軸のプログラムされた最終位置に到達するまで、補間関数発生器とは無関係に動作します。

また、補間関数発生器と軸補間器との間、軸補間器間にも時間的な関係はありません。

軸がインポジション (G09) に到達する時点で、すでに位置決め軸は、プログラムされた最終位置に到達しています。連続パスモード (G64) は、位置決め軸では使用できません。

## ■ ブロックチェンジ

位置決め軸は、個々に、または輪郭軸と組合せて NC ブロックにプログラムできます。

輪郭軸と位置決め軸は、常に別々に補間されているので（補間関数発生器および軸補間器）、それぞれの時間で、それぞれのプログラムされた最終位置に到達するようになっています。

位置決め軸には2種類あり、その応答はブロックチェンジに関連して異なります。

- タイプ1 すべての補間と位置決め軸が、それぞれのプログラムされた最終ポイントに到達してから、ブロックチェンジが起こります。
- タイプ2 すべての補間が、それぞれのプログラムされた最終ポイントに到達してから、ブロックチェンジが起こります。



## 位置決め軸タイプ 1

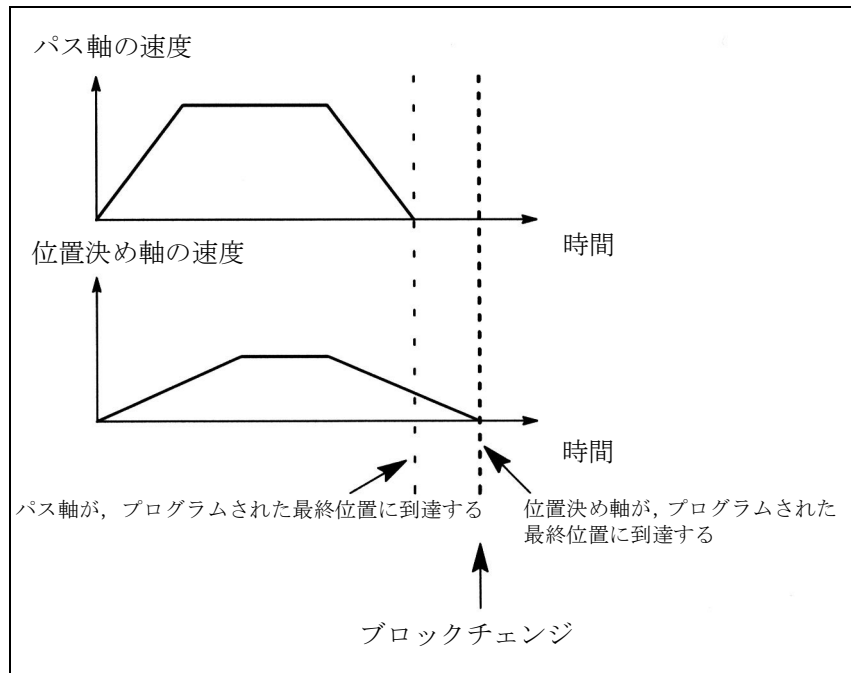


図 2.90 位置決め軸タイプ 1 のブロックチェンジの例

## 位置決め軸タイプ 1 の特性

- すべての補間および位置決め軸がそれぞれの最終位置に到達すると、ブロックチェンジが起こります (NC ブロックが完了する)。
- 位置決め軸が補間よりも先に最終位置に到達する場合は、連続パスモード (G64) のみが可能です (図 2.90 での例にはあてはまりません)。
- プログラミングは,  
POS[name] = end point FA[name] = feed ,  
あるいは,  
POSA[name] = end point と略されます  
これは、MD 32060: POS\_AX\_VELO にセットされているフィードが適用される場合に限りです。
- プログラムされた命令は、モードを用いず、有効です。ジオメトリ軸と同期軸は、組み合わされた補間から離れ、軸別速度でトラバースします。

## 位置決め軸タイプ 2

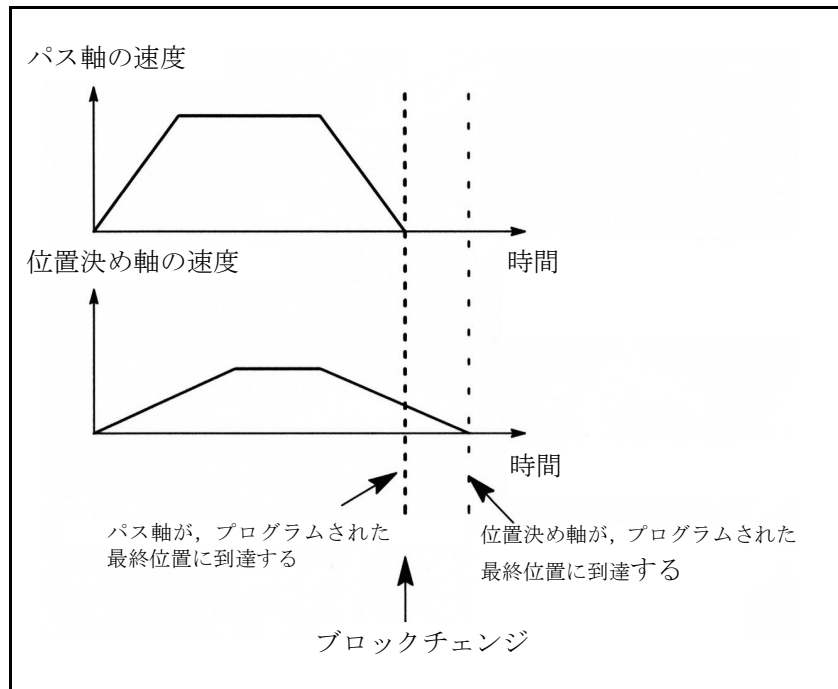


図 2.91 位置決め軸タイプ 2 のブロックチェンジ，シーケンス例

### 位置決め軸タイプ 2 の特性

- 補間がそれぞれの最終位置に到達すると、ブロックチェンジが起こります (NC ブロックが完了する)。
- 位置決め軸は、いくつかのブロック境界線を越えて、それぞれのプログラムされた最終位置にトラバースできます。
- "NC block finished" と、タイプ 2 の位置決め軸がそれぞれのプログラムされた最終位置に到達するポイントとの間には時間的關係がないので、WAITP 座標軸コマンドが、位置決め軸 (■プログラミングを参照) の同期化のために用意されています。
- 位置決め軸が前の位置に到達するまでに再度プログラムされると、"axis cannot be repositioned" のアラームが出されます。
- プログラミングは、  
POS[name] = end point FA[name] = feed ,  
あるいは、  
POSA[name] = end point と略されます  
これは、MD 32060: POS\_AX\_VELO にセットされているフィードが適用される場合に限ります。
- プログラムされた命令は、モードを使わずに、有効です。ジオメトリ軸と同期軸は、組み合わされたパス軸から離れ、軸別速度でトラバースします。

## ■ 速度

軸別速度リミットと加速度リミットは、位置決め軸に有効です。

位置決め軸は、直線軸および回転軸となります。

位置決め軸はまた、インデックス軸ともなります。

参照： 2.15 割出し軸 (T1)

## フィードレートオーバーライド

補間と位置決め軸は、別々のフィードレートオーバーライドを備えています。各位置決め軸は、それぞれの軸別フィードオーバーライドによって調整できます。

## 高速トラバースオーバーライド

急速トラバースオーバーライドは、パス軸にのみ使用できます。位置決め軸は、急速トラバース補間を設けてあるので（軸の直線補間 G01）、高速トラバースオーバーライドはありません。

## フィードレート

位置決め軸は、そのためにプログラムされた軸別フィードレートでトラバースします。■運動動作の図のように、フィードレートはパス軸の影響を受けません。

フィードレートは、分/ミリメートル、インチ/分、度/分の単位の軸別速度としてプログラムされます。

軸別フィードレートは、軸名により、位置決め軸に恒久的に割当てられます。

位置決め軸がプログラムされたフィードレートを持たない場合は、制御システムが自動的に軸別に設定されたフィードレート MD 32060:POS\_AX\_VELO（位置決め軸速度のための初期設定）を適用します。

プログラムされた軸別フィードレートは、プログラムの終了までアクティブのままです。

## 回転フィードレート

JOG モードでは、軸/スピンドルの応答もまた、SD 41100:JOG\_REV\_IS\_ACTIVE での設定で決まります（アクティブ JOG のための回転フィードレート）。

- この設定データがアクティブの場合、軸/スピンドルが常に、回転フィードレート MD 32050:JOG\_REV\_VELO（JOG のための回転フィードレート）または MD 32040:JOG\_REV\_VELO\_RAPID（高速トラバースオーバーライドで、JOG のための回転フィードレート）で、マスタスピンドルのファンクションとしてトラバースされます。
- 設定データがアクティブでない場合は、SD 43300:ASSIG\_FEED\_PER\_REV\_SOURCE（軸/スピンドルを位置決めするための回転フィードレート）の設定のファンクションとして、軸/スピンドルが応答します。
- 設定データがアクティブでない場合、回転のあるフレームの影響を受けるジオメトリ軸は、チャンネル別設定データ SD 42600:JOG\_FEED\_PER\_REV\_SOURCE でのファンクションとして応答します（JOG モードでは、回転のフレームがアクティブであるジオメトリ軸の回転

フィードレート)。

## ■ PLC による制御

### チャンネル別信号

すべてのチャンネル別信号は、パス軸と位置決め軸には同じ程度に作用します。

次の信号は、唯一の例外です (2.10.5 信号の説明を参照のこと) :

- IS "Feedrate override"(DB21,...DBB4)
- IS "Delete distance-to-go"(DB21,...DBX6.2)

### 軸別信号

次の追加信号は、位置決め軸に使用できます (2.10.5 信号の説明を参照のこと) :

- IS 「位置決め軸」(DB31,...DBX76.5)
- FA 位置決め軸のための F ファンクション (フィード)
- 軸別フィードオーバーライド
- IS 「移動距離削除」(DB31,...DBX2.2), 軸別

### FC15 のためのパラメータ

同時位置決め軸が PLC から起動すると、FC15 が呼び出されて次のパラメータデータが与えられます :

- 軸名/軸番号。
- 最終位置。
- フィードレート (0 のフィードレート設定による MD 32060:POS\_AX\_VELO) が適用される。
- 絶対寸法 (G90), 相対寸法 (G91), 回転軸の最短パスに沿った絶対寸法 (回転軸名 = DC (値))。

次のファンクションが定義されます :

- 直線補間 (G01)。
- 単位 mm/ 分あるいは度/分 (G94) でのフィードレート。
- イグザクトストップ (G09)。
- 現時点で選択されている設定可能ゼロオフセットは有効

## ■ プログラミング

(注) 注記

下記の資料は、位置決め軸のプログラミングを行なう前に調べておくことをお勧めします：

参照： YS840DI ユーザーズマニュアル・プログラミング編

1つのブロックにプログラムできる位置決め軸の最大数は、5つに限られています。

### 定義

位置決め軸は、下記のパラメータを使って定義されます：

- ・ 軸タイプ：位置決め軸タイプ 1 またはタイプ 2。
- ・ 最終ポイント寸法。
- ・ 最終ポイント寸法の絶対寸法、または相対寸法。
- ・ 直線軸のフィードレートの単位は [mm/min]、回転軸の単位は [度/min] である

### 構文

位置決め軸タイプ 1：

POS[Q1] = 200 FA[Q1] = 1000; 軸 Q1 は、フィードレート 1000 mm/min で、位置 200 まで

位置決め軸タイプ 2:

POSA[Q2] = 300 FA[Q2] = 1500; 軸 Q2 は、フィードレート 1500 mm/min で位置 300 まで

パートプログラム内では、軸を、パス軸または位置決め軸のいずれかにすることができます。一方、動作ブロック内では、各軸には特定の軸タイプを割当てなければなりません。

### 絶対寸法／相対寸法

最終ポイント寸法は、絶対寸法 (G90) または相対寸法 (G91) としてプログラムされます：

#### 絶対寸法

G90 POS[Q1] = 200

G91 POS[Q1] = AC(200)

#### 相対寸法

G91 POS[Q1] = 200

G90 POS[Q1] = IC(200)

## 協調

タイプ 2 の位置決め軸（ブロックリミットを超える動作）の場合、軸が最終位置に到達したかどうかは、パートプログラムによって決めなければなりません。この位置決め軸を再プログラムできるのは、こうした場合のみです（そうでない場合は、アラームが出されます）。

WAITP 協調コマンドは、最終位置に到達したかどうかをパートプログラムに問い合わせるために用いられます。

強調コマンドの WAITP は、別のブロックでプログラムされます。

プログラムが待機することになる軸はどれでも、はっきりと参照される必要があります。

見本プログラム：

N10 G01 G90 X200 F1000 POSA[Q1] = 200 FA[Q1] = 500

N15 X400

N20 WAITP(Q1); Q1 が位置につくまで、プログラムの実行は自動的に停止する

N25 X600 POS[Q1] = 300; Q1 は位置決め軸タイプ 1 である（ブロック N10 から FA[Q1] をフィードする）

N30 X800 Q1 = 500; Q1 はパス軸となる（ブロック N10 からの パスフィード F1000）

## ツールオフセット

位置決め軸のツール長さ補償は、軸ゼロオフセットを使って実行できます（たとえば、ローダの位置決めパスが変更される）。軸のゼロオフセットがツール長さ補償の代わりに使用されるような場合、多様な寸法のツールを含むローダは、障害物を回避しなくてはなりません。

## プログラムエンド

プログラムエンド（選択されたプログラムステータス）は、すべての軸（パス軸 + 位置決め軸）が、プログラムされた各最終ポイントに到達するまで遅延されます。

### ■ 外部からのプログラミング

外部から与えられた回転軸フィードレートでトラバースするには、軸データ SD 43300: ASSIGN\_FEED\_PER\_REV\_SOURCE ( 軸の回転フィードレート ) および JOG モードで、チャンネル別設定データ SD 42600: JOG\_FEED\_PER\_REV\_SOURCE を介して選択できます。次の設定は、設定データを介して可能になります。

- >0: 回転軸フィードレートが導かれる回転軸／スピンドルのマシン軸番号
- -1: 回転フィードレートが、いずれの場合も軸／スピンドルがアクティブのチャンネルのマスラスピンドルから導かれる
- 0: ファンクションが選択解除される

## ■ 特殊ファンクションでの応答

### ■ ドライランフィードレート (DRY RUN)

ドライランフィードレートもまた、プログラムされたフィードレートがドライランフィードレートより大きい場合を除いて、位置決め軸には有効です。

### ■ 単一ブロック

#### 位置決め軸タイプ 1

単一ブロックは、タイプ 1 の位置決め軸では有効です。

#### 位置決め軸タイプ 2

タイプ 2 の位置決め軸は、単一ブロックモードでもブロックリミットを超えて作動を続けます。

## 2.10.3 補足条件

ファンクションの説明のために規定された補足条件はありません。

## 2.10.4 データの説明 (MD, SD)

### ■ 軸／スピンドル別マシンデータ

30450 MD 番号	IS_CONCURRENT_POS_AX RESET リセットでデフォルト：中立軸またはチャンネル軸	
デフォルト値：0	最小入力リミット：0	最大入力リミット：1
変更が有効になるための条件： RESET リセット	保護レベル：2	単位：なし
データタイプ：BOOLEAN	適用開始 SW バージョン：1.1 以降	
意味：	軸は同時位置決め軸である。 ソフトウェアバージョン 4.3 以降（FM-NC ではない）： 偽の場合：RESET で再び中立軸がチャンネル軸になる。 真の場合：RESET で中立軸は中立軸ステータスのまま、チャンネル軸が中立軸になる。	

32060 MD 番号	POS_AX_VELO 位置決め軸速度の初期設定	
デフォルト値：0	最小入力リミット：0	最大入力リミット：
変更が有効になるための条件：電源オン	保護レベル：2	単位：mm/min, rpm
データタイプ：DOUBLE	適用開始 SW バージョン：1.1 以降	
意味：	軸別フィードレートを指定せず、パートプログラムに位置決め軸をプログラムする場合、MD: POS_AX_VELO に入力されたフィードレートが自動的に使用される。MD: POS_AX_VELO からのフィードレートは、この位置決め軸用の軸別フィードレートが、パートプログラムにプログラムされるまで適用される。	

32060 MD 番号	POS_AX_VELO 位置決め軸速度の初期設定
この MD は次の場合は無効：	POS_AX_VELO は、他のすべての軸タイプに関して、位置決め軸としては無意味である。
例外，エラー：	ゼロ速度設定が POS_AX_VELO に入力されている場合、フィードなしとプログラムされていれば、位置決め軸はトラバースしない。速度設定が POS_AX_VELO に入力されている場合、軸の最高速度 (MD 32000: MAX_AX_VELO) よりも高くなると、速度は自動的に最高速度に制限される。

## 2.10.5 信号の説明

NCK/MMC/PLC インタフェースにおける下記の信号やコマンドは、位置決め軸にとっては意味があります：

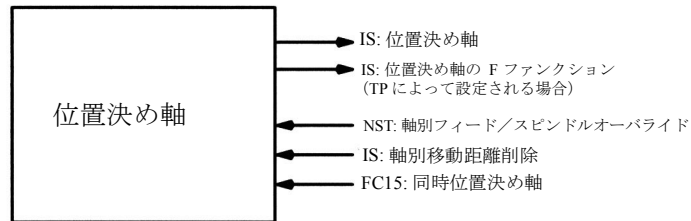


図 2.92 PLC による信号修正

### ■ 軸／スピンドル別信号

DB31, ... DBB0 データブロック	フィードレートオーバーライド／スピンドル速度オーバーライド，軸別軸／スピンドルからの信号 (NCK → PLC)	
エッジ評価：なし	信号更新：周期的	信号が有効となる 最初の SW バージョン：1.1
信号ステータス 1 あるいは信号遷移 0 → 1	位置決め軸が専用軸別フィードオーバーライドを持つ。軸別フィードオーバーライドの評価は、チャンネル別フィードオーバーライドと同じである。	
この信号は次の場合は無効	IS "Positioning axis" (DB31, ... DBX74.5) = ZERO	
参照	評価について、IS “フィードレートオーバーライド” (DB21, ... DBB4) チャンネル別を参照のこと。	

DB31, ... DBX2.2 データブロック	移動距離削除，軸別軸／スピンドルからの信号 (NCK → PLC)	
エッジ評価：あり	信号更新：周期的	信号が有効となる 最初の SW バージョン：1.1
信号ステータス 1 あるいは信号遷移 0 → 1	位置決め軸の軸別移動距離がキャンセルされる。位置決め軸は減速され、次のエラーが削除される。プログラムされた最終位置に到達したとみなされる。パス軸は、軸別「移動距離削除」の影響を受けない。チャンネル別「移動距離削除」インタフェース信号は、このために使用される。	
例外，エラー	軸別「移動距離削除」インタフェース信号がイネーブルされる場合、位置決め軸がこのブロックにプログラムされていなくても、NCK は応答しない。	
関連性	IS “移動距離削除” (DB21, ... DBX6.2)，チャンネル別，パス軸について	



DB31, ... DBX76.5 データブロック	位置決め軸 軸／スピンドルからの信号 (NCK → PLC)	
エッジ評価：なし	信号更新：周期的	信号が有効となる 最初の SW バージョン：1.1
信号ステータス 1 あるいは信号 遷移 0 → 1	NCK は軸を位置決め軸として扱う。NCK は次のものを装備している： 専用軸補間器（直線補間器） 専用フィードレート（F 値） 専用フィードオーバーライド プログラムされた最終位置での正確停止 (G09)	

DB31, ... DBB78-81 データブロック	位置決め軸のためのファンクション（フィードレート） 軸／スピンドルからの信号 (NCK → PLC)	
エッジ評価：なし	信号更新：変更時	信号が有効となる 最初の SW バージョン：1.1
信号ステータス 1 あるいは信号 遷移 0 → 1	プログラムされた軸フィードは、プログラムされた軸名によって位置決め軸に割当てられ、この軸のために PLC に出力される。F15 を介して、プリセットされた値の出力はない。下記参照。	
この信号は次の場合は無効	IS: 位置決め軸 = ZERO	
例外，エラー	位置決め軸が MD 32060: POS_AX_VELO（位置決め軸速度の初期設定）で設定されたフィードレートでトラバースされる場合、NC は、PLC に対し F ファンクション（フィード）を出力しない。	
関連性	IS "Positioning axis"(DB31, ... DBX74.5) MD 22240: AUXFU_F_SYNC_TYPE F ファンクションの出力時間	

## ■ ファンクションコール

### FC15

PLC ファンクションコール FC15 を用いて、PLC から同時位置決め軸を始動することができます。次のパラメータはファンクションに渡されます：

- ・ 軸名／軸番号。
- ・ 最終位置。
- ・ フィードレート（フィードレート設定 = 0 による MD 32060:POS\_AX\_VELO）にセットされたフィードが適用されます。
- ・ FC15 の F 値は、軸別 IS “F function (feedrate) for positioning axis” DB31, ...DBB78-81 には送られません。
- ・ 絶対座標 (G90), 相対座標 (G91), 回転軸の最短パスに沿った絶対座標（回転軸名 = DC（値））

各軸は 1 つのチャンネルに割当てられるので、制御装置が軸名／軸番号から正確なチャンネルを選んで、このチャンネルで同時位置決め軸を始動することができます。

参照： 1.15 基本的 PLC プログラム (P3)

## 2.10.6 例

下記の例では、2つの位置決め軸の Q1 と Q2 が分離した 2つの動作ユニットを表します。2つの軸に補間関係はありません。この例では、位置決め軸がタイプ 1（例では、N20 に）とタイプ 2（例では、N40 に）としてプログラムされます。

### プログラム例

```
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
```

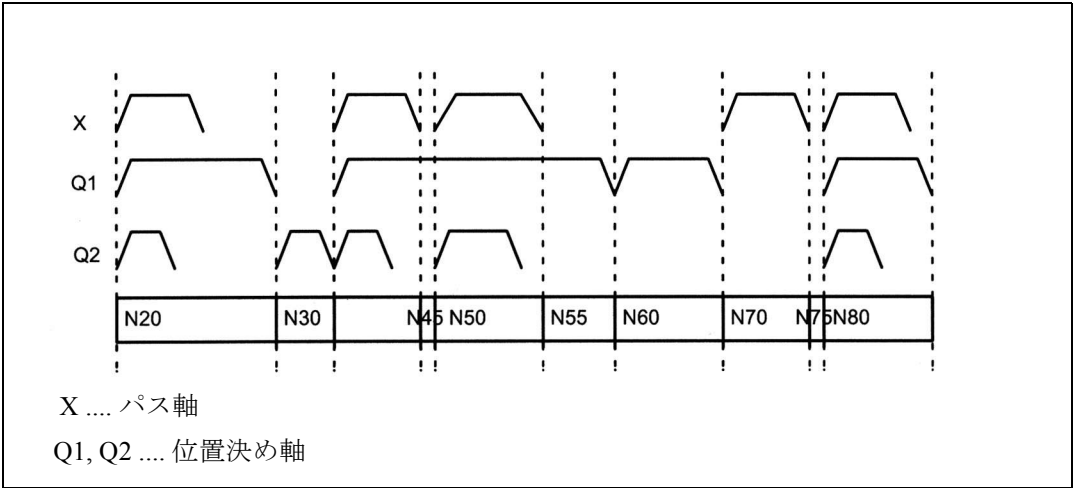


図 2.93 パス軸および位置決め軸

## 2.10.7 データフィールド, リスト

### ■ インタフェース信号

DB 番号	ビット, バイト	名称	参照
軸／スピンドル別			
31-48	0	軸別フィードオーバーライド	
31-48	2.2	軸別移動距離削除	
31-48	74.5	位置決め軸	
31-48	78-81	位置決め軸の F ファンクション (フィード)	

### ■ マシンデータ

番号	識別子	名称	参照
チャンネル別 (\$MC_...)			
22240	AUXFU_F_SYNC_TYPE F	ファンクションの出力時間	H2
軸別 (\$MC_...)			
30450	IS_CONCURRENT_POS_AX	同時位置決め軸	
32060	POS_AX_VELO	位置決め軸のフィードレート	

---

## ■ 7.3 アラーム

アラームの詳細説明については次を参照してください。

参照： アラーム一覧

あるいは MMC 101/102/103 を有するシステムのオンラインヘルプ Alarms

## 2.11 振動機能 (P5)

### 2.11.1 詳細説明

#### 定義

「振動」機能の選択時、振動軸は、プログラムされた送り速度または2点の反転位置間から導かれた送り速度(回転速度)で前後に振動します。複数の振動軸を同時に有効にすることができます。

#### 振動方式

振動機能は、送り内の反転位置の軸応答によって分類できます。

- ブロックリミットを超える非同期振動  
非同期振動動作中、他の軸は随意に補間が可能です。振動軸は、動の変換入力軸またはガントリ軸やカップリングされたモーション軸のためのマスタ軸として機能します。振動は、AUTOMATIC モードに自動的に接続されません。
- 連続送りで生じる振動  
送りは、複数の軸で同時に実行することができます。しかし、送り動作と振動動作との間には、補間関係はありません。
- 左側か右側のいずれかの反転位置または両方の反転位置での送りを伴った振動。反転位置からのプログラム可能距離で、送りが開始できます。
- 振動後の退避ストローク

#### 反転位置の応答

方向変更が起こるのは下記の場合です。

- 正確な到達停止リミットがない場合(正確停止(微または粗))
- プログラムされた位置への到達後
- プログラムされた位置へ到達し、滞在終了後

#### 制御方法

振動動作は、下記のような多様な方法で制御できます。

- 振動動作および／または送りは移動距離削除によって中断できます。
- 反転位置は、NC プログラム、PLC、MMC、ハンドル、または方向キーによって修正できます。
- 振動軸の送り速度は、NC プログラムに入力された値から修正できます。送り速度は、主軸、回転軸またはスピンドルで決まるようにプログラムできます(回転速度)。

参照： 1.18 送り機能 (V1)

- 振動動作は、PLC によって全体を制御できます。

## 2.11.2 詳細説明

### 振動制御方法

振動のモードには次の 2 種類があります。

#### 1. 非同期振動

ブロックリミットを超えても有効で、PLC/MMC から始動できます。

#### 2. モーション同期作用により制御された振動

この場合、非同期振動および送り動作が同期動作を介して互いに結合されます。モーダルでない場合にアクティブである反転位置での、送りを伴う振動をプログラムすることが可能です。

### ■ 非同期振動

#### 特性

非同期振動の特性は、下記のとおりです。

- 振動軸は、指定された送り速度の反転位置 1 と 2 の間を、振動動作が停止するまであるいは補足条件を満たした適切な応答があるまで、前後に振動します。振動軸は、動作の開始時に反転位置 1 で位置決めされない場合は、まずこの反転位置 1 に移動します。
- 直線補間 G01 は、プログラムで現時点で有効である G コードに関係なく、振動軸についてアクティブです。そうでなければ、回転速度 G95 が起動できません。
- 非同期振動は、ブロックリミットを超えても、軸別にアクティブです。
- 複数の振動軸が (すなわち、位置決め軸の最大数) 同時にアクティブであることが可能です。
- 振動動作中、振動軸以外の軸は随意に補間できます。連続送りはパス動作または位置決め軸によって実行できます。しかし、この場合、振動と送り動作との間には、補間的な関係はありません。
- PLC が、軸をコントロールしていない場合、軸は、非同期振動中、軸を通常位置決めするように扱われます。PLC 制御の場合、PLC プログラムは、必ず VDI インタフェースの適切な停止ビットを介して、軸は VDI 信号に適した方法で反応する必要があります。これらの信号には、プログラムエンド、操作モード切替え、単一ブロックがあります。
- 振動軸は、変形 (例えば、傾斜軸) の入力軸にすることができます。  
参照： 2.7 座標変換機能 (M1)
- 振動軸は、ガントリ軸と結合された動作軸のためのマスタ軸になることができます。  
参照： 3.2 ガントリ軸制御 (G1)
- ジャーク制限 (SOFT)、またはニーポイントのあるひざ型加速特性 (軸の位置決めに関して)
- パートプログラムを介して、振動動作のブロック同期起動が保証されます。
- 振動動作は、同様にして、PLC/MMC から開始、影響、停止が可能です。
- 補間振動はできません (斜めの振動など)。

## ■ 非同期振動への影響

### データの設定

振動に必要な設定データは、MMC または PLC を介して、NCK パートプログラムの特許言語コマンドを用いて設定できます。

### 送り速度

振動軸の送り速度は下記のとおりを選択またはプログラムされます。

- 位置決め軸としての軸に定義される速度は、送り速度として用いられます。この値は、FA[axis] を介してプログラムでき、モーダル作用があります。速度がプログラムされていない場合には、機械データ POS\_AX\_VELO に保存されている値が使用されます (位置決め軸参照)。
- 振動動作中は、振動軸の送り速度は設定データを介して変更できます。変更された速度が直ちに実行されなければならないか、あるいは次の反転位置で実行されるべきか、パートプログラムと設定データを介して指定できます。
- 送り速度は、オーバライド (軸 VDI 信号かつプログラム可能) によって影響を受けます。
- "Dry run" (ドライラン) が有効の場合、ドライラン速度設定が、現時点でプログラムされている速度よりも高ければ適用されます。
- 速度オーバレイ / パスオーバレイはハンドルで操作できます。表 2.17 を参照のこと。

参照： 2.4 JOG 運転とハンドル (H1)

- 振動軸は、回転速度で移動可能です。

### 回転速度

反転送りは、振動軸にも使用できます。

### 反転位置

反転位置は、振動動作の始動前か、進行中に、設定データを介して入力できます。

- 反転位置は、振動動作の前あるいはその途中で、振動動作がすでに中断されているかどうかに関係なく、手動動作 (ハンドル, JOG キー) によって入力できます。

下記は、反転位置の変更に適用されます。振動動作がすでに進行しているときには、反転位置の変更位置は、この反転位置に再アプローチするまでは、有効にはなりません。軸がすでに変更位置にアプローチしているのであれば、次の振動ストロークで修正が有効になります。

(注)

VDI インタフェース信号 "Activate DRF" がセットされると同時に反転位置を変更しなければならない場合には、DRF オフセットにも反転位置のオフセットにも、ハンドル信号が割当てられます。すなわち、反転位置は、距離の 2 倍に相当する量だけ無条件に変更されます。

## 停止時間

停止時間は、各反転位置の設定データを介してプログラムできます。設定は、NCプログラムの次のブロックで変更できます。その場合、次の適用可能な反転位置からのブロック同期が有効です。停止時間は、設定データを介して非同期に変更できます。停止時間は適切な反転位置が次に移動する瞬間から、有効になります。次の表は、停止時間入力によって、正確停止範囲や反転位置での動作を説明しています。

表 2.14 停止時間の効果

停止時間設定	応答
-2	正確停止を待たずに補間が継続する
-1	反転位置での粗正確停止を待つ
0	反転位置での微正確停止を待つ
>0	反転位置で微正確停止を待った後に停止時間を待つ

## 振動の停止

次のオプションは、振動モードの停止時に、振動動作を終了するために設定します。

- 次の反転位置で振動動作を終了
- 反転位置 1 で振動動作を終了
- 反転位置 2 で振動動作を終了

この終了プロセスに続いて、退避ストロークが処理され、プログラムされている場合は、終了位置にアプローチします。

非同期振動から退避に切替え時および退避中は、正確停止に関する反転位置での応答は、適切な反転位置についてプログラムされた停止時間によって決定された応答に対応します。退避ストロークとは、他の反転位置に向かってから戻るといふ動作のことです（表参照）。

表 2.15 振動の停止の操作シーケンス

ファクション	入力	説明
定義された反転位置で停止	退避ストローク数が 0 である，終了位置はアクティブでない	適切な反転位置で振動動作が停止する
退避ストロークの指定回数で停止	退避ストローク数が 0 である，終了位置はアクティブでない	適切な反転位置に到達後，コマンドで指定された退避ストローク数が処理される。
退避ストロークと定義された終了位置（オプション）で停止	退避ストローク数が 0 ではない，終了位置はアクティブである	適切な反転位置に到達後，コマンドで指定された退避ストロークの数が処理され，その後，指定終了位置にアプローチする。
退避ストロークはないが定義された終了位置で停止（オプション）	退避ストローク数が 0 ではない，終了位置はアクティブである	適切な反転位置に到達後，軸は指定終了位置にトラバースする。



## NC 言語

NC プログラミング言語によって、非同期振動をパートプログラムから制御できます。下記のファンクションによって、非同期振動が起動し、NC プログラム処理の関数として起動および制御できます。

(注)

設定データがパートプログラムに直接書込まれる場合には、パートプログラムの処理に、早まってデータ変更が起こります ( 前処理時に )。前処理停止 (STOPRE) を使えば、パートプログラムと振動機能コマンドを再同期化できます。

参照： YS840DI ユーザーズマニュアル・プログラミング編

### 1. 振動の起動、停止：

- OS[oscillation axis] = 1;      振動軸の振動起動
- OS[oscillation axis] = 0;      振動軸の振動停止

(注)

どの軸でも振動軸として使用できます。

### 2. 振動の終了：

- WAITP( 振動軸 )

位置決め軸コマンド - 振動軸が微停止し、前処理とメインランを同期化するまでブロックを停止します。振動軸は、再度位置決め軸として入力でき、普通に使用できます。軸を振動に用いる場合は、WAITP( 軸 ) コールを実行して、前もってイネーブルにしなければなりません。

振動が PLC/MMC から起動する必要がある場合にも、これが適用できます。

NC プログラムであらかじめ軸がプログラムしてある場合にも、WAITP( 軸 ) コールが必要です。ソフトウェアバージョン 3.2 以降であれば、WAITP() の実行を、プログラミングで行なうにしても自動的であっても、\$MA\_AUTO\_GET\_TYPE を介して選択できます。

(注)

WAITP は、振動動作の実行が済むまで、効果的に時間を遅延させます。たとえば、振動動作の終了は、NC プログラムのプログラムされた停止コマンドまたは移動距離削除を使って PLC または MMC から実行できます。

### 3. 反転位置の設定：

- OSP1[oscillation axis] = 反転位置 1 の位置
- OSP2[oscillation axis] = 反転位置 2 の位置

メインランのブロックと同期した状態で、適切な設定データに、位置が入力されると、設定データが次に変更されるまで、有効になります。

加算移動の場合、位置は、NC プログラムでプログラムされた最新の反転位置に加算されます。

#### 4. 反転位置の停止時間

- OST1[oscillation axis] = [s] における反転位置 1 での停止時間
- OST2[oscillation axis] = [s] における反転位置 2 での停止時間

メインランでのブロックと同期した状態で、適切な設定データに停止時間が入力されると、設定データが次に変更されるまで、有効になります。

停止時間の単位は、G04 でプログラムされた停止時間のために選択された単位と同じです。

#### 5. 移動速度の設定：

- FA[axis] = Fvalue  
位置決め軸移動速度

移動速度は、メインランでのブロックと同期した状態で適切な設定データに送られます。振動軸が回転速度で移動する場合、ファンクション V1 の説明に記載しているように、属性が指定されていなければなりません。

#### 6. 動作のシーケンスに制御設定値を設定する：

- OSCTRL[oscillation axis] = ( 設定オプション，リセットオプション )  
セットオプションは、下記のように定義されます ( リセットオプションでは、設定値を選択しません )。

表 2.16 セットオプション／リセットオプション

オプション値	意味
0	振動動作の停止時 ( デフォルト ) に、次の反転位置で停止。オプション値 1 および 2 をリセットしてのみ可能になる。
1	振動動作の停止時に反転位置 1 で停止。
2	振動動作の停止時に反転位置 2 で停止。
3	振動動作の停止時、退避ストロークがプログラムされていなければ、反転位置にアプローチしない。
4	退避処理後、終了位置にアプローチする。
8	振動動作が移動距離削除によって終了される場合は、指定された退避ストロークが実行され、終了位置 ( プログラムされている場合 ) に必ずアプローチする。
16	振動動作が移動距離削除によって終了される場合は、振動動作の停止時にプログラムされた反転位置に、必ずアプローチしなければならない。
32	変更されたフィードレートは、次の反転位置からのみ有効になる。
64	フィードレート設定が 0 の場合、パスオーバーレイがアクティブであるか、もしくは速度オーバーレイがアクティブである。
128	回転軸 DC ( 最短パス ) のため。

(注)

オプション値 0 ～ 3 は、電源オフ時、反転位置での挙動をエンコードします。このオプション値 0 ～ 3 のうち 1 つを選ぶことができます。他の設定値は、個々の要求事項にしたがって選択されたオプション値と組み合わせることができます。 "+" という記号を挿入して、オプションのストリングを作ることができます。

例：

軸 Z の振動動作は、停止時に反転位置 1 で停止しなければなりません。終了位置には、必ずアプローチし、新たにプログラムされた送り速度が即時有効になります。軸は、移動距離削除直後停止しなければなりません。

OSCTRL[Z] = (1+4, 16+32+64)

セット／リセットオプションは、メインランのブロックと同期した状態で適切な設定データに入力されると、設定データが次に変更されるまで、有効になります。

(注)

制御装置が、最初にリセットオプションを評価し、次に設定オプションを評価します。

7. 退避ストローク：

- OSNSC[oscillation axis] = 退避ストローク

退避ストローク数は、メインランのブロックと同期した状態で適切な設定データに入力されると、設定データが次に変更されるまで有効になります。

8. 振動の停止後、終了位置にアプローチする：

- OSE[oscillation axis] = 振動軸の終了位置

終了位置は、メインランのブロックと同期した状態で適切な設定データに入力されるので、設定データが次に変更されるまで有効になります。オプション値 4 は、セットされた終了位置にアプローチするというように、表 2.19 に従って、暗示的にセットされます。

## プログラミング例

2.11.6 「例」では、非同期振動の重要な要素をすべて含む例を示します。

### ■ PLC 制御下の非同期振動

#### 起動

ファンクションは、MDA Ref および JOG Ref を除いて、すべての操作モードで、設定データ OSCILL\_IS\_ACTIVE を介して、PLC から選択できます。

#### 設定

下記の設定は、設定データを介して、PLC から制御できます：振動動作の起動および停止、反転位置 1 および 2 の位置、反転位置での停止回数、送り速度、反転位置でのオプション、退避ストロークの回数および停止後の終了位置。

しかし、これらの値は、MMC から直接あるいは NC プログラムを介して、設定データとしてあらかじめ設定することもできます。これらの設定は、電源 ON 後も有効で、PLC も設定データ OSCILL\_IS\_ACTIVE (変数サービスを介して) を直接介して行なう方法で、振動動作を始動することも可能です。

## 補足条件

PLC を介して開始される振動動作を実行する軸として動作させなければならない主軸は、位置決め軸として動作できるようにする条件を満たしていなければなりません。主軸などは、あらかじめ位置制御 (SPOS) に切替えておかねばなりません。

軸は、PLC によって制御されているかどうかに関係なく、VDI インタフェース (DBX28.5,DBX28.6) の 2 とおりの停止ビットに常時反応します。

## ■ 非同期振動中の特殊リアクション

### PLC 制御ありの場合

PLC プログラムは、VDI 信号を介して振動軸の制御を担うことができます。この信号には、プログラム終了、操作モード切替えおよび単一ブロックも含まれます。次のような VDI インタフェース信号は無視されます：送り／主軸停止および NC STOP。

この結果、移動距離削除の場合には、減速要求が抑止されます。

### PLC 制御なしの場合

PLC が軸を制御しない場合は、軸は、非同期振動中標準位置決め軸のように扱われます。

参照： 2.10 補助位置決め軸 (P2)

## 非常停止

EMERGENCY STOP (非常停止) の場合、軸は (サーボイネーブルのキャンセルとフォローアップによって) サーボによって減速されます。振動動作は、終了し、必要であれば再始動しなければなりません。

## 移動距離削除

チャンネル別移動距離削除が無視されます。

軸移動距離削除：

### PLC 制御なしの場合

振動軸が PLC 制御されていない場合、減速カーブによって停止します。

### PLC 制御ありの場合

この場合、減速が抑止し、必ず PLC によって開始される必要があります。次のように、両方の場合に適用されます：軸が停止した後、適切な反転位置にアプローチし、(OSCILL\_CONTROL\_MASK 2.11.4 「データの説明」を参照のこと) 移動距離が削除されます。退避ストロークが OSCILL\_CONTROL\_MASK のように設定されていれば、実行され、終了位置にアプローチします。

振動動作は、このように終了します。

(注)

研削中は、キャリパが軸移動距離削除を介して作動します。

## リセット

振動動作は減速カーブで、中断され、選択解除されます。その後に選択されたオプションは、処理されません ( 退避ストローク、終了ポイントアプローチ )。

## 非同期振動中の特殊リアクション ( 継続 )

### 作業エリア制限, リミットスイッチ

振動動作が有効な制限に違反するということが前処理中に見つかれば、アラームが出力され、振動動作は始動しません。その間に ( 第 2 ソフトウェアリミットスイッチなど ) 起動されていた制限から振動軸がはずれる場合には、振動軸はランプにそって減速され、アラームが出力されます。



注意

保護ゾーンは有効ではありません。

### フォローアップモード

位置決め軸には差がありません。

### プログラム終了

軸が PLC で制御されていない場合には、軸は、振動動作が停止するまでプログラム終了には到達しません (POSA:Positioning beyond block limits の場合のリアクション)。

軸が PLC で制御されている場合には、軸は、プログラム終了後も振動し続けます。

### 操作モード切替え

次の表は、振動を実行できる操作モードを示します。振動を可能にする操作モードへの切替えは、振動動作に影響しません。許容されない運転モードへの切替えは、アラームとなります。NC プログラムからまたはオペレータの入力 (JOG) を介して制御コマンドを実行している間は、振動モードで軸を移動することはできません。移動しようとする、アラームが出力されます。一般的に、最初に始動した動作モードが優先します。

表 2.17 振動を許容する操作モード

操作モード	振動の許容
AUTO	許容される
MDA	許容される
MDA Repos	許容される
MDA Teachin	許容される
MDA Ref	許容されない
JOG	許容される
JOG Ref	許容されない
JOG Repos	許容される

## 単一ブロック処理

軸が PLC で制御されていない場合には、位置決め軸 (POSA) と同じように単一ブロックに反応します。つまり、軸は動作し続けます。

## オーバライド

オーバライドは、VDI interface (VDI インタフェース) で指定されます。

軸オーバライドは、振動軸に作用します。

プログラミング

オーバライドは、位置決め軸と同様に振動軸に作用します。

## ブロックサーチ

ブロックサーチの場合には、( マシンデータ OSCILL\_MODE\_MASK に従って ), NC スタート直後 ( ブロックサーチ後のアプローチ位置へのアプローチ時 ) あるいは、ブロックサーチ ( デフォルト設定 ) 後のアプローチ位置に到達後のどちらかに、最新の有効な振動ファンクションが記憶され、起動します。

OSCILL\_MODE\_MASK ビット 0:

- |   |                     |
|---|---------------------|
| 0 | アプローチ位置に到達後、振動が開始する |
| 1 | NC スタート直後に振動が開始する   |

## REORG

振動が持続するまでに、まず反転位置 1 に必ずアプローチします。

## ASUP

振動動作は、ASUP ( 非同期サブプログラム ) が実行中は、振動動作を持続します。

## ■ 同期化された動作によって制御された振動

### 原則

非同期振動動作は、同期化されたアクションを介して送り動作と結合され、それに合わせて制御されます。

参照： /FB/,S5,"Synchronized Actions"

次の説明は、振動機能と関連付けられたモーション同期動作について述べたものです。

### 機能

次の機能の合成したものを、下記に記載した言語ツールを用いて実装できます。

1. 反転位置での送り ( 「■反転位置 1 または 2 での送り」 参照 )
2. 反転位置範囲内での送り ( 「■反転位置範囲での送り」 参照 )
3. 両方の反転位置での送り ( 「■両方の反転位置での送り」 参照 )
4. 送りが終了するまで、反転位置で停止振動動作 ( 「■反転位置での振動動作の停止方法」 参照 )

6. 部分送りが早まって始動しないようにする（「**■**パーシャル送りを早まって始動させないようにする」参照）

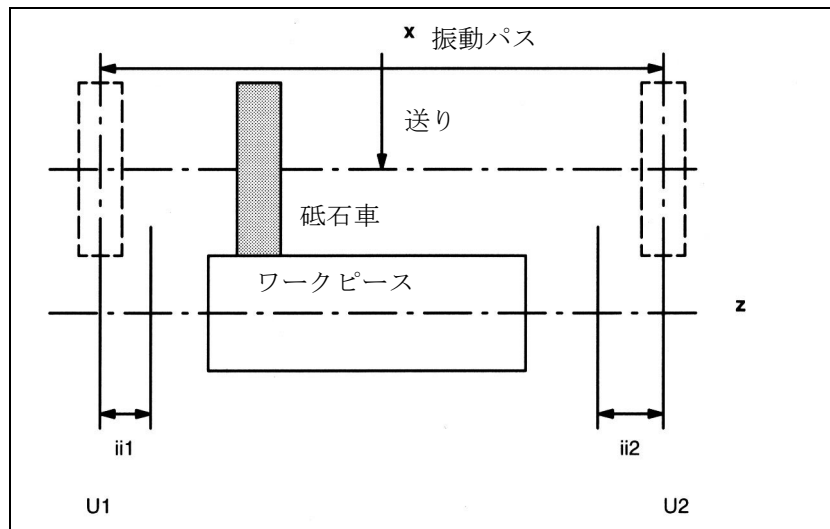


図 2.94 振動軸と送り軸の配置と条件

凡例：U1      反轉位置 1

U2 反転位置 2

ii1 反轉範圍 1

ii2 反轉範圍 2

# プログラミング

振動を決定するパラメータ (「**■非同期振動への影響**」参照) は、送り軸と振動軸 (「**■振動と送り軸の OSCILL の割当て**」参照) との割当てを含む動作ブロック、送り定義 (POSP) およびモーション同期アクションの前に、定義しておかなくてはなりません。

振動軸は、WAITP[oscillation axis] (MD \$MA\_AUTO\_GET\_TYPE 参照) を介してイネーブルされると、振動パラメータを、設定データに同時に送ることができます。記号名、たとえば \$SA\_REVERSE\_POS1 は、モーション同期アクションをプログラムするために用いることができます。

(注)

`SSA_REVERSE_POS` という値でのモーション同期アクションの場合、通訳時間での比較値が有効です。設定データがそれに従って修正される場合には影響しません。

\$\$\$A\_REVERSE\_POS という値を持つモーション同期アクションの場合は、補間内での比較値が有効です。これによって、修正された反転位置へ確実にリアクションします。

- ・ モーション同期条件 WHEN, WHENEVER
- ・ モーションブロックからの起動
  - 振動軸と送り軸を互いに OSCILL に割当てる
  - 送りレスポンス POSP を指定

これまでに述べられていない要素は、次のセクションでさらに詳しく説明します。

2.11.6 「例」では、いくつかの例について述べています。

(注)

モーション同期アクション (WHEN と WHENEVER) が定義された条件が有効でなくなった場合は、前にオーバーライドが 0% に設定されていれば、この条件のためのオーバーライドは自動的に 100% に設定されます。

## メインラン評価

ソフトウェアバージョン 3.2 以降、メインランの補間サイクルでの同期化条件と実際値 ( 比較条件の右側の変数 \$\$) を比較できます。標準的なシステム変数の比較であれば、最初の運転で式が評価されます。同期化アクションの全面的な拡張可能性は、次を参照してください。

参照： /FB/,S5."Motion-synchronous actions"

例 1

振動，設定データを介して，確実に設定された反転位置：

\$\$SA\_OSCILL\_REVERSE\_POS1[Z]=-10

\$\$SA\_OSCILL\_REVERSE\_POS[Z]=10

G0 X0 Z0

WAITP(Z)

ID=1 WHENEVER \$AA\_IM[Z]<\$\$SA\_OSCILL\_REVERSE\_POS1[Z] DO

\$AA\_OVR[X]=0

ID=2 WHENEVER \$AA\_IM[Z]>\$\$SA\_OSCILL\_REVERSE\_POS2[Z] DO

\$AA\_OVR[X]=0

；振動軸の実際値が反転位置を

；超えた場合は，

；送り軸が停止する。

OS[Z]=1 FA[X]=1000 POS[X]=40 ；振動オン

OS[Z]=0 ；振動オフ

M30

例 2

反転位置のオンライン変更を行う振動，すなわち，ユーザ表面を介して反転位置 1 の修正があれば，アクティブな振動動作と共に即時考慮されます。

\$\$SA\_OSCILL\_REVERSE\_POS1[Z]=-10

\$\$SA\_OSCILL\_REVERSE\_POS2[Z]=10

G0 X0 Z0

WAITP(Z)

ID=1 WHENEVER \$AA\_IM[Z]<\$\$SA\_OSCILL\_REVERSE\_POS1[Z] DO

\$AA\_OVR[X]=0

ID=2 WHENEVER \$AA\_IM[Z]>\$\$SA\_OSCILL\_REVERSE\_POS2[Z] DO

\$AA\_OVR[X]=0

；振動軸の実際値が反転位置を

；超えた場合は，



---

```
OS[Z]=1 FA[X]=1000 POS[X]=40 ; 送り軸が停止する。
OS[Z]=0 ; 振動オン
OS[Z]=0 ; 振動オフ
```

```
M30
```

## ■ 反転位置 1 または 2 での送り

### ファンクション

振動軸が、反転位置に到達しない限り、送り軸は移動しません。

### アプリケーション

反転位置での直接送り

### プログラミング

反転位置 1 の場合：

```
WHENEVER $AA_IM[Z]<>$SA_OSCILL_REVERSE_POS1[Z] DO
$AA_OVR[X]=0 $AA_OVR[Z]=100
```

反転位置 2 の場合

```
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]	機械座標系での振動軸 Z の実際位置
\$SA_OSCILL_REVERSE_POS1[Z]	振動軸の反転位置 1 の位置
\$AA_OVR[X]	送り軸の軸オーバーライド
\$AA_OVR[Z]	振動軸の軸オーバーライド

キーワードの説明：

```
WHENEVER ...DO... 条件が満たされれば必ず...
```

### 送り

送り絶対値が命令 POSP によって定義されます。

「■送り POSP の定義」を参照のこと。

### 割当て

振動軸と送り軸との間の割当ては、命令 OSCILL によって定義されます。

「■振動と送り軸の OSCILL の割当て」を参照のこと。

## ■ 反転位置範囲での送り

### ファンクション

反転位置範囲 1:

振動軸が反転位置範囲 ( 変数 ii1 の内容を加えた反転位置 1 での位置に ) に到達していなければ、送りは作動しません。これは、反転位置 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。そうでない場合には、条件をそれに応じて変更しなければなりません。

### アプリケーション

反転位置範囲 1:

同期化アクションの目的は、振動動作が反転位置範囲 1 に到達するまで、送り動作を開始しないようにすることです。図 2.94 を参照。

### プログラミング

反転位置範囲 1 の場合 :

```
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]	振動軸 Z の実際位置
\$SA_OSCILL_REVERSE_POS1[Z]	振動軸の反転位置 1 の位置
\$AA_OVR[X]	送り軸の軸オーバーライド
ii1	反転範囲のサイズ ( ユーザ変数 )

キーワードの説明 :

WHENEVER ...DO...                      条件が満たされれば必ず...

### ファンクション

反転位置範囲 2 の場合 :

振動軸の現行の位置の値 が、変数 ii2 の内容を引いた反転位置 2 での位置より小さくなるまで、送り軸は停止します。このことは、反転位置の位置 2 の設定が、反転位置 1 の設定より高いという条件で適用されます。そうでなければ、それに応じて、条件を変更しなければなりません。

### アプリケーション

反転位置範囲 2:

同期化アクションの目的は、振動動作が反転位置範囲 2 に到達するまで、送り動作を開始しないようにすることです。図 2.94 を参照。

## プログラミング

反転位置 2 の場合：

```
WHENEVER $AA_IM[Z]<$SA_OSCILL_REVERSE_POS1[Z] - ii2 DO  
$AA_OVR[X]=0
```

システム変数の説明：

\$AA_IM[Z]	振動軸 Z の実際位置
\$SA_OSCILL_REVERSE_POS2[Z]	振動軸の反転位置 2 の位置
\$AA_OVR[X]	送り軸の軸オーバーライド
ii2	反転位置範囲 2 のサイズ ( ユーザ変数 )

## 送り

送り絶対値は命令 POSP によって定義されます。

「■送り POSP の定義」を参照のこと。

## 割当て

振動軸と送り軸との間の割当ては、命令 OSCILL によって定義されます。

「■振動と送り軸の OSCILL の割当て」を参照のこと。

## ■ 両方の反転位置での送り

### 原則

反転位置点と反転位置範囲での送りについて上記で述べたファンクションは組合せが自由です。

### 組合せ

送り：

点 U1	点 U2
点 U1	範囲 U2
範囲 U1	点 U2
範囲 U1	範囲 U2

### 片面送り

点 U1  
点 U2  
範囲 U1  
範囲 U2

これらのオプションについては、「■反転位置 1 または 2 での送り」と「■反転位置範囲での送り」に記載しています。

## ■ 反転位置での振動動作の停止方法

### ファンクション

反転位置 1:

振動軸は、反転位置 1 に到達するたびに、オーバーライド停止し、送り動作を開始しなければなりません。

### アプリケーション

同期化アクションによって、パート送りが実行されるまで、振動軸は静止した状態になります。パート送りが実行されるまで、反転位置 1 で待機する必要がなければ、この同期化アクションは省略できます。同時に、この同期化アクションは、前の同期化アクション(まだアクティブのまま)によって停止していれば、送り動作を始動するために用いることができます。

### プログラミング

```
WHENEVER $$AA_IM[oscillation axis]==  
$SA_OSCILL_REVERSE_POS1[oscillation axis]  
DO $AA_OVR[oscillation axis]=0 $AA_OVR[infeed axis]=100
```

システム変数の説明:

\$AA_IM[oscillation axis]	振動軸の現行位置
\$SA_OSCILL_REVERSE_POS1[oscillation axis]	振動軸の反転位置 1
\$AA_OVR[oscillation axis]	振動軸の軸オーバーライド
\$AA_OVR[infeed axis]	送り軸の軸オーバーライド

### ファンクション

反転位置 2:

振動軸は、反転位置 2 に到達するたびに、オーバーライド 0 で停止し、送り動作を始動しなければなりません。

### アプリケーション

同期化アクションによって、パート送りが実行されるまで、振動軸は静止した状態になります。振動軸は、パート送りが実行されるまで、反転位置 2 で待つ必要がなければ、この同期化されたアクションは省略できます。同時に、この同期化アクションは、前の同期化アクション(まだアクティブのまま)によって停止していれば、送り動作を始動するために用いることができます。

## プログラミング

```
WHENEVER $AA_IM[oscillation axis]==  
$SA_OSCILL_REVERSE_POS2[oscillation axis]  
DO $AA_OVR[oscillation axis]=0 $AA_OVR[infeed axis]=100
```

システム変数の説明：

\$AA_IM[oscillation axis]	振動軸の現行位置
\$SA_OSCILL_REVERSE_POS2[oscillation axis]	振動軸の反転位置 2
\$AA_OVR[oscillation axis]	振動軸の軸オーバーライド
\$AA_OVR[infeed axis]	送り軸の軸オーバーライド

### ■ 振動動作の再始動

#### ファンクション

送り軸 = 0 の最新で移動されたパスの移動距離すなわちパート送りが実行されると必ず、振動軸がパスオーバーライドを介してスタートします。

#### アプリケーション

同期化アクションの目的は、パート送り動作の完了時に振動軸の動作を継続することです。振動軸がパーシャル送りの完了を待つ必要がない場合には、振動軸が反転位置で停止する状態のモーション同期アクションは省略しなければなりません。

## プログラミング

```
WHENEVER $AA_DTEPW[infeed axis] == 0 DO  
$AA_OVR[oscillation axis]= 100
```

システム変数の説明：

\$AA_DTEPW[infeed axis]	ワークピース座標系の送り軸の 軸移動距離 送り軸のパスセクション
\$AA_OVR[oscillation axis]	振動軸の軸オーバーライド

キーワードの説明：

WHENEVER ...DO... 条件が満たされれば必ず…,

### ■ パーシャル送りを早まって始動させないようにする

#### ファンクション

上記のファンクションは、反転位置外または反転位置範囲外での送り動作をすべて止めます。送り動作の完了時、次のパーシャル送りの再始動は止められなければなりません。

## アプリケーション

チャンネル別フラグは次の目的で使用されます。フラグは、パーシャル送りの最後に設定され (パーシャル移動距離 == 0), 軸が反転位置範囲を出るときに削除されます。次の送り動作は止められます。

## プログラミング

```
WHENEVER $$AA_DTEPW[infeed axis] == 0 DO  
$AC_MARKER[index]=1
```

および、たとえば、反転位置 1 の場合：

```
WHENEVER $AA_IM[Z]<>$SA_OSCILL_REVERSE_POS1[Z]DO  
$AC_MARKER[index]=0  
WHENEVER $AC_MARKER[index]==1 DO $AA_OVR[infeed axis]=0
```

システム変数の説明：

\$AA_DTEPW[infeed axis]	ワークピース座標系の送り軸の 軸移動距離 送り軸のパスセクション
\$AC_MARKER[index]	インデックス付きチャンネル別フラグ
\$AA_IM[oscillation axis]	振動軸の現行位置
\$SA_OSCILL_REVERSE_POS1[oscillation axis]	振動軸の反転位置 1
\$AA_OVR[infeed axis]	送り軸の軸オーバーライド

キーワードの説明：

WHENEVER ...DO... 条件が満たされれば必ず…,

## ■ 振動と送り軸の OSCILL の割当て

### ファンクション

コマンド OSCILL を用いて、1 つ以上の送り軸を振動軸に割当てます。振動動作が始動します。

PLC は、VDI インタフェースを介して軸が割当てられたということを通知されます。PLC が振動軸を制御している場合は、送り軸を監視し、送り軸の信号を使って、インタフェースの 2 停止ビットを介して、振動軸へのリアクションを起こします。

## アプリケーション

応答がすでに同期条件によって定義されている軸は、振動モードを起動するため、互いに割当てられます。振動動作が始動します。

## プログラミング

OSCILL[oscillation axis]=(infeed axis1, infeed axis2, infeed axis3)

括弧内の送り軸 2 と送り軸 3 にデリミッターを加えることが必要でない場合は省略することができます。

### ■ 送り POSP の定義

#### ファンクション

送り軸についての下記のデータを制御装置が受取ります。

- 全体送り
- 反転位置点／反転位置範囲でのパート送り
- 終了時のパートフィード応答

#### アプリケーション

この命令は、OSCILL で振動を起動後、反転位置点／反転位置範囲で必要な送り値の制御を知らせるために必ず与えられます。

## プログラミング

POSP[infeed axis]=(end position, part section, mode)

End position

( 終了位置 )                      すべてのパート送りがトラバースされた後、送り軸のための終了位置

Part section

( パートセクション )      反転位置／反転位置範囲でのパート送り

Mode ( モード )

0

最後 2 つのパートステップについては、目標ポイントまでの残りのパスが、大きさの等しい 2 つの残りのステップに分けられる ( デフォルト設定 )。

Mode

1

パート長さは、すべての計算されたパート長さ全体が目標ポイントまでのパスにぴったり合うように調節される。

### 2.11.3 補足条件

#### "Oscillation" ファンクションの可用性

振動は、オーダ番号 6FC5 251-0AB04-0AA0 のオプションです。非同期振動とモーダル振動は、NCU570, 571, 572, 573 の SW2 から使用できます。

モーション同期アクションでの振動は、NCU の 572 と 573 で利用できます。

### 2.11.4 データの説明 (MD,SD)

#### ■ マシンデータ

11460 MD 番号	OSCILL_MODE_MASK 非同期振動のモードスクリーン形式	
初期設定: 0	最小入力リミット: 0	最大入力リミット: 0xFFFF
変更が有効になるための条件: 電源オン	保護レベル: 2/7	単位: -
データタイプ: DWORD	適用開始 SW バージョン: 2.0	
意味:	ビット 0 値 1 ブロックサーチの場合、振動動作が処理中のプログラムセクションです で起動しているのであれば、NC スタート直後、すなわちアプローチ 位置へのアプローチ中に、振動動作が開始します。 値 0 アプローチ位置に到達するまで、振動動作は開始しません。	

#### ■ 設定データ

#### 軸別／スピンドル別データ

43700 SD 番号	OSCILL_REVERSE_POS[axis] 振動反転位置 1	
初期設定: 0	最小入力リミット: ***	最大入力リミット: ***
変更が有効になるための条件: 即時	単位: mm, 度	
データタイプ: DOUBLE	適用開始 SW バージョン: 2.0	
意味:	反転位置 1 での振動軸の位置	
アプリケーション	NC 言語: OSP1[axis]=position	
関連性	OSCILL_REVERSE_POS2	

43710 SD 番号	OSCILL_REVERSE_POS2[axis] 振動反転位置 2	
初期設定: 0	最小入力リミット: ***	最大入力リミット: ***
変更が有効になるための条件: 即時	単位: mm, 度	
データタイプ: DOUBLE	適用開始 SW バージョン: 2.0	
意味:	反転位置 2 での振動軸の位置	
アプリケーション	NC 言語: OSP2[axis]=position	
関連性	OSCILL_REVERSE_POS1	



43720 SD 番号	OSCILL_DWELL_TIME[axis] 振動反転位置 1 での停止時間	
初期設定 : 0	最小入力リミット : -2	最大入力リミット : ***
変更が有効になるための条件 : 即時	単位 : s	
データタイプ : DOUBLE	適用開始 SW バージョン : 2.0	
意味 :	反転位置 1 での振動軸の停止時間	
アプリケーション	NC 言語 : OST2[axis]=time	
関連性	OSCILL_DWELL_TIME2	

43730 SD 番号	OSCILL_DWELL_TIME2[axis] 振動反転位置 2 での停止時間	
初期設定 : 0	最小入力リミット : -2	最大入力リミット : ***
変更が有効になるための条件 : 即時	単位 : s	
データタイプ : DOUBLE	適用開始 SW バージョン : 2.0	
意味 :	反転位置 2 での振動軸の停止時間	
アプリケーション	NC 言語 : OST2[axis]=time	
関連性	OSCILL_DWELL_TIME2	

43740 SD 番号	OSCILL_VELO[axis] 振動軸の送り速度	
初期設定 : 0	最小入力リミット : ***	最大入力リミット : ***
変更が有効になるための条件 : 即時	単位 : mm/min rev/min	
データタイプ : DOUBLE	適用開始 SW バージョン : 2.0	
意味 :	振動軸の送り速度	
アプリケーション	NC 言語 : FA[axis]=Fvalue F 値	

43750 SD 番号	OSCILL_NUM_SPARK_CYCLES[axis] 退避ストローク数	
初期設定 : 0	最小入力リミット : 0	最大入力リミット : ***
変更が有効になるための条件 : 即時	単位 : 1	
データタイプ : DWORD	適用開始 SW バージョン : 2.0	
意味 :	振動動作の完了時に実行される退避ストローク数	
アプリケーション	NC 言語 : OSNSC[axis]=number of strokes ストローク数	

43760 SD 番号	OSCILL_END_POS[axis] 振動軸の終了位置	
初期設定 : 0	最小入力リミット : ***	最大入力リミット : ***
変更が有効になるための条件 : 即時	単位 : mm, 度	
データタイプ : DOUBLE	適用開始 SW バージョン : 2.0	
意味 :	退避ストローク実行後振動軸がアプローチする位置	
アプリケーション	NC 言語 : OSE[axis]=position	

43770 SD 番号	OSCILL_CTRL_MASK[axis] 振動シーケンス制御スクリーン形式	
初期設定 : 0	最小入力リミット : -	最大入力リミット : -
変更が有効になるための条件 : 即時	単位 : -	
データタイプ : BYTE	適用開始 SW バージョン : 2.0	
意味 :	ビットスクリーン形式, 下記の表 2.18 参照。	
アプリケーション	NC 言語 : OSCTRL[axis]=(setting options, resetting options)	

表 2.18 スクリーン形式 OSCILL\_CTRL\_MASK におけるビットの意味

ビット番号	OSCILL_CTRL_MASK での意味
0,1	0: 振動動作の停止時に次の反転位置で停止 1: 振動動作の停止時に反転位置 1 で停止 2: 振動動作の停止時に反転位置 2 で停止 3: 退避ストロークがプログラムされていない場合, 振動動作の停止時に, 反転位置にアプローチしない
2	1: 次の退避後の終了位置にアプローチ
3	1: 振動動作が移動距離削除によって強制終了される場合は, 必ず, 退避ストロークが実行され, 終了位置にアプローチしなければならない (プログラムされている場合)
4	1: 振動動作が移動距離削除によって強制終了される場合は, 停止のため, 適切な反転位置にアプローチする
5	1: 新しい送り速度設定は, 次の反転位置まで有効ではない
6	1: 送り速度設定が 0 の場合, パスオーバーレイもしくは速度オーバーレイがアクティブになる
7	1: 回転軸 DC の場合 (最短パス)

43780 SD 番号	OSCILL_CTRL_MASK[axis] 振動動作を起動	
初期設定 : 0	最小入力リミット : 0	最大入力リミット : 1
変更が有効になるための条件 : 即時	単位 : -	
データタイプ : BOOLEAN	適用開始 SW バージョン : 2.0	
意味 :	振動動作の起動および停止	
アプリケーション	NC 言語 : OS[axis]=1 OS[axis]=0	

## 2.11.5 信号の説明

### VDI 入力信号

PLC ユーザプログラムは、下記の信号を使って、振動プロセスを制御します。

DB31, ... DBX2.3 データブロック	Set reversal point ( 反転位置の設定 ) 軸／スピンドルへの信号	
エッジ評価：なし	信号更新：周期的	信号が有効となる最初の SW バージョン：2.0
信号ステータス 1 あるいは 信号遷移 0 ---> 1	反転位置 2	
信号ステータス 0 あるいは 信号遷移 1 ---> 0	反転位置 1	

DB31, ... DBX28.4 データブロック	Alter reversal point ( 反転位置の変更 ) 送り軸／主軸への信号	
エッジ評価：なし	信号更新：周期的	信号が有効となる最初の SW バージョン：2.0
信号ステータス 1 あるいは 信号遷移 0 ---> 1	選択された反転位置は、手動トラバースで変更できます。	
信号ステータス 0 あるいは 信号遷移 1 ---> 0	選択された反転位置は、手動トラバースでは変更できません。	
関連性	DBX28.3	

DB31, ... DBX28.5 データブロック	Stop at next reversal point ( 次の反転位置での停止 ) 送り軸／主軸への信号	
エッジ評価：なし	信号更新：周期的	信号が有効となる最初の SW バージョン：2.0
信号ステータス 1 あるいは 信号遷移 0 ---> 1	振動動作は、次の反転位置で中断されます。	
信号ステータス 0 あるいは 信号遷移 1 ---> 0	振動動作は、次の反転位置後に続きます。	
関連性	DBX28.6, DBX28.7	

DB31, ... DBX28.6 データブロック	Stop along braking ramp ( ブレーキングランプに沿って停止 ) 送り軸／主軸への信号	
エッジ評価：なし	信号更新：周期的	信号が有効となる最初の SW バージョン：2.0
信号ステータス 1 あるいは 信号遷移 0 ---> 1	軸がランプに沿って減速し、振動動作が中断します。	
信号ステータス 0 あるいは 信号遷移 1 ---> 0	振動動作は、中断しないで連続します。	
関連性	DBX28.5, DBX28.7	

DB31, ... DBX28.7 データブロック	PLC controls axis (PLC 制御軸 ) 送り軸／主軸への信号	
エッジ評価：なし	信号更新：周期的	信号が有効となる最初の SW バージョン：2.0
信号ステータス 1 あるいは 信号遷移 0 ---> 1	軸は、PLC によって制御されます。 インタフェース信号に対するリアクションが、2 停止ビットを使って、PLC で制御され、減速アクションの他の信号は無視されます。	

DB31, ... DBX28.7 データブロック	PLC controls axis (PLC 制御軸) 送り軸／主軸への信号
信号ステータス 0 あるいは 信号遷移 1 ---> 0	軸は、PLC で制御されません。
関連性	DBX28.5, DBX28.6

## VDI 出力信号

NCK によって、下記の信号は、PLC ユーザプログラムに使用できるようになります。

DB31, ... DBX100.3 データブロック	Oscillation cannot start ( 振動は始動できない ) 送り軸／主軸からの信号
エッジ評価：なし	信号更新：周期的      信号が有効となる最初の SW バージョン：2.1
信号ステータス 1 あるいは 信号遷移 0 ---> 1	振動軸は、プログラミングが不正確であるため始動できません。このステータスは、軸が移動してからでも発生する場合があります。
信号ステータス 0 あるいは 信号遷移 1 ---> 0	振動動作が始動可能です。

DB31, ... DBX100.4 データブロック	Error during oscillation movement ( 振動動作中のエラー ) 送り軸／主軸からの信号
エッジ評価：	信号更新：更新      信号が有効となる最初の SW バージョン：2.1
信号ステータス 1 あるいは 信号遷移 0 ---> 1	振動動作は、すでに強制終了しました。
信号ステータス 0 あるいは 信号遷移 1 ---> 0	振動動作は、正しく実行中です。

DB31, ... DBX100.5 データブロック	Sparking-out active ( 退避アクティブ ) 送り軸／主軸からの信号
エッジ評価：なし	信号更新：周期的      信号が有効となる最初の SW バージョン：2.1
信号ステータス 1 あるいは 信号遷移 0 ---> 1	軸は、退避ストロークを実行中です。
信号ステータス 0 あるいは 信号遷移 1 ---> 0	軸は、退避ストロークを現時点では実行していません。
関連性	DBX100.7

DB31, ... DBX100.6 データブロック	Oscillation movement active ( 振動動作アクティブ ) 送り軸／主軸からの信号
エッジ評価：なし	信号更新：周期的      信号が有効となる最初の SW バージョン：2.1
信号ステータス 1 あるいは 信号遷移 0 ---> 1	軸は、反転位置 2 点間での振動動作を実行中です。
信号ステータス 0 あるいは 信号遷移 1 ---> 0	軸は、現時点は振動していません。
この信号は次の場合は無意味	DBX100.7=0
関連性	DBX100.7

DB31, ... DBX100.7 データブロック	Oscillation active ( 振動アクティブ ) 送り軸／主軸からの信号	
エッジ評価 : なし	信号更新 : 周期的	信号が有効となる最初の SW バージョン : 2.1
信号ステータス 1 あるいは 信号遷移 0 ---> 1	軸は、振動軸として現在移動中です。	
信号ステータス 0 あるいは 信号遷移 1 ---> 0	軸は、位置決め軸です。	
関連性	DBX100.5, DBX100.6	

DB31, ... DBX104.0-7 データブロック	Active infeed axes ( アクティブ送り軸 ) 送り軸／主軸からの信号	
エッジ評価 : なし	信号更新 : 周期的	信号が有効となる最初の SW バージョン : 2.1
信号ステータス 1 あるいは 信号遷移 0 ---> 1	信号を送る軸はその時点での振動軸であり、それがこのフィールドでのアクティブな送り軸であることを示します (104.0 軸 1 が送り軸である、104.1 軸 2 が送り軸である、等 )。	
信号ステータス 0 あるいは 信号遷移 1 ---> 0	関連付けられた軸は、送り軸ではありません。	
関連性	DBX100.7	

## 2.11.6 例

### 前提条件

次に示す例には、下記のタイトルの項で指定された NC 言語の構成要素が必要になります。

- 非同期振動
- 同期化されたアクションによって制御された振動

### ■ 同期振動の例

#### タスク

振動軸 Z は、-10 と 10 の間で振動しなければなりません。粗正確停止で反転位置にアプローチし、正確停止でない場合は、反転位置 2 にアプローチします。振動軸フィードレートは 5000 でなければなりません。3 退避ストロークが、加工操作の終わりに実行されてから、その後に、終了位置 30 に振動軸がアプローチしなければなりません。送り軸のフィードレートが 1000 であり、X 方向の送りのエンドは 15 です。

#### プログラムセクション

```
OSP1[Z]=-10      ; 反転位置 1
OSP2[Z]=10       ; 反転位置 2
OST1[Z]=-1       ; 反転位置 1 での停止時間：正確停止粗
OST2[Z]=-2       ; 反転位置 2 での停止時間：正確停止なし
FA[Z]=5000       ; 振動軸のフィードレート
OSNSC[Z]=3       ; 3 退避ストローク
OSE[Z]=30        ; 終了位置
OS[Z]=1 F500X15  ; 振動始動、フィードレート 5000 の送り X 軸,
                  ; 送り目標 15
```

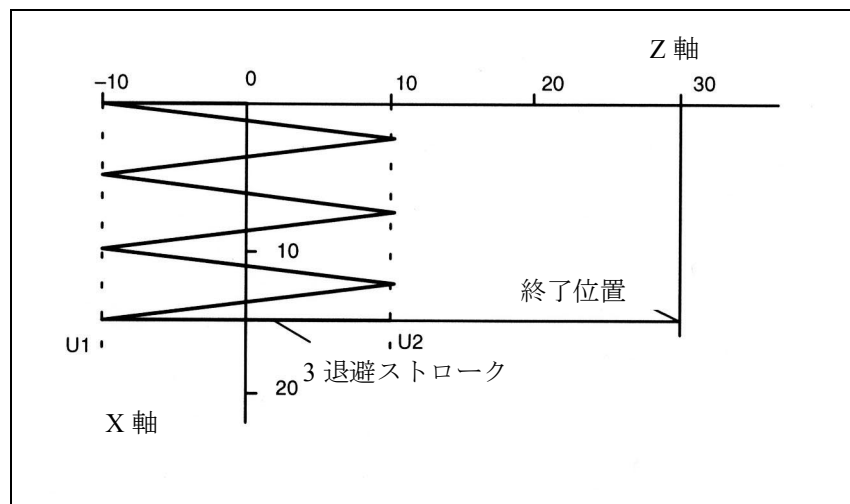


図 2.95 振動動作と送りのシーケンス，例 1

## ■ アクションの同期化された振動の例 1

### タスク

直接送りは 必ず反転位置 1 で起こります。振動軸は、パート送りが実行されて、移動し続けることができるまで待機しなければなりません。反転位置 2 では、反転位置 2 から -6 の距離で送りが起こります。振動軸は、パート送りが実行されるまで反転位置で決して待機してはなりません。軸 Z は振動軸で、軸 X は送り軸です (2.11.2 「■ PLC 制御下の非同期振動」を参照)。

(注)

設定データ OSCILL\_REVERSE\_POS1/2 は、マシン座標系における値です；したがって、比較は、\$AA\_IM[n] にのみ適しています。

### プログラムセクション

；例 1: 同期化されたアクションの振動

```
OSP1[Z]=10 OSP2[Z]=60 ;反転位置 1 と 2 を定義する
OST1[Z]=-2 OST2[Z]=0 ;反転位置 1: 正確停止なし
;反転位置 2: 正確停止微
FA[Z]=5000 FA[X]=250 ;振動軸フィードレート,
;送り軸フィードレート
OSCTRL[Z]=(1+8+16,0) ;反転位置 1 での振動動作を停止する
;移動距離削除後退避
;および終了位置にアプローチ
;移動距離削除後に
;プログラムされた反転位置にアプローチ
OSNSC[Z]=3 ;3 退避ストローク
OSE[Z]=0 ;終了位置 =0;
WAITP(Z) ;Z 軸 の振動をイネーブルする
;モーション同期アクション
;
; Whenever マシン座標系での振動軸の現在位置のときは必ず
; is not equal to 反転位置 1 には等しくない
; then この場合、インデックス 1 のフラグを値 0 に設定する
(リセットフラグ 1)
; WHENEVER $AA_IM[Z]<>$SA_OSCILL_REVERSE_POS1[Z]DO
$AC_MARKER[1]=0
;
; Whenever マシン座標系での振動軸の現在位置のときは必ず
; is lower than 反転位置範囲 2 の始め
(ここでは: 反転位置 2 ~ 6) より小さい
; then この場合、送り軸の軸オーバーライドを
0% に設定する
; and さらに、インデックス 2 のフラグを値 2 に設定する
(フラグ 2 をリセットする)
WHENEVER $AA_IM[Z]<$SA_OSCILL_REVERSE_POS2[Z]-6 DO $AA_OVR[X]=0
```

```

$AC_MARKER[2]=0
;
; Whenever      機械座標系での振動軸の現行位置のときは必ず
; is the same as 反転位置 1 と同じ
; then          この場合、振動軸の軸オーバーライドを 0% に設定する
; and           さらに、送り軸の軸オーバーライドを
                100% に設定する ( すなわち、前述の同期化アクションを
                キャンセルすること )
WHENEVER $AA_IM[Z]==$SA_OSCILL_REVERSE_POS1[Z]DO $AA_OVR[Z]=0
$AA_OVR[X]=100
;

; Whenever      パート送りの移動距離は必ず
; is equal to    0 に等しい
; then          この場合、インデックス 2 のフラグは 1 の値 に設定する
; and           さらに、インデックス 1 のフラグを 1 の値に設定する
WHENEVER $AA_DTEP[X]==0 DO $AC_MARKER[2]=1 $AC_MARKER[1]=1
;
; Whenever      インデックス 2 のフラグ
; is equal to    1 に等しい、
; then          この場合、早まった送りをさせないように
                送り軸の軸オーバーライドを 0% に設定する
WHENEVER $AC_MARKER[2]==1 DO $AA_OVR[X]=0
;
; Whenever      インデックス 1 のフラグは必ず
; is equal to    1 に等しい、
; then          この場合、早まった送りをさせないように
                送り軸の軸オーバーライドを 0% に設定する
                ( 振動軸は反転位置範囲 2 からまだ出ていない )
; and           および、振動軸の軸オーバーライドを 100% に設定する
                ( 振動を ' スタート ' )
WHENEVER $AC_MARKER[2]==1 DO $AA_OVR[X]=0 $AA_OVR[Z]=100
;

; If            マシン座標系での振動軸の現在位置の場合、
; is equal to    反転位置 1 に等しい、
; then          振動軸の軸オーバーライドを 100% に設定する
; and           送り軸の軸オーバーライドを 0% に
                設定する ( 第 2 の同期化されたアクションをいったん
                キャンセルするため )。
WHEN $AA_IM[Z]==$SA_OSCILL_REVERSE_POS1[Z]DO $OVR[Z]=100
$AA_OVR[X]=0
;
;-----
OSCILL[Z]=(X) POSP[X]=(5,1,1) ; 送り軸として軸 X を割当てる

```



```

;振動軸 Z に対して;軸 X は,
;1 のパートステップで, 終了位置 5 まで
;必ず送りをし, すべてのパート長さの
;合計は終了位置に必ずぴったりと
;合わねばならない。

;
M30                                ;プログラム終了

```

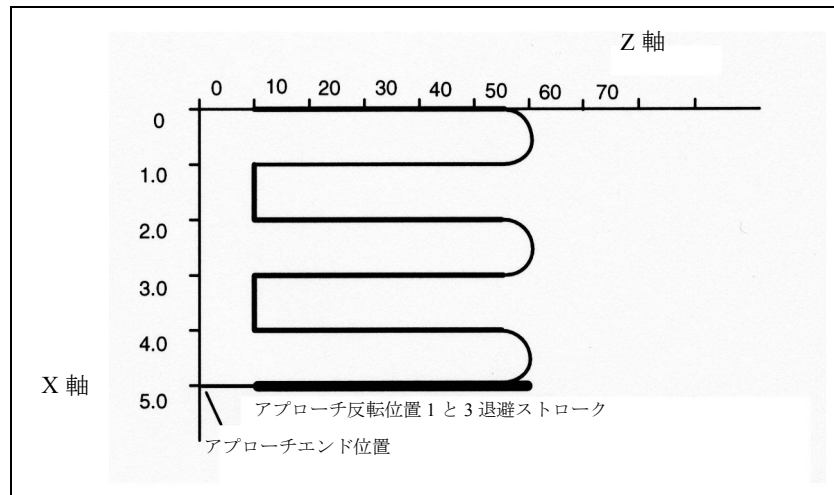


図 2.96 振動動作と送りのシーケンス, 例 1

## ■ 同期化されたアクションでの振動の例 2

### タスク

送りは反転位置 1 では起こりません。反転位置 2 では, 反転位置 2 から ii2 の距離で送りが起こらなければなりません。つまり振動軸は, パート送りが実行されるまで, この反転位置で待機しなくてはなりません。軸 Z は振動軸で, 軸 X は送り軸です。

### プログラムセクション

例 2: アクションが同期化された振動

```

DEF INT ii2                                ;反転位置範囲 2 の変数を定義する
;
OSP1[Z]=10 OSP2[Z]=60                    ;反転位置 1 と 2 を定義する
OST1[Z]=0 OST2[Z]=0                      ;反転位置 1: 微正確停止
                                           ;反転位置 2: 微正確停止
FA[Z]=5000 FA[X]=100                     ;振動軸の送り速度,
                                           ;送り軸の送り速度
OSCTRL[Z]=(2+8+16.1)                     ;反転位置 2 での振動動作を停止する
                                           ;移動距離削除後, 退避と
                                           ;アプローチエンド位置
                                           ;移動距離削除後, 適切な反転位置に
                                           ;アプローチする

```

```

OSNSC[Z]=3          ; 3 退避ストローク
OSE[Z]=70           ; エンド位置 =70;
ii2=2               ; 反転位置範囲を設定する
WAITP(Z)            ; Z 軸の振動をイネーブルする

; Motion-synchronous actions: (モーション同期アクション)
; Whenever          マシン座標系での振動軸の現行位置は必ず
; is lower than     反転位置範囲 2 のスタートより小さい
; then              この場合に、送り軸の
                    軸オーバーライドを 0% に設定する
; and               および インデックス 0 のフラグを値 0 に設定する
WHENEVER $AA_IM[Z]<$SA_OSCILL_REVERSE_POS2[Z]-ii2 DO $AA_OVR[X]=0
$AC_MARKER[0]=0
;
; Whenever          機械座標系での振動軸の現在位置は必ず
; is equal to or greater than reversal position 2, 反転位置 2 以上
; then              この場合、振動軸の軸オーバーライドを
                    0% に設定する
WHENEVER $AA_IM[Z]>=$SA_OSCILL_REVERSE_POS2[Z]DO $AA_OVR[Z]=0
;
; Whenever          パート送りの移動距離は必ず、
; is equal to       0 に等しい
; then              この場合、インデックス 0 のフラグを 1 の値に設定
                    する
WHENEVER $AA_DTEPW[X]==0 DO $AC_MARKER[0]=1
;
; Whenever          インデックス 0 のフラグは必ず
; is equal to       1 に等しい
; then              早まった送りをさせないようにするため、
                    送り軸の軸オーバーライドを 100% に
                    設定する（振動軸は、反転位置 2 からは
                    まだ出ていない、送り軸は再度
                    送りをするレディ状態である）
and                および 振動軸の軸オーバーライドを 100% に
                    設定する（これで、第 2 の同期化アクションを
                    キャンセルする）
WHENEVER $AC_MARKER[0]==1 DO $AA_OVR[X]=0 $AA_OVR[Z]=100
;
OSCILL[Z]=(X)POSP[X]=(5,1,1) ; 軸をスタートする
                                ; 送り軸として、
                                ; 振動軸 Z に軸 X が割当てられる
                                ; 軸 X はエンド位置に必ず移動する
                                ; 1 のステップでは 5
;
M30

```

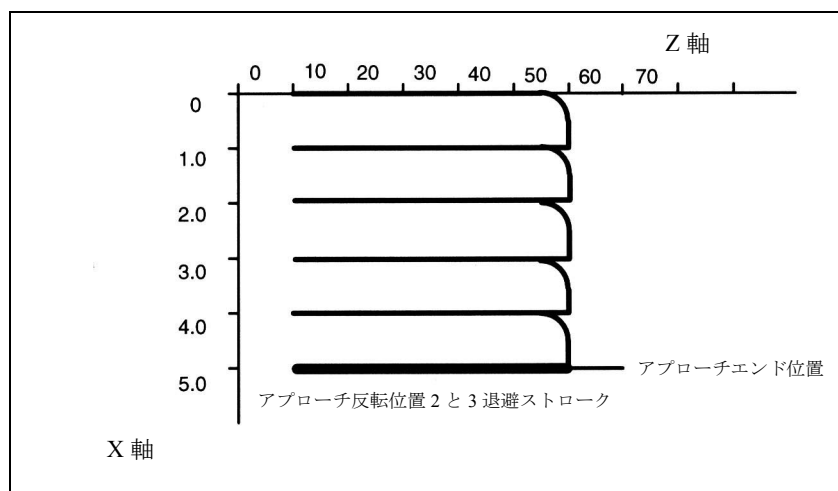


図 2.97 振動動作と送りのシーケンス，例 2

## 2.11.7 データフィールド, リスト

### ■ インタフェース信号

DB 番号	ビット, バイト	名称	参照
送り軸別／主軸別		送り軸／主軸への信号	
31, ...	28.3	反転位置の設定	
31, ...	28.4	反転位置の変更	
31, ...	28.5	次の反転位置での停止	
31, ...	28.6	ブレーキランプに沿った停止	
31, ...	28.7	PLC 制御装置軸	
		送り軸／主軸からの信号	
31, ...	100.3	振動が始動できない	
31, ...	100.4	振動動作中エラー	
31, ...	100.5	退避アクティブ	
31, ...	100.6	振動動作アクティブ	
31, ...	100.7	振動アクティブ	

### ■ マシンデータ

番号	識別子	名称	参照
一般 (\$MN_...)			
11460	OSCILL_MODE_MASK	非同期振動の制御画面フォーム	

### ■ 設定データ

番号	識別子	名称	参照
軸別 (\$SA_...)			
43700	OSCILL_REVERSE_POS1	反転位置 1	
43710	OSCILL_REVERSE_POS2	反転位置 2	
43720	OSCILL_DWELL_TIME1	反転位置 1 での停止時間	
43730	OSCILL_DWELL_TIME2	反転位置 2 での停止時間	
43740	OSCILL_VELO	振動軸のフィード速度	
43750	OSCILL_NUM_SPARK_CYCLES	退避ストローク数	
43760	OSCILL_END_POS	退避ストローク後／振動動作の終了時の位置	
43770	OSCILL_CTRL_MASK	振動のための制御画面フォーム	
43780	OSCILL_IS_ACTIVE	振動動作オン／オフ	

### ■ アラーム

アラームの詳細説明については次を参照してください。

参照： アラーム一覧

あるいは MMC 101/102/103 を有するシステムのオンラインヘルプ

## ■ モーション同期アクションのメインラン変数

メインラン変数 -read は、下記の変数のことです。

Main run variable\_read:

ISA_IN[<arith.expression>]	デジタル入力 (Boolean)
ISA_OUT[<arith.expression>]	デジタル出力 (Boolean)
ISA_INA[<arith.expression>]	アナログ入力 (Boolean)
ISA_OUTA[<arith.expression>]	アナログ出力 (Boolean)
ISA_INCO[<arith.expression>]	コンパレータ入力 (Boolean)
ISAA_IW[<arith.expression>]	実際位置, 軸 PCS (Real)
ISAA_IB[<arith.expression>]	実際位置, 軸 BCS (Real)
ISAA_IM[<arith.expression>]	実際位置, 軸 MCS(IPO セットポイント) (Real)
	\$AA_IM[S1] では、スピンドルの実際値が評価できる。スピンドルと回転軸については、マシンデータ \$MA_ROT_IS_MODULO および \$MA_DISPLAY_IS_MODULO モジュール計算によってはモジュール計算が適用できる。
ISAC_TIME	s でのブロックスタート (Real) からの時間
ISAC_TIMEC	IPO サイクルでのブロックスタート (Real) からの時間
ISAC_DTBB	BCS でのブロックスタートまでの距離 (開始距離, 基本座標系) (Real)
ISAC_DTBW	PCS でのブロックスタートまでの距離 (開始距離, ワークピース座標系) (Real)
ISAC_DTBB[<axial expression>]	BCS ブロックスタートまでの軸パス (開始距離, 基本座標系) (Real)
ISAA_DTBW[<axial expression>]	PCS でのブロックスタートへの軸パス (開始距離, ワークピース座標系) (Real)
ISAC_DTEB	BCS でのブロックエンドまでの距離 (エンドまでの距離) (開始距離, ワークピース座標系) (Real)
ISAC_DTEW	PCS でのブロックエンドまでの距離 (エンドまでの距離, ワークピース座標系) (Real)
ISAA_DTEB[<axial expression>]	BCS での動作のエンドまでの軸パス (開始距離, 基本座標系) (Real)
ISAA_DTEW[<axial expression>]	PCS での動作のエンドまでの軸パス (エンドまでの距離, ワークピース座標系) (Real)
ISAC_PLTBB	BCS でのブロックスタートからのパス長さ (開始からのパス長さ) (Real)
ISAC_PLTEB	BCS でのブロックエンドまでのパス長さ (エンドまでの距離) (エンドまでのパス長さ, 基本座標系) (Real)

\$AC_VACTB	BCS でのパス速度 ( 実際速度, 基本座標系 )(Real)
\$AC_VACTW	PCS でのパス速度 ( 実際値, ワークピース座標系 )(Real)
\$AA_VACTB[<axial expression>]	BCS での軸速度 ( 実際値, 基本座標系 )(Real)
\$AA_VACTW[<axial expression>]	PCS での軸速度 ( 実際速度, ワークピース座標系 )(Real)
\$AA_DTEPB[<axial expression>]	送りのための軸移動距離, BCS での振動 ( エンドまでの距離, 振子, 基本座標系 )(Real)
\$AA_DTEPW[<axial expression>]	送りのための軸移動距離, 振動 PCS( エンドまでの距離, 振子, ワークピース座標系 )(Real)
\$AC_DTEPB	BCS での振動, 残りの送り用パス (P2 ではない) ( エンドまでの距離, 振子, 基本座標系 )(Real)
\$AC_DTEPW	PCS での振動, 残りの送り用パス (P2 ではない) ( エンドまでの距離, 振子, 基本座標系 )(Real)
\$AC_PATHN	( パスパラメータの標準化 )(Real) 標準化されたパスパラメータ:ブロックスタートの場合は 0, ブロックエンドの場合は 1 まで (Real)
\$AA_LOAD[<axial expression>]	ドライブロード
\$AA_POWER[<axial expression>]	W でのリアルドライブ出力
\$AA_TORQUE[<axial expression>]	Nm でのドライブトルクセットポイント
\$AA_CURR[<axial.expression>]	軸の現在値
\$AC_MARKER[<arithmetic expression>](int)	<p>マーカー変数: 複雑な条件を設定するために, 同期化されたアクションで使用できる。</p> <p>8 種類のマーカーがある ( インデックス 0 ~ 7)。</p> <p>リセットすると, マーカーは 0 にセットされる。</p> <p>例: WHEN...DO \$AC_MARKER[0]=2  WHEN...DO \$AC_MARKER[0]=3  WHEN \$AC_MARKER[0]==3 DO  \$AC_OVR=50</p> <p>パートプログラムで同期化されたアクションとは関係なく, マーカーを読取り, 上書きできる:</p> <p>IF \$AC_MARKER==4 GOTOF SPRUNG</p>
\$AC_PARAM[<arithmetic expression>](Real)	<p>同期化アクションのための一定していない小数パラメータ。同期化アクションのバッファリングと評価に使用。</p> <p>50 種類のパラメータがある ( インデックス 0 ~ 49)。</p>

---

`ISAA_OSCILL_REVERSE_POS1[<axial.expression>](Real)`

`ISAA_OSCILL_REVERSE_POS2[<axial.expression>](Real)`

振動のための現行の反転位置 1 と 2:

現在の設定データ値は、それぞれ、

`$SA_OSCILL_REVERSE_POS1`

`$SA_OSCILL_REVERSE_POS2` から読取られる。

設定データでの反転位置の変更は、同期化された

アクションがアクティブの間は意味を持つアク

ティブな振動によって、有効になる。

## 条件

モーション同期アクションの条件を式で表します。

メインラン変数関連オペレータ数式

---



## 2.12 回転軸 (R2)

### 2.12.1 概要説明

#### 工作機械の回転軸

回転軸は多くの最新式工作機械に利用されています。ツールやワークピースの向き（オリエンテーション）、補助的動作、テクノロジ的あるいは運動学的（キネマティック）動作など様々な動作のために回転軸が必要です。

回転軸を必要とする工作機械の典型的な例が 5 軸フライス盤です。このタイプのマシンは、回転軸の助けを借りて初めて、そのツールの先端をマシン上にあるワークピースのどの個所にでも位置決めすることが可能となります。

マシンのタイプにより、回転軸に課せられる要求は様々です。制御装置が様々なタイプのマシンに適応できるように、個々の回転軸のファンクションをマシンデータや特殊プログラミングで起動することが可能です。

回転軸は常に度単位でプログラムされます。また一般に、正確に一回転（モジュロ  $360^\circ$ ）すると同じ位置にくるという特徴があります。用途により、回転軸のトラバース動作範囲を（例えばツールホルダ用スイベル軸に対して） $360^\circ$  未満に制限したり、あるいはまた（例えばツールやワークピースを回転させる場合は）無限にすることも可能です。

回転軸の動作や特徴は多くの点で直線軸の場合と同じです。したがって、以後の機能に関する説明は、回転軸に特有なものや直線軸との違いに限定しています。

### 2.12.2 詳細説明

#### ■ 一般事項

#### 回転軸の定義

マシンデータ IS\_ROT\_AX で軸を回転軸として宣言することができます。その軸がすでにジオメトリ軸として定義されている場合は、アラーム 4200 が出力されます。軸は回転軸として宣言済みの場合のみ、以後のページに記載されている機能（例えば無制限のトラバース動作範囲や軸位置のモジュロ表示など）を実行あるいは利用することができます。また複数の軸を回転軸として同時に宣言することもできます。

#### 回転軸のタイプ

用途により、回転軸の動作範囲を無制限にすること（すなわち両方向への無限の回転 MD: ROT\_IS\_MODULO = 1）やソフトウェアリミットスイッチで制限すること（例えば  $0 \sim 60^\circ$  の間の動作範囲）、あるいは特定の回転数に限定すること（例えば  $1000^\circ$ ）などが可能です。

下記のリストは回転軸の典型的な用途をいくつか示したものです。

## 典型的な用途

- 5 軸加工（動作範囲は制限あるいは無制限）
- 偏心加工用回転軸（動作範囲無制限）
- 円柱状あるいは輪郭グライディング回転軸（動作範囲無制限）
- TRANSMIT による C 軸（動作範囲無制限）
- ワインディングマシン上の回転軸（動作範囲無制限）
- ホブ盤上の回転ワークピース軸 (C)（動作範囲無制限）
- 丸ツールマガジンおよびツールタレット（動作範囲無制限）
- 生成面変換用回転軸（動作範囲制限）
- グリップ動作用スイベル軸（動作範囲  $360^\circ$  ）
- スイベル動作用回転軸（動作範囲  $< 360^\circ$  ；例えば  $60^\circ$  ）
- ホブ盤上のフライススイベル軸 (A)（動作範囲例えば  $90^\circ$  ）

## 軸アドレス

数値制御工作機械の座標軸および動作方向は、DIN によって定義されています。DIN 66025 は回転軸およびスイベル軸に対して次の軸アドレスを指定しています：中心軸としての X, Y, Z に対して A, B, および C；すなわち A は X の周りを、B は Y の周りを、C は Z の周りをそれぞれ回ります（下図参照）。回転軸の正の方向は、対応する中心軸の正の軸方向から見て右回りとなります。

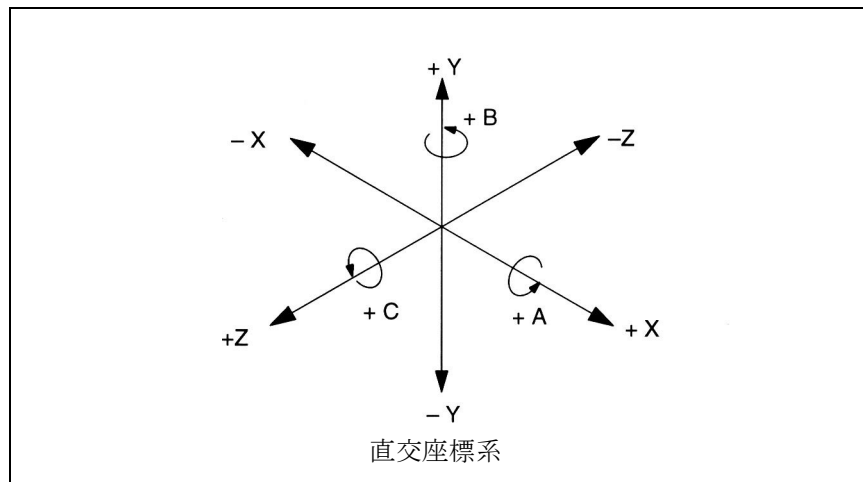


図 2.98 回転軸の軸識別と動作方向

拡張アドレス指定（例えば C2=）や自由構成の軸アドレスは追加回転軸の場合に利用できます。

(注)

X1 軸, Y1 軸, あるいは Z1 軸を回転軸として宣言する場合は、当該動作モードを選択あるいは変更することはできません。

MD 20050: AXCONF\_GEOAX\_ASSIGN\_TAB（チャンネル軸へのジオメトリ軸の割当て）は対応する軸に合うように変えなければなりません。

## 測定単位

標準測定単位として下記のものが回転軸のデータ入出力値に適用されます：

表 2.19 回転軸の測定単位

物理量	単位
角度位置	度
プログラムされた角速度	度／分
角速度用 MD	回転／分
角加速度用 MD	回転／秒 <sup>2</sup> 1)
角度平滑用 MD	回転／秒 <sup>3</sup> 1)

1) これらの単位は、軸が回転軸として宣言されると直ちに軸別マシンデータとともに制御装置によって解釈されます。

ユーザはオプションとして、マシンデータを使ってデータ入出力値に対して他の単位を定義することができます。

参照： 1.8 位置指令単位と検出単位 (G2)

## 動作範囲

軸動作範囲は軸別マシンデータおよび設定データ（ソフトウェアリミットスイッチと作業エリア制限値）を使って定義することができます。回転軸に対してモジュロ変換 (MD: ROT\_IS\_MODULO = "1") が起動されるとすぐに動作範囲の制限がなくなり、ソフトウェアリミットスイッチおよび作業エリア制限値もイナクティブとなります。

## 位置表示

位置表示用の値の範囲は、回転軸について頻繁に選択されるモジュロ 360° 表現に合わせて設定することができます (MD: DISPLAY\_IS\_MODULO = "1")。

## フィードレート

プログラムフィードレート F は回転軸の場合、角速度 [度／分] に一致します。

回転軸と直線軸が共通のパスに沿って G94 または G95 でトラバースする場合は、当該フィードレートは直線軸の測定単位 [例えば mm/min, inch/min] で解釈されるべきです。

回転軸の接線（タンジェンシャル）速度は、直径 D E（単位径 D E = 360 / π）に関係します。単位系 D = D E の場合、度／min 単位のプログラム角速度と mm/min（または inch/min）単位の接線速度は数値的に一致します。

接線速度については一般に下記事項が適用されます：

$$F = F W * D / D E F \quad = \text{接線速度 [mm/min]}$$

$$F W = \text{角速度 [度／min]}$$

$$D = F \text{ が有効な直径 [mm]}$$

$$\text{ここで } D E = 360 / \pi \quad D E = \text{単位径 [mm]}$$

$$\pi = \text{円周率 Pi}$$

## 回転フィードレート

JOG モードの場合、軸／スピンドルの動作は設定データ JOG\_REV\_IS\_ACTIVE (JOG がアクティブの時の回転フィードレート) の設定値にも依存します。

- 当該設定データがアクティブの場合、軸／スピンドルは、マスタスピンドルに依存しながら、常に回転フィードレート MD JOG\_REV\_VELO (JOG に関するフィードレート) または MD JOG\_REV\_VELO\_RAPID (急速トラバースオーバーライドでの JOG に関する回転フィードレート) で移動します。
- 当該設定データがアクティブでない場合、軸／スピンドルの動作は設定データ ASSIGN\_FEED\_PER\_REV\_SOURCE (位置決め軸／スピンドル用回転フィードレート) に依存します。
- 当該設定データがアクティブでない場合、回転有りフレームが有効なジオメトリ軸の動作は、チャンネル別設定データ JOG\_FEED\_PER\_REV\_SOURCE (JOG モードにおいて、回転有りフレームが有効なジオメトリ軸の回転フィードレート) に依存します。

### ■ モジュロ 360°

#### 用語「モジュロ 360°」

回転軸はしばしば 360° 表現方式でプログラムされます。このモジュロ機能を利用するには、軸を回転軸として定義しなければなりません。

回転軸に関しては、「モジュロ」というと 0° から 359.999° の範囲内で制御装置に軸位置をイメージすることを意味します。パス設定値 > 360° (例えば G91 による相対寸法プログラミングの場合) なら、軸位置は変換プロセスによって 0° から 360° の値域で制御装置内でイメージされます。このイメージプロセスは JOG モードおよび AUTOMATIC モードで適用されます。サービス表示は除外です。

下記の図では、正の回転方向における回転軸の絶対位置が螺旋状に描かれています。矢印は実際の絶対位置を示します (例: 点 C' = 420°)。矢印を円の周りをスライドさせながら後退させていくと (螺旋と円の位置 0° は一致)、360° の範囲内で各絶対位置に対応するモジュロ位置を決定することができます。下記の例では、絶対位置点 C' = 420° はモジュロ変換プロセスによって点 C = 60° 上にイメージされています。

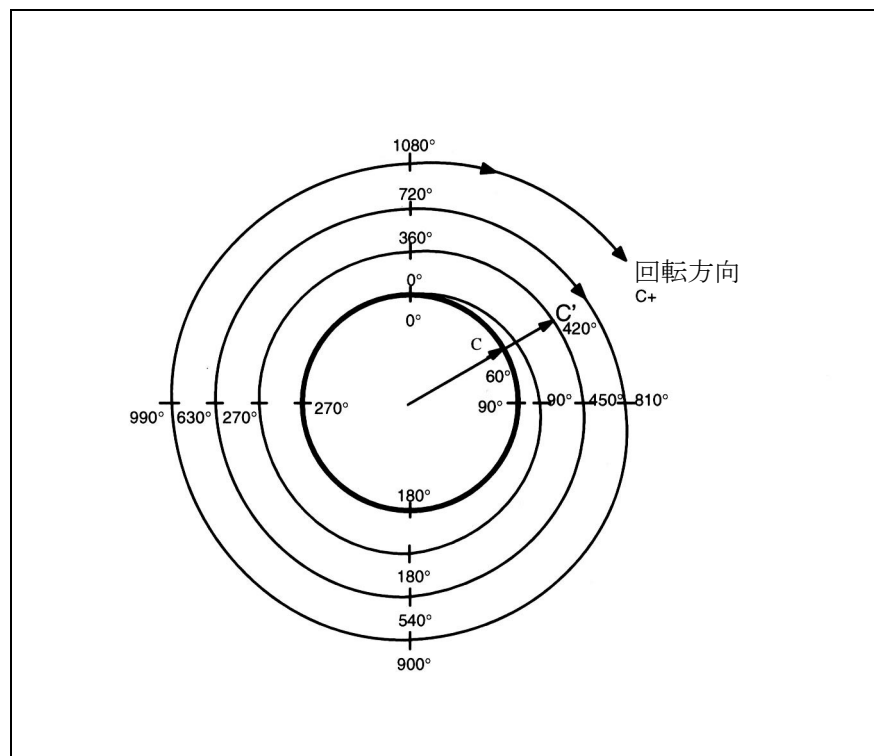


図 2.99 モジュロ 360° イメージ

## マシンデータ設定

個々の工作機械の要求条件を満たすには、マシンデータを使い、各回転軸に対してプログラミングおよび位置決め設定値 (MD: ROT\_IS\_MODULO) とモジュロ 360° での位置表示 (MD: DISPLAY\_IS\_MODULO) を定義することが可能です。

## モジュロ軸

MD: ROT\_IS\_MODULO = "1":

このマシンデータの起動により、プログラミング (G90, AC, ACP, ACN あるいは DC) に対して回転軸の位置決め動作を定義することで、システムで実行される特殊な回転軸動作を活用することが可能となります (「**アクティブモジュロ変換を伴う回転軸 (無限に回転する回転軸)**」参照)。モジュロ 360° イメージプロセスは現在のゼロオフセットが考慮された後、制御装置で内部的に実行されます。算出された移動先位置は続いて一回転以内でアプローチされます。

ソフトウェアリミットスイッチや作業エリア制限は無効となっており、また動作範囲も無制限です (無限に回転する回転軸)。

詳細については「**回転軸のプログラミング**」, すなわち MD: ROT\_IS\_MODULO を参照してください。

## モジュロ位置表示

MD: DISPLAY\_IS\_MODULO = 1:

回転軸に関しては「モジュロ 360°」(1 回転) 位置表示がたびたび必要となります。すなわち、軸が正方向に回転していると、制御装置で内部的に表示が 359.999° から 0.000° まで周期的に再設定されます；負方向の回転の場合、軸位置は同様に 0° ～ 359.999° の範囲で表示されます。

MD: DISPLAY\_IS\_MODULO = 0:

モジュロ 360° 表示方式は、モジュロ軸 (ROT\_IS\_MODULO = "1") の場合に必ず選択しなければなりません。

モジュロ 360° 表示方式は、モジュロ軸 (ROT\_IS\_MODULO = "1") の場合に必ず選択しなければなりません。

(注)

モジュロ 360° 表示方式は、モジュロ軸 (ROT\_IS\_MODULO = "1") の場合に必ず選択しなければなりません。

## ■ 回転軸のプログラミング

(注)

プログラミングに関する一般的な情報に関しては、下記を参照してください；

参照： YS840DI ユーザーズマニュアル・プログラミング編

## 一般事項

マシンデータ ROT\_IS\_MODULO (回転軸のモジュロ変換) は、プログラミングおよび位置決めの間回転軸が直線軸と同じように動くのか、あるいは回転軸の特殊機能を組込むのかを定義します。これらの特徴と相違（主に絶対寸法プログラミングに関する）については次ページ以降で説明します。

## ■ アクティブモジュロ変換を伴う回転軸（無限に回転する回転軸）

### モジュロ変換起動

=> MD: ROT\_IS\_MODULO = "1"

お勧め：位置表示もモジュロ 360° に設定した方がよいでしょう (MD: DISPLAY\_IS\_MODULO = "1")。

## 絶対寸法プログラミング (AC, ACP, ACN, G90)

軸位置付け例: POS[ 軸名 ] = ACP (値)

- この値は 0 ～ 359.999° の範囲内で回転軸のターゲット位置を示しています。負の記号を持つ値あるいは 360° 以上の値の場合は、アラーム 16830 "Incorrect modulo position programmed" が出力されます。
- ACP (正) および ACN (負) は回転軸のトラバース方向を明確に（実際の位置と関係なく）定義します。
- AC あるいは G90 を使ったプログラミングの場合のみ、トラバース方向は回転軸の実際の位置に依存します。移動先位置が実際の位置より数値的に大きい場合は、軸は正の方向にトラバースしますが、そうでない場合は負の方

向にトラバースします。

- ACP および CAN の使用：非対称のワークピースに関しては、回転中の衝突を避けるためにトラバース方向を定義することが可能でなければなりません。

例：

(下記の図参照)：C のスタート位置は 0°

プログラミング	効果
1 POS[C] = ACP(100)	回転軸 C が正の回転方向に位置 100° までトラバースする
2 POS[C] = ACN(300)	C が負の回転方向に位置 300° までトラバースする
3 POS[C] = ACP(240)	C が正の回転方向に位置 240° までトラバースする
4 POS[C] = AC (0)	C が負の回転方向に位置 0° までトラバースする

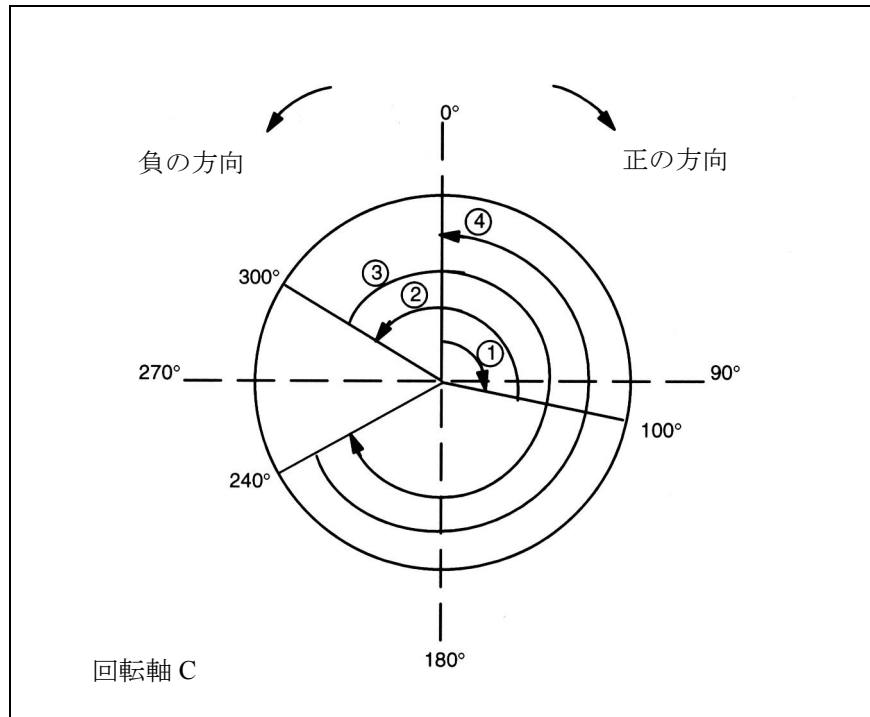


図 2.100 モジュロ軸に関する絶対寸法プログラミングの例

## 最短ルート (DC) での絶対寸法プログラミング

POS [ 軸名 ] = DC (値)

- この値は回転軸の移動先位置を 0° から 359.999° の範囲で示しています。負の記号を持つ値あるいは 360° 以上の値の場合は、アラーム 16830 "Incorrect modulo position programmed" が出力されます。
- DC (Direct Control) により、回転軸は一回転以内に（トラバース動作最大 ± 180° ）最短ルートを通してプログラム絶対位置に接近します。
- 制御装置は実際の位置に応じて回転方向とトラバース距離を計算します。トラバース距離が両方向 (180° ) に同じ場合は、正の方向が優先されます。
- DC の適用例：回転テーブルは最短時間で（したがって最短パスを抜けて）切換え位置に接近する必要がある場合。

- DC が直線軸に関してプログラムされている場合は、アラーム 16800 "DC traversing instruction cannot be used" が出力されます。

例：

(下記の図参照)：C のスタート位置は  $0^{\circ}$

プログラミング

効果

1 POS[C] = DC(100)

C 軸は最短パスに沿って位置  $100^{\circ}$  までトラバースする

2 POS[C] = DC(300)

C 軸は最短パスに沿って位置  $300^{\circ}$  までトラバースする

3 POS[C] = DC(240)

C 軸は最短パスに沿って位置  $240^{\circ}$  までトラバースする

4 POS[C] = DC (60)

C 軸は最短パスに沿って位置  $60^{\circ}$  までトラバースする。  
この場合の距離は両方向で  $180^{\circ}$  に等しいので、正の方向が優先される。

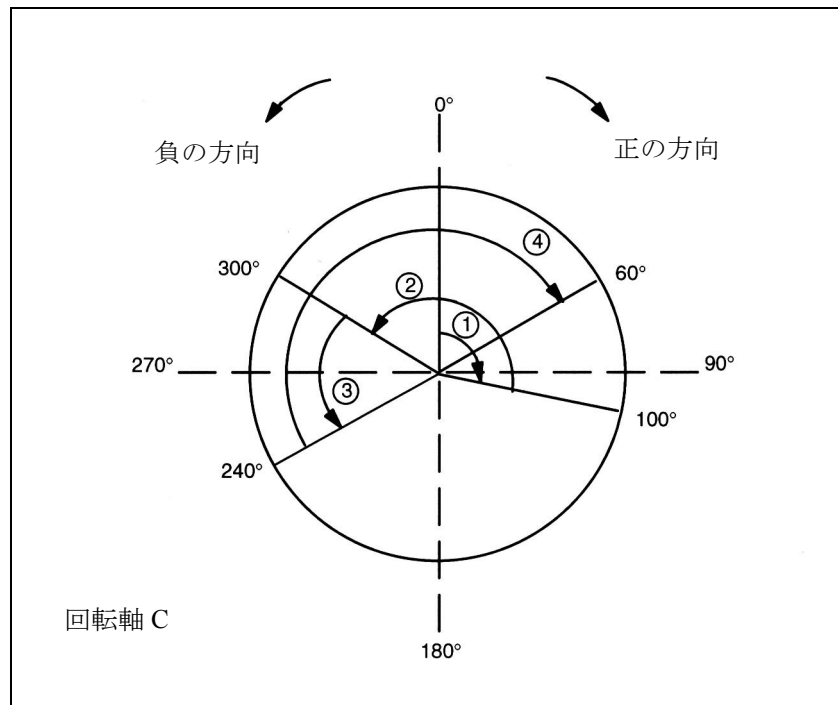


図 2.101 DC プログラミングの例

## 相対寸法プログラミング (IC, G91)

軸位置付け例： POS[ 軸名 ] = IC (+/- 値)

- この値は回転軸のトラバース距離を示しています。負あるいは  $\geq \pm 360$  の値も可能です。
- この値の記号は強制的に回転軸の方向を規定します。
- 適用例：複数回の回転にわたる螺旋状の溝のフライス加工



例：

POS[C] = IC(720)	C 軸は正方向に相対的に 720° (2 回転) トラバースする
POS[C] = IC(-180)	C 軸は負方向に相対的に 180° トラバースする

## 無限のトラバース範囲

モジュロファンクションがアクティブになるとすぐに、トラバース範囲に関して制限がなくなります（ソフトウェアリミットスイッチはイナクティブ）。回転軸はここで連続してトラバースするようにプログラムすることができます。

例：

```
LOOP:
POS[C] = IC(720)
GOTOB LOOP
```

## ■ モジュロ変換を伴わない回転軸

### モジュロ変換停止

=> MD: ROT\_IS\_MODULO = "0"

## 絶対寸法プログラミング (AC, G90)

軸位置付け例： POS[ 軸名 ] = AC (+/- 値)

- この値と先行する記号は回転軸の移動先位置をユニークに示しています。≥ +/-360 の値も可能です。  
この位置値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。
- トラバース方向は、回転軸の実際位置の先行記号に応じて制御装置によって確認されます。
- ACP あるいは ACN がプログラムされている場合は、アラーム 16810 "ACP traversing instruction cannot be used" あるいはアラーム 16820 "ACN traversing instruction cannot be used" が出力されます。
- 適用例：直線動作が回転軸に組込まれる場合（カムギア）；したがって特定のエンド位置を超えて移動してはならない。

例：

プログラミング	効果
POS[C] = AC (-100)	回転軸 C は位置 -100° に接近する；トラバース方向はスタート位置に依存する
POS[C] = AC (1500)	回転軸 C は位置 1500° までトラバースする

## 最短ルート (DC) での絶対寸法プログラミング

POS [ 軸名 ] = DC (値)

回転軸がモジュロ軸として定義されていなくても、DC (直接制御) で位置決めすることができます。応答はモジュロ軸の場合と同じです。

- この値は  $0^{\circ}$  から  $359.999^{\circ}$  の範囲で回転軸の移動先位置を示しています。負の記号を持つ値あるいは  $360^{\circ}$  以上の値の場合は、アラーム 16830 "Incorrect modulo position programmed" が出力されます。
- DC (直接制御) により、回転軸は一回転以内に (トラバース動作最大  $\pm 180^{\circ}$ ) 最短ルートを通してプログラム絶対位置に接近します。
- 制御装置は実際の位置 (モジュロ  $360^{\circ}$  を基準とする) に応じて回転方向とトラバース距離を計算します。トラバース距離が両方向 ( $180^{\circ}$ ) に同じ場合は、正の方向が優先されます。
- DC の適用例: 回転テーブルが最短時間で (したがって最短パスを抜けて) 切換え位置に接近する必要がある場合。
- DC が直線軸に関してプログラムされている場合は、アラーム 16800 "DC traversing instruction cannot be used" 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 $7200^{\circ}$ までトラバースする; トラバース方向はスタート位置しだい。
POS[C] = DC (300)	回転軸 C は最短パスに沿って $300^{\circ}$ の「モジュロ位置」までトラバースする。軸 C はしたがって負の回転方向に $60^{\circ}$ までトラバースし、 $7140^{\circ}$ の絶対位置で停止する
POS[C] = AC (7000)	回転軸 C は $7000^{\circ}$ の絶対位置までトラバースする; ここで C は負の回転方向に $140^{\circ}$ までトラバースする

注記; この例の場合は、モジュロ  $360^{\circ}$  表示 を起動した方がよいでしょう (MD: DISPLAY\_IS\_MODULO = "1")。

## 相対寸法プログラミング (IC, G91)

軸位置付け例: POS [ 軸名 ] = IC(+/- 値)

相対寸法でプログラミングする場合は、回転軸はモジュロ軸と同じパスを抜けてトラバースします。ただしこの場合、トラバース範囲はソフトウェアリミットスイッチで制限されます。

- この値は回転軸のトラバース範囲を示しています。負あるいは  $\geq +/360^{\circ}$  の値も可能です。
- 値の前の記号は回転軸のトラバース方向を規定します。

例については「■モジュロ変換を伴わない回転軸」を参照してください。

## トラバース範囲の制限

直線軸に関してはトラバース範囲は制限されます。範囲制限値はプラスとマイナスのソフトウェアリミットスイッチで定義されます。

## ■ 回転軸に関する種々のプログラミング機能

### オフセット

TRANS（絶対）および ATRANS（付加）を回転軸に適用することができます。

### スケール

SCALE や ASCALE は、制御装置がモジュロ計算を常に  $360^\circ$  の完全な円に基づいて行うため、回転軸には適しません。

### 実際値の設定

PRESETON は可能です。

### インデックス軸

参照： 2.15 割出し軸 (T1)

## ■ 回転軸のスタートアップ

### 手順

回転軸スタートアップ手順は、若干の例外はありますが、直線軸に関するものと同じです。軸が回転軸として定義されると (MD: IS\_ROT\_AX = 1) 直ちに、制御に関する軸別マシンデータと設定データは以下のように解釈されます；

位置	「度」単位
速度	「回転／分」単位
加速度	「回転／秒 <sup>2</sup> 」単位
スムージング	「回転／秒 <sup>3</sup> 」単位

### 特殊 MD

2.12.4 「データの説明 (MD, SD)」で説明する回転軸の特殊マシンデータも、適用操作しだいで入力しなければなりません。

- MD: ROT\_IS\_MODULO                      位置決めおよびプログラミングのモジュロ変換
- MD: DISPLAY\_IS\_MODULO                位置表示のモジュロ変換
- MD: INT\_INCR\_PER\_DEG                角位置計算の精度

以下に、回転軸に関するこれらのマシンデータの可能な組み合わせの概要をリストアップします。

表 2.20 回転軸のマシンデータ組合わせの可能性

MD: IS_ROT_AX 「回転軸」	MD: ROT_IS_MODULO 「回転軸のモジュロ変換」	MD: DISPLAY_IS_MODULO 「モジュロ実際値表示」	適用可否	備考
0	0	0	可	軸は直線軸（標準的なケース）
0	0 あるいは 1	0 あるいは 1	勧められない	軸は回転軸でない；したがって他のマシンデータの信号ステータスは無関係
1	0	0	可	軸は回転軸；位置決めに際しモジュロ変換使用せず、つまりソフトウェアリミットスイッチはアクティブ；位置表示は絶対値
1	1	0	可（勧められない）	軸は回転軸；位置決めはモジュロ変換で実行、すなわちソフトウェアリミットスイッチはイナクティブ；位置表示は絶対値

表 2.21 回転軸のマシンデータ組合わせの可能性

MD: IS_ROT_AX 「回転軸」	MD: ROT_IS_MODULO 「回転軸のモジュロ変換」	MD: DISPLAY_IS_MODULO 「モジュロ実際値表示」	適用可否	備考
1	1	1	可	軸は回転軸；位置決めはモジュロ変換で実行、つまりソフトウェアリミットスイッチはイナクティブ、動作範囲は無制限；位置表示はモジュロ（回転軸の場合最も頻繁に使用される設定）
1	0	1	可	軸は回転軸；位置決めにモジュロ変換使用せず、つまりソフトウェアリミットスイッチはアクティブ；位置表示はモジュロ；適用：例えば動作範囲 +/-10005 の軸の場合

## 回転軸の JOG 速度

全ての回転軸に有効なジョグ速度を SD: JOG\_ROT\_AX\_SET\_VELO（回転軸の JOG 速度）で設定することができます。

値 "0" が設定データに入力されると、軸 MD: JOG\_VELO（通常の軸速度）が回転軸の JOG 速度として機能します。

参照： 2.4 JOG 運転とハンドル (H1)

## ■ 回転軸に関する注意事項

### 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および作業エリア制限は機能しており、動作範囲に制限のあるスイベル軸には必要です。ただし、無限に回転する回転軸 (MD: ROT\_IS\_MODULO=1) の場合は、ソフトウェアリミットスイッチと作業エリア制限はイナクティブに設定されます。

参照： 1.2 送り軸監視機能及びプロテクションゾーン (A3)

### 回転軸のミラーリング

プログラミングコマンド MIRROR(C) および AMIRROR(C) で回転軸のミラーリングが実行可能です。

### 基準点アプローチ

参照： 1.16 原点復帰 (R1)

### 回転軸としてのスピンドル

スピンドルおよび回転軸の使用 (C 軸操作) に関する注意事項については下記を参照してください；

参照： 1.17 主軸機能 (S1)

## 2.12.3 補足条件

当該機能説明に関して明記すべき補足条件はありません。

## 2.12.4 データの説明 (MD, SD)

### ■ 軸／スピンドル別マシンデータ

30320 MD 番号	DISPLAY_IS_MODULO 位置表示はモジュロ 360°	
デフォルト値: 1	最小入力リミット: 0	最大入力リミット: 1
変更が有効になるための条件: 電源オン	保護レベル: 2	単位: -
データタイプ: BOOLEAN	適用開始 SW バージョン: 1.1 以上	
意味:	<p>1: 「モジュロ 360°」位置表示アクティブ: 回転軸あるいはスピンドルの位置表示（基本あるいはマシン座標系）は「モジュロ 360°」と定義される。制御装置は内部的に位置表示を各周期 359.999 度に続く 0.000 度にリセットする。表示範囲は常に正であり、常に 0° から 359.999° の間。</p> <p>0: 絶対位置表示アクティブ: モジュロ 360° 表示方式と対照的に、絶対位置が絶対位置表示で示される（例えば 1 回転後 +360° や 2 回転後 +720° など）。この場合、表示範囲は直線軸に従って制御装置に制限される。</p>	
この MD は次の場合は無意味	直線軸 MD: IS_ROT_AX = "0"	
アプリケーション	<ul style="list-style-type: none"> <li>連続して回転する軸 (MD: ROT_IS_MODULO = "1") の場合は、モジュロ 360° の位置表示を起動した方がよい。</li> <li>スピンドルの位置表示は常にモジュロ 360° で起動しなければならない。</li> </ul>	
関連性	MD: IS_ROT_AX = 1 「軸は回転軸」	

34220 MD 番号	ENC_ABS_TURNS_MODULO[n] 回転エンコーダの絶対エンコーダ範囲: 0 ~ 最大エンコーダ数マイナス 1	
デフォルト値: 4096, 4096	最小入力リミット: 1	最大入力リミット: 4096
変更が有効になるための条件: 電源オン	保護レベル: 2 / 7	単位: -
データタイプ: DWORD	適用開始 SW バージョン: 2.2 以上	
意味:	<p>絶対エンコーダのスイッチを入れると、回転軸の絶対位置が下記の範囲まで下げられる: つまり、MODULO 変換が実行された場合、実際の読み込み位置が MD ENC_ABS_TURNS_MOTOR で許されている位置を超える。</p> <p>0 度 ≤ 位置 ≤ n*360 度, (n = ENC_ABS_TURNS_MODULO)</p> <p>注記: SW 2.2. の場合、制御装置／エンコーダのスイッチを入れると、この範囲まで位置が下げられる。SW 3.6 以上に関しては、制御装置あるいはエンコーダのスイッチが切られると、この半分の値で最大許容トラバースパスを表示する。</p>	
例外, エラー	<p>値として許されるのは 2 の累乗のみ (1, 2, 4, 8, 16, ..., 4096)。</p> <p>他の値が入力されると、ソフトウェアバージョン 4.1 までは、メッセージを出すことなくそれらは四捨五入される。ソフトウェアバージョン 4.1 以上では、四捨五入されマシンデータに表示される; 変更はアラーム 26025 に表示される。</p> <p>この MD だけが (直線軸および回転軸の) 回転エンコーダに関係している。</p> <p>重要な勧告:</p> <p>デフォルト値 "1 encoder revolution" は、ソフトウェアバージョン 3.6 から "4096" に変更されている。この新しい値が最も一般的に使用されるエンコーダタイプの設定をより確実なものとしている。より少ない多重回転情報のエンコーダを使用するか、あるいは単一回転のエンコーダを使用する場合は、この値はそれに応じて減らさなければならない。多重回転絶対エンコーダに関してはいかなる場合も、結果として拡大される明確なトラバース範囲を活用できるように、この値をエンコーダがサポートする最大量に変更すべきである (注記: エンコーダあるいは電源のスイッチが入れられると、この値は許容位置オフセットにも影響を及ぼす)。</p>	

34220 MD 番号	ENC_ABS_TURNS_MODULO[n] 回転エンコーダの絶対エンコーダ範囲：0～最大エンコーダ数マイナス 1
関連性	Drive MD 1021, ENC_ABS_TURNS_MOTOR, Drive MD 1031, ENC_ABS_TURNS_DIRECT

30300 MD 番号	IS_ROT_AX 回転軸	
デフォルト値：0	最小入力リミット：0	最大入力リミット：1
変更が有効になるための条件：電源オン	保護レベル：2	単位：-
データタイプ：BOOLEAN	適用開始 SW バージョン：1.1 以上	
意味：	<p>1: 軸：この軸は「回転軸」として定義される</p> <ul style="list-style-type: none"> <li>回転軸の特殊機能がアクティブであるか、もしくは必要とされるマシンのタイプに応じて追加マシンデータにより起動が可能である（下記参照）。</li> <li>測定単位は度。</li> <li>軸別マシンデータおよび設定データの単位は、標準制御設定で次の通り解釈される： <ul style="list-style-type: none"> <li>位置，単位 「度」</li> <li>速度，単位 「回転／分」</li> <li>加速度，単位 「回転／秒<sup>2</sup>」</li> <li>スムージング，単位 「回転／秒<sup>3</sup>」</li> </ul> </li> </ul> <p>スピンドル：</p> <p>スピンドルの場合，このマシンデータは常に "1" に設定しなければならない，そうでなければアラーム 4210 "rotary axis declaration missing" が出力される。</p> <p>0: 軸は「直線軸」として定義される。</p>	
例外，エラー	軸の場合：当該軸がすでにジオメトリ軸として定義されている場合はアラーム 4200。スピンドルの場合：アラーム 4210	
関連性	<p>下記のマシンデータは，MD:IS_ROT_AX・="1" の起動後のみ有効である：</p> <ul style="list-style-type: none"> <li>MD: ROT_IS_MODULO 「回転軸のモジュロ変換」</li> <li>MD: DISPLAY_IS_MODULO 「位置表示はモジュロ」</li> <li>MD: INT_INCR_PER_DEG 「角位置の計算精度」</li> </ul>	
参考	表 2.20	

30310 MD 番号	ROT_IS_MODULO 回転軸のモジュロ変換	
デフォルト値：0	最小入力リミット：0	最大入力リミット：1
変更が有効になるための条件：電源オン	保護レベル：2	単位：-
データタイプ：BOOLEAN	適用開始 SW バージョン：1.1 以上	
意味：	<p>1: モジュロ変換が回転軸のセットポイントに対して実行される。ソフトウェアリミットスイッチおよび作業エリア制限は無効；したがってトラバース範囲は両方向に無制限。</p> <p>MD: ROT_IS_AX は "1" に設定しなければならない。</p> <p>詳細については 2.12.2 「<b>モジュロ 360°</b>」を参照のこと。</p> <p>0: モジュロ変換なし</p>	

30310 MD 番号	ROT_IS_MODULO 回転軸のモジュロ変換
この MD は次の場合は無意味	MD: IS_ROT_AX = "0" (直線軸)
表 2.20	マシンデータの可能な組み合わせ
アプリケーション	連続して回転する軸 (例えば偏心回転, 研削加工, ワインディング)
関連性	MD: DISPLAY_IS_MODULO 「位置表示はモジュロ 360°」 MD: IS_ROT_AX = 1 「回転軸」 MD: POS_LIMIT_MINUS 「ソフトウェアリミットスイッチ マイナス」 MD: POS_LIMIT_PLUS 「ソフトウェアリミットスイッチ プラス」 SD: WORKAREA_LIMIT_MINUS 「作業エリア制限マイナス」 SD: WORKAREA_LIMIT_PLUS 「作業エリア制限プラス」

## 2.12.5 信号の説明

当該機能説明に関しては明記すべき個別の信号はありません。

## 2.12.6 例

### フォークヘッド, 傾斜軸ヘッド

回転軸は, ツール軸をスイベルさせたりワークピースを回転させるために, 5 軸フライス盤で頻繁に使用されます。これらのマシンはワークピースのどの個所にもツールの先端を位置づけることができ, またツール軸に対してどんな配置をとることもできます。アプリケーションに応じて様々なフライスヘッドが必要です。図 2.102 は回転軸の構成例としてフォークヘッドと傾斜軸を示しています。

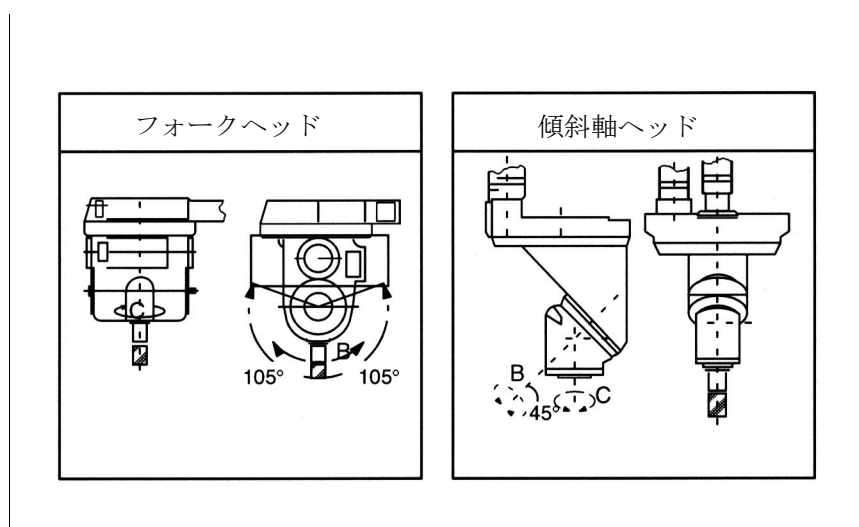


図 2.102 フォークヘッド, 傾斜軸ヘッド



## 2.12.7 データフィールド, リスト

### ■ インタフェース信号

- なし -

### ■ マシンデータ

番号	識別子	名称	参考
一般事項 (\$MN_...)			
10210	INT_INCR_PER_DEG	角位置の計算精度	1.7 (G2)
軸／スピンドル別 (\$MA_...)			
30320	DISPLAY_IS_MODULO	モジュロ実際値表示	
30300	IS_ROT_AX	軸は回転軸	
36100	POS_LIMIT_MINUS	ソフトウェアリミットスイッチマイナス	1.2 (A3)
36110	POS_LIMIT_PLUS	ソフトウェアリミットスイッチプラス	1.2 (A3)
30310	ROT_IS_MODULO	回転軸のモジュロ変換	

### ■ 設定データ

番号	識別子	名称	参考
一般事項 (\$MN_...)			
41130	JOG_ROT_AX_SET_VELO	回転軸の JOG 速度	2.4 (H1)
軸別 (\$SA_...)			
43430	WORKAREA_LIMIT_MINUS	作業エリア制限マイナス	1.2 (A3)
43420	WORKAREA_LIMIT_PLUS	作業エリア制限プラス	1.2 (A3)

---

## ■ アラーム

アラームの詳細説明については次を参照してください。

参照： アラーム一覧

あるいは MMC 101/102/103 を有するシステムのオンラインヘルプ

## 2.13 主軸同期 (S3)

### 2.13.1 概略説明

#### 主軸同期

このファンクション (2.13.3 「補足条件」を参照) を使用すれば、高い角度精度で、リードスピンドルと追従スピンドルとの間で同期をとることができます。

また、たとえば、最終的な加工のために、旋盤上でワークピースをスピンドル 1 からスピンドル 2 へ瞬時に移すこともできます。長所は、ダウンタイムを省けることです。

スピンドル速度の同期とは別に、スピンドル間の相対角度位置 (たとえば、ワークピースの向きを指定しての瞬時の移動) も指定できます。

リードスピンドル (LS) と追従スピンドル (FS) との間のワークピースの瞬時移動:

- $n_{FS} = n_{LS}$  速度同期
- $FS = LS$  位置同期
- または
- $FS = LS + \pi$  角度オフセットの位置同期

#### 多角形加工

補足機能として、メインスピンドルと「ツールスピンドル」との間の整数倍速比  $k_U$  の指定が、多角形加工 (多角形旋削) の基礎を提供します。

多角形旋削:

- $n_{FS} = k_U [n_{LS}]$

同期運転は、CNC パートプログラムを介して選択/選択解除されます。

各マシン用の主軸同期ペアには、チャンネル別マシンデータによって固定構成を割当てるとも、CNC パートプログラムを介して特定のアプリケーション用に定義することもできます。

NC チャンネルごとに、最高 2 組までの主軸同期ペアが運転可能です。

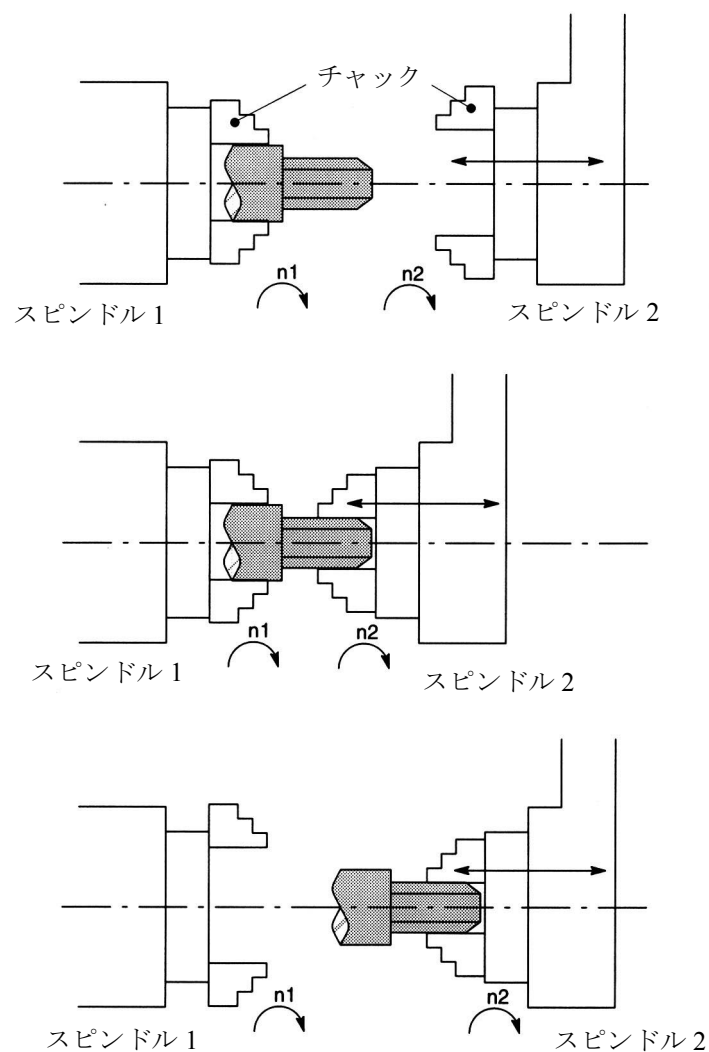


図 2.103 同期モード，スピンダル 1 からスピンダル 1 への瞬時ワークピース移動

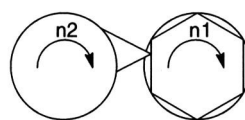


図 2.104 多角形旋削

## 2.13.2 詳細説明

### ■ 一般機能性

### ■ 同期モード

#### 説明

< 軸の式 > は:            - 軸識別子  
                              または  
                              - スピンドル識別子です。

< 軸識別子 >:            C (スピンドル 1 に、軸モードで識別子 C が付いている場合)

< スピンドル識別子 >: n = スピンドル番号で、Sn, SPI(n)

< スピンドル番号 >: 1, 2, ... MD 35000:SPIND\_ASSIGN\_TO\_MACHAX で定義されたスピンドル番号に従って。

#### 主軸同期ペア

同期モードでは、追従スピンドル (FS) とリードスピンドル (LS) が関与します。これらは、通常、主軸同期ペアと呼びます。追従スピンドルは、定義されたファンクションの相互関係に従って、アクティブカップリング（同期運転）によってリードスピンドルの動きを真似ます。

#### 同期モード

同期モード（「主軸同期運転」とも呼ぶ）も、スピンドル運転モードです。同期モードの起動の前に、追従（スレーブ側）スピンドルが位置制御に切り換えられていなければなりません。カップリングが起動すると、追従スピンドルの同期運転も起動します。カップリングが停止すると、追従スピンドルは直ちに開ループ制御モードに戻ります。

追従スピンドルの同期運転が起動すると、IS "Synchronous operation"（同期運転）(DB31, ... DBX84.4) = 1 が PLC に伝えられます。

#### 主軸同期の数

NC チャンネルごとに、2 つの主軸同期カップリングが同時に運転できます。

#### 構成カップリングまたはユーザ定義カップリング

主軸同期カップリングは:

- ・ チャンネル別マシンデータを介した固定構成（以下、「固定カップリング構成」と呼ぶ）として定義されるか、
- ・ パートプログラムで言語命令 (COUP...) を介して自由に定義されたカップリング（以下、「ユーザ定義カップリング」と呼ぶ）として定義されます。

次のような種類が可能です:

11. カップリング用の固定構成は、マシンデータを介してプログラムをすることができます。一方、別のカップリングも、パートプログラムを介して自由に定義できます。

12.マシンデータを介してカップリングは構成されません。この場合、2 つのカップリングは、ユーザ定義カップリングで、パートプログラムを介してパラメータ化することができます。

## 同期モードのオプション

同期モードでは、次のファンクションオプションが使用できます。

- 追従スピンドルとリードスピンドルが同じ速度で回転します。  
( $n_{FS} = n_{LS}$ ; 速度比  $k_U = 1$ )
- LS と FS が同じ方向に回転するか逆に回転するかを（正または負の速度比  $k_U$  を指定することによって）指定できます。
- 追従スピンドルとリードスピンドルが異なる速度で回転します。  
( $n_{FS} = k_U \cdot n_{LS}$ ; 速度比  $k_U \neq 1$ )  
アプリケーション: 多角形旋削
- LS と FS との間の可変角度位置 ( $\theta_{FS} = \theta_{LS} + \pi$ )  
LS と FS との間に定義された角度オフセット（位置同期カップリング）をもたせてスピンドルが同期速度で回転します。  
アプリケーション: ワークピースの整形
- スピンドルが回転中でも停止中でも、LS と FS との間の同期運転が可能です。
- リードスピンドルに関し、開ループと位置制御モードの全ファンクションが使用できます。
- 同期モードがアクティブでない場合、他のいかなるスピンドルモードでも FS および LS を運転することができます。
- アクティブ同期モードでスピンドルが回転中の時、速度比を変更することができます。
- 主軸同期カップリングがオンになると、LS への FS のオフセット（オーバーレイ動作）が変更できます。

## 主軸同期ペアの定義

同期モードの起動の前に、カップリングすべきスピンドル (FS, LS) が定義されていなければなりません。

これは、対象のアプリケーションに応じて、2 通りの方法で行うことができます。

### 1. 固定カップリング構成

リードおよび追従スピンドルとして作動するマシン軸は、チャンネル別 MD: COUPLE\_AXIS\_1[n] で定義されなければなりません。

このカップリング構成に合わせて、LS および FS としてプログラムされたマシン軸は、NC パートプログラムでは変更することができません。

必要に応じて、NC パートプログラムで、カップリングパラメータを修正することができます。

### 2. ユーザ定義カップリング

言語命令 "COUPDEF(FS,LS...)" によって、NC パートプログラムで、カップリングの生成と変更が可能になります。新しいカップリングの関係を定義しなければならない場合には、（言語命令 COUPDEL(FS,LS) で）既存のユーザ定義

カップリングをあらかじめ削除する必要があります。

主軸同期カップリングが明確に定義されるように、各言語命令 COUP..., に合わせて、FS および LS で追従およびリードスピンドル用の軸識別子 (Sn, SPI(n)) がプログラムされなければなりません。

有効なスピンドル番号は、軸別 MD: SPIND\_ASSIGN\_TO\_MACHAX で、マシン軸に割当てられていなければなりません。

参照： /PA/, Programming Guide

インタフェース信号 "FS active" (FS アクティブ) (DB31, ... DBX99.1) によって、追従またはリードスピンドルのどちらにマシン軸がプログラムされているかが、PLC に通知されます。

## 速度比

速度比は、分子と分母（速度比パラメータ）に別々の数値を指定してプログラムされています。従って、有理数でも、正確に速度比を指定することができます。

$$\text{一般法則: } k_U = \frac{\text{速度比パラメータ 分子}}{\text{速度比パラメータ 分母}} = \frac{U \text{ 分子}}{U \text{ 分母}}$$

速度比パラメータの値の範囲（U 分子, U 分母）は、制御内部で、ほぼ無制限です。

マシンデータを介して構成されたカップリング用の速度パラメータは、チャンネル別設定データ (SD: COUPLE\_RATIO\_1[n]) でセットすることができます。更に、言語命令 COUPDEF (FS, LS, U 分子, U 分母, ...) で、比を変更することができます。この場合、設定データへの入力値は上書きされません（初期設定）。

言語命令 COUPDEF (...) では、NC パートプログラムを介して定義されたカップリングの比を入力することしかできません。

COUPDEF 命令の処理が終了すると、新しい比のパラメータが有効になります。

## カップリング特性

主軸同期カップリングごとに、次の特性が定義できます。

- ブロック変更応答

カップリングがアクティブの時の、速度定義角度オフセットまたは比の変更時、または、同期運転の起動時に、満たすべき次のブロック変更条件が定義できます。

- 直ちにブロック変更が起こります。

- "Fine synchronism"（微同期）に対するブロック変更応答

- "Coarse synchronism"（粗同期）に対するブロック変更応答

- IPOSTOP（即ち、セットポイント同期の後）に対するブロック変更応答

- WAITC で、任意の瞬間における同期条件のチェック。

- FS と LS との間のカップリングのタイプ

リードスピンドルの位置セットポイントあるいは位置の実際値を、追従スピンドルの基準値とすることができます。次の2つのタイプのカップリングが選択できます。

#### 1. セットポイントカップリング (DV)

位置制御モードでのアプリケーション。両方のスピンドルの動的制御応答は、可能な限り一致していなければなりません。ただし、セットポイントカップリングの方が優先されます。

#### 2. 実際値カップリング (AV)

LS の位置制御が可能でないか、FS と LS との間に制御特性が大きく異なっている場合のアプリケーション。FS 用のセットポイントは、LS の実際値から得られます。セットポイントカップリングと比べると同期品質が劣ります。

#### 3. 速度カップリング (VV)

内部的には、速度カップリングはセットポイントカップリングと同等ですが、FS と LS に対する要件が低く設定されています。FS および LS には、位置制御系あるいは測定系は要求されません。

FS と LS との間の位置オフセットは定義されません。

カップリング特性は、固定カップリング構成 (@セクション 2.3 を参照) の場合、マシンデータを介して選択されて、ユーザ定義カップリング (@セクション 2.2.1 を参照) の場合、言語命令 COUPDEF を介して選択されます。

更に、カップリング特性 "Type of coupling" (カップリングのタイプ) および "Block change re-sponse" (ブロック変更応答) は、言語命令 COUPDEF によって、固定カップリング構成に合わせて変更することができます。

### カップリング特性の保護変更

構成カップリングパラメータ "Speed ratio" (速度比), "Type of coupling" (カップリングのタイプ) および "Block change response" (ブロック変更応答) が NC パートプログラムで変更できるかどうか定義するために、チャンネル別 MD: COUPLE\_IS\_WRITE\_PROT\_1 が次のようにセットされます。

0: カップリングパラメータは、COUPDEF を使用して NC パートプログラムで変更できます。

1: カップリングパラメータは、NC パートプログラムでは変更できません。変更しようとしても拒否されてアラームメッセージがでます。

### オーバレイ動作

同期モードがアクティブの時、リードスピンドルの動作を主軸同期がプログラムされた速度比に従って真似します。

同時に、LS と FS が、特定の相対角度位置を保ちながら作動できるように、主軸同期をオーバレイでトラバースさせることができます。

FS のオーバレイトラバース動作は、次のような方法で開始できます。

#### 1. AUTOMATIC および MDA でプログラム可能な FS の位置オフセット:

LS と FS との間の位置基準は、言語命令 COUPON および SPOS (「**同期モードの選択**」を参照) で、アクティブな同期運転中に変更することができます。



---

## 2. FS の手動位置オフセット :

- JOG 運転モード (連続 JOG または相対 JOG) で :

アクティブ同期運転で, プラスかマイナスのトラバースキーまたはハンドルを使用した FS のオーバーレイ。

- AUTOMATIC および MDA 運転モードで :

DRF オフセットを介して, ハンドルを使用した FS のオーバーレイ。

FS がオーバーレイトラバース動作を実行すると, IS "Overlaid movement" (オーバーレイ動作) (DB31, ... DBX98.4) が "1" にセットされます。オーバーレイ動作は, 可能な限りの最高速度で, 最適の時間で実行されます。SPOS によるオフセット変更で, 位置決め速度は, FA[Sn] でセットすることができ, かつ, オーバライド (DB31,... DBX17.0 で選択の可能性) の影響を受けることがあります。

参照 : 1.17 主軸機能 (S1)

## ■ 同期モードの選択

### カップリングの起動

言語命令 COUPON によって、最後の有効なパラメータで、プログラムされたスピンドル間のカップリングが起動するとともに、同期モードも起動します。このカップリングには、固定構成またはユーザ定義の 2 種類があります。起動の瞬間に、リードスピンドルおよび（または）追従スピンドルは停止中でも回転中でもかまいません。

同期運転が起動できる状態にするためには、一定の条件が満たされていなければなりません（「■同期モードの前提条件」を参照）。

### 起動方法

次のような 2 通りの方法の中から、同期モードの起動方法が選択できます。

1. リードスピンドルと追従スピンドルとの間の角度基準を指定せずに、可能な限り高速でカップリングを起動します。

COUPON(FS, LS)

2. リードスピンドルと追従スピンドルとの間の角度オフセット POS<sub>FS</sub> を定義した状態でカップリングを起動します。この方法では、角度オフセットが選択時にプログラムされなければなりません。

COUPON(FS, LS, POS FS)

(注)

ソフトウェアバージョン 3.2 以降から、同期カップリングをオンにする前に LS および（または）FS が軸モードになっていれば、スピンドル識別子によって、軸モードが解除されてスピンドルモードが起動します。

軸識別子でスピンドルをオンにした場合には、切り換えが起こりません。

### ブロック変更応答

同期運転が選択される前に、同期モードの起動時に、ブロック変更がどのような条件の下で行われるかを判定しなければなりません（「■準備プログラミング命令」を参照）。

### 現在のカップリングステータスの判定

軸システム変数 \$AA\_COUP\_ACT[<軸の式>] によって、NC パートプログラムで、指定された軸／スピンドルの現在のカップリングステータスを判定することができます（「主軸同期用の軸システム変数」参照）。追従スピンドル用の主軸同期カップリングがアクティブになると、読取り時にビット 2 が "1" になっています。

### 定義された角度オフセットの変更

言語命令 COUPON および SPOS を使用すれば、同期モードがアクティブの時に、定義された角度オフセットを変更することができます。追従スピンドルは、POS<sub>FS</sub> でプログラムされた角度オフセットで、オーバーレイ動作として位置決めされます。IS "Overlaid movement"（オーバーレイ動作）(DB31, ... DBX98.4) は、この期間中にセットされます。

## 角度オフセット POS<sub>FS</sub>

定義された角度オフセット POS<sub>FS</sub> は、正の回転方向へのリードスピンドルの 0 度位置を基準に、絶対位置として指定されなければなりません。

位置制御スピンドルの「0 度位置」は、測定系のゼロマーク信号または Bero 信号、または、軸別マシンデータ (MD: REFP\_SET\_POS, REFP\_MOVE\_DIST, REFP\_MOVE\_DIST\_CORR) にストアされたオフセットから計算されます。

POS<sub>FS</sub> の範囲 : 0 ~ 359.999 度

参照 : 1.16 原点復帰 (R1)

## 現在の角度オフセットの読取り

軸システム変数を使用すれば、NC パートプログラムで、FS と LS との間の現在の位置オフセットを読取ることができます。位置オフセットには次の 2 種類があります。

a) FS と LS との間のセットポイントの現在の位置オフセット

\$AA\_COUP\_OFFS [<FS 用の軸識別子>]

b) FS と LS との間の実値の現在の位置オフセット

\$VA\_COUP\_OFFS [<FS 用の軸識別子>]

(<軸識別子>の詳細は、「■同期モード」を参照してください。)

## 電源オン後の起動

基準化／同期化されていない LS または FS の同期運転を起動させることもできます (IS: "Referenced/synchronized 1 or 2" DB31, ... DBX60.4 or DBX60.5 = 0)。この場合、警告メッセージが表示されます。

例 : 電源オン後、LS および FS は、ワークピースを介して摩擦ロックで既にカップリングされています。

## ■ 同期モードの選択解除

### カップリングの停止

言語命令 COUPOF を使用すれば、プログラムされたスピンドル間の同期モードが取消されます。関連するカップリングには、固定構成およびユーザ定義の 2 種類があります。同期運転を停止する時、リードおよび追従スピンドルは、停止中でも回転中でもかまいません。

同期モードをオフにすると、追従スピンドルが制御モードに切換えられます。最初にプログラムされた FS 用の S ワードが有効ではなくなるため、他の通常のスピンドルと同様に追従スピンドルを運転することができます。

通常、カップリングが停止すると、ブロック前処理停止 (STOPRE) が制御装置内部で開始します。

## スピンドル回転中の停止

スピンドルの回転中に同期モードが停止した場合、追従スピンドルは、引き続き、現在の速度 ( $n_{FS}$ ) で回転します。現在の速度は、NC パートプログラムで、システム変数 \$AA\_S で読取ることができます。

その後、スピンドルは、M05, SPOS または SPOSA でパートプログラムから停止させることも、適切な位置決め信号で PLC から停止させることもできます。

## 選択解除方法

次の 3 つの方法の中から、同期モードの選択解除方法が選択できます。

1. できるだけ短時間でカップリングを停止します。

ブロック変更が直ちにイネーブルされます。

COUPOF(FS, LS)

2. プログラムされた停止位置  $POS_{FS}$  を追従スピンドルがオーバトラベルするまで、カップリングは選択解除されません。

解除後、ブロック変更がイネーブルされます。

COUPOF(FS, LS,  $POS_{FS}$ )

3. プログラムされた停止位置  $POS_{FS}$  および  $POS_{LS}$  を追従スピンドルとリードスピンドルがオーバトラベルするまで、カップリングは選択解除されません。

解除後、ブロック変更がイネーブルされます。

COUPOF(FS, LS,  $POS_{FS}$ ,  $POS_{LS}$ )

(注)

ソフトウェアバージョン 3.2 以降から、同期モードをオフにする前に LS および (または) FS が軸モードになっている場合は、スピンドル識別子で、軸モードが解除されて速度制御モードが起動します。

軸識別子を使用してスピンドルをオフにした場合は、切り換えは起こりません。

シャットダウンの前に LS が停止していなければなりません (セットポイント側)。

## $POS_{FS}$ , $POS_{LS}$

停止位置  $POS_{FS}$  および  $POS_{LS}$  は、定義された基準点の値を基準として、FS および LS の実際位置に対応します (「**同期モードの選択**」を参照)。

$POS_{FS}$ ,  $POS_{LS}$  の範囲 : 0 ~ 359.999 度

参照 : 1.16 原点復帰 (R1)

## ■ 同期モードの前提条件

### 同期モードの選択条件

主軸同期カップリングが起動させられるか、アラームメッセージが出される前に、次の条件が満たされていなければなりません。

- 事前に、主軸同期カップリングが、マシンデータを介して固定構成に定義されているか、パートプログラムを介してユーザ定義に従って定義されていなければなりません。
- カップリングすべきスピンドルは、カップリングが起動される NC チャンネルで定義されていなければなりません。

チャンネル別 MD 20070: AXCONF \_MACHAX\_USED

軸別 MD 35000: SPIND\_ASSIGN\_TO\_MACHAX

- 追従スピンドルが、カップリングが起動される NC チャンネルに割当てられていなければなりません。

軸別 MD30550: AXCONF\_ASSIGN\_MASTER\_CHAN での初期設定

- LS および FS は、最低 1 つの位置検出用の位置測定系を備えている必要があります。
- 同期運転の起動の前に、FS が速度制御モード (IS "Position controller active" DB31, ... DBX61.5 = 0) にある場合には、言語命令 SPCON で、位置制御に切換えられなければなりません。

(注)

位置制御が起動した時、LS の最高セットポイント速度は、位置エンコーダのリミット周波数で決まる最高速度の 90% (制御リザーブ) に自動的に制限されます。制限は、IS "Setpoint speed limited" (DB31, ... DBX83.1) を介して通知されます。

同期運転の停止後、言語命令 SPCOF で、位置制御モードが再び選択解除できます。

参照： 1.17 主軸機能 (S1)

- 同期精度を高めるため、LS と FS との間でセットポイントカップリングが確立できるように、カップリングの起動前に、LS は、位置制御モード (言語命令 SPCON) になっている必要があります。

LS 用の測定系がある場合には、実際値のカップリングが可能です。

- 同期モードを選択する前に、FS および LS に必要なギアステージが選択されていなければなりません。同期モードでは、FS および LS のギアステージ切換え (従って、発振モード) が不可能です。リクエストしても拒否されて、アラームメッセージが出されます。
- FS および (または) LS が軸モードになっていてスピンドル識別子で起動された場合は、スピンドルモードが起動します。関連スピンドル用の VDI インタフェース信号が修正され、アクティブパラメータブロックが切換えられ、フィードフォワード制御が起動します。

軸識別子でスピンドルが起動した場合、切換えは起こりません。

---

## クロスチャンネルカップリング

- セットポイント値カップリングの場合は、位置決め基準が必要なため、LS は FS と同じチャンネル内になければなりません。
- LS は、パートプログラム、PLC (FC18) を介して、どのカップリングタイプにも合うように任意にプログラムすることができます。ソフトウェアバージョン 4 以降からは、同期アクションを介してプログラムできます。
- 速度カップリングの場合は、LS は、"Axis exchange" (軸交換) によって交換することができます。
- ソフトウェアバージョン 4 以降：  
LS は、速度カップリングの場合、どのチャンネル内でも配置可能です。
- ソフトウェアバージョン 4 以降：

(注)

速度カップリングを起動した状態で LS がスワップされ、チャンネルのシーケンスが変更された場合は、カップリングを停止する必要があります。

例：

チャンネル 1:

チャンネル 2:

チャンネル 3: チャンネル 3 で FS。COUPON アクティブ。

チャンネル 4:

チャンネル 5:

LS に関し、次のチャンネル間で容易に交換できます：

チャンネル 1 <--> チャンネル 2,

チャンネル 1 <--> チャンネル 3,

チャンネル 2 <--> チャンネル 3,

チャンネル 4 <--> チャンネル 5

カップリングを停止する必要がある場合、LS に関して交換できます。

チャンネル 1 <--> チャンネル 4 から

チャンネル 2 <--> チャンネル 4 から

チャンネル 3 <--> チャンネル 4 から

チャンネル 1 <--> チャンネル 5 から

チャンネル 2 <--> チャンネル 5 から

チャンネル 3 <--> チャンネル 5 から

## ■ 同期運転の監視

### 微／粗同期

同期モードでは、通常のス핀ドル監視運転の他に、FS と LS との間の同期運転も監視されます。

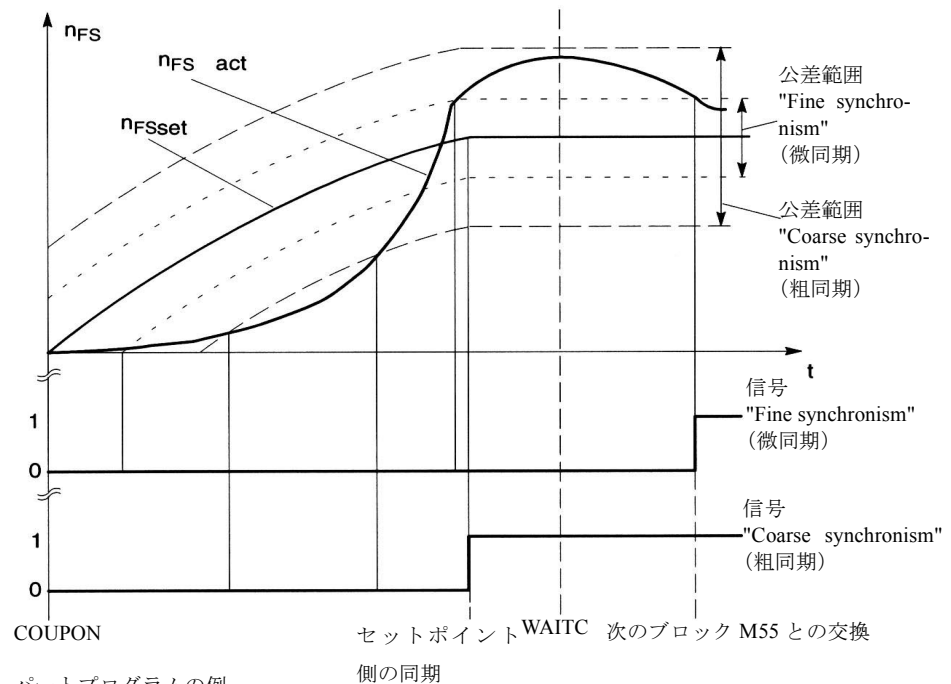
この場合、IS: "Fine synchronism" (DB31, ... DBX98.0) または "Coarse synchronism" (DB31, ... DBX98.1) が PLC に伝えられて、追従ス핀ドルの現在位置 (AV, DV) または実際速度 (VV) が指定公差範囲内に収まっているかどうかを通知します。

カップリングがオンになると、セットポイント側で同期が達成されるとともに、信号 "Coarse synchronism" (粗同期) および "Fine synchronism" (微同期) が更新されます。

公差範囲のサイズは、FS の MD でセットされます。

同期の達成は、次の要因の影響を受けます。

- AV, DV: FS と LS との間の位置偏差
- VV: FS と LS との間の速度差



パートプログラムの例

```
:  
M3 S500 M2=5  
G4 F10  
COUPON(S2, S1)  
Z10  
WAITC(S2, "Fine")  
M55  
:
```

n FS = 追従ス핀ドル実際速度 act  
n FS = 追従ス핀ドル基準速度 set  
n FS = 追従ス핀ドル速度

図 2.105 COUPON および同期テストマーカ WAITC での同期監視

---

## スレッシュホールド値

リードスピンドルに対する追従スピンドルの相対位置または速度公差範囲は、度あるいは r/min の単位で指定しなければなりません。

- "Coarse synchronism" (粗同期) のスレッシュホールド値

軸別 MD 37200: AV, DV: COUPLE \_POS \_TOL \_COARSE

MD 37220: VV: COUPLE \_VELO \_TOL \_COARSE

- "Fine synchronism" (微同期) のスレッシュホールド値

軸別 MD 37210: AV, DV: COUPLE \_POS \_TOL \_FINE

MD 37230: VV: COUPLE \_VELO \_TOL \_FINE

## 速度／加速リミット

同期モードでは、リードスピンドルの速度および加速リミット値は、追従スピンドルがその動きを模倣できるように制御装置内で調整されます。この時、固有のリミット値に違反しないように選択中のギアステージおよび有効速度比が考慮されます。

たとえば、スピンドル間の同期を保つため、LS は自動的に減速し、FS が最高速度を超えないようにします。



## ■ 主軸同期カップリングのプログラミング

表 2.22 概要

プログラムされたカップリング	構成カップリング	備考
カップリング定義： COUPDEF()	構成データの修正 COUPDEF()	カップリングパラ メータの設定
カップリング オン： COUPON()		カップリング オン／オフ
カップリング オフ： COUPOF()		
カップリングデータの削除： COUPDEL()	構成データの再起動： COUPRES()	配置 , 復元

## ■ 準備プログラミング命令

### ユーザ定義カップリング

上記のように、2つの主軸同期カップリングは、どのチャンネルでも、同時にアクティブになることができます。固定カップリング構成がプログラムされていない場合には、NC パートプログラムによって両方のカップリングが自由に定義できます。

これらのカップリングも、NC パートプログラムによってパラメータ化されていなければなりません。デフォルト値は、プログラムされていないパラメータに対し使用されます。

固定構成を持たない FS/LS カップリングの関係が言語命令 COUP-DEF でプログラムされている場合には、新規の主軸同期カップリングが定義されます。たとえば、その他のスピンドル間で主軸同期カップリングが更に必要になる場合、このカップリングは、言語命令 COUP-DEL で再び無効にすることができます。これらの手順（即ち、カップリングの生成と削除）によって、更に多くの（2つ以上の）カップリング関係が、NC チャンネルでプログラムすることができます。

### 恒久カップリング構成

恒久構成主軸同期カップリングのカップリング特性および速度比は、書込み保護されていない場合、NC パートプログラムによって変更することができます。LS および FS 用のマシン軸は変更することができません。

### 新規カップリングの定義

言語命令 "COUPDEF" を使用すれば、新規の主軸同期カップリング（ユーザ定義）を生成したり、既存のカップリング用にパラメータを修正したりすることができます。

カップリングパラメータが十分に指定されている場合には、次のように適用されます。

COUPDEF

(FS, LS, U 分子, U 分母, ブロック変更応答, カップリングタイプ)

FS および LS で、COUP... 命令ごとにプログラムする必要があるスピンドルカップリングが明確に定義されます。定義されていない場合には、アラームメッセージが出されます。

変更する必要がある場合には、その他のカップリングパラメータに対し、プログラミングのみを行う必要があります。最後の有効ステータスは、まだ、指定されないパラメータに対し適用できます。

各カップリングパラメータは次のとおりです。

- FS, LS: 追従およびリードスピンドル用の軸識別子

たとえば: S1, SPI(1), S2, SPI(2)

軸別 MD: SPIND\_ASSIGN\_TO\_MACHAX で、マシン軸に、該当するスピンドル番号が割当てられていなければなりません。

- U 分子, U 分母: 分子と分母のパラメータ

速度比は、分子と分母を数値の形に指定して入力されます (@セクション 2.1.1 を参照)。

分子は、必ず、プログラムされなければなりません。分母が指定されていない場合には、その値は、必ず、"1,0" であると見なされます。

- ブロック変更応答

このパラメータによって、同期運転の選択時のブロック変更条件が定義できます。

NOC --> ブロック変更が直ちにイネーブルされます。

FINE --> "Fine synchronism" (微同期) に対するブロック変更応答

COARSE --> "Coarse synchronism" (粗同期) に対するブロック変更応答

IPOSTOP --> IPOSTOP に対するブロック変更応答 (たとえば、セットポイントベースの同期の後)

ブロック変更応答は、文字列 (引用符とともに) として入力されます。

太字で文字を書込むだけで、ブロック変更応答が指定できます。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されるものであり、それ以外の意味はありません。

ブロック変更応答が指定されていない場合、選択中の応答が引き続き適用されます。

プログラム可能な同期テストマーク WAITC で、指示した同期に到達するまで、新規ブロックとの交換が遅延させられます。

- カップリングのタイプ

DV (目標値) --> FS と LS との間のセットポイントカップリング

AV (実際値) --> FS と LS との間の実際値カップリング

VV (速度値) --> FS と LS との間の速度カップリング

カップリングタイプが指定されていない場合、現在選択されているタイプが引き続き適用されます。

(注)

同期運転が停止した場合には、カップリングの変更のみが行われる場合があります。

## 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

ユーザ定義カップリングに対し、次の初期設定が適用されます。

- U 分子 = 1,0
- U 分母 = 1,0
- ブロック変更応答 = IPOSTOP (セットポイント同期で、ブロック変更イネーブル)
- カップリングタイプ = DV (セットポイントカップリング)

## カップリングの削除

言語命令 "COUPDEL" は、ユーザ定義カップリングの削除に使用されます。

COUPDEL (FS, LS)

新規のカップリング関係を定義する必要があり、予備のユーザ定義カップリング (1 または 2) がない場合には、既存のユーザ定義主軸同期カップリングが削除されなければなりません。

アクティブカップリング用の COUPDEL がプログラムされる場合には、アラームメッセージが出されます。同期運転は、アクティブのままです。事前に、COUPOF で選択解除する必要があります。

(注)

COUPDEL では、固定カップリング構成が削除できません。

## オリジナルカップリングパラメータの起動

言語命令 "COUPRES" によって、構成カップリングパラメータが再起動できます。

COUPRES (FS, LS)

それから、COUPDEF でプログラムされたパラメータ (速度比を含めて) が、上書きされます。

言語命令 "COUPRES" は：

- マシンデータおよび設定データ (固定カップリング構成) にストアされたパラメータを起動し、
- 初期設定 (ユーザ定義構成) を起動します。

## プログラム可能なブロック変更

ソフトウェアバージョン 3.2 以降から,"WAITC"を使用した NC プログラムに点をマークすることができます。システムは、この点で、指定された FS の同期条件が満たされるまで待機して、指定された同期ステータスに到達するまで新規ブロックへの変更を遅延します (図 2.103 を参照)。

WAITC (FS)

長所 : 同期カップリングをオンにしてから同期に到達するまでの期間は、技術的に

見て有用です。

(注)

基本的には、いかなる時でも、WAITC を書込むことができます。指示したスピンドルが FS としてアクティブでない場合には、このスピンドルに対する命令が有効になりません。

同期条件が指示されていない場合には、最低限、セットポイント同期のために、各カップリングでプログラム／構成された同期条件が満たされているかどうかを、必ず、チェックします。

## 例

WAITC(S2),

WAITC(S2, "Fine"),

WAITC(S2, S4, "Fine")

## ■ カップリングの起動と停止のためのプログラミング命令

### 同期モードの起動

言語命令 COUPON は、カップリングおよび同期モードの起動に使用されます。

同期運転の起動方法には、次の 2 つがあります。

#### 1. COUPON(FS, LS)

リードおよび追従スピンドルの間の角度基準で、可能な限り高速に同期運転を起動します。

#### 2. COUPON(FS, LS, POS<sub>FS</sub>)

リードおよび追従スピンドルの間で定義された角度オフセット POS<sub>FS</sub> で、同期運転を起動します。このオフセットは、正の回転方向へのリードスピンドルの 0 度位置を基準にします。ブロック変更は、定義設定に従ってイネーブルされます。POS<sub>FS</sub> の範囲: 0 ... 359.999 度

同期運転が既にアクティブの時に、COUPON(FS, LS, POS<sub>FS</sub>) または SPOS をプログラムすると、LS および FS の間の角度オフセットが変更できます。

### 同期モードの停止

同期運転の停止方法には、次の 3 つがあります。

#### 1. COUPOF(FS, LS)

可能な限り高速に同期運転を停止します。ブロック変更は、直ちにイネーブルされます。

#### 2. COUPOF(FS, LS, POS<sub>FS</sub>)

停止位置 POS<sub>FS</sub> がオーバトラベルされた後、同期運転を選択解除します。この位置がオーバトラベルされるまで、ブロック変更はイネーブルされません。

#### 3. COUPOF(FS, LS, POS<sub>FS</sub>, POS<sub>LS</sub>)

2 つの停止位置 (POS<sub>FS</sub> および POS<sub>LS</sub>) がオーバトラベルされた後、同期運転を選択解除します。両方の位置がオーバトラベルされるまで、ブロック変更はイネーブルされません。

POS<sub>FS</sub>, POS<sub>LS</sub> の範囲 : 0 ~ 359.999 度

連続パス制御 (G64) がプログラムされる場合には、制御内で、ノンモーダル停止がかけられます。

#### 例

COUPDEF (S2, S1, 1.0, 1.0, "FINE", "DV")

:

COUPON (S2, S1, 150)

:

COUPOF (S2, S1, 0)

:

COUPDEL (S2, S1)

### ■ 主軸同期用の軸システム変数

#### 現在のカップリングステータスの読取り

追従スピンドル用の現在のカップリングステータスは、次の軸システム変数を使用して NC パートプログラム中で読取られます。

\$AA\_COUP\_ACT[<軸の式>]

(<軸の式>の詳細は、「■同期モード」を参照してください。)

#### 例

\$AA\_COUP\_ACT[SPI(2)]

読取り値には、追従スピンドルに対し、次の意味があります。

バイト = 0: アクティブカップリングなし

ビット 2 = 1: 主軸同期カップリングアクティブ

#### 現在の角度オフセットの読取り

FS と LS との間の現在の位置オフセットは、次の軸システム変数を使用して、NC パートプログラム中で読取られます。

a) FS と LS との間のセットポイントベースの位置オフセット:

\$AA\_COUP\_OFFS[<軸の式>]

b) FS と LS との間の実際値ベースの位置オフセット:

\$VA\_COUP\_OFFS[<軸の式>]

## 例

\$AA\_COUP\_OFFS[S2]

COUPON で角度オフセットがプログラムされる場合には、セットポイント同期の読取りの後で、これが読取り値と一致します。

(注)

同期運転および追従モードがアクティブの時に、サーボイネーブル信号が取消されると、コントローラが再びイネーブルされる時にかけられる位置オフセットは、最初にプログラムされた値と異なります。この場合、変更された位置オフセットは、必要に応じて、NC パートプログラムで読取られて修正することができます。

## ■ マシンデータを介して行われる主軸同期ペアの構成

### カップリングパラメータ

NC チャンネルごとに、1 組づつの主軸同期ペアが、マシンデータを介して恒久的に構成できます。

その後に、どのマシン軸（スピンドル）をカップリングすべきか定義したり、このカップリングの特性をどのようにすべきかを定義したりする必要があります。

主軸同期カップリング用の固定設定として、次のパラメータが構成できます。

- 主軸同期ペア（チャンネル別 MD: COUPLE\_AXIS\_1[n]

このマシンデータは、主軸同期ペア（追従スピンドル (n=0)、リードスピンドル (n=1)) を構成する 2 つのマシン軸を定義します。

"0" の軸番号設定は、マシンデータを介してカップリングが構成されていないことを意味します。カップリング特性用のマシンデータは無関係です。

NC パートプログラムでは、構成されたカップリング構成に合わせて、LS および FS 用のマシン軸番号を変更することができません。

- 速度比

この比は、設定データ（電源オンアクティブ）を介して、2 つの速度パラメータ（チャンネル別 SD: COUPLE\_RATIO\_1[n]）を分子と分母の形に指定して入力されます。制御内で、比率が出されます。

$$k_U = \frac{\text{速度パラメータ 分子}}{\text{速度パラメータ 分母}} = \frac{\$SC\_COUPLE\_RATIO[0]}{\$SC\_COUPLE\_RATIO[1]}$$

書込み保護されていない場合、速度比は、NC パートプログラムで、言語命令 COUPDEF を出しても変更することができません。

- ブロック変更応答

(チャンネル別 MD: COUPLE\_BLOCK\_CHANGE\_CTRL\_1)

ブロック変更応答条件として、次のオプションの中から 1 つが選択できます。

0: ブロック変更が直ちに起こります。

1: "Fine synchronism"（微同期）に対するブロック変更応答

2: "Coarse synchronism"（粗同期）に対するブロック変更応答

3: IPOSTOP（即ち、セットポイントベースの同期の後）に対するブロック変更応答

- LS と FS との間のカップリングのタイプ：  
(チャンネル別 MD: COUPLING\_MODE\_1)

0: 実際値カップリング

1: セットポイントカップリング

2: 速度カップリング

- NC スタートでのカップリングの強制終了

チャンネル別 MD: COUPLE \_RESET \_MODE\_1（表 2.24 を参照）

- カップリングパラメータの書き込み保護

(チャンネル別 MD: COUPLE \_IS \_WRITE \_PROT\_1)

このマシンデータにおいて、構成カップリングパラメータ "Speed ratio"（速度比）, "Type of coupling"（カップリングタイプ）および "Block change response"（ブロック変更応答）が NC パートプログラムの影響を受けるかどうかを定義することができます。

0: NC パートプログラムによって、カップリングパラメータが変更できます。

1: NC パートプログラムによって、カップリングパラメータが変更できません。  
変更しようとする、アラームメッセージが出て受け付けません。

## ■ NC スタートでの挙動の構成

NC 加工プログラムスタートに対する応答は、チャンネル別マシンデータによって定義されます。

表 2.23 NC スタートでの同期カップリング挙動

	構成カップリング	プログラムされたカップリング (@セクション 2.3 を参照)
	MD: COUPLE_RESET_MODE	MD: START_MODE_MASK
カップリングが保たれる	ビット 0 = 0	ビット 10 = 0
カップリングを選択解除する	ビット 0 = 1	ビット 10 = 1
構成データを起動する	ビット 5 = 1	-
カップリングをオンにする	ビット 9 = 1	-

## ■ リセット挙動の設定

ソフトウェアバージョン 3.2 以降から、NC 加工プログラムのリセットと終わりで、チャンネル別マシンデータで次の挙動が設定できます。

表 2.24 NC 加工プログラム終了とリセット後の同期カップリング挙動

	構成されたカップリング	プログラムされたカップリング (「■マシンデータを介して行われる主軸同期ペアの構成」を参照)
カップリングが保たれる	MD: COUPLE_RESET_MODE ビット 1 = 0	MD: RESET_MODE_MASK- ビット 10 = 1
カップリングを選択解除する	MD: COUPLE_RESET_MODE ビット 1 = 1 MD: RESET _MODE_MASK ビット 0 = 1 (RESET でブロック生成)	MD: RESET_MODE_MASK- ビット 10 = 0 ビット 0 = 1
構成データを起動する	MD: COUPLE_RESET_MODE ビット 6 = 1 MD: RESET _MODE_MASK ビット 0 = 1	-

## ■ 同期運転の特殊機能

### ■ 一般的な同期運転の特殊機能

#### 制御力学

セットポイントカップリングが使用される場合、FS および LS の位置コントローラパラメータ（たとえば K V 係数）がマッチングしていなければなりません。速度制御モードおよび同期運転のパラメータセット (M41...M45) を異なる値に指定して起動する必要がある場合もあります。

#### フィードフォワード制御

制御システムの動的応答の向上の結果、フィードフォワード制御は、同期モードで、追従およびリードスピンドルに対し必ずアクティブです。ただし、軸別 MD: FFW\_MODE (=0) で、FS および LS に対し選択解除することができます。NC パートプログラムは、FFWOF で LS および FS 用のフィードフォワード制御を停止することができません。

フィードフォワード制御モード（速度またはトルクフィードフォワード制御）は、軸別 MD: FFW\_MODE で定義されます。

参照： 2.5 補正機能 (K3)

#### 速度／加速リミット

同期モードで作動するスピンドルの速度／加速リミットは、カップリングにおいて「最も弱い」スピンドルによって判定されます。この目的のため、現在のギアステージ、プログラムされた加速に加えて、リードスピンドルに関し、有効位置制御ステータス（オン／オフ）が考慮されます。

たとえば、リードスピンドルの最高速度は、制御内部で、追従スピンドルのスピンドル制限および速度比に基づき計算されます。



## 複数のカップリング

同期モードの起動時に、FS および LS に対しカップリングが既にアクティブであることをシステムが検出した場合には、起動プロセスが無視されて、アラームメッセージが出されます。

複数のカップリングの例

- 複数の FS に対し、1 つのスピンドルが LS として作動しています。
- 複数の LS に対し、1 つのスピンドルが FS として作動しています。

## 2 段加減速

リードスピンドルに関して、(MD: ACCEL\_REDUCTION\_SPEED\_POINT および ACCEL\_REDUCTION\_FACTOR によって識別される) 追従スピンドルに対する 2 段加減速特性の効果が考慮されます。

ただし、追従スピンドルに関して、全速度範囲を通じて一定の加速が得られているべきです。しかし、「低下する」加速特性が上記の追従スピンドル用のマシンデータに設定される場合、「低下する」LS 加速特性は、加速のときに FS が過負荷にならなかったり、必要に応じて LS に合わせてプログラムされた加速が使用されるようなものでなければなりません。

## PLC による同期モードの直接制御

PLC が処理する ASUPs (非同期サブプログラムの起動) は、AUTOMATIC または MDA モードで、任意の時点で同期モードを起動または終了するために使用することができます。

## アラーム応答

同期運転中にアラーム (たとえば、サーボアラーム) が生じて、制御およびアクティブ追従モードでサーボイネーブル信号が取消された後は、IS "Servo enable" (サーボイネーブル) (DB31, ...DBX2.1) が PLC によって取消された場合と同様に応答します (更に、IS "Follow-up mode" (追従モード) (DB31, ...DBX1.4) がセットされます)。-->「■ PLC インタフェースを介した同期運転への影響」を参照してください。

## ■ PLC インタフェースを介した同期運転への影響

### PLC インタフェース信号

同期運転では、LS および FS インタフェース信号設定によって生じるカップリングに PLC がどのような影響を及ぼすか注意してください。

主軸同期カップリングに対するメイン PLC インタフェース信号の効果は、次のとおりです。

### Spindle speed override (スピンドル速度オーバライド) (DB31, ...DBB19)

同期運転で PLC が行うスピンドル速度オーバライド値入力は、リードスピンドルのみに適用します。

## Axis/spindle disable (軸／スピンドルディスエーブル) (DB31, ... DBX1.3)

関連する軸の挙動は、次の表に示されています (ソフトウェアバージョン 4 以降)。

セットする : 1 セットしない : 0

No.	LS/LA	FS/FA	カップリング	応答
1	0	0	オフ	軸セットポイントが出力される
2	0	1	オフ	FS/FA へのセットポイント出力なし
3	1	0	オフ	LS/LA へのセットポイント出力なし
4	1	1	オフ	LS/LA および FS/FA へのセットポイント出力なし
5	0	0	オン	軸セットポイントが出力される。
6	0	1	オン	FS/FA に対しディスエーブルが無効である。
7	1	0	オン	FS/FA に対してもディスエーブルが有効である。
8	1	1	オン	LS/LA および FS/FA へのセットポイント出力なし

- カップリング起動で、FS/FA に対し信号が無効です。--> "No. 6
- LS/LA に対し信号がセットされた場合、FS/FA(s) に影響を及ぼします。  
--> "No. 7
- 2 つのスピンドル間にクランプされたワークピース (前面から後面加工へのワークピース移動) は破壊されません。

## Servo enable (サーボイネーブル) (DB31, ... DBX2.1)

PLC インタフェースを介した LS の "Servo enable" (サーボイネーブル) の取消し (故障発生時、PLC インタフェースを介して取消されるか、制御内で取り消されます)。

同期運転中に、LS のサーボイネーブル信号が "0" にセットされて、セットポイントカップリングがアクティブの時には、制御内で、実際値カップリングに切り換えられます。この時に、LS が移動中の時は、減速停止されて、アラームメッセージが出されます。同期運転はアクティブのままです。

同期運転の FS 用 "Servo enable" (サーボイネーブル) の取消し (故障発生時に、PLC インタフェースを介して取消されるか、制御内で取消されます)。

信号がリセットされるまで、カップリングが内部で取消されます。

同期運転の選択時に、どちらかのスピンドルに合わせて "Servo enable" (サーボイネーブル) 信号がセットされていない場合には、カップリングがオンになっても同期運転はまだ起動されています。ただし、LS および FS は、両方のサーボイネーブル信号がセットされるまで停止したままです。

LS および FS 用の "Servo enable" (サーボイネーブル) 信号のセット :

IS "Servo enable" (サーボイネーブル) の信号エッジが 1 に切り換わると、IS "Follow-up mode" (追従モード) に応じて、スピンドルが、元の位置 (サーボイネーブルの取消し位置) (信号ステータス = 0: ストップアクティブ) に復帰するか、現在位置 (位置オフセット) が再び使用されます (信号ステータス = 1: 追従アクティブ)。

(注)

カップリングを事前に停止しなかったままスピンドルが停止した後、FS 用の "Servo enable" (サーボイネーブル) 信号が取消されると、"Servo enable" 信号が再びかけられても、外部からの影響 (たとえば、手動回転) による同期エラーが補償されません。そのような場合、LS および FS の間の角度基準は、定義された角度基準で特殊なアプリケーションのために失われる場合があります。

### Follow-up mode (追従モード) (DB31, ... DBX1.4)

ドライブのサーボイネーブル信号が取消された場合にのみ、IS "Follow-up mode" (追従モード) が関連します。FS および LS 用のサーボイネーブル信号がセットされた場合、IS "Follow-up mode" (追従モード) に応じて、イネーブル信号が取消された時の位置 (信号ステータス = 0: ストップアクティブ) にスピンドルが復帰するか、現在位置が使用されます (信号ステータス = 1: 追従アクティブ)。

### Position measuring system (位置測定系) 1/2 (DB31, ... DBX1.5 および 1.6)

同期運転中、LS および FS 用の位置測定系の切換えは阻止されていません。切換えても、カップリングはアクティブのままです。ただし、同期運転が停止させられた時に測定系を切換えることをお勧めします。

同期運転中に、LS または FS に対しパークステータスが選択された場合には、"Servo enable" (サーボイネーブル) の取消しと同様に応答します。

### Spindle reset(delete distance-to-go) (スピンドルリセット (移動距離の削除)) (DB31, ... DBX2.2)

同期運転中に、LS に対しスピンドルリセットがセットされた場合、選択した加速で、LS が減速停止されます。LS および FS は、同期運転のままです。オーバーレイ動作は、(COUP... を除き) できるだけ短時間で終了させられます。

### Spindle stop (feed stop) (スピンドル停止 (フィード停止)) (DB31, ... DBX4.3)

LS または FS 用の "Spindle stop" (スピンドル停止) が起動した場合には、カップリングされたスピンドルの両方がランプに沿って減速停止します。同期モードはアクティブのままです。

カップリングにおいて、どちらかのスピンドルに対し、IS "Spindle stop" (スピンドル停止) がかけられなくなると、前の速度セットポイントまで加速されます。

### アプリケーション

まだ位置制御モードにあるため、"Spindle stop" (スピンドル停止) によって、オフセットなしで、主軸同期ペアを停止することができます。

## 例

主軸同期カップリングがアクティブの間に保護ドアを開放すると、カップリング上の関係を失わないように LS および FS が停止しなければなりません。この場合、IS "Spindle stop" (スピンドル停止) をかけて、LS および FS を減速して停止しなければなりません (IS "Axis/spindle at standstill" (軸／スピンドル停止) (DB31, ... DBX61.4) = 1))。両方のスピンドルに対し、"Servo enable" (サーボイネーブル) を取消することができます。

## S 値の削除 (DB31, ... DBX16.7)

プログラムされた LS 用の S 値が削除されて、ランプに沿って LS が減速停止します。LS および FS は同期運転のままです。

一方、IS "Delete S value" (S 値の削除) は、同期運転で、FS に影響を与えません。

## Re-synchronize spindle 1/2 (スピンドルの再同期) (DB31, ... DBX16.4 および 16.5)

同期モードで動作している場合でも、LS は、位置測定系と同期させることができます。ただし、同期モードが選択解除される場合にのみ、リードスピンドルを再同期することをお勧めします。

## JOG 用のトラバースキー (DB31, ... DBX4.6 および 4.7)

FS のオーバレイトラバース移動できるように作動するため、JOG 用の "Traversing keys plus and minus" (トラバースキープラスおよびマイナス) は、同期運転で、FS に対し阻止されます。

(注)

オーバレイトラバース移動できないようにする必要がある場合には、PLC ユーザプログラムによって、阻止されていなければなりません。

## NC stop axes plus spindles (NC 停止軸プラススピンドル) (DB21, ... DBX7.4)

"NC stop axes plus spindles" (NC 停止軸プラススピンドル) が同期運転で適用された場合には、セット動的応答に対する余裕をとることができるように、カップリングされたスピンドルが減速されます。同期運転モードは、アクティブのままです。

## NC start (NC スタート) (DB21, ... DBX7.1)

「■ NC スタートでの挙動の構成」を参照してください。

(注)

NC 停止後に NC をスタートしても、同期運転を選択解除しません。

## ■ 主軸同期カップリングのスタートアップに関する特殊機能

### スピンドルスタートアップ

リードおよび追従スピンドルは、通常のスピンドルと同様に、最初にスタートアップされなければなりません。関連手順は、次の参照に説明されています。

参照： /IAD/, SINUMERIK 840D Installation and Start-Up Guide

および

参照： 1.17 主軸機能 (S1)

### 定義

主軸同期ペアに対し、次の機能が定義されなければなりません。

- リードおよび追従スピンドル用のマシン軸番号  
(固定カップリング構成に対し、チャンネル別 MD: COUPLE \_AXIS\_1[n] で)。
- 必要なカップリングタイプ (セットポイント, 実際値または速度カップリング)  
(固定カップリング構成に対し、チャンネル別 MD: COUPLING\_MODE\_1[n] で)。
- 同期運転用の LS および FS のギアステージ (n)
- 固定主軸同期カップリング構成用のカップリング特性は、次のとおりです  
(2.13.4 「■マシンデータの説明」を参照) :
  - 主軸同期運転におけるブロック変更応答  
チャンネル別 MD: COUPLE \_BLOCK\_CHANGE\_CTRL\_1
  - カップリング強制終了応答  
チャンネル別 MD: COUPLE \_RESET\_MODE\_1
  - カップリングパラメータの書き込み保護  
チャンネル別 MD: COUPLE \_IS\_WRITE\_PROT\_1
  - 主軸同期カップリング用の速度パラメータ  
チャンネル別 SD: COUPLE \_RATIO\_1[n]

### セットポイント変更応答

セットポイントカップリングと組合わせて可能な限り最高の同期を得るためには、追従およびリードスピンドルが、セットポイント変更に対し同じ動的応答を示さなければなりません。軸制御ループ (位置, 速度および現在のコントローラ) は、可能な限り迅速かつ効率的に故障を除去するため、それぞれ、最適にセットされなければなりません。制御の質を失わない状態で、異なる軸動的応答特性が適合できるように、セットポイント分岐において動的応答適合が得られます。

次の制御パラメータは、それぞれ、LS および FS に対し、最適の軸でセットされなければなりません。

- サーボゲイン係数 (MD 32200 POSCTRL\_GAIN)
- フィードフォワード制御パラメータ
  - MD 32620 FFW\_MODE
  - MD 32610 VELO\_FFW\_WEIGHT
  - MD 32650 AX\_INERTIA

MD 32800 EQUIV\_CURRCTRL\_TIME

MD 32810 EQUIV\_SPEEDCTRL\_TIME

参照： 2.5 補正機能 (K3)

LS および FS に対し、次の制御パラメータが同様にセットされなければなりません。

- 微補間器タイプ (MD 33000: FIPO\_TYPE)
- 軸ジャーク制限

MD 32400 AX\_JERK\_ENABLE

MD 32410 AX\_JERK\_TIME

MD 32420 JOG\_AND\_POS\_JERK\_ENABLE

MD 32430 JOG\_AND\_POS\_MAX\_JERK

参照： 1.8 位置指令単位と検出単位 (G2)

## 動的マッチング

カップリングされた追従およびリードスピンドルは、それぞれ、同じ方法で、セットポイント変更に対し動的に応答しなければなりません。「同じ」動的応答とは、スピンドルが同じ速度で動作する時に追従エラーも同じように拡大しなければならないことを意味します。

セットポイント分岐に動的適合した状態で、異なる動的応答特性（制御ループ）の軸セットポイント変更に対する応答をマッチングすることができます。各ケースにおける「最も弱い」スピンドルおよびその他のスピンドルの間の等価時定数の差は、動的応答適合時定数として入力されなければなりません。

## 例

速度フィードフォワード制御ファンクションがアクティブの時に、動的応答は、基本的に、「最も遅い」速度制御ループの等価時定数によって判定されます。

リードスピンドル： MD 32810: EQUIV\_SPEEDCTRL\_TIME [n] = 5ms

追従スピンドル： MD 32810: EQUIV\_SPEEDCTRL\_TIME [n] = 3ms

--> 追従スピンドルのための動的応答適合の時定数

MD 32910: DYN\_MATCH\_TIME [n] = 5ms - 3ms = 2ms

動的応答適合は、MD 32900 DYN\_MATCH\_ENABLE で軸方向に起動させられなければなりません。

(運転エリア "Diagnosis; Service Axes" (診断; サービス軸) 表示で) LS および FS の追従エラーを比較すれば、適用された動的応答をチェックすることができます。スピンドルが同じ速度で作動中の時、それぞれ、同じ規模の追従エラーが出されていなければなりません。

微調整のため、最適の結果が得られるように、サーボゲイン係数またはフィードフォワード制御を少しだけ調整する必要がある場合もあります。

## 閉ループ制御パラメータセット

スピンドルで、ギアステージごとに、別々の位置制御パラメータセットが割当てられます。

たとえば、パラメータセットを使用すれば、同期運転で、リードスピンドルの動的応答を追従スピンドルの動的応答に調整することができます。従って、（速度または位置決めモードで）カップリングが停止させられる時、LS および FS 用として、その他の位置制御パラメータを選択することができます。この機能を使用するため、同期運転に、個別のギアステージが使用されなければなりません。同期運転の選択前に、このギアステージが起動させられなければなりません。

参照： 1.8 位置指令単位と検出単位（G2）

## 実際値リンク

実際値リンクが使用される場合には、リードスピンドルのドライブよりも、追従スピンドルのドライブの方が動的であるべきです。各ドライブは、動的ドライブ性能の面でも最適にセットされます。

実際値カップリングは、特殊な場合にのみ使用されます。

## 速度リンク

速度リンクは、FS および LS における低い動的要件の内部セットポイントリンクです。FS、LS または両方には位置制御の必要がないため、測定系の必要はありません。

## 粗／微同期用のスレッシュホールド値

フィードフォワード制御ファンクションの設定およびコントローラの最適化の後、粗／微同期用のスレッシュホールド値が入力されなければなりません。

- "Coarse synchronism"（粗同期）用のスレッシュホールド値  
軸別 MD 37200: AV, DV: COUPLE\_POS\_TOL\_COARSE  
MD 37220: VV: COUPLE\_VELO\_TOL\_COARSE
- "Fine synchronism"（微同期）用のスレッシュホールド値  
軸別 MD 37210: AV, DV: COUPLE\_POS\_TOL\_FINE  
MD 37230: VV: COUPLE\_VELO\_TOL\_FINE

値は、マシン製造業者が要求する精度に応じて計算されなければなりません（PLC インタフェースまたは FS の サービス表示を介してチェックしてください）。

---

## LS/FS 間の角度オフセット

LS および FS の間で定義された角度オフセットをプログラムする必要がある場合には、たとえば、同期モードの起動時に、FS および LS の「0 度位置」が相互に調整されなければなりません。これは、次のマシンデータで実行することができます。

MD 34100 REFP \_SET \_POS

MD 34080 REFP \_MOVE \_DIST

MD 34090 REFP · MOVE \_DIST \_CORR

参照： 1.16 原点復帰 (R1)

## FS 用のサービス表示

同期モードでのスタートアップのための運転エリアの "Diagnosis" (診断) 運転エリアの "Service values axes" (サービス値軸) 表示における追従スピンドルに関する値は、次のように表示されます。

- FS および LS のセットポイント間の実際偏差

表示値：リードスピンドル（セットポイント）に対する位置オフセット（軸変数 \$AA\_COUP\_OFFS で、パートプログラムに読取り可能な、LS と FS の間の角度オフセットに対応します。）

- FS および LS の実際値の間の実際偏差

表示値：リードスピンドル（実際値）に対する位置オフセット

### 2.13.3 補足条件

#### "Synchronous spindle" (主軸同期) ファンクションの可用性

このファンクションは、オプションです。

#### "WAITC" ファンクションの可用性

このファンクションは、主軸同期とともに使用できます。



## 2.13.4 データの説明 (MD, SD)

### ■ マシンデータの説明

### ■ チャンネル別マシンデータ

21300 MD 番号	COUPLE_AXIS_1[n] 主軸同期ペア [n] の定義	
デフォルト値: 0	最小入力リミット: 0	最大入力リミット: 8
変更が有効になるための条件: 電源オン	保護レベル: 2/7	単位: -
データタイプ: BYTE	適用ソフトウェアバージョン: 2.1	
意味:	NC チャンネルごとに 1 組の主軸同期ペアが、このマシンデータで固定構成に定義できる。 追従スピンドル [n=0] およびリードスピンドル [n=1] に関し、NC チャンネルに適用可能なマシン軸番号 (チャンネル別 MD: AXCONF_MACHAX_USED) が入力されなければならない。 "0" の値が入力される場合には、カップリングが構成されないため、NC パートプログラムを介して、2 つのカップリングが自由に構成される。	
この MD は次の場合は無効	ユーザ定義カップリング	
例外, エラー, ....	チャンネル別 MD: COUPLING_MODE_1 (主軸同期モードのカップリングタイプ) チャンネル別 MD: COUPLE_IS_WRITE_PROT_1 (カップリングパラメータの書込み保護) チャンネル別 MD: COUPLE_RESET_MODE_1 (カップリング強制終了応答) チャンネル別 MD: COUPLE_BLOCK_CHANGE_CTRL_1 (主軸同期モードにおけるブロック変更応答) SD: \$SC_COUPLE_RATIO_1 (主軸同期モードの速度比パラメータ)	

21310 MD 番号	COUPLE_IS_DES_POS_1 主軸同期モードのカップリングタイプ	
デフォルト値: 1	最小入力リミット: 0	最大入力リミット: 2
変更が有効になるための条件: 電源オン	保護レベル: 2/7	単位: -
データタイプ: BOOLEAN	適用ソフトウェアバージョン: 2.1	
意味:	<p>このマシンデータは、マシンデータ COUPLE_AXIS_1[n] で定義された固定カップリング構成用のカップリングタイプを判定する。</p> <p>1: セットポイントカップリング起動</p> <p>FS および LS 用のセットポイントが同時に入力できるように、セットポイントカップリングで、追従スピンドル用の基準値が、リードスピンドルの位置セットポイントから計算される。これは、加速/減速プロセスで、特に、スピンドル同期に対しポジティブな効果を及ぼす。</p> <p>従って、実際値カップリングよりもセットポイントカップリングの方が、セットポイント変更に対しすぐれた応答を示している。</p> <p>セットポイントカップリングが選択される場合には、同期運転の起動の前に次の条件が満たされていなければならない。</p> <ul style="list-style-type: none"> <li>• LS が、FS と同じ NC チャンネルに割当てられていなければならない。</li> <li>• FS および LS が、位置制御モード (SPCON) にあること。</li> <li>• FS および LS が、同じ動的制御応答を示していなければならない (2.13.2 「主軸同期カップリングのスタートアップに関する特殊機能」を参照)。</li> </ul> <p>0: 実際値カップリング起動</p> <p>実際値カップリングで、追従スピンドル用の基準値は、リードスピンドルの位置実際値から計算される。このタイプのカップリングでは、追従ドライブは、リードドライブよりもずっと動的である必要があるが、リードドライブは、追従ドライブよりも動的である必要がない。</p> <p>実際値カップリングは、たとえば、次のアプリケーションに使用することができる。</p> <ul style="list-style-type: none"> <li>• LS は、FS と異なる NC チャンネルに割当てられていなければならない。</li> <li>• 位置制御に適合しないリードスピンドルに対し。</li> <li>• リードスピンドルの動的制御応答が、追従スピンドルの動的制御応答よりもずっと遅い場合。</li> </ul> <p>実際値カップリングがアクティブになると、FS 用の IS "Actual-value coupling" (実際値カップリング) が "1" にセットされる。</p> <p>2: 速度カップリング起動</p> <p>速度カップリングは、内部ではセットポイントカップリングである。FS および LS に対し要求される値が低い。FS と LS との間の位置関係は、定義することができない。</p> <p>次の場合には、速度カップリングが適用される。</p> <ul style="list-style-type: none"> <li>• LS, FS または両方が位置制御にない。</li> <li>• 測定系がない。</li> </ul> <p>言語命令 COUPDEF によってカップリングが停止させられる場合には、チャンネル別 MD: COUPLE_IS_WRITE_PROT_1 でこのオプションが阻止されていない限り、NC パートプログラムでカップリングタイプが変更できる。ただし、チャンネル別 MD: COUPLING_MODE_1 のパラメータ値は、変更されない。</p>	
この MD は次の場合は無効	ユーザ定義カップリング	
例外, エラー, ....	<p>チャンネル別 MD: COUPLE_AXIS_1 (主軸同期ペアの定義)</p> <p>チャンネル別 MD: COUPLE_IS_WRITE_PROT_1 (構成パラメータの書き込み保護)</p> <p>IS "Actual-value coupling" (実際値カップリング) (DB31 ~ 48, DBX98.2)</p>	

21320 MD 番号	COUPLE_BLOCK_CHANGE_CTRL_1 主軸同期モードにおけるブロック変更応答	
デフォルト値: 3	最小入力リミット: 0	最大入力リミット: 3
変更が有効になるための条件: 電源オン	保護レベル: 2/7	単位: -
データタイプ: BYTE	適用ソフトウェアバージョン: 2.1	
意味:	<p>チャンネル別マシンデータ COUPLE_AXIS_1[n] で定義される固定カップリング構成に合わせて同期モードが起動させられる場合、このマシンデータは、どのような条件で、ブロック変更が実行されるべきか判定する。</p> <p>次のオプションが使用できる。</p> <p>0: ブロック変更が直ちにイネーブルされる。</p> <p>1: "Fine synchronism" (微同期) に対するブロック変更応答</p> <p>2: "Coarse synchronism" (粗同期) に対するブロック変更応答</p> <p>3: IPOSTOP (即ち、セットポイントベースの同期後) に対するブロック変更応答</p> <p>チャンネル別 MD: COUPLE_IS_WRITE_PROT_1 でこのオプションが阻止されていない限り、NC パートプログラムで言語命令 COUPDEF を使用すれば、ブロック変更応答を変更することができる。ただし、チャンネル別 MD: COUPLE_BLOCK_CHANGE_CTRL_1 のパラメータ値は、変更されない。</p> <p>選択したブロック変更応答は、カップリングがアクティブである限り、定義された角度オフセットがプログラムされた時や、速度比が変更された時でも有効のままである。</p>	
この MD は次の場合は無効	ユーザ定義カップリング	
例外, エラー, ....	<p>チャンネル別 MD: COUPLE_AXIS_1 (主軸同期ペアの定義)</p> <p>チャンネル別 MD: COUPLE_IS_WRITE_PROT_1 (カップリングパラメータ変更不可能)</p> <p>チャンネル別 MD: COUPLE_POS_TOL_COARSE または COUPLE_VELO_TOL_COARSE (粗同期用のスレッシュホールド値)</p> <p>チャンネル別 MD: COUPLE_POS_TOL_FINE または COUPLE_VELO_TOL_FINE (微同期用のスレッシュホールド値)</p>	

21330 MD 番号	COUPLE_RESET_MODE_1 カップリング強制終了応答	
デフォルト値: 1	最小入力リミット: 0	最大入力リミット: 0x3FF
変更が有効になるための条件: 電源オン	保護レベル: 2/7	単位: -
データタイプ: BYTE	適用ソフトウェアバージョン: 2.1	
意味:	<p>マシンデータ COUPLE_AXIS_n で構成された主軸同期ペアの同期の挙動は、このマシンデータで定義される。</p> <p>ビット 0=0: 新規にプログラムが起動しても、同期はアクティブのままである。制御がオンのままでは、COUPOF を使用しない限り、取消することができない。</p> <p>ビット 0=1: (リセット条件から) プログラム起動とともに同期が取消される。</p> <p>ビット 1=0: プログラム終了とリセットが起こっても、同期がアクティブのままである。制御がオンのままでは、COUPOF を使用しない限り、取消することができない。</p> <p>ビット 1=1: プログラム終了またはリセットが起こると、同期が取消される。</p> <p>ビット 5=1: プログラムが起動すると、構成データも起動させられる。</p> <p>ビット 6=1: プログラム終了またはリセットが起こると、構成データが起動させられる。</p> <p>ビット 9=1: プログラムが起動すると、同期がオンになる。</p> <p>注記: NC 停止の後で、NC がスタートしても、同期は選択解除されない。</p>	
この MD は次の場合は無効	ユーザ定義カップリング	
例外, エラー, ....	チャンネル別 MD; COUPLE_AXIS_1 (主軸同期ペアの定義) IS "Synchronous operation" (同期運転) (DB31 ~ 48, DBX84.4)	

21340 MD 番号	COUPLE_IS_WRITE_PROT_1 カップリングパラメータが書き込み保護される	
デフォルト値 : 0	最小入力リミット : 0	最大入力リミット : 1
変更が有効になるための条件 : 電源オン	保護レベル : 2/7	単位 : -
データタイプ : BOOLEAN	適用ソフトウェアバージョン : 2.1	
意味 :	<p>このマシンデータによって、チャンネル別マシンデータ COUPLE_AXIS_1[n] で構成される主軸同期ペア用のカップリングパラメータ（速度比、ブロック変更応答、カップリングタイプ）が NC パートプログラムで変更できるかどうかを指定する。</p> <p>1: カップリングパラメータは、NC プログラム（書き込み保護アクティブ）によって変更することができない。パラメータを変更しようとすると、アラームメッセージが出される。</p> <p>0: NC パートプログラムは、言語命令 COUPDEF でカップリングパラメータを変更できる。</p>	
この MD は次の場合は無効	ユーザ定義カップリング	
例外, エラー, ....	<p>チャンネル別 MD: COUPLE_AXIS_1 (主軸同期ペアの定義)</p> <p>チャンネル別 MD: COUPLING_MODE_1 (主軸同期モードのカップリングタイプ)</p> <p>チャンネル別 MD: COUPLE_RESET_MODE_1 (カップリング強制終了応答)</p> <p>チャンネル別 MD: COUPLE_BLOCK_CHANGE_CTRL_1 (主軸同期モードにおけるブロック変更応答)</p> <p>SD: \$SC_COUPLE_RATIO_1 (主軸同期モードの速度比パラメータ)</p>	

## ■ 軸別マシンデータ

37200 MD 番号	COUPLE_POS_TOL_COARSE 粗同期用のスレッシュホールド値	
デフォルト値 : 1.0	最小入力リミット : 0.0	最大入力リミット : PLUS
変更が有効になるための条件 : NEW_CONF	保護レベル : 2/7	単位 : 直線軸 : mm 回転軸 : 度
データタイプ : DOUBLE	適用ソフトウェアバージョン : 2.1	
意味 :	<p>同期運転で、リードおよび追従スピンドルの間の位置偏差が監視される (DV モードと AV モードのみ)。</p> <p>スレッシュホールド値が指定した公差範囲内に現在の位置偏差が入っている時には、IS "Coarse synchronism" (粗同期) がセットされる。</p> <p>更に、このスレッシュホールド値は、"Coarse synchronism" (粗同期) がブロック変更応答条件として選択されている場合に、カップリングがアクティブになる時の同期運転の起動または伝送パラメータの変更のブロック変更基準を表している (チャンネル別 MD: COUPLE_BLOCK_CHANGE_CTRL_1 または言語命令 COUPDEF を参照)。</p> <p>"0" の値が入力される時、DV および AV モードで IS "Coarse synchronism" (粗同期) が必ず "1" にセットされる。</p>	
例外, エラー, ....	<p>チャンネル別 MD: COUPLE_BLOCK_CHANGE_CTRL_1 (主軸同期運転におけるブロック変更応答)</p> <p>IS "Coarse synchronism" (粗同期) (DB31 ~ 48, DBX98.1)</p>	

37210 MD 番号	COUPLE_POS_TOL_FINE 微同期用のスレッシュヨルド値	
デフォルト値 : 0.5	最小入力リミット : 0.0	最大入力リミット : PLUS
変更が有効になるための条件 : NEW_CONF	保護レベル : 2/7	単位 : 直線軸 : mm 回転軸 : 度
データタイプ : DOUBLE	適用ソフトウェアバージョン : 2.1	
意味 :	<p>同期運転では、リードおよび追従スピンドルの間の位置偏差が監視される (DV および AV モードのみ)。</p> <p>スレッシュヨルド値が指定する公差範囲内に現在の位置偏差が入っている時、IS "Fine synchronism" (微同期) がセットされる。</p> <p>更に、このスレッシュヨルド値は、"Fine synchronism" (微同期) がブロック変更応答条件として選択される場合に、カップリングがアクティブになる時の同期運転の起動および伝送パラメータの変更のブロック変更基準を表している (チャンネル別 MD: COUPLE_BLOCK_CHANGE_CTRL_1 または言語命令 COUPDEF を参照)。</p> <p>"0" の値が入力される時、DV および AV モードで、IS "Fine synchronism" (微同期) が必ず "1" にセットされる。</p>	
例外, エラー, ....	<p>チャンネル別 MD; COUPLE_BLOCK_CHANGE_CTRL_1 (主軸同期運転におけるブロック変更応答)</p> <p>IS "Fine synchronism" (微同期) (DB31 ~ 48, DBX98.0)</p>	

37220 MD 番号	COUPLE_VELO_TOL_COARSE リードおよび追従スピンドルの間の "Coarse" (粗) 速度公差	
デフォルト値 : 1.0	最小入力リミット : 0.0	最大入力リミット : PLUS
変更が有効になるための条件 : NEW_CONF	保護レベル : 2/7	単位 : 直線軸 : mm/min 回転軸 : rpm
データタイプ : DOUBLE	適用ソフトウェアバージョン : 3.1	
意味 :	<p>同期運転では、リードおよび追従スピンドルの間の速度差が監視される (VV モードのみ)。</p> <p>スレッシュヨルド値が指定する公差範囲内に現在の速度差が入っている時、IS "Coarse synchronism" (粗同期) がセットされる。</p> <p>更に、このスレッシュヨルド値は、"Coarse synchronism" (粗同期) がブロック変更応答条件として選択される場合に、カップリングがアクティブになる時の同期運転の起動または伝送パラメータの変更のブロック変更基準を表している (チャンネル別 MD: COUPLE_BLOCK_CHANGE_CTRL_1 または言語命令 COUPDEF を参照)。</p> <p>"0" の値が入力される時、VV モードで、IS "Coarse synchronism" (粗同期) が必ず "1" にセットされる。</p>	
例外, エラー, ....	<p>チャンネル別 MD; COUPLE_BLOCK_CHANGE_CTRL_1 (主軸同期運転におけるブロック変更応答)</p> <p>IS "Coarse synchronism" (粗同期) (DB31 ~ 48, DBX98.1)</p>	

37230 MD 番号	COUPLE_POS_TOL_FINE リードおよび追従スピンドルの間の "Fine" (微) 速度公差	
デフォルト値 : 0.5	最小入力リミット : 0.0	最大入力リミット : PLUS
変更が有効になるための条件 : NEW_CONF	保護レベル : 2/7	単位 : 直線軸 : mm/min 回転軸 : rpm
データタイプ : DOUBLE	適用ソフトウェアバージョン : 3.1	
意味 :	同期運転では、リードおよび追従スピンドルの間の速度差が監視される (VV モードのみ)。 スレッシュホールド値が指定する公差範囲内に現在の速度差が入っている時、IS "Fine synchronism" (微同期) がセットされる。 更に、このスレッシュホールド値は、"Fine synchronism" (微同期) がブロック変更応答条件として選択される場合に、カップリングがアクティブになる時の同期運転の起動または伝送パラメータの変更のブロック変更基準を表している (チャンネル別 MD: COUPLE_BLOCK_CHANGE_CTRL_1 または言語命令 COUPDEF を参照)。 "0" の値が入力される時、VV モードで、IS "Fine synchronism" (微同期) が必ず "1" にセットされる。	
例外, エラー, ....	チャンネル別 MD: COUPLE_BLOCK_CHANGE_CTRL_1 (主軸同期運転におけるブロック変更応答) IS "Fine synchronism" (微同期) (DB31 ~ 48, DBX98.0)	

## ■ 設定データの説明

42300 SD 番号	COUPLE_RATIO_1[n] 主軸同期モード [n] の速度比パラメータ	
デフォルト値 : 1.0	最小入力リミット : -1000	最大入力リミット : 1000
変更が有効になるための条件 : NEW_CONF	保護レベル : MMC-MD 9220	単位 : -
データタイプ : DOUBLE	適用ソフトウェアバージョン : 2.1	
意味 :	この設定データは、チャンネル別 MD: COUPLE_AXIS_1[n] で定義された固定カップリング構成用の速度比パラメータを判定する。 リードおよび追従スピンドル間の直線相関は、速度比 $k_U$ によって判定される。2つの速度比パラメータを分子 [n=0] と分母 [n=1] の形に指定してこの比を入力することによって、かなり正確に速度比が指定できるようにする。 $k_U \left[ \frac{\text{速度パラメータ 分子}}{\text{速度パラメータ 分母}} \right] \frac{\$SC\_COUPLE\_RATIO[0]}{\$SC\_COUPLE\_RATIO[1]}$ チャンネル別 COUPLE_IS_WRITE_PROT_1 でこのオプションが阻止されていない限り、NC パートプログラムで、命令 COUPDEF によって速度比パラメータを変更することができる。ただし、SD: \$SC_COUPLE_RATIO_1 のパラメータ値は変更されない。電源オンで、 $k_U$ の計算が開始する。	
この SD は次の場合は無効	ユーザ定義カップリング	
例外, エラー, ....	SD: \$SC_COUPLE_RATIO_1 は、現在、マシンデータと同じアクションを示している (たとえば、電源オン後にアクティブになる)。従って、SD データは、チャンネル別マシンデータと同様に表示されて入力される。	
参照	チャンネル別 MD: COUPLE_AXIS_1 (主軸同期ペアの定義)	

## 2.13.5 信号の説明

### ■ 軸／スピンドル別信号

### ■ 軸／スピンドルからの信号

DB31 ～ 48 DBX84.4 データブロック	Synchronous mode（同期モード） 軸／スピンドルから PLC への信号 (NCK +/- PLC)	
エッジ評価：なし	信号更新：周期的	信号が有効となる最初の SW バージョン：2.1
信号ステータス 1 あるいは 信号遷移 0 ---> 1	スピンドルは, "Synchronous operation"（同期運転）モードで動作している。 従って, 追従スピンドルは, 伝送比に従って, リードスピンドルのモーショ ンに追従している。 粗／微同期用の監視ファンクションは, 同期運転中に実行される。 注記：追従スピンドルとして働くマシン軸用のみに関して, 信号がセットされ る（IS "FS active"（FS アクティブ）= 1）。	
信号ステータス 0 あるいは 信号遷移 1 ---> 0	スピンドルは, "synchronous mode"（同期モード）で, 追従スピンドルとして 運転されない。 （同期運転の選択解除によって）カップリングが停止させられる時, 追従スピ ンドルが, "open-loop control mode"（開ループ制御モード）に切換えられる。	
関連性 ....	IS "Coarse synchronism"（粗同期）(DB31 ～ 48, DBX98.1)IS "Fine synchronism"（微同期）(DB31 ～ 48, DBX98.0)IS "FS active"（FS アクティブ）(DB31 ～ 48, DBX99.1)	

DB31 ～ 48 DBX98.0 データブロック	Fine synchronism（微同期） 軸／スピンドルから PLC への信号 (NCK +/- PLC)	
エッジ評価：なし	信号更新：周期的	信号が有効となる最初の SW バージョン：2.1
信号ステータス 1 あるいは 信号遷移 0 ---> 1	追従スピンドルおよびリードスピンドルの間の位置偏差は, "Fine synchronism"（微同期）公差範囲内にある（2.13.2「■同期運転の監視」を参 照）。	
信号ステータス 0 あるいは 信号遷移 1 ---> 0	追従スピンドルおよびリードスピンドルの間の位置偏差は, "Fine synchronism"（微同期）範囲内にある（2.13.2「■同期運転の監視」を参照）。 注記：同期運転で, 追従スピンドル用の信号だけが関連する。	
アプリケーション例	リードスピンドルからの転送時に, 追従スピンドルでワークピースをクラン プする。スピンドルが十分に同期させられるまで, PLC ユーザプログラムは ワークピースのクランプを開始しない。	
関連性 ....	IS "Synchronous operation"（同期運転）(DB31 ～ 48, DBX84.4) MD: \$MA_COUPLE_POS_TOL_FINE 微同期用のスレッシュホールド値または MD: \$MA_COUPLE_VELO_TOL_FINE "fine"（微）速度公差	



DB31 ～ 48 DBX98.1 データブロック	Coarse synchronism (粗同期) 軸／スピンドルから PLC への信号 (NCK +/- PLC)	
エッジ評価：なし	信号更新：周期的	信号が有効となる最初の SW バージョン：2.1
信号ステータス 1 あるいは 信号遷移 0 ---> 1	追従スピンドルおよびリードスピンドルの間の位置偏差は, "Coarse synchronism" (粗同期) 公差範囲内にある (2.13.2 「 <b>■</b> 同期運転の監視」を参照)。 注記：同期運転で、追従スピンドル用の信号だけが関連する。	
信号ステータス 0 あるいは 信号遷移 1 ---> 0	追従スピンドルおよびリードスピンドルの間の位置偏差は, "Coarse synchronism" (粗同期) 公差範囲内にある (2.13.2 「 <b>■</b> 同期運転の監視」を参照)。	
アプリケーション例	リードスピンドルからの転送時に、追従スピンドルでワークピースをクランプする。スピンドルが十分に同期させられるまで、PLC ユーザプログラムはワークピースのクランプを開始しない。	
関連性 ....	IS "Synchronous operation" (同期運転) (DB31 ～ 48, DBX84.4) 軸別 MD: COUPLE _POS _TOL _COARSE 粗同期用のスレッシュホールド値 または 軸別 MD: COUPLE _VELO _TOL _COARSE "coarse" (粗) 速度公差	

DB31 ～ 48 DBX98.2 データブロック	Actual-value coupling (実際値カップリング) 軸／スピンドルから PLC への信号 (NCK +/- PLC)	
エッジ評価：なし	信号更新：周期的	信号が有効となる最初の SW バージョン：2.1
信号ステータス 1 あるいは 信号遷移 0 ---> 1	実際値カップリングは、リードおよび追従スピンドル間のカップリングタイプとしてアクティブである (チャンネル別 MD: COUPLING _MODE_1 を参照)。 注記：同期運転で、アクティブ追従スピンドル用の信号だけが関連する。	
信号ステータス 0 あるいは信号 遷移 1 ---> 0	セットポイントカップリングは、リードおよび追従スピンドルの間のカップリングタイプとしてアクティブである (チャンネル別 MD: COUPLING _MODE_1 を参照)。	
例外, エラー, ....	追従スピンドルの故障／障害によって、FS "servo enable" (サーボイネーブル) が取消された場合には、FS と LS との間のカップリング関係が、一定の状況の下に、制御内で逆転されて、実際値カップリングに切換えられる。	
関連性 ....	IS "Synchronous operation" (同期運転) (DB31 ～ 48, DBX84.4) チャンネル別 MD: COUPLING _MODE_1 (主軸同期運転におけるカップリングタイプ)	

DB31 ～ 48 DBX98.4 データブロック	Overlaid motion (オーバレイモーション) 軸／スピンドルから PLC への信号 (NCK +/- PLC)	
エッジ評価：なし	信号更新：周期的	信号が有効となる最初の SW バージョン：2.1
信号ステータス 1 あるいは 信号遷移 0 ---> 1	<p>追従スピンドルは、リードスピンドルとのカップリングからのモーションにオーバレイされて追加されたモーションコンポーネントをトラバースする。</p> <p>FS のオーバレイ動作の例：</p> <ul style="list-style-type: none"> <li>・ FS と LS との間で定義された角度オフセットで、同期運転を起動する。</li> <li>・ LS を回転して、同期運転を起動する。</li> <li>・ 同期運転が選択される時に、伝送比を変更する。</li> <li>・ 同期運転が選択される時に、新規に定義された角度オフセットを入力する。</li> <li>・ JOG で、同期運転が選択される時に、プラスまたはマイナスのトラバースキーまたはハンドルで FS をトラバースする。</li> </ul> <p>FS がオーバレイ動作を実行した時、(スレッシュホールド値に応じて) IS "Fine synchronism" (微同期) または IS "Coarse synchronism" (粗同期) が直ちに取消される場合もある。</p> <p>注記：同期運転で、追従スピンドル用の信号だけが関連する。</p>	
信号ステータス 0 あるいは 信号遷移 1 ---> 0	追加されたモーションコンポーネントを追従スピンドルがトラバースしないか、このモーションが終了させられている。	
関連性 ....	IS "Synchronous operation" (同期運転) (DB31 ～ 48, DBX84.4)	

DB31 ～ 48 DBX99.0 データブロック	LS (leading spindle) active (LS (リードスピンドル) アクティブ) 軸／スピンドルから PLC への信号 (NCK +/- PLC)	
エッジ評価：なし	信号更新：周期的	信号が有効となる最初の SW バージョン：2.1
信号ステータス 1 あるいは 信号遷移 0 ---> 1	現在、マシン軸が、リードスピンドルとしてアクティブである。 注記：同期運転中にのみ、信号が関連する。	
信号ステータス 0 あるいは信号 遷移 1 ---> 0	現在、マシン軸が、リードスピンドルとしてアクティブでない。	
関連性 ....	<p>追従スピンドルの故障／障害によって、FS "servo enable" (サーボイネーブル) が取消される場合には、FS と LS との間のカップリング関係が、一定の状況の下に、制御内で逆転されて実際値カップリングに切換えられる。</p> <p>この場合、リードスピンドルが、新規のアクティブな追従スピンドルになる (IS "FS active" (FS アクティブ))。</p>	
関連性 ....	IS "Synchronous operation" (同期運転) (DB31 ～ 48, DBX84.4) IS "FS active" (FS アクティブ) (DB31 ～ 48, DBX99.1)	

DB31 ～ 48 DBX99.1 データブロック	FS (following spindle) active (FS (追従スピンドル) アクティブ) 軸／スピンドルから PLC への信号 (NCK +/- PLC)	
エッジ評価 : なし	信号更新 : 周期的	信号が有効となる最初の SW バージョン : 2.1
信号ステータス 1 あるいは 信号遷移 0 ---> 1	現在, マシン軸は, 追従スピンドルとして動作している。従って, 追従スピンドルは, 同期運転で, 伝送比に従ってリードスピンドルの動きに追従する。 注記 : 本信号は同期運転にのみ関連する。	
信号ステータス 0 あるいは 信号遷移 1 ---> 0	現在, マシン軸は, 追従スピンドルとして動作していない。	
関連性 ....	追従スピンドルの故障／障害によって, FS "servo enable" (サーボイネーブル) が取消される場合には, FS と LS との間のカップリング関係が, 一定の状況の下に, 制御内で逆転されて実際値カップリングに切換えられる。	
関連性 ....	IS "Synchronous operation" (同期運転) (DB31 ～ 48, DBX84.4) IS "LS active" (LS アクティブ) (DB31 ～ 48, DBX99.0)	

## 2.13.6 例

```

; リードスピンドル = マスタスピンドル =
; スピンドル 1
; 追従スピンドル = スピンドル 2
N05 M3 S3000 M2=4 S2=500 ; リードスピンドルは、 3000/min で回転します。
; FS: -500/min.
N10 COUPDEF (S2, S1, 1, 1, "No", "Dv") ; カップリング定義の構成も可能です。
...
N70 SPCON ; リードスピンドルを位置制御に切換えます。
; (セットポイントカップリング)
N75 SPCON(2) ; 追従スピンドルを位置制御に切換えます。
N80 COUPON (S2, S1, 45) ; 架空カップリング
; オフセット位置へ = 45 度
...
N200 FA [S2] = 100 ; 位置決め速度 = 100 度 /min
N205 SPOS[2] = IC(-90) ; 負の方向へ 90 度オーバレイ動作します。
N210 WAITC(S2, "Fine") ; "fine" (微) 同期を待機します。
N212 G1 X., Y. F... ; 処理
...
N215 SPOS[2] = IC(180) ; 正の方向へ 180 度オーバレイ動作します。
N220 G4 S50 ; ドエル = マスタスピンドルの 50 回転。
N225 FA [S2] = 0 ; 構成速度を起動します (MD)。
N230 SPOS[2] = IC (-7200) ; 負の方向へ、構成速度で 20 回転します。
...
N350 COUPOF (S2, S1) ; 架空デカップリング, S = S2 = 3000
N355 SPOSA[2] = 0 ; 0 度で、FS を停止します。
N360 G0 X0 Y0
N365 WAITS(2) ; スピンドル 2 を待機します。
N370 M5 ; FS を停止します。
N375 M30
```

## 2.13.7 データフィールド, リスト

### ■ インタフェース信号

DB 番号	ビット, バイト	名称	参照
チャンネル別			
21, ...	7.1	NC start (NC 開始)	K1
21, ...	7.4	NC stop axes plus spindle (NC 停止軸プラススピンドル)	K1
21, ...	24.6	Dry run feedrate selected (ドライランフィードレート選択)	V1
21, ...	25.3	Feedrate override for rapid traverse selected (急速トラバース用フィードレートオーバーライド選択)	V1
軸／スピンドル別			
31, ...	1.3	Axis/spindle disable (軸／スピンドルディスエーブル)	A2
31, ...	1.4	Follow-up mode (追従モード)	A2
31, ...	1.5/1.6	Position measuring system 1, position measuring system 2 (位置測定系 1, 位置測定系 2)	A2
31, ...	2.1	Servo enable (サーボイネーブル)	A2
31, ...	2.2	Spindle RESET (スピンドルリセット)	A2
31, ...	4.3	Spindle stop/feed stop (スピンドル停止／フィード停止)	V1
31, ...	4.6 ~ 4.7	Traversing keys for JOG (JOG 用のトラバースキー)	V1
31, ...	16.4/16.5	Re-synchronize spindle 1, re-synchronize spindle 2 (スピンドル再同期 1, スピンドル再同期 2)	S1
31, ...	16.7	Delete S value (S 値の削除)	S1
31, ...	17.0	Feedrate override valid (フィードレートオーバーライド有効)	S1
31, ...	19	Spindle speed override (スピンドル速度オーバーライド)	V1
31, ...	60.4/60.5	Referenced/synchronized 1, referenced/synchronized 2 (基準化／同期 1, 基準化／同期 2)	R1
31, ...	84.4	Synchronous operation (同期運転)	
31, ...	98.0	Fine synchronism (微同期)	
31, ...	98.1	Coarse synchronism (粗同期)	
31, ...	98.2	Actual-value coupling (実際値カップリング)	
31, ...	98.4	Overlaid movement (オーバーレイ動作)	
31, ...	99.0	LS/LA active (LS/LA アクティブ)	
31, ...	99.1	FS/FA active (FS/FA アクティブ)	

## ■ マシンデータ

番号	識別子	名称	参照
一般事項 (\$MN _...)			
10000	AXCONF_MACHAX_NAME_TAB	マシン軸名称	K2
チャンネル別 (\$MC _...)			
21300	COUPLE_AXIS_1	主軸同期ペアの定義	
21320	COUPLE_BLOCK_CHANGE_CTRL_1	主軸同期運転におけるブロック変更応答	
21310	COUPLE_IS_DES_POS_1	主軸同期運転におけるカップリングタイプ	
21330	COUPLE_RESET_MODE_1	カップリング強制終了応答	
21340	COUPLE_IS_WRITE_PROT_1	カップリングパラメータの書き込み保護	
20070	AXCONF_MACHAX_USED	チャンネルで有効なマシン軸番号	K2
軸／スピンドル別 (\$MA _...)			
30550	AXCONF_ASSIGN_MASTER_CHAN	軸変更用のチャンネルの初期設定	K5
32200	POSCTRL_GAIN	サーボゲイン係数	G2
32400	AX_JERK_ENABLE	軸ジャーク制限	B2
32410	AX_JERK_TIME	軸ジャークフィルタの時定数	B2
32420	JOG_AND_POS_JERK_ENABLE	軸ジャーク制限の基本設定	B2
32430	JOG_AND_POS_MAX_JERK	軸ジャーク	B2
32610	VELO_FFW_WEIGHT	速度フィードフォワード制御用のフィードフォワード制御係数	K3
32620	FFW_MODE	フィードフォワード制御タイプ	K3
32650	AX_INERTIA	トルクフィードフォワード制御慣性モーメント	K3
32800	EQUIV_CURRCTRL_TIME	フィードフォワード制御用の等価時定数, 現在の制御ループ	K3
32810	EQUIV_SPEEDCTRL_TIME	フィードフォワード制御用の等価時定数, 速度制御ループ	K3
34080	REFP_MOVE_DIST	基準点距離	R1
34090	REFP_MOVE_DIST_CORR	基準点オフセット	R1
34100	REFP_SET_POS	基準点の値	R1
35000	SPIND_ASSIGN_TO_MACHAX	マシン軸へのスピンドルの割当て	S1
37200	COUPLE_POS_TOL_COARSE	粗同期用のスレッシュホールド値	
37210	COUPLE_POS_TOL_FINE	微同期用のスレッシュホールド値	
37220	COUPLE_VELO_TOL_COARSE	リードおよび追従スピンドルの間の速度公差 "coarse" (粗)	
37230	COUPLE_VELO_TOL_FINE	リードおよび追従スピンドルの間の速度公差 "fine" (微)	

---

■ 設定データ

番号	識別子	名称	参照
軸別 (\$SA_...)			
42300	COUPLE_RATIO_1	主軸同期運転用の伝送パラメータ	

---

## ■ アラーム

アラームの詳細説明については次を参照してください。

参照： アラーム一覧

あるいは MMC 101/102/103 を有するシステムのオンラインヘルプ



---

## 2.14 メモリ構成 (S7)

### 2.14.1 概略説明

どの CNC にも、データの保存や管理用のメモリが必要です。このメモリは基本的には 2 つのエリアに分けられます。1 つは、変化しないデータ、例えば CNC のソフトウェアなどを格納するエリアです。このタイプのデータは、EPROM のような電子メモリチップに保存します。もう 1 つは、マシンの製造業者やユーザが制御装置上に保存しているデータを格納するエリアです。このようなデータは、RAM のような電子メモリチップに保存されます。制御系を使用して、ユーザはさまざまな仕様に対して RAM エリアを設定できます。ここでは、ユーザが使用できる RAM のエリアとその設定方法を説明します。

(注)

現在使用可能な SRAM メモリは、プログラム概要（ダイアログライン）のプログラム操作領域に表示されます。

## 2.14.2 詳細説明

### ■ 一般事項

#### 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は CNC のメインメモリです。ここには、制御装置をオペレーションするのに必要な現在のシステムやユーザのデータ（例えば、マシンデータ）が保存されています。

#### パッシブファイルシステム

パッシブファイルシステムは、データやプログラムをバッファするために使用される制御系のメモリエリアです。パッシブファイルシステムは階層別の構成になっていて、ディレクトリとサブディレクトリで構成されています。

#### SRAM

SRAM という用語は、ユーザがデータのバックアップに使用することができる制御系のスタティック RAM を意味します。SRAM はまた、本文書では、バックアップメモリ、非揮発性メモリ、またはスタティックメモリを指す場合もあります。現在利用可能な SRAM メモリは、プログラム概要（ダイアログライン）のプログラム操作領域に表示されます。

#### DRAM

DRAM という用語は、ユーザが利用できる制御系のダイナミック RAM を意味します。このエリアで使用されるデータは、制御装置のみが作成し、一時的にしか必要とされないため、バックアップを必要としません。DRAM は、本文書では揮発性メモリまたはダイナミックメモリを指す場合もあります。

### ■ メモリ構成

#### ハードウェア構成

次の表は、利用可能な NC-CPU のハードウェア構成を示したものです。

	D-RAM	S-RAM バッファされていない	S-RAM バッファされている	FLASH	JEIDA
NCU 570		1.5 MB	0.25 MB	2.25 MB	
NCU 571	4 MB		0.5 MB/2.0 MB*		4 MB
NCU 572	8 MB		0.5 MB/2.0 MB*		4 MB
NCU 573	8 MB		0.5 MB/2.0 MB*		4 MB
NCU 573.2	8 MB		2.0 MB		4 MB
NCU 573.2	32 MB*		2.0 MB		4 MB

## SRAM

ユーザが使用することができるのが SRAM です。本書に述べてあるマシンデータを  
使用して構成できます。

- NCU 570            128 KB
- NCU 571            256 KB/ 1.5 MB\*
- NCU 572            256 KB/ 1.5 MB\*
- NCU 573            256 KB/ 1.5 MB\*
- NCU 573.2        1.5 MB

\*) オプションです。Catalog NC 60.1（カタログ NC 60.1）を参照してください。

バッファされたユーザメモリは、MD 18230: MM\_USER\_MEM\_BUFFERED に KB  
単位で入力されます。NCU が異なる 2 つのメモリ容量を持つ場合、標準エントリ  
には最小値を使用する必要があります。2.0 MB バージョンを使用しているなら、  
MD 18230 を 1900 に設定してください。（総値が 2000 であっても、内部使用には  
減らす必要があります）。

## DRAM

ユーザが使用することができる DRAM の総容量は、MD 18210:

MM\_USER\_MEM\_DYNAMIC ( DRAM のダイナミックユーザメモリ ) に表示されま  
す。その値はシステムに依存し、ソフトウェアバージョンによっても幾分異なり  
ます。

個々のデータグループ（例えば、グローバルユーザデータ、チャンネル別ユーザ  
データ、軸別ユーザデータなど）が入っているメモリエリアは、SRAM と DRAM  
に連続して配置されます。使用するデータエリアの大きさは、マシンデータに設  
定することができます。データエリアが配置される順序は常に CNC によって定義  
されます。

## メモリエリアの変更

上記のメモリ構成から、メモリエリアにどのような変更が加えられた場合でも保  
存データに影響を与えることは明白です。システムが起動する度に、CNC は、  
個々のデータエリアに関するマシンデータを基に、現在のメモリ要求と既存のメ  
モリ空間を比較します。この比較によって、1 つまたは複数の修正データエリアが  
エリアの再配置を必要とすることが確認された場合、メモリは再編成されます。

## メモリ再編成中のデータの喪失

メモリエリアの再編成時には、バックアップメモリの全データが喪失します。

SRAM と DRAM 中のそれぞれのメモリエリアが合計で利用可能なメモリ空間を超  
過した場合にも、バックアップデータは喪失します。したがって、メモリ構成を  
変更する前に、CNC に保存されているデータを保存することが絶対に不可欠です。  
割当てに利用できる SRAM メモリは、マシンデータ中で識別子  
INFO\_FREE\_MEM\_ が付いています。

## メモリ構成マシンデータ

次に、メモリ構成に影響を及ぼすマシンデータをいくつか挙げます。

- システム別メモリ設定
- チャンネル別メモリ設定
- 軸別メモリ設定

これらのグループに属する SRAM 用のマシンデータを変更すると必ずバックアップメモリ内のデータが喪失します。このようなマシンデータには、MM\_ で始まる名前（例えば MM\_NUM\_TOA\_MODULES）がついています。

また、有効チャンネルの数もメモリ構成に影響を与えます。チャンネル数が増えたと、変更されたチャンネルは、システムが起動する際に、チャンネル別メモリエリア用の初期設定に従ってセットアップされます。これらのエリアは SRAM 内でもあるため、チャンネル数の変更によりバックアップユーザメモリ内のデータも喪失します。

新規の値が、SRAM のメモリエリアを再配置するマシンデータへ入力されると（「■ SRAM および DRAM のメモリ割当て」を参照）、メッセージ 4400 "MD alteration will cause reorganization of buffer (data loss!)" が出力されます。このメッセージは、マシンデータが増えられ、それにより NCK が起動した際にバックアップメモリが再編成され、その結果そのメモリに保存されているユーザデータが全て失われることを警告しています。メモリが再編成され、かつ制御装置がバックアップされていないデータを持つ場合、NCK を終了する前に、必ずこれらのデータを保存しておいてください。

(注)

修正値を前回に起動した時点の元の値に変更し直すことにより、再編成を回避することができます。

MD 変更によりメモリの再編成が行われない例外的な場合もあります。

MD 18350: MM\_USER\_FILE\_MM\_MINIMUM（最小パートプログラムメモリ）の場合、メモリの再編成は RAM の空き量が小さすぎる場合にのみ行われます。

システム別 MD 11200: INIT\_MD（次の起動時に標準マシンデータをロードする）を設定して、次のシステム起動時に、メモリ設定標準データを 2 にロードすると、メモリエリアがシステムの起動プロセスに先立ちメモリ初期設定に従って編成されていない場合、バックアップしてあったユーザデータが喪失します。

## ■ メモリ構成アラーム

不正であったり、メモリ構成を必要とするメモリ割当てに修正を加えると、CNCシステムの起動後にアラームメッセージが出力されます。誤りの原因およびCNCの応答は次のとおりです：

### アラーム 6000

ユーザメモリ（スタティクまたはダイナミック）は、利用可能な総メモリ領域（スタティクまたはダイナミック）がマシンデータによってセットされているメモリ領域の総数より小さいので、再割当てすることができません。この場合、メモリを設定するために必要なマシンデータは全て削除され、その初期値が割当てられます。

NC加工はできません。この場合、アラーム 6000 "Memory allocation with standard machine data"（標準マシンデータを使用するメモリ割当て）が表示されます。特定のマシンデータをメモリ割当てのエラーの原因として正確に特定することはできません。しかし、メモリ設定用のマシンデータを一つずつ変更することによってエラーを見つけることができます。アラームは RESET でキャンセルできます。ユーザデータがロードされている場合に限りマシニングが可能です。

### アラーム 6010

サイクルプログラム、マクロ定義、またはグローバルユーザデータの定義が組み込まれると、アラーム 6010 "Channel [name 1] data block [name 2] has not been or is only partially created, error number [identifier]" が、エラー内容に応じて出力されます。対応するメモリエリア用のマシンデータが不正に設定されているか、あるいはファイルにエラーが含まれているかのいずれかです。例えば、マクロ定義 `_N_SMAC_DEF`, `_N_MMAC_DEF` および `_N_UMAC_DEF` 用の 3 つのファイルに、総計 30 のマクロ定義が含まれているにもかかわらず、MD 18160: `MM_NUM_USER_MACROS`（マクロ数）の設定がマクロ数を 10 に制限している場合などです。

識別子 [名称 1] は、エラーが発生したチャンネル名を示しています。識別子 [名称 2] は、エラーがあるファイルの名前を示しています。エラー番号は原因によって次のようにコード化されています。

エラー番号	説明
1	利用できるメモリがない
2	最大記号数を超過している
3	インデックス 1 が値の許容範囲を超過している
4	名前がすでにチャンネル上に存在している
5	名前がすでに NCK 上に存在している
> 100000	修復不可能なシステムエラー

エラー番号出力が 1～5 であれば、ユーザは独力でエラーを修復することができます。しかし、エラー番号が > 100000 である場合、そのエラーは修復不可能なシステムエラーです。

マシンデータまたはファイルがすでに修正されている場合、または変更がキャンセルされシステムが再起動されている場合、マシニングは可能です。

---

## アラーム 6020

スタティックメモリ割当てに変更が加えられると SRAM が再編成されます。保存データはマシンデータを除きすべて喪失します。この場合、アラーム 6020

"Machine data altered - memory reallocated" が出力されます。CNC チャンネル番号あるいは、スタティックメモリ用のシステム別、チャンネル別または軸別のメモリ設定が変更されると、SRAM が再割当てされます。アラームは RESET でキャンセルできます。マシニングは、ユーザデータがロードされると再開できます。

## アラーム 6030

MD 18210: MM\_USER\_MEM\_DYNAMIC (DRAM のユーザメモリ), MD 18220: MM\_USER\_MEM\_DPR (デュアルポート RAM のユーザメモリ) または MD 18230: MM\_USER\_MEM\_BUFFERED (SRAM のユーザメモリ) のメモリエリアは、実際に利用することができる実質メモリよりも大きく設定されています。この場合、CNC は対応するマシンデータに利用可能なメモリを入力し、アラーム 6030 "User memory limit has been adapted" でそのことを示します。この場合はユーザメモリの喪失は起こりません。アラームは RESET を使用してキャンセルすることができます。しかし、マシンデータが更にメモリ割当てに使用された場合は、その過大データは正しく、実際に利用できるエリアよりも多くのエリア用にメモリが割当てられたと見なして、アラーム 6000 "Memory allocation with standard machine data" が出力されます。

## ■ SRAM および DRAM のメモリ割当て

通常の運転では SRAM と DRAM メモリは、スタートアッププロセスの一部としてしか割当てられていないため、次の例のような SRAM を使用したメモリ割当て手順が推奨されます。

- 標準マシンデータをロードします。
- MD 18230: MM\_USER\_MEM\_BUFFERED (SRAM 中のユーザメモリ) を高い値 (3000) にセットします。その後、NCK を起動します。アラーム 6030 "User memory limit has been adapted" が出力され、ユーザが利用できる最大メモリ容量が MD 18230: MM\_USER\_MEM\_BUFFERED に入力されます。他のメモリ設定マシンデータは全てデフォルト値にセットされます。
- 必要なチャンネル番号と軸番号をアクティブにします。詳細については以下を参照してください。

参照： /IAD/ Installation and Start-Up Guide SINUMERIK 840D

使用可能なスタティックメモリの空き量が、MD

18060: INFO\_FREE\_MEM\_STATIC (スタティックメモリ空き領域の表示) に表示されます。

- メモリの初期設定がメモリをスタティックに割当てていない場合、メモリエリアを設定し直す (マシンデータを介して個々または複数のエリアを増加／低減する) ことによって、メモリを工作機械の要件に適合させることができます。

確認事項： どのエリアが、メモリエリアを更に必要としていますか？

どのエリアが、問題のアプリケーションにとって重要度が低いですか？

- 選択したメモリエリア用の適切なマシンデータがセットされ、メモリ要件を定義すると、NCK はリセットされメモリを再編成します。

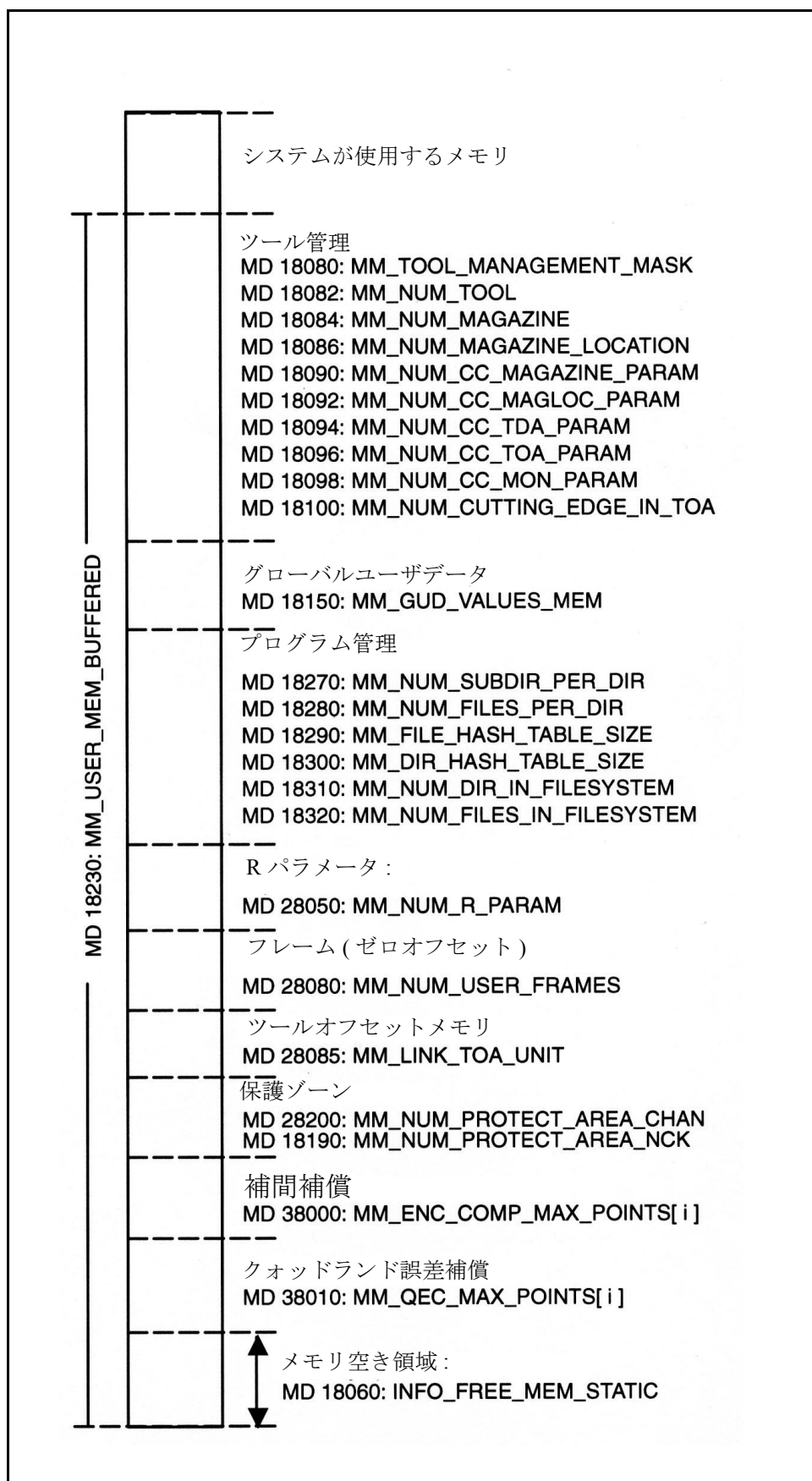


図 2.106 スタティック RAM (SRAM) の割当て



## ■ メモリ割当て SRAM

表 2.25 SRAM のメモリ空間割当て

マシンデータ	メモリ要件	注
ツール管理		
MD 18080: MM_TOOL_MANAGEMENT_MASK		詳細については、2.14.4「データの説明 (MD, SD)」の MD の説明を参照のこと。
MD 18082: MM_NUM_TOOL	ツールごとに 84 バイト	
MD 18084: MM_NUM_MAGAZINE	マガジンごとに 332 バイト	
MD 18086: MM_NUM_MAGAZINE_LOCATION	マガジン位置ごとに 64 バイト	
MD 18090: MM_NUM_CC_MAGAZINE_PARAM	入力 x マガジン数 x 4 バイト	MD 18084 : MM_NUM_MAGAZINE に 対応
MD 18092: MM_NUM_CC_MAGLOC_PARAM	入力 x マガジン数 x 4 バイト	MD 18084: MM_NUM_MAGAZINE に 対応
MD 18094: MM_NUM_CC_TDA_PARAM	入力 x ツール数 x 4 バイト	MD 18082 : MM_NUM_TOOL に対応
MD 18096: MM_NUM_CC_TOA_PARAM	入力 x TOs の数 x 8 バイト	MD 18100 : MM_NUM_CUTTING_EDGES_ IN_TOA に対応
MD 18098: MM_NUM_CC_MON_PARAM	入力 x TOs の数 x 4 バイト	MD 18100 : MM_NUM_CUTTING_EDGES_ IN_TOA に対応
MD 18100: MM_NUM_CUTTING_EDGES_IN_ TOA	アクティブモニタなし: ツール エッジごとに 250 バイト アクティブモニタあり: ツール エッジごとに 48 バイト 追加	
グローバルユーザデータ		
MD 18118: MM_NUM_GUD_MODULES		次の例を参照のこと
MD 18120: MM_NUM_GUD_NAMES_NCK	NCK 名ごとに 80 バイト	次の例を参照のこと
MD 18130: MM_NUM_GUD_NAMES_CHAN	チャンネル名ごとに 80 バイト	次の例を参照のこと
MD 18140: MM_NUM_GUD_NAMES_AXIS	軸名ごとに 80 バイト	
MD 18150:MM_GUD_VALUES_MEM		次の例を参照のこと

## GUD の例

次に、グローバルユーザデータのメモリ要件の計算方法の例を示します。

### 補足条件

- 2つのチャンネルを持つマシン
- 次の GUD モジュールが定義される：
  - UGUD ユーザ別
  - SGUD Siemens 別
  - MGUD マシン製造業者別
  - GUD7 (輪郭テーブルストック除去サイクル, CYCLE95 に必要, サイクルバージョン 3.4 以降)
- NCK 別およびチャンネル別変数が定義される。

### NCK 変数

2 REAL 値 -> 2 x 8 バイト	= 16 バイト
1 BOOL 値 -> 1 x 1 バイト	= 1 バイト
合計 1	= 17 バイト
3 = 合計 NCK ( 値の数 )	

### CHAN 変数

2 BOOL 値 -> 2 x 1 バイト	= 2 バイト
1 INT 値 -> 1 x 4 バイト	= 4 バイト
合計 2	= 6 バイト
3 = 合計 CHAN ( 値の数 )	
6 バイト ( 合計 2 ) x 2 ( チャンネルの数 ) = 12 バイト ( 合計 3 )	

### 必要なメモリ量の計算

- MD 18120: MM\_NUM\_GUD\_NAMES\_NCK = 3 ( 合計 NCK )  
NCK 名管理用のメモリ空間  
=> 3 x 80 バイト = 240 バイト
- MD 18130: MM\_NUM\_GUD\_NAMES\_CHAN = 3 ( 合計 CHAN )  
CHAN 名管理用のメモリ空間  
=> 3 x 80 バイト = 240 バイト
- 最大定義 GUD モジュール数 = 7 (GUD7)  
GUD モジュール管理用のメモリ空間  
=> 7 x 120 バイト = 840 バイト
- 変数のメモリ要件  
合計 1 + 合計 3 =  
17 バイト + 12 バイト = 29 バイト , KB 単位まで切上げるので次となる :  
MD 18150: MM\_GUD\_VALUES\_MEM = 1

- GUDに必要な総メモリ空間は次のように計算される：

NCK 名管理用のメモリ空間           = 240 バイト  
CHAN 名管理用のメモリ空間           = 240 バイト  
GUD モジュール管理用のメモリ空間= 840 バイト  
変数に必要なメモリ空間               = 1024 バイト  
合計                                       = 2344 バイト

マシンデータ	メモリ要件	注
<b>プログラム管理</b>		
MD 18270: MM_NUM_SUBDIR_PER_DIR	サブディレクトリごとに 40 バイト	
MD 18280: MM_NUM_FILES_PER_DIR	ファイルごとに 40 バイト	
MD 18290: MM_FILE_HASH_TABLE_SIZE		システムによって内部で割当てられるのでユーザが変更してはならない。
MD 18300: MM_DIR_HASH_TABLE_SIZE		システムによって内部で割当てられるのでユーザが変更してはならない。
MD 18310: MM_NUM_DIR_IN_FILESYSTEM		詳細については、2.14.4「データの説明 (MD, SD)」の MD の説明を参照のこと。
MD 18320: MM_NUM_FILES_IN_FILESYSTEM	ファイルごとに 320 バイト	
<b>R パラメータ</b>		
MD 28050: MM_NUM_R_PARAM	R パラメータごとに 8 バイト	
<b>フレーム (ゼロオフセット)</b>		
MD 28080: MM_NUM_USER_FRAMES	フレームごとに 232 バイト 管理用に 120 バイトの追加が必要である。	
<b>ツールオフセットメモリ</b>		
MD 28085: MM_LINK_TOA_UNIT	ユニットごとに 500 バイト	
<b>保護ゾーン</b>		
MD 28200: MM_NUM_PROTECT_AREA_CHAN	定義されたブロックごとに 400 バイト	
MD 18190: MM_NUM_PROTECT_AREA_NCK	定義されたエリアごとに 400 バイト	
<b>補償</b>		
MD 18340: MM_NUM_CEC_NAMES	1 KB が常に割当てられている	
MD 18342: MM_CEC_MAX_POINTS	各補償点ごとに 8 バイト 管理用に 2 バイトの追加が必要である	
MD 38000: MM_ENC_COMP_MAX_POINTS	各補償点ごとに 8 バイト 管理用に 2 バイトの追加が必要である	
MD 38010: MM_QEC_MAX_POINTS	各補償点ごとに 4 バイト 管理用に 2 バイトの追加が必要である	

---

## ■ メモリ割当て DRAM

- ダイナミックメモリ空き領域は、MD 18050:INFO\_FREE\_MEM\_DYNAMIC (利用可能なダイナミックメモリのデータ表示) に表示されます。
- メモリ初期設定のメモリ割当てが満足がいくものでない場合、そのときにはメモリエリアを設定し直し (マシンデータを介して個々または複数のエリアを増加／低減する)、メモリを工作機械の要件に適合させることができます。

確認事項：どのエリアが、メモリ空間を更に必要としていますか？

どのエリアが、問題のアプリケーションにとって重要度が低いですか？

- 選択したメモリエリア用の適切なマシンデータがセットされ、メモリ要件を定義すると、NCK はリセットされメモリを再編成します。

(注)

実際に利用可能な量よりも大きなダイナミックメモリが要求される場合、次の起動時に SRAM もまた自動的に削除され、メモリ構成用のマシンデータが全てリセットされデフォルト値に戻ります。

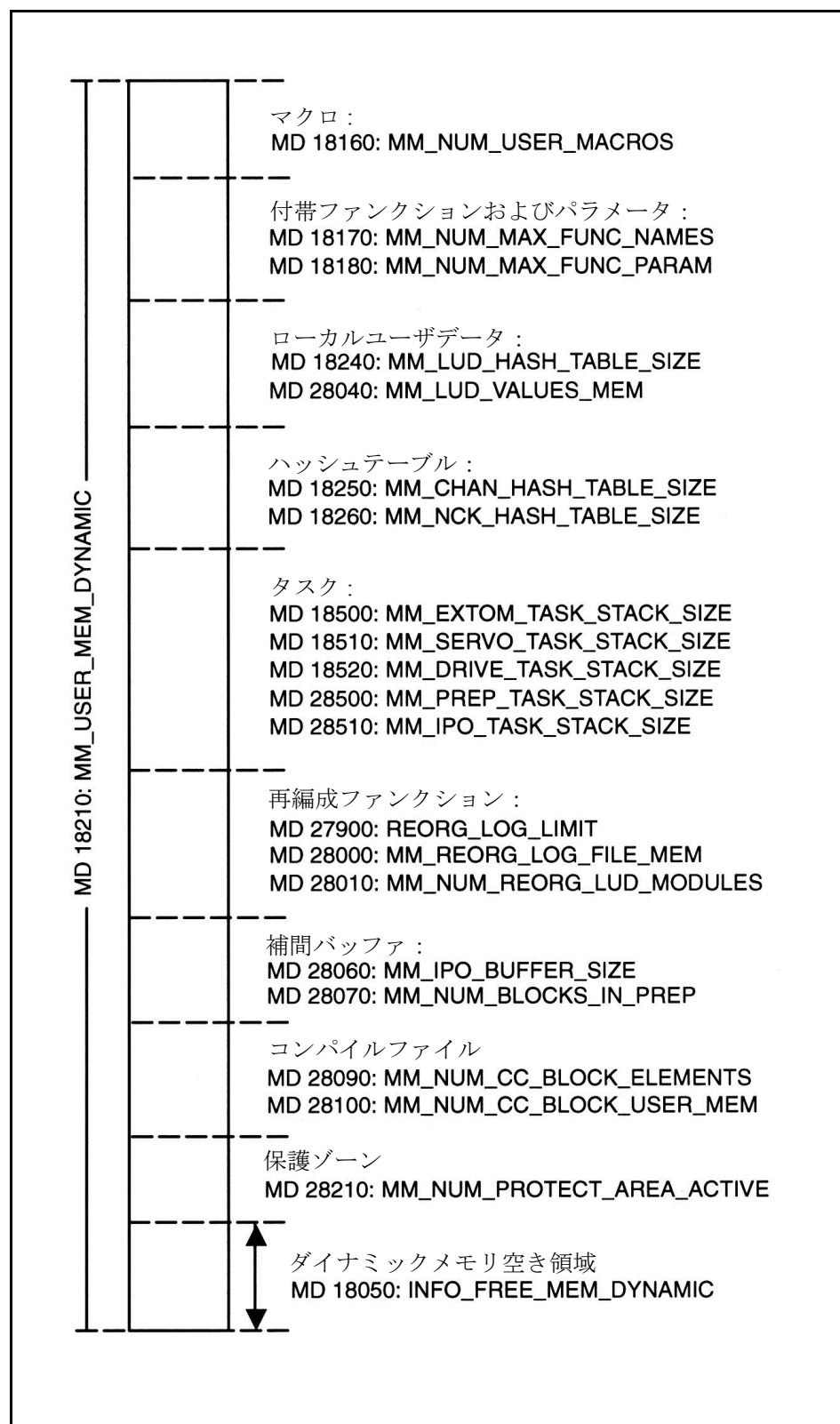


図 2.107 ダイナミック RAM のメモリ空間割当て (DRAM)

マシンデータ	メモリ要件	注
マクロ		
MD 18160: MM_NUM_USER_MACROS	マクロごとに 375 バイト	
付帯ファンクションおよびその追加パラメータ		
MD 18170: MM_NUM_MAX_FUNC_NAMES	付帯ファンクションごとに 150 バイト	
MD 18180: MM_NUM_MAX_FUNC_PARAM	パラメータごとに 72 バイト	入力した値は、必要である全ての付帯ファンクションパラメータの総計である。
ローカルユーザデータ		
MD 18240: MM_LUD_HASH_TABLE_SIZE		システムによって内部で割当てられるのでユーザが変更してはならない。
MD 18242: MM_MAX_SIZE_OF_LUD_VALUE	ブロックサイズは使用される変数により異なる データタイプ   メモリ要件 REAL           8 バイト INT            4 バイト BOOL           1 バイト CHAR           1 バイト STRING        キャラクタごとに 1 バイト、 ストリングごとに 100 キャラクタ AXIS           4 バイト FRAME          400 バイト	最もメモリ空間を必要とする変数に対してマシンデータを設定する必要がある。しかし、この変数より大きな値を設定してはならない。値が大きすぎると、アラームが出力される。  次の例を参照のこと
MD 28010: MM_NUM_REORG_LUD_MODULES		システムによって内部で割り当てられるのでユーザが変更してはならない。
MD 28020: MM_NUM_LUD_NAMES_TOTAL	LUD 名ごとに 150 バイト	次の例を参照のこと
MD 28040: MM_LUD_VALUES_MEM	LUD に必要な総メモリ空間	次の例を参照のこと

## ローカルユーザデータの例

パートプログラムに定義されたローカルユーザデータは、そのデータが定義されているプログラムが実行されている間は DRAM に保存されます。次に、ローカルユーザデータのメモリ要件の計算方法の例を示します。

### 例のパラメータ

1 REAL 値 -> 1 x 8 バイト = 8 バイト
2 BOOL 値 -> 2 x 1 バイト = 2 バイト
合計 1 = 10 バイト
3 = 合計 A ( 値の数 )

### 必要メモリの計算

次の変数を必ず使用すること：

- 変数に必要なメモリ  
合計 1 = 10 バイト
- MD 18242: MM\_MAX\_SIZE\_OF\_LUD\_VALUES = 8 バイト  
マシンデータは、最もメモリ容量を必要とする変数を基にして設定する必要があります。上の例では、この値は 8 バイトの REAL 値となります。
- MD 28020: MM\_NUM\_LUD\_NAMES\_TOTAL = 3 (total A)  
LUD 名管理用のメモリ空間  
=> 3 x 150 バイト = 450 バイト
- MD 28040: MM\_LUD\_VALUES\_MEM  
LUD に必要な総メモリ空間は次のように計算します：  
LUD NAMES 管理用のメモリ空間 = 450 バイト  
変数に必要なメモリ空間 = 10 バイト  
合計 = 460 バイト

計算した合計は、KB 単位となるように切上げて MD 28040:

MM\_LUD\_VALUES\_MEM に入力する必要があります（この例では 1 KB）。この設定で得たメモリは、それぞれ 8 バイトの大きさのブロックに割当てられます（MD 18242 を基にする）。例えば、プログラムで 1 REAL 値 (8 バイト) と 1 BOOL 値 (1 バイト) が使用される場合、2 ブロックのメモリにそれぞれ 8 バイトが割当てられます。

## パートプログラムに定義された LUD

パートプログラムに定義されたローカルユーザメモリは、そのメモリが定義されているプログラムの実行中は DRAM に保存されます。

その期間は、ソフトキー **PARAMETER** を使用して割当てられた値を表示させることができます。

## PP での変数の定義

DEF INT LUD\_VARIABLE1      VARIABLE1 の名前を持つ Integer 変数

DEF REAL LUD\_VARIABLE2 VARIABLE2 の名前を持つ REAL 変数

DEF REAL LUD_PAUL[19]	PAUL[0] ~ PAUL[19] の 20 の REAL 変数を持つ フィールド
-----------------------	---

## メモリ管理

システムが自動的にメモリブロックの割当てを管理します。

- メモリブロックの予約（LUD 定義を含むパートプログラムが処理される場合）
- 更なるブロックの予約（その番号の変数に提供されたメモリが不十分である場合）
- メモリ空間のリリース（LUD がすでに必要でない場合：プログラムの終了）

## 変数定義の変更

例えば 20 REAL 変数のように多数の変数が必要とされる場合、パートプログラムの初めに ARRAY (フィールド) を定義することによってメモリ空間を節約できます。変数を個別に定義しても有効ではありません。

例：

### 事例 1

DEF REAL LUD\_PAUL[19]

20 の LUD 変数 PAUL[0] ～ PAUL[19] を使用するこのフィールドには次のメモリ空間が必要です：

MD 28080 = 1                    =>    1 x 150 バイト =    150 バイト

20 の変数用のメモリ  $\Rightarrow 20 \times 8 \text{ バイト} = 160 \text{ バイト}$

20 の変数が必要とするメモリ総計 = 310 バイト

## 事例 2

20 の変数の個別の定義: PAUL0, PAUL1 ~ PAUL19

MD 28080 = 20                    => 20 x 150 バイト = 3000 バイト

20 の変数に必要な総メモリ  $\Rightarrow 20 \times 8$  バイト = 160 バイト

20 の変数が必要とするメモリ総計 = 3160 バイト

(注)

この変数定義の変更方法は、GUD 変数にも適用できます。

参照, LUD 変数用の MD 18242: MM MAX SIZE OF LUD VALUES



マシンデータ	メモリ要件	注
ハッシュテーブル		
MD 18250: MM_CHAN_HASH_TABLE_SIZE	入力 x チャンネル別名の数 x チャンネル別メモリ設定 68 バイト	システムによって内部で割当てられる のでユーザが変更してはならない。
MD 18260: MM_NCK_HASH_TABLE_SIZE	入力 x NCK 別名の数 x 68 バイト	システムによって内部で割当てられる のでユーザが変更してはならない。
タスク		
MD 18500: MM_EXTCOM_TASK_STACK_SIZE	入力 x 1 KB	
MD 18510: MM_SERVO_TASK_STACK_SIZE	入力 x 1 KB	
MD 18520: MM_DRIVE_TASK_STACK_SIZE	入力 x 1 KB	
MD 28500: MM_PREP_TASK_STACK_SIZE	入力 x 1 KB	
MD 28510: MM_IPO_TASK_STACK_SIZE	入力 x 1 KB	
Reorg ファンクション		
MD 27900: REORG_LOG_LIMIT	入力 x 1 KB	システムによって内部で割当てられる のでユーザが変更してはならない。
MD 28000: MM_REORG_LOG_FILE_MEM	入力 x 1 KB	システムによって内部で割当てられる のでユーザが変更してはならない。
MD 28010: MM_NUM_REORG_LUD_MODULES	入力 x 1 KB	システムによって内部で割当てられる のでユーザが変更してはならない。
補間バッファ		
MD 28060: MM_IPO_BUFFER_SIZE	IPO バッファの各 NC ブロック に 10 KB	
MD 28070: MM_NUM_BLOCKS_IN_PREP	準備用の各 NC ブロックに 10 KB	
コンパイルファイル		
MD 28090: MM_NUM_CC_BLOCK_ ELEMENTS	ブロックエレメントごとに 1.2 KB	
MD 28100: MM_NUM_CC_BLOCKS_USER_ MEM	入力 / 128 バイト = ブロック数	メモリは 128 バイトのブロックのグ リッド内でイネーブルとなっているの で、入力値は必ず 128 の倍数とする。
アクティブ保護ゾーン		
MD 28210: MM_NUM_PROTECTED_AREA_ ACTIVE		入力値は、MD 18190 および MD 28200 の設定値の合計によって定義されるも のとする。 MD 18190 = 2 MD 28200 = 2 => MD 28210 = 4

---

## ■ メモリ要件の計算

(注)

メモリ要件は、ソフトウェアのバージョンと NC 制御のタイプによって多少異なります。マシンデータに加えられた変更を基にしたメモリ要件変更の指定値（下表）は、ソフトウェアバージョン 4 および NCU 572 用の標準値として与えられます。標準値とマシンデータによって、他のソフトウェアや NC 制御用の標準値やマシンデータリミットは、次を参照してください。

参照： パラメーター一覧

### 概要

テーブルは次の順番で配置されています。

- DRAM
  - 一般マシンデータ
  - チャンネル別マシンデータ
  - 軸別マシンデータ
- SRAM
  - 一般マシンデータ
  - チャンネル別マシンデータ
  - 軸別マシンデータ

## テーブル入力

### 1. MD 番号

マシンデータの番号。関連の拡張子は /LIS/ に示されています。

### 2. 意味

マシンデータの意味

新規ライン：

GD: 基本 DRAM (オーバーヘッド) , GS: 基本 SRAM (オーバーヘッド)

(このオーバーヘッドは、MD が制御するファンクションが使用されると作成されます。値は MD にのみ指定され、カラムに指定されている値に正比例しません、また計算することもできません。)

### 3. 初期値 (def)

ソフトウェアが出荷されるときに設定されている値。

### 4. 初期値を 1 だけ増やしたときのメモリ要件の増加量 (バイト)

初期値が 1 だけ増やされたときにメモリ要件が何バイト増えるかを示します。

項 2 に指定された GD および GS 用の基本オーバーヘッドは、表示された変更に含まれています。

### 5. 初期値を (n) だけ増やす

容量計算でマシンデータ値を増やすために追加されたユニット数を指定します。増やしたメモリ割当ては項 6 に指定されます。

### 6. 初期値を n だけ増やしたことによるメモリ要件の増加量

マシンデータが項 5 に指定されている値によって増えた場合に必要となる追加メモリ量を指定します。

カラム 2 に指定された GD および GS 用の基本 (オーバーヘッド) は、表示された変更に含まれています。

### 7. 影響される SRAM

影響される DRAM

比例して他のタイプのメモリも使用される場合、カラムに x が 1 つ現れます。

(注)

マシンデータと要求メモリの実際の依存関係はもっと複雑です。MD が更にメモリを使用する別のファンクションを起動することもあります。使用するメモリ量と MD の数の関係は、必ずしも直線的なものではありません。したがって、下に示した表は、必要な構成を得るためにメモリを増減する際の一例を示したに過ぎません。これは、マシンデータに指定されている値を増やす際にも減らす際にも適用できます。

## ■ DRAM メモリ要件

表 2.26 一般マシンデータ, DRAM

MD 番号	意味	初期値	初期値を 1 だけ増やしたときのメモリ要件の増加量 (バイト)	初期値を (n) だけ増やす	初期値を n だけ増やしたことによるメモリ要件の増加量	影響される SRAM
10010	チャンネル	1	1134608		SRAM を参照	x
10134	MMC 通信パートナ数	6	28236			
18082	ツール数	30	120	10	1244	x
18088	ツールホルダ数 GD: 588, GS: 1293	0	588	n	SRAM を参照	x
18120	グローバルユーザ変数の数	10	84	10	828	x
18130	チャンネル別ユーザ変数の数	40	84	10	828	x
18140	軸別ユーザ変数の数	0	84	10	828	x
18160	マクロ数	10	680	10	6864	
18170	付帯ファンクション数 (サイクル)	40	120	10	1272	
18180	サイクル用の追加パラメータ数	300	60	10	612	
18190	マシン関連保護ゾーン用のファイル数 GD: 504, GS: 1062	0	504	n	SRAM を参照	x
18210	ダイナミックユーザメモリ	3370	1024	10	10240	
18280	ディレクトリ別のファイル数	100	76			
18342	ビームの垂れ補償用の補間点の最大数 GD: 380, GS: 1680	0	380	10	SRAM を参照	x

表 2.27 チャンネル別マシンデータ, DRAM

MD 番号	意味	初期値	初期値を 1 だけ増やしたときのメモリ要件の増加量 (バイト)	初期値を (n) だけ増やす	初期値を n だけ増やしたことによるメモリ要件の増加量	影響される SRAM
28000	REORG 用のメモリ容量	10	1084	10	10636	
28020	ローカルユーザ変数の数	200	160	10	1688	
28040	ローカルユーザ変数用のメモリ容量	8	1024	10	10260	
28060	IPO バッファの NC ブロック数	10	15452			
28070	ブロック準備用のブロック数	36	15576			
28080	設定可能なフレームの数	5	76	10	784	x
28085	チャンネルへの TO ユニットの割当て	1,2,3...	84		SRAM を参照	x
28090	コンパイルサイクル用のブロックエレメントの数	0	924			

MD 番号	意味	初期値	初期値を 1 だけ増 やしたときのメモ リ要件の増加量 (バイト)	初期値を (n) だけ増 やす	初期値を n だけ 増やしたことによ るメモリ要件 の増加量	影響される SRAM
28100	コンパイルサイクル用のブ ロックメモリの容量	256	1056			
28150	PLC 変数書込み用のエレ メント数	0	56			
28200	チャンネル別保護ゾーン用 のファイル数 GD: 504, GS: 1062	0	504		SRAM を参照	x
28210	同時にアクティブになる保 護ゾーン数	0	~18000	10	174852	
28250	同期化アクション計算式用 のエレメント数	159	104			
28252	FCTDEF 定義用のエレメン ト数	3	32			

表 2.28 軸別マシンデータ , DRAM

MD 番号	意味	初期値	初期値を 1 だけ増 やしたときのメモ リ要件の増加量 (バイト)	初期値を (n) だけ増 やす	初期値を n だけ 増やしたことによ るメモリ要件 の増加量	影響される SRAM
38000	補間補償用の補間点数 GD: 212, GS: 976	0		10	212	x
38010	クォッドラント誤差補償用 の値の数 GD: 548, GS: 1932	0	10	548	x	

## ■ SRAM メモリ要件

表 2.29 一般マシンデータ, SRAM

MD 番号	意味	初期値	初期値を 1 だけ増 やしたときのメモ リ要件の増加量 (バイト)	初期値を (n) だけ増 やす	初期値を n だけ 増やしたことによ るメモリ要件 の増加量	影響される SRAM
10010	チャンネル	1	10032			x
18082	ツール数	30	80	10	812	x
18084	マガジン数	3	244	10	2416	
18086	マガジン位置の数 GD: 0, GS: 6	30	244	31	7612	
18088	ツールホルダ数 GD: 588, GS: 1293	0	1408	10	1152	x
18090	コンパイルサイクル用のマガジ ンデータの量 GD: 0, GS: 36	0	40	10	32	
18092	コンパイルサイクル用のマガジ ンデータの量 GD: 0, GS: 36	0	40	10	32	
18094	ツールごとのツール別データの 量 GD: 0, GS: 31	0	40	9	68	
18096	ツールエッジ別のデータの量 GD: 0, GS: 31	0	40	9	68	
18098	ツールエッジ別の監視データの 量 GD: 0, GS: 36	0	40	9	32	
18100	TOA モジュールごとのツール オフセット GD: 0, GS: 13	30	244	10	2408	
18102	D 番号プログラミングのタイプ	0	-2344 (1: 直接 D 番号プログラミン グの場合の要件)			
18118	アクティブファイルシステム中 の GUD ファイル数	7	628			
18120	グローバルユーザ変数の数	10	120	10	1200	x
18130	チャンネル別ユーザ変数の数	40	120	10	1200	x
18140	軸別ユーザ変数の数	0	120	10	1200	x
18150	ユーザ変数用のメモリ容量	12	1056	10	10548	
18190	マシン関連保護ゾーン用のファ イル数 GD: 504, GS: 1062	0	1464	5	2012	x
18230	SRAM のユーザメモリ	1900	1024	10	10232	
18310	パッシブファイルシステム内の ディレクトリ数	30	1236			
18320	パッシブファイルシステム内の ファイル数	100	344			
18342	ビームの垂れ補償用の補間点数 GD: 380, GS: 1680	0		10	1748	x

表 2.30 一般マシンデータ, SRAM

MD 番号	意味	初期値	初期値を 1 だけ増 やしたときのメモ リ要件の増加量 (バイト)	初期値を (n) だけ増 やす	初期値を n だけ 増やしたことによ るメモリ要件 の増加量	影響される SRAM
18400	曲線テーブル数 GD: 0, GS: 4	0	104	1	100	
18402	曲線セグメント数 GD: 0, GS: 4	0	128	1	124	
18404	曲線テーブル多項式の数 GD: 0, GS: 4	0	60	1	56	

表 2.31 チャンネル別マシンデータ, SRAM

MD 番号	意味	初期値	初期値を 1 だけ増 やしたときのメモ リ要件の増加量 (バイト)	初期値を (n) だけ増 やす	初期値を n だけ 増やしたことによ るメモリ要件 の増加量	影響される SRAM
28050	チャンネル別 R パラメー タ数	100	8	10	64	
28080	設定可能なフレーム数	5	428	10	4220	x
28085	チャンネルへの TO ユニッ トの割当て	1,2,3 ...	2124			x
28200	チャンネル別保護ゾーン用 のファイル数 GD: 504, GS: 1062	0	1468	5	2008	x

表 2.32 軸別マシンデータ, SRAM

MD 番号	意味	初期値	初期値を 1 だけ増 やしたときのメモ リ要件の増加量 (バイト)	初期値を (n) だけ増 やす	初期値を n だけ 増やしたことによ るメモリ要件 の増加量	影響される SRAM
38000	補間補償用の補間点数 GD: 212, GS: 976	0		10	1040	x
38010	クォッドラント誤差補償用 の値の数 GD: 548, GS: 1932	0		10	1996	x

### 2.14.3 補足条件

なし

### 2.14.4 データの説明 (MD, SD)

#### ■ 一般マシンデータ

18050 MD 番号	INFO_FREE_MEM_DYNAMIC ダイナミックメモリ空き領域のデータ	
デフォルト値:-	最小入力リミット:-	最大入力リミット:-
変更が有効になるための条件: 電源オン	保護レベル:0	単位:バイト
データタイプ:DWORD	適用開始 SW バージョン:1.1 以降	
意味:	<p>ダイナミックメモリで利用できるバイト数を表示するためのデータです。このデータを定義することはできません。NCK の起動の度に表示は更新されます。必要なメモリを手動で求める方法は次のとおりです。</p> <ul style="list-style-type: none"><li>- 入力値を 1 だけ増やす</li><li>- NCK を起動する</li><li>- メモリ要件を讀出す</li><li>- 拡張の準備をする</li></ul> <p>マシンデータの内容により、揮発性ユーザデータエリアの拡張に MD を介して現在使できるダイナミック RAM の空き量が特定されます。ローカルユーザデータ (LUD) などのパラメータを拡張する前には、必ず十分なメモリが利用できるか確認する必要があります。</p>	
例外:	ダイナミックメモリの要求が、現在利用可能な量よりも大きい場合、次の起動時に SRAM は削除され、マシンデータは全て初期設定に初期化されます。	



18060 MD 番号	INFO_FREE_MEM_STATIC スタティックメモリの空き量の表示データ	
デフォルト値:-	最小入力リミット:-	最大入力リミット:-
変更が有効になるための条件: 電源オン	保護レベル:0	単位: バイト
データタイプ: DWORD	適用開始 SW バージョン: 1.1	
意味:	<p>起動時に、マシンデータの内容が、パッシブファイルシステムで利用可能な非揮発性メモリの空き量を表示します。この値の更新はありません。その時の最新の値を求めるときは、次のオペレーションが利用できます。</p> <p>MMC102: プログラミングのサービス: NCK 空き量</p> <p>MM100: プログラミング: ソフトキーメモリ情報</p> <p>必要となるバックアップメモリの量に影響を与える MD が変更された場合、パッシブファイルシステムで利用可能なメモリ量も変化します。というのも、パッシブファイルシステムに割当てられたメモリ量は、MD 18230: MM_USER_MEM_BUFFERED (SRAM ユーザメモリ) のメモリ設定から他の全てのユーザバックアップデータ (MD 18350: MM_USER_FILE_MEM_MINIMUM (最小パートプログラムメモリ) の説明も参照) を引いたものだからです。</p> <p>このデータは定義できません。表示は NCK が起動するごとに更新されます。</p>	
例外:	ダイナミックメモリの要求が、現在利用可能な量よりも大きい場合、次の起動時に SRAM は削除され、マシンデータは全て初期設定に初期化されます。	

18070 MD 番号	INFO_FREE_MEM_DPR DUAL-PORT RAM の空き容量の表示データ	
デフォルト値:-	最小入力リミット:-	最大入力リミット:-
変更が有効になるための条件: 電源オン	保護レベル:0	単位: バイト
データタイプ: DWORD	適用開始 SW バージョン:	
意味:	なし	
この MD は次の場合は無意味:	この機能は、ソフトウェアバージョン 2 では利用できません。	

18080 MD 番号	MM_TOOL_MANAGEMENT_MASK TM ファンクション用のメモリ予約画面	
デフォルト値: 00H	最小入力リミット: 00H	最大入力リミット: FFFFH
変更が有効になるための条件: 電源オン	保護レベル: 1	単位: -
データタイプ: DWORD	適用開始 SW バージョン: 2 以降	
意味:	<p>TM 別メモリ予約は、この MD がビットコードを基準にして段階を追って実行します。個々のメモリエリアの予約は、次の MD によって定義されます。</p> <ul style="list-style-type: none"> <li>MD 18086: MM_NUM_MAGAZINE_LOCATION</li> <li>MD 18084: MM_NUM_MAGAZINE</li> <li>MD 18096: MM_NUM_CC_TOA_PARAM</li> <li>MD 18094: MM_NUM_CC_TDA_PARAM</li> <li>MD 18098: MM_NUM_CC_MON_PARAM</li> <li>MD 18092: MM_NUM_CC_MAGLOC_PARAM</li> <li>MD 18090: MM_NUM_CC_MAGAZINE_PARAM</li> </ul> <p>個々のメモリ構成マシンデータを設定するだけでは、メモリは予約できません。また、次の起動中に該当するマシンデータがアクティブになって初めて、マシンデータは変更されます。</p> <p>ビット 1: ツール管理データを利用可能にする:</p> <ul style="list-style-type: none"> <li>ツールの基本機能用のメモリ予約 MD をセットする必要がある: MD 18092: MM_NUM_TOOL MD 18100: MM_NUM_CUTTING_EDGES_IN_TOA</li> <li>ツール管理機能用のメモリ予約 MD をセットする必要がある: MD 18086: MM_NUM_MAGAZINE_LOCATION MD 18084: MM_NUM_MAGAZINE</li> </ul> <p>ビット 1 がセットされると、TM 別メモリが MM_NUM_TOOL に定義されたメモリスペースに追加されます。</p> <p>ビット 2: ツール監視データを使用可能にする:</p> <ul style="list-style-type: none"> <li>ツールの基本機能用のメモリ予約 MD をセットする必要がある: MD 18092: MM_NUM_TOOL MD 18100: MM_NUM_CUTTING_EDGES_IN_TOA</li> </ul> <p>ビット 2 がセットされた場合、監視データ用のメモリが利用できるようになります。TM 別メモリが、MD 18100: MM_NUM_CUTTING_EDGES_IN_TOA に設定されたメモリスペースに追加されます。</p> <p>ビット 3: OEM/CC データが利用可能:</p> <ul style="list-style-type: none"> <li>メモリ予約 MD をセットする必要がある: MD: MM_NUM_CC_...</li> </ul> <p>ビット 3 がセットされた場合、OEM アプリケーションメモリが利用可能となります。</p> <p>ビット 4: "Allow for adjacent location" ツール管理:</p> <ul style="list-style-type: none"> <li>メモリは、TM ファンクション "Consider adjacent location" に利用できる。</li> </ul>	
例外, エラー	このマシンデータを変更すると、バックアップデータが喪失します。	
関連	<p>MD 18084: MM_NUM_MAGAZINE (NCK が管理できるマガジン数)</p> <p>MD 18086: MM_NUM_MAGAZINE_LOCATION (NCK が管理できるマガジン位置数)</p> <p>MD 18090: MM_NUM_CC_MAGAZINE_PARAM (CC がセットアップして評価するマシンデータ数)</p> <p>MD 18092: MM_NUM_CC_MAGLOC_PARAM (CC がセットアップして評価するマガジン位置データ数)</p> <p>MD 18094: MM_NUM_CC_TDA_PARAM (OEM とコンパイルサイクル用ツールごとのツール別データ数)</p> <p>MD 18096: MM_NUM_CC_TOA_PARAM (OEM とコンパイルサイクル用のツール切削エッジごとのデータ数)</p> <p>MD 18098: MM_NUM_CC_MON_PARAM (OEM とコンパイルサイクル用ツール切削エッジごとの監視データ数)</p>	

18080 MD 番号	MM_TOOL_MANAGEMENT_MASK TM ファンクション用のメモリ予約画面
参照	/FBW/, Description of Functions, Tool Management (ファンクション説明, ツール管理)

18082 MD 番号	MM_NUM_TOOL NCK が管理するツール数	
デフォルト値 : 30	最小入力リミット : 0	最大入力リミット : 600
変更が有効になるための条件 : 電源オン	保護レベル : 2	単位 : -
データタイプ : DWORD	適用開始 SW バージョン : 2 以降	
意味 :	NC は、ツール数が MD に入力された最大値を超えると管理ができません。1 つのツールには最低 1 つの切削エッジがあります。	
例外, エラー	最大ツール数は切削エッジ数に対応しています。 ツール管理ファンクションが使用されていない場合、この MD をセットする必要があります。 このマシンデータが変更されると、バックアップデータが喪失します。	
関連	MD 18100: MM_NUM_CUTTING_EDGES_IN_TOA ( NCK ファンクション用の予約メモリの画面フォーム )	

18084 MD 番号	MM_NUM_MAGAZINE NC が管理するマガジン数	
デフォルト値 : 3	最小入力リミット : 0	最大入力リミット : 32
変更が有効になるための条件 : 電源オン	保護レベル : 2	単位 : -
データタイプ : DWORD	適用開始 SW バージョン : 2 以降	
意味 :	NC が管理できるマガジン数 TM MD 20310: TOOL_MANAGEMENT_MASK, MD 18080: MM_TOOL_MANAGEMENT_MASK 用の MD およびオプションの TM\$ON_TECHNO_FUNCTION_MASK がセットされている必要があります。	
この MD は次の場合は無意味	ツール管理機能を使用していない場合, MD は無関係となります。	
例外, エラー	ツール管理ステップ 2 のみ : Value = 0 -> ツール管理ステップ 2 は, このデータにメモリが設定されていないため, 起動できません。このマシンデータが変更されると, バックアップデータが喪失します。	
関連	MD 18080: MM_TOOL_MANAGEMENT_MASK ( TM ファンクション用のメモリ予約の画面フォーム ) MD 20310: TOOL_MANAGEMENT_MASK ( さまざまなツール管理ファンクションの起動 ) \$ON_TECHNO_FUNCTION_MASK	
参照	/FBW/, Description of Functions, Tool Management	

18086 MD 番号	MM_NUM_MAGAZINE_LOCATION マガジン位置数	
デフォルト値: 30	最小入力リミット: 0	最大入力リミット: 600
変更が有効になるための条件: 電源オン	保護レベル: 2	単位: -
データタイプ: DWORD	適用開始 SW バージョン: 2 以降	
意味:	NC が管理できるマガジン数 TM MD 20310: TOOL_MANAGEMENT_MASK, MD 18080: MM_TOOL_MANAGEMENT_MASK 用の MD およびオプションの TM\$ON_TECHNO_FUNCTION_MASK がセットされている必要があります。	
この MD は次の場合は無意味	ツール管理機能を使用していない場合、MD は無関係です。	
例外, エラー	ツール管理ステップ 2 のみ: Value = 0 -> ツール管理ステップ 2 は、このデータにメモリが設定されていないため、起動できません。このマシンデータが変更されると、バックアップデータが喪失します。	
関連	MD 18080: MM_TOOL_MANAGEMENT_MASK ( TM ファンクション用のメモリ予約の画面フォーム ) MD 20310: TOOL_MANAGEMENT_MASK ( 様々なツール管理ファンクションの起動 ) \$ON_TECHNO_FUNCTION_MASK	
参照	/FBW/, Description of Functions, Tool Management	

18090 MD 番号	MM_NUM_CC_MAGAZINE_PARAM TM コンパイルサイクル: マガジンデータ数	
デフォルト値: 0	最小入力リミット: 0	最大入力リミット: 10
変更が有効になるための条件: 電源オン	保護レベル: 2	単位: -
データタイプ: DWORD	適用開始 SW バージョン: 2 以降	
意味:	ツール管理用 MD とツール管理オプションがセットされている場合に限り: メモリエリアがセットアップされ、またコンパイルサイクルによって評価可能な、マガジンデータ (フォーマット IN_int) 数	
この MD は次の場合は無意味	ツール管理機能を使用していない場合、MD は無関係です。	
関連	MD 18080: MM_TOOL_MANAGEMENT_MASK ( TM ファンクション用のメモリ予約の画面フォーム ) MD 18084: MM_NUM_MAGAZINE ( NC が管理するマガジン数 )	
参照	/FBW/, Description of Functions, Tool Management	

18092 MD 番号	MM_NUM_CC_MAGLOC_PARAM TN コンパイルサイクル：マガジン位置データ	
デフォルト値：0	最小入力リミット：0	最大入力リミット：10
変更が有効になるための条件： 電源オン	保護レベル：2	単位：-
データタイプ：DWORD	適用開始 SW バージョン：2 以降	
意味：	ツール管理用 MD とツール管理オプションがセットされている場合に限り： メモリエリアがセットアップされ、またコンパイルサイクルによって評価可能な、マガジンデータ（フォーマット IN_int）数	
この MD は次の場合は無意味	ツール管理機能を使用していない場合、MD は無関係です。	
例外、エラー	このマシンデータが変更されると、バックアップデータが喪失します。	
関連	MD 18080: MM_TOOL_MANAGEMENT_MASK ( TM ファンクション用のメモリ予約の画面フォーム )	
参照	/FBW/, Description of Functions, Tool Management	

18094 MD 番号	MM_NUM_CC_TDA_PARAM TM コンパイルサイクル：TDA データ数	
デフォルト値：0	最小入力リミット：0	最大入力リミット：10
変更が有効になるための条件： 電源オン	保護レベル：2	単位：-
データタイプ：DWORD	適用開始 SW バージョン：2 以降	
意味：	ツール管理用 MD とツール管理オプションがセットされている場合： メモリエリアがセットアップされコンパイルサイクルによって評価可能な TDA（ツール別）データ（フォーマット IN_int）の数	
この MD は次の場合は無意味	ツール管理機能を使用していない場合、MD は無関係です。	
例外、エラー	このマシンデータが変更されると、バックアップデータが喪失します。	
関連	MD 18080: MM_TOOL_MANAGEMENT_MASK ( TM ファンクション用のメモリ予約の画面フォーム ) MD 18082: MM_NUM_TOOL ( NCK が管理するツール数 )	
参照	/FBW/, Description of Functions, Tool Management	

18096 MD 番号	MM_NUM_CC_TOA_PARAM TM コンパイルサイクル：TOA データ数	
デフォルト値：0	最小入力リミット：0	最大入力リミット：10
変更が有効になるための条件： 電源オン	保護レベル：2	単位：-
データタイプ：DWORD	適用開始 SW バージョン：2 以降	
意味：	ツール管理用 MD とツール管理オプションがセットされている場合： メモリエリアがセットアップされコンパイルサイクルによって評価可能な、 切削エッジ別の TOA（ツール別）データ（フォーマット IN_int）の数	
この MD は次の場合は無意味	ツール管理ステップの 1 と 2 が起動していません。	
例外，エラー	このマシンデータが変更されると，バックアップデータが喪失します。	
関連	MD 18080: MM_TOOL_MANAGEMENT_MASK ( TM ファンクション用のメモ リ予約の画面フォーム ) MD 18100: MM_NUM_CUTTING_EDGES_IN_TOA ( NCK のツールオフセット 数 )	
参照	/FBW/, Description of Functions, Tool Management	

18098 MD 番号	MM_NUM_CC_MON_PARAM TM コンパイルサイクル：監視データ数	
デフォルト値：0	最小入力リミット：0	最大入力リミット：10
変更が有効になるための条件： 電源オン	保護レベル：2	単位：-
データタイプ：DWORD	適用開始 SW バージョン：2 以降	
意味：	ツール管理コンパイルサイクルの場合： コンパイルサイクルがそれぞれのツールに対して生成し，かつ評価すること ができる監視データの数	
この MD は次の場合は無意味	ツール管理機能を起動していない場合は MD は無関係です。	
例外，エラー	このマシンデータが変更されると，バックアップデータが喪失します。	
関連	MD 18080: MM_TOOL_MANAGEMENT_MASK ( TM ファンクション用のメモ リ予約の画面フォーム ) MD 18100: MM_NUM_CUTTING_EDGES_IN_TOA ( NCK のツールオフセット 数 )	
参照	/FBW/, Description of Functions, Tool Management	

18100 MD 番号	MM_NUM_CUTTING_EDGES_IN_TOA NCK のツールオフセット数	
デフォルト値：30	最小入力リミット：0	最大入力リミット：600
変更が有効になるための条件： 電源オン	保護レベル：2	単位：-
データタイプ：DWORD	適用開始 SW バージョン：1.1 以降	
意味：	NCK にツール切削エッジ数を定義します。このマシンデータが，各ツール エッジ用の TOA モジュールごとに，ツールタイプには関係なく，約 250 バイ トのバックアップメモリを予約します	
例外，エラー	このマシンデータが変更されると，バックアップデータが喪失します。	
参照	/FBW/, Description of Functions, Tool Management	

18118 MD 番号	MM_NUM_GUD_MODULES GUD モジュール数	
デフォルト値: 3	最小入力リミット: 1	最大入力リミット: 9
変更が有効になるための条件: 電源オン	保護レベル: 2	単位: -
データタイプ: DWORD	適用開始 SW バージョン: 2 以降	
意味:	<p>GUD ブロックは、ユーザ定義データを保存することができるファイルに対応します。9 GUD ブロックが利用できますが、そのうちの 3 ブロックはすでに特定のユーザ/アプリケーションに割当てられています。</p> <p>UGUD_DEF_USER (ユーザ用のブロック)</p> <p>SGUD_DEF_USER (安川用のブロック)</p> <p>MGUD_DEF_USER (マシン製造業者用のブロック)</p>	
例外, エラー	<p>GUD が、最大数を使用して GUD モジュール数を確定する。</p> <p>例: GUD モジュールが次のように定義された場合、</p> <p>UGUD</p> <p>MGUD</p> <p>GUD5</p> <p>GUD8</p> <p>マシンデータは値 8 に設定されなければならない、よってメモリ容量は 8 x 120 バイト = 960 バイトとなる。</p> <p>したがって、考えられる「最低」の GUD モジュールを選択することを推奨します。GUD モジュールの UGUD や MGUD がどこにも割当てられていない場合、これらを本目的に使用できます。</p>	
関連:	MD 18150: MM_GUD_VALUES_MEM (ユーザ変数用のメモリ)	

18120 MD 番号	MM_NUM_GUD_NAMES_NCK グローバルユーザ変数の数	
デフォルト値: 10	最小入力リミット: 0	最大入力リミット: 400
変更が有効になるための条件: 電源オン	保護レベル: 2	単位: -
データタイプ: DWORD	適用開始 SW バージョン: 1.1 以降	
意味:	<p>NCK グローバルユーザデータ (GUD) 用のユーザ変数の数を定義します。変数ごとに約 8 バイトのメモリが、変数の名前用に SRAM に予約されます。変数の値に必要な追加メモリは、変数のデータタイプによって異なります。利用可能な NCK グローバルユーザ変数の数は、MM_NUM_GUD_NAMES_NCK または MD 18150: MM_GUD_VALUES_MEM (ユーザ変数用メモリ) にセットされているリミット値によって制限されています。</p>	
例外, エラー	このマシンデータが変更されると、バックアップデータが喪失します。	
関連:	MD 18150: MM_GUD_VALUES_MEM (ユーザ変数用メモリ)	

18130 MD 番号	MM_NUM_GUD_NAMES_CHAN チャンネル別ユーザ変数の数	
デフォルト値: 10	最小入力リミット: 0	最大入力リミット: 200
変更が有効になるための条件: 電源オン	保護レベル: 2	単位: -
データタイプ: DWORD	適用開始 SW バージョン: 1.1 以降	
意味:	NCK グローバルユーザデータ (GUD) 用のユーザ変数の数を定義します。変数ごとに約 8 バイトのメモリが、変数の名前用に SRAM に予約されます。変数に必要となる追加メモリは、変数のデータサイズにチャンネル数を掛けた値に等しくなります。すなわち、各チャンネルにはそれぞれ変数値に利用可能なメモリが割当てられています。利用可能なチャンネル別のグローバルユーザ変数は、MD 18130: MM_NUM_GUD_NAMES_CHAN または MD 18150: MM_GUD_VALUES_MEM (ユーザ変数用のメモリ) に定義されているリミットに到達すると、全変数が消費されたことになります。	
例外, エラー	このマシンデータが変更されると、バックアップデータが喪失します。	
関連:	MD 18150: MM_GUD_VALUES_MEM (ユーザ変数用メモリ)	

18140 MD 番号	MM_NUM_GUD_NAMES_AXIS 軸別ユーザ変数の数	
デフォルト値: 0	最小入力リミット: 0	最大入力リミット: 100
変更が有効になるための条件: 電源オン	保護レベル: 0	単位: -
データタイプ: DWORD	適用開始 SW バージョン:	
例外, エラー	このマシンデータが変更されると、バックアップデータが喪失します。	
この MD は次の場合は無意味	この機能は、ソフトウェアバージョン 2 では利用できません。	



18150 MD 番号	MM_GUD_VALUES_MEM ユーザ変数のメモリ空間	
デフォルト値: 2	最小入力リミット: 0	最大入力リミット: 50
変更が有効になるための条件: 電源オン	保護レベル: 2	単位: KB
データタイプ: DWORD	適用開始 SW バージョン: 1.1	
意味:	指定値が、グローバルユーザデータ (GUD) の変数用のメモリを予約します。 どのくらいメモリが必要かは、変数に使用されるデータタイプによります。 必要なデータタイプ別のメモリ量: データタイプ    必要なメモリ量 REAL            8 バイト INT             4 バイト BOOL            1 バイト CHAR            1 バイト STRING          キャラクタごとに 1 バイト、ストリングごとに 100 キャラクタ AXIS            4 バイト FRAME           400 バイト  チャンネル別または軸別グローバルユーザデータに必要な総メモリは、変数にチャンネル数または軸数を掛けた値に等しくなります。利用可能なグローバルユーザ変数は、MD: MM_NUM_GUD_NAMES_xxxx または MM_GUD_VALUES_MEM に定義されているリミットに達すると、変数が全て消費されたこととなります。 非揮発性メモリが使用されます。	
例外, エラー	このマシンデータが変更されると、バックアップデータが喪失します。	
関連:	MD 18118: MM_NUM_GUD_MODULES: ( GUD モジュールの変数 ) MD 18120: MM_NUM_GUD_NAMES_NCK ( グローバルユーザ変数の数 ) MD 18130: MM_NUM_GUD_NAMES_CHAN ( チャンネル別ユーザ変数の数 )	

18160 MD 番号	MM_NUM_USER_MACROS マクロ数	
デフォルト値: 10	最小入力リミット: 0	最大入力リミット: 100
変更が有効になるための条件: 電源オン	保護レベル: 2	単位: -
データタイプ:	DWORD 適用開始 SW バージョン: 1.1 以降	
意味:	<p>ファイル _N_SMAC_DEF, _N_NMAC_DEF および _N_UMAC_DEF に保存することができるマクロ総数を定義します。これらのファイルは開いた際にはそれぞれファイルコード用に最低 1 KB のメモリを必要とします。ファイルコード用のこのリミットを超えた場合は、別の 1 KB のメモリがファイル用に予約されます。ダイナミックユーザメモリが使用されます。管理タスク用の指定マクロ数に対して 1 マクロごとに約 375 バイトが予約されます。</p>	
例外, エラー	このマシンデータが変更されると、バックアップデータが喪失します。	

18170 MD 番号	MM_NUM_MAX_FUNC_NAMES 付帯ファンクションの数	
デフォルト値: 30	最小入力リミット: 0	最大入力リミット: 50
変更が有効になるための条件: 電源オン	保護レベル: 2	単位: -
データタイプ: DWORD	適用開始 SW バージョン: 1.1 以降	
意味:	<p>マシンデータは、次の 2 つで利用できる事前定義ファンクション（例えば、サイン、コサインなど）以外の付帯ファンクションの最大数を定義します。</p> <ul style="list-style-type: none"> <li>- サイクルプログラム</li> <li>- コンパイルサイクルソフトウェア</li> </ul> <p>ファンクション名をグローバル NCK ディレクトリに入力しますが、すでに存在する名前は使用しないでください。</p> <p>提供のサイクルパッケージのソフトウェアバージョン 1 には、MD の初期設定を行った付帯ファンクションが入っています。</p> <p>このデータは揮発性メモリに保存されます。付帯ファンクションの管理にはそれぞれ約 150 バイトが必要です。</p>	
関連	MD 18180: MM_NUM_MAX_FUNC_PARAM (付帯ファンクションパラメータの数)	

18180 MD 番号	MM_NUM_MAX_FUNC_PARAM 追加パラメータ数	
デフォルト値: 300	最小入力リミット: 0	最大入力リミット: 500
変更が有効になるための条件: 電源オン	保護レベル: 2	単位: -
データタイプ: DWORD	適用開始 SW バージョン: 1.1 以降	
意味:	<p>次の 2 つで利用できる付帯ファンクションに必要なパラメータの最大数を定義します。</p> <ul style="list-style-type: none"> <li>- サイクルプログラム</li> <li>- コンパイルサイクルソフトウェア</li> </ul> <p>提供のサイクルパッケージのソフトウェアバージョン 1 の付帯ファンクションには 50 パラメータが必要です。</p> <p>このデータは揮発性メモリに保存されます。各パラメータにはそれぞれ約 72 バイトが必要です。</p>	
関連	MD 18170: MM_NUM_MAX_FUNC_NAMES (付帯ファンクション数)	

18190 MD 番号	MM_NUM_PROTECT_AREA_NCK NCK の保護ゾーン数	
デフォルト値： NCU 570: 0 NCU 571: 0 NCU 572: 0 NCU 573: 0	最小入力リミット： NCU 570: 0 NCU 571: 0 NCU 572: 0 NCU 573: 0	最大入力リミット： NCU 570: 4 NCU 571: 4 NCU 572: 10 NCU 573: 10
変更が有効になるための条件： 電源オン	保護レベル：2	単位：-
データタイプ：DWORD	適用開始 SW バージョン：1.1 以降	
意味：	このマシンデータは、NCK で利用可能な保護ゾーンの作成に必要なブロック数を定義します。	
例外，エラー	このマシンデータが変更されると、バックアップデータが喪失します。	
参照	1.2 送り軸監視機能及びプロテクションゾーン (A3)	

18210 MD 番号	MM_USER_MEM_DYNAMIC DRAM のユーザメモリ	
デフォルト値： NCU 570: 1750 NCU 571: 1750 NCU 572: 3370 NCU 573: 3500	最小入力リミット： NCU 570: - NCU 571: - NCU 572: - NCU 573: -	最大入力リミット： NCU 570: - NCU 571: - NCU 572: - NCU 573: -
変更が有効になるための条件： 電源オン	保護レベル：2/7	単位：KB
データタイプ：DWORD	適用開始 SW バージョン：1.1 以降	
意味：	<p>NC 中に物理的に存在する DRAM は、システムとユーザの両方によって共同で使用されます。MM_USER_MEM_DYNAMIC は、ユーザが使用できる DRAM のメモリ量を定義します。入力リミットは、CNC のハードウェアとソフトウェアの構成によって異なります。このメモリエリアには、次の 2 つを初めとするさまざまなタイプのユーザデータが含まれます。</p> <ul style="list-style-type: none"> <li>- ローカルユーザデータ</li> <li>- REORG-LOG データ</li> </ul> <p>ダイナミックユーザメモリのデータはバックアップされません。</p> <p>入力リミットによって制限されるため、予約されたメモリ空間がハードウェアで実際に利用できるメモリ量を超えることはありません。</p>	
アプリケーション	<p>デフォルト値がセットされる際には、NCU 572/573 を使用する場合、定義されたチャンネル数によって次の DRAM メモリがユーザの使用可能領域となります。</p> <ul style="list-style-type: none"> <li>- 約 1 MB (1 チャンネルの定義)</li> <li>- 約 300 KB (2 チャンネルの定義)</li> </ul>	
例外，エラー	<p>起動中に、システムソフトウェアが、DRAM の総必要量と MD: MM_USER_MEM_DYNAMIC にセットされた値を比較します。要求量がマシンデータに定義された容量を越える場合、アラーム 6000 "Memory allocation with standard machine data" が出力されます。起動中に、制御系が MM_USER_MEM_DYNAMIC を通じて要求されたメモリ容量が実際のメモリサイズより大きいことを検出した場合、アラーム 6030 "User memory limit has been adjusted" が出力されます。</p>	
関連	<p>使用できるダイナミックメモリは、MD 18050: INFO_FREE_MEM_DYNAMIC (利用可能な DRAM の表示データ) に表示されます。</p>	

18220 MD 番号	MM_USER_MEM_DPR デュアルポート RAM のユーザメモリ	
デフォルト値: 0	最小入力リミット: 0	最大入力リミット: 0
変更が有効になるための条件: 電源オン	保護レベル: 0	単位: KB
データタイプ: DWORD	適用開始 SW バージョン:	
この MD は次の場合は無意味	この機能は、ソフトウェアバージョン 2 では利用できません。	

18230 MD 番号	MM_USER_MEM_BUFFERED SRAM のユーザメモリ	
デフォルト値: NCU 570: 238 NCU 571: 480 NCU 572: 480 NCU 573: 480	最小入力リミット: NCU 570: 100 NCU 571: 100 NCU 572: 100 NCU 573: 100	最大入力リミット: NCU 570: 238 NCU 571: 1900 NCU 572: 1900 NCU 573: 1900
変更が有効になるための条件: 電源オン	保護レベル: 2	単位: KB
データタイプ: DWORD	適用開始 SW バージョン : 1.1 以降	
意味:	<p>非揮発性ユーザメモリの大きさを定義する。このエリアには以下のようなさまざまなタイプのユーザメモリが保存されます。</p> <ul style="list-style-type: none"> <li>- NC パートプログラム</li> <li>- R パラメータ</li> <li>- ツールデータ</li> <li>- ユーザマクロス</li> <li>- グローバルユーザデータ</li> </ul> <p>設定可能な値は、ハードとソフトウェアの構成によって異なります。NCU 571 のハードウェアでは、512 KB が利用可能です。ハードウェアの構成によっては、NCU 572/573 で、512 KB あるいは 2 MB が設定可能です。CNC はオペレーティングシステムに約 30 KB を必要とするため、利用可能な値は 480 KB となります。この利用可能なメモリの中から、マシンデータ、設定データ、およびデータ管理に常に予約されているエリアにメモリが割当てられます。CNC 製造業者は、SRAM のユーザメモリ使用には 256 KB を保証しています。しかし、次のソフトウェアバージョンで使用する場合、256 KB 以上のユーザメモリの利用は保証できません。</p> <p>2 MB の SRAM:</p> <p>NCU 572/573 を大きめのメモリで使用する場合、そのメモリはリリースされなければなりません。</p> <ul style="list-style-type: none"> <li>• MD 18230 に値 1900 を入力する</li> <li>• シリーズスタートアップファイルのコピーを作成する</li> <li>• 電源オンを実行する（メモリを再構成するため）</li> <li>• シリーズスタートアップファイルを制御系にロードし直す</li> </ul> <p>起動中に、システムソフトウェアが、非揮発性メモリの総容量と MD 18230: MM_USER_MEM_BUFFERED にセットされた値を比較します。要求量がマシンデータに定義された容量を越える場合、アラーム 6000 "Memory allocation with standard machine data" が出力されます。起動中に、制御系が MD 18230: MM_USER_MEM_BUFFERED に要求されたメモリ容量が実際のメモリサイズより大きいことを検出した場合、アラーム 6030 "User memory limit has been adjusted" が出力されます。</p>	
例外, エラー	このマシンデータが変更されると、バックアップデータが喪失します。	

18240 MD 番号	MM_LUD_HASH_TABLE_SIZE ユーザ変数のハッシュテーブルサイズ	
デフォルト値 : 11	最小入力リミット : 3	最大入力リミット : 107
変更が有効になるための条件 : 電源オン	保護レベル : 0	単位 : 1 次数
データタイプ : DWORD	適用開始 SW バージョン : 1.1 以降	
意味 :	ローカルユーザメモリデータ (LUD) 用のメモリサイズを定義します。入力値は必ず 1 次数とします。この設定で、次のように最適化されます。 - インタプリタ実行時間 (値が小さくなる = 実行時間が大きくなる) - メモリ必要量 (値が小さくなる = ダイナミックメモリが少なくなる) 要求するテーブルが大きいと、変数のデコーディングのデコーディングオペレーションの数が小さくなり、またインタプリタ実行時間が短くなります。このマシンデータにセットされた値は、REORG を使用するローカルユーザ変数用のモジュール管理に必要なダイナミックメモリ量に影響します。MD 28010: MM_NUM_REORG_LUD_MODULES (REORG (DRAM) を使用するローカルユーザ変数のモジュール数) を参照。	
注記	制御装置によって内部で割当てられるのでユーザが変更してはなりません。	

18242 MD 番号	MM_MAX_SIZE_OF_LUD_VALUE LUD 変数の最大フィールドサイズ	
デフォルト値 : 8192 初期設定 : 496 (SW4.1 以降)	最小入力リミット : 128 240 (SW4.1 以降)	最大入力リミット : 8192 496 (SW4.1 以降)
変更が有効になるための条件 : 電源オン	保護レベル : 2	単位 : バイト
データタイプ : DWORD	適用開始 SW バージョン : 2 以降	
意味 :	MD 18242: MM_MAX_SIZE_OF_LUD_VALUE は、MD : 28040: MM_LUD_VALUES_MEM に定義された総メモリ量がチャンネルのパートプログラムに割当てられるブロックの大きさを指定します。  パートプログラムに発生する最初の変数は、MD 18242: MM_MAX_SIZE_OF_LUD_VALUE に指定された大きさのブロックを占有します。また、次の変数もこのブロックに保存されます。ブロックが変数で一杯になり他の変数を収容する余裕がなくなると、別のブロックが要求されます。  MD 18242: MM_MAX_SIZE_OF_LUD_VALUE には、使用される最大可能値に必要な値をセットする必要があります。  データタイプ 使用メモリ REAL            8 バイト INT             4 バイト BOOL            1 バイト CHAR            1 バイト STRING          キャラクタごとに 1 バイト、ストリングごとに 100 キャラクタ AXIS            4 バイト FRAME           400 バイト	
関連	MD 28040: MM_LUD_VALUES_MEM (ローカルユーザ変数用のメモリサイズ (DRAM))	
例外, エラー	このマシンデータが変更されると、バックアップデータが喪失します。	

18250 MD 番号	MM_CHAN_HASH_TABLE_SIZE チャンネル別データ用のハッシュテーブルサイズ	
デフォルト値: 7	最小入力リミット: 7	最大入力リミット: 193
変更が有効になるための条件: 電源オン	保護レベル: 0	単位: 1 次数
データタイプ: DWORD	適用開始 SW バージョン: 1.1	
意味:	チャンネル別名前用のメモリサイズを定義します。入力値は必ず 1 次数とします。この設定で、次のように最適化されます。 - インタプリタ実行時間 (値が小さくなる = 実行時間が大きくなる) - メモリ必要量 (値が小さくなる = ダイナミックメモリが少なくなる) 要求するテーブルが大きいと、変数のデコーディングのデコーディングオペレーションの数が小さくなり、またインタプリタ実行時間が短くなります。 このマシンデータにセットされた値は、必要なダイナミックメモリ量に影響します。各チャンネル用の要求メモリのバイト数は、入力値に 68 を掛けた値となります。	
例外, エラー	このマシンデータが変更されると、バックアップデータが喪失します。	
注記	このマシンデータは、システムによって内部で割当てられるのでユーザが変更してはいけません。	

18260 MD 番号	MM_NCK_HASH_TABLE_SIZE グローバルデータのハッシュテーブルサイズ	
デフォルト値: 1201	最小入力リミット: 537	最大入力リミット: 1201
変更が有効になるための条件: 電源オン	保護レベル: 0	単位: 1 次数
データタイプ: DWORD	適用開始 SW バージョン: 1.1 以降	
意味:	NCK 別名前用のメモリサイズを定義します。入力値は必ず 1 次数とします。この設定で、次のように最適化されます。 - インタプリタ実行時間 (値が小さくなる = 実行時間が大きくなる) - メモリ必要量 (値が小さくなる = ダイナミックメモリが少なくなる) 要求するテーブルが大きいと、変数のデコーディングのデコーディングオペレーションの数が小さくなり、またインタプリタ実行時間が短くなります。 このマシンデータにセットされた値は、必要なダイナミックメモリ量に影響します。各チャンネル用に必要メモリのバイト数は、入力値に 68 を掛けた値となります。	
例外, エラー	このマシンデータが変更されると、バックアップデータが喪失します。	
注記	このマシンデータは、システムによって内部で割当てられるのでユーザが変更してはなりません。	

18270 MD 番号	MM_NUM_SUBDIR_PER_DIR サブディレクトリの数	
デフォルト値： NCU 570: 16 NCU 571: 24 NCU 572: 30 NCU 573: 30	最小入力リミット： NCU 570: 16 NCU 571: 24 NCU 572: 24 NCU 573: 24	最大入力リミット： NCU 570: 32 NCU 571: 32 NCU 572: 32 NCU 573: 32
変更が有効になるための条件： 電源オン	保護レベル：1	単位：-
データタイプ：DWORD	適用開始 SW バージョン：1.1	
意味：	パッシブファイルシステムのディレクトリが保有することができるサブディレクトリの最大数を定義します。ディレクトリ数は MD 18310: MM_NUM_DIR_IN_FILESYSTEM (パッシブファイルのディレクトリ数) によって制限されます。メモリ容量には、ディレクトリ別のファイル数に必要なメモリも含まれます (参照, MD 18260: MM_NUM_FILES_PER_DIR)。	
関連	MD 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)	

18280 MD 番号	MM_NUM_FILES_PER_DIR ディレクトリ別のファイル数	
デフォルト値： NCU 570: 32 NCU 571: 24 NCU 572: 100 NCU 573: 100	最小入力リミット： NCU 570: 16 NCU 571: 24 NCU 572: 64 NCU 573: 64	最大入力リミット： NCU 570: 104 NCU 571: 512 NCU 572: 512 NCU 573: 512
変更が有効になるための条件： 電源オン	保護レベル：1	単位：-
データタイプ：DWORD	適用開始 SW バージョン：1.1 以降	
意味：	ディレクトリまたはサブディレクトリに作成可能なファイルの最大数を定義します。ファイルの総数は、MD 18320: MM_NUM_FILES_IN_FILESYSTEM (パッシブファイルシステムのファイル数) によって制限されます。ディレクトリのファイル管理に必要なメモリのバイト数は、入力値に 40 を掛けた値となります。MD 設定には MD 18280: MM_NUM_FILES_PER_DIR (ディレクトリ別のファイル数) および MD 18270: MM_NUM_SUBDIR_PER_DIR (サブディレクトリ数) の最大値を入力する必要があります。パッシブファイルシステムのファイル管理に必要なメモリは、MD 18320: MM_NUM_FILES_IN_FILESYSTEM によって予約されます。	
例外, エラー	このマシンデータが変更されると、バックアップデータが喪失します。 注記： MD の変更は、変更後に作成されるディレクトリに影響を与えます。すなわち、既存のディレクトリのファイル数が増えたり減ったりすると、最初にそのディレクトリが削除され、その後新規ディレクトリが作成されます (ただし、ファイルがまず保存されていなければ作成されません)。	
関連	MD 18320: MM_NUM_FILES_IN_FILESYSTEM (パッシブファイルシステムのファイル数)	

18290 MD 番号	MM_FILE_HASH_TABLE_SIZE ディレクトリのファイル用のハッシュテーブルサイ	
デフォルト値： NCU 570: 19 NCU 571: 19 NCU 572: 19 NCU 573: 19	最小入力リミット： NCU 570: 3 NCU 571: 3 NCU 572: 3 NCU 573: 3	最大入力リミット： NCU 570: 67 NCU 571: 67 NCU 572: 299 NCU 573: 299
変更が有効になるための条件： 電源オン	保護レベル：0	単位：1 次数
データタイプ：DWORD	適用開始 SW バージョン：1.1	
意味：	ディレクトリのファイル用のハッシュテーブルの大きさを定義します。入力値は必ず 1 次数とします。この設定で、次のように最適化されます。 - インタプリタ実行時間（値が小さくなる = 実行時間が大きくなる） - メモリ必要量（値が小さくなる = ダイナミックメモリが少なくなる） このマシンデータの値は、ディレクトリ管理に必要なスタティックメモリ量に影響を与えます。MD 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 ) を参照。	
例外，エラー	このマシンデータが変更されると、バックアップデータが喪失します。	
注記	このマシンデータは、システムによって内部で割当てられるのでユーザが変更してはいけません。	

18300 MD 番号	MM_DIR_HASH_TABLE_SIZE サブディレクトリのハッシュテーブルサイズ	
デフォルト値：7	最小入力リミット：3	最大入力リミット：37
変更が有効になるための条件： 電源オン	保護レベル：0	単位：1 次数
データタイプ：DWORD	適用開始 SW バージョン：1.1 以降	
意味：	サブディレクトリのファイル用のハッシュテーブルの大きさを定義します。入力値は必ず 1 次数とします。この設定で、次のように最適化されます。 - インタプリタ実行時間（値が小さくなる = 実行時間が大きくなる） - メモリ必要量（値が小さくなる = ダイナミックメモリが少なくなる） このマシンデータの値は、ディレクトリ管理に必要なスタティックメモリ量に影響を与えます。MD 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 ) を参照。	
例外，エラー	このマシンデータが変更されると、バックアップデータが喪失します。	
注記	このマシンデータは、システムによって内部で割当てられるのでユーザが変更してはいけません。	



18310 MD 番号	M_NUM_DIR_IN_FILESYSTEM パッシブファイルシステムのディレクトリ数	
デフォルト値： NCU 570: 32 NCU 571: 30 NCU 572: 30 NCU 573: 30	最小入力リミット： NCU 570: 24 NCU 571: 30 NCU 572: 30 NCU 573: 30	最大入力リミット： NCU 570: 64 NCU 571: 150 NCU 572: 150 NCU 573: 150
変更が有効になるための条件： 電源オン	保護レベル：2	単位：-
データタイプ：DWORD	適用開始 SW バージョン：1.1 以降	
意味：	<p>このマシンデータによりパッシブファイルシステムのディレクトリ数が制限されます。またこのマシンデータを使用してディレクトリの管理用に SRAM にメモリを予約することができます。システムによってセットアップされたパッシブファイルシステムのディレクトリまたはサブディレクトリは、このマシンデータに含まれます。ディレクトリ管理に必要なメモリは、次のように計算されます。</p> <p>メモリ要求量 = a (440+28 (b+c)) バイト</p> <p>a = MD 18310: MM_NUM_DIR_IN_FILESYSTEM (パッシブファイルのディレクトリ数) の入力値</p> <p>b = MD 19300: MM_DIR_HASH_TABLE_SIZE (サブディレクトリ用のハッシュテーブル) の入力値</p> <p>c = MD 18290: MM_FILE_HASH_TABLE_SIZE (ディレクトリのファイル用のハッシュテーブルサイズ) の入力値</p>	
例外、エラー	このマシンデータが変更されると、バックアップデータが喪失します。	
関連	MD 18270: MM_NUM_SUBDIR_PER_DIR (サブディレクトリ数)	

18320 MD 番号	MM_NUM_FILES_IN_FILESYSTEM パッシブファイルシステムのファイル数	
デフォルト値： NCU 570: 64 NCU 571: 128 NCU 572: 100 NCU 573: 100	最小入力リミット： NCU 570: 24 NCU 571: 24 NCU 572: 64 NCU 573: 64	最大入力リミット： NCU 570: 128 NCU 571: 512 NCU 572: 512 NCU 573: 512
変更が有効になるための条件： 電源オン	保護レベル：2	単位：-
データタイプ：DWORD	適用開始 SW バージョン：1.1	
意味：	<p>パートプログラムメモリで利用できるファイル数を定義します。このマシンデータを使用してファイル管理用に SRAM にメモリを予約することができます (約 320 バイト)。作成されたファイルはそれぞれ、ファイルコード用に最低 1 K バイトを必要とします。ファイルコード用のこの 1 K バイトリミットを超えた場合、別にまた 1 K バイトがファイル用に予約されます。</p>	
例外、エラー	このマシンデータが変更されると、バックアップデータが喪失します。	
関連	MD 18280: MM_NUM_FILES_PER_DIR (ディレクトリのファイル数)	

18330 MD 番号	MM_CHAR_LENGTH_OF_BLOCK NC ブロックの最大長	
デフォルト値: 256	最小入力リミット: 32	最大入力リミット: 512
変更が有効になるための条件: 電源オン	保護レベル: 0	単位: キャラクタ
データタイプ: DWORD	適用開始 SW バージョン: -	
意味:	なし	
この MD は次の場合は無意味	この機能は、ソフトウェアバージョン 2 では利用できません。	

18340 MD 番号	MM_NUM_CEC_NAMES LEC テーブル数	
デフォルト値: 4	最小入力リミット: 4	最大入力リミット: 4
変更が有効になるための条件: 電源オン	保護レベル: 0	単位: -
データタイプ: DWORD	適用開始 SW バージョン: 1.1 以降	
意味:	MM_NUM_CEC_NAMES を変更することはできません。 CEC（クロス誤差補償）またはリードスクリュエラー補償用データの参照は NC プログラムで実行されます。このため、MM_NUM_CEC_NAMES に指定された名前の最大数を格納するためのデータブロックが設けられています。実際の補償テーブルはこれらの名前のいずれか 1 つに割り当てられています。補償値は、バックアップユーザメモリに保存されますが、その数は MD 38000: MM_ENC_COMP_MAX_POINTS（補間補償を使用する補間点の数）に定義することができます。メモリ要求量は、このソフトウェアバージョンの場合、約 1 KB です。	
例外, エラー	ソフトウェアバージョン 2 では、CEC データに含まれる名前は恒久的に割り当てられます。このマシンデータが変更されると、バックアップデータが喪失します。	

18342 MD 番号	MM_CEC_MAX_POINTS ビームの垂れ補償用の最大テーブルサイズ	
デフォルト値: 0	最小入力リミット: 0	最大入力リミット: 2000
変更が有効になるための条件: 電源オン	保護レベル: 2	単位: -
データタイプ: DWORD	適用開始 SW バージョン: 2 以降	
意味:	軸同士の補間補償用の最大テーブルサイズ MM_CEC_MAX_POINTS = 0 のとき、テーブル用にメモリはセットされません。その場合、ビームの垂れの補償ファンクションは使用できません。	
例外:	このマシンデータが変更されると、バッファメモリエリアが再構成されます。	

18350 MD 番号	MM_USER_FILE_MEM_MINIMUM 最小パートプログラムメモリ	
デフォルト値 : 20	最小入力リミット : 20	最大入力リミット : 100
変更が有効になるための条件 : 電源オン	保護レベル : 1	単位 : KB
データタイプ : DWORD	適用開始 SW バージョン : 1.1	
意味 :	パッシブファイルシステムのファイル用に残されている最小バックアップメモリを定義します。設定可能な値は、ハードウェアとソフトウェアの構成 (SRAM メモリ割当て) および MD 18230: MM_USER_MEM_BUFFERED (SRAM ユーザメモリ) によって異なります。SRAM メモリが割当てられると、残メモリがパッシブファイルシステムに割当てられます。ファイルシステムを確実に作動させるために、MM_USER_FILE_MEM_MINIMUM に指定されたメモリ空間がメモリに利用できなければなりません。それができない場合、制御のデフォルト値によってメモリが割当てられ、ユーザが SRAM に保存したデータは全て喪失し、アラーム 6000 "Memory allocation with standard machine data" が出力されます。パートプログラムの利用可能メモリ容量は、MD 18060: INFO_FREE_MEM_STATIC (スタティックメモリ空き量) に表示されます。	
例外, エラー	このマシンデータが変更されると、バックアップデータが喪失します。このマシンデータが変更され、遺留メモリ量が MM_USER_FILE_MEM_MINIMUM の値よりも小さい場合、バックアップデータは喪失します。	

18500 MD 番号	MM_EXTCOM_TASK_STACK_SIZE 外部通信タスク用のスタックサイズ	
デフォルト値 : 8	最小入力リミット : 4	最大入力リミット : 20
変更が有効になるための条件 : 電源オン	保護レベル : 0	単位 : KB
データタイプ : DWORD	適用開始 SW バージョン : 1.1 以降	
意味 :	外部通信用のスタックサイズ。ダイナミックメモリエリアが使用されます。	
注記	このマシンデータは、システムによって内部で割当てられるのでユーザが変更してはいけません。	

18510 MD 番号	MM_SERVO_TASK_STACK_SIZE サーボタスク用のスタックサイズ	
デフォルト値 : 8	最小入力リミット : 4	最大入力リミット : 20
変更が有効になるための条件 : 電源オン	保護レベル : 0	単位 : KB
データタイプ : DWORD	適用開始 SW バージョン : 1.1	
意味 :	サーボタスク用のスタックサイズを定義します。ダイナミックメモリが使用されます。	
注記	このマシンデータは、システムによって内部で割当てられるのでユーザが変更してはいけません。	

18520 MD 番号	MM_DRIVE_TASK_STACK_SIZE ドライブタスク用のスタックサイズ	
デフォルト値 : 8	最小入力リミット : 4	最大入力リミット : 20
変更が有効になるための条件 : 電源オン	保護レベル : 0	単位 : KB
データタイプ : DWORD	適用開始 SW バージョン : 1.1 以降	
意味 :	このマシンデータを使用して、SIMODRIVE タスク用のスタックサイズが定義されます。このスタックはダイナミックメモリエリアに保存されます。	
注記	このマシンデータは、システムによって内部で割当てられるのでユーザが変更してはいけません。	

## ■ チャンネル別マシンデータ

27900 MD 番号	REORG_LOG_LIMIT ログファイルリリース用の IPO バッファのパーセント	
デフォルト値 : 1	最小入力リミット : -127	最大入力リミット : 127
変更が有効になるための条件 : 電源オン	保護レベル : 0	単位 : -
データタイプ : BYTE	適用開始 SW バージョン : 1.1	
意味 :	ブロック準備が REORG LOG データメモリのオーバフローのために中断された場合に RE-ORG LOG メモリ内のデータが各段階でそのパーセントを超えてリリースできる IPO バッファのパーセントを定義します。リリースされたデータは REORG ファンクション ( 参照 : /FB/, K1, "Mode Groups, Channels, Pro-gram Operation Mode") には今後使用することはできません。このステータスの結果, エラーメッセージと共にそれから先の REORG コマンドはキャンセルされます。"non-reorganizability" のステータスが発生すると, 警告 15110 が出力されます。最上位ビットをイネーブルにすることで, 警告の出力を防止することができます。このビットは, REORG_LOG_LIMIT の入力値に 128 を加えると設定されます。NC ブロックのインストラクションの他に, IPO バッファならびに REORG データメモリもデータリリースの頻度に影響を与えます。	
関連	MD 28000: MM_REORG_LOG_FILE_MEM ( REORG のメモリサイズ ) MD 28060: MM_IPO_BUFFER_SIZE ( IPO バッファ内のブロック数 )	

28000 MD 番号	MM_REORG_LOG_FILE_MEM REORG 用のメモリサイズ	
デフォルト値 : NCU 570: 10 NCU 571: 10 NCU 572: K1=10, K2=1 NCU 573: K1=10, K2=1	最小入力リミット : NCU 570: 0 NCU 571: 0 NCU 572: 0 NCU 573: 0	最大入力リミット : NCU 570: 50 NCU 571: 500 NCU 572: 500 NCU 573: 500
変更が有効になるための条件 : 電源オン	保護レベル : 2	単位 : KB
データタイプ : DWORD	適用開始 SW バージョン : 1.1 以降	
意味 :	REORG LOG データ用のダイナミックメモリの大きさを定義します。このメモリサイズが REORG ファンクションに利用できるデータ量を定義します。	
参照	1.10 運転モードの種類とプログラム運転モード (K1)	

28010 MD 番号	MM_NUM_REORG_LUD_MODULES REORG を使用するローカルユーザ変数用のモジュール数	
デフォルト値: 4	最小入力リミット: 0	最大入力リミット: 100
変更が有効になるための条件: 電源オン	保護レベル: 2	単位: -
データタイプ: DWORD	適用開始 SW バージョン: 1.1	
意味:	<p>REORG ファンクション用に設定された追加 LUD モジュールの数を定義します (参照, De-scription of Functions, Mode Groups, Channels, Program Operation Mode (K1)) (ファンクションの説明, モードグループ, チャンネル, プログラムオペレーションモード (K1))。REORG ファンクションが使用されない場合, この値は 0 となります。CNC は常に 12 LUD モジュールをオープンにします: NC プログラム用に 8, 非同期サブグループ用に 4。ローカル変数の定義を持つ NC プログラムまたは非同期サブグループのそれぞれ 1 つにつき LUD モジュールが 1 つ必要です。IPO バッファが大きくなると REORG に対するこの値も増す必要があります, また LUD 変数定義を持つ数多くのショート NC プログラムがアクティブになります (プログラムの NC ブロックは, IPO バッファ内に作成されたフォーマットで保存されます)。これらのプログラムの各々に LUD モジュールが 1 つ必要です。予約メモリの容量は, NC プログラム別の LUD 数ならびにそれぞれのメモリの影響を受けます。</p> <p>LUD モジュールはダイナミックメモリに保存されます。</p> <p>REORG を使用するローカルユーザ変数用のモジュール管理に必要なメモリは, 次のように計算します。</p> <p>メモリ = <math>a \times (200 + b \times 160)</math> バイト</p> <p><math>a</math> = LUD モジュールの総数 = <math>8 + 4 + \text{MD 内の値:}</math> MM_NUM_REORG_LUD_MODULES</p> <p><math>b</math> = MD 18240 の入力値: MM_LUD_HASH_TABLE_SIZE (ユーザ変数用のハッシュテーブルサイズ)</p> <p>LUD モジュールの大きさは, アクティブ LUD とそのデータタイプによって異なります。</p> <p>LUD モジュール用のメモリは, MD 28000: MM_REORG_LOG_FILE_MEM (REORG 用のメモリサイズ) によって制限されます。</p>	
アプリケーション	<p>例:</p> <p>4 つの NC ブロックを持つメインプログラム:</p> <ul style="list-style-type: none"> <li>- 最初のブロックに, LUD 変数が 1 つ定義される。</li> <li>- 第 2 と第 3 のブロックのそれぞれに, サブプログラム (最大 8 レベルまでネストされる) が呼び出される。</li> <li>- 第 3 ブロックがサブプログラムを終了する</li> </ul> <p>サブプログラムは各々 3 つの NC ブロックを保有する:</p> <ul style="list-style-type: none"> <li>- 最初のブロックに, LUD 変数が 1 つ定義される。</li> <li>- 次のプログラムに呼び出されたサブプログラムが, 第 2 ブロックで実行される。</li> <li>- 第 3 ブロックがサブプログラムを終了する。</li> </ul> <p>サブプログラムコールの他に, 最後のプログラムレベルのサブプログラムには, トラバース動作などのさまざまなコマンドが含まれています。それにより, 46 の NC ブロックを持つ総計 15 のプログラムを構成し, 準備されたフォーマットでその全てを IPO バッファ内に保存することができます。</p> <p>REORG ファンクションには 46 のブロックの全てのデータが必要となるため, 3 つのプログラムの LUD モジュールが未着となります。追加として必要となる Lud データブロックに対して, 値 3 を MM_NUM_REORG_LUD_MODULES に任意の例として入力する必要があります。</p>	

28020 MD 番号	MM_NUM_LUD_NAMES_TOTAL ローカルユーザ変数の数	
デフォルト値： NCU 570: 200 NCU 571: 200 NCU 572: K1=200, K2=20 NCU 573: K1=200, K2=20	最小入力リミット： NCU 570: 0 NCU 571: 0 NCU 572: 0 NCU 573: 0	最大入力リミット： NCU 570: 300 NCU 571: 500 NCU 572: 500 NCU 573: 500
変更が有効になるための条件： 電源オン	保護レベル：0	単位：-
データタイプ：DWORD	適用開始 SW バージョン：1.1 以降	
意味：	<p>プログラムのアクティブセクションに存在することが許可されたローカルユーザデータ (LUD) 用の変数の数を定義します。変数の名前および変数値に対して、変数ごとに約 150 バイトのメモリが予約されます。変数値に必要なメモリは、データタイプのサイズと同等です。アクティブメインプログラムおよび関連のサブプログラムから要求されるローカルユーザ変数の総数が定義値リミットを超える場合、リミットを越えた変数は、そのプログラムの実行中は、受諾されません。変数名と変数値にはダイナミックメモリが使用されます。</p> <p>データタイプが使用するメモリの概要：</p> <p>データタイプ 使用メモリ</p> <p>REAL            8 バイト</p> <p>INT             4 バイト</p> <p>BOOL            1 バイト</p> <p>CHAR            1 バイト</p> <p>STRING        キャラクタごとに 1 バイト、ストリングごとに 200 キャラクタ</p> <p>AXIS            4 バイト</p> <p>FRAME          400 バイト</p>	

28040 MD 番号	MM_LUD_VALUES_MEM ローカルユーザ用メモリサイズ	
デフォルト値：20, 20, 20, 20, 20, 20, 20, ...,	最小入力リミット：1	最大入力リミット：100
変更が有効になるための条件： 電源オン	保護レベル：2	単位：KB
データタイプ：DWORD	適用開始 SW バージョン：4.3	
意味：	<p>この MD は、LUD 変数に利用可能なメモリ空間の量を定義します。</p> <p>変数 LUD は、MD 28020: MM_NUM_LUD_NAMES_TOTAL または MM_LUD_VALUES_MEM のいずれかの制限値のどれか一つにでも達した場合、変数が全て消費されたとみなされます。ここに定義されるメモリは、<math>(MM\_LUD\_VALUES\_MEM * 1024) / MM\_MAX\_SIZE\_OF\_LUD\_VALUE</math> ブロックに再分割され、メモリを要求するパートプログラムに割当てられます。パートプログラムはそれぞれ、少なくとも LUD 変数の定義を 1 つ持っているか、あるいはコールパラメータを持っているので、当該ブロックを少なくとも 1 つは使用します。ただし、複数のパートプログラムが同時にオープンし、NCK のメモリを使用する場合があることに注意してください。この数は、設定のタイプ、プログラム長、<math>(MM\_IPO\_BUFFER\_SIZE, MM\_NUM\_BLOCKS\_IN\_PREP)</math> より上流の内部 NCK ブロックメモリのサイズによって異なります。</p>	
関連：	MD 28020: MM_NUM_LUD_NAMES_TOTAL (ローカルユーザ変数の数 (DRAM))	

28050 MD 番号	MM_NUM_R_PARAM チャンネル別 R パラメータの数	
デフォルト値： 100, 100, 100, 100, 100, 100, 100, ...	最小入力リミット：0	最大入力リミット：10000
変更が有効になるための条件： 電源オン	保護レベル：0/0	単位：-
データタイプ：DWORD	適用開始 SW バージョン：4.3	
意味：	チャンネルで使用できる R パラメータの数を定義します。各チャンネルには最大で 1000 R パラメータが利用できます。このマシンデータを使用し、R パラメータのそれぞれに（バイトのバックアップユーザメモリを予約することができます。R パラメータは基本的に、LUD や GUD 変数と比べて管理オーバーヘッドが少なく済みます。	
例外，エラー	このマシンデータが変更されると、バックアップデータが喪失します。	

28060 MD 番号	MM_IPO_BUFFER_SIZE IPO バッファの NC ブロック数	
デフォルト値： 10, 10, 10, 10, 10, 10, 10, 10	最小入力リミット：2	最大入力リミット： NCU 570: 30 NCU 571: 15 NCU 572, 573: 300
変更が有効になるための条件： 電源オン	保護レベル：0/0	単位：-
データタイプ：DWORD	適用開始 SW バージョン：4.3	
意味：	補間バッファのブロック数を定義します。このバッファには、保管用に設定された準備 NC ブロックが含まれます。ダイナミックユーザメモリが約 10 K バイト、各 NC ブロックに予約されます。さらに、このデータは、Look Ahead（ルックアヘッド）ファンクションの速度制限用の Look Ahead（ルックアヘッド）ブロックの数を制限します。MM_IPO_BUFFER_SIZE がシステムによってセットされます。	
関連：	MD 28070: MM_NUM_BLOCKS_IN_PREP（ブロック準備用のブロック数）	



28070 MD 番号	MM_NUM_BLOCKS_IN_PREP ブロック準備用のブロック数	
デフォルト値： NCU 570: 36, 36, 36, 36, 36, 36, 36, 36 NCU 571: 26, 26, 26, 26, 26, 26, 26, 26 NCU 572: 36, 36, 36, 36, 36, 36, 36, ... NCU 573: 36, 36, 36, 36, 36, 36, 36, ...	最小入力リミット： 20	最大入力リミット： NCU 570: *** NCU 571: *** NCU 572: *** NCU 573: ***
変更が有効になるための条件： 電源オン	保護レベル：0/0	単位：内部ブロック数
データタイプ：DWORD	適用開始 SW バージョン：4.3	
意味：	ブロック準備に利用できるブロック数を定義します。この数値は、主としてシステムソフトウェアによって求められ、最適化に使用されます。ダイナミックメモリが約 10 K バイト各 NC ブロックに予約されます。	
関連：	MD 28060: MM_IPO_BUFFER_SIZE (IPO バッファを使用する NC ブロックの数)	

28080 MD 番号	MM_NUM_USER_FRAMES 設定可能なフレーム数	
デフォルト値：5	最小入力リミット：5	最大入力リミット：100
変更が有効になるための条件： 電源オン	保護レベル：2	単位：-
データタイプ：DWORD	適用開始 SW バージョン：1.1	
意味：	事前定義されたユーザフレームの数を定義します。フレームには、約 400 バイトのバックアップメモリが予約されています。システムの標準設定では、G54 から G57 の 4 つのフレームと G500 用のフレーム 1 つが提供されます。	
例外，エラー	このマシンデータが変更されると、バックアップデータが喪失します。	

28085 MD 番号	MM_LINK_TOA_UNIT チャンネルへの TO ユニットの割当て	
デフォルト値： NCU 570: 1 NCU 571: 1 NCU 572: [ チャンネル番号 ] NCU 573: [ チャンネル番号 ]	最小入力リミット： NCU 570: 1 NCU 571: 1 NCU 572: 1 NCU 573: 1	最大入力リミット： NCU 570: 1 NCU 571: 1 NCU 572: 2 NCU 573: 2
変更が有効になるための条件： 電源オン	保護レベル：2	単位：-
データタイプ：DWORD	適用開始 SW バージョン：2 以降	
意味：	<p>初期設定を通じて TO ユニットが各チャンネルに割当てられます。したがって、メモリはデータブロック（ツール、マガジン）に予約されます。また、TOA ユニットは複数のチャンネルにも割当てることができます。</p> <p>定義：TOA エリアは、NC の全ての TOA とマガジンブロックの合計です。TOA ユニットには、TOA ブロックと、TM ファンクションを使用している場合にはマガジンブロックが含まれます。</p>	
例外，エラー	このマシンデータが変更されると、バックアップデータが喪失します。	

28090 MD 番号	MM_NUM_CC_BLOCK_ELEMENTS コンパイルサイクル用のブロックメモリ数	
デフォルト値： NCU 570: 0 NCU 571: 0 NCU 572: 0 NCU 573: 0	最小入力リミット： NCU 570: 0 NCU 571: 0 NCU 572: 0 NCU 573: 0	最大入力リミット： NCU 570: 0 NCU 571: 0 NCU 572: 10 NCU 573: 10
変更が有効になるための条件： 電源オン	保護レベル：0	単位：-
データタイプ：DWORD	適用開始 SW バージョン：1.1 以降	
意味：	この値は、コンパイルサイクルに使用されるブロックエレメントの数を定義します。ソフトウェアバージョン 2 では、それぞれのブロックエレメントには約 1.2 KB のダイナミックメモリが必要です。	

28100 MD 番号	MM_NUM_CC_BLOCK_USER_MEM コンパイルサイクル用のブロックメモリサイズ	
デフォルト値： NCU 570: 0 NCU 571: 0 NCU 572: 256 NCU 573: 256	最小入力リミット： NCU 570: 0 NCU 571: 0 NCU 572: 0	最大入力リミット： NCU 570: 256 NCU 571: 256 NCU 572: 256 NCU 573: 256
変更が有効になるための条件： 電源オン	保護レベル：0	単位：KB
データタイプ：DWORD	適用開始 SW バージョン：1.1	
意味：	この値は、ダイナミックメモリ内でユーザがコンパイルサイクルに使用できるブロックメモリの総容量を定義します。このメモリは、複数のバイトから構成されるブロック単位で割当てられます。	

28200 MD 番号	MM_NUM_PROTECT_AREA_CHAN チャンネル別保護ゾーン用のブロック数	
デフォルト値： NCU 570: 0 NCU 571: 0 NCU 572: 0 NCU 573: 0	最小入力リミット： NCU 570: 0 NCU 571: 0 NCU 572: 0 NCU 573: 0	最大入力リミット： NCU 570: 0 NCU 571: 0 NCU 572: 10 NCU 573: 10
変更が有効になるための条件： 電源オン	保護レベル：2	単位：
データタイプ：DWORD	適用開始 SW バージョン：2 以降	
意味：	このマシンデータは、チャンネル別保護ゾーンに対して作成するブロック数を定義します。	
関連	MD 28210: MM_NUM_PROTECT_AREA_ACTIVE (同時にアクティブになる保護ゾーンの数) MD 18190: MM_NUM_PROTECT_AREA_NCK (マシン関連保護ゾーン用のファイル数 (SRAM))	
参照	1.2 送り軸監視機能及びプロテクションゾーン (A3)	

28210 MD 番号	MM_NUM_PROTECT_AREA_ACTIVE 同時にアクティブになる保護ゾーンの数	
デフォルト値： NCU 570: 0 NCU 571: 0 NCU 572: 0 NCU 573: 0	最小入力リミット： NCU 570: 0 NCU 571: 0 NCU 572: 0 NCU 573: 0	最大入力リミット： NCU 570: 4 NCU 571: 4 NCU 572: 10 NCU 573: 10
変更が有効になるための条件： 電源オン	保護レベル：2	単位：-
データタイプ：DWORD	適用開始 SW バージョン：2 以降	
意味：	このマシンデータは、チャンネル別保護ゾーンに対して作成するブロック数を定義します。 次に設定されているより大きな値は入力しないでください。 MD 18190: MM_NUM_PROTECT_AREA_NCK + MD 28200: MM_NUM_PROTECT_AREA_CHAN.	
関連	MD 28200: MM_NUM_PROTECT_AREA_CHAN (チャンネル別保護ゾーン用のブロック数) MD 18190: MM_NUM_PROTECT_AREA_NCK (マシン関連保護ゾーン用のファイル数 (SRAM))	
参照	1.2 送り軸監視機能及びプロテクションゾーン (A3)	

28500 MD 番号	MM_PREP_TASK_STACK_SIZE 準備タスク用のスタックサイズ	
デフォルト値： NCU 570: 12 NCU 571: 12 NCU 572: 20 NCU 573: 20	最小入力リミット： NCU 570: 4 NCU 571: 4 NCU 572: 4 NCU 573: 4	最大入力リミット： NCU 570: 40 NCU 571: 40 NCU 572: 40 NCU 573: 40
変更が有効になるための条件： 電源オン	保護レベル：0	単位：KB
データタイプ：DWORD	適用開始 SW バージョン：1.1 以降	
意味：	このマシンデータは、チャンネル別保護ゾーンに対して作成するブロック数を定義します。	
注記	このマシンデータは、システムによって内部で割当てられるのでユーザが変更してはいけません。	

28510 MD 番号	MM_IPO_TASK_STACK_SIZE IPO タスク用のスタックサイズ	
デフォルト値： NCU 570: 8 NCU 571: 8 NCU 572: 12 NCU 573: 12	最小入力リミット： NCU 570: 4 NCU 571: 4 NCU 572: 4 NCU 573: 4	最大入力リミット： NCU 570: 40 NCU 571: 40 NCU 572: 40 NCU 573: 40
変更が有効になるための条件： 電源オン	保護レベル：0	単位：KB
データタイプ：DWORD	適用開始 SW バージョン：1.1	
意味：	IPO タスク用のスタックサイズはこのデータに保存されます。このスタックは、ダイナミックメモリにセットアップされます。	
注記	このマシンデータは、システムによって内部で割当てられるのでユーザが変更してはいけません。	

28550 MD 番号	MM_PRSATZ_MEM_SIZE 内部ブロックに利用可能なメモリ	
デフォルト値：400	最小入力リミット：100	最大入力リミット：4000
変更が有効になるための条件： 電源オン	保護レベル：0	単位：KB
データタイプ：DWORD	適用開始 SW バージョン：1.1 以降	
意味：	なし。 この MD は、ソフトウェアバージョン 2 には含まれていません。	

## ■ 軸別マシンデータ

38000 MD 番号	MM_ENC_COMP_MAX_POINTS 補間補償用の補間点数	
デフォルト値 : 0	最小入力リミット : 0	最大入力リミット : 5000
変更が有効になるための条件 : 電源オン	保護レベル : 2	単位 : -
データタイプ : DWORD	適用開始 SW バージョン : 1,1	
意味 :	軸用のエンコーダ別のリードスクリュウ補償値の数を定義します。この値は、補償値ごとに 8 バイトのユーザメモリを予約します。SRAM で使用できるより以上の補償値メモリが要求される場合、制御はアラーム 6000 "Memory allocation with standard machine data" を起動時に出力します。	
例外, エラー	このマシンデータが変更されると、バックアップデータが喪失します。	
参照	2.5 補正機能 (K3)	

38010 MD 番号	MM_QEC_MAX_POINTS クォッドラント誤差補償値の数	
デフォルト値 : 0	最小入力リミット : 0	最大入力リミット : 1040
変更が有効になるための条件 : 電源オン	保護レベル : 2	単位 : -
データタイプ : DWORD	適用開始 SW バージョン : 2 以降	
意味 :	ニューラルネットワーク (オプション) を使用するクォッドラント誤差補償用に使用できる値の数。 "0" が設定されている場合 : クォッドラント誤差補償ファンクションは起動しません。このためファンクション用のメモリはセットアップされません。	
例外, エラー	このマシンデータが変更されると、バックアップデータが喪失します。	
参照	/IAD/, "SINUMERIK 840D Installation and Start-up Guide" /FB/, K3, "Compensations"	

### 2.14.5 信号の説明

なし

### 2.14.6 例

なし

### 2.14.7 データフィールド、リスト

#### ■ マシンデータ

番号	識別子	名称	参照
一般事項 (\$MN_ ... )			
18050	INFO_FREE_MEM_DYNAMIC	ダイナミックメモリ空き量の表示データ	
18060	INFO_FREE_MEM_STATIC	スタティックメモリ空き量の表示データ	
18070	INFO_FREE_MEM_DPR	デュアルポート RAM のメモリ空き量の表示データ	
18080	MM_TOOL_MANAGEMENT_MASK	TM メモリ予約用の画面フォーム	/FBW/
18082	MM_NUM_TOOL	NCK が管理するツール数	
18084	MM_NUM_MAGAZINE	NCK が管理するマガジン数	/FBW/
18086	MM_NUM_MAGAZINE_LOCATION	マガジン位置の数	/FBW/
18090	MM_NUM_CC_MAGAZINE_PARAM	TM コンパイルサイクル: マガジンデータ数	/FBW/
18092	MM_NUM_CC_MAGLOC_PARAM TM	コンパイルサイクル: マガジン位置データの数	/FBW/

番号	識別子	名称	参照
一般事項 (\$MN_ ... )			
18094	MM_NUM_CC_TDA_PARAM	TM コンパイルサイクル：TDA データの数	/FBW/
18096	MM_NUM_CC_TOA_PARAM	TM コンパイルサイクル：TOA データの数	/FBW/
18098	MM_NUM_CC_MON_PARAM	TM コンパイルサイクル：監視データの数	/FBW/
18100	MM_NUM_CUTTING_EDGES_IN_TOA	NCK のツールオフセット数	
18118	MM_NUM_GUD_MODULES	GUD ブロック数	
18120	MM_NUM_GUD_NAMES_NCK	グローバルユーザ変数の数	
18130	MM_NUM_GUD_NAMES_CHAN	チャンネル別ユーザ変数の数	
18140	MM_NUM_GUD_NAMES_AXIS	軸別ユーザ変数の数	
18150	MM_GUD_VALUES_MEM	グローバルユーザ変数のメモリ容量	
18160	MM_NUM_USER_MACROS	マクロ数	
18170	MM_NUM_MAX_FUNC_NAMES	付帯ファンクション数	
18180	MM_NUM_MAX_FUNC_PARAM	付帯ファンクションパラメータの数	
18190	MM_NUM_PROTECT_AREA_NCK	NCK の保護ゾーン数	/FB/, A3
18210	MM_USER_MEM_DYNAMIC	DRAM のユーザメモリ	
18220	MM_USER_MEM_DPR	デュアルポート RAM のユーザメモリ	
18230	MM_USER_MEM_BUFFERED	SRAM のユーザメモリ	
18240	MM_LUD_HASH_TABLE_SIZE	ユーザ変数のハッシュテーブルサイズ	
18242	MM_MAX_SIZE_OF_LUD_VALUE	LUD 変数の最大フィールドサイズ	
18250	MM_CHAN_HASH_TABLE_SIZE	チャンネル別データ用のハッシュテーブルサイズ	
18260	MM_NCK_HASH_TABLE_SIZE	グローバルデータ用のハッシュテーブルサイズ	
18270	MM_NUM_SUBDIR_PER_DIR	サブディレクトリ数	
18280	MM_NUM_FILES_PER_DIR	ディレクトリ別ファイル数	
18290	MM_FILE_HASH_TABLE_SIZE	ディレクトリのファイル用のハッシュテーブルサイズ	
18300	MM_DIR_HASH_TABLE_SIZE	サブディレクトリ用のハッシュテーブルサイズ	
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	
18330	MM_CHAR_LENGTH_OF_BLOCK	NC ブロックの最大長	
18340	MM_NUM_CEC_NAMES	CEC 補償テーブル数	
18342	MM_CEC_MAX_POINTS	垂れ補償用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	
18500	MM_EXTCOM_TASK_STACK_SIZE	外部通信タスク用のスタックサイズ	
18510	MM_SERVO_TASK_STACK_SIZE	サーボタスク用のスタックサイズ	
18520	MM_DRIVE_TASK_STACK_SIZE	ドライブタスク用のスタックサイズ	
チャンネル別 (\$MC_ ... )			
27900	REORG_LOG_LIMIT	ログファイルリリース用の IPO バッファの割合	
28000	MM_REORG_LOG_FILE_MEM	REORG 用のメモリ容量	/FB/, K1

番号	識別子	名称	参照
一般事項 (\$MN_ ... )			
28010	MM_NUM_REORG_LUD_MODULES	REORG を使用するローカルユーザメモリ用モジュールの数	
28020	MM_NUM_LUD_NAMES_TOTAL	ローカルユーザ変数の数	
28040	MM_LUD_VALUES_MEM	ローカルユーザメモリ変数用のメモリ容量	
28050	MM_NUM_R_PARAM	チャンネル別 R パラメータ数	
28060	MM_IPO_BUFFER_SIZE	IPO バッファの NC ブロック数	
28070	MM_NUM_BLOCKS_IN_PREP	ブロック準備用のブロック数	
28080	MM_NUM_USER_FRAMES	設定可能フレームの数	
28085	MM_LINK_TOA_UNIT	チャンネルへの TO ユニットの割当て	
28090	MM_NUM_CC_BLOCK_ELEMENTS	コンパイルサイクル用のブロックエレメント数	
28100	MM_NUM_CC_BLOCK_USER_MEM	コンパイルサイクル用のブロックエレメントの容量	
28200	MM_NUM_PROTECT_AREA_CHAN	チャンネル別保護ゾーン用のブロック数	/FB/, A3
28210	MM_NUM_PROTECT_AREA_ACTIVE	同時にアクティブになる保護ゾーンの数	/FB/, A3
28500	MM_PREP_TASK_STACK_SIZE	準備タスク用のスタックサイズ	
28510	MM_IPO_TASK_STACK_SIZE	IPO タスク用のスタックサイズ	
28550	MM_PRSATZ_MEM_SIZE	内部ブロックに利用可能なメモリ	
軸別 (\$MC_ ... )			
38000	MM_ENC_COMP_MAX_POINTS	補間補償用の補間点数	/FB/, K3
38010	MM_QEC_MAX_POINTS	クォッドラント誤差補償用の値の数	/FB/, K3/ IAD/

## ■ アラーム

アラームの詳細説明については次を参照してください。

参照： アラーム一覧

あるいは MMC 101/102/103 を有するシステムのオンラインヘルプ



## 2.15 割出し軸 (T1)

### 2.15.1 概略説明

#### 工作機械上のインデックス軸

アプリケーションによっては、軸は決まった座標（たとえば、ロケーション番号）にアプローチすることだけが要求されます。

自動モードと設定モードの両方で、定義された座標（インデックスと呼びます）にアプローチする必要があります。そのような軸は "indexing axes"（インデックス軸）と呼ばれます。インデックス軸上で定義される位置は "coded positions"（コード化された位置）あるいは "index positions"（インデックス位置）と呼ばれます。

ソフトウェアバージョン 4.3 以降では、特殊ファンクションは、等間隔インデックス用に直線軸および回転軸で、および Hirth tooth システム用に使用できます。

#### アプリケーション

インデックス軸は主に、ツールタレット、ツールチェーンマガジン、ツールカートリッジマガジンなどの特定のタイプのツールマガジンと関連して使用されます。コード化された位置とは、マガジン中のツールの個々のロケーションのことをいいます。ツールチェンジの間に、マガジンはロード予定のツールが含まれるロケーションに位置決めされます。

### 2.15.2 詳細説明

#### ■ インデックス軸のトラバース

##### 一般事項

インデックス軸のトラバースは、設定モードタイプ JOG および INC で手動で行うか、"Coded positions"（コード化された位置）用の特殊なインストラクションを使用したパートプログラムから行うか、あるいは PLC によって行います。インデックス位置に到達すると、"indexing axis in position"（インデックス軸が定位置にある）インタフェース信号 (DB31-61, DBX76.6) が PLC に出力されます。

Hirth インデックス軸は、基準点がアプローチされるまで、JOG モードでトラバースできません。

#### ■ 手動 JOG モードでのインデックス軸のトラバース

##### 基準点アプローチ

インデックス軸は他の軸と同様の方法で基準点にアプローチします。基準点がインデックス位置にマッチする必要はありません。基準点に到達したとき (IS "Referenced/synchro-nized 1 or 2" (基準化／同期化 1 または 2) (DB31-61, DBX60.4 or 5) = "1") にのみ、インデックス軸は、通常トラバースおよび相対トラバースを使用して、JOG モードでインデックス位置にだけアプローチします。

例外：ハンドルを使用してトラバースするときは、インデックス位置はアプロー

チされません。

軸が基準化されない場合 (IS "Referenced/synchronized 1 or 2" = "0"), 軸が手動寸動モードでトラバースされると、インデックス位置が無視されます。

## JOG での連続トラバース

- 連続トリガモード (SD: JOG\_CONT\_MODE\_LEVELTRIGGRD = "1"):

"+" または "-" トラバースキーを押すと、インデックス軸が、通常の JOG トラバースを使用した場合と同じように移動します。トラバースキーを放すと、インデックス軸は、トラバース方向にある次のインデックス位置までトラバースします。

- 一時トリガモード (SD: JOG\_CONT\_MODE\_LEVELTRIGGRD = "0"):

トラバースキーを短く押すと (第 1 立上がり信号エッジ)、希望する方向に向かって、インデックス軸のトラバース動作がスタートします。トラバースキーを放すと、トラバースは継続されます。トラバースキーをもう一度押すと (第 2 立上がり信号エッジ)、インデックス軸はトラバース方向にある次のインデックス位置までトラバースします。

インデックス軸は通常、JOG モードでトラバースされます (標準設定)。連続モードはあまり重要ではありません。

インデックス位置に到達する前に、オペレータがトラバース方向を変更した場合、インデックス軸はトラバース方向にある次のインデックス位置に位置決めされます。トラバース動作は反対方向にスタートしなければなりません。

寸動モードまたは連続モードでの連続トラバースに関する詳細情報については、次を参照してください。

参照： 2.4 JOG 運転とハンドル (H1)

## JOG での相対トラバース (INC)

"+" または "-" トラバースキーを押すと、現在セットされているインクリメント値 (INC1; ... ,INCvar) に関係なく、インデックス軸は常に 1 インデックス位置単位で選択された方向にトラバースします。

寸動モードでは、トラバースキーを放すと、トラバース動作が中断されます。トラバースキーをもう一度押すと、インデックス位置にアプローチできます。

連続モードでは、トラバースキーをもう一度押すと、トラバース動作が強制終了します。この場合、インデックス軸はインデックス位置に位置決めされません。

## 2 つのインデックス位置の間

インデックス軸が 2 つのインデックス位置の間に位置決めされた場合、JOG-INC モードで、"+" トラバースキーを押すと、インデックス軸は次のより高いインデックス位置にアプローチします。同様に、 "-" トラバースキーを押すと、次のより低いインデックス位置がアプローチされます。

## ハンドルトラバース

JOG モードで、ハンドルを使用してインデックス軸がトラバースされると、インデックス位置が無視されます。ハンドルを回転させると、標準の測定単位に応じ

て、mm、インチあるいは度で表された任意の位置までインデックス軸がトラバースされます。

PLC ユーザプログラムは、インデックス軸をトラバースさせるためのハンドルをディスエーブルできます。

## PLC からの信号 "Indexing axis in position" (インデックス軸が定位置にある)

JOG モードでインデックス軸がトラバースされると、信号 "Indexing axis in position"(DB31-61, DBX76.6) が PLC インタフェースで出力され、インデックス位置に到達したことを示します。インデックス軸は基準化されていなければなりません (IS "Referenced/synchronized 1 or 2" = "1")。

## JOG モードでのアラーム

JOG モードでトラバース中に、インデックス軸が、インデックス位置テーブル (「**■インデックス軸のパラメータ化**」を参照) で定義されたトラバース範囲から外れた場合、アラーム 20054 "wrong index for indexing axis in JOG" (JOG でのインデックス軸のインデックスの誤り) が出力されます。

## 回転フィードレート

JOG モードでは、軸／スピンドルの挙動も同様に設定データ

JOG\_REV\_IS\_ACTIVE (JOG がアクティブなときの回転フィードレート) の設定に基づいています。

- この設定データがアクティブである場合、軸／スピンドルは常に、そのマスタスピンドルに基づいて、回転フィードレート MD JOG\_REV\_VELO (JOG の場合の回転フィードレート) あるいは MD JOG\_REV\_VELO\_RAPID (急速トラバースオーバーレイを使用した JOG の場合の回転フィードレート) を使用して移動します。
- 設定データがアクティブでない場合、軸／スピンドルの挙動は設定データ SSIGN\_FEED\_PER\_REV\_SOURCE (軸／スピンドルを位置決めするための回転フィードレート) によって決定されます。
- 設定データがアクティブでない場合、回転付きのフレームが有効なジオメトリ軸の挙動はチャンネル別設定データ JOG\_FEED\_PER\_REV\_SOURCE によって決まります (運転モード JOG では、回転付きのフレームが有効なジオメトリ軸用の回転フィードレート)。

## ■ AUTOMATIC モードでのインデックス軸のトラバース

### 選択された位置へのトラバース

AUTOMATIC モードでは、NC パートプログラムから、インデックス軸と定義された軸を選択された位置にアプローチさせることができます。これには定義されたインデックス位置間の位置が含まれます。これらの位置は、通常どおり、その軸についての測定単位 (mm / インチまたは度) でプログラムされます。この目的に使用される通常のプログラミングインストラクション (G90, G91, AC, IC など) についてはプログラミングガイドで説明しています。

## "Coded Positions"（コード化された位置）へのトラバース

パートプログラムに特殊インストラクションをプログラムすることもできます。  
指定された方法でトラバースするには、以下を実行します。

- CAC 絶対コード化位置へアプローチする
- CACP 正の方向に絶対コード化位置へアプローチする
- CACN 負の方向に絶対コード化位置へアプローチする
- CIC 相対コード化位置へアプローチする
- CDC 直接（最短）パスに沿ったコード化位置へアプローチする

絶対位置決めを使用すると、アプローチされるインデックス位置がプログラムされ、相対位置決めを使用すると、"+"または "-" 方向にトラバースされるインデックスの数がプログラムされます。

回転軸上では、インデックス位置へのアプローチは、最短パスに沿って直接 (CDC) 行るか、あるいは定義された回転方向 (CACP, CACN) を使用して行います。

インデックス軸用の特殊プログラミングインストラクションに関する詳細情報については、「**■インデックス軸のプログラミング**」を参照してください。

## "Indexing axis in position" インタフェース信号（ソフトウェアバージョン 4.3/2.3 まで）

"Coded positions"（コード化された位置）へアプローチするための特殊なインストラクション (CAC, CIC, など) が使用されている場合にかぎり、AUTOMATIC モードでインデックス軸がトラバースしているときに、インタフェース信号 "Indexing axis in position" がセットされます。その他の場合ではすべて、G90 または G91 を使用してアプローチされた位置がインデックス位置と正確にマッチしたとしても、インタフェース信号は "0" にセットされたままになります。

ソフトウェアバージョン 4.3 以降：

"Exact stop fine"（正確停止微）ウインドウに到達してから、インデックス軸がインデックス位置上に位置決めされた場合は、インデックス位置に到達した方法にかかわらず、この信号はイネーブルされます。

## ■ PLC によるインデックス軸のトラバース

### PLC によるトラバース

PLC ユーザプログラムによってインデックス軸をトラバースすることもできます。  
次のような方法があります。

- 同時位置決め軸を使用する方法  
この場合、アプローチされるインデックス位置を PLC によって指定できます。  
参照： 2.10 補助位置決め軸 (P2)
- 非同期サブプログラム (ASUP) を使用する方法  
参照： 1.10 運転モードの種類とプログラム運転モード (K1)

## ■ インデックス軸のパラメータ化

### インデックス軸の定義

軸別マシンデータ INDEX\_AX\_ASSIGN\_POS\_TAB を使用することにより、ある軸（直線軸あるいは回転軸）をインデックス軸と定義することができます。インデックス位置テーブルの番号（1 または 2）がこのマシンデータに入力されなければなりません。

すべての軸が同じタイプの軸である（直線軸，回転軸，モジュロ 360° ファンクション）という条件で，複数の軸を 1 つのインデックス位置テーブルに割り当てることができます。そうでない場合は，アラーム 4080 "Incorrect configuration of indexing axis in MD [name]"（MD のインデックス軸構成が正しくない）が電源オン中に出力されます。

### インデックス位置テーブル

インデックスに割り当てられた（mm または度で表される）軸位置は，各インデックス軸について，テーブルの形でマシンデータに保存されなければなりません。個々のインデックス位置間の距離について任意の値を入力できます。インデックス位置を入力する際には，次のことに注意してください。

### テーブルの数

最大 2 つのインデックス位置テーブルが許可されています。

MD 10910: INDEX\_AX\_POS\_TAB\_1 [n]

MD 10930: INDEX\_AX\_POS\_TAB\_2 [n]

### 各テーブルに対する入力数

最大 60 個の位置を各インデックス位置テーブルに入力できます [n = 0 ... 59]。

テーブル 1 および／または 2 について，使用される実際の入力数が次のマシンデータで定義されなければなりません。

MD 10900: INDEX\_AX\_LENGTH\_POS\_TAB\_1 bzw.

MD 10920: INDEX\_AX\_LENGTH\_POS\_TAB\_2

テーブルに入力された位置で，上のマシンデータで定義された数よりも大きいものはすべて非アクティブになります。

### インチ／メートルの切換え

ソフトウェアバージョン 5 以前では，回転軸インデックス位置をインチマシンに入力すると (MD 10240: SCALING\_SYSTEM\_IS\_METRIC=0)，その値は，度で入力されたにもかかわらず，25.4 で除算される必要があります。

ソフトウェアバージョン 5 以降で，MD 10260: CONVERT\_SCALING\_SYSTEM=1 (/G2/ を参照) の場合，インデックス位置はセットされた基本系ではなく，MD 10270: POS\_TAB\_SCALING\_SYSTEM でセットされた測定系のことをいいます。

MD 10270: POS\_TAB\_SCALING\_SYSTEM=0: メートル

MD 10270: POS\_TAB\_SCALING\_SYSTEM=1: インチ

(注)

MD 10270 は、次のマシンデータについて位置指定用の測定系を定義します。

MD 10900: INDEX\_AX\_POS\_TAB\_1

MD 10920: INDEX\_AX\_POS\_TAB\_2

MD 10270 は SD 41500 ～ SD 41507 にも影響を及ぼします (/N3/ を参照)。

## 入力フォーマット

- ・インデックス位置は、昇順で（負のトラバース範囲からスタートして、正のトラバース範囲へ）、入力と入力の間隔を空けずに、テーブルへ入力されるべきです。連続した位置値が同一であることはできません。
- ・軸位置は標準座標系に入力されるべきです。

インデックス軸が、モジュロ 360° を使用して回転軸と定義された場合 (MD: IS\_ROT\_AX="1" および MD: ROT\_IS\_MODULO="1"), インデックス位置に関して次の点も同様に監視されるべきです。

- ・インデックス位置は、 $0^\circ \leq \text{pos} < 360^\circ$  の範囲にプログラムできます。位置がこの範囲を超えると、電源オン時にアラーム 4080 が生成されます。
- ・インデックス軸は連続的に回転する回転軸と定義されます。INC で正の方向に回転させる場合は、テーブル中の最高有効インデックス位置に到達してから、インデックス位置 1 がアプローチされ、その後、インデックス軸は正の方向へトラバースを続けます。同様に、INC で負の方向に回転させる場合は、インデックス位置 1 の次には最高有効インデックス位置が続きます。

## ■ インデックス軸のプログラミング

### 注記

インデックス軸のプログラミングに関するガイドラインについては次を参照してください。

参照： YS840DI ユーザーズマニュアル・プログラミング編

### コード化された位置

NC パートプログラムからインデックス軸を位置決めできるように、特殊なインストラクション（いわゆるコード化された位置）が提供されています。このインストラクションを使用すると、軸位置が mm や度で表されるのではなく、インデックス番号（たとえば、ロケーション番号）がプログラムされます。インデックス軸が直線軸と定義されているか、回転軸と定義されているかによって、可能なコード化位置インストラクションは次のようになります。

インデックス軸が直線軸の場合：CAC(i), CIC(i)

インデックス軸が回転軸の場合：CAC(i), CIC(i), CACP(i), CACN(i), CDC(i)

i = コード化された位置（インデックス位置）

i の値の範囲：0 ～ 59；整数；例外は CIC が正の場合のみ。

### 絶対位置

**POS[B]=CAC(20)**

インデックス軸 B は、絶対モードで、コード化された位置（インデックス）20 にアプローチします。トラバース方向は、現在の実際位置によって決まります。

### 正方向に絶対位置

**POS[B]=CACP(10)**

インデックス軸 B は、絶対モードで、正の回転方向へ、コード化された位置（インデックス）10 にアプローチします（回転軸の場合にのみ可能）。

### 負方向に絶対位置

**POS[B]=CACN(10)**

インデックス軸 B は、絶対モードで、負の回転方向へ、コード化された位置（インデックス）10 にアプローチします（回転軸の場合にのみ可能）。

### 直接絶対位置

**POS[B]=CDC(50)**

インデックス軸 B は、最短パスに沿って直接 インデックス位置 50 にアプローチします（回転軸の場合にのみ可能）。

### 相対位置

**POS[B]=CIC(-4)**

インデックス軸 B は、現在の位置から負の方向へ 4 インデックス位置単位でトラバースします。

**POS[B]=CIC(35)**

インデックス軸 B は、現在のインデックス位置から正の方向へ 35 インデックス位置単位でトラバースします。

リーディング符号がアプローチ方向を定義します。

(注)

モジュロ回転軸上では、インデックス位置は 360 度で割算され、直接アプローチされます。

### インデックス位置の間

自動モードで、インデックス軸が 2 つのインデックス位置の間に位置決めされた場合、プログラムコマンド **POS[B]=CIC(1)** を使用すると、その次のより高いインデックス位置がアプローチされます。同様に、プログラムインストラクション **POS[B]=CIC(-1)** を使用すると、その次のより低いインデックス位置がアプローチされます。**POS[B]=CIC(0)** を使用すると、インデックス軸はトラバースしません。

## アラーム

インデックス位置がインデックス位置テーブルの有効範囲外にプログラムされた場合、アラーム 17510 "Impermissible index for indexing axis"（インデックス軸に許可されていないインデックス）が出力されます。インデックス位置がある軸についてプログラムされたとき、この軸にインデックス位置テーブルが割当てられていない場合（MD: INDEX\_AX\_ASSIGN\_POS\_TAB（軸はインデックス軸である））は、アラーム 17500 "Axis is not an indexing axis"（軸はインデックス軸ではない）が出力されます。

## FRAMES（フレーム）

制御装置はインデックス位置テーブルに保存された位置を mm，インチ，または度でプログラムされた位置と解釈するので、インデックス軸で FRAMES はディスエーブルされません。

アプリケーションによっては、インデックス軸には通常 FRAMES は必要ありません。したがって、多くの場合では、インデックス軸用のパートプログラムで FRAMES とゼロオフセットを抑止することが望ましいといえます。

### ■ 拡張機能

#### 一般情報

拡張には次のことが含まれます。

- 任意の数の等間隔インデックスの間隔
- インデックス軸用の MD 起動変更

### ■ 等間隔インデックスの間隔

等間隔インデックスの間隔は、次で使用できます。

- 直線軸
- モジュール回転軸
- 回転軸

## インデックス間の距離

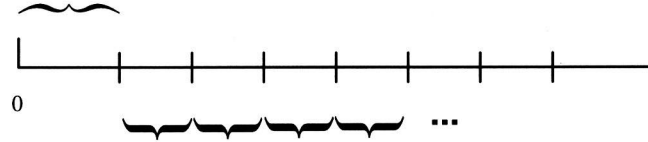
インデックス距離は、等間隔インデックスの間隔について、次のように決定されます。

$$\text{距離} = \frac{\text{分子 (MD 30501: \$MA_INDEX_AX_NUMERATOR)}}{\text{分母 (MD 30502: \$MA_INDEX_AX_DENOMINATOR)}}$$



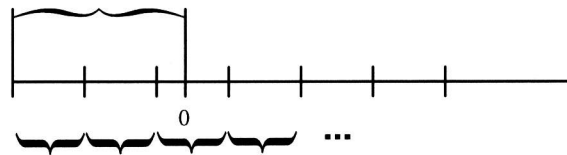
## 直線軸

MD 30503: \$MA\_INDEX\_AX\_OFFSET (ゼロ点からの第 1 距離, 正)



距離 (すべてのインデックスで同じ, 上を参照)

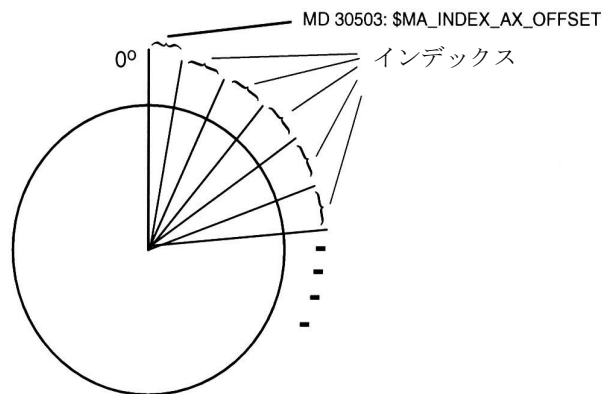
MD 30503: \$MA\_INDEX\_AX\_OFFSET (ゼロ点からの第 1 距離, 負)



距離 (すべてのインデックスで同じ, 上を参照)

## モジュロ回転軸

$$\text{インデックス} = \frac{\text{分子 (MD 30330: \$MA_MODULO_RANGE)}}{\text{分母 (MD 30502: \$MA_INDEX_AX_DENOMINATOR)}}$$



$$\text{インデックス} = \frac{\text{モジュロ範囲のサイズ (MD 30330: \$MA_MODULO_RANGE)}}{\text{モジュロ範囲中のインデックスの数 (MD 30502: \$MA_INDEX_AX_DENOMINATOR)}}$$

---

## 起動

直線軸および回転軸またはモジュロ回転軸用の等間隔インデックスを使用したファンクションは、MD 30500: \$MA\_INDEX\_AX\_ASSIGN\_POS\_TAB[axis] で "table number" (テーブル番号) 3 を指定することによって起動します。

## Hirth tooth システム

### 概要

Hirth tooth システムを使用すると、回転軸上の回転位置は通常、直線軸を介して、ラッチなどの歯状ホイールを使用してインタロックされます。インタロックは、インデックス位置に正確に到達したときにのみ起動する必要があります。インデックス位置間の距離は、全周にわたって同じ（等間隔）です。

### 前提条件

回転軸がインデックス軸でなければなりません。この軸は基準化されなければなりません。次を参照してください。

参照： 1.16 原点復帰 (R1)

## 起動

MD 30505: \$MA\_HIRTH\_IS\_ACTIVE が 1 にセットされなければなりません。

MD 30500: \$MA\_INDEX\_AX\_ASSIGN\_POS\_TAB が 3 にセットされなければなりません（等間隔インデックス）。

## 影響

- 回転軸だけがすべてのモードおよびオペレーティングステータスで、インデックス位置にアプローチできます。
- JOG モードでは、軸を次のようにトラバースできます。

通常どおりに、あるいは

相対的に

前提条件：軸が基準化されていること

- ハンドルを使用した寸動は不可能です。次を参照してください。

参照： /H1/, Jogging with the Handwheel

- AUTO, MDA では、あるいは ASUP を使用した場合は、"coded positions"（コード化された位置）にしかアプローチできません。
- PLC は軸をインデックス位置にしか移動させることができません。他の位置へアプローチしようとする、アラームが出力されます。

## 特定の状況での Hirth 軸の応答

### STOP/RESET（停止／リセット）

トラバース動作中に NC STOP および RESET すると、このコマンドが起動する前に、次のインデックス位置がアプローチされます。

---

## EMERGENCY STOP（非常停止）

EMERGENCY STOP 後、PLC あるいはオペレータが JOG モードでインデックス軸をインデックス位置まで戻してから、縦軸が内側または下側に移動されなければなりません。

## オーバーライド = 0 または "Stop axis"（停止軸）信号

これらのイベントが発生したときに、軸が前のインデックス位置からすでに移動している場合、この応答が開始される前に、制御装置は軸を次に可能なインデックス位置まで移動させます。

## 移動距離削除

次の可能なインデックス位置へトラバースした後、この位置で動作が強制終了します。

## コマンド軸

次を参照してください。

参照： /FBSY/, Synchronized Actions

移動中のコマンド軸に対して MOV = 0 が指定された場合、この軸は次に可能なインデックス位置までトラバースを続けてから、停止します。

## 制限事項

### 変換

Hirth tooth システムが定義された軸は、キネマティックス変換に関与できません。

## PRESET（プリセット）

Hirth tooth システムが定義された軸を、PRESET を使用して新しい値にセットすることはできません。

## 回転フィードレート

Hirth tooth システムが定義された軸を、回転フィードレートを使用してトラバースすることはできません。

## パス／速度オーバーレイ

Hirth tooth システムが定義された軸を、パスあるいは速度オーバーレイと組合わせて使用することはできません。

## フレーム、外部 ZO、DRF

Hirth tooth システムが定義された軸は、フレームあるいは外部ゼロオフセット、DRF などの補間補償をサポートしません。

## カップリング

Hirth tooth システムが定義された軸は、次の軸にはなりません。

- マスタ値カップリングを使用した従属軸
- カップリングされたモーション軸
- ガントリ従属軸

### ■ マシンデータ起動変更

ソフトウェアバージョン 4.3 以降では、下記の MD は、新しい値を MD に割当ててから、RESET するだけで起動します (POWER ON が予め必要となります)。

MD 10900: \$MN\_INDEX\_AX\_LENGTH\_POS\_TAB\_1

MD 10920: \$MN\_INDEX\_AX\_LENGTH\_POS\_TAB\_2

MD 10910: \$MN\_INDEX\_AX\_POS\_TAB\_1

MD 10930: \$MN\_INDEX\_AX\_POS\_TAB\_2

MD 30500: \$MA\_INDEX\_AX\_ASSIGN\_POS\_TAB

インデックス軸用の MD の完全なリストについては、2.15.4 「データの説明 (MD, SD)」を参照してください。

### ■ 等間隔インデックスの例

#### モジュロ回転軸

\$MA\_\_INDEX\_AX\_DENOMINATOR[AX4]=18

\$MA\_INDEX\_AX\_OFFSET[AX4]=5

\$MA\_INDEX\_AX\_ASSIGN\_POS\_TAB[AX4]=3

\$MA\_IS\_ROT\_AX[AX4]=TRUE

\$MA\_ROT\_IS\_MODULO[AX4]=TRUE

上記のマシンデータを使用すると、軸 4 がモジュロ回転軸およびインデックス軸と定義され、等間隔位置が 5 度からスタートして 20 度置きになるように定義されます。インデックス位置は次のようになります : 5, 25, 45, 65, 85, 105, 125, 145, 165, 185, 205, 225, 245, 265, 285, 305, 325, 345 度。

(注)

\$MA\_MODULO\_RANGE 用のデフォルトは 360 であるので、

\$MA\_\_INDEX\_AX\_DENOMINATOR[AX4]=18 割当てにより 20° 分割が生成されます。

#### 回転軸

\$MA\_INDEX\_AX\_\_NUMERATOR[AX4]=360

\$MA\_\_INDEX\_AX\_DENOMINATOR[AX4]=18

\$MA\_INDEX\_AX\_OFFSET[AX4]=100

\$MA\_INDEX\_AX\_ASSIGN\_POS\_TAB[AX4]=3

\$MA\_IS\_ROT\_AX[AX4]=TRUE

\$MA\_POS\_LIMIT\_MINUS[AX1]=100

\$MA\_POS\_LIMIT\_PLUS[AX1]=260

上記のマシンデータを使用すると、軸 4 が回転軸およびインデックス軸と定義され、等間隔位置が 100 度からスタートして 20 度置きになるように定義されます。インデックス位置は次のようになります: 100, 120, 140 度など。100 度よりも小さい位置をインデックス位置としてアプローチすることはできません。したがって、この場合は、より低いソフトウェアリミットスイッチを置くのが望ましいといえます。インデックス位置は、ソフトウェアリミットスイッチに到達するまで（この場合は、260 度）継続します。この結果、回転軸は 100 ～ 260 度の間しかトラバースできません。

### 直線軸

```
$MA_INDEX_AX__NUMERATOR[AX1] = 10  
$MA__INDEX_AX_DENOMINATOR[AX1] = 1  
$MA_INDEX_AX__OFFSET[AX1] = -200  
$MA_INDEX_AX_ASSIGN_POS_TAB[AX1] = 3  
$MA_IS_ROT_AX[AX1] = FALSE  
$MA_POS_LIMIT_MINUS[AX1] = -200  
$MA_POS_LIMIT_PLUS[AX1] = 200
```

上記のマシンデータを使用すると、軸 4 が直線軸およびインデックス軸と定義され、等間隔位置が -200 mm からスタートして 10 mm 置きになるように定義されます。インデックス位置は次のようになります: -200, -190, -180 mm など。これらのインデックス位置は、ソフトウェアリミットスイッチが到達されるまで（この場合は、200 mm）、継続します。

### Hirth tooth システム

```
$MA__INDEX_AX_DENOMINATOR[AX4] = 360  
$MA_INDEX_AX__OFFSET[AX4] = 0  
$MA_INDEX_AX_ASSIGN_POS_TAB[AX4] = 3  
$MA_IS_ROT_AX[AX4] = TRUE  
$MA_ROT_IS_MODULO[AX5] = TRUE  
$MA_HIRTH_IS_ACTIVE[AX4] = TRUE
```

上記のマシンデータを使用すると、軸 4 はモジュロ回転軸およびインデックス軸と定義され、Hirth tooth システムおよび等間隔位置が 0 度からスタートして 1 度置きになるように定義されます。

## ■ インデックス軸のスタートアップ

### 手順

インデックス軸をスタートアップする手順は、通常の NC 軸（直線軸および回転軸）と同じです。

### 回転軸

モジュロ 360 変換 (MD: ROT\_IS\_MODULO = "1") を使用してインデックス軸が回転軸と定義された場合 (MD: IS\_ROT\_AX = "1"), そのインデックス位置はモジュロ

360 でトラバースされます。したがって、0 ～ 359.999 の範囲内にある位置しかインデックス位置テーブルに入力できません。そうでない場合は、アラーム 4080 "Incorrect configuration for indexing axis in MD [Name]" (MD のインデックス軸構成が正しくない [Name]) が出力されます。

MD: DISPLAY\_IS\_MODULO = "1" を使用して、位置表示をモジュロ 360° にセットできます。

## 特殊マシンデータ

2.15.4 「データの説明 (MD, SD)」で説明される、次のマシンデータも同様に定義される必要があります。

## 一般マシンデータ

MD: INDEX_AX_LENGTH_POS_TAB_1	テーブル 1 で使用されるインデックス位置の数
MD: INDEX_AX_LENGTH_POS_TAB_2	テーブル 2 で使用されるインデックス位置の数
MD: INDEX_AX_POS_TAB_1 [n]	インデックス位置テーブル 1
MD: INDEX_AX_POS_TAB_2 [n]	インデックス位置テーブル 2

## 軸別マシンデータ

MD: INDEX_AX_ASSIGN_POS_TAB	軸はインデックス軸である (インデックス位置テーブル 1 または 2, あるいは等間隔インデックス用の 3 の割当て)
-----------------------------	---

ソフトウェアバージョン 4.3 以降 :

MD: HIRTH_IS_ACTIVE	軸は "Hirth tooth system" (Hirth tooth システム) の特性を持つ
MD: INDEX_AX_NUMERATOR	等間隔インデックス用の分子
MD: INDEX_AX_DENOMINATOR	等間隔インデックス用の分母
MD: INDEX_AX_OFFSET	ゼロからの第 1 インデックス位置の距離

## マシンデータの例

上記のマシンデータの割当てについては、以下に、2 つの例を用いて説明します。

## 回転軸としてのインデックス軸の例

8 つのタレットロケーションがあるツールタレット

円形マガジンは、連続して回転する回転軸と定義されます。8 つのタレットロケーション間の距離は一定であって、第 1 ロケーションは位置 0° になります (図 2.108 を参照)。

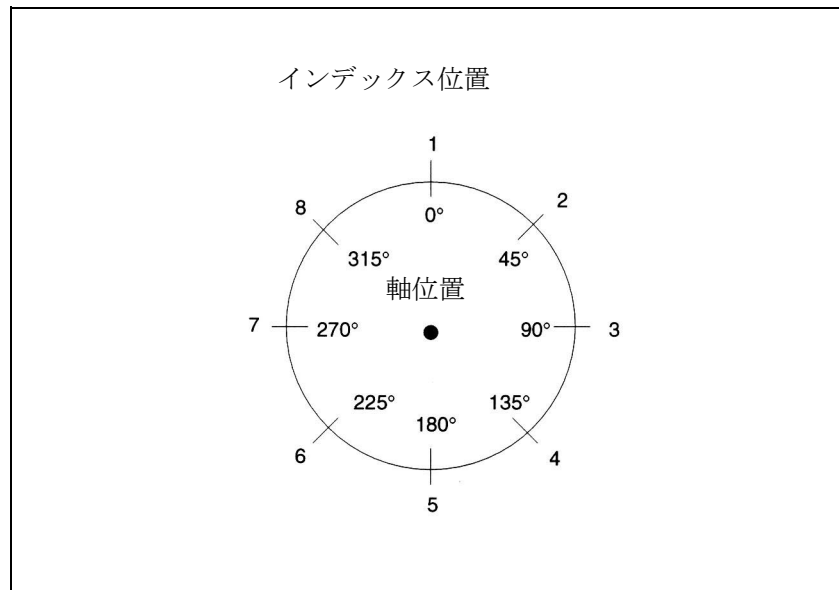


図 2.108 例：8つのロケーションがあるツールタレット

## インデックス位置テーブル

円形マガジン用のインデックス位置はテーブル 1 に入力されます。

$\$MN\_INDEX\_AX\_POS\_TAB\_1[0] = 0$  ; 第 1 インデックス位置は 0 度  
 $\$MN\_INDEX\_AX\_POS\_TAB\_1[1] = 45$  ; 第 2 インデックス位置は 45 度  
 $\$MN\_INDEX\_AX\_POS\_TAB\_1[2] = 90$  ; 第 3 インデックス位置は 90 度  
 $\$MN\_INDEX\_AX\_POS\_TAB\_1[3] = 135$  ; 第 4 インデックス位置は 135 度  
 $\$MN\_INDEX\_AX\_POS\_TAB\_1[4] = 180$  ; 第 5 インデックス位置は 180 度  
 $\$MN\_INDEX\_AX\_POS\_TAB\_1[5] = 225$  ; 第 6 インデックス位置は 225 度  
 $\$MN\_INDEX\_AX\_POS\_TAB\_1[6] = 270$  ; 第 7 インデックス位置は 270 度  
 $\$MN\_INDEX\_AX\_POS\_TAB\_1[7] = 315$  ; 第 8 インデックス位置は 315 度

## その他のマシンデータ

$\$MN\_INDEX\_AX\_LENGTH\_POS\_TAB\_1 = 8$  ; 8 つのインデックス位置が テーブル 1 にある  
 $\$MA\_INDEX\_AX\_ASSIGN\_POS\_TAB [AX5] = 1$  ; 軸 5 はインデックス軸と定義され、インデックス位置はテーブル 1 にある  
 $\$MA\_IS\_ROT\_AX [AX5] = 1$  ; 軸 5 は回転軸である  
 $\$MA\_ROT\_IS\_MODULO [AX5] = 1$  ; モジュロ変換がアクティブである

## 直線軸としてのインデックス軸の例

10 個のロケーションがあるパレット（図 2.109 を参照）。

10 個のロケーション間の距離は変化します。第 1 パレットロケーションは -100 mm になります。

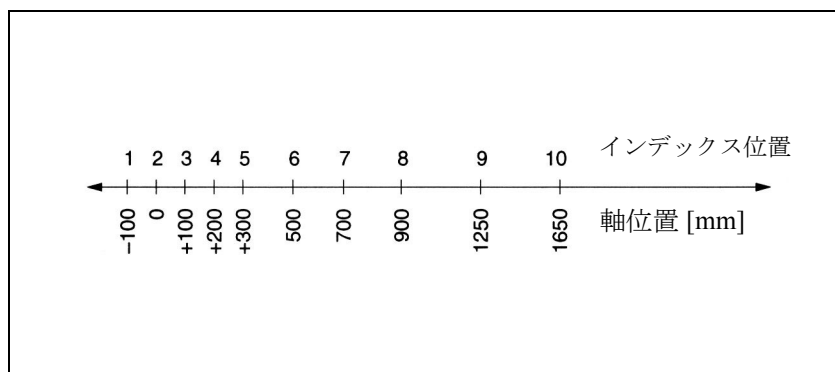


図 2.109 例：インデックス軸としてのツールパレット

## インデックス位置テーブル

円形マガジン用のインデックス位置はテーブル 2 に入力されます。

\$MN\_INDEX\_AX\_POS\_TAB\_2[0] = -100 ; 第 1 インデックス位置は -100 度

\$MN\_INDEX\_AX\_POS\_TAB\_2[1] = 0 ; 第 2 インデックス位置は 0 度

\$MN\_INDEX\_AX\_POS\_TAB\_2[2] = 100 ; 第 3 インデックス位置は 100 度

\$MN\_INDEX\_AX\_POS\_TAB\_2[3] = 200 ; 第 4 インデックス位置は 200 度

\$MN\_INDEX\_AX\_POS\_TAB\_2[4] = 300 ; 第 5 インデックス位置は 300 度

\$MN\_INDEX\_AX\_POS\_TAB\_2[5] = 500 ; 第 6 インデックス位置は 500 度

\$MN\_INDEX\_AX\_POS\_TAB\_2[6] = 700 ; 第 7 インデックス位置は 700 度

\$MN\_INDEX\_AX\_POS\_TAB\_2[7] = 900 ; 第 8 インデックス位置は 900 度

\$MN\_INDEX\_AX\_POS\_TAB\_2[8] = 1250 ; 第 9 インデックス位置は 1250 度

\$MN\_INDEX\_AX\_POS\_TAB\_2[9] = 1650 ; 第 10 インデックス位置は 1650 度

## その他のマシンデータ

\$MN\_INDEX\_AX\_LENGTH\_POS\_TAB\_2=10 ; 10 個のインデックス位置がテーブル 2 にある

\$MA\_INDEX\_AX\_ASSIGN\_POS\_TAB[AX6]=2 ; 軸 6 がインデックス軸と定義される

; インデックス位置はテーブル 2 にある



## ■ インデックス軸の特殊な機能

### DRF

DRF ファンクションを使用すると、ハンドルを用いた AUTOMATIC モードでインデックス軸用に追加の相対ゼロオフセットを生成することができます。

### ソフトウェアリミットスイッチ

インデックス軸が基準化された後に、軸がトラバースされると、ソフトウェアリミットスイッチがアクティブになります。

連続 JOG モードあるいは相対 JOG モードで手動トラバースを行うと、ソフトウェアリミットスイッチの前の、最後のインデックス位置でインデックス軸が停止します。

### 基準点アプローチ

インデックス軸は、基準点に到達してからしか (IS "Referenced/syn-chronized 1 or 2" (基準化/同期化 1 または 2) (DB31-48, DBX60.4 or 5) = "1"), JOG モード (連続あるいは相対) でインデックス位置にアプローチしません。軸が基準化されていない場合 ("referenced/synchronized 1 or 2" インタフェース信号 = "0"), 手動でトラバースすると、インデックス位置は無視されます。

インデックス位置テーブルに保存されている軸位置は、軸が基準化されたときにしかマシン位置に対応しないので、インデックス軸が基準化されるまで NC スタートはディスエーブルされなければなりません。

### 位置表示

インデックス軸上の位置は、軸について通常使用される測定単位 (mm, インチまたは度) で表示されます。

### リセットによる強制終了

リセットを行うと、インデックス軸上でのトラバース動作が強制終了され、インデックス軸が停止します。インデックス軸はもはやインデックス位置上に位置決めされません。

(注)

Hirth tooth システムの応答については「■等間隔インデックスの間隔」で説明しています。

## 2.15.3 補足条件

本ファンクション説明用に規定された補足条件はありません。

## 2.15.4 データの説明 (MD, SD)

### ■ 一般マシンデータ

10270 MD 番号	POS_TAB_SCALING_SYSTEM 位置テーブルの測定系	
初期値 : 0	最小入力リミット : 0	最大入力リミット : 1
変更が有効になるための条件 : RESET	保護レベル : 2 / 7	単位 : -
データタイプ : BYTE	適用開始ソフトウェアバージョン : 5	
意味 :	<p>このマシンデータは、インデックス軸テーブルの位置指定用の測定系をセットし、ソフトウェアカム用の点を切換えるためのものです。</p> <ul style="list-style-type: none"> <li>• MD 10270=0: メートル</li> <li>• MD 10270=1: インチ</li> </ul> <p>MD 10270 は、次のマシンデータについて位置指定用の測定系を定義します。</p> <p>MD 10900: INDEX_AX_POS_TAB_1 MD 10920: INDEX_AX_POS_TAB_2 SD 41500: SW_CAM_MINUS_POS_TAB_1 SD 41501: SW_CAM_PLUS_POS_TAB_1 SD 41502: SW_CAM_MINUS_POS_TAB_2 SD 41503: SW_CAM_PLUS_POS_TAB_2 SD 41504: SW_CAM_MINUS_POS_TAB_3 SD 41505: SW_CAM_PLUS_POS_TAB_3 SD 41506: SW_CAM_MINUS_POS_TAB_4 SD 41507: SW_CAM_PLUS_POS_TAB_4</p> <p>注記 :</p> <p>MD 10260: CONVERT_SCALING_SYSTEM=1 のときにのみ有効 (1.8「位置指令単位と検出単位 (G2)」を参照)。</p>	
関連性	<p>次のタイトルのマシンデータおよび設定データを参照</p> <p>MD 10260: CONVERT_SCALING_SYSTEM</p>	

10900 MD 番号	INDEX_AX_LENGTH_POS_TAB_1 テーブル 1 で使用されるインデックス位置の数	
初期値 : 0	最小入力リミット : 0	最大入力リミット : 60
変更が有効になるための条件 : 電源オン, または SW 4.3 以降ではリセット	保護レベル : 2 / 7	単位 : -
データタイプ : DWORD	適用開始ソフトウェアバージョン : 1.1	

10900 MD 番号	INDEX_AX_LENGTH_POS_TAB_1 テーブル 1 で使用されるインデックス位置の数
意味:	<p>インデックス位置テーブルは、有効な測定単位 (mm, インチまたは度) で表された軸位置をインデックス軸上のインデックス位置 [n] に割当ての際に使用されます。</p> <p>テーブル 1 で使用されるインデックス位置の数は、MD: INDEX_AX_LENGTH_POS_TAB_1 によって定義されます。</p> <p>これらのインデックス位置にはテーブル 1 で有効な値が含まれなければなりません。テーブル中のインデックス位置で、マシンデータで指定された数より大きいものはすべて無視されます。</p> <p>最大 60 個のインデックス位置 (0 ~ 59) をテーブルに入力することができます。</p> <p>テーブルの長さ = 0 はテーブルが評価されていないことを意味します。この長さが 0 でない場合、MD: INDEX_AX_ASSIGN_POS_TAB を使用して、テーブルを軸に割当てなければなりません。</p> <p>モジュロ 360 (MD: ROT_IS_MODULO = "1") を使用してインデックス軸が回転軸と定義された場合 (MD: IS_ROT_AX = "1"), マシンデータは最後のインデックス位置を定義します (正方向回転の場合、その後にインデックス位置が再度 1 からトラバースを再開する)。</p>
アプリケーション	ツールマガジン (ツールタレット, チェーンマガジン)
例外, エラー	MD: INDEX_AX_LENGTH_POS_TAB_1 に 60 以上の値が入力された場合, アラーム 17090 "Value violates upper limit" (値が上限を超えている) が出力されます。
関連性	MD: INDEX_AX_ASSIGN_POS_TAB (軸はインデックス軸である) MD: INDEX_AX_POS_TAB_1 (インデックス位置テーブル 1) MD: IS_ROT_AX (回転軸) MD: ROT_IS_MODULO (回転軸用のモジュロ変換)

10920 MD 番号	INDEX_AX_LENGTH_POS_TAB_2 テーブル 2 で使用されるインデックス位置の数	
初期値 : 0	最小入力リミット : 0	最大入力リミット : 60
変更が有効になるための条件 : 電源オン, または SW 4.3 以降で はリセット	保護レベル : 2 / 7	単位 : -
データタイプ :DWORD	適用開始ソフトウェアバージョン : 1.1	
意味 :	インデックス位置テーブルは、有効な測定単位 (mm, インチまたは度) で表 された軸位置をインデックス軸上のインデックス位置 [n] に割当てるときに使用 されます。 テーブル 2 で使用されるインデックス位置の数は、MD: INDEX_AX_LENGTH_POS_TAB_2 によって定義されます。 これらのインデックス位置にはテーブル 2 で有効な値が含まれなければなり ません。テーブル中のインデックス位置で、マシンデータで指定された数よ り大きいものはすべて無視されます。 最大 60 個のインデックス位置 (0 ~ 59) をテーブルに入力することができま す。 テーブルの長さ = 0 はテーブルが評価されていないことを意味します。この長 さが 0 でない場合、MD: INDEX_AX_ASSIGN_POS_TAB を使用して、テーブ ルを軸に割当てなければなりません。  モジュロ 360 (MD: ROT_IS_MODULO = "1") を使用してインデックス軸が回転 軸と定義された場合 (MD: IS_ROT_AX = "1"), マシンデータは最後のインデッ クス位置を定義します (正方向回転の場合、その後にインデックス位置が再 度 1 からトラバースを再開する)。	
この MD は次の場合は無意味	ツールマガジン (ツールタレット, チェーンマガジン)	

10920 MD 番号	INDEX_AX_LENGTH_POS_TAB_2 テーブル 2 で使用されるインデックス位置の数
例外, エラー	MD: INDEX_AX_LENGTH_POS_TAB_2 に 60 以上の値が入力された場合, アラーム 17090 "Value violates upper limit" (値が上限を超えている) が出力されます。
関連性	MD: INDEX_AX_ASSIGN_POS_TAB (軸はインデックス軸である) MD: INDEX_AX_POS_TAB_2 (インデックス位置テーブル 2) MD: IS_ROT_AX (回転軸) MD: ROT_IS_MODULO (回転軸用のモジュロ変換)

10910 MD 番号	INDEX_AX_POS_TAB_1[n] インデックス位置テーブル 1 [n]	
初期値 : 0	最小入力リミット : ***	最大入力リミット : ***
変更が有効になるための条件 : 電源オン, または SW 4.3 以降ではリセット	保護レベル : 2 / 7	単位 : mm または度
データタイプ : DOUBLE	適用開始ソフトウェアバージョン : 1.1	
意味 :	<p>インデックス位置テーブルは, 有効な測定単位 (mm, インチまたは度) で表された軸位置をインデックス軸上のインデックス位置 [n] に割当ての際に使用されます。</p> <p>[n] = インデックス位置テーブルにインデックス位置を入力するためのインデックス。</p> <p>範囲 : 0 _ n _ 59 (0 は第 1 インデックス位置であり, 59 は第 60 インデックス位置に対応する)。</p> <p>注記 : 絶対インデックス位置を使用したプログラミング (たとえば, CAC) は, インデックス位置 1 からスタートします。これは, テーブルでインデックス n = 0 であるインデックス位置に対応します。</p> <p>インデックス位置を入力する際には, 次の点に注意すべきです。</p> <ul style="list-style-type: none"> <li>最大 60 個のインデックス位置をテーブルに保存することができる。</li> <li>テーブルへの第 1 入力 はインデックス位置 1 に対応する ; 第 n 入力はインデックス位置 n に対応する。</li> <li>インデックス位置は, 昇順で (負のトラバース範囲から正のトラバース範囲へ), 入力と入力の間を空けずにテーブルに入力されるべきである。連続的した位置値を同一にはできない。</li> <li>モジュロ 360 (MD: ROT_IS_MODULO = "1") を使用してインデックス軸が回転軸と定義された場合 (MD: IS_ROT_AX = "1"), その位置値は, 0 pos. &lt; 360 の範囲に制限される。</li> </ul> <p>テーブルで使用されるインデックス位置の数は, MD: INDEX_AX_LENGTH_POS_TAB_1 によって定義されます。</p> <p>MD: INDEX_AX_ASSIGN_POS_TAB に値 1 を入力すると, インデックス位置テーブル 1 が現在の軸に割当てられます。</p>	
アプリケーション	ツールマガジン (ツールタレット, チェーンマガジン)	
例外, エラー	テーブルに 60 個以上の位置が入力された場合, アラーム 17020 "Illegal array index" (不正な配列インデックス) が出力されます。	
関連性	MD: INDEX_AX_ASSIGN_POS_TAB (軸はインデックス軸である) MD: INDEX_AX_LENGTH_POS_TAB_1 (テーブル 1 で使用されるインデックス位置の数) MD: IS_ROT_AX (回転軸) MD: ROT_IS_MODULO (回転軸用のモジュロ変換)	

10930 MD 番号	INDEX_AX_POS_TAB_2[n] インデックス位置テーブル 2 [n]	
初期値 : 0	最小入力リミット : ***	最大入力リミット : ***

10930 MD 番号	INDEX_AX_POS_TAB_2[n] インデックス位置テーブル 2 [n]	
変更が有効になるための条件： 電源オンまたは SW 4.3 以降では リセット	保護レベル：2 / 7	単位：mm または度
データタイプ：DOUBLE	適用開始ソフトウェアバージョン：1.1	
意味：	<p>インデックス位置テーブルは、有効な測定単位（mm、インチまたは度）で表された軸位置をインデックス軸上のインデックス位置 [n] に割当てるときに使用されます。</p> <p>[n] = インデックス位置テーブルにインデックス位置を入力するためのインデックス。</p> <p>範囲：0_n_59（0 は第 1 インデックス位置であり、59 は第 60 インデックス位置に対応する）。</p> <p>注記：絶対インデックス位置を使用したプログラミング（たとえば、CAC）は、インデックス位置 1 からスタートします。これは、テーブルでインデックス n = 0 であるインデックス位置に対応します。</p> <p>インデックス位置を入力する際には、次の点に注意すべきです。</p> <ul style="list-style-type: none"> <li>• 最大 60 個のインデックス位置をテーブルに保存することができる。</li> <li>• テーブルへの第 1 入力 はインデックス位置 1 に対応する；第 n 入力 はインデックス位置 n に対応する。</li> <li>• インデックス位置は、昇順で（負のトラバース範囲から正のトラバース範囲へ）、入力と入力の間を空けずにテーブルに入力されるべきである。連続した位置値を同一にはできない。</li> <li>• モジュール 360 (MD: ROT_IS_MODULO = "1") を使用してインデックス軸が回転軸と定義された場合 (MD: IS_ROT_AX = "1"), その位置値は、0 pos. &lt; 360 の範囲に制限される。</li> </ul> <p>テーブルで使用されるインデックス位置の数は、MD: INDEX_AX_LENGTH_POS_TAB_2 によって定義されます。</p> <p>MD: INDEX_AX_ASSIGN_POS_TAB に値 1 を入力すると、インデックス位置テーブル 1 が現在の軸に割当てられます。</p>	
アプリケーション	ツールマガジン（ツールタレット、チェーンマガジン）	
例外、エラー	テーブルに 60 個以上の位置が入力された場合、アラーム 17020 "Illegal array index"（不正な配列インデックス）が出力されます。	
関連性	<p>MD: INDEX_AX_ASSIGN_POS_TAB      (軸はインデックス軸である)</p> <p>MD: INDEX_AX_LENGTH_POS_TAB_2   (テーブル 2 で使用されるインデックス位置の数)</p> <p>MD: IS_ROT_AX                      (回転軸)</p> <p>MD: ROT_IS_MODULO                (回転軸用のモジュール変換)</p>	

## ■ 軸別マシンデータ

30500 MD 番号	INDEX_AX_ASSIGN_POS_TAB 軸はインデックス軸である	
初期値 : 0	最小入力リミット : 0	最大入力リミット : 3
変更が有効になるための条件 : 電源オンまたは SW 4.3 以降では リセット	保護レベル : 2 / 7	単位 : -
データタイプ : BYTE	適用開始ソフトウェアバージョン : 1.1	
意味 :	<p>インデックス位置テーブル 1 あるいは 2 の割当てによって、軸がインデックス軸と宣言されます。</p> <p>0: この軸はインデックス軸と宣言されていない。</p> <p>1: この軸はインデックス軸である。インデックス位置はテーブル 1 に保存されている (MD: INDEX_AX_POS_TAB_1)。</p> <p>2: この軸はインデックス軸である。インデックス位置はテーブル 2 に保存されている (MD: INDEX_AX_POS_TAB_2)。</p> <p>3: SW 4.3 以降 (840D) および SW 2.3 以降 (810D) では、等間隔インデックス &gt;3: アラーム 17090 "Value violates upper limit" (値が上限を超えている)</p>	
アプリケーション	ツールマガジン (ツールタレット, チェーンマガジン)	
例外, エラー	<p>すべての軸が同じタイプ (直線軸, 回転軸, モジュロ 360_ファンクション) であるという条件で、複数の軸を 1 つのインデックス位置テーブルに割当てることができます。</p> <p>そうでない場合は、アラーム 4000 が電源オン中に出力されます。</p> <p>Alarm 17500 "Axis is not an indexing axis" (軸はインデックス軸ではない)</p> <p>Alarm 17090 "Value violates upper limit" (値が上限を超えている)</p>	
関連性	<p>MD: INDEX_AX_POS_TAB1 (インデックス位置テーブル 1)</p> <p>MD: INDEX_AX_LENGTH_POS_TAB_1 (テーブル 1 で使用されるインデックス位置の数)</p> <p>MD: INDEX_AX_POS_TAB2 (インデックス位置テーブル 2)</p> <p>MD: INDEX_AX_LENGTH_POS_TAB_1 (テーブル 2 で使用されるインデックス位置の数)</p> <p>値が 3 の場合の等間隔インデックスでは :</p> <p>MD: INDEX_AX_NUMERATOR 分子</p> <p>MD: INDEX_AX_DENOMINATOR 分母</p> <p>MD: INDEX_AX_OFFSET 第 1 インデックス位置</p> <p>MD: HIRTH_IS_ACTIVE Hirth tooth システム</p>	

30501 MD 番号	INDEX_AX_NUMERATOR 等間隔位置を使用したインデックス軸用の分子	
初期値 : 0	最小入力リミット : >0	最大入力リミット : ***
変更が有効になるための条件 : RESET	保護レベル : 2/7	単位 : mm / インチ / 度
データタイプ : DOUBLE	適用開始ソフトウェアバージョン : 4.3	
意味 :	2 つのインデックス位置が等間隔のときに、その 2 つのインデックス位置間の距離を計算するための分子の値を定義します。モジュロ軸はこの値を無視し、代わりに \$MA_MO-DULO_RANGE を使用します。	
この MD は次の場合は無意味	テーブルに応じて、等間隔インデックスが使用されない場合	
アプリケーション	「■ 拡張機能」を参照。	
関連性	MD 30502: INDEX_AX_DENOMINATOR, MD 30503: INDEX_AX_OFFSET; MD 30500: INDEX_AX_ASSIGN_POS_TAB	

30502 MD 番号	INDEX_AX_DENOMINATOR 等間隔位置を使用したインデックス軸用の分母	
初期値 : 1	最小入力リミット : 1	最大入力リミット : ***
変更が有効になるための条件 : RESET	保護レベル : 2/7	単位 : -
データタイプ : DWORD	適用開始ソフトウェアバージョン : 4.3	
意味 :	2 つのインデックス位置が等間隔のときに、その 2 つのインデックス位置間の距離を計算するための分母の値を定義します。モジュロ軸については、インデックス位置の数を指定します。	
この MD は次の場合は無意味	テーブルに応じて、等間隔インデックスが使用されない場合	
アプリケーション	「 <b>■</b> 拡張機能」を参照	
関連性	MD 30501: INDEX_AX_NUMERATOR, MD 30503: INDEX_AX_OFFSET; MD 30500: INDEX_AX_ASSIGN_POS_TAB	

30503 MD 番号	INDEX_AX_OFFSET 等間隔位置を使用したインデックス軸用の第 1 インデックス位置	
初期値 : 0.0	最小入力リミット : ***	最大入力リミット : ***
変更が有効になるための条件 : RESET	保護レベル : 2/7	単位 : mm / インチ / 度
データタイプ : DOUBLE	適用開始ソフトウェアバージョン : 4.3	
意味 :	等間隔位置を使用したインデックス軸についてゼロからの第 1 インデックス位置の位置を定義します。	
この MD は次の場合は無意味	テーブルに応じて、等間隔インデックスが使用されない場合	
アプリケーション	「 <b>■</b> 拡張機能」を参照。	
関連性	MD 30501, 30502, 30500	

30505 MD 番号	HIRTH_IS_ACTIVE Hirth tooth システムがアクティブである	
初期値 : 0	最小入力リミット : 0	最大入力リミット : 1
変更が有効になるための条件 : RESET	保護レベル : 2/7	単位 : -
データタイプ : BOOLEAN	適用開始ソフトウェアバージョン : 4.3	
意味 :	1 の値がセットされると、Hirth tooth システムがアクティブになります。	
この MD は次の場合は無意味	テーブルに応じて、等間隔インデックスが使用されない場合	
アプリケーション	「 <b>■</b> 拡張機能」を参照	
関連性	MD 30500, 30501, 30502, 30503	

## ■ システム変数

名称	\$AA_ACT_INDEX_AX_POS_NO[Achse]			
意味	到達あるいはオーバトラベルされた最後のインデックス位置の番号			
データタイプ	INTEGER			
値の範囲	0: インデックス軸でないで、インデックス位置が利用できない >0: インデックス軸の番号			
インデックス	軸	値の範囲	-	
アクセス	パートプログラムを読む		同期化アクションを読む	
前処理停止の実行	x			

名称	\$AA_PROG_INDEX_AX_POS_NO[Achse]			
意味	プログラムされたインデックス位置の番号			
値の範囲	0: インデックス軸でないで、インデックス位置が利用できない。あるいはインデックス軸が現在インデックス位置へ向かって移動していない。 >0: インデックス位置の番号			
インデックス	軸	値の範囲	-	
アクセス	パートプログラムを読む		同期化アクションを読む	
前処理停止の実行	x			



## 2.15.5 信号の説明

### ■ 軸別信号

DB31, ... DBX76.6 データブロック	Indexing axis in position (インデックス軸が定位置にある) 軸／スピンドルからの信号 (NCK --> PLC)	
エッジ評価: なし	信号更新: 周期的	信号が有効となる最初の SW バージョン: 1.1
信号ステータス 1 あるいは信号 遷移 0 ---> 1	<p>この信号は, "Exact stop fine" (正確停止微) の影響を受けます。</p> <p>"Exact stop fine" (正確停止微) が実行されると, この信号がセットされます。</p> <p>"Exact stop fine" から出ると, 信号は再度リセットされます。</p> <ul style="list-style-type: none"> <li>インデックス軸がインデックス位置上に位置決めされる。</li> </ul> <p>インデックス軸が, コード化位置インストラクションによって位置決めされている。</p> <p>SW 4.3 以降 (840D), SW 2.3 以降 (810D)</p> <p>"Exact stop fine" ウィンドウが到達してからインデックス軸がインデックス位置上に位置決めされた場合, どのようにしてインデックス位置が到達されたかに関係なく, この信号はイネーブルされます。</p>	
信号ステータス 0 あるいは信号 遷移 1 ---> 0	<ul style="list-style-type: none"> <li>軸がインデックス軸と定義されていない。</li> <li>インデックス軸がトラバース中である (IS "Travel command +/-" (移動コマンド +/-) (DB31, ... DBX64.7/64.6) がアクティブである)</li> <li>インデックス軸が, インデックス位置に対応していない位置に位置決めされる。</li> </ul> <p>例:</p> <p>JOG モードでは, たとえば RESET を使用した移動動作の強制終了後</p> <ul style="list-style-type: none"> <li>AUTOMATIC では: たとえば, インデックス軸が, AC または DC インストラクションによって制御されている選択位置にアプローチした場合。</li> <li>インデックス軸が, 自動モードで, コード化位置インストラクション (CAC, CACP, CACN, CDC, CIC) によって位置決めされていない。</li> <li>インデックス軸用の "Servo enable" (サーボイネーブル) 信号がキャンセルされている (IS "Servo enable" DB31, ... DBX2.1)。</li> </ul>	
この信号は次の場合は無意味	インデックス軸と定義されていない軸の場合 (MD 30500: INDEX_AX_ASSIGN_POS_TAB = "0")	
アプリケーション	ツールマガジン: マガジンからツールを取除くためのグリッパの起動は, インデックス軸が定位置にある ("indexing axis in position" = 1) ときにトリガされます。これは PLC ユーザプログラムにプログラムされなければなりません。	
例外, エラー	<p>注記:</p> <p>個々の分割についてインデックス位置テーブルに入力された軸位置を, (DRF を含む) ゼロオフセットによって変更することができます。インデックス軸の実際位置がインデックステーブルに入力された値とオフセットにマッチすると, "indexing axis in position" (インデックス軸が定位置にある) インタフェース信号が 1 にセットされます。AUTOMATIC モードで, DRF がインデックス軸に適用された場合, 軸がインデックス位置から移動した後も, インタフェース信号 "Indexing axis in position" はアクティブなままです。例外については, 「<b>■ 拡張機能</b>」の「Hirth tooth システム」を参照。</p>	
関連性	MD 30500: INDEX_AX_ASSIGN_POS_TAB (軸はインデックス軸である)	

## 2.15.6 例

例については、2.15.2「**■ 拡張機能**」と「**■ インデックス軸のスタートアップ**」を参照してください。

## 2.15.7 データフィールド

### ■ インタフェース信号

DB 番号	ビット, バイト	名称	参照
軸／スピンドル別			
31-61	76.6	インデックス軸が定位置にある	
31-61	60.4, 60.5	基準化／同期化 1, 基準化／同期化 2	R1

### ■ マシンデータ

番号	識別子	名称	参照
一般 (\$MN_...)			
10260	CONVERT_SCALING_SYSTEM	基本系切換えアクティブ	G2
10270	POS_TAB_SCALING_SYSTEM	位置テーブルの測定系	
10900	INDEX_AX_LENGTH_POS_TAB_1	テーブル 1 で使用されるインデックス位置の数	
10920	INDEX_AX_LENGTH_POS_TAB_2	テーブル 2 で使用されるインデックス位置の数	
10910	INDEX_AX_POS_TAB_1[n]	インデックス位置テーブル 1	
10930	INDEX_AX_POS_TAB_2[n]	インデックス位置テーブル 2	
軸／スピンドル別 (\$MA_...)			
30300	IS_ROT_AX	回転軸	R2
30310	ROT_IS_MODULO	回転軸用のモジュロ変換	R2
30320	DISPLAY_IS_MODULO	位置表示はモジュロ 360° である	R2
30500	INDEX_AX_ASSIGN_POS_TAB	軸はインデックス軸である	
30501	INDEX_AX_NUMERATOR	等間隔位置を使用したインデックス軸用の分子	
30502	INDEX_AX_DENOMINATOR	等間隔位置を使用したインデックス軸用の分母	
30503	INDEX_AX_OFFSET	等間隔位置を使用したインデックス軸用の第 1 インデックス位置	
30505	HIRTH_IS_ACTIVE	Hirth tooth システムがアクティブ	

### ■ 設定データ

番号	識別子	名称	参照
一般 (\$MN_...)			
41050	JOG_CONT_MODE_LEVELTRIGGRD	JOG 連続モード	H1

## ■ システム変数

ソフトウェアバージョン 4.3 以降では、次のシステム変数が存在します。

識別子	名称, 意味	参照
\$AA_ACT_INDEX_AX_POS_NO[axis]	到達あるいはオーバトラベルされた最後のインデックス位置の番号	PGA
\$AA_PROG_INDEX_AX_POS_NO[axis]	プログラムされたインデックス位置の番号	PGA

## ■ アラーム

アラームの詳細説明については次を参照してください。

参照： アラーム一覧

あるいは MMC 101/102 /103 を有するシステムのオンラインヘルプ

---

---

## 2.16 工具交換機能 (W3)

### 2.16.1 概略説明

CNC で制御された工作機械には、ワークピースを完全に加工するためのツールマガジンおよび自動ツールチェンジ能力が備わっています。

#### 手順

切削ごとにツールを変更するための手順は、次の 3 つのステップから構成されています。

- ツールキャリアを加工位置からツールチェンジ位置に移動させる
- ツールチェンジ
- ツールキャリアをツールチェンジ位置から新しい加工位置に移動させる

#### 始動

次を使用してツールチェンジを始動できます。

- T ファンクション，あるいは
- M コマンド (M06 が好ましい)

ツールチェンジには、次の 2 つのオプションがあります。

T 番号を使用した即時チェンジ，または T 番号を使用した準備：

##### 1. 即時チェンジ

- T ファンクションにより，新しいツールが直ちにロードされます
- 通常のアプリケーション：ツールタレットを使用した旋盤

##### 2. 準備

- T ファンクション実行時に，新しいツールがツールチェンジ用に準備されます。
- 古いツールをスピンドルから取除き，新しいツールをロードするのに M ファンクションが使用されます。
- ツールチェンジ用の M コマンドをマシンデータで定義できます。
- 通常のアプリケーション：ツールマガジンを使用したフライス盤（加工プロセスを中断せずに新しいツールをツールチェンジ位置に移動させるため）。

## ツールチェンジ点

ツールチェンジ点を選択することによって、切削から切削までの時間は重大な影響を受けます。ツールチェンジ点は、工作機械の概念に応じて選択されますが、現在の加工タスクに応じて選択される場合もあります。

固定点アプローチファンクション (G75) を使用することにより、マシン軸上の固定位置にアプローチすることができます。このような固定位置はマシンデータに保存されます。複数のツールチェンジ点を定義および始動することができます。

ツールチェンジには、ロード予定のツールが適時にツールチェンジ位置で利用できることを保証するツール管理システムが必要となります。

## 2.16.2 詳細説明

### ■ ツールチェンジオペレーションの概要

CNC で制御された工作機械には、ワークピースを完全に加工するためのツールマガジンおよび自動ツールチェンジ能力が備わっています。

### ツールチェンジ機器

ツールマガジンおよびツールチェンジ機器は、マシンタイプに応じて選択されます。

旋盤： ツールタレット（円盤、フラットまたは傾斜ツールタレット）；特殊な機器は必要ありません：ツールタレットを回転させることにより、ツールを変更します。

フライス盤： チェンジ用グリッパ／ダブルグリッパ付きマガジン（チェーン、プレート、円盤、カセット）

ツールチェンジによって加工が中断されるので、アイドル時間は最小化されなければなりません。

### ツールチェンジ時間

ツールチェンジ時間は、主として工作機械の構成によって決定されます。

通常のツールチェンジ時間は次のようになります。

0.1 ～ 0.2 秒                      ツールタレットの回転時間

0.3 ～ 2 秒                      ツールが準備されているときの、グリッパを使用したツールチェンジ時間

### 切削から切削までの時間

切削から切削までの時間とは、輪郭上の中断点から後退してから（切削中断）、ツールが変更され、スピンドルが回転し、新しいツールで中断点に再位置決めする（切削再開）までに要する時間のことを言います。

通常の、切削から切削までの時間は次のようになります。

0.3 ～ 1 秒                      ツールタレットを使用した旋盤の場合

0.5 ～ 5 秒                      ツールチェンジャを使用したフライス盤の場合

## 要求事項

ツールチェンジオペレーションは、次の要求事項を満足しなければなりません。

- 短いアイドル時間
- 加工中の、ツールの急速サーチ、準備および戻し
- ツールチェンジサイクルの単純なプログラミング
- 必要な軸およびグリッパ動作の自動オペレーション
- 簡単な故障修復

## ■ オペレーションの順番

### ツールチェンジの順番

切削から切削までのツールチェンジオペレーションは、次の3つのステップで実行されます。

- ツールキャリアを加工位置からツールチェンジ位置へ移動させる
- ツールチェンジ
- ツールキャリアをツールチェンジ位置から新しい加工位置へ移動させる

ツールチェンジ位置はそのマシンの概念によって決まります。詳細は「■ ツールチェンジ点」で説明します。

### スピンドルの制御

ツールチェンジ中にスピンドルを制御する方法も同様にそのマシンの構造によって決まります。様々なオプションには次のようなシステムが含まれます。

- スピンドルが回転し続けるシステム
- あるいは、スピンドルが停止されるシステム
- あるいは、スピンドルが位置決めされるシステム

## ■ 制御

### 始動

次の機能を利用してツールチェンジを始動できます。

- T ファンクション、あるいは
- M コマンド (M06 が好ましい)

以下のように、MD: TOOL\_CHANGE\_MODE で選択を行います。

TOOL\_CHANGE\_MODE = 0

- T ファンクションを実行すると直ちに新しいツールがロードされます。
- 通常のアプリケーション：ツールタレットを使用した旋盤

TOOL\_CHANGE\_MODE = 1

- T ファンクションを実行すると、新しいツールがツールチェンジ用に準備されます。
- 古いツールをスピンドルから取除き、新しいツールをロードするには、M

ファンクションを使用します。

- ツールチェンジ用の M コマンドは、MD: TOOL\_CHANGE\_M\_CODE で定義されます。
- 通常のアプリケーション：ツールマガジンを使用したフライス盤（加工プロセスを中断することなく、新しいツールをツールチェンジ位置へ移動させるため）

注記：ツールオフセット番号が PLC あるいは MMC ツールマネージャから提供されている場合は、"STOPRE" ブロックサーチ停止が適切な点で挿入される必要があります。

ですが、ツール半径補償 (G41/G42) あるいはスプライン補間が実行中のときは、パス計算用に複数のブロックが前もって必要となるので、STOPRE は回避されるべきです。

M06 ツールチェンジに適用される、以下のような M ファンクションに関する詳細情報については、

- 拡張アドレス
- PLC への出力時間
- 補助ファンクショングループ
- ブロックサーチ
- オーバストア

## ■ ツールチェンジ点

### ツールチェンジ点

ツールチェンジ点の選択は、切削から切削までの時間に重大な影響を与えます。ツールチェンジ点は、工作機械の概念に応じて選択されますが、現在の加工状況に応じて選択される場合もあります。

固定点アプローチファンクション (G75) を使用することにより、マシン軸上の固定位置にアプローチすることができます。

N20 G75 FP=2 X1=0 Y1=0 Z1=0 LF

### 固定点

1 つのマシン軸につき 2 つの固定位置が MD: FIX\_POINT\_POS[N] に保存されます。これらは FP=1 または FP=2 で指定されます。値が定義されない場合は、FP=1 となります。

これら 2 つの点のどちらかへ移動するマシン軸は、そのマシン名とダミー位置（使用されない）によって定義されなければなりません。

MD に保存されている位置には、早送り G0 でアプローチします。

G75 を使用したブロックでは、SPOS および SPOSA を使用してスピンドルを位置決めすることができます。



### 2.16.3 補足条件

ツールチェンジには、ロード予定のツールが適時にツールチェンジ位置で利用できるということを保証するツール管理システムが必要となります。

### 2.16.4 データの説明 (MD, SD)

#### ■ マシンデータ

ツールチェンジで必要となるマシンデータについては、次のセクションを参照してください。

MD 番号	識別子	ファンクション説明
22500	TOOL_CHANGE_MODE	W1
22600	TOOL_CHANGE_M_CODE	W1
30600	FIX_POINT_POS[n]	K1
22200	AUXFU_M_SYNC_TYPE	H2
22220	AUXFU_T_SYNC_TYPE	H2

### 2.16.5 信号の説明

本ファンクション説明には別の信号はありません。

### 2.16.6 例

フライス盤について、次の例では、ツールチェンジャおよび位置固定ツールチェンジ点を使用したツールチェンジオペレーションの通常の順番を示しています。

加工プログラム

```
N 970   G0 X= Y= Z= LF           ; 輪郭からの後退
N 980   T1 LF                     ; ツール選択
N 990   W_WECHSEL LF             ; パラメータを使用しないサブプログラムコール
N 1000  G90 G0 X= Y= Z= M3 S1000 LF; 加工再開
```

ツールチェンジ用のサブプログラム

```
PROC    W_WECHSEL LF
N 10     SPOSA = S0 LF           ; スピンドル位置決め
N 20     G75 FP=2 X1=0 Y1=0 Z1=0 ; ツールチェンジ点へのアプローチ
                                   (「■ ツールチェンジ点」を参照)
N 30     M06 LF                  ; ツールの変更
N 40     M17 LF
```

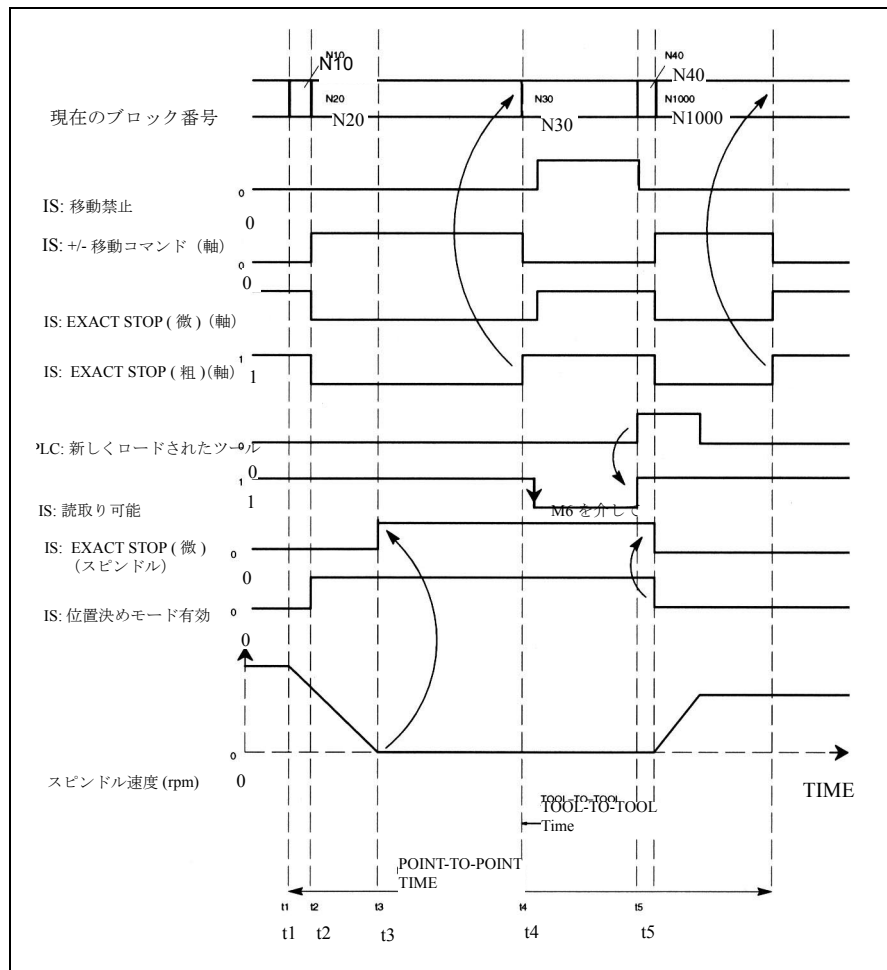


図 2.110 ツールチェンジの順番

- t1 軸が静止  
スピンドルが回転中  
N 10 で、ツールチェンジサイクルのスタート
- t2 N 20 で、G75 を使用して軸がツールチェンジ点に移動します
- t3 スピンドルが、ブロック N 10 でプログラムされた位置に到達します
- t4 軸が N 20 から EXACT STOP(粗)に到達します；N 30 はここからスタートします：  
M06 が以前のツールをスピンドルから取除き、新しいツールをロードおよび固定します。
- t5 ツールチェンジャが旋回して元の位置へ戻ります。
- この後、メインプログラム呼出しの N 1000 で、
- 新しいツールオフセットを選択できます
  - あるいは、軸を輪郭に戻すことができます
  - あるいは、スピンドルを加速させることができます。

## 2.16.7 データフィールド, リスト

### ■ インタフェース信号

DB 番号	ビット, バイト	名称	参照
チャンネル別			
21-28	194.6	M ファンクション M06	

### ■ マシンデータ

番号	識別子	名称	参照
一般 (\$MN_...)			
18082	MM_NUM_TOOL	ツールの数	S7
チャンネル別 (\$MC_...)			
22200	AUXFU_M_SYNC_TYPE	M ファンクションの出力時間	H2
22220	AUXFU_T_SYNC_TYPE	T ファンクションの出力時間	H2
22550	TOOL_CHANGE_MODE	M ファンクションを使用した新しいツールオフセット	
22560	TOOL_CHANGE_M_CODE	ツールチェンジ用の M ファンクション	
軸別 (\$MC_...)			
30600	FIX_POINT_POS[n]	G75 を使用したマシン軸の固定点位置	

### ■ アラーム

アラームの詳細説明については次を参照してください。

参照： アラーム一覧

あるいは MMC 101/102/103 を有するシステムのオンラインヘルプ

## Yaskawa Siemens CNC シリーズ

本製品の最終使用者が軍事関係であったり、用途が兵器などの製造用である場合には、「外国為替及び外国貿易法」の定める輸出規制の対象となることがありますので、輸出される際には十分な審査及び必要な輸出手続きをお取りください。

製品改良のため、定格、寸法などの一部を予告なしに変更することがあります。  
この資料についてのお問い合わせは、当社代理店もしくは、下記の営業部門にお尋ねください。

製造

株式会社 安川電機

シーメンスAG

販売

シーメンス・ジャパン株式会社

工作機械営業本部

東京都品川区大崎1-11-1 ゲートシティ大崎ウエストタワー 〒141-8644  
TEL (03) 3493-7411 FAX (03) 3493-7422

アフターサービス

カスタマーサービス事業本部

TEL 0120-996095(フリーダイヤル) FAX (03)3493-7433

シーメンス・ジャパン株式会社  
<http://www.siemens.co.jp>