

SIEMENS

SINUMERIK 840D/810D/FM-NC
Notions complémentaires

Manuel de
programmation

Edition 12.97

Documentation utilisateur

SINUMERIK 840D/810D/FM-NC

Notions complémentaires

Manuel de programmation

Documentation utilisateur

Valable pour :

<i>Commande</i>	<i>Version de logiciel</i>	
SINUMERIK 840D		4
SINUMERIK 840DE (version exportation)		4
SINUMERIK 810D		2
SINUMERIK 810DE (version exportation)		2
SINUMERIK FM-NC		3

Edition 12.97

Programmation NC flexible	1
Sous-programmes, macroprogrammation	2
Gestion des fichiers et programmes	3
Zones de protection	4
Instructions de déplacement spéciales	5
Frames	6
Usinage cinq axes	7
Correcteurs d'outils	8
Mode de déplacement	9
Actions synchrones au déplacement	10
Oscillation	11
Poinçonnage et grignotage	12
Autres fonctions	13
Programmes de chariotage personnalisés	14
Tables	15
Annexe	A

Documentation SINUMERIK®

Récapitulatif des éditions

Les éditions mentionnées ci-dessous ont paru avant la présente édition.

La colonne "Observations" comporte des lettres majuscules caractérisant la nature des éditions parues jusqu'ici.

Signification des lettres :

- A** Documentation nouvelle
- B** Réimpression inchangée portant le nouveau numéro de référence
- C** Edition remaniée portant la nouvelle date de publication

Si l'exposé figurant sur une page a été modifié sur le plan technique par rapport à l'édition précédente, la date de publication de la nouvelle édition figure dans l'en-tête de la page concernée.

Edition	N° de référence	Observations
02.95	6FC5298-2AB00-0DP0	A
04.95	6FC5298-2AB00-0DP1	C
12.95	6FC5298-3AB10-0DP0	C
03.96	6FC5298-3AB10-0DP1	C
08.97	6FC5298-4AB10-0DP0	C
12.97	6FC5298-4AB10-0DP1	C

Ce manuel fait partie intégrante de la documentation sur CD-ROM (en anglais) (**DOCONCD**)

Edition	N° de référence	Observations
02.98	6FC5 298-4CA00-0BG1 (Read)	C
02.98	6FC5 298-4CB00-0BG1 (Print)	C
02.98	6FC5 298-4CC00-0BG1 (Net)	C

Vous trouverez d'autres informations sur Internet sous :
<http://www.aut.siemens.de/sinumerik>

La présente documentation a été établie avec WinWord V 7.0 et Designer V 4.0.

Toute reproduction de ce support d'information, toute exploitation, communication de son contenu sont interdites, sauf autorisation expresse. Tout manquement à cette règle est illicite et expose son auteur au versement de dommages et intérêts. Tous nos droits sont réservés, notamment pour le cas de la délivrance d'un brevet ou celui de l'enregistrement d'un modèle d'utilité.

© Siemens AG 1996 - 1997. All Rights Reserved.

La commande numérique peut posséder des fonctions qui dépassent le cadre de la présente description. Le client ne peut toutefois pas faire valoir de droit en liaison avec ces fonctions, que ce soit dans le cas de matériels neufs ou dans le cadre d'interventions du service après-vente.

Nous avons vérifié que le contenu de la présente documentation correspondait bien au matériel et logiciel décrits. Des différences peuvent cependant exister, de sorte que nous ne pouvons assumer la responsabilité d'une concordance totale. Le contenu de cette documentation est contrôlé régulièrement et les corrections nécessaires sont intégrées aux éditions ultérieures. Il sera réservé le meilleur accueil à toute suggestion visant à améliorer la présente documentation.

Sous réserve de modifications.

Table des matières

Programmation CN flexible	1-19
1.1 Variables et paramètres de calcul.....	1-20
1.2 Définition des variables.....	1-23
1.3 Définition de tableaux.....	1-27
1.4 Programmation indirecte.....	1-33
1.5 Affectations.....	1-35
1.6 Opérateurs/Fonctions de calcul.....	1-36
1.7 Opérateurs relationnels et opérateurs logiques.....	1-38
1.8 Priorité des opérateurs.....	1-41
1.9 Conversion du type de variable.....	1-42
1.10 Opérations sur les chaînes de caractères.....	1-43
1.11 Sauts de programme inconditionnels.....	1-53
1.12 Sauts de programme conditionnels.....	1-55
1.13 Instruction CASE.....	1-57
1.14 Structures de contrôle.....	1-59
1.15 Coordination de programmes.....	1-64
1.16 Routine d'interruption.....	1-69
1.17 Permutation d'axe, permutation de broche.....	1-77
1.18 Axes géométriques permutables, GEOAX.....	1-81
1.19 Répétition de parties de programme (à partir du logiciel 4.3).....	1-86
Sous-programmes, macroprogrammation	2-95
2.1 Utilisation des sous-programmes.....	2-96
2.2 Sous-programme avec mécanisme SAVE.....	2-98
2.3 Sous-programmes avec transfert de paramètres.....	2-99
2.4 Appel d'un sous-programme.....	2-103
2.5 Sous-programme à répétition.....	2-106
2.6 Sous-programme modal, MCALL.....	2-107
2.7 Appel indirect d'un sous-programme.....	2-108
2.8 Appel d'un sous-programme avec indication de chemin et paramètres, PCALL.....	2-109

2.9 Inhibition de l'affichage du bloc courant, DISPLOF	2-110
2.10 Inhibition du bloc par bloc, SBLOF, SBLON (à partir de SW 4.3)	2-111
2.11 Macroprogrammation	2-114
2.12 Exécution d'un sous-programme externe (à partir du logiciel SW 4.2)	2-116
2.13 Cycles : paramétrer des cycles utilisateur.....	2-119
Gestion des fichiers et des programmes	3-125
3.1 Vue d'ensemble.....	3-126
3.2 Mémoire de programmes	3-127
3.3 Mémoire de travail.....	3-132
3.4 Définition des données utilisateur	3-135
3.5 Définition des niveaux de protection des données utilisateur (GUD).....	3-138
3.6 Définition des macros.....	3-141
Zones de protection	4-143
4.1 Définition des zones de protection CPROTDEF, NPROTDEF	4-144
4.2 Activation / Désactivation des zones de protection CPROT, NPROT.....	4-148
Instructions de déplacement spéciales	5-153
5.1 Accostage de positions codées, CAC, CIC, CDC, CACP, CACN.....	5-154
5.2 Interpolation de type spline.....	5-156
5.3 Compactage	5-164
5.4 Interpolation polynomiale - POLY.....	5-167
5.5 Mesure avec palpeur à déclenchement, MEAS, MEAW	5-173
5.6 Mesure étendue MEASA, MEAWA, MEAC (à partir de SW 4, option)	5-176
5.7 Fonctions spéciales pour l'utilisateur OEM	5-186
Frames	6-187
6.1 Appel de transformations de coordonnées par variables frames.....	6-188
6.2 Affectation de valeurs à des variables frames ou à des frames	6-192
6.3 Décalage grossier et décalage fin	6-199
6.4 Décalage DRF	6-200
6.5 Décalage d'origine externe.....	6-201
6.6 Programmation d'un décalage Preset, PRESETON	6-202

6.7 Désactivation des frames.....	6-203
6.8 Calcul d'un frame à partir de 3 points mesurés dans l'espace MEAFRAME	6-204
Usinage cinq axes	7-209
7.1 Transformation cinq axes - TRAORI, TRAF00F	7-210
7.2 Référence des axes d'orientation - ORIWKS, ORIMKS	7-219
7.3 Positions singulières et leur traitement	7-221
7.4 TRANSMIT, TRAF00F	7-222
7.5 TRACYL, TRAF00F, OFFN.....	7-226
7.6 TRAANG, TRAF00F	7-232
Corrections d'outil	8-239
8.1 Mémoire de correcteurs	8-240
8.2 Liste des types d'outils	8-242
8.3 Instructions de gestion d'outils	8-243
8.4 Correction d'outil en ligne PUTFTOCF, PUTFTOC, FTOCON, FTOCOF	8-246
8.5 Activation des corrections d'outil 3D	8-252
8.6 Orientation de l'outil.....	8-259
Mode de déplacement	9-265
9.1 Positionnement tangentiel, TANG, TANGON, TANGOF	9-266
9.2 Déplacements conjugués, TRAILON, TRAILOF	9-271
9.3 Table de courbe, CTABDEF, CTABEND, CTAB, CTABINV	9-275
9.4 Couplage de deux axes par valeur pilote, LEADON, LEADOF.....	9-284
9.5 Couplage à interpolation par valeur pilote, LEADONP, LEADOFP.....	9-290
9.5.1 Couplage d'axes à interpolation par valeur pilote Type A.....	9-291
9.5.2 Couplage d'axes à interpolation par valeur pilote Type B.....	9-293
9.6 Variation de l'avance, FNORM, FLIN, FCUB, FPO.....	9-295
9.7 Tampon d'exécution, STARTFIFO, STOPFIFO, STOPRE	9-300
9.8 Réaccostage du contour, REPOSA, REPOS L, REPOSQ, REPOSH.....	9-302
Action synchrone au déplacement	10-307
10.1 Structures, bases générales	10-309
10.1.1 Programmation et éléments d'instructions.....	10-311
10.1.2 Plage de validité : Numéro d'identification ID.....	10-312

10.1.3 Mot-clé.....	10-313
10.1.4 Actions.....	10-315
10.1.5 Vue d'ensemble des actions synchrones.....	10-317
10.2 Notions de base pour conditions et actions.....	10-319
10.3 Variables temps réel spéciales pour actions synchrones	10-322
10.3.1 Mémentos/compteurs \$AC_MARKER[n]	10-322
10.3.2 Variable de temporisation \$AC_TIMER[n], à partir de SW 4	10-322
10.3.3 Paramètres d'actions synchrones \$AC_PARAM[n]	10-323
10.3.4 Accès aux paramètres R \$Rxx	10-324
10.3.5 Ecriture/Lecture des paramètres machine et des données de réglage, à partir de SW 4	10-325
10.3.6 Variable FIFO \$AC_FIFO1[n] ... \$AC_FIFO10[n], à partir de SW 4.....	10-326
10.4 Actions dans actions synchrones.....	10-328
10.4.1 Sortie de fonctions auxiliaires	10-328
10.4.2 Activation du blocage de l'introduction via l'interface RDISABLE	10-329
10.4.3 Annuler l'arrêt du prétraitement des blocs STOPREOF.....	10-330
10.4.4 Effacement de la distance restant à parcourir	10-331
10.4.5 Effacement de la distance restant à parcourir, avec préparation, DELDTG, DELTG(axe1,..).....	10-331
10.4.6 Effacement de la distance restant à parcourir, sans préparation, DELD, DELD (axe1, ...), à partir de SW 4.3	10-333
10.4.7 Définition d'un polynôme, FCTDEF, synchrone avec un bloc.....	10-334
10.4.8 Positionnement par laser	10-336
10.4.9 Fonction d'évaluation SYNFACT	10-337
10.4.10 Régulation AC (additive)	10-338
10.4.11 Régulation AC (multiplicative).....	10-339
10.4.12 Régulation d'écartement ("de distance de tir") avec correction limitée	10-340
10.4.13 Correction d'outil en ligne FTOC.....	10-342
10.4.14 Déplacements de positionnement.....	10-344
10.4.15 Positionner l'axe : POS	10-344
10.4.16 Lancer/Arrêter l'axe : MOV.....	10-345
10.4.17 Avance axiale : AA	10-346
10.4.18 Fins de course logiciels.....	10-346
10.4.19 Coordination des axes	10-346
10.4.20 Préréglage des mémoires de valeurs réelles.....	10-348
10.4.21 Déplacement des broches	10-349
10.4.22 Déplacements conjugués : TRAILON, TRAILOF	10-350
10.4.23 Couplage par valeur pilote : LEADON, LEADOF.....	10-351
10.4.24 Mesure	10-353
10.4.25 Activer/Effacer les marques d'attente : SETM, CLEARM	10-353
10.4.26 Réactions en cas de défaut.....	10-354
10.5 Cycles technologiques.....	10-355

10.5.1 Bloquer, libérer, interrompre : LOCK, UNLOCK, RESET	10-357
10.6 Effacer une action synchrone : CANCEL	10-359
10.7 Conditions marginales.....	10-360
Oscillation	11-365
11.1 Oscillation asynchrone	11-366
11.2 Oscillation commandée par des actions synchrones.....	11-372
Poinçonnage et grignotage	12-383
12.1 Activation, désactivation.....	12-384
12.1.1 Instructions de langage.....	12-384
12.1.2 Utilisation des fonctions M	12-387
12.2 Segmentation automatique de la distance à parcourir.....	12-388
12.2.1 Segmentation de la distance à parcourir avec des axes à interpolation.....	12-389
12.2.2 Segmentation de la distance à parcourir avec des axes individuels.....	12-390
12.2.3 Exemples de programmation	12-392
Autres fonctions	13-395
13.1 Fonctions spécifiques à un axe AXNAME, SPI, ISAXIS	13-396
13.2 Apprentissage de compensation : QECLRNON, QECLRNOF	13-397
13.3 Broche synchrone	13-399
13.4 Fonctions H.....	13-409
13.5 Définition des sorties de fonctions	13-410
Programmes de chariotage personnalisés	14-413
14.1 Fonctions évoluées pour le chariotage	14-414
14.2 Préparation du contour - CONTPRON.....	14-415
14.3 Point d'intersection de deux éléments de contour - INTERSEC.....	14-419
14.4 Exécution d'un élément de contour de la table - EXECTAB	14-421
14.5 Calcul des données d'un cercle - CALCDAT	14-422
Tables	15-425
15.1 Liste des instructions.....	15-426
15.2 Liste des adresses	15-450
15.3 Liste des fonctions G/Fonctions préparatoires.....	15-459
15.4 Liste des sous-programmes prédéfinis	15-469

15.5 Liste des variables système (programmes pièce)	15-489
15.6 Liste des variables système	15-496
15.6.1 Paramètres R	15-496
15.6.2 Frames	15-496
15.6.3 Données du porte-outil.....	15-496
15.6.4 Zones de protection spécifiques à un canal.....	15-497
15.6.5 Entrées/sorties	15-499
15.6.6 Lecture et écriture de variables AP	15-499
15.6.7 Temps	15-500
15.6.8 Etats des canaux.....	15-500
15.6.9 Variables pour trajectoire	15-501
15.6.10 Valeurs polynomiales pour action synchrone.....	15-503
15.6.11 Variables système spécifiques à un axe	15-504
15.6.12 Données des broches	15-508
15.6.13 Variables système pour Safety Integrated	15-510
15.6.14 Valeurs programmées (synchrones au prétraitement des blocs)	15-510
15.6.15 Ordres vers/du canal.....	15-511
Annexe	A-513
A Index	A-515
B Commandes, descripteurs	A-521

840 D
NCU 571840 D
NCU 572
NCU 573

FM-NC



810D

Avant-propos

Composition de la documentation

La documentation SINUMERIK comporte trois volets :

- documentation générale
documentation utilisateur
- documentation constructeur/SAV

Destinataires

Le présent manuel s'adresse au programmeur de la commande numérique et contient les informations détaillées nécessaires pour programmer les commandes SINUMERIK 840D/810D et FM-NC.

Etendue

Le présent manuel de programmation décrit les fonctionnalités de la version standard. Les compléments ou modifications apportés par le constructeur de la machine sont documentés par celui-ci.

De plus amples informations sur les autres publications relatives à la SINUMERIK 840D, 810D OU FM-NC ainsi que sur les publications s'appliquant à l'ensemble des commandes SINUMERIK (par exemple interface universelle, cycles de mesure...) seront fournies par votre agence SIEMENS.

La commande numérique peut posséder des fonctions qui dépassent le cadre de la présente description. Le client ne peut toutefois pas faire valoir de droit en liaison avec ces fonctions, que ce soit dans le cas de matériels neufs ou dans le cadre d'interventions du service après-vente.

Composition de la documentation



840 D
NCU 571



840 D
NCU 572



FM-NC



810D
NCU 573

Validité :

Le présent manuel de programmation est valable pour les commandes suivantes :

SINUMERIK 840D	SW4
SINUMERIK 840DE (version exportation)	SW4
SINUMERIK 810D	SW2
SINUMERIK 810DE (version exportation)	SW2
SINUMERIK FM-NC	SW3

avec les tableaux de commande MMC 100 ou MMC 102/103.

Version exportation

Les fonctions suivantes ne sont pas disponibles dans la version exportation :

Fonction	FM-NC	810DE	840DE
Paquet pour usinage 5 axes	–	–	–
Paquet de transformation manipulation (5 axes)	–	–	–
Interpolation multiaxe (> 4 axes)	–	–	–
Interpolation hélicoïdale 2D+6	–	–	–
Actions synchrones niveau 2	–	–	O1)
Mesure niveau 2	–	–	O1)
Commande adaptative	–	–	O1)
Continuous dressing	–	–	O1)
Utilisation des cycles de compilation (OEM)	–	–	–
Compensation multidimensionnelle de flèche	–	–	O1)

– Fonction impossible

1) Fonctionnalité restreinte

840 D
NCU 571840 D
NCU 572
NCU 573

FM-NC



810D

Présentation des descriptions

Toutes les fonctions et possibilités de programmation sont décrites en suivant la même structure interne, dans la mesure où cela est utile et possible. La décomposition des descriptions en différents niveaux vous permet de trouver directement les informations qu'il vous faut.

1. La vue d'ensemble rapide



Si vous souhaitez consulter une instruction rarement utilisée ou la signification d'un paramètre, vous trouverez rapidement la manière de programmer la fonction et les explications relatives aux instructions et paramètres.



Ces informations figurent toujours en début de page

Remarque :

Pour des raisons de place, il n'est pas possible de faire figurer, pour les différentes instructions, adresses et paramètres, tous les modes de représentation autorisés par le langage de programmation. Par conséquent, la programmation des fonctions est toujours indiquée de la manière dont celles-ci se présentent le plus souvent à l'atelier.

2 Cycles et images de perçage/taurage/alésage
3.95'

2.1 Cycles de perçage/taurage/alésage

2.1.2 Perçage, centrage – CYCLE81

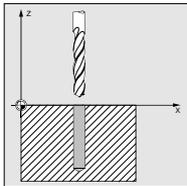
Programmation

CYCLE81 (RTP, RFP, SDIS, DP, DPR)

RTP	real	Plan de retrait (en absolu)
RFP	real	Plan de référence (en absolu)
SDIS	real	Distance de sécurité (introduire sans signe)
DP	real	Profondeur finale de perçage (en absolu)
DPR	real	Profondeur finale de perçage par rapport au plan de référence (introduire sans signe)

But et utilisation du cycle

L'outil perce avec la vitesse de rotation de broche et la vitesse d'avance programmées, jusqu'à la profondeur finale de perçage introduite.



Déroulement du cycle

Position atteinte avant le début du cycle :
La position de perçage est la position suivant les deux axes du plan sélectionné.

Le cycle génère la séquence de déplacement suivante :

- Accostage en G0 du plan de référence décalé vers l'avant de la distance de sécurité
- Déplacement à la profondeur finale de perçage avec l'avance programmée (G1) dans le programme appelant
- Retrait au plan de retrait en G0

2-8
© Siemens AG 1997 All rights reserved. SIC029-34647
SINUMERIK 840D/810D/NC



840 D
NCU 571



840 D
NCU 572
NCU 573



FM-NC



810D

2. Explications détaillées

Dans la partie théorique, vous trouverez les réponses aux questions suivantes :



A quoi sert l'instruction ?



Quel est l'effet de l'instruction ?



Comment se présente la séquence des opérations ?



A quoi servent les paramètres ?

Quels sont les autres points dont il faut plus particulièrement tenir compte ?

Les parties théoriques serviront avant tout à l'apprentissage des utilisateurs de CN débutants. Nous leur recommandons de lire attentivement le manuel en entier pour avoir une idée des possibilités et des performances de leur commande SINUMERIK.

3/96 2.1 Cycles de perçage/taraudage/alésage

Signification des paramètres

RFP et RTP
En règle générale, les plans de référence (RFP) et de retrait (RTP) ont des valeurs différentes. Dans le cycle, on est parti du principe que le plan de retrait se trouve avant le plan de référence. La distance du plan de retrait à la profondeur finale de perçage est donc plus grand que la distance du plan de référence à la profondeur finale de perçage.

SDIS
La distance de sécurité (SDIS) agit par rapport au plan de référence. Celui-ci est décalé vers l'avant de la valeur de cette distance de sécurité.

DP et DPR
La profondeur de perçage peut être déclarée en absolu (DP) ou en relatif (DPR) par rapport au plan de référence.
En absolu, cette valeur est directement parcourue dans le cycle.
En relatif, le cycle calcule la profondeur qui en résulte à l'aide de la position des plans de référence et de retrait.

Informations complémentaires
Si une valeur est introduite pour DP et pour DPR, la profondeur finale de perçage est déduite du DPR. Dans le cas où celle-ci ne correspond pas à la profondeur absolue programmée sous DP, le message «Profondeur : selon valeur pour profondeur relative» est émis dans la ligne de dialogue. Pour des valeurs identiques du plan de référence et du plan de retrait, il n'est pas permis de déclarer une profondeur en relatif.

© Siemens AG 1997 All rights reserved. 6FC5283-3AB40 SINUMERIK 840D/810D/FM-NC (PG)



3. De la théorie à la pratique

Pour l'application concrète de l'instruction, veuillez vous référer à l'exemple de programmation.

La partie théorique est suivie, pour pratiquement toutes les instructions, d'un exemple d'application.

3/96 2.1 Cycles de perçage/taraudage/alésage

Il en résulte le message d'erreur 61101 «Plan de référence mal défini» et le cycle n'est pas exécuté. Ce message d'erreur est aussi émis lorsque le plan de retrait se trouve après le plan de référence, sa distance à la profondeur finale de perçage est donc plus petite.

Exemple de programmation

Perçage_centrage
Avec ce programme, vous pouvez réaliser 3 trous en utilisant le cycle de perçage CYCLE81, celui-ci étant appelé avec différentes dotations de paramètres. L'axe de perçage est toujours l'axe Z.

DEF REAL DP, DPR	Définition des paramètres
N10 DP=35	Valeur pour profondeur finale de perçage en absolu
N20 G0 G90 F200 S300 M3	Définition des conditions technologiques
N30 D3 T3 B10	Accostage du plan de retrait
N40 X40 Y120	Accostage de la première position de perçage
N50 CYCLE81 (110, 100, 2, DP)	Appel du cycle avec profondeur finale de perçage en absolu, distance de sécurité et liste de paramètres incomplète
N60 Y30	Accoster la position de perçage suivante
N70 CYCLE81 (110, 102, ., DPR)	Appel de cycle sans distance de sécurité
N80 DPR=65	Valeur pour profondeur finale de perçage en relatif
N90 G0 G90 F180 S300 M03	Définition des conditions technologiques
N100 X90	Accoster la position suivante
N110 CYCLE81 (110, 100, 2, . DPR)	Appel du cycle avec profondeur finale de perçage en relatif et distance de sécurité
N120 M30	Fin de programme

© Siemens AG 1997 All rights reserved. 6FC5283-3AB40 SINUMERIK 840D/810D/FM-NC (PG)



840 D
NCU 571



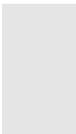
840 D
NCU 572
NCU 573



FM-NC



810D

 **Signification des symboles**



Procédure



Signification



Fonction



Paramètres



Exemple de programmation



Programmation



Remarques complémentaires



Renvoi à d'autres documentations et chapitres



Dangers ou sources d'erreur



840 D
NCU 571840 D
NCU 572
NCU 573

FM-NC



810D

Principe

Votre SINUMERIK 840D, 810D ou FM-NC est conçue d'après l'état actuel de la technique et des règles, normes et règlements de sécurité généralement reconnus.

Equipements supplémentaires

Des appareils et équipements supplémentaires ainsi que des extensions proposées par SIEMENS permettent d'élargir le domaine d'application des commandes SIEMENS.

Personnel

L'utilisation de la commande doit être réservée au **personnel formé, autorisé et fiable**. Les personnes ne possédant pas la formation nécessaire ne sont pas autorisées à utiliser la commande, même brièvement.

Les **compétences** du personnel en matière de réglage, d'utilisation et de maintenance doivent être clairement **définies** et le respect de ces compétences doit être **contrôlé**.

Comportement

Avant la mise en service de la commande, s'assurer que le manuel de mise en oeuvre a été lu et compris par le personnel compétent. L'exploitant est également tenu de mettre en place une **surveillance permanente** de l'état technique général de la commande (défauts et dommages visibles de l'extérieur de même que modifications du comportement).

840 D
NCU 571840 D
NCU 572
NCU 573

FM-NC



810D

Maintenance

Les réparations doivent être effectuées uniquement par des **personnes spécialement formées et qualifiées** conformément aux indications qui figurent dans les instructions d'entretien et de maintenance. Observer tous les règlements de sécurité en vigueur.



Remarques complémentaires :

Cas d'utilisation **non conformes, excluant toute responsabilité du constructeur** :

Toute utilisation ne respectant pas les indications ci-dessus ou sortant du cadre de celles-ci.

Si la commande **n'est pas en parfait état technique** ou n'est pas utilisée en respectant les règles de sécurité, en tenant compte de la sécurité et des dangers et en observant toutes les instructions du manuel de mise en oeuvre.

S'il n'est pas remédié, **avant** la mise en service de la commande, aux dérangements susceptibles de nuire à la sécurité.

Toute modification, tout pontage ou toute mise hors fonction d'équipements de la commande qui servent à garantir un parfait fonctionnement, une utilisation sans restriction de même que la sécurité active et passive.



Des **dangers imprévisibles** sont possibles pour :

- les personnes,
- la commande, la machine et les autres biens de l'entreprise et de l'exploitant.

Composition de la documentation840 D
NCU 571840 D
NCU 572
NCU 573

FM-NC



810D

Programmation CN flexible

1.1 Variables et paramètres de calcul.....	1-20
1.2 Définition des variables.....	1-23
1.3 Définition de tableaux.....	1-27
1.4 Programmation indirecte.....	1-33
1.5 Affectations.....	1-35
1.6 Opérateurs/Fonctions de calcul.....	1-36
1.7 Opérateurs relationnels et opérateurs logiques.....	1-38
1.8 Priorité des opérateurs.....	1-41
1.9 Conversion du type de variable.....	1-42
1.10 Opérations sur les chaînes de caractères.....	1-43
1.11 Sauts de programme inconditionnels.....	1-53
1.12 Sauts de programme conditionnels.....	1-55
1.13 Instruction CASE.....	1-57
1.14 Structures de contrôle.....	1-59
1.15 Coordination de programmes.....	1-64
1.16 Routine d'interruption.....	1-69
1.17 Permutation d'axe, permutation de broche.....	1-77
1.18 Axes géométriques permutables, GEOAX.....	1-81
1.19 Répétition de parties de programme (à partir du logiciel 4.3).....	1-86

1.1 Variables et paramètres de calcul



840 D
NCU 571



840 D
NCU 572



FM-NC



810D

NCU 573

1.1 Variables et paramètres de calcul



Fonction

En utilisant des variables à la place de valeurs fixes, vous pouvez flexibiliser un programme. Vous pouvez ainsi le faire réagir à des signaux, à des valeurs de mesure par exemple, ou en utilisant des variables comme valeurs de consigne, vous pouvez aussi exploiter un même programme pour des géométries différentes.

En combinant les variables et les sauts de programme, le programmeur habile saura mettre en place une programméthèque d'une grande flexibilité et réduire de façon conséquente le travail de programmation.



Types de variables

La commande distingue 3 types de variables :

Variables utilisateur	Variables dont l'identification et le type sont définis par l'utilisateur, par. ex. paramètres de calcul.
Paramètres de calcul	Variables de calcul particulières, prédéterminées, caractérisées par l'adresse R suivie d'un numéro. Les variables de calcul prédéterminées sont du type REAL.
Variables système	Variables mises à disposition par la commande, qui peuvent être exploitées dans le programme (édition, lecture). Les variables système donnent accès aux décalages d'origine, aux corrections d'outil, aux valeurs instantanées, aux relevés métrologiques des axes, aux états de la commande, etc. (Signification des variables système : voir en annexe)

840 D
NCU 571840 D
NCU 572
NCU 573

FM-NC



810D



Types de variables

Type	Signification	Plage de valeurs
INT	nombres entiers avec signe	$\pm(2^{31} - 1)$
REAL	nombres réels (nombres rationnels à point décimal, LONG REAL selon IEEE)	$\pm(10^{-300} \dots 10^{+300})$
BOOL	valeurs logiques : TRUE (1) et FALSE (0)	1, 0
CHAR	1 caractère ASCII, selon code	0 ... 255
STRING	chaîne de caractères, nombre de caractères entre crochets [...], 200 caractères maximum	suite de valeurs 0 ... 255
AXIS	uniquement des noms d'axe (adresses d'axe)	tous les descripteurs d'axe et de broche présents dans le canal
FRAME	indications géométriques pour translation, rotation, échelle, fonction miroir (voir chapitre 4).	



Variables de calcul

En version standard, 100 variables de calcul du type REAL sont disponibles sous l'adresse R, sans autre définition.



Le nombre exact des variables de calcul (maximum 1000) est déterminé par les paramètres machine.

Exemple : R10=5

Variables système

La commande met à disposition des variables système qui sont exploitables dans l'ensemble des programmes en cours d'exécution.

Les variables système fournissent des valeurs de type défini. Cependant, elles ne sont en partie pas traduisibles.

1.1 Variables et paramètres de calcul



840 D
NCU 571



840 D
NCU 572



FM-NC



810D

NCU 573



Pour bien les identifier, le nom des variables système commence toujours par le caractère "\$".
Puis suivent les désignations spécifiques.

Vue d'ensemble des types de variables système

1e lettre	Signification
\$M	paramètres machine
\$S	données de réglage
\$T	paramètres de gestion des outils
\$P	valeurs programmées
\$A	valeurs courantes
\$V	données de maintenance

2e lettre	Signification
N	globale pour NCK
C	spécifique à un canal
A	spécifique à un axe

Exemple : \$AA_IM

signifie : valeur réelle axiale courante dans le système de coordonnées machine.

840 D
NCU 571840 D
NCU 572
NCU 573

FM-NC



810D

1.2 Définition des variables



Variables définies par l'utilisateur

Outre les variables prédéfinies, le programmeur peut définir ses propres variables et leur affecter des valeurs.

Les variables locales ne sont valables que dans le programme, dans lequel elles ont été définies.

Les variables globales sont valables dans tous les programmes.

Nom de variable

Un nom de variable se compose de 32 caractères au maximum. Les deux premiers caractères sont impérativement une lettre ou un caractère de soulignement.

Le caractère \$ n'est pas autorisé pour les variables utilisateur, car il est réservé aux variables système.



Programmation

```

DEF INT nom
OU DEF INT nom=valeur

DEF REAL nom
OU DEF REAL nom1,nom2=3,nom4
OU DEF REAL nom[indice de tableau1, indice
de tableau2]

DEF BOOL nom

DEF CHAR nom
OU DEF CHAR nom[indice de tableau] =
("A", "B", ...)

DEF STRING[longueur de chaîne de
caractères] nom

DEF AXIS nom
OU DEF AXIS nom[indice de tableau]

DEF FRAME nom
  
```

1.2 Définition des variables

840 D
NCU 571840 D
NCU 572

FM-NC

810D
NCU 573

Lorsqu'au moment de la définition des variables, on n'affecte pas de valeur à l'une d'entre elles, le système lui attribue la valeur 0.

Avant leur utilisation, les variables sont à définir au début du programme. La définition doit s'inscrire dans un bloc séparé ; un même bloc n'autorise que la définition d'une seule variable.



Signification

INT	type de variable : integer, c'est-à-dire nombre entier
REAL	type de variable : real, c'est-à-dire nombre rationnel à point décimal
BOOL	type de variable : bool, c'est-à-dire 1 ou 0 (TRUE (vrai) ou FALSE (faux))
CHAR	type de variable : char, c'est-à-dire caractère conforme au code ASCII (0 à 255)
STRING	type de variable : string, c'est-à-dire chaîne de caractères
AXIS	type de variable : Axis, c'est-à-dire adresses d'axe et de broche
FRAME	type de variable : FRAME, c'est-à-dire indications géométriques
nom	Nom de variable



Exemple de programmation

Type de variable INT

DEF INT NOMBRE	Une variable du type INT est créée sous le nom NOMBRE. Préréglage par le système avec 0.
----------------	--

DEF INT NOMBRE=7	Une variable du type INT est créée sous le nom NOMBRE. La variable possède la valeur initiale 7.
------------------	--

Type de variable REAL

DEF REAL PROFONDEUR	Une variable du type REAL est créée sous le nom PROFONDEUR. Attribution de la valeur zéro par le système (0.0).
---------------------	---

DEF REAL PROFONDEUR=6.25	Une variable du type REAL est créée sous le nom PROFONDEUR. La variable possède la valeur initiale 6.25.
--------------------------	--

DEF REAL PROFONDEUR=3.1, LONGUEUR=2, NOMBRE	On peut définir plusieurs variables dans une ligne.
--	---

840 D
NCU 571840 D
NCU 572
NCU 573

FM-NC



810D

Type de variable BOOL

```
DEF BOOL SI_TROP
```

Une variable du type BOOL est créée sous le nom SI_TROP.

Attribution de la valeur zéro par le système (FALSE).

```
DEF BOOL SI_TROP=1 ou
DEF BOOL SI_TROP=TRUE ou
DEF BOOL SI_TROP=FALSE
```

Une variable du type BOOL est créée sous le nom SI_TROP.

Type de variable CHAR

```
DEF CHAR GUSTAVE_1=65
```

Possibilité d'affecter aux variables de type CHAR : soit un code pour le caractère ASCII correspondant,

```
DEF CHAR GUSTAVE_1="A"
```

soit directement le caractère ASCII (le code 65 correspond à la lettre "A").

Type de variable STRING

```
DEF STRING[ 6 ] EXEMPLE_1="DEBUT"
```

Les variables du type STRING peuvent contenir une chaîne de caractères. Le nombre de caractères figure entre crochets, après le type de variable.

Type de variable AXIS

```
DEF AXIS NOM_AXE=(X1)
```

La variable du type AXIS porte le nom NOM_AXE et contient la désignation d'un axe du canal, ici X1 (les adresses avec extension s'écrivent entre parenthèses).

Type de variable FRAME

```
DEF FRAME OBLIQUE_1
```

La variable du type FRAME porte le nom OBLIQUE_1.

1.2 Définition des variables



840 D
NCU 571



840 D
NCU 572
NCU 573



FM-NC



810D



Remarques complémentaires

Une variable du type AXIS accepte les descripteurs d'axe et de broche d'un canal.

Remarque :

les désignations d'axe à adresse étendue s'écrivent entre parenthèses.



Exemple de programmation avec des variables locales

```
DEF INT COMPTEUR
BOUCLE : G0 X... ;boucle
COMPTEUR=COMPTEUR+1
IF COMPTEUR<50 GOTOB BOUCLE
M30
```



Exemple de programmation

Interrogation d'existence d'un axe géométrique

```
DEF AXIS ABSCISSE ;1er axe géométrique
IF ISAXIS(1) == FALSE GOTOF PLUSLOIN
ABSCISSE = $P_AXN1
...
PLUSLOIN :
```

Programmation indirecte de broche

```
DEF AXIS BROCHE
BROCHE=(B1)
OVRA[BROCHE]=80 ;correction de vitesse de rotation de
broche = 80%
BROCHE=(B3)
...
```

840 D
NCU 571840 D
NCU 572
NCU 573

FM-NC



810D

1.3 Définition de tableaux



Programmation

```

DEF CHAR NOM[n,m]
DEF INT NOM[n,m]
DEF REAL NOM[n,m]
DEF AXIS NOM[n,m]
DEF FRAME NOM[n,m]
DEF STRING[longueur de chaîne de caractères] NOM[m]
DEF BOOL[n,m]

```



Signification

INT NOM[n,m]	Type de variable (CHAR, INTEGER, REAL, AXIS, FRAME, BOOL)
REAL NOM[n,m]	n = taille du tableau dans la 1e dimension m = taille du tableau dans la 2e dimension
DEF STRING[longueur de chaîne de caractères] NOM[m]	Le type de données STRING ne peut être défini qu'avec un tableau unidimensionnel
NOM	Nom de variable

Jusqu'au logiciel SW2 :

Le comportement qui vient d'être décrit est valable jusqu'au logiciel SW2, sachant que la capacité de mémoire requise pour le type BOOL est identique à celle requise pour le type CHAR.

Jusqu'au logiciel SW3 :

La taille maximale d'un tableau est réglée par le biais des paramètres machine.

Type	Besoin en mémoire par élément de tableau
BOOL	1 octet
CHAR	1 octet
INT	4 octets
REAL	8 octets
STRING	longueur de chaîne de caractères + 1
FRAME	~ 400 octets, dépend du nombre d'axes
AXIS	4 octets

La taille maximale des tableaux détermine la taille des blocs de mémoire dans lesquels est gérée la mémoire des variables. Il convient de ne pas la régler sur une valeur trop élevée.

Version standard : 812 octets

1.3 Définition de tableaux



840 D
NCU 571



840 D
NCU 572



FM-NC



810D
NCU 573

Si vous ne définissez pas de grands champs, choisissez : 256 octets.

A partir du logiciel SW4 :

Un tableau peut être plus grand qu'un bloc de mémoire. Choisissez la valeur du PM pour la taille du bloc de façon à éviter, sauf exception, la fragmentation des tableaux.

Version standard : 256 octets

Le besoin en mémoire par élément est identique à celui requis dans SW 3 (voir plus haut).

Exemple pour la définition de tableaux de type BOOL :

Les données utilisateur globales contiennent les paramètres machine de l'AP pour la mise en/hors marche de la commande.

Remarques complémentaires

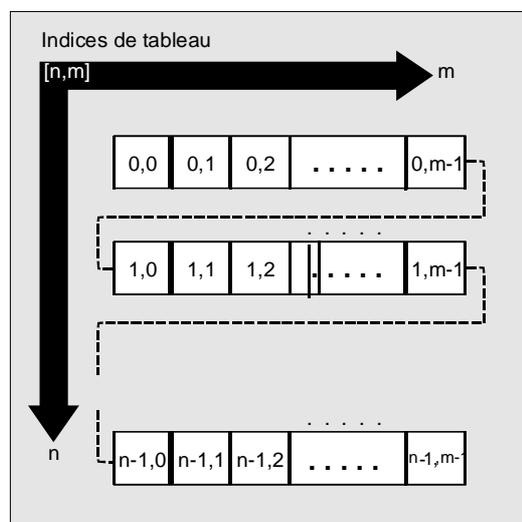
On peut définir des tableaux avec 2 dimensions au maximum.

Les tableaux avec des variables du type STRING ne peuvent être qu'unidimensionnels. La longueur de la chaîne de caractères est spécifiée à la suite du type de données STRING.

Indice de tableau

L'indice de tableau permet d'accéder aux éléments d'un tableau. Avec cet indice, on peut affecter des valeurs aux éléments du tableau ou bien lire ces valeurs.

Le premier élément du tableau commence avec l'indice [0,0] ; pour un tableau de taille [3,4] par exemple, l'indice maximal possible est [2,3].



840 D
NCU 571840 D
NCU 572
NCU 573

FM-NC



810D



Dans l'exemple ci-dessus, les valeurs d'initialisation ont été choisies de sorte à représenter en même temps l'indice de l'élément de tableau. Ceci fait apparaître l'ordre des éléments du tableau.



Initialisation de tableaux

Des valeurs d'initialisation peuvent être affectées aux éléments de tableau pendant l'exécution d'un programme ou plus tôt encore, dès la définition du tableau.

Dans un tableau bidimensionnel, l'indice de droite du tableau est incrémenté en premier lieu.



Initialisation avec listes de valeurs, SET

1. Possibilités fournies lors de la définition du tableau

```
DEF type VARIABLE = SET(VALEUR)
DEF type TABLEAU[n,m] = SET(VALEUR,valeur,...)
```

ou :

```
DEF type VARIABLE = valeur
DEF type TABLEAU[n,m] = (valeur, valeur, ...)
```

- Il y a autant d'éléments de tableau attribués que de valeurs d'initialisation programmées.
- La valeur 0 est automatiquement affectée aux éléments de tableau sans valeur (lacunes dans la liste de valeurs).
- Les lacunes dans la liste de valeurs ne sont pas admises pour les variables de type AXIS.
- Si le nombre de valeurs programmées est supérieur à celui des éléments de tableaux résiduels disponibles, le système déclenche une alarme.

Exemple :

```
DEF REAL TABLEAU[2,3]=(10, 20, 30, 40)
```



Lors de la définition de tableaux, SET est facultatif.

1.3 Définition de tableaux



840 D
NCU 571



840 D
NCU 572



FM-NC



810D
NCU 573

2. Possibilités fournies en cours d'exécution du programme

```
TABLEAU[n,m]= SET(valeur, valeur,  
valeur,...)
```

```
TABLEAU[n,m]=SET(expression,  
expression, expression,...)
```

- L'initialisation s'effectue comme pour la définition.
- Il est également possible, dans ce cas, d'utiliser des expressions comme valeurs.
- L'initialisation débute par les indices de tableau programmés. De cette façon, on peut également affecter des valeurs à des tableaux partiels.

Exemple :

Affectation d'expressions

```
DEF INT TABLEAU[5, 5]  
TABLEAU[0,0] = SET(1, 2, 3, 4, 5)  
TABLEAU[2,3] = SET(VARIABLE, 4*5.6)
```

L'indice d'axe n'est pas traité dans le cas de variables d'axe :

Exemple :

Initialisation sur une ligne

```
$MA_AX_VELO_LIMIT[1, AX1] = SET(1.1, 2.2, 3.3)
```

Correspond à :

```
$MA_AX_VELO_LIMIT[1,AX1] = 1.1  
$MA_AX_VELO_LIMIT[2,AX1] = 2.2  
$MA_AX_VELO_LIMIT[3,AX1] = 3.3
```

840 D
NCU 571840 D
NCU 572
NCU 573

FM-NC



810D



Initialisation avec valeurs identiques, REP

1. Possibilités fournies lors de la définition des tableaux

```
DEF type TABLEAU[n,m] = REP(valeur)
```

La même valeur (constante) est affectée à tous les éléments du tableau.



Il n'est pas possible d'initialiser des variables de type FRAME.

Exemple :

```
DEF REAL TABLEAU5[10,3] = REP(9.9)
```

2. Possibilités fournies en cours d'exécution du programme

```
TABLEAU[n,m] = REP(valeur)
```

```
TABLEAU[n,m] = REP(expression)
```

- Il est également possible, dans ce cas, d'utiliser des expressions comme valeurs.
- Tous les éléments du tableau sont initialisés avec la même valeur.
- L'initialisation débute par les indices de tableau programmés. De cette façon, on peut également affecter des valeurs à des tableaux partiels.



Les variables de type FRAME sont admises et peuvent ainsi être initialisées très facilement.

Exemple :

Initialisation de tous les éléments avec la même valeur

```
DEF FRAME FRM[10]
```

```
FRM[5] = REP(CTTRANS (X,5))
```

1.3 Définition de tableaux



840 D
NCU 571



840 D
NCU 572



FM-NC



810D
NCU 573



Exemple de programmation

Initialisation de tableaux de variables complets.

La figure représente chaque fois l'affectation courante.

```
N10 DEF REAL TABLEAU1[10,3] = SET(0, 0, 0, 10, 11, 12, 20, 20, 20, 30,
30, 30, 40, 40, 40,)
```

```
N20 TABLEAU1[0,0] = REP(100)
```

```
N30 TABLEAU1[5,0] = REP(-100)
```

```
N40 TABLEAU1[0,0] = SET(0, 1, 2, -10, -11, -12, -20, -20, -20, -30, , , ,
-40, -40, -50, -60, -70)
```

```
N50 TABLEAU1[8,1] = SET(8.1, 8.2, 9.0, 9.1, 9.2)
```

Indices de tableaux

[1,2]	N10 : initialisation lors de la définition			N20/N30 : initialisation avec des valeurs identiques			N40/N50 : initialisation avec des valeurs différentes		
	0	1	2	0	1	2	0	1	2
0	0	0	0	100	100	100	0	1	2
1	10	11	12	100	100	100	-10	-11	-12
2	20	20	20	100	100	100	-20	-20	-20
3	30	30	30	100	100	100	-30	0	0
4	40	40	40	100	100	100	0	-40	-40
5	0	0	0	-100	-100	-100	-50	-60	-70
6	0	0	0	-100	-100	-100	-100	-100	-100
7	0	0	0	-100	-100	-100	-100	-100	-100
8	0	0	0	-100	-100	-100	-100	8.1	8.2
9	0	0	0	-100	-100	-100	9.0	9.1	9.2
	Les éléments [5,0] à [9,3] du tableau sont initialisés avec la valeur par défaut (0.0).						Les éléments [3,1] à [4,0] du tableau sont initialisés avec la valeur par défaut (0.0) ; les éléments [6,0] à [8,0] ne sont pas modifiés.		

840 D
NCU 571840 D
NCU 572
NCU 573

FM-NC



810D

1.4 Programmation indirecte



La programmation indirecte permet d'exploiter les programmes de façon très universelle. Dans ce cas, l'adresse indiquée est remplacée par une variable de type approprié



Toutes les adresses sont paramétrables, sauf :

- N – numéro de bloc
- G – fonction G
- L – sous-programme

Les adresses réglables n'autorisent pas de programmation indirecte.

Exemple : X[1] au lieu de X1 n'est pas autorisé.



Programmation

ADRESSE [INDICE]



Exemples de programmation

Broche

S1=300

Programmation directe

DEF INT VITBRO=1
S[VITBRO]=300

Programmation indirecte :

Vitesse de rotation de 300 tr/min. pour la broche, dont le numéro est consigné dans la variable VITBRO (dans cet exemple : 1).

Avance

FA[U]=300

Programmation directe

DEF AXIS AXVAR2=U
FA[AXVAR2]=300

Programmation indirecte :

Avance d'un axe de positionnement, dont l'adresse est consignée dans la variable du type AXIS sous le nom AXVAR2.

Valeur de mesure

\$AA_MM[X]

Programmation directe

DEF AXIS AXVAR3=X
\$AA_MM[AXVAR3]

Programmation indirecte :

Valeur mesurée en coordonnées machine suivant l'axe, dont le nom est consigné dans la variable AXVAR3.

1.4 Programmation indirecte

840 D
NCU 571840 D
NCU 572

FM-NC



810D

NCU 573

Élément de tableau

```
DEF INT TABLEAU1[4,5]
```

Programmation directe

```
DEFINE DIM1 AS 4
```

```
DEFINE DIM2 AS 5
```

```
DEF INT TABLEAU[DIM1,DIM2]
```

```
TABLEAU[DIM1-1,DIM2-1]=5
```

Programmation indirecte :

Les dimensions de tableau sont à indiquer comme valeurs fixes.

Instruction d'axe avec variables d'axe

```
X1=100 X2=200
```

Programmation directe

```
DEF AXIS AXVAR1 AXVAR2
```

```
AXVAR1=(X1) AXVAR2=(X2)
```

```
AX[AXVAR1]=100 AX[AXVAR2]=200
```

Programmation indirecte :

Définition des variables

Affectation des noms d'axes. Déplacement des axes consignés dans les variables aux positions 100, voire 200.

Paramètre d'interpolation avec variables d'axe

```
G2 X100 I20
```

Programmation directe

```
DEF AXIS AXVAR1=X
```

```
G2 X100 IP[AXVAR1]=20
```

Programmation indirecte :

Définition et affectation du nom d'axe

Programmation indirecte du centre

Remarques complémentaires

Les paramètres R aussi peuvent être compris comme des tableaux unidimensionnels en notation abrégée (R10 correspond à R[10]).

840 D
NCU 571840 D
NCU 572
NCU 573

FM-NC



810D

1.5 Affectations

Des valeurs d'un type approprié peuvent être assignées dans le programme aux variables/paramètres de calcul.



L'affectation doit se faire dans un bloc séparé ; ce bloc peut comporter plusieurs affectations.

L'affectation à des adresses d'axe (instructions de déplacement) réclame un bloc séparé par rapport aux affectations de variables.



Exemple de programmation

```
R1=10.518 R2=4 VARI1=45
X=47.11 Y=R2
```

Affectation d'une valeur numérique

```
R1=R3 VARI1=R4
```

Affectation d'une variable de type approprié

```
R4=-R5 R7=-VARI8
```

Affectation de valeur avec signe opposé
(uniquement autorisé pour les types INT et REAL)



Affectation de valeur à une variable de type STRING

A l'intérieur d'une chaîne de caractères CHAR ou STRING, une distinction est faite entre les majuscules et les minuscules.

Si les caractères ...'... ou ..."... doivent faire partie intégrante de la chaîne de caractères, cela est à inscrire entre '...'.
...

Exemple :

```
MSG("Viene lavorata l' 'ultima  
figura")
```

devient à l'écran 'Viene lavorata l'ultima figura'.

Des caractères non représentables peuvent être admis dans la chaîne de caractères sous forme de constantes binaires ou hexadécimales.

1.6 Opérateurs/Fonctions de calcul

840 D
NCU 571840 D
NCU 572

FM-NC

810D
NCU 573

1.6 Opérateurs/Fonctions de calcul



Les fonctions de calcul sont applicables en priorité aux paramètres R et aux variables (ou constantes et fonctions) de type REAL. Les types INT et CHAR sont également admis.



Quand vous utilisez des opérateurs/fonctions de calcul, respectez la notation arithmétique usuelle.

Les priorités de traitement sont fixées par des parenthèses. Les fonctions trigonométriques et leurs inverses emploient la notation en degrés (angle droit = 90°).



Opérateurs/Fonctions de calcul

+	addition
-	soustraction
*	multiplication
/	division Attention : (type INT)/(type INT)=(type REAL); exemple : 3/4 = 0.75
DIV	division, uniquement pour type de variable INT Attention : (type INT)DIV(type INT)=(type INT); exemple: 3 DIV 4 = 0
MOD	division modulo (INT ou REAL) fournit le reste d'une division INT, par ex. 3 MOD 4=3
:	opérateur de concaténation (pour variables de frame)
Sin()	sinus
COS()	cosinus
TAN()	tangente
ASIN()	arc sinus
ACOS()	arc cosinus
ATAN2(,)	arc tangente 2
SQRT()	racine carrée
ABS()	valeur absolue
POT()	puissance 2 (carré)
TRUNC()	partie entière
ROUND()	arrondir à nombre entier
LN()	logarithme naturel
EXP()	fonction exponentielle
CTTRANS()	décalage
CROT()	rotation

840 D
NCU 571840 D
NCU 572
NCU 573

FM-NC



810D

CSCALE ()	changement d'échelle
------------	----------------------

CMIRROR ()	fonction miroir
-------------	-----------------



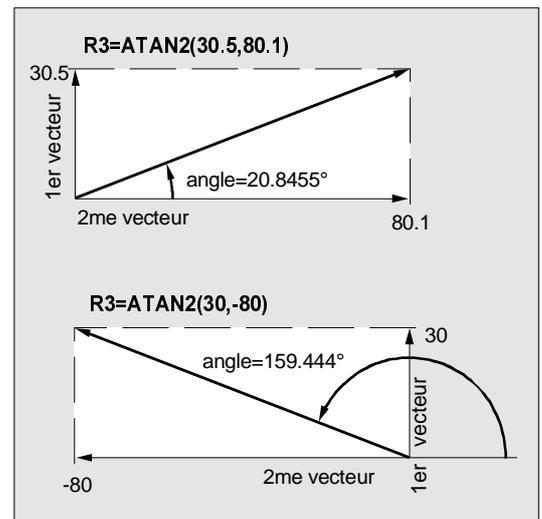
Exemple de programmation

R1=R1+1	nouvel R1 = ancien R1 +1
R1=R2+R3 R4=R5-R6 R7=R8*R9	
R10=R11/R12 R13=SIN(25.3)	
R14=R1*R2+R3	la multiplication est prioritaire
R14=(R1+R2)*R3	les parenthèses sont traitées en priorité
R15=SQRT(POT(R1)+POT(R2))	les parenthèses internes sont levées d'abord R15 = racine carrée de (R1 ² +R2 ²)
RESFRAME= FRAME1:FRAME2	L'opérateur de concaténation combine des frames en un frame résultant ou assigne des valeurs aux composantes de frame
FRAME3=CTTRANS(...):CROT(...)	



Fonction de calcul ATAN2(,)

A partir de deux vecteurs orthogonaux, la fonction calcule l'angle du vecteur cumulatif. Le résultat s'inscrit dans la plage des quatre quadrants ($-180^\circ < \theta < +180^\circ$). La base de la référence angulaire est toujours représentée par la 2^{me} valeur dans le sens positif.



1.7 Opérateurs relationnels et opérateurs logiques



840 D
NCU 571



840 D
NCU 572
NCU 573



FM-NC



810D

1.7 Opérateurs relationnels et opérateurs logiques



Opérateurs relationnels

Les opérateurs relationnels s'appliquent aux variables des types CHAR, INT, REAL et BOOL.

Dans le cas du type CHAR, la comparaison porte sur la valeur de code.

Possibilités en cas de variables des types STRING, AXIS et FRAME: == et <>.

Le résultat d'opérations relationnelles est toujours de nature booléenne.

Les opérations relationnelles peuvent servir par exemple à formuler une condition de saut dans le programme. Elles s'appliquent également aux expressions complexes.



Signification des opérateurs relationnels

==	égal
<>	différent
>	supérieur
<	inférieur
>=	supérieur ou égal
<=	inférieur ou égal
<<	concaténation de chaînes de caractères (pas pour 810D)



Exemple de programmation

```
IF R10>=100 GOTO DESTINATION
OU
R11=R10>=100
IF R11 GOTO DESTINATION
```

Le résultat de la comparaison R10>=100 est d'abord mémorisé dans R11.

840 D
NCU 571840 D
NCU 572
NCU 573

FM-NC



810D



Opérateurs logiques

Les opérateurs logiques servent à la combinaison des variables logiques.

Les opérateurs AND, OR, NOT et XOR ne s'appliquent qu'aux variables du type BOOL. Par l'intermédiaire de conversions implicites de types, elles s'appliquent également aux données des types CHAR, INT et REAL.

Entre les opérandes booléens et les opérateurs doivent figurer des espaces.

Dans le cas des opérations logiques (booléennes), la convention suivante s'applique aux données du type BOOL, CHAR, INT et REAL :

valeur 0 correspond à FALSE

valeur différente de 0 correspond à TRUE



Signification des opérateurs logiques

AND	ET
OR	OU
NOT	négation
XOR	OU exclusif

Dans les expressions arithmétiques, l'ordre de traitement de tous les opérateurs peut être fixé par des parenthèses et peut, de ce fait, s'écarter des règles normales de priorité.



Exemple de programmation

```
IF (R10<50) AND ($AA_IM[X]>=17.5) GOTOF DESTINATION
```

```
IF NOT R10 GOTOB DEPART
```

L'opérateur NOT ne se rapporte qu'à un seul opérande.

1.7 Opérateurs relationnels et opérateurs logiques



840 D
NCU 571



840 D
NCU 572
NCU 573



FM-NC



810D



Opérateurs logiques bit à bit

Les variables des types CHAR et INT permettent également d'effectuer des opérations logiques bit à bit. Le cas échéant, il est procédé à une conversion automatique de type.



Signification des opérateurs logiques bit à bit

B_AND	ET bit à bit
B_OR	OU bit à bit
B_NOT	négation bit à bit
B_XOR	OU exclusif bit à bit



L'opérateur B_NOT ne se rapporte qu'à un seul opérande ; celui-ci figure après l'opérateur.



Exemple de programmation

```
IF $MC_RESET_MODE_MASK B_AND 'B10000' GOTOF ACT_PLANE
```

840 D
NCU 571840 D
NCU 572
NCU 573

FM-NC



810D

1.8 Priorité des opérateurs



Priorité des opérateurs

A chaque opérateur est affectée une priorité. Lors du traitement d'une expression, les opérateurs au degré de priorité le plus élevé sont appliqués en premier lieu. En cas d'opérateurs d'égale priorité, le traitement s'opère de gauche à droite.

Dans les expressions arithmétiques, l'ordre de traitement de tous les opérateurs peut être fixé par des parenthèses et peut, de ce fait, s'écarter des règles normales de priorité.

Ordre des opérateurs

(de la priorité la plus élevée à la moins élevée)

1.	NOT, B_NOT	négation, négation bit à bit
2.	*, /, DIV, MOD	multiplication, division
3.	+, -	addition, soustraction
4.	B_AND	ET bit à bit
5.	B_XOR	OU exclusif bit à bit
6.	B_OR	OU bit à bit
7.	AND	ET
8.	XOR	OU exclusif
9.	OR	OU
10.	<<	concaténation de chaînes de caractères, type de résultat STRING
11.	==, <>, >, <, >=, <=	opérateurs relationnels



L'opérateur de concaténation ":" pour frames ne doit pas figurer avec d'autres opérateurs dans une même expression. Cet opérateur ne nécessite donc pas d'ordre de priorité.

1.9 Conversion du type de variable



840 D
NCU 571



840 D
NCU 572



FM-NC



810D
NCU 573

1.9 Conversion du type de variable



Conversion du type de variable lors de l'affectation

La valeur numérique constante, la variable ou l'expression assignée à une variable doit être compatible avec le type de cette variable. Si cette condition n'est pas remplie, le type est automatiquement converti au moment de l'affectation.

Possibilités de conversion des types de variables

de \ vers	REAL	INT	BOOL	CHAR	STRING	AXIS	FRAME
REAL	oui	oui*	oui ¹⁾	oui*	–	–	–
INT	oui	oui	oui ¹⁾	oui ²⁾	–	–	–
BOOL	oui	oui	oui	oui	oui	–	–
CHAR	oui	oui	oui ¹⁾	oui	oui	–	–
STRING	–	–	oui ⁴⁾	oui ³⁾	oui	–	–
AXIS	–	–	–	–	–	oui	–
FRAME	–	–	–	–	–	–	oui

* En cas de conversion du type REAL en type INT, une partie décimale ≥ 0.5 est arrondie par excès ; dans le cas contraire, elle est arrondie par défaut (voir fonction ROUND).

¹⁾ Une valeur $\neq 0$ équivaut à TRUE, la valeur $= 0$ équivaut à FALSE

²⁾ Quand la valeur se situe dans la plage numérique admise

³⁾ Dans le cas d'un seul caractère

⁴⁾ Longueur de chaîne de caractères 0 = FALSE, sinon TRUE



Un message d'erreur est émis lorsqu'une valeur de conversion est supérieure à la plage cible.



Remarques complémentaires

Lorsque des types différents figurent dans une même expression, leur adaptation s'effectue automatiquement.

840 D
NCU 571840 D
NCU 572
NCU 573

FM-NC



810D

1.10 Opérations sur les chaînes de caractères



Aperçu

En dehors des opérations classiques "affectation" et "comparaison" décrites dans ce chapitre, il existe d'autres possibilités de manipulation des chaînes de caractères :



Signification

Conversion de type vers STRING

STRING_ERG = <<type quelc. ¹⁾	Type de résultat : STRING
--	---------------------------

STRING_ERG = AXSTRING (AXIS)	Type de résultat : STRING
------------------------------	---------------------------

Conversion de type depuis STRING :

BOOL_ERG = ISNUMBER (STRING)	Type de résultat : BOOL
------------------------------	-------------------------

REAL_ERG = NUMBER (STRING)	Type de résultat : REAL
----------------------------	-------------------------

AXIS_ERG = AXNAME (STRING)	Type de résultat : AXIS
----------------------------	-------------------------

Concaténation de chaînes de caractères :

type quelc. ¹⁾ << type quelc. ¹⁾	Type de résultat : STRING
--	---------------------------

Transformation en minuscules / majuscules :

STRING_ERG = TOUPPER (STRING)	Type de résultat : STRING
-------------------------------	---------------------------

STRING_ERG = TOLOWER (STRING)	Type de résultat : STRING
-------------------------------	---------------------------

Longueur de la chaîne de caractères :

INT_ERG = STRLEN (STRING)	Type de résultat : INT
---------------------------	------------------------

Recherche de caractères / de suites de caractères dans la chaîne:

INT_ERG = INDEX (STRING, CHAR)	Type de résultat : INT
--------------------------------	------------------------

INT_ERG = RINDEX (STRING, CHAR)	Type de résultat : INT
---------------------------------	------------------------

INT_ERG = MINDEX (STRING, STRING)	Type de résultat : INT
-----------------------------------	------------------------

INT_ERG = MATCH (STRING, STRING)	Type de résultat : INT
----------------------------------	------------------------

Sélection d'une chaîne partielle de caractères :

STRING_ERG = SUBSTR (STRING, INT)	Type de résultat : INT
-----------------------------------	------------------------

STRING_ERG = SUBSTR (STRING, INT, INT)	Type de résultat : INT
--	------------------------

Sélection d'un caractère individuel:

CHAR_ERG = STRINGVAR [IDX]	Type de résultat : CHAR
----------------------------	-------------------------

CHAR_ERG = STRINGFELD [IND-TABLEAU, IND_CARAC]	Type de résultat : CHAR
--	-------------------------

¹⁾ "type quelc." représente ici les types de variables INT, REAL, CHAR, STRING et BOOL.

1.10 Opérations sur les chaînes de caractères



840 D
NCU 571



840 D
NCU 572



FM-NC



810D
NCU 573

Signification particulière du caractère 0

Le caractère 0 est interprété de façon interne comme fin d'une chaîne de caractères.

Si un caractère est remplacé par le caractère 0, la chaîne de caractères est en conséquence raccourcie.

Exemple :

```
DEF STRING[20] STRG = "Axe . arrêté"
STRG[6] = "X"
```

```
;fournit le message "Axe X
arrêté"
```

```
MSG(STRG)
STRG[6] = 0
MSG(STRG)
```

```
;fournit le message "Axe"
```



1. Conversion des types de variables

On peut ainsi utiliser des variables de types différents comme parties intégrantes d'un message (MSG).

Conversion vers STRING

S'effectue implicitement quand on utilise l'opérateur << pour les types de données INT, REAL, CHAR et BOOL (cf. "Concaténation de chaînes de caractères").

Une valeur INT est convertie sous la forme normale lisible. Les valeurs REAL sont données avec un nombre de décimales pouvant aller jusqu'à 10.

Des variables de type AXIS peuvent être converties en type STRING avec la fonction AXSTRING.

Les variables de type FRAME ne sont pas convertibles.

Exemple :

```
MSG("Position:" << $AA_IM[X])
```

840 D
NCU 571840 D
NCU 572
NCU 573

FM-NC



810D

Conversion depuis STRING

Avec la fonction NUMBER, la conversion se fait de STRING vers REAL.

Si ISNUMBER fournit la valeur FALSE, une alarme est déclenchée dès que NUMBER est appelé avec le même paramètre.

Avec la fonction AXNAME une chaîne de caractères de type STRING peut être convertie en type AXIS. Une alarme est émise si le résultat ne peut être affecté à un descripteur d'axe existant.

Syntaxe :

BOOL_ERG = ISNUMBER (STRING)	Type de résultat : BOOL
REAL_ERG = NUMBER (STRING)	Type de résultat : REAL
STRING_ERG = AXSTRING (AXIS)	Type de résultat : STRING
AXIS_ERG = AXNAME (STRING)	Type de résultat : AXIS

Sémantique :

ISNUMBER (STRING) donne TRUE, quand la chaîne de caractères représente un nombre REAL valide selon les règles du langage. Cela permet de contrôler si la chaîne de caractères est convertible en un nombre valable.

NUMBER (STRING) donne en retour le nombre représenté par la chaîne comme valeur REAL.

AXSTRING (AXIS) transforme le descripteur d'axe indiqué en une chaîne de caractères.

AXNAME (STRING) transforme la chaîne de caractères indiquée en un descripteur d'axe.

Exemples :

DEF BOOL BOOL_ERG	
DEF REAL REAL_ERG	
DEF AXIS AXIS_ERG	
DEF STRING[32] STRING_ERG	
BOOL_ERG = ISNUMBER ("1234.9876Ex-7")	;maintenant : BOOL_ERG == TRUE
BOOL_ERG = ISNUMBER ("1234XYZ")	;maintenant : BOOL_ERG == FALSE
REAL_ERG = NUMBER ("1234.9876Ex-7")	;maintenant : REAL_ERG == 1234.9876Ex-7
STRING_ERG = AXSTRING (X)	;maintenant : STRING_ERG == "X"
AXIS_ERG = AXNAME ("X")	;maintenant : AXIS_ERG == X

1.10 Opérations sur les chaînes de caractères



840 D
NCU 571



840 D
NCU 572



FM-NC



810D
NCU 573



2. Concaténation de chaînes de caractères :

Cette fonctionnalité permet de composer une chaîne de caractères à partir de différentes composantes.

La concaténation est réalisée par l'opérateur : <<.

Cet opérateur possède, pour toutes les combinaisons des types de base, CHAR, BOOL, INT, REAL et STRING comme type de destination STRING. Une conversion éventuellement indispensable est entreprise selon les règles établies. Les types FRAME et AXIS ne peuvent pas être utilisés avec cet opérateur.

Syntaxe :

type quelc. << type quelc.

Type de résultat : STRING

Sémantique :

Les chaînes de caractères données (le cas échéant l'autre type transformé implicitement) sont concaténées ensemble.

Cet opérateur est également disponible en tant que variante unaire. Il est ainsi possible de réaliser explicitement une transformation de type vers STRING (pas valable pour FRAME et AXIS).

Syntaxe :

<< type quelc.

Type de résultat : STRING

Sémantique :

Le type spécifié est converti implicitement en type STRING.

C'est ainsi par exemple que vous pouvez créer un message ou un ordre à partir de listes de textes et y intégrer des paramètres (un nom de bloc par exemple) :

```
MSG ( STRG_TAB [ LOAD_IDX ] <<NOM_BLOC_DONNEES )
```

1.10 Opérations sur les chaînes de caractères



840 D
NCU 571



840 D
NCU 572
NCU 573



FM-NC



810D



Les résultats intermédiaires de la concaténation ne doivent pas dépasser la longueur maximale admise pour une chaîne de caractères.



Exemple de programmation

```
DEF INT IDX = 2
DEF REAL VALUE = 9.654
DEF STRING[20]STRG = "INDICE:2"
IF STRG == "Indice:" <<IDX GOTOF
NO_MSG
MSG ("Indice:" <<IDX <<"/valeur:" ;affichage : "indice: 2/valeur : 9.654"
<<VALUE)
NO_MSG:
```



3. Transformation en minuscules / majuscules :

Cette fonctionnalité permet d'harmoniser l'écriture au sein d'une chaîne de caractères.

Syntaxe :

STRING_ERG = TOUPPER	(STRING)	Type de résultat : STRING
STRING_ERG = TOLOWER	(STRING)	Type de résultat : STRING

Sémantique :

Toutes les lettres sont transformées soit en minuscules, soit en majuscules.

Exemple :

Etant donné qu'il est possible de rattacher des indications introduites par l'utilisateur, cette fonctionnalité permet d'obtenir une présentation uniformisée en lettres minuscules ou majuscules :

```
DEF STRING [29] STRG
...
IF "LEARN.CNC" == TOUPPER (STRG) GOTOF LOAD_LEARN
```



4. Longueur de la chaîne de caractères :

Cette fonctionnalité permet de déterminer la longueur d'une chaîne de caractères.

1.10 Opérations sur les chaînes de caractères



840 D
NCU 571



840 D
NCU 572
NCU 573



FM-NC



810D

Syntaxe :

```
INT_ERG = STRLEN (STRING)
```

Type de résultat : INT

Sémantique :

Il s'agit de la restitution du nombre de caractères différents de 0, comptés depuis le début de la chaîne de caractères.

Exemple :

Cela permet, par exemple, de déterminer la fin de la chaîne, grâce à la recherche d'un caractère individuel comme décrit ci-après :

```
IF(STRLEN (BLOC_NOM) > 10) GOTOF DEFAULT
```



5. Recherche de caractères / de suites de caractères dans la chaîne :

Cette fonctionnalité permet de rechercher des caractères isolés ou une suite de caractères dans une chaîne de caractères. Les résultats signalent à quel endroit de la chaîne de caractères se trouve le caractère / la suite de caractères qui fait l'objet de la recherche.

```
INT_ERG = INDICE (STRING, CHAR) Type de résultat : INT
```

```
INT_ERG = RINDICE (STRING, CHAR) Type de résultat : INT
```

```
INT_ERG = MINDEX (STRING, STRING) Type de résultat : INT
```

```
INT_ERG = MATCH (STRING, STRING) Type de résultat : INT
```

Sémantique :

Fonctions de recherche : Elles donnent la position dans la chaîne de caractères (premier paramètre) pour laquelle la recherche fut une réussite. Si le caractère/la suite de caractères n'est pas trouvé/e, la valeur -1 est restituée. Le premier caractère se trouve à la position 0.

1.10 Opérations sur les chaînes de caractères

840 D
NCU 571840 D
NCU 572
NCU 573

FM-NC



810D

INDEX	recherche dans le premier paramètre le caractère spécifié comme second paramètre (depuis le début).
RINDEX	recherche dans le premier paramètre le caractère spécifié comme second paramètre (depuis la fin).
MINDEX	correspond à INDEX, sauf qu'ici cela concerne une liste de caractères (comme chaîne de caractères), parmi lesquels est restitué l'indice du premier caractère trouvé.
MATCH	recherche une suite de caractères dans une chaîne de caractères.

Il est ainsi possible de décomposer des chaînes selon des critères précis, par exemple au niveau d'un caractère d'espace ou d'un séparateur de chemin ("/").

**Exemple de programmation**

Un exemple pour la décomposition d'une introduction en noms de bloc et noms de chemin d'accès :

```
DEF INT PFADIDX, PROGIDX
```

```
DEF STRING[26] INTRO
```

```
DEF INT LISTIDX
```

```
INTRO = "_N_MPF_DIR/_N_EXECUTE_MPF"
```

```
LISTIDX = MINDEX (INTRO, „M,N,O,P“)
                + 1
```

La valeur 3 est restituée dans LISTIDX, puisque „N“ est le premier caractère dans le paramètre INTRO, dans la liste d'options depuis le début.

```
PFADIDX = INDEX (INTRO, "/") +1
```

; ainsi : PFADIDX = 1

```
PROGIDX = RINDEX (INTRO, "/") +1
```

; ainsi : PROGIDX = 12

; la fonction SUBSTR expliquée dans l'étape suivante peut décomposer la variable INTRO suivant les composantes "chemin" et "bloc" :

```
VARIABLE = SUBSTR (INTRO, PFADIDX,
                  PROGIDX-PFADIDX-1)
```

fournit alors "_N_MPF_DIR"

```
VARIABLE = SUBSTR (INTRO, PROGIDX)
```

fournit alors "_N_EXECUTE_MPF"

1.10 Opérations sur les chaînes de caractères



840 D
NCU 571



840 D
NCU 572
NCU 573



FM-NC



810D



6. Sélection d'une chaîne partielle de caractères :

Cette fonctionnalité permet d'extraire une chaîne partielle de caractères d'une chaîne de caractères. Pour ce faire, on précise l'indice du premier caractère et, le cas échéant, la longueur désirée. Si l'information de longueur n'est pas précisée, c'est toute la chaîne de caractères restante qui est prise en compte.

STRING_ERG = SUBSTR (STRING, INT) Type de résultat : INT

STRING_ERG = SUBSTR (STRING, INT, INT) Type de résultat : INT

Sémantique :

Dans le premier cas, la chaîne partielle de caractères est restituée à partir de la position définie par le premier paramètre et jusqu'à la fin de la chaîne de caractères.

Dans le second cas, la chaîne restituée est limitée à la longueur indiquée par le troisième paramètre.

Si la position de départ se trouve après la fin de la chaîne, il y aura restitution d'une chaîne de blancs (" ").

Si la position de départ ou la longueur est négative, il y aura déclenchement d'une alarme.

Exemple :

DEF STRING [29] ERG

ERG = SUBSTR ("ACQUITTEMENT:
10 à 99", 10, 2) ;ainsi : ERG == "10"

1.10 Opérations sur les chaînes de caractères

840 D
NCU 571840 D
NCU 572
NCU 573

FM-NC



810D



7. Sélection d'un caractère individuel

Cette fonctionnalité permet de sélectionner des caractères individuels dans une chaîne. Cela est aussi bien valable pour l'accès en lecture que pour l'accès en écriture.

Syntaxe :

CHAR_ERG = STRINGVAR [IDX]	Type de résultat : CHAR
CHAR_ERG = STRINGTAB [IDX_TABLEAU, IDX_CARAC]	Type de résultat : CHAR

Sémantique :

Le caractère situé à l'emplacement spécifié dans la chaîne de caractères est lu/écrit. Si l'indication de position est négative ou plus grande que la chaîne de caractères, une alarme est déclenchée.

Exemple de messages :

Installation d'un descripteur d'axe dans une chaîne de caractères préétablie.

```
DEF STRING [50] MESSAGE = "Axe n a
atteint sa position"
MESSAGE [6] = "X"
MSG (MESSAGE)
```

;fournit le message "Axe X a atteint sa position"

1.10 Opérations sur les chaînes de caractères



840 D
NCU 571



840 D
NCU 572



FM-NC



810D

NCU 573

L'accès à un caractère individuel est possible uniquement dans les variables définies par l'utilisateur (données LUD et GUD).

De plus, dans le cas de l'appel d'un sous-programme, ce mode d'accès n'est possible que pour des paramètres de Type "Call-By-Value".

Exemples :

Accès à un caractère individuel dans un paramètre machine ou un paramètre système ... :

```
DEF STRING [50] STRG
DEF CHAR ACQUITTEMENT
...
STRG = $P_MMCA
ACQUITTEMENT = STRG [0] ;exploitation des composantes de
                          l'acquittement
```

Accès à un caractère individuel dans le paramètre de type "Call-By-Reference" :

```
DEF STRING [50] STRG
DEF CHAR CHR
EXTERN UP_CALL (VAR CHAR) ;paramètre de type "Call-By-Reference"!
...
CHR = STRG [5]
UP_CALL (CHR) ;Call-By-Reference
STRG [5] = CHR
```

840 D
NCU 571840 D
NCU 572
NCU 573

FM-NC



810D

1.11 Sauts de programme inconditionnels



Programmation

Etiquette :
 GOTOB ETIQUETTE
 ou
 GOTOF ETIQUETTE
 Label :



Signification

GOTOB	Instruction de saut avec destination en amont (vers début du programme)
GOTOF	Instruction de saut avec destination en aval (vers fin de programme)
ETIQUETTE	Destination (marquage interne au programme)
ETIQUETTE :	Destination du saut est suivi d'un deux-points



Fonction

En version standard, les programmes principaux, les sous-programmes, les cycles et les routines d'interruption traitent les blocs dans l'ordre de leur programmation.

Cet ordre peut être modifié par des sauts dans le programme.



Procédure

Dans un programme, des destinations de saut peuvent être définies par des noms choisis par l'utilisateur. En partant de points arbitraires à l'intérieur du même programme, les ordres GOTOF ou GOTOB permettent un branchement vers une destination de saut. L'exécution du programme se poursuit alors à l'instruction qui figure immédiatement après la destination du saut.

1.11 Sauts de programme inconditionnels



840 D
NCU 571



840 D
NCU 572
NCU 573



FM-NC



810D



Exemple de programmation

```

N10 ...
N20 GOTOF MARQUE_0                               Saut en avant jusqu'à MARQUE_0
N30 ...
N40 ...
N50 MARQUE_1: R1=R2+R3
N60 ...
N65 GOTOF MARQUE_2                               Saut en avant jusqu'à MARQUE_2
N70 ...
N80 ...
N90 MARQUE_0:
N100 ...
N110 GOTOB MARQUE_1                             Saut en arrière jusqu'à MARQUE_1
N115 ...
N120 MARQUE_2:
N130 ...

```



Remarques complémentaires

Le saut inconditionnel est à programmer dans un bloc séparé.

Dans le cas de programmes à sauts inconditionnels, l'instruction fin de programme M2/M30 ne doit pas nécessairement figurer en fin de programme.

840 D
NCU 571840 D
NCU 572
NCU 573

FM-NC



810D

1.12 Sauts de programme conditionnels



Programmation

ETIQUETTE :
 IF expression GOTOB ETIQUETTE
 OU
 IF expression GOTOF ETIQUETTE
 ETIQUETTE :



Signification des instructions

IF	Mot-clé de condition
GOTOB	Instruction de saut avec destination en amont (vers début de programme)
GOTOF	Instruction de saut avec destination en aval (vers fin de programme)
ETIQUETTE	Etiquette (marquage interne au programme);
ETIQUETTE :	Destination du saut est suivi d'un deux-points



Fonction

Des conditions de saut peuvent être formulées en utilisant les instructions IF. Le saut à la destination programmée ne s'effectue que si la condition de saut est remplie.



Procédure

La condition de saut admet tous les opérateurs relationnels et logiques (résultat : TRUE ou FALSE). Le saut dans le programme est réalisé si le résultat de cette opération est TRUE.

Une destination de saut ne peut être qu'un bloc à étiquette situé à l'intérieur du programme.



Un même bloc peut comporter plusieurs sauts conditionnels.

1.12 Sauts de programme conditionnels



840 D
NCU 571



840 D
NCU 572
NCU 573



FM-NC



810D

Exemples de conditions de saut

IF R1>R2 GOTOF MARQUE1	Si R1 supérieur à R2, saut jusqu'à MARQUE1
IF R7<=(R8+R9)*743 GOTOB MARQUE1	Condition sous forme d'expression complexe
IF R10 GOTOF MARQUE1	L'indication d'une variable (INT, REAL, BOOL ou CHAR) est également possible. Si la valeur de la variable est nulle (=FALSE), la condition n'est pas remplie ; à toutes les autres valeurs correspond TRUE.
IF R1==0 GOTOF MARQUE1 IF R1==1 -> -> GOTOF MARQUE2	Plusieurs conditions au sein d'un même bloc



Exemple de programmation

N40 R1=30 R2=60 R3=10 R4=11 R5=50 R6=20	Affectation des valeurs initiales
N41 MA1: G0 X=R2*COS(R1)+R5 -> -> Y=R2*SIN(R1)+R6	Calcul et affectation à une adresse d'axe
N42 R1=R1+R3 R4=R4-1	Indication de variables
N43 IF R4>0 GOTOB MA1	Instruction de saut
N44 M30	Fin de programme

840 D
NCU 571840 D
NCU 572

FM-NC

810D
NCU 573

1.13 Instruction CASE



Programmation

```
CASE (expression) OF constante1 GOTO# ETIQUETTE1 ... DEFAULT GOTO# ETIQUETTE#n
CASE (expression) OF constante1 GOTOB ETIQUETTE1 ... DEFAULT GOTOB ETIQUETTE#n
```



Signification des instructions

CASE	Mot-clé d'instruction de saut
GOTO#	Instruction de saut avec destination en aval (vers fin de programme)
GOTOB	Instruction de saut avec destination en amont (vers début de programme)
ETIQUETTE	Destination (marquage interne au programme)
ETIQUETTE :	Destination de saut suivi d'un deux-points
expression	Expression arithmétique
Constante	Constante du type INT
DEFAULT	Chemin du programme si aucune des constantes indiquées antérieurement n'est valide



Fonction

L'instruction CASE permet de réaliser des branchements distincts en fonction de la valeur courante de type INT.



Procédure

Selon la valeur que prend la constante vérifiée par l'instruction CASE, le programme réalise un branchement à l'endroit déterminé par la destination de saut.

1.13 Instruction CASE



840 D
NCU 571



840 D
NCU 572



FM-NC



810D

NCU 573

Au cas où l'expression arithmétique ne prend aucune des valeurs indiquées, la destination du saut est définie par l'instruction DEFAULT.

Au cas où l'instruction DEFAULT n'est pas programmée, le bloc qui suit l'instruction CASE devient destination du saut.



Exemple de programmation

Exemple 1

```
CASE(expression) OF 1 GOTOF ETIQUETTE1 2 GOTOF ETIQUETTE2 ... DEFAULT
GOTOF ETIQUETTEn
```

"1" et "2" sont des constantes possibles.

Si la valeur de l'expression = K1 (constante interne), sauter au bloc comportant ETIQUETTE1

Si la valeur de l'expression = K2 (constante interne), sauter au bloc comportant ETIQUETTE2

...

sinon sauter au bloc comportant ETIQUETTEn

Exemple 2

```
DEF INT VAR1 VAR2 VAR3
```

```
CASE(VAR1+VAR2-VAR3) OF 7 GOTOF MARQUE1 9 GOTOF MARQUE2 DEFAULT GOTOF
MARQUE3
```

```
MARQUE1 : G0 X1 Y1
```

```
MARQUE2 : G0 X2 Y2
```

```
MARQUE3 : G0 X3 Y3
```

840 D
NCU 571840 D
NCU 572
NCU 573

FM-NC



810D

1.14 Structures de contrôle



Signification

IF-ELSE-ENDIF	Choix entre 2 alternatives
LOOP-ENDLOOP	Boucle sans fin
FOR-ENDFOR	Boucle de comptage
WHILE-ENDWHILE	Boucle avec condition en début de boucle
REPEAT-UNTIL	Boucle avec condition en fin de boucle



Fonction

En version standard, la commande traite les blocs CN dans l'ordre de leur programmation.

Ces instructions permettent de définir des alternatives et des boucles de programme complémentaires des sauts de programme décrits dans le présent chapitre.

Les instructions permettent une programmation structurée et une bonne lisibilité du programme.



Procédure

1. IF-ELSE-ENDIF

Une instruction IF-ELSE-ENDIF sert à la sélection entre 2 variantes :

IF (expression)

Blocs CN

ELSE

Blocs CN

ENDIF

Si l'expression a la valeur TRUE, autrement dit si la condition est remplie, la section de programme suivante sera exécutée. Si la condition n'est pas remplie, c'est la branche ELSE qui sera exécutée. La branche ELSE peut être omise.

1.14 Structures de contrôle



840 D
NCU 571



840 D
NCU 572



FM-NC



810D
NCU 573

2. Boucle de programme sans fin LOOP

La boucle infinie trouve son application dans des programmes infinis. Un retour au début de boucle a toujours lieu en fin de boucle.

LOOP

Blocs CN

ENDLOOP

3. Boucle de comptage FOR

La boucle FOR est utilisée lorsqu'une section de programme doit être répétée un certain nombre de fois. La variable de comptage incrémente de façon croissante depuis la valeur initiale jusqu'à la valeur finale. La valeur de début doit être inférieure à la valeur de fin. La variable doit être du type INT (entier).

FOR Variable = valeur initiale **TO** valeur finale

Blocs CN

ENDFOR

4. Boucle de programme avec condition en début de boucle WHILE

La boucle WHILE est exécutée tant que la condition est remplie.

WHILE expression

Blocs CN

ENDWHILE

840 D
NCU 571840 D
NCU 572

FM-NC



810D

NCU 573

5. Boucle de programme avec condition en fin de boucle REPEAT

La boucle REPEAT est exécutée une fois, puis répétée jusqu'à ce que la condition soit remplie.

REPEAT

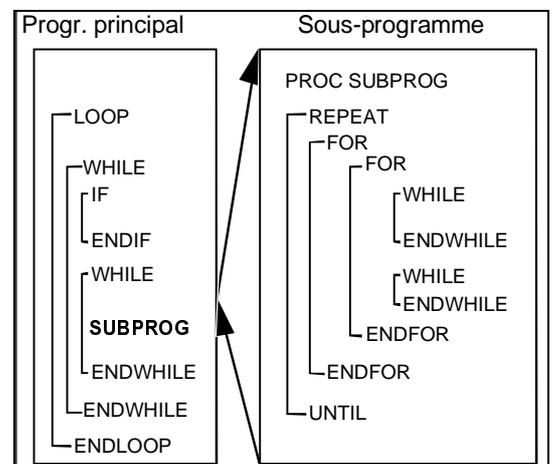
Blocs CN

UNTIL (expression)



Profondeur d'imbrication

Les structures de contrôle sont valables localement dans le programme. Une imbrication de 8 structures de contrôle au maximum est possible à l'intérieur de chaque niveau de sous-programme .



Durée d'exécution des programmes

En mode interpréteur standard, l'utilisation de sauts permet d'obtenir une exécution du programme plus rapide qu'avec les structures de contrôle.

Dans des cycles précompilés, il n'y a pas de différence entre les sauts dans le programme et les structures de contrôle.

1.14 Structures de contrôle



840 D
NCU 571



840 D
NCU 572



FM-NC



810D
NCU 573



Conditions marginales

Les blocs comportant des éléments de structures de contrôle ne peuvent pas être sautés. Dans de tels blocs, les étiquettes ne sont pas autorisées.

Les structures de contrôle sont exécutées de façon interprétative. En cas d'identification d'une fin de boucle, le début de la boucle est recherché compte tenu des structures de contrôle ainsi trouvées.

Par conséquent, en mode interpréteur, la structure d'un programme n'est pas complètement testée.

En principe, il est recommandé de ne pas utiliser en commun des structures de contrôle et des sauts dans le programme.

Si des cycles subissent un prétraitement, l'imbrication correcte des structures de contrôle est vérifiée.



Les structures de contrôle ne sont possibles qu'à l'intérieur de la partie instructions d'un programme.

Les définitions dans l'en-tête de programme ne peuvent pas être exécutées de façon conditionnelle ou répétitive.

Les mots-clés pour les structures de contrôle, tout comme les destinations de sauts, ne doivent pas être utilisés dans des macros. Ceci ne fait pas l'objet d'une vérification lors de la définition des macros.



Exemple de programmation

1. Programme sans fin

```
%_N_LOOP_MPF
```

```
LOOP
```

```
IF NOT $P_SEARCH ;pas de recherche de bloc
```

```
G01 G90 X0 Z10 F1000
```

```
WHILE $AA_IM[X] <= 100
```

```
G1 G91 X10 F500 ;réseau de trous
```

```
Z-5 F100
```

```
Z5
```

840 D
NCU 571840 D
NCU 572

FM-NC



810D

NCU 573

```

ENDWHILE
Z10
ELSE ;recherche de bloc
MSG("pas de perçage pendant la recherche de bloc")
ENDIF
$A_OUT[1] = 1 ;plaque à percer suivante
G4 F2
ENDLOOP
M30

```

2. Fabrication d'un nombre de pièces défini

```

%_N_SERIE_MPF
DEF INT SERIE
FOR SERIE = 0 TO 100
G01 ...
ENDFOR
M30

```

1.15 Coordination de programmes



840 D
NCU 571



840 D
NCU 572



FM-NC



810D

NCU 573

1.15 Coordination de programmes

Canaux

Un canal est susceptible d'exécuter son propre programme, indépendamment des autres canaux. C'est par ce moyen que les axes et broches qui lui sont affectés temporairement, sont contrôlables par programme.

A la mise en service, on peut créer deux canaux ou plus pour la commande.

Coordination de programmes

Si plusieurs canaux sont partie prenante dans la réalisation d'une pièce, alors une synchronisation du déroulement des programmes peut devenir nécessaire.

Pour réaliser cette synchronisation de programmes, il existe des instructions spéciales (ordres) qui doivent figurer dans un bloc spécifique.

840 D
NCU 571840 D
NCU 572
NCU 573

FM-NC



810D

Instructions de coordination de programmes

• Indication de chemin absolu

INIT (n, "Nom de programme") ou
INIT (n, "Nom de programme")

G01 F0.1
START

Exemple :
INIT(2, "DRESSER")

Sélection d'un programme donné en vue de son exécution dans un canal donné :
n : Numéro du canal, valeur selon la configuration de la commande
Nom complet du programme

Jusqu'au logiciel SW 3 :

Entre une instruction **init** (sans synchronisation) et un **Départ programme**, doit figurer au moins un bloc exécutable.

• Indication de chemin relatif

Exemple :
INIT(2, "DRESSER")

Le chemin absolu est formé selon les règles suivantes :
Répertoire courant/_N_nom_MPF
"Répertoire courant" représentant le répertoire pièce sélectionné ou le répertoire standard /_N_MPF_DIR.
Pour l'indication du chemin relatif, les règles à suivre sont les mêmes que pour les appels de sous-programmes.

START (n,n)

Démarrage des programmes sélectionnés dans les autres canaux.

n,n,: énumération des n° de canaux : valeur selon la configuration de la CN

WAITM (N° marque,n,n,...)

Définition de la marque "N° marque" dans le propre canal. Fin du bloc précédent avec arrêt précis. Attente des marques avec le même "N° marque" dans les canaux indiqués "n" (il n'est pas nécessaire que le propre canal soit indiqué). Effacement des marques après la synchronisation.

Jusqu'à 10 marques max. peuvent être définies dans un même canal.

1.15 Coordination de programmes



840 D
NCU 571



840 D
NCU 572



FM-NC



810D
NCU 573

WAITMC(N° marque, n, n, ...)

Définition de la marque "N° marque" dans le propre canal. L'arrêt précis est enclenché uniquement si les autres canaux n'ont pas encore atteint la marque. Attente de la marque avec le même "N° marque" dans les canaux indiqués "n" (il n'est pas nécessaire que le propre canal soit indiqué). Dès que la marque "N° marque" est atteinte dans les canaux indiqués, poursuite de l'usinage sans mettre fin à l'arrêt précis.

WAITE (n,n)

Attente de la fin de programme dans les canaux indiqués (ne pas indiquer le propre canal)

SETM(N° marque, N° marque, ...)

Définition des marques "N° marque" dans le propre canal, sans impact sur l'usinage en cours. SETM() conserve sa validité au-delà de RESET et de Départ programme. SETM() est aussi programmable à partir d'une action de synchronisation.

CLEARM(N° marque, N° marque, ...)

Effacement des "N° marque" dans le propre canal, sans impact sur l'usinage en cours. Tous les marques dans le canal peuvent être effacés avec CLEARM(0). CLEARM() conserve sa validité au-delà de RESET et de Départ programme. CLEARM() est aussi programmable à partir d'une action de synchronisation.

Remarque :

Toutes les instructions ci-dessus doivent occuper des blocs qui leur sont propres.

Noms des canaux

Les noms des canaux doivent être convertis en numéros par le biais de variables (cf. chapitre 10 "Variables et paramètres de calcul").



Protéger l'affectation des numéros pour éviter toute modification accidentelle.

840 D
NCU 571840 D
NCU 572

FM-NC



810D

NCU 573

Exemple :

Au canal avec le nom "MACHINE" on affecte le numéro de canal 1,

au canal avec le nom "CHARGEUR" on affecte le numéro de canal 2 :

```
DEF INT MACHINE=1, CHARGEUR=2
```

Les variables reçoivent le même nom que les canaux.

De ce fait par exemple, l'instruction START est libellée :

```
DEPART (MACHINE)
```

Exemple d'une coordination de programmes

Canal 1 :

```
%_N_MPF100_MPF
```

```
N10 INIT(2, "MPF200")
```

```
N11 START (2)
```

```
.
```

```
N80 WAITM(1, 1, 2)
```

```
.
```

```
N180 WAITM(2, 1, 2)
```

```
.
```

```
N200 WAITE(2)
```

```
N201 M30
```

```
...
```

Exécution dans le canal 2

Attendre marque WAIT 1 dans le canal 1 et dans le canal 2 et continuer l'exécution dans le canal 1

Attendre marque WAIT 2 dans le canal 1 et dans le canal 2 et continuer l'exécution dans le canal 1

Attendre la fin de programme dans le canal 2

Fin de programme dans le canal 1, fin globale

Canal 2 :

```
%_N_MPF200_MPF
```

```
;$PATH=/_N_MPF_DIR
```

```
N70 WAITM(1, 1, 2)
```

```
.
```

```
N270 WAITM(2, 1, 2)
```

```
.
```

```
N400 M30
```

Exécution dans le canal 2

Attendre marque WAIT 1 dans le canal 1 et dans le canal 2
continuer l'exécution dans le canal 1

Attendre marque WAIT 2 dans le canal 1 et dans le canal 2 et continuer l'exécution dans le canal 2

Fin de programme dans le canal 2

1.15 Coordination de programmes



840 D
NCU 571



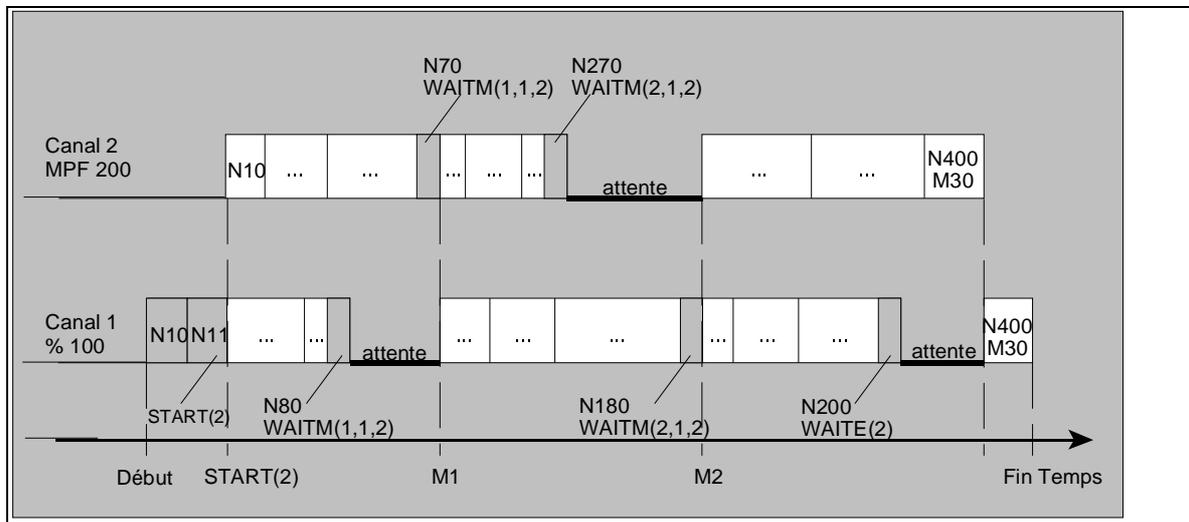
840 D
NCU 572



FM-NC



810D
NCU 573



Exemple de programme à partir d'une pièce

```
N10 INIT(2, "/_N_WKS_DIR/_N_BROCHE1_WPD/_N_DEBRID1_MPF" )
```

Exemple d'une instruction INIT avec indication de chemin relatif

```
;le programme /_N_MPF_DIR/_N_MAIN_MPF est sélectionné dans le canal 1
N10 INIT(2,"MYPROG") ; choisir le programme /_N_MPF_DIR/_N_MYPROG_MPF
dans le canal 2.
```

Remarques complémentaires

Les variables à la disposition commune des canaux (variables globales spécifiques à NCK) peuvent être utilisées pour l'échange de données entre programmes. Sinon l'élaboration des programmes sera entreprise séparément pour chaque canal.



Jusqu'au logiciel SW 3 :

Il ne faut pas qu'un ordre WAITE soit lu tout de suite après l'instruction de démarrage de programme, sinon il y a reconnaissance d'une fin de programme avant que le programme n'ait démarré.

Aide : Programmation d'un arrêt temporisé.

Exemple :

```
N30 START (2)
N31 G4 F0.01
N40 WAITE(2)
```

840 D
NCU 571840 D
NCU 572
NCU 573

FM-NC



810D

1.16 Routine d'interruption



Programmation

```

SETINT(3) PRIO=1 NOM
SETINT(3) PRIO=1 LIFTFAST
SETINT(3) PRIO=1 NAME LIFTFAST
G... X... Y... ALF=...
DISABLE(3)
ENABLE(3)
CLRINT(3)

```



Signification des instructions

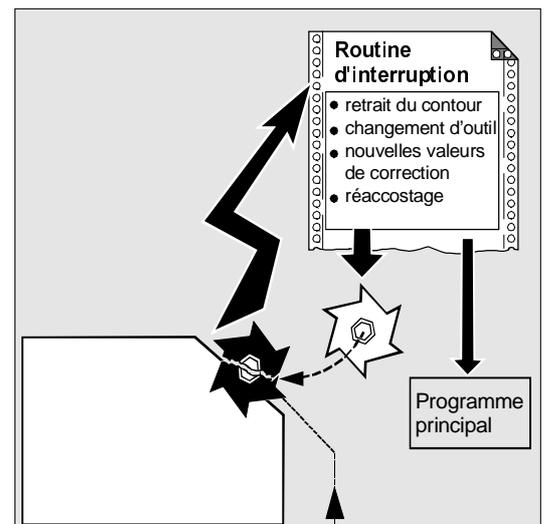
SETINT(n)	Démarrage de la routine d'interruption dès que l'entrée n bascule, n (1...8) représente le numéro d'une des 7 entrées.
PRIO=1	Définition de la priorité de 1 à 128 (1 est prioritaire)
LIFTFAST	Retrait rapide du contour
NOM	Ici figure le nom du sous-programme à exécuter
ALF=...	Sens de déplacement programmable (figure dans le bloc de déplacement)
DISABLE(n)	Désactivation de la routine d'interruption N° n
ENABLE(n)	Réactivation de la routine d'interruption N° n
CLRINT(n)	Effacement des affectations d'interruption de la routine d'interruption N° n



Fonction

Exemple : l'outil se brise pendant l'usinage. Ceci déclenche un signal qui arrête la phase d'usinage en cours et lance simultanément un sous-programme appelé routine d'interruption. Ce sous-programme contient toutes les instructions qui doivent être exécutées dans ce cas.

Lorsque l'exécution du sous-programme est terminée (et la machine à nouveau prête), la commande retourne dans le programme principal et poursuit l'usinage – selon l'instruction REPOS – à l'endroit où il avait été interrompu.



Pour plus d'informations sur REPOS, consultez le chapitre 9 Mode de déplacement, Réaccostage du contour

1.16 Routine d'interruption



840 D
NCU 571



840 D
NCU 572
NCU 573



FM-NC



810D



Procédure

Créer une routine d'interruption comme sous-programme

La routine d'interruption est définie comme un sous-programme.

Exemple :

```
PROC RETR_Z  
N10...  
N50 M17
```

Nom du programme RETR_Z suivi des blocs CN, puis M17 pour fin de programme et retour dans le programme principal.



Remarque :

Des instructions SETINT peuvent être programmées à l'intérieur de la routine d'interruption pour déclencher d'autres routines d'interruption. Le démarrage de ces dernières n'a lieu que lorsque l'entrée correspondante bascule.



Pour plus d'informations sur la création de sous-programmes, consultez le chapitre 9.

Mémoriser la position d'interruption, SAVE

Au moment de la définition, on peut identifier la routine d'interruption avec SAVE.

Exemple :

```
PROC RETR_Z SAVE  
N10...  
N50 M17
```

Avec SAVE, on mémorise en même temps l'état courant de l'usinage, qui est restauré après l'interruption. On retrouve les fonctions G modales, l'avance courante et le frame global courant.

Ceci permet de poursuivre ultérieurement l'usinage au point d'interruption.

840 D
NCU 571840 D
NCU 572
NCU 573

FM-NC



810D

Affecter et lancer une routine d'interruption, SETINT

La commande dispose de 8 signaux (entrée 1...8) qui peuvent déclencher une interruption du programme en cours et lancer la routine d'interruption correspondante.

La définition „quelle entrée lance quel programme“ s'effectue dans le programme principal.

Exemple :

```
N10 SETINT(3) PRIO=1 RETR_Z
```

La routine RETR_Z est lancée dès que l'entrée 3 est basculée.

Lancer plusieurs routines d'interruption, définir les priorités de lancement, PRIO=

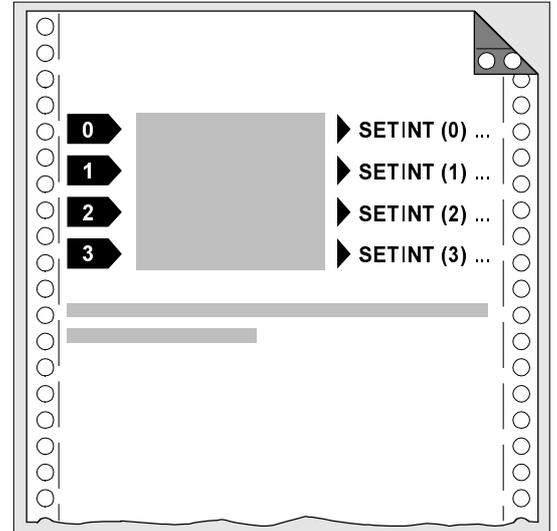
Si votre programme CN contient plusieurs instructions SETINT et si par conséquent plusieurs signaux peuvent arriver simultanément, il est nécessaire de préciser la priorité d'exécution des routines d'interruption : PRIO de 1 à 128, 1 étant prioritaire.

Exemple :

```
N10 SETINT(3) PRIO=1 RETR_Z
N20 SETINT(2) PRIO=2 RETR_X
```

Lorsque les signaux arrivent simultanément aux entrées, les routines sont exécutées selon la priorité qui a été définie. D'abord SETINT(3), puis SETINT(2).

Si de nouveaux signaux arrivent pendant l'exécution d'une routine d'interruption, cette routine en cours est interrompue, s'il y a lieu, par les routines prioritaires.



1.16 Routine d'interruption



840 D
NCU 571



840 D
NCU 572



FM-NC



810D

Désactiver / Réactiver une routine d'interruption, DISABLE, ENABLE

On peut désactiver des routines d'interruption dans le programme CN avec DISABLE(n) et les réactiver avec ENABLE(n) (n correspond au N° de l'entrée).



La correspondance entre l'entrée et la routine est conservée avec DISABLE et est à nouveau activée avec ENABLE.

Réaffecter des routines d'interruption

Si une nouvelle routine est affectée à une entrée déjà occupée, l'ancienne affectation est désactivée automatiquement.

Exemple :

```
N20 SETINT(3) PRIO=2 RETR_Z
...
...
N120 SETINT(3) PRIO=1 RETR_X
```

Effacer l'affectation, CLRINT

CLRINT(n) permet d'effacer des affectations.

Exemple :

```
N20 SETINT(3) PRIO=2 RETR_Z
N50 CLRINT(3)
```

L'affectation de la routine RETR_Z à l'entrée 3 est effacée.

840 D
NCU 571840 D
NCU 572
NCU 573

FM-NC



810D

Retrait rapide du contour, LIFTFAST

Avec LIFTFAST, dès qu'il y a basculement d'une entrée, l'outil effectue un retrait rapide l'éloignant du contour de la pièce .

Si l'instruction SETINT contient une routine d'interruption en plus de LIFTFAST, le retrait rapide est exécuté avant la routine d'interruption.

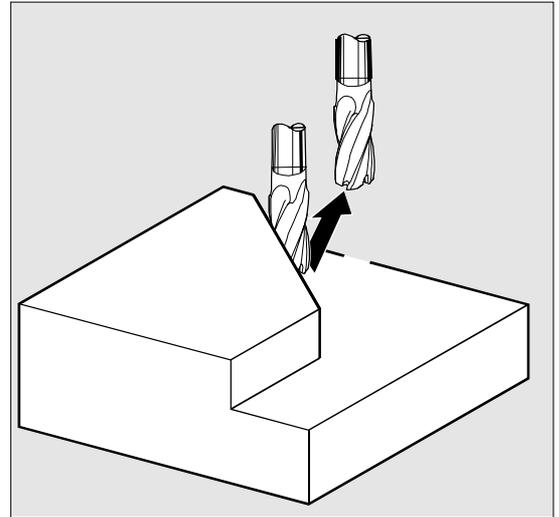
Exemple :

```
N10 SETINT(2) PRIO=1 LIFTFAST
```

ou

```
N30 SETINT(2) PRIO=1 RETR_Z LIFTFAST
```

Dans les deux cas, quand c'est l'entrée 2 qui bascule, le retrait rapide est exécuté prioritairement . La routine d'interruption est lancée ensuite.



Séquence de déplacement lors du retrait rapide

La valeur du déplacement opéré par les axes géométriques dans le cas d'un retrait rapide pour éloigner l'outil du contour, est réglable par le biais d'un paramètre machine.

Sens de déplacement programmable, ALF=...

Indiquer, dans le programme CN, la direction dans laquelle l'outil doit se déplacer en cas de retrait rapide.

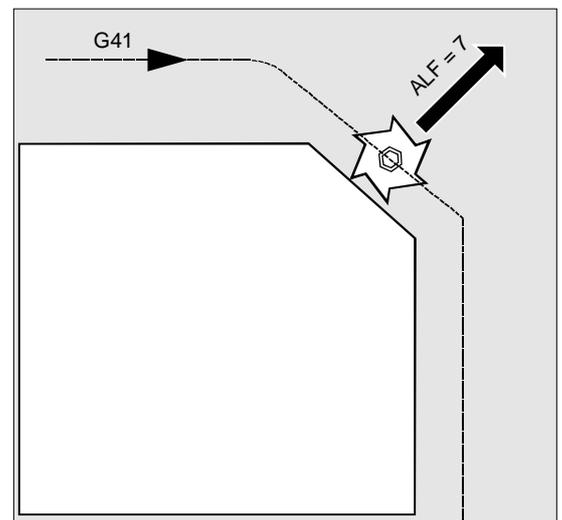
Les sens de déplacement possibles sont enregistrés dans la commande sous des n° de code spéciaux et peuvent être appelés par le biais de ces numéros.

Exemple :

```
N10 SETINT(2) PRIO=1 LIFTFAST
```

```
ALF=7
```

Vu du dessus – quand G41 est activée (outil à gauche du contour) – l'outil s'éloigne suivant la normale au contour.



1.16 Routine d'interruption



840 D
NCU 571



840 D
NCU 572



FM-NC

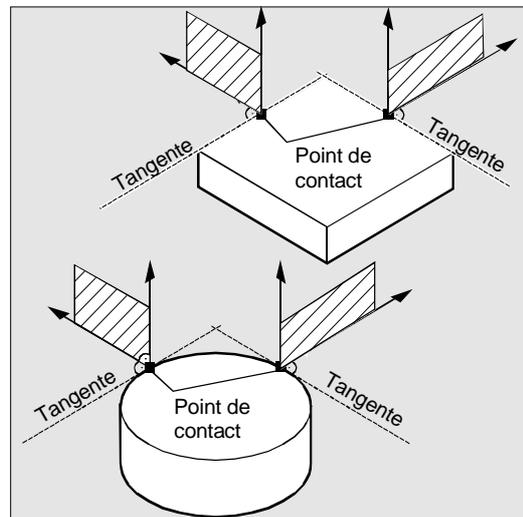


810D
NCU 573

Plan de référence pour la définition des sens de déplacement

Un plan est défini au point de contact de l'outil sur le contour programmé. Ce plan sert de référence pour préciser le mouvement de retrait avec le numéro de code correspondant.

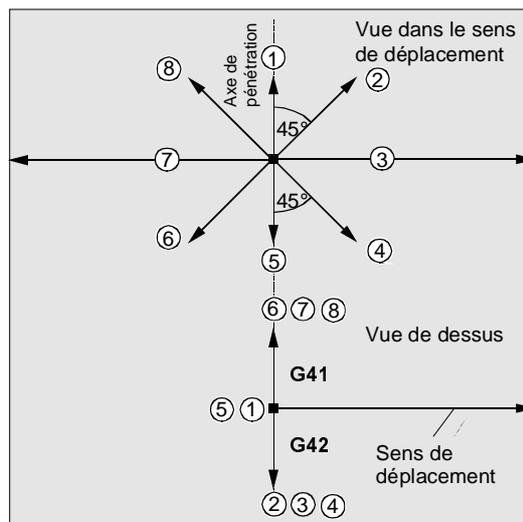
Le plan de référence est défini à partir de l'axe longitudinal de l'outil (direction de pénétration) et d'un vecteur perpendiculaire à ce dernier et à la tangente au point de contact de l'outil avec le contour.



Numéros de code et sens de déplacement (vue d'ensemble)

La figure ci-contre représente les numéros de code et les sens de déplacement par rapport au plan de référence.

ALF=0 désactive la fonction de retrait rapide.



A observer :

Quand la correction du rayon d'outil est activée, les codes 2, 3, 4 avec G41 et les codes 6, 7, 8 avec G42 ne doivent pas être utilisés.

Dans ces cas, l'outil risque de rejoindre le contour et d'entrer en collision avec la pièce.

840 D
NCU 571840 D
NCU 572
NCU 573

FM-NC



810D

Déplacement de retrait à partir de SW 4.3

La direction du déplacement de retrait est programmée à l'aide des instructions **LFTXT** ou **LFWP** avec la variable **ALF**.

- **LFTXT**

Le plan du déplacement de retrait est déterminé par la tangente à la trajectoire et la direction de l'outil. Cette instruction (préréglage) correspond à la programmation du retrait rapide dans les versions de logiciel précédentes.

- **LFWP**

Le plan du déplacement de retrait est le plan de travail actif sélectionné avec les fonctions G17, G18 et G19. La direction du déplacement de retrait est indépendante de la tangente à la trajectoire. Ceci permet de programmer un retrait rapide paraxial.

Dans le plan du déplacement de retrait, la direction de retrait est programmée comme précédemment avec **ALF** par incréments de 45 degrés.

Dans le cas de **LFTXT**, ALF=1 correspond au retrait dans la direction de l'outil.

Dans le cas de **LFWP**, la direction du retrait dans le plan de travail est déterminée comme suit :

- **G17** : plan X/Y ALF=1 retrait en X
 ALF=3 retrait en Y
- **G18** : plan Z/X ALF=1 retrait en Z
 ALF=3 retrait en X
- **G19** : plan Y/Z ALF=1 retrait en Y
 ALF=3 retrait en Z

1.16 Routine d'interruption



840 D
NCU 571



840 D
NCU 572



FM-NC



810D

NCU 573



Exemple de programmation

Dans cet exemple, un outil brisé doit être remplacé automatiquement par un outil de rechange.

L'usinage se poursuit alors avec le nouvel outil.

Programme principal

```
N10 SETINT(1) PRIO=1 CHANG_O ->
-> LIFTFAST
```

Si l'entrée 1 bascule, un retrait rapide éloigne immédiatement l'outil du contour (code n° 7 pour la correction de rayon d'outil G41). Il est suivi de l'exécution de la routine d'interruption CHANG_O.

```
N20 G0 Z100 G17 T1 ALF=7 D1
```

```
N30 G0 X-5 Y-22 Z2 M3 S300
```

```
N40 Z-7
```

```
N50 G41 G1 X16 Y16 F200
```

```
N60 Y35
```

```
N70 X53 Y65
```

```
N90 X71.5 Y16
```

```
N100 X16
```

```
N110 G40 G0 Z100 M30
```

Sous-programme

```
PROC CHANG_O SAVE
```

Sous-programme avec mise en mémoire de l'état courant

```
N10 G0 Z100 M5
```

Position de changement d'outil, arrêt de la broche

```
N20 T11 M6 D1 G41
```

Changement d'outil

```
N30 REPOS L RMB M3
```

Réaccostage du contour et retour au programme principal

-> doit être programmé dans un bloc



Si vous ne programmez pas d'instruction REPOS dans le sous-programme, il y a positionnement au point final du bloc qui suit le bloc interrompu.

1.17 Permutation d'axe, permutation de broche

840 D
NCU 571840 D
NCU 572
NCU 573

FM-NC



810D

1.17 Permutation d'axe, permutation de broche



Signification des instructions

RELEASE(nom d'axe, nom d'axe, ...)	Libérer l'axe
GET(nom d'axe, nom d'axe, ...)	Prise en charge de l'axe
GETD(nom d'axe, nom d'axe, ...)	Prise en charge directe de l'axe
Nom d'axe	Affectation d'axe dans le système : AX1, AX2, ... ou indication des noms d'axes machine
RELEASE(S1)	Libération des broches S1, S2, ...
GET(S2)	Prise en charge des broches S1, S2, ...
GETD(S3)	Prise en charge directe des broches S1, S2, ...



Fonction

Un axe ou plusieurs axes (une ou plusieurs broches) ne peuvent être utilisés que dans un seul canal. Si un axe doit pouvoir être utilisé alternativement dans deux canaux différents (par ex. dans le cas d'un changeur de palettes), il doit être libéré dans le canal courant et ensuite pris en charge dans l'autre canal : l'axe est permuté entre les canaux.



Procédure

Conditions nécessaires à la permutation d'axe

L'axe doit être défini dans tous les canaux par l'intermédiaire des paramètres machine.

Il faudra définir, par l'intermédiaire du paramètre machine spécifique à l'axe, à quel canal l'axe devra être affecté après un POWER ON.

1.17 Permutation d'axe, permutation de broche



840 D
NCU 571



840 D
NCU 572



FM-NC



810D
NCU 573

Libération d'axe : RELEASE

Lors de la libération de l'axe, veillez aux points suivants :

1. L'axe ne doit participer à aucune transformation.
2. En cas de couplage d'axes (positionnement tangentiel, déplacements conjugués), tous les axes du groupe doivent être libérés.
3. Un axe de positionnement concourant ne doit pas être permuté.
4. Dans le cas d'un axe maître Gantry (axe de portique), tous les axes asservis sont aussi permutés.

Prise en charge de l'axe : GET

La permutation d'axe devient effective avec cette instruction. Le canal dans lequel l'instruction a été programmée prend entièrement en charge l'axe.

Conséquences de GET :

Permutation d'axe avec synchronisation :

Un axe doit alors toujours être synchronisé s'il a été affecté entre temps à un canal différent ou à l'AP, et qu'il n'y a pas eu synchronisation avant GET par "WAITP", G74 ou effacement de la distance restant à parcourir.

- Il y a arrêt du prétraitement des blocs (comme pour STOPRE).
- **L'usinage est interrompu jusqu'à ce que la permutation d'axe soit terminée.**

Permutation d'axe sans synchronisation :

Si l'axe ne doit pas être synchronisé, GET ne crée pas d'arrêt du prétraitement des blocs.

Exemple :

```
N01 G0 X0
N02 RELEASE (AX5)
N03 G64 X10
N04 X20
N05 GET (AX5)

N06 G01 F5000
N07 X20

N08 X30
N09 ...
```

Si aucune synchronisation n'est requise, ce bloc sera non exécutable.

Bloc non exécutable.

Bloc non exécutable, car position X comme dans N04.

Premier bloc exécutable après N05.

1.17 Permutation d'axe, permutation de broche



840 D
NCU 571



840 D
NCU 572
NCU 573



FM-NC



810D

"GET" automatique

Si un axe est disponible par principe dans le canal, mais rendu momentanément indisponible comme «axe de canal», alors un GET est automatiquement exécuté. Au cas où l'(les)axe(s) était(ent) déjà synchronisé(s), il n'y a pas d'arrêt du prétraitement.



Un axe pris en charge par un canal reste affecté à ce canal, même après un reset manuel ou un reset programmé. En cas de redémarrage du programme, l'affectation de l'axe permuté ou de la broche permutée doit être programmée si l'axe doit être utilisé dans son canal de base.

A la mise sous tension de la CN, il est affecté au canal spécifié dans les paramètres machine.

Prise en charge directe de l'axe : GETD

Avec GETD (GET Directly), un axe est extrait directement d'un autre canal. Cela signifie que, pour ce GETD, il n'est pas nécessaire de programmer un RELEASE approprié dans un autre canal. Par contre, cela signifie aussi qu'il faut établir un autre type de communication (par ex. avec des marques d'attente).

1.17 Permutation d'axe, permutation de broche

840 D
NCU 571840 D
NCU 572
NCU 573

FM-NC



810D



Exemple de programmation

Des 6 axes seront utilisés pour l'usinage dans le canal 1 : les 1er, 2e, 3e et 4e axes.

Les 5e et 6e axes seront utilisés dans le canal 2 pour le changement d'outil.

L'axe 2 doit pouvoir être permuté entre les deux canaux et, lors de la mise sous tension de la CN, affecté au canal 1.

Programme „MAIN“ dans canal 1

%_N_MAIN_MPF	
INIT (2, "PERMUT2")	Sélectionner le programme PERMUT2 dans le canal 2
N... START (2)	Démarrer le programme dans le canal 2
N... GET (AX2)	Prendre en charge l'axe AX2
...	
...	
N... RELEASE (AX2)	Libérer l'axe AX2
N... WAITM (1, 1, 2)	Attendre la marque WAIT dans les canaux 1 et 2 pour la synchronisation dans les deux canaux
N...	Poursuite des opérations après la permutation d'axe
N... M30	

Programme „Permut2“ dans canal 2

%_N_PERMUT2_MPF	
N... RELEASE (AX2)	
N160 WAITM (1, 1, 2)	Attendre la marque WAIT dans les canaux 1 et 2 pour la synchronisation dans les deux canaux
N150 GET (AX2)	Prendre en charge l'axe AX2
N...	Poursuite des opérations après la permutation d'axe
N...M30	

840 D
NCU 571840 D
NCU 572
NCU 573

FM-NC



810D

1.18 Axes géométriques permutables, GEOAX



Programmation

GEOAX(n,axe de canal,n,axe de canal,...)

GEOAX()



Signification des paramètres

GEOAX(n,axe de canal,n,axe de canal,...)	Permutation des axes géométriques
GEOAX()	Appel de la configuration de base des axes géométriques
n	Numéro de l'axe géométrique (n=1, 2 ou 3) auquel doit être affecté un autre axe de canal. n=0: extraire du groupe d'axes géométriques le canal d'axe indiqué, sans le remplacer
Axe de canal	Nom de l'axe de canal qui doit être intégré dans le groupe d'axes géométriques.



Fonction

La fonction "Axes géométriques permutables" permet de modifier, à partir du programme pièce, le groupe d'axes géométriques qui a été configuré dans les paramètres machine. Ainsi, un axe de canal défini comme axe supplémentaire synchrone peut remplacer un axe géométrique quelconque.

Exemple :

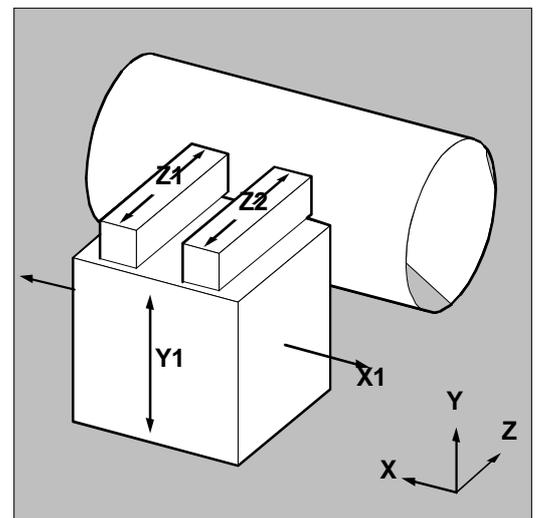
Un chariot porte-outil peut être déplacé par le biais des axes de canal X1, Y1, Z1, Z2. Dans le programme pièce, les axes Z1 et Z2 doivent intervenir de façon alternée comme axe géométrique Z. La permutation entre les axes se fait avec GEOAX dans le programme pièce.

Après la mise en service, la liaison X1, Y1, Z1 est active (réglable avec PM).

N100 GEOAX (3,Z2)

N110 G1

N120 GEOAX (3,Z1)



L'axe de canal Z2 tient lieu d'axe Z.

L'axe de canal Z1 tient lieu d'axe Z.

1.18 Axes géométriques permutables, GEOAX



840 D
NCU 571



840 D
NCU 572



FM-NC



810D
NCU 573



Procédure

Le numéro d'axe géométrique

Dans l'instruction GEOAX([n,axe de canal]...) le numéro n désigne l'axe géométrique auquel doit être affecté l'axe de canal indiqué à sa suite.

Pour la permutation d'un axe de canal, les numéros d'axes géométriques 1 à 3 (axes X, Y, Z) sont admis.

Avec $n = 0$ on extrait du groupe d'axes géométriques un axe de canal donné, sans remplacement de l'axe géométrique.

Un axe remplacé par permutation dans le groupe d'axes géométriques est programmable comme axe supplémentaire, après la permutation, par le biais de son nom d'axe.



La permutation des axes géométriques provoque l'effacement de tous les frames, des zones de protection et des limitations de la zone de travail. Un éventuel décalage par manivelle (DRF) ou un décalage d'origine externe reste activé après la permutation.

Permuter les positions d'axe

En procédant à une réaffectation des numéros d'axe aux axes de canal déjà affectés, on peut aussi permuter la position des axes dans le groupe d'axes géométriques.

N... GEOAX (1, XX, 2, YY, 3, ZZ)

N... GEOAX (1, U, 2, V, 3, W)

L'axe de canal XX est le premier, l'axe de canal YY le second et l'axe de canal ZZ le troisième axe géométrique.

L'axe de canal U est le premier, l'axe de canal V le second et l'axe de canal W le troisième axe géométrique.

840 D
NCU 571840 D
NCU 572
NCU 573

FM-NC



810D



Conditions et restrictions

1. La permutation des axes géométriques n'est pas possible quand :
 - la transformation est activée,
 - l'interpolation de type spline est activée,
 - la correction du rayon d'outil est activée,
 - la correction d'outil fine est activée.
2. Quand l'axe géométrique et l'axe de canal portent le même nom, la permutation de l'axe géométrique est impossible.
3. Aucun des axes participant à la permutation ne doit intervenir dans une action qui perdure au-delà des limites de bloc, comme c'est le cas par exemple des axes de positionnement de type A ou des axes asservis.
4. Avec l'instruction GEOAX, on ne peut remplacer que des axes géométriques qui étaient définis au moment de la mise en service (autrement dit, on ne peut pas en définir des nouveaux en sus).

(Manuel de programmation - Notions de base : Chap. 8)

(Manuel de programmation - Notions de base : Chap. 8)

Désactiver la permutation

L'instruction GEOAX() appelle la configuration de base du groupe d'axes géométriques.

Après un POWER ON et après une commutation en mode de fonctionnement Accostage du point de référence, la configuration de base est remise en place automatiquement.



Remarques complémentaires

Permutation et correction de longueur d'outil

Une correction de longueur d'outil activée reste activée après une permutation. Toutefois, elle est considérée pour les nouveaux axes géométriques et pour les axes géométriques dont la position a été permuée, comme n'ayant pas encore été exécutée. Par conséquent, à la première instruction de déplacement concernant ces axes géométriques, la course résultante se compose de la distance à parcourir telle qu'elle a été programmée, à laquelle s'ajoute la correction de longueur d'outil.

1.18 Axes géométriques permutables, GEOAX



840 D
NCU 571



840 D
NCU 572



FM-NC



810D

NCU 573

Les axes géométriques qui conservent leur position dans le groupe d'axes après une permutation, conservent aussi leur état en regard de la correction de longueur d'outil.

Configuration des axes géométriques et changement de transformation

La configuration des axes géométriques (définie dans les paramètres machine) telle qu'elle figure dans une transformation activée n'est pas modifiable par le biais de la fonction "Axes géométriques permutables".

S'il est nécessaire de modifier la configuration des axes géométriques en liaison avec des transformations, il faut alors faire intervenir une nouvelle transformation.

Une configuration d'axes géométriques modifiée avec GEOAX est effacée par l'activation d'une transformation.

S'il y a contradiction dans les paramètres machine entre les réglages pour la transformation et les réglages pour la permutation des axes géométriques, ce sont les réglages concernant la transformation qui sont prioritaires.

Exemple :

Supposons qu'une transformation soit active. D'après les paramètres machine, la transformation doit être conservée après un reset, mais en même temps, un reset doit rétablir la configuration de base des axes géométriques. Dans ce cas, c'est la configuration des axes géométriques telle que définie avec la transformation, qui est conservée.

840 D
NCU 571840 D
NCU 572
NCU 573

FM-NC



810D



Exemple de programmation

Une machine comporte 6 axes de canal désignés par XX, YY, ZZ, U, V, W. Le préréglage de la configuration des axes géométriques effectué dans les paramètres machine est le suivant :

axe de canal XX = 1e axe géométrique (axe X)

axe de canal YY = 2e axe géométrique (axe Y)

axe de canal ZZ = 3e axe géométrique (axe Z)

N10	GEOAX()	La configuration de base des axes géométriques est activée.
N20	G0 X0 Y0 Z0 U0 V0 W0	Tous les axes accostent la position 0 en vitesse rapide.
N30	GEOAX(1, U, 2, V, 3, W)	L'axe de canal U devient premier (X), l'axe de canal V second (Y), l'axe de canal W troisième (Z) axe géométrique.
N40	GEOAX(1, XX, 3, ZZ)	L'axe de canal XX devient premier (X), l'axe de canal ZZ troisième (Z) axe géométrique. L'axe de canal V reste second axe géométrique (Y).
N50	G17 G2 X20 I10 F1000	Cercle complet dans le plan X, Y. Déplacement des axes de canal XX et V
N60	GEOAX(2, W)	L'axe de canal W devient second axe géométrique (Y).
N80	G17 G2 X20 I10 F1000	Cercle complet dans le plan X, Y. Déplacement des axes de canal XX et W.
N90	GEOAX()	Retour à l'état de base
N100	GEOAX(1, U, 2, V, 3, W)	L'axe de canal U devient premier (X), l'axe de canal V second (Y), l'axe de canal W troisième (Z) axe géométrique.
N110	G1 X10 Y10 Z10 XX=25	Les axes de canal U, V, W se déplacent respectivement sur la position 10, l'axe XX en tant qu'axe supplémentaire se déplace sur la position 25.
N120	GEOAX(0, V)	V est extrait du groupe d'axes géométriques. U et W restent premier (X) et troisième (Z) axes géométriques. Le second axe géométrique (Y) n'est pas affecté.
N130	GEOAX(1, U, 2, V, 3, W)	L'axe de canal U reste premier (X), l'axe de canal V second (Y) et l'axe de canal W troisième (Z) axe géométrique.
N140	GEOAX(3, V)	V devient troisième axe géométrique (Z), écrasant W qui est ainsi extrait du groupe d'axes géométriques. Le second axe géométrique (Y) n'est toujours pas encore affecté.

1.19 Répétition de parties de programme (à partir du logiciel 4.3)



840 D
NCU 571



840 D
NCU 572
NCU 573



810D

1.19 Répétition de parties de programme (à partir du logiciel 4.3)



Fonction

A la différence des sous-programmes, cette fonction permet la répétition de parties d'un programme dans un ordre quelconque. Le bloc ou la section de programme devant être répété est repéré par des étiquettes.

En ce qui concerne les étiquettes, voir :

Bibliographie : /PG/, Manuel de programmation - Notions de base, chap. 2.2
/PGA/, Manuel programmation - Notions complémentaires, chap. 1.11, 1.12



Signification

ETIQUETTE :	Destination de saut ; l'étiquette doit être suivie d'un deux-points
REPEAT	Répète
REPEATB	Répète bloc



Programmation

Répéter un bloc

```
ETIQUETTE : xxx
YYY
REPEATB ETIQUETTE P=n
zzz
```

La ligne de programme repérée par l'étiquette est répétée P=n fois.

Si P n'est pas indiqué, le bloc est répété une fois seulement. Après la dernière répétition, l'exécution du programme reprend à la ligne zzz qui suit la ligne REPEATB.



Le bloc repéré par l'étiquette peut se trouver avant ou après l'instruction REPEATB.

La recherche commence en direction du début du programme. Si l'étiquette n'est pas trouvée dans cette direction, la recherche est poursuivie en direction de la fin du programme.

1.19 Répétition de parties de programme (à partir du logiciel 4.3)



840 D
NCU 571



840 D
NCU 572
NCU 573



810D



Exemple de programmation

Répétition de positions

N10 POSITION1: X10 Y20	
N20 POSITION2: CYCLE(0,,9,8)	Cycle de positions
N30 ...	
N40 REPEATB POSITION1 P=5	exécute le bloc N10 cinq fois
N50 REPEATB POSITION2	exécute le bloc N20 une fois
N60 ...	
N70 M30	



Programmation

Répéter une section de programme à partir d'une étiquette

```
ETIQUETTE: xxx
YYY
REPEAT ETIQUETTE P=n
zzz
```

La section de programme se trouvant entre l'étiquette et l'instruction REPEAT est répétée P=n fois.

Si le bloc comportant l'étiquette contient d'autres instructions, celles-ci sont exécutées à chaque répétition.

Si P n'est pas indiqué, la section de programme est répétée une fois seulement.

Après la dernière répétition, l'exécution du programme reprend à la ligne zzz qui suit la ligne REPEAT.



L'étiquette doit figurer avant l'instruction REPEAT. La recherche a lieu uniquement en direction du début du programme.

1.19 Répétition de parties de programme (à partir du logiciel 4.3)840 D
NCU 571840 D
NCU 572
NCU 573

810D

**Exemple de programmation**

Réalisation de 5 carrés ayant des côtés croissants

```

N5 R10=15
N10 DEBUT: R10=R10+1                                Côté
N20 Z=10-R10
N30 G1 X=R10 F200
N40 Y=R10
N50 X=-R10
N60 Y=-R10
N70 Z=10+R10
N80 REPEAT DEBUT P=4                                exécute les blocs N10 à N70 quatre fois
N90 Z10
N100 M30

```

**Programmation****Répéter une section de programme entre deux étiquettes**

```

ETIQUETTE_DEBUT: xxx
ooo
ETIQUETTE_FIN: yyy
ppp
REPEAT ETIQUETTE_DEBUT ETIQUETTE_FIN
P=n
zzz

```

La section de programme située entre les deux étiquettes est répétée P=n. Des noms quelconques peuvent être attribués aux étiquettes.

La première ligne de la section répétée est celle comportant l'étiquette de début, la dernière ligne celle comportant l'étiquette de fin. Si la ligne comportant l'étiquette de début ou de fin contient d'autres instructions, celles-ci sont exécutées à chaque répétition.

Si P n'est pas indiqué, la section de programme est répétée une fois seulement. Après la dernière répétition, l'exécution du programme reprend à la ligne zzz qui suit la ligne REPEAT.



La section de programme à répéter peut se trouver avant ou après l'instruction REPEAT. La recherche commence en direction du début du programme.

1.19 Répétition de parties de programme (à partir du logiciel 4.3)



840 D
NCU 571



840 D
NCU 572
NCU 573



810D

Si l'étiquette de début n'est pas trouvée dans cette direction, la recherche continue en direction de la fin du programme.

Il n'est pas permis de placer l'instruction REPEAT entre les deux étiquettes. Si l'étiquette de début est trouvée avant l'instruction REPEAT et si l'étiquette de fin n'est pas atteinte avant l'instruction REPEAT, seule la section de programme entre étiquette de début et instruction REPEAT est répétée.



Exemple de programmation

Répéter la section de programme entre DEBUT et FIN

N5 R10=15	
N10 DEBUT: R10=R10+1	Côté
N20 Z=10-R10	
N30 G1 X=R10 F200	
N40 Y=R10	
N50 X=-R10	
N60 Y=-R10	
N70 FIN: Z=10	
N80 Z10	
N90 CYCLE(10, 20, 30)	
N100 REPEAT DEBUT FIN P=3	exécute les blocs N10 à N70 trois fois
N110 Z10	
N120 M30	
N100 M30	



Programmation

Répétition d'une section de programme entre une étiquette et une étiquette de fin

```
ETIQUETTE xxx
ooo
ENDLABEL: yyy
REPEAT ETIQUETTE P=n
zzz
```

ENDLABEL est une étiquette prédéfinie ayant un nom invariable. ENDLABEL marque la fin d'une section de programme et peut être utilisé plusieurs fois dans le programme.

Le bloc comportant ENDLABEL peut contenir d'autres instructions.

1.19 Répétition de parties de programme (à partir du logiciel 4.3)



840 D
NCU 571



840 D
NCU 572
NCU 573



810D

La section de programme entre une étiquette et le prochain ENDLABEL est répétée P=n fois. L'étiquette de début peut porter un nom quelconque. Si le bloc comportant l'étiquette de début ou ENDLABEL contient d'autres instructions, celles-ci sont exécutées à chaque répétition.



Si aucun ENDLABEL n'est trouvé entre l'étiquette de début et l'instruction REPEAT, la boucle se termine devant le bloc comportant REPEAT. Ceci équivaut à la structure "Répéter une section de programme à partir d'une étiquette" décrite précédemment. Si P n'est pas indiqué, la section de programme est répétée une fois seulement. Après la dernière répétition, l'exécution du programme reprend à la ligne zzz qui suit la ligne REPEAT.



Exemple de programmation

```

N10 G1 F300 Z-10
N20 DEBUT1 :
N30 X10
N40 Y10
N50 DEBUT2 :
N60 X20
N70 Y30
N80 ENDLABEL :           Z10
N90 X0 Y0 Z0
N100 Z-10
N110 DEBUT3 : X20
N120 Y30
N130 REPEAT DEBUT3 P=3           exécute les blocs N110 à N120 trois fois
N140 REPEAT DEBUT2 P=2           exécute les blocs N50 à N80 deux fois
N150 M100
N160 REPEAT DEBUT1 P=2           exécute les blocs N20 à N80 deux fois
N170 Z10
N180 X0 Y0
N190 M30

```

1.19 Répétition de parties de programme (à partir du logiciel 4.3)



840 D
NCU 571



840 D
NCU 572
NCU 573



810D



Conditions marginales

- Des imbrications sont autorisées dans une partie de programme à répéter. Chaque appel correspond à un niveau de sous-programme.
- Si M17 ou RET est programmé dans une partie de programme à répéter, il y a abandon de la répétition. L'exécution du programme reprend au bloc qui suit la ligne REPEAT.
- Dans l'affichage du programme en cours, la répétition de la partie de programme est visualisée en tant que niveau spécifique de sous-programme.
- Si un abandon est déclenché pendant l'exécution de la partie de programme, l'exécution du programme reprend après le bloc comportant l'instruction REPEAT.

Exemple :

```
N5 R10=15
N10 DEBUT: R10=R10+1
N20 Z=10-R10
N30 G1 X=R10 F200
N40 Y=R10
N50 X=-R10
N60 Y=-R10
N70 FIN: Z10
N80 Z10
N90 CYCLE(10,20,30)
N100 REPEAT DEBUT FIN P=3
N120 Z10
N130 M30
```

Côté

Abandon

Poursuite de l'exécution du programme

- Structures de contrôle et répétition de parties de programme peuvent être combinées. Il y a cependant lieu d'éviter les chevauchements. Une répétition de partie de programme devrait se trouver à l'intérieur d'une branche de structure de contrôle et une structure de contrôle à l'intérieur d'une répétition de partie de programme.
- En cas de programmation mixte de sauts et d'une répétition de partie de programme, les blocs sont exécutés de façon purement séquentielle.

1.19 Répétition de parties de programme (à partir du logiciel 4.3)



840 D
NCU 571



840 D
NCU 572
NCU 573



810D

En cas de saut à partir d'une boucle de répétition de partie de programme p. ex., l'exécution continue jusqu'à ce que la fin de la partie de programme à répéter soit trouvée.

Exemple :

```
N10 G1 F300 Z-10  
N20 DEBUT1 :  
N30 X10  
N40 Y10  
N50 GOTOF DEBUT2  
N60 ENDLABEL :  
N70 DEBUT2 :  
N80 X20  
N90 Y30  
N100 ENDLABEL : Z10  
N110 X0 Y0 Z0  
N120 Z-10  
N130 REPEAT DEBUT1 P=2  
N140 Z10  
N150 X0 Y0  
N160 M30
```



Activation

Une répétition de partie de programme est activée par programmation.

1.19 Répétition de parties de programme (à partir du logiciel 4.3)



840 D
NCU 571



840 D
NCU 572
NCU 573



810D



Exemple de programmation

Réseaux de trous : réaliser des trous taraudés en différentes positions

N10 FORET_CENTRER()	Mettre en place foret à centrer
N20 POS_1 :	Réseau de positions 1
N30 X1 Y1	
N40 X2	
N50 Y2	
N60 X3 Y3	
N70 ENDLABEL :	
N80 POS_2 :	Réseau de positions 2
N90 X10 Y5	
N100 X9 Y-5	
N110 X3 Y3	
N120 ENDLABEL :	
N130 FORET()	mettre en place foret et cycle de perçage
N140 TARAUD(6)	mettre en place taraud M6 et cycle de taraudage
N150 REPEAT POS_1	répéter section de programme à partir de POS_1 une fois jusqu'à ENDLABEL
N160 FORET()	mettre en place foret et cycle de perçage
N170 TARAUD(8)	mettre en place taraud M8 et cycle de taraudage
N180 REPEAT POS_2	répéter section de programme à partir de POS_2 une fois jusqu'à ENDLABEL
N190 M30	

1.19 Répétition de parties de programme (à partir du logiciel 4.3)



840 D
NCU 571



840 D
NCU 572
NCU 573



810D

Notes

Sous-programmes, macroprogrammation

2.1 Utilisation des sous-programmes.....	2-96
2.2 Sous-programme avec mécanisme SAVE	2-98
2.3 Sous-programmes avec transfert de paramètres	2-99
2.4 Appel d'un sous-programme	2-103
2.5 Sous-programme à répétition	2-106
2.6 Sous-programme modal, MCALL	2-107
2.7 Appel indirect d'un sous-programme	2-108
2.8 Appel d'un sous-programme avec indication de chemin et paramètres, PCALL.....	2-109
2.9 Inhibition de l'affichage du bloc courant, DISPLOF.....	2-110
2.10 Inhibition du bloc par bloc, SBLOF, SBLON (à partir de SW 4.3).....	2-111
2.11 Macroprogrammation.....	2-114
2.12 Exécution d'un sous-progr. externe (à partir du logiciel SW 4.2).....	2-116
2.13 Cycles : paramétrer des cycles utilisateur.....	2-119

2.1 Utilisation des sous-programmes

840 D
NCU 571840 D
NCU 572

FM-NC



810D

NCU 573

2.1 Utilisation des sous-programmes



Qu'est-ce qu'un sous-programme ?

En principe, la structure d'un sous-programme est la même que celle d'un programme pièce. Il se compose de blocs CN avec des instructions de déplacement et de commutation.

Sur le plan théorique, il n'y a pas de différence entre le programme principal et le sous-programme. Le sous-programme contient des séquences ou des phases d'usinage qui doivent être exécutées plusieurs fois.

Utilisation des sous-programmes

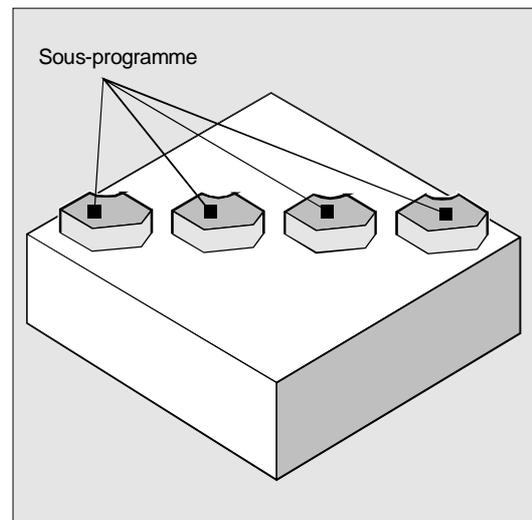
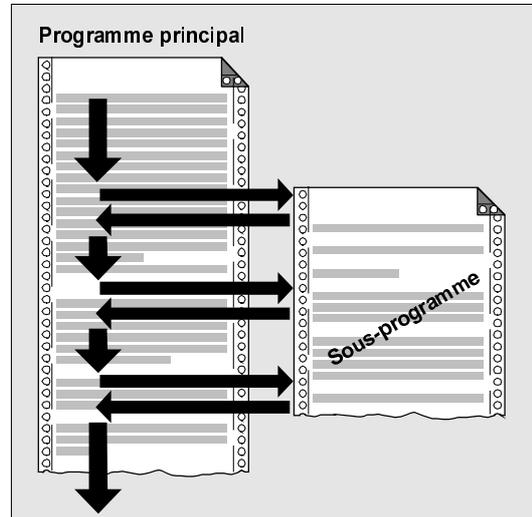
On ne programme qu'une seule fois dans un sous-programme les séquences qui reviennent régulièrement. Il s'agit, par exemple, de formes bien définies qui se répètent ou encore de cycles d'usinage.

Le sous-programme peut alors être appelé et exécuté dans tout programme principal.

Structure du sous-programme

La structure du sous-programme est identique à celle du programme principal.

Un en-tête de programme avec des définitions de paramètres peut être programmé en plus dans le sous-programme.



840 D
NCU 571840 D
NCU 572

FM-NC



810D

NCU 573

Profondeur d'imbrication

Imbrication de sous-programmes

Un sous-programme peut également contenir l'appel d'un sous-programme. dans lequel figure également un appel de sous-programme, etc.

Le nombre maximal de niveaux d'imbrication est de 12.

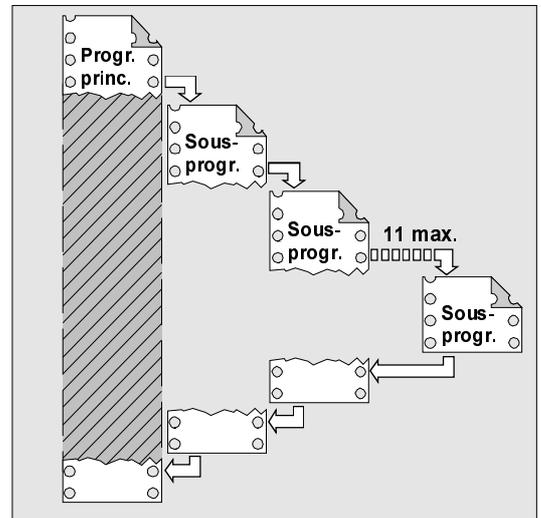
Cela signifie :

11 appels de sous-programme imbriqués peuvent émaner d'un programme principal.

Restrictions

L'appel de sous-programmes peut également figurer dans des routines d'interruption. Pour travailler avec des routines d'interruption, laisser quatre niveaux libres ou n'imbriquer que 7 appels de sous-programme.

Trois niveaux sont nécessaires pour les cycles d'usage et de mesure SIEMENS. Si un cycle doit être appelé par un sous-programme, cet appel est possible au niveau 5 au maximum (si 4 niveaux sont réservés pour les routines d'interruption).



2.2 Sous-programme avec mécanisme SAVE

840 D
NCU 571840 D
NCU 572

FM-NC

810D
NCU 573

2.2 Sous-programme avec mécanisme SAVE

Vous ajoutez ici l'instruction SAVE à l'instruction de définition PROC.

Avec cette fonction, les "données d'exploitation" telles que les fonctions G modales ou les frames globaux actuellement valides dans le programme principal, sont mises en mémoire lors de l'appel du sous-programme. L'état initial est rétabli automatiquement lors du retour au programme appelant.

Exemple :

Définition de sous-programme

```
PROC CONTOUR SAVE
N10 G91 ...
N100 M17
```

Programme principal

```
%123
N10 G0 X... Y... G90
N20...
N50 CONTOUR
N60 X... Y...
```

Dans le sous-programme CONTOUR, c'est G91, cotation relative, qui s'applique. Après le retour au programme principal, la cotation absolue s'applique à nouveau vu que les fonctions modales du programme principal ont été mises en mémoire avec SAVE.

840 D
NCU 571840 D
NCU 572
NCU 573

FM-NC



810D

2.3 Sous-programmes avec transfert de paramètres



Début de programme, PROC

Tout sous-programme qui, en cours d'exécution, doit faire appel à des paramètres figurant dans le programme d'où il émerge, doit être identifié par le mot-clé PROC.

Fin de programme M17, RET

Avec l'instruction M17, on signale la fin d'un sous-programme, tout en signifiant le retour dans le programme principal d'où il a été appelé.
Comme alternative à M17: Le mot-clé RET caractérise la fin d'un sous-programme, mais sans interruption du mode de déplacement tangentiel et sans sortie de fonction vers l'AP.



RET est à programmer dans un bloc CN qui lui est propre.

Exemple :

```
PROC CONTOUR
N10...
...
N100 M17
```

Transfert de paramètres entre programme principal et sous-programme

Si vous travaillez avec des paramètres dans le programme principal, vous pouvez aussi utiliser dans le sous-programme les valeurs calculées ou affectées du programme principal.
A cet effet, au moment de l'appel du sous-programme, les valeurs des **paramètres courants** du programme principal sont transférées vers les **paramètres formels** du sous-programme et y sont exploitées ensuite au cours de l'exécution du sous-programme.

2.3 Sous-programmes avec transfert de paramètres



840 D
NCU 571



840 D
NCU 572



FM-NC



810D
NCU 573

Exemple :

```
N10 DEF REAL LONGUEUR, LARGEUR
N20 LONGUEUR=12 LARGEUR=10
N30 CHASSIS (LONGUEUR, LARGEUR)
```

Les valeurs affectées dans N20 dans le programme principal sont transférées dans N30 lors de l'appel du sous-programme.

Le transfert des paramètres se fait dans l'ordre suivant.

Les noms des paramètres peuvent être différents dans le programme principal et le sous-programme.

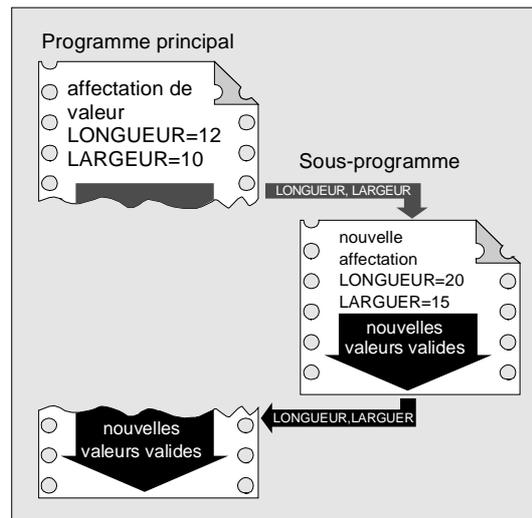
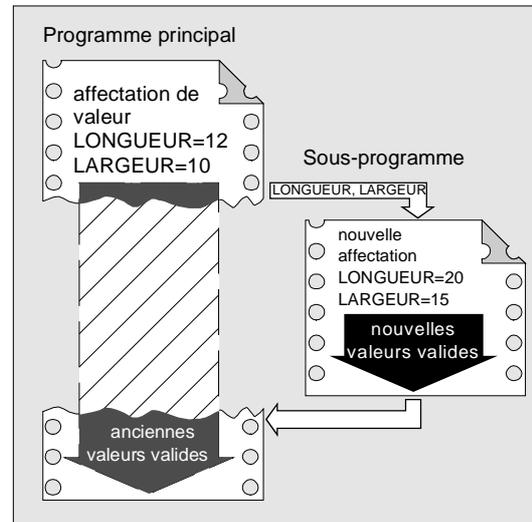
Deux possibilités de transfert des paramètres

Transfert de valeurs uniquement (Call-by-value)

Si les paramètres transférés subissent une modification au cours de l'exécution du sous-programme, ceci n'a aucun effet sur le programme principal. Les paramètres restent ici inchangés (voir schéma).

Transfert de paramètres avec échange de données (Call-by-reference)

Chaque modification de paramètres dans le sous-programme est suivie immédiatement d'une modification des paramètres dans le programme principal (voir schéma).



840 D
NCU 571840 D
NCU 572
NCU 573

FM-NC



810D



Programmation

Les paramètres concernés par le transfert doivent figurer avec leur type et leur nom au début du sous-programme.

Transfert de paramètres de type Call-by-value

```
PROC NOM_PROG(VARI_TYPE_1 VAR1,VAR_TYPE_2 VAR2,...)
```

Exemple :

```
PROC CONTOUR(REAL LONGUEUR, REAL LARGEUR)
```

Transfert de paramètres de type Call-by-reference, caractérisé par le mot-clé VAR

```
PROC NOM PROGRAMME(VAR TYPE VARIABLE1 VARIABLE1,VAR TYPE VARIABLE2 ...)
```

Exemple :

```
PROC CONTOUR(VAR REAL LONGUEUR, VAR REAL LARGEUR)
```

Transfert de tableaux de type Call-by-reference, caractérisé par le mot-clé VAR

```
PROC NOM PROGRAMME(VAR TYPE VARIABLE1 NOM TABLEAU1[taille du tableau],  
VAR TYPE VARIABLE2 NOM TABLEAU2[taille du tableau], VAR TYPE VARIABLE3  
NOM TABLEAU3[taille du tableau1, taille du tableau2], VAR TYPE VARIABLE4  
NOM TABLEAU4[ ],VAR TYPE VARIABLE5NOM TABLEAU5 [,taille du tableau])
```

Exemple :

```
PROC PALETTE(VAR INT TAB[,10])
```



Remarques complémentaires

L'instruction de définition avec PROC est à programmer dans un bloc CN séparé. Le transfert est limité à 127 paramètres maximum.

2.3 Sous-programmes avec transfert de paramètres



840 D
NCU 571



840 D
NCU 572



FM-NC



810D
NCU 573



Définition de tableaux

Règles de définition des paramètres formels :

Dans le cas des tableaux bidimensionnels, il n'est pas obligatoire d'indiquer le nombre de la 1ère dimension, mais par contre, il convient d'écrire la virgule.

Exemple :

```
VAR REAL TAB[ , 5 ]
```

Quand on indique une longueur de tableau indéfinie, les sous-programmes peuvent traiter des tableaux de longueur variable. Toutefois, lors de la définition des variables, il convient de préciser le nombre d'éléments à entrer.

Vous trouverez d'autres informations sur la définition des tableaux dans le manuel de programmation.



Exemple de programmation

Programmation avec longueurs de tableaux variables.

<code>%_N_PLAQUE_PERFOREE_MPF</code>	Programme principal
<code>DEF REAL TABLEAU[100,2]</code>	Définir le tableau des positions
<code>EXTERN RESEAU_TROUS (VAR REAL[, 2], INT)</code>	
<code>TABLEAU[0,0]=-17.5</code>	Définir les positions
...	
<code>TABLEAU[99,1]=45</code>	
<code>RESEAU_TROUS(TABLEAU,100)</code>	Appel du sous-programme
<code>M30</code>	

Réalisation d'un réseau de trous avec l'appui du transfert d'un tableau de positions de longueur variable

<code>%_N_RESEAU_TROUS_SPF</code>	Sous-programme
<code>PROC RES_TROUS(VAR REAL TAB[, 2], -> -> INT NOMBRE)</code>	Transfert de paramètres
<code>DEF INT COMPTEUR</code>	
<code>STEP: G1 X=TAB[COMPTEUR,0]-></code>	Gamme opératoire
<code>-> Y=TAB[COMPTEUR,1] F100</code>	
<code>Z=IC(-5)</code>	
<code>Z=IC(5)</code>	
<code>COMPTEUR=COMPTEUR+1</code>	
<code>IF COMPTEUR<NOMBRE GOTOB STEP</code>	
<code>RET</code>	Fin du sous-programme

840 D
NCU 571840 D
NCU 572

FM-NC

810D
NCU 573

2.4 Appel d'un sous-programme

Appel d'un sous-programme sans transfert de paramètres

Pour ce faire, dans le programme principal, appeler le sous-programme soit avec l'adresse L et le numéro du sous-programme, soit en indiquant le nom du sous-programme.

Exemple :

```
N10 L47 ou
N10 PIVOT_2
```

Sous-programme avec transfert de paramètres, explication avec l'instruction EXTERN

Les sous-programmes avec transfert de paramètres doivent figurer avec l'instruction EXTERN avant leur appel dans le programme principal, par exemple en début de programme.

Indiquer le nom du sous-programme et les types de variable dans l'ordre du transfert.

L'instruction EXTERN ne doit être indiquée que si le sous-programme se trouve dans le répertoire pièce ou dans le répertoire des sous-programmes globaux.

Il n'est pas nécessaire de déclarer les cycles avec l'instruction EXTERN.

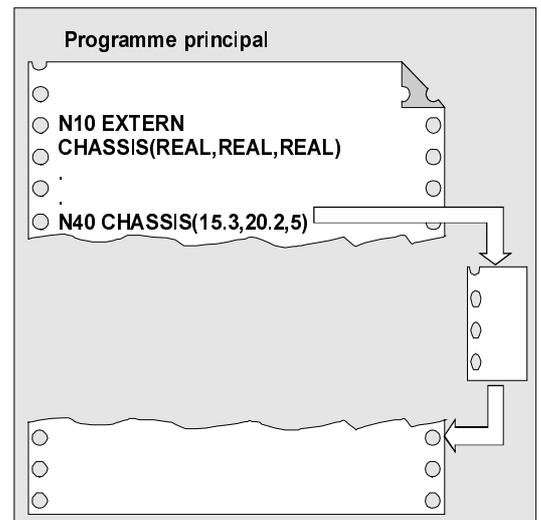
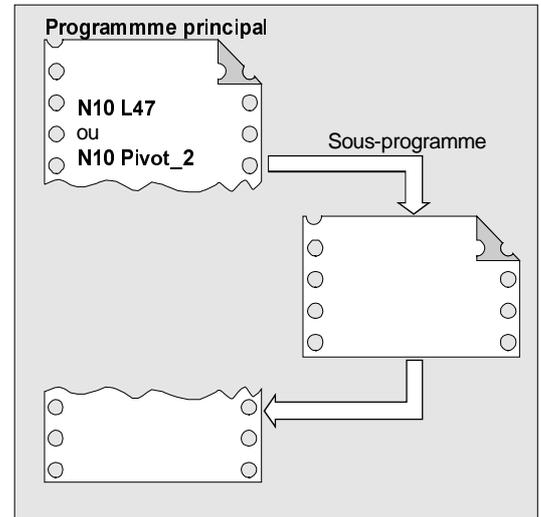
Instruction EXTERN

```
EXTERN NOM(TYPE1, TYPE2, TYPE3, ...) ou
EXTERN NOM(VAR TYPE1, VAR TYPE2, ...)
```

Exemple :

```
N10 EXTERN CHASSIS(REAL, REAL, REAL)
...
N40 CHASSIS(15.3, 20.2, 5)
```

N10 indication du sous-programme, N40 appel du sous-programme avec transfert de paramètres.



2.4 Appel d'un sous-programme



840 D
NCU 571



840 D
NCU 572



FM-NC



810D
NCU 573

Appel de sous-programme avec transfert de paramètres

Appelez le sous-programme dans le programme principal en précisant son nom et le transfert de paramètres. Vous pouvez transférer des variables ou directement des valeurs (sauf pour les paramètres VAR).

Exemple :

```
N10 DEF REAL LONGUEUR, LARGEUR,
PROFONDEUR
N20 ...
N30 LONGUEUR=15.3 LARGEUR=20.2
PROFONDEUR=5
N40
CHASSIS (LONGUEUR, LARGEUR, PROFONDEUR)
ou
N40 CHASSIS(15.3, 20.2, 5)
```

Equivalence entre définition du sous-programme et appel du sous-programme

Les types de variables et l'ordre de transfert doivent correspondre aux définitions qui ont été convenues sous PROC dans le nom du sous-programme. Les noms des paramètres peuvent être différents dans le programme principal et le sous-programme.

Exemple :

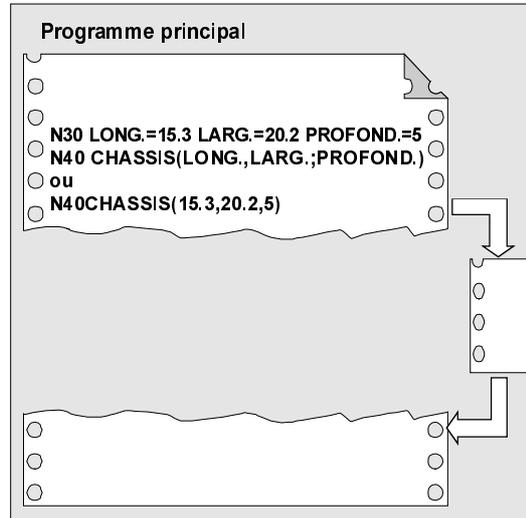
Définition dans le sous-programme :

```
PROC CHASSIS (REAL LONGUEUR, REAL LARGEUR, REAL PROFONDEUR)
```

Appel dans le programme principal :

```
N30 CHASSIS (LONGUEUR, LARGEUR,
PROFONDEUR)
```

Vous trouverez un exemple complet de programmation de sous-programmes avec transfert de paramètres dans le manuel de programmation "Notions complémentaires".



840 D
NCU 571840 D
NCU 572

FM-NC



810D

NCU 573

Transfert incomplet de paramètres

Lors de l'appel du sous-programme, on peut choisir de ne pas transférer certaines valeurs ou paramètres prescrits. Dans ce cas, on attribue la valeur **zéro** aux paramètres concernés dans le sous-programme.

La virgule est nécessaire pour indiquer l'ordre. La virgule n'est pas nécessaire après le dernier paramètre.

Retour au dernier exemple :

```
N40 CHASSIS(15.3, ,5)
```

La valeur médiane 20.2 a été éclipsée ici.

Remarque



Les paramètres courants de type AXIS ne doivent pas être éclipsés.

Les paramètres VAR doivent être transmis intégralement.

Appel d'un programme principal comme sous-programme

Un programme principal peut être appelé en tant que sous-programme. Dans ce cas, la fin du programme M2 ou M30 spécifiée dans le programme principal est traitée comme M17 (fin du programme avec retour au programme appelant).

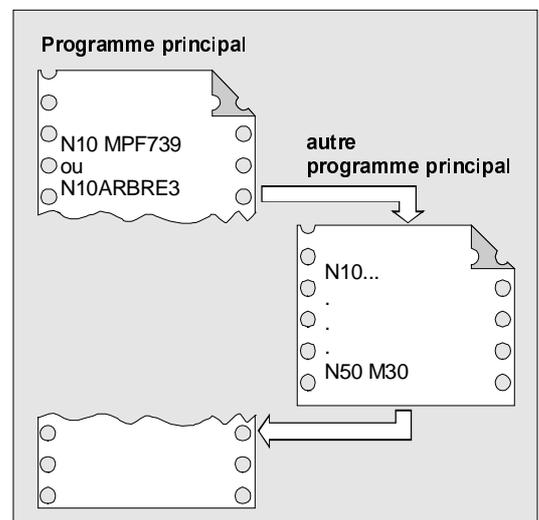
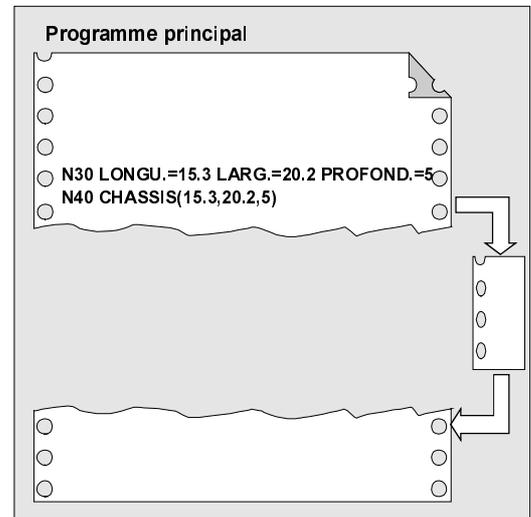
L'appel est programmé en indiquant le nom du programme.

Exemple :

```
N10 MPF739 ou  
N10 ARBRE3
```



De même, un sous-programme peut également être lancé comme programme principal.



2.5 Sous-programme à répétition



840 D
NCU 571



840 D
NCU 572



FM-NC



810D
NCU 573

2.5 Sous-programme à répétition

Répétition du programme, P

Si un sous-programme doit être exécuté plusieurs fois d'affilée, le nombre souhaité de répétitions peut être programmé sous l'adresse P, dans le bloc qui contient l'appel du sous-programme.

Exemple :

```
N40 CHASSIS P3
```

Le sous-programme Châssis doit être exécuté trois fois d'affilée.

Plage de valeurs :

P : 1...9999

Règle régissant chaque appel de sous-programme :



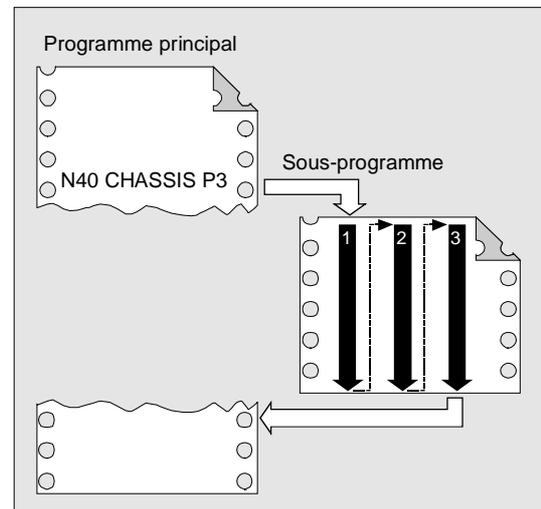
L'appel d'un sous-programme doit toujours être programmé dans un bloc CN spécifique.

Appel d'un sous-programme à répétition et transfert de paramètres



Les paramètres sont transférés uniquement lors de l'appel du sous-programme, c'est-à-dire à la première exécution. Les paramètres restent inchangés pour l'exécution des répétitions.

Si vous souhaitez modifier les paramètres dans les répétitions du sous-programme, il est nécessaire de définir des conventions correspondantes dans le sous-programme.



840 D
NCU 571840 D
NCU 572
NCU 573

FM-NC



810D

2.6 Sous-programme modal, MCALL

Appel d'un sous-programme à effet modal, MCALL

Cette fonction provoque l'appel et l'exécution automatiques du sous-programme, après chaque bloc à déplacement tangentiel.

Ceci permet d'automatiser l'appel de sous-programmes qui doivent être exécutés à différentes positions de la pièce, par exemple pour la réalisation de réseaux de trous.

Exemples :

```
N10 G0 X0 Y0
N20 MCALL L70
N30 X10 Y10
N40 X50 Y50
```

La position programmée est accostée dans les blocs N30 et N40 et suivie de l'exécution du sous-programme L70.

```
N10 G0 X0 Y0
N20 MCALL L70
N30 L80
```

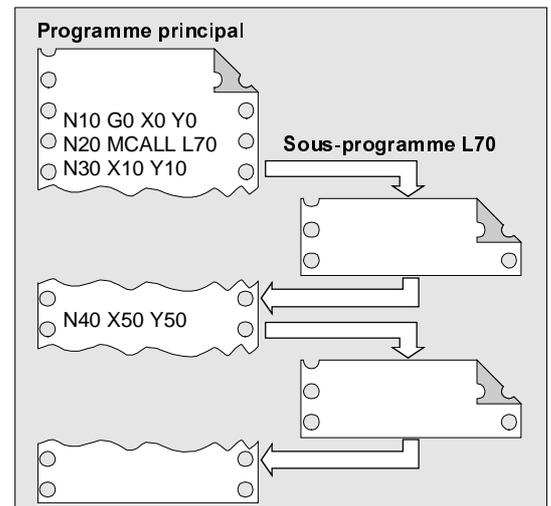
Dans cet exemple, les blocs CN dans lesquels sont programmés des axes à interpolation figurent dans le sous-programme L80. L70 est appelé par le biais de L80.



On ne peut faire figurer qu'un seul appel MCALL à la fois dans l'exécution d'un programme. Les paramètres sont transférés une seule fois lors de l'appel MCALL.

Désactivation de l'appel modal d'un sous-programme

Avec MCALL sans appeler de sous-programme ou en programmant un nouvel appel modal pour un nouveau sous-programme.



2.7 Appel indirect d'un sous-programme

840 D
NCU 571840 D
NCU 572
NCU 573

FM-NC



810D

2.7 Appel indirect d'un sous-programme

Appel indirect d'un sous-programme, CALL

Dans certaines conditions, il peut être nécessaire de faire appel à différents sous-programmes dans un endroit donné du programme.

Pour ce faire, indiquer le nom du sous-programme dans une variable du type STRING. L'appel du sous-programme s'effectue avec CALL et le nom de la variable.



L'appel indirect de sous-programmes ne s'adresse qu'à des sous-programmes sans transfert de paramètres.



Pour l'appel direct d'un sous-programme, rangez le nom de ce dernier dans une constante de type STRING.

Exemple :

Appel direct avec constante de type STRING :

```
CALL "/_N_WKS_DIR/_N_SOUSPROG_WPD/_N_PIECE1_SPF"
```

Appel indirect par le biais d'une variable :

```
DEF STRING[100] NOM_PROG
NOM_PROG="/_N_SCP_DIR/_N_SOUSPROG_WPD/_N_PIECE1_SPF"
CALL NOM_PROG
```

Le sous-programme PIECE1 est affecté à la variable NOM_PROG. Le sous-programme est appelé indirectement avec CALL et l'indication du chemin.

2.8 Appel d'un sous-prog. avec indication de chemin et paramètres

840 D
NCU 571840 D
NCU 572
NCU 573

FM-NC



810D

2.8 Appel d'un sous-prog. avec indication de chemin et paramètres PCALL

PCALL permet d'appeler des sous-programmes avec indication de chemin absolu et transfert de paramètres. :

```
PCALL chemin/nom_programme(paramètre 1, ..., paramètre n)
```



Signification

PCALL

Mot-clé pour l'appel d'un sous-programme avec indication de chemin absolu

Nom_chemin

Indication du chemin absolu débutant par "/", y compris nom des sous-programmes
Si aucune indication de chemin absolu n'est faite, PCALL se comporte comme un appel standard de sous-programme avec descripteur de programme.
Le descripteur de programme est indiqué sans texte préliminaire `_N_` et sans extension.
Si le nom du programme doit être programmé avec préliminaire et extension, il convient de le spécifier de manière explicite avec texte préliminaire et extension comme instruction EXTERN.

Paramètres 1 à n

Paramètre courant satisfaisant à l'instruction PROC du sous-programme.

Exemple :

```
PCALL/_N_SCP_DIR/_N_ARBRE_WPD/_N_ARBRE:MPF( paramètre1, paramètre2, ...)
```

2.9 Inhibition de l'affichage du bloc courant, DISPLOF

840 D
NCU 571840 D
NCU 572

FM-NC

810D
NCU 573

2.9 Inhibition de l'affichage du bloc courant, DISPLOF



Programmation

PROC ... DISPLOF



Fonction

DISPLOF permet d'inhiber l'affichage du bloc courant pour un sous-programme donné. DISPLOF figure à la fin de l'instruction PROC.

C'est l'appel du cycle ou du sous-programme qui est affiché à la place du bloc courant.

En version standard, c'est l'affichage des blocs qui est activée. La désactivation de l'affichage des blocs avec DISPLOF reste en vigueur jusqu'à la sortie du sous-programme ou la fin du programme. Si, depuis ce sous-programme auquel on a attribué DISPLOF, vous appelez d'autres sous-programmes, l'affichage du bloc courant sera inhibé de la même façon dans ces sous-programmes. Par contre, quand un sous-programme avec inhibition de l'affichage du bloc courant est interrompu par un sous-programme asynchrone, les blocs du sous-programme courant seront affichés.



Exemple de programmation

Inhiber l'affichage du bloc courant dans le cycle

```

%_N_CYCLE_SPF
; $PATH=/_N_CUS_DIR
PROC CYCLE (AXIS TOMOV, REAL POSITION) SAVE DISPLOF
;inhiber l'affichage du bloc courant
;maintenant c'est l'appel du cycle qui est
affiché comme bloc courant
;par ex. CYCLE(X, 100.0)
DEF REAL DIFF ;contenu des cycles
G01 ... ;
...
RET ;retour du sous-programme dans le pro-
gramme appelant; le bloc suivant du pro-
gramme appelant est de nouveau affiché

```

840 D
NCU 571840 D
NCU 572
NCU 573

810D

2.10 Inhibition du bloc par bloc, SBLOF, SBLON (à partir de SW 4.3)



Programmation

```
PROC ... SBLOF
PROC ... SBLON
```

Désactivation du bloc par bloc

Réactivation du bloc par bloc



Fonction

Inhibition du bloc par bloc pour un programme

Dans tous les types de mode bloc par bloc, les programmes identifiés par SBLOF sont exécutés entièrement comme un bloc. SBLOF figure dans la ligne PROC et est actif jusqu'à la fin ou l'abandon du sous-programme.

SBLOF est également actif dans les sous-programmes appelés.

Exemple :

```
PROC EXEMPLE SBLOF
G1 X10
RET
```

Inhibition du bloc par bloc dans un programme

SBLOF peut figurer seul dans le bloc. A partir de ce bloc, le mode bloc par bloc est désactivé

- jusqu'au prochain SBLON ou
- jusqu'à la fin du niveau de sous-programme actif.

Exemple :

```
N10 G1 X100 F1000
N20 SBLOF
N30 Y20
N40 M100
N50 R10=90
N60 SBLON
N70 M110
N80 ...
```

Désactivation du bloc par bloc

Réactivation du bloc par bloc



En mode bloc par bloc, la section de programme entre N20 et N60 est exécutée comme un pas.

840 D
NCU 571840 D
NCU 572
NCU 573

810D

Blocage du bloc par bloc pour des sous-programmes asynchrones

Les sous-programmes asynchrones ASUP1.SYF et ASUP2.SYF lancés par la CN lors de Repos doivent être exécutés pas par pas. A partir de SW 4.3, ces sous-programmes peuvent être exécutés comme un pas en programmant SBLOF.

Exemple :

```
N10 SBLOF
N20 IF $AC_ASUP== 'H200 '
N30 RET
N40 ELSE
N50 REPOSA
N60 ENDIF
N70 RET
```

pas de Repos si chang.mode fonct.

Repos dans tous les autres cas

Conditions marginales

- DISPLOF permet d'inhiber l'affichage du bloc courant pour un cycle.
- Si DISPLOF est programmé en liaison avec SBLOF, l'appel de cycle est encore affiché en cas d'arrêts "bloc par bloc" dans le cycle.
- Le pré réglage effectué avec le PM 20117: IGNORE_SINGLEBLOCK_ASUP pour le comportement de sous-programmes asynchrones en cas de bloc par bloc peut être écrasé en programmant SBLOF pour les sous-programmes.
- SBLOF peut être inhibé avec une variable OPI à des fins de test (voir documentation OEM).

840 D
NCU 571840 D
NCU 572
NCU 573

810D



Exemple de programmation 1

Un cycle doit apparaître comme une instruction pour le programmeur

Programme principal

```

N10 G1 X10 G90 F200
N20 X-4 Y6
N30 CYCLE1
N40 G1 X0
N50 M30

```

Programme CYCLE1

```

N100 PROC CYCLE1 DISPLOF SBLOF          inhibition du bloc par bloc
N110 R10=3*SIN(R20)+5
N120 IF (R11 <= 0)
N130     SETAL(61000)
N140 ENDIF
N150 G1 G91 Z=R10 F=R11
N160 RET

```



Le programme CYCLE1 est exécuté comme un pas lorsque bloc par bloc est actif.



Exemple de programmation 2

Un ASUP lancé par l'AP pour l'activation de décalages d'origine et de corrections d'outil modifiés ne doit pas être visible.

```

N100 PROC NV SBLOF DISPLOF
N110 CASE $P_UIFRNUM OF 0 GOTOF _G500
      -->1 GOTOF _G54 2 GOTOF _G55 3
      -->GOTOF _G56 4 GOTOF _G57
      -->DEFAULT GOTOF END
N120 _G54: G54 D=$P_TOOL T=$P_TOOLNO
N130 RET
N140 _G54: G55 D=$P_TOOL T=$P_TOOLNO
N150 RET
N160 _G56: G56 D=$P_TOOL T=$P_TOOLNO
N170 RET
N180 _G57: G57 D=$P_TOOL T=$P_TOOLNO
N190 RET
N200 END: D=$P_TOOL T=$P_TOOLNO
N210 RET

```

2.11 Macroprogrammation

840 D
NCU 571840 D
NCU 572

FM-NC



810D

NCU 573

2.11 Macroprogrammation

Qu'est-ce qu'une macro ?

Une macro réunit différentes instructions en une nouvelle instruction globale qui a son propre nom. Des fonctions G, M et H et des sous-programmes L peuvent également être définis sous forme de macros.

En appelant la macro pendant l'exécution du programme, les instructions programmées dans la macro sont exécutées l'une après l'autre.

Utilisation des macros

Les suites d'instructions qui reviennent souvent sont programmées une seule fois sous la forme d'une macro-instruction dans un bloc spécifique ou bien une seule fois en début du programme.

La macro-instruction peut alors être appelée et exécutée dans tout programme principal ou sous-programme.

Programmation

Les macros sont identifiées avec le mot-clé `DEFINE...AS`.

La définition de la macro est :

```
DEFINE NOM AS séquence d'instructions
```

Exemple :

Définition d'une macro :

```
DEFINE LIGNE AS G1 G94 F300
```

Appel dans le programme CN :

```
N20 LIGNE X10 Y20
```

840 D
NCU 571840 D
NCU 572
NCU 573

FM-NC



810D



Remarques complémentaires

L'imbrication de macros n'est pas possible.



Les mots-clés et les noms réservés ne doivent pas faire l'objet d'une surdéfinition avec des macros.



Les macros peuvent entraîner une forte modification du langage de programmation de la CN. Par conséquent, utiliser les macros avec précaution.



Des macros peuvent également être convenues dans le programme CN. Seuls des descripteurs sont autorisés comme noms de macros. Des macros de fonctions G ne peuvent être définies qu'au niveau de la CN, dans un module de macros.



Avec les macro-instructions, on peut définir des descripteurs, des fonctions G, M, H et des sous-programmes L quelconques.



Exemple de programmation

Dans ce chapitre, vous trouverez quelques exemples possibles pour la définition de macros.

```
DEFINE M6 AS L6
```

Lors du changement d'outil, un sous-programme est d'abord appelé, qui assure le transfert des données requis. La fonction M de changement d'outil proprement dite est alors exécutée dans le sous-programme (par exemple M106).

```
DEFINE G81 AS DRILL( 81 )
```

Simulation de la fonction G DIN.

```
DEFINE G33 AS M333 G333
```

La synchronisation avec l'AP est requise pour le filetage. La fonction G33 initiale a été renommée en G333 par un paramètre machine, si bien que la programmation ne change pas pour l'utilisateur.

2.12 Exécution d'un sous-progr. externe (à partir du logiciel SW 4.2)840 D
NCU 571840 D
NCU 572
NCU 573

810D

2.12 Exécution d'un sous-progr. externe (à partir du logiciel SW 4.2)

Cette fonction s'applique uniquement aux MMC 102/103.

Avec EXTCALL, vous pouvez charger un programme du MMC 102/103 dans le mode "Exécution d'un programme de provenance externe".

EXTCALL chemin/nom du programme

**Signification**

EXTCALL

Nom_chemin

nom du programme

Exemple :

EXTCALL "ARBRE" ou EXTCALL"/_N_SCP_DIR/_N_ARBRE_WPD/ARBRE"

Mot-clé pour appel de sous-programme

facultatif

Constante/Variable de type STRING.

Indication du chemin absolu débutant par "/",

Le descripteur de programme est indiqué

avec/sans texte préliminaire _N_ et sans

extension. Une extension peut être rattachée

au nom du programme avec le caractère "<".

**Fonction**

Dans le cas de l'usinage d'une pièce complexe, on peut avoir des séquences de programmation pour certaines étapes d'usinage qui, en raison de leur volume, ne peuvent pas être rangées directement dans la mémoire de travail.

Avec EXTCALL, vous pouvez charger un programme du MMC 102/103 dans le mode "Exécution d'un programme de provenance externe".

Vous pouvez charger de cette façon tous les programmes qui sont accessibles par le biais de la structure des répertoires du MMC102.

SD 42700 EXT_PROG_PATH

La donnée de réglage SD 42700 EXT_PROG_PATH, spécifique à un canal, permet de régler avec souplesse le chemin d'appel.

2.12 Exécution d'un sous-progr. externe (à partir du logiciel SW 4.2)



840 D
NCU 571



840 D
NCU 572
NCU 573



810D

La donnée SD 42700 contient une indication de chemin qui, associée au descripteur de sous-programme, forme le nom de chemin absolu du programme à appeler.

Quand on appelle un sous-programme externe sans indication de chemin absolu, le même chemin de recherche sera parcouru sur le MMC que lors de l'appel d'un sous-programme dans la mémoire de travail.

Mémoire de chargement réglable (tampon FIFO)

Pour l'exécution d'un programme en mode "Exécution d'un programme de provenance externe" (programme principal ou sous-programme), on a besoin d'une mémoire de chargement dans le NCK. La capacité de la mémoire de chargement est prééglée sur 30Ko.

Elle peut être modifiée dans le PM 18360
EXT_PROG_BUFFER_SIZE.

POWER ON, RESET

Avec Reset et POWER ON, on efface des appels de sous-programmes externes et on efface les mémoires de chargement respectives.

Remarques complémentaires

Les sous-programmes externes ne doivent pas contenir des instructions de saut telles que GOTO, GOTOB, CASE, IF - ELSE, FOR, LOOP, WHILE ou REPEAT. L'appel de sous-programmes est possible.

Exemple de programmation

Dans la donnée de réglage \$SC_EXT_PROG_PATH, on a introduit le chemin suivant : "_N_WKS_DIR/_N_WST1". Le programme principal _N_MAIN_MPF figure dans la mémoire de travail et y est sélectionné.

```
%_N_MACHINE1_MPF
-----
N10 PROC MAIN
-----
N20 ...
-----
N30 EXTCALL EBAUCHER_SPF                ; appel du sous-programme externe
                                           EBAUCHER_SPF
-----
N40 ...
-----
N50 M30
-----
```

2.12 Exécution d'un sous-progr. externe (à partir du logiciel SW 4.2)840 D
NCU 571840 D
NCU 572
NCU 573

810D

Le sous-programme EBAUCHER_SPF (figure dans la structure des répertoires du MMC sous Pièces ->WST1) :

N10 PROC EBAUCHER

N20 G1 F1000

N30 X=... Y=... Z=...

N40 ...

N90 M17

840 D
NCU 571840 D
NCU 572
NCU 573

810D

2.13 Cycles : paramétrer des cycles utilisateur

Fichiers et chemins d'accès



Signification

cov.com	Liste des cycles
uc.com	Description des appels de cycle



Fonction

Ces fichiers permettent à l'utilisateur de paramétrer des cycles spécifiques.



Procédure

Le fichier cov.com est livré avec les cycles standard et peut être étendu. Le fichier uc.com doit être créé par l'utilisateur.

Ces deux fichiers doivent être chargés dans le système passif de fichiers, dans le répertoire "Cycles utilisateur" ou pourvus de l'indication correspondante de chemin :

```
; $PATH=/_N_CST_DIR
```

dans le programme.

2.13 Cycles : paramétrer des cycles utilisateur

840 D
NCU 571840 D
NCU 572
NCU 573

810D

Adaptation de cov.com - liste des cycles

Le fichier cov.com fourni avec les cycles standard a la structure suivante :

<code>%_N_COV_COM</code>	Nom du fichier
<code>;\$PATH=/_N_CUS_DIR</code>	Indication du chemin
<code>;Vxxx 11.12.95 Sca Liste des cycles</code>	Ligne de commentaires
<code>C1(CYCLE81) Perçage, centrage</code>	Appel du 1er cycle
<code>C2(CYCLE82) Perçage, lamage</code>	Appel du 2me cycle
<code>...</code>	
<code>C24(CYCLE98) Concaténation de filetages</code>	Appel du dernier cycle
<code>M17</code>	Fin de fichier

Pour tout nouveau cycle à ajouter, il y a lieu d'insérer une ligne dont la syntaxe est la suivante :

`C<numéro> (<nom du cycle>) texte de commentaire`

Numéro : un nombre entier quelconque qui n'a pas encore été utilisé dans le fichier ;

Nom du cycle : le nom du cycle à insérer

Texte de commentaire : un commentaire au choix, relatif au cycle

Exemple :

```
C25 (MON_CYCLE_1) Cycle_utilisateur_1
C26 (CYCLE_SPECIAL)
```

840 D
NCU 571840 D
NCU 572
NCU 573

810D

Exemple pour fichier uc.com - description des cycles utilisateur

L'explication qui suit est le développement de
l'exemple précédent :

Exemple :

Pour les deux cycles suivants, un nouveau
paramétrage doit être créé :

```

PROC MON_CYCLE_1 (REAL PAR1, INT PAR2, CHAR PAR3, STRING[10] PAR4)
;Le cycle contient les paramètres suivants :
;
;PAR1:          réel compris dans la plage -1000.001 <= PAR2 <= 123.456, pré réglage
                100
;PAR2:          nombre entier positif compris dans la plage 0 <= PAR3 <= 999999,
                pré réglage 0
;PAR3:          1 caractère ASCII
;PAR4:          chaîne de caractères de longueur 10 pour un nom de sous-programme
;
...
M17

```

```

PROC CYCLE_SPECIAL (REAL VALEUR1, INT VALEUR2)
;Le cycle contient les paramètres suivants :
;
;VALEUR1:       réel sans limitation de plage de valeurs ni pré réglage
;VALEUR2:       nombre entier sans limitation de plage de valeurs ni pré réglage
;
...
M17

```

2.13 Cycles : paramétrer des cycles utilisateur840 D
NCU 571840 D
NCU 572
NCU 573

810D

Fichier uc.com correspondant

```

%_N_UC_COM
; $PATH=/_N_CUS_DIR
//C25 (MON_CYCLE_1) Cycle_utilisateur_1
(R/-1000.001 123.456 / 100 /paramètre_2 du cycle)
(I/0 999999 / 1 / nombre entier)
(C// "A" / caractère)
(S///nom du sous-programme)

//C26(CYCLE_SPECIAL)
(R///longueur totale)
(I/*123456/3/type d'usinage)
M17

```

**Description de la syntaxe pour le fichier uc.com -
description des cycles utilisateur****Ligne d'en-tête propre à chaque cycle :**

identique au fichier cov.com, précédée de "///"

//C <numéro> (<nom du cycle>) texte de commentaire

Exemple :

//C25 (MON_CYCLE_1) Cycle_utilisateur_1

840 D
NCU 571840 D
NCU 572
NCU 573

810D

Ligne de description pour chaque paramètre :

(<identificateur du type de donnée> / <valeur minimale> <valeur maximale> (<préréglage> / <commentaire>

Identificateurs pour type de donnée :

R	pour un réel
I	pour un entier
C	pour un caractère
S	pour une chaîne de caractères

Valeur minimale, maximale (facultatif)

Valeurs limites de la valeur à introduire contrôlées lors de l'introduction ; les valeurs en dehors de cette plage ne peuvent pas être introduites.

On peut aussi indiquer des valeurs de sélection auxquelles on pourra accéder avec la touche de basculement ; celles-ci seront scrutées en commençant par "*", d'autres valeurs ne sont alors pas autorisées.

Exemple :

(I/*123456/1/type d'usinage)

Pour les types Chaîne de caractères et caractère, il n'y a pas de limite ;

Préréglage (facultatif)

C'est la valeur qui est préréglée dans le masque correspondant lors de l'appel du cycle ; elle peut être modifiée par l'opérateur.

Commentaire

C'est un texte comportant 50 caractères au maximum qui est affiché dans le masque d'appel du cycle, devant le champ d'introduction du paramètre.

2.13 Cycles : paramétrer des cycles utilisateur



840 D
NCU 571



840 D
NCU 572
NCU 573



810D

Exemple d'affichage pour les deux cycles

Masque d'affichage pour le cycle `MON_CYCLE_1`

Paramètre 2 du cycle	100
Entier	1
Paramètre caractère	
Sous-programmes	

Masque d'affichage pour le cycle `CYCLE_SPECIAL`

Longueur totale	100
Type d'usinage	1

Gestion des fichiers et des programmes

3.1 Vue d'ensemble	3-126
3.2 Mémoire de programmes	3-127
3.3 Mémoire de travail.....	3-132
3.4 Définition des données utilisateur	3-135
3.5 Définition des niveaux de protection des données utilisateur, (GUD).....	3-138
3.6 Définition des macros	3-141

3.1 Vue d'ensemble



840 D
NCU 571



840 D
NCU 572



FM-NC



810D

NCU 573

3.1 Vue d'ensemble



Structure de la mémoire

L'utilisateur dispose d'une mémoire subdivisée en deux domaines :

1. La mémoire de travail

La mémoire de travail contient les données systèmes et les données utilisateurs courantes, qui permettent à la commande de fonctionner (système actif de fichiers).

Exemple :

Paramètres machine actifs, données de correction d'outil, décalages d'origine

2. La mémoire de programmes

Comme son nom l'indique, la mémoire de programmes contient les fichiers et les programmes qui y sont stockés à demeure (système passif de fichiers).

Exemple :

Programmes principaux et sous-programmes, définition des macros

840 D
NCU 571840 D
NCU 572

FM-NC



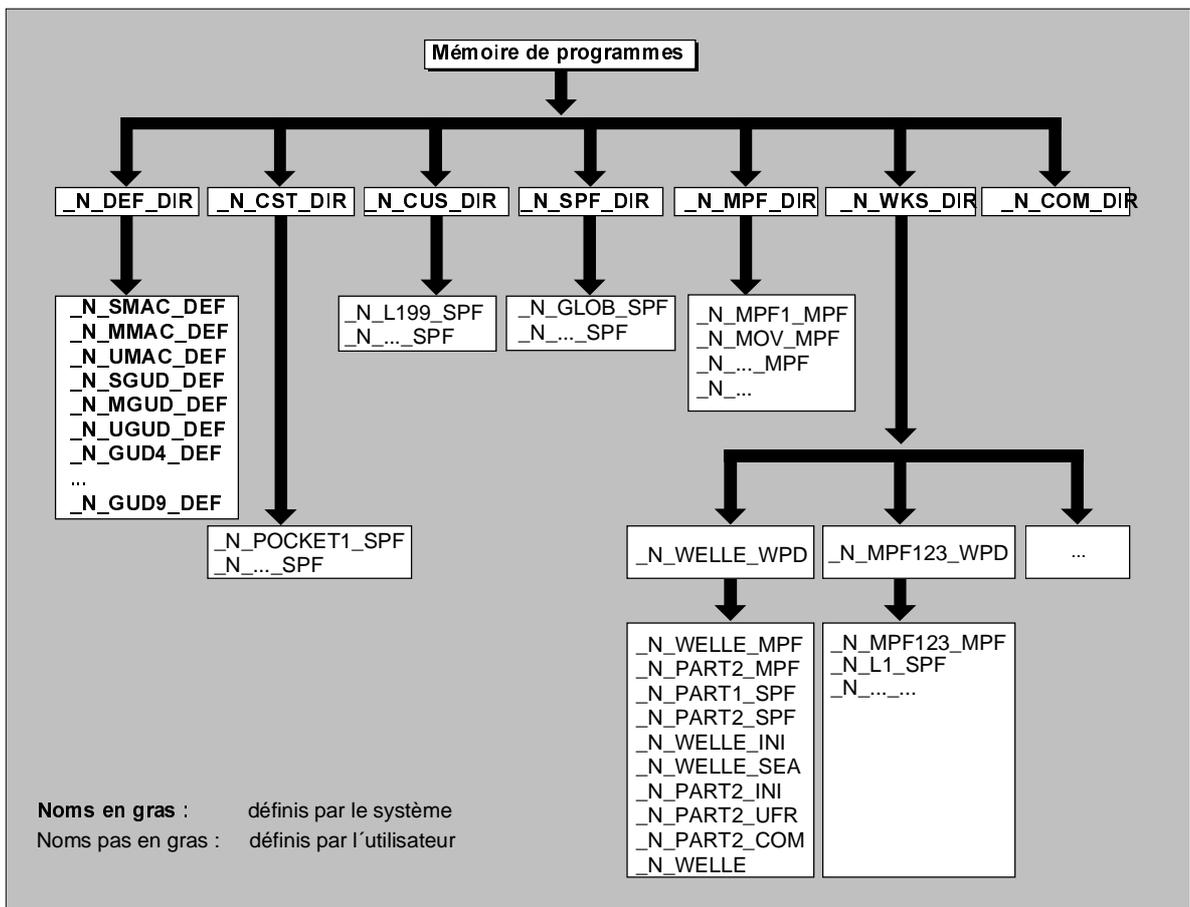
810D

NCU 573

3.2 Mémoire de programmes

Vue d'ensemble

Les programmes principaux et les sous-programmes sont stockés dans la mémoire de programmes. A coté de ceux-ci, il y a un certain nombre de types de fichiers qui peuvent y être stockés de façon provisoire pour être transférés, sur demande (par ex. lors de la réalisation d'une pièce déterminée) dans la mémoire de travail (par ex. pour des raisons d'initialisation).



3.2 Mémoire de programmes



840 D
NCU 571



840 D
NCU 572



FM-NC



810D
NCU 573

Répertoires

Les répertoires/catalogues suivants sont disponibles en standard en cas de raccordement d'une unité d'affichage et de commande :

1. <code>_N_DEF_DIR</code>	Blocs de données et blocs macros (affectation à la mise en service)
2. <code>_N_CST_DIR</code>	Cycles standard (affectation à la mise en service)
3. <code>_N_CUS_DIR</code>	Cycles utilisateur (affectation à la mise en service)
4. <code>_N_SCP_DIR</code>	Pièces
5. <code>_N_SPF_DIR</code>	Sous-programmes globaux
6. <code>_N_MPF_DIR</code>	Répertoire standard des programmes principaux
7. <code>_N_COM_DIR</code>	Répertoire standard des commentaires

Types de fichiers

Les types de fichiers suivants peuvent être introduits dans la mémoire de programmes :

<code>nom_MPF</code>	Programme principal
<code>nom_SPF</code>	Sous-programme
<code>nom_TEA</code>	Paramètres machine
<code>nom_SEA</code>	Données de réglage
<code>nom_TOA</code>	Correcteurs d'outils
<code>nom_UFR</code>	Décalages d'origine / frames
<code>nom_INI</code>	Fichier d'initialisation
<code>nom_GUD</code>	Données utilisateur globales
<code>nom_RPA</code>	Paramètres R
<code>nom_COM</code>	Commentaire
<code>nom_DEF</code>	Définitions pour données utilisateurs globales et macros

840 D
NCU 571840 D
NCU 572

FM-NC



810D

NCU 573

Catalogue de pièces, `_N_SCP_DIR`

Le catalogue des pièces figure en version standard dans la mémoire de programmes sous la désignation `_N_SCP_DIR`.

Il contient les répertoires pièce pour toutes les pièces que vous avez programmées.

Répertoires pièce, Identificateur WPD

Pour une gestion plus aisée des données et des programmes, il est possible de regrouper certaines données et certains programmes ou de les ranger dans des répertoires pièce distincts.

Un répertoire pièce contient tous les fichiers qui sont nécessaires pour l'usinage d'une pièce.

Ceux-ci peuvent être des programmes principaux, des sous-programmes, tous programmes d'initialisation et fichiers de commentaires.

Exemple :

Le répertoire pièce `_N_ARBRE_WPD`, qui a été créé pour la pièce `ARBRE`, contient les fichiers suivants :

<code>_N_ARBRE_MPF</code>	Programme principal
<code>_N_PART2_MPF</code>	Programme principal
<code>_N_PART1_SPF</code>	Sous-programme
<code>_N_PART2_SPF</code>	Sous-programme
<code>_N_ARBRE_INI</code>	Programme général d'initialisation des données pour la pièce
<code>_N_ARBRE_SEA</code>	Programme d'initialisation des données de réglage
<code>_N_PART2_INI</code>	Programme général d'initialisation des données pour le programme pièce 2
<code>_N_PART2_UFR</code>	Programme d'initialisation des données frame pour le programme pièce 2
<code>_N_ARBRE_COM</code>	Fichier de commentaires

3.2 Mémoire de programmes



840 D
NCU 571



840 D
NCU 572



FM-NC



810D
NCU 573

Création d'un répertoire pièce sur un PC externe

La procédure décrite ci-dessous est exécutée à un ordinateur personnel externe.

Vous trouverez toutes les informations utiles pour gérer les fichiers et les programmes (du PC à la commande) à partir de la commande, dans votre manuel d'utilisation.

;instruction \$PATH

Dans la deuxième ligne du fichier on spécifie le chemin de destination sous \$PATH=....

Exemple :

```
; $PATH=/_N_SCP_DIR/_N_ARBRE_WPD
```

Le fichier est rangé sous le chemin indiqué.

Important

Si le chemin n'est pas indiqué, les fichiers de type SPF sont rangés dans /_N_SPF_DIR,, les fichiers avec terminaison _INI dans la mémoire de travail et tous les autres dans /_N_MPF_DIR.

Exemple avec indication de chemin pour l'exemple précédent ARBRE :

```
%_N_ARBRE_MPF
; $PATH=/_N_SCP_DIR/_N_ARBRE_WPD
N10 G0 X... Z...
.
M2
.
.
%_N_ARBRE_SPF
```

840 D
NCU 571840 D
NCU 572

FM-NC

810D
NCU 573

Sélection d'une pièce pour son usinage

Un répertoire pièce peut être sélectionné dans un canal en vue de son usinage.

S'il existe dans ce répertoire un programme principal **de même nom**, il sera activé automatiquement pour l'usinage.

Exemple :

Nous supposons que le répertoire pièce

`/_N_SCP_DIR/_N_ARBRE_WPD` contienne les fichiers `_N_ARBRE_SPF` et `_N_ARBRE_MPF`.

Chemin de recherche pour l'appel d'un sous-programme

Si le chemin d'appel n'est pas indiqué de façon explicite dans le programme pièce lors de l'appel d'un sous-programme (ou d'un fichier d'initialisation), la recherche de ce sous-programme appelé sera faite suivant un chemin prédéfini.

Exemple d'appel d'un sous-programme avec indication de chemin absolu.

```
CALL "/_N_CST_DIR/_N_CYCLE1_SPF"
```

En règle générale, les programmes sont appelés sans indication de chemin :

Exemple :

```
CYCLE1
```

Ordre de recherche du chemin

1. répertoire courant / *nom*
2. répertoire courant / *nom_SPF*
3. répertoire courant / *nom_MPF*
4. `/_N_SPF_DIR` / *nom_SPF*
5. `/_N_CUS_DIR` / *nom_SPF*
6. `/_N_CST_DIR` / *nom_SPF*

Catalogue des pièces ou
répertoire standard `_N_MPF_DIR`

sous-programmes globaux
cycles utilisateur
cycles standard

3.3 Mémoire de travail

840 D
NCU 571840 D
NCU 572

FM-NC



810D

NCU 573

3.3 Mémoire de travail

Programmes d'initialisation

Les programmes d'initialisation servent à l'initialisation des données de la mémoire de travail.

Les types de fichier suivants peuvent être utilisés :

<i>nom</i> _TEA	paramètres machine
<i>nom</i> _SEA	données de réglage
<i>nom</i> _TOA	corrections d'outils
<i>nom</i> _UFR	décalages d'origine / frames
<i>nom</i> _INI	fichiers d'initialisation
<i>nom</i> _GUD	données utilisateur globales
<i>nom</i> _RPA	paramètres R

Zones de données

On peut intégrer les données dans différentes zones au sein desquelles elles doivent être en vigueur.

Ainsi, une commande peut disposer de plusieurs canaux (exceptions : SINUMERIK FM-NC, 810D, 840D NCU 571) ou d'une manière générale de plusieurs axes aussi. La répartition est la suivante :

Zones de données	Identificateur de la zone de données
données spécifiques NCK	NCK
données spécifiques à un canal (n indique le numéro du canal)	CH<n>
données spécifiques à un axe (n indique le numéro de l'axe machine)	AX<n>
données d'outils	TO
toutes les données	COMPLETE

840 D
NCU 571840 D
NCU 572

FM-NC

810D
NCU 573

Création d'un programme d'initialisation sur PC externe

A l'aide des identificateurs des zones de données et des types de données, on peut déterminer les zones à considérer comme un ensemble lors de la sauvegarde des données.

Exemple :

<code>_N_AX5_TEA_INI</code>	paramètres machine pour l'axe 5
<code>_N_CH2_UFR_INI</code>	frames du canal 2
<code>_N_COMPLETE_TEA_INI</code>	tous les paramètres machine

Après mise en service de la commande, un enregistrement garantissant un fonctionnement correct de cette dernière est présent dans la mémoire de travail.

Sauvegarde des programmes d'initialisation

Les fichiers de la mémoire de travail peuvent être sauvegardés dans un PC externe et réimportés dans la commande à partir de ce dernier.

- Les fichiers sont sauvegardés avec `COMPLETE`.
- Avec `INITIAL` un fichier INI :
(`_N_INITIAL_INI`) est créé, qui couvre toutes les zones.

Chargement des programmes d'initialisation

Les programmes INI peuvent également être sélectionnés et appelés en tant que programmes pièce si on n'utilise que les données d'un seul canal. Ceci constitue une autre manière d'initialiser des données.



Vous trouverez des informations sur tous les types de fichier dans le manuel d'utilisation.

3.3 Mémoire de travail

840 D
NCU 571840 D
NCU 572

FM-NC



810D

NCU 573

Procédure dans le cas des commandes multicanaux

CHANDATA(numéro de canal) pour plusieurs canaux est uniquement autorisé dans le fichier

N_INITIAL_INI.

N_INITIAL_INI est le fichier de mise en service avec lequel toutes les données de la commande sont initialisées.

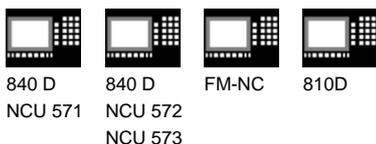
Exemple :

```
%_N_INITIAL_INI
CHANDATA(1)
; affectation des axes machine au canal 1
$MC_AXCONF_MACHAX_USED[0]=1
$MC_AXCONF_MACHAX_USED[1]=2
$MC_AXCONF_MACHAX_USED(2) = 3
CHANDATA(2)
; affectation des axes machine - canal 2:
$MC_AXCONF_MACHAX_USED(0) = 4
$MC_AXCONF_MACHAX_USED(1) = 5
CHANDATA(1)
; paramètres machine spécifiques à un axe
; fenêtre d'arrêt précis grossier :
$MA_STOP_LIMIT_COARSE[AX1]=0.2 ; axe 1
$MA_STOP_LIMIT_COARSE[AX2]=0.2 ; axe 2
; fenêtre d'arrêt précis fin :
$MA_STOP_LIMIT_COARSE[AX1]=0.01 ; axe 1
$MA_STOP_LIMIT_COARSE[AX1]=0.01 ; axe 2
```



Dans le programme pièce, l'instruction CHANDATA ne peut être utilisée que pour le canal dans lequel le programme est exécuté ; c'est-à-dire que l'instruction peut être utilisée pour protéger le programme CN contre l'exécution dans un canal non prévu à cet effet.

En cas d'erreur, l'exécution du programme est abandonnée.

840 D
NCU 571840 D
NCU 572

FM-NC

810D
NCU 573

3.4 Définition des données utilisateur



Fonction



La définition des données utilisateur (GUD) se fait au moment de la mise en service.

Les paramètres machine nécessaires doivent avoir été réglés en conséquence.

La mémoire utilisateur doit être configurée ; la configuration de la mémoire doit figurer dans le fichier %_N_INITIAL_INI qui est chargé après le fichier de définition. Tous les paramètres machine à régler intègrent dans leur nom la composante GUD.

Noms réservés pour blocs de données

Les blocs de données suivants peuvent être rangés dans le répertoire /_N_DEF_DIR :

_N_SMAC_DEF	contient les définitions de macros (Siemens)
_N_MMAC_DEF	contient les définitions de macros (constructeur machine)
_N_UMAC_DEF	contient les définitions de macros (utilisateur)
_N_SGUD_DEF	contient des définitions pour les données globales (Siemens)
_N_MGUD_DEF	contient les définitions pour les données globales (constructeur machine)
_N_UGUD_DEF	contient les définitions pour les données globales (utilisateur)
_N_GUD4_DEF	contient les définitions pour les cycles de rectification (Siemens / constructeur machine)
_N_GUD5_DEF	contient les définitions pour les cycles de mesure (Siemens / constructeur machine)
_N_GUD6_DEF	contient les définitions pour les cycles de fraisage (Siemens / constructeur machine)
_N_GUD7_DEF	contient les définitions pour les cycles de tournage (Siemens / constructeur machine)
_N_GUD8_DEF	contient des définitions pour le constructeur machine
_N_GUD9_DEF	contient des définitions pour les données globales (Siemens)

3.4 Définition des données utilisateur



840 D
NCU 571



840 D
NCU 572



FM-NC



810D

NCU 573

Définition des données utilisateur (GUD)

1. Sauvegarder le bloc de données `_N_INITIAL_INI`.
2. Elaborer le fichier de définition des données utilisateur sur un PC externe.
 - Il existe des noms de fichiers prédéfinis (voir page précédente) :
 - `_N_SGUD_DEF`
 - `_N_MGUD_DEF`
 - `_N_UGUD_DEF`
 - `_N_GUD4_DEF ... _N_GUD9_DEF`
 Les fichiers avec ces noms peuvent contenir des définitions de variables GUD.
 - En outre, un attribut est nécessaire pour identifier la variable comme variable GUD et fixer le domaine dans lequel la définition doit être valable :
 - `NCK`, `CHAN`.
 - Un arrêt implicite du prétraitement des blocs peut aussi être défini pour permettre ultérieurement la lecture et/ou l'écriture de la variable :
 - `SYNR`: Arrêt du prétraitement lors de la lecture
 - `SYNRW`: Arrêt du prétraitement lors de la lecture et de l'écriture
3. Charger le fichier de définition dans la mémoire de programmes de la commande.

Par défaut, la commande crée toujours un répertoire `_N_DEF_DIR`.

Ce nom est intégré en guise de chemin dans l'entête des fichiers de définition GUD et exporté lors de la lecture des fichiers par intermédiaire de l'interface V.24.

Exemple d'un fichier de définition pour données globales (Siemens) :

```

%_N_SGUD_DEF
; $PATH=/_N_DEF_DIR
DEF NCK REAL RTP ;plan de retrait
DEF CHAN INT SDIS ;distance de sécurité
M30
  
```

840 D
NCU 571840 D
NCU 572

FM-NC

810D
NCU 573

4. Activer les fichiers de définition

Les fichiers de définition chargés ne sont actifs qu'après la lecture d'un fichier `_N_INITIAL_INI` et un POWER ON.



Avant le chargement de `_N_INITIAL_INI`, sauvegardez tous les programmes, frames et paramètres machine, car la mémoire statique fera l'objet d'un formatage.

5. Sauvegarde des fichiers

Lors de l'archivage du fichier `_N_COMPLETE_GUD` à partir de la mémoire de travail, seuls les contenus en données sont sauvegardés. Les fichiers de définition des variables utilisateur globales créés doivent être archivés séparément.

Les variables affectées aux données utilisateur globales sont également mémorisées dans `_N_INITIAL_INI` ; les noms des variables doivent être identiques à ceux utilisés dans les fichiers de définition.

Exemple d'un fichier de définition pour données globales (constructeur machine) :

```
%_N_MGUD_DEF
```

```
;$PATH=/_N_DEF_DIR
```

```
;définition des données globales du constructeur de machine
```

```
DEF NCK SYNRW INT NOMBRPIECE ; paramètre spécifique dans la commande
; utilisable par tous les canaux
```

```
DEF CHAN INT TABLOUTIL[100] ; arrêt implicite du prétraitement des blocs
pour lecture / écriture
; table d'outils pour reproduction spécifique à
un canal
; des N° d'outils sur les emplacements
d'outils
```

```
M30 ; création d'une table séparée pour chaque
canal
```

3.5 Définition des niveaux de protection des données utilisateur

840 D
NCU 571840 D
NCU 572
NCU 573

FM-NC



810D

3.5 Définition des niveaux de protection des données utilisateur (GUD)



Signification

APR n	Protection d'accès en lecture
APW n	Protection d'accès en écriture
n	Echelon de protection n, de 0/10 (niveau le plus élevé) à 7/17 (niveau le plus bas)

Pour plus d'informations concernant les niveaux de protection, voir le "Manuel d'utilisation".



Remarque :

L'ordre d'introduction des instructions est le suivant :

APR.. APW..

sinon vous risquez des erreurs de syntaxe.

Pour protéger un fichier complet, les instructions doivent figurer dans la première ligne du fichier !



Fonction

En définissant des critères d'accès, on peut protéger les blocs de données GUD contre des manipulations. Par ce moyen, les cycles mis en place par le constructeur par exemple peuvent être protégés contre toute tentative de modifications.

La protection d'accès s'applique à toutes les variables qui sont définies dans le bloc de données GUD.

Lors d'une tentative d'accès non autorisée, la commande affiche l'alarme correspondante.

3.5 Définition des niveaux de protection des données utilisateur



840 D
NCU 571



840 D
NCU 572



FM-NC



810D
NCU 573



Procédure

La protection d'accès est à programmer dans le bloc de données lui-même, avant la première définition d'une variable, en précisant le niveau de protection. Les deux mots-clés doivent se trouver dans un même bloc ne contenant pas d'autre indication.

Protection d'accès "en lecture" APR signifie : aucun affichage à l'écran si pas de droit d'accès.

Protection d'accès "en écriture" APW signifie : toute modification impossible si pas de droit d'accès.

APW 0-7, APR 0-7 :

Les variables ne sont pas accessibles en écriture ou en lecture via le programme CN et en mode MDA.

APW 10-17, APR 10-17:

Les variables sont accessibles en écriture ou en lecture via le programme CN et en mode MDA.

3.5 Définition des niveaux de protection des données utilisateur



840 D
NCU 571



840 D
NCU 572
NCU 573



FM-NC



810D

Niveaux de protection

0/10 = SIEMENS

1/11 = OEM_HIGH

2/12 = OEM_LOW

3/13 = Utilisateur final

4/14 à 7/17 = Positions 3 à 0 du commutateur à clé.

Exemple d'un fichier de définition comportant une protection d'accès en Lecture/Ecriture (constructeur machine)

```

%_N_GUD6_DEF
; $PATH=/_N_DEF_DIR
APR 5 APW 2                                ;lecture/affichage avec protection en
                                             position 2 du commutateur à clé
                                             ;écriture avec niveau de protection
                                             OEM_LOW
DEF CHAN REAL_CORRVAL
...
M30                                         ;

```

840 D
NCU 571840 D
NCU 572

FM-NC



810D

NCU 573

3.6 Définition des macros

Les macros servent à regrouper des instructions. Ils permettent à l'utilisateur de structurer clairement ses programmes.

Les macros sont définis de façon globale pour tous les canaux et tous les programmes. Ils peuvent également être utilisés dans un programme spécifique. Il existe des noms de fichier prédéfinis pour la définition des macros (voir pages précédentes).

Exemple d'un fichier de macros défini de façon globale

Après chargement du fichier de macros dans la CN, les macros sont activables après le prochain POWER ON et sont alors utilisables dans le programme pièce.

```

%_N_UMAC_DEF
; $PATH=/_N_DEF_DIR
; macros spécifiques à l'utilisateur
DEFINE PI AS 3.14
DEFINE TC AS M3 S1000
DEFINE M13 AS M3 M7 ; broche sens horaire, arrosage en
marche
DEFINE M14 AS M4 M7 ; broche sens antihoraire, arrosage en
marche
DEFINE M15 AS M5 M9 ; arrêt broche, arrêt arrosage
DEFINE M6 AS L6 ; appel du programme de changement
d'outil
DEFINE G80 AS MCALL ; annulation du cycle de perçage
M30 ;

```

Vous trouverez plus d'informations concernant la macroprogrammation dans le chapitre 2 "Macroprogrammation".

Zones de protection

- 4.1 Définition des zones de protection CPROTDEF, NPROTDEF4-144
- 4.2 Activation / Désactivation des zones de protection, CPROT, NPROT4-148

4.1 Définition des zones de protection CPROTDEF, NPROTDEF

840 D
NCU 571840 D
NCU 572
NCU 573

FM-NC



810D

4.1 Définition des zones de protection CPROTDEF, NPROTDEF



Programmation

```
DEF INT NOT_USED
CPROTDEF(n,t,applim,applus,appminus)
NPROTDEF(n,t,applim,applus,appminus)
EXECUTE (NOT_USED)
```



Signification des instructions

DEF INT NOT_USED	Variable locale, définition du type INT (entier) (voir chap. 10)
CPROTDEF	Définition des zones de protection spécifiques au canal (pour NCU 572/573 uniquement)
NPROTDEF	Définition des zones de protection spécifiques à la machine
EXECUTE	Fin de la définition



Signification des paramètres

n	Numéro de la zone de protection définie
t	TRUE = zone de protection orientée outil FALSE = zone de protection orientée pièce
applim	Type de limitation dans la 3e dimension 0 = sans limitation 1 = limitation dans le sens positif 2 = limitation dans le sens négatif 3 = limitation dans le sens positif et négatif
applus	Valeur de la limitation dans le sens positif de la 3e dimension
appminus	Valeur de la limitation dans le sens négatif de la 3e dimension
NOT_USED	La variable d'erreur est inopérante en présence de zones de protection avec EXECUTE

4.1 Définition des zones de protection CPROTDEF, NPROTDEF

840 D
NCU 571840 D
NCU 572
NCU 573

FM-NC



810D



Fonction

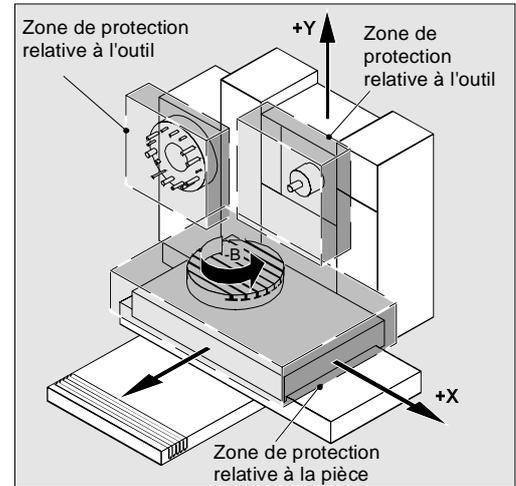
Grâce aux zones de protection, il est possible de protéger différents éléments de la machine, l'outillage et la pièce contre les déplacements erronés.

Zones de protection orientées outil :

Pour des éléments faisant partie de l'outil (par ex. : outil, support porte-outil).

Zones de protection orientées pièce :

Pour des éléments faisant partie de la pièce (par ex. : parties de la pièce, table porte-pièce, pinces, mandrins, contre-poupées).



Procédure

Définition des zones de protection

Font partie de la définition des zones de protection :

- CPROTDEF pour les zones de protection spécifiques au canal
- NPROTDEF pour les zones de protection spécifiques à la machine
- la description du contour de la zone de protection
- la clôture de la définition avec EXECUTE



Lors de l'activation de la zone de protection dans le programme pièce CN, on peut déplacer, en valeur relative, le point de référence de la zone de protection.

4.1 Définition des zones de protection CPROTDEF, NPROTDEF



840 D
NCU 571



840 D
NCU 572



FM-NC



810D

NCU 573

Point de référence de la description du contour

Les zones de protection orientées pièce sont définies dans le système de coordonnées de base. Les zones de protection orientées outil sont définies par rapport au point de référence F du porte-outil.

Description du contour des zones de protection

On peut décrire le contour d'une zone de protection avec 11 déplacements au maximum dans le plan sélectionné. Le premier déplacement est constitué par l'accostage du contour. Les déplacements qui figurent entre CPROTDEF ou NPROTDEF et EXECUTE ne sont pas exécutés, mais définissent la zone de protection.

Plan de travail

Le plan de travail désiré est à sélectionner avant CPROTDEF ou NPROTDEF avec G17, G18, G19 et ne doit pas être modifié avant EXECUTE. La programmation d'une cote (3^{me} dimension) entre CPROTDEF ou NPROTDEF et EXECUTE n'est pas autorisée.

Éléments de contour

Sont autorisés :

- G0 et G1 pour les parties rectilignes du contour
- G2 pour les arcs de cercle dans le sens horaire (seulement pour les zones de protection orientées outil)
- G3 pour les arcs de cercle dans le sens antihoraire

Dans le cas de la SINUMERIK FM-NC, on dispose de 4 éléments de contour maximum pour définir respectivement une zone de protection (max. 4 zones de protection).

Dans le cas de la 810D, on dispose de 4 éléments de contour maximum pour définir respectivement une zone de protection (max. 4 zones de protection spécifique à un canal et 4 zones de protection spécifique à NCK).

4.1 Définition des zones de protection CPROTDEF, NPROTDEF

840 D
NCU 571840 D
NCU 572
NCU 573

FM-NC



810D



Si la zone de protection doit être définie par un cercle complet, il faut subdiviser ce dernier en deux cercles partiels. Les séquences G2, G3 et G3, G2 ne sont pas autorisées. Il faut, dans ce cas, insérer un bloc court avec G1.

Le dernier point et le premier point de la description du contour doivent coïncider.

Les **zones de protection extérieures** (seulement pour les zones de protection orientées outil) doivent être décrites **dans le sens horaire**.

Dans le cas de zones de protection **symétriques en rotation** (par ex. sur un mandrin), vous devez décrire **l'ensemble du contour** (vous ne devez pas vous limiter à la seule partie du contour allant jusqu'au centre de rotation !).

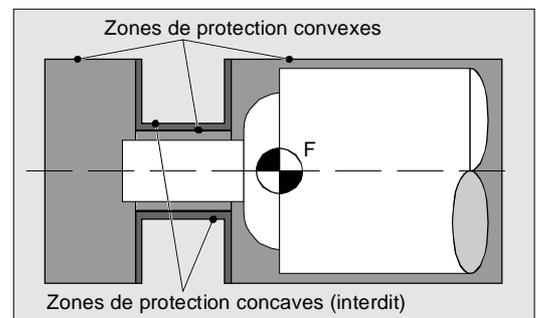
Les zones de protection **orientées outil** doivent toujours être **convexes**. S'il faut une zone de protection concave, il faut la fractionner en plusieurs zones de protection convexes.



Aucune des fonctions suivantes ne doit être active pendant la définition des zones de protection :

- correction du rayon de la fraise ou de la plaquette,
- transformation,
- frame.

D'autre part, les fonctions suivantes ne doivent pas être programmées : accostage du point de référence (G74), accostage d'un point fixe (G75), arrêt du prétraitement des blocs et fin de programme.



4.2 Activation / Désactivation des zones de protection



840 D
NCU 571



840 D
NCU 572
NCU 573



FM-NC



810D

4.2 Activation / Désactivation des zones de protection CPROT, NPROT



Programmation

CPROT (*n, state, xMov, yMov, zMov*)

NPROT (*n, state, xMov, yMov, zMov*)



Signification des instructions et paramètres

CPROT	Appel de la zone de protection spécifique au canal (pour NCU 572/573 uniquement)
NPROT	Appel de la zone de protection spécifique à la machine
<i>n</i>	Numéro de la zone de protection
<i>state</i>	Indication de l'état 0 = désactiver la zone de protection 1 = préactiver la zone de protection 2 = activer la zone de protection
<i>xMov, yMov, zMov</i>	Décaler dans les axes géométriques la zone de protection qui a été définie



Fonction

Activer les zones de protection définies précédemment pour assurer une surveillance anticollision ou désactiver les zones de protection actives.

Le nombre maximal de zones de protection qui peuvent être activées simultanément dans un même canal, est fixé par un paramètre machine.

Si aucune zone de protection orientée outil n'est activée, la trajectoire d'outil fera l'objet d'une surveillance rapportée aux zones de protection orientées pièce.



Si aucune zone de protection orientée pièce n'est activée, il n'y aura aucune surveillance de la trajectoire de l'outil.

4.2 Activation / Désactivation des zones de protection



840 D
NCU 571



840 D
NCU 572
NCU 573



FM-NC



810D



Procédure

Etat d'activation

Une zone de protection est activée en général dans un programme pièce avec l'état = 2.

L'état est toujours spécifique au canal, même pour des zones de protection orientées machine.

Si la possibilité d'activation d'une zone de protection est prévue dans le programme AP, la préactivation nécessaire à cet effet se fait avec l'état = 1.

La désactivation des zones de protection s'effectue avec l'état = 0. Aucun décalage n'est nécessaire à cet effet.

Décalage des zones de protection

lors de la (pré)activation

Le décalage peut être réalisé dans 1, 2 ou 3 dimensions.

La notification du décalage se rapporte :

- à l'origine machine dans le cas de zones de protection orientées pièce,
- au point de référence F du porte-outil dans le cas de zones de protection orientées outil.



Remarques complémentaires

Les zones de protection peuvent être activées dès le démarrage, lors de l'accostage du point de référence. Dans ce but, il convient de régler la variable système `$SN_PA_ACTIV_IMMED [n]` ou `$SN_PA_ACTIV_IMMED[n] = TRUE`. Dans ce cas, les zones de protection sont activées avec l'état = 2 et ne sont pas décalées.

4.2 Activation / Désactivation des zones de protection



840 D
NCU 571



840 D
NCU 572



FM-NC



810D

NCU 573

Activation multiple des zones de protection

Une zone de protection peut être activée dans plusieurs canaux à la fois (par ex. : un fourreau de contre-poupée quand deux chariots se font face).

La surveillance des zones de protection se fait uniquement si la prise de référence a été effectuée dans tous les axes géométriques. Les règles suivantes doivent être respectées :

- Une zone de protection ne peut pas être activée simultanément avec différents décalages dans un même canal.
- Les zones de protection orientées machine doivent présenter la même orientation dans les deux canaux.



Exemple de programmation

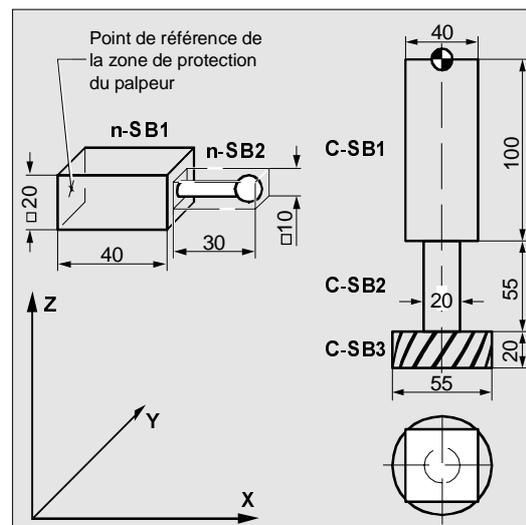
Sur une fraiseuse, il convient d'éviter une collision entre la fraise et le palpeur de mesure. La position du palpeur doit être indiquée par un décalage lors de l'activation.

A cet effet, on définit les zones de protection suivantes :

- Une zone de protection spécifique à la machine et une zone de protection orientée pièce pour le support du palpeur (n-SB1) et pour le palpeur lui-même (n-SB2).
- Une zone de protection spécifique au canal et une zone de protection orientée outil pour le support porte-fraise (c-SB1), la queue de la fraise (c-SB2) et la fraise elle-même (c-SB3).

Toutes les zones de protection sont orientées en Z.

Le point de référence du palpeur de mesure doit se trouver en $X = -120$, $Y = 60$ et $Z = 80$ lors de l'activation.



4.2 Activation / Désactivation des zones de protection

840 D
NCU 571840 D
NCU 572
NCU 573

FM-NC



810D

DEF INT ZONEPROT

Définition d'une variable auxiliaire

Définition des zones de protection

G17

Réglage de l'orientation

NPROTDEF(1, FALSE, 3, 10, -10)

Zone de protection n-SB1

G01 X0 Y-10

X40

Y10

X0

Y-10

EXECUTE (ZONEPROT)

NPROTDEF(2, FALSE, 3, 5, -5)

Zone de protection n-SB2

G01 X40 Y-5

X70

Y5

X40

Y-5

EXECUTE (ZONEPROT)

CPROTDEF(1, TRUE, 3, 0, -100)

Zone de protection c-SB1

G01 X-20 Y-20

X20

Y20

X-20

Y-20

EXECUTE (ZONEPROT)

CPROTDEF(2, TRUE, 3, -100, -150)

Zone de protection c-SB2

G01 X0 Y-10

G03 X0 Y10 J10

X0 Y-10 J-10

EXECUTE (ZONEPROT)

CPROTDEF(3, TRUE, 3, -150, -170)

Zone de protection c-SB3

G01 X0 Y-27,5

G03 X0 Y27,5 J27,5

X0 Y27,5 J-27,5

EXECUTE (ZONEPROT)

Activation des zones de protection :

NPROT(1, 2, -120, 60, 80)

Activer zone de protection n-SB1 av. décalage

NPROT(2, 2, -120, 60, 80)

Activer zone de prot. n-SB2 av. décalage

CPROT(1, 2, 0, 0, 0)

Activer zone de prot. n-SB1 av. décalage

CPROT(2, 2, 0, 0, 0)

Activer zone de prot. n-SB2 av. décalage

CPROT(3, 2, 0, 0, 0)

Activer zone de protection n-SB3 av. décalage

4.2 Activation / Désactivation des zones de protection



840 D
NCU 571



840 D
NCU 572



FM-NC



810D

NCU 573

Notes

Instructions de déplacement spéciales

5.1 Accostage de positions codées, CAC, CIC, CDC, CACP, CACN.....	5-154
5.2 Interpolation de type spline	5-156
5.3 Compactage.....	5-164
5.4 Interpolation polynomiale - POLY.....	5-167
5.5 Mesure avec palpeur à déclenchement, MEAS, MEAW	5-173
5.6 Mesure étendue MEASA, MEAWA, MEAC (à partir de SW 4, option)	5-176
5.7 Fonctions spéciales pour l'utilisateur OEM	5-186

5.1 Accostage de positions codées, CAC, CIC, CDC, CACP, CACN840 D
NCU 571840 D
NCU 572
NCU 573

FM-NC



810D

5.1 Accostage de positions codées, CAC, CIC, CDC, CACP, CACN**Signification des instructions**

CAC (n)	Accostage de la position codée par déplacement absolu
CIC (n)	Accostage de la position codée par déplacement incrémental de n positions en avant (+) ou en arrière (-)
CDC (n)	Accostage de la position codée par le plus court chemin (pour axes rotatifs uniquement)
CACP (n)	Accostage de la position codée par déplacement absolu dans le sens positif (pour axes rotatifs uniquement)
CACN (n)	Accostage de la position codée par déplacement absolu dans le sens négatif (pour axes rotatifs uniquement)
(n)	Numéros de position 1, 2, ... au maximum 60 positions par axe

**Fonction**

Fréquemment, c'est avec des axes de positionnement ou des broches (Axes indexés, par ex. tourelle revolver), que l'on accoste des positions fixes.

Les coordonnées de ces points fixes sont identifiées une fois pour toutes (dans des paramètres machine) par des numéros de positions.

De ce fait, lors de la programmation, les instructions de déplacement sont facilement programmables par le truchement des numéros.

5.1 Accostage de positions codées, CAC, CIC, CDC, CACP, CACN



840 D
NCU 571



840 D
NCU 572
NCU 573



FM-NC



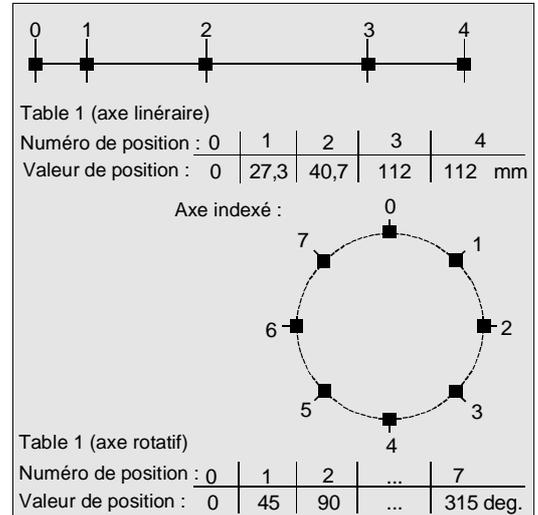
810D



Procédure

Par le biais des paramètres machine, on peut introduire respectivement pour 2 axes 60 positions au maximum (de 0 à 59) dans des tables de positions.

Exemple d'une table de positions – voir la figure ci-contre.



Remarques complémentaires

Quand un axe est immobilisé entre deux positions, aucun déplacement ne se fera si la position à accoster a été déclarée de façon incrémentale avec CIC(...).

Il est recommandé de toujours programmer la première instruction de déplacement par une déclaration de position absolue.



Exemple de programmation

N10 FA[B]= 300	Avance de l'axe de positionnement B
N20 POS[B]= CAC (10)	Accostage de la position codée 10 (déplacement absolu)
N30 POS[B]= CIC (-4)	Déplacement de 4 positions en arrière, à partir de la position courante

5.2 Interpolation de type spline



840 D
NCU 571



840 D
NCU 572



FM-NC

NCU 573

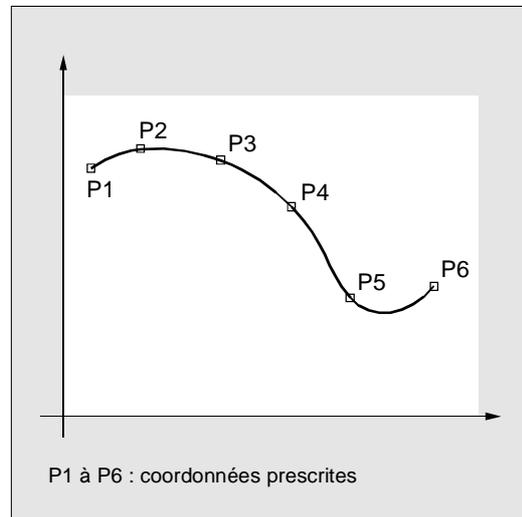
5.2 Interpolation de type spline



Introduction

L'interpolation de type spline permet de relier des suites de points par des courbes lissées. Les splines peuvent être utilisés, par exemple, pour réaliser des courbes en reliant des points numérisés.

Il existe divers types de spline avec des caractéristiques différentes qui mènent à des résultats différents. En plus du choix du type de spline, l'utilisateur peut également intervenir sur une série de paramètres. Quelques essais sont souvent nécessaires pour obtenir le résultat souhaité.



Programmation

```
ASPLINE X Y Z A B C
ou
BSPLINE X Y Z A B C
ou
CSPLINE X Y Z A B C
```



Fonction

On programme une interpolation de type spline, quand il y a lieu de relier une suite de points par une courbe.

Trois types de spline sont possibles :

- Spline A (spline Akima)
- Spline B (spline de base rationnel, non uniforme, NURBS)
- Spline C (spline cubique)

840 D
NCU 571840 D
NCU 572
NCU 573

FM-NC



Remarques complémentaires

Les splines A, B et C sont à effet modal et font partie du groupe des instructions de déplacement. La correction de rayon d'outil est utilisable. La surveillance anticollision s'effectue par projection dans le plan.

Les axes à interpoler dans le groupe spline sont sélectionnés avec l'instruction SPLINEPATH (pour plus d'informations, voyez les pages suivantes).



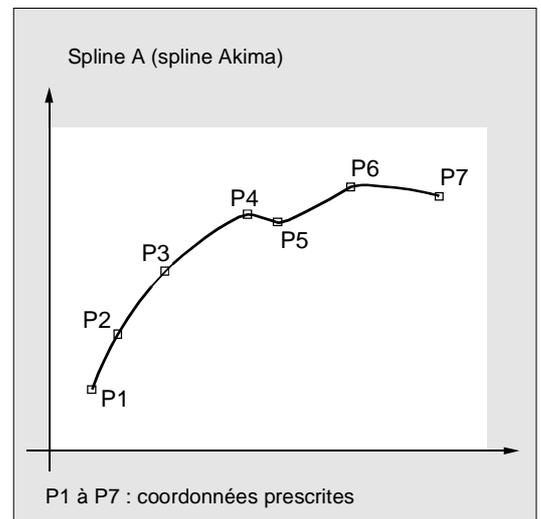
Procédure

SPLINE A

Le spline A (spline Akima) passe exactement par les points intermédiaires. Il génère peu d'oscillations indésirables, par contre sa courbure n'est pas constante aux points intermédiaires.

Le spline Akima est local, ce qui signifie que la modification d'un point intermédiaire n'influence pas plus de 6 points intermédiaires voisins.

Par conséquent, il convient essentiellement à l'interpolation de points numérisés. On peut programmer des conditions marginales pour le spline (pour plus d'informations, voyez les pages suivantes). Un polynôme du troisième degré est appliqué pour l'interpolation.



5.2 Interpolation de type spline



840 D
NCU 571



840 D
NCU 572



FM-NC
NCU 573

SPLINE B

Dans le cas du spline B, les positions programmées ne sont pas des points intermédiaires, mais uniquement des points de contrôle du spline, c'est-à-dire que la courbe ne passe pas directement par ces points, mais qu'elle "tend" vers eux.

La liaison des points par des droites forme le polygone de contrôle du spline. Le spline B est idéal pour décrire des trajectoires d'outils sur des surfaces de forme quelconque. Il est destiné en premier lieu à servir d'interface avec les systèmes de CAO. Un spline B du troisième degré ne génère pas d'oscillations, malgré des transitions à courbure constante.

Les conditions marginales programmables (pour plus d'information, voyez les pages suivantes) n'ont aucun effet sur le spline B. Au point de départ et au point final, le spline B est toujours tangent au polygone de contrôle.

Poids de point :

Un poids peut être indiqué pour chaque point intermédiaire.

Programmation :

PW = n

Plage de valeurs :

$0 \leq n \leq 3$; par incrément de 0.0001

Effet :

$n > 1$ la courbe tend plus fortement vers le point de contrôle

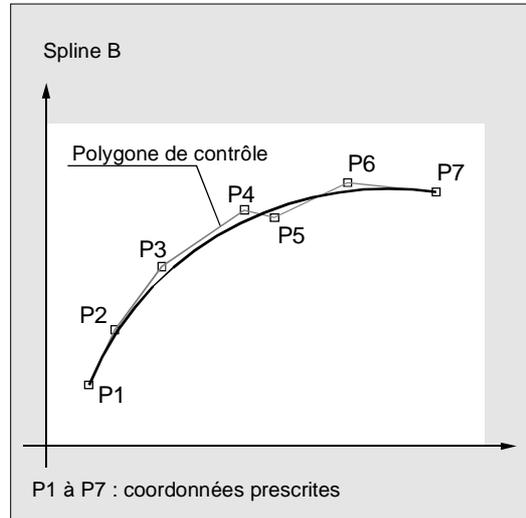
$n < 1$ la courbe tend moins fortement vers le point de contrôle

Degré de spline

Un polygone du troisième degré est appliqué de façon standard. On peut également utiliser un polygone du deuxième degré.

Programmation :

SD = 2



840 D
NCU 571840 D
NCU 572
NCU 573

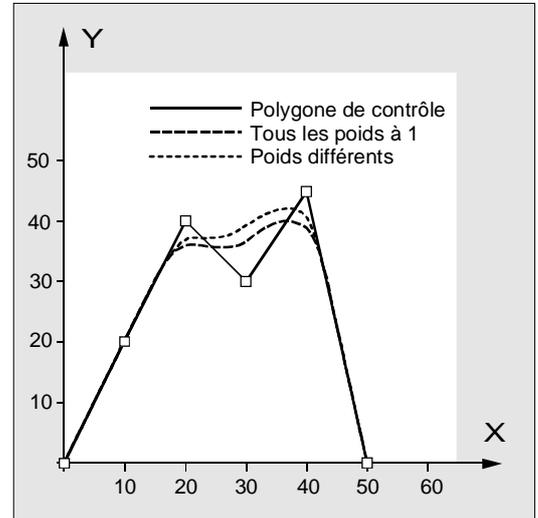
FM-NC

Ecart entre points de jonction :

Les écarts entre les points de jonction sont calculés par le système, mais la commande peut également traiter des écarts prescrits.

Programmation :

PL = Plage de valeurs comme pour positionnement non indexé

**Exemple d'une courbe de type spline B :****tous les poids 1**

```
N10 G1 X0 Y0 F300 G64
N20 BSPLINE
N30 X10 Y20
N40 X20 Y40
N50 X30 Y30
N60 X40 Y45
N70 X50 Y0
```

poids différents

```
N10 G1 X0 Y0 F300 G64
N20 BSPLINE PW=0.3
N30 X10 Y20 PW=2
N40 X20 Y40
N50 X30 Y30 PW=0.5
N60 X40 Y45
N70 X50 Y0
```

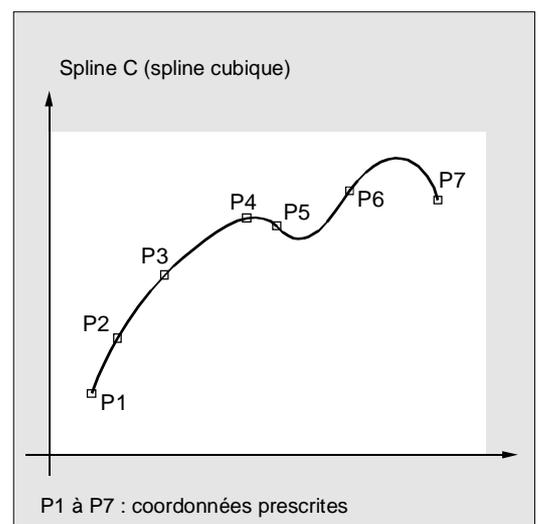
polygone de contrôle

```
N10 G1 X0 Y0 F300 G64
N20 ;sans objet
N30 X10 Y20
N40 X20 Y40
N50 X30 Y30
N60 X40 Y45
N70 X50 Y0
```

SPLINE C

A la différence du spline Akima, le spline cubique (Spline C) est à courbure constante. Toutefois, il a tendance à générer des oscillations inattendues. On peut l'utiliser lorsque les points se situent sur une courbe analytique connue. Le spline C utilise des polynômes du troisième degré.

Le spline n'est pas local, c'est-à-dire que la modification d'un point intermédiaire peut avoir une influence sur de nombreux blocs (avec une intensité décroissante).



5.2 Interpolation de type spline



840 D
NCU 571



840 D
NCU 572



FM-NC
NCU 573



Conditions marginales

Les conditions marginales s'appliquent uniquement au spline Akima et au spline cubique (spline A, spline C).

Le comportement de ces courbes spline aux transitions (début ou fin) peut être spécifier par le biais de deux groupes de trois instructions chacun.



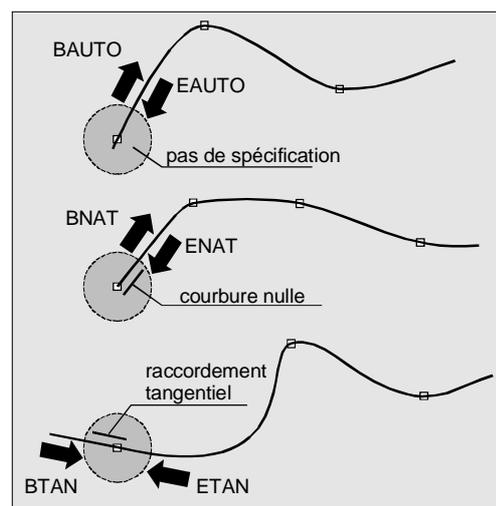
Signification des instructions

Début de la courbe spline :

BAUTO	pas de spécification ; le début résulte de la position du premier point
BNAT	courbure nulle
BTAN	transition tangentielle avec bloc précédent (fonction initialisée)

Fin de la courbe spline :

EAUTO	aucune spécification ; la fin résulte de la position du dernier point
ENAT	courbure nulle
ETAN	transition tangentielle avec bloc suivant (fonction initialisée)

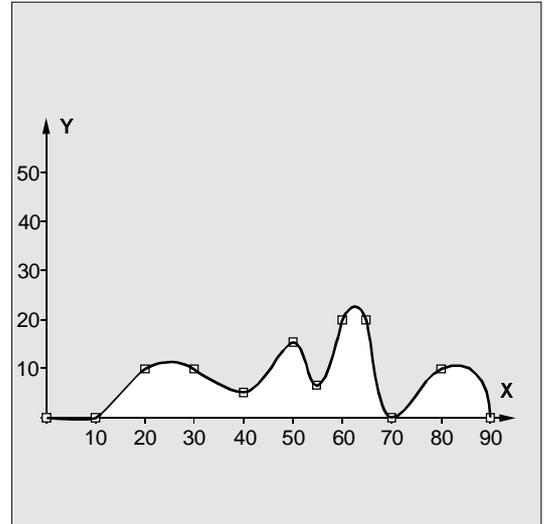


840 D
NCU 571840 D
NCU 572
NCU 573

FM-NC

**Exemple**

Spline C, courbure nulle au début et à la fin



N10 G1 X0 Y0 F300

N15 X10

N20 BNAT ENAT

spline C, courbure nulle au début
et à la fin

N30 CSPLINE X20 Y10

N40 X30

N50 X40 Y5

N60 X50 Y15

N70 X55 Y7

N80 X60 Y20

N90 X65 Y20

N100 X70 Y0

N110 X80 Y10

N120 X90 Y0

N130 M30

5.2 Interpolation de type spline



840 D
NCU 571



840 D
NCU 572



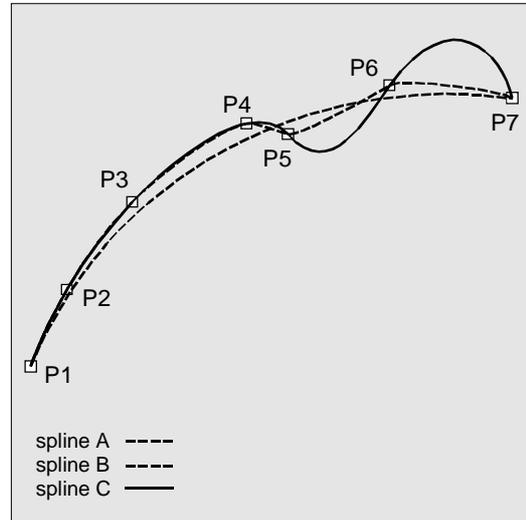
FM-NC
NCU 573



Quel est l'effet de chaque spline ?

Comparaison des trois types de spline avec des points intermédiaires identiques :

- spline A (spline Akima)
- spline B (spline de Bézier)
- spline C (spline cubique)



Groupe spline

L'interpolation de type spline admet jusqu'à 8 axes à interpolation. Avec l'instruction `SPLINEPATH`, vous sélectionnez les axes qui prendront part à l'interpolation. Cette sélection se fait dans un bloc séparé. Si l'instruction `SPLINEPATH` n'est pas programmée de façon explicite, ce sont les trois premiers axes du canal qui seront déplacés en tant que groupe de courbes de type spline.

840 D
NCU 571840 D
NCU 572
NCU 573

FM-NC



Programmation

SPLINEPATH(*n*, X, Y, Z, ...)



Signification

SPLINEPATH(*n*, X, Y, Z, ...)

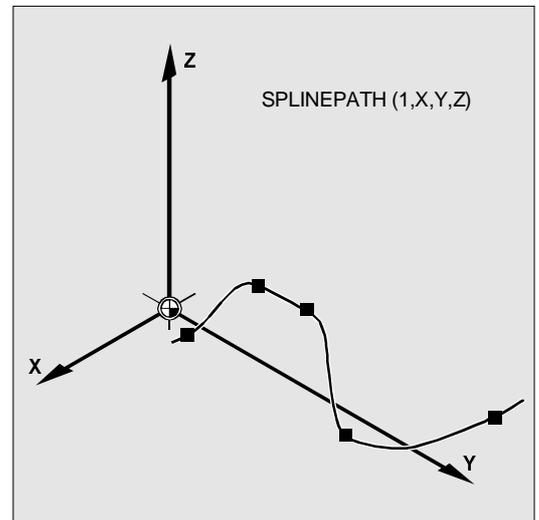
n = 1, valeur fixe

X, Y, Z, ... axes à interpolation



Exemple

Groupe spline avec trois axes à interpolation



N10 G1 X10 Y20 Z30 A40 B50 F350

N11 SPLINEPATH(1, X, Y, Z)

groupe spline

N13 CSPLINE BAUTO EAUTO X20 Y30 Z40 A50 B60

spline C

N14 X30 Y40 Z50 A60 B70

points intermédiaires

...

N100 G1 X... Y...

désactivation de l'interpolation de type spline

5.3 Compactage



840 D
NCU 572
NCU 573

5.3 Compactage



Les fonctions d'interpolation du type spline ont pour objet d'améliorer l'efficacité de l'échange de données avec le système de CAO et de proposer un traitement géométrique de même niveau que celui autorisé par la CAO.

Cependant, si les courbes de type spline générées par le système CAO sont approximées par des blocs linéaires, on obtient une quantité importante de données qui sollicite beaucoup les supports de mémoire, la communication et les performances des CNC des autres unités d'automatisation.

Ces sollicitations peuvent être réduites par l'utilisation de postprocesseurs qui convertissent directement le format CAO en un format polynomial ou spline B (pour plus d'informations, voyez les pages suivantes).

Le compactage est une autre possibilité de réduction des quantités de données en cas d'interpolation linéaire et de petits déplacements.

Le code modal COMPON, permet d'activer un "compactage de blocs CN" dans les parties critiques du programme. Dans le cas de l'interpolation linéaire, cette fonction regroupe une série de blocs d'interpolation linéaire (nombre limité) et réalise l'approximation par une courbe spline cubique, avec une tolérance à spécifier avec un paramètre machine. Au lieu de traiter de nombreux petits blocs, la CN traite un bloc de déplacement plus important.

Cette opération de compactage est effectuée exclusivement pour les blocs d'interpolation linéaire (G1). Elle est interrompue par toute autre instruction CN, par une sortie de fonction auxiliaire par exemple, mais ne l'est pas par le calcul de paramètres. Les blocs à compacter ne doivent comporter que numéro de bloc, G1, adresses d'axes, avance et commentaire. Cet ordre doit être respecté. Les variables sont interdites.



840 D
NCU 572
NCU 573



Programmation

COMPON Activation du compactage
COMPOF Désactivation du compactage



Il existe deux paramètres machine pour la fonction de compactage :

- Une longueur maximale jusqu'à laquelle les blocs sont considérés comme compactables. Les blocs plus longs ne sont pas compactés.
- La tolérance peut être spécifiée pour chaque axe. Cette valeur correspond à la différence maximale entre la courbe spline générée et les points finaux programmés. Plus ces valeurs sont élevées, plus il est possible de compacter de blocs.



Exemple

N10 COMPON	Activation du compactage
N11 G1 X0.37 Y2.9 F600	G1 doit figurer avant le point final et la valeur d'avance
N12 X16.87 Y-4.698	
N13 X16.865 Y-4.72	
N14 X16.91 Y-4.799	
...	
N1037 COMPOF	Désactivation du compactage
...	



Tous les blocs qui possèdent une syntaxe simple (p. ex. pas les adresses avec extension telles que C=100 ou A=ACNC) sont comprimés.

5.3 Compactage



840 D
NCU 572
NCU 573



Réglages pour les courbes de type spline

Tous les véritables modes d'interpolation de type spline ASPLINE, BSPLINE, CSPLINE (pas POLY) ont en commun le fait que la commande doit calculer les coefficients polynomiaux d'une section spline, à partir d'un nombre de points finaux programmés, nombre dépendant du type de spline.

Pour ce faire, la commande doit effectuer le prétraitement simultané d'une série de blocs qui doivent donc être rangés dans une mémoire temporaire. La capacité de cette mémoire tampon est de dix blocs en version standard.

Etant donné que les blocs CN ne contiennent pas tous des points finaux pour le calcul du spline, la capacité de la mémoire tampon doit être suffisante pour pouvoir accepter également une série d'autres blocs CN, en plus des blocs de déplacement nécessaires au calcul d'une section spline.

Le nombre de blocs tolérés sans point spline dépend du type de spline activé :

Spline A: sur dix blocs, quatre au moins doivent être des blocs spline. Les blocs de commentaire et les calculs de paramètres ne sont pas pris en compte.

Spline B: sur dix blocs, six au moins doivent être des blocs spline. Les blocs de commentaire et les calculs de paramètres ne sont pas pris en compte.

Spline C: la valeur tolérée est entrée dans le paramètre machine \$MC_CUBIC_SPLINE_BLOCKS. Version standard : sur dix blocs, huit au moins doivent être des blocs spline.



Si la valeur tolérée est dépassée, une alarme est émise, de même que si un axe participant à l'interpolation de spline est programmé comme axe de positionnement.



840 D
NCU 571



840 D
NCU 572
NCU 573

5.4 Interpolation polynomiale - POLY



La commande est en mesure d'opérer des déplacements suivant des courbes (trajectoires) pour lesquelles chaque axe à interpolation sélectionné suit une fonction (polynôme du troisième degré max.).

La forme générale de la fonction polynomiale est la suivante :

$$f(p) = a_0 + a_1p + a_2p^2 + a_3p^3$$

avec :

a_n : coefficients constants

p : paramètre

En affectant des valeurs concrètes aux coefficients, il est possible de générer les courbes les plus variées : droite, parabole, fonction puissance.

Quand on choisit les coefficients $a_2 = a_3 = 0$, on obtient par exemple une droite avec :

$$f(p) = a_0 + a_1p$$

Dans cette formule, les paramètres représentent :

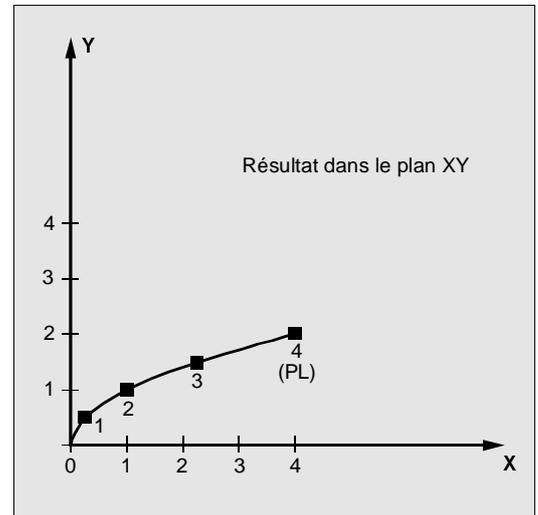
a_0 = position de l'axe à la fin du bloc précédent

a_1 = position de l'axe à la fin de la plage de définition (PL)

Définition

L'interpolation polynomiale (POLY) n'est en fait pas une interpolation de type spline. Elle est destinée avant tout à servir d'interface pour la programmation de courbes spline générées de manière externe. Les sections spline peuvent être programmées directement.

Ce mode d'interpolation évite à la CN de calculer les coefficients polynomiaux. Elle peut être utilisée de manière optimale lorsque les coefficients sont issus directement d'un système de CAO ou d'un postprocesseur.



5.4 Interpolation polynomiale - POLY



840 D
NCU 571



840 D
NCU 572
NCU 573



L'interpolation polynomiale figure, avec G0, G1, G2, G3, spline A, spline B et spline C, dans le premier groupe de fonctions G. Lorsqu'elle est active, il n'est pas nécessaire de programmer la syntaxe polynomiale : les axes qui sont programmés uniquement avec leur nom et leur point final sont déplacés linéairement vers leur point final. Si tous les axes sont programmés ainsi, la commande se comporte comme en mode G1.

L'interpolation polynomiale est désactivée par une autre instruction du groupe de fonctions G (par exemple G0, G1).



Coefficients polynomiaux

La valeur PO(PO[]=) indique tous les coefficients polynomiaux pour un axe. Selon le degré du polynôme, plusieurs valeurs séparées par des virgules sont indiquées. Différents degrés de polynômes sont possibles pour différents axes à l'intérieur d'un bloc.

840 D
NCU 571840 D
NCU 572
NCU 573

Programmation

POLY PO[X]=(x_{e1}, a_2, a_3) PO[Y]=(y_{e1}, b_2, b_3) PO[Z]=(z_{e1}, c_2, c_3) PL=n



Signification

POLY	activation de l'interpolation polynomiale avec un bloc contenant POLY.
PO[]=(... , ... , ...)	points finaux et coefficients polynomiaux
x_e, y_e, z_e	indication de la position finale pour l'axe concerné ; plage de valeurs identique au positionnement non indexé
a_2, a_3	les coefficients a_2 et a_3 sont écrits avec leur valeur ; plage de valeurs identique au positionnement non indexé. Le dernier coefficient peut être ignoré si sa valeur est nulle.
PL	longueur de l'intervalle de paramètres sur lequel sont définis les polynômes (plage de définition de la fonction f(p)). L'intervalle commence toujours à zéro. p peut prendre des valeurs de zéro à PL . Plage de valeurs théoriques pour PL: 0,0001 ... 99 999,9999. La valeur PL s'applique au bloc dans lequel elle figure. Si aucune valeur PL est programmée, PL=1 est appliqué.



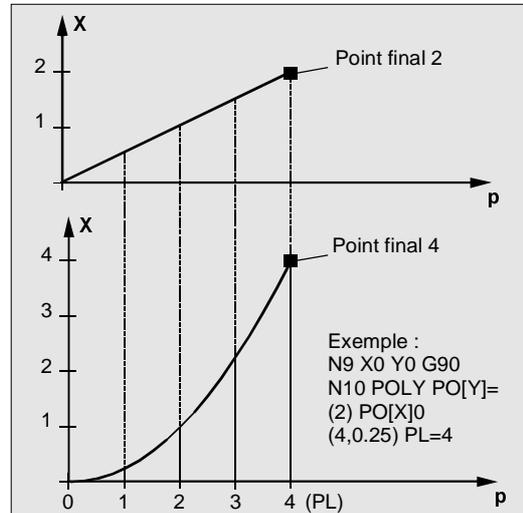
Exemple

N10 G1 X... Y... Z... F600	
N11 POLY PO[X]=(1,2.5,0.7) -> -> PO[Y]=(0.3,1,3.2) PL=1.5	activation de l'interpolation polynomiale
N12 PO[X]=(0,2.5,1.7) PO[Y]=(2.3,1.7) PL=3	
...	
N20 M8 H126 ...	
N25 X70 PO[Y]=(9.3,1,7.67) PL=5	indications mixtes pour les axes
N27 PO[X]=(10,2.5) PO[Y]=(2.3)	PL pas programmé ; PL=1 est valable
N30 G1 X... Y... Z.	désactivation de l'interpolation polynomiale
...	

5.4 Interpolation polynomiale - POLY

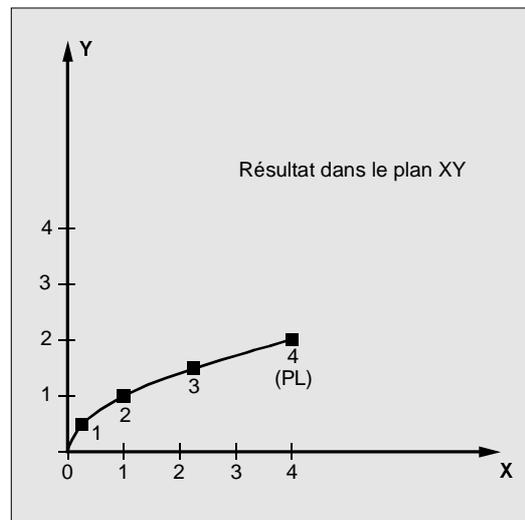
840 D
NCU 571840 D
NCU 572
NCU 573

Exemple d'une courbe dans le plan X/Y



```
N9 X0 Y0 G90 F100
```

```
N10 POLY PO[Y]=(2) PO[X]=(4,0.25) PL=4
```



840 D
NCU 571840 D
NCU 572
NCU 573

Particularité du polynôme dénominateur

Pour les axes géométriques, PO[] = (...) sans indication de nom d'axe permet aussi de programmer un polynôme dénominateur commun, c'est-à-dire que le déplacement des axes géométriques sera interpolé en tant que quotient de deux polynômes.

Ceci permet, par exemple, de représenter avec exactitude des coniques (cercle, ellipse, parabole, hyperbole).



Exemple

POLY G90 X10 Y0 F100

déplacement linéaire des axes géométriques à la position X10, Y0

PO[X]=(0, -10) PO[Y]=(10) PO[]=(2, 1)

déplacement des axes géométriques en quart de cercle sur position X0, Y10

Le coefficient constant (a_0) du polynôme dénominateur est toujours considéré comme étant égal à 1, le point final indiqué est indépendant de G90/G91.

L'exemple ci-dessus donne le résultat suivant :

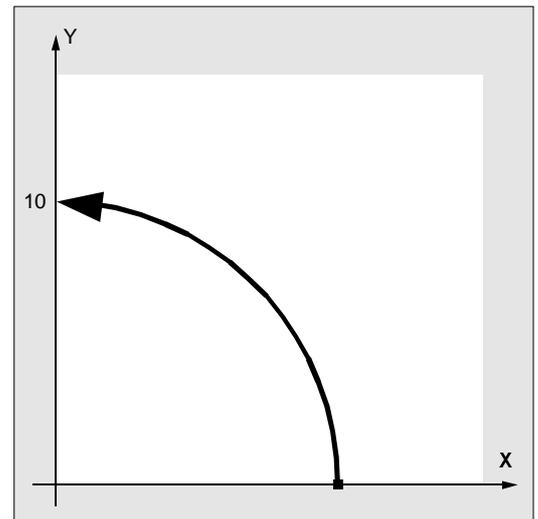
$X(p) = 10(1-p^2)/(1+p^2)$ et $Y(p) = 20p/(1+p^2)$
avec $0 \leq p \leq 1$

Compte tenu des points de départ, des points finaux, du coefficient a_2 et de $PL=1$, on obtient les valeurs intermédiaires suivantes :

Numérateur (X) = $10 + 0 \cdot p - 10p^2$

Numérateur (Y) = $0 + 20 \cdot p + 0 \cdot p^2$

Dénominateur = $1 + 2 \cdot p + 1 \cdot p^2$



5.4 Interpolation polynomiale - POLY



840 D
NCU 571



840 D
NCU 572
NCU 573

Si l'interpolation polynomiale est activée, la programmation d'un polynôme dénominateur possédant des racines à l'intérieur de l'intervalle $[0, PL]$ est refusée avec une alarme. Le polynôme dénominateur n'a aucun effet sur les déplacements des axes auxiliaires.



Remarques complémentaires

Dans le cas de l'interpolation polynomiale, on peut activer une correction de rayon d'outil avec G41, G42 et l'utiliser comme pour l'interpolation linéaire ou circulaire.

5.5 Mesure avec palpeur à déclenchement, MEAS, MEAW

840 D
NCU 571840 D
NCU 572
NCU 573810D
CCU 2

5.5 Mesure avec palpeur à déclenchement, MEAS, MEAW



Programmation

MEAS=±1	G... X... Y... Z...	(+1/+2 mesure avec effacement de la distance restant à parcourir et front montant)
MEAS=±2	G... X... Y... Z...	(-1/-2 mesure avec effacement de la distance restant à parcourir et front descendant)
MEAW=±1	G... X... Y... Z...	(+1/+2 mesure sans effacement de la distance restant à parcourir et front montant)
MEAW=±2	G... X... Y... Z...	(-1/-2 mesure sans effacement de la distance restant à parcourir et front descendant)



Signification des instructions

MEAS=±1	Mesure avec le palpeur 1 à l'entrée de mesure 1
MEAS=±2*	Mesure avec le palpeur 2 à l'entrée de mesure 2
MEAW=±1	Mesure avec le palpeur 1 à l'entrée de mesure 1
MEAW=±2*	Mesure avec le palpeur 2 à l'entrée de mesure 2

*2 entrées maxi selon la configuration



Procédure

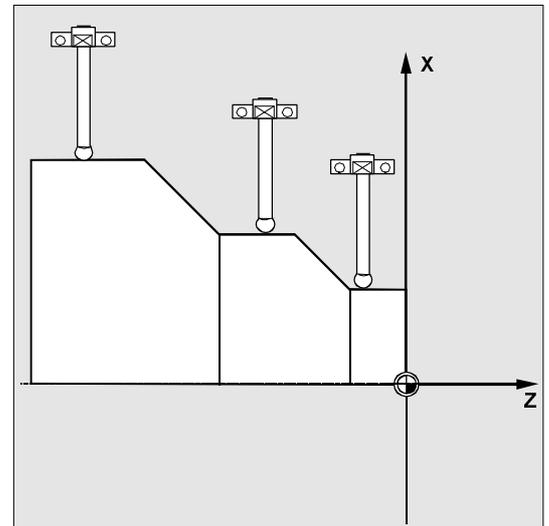
Pour tous les axes programmés dans le bloc CN, les positions sont saisies au front de commutation du palpeur et inscrites dans la cellule mémoire correspondant à l'axe. Deux palpeurs au maximum sont possibles.

Résultat de la mesure

Le résultat de la mesure est disponible pour ces axes sous les variables suivantes :

- dans le système de coordonnées machine sous \$AA_MM[axe]
- dans le système de coordonnées pièce sous \$AA_MW[axe]

A la lecture de ces variables, aucun arrêt du prétraitement des blocs n'est généré. Un arrêt du prétraitement des blocs est à programmer avec STOPRE à un endroit approprié du programme CN, faute de quoi il y aura lecture de valeurs erronées.



5.5 Mesure avec palpeur à déclenchement, MEAS, MEAW



840 D
NCU 571



840 D
NCU 572
NCU 573



810D
CCU 2

Etat du contrat de mesure

S'il est nécessaire, dans le programme, de savoir s'il y a eu déclenchement du palpeur ou non, on peut interroger la variable d'état $\$AC_MEA[n]$ (n=numéro du palpeur) :

- 0 Contrat de mesure non exécuté
- 1 Contrat de mesure exécuté correctement (le palpeur a déclenché)



Si le palpeur subit une déviation pendant le déroulement du programme, la variable est mise à 1. Au départ d'un bloc de mesure, la variable est automatiquement mise à 0 ou à 1, en fonction de l'état du palpeur.

Programmation de blocs de mesure, MEAS, MEAW

Avec l'instruction MEAS et en mode d'interpolation, des positions réelles sont accostées sur la pièce et des valeurs mesurées enregistrées. La distance restant à parcourir entre la position réelle et la position de consigne est effacée.

Pour les tâches de mesure spéciales, avec lesquelles la position programmée doit être accostée dans tous les cas, utilisez la fonction MEAW.

MEAS et MEAW sont programmées dans le bloc contenant les instructions de déplacement. Les avances et modes d'interpolation (G0, G1, ...) doivent être adaptés au problème de mesure concerné, de même que le nombre d'axes.

Exemple :

```
N10 MEAS=1 G1 F1000 X100 Y730 Z40
```

Bloc de mesure avec palpeur raccordé à la première entrée de mesure et interpolation linéaire. Un arrêt du prétraitement des blocs est généré automatiquement.

5.5 Mesure avec palpeur à déclenchement, MEAS, MEAW



840 D
NCU 571



840 D
NCU 572
NCU 573



810D
CCU 2

Enregistrement des valeurs de mesure

Les positions de tous les axes à interpolation et axes de positionnement déplacés dans le bloc (nombre maximal d'axes selon la configuration de la commande) sont enregistrées.

Avec MEAS, le déplacement est freiné de manière contrôlée dès que le palpeur a déclenché.

Remarque

Si un axe géométrique est programmé dans un bloc de mesure, les valeurs de mesure sont enregistrées pour tous les axes géométriques courants.

Quand un axe participant à une transformation est programmé dans un bloc de mesure, les valeurs de mesure sont enregistrées pour tous les axes participants à cette transformation.



Remarques complémentaires

Les fonctions MEAS et MEAW sont actives pendant un bloc.

5.6 Mesure étendue MEASA, MEAWA, MEAC (à partir de SW 4, option)



840 D
NCU 571



840 D
NCU 572
NCU 573



810D
CCU 2

5.6 Mesure étendue MEASA, MEAWA, MEAC (à partir de SW 4, option)



Programmation

MEASA[axe]=(mode, TE1,..., TE 4)	Mesure avec effacement de la distance restant à parcourir
MEAWA[axe]=(mode, TE 1,..., TE 4)	Mesure sans effacement de la distance restant à parcourir
MEAC[axe]=(mode, mémoire de mesure , TE 1,...TE4)	Mesure continue sans effacement de la distance restant à parcourir



Signification

Axe	Nom de l'axe de canal utilisé pour la mesure
Mode	Indication à deux chiffres du mode de fonctionnement représentant le mode de mesure (unité) et <ul style="list-style-type: none"> ...0 abandon du contrat de mesure ...1 mode 1: jusqu'à 4 événements déclencheurs activables simultanément ...2 mode 2: jusqu'à 4 événements déclencheurs activables successivement <p>le système de mesure (dizaine)</p> <ul style="list-style-type: none"> 0... ou aucune indication : système de mesure activé 1... système de mesure 1 2... système de mesure 2 3... les deux systèmes de mesure
TE 1...4	Événement déclencheur : <ul style="list-style-type: none"> 1 front montant, palpeur 1 -1 front descendant, palpeur 1 2 front montant, palpeur 2 -2 front descendant, palpeur 2
Mémoire de mesure	Numéro de la FIFO (mémoire à défilement)

5.6 Mesure étendue MEASA, MEAWA, MEAC (à partir de SW 4, option)



840 D
NCU 571



840 D
NCU 572
NCU 573



810D
CCU 2



Fonction

La mesure axiale est disponible à partir du logiciel SW 4.

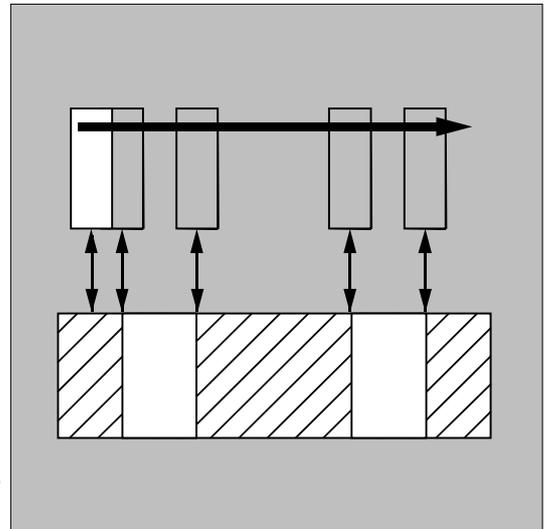
La fonction de mesure étendue permet d'effectuer des mesures axiales avec plusieurs palpeurs et plusieurs systèmes de mesure.

Avec MEASA et MEAWA, on peut saisir à chaque mesure jusqu'à quatre valeurs de mesure dans l'axe programmé et les ranger dans des variables système en fonction de l'événement déclencheur.

MEASA et MEAWA sont à effet modal.

Avec MEAC, on peut effectuer des mesures en continu.

Les résultats des mesures sont rangés dans des variables FIFO. Ici aussi, on peut saisir jusqu'à 4 valeurs par mesure.



Procédure

La programmation peut avoir lieu dans le programme pièce **ou** à partir d'une action synchrone (chapitre 10). On ne peut activer qu'un seul contrat de mesure par axe à un moment donné.



Remarques complémentaires

- L'avance est à adapter au problème de mesure posé.
- Dans le cas de **MEASA** et de **MEAWA**, des résultats corrects sont garantis uniquement avec des avances pour lesquelles il ne survient pas plus d'un même événement déclencheur et pas plus de quatre événements déclencheurs différents par période d'échantillonnage de l'asservissement de position.
- Dans le cas de la mesure continue avec **MEAC**, le rapport entre la période d'appel de l'interpolateur et la période d'échantillonnage de l'asservissement de position ne doit pas être supérieur à 8 : 1.

5.6 Mesure étendue MEASA, MEAWA, MEAC (à partir de SW 4, option)840 D
NCU 571840 D
NCU 572
NCU 573810D
CCU 2**Événements déclencheurs**

Un événement déclencheur se compose du numéro du palpeur de mesure et du critère de déclenchement du signal de mesure (front montant ou descendant).

Pour chaque mesure, le programme peut traiter respectivement jusqu'à 4 événements déclencheurs des palpeurs, autrement dit jusqu'à deux palpeurs avec deux fronts de mesure chacun.

L'ordre de traitement des événements déclencheurs et leur nombre maximum dépendent du mode choisi.

Un même événement déclencheur ne doit pas figurer plus d'une fois dans un contrat de mesure (pour mode 1 uniquement) !

Mode de fonctionnement

Le premier chiffre du mode sélectionne le système de mesure souhaité. S'il n'existe qu'un seul système de mesure, alors que le second a été programmé, c'est automatiquement le système de mesure existant qui sera pris en compte.

Le second chiffre, le **mode de mesure**, sert à adapter l'opération de mesure aux possibilités de la commande :

- **Mode 1:** Les événements déclencheurs sont traités dans l'ordre **chronologique** de leur apparition. Dans ce mode, quand on met en oeuvre des cartes 6 axes, on ne peut programmer qu'un seul événement déclencheur ou alors, si on en programme plusieurs, il y a commutation automatique dans le mode 2 (sans avertissement préalable).
- **Mode 2:** Les événements déclencheurs sont traités dans l'ordre de leur **programmation**.

Remarques complémentaires

Si vous travaillez avec 2 systèmes de mesure, seulement deux événements déclencheurs sont programmables.

5.6 Mesure étendue MEASA, MEAWA, MEAC (à partir de SW 4, option)

840 D
NCU 571840 D
NCU 572
NCU 573810D
CCU 2

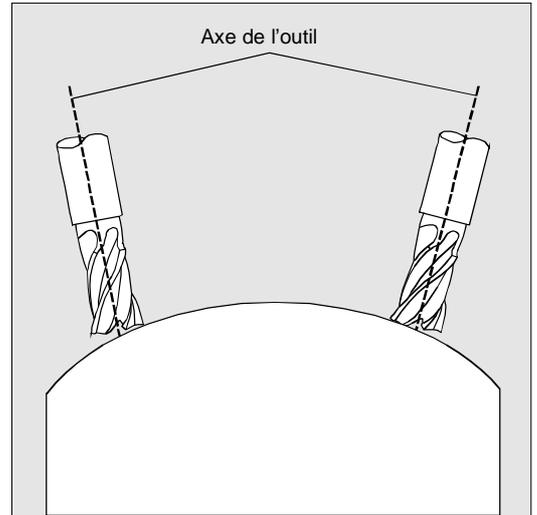
Mesure avec et sans effacement de la distance restant à parcourir

Quand vous programmez MEASA, l'effacement de la distance restant à parcourir est effectué seulement après la saisie de toutes les valeurs de mesure requises.

Pour des tâches de mesure spéciale où la position programmée doit être accostée impérativement, utilisez la fonction MEAW.

MEASA et MEAWA peuvent être programmées dans un même bloc.

Si vous programmez MEASA/MEAWA avec MEAS/MEAW dans un même bloc, un message d'erreur sera émis.



- MEASA n'est pas programmable dans des actions synchrones. Si on le souhaite, on peut programmer MEAWA plus effacement de la distance restant à parcourir en tant qu'action synchrone.
- Quand le contrat de mesure est lancé avec MEAWA à partir d'une action synchrone, les valeurs mesurées sont disponibles uniquement dans le système de coordonnées machine.

Résultats de mesure pour MEASA, MEAWA

Les résultats de mesure sont donnés pour ces axes sous les variables suivantes :

- Dans le système de coordonnées machine :

\$AA_MM1 [axe]	Valeur de mesure du système de mesure programmé pour l'événement déclencheur 1
...	...
\$AA_MM4 [axe]	Valeur de mesure du système de mesure programmé pour l'événement déclencheur 4
- Dans le système de coordonnées pièce :

\$AA_WM1 [axe]	Val. de mesure du syst. de mesure prog. pour l'événement déclencheur 1
...	...
\$AA_WM4 [axe]	Val. de mesure du syst. de mesure prog. pour l'événement déclencheur 4

5.6 Mesure étendue MEASA, MEAWA, MEAC (à partir de SW 4, option)840 D
NCU 571840 D
NCU 572
NCU 573810D
CCU 2**Remarques complémentaires**

A la lecture de ces variables, aucun arrêt du prétraitement des blocs n'est généré automatiquement. Un arrêt du prétraitement des blocs est à programmer avec STOPRE (chapitre 15.1) à un endroit approprié du programme CN. Dans le cas contraire, il y aura lecture de valeurs erronées.

Si la mesure axiale doit être lancée pour un axe géométrique, il convient de programmer de façon explicite le même contrat de mesure pour tous les autres axes géométriques.

Il en va de même pour les axes qui participent à une transformation.

Par ex. : N10 MEASA[X]=(1,-1) MEASA[Y]=(1,-1)
MEASA[Z]=(1,-1) G01 X100 F100;

**Contrat de mesure avec 2 systèmes de mesure**

Lorsqu'un contrat de mesure est exécuté avec deux systèmes de mesure, chacun des deux événements déclencheurs possibles est saisi par les deux systèmes de mesure de l'axe en question. Les variables réservées sont ainsi affectées :

\$AA_MM1[axe]	ou	\$AA_MW1[axe]	Valeur de mesure du système de mesure 1 pour l'événement déclencheur 1
\$AA_MM2[axe]	ou	\$AA_MW2[axe]	Valeur de mesure du système de mesure 2 pour l'événement déclencheur 1
\$AA_MM3[axe]	ou	\$AA_MW3[axe]	Valeur de mesure du système de mesure 1 pour l'événement déclencheur 2
\$AA_MM4[axe]	ou	\$AA_MW4[axe]	Valeur de mesure du système de mesure 2 pour l'événement déclencheur 2

L'état du palpeur de mesure peut être lu avec**\$A_PROBE[n]**

n=palpeur

1==le palpeur a dévié

0==le palpeur n'a pas dévié

5.6 Mesure étendue MEASA, MEAWA, MEAC (à partir de SW 4, option)



840 D
NCU 571



840 D
NCU 572
NCU 573



810D
CCU 2

Etat du contrat de mesure dans MEASA, MEAWA

S'il est nécessaire de connaître, dans le programme, l'état du contrat de mesure, on peut interroger cet état par le biais de **\$AC_MEA[n]**, avec n = numéro du palpeur de mesure.

Dès que tous les événements déclencheurs des palpeurs "n", programmés dans un bloc, ont été exécutés, cette variable fournit la valeur 1. Sinon, elle fournit la valeur 0.

Quand la mesure est lancée à partir d'actions synchrones, **\$AC_MEA** n'est plus actualisé. Dans ce cas, il convient d'interroger de nouveaux signaux d'état de l'automate DB(31-48) DBB62 Bit 3 ou la variable équivalente **\$AA_MEA**ACT["axe"].
Signification : **\$AA_MEA**ACT==1: mesure activée
\$AA_MEAACT==0: mesure non activée
Bibliographie : /FB/ M5, Mesures

Mesure continue MEAC

Avec MEAC, les valeurs de mesure sont fournies dans le système de coordonnées machine et rangées dans la mémoire FIFO [n] indiquée. Quand deux palpeurs ont été configurés pour la mesure, les valeurs de mesure du second palpeur seront rangées séparément dans la mémoire FIFO[n+1] configurée à cet effet (par le biais des PM).

La mémoire FIFO est une mémoire à défilement dans laquelle on enregistre les valeurs de mesure dans des variables **\$AC_FIFO** suivant le principe "premier entré, premier sorti".

Bibliographie : /PGA/ Chap. 10, actions synchrones

Remarques complémentaires

- Le contenu de la mémoire FIFO ne peut être lu qu'une seule fois. Si vous souhaitez pouvoir réutiliser les valeurs mesurées, mémorisez-les temporairement dans les données utilisateur.
- Dès que les valeurs mesurées destinées à la mémoire FIFO dépasse le nombre maximum prévu dans les paramètres machine, il sera mis fin automatiquement à la mesure.

5.6 Mesure étendue MEASA, MEAWA, MEAC (à partir de SW 4, option)



840 D
NCU 571



840 D
NCU 572
NCU 573



810D
CCU 2

- Pour réaliser une mesure sans fin, programmez la lecture cyclique des valeurs de mesure. La lecture doit être effectuée à la même fréquence que l'entrée de nouvelles valeurs de mesure.



Exemple de programmation

Mesure avec effacement de la distance restant à parcourir, en mode 1

(traitement chronologique)

a) avec 1 système de mesure

...	
N100 MEASA[X] = (1,1,-1) G01 X100 F100	Mesure en mode 1 avec système de mesure activé. Attente du signal de mesure avec front montant/descendant du palpeur 1 sur le trajet vers X = 100.
N110 STOPRE	Arrêt du prétraitement des blocs
N120 IF \$AC_MEA[1] == FALSE gotof FIN	Vérification du succès de la mesure.
N130 R10 = \$AA_MM1[X]	Mémorisation de la valeur de mesure correspondant au premier événement déclencheur programmé (front montant).
N140 R11 = \$AA_MM2[X]	Mémorisation de la valeur de mesure correspondant au second événement déclencheur programmé (front descendant).
N150 FIN :	



Exemple de programmation

b) avec 2 systèmes de mesure

...	
N200 MEASA[X] = (31,1-1) G01 X100 F100	Mesure en mode 1 avec les deux systèmes de mesure. Attente du signal de mesure avec front montant/descendant du palpeur 1 sur le trajet vers X = 100.
N210 STOPRE	Arrêt du prétraitement des blocs
N220 IF \$AC_MEA[1] == FALSE gotof FIN	Vérification du succès de la mesure.
N230 R10 = \$AA_MM1[X]	Mémorisation de la valeur mesurée par le système de mesure 1 avec front montant
N240 R11 = \$AA_MM2[X]	Mémorisation de la valeur mesurée par le système de mesure 2 avec front montant

5.6 Mesure étendue MEASA, MEAWA, MEAC (à partir de SW 4, option)



840 D
NCU 571



840 D
NCU 572
NCU 573



810D
CCU 2

N250 R12 = \$AA_MM3[X]	Mémorisation de la valeur mesurée par le système de mesure 1 avec front descendant
N260 R13 = \$AA_MM4[X]	Mémorisation de la valeur mesurée par le système de mesure 2 avec front descendant
N270 FIN :	



Mesure avec effacement de la distance restant à parcourir, en mode 2

(traitement dans l'ordre programmé)

...	
N100 MEASA[X] = (2,1,-1,2,-2) G01 X100 F100	Mesure en mode 2 avec système de mesure activé. Attente du signal de mesure dans l'ordre : front montant palpeur 1, front descendant palpeur 1, front montant palpeur 2, front descendant palpeur 2, sur la course X = 100.
N110 STOPRE	Arrêt du prétraitement des blocs
N120 IF \$AC_MEA[1] == FALSE gotof PALPEUR2	Vérification du succès de la mesure avec palpeur 1
N130 R10 = \$AA_MM1[X]	Mémorisation de la valeur de mesure correspondant au premier événement déclencheur programmé (front montant palpeur 1).
N140 R11 = \$AA_MM2[X]	Mémorisation de la valeur de mesure correspondant au second événement déclencheur programmé (front montant palpeur 1).
N150 PALPEUR2 :	
N160 IF \$AC_MEA[2] == FALSE gotof FIN	Vérification du succès de la mesure avec palpeur 2
N170 R12 = \$AA_MM3[X]	Mémorisation de la valeur de mesure correspondant au troisième événement déclencheur programmé (front montant palpeur 2).
N180 R13 = \$AA_MM4[X]	Mémorisation de la valeur de mesure correspondant au quatrième événement déclencheur programmé (front montant palpeur 2).
N190 FIN :	

5.6 Mesure étendue MEASA, MEAWA, MEAC (à partir de SW 4, option)840 D
NCU 571840 D
NCU 572
NCU 573810D
CCU 2**Exemple de programmation****Mesure continue en mode 1 :**

(traitement dans l'ordre chronologique)

Jusqu'à 100 valeurs de mesure

...

N110 DEF REAL VAL_MESURE[100]

N120 DEF INT INDICE = 0

N130 MEAC[X] = (1,1,-1) G01 X1000 F100 Mesure en mode 1 avec système de mesure activé, mémorisation des valeurs mesurées sous \$AC_FIFO1, attente du signal de mesure avec front descendant du palpeur 1 sur le trajet vers X = 1000.

N135 STOPRE

N140 MEAC[X] = (0) Abandon de la mesure dès l'accostage de la position d'axe

N150 R1 = \$AC_FIFO1[4] Mémorisation du nombre de valeurs mesurées accumulées dans le paramètre R1.

N160 FOR INDICE = 0 TO R1-1

N170 VAL_MESURE[INDICE] = \$AC_FIFO1[0] Lecture et mémorisation des valeurs mesurées rangées dans \$AC_FIFO1.

N180 ENDFOR

Mesure avec effacement de la distance restant à parcourir après 10 valeurs mesurées

...

N10 WHEN \$AC_FIFO1[4]>=10 DO
 MEAC[x]=(0) DELDTG(x) Effacement de la distance restant à parcourir

N20 MEAC[x]=(1,1,1,-1) G01 X100 F500

N30 MEAC[X]=(0)

N40 R1 = \$AC_FIFO1[4] Nombre de valeurs mesurées

...

5.6 Mesure étendue MEASA, MEAWA, MEAC (à partir de SW 4, option)



840 D
NCU 571



840 D
NCU 572
NCU 573



810D
CCU 2



Les erreurs de programmation suivantes sont reconnues et signalées par un message d'erreur :

- MEASA/MEAWA programmé avec MEAS/MEAW dans un même bloc

Exemple :

```
N01 MEAS=1 MEASA[X]=(1,1) G01 F100 POS[X]=100
```

- MEASA/MEAWA avec un nombre de paramètres <2 ou >5

Exemple :

```
N01 MEAWA[X]=(1) G01 F100 POS[X]=100
```

- MEASA/MEAWA avec événement déclencheur différent de 1/ -1/ 2/ -2

Exemple :

```
N01 MEASA[B]=(1,1,3) B100
```

- MEASA/MEAWA avec mode incorrect

Exemple :

```
N01 MEAWA[B]=(4,1) B100
```

- MEASA/MEAWA avec événement déclencheur programmé deux fois

Exemple :

```
N01 MEASA[B]=(1,1,-1,2,-1) B100
```

- MEASA/MEAWA et axe GEO manquant

Exemple :

```
N01 MEASA[X]=(1,1) MEASA[Y]=(1,1) G01 X50 Y50 Z50 F100      Axe GEO X/Y/Z
```

- Contrat de mesure incohérent au niveau des axes GEO

Exemple :

```
N01 MEASA[X]=(1,1) MEASA[Y]=(1,1) MEASA[Z]=(1,1,2) G01  
X50 Y50 Z50 F100
```

5.7 Fonctions spéciales pour l'utilisateur OEM



840 D
NCU 572
NCU 573

5.7 Fonctions spéciales pour l'utilisateur OEM



Adresses OEM

L'utilisateur OEM définit la signification des adresses OEM.

La fonctionnalité est entrée par le biais de cycles de compilation. Cinq adresses OEM sont réservées.

Les descripteurs d'adresse peuvent être spécifiés.

Les adresses OEM sont admises dans chaque bloc.

Interpolations OEM

L'utilisateur OEM peut définir deux interpolations supplémentaires. La fonctionnalité est entrée par le biais de cycles de compilation.

Les noms des fonctions G (OEMIPO1, OEMIPO2) sont spécifiés par l'utilisateur OEM.

Les adresses OEM (voir ci-dessus) peuvent être utilisées spécialement dans le cas de l'interpolation OEM.

Groupes de fonctions G réservés (G800 à G819)

Deux groupes de fonctions G comptant chacun dix fonctions G OEM sont réservés à l'utilisateur OEM.

Ceci permet de sortir, pour une utilisation externe, les fonctions introduites par l'utilisateur OEM.

Fonctions et sous-programmes

En outre, l'utilisateur OEM peut créer des fonctions et des sous-programmes avec transfert de paramètres.

Frames

6.1 Appel de transformations de coordonnées par variables frames	6-188
6.2 Affectation de valeurs à des variables frames ou à des frames	6-192
6.3 Décalage grossier et décalage fin.....	6-199
6.4 Décalage DRF.....	6-200
6.5 Décalage d'origine externe	6-201
6.6 Programmation d'un décalage Preset, PRESETON	6-202
6.7 Désactivation des frames.....	6-203
6.8 Calcul d'un frame à partir de 3 points mesurés dans l'espace, MEAFRAME	6-204

840 D
NCU 571840 D
NCU 572

FM-NC



810D

NCU 573

6.1 Appel de transformations de coordonnées par variables frames

Définition de transformations de coordonnées avec des variables frames

Parallèlement aux possibilités de programmation déjà décrites dans le manuel de programmation "Notions de base", vous pouvez également décrire la position des systèmes de coordonnées avec des variables frames prédéfinies.

Qu'est-ce qu'une variable frame prédéfinie?

La variable frame indique dans le programme, **quels** systèmes de coordonnées doivent être mis en corrélation.

Les variables frames prédéfinies sont des mots-clés dont l'effet est déjà défini dans le langage de la commande et qui peuvent être exécutées dans le programme CN.

Variables frames possibles :

- frame de base (décalage de base)
- frames réglables
- frame programmable

Corrélation entre variables frames et frames

Avec les frames déjà connus, on peut savoir **comment** est décrite la position réciproque des systèmes de coordonnées en spécifiant des règles opératoires (translation, rotation etc.).

Par conséquent, vous pouvez aussi réaliser une transformation de coordonnées en indiquant la variable frame et le frame.

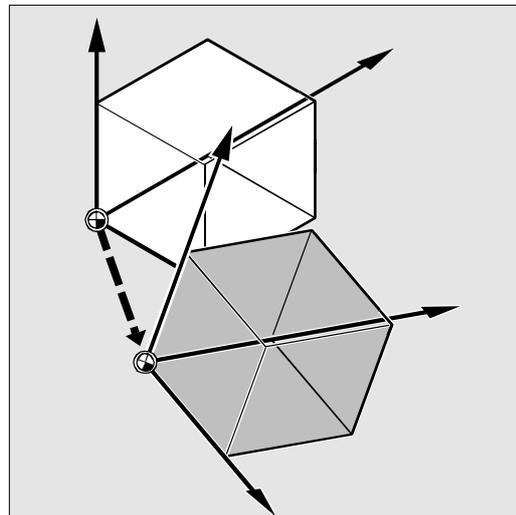
Exemple : `$P_PFRAME=CTRANS(X,10)`

Variable frame :

`$P_PFRAME` signifie : frame programmable courant.

Frame :

`CTRANS(X,10)` signifie : décalage d'origine programmable de l'axe X de 10 mm.



840 D
NCU 571840 D
NCU 572

FM-NC

810D
NCU 573

Vue d'ensemble des variables frames prédéfinies

\$P_IFRAME

Variable frame réglable courant qui établit la corrélation entre le système de coordonnées de base d'origine (BOS) et le système de coordonnées pièce d'origine (WOS).

\$P_IFRAME contient, après la programmation de G54 par exemple, la translation, la rotation, etc. définies par G54.

\$P_BFRAME

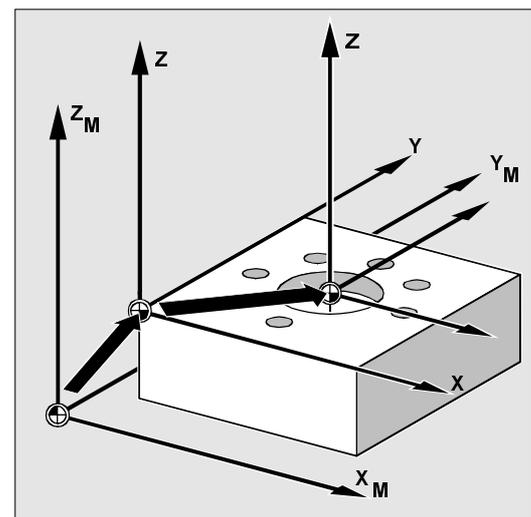
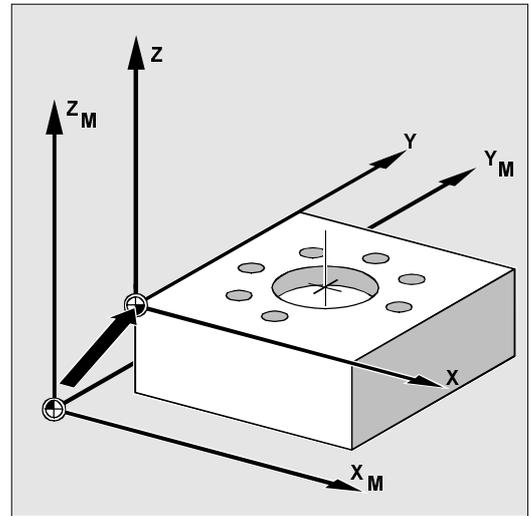
Variable frame de base courant qui établit la corrélation entre le système de coordonnées de base (SCB) et le système de coordonnées de base d'origine (BOS), que l'opérateur peut définir.

La transformation de coordonnées frame de base agit comme les frames réglables.

\$P_PFRAME

Variable frame programmable courant qui établit la corrélation entre le système de coordonnées pièce d'origine (WOS) et le système de coordonnées pièce (SCP).

\$P_PFRAME contient le frame qui résulte de la programmation de TRANS/ATRANS, ROT/AROT, SCALE/ASCALE, MIRROR/AMIRROR ou de l'affectation de CTRANS, CROT, CMIRROR, CSCALE au frame programmable.



6.1 Appel de transformations de coordonnées par variables frames



840 D
NCU 571



840 D
NCU 572



FM-NC



810D
NCU 573

\$P_ACTFRAME

Frame global résultant courant, qui se compose de la variable frame de base courant \$P_BFRAME, de la variable frame réglable courant \$P_IFRAME et de la variable frame programmable courant \$P_PFRAME.

\$P_ACTFRAME décrit l'origine pièce momentanément valide.

Si \$P_IFRAME, \$P_BFRAME ou \$P_PFRAME sont modifiés, \$P_ACTFRAME est recalculé.

\$P_ACTFRAME correspond à
\$P_BFRAME : \$P_IFRAME : \$P_PFRAME

Frames réglables prédéfinis \$P_UBFR[n]

L'affectation de valeurs à une variable frame prédéfinie \$P_UBFR[n] n'active pas automatiquement le frame de base correspondant ; l'activation n'a lieu que lors de l'exécution d'une instruction G500, G54...G599 (comme dans le cas des frames réglables prédéfinis \$P_UIFR[n]).

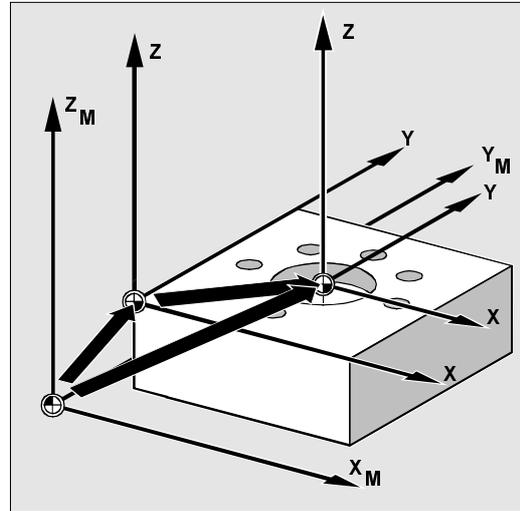
Frames réglables prédéfinis \$P_UIFR[n]

Une variable frame prédéfinie \$P_UIFR[n] permet de lire ou d'écrire dans le programme pièce les décalages d'origine réglables G54 à G599 .

Ces variables constituent un tableau unidimensionnel de type FRAME nommé \$P_UIFR[n].

Mise en correspondance avec les instructions G

En version standard, 5 frames réglables \$P_UIFR[0]...\$P_UIFR[4], c.-à-d. 5 instructions G équivalentes – G500 (désactivation) et G54 à G57 –, sont prédéfinis et des valeurs peuvent être enregistrées sous leurs adresses.



6.1 Appel de transformations de coordonnées par variables frames

840 D
NCU 571840 D
NCU 572
NCU 573

FM-NC



810D

$\$P_IFRAME=\$P_UIFR[0]$ correspond à G500

$\$P_IFRAME=\$P_UIFR[1]$ correspond à G54

$\$P_IFRAME=\$P_UIFR[2]$ correspond à G55

$\$P_IFRAME=\$P_UIFR[3]$ correspond à G56

$\$P_IFRAME=\$P_UIFR[4]$ correspond à G57

Le nombre de frames est modifiable par le biais d'un paramètre machine :

$\$P_IFRAME=\$P_UIFR[5]$ correspond à G505

... ..

$\$P_IFRAME=\$P_UIFR[99]$ correspond à G599



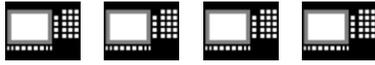
Ceci permet de créer 100 systèmes de coordonnées en tout, qui peuvent être appelés par tous les programmes, par exemple comme origines pour différents montages d'usinage.



La programmation de variables frames et de frames se fait dans un bloc CN spécifique.

Exception : programmation d'un frame réglable avec G54, G55,...

6.2 Affectation de valeurs à des variables frames ou à des frames



840 D 840 D FM-NC 810D
 NCU 571 NCU 572
 NCU 573

6.2 Affectation de valeurs à des variables frames ou à des frames



Dans le programme CN, des valeurs peuvent être affectées directement, des frames concaténés ou des frames affectés à d'autres frames.

Affectation directe de valeur



Programmation

```
$P_PFRAME=CTRANS (X, valeur d'axe, Y, valeur d'axe, Z, valeur d'axe,...)
$P_PFRAME=CROT (X, angle, Y, angle, Z, angle, ...)
$P_PFRAME=CSCALE (X, échelle, Y, échelle, Z, échelle, ...)
$P_PFRAME=CMIRROR (X, Y, Z)
```



La programmation de \$P_BFRAME est analogue à celle de \$P_PFRAME.



Signification des instructions

CTRANS	décalage (translation) dans les axes indiqués
CROT	rotation autour des axes indiqués
CSCALE	changement d'échelle dans les axes indiqués
CMIRROR	inversion du sens de l'axe indiqué



Fonction

Ces fonctions permettent d'affecter directement des valeurs à des frames ou à des variables frames, dans le programme CN.



Procédure

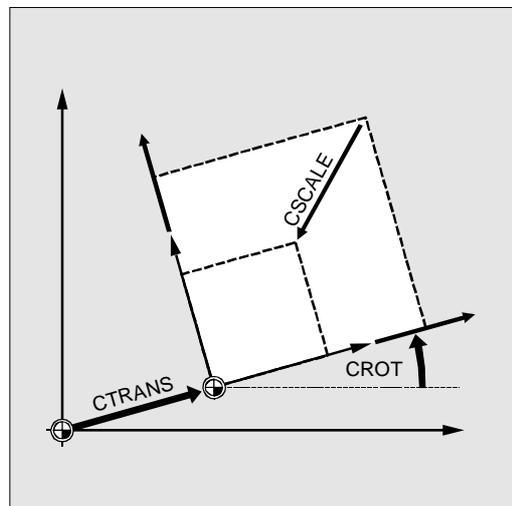
Vous pouvez programmer successivement plusieurs règles opératoires.

Exemple :

```
$P_PFRAME=CTRANS (...) : CROT (...) : CSCALE...
```

Attention : Les instructions doivent être liées entre elles par l'opérateur de concaténation deux-points (...): (...).

Les instructions sont ainsi, d'une part, combinées l'une à l'autre et, d'autre part, exécutées de manière additive dans l'ordre programmé.



840 D
NCU 571840 D
NCU 572
NCU 573

FM-NC



810D



Remarques complémentaires

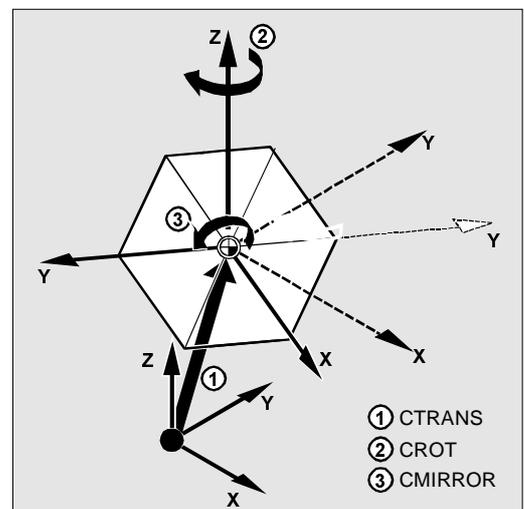
Les valeurs programmées avec les instructions indiquées sont affectées aux frames et mises en mémoire.

Les valeurs ne deviennent actives que lorsqu'elles sont affectées au frame d'une variable frame active \$P_BFRAME ou \$P_PFRAME.



Exemple de programmation

L'affectation des valeurs au frame programmable courant active la translation, la rotation et la fonction miroir.



```
N10 $P_PFRAME=CTRANS(X,10,Y,20,Z,5):CROT(Z,45):CMIRROR(Y)
```

6.2 Affectation de valeurs à des variables frames ou à des frames



840 D
NCU 571



840 D
NCU 572
NCU 573



FM-NC



810D

Lire et modifier des composantes d'un frame



Programmation (exemples)

```
R10=$P_UIFR[$P_UIFRNUM, X, RT]
```

L'angle de rotation RT autour de l'axe X issu du décalage d'origine réglable actuellement valide \$P_UIFRNUM doit être affecté à la variable R10.

```
R12=$P_UIFR[25, Z, TR]
```

La valeur de décalage TR dans Z, issue du bloc de données du frame n° 25 réglé, doit être affectée à la variable R12.

```
R15=$P_PFRAME[Y, TR]
```

La valeur de décalage TR en Y du frame programmable courant doit être affectée à la variable R15.

```
$P_PFRAME[X, TR]=25
```

La valeur de décalage TR en X du frame programmable courant doit être modifiée. X25 s'applique dès à présent.



Signification des instructions

\$P_UIFRNUM	Cette variable établit automatiquement le lien avec le décalage d'origine réglable actuellement valide.
P_UIFR[n, ..., ...]	En indiquant le n° de frame n, vous accédez au frame réglable n° n
TR	Indication de la composante qui doit être lue ou modifiée :
RT	TR translation, RT rotation, SC facteur d'échelle, MI fonction miroir.
SC	L'axe correspondant est également indiqué (voir les exemples).
MI	

840 D
NCU 571840 D
NCU 572
NCU 573

FM-NC



810D



Fonction

Vous avez la possibilité d'accéder à des données **individuelles** d'un frame, par ex. à une valeur de décalage ou à un angle de rotation bien défini.

Vous pouvez modifier ces valeurs ou les affecter à une autre variable.



Procédure

Appeler le frame

En indiquant la variable système `$P_UIFRNUM`, vous pouvez accéder directement au décalage d'origine courant, réglé avec `$P_UIFR` ou `G54, G55 ...` (`$P_UIFRNUM` contient le n° du frame réglé courant).

Vous appelez tous les autres frames réglables mémorisés `$P_UIFR` en indiquant le n° correspondant `$P_UIFR[n]`.

Pour les variables frames prédéfinies et les frames que vous avez définis vous-même, indiquez le nom, par exemple `$P_IFRAME`.

Appeler les données

Le nom de l'axe et la composante frame à laquelle vous voulez accéder ou que vous voulez modifier figurent entre crochets droits, par ex. `[X, RT]` ou `[Z, MI]`.

840 D
NCU 571840 D
NCU 572

FM-NC

810D
NCU 573

Combinaison de frames complets

Un frame complet peut être affecté à un autre frame.



Programmation (exemples)

```
DEF FRAME REGLAGE1
REGLAGE1=CTTRANS(X,10)
$P_PFRAME=REGLAGE1
```

```
DEF FRAME REGLAGE4
REGLAGE4=$P_PFRAME
$P_PFRAME=REGLAGE4
```

Les valeurs du frame REGLAGE1, que vous avez défini vous-même, sont affectées au frame programmable courant.

Le frame programmable courant est chargé en mémoire intermédiaire et renvoyé en mémoire principale, si nécessaire.



Remarques complémentaires

Plage de valeurs pour la rotation RT

Rotation autour du 1er axe géométrique :

-180° à +180°

Rotation autour du 2e axe géométrique :

-89.999° à +90°

Rotation autour du 3e axe géométrique :

-180° à +180°

Concaténation de frames



Programmation (exemples)

```
$P_IFRAME=$P_UIFR[15]:$P_UIFR[16]
```

```
$P_UIFR[3]=$P_UIFR[4]:$P_UIFR[5]
```

\$P_UIFR[15] contient par exemple des données pour des décalages d'origine. Les données de \$P_UIFR[16], par ex. des données pour des rotations, sont traitées ensuite sur cette base.

Le frame réglable 3 est créé par concaténation des frames réglables 4 et 5.

6.2 Affectation de valeurs à des variables frames ou à des frames



840 D
NCU 571



840 D
NCU 572
NCU 573



FM-NC



810D



Fonction

On utilise la concaténation de frames pour décrire par exemple plusieurs pièces disposées sur une palette et qui doivent être usinées en un seul cycle.



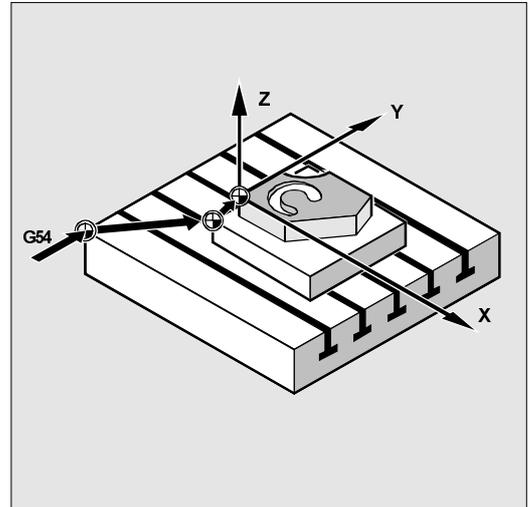
Procédure

Les frames sont concaténés dans l'ordre programmé, les composantes de frame (décalages, rotations, etc.) étant exécutées successivement et de manière additive.



Pour la description des tâches d'usinage sur palette, les composantes de frame pourraient, par exemple ne contenir que certaines valeurs partielles dont la concaténation pourraient générer différentes origines pièce.

N'oubliez pas que les frames doivent être reliés entre eux par l'opérateur de concaténation deux-points : .



840 D
NCU 571840 D
NCU 572
NCU 573

FM-NC



810D

Définition de nouveaux frames



Programmation

```
DEF FRAME PALETTE 1
```

```
PALETTE1=CTRANS (...) : CROT (...)...
```



Fonction

Outre les frames réglables prédéfinis décrits jusqu'ici, vous pouvez également créer de nouveaux frames.

Il s'agit de variables du type frame auxquelles vous attribuez un nom de votre choix.



Procédure

Les fonctions CTRANS, CROT, CSCALE et CMIRROR vous permettent d'affecter des valeurs à vos frames dans le programme CN.

Pour de plus amples informations, voyez les pages précédentes.

840 D
NCU 571840 D
NCU 572
NCU 573

FM-NC



810D

6.3 Décalage grossier et décalage fin



Fonction

Décalage fin

L'instruction `CFINE(X, ..., Y, ...)` permet de programmer un décalage fin du frame de base et de tous les frames réglables.

Décalage grossier

`CTRANS(...)` permet de définir le décalage grossier.

De la somme du décalage grossier et du décalage fin résulte le décalage total.

Exemples :

```
$P_UBFR=CTRANS(x, 10) : CFINE(x, 0.1) :  
CROT(x, 45)  
$P_UIFR[1]=CFINE(x, 0.5, y, 1.0, z, 0.1)
```

L'accès à la composante décalage fin d'un frame a lieu à l'aide de FI.

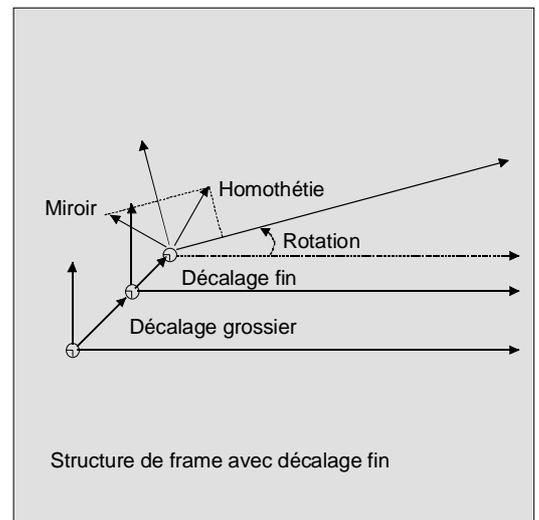
```
finex=$P_UIFR[$P_UIFRNUM, x, FI]
```

Un décalage fin n'est possible que si PM 18600:
`MM_FRAME_FINE_TRANS=1`.

Une modification de décalage fin effectuée par l'opérateur ne prend effet que lors de l'activation du frame correspondant, c.-à-d. que l'activation a lieu avec G500, G54...G599. Un décalage fin activé d'un frame reste actif tant que le frame est actif.



Le frame programmable ne possède pas de composante décalage fin. Si un frame avec décalage fin est affecté au frame programmable, le décalage total de ce dernier résulte de la somme du décalage grossier et du décalage fin. Lors de la lecture du frame programmable, le décalage fin est toujours nul.



6.4 Décalage DRF



840 D
NCU 571



840 D
NCU 572



FM-NC



810D
NCU 573

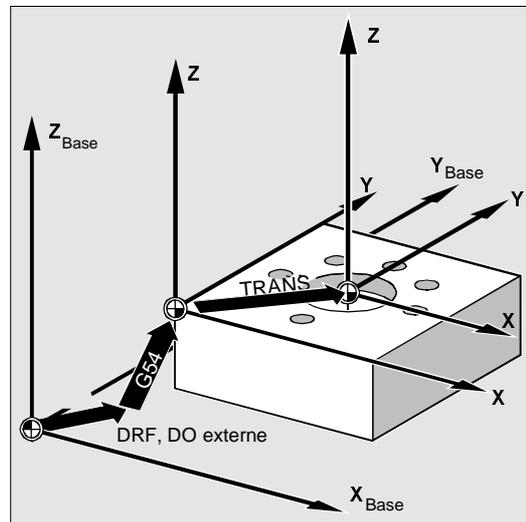
6.4 Décalage DRF

Décalage avec la manivelle, DRF

En plus de tous les décalages dont il a été question dans ce chapitre, vous pouvez définir des décalages d'origine avec la manivelle (décalage DRF).

Le décalage DRF agit dans le système de coordonnées de base. Voir le schéma.

Pour de plus amples informations, voir le "Manuel d'utilisation".



Effacer un décalage DRF, DRFOF

Avec DRFOF, le décalage généré avec la manivelle est effacé pour tous les axes affectés au canal. DRFOF doit se trouver dans un bloc CN spécifique.

840 D
NCU 571840 D
NCU 572
NCU 573

FM-NC



810D

6.5 Décalage d'origine externe

Décalage d'origine externe

Il s'agit d'une autre possibilité pour décaler l'origine entre le système de coordonnées de base et le système de coordonnées pièce.

Dans le cas du décalage d'origine externe, on ne peut programmer que des décalages linéaires.

Programmation des valeurs de décalage, \$AA_ETRANS

La programmation s'effectue en affectant des valeurs aux variables système spécifiques aux axes.

Affecter une valeur de décalage

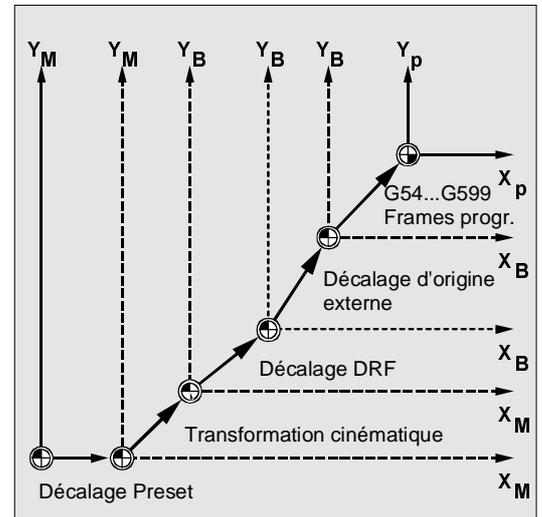
$\$AA_ETRANS[axe] = R_I$

R_I est la variable de calcul du type REAL qui contient la nouvelle valeur.

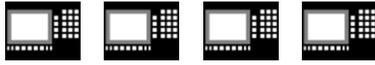
Le décalage externe n'est en général pas indiqué dans le programme pièce, mais spécifié par l'AP.



La valeur écrite dans le programme pièce ne prend effet que lorsque le signal correspondant est activé au niveau de l'interface VDI (interface NCU-AP).



6.6 Programmation d'un décalage Preset, PRESETON



840 D
NCU 571

840 D
NCU 572
NCU 573

FM-NC

810D

6.6 Programmation d'un décalage Preset, PRESETON



Programmation

PRESETON (AXE , VALEUR , ...)



Signification des instructions

PRESETON	préréglage des mémoires de valeurs réelles
AXE	indication de l'axe machine
VALEUR	nouvelle valeur réelle qui doit s'appliquer à l'axe indiqué



Fonction

Pour certaines applications spécifiques, il peut être nécessaire d'affecter une nouvelle valeur réelle programmée à la position courante d'un ou de plusieurs axes (à l'arrêt).



Procédure

L'affectation des valeurs réelles se fait dans le système de coordonnées machine - les valeurs se rapportent aux axes machine.

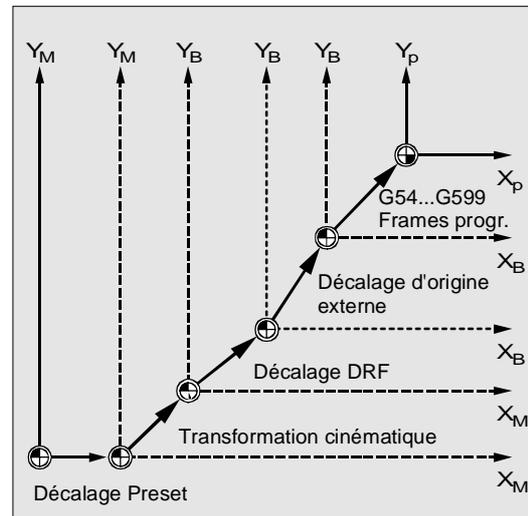
Exemple :

```
N10 G0 A760
N20 PRESETON(A1,60)
```

L'axe A accoste la position 760. A la position 760, l'axe machine A1 reçoit la nouvelle valeur réelle 60. Dès lors, le positionnement s'effectue dans le nouveau système de coordonnées.



Avec la fonction PRESETON, le point de référence perd sa validité. Par conséquent, utilisez cette fonction uniquement pour les axes sans point de référence obligatoire. Pour rétablir le système initial, il est nécessaire d'accoster les points de référence avec G74 – voir chapitre 3.1.



840 D
NCU 571840 D
NCU 572
NCU 573

FM-NC



810D

6.7 Désactivation des frames

Désactivation de transformations de coordonnées

On fait ici la distinction entre :

- désactivation pendant un bloc (non modale)
- désactivation modale.

Pour les instructions correspondantes, voir Tables.



Signification des instructions

G500	désactivation de tous les frames réglables
DRFOF	désactivation (effacement) des décalages par manivelle (DRF)
G53	désactivation non modale (pendant un bloc) de tous les frames programmables et réglables
G153	désactivation non modale (pendant un bloc) de tous les frames programmables et réglables ainsi que du frame de base
SUPA	désactivation non modale (pendant un bloc) de tous les frames programmables et réglables, des décalages par manivelle (DRF) et du décalage Preset



Remarques complémentaires

Pour effacer les frames programmables, il suffit de programmer une composante TRANS, ROT, SCALE, MIRROR sans indication d'axe.

Pour plus d'informations, voir la partie correspondante dans le présent chapitre.

6.8 Calcul d'un frame à partir de 3 points mesurés dans l'espace



840 D
NCU 571



840 D
NCU 572
NCU 573



FM-NC



810D

6.8 Calcul d'un frame à partir de 3 points mesurés dans l'espace MEAFRAME



MEAFRAME est une extension du langage 840D pour l'assistance des cycles de mesure.

Cette fonction est disponible à partir de SW 4.3



Fonction

Lorsqu'une pièce est positionnée pour l'usinage, sa position dans le système de coordonnées cartésiennes machine est en général décalée et pivotée par rapport à sa position idéale.

Pour un usinage ou une mesure précise, un réglage physique onéreux ou une modification des déplacements dans le programme pièce est nécessaire.

Il est également possible de déterminer un frame par palpation de trois points dans l'espace dont les positions idéales sont connues. Le palpation s'effectue avec un palpeur à contact ou optique qui détermine la position d'alésages ou de sphères disposés avec précision sur le support de la pièce.

La fonction MEAFRAME calcule le frame à partir de trois points idéaux et des points mesurés correspondants.

Pour que les coordonnées mesurées puissent être mises en relation avec les coordonnées idéales par une rotation/translation combinées, il doit y avoir congruence entre le triangle formé par les points mesurés et le triangle idéal. Ceci est obtenu par un algorithme de compensation qui minimise la somme des carrés des écarts qui permettent de passer du triangle mesuré au triangle idéal.

La distorsion effectivement nécessaire des points mesurés étant le reflet de la qualité de la mesure, elle est fournie par MEAFRAME en tant que variable supplémentaire.

6.8 Calcul d'un frame à partir de 3 points mesurés dans l'espace

840 D
NCU 571840 D
NCU 572
NCU 573

FM-NC



810D



Programmation

```
MEAFRAME (IDEAL_POINT, MEAS_POINT, FIT_QUALITY)
```



Signification des instructions

MEAFRAME	calcul d'un frame à partir de 3 points mesurés dans l'espace
IDEAL_POINT	tableau Real à 2 dim. qui contient les 3 coordonnées des points idéaux
MEAS_POINT	tableau Real à 2 dim. qui contient les 3 coordonnées des points mesurés
FIT_QUALITY	variable Real qui fournit les informations suivantes : <ul style="list-style-type: none"> -1: Les points idéaux se trouvent approximativement sur une droite : Le frame n'a pas pu être calculé. La variable frame en retour contient un frame neutre. -2: Les points mesurés se trouvent approximativement sur une droite : Le frame n'a pas pu être calculé. La variable frame en retour contient un frame neutre. -4: Le calcul de la matrice de rotation est impossible pour une autre raison valeur positive : Somme des distorsions (écarts entre les points) nécessaire pour passer du triangle mesuré à un triangle congruent au triangle idéal.



Exemple d'application

```
Programme pièce 1
;
DEF FRAME CORR_FRAME
;
; Définition des points
DEF REAL IDEAL_POINT[3,3] = SET(10.0,0.0,0.0, 0.0,10.0,0.0, 0.0,0.0,10.0)
DEF REAL MEAS_POINT[3,3] = SET(10.1,0.2,-0.2, -0.2,10.2,0.1, -0.2,0.2, 9.8);      pour test
DEF REAL FIT_QUALITY = 0
;
DEF REAL ROT_FRAME_LIMIT = 5;          décalage angulaire maxi de la position de la pièce : 5°
DEF REAL FIT_QUALITY_LIMIT = 3;        décalage maxi de 3 mm entre le triangle idéal ;
                                         et le triangle mesuré
DEF REAL SHOW_MCS_POS1[3]
DEF REAL SHOW_MCS_POS2[3]
DEF REAL SHOW_MCS_POS3[3]
; =====
```

840 D
NCU 571840 D
NCU 572
NCU 573

FM-NC



810D

```

;
N100 G01 G90 F5000
N110 X0 Y0 Z0
;
N200 CORR_FRAME=MEAFRAME(IDEAL_POINT,MEAS_POINT,FIT_QUALITY)
;
N230 IF FIT_QUALITY < 0
SETAL(65000)
GOTOF NO_FRAME
ENDIF
;
N240 IF FIT_QUALITY > FIT_QUALITY_LIMIT
SETAL(65010)
GOTOF NO_FRAME
ENDIF
;
N250 IF CORR_FRAME[X,RT] > ROT_FRAME_LIMIT;           Limitation du 1er angle RPY
SETAL(65020)
GOTOF NO_FRAME
ENDIF
;
N260 IF CORR_FRAME[Y,RT] > ROT_FRAME_LIMIT;           Limitation du 2me angle RPY
SETAL(65021)
GOTOF NO_FRAME
ENDIF
;
N270 IF CORR_FRAME[Z,RT] > ROT_FRAME_LIMIT;           Limitation du 3me angle RPY
SETAL(65022)
GOTOF NO_FRAME
ENDIF
;
N300 $P_IFRAME=CORR_FRAME;           Activer le frame de palpé avec un frame réglable
;
; Vérifier le frame en positionnant les axes géométriques aux points idéaux
;
N400 X=IDEAL_POINT[0,0] Y=IDEAL_POINT[0,1] Z=IDEAL_POINT[0,2]
N410 SHOW_MCS_POS1[0]=$AA_IM[X]
N420 SHOW_MCS_POS1[1]=$AA_IM[Y]
N430 SHOW_MCS_POS1[2]=$AA_IM[Z]
;
N500 X=IDEAL_POINT[1,0] Y=IDEAL_POINT[1,1] Z=IDEAL_POINT[1,2]
N510 SHOW_MCS_POS2[0]=$AA_IM[X]
N520 SHOW_MCS_POS2[1]=$AA_IM[Y]
N530 SHOW_MCS_POS2[2]=$AA_IM[Z]
;
N600 X=IDEAL_POINT[2,0] Y=IDEAL_POINT[2,1] Z=IDEAL_POINT[2,2]
N610 SHOW_MCS_POS3[0]=$AA_IM[X]

```

6.8 Calcul d'un frame à partir de 3 points mesurés dans l'espace



840 D
NCU 571



840 D
NCU 572
NCU 573



FM-NC



810D

```
N620 SHOW_MCS_POS3[1]=$AA_IM[Y]
N630 SHOW_MCS_POS3[2]=$AA_IM[Z]
;
N700 G500;    Désactiver le frame réglable
;
NO_FRAME:
M0
M30
```


Usinage cinq axes

7.1 Transformation cinq axes - TRAORI, TRAFOOF	7-210
7.2 Référence des axes d'orientation - ORIWKS, ORIMKS	7-219
7.3 Positions singulières et leur traitement	7-221
7.4 TRANSMIT, TRAFOOF	7-222
7.5 TRACYL, TRAFOOF, OFFN.....	7-226
7.6 TRAANG, TRAFOOF	7-232



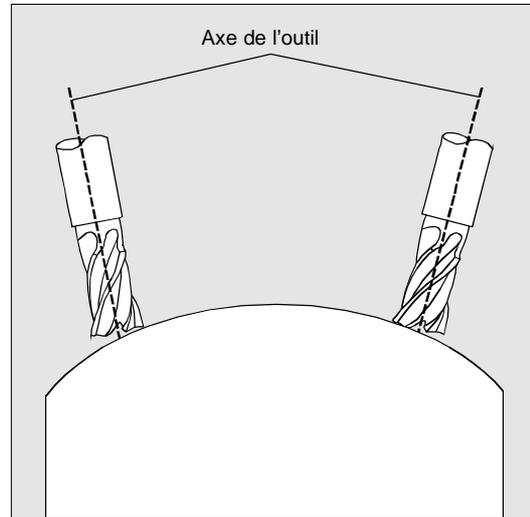
840 D
NCU 572
NCU 573

7.1 Transformation cinq axes - TRAORI, TRAFOOF



Pour réunir des conditions de coupe optimales lors de l'usinage de surface courbes, il est nécessaire de pouvoir modifier l'angle d'inclinaison de l'outil. Pour ce faire, deux axes rotatifs au moins sont nécessaires en plus des trois axes linéaires.

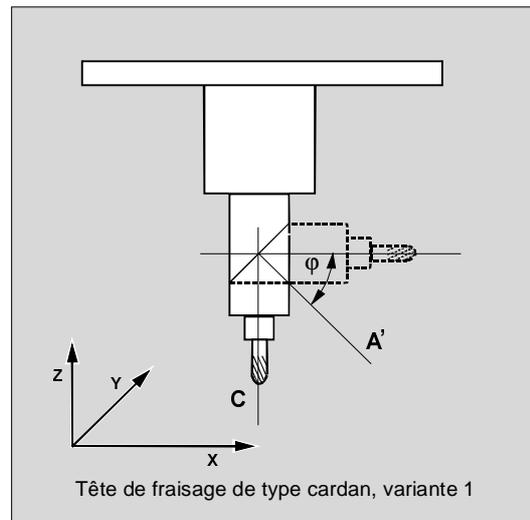
La configuration correspondante de la machine est rangée dans les paramètres machine relatifs aux axes.



Tête de fraissage de type cardan

Trois axes linéaires (X, Y, Z) et deux axes rotatifs déterminent l'angle d'inclinaison et le point d'usinage de l'outil. L'un des deux axes rotatifs - dans l'exemple A' - est incliné, souvent à 45°.

La disposition des axes rotatifs et l'orientation de l'outil sont définis au travers des paramètres machine, en fonction de la cinématique de la machine. Les exemples choisis ici se rapportent à la cinématique de la machine CA !





840 D
NCU 572
NCU 573

Les corrélations possibles sont les suivantes :

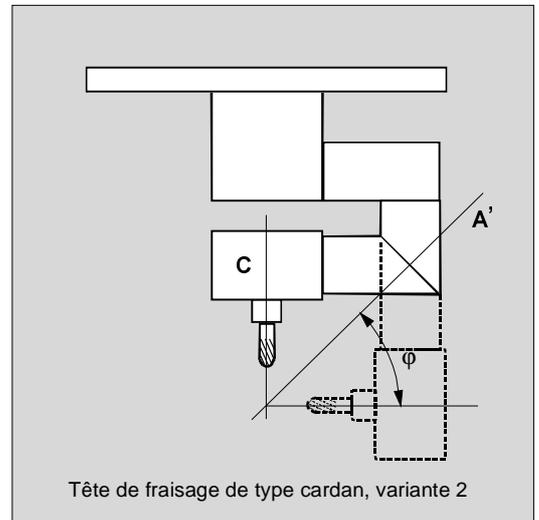
A' forme l'angle φ avec l'axe X

B' forme l'angle φ avec l'axe Y

C' forme l'angle φ avec l'axe Z

L'angle φ peut être défini dans la plage de 0° à $+89^\circ$ par le biais des paramètres machine.

Il convient de régler le plan de travail actif (G17, G18, G19) défini dans le programme CN, de façon à ce que la correction de longueur d'outil agisse dans la direction de l'orientation de l'outil.



Transformation avec axe linéaire pivotant

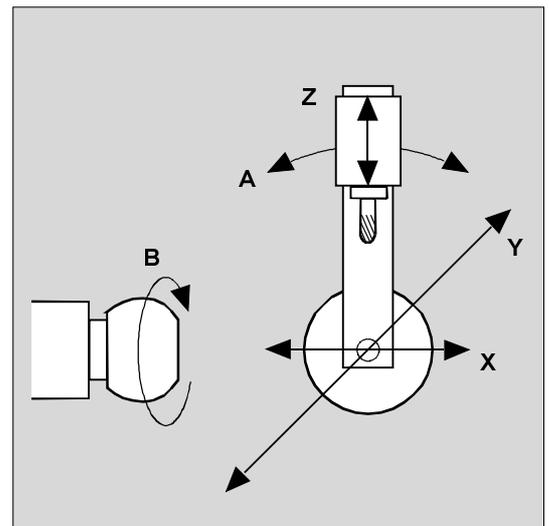
Il s'agit d'une configuration avec une pièce mobile et un outil mobile.

La cinématique se compose de trois axes linéaires (X, Y, Z) et de deux axes rotatifs orthogonaux. Le premier axe rotatif se déplace sur un chariot à deux axes linéaires croisés. L'outil est parallèle au troisième axe linéaire.

Le second axe rotatif fait pivoter la pièce.

Le troisième axe linéaire (axe de pivotement) se situe dans le plan du chariot croisé.

La disposition des axes rotatifs et l'orientation de l'outil sont définis au travers des paramètres machine, en fonction de la cinématique de la machine.



Les corrélations possibles sont les suivantes :

Axes :

1er axe rotatif

2e axe rotatif

Axe linéaire de pivotement

Configuration axiale :

A A B B C C

B C A C A B

Z Y Z X Y X



840 D
NCU 572
NCU 573

Transformations 3 axes / 4 axes

Les transformations 3 et 4 axes sont des formes spéciales de la transformation 5 axes.

L'utilisateur peut configurer deux ou trois axes linéaires et un axe rotatif. On suppose que l'axe rotatif est perpendiculaire au plan d'orientation.

L'orientation de l'outil est possible uniquement dans le plan perpendiculaire à l'axe rotatif. Les transformations sont plus particulièrement destinées aux machines avec outil et pièce mobiles.

La configuration et la programmation des transformations 3 axes ou 4 axes sont identiques aux transformations 5 axes.



Programmation

TRAORI (n)
TRAFOOF



Signification des instructions

TRAORI	Activation de la première transformation d'orientation convenue
TRAORI (n)	Activation de la n-ième transformation d'orientation convenue
n	numéro de la transformation (n = 1 ou 2), TRAORI(1) correspond à TRAORI
TRAFOOF	Désactivation de la transformation



Remarques complémentaires

Dès que la transformation est activée, les indications de position (X, Y, Z) se réfèrent toujours à la pointe de l'outil.

Toute modification de position des axes rotatifs impliqués dans la transformation entraîne des déplacements compensateurs des autres axes machine, qui font que la position de la pointe de l'outil reste inchangée.



840 D
NCU 572
NCU 573

Points à observer pour l'activation

- Il n'y a pas insertion d'un bloc intermédiaire de déplacement (chanfrein/rayon).
- Une séquence de blocs splines doit être clôturée
- La correction de rayon d'outil doit être désactivée
- Le frame actif avant TRAORI est désactivé par la commande
- Une limitation active de la zone de travail est désactivée par la commande pour les axes impliqués dans la transformation
- Le contournage et l'arrondissement du contour sont interrompus
- Les décalages DRF dans les axes impliqués dans la transformation ne doivent pas subir de modification entre le prétraitement des blocs et leur exécution.
- Quand la transformation est activée et en fonctionnement normal, différents frames réglables doivent être activés, car ces frames transforment le SCP en système de coordonnées de base SCB.

Changement d'outil

Un changement d'outil n'est autorisé que si la correction de rayon d'outil est désactivée.

Un changement de la correction de longueur d'outil et une activation/désactivation de la correction de rayon d'outil ne doivent pas être programmées dans le même bloc.

Changement de frame

Toutes les instructions qui se réfèrent uniquement au système de coordonnées de base sont autorisées (frame, correction de rayon d'outil). Un changement de frame avec G91 (cotes relatives) ne sera cependant pas traité à part, contrairement à ce qui se passe lorsque la transformation est inactive. L'incrément à parcourir est traité dans le système de coordonnées pièce du nouveau frame, quel que soit le frame qui était actif dans le bloc précédent.



840 D
NCU 572
NCU 573

Exclusions

Les axes impliqués dans la transformation ne peuvent pas être utilisés

- comme axes Preset (alarme)
- pour l'accostage d'un point fixe (alarme)
- pour la prise de référence (alarme)



Programmation de l'orientation de l'outil

En règle générale, les programmes à 5 axes sont générés par les systèmes CAO/FAO et ne sont pas à entrer au pied de la commande. Les explications qui suivent s'adressent par conséquent en priorité aux programmeurs de postprocesseurs.

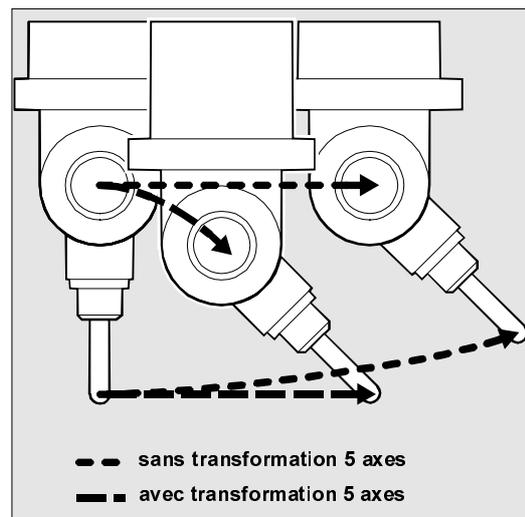
Il existe 3 possibilités pour programmer l'orientation de l'outil :

1. Programmation du déplacement des axes rotatifs
Le changement d'orientation s'effectue toujours dans le système de coordonnées de base ou dans le système de coordonnées machine. Les axes d'orientation sont déplacés en tant qu'axes synchrones.
2. Programmation en angles d'Euler ou en angles RPY par le biais de A2, B2, C2
ou
programmation du vecteur d'orientation par le biais de A3, B3, C3. Le vecteur d'orientation est dirigé de la pointe de l'outil vers le porte-outil.
3. Programmation de l'angle d'avance LEAD et de l'angle latéral TILT (fraisage en bout).

Dans tous les cas, la programmation de l'orientation est autorisée uniquement si une transformation 5 axe est activée.



Avantage : Ces programmes sont transposables sur n'importe quelle cinématique de machine.





840 D
NCU 572
NCU 573



Programmation

G1 X Y Z A B C	Programmation du déplacement des axes rotatifs
G1 X Y Z A2= B2= C2=	Programmation en angles d'Euler
G1 X Y Z A3= B3= C3=	Programmation du vecteur d'orientation
G1 X Y Z A4= B4= C4=	Programmation de la courbure de la trajectoire en début de bloc
G1 X Y Z A5= B5= C5=	Programmation de la courbure de la trajectoire en fin de bloc, pour fraisage en bout
LEAD	Angle d'avance pour la programmation de l'orientation de l'outil
TILT	Angle latéral pour la programmation de l'orientation de l'outil



Un paramètre machine permet de choisir entre les angles d'Euler et les angles RPY.



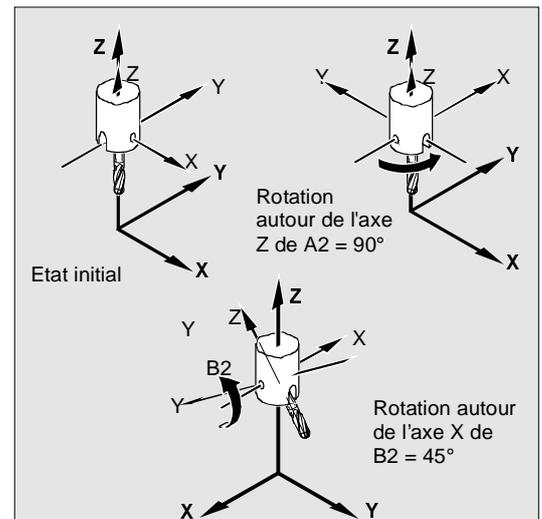
Programmation en angles d'Euler

Les valeurs programmées avec A2, B2, C2 lors de la programmation de l'orientation sont interprétées comme angles d'Euler (en degrés).

Le vecteur d'orientation est obtenu en faisant pivoter un vecteur en direction Z tout d'abord avec A2 autour de l'axe Z, puis avec B2 autour du nouvel axe X et enfin avec C2 autour du nouvel axe Z.



Dans ce cas, la valeur de C2 (rotation autour d'un nouvel axe Z) est sans signification et n'est pas à programmer.





840 D
NCU 572
NCU 573

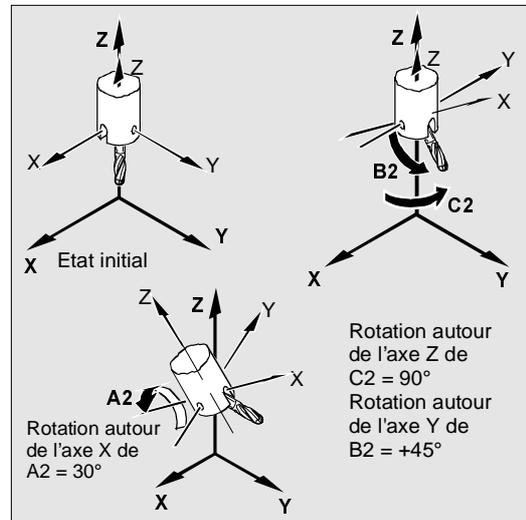


Programmation en angles RPY

Les valeurs programmées avec A2, B2, C2 lors de la programmation de l'orientation sont interprétées comme angles RPY (en degrés).

Le vecteur d'orientation est obtenu en faisant pivoter un vecteur en direction Z d'abord avec C2 autour de l'axe Z, puis avec B2 autour du nouvel axe Y et en fin avec A2 autour du nouvel axe X.

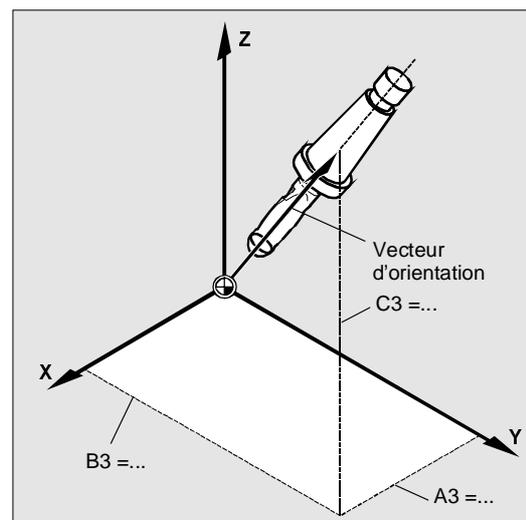
Contrairement à la programmation en angles d'Euler les trois valeurs agissent ici sur le vecteur d'orientation



Programmation du vecteur d'orientation

Les composantes du vecteur d'orientation sont programmées avec A3, B3, C3. Le vecteur est dirigé vers le porte-outil ; la longueur du vecteur est sans importance.

La commande attribue la valeur zéro aux composantes non programmées du vecteur.





840 D
NCU 572
NCU 573

Fraisage en bout

Le fraisage en bout sert à l'usinage des surfaces à courbure quelconque.

Pour ce type de fraisage 3D, vous avez besoin de la description, ligne par ligne, des différentes trajectoires 3D à la surface de la pièce.

Les calculs sont généralement réalisés à l'aide d'un système de FAO, en tenant compte de la forme et des dimensions de l'outil.

Les blocs CN générés sont ensuite lus dans la commande par l'intermédiaire de postprocesseurs.

Description des surfaces

La description de la courbure de la trajectoire s'effectue par le biais de vecteurs normaux à la surface ayant les composantes suivantes :

A4, B4, C4 : vecteur de début au début du bloc

A5, B5, C5 : vecteur de fin en fin du bloc

i un bloc ne contient que le vecteur de début, le vecteur normal à la surface restera constant pendant tout le bloc.

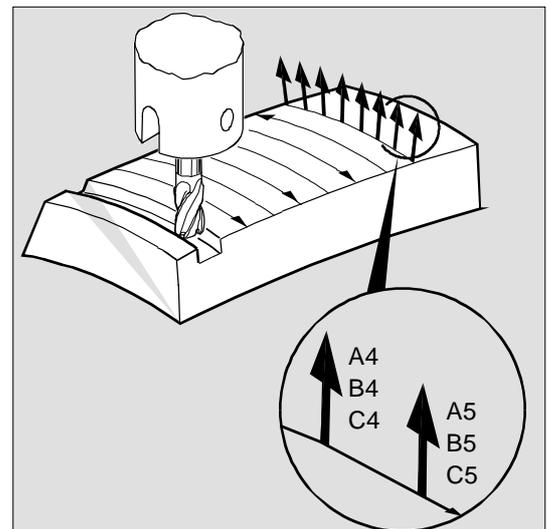
Si un bloc ne contient que le vecteur de fin, une interpolation circulaire de grand rayon est effectuée de la valeur de fin du bloc précédent jusqu'à la valeur de fin programmée.

Si le vecteur de début et le vecteur de fin sont programmés, une interpolation circulaire de grand rayon est également effectuée entre les deux directions. Ceci permet d'obtenir des trajectoires lisses.

Par défaut, en état de préréglage, les vecteurs normaux à la surface sont orientés dans la direction Z, indépendamment du plan G17 à G19 activé.

La longueur des vecteurs est sans importance.

La valeur zéro est attribuée par défaut aux composantes de vecteur non programmées.





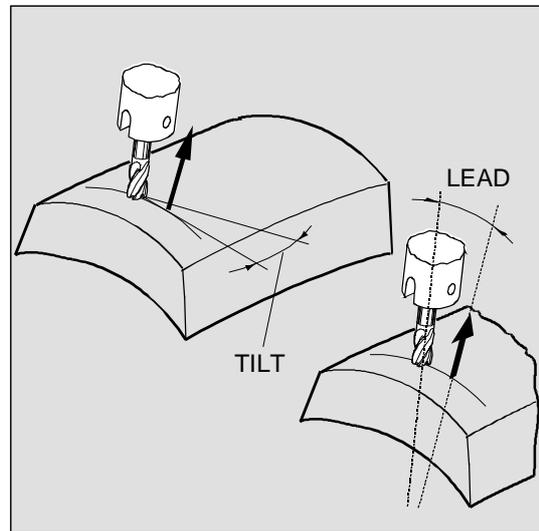
840 D
NCU 572
NCU 573

Dans le cas où ORIWKS (voir pages suivantes) est activée, les vecteurs normaux à la surface se réfèrent au frame actif et subissent également une rotation en cas de rotation du frame.

Le vecteur normal à la surface doit être perpendiculaire à la tangente à la trajectoire avec une zone de tolérance qui est réglable par le biais d'un paramètre machine, sinon une alarme est émise.

Programmation de l'orientation d'outil avec LEAD et TILT

L'orientation de l'outil est déterminée à partir de la tangente à la trajectoire, du vecteur normal à la surface, de l'angle d'avance LEAD et de l'angle latéral TILT en fin de bloc.



LEAD	Angle projeté, rapporté au vecteur normal à la surface
TILT	Angle projeté, rapporté à la tangente à la trajectoire

L'orientation est toujours acquise en fin de bloc, même si le bloc est tronqué, par exemple au niveau d'un angle rentrant.



840 D
NCU 572
NCU 573

7.2 Référence des axes d'orientation - ORIWKS, ORIMKS



Programmation

N. . . ORIMKS
OU
N. . . ORIWKS



Signification des instructions et paramètres

ORIMKS	Rotation dans le système de coordonnées machine
ORIWKS	Rotation dans le système de coordonnées pièce



Fonction

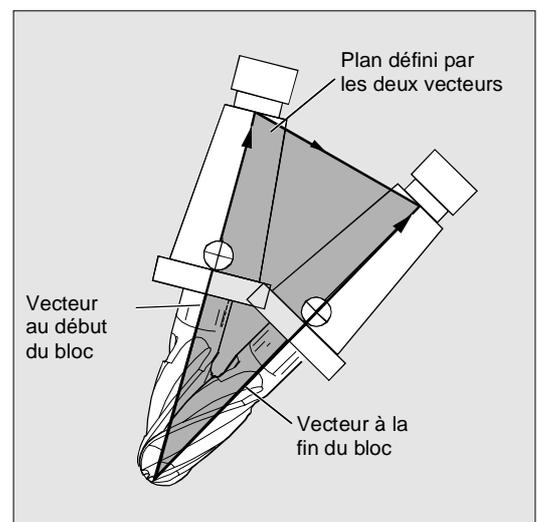
Lors de la programmation de l'orientation dans le système de coordonnées pièce par le biais des angles d'Euler ou RPY ou avec le vecteur d'orientation, les fonctions ORIMKS/ORIWKS permettent de régler le déroulement du mouvement de rotation.



Procédure

Avec ORIMKS, le déplacement effectué par l'outil dépend de la cinématique de la machine. Dès qu'il y a un changement d'orientation avec pointe de l'outil fixe, il y a exécution d'une interpolation linéaire entre les positions des axes rotatifs.

Avec ORIWKS, le déplacement de l'outil est indépendant de la cinématique de la machine. Dès qu'il y a un changement d'orientation avec pointe de l'outil fixe, l'outil se déplace dans le plan défini par le vecteur de début et le vecteur de fin.



7.2 Référence des axes d'orientation - ORIWKS, ORIMKS



840 D
NCU 572
NCU 573



Remarques complémentaires

ORIWKS est pré-réglée. Si on ne connaît pas d'avance la machine sur laquelle doit fonctionner un programme à 5 axes, travaillez toujours avec ORIWKS. Les déplacements effectivement exécutés par la machine dépendent de sa cinématique.

ORIMKS permet de programmer les mouvements réels de la machine pour éviter, par exemple, les collisions avec des montages d'usinage, etc.



840 D
NCU 572
NCU 573

7.3 Positions singulières et leur traitement



Remarques sur ORIWKS :

Les mouvements d'orientation au voisinage de la position singulière d'une machine à 5 axes nécessitent des déplacements importants de la part des axes machine. (C'est le cas par exemple sur une tête pivotante, avec C comme axe de rotation et A comme axe de pivotement, toutes les positions avec $A = 0$ sont des positions singulières).

Pour ne pas surcharger les axes machine, le pilotage de la vitesse réduit fortement la vitesse tangentielle au voisinage des points singuliers.

Les paramètres machine

```
$MC_TRAFO5_NON_POLE_LIMIT
```

```
$MC_TRAFO5_POLE_LIMIT
```

permettent de paramétrer la transformation de façon à ce que les déplacements d'orientation au voisinage du pôle passent effectivement par le pôle, ce qui donne un usinage plus rapide.

7.4 TRANSMIT, TRAFOOF

840 D
NCU 571840 D
NCU 572

FM-NC

810D
NCU 573

7.4 TRANSMIT, TRAFOOF



Programmation

TRANSMIT OU TRANSMIT (t)
TRAFOOF

Signification

TRANSMIT	Activation de la première fonction TRANSMIT convenue
TRANSMIT (n)	Activation de la n-ième fonction TRANSMIT convenue, n étant limité à 2 (TRANSMIT(1) correspond à TRANSMIT).
TRAFOOF	Désactivation d'une transformation active



Une transformation TRANSMIT active est également désactivée quand, dans le canal concerné, il y a activation de l'une des autres transformations (par ex. TRACYL, TRAANG, TRAORI).

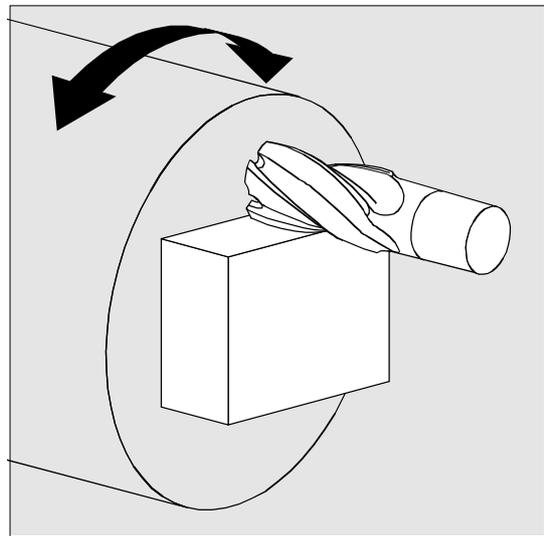


La fonction TRANSMIT offre les possibilités suivantes :

- Usinage frontal de pièces tournées sur le tour (trous, contours)
- Un système de coordonnées cartésiennes peut être utilisé pour la programmation de ces usinages.
- La commande transforme les déplacements programmés dans le système de coordonnées cartésiennes en déplacements des axes réels de la machine (cas standard) :
 - axe rotatif
 - axe de pénétration perpendiculaire à l'axe de rotation
 - axe longitudinal parallèle à l'axe de rotation

Les axes linéaires sont perpendiculaires l'un par rapport à l'autre.

- Un décalage du centre de l'outil par rapport à l'axe de rotation est permis.
- Le pilotage de vitesse tient compte des limitations définies pour les rotations.



840 D
NCU 571840 D
NCU 572
NCU 573

FM-NC



810D



Remarques complémentaires

Points à observer pour l'activation

- Il n'y a pas insertion d'un bloc intermédiaire de déplacement (chanfrein/rayon).
- Une séquence de blocs splines doit être clôturée
- La correction de rayon d'outil doit être désactivée
- Le frame actif avant TRANSMIT est désactivé par la commande.
- Une limitation active de la zone de travail est désactivée par la commande pour les axes impliqués dans la transformation
- Le contournage et l'arrondissement du contour sont interrompus
- Les décalages DRF dans les axes impliqués dans la transformation ne doivent pas subir de modification entre le prétraitement des blocs et leur exécution.
- Quand la transformation est activée et en fonctionnement normal, différents frames réglables doivent être activés, car ces frames transforment le SCP en système de coordonnées de base SCB.

Points à observer pour la désactivation

Ils sont les mêmes que pour l'activation, avec en plus :

- une correction de la longueur d'outil dans l'axe virtuel n'est plus exécutée.

Changement d'outil

Un changement d'outil n'est autorisé que si la correction de rayon d'outil est désactivée.

Un changement de la correction de longueur d'outil et une activation/désactivation de la correction de rayon d'outil ne doivent pas être programmées dans le même bloc.

7.4 TRANSMIT, TRAFOOF



840 D
NCU 571



840 D
NCU 572



FM-NC



810D
NCU 573

Changement de frame

Toutes les instructions qui se réfèrent uniquement au système de coordonnées de base sont autorisées (frame, correction de rayon d'outil). Un changement de frame avec G91 (cotes relatives) ne sera cependant pas traité à part, contrairement à ce qui se passe lorsque la transformation est inactive. L'incrément à parcourir est traité dans le système de coordonnées pièce du nouveau frame, quel que soit le frame qui était actif dans le bloc précédent.

Axe rotatif

L'axe rotatif ne peut pas être programmé, étant donné qu'il est affecté comme axe géométrique et qu'il n'est donc pas programmable directement comme axe de canal.

Pôle

Le franchissement du pôle (origine du système de coordonnées cartésiennes) n'est pas admis. Un déplacement passant par le pôle est stoppé au pôle et une alarme est émise. En cas de décalage du centre de la fraise, le déplacement s'arrête au bord de la zone non autorisée.

Exclusions

Les axes impliqués dans la transformation ne peuvent pas être utilisés

- comme axes Preset (alarme)
- pour l'accostage d'un point fixe (alarme)
- pour la prise de référence (alarme)

840 D
NCU 571840 D
NCU 572
NCU 573

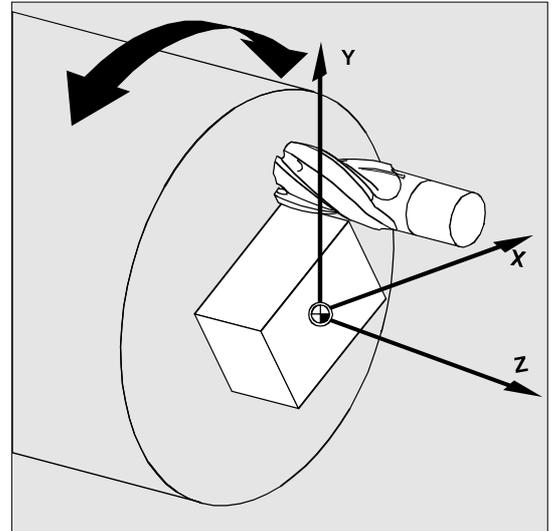
FM-NC



810D



Exemple de programmation



N10 T1 D1 G54 G17 G90 F5000 G94	Sélection de l'outil
N20 G0 X20 Z10 SPOS=45	Accostage de la position de départ
N30 TRANSMIT	Activation de la fonction TRANSMIT
N40 ROT RPL=-45	Réglage du frame
N50 ATRANS X-2 Y10	
N60 G1 X10 Y-10 G41 OFFN=1	Ebauche d'un carré ; surépaisseur 1 mm
N70 X-10	
N80 Y10	
N90 X10	
N100 Y-10	
N110 G0 Z20 G40 OFFN=0	Changement d'outil
N120 T2 D1 X15 Y-15	
N130 Z10 G41	
N140 G1 X10 Y-10	Finition du carré
N150 X-10	
N160 Y10	
N170 X10	
N180 Y-10	
N190 Z20 G40	Désactivation du frame
N200 TRANS	
N210 TRAFOOF	
N220 G0 X20 Z10 SPOS=45	Accostage de la position de départ
N230 M30	

7.5 TRACYL, TRAFOOF, OFFN

840 D
NCU 571840 D
NCU 572
NCU 573

FM-NC



810D

7.5 TRACYL, TRAFOOF, OFFN



Programmation

TRACYL(d) ou TRACYL(d,t)
TRAFOOF

TRACYL (d)

Activation de la première fonction TRACYL
convenue

TRACYL (d , n)

Activation de la n-ième fonction TRACYL
convenue, n étant limité à 2. TRACYL(d,1)
correspond à TRACYL(d).

d

Valeur pour le diamètre courant du cylindre
à usiner.

TRAFOOF

Désactivation de la transformation

OFFN

Décalage normal au contour : distance entre
la paroi de la rainure et le contour de
référence programmé

Une transformation TRACYL active est également désactivée quand, dans le canal concerné, il y a activation de l'une des autres transformations (par ex. TRANSMIT, TRAANG, TRAORI).



Fonction

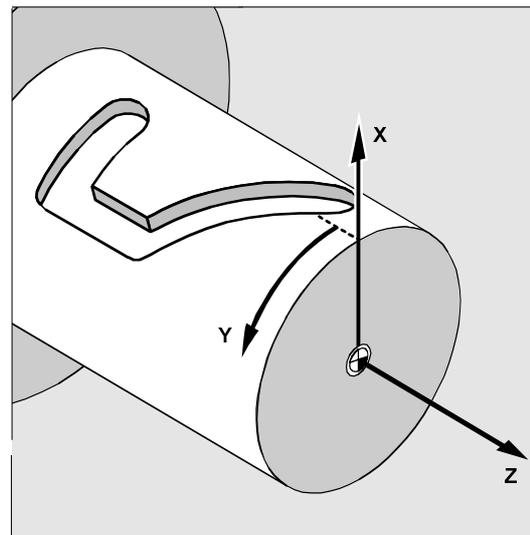
Transformation de surface latérale de cylindre
TRACYL

La transformation de la surface latérale du cylindre TRACYL offre les possibilités suivantes :

Usinage de

- rainures longitudinales sur des corps cylindriques,
- rainures transversales sur des corps cylindriques,
- rainures de forme quelconque sur des corps cylindriques.

La forme des rainures est programmée sur la surface développée du cylindre.



Système de coordonnées pièce

840 D
NCU 571840 D
NCU 572
NCU 573

FM-NC



810D

Il existe deux types de transformation de surface latérale de cylindre :

- sans correction des flancs de rainure
- avec correction des flancs de rainure

Sans correction des flancs de rainure :
la commande transforme les déplacements programmés dans le système de coordonnées cartésiennes en déplacements des axes réels de la machine :

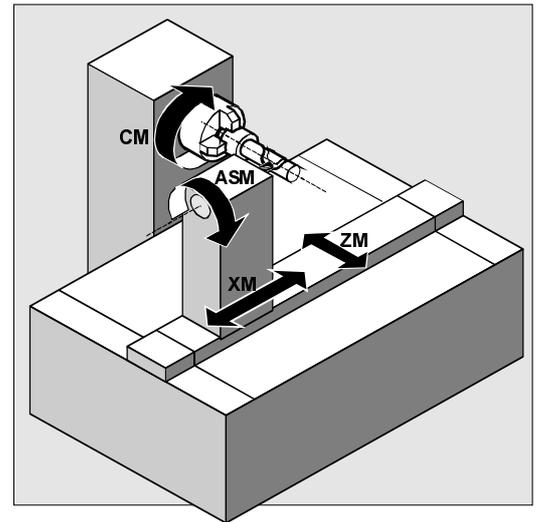
- axe rotatif
- axe de pénétration perpendiculaire à l'axe de rotation
- axe longitudinal parallèle à l'axe de rotation

Les axes linéaires sont orthogonaux. L'axe de pénétration coupe l'axe rotatif.

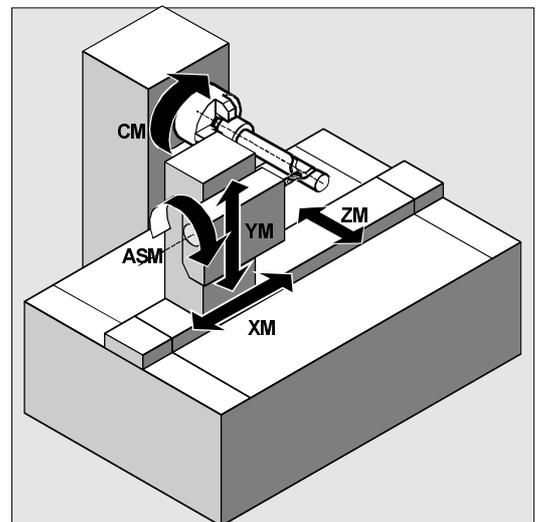
Avec une correction des flancs de rainure :
même cinématique que ci-dessus, mais en plus
– axe transversal perpendiculaire à l'axe de pénétration

Les axes linéaires sont orthogonaux.

Le pilotage de vitesse tient compte des limitations définies pour les rotations.



Système de coordonnées machine



Système de coordonnées machine

7.5 TRACYL, TRAFOOF, OFFN



840 D
NCU 571



840 D
NCU 572



FM-NC

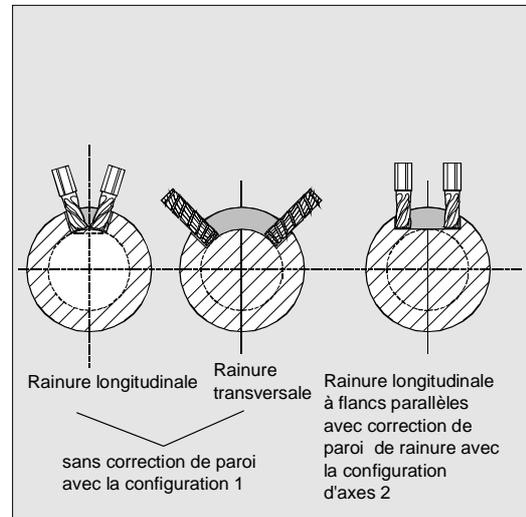


810D
NCU 573

Section de rainure

Avec la configuration d'axes 1, les flancs des rainures orientées dans le sens de l'axe rotatif sont parallèles uniquement si la largeur de la rainure correspond exactement au diamètre de l'outil.

Les flancs des rainures orientées parallèlement à la circonférence (rainures transversales) ne sont pas parallèles au début et à la fin.



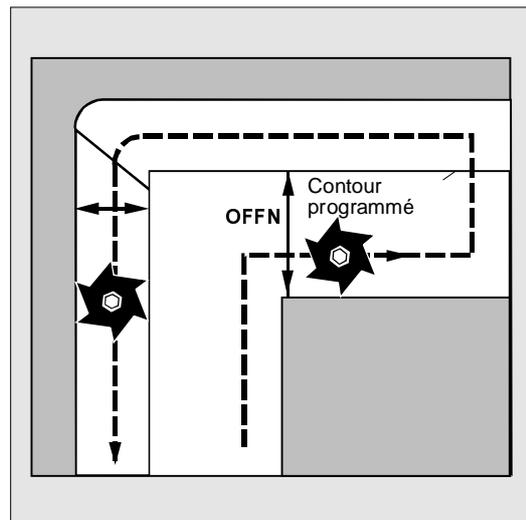
Décalage normal au contour OFFN

Vous pouvez utiliser l'instruction OFFN pour réaliser des trajectoires équidistantes, par ex. pour la semi-finition et l'ébauche de contours de type spline.

En version standard (préréglage), OFFN est inactive (OFFN=0).

OFFN accepte des valeurs du type REAL, positives ou négatives.

L'instruction est modale et agit de façon additive au rayon de l'outil, lorsque la correction de rayon est activée (G41, G42).



Si la transformation cinématique TRACYL est active, OFFN a la signification suivante : **Distance entre le flanc de la rainure et le contour de référence programmé.**

840 D
NCU 571840 D
NCU 572
NCU 573

FM-NC



810D



La transformation peut être activée et désactivée via le programme pièce ou en mode MDA.

Points à observer pour l'activation

- Il n'y a pas insertion d'un bloc intermédiaire de déplacement (chanfrein/rayon).
- Une séquence de blocs splines doit être clôturée
- La correction de rayon d'outil doit être désactivée
- Le frame actif avant TRACYL est désactivé par la commande.
- Une limitation active de la zone de travail est désactivée par la commande pour les axes impliqués dans la transformation
- Le contournage et l'arrondissement du contour sont interrompus
- Les décalages DRF dans les axes impliqués dans la transformation ne doivent pas subir de modification entre le prétraitement des blocs et leur exécution.
- Dans le cas d'une transformation de la surface latérale du cylindre avec correction des flancs de rainure, l'axe utilisé pour la correction devrait être positionné à zéro ($y=0$), pour que la rainure puisse être fraisée de façon centrée par rapport à sa ligne médiane, telle qu'elle a été programmée.
- Quand la transformation est activée et en fonctionnement normal, différents frames réglables doivent être activés, car ces frames transforment le SCP en système de coordonnées de base SCB.

Points à observer pour la désactivation

Pour la désactivation, observer les mêmes points que pour l'activation.

Changement d'outil

Un changement d'outil n'est autorisé que si la correction de rayon d'outil est désactivée.

Un changement de la correction de longueur d'outil et une activation/désactivation de la correction de rayon d'outil ne doivent pas être programmées dans le même bloc.

7.5 TRACYL, TRAFOOF, OFFN



840 D
NCU 571



840 D
NCU 572
NCU 573



FM-NC



810D

Frame

Toutes les instructions qui se réfèrent uniquement au système de coordonnées de base sont autorisées (frame, correction de rayon d'outil). Un changement de frame avec G91 (cotes relatives) ne sera cependant pas traité à part, contrairement à ce qui se passe lorsque la transformation est inactive. L'incrément à parcourir est traité dans le système de coordonnées pièce du nouveau frame, quel que soit le frame qui était actif dans le bloc précédent.

Axe rotatif

L'axe rotatif ne peut pas être programmé, étant donné qu'il est affecté comme axe géométrique et qu'il n'est donc pas programmable directement comme axe de canal.

Utilisation des axes

Les axes suivants ne peuvent pas être utilisés comme axes de positionnement ou comme axes d'oscillation :

- l'axe géométrique dans le sens de la circonférence de l'enveloppe du cylindre (axe Y).
- l'axe linéaire supplémentaire en cas de correction des flancs de rainure (axe Z).

Exclusions

Les axes impliqués dans la transformation ne peuvent pas être utilisés

- comme axes Preset (alarme)
- pour l'accostage d'un point fixe (alarme)
- pour la prise de référence (alarme)

840 D
NCU 571840 D
NCU 572
NCU 573

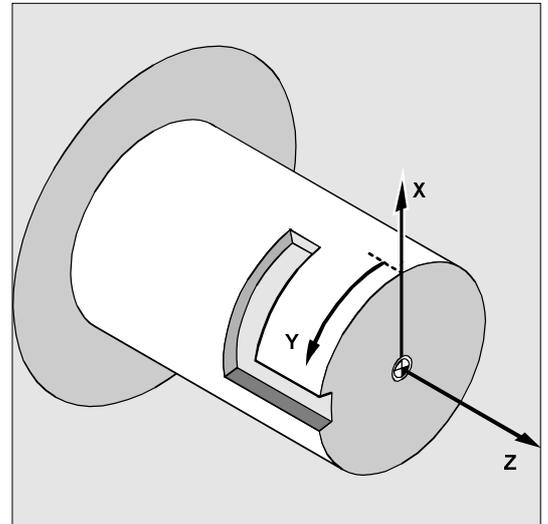
FM-NC



810D



Exemple de programmation



N10 T1 D1 G54 G90 F5000 G94

sélection de l'outil, compensation de la prise de pièce

N20 SPOS=0

accostage de la position de départ

N30 G0 X25 Y0 Z105 CC=200

N40 TRACYL (40)

activation de la transformation de surface latérale de cylindre

N50 G19

choix du plan

Réalisation d'une rainure en forme de crochet :

N60 G1 X20

pénétration de l'outil suivant la profondeur de la rainure

N70 OFFN=12

définition de la distance entre le flanc de rainure et le centre de rainure : 12 mm

N80 G1 Z100 G42

accostage du flanc droit de la rainure

N90 G1 Z50

section de rainure parallèle à l'axe du cylindre

N100 G1 Y10

section de rainure parallèle à la circonférence du cylindre

N110 OFFN=4 G42

accostage du flanc gauche de la rainure; définition de la distance entre flanc et centre de rainure : 4 mm

N120 G1 Y70

section de rain. parall. à la circonf. du cylind.

N130 G1 Z100

section de rainure parallèle à l'axe du cylindre

N140 G1 Z105 G40

retrait du flanc de rainure

N150 G1 X25

dégagement

N160 TRAFOOF

N170 G0 X25 Y0 Z105 CC=200

accostage de la position de départ

N180 M30

7.6 TRAANG, TRAFOOF



840 D
NCU 572
NCU 573

7.6 TRAANG, TRAFOOF



Programmation

TRAANG (α) ou TRAANG (α, n)
TRAFOOF



TRAANG (α)	Activation de la première transformation convenue d'axe oblique
TRAANG (α, n)	Activation de la n-ième transformation convenue d'axe oblique. "n" est limité à deux. TRAANG($\alpha, 1$) correspond à TRAANG(α).
α	angle de l'axe oblique
TRAFOOF	Désactivation de la transformation



Si α (angle) n'est pas indiqué ou vaut 0, la transformation est activée avec le paramétrage de la sélection précédente. Lors de la première activation, les valeurs spécifiées dans les paramètres machine s'appliquent.

Une transformation active TRAANG est également désactivée quand, dans le canal concerné, il y a activation de l'une des autres transformations (par ex. TRACYL, TRANSMIT, TRAORI).

840 D
NCU 571840 D
NCU 572
NCU 573

FM-NC



810D



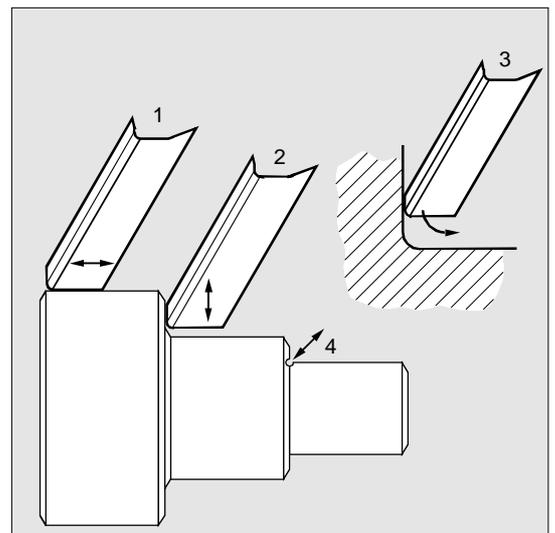
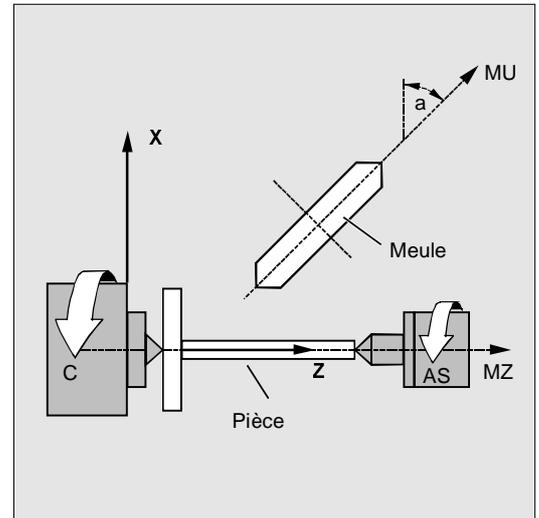
Fonction

La fonction axe oblique est destinée à l'utilisation en rectification et offre les possibilités suivantes :

- Usinage avec un axe de pénétration oblique
- Un système de coordonnées cartésiennes peut être utilisé pour la programmation.
- La commande transforme les déplacements programmés dans le système de coordonnées cartésiennes en déplacements des axes réels de la machine (cas standard) : axe de pénétration oblique.

Les usinages possibles sont les suivants :

1. Cylindrage
2. Dressage
3. Profilage
4. Rectification en plongée oblique





840 D
NCU 572
NCU 573

Les réglages suivants sont définis par paramètres machine :

- l'angle entre un axe de la machine et l'axe oblique
- la position de l'origine pièce par rapport à l'origine du système de coordonnées convenue pour la fonction "axe oblique"
- la réserve de vitesse à disposition sur l'axe parallèle pour le déplacement compensateur
- la réserve d'accélération à disposition sur l'axe parallèle pour le déplacement compensateur

Configuration des axes

Pour pouvoir programmer dans le système de coordonnées cartésiennes, il est nécessaire de communiquer à la commande les relations qui existent entre ce système de coordonnées et les axes réels de la machine (MU, MZ) :

- Dénomination des axes géométriques
- Affectation des axes géométriques aux axes de canal
 - cas général (axe oblique inactif)
 - axe oblique actif
- Affectation des axes de canal aux numéros d'axe de la machine
- Identification des broches
- Attribution de noms aux axes de la machine

A l'exception de "axe oblique actif", la manière de procéder est identique à celle utilisée pour la configuration d'axes normale.

840 D
NCU 571840 D
NCU 572
NCU 573

FM-NC



810D

Points à observer pour l'activation et la désactivation

La transformation peut être activée et désactivée par le biais du programme pièce ou en mode MDA.

- Il n'y a pas insertion d'un bloc intermédiaire de déplacement (chanfrein/rayon).
- Une séquence de blocs splines doit être clôturée
- La correction de rayon d'outil doit être désactivée
- Le frame courant est désactivé par la commande
- Une limitation active de la zone de travail est désactivée par la commande pour les axes impliqués dans la transformation
- Une correction de longueur d'outil activée est reprise dans la transformation par la commande
- Le contournage et l'arrondissement du contour sont interrompus
- Les décalages DRF dans les axes impliqués dans la transformation ne doivent pas subir de modification entre le prétraitement des blocs et leur exécution.
- Tous les axes indiqués dans les paramètres machine doivent être synchronisés au niveau des blocs
- Quand la transformation est activée et en fonctionnement normal, différents frames réglables doivent être activés, car ces frames transforment le SCP en système de coordonnées de base SCB.

Changement d'outil

Un changement d'outil n'est autorisé que si la correction de rayon d'outil est désactivée.

Un changement de la correction de longueur d'outil et une activation/désactivation de la correction de rayon d'outil ne doivent pas être programmées dans le même bloc.

7.6 TRAANG, TRAFOOF



840 D
NCU 572
NCU 573

Frame

Toutes les instructions qui se réfèrent au système de coordonnées pièce sont autorisées. Un changement de frame avec G91 (cotes relatives) ne sera cependant pas traité à part, contrairement à ce qui se passe lorsque la transformation est inactive. L'incrément à parcourir est traité dans le système de coordonnées pièce du nouveau frame, quel que soit le frame qui était actif dans le bloc précédent.

Exclusions

Les axes impliqués dans la transformation ne peuvent pas être utilisés

- comme axes Preset (alarme),
- pour l'accostage d'un point fixe (alarme)
- pour la prise de référence (alarme).

840 D
NCU 571840 D
NCU 572
NCU 573

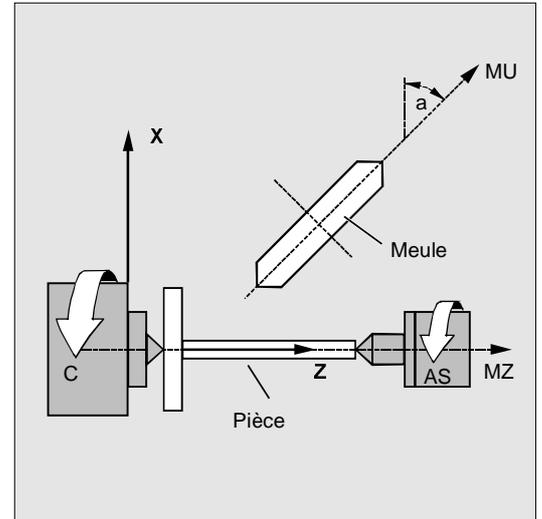
FM-NC



810D



Exemple de programmation



```
N10 G0 G90 Z0 MU=10 G54 F5000 ->
-> G18 G64 T1 D1
```

sélection de l'outil, compensation de la prise de pièce, sélection du plan

```
N20 TRAANG(45)
```

active la transformation de l'axe oblique

```
N30 G0 Z10 X5
```

accostage de la position de départ

```
N40 WAITP(Z)
```

libération des axes pour l'oscillation

```
N50 OSP[Z]=10 OSP2[Z]=5 OST1[Z]=-2 ->
-> OST2[Z]=-2 FA[Z]=5000
```

oscillation jusqu'à la cote finale
(Oscillation : voir chap. 9)

```
N60 OS[Z]=1
```

```
N70 POS[X]=4.5 FA[X]=50
```

```
N80 OS[Z]=0
```

```
N90 WAITP(Z)
```

libération des axes d'oscillation comme axes de positionnement

```
N100 TRAF00F
```

désactive la transformation

```
N110 G0 Z10 MU=10
```

dégagement

```
N120 M30
```

-> doit être programmé dans un bloc

Corrections d'outil

8.1 Mémoire de correcteurs	8-240
8.2 Liste des types d'outils	8-242
8.3 Instructions de gestion d'outils	8-243
8.4 Correction d'outil en ligne, PUTFTOCF, PUTFTOC, FTOCON, FTOCOF	8-246
8.5 Activation des corrections d'outil 3D	8-252
8.6 Orientation de l'outil.....	8-259

8.1 Mémoire de correcteurs



840 D
NCU 571



840 D
NCU 572
NCU 573



FM-NC



810D

8.1 Mémoire de correcteurs

Structure de la mémoire des correcteurs

Chaque champ de données peut être appelé à l'aide d'un numéro T et d'un numéro D et contient, outre les données géométriques de l'outil, d'autres indications tel que le type d'outil.

Les grandeurs géométriques (par.ex. longueur 1 ou rayon) sont constituées de plusieurs composantes. Ces dernières sont additionnées pour donner une grandeur résultante (par.ex. longueur totale 1, rayon total) qui est alors pris en compte pour la correction.

La valeur zéro est affectée aux correcteurs inutiles.



Les différentes valeurs de la mémoire de correction P1 à P25 peuvent être lues et écrites dans le programme à l'aide de variables système.

840 D
NCU 571840 D
NCU 572
NCU 573

FM-NC



810D

Paramètres d'outil Numéro de paramètre (DP)	Signification	Observations
\$TC_DP 1	Type d'outil	voir liste ci-dessous
\$TC_DP 2	Position du tranchant	uniquement outils de tournage
Géométrie		
Correction de longueur		
\$TC_DP 3	Longueur 1	Prise en compte selon type
\$TC_DP 4	Longueur 2	et plan
\$TC_DP 5	Longueur 3	
Rayon		
\$TC_DP 6	Rayon	
\$TC_DP 7	Largeur de rainure b (scie à rainurer), rayon d'arrondi (outils de fraisage)	
\$TC_DP 8	Dépassement k	uniquement pour scie à rainurer
\$TC_DP 11	Angle pour outils de fraisage coniques	
Usure		
Correction de longueur et de rayon		
\$TC_DP 12	Longueur 1	
\$TC_DP 13	Longueur 2	
\$TC_DP 14	Longueur 3	
\$TC_DP 15	Rayon	
\$TC_DP 16	Largeur de rainure b (scie à rainurer), rayon d'arrondi (outils de fraisage)	
\$TC_DP 17	Dépassement k	uniquement pour scie à rainurer
\$TC_DP 20	Angle pour outils de fraisage coniques	
Cote de base / adapdateur		
Corrections de longueur		
\$TC_DP 21	Longueur 1	
\$TC_DP 22	Longueur 2	
\$TC_DP 23	Longueur 3	
Technologie		
\$TC_DP 24	Dépouille	pour outils de tournage

8.2 Liste des types d'outils



840 D
NCU 571



840 D
NCU 572
NCU 573



FM-NC



810D

8.2 Liste des types d'outils



Remarques

Vous trouverez la liste des outils dans manuel de programmation "Notions de base", chapitre "Corrections d'outil".

Affectation des paramètres spécifiques aux outils

D'autres paramètres spécifiques outils peuvent être définis et affectés par l'utilisateur par le biais des paramètres machine.

Paramètre	Signification	Type de donnée
Paramètres spécifiques aux outils		
\$TC_TPG1	N° de broche	Integer
\$TC_TPG2	Règle de concaténation Les paramètres pour le côté gauche et le côté droit de la meule sont égalisés automatiquement.	Integer
\$TC_TPG3	Rayon minimal de la meule	Real
\$TC_TPG4	Largeur minimale de la meule	Real
\$TC_TPG5	Largeur courante de la meule	Real
\$TC_TPG6	Vitesse de rotation maximale	Real
\$TC_TPG7	Vitesse périphérique maximale	Real
\$TC_TPG8	Angle de la meule inclinée	Real
\$TC_TPG9	N° du paramètre pour calcul du rayon	Integer

840 D
NCU 571840 D
NCU 572
NCU 573

810D

8.3 Instructions de gestion d'outils



Signification des instructions

T="OUTIL"	Sélectionner un outil par son nom
NEWT ("OUTIL", N°_FRERE)	Créer un nouvel outil, numéro "frère" optionnel
DELT ("OUTIL", N°_FRERE)	Effacer un outil, numéro "frère" optionnel
GETT ("OUTIL", N°_FRERE)	Définir le n° T
SETPIECE (x, y)	Spécifier le nombre de pièces
GETSELT (x)	Lire le numéro de l'outil présélectionné (n° T)
"OUTIL"	Descripteur d'outil
N°_FRERE	Nombre de pièces
x	Numéro de broche, indication optionnelle



En cas d'utilisation de la gestion des outils, les outils peuvent être créés et appelés avec leurs noms, par ex. T = "FORET" ou T = "123".



Fonction NEWT

La fonction NEWT permet de créer un nouvel outil avec un nom dans le programme CN. Comme paramètre en retour, la fonction fournit le numéro T créé automatiquement, avec lequel l'outil pourra ensuite être adressé.

```
Param. en retour = NEWT ("OUTIL",  
N°_FRERE)
```

Si le numéro "frère" n'est pas indiqué, celui-ci est créé par la gestion d'outils.

Exemple :

```
DEF INT N°_FRERE  
DEF INT N°_T  
N°_FRERE = 7  
N°_T=NEWT("FORET", N°_FRERE)
```

Créer le nouvel outil "FORET" avec le n° "frère" 7. Le n° T est enregistré dans N°_T.

Fonction DELT

La fonction DELT permet d'effacer un outil sans tenir compte d'un numéro T.

```
DELT ("OUTIL", N°_FRERE)
```

8.3 Instructions de gestion d'outils



840 D
NCU 571



840 D
NCU 572
NCU 573



810D

Fonction GETT

La fonction GETT fournit, pour un outil qui n'est connu que par son nom, le numéro T nécessaire pour spécifier les paramètres de cet outil .

```
Paramètre en retour=GETT
("OUTIL", N°_FRERE)
```

S'il existe plusieurs outils avec le même descripteur, c'est le numéro T du premier de ces outils possibles qui restitué.

Retour = -1 : aucun outil ne peut être mis en correspondance avec la désignation de l'outil ou le numéro "frère".

Exemples :

```
T= "FORET"
```

```
R10=GETT("FORET", N°_FRERE)
```

N° T déterminé pour FORET avec le numéro "frère" = N°_FRERE

Le "FORET" doit, au préalable, être déclaré avec NEWT ou \$TC_TP1[].

```
$TC_DP1[GETT("FORET", N°_FRERE),1]=100
```

Ecriture d'un paramètre d'outil avec le nom de l'outil.

Fonction SETPIECE

Cette fonction sert à mettre à jour les données de surveillance du nombre de pièces.

La fonction concerne tous les outils qui ont été mis en place dans la broche indiquée depuis la dernière activation de SETPIECE.

SETPIECE(x,y)

x	Nombre de pièces usinées
---	--------------------------

y	y numéro de la broche, 0 = broche maître (préréglage)
---	---

840 D
NCU 571840 D
NCU 572
NCU 573

810D

Fonction GETSELT

Cette fonction fournit le numéro T de l'outil présélectionné pour la broche .

Ceci permet d'accéder aux données de correction d'outil avant même l'appel de M6 et de réaliser un peu plus tôt la synchronisation avec l'exécution des blocs.

Exemple de changement d'outil avec gestion d'outils

- T1 Présélection d'outil, autrement dit le magasin d'outils peut préparer le prochain outil parallèlement à l'usinage.
- M6 Mise en place d'un outil présélectionné (selon le préréglage effectué dans les paramètres machine, il est aussi possible de programmer sans M6).

Exemple :

T1 M6	Mettre en place l'outil 1
D1	Sélection de la correction de longueur d'outil
G1 X10 ...	Usinage avec T1
T= "FORET"	Présélection de l'outil "Foret"
D2 Y20 ...	Changement d'arête tranchante T1
X10 ...	Usinage avec T1
M6	Mise en place de l'outil "Foret"
SETPIECE(4)	Nombre de pièces usinées
D1 G1 X10 ...	Usinage avec l'outil "Foret"



Vous trouverez la liste complète de toutes les variables utilisées par la gestion des outils au chapitre 14 (liste des variables système).

8.4 Correction d'outil en ligne



840 D
NCU 572
NCU 573

8.4 Correction d'outil en ligne PUTFTOCF, PUTFTOC, FTOCON, FTOCOF



Programmation

FCTDEF(n° polynôme, LLimit, ULimit, a₀, a₁, a₂, a₃)
 PUTFTOCF(n° polynôme, valeur de référence, longueur_{1_2_3}, canal, broche)
 PUTFTOC(valeur, longueur_{1_2_3}, canal, broche)
 FTOCON
 FTOCOF



Signification des instructions

PUTFTOCF	Ecriture en continu de la correction d'outil en ligne
FCTDEF	Paramétrage de la fonction PUTFTOCF
PUTFTOC	Ecriture discrète de la correction d'outil en ligne
FTOCON	Activation de la correction d'outil en ligne
FTOCOF	Désactivation de la correction d'outil en ligne



Signification des paramètres

N° du polynôme	Valeur 1 à 3 : 3 polynômes au max. sont admis simultanément ; polynôme jusqu'au troisième degré
Valeur de référence	Valeur de référence dont est dérivée la correction
Longueur _{1_2_3}	Paramètre d'usure dans lequel est additionnée la valeur de correction d'outil
Canal	N° du canal dans lequel agit la correction d'outil ; le canal est indiqué uniquement s'il ne s'agit pas du propre canal
Broche	N° de la broche sur laquelle agit la correction d'outil en ligne ; l'indication est nécessaire uniquement pour les meules non actives.
LLimit	Valeur limite supérieure
ULimit	Valeur limite inférieure
a ₀ , a ₁ , a ₂ , a ₃	Coefficients de la fonction polynomiale
Valeur	Valeur additionnée au paramètre d'usure



840 D
NCU 572
NCU 573

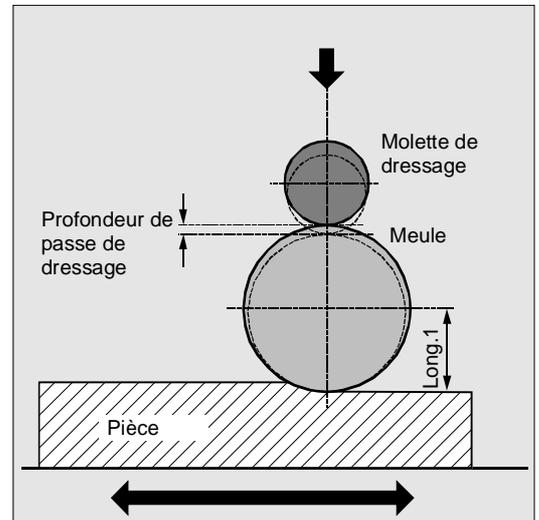


Fonction

Avec cette fonction, les corrections d'outil qui résultent de l'usinage sont prises en compte immédiatement par le biais d'une correction de longueur d'outil en ligne (par ex. dressage continu : la meule est dressée parallèlement à l'usinage). La correction de longueur d'outil peut être modifiée à partir du canal d'usinage ou d'un canal parallèle (canal de dressage).



La correction d'outil en ligne s'applique uniquement aux outils de rectification.



Généralités sur la correction d'outil en ligne

Selon le moment de l'opération de dressage, on utilise des fonctions différentes pour écrire la correction d'outil en ligne :

- Ecriture en continu bloc par bloc : PUTFTOCF
- Ecriture en continu à effet modal : ID=1 DO FTOC (voir le chapitre sur les actions synchrones)
- Ecriture discrète : PUTFTOC

Dans le cas de l'écriture en continu (suivant la période d'appel de l'interpolateur), après l'activation de la fonction de traitement, chaque modification est additionnée dans la mémoire des usures pour éviter de sauter des valeurs de consigne.

Dans tous les cas :

La correction d'outil en ligne peut agir dans chaque canal, pour chaque broche, pour la longueur 1, 2 ou 3 des paramètres d'usure.

Les longueurs sont mises en correspondance avec les axes géométriques à l'aide du plan courant.

La mise en correspondance entre la broche et l'outil se fait par le biais des paramètres d'outil de GWPERSON ou TMON, dans la mesure où il ne s'agit pas de la meule active (voyez le manuel de programmation "Notions de base").

8.4 Correction d'outil en ligne



840 D
NCU 572
NCU 573

La correction agit toujours sur le paramètre d'usure pour le côté courant de la meule ou le côté gauche quand les outils ne sont pas actifs.

Si la correction est identique pour plusieurs côtés de la meule, on applique une règle de concaténation (voyez le manuel de programmation "Notions de base") pour que les valeurs soient reprises automatique pour le côté suivant de la meule.

Si des corrections en ligne sont prescrites pour un canal d'usinage, les valeurs d'usure pour l'outil courant dans ce canal ne doivent pas être modifiées à partir du programme d'usinage ou au tableau de commande.

La correction d'outil en ligne est également prise en compte dans la vitesse périphérique constante de meule (VPM), dans la surveillance d'outil TMON et dans la rectification sans centre CLGON.

Procédure

PUTFTOCF = Ecriture en continu

L'opération de dressage s'effectue parallèlement à l'usinage :

dressage sur toute la largeur de la meule avec une molette de dressage ou un diamant.

L'usinage et le dressage peuvent s'effectuer dans des canaux différents. Quand aucun canal n'est programmé, la correction agit dans le canal actif.

`PUTFTOCF(n° polynôme, valeur de référence, longueur 1_2_3, canal, broche)`

La correction d'outil est modifiée continuellement dans le canal d'usinage suivant une fonction poly-nomiale du premier, du second ou du troisième degré qui doit être définie préalablement avec FCTDEF.

La correction est déduite de la variable "valeur de référence", par exemple une valeur réelle changeante.

Si aucun numéro de broche n'est programmé, l'outil actif, c'est-à-dire celui actuellement en œuvre, subit la correction.



840 D
NCU 572
NCU 573

Paramétrage de la fonction FCTDEF

Le paramétrage s'effectue dans un bloc spécifique :

```
FCTDEF(n° polynôme, LLimit, ULimit, a0, a1, a2, a3)
```

Le polynôme peut être du premier, du second ou du troisième degré.

Limit désigne les valeurs limites (L limit = limite inférieure, U limit = limite supérieure).

Exemple :

droite ($y = a_0 + a_1x$) avec une pente de 1

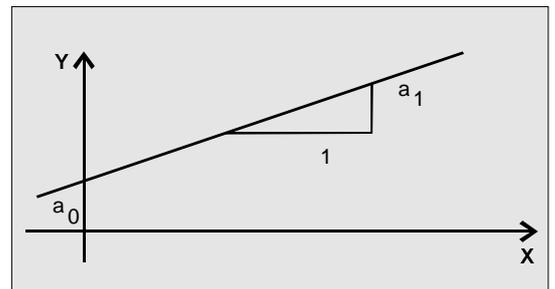
```
FCTDEF(1, -1000, 1000, -$AA_IW[X], 1)
```

Écriture discrète de la correction d'outil en ligne : PUTFTOC

Cette instruction sert à écrire une valeur de correction **une seule fois**. La correction agit immédiatement dans le canal de destination.

Application de PUTFTOC :

La meule est dressée dans un canal parallèle, mais pas parallèlement à l'usinage.



```
PUTFTOC(valeur, longueur1_2_3, canal, broche)
```

La correction d'outil en ligne pour la longueur donnée 1, 2 ou 3 est modifiée en fonction de la valeur indiquée, en l'occurrence la valeur indiquée est ajoutée au paramètre d'usure.

Prise en compte de la correction d'outil en ligne : FTOCON, FTOCOF

Le canal de destination ne peut recevoir des corrections d'outil en ligne que si FTOCON est active.

- FTOCON doit être écrite dans le canal dans lequel la correction doit agir.
Avec FTOCOF, la correction n'est plus appliquée, mais la valeur complète écrite avec PUTFTOC est corrigée dans les données de correction spécifiques au tranchant de la meule.
- FTOCOF correspond toujours à la position RESET.
- PUTFTOC n'a jamais un effet modal, autrement dit elle agit seulement dans le bloc de déplacement qui suit.
- Avec FTOC, on peut aussi choisir la correction d'outil en ligne pour qu'elle agisse de façon modale. Pour plus d'information à ce sujet, consultez le chapitre "Actions synchrones au déplacement".

8.4 Correction d'outil en ligne



840 D
NCU 572
NCU 573



Exemple de programmation

Problème à résoudre

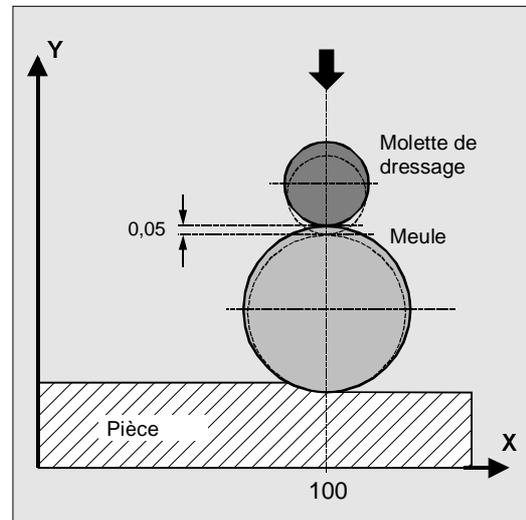
Sur une rectifieuse plane dont les caractéristiques sont précisées ci-après, la meule doit être dressée de 0,05 en X100 après le début de la rectification. La profondeur de passe de dressage doit agir avec une écriture continue de la correction d'outil en ligne.

Y : axe de pénétration pour la meule

V : axe de pénétration pour la molette de dressage

Usinage : canal 1 avec les axes X, Z, Y

Dressage : canal 2 avec l'axe V



Programme d'usinage dans le canal 1 :

```
%_N_USIN _MPF
```

```
...
```

```
N110 G1 G18 F10 G90
```

Préréglage

```
N120 T1 D1
```

Sélection de l'outil courant

```
N130 S100 M3 X100
```

Mise en marche de la broche, accostage de la position initiale

```
N140 INIT (2, "DRESSAGE", "S")
```

Sélection du programme de dressage dans le canal 2

```
N150 START (2)
```

Lancement du programme de dressage dans le canal 2

```
N160 X200
```

Accostage de la position de destination

```
N170 FTOCON
```

Activation de la correction en ligne

```
N... G1 X100
```

Autres opérations d'usinage

```
N...M30
```

Programme de dressage dans le canal 2 :

```
%_N_DRESSAG_MPF
```

```
...
```

```
N40 FCTDEF (1, -1000, 1000, -$AA_IW[V], 1)
```

Définition de la fonction : linéaire

```
N50 PUTFTOCF (1, $AA_IW[V], 3, 1)
```

Écriture continue de la correction d'outil en ligne : la longueur 3 de la meule courante est corrigée dans le canal 1 en fonction du déplacement de l'axe V.

```
N60 V-0.05 G1 F0.01 G91
```

Mouvement de pénétration pour dressage, PUTFTOCF agit uniquement dans ce bloc

```
...
```

```
N...M30
```



840 D
NCU 572
NCU 573

Programme de dressage modal :

```
%_N_DRESSAG_MPF
```

```
FCTDEF(1, -1000, 1000, -$AA_IW[V], 1)
```

Définir la fonction :

```
ID=1 DO FTOC(1, $AA_IW[V], 3, 1)
```

Activer la correction d'outil en ligne :
la valeur réelle de l'axe V est valeur
d'entrée pour le polynôme 1; le résultat
est additionné dans le canal 1 comme
valeur de correction de la longueur 3 de
la meule active.

```
WAITM(1, 1, 2)
```

Synchronisation avec canal d'usinage

```
G1 V-0.05 F0.01, G91
```

Mouvement de pénétration pour le
dressage

```
G1 V-0.05 F0.02
```

```
...
```

```
CANCEL(1)
```

Désactiver la correction en ligne

```
...
```

8.5 Activation des corrections d'outil 3D



840 D
NCU 572
NCU 573

8.5 Activation des corrections d'outil 3D



Signification

CUT3DC	Activation de la correction de rayon 3D pour le fraisage périphérique
CUT3DFS	Correction d'outil 3D pour le fraisage en bout avec orientation d'outil constante. L'orientation de l'outil est définie par G17 - G19 et n'est pas influencée par les frames.
CUT3DFF	Correction d'outil 3D pour le fraisage en bout avec orientation d'outil constante. L'orientation de l'outil est définie par G17 - G19 et, le cas échéant, par la composante rotationnelle d'un frame.
CUT3DF	Correction d'outil 3D pour le fraisage en bout avec modification de l'orientation d'outil (uniquement si la transformation 5 axes est activée).
G40 X Y Z	Pour la désactivation : bloc linéaire G0/G1 avec axes géométriques
ISD=valeur	Profondeur de pénétration



Les instructions sont à effet modal et figurent dans le même groupe que CUT2D et CUT2DF.

La désactivation n'a lieu que lors du prochain déplacement dans le plan courant. Ceci est toujours valable pour G40 et est indépendant de l'instruction CUT.



Fonction

L'orientation variable de l'outil est prise en compte dans la correction du rayon des outils cylindriques.

Les instructions pour la sélection de la correction de rayon d'outil 3D sont les mêmes que pour la correction de rayon d'outil 2D. La correction à gauche/droite dans le sens de déplacement est indiquée avec G41/G42. L'accostage se fait toujours suivant NORM.



840 D
NCU 572
NCU 573



Exemple

N10 A0 B0 X0 Y0 Z0 F5000	
N20 T1 D1	Appel de l'outil, appel des correcteurs d'outil
N30 TRAORI(1)	Activation de la transformation
N40 CUT3DC	Activation de la correction du rayon d'outil 3D
N50 G42 X10 Y10	Activation de la correction du rayon d'outil
N60 X60	
N70 ...	



Remarques complémentaires

Des blocs intermédiaires sont permis lorsque la correction du rayon d'outil 3D est active. Les définitions sont les mêmes que pour la correction du rayon d'outil 2 1/2D.

La correction de rayon d'outil 3D agit uniquement quand la transformation 5 axes est active.

Un bloc de raccordement par interpolation circulaire est toujours inséré aux angles saillants. G450/G451 sont sans signification.

L'instruction DISC n'est pas prise en compte.

8.5 Activation des corrections d'outil 3D



840 D
NCU 572
NCU 573

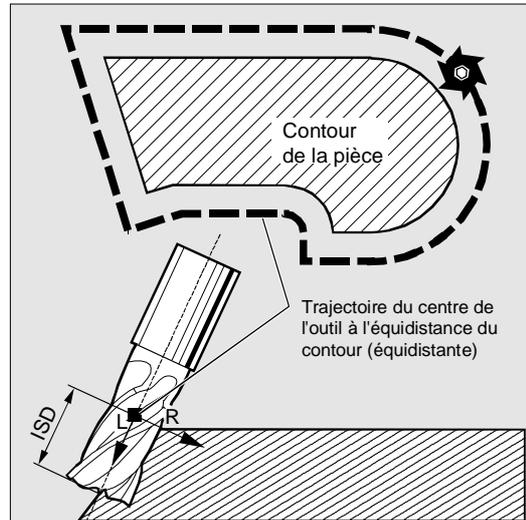
Différence entre corrections de rayon d'outil

2 1/2D et 3D

Dans le cas de la correction de rayon d'outil 3D, l'orientation de l'outil est variable.

Dans le cas de la correction de rayon d'outil 2 1/2D, on considère que l'outil a une orientation constante.

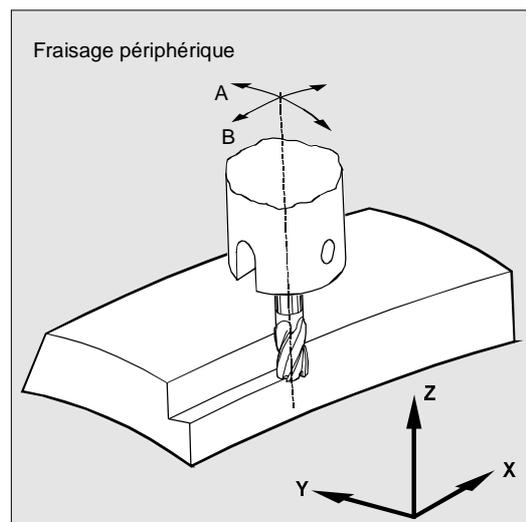
La correction de rayon d'outil 3D est également appelée correction 5D, vu que, dans ce cas, on dispose de cinq degrés de liberté pour la position de l'outil dans l'espace.



Fraisage périphérique

La variante de fraisage périphérique utilisée ici est réalisée par spécification d'un contour (ligne directrice) et de l'orientation correspondante. Dans ce type d'usinage, la forme de l'outil n'a aucune importance pour la trajectoire. Seul le rayon au point de contact de l'outil est décisif.

La fonction de correction de rayon d'outil 3D est limitée aux outils cylindriques.





840 D
NCU 572
NCU 573

Fraisage en bout

Pour ce type de fraisage 3D, on a besoin de la description des différentes trajectoires 3D qui balayent la surface de la pièce, ligne après ligne. Les calculs sont généralement réalisés à l'aide d'un système de FAO, en tenant compte de la forme et des dimensions de l'outil.

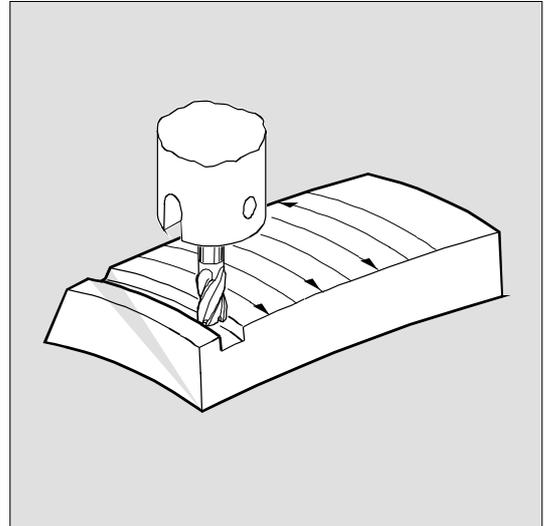
Le postprocesseur inscrit dans le programme pièce –outre les blocs CN –les orientations de l'outil (si la transformation 5 axes est activée) et le descripteur correspondant à la correction d'outil 3D désirée.

De cette façon, l'opérateur a la possibilité d'utiliser des outils un peu plus petits que ceux qui avaient servis au calcul des trajectoires CN.

Exemple :

Les blocs CN ont été déterminés pour une fraise de diamètre 10 mm.

Dans ce cas, un usinage avec une fraise de diamètre 9,9 mm serait également possible, la rugosité de la surface obtenue étant alors différente.



8.5 Activation des corrections d'outil 3D



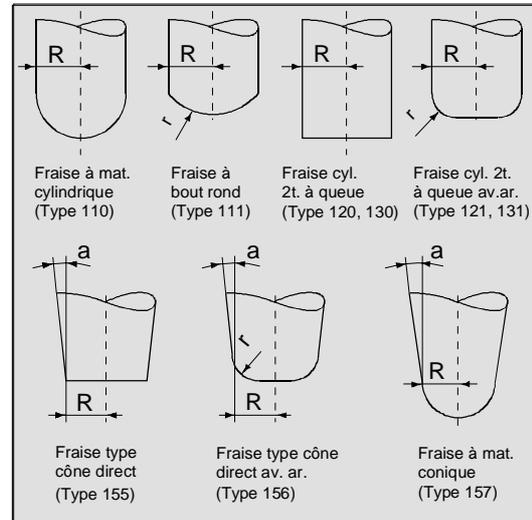
840 D
NCU 572
NCU 573

Formes des fraises, données d'outil

Le tableau ci-contre représente les différentes formes d'outils utilisables pour le fraisage en bout et les valeurs limites des données d'outil.

La forme de la queue des fraises n'est pas prise en compte ; les types d'outil 111 et 157 ont le même effet.

Si le programme CN indique un numéro de type d'outil qui ne figure pas dans le tableau, la commande utilise automatiquement le type d'outil 110, une fraise à bout rond. Une alarme est émise si les valeurs limites des données d'outil ne sont pas respectées.



Type de fraise	N°	R	r	a	
Fraise à bout rond	110		>0	X	X
Fraise à bout rond	111		>0	>R	X
Fraise cylindrique deux tailles à queue, fraise pour tête à renvoi d'angle	120, 130		>0	X	X
Fraise cylindrique deux tailles à queue, fraise pour tête à renvoi d'angle avec arrondi	121, 131		>r	>0	X
Fraise conique type cône direct	155		>0	X	>0
Fraise conique type cône direct avec arrondi	156		>rcosa	>0	>0
Fraise à matrices à bout conique	157		>0	X	>0

X=n'est pas évalué

Correction de longueur d'outil

La correction de longueur se réfère à la pointe de l'outil (point d'intersection axe longitudinal/surface frontale).



840 D
NCU 572
NCU 573

Correction d'outil 3D, changement d'outil

La déclaration d'un nouvel outil avec des dimensions différentes (R, r, a) ou une autre forme doit obligatoirement se faire en programmant G41 ou G42 (transition de G40 vers G41 ou G42 ou nouvelle programmation de G41 ou G42).

Toutes les autres données d'outil, la longueur par exemple, ne sont pas concernées par cette règle, autrement dit, il n'est pas nécessaire de reprogrammer G41 ou G42 si on passe à un outil de longueur différente.

Correction sur la trajectoire

Dans le cas du fraisage en bout, il faut considérer le cas où le point de contact change brusquement au niveau de la surface de l'outil, comme dans le cas de l'usinage d'une surface convexe avec un outil vertical (exemple ci-contre).

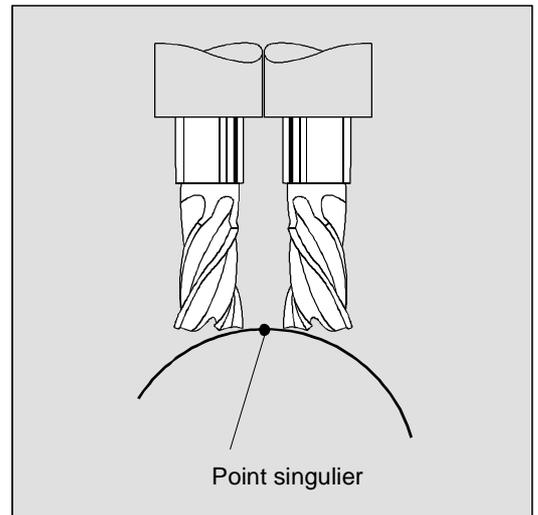
D'une façon générale, il est recommandé de choisir une forme et une orientation d'outil adaptées à la surface à réaliser.

C'est pourquoi l'exemple représenté ci-contre doit être considéré comme un cas limite.

Ce cas limite est surveillé par la commande qui détecte immédiatement toute modification brutale du point de contact en s'appuyant sur les angles formés par l'outil et les vecteurs normaux à la surface. A ces emplacements, la commande insère des blocs à interpolation linéaire, de sorte que le déplacement peut être effectué.

Pour le calcul de ces blocs à interpolation linéaire, des plages angulaires admissibles pour l'angle latéral figurent dans des paramètres machine.

Une alarme est émise si les plages angulaires admissibles figurant dans des paramètres machines sont dépassées.



8.5 Activation des corrections d'outil 3D



840 D
NCU 572
NCU 573

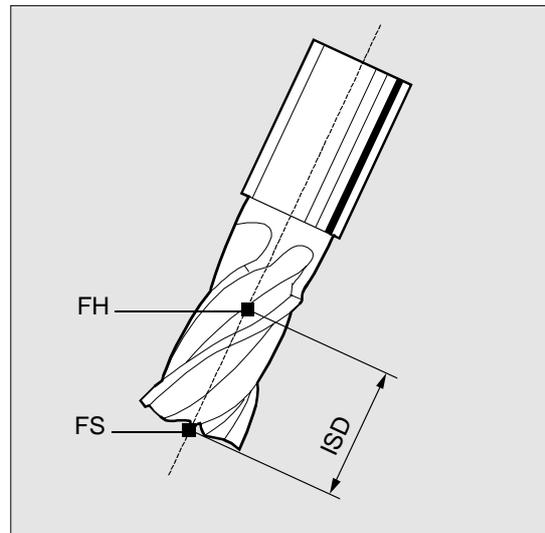
Courbure de la trajectoire

La courbure de la trajectoire ne fait pas l'objet d'une surveillance. C'est pourquoi il est également recommandé de n'utiliser que des outils avec lesquels un usinage sans violation du contour est garanti.

Profondeur de pénétration ISD

Avec l'instruction ISD (Insertion Depth) on programme la profondeur de pénétration de l'outil lors du fraisage périphérique. Cela permet de modifier la position du point d'usinage sur la surface latérale de l'outil.

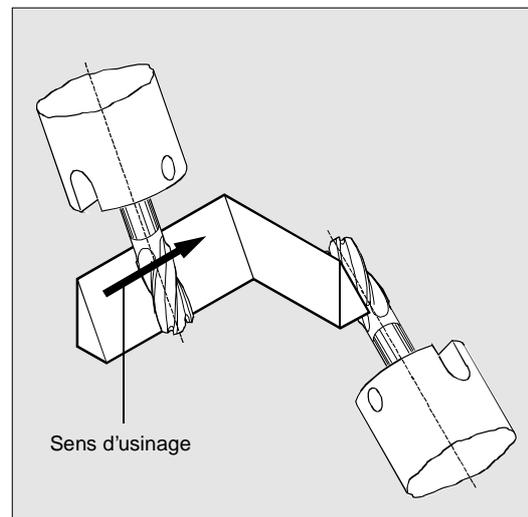
ISD indique la distance entre la pointe de la fraise (FS) et le point auxiliaire de la fraise (FH). Le point FH est obtenu en projetant le point d'usinage programmé sur l'axe de l'outil. ISD est prise en compte uniquement si la correction de rayon d'outil 3D est active.



Angles rentrants / Angles saillants

Les angles saillants et les angles rentrants sont traités séparément. L'appellation angle rentrant ou angle saillant dépend de l'orientation de l'outil.

En cas de changement d'orientation à un angle, il peut arriver que le type d'angle change pendant l'usinage. Dans ce cas, l'usinage est arrêté et un message d'erreur est émis.





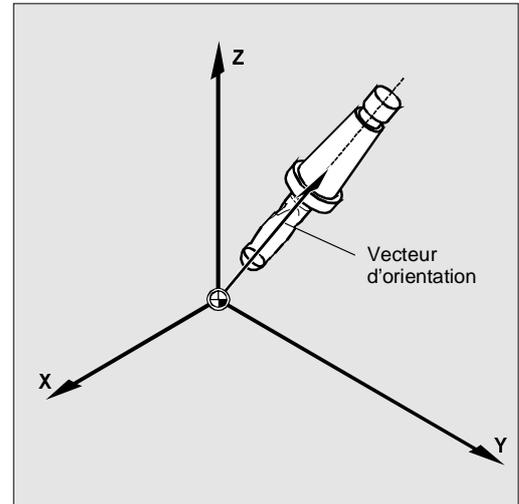
840 D
NCU 572
NCU 573

8.6 Orientation de l'outil



On entend par orientation de l'outil l'orientation géométrique de l'outil dans l'espace.

Sur une machine-outil à cinq axes, l'orientation de l'outil peut être spécifiée par des instructions du programme.



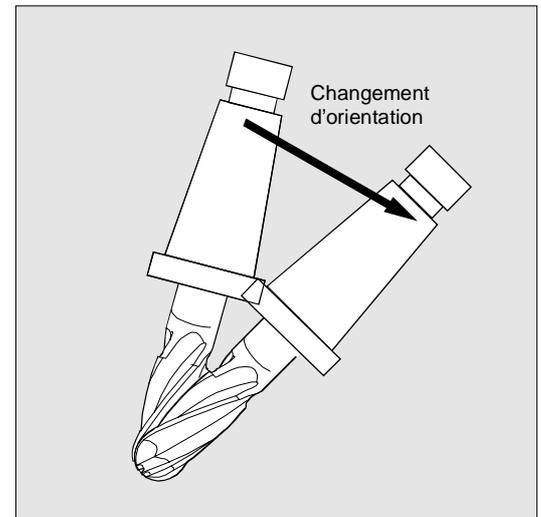
Programmation de l'orientation d'outil

On peut programmer un changement d'orientation de l'outil par :

- programmation directe des axes rotatifs
- Angles d'Euler ou angles RPY
- vecteur de direction
- LEAD/TILT (fraisage en bout)

Le système de coordonnées de référence est soit le système de coordonnées machine (ORIMKS), soit le système de coordonnées pièce courant (ORIWKS).

Un changement d'orientation peut être influencé par :



ORIC	orientation et déplacement le long de la trajectoire en parallèle
ORID	orientation et déplacement successifs le long de la trajectoire
OSOF	pas de lissage de l'orientation
OSC	orientation constante
OSS	lissage de l'orientation uniquement en début de bloc
OSSE	lissage de l'orientation en début et en fin de bloc
ORIS	vitesse de changement d'orientation quand le lissage de l'orientation est activé, en degrés par mm ; s'applique à OSS et OSSE

8.6 Orientation de l'outil



840 D
NCU 572
NCU 573

Comportement aux angles saillants

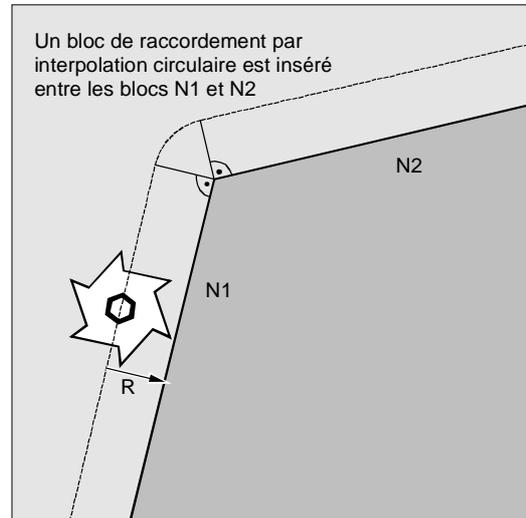
Un bloc à interpolation circulaire avec un rayon égal à celui de la fraise est toujours inséré aux angles saillants.

Les instructions ORIC et ORID permettent de spécifier si les changements d'orientation qui ont été programmés entre les blocs N1 et N2 sont exécutés avant le début du bloc à interpolation circulaire inséré ou en même temps que celui-ci.

Si un changement d'orientation est nécessaire aux angles saillants, celui-ci peut être effectué soit en parallèle à l'interpolation, soit séparément, en même temps que le déplacement le long de la trajectoire.

Avec ORID, les blocs insérés sont d'abord exécutés sans déplacement le long de la trajectoire. Le bloc de raccordement par interpolation circulaire est inséré directement avant le second des deux blocs de déplacement qui forment l'angle.

Quand plusieurs blocs d'orientation sont insérés au niveau d'un angle saillant et si ORIC est activée, le mouvement circulaire est réparti entre les différents blocs insérés, en fonction des valeurs des changements d'orientation des différents blocs insérés.



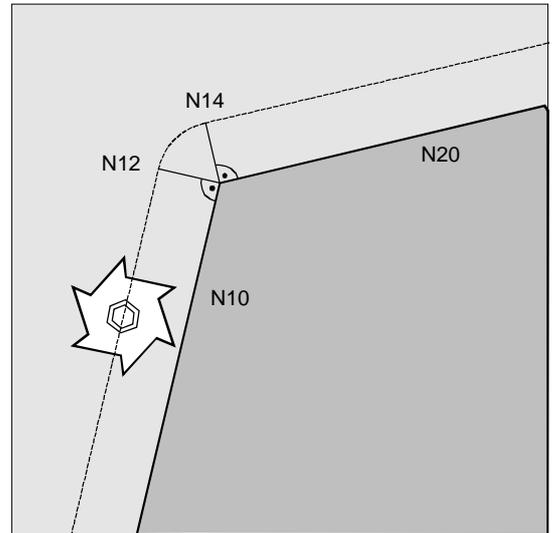


840 D
NCU 572
NCU 573



Exemple de programmation de ORIC

Quand deux ou plusieurs blocs avec des changements d'orientation (par exemple A2= B2= C2=) sont programmés entre les blocs de déplacement N10 et N20 et si ORIC est activée, le bloc à interpolation circulaire inséré est réparti sur ces blocs intermédiaires en fonction de la valeur des modifications angulaires.



ORIC

N8 A2=... B2=... C2=...

N10 X... Y... Z...

N12 C2=... B2=...

N14 C2=... B2=...

Le bloc de raccordement par interpolation circulaire qui est inséré à l'angle saillant est réparti sur N12 et N14 en fonction du changement d'orientation. Le mouvement circulaire et le changement d'orientation sont exécutés en parallèle.

N20 X=...Y=... Z=... G1 F200

8.6 Orientation de l'outil

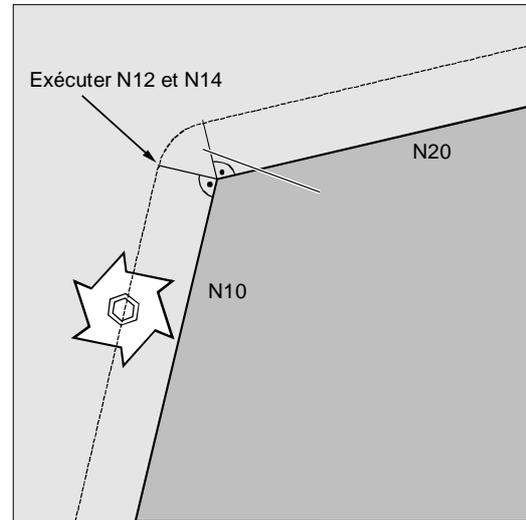


840 D
NCU 572
NCU 573



Exemple de programmation de ORID

Si ORID est activée, tous les blocs situés entre les deux blocs de déplacement sont exécutés à la fin du premier bloc de déplacement. Le bloc à interpolation circulaire avec orientation constante est exécuté directement avant le second bloc de déplacement.



ORID

N8 A2=... B2=... C2=...

N10 X... Y... Z...

N12 A2=... B2=... C2=...

Les blocs N12 et N14 sont exécutés à la fin de N10. Le bloc de raccordement par interpolation circulaire est ensuite exécuté avec l'orientation courante.

N14 M20

Fonctions auxiliaires, etc.

N20 X... Y... Z...



L'instruction de programme qui est active dans le premier bloc de déplacement d'un angle saillant est déterminante pour le type de changement d'orientation à cet angle saillant.



840 D
NCU 572
NCU 573



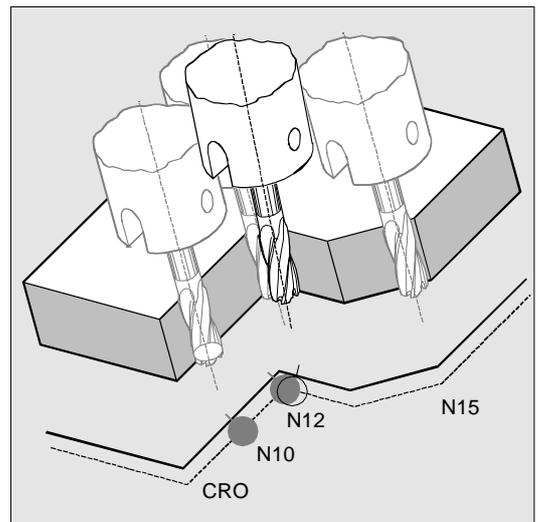
Sans modification de l'orientation

Si l'orientation ne change pas à la fin du bloc, la section de l'outil est un cercle en contact avec les deux contours.



Exemple de programmation

Changement d'orientation sur un angle rentrant



ORIC

N10 X... Y... Z... G1 F500

N12 X... Y... Z... A2=... B2=..., C2=...

N15 X Y Z A2 B2 C2

Mode de déplacement

9.1 Positionnement tangentiel, TANG, TANGON, TANGOF	9-266
9.2 Déplacements conjugués, TRAILON, TRAILOF	9-271
9.3 Table de courbe, CTABDEF, CTABEND, CTAB, CTABINV	9-275
9.4 Couplage de deux axes par valeur pilote, LEADON, LEADOF.....	9-284
9.5 Couplage d'axes à interpolation par valeur pilote, LEADONP, LEADOFD	9-290
9.5.1 Couplage d'axes à interpolation par valeur pilote Type A.....	9-291
9.5.2 Couplage d'axes à interpolation par valeur pilote Type B.....	9-293
9.6 Variation de l'avance, FNORM, FLIN, FCUB, FPO.....	9-295
9.7 Tampon d'exécution, STARTFIFO, STOPFIFO, STOPRE	9-300
9.8 Réaccostage du contour, REPOSA, REPOSL, REPOSQ, REPOSH.....	9-302

9.1 Positionnement tangentiel, TANG, TANGON, TANGOF



840 D
NCU 572
NCU 573



810D

9.1 Positionnement tangentiel, TANG, TANGON, TANGOF



Programmation

TANG (axeA, axeP1, axeP2, coupl., SC)
TANGON (axeA, angle)
TANGOF (axeA)
TLIFT (axeA)



Signification des instructions

TANG	Instruction préparatoire pour la définition d'un asservissement tangentiel
TANGON	Activation du positionnement tangentiel avec indication de l'axe asservi et de l'angle de décalage
TANGOF	Désactivation du positionnement tangentiel avec indication de l'axe asservi
TLIFT	Insertion d'un bloc intermédiaire aux angles du contour



Signification des paramètres

AxeA	Axe asservi : axe rotatif supplémentaire à asservissement tangentiel
AxeP1, AxeP2	Axes pilotes : axes à interpolation à partir desquels est définie la tangente pour l'asservissement
coupl.	Facteur de couplage : rapport entre les variations angulaires de la tangente et l'axe asservi. Indication facultative ; pré réglage : 1
SC	Lettres de codification du système de coordonnées „B“ = système de coordonnées de base ; „W“ = système de coordonnées pièce indication facultative ; pré réglage : "B"
angle	Angle de décalage de l'axe asservi

9.1 Positionnement tangentiel, TANG, TANGON, TANGOF



840 D
NCU 572
NCU 573



810D



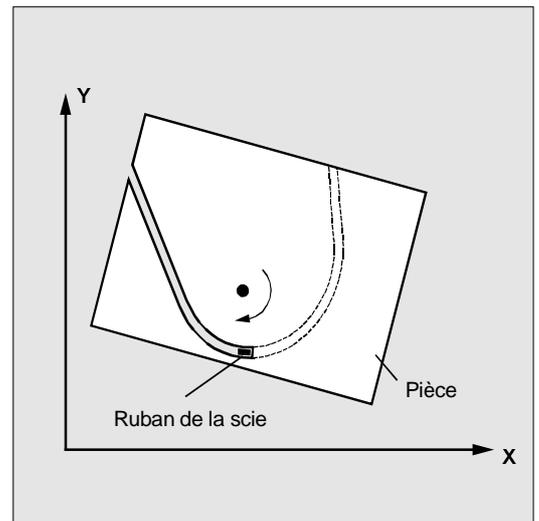
Fonction

Un axe rotatif = l'axe asservi est, comme son nom l'indique, asservi à la trajectoire programmée de deux axes pilotes. L'axe asservi présente un angle de décalage prédéfini par rapport à la tangente à la trajectoire.

Domaines d'application

Le positionnement tangentiel peut être utilisé, entre autres, pour les applications suivantes :

- positionnement tangentiel d'un outil indexable pour le grignotage
- asservissement du positionnement de la pièce dans le cas d'une scie à ruban (voir figure)
- positionnement d'un outil de dressage par rapport à une meule
- positionnement d'une molette de coupe pour la transformation du verre ou du papier
- aménagement tangentiel d'un fil pour le soudage cinq axes
- ...



Procédure

Définir l'axe asservi et les axes pilotes

La définition des axes asservis et des axes pilotes se fait avec TANG.

Un facteur de couplage indique le rapport entre les variations angulaires de la tangente et de l'axe asservi. Sa valeur est en général 1 (préréglage).

L'asservissement peut être effectué dans le système de coordonnées de base „B“ (préréglage) ou dans le système de coordonnées pièce „W“.

Exemple :

TANG (C , X , Y , 1 , " B ")

signifie :

l'axe rotatif C suit les axes géométriques X et Y.

9.1 Positionnement tangentiel, TANG, TANGON, TANGOF



840 D
NCU 572
NCU 573



810D

Activer, désactiver le positionnement tangentiel TANGON, TANGOF

Le positionnement tangentiel est activé avec TANGON, en indiquant l'axe asservi et son angle de décalage souhaité :

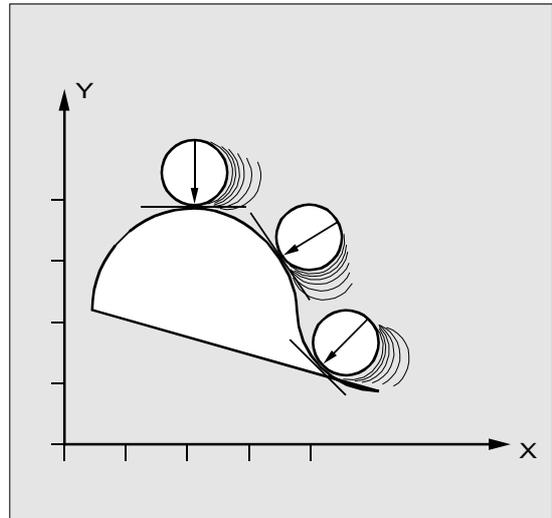
TANGON (C , 90)

signifie :

L'axe C est l'axe asservi. A chaque déplacement des axes à interpolation, il est pivoté pour rester à 90° par rapport à la tangente à la trajectoire.

Pour désactiver le positionnement tangentiel de l'axe asservi, indiquer :

TANGOF (C)



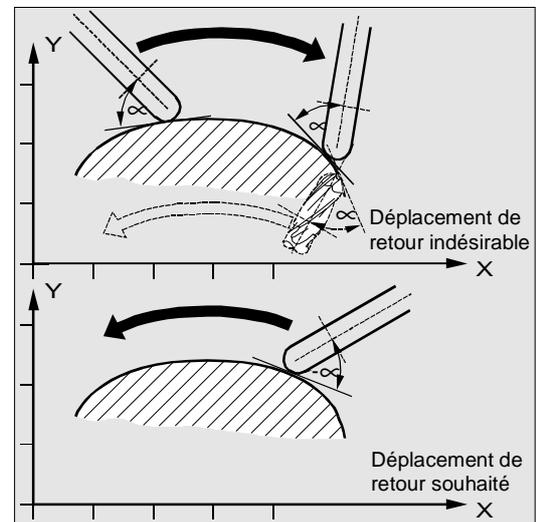
Angle de décalage par limitation de la zone de travail

Quand les déplacements se font en va-et-vient, la tangente fait un saut de 180° au point d'inversion de la trajectoire, l'axe asservi fait de même.

En règle générale, ce comportement n'est pas souhaitable : le déplacement de retour doit s'effectuer suivant le même angle de décalage négatif qu'à l'aller.

Dans ce but, vous limitez la zone de travail de l'axe asservi (G25, G26). La limitation de la zone de travail doit être active au moment de l'inversion de la trajectoire (WALIMON).

Dès que l'angle de décalage sort de la zone de travail limitée, la commande tente de l'y faire revenir avec un angle de décalage négatif.





840 D
NCU 572
NCU 573



810D

Insertion d'un bloc intermédiaire aux angles du contour, TLIFT

Quand on s'approche d'un angle de contour, la tangente au contour montre des variations angulaires brusques qui se répercutent sur la consigne de position de l'axe asservi. Celui-ci essaie normalement de compenser ces brusques variations angulaires par une vitesse maximale. Cependant, sur un certain trajet à la suite de l'angle de contour en question, on constate qu'il en résulte un écart par rapport à la position tangentielle désirée. Si cet écart n'est pas tolérable pour des raisons techniques, on peut avec l'instruction TLIFT et par le biais d'un bloc intermédiaire créé automatiquement, amener le positionnement tangentiel à s'arrêter avant l'angle de contour et à positionner l'axe asservi dans la nouvelle direction tangentielle. Ce positionnement se fait à la vitesse de rotation maximale de l'axe asservi.

Faites figurer l'instruction TLIFT(...) à la suite de la configuration d'axes définie avec TANG(...).

Exemple :

```
TANG(C, X, Y...)
TLIFT(C)
```

Désactivation de TLIFT

A cet effet, reprogrammer la configuration d'axes avec TANG(...), sans la faire suivre de TLIFT(...).



La variation angulaire à partir de laquelle un bloc intermédiaire est à insérer automatiquement, et à définir dans le paramètre machine

```
$MA_EPS_TLIFT_TANG_STEP.
```

9.1 Positionnement tangentiel, TANG, TANGON, TANGOF



840 D
NCU 572
NCU 573



810D

Remarques complémentaires

Influence sur les transformations

La position de l'axe rotatif asservi peut constituer la valeur initiale pour une transformation.

Positionnement explicite de l'axe asservi

Si l'un de vos axes asservis est positionné de manière explicite, la position indiquée s'ajoute à l'angle de décalage programmé.

Tous les types de déplacement sont autorisés : déplacements d'axes à interpolation ou d'axes de positionnement.

Etat du couplage

Vous pouvez consulter l'état du couplage dans le programme pièce CN avec la variable système suivante :

```
$AA_COUP_ACT[axe]
```

0	pas de couplage actif
1,2,3	asservissement tangentiel actif



840 D
NCU 572
NCU 573



810D

9.2 Déplacements conjugués, TRAILON, TRAILOF



Programmation

TRAILON(axeA , axeP , coupl.)

TRAILOF(AxeA , AxeP , Axe2)



Signification des instructions et paramètres

TRAILON	activation et définition d'un groupe d'axes à déplacements conjugués ; effet modal
TRAILOF	désactivation d'un groupe d'axes à déplacements conjugués
AxeA	désignation de l'axe conjugué (asservi)
AxeP	désignation de l'axe pilote
coupl. .	facteur de couplage = course de l'axe conjugué/course de l'axe pilote préréglage = 1

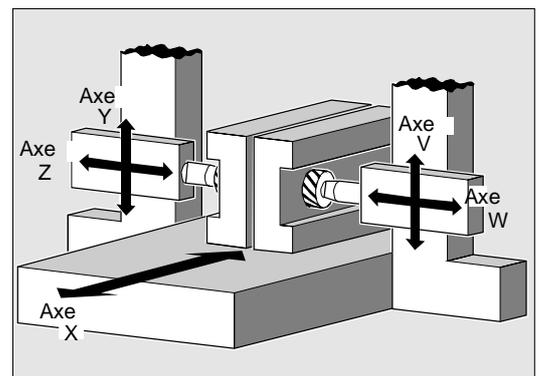


Fonction

Lors du déplacement d'un axe pilote défini, les axes conjugués qui lui sont affectés (= axes asservis) décrivent les courses qui découlent de ce déplacement, compte tenu d'un facteur de couplage. L'axe pilote et les axes asservis constituent un groupe d'axes à déplacements conjugués.

Domaines d'application

- Déplacement d'un axe par le biais d'un autre axe simulé. L'axe pilote est un axe simulé et l'axe conjugué un axe réel. Ainsi, l'axe réel peut être déplacé en tenant compte du facteur de couplage.
- Usinage sur deux faces avec deux groupes d'axes à déplacements conjugués :
 1. axe pilote Y, axe conjugué V
 2. axe pilote Z, axe conjugué W



9.2 Déplacements conjugués, TRAILON, TRAILOF



840 D
NCU 572
NCU 573



810D



Procédure

Définir le groupe d'axes à déplacements conjugués, TRAILON

La définition et l'activation d'un groupe d'axes à déplacements conjugués s'effectuent simultanément avec l'instruction à effet modal TRAILON.

TRAILON(V, Y)

V = axe conjugué, Y = axe pilote

Le nombre de groupes d'axes à déplacements conjugués que vous pouvez activer simultanément dépend uniquement des combinaisons d'axes possibles sur la machine.



Les déplacements conjugués s'effectuent toujours dans le système de coordonnées de base.

Types d'axes dans le groupe d'axes à déplacements conjugués

Un groupe d'axes à déplacements conjugués peut combiner à volonté axes linéaires et axes rotatifs. Un axe simulé peut également être défini comme axe pilote.

Axes conjugués

A un axe conjugué, vous pouvez affecter deux axes pilotes à la fois. L'affectation s'effectue dans des groupes d'axes à déplacements conjugués différents.

Un axe conjugué peut être programmé avec toutes les instructions de déplacement disponibles (G0, G1 G2, G3, ...). En plus des déplacements définis indépendamment, l'axe conjugué décrit les courses déduites de ses axes pilotes avec les facteurs de couplage.



Un axe conjugué peut également être l'axe pilote d'autres axes conjugués. Ceci permet de réaliser des groupes d'axes à déplacements conjugués interdépendants.



840 D
NCU 572
NCU 573



810D

Facteur de couplage

Le facteur de couplage indique le rapport souhaité entre la course de l'axe conjugué et celle de l'axe pilote.

$$\text{facteur de couplage} = \frac{\text{course de l'axe conjugué}}{\text{course de l'axe pilote}}$$

Si le facteur de couplage n'est pas indiqué lors de la programmation, le facteur de couplage 1 est pris en compte automatiquement.

Le facteur est indiqué sous forme de nombre rationnel avec point décimal (type REAL). En entrant une valeur négative, les déplacements de l'axe pilote et de l'axe conjugué s'effectuent en sens opposé.

Désactivation d'un groupe d'axes à déplacements conjugués

La désactivation du couplage à un axe pilote s'effectue avec l'instruction suivante :

```
TRAILOF ( V , Y )
```

V = axe conjugué, Y = axe pilote

TRAILOF avec 2 paramètres désactive uniquement le couplage à 1 axe pilote.

Quand un axe conjugué a 2 axes pilotes, par ex. V=axe conjugué et X,Y=axes pilotes, on peut faire appel à TRAILOF avec 3 paramètres pour désactiver le couplage :

```
TRAILOF ( V , X , Y )
```

Remarques complémentaires

Accélération et vitesse

Les limites d'accélération et de vitesse des axes couplés sont déterminées par l'axe "le moins performant" dans le groupe d'axes en question.

Etat du couplage

Vous pouvez interroger l'état du couplage dans le programme pièce CN avec la variable système suivante :

```
$AA_COUP_ACT [ axe ]
```

0 pas de couplage actif

9.2 Déplacements conjugués, TRAILON, TRAILOF



840 D
NCU 572
NCU 573



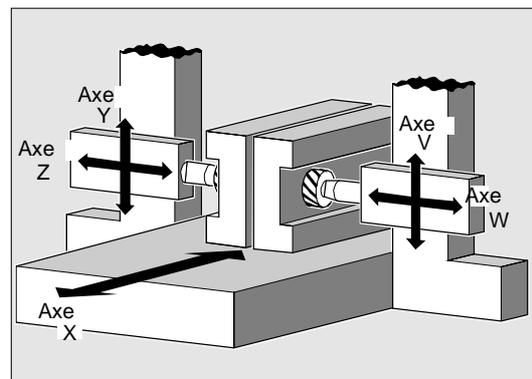
810D

8 déplacements conjugués actifs



Exemple de programmation

La pièce doit être usinée simultanément sur deux faces avec la configuration d'axes représentée ci-contre. Pour ce faire, former deux groupes d'axes à déplacements conjugués.



...

N100 TRAILON(V, Y)

activation du 1er groupe d'axes à déplacements conjugués

N110 TRAILON(W, Z, -1)

activation du 2e groupe d'axes à déplacements conjugués; facteur de couplage négatif : l'axe conjugué se déplace dans le sens opposé à celui de l'axe pilote

N120 G0 Z10

déplacement des axes Z et W en sens opposé

N130 G0 Y20

déplacement des axes Y et V dans le même sens

...

N200 G1 Y22 V25 F200

superposition d'un déplacement dépendant et d'un déplacement indépendant de l'axe conjugué „V“

...

TRAILOF(V, Y)

désactivation du 1er groupe d'axes à déplacements conjugués

TRAILOF(W, Z)

désactivation du 2e groupe d'axes à déplacements conjugués

9.3 Table de courbe, CTABDEF, CTABEND, CTAB, CTABINV



840 D
NCU 572
NCU 573



810D

9.3 Table de courbe, CTABDEF, CTABEND, CTAB, CTABINV



Programmation

La définition des tables de courbes se fait dans un programme pièce.

CTABDEF (axeA , axeP , n , applim)

Définir le début de la table de courbe

CTABEND ()

Définir la fin de la table

CTABDEL (n)

Effacement d'une table

R10=CTAB (VP , n , degré , axeA , axeP)

Valeur asservie correspondant à une valeur pilote

R10=CTABINV (VA , VPapprox , n , degré , axeA ,
axeP)

Valeur pilote correspondant à une valeur asservie



Pour plus d'informations sur le thème des valeurs pilotes et asservies, voir le paragraphe "Couplage de deux axes par valeur pilote" et "Couplage d'axes à interpolation par valeur pilote" dans ce chapitre.



Signification

AxeA	Axe asservi Axe qui sera programmé par le biais de la table des courbes.
AxeP	Axe pilote Axe sur lequel s'appuie la programmation de la valeur pilote.
n	Numéro de la table de courbe
applim	Identificateur de la périodicité de la table de courbe. 0 la table n'est pas périodique 1 la table est périodique
VP	Valeur pilote Position de l'axe pilote pour laquelle une valeur asservie devra être calculée.
dégré	Nom du paramètre de pente
VA	Valeur asservie Position de l'axe asservi pour laquelle la valeur pilote devra être calculée.
VPapprox	Solution approximative pour la valeur pilote, quand il n'a pas été possible de déterminer une valeur pilote univoque pour une valeur asservie donnée.
AxeA , AxeP	Indication facultative de l'axe asservi et / ou de l'axe pilote

9.3 Table de courbe, CTABDEF, CTABEND, CTAB, CTABINV



840 D
NCU 572
NCU 573



810D

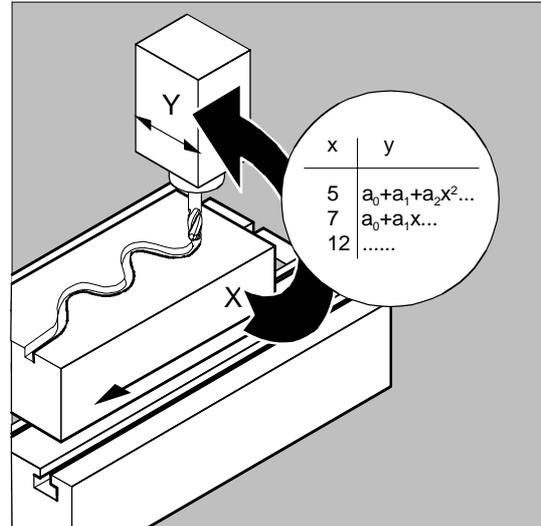


Fonction

Avec les tables de courbes, vous pouvez programmer une corrélation entre les positions et les vitesses de 2 axes.

Exemple : remplacer des cames mécaniques.

La table de courbe est à la base du couplage de deux axes par valeur pilote, puisqu'elle crée le lien fonctionnel entre valeur pilote et valeur asservie : à partir des positions en corrélation réciproque de l'axe asservi et de l'axe pilote, la commande calcule, en cours d'exécution du programme, un polynôme équivalent à la came qu'il est appelé à remplacer.



Remarques complémentaires

Avant de créer des tables de courbes, il convient de réserver suffisamment de mémoire en procédant à un réglage conséquent dans les paramètres machine.

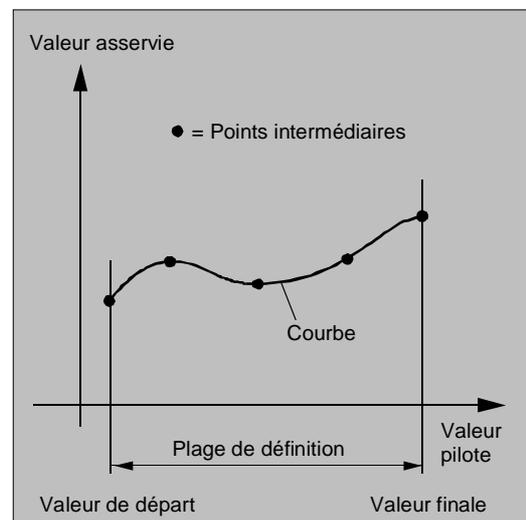


Définition de la table de courbe : CTABDEF, CTABEND

Une table de courbe représente un programme pièce ou une partie d'un programme pièce qui est identifié(e) par l'instruction CTABDEF en début et l'instruction CTABEND en fin de programme.

Au sein de ce programme pièce ou de cette partie de programme pièce, on met en correspondance, par le biais d'instructions de déplacement, des positions isolées de l'axe pilote avec des positions univoques de l'axe asservi ; ces positions servent de points intermédiaires pour le calcul de la courbe sous la forme d'un polynôme du 3e degré maximum.

Le couple de positions axiales corrélées que l'on indique en premier (il s'agit de la première instruction de déplacement) dans la définition de la table de courbe est considéré comme valeur de départ de la plage de définition de la table de courbe. La valeur finale de la plage de définition de la table de courbe est constituée par la dernière instruction de



9.3 Table de courbe, CTABDEF, CTABEND, CTAB, CTABINV



840 D
NCU 572
NCU 573



810D

déplacement.

Pour définir la table de courbe, vous disposez du langage CN dans son intégralité.



Remarques complémentaires

Ne sont pas admis :

- Arrêt du prétraitement des blocs
- la correction du rayon d'outil
- les variations brusques dans le déplacement de l'axe pilote (en cas de changement de transformation par exemple)
- une instruction de déplacement seule pour l'axe asservi
- une inversion du déplacement de l'axe pilote, autrement dit la position de l'axe pilote ne doit jamais être ambiguë.

Toutes les instructions à effet modal qui interviennent dans la définition de la table de courbe deviennent caduques dès que la définition est terminée. En d'autres termes, le programme pièce dans lequel a eu lieu la définition, présente le même état avant et après la définition de la table de courbe.



Les affectations à des paramètres R sont annulées.

Exemple :

```
...
R10=5 R11=20
...
CTABDEF
G1 X=10 Y=20 F1000
R10=R11+5 ;R10=25
X=R10
CTABEND
... ;R10=5
```

Réutilisation de la table de courbe

Le lien fonctionnel créé entre l'axe pilote et l'axe asservi par le biais de la table de courbe est sauvegardé sous le numéro de table qui a été choisi; il est conservé au-delà de la fin du programme et après coupure de l'alimentation.

Une fois créée, une table de courbe est utilisable pour des combinaisons quelconques entre axe pilote et axe asservi, indépendamment des axes qui ont

9.3 Table de courbe, CTABDEF, CTABEND, CTAB, CTABINV



840 D
NCU 572
NCU 573



810D

servi à la créer.



Comportement en marge de la table de courbe

Table de courbe non périodique

Quand la valeur pilote se situe en dehors de la plage de définition, le calcul donne comme valeur asservie la limite supérieure ou la limite inférieure.

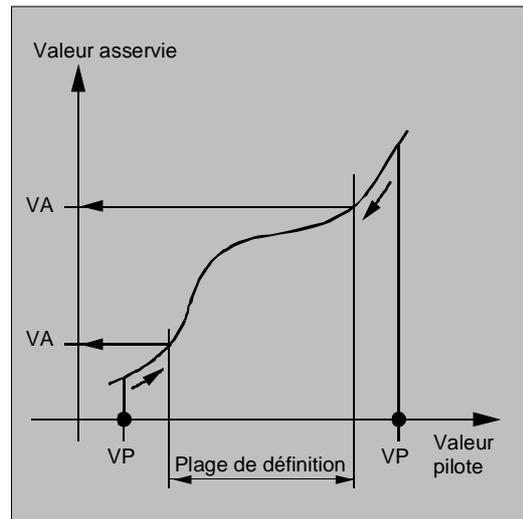
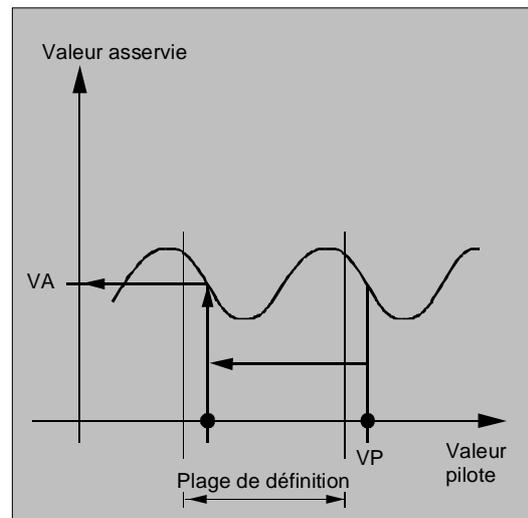


Table de courbe périodique

Quand la valeur pilote se situe en dehors de la plage de définition, la valeur pilote modulo de la plage de définition est prise en compte pour fournir la valeur asservie correspondante.



9.3 Table de courbe, CTABDEF, CTABEND, CTAB, CTABINV



840 D
NCU 572
NCU 573



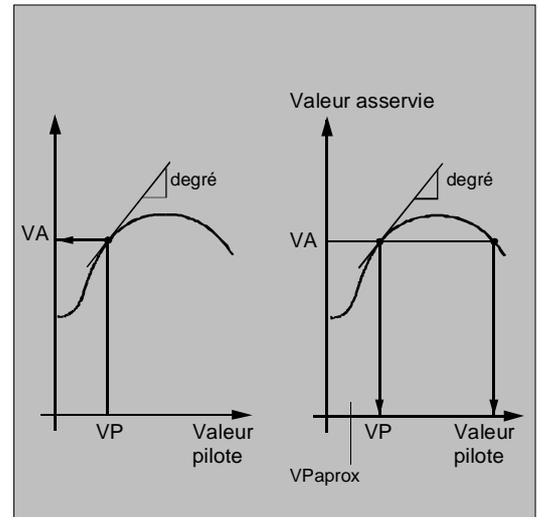
810D

Lecture des positions de la table, CTAB, CTABINV

Avec CTAB, on peut lire directement la valeur asservie correspondant à une valeur pilote, depuis le programme pièce ou une action synchrone (chapitre 10).

Avec CTABINV, on peut lire la valeur pilote correspondant à une valeur asservie. Cette correspondance n'est pas toujours sans équivoque. C'est la raison pour laquelle, CTABINV a toujours besoin d'une valeur approchée (VPaprox) de la valeur pilote recherchée. CTABINV fournit la valeur pilote la plus proche de la valeur approchée. La valeur approchée peut être représentée par exemple par la valeur pilote résultant de la dernière période d'appel de l'interpolateur.

Par ailleurs, les deux fonctions fournissent au paramètre de pente (degré) la pente de la fonction de la table à la position donnée, permettant ainsi le calcul de la vitesse de l'axe asservi et de l'axe pilote à la position en question.



Remarques complémentaires

L'indication optionnelle de l'axe pilote ou de l'axe asservi dans CTAB/CTABINV est importante pour le cas où axe pilote et axe asservi seraient projetés dans des unités de longueur différentes.



Effacement des tables de courbes, CTABDEL

Avec CTABDEL, vous pouvez effacer des tables de courbes. Vous ne pouvez pas effacer des tables de courbes quand elles sont activées dans un couplage.

Ecrasement des tables de courbes

Dès que vous redéfinissez une table de courbe sous un numéro de table déjà utilisé, vous écrasez la table existante. Une table activée ne peut pas être écrasée.

9.3 Table de courbe, CTABDEF, CTABEND, CTAB, CTABINV



840 D
NCU 572
NCU 573



810D



Remarques complémentaires

L'écrasement d'une table de courbe n'est pas accompagné d'un avertissement particulier !



Remarques complémentaires

Avec la variable système \$P_CTABDEF, vous pouvez à tout moment, depuis un programme pièce, interroger le système pour savoir si la définition d'une table de courbe est momentanément activée.

Dès que vous mettez entre parenthèses les instructions de définition de la table de courbe, vous pouvez utiliser cette partie de programme pièce comme un programme pièce réel.

9.3 Table de courbe, CTABDEF, CTABEND, CTAB, CTABINV



840 D
NCU 572
NCU 573



810D



Exemple de programmation

Une partie d'un programme doit servir, sans être modifiée, à la définition d'une table de courbe. L'instruction d'arrêt du prétraitement des blocs STOPRE qui y figure peut rester et sera réactivée dès que cette partie de programme ne sera plus utilisée pour la définition d'une table, c'est-à-dire, dès que CTABDEF et CTABEND auront été supprimés :

```
CTABDEF ( Y , X , 1 , 1 )  
...  
...  
IF NOT ( $P_CTABDEF )  
STOPRE  
ENDIF  
...  
...  
CTABEND
```



Tables de courbes et différents états de fonctionnement

Quand une recherche de blocs est activée, il est impossible de définir une table de courbe. Lorsque le bloc recherché se situe dans la définition d'une table, une alarme est donnée dès que la recherche arrive à CTABEND.

9.3 Table de courbe, CTABDEF, CTABEND, CTAB, CTABINV



840 D
NCU 572
NCU 573

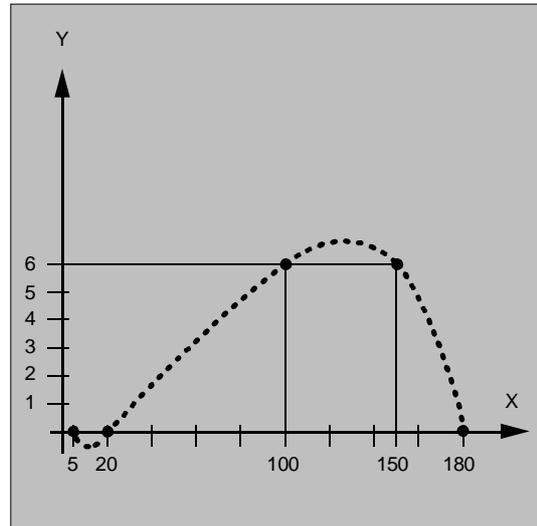


810D



Exemple de programmation 1

Définition d'une table de courbe



N100 CTABDEF(Y,X,3,0)

Début de la définition d'une table de courbe non périodique ayant le numéro 3

N110 X0 Y0

1. Instruction de déplacement, elle définit les valeurs de départ et le 1er pt interm. :
Valeur pilote : 5; Valeur asservie : 0

N120 X20 Y0

2. pt interm. : Valeur pilote 0...20;
Valeur asservie : Valeur de départ...0

N130 X100 Y6

3. pt intermédiaire : Valeur pilote 20...100;
Valeur asservie : 0 ... 6

N140 X150 Y6

4. pt intermédiaire : Valeur pilote 100...150;
Valeur asservie : 6...6

N150 X180 Y0

5. pt intermédiaire : Valeur pilote 150...180;
Valeur asservie : 6...0

N200 CTABEND

Fin de la définition ; la table de courbe est créée dans sa représentation interne sous la forme d'un polynôme du 3e degré maximum ; le calcul de la courbe avec les points intermédiaires donnés dépend du type d'interpolation choisi de façon modale (interpolation circulaire, linéaire, spline) ; le programme pièce retrouve l'état dans lequel il était avant la définition.

9.3 Table de courbe, CTABDEF, CTABEND, CTAB, CTABINV



840 D
NCU 572
NCU 573



810D



Exemple de programmation 2

Définition d'une table de courbe périodique ayant le numéro 2, une plage de valeurs pilotes de 0 à 360, un déplacement de l'axe asservi de 0 à 45

avec retour à 0 :

```

N10 DEF REAL DEPPOS ;
N20 DEF REAL GRADIENT ;
N30 CTABDEF(Y,X,2,1)                                Début de la définition
N40 G1 X=0 Y=0
N50 POLY
N60 PO[X]=(45.0)
N70 PO[X]=(90.0) PO[Y]=(45.0,135.0,-
90)
N80 PO[X]=(270.0)
N90 PO[X]=(315.0) PO[Y]=(0.0,-
135.0,90)
N100 PO[X]=(360.0)
N110 CTABEND                                        Fin de la définition

```

Test de la courbe par couplage de Y à X :

```

N120 G1 F1000 X0
N130 LEADON(Y,X,2)
N140 X360
N150 X0
N160 LEADOF(Y,X)

```

Lecture de la fonction de la table à la valeur pilote

75.0 :

```

N170 DEPPOS=CTAB(75.0,2,GRADIENT)

```

Positionnement de l'axe pilote et de l'axe asservi :

```

N180 G0 X75 Y=DEPPOS

```

Quand le couplage est activé, il n'est pas nécessaire de synchroniser l'axe asservi :

```

N190 LEADON(Y,X,2)
N200 G1 X110 F1000
N210 LEADOF(Y,X)
N220 M30

```

9.4 Couplage de deux axes par valeur pilote, LEADON, LEADOF



840 D
NCU 572
NCU 573



810D

9.4 Couplage de deux axes par valeur pilote, LEADON, LEADOF



Programmation

LEADON (axeA, axeP, n)

LEADOF (axeA, axeP, n)



Signification

LEADON	Activer le couplage par valeur pilote
LEADOF	Désactiver le couplage par valeur pilote
AxeA	Axe asservi
AxeP	Axe pilote
n	Numéro de la table de courbe



Fonction

Quand il y a couplage de deux axes par valeur pilote, l'axe pilote et l'axe asservi se déplacent de façon synchrone. Une position donnée de l'axe asservi correspond sans équivoque à une position de l'axe pilote – simulée le cas échéant – sur la base d'une table de courbe ou d'un polynôme en découplant.

Axe pilote est le nom donné à l'axe qui fournit les valeurs d'entrée à la table de courbe. **Axe asservi** est le nom donné à l'axe qui prend les positions calculées sur la base de la table de courbe.

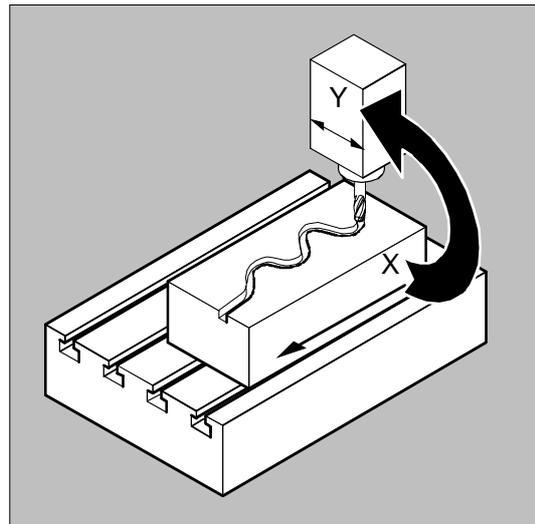
Le couplage par valeur pilote peut être activé et désactivé aussi bien depuis le programme pièce que pendant les déplacements depuis une action synchrone (chapitre 10).

Le couplage par valeur pilote se réfère toujours au système de coordonnées de base.

Pour créer une table de courbe, voir

"Tables de courbes" dans le présent chapitre.

Pour des informations sur le couplage par valeur pilote, voir sous /FB/, M3, déplacements conjugués et couplage par valeur pilote



9.4 Couplage de deux axes par valeur pilote, LEADON, LEADOF



840 D
NCU 572
NCU 573



810D

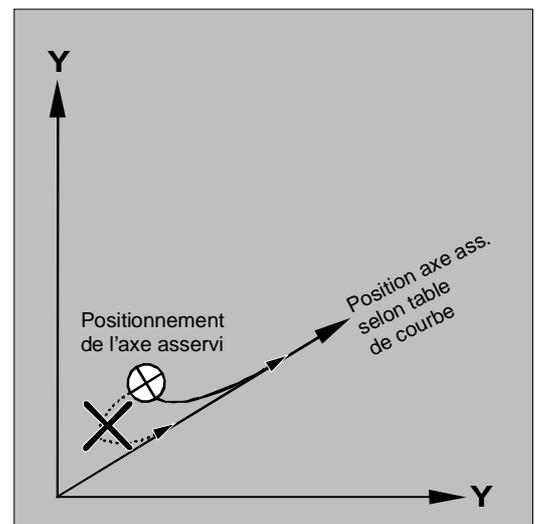


Procédure

Le couplage par valeur pilote exige la synchronisation de l'axe pilote et de l'axe asservi. Cette synchronisation est obtenue uniquement si l'axe asservi se trouve, au moment où le couplage par valeur pilote est activé, dans la plage de tolérance de la courbe calculée avec la table.

La plage de tolérance pour la position de l'axe asservi est définie par le paramètre machine 37200 COUPLE_POS_TOL_COARSE.

Si, lors de l'activation du couplage par valeur pilote, l'axe asservi n'est pas encore positionné correctement, le synchronisme s'enclenchera automatiquement dès que la position de consigne calculée pour l'axe asservi approchera de sa position effective. Pendant la synchronisation, l'axe asservi est déplacé dans la direction définie par sa vitesse de consigne (calculée à partir de la vitesse de l'axe pilote et CTAB).



Remarques complémentaires

Si la position calculée de l'axe asservi s'éloigne de sa position effective lors de l'activation du couplage par valeur pilote, il n'y a pas de synchronisation.



Couplage par valeur réelle et par valeur de consigne

Comme valeurs pilotes, autrement dit comme valeurs de départ pour déterminer les positions de l'axe asservi, on peut utiliser :

- les valeurs réelles de l'axe pilote : couplage par la valeur réelle
- les valeurs de consigne de l'axe pilote : couplage par la valeur de consigne



840 D
NCU 572
NCU 573



810D

Remarques complémentaires

Comparé au couplage par la valeur réelle, le couplage par la valeur de consigne fournit un meilleur synchronisme entre l'axe pilote et l'axe asservi ; c'est la raison pour laquelle il est pré-réglé en version standard.

Le couplage par la valeur de consigne est possible uniquement si l'axe pilote et l'axe asservi sont interpolés par la même NCU. Quand l'axe pilote est un axe externe, le couplage de l'axe asservi avec cet axe pilote ne peut se faire que par des valeurs réelles.

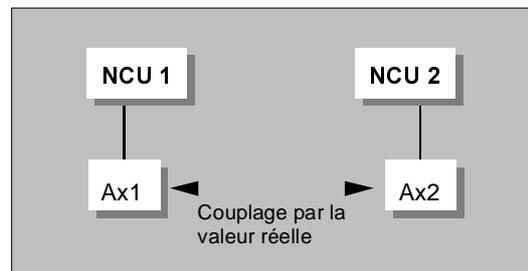
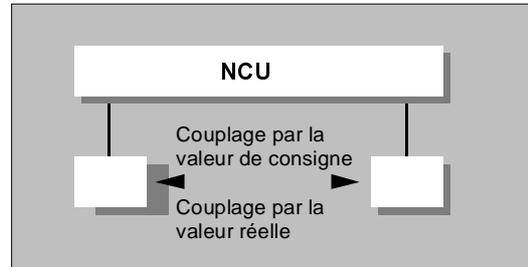
Basculement entre couplage par valeur réelle et couplage par valeur de consigne

Le basculement est possible par le biais de la donnée de réglage \$SA_LEAD_TYPE

Le basculement entre couplage par valeur réelle et couplage par valeur de consigne doit toujours se faire quand l'axe asservi est en état d'immobilisation. En effet, une resynchronisation après le basculement n'a lieu que dans cet état.

Exemple d'application :

Quand la machine est soumise à de fortes vibrations, la lecture des valeurs réelles ne peut pas se faire correctement. Pour mettre en oeuvre le couplage par valeur pilote dans un transfert sur presse, il peut être nécessaire de basculer du couplage par valeur réelle au couplage par valeur de consigne, dans les phases d'usinage où les vibrations sont les plus fortes.



9.4 Couplage de deux axes par valeur pilote, LEADON, LEADOF



840 D
NCU 572
NCU 573



810D



Simulation de valeurs pilotes

dans un couplage par la valeur de consigne

Par le biais d'un paramètre machine, on peut séparer l'interpolateur de l'axe pilotage et le servo. On peut alors, dans un couplage par valeur de consigne, créer des valeurs de consigne sans qu'il y ait un déplacement réel de l'axe pilote.

Les valeurs pilotes créées par un couplage par la valeur de consigne peuvent être lues dans les variables suivantes, afin d'être utilisées dans des actions synchrones par exemple :

- \$AA_LEAD_P
- \$AA_LEAD_V

Valeur pilote de position

Valeur pilote de vitesse



Remarques complémentaires

Vous pouvez aussi créer des valeurs pilotes avec d'autres procédés que vous avez vous-même programmés. Les valeurs pilotes créées sont écrites dans les variables

- \$AA_LEAD_SP
- \$AA_LEAD_SV

Valeur pilote de position

Valeur pilote de vitesse

et y sont lues. Pour utiliser ces variables, il convient de régler donnée de réglage \$SA_LEAD_TYPE = 2.

Etat du couplage

Vous pouvez interroger l'état du couplage dans le programme pièce CN avec la variable système suivante :

\$AA_COUP_ACT[axe]

- | | |
|----|----------------------------------|
| 0 | pas de couplage actif |
| 16 | couplage par valeur pilote actif |



Désactiver le couplage par valeur pilote, LEADOF

Après la désactivation du couplage par valeur pilote, l'axe asservi redevient axe de commande normal !.

Couplage de deux axes par valeur pilote et différents états de fonctionnement

Selon le réglage qui a été fait dans les paramètres machine, les couplages par valeur pilote sont désactivés avec RESET.



840 D
NCU 572
NCU 573



810D



Exemple de programmation

Sur une presse, un couplage mécanique conventionnel entre un axe pilote (axe d'emboutissage) et les axes d'un système de transfert est à remplacer par un couplage électronique.

Il démontre comment on peut remplacer sur une presse un système de transfert mécanique par un système de transfert électronique. Les phases de couplage et de découplage sont réalisées en tant qu'actions synchrones **statiques**.

Les axes de transfert et les axes auxiliaires sont pilotés par l'axe pilote VP (axe d'emboutissage) en tant qu'axes asservis, par le biais des tables de courbes.

Axes asservis

X	Axe d'avance, voire axe longitudinal
YL	Axe de fermeture, voire axe transversal
ZL	Axe de levage
U	Dispositif d'avance à rouleaux, axe auxiliaire
V	Tête de dressage, axe auxiliaire
W	Graissage, axe auxiliaire

Gestion des états

Les phases de commutation et de couplage sont gérées par le biais de variables temps réel :

`$AC_MEMENTO[i]=n`

avec :

i	numéro de memento
n	valeur d'état

Actions

Comme actions, on trouve par ex. dans les actions synchrones :

- Couplage, LEADON(axe asservi, axe pilote, numéro de la table de courbe)
- Découplage, LEADOF(axe asservi, axe pilote)
- Préréglage des mémoires de valeurs réelles, PRESETON(axe, valeur)
- Définition du memento, `$AC_MEMENTO[i]=valeur`
- Type de couplage : valeur réelle/virtuelle
- Accostage de positions d'axes, `POS[axe]=valeur`

Conditions

Des entrées TOR rapides, des variables temps réel `$AC_MEMENTO` et des comparaisons de positions, combinées avec l'opérateur logique AND, sont exploitées comme conditions.

Remarque

Dans l'exemple qui suit, nous avons utilisé des retours de lignes, des retraits et des caractères **gras** pour une meilleure lisibilité de la programmation. Pour la commande, tout ce qui figure sous un numéro de ligne représente effectivement une ligne de programme.

9.4 Couplage de deux axes par valeur pilote, LEADON, LEADOF



840 D
NCU 572
NCU 573



810D

Commentaire

; Définit toutes les actions synchrones statiques.

; **** Remise à zéro du memento

```
N2   $AC_MEMENTO[0]=0 $AC_MEMENTO[1]=0
      $AC_MEMENTO[2]=0 $AC_MEMENTO[3]=0
      $AC_MEMENTO[4]=0 $AC_MEMENTO[5]=0
      $AC_MEMENTO[6]=0 $AC_MEMENTO[7]=0
```

; **** E1 0=>1 Couplage transfert ACTIVE

```
N10  IDS=1   EVERY ($A_IN[1]==1) AND
      ($A_IN[16]==1) AND ($AC_MEMENTO[0]==0)
DO   LEADON(X,LW,1) LEADON(YL,LW,2)
      LEADON(ZL,LW,3) $AC_MEMENTO[0]=1
```

; **** E1 0=>1 Couplage dispositif d'avance à
rouleaux ACTIVE

```
N20  IDS=11  EVERY ($A_IN[1]==1) AND
      ($A_IN[5]==0) AND ($AC_MEMENTO[5]==0)
DO   LEADON(U,LW,4) PRESETON(U,0)
      $AC_MEMENTO[5]=1
```

; **** E1 0->1 Couplage tête de dressage
ACTIVE

```
N21  IDS=12  EVERY ($A_IN[1]==1) AND
      ($A_IN[5]==0) AND ($AC_MEMENTO[6]==0)
DO   LEADON(V,LW,4) PRESETON(V,0)
      $AC_MEMENTO[6]=1
```

; **** E1 0->1 Couplage graissage ACTIVE

```
N22  IDS=13  EVERY ($A_IN[1]==1) AND
      ($A_IN[5]==0) AND ($AC_MEMENTO[7]==0)
DO   LEADON(W,LW,4) PRESETON(W,0)
      $AC_MEMENTO[7]=1
```

; **** E2 0=>1 Couplage DESACTIVE

```
N30  IDS=3   EVERY ($A_IN[2]==1)
DO   LEADOF(X,LW) LEADOF(YL,LW)
      LEADOF(ZL,LW) LEADOF(U,LW)
      LEADOF(V,LW) LEADOF(W,LW) $AC_MEMENTO[0]=0
      $AC_MEMENTO[1]=0 $AC_MEMENTO[3]=0
      $AC_MEMENTO[4]=0 $AC_MEMENTO[5]=0
      $AC_MEMENTO[6]=0 $AC_MEMENTO[7]=0
```

....

```
N110 G04 F01
```

```
N120 M30
```

9.5 Couplage à interpol. par valeur pilote, LEADONP, LEADOFFP



840 D
NCU 572
NCU 573



810D

9.5 Couplage à interpol. par valeur pilote, LEADONP, LEADOFFP



Programmation

LEADONP(axeP, pos. activation, couplage)

Activer le couplage des axes à interpolation par valeur pilote **Type A**

LEADONP(axeP, pos. activation)

Activer le couplage des axes à interpolation par valeur pilote **Type B**

LEADOFFP

Désactiver le couplage des axes à interpolation par valeur pilote



Signification

AxeP	Axe pilote
Position d'activation	Avec Type A : valeur pilote pour laquelle doit se synchroniser "au vol" la trajectoire couplée. Le couplage est activé dès que la valeur pilote passe par la position d'activation. Avec Type B : Position de l'axe pilote à partir de laquelle le couplage devient actif.
couplage	facteur de couplage Le signe négatif du facteur de couplage est ignoré.



Fonction

Le couplage d'axes à interpolation par valeur pilote permet de coupler tous les axes à interpolation (sans table de courbe) à un axe pilote :

Type A : L'axe pilote n'est pas un axe à interpolation

Type B : L'axe pilote est un axe à interpolation utilisé dans le programme CN.



Remarques complémentaires

En cas de RESET et à la fin du programme, le couplage d'axes à interpolation par valeur pilote est désactivé.

Pour créer une table de courbe, voir

"Tables de courbes" dans le présent chapitre.

Pour des informations sur le couplage par valeur pilote, voir sous /FB/, M3, déplacements conjugués et couplage par valeur pilote

9.5 Couplage à interpol. par valeur pilote, LEADONP, LEADOFFP



840 D
NCU 572
NCU 573



810D

9.5.1 Couplage d'axes à interpolation par valeur pilote Type A



Procédure

L'axe pilote peut être un axe externe quelconque, mais ne doit pas être un axe à interpolation.

A partir de la position d'activation définie, l'avance des axes à interpolation couplés se fait en synchronisation avec l'avance de l'axe pilote.

Dans le bloc contenant l'appel du couplage d'axes à interpolation par valeur pilote avec LEADONP, l'avance tangentielle le long de la trajectoire programmée est nulle.

L'avance tangentielle est synchronisée avec l'avance de l'axe pilote jusqu'à la position d'activation définie.



Remarques complémentaires

Restrictions

- Il n'est pas possible d'inverser le sens de l'axe pilote.
- Pour utiliser le couplage d'axes à interpolation par valeur pilote, le contournage (G64) doit être activé.

9.5 Couplage à interpol. par valeur pilote, LEADONP, LEADOFF



840 D
NCU 572
NCU 573

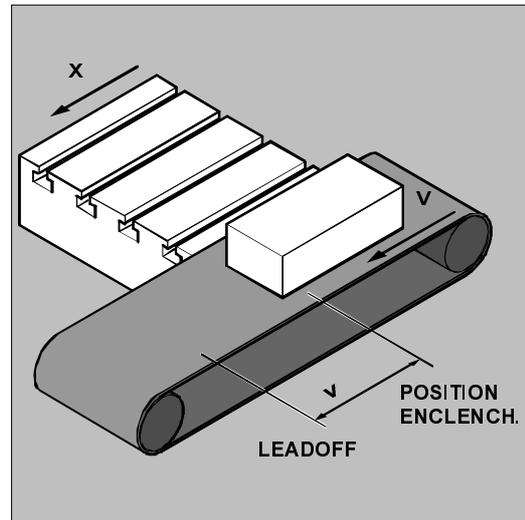


810D



Exemple de programmation

Une machine doit prendre en charge une pièce qui est acheminée sur un tapis roulant à commande externe. L'axe X de la machine parallèle au tapis roulant V doit être synchronisé temporairement avec le tapis roulant.



...

N100 G64 X0

Activer le contournage

N110 LEADONP(V,100,1)

Activer le couplage d'axes à interpolation par valeur pilote. V est l'axe pilote. L'axe X doit entrer en synchronisation à la position V=100. Facteur de couplage 1. La synchronisation se fait "au vol" pendant le déplacement de X, à la position V = 100.

N120 X150

La prise en charge de la pièce est terminée à X=150.

N130 LEADOFFP

Désactiver le couplage par valeur pilote et ...

N140 X160

...passer en continu de façon tangentielle à ...

N150 X... Y... Z...

...l'usage.

...

9.5 Couplage à interpol. par valeur pilote, LEADONP, LEADOPF



840 D
NCU 572
NCU 573



810D

9.5.2 Couplage d'axes à interpolation par valeur pilote Type B



Procédure

L'axe pilote peut être un axe externe quelconque, qui est programmé comme axe à interpolation. Au cours de l'exécution du programme CN, l'avance de tous les autres axes à interpolation se fera en fonction de cet axe pilote.

Dans le bloc contenant l'appel du couplage d'axes à interpolation par valeur pilote avec LEADONP, l'avance tangentielle le long de la trajectoire programmée est nulle.



Remarques complémentaires

Restrictions

- Les axes à interpolation couplés ne peuvent se déplacer que simultanément avec l'axe pilote.
- Il n'est pas possible d'inverser le sens de l'axe pilote.
- Pour utiliser le couplage d'axes à interpolation par valeur pilote, le contournage (G64) doit être activé.

9.5 Couplage à interpol. par valeur pilote, LEADONP, LEADOFF



840 D
NCU 572
NCU 573



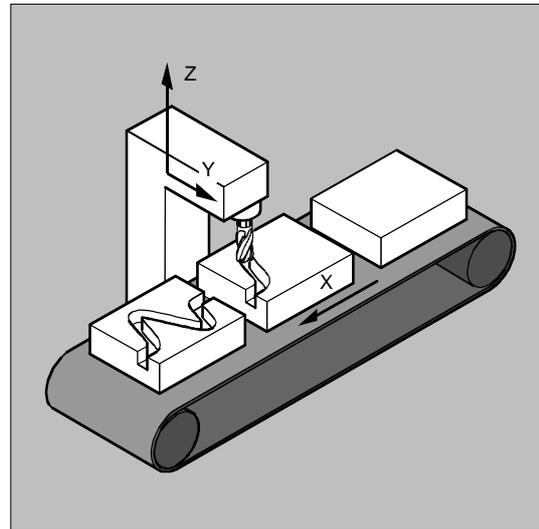
810D



Exemple de programmation

Dans le cas d'un usinage en série, il arrive souvent qu'on exécute sur une machine-outil un programme CN standard, mais que l'un des axes impliqués ne soit pas sous le contrôle de la CN ou du canal concerné :

Sur une machine à travailler le bois, un programme CN doit être exécuté dans les axes X, Y et Z. X est un axe de transport à commande externe, dont on ne dispose que les valeurs réelles. Pour adapter l'avance des axes à interpolation Y et Z en fonction de l'axe X, on fait précéder dans le programme d'usinage l'instruction d'activation du couplage d'axes à interpolation par valeur pilote.



...

N90 G64 X0

Activer le contourage

N100 LEADONP(X, XPOS)

L'axe pilote est X. Lancement du couplage quand X=XPOS.

N110 X... Y... Z...

...



840 D
NCU 572
NCU 573



810D

9.6 Variation de l'avance, FNORM, FLIN, FCUB, FPO



Programmation

F... FNORM
F... FLIN
F... FCUB
F=FPO(... , ... , ...)



Signification

FNORM	Préréglage. La valeur de l'avance est définie par le biais de la longueur de la trajectoire du bloc et vaut alors comme valeur modale.
FLIN	Variation linéaire de l'avance tangentielle : La valeur de l'avance varie linéairement de la valeur en début de bloc à la valeur en fin de bloc et vaut alors comme valeur modale.
FCUB	Variation cubique de l'avance tangentielle : Les valeurs F programmées dans les blocs - rapportées au point final de bloc - sont raccordées par un spline. Le spline débute et se termine de façon tangentielle aux valeurs d'avance programmées dans le bloc précédent, voire suivant. Si l'adresse F manque dans un bloc, c'est la dernière valeur F programmée qui sera utilisée.
F=FPO...	Variation de l'avance tangentielle définie par un polynôme : L'adresse F décrit la variation de l'avance selon un polynôme, de la valeur courante jusqu'à la valeur en fin de bloc. La valeur finale vaut alors comme valeur modale.



Fonction

En vue d'une introduction plus flexible de l'avance, la programmation de l'avance selon DIN 66025 est étendue aux variations linéaires et cubiques. Les variations cubiques peuvent être programmées directement ou sous la forme de splines interpolés.

Par ce moyen, il est possible de programmer des variations continues et lissées, dépendant de la courbure de la pièce à usiner.

Ces variations de la vitesse permettent des variations d'accélération sans à-coup et, de ce fait, la réalisation d'états de surface plus réguliers.

9.6 Variation de l'avance, FNORM, FLIN, FCUB, FPO



840 D
NCU 572
NCU 573



810D

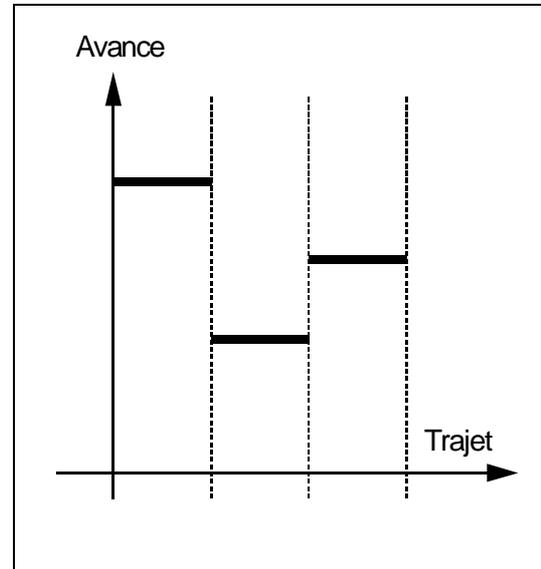


Procédure

FNORM

L'adresse d'avance F désigne l'avance tangentielle en tant que valeur constante selon DIN 66025.

Vous trouverez plus d'informations sur ce sujet dans "Manuel de programmation - Notions de base".

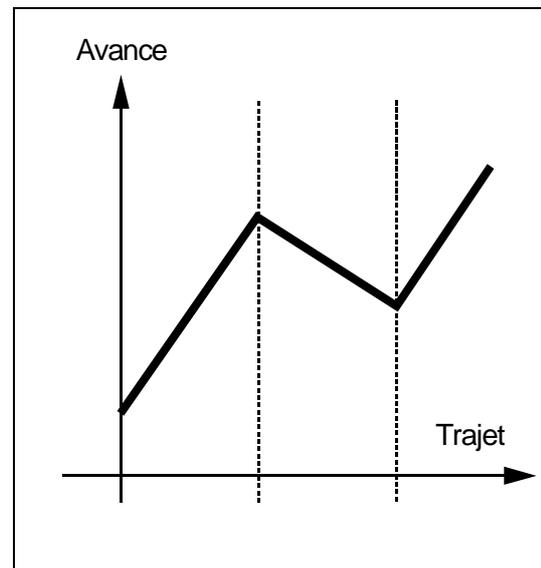


FLIN

L'avance varie linéairement jusqu'à la fin du bloc, depuis la valeur courante jusqu'à la valeur F programmée.

Exemple :

```
N30 F1400 FLIN X50
```





840 D
NCU 572
NCU 573



810D

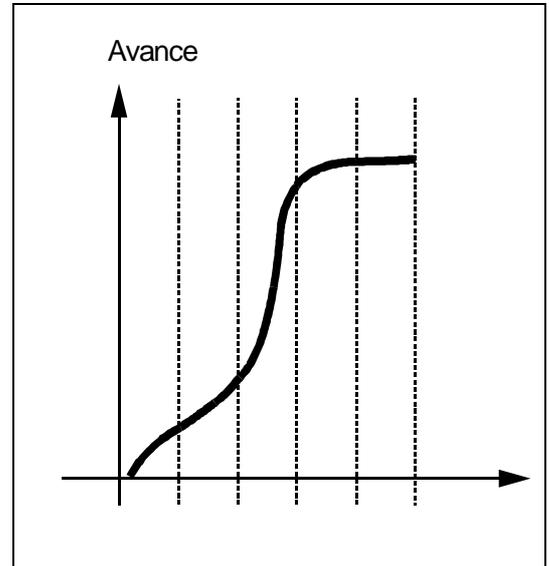
FCUB

L'avance varie de façon cubique jusqu'à la fin du bloc, depuis la valeur courante jusqu'à la valeur F programmée. La commande relie toutes les valeurs programmées dans les blocs avec un FCUB actif par des splines. Les valeurs d'avance servent dans ce cas de points intermédiaires pour le calcul de l'interpolation spline.

Exemple :

```
N50 F1400 FCUB X50
N60 F2000 X47
N70 F3800 X52
```

...



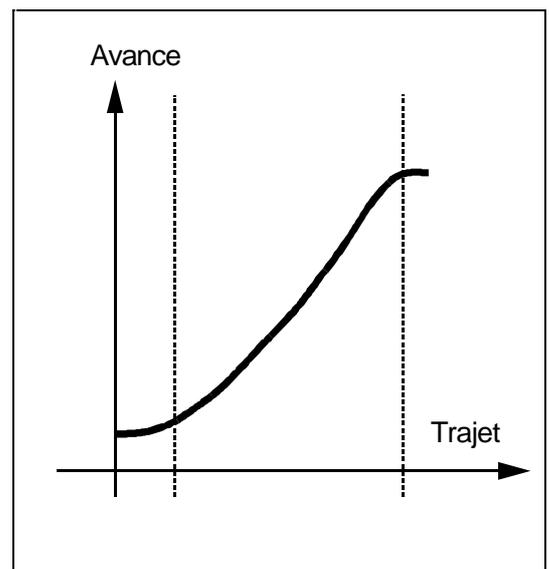
F=FPO(...,...)

La variation de l'avance est directement programmée par le biais d'un polynôme. L'introduction des coefficients du polynôme est la même que pour l'interpolation polynomiale.

Exemple :

```
F=FPO(endfeed, quadf, cubf)
```

endfeed, quadf et cubf sont des variables précédemment définies.



endfeed :	Avance en fin de bloc
quadf :	Coefficient de polynôme de degré 2
cubf :	Coefficient de polynôme de degré 3

Quand FCUB est activée, le spline se raccorde tangentiellement en début et en fin de bloc à l'avance définie par le biais de FPO.

Conditions marginales

Les fonctions programmées pour le mode de déplacement restent valables, indépendamment de la variation de l'avance.

9.6 Variation de l'avance, FNORM, FLIN, FCUB, FPO



840 D
NCU 572
NCU 573



810D

La variation programmable de l'avance est toujours valable de façon absolue, indépendamment de G90 ou de G91.



Remarques complémentaires

Compactage

Lorsque le compactage COMPON est activé, les règles ci-après sont appliquées quand plusieurs blocs sont regroupés en un même segment de spline :

FNORM:

Pour le segment de spline, le mot F du dernier bloc qui en fait partie est valide.

FLIN :

Pour le segment de spline, le mot F du dernier bloc qui en fait partie est valide.

La valeur F programmée est valide en fin du segment et est alors accostée de façon linéaire.

FCUB :

Le spline d'avance généré s'écarte au maximum des points finaux programmés d'une valeur égale à la valeur définie dans le paramètre machine

`$MC_COMPRESS_VELO_TOL` .

F=FPO (...,...)

Ces blocs ne sont pas compactés.

Optimisation de l'avance sur des sections de trajectoire incurvées

Le polynôme d'avance $F=FPO$ et le spline d'avance FCUB doivent toujours être exécutés à vitesse de coupe constante CFC. De ce fait, une variation de consigne d'avance à accélération continue est générée.

9.6 Variation de l'avance, FNORM, FLIN, FCUB, FPO



840 D
NCU 572
NCU 573

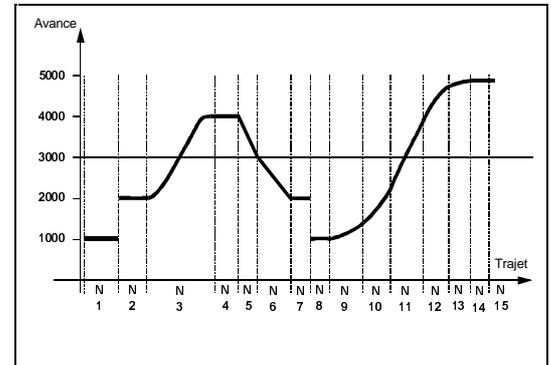


810D



Exemple de programmation

Vous trouverez, dans cet exemple, la programmation et la représentation graphique de différentes courbes d'avance.



N1 F1000 FNORM G1 X8 G91 G64	Courbe d'avance constante, indication en cotes relatives
N2 F2000 X7	Variation brusque de la vitesse de consigne
N3 F=FPO(4000, 6000, -4000)	Courbe d'avance par polynôme avec avance 4000 en fin de bloc
N4 X6	Avance linéaire 4000 prise comme valeur modale
N5 F3000 FLIN X5	Courbe d'avance linéaire
N6 F2000 X8	Courbe d'avance linéaire
N7 X5	Avance linéaire prise comme valeur modale
N8 F1000 FNORM X5	Courbe d'avance constante avec variation d'accélération par à-coup.
N9 F1400 FCUB X8	Toutes les valeurs de F programmées dans les blocs suivants sont reliées par des splines.
N10 F2200 X6	
N11 F3900 X7	
N12 F4600 X7	
N13 F4900 X5	Désactiver la courbe spline.
N14 FNORM X5	
N15 X20	

9.7 Tampon d'exécution, STARTFIFO, STOPFIFO, STOPRE



840 D
NCU 572
NCU 573



810D

9.7 Tampon d'exécution, STARTFIFO, STOPFIFO, STOPRE



Signification des instructions

STARTFIFO	Fin d'une phase d'usinage rapide, remplissage du tampon d'exécution
STOPFIFO	Début d'une phase d'usinage rapide
STOPRE	Arrêt du prétraitement des blocs



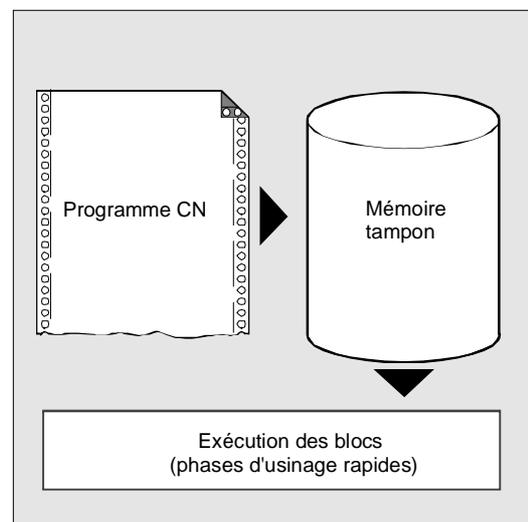
Fonction

Selon sa configuration, la commande dispose d'une certaine mémoire tampon d'une certaine capacité, capable de stocker les blocs prétraités afin de pouvoir ensuite les transmettre à un rythme rapide à la machine-outil.

Ainsi, on peut effectuer des trajets courts à grande vitesse.

La mémoire tampon est remplie au fur et à mesure que le permet la commande.

STARTFIFO arrête l'usinage jusqu'à ce que la mémoire tampon d'exécution soit remplie, plus précisément jusqu'à ce qu'apparaît l'instruction STOPFIFO ou STOPRE.



Procédure

Marquage de la phase d'usinage

La phase d'usinage qui doit être stockée dans la mémoire tampon est marquée au début et à la fin avec STARTFIFO et STOPFIFO.

Exemple :

```
N10 STARTFIFO
N20...
N100
N110 STOPFIFO
```

L'exécution des blocs débute seulement quand la mémoire tampon est remplie ou à la suite d'une instruction STOPFIFO.



840 D
NCU 572
NCU 573



810D

Restrictions

La mémoire tampon d'exécution ne se remplit pas, quand la phase d'usinage contient des instructions qui obligent à un fonctionnement sans tampon (accostage du point de référence, fonctions de mesure, ...).

Arrêt du prétraitement des blocs

Quand on programme STOPRE, le prochain bloc est exécuté seulement si tous les blocs prétraités et mémorisés antérieurement ont été entièrement exécutés. Le bloc précédent exécute un arrêt précis (comme G9).

Exemple :

```
N10 ...
N30 MEAW=1 G1 F1000 X100 Y100 Z50
N40 STOPRE
```

Quand il y a interrogation des données d'état de la machine (\$A...), la commande exécute un arrêt interne du prétraitement des blocs.

Exemple :

```
R10 = $AA_IM[X] ;lecture de la valeur réelle de l'axe X
```



Remarque :

Quand la correction d'outil est activée et pendant les interpolations de type spline, il est conseillé de ne pas programmer STOPRE, pour éviter d'interrompre des séquences de blocs qui forment un ensemble.

840 D
NCU 571840 D
NCU 572
NCU 573

9.8 Réaccostage du contour, REPOSA, REPOSL, REPOSQ, REPOSH



Programmation

REPOSA RMI DISPR=... OU REPOSA RMB OU REPOSA RME

REPOSL RMI DISPR=... OU REPOSL RMB OU REPOSL RME

REPOSQ RMI DISPR=... DISR=... OU REPOSQ RMB DISR=... OU REPOSQ RME DISR=... OU REPOSQA
DISR=...

REPOSH RMI DISPR=... DISR=... OU REPOSH RMB DISR=... OU REPOSH RME DISR=... OU
REPOSHA DISR=...



Signification des instructions

Distance d'accostage

REPOSA	Accostage sur une droite avec tous les axes
REPOSL	Accostage sur une droite
REPOSQ DISR=...	Accostage sur un quart de cercle avec rayon DISR
REPOSQA DISR=...	Accostage dans tous les axes sur un quart de cercle avec rayon DISR
REPOSH DISR=...	Accostage sur un demi-cercle avec diamètre DISR
REPOSHA DISR=...	Accostage dans tous les axes sur un demi-cercle avec rayon DISR

Point de réaccostage

RMI	Accostage du point d'interruption
RMI DISPR=...	Point d'entrée à une distance DISPR en mm/inch avant le point d'interruption
RMB	Accostage du point de début de bloc
RME DISPR=...	Accostage du point de fin de bloc à une distance DISPR avant le point final
A0 B0 C0	Axes dans lesquels le déplacement doit avoir lieu



840 D
NCU 571



840 D
NCU 572
NCU 573



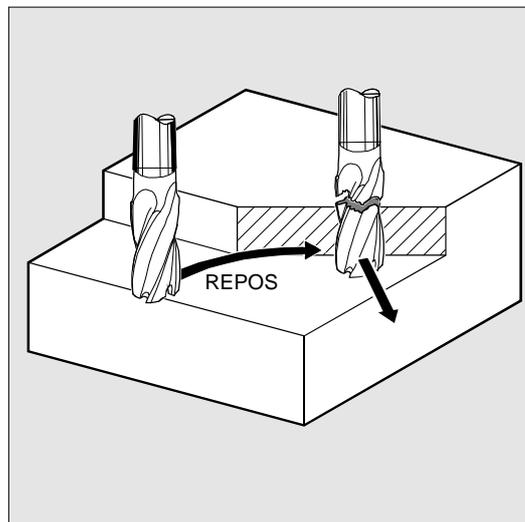
Fonction

Si vous stoppez le programme en cours d'usinage et si vous dégagez l'outil – pour éviter par exemple un bris d'outil ou parce que vous voulez faire une mesure – vous pouvez programmer le réaccostage du contour à un point de votre choix.

L'instruction REPOS a le même effet qu'un retour de sous-programme (par ex. avec M17). Les blocs suivants dans la routine d'interruption ne seront plus exécutés.



Dans le chapitre "Routine d'interruption" du manuel de programmation, vous trouverez de plus amples informations sur l'interruption de l'exécution d'un programme.



Procédure

Définir le point de réaccostage

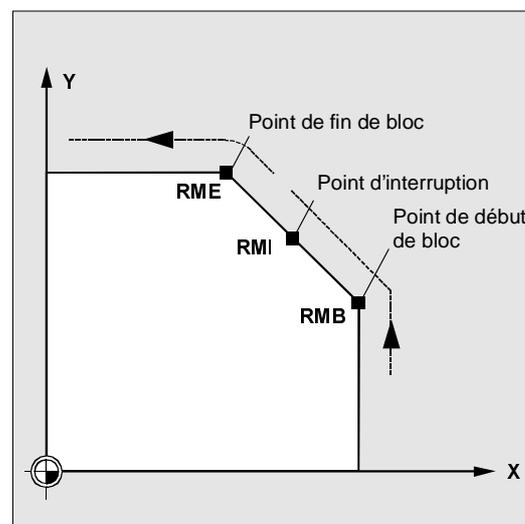
En fonction du bloc CN dans lequel l'interruption a eu lieu, vous avez le choix entre trois points de réaccostage :

- RMI, point d'interruption
- RMB, point de début de bloc ou dernier point final
- RME, point de fin de bloc

Avec RMI DISPR=..., vous pouvez définir un point de réaccostage du contour, situé entre le point de début de bloc et le point d'interruption.

Avec DISPR=..., vous décrivez en mm/inch la distance sur le contour du point de réaccostage **avant** le point d'interruption ou le point final. Ce point ne peut pas figurer au-delà du point de début de bloc – même pour les plus grandes valeurs.

Quand on ne programme pas DISPR=..., on a DISPR=0 et c'est le point d'interruption qui est point de réaccostage.





840 D
NCU 571



840 D
NCU 572
NCU 573

Accostage avec un nouvel outil

Si vous avez stoppé l'exécution du programme suite à un bris d'outil :

En programmant le nouveau numéro D, vous poursuivez l'exécution du programme à partir du point de réaccostage et avec des valeurs de correction d'outil modifiées.

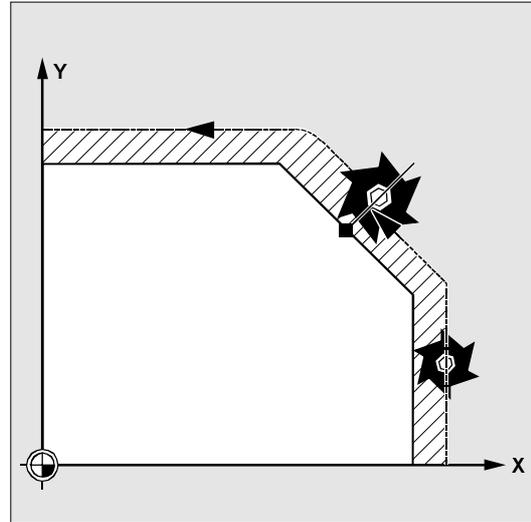
Suite aux différentes valeurs de correction d'outil, le point d'interruption risque de ne pas pouvoir être réaccosté. Dans ce cas, c'est le point le plus proche du point d'interruption qui devient point de réaccostage du contour (le cas échéant avec une modification DISPR).

Accostage du contour

Le déplacement à effectuer pour réaccoster le contour est programmable. Indiquez zéro pour les adresses des axes à déplacer.

Avec les instructions REPOSA, REPOSQA et REPOSHA, tous les axes sont repositionnés automatiquement. Il n'est pas nécessaire de préciser les axes.

Quand on programme REPOSL, REPOSQ ou REPOSH, tous les axes géométriques se déplacent automatiquement, autrement dit sans qu'il soit nécessaire de le préciser dans l'instruction. Tous les autres axes à repositionner sont à préciser dans l'instruction.





840 D

NCU 571



840 D

NCU 572

NCU 573

Accostage sur une droite, REPOSA, REPOSL

L'outil rallie le point de réaccostage sur une ligne droite.

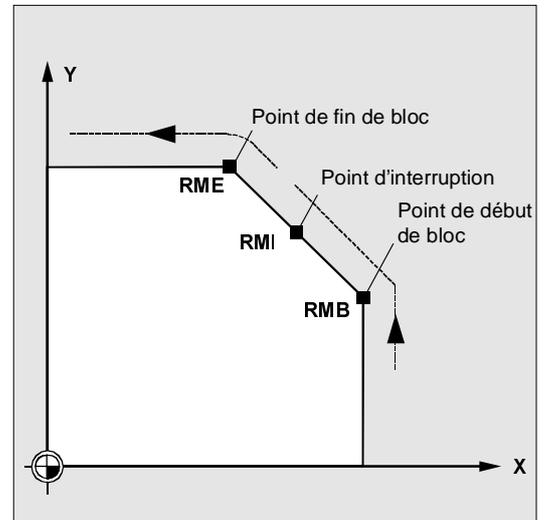
Avec REPOSA, tous les axes sont déplacés automatiquement. Avec REPOSL, vous pouvez préciser les axes à déplacer.

Exemple :

```
REPOSL RMI DISPR=6 F400
```

ou

```
REPOSA RMI DISPR=6 F400
```

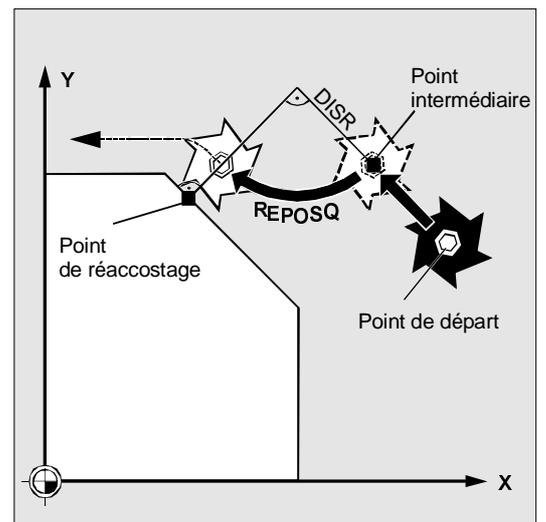


Accostage sur un quart de cercle, REPOSQ, REPOSA

L'outil rallie le point de réaccostage sur un quart de cercle de rayon $DISR=...$. La commande calcule automatiquement le point intermédiaire nécessaire entre le point de départ et le point de réaccostage.

Exemple :

```
REPOSQ RMI DISR=10 F400
```



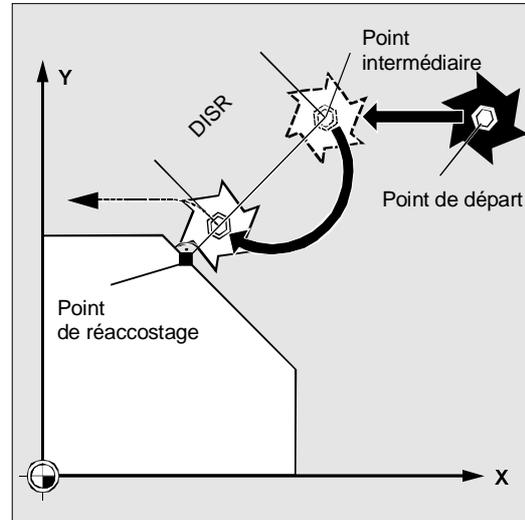
840 D
NCU 571840 D
NCU 572
NCU 573

Accostage sur un demi-cercle, REPOSH, REPOSHA

L'outil rallie le point de réaccostage sur un demi-cercle de diamètre $DISR=...$ La commande calcule automatiquement le point intermédiaire nécessaire entre le point de départ et le point de réaccostage.

Exemple :

```
REPOSH RMI DISR=20 F400
```



Dans le cas des trajectoires circulaires REPOSH et REPOSQ :

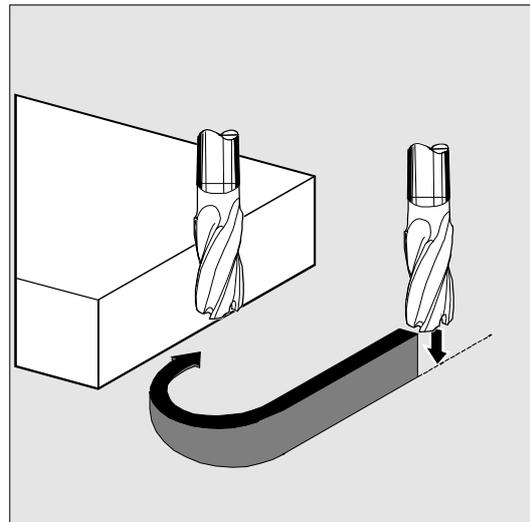
Le mouvement circulaire est décrit dans le plan de travail indiqué G17 à G19.

Si vous indiquez le troisième axe géométrique (pénétration radiale) dans le bloc d'accostage et si la position de l'outil et la position programmée dans la direction de pénétration radiale ne concordent pas, le réaccostage du contour se fera suivant un mouvement hélicoïdal.

Dans les cas suivants, il y a commutation automatique sur accostage linéaire REPOSL :

Vous n'avez indiqué aucune valeur pour DISR.

- Il n'existe aucun sens d'accostage défini (interruption de l'exécution du programme dans un bloc ne contenant aucune information de déplacement).
- Le sens d'accostage est perpendiculaire au plan de travail courant.



Action synchrone au déplacement

10.1 Structures, bases générales	10-309
10.1.1 Programmation et éléments d'instructions	10-310
10.1.2 Plage de validité : Numéro d'identification ID.....	10-312
10.1.3 Mot-clé	10-313
10.1.4 Actions	10-315
10.1.5 Vue d'ensemble des actions synchrones.....	10-317
10.2 Notions de base pour conditions et actions	10-319
10.3 Variables temps réel spéciales pour actions synchrones	10-322
10.3.1 Mémentos/compteurs \$AC_MARKER[n].....	10-322
10.3.2 Variable de temporisation \$AC_TIMER[n], à partir de SW 4.....	10-322
10.3.3 Paramètres d'actions synchrones \$AC_PARAM[n].....	10-323
10.3.4 Accès aux paramètres R \$Rxx	10-324
10.3.5 Ecriture/Lecture des paramètres machine et des données de réglage, à partir de SW 4	10-325
10.3.6 Variable FIFO \$AC_FIFO1[n] ... \$AC_FIFO10[n], à partir de SW 4	10-326
10.4 Actions dans actions synchrones.....	10-328
10.4.1 Sortie de fonctions auxiliaires	10-328
10.4.2 Activation du blocage de l'introduction via l'interface RDISABLE.....	10-329
10.4.3 Annuler l'arrêt du prétraitement des blocs STOPREOF	10-330
10.4.4 Effacement de la distance restant à parcourir	10-331
10.4.5 Effacement de la distance restant à parcourir, avec préparation, DELDTG, DELTG(axe1,..)	10-331
10.4.6 Effacement de la distance restant à parcourir, sans préparation, DELD, DELD (axe1, ...), à partir de SW 4.3.....	10-333
10.4.7 Définition d'un polynôme, FCTDEF, synchrone avec un bloc.....	10-334
10.4.8 Positionnement par laser	10-336
10.4.9 Fonction d'évaluation SYNFACT	10-337
10.4.10 Régulation AC (additive).....	10-338
10.4.11 Régulation AC (multiplicative).....	10-339
10.4.12 Régulation d'écartement ("de distance de tir") avec correction limitée.....	10-340
10.4.13 Correction d'outil en ligne FTOC.....	10-342
10.4.14 Déplacements de positionnement	10-344
10.4.15 Positionner l'axe : POS	10-344
10.4.16 Lancer/Arrêter l'axe : MOV	10-345
10.4.17 Avance axiale : AA.....	10-346
10.4.18 Fins de course logiciels.....	10-346
10.4.19 Coordination des axes	10-346
10.4.20 Préréglage des mémoires de valeurs réelles	10-348

10.4.21 Déplacement des broches	10-349
10.4.22 Déplacements conjugués : TRAILON, TRAILOF	10-350
10.4.23 Couplage par valeur pilote : LEADON, LEADOF	10-351
10.4.24 Mesure	10-353
10.4.25 Activer/Effacer les marques d'attente : SETM, CLEARM	10-353
10.4.26 Réactions en cas de défaut.....	10-354
10.5 Cycles technologiques.....	10-355
10.5.1 Bloquer, libérer, interrompre : LOCK, UNLOCK, RESET.....	10-357
10.6 Effacer une action synchrone : CANCEL	10-359
10.7 Conditions marginales.....	10-360



840 D
NCU 571



840 D
NCU 572



FM-NC
NCU 573



810D
CCU2

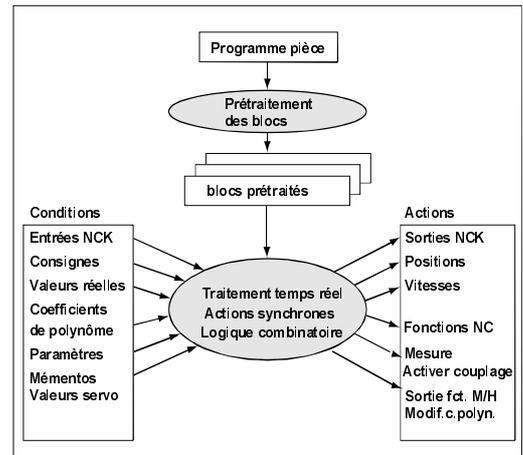
10.1 Structures, bases générales



Fonction

Les actions synchrones permettent, depuis le programme pièce courant, de lancer différentes actions et de les faire exécuter en synchronisation avec le programme.

On peut fixer le moment où ces actions doivent intervenir, en définissant des conditions qui sont traitées en temps réel (dans la période d'appel de l'interpolateur). Les actions constituent par conséquent une réaction à des événements en temps réel ; leur exécution n'est pas tributaire des limites de blocs. Par ailleurs, une action synchrone contient des indications sur la modalité des actions et sur la fréquence d'interrogation pour les variables temps réel programmées et, par conséquent, sur la fréquence d'exécution des actions à lancer. Ainsi, une action peut être lancée une fois seulement ou bien de façon cyclique (toujours dans la période d'appel de l'interpolateur).



Programmation

DO Action1 Action2 ...

MOT-CLE condition DO Action1 Action2 ...

ID=n MOT-CLE condition DO Action1 Action2 ...

IDS=n MOT-CLE condition DO Action1 Action2 ...



Signification

Numéro d'identification ID/IDS

ID=n Action synchrone à **effet modal** dans le mode automatique, **spécifique à un programme**; n = 1... 255

IDS=n Action synchrone à **effet modal** agissant dans tous les modes de fonctionnement, **statique**; n = 1... 255

sans ID/IDS Action synchrone à **effet non modal** dans le mode automatique

Mot-clé

pas de MOT-CLE L'exécution de l'action n'est liée à aucune condition. L'action est exécutée de façon cyclique dans la période d'appel de l'interpolateur.

10.1 Structures, bases générales

840 D
NCU 571840 D
NCU 572
NCU 573

FM-NC



810D

WHEN	La condition est vérifiée jusqu'à ce qu'elle soit remplie une première fois ; l'action correspondante est exécutée une fois.
WHENEVER	La condition est vérifiée de façon cyclique. L'action correspondante est exécutée de façon cyclique, aussi longtemps que la condition est remplie.
FROM	Dès que la condition est remplie pour la première fois, l'action est exécutée de façon cyclique aussi longtemps que l'action synchrone est activée.
EVERY	L'action est lancée une première fois quand la condition est remplie, puis elle est relancée dès que la condition passe de l'état FALSE à l'état TRUE. La condition est vérifiée de façon cyclique. A chaque fois qu'elle est remplie, l'action correspondante est exécutée une fois.
Condition	Logique de combinaison pour variables en temps réel ; la condition est vérifiée dans la période d'appel de l'interpolateur.
DO	Déclenche l'action quand la condition est remplie.
Action	Action lancée dès que la condition est remplie. Exemple : affecter des variables, activer le couplage axial, mettre à 1 les sorties NCK, sortir les fonctions M, H...
Coordination des actions synchrones/des cycles technologiques	
CANCEL[n]	Effacer une action synchrone
LOCK[n]	Bloquer le cycle technologique
UNLOCK[n]	Libérer le cycle technologique
Reset	Remettre le cycle technologique à zéro

**Exemple de programmation**

WHEN \$AA_IW[Q1]>5 DO M172 H510	Quand la position réelle de l'axe Q1 dépasse 5 mm, les fonctions auxiliaires M172 et H510 sont transférées à l'interface AP.
---------------------------------	--



Quand des variables temps réel apparaissent dans un programme pièce (par ex. valeur réelle, position d'une entrée ou sortie TOR, etc.), le prétraitement des blocs est stoppé jusqu'à ce que le bloc précédent soit entièrement traité et que les valeurs des variables temps réel soient connues.

Les variables temps réel utilisées sont évaluées dans la période d'appel de l'interpolateur.

Avantage des actions synchrones :

Il n'y a ici aucun arrêt du prétraitement des blocs.



840 D
NCU 571



840 D
NCU 572



FM-NC



810D

CCU2

Applications possibles:

- Optimisation des applications à durée critique (par ex. le changement d'outil)
- Réaction rapide à des événements externes
- Programmation de régulations AC
- Mise en place de fonctions de sécurité
-

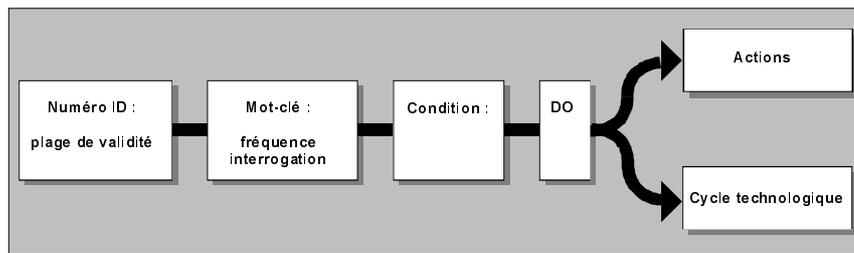
10.1.1 Programmation et éléments d'instructions



Fonction

Une action synchrone occupe un bloc à elle seule et agit dans le prochain bloc exécutable comportant une fonction machine (par ex. un déplacement avec G0, G1, G2, G3).

Les actions synchrones se composent d'éléments d'instructions au nombre de 5 maximum avec des tâches différentes :



Exemple :

ID=1	WHENEVER	\$A_IN[1]==1	DO	\$A_OUT[1]=1
------	----------	--------------	----	--------------

Action synchrone n° 1: A chaque fois que 1 est à l'entrée alors mettre sortie à 1

10.1 Structures, bases générales



840 D
NCU 571



840 D
NCU 572



FM-NC



810D
NCU 573

10.1.2 Plage de validité : Numéro d'identification ID



Fonction

La plage de validité d'une action synchrone est définie par le numéro d'identification (ID modal) :

- **aucun** ID modal

L'action synchrone agit uniquement dans le mode de fonctionnement automatique. Elle est valide uniquement pour le bloc exécutable suivant (bloc avec instruction de déplacement ou autre action machine), autrement dit elle est à effet **non modal**.

Exemple :

```
WHEN $A_IN[3]==TRUE DO $A_OUTA[4]=10
```

```
G1 X20
```

```
;bloc exécutable
```

- **ID=n**; n=1...255

L'action synchrone agit dans les blocs qui suivent, elle a donc un effet **modal**. Elle est désactivée avec CANCEL[n] ou par la programmation d'une autre action synchrone sous le même numéro ID.

Les actions synchrones en vigueur dans le bloc M30 restent actives (le cas échéant, les effacer avec l'instruction CANCEL).

Les actions synchrones ID agissent

uniquement dans le **mode automatique**.

Exemple :

```
ID=2 EVERY $A_IN[1]==1 DO POS[X]=0
```

- **IDS=n**; n=1...255

Ces actions synchrones **statiques** agissent avec un effet **modal** dans **tous les modes de fonctionnement**.

On peut les définir à partir d'un programme pièce, mais aussi directement après la mise sous tension, à partir d'un sous-programme asynchrone (ASUP) lancé par l'AP.

De cette façon, on peut activer des actions qui doivent être exécutées dans la CN, indépendamment du mode

840 D
NCU 571840 D
NCU 572
NCU 573

FM-NC



810D

CCU2

de fonctionnement.

Exemple :

```
IDS=1 EVERY $A_IN[1]==1 DO POS[X]=100
```



Application :

- Rectification AC en mode JOG
- Logique combinatoire pour Safety Integrated
- Fonctions de surveillance, réactions aux états machine dans tous les modes de fonctionnement.

Ordre d'exécution

Les actions synchrones à effet modal et statique sont traitées dans l'ordre de leur numéro ID(S) (dans la période d'appel de l'interpolateur).

Les actions synchrones à effet non modal (sans numéro ID) sont traitées dans l'ordre de programmation, après exécution des actions synchrones à effet modal.

10.1.3 Mot-clé



Fonction

Le mot-clé détermine le nombre de fois qu'il y aura interrogation de la condition et exécution de l'action correspondante :

- pas de mot-clé :
Quand aucun mot-clé n'est programmé, la condition est toujours considérée comme remplie. Les instructions synchrones sont exécutées cycliquement.
- WHEN
La condition est interrogée dans chaque période d'appel de l'interpolateur, jusqu'à ce qu'elle soit remplie une première fois et l'action est alors exécutée une fois.
- WHENEVER
La condition est interrogée dans chaque période d'appel de l'interpolateur. L'action est exécutée dans chaque période d'appel de l'interpolateur, aussi longtemps que la condition est remplie.

Exemple :

```
DO $A_OUTA[1]=$AA_IN[X]  
; Sortie valeur réelle sur sortie  
analogique
```

10.1 Structures, bases générales

840 D
NCU 571840 D
NCU 572

FM-NC

810D
NCU 573

- FROM

La condition est vérifiée dans chaque période d'appel de l'interpolateur jusqu'à ce qu'elle soit remplie une première fois. L'action est ensuite exécutée aussi longtemps que l'action synchrone est activée, autrement dit même si la condition n'est plus remplie.

- EVERY

La condition est interrogée dans chaque période d'appel de l'interpolateur. L'action est toujours exécutée une seule fois, quand la condition est remplie.

Commande par changement d'état :

L'action sera exécutée une nouvelle fois dès que la condition passe de l'état FALSE à l'état TRUE.

Exemple :

```
ID=1 EVERY $AA_IM[B]>75 DO
POS[U]=IC(10) FA[U]=900;
```

quand la valeur réelle de l'axe B dans le SCM dépasse la valeur 75, l'axe U doit poursuivre le positionnement de 10 avec avance axiale.

Condition

En comparant deux variables temps réel ou une variable temps réel et une expression calculée dans le prétraitement des blocs, la condition vérifie si une action doit être exécutée.

A partir du logiciel SW 4 :

Dans la condition, on peut combiner des résultats de comparaisons avec des opérateurs booléens ().

La condition est vérifiée dans la période d'appel de l'interpolateur. Si elle est remplie, l'action correspondante sera exécutée.



Exemple de programmation

WHENEVER \$AA_IM[X] > 10.5*SIN(45) DO ...	Comparaison avec une expression calculée dans le prétraitement des blocs
WHENEVER \$AA_IM[X] > \$AA_IM[X1] DO ...	Comparaison avec d'autres variables temps réel
WHENEVER (\$A_IN[1]==1) OR (\$A_IN[3]==0) DO ...	Deux comparaisons combinées ensemble

840 D
NCU 571840 D
NCU 572

FM-NC

810D
CCU2

NCU 573

CCU2



Conditions possibles :

- Comparaison de variables temps réel (E/S analogiques/TOR par ex.)
- Combinaison logique booléenne entre des résultats de comparaisons
- Calcul d'expressions temps réel
- Temporisation/Distance du début du bloc
- Distance de la fin du bloc
- Valeurs mesurées, résultats des mesures
- Valeurs Servo
- Vitesses, état des axes

10.1.4 Actions



Fonction

Dans chaque action synchrone, vous pouvez programmer une ou plusieurs actions. Toutes les actions programmées dans un bloc sont lancées dans la même période d'appel de l'interpolateur.

Actions possibles :

- Affecter des variables
- Ecrire des données de réglage
- Régler des paramètres
- DELDTG : Effacement rapide de la distance restant à parcourir
- RDISABLE : Mettre en place un blocage de l'introduction via l'interface
- Sortie des fonctions auxiliaires M, S et H
- STOPREOF : Annulation de l'arrêt du prétraitement des blocs
- FTOC : Correcteur d'outil en ligne
- Définir des fonctions de traitement (polynômes)
- SYNFACT : Activer des fonctions d'évaluation : Régulation AC
- Commuter entre plusieurs avances dans un bloc programmé, en fonction des signaux binaires ou analogiques.
- Corriger l'avance
- Lancer/Positionner/Arrêter des axes de positionnement (POS) et des broches (SPOS)
- PRESETON : Prérégler les mémoires de valeurs réelles

10.1 Structures, bases générales



840 D
NCU 571



840 D
NCU 572



FM-NC



810D
NCU 573

- Activer/Désactiver les déplacements conjugués/
le couplage par valeur pilote
- mesure
- Mettre en oeuvre des fonctions de sécurité
supplémentaires
- Sortie de signaux numériques et analogiques
- ...



Exemple de programmation

Action synchrone avec deux actions

```
WHEN $AA_IM[Y] >= 35.7 DO M135 $AC_PARAM=50
```

Si la condition est remplie, M135 sera délivré à l'AP et la correction par commutateur sera mise à 50%.



Une action peut aussi être constituée par un programme (programme à un seul axe, cycle technologique). Celui-ci ne doit comporter que des actions, qui peuvent aussi être programmées individuellement en actions synchrones. Les différentes actions d'un tel programme sont traitées de façon séquentielle dans la période d'appel de l'interpolateur.



Remarque

Les actions sont exécutables indépendamment du mode de fonctionnement.

Les actions suivantes sont actives uniquement en mode automatique quand le programme est en service :

- STOPREOF,
- DELDTG.

840 D
NCU 571840 D
NCU 572
NCU 573

FM-NC

810D
CCU2

10.1.5 Vue d'ensemble des actions synchrones

Jusqu'au logiciel SW 3.x

- Programmation des séquences dans la période d'appel de l'interpolateur au niveau utilisateur (programme pièce).
- Réaction à des événements / à des états dans la période d'appel de l'interpolateur
- Logique combinatoire en temps réel
- Accès à la périphérie, aux états de la commande et de la machine
- Programmation de procédures cycliques exécutées dans la période d'appel de l'interpolateur
- Déclenchement de fonctions CN spécifiques (blocage de l'introduction via l'interface, déplacement axial forcé, ..)
- Exécution de fonctions technologiques en parallèle avec l'interpolation
- Déclenchement de fonctions technologiques indépendamment des limites de blocs

A partir du logiciel SW 4

- Possibilités de diagnostic pour actions synchrones
- Extension de la variable d'exécution utilisable dans les actions synchrones
- Conditions complexes dans des actions synchrones
- Extension des expressions dans des actions synchrones :
Combinaison logique de variables temps réel avec des fonctions de calcul de base et des fonctions dans la période d'appel de l'interpolateur, adressage indirect de variables d'exécution modifiable en ligne avec indice, données de réglage modifiables et exploitables en ligne depuis les actions synchrones
- Personnalisation de la commande : réglage par le biais des paramètres machine du nombre d'actions synchrones à activer simultanément.



840 D 840 D FM-NC 810D
NCU 571 NCU 572
 NCU 573

- Lancer le déplacement des axes de positionnement et des broches à partir d'actions synchrones (axes de commande)
- Preset à partir d'actions synchrones
- Mise en marche/Arrêt, paramétrage des couplages axiaux : Couplage par valeur pilote, déplacements conjugués
- Mise en marche/Arrêt de la fonction de mesure axiale
- Cames logicielles
- Effacement de la distance restant à parcourir, sans arrêt du prétraitement des blocs
- Programmes à axe unique, cycles technologiques
- Actions synchrones actives en mode de fonctionnement JOG, au-delà des limites de programme
- Actions synchrones influençables par l'AP
- Actions synchrones protégées
- Extension pour déplacement forcé / régulation des distances

840 D
NCU 571840 D
NCU 572

FM-NC



810D

CCU2

10.2 Notions de base pour conditions et actions



Variable temps réel

Les variables temps réel sont traitées et écrites dans la période d'appel de l'interpolateur.

Les variables temps réel sont :

- \$A... , variable d'exécution,
- \$V... , variable servo (asservissement de position)

Pour les caractériser, on peut programmer ces variables dans les actions synchrones avec \$\$:

\$AA_IM[X] est équivalent à \$\$AA_IM[X].

Les données de réglage et les paramètres machine doivent être caractérisés par \$\$ pour que le traitement/l'affectation se fassent dans la période d'appel de l'interpolateur.



Vous trouverez la liste des variables en annexe.



Calculs en temps réel

Les calculs en temps réels sont limités aux types de données INT, REAL et BOOL.

Les expressions temps réel sont des calculs exécutables dans la période d'appel de l'interpolateur; elles peuvent être utilisées dans les conditions et dans les actions pour les affectations à des adresses CN et à des variables.

- **Comparaisons**

Dans des conditions, on peut comparer des variables ou des expressions partielles de même type. Le résultat est toujours du type BOOL.

Tous les opérateurs relationnels connus sont admis (==, <>, <, >, <=, >=).

- **Opérateurs booléens**

Variables, constantes ou comparaisons peuvent être mises en relation réciproque avec les opérateurs booléens connus (NOT, AND, OR, XOR)

10.2 Notions de base pour conditions et actions



840 D
NCU 571



840 D
NCU 572



FM-NC



810D
NCU 573

- **Opérateurs binaires**

Les opérateurs binaires sont possibles : B_NOT, B_AND, B_OR, B_XOR.

Les opérandes sont des variables ou des constantes de type INTEGER.

- **Opérations élémentaires**

Les variables temps réel de type INTEGER et REAL peuvent être combinées ensemble ou avec des constantes par le biais des opérations élémentaires (+, -, *, /, DIV, MOD).

- **Fonctions mathématiques**

Des fonctions mathématiques peuvent être appliquées aux variables temps réel de type REAL (SIN, COS, TAN, ASIN, ACOS, ABS, TRUNC, ROUND, LN, EXP, ATAN2, ATAN, POT, SQRT, CTAB, CTABINV).

Exemple :

```
DO $AC_PARAM[ 3 ] = COS ($AC_PARAM[1])
```

840 D
NCU 571840 D
NCU 572
NCU 573

FM-NC



810D

CCU2



Remarques

On ne peut combiner que des variables de même type.

Vrai : \$R10=\$AC_PARAM[1]

Faux : \$R10=\$AC_MARKER[1]

La multiplication est prioritaire sur la division, les parenthèses sont admises.

Les opérateurs DIV et MOD sont aussi admis pour les données de type REAL (à partir du logiciel SW 4).

Exemple :

```
DO $AC_PARAM[3] = $A_INA[1] - $AA_IM[Z1] ;soustraire deux variables temps réel
```

```
WHENEVER $AA_IM[x2] < $AA_IM[x1] - 1.9 DO $A_OUT[5] = 1
```

;soustraire une constante d'une variable temps réel

```
DO $AC_PARAM[3] = $INA[1] - 4 * SIN(45.7 $P_EP[Y]) * R4
```

;Expression constante, calculée dans le prétraitement des blocs

- **Affecter un indice**

Des variables temps réel peuvent se voir affecter des variables temps réel comme indice.



Remarques

Des variables qui n'ont pas été formées en temps réel, on ne doivent pas se voir affecter des variables temps réel comme indices.

Exemple :

```
WHEN...DO $AC_PARAM[$AC_MARKER[1]] = 3
```

N'est pas admis :

```
$AC_PARAM[1] = $P_EP[$AC_MARKER]
```



Exemple de programmation

Exemples d'expressions temps réel

```
ID=1 WHENEVER ($AA_IM[Y]>30) AND ($AA_IM[Y]<40)
```

```
DO $AA_OVR[S1]=80
```

Sélection d'une fenêtre de positionnement

```
ID=67 DO $A_OUT[1]=$A_IN[2] XOR $AN_MARKER[1]
```

Exploiter 2 signaux booléens

```
ID=89 DO $A_OUT[4]=$A_IN[1] OR ($AA_IM[Y]>10)
```

Sortir le résultat d'une comparaison

10.3 Variables temps réel spéciales pour actions synchrones



840 D
NCU 572
NCU 573



810 D
CCU2

10.3 Variables temps réel spéciales pour actions synchrones

Les variables temps réel énumérées ci-après sont admises dans les actions synchrones :

10.3.1 Mémentos/compteurs \$AC_MARKER[n]



Fonction

Les variables mémentos peuvent être lues et écrites dans des actions synchrones.

Mémentos/compteurs spécifiques à un canal

\$AC_MARKER[n]

Type de donnée : INTEGER

Une variable memento spécifique à un canal ne figure qu'une fois sous son nom dans le canal.

Exemple :

```
WHEN ... DO $AC_MARKER[0] = 2
```

```
WHEN ... DO $AC_MARKER[0] = 3
```

```
WHEN $AC_MARKER == 3 DO $AC_OVR=50
```

10.3.2 Variable de temporisation \$AC_TIMER[n], à partir de SW 4



Fonction

La variable système \$AC_TIMER[n] permet de lancer des actions après écoulement d'un temps d'attente défini.

Type de donnée : REAL

Unité : s

n : numéro de la variable de temporisation

- **Régler la temporisation**

L'incrémementation d'une variable de temporisation est lancée par l'affectation d'une valeur

\$AC_TIMER[n]=valeur

n : numéro de la variable de temporisation

valeur : valeur de départ (généralement 0)

10.3 Variables temps réel spéciales pour actions synchrones



840 D
NCU 571



840 D
NCU 572



FM-NC



810D

NCU 573

- **Arrêter la temporisation**

L'incrémentation d'une variable de temporisation est stoppée par l'affectation d'une valeur négative $\$AC_TIMER[n]=-1$

- **Lire la temporisation**

La valeur actuelle peut être lue que la variable de temporisation soit en activation ou arrêtée. Dès que la variable de temporisation a été stoppée par l'affectation de -1, la dernière valeur actualisée se fige et reste accessible à la lecture.

Exemple :

Sortie d'une valeur réelle à la sortie analogique
500 ms après la détection d'une entrée TOR

```
WHEN $A_IN[1] == 1 DO $AC_TEMPO[1]=0 ; remettre la temporisation à zéro et relancer
WHEN $AC_TIMER[1]>=0.5 DO $A_OUTA[3]=$AA_IM[X] $AC_TIMER[1]=-1
```

10.3.3 Paramètres d'actions synchrones \$AC_PARAM[n]



Fonction

Type de donnée : REAL

n : Numéro du paramètre 0-n

Les paramètres d'actions synchrones

$\$AC_PARAM[n]$ servent aux calculs et comme mémoires tampons dans des actions synchrones.

Le nombre de ces variables AC disponibles par canal est défini dans le PM 28254 :

MM_NUM_AC_PARAM.

Les paramètres ne figurent qu'une fois dans un canal sous le même nom. Les mementos $\$AC_PARAM$ sont conservés dans la mémoire dynamique.

10.3 Variables temps réel spéciales pour actions synchrones

840 D
NCU 571840 D
NCU 572

FM-NC

810 D
NCU 573

10.3.4 Accès aux paramètres R \$Rxx



Fonction

Type de donnée : REAL

Ces variables statiques sont généralement utilisées pour les calculs dans le programme pièce. Dans la période d'appel de l'interpolateur, on peut accéder à ces variables en ajoutant \$.

Exemples :

WHEN \$AA_IM[X]>=40.5 DO \$R10=\$AA_MM[Y]	Accès en lecture au paramètre R 10.
WHEN \$AA_IM[X]>=6.7 DO \$R[\$AC_MARKER[1]]=30.6	; Accès en lecture au paramètre R dont le numéro figure dans le mémento 1.



Remarques

Application

L'utilisation de paramètres R dans des actions synchrones permet

- Mise en mémoire des valeurs qui doivent être sauvegardées après la fin d'un programme, après un reset CN ou après un Power On.
- l'affichage des valeurs mémorisées dans l'image du paramètre R
- l'archivage des valeurs déterminées dans les actions synchrones

Le paramètre R doit être utilisé soit comme variable de calcul "normale" Rxx **ou** comme variable temps réel \$Rxx.

Si, après avoir utilisé le paramètre R dans une action synchrone, vous voulez le réutiliser comme variable de calcul "normale", vous devez l'exprimer de façon explicite dans la programmation de l'arrêt du prétraitement des blocs avec STOPRE, pour synchroniser le prétraitement des blocs et leur exécution.

Exemple :

WHEN \$AA_IM[X]>=40.5 DO \$R10=\$AA_MM[Y]	Utilisation de R10 dans action synchrone
G01 X500 Y70 F1000	
STOPRE	Arrêt du prétraitement des blocs
IF R10>20	Evaluation de la variable de calcul

10.3 Variables temps réel spéciales pour actions synchrones



840 D
NCU 572
NCU 573



810D
CCU2

10.3.5 Ecriture/Lecture des paramètres machine et des données de réglage, à partir de SW 4



Fonction

A partir du logiciel SW 4, la lecture et l'écriture des paramètres machine et des données de réglage (PM, SD) sont possibles depuis des actions synchrones.

- **Lecture des données PM, SD non modifiables**

Depuis les actions synchrones, on les sollicite comme dans les instructions normales du programme pièce en commençant par le caractère \$.

Exemple :

```
ID=2 WHENEVER $AA_IW[z]<$SA_OSCILL_REVERSE_POS2[Z]-6 DO $AA_OVR[X]=0
```

;ici on sollicite la plage d'inversion 2 pour l'oscillation, considérée comme étant non modifiable.

- **Lecture des données PM, SD modifiables**

Depuis l'action synchrone, ces données sont sollicitées en commençant par \$\$ et traitées dans la période d'appel de l'interpolateur.

Exemple :

```
ID=1 WHENEVER $AA_IW[z]<$$SA_OSCILL_REVERSE_POS2[Z]-6 DO $AA_OVR[X]=0
```

;ici on suppose qu'on pourrait intervenir pour modifier la position d'inversion pendant l'usinage.

- **Ecriture de PM, SD**

Condition :

le droit d'accès en vigueur doit autoriser l'accès en écriture. La modification d'un PM ou d'un SD à partir d'une action synchrone n'a de sens que si elle devient **immédiatement** efficace. L'efficacité des PM et SD est précisée dans la **bibliographie** :

/LIS/, Listes

Adressage :

Les PM et SD à modifier sont à solliciter en commençant par \$\$.

Exemple :

```
ID=1 WHEN $AA_IW[X]>10 DO $$SN_SW_CAM_PLUS_POS_TAB_1[0]=20
```

```
$$SN_SW_CAM_MINUS_POS_TAB_1[0]=30
```

;modification de la position de basculement des cames logicielles. Remarque : Les positions de basculement sont à modifier 2 à 3 périodes d'appel de l'interpolateur avant d'atteindre la position.

10.3 Variables temps réel spéciales pour actions synchrones



840 D
NCU 572
NCU 573



810 D
CCU2

10.3.6 Variable FIFO \$AC_FIFO1[n] ... \$AC_FIFO10[n], à partir de SW 4



Fonction

Type de donnée : REAL

Pour mémoriser des suites de données connexes, vous avez 10 variables FIFO à disposition (mémoire à défilement).

Application :

- Mesure cyclique
- Usinage en série

Chaque élément est accessible en lecture et en écriture.

Le nombre des variables FIFO disponibles est défini dans le PM 28260: NUM_AC_FIFO.

Le nombre de valeurs pouvant être écrites dans une variable FIFO est défini dans le PM

PM 28264: LEN_AC_FIFO. Les variables FIFO ont toutes la même longueur.

Les indices 0 à 5 ont une signification particulière :

n=0 : A l'écriture : la nouvelle valeur est rangée dans le FIFO

A la lecture : l'élément le plus ancien est lu et extrait de FIFO

n=1 : Accès à l'élément le plus ancien

n=2 : Accès à l'élément le plus récent

n=3 : Somme de tous les éléments FIFO

n=4 : Nombre d'éléments contenus dans FIFO.

Chaque élément de FIFO est accessible en lecture et en écriture.

La remise à zéro des variables FIFO se fait par remise à zéro du nombre d'éléments, par ex. pour la première variable FIFO :

AC_FIFO1[4]=0

n=5 : Index d'écriture courant par rapport au début de FIFO

n=6 à 6+n_{maxi} :

Accès au n-ième élément FIFO



840 D
NCU 572
NCU 573



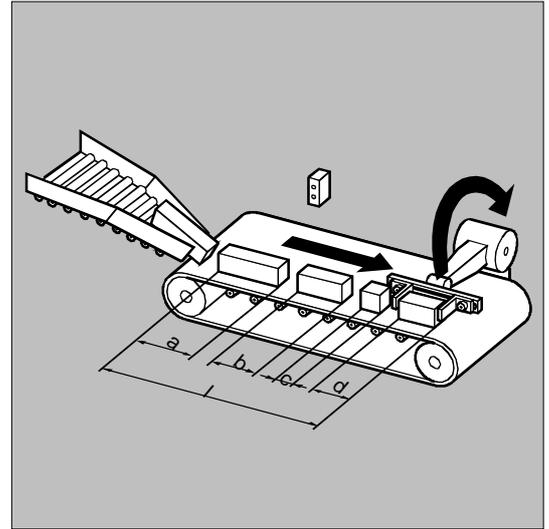
810D
CCU2



Exemple de programmation

Mémoire à défilement

En cours de production, on utilise un tapis roulant pour convoyer des produits de différentes longueurs (a, b, c, d). Par conséquent, sur le tapis roulant d'une longueur de convoyage „l“, le nombre des produits transportés varie par conséquent en fonction de leur longueur. Finalement, pour une vitesse constante du tapis roulant, il conviendra d'adapter le prélèvement des produits sur le tapis aux temps d'arrivée variables de ces produits.



```
DEF REAL ECART=2.5
```

Ecart constant entre les produits déposés.

```
DEF REAL TOTAL=270
```

Ecart entre la position de mesure de longueur et la position de prélèvement

```
EVERY $A_IN[1]==1 DO $AC_FIFO1[4]=0
```

Au début du process, remettre FIFO à zéro.

```
EVERY $A_IN[2]==1 DO $AC_TEMPO[0]=0
```

Dès qu'un produit passe la barrière lumineuse, lancer le chronométrage.

```
EVERY $A_IN[2]==0 DO $AC_FIFO1[0]=$AC_TIMER[0]*$AA_VACTM[B]
```

;dès que la barrière lumineuse est rétablie, calculer à partir de la vitesse du tapis roulant, la longueur du produit et la mémoriser dans FIFO.

```
EVERY $AC_FIFO1[3]+$AC_FIFO1[4]*ZWI>=TOTAL DO POS[Y]=-30
```

```
$R1=$AC_FIFO1[0]
```

;dès que la somme des longueurs des produits et des intervalles entre les produits est supérieure/égale à la longueur entre la position de dépose et la position de prélèvement, prélever le produit sur le tapis à la position de prélèvement et mémoriser sa longueur dans FIFO.

10.4 Actions dans actions synchrones



840 D
NCU 571



840 D
NCU 572



FM-NC



810 D
CCU2

10.4 Actions dans actions synchrones

10.4.1 Sortie de fonctions auxiliaires



Fonction

Quand les conditions sont remplies, au maximum 10 fonctions M, H ou H peuvent sortir par bloc d'usinage. La sortie de la fonction auxiliaire est amorcée avec le mot „DO“.

La sortie des fonctions d'aide a lieu **immédiatement** dans la période d'appel de l'interpolateur. L'instant de sortie de la fonction auxiliaire, rangé dans un paramètre machine, n'est pas pris en compte. Le moment de la sortie est déterminé dès que la condition est remplie.

Exemple :

Amorcer le liquide d'arrosage à une position d'axe bien définie :

```
WHEN $AA_IM[X]>=15 DO M07 G1 X20
F250
```



Procédure

Les fonctions auxiliaires sont à programmer dans les actions synchrones non modales (sans ID modal) uniquement avec les mots-clés WHEN ou EVERY. L'efficacité d'une fonction auxiliaire est déterminée par l'AP, par ex. avec Départ programme.



Remarques

Ne sont pas possibles à partir d'une action synchrone de déplacement :

- M0, M1, M2, M17, M30 : Arrêt/Fin du programme (M2, M17, M30 sont possibles pour les cycles technologiques)
- M70: Fonction broche
- M6 ou des fonctions M réglées par le biais de paramètres machine pour le changement d'outil
- M40, M41, M42, M43, M44, M45: changement de vitesse



Exemple de programmation

```
WHEN $AA_IW[Q1]>5 DO M172 H510
```

Lorsque la position réelle de l'axe Q1 dépasse 5 mm, les fonctions auxiliaires M172 et H510 sont transférées à l'interface AP.

840 D
NCU 571840 D
NCU 572
NCU 573

FM-NC



810D

CCU2

10.4.2 Activation du blocage de l'introduction via l'interface RDISABLE



Fonction

Quand la condition est remplie, RDISABLE bloque la poursuite du traitement du bloc dans le programme principal. Les actions synchrones au déplacement, qui sont programmées, continuent à être traitées et les blocs suivants continuent à être prétraités.

Au début du bloc avec RDISABLE, il y a toujours déclenchement d'un arrêt précis, que RDISABLE soit actif ou non.



Exemple de programmation

Lancer le programme dans la période d'appel de l'interpolateur, en fonction des entrées externes.

...

WHENEVER \$A_INA[2]<7000 DO RDISABLE

;si à l'entrée 2, la tension est inférieure à 7V,
bloquer la poursuite du programme
(1000= 1V).

N10 G1 X10

;si la condition est remplie, le blocage de
l'introduction via l'interface entre en vigueur à
la fin de N10

N20 G1 X10 Y20

...

10.4 Actions dans actions synchrones



840 D
NCU 571



840 D
NCU 572



FM-NC



810 D
CCU2

10.4.3 Annuler l'arrêt du prétraitement des blocs STOPREOF



Fonction

Quand le prétraitement de blocs a été arrêté par programmation explicite de STOPRE ou par activation implicite par le biais d'une action synchrone active, l'instruction STOPREOF annule cet arrêt du prétraitement des blocs après le prochain bloc d'usinage et ce, dès que la condition est remplie.



Remarques

STOPREOF doit être programmé avec le mot-clé WHEN et de façon non modale (sans numéro ID).



Exemple de programmation

Branchement rapide dans le programme en fin de bloc.

WHEN \$AC_DTEB<5 DO STOPREOF	;lorsque la distance par rapport au point de fin du bloc est < à 5 mm, annuler l'arrêt du prétraitement des blocs
G01 X100	;après exécution de l'interpolation linéaire, annulation de l'arrêt du prétraitement de blocs
IF \$A_INA[7]>500 GOTOF MARQUE1=X100	;quand à l'entrée 7 la tension dépasse 5V, sauter à l'étiquette 1.

840 D
NCU 571840 D
NCU 572
NCU 573

FM-NC

810D
CCU2

10.4.4 Effacement de la distance restant à parcourir

Un effacement de la distance restant à parcourir peut être déclenché pour la trajectoire et les axes indiqués, en fonction d'une condition.

A disposition :

- Effacement rapide de la distance restant à parcourir, avec préparation
- Effacement de la distance restant à parcourir, sans préparation (à partir de SW 4.3)

10.4.5 Effacement de la distance restant à parcourir, avec préparation, DELDTG, DELTG(axe1,..)



Fonction

L'effacement de la distance restant à parcourir, avec préparation, déclenché avec DELTDG, permet de réagir très rapidement à l'événement déclencheur. Il est utilisé par conséquent dans des applications critiques en temps, par exemple quand

- le temps entre l'effacement de la distance restant à parcourir et le lancement du bloc suivant doit être extrêmement court.
- la condition nécessaire pour l'effacement de la distance restant à parcourir est remplie avec une très forte probabilité.



Procédure

A la fin du bloc de déplacement, dans lequel on a déclenché un effacement de la distance restant à parcourir avec préparation, un arrêt du prétraitement des blocs est activé de façon implicite.

Ainsi, le déplacement d'interpolation et les déplacements d'axes de positionnement sont stoppés rapidement en fin de bloc.

La distance restant à parcourir est activable par le biais de la variable système \$AC_DELT ou \$AC_DELT[axe].

10.4 Actions dans actions synchrones

840 D
NCU 571840 D
NCU 572

FM-NC

810 D
NCU 573

CCU2



Exemple de programmation

Effacement rapide de la distance restant à parcourir sur la trajectoire

```
WHEN $A_IN[1]==1 DO DELDTG
```

```
N100 G01 X100 Y100 F1000 ; dès que l'entrée est mise à 1, le déplacement est suspendu
```

```
N110 G01 X...
```

```
IF $AC_DELT>50...
```



Exemple de programmation

Effacement rapide de la distance axiale restant à parcourir

```
POS[X1]=100 G1 Z100 F1000
```

Arrêter un déplacement de positionnement programmé :

```
ID=1 WHEN $A_IN[1]==1 DO MOV[V]=3 FA[V]=700 Lancer l'axe
```

```
WHEN $A_IN[2]==1 DO DELDTG(V) Effacer la distance restant à parcourir, l'arrêt de l'axe se fait avec MOV=0
```

Effacer la distance restant à parcourir en fonction de la tension à l'entrée :

```
WHEN $A_INA[5]>8000 DO DELDTG(X1)
```

```
;dès qu'à l'entrée 5 la tension dépasse 8V, effacement de la distance restant à parcourir de l'axe X1. Le déplacement d'interpolation se poursuit.
```



Restriction

L'effacement de la distance restant à parcourir, avec préparation,

- ne peut pas être appliqué quand la correction du rayon d'outil est active.
- ne doit être programmé que dans des actions synchrones non modales (sans numéro ID).



840 D
NCU 572
NCU 573



810D
CCU2

10.4.6 Effacement de la distance restant à parcourir, sans préparation, DELD, DELD(axe1, ...), à partir de SW 4.3



Fonction

L'effacement de la distance restant à parcourir, sans préparation, prend plus de temps pour être exécuté en réaction à un événement déclencheur. Il est préférable de l'utiliser quand la condition pour l'effacement de la distance restant à parcourir est très rarement remplie.



Procédure

L'effacement de la distance restant à parcourir, sans préparation, est possible dans actions synchrones modales ou non modales. Le contournage n'est pas interrompu.

Un long temps d'attente fait suite au déclenchement de l'effacement de la distance restant à parcourir, sans préparation.



Exemple de programmation

```
ID=1 EVERY ($A_IN[1]==1) AND $AA_IM[Y]>45.6 DO DELD
```

```
G01 X100 Y250 F1000
```

```
G02 ... ;l'effacement de la distance restant à parcourir reste actif
jusqu'à ce qu'il soit désactivé explicitement
```

10.4 Actions dans actions synchrones

840 D
NCU 571840 D
NCU 572

FM-NC

810 D
NCU 573

10.4.7 Définition d'un polynôme, FCTDEF, synchrone avec un bloc



Programmation

FCTDEF(N° de polynôme, LLIMIT, ULIMIT, a_0, a_1, a_2, a_3)



Signification

N° du polynôme	Numéro du polynôme de 3e degré
LLIMIT	limite inférieure de la valeur de la fonction
ULIMIT	limite supérieure de la valeur de la fonction
a_0, a_1, a_2, a_3	coefficients polynomiaux



Fonction

Avec FCTDEF, on peut définir des polynômes du 3ème degré de la forme $y=a_0+a_1 \cdot x+a_2 \cdot x^2+a_3 \cdot x^3$. Ces polynômes sont utilisés par la correction d'outil en ligne FTOC et par la fonction d'évaluation SYNFACT pour le calcul des valeurs de la fonction, à partir des variables d'exécution (variables temps réel).

Les polynômes sont définis avec la fonction FCTDEF en synchronisation avec les blocs ou par le biais de variables système :

\$AC_FCTLL[n]	limite inférieure de la valeur de la fonction
\$AC_FCTUL[n]	limite supérieure de la valeur de la fonction
\$AC_FCT0[n]	a_0
\$AC_FCT1[n]	a_1
\$AC_FCT2[n]	a_2
\$AC_FCT3[n]	a_3
n	numéro du polynôme

840 D
NCU 571840 D
NCU 572

FM-NC

810 D
NCU 573

Remarques

- Les variables système peuvent être écrites depuis le programme pièce ou à partir d'une action synchrone. Pour l'écriture depuis le programme pièce, il convient de programmer STOPRE pour faire la synchronisation avec les blocs.
- A partir de SW 4 :
Les variables système \$AC_FCTLL[n], \$AC_FCTUL[n], \$AC_FCT0[n] à \$AC_FCTn[n] sont modifiables depuis les actions synchrones (sauf SINUMERIK FM-NC et SINUMERIK 840D avec NCU 571).

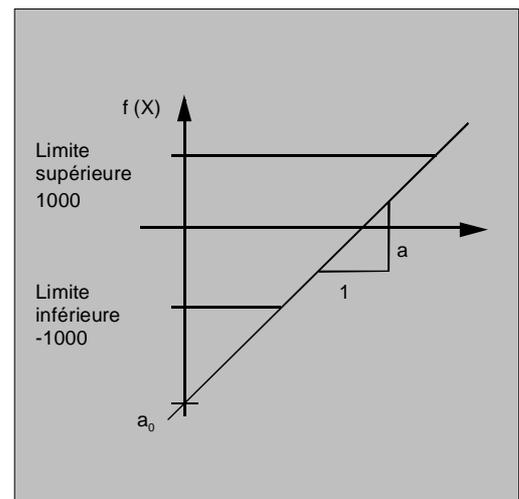
Dans le cas d'une écriture à partir d'actions synchrones, les coefficients polynomiaux et les valeurs limites de la fonction sont actifs immédiatement.



Exemple de programmation

Polynôme pour une section de droite :

Avec une limite supérieure 1000, une limite inférieure -1000, la section d'ordonnée $a_0 = \$AA_IM[X]$ et la pente de droite 1, la définition du polynôme est la suivante :



```
FCTDEF(1, -1000,1000,$AA_IM[X],1)
```

10.4 Actions dans actions synchrones



840 D
NCU 572
NCU 573

10.4.8 Positionnement par laser

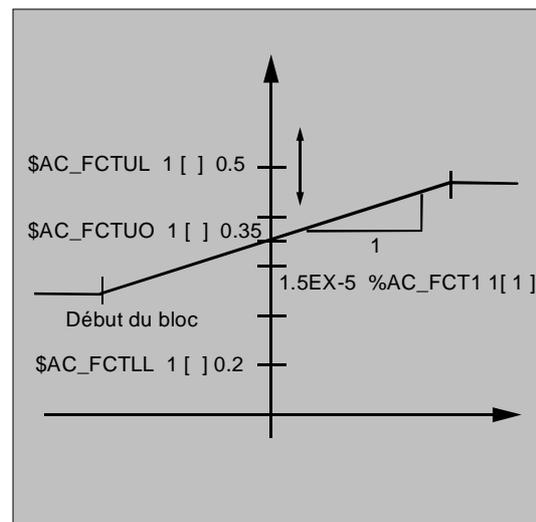


Exemple de programmation

Définition d'un polynôme par le biais d'une variable

L'une des applications possibles de la définition du polynôme est le positionnement par laser.

Le positionnement par laser signifie :
corriger une sortie analogique en fonction par
exemple de la vitesse tangentielle.



```
$AC_FCTLL[1]=0.2
```

définition des coefficients polynomiaux

```
$AC_FCTUL[1]=0.5
```

```
$AC_FCTO[1]=0.35
```

```
$AC_FCT1[1]=1.5EX-5
```

```
STOPRE
```

```
ID=1 DO $AC_FCTUL[1]=$A_INA[2]*0.1 +0.35
```

variation en ligne de la limite supérieure

```
ID=2 DO SYNFACT(1,$A_OUTA[1],$AC_VACTW)
```

;en fonction de la vitesse tangentielle (rangée dans \$AC_VACTW), le positionnement par laser est commandé par le biais de la sortie analogique 1.



Remarque

L'utilisation du polynôme défini plus haut se fait avec SYNFACT.



840 D
NCU 572
NCU 573

10.4.9 Fonction d'évaluation SYNFACT



Programmation

SYNFCT(n° polynôme, sortie variable temps réel, entrée variable temps réel)



Signification

N° du polynôme	Polynôme défini avec FCTDEF (voir "Définition des polynômes")
Sortie variable temps réel	Ecriture de la variable temps réel
Entrée variable temps réel	Lecture de la variable temps réel



Fonction

SYNFCT lit, en synchronisme avec l'usinage, des variables temps réel (par ex. entrée analogique, valeur réelle, ...) et calcule, en utilisant un polynôme d'évaluation du 3e degré maximum (FCTDEF), des valeurs de la fonction (par ex. correction de la vitesse de rotation, vitesse, position axiale, ...). Les résultats sont transmis à des variables temps réel et limités vers le haut et vers le bas avec FCTDEF (voir le chapitre 10.4.7).

Comme variables temps réel de sortie, on peut choisir des variables qui,

- avec un effet correcteur additif,
- avec un effet correcteur multiplicatif,
- comme décalage de position,
- entrent directement dans l'opération d'usinage.



Application

On utilise la fonction d'évaluation

- pour la régulation AC (Adaptive Control)
- pour le positionnement par laser
- pour le forçage de position



840 D
NCU 572
NCU 573

10.4.10 Régulation AC (additive)

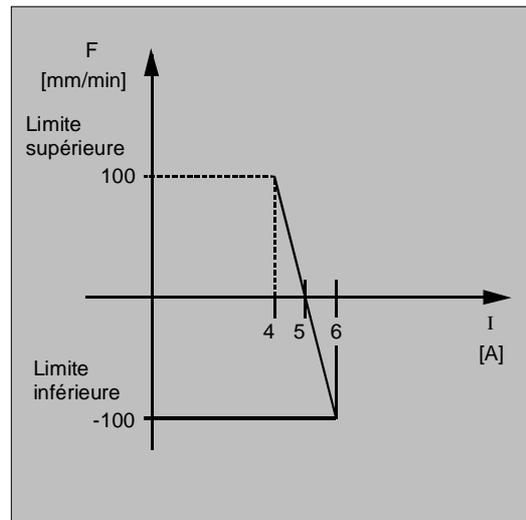


Exemple de programmation

Correction additive de l'avance programmée

Une avance programmée doit être corrigée de façon additive en fonction du courant de l'axe X (axe de pénétration) :

L'avance peut varier de +/- 100 mm/min, sachant que le courant fluctue sur +/-1A autour du point de travail à 5A.



1. Définition du polynôme

Détermination des coefficients

$$y = f(x) = a_0 + a_1x + a_2x^2 + a_3x^3$$

$$a_1 = -100\text{mm}/1 \text{ min A}$$

$$a_0 = -(-100) \cdot 5 = 500$$

$$a_2 = a_3 = 0 \text{ (pas d'opérateur carré, ni cubique)}$$

$$\text{Limite supérieure} = 100$$

$$\text{Limite inférieure} = -100$$

Il en résulte :

```
FCTDEF(1, -100, 100, 500, -100, 0, 0)
```

2. Activer la régulation AC

```
ID=1 DO SYNFACT(1, $AC_VC, $AA_LOAD[x])
```

;avec \$AA_LOAD[x] lire la charge courante de l'axe (en % du courant d'entraînement max.), puis calculer la correction de l'avance tangentielle avec le polynôme défini plus haut.



840 D
NCU 572
NCU 573

10.4.11 Régulation AC (multiplicative)

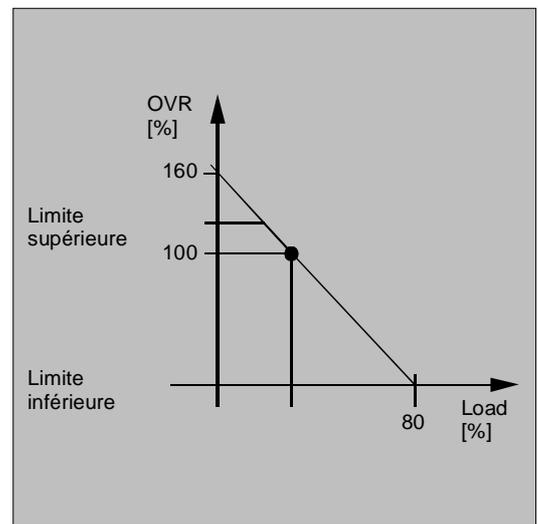


Exemple de programmation

Correction multiplicative de l'avance programmée

L'avance programmée doit être corrigée de façon multiplicative, sachant que l'avance – en fonction de la charge de l'entraînement – ne doit pas dépasser certaines limites :

- Quand la charge d'entraînement est de 80%, le prétraitement des blocs doit être arrêté : Correction = 0.
- Quand la charge d'entraînement est de 30%, le déplacement peut se faire avec avance programmée : Correction = 100 %.
- La vitesse d'avance ne doit pas être dépassée de plus de 20 % maximum : Correction max. = 120%.



1. Définition du polynôme

Détermination des coefficients

$$y = f(x) = a_0 + a_1x + a_2x^2 + a_3x^3$$

$$a_1 = -100\% / (80-30)\% = -2$$

$$a_0 = 100 + (2 \cdot 30) = 160$$

$$a_2 = a_3 = 0 \text{ (pas d'opérateur carré, ni cubique)}$$

$$\text{Limite supérieure} = 120$$

$$\text{Limite inférieure} = 0$$

Il en résulte :

```
FCTDEF ( 2 , 0 , 120 , 160 , -2 , 0 , 0 )
```

2. Activer la régulation AC

```
ID=1 DO SYNFACT ( 2 , $AC_OVR , $AA_LOAD[x] )
```

;avec \$AA_LOAD[x] lire la charge courante de l'axe (en % du courant d'entraînement max.), puis calculer la correction de l'avance par commutateur avec le polynôme défini plus haut.

10.4 Actions dans actions synchrones



840 D
NCU 572
NCU 573



810 D

10.4.12 Régulation d'écartement ("de distance de tir") avec correction limitée



Exemple de programmation

Calcul de la valeur des écartements avec vérification des zones limites

```
$AA_OFF_MODE = 1
```

Attention :

Le gain de la boucle de régulation dépend du réglage de la période d'appel de l'interpolateur.

Aide : Lire et calculer le PM pour la période d'appel de l'interpolateur.

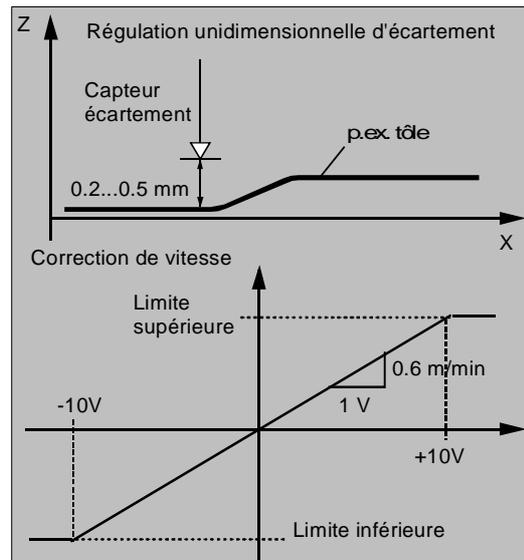
Remarque :

Limitation de la vitesse de l'interpolateur par le biais du PM 32020: JOG_VELO

pour une période d'appel de 12 ms :

Vitesse :

$$\frac{0.120\text{mm}}{0.6\text{ms}} / \text{mV} = 0.6 \frac{\text{m}}{\text{min}} / \text{V}$$



Sous-programme : Activer la régulation d'écartement

<code>%_N_AON_SPF</code>	Activer le sous-programme de régulation d'écartement
<code>PROC AON</code>	
<code>\$AA_OFF_LIMIT[Z]=1</code>	Définir la valeur limite
<code>FCTDEF(1, -10, +10, 0, 0.6, 0.12)</code>	Définition du polynôme
<code>ID=1 DO SYNFACT(1,\$AA_OFF[Z],\$A_INA[3])</code>	Régulation d'écartement active
<code>ID=2 WHENEVER \$AA_OFF_LIMIT[Z]<>0 DO \$AA_OVR[X] = 0</code>	Bloquer l'axe X en cas de dépassement de la zone limite
<code>RET</code>	
<code>ENDPROC</code>	

Sous-programme : Désactiver la régulation d'écartement

<code>%_N_AOFF_SPF</code>	
<code>PROC AOFF</code>	Sous-programme de régulation d'écartement, désactiver
<code>CANCEL(1)</code>	Action synchrone Régulation d'écartement, effacer
<code>CANCEL(2)</code>	Effacer la vérification des zones limites
<code>RET</code>	
<code>ENDPROC</code>	



840 D
NCU 572
NCU 573



810D

Programme principal :

```
%_N_MAIN_MPF
```

```
AON
```

```
Activer la régulation d'écartement
```

```
...
```

```
G1 X100 F1000
```

```
AOFF
```

```
Désactiver la régulation d'écartement
```

```
M30
```



Remarques

Décalage de position dans le système de coordonnées de base

Avec la variable système \$AA_OFF[axe], on peut programmer un déplacement forcé pour chaque axe dans le canal. Ce déplacement forcé agit comme un décalage de position dans le système de coordonnées de base.

Le décalage de position ainsi programmé est immédiatement appliqué sur l'axe en question, que cet axe soit programmé pour être déplacé ou non.

Depuis le logiciel SW 4, il est possible de limiter la valeur absolue à corriger (sortie variable temps réel) à la valeur rangée dans la donnée de réglage

SD 43350: AA_OFF_LIMIT.

Par le biais du PM 36750: AA_OFF_MODE, vous définissez le type de correction de l'écartement :

- 0 évaluation de type proportionnel
- 1 évaluation de type intégral

Avec la variable système \$AA_OFF_LIMIT[axe], on peut consulter, en fonction de la direction, si la valeur de correction se situe dans la zone limite. On peut interroger cette variable système dans des actions synchrones et, au moment où une valeur limite est atteinte, arrêter un axe ou mettre à 1 l'alarme par exemple.

- 0 la valeur de correction n'est pas dans la zone limite
- 1 la limite de la valeur de correction a été atteinte dans le sens positif
- 1 la limite de la valeur de correction a été atteinte dans le sens négatif

10.4 Actions dans actions synchrones



840 D
NCU 571



840 D
NCU 572



FM-NC



810 D
NCU 573

10.4.13 Correction d'outil en ligne FTOC



Programmation

FTOC(n° polynôme, EV, longueur1_2_3, canal, broche)



Signification

N° du polynôme	Polynôme défini avec FCTDEF, cf. "Définition des polynômes" dans ce chapitre
EV	Variable temps réel pour laquelle une valeur de la fonction doit être calculée par le biais du polynôme donné
Longueur1_2_3	Correction de longueur (\$TC_DP1 à 3) à laquelle sera additionnée la valeur de la fonction calculée.
Canal	Numéro du canal dans lequel la correction sera active. Aucune indication n'est à fournir pour une correction dans le canal propre. FTOCON doit être activé dans le canal de destination.
Broche	A indiquer uniquement si ce n'est pas la broche active qui doit être corrigée.



Fonction

FTOC permet un déplacement forcé pour un axe géométrique, suivant un polynôme programmé avec FCTDE et en fonction d'une valeur de référence qui peut être par exemple la valeur réelle d'un axe. Vous pouvez aussi programmer des corrections d'outil modales en ligne ou des régulations d'écartement comme actions synchrones.

Application

Usinage de la pièce et dressage de la meule dans le même canal ou dans des canaux différents (canal d'usinage et de dressage).

Les conditions marginales et les définitions pour le dressage des meules sont valables pour FTOC de manière analogue à la correction d'outil en ligne avec PUTFTOCF.

Pour plus d'informations, voyez le chapitre 5 "Correcteurs d'outil".



840 D
NCU 571



840 D
NCU 572
NCU 573



FM-NC

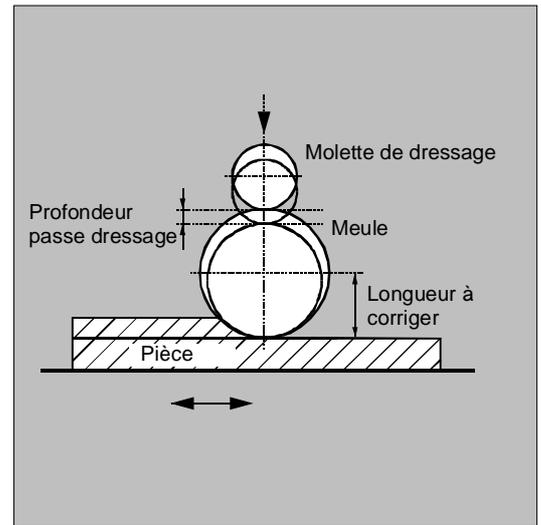


810D



Exemple de programmation

Dans l'exemple choisi, il convient de corriger la longueur de la meule active, en prise.



```
%_N_DRESSAG_MPF
```

```
FCTDEF(1, -1000, 1000, -$AA_IW[V], 1)
```

```
ID=1 DO FTOC(1, $AA_IW[V], 3, 1)
```

```
WAITM(1, 1, 2)
```

```
G1 V-0.05 F0.01 G91
```

```
G1 V-0.05 F0.02
```

```
...
```

```
CANCEL(1)
```

```
...
```

définir la fonction :

activer la correction d'outil en ligne :
la valeur réelle de l'axe V est valeur d'entrée pour le polynôme 1; le résultat est additionné dans le canal 1 comme valeur de correction de la longueur 3 de la meule active.

synchronisation avec canal d'usinage

mouvement de pénétration pour le dressage

désactivation de la correction en ligne

10.4 Actions dans actions synchrones



840 D
NCU 571



840 D
NCU 572



FM-NC



810 D
NCU 573

10.4.14 Déplacements de positionnement



Fonction

On peut positionner des axes de façon totalement asynchrones au programme pièce, à partir d'actions synchrones. La programmation des axes de positionnement à partir d'actions synchrones est recommandée pour des opérations cycliques, dans lesquelles les événements ont un rôle de pilotage important. Les axes programmés à partir d'actions synchrones sont appelés **axes de commande**.



Exemple de programmation

Bloquer un déplacement axial programmé

Si le déplacement de l'axe ne doit pas être lancé avec le début du bloc, on peut maintenir à 0 la correction par commutateur pour l'axe d'une action synchrone, jusqu'au moment souhaité pour le départ du déplacement.

```
WHENEVER $A_IN[1]==0 DO $AA_OVR[W]=0 G01 X10 Y25 F750 POS[W]=1500
AA=1000
```

;l'axe de positionnement reste immobilisé jusqu'à ce que l'entrée TOR 1 =0

10.4.15 Positionner l'axe : POS



Fonction

POS[axe]=valeur

Contrairement à la programmation à partir du programme pièce, le déplacement axial de positionnement n'a aucun impact sur l'exécution du programme pièce.



Signification

Axe : Nom de l'axe qui doit être déplacé

Valeur : Indication de la valeur du déplacement

840 D
NCU 571840 D
NCU 572

FM-NC

810D
NCU 573

Exemple de programmation

```
ID=1 EVERY $AA_IM[B]>75 DO POS[U]=100
```

;en fonction du mode de déplacement, déplacer l'axe U en relatif de 100 (inch/mm) ou en absolu sur la position 100 (inch/mm) à partir de l'origine commande.

```
ID=1 EVERY $AA_IM[B]>75 DO POS[U]=$AA_MW[V]-$AA_IM[W]+13.5
```

;déplacer l'axe U selon la distance calculée à partir des variables temps réel.

10.4.16 Lancer/Arrêter l'axe : MOV



Programmation

```
MOV [Axe]=valeur
```



Signification

Axe :	Nom de l'axe qui doit être lancé
Valeur :	Instruction de lancement du déplacement/de l'arrêt Le signe détermine le sens du déplacement Type de données de la valeur : INTEGER.
Valeur >0 (normalement +1) :	sens positif
Valeur <0 (normalement -1) :	sens négatif
Valeur ==0 :	Arrêter le déplacement axial



Fonction

Avec MOV[axe]=valeur, on peut lancer un axe de commande sans indication de la position de fin. L'axe en question est déplacé dans le sens programmé jusqu'à ce qu'une nouvelle commande de déplacement ou de positionnement indique un autre déplacement ou jusqu'à ce qu'il soit immobilisé par une instruction d'arrêt.



Exemple de programmation

```
... DO MOV[U]=0
```

L'axe U est arrêté

10.4 Actions dans actions synchrones



840 D
NCU 571



840 D
NCU 572



FM-NC



810 D
NCU 573



Remarque

Si on arrête un axe indexé avec MOV[axe]=0, l'arrêt se fera à la prochaine position indexée.

10.4.17 Avance axiale : AA



Exemple de programmation

AA[axe]=avance

```
ID=1 EVERY $AA_IM[B]>75 DO POS[U]=100 AA[U]=990
```

;imposer la valeur de l'avance

```
ID=1 EVERY $AA_IM[B]>75 DO POS[U]=100 AA[U]=$AA_VACTM[W]+100
```

;former la valeur de l'avance à partir des variables
temps réel

10.4.18 Fins de course logiciels



Fonction

La limitation de la zone de travail programmée avec G25/G26 est prise en compte pour les axes de commande en fonction de la donnée de réglage \$SA_WORKAREA_PLUS_ENABLE .

L'activation et la désactivation de la limitation de la zone de travail par le biais des fonctions G WALIMON/WALIMOF dans le programme pièce n'agissent pas sur les axes de commande.

10.4.19 Coordination des axes



Fonction

Par principe, un axe est déplacé soit à partir du programme pièce dans le bloc de déplacement, soit à partir d'une action synchrone comme axe de positionnement.

Mais si un même axe doit être déplacé de façon alternée à partir du programme pièce comme axe à interpolation ou axe de positionnement et à partir

840 D
NCU 571840 D
NCU 572

FM-NC



810D

d'une action synchrone, alors un transfert coordonné a lieu entre les deux déplacements de l'axe. Quand un axe de commande est ensuite déplacé à partir du programme pièce, ceci exige une réorganisation du prétraitement. Cette réorganisation entraîne de son côté une interruption de l'exécution du programme pièce, comparable à un arrêt du prétraitement des blocs.

**Exemple de programmation**

Déplacer l'axe X alternativement à partir du programme pièce et à partir d'actions synchrones :

N10 G01 X100 Y200 F1000	Axe X programmé dans le programme pièce
...	
N20 ID=1 WHEN \$A_IN[1]==1 DO POS[X]=150 AA[X]=200	Lancer le positionnement à partir de l'action synchrone, dès que l'entrée TOR est à 1.
...	
CANCEL(1)	Désactiver l'action synchrone
...	
N100 G01 X240 Y200 F1000	
;X devient axe à interpolation; un temps d'attente précède le déplacement à la suite à la transmission d'axe si l'entrée TOR était à 1 et X positionné à partir d'une action synchrone.	

**Exemple de programmation**

Modifier l'instruction de déplacement pour le même axe :

ID=1 EVERY \$A_IN[1]>=1 DO POS[V]=100 AA[V]=560	
;lancer le positionnement à partir de l'action synchrone, dès que l'entrée TOR est >= 1	
ID=2 EVERY \$A_IN[2]>=1 DO POS[V]=\$AA_IM[V] AA[V]=790	
l'axe suit; la 2ème entrée est mise à 1; autrement dit position finale et avance pour l'axe V, quand les deux actions synchrones sont activées en même temps, l'asservissement se fait en douceur pendant le déplacement.	

10.4 Actions dans actions synchrones



840 D
NCU 572
NCU 573



810 D
CCU2

10.4.20 Préréglage des mémoires de valeurs réelles



Fonction

La position courante de l'axe n'est pas modifiée lors de l'exécution de PRESETON (axe,valeur); une nouvelle valeur lui est affectée.



Remarques

On peut faire PRESETON à partir d'actions synchrones pour :

- des axes rotatifs modulus qui ont été lancés depuis le programme pièce
- tous les axes de commande qui ont été lancés depuis l'action synchrone

Restriction

On ne peut pas faire PRESETON pour les axes qui participent à la transformation.



Exemple de programmation

```
WHEN $AA_IM[a] >= 89.5 DO PRESETON(a4,10.5)
```

;décaler l'origine commande de l'axe a de 10.5 unités de longueur (inch ou mm)
dans le sens positif de l'axe.



Restriction

Le déplacement d'un même axe à partir du programme pièce ou d'une action synchrone ne peut se faire qu'avec un décalage de temps, par conséquent lors de la programmation d'un axe à partir du programme pièce, des temps d'attente peuvent apparaître si cet axe était programmé précédemment dans une action synchrone.

Si le même axe est utilisé alternativement, un transfert coordonné a lieu entre les deux mouvements axiaux. L'exécution du programme pièce doit être interrompue dans ce but.



840 D
NCU 572
NCU 573



810D
CCU2

10.4.21 Déplacement des broches



Fonction

A partir d'actions synchrones, on peut positionner des broches en asynchronisme total avec le programme pièce. Ce type de programmation est recommandé pour des opérations cycliques, dans lesquelles les événements ont un rôle de pilotage important.



Exemple de programmation

Lancer/Positionner/Arrêter une broche

ID=1 EVERY \$A_IN[1]==1 DO M3 S1000	régler le sens et la vitesse de rotation
ID=2 EVERY \$A_IN[2]==1 DO SPOS=270	positionner la broche



Ordre de traitement

Si des instructions concurrentes sont données pour une broche par des différentes actions synchrones activées simultanément, ce sera la dernière instruction dans l'ordre chronologique qui sera prise en compte.



Exemple de programmation

Régler le sens et la vitesse de rotation/
positionner la broche

ID=1 EVERY \$A_IN[1]==1 DO M3 S300	régler le sens et la vitesse de rotation
ID=2 EVERY \$A_IN[2]==1 DO M4 S500	imposer le nouveau sens de rotation et la nouvelle vitesse de rotation
ID=3 EVERY \$A_IN[3]==1 DO S1000	imposer la nouvelle vitesse de rotation
ID=4 EVERY (\$A_IN[4]==1) AND (\$A_IN[1]==0) DO SPOS=0	positionner la broche

10.4 Actions dans actions synchrones



840 D
NCU 572
NCU 573



810 D
CCU2

10.4.22 Déplacements conjugués : TRAILON, TRAILOF



Fonction

DO TRAILON (axe asservi, axe pilote, facteur de couplage)	Activer les déplacements conjugués
DO TRAILOF (axe asservi, axe pilote, axe pilote 2)	Désactiver les déplacements conjugués

Quand on active le couplage à partir de l'action synchrone, l'axe pilote peut être en cours de déplacement. Dans ce cas, l'axe asservi subit une accélération pour atteindre la vitesse de consigne. La position de l'axe pilote au moment de la synchronisation des vitesses constitue la position de départ des déplacements conjugués. Les déplacements conjugués sont décrits dans le chapitre "Mode de déplacement".

Activer les déplacements conjugués synchrones :

... DO **TRAILON**(AA, AP, Kf)

avec : AA: axe asservi
AP: axe pilote
Kf: facteur de couplage

Désactiver les déplacements conjugués synchrones :

... DO **TRAILOF**(AA, AP, LA2)

avec : AA: axe asservi
AP: axe pilote
AP2: axe pilote 2, optionnel



Exemple de programmation

\$A_IN[1]==0 DO TRAILON(Y,V,1)	Activation du 1er groupe d'axes à déplacements conjugués, dès que l'entrée est 1.
\$A_IN[2]==0 DO TRAILON(Z,W,-1)	Activation du 2ème groupe d'axes à déplacements conjugués
G0 Z10	Déplacement des axes Z et W dans le sens opposé
G0 Y20	Déplacement des axes Y et V dans le même sens
...	
G1 Y22 V25	Superposition d'un déplacement dépendant et d'un déplacement indépendant de l'axe conjugué „V“
...	
TRAILOF(Y,V)	Désactivation du 1er groupe d'axes à déplacements conjugués
TRAILOF(Z,W)	Désactivation du 2ème groupe d'axes à déplacements conjugués

840 D
NCU 571840 D
NCU 572

FM-NC

810D
NCU 573

10.4.23 Couplage par valeur pilote : LEADON, LEADOF



Fonction

Le couplage axial par valeur pilote est programmable sans restriction dans des actions synchrones.

Activer le couplage par valeur pilote :

```
...DO LEADON(AA, AP, NR)
```

avec : AA: axe asservi
AP: axe pilote
NR: numéro de la table de courbe en mémoire

Désactiver le couplage par valeur pilote :

```
...DO LEADOF(AA, AP)
```

avec : AA: axe asservi
AP: axe pilote



Exemple de programmation

Tronçonnage au vol

Un boyau, qui traverse à vitesse continue la zone de travail, doit être découpé en tronçons de même longueur.

Axe X : Axe dans lequel se déplace le boyau. SCP

Axe X1 : Axe machine du boyau, SCM

Axe Y : Axe dans lequel la scie "se déplace avec" le boyau

On suppose que le déplacement de la scie et sa commande sont pilotés par l'AP. Pour vérifier le synchronisme entre le boyau et la scie, on peut exploiter les signaux de l'interface AP.

Actions Activer le couplage, LEADON

Désactiver le couplage, LEADOF

Prérégler les mémoires des valeurs réelles, PRESETON

10.4 Actions dans actions synchrones

840 D
NCU 571840 D
NCU 572

FM-NC

810 D
NCU 573

```

%_N_SCHERE1_MPF
; $PATH=/_N_SCP_DIR/_N_DEMOFBE_WPD
N100 R3=1500 ;longueur du tronçon à découper
N200 R2=100000 R13=R2/300
N300 R4=100000
N400 R6=30 ;position de départ axe Y
N500 R1=1 ;condition de départ pour l'axe de la scie
N600 LEADOF(Y,X) ;effacement d'un couplage éventuel
N700 CTABDEF(Y,X,1,0) ;définition de la table
N800 X=30 Y=30 ;couple de valeurs
N900 X=R13 Y=R13
N1000 X=2*R13 Y=30
N1100 CTABEND ;fin de la définition de la table
N1200 PRESETON(X1,0) ;PRESET de début
N1300 Y=R6 G0 ;pos.départ axe Y, l'axe est linéaire
N1400 ID=1 WHENEVER $AA_IW[X]>$R3 DO PESETON(X1,0)
;PRESET après longueur R3, nouveau départ après
sectionnement
N1500 WAITP(Y)
N1800 ID=6 EVERY $AA_IM[X]<10 DO LEADON(Y,X,1)
; quand X < 10 coupler Y à X par le biais de
la table 1
N1900 ID=10 EVERY $AA_IM[X]>$R3-30 DO LEADOF(Y,X)
; > 30 désactiver le couplage avant le
déplacement sur la longueur à tronçonner
N2000 WAITP(X)
N2100 ID=7 WHEN $R1==1 DO MOV[X]=1 ;activer l'axe du boyau en cheminement
AA[X]=$R4 continu
N2200 M30

```

840 D
NCU 571840 D
NCU 572

FM-NC



810D

NCU 573

10.4.24 Mesure



En comparaison avec l'utilisation des blocs de déplacement du programme pièce, la fonction de mesure peut être activée et désactivée à volonté à partir d'actions synchrones.

- Mesure axiale sans effacement de la distance restant à parcourir :

MEAWA[axe]=(mode, événement déclencheur_1, ..._4)

- Mesure continue sans effacement de la distance restant à parcourir :

MEAC[axe]=(mode, mémoire de mesure, événement déclencheur_1, ..._4)

Autres informations sur la mesure : voir chapitre 5, „Mesure étendue“

10.4.25 Activer/Effacer les marques d'attente : SETM, CLEARM



Fonction

SETM (NuméroMarque)

Activer la marque d'attente pour le canal

CLEARM (NuméroMarque)

Effacer la marque d'attente pour le canal

Dans des actions synchrones, vous pouvez activer ou effacer des marques d'attente, par exemple pour coordonner des canaux entre eux.

SETM

L'instruction SETM peut être écrite dans le programme pièce et dans la partie action d'une action synchrone. Elle active la marque NuméroMarque pour le canal auquel elle est affectée.

CLEARM

L'instruction CLEARM peut être écrite dans le programme pièce et dans la partie action d'une action synchrone. Elle efface la marque NuméroMarque pour le canal auquel elle est affectée.

10.4 Actions dans actions synchrones



840 D
NCU 571



840 D
NCU 572



FM-NC



810 D
NCU 573

10.4.26 Réactions en cas de défaut



Fonction

Avec les actions synchrones, on peut programmer des réactions en cas de défaut, en interrogeant des variables d'état et en déclenchant respectivement des actions.

En réponse à des états de défaut, on peut avoir les réactions possibles suivantes :

- arrêter l'axe : correction=0
- activer une alarme avec SETAL on peut activer des alarmes cycliques à partir d'actions synchrones
- mettre sortie à 1
- toutes les actions possible dans des actions synchrones



Exemple de programmation

```
ID=67 WHENEVER ($AA_IM[X1]-$AA_IM[X2])<4.567 DO $AA_OVR[X2]=0
```

;si la distance de sécurité entre les axes X1 et X2 est trop faible, arrêter axe X2.

```
ID=67 WHENEVER ($AA_IM[X1]-$AA_IM[X2])<4.567 DO SETAL(61000)
```

;si la distance de sécurité entre les axes X1 et X2 est trop faible, actionner alarme.

840 D
NCU 571840 D
NCU 572

FM-NC

810D
NCU 573

10.5 Cycles technologiques



Fonction

Dans des actions synchrones, on peut aussi appeler comme actions des programmes, qui toutefois doivent contenir uniquement des fonctions admises elles aussi comme actions dans des actions synchrones. Ces programmes sont appelés des cycles technologiques.

Les cycles technologiques sont rangés comme sous-programmes dans la commande. Pour l'utilisateur, l'appel de ces cycles se fait comme pour des sous-programmes. Le transfert de paramètres n'est pas possible.

Plusieurs cycles technologiques ou actions peuvent être exécutées en parallèle dans un canal.

La fin du programme est à programmer avec M02/M17/M30/RET. Par bloc, on ne peut pas programmer plus d'un déplacement axial



Application

Cycles technologiques comme programmes d'axe : Chaque cycle technologique commande un seul axe. Ainsi, différents déplacements d'axes peuvent être lancés, en réaction à des événements, dans une même période d'appel de l'interpolateur. Dans le cas extrême, le programme pièce ne sert plus qu'à la gestion des actions synchrones.

10.5 Cycles technologiques

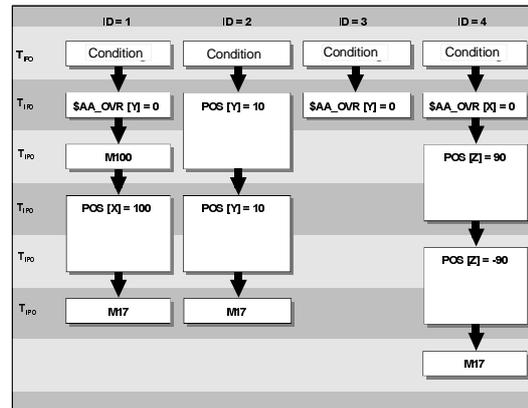
840 D
NCU 571840 D
NCU 572

FM-NC

810 D
NCU 573

Exemple de programmation

Les programmes d'axe sont lancés par la mise à 1 d'entrées TOR.



Programme principal

ID=1 EVERY \$A_IN[1]==1 DO AXE_X	Quand l'entrée 1 est à 1, lancer programme d'axe X
ID=2 EVERY \$A_IN[2]==1 DO AXE_Y	Quand l'entrée 2 est à 1, lancer programme d'axe Y
ID=3 EVERY \$A_IN[3]==1 DO \$AA_OVR[Y]=0	Quand l'entrée 3 est à 1, mettre la correction par commutateur de l'axe Y à 0
ID=4 EVERY \$A_IN[4]==1 DO AXE_Z	Quand l'entrée 4 est à 1, lancer programme d'axe Z

M30

Cycle technologique comme programme d'axe AXE_X :

\$AA_OVR[Y]=0

M100

POS[X]=100 AA[X]=300

M17

Cycle technologique comme programme d'axe AXE_Y :

POS[Y]=10 AA[Y]=200

POS[Y]=-10

M17

Cycle technologique comme programme d'axe AXE_Z :

\$AA_OVR[X]=0

POS[Z]=90 AA[Z]=250

POS[Z]=-90

M17

840 D
NCU 571840 D
NCU 572
NCU 573

FM-NC



810D

Les cycles technologiques sont lancés dès que leurs conditions sont remplies. Dans le cas des axes de positionnement, plusieurs périodes d'appel de l'interpolateur sont nécessaire pour l'exécution. D'autres fonctions (OVR) sont exécutées en une seule période.

Dans un cycle technologique, l'exécution des blocs se fait de façon séquentielle.



Remarques

S'il arrive dans une même période d'appel de l'interpolateur que des actions soient appelées qui s'excluent réciproquement, l'action lancée sera celle qui est appelée par l'action synchrone avec le numéro ID le plus élevé.

10.5.1 Bloquer, libérer, interrompre : LOCK, UNLOCK, RESET



Programmation

LOCK (n, n, ...)	Bloquer le cycle technologique, l'action activée est interrompue
UNLOCK (n, n, ...)	Libérer le cycle technologique
RESET (n, n, ...)	Interrompre le cycle technologique, l'action activée est interrompue
n	Numéro d'identification de l'action synchrone



Fonction

Le déroulement d'un cycle technologique peut être bloqué, libéré et interrompu à partir d'actions synchrones ou à partir d'un autre cycle technologique.

Bloquer un cycle technologique, LOCK

Un cycle technologique peut être bloqué avec LOCK à partir d'une autre action synchrone ou d'un autre cycle technologique .

Exemple :

```
N100 ID=1 WHENEVER $A_IN[1]==1 DO M130
```

```
...
```

```
N200 ID=2 WHENEVER $A_IN[2]==1 DO LOCK(1)
```

10.5 Cycles technologiques



840 D
NCU 571



840 D
NCU 572



FM-NC



810 D
NCU 573

Libérer un cycle technologique, UNLOCK

Un cycle technologique qui a été bloqué peut être libéré avec UNLOCK à partir d'une autre action synchrone ou d'un autre cycle technologique. Avec UNLOCK, le cycle technologique reprend à la position courante, de la même façon qu'un processus de positionnement quand il a été interrompu.

Exemple :

```
N100 ID=1 WHENEVER $A_IN[1]==1 DO M130
...
N200 ID=2 WHENEVER $A_IN[2]==1 DO LOCK(1)
...
N250 ID=3 WHENEVER $A_IN[3]==1 DO UNLOCK(1)
```

Interrompre un cycle technologique, RESET

Un cycle technologique peut être interrompu avec RESET à partir d'une autre action synchrone ou d'un autre cycle technologique.

Exemple :

```
N100 ID=1 WHENEVER $A_IN[1]==1 DO M130
...
N200 ID=2 WHENEVER $A_IN[2]==1 DO RESET(1)
```



Verrouillage depuis l'AP

Les actions synchrones modales avec les numéros ID **n=1 ... 64** peuvent être verrouillées par l'AP. Dans ce cas, la condition correspondante n'est plus traitée et l'exécution de la fonction dans NCK est bloquée. Sur un signal de l'interface AP, toutes les actions synchrones peuvent être bloquées en même temps.



Remarques

Une action synchrone programmée est active par défaut et peut être protégée contre un écrasement des données/un blocage par le biais d'un paramètre machine.

Application :

Le client final n'a pas à intervenir dans les actions synchrones qui ont été définies par le fabricant de la machine.

10.6 Effacer une action synchrone : CANCEL

840 D
NCU 571840 D
NCU 572

FM-NC



810D

NCU 573

10.6 Effacer une action synchrone : CANCEL



Programmation

CANCEL(n, n, . . .)

Effacer une action synchrone

n

Numéro d'identification de l'action synchrone



Signification

Avec CANCEL, vous pouvez effacer directement, depuis un programme pièce, une action synchrone modale désigné par ID(S)=n.

Exemple :

N100 ID=2 WHENEVER \$A_IN[1]==1 DO M130

. . .

N200 CANCEL (2)

effacer action synchrone N° 2



Remarques

Les déplacements encore en cours, qui ont été lancés par l'action synchrone que l'on souhaite effacer, sont menés à terme conformément à la programmation.

10.7 Conditions marginales



840 D
NCU 571



840 D
NCU 572



FM-NC



810 D
NCU 573

10.7 Conditions marginales

- **Power On**

Après un Power On, aucune action synchrone n'est activée

Toutefois, les actions synchrones statiques peuvent être activées immédiatement après un Power On, par le biais d'un sous-programme asynchrone (ASUP) lancé par l'AP.

- **Changement de mode de fonctionnement**

Avec le mot-clé IDS, les actions synchrones activées le restent au-delà du changement du mode de fonctionnement.

Toutes les autres actions synchrones sont désactivées par le changement de mode de fonctionnement (par ex. positionnement d'un axe) et sont réactivées en mode automatique avec le Repositionnement et le Retour.

- **Reset**

Avec un reset CN, on interrompt toutes les actions lancées par le biais d'actions synchrones. Les actions synchrones statiques restent activées. De nouvelles actions peuvent être lancées à partir de celles-ci.

Avec l'instruction **RESET** depuis une action synchrone ou un cycle technologique, on peut annuler une action synchrone à effet modal.

Quand une action synchrone est annulée, alors que le déplacement axial de positionnement qu'elle avait enclenché est encore actif, ce déplacement sera interrompu.

Les actions synchrones déjà exécutées de type WHEN ne seront plus traitées après un reset.

840 D
NCU 571840 D
NCU 572

FM-NC



810D

NCU 573

Comportement après un reset		
Action synchrone / Cycle technologique	à effet modal/non modal	statique (IDS)
	l'action activée est interrompue, les actions synchrones sont effacées	l'action activée est interrompue, le cycle technologique est annulé
Axe / broche de positionnement	le déplacement est interrompu	le déplacement est interrompu
Broche à régulation de la vitesse de rotation	\$MA_SPIND_ACTIVE_AFTER_RESET==1: la broche reste activée \$MA_SPIND_ACTIVE_AFTER_RESET==0: la broche s'immobilise.	\$MA_SPIND_ACTIVE_AFTER_RESET==1: la broche reste activée \$MA_SPIND_ACTIVE_AFTER_RESET==0: la broche s'immobilise.
Couplage par valeur pilote	\$MC_RESET_MODE_MASK, Bit13 == 1: le couplage par valeur pilote reste activé \$MC_RESET_MODE_MASK, Bit13 == 0: le couplage par valeur pilote est annulé	\$MC_RESET_MODE_MASK, Bit13 == 1: le couplage par valeur pilote reste activé \$MC_RESET_MODE_MASK, Bit13 == 0: le couplage par valeur pilote est annulé
Opérations de mesure	les opérations de mesure lancées à partir d'actions synchrones sont interrompues	les opérations de mesure lancées à partir d'actions synchrones statiques sont interrompues

- **Suspension du programme**

Les actions synchrones statiques restent activées après une suspension du programme. Les déplacements amorcés par les actions synchrones statiques ne sont pas interrompus. Les actions synchrones **spécifiques à un programme** et appartenant au bloc activé, restent activées, mais les déplacements qui avaient été amorcés sont interrompus.

- **Fin de programme**

Fin de programme et action synchrone n'ont aucune influence réciproque. Les actions synchrones en cours sont menées à terme, même après une fin de programme. Les actions synchrones activées dans le bloc M30 le restent au-delà de la fin du programme. Si vous ne le souhaitez pas, interrompez l'action synchrone avant la fin du programme avec CANCEL (cf. sous-chapitre suivant).

10.7 Conditions marginales

840 D
NCU 571840 D
NCU 572
NCU 573

FM-NC



810 D

Comportement après la fin du programme		
Action synchrone / Cycle technologique	à effet modal et non modal ils sont interrompus	statique (IDS) ils sont conservés
Axe / broche de positionnement	M30 est temporisé jusqu'à ce que l'axe / la broche soit immobilisé(e).	Le déplacement se poursuit
Broche à régulation de la vitesse de rotation	Fin de programme : \$MA_SPIND_ACTIVE_AFTER_RESET==1: la broche reste activée \$MA_SPIND_ACTIVE_AFTER_RESET==0: la broche s'immobilise la broche reste activée après un changement de mode de fonctionnement	la broche reste activée
Couplage par valeur pilote	\$MC_RESET_MODE_MASK, Bit13 == 1: le couplage par valeur pilote reste activé \$MC_RESET_MODE_MASK, Bit13 == 0: le couplage par valeur pilote est annulé	le couplage lancé à partir d'une action synchrone statique est conservé
Opérations de mesure	les opérations de mesure lancées à partir d'actions synchrones sont interrompues	les opérations de mesure lancées à partir d'actions synchrones statiques restent activées

- **Recherche de bloc**

Les actions synchrones identifiées lors de la recherche de blocs sont regroupées et traitées à la reprise du programme; les actions correspondantes sont relancées le cas échéant.

Les actions synchrones statiques restent actives pendant la recherche de bloc.

Si la recherche de blocs fait apparaître des coefficients polynomiaux programmés avec FCTDEF, ces derniers seront appliqués avec effet immédiat.

- **Interruption du programme par un sous-programme asynchrone**

Début ASUP :

A effet modal et statique Les actions synchrones de déplacement sont conservées et restent actives dans le sous-programme asynchrone.

Fin ASUP :

Si le sous-programme asynchrone n'est pas poursuivi avec Repos, les actions synchrones de déplacement à effet modal et statique modifiées dans le sous-programme asynchrone, restent actives dans le programme principal.

840 D
NCU 571840 D
NCU 572

FM-NC



810D

NCU 573

- **Repositionnement**

Les actions synchrones actives dans le bloc suspendu sont réactivées après repositionnement avec REPOS.

Après REPOS, les actions synchrones à effet modal qui ont été modifiées dans le sous-programme asynchrone ne sont plus actives dans l'exécution du reste du bloc.

Les coefficients polynomiaux programmés avec FCTDEF ne sont pas influencés par les sous-programmes asynchrones et par REPOS.

Indépendamment de l'endroit où ils ont été programmés, on peut les utiliser à tout moment dans le sous-programme asynchrone et dans le programme principal, même après un REPOS.

- **Désactivation avec CANCEL**

Quand on désactive avec **CANCEL** une action synchrone active, l'action activée n'est pas touchée. Les déplacements de positionnement sont menés à terme tels qu'ils ont été programmés.

Avec l'instruction CANCEL, on peut interrompre une action synchrone à effet modal ou statique. Quand une action synchrone est interrompue, alors que le déplacement axial de positionnement qu'elle avait enclenché est encore actif, ce déplacement sera mené à terme. Si vous ne le souhaitez pas, vous pouvez freiner le déplacement axial en faisant un effacement de la distance restant à parcourir avant l'instruction CANCEL :

Exemple :

ID=17 EVERY \$A_IN[3]=1 DO POS[X]=15 AA[X]=1500	; Lancer le déplacement axial de positionnement
...	
WHEN ... DO DELDTG(X)	;mettre fin au déplacement axial de positionnement
CANCEL(1)	

Oscillation

11.1 Oscillation asynchrone.....	11-366
11.2 Oscillation commandée par des actions synchrones.....	11-372

11.1 Oscillation asynchrone



840 D
NCU 571



840 D
NCU 572



FM-NC



810D
NCU 573

11.1 Oscillation asynchrone



Signification des instructions

OSP1[axe]=	Position du point d'inversion de sens 1
OSP2[axe]=	Position du point d'inversion de sens 2
OST1[axe]=	Temps d'arrêt aux points d'inversion de sens en secondes
OST2[axe]=	
FA[axe]=	Avance de l'axe d'oscillation
OSCTRL[axe]=	(déroulement du déplacement)
OSNSC[axe]=	Nombre de passes à lécher
OSE[axe]=	Position de fin
OS[axe]=	1 = activation de l'axe d'oscillation ; 0 = désactivation de l'axe d'oscillation

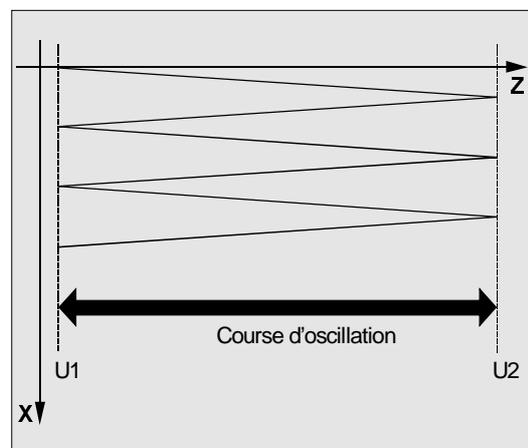
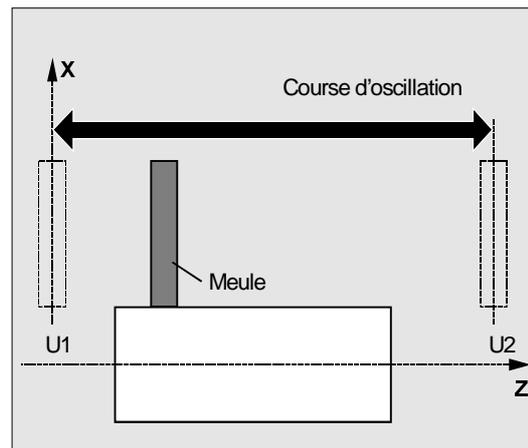


Fonction

Un axe d'oscillation va et vient, avec une avance donnée, entre deux points d'inversion de sens 1 et 2 jusqu'à désactivation de l'oscillation.

Pendant l'oscillation, d'autres axes peuvent être interpolés librement.

Une pénétration continue peut être obtenue par un déplacement de contournage ou avec un axe de positionnement. Il n'existe toutefois **aucun lien** entre l'oscillation et le mouvement de pénétration.



840 D
NCU 571840 D
NCU 572
NCU 573

FM-NC



810D



L'axe d'oscillation

Pour les axes d'oscillation il est convenu que :

- chaque axe peut être utilisé comme axe d'oscillation
- plusieurs axes d'oscillation peuvent être actifs simultanément (au maximum : nombre d'axes de positionnement).
- Pour l'axe d'oscillation – quelle que soit la fonction G momentanément active dans le programme – l'interpolation linéaire G1 est toujours activée.

L'axe d'oscillation peut être :

- un axe d'entrée pour la transformation dynamique,
- l'axe directeur dans le cas d'axes Gantry et d'axes conjugués,
- déplacé
 - sans limitation des à-coups (BRISK) ou
 - avec limitation des à-coups (SOFT) ou
 - avec ligne caractéristique d'accélération coudée (comme les axes de positionnement).

Points d'inversion du sens d'oscillation

Pour déterminer les points d'inversion de l'oscillation, il convient de tenir compte des décalages actuels ci-après :

- Déclaration en absolu

OSP1[Z]=valeur

Position du point d'inversion = Somme des décalages + valeur programmée

- Déclaration en relatif

OSP1[Z]= IC(valeur)

Position du point d'inversion = point d'inversion 1 + valeur programmée

Exemple :

N10 OSP1[Z]=100 OSP2[Z]=110

.

.

N40 OSP1[Z]= IC(3)

11.1 Oscillation asynchrone

840 D
NCU 571840 D
NCU 572

FM-NC

810D
NCU 573

Caractéristiques de l'oscillation asynchrone

- L'oscillation asynchrone agit au-delà des limites du bloc et est spécifique à un axe.
- Une activation de l'oscillation, synchronisée avec le bloc, est assurée par le programme pièce.
- Une interpolation commune de plusieurs axes et une correction de la course d'oscillation ne sont pas possibles.

Données de réglage

Les données de réglage nécessaires pour l'oscillation asynchrone peuvent être réglées dans le programme pièce.

Si les données de réglage sont écrites directement dans le programme pièce, la modification prend effet dès le prétraitement. Le comportement synchrone peut être obtenu par le biais d'un arrêt du prétraitement STOPRE.

Exemple :

Oscillation avec modification en ligne de la position d'inversion

```
$SA_OSCILL_REVERSE_POS1[Z]=-10
```

```
$SA_OSCILL_REVERSE_POS2[Z]=10
```

```
G0 X0 Z0
```

```
WAITP(Z)
```

```
ID=1 WHENEVER $AA_IW[Z] < $$AA_OSCILL_REVERSE_POS1[Z] DO $AA_OVR[X]=0
```

```
ID=2 WHENEVER $AA_IW[Z] < $$AA_OSCILL_REVERSE_POS2[Z] DO $AA_OVR[X]=0
```

; Lorsque la position réelle de l'axe

; d'oscillation se trouve au-delà du point

; d'inversion de sens, l'axe de pénétration est arrêté.

```
OS[Z]=1 FA[X]=1000 POS[X]=40
```

; Activation de l'oscillation

```
OS[Z]=0
```

; Désactivation de l'oscillation

```
M30
```

840 D
NCU 571840 D
NCU 572
NCU 573

FM-NC



810D



Remarques sur les différentes fonctions

Les adresses suivantes permettent d'activer et d'agir sur l'oscillation asynchrone à partir du programme pièce, en fonction de l'exécution du programme CN. Les valeurs programmées sont inscrites dans les données de réglage correspondantes pendant l'exécution du programme et restent actives jusqu'à la prochaine modification.

Activer, désactiver l'oscillation: OS

OS[axe] = 1 : activer

OS[axe] = 0 : désactiver



WAITP (axe):

- Pour faire osciller un axe géométrique, il faut débloquer ce dernier pour l'oscillation avec WAITP.
- Lorsque l'oscillation est terminée, l'axe d'oscillation est à nouveau défini comme axe de positionnement avec cette instruction et peut à nouveau être utilisé normalement.

Temps d'arrêt aux points d'inversion de sens : OST1, OST2

Temps d'arrêt	Comportement dans la zone d'arrêt précis, au point d'inversion de sens
-2	l'interpolation est poursuivie sans attente de l'arrêt précis
-1	attente de l'arrêt précis grossier
0	attente de l'arrêt précis fin
>0	attente de l'arrêt précis fin puis écoulement du temps d'arrêt

L'unité du temps d'arrêt est identique à celle de l'arrêt temporisé programmé avec G4.

11.1 Oscillation asynchrone



840 D
NCU 571



840 D
NCU 572



FM-NC



810D
NCU 573

Réglage de l'avance FA

La vitesse d'avance appliquée ici est celle définie pour l'axe de positionnement.

Si aucune vitesse d'avance n'a été définie, c'est la valeur qui figure dans le paramètre machine qui s'applique.

Définition de la séquence des déplacements : OSCTRL

La configuration de la commande numérique pour la séquence des déplacements est réglée avec des options d'activation et de désactivation.

Options de désactivation

Ces options sont mises hors fonction (uniquement si elles avaient été mises en fonction auparavant avec les options d'activation).

Options d'activation

Ces options sont basculées. Quand on programme OSE (position finale), c'est l'option 4 qui est active de façon implicite.

Valeur de l'option	Signification
0	A la désactivation de l'oscillation, arrêt au prochain point d'inversion de sens (préréglage) ; possible uniquement par remise à 0 des valeurs 1 et 2
1	A la désactivation de l'oscillation, arrêt au point d'inversion de sens 1
2	A la désactivation de l'oscillation, arrêt au point d'inversion de sens 2
3	A la désactivation de l'oscillation, ne pas accoster de point d'inversion de sens, si aucune passe à lécher n'est programmée
4	Après les passes à lécher, accoster une position de fin
8	Si l'oscillation est abandonnée par effacement de la distance restant à parcourir : exécuter ensuite les passes à lécher et accoster éventuellement la position de fin
16	Si l'oscillation est abandonnée par effacement de la distance restant à parcourir : accoster la position d'inversion de sens réglée pour la désactivation
32	L'avance modifiée ne prend effet qu'à partir du prochain point d'inversion de sens
64	FA = 0 : la correction de déplacement est active FA <> 0 : la correction de vitesse est active
128	Dans le cas d'un axe rotatif DC (course minimale)

Plusieurs options sont combinées en insérant le signe plus.

Exemple :

OSCTRL[Z] = (1+4,16+32+64)

840 D
NCU 571840 D
NCU 572
NCU 573

FM-NC

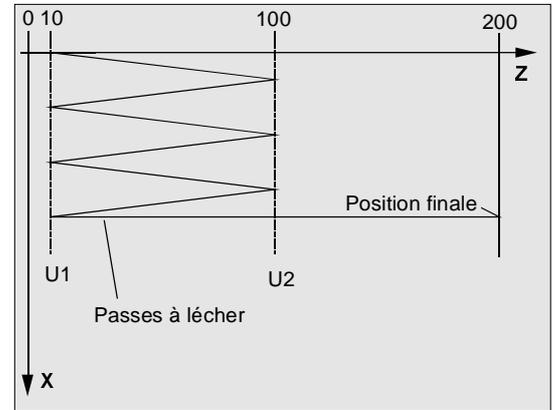


810D



Exemple de programmation

L'axe d'oscillation Z doit osciller entre 10 et 100. Accoster le point d'inversion de sens 1 avec arrêt précis fin, le point d'inversion de sens 2 avec arrêt précis grossier. Une avance de 250 doit être utilisée pour l'axe d'oscillation. A la fin de l'usinage, trois passes à lécher doivent être exécutées et la position finale 200 doit être rejointe par l'axe d'oscillation. L'avance pour l'axe de pénétration est 1, la fin de la pénétration en direction X est à 15.



WAITP(X, Y, Z)	Position de départ
G0 X100 Y100 Z100	Passage au mode axe de positionnement
N40 WAITP(X, Z)	
N50 OSP1[Z]=10 OSP2[Z]=100 -> -> OSE[Z]=200 -> -> OST1[Z]=0 OST2[Z]=-1 ->	Point d'inversion de sens 1, point d'inversion de sens 2 Position finale Temps d'arrêt en U1 : arrêt précis fin ; Temps d'arrêt en U2 : arrêt précis grossier Avance axe d'oscillation, axe de pénétration Options de réglage Trois passes à lécher Démarrage de l'oscillation
-> FA[Z]=250 FA[X]=1 -> -> OSCTRL[Z]=(4, 0) -> -> OSNSC[Z]=3 ->	
N60 OS[Z]=1	
N70 WHEN \$A_IN[3]==TRUE -> -> DO DELDTG(X)	Effacement de la distance restant à parcourir
N80 POS[X]=15	Positions de départ axe X
N90 POS[X]=50	
N100 OS[Z]=0	Arrêter l'oscillation
M30	

-> programmable dans un bloc.

11.2 Oscillation commandée par des actions synchrones



840 D
NCU 571



840 D
NCU 572



FM-NC



810D
NCU 573

11.2 Oscillation commandée par des actions synchrones



Programmation

1. Fixer les paramètres pour l'oscillation
2. Définir les actions synchrones aux déplacements
3. Affecter les axes, fixer la profondeur de pénétration

Paramètres pour l'oscillation

OSP1[axe d'oscillation]=	Position du point d'inversion de sens 1
OSP2[axe d'oscillation]=	Position du point d'inversion de sens 2
OST1[axe d'oscillation]=	Temps d'arrêt au point d'inversion de sens 1 en secondes
OST2[axe d'oscillation]=	Temps d'arrêt au point d'inversion de sens 2 en secondes
FA[axe d'oscillation]=	Avance de l'axe d'oscillation
OSCTRL	Options de réglage
[axe d'oscillation]=	
OSNSC	Nombre de passes à lécher
[axe d'oscillation]=	
OSE[axe d'oscillation]=	Position de fin
WAITP(axe d'oscillation)	Libération de l'axe pour l'oscillation

Affectation des axes, profondeur de pénétration

OSCILL[axe d'oscillation] = (axe de pénétration 1, axe de pénétration 2, axe de pénétration 3)

POSP[axe de pénétration] = (pos. de fin, pénétr. partielle, mode)

OSCILL	Affecter l'axe ou les axes de pénétration à l'axe d'oscillation
POSP	Fixer les pénétrations totale et partielle (cf. chapitre 3)
Endpos	Position de fin pour l'axe de pénétration, quand toutes les pénétrations partielles ont été effectuées.
Pénétration partielle	Valeur de la pénétration partielle au point d'inversion de sens/dans la zone d'inversion de sens
mode	Division de la pénétration totale en pénétrations partielles 0 = deux dernières pénétrations partielles identiques préréglage); 1 = toutes les pénétrations partielles identiques

Actions synchrones au déplacement

WHEN ... DO ...	si..., alors...
WHENEVER ... DO	à chaque fois que..., alors...

840 D
NCU 571840 D
NCU 572

FM-NC

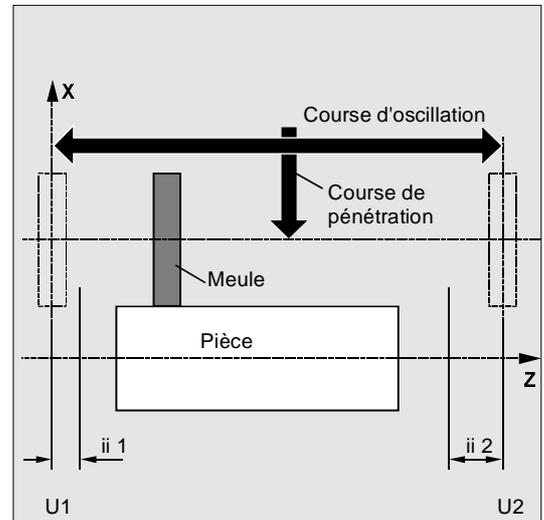
810D
NCU 573

Oscillation commandée par action synchrone

Avec ce type d'oscillation, le mouvement de pénétration n'est autorisé qu'au point d'inversion de sens ou dans des zones d'inversion de sens bien définies.

Selon les besoins, l'oscillation peut

- soit être poursuivie
- soit être arrêtée jusqu'à ce que la pénétration soit entièrement effectuée.



Procédure

1. Fixer les paramètres pour l'oscillation

Les paramètres pour l'oscillation sont à fixer dans le programme avant le bloc de déplacement qui contient d'une part la correspondance entre l'axe de pénétration et l'axe d'oscillation et d'autre part l'instruction avec les paramètres de pénétration (voyez "Oscillation asynchrone").

2. Définir les actions synchrones au déplacement

Sont effectués par le biais des actions synchrones :

- **Supprimer le mouvement de pénétration**, jusqu'à ce que l'axe d'oscillation se trouve à l'intérieur d'une zone d'inversion (ii1, ii2) ou à un point d'inversion (U1, U2).
- **Arrêter l'oscillation** pendant la pénétration au point d'inversion de sens.
- **Redémarrer l'oscillation** lorsque la pénétration partielle est terminée.
- **Définir le lancement de la pénétration partielle suivante.**

3. Affecter les axes d'oscillation et de pénétration et définir la pénétration totale et partielle.

11.2 Oscillation commandée par des actions synchrones



840 D
NCU 571



840 D
NCU 572
NCU 573



FM-NC



810D



Affectation des axes d'oscillation et de pénétration: OSCILL

`OSCILL[axe d'oscillation] = (axe de pénétration1, axe de pénétration2, axe de pénétration3)`

L'instruction OSCILL permet d'effectuer les affectations d'axe et le démarrage de l'oscillation.

Trois axes de pénétration peuvent être affectés au maximum à un axe d'oscillation.



Avant le démarrage de l'oscillation, les conditions synchrones pour le comportement des axes doivent être définies.



Définir les pénétrations: POSP

`POSP[axe de pénétration] = (pos. de fin, pénétr. partielle, mode)`

Avec l'instruction POSP, on fournit les indications suivantes à la commande :

- la pénétration totale (par le biais de la position de fin)
- la valeur de la pénétration partielle au point d'inversion de sens ou dans la zone d'inversion de sens
- le comportement lorsque la position de fin est atteinte (par le biais du mode)

Mode = 0 Pour les deux dernières pénétrations partielles, la distance restant à parcourir jusqu'à la destination est divisée en deux incréments identiques (préréglage).

Mode = 1 Toutes les pénétrations partielles sont identiques. Elles sont calculées à partir de la pénétration totale.

11.2 Oscillation commandée par des actions synchrones



840 D
NCU 571



840 D
NCU 572
NCU 573



FM-NC



810D



Les actions synchrones

De façon tout à fait générale, on utilise pour l'oscillation les actions synchrones au déplacement énumérées ci-après.

Des exemples sont donnés qui pourront vous servir de base pour réaliser vos propres programmes d'oscillation.



Dans certains cas particuliers, les conditions synchrones peuvent également être programmées différemment.



Mots-clés

WHEN ... DO ...	si..., alors...
WHENEVER ... DO	à chaque fois que..., alors...



Avec les instructions décrites en détail ci-après, vous pouvez réaliser les fonctions suivantes :

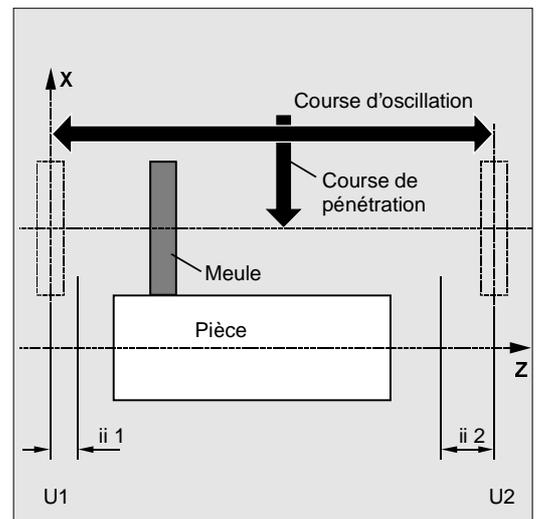
1. Pénétration au point d'inversion de sens.
2. Pénétration dans la zone d'inversion de sens.
3. Pénétration aux deux points d'inversion de sens.
4. Arrêt de l'oscillation au point d'inversion de sens.
5. Redémarrage de l'oscillation.
6. Inhibition de la pénétration partielle suivante.

Toutes les actions synchrones mentionnées ici à titre d'exemple sont basées sur les hypothèses suivantes :

- point d'inversion de sens 1 < point d'inversion de sens 2
- Z = axe d'oscillation
- X = axe de pénétration



Pour plus d'explications sur les actions synchrones au déplacement, voir le chapitre 11.3.



11.2 Oscillation commandée par des actions synchrones



840 D
NCU 571



840 D
NCU 572
NCU 573



FM-NC



810D



Pénétration dans la zone d'inversion de sens

Le mouvement de pénétration doit débiter à l'intérieur d'une zone d'inversion de sens, avant que le point d'inversion de sens ne soit atteint.

Ces actions synchrones inhibent le mouvement de pénétration jusqu'à ce que l'axe d'oscillation se trouve dans une zone d'inversion de sens.



Avec les hypothèses formulées (voir ci-dessus), les instructions sont les suivantes :



Zone d'inversion de sens 1 :

```
WHENEVER $AA_IW[Z] > $SA_OSCILL_REVERSE_POS1[Z] + ii1 DO $AA_OVR[X] = 0
```

A chaque fois que
supérieure
alors

la position courante de l'axe d'oscillation dans le SCP est
au début de la plage d'inversion de sens 1,
mettre la correction axiale de l'axe de pénétration à 0%.

Zone d'inversion de sens 2 :

```
WHENEVER $AA_IW[Z] < $SA_OSCILL_REVERSE_POS2[Z] + ii2 DO $AA_OVR[X] = 0
```

A chaque fois que
inférieure
alors

la position courante de l'axe d'oscillation dans le SCP est
au début de la zone d'inversion de sens 2,
mettre la correction axiale de l'axe de pénétration à 0%.

11.2 Oscillation commandée par des actions synchrones



840 D
NCU 571



840 D
NCU 572
NCU 573



FM-NC



810D



Pénétration au point d'inversion.

Tant que l'axe d'oscillation n'a pas atteint le point d'inversion, l'axe de pénétration n'effectue aucun déplacement.



Avec les hypothèses formulées (voir ci-dessus), les instructions sont les suivantes :

Point d'inversion 1 :

```
WHENEVER $AA_IW[Z] <> $SA_OSCILL_REVERSE_POS1[Z] DO $AA_OVR[X]=0 ->
-> $AA_OVR[Z]=100
```

A chaque fois que supérieure ou inférieure alors et	la position courante de l'axe d'oscillation Z dans le SCP est à la position du point d'inversion de sens 1 mettre la correction axiale de l'axe de pénétration X à 0% la correction axiale de l'axe d'oscillation Z à 100%.
--	--

Point d'inversion 2 :

Pour le point d'inversion 2 :

```
WHENEVER $AA_IW[Z] <> $SA_OSCILL_REVERSE_POS2[Z] DO $AA_OVR[X]=0 ->
-> $AA_OVR[Z]=100
```

A chaque fois que supérieure ou inférieure alors et	la position courante de l'axe d'oscillation Z dans le SCP est à la position du point d'inversion de sens 2, mettre la correction axiale de l'axe de pénétration X à 0% la correction axiale de l'axe d'oscillation Z à 100%.
--	---

11.2 Oscillation commandée par des actions synchrones



840 D
NCU 571



840 D
NCU 572



FM-NC



810D

NCU 573



Arrêt de l'oscillation au point d'inversion

L'axe d'oscillation s'arrête au point d'inversion de sens, le mouvement de pénétration commence au même moment.

L'oscillation est poursuivie lorsque la pénétration est entièrement effectuée.

Cette action synchrone peut également être utilisée pour redémarrer le mouvement de pénétration si celui-ci a été arrêté par une action synchrone précédente encore active.



Avec les hypothèses formulées (voir ci-dessus), les instructions sont les suivantes :

Point d'inversion 1 :

```
WHENEVER $SA_IW[Z]==$SA_OSCILL_REVERSE_POS1[Z]DO $AA_OVR[Z]=0 ->
-> $AA_OVR[X] = 100
```

A chaque fois que égale alors et	la position courante de l'axe d'oscillation dans le SCP est à la position d'inversion de sens 1, mettre la correction axiale de l'axe d'oscillation à 0%. la correction axiale de l'axe de pénétration à 100%.
---	---

Point d'inversion 2 :

```
WHENEVER $SA_IW[Z] ==$SA_OSCILL_REVERSE_POS2[Z]DO $AA_OVR[Z]= 0 ->
-> $AA_OVR[X]=100
```

A chaque fois que égale alors et	la position courante de l'axe d'oscillation dans le SCP est à la position d'inversion de sens 2, mettre la correction axiale de l'axe d'oscillation à 0%. la correction axiale de l'axe de pénétration à 100%.
---	---

11.2 Oscillation commandée par des actions synchrones



840 D
NCU 571



840 D
NCU 572
NCU 573



FM-NC



810D



Exploitation en ligne du point d'inversion de sens

Si une variable d'exécution marquée par \$\$ figure côté droit dans la comparaison, alors les deux variables seront exploitées en continu au rythme de la période d'appel de l'interpolateur et comparées entre elles.



Pour plus d'information à ce sujet, consultez le chapitre "Actions synchrones au déplacement".



Redémarrage de l'oscillation.

Cette action synchrone sert à reprendre le mouvement d'oscillation, quand la pénétration partielle est terminée.



Avec les hypothèses formulées (voir ci-dessus), les instructions sont les suivantes :

```
WHENEVER $AA_DTEPW[X]==0 DO $AA_OVR[Z]= 100
```

A chaque fois que égale alors	la distance restant à parcourir par l'axe de pénétration X dans le SCP est à zéro, mettre la correction par commutateur de l'axe d'oscillation à 100%.
-------------------------------------	--

11.2 Oscillation commandée par des actions synchrones



840 D
NCU 571



840 D
NCU 572



FM-NC



810D
NCU 573



Pénétration partielle suivante

Lorsque la pénétration a été effectuée, il convient d'éviter que la pénétration partielle suivante ne démarre trop tôt.

On utilise pour ce faire un memento spécifique au canal (`$AC_MARQUE[indice]`) qui est mis à 1 à la fin de la pénétration partielle (distance partielle restant à parcourir \equiv 0) et qui est effacé à la sortie de la zone d'inversion de sens. Une action synchrone empêche ensuite le mouvement de pénétration suivant.



Selon les hypothèses formulées (voir ci-dessus), les instructions sont les suivantes pour le point d'inversion de sens 1 par exemple :

1. Mise à 1 du memento :

```
WHENEVER $AA_DTEPW[X] == 0 DO $AC_MARQUE[1]=1
```

A chaque fois que la distance restant à parcourir par l'axe de pénétration X dans le SCP est égale à zéro, alors mettre à 1 le memento avec l'indice 1.

2. Effacer le memento

```
WHENEVER $AA_IW[Z]<>$SA_OSCILL_REVERSE_POS1[Z] DO $AC_MARQUE[1]=0
```

A chaque fois que la position courante de l'axe d'oscillation Z dans le SCP est supérieure ou inférieure à la position du point d'inversion de sens 1 alors mettre le memento 1 à 0.

3. Inhiber la pénétration

```
WHENEVER $AC_MARQUE[1]==1 DO $AA_OVR[X]=0
```

A chaque fois que le memento 1 est égal à 1, alors mettre la correction axiale de l'axe de pénétration X à 0%.

11.2 Oscillation commandée par des actions synchrones



840 D
NCU 571



840 D
NCU 572
NCU 573



FM-NC

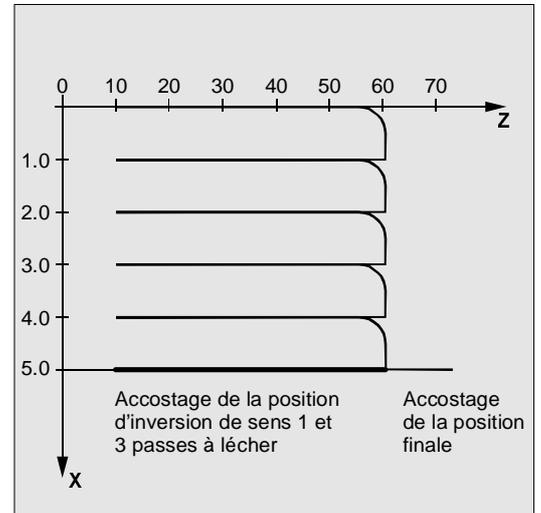


810D



Exemple de programmation

Il ne doit pas y avoir de pénétration au point d'inversion de sens 1. Au point d'inversion de sens 2, la pénétration doit déjà commencer à la distance $ii2$ du point d'inversion de sens et l'axe d'oscillation doit attendre la fin de la pénétration partielle. L'axe Z est axe d'oscillation et l'axe X axe de pénétration.



Section de programme

1. Paramètres pour l'oscillation

DEF INT $ii2$	définir la variable pour la zone d'inversion de sens 2
OSP1[Z]=10 OSP2[Z]=60	définir les points d'inversion de sens 1 et 2
OST1[Z]=0 OST2[Z]=0	point d'inversion de sens 1 : arrêt précis fin point d'inversion de sens 2 : arrêt précis fin
FA[Z]=150 FA[X]=0.5	avance de l'axe d'oscillation Z, avance de l'axe de pénétration X
OSCTRL[Z]=(2+8+16,1)	désactiver l'oscillation au point d'inversion de sens 2 ; après effacement de la distance restant à parcourir, passes à lécher et accostage de la position de fin ; après effacement de la distance restant à parcourir, accostage de la position d'inversion de sens correspondante
OSNC[Z]=3	3 passes à lécher
OSE[Z]=70	position de fin = 70
$ii2=2$	régler la zone d'inversion de sens
WAITP(Z)	autorise l'oscillation pour l'axe Z

11.2 Oscillation commandée par des actions synchrones

840 D
NCU 571840 D
NCU 572

FM-NC

810D
NCU 573

2. Actions synchrones au déplacement

```
WHENEVER $AA_IW[Z]<$SA_OSCILL_REVERSE_POS2[Z]-ii2 DO ->
-> $AA_OVR[X]=0 $AC_MARQUE[0]=0
```

A chaque fois que la position courante de l'axe d'oscillation Z dans le SCP est inférieure au début de la zone d'inversion de sens 2, alors mettre la correction axiale de l'axe de pénétration X à 0% et le memento avec l'indice 0 à la valeur 0.

```
WHENEVER $AA_IW[Z]>=$SA_OSCILL_REVERSE_POS2[Z] DO $AA_OVR[Z]=0
```

A chaque fois que la position courante de l'axe d'oscillation Z dans le SCP est supérieure ou égale à la position d'inversion de sens 2, alors mettre la correction par commutateur de l'axe d'oscillation Z à 0%.

```
WHENEVER $AA_DTEPW[X]==0 DO $AC_MARQUE[0]=1
```

A chaque fois que la distance restant à parcourir pour la pénétration partielle est égale à 0, alors mettre le memento avec l'indice 0 à la valeur 1.

```
WHENEVER $AC_MARQUE[0]==1 DO $AA_OVR[X]=0 $AA_OVR[Z]=100
```

A chaque fois que le memento avec l'indice 0 est égal à 1, alors mettre la correction axiale de l'axe de pénétration X à 0% afin d'éviter que la pénétration n'intervienne trop tôt (l'axe d'oscillation Z n'a pas encore quitté la zone d'inversion de sens 2 et l'axe de pénétration X est prêt pour une nouvelle pénétration), mettre la correction axiale de l'axe d'oscillation Z à 100% (pour supprimer la deuxième action synchrone).

-> doit être programmé dans un bloc

3. Démarrage de l'oscillation

```
OSCILL[Z]=(X) POSP[X]=(5,1,1)
```

démarrage des axes

L'axe X est affecté comme axe de pénétration à l'axe d'oscillation Z

L'axe X doit rejoindre la position de fin 5 par incréments de 1.

M30

Fin de programme

Poinçonnage et grignotage

12.1 Activation, désactivation.....	12-384
12.1.1 Instructions de langage.....	12-384
12.1.2 Utilisation des fonctions M.....	12-387
12.2 Segmentation automatique de la distance à parcourir.....	12-388
12.2.1 Segmentation de la distance à parcourir avec des axes à interpolation.....	12-389
12.2.2 Segmentation de la distance à parcourir avec des axes individuels.....	12-390
12.2.3 Exemples de programmation.....	12-392



840 D
NCU 572
NCU 573

12.1 Activation, désactivation

12.1.1 Instructions de langage



Programmation

PDELAYON
PON G... X... Y... Z...
PONS G... X... Y... Z...
PDELAYOF
SON G... X... Y... Z...
SONS G... X... Y... Z...
SPOF



Signification des paramètres

PON	Activation du poinçonnage
PONS	Activation du poinçonnage avec période d'appel de l'interpolateur
SON	Activation du grignotage
SONS	Activation du grignotage avec période d'appel de l'interpolateur
SPOF	Désactivation du poinçonnage, du grignotage
PDELAYON	Activation du poinçonnage avec temporisation
PDELAYOF	Désactivation du poinçonnage avec temporisation



Fonction

Activation/Désactivation du poinçonnage et du grignotage, PON/SON

Avec PON et SON, vous activez respectivement la fonction poinçonnage et la fonction grignotage.

SPOF met fin à toutes les fonctions spécifiques au poinçonnage et au grignotage.

Les instructions modales PON et SON s'excluent mutuellement, c'est-à-dire que PON désactive SON et vice versa.



840 D
NCU 572
NCU 573

Poinçonnage et grignotage avec période d'appel de l'interpolateur, PONS/SONS

Avec SONS et PONS, vous activez respectivement la fonction poinçonnage et la fonction grignotage. Mais contrairement à SON/PON – pilotage du poinçon au niveau d'interpolation – le pilotage du poinçon se fait ici par le biais de signaux au niveau servo (asservissement de position). Ceci vous permet de travailler avec une cadence de frappe plus rapide et donc une meilleure productivité.

Pendant le traitement des signaux dans la période d'appel de l'interpolateur, toutes les fonctions susceptibles de modifier la position des axes de poinçonnage ou de grignotage, sont verrouillées. Exemple : déplacement à la manivelle, modification de frames au niveau de l'AP, fonctions de mesure.

Pour le reste, PONS et SONS sont identiques à PON et SON.

Poinçonnage avec temporisation

PDELAYON temporise l'application du coup de poinçon. Cette instruction modale comporte une fonctionnalité préparatoire et se place de ce fait, en règle générale, avant PON.

Après PDELAYOF, le poinçonnage se poursuit de manière normale.



840 D
NCU 572
NCU 573



Exécution du poinçonnage

Déclenchement du premier coup de poinçon

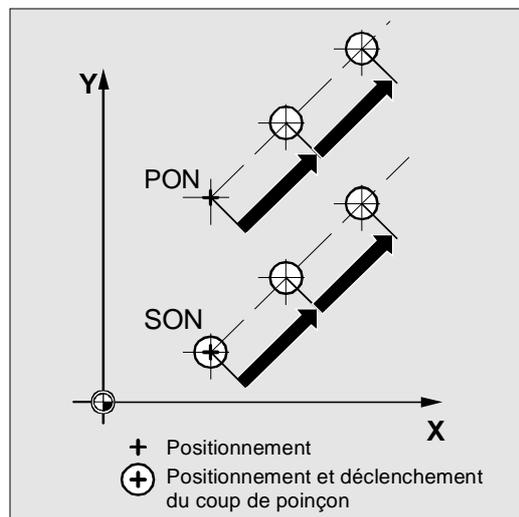
Selon qu'il s'agit du poinçonnage ou du grignotage, le déclenchement du premier coup de poinçon, après activation de la fonction, intervient avec une temporisation différente :

PON/PONS:

- Tous les coups de poinçon - y compris celui du premier bloc après l'activation - sont exécutés en fin de bloc.

SON/SONS:

- Le premier coup de poinçon après l'activation du grignotage, est toujours exécuté en début de bloc.
- Les autres coups de poinçon sont toujours libérés en fin de bloc.



Poinçonnage et grignotage sur place (sans déplacement)

Normalement, un coup de poinçon ne peut être déclenché que si le bloc contient une instruction de déplacement pour les axes de poinçonnage ou de grignotage (axes du plan actif).

Afin de pouvoir néanmoins libérer un coup de poinçon, tout en restant au même endroit, il y a lieu de programmer un déplacement nul pour l'un des axes de poinçonnage ou de grignotage.



Remarques complémentaires

Utilisation d'outils indexables

Pour positionner des outils indexables de façon tangentielle à la trajectoire programmée, utilisez la fonction Positionnement tangentiel.



840 D
NCU 572
NCU 573

12.1.2 Utilisation des fonctions M



En utilisant des macroprogrammes, vous pouvez utiliser des instructions M à la place des instructions de langage :

DEFINE M22 AS SON	Activation du grignotage
DEFINE M122 AS SONS	Activation du grignotage avec période d'appel de l'interpolateur
DEFINE M25 AS PON	Activation du poinçonnage
DEFINE M125 AS PONS	Activation du poinçonnage avec période d'appel de l'interpolateur
DEFINE M26 AS PDELAYON	Activation du poinçonnage avec temporisation
DEFINE M20 AS SPOF	Désactivation du poinçonnage, du grignotage
DEFINE M23 AS SPOF	Désactivation du poinçonnage, du grignotage



840 D
NCU 572
NCU 573

12.2 Segmentation automatique de la distance à parcourir



Programmation

SPP=
SPN=



Signification

SPP	Taille de la distance partielle (distance maximale entre coups de poinçon consécutifs) ; fonction à effet modal
SPN	Nombre de distances partielles dans un bloc ; fonction à effet non modal



Fonction

Segmentation en distances partielles

Quand le poinçonnage ou le grignotage est activé, SPP et SPN provoquent la segmentation en un nombre donné de distances partielles de même longueur (répartition équidistante) de l'ensemble du déplacement programmé pour les axes à interpolation. En interne, à chaque distance partielle correspond un bloc.

Nombre de coups de poinçon

En poinçonnage, le premier coup de poinçon intervient au point final de la première distance partielle ; par contre, en grignotage, il intervient au point de départ de la première distance partielle.

D'où découlent les formules suivantes:

Poinçonnage :

Nbre de coups = Nombre de distances partielles

Grignotage :

Nbre de coups = Nombre de distances partielles + 1

Fonctions auxiliaires

Les fonctions auxiliaires sont exécutées dans le premier des blocs générés.



840 D
NCU 572
NCU 573

12.2.1 Segmentation de la distance à parcourir avec des axes à interpolation



Procédure

Longueur de la distance partielle SPP

Avec SPP, vous indiquez la distance maximale entre les coups de poinçon successifs et par conséquent, la longueur maximale des distances partielles, en vue de la segmentation de l'ensemble du déplacement.

La désactivation de l'instruction se fait avec SPOF ou $SPP=0$.

Exemple :

```
N10 G1 SON X0 Y0
N20 SPP=2 X10
```

L'ensemble du déplacement de 10 mm est décomposé en 5 distances partielles de 2 mm chacune ($SPP=2$).



La répartition du déplacement par le biais de SPP donne toujours lieu à des déplacements équidistants : toutes les distances partielles sont de même longueur. Autrement dit, la taille de la distance partielle programmée (valeur de SPP) n'est valable que si, et seulement si, le quotient du déplacement total par la valeur de SPP est un entier.

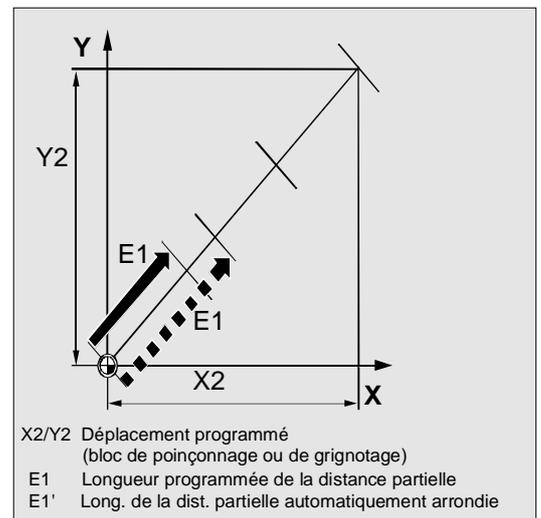
Si ce n'est pas le cas, la taille de la distance partielle est réduite, de façon interne, de manière à que le quotient soit un entier.

Exemple :

```
N10 G1 G91 SON X10 Y10
N20 SPP=3.5 X15 Y15
```

Avec un déplacement total de 15 mm et une distance partielle de 3,5 mm, on obtient un quotient qui n'est pas un nombre entier (4.28).

Une réduction de la valeur de SPP a donc lieu jusqu'à l'obtention d'un quotient entier. Il en résulte une taille de distance partielle de 3 mm.





840 D
NCU 572
NCU 573

Nombre de distances partielles SPN

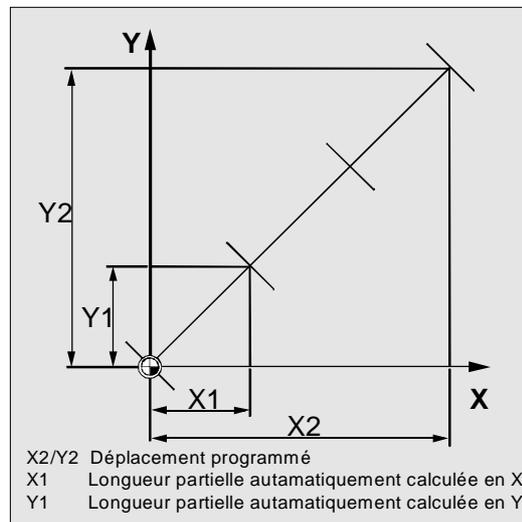
Avec SPN, on précise le nombre de distances partielles à générer sur l'ensemble du déplacement. La longueur des distances partielles est calculée automatiquement.

SPN ayant un effet non modal, il convient d'activer auparavant le poinçonnage ou le grignotage avec PON ou SON.

SPP et SPN dans un même bloc

Si vous programmez dans un même bloc, aussi bien la longueur (SPP) que le nombre de distances partielles (SPN), SPN sera valide seulement pour le bloc en question et SPP le sera aussi pour les blocs ultérieurs.

Si SPP a déjà été activé avant SPN, SPP sera à nouveau actif après le bloc contenant SPN.



Remarques complémentaires

Si le poinçonnage/grignotage font partie de la commande numérique, vous pouvez utiliser la programmation de la segmentation automatique en distances partielles avec SPN et SPF, indépendamment de cette technologie.

12.2.2 Segmentation de la distance à parcourir avec des axes individuels

Si, en plus des axes à interpolation, des axes individuels sont aussi configurés comme axes de poinçonnage/grignotage, ils supportent aussi la segmentation automatique de la distance à parcourir.

Comportement d'un axe individuel en présence de SPP

La longueur programmée de la distance partielle (SPP) s'adresse en premier lieu aux axes à interpolation. C'est la raison pour laquelle, la valeur SPP est ignorée dans un bloc qui contient un déplacement d'axe individuel et une valeur SPP, mais pas d'axe à interpolation.



840 D
NCU 572
NCU 573

Si, dans un bloc, des axes individuels aussi bien que des axes de à interpolation sont programmés, le comportement de l'axe individuel dépend du réglage du paramètre machine concerné.

1. Préréglage

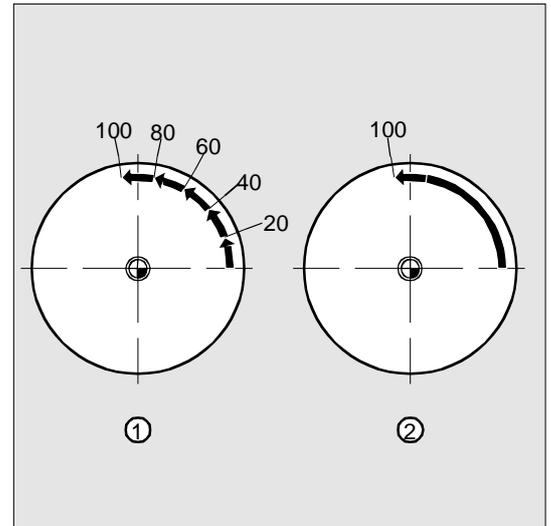
Le déplacement de l'axe individuel est reparti de façon égale sur les blocs intermédiaires générés par SPP.

Exemple :

N10 G1 SON X10 A0
N20 SPP=3 X25 A100

En fonction de la distance entre coups de poinçon de 3 mm, le déplacement global de l'axe X (axe à interpolation) de 15 mm donne lieu à 5 blocs.

L'axe A pivote de ce fait de 20° par bloc.



2. Axe individuel sans segmentation en distances partielles

L'axe individuel parcourt son déplacement global dans le premier des blocs générés.

3. Segmentation variable en distances partielles

Le comportement de l'axe individuel dépend de l'interpolation des axes à interpolation :

- Interpolation circulaire : Segmentation en distances partielles
- Interpolation linéaire : Pas de segmentation en distances partielles

Comportement avec SPN

Le nombre de distances partielles programmées reste valable, même si un axe à interpolation n'a pas été programmé en même temps.

Condition : l'axe individuel est défini comme axe de poinçonnage-grignotage.



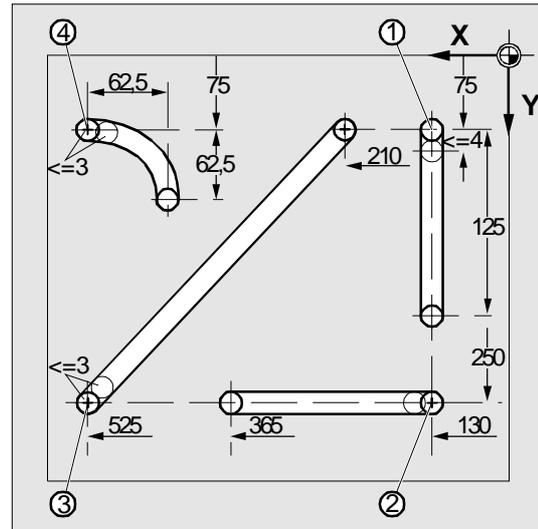
840 D
NCU 572
NCU 573

12.2.3 Exemples de programmation



Exemple de programmation 1

Les parcours de grignotage doivent être décomposés automatiquement en distances partielles identiques.



Section de programme

N100 G90 X130 Y75 F60 SPOF	Positionnement au point de départ 1
N110 G91 Y125 SPP=4 SON	Activation grignotage ; longueur max. de distance partielle pour segmentation auto. : 4 mm
N120 G90 Y250 SPOF	Arrêt grignotage; positionnt. au pt. départ 2
N130 X365 SON	Activation grignotage ; longueur max. de distance partielle pour segmentation auto. : 4 mm
N140 X525 SPOF	Arrêt grignotage; positionnt. au pt. départ 3
N150 X210 Y75 SPP=3 SON	Activation grignotage ; longueur max. de distance partielle pour segmentation auto. : 3 mm
N160 X525 SPOF	Arrêt grignotage ; positionnt. au pt. départ 4
N170 G02 X-62.5 Y62.5 I J62.5 SPP=3 SON	Activation grignotage ; longueur max. de distance partielle pour segmentation auto. : 3 mm
N180 G00 G90 Y300 SPOF	Arrêt grignotage

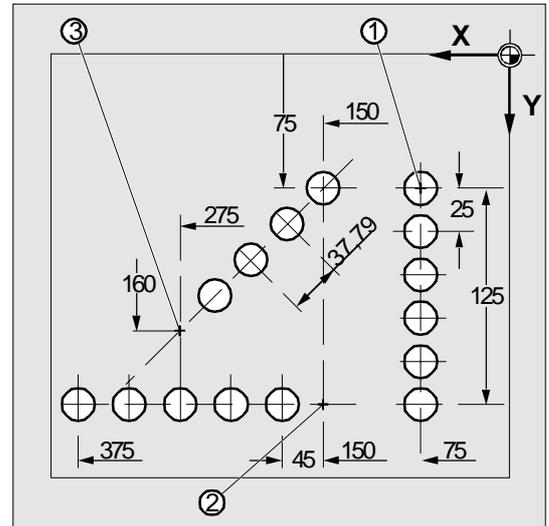


840 D
NCU 572
NCU 573



Exemple de programmation 2

On souhaite une segmentation automatique de la distance à parcourir pour chaque rangée de trous. Pour la segmentation, la longueur maximale de distance partielle (valeur de SPP) sera programmée.



Section de programme

N100 G90 X75 Y75 F60 PON	Positionnement au point de départ 1 ; activation poinçonnage ; poinçonner le trou.
N110 G91 Y125 SPP=25	Longueur maximale de distance partielle pour segmentation automatique : 25 mm
N120 G90 X150 SPOF	Arrêt poinçonnage ; positionnement au point de départ 2
N130 X375 SPP=45 PON	Activation poinçonnage ; longueur max. distance partielle pour segmentation auto. 45 mm
N140 X275 Y160 SPOF	Arrêt poinçonnage ; positionnement au point de départ 3
N150 X150 Y75 SPP=40 PON	Activation poinçonnage ; à la place de la longueur de distance partielle programmée de 40 mm, on utilise la valeur calculée de 37,79 mm.
N160 G00 Y300 SPOF	Arrêt poinçonnage ; positionnement

Autres fonctions

13.1 Fonctions spécifiques à un axe AXNAME, SPI, ISAXIS	13-396
13.2 Apprentissage de compensation : QECLRNON, QECLRNOF	13-396
13.3 Broche synchrone	13-399
13.4 Fonctions H.....	13-409
13.5 Définition des sorties de fonctions	13-410

13.1 Fonctions spécifiques à un axe AXNAME, SPI, ISAXIS



840 D
NCU 571



840 D
NCU 572
NCU 573



810D

13.1 Fonctions spécifiques à un axe AXNAME, SPI, ISAXIS



Programmation

```
AXNAME("AXE_TRANSVERSAL")
AX[AXNAME("Chaîne caractères")]
SPI(Numéro de broche)
ISAXIS(Numéro d'axe géométrique)
```



Signification des instructions

AXNAME	convertit, à son entrée, une chaîne de caractères en descripteur d'axe. La chaîne de caractères doit contenir un nom d'axe valide.
SPI	convertit un numéro de broche en descripteur d'axe. Le paramètre de transfert doit contenir un numéro de broche valide.
AX	descripteur d'axe variable
ISAXIS	vérifie si l'axe géométrique indiqué existe



Fonction

AXNAME est utilisé quand on crée des cycles à caractère général par exemple et que l'on ne connaît pas le nom d'un axe, (voyez également le chapitre 13.10, "Opérations sur les chaînes de caractères").

SPI est utilisé pour attribuer des fonctions d'axe à une broche, broche synchrone par exemple.

ISAXIS est utilisé dans des cycles de caractère général, pour vérifier l'existence d'un axe géométrique bien défini et pour éviter qu'un appel consécutif de \$P_AXNX soit suivi interrompu pour défaut.



Exemple de programmation

L'axe défini comme axe transversal doit effectuer un déplacement.

OVRA[AXNAME("axe transversal")]=10	axe transversal
AX[AXNAME("axe transversal")]=50.2	position de fin de l'axe transversal
OVRA[S1]=70	correction de la vitesse de la broche 1
IF ISAXIS(1) == FALSE GOTOF WEITER	abscisse présente ?
AX[\$P_AXN1]=100	déplacement abscisse

WEITER:

840 D
NCU 571840 D
NCU 572
NCU 573

810D

13.2 Apprentissage de compensation : QECLRNON, QECLRNOF



Signification des instructions

QECLRNON (axe . 1 , ...4)	activer la fonction "apprentissage de la compensation des défauts aux transitions entre quadrants"
QECLRNOF	désactiver la fonction "apprentissage de la compensation des défauts aux transitions entre quadrants"

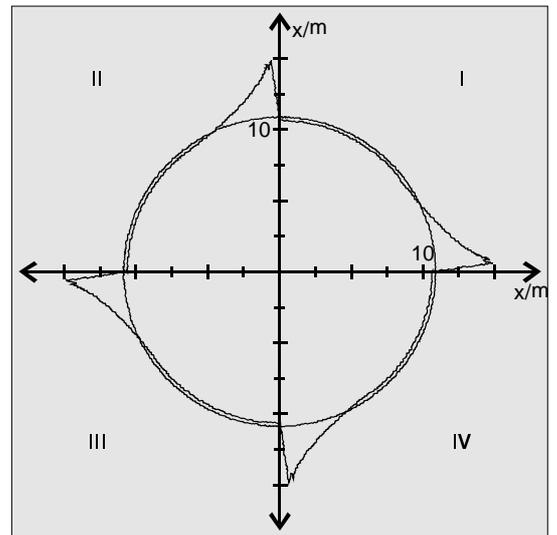


Fonction

La compensation des défauts aux transitions entre quadrants (QFK) réduit les défauts de contour qui apparaissent lors de l'inversion du sens de déplacement en raison de non linéarités mécaniques (p. ex. frottements, jeux) ou de la torsion.

Les paramètres de compensation peuvent être optimisés par la commande au cours d'une phase d'apprentissage à l'aide d'un réseau neuronal et les caractéristiques de compensation déterminées automatiquement.

L'apprentissage peut être effectué simultanément pour quatre axes au maximum.



Procédure

Les déplacements d'axe nécessaires à l'apprentissage sont générés à l'aide d'un programme CN qui contient les déplacements d'apprentissage sous forme d'un cycle d'apprentissage.

Premier apprentissage

Pour le premier apprentissage, à la mise en service, il existe sur la disquette qui contient le programme de base de l'AP, des programmes CN types pour s'initier aux déplacements d'apprentissage et à l'affectation des variables système QFK :

840 D
NCU 571840 D
NCU 572
NCU 573

810D

QECLRN.SPF	cycle d'apprentissage
QECDAT.MPF	programme CN type pour l'affectation des variables système et pour le paramétrage du cycle d'apprentissage
QECTEST.MPF	programme CN type pour le test de génération de forme circulaire.

Réapprentissage

Le réapprentissage permet d'optimiser les courbes caractéristiques déjà apprises. Les données qui se trouvent déjà dans la mémoire utilisateur sont alors utilisées.

Pour le réapprentissage, vous adaptez les programmes CN types à vos besoins. Les paramètres du cycle d'apprentissage (par exemple QECLRN.SPF) doivent, le cas échéant, être modifiés pour le "réapprentissage" :

- régler "mode d'apprentissage" = 1
- réduire le cas échéant le "nombre de cycles d'apprentissage"
- le cas échéant, activer "l'apprentissage par zone" et fixer les limites de ces zones

Activer le processus d'apprentissage:

QECLRNON

Le processus d'apprentissage proprement dit est à activer dans le programme CN avec l'instruction QECLRNON en indiquant les axes :

```
QECLRNON (X1, Y1, Z1, Q)
```

Les courbes caractéristiques ne sont modifiées que si cette instruction est active.

Désactiver l'apprentissage: QECLRNOF

Lorsque les déplacements d'apprentissage des axes souhaités sont terminés, on désactive le processus d'apprentissage pour tous les axes à la fois, avec QECLRNOF.



840 D
NCU 571



840 D
NCU 572
NCU 573

13.3 Broche synchrone



Programmation

COUPDEF (BA, BP, \ddot{U}_{BA} , \ddot{U}_{BP} , chang. bloc, coupl.)
 COUPDEL (BA, BP)
 COUPRES (BA, BP)
 COUPON (BA, BP, PS_{BA})
 COUPOF (BA, BP, POS_{BA} , POS_{BP})
 WAITC (BA, chang.bloc, BA, chang.bloc)



Signification des instructions

COUPDEF	créer, modifier un couplage défini par l'utilisateur
COUPON	activer le couplage
COUPOF	désactiver le couplage
COUPRES	remettre à zéro les paramètres de couplage
COUPDEL	effacer le couplage défini par l'utilisateur
WAITC	attendre la condition de synchronisme



Signification des paramètres

BA, BP	désignation de la broche asservie et de la broche pilote ; indication avec le n° de broche : par ex. S2
\ddot{U}_{BA} , \ddot{U}_{BP}	paramètres de transmission pour la broche asservie et la broche pilote préréglages = 1,0 ; indication du dénominateur en option
chang. bloc	mode de changement de bloc ; le changement de bloc a lieu : <ul style="list-style-type: none"> • "NOC" immédiatement (préréglage) • "FINE" au "synchronisme fin" • "COARSE" au "synchronisme grossier" • "IPOSTOP" lors de IPOSTOP (c.-à-d. quand la valeur de consigne du synchronisme est atteinte)
couplage	Type de couplage: couplage entre BA et BP <ul style="list-style-type: none"> • "DV" Couplage par la valeur de consigne (préréglage) • "AV" Couplage par la valeur réelle
PS_{BA}	décalage angulaire entre broche pilote et broche asservie
POS_{BA} , POS_{BP}	positions de désactivation de la broche asservie et de la broche pilote

13.3 Broche synchrone



840 D 840 D
 NCU 571 NCU 572
 NCU 573



Fonction

En mode "broche synchrone", il y a une broche pilote (BP) et une broche asservie (BA), qui forment la **paire de broches synchrones**. Quand le couplage est activé (mode "broche synchrone"), la broche asservie suit les déplacements de la broche pilote, en fonction du lien fonctionnel qui a été défini auparavant.

Sur les tours, cette fonction offre la possibilité d'un transfert de pièce à la volée de la broche 1 à la broche 2, par exemple pour la finition, en évitant les temps morts dus au changement d'ablocage.

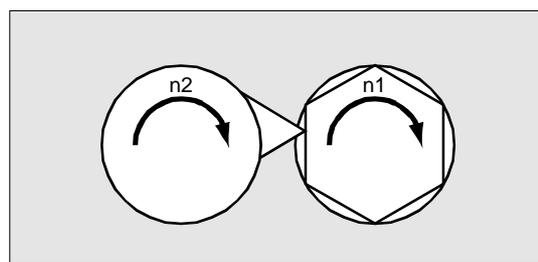
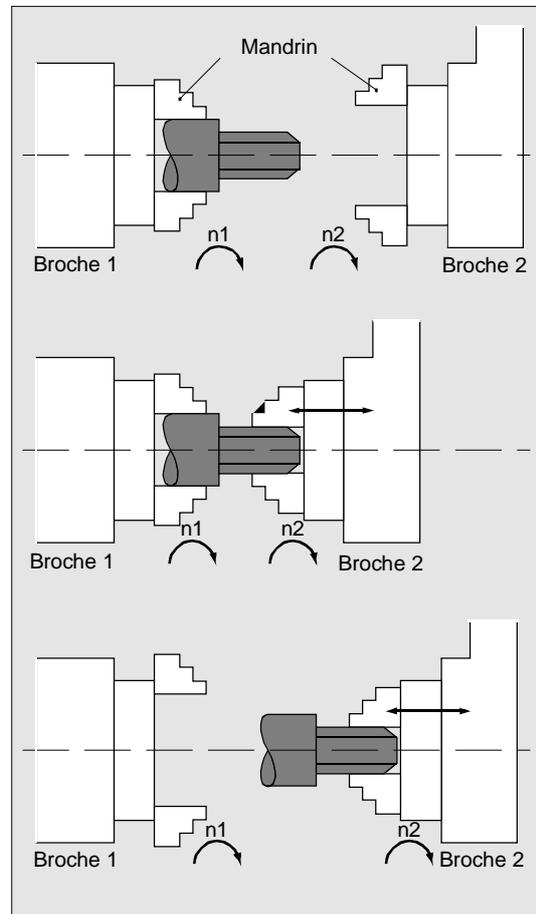
Le transfert de la pièce peut s'effectuer avec :

- synchronisme de vitesse de rotation ($n_{BA} = n_{BP}$)
- synchronisme de position ($\varphi_{BA} = \varphi_{BP}$)
- synchronisme de position avec décalage angulaire ($\varphi_{BA} = \varphi_{BP} + \Delta\varphi$)

Par ailleurs, la spécification d'un rapport de transmission k_{ij} entre la broche principale et une "broche d'outil" permet l'usinage de polygones (tournage de polygones).

Les paires de broches synchrones peuvent être aussi bien configurées de manière fixe pour chaque machine à l'aide de paramètres machine spécifiques à un canal ou définis en fonction de l'application par le programme pièce de la CNC.

Deux paires de broches synchrones sont utilisables simultanément par canal CN.



840 D
NCU 571840 D
NCU 572
NCU 573

Procédure

Définir la paire de broches synchrones :

Possibilités

Couplage configuré de manière fixe :

La broche pilote et la broche asservie sont définies dans les paramètres machine.

Dans le cas de ce couplage, les axes de la machine définis pour la BP et la BA ne peuvent pas être modifiés par le programme pièce CN. Par contre, le paramétrage du couplage peut avoir lieu dans le programme pièce avec COUPDEF (condition : pas de protection contre l'écriture).

Couplage défini par l'utilisateur :

L'instruction COUPDEF permet de créer et de modifier des couplages dans le programme pièce CN. Pour définir une nouvelle relation de couplage, il faut d'abord effacer un couplage existant défini par l'utilisateur avec COUPDEL.

Définir un nouveau couplage COUPDEF

Le paramétrage du sous-programme prédéfini est expliqué ci-après :

COUPDEF (BA, BP, \ddot{U}_{BA} , \ddot{U}_{BP} , chang. bloc, coupl.)

Broche asservie et broche pilote : BA et BP

Les descripteurs d'axe BA et BP définissent le couplage sans ambiguïté.

Ils doivent être programmés pour chaque instruction COUP. Les autres paramètres de couplage ne doivent être programmés que s'ils doivent être modifiés (action modale).

Exemple :

N... COUPDEF (S2 , S1 , \ddot{U}_{BA} , \ddot{U}_{BP})

signifie :

S2 = broche asservie, S1 = broche pilote

13.3 Broche synchrone



840 D
NCU 571



840 D
NCU 572
NCU 573

Positionnement de la broche asservie :

Possibilités

Quand le couplage de broches synchrones est activé, on peut toujours encore positionner des broches asservies dans une plage de $\pm 180^\circ$, indépendamment du déplacement déclenché par la broche pilote.

Positionnement SPOS

La broche asservie peut être interpolée avec SPOS=....

Pour de plus amples informations concernant SPOS voir le manuel de programmation Notions de Base.

Exemple :

```
N30 SPOS[2]=IC(-90)
```

Vitesse, accélération, FA, ACC, OVRA

Avec FA[SPI(S...)] ou FA[S...], ACC[SPI(S...)] ou ACC[S...] et OVRA[SPI(S...)] ou OVRA[S...], on peut programmer les vitesses de positionnement et les valeurs d'accélération des broches asservies (voyez le manuel de programmation "Notions de Base").

Changement de bloc programmable WAITC

Avec WAITC, on peut définir l'instant de changement de bloc selon différentes conditions de synchronisme (grossier, fin, IPOSTOP) pour la poursuite du programme après une modification des paramètres de couplage ou des opérations de positionnement par exemple.

On retarde le passage à de nouveaux blocs jusqu'à ce que la condition de synchronisme soit atteinte; le synchronisme gagne en rapidité.

En l'absence de conditions de synchronisme, l'instant de changement de bloc est tel qu'il a été programmé/configuré pour le couplage en question.



840 D
NCU 571



840 D
NCU 572
NCU 573

Exemples :

N200 WAITC

On attend les conditions de synchronisme pour toutes les broches asservies actives qui n'en ont pas.

N300 WAITC(S2, "FINE", S4, "COARSE")

On attend les conditions de synchronisme spécifiées "grossier" pour les broches asservies S2 et S4.

Rapport de transmission R_T

Le rapport de transmission est spécifié avec les valeurs de BA (numérateur) et de BP (dénominateur).

Possibilités :

- la broche asservie et la broche pilote tournent à la même vitesse ($n_{BA} = n_{BP}$; R_T positif)
- rotation dans le même sens ou en sens opposé (R_T négatif) entre la BA et la BP
- la broche asservie et la broche pilote tournent à des vitesses différentes ($n_{BA} = R_T \cdot n_{BP}$; $R_T \neq 1$)

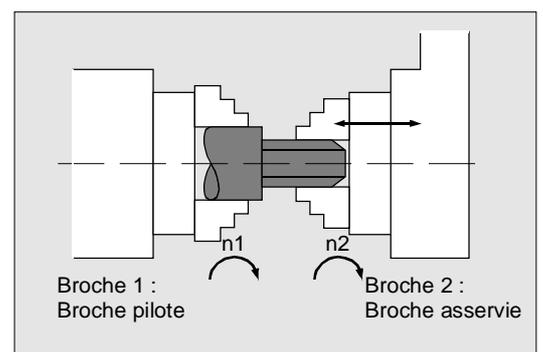
Application : tournage de polygones

Exemple :

N... COUPDEF(S2, S1, 1.0, 4.0)

signifie :

La broche pilote S2 et la broche asservie S1 tournent avec un rapport de transmission de 0.25.



- Le numérateur doit toujours être programmé. Si aucun dénominateur n'est indiqué, celui-ci est toujours mis à "1".
- Le rapport de transmission peut également être modifié au cours du déplacement pendant que le couplage est activé.

13.3 Broche synchrone



840 D
NCU 571



840 D
NCU 572
NCU 573

Instant de changement de bloc

Lors de la définition d'un couplage, on a les possibilités suivantes pour le changement de bloc :

" NOC "	immédiat (préréglage)
" FINE "	au "synchronisme fin"
" COARSE "	au "synchronisme grossier"
" IPOSTOP "	à IPOSTOP (c.à.d. quand la valeur de consigne de synchronisme est atteinte)

Pour l'indication du mode de changement de bloc, il suffit d'indiquer les lettres imprimées en gras.

Le mode de changement de bloc est à effet modal !

Type de couplage

" DV "	couplage par la valeur de consigne entre BA et BP (préréglage)
" AV "	couplage par la valeur réelle entre BA et BP

Le type de couplage est à effet modal.



Attention

La modification du type de couplage ne peut se faire que lorsque le couplage est désactivé !

840 D
NCU 571840 D
NCU 572
NCU 573

Activation du mode "broche synchrone"

- Activation aussi rapide que possible du couplage avec une référence angulaire quelconque entre BP et BA :

N ... COUPON (S2 , S1)

- Activer avec décalage angulaire POS_{BA}
Couplage synchronisé en position pour les pièces profilées.
 POS_{BA} se réfère à la position 0° de la broche pilote dans le sens de rotation positif.

Plage de valeurs POS_{BA} : $0^\circ \dots 359,999^\circ$

COUPON (S2 , S1 , 30)

Vous pouvez ainsi modifier le décalage angulaire, alors que le couplage est déjà actif.

Désactivation du mode "broche synchrone" COUPOF

Trois variantes sont possibles :

- Désactivation aussi rapide que possible du couplage ; le changement de bloc est validé immédiatement :

COUPOF (S2 , S1)

- Désactivation après le passage de positions d'arrêt; le changement de bloc est validé uniquement après le dépassement des positions d'arrêt POS_{BA} et éventuellement POS_{BP} .

Plage de valeurs $0^\circ \dots 359,999^\circ$:

COUPOF (S2 , S1 , 150)

COUPOF (S2 , S1 , 150 , 30)



840 D 840 D
 NCU 571 NCU 572
 NCU 573

Effacement d'un couplage COUPDEL

Un couplage de broches synchrones défini par l'utilisateur est à effacer si vous voulez en redéfinir un, alors que tous les couplages configurables possibles (1 ou 2) sont déjà utilisés.

N ... COUPON (S2,S1)

SPI(2) = broche asservie, SPI(1) = broche pilote



L'effacement est possible uniquement si le couplage est désactivé (COUPOF).



Un couplage configuré de manière fixe n'est pas effaçable avec COUPDEL.

Remise à zéro des paramètres de couplage COUPRES

Avec l'instruction "COUPRES"

- les paramètres spécifiés dans les paramètres machine et les données de réglage sont activés (couplage configuré de manière fixe)
- les préreglages sont activés (couplage défini par l'utilisateur)

Les paramètres programmés avec COUPDEF (y compris le rapport de transmission) sont alors perdus.

N ... COUPRES (S2,S1)

S2 = broche asservie, S1 = broche pilote

840 D
NCU 571840 D
NCU 572
NCU 573

Variables système

Etat courant de couplage Broche asservie

Avec la variable système axiale ci-après, vous pouvez lire dans le programme pièce CN l'état courant de couplage de la broche asservie :

```
$AA_COUP_ACT[ BA ]
```

BA = Descripteur d'axe pour broche asservie, avec numéro de broche, par ex. S2.

La valeur lue a la signification suivante pour la broche asservie :

0 : pas de couplage actif

4 : couplage de broche synchrone actif

Décalage angulaire courant

Le décalage entre les positions de consigne de la BA et de la BP peut être lu dans le programme pièce CN avec la variable système axiale suivante :

```
$AA_COUP_OFFS[ S2 ]
```

Le décalage de position en valeur réelle peut être lu avec :

```
$VA_COUP_OFFS[ S2 ]
```

BA = Descripteur d'axe pour broche asservie, avec numéro de broche, par ex. S2.



Si le déblocage des régulateurs est supprimé alors que le couplage et le fonctionnement en poursuite sont actifs, le décalage de position change par rapport à la valeur initialement programmée lorsque la libération des régulateurs est à nouveau validée. Dans ce cas, le décalage de position modifié peut être lu et éventuellement corrigé dans le programme pièce CN.

13.3 Broche synchrone

840 D
NCU 571840 D
NCU 572
NCU 573

Exemple de programmation

Usinage avec broche pilote et broche asservie

	;broche pilote = broche maître = broche 1
	;broche asservie = broche 2
N05 M3 S3000 M2=4 S2=500	;la broche pilote tourne à 3000/min, la broche asservie à 500/min
N10 COUPDEF (S2, S1, 1, 1, "NOC", "Dv")	;déf. du couplage ; peut être déterminé dès la configuration
...	
N70 SPCON	;mettre la broche pilote en asservissement de position (couplage par valeur de consigne)
N75 SPCON(2)	; mettre la broche asservie en asservissement de position
N80 COUPON (S2, S1, 45)	;réaliser le couplage, au vol, à la position de décalage = 45 degrés
...	
N200 FA [S2] = 100	;vitesse de positionnement = 100 degré/min.
N205 SPOS[2] = IC(-90)	;se déplacer d'une correction de 90 degrés en sens négatif
N210 WAITC(S2, "Fine")	;attendre le synchronisme "fin"
N212 G1 X... Y... F...	;usinage
...	
N215 SPOS[2] = IC(180)	Se déplacer d'une correction de 180 degré en sens positif
N220 G4 S50	;durée d'attente = 50 tours de la broche maître
N225 FA [S2] = 0	;activer la vitesse telle qu'elle a été configurée (MD)
N230 SPOS[2]=IC(-7200)	;20 tours en sens négatif avec vitesse telle que configurée
...	
N350 COUPOF (S2, S1)	;découpler au vol, S=S2=3000
N355 SPOSA[2] = 0	;arrêter BA à 0 degré
N360 G0 X0 Y0	
N365 WAITS(2)	;attendre la broche 2
N370 M5	; arrêter la broche asservie
N375 M30	

840 D
NCU 571840 D
NCU 572
NCU 573

13.4 Fonctions H

Les fonctions H permettent de transférer des informations à l'AP (automate programmable) pour déclencher certaines commutations.

Les valeurs des fonctions H sont des nombres de type REAL.

Exemple :

```
N10 G0 X20 Y50 H3=-11.3
```



La signification des fonctions est définie par le constructeur de machines.



Procédure

Nombre de fonctions par bloc CN

Trois fonctions H peuvent être programmées au maximum dans un bloc CN.



Remarque :

Les préréglages suivants sont effectués par le constructeur des machines.

Sorties de fonctions vers l'AP

Les fonctions suivantes peuvent être transférées vers l'AP (automate programmable)

- numéro d'outil T
- correcteur d'outil D
- avance F
- vitesse de rotation de broche S
- fonctions M et H

Il est possible de spécifier si les fonctions indiquées doivent être transférées à l'AP pendant l'exécution et quelles réactions doivent être déclenchées par l'AP dans certaines conditions.

Nombre de sorties de fonction par bloc CN
10 sorties de fonctions sont programmables au maximum dans un bloc CN.

13.5 Définition des sorties de fonctions



840 D
NCU 571



840 D
NCU 572



FM-NC



810D
NCU 573

Groupement

Les fonctions indiquées peuvent être regroupées. Pour certaines instructions M (cf. Fonctions M dans le manuel de programmation "Notions de base") la répartition en groupes est pré-réglée.

13.5 Définition des sorties de fonctions

Sorties de fonctions pendant les déplacements

Le transfert d'informations vers l'AP et l'attente des réactions correspondantes prend un certain temps et agit, par conséquent, sur les déplacements.

Pour chaque groupe de fonctions ou pour chaque fonction isolée, on définit si la sortie doit être déclenchée :

- avant le déplacement,
- pendant le déplacement,
- après le déplacement.

Sorties de fonctions en contournage



Les sorties de fonctions effectuées avant les déplacements interrompent le contournage (G64/G641) et génèrent un arrêt précis pour le bloc précédent.



Les sorties de fonctions effectuées après les déplacements interrompent le contournage (G64/G641) et génèrent un arrêt précis pour le bloc courant.



L'attente d'un signal d'acquiescement en provenance de l'AP peut également conduire à l'interruption du contournage, par exemple à l'interruption des séquences de fonctions M dans les blocs avec des trajectoires extrêmement courtes.

Sorties rapides de fonctions, QU

Les fonctions qui n'ont pas été configurées comme étant à sortie rapide peuvent être définies comme telles pour des sorties isolées, par le biais du mot-clé QU.

840 D
NCU 571840 D
NCU 572
NCU 573

L'exécution du programme se poursuit sans attente de l'acquittement de l'exécution de la fonction supplémentaire (l'acquittement du transfert est attendu).

Ceci permet d'éviter les points d'arrêt et les interruptions inutiles des déplacements.



Programmation

M=QU (...)

H=QU (...)

Exemples :

N10 H=QU(735)

Sortie rapide pour H735

N10 G1 F300 X10 Y20 G64

N20 X8 Y90 M=QU(7)

M7 a été programmée avec sortie rapide, si bien que le contournage (G64) n'est pas interrompu.



Utilisez cette fonction seulement dans des cas spécifiques, car la synchronisation peut être modifiée, par exemple en liaison avec d'autres sorties de fonctions.

Programmes de chariotage personnalisés

14.1 Fonctions évoluées pour le chariotage	14-414
14.2 Préparation du contour - CONTPRON.....	14-415
14.3 Point d'intersection de deux éléments de contour - INTERSEC	14-419
14.4 Exécution d'un élément de contour de la table - EXECTAB	14-421
14.5 Calcul des données d'un cercle - CALCDAT	14-422

14.1 Fonctions évoluées pour le chariotage



840 D
NCU 571



840 D
NCU 572
NCU 573



FM-NC



810D

14.1 Fonctions évoluées pour le chariotage



Programmes de chariotage personnels

Pour le chariotage, on vous propose des cycles d'usinage prêts à l'usage. En plus, vous avez la possibilité de créer vos propres cycles de chariotage grâce aux fonctions élaborées suivantes.

CONTPRON	Activer la préparation du contour, sous forme tabulaire
INTERSEC	Déterminer l'Intersection entre deux éléments de contour
EXECTAB	Exécution bloc par bloc des éléments de contour d'une table
CALCDAT	Calculer les rayons et les centres



Ces fonctions sont utilisables de façon universelle pas seulement pour le chariotage.

840 D
NCU 571840 D
NCU 572
NCU 573

FM-NC



810D

14.2 Préparation du contour - CONTPRON



Programmation

CONTPRON (NOMTAB, TYPUSIN, NN)
EXECUTE (ERREUR)



Signification des paramètres

CONTPRON	Activer la préparation du contour
NOMTAB	Nom de la table de contour
TYPUSIN	Paramètres pour type d'usinage : "G" : Cylindrage: usinage intérieur "L" : Cylindrage: usinage extérieur "N" : Dressage: usinage intérieur "P" : Dressage: usinage extérieur
NN	Nombre de détalonnages dans une variable de résultat de type INT
EXECUTE	Fin de la préparation du contour
ERREUR	Variable de signalisation d'erreur, type INT 1 = erreur ; 0 = pas d'erreur



Fonction

Les blocs qui font suite à l'instruction CONTPRON décrivent le contour à préparer.

Les blocs ne sont pas exécutés mais rangés dans la table de contour.

A chaque élément de contour correspond une ligne de la table de contour à deux dimensions.

Le nombre de détalonnages qui a été déterminé est indiqué en retour.

Avec EXECUTE, on arrête la préparation du contour et on retourne dans le mode d'exécution normal.

Exemple :

```
N30 CONTPRON(...)
N40 G1 X... Z...
N50...
N100 EXECUTE(...)
```

14.2 Préparation du contour - CONTPRON



840 D
NCU 571



840 D
NCU 572



FM-NC



810D
NCU 573

Remarques complémentaires

Conditions préalables à l'appel

Avant tout appel de CONTPRON, il faut :

- accoster un point de départ qui permet un usinage sans risque de collision,
- désactiver la correction du rayon de la plaquette G40.

Instructions de déplacements autorisées, systèmes de coordonnées

Seules les instructions G0 à G3 sont autorisées pour la programmation du contour, ainsi que les instructions chanfrein et arrondi/congé.

L'instruction "Cercle défini par point intermédiaire" génère un défaut.

Il en est de même pour les fonctions SPLINE, POLYNOME, FILETAGE.

Les modifications du système de coordonnées par activation d'un frame ne sont pas autorisées entre CONTPRON et EXECUTE.

Fin de la préparation du contour

Après l'écriture du contour, en appelant EXECUTE (ERREUR), vous effectuez un retour dans le déroulement normal du programme et mettez fin à la préparation du contour. Une signalisation en retour avec 1 = erreur ou 0 = pas d'erreur (le contour a pu être préparé sans erreur) est faite dans la variable spécifiée.

Éléments de détalonnage

La description des différents éléments de détalonnage peut être réalisée sous la forme d'un sous-programme ou de blocs isolés, au choix.

840 D
NCU 571840 D
NCU 572
NCU 573

FM-NC



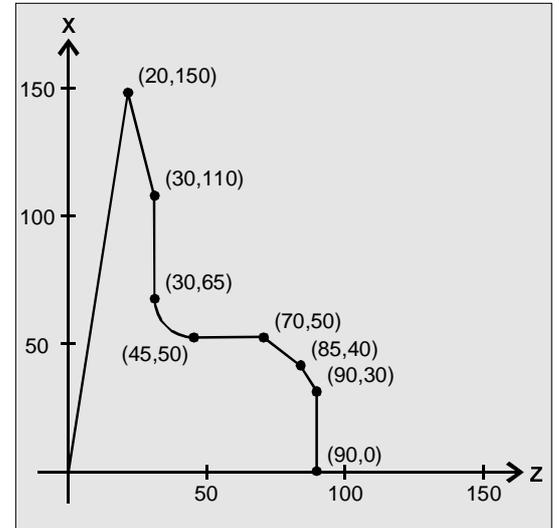
810D



Exemple de programmation

Création d'une table de contour avec

- le nom TABC
- 30 éléments de contour max. (arcs de cercle, segments de droite)
- une variable pour le nombre d'éléments de détalonnage générés
- une variable pour les messages d'erreur



Programme pièce CN

```
N10 DEF REAL TABC[30,11]
```

Table de contour nommée TABC avec par ex. 30 éléments de contour max.
La longueur 11 est fixe.

```
N20 DEF INT NBDETAL
```

Variable pour le nombre d'éléments de détalonnage, nommée NBDETAL

```
N30 DEF INT ERREUR
```

Variable pour acquittement
0= pas d'erreur, 1 = erreur

```
N40 G18
```

```
N50 CONTPRON (TABC, "L", NBDETAL)
```

Appel de la préparation du contour

```
N60 G1 X150 Z20
```

N60 à N120 : description du contour

```
N70 X110 Z30
```

```
N80 X50 RND=15
```

```
N90 Z70
```

```
N100 X40 Z85
```

```
N110 X30 Z90
```

```
N120 X0
```

```
N130 EXECUTE (ERREUR)
```

Fin du remplissage de la table de contour et retour au mode de programme normal

```
N140 ...
```

Suite du traitement de la table

14.2 Préparation du contour - CONTPRON

840 D
NCU 571840 D
NCU 572
NCU 573

FM-NC



810D



Table TABC correspondante

(0)	(1)	(2)	(3)	(4)	(5)	(6)	(7)	(8)	(9)	(10)
7	7	11	0	0	20	150	0	82.40535663	0	0
0	2	11	20	150	30	110	-	104.0362435	0	0
							1111			
1	3	11	30	110	30	65	0	90	0	0
2	4	13	30	65	45	50	0	180	45	65
3	5	11	45	50	70	50	0	0	0	0
4	6	11	70	50	85	40	0	146.3099325	0	0
5	7	11	85	40	90	30	0	116.5650512	0	0
6	0	11	90	30	90	0	0	90	0	0
0	0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0	0

Signification du contenu des colonnes

- (0) Index pointant l'élément de contour suivant (le numéro de ligne de la même table)
- (1) Index pointant l'élément de contour précédent
- (2) Codage des modes de contour pour le déplacement
Valeurs possibles pour X = abc
a = 10² G90 = 0 G91 = 1
b = 10¹ G70 = 0 G71 = 1
c = 10⁰ G0 = 0 G1 = 1 G2 = 2 G3 = 3
- (3), (4) Point de départ de l'élément de contour
(3) = abscisse, (4) = ordonnée dans le plan courant
- (5), (6) Point final de l'élément de contour
(5) = abscisse, (6) = ordonnée dans le plan courant
- (7) Afficheur max/min: identifie les maxima et minima locaux des contours
- (8) Valeur maximum entre éléments de contour et abscisses (lors d'opérations en longitudinal), et respectivement ordonnées (lors d'opérations de dressage).
L'angle dépend du type d'usinage programmé.
- (9), (10) Coordonnées du centre de l'élément de contour, si c'est un arc de cercle.
(9) = abscisse, (10) = ordonnée

14.3 Point d'intersection de deux éléments de contour - INTERSEC



840 D
NCU 571



840 D
NCU 572
NCU 573



FM-NC



810D

14.3 Point d'intersection de deux éléments de contour - INTERSEC



Programmation

VARIB=INTERSEC (NOMTAB1[n1], NOMTAB2[n2], NOMTAB3)



Signification des paramètres

VARIB	Variable d'état TRUE : existence d'une intersection FALSE : pas d'intersection
NOMTAB1 [n1]	Nom de table et élément de contour n1 de la première table
NOMTAB2 [n2]	Nom de table et élément de contour n2 de la seconde table
NOMTAB3	Nom de table pour les coordonnées du point d'intersection dans le plan courant (G17 à G19)



Fonction

INTERSEC détermine le point d'intersection de deux éléments de contour normés de la table de contour créée avec CONTPRON. L'état indiqué montre s'il existe une intersection (TRUE : existence d'une intersection) ou non (FALSE : pas d'intersection).



Remarques complémentaires

Une variable doit avoir été définie avant son utilisation.

14.3 Point d'intersection de deux éléments de contour - INTERSEC

840 D
NCU 571840 D
NCU 572
NCU 573

FM-NC



810D



Exemple de programmation

Il s'agit de déterminer l'intersection de l'élément de contour 3 de la table TABC1 et de l'élément de contour 7 de la table TABC2. Les coordonnées de l'intersection, dans le plan courant, sont rangées dans COUPE (premier élément = abscisse, second élément = ordonnée).

S'il n'existe pas d'intersection, il y a saut vers PASINTER (aucune intersection trouvée).

DEF REAL TABC1 [12, 11]	Table de contour 1
DEF REAL TABC2 [10, 11]	Table de contour 2
DEF REAL COUPE[2]	Table des points d'intersection
DEF BOOL ISPOINT	Variable d'état
...	
N10 ISPOINT=INTERSEC (KTAB1[3],KTAB2[7],COUPE)	Appel de la fonction Intersection des éléments de contour
N20 IF ISPOINT==FALSE GOTOF PASINTER	Saut vers PASINTER
...	

840 D
NCU 571840 D
NCU 572
NCU 573

FM-NC



810D

14.4 Exécution d'un élément de contour de la table - EXEC TAB



Programmation

EXEC TAB (NOMTAB[n])



Signification des paramètres

NOMTAB[n]

Nom de la table avec le numéro de l'élément



Fonction

Avec l'instruction EXEC TAB, vous pouvez exécuter, bloc par bloc, les éléments de contour d'une table qui a été créée avec l'instruction CONTPRON par exemple.



Exemple de programmation

Avec le sous-programme EXEC TAB, les éléments de contour de la table TABC sont exécutés bloc par bloc. Lors de l'appel, les éléments de 0 à 2 sont transmis successivement.

N10 EXEC TAB (TABC[0])

Exécuter l'élément 0 de la table TABC

N20 EXEC TAB (TABC[1])

Exécuter l'élément 1 de la table TABC

N30 EXEC TAB (TABC[2])

Exécuter l'élément 2 de la table TABC

14.5 Calcul des données d'un cercle - CALCDAT840 D
NCU 571840 D
NCU 572
NCU 573

FM-NC



810D

14.5 Calcul des données d'un cercle - CALCDAT**Programmation**

VARIB = CALCDAT(PT[n,2],NB,RES)

**Signification des paramètres**

VARIB	Variable d'état TRUE = cercle, FALSE = pas de cercle
PT[n, 2]	Points utilisés pour le calcul n = nombre de points (3 ou 4) ; 2 = indication des 2 coordonnées de point
NB	Nombre de points utilisés pour le calcul : 3 ou 4
RES[3]	Variable pour le résultat : coordonnées du centre du cercle et rayon 0 = abscisse, 1 = ordonnée du centre de cercle; 2 = rayon

**Fonction**

Calcul du rayon et du centre du cercle à partir de 3 ou de 4 points connus.

Les points indiqués doivent être distincts.

En cas d'indication de 4 points qui ne se trouvent pas exactement sur un cercle, des valeurs moyennes sont calculées pour le centre du cercle et le rayon.

840 D
NCU 571840 D
NCU 572
NCU 573

FM-NC

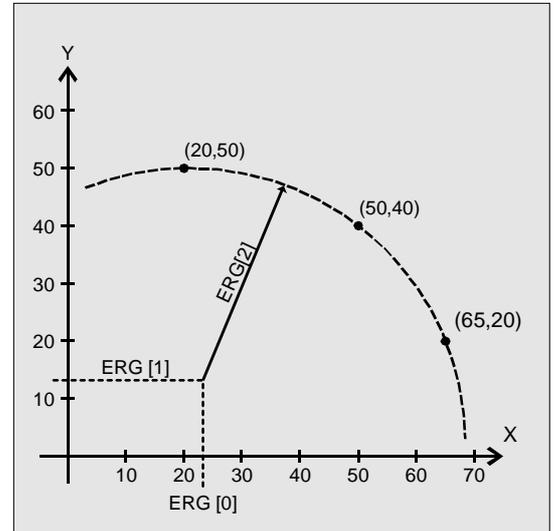


810D



Exemple de programmation

On veut savoir si trois points sont situés sur un arc de cercle.



```
N10 DEF REAL
```

Définition des points

```
PT[3,2]=(20,50,50,40,65,20)
```

```
N20 DEF REAL RES[3]
```

Résultat

```
N30 DEF BOOL ETAT
```

Variable d'état

```
N40 ETAT = CALCDAT(PT,3,RES)
```

Appel des données calculées

```
N50 IF ETAT == FALSE GOTOF ERREUR
```

Saut sur l'erreur

Tables

15.1 Liste des instructions.....	15-426
15.2 Liste des adresses	15-450
15.3 Liste des fonctions G/Fonctions préparatoires.....	15-459
15.4 Liste des sous-programmes prédéfinis	15-469
15.5 Liste des variables système (programmes pièce)	15-489
15.6 Liste des variables système.....	15-496
15.6.1 Paramètres R.....	15-496
15.6.2 Frames.....	15-496
15.6.3 Données du porte-outil.....	15-496
15.6.4 Zones de protection spécifiques à un canal.....	15-497
15.6.5 Entrées/sorties.....	15-499
15.6.6 Lecture et écriture de variables AP.....	15-499
15.6.7 Temps.....	15-500
15.6.8 Etats des canaux	15-500
15.6.9 Variables pour trajectoire.....	15-501
15.6.10 Valeurs polynomiales pour action synchrone.....	15-503
15.6.11 Variables système spécifiques à un axe.....	15-504
15.6.12 Données des broches.....	15-508
15.6.13 Variables système pour Safety Integrated.....	15-510
15.6.14 Valeurs programmées (synchrones au prétraitement des blocs)	15-510
15.6.15 Ordres vers/du canal.....	15-511

15.1 Liste des instructions

- ¹ Réglage par défaut en début de programme (préréglage de la commande à la livraison, en absence d'une autre programmation.
- ³ Points finaux absolus: à effet modal; points finaux relatifs : à effet non modal; sinon modal/non modal en fonction de la syntaxe de la fonction G.
- ⁴ Les paramètres IPO agissent de manière relative en tant que centres d'arcs de cercle. Ils peuvent être programmés de manière absolue avec AC. Avec une autre signification (par ex. pas de filet) la modification d'adresse est ignorée.
- ⁵ Mot-clé non valide pour SINUMERIK FM-NC/810D
- ⁶ Mot-clé non valide pour SINUMERIK FM-NC/810D/NCU571
- ⁷ Mot-clé non valide pour SINUMERIK 810D
- ⁸ L'utilisateur OEM a la possibilité d'introduire deux modes d'interpolation supplémentaires. Il peut également modifier les dénominations.
- ⁹ Mot-clé valide uniquement pour SINUMERIK FM-NC
- ¹⁰ L'écriture de l'adresse avec extension n'est pas admise pour ces fonctions.
- Remarque : Dans ce chapitre, "s" est à prendre dans le sens de "single block" (fonction non modale).

Nom	Signification	Affectation de valeur	Description, commentaire	Syntaxe	modal/non modal	Groupe
:	Numéro de bloc - Bloc principal (voir N)	0 ... 9999 9999 nombres entiers uniquement, sans signe	désignation particulière de blocs - à la place de N... ; ce bloc doit contenir toutes les instructions nécessaires à la phase d'usinage complète qui doit suivre.	par ex.. :20		
α	Angle d'un axe incliné	0, ..., 359.9999°				
A	axe	réel			m,s ³	
A2 ⁵	Orientation d'outil : angle d'Euler	réel			s	
A3 ⁵	Orientation d'outil : Composantes des vecteurs d'orientation	réel			s	
A4 ⁵	Orientation d'outil pour début de bloc	réel			s	
A5 ⁵	Orientation d'outil pour fin de bloc; composantes des vecteurs normaux	réel			s	
ABS	Valeur absolue (fonction mathématique)			ABS (p); p = paramètre		
AC	Introduction de cotes absolues	0, ..., 359.9999°		...=AC	s	
AC_TIMER	Variable de temporisation			AC_Timer[n]		
AC ⁵	Accélération axiale (acceleration axial)	réel,sans signe			m	
ACN	Indication de cotes absolue pour axes rotatifs, accoster position dans sens négatif			A=ACN(...) B=ACN(...) C=ACN(...)	s	

Nom	Signification	Affectation de valeur	Description, commentaire	Syntaxe	modal/non modal	Groupe
ACOS	Arcus-Cosinus (fonction trigonométrique)			ACOS (p); p = paramètre		
ACP	Indication de cotes absolue pour axes rotatifs, accoster position dans sens positif			A=ACP(...) B=ACP(...) C=ACP(...)	s	
ADIS	Distance de transition pour fonctions tangentielles G1, G2, G3, ...	réel, sans signe			m	
ADISPOS	Distance de transition pour vitesse rapide G0	réel, sans signe			m	
ALF	Sens de déplacement programmable	Nombre entier, sans signe	n : codé dans la commande Attention : Danger de collision !	ALF=n; n = 1, ..., 8: sens de déplacement		
AMIRROR	Fonction miroir programmable (additive mirror)			AMIRROR X0 Y0 Z0 ; bloc séparé	s	3
NBR	Nombre de points utilisés pour le calcul					
AP	Angle polaire (angle polar)	0, ..., $\pm 360^\circ$			m, s ³	
aplim	Paramètre de périodicité	0 ou 1	0: Table non périodique; 1: T. périodique			
VPaprox	Solution approximative pour la valeur pilote, quand il n'a pas été possible de déterminer une valeur pilote univoque pour une valeur asservie donnée.					
AR	Angle au centre (angle circular)	0, ..., 360°			m, s ³	
AROT	Rotation programmable (Additive ROTation)	rot. autour du 1er axe géo. : -180° .. 180° 2e axe géo. : -89.999° ... 90° 3e axe géo. : -180° .. 180°		AROT X... Y... Z... ; bloc AROT RPL= séparé	s	3
ASCALE	Mise à l'échelle programmable (Additive SCALE)			ASCALE X... Y... Z... ; bloc séparé	s	3
ASIN	Area-Sinus (fonction trigonométrique)			ASIN (p); p = paramètre		
ASPLINE ⁷	Spline Akima				m	1
ATRANS	Décalage additif programmable (Additive TRANSlation)			ATRANS X... Y... Z... ; bloc séparé	s	3
AV ¹	Type de couplage entre BA et BP : couplage par la valeur réelle		Modifier uniquement si couplage désactivé !		m	

15.1 Liste des instructions

Nom	Signification	Affectation de valeur	Description, commentaire	Syntaxe	modal/non modal	Groupe
AX	Descripteur d'axe variable	réel		par ex. AX[AXNAME("AXE TRANSVERSAL")]=50.2	m,s ³	
AXNAME	Convertir une chaîne de caractères en descripteur d'axe		La chaîne de caractères d'entrée doit contenir un nom d'axe valide	AXNAME("axe transversal")		
B	Axe	réel			m,s ³	
B2 ⁵	Orientation d'outil : angle d'Euler	réel			s	
B3 ⁵	Orientation d'outil : Composantes des vecteurs d'orientation	réel			s	
B4 ⁵	Orientation d'outil pour début de bloc	réel			s	
B5 ⁵	Orientation d'outil pour fin de bloc; composantes des vecteurs normaux	réel			s	
BAUTO ⁷	Détermination du premier segment spline à l'aide des trois points suivants (begin not a knot)				m	19
BEARBART	Paramètre pour type d'exécution		longitudinale, G: usinage int.; L: usinage ext.; transversale, N: usinage int.; P: usinage ext.;			19
BNA T ^{1,7}	Raccordement naturel avec le premier bloc spline (begin natural)				m	19
BRISK ¹	Accélération résultante sous forme d'échelon				m	21
BRISKA	Activation de l'accélération sous forme d'échelon pour les axes programmés					
BSPLINE ⁷	Spline B				m	1
BTAN ⁷	Raccordement tangentiel avec le premier bloc spline (begin tangential)				m	19
C	Axe	réel			m,s ³	
C2 ⁵	Orientation d'outil : angle d'Euler	réel			s	
C3 ⁵	Orientation d'outil : Composantes des vecteurs d'orientation	réel			s	
C4 ⁵	Orientation d'outil pour début de bloc	réel			s	
C5 ⁵	Orientation d'outil pour fin de bloc; composantes des vecteurs normaux	réel			s	
CAC	Accostage de la position codée par déplacement absolu	1, ..., 60	n = N° de position	CAC(n)		
CACN	Accostage de la position codée par déplacement absolu dans le sens négatif		uniquement pour	CACN(n)		
CACP	Accostage de la position codée par déplacement absolu dans le sens positif		axes rotatifs	CACP(n)		

Nom	Signification	Affectation de valeur	Description, commentaire	Syntaxe	modal/non modal	Groupe
<i>CANCEL</i>	Mot-clé : Effacer action synchrone		n = Numéro d'identification de l'action synchrone	CANCEL[n,n, ...]		
<i>CDC</i>	Accostage de la position codée par le plus court chemin		uniquement pour axes rotatifs	CDC(n)		
<i>CDOF</i> ¹	Surveillance anticollision désactivée (collision detection OFF)				m	23
<i>CDON</i>	Surveillance anticollision activée (collision detection ON)				m	23
<i>CFC</i> ¹	Avance constante au niveau du contour (constant feed at contour)				m	16
<i>CFTCP</i>	Avance constante au point de référence du tranchant (trajectoire du centre de l'outil) (constant feed in tool-center-point)				m	16
<i>CFIN</i>	Avance constante au niveau des courbures concaves, accélération au niveau des courbures convexes (constant feed at internal radius)				m	16
<i>CHF</i>	Chanrein (chamfer)	réel, sans signe			s	
<i>CIC</i>	Accostage de la position codée par déplacement incrémental de n positions en avant (+) ou en arrière (-)			CIC(n)		
<i>CIP</i>	Interpolation circulaire avec point intermédiaire			CIP X... Y... Z... I1=... J1=... K1=...	m	1
<i>CLEARM</i>	Effacer mémentos dans canal propre		Sans effet sur l'exécution en cours Valide au-delà de reset et de Départ programme. Egalement programmable à partir d'une action synchrone	CLEARM(N° memento, N° memento, ...) CLEARM(0): Effacer tous les mémentos dans le canal		
<i>CLGOF</i> ⁶	Const. Désactivation de la vitesse de rotation pièce pour rectification sans centre (CenterLess Grinding OFF)					
<i>CLGON</i> ⁶	Const. Activation de la vitesse de rotation pièce pour rectification sans centre (CenterLess Grinding ON)	réel, sans signe				
<i>CLRINT</i>	Effacement des affectations d'interruption de la routine d'interruption		n = Numéro de la routine	CLRINT(n)		
<i>COMPLETE</i>	Sauvegarde des programmes d'initialisation					
<i>COMPOF</i> ¹	Désactivation du compactage				m	30
<i>COMPON</i>	Activation du compactage				m	30
<i>CONTPRON</i>	Activation de la préparation du contour					
<i>COARSE</i>	Changement de bloc : au "synchronisme grossier"				m	
<i>COS</i>	Cosinus (fonction trigonométrique)		p = paramètres	COS (p)		

15.1 Liste des instructions

Nom	Signification	Affectation de valeur	Description, commentaire	Syntaxe	modal/non modal	Groupe
<i>COUPDEF</i>	Créer, modifier un couplage défini par l'utilisateur		Chgt bloc : NOC: immédiat; FINE/COARSE: au "synchronisme	COUPDEF (BA, BP, Ü _{BA} , Ü _{BP} , chang. bloc, coupl.)		
<i>COUPDEL</i>	Effacer le couplage défini par l'utilisateur		fin/grossier". IPOSTOP: après synchronisme des valeurs de consigne	COUPDEL(BA, BP)		
<i>COUPOF</i>	Désactiver le couplage		Coupl. DV: Coupl. par val. consigne	COUPOF(BA, BP, POS _{BP} , POS _{BP})		
<i>COUPON</i>	Activer le couplage		AV: Coupl. par val. réelle PS _{BA} : Décalage angulaire entre BP et BA	COUPON(BA, BP, PS _{BA})		
<i>COUPRES</i>	Remettre à zéro les paramètres de couplage		POS _{BA} , POS _{BP} : Positions de désactivation de la BA et de la BP	COUPRES(BA, BP)		
<i>CPRECOF</i> ^{1,6}	Désactivation de la précision de contour programmable (Contour PRECision OFF)				m	39
<i>CPRECON</i> ⁶	Activation de la précision de contour programmable (Contour PRECision ON)				m	39
<i>CPROT</i>	Activation d'une zone de protection spécifique au canal		uniquement pour NCU 572/573	CPROT (n, state, xMov, yMov, zMov)		
<i>CPROTDEF</i>	Définir zones de protection spécifiques au canal		uniquement pour NCU 572/573	CPROTDEF (n, t, applim, applus, appminus)		
<i>CR</i>	Rayon de cercle (Circle Radius)	réel, sans signe			s	
<i>CSPLINE</i> ⁷	Spline cubique				m	1
<i>CTAB</i>	Valeur asservie correspondant à une valeur pilote		n = N° de la table de courbe; degré= Nom du paramètre de pente	CTAB(LW, n, degré, [AxeA, AxeP])		
<i>CTABDEF</i>	Définir le début de la table de courbe		n = N° de la table de courbe; applim = 0/1: la table est non périodique/périodique	CTABDEF(AxeA, AxeP, n, applim)		
<i>CTABDEL</i>	Effacer une table		n = N° de la table de courbe	CTABDEL(n)		
<i>CTABEND</i>	Définir la fin de la table			CTABDEL()		

Nom	Signification	Affectation de valeur	Description, commentaire	Syntaxe	modal/non modal	Groupe
<i>CTABINV</i>	Valeur pilote correspondant à une valeur asservie		n = N° de la table de courbe; degré= Nom du paramètre de pente	CTABINV(FW,aproxLW,n,degré,[AxeA,AxeP])		
<i>CUT2D</i> ¹	Correction d'outil 2 1/2D (CUTter compensation type 2Dimensional)				m	22
<i>CUT2DF</i>	Correction d'outil 2 1/2D (CUTter compensation type 2dimensional frame) ; La correction d'outil agit de façon relative par rapport au frame courant (plan incliné)				m	22
<i>CUT3DC</i> ⁵	Correction d'outil 3D Fraisage périphérique (CUTter compensation type 3dimensional circumference)				m	22
<i>CUT3DF</i> ⁵	Correction d'outil 3D Fraisage en bout (CUTter compensation type 3dimensional circumference)				m	22
<i>CUT3DFF</i> ⁵	Correction d'outil 3D Fraisage en bout avec orientation d'outil constante dépendant du frame courant (CUTter compensation type 3dimensional face frame)				m	22
<i>CUT3DFS</i> ⁵	Correction d'outil 3D Fraisage en bout avec orientation d'outil constante indépendant du frame courant (CUTter compensation type 3dimensional face)				m	22
<i>D</i>	Numéro de correcteur d'outil	1, ..., 9	contient les valeurs de correction pour un outil bien défini T... ; D0 → Valeurs de correction pur un outil	D...		
<i>DC</i>	Indication de cotes absolue pour axes rotatifs, accoster position en direct			A=DC(...) B=DC(...) C=DC(...)	s	
<i>DEF</i>	Définir les variables locales			par ex. DEF INT NOT_USED		
<i>DELD</i>	Effacement de la distance restant à parcourir, sans préparation			DELD ou DELD(axe1, ...)		
<i>DELT</i>	Effacement de l'outil		N°_DUPLO optionnel	DELT("TO",N°_DUPLO)		
<i>DELTG</i>	Effacement de la distance restant à parcourir, avec préparation			DELG ou DELG(axe1, ...)		
<i>DIAMOF</i> ¹	Désactivation de la programmation du diamètre (DIAMetral programming OFF)				m	29
<i>DIAMON</i>	Activation de la programmation du diamètre (DIAMetral programming ON)				m	29
<i>DISABLE</i>	Désactivation de la routine d'interruption			DISABLE(n); n = Numéro de la routine		
<i>DISC</i>	Surcourse pour cercle de raccordement correction de rayon d'outil	0, ..., 100			m	
<i>DISPLOF</i>	Inhiber l'affichage du bloc courant			PROC ... DISPLOF	m	

15.1 Liste des instructions

Nom	Signification	Affectation de valeur	Description, commentaire	Syntaxe	modal/non modal	Groupe
DISPR	Différence sur le contour pour repositionnement	réel,sans signe			s	
DISR	Distance de repositionnement	réel,sans signe			s	
DO	Mot-clé : Déclenche l'action quand la condition est remplie.					
DRFOF	Désactivation des décalages par manivelle (DRF)				m	
DRIVE ⁹	Accélération résultante fonction de la vitesse				m	21
DRIVEA ⁹	Activation de la courbe d'accélération infléchie pour les axes programmés					
N°_DUPL0	N° de l'outil de rechange	Integer				
DV ^{1,5}	Type de couplage entre BA et BP : couplage par la valeur de consigne		Modifier uniquement si couplage désactivé !		m	21
EAUTO ⁵	Détermination du dernier segment spline à l'aide des trois derniers points				m	20
ENABLE	Réactivation de la routine d'interruption			ENABLE(n); n = Numéro de la routine		
ENAT ^{1,5}	Raccordement naturel avec le bloc de déplacement suivant (End NATural)				m	20
RES	Résultat			par ex. RES[3]: coordonnées du centre du cercle et rayon : 0 = abscisse, 1 = ordonnée du centre du cercle, 2 = rayon		
ETAN ⁵	Raccordement tangentiel avec le bloc de déplacement suivant au début de la spline (End TANGential)				m	20
EVERY	Mot-clé : L'action est toujours exécutée une seule fois, quand la condition est remplie.					
EXECTAB	Exécuter les éléments de contour d'une talbe			EXECTAB (NOMTAB[n])	s	
EXECUTE	Mettre fin à la définition/préparation du contour		par ex. après CPROTDEF (définition des zones de prot.)			
EXP	Exposant (fonction mathématique)			EXP(p); p = paramètre		
F...	Avance			FNORM, FLIN, FCUB, FPO		

Nom	Signification	Affectation de valeur	Description, commentaire	Syntaxe	modal/non modal	Groupe
F	Valeur de l'avance (en liaison avec G4, on programme aussi un arrêt temporisé sous F)	0.001, ..., 99 999.999	Vitesse tangentielle outil/pièce; Unité en mm/min ou mm/tour en fonction de G94 ou G95	F...		
AA	Avance axiale (Feed Axial)	0.001, ..., 999999.999 mm/min, degré/min; 0.001, ..., 39999.9999 inch/min		AA[axe]=avance	m	
AxeA	Axe conjugué		Axe rotatif supplémentaire à asservissement tangentiel			
FCTDEF	Paramétrage de la fonction PUTFTOCF			FCTDEF(N° polynôme, LLimit, ULimit, a ₀ , a ₁ , a ₂ , a ₃)		
FCUB ⁶	Avance modifiable suivant spline cubique (Feed CUBic)			FCUB	m	37
FD	Avance tangentielle pour correction par manivelle (Feed DRF)	réel, sans signe			s	
FDA	Avance axiale pour correction par manivelle (Feed DRF Axial)	réel, sans signe			s	
ERREUR	Message d'erreur	Integer	0: pas d'erreur; 1: erreur			
FFWOF ¹	Désactivation de la commande anticipatrice (Feed Forward OFF)				m	24
FFWON	Activation de la commande anticipatrice (Feed Forward ON)				m	24
FGROUP	Définition de ou des axes avec avance tangentielle		F est valable pour tous les axes indiqués sous FGROUP	FGROUP (axe1, [axe2], ...)		
FINE	Changement de bloc : au "synchronisme fin"				m	
FL	Vitesse limite pour axes synchrones (Feed Limit)	réel, sans signe	l'unité réglée avec G93, G94, G95 (vitesse rapide max.) est valide	FL [axe] =...	m	
FLIN ⁶	Avance modifiable linéairement (feed linear)			FLIN	m	37
FMA	Avance synchrone axiale (Feed Multiple Axial)	réel, sans signe			m	
FNORM ^{1,6}	Avance normale selon DIN66025 (Feed NORMal)		Préréglage	FNORM	m	37
FP	Point fixe : numéro du point fixe à accoster	Nombre entier, sans signe			s	

15.1 Liste des instructions

Nom	Signification	Affectation de valeur	Description, commentaire	Syntaxe	modal/non modal	Groupe
FPO	Variation de l'avance tangentielle définie par un polynôme		L'adresse F décrit la variation de l'avance selon un polynôme, depuis la valeur courante jusqu'à la fin de bloc. Valeur finale ensuite modale	par ex. F = FPO(endfeed, quadf, cubf)		
FPR	Désignation axe rotatif	0.001 ... 999999.999		FPR (axe rotatif)		
FRAOF	Désactivation de l'avance par tour					
FPRAON	Activation de l'avance par tour					
FROM	Mot-clé : L'exécution de l'action est répétée aussi longtemps que l'action synchrone est active					
FTOC	Correcteur d'outil en ligne			FTOC(n° polynôme, EV, longueur, n° de canal, n° de broche)		
FTOCOF ^{1,6}	Désactivation de la correction d'outil fine en ligne (Fine Tool Offset Correction OFF)				m	33
FTOCON ⁶	Activation de la correction d'outil fine en ligne (Fine Tool Offset Correction ON)				m	33
FXS	Activation de l'accostage d'une butée (FiXed Stop)	Nombre entier, sans signe	1 = activer, 0 = désactiver		m	
FXST	Limite de couple pour accostage d'une butée (FiXed Stop Torque)	%	Indication facultative		m	
FXSW	Fenêtre de surveillance pour accostage d'une butée (FiXed Stop Window)	mm, inch ou degré	Indication facultative			
VA	Valeur asservie : Valeur de la position de l'axe asservi pour laquelle la valeur pilote devra être calculée.					

Nom	Signification	Affectation de valeur	Description, commentaire	Syntaxe	modal/non modal	Groupe
Fonctions G						
G	Fonction G (fonction préparatoire) Les fonctions G sont réparties en groupes G. On ne peut écrire qu'une seule fonction G d'un groupe dans un bloc. Une fonction G peut avoir un effet modal (jusqu'à ce qu'une autre fonction du même groupe vienne la relayer) ou peut agir uniquement dans le bloc dans lequel elle figure (effet non modal).	Uniquement valeurs entières définies		G...		
G0	Interpolation linéaire avec vitesse rapide		Instructions de déplacement	G0 X... Z...	m	1
G1 ¹	Interpolation linéaire avec avance			G1 X... Z... F...	m	1
G2	Interpolation circulaire dans le sens horaire			G2 X... Z... I... K... F... ; centre et point final G2 X... Z... CR=... F... ; rayon et point final G2 AR=... I... K... F... ; angle au centre et centre G2 AR=... X... Z... F... ; angle au centre et point final	m	1
G3	Interpolation circulaire dans le sens antihoraire			G3 ... ; sinon comme G2	m	1
G4	Arrêt temporisé		déplacement spécial	G4 F... ou G4 S... ; bloc séparé	s	2
G5	Interpolation circulaire avec point intermédiaire		Instruction de déplacement	G5 X... Z... IX=... KZ=... F...	m	1
G9	Arrêt précis - Ralentissement				s	11
G17 ¹	Choix du plan de travail X/Y		Pénétration radiale Z		m	6
G18	Choix du plan de travail Z/X		Pénétration radiale Y		m	6
G19	Choix du plan de travail Y/Z		Pénétration radiale X		m	6
G22	Indication du rayon				m	29
G23 ¹	Indication du diamètre				m	29
G25	Limite inférieure de la vitesse de rotation de broche		Mémoire écrire	G25 S... ; bloc séparé	s	3
G26	Limite supérieure de la vitesse de rotation de broche			G26 S... ; bloc séparé	s	3

Nom	Signification	Affectation de valeur	Description, commentaire	Syntaxe	modal/non modal	Groupe
G33	Interpolation hélicoïdale avec pas constant	0.001, ..., 2000.00 mm/tr	Instruction de déplacement	G33 Z... K... SF=... ; filetage cylindrique G33 X... I... SF=... ; filetage plan G33 Z... X... K... SF=... ; filetage conique (dans axe Z, course plus grande que dans axe X) G33 Z... X... I... SF=... ; filetage conique (dans axe X, course plus grande que dans axe Z)	m	1
G40 ¹	Désactivation de la correction du rayon d'outil				m	7
G41	Correction du rayon de l'outil à gauche du contour				m	7
G42	Correction du rayon de l'outil à droite du contour				m	7
G53	Inhibition du décalage d'origine réglable		décalages programmés inclus		s	9
G54	1er décalage d'origine réglable				m	8
G55	2ème décalage d'origine réglable				m	8
G56	3ème décalage d'origine réglable				m	8
G57	4ème décalage d'origine réglable				m	8
G60 ¹	Arrêt précis - Ralentissement				m	10
G63	Taraudage avec porte-taraud compensateur		déplacement spécial	G63 Z...	s	2
G64	Arrêt précis - contournage				m	10
G70	Cotes en Inch				m	13
G71 ¹	Cotes en métrique				m	13
G74	Accostage du point de référence		déplacement	G74 X... Z...; bloc séparé	s	2
G75	Accostage point fixe		spécial	G75 X... Z...; bloc séparé	s	2
G90 ¹	Introduction de cotes absolues			G90 X... Y... Z...(...) Y=AC(...) ou X=AC Z=AC(...)	m s	14
G91	Indication de cotes relatives			G91 X... Y... Z... ou X=IC(...) Y=IC(...) Z=IC(...)	m s	14
G93 ⁵	Avance en inverse de temps en 1/min				m	15
G94 ¹	Avance linéaire F en mm/min oder inch/min et °/min				m	15
G95	Avance par tour F en mm/tr ou inch/tr				m	15
G96	Activation de la vitesse de coupe constante			G96 S... LIMS=... F...	m	15
G97	Désactivation de la vitesse de coupe constante				m	15

Nom	Signification	Affectation de valeur	Description, commentaire	Syntaxe	modal/non modal	Groupe
G110	Programmation du pôle par rapport à la dernière position de consigne programmée		écrire en mémoire		s	3
G111	Programmation du pôle par rapport au système de coordonnées pièce courant				s	3
G112	Programmation du pôle par rapport au dernier pôle valable				s	3
G158	Décalage programmable			G158 X... Z...; bloc séparé	s	3
G331	Taraudage sans porte-taraud compensateur	±0.001, ...,	Instructions de déplacement		m	1
G332	Retrait (après G331)	2000.00 mm/tr			m	1
G450 ¹	Arc de raccordement		Comportement aux angles		m	18
G451	Point d'intersection des équidistantes		Correction du rayon d'outil		m	18
G500 ¹	Désactivation G54 - G5XX, remise à zéro du frame réglable				m	8
G505 G599	5. ... 99. décalage d'origine réglable				m	8
G601 ¹	Changement de bloc lors d'un arrêt précis fin		actif uniquement si G60 activé ou G9 avec arrondi de transition programmable		m	12
G602	Changement de bloc lors d'un arrêt précis grossier				m	12
G603	Changement de bloc à la fin du bloc d'interpolation				m	12
G641	Arrêt précis - contournage			G641 ADIS=...	m	10
G810 ¹ , ..., G819	Groupe de fonctions G réservé à l'utilisateur OEM.					31
G820 ¹ , ..., G829	Groupe de fonctions G réservé à l'utilisateur OEM.					32
GEOAX	Permutation des axes géométriques		n=0: extraire du groupe d'axes géométriques le canal d'axe indiqué, sans le remplacer n=1, 2 ou 3: Numéro de l'axe géométrique auquel doit être affecté un autre axe de canal	GEOAX([n,axe de canal],[n,axe de canal],...) GEOAX(): Appel de la configuration de base des axes géométriques		
GET	Reprendre axe ou broche			GET(nom d'axe,nom d'axe,...) ou GET(S...)		
GETD	Reprendre directement axe ou broche			GETD(nom d'axe,nom d'axe,...) ou GETD(S...)		
GETSELT	Lire le numéro d'outil sélectionné			GETSELT(N° T)		

15.1 Liste des instructions

Nom	Signification	Affectation de valeur	Description, commentaire	Syntaxe	modal/non modal	Groupe
<i>GETT</i>	Définir le n° T		N°_DUPLO optionnel	GETT("TO",N°_FRERE)		
<i>GUD</i>	Données utilisateur globales					
<i>GWPSOF</i>	Désactiver vitesse périphérique de meule constante (SUG)			GWPSOF (n° T.)	s	
<i>GWPSON</i>	Activer vitesse périphérique de meule constante (SUG)			GWPSON (N° T)	s	
<i>H...</i>	Transfert d'informations à l'AP	Réel	défini par le fabricant de la machine			
<i>I⁴</i>	Paramètres d'interpolation	Réel			s	
<i>I1</i>	Coordonnées de points intermédiaires	Réel			s	
<i>IC</i>	Indication de cotes relatives	0, ..., ±99999.999°		...=IC	s	
<i>ID</i>	Action synchrone en mode automatique		propre au programme	ID/IDS=n;	m	
<i>IDS</i>	Action synchrone dans chaque mode de fonctionnement		statique (global) sans ID/IDS Actions synchrones non modales	n = 1, ..., 255	m	
<i>INIT</i>	Sélection d'un programme en vue de son exécution dans un canal donné :		Si aucun mode d'acquittement n'est indiqué, l'acquittement synchrone est pris en compte.	INIT(n, "nom de programme", "q") n: numéro de canal q: mode d'acquittement q=n: sans mode d'acquittement q=s: synchrone		
<i>INITIAL</i>	Créer un fichier INI		_N_INITIAL_INI: Fichier d'initialisation			
<i>IPOSTOP</i>	Changement de bloc : après synchronisme des valeurs de consigne				m	
<i>ISAXIS</i>	Vérifier l'axe géométrique donné		pour éviter des erreurs consécutives	ISAXIS(numéro d'axe géométrique)		
<i>ISD</i>	Profondeur de pénétration (insertion depth)	réel			m	
<i>J⁴</i>	Paramètres d'interpolation	réel			s	
<i>J1</i>	Coordonnées de points intermédiaires	réel			s	
<i>JERKA</i>	Activer pour les axes programmés, le comportement à l'accélération, réglé à l'aide du paramètre machine.					
<i>K⁴</i>	Paramètres d'interpolation	réel			s	
<i>K1</i>	Coordonnées de points intermédiaires	réel			s	
<i>CONT</i>	Contournement du contour suite à la correction d'outil				m	17
<i>Coupl.</i>	Facteur de couplage	réel	Rapport entre course de l'axe conjugué / course de l'axe pilote	Valeur standard : 1		

Nom	Signification	Affectation de valeur	Description, commentaire	Syntaxe	modal/non modal	Groupe
SC	Lettres de codification du système de coordonnées		B = système de coordonnées de base (standard); P = système de coordonnées pièce			
L	Numéro de sous-programme	Entier, jusqu'à 7 chiffres	Les zéros de tête sont importants !	L...	s	
AxeP	Axe pilote		Axes à interpolation à partir desquels est définie la tangente pour l'asservissement	axeP1, axeP2		
LEAD ⁵	Angle d'avance	réel			m	
LEADOF	Désactiver le couplage par valeur pilote		n = N° de la table de courbe	LEADOF(AxeA,AxeP,n)		
LEADOFF	Désactiver le couplage tangentiel par valeur pilote			LEADOFF		
LEADON	Activer le couplage par valeur pilote		n = N° de la table de courbe	LEADON(AxeA,AxeP,n)		
LEADONP	Activer le couplage tangentiel par valeur pilote		Activer le type A : Activer le type B :	LEADONP(AxeP,pos. act.,Coupl) ou LEADONP(AxeP,pos. act.)		
LIFTFAST	Retrait rapide de l'outil					
LIMS	Limitation de la vitesse de rotation (LIMit Spindle Speed) avec G96	0.001 ... 99 999.999			m	
LN	Logarithme naturel (fonction mathématique)			LN(p); p = paramètre		
LOCK	Mot-clé : Bloquer une action synchrone		n = N° Id de l'action synchrone	LOCK[n,n, ...]		
VP	Valeur pilote : Valeur de la position de l'axe pilote pour laquelle une valeur asservie devra être calculée.					
M...	Opérations de commutation	0, ..., 99999999	au max. 5 fonctions M à définir par le fabricant de la machine			
M0 ¹⁰	Arrêt programmé					
M1 ¹⁰	Arrêt optionnel					
M2 ¹⁰	Fin du programme principal avec retour au début du programme					
M3	Rotation à droite de la broche maître					
M4	Rotation à gauche de la broche maître					
M5	Broche arrêt pour broche maître					

15.1 Liste des instructions

Nom	Signification	Affectation de valeur	Description, commentaire	Syntaxe	modal/non modal	Groupe
M6	Changement d'outil					
M17 ¹⁰	Fin du sous-programme					
M30 ¹⁰	Fin de programme, comme M2					
M40	Changement de vitesse automatique					
M41... M45	Rapport boîte de vitesse 1, ..., 5					
M70	Transition vers mode axe					
MCALL	Appel modal d'un sous-programme				m	
MEAC	Mesure continue sans effacement de la distance restant à parcourir		s: syst. de mes. : =0/néant : syst. de mesure actif; =1: système de mesure 1; =2: système de mesure 2; =3: les deux M.-S.	MEAC[axe]=(sm, mémoire des mesures, TE1, ..., TE4)		
MEASA	Mesure avec effacement de la distance restant à parcourir		m: mode de mes. : =0: interrompt tâche de mesure; =1: jusqu'à 4 événements déclencheurs activables simultanément; =2: jusqu'à 4 événements déclencheurs activables consécutivement TE.	MEASA[axe]=(sm, TE1, ..., TE4)		
MEAWA	Mesure sans effacement de la distance restant à parcourir		TE1, ...4: événement déclencheur : =1: front montant, palpeur 1; =-1: desc. Fl., MT1; =2: mont. Fl., MT2; =-2: desc. Fl., MT2.	MEAWA[axe]=(sm, TE1, ..., TE4)		
MEAS	Mesure avec palpeur à déclenchement (mesure)	Nombre entier, sans signe			s	
MEAW	Mesure avec palpeur à déclenchement sans effacement de la dist. restant à parcourir (measure without deleting distance to go)	Nombre entier, sans signe			s	
MIRROR	Fonction miroir programmable			MIRROR X0 Y0 Z0 ; bloc séparé	s	3
MOV	Lancer le déplacement		Valeur : >0: sens pos. ; <0: sens nég. ; ≡0: arrêter-déplacement	MOV[axe]=valeur		

Nom	Signification	Affectation de valeur	Description, commentaire	Syntaxe	modal/non modal	Groupe
N	Numéro de bloc - Bloc secondaire	0, ..., 9999 9999 nombres entiers uniquement, sans signe	utilisable pour désigner des blocs avec un numéro; figure au début d'un bloc	par ex. N20		
NEWT	Créer un nouvel outil		N°_DUPLO optionnel	NEWT("TO",N°_FRERE)		
NN	Nombre de détalonnages dans une variable de résultat	entier				
NOC ¹	Changement de bloc : immédiat				m	
NORM ¹	Réglage normal au point de départ, au point final pour la correction d'outil				m	17
NPROT	Appel de la zone de protection spécifique à la machine			NPROT (n,state,xMov, yMov,zMov)		
NPROTDEF	Définir les zones de protection spécifiques à la machine			NPROTDEF (n,t,applim,ap-plus,appminus)		
OEMIPO1 ^{6,8}	Interpolation OEM 1				m	1
OEMIPO2 ^{6,8}	Interpolation OEM 2				m	1
OFFN	Décalage de contour - normal	réel			m	
OMA1/2/3/4/5 ⁶	Adresse OEM 1/2/3/4/5	réel			m	
ORIC ^{1,6}	Les modifications d'orientation aux angles saillants s'ajoutent au bloc à élément circulaire à insérer (orientation change continuously)				m	27
ORID ⁶	Les modifications d'orientation sont exécutées avant le bloc à interpolation circulaire (orientation change discontinuously)				m	27
ORIMKS ⁶	Orientation d'outil dans le système de coordonnées machine (tool orientation in machine coordinate system)				m	25
ORIPATH ⁶	Orientation d'outil (tool ORientation PATH)				m	25
ORIS ⁵	Modification d'orientation (orientation smoothing factor)	réel	rapportée à la trajectoire		m	
ORIWKS ^{1,6}	Orientation d'outil dans le système de coordonnées pièce (tool orientation In workpiece coordinate system)				m	25
OS	Activation/désactivation de l'oscillation	entier, sans signe	1: activation axe; 2: désact. axe	OS[axe]=		
OSC ⁶	Lissage constant de l'orientation de l'outil				m	34
OSCILL	Affectation des axes d'oscillation activation de l'oscillation		axes : 1 - 3 axes de pénétration	OSCILL[axe]=(axe de pénétration1,axe de pénétration2,axe de pénétration3)	m	
OSCTRL	Options d'oscillation	entier, sans signe		OSCTRL[axe]=	m	
OSE	Oscillation: point final			OSE[axe]=	m	
OSNSC	Oscillation: Nombre de passes à lécher (oscillating: number spark out cycles)			OSNSC[axe]=	m	

15.1 Liste des instructions

Nom	Signification	Affectation de valeur	Description, commentaire	Syntaxe	modal/non modal	Groupe
OSOF ^{1,6}	Désactivation du lissage de l'orientation de l'outil				m	34
OSP1/2	Oscillation: point d'inversion gauche/droite (oscillating: Position 1/2)	réel		OSP1/2[axe]=	m	
OSS ⁶	Lissage de l'orientation de l'outil en fin de bloc				m	34
OSSE ⁶	Lissage de l'orientation de l'outil en début et fin de bloc				m	34
OST1/2	Oscillation: Temporisation à gauche/droite Point d'inversion :	réel	en secondes	OST1/2[axe]=	m	
OVR	Correction	1, ..., 200%			m	
OVRA	Correction avance axiale	1, ..., 200%		par ex. OVRA[SPI(S...)]	m	
P	Nombre d'exécutions de sous-programme	1 ... 9999, entier sans signe		par ex. L781 P... ; bloc séparé		
PCALL	Appel d'un sous-programme avec indication de chemin et paramètres			PCALL chemin/nom_programme (paramètre 1, ..., param. n)		
PDELAY-OFF ⁶	Temporisation de la désactivation du poinçonnage (Punch with DELAY OFF)				m	36
PDELAY-ON ^{1,6}	Temporisation de l'activation du poinçonnage (Punch with DELAY ON)				m	36
PT	Points utilisés pour le calcul			par ex. PT[n,2]: n: nombre de points (3 ou 4) 2 = indication de 2 coordonnées de point		
PL	Paramètre - intervalle - longueur	Réel, sans signe			s	
POLY ⁵	Interpolation polynomiale				m	1
PON ⁶	Activation du poinçonnage (Punch ON)		Antagoniste de SON	PON G... X... Y... Z...	m	35
PONS ⁶	Activation du poinçonnage dans la période d'appel de l'interpolateur (Punch ON Slow)		avec période d'appel de l'interp.	PONS G... X... Y... Z...	m	35
POS						
POSP	Définir les pénétrations totales et partielles		Mode=0: 2 pénétrations partielles restantes de même longueur: =1: toutes les pénétrations partielles de même longueur	POSP[axe de pénétration] = (pos. de fin, pénétr. partielle, mode)		
PRESETON	Préréglage des mémoires de valeurs réelles pour les axes programmés.		On programme resp. un descripteur d'axe et dans le paramètre suivant la valeur correspondante. max. 8 axes possibles	comme appel d'un sous-programme		

Nom	Signification	Affectation de valeur	Description, commentaire	Syntaxe	modal/non modal	Groupe
<i>PROC</i>	Sous-programme reprenant des paramètres du programme dont il est issu			par ex. PROC CONTOUR		
<i>PUTFTOC</i>	Ecriture discrète de la correction d'outil en ligne			PUTFTOC(valeur de référence, longueur 1_2_3, canal, broche)		
<i>PUTFTOCF</i>	Ecriture en continu de la correction d'outil en ligne			PUTFTOCF(n° polynôme, valeur de référence, longueur1_2_3, canal, broche)		
<i>PW</i>	Poids de point (point weight)	Nbre réel, sans signe			s	
<i>QECLRNOF</i>	Désactiver la fonction "apprentissage de la compensation des défauts aux transitions entre quadrants"/			QECLRNOF		
<i>QECLRNON</i>	Activer			QECLRNON(axe1, ..., 4)		
<i>QU</i>	Mot-clé: sortie rapide de fonction		Attention : Influence sur correspondance chronologique !	M=QU(...); H=QU(...)		
<i>R...</i>	Paramètres de calcul	±0.0000001, ..., 9999 9999	R0, ..., R99: livres; R100, ..., R249: paramètres de transfert pour cycles d'usinage			
<i>RELEASE</i>	libérer axe ou broche			RELEASE(nom d'axe, nom d'axe, ...) ou RELEASE(S...)	s	2
<i>RDISABLE</i>	Activer le blocage de l'introduction via l'interface					
<i>REPOSA</i>	Repositionierung linear all axes: Réaccostage du contour selon une droite avec tous les axes				s	2
<i>REPOSH</i>	Repositionierung semi circle: Réaccostage du contour selon un demi-cercle				s	2
<i>REPOSHA</i>	Repositionierung semi circle all axes: Réaccostage du contour avec tous les axes, les axes géométriques selon un demi-cercle				s	2
<i>REPOSL</i>	Repositionierung linear: Réaccostage du contour selon une droite				s	2
<i>REPOSQ</i>	Repositionierung quarter circle: Réaccostage du contour selon un quart de cercle				s	2
<i>REPOSQA</i>	Repositionierung quarter circle all axes: Réaccostage du contour avec tous les axes, des axes géométriques selon un quart de cercle				s	2
<i>RET</i>	Fin du sous-programme	0.001 ... 99 999.999	Au lieu de M2 pour conserver un contourage			
<i>RMB</i>	Réaccostage au point de début de bloc (Repos mode begin of block)				m	26
<i>RME</i>	Réaccostage au point de fin de bloc (Repos mode end of block)				m	26

15.1 Liste des instructions

Nom	Signification	Affectation de valeur	Description, commentaire	Syntaxe	modal/non modal	Groupe
RMI ¹	Réaccostage au point d'interruption (Repos mode interrupt)				m	26
RND	Arrondir les angles	Réel, sans signe		RND=...	s	
RNDM	Arrondi modal	Réel, sans signe		RNDM=... RNDM=0: désactiver M. V.	m	
ROT	Rotation programmable (ROTation)	Rotation autour du 1er axe géom. : -180° .. 180° 2e axe géo : -89.999°, ..., 90° 3e axe géo. : -180° .. 180°		ROT X... Y... Z... ROT RPL= ; bloc séparé	s	3
ROUND	Arrondir décimale inf./sup. (fonction mathématique)			ROUND(p); p = paramètre		
RP	Rayon polaire (Radius Polar)	Réel			m, s ³	
RPL	Rotation dans le plan (Rotation Plane)	Réel, sans signe			s	
S	Vitesse de rotation de broche ou (avec G4, G96) autre signification	0.1 ... 99999999.9	Vitesse de rotation en tr/min G4: Arrêt temporisé en nombre de tours de broche G96: Vitesse de coupe en m/min	S...: SUG pour broche maître S1...: SUG pour broche 1 (SUG = Vitesse périphérique de meuble)	m, s	
SAVE	Sauvegarder les données du programme principal lors de l'appel d'un sous-programme			par ex. PROC CONTOUR SAVE		
SCALE	Mise à l'échelle programmable (scale)			SCALE X... Y... Z... ; bloc séparé	s	3
SD	Degré spline (Spline Degree)	Nombre entier, sans signe			s	
SETINT	Activation de la routine d'interruption		p = 1: priorité majeure	SETINT(n) PRIO=p NAME; ou SETINT(n) PRIO=p LIFTFAST; ou SETINT(n) PRIO=p NAME LIFTFAST; n = 1, ..., 8 (entrée); p = 1, ..., 128 (priorité)		
SETM	Positionner mémentos dans canal propre		Sans effet sur l'exécution en cours Valide au-delà de reset et de Départ programme.	SETM(N° mémento, N° mémento,)		

Nom	Signification	Affectation de valeur	Description, commentaire	Syntaxe	modal/non modal	Groupe
			Egalement programmable à partir d'une action synchrone			
SETMS	Retour à la broche maître définie dans les paramètres machine					
SETMS(n)	La broche indiquée doit faire office de broche maître					
SETPIECE	Spécifier le nombre de pièces		y = 0: broche maître (standard)	SETPIECE(x,y); x = nombre de pièces usinées; y = numéro de broche		
SF	Décalage du point de départ pour filetage (spindle offset)	0.0000, ..., 359.999°			m	
SIN	Sinus (fonction trigonométrique)			SIN (p); p = paramètre		
SOFT	Accélération résultante progressive				m	21
SOFTA	Activation de l'accélération progressive pour les axes programmés					
SON ⁶	Activation du grignotage (Stroke ON)		Antagoniste de SON	SON G... X... Y... Z...	m	35
SONS ⁶	Activation du poinçonnage dans la période d'appel de l'interpolateur (Stroke ON Slow)		avec période d'appel de l'interpolation	SONS G... X... Y... Z...	m	35
SPCOF	Commuter broche maître ou broche (n) de la régulation de vitesse de rotation sur asservissement de position			SPCON SPCON (n)		
SPCON	Commuter broche maître ou broche (n) de l'asservissement de position sur la régulation de vitesse de rotation			SPCON SPCON (n)		
SPOS	Interpoler broche asservie			SPOS=...		
SPI	Convertir un numéro de broche en un descripteur d'axe					
SPIF1 ^{1,6}	E/S NCK rapides pour poinçonnage/grignotage Octet 1 (Stroke/Punch InterFace 1)				m	38
SPIF2 ⁶	E/S NCK rapides pour poinçonnage/grignotage Octet 2 (Stroke/Punch Interface 2)				m	38
SPLINE-PATH ⁷	Définir le groupe de type spline		max. 8 axes			
SPOF ^{1,6}	Désactiver coup, poinçonnage, grignotage (Stroke/Punch OFF)				m	35
SPN ⁶	Nombre de distances partielles dans un bloc (Stroke/Punch Number)	entier		SPN=	s	
SPP ⁶	Longueur d'une distance partielle (Stroke/Punch Path)	entier		SPP=	m	
SPOS	Position de broche			SPOS= ou SPOS(n)=	m	
SPOSA	Position de broche au-delà des limites de bloc			SPOSA= ou SPOSA(n)=	m	

15.1 Liste des instructions

Nom	Signification	Affectation de valeur	Description, commentaire	Syntaxe	modal/non modal	Groupe
SR	Trajet de retrait (Sparking out Retract path)	réel,sans signe			s	
SRA	Trajet de retrait axial avec entrée externe (Sparking out Retract Axial)				m	
ST	Durée des passes à lécher (Sparking out Time)	réel,sans signe			s	
STA	Durée axiale des passes à lécher (Sparking out Time Axial)				m	
DEPART	Départ programme CN		n,n: Enumération des numéros de canaux; valeur en fonction de la configuration de la commande	DEPART(n,n)		
STARTFIFO	Exécution; parallèlement remplissage de la mémoire tampon				m	4
STOPFIFO	Arrêt de l'exécution; remplissage du tampon d'exécution, jusqu'à reconnaissance de STARTFIFO, saturation de la mémoire ou fin du programme				m	4
STOPREOF	Annulation de l'arrêt du prétraitement des blocs		avec WHEN et avec effet non modal (sans programmer un N° ID)			
SUG	Vitesse périphérique de meule					
SUPA	Inhibition du décalage d'origine réglable		y compris des décalages programmés, des décalages DRF, du décalage d'origine externe et du décalage Preset.		s	9
SYNFCT	Lire variable temps réel, la traiter avec polynôme $\leq 3e$ degré et la sortir comme nouvelle variable temps réel			SYNFCT(n° polynôme, sortie variable temps réel, entrée variable temps réel)		
T	Appel d'outil (changer uniquement si défini dans PM, sinon instruction M6 nécessaire)	1 ... 32 000	Appel via N° T : ou descripteur d'outil :	par ex. T3 ou T=3 par ex. T="FORET"		
NOMTAB	Nom de la table de contour			NOMTAB1[n1]: Nom et n1. élément de contour de la 1. table; NOMTAB2[n2]: Nom et n2. élément de contour de la 2. table; NOMTAB3: Nom de la table des coordonnées du point d'in-		

Nom	Signification	Affectation de valeur	Description, commentaire	Syntaxe	modal/non modal	Groupe
				tersection dans le plan activé (G17, ..., G19)		
TAN	Tangente (fonction trigonométrique)			TAN (p); p = paramètre		
TANG	Instruction préparatoire pour la définition d'un asservissement tangentiel			TANG (AxeA,AxeP1,AxeP2, Coupl.,SC)		
TANGOF	Désactiver le positionnement tangentiel			TANGOF (axeA)		
TANGON	Activer le positionnement tangentiel			TANGON (AxeA,[Offset-] angle)		
TCARR	Appeler support d'outil (numéro "m")	entier	m=0: désactiver support d'outil actif	TCARR=[m]		
TCOABS	Déterminer les composantes de longueur d'outil à partir de l'orientation courante de l'outil		Nécessaire après chgt d'outil, par ex.			
TCOFR	Déterminer les composantes de longueur d'outil à partir de l'orientation du frame actif		Réglage manuel			
TILT ⁵	Angle latéral	réel			m	
TLIFT	Insertion d'un bloc intermédiaire aux angles du contour			TLIFT (AxeA)		
TMOF	Désactiver la surveillance d'outil		N° T nécessaire uniquement si l'outil avec ce N° n'est pas actif	TMOF [N° T]		
TMON	Activer la surveillance d'outil		N° T. = 0: désactiver surveillance pour tous les outils	TMON [N° T]		
TOFRAME	Application du frame courant programmable au système de coordonnées relatif à l'outil				s	3
TRAANG	Activer la transformation axe incliné convenue		TRAANG(a) = TRAANG(a,1) = 1. T.A.I. convenue Désactiver avec TRAFOOF	TRAANG(a,n); a = angle de l'axe incliné; n = 1,2		
TRACYL	Activer la transformation enveloppe cylindrique convenue		TRACYL(d) = TRACYL(d,1) = 1. F.T. convenue Désactiver avec TRAFOOF	TRACYL(d,n); d = diamètre cylindre; n = 1,2		
TRAFOOF	Désactiver transformation active					
TRAILOF	Désactiver groupe d'axes à déplacements conjugués			TRAILOF(AxeA,AxeP)		
TRAILON	Activer et définir groupe d'axes à déplacements conjugués			TRAILON(AxeA,AxeP, Coupl.)	m	

15.1 Liste des instructions

Nom	Signification	Affectation de valeur	Description, commentaire	Syntaxe	modal/non modal	Groupe
TRANS	Décalage programmable (translation)			TRANS X... Y... Z... ; bloc séparé	s	3
TRANSMIT	Activer la fonction TRANSMIT convenue		TRANSMIT = TRANSMIT(1) = 1. F.T. convenue Désactiver avec TRAFOOF	TRANSMIT(n); n = 1,2		
TRAORI	Activer la transformation d'orientation convenue		TRAORI = TRAORI(1) = 1. T.O. convenue Désactiver avec TRAFOOF	TRAORI(n); n = 1,2		
TRUNC	Trancher décimales (fonction mathématique)			TRUNC(p); p = paramètre		
TURN	Nombre de tours pour une hélice	0, ..., 999			s	
UNLOCK	Mot-clé : Libérer une action synchrone		n = N° Id de l'action synchrone	UNLOCK[n,n,...]		
VARIB	Variable d'état			par ex. Etat interface : TRUE : Pt d'intersection trouvé FALSE: pas de pt d'inter. trouvé		
WAITC	Attendre la condition de synchronisme			WAITC (BA,Chgtbloc,BA,Chgtbloc)		
WAITE	Attendre la fin du programme des canaux indiqués		Ne pas déclarer le propre canal.	WAITE (n,n)		
WAITM	- définir memento dans propre canal, - terminer bloc précédent avec arrêt précis, - attendre le memento, - effacer le memento après synchronisation.		Il n'est pas nécessaire de déclarer le propre canal. Max. 10 mementos par canal	WAITM(N° memento,n,n,...)		
WAITMC	- définir memento dans propre canal, - amorcer l'arrêt précis dans le bloc précédent, - attendre le memento, - quand le memento est atteint : poursuivre l'usinage sans terminer l'arrêt précis, - effacer le memento après synchronisation.		Il n'est pas nécessaire de déclarer le propre canal. Si le memento est déjà défini dans d'autres canaux : L'arrêt précis n'est pas amorcé et l'usinage est poursuivi sans freinage. Cette fonctionnalité est active uniquement en mode d'avance	WAITMC (N° memento, n, n,)		

Nom	Signification	Affectation de valeur	Description, commentaire	Syntaxe	modal/non modal	Groupe
			tangentielle (G64) et quand la fonction Look-Ahead est activée.			
WAITS	Attendre que soit atteinte la position de broche			WAITS (broche principale) WAITS (n,n,n)		
WALIMOF	Désactivation limitation zone de travail (working area limitation OFF)			; bloc séparé	m	28
WALIMON ¹	Activation limitation zone de travail (working area limitation ON)			; bloc séparé	m	28
WHEN	Mot-clé : L'action est exécutée une fois, quand la condition est remplie.					
WHENEVER	Mot-clé : L'exécution de l'action est réitérée aussi longtemps que la condition est remplie					
TO	Descripteur d'outil					
X	Axe	réel			m,s ³	
Y	Axe	réel			m,s ³	
Z	Axe	réel			m,s ³	

Légende :

¹ Réglage par défaut en début de programme (préréglage de la commande à la livraison, si aucune autre programmation n'a été demandée).

² La numérotation des groupes correspond à la table "Liste des instructions" dans le chapitre 11.3

³ Points finaux absolus: à effet modal; points finaux relatifs : à effet non modal; sinon modal/non modal en fonction de la syntaxe de la fonction G.

⁴ Les paramètres IPO agissent de manière relative en tant que centres d'arcs de cercle. Ils peuvent être programmés de manière absolue avec AC. Avec une autre signification (par ex. pas de filet) la modification d'adresse est ignorée.

⁵ Mot-clé non valide pour SINUMERIK FM-NC/810D

⁶ Mot-clé non valide pour SINUMERIK FM-NC/810D/NCU571

⁷ Mot-clé non valide pour SINUMERIK 810D

⁸ L'utilisateur OEM a la possibilité d'introduire deux modes d'interpolation supplémentaires. Il peut également modifier les dénominations.

⁹ Mot-clé valide uniquement pour SINUMERIK FM-NC

¹⁰ L'écriture de l'adresse avec extension n'est pas admise pour ces fonctions.

Remarque : Dans ce chapitre, "s" est à prendre dans le sens de "single block" (fonction non modale).

15.2 Liste des adresses

Lettres adresses

Lettre	Signification	Extension numérique
A	descripteur réglable	x
B	descripteur réglable	x
C	descripteur réglable	x
D	activation/désactivation de la correction d'outil, de l'arête tranchante de l'outil	
E	descripteur réglable	
F	avance arrêt temporisé en secondes	x
G	fonction G	
H	fonction H	x
I	descripteur réglable	x
J	descripteur réglable	x
K	descripteur réglable	x
L	sous-programme, appel de sous-programme	
M	fonction M	x
N	numéro de bloc secondaire	
O	inutilisé	
P	nombre d'exécutions du programme	
Q	descripteur réglable	x
R	descripteur de variable (paramètre R)	x
S	valeur broche arrêt temporisé en tours de broche	x x
T	numéro d'outil	x
U	descripteur réglable	x
V	descripteur réglable	x
W	descripteur réglable	x
X	descripteur réglable	x
Y	descripteur réglable	x
Z	descripteur réglable	x
%	caractère de début et de séparation lors du transfert de fichiers	
:	numéro de bloc principal	
/	bloc optionnel	

Adresses fixes

Descripteur d'adresse	Type d'adresse	à effet modal / non modal	Inch/ métrique	G90/G91	IC	AC	DC, ACN, ACP	CIC, CAC, CDC, CACN, CACP	Qu	Type de donnée
L	numéro de sous-programme	s								entier, sans signe
P	nombre d'exécutions de sous-programme	s								entier, sans signe
N	numéro de bloc	s								entier, sans signe
G	fonction G	v. liste des fonctions G								entier, sans signe
F	avance, arrêt temporisé	m, s							x	réel, sans signe
OVR	correction	m								réel, sans signe
S	broche, arrêt temporisé	m, s							x	réel, sans signe
SPOS	position de broche	m			x	x	x			réel
SPOSA	position de broche au-delà des limites de bloc	m			x	x	x			réel
T	numéro d'outil	m							x	entier, sans signe
D	numéro de correcteur	m							x	entier, sans signe
M, H,	fonctions auxiliaires	s							x	M: entier, sans signe H: réel

Adresses fixes avec extension d'axe

Descripteur d'adresse	Type d'adresse	à effet modal / non modal	Inch/ métrique	G90/G91	IC	AC	DC, ACN, ACP	CIC, CAC, CDC, CACN, CACP	Qu	Type de donnée
AX: Axis	descripteur d'axe variable	*)	x	x	x	x	x			réel
IP: Interpolation paramètre	paramètre d'interpolation variable	s	x	x	x	x				réel
POS: Positioning axis	axe de positionnement	m	x	x	x	x	x	x		réel
POSA: Positioning axis above end of block	axe de positionnement au-delà des limites de bloc	m	x	x	x	x	x	x		réel
POSP: Positioning axis in parts	positionnement par étapes (oscillation)	m	x	x	x	x	x			réel : position finale réel : longueur pièce entier: option
PO: polynôme:	coefficient polynomial	s	x							réel, sans signe 1 à 8 fois
FA: Feed axial	avance axiale	m							x	réel, sans signe
FL: Feed limit	limite avance axiale	m								réel, sans signe
OVR: correction	correction avance axiale	m								réel, sans signe

*) points finaux absolus : modale, points finaux relatifs : non modale, sinon modale/non modale en fonction de la syntaxe de la fonction G.

Mot-clé **non** valide pour la SINUMERIK FM-NC.

Adresses fixes avec extension d'axe

Descripteur d'adresse	Type d'adresse	à effet modal / non modal	Inch/métrique	G90/G91	IC	AC	DC, ACN, ACP	CIC, CAC, CDC, CACN, CACP	Qu	Type de donnée
ACC #: Accélération axiale	accélération axiale	m								réel, sans signe
FMA: Feed multiple axial	avance synchrone axiale	m								réel, sans signe
STA: Sparking out time axial	durée axiale des passes à lécher	m								réel, sans signe
SRA: Sparking out retract	course axiale de retrait suite à action externe	m								réel, sans signe
OS : Oscillating on/off	activ./désactiv. oscillation	m								entier, sans signe
OST1: Oscillating time 1	temps d'arrêt au point d'inversion gauche (oscillation)	m								réel
OST2: Oscillating time 2	temps d'arrêt au point d'inversion droit (oscillation)	m								réel
OSP1: Oscillating Position 1	point d'inversion gauche (oscillation)	m	x	x	x	x	x			réel
OSP2: Oscillating Position 2	point d'inversion droit (oscillation)	m	x	x	x	x	x			réel
OSE: Oscillating end position	point d'arrêt de l'oscillation	m	x	x	x	x	x			réel
OSNSC: Oscillating: number spark out cycles	nombre de passes à lécher (oscillations)	m								entier, sans signe
OSCTRL: Oscillating control	options d'oscillation	m								entier, sans signe: Options de réglage des entiers sans signe: Options d'annulation
OSCILL: Oscillation	affectation des axes d'oscillation, activation de l'oscillation	m								axes : 1 à 3 axes de pénétration
FDA: Feed DRF axial	avance axiale pour correction par manivelle	s								réel, sans signe

15.2 Liste des adresses

Descripteur d'adresse	Type d'adresse	à effet modal / non modal	Inch/ métrique	G90/G91	IC	AC	DC, ACN, ACP	CIC, CAC, CDC, CACN, CACP	Qu	Type de donnée
FXS: Fixed stop	activation "accostage de butée"	m								entier, sans signe
FXST: Fixed stop torque	limite de couple pour "accostage de butée"	m								réel
FXSW: Fixed stop window	fenêtre de surveillance pour "accostage de butée"	m								réel

Pour ces types d'adresses, on mentionne, entre crochets droits, un axe ou une expression de type axe. Le type de donnée spécifié dans la colonne correspondante est le type de la valeur attribuée.

Le mot-clé n'est pas valide pour la SINUMERIK FM-NC/810 D.

Adresses réglables

Réglage standard	Type d'adresse	modal/ non modal	Inch/ métrique	G90/G91 1	IC	AC	DC, ACN, ACP	CIC, CAC, CDC, CACN, CACP	Qu	Nombre maxi.	Type de donnée
Valeurs d'axes et points finaux											
X, Y, Z, A, B, C	axe	*)	x	x	x	x	x			8	réel
AP: Angle polar	angle polaire	m/s*		x	x	x				1	réel
RP: Radius polar	rayon polaire	m/s*	x	x	x	x				1	réel, sans signe
Orientation d'outil											
A2, B2, C2#	angle d'Euler	s								3	réel
A3, B3, C3#	composantes des vecteurs d'orientation	s								3	réel
A4, B4, C4 pour début de bloc #		m								3	réel
A5, B5, C5 pour fin de bloc #	composante de vecteur normal	s								3	réel
LEAD: Lead Angle #	angle d'avance	m								1	réel
TILT: Tilt Angle #	angle latéral	m								1	réel
ORIS:# Orientation Smoothing Factor	modification de l'orientation (rapportée à la trajectoire)	m								1	réel

Réglage standard	Type d'adresse	modal/ non modal	Inch/ métrique	G90/G9 1	IC	AC	DC, ACN, ACP	CIC, CAC, CDC, CACN, CACP	Qu	Nombre maxi.	Type de donnée
Paramètres d'interpolation											
I, J, K**	Paramètres d'interpolation	s	x		x**	x**				3	réel
I1, J1, K1	coordonnées de points inter- médiaires	s	x	x	x	x				3	réel
RPL: Rotation plane	rotation dans un plan	s								1	réel
CR: Circle – Radius	rayon de cercle	s	x							1	réel, sans signe
AR: Angle circular	angle au centre									1	réel, sans signe
TURN	Nombre de tours pour une hélice	s								1	entier, sans signe
PL: paramètre - Interval - Length	paramètre - intervalle - lon- gueur	s								1	réel, sans signe
PW: Point – Weight	poids de point	s								1	réel, sans signe
SD: Spline – Degree	degré de spline	s								1	entier, sans signe
SF: Spindle offset	décalage du point d'attaque pour filetage	m								1	réel
DISR: Distance for repositioning	distance de re- positionnement	s	x							1	réel, sans signe
DISPR: Distance path for reposi- tioning	différence sur le contour pour re- positionnement	s	x							1	réel, sans signe
ALF: Angle lift fast	angle de retrait rapide	m								1	entier, sans signe
FP	point fixe : numéro du point fixe à accoster	s								1	entier, sans signe
RNDM: Round modal	arrondissement modal	m	x							1	réel, sans signe
RND: Round	arrondissement non modal	s	x							1	réel, sans signe
CHF: Chamfer	chanfrein non modal	s	x							1	réel, sans signe
ISD: Insertion depth	profondeur de pénétration	m								1	réel

Réglage standard	Type d'adresse	modal/ non modal	Inch/ métrique	G90/G91	IC	AC	DC, ACN, ACP	CIC, CAC, CDC, CACN, CACP	Qu	Nombre maxi.	Type de donnée
DISC: Distance	surcourse pour cercle de ra- ccordement – correction de rayon d'outil	m	x							1	réel, sans signe
OFFN	décalage de contour - normal	m								1	réel

*) points finaux absolus : modale, points finaux relatifs : pendant un bloc en fonction de la synt axe de la fonction G.

**Les paramètres IPO agissent de manière relative en tant que centres d'arcs de cercle. Ils peuvent être programmés de manière absolue avec AC.

Lorsque leur signification est différente (ex : pas d'un filetage), la modification d'adresse est ignorée.

Mot-clé **non** valide pour la SINUMERIK FM-NC/810 D/NCU571.

Adresses réglables

Grignotage/ poinçonnage	Type d'adresse	Modale/ non modale	Inch/ métri- que	G90/G9 1	IC	AC	DC, ACN, ACP	CIC, CAC, CDC, CACN, CACP	Qu	Nombre maxi.	Type de donnée
SPN: Stroke/ Punch Number #	nombre de dis- tances partielles par bloc	s								1	INT
SPP: Stroke/ Punch Path #	longueur d'une distance partielle	m								1	réel

Rectification/ actions syn- chrones au déplacement	Type d'adresse	Modale/ non modale	Inch/ métri- que	G90/G9 1	IC	AC	DC, ACN, ACP	CIC, CAC, CDC, CACN, CACP	Qu	Nombre maxi.	Type de donnée
ST: Sparking out time	durée des passes à lécher	s								1	réel, sans signe
SR: Sparking out retract path	course de retrait	s								1	réel, sans signe

Critères de transition entre blocs											
ADIS	distance de transition	m	x							1	réel, sans signe
ADISPOS	distance de transition en rapide	m	x							1	réel, sans signe

Mesure											
MEAS: Measure	mesure avec palpeur à déclenchement	s								1	entier, sans signe
MEAW: Measure without deleting distance to go	mesure avec palpeur à déclenchement sans effacement de la distance restant à par- courir	s								1	entier, sans signe

Contrôle d'axe/de broche											
LIMS: Limit spindle speed	limitation de la vitesse de rota- tion de la broche	m								1	réel, sans signe

15.2 Liste des adresses

Avances											
FD: Feed DRF	avance tangentielle pour correction par manivelle	s								1	réel, sans signe

Adresses OEM											
OMA1: OEM – Adress 1 #	OEM – Adresse n° 1	m			x	x	x			1	réel
OMA2: OEM – Adress 2 #	OEM – Adresse n° 2	m			x	x	x			1	réel
OMA3: OEM – Adress 3 #	OEM – Adresse n° 3	m			x	x	x			1	réel
OMA4: OEM – Adress 4 #	OEM – Adresse n° 4	m			x	x	x			1	réel
OMA5: OEM – Adress 5 #	OEM – Adresse n° 5	m			x	x	x			1	réel

Mot-clé non valide pour la SINUMERIK FM-NC/810D/NCU571.

15.3 Liste des fonctions G/Fonctions préparatoires

Légende pour la description des groupes G

N° : numéro interne utilisé par ex. pour une interface AP, par ex. m : fonction modale
 s : fonction non modale ("single block")
 Std.: standard

Groupe 1 : Instructions de déplacement modales				
Nom	N°	Signification	m/s	std.
G0		déplacement en rapide	m	
G1		interpolation linéaire	m	std.
G2		interpolation circulaire dans le sens horaire	m	
G3		interpolation circulaire dans le sens antihoraire	m	
CIP		circle through points: Interpolation circulaire avec point intermédiaire	m	
ASPLINE		spline Akima	m	
BSPLINE		spline B	m	
CSPLINE		spline cubique	m	
POLY ##		polynôme: interpolation polynomiale	m	
G33		filetage à l'outil, à pas constant	m	
G331		taroudage sans porte-taroud compensateur	m	
G332		retrait (après G331)	m	
OEMIPO1 ####		interpolation OEM 1 *)	m	
OEMIPO2 ####		interpolation OEM 2 *)	m	
CT		cercle avec transition tangentielle	m	

*) L'utilisateur OEM a la possibilité d'introduire deux modes d'interpolation supplémentaires. Il peut également modifier les dénominations.

Dans les fonctions G modales, quand aucune fonction du groupe n'est programmée, c'est le réglage par défaut modifiable par le biais d'un paramètre machine, qui est actif : \$MC_GCODE_RESET_VALUES, modifiable à l'aide d'un paramètre machine, s'applique par défaut lorsqu'aucune fonction du groupe n'est programmée.

Mot-clé non valide pour la SINUMERIK 810D.

Mot-clé non valide pour la SINUMERIK FM-NC/810D.

Mot-clé non valide pour la SINUMERIK FM-NC/810D/NCU571

Groupe 2 : instructions de déplacement non modales, arrêt temporisé				
Nom	N°	Signification :	m/s	std.
G4	1.	arrêt temporisé	s	
G63	2.	taroudage sans synchronisation	s	
G74	3.	accostage du point de référence avec synchronisation	s	
G75	4.	accostage de point fixe	s	
REPOS	5.	Repositioning linear: réaccostage du contour selon une droite	s	

REPOSQ	6.	Repositioning quarter circle: réaccostage du contour selon un quart de cercle	s	
Nom	N°	Signification :	m/s	std.
REPOSH	1.	Repositioning semi circle: réaccostage du contour selon un demi-cercle	s	
REPOSA	2.	Repositioning linear all axis: réaccostage du contour selon une droite avec tous les axes	s	
REPOSQA	3.	Repositioning Quarter Circle All Axis: réaccostage du contour avec tous les axes, des axes géométriques selon un quart de cercle	s	
REPOSHA	4.	Repositioning Semi Circle All Axis: réaccostage du contour avec tous les axes, les axes géométrique selon un demi-cercle	s	

Liste des fonctions G/fonctions préparatoires

Groupe 3 : écrire en mémoire				
Nom	N°	Signification	m/s	std.
TRANS	1.	TRANSLATION : décalage programmable	s	
ROT	2.	ROTATION : rotation programmable	s	
SCALE	3.	SCALE : facteur d'échelle programmable	s	
MIRROR	4.	MIRROR : fonction miroir programmable	s	
ATRANS	5.	Additive TRANSLATION : décalage additif programmable	s	
AROT	6.	Additive ROTATION : rotation programmable	s	
ASCALE	7.	Additive SCALE : facteur d'échelle programmable	s	
AMIRROR	8.	Additive MIRROR : fonction miroir programmable	s	
TOFRAME	9.	Application du frame courant programmable au système de coordonnées relatif à l'outil	s	
G25	10.	Limite minimale de la zone de travail/vitesse de rotation de broche minimale	s	
G26	11.	Limite maximale de la zone de travail/vitesse de rotation de broche maximale	s	
G110	12.	Programmation du pôle par rapport à la dernière position de consigne programmée	s	
G111	13.	Programmation du pôle par rapport à l'origine pièce courante	s	
G112	14.	Programmation du pôle par rapport au dernier pôle valable	s	

Groupe 4 : FIFO				
Nom	N°	Signification :	m/s	std.
STARTFIFO	1.	Start FIFO : arrêt de l'exécution du programme et remplissage du tampon d'exécution jusqu'à décodage de STOPFIFO, tampon d'exécution plein ou fin de programme	m	std.
STOPFIFO	2.	STOP FIFO : arrêt de l'usinage; remplissage du tampon d'exécution jusqu'à reconnaissance de STARTFIFO, saturation du tampon d'exécution ou fin de programme	m	

Groupe 6 : choix du plan				
Nom	N°	Signification :	m/s	std.
G17	1.	Sélection du plan 1er - 2me axe géométrique	m	std.
G18	2.	Sélection du plan 3me - 1er axe géométrique	m	
G19	3.	Sélection du plan 2me - 3me axe géométrique	m	

Groupe 7 : Correction du rayon d'outil				
Nom	N°	Signification :	m/s	std.
G40	1.	Pas de correction du rayon de l'outil	m	std.
G41	2.	Correction du rayon de l'outil à gauche du contour	m	
G42	3.	Correction du rayon de l'outil à droite du contour	m	

Liste des fonctions G/fonctions préparatoires

Groupe 8 : décalage d'origine réglable				
Nom	N°	Signification	m/s	std.
G500	1.	Désactivation G54 - G5XX, remise à zéro du frame réglable	m	std.
G54	2.	1er décalage d'origine réglable	m	
G55	3.	2e décalage d'origine réglable	m	
G56	4.	3e décalage d'origine réglable	m	
G57	5.	4e décalage d'origine réglable	m	
G5xx	n	n. décalage d'origine réglable	m	
G599	100.	100e décalage d'origine réglable	m	

Les fonctions G de ce groupe activent chacune un frame utilisateur réglable : \$P_UIFR[].

A G54 correspond le frame \$P_UIFR[1], à G505 correspond le frame \$P_UIFR[5].

Le nombre de frames utilisateur réglables et, de ce fait, le nombre de fonctions G de ce groupe est paramétrable à l'aide du paramètre machine \$MC_MM_NUM_USER_FRAMES.

Groupe 9 : inhibition des frames				
Nom	N°	Signification :	m/s	std.
G53	1	Inhibition des frames courants	s	
SUPA	2.	Suppress all : supprime frames, décalage Preset, décalage DRF	s	

Groupe 10 : arrêt précis - contournage				
Nom	N°	Signification :	m/s	std.
G60	1.	Ralentissement, arrêt précis	m	std.
G64	2.	Contournage	m	
G641	3.	Contournage avec distance de transition programmable entre blocs	m	

Groupe 11 : arrêt précis non modal				
Nom	N°	Signification :	m/s	std.
G9	1.	Arrêt précis	s	

Groupe 12 : critères de changement de bloc pour "arrêt précis" (G60/G09)				
Nom	N°	Signification :	m/s	std.
G601	1.	Changement de bloc lors d'un arrêt précis fin	m	std.
G602	2.	Changement de bloc lors d'un arrêt précis grossier	m	
G603	3.	Changement de bloc à la fin de l'interpolation	m	

Groupe 13 : système d'introduction				
Nom	N°	Signification :	m/s	std.
G70	1.	Système d'introduction anglo-saxon	m	
G71	2.	Système d'introduction métrique	m	std.

Liste des fonctions G/fonctions préparatoires

Groupe 14 : type de programmation de cotes				
Nom	N°	Signification	m/s	std.
G90	1.	Programmation de cotes absolues	m	std.
G91	2.	Programmation de cotes relatives	m	

Groupe 15 : type d'avance				
Nom	N°	Signification :	m/s	std.
G93 #	1.	Avance en inverse du temps	m	
G94	2.	Avance linéaire en mm/min, inch/min	m	std.
G95	3.	Avance par tour en mm/tr, inch/tr	m	
G96	4.	Activation de la vitesse de coupe constante		
G97	5.	Désactivation de la vitesse de coupe constante		

Mot-clé non valide pour la SINUMERIK FM-NC/810D.

Groupe 16 : correction de l'avance au niveau des courbures concaves et convexes				
Nom	N°	Signification :	m/s	std.
CFC	1.	Constant feed at contour Avance constante au niveau du contour	m	std.
CFTCP	2.	Constant feed in tool-center-point Avance constante au point de référence du tranchant (trajectoire du centre de l'outil)	m	
CFIN	3.	Constant feed at internal radius, acceleration at external radius Avance constante au niveau des courbures concaves, accélération au niveau des courbures convexes	m	

Groupe 17 : comportement à l'accostage/au retrait du contour avec correction d'outil				
Nom	N°	Signification :	m/s	std.
NORM	1.	Positionnement à la normale du point de départ/de fin	m	std.
KONT	2.	Contournement du point de départ/d'arrivée	m	

Groupe 18 : comportement aux angles saillants avec correction d'outil				
Nom	N°	Signification :	m/s	std.
G450	1.	Arc de raccordement	m	std.
G451	2.	Point d'intersection des équidistantes	m	

Groupe 19 : raccordement en début de fonction spline				
Nom	N°	Signification :	m/s	std.
BNAT #	1.	Begin natural: raccordement naturel avec le premier bloc spline	m	std.
BTAN #	2.	Begin tangential : raccordement tangential avec le premier bloc spline	m	
BAUTO #	3.	Begin not a knot : (pas de jonction) détermination du premier segment spline à l'aide des trois points suivants	m	

Mot-clé non valide pour la SINUMERIK 810D.

Liste des fonctions G/Fonctions préparatoires

Groupe 20 : raccordement en fin de fonction spline				
Nom	N°	Signification	m/s	std.
ENAT #	1.	End natural : raccordement naturel avec le bloc de déplacement suivant	m	std.
ETAN #	2.	End tangential : raccordement tangential avec le bloc de déplacement suivant au début de la spline	m	
EAUTO #	3.	End not a knot : (pas de jonction) détermination du dernier segment spline à l'aide des trois derniers points	m	

Mot-clé non valide pour la SINUMERIK 810D.

Groupe 21 : accélération				
Nom	N°	Signification :	m/s	std.
BRISK	1.	Accélération résultante sous forme d'échelon	m	std.
SOFT	2.	Accélération résultante progressive	m	
DRIVE ##	3.	Accélération résultante fonction de la vitesse	m	

Mot-clé non valide pour la SINUMERIK FM-NC

Groupe 22 : type de correction d'outil				
Nom	N°	Signification :	m/s	std.
CUT2D	1.	Cutter – compensation – type 2dimensional : correction d'outil 2D1/2	m	std.
CUT2DF	2.	Cutter-compensation-type 2dimensional : correction d'outil 2D1/2 La correction d'outil agit de façon relative par rapport au frame courant (plan incliné)	m	
CUT3DC #	3.	Cutter – compensation – type 3dimensional circonférence : correction d'outil 3D - fraisage périphérique	m	
CUT3DF #	4.	Cutter – compensation – type 3dimensional face: correction d'outil 3D - fraisage en bout	m	
CUT3DFS #	5.	Cutter – compensation – type 3dimensional face: Correction d'outil 3D pour fraisage en bout avec positionnement constant de l'outil indépendant du frame actif	m	
CUT3DFF #	6.	Cutter – compensation – type 3dimensional face frame: Correction d'outil 3D pour fraisage en bout avec positionnement constant de l'outil dépendant du frame actif	m	

Mot-clé non valide pour la SINUMERIK FM-NC/810D.

Groupe 23 : détection des violations de contours internes				
Nom	N°	Signification :	m/s	std.
CDOF	1.	Collision detection of : désactivation de la détection des violations	m	std.
CDON	2.	Collision detection on : activation de la détection des violations	m	

Groupe 24 : commande anticipatrice				
Nom	N°	Signification :	m/s	std.
FFWOF	1.	Feed forward off : désactivation de la commande anticipatrice	m	std.
FFWON	2.	Feed forward off : activation de la commande anticipatrice	m	

Groupe 25 : référence pour l'orientation de l'outil				
Nom	N°	Signification :	m/s	std.
ORIWKS #	1.	Tool – orientation in workpiece coordinate system : Positionnement de l'outil dans le système de coordonnées pièce	m	std.
ORIMKS #	2.	Tool – orientation in machine coordinate system: Positionnement de l'outil dans le système de coordonnées machine	m	
ORIPATH #	3.	Tool – orientation path	m	

Mot-clé non valide pour la SINUMERIK FM-NC/810D/NCU571.

Groupe 26 : type de repositionnement				
Nom	N°	Signification :	m/s	std.
RMB	1.	Repos - mode begin of block : réaccostage du point de début de bloc	m	
RMI	2.	Repos - mode interrupt : réaccostage au point d'interruption	m	std.
RME	3.	Repos - mode end of block : réaccostage du point final de bloc	m	

Liste des fonctions G/fonctions préparatoires

Groupe 27 : correction d'outil en cas de changement d'orientation au niveau des angles saillants				
Nom	N°	Signification	m/s	std.
ORIC #	1.	Orientation change continuously : Les changements d'orientation aux angles saillants sont superposés au bloc à interpolation circulaire à insérer.	m	std.
ORID #	2.	Orientation change discontinuously : Les changements d'orientation sont exécutés avant le bloc à interpolation circulaire	m	

Mot-clé non valide pour la SINUMERIK FM-NC/810D/NCU571.

Groupe 28 : limitation de la zone de travail				
Nom	N°	Signification :	m/s	std.
WALIMON	1.	Working area limitation on : activation de la limitation de la zone de travail	m	std.
WALIMOF	2.	Working area limitation off : désactivation de la limitation de la zone de travail	m	

Groupe 29 : rayon - diamètre				
Nom	N°	Signification :	m/s	std.
DIAMOF	1.	Désactivation de la programmation du diamètre	m	std.
DIAMON	2.	Activation de la programmation du diamètre	m	

Groupe 30 : compactage				
Nom	N°	Signification :	m/s	std.
COMPOF #	1.	Désactivation du compactage	m	std.
COMPON #	2.	Activation du compactage	m	

Groupe 31 : fonctions G OEM				
Nom	N°	Signification :	m/s	std.
G810 #	1.	Fonction G OEM		std.
G811 #	2.	Fonction G OEM		
G812 #	3.	Fonction G OEM		
G813 #	4.	Fonction G OEM		
G814 #	5.	Fonction G OEM		
G815 #	6.	Fonction G OEM		
G816 #	7.	Fonction G OEM		
G817 #	8.	Fonction G OEM		
G818 #	9.	Fonction G OEM		
G819 #	10.	Fonction G OEM		

Deux groupes de fonctions G sont réservés à l'utilisateur OEM.

Ce faisant, il autorise la programmation des fonctions spécifiques élaborées par ses soins par des services de programmation extérieurs.

#Mot-clé non valide pour la SINUMERIK FM-NC/810D/NCU571.

Liste des fonctions G/Fonctions préparatoires

Groupe 32 : fonctions G OEM				
Nom	N°	Signification	m/s	std.
G820 #	1.	Fonction G OEM		std.
G821 #	2.	Fonction G OEM		
G822 #	3.	Fonction G OEM		
G823 #	4.	Fonction G OEM		
G824 #	5.	Fonction G OEM		
G825 #	6.	Fonction G OEM		
G826 #	7.	Fonction G OEM		
G827 #	8.	Fonction G OEM		
G828 #	9.	Fonction G OEM		
G829 #	10.	Fonction G OEM		

Deux groupes de fonctions G sont réservés à l'utilisateur OEM.

Ce faisant, il autorise la programmation des fonctions spécifiques élaborées par ses soins par des services de programmation extérieurs.

Groupe 33 : correction d'outil fine réglable				
Nom	N°	Signification :	m/s	std.
FTOCOF #	1.	Fine - Tool - Offset - Compensation off : désactivation de la correction d'outil fine en ligne	m	std.
FTOCON #	2.	Fine - Tool - Offset - Compensation on : activation de la correction d'outil fine en ligne	m	

Groupe 34 : lissage de l'orientation de l'outil				
Nom	N°	Signification :	m/s	std.
OSOF #	1.	Désactivation du lissage de l'orientation de l'outil	m	std.
OSC #	2.	Lissage constant de l'orientation de l'outil	m	
OSS #	3.	Lissage de l'orientation de l'outil en fin de bloc	m	
OSSE #	4.	Lissage de l'orientation de l'outil en début et fin de bloc	m	

Groupe 35 : poinçonnage et grignotage				
Nom	N°	Signification :	m/s	std.
SPOF #	1.	Stroke/Punch Off : désactivation de la course, du poinçonnage, du grignotage	m	std.
SON #	2.	Stroke On : activation du grignotage	m	
PON #	3.	Punch On : activation du poinçonnage	m	
SONS #	4.	Stroke On Slow : activation du grignotage à la période d'appel de l'interpolateur	m	
PONS #	5.	Punch On Slow : activation du poinçonnage à la période d'appel de l'interpolateur	m	

Groupe 36 : poinçonnage avec temporisation				
Nom	N°	Signification :	m/s	std.
PDELAYON #	1.	Punch with Delay On : activation du poinçonnage avec temporisation	m	std.
PDELAYOF #	2.	Punch with Delay Off : désactivation du poinçonnage avec temporisation	m	

Mot-clé non valide pour la SINUMERIK FM-NC/810D/NCU571.

Liste des fonctions G/Fonctions préparatoires

Groupe 37 : variation de l'avance				
Nom	N°	Signification :	m/s	std.
FNORM #	1.	Feed Normal : avance normale selon DIN66025	m	std.
FLIN #	2.	Feed Linear : avance variable linéairement	m	
FCUB #	3.	Feed Cubic : avance variable selon un spline cubique		

Groupe 38 : affectation des entrées/sorties NCK pour le poinçonnage/grignotage				
Nom	N°	Signification :	m/s	std.
SPIF1 #	1.	Stroke/Punch Interface 1: E/S NCK rapides pour poinçonnage/grignotage Octet 1	m	std.
SPIF2 #	2.	Stroke/Punch Interface 2: E/S NCK rapides pour poinçonnage/grignotage Octet 2	m	

Groupe 39 : précision de contour programmable				
Nom	N°	Signification :	m/s	std.
CPRECOF	1.	Contour Precision Off : désactivation de la précision de contour programmable	m	std.
CPRECON	2.	Contour Precision On : activation de la précision de contour programmable	m	

Mot-clé non valide pour la SINUMERIK FM-NC/810D/NCU571.

15.4 Liste des sous-programmes prédéfinis

Système de coordonnées

On active certaines fonctions de commande en utilisant la syntaxe des appels de sous-programmes.

1. Système de coordonnées					
Mot-clé/ Descripteur de fonction	Paramètre 1	Paramètre 2	Paramètre 3-15	Paramètre 4-16	Signification
PRESETON	AXIS*: descripteur d'axe machine	REAL: décalage preset	paramètres 3-15 idem 1 ...	paramètres 4-16 idem 2 ...	Préréglage des mémoires de valeurs réelles pour les axes programmés. On programme un descripteur d'axe, puis la valeur correspondante dans le paramètre suivant. La fonction PRESETON permet de programmer des décalages preset pour 8 axes au maximum.
DRFOF					Effacement du décalage DRF pour tous les axes affectés au canal

*) A la place des descripteurs d'axes machine, on peut aussi avoir, d'une manière générale, des descripteurs d'axes géométriques ou d'axes supplémentaires, dans la mesure où il n'y a pas d'ambiguïté.

Appels de sous-programmes prédéfinis

2. Groupes d'axes		
Mot-clé/ Descripteur de fonction	Paramètres 1-8	Signification
FGROUP	Descripteur d'axe du canal	Référence à une valeur F variable : détermination des axes auxquels se rapporte l'avance tangentielle. Nombre maximal d'axes: 8 FGROUP() sans indication de paramètres active le réglage standard comme référence de valeur F.
CLGON #	REAL: vitesse de rotation max.. meule d'entraînement	Centerless grinding on : activation rectification "sans centres"
CLGOF #		Centerless grinding off : désactivation rectification "sans centres"

Mot-clé/ Descripteur de fonction	Paramètres 1-8	Paramètres 2.-9	Signification
SPLINEPATH ##	INT : groupe Spline (doit être égal à 1)	AXIS : descripteurs géométriques ou supplémentaires	Détermination du groupe d'axes pour la fonction spline. Nombre maximal d'axes : 8
BRISKA	AXIS		Activation de l'accélération sous forme d'échelon pour les axes programmés
SOFTA	AXIS		Activation de l'accélération progressive pour les axes programmés
DRIVEA ###	AXIS		Activation de la courbe d'accélération infléchie pour les axes programmés
JERKA	AXIS		Le comportement à l'accélération, réglé à l'aide du paramètre machine \$MA_AX_JERK_ENABLE, est valable pour les axes programmés.

Mot-clé non valide pour la SINUMERIK FM-NC/810D/NCU571.

Mot-clé non valide pour la SINUMERIK 810D.

Mot-clé valide uniquement pour la SINUMERIK FM-NC.

Appels de sous-programmes prédéfinis

2. Groupes d'axes						
Mot-clé/ Descripteur de sous-programme	Paramètre 1	Paramètre 2	Paramètre 3	Paramètre 4	Paramètre 5	Signification
TANG #	AXIS : Nom d'axe axe asservi	REAL: axe pilote 1	AXIS : axe pilote 2	REAL: facteur de couplage	CHAR: Option: "B" : asservissement dans le système de coordonnées de base "W" : asservissement dans le système de coordonnées pièce.	Instruction préliminaire pour la définition d'un asservissement tangential : On détermine la tangente pour l'asservissement à partir des deux axes pilotes indiqués. Le facteur de couplage indique le rapport entre les varia- tions angulaires de la tangente et de l'axe asservi. En général, il vaut 1.

Mot-clé/ Descripteur de sous-programme	Paramètre 1	Paramètre 2	Paramètre 3	Paramètre 4	Paramètre 5	Signification
TANGON #	AXIS : Nom d'axe axe asservi	REAL: décalage angle				Tangential follow up mode on : activation asservissement tangential
TANGOF #	AXIS : Nom d'axe axe asservi					Tangential follow up mode off : désactivation asservissement tangential
TLIFT #	AXIS : axe asservi		REAL: distance			Tangential lift : activation de l'asservissement tangential
TRAILON ##	AXIS : axe conjugué	AXIS : axe pilote	REAL: facteur de couplage			Trailing on : activation déplacements conjugués à synchronisme axial
TRAILOF ##	AXIS : axe conjugué	AXIS : axe pilote				Trailing off : désactivation déplacements conjugués à synchronisme axial

Mot-clé non valide pour la SINUMERIK FM-NC/840D/NCU571.

Mot-clé valide uniquement pour la SINUMERIK 810D.

Mot-clé/ Descripteur de fonction	Paramètre 1	Paramètre 2	Signification
FPRAON	AXIS : axe pour lequel l'avance par tour est activée	AXIS : axe/broche de laquelle l'avance par tour est déduite. Si aucun axe n'est programmé, l'avance par tour sera déduite de la broche maître.	Feedrate per Revolution axial On : activation de l'avance par tour pour un axe
FRAOF	AXIS : axes, pour lesquels l'avance par tour est désactivée.		Feedrate per Revolution axial Off : désactivation de l'avance par tour L'avance par tour peut être désactivée simultanément pour plusieurs axes à la fois. Il est possible de programmer autant d'axes que le nombre maximal permis par bloc.

Mot-clé/ Descripteur de sous-programme	Paramètre 1	Paramètre 2	Paramètre 3	Paramètre 4	Paramètre 5	Signification
FPR	AXIS : axe/broche de laquelle est déduite l'avance par tour. Si aucun axe n'est programmé, l'avance par tour sera déduite de la broche maître.					Feedrate per Revolution : sélection d'un axe rotatif/d'une broche duquel/de laquelle sera déduite l'avance par tour (G95) des axes de contournage. Si aucun axe/aucune broche n'est programmé(e), l'avance par tour sera déduite de la broche maître. Le réglage effectué avec FPRest valide de façon modale.

A la place de l'axe, on peut aussi programmer une broche : FPR(S1) ou FPR(SPI(1))

Appels de sous-programmes prédéfinis

3. Transformations						
Mot-clé/ Descripteur de sous-programme	Paramètre 1	Paramètre 2	Signification			
TRACYL	REAL: diamètre de travail	INT : numéro de la transformation	Cylindre: transformation de surface latérale. On peut régler plusieurs transformations dans un même canal. Le numéro de transformation indique quelle est la transformation à activer. Si le paramètre 2 manque, le groupe de transformation figurant dans les paramètres machine (PM) est activé.			
TRANSMIT	INT : numéro de la transformation		Transmit: transformation polaire. On peut régler plusieurs transformations dans un même canal. Le numéro de transformation indique quelle est la transformation à activer. Si le paramètre manque, le groupe de transformation figurant dans les paramètres machine (PM) est activé.			
TRAANG	REAL: angle	INT : numéro de la transformation	Transformation axe oblique : On peut régler plusieurs transformations dans un même canal. Le numéro de transformation indique quelle est la transformation à activer. Si le paramètre 2 manque, le groupe de transformation figurant dans les paramètres machine (PM) est activé. Si l'angle n'est pas programmé(TRAANG(,2) ou TRAANG), c'est l'angle précédent qui agit de manière modale.			
TRAORI #	INT : numéro de la transformation		Transformation orientated: transformation 4 ou 5 axes On peut régler plusieurs transformations dans un même canal. Le numéro de transformation indique quelle est la transformation à activer.			
TRAFOOF			Désactivation de transformation			

A chaque type de transformation correspond une instruction relative à une transformation par canal. En cas de pluralité de transformations du même type par canal, il est possible de choisir la transformation correspondante en paramétrant l'instruction.

Il est possible de désactiver la transformation soit en effectuant un changement de transformation, soit en procédant à une désactivation explicite.

Mot-clé non valide pour la SINUMERIK FM-NC/810D/NCU571.

4. Broches			
Mot-clé/ Descripteur de sous-programme	Paramètre 1	Paramètre 2 et suivants	Signification
SPCON	INT : numéro de broche	INT : numéro de broche	Spindle position control on : commutation mode broche en asservissement de position
SPCOF	INT : numéro de broche	INT : numéro de broche	Spindle position control off : commutation mode broche en régulation de vitesse de rotation
SETMS	INT : numéro de broche		Set master-spindle: déclaration de la broche comme broche maître pour le canal courant. SETMS() sans spécification de paramètre active le pré réglage défini par les paramètres machine.

5. Rectification		
Mot-clé/ Descripteur de sous-programme	Paramètre 1	Signification
GWPSO	INT : numéro de broche	Grinding wheel peripheral speed on: activation vitesse périphérique de la meule constante. Si on ne programme pas de numéro de broche, la vitesse périphérique de meule est activée pour la broche entraînant la meule active.
GWPSOF	INT : numéro de broche	Grinding wheel peripheral speed off: désactivation vitesse périphérique de la meule constante. Si on ne programme pas de numéro de broche, la vitesse périphérique de meule est désactivée pour la broche entraînant la meule active.
TMON	INT : numéro de broche	Tool monitoring on: activation de la surveillance d'outil. Si on ne programme pas de numéro T, on active la surveillance pour l'outil actif.
TMOF	INT : numéro T	Tool monitoring off: désactivation de la surveillance d'outil. Si on ne programme pas de numéro T, on désactive la surveillance pour l'outil actif.

Appels de sous-programmes prédéfinis

6. Chariotage				
Mot-clé/ Descripteur de sous-programme	Paramètre 1	Paramètre 2	Paramètre 3	Signification
CONTPRON	VAR REAL [, 11]: table de contour	CHAR: direction de chariotage "L": longitudinale "P": transversale "N": transversale, usinage int. "G": longitudinale, usinage int.		Contour preparation on : activation de la préparation des références. Les programmes de contour appelés par la suite sont décomposés en déplacements élémentaires qui sont rangés dans la table de contour. On obtient, en retour, l'indication du nombre de détalonnages.
EXECUTE	VAR INT : état d'erreur			EXECUTE : activation de l'exécution du programme. Retour à l'exécution normale du progr. après préparation des références ou définition d'une zone de protection.

7. Exécution d'une table		
Mot-clé/ Descripteur de sous-programme	Paramètre 1	Signification
EXECTAB	VAR REAL [11] : Elément de la table de déplacements	Execute table: exécution d'un élément de la table de déplacements.

8. Zones de protection						
Mot-clé/ Descripteur de fonction	Paramètre 1	Paramètre 2	Paramètre 3	Paramètre 4	Paramètre 5	Signification
CPROTDEF	INT : numéro de zone de protection	BOOL: TRUE: zone de protection orientée outil	INT: 0: les paramètres 4 et 5 ne sont pas exploités 1: le paramètre 4 est exploité 2: le paramètre 5 est exploité 3: les paramètres 4 et 5 sont exploités	REAL: limitation en sens plus	REAL: limitation en sens moins	Chanel-specific protection area definition: définition d'une zone de protection spécifique au canal

Mot-clé/ Descripteur de fonction	Paramètre 1	Paramètre 2	Paramètre 3	Paramètre 4	Paramètre 5	Signification
NPROTDEF	INT : numéro de zone de protection	BOOL : TRUE : zone de protection orientée outil	INT : 0: les paramètres 4 et 5 ne sont pas exploités 1: le paramètre 4 est exploité 2: le paramètre 5 est exploité 3: les paramètres 4 et 5 sont exploités	REAL : limitation en sens plus	REAL : limitation en sens moins	NCK-specific protection area definition : définition d'une zone de protection spécifique machine
CPROT	INT : numéro de zone de protection	INT : option 0: zone de protection off 1:préactivation zone de protection 2: zone de protection on	REAL: décalage de la zone de protection dans le 1er axe du canal (= axe sur lequel sont reproduits des axes géométriques)	REAL: décalage de la zone de protection dans le 2me axe du canal	REAL: décalage de la zone de protection dans le 3me axe du canal	Activation/désactivation d'une zone de protection spécifique au canal
NPROTNPROT	INT : numéro de zone de protection	INT : option 0: zone de protection off 1:préactivation zone de protection 2: zone de protection on	REAL: décalage de la zone de protection dans le 1er axe du canal (= axe sur lequel sont reproduits des axes géométriques)	REAL: décalage de la zone de protection dans le 2me axe du canal	REAL: décalage de la zone de protection dans le 3me axe du canal	Activation/désactivation d'une zone de protection spécifique à la machine
EXECUTE	VAR INT: état d'erreur					EXECUTE: activation de l'exécution du programme. Retour à l'exécution normale du programme après préparation des références ou définition d'une zone de protection.

Appels de sous-programmes prédéfinis

9. Prétraitement des blocs		
Mot-clé/ Descripteur de fonction	Paramètre 1	Signification
STOPRE		Stop processing: arrêt du prétraitement des blocs jusqu'à ce que tous les blocs préparés pour l'exécution aient été exécutés

10. Interruptions		
Mot-clé/ Descripteur de fonction	Paramètre 1	Signification
ENABLE #	INT : numéro de l'entrée d'interruption	Activation d'une interruption: La routine d'interruption, affectée à l'entrée hardware dont le numéro est indiqué, est activée. Après l'instruction SETINT, une interruption est activée.
DISABLE #	INT : numéro de l'entrée d'interruption	Désactivation d'une interruption: La routine d'interruption, affectée à l'entrée hardware dont le numéro est indiqué, est désactivée. Même une instruction de retrait rapide n'est pas exécutée. L'affectation entre entrée hardware et routine d'interruption effectuée à l'aide de SETINT est conservée et peut être réactivée à l'aide de l'instruction ENABLE.
CLRINT #	INT : numéro de l'entrée d'interruption	Annuler une interruption: effacement de l'affectation routine d'interruption et attributs - entrée d'interruption. De ce fait, la routine d'interruption est annulée. Lorsque survient l'interruption, aucune réaction n'a lieu.

Mot-clé non valide pour la SINUMERIK FM-NC/810D.

11. Synchronisation avec le déplacement		
Mot-clé/ Descripteur de fonction	Paramètre 1	Signification
CANCEL	INT : numéro de l'action synchrone	Abandon de l'action synchrone au déplacement modale dont le numéro est indiqué.

12. Définition de fonctions					
Mot-clé/ Descripteur de fonction	Paramètre 1	Paramètre 2	Paramètre 3	Paramètre 4-7	Signification
FCTDEF	INT : numéro de fonction	REAL: valeur limite inférieure	REAL: valeur limite supérieure	REAL: coefficients a0 - a3	Définition d'un polynôme utilisé pour les fonctions SYNFACT ou PUTFTOCF.

Appels de sous-programmes prédéfinis

14. Coordination de programmes							
Mot-clé/ Descripteur de sous- programme	Paramètre 1	Paramètre 2	Paramètre 3	Paramètre 4	Paramètre 5	Paramètre 6-8	Signification
INIT #	INT : numéro de canal	STRING: Indication de chemin	CHAR: Mode d'acquitte- ment **				Sélection d'un pro- gramme pour exécution dans un canal.
START #	INT : numéro de canal	INT : numéro de canal					Lancement des pro- grammes sélectionnés simultanément dans plusieurs canaux, à partir du programme en cours. Cet ordre n'a aucun effet pour le canal propre. 1: canal 1 ; 2: canal 2
WAITE #	INT : numéro de canal	INT : numéro de canal					Wait for end of program: attente de la fin du programme dans un autre canal
WAITM #	INT : numéro de memento 0-9	INT : numéro de canal	INT : numéro de canal	INT : numéro de canal			Wait: attendre qu'un memento marque soit atteint dans d'autres canaux. L'attente dure jusqu'à ce que l'instruction WAITM avec le même memento soit atteinte dans les autres canaux. Le numéro du canal propre peut aussi être spécifié.
WAITP	AXIS : descripteur d'axe	AXIS : descripteur d'axe	AXIS : descripteur d'axe	AXIS : descripteur d'axe	AXIS : descripteur d'axe	AXIS : descripteur d'axe	Wait for positioning axis: attendre que les axes de positionnement aient atteint leurs points finaux programmés.
WAITS	INT : numéro de broche	INT : numéro de broche	INT : numéro de broche	INT : numéro de broche	INT : numéro de broche		Wait for positioning spindle: attendre que les broches, programmées au préalable avec l'instruction SPOSA, aient atteint leurs points finaux programmés.

Mot-clé/ Descripteur de sous- programme	Paramètre 1	Paramètre 2	Paramètre 3	Paramètre 4	Paramètre 5	Paramètre 6-8	Signification
RET							Fin de sous-programme sans sortie de fonction vers l'AP
GET #	AXIS	AXIS	AXIS	AXIS	AXIS	AXIS	Prise en charge d'axe machine
GETD#	AXIS	AXIS	AXIS	AXIS	AXIS	AXIS	Prise en charge directe d'axe machine
RELEASE #	AXIS	AXIS	AXIS	AXIS	AXIS	AXIS	Libération d'axe machine

On peut aussi programmer une broche à la place de l'axe en utilisant la fonction SPI : GET(SPI(1))

#Mot-clé non valide pour la SINUMERIK FM-NC/810D/NCU571.

** Mode d'acquiescement :

Les ordres sont acquiescés par les éléments assurant leur exécution (canal, CN, ...).

Sans acquiescement : L'exécution du programme principal est poursuivie après envoi de l'ordre. Il n'y a pas de signalisation en retour de l'exécution de l'ordre.
Mode d'acquiescement "N" ou "n".

Acquiescement synchrone : L'exécution du programme principal est suspendue tant que l'élément récepteur n'a pas acquiescé l'ordre.

En cas d'acquiescement positif, l'ordre suivant est exécuté.

En cas d'acquiescement négatif, il y a affichage d'un message d'erreur. Mode d'acquiescement "S", "s" ou pas d'indication.

Pour certains ordres, le mode d'acquiescement est prédéterminé ; pour d'autres, il est programmable. Le mode d'acquiescement des ordres de coordination de programmes est toujours synchrone. Le mode synchrone est le mode d'acquiescement par défaut.

Appels de sous-programmes prédéfinis

15. Communication			
Mot-clé/ Descripteur de sous- programme	Paramètre 1	Paramètre 2	Signification
MMC #	STRING : ordre	CHAR : Mode acquiescement** "N": sans acquiescement "S": Acquiescement synchrone : Acquiescement asynchrone :	Commande MMC : ordre à l'interpréteur d'ordre de MMC

Mot-clé **non** valide pour la SINUMERIK FM-NC/810D.

** Mode d'acquiescement :

Les ordres sont acquiescés par les éléments assurant leur exécution (canal, CN, ...).

Sans acquiescement : L'exécution du programme principal est poursuivie après envoi de l'ordre. Il n'y a pas de signalisation en retour de l'exécution de l'ordre.

16. Coordination de programmes							
Mot-clé/ Descripteur de sous- programme	Paramètre 1	Paramètre 2	Paramètre 3	Paramètre 4	Paramètre 5	Paramètre 6-8	Signification
PUTFTOC #	REAL: valeur de correction	INT : numéro de paramètre	INT : numéro de canal	INT : numéro de broche			Put fine tool correction : correction d'outil fine
PUTFTOCF #	INT : N° de la fonction Le numéro utilisé ici est à indiquer dans FCTDEF.	VAR REAL: valeur de référence *)	INT : numéro de paramètre	INT : numéro de canal	INT : numéro de broche		Put fine tool correction function dependend: modification de la correction fine d'outil en ligne selon une fonction définie avec FCTDEF (polynôme de 3e degré max.).

*) Seules des variables système particulières sont autorisées comme valeur de référence. Elles sont décrites dans la liste des variables système.

#Mot-clé non valide pour la SINUMERIK FM-NC/810D/NCU571.

17. Accès aux données		
Mot-clé/ Descripteur de sous- programme	Paramètre 1	Signification
CHANDATA	INT : numéro de canal	Régler le numéro du canal pour l'accès aux données de canal (possible uniquement dans le bloc d'initialisation) ; les accès suivants se réfèrent au canal réglé avec CHANDATA.

18. Messages			
Mot-clé/ Descripteur de sous- programme	Paramètre 1	Paramètre 2	Signification
MSG	CHAINE- CARAC- TERES: Message		Message modal: affichage qui dure aussi longtemps qu'il n'y a pas de nouveau message.

18. Alarmes			
Mot-clé/ Descripteur de sous- programme	Paramètre 1	Paramètre 2	Signification
SETAL	INT : numéro d'alarme (alarme de cycle)		Set alarme : activation d'une alarme

19. Compensation			
Mot-clé/ Descripteur de sous- programme	Paramètre 1 à Paramètre 4		Signification
QECLRNON #	AXIS : numéro d'axe		Quadrant error compensation learning on: activation de la fonction "Apprentissage de la compensation des défauts aux transitions entre quadrants"
QECLRNOF #			Quadrant error compensation learning off: désactivation de la fonction "Apprentissage de la compensation des défauts aux transitions entre quadrants"

#) Mot-clé non valide pour la SINUMERIK FM-NC.

Appels de sous-programmes prédéfinis

20. Gestion des outils			
Mot-clé/ Descripteur de sous- programme	Paramètre 1	Paramètre 2	Signification
DELT #	STRING [32]: descripteur d'outil	INT : numéro "frère"	Effacement de l'outil. Le numéro "frère" peut être omis.
GETSELT #	VAR INT: numéro T (valeur en retour)	INT : numéro de broche	Délivre le numéro T présélectionné. En cas d'absence de numéro de broche, l'instruction s'applique à la broche maître.
SETPIECE #	INT : nombre de pièces	INT : numéro de broche	Tenir compte du nombre de pièces pour tous les outils affectés à la broche. En cas d'absence de numéro de broche, l'instruction s'applique à la broche maître.

#) Mot-clé non valide pour la SINUMERIK FM-NC.

24. Broches synchrones							
Mot-clé/ Descripteur de sous- programme	Paramètre 1	Paramètre 2	Paramètre 3	Paramètre 4	Paramètre 5	Paramètre 6	Signification
COUPDEF #	AXIS : axe conjugué	AXIS : axe pilote	REAL : numérateur - rapport de transmission	REAL : numérateur - rapport de transmission	STRING[8] : mode de changement de bloc : "NOC" : pas de commande de change-ment de bloc, changement de bloc immédiat, "FINE" : changement de bloc au synchronisme fin, "COARSE" : changement de bloc au synchronisme grossier, "IPOSTOP" : changement de bloc lors-que la valeur de consigne de synchro-nisme est atteinte. En l'absence de spécification du mode de changement de bloc, il n'in- tervient aucun changement par rapport au mode pré réglé.	STRING[2] : "DV": couplage par la valeur de consigne "AV" : couplage par la valeur réelle	Couple definition : définition paire de broches synchrones
COUPDEL #	AXIS : axe conjugué	AXIS : axe pilote					Couple delete : effacement groupe de broches synchrones
COUPRES #	AXIS : axe conjugué	AXIS : axe pilote					Couple reset : remise à zéro paire de broches synchrones. Les valeurs programmées deviennent invalides. dans les paramètres machine.

Pour les broches synchrones, la programmation des paramètres d'axe a lieu avec SPI(1) ou S1.

Mot-clé non valide pour la SINUMERIK FM-NC/810 D.

Appels de sous-programmes prédéfinis

24. Réducteur électronique/ Broche synchrone							
Mot-clé/ Descripteur de sous- programme	Paramètre 1	Paramètre 2	Paramètre 3	Paramètre 4	Paramètre 5	Paramètre 6-8	Signification
COUPON #	AXIS : axe conjugué	AXIS : axe pilote	REAL : position de l'axe asservi pour activation				Couple on : Activation du couplage des broches syn- chrones. Si aucune position n'est indiquée, le couplage est réalisé le plus vite possible (rampe). Lorsqu'une position est spécifiée pour l'axe ou la broche asservi, elle se réfère de manière absolue ou relative à l'axe ou à la broche pilote. Ce n'est que dans le cas où le paramètre 3 est indiqué, qu'il faut programmer les paramètres 4 et 5.
COUPOF #	AXIS : axe conjugué	AXIS : axe pilote	REAL : position (absolue) de l'axe asservi pour désactivation	REAL : position (absolue) de l'axe pilote pour désactivation			Couple off : Désactivation du couplage des broches synchrones. Les para- mètres de couplage sont conservés. Si des positions sont spécifiées, le couplage n'est supprimé qu'après que toutes les positions indiquées ont été dépassees. La broche asservie continue à tourner avec la dernière vitesse de rotation atteinte avant le découplage.
WAITC #	AXIS : axe/broche	STRING[8] : critère de changement de bloc	AXIS : axe/broche	STRING[8] : critère de changement de bloc			Wait for couple condition: Attente jusqu'à satisfaction du critère de changement de bloc Il est possible de programmer jusqu'à 2 axes/broches

							<p>Critère de changement de bloc : "NOC" : sans Critère de changement de bloc, changement de bloc immédiat, "FINE" : changement de bloc au "synchronisme fin", "COARSE": changement de bloc au "synchronisme grossier" et "IPOSTOP" : changement de bloc autorisé dès que la consigne de synchronisme est atteinte.</p> <p>Si le mode de changement de bloc n'est pas indiqué, le mode indiqué préreglé n'est pas modifié.</p>
--	--	--	--	--	--	--	--

Mot-clé non valide pour la SINUMERIK FM-NC/810D.

Appels de sous-programmes prédéfinis dans des actions synchrones au déplacement

Les sous-programmes prédéfinis suivants figurent exclusivement dans des actions synchrones au déplacement.

25. Procédures synchrones				
Mot-clé/ Descripteur de fonction	Paramètre 1	Paramètre 2	Paramètre 3 à paramètre 5	Signification
STOPREOF				<p>Stop preparation off : annulation de l'arrêt du prétraitement des blocs.</p> <p>Une action synchrone avec un STOPREOF provoque un arrêt du prétraitement des blocs après le prochain bloc prétraité (= bloc à exécuter). L'arrêt du prétraitement est annulé à la fin du bloc à exécuter ou quand la condition pour STOPREOF est remplie. L'ensemble des instructions d'actions synchrones avec STOPREOF est alors considéré comme exécuté.</p>
RDISABLE				Read in disable: blocage de l'introduction via l'interface.
DELDTG	<p>AXIS : axe (facultatif) pour effacement de la distance restant à parcourir. Si l'axe manque, l'effacement est déclenché pour les axes de contournage.</p>			<p>Delete distance to go: effacement de la distance restant à parcourir.</p> <p>Une action synchrone avec DELDTG provoque un arrêt du prétraitement des blocs après le prochain bloc prétraité (= bloc à exécuter). L'arrêt du prétraitement est annulé à la fin du bloc à exécuter ou quand la première condition pour DELDTG est remplie. La distance axiale au point final en cas d'effacement axial de la distance restant à parcourir se trouve dans la variable \$AA_DELT[<axe>].</p>

Mot-clé/ Descripteur de fonction	Paramètre 1	Paramètre 2	Paramètre 3 à paramètre 5	Signification
SYNFCT	INT : numéro de la fonction polynomiale définie avec FCTDEF	VAR REAL : variable résultat *)	VAR REAL : variable d'entrée **)	Si la condition pour l'action synchrone au déplacement est remplie, la fonction polynomiale définie par la première expression est appliquée à la variable d'entrée. La valeur obtenue est ensuite écrêtée vers le bas et vers le haut et affectée à la variable résultat.
FTOC	INT : numéro de la fonction polynomiale définie avec FCTDEF	VAR REAL : variable d'entrée **)	INT : longueur 1,2,3 INT : numéro de canal INT : numéro de broche	Modification de la correction d'outil fine en liaison avec une fonction définie avec FCTDEF (polynôme de 3e degré au maximum) Le numéro utilisé ici doit être indiqué dans l'instruction FCTDEF.

*) Seules des variables système particulières sont autorisées comme variables résultat. Elles sont décrites dans le manuel "Notions préliminaires" sous "Variable d'exécution".

***) Seules des variables système particulières sont autorisées comme variables d'entrée. Elles sont décrites dans le manuel "Notions préliminaires" dans la liste des variables système.

Fonctions prédéfinies

Un appel de fonction lance l'exécution d'une fonction prédéfinie. Les appels de fonction délivrent une valeur en retour. Ils peuvent figurer sous forme d'opérande dans une expression.

1. Système de coordonnées						
Mot-clé/ Descripteur de fonction	Résultat	Paramètre 1	Paramètre 2			Signification
CTRANS	FRAME	AXIS	REAL: décalage	paramètres 3-15 idem 1 ...	paramètres 4-16 idem 2 ...	Translation: décalage d'origine pour plusieurs axes. On programme un descripteur d'axe, puis la valeur correspondante dans le paramètre suivant. Avec CTRANS, on peut programmer des décalages pour 8 axes au maximum.
CROT	FRAME	AXIS	REAL: angle de rotation	paramètres 3/5, idem 1...	paramètres 4/6, idem 2...	Rotation: rotation du système de coordonnées courant. Nombre maximal de paramètres: 6 (un descripteur d'axe et une valeur par axe géométrique).

Mot-clé/ Descripteur de fonction	Résultat	Paramètre 1	Paramètre 2			Signification
CSCALE	FRAME	AXIS	REAL: facteur d'échelle	paramètres 3-15 idem 1 ...	paramètres 4-16 idem 2 ...	Scale: facteur d'échelle pour plusieurs axes. Nombre maximal de paramètres : 2 * nombre maxi. d'axes (couples descripteur d'axe - valeur). On programme des couples descripteur d'axe - valeur correspondante. CSCALE permet de programmer des facteurs d'échelle pour 8 axes au maximum.
CMIRROR	FRAME	AXIS		paramètres 2-8, idem 1...		Mirror: fonction miroir par rapport à un axe de coordonnées.

Les fonctions de type frame telles que CTRANS, CSCALE, CROT et CMIRROR servent à générer des expressions frame.

3. Fonctions géométriques					
Mot-clé/ Descripteur de fonction	Résultat	Paramètre 1	Paramètre 2	Paramètre 3	Signification
CALCDAT	BOOL: état d'erreur	VAR REAL [,2]: table avec points d'entrée (respec- tivement abscisse et or-donnée pour points 1,2,3, etc.)	INT : nombre de points d'entrée pour le calcul (3 ou 4)	VAR REAL [3]: résultat: abscisse, ordonnée du centre du cercle et rayon obtenus par le calcul	CALCDAT: Calculate circle data Calcule le rayon et le centre d'un cercle passant par 3 ou 4 points (d'après paramètre 1). Les points doivent être distincts.
INTERSEC	BOOL: état d'erreur	VAR REAL [11]: premier élément de contour	VAR REAL [11]: deuxième élément de contour	VAR REAL [2]: vecteur résultat: coordonnées du point d'intersection, abscisse et ordonnée	Intersection: Calcul du point d'intersection Le point d'intersection entre deux éléments de contour est calculé. Les coordonnées du point d'intersection sont des valeurs en retour. L'état d'erreur indique si un point d'intersection a été trouvé.

Fonctions prédéfinies

4. Fonctions d'axe				
Mot-clé/ Descripteur de fonction	Résultat	Paramètre 1	Paramètre 2	Signification
NOMAXE	AXIS : descripteur d'axe	STRING []: chaîne de caractères d'entrée		AXNAME: Get axname Convertit la chaîne de caractères d'entrée en un descripteur d'axe. Il se produit une alarme si la chaîne de caractères d'entrée ne comporte pas de nom d'axe valide.
SPI	AXIS : descripteur d'axe	INT : numéro d'axe		SPI: Convert spindle to axis Convertit le numéro de broche en descripteur d'axe. Il se produit une alarme si le paramètre ne comporte pas de numéro de broche valide.
ISAXIS	BOOL TRUE: axe disponible, sinon : FALSE	INT : numéro de l'axe géométrique (1 à 3)		Vérifie si les axes géométriques 1 à 3, qui ont été indiqués avec le paramètre machine \$MC_AXCONF_GEOAX_ASSIGN_TAB, sont disponibles.

4. Gestion des outils				
Mot-clé/ Descripteur de fonction	Résultat	Paramètre 1	Paramètre 2	Signification
NEWT #	INT : numéro T	STRING [32]: nom de l'outil	INT : numéro "frère"	Créer un nouvel outil (mettre à disposition les données d'outil). Le numéro "frère" peut être omis.
GETT #	INT : numéro T	STRING [32]: nom de l'outil	INT : numéro "frère"	Déterminer le numéro T correspondant à un nom d'outil

#) Mot-clé **non** valide pour la SINUMERIK FM-NC.

6. Fonctions de calcul				
Mot-clé/ Descripteur de fonction	Résultat	Paramètre 1	Paramètre 2	Signification
SIN	REAL	REAL		sinus
ASIN	REAL	REAL		arc sinus
COS	REAL	REAL		cosinus
ACOS	REAL	REAL		arc cosinus
TAN	REAL	REAL		tangente
ATAN2	REAL	REAL	REAL	arc tangente 2

Mot-clé/Descripteur de fonction	Résultat	Paramètre 1	Paramètre 2	Signification
SQRT	REAL	REAL		racine carrée
POT	REAL	REAL		puissance 2
TRUNC	REAL	REAL		troncature des décimales
ROUND	REAL	REAL		arrondissement des décimales
ABS	REAL	REAL		formation de la valeur absolue
LN	REAL	REAL		logarithme naturel
EXP	REAL	REAL		fonction exponentielle e^x

7. Opérations sur chaînes de caractères				
Mot-clé/Descripteur de fonction	Résultat	Paramètre 1	Paramètre 2 à paramètre 3	Signification
ISNUMBER	BOOL	STRING		Vérifie si une chaîne de caractères d'entrée peut être convertie en un nombre. Le résultat est TRUE si la conversion est possible.
NUMBER	REAL	STRING		Convertit une chaîne de caractères d'entrée en un nombre
TOUPPER	STRING	STRING		Convertit tous les caractères de la chaîne de caractères d'entrée en majuscules
TOLOWER	STRING	STRING		Convertit tous les caractères de la chaîne de caractères d'entrée en minuscules
STRLEN	INT	STRING		Le résultat est la longueur de la chaîne de caractères d'entrée jusqu'à la fin de la chaîne de caractères (0).
INDEX	INT	STRING	CHAR	Recherche le caractère (second paramètre) de la chaîne de caractères d'entrée (premier paramètre). En retour, il est fourni la position où ce caractère a été trouvé la première fois. La recherche s'effectue de gauche à droite. Le 1er caractère de la chaîne de caractères est affecté de l'indice 0.
RINDEX	INT	STRING	CHAR	Recherche le caractère (second paramètre) de la chaîne de caractères d'entrée (premier paramètre). En retour, il est fourni la position où ce caractère a été trouvé la première fois. La recherche s'effectue de droite à gauche. Le 1er caractère de la chaîne de caractères est affecté de l'indice 0.
MINDEX	INT	STRING	STRING	Recherche, dans la chaîne de caractères d'entrée (premier paramètre), l'un des caractères indiqués dans le second paramètre. En retour, il est fourni la position où ce caractère a été trouvé la première fois. La recherche s'effectue de gauche à droite. Le 1er caractère de la chaîne de caractères est affecté de l'indice 0.
SUBSTR	INT	STRING	STRING	Recherche, dans la chaîne de caractères d'entrée (premier paramètre), la chaîne de caractères indiquée dans le second paramètre. En retour, il est fourni l'indice du caractère où cette chaîne de caractères a été trouvée la première fois. Le 1er caractère de la chaîne de caractères est affecté de l'indice 0.

Mot-clé/ Descripteur de fonction	Résultat	Paramètre 1	Paramètre 2 à paramètre 3	Signification
AXSTRING	STRING	AXIS		Convertit un descripteur d'axe en une chaîne de caractères.
AXNAME	AXIS	STRING		Convertit une chaîne de caractères en descripteur d'axe.

Types de données

Types de données		
Type	Observations	Valeurs disponibles
INT	nombres entiers avec signe	$\pm (2^{31} - 1)$
REAL	nombres réels (nombres rationnels avec point décimal, LONG REAL selon norme IEEE)	$\pm (10^{-300} \dots 10^{+300})$
BOOL	valeurs booléennes TRUE, FALSE ou 1, 0	1, 0
CHAR	1 caractère selon code ASCII	0 ... 255
STRING	chaîne de caractères, nombre de caractères entre [], maximum 200 caractères	suite de caractères ASCII
AXIS	nom d'axe exclusivement (adresse d'axe)	tous les descripteurs d'axe disponibles dans le canal
FRAME	indications géométriques pour translation, rotation, facteur d'échelle, fonction miroir	—

15.5 Liste des variables système (programmes pièce)

Paramètres d'outil

	Nom	Type	Observations		
Paramètres de correction d'outil	\$TC_DPx[t,d]		t : numéro T 1 – 32000 d: numéro de tranchant / numéro D 1 – 9 x: le nombre de paramètres est spécifié par paramètre machine		
	\$TC_DP1	INT	type d'outil		
	\$TC_DP2	REAL	position du tranchant		
	\$TC_DP3		géométrie – Longueur 1		
	\$TC_DP4		géométrie – Longueur 2		
	\$TC_DP5		géométrie – Longueur 3		
	\$TC_DP6		géométrie – Rayon		
	\$TC_DP7		géométrie – largeur de rainure b/ rayon d'arrondi		
	\$TC_DP8		géométrie – dépassement k		
	\$TC_DP11		géométrie – angle pour outils de fraisage coniques		
	\$TC_DP12		usure – longueur 1		
	\$TC_DP13		usure – longueur 2		
	\$TC_DP14		usure – longueur 3		
	\$TC_DP15		usure – rayon		
	\$TC_DP16		usure – largeur de rainure b/ rayon d'arrondi		
	\$TC_DP17		usure – dépassement k		
	\$TC_DP20		angle pour outils de fraisage coniques		
	\$TC_DP21		cote de base - Longueur 1		
	\$TC_DP22		cote de base - Longueur 2		
	\$TC_DP23		cote de base - Longueur 3		
	\$TC_DP24		dépouille		
	Paramètres de tranchant Utilisateur OEM		\$TC_DPCx[t,d]	REAL	t : numéro T 1 – 32000 d: numéro de tranchant / numéro D 1 – 9 x: le nombre de paramètres est spécifié par paramètre machine
			\$TC_DPC1		
			\$TC_DPC2		
\$TC_DPC3					
\$TC_DPC4					
\$TC_DPC5					
\$TC_DPC6					
\$TC_DPC7					

15.5 Liste des variables système (programmes pièce)

	Nom	Type	Observations
	\$TC_DPC8		
	\$TC_DPC9		
Paramètres spécifiques à un outil de rectification	\$TC_TPGx[t]		t : numéro T 1 – 32000
	\$TC_TPG1	INT	numéro de broche
	\$TC_TPG2		règle de concaténation
	\$TC_TPG9		numéro de paramètre pour calcul du rayon
	\$TC_TPG3	REAL	rayon minimal de la meule
	\$TC_TPG4		largeur minimale de la meule
	\$TC_TPG5		largeur courante de la meule
	\$TC_TPG6		vitesse de rotation maximale
	\$TC_TPG7		vitesse périphérique maximale
\$TC_TPG8	angle d'inclinaison pour meule inclinée		
Paramètres de surveillance Gestion des outils	\$TC_MOPx[t,d]	INT	t : numéro T 1 1 – 32000 d : numéro tranchant / numéro D 1 – 9
	\$TC_MOP1		limite de préavis durée de vie
	\$TC_MOP2		durée de vie résiduelle
	\$TC_MOP3		limite de préavis nombre de pièces
	\$TC_MOP4		Nombre de pièces résiduel
Paramètres de surveillance Utilisateur OEM	\$TC_MOPCx[t,d]	INT	t : numéro T 1 – 32000 d : numéro de tranchant / numéro D 1 – 9 x : le nombre de paramètres est spécifié par paramètre machine
	\$TC_MOPC1		
	\$TC_MOPC2		
	\$TC_MOPC3		
	\$TC_MOPC4		
	\$TC_MOPC5		
	\$TC_MOPC6		
	\$TC_MOPC7		
	\$TC_MOPC8		
\$TC_MOPC9			
Paramètres spécifiques à un outil Gestion des outils	\$TC_TPx[t]		t : numéro T 1 – 32000
	\$TC_TP2	STRING	descripteur d'outil
	\$TC_TP1	INT	numéro "frère"
	\$TC_TP3		dimension vers la gauche
	\$TC_TP4		dimension vers la droite
	\$TC_TP5		dimension vers le haut

	Nom	Type	Observations
	\$TC_TP6		dimension vers le bas
	\$TC_TP7		type d'emplacement de magasin
	\$TC_TP8		etat
	\$TC_TP9		type de surveillance d'outil
	\$TC_TP11		stratégie de remplacement d'outil
	\$TC_TP10		info outil
Paramètres spécifiques à un outil Utilisateur OEM	\$TC_TPCx[t]	REAL	t : numéro T 1 – 32000 x: le nombre de paramètres est spécifié par paramètre machine
	\$TC_TPC1		
	\$TC_TPC2		
	\$TC_TPC3		
	\$TC_TPC4		
	\$TC_TPC5		
	\$TC_TPC6		
	\$TC_TPC7		
	\$TC_TPC8		
	\$TC_TPC9		
	\$TC_TPC10		
Paramètres d'emplacement de magasin Gestion des outils	\$TC_MPPx[n,m]		n : numéro de magasin physique m : numéro d'emplacement physique
	\$TC_MPP3	BOOL	activation / désactivation prise en compte emplacements voisins
	\$TC_MPP1	INT	nature d'emplacement
	\$TC_MPP2		type d'emplacement
	\$TC_MPP6		numéro T de l'outil affecté à cet emplacement
	\$TC_MPP4		état de l'emplacement
	\$TC_MPP5		indice de nature d'emplacement
Paramètres d'emplacement de magasin Utilisateur OEM	\$TC_MPPCx[n,m]	INT	n : numéro de magasin physique m : numéro d'emplacement physique xx : le nombre de paramètres est spécifié par paramètre machine
	\$TC_MPPC1		
	\$TC_MPPC2		
	\$TC_MPPC3		
	\$TC_MPPC4		
	\$TC_MPPC5		
	\$TC_MPPC6		
	\$TC_MPPC7		
	\$TC_MPPC8		

15.5 Liste des variables système (programmes pièce)

	Nom	Type	Observations
	\$TC_MPPC9		
	\$TC_MPPC10		
Paramètres de magasin : distance au poste de changement d'outil	\$TC_MDPx[n,m]	INT	n : numéro de magasin physique m : numéro d'emplacement physique
	\$TC_MDP1		distance entre le poste de changement d'outil du magasin n et l'emplacement m du 1er magasin interne
	\$TC_MDP2		dist. entre le poste de changement d'outil du magasin n et l'emplacement m du 2me magasin interne
Paramètres de magasin : Hiérarchie des types d'emplacements	\$TC_MPTH[n,m]	INT	hiérarchie des types d'emplacement n : hiérarchie 0 – 7 m : type d'emplacement 0 – 7
Paramètres de description du magasin Gestion des outils	\$TC_MAPx[n]		n : numéro de magasin 1 – ...
	\$TC_MAP2	STRING	descripteur du magasin
	\$TC_MAP1	INT	type de magasin
	\$TC_MAP3		état du magasin
	\$TC_MAP4		relation avec un magasin suivant
	\$TC_MAP5		relation avec un magasin précédent
	\$TC_MAP6		nombre de lignes
	\$TC_MAP7		nombre de colonnes
	\$TC_MAP8		position courante du magasin par rapport à position de changement d'outil
	\$TC_MAMPx[n]		n : numéro de magasin 1 – ...
	\$TC_MAMP1	STRING	descripteur du bloc AP de magasin
	\$TC_MAMP2	INT	type de recherche d'outil et type de recherche d'emplacement libre
	Paramètres de description du magasin Utilisateur OEM	\$TC_MAPCx[n]	INT
\$TC_MAPC1			
\$TC_MAPC2			
\$TC_MAPC3			
\$TC_MAPC4			
\$TC_MAPC5			
\$TC_MAPC6			
\$TC_MAPC7			
\$TC_MAPC8			
\$TC_MAPC9			
\$TC_MAPC10			

Paramètres de magasin

	Nom	Type	Observations
Relation entre emplacement magasin et broche	\$TC_MLSR[n,m]=0	INT	<p>relation entre emplacement tampon n et emplacement tampon m :</p> <p>n : Physique N° emplacement magasin, type emplacement≠ 'broche',</p> <p>m: Physique N° emplacement magasin, type emplacement'broche',m:</p> <p>Ceci permet, par ex., de définir des affectations préhenseurs...broches.</p> <p>La valeur du paramètre est figée à zéro.</p> <p>L'opération d'écriture définit une relation ; l'opération de lecture vérifie si une relation donnée existe. Si la relation n'existe pas, une alarme est émise : "define links of grippers, ... to spindles".</p>

Compensations

	Nom	Type	Observations
Valeurs de compensation des systèmes de mesure	\$AA_ENC_COMP_x[n,m,a]		n: numéro capteur 0 – 1 m: numéro point a: axe machine
	\$AA_ENC_COMP[n,m,a]	REAL	valeurs de compensation
	\$AA_ENC_COMP_STEP[n,a]		pas
	\$AA_ENC_COMP_MIN[n,a]		position de départ de la compensation
	\$AA_ENC_COMP_MAX[n,a]		position de fin de la compensation
	\$AA_ENC_COMP_IS_MODULO[n,a]	BOOL	Compensation = Modulo
Compensation des défauts aux transitions entre quadrants	\$AA_QEC_x[n,m,a]		n : 0 m : n° du point, 0 – contenu = paramètre machine a : axe machine
	\$AA_QEC[n,m,a]	REAL	résultat du processus d'apprentissage
	\$AA_QEC_COARSE_STEPS[n,a]	INT	valeurs de compensation : quantification grossière de la caractéristique
	\$AA_QEC_FINE_STEPS[n,a]		quantification fine de la caractéristique
	\$AA_QEC_ACCEL_1[n,a]	REAL	accélération au 1er coude
	\$AA_QEC_ACCEL_2[n,a]		accélération au 2ème coude
	\$AA_QEC_ACCEL_3[n,a]		accélération au 3ème coude
	\$AA_QEC_MEAS_TIME_1[n,a]		durée de mesure pour zone \$AA_QEC_ACCEL_1

	Nom	Type	Observations
	\$AA_QEC_MEAS_TIME_2 [n,a]		durée de mesure pour zone \$AA_QEC_ACCEL_2
	\$AA_QEC_MEAS_TIME_3 [n,a]		durée de mesure pour zone \$AA_QEC_ACCEL_3
	\$AA_QEC_TIME_1[n,a]		1er temps de filtrage pour opérateur de commande anticipatrice
	\$AA_QEC_TIME_2[n,a]		2ème temps de filtrage pour opérateur de commande anticipatrice
	\$AA_QEC_LEARNING_RATE [n,a]		Vitesse d'apprentissage pour réseau neuronal
	\$AA_QEC_DIRECTIONAL [n,a]	BOOL	1 (TRUE): compensation dépendante du sens 0 (FALSE): compensation indépendante du sens

Compensations

	Nom	Type	Observations
Compensation avec interpolation	\$AN_CEC_xxx_yyy[n,m]		n : n° de table de compensation, 0 – valeur maxi (réglable via PM) m: n° des points intermédiaires, 0 – valeur maxi (réglable via PM)
	\$AN_CEC[n,m]	REAL	valeurs de compensation
	\$AN_CEC_INPUT_AXIS[n]	AXIS	nom de l'axe dont la valeur de consigne sert d'entrée pour la table de compensation
	\$AN_CEC_OUTPUT_AXIS[n]		nom de l'axe sur laquelle agit la sortie de la table de compensation
	\$AN_CEC_STEP[n]	REAL	espace entre les valeurs de compensation
	\$AN_CEC_MIN[n]		position de départ de la compensation
	\$AN_CEC_MAX[n]		position finale de la table de compensation
	\$AN_CEC_DIRECTION[n]	INT	active la fonction de dépendance du sens pour la compensation
	\$AN_CEC_MULT_BY_TABLE [n]		numéro de la table dont la valeur de sortie doit être multipliée avec la valeur de sortie de la table de compensation 0 : deux sens de déplacement de l'axe de base 1 : sens de déplacement positif de l'axe de base -1 : sens de déplacement négatif de l'axe de base
\$AN_CEC_IS_MODULO[n]	BOOL	1 (TRUE): répétition cycl. de la table de compensation 0 (FALSE): pas de répétition cyclique de la table de compensation	

Zones de protection

	Nom	Type	Observations
Zones de protection spécifiques NCK	\$SN_PA_xxx_yyy[n,m]		n : numéro de la zone de protection, 0 – valeur maxi (réglable via PM) m: numéro de l'élément de contour 0 – 10
	\$SN_PA_ACTIV_IMMED[n]	BOOL	zone de protection activée immédiatement ?
			1 (TRUE) 0 (FALSE) après le lancement de la commande et la prise de référence des axes non
	\$SN_PA_T_W[n]	CHAR	zone de protection spécifique à la pièce / l'outil 0 : zone de protection spécifique à la pièce 3 : zone de protection spécifique à l'outil
	\$SN_PA_ORI[n]	INT	orientation de la zone de protection 0 : polygone dans le plan 1er - 2me axe géo 1 : polygone dans le plan 3me - 1er axe géo 2 : polygone dans le plan 2me - 3me axe géo
	\$SN_PA_LIM_3DIM[n]		code pour la limitation de la zone de protection dans l'axe perpendiculaire au polygone 0 : pas de limitation 1 : limitation en sens positif 2 : limitation en sens négatif 3 : limitation dans les deux sens
	\$SN_PA_PLUS_LIM[n]	REAL	imitation positive de la zone de protection dans l'axe perpendiculaire au polygone :
	\$SN_PA_MINUS_LIM[n]		limitation négative de la zone de protection dans l'axe perpendiculaire au polygone :
	\$SN_PA_CONT_NUM[n]	INT	nombre d'éléments de contour validesn :
	\$SN_PA_CONT_TYPE[n,m]		type (G1, G2, G3) de l'élément de contourn :
	\$SN_PA_CONT_ORD[n,m]	REAL	point final de l'élément de contour (ordonnée)
	\$SN_PA_CONT_ABS[n,m]		point final de l'élément de contour (abscisse)
	\$SN_PA_CENT_ORD[n,m]		centre de l'élément de contour (ordonnée)
\$SN_PA_CENT_ABS[n,m]	centre de l'élément de contour (abscisse)		

15.6 Liste des variables système

15.6.1 Paramètres R

Nom	Type	Accès		Arrêt du prétraitement des blocs		Signification Plage de valeurs	Indice Plage de valeurs
		Prog. pièce	Action synchr.	Prog. pièce	Action synchr.		
Rn ou R[n]	REAL			non (à créer avec STOPRE		Variable de calcul dans mémoire statique	n : Numéro variable calcul 0 - Maximum (défini par PM)
\$Rn ou \$R[n]			R/W	éventuellement)			

15.6.2 Frames

\$P_UIFR[n]	FRAME					frames réglables (G54 etc.)	n : Nombre 1 - 100 (réglable via PM)
-------------	-------	--	--	--	--	-----------------------------	--

15.6.3 Données du porte-outil

\$TC_CARRx[n]	REAL					Préréglage standard = 0; c.à.d. NCK ne connaît pas ces données.	n : nb de paramètres max. (réglable via PM)
\$TC_CARR1[n]						composante x du vecteur offset I1	
\$TC_CARR2[n]						comp. y du vecteur offset I1	
\$TC_CARR3[n]						comp. z du vecteur offset I1	
\$TC_CARR4[n]						comp. x du vecteur offset I2	
\$TC_CARR5[n]						comp. y du vecteur offset I2	
\$TC_CARR6[n]						comp. z du vecteur offset I2	
\$TC_CARR7[n]						comp. x de l'axe de rotation v1	
\$TC_CARR8[n]						comp. y de l'axe de rotation v1	
\$TC_CARR9[n]						comp. z de l'axe de rotation v1	
\$TC_CARR10[n]						comp. x de l'axe de rotation v2	
\$TC_CARR11[n]						comp. y de l'axe de rotation v2	
\$TC_CARR12[n]						comp. z de l'axe de rotation v2	
\$TC_CARR13[n]						angle de rotation α_1 en degrés	
\$TC_CARR14[n]						angle de rotation α_2 en degrés	

15.6.4 Zones de protection spécifiques à un canal

Nom	Type	Accès		Arrêt du prétraitement des blocs		Signification Plage de valeurs	Indice Plage de valeurs
		Prog. pièce	Action synchr.	Prog. pièce	Action synchr.		
\$SC_PA_xxx_yyy[n,m]							n : numéro de la zone de protection 0 – valeur maxi
\$SC_PA_ACTIV_IMM ED[n]	BOOL					zone de protection activée immédiatement ? 0, 1 1 (TRUE) 0 (FALSE) après le lancement de la commande et la prise de référence des axes non	(réglable via PM) m: numéro de l'élément de contour 0 - 10
\$SC_PA_T_W[n]	CHAR					zone de protection spécifique à la pièce / l'outil 0 : zone de protection spécifique à la pièce 3 : zone de protection spécifique à l'outil	
\$SC_PA_PA_ORI[n]	INT					Orientation de la zone de protection : Tracé polygonal dans le plan ... 0 : depuis le 1er et 2e axe géo 1 : depuis le 3e et 1er axe géo 2 : depuis le 2e et 3e axe géo	
\$SC_PA_LIM_3DIM[n]						code pour la limitation de la zone de protection dans l'axe perpendiculaire au polygone 0 : pas de limitation 1 : limitation en sens positif 2 : en sens négatif 3 : limitation dans les deux sens	
\$SC_PA_PLUS_LIM [n]	REAL					Limitation positive de la zone de protection dans le sens moins de l'axe perpendiculaire au tracé polygonal	
\$SC_PA_MINUS_LIM[n]						Limitation négative de la zone de protection dans le sens moins de l'axe perpendiculaire au tracé polygonal	
\$SC_PA_CONT_NUM [n]	INT					nombre d'éléments de contour valides	
\$SC_PA_CONT_TYP [n,m]						type (G1, G2, G3) de l'élément de contour	

15.6 Liste des variables système

Nom	Type	Accès		Arrêt du prétraitement des blocs		Signification Plage de valeurs	Indice Plage de valeurs
		Prog. pièce	Action synchr.	Prog. pièce	Action synchr.		
\$SC_PA_CONT_ORD [n,m] \$SC_PA_CONT_ABS [n,m] \$SC_PA_CONT_ORD [n,m] \$SC_PA_CONT_ABS [n,m]	REAL					point final de l'élément de contour (ordonnée) point final de l'élément de contour (abscisse) centre de l'élément de contour (ordonnée) centre de l'élément de contour (abscisse)	
\$AC_PARAM[n]	REAL	R/W	R/W	R/W		Paramètre dynamique pour les actions synchrones au déplacement	n : numéro du paramètre 0 - 49.
\$AC_MEMENTO[n]	INT					mémento, compteur pour actions synchrones au déplacement	n : n° de memento 0 - maximum (défini dans PM \$MC_...)
\$AN_MEMENTO[n]	BOOL					Variable memento pour actions synchrones au déplacement (accessible depuis tous les canaux)	n : n° de memento 0 - maximum (défini dans PM \$MC_...)
\$AC_FIFOx[n]	REAL	R/W	R/W	R/W		FIFO pour actions synchrones au déplacement et mesures cycliques x: 1-10	n : numéro de paramètre 0 - max élément FIFO n=0: écrire : ranger nouvelle valeur dans FIFO lire : lire l'élément le plus ancien et l-extraire du FIFO n=1: Accès en lecture à l'élément le plus ancien n=2: Accès en lecture à l'élément le plus récent n=3: Somme de tous les éléments dans FIFO, quand Bit 0 dans MD \$MC_MM_MODE_FI FO n=4: Accès en lecture au nombre courant d'éléments FIFO n=5: élément le plus ancien n=6: l'élément précédant le plus ancien etc.

Frames

Nom	Type	Accès		Arrêt du prétraitement des blocs		Signification Plage de valeurs	Indice Plage de valeurs
		Prog. pièce	Action synchr.	Prog. pièce	Action synchr.		
\$P_PFRAME	FRAME	R/W				frame programmable courant	
\$P_ACTFRAME						frame global courant	
\$P_IFFRAME						frame réglable courant	
\$P_UIFRNUM	INT	R				numéro du \$P_UIFR actif	

15.6.5 Entrées/sorties

Nom	Type	Accès		Arrêt du prétraitement des blocs		Signification Plage de valeurs	Indice Plage de valeurs
		Prog. pièce	Action synchr.	Prog. pièce	Action synchr.		
\$A_xxx[n]							n : n° E/S
\$A_IN[n]	BOOL	R	R	R		entrée TOR CN	1 - valeur maxi définie via PM
\$A_OUT[n]		R/W	R/W			sortie TOR NC	
\$A_INA[n]	REAL	R	R			entrée analogique CN	
\$A_OUTA[n]						sortie analogique CN	
\$A_INCO[n]	BOOL					entrée de comparateur CN	

15.6.6 Lecture et écriture de variables AP

Nom	Type	Accès		Arrêt du prétraitement des blocs		Signification Plage de valeurs	Indice Plage de valeurs	
		Prog. pièce	Action synchr.	Prog. pièce	Action synchr.			
\$A_DBx[n]				R			n : offset de position dans la Dual Port Ram ent.CN et AP x : B ou W ou D ou R	
\$A_DBB[n]	INT		R/W			Lire/Ecrire octet (8 bits) de/dans AP		
\$A_DBW[n]						R		Lire/Ecrire mot (16 bits) de/dans AP
\$A_DBD[n]						R		Lire/Ecrire double mot (32 bits) de/dans AP
\$A_DBR[n]	REAL		R/W			Lire/Ecrire données réelles (32 Bit) de/dans AP		

15.6.7 Temps

Nom	Type	Accès		Arrêt du prétraitement des blocs		Signification Plage de valeurs	Indice Plage de valeurs
		Prog. pièce	Action synchr.	Prog. pièce	Action synchr.		
\$A_YEAR	INT	R	R	R		Système - Année 0 - 99	
\$A_MONTH						Système - Mois 1 - 12	
\$A_DAY						Système - Jour 1- 31	
\$A_HOUR						Système - Heure 0 - 23	
\$A_MINUTE						Système - Minute 0 - 59	
\$A_SECOND						Système - Seconde 0 - 59	
\$A_MSECOND						Syst. - Milliseconde 0 - 999	
\$AC_TIMER[n]	REAL	R/W	R/W	R/W		Cellule temps en secondes La temporisation est comptée en multiples de la période d'appel de l'interpolateur;	n : numéro de la cellule de temps 1 - ... (définie dans PM)
\$AC_TIME			R			temporisation depuis le début de bloc en secondes	
\$AC_TIMEC						temporisation depuis le début de bloc en périodes d'appel de l'interpolateur	

15.6.8 Etats des canaux

Nom	Type	Accès		Arrêt du prétraitement des blocs		Signification Plage de valeurs	Indice Plage de valeurs
		Prog. pièce	Action synchr.	Prog. pièce	Action synchr.		
\$AC_MEA[n]	INT	R	R	R		1 (TRUE): le palpeur a déclenché	n : n° du palpeur de mesure 1 - 2
\$AC_TRAFO	INT					code de la transf. active; 0 (FALSE): Transf. désactivée	
\$AC_STAT						Etat courant du canal 1: abandon 2: actif 4: interruption 8: reset	
\$AC_PROG						Etat courant du programme 1: en cours 2: en attente 4: arrêté 8: interrompu	

Nom	Type	Accès		Arrêt du prétraitement des blocs		Signification Plage de valeurs	Indice Plage de valeurs
		Prog. pièce	Action synchr.	Prog. pièce	Action synchr.		
\$AC_IPO_BUF		R				Etat de remplissage du tampon de l'interpolateur (indique le nombre de blocs prédécodés) 0 - \$MC_MM_IPO_BUFFER_SIZE	
\$AC_SYNA_MEM						Nombre d'éléments de mémoire encore libres pour les actions synchrones au déplacement 0 - \$MC_MM_NUM_SYNC_ELEMENTS	
\$AC_LIFTFAST	INT	R/W		R/W		Code de la transf. active 0 : aucune course retour n'avait été activée 1 : course de retour avait été activée	
\$SP_ISTEST	BOOL	R				Vérifier le mode Test dans le programme pièce 0 (FALSE) : le test du programme n'était pas activé 1 (TRUE) : le test du programme était activé	

15.6.9 Variables pour trajectoire

Distances

Nom	Type	Accès		Arrêt du prétraitement des blocs		Signification Plage de valeurs	Indice Plage de valeurs
		Prog. pièce	Action synchr.	Prog. pièce	Action synchr.		
\$AC_PATHN	REAL		R			Paramètre de trajectoire normé / Valeur entre 0 = début de bloc et 1 = fin de bloc	
\$AC_DTBW						Distance géométrique depuis début de bloc dans SCP	
\$AC_DTBB						Distance géométrique depuis début de bloc dans SCB	
\$AC_DTEW						Distance géométrique depuis fin de bloc dans SCP	
\$AC_DTEB						Distance géométrique depuis fin de bloc dans SCB	

15.6 Liste des variables système

Nom	Type	Accès		Arrêt du prétraitement des blocs		Signification Plage de valeurs	Indice Plage de valeurs
		Prog. pièce	Action synchr.	Prog. pièce	Action synchr.		
\$AC_PLTBB						Distance curviligne depuis le début du bloc dans le SCB	
\$AC_PLTEB						Distance curviligne jusqu'en fin de bloc dans le SCB	
\$AC_DTEPW						Distance restant à parcourir pour pénétration, oscillation dans le SCP	
\$AC_DTEPB						Distance restant à parcourir pour pénétration, oscillation dans le SCB	
\$AC_DELT		R		R		Trajectoire non parcourue dans le SCP après effacement de la distance restant à parcourir sur la trajectoire avec DELDTG dans actions synchrones au déplacement	

Vitesses

Nom	Type	Accès		Arrêt du prétraitement des blocs		Signification Plage de valeurs	Indice Plage de valeurs
		Prog. pièce	Action synchr.	Prog. pièce	Action synchr.		
\$AC_OVR	REAL		R/W			correction avance tangentielle pour actions synchrones	
\$AC_VC						correction additive de l'avance tangentielle pour actions synchrones	
\$AC_VACTW			R			vitesse tangentielle dans le syst. de coordonnées pièce	
\$AC_VACTB						vitesse tangentielle dans le syst. de coordonnées de base	

\$P_F	REAL		R			dernière avance tangentielle F programmée	
-------	------	--	---	--	--	---	--

15.6.10 Valeurs polynomiales pour action synchrone

Nom	Type	Accès		Arrêt du prétraitement des blocs		Signification Plage de valeurs	Indice Plage de valeurs
		Prog. pièce	Action synchr.	Prog. pièce	Action synchr.		
\$AC_FCTxyL	REAL	R/W	R/W			Valeur limite x. Polynôme pour action synchrone SYNFACT / Fonction de traitement FCTDEF x	
\$AC_FCT1LL						Valeur limite inférieure	x = 1
\$AC_FCT2LL						Valeur lim. inf.	x = 2
\$AC_FCT3LL						Valeur lim. inf.	x = 3
\$AC_FCT1UL						Val. limite sup.	x = 1
\$AC_FCT2UL						Val. limite sup.	x = 2
\$AC_FCT3UL						Val. limite sup.	x = 3
\$AC_FCTxC[n]	REAL	R/W	R/W			Coefficient polynomial a0 – a3 pour polynôme x (SYNFACT) dans actions synchrones / fonction de traitement FCTDEF x	n : numéro du coefficient polynomial a0 – a3 0 - 3
\$AC_FCT1C[n]						x = 1	
\$AC_FCT2C[n]						x = 2	
\$AC_FCT3C[n]						x = 3	
\$AC_FCTLL[n]	REAL	R/W	R/W			Valeur limite inférieure polynôme n pour action synchrone SYNFACT / Fonction de traitement FCTDEF n	n : numéro du polynôme 1 - valeur PM \$MC_MM_NUM_FC_TDEF_ELEMENTS
\$AC_FCTUL[n]						Valeur limite supérieure polynôme n pour action synchrone SYNFACT / Fonction de traitement FCTDEF n	
\$AC_FCTx[n]	REAL	R/W	R/W			ax-coefficient polynôme n pour action synchrone SYNFACT / fonction de traitement FCTDEF n	n : numéro du polynôme 1 - valeur PM \$MC_MM_NUM_FC_TDEF_ELEMENTS
\$AC_FCT0[n]						x = 0	
\$AC_FCT1[n]						x = 1	
\$AC_FCT2[n]						x = 2	
\$AC_FCT3[n]						x = 3	

15.6.11 Variables système spécifiques à un axe

Positions et trajets

Nom	Type	Accès		Arrêt du prétraitement des blocs		Signification Plage de valeurs	Indice Plage de valeurs
		Prog. pièce	Action synchr.	Prog. pièce	Action synchr.		
\$AA_IW[axe]	REAL	R	R	R		valeur réelle dans le SCP	axe: axe machine
\$AA_IB[axe]						valeur réelle dans le SCB	
\$AA_IM[axe]						valeur réelle dans le SCM	
\$AA_ENC_ACTIVE [axe]	BOOL					1 (TRUE): le syst. de mesure activé travaille au-dessous de la fréquence limite du capteur	axe
\$AA_ENC1_ACTIVE [axe]						1 (TRUE): le codeur 1 travaille au-dessous de la fréquence limite du capteur	
\$AA_ENC2_ACTIVE [axe]						1 (TRUE): le codeur 2 travaille au-dessous de la fréquence limite du capteur	
\$VA_IMx[axe]	REAL	R	R	R		valeur réelle courante (mesurée) dans SCM ; les compensations (température, CEC) sont corrigées la variable fournit une valeur indéfinie quand la fréquence du codeur est dépassée	axe: axe machine
\$VA_IM[axe]						système de mesure activé	
\$VA_IM1[axe]						codeur 1	
\$VA_IM2[axe]						codeur 2	
\$AA_MW[axe]	REAL	R	R	R		valeur mesurée dans le SCP	axe: axe machine
\$AA_MM[axe]						valeur mesurée dans le syst. de coordonnées machine	
\$AC_DRF[axe]						décalage DRF	
\$AC_PRESET[axe]						dernière valeur Preset prescrite	
\$AA_ETRANS[axe]	REAL	R/W	R/W	R/W		décalage d'origine externe	
\$AA_OFF[axe]			R/W			déplacement forcé pour l'axe programmé	
\$AC_RETPOINT[axe]		R	R	R		pt de réaccostage du contour	
\$AA_SOFTENDP[axe]	REAL	R	R	R		position fin de course logiciel, sens positif	axe: axe machine
\$AA_SOFTENDN[axe]						position fin de course logiciel, sens négatif	
\$AA_DTxx[axe]	REAL	R	R	R		trajet axial pour axes de positionnement et axes synchrones dans les actions synchrones au déplacement	axe: axe machine axe de positionnement axe synchrone

Nom	Type	Accès		Arrêt du prétraitement des blocs		Signification Plage de valeurs	Indice Plage de valeurs
		Prog. pièce	Action synchr.	Prog. pièce	Action synchr.		
\$AA_DTBW[axe]						depuis début de bloc dans SCP	
\$AA_DTBB[axe]						depuis déb. du bloc dans SCB	
\$AA_DTEW[axe]						jusqu'en fin de bloc dans SCP	
\$AA_DTEB[axe]						jusqu'en fin de bloc dans SCB	
\$AA_DTEPW[axe]						pénétration axiale résiduelle d'oscillation dans le SCP	
\$AA_DTEPB[axe]						pénétration axiale résiduelle d'oscillation dans le SCB	
\$AA_DELT[axe]						distance axiale non parcourue dans le SCP après effacement de la distance axiale restant à parcourir avec DELDTG(axe) dans actions synchrones au déplacement	
\$AA_OSCILL_REVERSE_POS1[axe]	REAL		R	R		position d'inversion courante 1 pour oscillation; dans les actions synchrones la valeur de la donnée de réglage \$SA_OSCILL_REVERSE_POS1 est traitée en ligne.	axe: axe machine
\$AA_OSCILL_REVERSE_POS2[axe]						position d'inversion courante 2 pour oscillation; dans les actions synchrones la valeur de la donnée de réglage \$SA_OSCILL_REVERSE_POS2 est traitée en ligne.	
\$AA_MWx[axe]	REAL	R				résultat de mesure axiale événement déclencheur 1 dans SCP	
\$AA_MW1[axe]						événement décl. 2 dans SCP	
\$AA_MW2[axe]						événement décl. 3 dans SCP	
\$AA_MW3[axe]						événement décl. 4 dans SCP	
\$AA_MMx[axe]	REAL	R				résultat de mesure axiale événement décl. 1 dans SCM	
\$AA_MM1[axe]						événement décl. 2 dans SCM	
\$AA_MM2[axe]						événement décl. 3 dans SCM	
\$AA_MM3[axe]						événement décl. 4 dans SCM	
\$AA_MEACT[axe]	BOOL	R				1 (TRUE): mesure axiale activée pour X	
\$P_EP[axe]	REAL					dernière valeur de consigne programmée	
\$P_FA[axe]	REAL					dernière avance axiale progr.	

Vitesses axiales

Nom	Type	Accès		Arrêt du prétraitement des blocs		Signification Plage de valeurs	Indice Plage de valeurs
		Prog. pièce	Action synchr.	Prog. pièce	Action synchr.		
\$AA_OVR[axe]	REAL		R/W			correction axiale pour actions synchrones au déplacement	axe ou broche
\$AA_VC[axe]						valeur de correction additive pour l'avance tangentielle ou l'avance axiale	
\$xA_VACTy[axe]	REAL		R			vitesse axiale	axe
\$AA_VACTW[axe]						dans le SCP	
\$AA_VACTB[axe]						dans le SCB	
\$AA_VACTM[axe]						côté valeur de consigne dans le SCM	
\$VA_VACTM[axe]						côté valeur réelle dans le SCM	

Données d'entraînement (uniquement pour 611D)

Nom	Type	Accès		Arrêt du prétraitement des blocs		Signification Plage de valeurs	Indice Plage de valeurs
		Prog. pièce	Action synchr.	Prog. pièce	Action synchr.		
\$AA_LOAD[axe]	REAL	R	R	R		Sollicitation de l'entraînement en %	axe
\$AA_TORQUE[axe]						Valeur de consigne du couple d'entraînement en Nm	
\$AA_POWER[axe]						puissance active d'entraînement en W	
\$AA_CURR[axe]						valeur réelle du courant de l'axe de broche en A	
\$AA_SCTRACE[axe]	BOOL	R/W				écriture : 0 (FALSE): pas d'action 1 (TRUE): IPO-Trigger für Servo-Trace auslösen lecture : toujours 0 (car le bit déclencheur autoeffaçable est lu au retour de l'interface)	

Etats des axes

Nom	Type	Accès		Arrêt du prétraitement des blocs		Signification Plage de valeurs	Indice Plage de valeurs
		Prog. pièce	Action synchr.	Prog. pièce	Action synchr.		
\$AA_STAT[axe]	INT	R	R	R		Etat des axes 0: pas d'état à disposition 1: déplacement en cours 2: l'axe a atteint la fin de la période d'interpolation uniquement pour les axes du canal) 3: axe en position (arrêt précis grossier) pour tous les axes 4: axe en position (arrêt précis fin) pour tous les axes	axe
\$AA_TYP[axe]						Type d'axe 0: axe neutre 1: axe à interpolation 2: axe de positionnement programme pièce 3: axe de positionnement action synchrone 4: axe AP 5: axe dans un autre canal	
\$AA_FXS[axe]						Etat "Accostage d'une butée" 0: axe pas en butée 1: accostage réussi de la butée 2: échec de l'accostage de la butée	
\$AA_OFF_LIMIT[axe]						Valeur limite pour correction axiale \$AA_OFF ... 0: pas atteinte 1: atteinte dans sens positif de l'axe -1: atteinte dans sens négatif de l'axe	
\$AA_LEAD_TYP[VP]	REAL	R	R	R		Source de la valeur pilote 1: valeur réelle 2: valeur de consigne 3: valeur pilote simulée	axe pilote
\$AA_LEAD_SP[VP]						Position valeur pilote simulée	
\$AA_LEAD_SV[VP]						Vitesse valeur pilote simulée	
\$AA_LEAD_P[VP]						Position valeur pilote réelle	

15.6 Liste des variables système

Nom	Type	Accès		Arrêt du prétraitement des blocs		Signification Plage de valeurs	Indice Plage de valeurs
		Prog. pièce	Action synchr.	Prog. pièce	Action synchr.		
\$AA_LEAD_V[VP]		R	R	R		Vitesse valeur pilote réelle	
\$AA_SYNC [AA]	INT					Etat de couplage de l'axe asservi avec couplage par valeur pilote 0: pas de synchronisme 1: synchronisme grossier 2: synchronisme fin 3: grossier et fin	
\$AA_COUP_ACT[AA]	INT					Type de couplage 0: pas de couplage 3: positionn. tangentiel 4: broche synchrone 8: axe conjugué 16: axe asservi en couplage par val. pilote	

15.6.12 Données des broches

Nom	Type	Accès		Arrêt du prétraitement des blocs		Signification Plage de valeurs	Indice Plage de valeurs
		Prog. pièce	Action synchr.	Prog. pièce	Action synchr.		
\$AC_MSNUM	INT	R	R	R		Numéro de la broche maître	
\$AC_SDIR[n]						Sens de rotation broche activée actuellement 3: sens horaire 4: sens antihoraire 5: broche arrêt	n : numéro de broche 0: broche maître 0 - numéro max. de broche
\$AC_SMODE[n]						Mode broche activé actuellement 0 : pas de broche 1 : mode régulation de vitesse de rotation 2 : mode positionnement 3 : mode broche synchrone 4 : mode axe	
\$AA_S[n]	REAL					Vitesse de rotation réelle de broche; le signe correspond au sens de rotation	
\$AA_COUP_OFFS [axe]						Décalage de position (valeurs de consigne) des broches synchrones	Broches (S1, S2) - numéro max. de broche

Données d'entraînement broche

Nom	Type	Accès		Arrêt du prétraitement des blocs		Signification Plage de valeurs	Indice Plage de valeurs
		Prog. pièce	Action synchr.	Prog. pièce	Action synchr.		
\$VA_COUP_OFFS [axe]	REAL	R	R	R		Décalage de position (valeurs réelles) des broches synchrones	Broches (S1, S2) - numéro max. de broche
\$VA_COUP_ACT[axe]						État courant du couplage - S2 = broche asservie	

Données des broches

Nom	Type	Accès		Arrêt du prétraitement des blocs		Signification Plage de valeurs	Indice Plage de valeurs
		Prog. pièce	Action synchr.	Prog. pièce	Action synchr.		
\$XX_xxx[n]		R					n : numéro de broche
\$P_S[n]	REAL					Dernière vitesse de rotation de broche programmée :	
\$P_SDIR[n]	INT					Dernier sens de rotation de broche programmé 3 : sens horaire 4 : sens antihoraire 5 : arrêt broche	
\$P_SMODE[n]						Mode broche programmé en dernier 0 : pas de broche ou dans un autre canal ou broche AP 1 : mode régulation de vitesse de rotation 2 : mode positionnement 3 : mode broche synchrone 4 : mode axe	
\$P_GWPS[n]	BOOL					1 (TRUE): vitesse périphérique de la meule constante activée	

15.6.13 Variables système pour Safety Integrated

Nom	Type	Accès		Arrêt du prétraitement des blocs		Signification Plage de valeurs	Indice Plage de valeurs
		Prog. pièce	Action synchr.	Prog. pièce	Action synchr.		
\$VA_IS[axe]	REAL	R	R	R		position réelle sûre (Safety)	
\$A_XXX[n]	BOOL	R/W	R/W	R/W		E/S NCK pour logique combinatoire Safety Integrated. La variable est disponible uniquement en liaison avec Safety Integrated dans la commande.	N: numéro de l'entrée 1 - 64
\$A_INSI[n]						entrée	
\$A_OUTSI[n]						sortie	
\$A_INSE[n]						entrée externe	
\$A_OUTSE[n]						sortie externe	
\$AN_MEMENTO[n]						Mémento pour logique combinatoire Safety Integrated	

15.6.14 Valeurs programmées (synchrones au prétraitement des blocs)

Nom	Type	Accès		Arrêt du prétraitement des blocs		Signification Plage de valeurs	Indice Plage de valeurs
		Prog. pièce	Action synchr.	Prog. pièce	Action synchr.		
\$P_ACTID[n]	BOOL	R/W				1 (TRUE): action synchrone modale avec ID n activé	n : 1 – 16
\$P_AXN1	AXIS	R				adresse courante de l'axe géométrique - abscisse	
\$P_AXN2						adr. courante de l'axe géométrique – ordonnée	
\$P_AXN3						adr. courante de l'axe géométrique – cote	
\$P_ACTGEOAX						Relation courante entre axes géométrique, en fonction des plans. Affectation courante des axes géométriques programmés avec GEOAX(1,X, 2,Y,3,Z)	
\$P_DRYRUN						0 (FALSE) : Marche d'essai activée 1 (TRUE) : Marche d'essai désactivée	
\$P_SEARCH		R				1 (TRUE): la recherche de bloc (avec ou sans calcul) active	

Nom	Type	Accès		Arrêt du prétraitement des blocs		Signification Plage de valeurs	Indice Plage de valeurs
		Prog. pièce	Action synchr.	Prog. pièce	Action synchr.		
\$P_SEARCH1						1 (TRUE): la recherche de bloc avec calcul est active	
\$P_SEARCH2						1 (TRUE): la recherche de bloc sans calcul est active	
\$P_CTABDEF						1 (TRUE): la définition de la table de courbe est active	
\$PI	REAL	R				constante du cercle $\pi = 3.1415927$	
\$P_PROGPATH	STRING	R				appel d'un sous-programme à partir du répertoire courant PCALL \$P_PROGPATH <<_N_MYSUB_SPF	
\$P_MC	INT					0 (FALSE): pas d'appel modal d'un sous-programme 1 (TRUE): appel modal d'un sous-programme	
\$P_REPINF	INT					Info d'état pour repositionnement avec instruction REPOS 0 (FALSE): repositionnement pas possible, car - appel pas déposé dans un Asup - appel déposé par un Asup lancé à l'état reset - appel déposé par un Asup lancé en mode Jog 1 (TRUE): repositionnement possible	
\$P_SIM	BOOL					1 (TRUE): simulation en cours	
\$P_GG[n]	INT					act. Fonction G d'un groupe G Ind. comme interface AP	n : numéro du groupe G

15.6.15 Ordres vers/du canal

Nom	Type	Accès		Arrêt du prétraitement des blocs		Signification Plage de valeurs	Indice Plage de valeurs
		Prog. pièce	Action synchr.	Prog. pièce	Action synchr.		
\$P_MMCA	STRING	R/W		R/W		Acquittement de la tâche MMC	
\$A_PROTO	BOOL					Activer/Désactiver la fonction protocole	

15.6.16 Données d'outil

Nom	Type	Accès		Arrêt du prétraitement des blocs		Signification Plage de valeurs	Indice Plage de valeurs
		Prog. pièce	Action synchr.	Prog. pièce	Action synchr.		
\$P_AD[n]	REAL	R/W				corrections d'outil activées	n : numéro du paramètre 1 – 25
\$P_ATPG[n]						paramètres de rectification courants relatifs à l'outil	n : numéro du paramètre 1 – 9
\$P_TOOLxxx[t]		R					t : n° T 1 - SLMAXTOOLNUMBER
\$P_TOOL	INT					tranchant d'outil actif D0 – D9	
\$P_TOOLL[n]	REAL					longueur totale d'outil active:	n : longueur 1 – 3
\$P_TOOLNO	INT					numéro d'outil actif T0 – T32000	
\$P_TOOLR	REAL					rayon d'outil actif (total)	
\$P_TOOLND[t]	INT					nombre de tranchants de l'outil t	
\$P_TOOLEXIST[t]	BOOL					1 (TRUE): l'outil avec n° T existe	
\$P_TOOLMN[t]	INT					n° magasin de l'outil t	
\$P_TOOLMLN[t]	INT					n° magasin de l'outil t	
\$_VDITCP[n]	INT	R/W				Param. libres pour gestion outils dans interface VDI	n : 1 - 3

Annexe

A Index.....	A-515
B Commandes, descripteurs.....	A-521

A Index**A**

Accostage de positions codées 5-154
Actions synchrones
 statiques 9-288
Actions synchrones au déplacement
 Programmation 10-309
Activation du grignotage 12-384
Activation du poinçonnage 12-384
Affectations 1-35
Affecter et lancer une routine d'interruption 1-71
Affichage 2-110
Affichage du bloc courant 2-110
Afficheur max/min 14-418
Angle d'avance 7-215
Angle d'Euler 8-259
Angle latéral 7-215
Angle RPY 8-259
Appel d'un sous-programme avec indication de chemin 2-109
Appel d'un sous-programme, chemin de recherche 3-131
Appeler le frame 6-195
Apprentissage des courbes caractéristiques de compensation 13-397
Arrêt temporisé 1-68
Axe d'oscillation 11-367
Axe oblique, TRAANG 7-233
Axes de commande 10-344
Axes d'orientation 7-214
Axes indexés 5-154

B

Broche synchrone 13-399
 Activation du mode 13-405
 Désactivation du mode 13-405
 Instant de changement de bloc 13-404
 Rapport de transmission RT 13-403

 Type de couplage 13-404

 Variables système 1-21

Broches synchrones

 Effacement d'un couplage 13-406

 paire de 13-400

 paire de, définir 13-401

C

Calcul des données d'un cercle 14-422

Catalogue de pièces 3-129

Chariotage 14-414

Compactage 5-164

Compensation des défauts aux transitions entre quadrants

 Activer le processus d'apprentissage 13-398

 Désactiver l'apprentissage 13-398

 Réapprentissage 13-398

Concaténation de frames 6-196

Conditions marginales 1-62

Coordination de programmes 1-64

 Exemple 1-67

 Instructions de coordination de programmes 1-65

Correction de rayon d'outil, 3D

 Fraisage périphérique 8-254

 Orientation de l'outil 8-259

 Profondeur de pénétration ISD 8-258

 Programmation de l'orientation d'outil 8-259

Correction d'outil

 en ligne 8-246

 Fraisage en bout 8-252

Correction d'outil 3D

 Fraisage en bout 8-255

Correction d'outil, 3D

 Fraisage en bout 8-255

Correction d'outils

 Mémoire de correcteurs 8-240

Correction du rayon d'outil, 3D 8-252

 Angles rentrants / Angles saillants 8-258

- Comportement aux angles saillants 8-260
- Correction par commutateur 11-379
- Couplage d'axes à interpolation par valeur pilote 9-290
- Couplage d'axes à interpolation par valeur pilote
 - Type B 9-293
 - Type A 9-291
- Couplage par la valeur de consigne 13-399
- Couplage par la valeur réelle 13-399
- Courant
 - Décalage angulaire courant 13-407
 - Etat courant de couplage broche asservie 13-407
- cov.com, cycles utilisateur 2-120
- Créer une routine d'interruptionI comme sous-programme 1-70
- Cylindrage
 - usinage extérieur 14-415
 - usinage intérieur 14-415
- D
- Décalage d'origine
 - Décalage avec la manivelle 6-200
 - Décalage d'origine externe 6-201
 - Désactivation des transformations 6-203
 - PRESETON 6-202
- Décalage d'origine externe 6-201
- Décalage DRF 6-200
- décalage fin 6-199
- Décalage normal au contour OFFN 7-228
- Décalage Preset 6-202
- Définition de tableaux 1-27
- Définition de tableaux, listes de valeurs 1-29
- Définition des données utilisateur 3-135
- Définition des variables| 1-23
- Déplacement d'axe individuel 12-390
- Déplacements conjugués
 - Axes conjugués 9-272
 - Facteur de couplage 9-273
- Désactivation des frames 6-203
- Désactiver / Réactiver une routine d'interruption 1-72
- Détalonnage 14-415
- Distance partielle 12-388
- Distances partielles 12-388
- Données de réglage 11-368
- Données du porte-outil 15-496
- Dressage
 - usinage extérieur 14-415
 - usinage intérieur 14-415
- Durée d'exécution des programmes 1-61
- E
- Effacement de la distance restant à parcourir 11-371
- Effacement d'un couplage 13-406
- Elément de contour 14-418
- Eléments de détalonnage 14-416
- Exécution d'un élément de contour 14-421
- F
- Fin de programme 1-68
- Fonctions auxiliaires 12-388
- Fonctions H
 - Sorties rapides de fonctions, QU 13-410
- Fonctions OEM 5-186
- Fraisage en bout 7-217
- Fraisage périphérique 8-252
- Fraise
 - point auxiliaire (FH) 8-258
 - pointe (FS) 8-258
- Frottements 13-397
- G
- Gestion d'outils 8-243
- Grignotage 12-384
- Groupe spline 5-162
- I
- Indication de chemin
 - absolu 1-65
 - relatif 1-65
- Inhibition du bloc par bloc 2-111
- Instruction CASE 1-57

- Instruction de saut 1-53
- Instruction de saut
 - instruction CASE 1-57
- Instructions M 12-387
- Interpolation de type spline 5-156
 - Spline A 5-157
 - Spline B 5-158
 - Spline C 5-159
- Interpolations OEM 5-186
- Interpolation polynômiale
 - Polynôme dénominateur 5-171
- Intersection entre éléments de contour 14-414
- Inversion
 - Point d'inversion 11-373
 - point d'inversion de sens 11-373
 - Zone d'inversion 11-373
- ISD (Insertion Depth) 8-252

- J

- Jeux 13-397

- L

- Liste des instructions 15-426

- M
- Macroprogrammation 2-114
- Macroprogrammes 12-387
- MEAFRAME 6-204
- Mémoire
 - Mémoire de programmes 3-126
 - Structure de la mémoire 3-126
- Mémoire de programmes 3-126
 - Catalogue de pièces 3-129
 - Chemin de recherche pour l'appel d'un sous-programme 3-131
 - Création d'un répertoire pièce 3-130
 - Répertoires 3-128
 - Répertoires pièce 3-129
 - Sélection d'une pièce pour son usinage 3-131
 - Types de fichiers 3-128
 - Vue d'ensemble 3-127
- Mémoire de travail
 - Noms réservés pour blocs de données 3-135
 - Programmes d'initialisation 3-132
 - Zones de données 3-132
- Mesure avec palpeur à déclenchement
 - Programmation de blocs de mesure 5-174
 - Variable d'état 5-174

- N
- Niveaux de protection des données
 - utilisateur 3-138
- Nombre de pièces, défini 1-63

- O
- Opérateurs logiques 1-39
- Opérateurs relationnels et opérateurs logiques 1-38
 - Priorité des opérateurs 1-41
- Opérateurs/Fonctions de calcul 1-36
- Opérations sur les chaînes de caractères 1-43
- Orientation de l'outil 7-214; 8-259
- Oscillation
 - Activer, désactiver l'oscillation 11-369
 - commandée par action synchrone 11-373
 - Définition de la séquence des déplacements 11-370
 - Oscillation asynchrone 11-366
 - Oscillation asynchrone 11-368
 - Oscillation synchrone 11-372
- Oscillation asynchrone 11-366
- Oscillation synchrone
 - Actions synchrones 11-375
 - Affectation des axes d'oscillation et de pénétration 11-374
 - Arrêt de l'oscillation au point d'inversion 11-378
 - Définir les pénétrations 11-374
 - Pénétration dans la zone d'inversion de sens 11-376
- Outil de rectification, paramètres spécifiques 15-490

P

Paramètres de correction d'outil 15-489
Paramètres de description du magasin
 Gestion des outils 15-492
 Utilisateur OEM 15-492
Paramètres de magasin
 distance au poste de changement
 d'outil 15-492
 Hiérarchie des types d'emplacements 15-492
Paramètres de surveillance
 Gestion des outils 15-490
 Utilisateur OEM 15-490
Paramètres de tranchant Utilisateur OEM 15-489
Paramètres d'emplacement de magasin
 Gestion des outils 15-491
 Utilisateur OEM 15-491
Paramètres d'outil 15-489
Paramètres spécifiques à un outil
 Gestion des outils 15-490
 Utilisateur OEM 15-491
Paramètres spécifiques à un outil de
 rectification 15-490
Passe à lécher 11-371
Pénétration
 axe de 11-381
 mouvement de 11-376; 11-378
 supprimer 11-373
Pénétration partielle 11-373
Période d'appel de l'interpolateur 11-379
Permutation d'axe
 GET 1-77
 Prise en charge de l'axe 1-78
 RELEASE 1-77
Permutation d'axe
 Libération d'axe 1-77
Permutation de broche
 GET 1-77
 RELEASE 1-77
Poinçonnage 12-384; 12-388
Polynôme dénominateur 5-171

Position d'arrêt 13-405
Position d'activation 9-290
Positionnement tangentiel
 Angle de décalage par limitation de la zone de
 travail 9-268
 Définir l'axe asservi et les axes pilotes 9-267
Préparation du contour 14-415
 Éléments de détalonnage 14-416
Profondeur d'imbrication 1-61
Programmation de l'orientation d'outil
 avec LEAD et TILT 7-218
Programmation indirecte 1-33
Programme d'initialisation 3-132
 Chargement d'un programme
 d'initialisation 3-133
 Création d'un programme d'initialisation 3-133
 Sauvegarde des programmes
 d'initialisation 3-133
Programme sans fin 1-62

R

Rapport de transmission 13-403
Réaccostage du contour 9-302
 Accostage avec un nouvel outil 9-304
 Accostage sur un demi-cercle 9-306
 Accostage sur un quart de cercle 9-305
 Accostage sur une droite 9-305
 Point de réaccostage 9-303
Référence angulaire 13-405
Répertoires pièce 3-129
Routine d'interruption 1-69
 Définir les priorités 1-71
 Mémoriser la position d'interruption 1-70
 Retrait rapide du contour 1-73
 Sens de déplacement programmable 1-69

S

Segmentation de la distance à parcourir 12-390
Séquences de fonctions M 13-410
Signalisation d'erreur 14-415
Sorties de fonctions
 en contournage 13-410

- pendant les déplacements 13-410
 - Sorties rapides de fonctions, QU 13-410
 - Sous-programme
 - Appel d'un sous-programme 2-103
 - Mécanisme SAVE 2-98
 - Sous-programmes 2-96
 - Appel indirect d'un sous-programme 2-108
 - Appel modal d'un sous-programme 2-107
 - Imbrication 2-97
 - Répétition d'un programme 2-106
 - Sous-programme avec transfert de paramètres 2-103
 - Sous-programmes avec transfert de paramètres
 - Définition de tableaux 2-102
 - Transfert de paramètres entre programme principal et sous-programmes 2-99
 - Structures de contrôle 1-59
 - Surveillance d'outil, spécifique à la rectification 8-248
 - Synchronisme
 - fin 13-399
 - grossier 13-399
 - valeur de consigne du 13-399
 - Synchronisme de position 13-400
- T
- Table de contour 14-415
 - Tampon d'exécution 9-300
 - Temps d'arrêt 11-369
 - Torsion 13-397
 - Transformation avec axe linéaire pivotant 7-211
 - Transformation Axe oblique 7-232
 - Transformation de la surface latérale du cylindre 7-229
 - Décalage normal au contour OFFN 7-228
 - Transformation de surface latérale de cylindre 7-226
 - Transformation TRACYL 7-226
 - Transformation TRANSMIT 7-222
 - Transformation TRAORI 7-212
 - Transformation, 3 axes / 4 axes 7-212
 - Transformation, 5 axes
 - orientation de l'outil 7-214
 - Programmation du vecteur d'orientation 7-216
 - Programmation en angles d'Euler 7-215
 - Programmation en angles RPY 7-216
 - Transformation, 5 axes, fraisage en bout 7-217
 - Transformation, 5 axes, programmation avec LEADITILT 7-214
 - Type de couplage 13-399
- U
- uc.com, cycles utilisateur 2-121
- V
- Variable
 - Variables globales spécifiques au NCK 1-68
 - Variable frame 6-188
 - Affectation de valeurs 6-192
 - Appel de transformations de coordonnées 6-188
 - Définition de nouveaux frames 6-198
 - Lire et modifier des composantes d'un frame 6-194
 - Variable frame prédéfinie 6-189
 - Variables 1-20
 - Affectations 1-35
 - Conversion du type de variable 1-42
 - Définition de tableaux 1-27
 - Programmation indirecte 1-33
 - Types de variables 1-20; 1-21
 - Variables de calcul 1-21
 - Variables définies par l'utilisateur 1-23
 - Variables système 1-21
 - Variables système
 - Actions synchrones 15-496
 - Programmes pièce 15-489
- W
- WHEN-DO 11-375

Z	
Zones de protection 4-143	
Activation / Désactivation des zones de protection 4-148	
	Définition des zones de protection 4-145
	Définition du contour des zones de protection 4-146
	Zones de protection spécifiques à la machine 4-144
	Zones de protection spécifiques au canal 4-144

B Commandes, descripteurs

#

 α 7-232

\$

\$AA_COUP_ACT 9-273; 9-287; 13-407

\$AA_COUP_OFFS 13-407

\$AA_LEAD_SP 9-287

\$AA_LEAD_SV 9-287

\$MC_COMPRESS_VELO_TOL 9-298

\$SA_LEAD_TYPE 9-286; 9-287

\$VA_COUP_OFFS 13-407

A

ACC 13-402

ACTFRAME 6-190

ALF 1-69

applim 9-275

APR 3-138

APW 3-138

ASPLINE 5-156

AV 13-404

AX 13-396

AxeA 9-266; 9-271; 9-275; 9-284

AxeP 9-266; 9-271; 9-275; 9-284; 9-290

AXNAME 1-45; 13-396

AXSTRING 1-45

B

BA 13-399

BAUTO 5-160

BFRAME 6-189

BNAT 5-160

BP 13-399

BRISK 11-367

BSPLINE 5-156

BTAN 5-160

C

CAC 5-154

CACN 5-154

CACP 5-154

CALCDAT 14-414

CALL 2-108

CASE 1-57

CDC 5-154

CFINE 6-199

CHANADATA 3-134

CIC 5-154

CLEARM 1-66

CLRINT 1-69

CMIRROR 6-192

COARSE 13-399; 13-403; 13-404

COMPLETE 3-132; 3-133

COMPOF 5-165

COMPON 5-165; 9-298

CONTPRON 14-414; 14-415; 14-419; 14-421

COUP 13-401

COUPDEF 13-399; 13-401; 13-403

COUPDEL 13-399; 13-401; 13-406

COUPE 14-420

coupl. 9-266; 9-271

couplage 9-290

AV 13-399

DV 13-399

COUPOF 13-399; 13-405; 13-406

COUPON 13-399; 13-405; 13-406

COUPRES 13-399; 13-406

CPROT 4-148

CPROTDEF 4-144; 4-146

CROT 6-192

CSCALE 6-192

CSPLINE 5-156

CTAB 9-275

CTABDEF 9-275

CTABDEL 9-275

CTABEND 9-275

CTABINV 9-275

CTRANS 6-192

CUT3DC 8-252
CUT3DF 8-252
CUT3DFF 8-252
CUT3DFS 8-252

D

d 7-226
degré 9-275
DELT 8-243
DISABLE 1-69
DISPLOF 2-110
DRFOF 6-203
DV 13-404

E

EAUTO 5-160
ENABLE 1-69
ENAT 5-160
Endpos 11-372
ERREUR 14-415
ETAN 5-160
EXECTAB 14-414; 14-421
EXECUTE 4-144; 4-146; 14-415
EXTERN 2-103

F

FA 11-370; 13-402
FCTDEF 8-246
FCUB 9-295
FILETAGE 14-416
FINE 13-399; 13-404
FLIN 9-295
FNORM 9-295
FOR-ENDFOR 1-59
FPO 9-295
FTOCOF 8-246
FTOCON 8-246

G

G... 1-69
G1 11-367

G25,G26 9-268
G4 11-369
G64,G641 13-410
GEOAX 1-81
GET 1-77
GETD 1-77
GETSELT 8-243
GETT 8-243
GOTOB 1-57
GOTOF 1-57
Grignotage 12-388
GUD 3-128; 3-132; 3-136; 3-138
GUD 3-135

I

IF-ELSE-ENDIF 1-59
IFRAME 6-189
ii1,ii2 11-373
INDICE 1-48
INIT 1-65
INITIAL 3-133
INTERSEC 14-414; 14-419
IPOSTOP 13-399; 13-402; 13-404
ISAXIS 13-396
ISD 8-252; 8-258
ISNUMBER 1-45

K

KTAB 14-417

L

LEAD 7-215; 8-259
LEADOF 9-284
LEADOFP 9-290
LEADON 9-284
LEADONP 9-290
LIFTFAST 1-69
LOOP-ENDLOOP 1-59; 1-60

M

M17 2-99

M7 13-411
MATCH 1-48
MCALL 2-107
MEAC 5-176; 5-184
MEAFRAME 6-205
MEAS 5-173
MEASA 5-176
MEAW 5-173
MEAWA 5-176
MINDEX 1-48
mode 11-372
MU 7-234
MZ 7-234

N

n 9-275
N°_FRERE 8-243
NB 14-422
NBDETAL 14-417
NEWT 8-243
NN 14-415
NOC 13-399; 13-404
NOMTAB 14-415; 14-419; 14-421
NPROT 4-148
NPROTDEF 4-144; 4-146
NUMBER 1-45

O

OEMIPO1/2 5-186
OFFN 7-225; 7-226
ORIC 8-259
ORID 8-259
ORIMKS 7-219; 8-259
ORIS 8-259
ORIWKS 7-219; 8-259
OS 11-366; 11-369
OSC 8-259
OSCILL 11-372
OSCTRL 11-366; 11-370
OSE 11-366; 11-370
OSNSC 11-366
OSOF 8-259

OSP 11-366; 11-367
OSS 8-259
OSSE 8-259
OST 11-366; 11-369
OUTIL 8-243
OVRA 13-402

P

PCALL 2-109
PDELAYOF 12-384
PDELAYON 12-384
Pénétration partielle 11-372
PFRAME 6-189
POLY 5-169
POLYNOME 14-416
PON 12-384; 12-390
PONS 12-384
POS 13-405
POSP 11-372
PRESETON 6-202
PRIO 1-69
PROC 2-99
PT 14-422
PUTFTOC 8-246
PUTFTOCF 8-246

Q

QECDAT.MPF 13-398
QECLRN.SPF 13-398
QECLRNOF 13-397
QECLRNON 13-397
QECTEST.MPF 13-398
QFK 13-397
QU 13-410

R

RELEASE 1-77
REP 1-31
REPEAT-UNTIL 1-59
REPOS 1-69; 1-76
REPOSA 9-302
REPOSH 9-302

REPOSHA 9-302
REPOSL 1-76; 9-302
REPOSQ 9-302
REPOSQA 9-302
RES 14-422
RESET 9-290
RET 2-99
RINDEX 1-48
RPY 8-259

S

S1,S2 13-401; 13-406
SAVE 1-70; 2-98
SBLOF 2-111
SC 9-266
SCP 11-379
SET 1-29
SETINT 1-69
SETM 1-66
SETPIECE 8-243
SOFT 11-367
SON 12-384; 12-389; 12-390
SONS 12-384
SPI 13-396; 13-402
SPLINE 14-416
SPLINEPATH 5-162
SPN 12-388
SPOF 12-384
SPOS 13-402
SPP 12-388
START 1-65
STARTFIFO 9-300
STOPFIFO 9-300
STOPRE 5-173; 5-180; 5-182; 9-300; 11-368
STRLEN 1-48
SUBSTR 1-50
SUPA 6-203

T

TABC 14-421
TANG 9-266
TANGOF 9-266

TANGON 9-266
TILT 7-215; 8-259
TLIFT 9-266
TOWER 1-47
TOUPPER 1-47
TRAANG 7-226; 7-232
TRACYL 7-222; 7-226
TRAFOOF 7-210; 7-222; 7-226; 7-232
TRAILOF 9-271
TRAILON 9-271
TRANSMIT 7-222
TRAORI 7-212
TYPUSIN 14-415

U

U1,U2 11-373
UNTIL 1-61

V

VA 9-275
VAR 2-101
VARIB 14-419; 14-422
VP 9-275
VPapprox 9-275

W

WAITC 13-399; 13-402
WAITE 1-66
WAITM 1-65
WAITMC 1-66
WAITP 11-369
WALIMON 9-268
WHEN-DO 11-372
WHENEVER-DO 11-372; 11-375
WHILE-ENDWHILE 1-59

X

x 8-243
X... 1-69

SIEMENS AG
A&D MC V5
Postfach 3180
D-91050 Erlangen
République fédérale d'Allemagne

Propositions	
Corrections	
Imprimé :	
SINUMERIK 840D/810D/FM-NC Manuel de programmation Notions complémentaires	
Documentation utilisateur	
Expéditeur	
Nom :	N° de réf. : 6FC5298-4AB10-0DP1
Adresse de votre société/service	
Rue :	Edition : 12.97
Code postal : Localité :	Si, à la lecture de cet imprimé, vous deviez relever des fautes d'impression, nous vous serions très obligés de nous en faire part en vous servant de ce formulaire. Nous vous remercions également de toute suggestion et proposition d'amélioration.
Téléphone : /	
Télécopieur : /	

Propositions et/ou corrections

Herausgegeben von Siemens AG
Bereich Automatisierungstechnik
Geschäftsgebiet Automatisierungssysteme
für Werkzeugmaschinen, Roboter
und Sondermaschinen
Postfach 3180, D - 91050 Erlangen
Federal Republic of Germany

Geprüfte Siemens-Qualität für Software und Training
nach DIN ISO 9000, Reg. Nr. 2160-01.
Der Inhalt dieser Dokumentation wurde auf
umweltfreundlichem, chlorfrei gebleichtem Papier
gedruckt.
Copyright Siemens AG 1997 All Rights Reserved
Änderungen vorbehalten

Progress
in Automation.
Siemens