

## **UNIPROG+ version 1.02ND**

**Langage de programmation universel  
pour commandes E300-ND et E300-CMP**

Version: **6 juin 2007**

## Table des matières :

<b>1</b>	<b>Introduction .....</b>	<b>6</b>
1.1	Utilisation de ce manuel .....	6
<b>2</b>	<b>Organisation des zones de stockage .....</b>	<b>7</b>
2.1	Mémoires de la Commande E-300.....	7
<b>3</b>	<b>Géométrie et cinématique .....</b>	<b>8</b>
3.1	Position de référence.....	8
3.2	Changement de Référentiel.....	9
3.3	Constantes Cinématiques .....	10
3.3.1	<i>Le facteur d'échelle SCALEK.....</i>	<i>10</i>
3.3.2	<i>Fréquence maximale, Paramètre DIV.....</i>	<i>11</i>
3.3.3	<i>Constante d'accélération KUP, et de décélération KDN .....</i>	<i>11</i>
3.3.4	<i>Constante de Vitesse FEEDK.....</i>	<i>11</i>
3.3.5	<i>Course de l'axe, STROKE + / STROKE -.....</i>	<i>12</i>
3.3.6	<i>Sur-excitation du moteur : commande BOOST.....</i>	<i>12</i>
<b>4</b>	<b>Mode opératoire du clavier et fonctions de UNIPROG+ .....</b>	<b>13</b>
4.1	Mise sous tension de la commande E-300.....	13
4.2	Sélection des Menus .....	14
4.3	Menu "OTHER" .....	14
4.3.1	<i>Accès aux protections.....</i>	<i>14</i>
4.3.2	<i>Version.....</i>	<i>15</i>
4.4	Menu "CONFIGURATION".....	15
4.4.1	<i>Configuration des Générateurs de Mouvement (Motion GENERators) .....</i>	<i>15</i>
4.4.2	<i>Configuration de la Prise de Référence (REFerence).....</i>	<i>15</i>
4.4.3	<i>Configuration des Entrées de Contrôle (ConTRoL) .....</i>	<i>17</i>
4.5	Menu "MOTION CONTROL" .....	18
4.5.1	<i>Tableau d'outils "TOOL".....</i>	<i>18</i>
4.5.2	<i>Mouvements Manuels, JOG.....</i>	<i>18</i>
4.5.3	<i>Contrôle de fermeture, CLOS .....</i>	<i>19</i>
4.5.4	<i>Affichage des cotes et modes d'exécution, MODE .....</i>	<i>19</i>
4.6	Menu "PROGRAMMING".....	20
4.6.1	<i>Programmes à exécuter (start VECTors).....</i>	<i>20</i>
4.6.2	<i>Vitesses Présélectionnées (FEED rates) .....</i>	<i>20</i>
4.6.3	<i>Sauvegarde en mémoire FLASH (SAVE).....</i>	<i>20</i>
4.7	Manipulation de Fichiers, FILE UTILITIES .....	21
4.7.1	<i>Répertoire, (DIRectory).....</i>	<i>21</i>
4.7.2	<i>Effacement d'un Fichier (DELete).....</i>	<i>22</i>
4.7.3	<i>Copie d'un Fichier (COPY).....</i>	<i>22</i>
4.7.4	<i>Chargement depuis la mémoire FLASH (LOAD) .....</i>	<i>22</i>
4.8	Menu "DEBUGGING" .....	23
4.8.1	<i>Utilitaire "TRACE".....</i>	<i>23</i>
4.8.2	<i>Utilitaire "I/O" control.....</i>	<i>23</i>
<b>5</b>	<b>Instructions du langage UNIPROG+ .....</b>	<b>25</b>

5.1	Instructions de Positionnement .....	25
5.1.1	<i>Positionnement absolu</i> : .....	25
5.1.2	<i>Positionnement relatif</i> : .....	26
5.1.3	<i>Définition des outils</i> .....	26
5.2	Autres instructions Cinématiques .....	26
5.2.1	<i>Prise de référence</i> : .....	26
5.2.2	<i>Test de fermeture</i> : .....	26
5.2.3	<i>Instruction d'apprentissage</i> : .....	26
5.2.4	<i>Définition de paramètres</i> .....	27
5.2.5	<i>Cycle de perçage</i> .....	27
5.2.6	<i>Instruction de taraudage</i> .....	28
5.2.7	<i>Déplacement rectiligne</i> .....	28
5.2.8	<i>Angle</i> .....	29
5.2.9	<i>Rayon</i> .....	29
5.2.10	<i>Décalage angulaire</i> .....	29
5.3	Instructions Entrées/Sorties.....	29
5.3.1	<i>Attente sur entrée</i> : .....	29
5.3.2	<i>Branchement sur entrée</i> : .....	29
5.3.3	<i>Commande des sorties</i> : .....	29
5.3.4	<i>Inversion d'une sortie</i> .....	31
5.3.5	<i>Affichage d'une valeur</i> .....	31
5.4	Manipulation de Quantités Numériques .....	31
5.4.1	<i>Chargement de l'accumulateur</i> .....	31
5.4.2	<i>Dépose de l'accumulateur</i> .....	32
5.4.3	<i>Incrémentation/Décrémentation d'un pointeur</i> : .....	32
5.4.4	<i>Sauvegarde d'une variable en mémoire FLASH</i> : .....	32
5.4.5	<i>Vitesse de la broche</i> .....	32
5.4.6	<i>Vitesse de rotation des moteurs</i> .....	32
5.5	Instructions de Contrôle du Programme.....	33
5.5.1	<i>Saut inconditionnel</i> : .....	33
5.5.2	<i>Appel de sous-programme</i> : .....	33
5.5.3	<i>Fin de programme, de sous-programme</i> : .....	33
5.5.4	<i>Répétition</i> : .....	33
5.5.5	<i>Appel de programmes simultanés</i> : .....	34
5.5.6	<i>Branchements conditionnels (Test de l'accumulateur)</i> .....	34
5.6	Instructions de Temporisation .....	34
5.7	Instructions Arithmétiques .....	34
5.8	NOP et "Pseudo-Instructions" .....	34
5.9	Marqueur de Pause .....	35
<b>6</b>	<b>L'éditeur UNIPROG+ .....</b>	<b>36</b>
6.1	Examen d'un programme .....	36
6.2	Modification du Contenu d'une ligne .....	36
6.3	Insertion et Suppression d'une ligne .....	37
6.4	Mise en Place du Marqueur de Pause .....	37
<b>7</b>	<b>Exécution des programmes .....</b>	<b>38</b>
7.1	Fonctions des modes MOD1, MOD2 .....	38
7.2	Fonctions des touches START, STOP.....	38
7.3	Gestion des fautes.....	39

<b>8</b>	<b>Génération de vecteurs et contournage.....</b>	<b>40</b>
8.1	Introduction: Possibilités et Définition de l'Espace.....	40
8.2	Génération d'un Vecteur.....	40
8.3	Définition géométrique d'un Contour.....	40
	8.3.1 Définition d'un segment linéaire.....	41
	8.3.2 Définition d'un arc de cercle.....	42
8.4	Interprétation des Fichiers de Définition des Contours.....	44
8.5	Limitations de longueur pour l'interpolation linéaire.....	45
8.6	Exécution des Contours.....	45
8.7	Cas n'acceptant pas la correction d'outil.....	46
8.8	Affichage des erreurs de contour.....	46
8.9	Exemple de contournage.....	48
8.10	Résumé des Instructions et Pseudo-Instructions de Définition et de Génération de Contours 49	
<b>9</b>	<b>Récapitulation UNIPROG+.....</b>	<b>50</b>
9.1	Instructions.....	50
9.2	Entrées-Sorties (voir aussi 4.4.2).....	52
<b>10</b>	<b>Raccordement de la commande E300.....</b>	<b>53</b>
10.1	Commande compacte type E300-CMP.....	53
	10.1.1 Compatibilité avec E-600.....	53
	10.1.2 Connecteur I/O.....	53
	10.1.3 Connecteur I/O EXT.....	54
	10.1.4 Connecteur RS 232.....	54
	10.1.5 Module E-600-3, pour moteurs 2 phases, drivers EIP.....	54
	10.1.6 Connecteur ANALOG I/O.....	55

### Liste des figures :

Figure 3-1 : Courses et prise de référence.....	8
Figure 3-2 : Les référentiels UNIPROG+.....	10
Figure 3-3 : Fréquence ou vitesse en fonction de t.....	11
Figure 4-1 : Menu "I/O Control".....	23
Figure 5-1 : Cycle de perçage.....	28
Figure 8-1 : Contour avec diamètre nul.....	42
Figure 8-2 : Modes de rotation.....	43
Figure 8-3 : Exemple de contour.....	44
Figure 8-4 : Position d'attente selon le sens de déplacement.....	46

## Liste des tableaux :

Tableau 5-1 : Entrées/Sorties UNIPROG+ .....	30
Tableau 5-2 : Adresse des modules IN/OUT .....	30
Tableau 10-1 : Comparaison I/O E300 et E600 .....	53
Tableau 10-2 : E300, connecteur I/O, Burndy 19 broches .....	54
Tableau 10-3 : E600-3, connecteur, Burndy 8 broches.....	54
Tableau 10-4 : E600-3, réglage du courant.....	55
Tableau 10-5 : Connecteur ANALOG I/O .....	55

# 1 Introduction

La commande de positionnement et de contournage de la série E-300 est destinée à l'équipement de manipulateurs et de petites machines spéciales et résout les problèmes de cinématique et d'automatisme les plus variés.

- Elle est actuellement disponible en version 1 ou 2 axes pour moteurs pas-à-pas 2 phases.
- E.I.P. SA a créé le langage de programmation PINX-E et l'outil de développement APEX afin de répondre de façon efficace aux situations les plus variées et les plus exigeantes. La pratique a montré que nombre de fonctions se retrouvent dans la plupart des applications mais que les séquences de travail sont infiniment variables. D'autre part, la maîtrise du langage PINX-E demande un effort certain et l'utilisation d'un PC comme outil de développement n'est pas souhaitable dans un environnement industriel.
- Ces considérations ont conduit E.I.P. au programme **UNIPROG+** qui offre un ensemble de fonctions utilitaires et la possibilité d'écrire des programmes performants simplement à partir du panneau de la commande.

## 1.1 Utilisation de ce manuel

Ce manuel doit permettre à un utilisateur sans expérience en informatique de maîtriser UNIPROG+ après une lecture attentive. Quelques connaissances des entraînements par moteurs pas-à-pas et particulièrement des systèmes asservis sont nécessaires pour éviter des tâtonnements laborieux. Par contre, il n'est pas utile de connaître la programmation des automates programmables.

- Le lecteur aura intérêt à bien étudier les chapitres 2 et 3 avant d'essayer d'écrire un programme ou d'utiliser les fonctions utilitaires.
- Le chapitre 4 décrit le mode opératoire du clavier à partir de la mise sous tension.
- Le chapitre 8 est entièrement dédié au problème de la génération de vecteurs et du contournage (interpolation linéaire et circulaire).

## 2 Organisation des zones de stockage

### 2.1 Mémoires de la Commande E-300

Les programmes écrits par l'utilisateur sont mémorisés dans la mémoire vive, sauvegardée par batterie, de la commande E-300. L'utilitaire SAVE permet de copier toute la zone utilisateur de la mémoire vive vers de la mémoire FLASH.

A l'intérieur de la zone mémoire utilisateur, un espace est réservé aux paramètres de configuration (voir paragraphe 4.4). Le reste de la mémoire est utilisable pour des programmes.

L'éditeur UNIPROG travaille par lignes. Une instruction ou une donnée numérique est toujours mémorisé sous forme d'une ligne.

- 1250 lignes sont disponibles.
- 100 fichiers peuvent être créés. Ils sont numérotés de 00 à 99.

### 3 Géométrie et cinématique

Ce chapitre fixe le système de référentiels utilisé par UNIPROG+ et donne les informations utiles sur les générateurs de mouvements de la commande E-300.

#### 3.1 Position de référence

L'entraînement à moteur pas-à-pas ou asservi étant incrémentiel par nature, tout cycle de travail doit débuter par la fixation des positions de référence des axes. Deux cas pratiques se présentent :

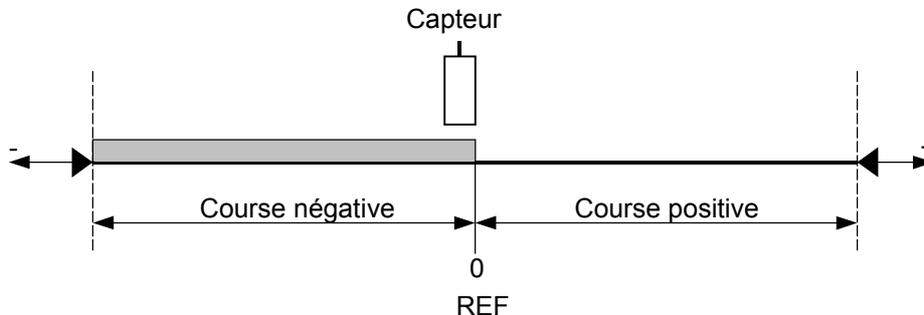
- a) Le référentiel est lié au bâti de la machine, c'est le cas d'une table de palettisation, par exemple.
- b) Le référentiel est fixé par l'opérateur en un point de la course et à un instant choisi par lui, c'est le cas d'un appareil à diviser.

Dans le cas a), l'axe doit être muni d'un contact de référence qui permet la fixation automatique et précise du **référentiel initial**.

La prise de la position de référence peut se faire sur ordre manuel ou être intégrée à un programme d'initialisation.

UNIPROG+ permet de placer le contact de référence n'importe où dans la course.

**Le contact de référence doit être actif en permanence sur l'une des courses afin de signaler à la commande de quel côté se situe la coulisse (voir Figure 3-1)**



**Figure 3-1 : Courses et prise de référence**

UNIPROG+ effectue la prise de référence en trois phases :

- Phase 1 : Cette phase de la prise de référence est exécutée uniquement si le contact est déjà actif : Sortie du contact de référence à une vitesse configurée avec décélération configurée à la sortie du contact.
- Phase 2 : Déplacement vers le contact de référence à une vitesse configurée avec décélération sur contact.
- Phase 3 : Déplacement en sens opposé à vitesse réduite (fraction configurée de la vitesse de prise de référence) et arrêt instantané dès que le contact n'est plus actionné.

- La phase 3 donne la précision de la position de référence, variable en fonction de la vitesse de retour configurée. Il peut être intéressant de noter le sens de déplacement de la phase 2 : l'utilisateur sait ainsi dans quel sens sont pris en compte les éventuels jeux et hystérésis de la transmission.
- La commande E-300 offre 2 entrées par axe pouvant fonctionner comme signal de référence, **INA** et **INB**.
- Lorsqu'un axe n'utilise pas de contact de référence, l'origine sera fixée par l'exécution d'une prise de référence sans aucun mouvement.
- La course (STROKE +, resp. STROKE -) introduite par le menu de configuration délimite la course positive, resp. négative de 0 à STROKE +, resp. STROKE -, dans laquelle se déplacent les axes.
- Si la course est uniquement positive (STROKE - = 0), seuls les points dont les coordonnées dans le référentiel initial sont positives seront accessibles, voir Figure 3-1. Le sens de prise de référence tient compte du signe de la vitesse de prise de référence introduite par le menu de configuration.
- Lorsqu'une limite de course n'est pas souhaitée sur un axe, par exemple dans la commande d'un plateau-diviseur, il suffit de configurer une course nulle. Dans ce cas, aucune entrée de référence ne doit être attribuée à cet axe.

## 3.2 Changement de Référentiel

Les mouvements à effectuer sont donnés en valeur "relative" ou en coordonnée "absolue". La notion de mouvement relatif et absolu est applicable tant aux déplacements commandés manuellement dans le menu "JOGGING" qu'aux instructions de mouvement.

Dans un mouvement relatif, le déplacement est relatif au point où se trouve l'axe avant le mouvement. La notion de référentiel est évidemment sans objet.

Dans un mouvement absolu, c'est la valeur des coordonnées du point à atteindre qui est donnée. Le référentiel dans lequel sont mesurées les coordonnées prend alors tout son sens.

Le **référentiel courant** est introduit par une instruction **TOOL**. Le programmeur peut changer le référentiel courant aussi souvent qu'il en éprouve le besoin. Si aucune instruction TOOL n'est exécutée, le référentiel courant est confondu avec le référentiel de base. Le changement de référentiel courant est utile dans les situations telles que les machines à outils multiples ou les manipulateurs dans lesquels on distingue un espace de prise et un espace de dépose.

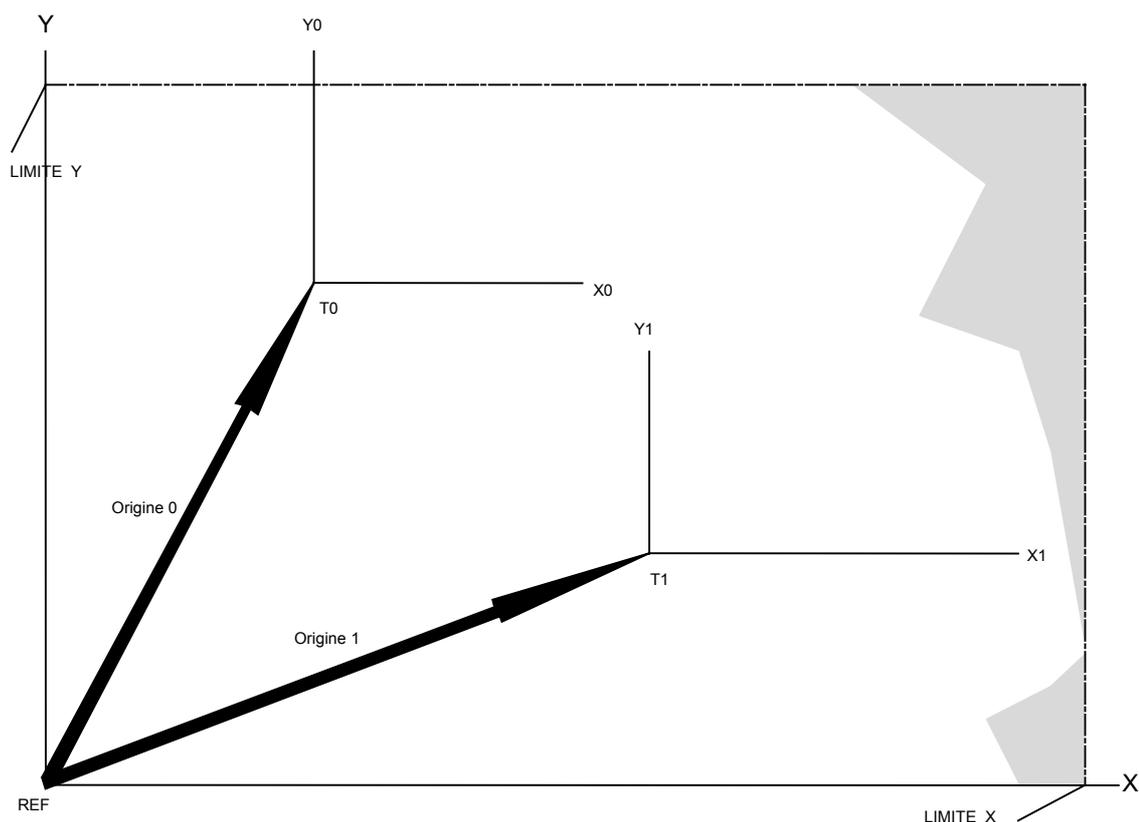


Figure 3-2 : Les référentiels UNIPROG+

### 3.3 Constantes Cinématiques

Les constantes discutées dans ce paragraphe seront introduites par le menu de configuration. Ces constantes sont propres à chaque axe.

#### 3.3.1 Le facteur d'échelle SCALEK

Ce facteur permet de programmer et d'afficher les cotes et les déplacements en unités techniques.

Dans le cas des moteurs pas-à-pas, SCALEK est le nombre d'impulsions requis à l'entrée de l'étage de puissance pour produire un déplacement de une unité.

Les étages de puissance E.I.P. pour moteurs 2 phases, opèrent toujours en mode micro-pas avec une division par 8 du pas entier. Avec des moteurs courants (200 pas par tour) il faut donc 1600 impulsions par tour de moteur.

##### Exemples: coulisse à vis

Moteur 2 phases, 1,8 deg. par pas, transmission par courroie de rapport 1:2, vis au pas de 5 mm, unité: 1 mm:

1600 impulsions pour 1 tour du moteur,  
3200 impulsions pour 1 tour de la vis,

Donc  $3200/5 = 640$  impulsions par mm, **SCALEK = 640**.

### 3.3.2 Fréquence maximale, Paramètre DIV

Lors de l'exécution d'un mouvement, la fréquence de l'axe (ou la vitesse de l'axe) a l'allure reproduite à la Figure 3-3. La grandeur de l'accélération ou de la décélération décroît linéairement en fonction de la vitesse pour prendre en compte la baisse du couple du moteur aux vitesses élevées. La fréquence maximale doit être déterminée expérimentalement pour chaque axe de façon à conserver une marge de couple suffisante aux plus hautes vitesses.

Dans tous les cas de fonctionnement, la vitesse sera inférieure ou égale à la vitesse maximale résultant de DIV. Lors de l'accélération, le générateur de mouvement tronque la courbe de vitesse à la vitesse palier prescrite. Les pentes d'accélération et de décélération sont conservées.

Si  $f_{max}$  est la fréquence maximale des impulsions à l'entrée de l'étage de puissance du moteur pas-à-pas, le paramètre DIV est donné par :

$$\text{DIV} = 11906/f_{max} \text{ [kHz]}$$

Avec DIV = 120 on obtient approximativement 100 kHz, avec 300, 40 kHz.

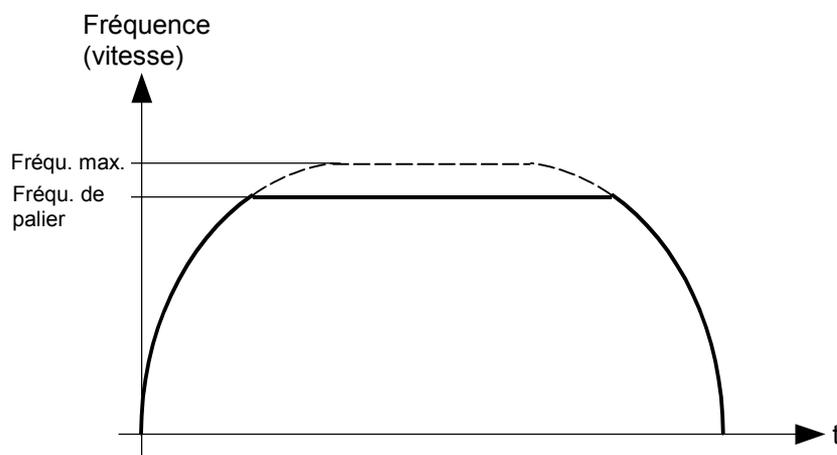


Figure 3-3 : Fréquence ou vitesse en fonction de t

### 3.3.3 Constante d'accélération KUP, et de décélération KDN

Ces paramètres fixent la pente initiale (respectivement finale) des rampes. Ils s'expriment en kHz/s ou en kimpulsions/s<sup>2</sup>. Comme au paragraphe précédent, il s'agit des impulsions à l'entrée de l'étage de puissance.

Des valeurs de 200 à 2000 kHz/s sont en général satisfaisantes.

### 3.3.4 Constante de Vitesse FEEDK

La constante de vitesse (ou constante d'avance) a pour but l'expression des vitesses en unités techniques: mm/s, m/min, t/min etc.

Pour les entraînements pas-à-pas :

$$\text{FEEDK} = \text{fréquence des impulsions pour une unité de vitesse [kHz]}$$

**Exemple:** (Voir paragraphe 3.3.1.)

On veut exprimer la vitesse en m/min.

SCALEK = 640, soit 640 impulsions pour 1 mm

ou encore 640'000 impulsions pour 1 m,

ou encore une fréquence de 640 kHz pour 1 m/s,

soit 640/60 kHz pour 1 m/min, **FEEDK = 10,667**

La vitesse effectivement atteinte ne saurait dépasser la vitesse maximale fixée par DIV. Exprimée dans l'unité choisie, la plus grande vitesse réalisable est donc :

$$\boxed{11906/DIV/FEEDK}$$

**N.B.** La vitesse d'avance ne correspond au calcul que si le potentiomètre est tourné à fond dans le sens horaire. Le potentiomètre est en jeu si son témoin est allumé. Il est en jeu dans tous les mouvements à l'exception de la prise de référence et du test de fermeture.

### 3.3.5 Course de l'axe, STROKE + / STROKE –

Ces paramètres, exprimés dans l'unité de longueur choisie au paragraphe 3.3.1, limite la course dans le sens positif et négatif de l'axe dans les mouvements programmés et les mouvements manuels.

### 3.3.6 Sur-excitation du moteur : commande BOOST

Ce paramètre définit l'action du signal /BOOST et ne peut prendre que l'une des valeurs 0, 1, 2, 3.

BOOST = 0 : Le signal /BOOST est toujours inactif (haut).

BOOST = 1 : Le signal /BOOST est "bas" pendant les mouvements, "haut" en dehors des mouvements de l'axe

BOOST = 2 : Le signal /BOOST est toujours actif (bas).

BOOST = 3 : Le signal /BOOST est "haut" pendant les mouvements, "bas" en dehors des mouvements de l'axe.

Avec les translateurs E-300, on utilise normalement BOOST = 1. De cette façon, le courant est nominal pendant le mouvement et il est réduit à 60 % au repos.

On peut travailler avec BOOST = 0 pour de petits moteurs ou alors avec BOOST = 2 si un couple de maintien maximal est requis à l'arrêt.

## 4 Mode opératoire du clavier et fonctions de UNIPROG+.

Ce chapitre décrit l'utilisation de la commande E-300 sous le programme UNIPROG+. La description part de la mise sous tension et suppose le raccordement effectué. Les menus ne sont pas abordés dans l'ordre d'affichage, mais bien dans l'ordre chronologique requis pour une première mise en service. La programmation, l'usage de l'éditeur, l'exécution des programmes et leur test font l'objet des chapitres suivants.

### 4.1 Mise sous tension de la commande E-300.

Le message suivant apparaît indiquant quelles versions sont contenues dans la commande.

```
* UNIPROG+ V x.xx *  
E-300 x.xx          mm.dd.yy
```

Après le chargement du logiciel UNIPROG+ le contrôleur E-300 affiche :

```
ERREUR RAM  
FORMATTER  OUI  NON
```

**Répondre NO** après le chargement afin de configurer le paramètre "LAST TOOL NB" et d'éviter la perte du fichier "0".

- En répondant NO le paramètre "LAST TOOL NB" ne sera pas utilisé pour l'ouverture du fichier 0 et le contrôle de perte de la mémoire vive ne sera pas mis à jour.
- En répondant YES le fichier 0 est ouvert et son contenu écrasé. Le contrôle de la RAM est mis à jour.

Après cela, UNIPROG+ exécute directement le programme d'initialisation et présente le premier menu:

```
1 MOTION CONTROL  
TOOL JOG  CLOS DISP
```

Pendant l'exécution du programme d'initialisation (et de tout autre programme) l'opérateur peut utiliser les menus et accéder aux utilitaires. A ce stade de notre mise en route, mentionnons la possibilité d'annuler le programme en exécution par la touche STOP.

## 4.2 Sélection des Menus

Pour choisir un menu, il faut l'amener à l'affichage à l'aide des touches ↓ ou ↑. Les menus suivants défilent:

1 MOTION CONTROL TOOL JOG CLOS MODE
--

2 PROGRAMMING EDIT VECT FEED SAVE
--------------------------------------

3 DEBUGGING TRACE I/O
--------------------------

4 FILE UTILITIES DIR DEL COPY LOAD
---------------------------------------

5 CONFIGURATION MGEN REF CTRL
----------------------------------

6 OTHER VER COUNT ACCES
----------------------------

Si le menu n'offre qu'une option, on y entre par la touche ENTER, si plusieurs possibilités sont offertes, 4 dans le menu "CONFIGURATION" par exemple, on choisit l'option par la touche de fonction F1 à F4. **La touche ESC permet toujours de sortir d'une fonction.**

Si le message :

NON ACCESSIBLE Presser une touche
--------------------------------------

apparaît lorsqu'on tente d'entrer dans une fonction, cette fonction est protégée par le code d'accès, voir paragraphe 4.3.1.

## 4.3 Menu "OTHER"

### 4.3.1 Accès aux protections

Afin de donner un accès sélectif aux différentes fonctions d'UNIPROG+, des drapeaux d'accès peuvent être attribués individuellement. On peut, par exemple, permettre à l'utilisateur final d'accéder aux mouvements manuels mais pas à l'éditeur.

Quelque soit l'état des drapeaux d'accès, l'introduction de la clef numérique donne l'accès

général. A la mise sous tension de la commande, l'accès aux fonctions protégées est toujours fermé.

Pour obtenir l'accès général à toutes les fonctions, amener le menu ACCESS. Le message "ENTER ACCESS CODE" invite à entrer la clef d'accès qui est le nombre

**31415**

(En général, toute entrée de nombre se termine par une pression sur ENTER et pendant l'introduction d'un nombre on peut corriger les erreurs de frappe par la touche CLR.)

Une pression sur ESC permet de retourner aux menus; toutes les fonctions sont maintenant accessibles.

L'introduction de la clef lorsque l'accès général est établi referme les fonctions protégées.

Pour placer les drapeaux d'accès individuels, rester dans le menu "ACCESS" après avoir introduit la clef. Les fonctions ou groupes de fonctions apparaissent à l'affichage. Comme dans tous les menus, les flèches permettent de parcourir la liste. L'introduction d'un "1" donne l'accès, un "0" ne donne l'accès à la fonction qu'à l'aide de la clef.

### 4.3.2 Version

Tant que la pression sur la touche F1 est maintenue, l'écran présente le nom et les numéros des différentes versions installées.

Ces informations sont importantes à relever lors d'éventuelles questions à notre service technique.

## 4.4 Menu "CONFIGURATION"

### 4.4.1 Configuration des Générateurs de Mouvement (Motion GENERators)

La touche F2 introduit ce sous-menu à partir du menu CONFIGURATION. Dans ce cas, les paramètres à configurer se présentent sous la forme d'un tableau à 2 dimensions : les paramètres se parcourent avec les touches flèches et l'axe auquel est destiné le paramètre se choisit avec les touches de sélection d'axes X, Y.

Tous les paramètres de ce menu ont été discutés au chapitre 3. Pour mémoire, rappelons leur signification:

<b>DIV</b>	Diviseur de fréquence
<b>KUP</b>	Constante d'accélération
<b>KDN</b>	Constante de décélération
<b>SCALEK</b>	Facteur d'échelle pour la longueur
<b>FEEDK</b>	Facteur d'échelle pour la vitesse
<b>STROKE +</b>	Course de l'axe positif
<b>STROKE -</b>	Course de l'axe négatif
<b>BOOST</b>	Surexcitation (voir description des modules de puissance)

### 4.4.2 Configuration de la Prise de Référence (REFerence)

La touche F3 introduit le menu de configuration du dispositif de prise de référence. Il s'agit comme ci-dessus d'un tableau à deux entrées.

<b>REF INPUT NUMBER</b>	Numéro de l'entrée pour prise de référence (voir tableau). La prise de référence peut être inhibée en spécifiant le numéro 8. Dans ce cas, lors de l'appel de la fonction de prise de référence, le compteur de position est remis à 0 sans qu'il n'y ait de mouvement.
<b>SPEED TO REF</b>	Vitesse pour prise de référence. Cette vitesse est donnée dans l'unité définie par FEEDK. Une vitesse négative, inverse le sens de prise de référence.
<b>CLOSURE GAP</b>	Tolérance de fermeture, voir paragraphe 4.5.3.
<b>SWITCH:</b>	Nature du contact ou du détecteur de référence. Entrer "1" pour un contact normalement ouvert, "0" pour un contact normalement fermé, " <b>1</b> " pour un moteur pas-à-pas avec index.
<b>REF SPEED BACK</b>	Divise la vitesse de prise de référence pour quitter le contact, plus le diviseur est grand, plus grande sera la précision.

Paramètre (REF INPUT)	Entrée physique	Remarque
0..7	IN 0..7	
8	-	Mise à 0 sans mouvement
9	Signal FAULT de l'étage de puissance	
18	INA issu du E-600-18 pour Yaskawa	Pour Yaskawa et en cas de switch commun limite-référence
60	INA 0	
61	INB 0	
62	INA 1	
63	INB 1	

***Paramètre 'REF INPUT'***

### 4.4.3 Configuration des Entrées de Contrôle (ConTRoL)

Ce sous-menu introduit par F4 permet de fixer l'attribution d'entrées aux fonctions de contrôle de l'exécution du programme, en parallèle avec les touches de la face avant, voir le chapitre 7 et le paragraphe 5.3.

On introduit le numéro de l'entrée attribuée. Si la fonction n'est pas requise, entrer 64.

<b>EXTERNAL START</b>	Départ du programme, contact normalement ouvert
<b>EXTERNAL PAUSE</b>	Suspensions de l'exécution et des mouvements, contact normalement fermé.
<b>EXTERNAL STOP</b>	Avorte les programmes en cours. Contact normalement fermé (élément d'une chaîne de sécurité).

Ce sous-menu contient également les paramètres suivants :

<b>DISPLAY FORMAT 1-6.</b>	Ce sous-menu contient également la définition du format de l'affichage des cotes; le nombre introduit fixe le nombre de chiffres à droite du point décimal
<b>2 HANDS START</b>	Permet de configurer le start à deux mains sur 2 entrées choisies sous EXTERNAL START et 2 HANDS START. (Valeur de 0 à 8). <ul style="list-style-type: none"><li>• Dans ce cas, seules les entrées 0 à 7 sont utilisables pour ces deux paramètres.</li><li>• La valeur 8 introduite dans le paramètre 2 HANDS START rétablit le start simple.</li></ul>
<b>LAST TOOL NB</b>	(Dernier numéro d'outil) permet de limiter le nombre d'outils utilisés, afin de ne pas remplir la mémoire de programme inutilement. (Valeur de 0 à 63).
<b>MAX RPM 10 volts</b>	Permet de configurer la vitesse de la broche en fonction de la tension du DAC. Entrer la vitesse de la broche en t/min qui doit correspondre à une valeur DAC de 10V (cette valeur peut varier entre 0 et 99999).
<b>FEED CTRL BY ADC</b>	Si ce paramètre vaut 0, la vitesse des moteurs pas à pas est contrôlée par l'entrée ADC. Si ce paramètre vaut 1, la vitesse est contrôlée par le potentiomètre.
<b>LAST DELAYED OUT</b>	Permet de configurer un nombre de sorties dont la remise à zéro est retardée. (Valeur de 0 à 7). <ul style="list-style-type: none"><li>• Le paramètre est compris entre 0 et 7, il donne le nombre total de sorties retardées à partir de 0.</li><li>• Avec 0 aucune sortie n'est retardée. Avec 7 les sorties 0,1,2,3,4,5,6 et 7 sont retardées.</li><li>• La valeur du retard par défaut vaut 1 s, cette valeur pouvant être modifiée dans le programme (voir SET BRK_D, au paragraphe 5.2.4).</li></ul>
<b>TOOL L/R INVERT</b>	Permet de rétablir la correspondance du coté de départ de l'outil lors du contournage avec correction de trajectoire. Avec la valeur "0", le sens des axes est considéré normal, le référentiel du plan est positif vers la droite et vers le haut. Entrer "1" pour le cas contraire.
<b>LANGUAGE</b>	Permet d'afficher certain message dans les langues suivantes: 0 = Anglais, 1 = Français, 2 = Allemand. (Valeur de 0 à 2).

## 4.5 Menu "MOTION CONTROL"

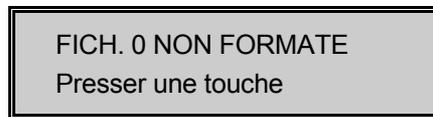
Ce menu introduit 4 fonctions ou sous-menus permettant d'effectuer des mouvements sous contrôle manuel et d'afficher les cotes des axes.

Les fonctions TOOL (Tableau d'outils), JOG (Jogging) et CLOS (Closure check) ne sont pas accessibles pendant l'exécution d'un programme.

### 4.5.1 Tableau d'outils "TOOL".

Dans ce menu, les origines et le diamètre des outils sont accessibles. A chaque outil sont associés 4 origines et un diamètre pour la correction de la trajectoire d'outil.

Les valeurs visibles dans ce tableau d'outils sont déposées dans le fichier "0" qui est ouvert automatiquement lorsque l'on configure le paramètre "LAST TOOL NB". Si par le fait d'une manipulation erronée, le fichier "0" ne présente pas la taille adaptée, le message suivant apparaît :



Dans ce cas, le paramètre "LAST TOOL NB" doit être reconfiguré.

### 4.5.2 Mouvements Manuels, JOG

Dans ce sous-menu, la ligne supérieure de l'affichage montre l'axe sélectionné et sa cote dans le référentiel de base, la ligne inférieure indique l'outil courant et l'incrément en vigueur lors de la pression des touches "flèches".

- **Action des touches flèches:**

Une pression sur l'une de ces touches effectue un mouvement de la longueur affichée pour autant que la touche soit maintenue pendant tout le mouvement. Le relâchement de la touche, stoppe le mouvement avec une décélération gouvernée par KDN. Une nouvelle pression sur les touches JOG+/- produit à nouveau un incrément complet.

- **Action des touches F3 et F4 :**

Ces touches permettent de choisir la valeur de l'incrément de mouvement.

- **Action de la touche F8:**

Elle permet de se déplacer entre la ligne supérieure et inférieure.

Si le curseur est sur la ligne inférieure, une valeur d'incrément peut être entrée avec le pavé numérique, au lieu des touches F3 et F4.

Si le curseur est sur la ligne supérieure, une coordonnée de destination peut être entrée au moyen du pavé numérique. Le mouvement est lancé à la pression de la touche ENTER. Cette touche doit rester pressée durant tout le mouvement.

**Le menu JOG permet en plus d'enregistrer les origines des outils.** La sélection de l'outil se fait au moyen des touches F1 et F2.

L'enregistrement d'une origine se fait de la manière suivante:

- Atteindre la position voulue en Jogging.
- Sélectionner l'outil et l'axe concerné.
- Une pression de la touche F5 autorise l'entrée d'une nouvelle valeur d'origine pour l'outil et l'axe sélectionné.
- Entrée de la valeur au pavé numérique puis validation de cette dernière en pressant la touche ENTER.

Les nouvelles origines sont stockées dans le fichier 0, qui contient des "FDATA". Pour consulter ces origines plus facilement, aller dans le menu TOOL du groupe MOTION CONTROL. Des modifications peuvent également être faites depuis ce menu.

Pour de petits ajustements, la correction d'origine d'outil peut être entrée de façon incrémentale avec les touches "+" et "-" du jogging. La valeur, affichée dans l'angle inférieur droit (l'incrément), sera soustraite ou additionnée à l'origine. Cela sans générer de déplacement.

Dans l'angle inférieur gauche l'affichage indique "ALL". Si F1 est pressé la correction d'origine s'applique sur tous les outils, La sélection est confirmée par la LED F1 clignotante et par le texte en bas à gauche "ADJ ALL" (ajuste tout). Dans cette situation chaque pression sur une touche "flèche" décale l'origine de tous les outils.

Une pression de la touche F7 (REF), provoque une prise de référence. A la mise sous tension aucun mouvement ne peut s'effectuer si la prise de référence n'a pas été faite. Toutefois les mouvements avec le jogging sont possibles.

### 4.5.3 Contrôle de fermeture, CLOS

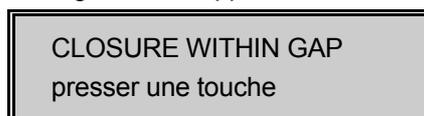
L'entraînement à moteur pas-à-pas se fait en boucle ouverte (sans feed-back). Il est souvent souhaitable de détecter les éventuelles anomalies de fonctionnement conduisant à des erreurs de position en vérifiant périodiquement que le polygone parcouru par un axe se referme dans une tolérance donnée.

Le contrôle de fermeture se fait naturellement autour du contact de référence et la tolérance est introduite sous "CLOSURE GAP". Ce paramètre est exprimé dans l'unité de l'axe et le contrôle est réputé bon si la fermeture a lieu dans une bande de  $\pm$  GAP autour du contact.

La mesure s'effectue dans les sens de la prise de référence pour éliminer l'influence des jeux et hystérésis. Si le contrôle détecte une erreur de position, le programme est stoppé et l'erreur est affichée. Dans tous les cas, l'axe se trouve à la position de référence après le contrôle de fermeture.

Presser la touche F3 pour accéder à ce sous-menu. Après avoir choisi l'axe, il suffit de presser la touche REF. Le résultat du contrôle s'affiche à l'écran:

Si le contrôle est bon, le message suivant apparaît :



CLOSURE WITHIN GAP  
presser une touche

dans le cas contraire :



POS ERROR 1.234  
presser une touche

Si l'axe n'a pas de contact de référence attribué, le contrôle réagit comme s'il était bon. Rappelons qu'après un contrôle de fermeture, l'axe se trouve à son point de référence.

### 4.5.4 Affichage des cotes et modes d'exécution, MODE

#### Affichage des cotes:

L'affichage des cotes se fait sur la ligne supérieure. Contrairement aux sous-menus TOOL, JOG, CLOS qui ne peuvent pas être appelés durant l'exécution d'un programme, l'affichage de la position des 2 axes est toujours possible. Le format de l'affichage a été fixé au paragraphe 4.4.3. **Les cotes sont mesurées dans le référentiel courant.**

### Choix du mode d'exécution:

Le choix du mode d'exécution se fait au moyen des touches F1 à F3 et le mode courant est affiché sur la 2<sup>ème</sup> ligne.

- F1** Choix du mode SAT, saturation des vitesses de déplacement des axes.
- F2** Choix du mode MOD1, exécution normale, les marqueurs de pause ne sont pas pris en compte.
- F3** Choix du mode MOD2, le programme s'arrête avant l'exécution de l'instruction marquée.

Les modes d'exécution sont décrits en détail au chapitre 7.

## 4.6 Menu "PROGRAMMING".

Ce menu contient toutes les fonctions nécessaires à l'écriture, à l'exécution et à la sauvegarde des programmes. L'éditeur UNIPROG+ sera discuté au chapitre 6 après l'étude des instructions.

### 4.6.1 Programmes à exécuter (start VECTors)

Par ce sous-menu, l'opérateur choisit le "POWER-ON-PROGRAMME". Ce programme est celui qui est exécuté automatiquement à la mise sous tension ou lorsque la lampe rouge du bouton STOP est allumée.

L'opérateur choisit également le "START PROGRAMME". Ce programme démarre à chaque pression sur START si la lampe rouge est éteinte (ou à chaque activation de l'entrée désignée dans le menu de configuration).

Il faut remarquer que le programme d'initialisation est exécuté à nouveau après un arrêt complet du programme et une pression de la touche STOP ou de l'entrée désignée. Si aucun programme d'initialisation n'est souhaité, on peut lui attribuer un fichier vide mais ouvert ou plus simplement le numéro de fichier 100.

### 4.6.2 Vitesses Présélectionnées (FEED rates)

La vitesse présélectionnée entrant dans les instructions de mouvements est puisée dans le tableau de ce menu. Les 7 vitesses, 0 à 6, peuvent être introduites. Il faut les exprimer dans l'unité choisie pour le calcul de FEEDK.

### 4.6.3 Sauvegarde en mémoire FLASH (SAVE)

Il n'est pas nécessaire d'effectuer cette opération après chaque modification de programme ou de paramètre de configuration. En effet, la mémoire volatile (RAM) est maintenue sous tension par une batterie. Cette sauvegarde est donc une sécurité additionnelle au cas où un problème de batterie interviendrait.

2 catégories de données peuvent être sauvées séparément:

1. Tous les fichiers (programmes).
2. Les paramètres de configuration.



Sélectionner la touche F1 pour sauver les programmes, la touche F2 pour sauvegarder les paramètres de configuration ou la touche F4 pour quitter ce menu sans rien faire. La sauvegarde des programmes dure environ 30 secondes, celle des paramètres quelques secondes. En fin de sauvegarde l'un des 2 messages suivants apparaît si l'opération a réussi ou

non.

SAUVEGARDE CORRECTE presser une touche
---

ou

SAUVEGARDE INCORRECTE presser une touche
---

## 4.7 Manipulation de Fichiers, FILE UTILITIES

Les utilitaires d'exploitation travaillent toujours sur le contenu de la mémoire vive.

### 4.7.1 Répertoire, (DIRectory)

L'écran présente des informations sur chaque fichier ouvert. Un fichier peut être ouvert par l'éditeur ou par copie d'un fichier existant.

FILE	SIZE	PROT	FREE
12	45	NO	670

Cet écran signifie que le fichier (FILE) 12 existe, que sa taille (SIZE) est de 45 lignes, qu'il n'est pas protégé (PROT) et qu'il reste 670 lignes disponibles (FREE). Si le fichier 18 n'est pas ouvert, le message suivant apparaît :

FILE	SIZE	PROT	FREE
INTROUVABLE			670

Différentes façons d'utiliser l'utilitaire DIR sont offertes :

- Examen de tous les fichiers :  
Les touches flèches permettent de balayer, dans les deux sens, l'ensemble des fichiers ouverts.

- Examen d'un fichier particulier :  
Entrer directement son numéro (suivi de ENTER), l'un des deux écrans ci-dessus apparaît.
- Modification de la protection :  
La touche F3 agit en pas-à-pas (toggle) et fait apparaître à la rubrique PROT les mots YES ou NO. Un fichier protégé ne peut être modifié par l'éditeur ou effacé. En bloquant l'accès à l'utilitaire DIR et en laissant l'éditeur ouvert, on pourra protéger sélectivement les fichiers.

#### 4.7.2 Effacement d'un Fichier (DELeTe)

L'écran invite à composer le numéro du fichier à effacer. Afin d'éviter tout effacement involontaire, le message "presser CLR pour EFF" invite à presser la touche CLR. On peut alors encore revenir sans action au menu de base en pressant ESC. Après effacement on retrouve le menu "FILE UTILITIES". Si le fichier est protégé, on ne retourne pas au menu de base mais au répertoire avec le fichier protégé à l'affichage.

#### 4.7.3 Copie d'un Fichier (COPY)

L'écran invite à composer le numéro du fichier à copier (SOURCE FILE), puis celui du fichier de destination (DEST. FILE). Plusieurs actions peuvent se produire :

- Le fichier source n'est pas ouvert: aucune action, retour à "FILE UTILITIES"
- Le fichier destination n'est pas créé ou ouvert: un nouveau fichier est créé.
- Le fichier destination existe déjà: le fichier source est mis à la suite du fichier destination.
- Il n'y a pas assez de place dans l'espace des 1250 lignes pour la copie du fichier source: aucune action mais le fait est signalé :

TOO LARGE  
press any key

#### 4.7.4 Chargement depuis la mémoire FLASH (LOAD)

Cette opération est l'inverse de la fonction SAVE, le contenu de la mémoire FLASH vient écraser le contenu de la mémoire RAM.

2 opérations sont possibles:

- Restauration de tous les fichiers (programmes)
- Restauration des paramètres de configuration.

L'affichage indique:

CHARGER LA FLASH  
PROG CONF ESC

Presser F1 pour copier tous les programmes sauvegardés en FLASH dans la RAM. Presser F2 pour copier tous les paramètres de configuration sauvegardés en FLASH dans la RAM. Presser F4 pour quitter ce menu sans rien faire.

En fin d'opération l'un des 2 messages suivants apparaît si l'opération a réussi ou non.

CHARGEMENT CORRECT  
presser une touche

ou

CHARGEMENT INCORRECT  
presser une touche

## 4.8 Menu "DEBUGGING"

### 4.8.1 Utilitaire "TRACE"

Cet utilitaire n'est significatif que pendant l'exécution d'un programme. Il permet de visualiser l'instruction en cours d'exécution. UNIPROG+ étant capable d'un fonctionnement multi-tâches, il faut choisir le "programme simultané" (la tâche) que l'opérateur veut visualiser. La touche F1 permet le choix rotatif du programme simultané.

La ligne supérieure de l'écran affiche l'instruction sous le même format que l'éditeur. La ligne inférieure montre le programme simultané dans lequel se fait la trace, la ligne et le programme :

**S: 1 L: 45 P: 12**

Cet affichage signifie que l'on trace dans le simultané 1, et l'instruction en cours est à la ligne 45 du programme 12.

### 4.8.2 Utilitaire "I/O" control

Cette fonction est extrêmement utile pour la mise au point d'une installation et pour la recherche de pannes. Elle permet la lecture des états de toutes les entrées et sorties sous contrôle d'UNIPROG+, la modification de l'état des sorties et le chargement du registre du convertisseur digital/analogique (DAC).

Pour la signification et la numérotation des entrées et des sorties, se reporter au Tableau 5-1

L'écran du menu I/O CONTROL regroupe les affichages d'une entrée, d'une sortie et du DAC:

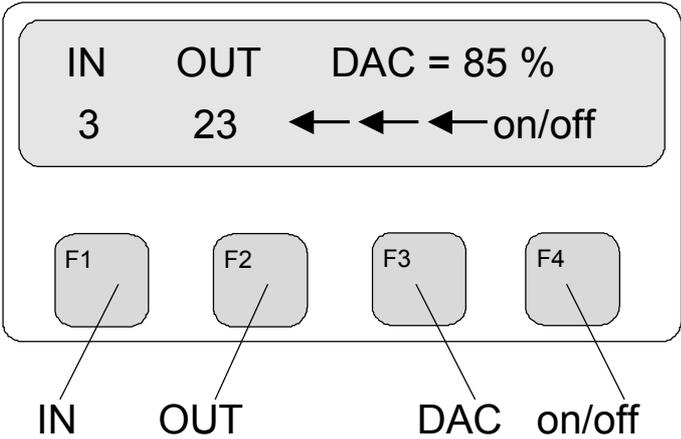


Figure 4-1 : Menu "I/O Control"

Dans cet exemple, l'état de l'entrée 3 est affiché sur le témoin de la touche F1, l'état de la sortie 23 sur le témoin de F2 et le DAC est à 85 % de sa valeur de fin d'échelle, soit 8,5 V (UNIPROG+ ne connaît que le mode unipolaire du DAC, 0.. 10 V pont en position "a" sur la carte.).

- Les touches F1, F2 et F3 déplacent le curseur respectivement sur IN, OUT et DAC. Pour changer le numéro d'une entrée ou d'une sortie, placer le curseur et entrer le numéro. Le procédé est le même pour la valeur à charger dans le DAC.

- La touche F4 modifie l'état de la sortie affichée de façon rotative (toggle).
- Il faut remarquer que cette fonction peut être utilisée pendant l'exécution d'un programme ce qui facilite le diagnostic de dysfonctions telles que le défaut de quittance.
- Selon le Tableau 5-1, les pseudo-I/O numéro 8, 9, 10 sont les états d'activation des programmes simultanés (paragraphe 5.5.5); la touche F4 n'a pas d'effet sur ces trois pseudo-I/O. Les pseudo-I/O 11...15 sont des drapeaux et les états IN et OUT coïncident.

## 5 Instructions du langage UNIPROG+

Les instructions sont décrites ci-après en utilisant les symboles "mnémoniques" apparaissant à l'écran lors de l'édition ou dans l'emploi de l'utilitaire TRACE. Le code numérique nécessaire pour l'entrée au clavier de la commande E-300 accompagne la description des instructions. Dans le descriptif formel d'une instruction, les symboles seront écrits en majuscule et le genre de l'argument en minuscules.

Les instructions et pseudo-instructions utilisées pour la génération de vecteurs et le contournage seront traitées au chapitre 8.

Une instruction ou une donnée numérique occupe une ligne dans la zone de la mémoire réservée à l'utilisateur. Nous appellerons **adresse** d'une ligne le nombre formé par la mise bout à bout du numéro de la ligne dans le fichier (ou le programme) et du numéro du fichier **écrit à deux chiffres**.

**Exemples:** 1245 est l'adresse de la ligne 12 du programme 45,  
102 est l'adresse de la ligne 1 du programme 2,  
6 est l'adresse de la ligne 0 du programme 6....

L'éditeur utilise le vocable "LINE/PROG" pour "adresse".

Les instructions peuvent puiser leur argument numérique principal -la cote dans les déplacements, le temps dans un temporisateur- de plusieurs façons.

- Argument numérique **immédiat** : La valeur numérique se trouve dans la ligne contenant l'instruction.
- Argument numérique **direct** : Dans l'instruction on trouve l'adresse de la ligne contenant la valeur numérique.
- Argument numérique **indirect** : l'instruction contient l'adresse d'un pointeur contenant lui-même l'adresse de la valeur numérique.

**exemples:** voir instructions POSA, POSAD, POSAI ci-dessous

UNIPROG+ peut gérer trois programmes simultanés (fonctionnement multi-tâches). Chaque programme simultané possède un **accumulateur**, registre par lequel on peut faire transiter des quantités numériques.

### 5.1 Instructions de Positionnement

UNIPROG+ prévoit 6 instructions de positionnement : 3 effectuent des positionnements "absolus", c'est-à-dire qu'elles exigent comme argument une cote dans le référentiel courant (voir chapitre 3); les 3 autres effectuent des déplacements "relatifs".

Il faut remarquer qu'UNIPROG+ calcule le point d'arrivée d'un déplacement relatif dans le référentiel initial; une suite de déplacements relatifs ne conduit donc pas à un cumul des erreurs d'arrondi.

#### 5.1.1 Positionnement absolu :

10	POSA	<axe>	<vitesse>	<cote>	<mode-e>
11	POSAD	<axe>	<vitesse>	<adresse de la cote>	<mode-e>
12	POSAI	<axe>	<vitesse>	<adresse du pointeur>	<mode-e>

### 5.1.2 Positionnement relatif :

---

14	POSR	<axe>	<vitesse>	<déplacement>	<mode-e>
15	POSRD	<axe>	<vitesse>	<adresse déplacement>	<mode-e>
16	POSRI	<axe>	<vitesse>	<adresse du pointeur>	<mode-e>

---

Argument "vitesse " : Accepte un entier 0..7. De 0 à 6, la vitesse est prise dans le tableau du paragraphe 4.6.2. Si vitesse = 7, sa valeur est prise dans l'accumulateur.

Argument "mode-e" : Précise le mode d'exécution, accepte 0, 1, 2, 3.  
0: Le mouvement est pris en compte mais pas exécuté.  
1: Tous les mouvements pris en compte sur l'axe désigné sont exécutés.  
2: Tous les mouvements pris en compte sur tous les axes sont exécutés.  
3: UNIPROG+ engendre un vecteur dans l'espace désigné, voir chapitre 8.

Il est ainsi possible d'additionner des mouvements relatifs sur un axe et de les exécuter comme un seul déplacement. Si un positionnement absolu porte le mode 1 ou 2, les mouvements préalablement pris en compte sur l'axe désigné sont sans effet.

### 5.1.3 Définition des outils

---

19	TOOL	<numéro d'outil>
----	------	------------------

---

L'instruction TOOL fixe un nouveau référentiel valable pour toutes les instructions de positionnement subséquentes. Les composantes du vecteur de translation de l'origine sont déposées dans le fichier "0".

- Le tableau "TOOL" permet la consultation ou la modification de ces origines.
- Ces origines peuvent aussi être apprises et validées directement depuis le jogging.
- L'argument "numéro d'outil" sélectionne le groupe des 4 composantes associées à un numéro d'outil.

## 5.2 Autres instructions Cinématiques

### 5.2.1 Prise de référence :

---

17	REF	<axe>
----	-----	-------

---

La prise de référence se fait comme indiqué au paragraphe 3.1. La vitesse de mouvement est prise dans la configuration REF, voir paragraphe 4.4.2.

### 5.2.2 Test de fermeture :

---

18	CLOS	<axe>	<vitesse>
----	------	-------	-----------

---

La fonction du test de fermeture fait l'objet du paragraphe 4.5.3. L'argument "vitesse" est celui des instructions de positionnement (paragraphe 5.1.2).

### 5.2.3 Instruction d'apprentissage :

---

13	TEACH	<axe>	<vitesse>	<adresse>
----	-------	-------	-----------	-----------

---

Les arguments "axe" et "vitesse" sont ceux des instructions de positionnement (voir paragraphe 5.1.2).

Cette instruction permet, dans le cours d'un programme, d'introduire ou de corriger des positionnements par apprentissage. L'instruction TEACH n'est exécutée qu'en mode pas-à-pas, voir chapitre 7. Elle doit nécessairement comporter un marqueur de pause, voir paragraphe 5.9.

TEACH affiche le message suivant :

TEACH-IN (+ - START)
Y 123.345

La deuxième ligne affiche l'axe et le contenu de l'argument "adresse". Deux façons de corriger ce contenu sont offertes à l'utilisateur: le mouvement effectif de l'axe par les touches de jogging ou l'introduction directe de corrections numériques sans déplacement de l'axe. Les deux méthodes peuvent être employées dans une même séance d'apprentissage. Les corrections sont toujours ajoutées au contenu de la ligne adressée par l'instruction. Cette ligne peut être l'argument immédiat, direct ou indirect d'une instruction de positionnement. Il est aussi envisageable de déposer les corrections dans une ligne quelconque pour son utilisation par un algorithme de correction plus compliqué.

Avec les touches de jogging, la position peut être approchée et corrigée en plusieurs mouvements. En programmant une vitesse faible et en se servant du potentiomètre on pourra faire un travail précis. Lorsque la position est jugée atteinte, une pression sur START permet de continuer l'exécution du programme.

## 5.2.4 Définition de paramètres

83	SET	<numéro du paramètre>	<Valeur du paramètre>
----	-----	-----------------------	-----------------------

Numéro du paramètre:

- |   |       |  |
|---|-------|--|
| 0 | PASSE | Valeur de la passe de déburrage.                                 |
| 1 | GAP   | Lors du retour rapide, garde avant l'avance lente.               |
| 2 | DELAY | Délai de fond de perçage.  |
| 3 | BRK-D | Retard de déclenchement des sorties 0 à 7 (LAST DELAYED OUTPUT). |

Les numéros de paramètre 0, 1, 2 concernent le perçage.

## 5.2.5 Cycle de perçage

84	PECK	<axe>	<vitesse lente>	<position fond de perçage>	<mode-d>
----	------	-------	-----------------	----------------------------	----------

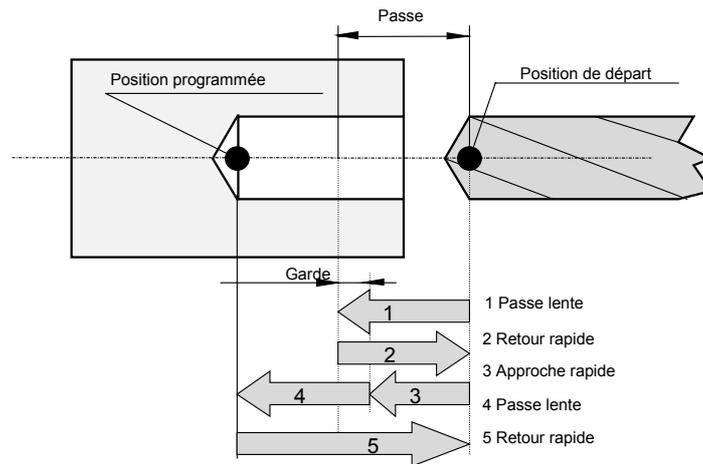
Le cycle de perçage détermine le nombre de passes à effectuer. La passe sera toujours un nombre positif. La position atteinte en fin de cycle de perçage est la position absolue par rapport à l'outil sélectionné.

4 modes sont possibles :

- |           |   |
|-----------|---|
| Mode-d =0 | Déburrage avec retour en fin de cycle à la position de départ.        |
| Mode-d =1 | Brise copeaux avec retour en fin de cycle.                            |
| Mode-d =2 | Déburrage <b>sans</b> retour en fin de cycle à la position de départ. |
| Mode-d =3 | Brise copeaux <b>sans</b> retour en fin de cycle.                     |

- Le retour rapide s'arrête en retrait de l'usinage. La valeur du retrait est appelée la garde (GAP), elle est configurée par l'instruction SET 1 GAP et vaut par défaut 0.1 mm.
- Une temporisation configurée par l'instruction SET 2 DELAY est active en fin de perçage en mode 0 et 1. La temporisation vaut par défaut 0.1 s.

- Les modes 2 et 3 permettent d'enchaîner plusieurs perçages pour obtenir de la progressivité.



**Figure 5-1 : Cycle de perçage**

## 5.2.6 Instruction de taraudage

---

**81 TPING <axe> <pas> <position fond de taraudage>**

---

Cette instruction facilite le choix de l'avance en fonction de la vitesse de rotation de la broche et du pas.

- L'instruction SPVEL détermine la vitesse de rotation de la broche, pilotée par un convertisseur de fréquence.
- Le pas est choisi dans le menu FEED.
- Pour inverser la rotation en fin de taraudage, la sortie 7 s'inverse. Elle peut ainsi être câblée sur l'entrée "sens" du convertisseur de fréquence.
- La rotation de la broche s'inverse lorsque la position finale de l'axe d'avance est atteinte. La rotation de la broche n'est pas mesurée et n'a donc pas d'incidence sur la durée du taraudage. Par conséquent, la profondeur du taraudage peut varier en raison inverse du couple de taraudage.

**Exemple:** Taraudage pas de 1 mm; Rotation 600 t/min; profondeur 50

Menu FEED : RATE # 3 = 1.00 = Pas

```

00 30    10  POSA    X 4 0.00 1
01 30    57  SPVEL   600           ; Vitesse de rotation
02 30    29  ON      7           ; Impose le sens sortie 22 à 1
03 30    29  ON      6           ; Démarre le convertisseur
04 30    72  TPING   X 3 50      ; Démarre le taraudage
  
```

**Remarque:**

l'avance est déterminée sans être asservie sur la vitesse de rotation de la broche ce qui exige l'usage d'un porte-outil avec **compensation de longueur** pour absorber les variations de synchronisation.

## 5.2.7 Déplacement rectiligne

---

**46 CORR <vitesse>**

---

L'instruction CORR effectue un déplacement rectiligne à la vitesse indiquée sur la position de départ du contour en tenant compte du rayon d'outil et de son sens.

### 5.2.8 Angle

---

86	ANGLE	<vitesse>	<valeur>	<mode-e>
----	-------	-----------	----------	----------

---

L'instruction ANGLE donne l'angle par rapport au 0 fixé à trois heures. Un décalage du 0 est possible avec l'instruction ORGA. Le sens positif est anti-horaire.

Si le mode 0 est sélectionné, l'angle est enregistré en attendant l'exécution du mouvement avec l'instruction RADIUS. En mode 1 et 2 le déplacement sur chaque axes est effectué chacun à leur vitesse, en mode 3 le déplacement est linéaire jusqu'au prochain point.

### 5.2.9 Rayon

---

87	RADIUS	<vitesse>	<valeur>	<mode-e>
----	--------	-----------	----------	----------

---

L'instruction RADIUS donne le rayon par rapport au 0 fixé au centre. Le rayon est toujours positif.

Si le mode 0 est sélectionné, le rayon est enregistré en attendant l'exécution du mouvement avec l'instruction ANGLE. En mode 1 et 2 le déplacement sur chaque axes est effectué chacun à leur vitesse, en mode 3 le déplacement est linéaire jusqu'au prochain point.

### 5.2.10 Décalage angulaire

---

88	ORGA	<décalage angulaire>
----	------	----------------------

---

Si le zéro fixé à trois heures ne convient pas, un décalage angulaire est possible en positif ou négatif.

## 5.3 Instructions Entrées/Sorties

Ces instructions permettent de brancher le programme en fonction de l'état d'une entrée, d'attendre selon cet état et modifier l'état d'une sortie.

### 5.3.1 Attente sur entrée :

---

20	WAIT0	<entrée>
21	WAIT1	<entrée>

---

Le programme attend tant que l'entrée désignée soit 0 ou 1 respectivement.

### 5.3.2 Branchement sur entrée :

---

22	BRINO	<entrée>	<adresse>
23	BRIN1	<entrée>	<adresse>

---

Si l'entrée désignée est 0 ou 1 resp., le programme branche à la ligne "adresse". Dans le cas contraire, l'instruction suivante est exécutée. Au sujet de l'adresse de branchement, voir la remarque du paragraphe 5.5.1.

### 5.3.3 Commande des sorties :

---

28	OFF	<sortie>
29	ON	<sortie>

---

La sortie désignée est déclenchée ou enclenchée.

UNIPROG+ reconnaît comme entrées et sorties les ensembles de variables booléennes du Tableau 5-1

Numéro dans instruction	Entrée physique	Numéro dans instruction	Sortie physique
0	IN(0)	0	OUT(0)
1	IN(1)	1	OUT(1)
2	IN(2)	2	OUT(2)
3	IN(3)	3	OUT(3)
4	IN(4)	4	OUT(4)
5	IN(5)	5	OUT(5)
6	IN(6)	6	OUT(6)
7	IN(7)	7	OUT(7)
8	SIM(0)	8	SIM(0)
9	SIM(1)	9	SIM(1)
10	SIM(2)	10	SIM(2)
11	FLAG(0)	11	FLAG(0)
12	FLAG(1)	12	FLAG(1)
13	FLAG(2)	13	FLAG(2)
14	FLAG(3)	14	FLAG(3)
15	FLAG(4)	15	FLAG(4)
16..49	IN(16)..IN(49)	16..63	OUT(16)..OUT(63)
50..59	TOUCHES 0..9		
60	INA 0		
61	INB 0		
62	INA1		
63	INB 1		

**Tableau 5-1 : Entrées/Sorties UNIPROG+**

- Les pseudos I/O, SIM0, SIM1, SIM2, sont les états d'activation des trois programmes simultanés de UNIPROG+.
- Les FLAG(0..4) ne sont pas véritablement des entrées mais des marqueurs -ou drapeaux- internes, librement utilisables par le programmeur. Ces 5 marqueurs sont mis à 0 ou à 1 par les instructions OFF et ON.
- Les entrées 50 à 59 sont les états des touches du clavier.
- Les entrées IN(16..49) et les sorties OUT(16..63) sont matérialisées par des modules d'extension I/O connectés par le bus d'extension. Chaque module porte une adresse à sélectionner sur son commutateur.

O = Open  
C = Closed

Sélecteur 4 3 2 1	Adresse IN OUT
O O C O	16...23
O O C C	24...31
O C O O	32...39
O C O C	40...47
O C C O	48...55
O C C C	56...63
C O O O	64...71
C C C C	0...7

**Tableau 5-2 : Adresse des modules IN/OUT**

Les sorties OUT(0) à OUT(7) sont les sorties comprises dans le boîtier E-300 et disponibles sur le connecteur de la face arrière.

### 5.3.4 Inversion d'une sortie

---

**95 CPL <numéro de la sortie>**

---

Cette instruction complémente (inverse) la sortie en argument.

**Exemple:**

Pour l'activation et la désactivation (toggle) d'une sortie 18 dans un programme manuel 90 (touche 0).

```
90 00    95  CPL    18
90 01    90  NOP
```

### 5.3.5 Affichage d'une valeur

---

**79 DISPD <position>< adresse>**

---

<position> est un nombre entre 0 et 3 qui défini la position où le nombre sera affiché:

0 = Coin supérieur gauche

1 = Coin supérieur droite

2 = Coin inférieur gauche

3 = Coin inférieur droite

< adresse> définit l'adresse où est stocké le nombre. (IDATA or FDATA)

---

**80 RBW**

---

Cette instruction sélectionne la fenêtre qui était affichée avant DISPD.

## 5.4 Manipulation de Quantités Numériques

UNIPROG+ distingue les nombres réels (Floating Numbers) des nombres entiers (Integers). Les vitesses, les cotes, les origines, les temps sont des réels; les adresses, les nombres de cycles sont des entiers.

Les instructions de manipulation de nombre et les instructions arithmétiques supposent a priori que l'argument adressé contient un réel, à l'exception d'une ligne "IDATA" dont le contenu est alors considéré comme entier. Il est donc sans autre possible de modifier par le programme le contenu de l'argument numérique - même immédiat- d'une instruction. Ce contenu doit être un réel.

### 5.4.1 Chargement de l'accumulateur

Chargement immédiat de l'accumulateur:

---

**50 FLOAD <nombre réel>**

---

---

**51 ILOAD <nombre entier>**

---

Chargement direct de l'accumulateur:

---

**52 LOADD <adresse>**

---

Chargement indirect de l'accumulateur :

---

**53 LOADI <adresse pointeur>**

---

#### 5.4.2 Dépose de l'accumulateur

Dépose directe de l'accumulateur :

---

**55 STORD<adresse>**

---

Dépose indirecte de l'accumulateur:

---

**56 STORI <adresse pointeur>**

---

#### 5.4.3 Incrémentation/Décrémentation d'un pointeur :

---

**58 INCD <adresse>**

**59 DECD <adresse>**

---

Ces deux instructions sont utilisées en liaison avec l'adressage indirect; elles permettent de passer à la ligne suivante ou précédente en ajoutant ou en retranchant 100 à l'adresse composite, voir début du présent chapitre. Ces instructions ne permettent pas de franchir les limites d'un fichier.

#### 5.4.4 Sauvegarde d'une variable en mémoire FLASH :

---

**54 SAVE <adresse>**

---

La variable écrite à la position "adresse" est sauvée en mémoire FLASH à la même adresse. Cette opération est relativement lente (plusieurs secondes dans les cas défavorables) et ne devrait être utilisée que dans des portions de programme n'exigeant pas de réaction rapide.

#### 5.4.5 Vitesse de la broche

---

**57 SPVEL <tour par minute>**

---

L'argument de cette instruction est directement la vitesse en tour par minute. Le paramètre MAX RPM 10 volts doit être configuré (avec la valeur 100 la vitesse de rotation est en pour-cent).

#### 5.4.6 Vitesse de rotation des moteurs

---

**85 MOTOR <numéro du moteur> <vitesse de rotation>**

---

Cette instruction réalise le changement de vitesse des moteurs de broches en assurant les délais impératifs de commutation des contacteurs du convertisseur de fréquence.

En effet le convertisseur de fréquence doit être commuté impérativement sans courant, donc à vitesse nul.

Le cycle de l'instruction MOTOR met à zéro la vitesse puis attend le délai de l'instruction SET 4

BRK-D (BReaK-Delay), pour ensuite réactiver la vitesse directement mise en tour par minute.

Avec "MOTOR 0" tous les moteurs sont mis à zéro.  
Avec "MOTOR 1" le moteur 2 est mis à zéro et le moteur 1 démarre (OUT 1).  
Avec "MOTOR 2" le moteur 1 est mis à zéro et le moteur 2 démarre (OUT 2).  
etc... jusqu'à MOTOR 7.

## 5.5 Instructions de Contrôle du Programme

### 5.5.1 Saut inconditionnel :

---

60	JMP	<adresse>
----	-----	-----------

---

L'exécution du programme est transférée à "adresse".

**Remarque :** L'éditeur d'UNIPROG+ peut insérer et supprimer des lignes de programme, opérations qui modifient la numérotation des lignes dans un fichier. Il est donc recommandé (mais pas impératif) d'organiser les programmes de façon à ce que les sauts, appels de sous-programmes etc. se fassent à la ligne 0 d'un fichier.

### 5.5.2 Appel de sous-programme :

---

61	CALL	<adresse>
----	------	-----------

---

Le contrôle est transféré au sous-programme débutant à "adresse". A la fin du sous-programme, le contrôle revient à l'instruction qui suit directement l'appel.

- 10 niveaux d'imbrication des sous-programmes sont autorisés.

### 5.5.3 Fin de programme, de sous-programme :

---

62	END
----	-----

---

Cette instruction marque la fin d'un programme ou d'un sous-programme. Si la dernière instruction d'un programme ou d'un sous-programme coïncide avec la dernière ligne du fichier, l'instruction END est superflue.

### 5.5.4 Répétition :

Les instructions suivantes permettent de construire des boucles à répétition sans avoir à préparer des compteurs de cycles. Il est licite d'imbriquer jusqu'à 10 boucles et une boucle peut s'étendre sur plusieurs fichiers. Une boucle doit nécessairement se terminer par ENDRP.

---

63	REP	<nombre>	argument immédiat
64	REPD	<adresse>	argument direct, nombre
65	ENDRP		limite de la boucle

---

Dans l'instruction REP, l'argument immédiat est nécessairement un entier; dans l'instruction REPD, le contenu de la ligne adressée peut être un entier ou un réel. Dans ce dernier cas, le nombre de répétitions est égal à la partie entière.

### 5.5.5 Appel de programmes simultanés :

---

67	<b>SIM1</b>	<adresse>
68	<b>SIM2</b>	<adresse>

---

Le programme simultané -ou programme parallèle- est mis en marche à l'adresse donnée dans l'instruction. Un programme simultané s'arrête de lui-même s'il contient une instruction END implicite ou explicite. Un programme simultané peut être mis en pause - à partir d'un programme parallèle - en imposant un zéro dans l'élément SIM1/2 (voir Tableau 5-1) par l'instruction OFF et réactivé par l'instruction ON. L'échantillonnage de SIMO.. SIM2 permet à un programme de connaître l'état des autres simultanés.

L'appel d'un simultané actif à une adresse quelconque suspend l'exécution en cours et ce simultané reprend son activité à la nouvelle adresse. La touche STOP avorte tous les simultanés.

### 5.5.6 Branchements conditionnels (Test de l'accumulateur):

Ces instructions permettent de tester le contenu de l'accumulateur, en général après une opération arithmétique. Veuillez noter que le code numérique de ces instructions les placent après BRINO et BRIN1.

Si la condition de branchement n'est pas remplie, le programme exécute l'instruction suivante. Au sujet de l'argument "adresse", voir la remarque du paragraphe 5.5.1.

---

24	<b>BRM</b>	<adresse>	Branchement si le contenu de l'accu. est négatif
25	<b>BRP</b>	<adresse>	Branchement si le contenu de l'accu. est positif ou nul.

---

---

26	<b>BRZ</b>	<adresse>	Branchement si le contenu de l'accu. est nul.
27	<b>BRNZ</b>	<adresse>	Branchement si le contenu de l'accu. est non nul.

---

## 5.6 Instructions de Temporisation

---

70	<b>WAIT</b>	<temps>	Argument immédiat
71	<b>WAITD</b>	<adresse>	Argument direct, temps dans [adresse]

---

Ces instructions introduisent des temps morts dans un programme. Le temps est exprimé en secondes et le contenu de la ligne adressée par WAITD doit être un réel.

## 5.7 Instructions Arithmétiques

Les Instructions arithmétiques opèrent sur l'accumulateur et le contenu de leur argument direct ; Le résultat est retourné à l'accumulateur. La nature de l'opération est dictée par le type de l'argument direct. Plus précisément : si la ligne adressée est une ligne "IDATA", l'instruction effectue une opération sur des entiers, dans tous les autres cas, elle traite les opérandes comme des réels.

---

91	<b>ADDD</b>	<adresse>	Accu = Accu + [adresse]
92	<b>SUBD</b>	<adresse>	Accu = Accu - [adresse]
93	<b>MULD</b>	<adresse>	Accu = Accu . [adresse]
94	<b>DIVD</b>	<adresse>	Accu = Accu / [adresse]

---

## 5.8 NOP et "Pseudo-Instructions"

---

90	<b>NOP</b>
----	------------

---

L'instruction NOP (NO Operation) est en général utilisée pour réserver des lignes en vue d'adjonctions ultérieures. Lors de l'édition, NOP apparaît toujours dans la ligne qui n'est pas encore ouverte.

---

<b>98</b>	<b>FDATA</b>	<b>&lt;nombre réel&gt;</b>
<b>99</b>	<b>IDATA</b>	<b>&lt;nombre entier&gt;</b>

---

Il ne s'agit pas d'instructions mais l'éditeur utilise les préfixes FDATA et IDATA pour introduire des quantités numériques dans un fichier. Le préfixe "IDATA" est utilisé par les instructions pour reconnaître la nature des nombres.

Il est intéressant de noter que les pseudo-instructions 98 et 99 ont l'effet d'un NOP lorsqu'elles sont écrites dans un programme. Il est donc possible de placer des données numériques dans le corps d'un programme.

## 5.9 Marqueur de Pause

Chaque instruction peut être dotée d'un marqueur de pause qui est transparent lorsque le programme est exécuté en mode 1 mais qui suspend le déroulement du programme en mode 2. Pour plus de détails, se reporter au chapitre 7.

## 6 L'éditeur UNIPROG+

On entre dans l'éditeur en sélectionnant le menu "PROGRAMMING" puis la fonction EDIT comme indiqué au paragraphe 4.2. L'affichage invite alors l'opérateur à composer le numéro du fichier ou programme à éditer. La ligne 0 du programme arrive alors à l'écran:

POSA X 3	12.234 0
10	0 p11

La première ligne affiche le symbole de l'instruction et la valeur de ses éventuels arguments. La ligne inférieure contient le code numérique de l'instruction (sous le symbole) et les numéros de ligne et de fichier. Le curseur clignotant se trouve sur le symbole.

Si le fichier n'est pas ouvert, l'écran affiche un NOP et le fichier ne sera effectivement ouvert qu'après le stockage d'au moins une instruction.

### 6.1 Examen d'un programme

- Touches flèches : Affichage de la ligne précédente ↑ ou de la ligne suivante ↓
- Touche F2 : Permet d'afficher directement la ligne introduite au clavier
- Touche ESC : Retour à l'éditeur, invitation à choisir un autre fichier. Une nouvelle pression sur ESC est nécessaire pour retourner au menu de base.

### 6.2 Modification du Contenu d'une ligne

L'introduction d'une nouvelle ligne dans un fichier revient à modifier une ligne contenant NOP. Il suffit donc de décrire la modification d'une ligne.

- Touche ENTER: Déplace le curseur vers la droite sur le prochain argument. La ligne inférieure indique la nature de l'argument. Après le dernier argument, ou après l'instruction s'il n'y pas d'argument, la ligne est stockée et la ligne suivante est amenée à l'écran.
- Touche CLR : Déplace le curseur vers la gauche sur l'argument précédent. Pas d'effet lorsque le curseur se trouve sur l'instruction.
- Touche F5 : Si le curseur se trouve sur le symbole de l'instruction ou sur un argument symbolique, F5 fait défiler de façon rotative et dans le sens croissant tous les choix correspondant à la position de curseur. Sur un argument numérique, F5 est sans effet.

Pour entrer une instruction ou un argument symbolique, il n'est pas nécessaire de faire défiler tous ceux possibles à l'aide de F5. Il est en général plus rapide de taper directement le code numérique correspondant. Avec la pratique, les codes se mémorisent facilement, à tout le moins la classe de l'instruction: les positionnements, classe 10, les manipulations de données, classe 50, les temporisateurs, classe 70 etc. L'usage de la touche F5 sera alors limité à quelques pressions. Il faut remarquer qu'afin d'accélérer la frappe du code de l'instruction, l'entrée du nombre est automatique (sans Enter) après le deuxième chiffre. Pour utiliser F5, il faut dans ce cas reculer le curseur.

Il est naturellement possible de passer de l'examen à la modification de lignes et inversement à chaque instant.

**Exemple:** Entrer l'instruction POSR dans une ligne vide.

- L'instruction est de la classe 10, entrer 10 (sans ENTER), le symbole POSA est affiché, le curseur est sur l'argument "axe".
- Reculer le curseur avec CLR, puis presser F5 en observant le symbole; POSR arrive avec le code 14. Presser alors ENTER.
- L'affichage invite alors à choisir l'axe, ce qui est rapidement fait avec F5. On peut aussi entrer directement 0 pour X, 1 pour Y.
- ENTER introduit l'argument suivant: la vitesse présélectionnée (SEL.SPEED). Il s'agit d'un argument numérique, F5 est sans effet. L'argument suivant est le déplacement (DISP'MENT) qui sera entré en unités techniques.
- Le dernier argument est le mode d'exécution (EXEC MODE), F5 est active et l'entrée directe est aussi aisée:
  - 0 : Pas d'exécution
  - 1 : Exécution sur l'axe de l'instruction
  - 2 : Exécution sur tous les axes
  - 3 : Génération de vecteur.
- ENTER met l'instruction en mémoire et présente la ligne suivante pour édition.

On peut aussi corriger un seul argument d'une instruction en amenant le curseur sur cet argument. Après avoir entré le nouvel argument il faut presser ENTER jusqu'à ce que le curseur disparaisse à droite.

## 6.3 Insertion et Suppression d'une ligne

Une pression sur F3 insère une ligne à la position affichée.

**Exemple:** La ligne 12 est affichée et contient l'instruction WAIT. Après insertion, la ligne 12 contient un NOP et l'instruction WAIT est dans la ligne 13.

Une pression sur F4 supprime la ligne affichée ; la ligne qui la suivait prend son numéro et vient s'afficher.

## 6.4 Mise en Place du Marqueur de Pause

Le marqueur de pause peut être placé ou enlevé de l'instruction à n'importe quel stade de son édition en pressant F1. Le témoin de F1 indique la présence du marqueur. Pour modifier seulement l'état du marqueur il est nécessaire de quitter l'instruction par une ou plusieurs pressions sur ENTER.

## 7 Exécution des programmes

L'exécution d'un programme est contrôlée d'une part par les touches 'START' et 'STOP' (ou les entrées associées désignées par la configuration CTRL) et d'autre part par le mode défini dans le menu MODE (voir paragraphe 4.5.4).

### 7.1 Fonctions des modes MOD1, MOD2

#### MOD1:

Exécution normale, les marqueurs de pause ne sont pas pris en compte. Le témoin START est allumé.

#### MOD2:

Le marqueur de pause arrête le programme **avant** l'exécution de l'instruction marquée. Les programmes simultanés continuent sauf s'ils contiennent aussi des instructions marquées.

Pendant la pause, le témoin START clignote. Une pression sur 'START' redémarre l'exécution jusqu'à la prochaine instruction marquée. Ce mode est particulièrement utile en liaison avec la fonction TRACE.

Le mode d'exécution 2 (LED clignotante) offre l'avantage de saturer les vitesses de déplacement tout en respectant les points d'arrêt introduit en édition par la touche F1.

Dans ce mode, durant la passe de calcul du contour les rayons sont indiqués dans l'ordre d'exécution du contour. La touche START clignotante invite à poursuivre le calcul.

### 7.2 Fonctions des touches START, STOP

Rappelons que ces commandes sont doublées par des entrées définies dans la configuration CTRL.

#### START

Si les témoins START et STOP sont éteints, le programme désigné sous "START PROGRAMME" dans "VECT" est exécuté.

Si le témoin STOP rouge est allumé, le programme désigné sous "POWER ON PROGRAMME" est exécuté.

#### PAUSE

Le déroulement du programme peut être stoppé à la fin de l'instruction en cours en appuyant sur la touche 'F4'. Tous les mouvements sont cependant stoppés selon les rampes de décélération configurées, sans perdre leurs positions. Le témoin START clignote et le témoin F4 est allumé.

#### STOP

Si un cycle est en exécution, une première pression sur la touche 'STOP' avorte immédiatement tous les programmes en cours. Les sorties, la vitesse de broche sont remis à zéro. Les mouvements sont stoppés selon les rampes de décélération configurées, sans perdre leurs positions.

Une deuxième pression sur la touche 'STOP' conserve le témoin STOP rouge allumé, indiquant que le prochain programme exécuté sera le programme d'initialisation "POWER ON PROGRAMME" dans "VECT".

A la mise sous tension de la commande E-300, le mode MOD1 est établi et le "POWER ON PROGRAMME" est exécuté sans aucune intervention. Si on ne désire pas de programme initial, on peut introduire 100 comme "POWER ON PROGRAMME".

#### Remarques:

- La plupart des fonctions des utilitaires peuvent être utilisées pendant l'exécution des

programmes. On peut naturellement perturber gravement le déroulement des programmes en modifiant les fichiers en cours d'exécution. En particulier, les insertions et suppressions de lignes. Il peut en résulter des mal fonctions catastrophiques.

## 7.3 Gestion des fautes

Deux conditions de fautes sont affichées par UNIPROG+:

1. La faute générée par l'étage de puissance,
2. Le dépassement de la course configurée détecté par le logiciel, (seulement en contournage, en positionnement les courses sont a priori limitées),

Quand une faute est active, UNIPROG+ stoppe immédiatement les mouvements mais le sorties restent inchangées. L'écran affiche les messages suivant:



"a" prend la valeur de l'axe en défaut soit: X ou Y.

Une pression sur la touche STOP annule les sorties et le DAC. Dans le premier cas de faute l'affichage indique :



L'axe en défaut peut être déplacé lentement avec les touches de jogging afin de quitter les fins de courses. La touche <ESC> permet de revenir au menu initial. Si la condition de faute est toujours présente, l'arrêt de la commande est nécessaire.

Suite à tous les cas d'erreur la commande exécutera le "POWER ON PROGRAMME".

### **ATTENTION:**

Le bouton 'STOP', ainsi que l'entrée configurée comme STOP externe, ne peuvent pas être considérés comme arrêt d'urgence au sens des réglementations en vigueur.

## 8 Génération de vecteurs et contournage

### 8.1 Introduction: Possibilités et Définition de l'Espace

Les commandes E-300 sont capables d'engendrer des vecteurs dans les espaces X et Y. Les mouvements de contournage selon des courbes quelconques s'obtiennent en exécutant de façon continue de petits segments linéaires mis bout à bout. Le langage de base PINX-E permet de résoudre les problèmes de contournage les plus variés dans des systèmes de coordonnées cartésiennes, polaires ou cylindriques.

UNIPROG+ simplifie grandement la tâche du programmeur en proposant un jeu d'instructions pour l'interpolation linéaire, circulaire et hélicoïdale en **coordonnées cartésiennes**.

Il faut distinguer la génération d'un vecteur et la production d'un contour formé de plusieurs segments linéaires ou/et circulaires exécutés à vitesse constante. Un vecteur unique est produit simplement en imposant le mode d'exécution 3 dans les instructions de positionnement. Par contre, la géométrie d'un contour est définie à l'aide de pseudo-instructions en dehors du programme proprement dit.

Lors de la génération d'un vecteur ou de l'exécution d'un contour, la vitesse est contrôlée sur **l'ordonnée curviligne** qui joue alors le rôle d'un axe fictif. (L'ordonnée curviligne mesure le chemin le long d'un vecteur ou d'un contour.)

Un contour ayant des points anguleux induit des discontinuités de la vitesse des axes; ces discontinuités ne pourront être absorbées par les moteurs pas à pas que dans des mouvements lents.

### 8.2 Génération d'un Vecteur

---

**30**    **DPATH** <espace>    <outil gauche-droite>

---

L'instruction DPATH (**Define PATH**) fixe le plan dans lequel le contour sera effectué.

- L'argument gauche-droite (L-R) indique le côté du départ de la correction de trajectoire d'outil.
- Cette instruction doit impérativement figurer à la première ligne d'un fichier de contour.
- Un vecteur de déplacement nécessite le mode 3 dans les instructions POSA/POSR.

Trois points importants sont à relever:

1. Les mouvements en attente (défini par POSR en mode 0) seront toujours exécutés sur les axes en dehors de l'espace défini.
2. La position de référence doit être définie sur les 2 axes avant la génération de vecteurs en mode 3
3. Le déplacement sera automatiquement limité aux valeurs définies dans les variables STROKE (voir paragraphe 3.3.5)

---

**40**    **ORGP** <axe>    <position absolue>

---

L'instruction ORGP (**ORgin Path**) fixe le référentiel du contour et doit suivre l'instruction DPATH. Si le référentiel du contour est situé sur le point de départ, l'instruction ORGP est superflue.

### 8.3 Définition géométrique d'un Contour

Un contour est un enchaînement de segments linéaires, d'arcs de cercles, d'arcs d'hélices

exécutés comme un seul mouvement. La définition de la géométrie d'un contour est contenue dans un fichier en dehors du programme exécutable. Un fichier de définition géométrique peut contenir plusieurs contours. Le système d'interpolation des commandes E-300 nécessite une interprétation de la définition géométrique UNIPROG+. Cette interprétation produit des tables en mémoire vive, tables utilisées en temps réel par le programme. Le programmeur doit donc demander l'interprétation des contours. Ce problème sera traité au paragraphe 8.4.

Un contour est défini dans son propre référentiel, sans aucun rapport avec les référentiels du chapitre 3. Un contour part toujours de l'origine de son référentiel. Lors de l'exécution, le contour part de la position actuelle des axes; un contour peut être "dessiné" en des endroits différents lors de l'exécution d'un programme. On peut donc considérer le fichier de définition d'un contour comme un sous-programme géométrique.

La définition géométrique d'un contour débute nécessairement par la définition de l'espace. L'instruction "Define PATH" -DPATH- déjà rencontrée au paragraphe 8.2. devient ici une pseudo-instruction. Lorsque plusieurs contours sont stockés dans un même fichier, la pseudo-instruction DPATH sert naturellement de séparateur. (La directive END ne peut être utilisée qu'à la fin du dernier contour du fichier.)

### 8.3.1 Définition d'un segment linéaire

<b>32</b>	<b>LINA</b>	<b>&lt;axe&gt;</b>	<b>&lt;coordonnée&gt;</b>	<b>&lt;mode-e&gt;</b>
<b>33</b>	<b>LINR</b>	<b>&lt;axe&gt;</b>	<b>&lt;composante&gt;</b>	<b>&lt;mode-e&gt;</b>

Les pseudo-instructions LINA et LINR ont un comportement analogue aux instructions POSA et POSR. Les notions de coordonnée absolue et de déplacement relatif se réfèrent à l'espace du contour, (Figure 8-1). et non aux référentiels du chapitre 3. Une pseudo-instruction est requise pour chaque composante du vecteur qui peuvent être données dans un ordre quelconque. La dernière pseudo-instruction doit porter le mode 2.

Les coordonnées ou les composantes sont naturellement données dans l'unité définie par SCALEK. Le mode est 0 pour toutes les composantes sauf la dernière écrite qui doit avoir le mode 2.

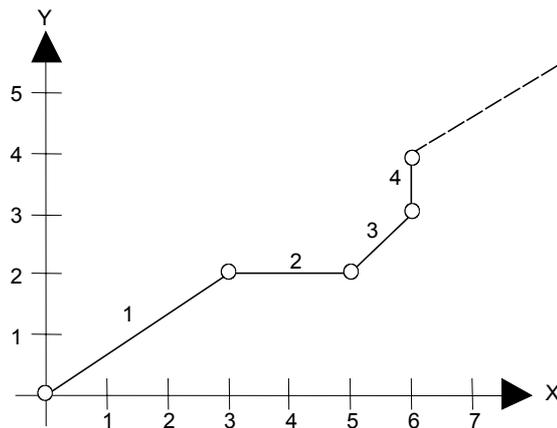
Si l'une des coordonnées ou l'une des composantes du vecteur n'est pas spécifiée, la dernière évocation de l'axe est prise en compte. Cette particularité permet d'économiser des écritures et de programmer simplement des fonctions tabulées, comme le montrent l'exemple ci-après. Cette simplification ne fonctionne plus après la programmation d'un arc de cercle ou d'une instruction POINT.

Les instructions LINA et LINR doivent toujours suivre les instructions POINT voir (POINT).

```

DPATH   XY   L
LINA    X    30
LINA    Y    22   ;seg. 1
LINA    X    52   ;seg. 2
LINA    X    60
LINA    Y    32   ;seg. 3
LINA    Y    52   ;seg. 4
.....
.....

```



**Figure 8-1 : Contour avec diamètre nul**

Si un diamètre de correction d'outil est requis, il est nécessaire d'utiliser l'instruction POINT avec un arc de rayon nul ( $RAD = 0$ ), à la place des instructions LINA ou LINR.

```

DPATH   XY   L
POINT   X    30
POINT   Y    22   ;seg. 1
POINT   X    52   ;seg. 2
POINT   X    60
POINT   Y    32   ;seg. 3
POINT   Y    52   ;seg. 4
.....
.....

```

---

**42    POINT <axe>   <position absolue de l'angle>   <mode-e>**

---

L'instruction POINT offre la possibilité de marquer les coordonnées de l'angle dans lequel le rayon du congé donné par l'instruction RAD, viendra s'inscrire.

Les coordonnées des sommets sont données dans le référentiel du contour, c'est à dire par rapport à un point quelconque du plan du contour. Dans le cas d'une figure présentant une symétrie centrale, les coordonnées seront naturellement données par rapport au centre.

Dans un contour formé de plusieurs instructions POINT, il n'est pas nécessaire de rappeler la dernière coordonnée si elle est identique.

Pour achever un contour formé par les instructions POINT, le fichier doit être terminé par l'instruction LINA ou LINR. Les 2 coordonnées du plan doivent être rappelées pour terminer le contour.

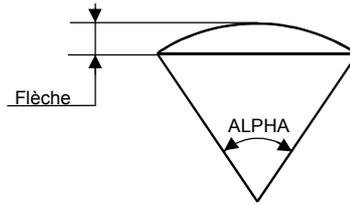
### 8.3.2 Définition d'un arc de cercle

---

**35    CDEF <flèche maximum>**

---

L'instruction CDEF détermine l'angle de segmentation pour obtenir la flèche maximum admissible sur un segment de chaque cercle subséquent.




---

**34 RAD <mode-r > <rayon>**

---

Le mode de l'instruction RAD associée aux instructions CIRR et CIRA, signale le cadran dans lequel le cercle doit être effectué et avec quel rayon. Toutefois cette instruction devient superflue si le cercle suit un autre cercle ou une droite. Le programme UNIPROG+ est capable de générer automatiquement le rayon et le mode pour respecter le contour sans point anguleux.

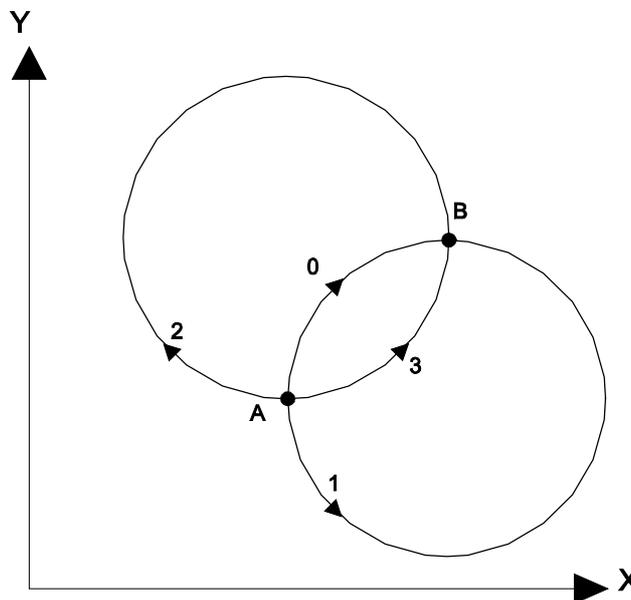
L'instruction RAD associée aux instructions POINT permet de générer le rayon du congé. Dans ce cas le mode est superflu.

Si le rayon est identique, il n'est pas nécessaire de le rappeler pour les cercles subséquents.

Un arc de cercle est donné par les éléments suivants:

- les composantes de la corde sous-tendue par l'arc ou les coordonnées de l'extrémité de l'arc,
- le rayon du cercle,
- le mode qui choisit l'une des 4 solutions possibles, Figure 8-2

Le rayon et le mode sont obligatoires sur le premier arc de cercle. Pour les suivants, ils deviennent facultatifs dans le cas d'un contour sans point anguleux. En effet, UNIPROG+ est capable de déterminer le rayon automatiquement.



*Figure 8-2 : Modes de rotation*

---

<b>36</b>	<b>CIRA</b>	<b>&lt;axe&gt;</b>	<b>&lt;coordonnée&gt;</b>	<b>&lt;mode-e&gt;</b>
<b>37</b>	<b>CIRR</b>	<b>&lt;axe&gt;</b>	<b>&lt;composante&gt;</b>	<b>&lt;mode-e&gt;</b>

---

Les pseudo-instructions CIRA et CIRR se comportent comme LINA et LINR. Si plusieurs arcs de cercle dans le même contour ont le même rayon et le même mode, la pseudo-instruction RAD peut n'être écrite qu'une fois. CDEF et RAD doivent précéder CIRA/CIRR.

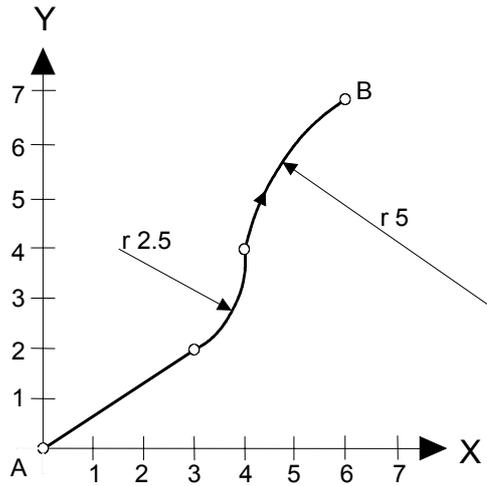


Figure 8-3 : Exemple de contour

A titre d'exemple, donnons deux façons de coder le contour de la Figure 8-3

DPATH	XY	L		DPATH	XY	L	
LINA	X	30		LINR	X	30	
LINA	Y	22		LINR	Y	22	
RAD	3	2.5		RAD	3	2.5	;facultatif
CIRA	X	40		CIRR	X	10	
CIRA	Y	42		CIRR	Y	22	
RAD	0	5		RAD	0	5	;facultatif
CIRA	X	60		CIRR	X	20	
CIRA	Y	72		CIRR	Y	32	
END				END			

## 8.4 Interprétation des Fichiers de Définition des Contours

Avant de pouvoir exécuter un contour, il faut l'interpréter par une instruction PCOMP (Path COMPutation). PCOMP produit pour chaque contour interprété une table dans la mémoire vive (C-MOS). Comme la puissance de calcul est limitée, cette opération est souvent assez longue et il est nécessaire de l'effectuer en temps masqué. Deux cas se présentent :

- La place en mémoire est suffisante pour tous les contours du programme. On placera alors toutes les instructions PCOMP dans le programme d'initialisation. L'interprétation ne se fera qu'une fois à la mise sous tension ou après un arrêt par la touche STOP.
- La place en mémoire n'est pas suffisante ; il faut alors organiser le programme pour interpréter les contours en temps masqué et avec une anticipation suffisante.

L'instruction PCOMP construit les tables en mémoire de façon rotative, c'est à dire que lorsque la mémoire est remplie, PCOMP écrit à nouveau au début. Avec ce mode de travail il faut donc veiller à ce qu'un contour soit exécuté avant que ses calculs ne soient détruits par l'interprétation.

Syntaxe de l'instruction PCOMP :

---

**31 PCOMP <fichier>**

---

L'argument est le numéro du fichier contenant les géométries des contours à interpréter. Si le fichier contient plusieurs contours séparés par DPATH, l'interprétation continue jusqu'à la fin du fichier ou jusqu'à une directive END.

---

**47 TOOLP <numéro d'outil> <numéro de fichier de contour>**

---

L'instruction TOOLP fixe un nouveau référentiel et charge le fichier de contour associé à cet outil pour toutes les instructions de positionnement et de contournage subséquentes. Les composantes du vecteur de translation de l'origine sont déposées dans le fichier "0".

- Le tableau "TOOL" permet la consultation ou la modification de ces origines ainsi que le diamètre de l'outil associé.
- Ces origines peuvent aussi être apprises et validées directement depuis le jogging.
- L'argument "numéro d'outil" sélectionne le groupe des 4 composantes associées à un numéro d'outil.
- Lorsque le programme exécute l'instruction TOOLP, la LED rouge STOP clignote pour signaler la passe de calcul du fichier de contour.
- Le fichier de contour n'est pas recalculé tant qu'il n'a pas été modifié.
- Le contour utilise le numéro d'outil pour déterminer le rayon de correction de trajectoire d'outil.
- Le rayon est déposé dans le tableau "TOOL".

## 8.5 Limitations de longueur pour l'interpolation linéaire

Le module S du segment (longueur dans le plan ou l'espace) en unité spécifique (mm, pouce,...) positionnement en mode d'exécution 3, ne doit pas excéder:

$$S < 67'000'000 / KMUL_{max}$$

### Exemple:

Sélection d'axe : DPATH XY

SCALEK(X) = 1000 pas/mm

DIV(X)= 300

KMUL(X) = 300'000

SCALEK(Y) = 1000 pas/mm

DIV(Y)= 200

KMUL(Y) = 200'000

KMUL<sub>max</sub> = 300'000

S < 223 mm

## 8.6 Exécution des Contours

---

### 48 PATH <vitesse>

---

Si l'instruction PATH est **directement** suivie par l'instruction WAITP alors, le contour démarre sans attendre la fin de celui-ci. Pour éviter des enchaînements de mouvement désordonnés, la suite du programme doit **impérativement** contenir une attente de fin de contour (instruction 66 ENDP), avant de lancer tout autre nouveau mouvement.

---

### 66 ENDP

---

Attend la fin d'un contour (END Path). A utiliser **impérativement** si l'instruction PATH est directement suivie par l'instruction WAITP.

---

### 82 WAITP <axe> <vitesse> <position> <mode-w>

---

Attend si la position absolue de l'axe concerné est plus petite ou plus grande selon le mode.

Mode-w = 0 Attend tant que la position absolue est plus petite que la valeur position programmée.

Mode-w = 1 Attend tant que la position absolue est plus grande que la valeur position

programmée.

- La position programmée tient compte de l'origine d'outil mais pas de l'origine du fichier de contour.
- Cette instruction se place **directement** derrière l'instruction PATH, sans cette condition le contour est exécuté jusqu'à la fin.
- Le temps de réaction de cette instruction dépendent du nombre de fonctions simultanées actives dans la commande.
- Si la position programmée n'est pas atteinte le programme doit être avorté avec la touche STOP.

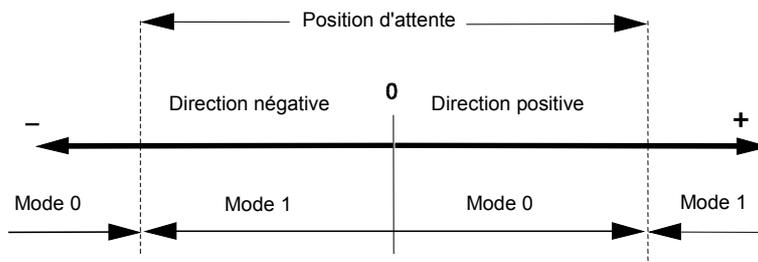


Figure 8-4 : Position d'attente selon le sens de déplacement

## 8.7 Cas n'acceptant pas la correction d'outil

UNIPROG+ ne résout pas tous les cas de contourage, en particulier lorsque des discontinuités (points sécants) apparaissent dans le contour.

Pour résoudre ces cas, il faut créer un segment linéaire tangent à l'arc (longueur minimum de 0.02 mm). Par exemple une droite coupant un arc ou 2 arcs non tangents.



## 8.8 Affichage des erreurs de contour

Durant la passe de calcul du contour des ERREURS numérotées peuvent apparaître en cas de données erronées. Ces erreurs stoppent le calcul.

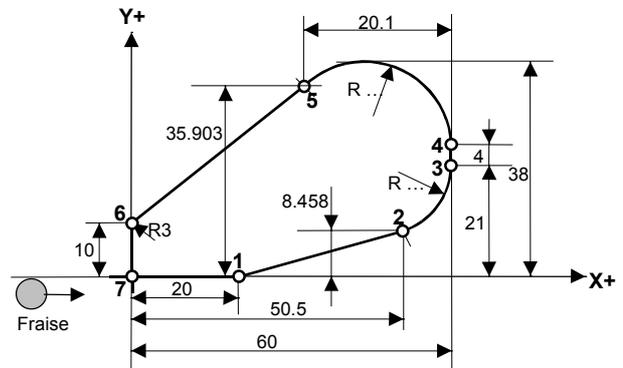
- Erreur 0:** Le rayon est négatif.
- Erreur 1:** Le contour est impossible. Les coordonnées d'arrivée de l'arc (CIRR CIRA) sont au-delà du rayon. Le rayon est négatif.
- Erreur 2:** Division par 0 lors de l'utilisation des instructions POINT. Le rayon est négatif.
- Erreur 3:** Division par 0 lors de la génération automatique d'un arc tangent à un autre arc.
- Erreur 4:** Imprécision dans le calcul de l'angle.
- Erreur 5:** Imprécision dans le calcul du centre.
- Erreur 6:** La détermination du départ du contour est impossible. Le premier segment (LIN) doit mesurer plus de 0.01 mm.
- Erreur 7:** Les instructions LINR ou LINA ne doivent pas précéder l'instruction POINT.

- Erreur 8:** 2 segments linéaires se suivent sur le même axe (LINR, LINA). Le contour peut présenter une discontinuité. Ce cas n'est accepté que lorsque le rayon d'outil est nul.  
L'usage de l'instruction POINT avec un rayon nul peut compenser cet inconvénient.
- Erreur 9:** Division par 0, le contour est impossible, les coordonnées du cercle sont au-delà du rayon.
- Erreur 10:** Il manque l'instruction RAD. En effet la détermination du rayon est impossible lorsque le contour commence par un cercle.
- Erreur 11:** Les coordonnées d'arrivées de l'arc sont confondues avec celles de départ.
- Erreur 12:** La génération du congé est impossible, les 3 coordonnées formant l'angle du polygone sont alignées, par conséquent le rayon est infini.
- Erreur 13:** Il n'existe pas de fichier de contour.
- Erreur 14:** Il manque l'instruction DPATH en première ligne du fichier de contour.
- Erreur 15:** Au moment de l'exécution, le dépôt du contour (BUFFER) est hors zone mémoire. Une manipulation erronée a modifié la première ligne du fichier de contour.
- Erreur 17:** Les instructions POINT ne doivent pas suivre les instructions LINA ou LINR.
- RAYON < 1:** Le contour présente au moins un rayon plus petit que 1 mm. Par conséquent la vitesse d'exécution du contour devra être adaptée pour négocier le rayon.
- SEGMENTATION NULLE:** L'arc est suffisamment court pour ne plus être approché par des segments de droite. Il génère donc une droite sur les coordonnées d'arrivée de l'arc.

## 8.9 Exemple de contournage

### Tableau des outils

ORIGIN X	TOOL 0	?
ORIGIN Y	TOOL 0	?
DIAMETRE	TOOL 0	?
ORIGIN X	TOOL 1	13
ORIGIN Y	TOOL 1	54
DIAMETRE	TOOL 1	8



### File 1

```

0 01 47 TOOLP 1 23
1 01 46 CORR 2
2 01 48 PATH 1
3 01 * 10 POSA X 0 50.0000 0
4 01 10 POSA Y 0 50.0000 2

```

```

;Charge les origines de l'outil 1 le fichier de contour 23
;Origine X 13 et Y 54, diamètre d'outil 8 mm pris dans le tableau TOOL
;Place L'outil sur le contour corrigé, vitesse 2
;Exécute le contour 23 chargé par TOOLP, vitesse 1
;Dégage l'outil, vitesse 0

```

### File 23 sous programme de contour

```

0 23 30 DPATH XY L
1 23 40 ORGP X -1.0000
2 23 35 CDEF 0.01
3 23 42 POINT X 20.0000 0
4 23 34 RAD ? 0.0000
5 23 42 POINT Y 0.0000 2
6 23 32 LINA X 50.5000 0
7 23 32 LINA Y 8.4580 2
8 23 36 CIRA X 60.0000 0
9 23 36 CIRA Y 21.0000 2
10 23 33 LINR X 0.0000 0
11 23 32 LINR Y 4.0000 2
12 23 37 CIRR X -20.100 0
13 23 36 CIRA Y 35.9030 2
14 23 42 POINT X 0.0000 0
15 23 34 RAD ? 3.0000
16 23 42 POINT Y 10.0000 2
17 23 32 LINA X 0.0000 0
18 23 32 LINA Y -2.0000 2

```

```

;Défini l'espace XY et la correction à gauche "L"
;Début du contour en retrait sur X de 1 mm
;Flèche admissible sur l'arc 0.01 mm
;Rayon du congé nul, mode superflu
;1er point, anguleux
;2ème point, tangent avec l'arc
;3ème point, arrivée de l'arc, génération automatique du rayon de l'arc
;4ème point, tangent avec l'arc
;5ème point, arrivée de l'arc, génération automatique du rayon de l'arc
;Rayon du congé 3 mm, mode superflu
;6er point angle du congé
;7ème point, fin du contour en retrait sur Y de 2 mm

```

## 8.10 Résumé des Instructions et Pseudo-Instructions de Définition et de Génération de Contours

Code	Instruction	1 <sup>er</sup> arg.	2 <sup>ème</sup> arg.	3 <sup>ème</sup> arg.	4 <sup>ème</sup> arg.	Description	Page
35	<b>CDEF</b>	Flèche max.				Def. Valeur segmentation	42
36	<b>CIRA</b>	Axe	Coordon.	Mode-e		Definit mvt circulaire abs.	43
37	<b>CIRR</b>	Axe	Compos.	Mode-e		Definit mvt circulaire rel.	43
30	<b>DPATH</b>	espace	Outil g-d			Définit plan de travail	40
66	<b>ENDP</b>					Fin d'un contour	45
32	<b>LINA</b>	Axe	Coordon.	Mode-e		Définit segment droit absolu	41
33	<b>LINR</b>	Axe	Compos.	Mode-e		Définit segment droit relatif	41
40	<b>ORGP</b>	Axe	Pos. Abs.			Définit l'origine d'un contour	40
48	<b>PATH</b>	Vitesse				Exécution d'un contour	45
31	<b>PCOMP</b>	Fichier				Interprétation d'un fichier contour	44
42	<b>POINT</b>	Axe	Pos. Angul.	Mode-e		Liaison de segments av. arrondis	42
34	<b>RAD</b>	Mode-r	Rayon			Def. Rayon et dir. rotation	43
47	<b>TOOLP</b>	N° outil	N° fichier			Fix. réf. et interprétation contour	44
82	<b>WAITP</b>	Axe	Vitesse	position	Mode-w	Déf. Position d'attente	45

## 9 Récapitulation UNIPROG+

### 9.1 Instructions

Code	Instruction	1 <sup>er</sup> arg.	2 <sup>ème</sup> arg.	3 <sup>ème</sup> arg.	4 <sup>ème</sup> arg.	Description	Page
91	ADDD	adresse				Add directe ACCU	34
86	ANGLE	Vitesse	Valeur	Mode-e		Dépl. Val. angle	29
22	BRIN0	entrée	adresse			Branch. Si entrée fausse	29
23	BRIN1	entrée	adresse			Branch. Si entrée vraie	29
24	BRM	adresse				Branch. si accu < 0	34
27	BRNZ	adresse				Branch. si accu <> 0	34
25	BRP	adresse				Branch. si Accu >= 0	34
26	BRZ	adresse				Branch. si Accu = 0	34
61	CALL	adresse				Appel sous-programme	33
35	CDEF	Flèche max.				Def. Valeur segmentation	42
36	CIRA	Axe	Coordon.	Mode-e		Définit mvt circulaire abs.	43
37	CIRR	Axe	Compos.	Mode-e		Définit mvt circulaire rel.	43
18	CLOS	Axe	vitesse			Test fermeture	26
46	CORR	Vitesse				Déplacement linéaire	28
95	CPL	N° sortie				Compl. Etat sortie	31
59	DECD	adresse				Décrémentement directe (adresse)	32
79	DISPD	position	adresse			Affiche une valeur	31
94	DIVD	adresse				Division directe accu	34
30	DPATH	espace	Outil g-d			Définit plan de travail	40
62	END					Fin de programme et routines	33
66	ENDP					Fin d'un contour	45
65	ENDRP					Fin d'une boucle de répétition	33
98	FDATA	réel				Définition d'un nombre réel	35
50	FLOAD	réel				Charge Accu Immédiat, réel	31
99	IDATA	Entier				Définition d'un nombre entier	35
51	ILOAD	Entier				Charge Accu Immédiat, entier	32
58	INCD	adresse				Incrémentement directe (adresse)	32
60	JMP	adresse				Saut incondtionnel	33
32	LINA	Axe	Coordon.	Mode-e		Définit segment droit absolu	41
33	LINR	Axe	Compos.	Mode-e		Définit segment droit relatif	41
52	LOADD	adresse				Charge accu direct (adresse)	32
53	LOADI	pointeur				Charge accu indirect (pointeur)	32
85	MOTOR	N° moteur	Vitesse rot.			Déf. Vitesse rotation moteurs	32
93	MULD	adresse				Multipl. direct accu (adresse)	34
90	NOP					No Operation	34
28	OFF	Sortie				Sortie à 0	29
29	ON	sortie				Sortie à 1	29
88	ORGA	Déc. Angul.				Décalage angulaire	29
40	ORGP	Axe	Pos. Abs.			Définit l'origine d'un contour	40
48	PATH	Vitesse				Exécution d'un contour	45
31	PCOMP	Fichier				Interprétation d'un fichier contour	44
84	PECK	Axe	Vit. lente	Pos. Perc.	Mode-d	Cycle de perçage	27

Code	Instruction	1 <sup>er</sup> arg.	2 <sup>ème</sup> arg.	3 <sup>ème</sup> arg.	4 <sup>ème</sup> arg.	Description	Page
42	POINT	Axe	Pos. Angul.	Mode-e		Liaison de segments av. arrondis	42
10	POSA	axe	vitesse	Coordon.	Mode-e	Positionnement absolu imméd.	25
11	POSAD	axe	vitesse	adresse	Mode-e	Positionnement absolu direct	25
12	POSAI	axe	vitesse	pointeur	Mode-e	Positionnement absolu indirect	25
14	POSR	axe	vitesse	Dépl.	Mode-e	Positionnement relatif Imméd.	26
15	POSRD	axe	vitesse	adresse	Mode-e	Positionnement relatif direct	26
16	POSRI	axe	vitesse	pointeur	Mode-e	Positionnement relatif indirect	26
34	RAD	Mode-r	Rayon			Def. Rayon et dir. rotation	43
87	RADIUS	Vitesse	Valeur	Mode-e		Rayon de dépl. (coord. polaires)	29
90	RBW					Restore la fenêtre courante	31
17	REF	axe				Déf. Point de référence	26
63	REP	Nb de fois				Répétition, immédiat	33
64	REPD	Adresse				Répétition, directe	33
54	SAVE	Adresse				Sauvegarde en EEPROM	32
83	SET	N° param	Val. param.			Attribution de variables	27
67	SIM1	adresse				Appel 1 <sup>er</sup> programme simultané	34
68	SIM2	adresse				Appel 2 <sup>ème</sup> programme simultané	34
57	SPVEL	Rot/min				Vitesse broche	32
55	STORD	adresse				Sauvegarde Accu, Directe	32
56	STORI	pointeur				Sauvegarde Accu, Indirecte	32
92	SUBD	adresse				Soustraction Accu, Directe	34
13	TEACH	axe	vitesse	adresse		Corr. Pos. par apprentissage	26
19	TOOL	N° outil				Définition référentiel outil	26
47	TOOLP	N° outil	N° fichier			Fix. réf. et interprétation contour	44
81	TPING	Axe	Pas	Pos. fin		Définition d'un taraudage	28
70	WAIT	Temps				Temporisation, adr. immédiat	34
20	WAIT0	Entrée				Attente entrée <> 0	29
21	WAIT1	Entrée				Attente entrée <> 1	29
71	WAITD	adresse				Temporisation, adr. directe	34
82	WAITP	Axe	Vitesse	position	Mode-w	Déf. Position d'attente	45
89	ZTOOL	Axis	Input nb	direction			
72	2HON						
73	2HOFF						

## 9.2 Entrées-Sorties (voir aussi 4.4.2)

Input	Item	Output	Item
0	IN(0)	0	OUT(0)
1	IN(1)	1	OUT(1)
2	IN(2)	2	OUT(2)
3	IN(3)	3	OUT(3)
4	IN(4)	4	OUT(4)
5	IN(5)	5	OUT(5)
6	IN(6)	6	OUT(6)
7	IN(7)	7	OUT(7)
8	SIM(0)	8	SIM(0)
9	SIM(1)	9	SIM(1)
10	SIM(2)	0	SIM(2)
11	FLAG(1)	11	FLAG(1)
12	FLAG(2)	12	FLAG(2)
13	FLAG(3)	13	FLAG(3)
14	FLAG(4)	14	FLAG(4)
15	FLAG(5)	15	FLAG(5)
16..49	IN(16..49)	16..63	OUT(16..63)
50..59	TOUCHES 0..9		
60	INA0		
61	INB 0		
62	INA1		
63	INB 1		

# 10 Raccordement de la commande E300

## 10.1 Commande compacte type E300-CMP

### 10.1.1 Compatibilité avec E-600

Les connecteurs I/O EXT et modules E600-3 sont identiques entre E300 CMP et E600 Base. Le connecteur I/O est semblable mais pas compatible avec le E600. Toutefois, l'insertion accidentelle d'un connecteur confectionné pour E600 dans un E300 n'est pas destructive. La position dans le connecteur de certains signaux est la même que dans le E600.

Le tableau suivant compare les deux connecteurs I/O:

Comparaison des connecteurs I/O 19 pôles E300 et E600		
Broche	E300	E600
A	0V	0V
B	OUT4	OUT4
C	OUT5	OUT5
D	OUT6	OUT6
E	OUT7	OUT7
F	IN0	Analog GND
G	IN4	Sortie DAC
H	OUT0	OUT0
J	IN1	+5VDC (sortie)
K	IN5	Entrée ADC
L	IN2	IN2
M	IN6	IN6
N	IN3	IN3
P	IN7	IN7
R	0V	AGND
S	OUT1	OUT1
T	OUT2	OUT2
U	OUT3	OUT3
V	+24VDC	+24VDC

*Tableau 10-1 : Comparaison I/O E300 et E600*

Le connecteur ANALOG I/O n'existe pas dans le E600. Le connecteur RS-232 n'est pas le même que dans le E600.

### 10.1.2 Connecteur I/O

Le connecteur désigné I/O regroupe les entrées/sorties et les tensions d'alimentation 24V pour celles-ci.

Broche	Signal et Destination
A	0 V, retour pour les sorties
B	OUT(4), sortie active haute, 24 V, 1 A
C	OUT(5), sortie active haute, 24 V, 1 A
D	OUT(6), sortie active haute, 24 V, 1 A
E	OUT(7), sortie active haute, 24 V, 1 A
F	IN(0) entrée active haute
G	IN(4) entrée active haute
H	OUT(0), sortie active haute, 24 V, 1 A
J	IN(1) entrée active haute
K	IN(5) entrée active haute
L	IN(2) entrée active haute
M	IN(6) entrée active haute
N	IN(3) entrée active haute
P	IN(7) entrée active haute
R	0 V, retour pour les sorties
S	OUT(1), sortie active haute, 24 V, 1 A
T	OUT(2), sortie active haute, 24 V, 1 A
U	OUT(3), sortie active haute, 24 V, 1 A
V	+ 24 V non régulée

**Tableau 10-2 : E300, connecteur I/O, Burndy 19 broches**

1) Chaque sortie peut délivrer 1A mais le courant total des 8 sorties ne doit pas dépasser 4 A.

### 10.1.3 Connecteur I/O EXT.

Le connecteur désigné I/O EXT. regroupe les signaux nécessaires pour le bus des modules externes entrées E-500-I1, I2, I3 et sorties E-500-ODC1.

### 10.1.4 Connecteur RS 232

Le connecteur désigné RS 232 permet de connecter une interface reconnue par la commande E-300, câble 1:1 pour un PC.

- Compatible IBM PC ou AT et logiciel NEWWINCOM ou APEX

### 10.1.5 Module E-600-3, pour moteurs 2 phases, drivers EIP

Etage de puissance bipolaire à 1600 micro-pas par tour. Contrôle du courant combiné "slow/fast decay"

Broche	Signal et destination
A	Enroulement phase B
B	Enroulement phase B
C	Enroulement phase A
D	Enroulement phase A
E	Entrée active haute 24 V, INA
F	Entrée active haute 24 V, INB
G	Alimentation des détecteurs 24V
H	Commun des détecteurs 0V

**Tableau 10-3 : E600-3, connecteur, Burndy 8 broches**

#### 10.1.5.1 Réglage du courant

Le commutateur rotatif permet de choisir le courant de crête par phase adapté au moteur.

Valeur du courant valable lorsque le signal "BOOST" est actif. S'il est inactif le courant n'est que de 60 %.

Position	Courant	Position	Courant
0	2.0 A	5	5.3 A
1	2.7 A	6	6.0 A
2	3.3 A	7	6.7 A
3	4.0 A	8	7.3 A
4	4.6 A	9	8.0 A

*Tableau 10-4 : E600-3, réglage du courant*

### 10.1.6 Connecteur ANALOG I/O

Ce connecteur sert à connecter des éléments nécessitant des entrées ou des sorties analogiques programmables. Ce connecteur est de type DB 9 mâle. Son brochage est le suivant:

Pin	Description	Remarque
1	+5Vref	OUT
2	ADC1	IN
3	ADC2	IN
4	DAC0	OUT
5	DAC1	OUT
6	AGND	-
7	AGND	-
8	AGND	-
9	AGND	-

*Tableau 10-5 : Connecteur ANALOG I/O*