

# **UNIPROG+ V7-416**

**Programme universel  
Pour commandes E-600 Base et E-600 ND**

Version: août 2006

# Table des matières :

<b>1</b>	<b>Introduction .....</b>	<b>6</b>
1.1	Caractéristiques d'UNIPROG+ .....	6
1.2	Compatibilité avec UNIPROG .....	7
1.3	Compatibilité avec POLYTOOL.....	7
1.4	Utilisation de ce manuel .....	7
<b>2</b>	<b>Organisation des zones de stockage .....</b>	<b>8</b>
2.1	Mémoires de la Commande E-600.....	8
2.2	L'Espace de Stockage Accessible à l'Utilisateur .....	8
<b>3</b>	<b>Géométrie et cinématique .....</b>	<b>9</b>
3.1	Position de référence.....	9
3.2	Changement de Référentiel.....	11
3.3	Constantes Cinématiques .....	12
3.3.1	<i>Le facteur d'échelle SCALEK.....</i>	<i>12</i>
3.3.2	<i>Fréquence maximale, Paramètre DIV.....</i>	<i>13</i>
3.3.3	<i>Constante d'Accélération KUP, Décélération, KDN .....</i>	<i>14</i>
3.3.4	<i>Constante de Vitesse FEEDK.....</i>	<i>14</i>
3.3.5	<i>Course de l'axe, STROKE + / STROKE –.....</i>	<i>15</i>
3.3.6	<i>Déplacement permanent de l'origine, OFFSET .....</i>	<i>15</i>
3.3.7	<i>Sur-excitation du moteur, BOOST .....</i>	<i>15</i>
3.4	Constantes Cinématiques des axes asservis SERV.....	16
3.4.1	<i>Asservissement actif SERVO ENB. (Servo Enable) .....</i>	<i>16</i>
3.4.2	<i>Plage de l'erreur de position POS ZONE. (Position Zone).....</i>	<i>16</i>
3.4.3	<i>Plage de l'erreur de poursuite ALM ZONE. (Alarm Zone).....</i>	<i>16</i>
3.4.4	<i>Valeur du gain HI GAIN, LOW GAIN (High, Low Gain).....</i>	<i>16</i>
3.5	Contrôle de Fermeture .....	16
<b>4</b>	<b>Mode opératoire du clavier et fonctions de UNIPROG+.....</b>	<b>18</b>
4.1	Mise sous tension de la commande E-600. ....	18
4.2	Sélection des Menus .....	19
4.3	Menu "OTHER".....	20
4.3.1	<i>Accès aux protections.....</i>	<i>20</i>
4.3.2	<i>Version.....</i>	<i>21</i>
4.4	Menu "CONFIGURATION" .....	21
4.4.1	<i>Configuration des Générateurs de Mouvement (Motion GENerators) .....</i>	<i>21</i>
4.4.2	<i>Configuration de la Prise de Référence (REFerence).....</i>	<i>21</i>
4.4.3	<i>Configuration des Entrées de Contrôle (CONTRoL) .....</i>	<i>22</i>
4.4.4	<i>Configuration des axes asservis (SERVo).....</i>	<i>22</i>
4.5	Menu "MOTION CONTROL" .....	23
4.5.1	<i>Tableau d'outils "TOOL".....</i>	<i>23</i>
4.5.2	<i>Mouvements Manuels, JOG.....</i>	<i>23</i>
4.5.3	<i>Contrôle de fermeture, CLOS .....</i>	<i>25</i>
4.5.4	<i>Affichage des cotes, DISP .....</i>	<i>26</i>
4.6	Menu "PROGRAMMING". .....	26
4.6.1	<i>Programmes à Exécuter (start VECTors).....</i>	<i>26</i>

4.6.2	Vitesse Présélectionnées (FEED rates).....	26
4.6.3	Sauvegarde sur la Carte-Données (SAVE).....	26
4.7	Manipulation de Fichiers, FILE UTILITIES.....	27
4.7.1	Répertoire, (DIRectory).....	27
4.7.2	Effacement d'un Fichier (DELeTe).....	28
4.7.3	Copie d'un Fichier (COPI).....	28
4.7.4	Chargement de la Carte-Données dans la Mémoire (LOAD).....	28
4.8	Menu "DEBUGGING".....	29
4.8.1	Utilitaire "TRACE".....	29
4.8.2	Utilitaire "I/O" control.....	29
4.8.3	Utilitaire "PRT".....	30
4.8.4	Utilitaire "SERVO".....	30
<b>5</b>	<b>Instructions du langage UNIPROG+.....</b>	<b>32</b>
5.1	Instructions de Positionnement.....	32
5.1.1	Positionnement absolu :.....	32
5.1.2	Positionnement relatif :.....	33
5.1.3	Définition des outils.....	34
5.2	Autres instructions Cinématiques.....	35
5.2.1	Prise de référence :.....	35
5.2.2	Test de fermeture :.....	35
5.2.3	Instruction d'apprentissage :.....	35
5.2.4	Définition de paramètres.....	36
5.2.5	Cycle de perçage.....	36
5.2.6	Instruction de taraudage.....	37
5.2.7	Déplacement rectiligne.....	37
5.2.8	Angle.....	38
5.2.9	Rayon.....	38
5.2.10	Décalage angulaire.....	38
5.2.11	Prise de zéro pour ajustement automatique d'un outil.....	38
5.3	Instructions d' Entrées/Sorties.....	39
5.3.1	Attente sur entrée :.....	39
5.3.2	Branchement sur entrée :.....	39
5.3.3	Commande des sorties :.....	39
5.3.4	Inversion d'une sortie.....	41
5.4	Vitesses de broches.....	41
5.4.1	Vitesse de la broche.....	41
5.4.2	Vitesse de rotation des moteurs.....	41
5.5	Manipulation de Quantités Numériques.....	41
5.5.1	Chargement de l'accumulateur.....	42
5.5.2	Dépose de l'accumulateur.....	42
5.5.3	Incrémentation/Décrémentation d'un pointeur :.....	42
5.5.4	Sauvegarde d'une variable dans la carte "données" :.....	42
5.6	Instructions de Contrôle du Programme.....	43
5.6.1	Saut incondtionnel :.....	43
5.6.2	Appel de sous-programme :.....	43
5.6.3	Fin de programme, de sous-programme :.....	43
5.6.4	Répétition :.....	43
5.6.5	Appel de programmes simultanés :.....	43
5.6.6	Branchements conditionnels (Test de l'accumulateur):.....	44
5.7	Instructions de Temporisation.....	44

5.8	Instructions Arithmétiques .....	44
5.9	NOP et "Pseudo-Instructions" .....	44
5.10	Marqueur de Pause .....	45
<b>6</b>	<b>L'éditeur UNIPROG+ .....</b>	<b>46</b>
6.1	Examen d'un programme .....	46
6.2	Modification du Contenu d'une ligne .....	46
6.3	Insertion et Suppression d'une ligne .....	47
6.4	Mise en Place du Marqueur de Pause .....	47
<b>7</b>	<b>Exécution des programmes .....</b>	<b>48</b>
7.1	Fonctions des modes MOD1, MOD2 .....	48
7.2	Fonctions des touches START, PAUSE, STOP.....	48
7.3	Gestion des fautes.....	49
<b>8</b>	<b>Génération de vecteurs et contournage.....</b>	<b>50</b>
8.1	Introduction: Possibilités et Définition de l'Espace .....	50
8.2	Génération d'un Vecteur.....	50
8.3	Définition géométrique d'un Contour .....	50
	8.3.1 Définition d'un segment linéaire.....	51
	8.3.2 Définition d'un arc de cercle.....	52
8.4	Interprétation des Fichiers de Définition des Contours .....	54
8.5	Limitations de longueur pour l'interpolation linéaire .....	55
8.6	Limitations du nombre total de segments.....	55
8.7	Exécution des Contours .....	56
8.8	Cas n'acceptant pas la correction d'outil.....	57
8.9	Affichage des erreurs de contour .....	58
8.10	Exemple de contournage.....	59
8.11	Résumé des Instructions et Pseudo-Instructions de Définition et de Génération de Contours 60	
<b>9</b>	<b>Raccordements de la commande E-600.....</b>	<b>61</b>
9.1	Introduction .....	61
9.2	E-600-base, alimentation, logique et panneau.....	61
	9.2.1 Connecteur I/O.....	62
	9.2.2 Connecteur I/O EXT.....	62
	9.2.3 Connecteur RS 232 .....	62
9.3	Compatibilité avec E-500.....	62
9.4	Module E-600-1, drivers BERGER, pour moteurs 5 phases.....	63
9.5	Module E-600-2, contrôle de rotation, drivers BERGER.....	63
9.6	Module E-600-3, pour moteurs 2 phases, drivers EIP .....	64
	9.6.1 Réglage du courant.....	64
9.7	Module E-600-13, pour asservissement.....	64
9.8	Module E-600-8, pour YASKAWA.....	65
9.9	Module E-600-9, pour translateur pas-à-pas.....	65
9.10	Module E-600-11, Pour entrée codeur .....	66
<b>10</b>	<b>Raccordements de la commande E-600-ND .....</b>	<b>67</b>

10.1	E-600-ND-BASE .....	67
	10.1.1 E-600-ND connecteur I/O.....	67
	10.1.2 E-600-ND connecteur I/O EXT.....	67
	10.1.3 E-600-ND connecteur RS 232 .....	67
10.2	Module E-600-13N, pour asservissement .....	68
10.3	Module E-600-8N, pour YASKAWA .....	68
10.4	Module E-600-9N, pour translateur pas-à-pas .....	68
<b>11</b>	<b>Récapitulation UNIPROG+ .....</b>	<b>69</b>
11.1	Instructions .....	69
11.2	Entrées-Sorties .....	71
11.3	Divers.....	71

### Liste des figures :

Figure 3-1	: Courses et prise de référence.....	9
Figure 3-2	: Les référentiels UNIPROG+.....	12
Figure 3-3	: Fréquence ou vitesse en fonction de t.....	14
Figure 4-1	: Menu "I/O Control" .....	29
Figure 5-1	: Cycle de perçage .....	37
Figure 8-1	: Contour avec diamètre nul .....	51
Figure 8-2	: Modes de rotation .....	53
Figure 8-3	: Exemple de contour .....	53
Figure 8-4	: Position d'attente selon le sens de déplacement.....	56

### Liste des tableaux :

Tableau 5-1	: Entrées/Sorties UNIPROG+.....	40
Tableau 5-2	: Adresse des modules IN/OUT .....	40
Tableau 9-1	: E600, connecteur I/O, Burndy 19 broches .....	62
Tableau 9-2	: E600-1, connecteur, Burndy 19 broches.....	63
Tableau 9-3	: E600-2, connecteur, Sub-D 9 broches pour codeur .....	63
Tableau 9-4	: E600-3, connecteur, Burndy 8 broches.....	64
Tableau 9-5	: Numérotation des axes .....	64
Tableau 9-6	: E600-3, réglage du courant.....	64
Tableau 9-7	: E600-13, connecteur J1 .....	65
Tableau 9-8	: E600-13, connecteur J2 .....	65
Tableau 9-9	: E600-9, connecteur, Sub-D femelle 9 broches .....	66
Tableau 9-10	: E600-11, connecteur, Sub-D 9 broches pour codeur .....	66
Tableau 10-1	: E600-ND, connecteur I/O, Sub-D femelle 15 broches .....	67

# 1 Introduction

La commande de positionnement et de contournage de la série E-600 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 à 4 axes pour moteurs pas-à-pas hybrides à 2 et 5 phases dont 2 peuvent être des axes asservis.

Les connexions sont largement compatibles avec la commande E-500, toutefois **les cartes programme ne sont pas directement compatibles** (possibilité de conversion par le logiciel UNICOM).

Le coffret "**E-600 base**" en est le composant principal. Il est équipé de l'alimentation, la logique de commande, le clavier, l'affichage et les entrées/sorties interne. Ce coffret peut recevoir divers modules de puissance. Il existe en outre une version "**E-600-ND**" (No drive), sans boîtier ni puissance, destinée au montage encastré.

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+**<sup>1</sup> qui offre un ensemble de fonctions utilitaires et la possibilité d'écrire des programmes performants simplement à partir du panneau de la commande.

Les programmes écrits par l'utilisateur et la configuration de la commande sont sauvés dans une carte-mémoire (BEE-CARD) ce qui constitue le moyen d'archivage le plus pratique et le plus fiable.

## 1.1 Caractéristiques d'UNIPROG+

Le programme **UNIPROG+** est lui-même écrit en langage PINX-E et il est construit de façon à pouvoir évoluer, par exemple en incorporant des instructions spécifiques à certaines applications.

- Le programme UNIPROG+ V 7.416 est destiné à remplacer les logiciels UNIPROG et POLYTOOL avec une ergonomie très semblable.
- Il présente l'avantage d'accepter un correcteur de trajectoire de rayon d'outil sur des contours, (instructions LIN, CIR, POINT).
- Une restriction exige que le contour se présente sans point anguleux, tous les segments et les arcs doivent être tangents. Cet inconvénient est compensé par la génération automatique du rayon et du mode, lorsque plusieurs arcs s'enchaînent; ainsi l'instruction RAD devient superflue.
- Pour simplifier la programmation, UNIPROG+ permet aussi de programmer des angles avec leurs congés (anciennement POLYTOOL).
- Durant le calcul du contour de nombreux messages signalent certaines erreurs de contournage. D'autre part, en mode 2, l'affichage indique les rayons qui définissent le contour.

**Pour installer UNIPROG+** sur le contrôleur E-600 équipé des logiciels UNIPROG ou POLYTOOL, veuillez lire les paragraphes 1.2 et 1.3 si les anciens programmes pièces doivent être conservés.

---

<sup>1</sup> Le nom de ce logiciel ne doit pas être confondu avec UNIPPLUS développé par la société COMBITEC à Bienne.

## 1.2 Compatibilité avec UNIPROG

- 1) Si dans les anciens fichiers l'instruction 19 ORG n'est pas utilisée, le paramètre "LAST TOOL NB" doit être mis à zéro. Dans le cas contraire, il faut libérer le fichier "0" qui sera utilisé pour déposer les origines d'outil. Par conséquent il est nécessaire de le copier dans un autre fichier et d'adapter les adresses faisant appel au fichier "0".
- 2) Pour obtenir la valeur du DAC en % le paramètre "MAX RPM 10 volts" est mis à 100. D'autre part, dans le programme l'association des instructions FLOAD et LDAC est remplacée par l'instruction 57 SPVEL suivit de la vitesse de rotation.
- 3) La définition angulaire donnée par l'instruction CDEF devra être remplacée par la flèche maximum admissible.
- 4) Pour obtenir le START uniquement sur une entrée, le paramètre "2 HAND START" doit être mis à 8.
- 5) Pour éviter l'appel des programmes en jogging, le paramètre "MANUAL PROGRAMME" doit être mis à 100 (voir CTRL et JOG).

## 1.3 Compatibilité avec POLYTOOL

- 1) Le fichier "0" qui est utilisé pour déposer les origines d'outil passe de 4 à 5 données par outil. Par conséquent, les origines d'outil seront décalées.
- 2) Voir paragraphes numéro 2, 3, 4, 5 de la compatibilité avec UNIPROG ci-dessus.
- 3) Les 2 premières instructions POINT seront remplacées par les instructions 40 ORGP.
- 4) Le code 32 de l'instruction POINT devient le code 42 et le mode 2 devra être ajouté à la deuxième coordonnée.
- 5) L'instruction DTOOL disparaît et l'argument gauche/droite est mis dans l'instruction DPATH. Il faut donner le diamètre déposé dans le tableau "TOOL". Le diamètre est associé au numéro d'outil choisi par l'instruction TOOLP.
- 6) Le paramètre TOOL L/R INVERT doit s'adapter au sens des axes.
- 7) Les contours sont toujours ouverts et terminés par les instructions LINA ou LINR.

## 1.4 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 9 donne les indications pour le branchement des signaux de contrôle. Pour des informations plus détaillées se référer au manuel des modules intégrés dans la commande.
- 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 contourage (interpolation linéaire et circulaire).

## 2 Organisation des zones de stockage

### 2.1 Mémoires de la Commande E-600

Les commandes E-600 possèdent 4 mémoires physiques distinctes :

- une mémoire non modifiable (EPROM) de 32 kbytes, la "mémoire-système basse",
- une mémoire modifiable (FEPRM) de 32 kbytes, la "mémoire-système haute",
- une mémoire vive interne de 32 kbytes maintenue par une pile,
- une carte-mémoire modifiable (EEPROM) de 8 kbytes, la "carte-données".

La mémoire-système basse contient l'interpréteur du langage PINX-E et des routines de programme.

En chargeant la mémoire-système haute, une commande E-600 peut être instantanément reconvertie pour exécuter des tâches n'utilisant pas UNIPROG+.

La mémoire-système ne peut être modifiée qu'avec l'aide des outils de développement APEX et d'un programmeur de EPROM muni d'un adaptateur.

La carte-données constitue le moyen d'archivage des programmes de l'utilisateur (programme pièce) et de la configuration de l'équipement.

La mémoire vive (RAM C-MOS) est la véritable zone de travail. Si une carte-données est enfichée, son contenu est transféré dans la mémoire vive à la mise sous tension de la commande. L'exécution d'un programme d'utilisateur se fait toujours à partir de la mémoire vive à la mise sous tension de la commande. L'édition de programmes, la modification des paramètres de configuration n'affectent aussi que la mémoire vive. Le sauvetage du contenu de la mémoire vive dans une carte-données est une opération volontaire de l'opérateur.

### 2.2 L'Espace de Stockage Accessible à l'Utilisateur

A l'aide des différents utilitaires d'UNIPROG+, l'utilisateur construit ou modifie son espace de stockage en mémoire vive. Cette zone de stockage sera copiée en bloc sur la carte-données par l'utilitaire de sauvegarde.

Une zone fixe de l'espace de stockage est réservée aux paramètres de configuration, voir paragraphe 4.4.

Le solde de cet espace, soit 7500 bytes, contient les programmes et les fichiers de données numériques de l'utilisateur.

L'unité de stockage de l'utilisateur est la "**ligne**", soit 6 bytes. Une instruction ou une donnée numérique occupe toujours une ligne. 1250 lignes sont disponibles groupées librement en "**programmes**" ou "**fichiers**". On utilisera le terme "programme" pour un fichier contenant des instructions bien que les deux termes recouvrent la même réalité.

100 fichiers numérotés de 00 à 99 peuvent être ouverts dans l'ensemble de 1250 lignes. Un fichier ou un programme est ouvert par l'utilitaire d'édition ou par copie d'un fichier existant.

Naturellement, l'espace utilisateur peut être multiplié à volonté par changement de la carte-données.



### 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-600.

#### 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 :

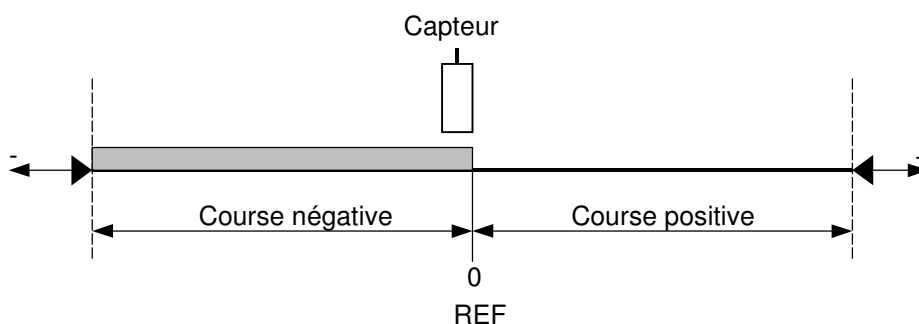
- a) Le référentiel est lié au bâti de la machine, c'est le cas d'une table de palettisation, par exemple.
- b) Le référentiel est fixé par l'opérateur en un point de la course et à un instant choisi par lui, c'est le cas d'un appareil à diviser.

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-600 offre 8 entrées pouvant fonctionner comme signal de référence, "REF INPUT 0..7". L'attribution d'un REF INPUT à un axe se fait dans le menu de configuration.
- Lorsqu'un axe n'utilise pas de contact de référence, on lui attribuera par convention le numéro REF INPUT 8 et l'exécution de la fonction de prise de référence fixera le référentiel initial sans effectuer de mouvement.
- Lorsqu'un axe est utilisé en asservissement, il est possible de faire une référence en utilisant un capteur donnant la position grossière et l'index du codeur donnant la précision. On lui attribuera par convention un numéro REF INPUT de 9 à 17.
- Le numéro REF INPUT 9 impose la référence en bout de course. Elle utilise les capteurs de fin de course câblés sur le module E-600-4 / E-600-7 / E-600-12 en interprétant la condition de FAUTE signalée dans cette situation.
- Lorsqu'un axe est utilisé avec un étage de puissance YASKAWA câblé sur un module E-600-8, la référence avec index est possible en attribuant le numéro REF INPUT 0 pour l'axe X, 1 pour Y, 2 pour Z et 3 pour U. Le numéro REF INPUT 18 permet d'utiliser un capteur de fin de course câblé comme prise de zéro, les même entrées (0 à 3) sont utilisées.

N° REF INPUT configuré	Type de module	N° de l'entrée physique	Remarque
X=0, Y=1, Z=2, U=3	E-600-8	IN(0) à (3)	Capteur de référence + index
0 à 7	Tous, excepté 4 à 7 pour E-600-8	IN(0) à (7)	Sans index
8	Tous	rien	Pas de déplacement
9	E-600-4/7	rien	Fin de course + index
10 à 17	E-600-4/7	IN(0) à (7)	Capteur de référence + index
18	E-600-8	IN(0) à (3)	Fin de course + index

- 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, il faut aussi attribuer le REF INPUT 8 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.

UNIPROG+ prévoit des changements de référentiel par translation. La rotation des coordonnées n'est pas disponible.

Pour faire face à toutes les situations, deux niveaux de translation de coordonnées sont prévus :

- Le **référentiel de base** est obtenu à partir du point de référence par la translation OFFSET, voir Figure 3-2.
- Le **référentiel courant** se déduit du référentiel de base par la translation de vecteur "ORIGINE".

Le référentiel initial n'est pas accessible à l'utilisateur, c'est-à-dire qu'on ne peut donner des coordonnées mesurées directement à partir du point de référence. (Si OFFSET = 0, le référentiel initial et le référentiel de base sont confondus.) Le vecteur OFFSET est introduit par le menu de configuration.

L'utilité de la translation OFFSET est évidente dans les applications où existe un point de travail fixe : perceuses, poinçonneuses, encolleuses... Le vecteur OFFSET part du point de référence et aboutit au point de travail.

Les mouvements absolus commandés dans l'utilitaire JOGGING sont mesurés dans le référentiel de base.

Le **référentiel courant** est introduit par une instruction ORG. Le programmeur peut changer le référentiel courant aussi souvent qu'il en éprouve le besoin. Si aucune instruction ORG 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.

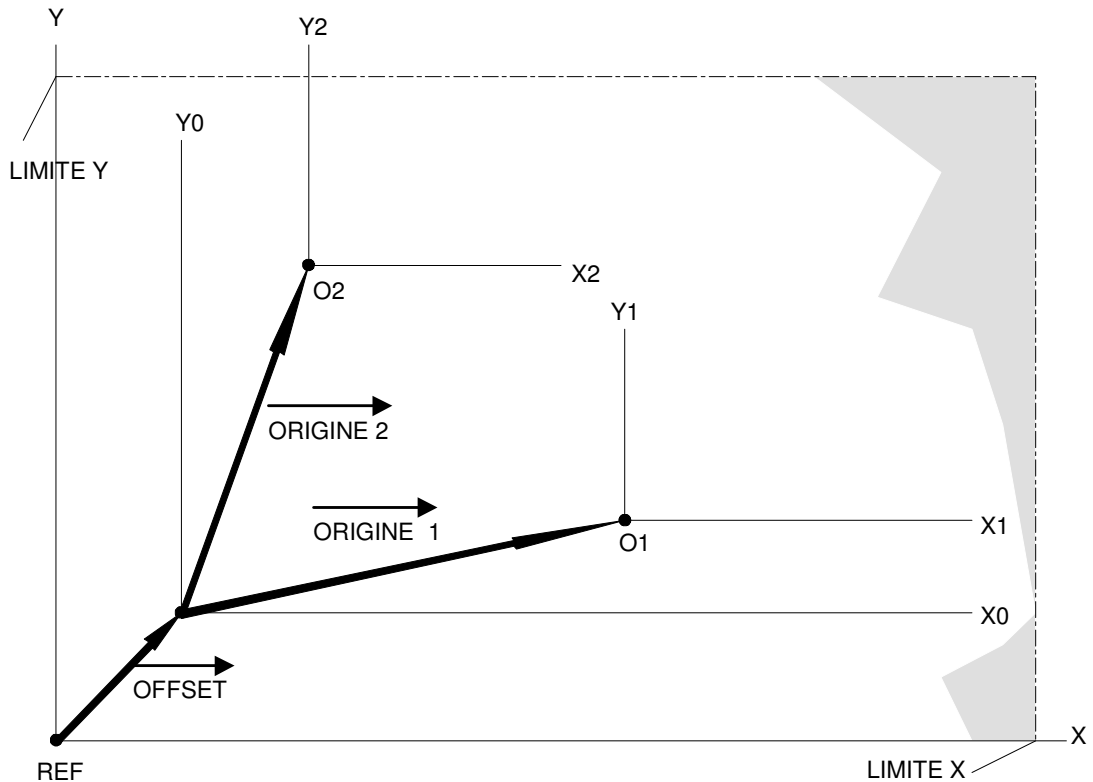


Figure 3-2 : Les référentiels UNIPROG+

### 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, E-600-3, 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.

Les moteurs à 5 phases de BERGER avec leur étage de puissance E600-1/2 demandent 500 ou 1000 impulsions par tour (sélection sur la carte).

Dans le cas d'un axe asservi, SCALEK est 4 fois le nombre d'impulsions produit par le codeur de position lors d'un déplacement de une unité.

Le facteur 4 provient de l'exploitation de tous les flancs des deux signaux du codeur.

### Exemples:

#### a) 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**.

#### b) Plateau-diviseur:

Moteur 5 phases, sélection 1000 pas par tour sur carte de puissance, entraînement direct d'une vis tangente, rapport 1:40, unité : 1 deg.

1000 impulsions pour 1 tour du moteur,  
40'000 impulsions pour un tour du plateau,

$40'000/360 \text{ deg} = 111.1111\dots$  impulsions par deg, **SCALEK = 111.111**.

#### c) Axe linéaire courroie:

Moteur asservi avec codeur 1000 divisions, réducteur 1:10, courroie au pas de 3 mm, poulie menante à 35 dents, unité :1 pouce.

4 \* 1000 impulsions par tour du moteur,  
40'000 impulsions par tour de poulie, 1 tour de poulie = 105 mm = 4.133858 ",

$40'000/4.133858 = 9676.19$  impulsions par pouce, **SCALEK = 9676.19**.

### 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.

Dans le cas d'axes asservis, il peut être avantageux de choisir une vitesse maximale très élevée, la courbe étant toujours limitée par une vitesse programmée beaucoup plus basse. Les rampes deviennent alors quasi-linéaires.

Si  $f_{max}$  est la fréquence maximale des impulsions à l'entrée de l'étage de puissance du moteur pas-à-pas ou 4 fois la fréquence des impulsions du codeur à la vitesse maximale de l'axe asservi, le paramètre DIV est donné par :

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

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

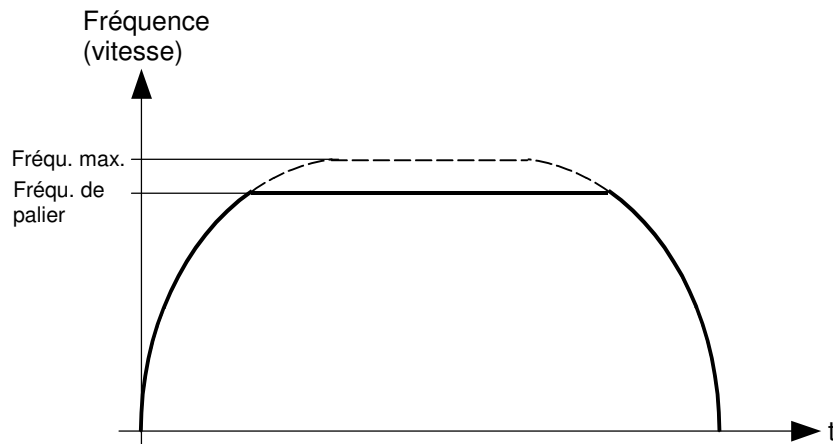
**Exemple:**

Codeur 500 divisions, vitesse maximale 1500 t/min.

1500 t/min = 25 t/sec,

$f_{\max} = 25 \cdot 500 \cdot 4 = 50'000 \text{ Hz}$ ,

**DIV = 11906/50 = 238.**



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

### 3.3.3 Constante d'Accélération KUP, 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 ou de 4 impulsions par division du codeur.

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 entraînements pas-à-pas :

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

Pour entraînements asservis :

$$\text{FEEDK} = 4 * \text{fréquence des impulsions codeur pour une unité de vitesse [kHz]}.$$

#### Exemples:

(Voir les exemples du paragraphe 3.3.1.)

#### a) 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**

**b) On veut exprimer la vitesse en degré/seconde.**

SCALEK = 111.111 , soit 111.111 impulsions pour 1 deg,  
ou 0.111111 kHz pour 1 deg/s, **FEEDK = 0.111111**

**c) On veut exprimer la vitesse en yard/min.**

1 yard = 36 pouces, il faut donc 9676.19 \* 36 impulsions, pour 1 yard, pour 1 yard/min il faut une fréquence de :

$(9.6762 * 36)/60$  kHz, **FEEDK = 5.806** .

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 Déplacement permanent de l'origine, OFFSET

La signification de ce paramètre apparaît clairement dans la Figure 3-2. Il s'exprime naturellement dans l'unité de longueur.

### 3.3.7 Sur-excitation du moteur, 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 modules E600-3 pour moteurs 2 phases, 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.

Avec les modules E600-1 pour moteurs 5 phases BERGER, le signal BOOST remplit plusieurs fonctions selon la sélection sur la carte de puissance ou sur l'adaptateur de connexion, voir la notice E600-1. Avec BOOST = 0, la carte fournit le courant nominal.

Le paramètre BOOST est indifférent pour les axes asservis.

## 3.4 Constantes Cinématiques des axes asservis SERV.

Les commandes E-600 possèdent deux canaux d'asservissement désignés par CHANNEL 0 et CHANNEL 1. Dans la présente version de UNIPROG+, CHANNEL 0 correspond à l'axe X et CHANNEL 1 à Y.

Le canal (CHANNEL) est sélectionné par la touche F1.

### 3.4.1 Asservissement actif SERVO ENB. (Servo Enable)

Avec la valeur 0, l'axe X fonctionne en pas-à-pas, avec la valeur 1, l'axe X est un asservissement de position. Idem pour l'axe Y.

### 3.4.2 Plage de l'erreur de position POS ZONE. (Position Zone)

L'axe est en position lorsque la valeur absolue de l'erreur de position est inférieure ou égale à la zone de position. Un positionnement n'est pas terminé tant que cette condition n'est pas remplie.

Valeurs acceptées: 0... 32'767 [impulsions codeur \* 4].

### 3.4.3 Plage de l'erreur de poursuite ALM ZONE. (Alarm Zone)

Lorsque la valeur absolue de l'erreur de position dépasse la zone d'alarme, l'axe s'arrête et le programme signale l'erreur.

Valeurs acceptées: 0... 32'767 [impulsions codeur \* 4].

### 3.4.4 Valeur du gain HI GAIN, LOW GAIN (High, Low Gain)

Valeur des gains numériques de la boucle de positionnement.

En utilisant le Servo Adapter E600-4, E600-7, le signal d'erreur analogique a la valeur :

$$\begin{aligned} \text{AERR} &= \text{NERR} * 10/500 * \text{GAIN}/2^{16} \\ &= \text{NERR} * \text{GAIN} * 3.052 * 10^{-7} \end{aligned}$$

AERR est la tension d'erreur en Volts,  
NERR est l'erreur numérique en impulsions codeur \* 4,  
GAIN est la valeur du paramètre.

- La commutation du gain est possible uniquement dans l'utilitaire "SERVO".
- Par défaut le petit gain est sélectionné.
- Pour plus de détails, se rapporter à la notice E600-4 / E600-7.

## 3.5 Contrôle de Fermeture

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 + - 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.

Le contrôle de fermeture est possible uniquement avec les numéros REF INPUT configurés de 0 à 7 .

Avec le module E-600-8 le contrôle de fermeture est possible uniquement avec les numéros REF INPUT 4 à 7 (sans la référence sur l'index).

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

Ce chapitre décrit l'utilisation de la commande E-600 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.

La désignation des touches et autres organes de la plaque frontale est donnée par le dépliant à la fin du fascicule.

### 4.1 Mise sous tension de la commande E-600.

Le message suivant apparaît indiquant quelles versions sont contenues dans la commande. Elles peuvent être maintenues dès l'enclenchement de l'appareil en pressant la touche F2.

\* UNIPROG+ V x.xx \*  
INT x.xx NEC x.xx

Après le chargement du logiciel UNIPROG+ le contrôleur E-600 affiche :

ERREUR RAM  
FORMATEE YES NO

**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, la réaction de UNIPROG+ à la mise sous tension est différente selon que la fente "DATA" est occupée ou non par une carte-données. Une carte-données n'ayant pas été écrite par UNIPROG+ équivaut à l'absence de carte.

#### La carte-données est absente :

L'affichage montre pendant quelques secondes le message

LOADING BEE CARD  
0.00, WAIT

suivi du message :

CARD NOT FORMATTED  
press any key

L'opérateur est ainsi informé qu'aucune carte-données n'a été chargée et que toutes les opérations ultérieures se feront à partir du contenu de la mémoire vive (RAM).

En pressant une touche, l'opérateur déclenche l'exécution du programme d'initialisation (Power-

On Programme), voir le chapitre 7, et accède aux menus de base d'UNIPROG+. L'affichage donne le premier menu :

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

**Une carte-données valable est enfichée :**

Le message suivant apparaît

```
LOADING BEE-CARD  
x.xx WAIT
```

x.xx est le numéro de la carte, choisi lors de la sauvegarde.

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 (rouge).

## 4.2 Sélection des Menus

Pour choisir un menu, il faut l'amener à l'affichage à l'aide des touches ↓ ou ↑ pour remonter dans la liste. Les menus suivant défilent:

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

```
2 PROGRAMMING  
EDIT VECT FEED SAVE
```

```
3 DEBUGGING  
TRACE PRT SERV I/O
```

```
4 FILE UTILITIES  
DIR DEL COPY LOAD
```

```
5 CONFIGURATION
SERV MGEN REF CTRL
```

```
6 OTHER
VER          ACCES
```

Sur demande il est possible de construire un menu 0 qui permet d'entrer une liste de paramètres, définies en texte claire, dans le fichier 0. Les textes, le nombre et le type des quantités numériques (entiers ou réels) sont programmés en langage PINX-E, veuillez nous consulter.

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 :

```
NO ACCESS
press any key
```

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, Z, U.

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>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 3.5.
<b>REF. INPUT</b>	Attribution de l'une des 7 entrées IN(O)..IN(7) comme contact de référence de l'axe, attribuer 8 pour un axe sans référence physique et 9 à 17 pour référence avec index, voir paragraphe 3.1.
<b>SWITCH:</b>	Nature du contact ou du détecteur de référence. Entrer "1" pour un contact normalement ouvert, "0" pour un contact normalement fermé, voir paragraphe 3.1.
<b>REF BACK</b>	<b>SPEED</b> Divise la vitesse de prise de référence pour quitter le contact, plus le diviseur est grand, plus grande sera la précision.

### 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, agit en parallèle avec la touche MAN, contact normalement fermé.
<b>EXTERNAL STOP</b>	Avorte les programmes en cours. Contact normalement fermé (élément d'une chaîne de sécurité).

Avec UNIPROG+, de nouveaux paramètres doivent être configurés avant l'utilisation du logiciel :

<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>MANUEL PROGRAMME</b>	Permet d'associer le start d'un programme avec une touche du pavé numérique, et ce depuis le menu jogging (après disparition du curseur). (Valeur de 0 à 100).
<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 valant 10 volts au convertisseur de fréquence. (Valeur de 0 à 99999).
<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).</li></ul>
<b>TOOL L/R INVERT</b>	Permet de rétablir la correspondance du côté de départ de l'outil lors du contourage 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. (Valeur de 0 à 1).
<b>LANGUAGE</b>	Permet d'afficher certain message dans les langues suivantes: 0 = Anglais, 1 = Français, 2 = Allemand. (Valeur de 0 à 2).

### 4.4.4 Configuration des axes asservis (SERVo)

La touche F1 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 le canal auquel est destiné le paramètre

se choisit avec la touches de sélection du canal F1.

- Rappelons que la cinématique de l'axe X correspond au canal 0 et le canal 1 à l'axe Y
- Tous les paramètres de ce menu ont été discutés au chapitre 3. Pour mémoire, rappelons leur signification:

<b>SERVO ENB</b>	Utilisation d'un axe asservi
<b>POS ZONE</b>	Zone de position
<b>ALM ZONE</b>	Zone d'erreur de poursuite
<b>HI GAIN</b>	Grand gain numérique
<b>LOW GAIN</b>	Petit gain numérique

## 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. Un message le signale :

PROG. IN EXECUTION  
press any key

### 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 :

FICH. 0 NON FORMATE  
Presser une touche

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'"incrément" en vigueur dans l'action des touches JOG+ et JOG-. La prise de référence est aussi accessible comme dans le paragraphe 4.5.1.

- **Action des touches JOG+ et JOG-:**

Une pression sur l'une de ces touches effectue un mouvement de 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 l'incrément de mouvement par puissance de 10, de  $10^{-3}$  unités à  $10^3$  unités.

- **Action des touches flèches :**

Elles permettent de déplacer verticalement le pavé situé à l'extrême droite de l'écran. Si le pavé est en bas, le clavier numérique permet l'introduction d'une valeur de l'incrément en lieu et place de la valeur sélectionnée par les touches F3 et F4.

Si le pavé est à la ligne supérieure, on introduit la destination de l'axe dans le référentiel de base. En pressant la touche ENTER, on lance le mouvement qui s'arrête lorsqu'on lâche la touche ENTER ou si la cote est atteinte. Si on lâche prématurément la touche, il faut réintroduire la destination.

- **Fonction spéciale pour division, touche F1:**

Cette fonction permet l'utilisation d'un appareil à diviser sans écrire aucun programme. L'axe commandant la rotation du diviseur est sélectionné par les touches X, Y... et l'unité de mesure adoptée dans le calcul du facteur d'échelle est supposée être le degré.

L'écran montre

DIVIDER	xx
CUR. DIVISION	yy

A l'aide des flèches et du pavé, introduire comme ci-dessus, le nombre de divisions à la ligne supérieure et la division courante à la ligne inférieure (en général on commence à 0). Le diviseur étant ainsi initialisé, les touches JOG+ et JOG- commandent les mouvements du diviseur dans un sens ou dans l'autre.

On peut introduire un nombre négatif comme nombre de divisions, le sens de rotation est alors inversé.

La touche ESC ne retourne pas au menu de base mais au sous-menu JOG ce qui permet de mouvoir les autres axes entre les divisions. On revient à la fonction diviseur par la touche F1 pour continuer le processus de division.

- **Définition des origines Outils**

Cet utilitaire permet, en plus de sa fonction de déplacement manuel, d'enregistrer les origines de 64 outils différents (de 0 à 63). Le nombre d'outils utilisés est configuré par le paramètre "LAST TOOL NB".

- La sélection du numéro d'outil s'effectue avec les touches "F1 et F2"
- Les nouvelles origines sont stockées dans le fichier 0 ouvert automatiquement, (celui-ci contient des FDATA"). Pour consulter aisément ces origines, un tableau d'outil nommé "TOOL" est accessible depuis le menu "MOTION CONTROL"
- Ces nouvelles origines seront interprétées simplement par l'instruction 47, TOOLP si un fichier de contour est associé, ou l'instruction 19, TOOL.
- Un seul argument suit l'instruction TOOL (numéro de l'outil). Deux arguments suivent l'instruction TOOLP (numéro de l'outil et numéro du fichier de contour).
- La touche "PROG LINE" entre dans l'éditeur où celui-ci a été quitté la dernière fois.
- Durant le programme il est possible de rester dans le menu JOGGING, les mouvements sont alors verrouillés.
- Les touches numériques de 0 à 9 démarrent un programme édité en UNIPROG+.
- L'attribution du programme à une touche est définie par le paramètre MANUAL PROGRAMME dans la configuration CTRL. Le numéro du programme exécuté est égal au numéro paramétré plus la touche numérique.

**Par exemple :**

MANUAL PROGRAMME = 90

La touche 0 active le programme 90,

La touche 1 active le programme 91, etc.

Si la valeur configurée dépasse 90 il n'y a plus de programme manuel.

Ces fonctions manuelles sont actives uniquement dans le menu jogging (JOG) et lorsque les LEDs rouge STOP et verte START sont éteintes.



- **Modification de l'incrément du déplacement:**  
Pour modifier l'incrément du déplacement ou effectuer un déplacement absolu en jogging il est nécessaire de rappeler le curseur clignotant à l'aide des touches flèche (basse et haute) et de placer celui-ci en face de la valeur à modifier.
  - Lorsque le curseur disparaît, les fonctions manuelles sont actives.
  - Par défaut le curseur est en mode incrémental.
  - La touche "-" est assimilée à la touche "0".
- **L'enregistrement d'une position s'exécute de la façon suivante:**
  - 1) Approche de la position en jogging.
  - 2) Sélection de l'outil "F1" ou "F2" et de l'axe concerné.
  - 3) Pression de la touche "F5" pour autoriser l'entrée de la valeur de la nouvelle origine sur l'axe et l'outil sélectionnés.
  - 4) Entrée de la valeur par le pavé numérique, puis validation par la touche "ENTER".

**Remarque:**

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.

- **Décalage sur tous les outils**

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" (ajust tout).

AXIS X = 87.677 ADJ ALL .100000
------------------------------------

Dans cette situation chaque pression sur la touche "+" du jogging décale l'origine de tous les outils en ajoutant 0.1 mm et réciproquement avec la touche "-". L'incrément peut être modifié par "F1" ou "F2".

**La modification des coordonnées d'origine peut également se faire depuis le tableau d'outils "TOOL".**

### 4.5.3 Contrôle de fermeture, CLOS

Effectuer sous ordre manuel un contrôle de fermeture comme décrit au paragraphe 3.5 est extrêmement utile pour la mise au point et la vérification d'une installation. Après avoir choisi l'axe, il suffit de presser la touche REF.

Si le contrôle est bon, le message suivant apparaît :

CLOSURE WITHIN GAP press any key
-------------------------------------

dans le cas contraire :

POS ERROR 1.234 press any key
----------------------------------

Si l'axe n'a pas de contact de référence attribué ou utilise le mode index, le contrôle réagit comme s'il était bon. Rappelons qu'après un contrôle de fermeture, l'axe se trouve dans sa

tolérance ou a son point de référence.

#### 4.5.4 Affichage des cotes, DISP

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 4 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.**

## 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 allumé.

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 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 Vitesse 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 sur la Carte-Données (SAVE)

Il faut remarquer que l'opération de sauvegarde écrit dans la carte-données toutes les données de configuration et tous les fichiers ouverts. Le temps d'écriture dépend du nombre de fichiers et de la quantité d'informations différentes du contenu préalable de la carte.

Pour éviter la destruction du contenu d'une carte et permettre de lui attribuer un code d'identification, l'écran affiche :

```
WANT TO SAVE B-CARD
2.23 ? YES NO
```

Le code d'identification qui apparaît dans ce message est celui qui avait été mémorisé dans la zone de travail lors du chargement précédent. Si l'opérateur ne désire pas changer de code, il répond "YES" en pressant F3 ce qui déclenche le processus d'écriture. Dans le cas contraire, il presse F4 :

```
WANT TO CHANGE
NAME ? NO
```

On peut alors introduire un nouveau code (NAME) ou presser F4 pour retourner au menu de

base. L'introduction d'un code produit l'écriture de la carte.

Pendant l'écriture de la carte, opération dont la durée dépend des informations mais peut dépasser 1 minute, l'affichage invite à attendre puis quitte l'opération :

SUCCESSFUL WRITING press any key
-------------------------------------

ou

WRITE ERROR press any key
------------------------------

Une erreur d'écriture indique l'absence de carte ou une carte défectueuse.

## 4.7 Manipulation de Fichiers, FILE UTILITIES

Les utilitaires d'exploitation travaillent toujours sur le contenu de la mémoire vive; pour manipuler des fichiers contenus dans une carte-données, il faut préalablement effectuer un chargement en mémoire, voir paragraphe 4.7.4.

### 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
18	NOT FOUND		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 d'une carte.

#### 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 "press 'CLR' to DELETE" 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 de la Carte-Données dans la Mémoire (LOAD)

Le chargement de la carte détruit la zone de travail en mémoire vive. Pour cette raison, une manoeuvre de sécurité est introduite par le message

WANT TO LOAD B-CARD 1.03 ?      YES NO
---

Le code d'identification de la carte est lu (ici 1.03) et si le chargement n'est pas désiré, F4 ramène le menu de base. En pressant F3 (YES), l'affichage invite à patienter pendant le chargement puis donne l'un des messages suivants:

SUCCESSFUL LOADING  
press any key

ou

CARD NOT FORMATTED  
press any key

Dans le deuxième cas, la carte n'est pas enfichée ou n'a pas été écrite par UNIPROG+ ou encore elle est défectueuse.

## 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:

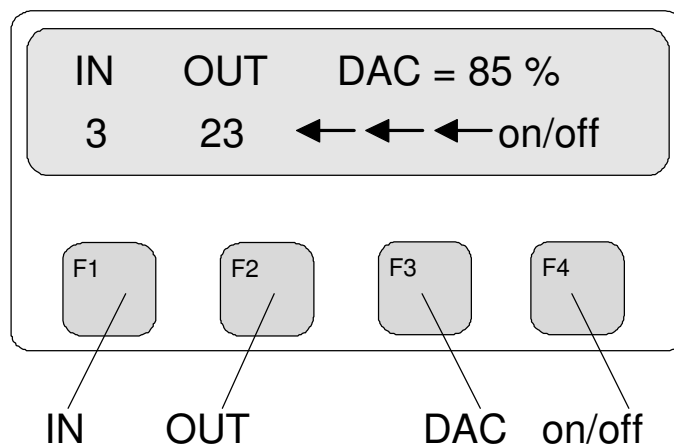


Figure 4-1 : Menu "I/O Control"

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 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 5.6.5); 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.

### 4.8.3 Utilitaire "PRT"

Une imprimante ou un PC connecté sur la liaison RS 232 donne la possibilité de lister les programmes et fichiers UNIPROG+. Il suffit d'entrer le numéro du fichier et l'impression démarre par la touche ENTER.

Le format de l'impression d'une ligne est le suivant :

- Numéro de la ligne
- Numéro du fichier
- Code numérique de l'instruction
- Mnémonique de l'instruction et arguments.

L'imprimante ou PC doit avoir une liaison série initialisée comme suit :

**9600 Bauds, 8 bits, 1 stop bit, pas de parité.**

Le PC peut recevoir des fichiers grâce au "COPY" du "DOS"

#### Exemple:

COPY COM1: <nom\_fichier>

### 4.8.4 Utilitaire "SERVO"

Cette fonction est extrêmement utile pour la mise au point d'une installation avec des axes asservis et pour la recherche de défauts. Pour plus de précision se référer au manuel des modules servo-adaptateurs E-600.

Le choix du canal se fait par la touche F1 "CHA", l'affichage indique le canal sélectionné.

- **En boucle de position fermée** ou contrôle de position, témoin F2 allumé, la partie supérieure de l'écran affiche en permanence l'erreur de poursuite dans l'unité choisie "P-ERR =". Le mode "POSL" (POSition Loop) permet d'ouvrir la boucle de position et de travailler en contrôle de vitesse.

La touche F3 "ENB" (ENaBle) active ou désactive la puissance de l'ampli. Le témoin F3 allumé indique que la puissance est active.

La touche F4 "HIGH" permet de commuter le gain numérique ainsi que le mode de contre réaction tachymétrique, configuré dans le menu 5, sous menu SERV. Le témoin F4 allumé indique que le grand gain et la contre réaction sont actifs.

- **En boucle de position ouverte** ou contrôle de vitesse, témoin F2 éteint, la partie supérieure de l'écran affiche la consigne de vitesse analogique  $\pm 10$  V, "A-OUT [V] =". La fonction "HIGH" n'est plus active et devient l'indication de la durée pour la génération d'un cycle de forme carré symétrique à  $\pm$  la valeur de la consigne de vitesse.

La touche F4 permet d'introduire la durée de la demi-période du cycle, le témoin F4 clignote.

Le cycle commence dès l'introduction au pavé numérique, de la consigne de vitesse, en Volt. Un temps nul permet de générer une consigne continue.

Une pression sur la touche F2 retourne en boucle fermée.

Les touches F1 et F3 conservent les mêmes fonctions.

La touche CLR met la consigne de vitesse à zéro.

La touche STOP rappelle tous les paramètres d'asservissement configurés dans SERV.

Les fonctions "POSL" et "ENB" sont inactives lorsqu'un programme UNIPROG+ est en exécution.

## 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-600 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 par 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 chap.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 "axe " : Accepte un entier 0..5. De 0 à 3 (X, Y, Z, U) pour les déplacements avec unités cartésiennes, 4 et 5 (A et R) pour les déplacements dont les cotes sont spécifiées en degrés pour A (Angle) et en unités de longueur pour R (Rayon).

**Pour les arguments "coordonnées polaires" (4 et 5), seuls les modes d'exécution (mode-e) 2 et 3 sont possibles. Ne pas oublier auparavant de définir le plan de travail au moyen de l'instruction DPATH.**

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 par la dernière commande DPATH. 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.

Utilisation de la notation polaire avec A et R spécifiés dans l'argument <axe>:

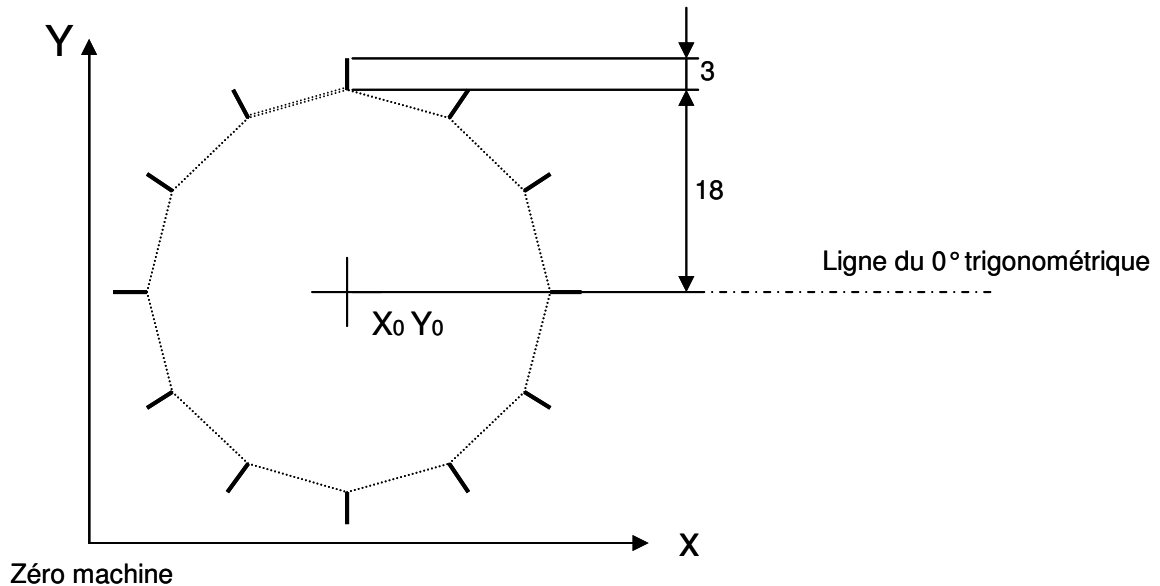
Le rayon R est spécifié depuis l'origine de l'outil courant. L'angle A est celui du rayon formé par la position actuelle de l'outil par rapport à l'origine de l'outil courant et le zéro de la convention trigonométrique. Le décalage d'angle introduit optionnellement par l'instruction ORGA est pris en compte. Si l'outil se trouve à son origine, une instruction POS.. avec l'argument A n'aura pas d'effet, un rayon doit être spécifié.

Remarques :

- L'outil doit être spécifié avec l'instruction TOOL (19)
- L'instruction ORGA (88) pour un décalage de l'origine de l'angle est optionnelle
- L'espace (XY, XZ, etc...) doit être spécifié avec DPATH

Exemple :

- Avec  $X_0 Y_0$  origine de l'outil 0
- Vitesse 0 rapide
- Vitesse 1 usinage
- Gravage (aller-retour) de 12 segments de 3 de long depuis la position midi en tournant dans le sens anti-horaire



```

30 DPATH XY L ; Travail sur X et Y
19 TOOL 0 ; Choix de l'outil 0
88 ORGA 90.0 ; Décalage de 90° pour avoir le 0° à midi
10 POSA R 0 18.0 0 ; Positionnement absolu à midi
10 POSA A 0 0.0 3
63 REP 12 ; 12 fois l'opération
61 CALL 2 ; Descente de l'outil (sous-programme 2 à définir)
14 POSR R 1 3.0 3 ; Gravage vers l'extérieur
14 POSR R 1 -3.0 3 ; Gravage vers l'intérieur
61 CALL 3 ; Remontée de l'outil (sous-programme 3 à définir)
14 POSR A 0 30.0 3 ; Positionnement relatif de 30° dans le sens anti-horaire
65 ENDRP ; Fin de la boucle

```

### 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.

Pour que cette commande fonctionne sans problème, il est nécessaire que les limites de course sur l'axe (STROKE+ et STROKE-) ne soient pas de valeur trop élevée. La relation suivante doit être satisfaite:

$$\text{DIV} * \text{SCALEK} * 1.1 * (\text{STROKE} - | + \text{STROKE} +) \leq 2^{31} - 1$$

### 5.2.2 Test de fermeture :

---

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

---

La fonction du test de fermeture fait l'objet du paragraphe 3.5. 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 5.10.

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 JOG+ et JOG- 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 JOG+ et JOG-, 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>
----	-----	-----------------------	-----------------------

---

- 0) SET PASSE Longueur de la passe (profondeur par défaut = 1mm).  
Cette valeur sera légèrement modifiée pour qu'il y ait un nombre rond de passes (voir Figure 5-1).

$$\text{Longueur effective de la passe: } \frac{\text{total}}{\text{Round}\left(\frac{\text{total}}{\text{passe}}\right)}$$

$$\text{Nombre de passes: } \text{round}\left(\frac{\text{total}}{\text{passe}}\right)$$

- 1) SET GAP Lors du retour rapide, garde avant l'avance lente.  
2) SET DELAY Délai de fond de perçage.  
3) SET 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 Débourrage 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 Débourrage **sans** retour en fin de cycle à la position de départ.  
Mode-d =3 Brise copeaux **sans** 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.

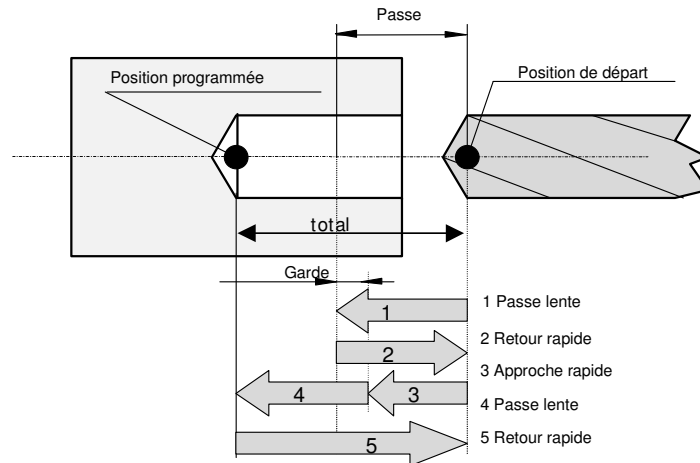


Figure 5-1 : Cycle de perçage

## 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.

**Attention:** Ce mouvement est effectué par interpolation. Il doit donc obéir aux règles définies au paragraphe 8.5.

### 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.

- Attention:**
- Quel que soit le mode, ne pas oublier de définir le plan de travail au moyen de la commande DPATH.
  - Cette commande ne peut être utilisée que dans un seul simultanément à la fois.

### 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.

### 5.2.11 Prise de zéro pour ajustement automatique d'un outil

---

**89 ZTOOL** <axe> <numéro de l'entrée> <sens du déplacement>

---

Cette instruction est utile pour des prises d'origine d'outil automatique (usure d'un outil ou hauteur de pièce variable p. ex.).

Elle permet de récupérer la valeur de la position absolue sur un axe au moment où l'entrée choisie est excitée, et de la déposer dans l'accumulateur. Cette valeur peut ensuite être traitée et déposée dans une variable (par exemple FDATA de l'origine d'outil à modifier).

Pour réaliser ce mouvement de prise d'origine d'outil, l'instruction utilise les valeurs des paramètres de vitesse configurés dans le menu REF de l'axe concerné.

Les numéros d'entrées vont de 0 à 7.

### Exemple:

Système X-Y avec ajustement automatique du zéro d'outil Y au cours du cycle.

5 outils sont définis:

- 0: origine machine
- 1: origine pièce
- 2: origine pour approche rapide près du capteur
- 3: origine pièce courant
- 4: origine capteur

Les outils 0, 1, 2, et 4 sont réglés une seule fois. L'outil 3 est réglé une fois avec les mêmes valeurs que l'outil 1, puis il est modifié par le programme en cours de cycle.

File 1

0 01	19	TOOL	3						<b>; Travail avec l'outil 3:</b>
1 01	10	POSA	X	0	0.0000	0			; (exemple)
2 01	10	POSA	Y	0	0.0000	2			
3 01	19	TOOL	2						<b>; Ajustement automatique:</b>
4 01	10	POSA	X	0	0.0000	0			; Approche capteur (outil 2)
5 01	10	POSA	Y	0	0.0000	2			
6 01	89	ZTOOL	Y	2	0				; Mesure, position dans l'accumulateur
7 01	92	SUBD	2100						<b>; Calculs:</b> Différence avec valeur origine capteur (outil 4)
8 01	91	ADDD	600						; Addition de la correction avec zéro pièce original (outil 1)
9 01	55	STORD	1600						; Stocke dans zéro pièce courant (outil 3)
10 01	60	JMP	1						

## 5.3 Instructions d' 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 est 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 5.6.1.

### 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	SIM0	8	SIM0
9	SIM1	9	SIM1
10	SIM2	10	SIM2
11..15	MARQ(1..5)	11..15	MARQ(1..5)
16..63	IN(16..63)	16..63	OUT(16..63)

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

- Les entrées IN(0..7) sont les fins-de-course ou REF INPUT dont il a été question au paragraphe 4.4.2. Qu'ils soient attribués comme contacts de référence ou non, ils peuvent être échantillonnés par les instructions 20..23.
- Les pseudos I/O, SIM0, SIM1, SIM2, sont les états d'activation des trois programmes simultanés de UNIPROG+.
- Les éléments MARQ(1..5) 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 IN(16..63) 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.

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

O = Open  
C = Closed

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

Les sorties OUT(0) à OUT(7) sont les sorties comprises dans le boîtier E-600 Base et disponibles sur le connecteur de la face arrière, voir chapitre 9.

Le E-600 ND Base ne possède pas de sorties, voir chapitre 10.



### 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	

## 5.4 Vitesses de broches

### 5.4.1 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).

### 5.4.2 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 (BRaK-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 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.

### 5.5.1 Chargement de l'accumulateur

Chargement immédiat de l'accumulateur:

---

<b>50</b>	<b>FLOAD</b>	<b>&lt;nombre réel&gt;</b>
<b>51</b>	<b>ILOAD</b>	<b>&lt;nombre entier&gt;</b>

---

Chargement direct de l'accumulateur:

---

<b>52</b>	<b>LOADD</b>	<b>&lt;adresse&gt;</b>
-----------	--------------	------------------------

---

Chargement indirect de l'accumulateur :

---

<b>53</b>	<b>LOADI</b>	<b>&lt;adresse pointeur&gt;</b>
-----------	--------------	---------------------------------

---

### 5.5.2 Dépose de l'accumulateur

Dépose directe de l'accumulateur :

---

<b>55</b>	<b>STORD</b>	<b>&lt;adresse&gt;</b>
-----------	--------------	------------------------

---

Dépose indirecte de l'accumulateur:

---

<b>56</b>	<b>STORI</b>	<b>&lt;adresse pointeur&gt;</b>
-----------	--------------	---------------------------------

---

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

---

<b>58</b>	<b>INCD</b>	<b>&lt;adresse&gt;</b>
<b>59</b>	<b>DECD</b>	<b>&lt;adresse&gt;</b>

---

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.

### 5.5.4 Sauvegarde d'une variable dans la carte "données" :

---

<b>54</b>	<b>SAVE</b>	<b>&lt;adresse&gt;</b>
-----------	-------------	------------------------

---

La variable écrite à la position "adresse" est sauvée dans la carte EEPROM 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.

## 5.6 Instructions de Contrôle du Programme

### 5.6.1 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.

### 5.6.2 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.

### 5.6.3 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.

### 5.6.4 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.

### 5.6.5 Appel de programmes simultanés :

---

67	SIM1	<adresse>
68	SIM2	<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.

### 5.6.6 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 5.6.1.

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

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

## 5.7 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.

## 5.8 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]

## 5.9 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.

98	<b>FDATA</b>	<nombre réel>	
99	<b>IDATA</b>	<nombre entier>	

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.

## 5.10 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 _
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 PROG/LINE : 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, 2 pour Z, 3 pour U.
- 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', 'STOP' et 'MAN' (ou les entrées associées désignées par la configuration CTRL) et d'autre part par le mode choisi par les touches MOD1, MOD2.

### 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.

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.

Ce mode permet la saturation ou non des vitesses de déplacement. La touche MOD2 fonctionne comme une bascule et passe d'un mode à l'autre:

- Lorsque la LED MOD2 clignote, cela signifie que l'on se trouve dans le cas où les vitesses sont saturées.
- Lorsque la LED MOD2 est allumée, cela signifie que les vitesses ne sont pas saturées.

### 7.2 Fonctions des touches START, PAUSE, 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 'MAN' ou en excitant l'entrée extérieure correspondante. Tous les mouvements sont cependant stoppés selon les rampes de décélération configurées, sans perdre leurs positions. Le témoin START clignote et le témoin MAN 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-600, 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.
- L'exécution d'un programme est contrôlée d'une part par les touches 'START', 'STOP' et 'MAN' (ou les entrées associées désignées par la configuration CTRL) et d'autre part par le mode choisi par les touches MOD1, MOD2.

## 7.3 Gestion des fautes

Trois conditions de fautes sont affichée 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),
3. L'erreur de poursuite ou "alarm zone".

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

**AXIS a FAULT**  
**press STOP**

**STROKE a TOO LARGE**  
**press STOP**

**AXIS a ALARM ZONE**  
**press STOP**

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

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

**AXIS a FAULT**  
**JOGGING +/- → ESC**

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 de la série E-600 sont capables d'engendrer des vecteurs dans tous les espaces sous-tendus par X, Y, Z, U. 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.

UNIPROG+ peut engendrer des vecteurs et des contours dans 6 espaces cartésiens différents :

0	1	2	3	4	5
XY	XZ	XU	YZ	YU	ZU

### 8.2 Génération d'un Vecteur

---

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

---

L'instruction DPATH (**D**efine **P**ATH) 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.

---

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

---

L'instruction ORGP (**O**RGin **P**ath) 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 exécuté 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-600 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

32	LINA	<axe>	<coordonnée>	<mode-e>
33	LINR	<axe>	<composante>	<mode-e>

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).

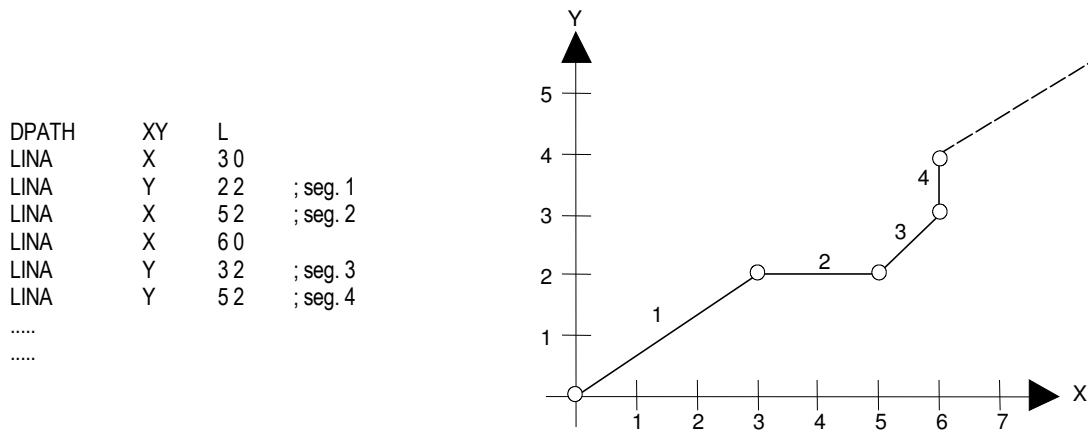


Figure 8-1 : Contour avec diamètre nul

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.

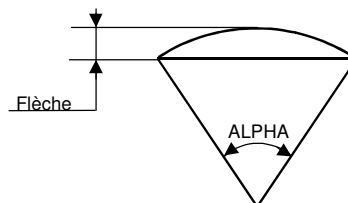
### 8.3.2 Définition d'un arc de cercle

---

#### 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.

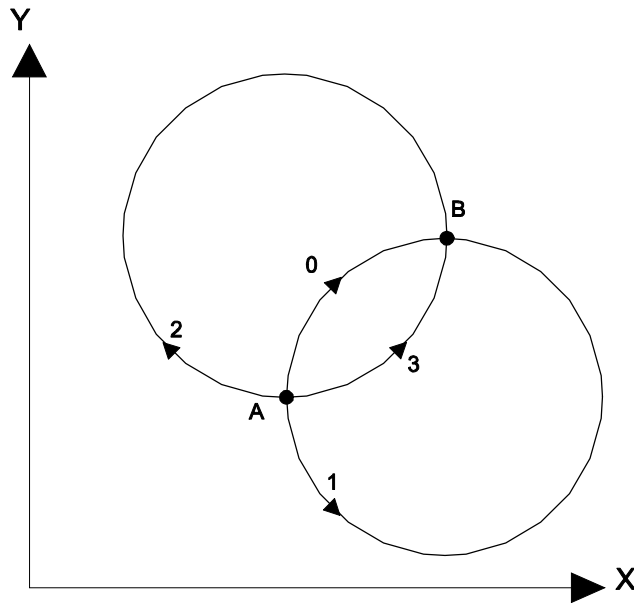


Figure 8-2 : Modes de rotation

---

36	<b>CIRA</b>	<axe>	<coordonnée>	<mode-e>
37	<b>CIRR</b>	<axe>	<composante>	<mode-e>

---

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.

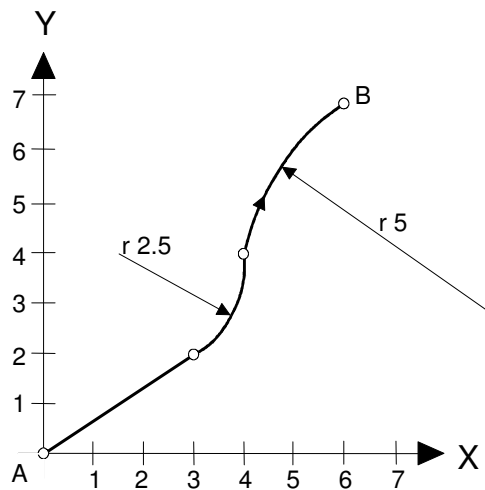


Figure 8-3 : Exemple de contour

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:

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

### Détermination de $KMUL_{max}$ :

- Pour chacun des axes sélectionnés par l'instruction DPATH, on calcul la constante  $KMUL(axe) = SCALEK(axe) * DIV(axe)$
- La constante  $KMUL_{max}$  est la plus grande des constantes  $KMUL(axe)$ , soit  $KMUL_{max} = \max(KMUL(axe))$

### 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

$KMUL(Y) = 200'000$

$KMUL_{max} = 300'000$

$S < 223 \text{ mm}$

## 8.6 Limitations du nombre total de segments

- Mémoire vive disponible : 30'000 bytes
- Place pour un contour :  $16 + S \cdot (3 + 6 \cdot d)$  bytes  
S = nombre de segments  
d = dimension de l'espace.

**Exemple** Contour rectangulaire fermé avec arrondis sur les 4 angles, résolution circulaire 5 degrés, espace XY.

- $S = 4 + 4 \cdot 18 = 76$  segments (4 côtés et 4 quarts de cercle à 18 segments)
- Place requise :  $16 + 76 \cdot (3 + 6 \cdot 2) = 1156$  bytes.

Le tableau suivant donne le nombre maximum de segments en fonction de la dimension de l'espace:

Dimension	Nb max de segments
2	2021
3	1443
4	1122

## 8.7 Exécution des Contours

---

### 48 PATH <vitesse>

---

Cette instruction était anciennement l'instruction 38. L'argument, numéro de fichier, disparaît, le fichier est chargé par la dernière instruction TOOLP.

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>

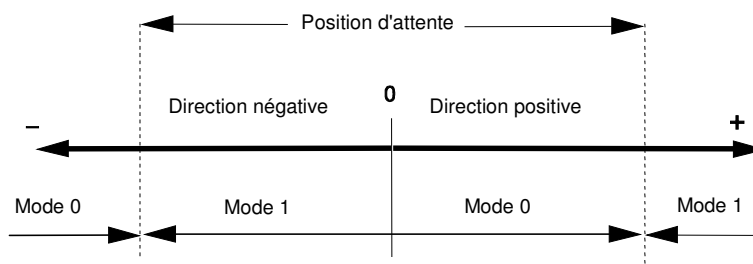
---

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.



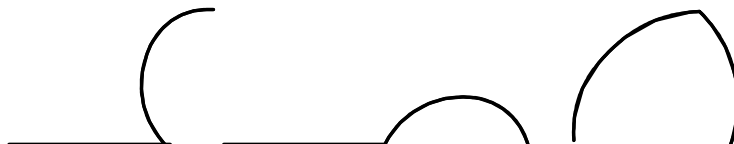
**Figure 8-4 : Position d'attente selon le sens de déplacement**



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

UNIPROG+ ne résout pas tous les cas de contournage, 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.



## 8.9 Affichage des erreurs de contour

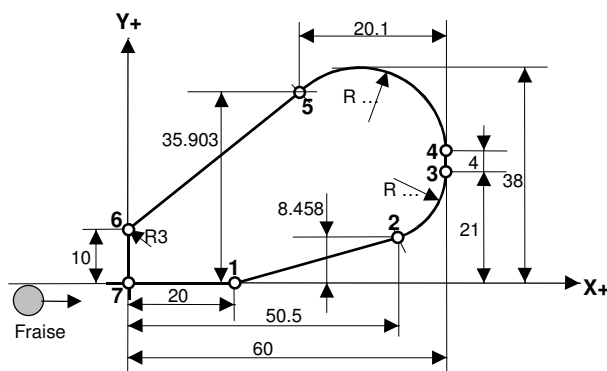
Durant la passe de calcul du contour des ERREURS numérotées peuvent apparaître en cas de données erronées. Ces erreurs stoppent le calcul.

- 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 instruction 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 suivrent 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.

## 8.10 Exemple de contournage

### Tableau des outils

ORIGIN X	TOOL 0	?
ORIGIN Y	TOOL 0	?
ORIGIN Z	TOOL 0	?
ORIGIN U	TOOL 0	?
DIAMETRE	TOOL 0	?
ORIGIN X	TOOL 1	13
ORIGIN Y	TOOL 1	54
ORIGIN Z	TOOL 1	?
ORIGIN U	TOOL 1	?
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

```

## 8.11 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	52
36	<b>CIRA</b>	Axe	Coordon.	Mode-e		Definit mvt circulaire abs.	53
37	<b>CIRR</b>	Axe	Compos.	Mode-e		Definit mvt circulaire rel.	53
30	<b>DPATH</b>	espace	Outil g-d			Définit plan de travail	50
66	<b>ENDP</b>					Fin d'un contour	56
32	<b>LINA</b>	Axe	Coordon.	Mode-e		Définit segment droit absolu	51
33	<b>LINR</b>	Axe	Compos.	Mode-e		Définit segment droit relatif	51
40	<b>ORGP</b>	Axe	Pos. Abs.			Définit l'origine d'un contour	50
48	<b>PATH</b>	Vitesse				Exécution d'un contour	56
31	<b>PCOMP</b>	Fichier				Interprétation d'un fichier contour	54
42	<b>POINT</b>	Axe	Pos. Angul.	Mode-e		Liaison de segments av. arrondis	52
34	<b>RAD</b>	Mode-r	Rayon			Def. Rayon et dir. rotation	52
47	<b>TOOLP</b>	N° outil	N° fichier			Fix. réf. et interprétation contour	54
82	<b>WAITP</b>	Axe	Vitesse	position	Mode-w	Déf. Position d'attente	56

## 9 Raccordements de la commande E-600

Ce chapitre décrit les connexions nécessaires à l'utilisation d'UNIPROG+.

### 9.1 Introduction

La logique de commande est composée du clavier, de la carte CPU INTEL et de la carte cinématique NEC sur laquelle se raccordent les modules. Sur cette carte, il faut distinguer les axes physiques 0, 1, 2, 3 (disponibles sur les connecteurs internes) des axes symboliques X, Y, Z, U.

Avec l'utilisation du logiciel UNIPROG+ les axes physiques (connexions internes) ont comme correspondances: X=0, Y=1, Z=2, U=3.

Sur les commandes E-600, les connecteurs destinés aux axes peuvent être de nature différente selon les modules utilisés.

Une erreur de connexion du câble plat des modules, peut causer des dommages important.

Les raccordements des modules de puissance les plus usuels sont rapidement décrits. Pour plus de précision se référer au manuel des modules.

Seul le brochage externe est détaillé, toutefois il est possible de modifier l'attribution des modules par rapport aux axes physiques en croisant les câbles plats internes.

### 9.2 E-600-base, alimentation, logique et panneau

La commande est équipée de 3 fusibles temporisés:

- Dans l'interrupteur, 5A sur le 220 V (Entrée du câble secteur)
- Sous le capot plastique supérieur de la commande, 8 A sur le 70 V et 12,5 A sur le 24 V.

### 9.2.1 Connecteur I/O

Le connecteur désigné I/O regroupe les entrées et sorties disponibles dans la commande E-600 de base, les signaux analogiques et les tensions d'alimentation.

Broche	Signal et Destination
A	0 V, retour pour les sorties <sup>2)</sup>
B	UNIPROG+ OUT(4), active haut, 24 V, 1 A
C	UNIPROG+ OUT(5), active haut, 24 V, 1 A
D	UNIPROG+ OUT(6), active haut, 24 V, 1 A
E	UNIPROG+ OUT(7), active haut, 24 V, 1 A
F	Analog. Ground pour ADC ou WATCHDOG <sup>1)3)</sup>
G	Signal sortie DAC 0..10 V
H	UNIPROG+ OUT(0), active haut, 24 V, 1 A
J	+ 5 Volt, max. 50mA
K	Signal entrée ADC
L	fin-de-course ou entrée active haute IN(2)
M	fin-de-course ou entrée active haute IN(6)
N	fin-de-course ou entrée active haute IN(3)
P	fin-de-course ou entrée active haute IN(7)
R	Analog. Ground pour DAC <sup>1)</sup>
S	UNIPROG+ OUT(1), active haut, 24 V, 1 A
T	UNIPROG+ OUT(2), active haut, 24 V, 1 A
U	UNIPROG+ OUT(3), active haut, 24 V, 1 A
V	+ 24 V non régulée

**Tableau 9-1 : E600, connecteur I/O, Burndy 19 broches**

- 1) Les "analogique ground" ne supportent qu'un faible courant et ne doivent être utilisés que comme retour des circuits analogiques.
- 2) Chaque sortie peut délivrer 400 mA mais le courant total des 8 sorties ne doit pas dépasser 4 A.
- 3) Dès le numéro de série 707 du circuit I/O, cette broche devient la sortie de contrôle de programme "WATCHDOG". Cette sortie 24 V DC 500 mA est active si le programme système suit un déroulement normal, certaines perturbations désactivent cette sortie.

### 9.2.2 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.

### 9.2.3 Connecteur RS 232

Le connecteur désigné RS 232 permet de connecter une interface reconnue par la commande E-600.

- Compatible IBM PC ou AT et logiciel UNICOM ou APEX A-600
- Imprimante série ou "COPY" du DOS pour la fonction "PRT"

## 9.3 Compatibilité avec E-500

Dans la situation où un E-500 remplace un E-600, il faut retirer sur le connecteur I/O les broches H, S, T, U, effectuer un pont entre les broches H et K et vérifier la correspondance des entrées.

Dans la situation où un E-600 remplace un E-500, il faut supprimer le pont entre les broches H et K et vérifier la correspondance des entrées.

## 9.4 Module E-600-1, drivers BERGER, pour moteurs 5 phases

Le câble plat est raccordé sur le connecteur interne J2, J3, J4 ou J5

Broche	Signal et Destination
A	Enroulement W1A jaune (moteur original BERGER)
B	Enroulement W1E blanc
C	Enroulement W2A bleu
D	Enroulement W2E rouge
E	Enroulement W3A orange
F	Enroulement W3E vert
G	Enroulement W4A gris
H	Enroulement W4E noir
J	Enroulement W5A brun
K	Enroulement W5E violet
L	Entrée active haute 24 V, IN(i)
M	Entrée active haute 24 V, IN(i+4)
N	Alimentation des détecteurs 24V
P	Commun des détecteurs 0V
R,S,T,U	N.C.
V	Terre de protection

**Tableau 9-2 : E600-1, connecteur, Burndy 19 broches**

i= numéro de l'axe physique voir 9.6.

## 9.5 Module E-600-2, contrôle de rotation, drivers BERGER

Le câble plat est raccordé sur le connecteur interne J2, J3, J4 ou J5

- Connecteur, Burndy 19 broches, identique au module E-600-1.

Broche	Signal et Destination
1	Canal A complémentaire
2, 7	+ 5 V
3, 4, 8	0 V
5	Canal B
6	Canal A
9	Canal B complémentaire

**Tableau 9-3 : E600-2, connecteur, Sub-D 9 broches pour codeur**

- La faute est gérée par UNIPROG+.
- Un deuxième câble plat relie le connecteur Sub-D 9 pôles au translateur BERGER.

## 9.6 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"

Le câble plat est raccordé sur le connecteur interne J2, J3, J4 ou J5

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, IN(i)
F	Entrée active haute 24 V, IN(i+4)
G	Alimentation des détecteurs 24V
H	Commun des détecteurs 0V

**Tableau 9-4 : E600-3, connecteur, Burndy 8 broches**

i = numéro de l'axe, selon le raccordement interne du câble plat.

Connecteur interne	J2	J3	J4	J5	J6	J7
i	0	1	2	3	0	1
Axe physique	0	1	2	3		
Chanel					0	1

**Tableau 9-5 : Numérotation des axes**

### 9.6.1 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 9-6 : E600-3, réglage du courant**

## 9.7 Module E-600-13, pour asservissement

Ce module est destiné au raccordement d'un étage de puissance extérieur pour servo-moteur. La commande E-600 supporte au maximum deux modules E-600-13.

Il est disponible pour les axes 0, 1 respectivement canal 0, 1. Voir manuel E-600-13 : SERVO-ADAPTEUR MODULE

Le câble plat est raccordé sur le connecteur interne J7 ou J8



Broche J1	Signaux codeur
1 /CHA	Canal A complémentaire
2 /CHB	Canal B complémentaire
3 /INDEX	Canal INDEX complémentaire
4 +5V	Alimentation +5 V
5 SHIELD	Blindage
6 CHA	Canal A
7 CHB	Canal B
8 INDEX	Canal INDEX
9 0 V	Retour alimentation 0 V

**Tableau 9-7 : E600-13, connecteur J1**

Broche J2	Signaux de contrôle
1 -12V	Alimentation -12 V
2 ENOUT+	Sortie enable, collecteur de l'opto-coupleur
3 FAULT+	Entrée faute
4 FLS	Limite course positive
5 TACHO+	Entrée tachymétrique positive
6 ERROR	Sortie analogique +/-10V
7 0V	0V
8 SHIELD	Blindage
9 +12V	Alimentation +12V
10 ENOUT-	Sortie enable, émetteur de l'opto-coupleur
11 FAULT-	Sortie faute, émetteur de l'opto-coupleur
12 BLS	Limite course négative
13 TACHO-	Entrée tachymétrique positive
14 0V	0V
15 0V	0V

**Tableau 9-8 : E600-13, connecteur J2**

## 9.8 Module E-600-8, pour YASKAWA

Ce module est destiné au raccordement d'étage de puissance de type YASKAWA série R, SIGMA, DR2 en contrôle de position. Ce matériel (moteur et amplificateur AC) comporte sa propre boucle de position. La commande E-600 et les logiciels supportent 4 modules YASKAWA.

Le câble plat est raccordé sur le connecteur interne J2, J3, J4 ou J5

Voir manuel E-600-8 : YASKAWA MODULE

## 9.9 Module E-600-9, pour translateur pas-à-pas

Ce module se résume à un simple câble il est destiné au raccordement d'un étage de puissance extérieur pas-à-pas.

Le câble plat est raccordé sur le connecteur interne J2, J3, J4 ou J5

PIN	SIGNAL E-600-9
1	+12V
2	FAULT Faute
3	/BOOST Surexcitation
4	/PULS Impulsion
5	/DIR Direction
6	GND 0V, Retour des signaux logiques
7	GND
8	GND
9	Entrée active haute 24 V, IN(i+4)

**Tableau 9-9 : E600-9, connecteur, Sub-D femelle 9 broches**

i= numéro de l'axe voir Tableau 9-5

## 9.10 Module E-600-11, Pour entrée codeur

Ce module exige un codeur de position, 5 [V], sortie RS 422, utilisé uniquement avec un logiciel adapté. La commande E-600 supporte au maximum deux modules E-600-7.

Broche	Signal et Destination
1	Canal A
2	Canal B
3	Canal C
4	Alimentation +5V
5	Terre (blindage)
6	Canal A complémentaire
7	Canal B complémentaire
8	Canal C complémentaire
9	0V

**Tableau 9-10 : E600-11, connecteur, Sub-D 9 broches pour codeur**

## 10 Raccordements de la commande E-600-ND

Ce chapitre décrit les Connexions nécessaires à l'utilisation d'UNIPROG+. Voir l'introduction au chapitre 9.1.

### 10.1 E-600-ND-BASE

Sur les commandes E-600-ND, les connecteurs destinés aux axes peuvent être de nature différente selon les étages de puissance à raccorder.

- La commande est équipée d'un fusible accessible après avoir retiré la tôle d'habillage zinguée: 1,6A 24 V.
- Un connecteur 3 pôles permet d'alimenter la commande avec une tension continue non régulée de 24 Volts.
- La commande E-600-ND ne possède pas de sorties. Celles-ci doivent être prises sur des modules de sorties E-500-ODC1.
- Pour utiliser les sorties 0 à 7 UNIPROG+, le module de sortie doit porter l'adresse la plus haute (c,c,c,c), voir table 4.5

#### 10.1.1 E-600-ND connecteur I/O

Le connecteur désigné I/O regroupe les entrées et sorties analogiques disponibles dans la commande E-600-ND ainsi que les entrées destinées à la prise de références (IN 0 à 7).

PIN	SIGNAL	PIN	SIGNAL
1	IN6	9	GND
2	IN2	10	ADC
3	IN7	11	AGND(ADC)
4	IN3	12	AGND(DAC)
5	+5V	13	DAC(OUT)
6	IN5	14	IN1
7	IN4	15	IN0
8	GND		

*Tableau 10-1 : E600-ND, connecteur I/O, Sub-D femelle 15 broches*

Les "analogue ground AGND" ne supportent qu'un faible courant et ne doivent être utilisés que comme retour des circuits analogiques

#### 10.1.2 E-600-ND 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.

#### 10.1.3 E-600-ND connecteur RS 232

Le connecteur désigné RS 232 permet de connecter une interface reconnue par la commande E-600.

- Compatible IBM PC ou AT et logiciel UNICOM ou APEX A-600
- Imprimante série ou "COPY" du DOS pour la fonction "PRT"

## **10.2 Module E-600-13N, pour asservissement**

Les connecteurs codeur et signaux sont disponibles sur le module E-600-13N. Ils sont destinés au raccordement d'un étage de puissance pour servo-moteur, son fonctionnement est identique au module E-600-13, voir chap. 9.7.

## **10.3 Module E-600-8N, pour YASKAWA**

Le connecteur désigné YASKAWA, disponible sur le module E-600-8N est destiné au raccordement d'étage de puissance de type YASKAWA série R, son fonctionnement est identique au module E-600-8, voir chap. 9.8.

## **10.4 Module E-600-9N, pour translateur pas-à-pas**

Le connecteur, désigné STEPPER est destiné au raccordement d'un étage de puissance pas-à-pas, voir chap. 9.9.

# 11 Récapitulation UNIPROG+

## 11.1 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	44
86	ANGLE	Vitesse	Valeur	Mode-e		Dépl. Val. angle	38
22	BRIN0	entrée	adresse			Branch. Si entrée fausse	39
23	BRIN1	entrée	adresse			Branch. Si entrée vraie	39
24	BRM	adresse				Branch. si accu < 0	44
27	BRNZ	adresse				Branch. si accu <> 0	44
25	BRP	adresse				Branch. si Accu >= 0	44
26	BRZ	adresse				Branch. si Accu = 0	44
61	CALL	adresse				Appel sous-programme	43
35	CDEF	Flèche max.				Def. Valeur segmentation	52
36	CIRA	Axe	Coordon.	Mode-e		Definit mvt circulaire abs.	53
37	CIRR	Axe	Compos.	Mode-e		Definit mvt circulaire rel.	53
18	CLOS	Axe	vitesse			Test fermeture	35
46	CORR	Vitesse				Déplacement linéaire	37
95	CPL	N° sortie				Compl. Etat sortie	41
59	DECD	adresse				Décrémentation directe (adresse)	42
94	DIVD	adresse				Division directe accu	44
30	DPATH	espace	Outil g-d			Definit plan de travail	50
62	END					Fin de programme et routines	43
66	ENDP					Fin d'un contour	56
65	ENDRP					Fin d'une boucle de répétition	43
98	FDATA	réel				Définition d'un nombre réel	44
50	FLOAD	réel				Charge Accu Immédiat, réel	42
99	IDATA	Entier				Définition d'un nombre entier	44
51	ILOAD	Entier				Charge Accu Immédiat, entier	42
58	INCD	adresse				Incrémentation directe (adresse)	42
60	JMP	adresse				Saut incondtionnel	43
32	LINA	Axe	Coordon.	Mode-e		Definit segment droit absolu	51
33	LINR	Axe	Compos.	Mode-e		Definit segment droit relatif	51
52	LOADD	adresse				Charge accu direct (adresse)	42
53	LOADI	pointeur				Charge accu indirect (pointeur)	42
85	MOTOR	N° moteur	Vitesse rot.			Déf. Vitesse rotation moteurs	41
93	MULD	adresse				Multiplic. direct accu (adresse)	44
90	NOP					No Operation	44
28	OFF	Sortie				Sortie à 0	39
29	ON	sortie				Sortie à 1	39
88	ORGA	Déc. Angul.				Décalage angulaire	38
40	ORGP	Axe	Pos. Abs.			Definit l'origine d'un contour	50
48	PATH	Vitesse				Exécution d'un contour	56
31	PCOMP	Fichier				Interprétation d'un fichier contour	54
84	PECK	Axe	Vit. lente	Pos. Perc.	Mode-d	Cycle de perçage	36
42	POINT	Axe	Pos. Angul.	Mode-e		Liaison de segments av. arrondis	52
10	POSA	axe	vitesse	Coordon.	Mode-e	Positionnement absolu imméd.	32

11	<b>POSAD</b>	axe	vitesse	adresse	Mode-e	Positionnement absolu direct	32
12	<b>POSAI</b>	axe	vitesse	pointeur	Mode-e	Positionnement absolu indirect	32
14	<b>POSR</b>	axe	vitesse	Dépl.	Mode-e	Positionnement relatif Imméd.	33
15	<b>POSRD</b>	axe	vitesse	adresse	Mode-e	Positionnement relatif direct	33
16	<b>POSRI</b>	axe	vitesse	pointeur	Mode-e	Positionnement relatif indirect	33
34	<b>RAD</b>	Mode-r	Rayon			Def. Rayon et dir. rotation	52
87	<b>RADIUS</b>	Vitesse	Valeur	Mode-e		Rayon de dépl. (coord. polaires)	38
17	<b>REF</b>	axe				Déf. Point de référence	35
63	<b>REP</b>	Nb de fois				Répétition, immédiat	43
64	<b>REPD</b>	Adresse				Répétition, directe	43
54	<b>SAVE</b>	Adresse				Sauvegarde en EEPROM	42
83	<b>SET</b>	N° param	Val. param.			Attribution de variables	36
67	<b>SIM1</b>	adresse				Appel 1 <sup>er</sup> programme simultané	43
68	<b>SIM2</b>	adresse				Appel 2 <sup>ème</sup> programme simultané	43
57	<b>SPVEL</b>	Rot/min				Vitesse broche	41
55	<b>STORD</b>	adresse				Sauvegarde Accu, Directe	42
56	<b>STORI</b>	pointeur				Sauvegarde Accu, Indirecte	42
92	<b>SUBD</b>	adresse				Soustraction Accu, Directe	44
13	<b>TEACH</b>	axe	vitesse	adresse		Corr. Pos. par apprentissage	35
19	<b>TOOL</b>	N° outil				Définition référentiel outil	34
47	<b>TOOLP</b>	N° outil	n° fichier			Fix. réf. et interprétation contour	54
81	<b>TPING</b>	Axe	Pas	pos. fin		Définition d'un taraudage	37
70	<b>WAIT</b>	Temps				Temporisation, adr. immédiat	44
20	<b>WAIT0</b>	Entrée				Attente entrée <> 0	39
21	<b>WAIT1</b>	Entrée				Attente entrée <> 1	39
71	<b>WAITD</b>	adresse				Temporisation, adr. directe	44
82	<b>WAITP</b>	Axe	vitesse	position	Mode-w	Déf. Position d'attente	56
89	<b>ZTOOL</b>	Axe	n°entrée	direction		Prise de zéro pour ajustement automatique d'un outil	38

## 11.2 Entrées-Sorties

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	SIM0	8	SIM0
9	SIM1	9	SIM1
10	SIM2	0	SIM2
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..63	IN(16..63)	16..63	OUT(16..63)

## 11.3 Divers

<b>Axes:</b> X = 0 Y = 1 Z = 2 U = 3										
<b>Canal:</b> X = 0 Y = 1 touche F1										
<b>Espaces:</b>	0	1	2	3	4	5				
	XY	XZ	XU	YZ	YU	ZU				
<b>Pause Flag:</b> Touche et Led F1			<b>Insert:</b> touche F3			<b>Delete:</b> touche F4				