

E

I

P

E-500

Français

UNIPROG E500
Programme universel

Version: 1990

1. INTRODUCTION	1
2. ORGANISATION DES ZONES DE STOCKAGE	1
2.1. Mémoires de la Commande E-500	1
2.2. L'Espace de Stockage Accessible à l'Utilisateur	2
3. GEOMETRIE ET CINEMATIQUE	1
3.1. Références de Position	1
3.2. Référentiel Initial et Limites de Course	1
3.3. Changement de Référentiels	2
3.4. Constantes Cinématiques	3
3.4.1. Le facteur d'échelle SCALEK	4
3.4.2. Fréquence Maximale SPMX	4
3.4.4. Constante de Rampe linéaire KLIN	5
3.4.5. Constante de Vitesse FEEDK	5
3.5. Contrôle de Fermeture	6
4. MODE OPERATOIRE DU CLAVIER ET FONCTION DE UNIPROG	1
4.1. Mise sous tension de la commande E-500.	1
4.2. Sélection des Menus	2
4.3. Menu "ACCESS"	3
4.4. Menu "CONFIGURATION"	3
4.4.1. Configuration des Générateurs de Mouvement (Motion GENERators) ...	3
4.4.2. Configuration de la Prise de Référence (REFERence)	4
4.4.3. Configuration des Entrées de Contrôle (ContROl)	4
4.5. Menu "MOTION CONTROL"	4
4.5.1. Prise de Référence, REF	5
4.5.2. Mouvements Manuels, JOG	5
4.5.3. Contrôle de fermeture, CLOS	6
4.5.4. Affichage des cotes, DISP	6
4.6. Menu "PROGRAMMING".	6
4.6.1. Programmes à Exécuter (start VECTors)	7
4.6.2. Vitesse Présélectionnées (FEED rates)	7
4.6.3. Sauvegarde sur la Carte-Données (SAVE)	7
4.7. Manipulation de Fichiers, FILE UTILITIES	8
4.7.1. Répertoire, (DIRectory)	8
4.7.2. Effacement d'un Fichier (DElete)	8
4.7.3. Copie d'un Fichier (COPY)	9
4.7.4. Chargement de la Carte-Données dans la Mémoire (LOAD)	9
4.8. Menu "DEBUGGING"	9
4.8.1. Utilitaire "TRACE"	9
4.8.2. Utilitaire "I/O" control	10
4.8.3. Utilitaire "PRINT"	11
5. INSTRUCTIONS DU LANGAGE UNIPROG	1
5.1. Instructions de Positionnement	1
5.2. Autres instructions Cinématiques	2
5.3. Instructions Entrées/Sorties	3
5.4. Instructions d'Affichage	5
5.5. Manipulation de Quantités Numériques	5
5.6. Instructions de Contrôle du Programme	6
5.7. Instructions de Temporisation	8
5.8. Instructions Arithmétiques	8
5.9. NOP et "Pseudo-Instructions"	9
5.10. Marqueur de Pause	9

6. L'EDITEUR UNIPROG	1
6.1. Examen d'un programme	1
6.2. Modification du Contenu d'une ligne	1
6.3. Insertion et Suppression d'une ligne	2
6.4. Mise en Place du Marqueur de Pause	2
7. EXECUTION DES PROGRAMMES	1
8. GENERATION DE VECTEUR ET CONTOURNAGE	1
8.1. Introduction: Possibilités et Définition de l'Espace	1
8.2. Configuration des Paramètres Cinématiques Curvilignes	1
8.3. Génération d'un Vecteur	2
8.4. Définition géométrique d'un Contour	3
8.4.1. Définition d'un segment linéaire	3
8.4.2. Définition d'un arc de cercle	5
8.4.3. Arc d'hélice ou d'Hyperhélice	7
8.5. Interprétation des Fichiers de Définition des Contours	8
8.6. Exécution des Contours	8
9. RACCORDEMENTS DE LA COMMANDE E-500	1
9.1. Commande E-502 B, 2 axes, moteurs 5 phases, drivers BERGER	1
9.2. Commande E-504 B, 4 axes, moteurs 5 phases, drivers BERGER	3
9.3. Commande E-502 S, 2 axes, moteurs 2 phases, drivers SDM 50	3
9.4. Commande E-504 S, 4 axes, moteurs 2 phases, drivers SDM 50	4

1. INTRODUCTION

Les commandes de positionnement et de contournage de la série E-500 sont destinées à l'équipement de manipulateurs et de petites machines spéciales et résolvent les problèmes de cinématique et d'automatisme les plus variés. Elles sont actuellement disponibles en version 2 et 4 axes pour moteurs pas-à-pas hybrides à 2 et 5 phases.

E.I.P. SA a créé le langage de programmation PINX-500 et l'outil de développement APEX afin de répondre de façon efficace aux situations les plus variées et les plus exigeantes. La pratique a montré que nombre de fonctions se retrouvent dans la plupart des applications mais que les séquences de travail sont infiniment variables. D'autre part, la maîtrise du langage PINX-E demande un effort certain et l'utilisation d'un PC comme outil de développement n'est pas souhaitable dans un environnement industriel.

Ces considérations ont conduit E.I.P. au programme **UNIPROG** qui offre un ensemble de fonctions utilitaires et la possibilité d'écrire des programmes performants simplement à partir du panneau de la commande. 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.

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.

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 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 moteurs et des signaux de contrôle. Pour des informations plus détaillées se référer au manuel de service.

Le chapitre 4 décrit le mode opératoire du clavier à partir de la mise sous tension.

Le chapitre 8 est entièrement dédié au problème de la génération de vecteurs et du contournage (interpolation linéaire et circulaire).

2. ORGANISATION DES ZONES DE STOCKAGE

2.1. Mémoires de la Commande E-500

Les commandes E-500 possèdent trois mémoires physiques distinctes :

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

La carte-système contient l'interpréteur du langage PINX-E et le programme UNIPROG en entier.

Figure 1

En remplaçant la carte-système, une commande E-500 peut être instantanément reconvertie pour exécuter des tâches n'utilisant pas UNIPROG.

La carte-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 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'exécution d'un programme d'utilisateur se fait toujours à partir de la mémoire vive. 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.

Figure 2.1.

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. GEOMETRIE ET CINEMATIQUE

Ce chapitre fixe le système de référentiels utilisé par UNIPROG et donne les informations utiles sur les générateurs de mouvements de la commande E-500.

3.1. Références de Position

L'entraînement à moteurs pas-à-pas 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**.

UNIPROG suppose un contact de référence placé à l'une des extrémités de la course.

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

UNIPROG effectue la prise de référence en deux phases :

Phase 1 : Déplacement vers le contact de référence à vitesse choisie (dans le menu de configuration) et arrêt instantané au contact.

Phase 2 : Déplacement en sens opposé à vitesse très basse et arrêt dès que le contact n'est plus actionné.

La phase 2 donne la précision de la position de référence. 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-500 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 l'entrée REF INPUT 8 et l'exécution de la fonction de prise de référence fixera le référentiel initial sans effectuer de mouvement.

3.2. Référentiel Initial et Limites de Course

La course (STROKE) introduite par le menu de configuration délimite l'espace dans lequel se meuvent les axes. Si la course est positive, seuls les points dont la coordonnée dans le référentiel initial est positive sont accessibles, voir figure 3.2. La fonction de prise de référence tient compte du signe de la course.

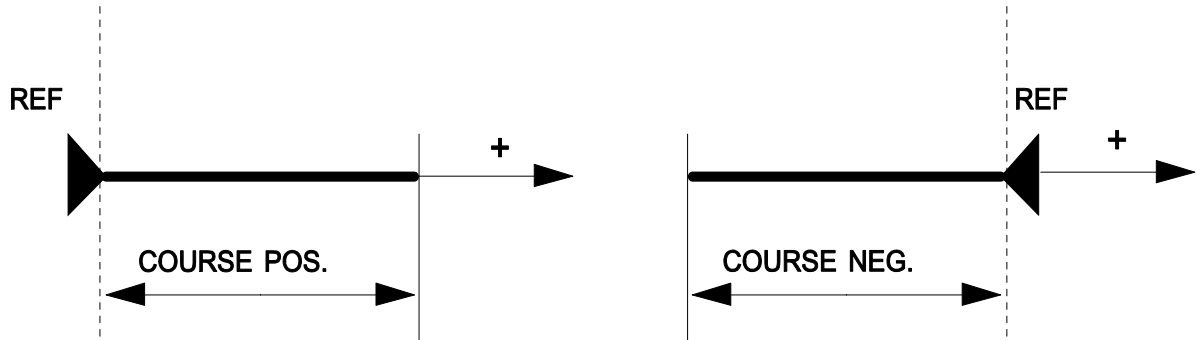


Figure 3.2. Référentiel Initial et Limites de Course

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.

3.3. Changement de Référentiels

Les mouvements à effectuer sont donnés en valeur "relative" ou en coordonnée "absolue". La notion de mouvements relatifs et absolus est applicables 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 3.3;

- 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.

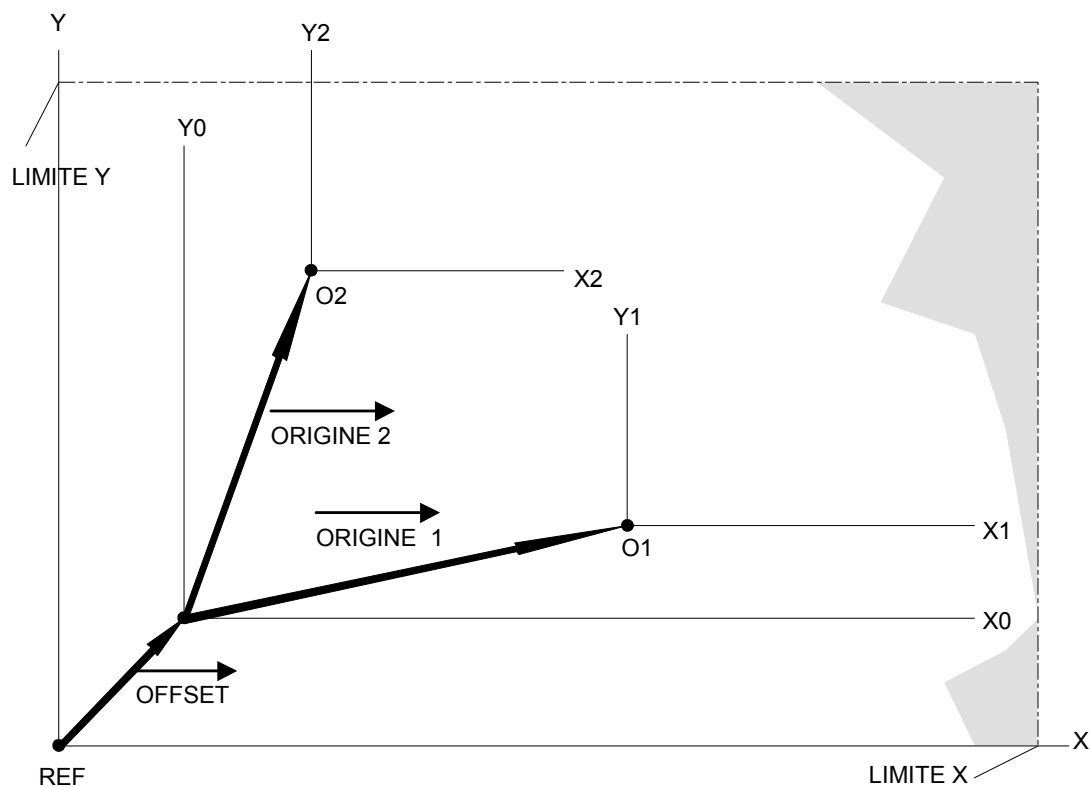


Figure 3.3 : Les Référentiels UNIPROG

3.4. Constantes Cinématiques

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

3.4.1. Le facteur d'échelle **SCALEK**

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

Soit N/U le nombre d'impulsions requises à l'entrée de l'étage de puissance pour une unité. (Pour 1 mm, 1 pouce, 1 degré, etc.). Le facteur d'échelle est donné par :

N/U dépend des options choisies sur l'étage de puissance. Le module BERGER pour moteurs à 5 phases peut demander 500 ou 1000 impulsions par tour, le module SDM 50 pour moteurs à 2 phases est généralement réglé pour 1000 ou 2000 impulsions par tour. Pour de plus amples informations, se référer au manuel de service E-500.

3.4.2. Fréquence Maximale SPMX

Lors de l'exécution d'un mouvement, la fréquence des impulsions -ou la vitesse de l'axe- a l'allure reproduite à la figure 3.4. L'accélération et la décélération décroissent linéairement avec la vitesse. De cette façon, on optimise le temps de positionnement puisque le couple d'un moteur pas-à-pas décroît linéairement avec la fréquence des pas. La fréquence maximale, SPMX, doit être déterminée expérimentalement pour chaque axe de façon à conserver une réserve de couple suffisante à la plus haute fréquence. Le choix de SPMX peut également être dicté à la baisse par d'autres exigences.

$$\text{SPMX} = (\text{Fréquence max. [Hz]})/121,1$$

Dans tous les cas de fonctionnement, la vitesse de l'axe sera inférieure ou égale à la vitesse correspondant à Fréquence max., le générateur engendrant la vitesse de palier par plafonnement de la courbe. Les pentes d'accélération et de décélération sont conservées.

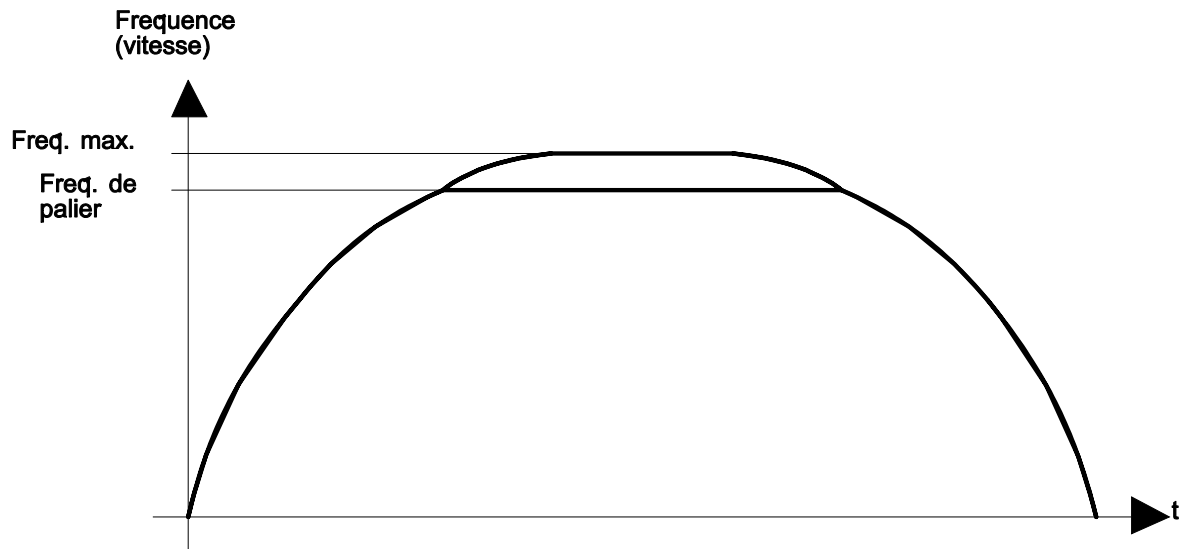


Figure 3.4 : Fréquence ou Vitesse en Fonction de t

3.4.3. Constante d'Accélération KUP, Constante de Décélération, KDN

Ces constantes fixent les rampes de montée et de descente de la vitesse. Leur valeur numérique est liée de façon complexe à la durée de la rampe. A titre d'indication, pour atteindre la fréquence maximale :

<u>KUP ou KDN</u>	<u>Durée de la Rampe</u>
50	280/SPMX [s]
75	92/SPMX [s]
100	30/SPMX [s]

Les valeurs de KUP et de KDN seront déterminées expérimentalement afin de réserver une marge de couple suffisante.

3.4.4. Constante de Rampe linéaire **KLIN**

Lorsque la vitesse de palier de la figure 3.4. est modifiée pendant un mouvement, la vitesse varie linéairement en fonction du temps. Une telle modification de la vitesse peut se produire en agissant sur le potentiomètre de la plaque frontale ou lors de l'arrêt d'un mouvement commandé par une touche (JOGGING). La durée T de la rampe linéaire, pour passer de la vitesse correspondant à SPMX à 0, est approximativement

$$T = 262/KLIN \quad [s]$$

3.4.5. Constante de Vitesse **FEEDK**

La constante de vitesse -ou constante d'avance- a pour but l'expression des vitesses physiques des axes en unités techniques : mm/s, m/min, ...

Soit f_1 , la fréquence requise à l'entrée de l'étage de puissance pour produire une vitesse de 1 unité; la constante de vitesse est donnée par

$$FEEDK = f_1 [Hz] \cdot 2,105$$

Exemple : on veut exprimer les avances en mètres par minute. La transmission et les options de l'étage de puissance exigent 100 impulsions par mm ou 100'000 par m. La fréquence f_1 pour une avance de 1 m/min est donc de $100'000/60 = 1'666,67$ Hz. La constante d'avance à introduire est donc de 3'508.

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 commandé par les touches, voir paragraphes 3.5. et 4.5.

Si on désire simplement exprimer les vitesses en % de la vitesse max. de la figure 3.4. on doit introduire la valeur $2.55 \cdot SPMX$.

3.5. Contrôle de Fermeture

L'entraînement à moteur pas-à-pas se fait en boucle ouverte (sans feedback). 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 $+ - \text{GAP}$ 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.

4. MODE OPERATOIRE DU CLAVIER ET FONCTION DE UNIPROG

Ce chapitre décrit l'utilisation de la commande E-500 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-500.

Enficher la carte UNIPROG 500 V2. x dans la position "SYSTEM" avant la mise sous tension.

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.

Attention: Les commandes E-502 et E-504 dont la désignation de type sur la plaquette de la face arrière se termine par S1 ou B1 ne permettent pas de retirer ou d'enficher la carte-données sans couper l'alimentation. Cette manoeuvre n'est pas destructrice mais le programme peut être perturbé.

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.

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:
REF 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:
REF 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:
REF JOG CLOS DISP
```

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

```
3 DEBUGGING
TRACE PRINT I/O
```

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

```
5 CONFIGURATION
PATH MGEN REF CTRL
```

```
6 ACCESS
press ENTER
```

A partir de la version 2.8, on peut construire un menu 0 qui permet d'entrer une liste de paramètres 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.

4.3. Menu "ACCESS"

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.

4.4. 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.

SPMX	Constante de vitesse asymptotique
KUP	Constante d'accélération
KDN	Constante de décélération
KLIN	Constante de variation linéaire de la vitesse
SCALEK	Facteur d'échelle pour la longueur
FEEDK	Facteur d'échelle pour la vitesse
STROKE	Course de l'axe
OFFSET	Translation point de référence-Référentiel de base.

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.

SPEED TO REF	Vitesse pour prise de référence. Cette vitesse est donnée dans l'unité définie par FEEDK
--------------	--

CLOSURE GAP	Tolérance de fermeture, voir par. 3.5.
-------------	--

REF. INPUT	Attribution de l'une des 8 entrées IN(0)..IN(7)
------------	---

comme contact de référence de l'axe, attribuer 8 pour un axe sans référence physique.

SWITCH: Nature du contact ou du détecteur de référence
Entrer "1" pour un contact normalement ouvert,
"0" pour un contact normalement fermé.

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.

EXTERNAL START Départ du programme, contact normalement ouvert

EXTERNAL PAUSE Suspensions de l'exécution, agit en parallèle avec la touche MAN, contact normalement ouvert.

EXTERNAL STOP Avorte les programmes en cours. Contact normalement fermé (élément d'une chaîne de sécurité).

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 :

DISPLAY FORMAT 1-6.

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 REF (Reference), 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. Prise de Référence, REF

Dans cette fonction, l'affichage indique l'axe sélectionné et sa cote dans le référentiel de base. Pour commander la prise de référence, presser REF. Il faut remarquer que le potentiomètre de la face-avant n'est pas en jeu, la

vitesse est le paramètre SPEED TO REF du paragraphe 4.4.2.

Après la mise sous tension de la commande, aucun mouvement programmé ne peut se faire avant la prise de référence, manuelle ou programmée. Les mouvements manuels sont toujours possibles.

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 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 KLIN. 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.

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 rien ne se passe, l'axe n'a pas de contact de référence attribué. Rappelons qu'après un contrôle de fermeture, l'axe se trouve à son point de référence quel que soit le résultat du test.

4.5.4. Affichage des cotes, DISP

Contrairement aux sous-menus REF, 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 programme qui sera exécuté automatiquement à la mise sous tension, le "POWER-ON-PROGRAMME" et le programme qui démarre à chaque pression sur START (ou à chaque activation de l'entrée désignée dans le menu de configuration), le "START PROGRAMME".

Il faut remarquer que le programme d'initialisation est exécuté à nouveau après un arrêt complet du programme en cours par 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 quittance 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 (suivit 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 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.

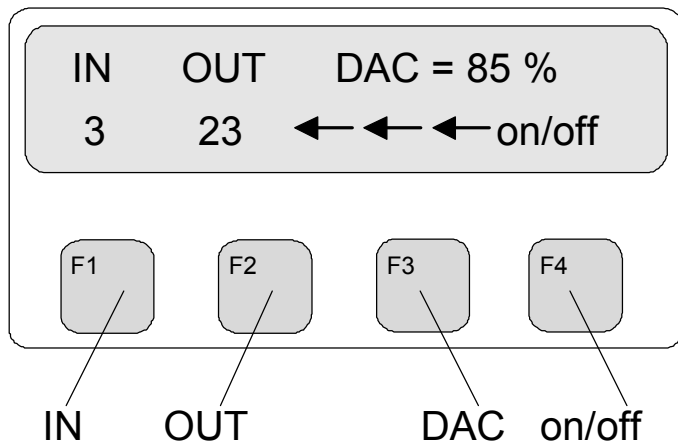
4.8.2. Utilitaire "I/O" control

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 à la table du paragraphe 5.3.

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



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 la table 5.3., les pseudo-I/O numéro 8, 9, 10 sont les états d'activation des programmes simultanés (paragraphe 5.6.); 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 "PRINT"

Une imprimante connectée 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 doit avoir une liaison série initialisée comme suit :

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

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-500 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.
exemple: voir instruction 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.

Positionnement absolu :

10	POSA	axe vitesse	cote	mode
11	POSAD	axe vitesse	adresse de la cote	mode
12	POSAI	axe vitesse	adresse du pointeur	mode

Positionnement relatif:

14	POSR	axe vitesse	déplacement	mode
15	POSRD	axe vitesse	adresse déplacement	mode
16	POSRI	axe vitesse	adresse du pointeur	mode

Argument "vitesse " : Accepte un entier 0..7.

De 0 à 6, la vitesse est prise dans le tableau du paragraphe 4.6. Si vitesse = 7, sa valeur est prise dans l'accumulateur.

Argument "mode" : Précise le mode d'exécution, accepte 0, 1, 2, 3. 0 : le mouvement est pris en compte mais pas exécuté.

1: Tous les mouvements pris en compte sur l'axe désigné sont exécutés.

2: Tous les mouvements pris en compte sur tous les axes sont exécutés.

3: UNIPROG engendre un vecteur dans l'espace désigné, voir chapitre 8.

Dans l'éditeur, les modes sont indiqués par les signes suivants :

0 = _ 1 = | 2 = || 3 = /

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

5.2. Autres instructions Cinématiques

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.

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.

Changement de référentiel :

19

ORG adresse

L'instruction ORG fixe un nouveau référentiel valable pour toutes les instructions de positionnement subséquentes, voir paragraphe 3.3. Les composantes du vecteur de translation de l'origine doivent être déposées, dans l'ordre X, Y, Z, U, dans 4 lignes à partir de "adresse", ceci même si la commande E-500 n'est équipée que de 2 axes.

Instruction d'apprentissage :

13

TEACH axe vitesse adresse

Les arguments "axe" et "vitesse" sont ceux des instructions de positionnement.

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, 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.

5.3. Instructions Entrées/Sorties

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

Attente sur entrée :

20	WAIT0	entrée
21	WAIT1	entrée

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

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. Voir, au sujet de l'adresse de branchement, la remarque du paragraphe 5.6.

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 de la table 5.3.

Table 5.3. Entrées/Sorties UNIPROG

Numéro dans instruction	Entrée physique	Numéro dans instruction	Sortie physique
0	IN(0)	0	BST(0)
1	IN(1)	1	BST(1)
2	IN(2)	2	BST(2)
3	IN(3)	3	BST(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)

Les entrées IN(1) 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, SIM1, SIM2, sont les états d'activation des trois programmes simultanés de UNIPROG.

Les éléments MARQ(i) ne sont pas véritablement des entrées mais des marqueurs -ou drapeaux- internes, librement utilisables par le programmeur. Ces 8 marqueurs sont mis à 0 ou à 1 par les instructions OFF et ON.

Les entrées IN(i) et les sorties OUT(i) sont matérialisées par des modules d'extension I/O connectés par le bus d'extension.

Les sorties BST(i) sont les commandes de sur-excitation des étages de puissance des moteurs pas-à-pas, pour plus de détails, se reporter au manuel de service E-500.

Les sorties OUT(4) à OUT(7) sont les sorties comprises dans le boîtier E-500 et disponibles sur le connecteur de la face arrière, voir chapitre 9.

5.4. Instructions d'Affichage

40	DISPD	U/L	adresse
----	-------	-----	---------

L'instruction DISPD(Display Direct) affiche le contenu numérique de la ligne adressée à la ligne supérieure de l'écran si U/L est 0, à la ligne inférieure si U/L est 1.

L'instruction DISPD commute la fenêtre d'écran, c'est-à-dire que les informations affichées par un utilitaire ne sont pas perdues mais peuvent être restituées par l'instruction RBW (Restore Basic Window).

49	RBW
----	-----

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.

Chargement immédiat de l'accumulateur:

50	FLOAD	nombre réel
51	ILOAD	nombre entier

Chargement direct de l'accumulateur:

52	LOADD	adresse
----	-------	---------

Chargement indirect de l'accumulateur :

53] LOADI adresse pointeur [

Dépose directe de l'accumulateur :

55] STORD adresse [

Dépose indirecte de l'accumulateur:

56] STORI adresse pointeur [

Incrémentation/Décrémentation d'un pointeur :

58] INCD adresse [
59] DECD adresse [

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

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

54] SAVE adresse [

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.

Chargement du convertisseur numérique /analogique, Load DAC :

57] LDAC [

Le contenu de l'accumulateur exprimé en % de 10 V est chargé dans le registre du DAC. L'accumulateur doit contenir un réel positif. UNIPROG ne supporte que l'option unipolaire du DAC soit une sortie de 0 à 10V.

5.6. Instructions de Contrôle du Programme

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.

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.

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.

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 dans [adresse]
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.

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 table 5.3) par 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.

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 place 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.

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

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

5.7. Instructions de Temporisation

70	WAIT temps	Argument immédiat
71	WAITD 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	ADD adresse	Accu = Accu + [adresse]
92	SUBD adresse	Accu = Accu - [adresse]
93	MULD adresse	Accu = Accu . [adresse]
94	DIVD adresse	Accu = Accu / [adresse]

5.9. NOP et "Pseudo-Instructions"

90

NOP

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

FDATA	nombre réel
--------------	--------------------

99

IDATA	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 sont 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'EDITEUR UNIPROG

On entre dans l'éditeur en sélectionnant le menu "PROGRAMMING" 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 la 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 amenés 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 numérique, il n'est pas nécessaire de faire défiler tous les 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. Presse 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. EXECUTION 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.

MOD1:

Exécution normale, les marqueurs de pause et les instructions d'apprentissage 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.

Fonctions des Commandes START, PAUSE, STOP:

Rappelons que ces commandes sont doublées par des entrées définies dans la configuration CTRL. La commande PAUSE impose le mode manuel.

START

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

Si le témoin STOP est allumé, le programme désigné sous POWER-ON PROG est exécuté.

Si le témoin START clignote et que le témoin MAN est allumé, le programme suspendu est repris en MOD1, les mouvements stoppés repartent selon les rampes KLIN correspondantes.

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 sans perdre la position selon les rampes KLIN de la configuration. Le témoin START clignote et le témoin MAN est allumé.

STOP

Tous les programmes en cours sont immédiatement avortés. Les sorties, les booster, le convertisseur D/A sont remis à zéro. Les mouvements sont stoppés de façon abrupte et une prise de référence est requise avant tout nouveau mouvement. La commande E-500 se trouve pratiquement à l'état de mise sous tension.

A la mise sous tension de la commande E-500, 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 ou spécifier un programme ne contenant qu'une ligne END ou NOP.

Remarque: Toutes les fonctions 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 modifient la position physique en mémoire de tous les fichiers dont les numéros sont supérieurs au numéro du fichier en édition. Il peut en résulter des malfunctions catastrophiques.

Si l'instruction d'affichage est utilisée, elle interfère avec la fonction en cours, mais l'écran est restitué si le programme exécute l'instruction RBW.

8. GENERATION DE VECTEUR ET CONTOURNAGE

8.1. Introduction: Possibilités et Définition de l'Espace

Les commandes de la série E-500 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 (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. Cet axe fictif requiert la donnée de toutes ses constantes cinématiques dans le menu de configuration, sous-menu PATH, voir le paragraphe 8.2. (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 11 espaces cartésiens différents:

0	1	2	3	4	5	6	7	8	9	10
XY	XZ	XU	YZ	YU	ZU	XYZ	XYU	XZU	YZU	XYZU

8.2. Configuration des Paramètres Cinématiques Curvilignes

Le sous-menu "PATH" du menu de configuration est un tableau à deux entrées: le paramètres SPMX, KUP, KDN, KLIN, SCALEK et FEEDK sont introduits par les touches-flèches, les différents espaces défilent sous l'action de la touche PATH dans l'ordre du tableau ci-dessus.

Pour l'ordonnée curviligne dans l'espace considéré, les paramètres cinématiques ont la signification qu'ils ont pour les axes, voir chapitre 3. Quelles valeurs faut-il donner aux paramètres de l'espace XY en fonction des paramètres des axes X et Y ? La réponse est évidente lorsque les axes impliqués ont des caractéristiques physique identiques et donc des paramètres indentiques : les paramètres curvilignes sont ceux des axes de l'espace. Dans le cas contraire, l'utilisateur se conforme aux règles ci-après.

Fréquence maximale SPMX:

Le contour à parcourir peut avoir une tangente parallèle à l'axe le moins performant, c'est à dire l'axe exigeant le plus petit SPMX. Le SPMX curviligne doit être pris égal au plus petit SPMX des axes impliqués.

Constantes d'accélération KUP, KDN, KLIN:

Les mêmes considérations conduisent à choisir pour l'ordonnée curviligne la plus petite des constantes d'accélération des axes impliqués.

Facteur d'échelle SCALEK:

Les axes impliqués dans l'espace d'un contour peuvent avoir des constantes d'échelle différentes parce que les rapports de transmission sont différents selon les axes. Nous supposons que les axes utilisent la même unité de mesure qui sera aussi l'unité de l'abscisse curviligne. On doit alors avoir:

$$SCALEK_{path} = \max (SCALEK_i, SCALEK_j, \dots)$$

Dans la parenthèse figure tous les axes impliqués dans l'espace considéré; autrement dit, il faut choisir le plus grand facteur d'échelle parmi les axes impliqués.

Constante de vitesse FEEDK:

On suppose à nouveau que les vitesses des axes impliqués sont exprimées dans une même unité. La constante de vitesse de l'ordonnée curviligne, $FEEDK_{path}$, doit être prise égale à la constante de vitesse de l'axe doté du plus grand facteur d'échelle.

Il faut noter que seuls les espaces effectivement utilisés par le programme nécessitent des paramètres, les autres peuvent être laissés indéfinis.

8.3. Génération d'un Vecteur

Pour produire un vecteur dans le cours du programme il faut préalablement définir l'espace de façon à ce que les paramètres cinématiques voulus soient mis en jeu. L'instruction "Define PATH" est prévue à cet effet.

30

```
[ DPATH espace ]
```

L'argument "espace " est l'un des espaces énumérés au paragraphe 8.1. L'espace défini reste actif jusqu'à la définition d'un nouvel espace.

Le mouvement vectoriel s'effectue lorsqu'une instruction POSA ou POSR avec le mode d'exécution 3 est exécutée.

Il est important de noter que tous les mouvements cumulés par des instructions POSR avec 0 sont alors exécutés, même sur les axes extérieurs à l'espace défini.

8.4. 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 avec les paramètres cinématiques curvilignes introduits par le menu de configuration. 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-500 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 8.5.

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 8.3. 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.)

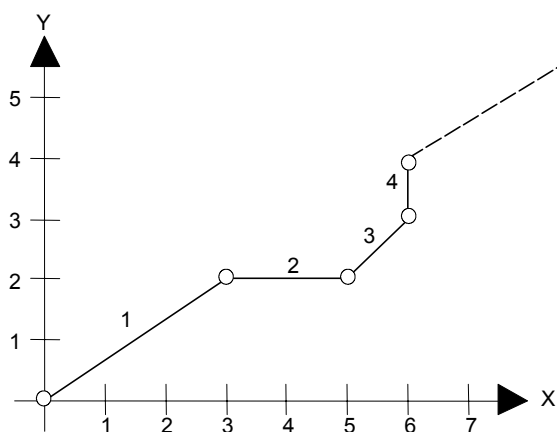
8.4.1. Définition d'un segment linéaire

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 8.4.1. et non aux référentiel 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.

32	LINA	axe	coordonnée	mode
33	LINR	axe	composante	mode

Les coordonnées ou les composantes sont naturellement données dans l'unité définie par SCALEK. Le mode est 0 (0 = _) pour toutes les composantes sauf la dernière écrite qui doit avoir le mode 2 (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 les exemples ci-après. Cette simplification ne fonctionne plus après la programmation d'un arc de cercle.



```

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

```

8.4.2. Définition d'un arc de cercle

Un arc de cercle est approximé par un tracé polygonal fait de segments d'égale longueur, figure 8.4.2. L'angle sous-tendu par un segment est imposé par le programmeur par la pseudo-instruction "Circular DEFinition". (UNIPROG ajuste la définition angulaire vers le bas pour obtenir des segments égaux). La pseudo-instruction CDEF peut ne figurer qu'une fois dans un contour contenant plusieurs cercles.

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 8.4.2.

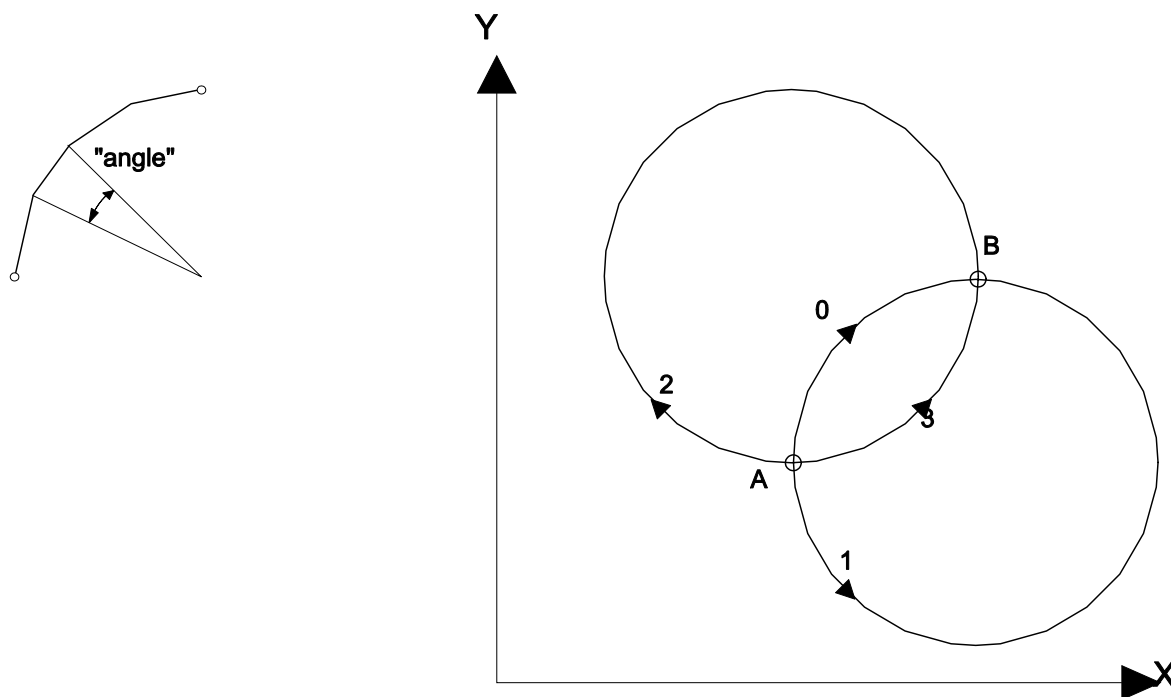


Figure 8.4.2.: Résolution angulaire et Modes

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.

Définition circulaire:

```
35  [ CDEF  angle [ ° ] ]
```

Par défaut, la résolution angulaire est de 5 degrés.

Rayon et Mode:

```
34  [ RAD  mode  rayon ]      voir fig. 8.4.2.
```

Coordonnées de l'extrémité de l'arc :

```
36  [ CIRA  axe  coordonnée mode ]
```

Composantes de la corde:

```
37  [ CIRR  axe  composante mode ]
```

A titre d'exemple, donnons deux façons de coder le contour de la figure 8.4.3.

DPATH	XY		DPATH	XY	
CDEF	6		CDEF	6	; résol. 6 deg.
LINA	X	3 _	LINR	X	3 _
LINA	Y	2	LINR	Y	2
RAD	3	2.5	RAD	3	2.5
CIRA	X	4 _	CIRR	X	1 _
CIRA	Y	4	CIRR	Y	2
RAD	0	5	RAD	0	5
CIRA	X	6 _	CIRR	X	2 _
CIRA	Y	7	CIRR	Y	3

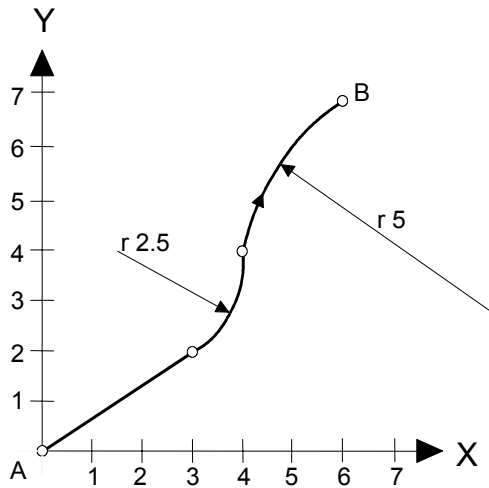


Figure 8.4.3.: Exemple de contour

8.4.3. Arc d'hélice ou d'Hyperhélice

Lorsque l'espace dans lequel on définit un contour est de dimension 3 ou 4, les arcs de cercle deviennent nécessairement des arcs d'hélice ou "d'hyperhélice". Les axes extérieurs au plan du cercle effectuent des mouvements linéaires qui commencent et finissent aux mêmes instants que le mouvement circulaire. Une application commune consiste à tracer une hélice dans un système tri-rectangulaire tout en maintenant un outil normal à la trajectoire.

Les pseudo-instructions LINA/R d'un arc d'hélice doivent nécessairement être écrites après les pseudo-instructions CIRA/R.

Exemple d'un fichier de définition:

```

DPATH  XYZU
CDEF   6.0           ; 6 degrés de résolution
RAD    0    12.0     ; rayon 12 mm, mode 0
CIRA   X    24.0 _
CIRA   Y     0  _    ; demi-cercle
LINA   Z    15.0 _   ; pas de 30 mm
LINA   U     0.5 ||  ; 0.5 unité sur axe U
.....
.....

```

8.5. 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.

- Mémoire vive disponible: 20'000 bytes

- Place pour un contour: $11 + S \cdot (3 + 6 \cdot d)$ 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: $11 + 76 \cdot (3 + 6 \cdot 2) = 1151$ bytes.

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.

8.6. Exécution des Contours

Une seule instruction commande l'exécution d'un contour:

```
38  [ PATH  vitesse  adresse ]
```

L'argument "vitesse" accepte un entier 0..7, se référer au paragraphe 5.1.

L'argument "adresse" pointe la ligne et le fichier contenant la première pseudo-instruction du contour à exécuter.

Il faut remarquer qu'avant l'exécution d'un contour il faut amener physiquement les axes au point de départ du contour. Les éventuels mouvements préparés par des instructions POSA/POSR mais pas encore exécutés sont perdus.

8.7. Résumé des Instructions et Pseudo-Instructions de Définition et de Génération de Contours

30	DPATH	espace	Définition de l'espace, première directive de tout fichier de géométrie, exécutable pour la génération de vecteur.
31	PCOMP	fichier	Instruction d'interprétation d'un fichier de géométrie.
32	LINA	axe coord. mode	Segment linéaire
33	LINR	axe compos. mode	
34	RAD	mode rayon	
35	CDEF	angle	Résolution angulaire
36	CIRA	axe coord. mode	Arc de cercle
37	CIRR	axe compos. mode	
38	PATH	vitesse adresse	Instruction d'exécution d'un contour.

9. RACCORDEMENTS DE LA COMMANDE E-500

Connexions pour Utilisation UNIPROG

Sur toutes les commandes E-500, les connecteurs destinés aux axes X, Y, Z, U sont désignés par A0, A1, A2, A3 respectivement. Le connecteur désigné I/O regroupe les entrées et sorties disponibles dans la commande, les signaux analogiques et les tensions d'alimentation.

9.1. Commande E-502 B, 2 axes, moteurs 5 phases, drivers BERGER

Connecteur A0, Burndy 19 broches :

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	LS(0), fin-de-course ou entrée active haute
M	LS(4), fin-de-course ou entrée active haute
N	+ 24 Volt, retour des inter. de fin de course
P	0 Volt, pour détecteur de prox. 3 fils
R,S	Contact défaut driver, normalement fermé
T	N.C.
U	N.C.
V	Terre de protection

Connecteur A1, Burndy 19 broches:

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	LS(1), fin-de-course ou entrée active haute
M	LS(5), fin-de-course ou entrée active haute
N	+ 24 Volt, retour des inter. de fin de course
P	0 Volt, pour détecteur de prox. 3 fils
R,S	Contact défaut driver, normalement fermé
T	N.C.
U	N.C.
V	Terre de protection

Connecteur I/O, Burndy 19 broches:

Broche	Signal et Destination
A	0 V, retour pour les sorties ²⁾
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 ¹⁾
G	Signal sortie DAC 0..10 V
H	Curseur du potentiomètre, à relier à broche K
J	+ 5 Volt, max. 50mA
K	Signal entrée ADC, curseur du potentiomètre
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 ¹⁾
S	N.C.
T	N.C.
U	N.C.
V	+ 24 V non régulée

1) Les "analogue ground" ne supportent qu'un faible courant et ne doivent être utilisées que comme retour des circuits analogiques.

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

9.2. Commande E-504 B, 4 axes, moteurs 5 phases, drivers BERGER

Connecteurs A0, A1, A2, A3:

Brochage identique aux connecteurs d'axes de la version 2 axes à l'exception de l'attribution des entrées:

A0, broche L IN(0)
A0, broche M IN(4)

A1, broche L IN(1)
A1, broche M IN(5)

A2, broche L IN(2)
A2, broche M IN(6)

A3, broche L IN(3)
A3, broche M IN(7)

Connecteur I/O:

Absolument identique à celui de la version deux axes. Remarquer que les broches L, M, N, P sont en parallèle avec les broches L et M des connecteurs

d'axes A2 et A3, c'est à dire que les entrées IN(2), IN(6), IN(3) et IN(7) sont disponibles en deux endroits afin de simplifier la confection des câbles.

9.3. Commande E-502 S, 2 axes, moteurs 2 phases, drivers SDM 50

Connecteur d'axe A0, Bunrdy 8 broches:

Broche	Signal et Destination
A	Enroulement phase A
B	Enroulement phase A
C	Enroulement phase B
D	Enroulement phase B
E	fin-de-course ou entrée active haute IN(0)
F	fin-de-course ou entrée active haute IN(4)
G	+ 24V, commun des inter, de fin-de-course
H	Sortie défaut driver, peut être connectée sur une entrée.

Connecteur d'axe A1, Bunrdy 8 broches:

Broche	Signal et Destination
A	Enroulement phase A
B	Enroulement phase A
C	Enroulement phase B
D	Enroulement phase B
E	fin-de-course ou entrée active haute IN(1)
F	fin-de-course ou entrée active haute IN(5)
G	+ 24V, commun des inter, de fin-de-course
H	Sortie défaut driver, peut être connectée sur une entrée.

Connecteur I/O, Burndy 19 broches:

Broche	Signal et Destination
A	0 V, retour pour les sorties ²⁾
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 ¹⁾
G	Signal sortie DAC 0..10 V
H	Curseur du potentiomètre, à relier à broche K
J	+ 5 Volt, max. 50mA

K	Signal entrée ADC, curseur du potentiomètre
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 ¹⁾
S	N.C.
T	N.C.
U	N.C.
V	+ 24 V non régulée

1) Les "analogue ground" ne supportent qu'un faible courant et ne doivent être utilisées que comme retour des circuits analogiques.

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

9.4. Commande E-504 S, 4 axes, moteurs 2 phases, drivers SDM 50

Brochage identique aux connecteurs d'axes de la Version 2 axes à l'exception de l'attribution des entrées:

A0, broche E	IN(0)
A0, broche F	IN(4)
A1, broche E	IN(1)
A1, broche F	IN(5)
A2, broche E	IN(2)
A2, broche F	IN(6)
A3, broche E	IN(3)
A3, broche F	IN(7)

Connecteur I/O:

Absolument identique à celui de la version deux axes. Remarquer que les broches L, M, N, P sont en parallèle avec les broches E et F des connecteurs d'axes A2 et A3, c'est à dire que les entrées IN(2), IN(6), IN(3) et IN(7) sont disponibles en deux endroits afin de simplifier la confection des câbles.

10	POSA	axis rate position mode	Absolute Indexing, Immed.
11	POSAD	axis rate address mode	Absolute Indexing, Direct
12	POSAI	axis rate pointer mode	Absolute Index. Indirect
13	TEACH	axis rate address	Teach-In, Direct Argument
14	POSR	axis rate disp'ent mode	Relative Indexing, Immed.
15	POSRD	axis rate address mode	Relative Indexing, Direct
16	POSRI	axis rate pointer mode	Relative Index. Indirect
17	REF	axis	Reference Point
18	CLOS	axis rate	Closure Check
19	ORG	address (4 lines)	Setting a new Origin
20	WAIT0	input	Wait if Input is False
21	WAIT1	input	Wait if Input is True
22	BRIN0	input address	Branch if Input is False
23	BRIN1	input address	Branch if Input is True
24	BRM	address	Branch if Acc is < 0
25	BRP	address	Branch if Acc is ≥ 0
26	BRZ	address	Branch if Acc is zero
27	BRNZ	address	Branch if Acc is non zero
28	OFF	output	Set Output to 0
29	ON	output	Set Output to 1
30	DPATH	space	Space Definition
31	PCOMP	file	Path Computation
32	LINA	axis position mode	Linear Segment Definition
33	LINR	axis component mode	Linear Segment Definition
34	RAD	mode radius	Radius and Mod Definition
35	CDEF	angle	Angular Resolution
36	CIRA	axis position mode	Circ. Segment Definition
37	CIRR	axis component mode	Circ. Segment Definition
38	PATH	rate address	Path Execution
40	DISPD	u/1 address	Display Direct
49	RBW		Restore Basic Window
50	FLOAD	floating	Load Accum Immed., Float.
51	ILOAD	integer	Load Accum Immed., Integer
52	LOADD	address	Load Accum Direct
53	LOADI	pointer	Load Accum Indirect
54	SAVE	address	Save to EEPROM
55	STORD	address	Store Accum Direct
56	STORI	pointer	Store Accum Indirect
57	LDAC		Load DAC from Accum
58	INCD	address	Increment Address Direct
59	DECD	address	Decrement Address Direct
60	JMP	address	Unconditional Jump
61	CALL	address	Sub-Routine Call
62	END		End of Prog. and Routines
63	REP	n times	Repeat, Immediate n
64	REPD	address	Repeat, Direct n
65	ENDRP		End of Repeat Loop
67	SIM1	address	1st Simultan. Prog. Call
68	SIM2	address	2nd Simultan. Prog. Call

70	WAIT	time	Dead Timer, Immed. Time
71	WAITD	address	Dead Timer, Direct Time
90	NOP		No Operation
91	ADDD	address	Add to Accum Direct
92	SUBD	address	Subtract from Accum Direct
93	MULD	address	Multiply Accum Direct
94	DIVD	address	Divide Accum Direct
98	FDATA	floating number	Define a Floating Number
99	IDATA	integer	Define an Integer Number

Axes: X = 0 Y = 1 Z = 2 U = 3

Spaces: 0 1 2 3 4 5 6 7 8 9 10
 XY XZ XU YZ YU ZU XYZ XYU XZU YZU XYZU

Exec Modes (mode): _ = 0, I = 1, II = 2, / = 3

Pause Flag: Key and Led F1 Insert: Key F3 Delete: Key F4

Input	Item	Output	Item
0	IN(0)	0	BST(0)
1	IN(1)	1	BST(1)
2	IN(2)	2	BST(2)
3	IN(3)	3	BST(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	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)