

4/89

الجمهورية الجزائرية الديمقراطية الشعبية  
REPUBLIQUE ALGERIENNE DEMOCRATIQUE ET POPULAIRE

وزارة التعليم العالي  
MINISTERE DE L'ENSEIGNEMENT SUPERIEUR

المدرسة الوطنية المتعددة التقنيات  
BIBLIOTHEQUE — المكتبة  
Ecole Nationale Polytechnique

**ECOLE NATIONALE POLYTECHNIQUE**

DEPARTEMENT : ELECTRONIQUE

**PROJET DE FIN D'ETUDES**

**SUJET**

**CONTRIBUTION A LA  
REALISATION D'UN  
SYSTEME DE PERÇAGE  
AUTOMATIQUE**

Proposé par : Mme BEDDEK    Etudié par : Y. BOUOUDENE    Dirigé par : Mme BEDDEK

Mr SADOON

M. AISSAOUI

Mr SADOON

PROMOTION : JUIN 89

الجمهورية الجزائرية الديمقراطية الشعبية  
REPUBLIQUE ALGERIENNE DEMOCRATIQUE ET POPULAIRE

وزارة التعليم العالي  
MINISTERE DE L'ENSEIGNEMENT SUPERIEUR

المركز الوطني للتقنيات  
BIBLIOTHEQUE — المكتبة  
Ecole Nationale Polytechnique

**ECOLE NATIONALE POLYTECHNIQUE**

DEPARTEMENT : ELECTRONIQUE

**PROJET DE FIN D'ETUDES**

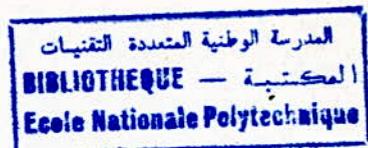
**SUJET**

**CONTRIBUTION A LA  
REALISATION D'UN  
SYSTEME DE PERÇAGE  
AUTOMATIQUE**

Proposé par : Mme BEDDEK    Etudié par : Y. BOUOUDENE    Dirigé par : Mme BEDDEK  
Mr SADOUN    M. AISSAOUI    Mr SADOUN

PROMOTION : JUIN 89

AVANT-PROPOS



Ce travail a été effectué au laboratoire de robotique, du Département d'Electronique de l'Ecole Nationale Polytechnique.

Nous tenons à exprimer notre vive reconnaissance à Madame BEDDEK et à Monsieur SADOUN, nos promoteurs, pour nous avoir aidés et dirigés pendant l'élaboration de ce projet.

Nous adressons nos remerciements à Monsieur ZAIM, du Département du Génie Electrique, pour son précieux support bibliographique.

Nos remerciements vont également à nos collègues et amis pour leur collaboration, en particulier à Melle F.AISSAOUI, MM. A.ZEMMOURI et R.BELDJOUDI.

Nous remercions enfin Mademoiselle D.HAMOULI pour le soin mis à la dactylographie de ce mémoire.

المدرسة الوطنية المتعددة التقنيات  
BIBLIOTHEQUE — المكتبة  
Ecole Nationale Polytechnique

# SOMMAIRE

INTRODUCTION

CHAPITRE I - ORGANISATION GENERALE DU SYSTEME ET CAHIER DES CHARGES

I.1 - La conception assistée par ordinateur de circuits imprimés	2
I.2 - Le système mécanique	4
I.3 - La carte de commande	4
I.4 - Le logiciel de commande	5
I.5 - Conclusion	6

CHAPITRE II - SYSTEME MECANIQUE

II.1 - Types de déplacements	7
II.1.1 - Déplacement en $r, \theta$	
II.1.2 - Déplacement en X,Y	
II.2 - Système mécanique adopté	8
II.2.1 - Chaîne cinématique	
II.2.2 - Choix des moteurs	
II.3 - Conclusion	15

CHAPITRE III - PRINCIPES DE FONCTIONNEMENT ET ALIMENTATION DES MOTEURS PAS A PAS

PREMIERE PARTIE - PRINCIPES DE FONCTIONNEMENT DES MOTEURS PAS A PAS

III.1 - Types de moteurs pas à pas	16
III.1.1 - Moteur pas à pas à réluctance variable	
III.1.2 - Moteur pas à pas à aimant permanent	
III.1.3 - Moteur pas à pas hybride ou réluctant polarisé	
III.2 - Caractéristiques communes aux moteurs pas à pas	19
III.2.1 - Caractéristique couple-fréquence	
III.2.2 - Phénomène d'oscillation	
III.3 - Comparaisons fondamentales des moteurs pas à pas	20
III.4 - Conclusion	24

## SECONDE PARTIE - ALIMENTATION DES MOTEURS PAS A PAS

III.5 - Schéma classique de l'alimentation	25
III.6 - Alimentation à deux niveaux de tension	25
III.7 - Alimentation par hacheur	27
III.8 - Alimentation par hacheur programmable	29
III.9 - Conclusion	32

## TROISIEME PARTIE - REALISATION DE L'INTERFACE DE PUISSANCE

III.10 - Fonctionnement du circuit hacheur	33
III.11 - Réalisation du circuit	36
III.12 - Conclusion	40

## CHAPITRE IV - LA CARTE ET LE LOGICIEL DE COMMANDE

### PREMIERE PARTIE - LA CARTE DE COMMANDE

IV.1 - La carte TM 990/189 UNIVERSITY	43
IV.1.1 - L'interface programmable multifonctions TMS 9901	
IV.1.2 - Le contrôleur de communications asynchrone TMS 9902	
IV.2 - Utilisation de la carte TM 990/189 pour la réalisation de la carte de commande	46

### SECONDE PARTIE - LE LOGICIEL DE GESTION

IV.3 - Décodage du fichier image du circuit imprimé	50
IV.3.1 - Architecture du 8086	
IV.3.2 - Détournement de l'interruption "BREAK CLAVIER" du BIOS	
IV.4 - Traitement par la carte de commande	59
IV.4.1 - Techniques de commande du moteur pas à pas	
IV.4.2 - Détermination des fréquences maximales de démarrage et absolue du moteur	
IV.4.3 - Sous-programmes de commande du système	

TROISIEME PARTIE - ORGANISATION GENERALE DU LOGICIEL DE GESTION

IV.5 - Mode de commande à distance	73
IV.5.1 - Perçage à la chaîne des trous	
IV.5.2 - Perçage individuel des trous (curseur viseur)	
IV.6 - Mode de commande locale	74
IV.7 - Conclusion	79
<u>CONCLUSION</u>	80
<u>ANNEXE</u>	82
<u>BIBLIOGRAPHIE</u>	

المدرسة الوطنية المتعددة التقنيات  
BIBLIOTHEQUE — المكتبة  
Ecole Nationale Polytechnique

# INTRODUCTION

L'évolution de la technologie moderne s'oriente vers l'automatisation de processus industriels. Cette automatisation s'est accélérée avec l'avènement du microprocesseur

En effet, celui-ci peut effectuer, par programme, plusieurs tâches et permet une certaine flexibilité et une souplesse d'utilisation.

Les techniques de fabrication de circuits imprimés évoluent rapidement et deviennent de plus en plus perfectionnées.

Le recours aux circuits multicouches et à la technique de montage en surface, dans le but de l'augmentation de la densité des circuits, nécessite des techniques de montage et de contrôle particulièrement fiables.

De ce fait, l'automatisation de ces techniques est devenue nécessaire.

Dans cette optique, une contribution à l'étude et à la réalisation d'un système de perçage automatique de circuits imprimés nous a été confiée.

Le but de ce travail est l'automatisation d'une étape importante dans la fabrication de circuits imprimés, en l'occurrence l'opération de perçage des trous.

Ce système est destiné à être piloté par un micro-ordinateur compatible IBM-PC.

Notre étude se décompose en quatre parties principales :

- la première partie est une présentation du projet et du cahier des charges.
- La seconde partie porte sur l'étude du système mécanique adopté, qui doit assurer le positionnement de l'outil.
- La troisième partie présente les caractéristiques fondamentales des moteurs pas à pas, leurs principes de fonctionnement, leur alimentation, ainsi que la réalisation de l'interface de puissance.
- La dernière partie concerne la carte de commande et le logiciel de gestion du système de perçage.

# CHAPITRE 1

ORGANISATION GENERALE DU  
SYSTEME ET  
CAHIER DES CHARGES

Le système de perçage automatique de circuits imprimés est un outil qui doit permettre le passage de la conception assistée par ordinateur (C.A.O.) à la fabrication assistée par ordinateur (F.A.O.) de circuits imprimés.

A partir du schéma réalisé à l'aide d'un logiciel de conception de circuits, le logiciel de gestion du système effectue un décodage des paramètres nécessaires à l'opération de perçage, en l'occurrence la position des trous sur le circuit imprimé et leurs diamètres.

Ensuite la commande de perçage est envoyée aux éléments moteurs du système mécanique.

Le schéma synoptique du système, tel que défini par le cahier des charges, est donné en figure I.1.

#### 1.1) - LA CONCEPTION ASSISTEE PAR ORDINATEUR DE CIRCUITS IMPRIMES :

Il existe différents logiciels pour le dessin de circuits imprimés ; les plus simples utilisent des procédures graphiques pour le tracé de pistes, et des symboles pour le dessin de pastilles.

D'autres, plus sophistiqués, procèdent à un codage de tous les paramètres de définition du circuit ; le micro-ordinateur se charge de l'implantation selon un algorithme déterminé.

Dans ce cas, la conception fait appel à une bibliothèque contenant les dimensions physiques, les caractéristiques électriques et les schémas élémentaires des composants ainsi que les instructions relatives au groupement optimal des éléments.

La transformation des connexions en routes se fait avec l'assistance de l'ordinateur.

Une fois le dessin terminé, on sauvegarde sur disque les données constituant ainsi le archives du circuit.

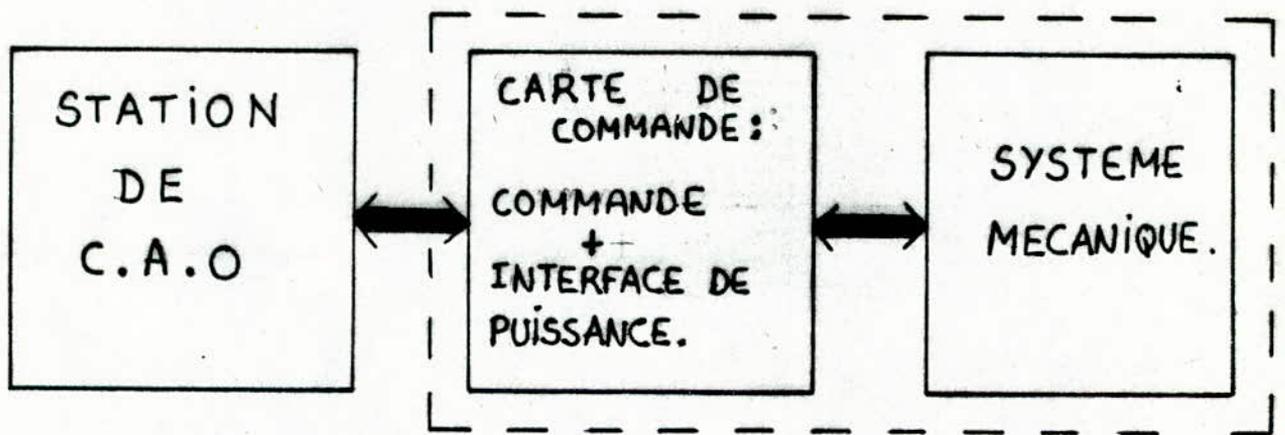


FIGURE I.1- SCHEMA SYNOPTIQUE DU SYSTEME DE PERÇAGE.

La dernière phase consiste à sortir le dessin sur mylar ou calque pour la photo-gravure, en utilisant un organe de sortie graphique.

### 1.2) - LE SYSTEME MECANIQUE :

Le système mécanique qui porte l'outil, doit posséder un bâti rigide indéformable de façon à garantir la précision du perçage.

La réalisation d'un tel système doit-être faite avec des tolérances minimales afin de réduire les frottements et permettre un positionnement précis de l'outil.

L'entraînement des parties mobiles du système mécanique sera effectuée à l'aide de moteurs.

Le choix du type de déplacement de la partie mécanique ainsi que le type de moteurs doit répondre aux exigences de la fonction à réaliser et dépend de la charge maximale à supporter.

Un changement automatique d'outil est à prévoir de manière à percer les trous de différents diamètres.

### 1.3) - LA CARTE DE COMMANDE :

La carte de commande permet de traiter les informations relatives à la conception du circuit imprimé, et d'assurer l'exécution de la fonction désirée.

Elle est constituée de deux parties :

l'interface de puissance qui sert à l'alimentation des moteurs, et la partie commande qui fournit les signaux numériques nécessaires au fonctionnement et au contrôle du système de perçage.

Celui-ci devant être piloté par microprocesseur, la partie commande peut-être conçue de deux manières différentes selon que l'on veuille réaliser un système indépendant, ou spécifique à un micro-ordinateur, par exemple l'IBM-PC.

Le système indépendant dispose de sa propre unité centrale (C.P.U.) et peut communiquer avec n'importe quel terminal à travers une liaison série asynchrone.

Le système spécifique à un compatible IBM-PC est réalisé avec une carte imprimée au format standard PC.

Celle-ci s'interconnecte au bus du PC par l'intermédiaire du slot d'extension disponible sur la carte mère du PC.

#### 1.4) - LE LOGICIEL DE COMMANDE :

C'est le logiciel de gestion du système de perçage.

Celui-ci doit assurer la récupération de toutes les données, qui serviront à l'exécution de la tâche, à partir du fichier image du circuit imprimé.

La récupération des données doit-être possible pour tout logiciel de conception de circuits imprimés.

Le logiciel de gestion doit en outre disposer d'un ensemble de commandes qui permettent de gérer le système et de faciliter le dialogue avec l'utilisateur.

I.5) - CONCLUSION :

Dans ce chapitre, nous avons présenté les différentes étapes de notre projet, et nous avons défini le rôle de chacune des parties du système de perçage ainsi que les exigences auxquelles elles doivent répondre conformément au cahier des charges.

Dans les prochains chapitres, nous procéderons à une étude détaillée de chaque partie en examinant les différentes possibilités de réalisation. Ceci nous permettra de mieux cerner le choix le plus adapté à notre système.

# CHAPITRE 2

SYSTEME MECANIQUE

Le système mécanique est constitué d'un ensemble de parties mobiles permettant le déplacement de l'outil dans un système à trois dimensions.

Ces déplacements sont assurés par des moteurs convenablement disposés sur les bâtis du système.

Dans ce chapitre, nous verrons d'abord les différents types de déplacement qui existent avant de faire le choix adapté au système ~~du système~~ de perçage. Nous proposerons ensuite un schéma de principe du système mécanique adopté.

Enfin, nous déduirons le type de moteur qui répond le plus aux exigences de la fonction à réaliser et du système mécanique.

## II.1) - TYPES DE DEPLACEMENTS :

### II.1-1) - Déplacement en $r, \theta$ :

Un système utilisant un tel déplacement est schématisé à la figure II.1. L'une des deux extrémités du système est fixée à la plaque de base par une liaison cylindrique lui permettant de s'orienter par rotation selon  $\theta$  grâce à un premier moteur de direction.

Une liaison prismatique entre le bras et le chariot porte-outil permet à l'outil de fixer le rayon d'action grâce à un second moteur de déplacement.

La précision de positionnement est définie comme étant le degré de coïncidence entre la position commandée et la position réelle.

Pour un tel type de déplacement, la précision de positionnement est liée au pas du moteur. L'erreur est de :

$$\Delta S = r \cdot \Delta \theta$$

où  $\Delta\theta$  est le pas du bras de direction et  $r$  le déplacement du chariot porte-outil.

Comme  $r$  est variable, la précision ne possède pas une distribution uniforme sur le champ d'action, elle est meilleure pour les faibles rayons.

De plus, ce système présente un manque de guidage selon  $\theta$ .

Son avantage est qu'il présente une possibilité d'extension.

#### II. 1-2) - DEPLACEMENT EN X,Y :

Le déplacement se fait grâce à des mouvements parallèles aux axes X et Y (figure II.2)

En considérant les deux moteurs identiques, la précision de positionnement est uniforme sur tout le champ d'action ; elle est donc indépendante du format.

Les systèmes réalisant ce type de déplacement présentent un bon guidage dans les deux directions.

Leur inconvénient est un format limité qui dépend des courses maximales en X et Y [1]

#### II.2) - SYSTEME MECANIQUE ADOPTE :

Etant donné que les dimensions des circuits imprimés sont de plus en plus réduites grâce à l'intégration croissante des composants électroniques, il n'est pas nécessaire de prévoir un grand champ d'action ; par contre on doit avoir une bonne précision de positionnement en tout point de ce champ.

Comme le système XY répond bien à ces exigences et qu'il présente un bon guidage sur ses axes, nous avons opté pour un tel système.

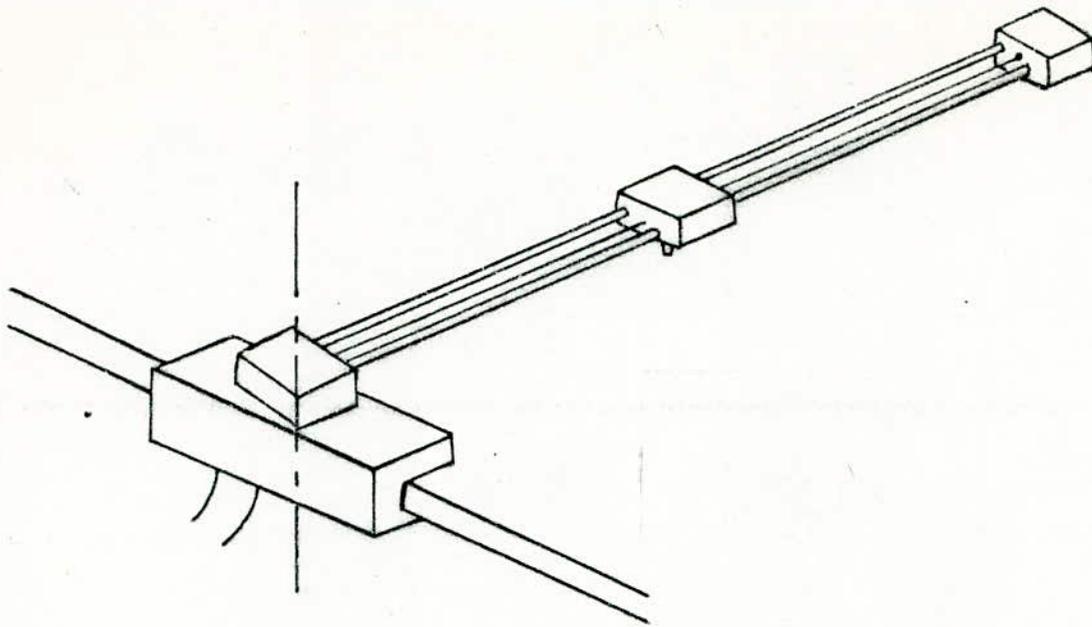


FIGURE II.1-SYSTEME A DEPLACEMENTS  
EN  $\pi, \theta$

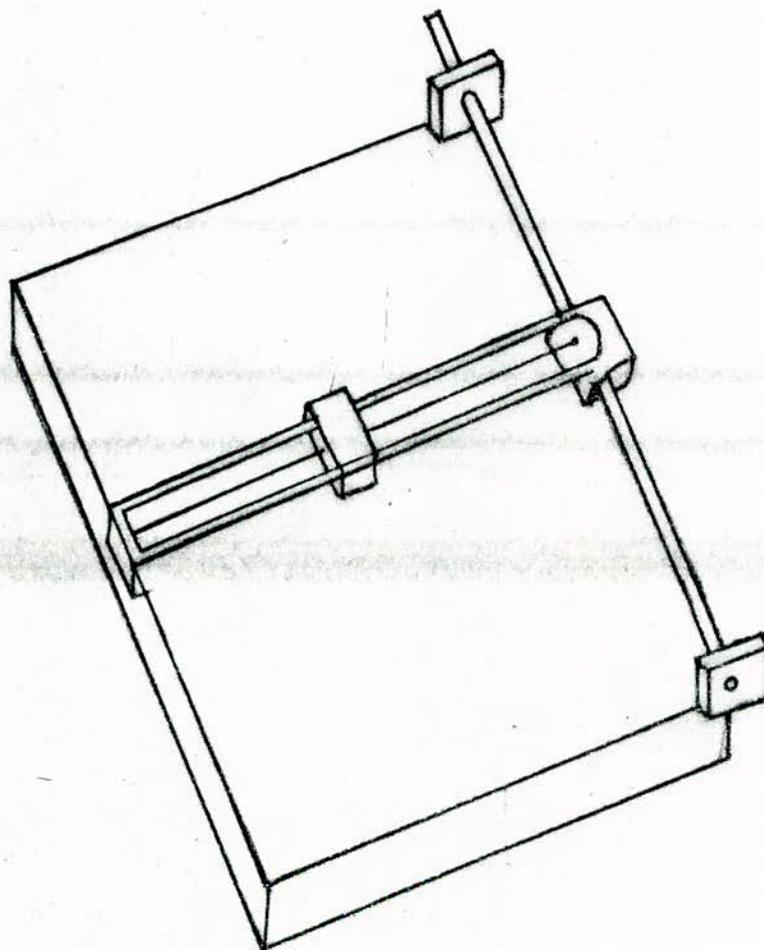


FIGURE II.2- SYSTEME XY.

### II.2-1) - CHAINE CINEMATIQUE :

La figure II.3 montre la chaîne cinématique du système mécanique proposé.

Nous avons prévu trois degrés de liberté pour ce système.

Chacun des trois moteurs assure le déplacement dans une direction. Un accouplement rigide permet de transmettre la rotation de l'arbre du moteur vers une tige filetée à pas trapézoïdal.

Le mouvement de rotation de la tige est transformé en mouvement de translation grâce à un écrou associé à cette tige et lié au bâti à déplacer (système vis-écrou).

Le système de guidage du mouvement en Y est assuré par des glissières en queue d'aronde.

Un schéma de principe de ce système est donné à la figure II.4.

Nous avons prévu des courses maximales X, Y et Z respectivement de 400 mm, 250 mm et 40 mm.

Pour la tige filetée, nous avons retenu pour le diamètre une valeur normalisée de 25 mm pour laquelle correspond un pas de 4 mm [2].

### II.2-2) - CHOIX DES MOTEURS :

L'opération de perçage de circuits imprimés consiste en une suite de mouvements discrets ; de ce fait, nous utiliserons des moteurs pas à pas pour assurer les différents déplacements.

En effet, comparés aux moteurs à courant continu, la commande des moteurs pas à pas est plus facilement réalisable par microprocesseur ; de plus, ces derniers réalisent des mouvements incrémentaux et des arrêts nets sans recours au freinage.

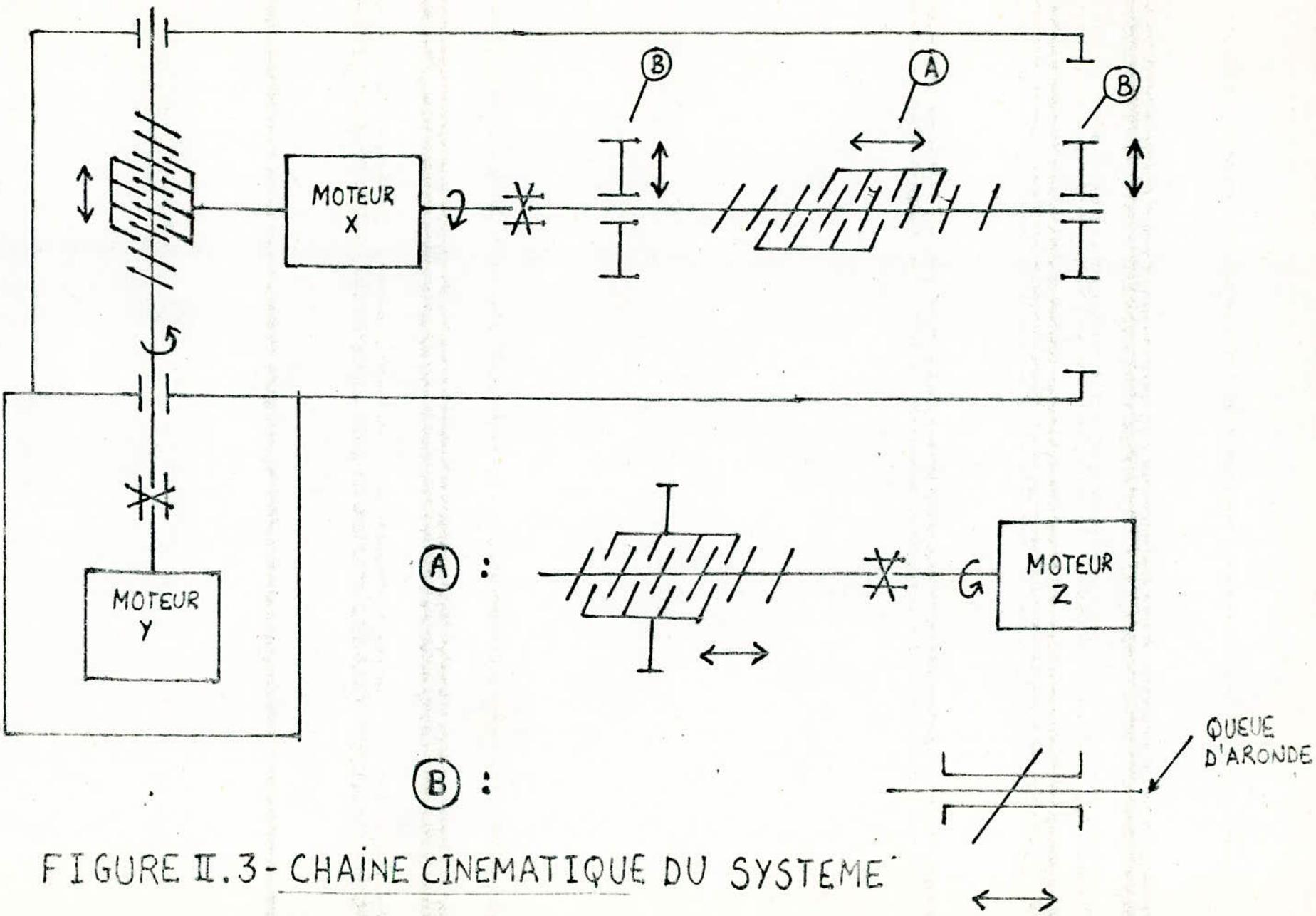


FIGURE II.3 - CHAÎNE CINEMATIQUE DU SYSTEME MECANIQUE.

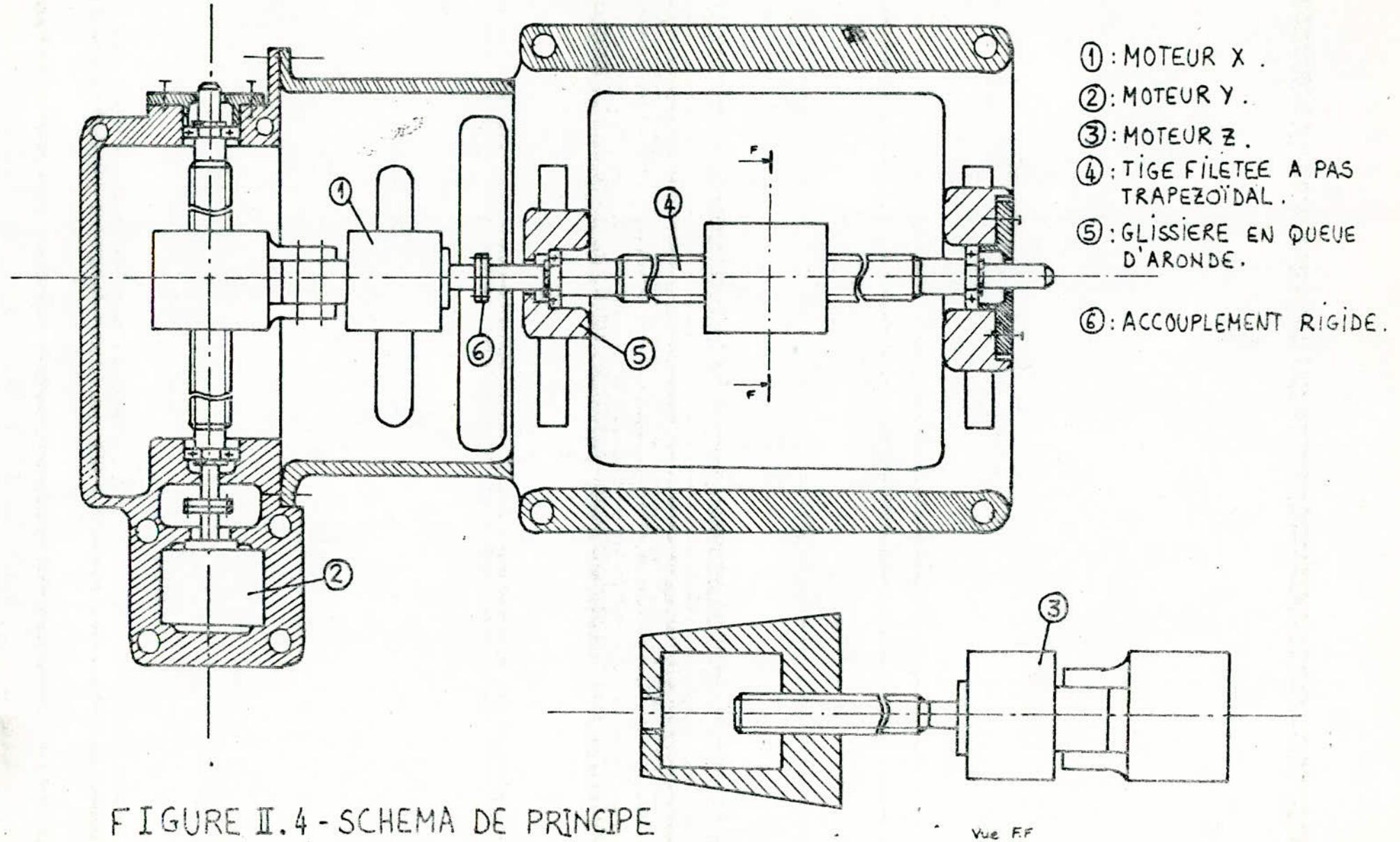


FIGURE II.4 - SCHEMA DE PRINCIPE  
DU SYSTEME MECANIQUE.

Vue FF

Pour une résolution donnée, les moteurs doivent avoir un nombre de pas par tour égal au rapport du pas de la vis trapezoïdale par la résolution.

Par exemple pour une résolution de 0,1 mm, le pas de vis étant de 4 mm, les moteurs doivent avoir 40 pas par tour.

Pour déplacer les bâtis sans contrainte, chaque moteur doit disposer d'un couple suffisant.

On démontre [3] que pour une liaison directe (figure II.5), le couple moteur est donné par :

$$\Gamma_M = \Gamma_R = (J_M + J_C) \cdot \frac{d\Omega_M}{dt} + (F_M + F_C) \cdot \Omega_M$$

où :  $\Gamma_R$  = couple résistant.

$J_M$  = moment d'inertie du moteur.

$J_C$  = moment d'inertie de la charge.

$F_M$  = frottement mécanique du moteur.

$F_C$  = frottement mécanique de la charge.

La puissance transmise en rotation est alors de :

$$P = \Gamma_M \cdot \Omega_M \quad \text{où : } \Omega_M = \text{vitesse de rotation en rad/S.}$$

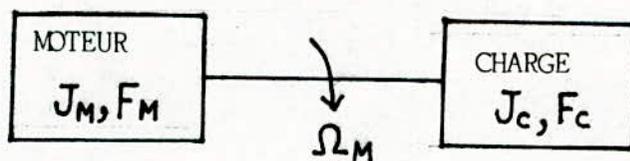


FIGURE II.5

Le calcul rigoureux du couple nécessite la connaissance de tous ces paramètres, ce qui dépend de la réalisation mécanique.

Néanmoins, sachant que la charge sur chaque moteur ne dépasse pas 02 Kg pour le système mécanique proposé, on peut estimer la valeur du couple maximal à  $\Gamma_M = 11 \text{ kg cm}$ .

### II.3) - CONCLUSION :

Dans ce chapitre, nous avons comparé différents types de déplacements, puis nous avons proposé un schéma de principe du système mécanique qui correspond au type de déplacement le plus adapté à l'opération de perçage.

Ensuite nous avons opté pour des moteurs pas à pas afin de réaliser les déplacements de l'outil selon les trois directions.

Pour assurer ces déplacements avec précision et sans contraintes, une estimation des caractéristiques des moteurs a été faite.

Néanmoins, pour concrétiser le système de perçage, il est nécessaire de commander correctement chacun des moteurs à travers un étage de puissance, et de coordonner les mouvements des moteurs par un logiciel de gestion.

Dans ce but, nous présenterons au prochain chapitre une étude sur les moteurs pas à pas et la réalisation de la carte d'interface de puissance pour l'alimentation des moteurs.

# CHAPITRE 3

PRINCIPES DE FONCTIONNEMENT  
ET ALIMENTATION  
DES MOTEURS PAS A PAS

## PREMIERE PARTIE : PRINCIPES DE FONCTIONNEMENT DES MOTEURS PAS A PAS

Le moteur pas à pas est un élément électromagnétique incrémental qui convertit des informations numériques en un positionnement angulaire ou linéaire.

Chaque impulsion d'alimentation provoque un pas de déplacement de l'axe du moteur et l'accroche magnétiquement au point précis auquel celui-ci doit se mettre.

Il peut fonctionner dans les deux sens de rotation avec des accélérations, des arrêts et une réversibilité rapides ; de plus il se caractérise par un écart de positionnement non cumulatif.

Comme notre système de perçage doit réaliser une suite de mouvements discrets de précision, et que sa commande se fera par un système microprogrammé, l'utilisation des moteurs pas à pas est justifiée.

### III - 1) - TYPES DE MOTUERS PAS A PAS :

Le moteur pas à pas est basé sur un principe semblable à celui d'un moteur synchrone, ce principe repose sur l'application des systèmes réductants polarisés ou non.

Le couple moteur résulte de la variation des inductances propres, il est associé à une déformation du circuit magnétique [ 3 ].

La partie fixe ou stator, possède un certain nombre de bobines disposées régulièrement sur son périmètre.

La partie tournante ou rotor dépend du type de moteur.

Il existe trois types de moteurs pas à pas :

#### III - 1.1. - MOTEUR PAS A PAS A RELUCTANCE VARIABLE :

La figure III.1 - montre la coupe d'un tel moteur. Le rotor est en matériau magnétique non polarisé, il dispose à sa périphérie d'un nombre de dents différent de celui du stator.

Lorsqu'une phase est alimentée, les deux bobines correspondantes vues du rotor sont de sens contraire, nord - sud ou sud - nord ; le rotor tend à se déplacer de façon à aboutir à une inductance maximale du circuit magnétique, en partant d'un état d'équilibre à inductance minimale.

Pour que le rotor fasse un tour complet dans le sens trigonométrique, il faut répéter quatre fois la séquence de commutation des phases suivantes : A - A', B - B' et C - C' ; ce qui nécessite 12 commutations ; ceci correspond donc à 12 pas par tour [ 3 ][ 4 ].

### III - 1.2. - MOTEUR PAS A PAS A AIMANT PERMANENT :

Le rotor comporte des pôles nord et sud alternés.

La figure III.2 montre un moteur ayant quatre phases au stator.

Si la phase A est excitée par la fermeture du commutateur C1, les pôles du rotor sont attirés par les pôles opposés du stator ; ce qui donne la position stable P1.

L'excitation de la phase B avec l'extinction de la phase A provoquera la rotation du rotor vers la position P2 de telle façon qu'il aligne son champ magnétique avec celui développé par le stator (règle du flux maximal) [ 4 ][ 5 ].

La séquence d'excitation est donnée par le tableau suivant :

COMMUTATEUR FERME	PHASE EXCITEE	POSITION DU ROTOR	SENS DE ROTATION
C1	A	↑ (P1)	
C2	B	→ (P2)	
C3	C	↓ (P3)	
C4	D	← (P4)	
C1	A	↑ (P1)	

TABLEAU 1 : SEQUENCE D'EXCITATION

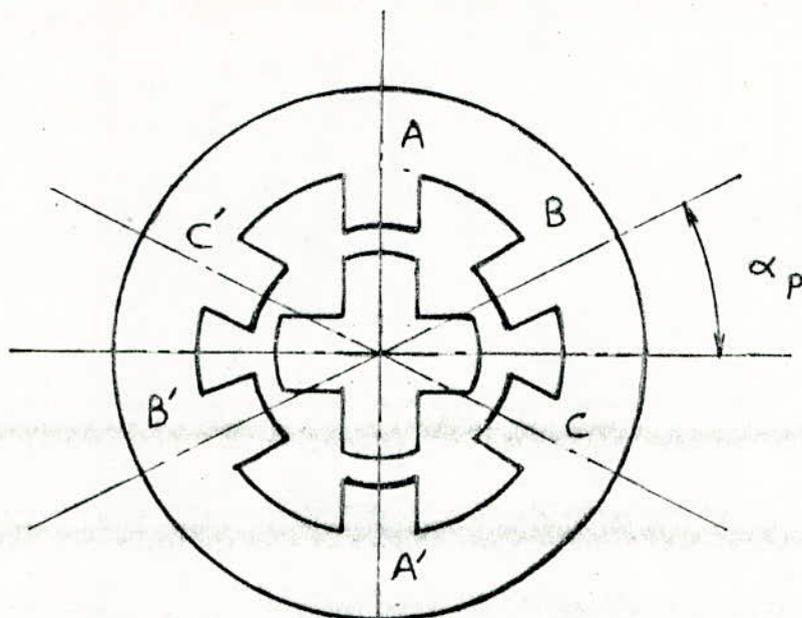


FIGURE III.1-COUCPE D'UN MOTEUR PAS A PAS A RELUCTANCE VARIABLE.

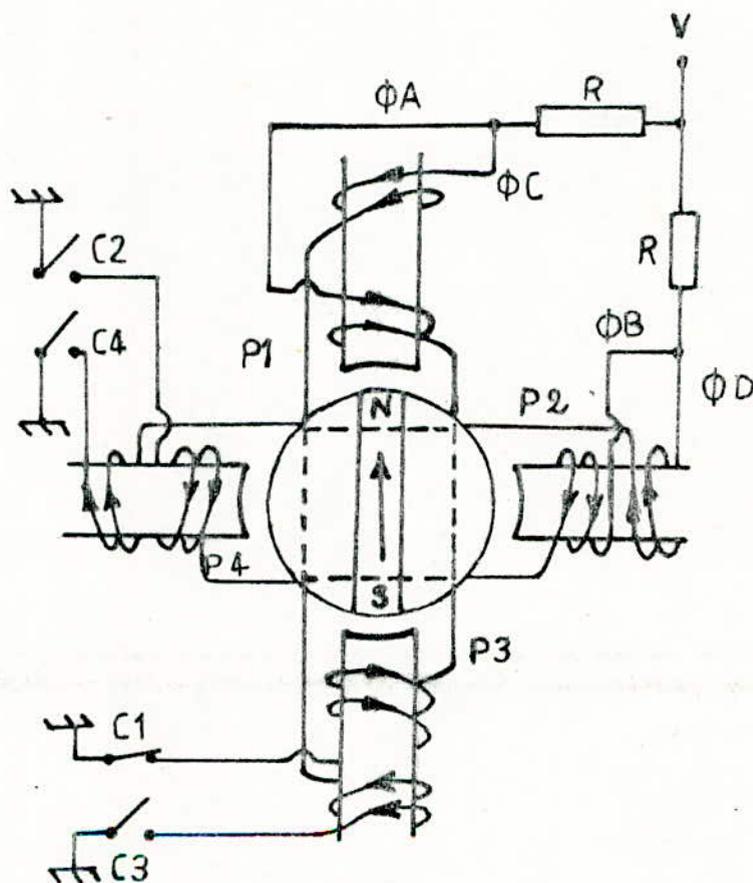


FIGURE III.2-MOTEUR PAS A PAS A AIMANT PERMANENT.

Le nombre de pas correspond au nombre de commutations.

### III.1.3.) - MOTEUR PAS A PAS HYBRIDE OU RELUCTANT POLARISE :

Ce moteur combine les deux principes précédents ; le positionnement du rotor est obtenu par la combinaison de deux effets :

- effet de réluctance variable dû aux "dents",
- propriété des aimants permanents.

### III.2) - CARACTERISTIQUES COMMUNES AUX MOTEURS PAS A PAS :

#### III.2.1.) - CARACTERISTIQUE COUPLE-FREQUENCE :

Le fonctionnement d'un moteur pas à pas est conditionné par la caractéristique couple - fréquence des pas (figure III.3) ; ainsi on distingue le fonctionnement à charge fixe et le fonctionnement à fréquence fixe :

#### - Fonctionnement à charge fixe :

Pour une charge donnée C, le moteur ne peut démarrer sans perte de pas que si la fréquence est inférieure à la fréquence maximale de démarrage en charge (FMD) ; il peut être accéléré ensuite (dans la zone de survitesse) jusqu'à la fréquence maximale d'entraînement en charge ou fréquence maximale absolue (FMA).

Dans cette zone, le ~~moteur~~ ne peut ni s'arrêter, ni ~~changer~~ de sens de rotation.

Pour arrêter le moteur sans perte de pas, il faut repasser par la fréquence FMD et de là on peut effectuer soit une décélération progressive, soit une suppression des impulsions [3].

- Fonctionnement à fréquence fixe :

Dans ce cas un raisonnement analogue peut être fait.

III - 2.2. - PHENOMENE D'OSCILLATION :

Selon la fréquence des impulsions de commande, une oscillation apparaît autour de la position d'équilibre (figure III.4). Celle-ci est due à l'inertie du rotor et au circuit électrique d'alimentation.

L'enchaînement des pas à vitesse croissante diminue l'amplitude des oscillations. En effet, au delà de la fréquence de résonance (fréquence des pas égale à la fréquence d'oscillation), chaque commutation a lieu avant que les oscillations dues au pas précédent n'aient cessé comme le montre la figure III.5 [3][4].

III. 3 - COMPARAISONS FONDAMENTALES DES MOTEURS PAS A PAS :

Une comparaison entre différents types de moteurs pas à pas de taille égales est faite dans le tableau 2 ; Les moteurs à aimant permanent étant comparés aux moteurs à réluctance variable.

TYPE DE MOTEUR	CARACTERISTIQUES
à réluctance variable	<ul style="list-style-type: none"><li>- grandes vitesses de fonctionnement .</li><li>- un nombre de pas par tour compris entre 6 et 200 .</li><li>- bonne résolution .</li></ul>
à aimant permanent	<ul style="list-style-type: none"><li>- un couple plus élevé .</li><li>- un couple de maintien sans alimentation .</li><li>- une plus grande inertie propre donc meilleure stabilité .</li><li>- un amortissement interne des oscillations plus important .</li><li>- un nombre de pas par tour plus faible (compris entre 2 et 48).</li></ul>

Hybride	<ul style="list-style-type: none"> <li>- un couple plus important que les moteurs à réluctance variable .</li> <li>- puissance utile plus importante .</li> <li>- nombre de pas par tour élevé (compris entre 24 et 400).</li> </ul>
---------	--

TABLEAU 2 : COMPARAISON DES MOTEURS PAS A PAS

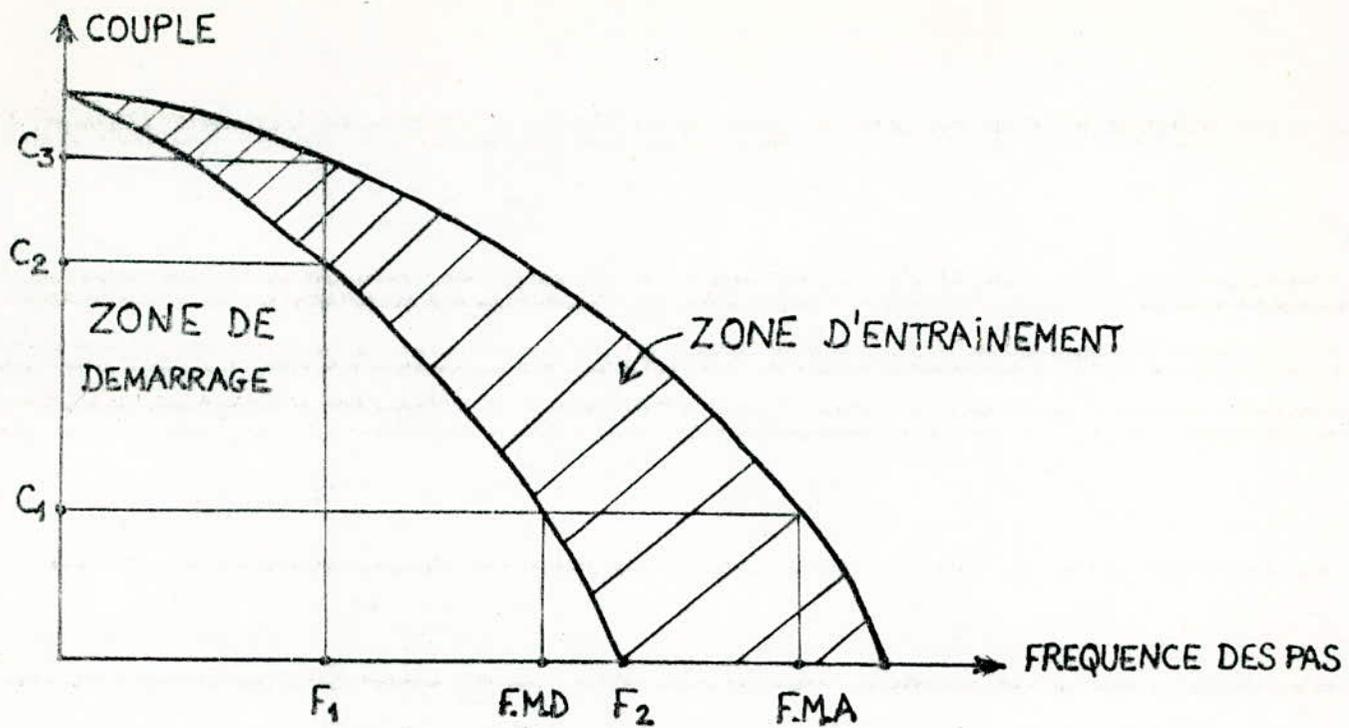


FIGURE III.3-CARACTERISTIQUE COUPLE-FREQUENCE DES PAS.

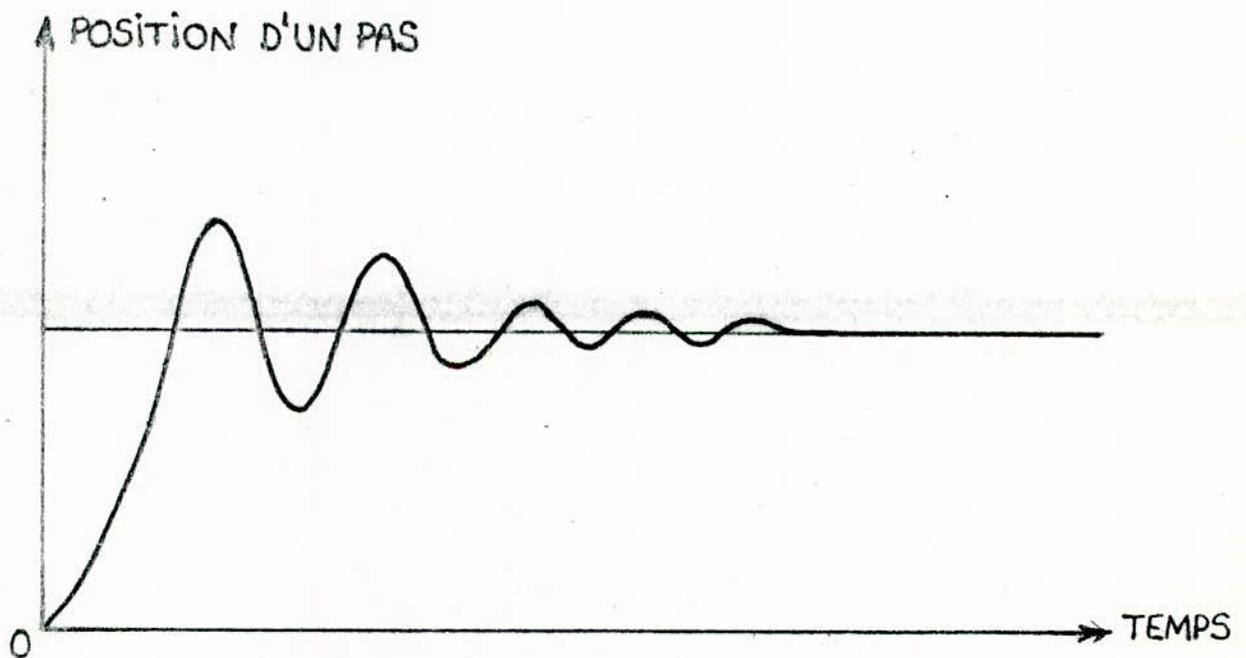
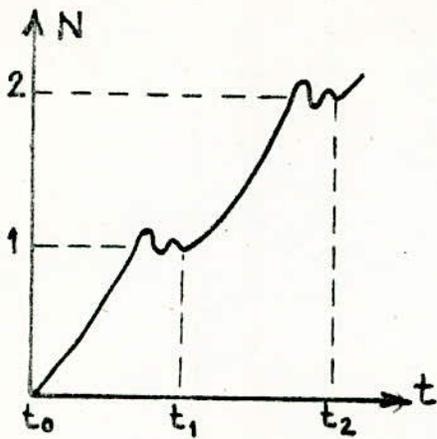
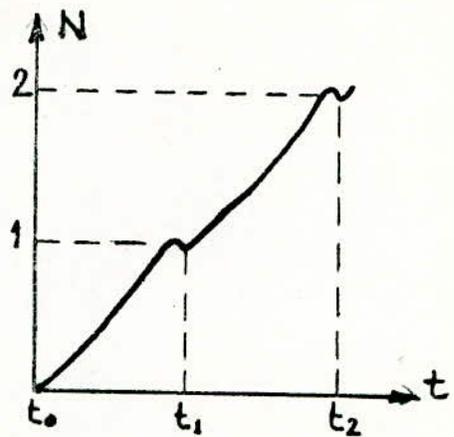


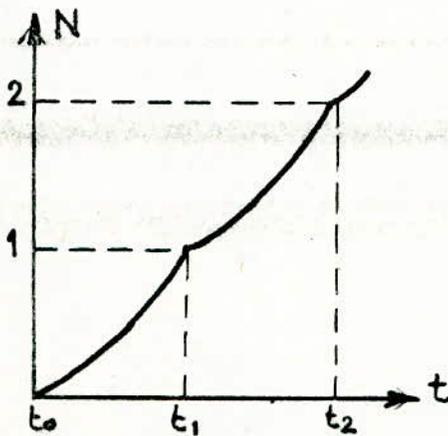
FIGURE III.4-PHENOMENE D'OSCILLATION



FAIBLE FREQUENCE.



FREQUENCE DE RESONANCE.



FORTE FREQUENCE

FIGURE III.5

### III. 4) CONCLUSION

Dans cette partie, nous avons présenté les principes des moteurs pas à pas, leur fonctionnement ainsi que leurs performances.

Du fait que les moteurs à réluctance variable possèdent une bonne résolution et qu'ils peuvent fournir un couple suffisant pour notre application, ils sont bien adaptés à la réalisation de notre système.

On notera cependant que les performances du moteur peuvent être améliorées par le type d'alimentation et par la commande.

## SECONDE PARTIE : ALIMENTATION DES MOTEURS PAS A PAS

Le rôle de l'alimentation est d'assurer une amplification de la commande de façon à appliquer la tension successivement aux différentes phases du moteur.

Etant donné que le courant est le paramètre essentiel pour le couple moteur : pour conserver un couple suffisant il est nécessaire d'alimenter chaque phase avec un courant constant. Celui-ci devra donc atteindre sa valeur finale immédiatement après l'instant de commutation ; l'alimentation doit en outre assurer la coupure du courant dans la phase.

On aboutit ainsi à la forme en créneaux du courant nécessaire au bon fonctionnement du moteur pas à pas (figure III.6).

### III.5) - SCHEMA CLASSIQUE DE L'ALIMENTATION (figure III.7) :

Cette alimentation est constituée d'un transistor travaillant en bloqué-saturé et d'un circuit d'extinction pour éviter les surtensions aux coupures.

Pour réduire le temps d'établissement du courant, une résistance relativement importante est ajoutée en série avec chaque enroulement du moteur.

Un tel schéma présente un inconvénient majeur pour les moteurs de puissance moyenne (50 W à 1KW), en effet la résistance supplémentaire dissipe la plus grande partie de la puissance fournie par l'alimentation et le rendement est affecté.

### III.6) - ALIMENTATION A DEUX NIVEAUX DE TENSION (figure III.8) :

On peut réduire le temps d'établissement du courant avec une moindre dissipation d'énergie en utilisant un montage à surtension.

Cette technique consiste à commuter l'alimentation du niveau  $U_1$  au niveau  $U_0$  nettement plus faible dès que le courant nominal est atteint [6].

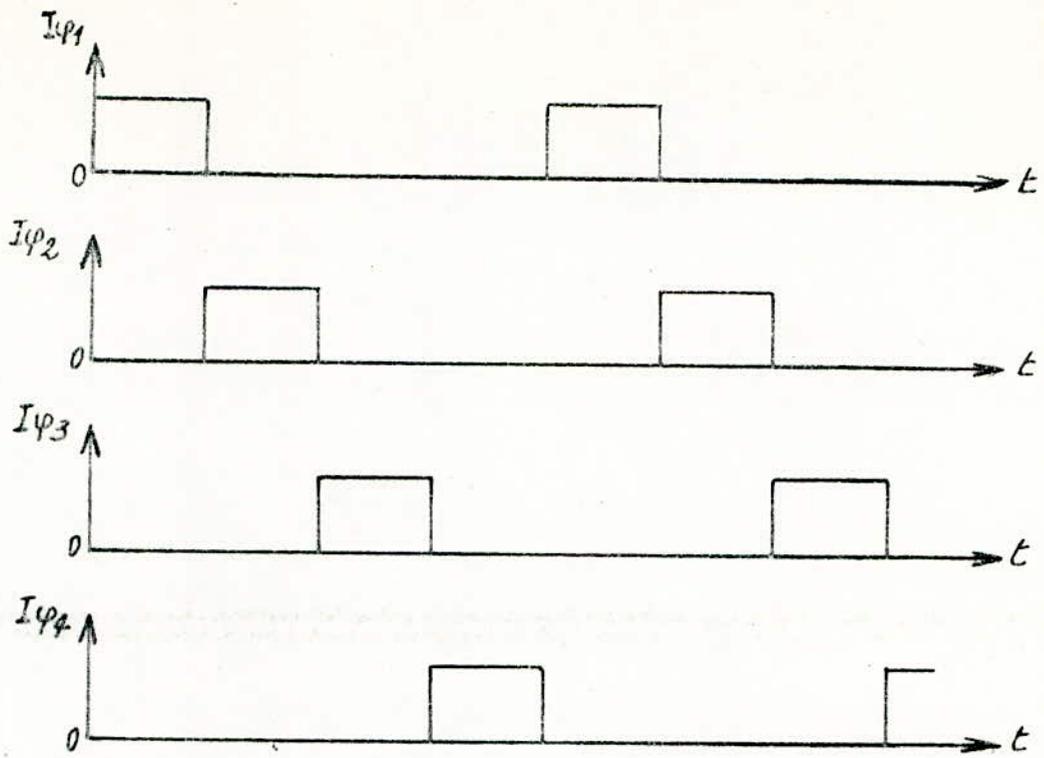


FIGURE III.6 - FORME DU COURANT POUR UN MOTEUR PAS A PAS A QUATRE PHASES.

$T_c$ : TRANSISTOR DE COMMUTATION.  
 D: DIODE DE ROUE LIBRE.  
 $V_{com}$ : TENSION DE COMMANDE.

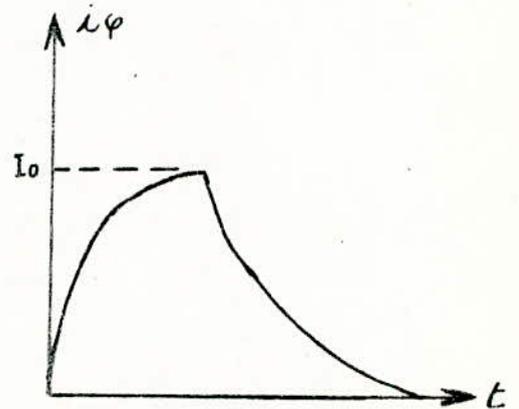
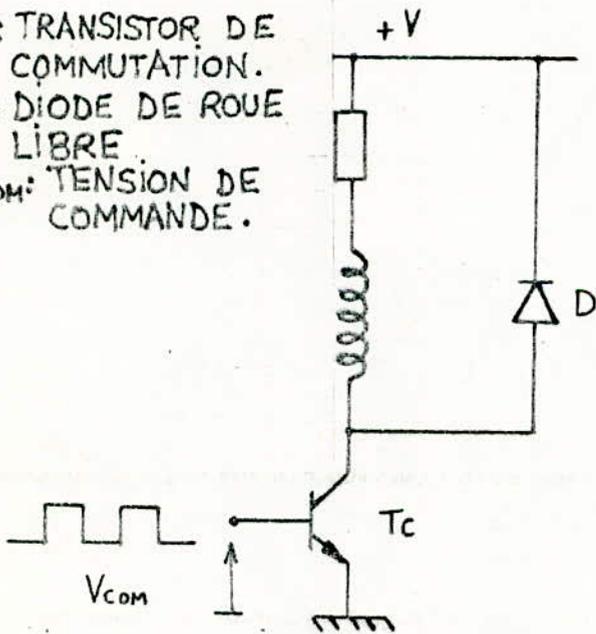


FIGURE III.7 - SCHEMA CLASSIQUE DE L'ALIMENTATION.

La constante de temps apparente d'établissement du courant est réduite dans le rapport suivant :

$$\tau' = \tau \cdot U_0 / U_1, \quad \text{avec } \tau = L/r$$

L'inconvénient de ce montage est qu'il nécessite une alimentation bi-tension.

### III.7) - ALIMENTATION PAR HACHEUR :

On peut améliorer le rendement, tout en conservant un couple suffisant en utilisant une alimentation par hacheur qui permet d'approcher la forme en créneau du courant dans la phase.

Un hacheur permet d'alimenter chaque phase avec un courant découpé (figure III.9).

Pour réaliser la fonction de hachage, on peut utiliser soit des transistors comme le montre le montage de la figure III.10, soit des circuits intégrés. Un schéma d'une telle alimentation pour une phase du moteur pas à pas est donné à la figure III.11.

Celui-ci utilise des comparateurs intégrés montés en bascules de SCHMIDT [7],

Le principe de fonctionnement est simple, la bascule de SCHMIDT compare le courant dans la phase avec la valeur maximale admissible et bascule lorsque le courant atteint cette valeur, ce qui bloque le transistor  $T_C$  et met en service le circuit d'extinction. Le courant décroît et lorsqu'il tombe en dessous de la valeur maximale admissible, la phase est de nouveau alimentée.

La fréquence de hachage est pratiquement fixée par la constante de temps  $R_C \cdot C$ .

On notera enfin que le transistor se bloque lorsque le signal de commande passe au niveau logique zéro.

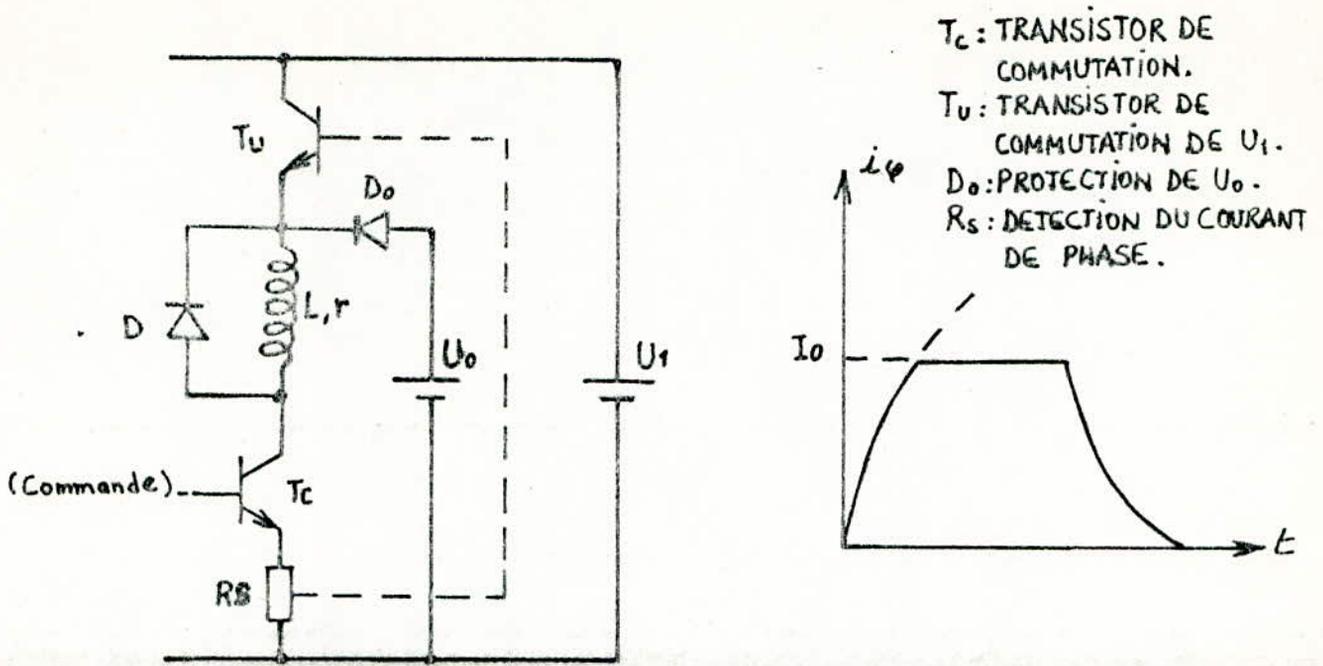


FIGURE III.8 - ALIMENTATION A DEUX NIVEAUX DE TENSION.

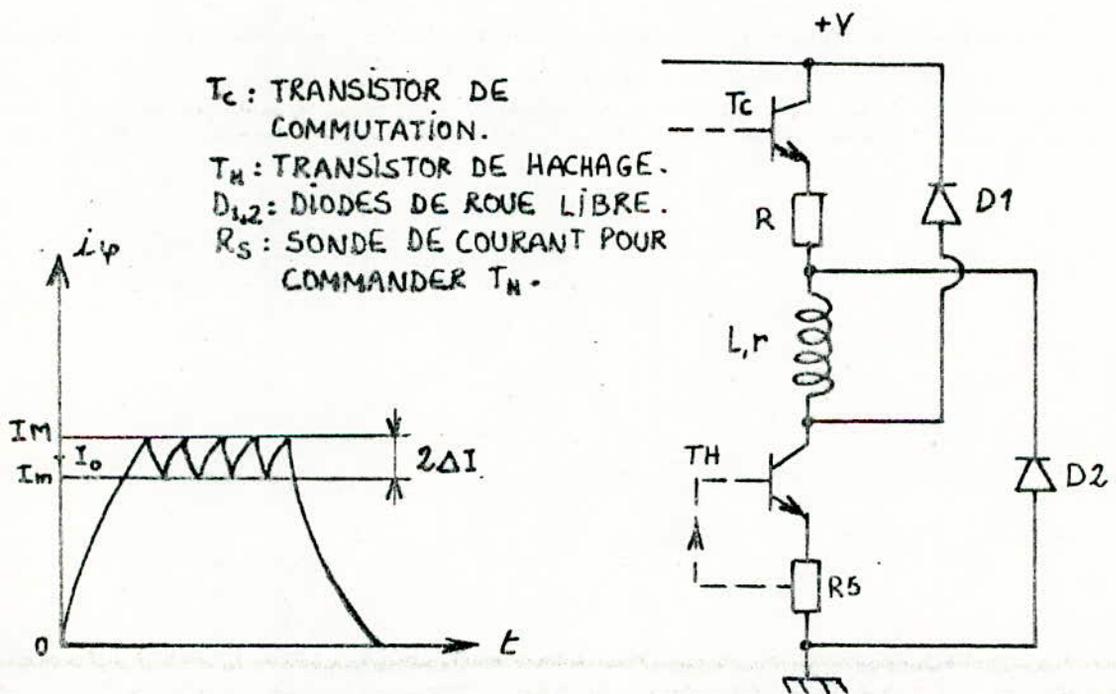


FIGURE III.9 - COURANT OBTENU PAR HACHEUR.

FIGURE III.10 - HACHEUR A TRANSISTOR.

### III.8) - ALIMENTATION PAR HACHEUR PROGRAMMABLE :

Dans les structures des schémas précédents, nous pouvons constater que la charge était supposée constante, ce qui en fait limite le champ d'application de la carte d'alimentation.

On peut remédier à cet inconvénient en utilisant une interface "intelligente" qui dispose d'une structure de commutateurs d'application la plus vaste possible, et qui permet de plus le dialogue entre le composant et le calculateur de commande.

Ainsi donc, suivant le courant que l'on veut obtenir par le hacheur, la stratégie de commande doit-être programmable de même que les temps morts de commutation et la fréquence de répétition, et ce pour une protection efficace des organes de puissance [8].

Le calculateur de commande peut réaliser une telle tâche, mais pour soulager au maximum le travail du microprocesseur la carte d'interface doit disposer de sa propre unité centrale.

La structure de hacheur adaptée à cette interface est la structure en H à quatre quadrants dont le schéma est donné en figure III.12.

Les stratégies de commande de cet hacheur sont au nombre de quatre :

- la stratégie bipolaire qui consiste à fermer cycliquement simultanément  $S_1$  et  $S_3$  puis  $S_2$  et  $S_4$ .
- La stratégie unipolaire positive qui consiste à laisser  $S_4$  et  $S_3$  fermés en permanence.  $S_1$  et  $S_2$  ayant des états complémentaires pour chaque cycle.
- La stratégie unipolaire négative : les rôles précédents de  $S_1$  et  $S_3$  et de  $S_2$  et  $S_4$  sont inversés.
- La stratégie unipolaire modifiée : celle-ci est explicitée par la figure III.13

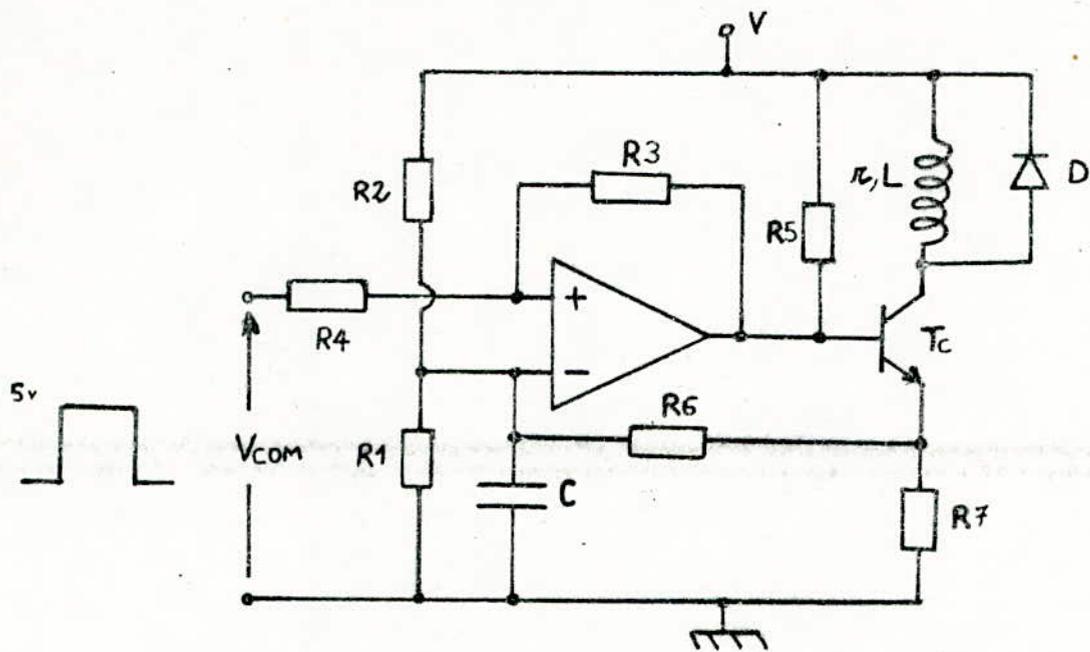


FIGURE III.11-SCHEMA DE HACHEUR A BASCULE DE SCHMIDT.

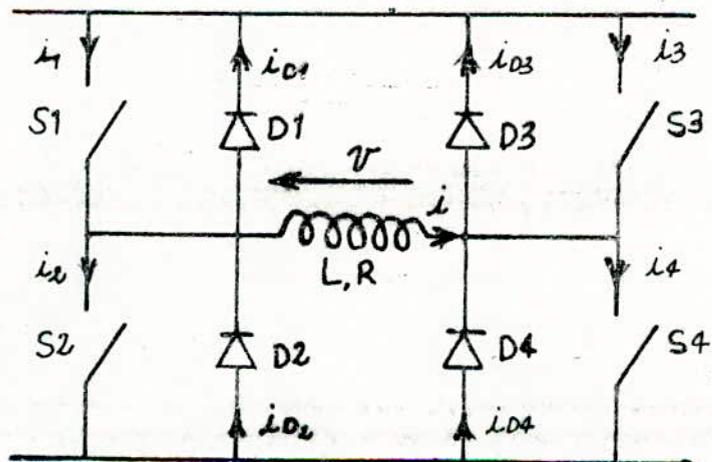
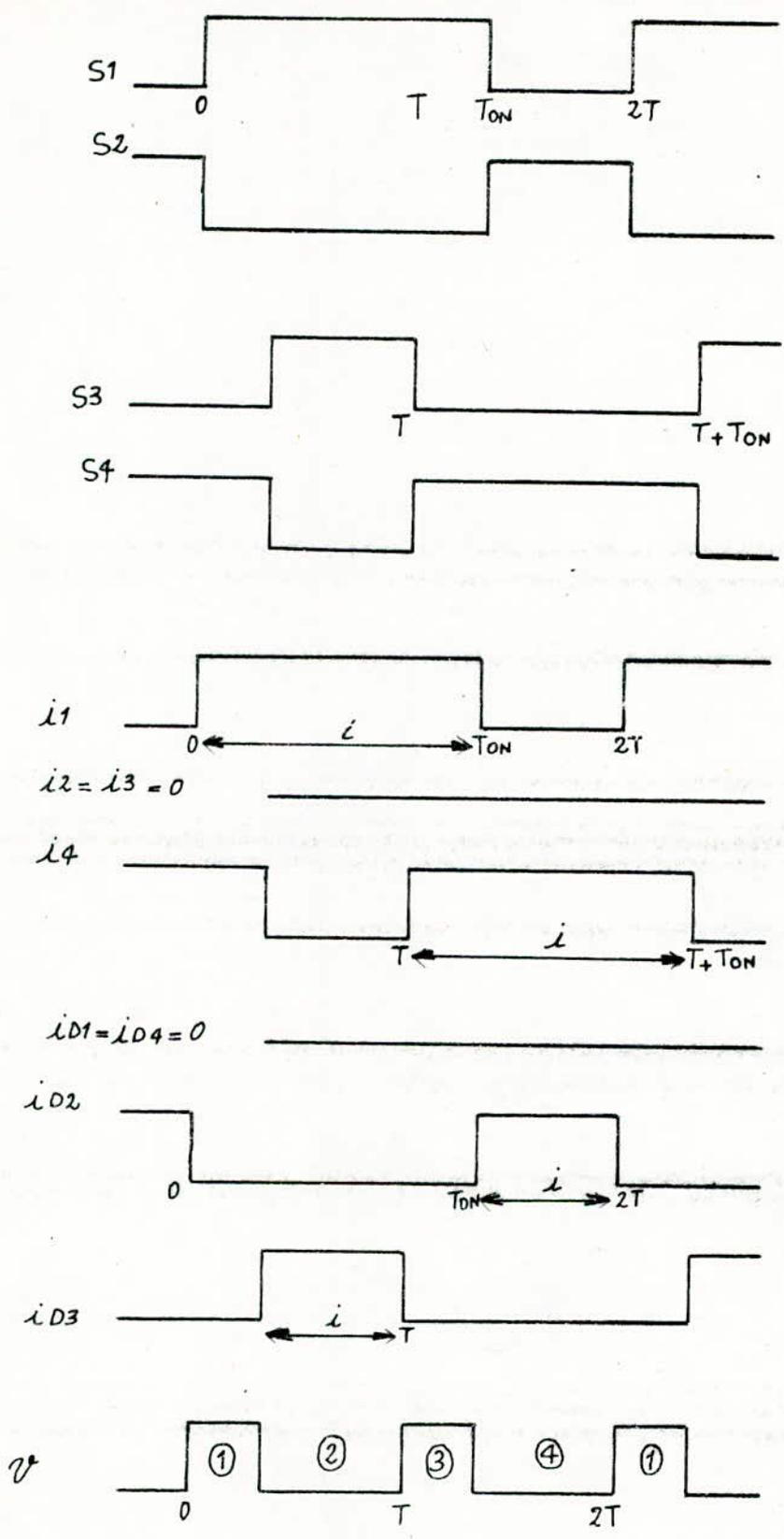


FIGURE III.12-HACHEUR A QUATRE QUADRANTS EN H



- ①: CONDUCTION DE S<sub>1</sub> ET S<sub>4</sub>.
- ②: CONDUCTION DE S<sub>1</sub> ET D<sub>3</sub>.
- ③: CONDUCTION DE S<sub>3</sub> ET S<sub>4</sub>.
- ④: CONDUCTION DE S<sub>4</sub> ET D<sub>2</sub>.

FIGURE III.13 - STRATEGIE DE COMMANDE UNIPOLAIRE MODIFIEE.

### III.9.) - CONCLUSION

Le but recherché dans cette partie est d'améliorer les performances du moteur pas à pas en présentant différents schémas d'alimentation.

En effet, comme le couple maximum diminue avec la fréquence par suite de la diminution du courant moyen, liée à l'accroissement de la constante de temps d'établissement du courant, l'amélioration par le biais de l'alimentation a pour but de transformer, idéalement, la source de tension en source de courant. On y parvient soit en utilisant le schéma de tension à deux niveaux, soit en ayant recours à un hacheur.

Le hacheur à comparateurs intégrés possède une structure simple, de plus il permet d'avoir un bon rendement et une réduction de l'encombrement de la carte d'alimentation ; nous opterons donc pour un tel schéma.

### TROISIEME PARTIE : REALISATION DE L'INTERFACE DE PUISSANCE

#### III.10) - FONCTIONNEMENT DU CIRCUIT HACHEUR :

Le circuit de réalisation de l'alimentation pour une phase du moteur pas à pas est schématisé à la figure III.14.

Dans le fonctionnement de ce circuit, il faut distinguer deux étapes :

- lorsque le signal de commande VCOM est au zéro logique, la sortie du comparateur est à zéro et le transistor de puissance est bloqué.

Comme le zéro logique ne correspond pas nécessairement à un niveau de tension nul, l'entrée inverseuse est polarisée à une tension positive de 0,4 V grâce au pont de résistances  $R_1/R_2$ .

Dans ce cas, le hacheur est en régime d'extinction.

- Le fonctionnement en régime de hachage, lorsque VCOM est au niveau logique un ; le transistor est saturé et alimente la phase du moteur.

Le courant dans celle-ci est mesuré grâce à la résistance de faible valeur  $R_7$  ; la tension aux bornes de cette résistance, qui est proportionnelle au courant dans la phase, est recopiée par la capacité C avec une constante de temps :

$$\tau = R_6 C.$$

Lorsque le courant dans la phase du moteur atteint le premier seuil de basculement  $V_+$  du trigger de SCHMIDT, celui-ci bascule et le transistor est bloqué.

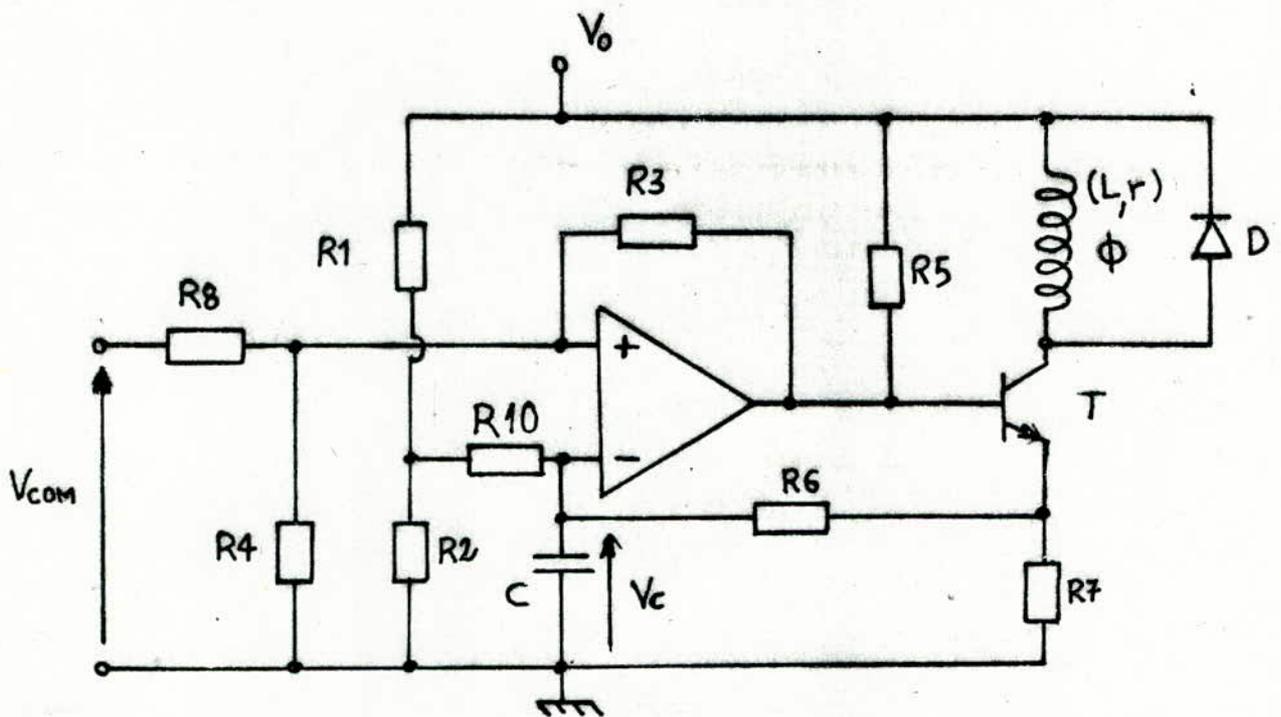


FIGURE III.14 - SCHEMA DU CIRCUIT D'ALIMENTATION ADOPTE.

A ce moment, la diode de roue libre D permet l'extinction du courant pendant que la capacité C se décharge à travers les résistances  $R_6$  et  $R_7$  ( $R_7 \ll R_6$ ).

Lorsque la tension aux bornes de la capacité tombe en dessous du second seuil de basculement  $V_-$ , le trigger rebascule et permet à nouveau l'alimentation de la phase.

Les tensions de seuil de la bascule de SCHMIDT sont fixées par les résistances  $R_8/R_4$  et  $R_3$ .

L'application du théorème de superposition au calcul de  $V_+$  et  $V_-$  donne :

$$V_+ = \frac{(R_3//R_4)}{(R_3//R_4) + R_8} V_{COM} + \frac{(R_8//R_4)}{(R_8//R_4) + R_3} V_0$$

$$V_- = \frac{(R_3//R_4)}{(R_3//R_4) + R_8} V_{COM}$$

Pour avoir un fonctionnement correct du hacheur pour les vitesses élevées du moteur les tensions de seuil  $V_+$  et  $V_-$  doivent être proches. Pour cela, on doit avoir :

$$(R_8//R_4) \ll R_3$$

Une résistance élevée  $R_{10}$  relie le pont  $R_1/R_2$  à l'entrée inverseuse de l'amplificateur opérationnel ; cette résistance permet de limiter le courant de charge et de décharge de la capacité C à travers le pont  $R_1/R_2$ , et ce, pour ne pas perturber la fréquence de découpage.

La diode de roue libre D aux bornes de chaque phase du moteur est une diode de puissance à faible temps de commutation.

Le transistor T est un transistor de commutation de puissance à fréquence de transition  $f_T$  assez élevée.

Ceci permettra de minimiser les temps de commutation et d'améliorer le rendement.

### III.11)- REALISATION DU CIRCUIT :

Le circuit imprimé de la carte d'interface de puissance, pour un moteur pas à pas à quatre phases, est donné à la figure III.15 Un buffer est utilisé pour adapter cette interface à la carte à microprocesseur qui génère les signaux de commande VCOM..

L'adaptation permet de protéger cette carte contre des surcharges accidentelles.

Le montage utilise une alimentation commune  $V_{CC} = 12 V$ .

Le transistor T est du type BDX 33C. C'est un Darlington qui peut supporter un courant collecteur maximal  $I_{Cmax}$  de 12 A et une tension collecteur maximale  $V_{cemax}$  de 100V.

Sa fréquence de transition est  $f_T = 20 MHz$ .

La diode de roue libre est du type BY 249 dont le temps de recouvrement inverse est de 35 ns.

La résistance  $R_G$  est fixée à  $2,2 K\Omega$  et la capacité C est égale à 22 nF ; ce qui donne une fréquence de hachage d'environ :

$$f_H = \frac{1}{R_G \cdot C} = \frac{1}{22 \cdot 10^{-9} \cdot 2,2 \cdot 10^3} = 20,6 KHZ.$$

Les résistances  $R_8$ ,  $R_4$  et  $R_3$  sont respectivement de  $33 K\Omega$ ,  $10 K\Omega$  et  $220 K\Omega$ , d'où l'on déduit les seuils de basculement du trigger de SCHMIDT :

$$V_+ = 1,12 V + 0,4 V = 1,5 V$$

$$V_- = 1,12 V.$$

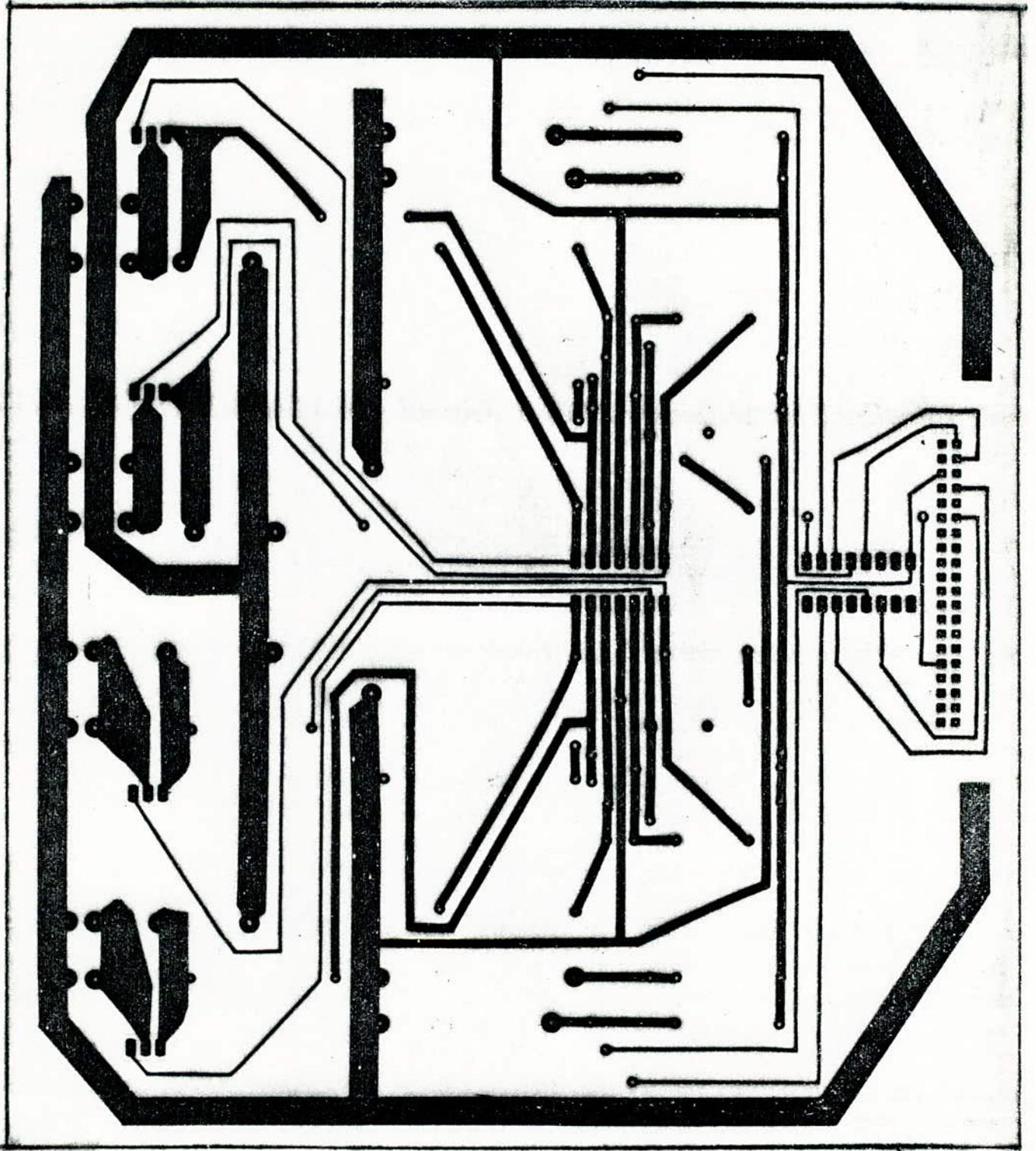


FIGURE III.15 - CARTE D'INTERFACE DE  
PUISSANCE.

La sonde du courant de phase ayant une valeur  $R_7 = 0,5 \Omega$ , le courant nominal  $I_N$  que l'on obtient est de :

$$I_N + \Delta I = \frac{V_+}{R_7} = \frac{1,5}{0,5} = 3 \text{ A.}$$

$$I_N - \Delta I = \frac{V_-}{R_7} = \frac{1,12}{0,5} = 2,2 \text{ A.}$$

$$\text{Soit : } I_N = \frac{3+2,2}{2} = 2,6 \text{ A.}$$

La forme du courant dans la phase du moteur est montrée à la figure III.16.

Si l'on désire alimenter un moteur qui nécessite une valeur nominale du courant différente de la valeur obtenue, on peut agir de deux manières :

- soit modifier la résistance de détection de courant  $R_7$  en fonction du courant à mesurer, pour les mêmes seuils de basculement  $V_+$  et  $V_-$ .

- Soit conserver  $R_7$  et modifier le rapport  $R_4/R_8$  pour adapter les seuils de basculement à la tension aux bornes de  $R_7$  parcourue par le courant nécessaire au nouveau moteur.

La liste des composants ainsi que leur schéma de brochage sont donnés en annexe.

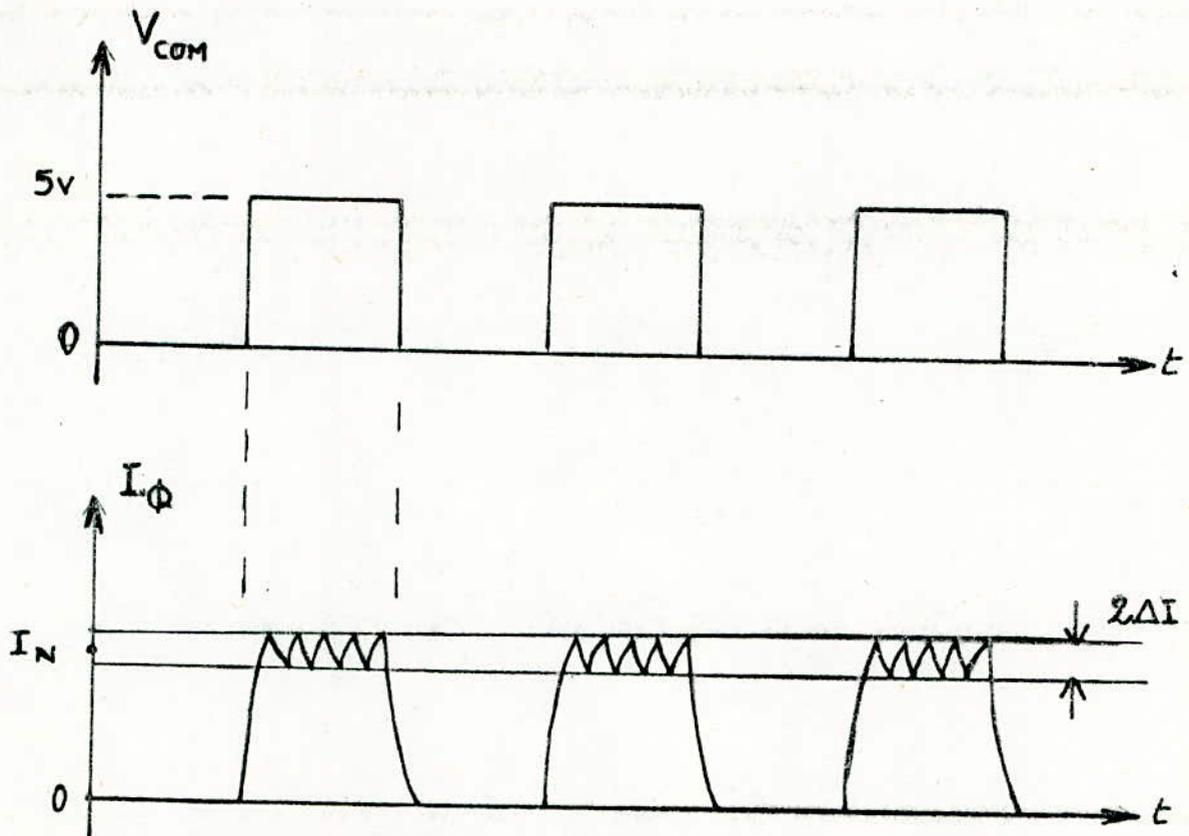


FIGURE III.16 - FORME DU COURANT DE PHASE OBTENUE AVEC L'INTERFACE DE PUISSANCE.

### III.12)- CONCLUSION :

Nous avons présenté dans cette partie la réalisation de la carte d'interface de puissance pour l'alimentation d'un moteur pas à pas à quatre phases.

Grâce à cette carte, le pilotage d'un moteur pas à pas par microprocesseur est rendu possible.

En effet, le hacheur réalisé alimente la phase du moteur avec un courant en forme de créneaux.

Dans le but de commander ce type de moteur, une carte à microprocesseur sera connectée à notre interface de puissance.

Nous expliciterons cette opération dans le prochain chapitre, qui traitera aussi du logiciel de commande associé.

# CHAPITRE 4

LA CARTE ET LE LOGICIEL DE  
COMMANDE

La commande générale du système ordonne et gère l'opération de perçage des trous du circuit imprimé grâce à un ensemble de routines appropriées.

Chacune d'elles provoque et coordonne les mouvements des moteurs de façon spécifique.

La commande des moteurs pas à pas assure la succession des ordres de commutation de phase à un rythme compatible avec la fonction à réaliser.

Son rôle est déterminant en ce qui concerne les performances à hautes fréquences et en régime d'accélération ou de décélération.

Nous avons vu au chapitre précédent que le moteur pas à pas avait un caractère synchrone, ce qui permet de le commander sans asservissement ; néanmoins, on peut effectuer un contrôle de la position pour assurer une meilleure précision (figure IV.1).

La méthode qui permet d'avoir les meilleures performances et une grande flexibilité est la commande électronique par microprocesseur.

#### 1ère PARTIE : LA CARTE DE COMMANDE :

Nous avons vu au Chapitre I qu'on pouvait prévoir deux possibilités pour la réalisation de la carte de commande.:

- le système spécifique à un compatible IBM-PC

Dans ce cas la carte ne peut-être utilisée directement avec un micro-ordinateur non compatible PC, car celle-ci étant connectée directement sur le bus PC, n'interprète que les signaux émis par la carte mère du PC.

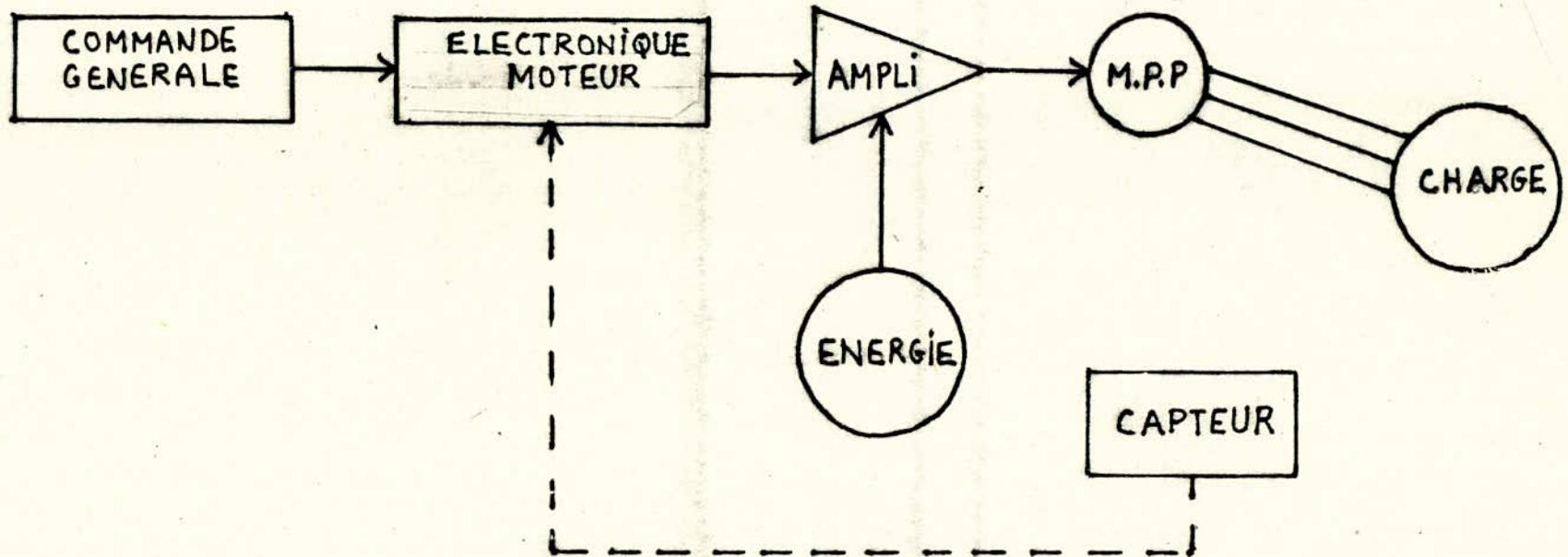


FIGURE IV.1

- Le système indépendant qui dispose de sa propre unité centrale et qui peut-être programmé soit localement soit à distance, à travers une liaison série asynchrone.

Nous opterons pour cette dernière possibilité, qui contrairement à la première, permet de concevoir un système flexible.

Dans ce but, nous utilisons une carte à microprocesseur (la TM 990/189 UNIVERSITY de TEXAS INSTRUMENTS) qui dispose d'un port d'entrée-sortie parallèle, et d'une interface de communications asynchrone à la norme RS 232C.

#### IV.1- ) - LA CARTE TM 990/189 UNIVERSITY :

C'est une carte construite autour du microprocesseur 16 bits TMS 9980. Celui-ci dispose d'un bus de données de 8 bits, de 16 registres de travail définis en mémoire par l'utilisateur et d'un jeu d'instructions incluant la multiplication et la division.

La carte est dotée d'un assembleur symbolique permettant de programmer directement en mnémoniques.

De plus, cette carte s'interface avec les dispositifs d'entrée-sortie par l'intermédiaire de deux ports :

#### IV.1.1) - L'interface programmable multifonctions TMS 9901 :

Le TMS 9901 est un circuit d'interface qui permet de gérer 22 lignes d'entrée/sortie/interruption ainsi qu'un décompteur programmable permettant de réaliser une horloge temps réel (figure IV.2).

Cette interface programmable se connecte au microprocesseur TMS 9980 par l'intermédiaire du registre de communications série (CRU).

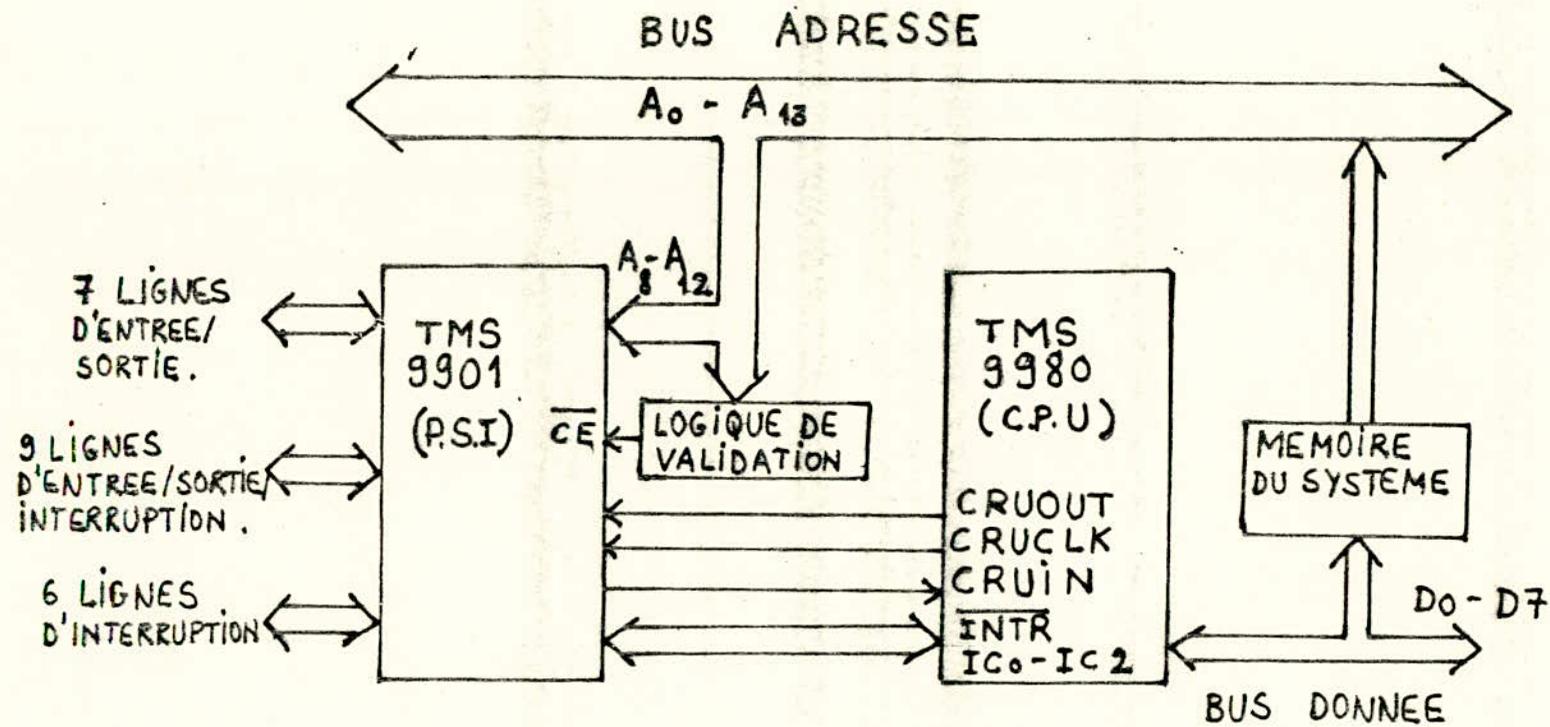


FIGURE IV.2- EXEMPLE D'APPLICATION UTILISANT LE TMS 9901PSI.

Celui-ci est une voie d'entrée/sortie particulière entièrement réservée aux échanges de données entre le processeur et les dispositifs externes [9].

Ce mode de transmission série se fait par l'intermédiaire de 03 Lignes de contrôle CRU indépendantes du bus de données, et sans interaction avec la gestion du système mémoire.

Ces lignes servent à la réception des données en série (CRUIN), à l'émission de données en série (CRUOUT) et à leur échantillonnage (CRUCLK) [10].

Les lignes d'entrée/sortie/interruption se décomposent en trois groupes :

- le groupe 1 comprend 06 lignes pouvant être configurées individuellement comme entrées ou interruptions.
- Le groupe 2 comprend 09 lignes pouvant être configurées individuellement comme entrées ou sorties.
- Le groupe 3 comprend 07 lignes pouvant être configurées individuellement comme entrées ou sorties.

Il existe trois instructions permettant de réaliser des manipulations sur un bit.

Il s'agit de :

- SBO : positionnement à un d'un bit.
- SBZ : positionnement à zéro d'un bit.
- TB : test de la valeur d'un bit.

Dans tous les cas, l'adresse du bit est calculée à partir d'une base contenue dans les bits 4 à 14 du registre 12, et d'un déplacement signé défini par le champ opérande de l'instruction.

Le jeu d'instructions autorise également un échange de données entre mémoire et CRU.

Il s'agit des instructions :

- LDCR : transfert de 1 à 16 bits de la mémoire vers le CRU.
- STCR : transfert de 1 à 16 bits du CRU vers la mémoire.

L'adresse du premier bit transmis est chargée dans les bits 4 à 14 du registre 12.

Le nombre de bits à transmettre est spécifié dans le champ opérande.

#### IV.1-2) Le contrôleur de communications asynchrone TMS 9902 :

Il permet l'interface entre le microprocesseur et un canal de communications asynchrone conformément au protocole RS 232C.

Tous les paramètres de la transmission sont programmables (parité, bits stop, vitesse de transmission, nombre de bits par caractère...).

La figure IV.3 montre l'utilisation du TMS 9902 comme interface entre le microprocesseur TMS 9980 et un terminal.

L'échange de données entre le microprocesseur et le contrôleur de communications asynchrone se fait par l'intermédiaire du CRU.

#### IV. -2) - Utilisation de la carte TM 990/189 pour la réalisation de la carte de commande :

Nous avons vu que l'interface programmable TMS 9901 disposait de trois groupes de lignes d'entrée/sortie/interruption.

Du fait que notre système utilise trois moteurs (déplacements X,Y et Z) de quatre phases chacun, nous avons réservé 12 lignes, parmi les groupes 2 et 3, configurées en entrée/sortie.

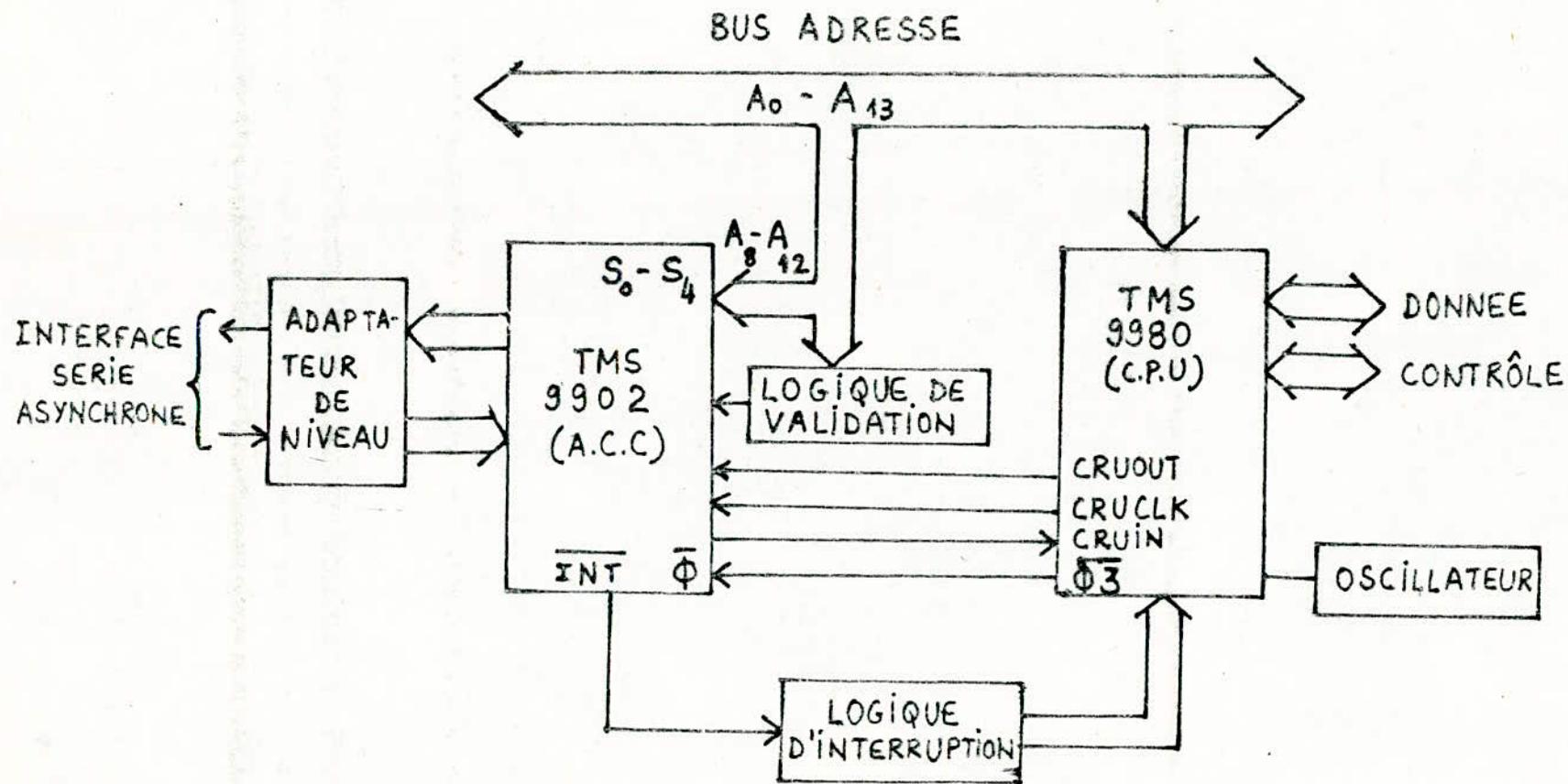


FIGURE IV.3-EXEMPLE D'APPLICATION DU TMS 9902 ACC.

Ces lignes sont affectées de la manière suivante :

- moteur X, lignes P4 à P7.
- moteur Y, lignes P8 à P11.
- moteur Z, lignes P12 à P15.

Chaque ligne réservée est connectée à un étage de puissance de la carte d'alimentation et permet de commander une phase de l'un des moteurs.

Cette commande se fera sans asservissement. Ainsi reliées, la carte TEXAS et l'interface de puissance forment la carte de commande de notre système.

Etant donné que celle-ci sera pilotée par un compatible IBM-PC, elle sera connectée à celui-ci via le contrôleur de communications asynchrone.

La figure IV.4 donne le schéma du circuit de commande générale du système de perçage .

Pour que ce circuit puisse réaliser les opérations désirées, il est nécessaire de lui associer un logiciel permettant de gérer tout le système.

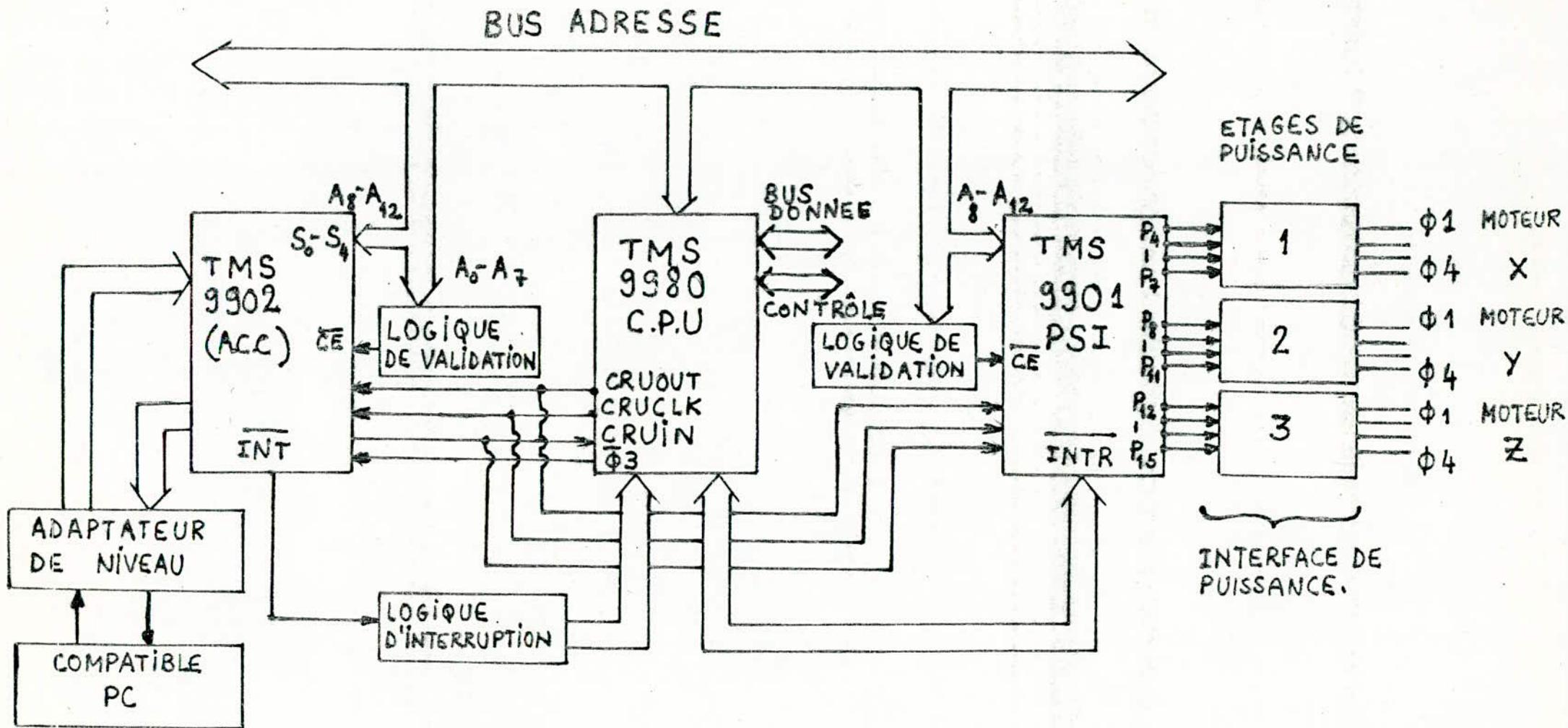


FIGURE IV.4 - CIRCUIT GENERAL DE COMMANDE DU SYSTEME DE PERÇAGE.

## SECONDE PARTIE : LE LOGICIEL DE GESTION

Le logiciel de gestion du système de perçage se compose de deux parties :

- une partie exécutée sous compatible IBM-PC réalise le décodage du fichier image du circuit imprimé, le transcodage des données ainsi récupérées et l'envoi d'un ordre d'exécution à la carte de commande à travers le canal de communications asynchrone.

- Une autre partie exécutée par la carte effectue un traitement des données envoyées par le PC pour générer ensuite les signaux de commande nécessaires à l'exécution de l'ordre émis.

### IV.3) - DECODAGE DU FICHIER IMAGE DU CIRCUIT IMPRIME :

Le but de cette opération est la récupération, par programme, des coordonnées et du diamètre des trous.

Chaque logiciel de conception de circuits imprimés possède une manière spécifique de coder ses fichiers image, il est donc très difficile d'effectuer un décodage direct de ces fichiers ; car ceci reviendrait à une recherche du code utilisé par essais successifs.

Une méthode originale permettant de contourner cette difficulté est de concevoir une routine résidente en mémoire qui effectue une reconnaissance des paramètres recherchés au niveau du buffer d'écran ; cette routine fonctionne normalement même quand d'autres logiciels s'exécutent, en particulier le logiciel de conception de circuits imprimés.

Celle-ci est conçue de façon à être appelée chaque fois que l'on appuie sur une touche du clavier spécialisée à cet effet.

Dans notre cas, nous utilisons la touche BREAK.

Ceci est fait en détournant l'interruption 1B(H) (BREAK clavier) du BIOS (Basic input/output system).

Les appels au BIOS se font par des interruptions logicielles ; pour réaliser convenablement ces appels, il est indispensable de connaître la structure du microprocesseur INTEL 8086 utilisé par les compatibles IBM-PC.

#### IV.3-1) - ARCHITECTURE DU 8086 :

La figure IV.5 illustre les registres internes de ce microprocesseur.

On y trouve quatre registres d'usage général de 16 bits notés AX, BX, CX et DX, chacun d'eux pouvant être divisé en deux registres de 8 bits.

SI et DI sont deux registres d'index. Viennent ensuite Bp, pointeur de base et SP, pointeur de pile, utilisé en particulier pour sauvegarder l'adresse de retour des sous-programmes.

Le pointeur d'instructions est noté IP et le registre d'état FLAGS.

Les quatre derniers registres CS, DS, ES et SS sont appelés registres de segment.

En effet, la mémoire adressable par le 8086 est segmentée. Une adresse logique est composée d'un segment et d'un offset ; la correspondance avec une adresse physique, émise sur le bus d'adresse de 20 bits, est donnée par :

$$\text{Adresse physique} = (16 \times \text{segment}) + \text{offset}.$$

La multiplication par 16 revient à un décalage de 4 bits vers la gauche, ce qui correspond bien à une adresse physique de 20 bits.

Le 8086 cherche systématiquement ses instructions dans le segment de code pointé par CS ; IP indique alors l'offset de l'instruction dans ce segment.

b15

b0

AH	AL	AX
BC	BL	BX
CH	CL	CX
DH	DL	DX

Source index	SI
Destination index	DI

Base pointer	BP
Stack pointer	SP

Instruction pointer	IP
---------------------	----

Status	FLAGS
--------	-------

Code segment	CS
Data segment	DS
Extra segment	ES
Stack segment	SS

FIGURE IV.5 - REGISTRES INTERNES DU 8086.

De même, l'adressage des données se fait dans le segment pointé par DS ; enfin SP et BP indiquent un offset dans le segment de pile pointé par SS [11].

#### IV.3-2) - DETOURNEMENT DE L'INTERRUPTION "BREAK CLAVIER" DU BIOS :

Cette interruption logicielle est appelée par la routine d'interruption du clavier lorsqu'un "BREAK" est détecté.

Cette routine exécute une instruction de retour d'interruption (IRET) [12].

Pour détourner cette interruption de son usage d'origine, on utilise les deux services de MS-DOS (système d'exploitation) 35 (H) et 25 (H) autorisant la manipulation d'un vecteur d'interruption en fournissant son numéro [13].

Les appels aux services de MS-DOS se font via l'interruption logicielle 21 (H).

Les fonctions 35 (H) et 25 (H) sont définies et utilisées comme suit [14]:

\* service 35 (H) : GET INTERRUPT VECTOR VALUE.

Récupération de l'adresse d'une routine d'interruption par lecture dans la table des vecteurs d'interruption.

- paramètres d'entrée :

AH = 35 (H).

AL = numéro du vecteur d'interruption concerné (dans notre cas, c'est l'interruption 1B (H)).

- paramètres de sortie :

Le segment de la routine d'interruption est retourné dans le registre ES, et son offset dans BX.

\* service 25 (H) : SET INTERRUPT VECTOR

Installation d'une nouvelle routine d'interruption par écriture de son adresse dans la table des vecteurs d'interruption.

- paramètres d'entrée :

AH = 25(H)

AL = numéro du vecteur d'interruption concerné (1B(H) dans notre cas).

DS/DX = segment/offset de la routine d'interruption (dans notre cas, c'est la routine de reconnaissance de pastilles).

- paramètres de sortie :

Aucun.

Jusqu'à présent, nous avons détourné l'interruption BREAK (1B(H)) du BIOS de façon qu'elle exécute la routine de reconnaissance des paramètres du perçage.

Pour que cette routine s'exécute chaque fois que l'on appuie sur la touche BREAK du clavier, même lorsqu'un logiciel résident s'exécute, il est nécessaire de la rendre résidente en mémoire.

Pour cela, MS-DOS procure le service 31 (H) qui déclare occupée la zone mémoire concernée [14].

Cette fonction est définie et utilisée comme suit :

\* service 31 (H) : TERMINATE, STAY RESIDENT.

Termine le programme en cours, réserve la mémoire demandée et rend le contrôle à l'appelant.

- paramètres d'entrée :

AH = 31 (H)

DX = blocs de 16 octets demandés.

- paramètres de sortie :

Aucun.

L'organigramme général de la méthode est donné à la figure IV.6.

Le programme principal correspondant installe la nouvelle routine d'interruption du vecteur 1B (H) après avoir pris soin de sauvegarder l'adresse de l'ancienne.

Le nouveau sous-programme d'interruption sauvegarde le contexte du microprocesseur dans la pile, autorise de nouveau les interruptions matérielles (car le BIOS les utilise pour ses propres besoins), effectue la reconnaissance et la sauvegarde sur disque des paramètres du circuit imprimé, et après avoir restauré le contexte fait un saut à l'adresse de l'ancienne routine d'interruption pour le cas où le vecteur 1B (H) était utilisé par d'autres logiciels résidents. Nous leur laissons ainsi la possibilité de s'en servir.

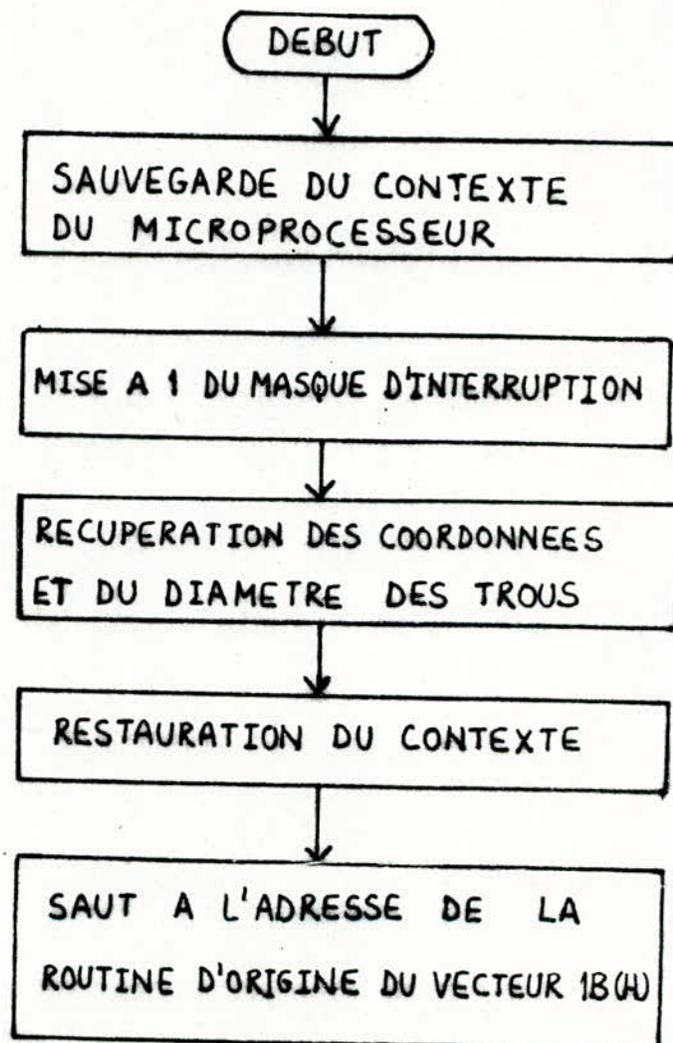
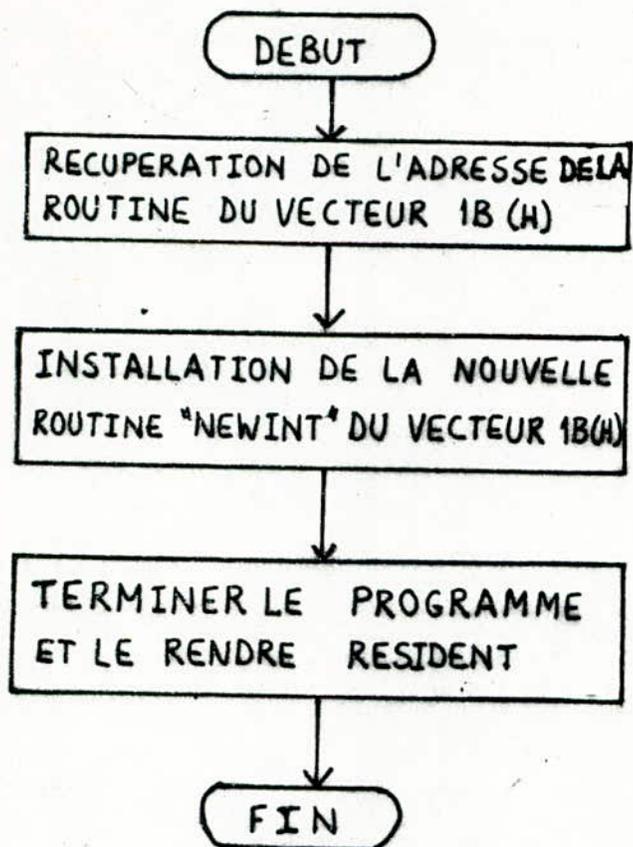
Le programme principal appelle à la fin le service 31 (H) de MS-DOS qui termine le programme et le rend résident.

La programmation d'un tel algorithme en assembleur 8086 fournirait une grande rapidité d'exécution et une économie de mémoire, mais la mise au point du programme risque d'être longue et fastidieuse.

Comme la capacité mémoire disponible sur les compatibles PC est généralement suffisante, nous avons opté dans cette partie pour un langage évolué plus lent (comparativement à l'assembleur), mais qui fournit une grande souplesse et qui permettra d'effectuer facilement toute modification ultérieure éventuelle.

Le langage choisi est le TURBO-PASCAL, car il possède les qualités requises pour la programmation de cet algorithme. En effet [15] :

- il peut s'interface facilement au langage machine grâce à l'instruction `INLINE` (code machine en ligne).



SOUS-PROGRAMME "NEWINT"

FIGURE IV.6 - DETOURNEMENT DE L'INTERRUPTION 1B (H) DU BIOS.

† il dispose de deux instructions permettant d'utiliser les services du BIOS et de MS-DOS, il s'agit respectivement de :

. INTR (n,register).

. MSDOS (register).

où "n" est le numéro de l'interruption et "register" une variable permettant l'échange de données avec les registres du microprocesseur.

- il permet enfin d'obtenir facilement les renseignements nécessaires à l'installation d'une nouvelle routine d'interruption, à savoir le segment de code par la fonction CSEG et l'offset de la routine par la fonction OFS.

Pour récupérer les coordonnées et le diamètre des trous, il faut exécuter le sous-programme "NEWINT" (par appui sur la touche BREAK) lorsque le logiciel de circuits imprimés est installé et que le circuit est affiché à l'écran.

Comme les logiciels de circuits imprimés représentent les pastilles, les pistes et les composants avec des couleurs différentes, nous effectuons la reconnaissance des pastilles en testant la couleur de tous les points de l'écran (320 x 200 points en mode graphique moyenne résolution) ; dès qu'un point ayant la couleur d'une pastille est repéré, le programme cherche à cerner le contour pour retrouver la forme particulière de la pastille.

Lorsque ceci se réalise, on sauvegarde sur disque les coordonnées et le diamètre du trou. Ce dernier est déterminé en comptant les points ayant la couleur du fond à l'intérieur de la pastille.

Les coordonnées des trous ayant le même diamètre sont classés ensemble dans le fichier disque.

L'organigramme de cette reconnaissance est donné à la figure IV.7.

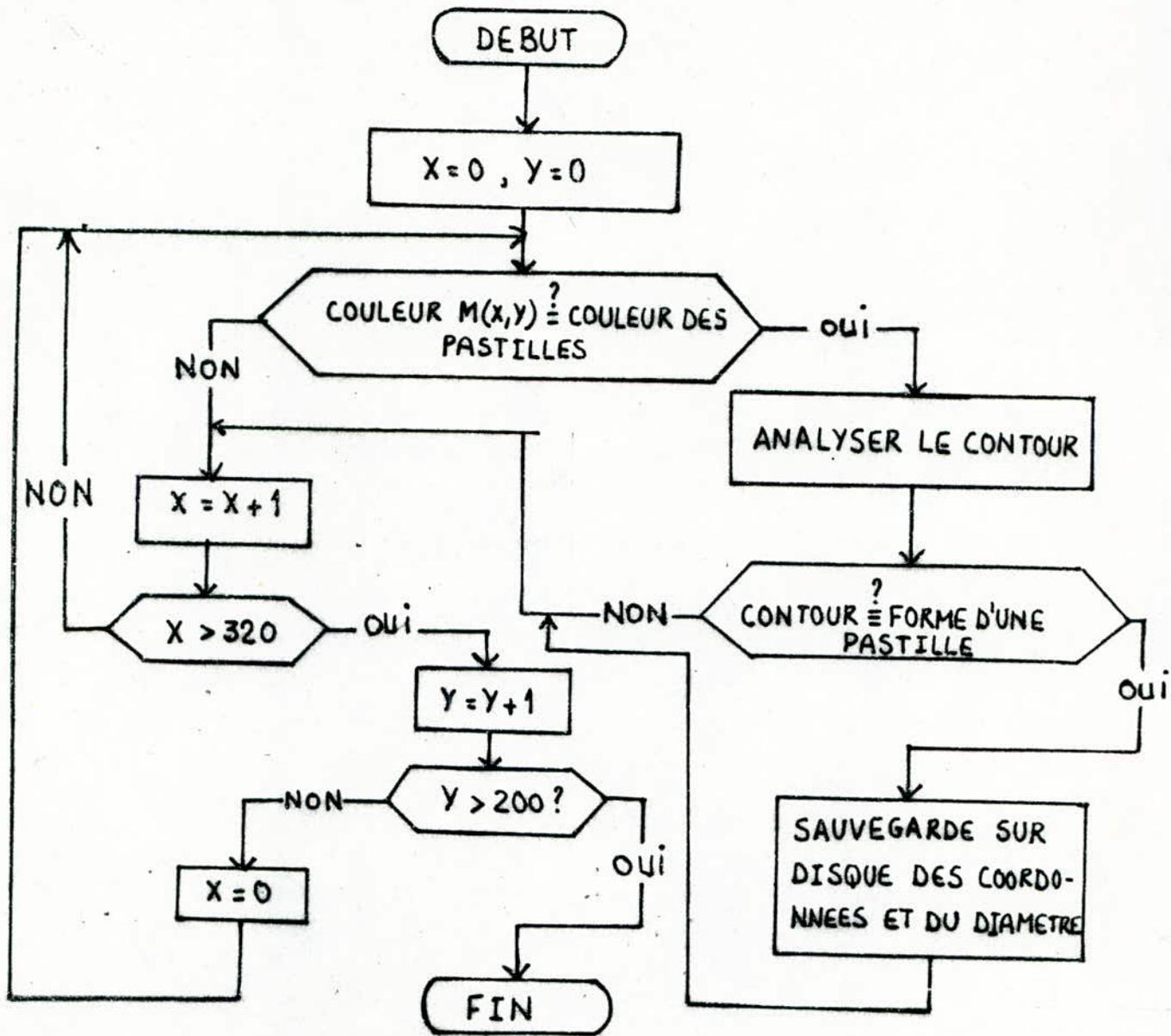


FIGURE IV.7- SOUS-PROGRAMME DE RECONNAISSANCE  
DES PASTILLES.

#### IV.4) - TRAITEMENT PAR LA CARTE DE COMMANDE :

##### IV.4-1) - TECHNIQUES DE COMMANDE DU MOTEUR PAS A PAS :

Comme nous l'avons vu, la technique de commande du moteur pas à pas la plus simple consiste à alimenter cycliquement une seule phase du moteur à la fois.

Cependant, le moteur est caractérisé par une fréquence limite de démarrage au delà de laquelle il perd rapidement son synchronisme.

Tout accroissement de cette limite permet de réduire la durée du démarrage ou du freinage, ce qui est exigé par notre application.

Or on démontre que cette fréquence décroît avec le couple résistant et l'inertie entraînée [6].

Par conséquent, une augmentation du couple moteur entraînerait un accroissement de la fréquence limite de démarrage.

Dans ce but, nous utilisons la technique de superposition des phases qui consiste à alimenter simultanément deux phases, le couple résultant est la somme des couples partiels.

La valeur moyenne est ainsi augmentée.

La figure IV.8 montre la séquence des tensions avec superposition, ainsi que les conditions d'excitation des phases.

Ces conditions sont facilement réalisées à l'aide de décalages cycliques sur une position mémoire, désignée par EXCIT, préalablement initialisée à un état d'excitation et du chargement de 4 bits du CRU à partir de EXCIT.

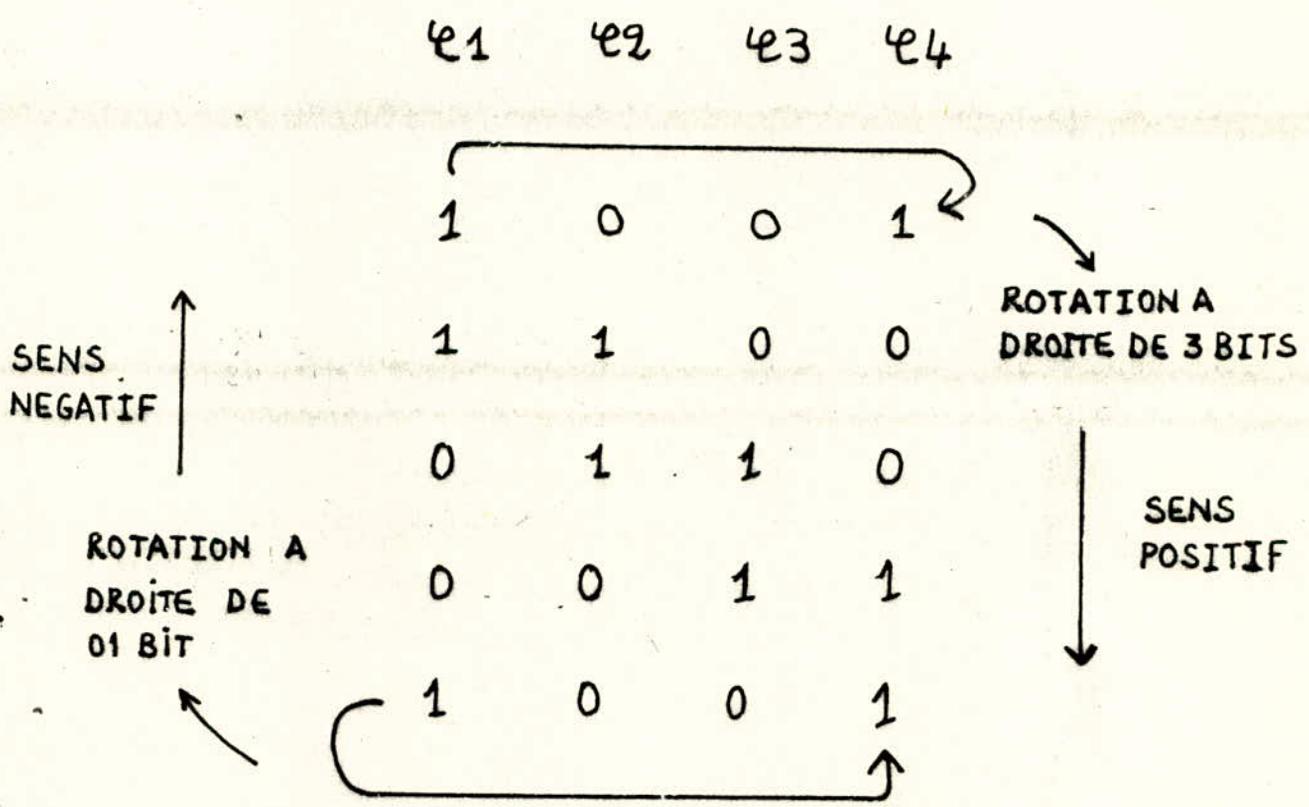
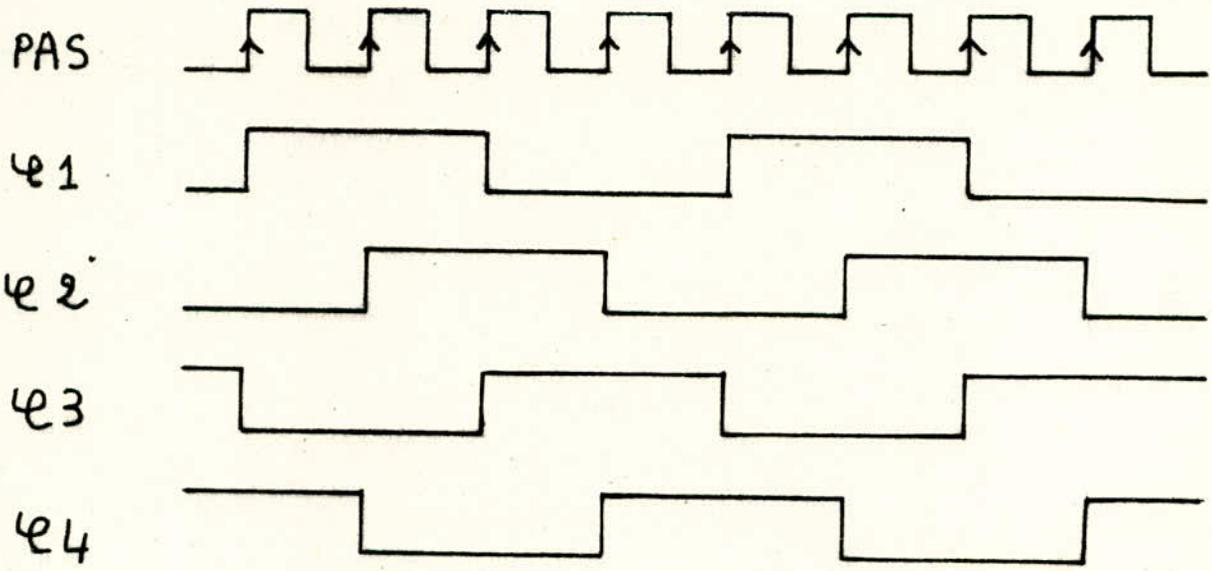


FIGURE IV.8 - COMMANDE PAR SUPERPOSITION DES PHASES.

La figure IV.9 montre l'organigramme du sous-programme "Avance d'un pas" qui exécute un pas dans un sens ou dans l'autre, maintient le nouvel état pendant un intervalle de temps donné et rend le contrôle au programme principal.

La position mémoire COMMUT contient un nombre signé, le bit  $b_{15}$  indiquant le sens de rotation :

-  $b_{15} = 0$ , sens positif

-  $b_{15} = 1$ , sens négatif

Les bits  $b_0$  à  $b_{14}$  représentent l'intervalle pendant lequel le nouvel état sera maintenu.

La vitesse du moteur est déterminée par cet intervalle.

Avant de développer les sous-programmes de commande du système, nous devons connaître les performances des moteurs que nous utilisons.

Pour cela, nous proposons deux méthodes expérimentales pour déterminer respectivement la fréquence maximale de démarrage (F.M.D.) et la fréquence maximale absolue (F.M.A.) des moteurs.

#### IV.4-2) - DETERMINATION DES FREQUENCES MAXIMALES DE DEMARRAGE ET ABSOLUE DU MOTEUR

Comme la source de commutation est le microprocesseur, pour régler la fréquence à laquelle fonctionnera le moteur, il est nécessaire de calculer le temps d'exécution de chaque instruction utilisée.

Le sous-programme "Avance d'un pas" demande pour son exécution :

$$t_{\text{PAS}} = (170 + 28/\text{COMMUT}/) \times t_c$$

où  $t_c$  est le temps de cycle du microprocesseur, et  $/\text{COMMUT}/$  l'intervalle de commutation.

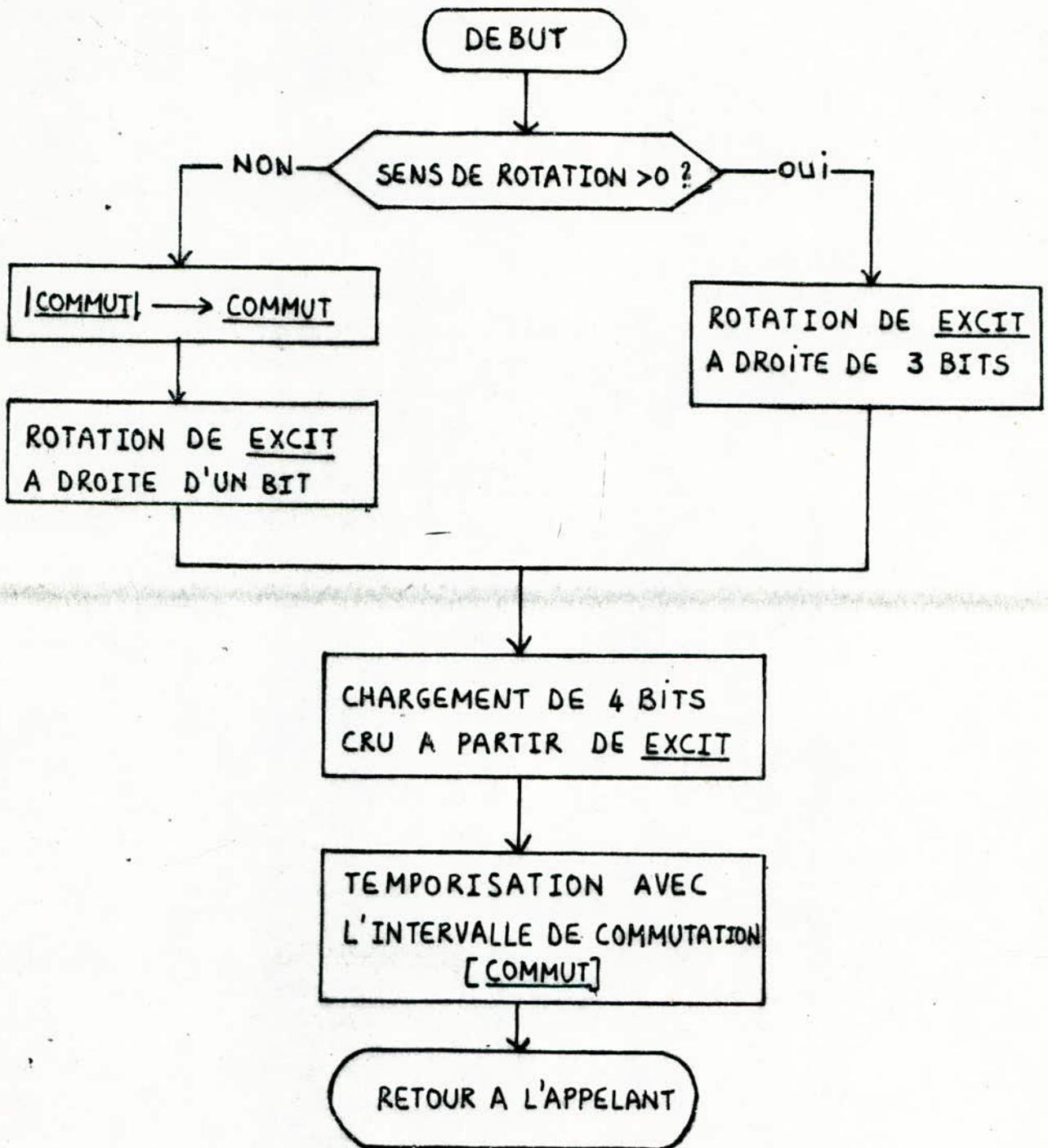


FIGURE II.9- SOUS-PROGRAMME "AVANCE D'UN PAS".

La figure IV.10 donne l'organigramme de la méthode permettant la détermination de la fréquence maximale de démarrage (F.M.D.).

Cette méthode consiste à faire tourner le moteur avec un intervalle de commutation donné. Si le moteur démarre bien, on réexécute le programme avec un intervalle plus faible de façon à avoir une fréquence plus élevée, jusqu'à ce que le moteur ne suive plus la commande.

Du fait que la boucle dans le programme prend 42 cycles, plus le temps d'exécution du sous-programme "Avance d'un pas", la F.M.D. est alors donnée par :

$$\frac{1}{FMD} = t_{PAS} + (42 \times t_c)$$

Une méthode expérimentale permettant de déterminer la fréquence maximale absolue (FMA) est explicitée par l'organigramme de la figure IV.11. Elle est obtenue en réalisant l'accélération du moteur à l'aide d'un tableau dans lequel sont chargés des intervalles de commutation décroissants.

Lorsqu'on atteint la fin du tableau, on maintient une vitesse constante ; celle-ci correspond à la fréquence maximale absolue proposée.

Si le moteur n'a pas perdu de pas, on recommence l'opération avec une F.M.A. proposée plus élevée [4].

#### IV.4-3) - SOUS-PROGRAMMES DE COMMANDE DU SYSTEME :

Les commandes que nous proposons pour le système de perçage sont les suivantes :

- MOVE ABSOLUTE: déplacement de l'outil, par rapport à l'origine, vers les coordonnées spécifiées. Cette commande sera désignée la lettre "A".

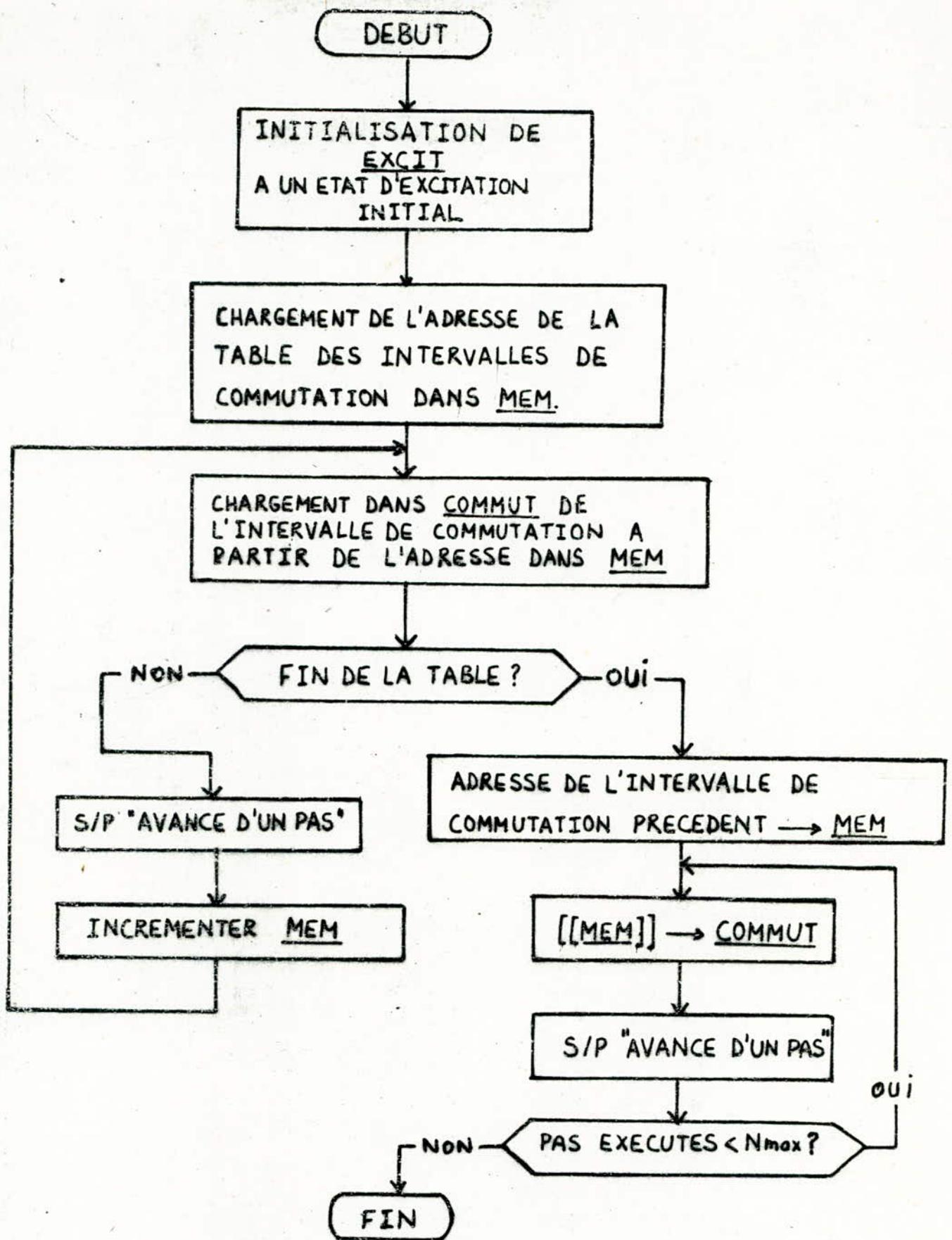


FIGURE IV.11-DETERMINATION EXPERIMENTALE DE LA FREQUENCE MAXIMALE ABSOLUE(FMA).

- DRILL : exécute l'opération de perçage à la position actuelle de l'outil. A la fin du perçage, l'outil est remis à la position levée. Cette commande sera désignée par la lettre "D".

- HOME : déplace l'outil à l'origine des coordonnées. Elle sera désignée par la lettre "H".

Toutes ces commandes sont conçues autour de la sous-routine DEPLACEMENT dont l'organigramme est donné à la figure IV.12.

Le nombre de pas à exécuter pour réaliser le déplacement est rangé dans une position mémoire désignée par NPAS.

NDECE désigne la position mémoire contenant le nombre de pas nécessaires à la décélération.

PT.ACC et PT.DECE sont des positions mémoire contenant respectivement le pointeur de la table d'accélération et celui de la table de décélération.

Dans un premier temps, le moteur est accéléré selon la table d'accélération. Dès que le nombre de pas restant  $N_r$  est inférieur ou égal au nombre de pas nécessaire à la décélération, on compare le prochain intervalle de commutation de la table d'accélération à l'intervalle de la table de décélération qui correspond à  $N_r$  pas avant la fin de cette table.

Lorsque ce dernier devient plus grand, on commence la décélération jusqu'à l'arrêt.

Le sous-programme de la commande MOVE ABSOLUTE est explicité par l'organigramme de la figure IV.13.

CRUBAS est une position mémoire contenant l'adresse des lignes d'entrée/sortie excitant l'un des moteurs.

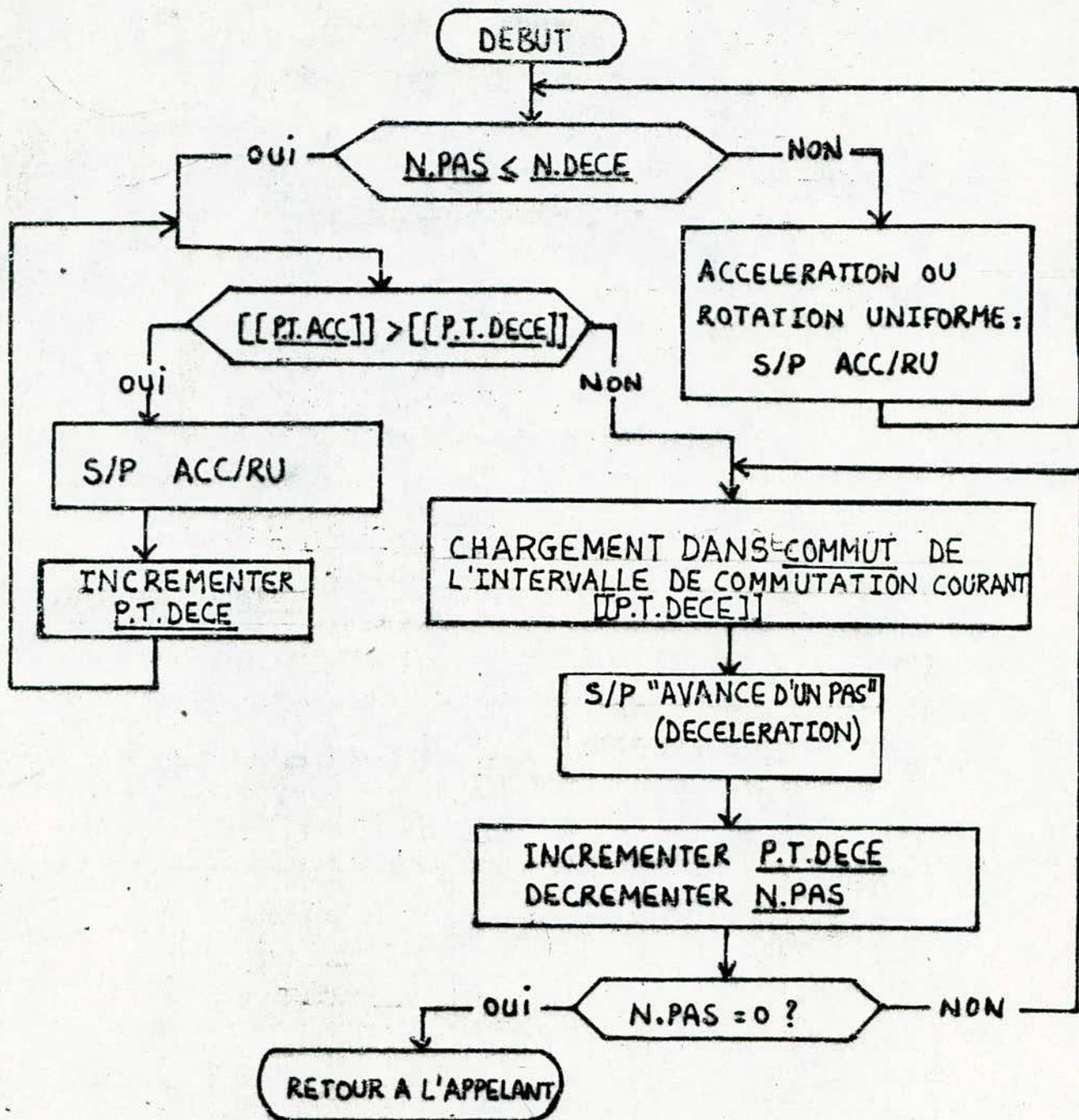


FIGURE IV.12-a- SOUBROUTINE DEPLACEMENT.

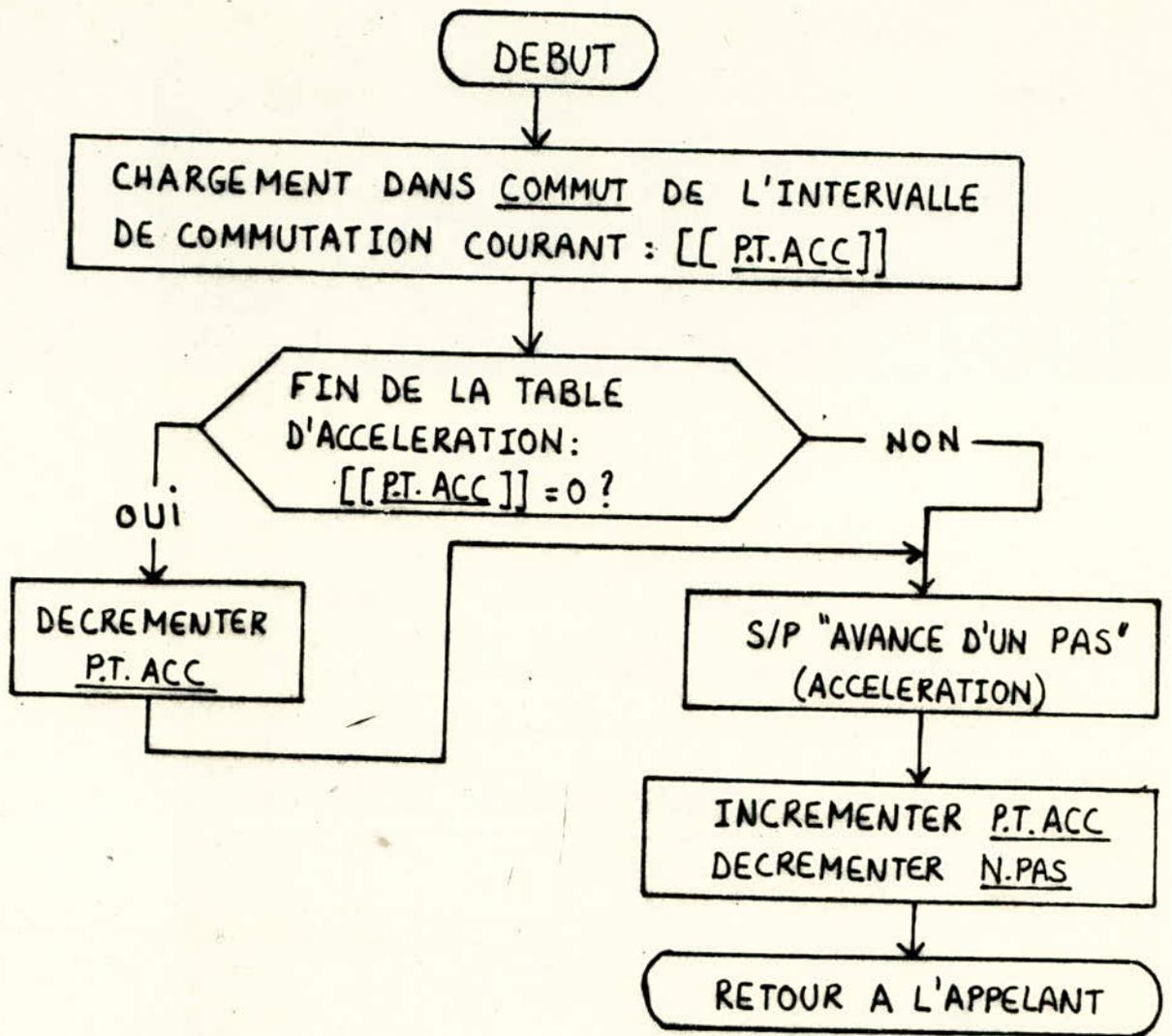


FIGURE IV.12-b- SOUS-PROGRAMME ACCELERATION/  
ROTATION UNIFORME (ACC/RU).

FIGURE IV.13- SOUS-PROGRAMME MOVE ABSOLUTE.

XOPAS et YOPAS contiennent les coordonnées de la position actuelle de l'outil.

Enfin, XRPAS et YRPAS contiennent les coordonnées, spécifiées par la commande, de la position vers laquelle se déplacera l'outil.

Le nombre de pas à exécuter par le moteur X, par exemple, est obtenu en retranchant XOPAS de XRPAS; selon le signe de la différence, le bit 15 de la position mémoire COMMUT est mis à un ou zéro afin d'obtenir une rotation dans un sens ou dans l'autre.

La commande HOME est facilement obtenue à partir du sous-programme MOVE ABSOLUTE comme le montre la figure IV.14.

La figure IV.15 montre l'organigramme de la commande DRILL.

Celle-ci excite le moteur Z de façon à effectuer un déplacement vertical constant, correspondant à la course Z.

La pénétration du stratifié par l'outil se fait à vitesse constante.

Une fois le perçage terminé, une temporisation est effectuée puis l'outil est levé.

Le programme complet du traitement par la carte de commande dépend du mode utilisé :

- mode de commande à distance via le PC:
- mode de commande locale via le clavier de la carte à microprocesseur TM 990/189.

Dans la troisième partie de ce chapitre, nous présentons le programme complet et sa liaison avec la partie du logiciel sous PC.

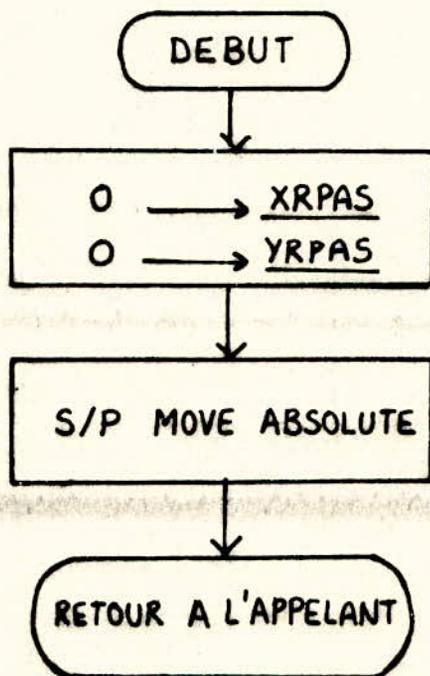


FIGURE IV.14 - SOUS-PROGRAMME HOME.

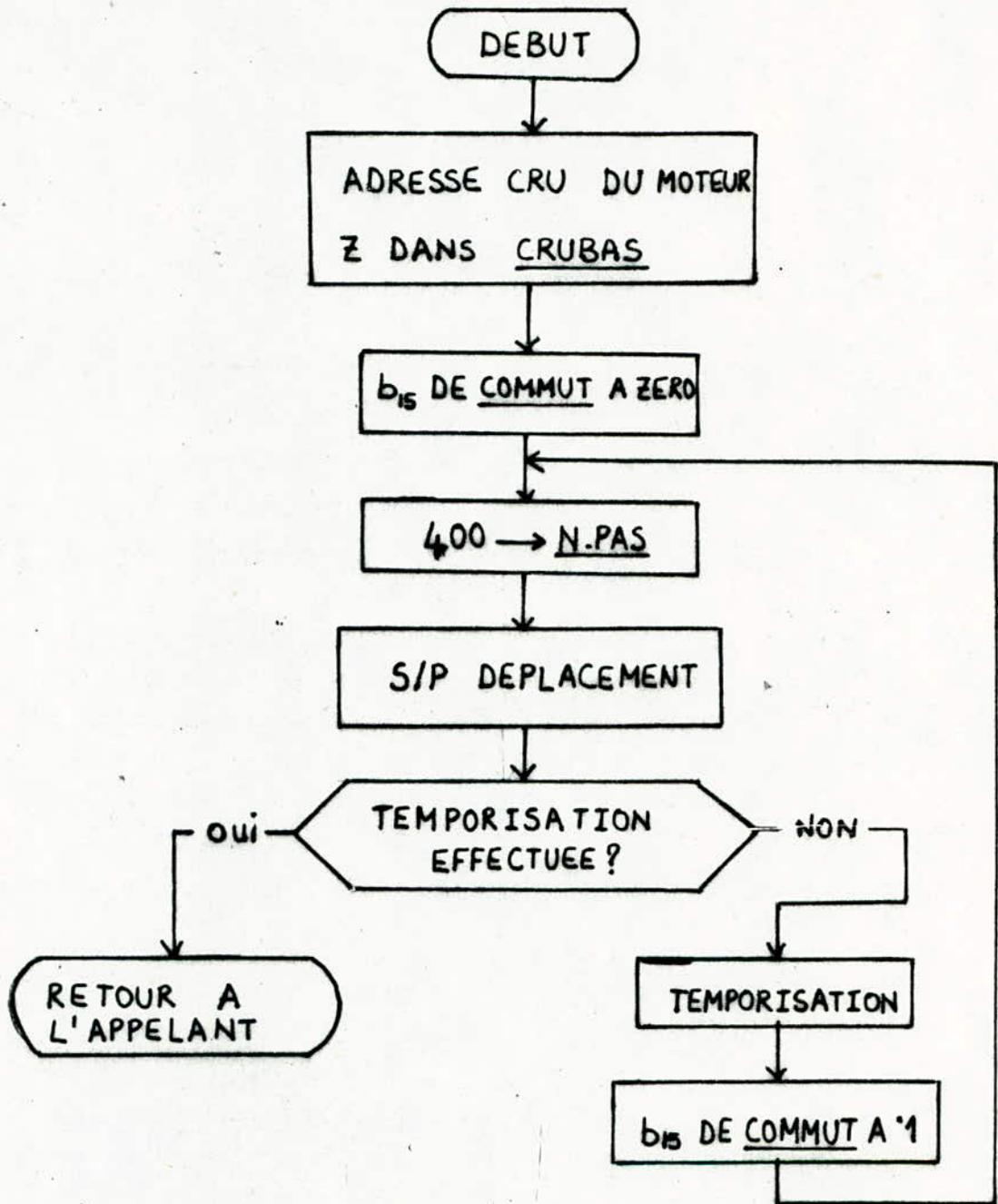


FIGURE IV.15- SOUS-PROGRAMME DRILL.

### TROISIEME PARTIE : ORGANISATION GENERALE DU LOGICIEL DE GESTION

Selon le mode de commande choisi, le codage des paramètres d'exécution diffère, et ce pour permettre une identification correcte de la commande au niveau de la carte à microprocesseur.

#### IV.5) - MODE DE COMMANDE A DISTANCE :

Dans ce cas, deux possibilités se présentent à l'utilisateur :

##### IV.5-1) - PERÇAGE A LA CHAINE DES TROUS :

Ici, le PC lit les coordonnées du trou à percer préalablement sauvegardées sur fichier par la routine de décodage.

Ces coordonnées sont délimitées par un caractère de fin de bloc, et un caractère de début de bloc.

Ce dernier est spécifique à ce mode d'exécution.

La commande ainsi codée est envoyée à la carte de commande.

Son format est le suivant :

\$ X, Y &

Une fois la commande exécutée, la carte de commande envoie le caractère > pour l'informer qu'elle est prête à recevoir les coordonnées du prochain trou.

Dès que le diamètre des trous change, le logiciel demande à l'utilisateur de changer l'outil et de valider la suite de l'opération en tapant une touche au clavier.

#### IV.5-2) - PERÇAGE INDIVIDUEL DES TROUS (CURSEUR VISEUR) :

Cette possibilité consiste à afficher à l'écran le circuit imprimé, préalablement sauvegardé par la routine sous-interruption.

Ensuite l'utilisateur vise le trou à percer grâce à un curseur, et valide l'opération de perçage en appuyant sur une touche du clavier.

Le PC récupère les coordonnées du curseur, et transmet une commande ayant le format : # X, Y &.

Une fois l'exécution terminée, la carte de commande envoie à l'écran le message "READY".

#### IV.6) - MODE DE COMMANDE LOCALE :

Ce mode permet de commander le système de perçage par appui sur des touches au clavier de la carte TM 990/189.

Dans ce cas l'utilisateur dispose de trois commandes associées à des touches spécialisées du clavier, qui sont :

- le MOVE utilisant quatre touches pour assurer un déplacement de l'outil dans les directions nord, sud, est et ouest.

- Le HOME qui utilise une touche du clavier, permettant de remettre l'outil à la position 0,0.

- Le DRILL qui utilise une autre touche du clavier pour effectuer une opération de perçage.

Les deux modes de commande du système (local ou à distance) sont associés à l'interrupteur LOAD de la carte TEXAS.

L'action sur cet interrupteur génère une interruption matérielle prioritaire, et provoque le déroulement de la routine correspondante.

Au début de cette routine, un test, sur un compteur à deux états, se fait pour déterminer le mode de commande actuel, et passer à l'autre mode.

L'utilisateur pourra ainsi changer de mode à n'importe quel moment, en actionnant l'interrupteur LOAD.

Cette possibilité est très utile au cas où un problème surviendrait au moment de l'exécution par le mode de commande à distance.

L'organigramme complet du logiciel de commande est donné à la figure IV.16.

On notera que les valeurs maximales de X et Y sont respectivement de 4000 et 2500, ce qui correspond aux courses  $X = 400$  mm et  $Y = 250$  mm.

Le nombre de pas à effectuer par un moteur est obtenu en divisant le déplacement correspondant par la résolution, qui est de 0,1 mm dans notre cas.

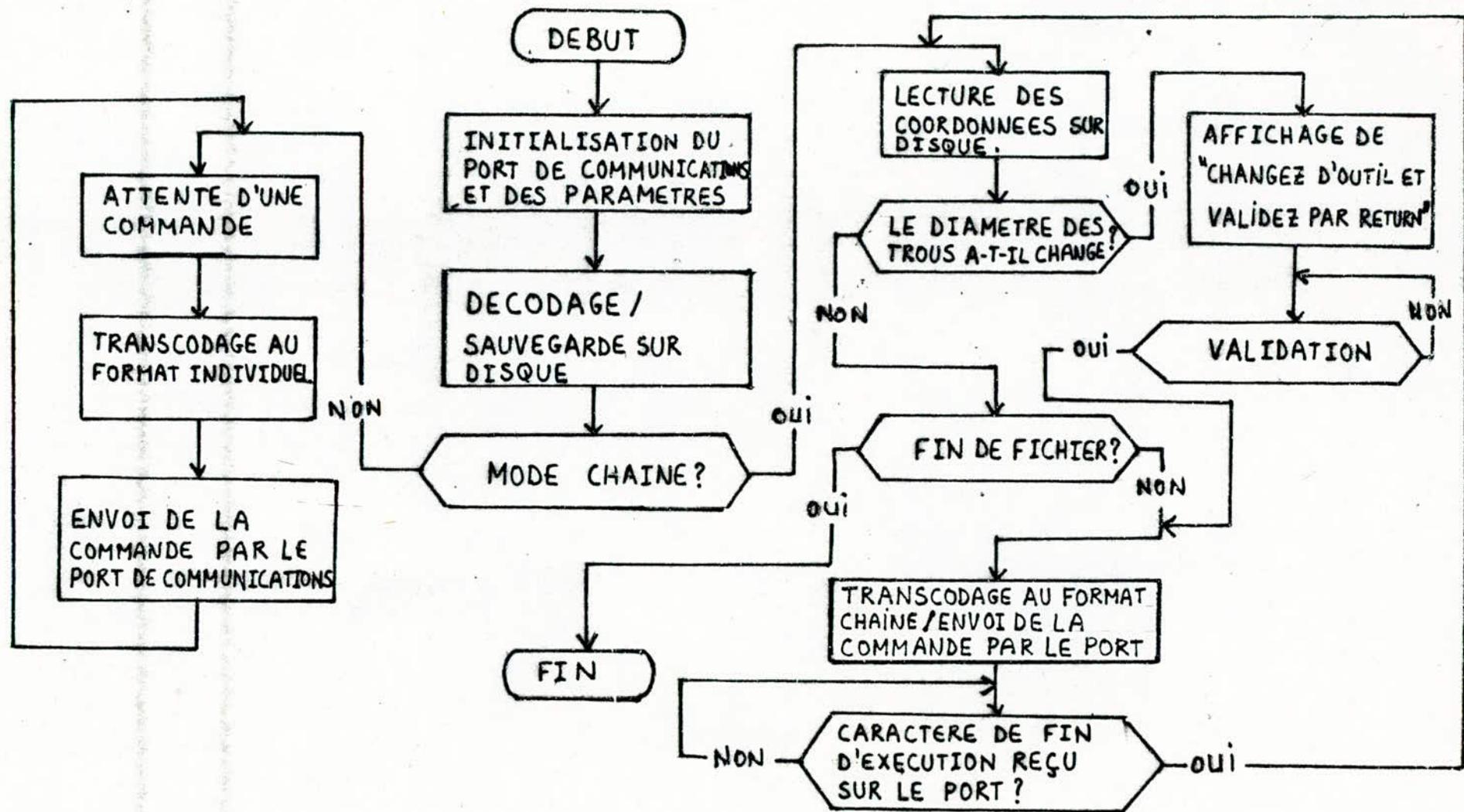


FIGURE IV.16-a - ORGANIGRAMME GENERAL DU PROGRAMME D'INTERACTION DU SYSTEME AVEC UN LOGICIEL DE C.A.O .

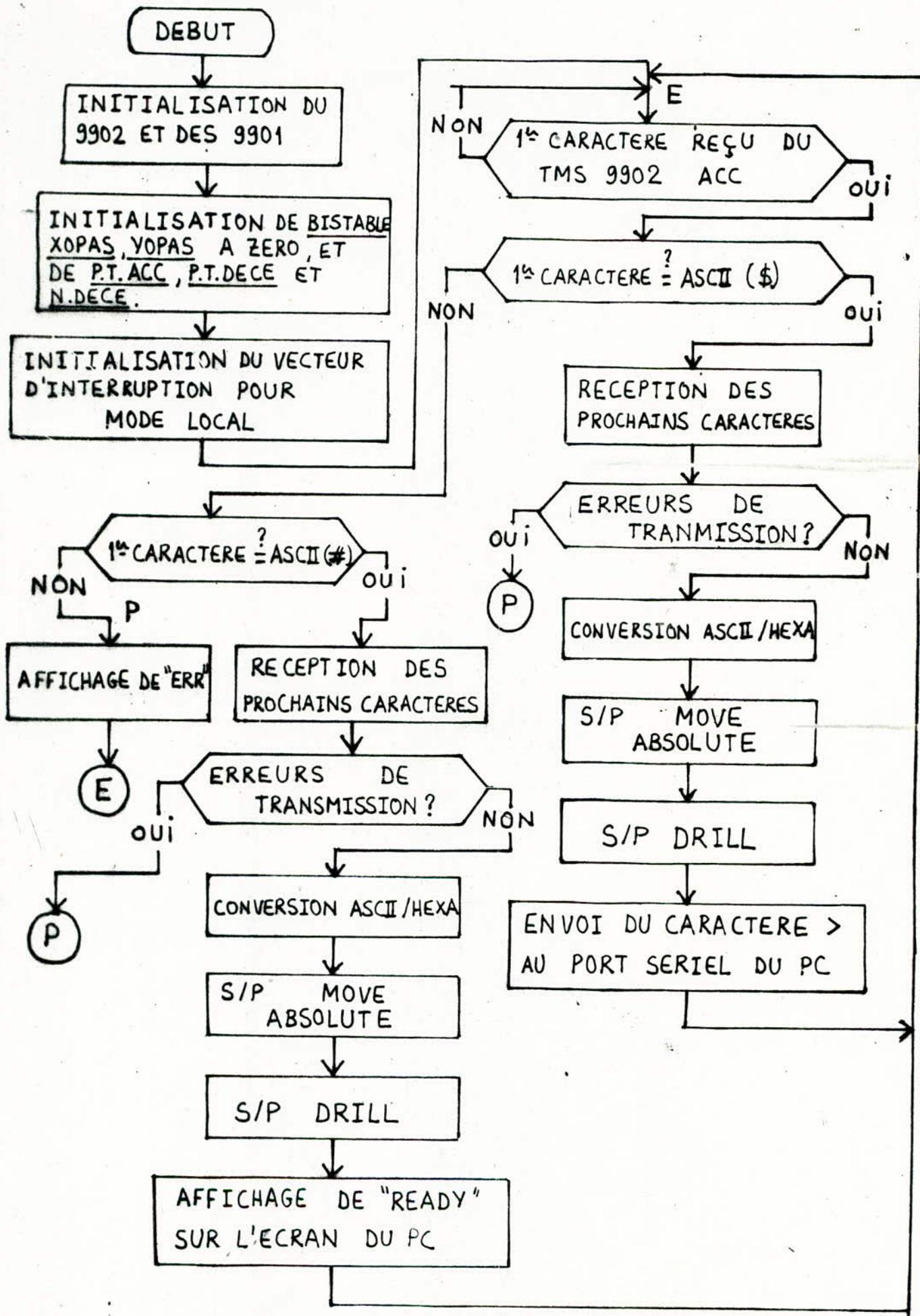


FIGURE IV.16-b- ORGANIGRAMME GENERAL SOUS CARTE DE COMMANDE.

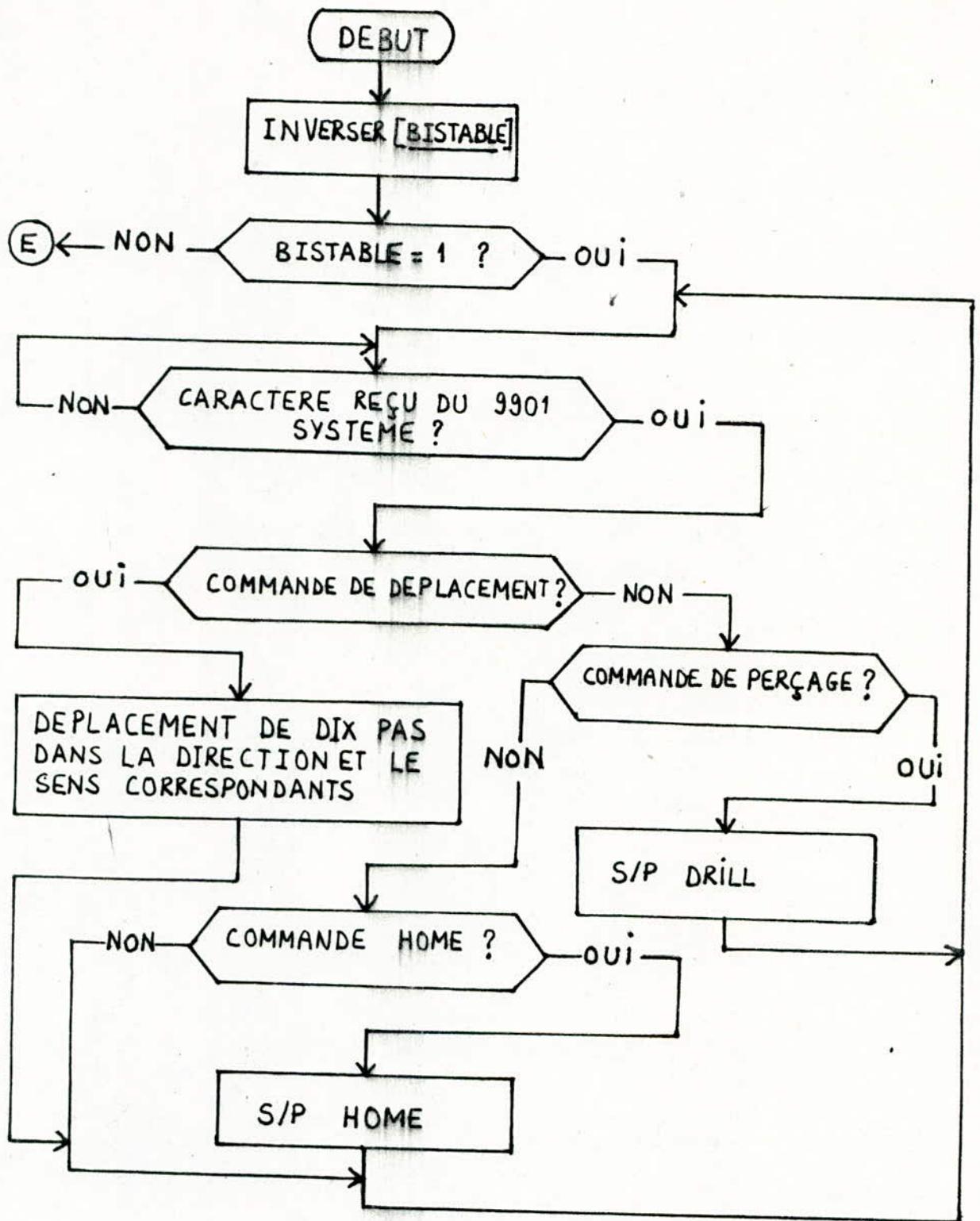


FIGURE IV.16 - C - ROUTINE D'INTERRUPTION DU VECTEUR "LOAD" POUR LE CHOIX DU MODE DE COMMANDE.

#### IV.7) - CONCLUSION :

Dans ce chapitre, nous avons d'abord présenté la carte de commande du système qui a été réalisée en connectant la carte à microprocesseur TM 990/189 à l'interface de puissance

Puis nous avons élaboré un logiciel complet permettant de piloter le système de perçage par un compatible IBM-PC.

Ce logiciel récupère les paramètres nécessaires à l'opération de perçage, à partir du schéma réalisé à l'aide d'un logiciel de conception de circuits imprimés, et effectue un traitement de ces informations afin de réaliser la fonction désirée.

# CONCLUSION

Notre contribution à la conception et à la réalisation d'un système de perçage automatique de circuits imprimés comporte une étude du système mécanique, la réalisation de la carte de puissance et l'élaboration du logiciel de gestion du système.

Nous avons commencé par l'étude du système mécanique pour lequel nous avons proposé un schéma de principe ; celui-ci illustre bien les liaisons entre les différents bâtis et la façon dont peuvent être réalisés les déplacements de l'outil.

Nous avons ensuite fait une étude des moteurs pas à pas, que nous avons retenu pour le déplacement et le positionnement de l'outil, et poursuivi par l'étude de différents schémas d'alimentation de ces moteurs.

De cette étude comparative, nous avons retenu une alimentation par hacheur qui nous a permis d'obtenir un courant en forme de créneaux dans chaque phase du moteur avec un rendement énergétique élevé.

A partir de ce schéma, nous avons réalisé la carte d'interface de puissance pour un moteur pas à pas à quatre phases.

Pour pouvoir réaliser les différentes fonctions inhérentes au perçage, l'interface de puissance est associée à la carte à microprocesseur TEXAS TM 990/189.

Le système de perçage ainsi constitué nécessite, pour son interaction avec un logiciel de C.A.O. (circuits imprimés) sur PC, l'élaboration d'un logiciel disposant de routines appropriées.

La première routine est une méthode originale pour le décodage du fichier image du circuit imprimé. Cette routine permet de récupérer les paramètres exigés par l'opération de perçage.

Les autres routines servent à déplacer l'outil sur le champ d'action et à exécuter l'opération de perçage d'un trou.

Le logiciel donne à l'utilisateur le choix entre deux modes de commande. Le mode de commande à distance via un compatible IBM-PC, ou le mode local qui permet d'utiliser directement le clavier de la carte TM 990/189.

Ce travail peut-être complété par une étude détaillée du système mécanique afin de pouvoir le réaliser.

On peut éventuellement le perfectionner en concevant un changement automatique de l'outil.

D'autre part, des améliorations pourraient être apportées au niveau de la commande. Il s'agit d'optimiser la méthode que nous avons utilisée pour les très petits déplacements de l'outil.

De plus, pour une commande plus fiable, il faudrait organiser le dialogue entre la partie puissance-moteurs et la carte à microprocesseur. Ceci peut-être fait grâce à des lignes de contrôle reliant cette partie au microprocesseur.

Le système de perçage permet effectivement d'automatiser un processus industriel exigeant une rapidité d'exécution et une précision élevée. Ceci est fait par le passage de la C.A.O. à la F.A.O.

Il peut trouver un champ d'application très vaste dans les laboratoires de maquettes et surtout à l'échelle industrielle, où l'évolution rapide des méthodes de fabrication de circuits imprimés exige des techniques de réalisation particulièrement fiables.

Ainsi des extensions du système, en terme d'application, peuvent être envisagées. Nous citerons entre autres l'automatisation de l'opération de fraisage et l'insertion automatique de composants à montage en surface.

ANNEXE

## LISTE DES COMPOSANTS

### RESISTANCES

$R_1 : 56 \text{ k}\Omega / 0,5 \text{ W}$   
 $R_2 : 2,2 \text{ k}\Omega / 0,5 \text{ W}$   
 $R_3 : 220 \text{ k}\Omega / 0,5 \text{ W}$   
 $R_4 : 10 \text{ k}\Omega / 0,5 \text{ W}$   
 $R_5 : 780 \Omega / 0,5 \text{ W}$   
 $R_6 : 2,2 \text{ k}\Omega / 0,5 \text{ W}$   
 $R_7 : 0,5 \Omega / 3 \text{ W}$   
 $R_8 : 33 \text{ k}\Omega / 0,5 \text{ W}$   
 $R_{10} : 10 \text{ k}\Omega / 0,5 \text{ W}$

### CAPACITES

$C : 22 \text{ nF}$

### DIODES

$D : \text{BY } 249 / 600$

### TRANSISTORS

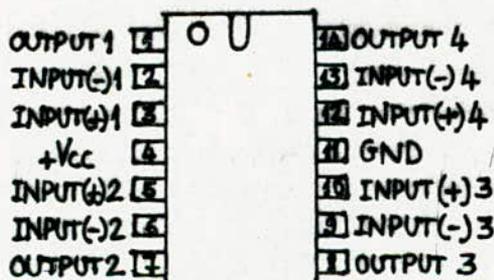
$T : \text{BDX } 33 \text{ C}$

### CIRCUITS INTEGRES

LM 224 N - QUADRUPLE AMPLIFICATEUR OPERATIONNEL

CD 4009 - BUFFER INVERSEUR

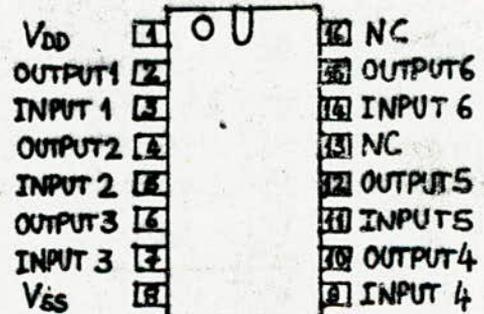
### BROCHAGES / CARACTERISTIQUES



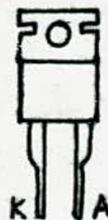
LM 224 N.



BDX 33 C  
 $V_{CEMAX} = 100 \text{ V}$   
 $I_{CMAX} = 12 \text{ A}$   
 $\beta = 750 / I_C = 3 \text{ A}$   
 $F_T = 20 \text{ MHz.}$



CD 4009



BY 249/600  
 $V_{RRM} = 600 \text{ V}$   
 $I_{FMAX} = 7 \text{ A}$   
 $t_{tr} = 35 \text{ ns.}$

# BIBLIOGRAPHIE

- 1 M.BENSEBTI et R.BISSEKRI  
"Développement d'un nouveau système d'E/S graphique pour micro-ordinateurs"  
PROJET DE FIN D'ETUDES, E.N.P., JUIN 1987.
- 2 LENORMAND et TINEL  
"Memento de dessin industriel"  
FOUCHER, PARIS, 1979.
- 3 MICHEL GIRARD  
"Electronique industrielle"  
MC GRAW-HILL, PARIS, 1987.
- 4 HIROYUKI MIYAMOTO  
"Modélisation et commande optimale d'un moteur pas à pas par microprocesseur"  
THESE DE DOCTEUR-INGENIEUR, I.N.P. LORRAINE, NANCY, 1979.
- 5 R.DELSOL  
"Circuits intégrés et techniques numériques"  
CEPADUES EDITIONS, TOULOUSE, 1978.
- 6 M.JUFER  
"ELECTRO-MECANIQUE"  
TRAITE D'ELECTRICITE, D'ELECTRONIQUE ET D'ELECTROTECHNIQUE, DUNOD, PARIS, 1986.
- 7 A.DE LUSTRAC  
"Carte d'alimentation en courant pour moteur pas à pas"  
ELECTRONIQUE APPLICATIONS N°52, PARIS, MARS, 1987.
- 8 Y.LE PRIOUX ET S.ELGOLLI  
"Interface de puissance "microprocesseur" pour commande de hacheurs"  
ELECTRONIQUE APPLICATIONS N°49, PARIS, AOUT/SEPT, 1986.

- 9 TEXAS INSTRUMENTS, 9900 family systems design and data book - Microprocessors séries  
AUSTIN, TEXAS, 1977.
- 10 H.BEGORRE  
"La carte TM 990/189 UNIVERSITY"  
INP. LORRAINE, NANCY, 1977.
- 11 RUSSELL RECTOR  
"The 8086 book"  
OSBORNE/MC GRAW-HILL, BERKELEY, CALIFORNIA, 1984.
- 12 PASCAL HAUSMANN  
"Manuel technique de l'AMSTRAD PC 1512"  
MICRO-APPLICATIONS, PARIS, 1986.
- 13 B.MARCHALL  
"Methodologie de création d'une routine résidente sous MS-DOS"  
MICRO-SYSTEMES N°76, PARIS, JUIN 1987.
- 14 RICHARD ALLEN KING  
"The MS-DOS handbook"  
SYBEX/BPB PUBLICATIONS, NEW-DELHI, 1987.
- 15 MANUEL DU TURBO-PASCAL, version 3.  
BORLAND INTERNATIONAL, PARIS