UNIVERSITE DE MONTREAL

SYSTEME DE REGLAGE DE MACHINE ASYNCHRONE MONOPHASEE

Par

Richard D'ULISSE
DEPARTEMENT DE GENIE ELECTRIQUE
ECOLE POLYTECHNIQUE

MEMOIRE PRESENTE EN VUE DE L'OBTENTION
DU GRADE DE MAITRE ES SCIENCES APPLIQUEES (M.Sc.A)
Avril 1989

c Richard D'Ulisse 1989

National Library of Canada
Bibliothèque nationale du Canada

Canadian Theses Service
Service des thèses canadiennes

Ottawa, Canada
K1A 0N4

The author has granted an irrevocable non-exclusive licence allowing the National Library of Canada to reproduce, loan, distribute or sell copies of his/her thesis by any means and in any form or format, making this thesis available to interested persons.

L'auteur a accordé une licence irrévocable et non exclusive permettant à la Bibliothèque nationale du Canada de reproduire, prêter, distribuer ou vendre des copies de sa thèse de quelque manière et sous quelque forme que ce soit pour mettre des exemplaires de cette thèse à la disposition des personnes intéressées.

The author retains ownership of the copyright in his/her thesis. Neither the thesis nor substantial extracts from it may be printed or otherwise reproduced without his/her permission.

L'auteur conserve la propriété du droit d'auteur qui protège sa thèse. Ni la thèse ni des extraits substantiels de celle-ci ne doivent être imprimés ou autrement reproduits sans son autorisation.

ISBN 0-315-58175-1

Canada

UNIVERSITE DE MONTREAL

ECOLE POLYTECHNIQUE

Ce mémoire intitulé:

SYSTEME DE REGLAGE DE MACHINE

ASYNCHRONE MONOPHASEE

présenté par: .....Richard D'ULISSE

en vue de l'obtention du grade de: .....M.Sc.A.

a été dûment accepté par le jury d'examen constitué de:

M. ROLAND MALHAME.....,

M. GUY OLIVIER.....,

M. GILLES ROY.....,

SOMMAIRE

Ce projet s'insère dans le cadre d'un programme de recherche visant à améliorer les performances de la machine asynchrone monophasée. L'objectif principal est le développement d'un système de réglage de machine asynchrone utilisant un convertisseur direct composé d'interrupteurs à mosfets à hautes performances.

Le système vise l'amélioration à faible coût des performances en régime permanent de la machine asynchrone, comparativement à une optimisation à coût d'investissement élevé. Il est composé d'un système à microprocesseurs commandant les interrupteurs du convertisseur et réalisant l'asservissement de l'entraînement. La flexibilité du système de commande permet son utilisation comme banc d'essais pour les stratégies d'opérations des interrupteurs ainsi que pour les asservissements. Ces derniers n'utilisent que les courants et les tensions de la machine comme mesures d'état; aucun capteur de position, de vitesse ou de couple n'est utilisé. Le travail consiste principalement à trouver des façons de commander la machine en régime permanent et de réaliser l'entraînement en fonctions des contraintes énoncées ci-haut. Les points majeurs de cette dernière sont:

- l'estimation numérique du couple et une mesure analogique de la déviation relative du point d'opération de la machine par rapport à son point d'opération amélioré;
- le développement d'un algorithme d'évolution d'optimisation minimisant une fonction de coût donnée;
- la réalisation d'un entraînement généralisé.

L'algorithme d'évolution est bien adapté aux procédés peu ou mals définis; il peut être utilisé là où le développement d'un modèle du système est trop coûteux ou trop difficile, ou encore que son utilisation "on-line" demande un système de commande très performant et très dispendieux.

L'entraînement généralisé constitue un outil de développement pour:

- les topologies de convertisseurs;
- les stratégies d'opération des interrupteurs d'un convertisseur;
- l'étude de critère d'optimisation;
- le test de boucles d'asservissements.

Cette étude met en évidence la possibilité d'utiliser un estimateur de couple en boucle ouverte et des tables de point d'opération optimal en fonction du couple (obtenues par l'évolution d'optimisation ou par calculs "off-line") afin de régler rapidement le point d'opération du système.

ABSTRACT

This project is taking place in research program on single phase induction machine. The main goal of this study is to improve the steady state performances of single phase induction machines with the use of a fixed frequency drive.

The system is aiming a good performance improvement at low cost rather than a real optimization at a higher cost. The drive is composed by a microprocessor system controlling a single phase AC converter feeding the machine. The microprocessor system uses only the currents and voltages measured at the windings to realize the control of the drive. Some of the mosts importants topics touched by that study are:

- steady state numerical torque estimation and analog mesurement of the deviation from the optimal operating point;
- development of an algorithm for steady state evolutionnary optimization of an objective function (or minimization of a cost function);
- realization of a generalized drive for a single phase motor.

The optimization algorithm is well suited for badly or none defined system. It can be used where the cost for developing a model is high or the model difficult to do.

The drive is a good development tool for:

- cycloconverter commutation strategy;
- control law development;
- testing of new feedback parameters.

This study also demonstrates the possibility of using torque estimation, with tables of optimized operating points, to rapidly set the "best" operating point.

REMERCIEMENTS

J'aimerais remercier toutes les personnes qui ont participé à la bonne marche de ce projet.

En premier lieu Gilles Roy, mon directeur, pour son assistance technique, sa compréhension et son ouverture d'esprit.

Ensuite, le CRSNG pour son aide financière (bourse d'étude).

Je suis aussi très reconnaissant à Gaétan Décarie pour les figures du rapport: de croquis ils sont devenus chefs d'oeuvre grâce à lui. S'il passait autant de temps a améliorer son coup droit au badminton, il serait redoutable.

Je m'en voudrais de ne pas souligner l'importance du support de mes parents et amis que j'ai négligés trop souvent ces derniers temps. Je remercie aussi mes amis et collègues de laboratoire de leurs attentions et soutien.

Je ne peux terminer cette longue énumération sans penser à ma compagne, Chantal, qui a accepté de me partager avec le laboratoire. Je lui dédie ce travail.

TABLE DES MATIERES

Sommaire iv
Abstract vi
Remerciements vii
Liste des figures xi
Liste des tableaux xiv
Liste des symboles xv
Liste des annexes xviii
INTRODUCTION 1
CHAPITRE I 4
LE CONVERTISSEUR ET SA COMMANDE
1.1 Les contraintes imposées au convertisseur par moteur asynchrone 7
1.2 Définition du problème 16
1.3 Le système à microprocesseur 22
1.3.1 Design du logiciel 23
1.3.2 Le diagramme hiérarchique 28
1.3.3 Le matériel 30
1.3.3.1 Le conditionnement des signaux. 30
1.3.3.2 Interfaçage des entrées et sorties 34
1.3.3.3 Le module microprocesseur au sens large 36
1.3.4 Les procédures du logiciel 36
1.3.4.1 L'exécution des événements et la synchronisation 38
1.3.4.1.1 Les horloges 39
1.3.4.1.1.1 PLL numérique 40
1.3.4.1.1.2 L'horloge du cycle déphasé 42
1.3.4.1.1.3 La précision du réglage . 45

1.3.4.1.1.4 Les problèmes reliés à la discontinuité dans les horloges 46
1.3.4.1.2 L'exécution des événements . 48
1.3.4.2 La progression vers la consigne 55
1.3.4.3 Le programme de communication . 59
1.3.4.4 La commande du convertisseur . 62
1.3.5 Le système en pratique 62
CHAPITRE II 65
L'ASSERVISSEMENT DE L'ENTRAINEMENT
2.1 La commande du moteur asynchrone 65
2.1.1 La régulation d'un paramètre représentatif du point d'opération . . . 75
2.1.1.1 Le facteur de déplacement au moteur 80
2.1.1.2 Un autre paramètre obtenu à l'aide de la tension induite à l'enroulement de démarrage . . 84
2.1.2 L'évolution d'optimisation en régime permanent 95
2.1.2.1 La stratégie d'évolution . . . 99
2.1.3 Utilisation d'un estimateur en boucle ouverte 106
2.2 Les stratégies d'opération des interrupteurs . 108
2.2.1 La simulation 109
2.2.2 Les critères de sélection 114
CHAPITRE III 119
DESCRIPTION DE L'ENTRAINEMENT
3.1 La définition du problème 120
3.2 Description du système à microprocesseur . . . 123
3.2.1 Le logiciel 123
3.2.2 Le matériel 126

3.2.2.1 Le module microprocesseur au sens large 128
3.2.2.2 Le module d'interfaçage des signaux 128
3.2.2.3 L'interfaçage des entrées et sorties 132
3.2.3 Les modules logiciels 136
3.2.3.1 La mesure des signaux 136
3.2.3.2 Le traitement des valeurs mesurées 139
3.2.3.3 L'asservissement 143
3.2.3.3.1 Régulation d'un paramètre . 144
3.2.3.3.2 Evolution d'optimisation . . 149
3.2.3.3.3 Fonctionnement en boucle ouverte à l'aide de l'estimateur de couple et de tables 152
3.2.3.4 L'interface avec l'utilisateur. 153
3.2.3.5 La commande de l'entraînement . 155
3.2.4 Quelques détails sur le matériel 157
CONCLUSION 165
BIBLIOGRAPHIE 167
ANNEXES 170

LISTE DES FIGURES

1.0-1 Composition du convertisseur 5
1.0-2 Topologie du convertisseur 8
1.1-1 Circuit équivalent dérivé du modèle DQ 9
1.1-2 Circuit équivalent simplifié de la machine alimentée par un convertisseur simple 11
1.1-3 Variation de 80V à 110V en un cycle 12
1.1-4 Progression de 1V à par cycle 12
1.1-5 Variation 80V à 110V par incrément de 7,5V par cycle 14
1.2-1 Schéma de principe de l'entraînement 18
1.2-2 Le microordinateur de commande du convertisseur 19
1.3.1-1 Les deux volets du logiciel 25
1.3.1-2 Le diagramme de flux des données du logiciel de commande du convertisseur 26
1.3.2-1 Diagramme hiérarchique du logiciel de commande du convertisseur 29
1.3.3-1 Les modules matériels du système de commande du convertisseur 31
1.3.3.1-1 Module de conditionnement des signaux d'entrée 33
1.3.3.1-2 Commande des interrupteurs 35
1.3.3.2-1 Le module d'interface du microprocesseur . . 37
1.3.4.1.1.1-1 Le modèle de la P.L.L. numérique 41
1.3.4.1.1.2-1 Réglage de l'horloge déphasée 43
1.3.4.1.1.4-1 Discontinuités des horloges de cycle . . 47
1.3.4.1.2-1 Les tables et le format de la commande . . 50
1.3.4.1.2-2 Les quatre zones du convertisseur 52
1.3.4.2-1 Boucle de réglage de la tension de sortie du convertisseur 56

1.3.5-1 Organisation physique des éléments de l'entraînement 63
2.1-1 Rendement et facteur de puissance du moteur pour les cas de:
1)tension nominale;
2)tension ajustée pour maximiser le paramètre 67
2.1-2 Effets d'une diminution de tension statorique sur les courants 68
2.1-3 Couple en fonction de la vitesse pour différentes tensions statoriques 70
2.1-4 Système réglé recherché 73
2.1.1-1 Le système multivariable équivalent de l'entraînement 76
2.1.1.1-1 Courbes du déphasage et du couple en fonction de la vitesse 82
2.1.1.2-1 Modèle équivalent de la machine asynchrone . 85
2.1.1.2-2 Modèle après démarrage 87
2.1.1.2-3 Simplifications en régime permanent 88
2.1.1.2-4 Montage du circuit d'estimation du couple . 91
2.1.1.2-5 Graphique de $V_{circuit}$ en fonction du couple pour différentes tensions 92
2.1.1.2-6 Graphique de $V_{circuit}$ en fonction de la vitesse pour différentes tensions 92
2.1.1.2-7 Estimation du couple
a)Graphique donnant la compensation de couple en fonction de la tension
b)Estimation à l'aide des tables de compensation et de couple de référence 94
2.1.2-1 Le principe et la procédure de l'évolution d'optimisation 98
2.1.2-2 Indice de performance du système 100

2.1.2.1-1 Trajectoire d'évolution 104
2.1.2.1-2 Mesure du gradient au point "p" 106
2.2-1 Formes d'ondes du courant et de la tension au moteur à 119V, 90V et 60V pour les stratégies symétrique et à plusieurs commutations 110
2.2-2 Formes d'ondes du courant et de la tension au moteur à 119V, 90V et 60V pour la stratégie asymétrique 111
3.2.1-1 Diagramme hiérarchique du logiciel de commande de l'entraînement 127
3.2.2.2-1 Conditionnement des signaux d'entrée 129
3.2.2.3-1 L'interfaçage des entrées et des sorties . . 133
3.2.3.1-1 La mesure des signaux et "RCOMES" 138
3.2.3.2-1 Les facteurs de conversion et "RCOMFL" . . . 141
3.2.3.2-2 Le filtrage et "RCOMFL" 142
3.2.3.3.1-1 Le régulateur 145
3.2.3.3.1-2 La boucle de régulation
a)Eléments à considérer
b)En pratique 147
3.2.3.4-1 Affichage et RCAFFI 156
3.2.4-1 Amplificateur à gain programmable 158
3.2.4-2 "Filtre-AGC-DPZ" numérique 159
3.2.4-3 "Filtre-AGC-DPZ" analogique 162
3.2.4-4 Mesure du déphasage 164

LISTE DES TABLEAUX

2.2.2-1 Evaluation qualitative des trois stratégies . 116
2.2.2-2 Analyse fréquentielle à 90V pour les trois stratégies 117

LISTE DES SYMBOLES

A: une des matrices d'un modèle d'état ou une des zones d'opération du convertisseur, selon le contexte.

AGC: amplificateur à gain auto-asservi.

B: une des matrices d'un modèle d'état ou une des zones d'opération du convertisseur, selon le contexte.

C: une des matrices d'un modèle d'état ou une des zones d'opération du convertisseur, selon le contexte.

CA/N: convertisseur analogique-numérique.

D: une des matrices d'un modèle d'état ou une des zones d'opération du convertisseur, selon le contexte.

D_i: composante du vecteur de norme unitaire, selon la variable de commande "i".

D.P.Z. ou DPZ: détecteur de passage par zéro.

E: tension.

E_m: tension au moteur.

F^p: indice de performance au point "p".

G^p: gradient au point "p".

G_i^p: composante "i" du gradient au point "p".

G_{ij}: composante "ij" de la matrice des gains de couplage

G_{Dij}: composante "ij" de la matrice de gains d'un pré-découpleur.

I: matrice identité ou courant, selon le contexte.

i_d: courant dans l'enroulement direct.

i_d^s: courant dans l'enroulement direct vu du stator.

I_m: courant au moteur.

I_{pk}: valeur maximale du courant.

i_q: courant dans l'enroulement en quadrature.

i_q^s: courant dans l'enroulement en quadrature vu du stator.

I_s: courant à la source.

K_i^p: facteur d'échelle, dans la direction de la variable de commande "i", au point "p".

Knom: constante de proportionnalité entre le couple et le glissement à tension nominale.

L_m: inductance équivalente de magnétisation.

L_r: inductance au rotor du moteur.

L_s: inductance au stator du moteur.

L_{th}: inductance équivalente vue du stator du moteur.

p: un point sur la surface de réponse.

P_{in}: puissance d'entrée au moteur.

P.L.L. (ou PLL): boucle à verroillage de phase.

P_m: indice de performance du moteur.

P_s: indice de performance du système vu de la source.

RMS: valeur efficace.

R.P.M. (ou t.p.m): unité de vitesse rotations ou tours par minute.

R_r: résistance au rotor du moteur.

R_s: résistance au stator du moteur.

R_{th}: résistance équivalente vue du stator du moteur.

S: glissement.

t: le temps.

Temg: couple électromagnétique.

$TETA1_m$: déplacement au moteur.

$TETA1_s$: déplacement à la source.

$T_{friction}$: couple dû à la friction.

U: entrée du système.

V_1: tension de référence du convertisseur.

V_2: tension à 180 degrés de V_1.

$V_{circuit}$: Tension de sortie du circuit de mesure servant à l'estimation du couple.

VCO: oscillateur commandé en tension.

V_d: tension à l'enroulement direct.

VIA: "Versatile interface adaptor"; circuit intégré 6522.

V^s_{dj}: tension à l'enroulement direct indiqué par "j". Défini dans un système stationnaire par rapport au stator.

V_q: tension à l'enroulement en quadrature.

V^s_{qj}: tension à l'enroulement en quadrature indiqué par "j". Défini dans un système stationnaire par rapport au stator .

W_r (ou ω): vitesse angulaire (du rotor) du moteur.

X^p_i: variable de commande "i" au point "p".

X_i: variable d'état du modéle du moteur.

Φ: matrice de transition d'état.

ψ^s_{ij}: flux, dans un système stationnaire par rapport au stator, dû à l'enroulement i (peut être "d" ou "q") au rotor ou au stator selon l'indice j ("r" ou "s" respectivement).

LISTE DES ANNEXES

ANNEXE A: Schéma simplifié d'un microordinateur SYM . . 170
ANNEXE B: Le modèle DQ du moteur 172
ANNEXE C: Programme de simulation de la chaîne
convertisseur - moteur 176

INTRODUCTION

Le moteur asynchrone monophasé est très répandu. Il vaut donc la peine de s'attarder à améliorer ses performances qui laissent à désirer dans certaines plages d'opération.

On distingue deux grandes méthodes pour le faire: l'ajustement du point d'opération par la fréquence et la tension (commande à fréquence variable),et par la variation de la tension seulement (commande à fréquence fixe).

Nous nous intéressons à la deuxième méthode à cause de sa relative simplicité et son probablement plus bas coût. On y agit sur la fondamentale de tension afin d'augmenter le rendement de la machine, et sur la forme de la tension pour améliorer les facteurs de puissances et de déplacement vus de la source.

L'Ecole poursuit depuis quelques années déjà un programme de recherche dans ce domaine dont le but est d'utiliser la commutation forçée plutôt que la commutation naturelle. Ce programme à déjà permis la réalisation d'un variateur à mosfets très performant [1]. Le présent projet s'attardera surtout à la partie asservissement de la chaîne variateur-moteur afin de permettre l'amélioration des performances (rendement, facteur de puissance, etc.) du moteur asynchrone, et celles vues de la source. De façon plus précise, on y poursuivra les buts suivants:

1) développer un algorithme polyvalent de commande de convertisseur en temps réel;

2) déterminer une façon d'asservir la machine, dans le but d'améliorer ses performances, en n'utilisant que les paramètres électriques découlant de la mesure des seuls courants et tensions à la machine;

3) développer un système à microprocesseur réalisant un entraînement polyvalent de machine asynchrone pour améliorer les performances en régime permanent établi sous diverses conditions d'opération.

Le premier point consiste en une modification en profondeur visant à généraliser un logiciel de commande utilisé dans [1]. Le logiciel conserve, au point de vu de l'exécution des événements (commutations) en temps réel, la même philosophie. On a reprogrammé le tout de façon structurée en y apportant les modifications nécessaires pour répondre aux nouvelles spécifications: déphaser la tension de sortie, régler la tension de sortie du convertisseur en boucle fermée et utiliser le convertisseur commandé par microprocesseur comme élément d'un entraînement multiprocesseur.

Au deuxième point, on recherche soit une mesure de l'état de charge de la machine pour servir de feedback à l'asservissement, soit une méthode de réglage ne nécessitant pas ce genre de mesure, pour optimiser les performances de la machine en régime permanent. Les modèles de la machine (DQ, et en régime permanent) y sont utilisés. Au troisième point, on réalise un système à microprocesseur pour commander l'entraînement. Le logiciel est flexible et général aux dépens d'une programmation optimisée, dédiée à un cas particulier; cela en fait un outil de développement. Il permet la réalisation des différents asservissements retenus par l'étude au deuxième point.

Le travail est divisé en trois chapitres, chacun traitant un des points énoncés ci-haut. Au chapitre 1, on décrit le convertisseur et sa commande. C'est l'élément de base de l'entraînement. La majeure partie des éléments de puissance n'a pas changée; sa conception est une partie du travail réalisé par Beauregard [1]; on ne considèrera donc

que le système de commande du convertisseur. Le chapitre constitue une description du logiciel et du matériel du système à microprocesseur commandant le convertisseur. On y donne de nombreuses précisions sur le pourquoi et le comment des procédures et sur le matériel utilisé.

Le chapitre 2 est une étude des asservissements possibles pour l'entraînement. On y remarque deux parties: les asservissements possibles et une discussion sur les stratégies de commutations des interrupteurs. On regroupe donc la façon de déterminer les variables de commande (consignes de tension et de déphasage) et la façon de les réaliser (stratégie de commutation employée par le convertisseur). Dans les asservissements possibles, on notera la description d'une méthode itérative d'optimisation des performances, ainsi que l'introduction d'une mesure mettant à profit l'enroulement de démarrage pour l'estimation du couple en régime permanent. Cet estimateur permet une utilisation directe de résultats tabulés pour optimiser les performances.

Et finalement, le chapitre 3 décrit l'entraînement réalisant les asservissements vus au chapitre 2 en se servant du convertisseur vu au chapitre 1. Il s'agit d'un outil de développement: l'utilisateur a plusieurs degrés de liberté pour programmer l'entraînement. En fait, il s'agit d'un entraînement dont on peut choisir, et changer, la structure en tout temps. L'ensemble du logiciel et du matériel utilisés y est décrit.

CHAPITRE I

Le convertisseur et sa commande.

On décrit le convertisseur utilisé pour alimenter le moteur asynchrone; c'est l'élément de base de l'entraînement. Une définition des besoins et contraintes du convertisseur et une présentation détaillée du système réalisé font l'objet du chapitre. La connaissance du convertisseur, et de ses possibilités permettent une meilleure compréhension du genre d'asservissement qu'on peut utiliser pour l'entraînement.

Le convertisseur recherché nous permet de générer une tension alternative d'amplitude et de phase commandées par rapport à sa tension d'entrée, la fréquence restant la même. On veut donc un convertisseur monophasé CA-CA à fréquence fixe (celle de la tension d'entrée). De plus, on voudrait qu'il soit direct: on ne veut pas d'un étage de redressement suivi d'un étage d'ondulation. Ce convertisseur est commandé par microprocesseur afin de permettre une grande flexibilité au niveau des différentes stratégies d'opération des interrupteurs du convertisseur. Cette flexibilité se heurte à la complexité possible des stratégies: seules des stratégies à faibles nombres de commutations dans une période de la tension de sortie sont envisagées. Précisons que le microordinateur de commande lui aussi doit être modeste.

Comme c'est illustré à la figure 1.0-1, le convertisseur est composé d'un système de commande à microprocesseur et

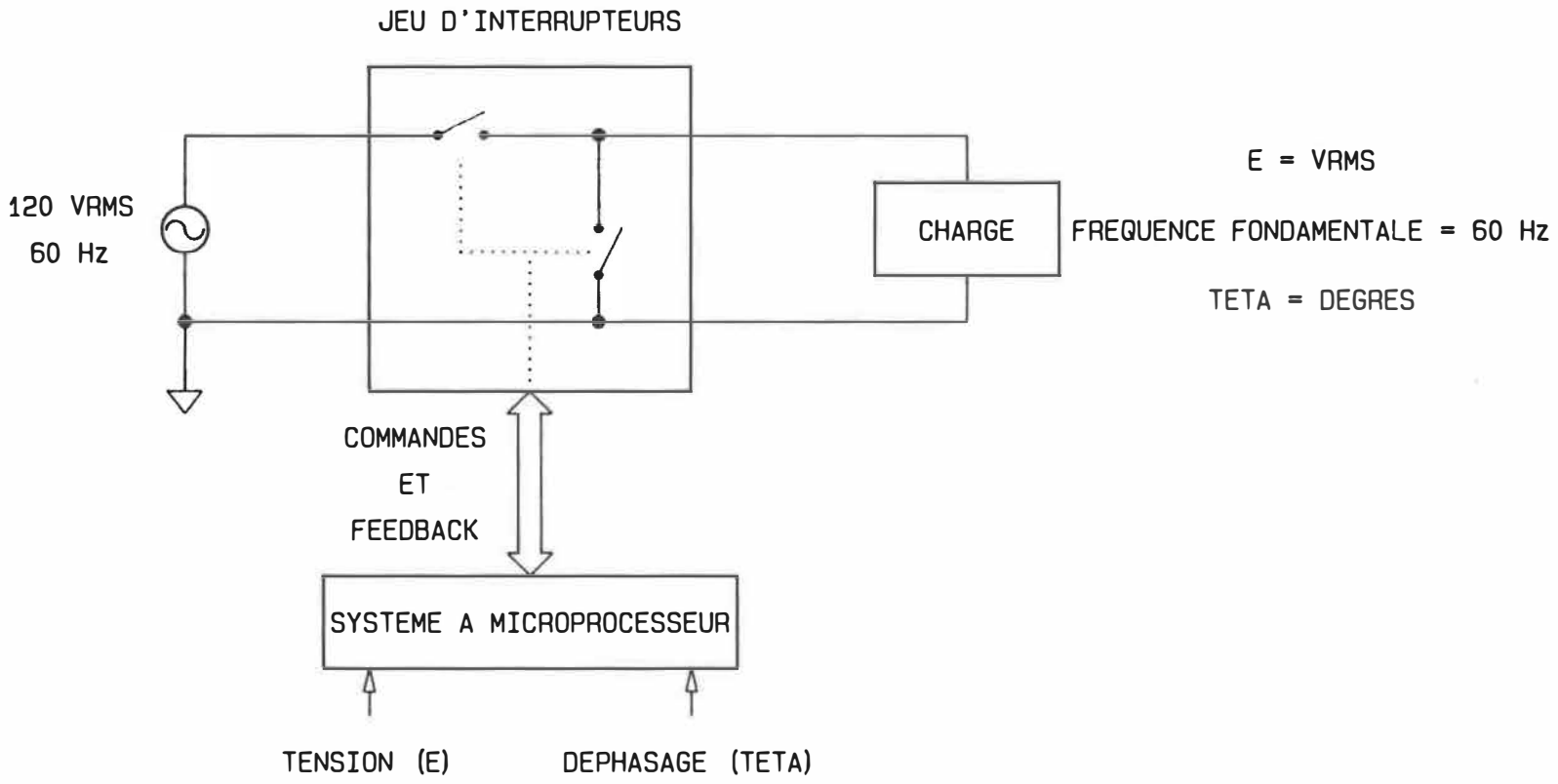


fig 1.0-1 COMPOSITION DU CONVERTISSEUR

d'interrupteurs commutant la tension d'entrée ou la charge en fonction des stratégies programmées et des consignes de tension et de phase. Il est appelé à entraîner un moteur asynchrone monophasé.

La relation entre le microprocesseur de commande et le reste du convertisseur ne dépend pas nécessairement de la charge, mais la stratégie d'opération des interrupteurs ainsi que le stress qu'ils subissent en dépendent. En pratique, le microordinateur doit prévoir des protections et des vérifications logicielles utiles pour les différents éléments constituant la chaîne "convertisseur-charge" lorsqu'utilisé avec sa charge normale.

Les stratégies d'opération sont dépendantes de la charge. C'est au programmeur de s'assurer qu'elles respectent les contraintes imposées par la charge. Plus loin, nous verront en détails les contraintes imposées par le moteur asynchrone à l'opération du convertisseur.

Afin d'obtenir une tension alternative de phase et d'amplitude commandées, au moins deux sources de tension alternatives déphasées sont nécessaires. Comme on fonctionne à partir d'une source monophasée, on peut s'en générer une seconde inversée (180 degrés de déphasage) à l'aide d'un transformateur à prise médiane ou utiliser un pont d'interrupteurs bipolaires comme convertisseur. Comme le but n'est pas de réaliser le meilleur choix pour une application industrielle mais bien d'utiliser des moyens simplifiant la réalisation (et sutout la commande de l'ensemble), on préfère utiliser un montage étoile avec un interrupteur de roue libre. La sortie du convertisseur est composée d'une partie de l'une, de l'autre ou d'aucune des sources de tension disponibles. Ce qui nous permet de s'approcher de l'une ou de l'autre des sources selon le temps passé sur chacune d'elles. Il s'agit donc d'un

multiplexage temporel de deux sources. La sortie est une combinaison de ces deux vecteurs déphasés. Une variation d'amplitude est rendue possible par la roue libre à la charge. On peut donc régler la tension efficace de sortie. La figure 1.0-2 illustre la structure adoptée. Elle nous permet de régler la tension efficace de sortie et son déphasage.

Dans ce qui suit, on verra plus en détails les contraintes imposées au convertisseur par le moteur asynchrone ainsi qu'une définition complète du système. Puis les détails importants du système à microprocesseur seront donnés. Enfin, l'organisation physique du convertisseur et sont système de commande tels que réalisés vous seront présentés.

1.1 Les contraintes imposées au convertisseur par le moteur asynchrone.

Le moteur asynchrone monophasé vu du stator apparaît comme une charge active de nature inductive. L'interdépendance des systèmes mécanique et électromagnétique qui forment le moteur en fait une charge active. A l'aide du modèle "DQ" de la machine, on peut très bien le visualiser. La figure 1.1-1 montre le circuit électrique équivalent du modèle donné par (1). On y remarque deux circuits symétriques qui correspondent au circuit équivalent vu aux bornes de chacun des enroulements statoriques. Ils sont interreliés par le flux. Après le démarrage, l'enroulement en quadrature s'ouvre; il ne reste que l'enroulement direct. Le modèle se simplifie donc. Le modèle résultant est donné en 1.1-1b. On peut simplifier davantage le modèle en régime sinusoïdal; on peut le voir

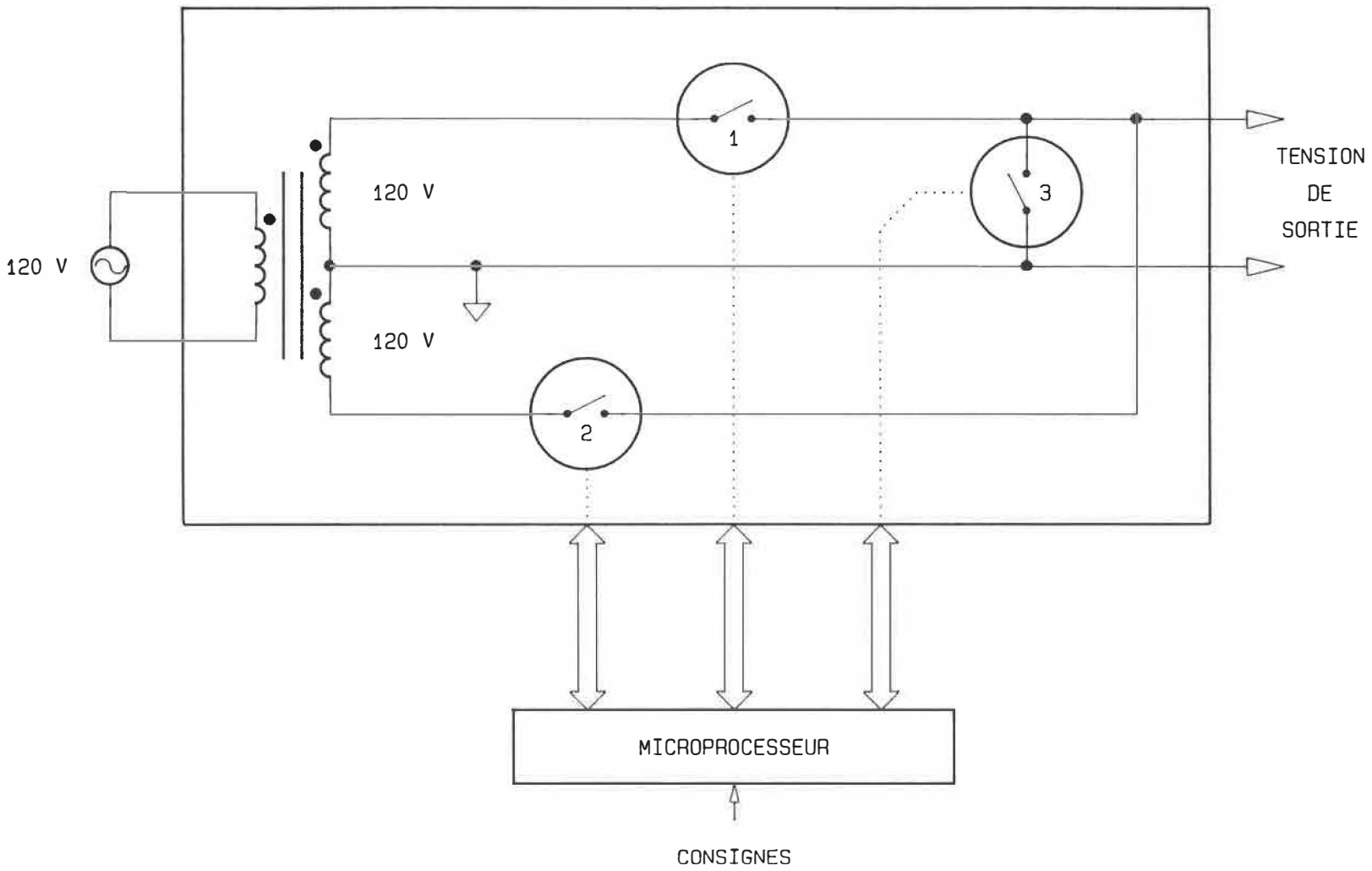


fig 1.0-2 TOPOLOGIE DU CONVERTISSEUR

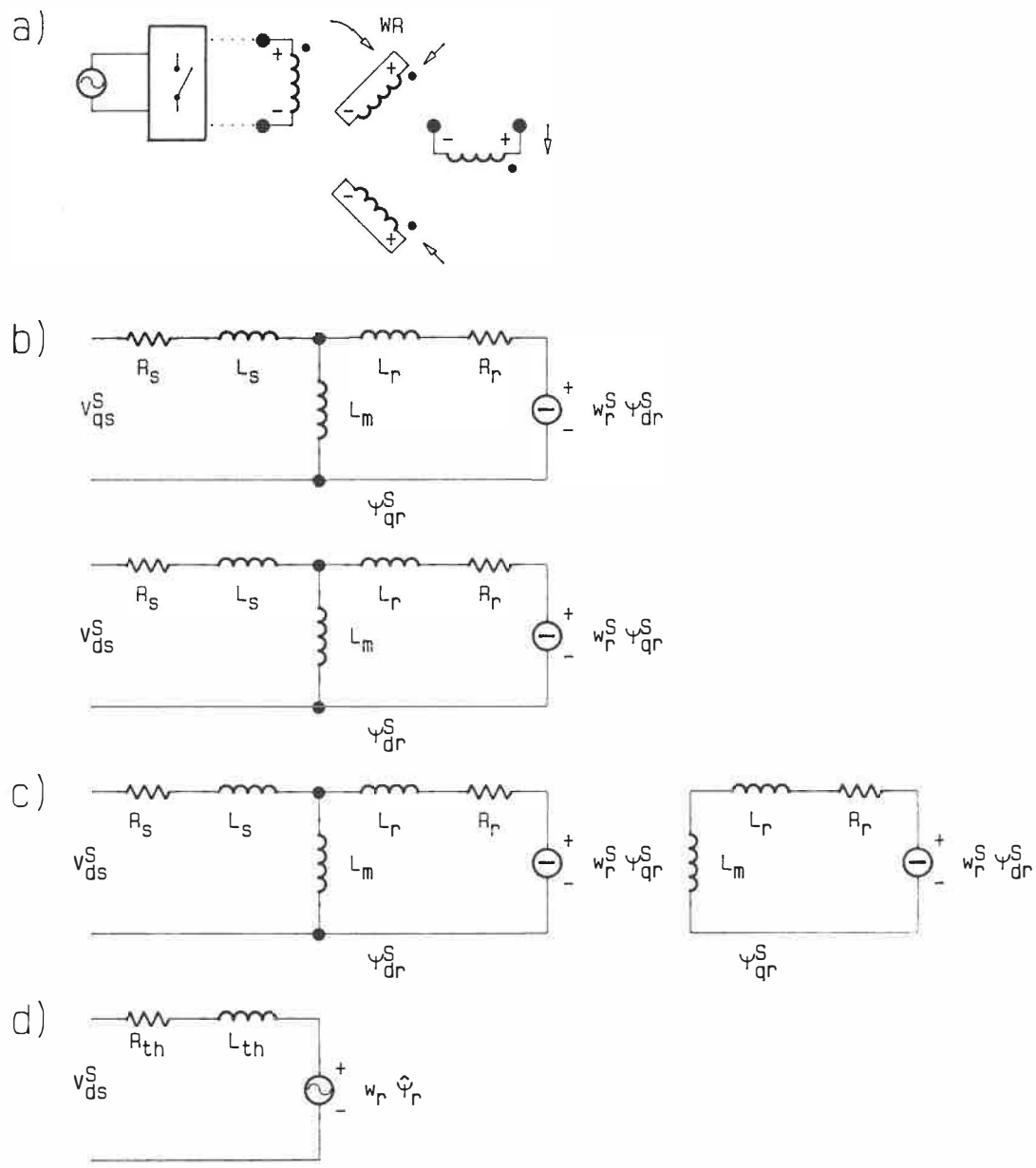


fig 1.1-1 CIRCUIT EQUIVALENT DERIVE DU MODELE DQ

a) SCHEMA EQUIVALENT DE LA MACHINE ASYNCHRONE

b) CIRCUIT EQUIVALENT VU DU STATOR

c) SIMPLIFICATION APRES DEMARRAGE

d) MODELE SIMPLIFIE EN REGIME SINUSOIDALE

en 1.1-1c. Ce dernier modèle nous aidera à déterminer la une stratégie sécuritaire pour le moteur et pour les interrupteurs.

Si on relie un convertisseur très simple formé que de deux interrupteurs, tel qu'illustré à la figure 1.1-2, on peut faire la constatation suivante : on ne peut pas couper de façon brusque le courant circulant dans la charge à cause de sa nature inductive; il faut prévoir son transfert par un autre chemin, autrement l'interrupteur s'ouvrant risque de claquer (e=L di/dt...). De plus, le modèle nous montre la tension contre-électromotrice. Si on passe en roue libre un bon moment (interrupteur 1 ouvert et interrupteur 2 fermé à la figure 1.1-2) et qu'on revient en tension directe (interrupteur 1 fermé et interrupteur 2 ouvert à la figure 1.1-2), instantannément il y a une différence entre la tension de source et de contre-électromotrice qui s'accroît davantage avec le temps. Lorsque l'interrupteur 1 se ferme, une variation du courant est due à cette différence de potentiel, et elle est limitée par la résistance (Rth) et l'inductance (Lth) équivalentes du moteur. Cette variation brusque se produit donc à chaque passage de l'état de roue libre à celui de tension directe. Dans le cas d'une stratégie symétrique à deux événements, une telle variation brusque de tension d'excitation entraîne une augmentation importante du courant statorique et produit un stress indésirable pour les interrupteurs du convertisseur.

A partir d'un logiciel de simulation on évalue l'effet d'une variation brusque de la consigne de tension entre deux cycles successifs. Par conséquent il y aura aussi une variation brusque d'un cycle au suivant des instants d'arrivé des états de tension directe et de roue libre. La figure 1.1-3 nous montre le courant et la tension lors d'un

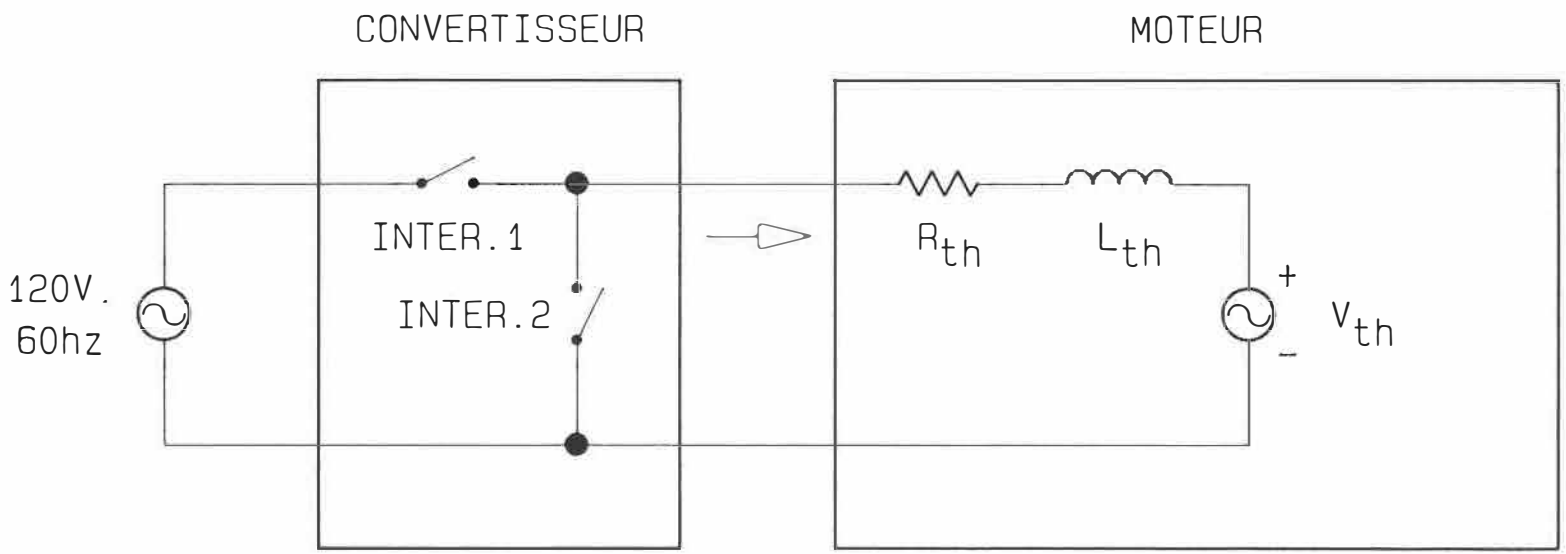


fig 1.1-2 CIRCUIT EQUIVALENT SIMPLIFIE DE LA MACHINE ALIMENTEE PAR UN CONVERTISSEUR SIMPLE

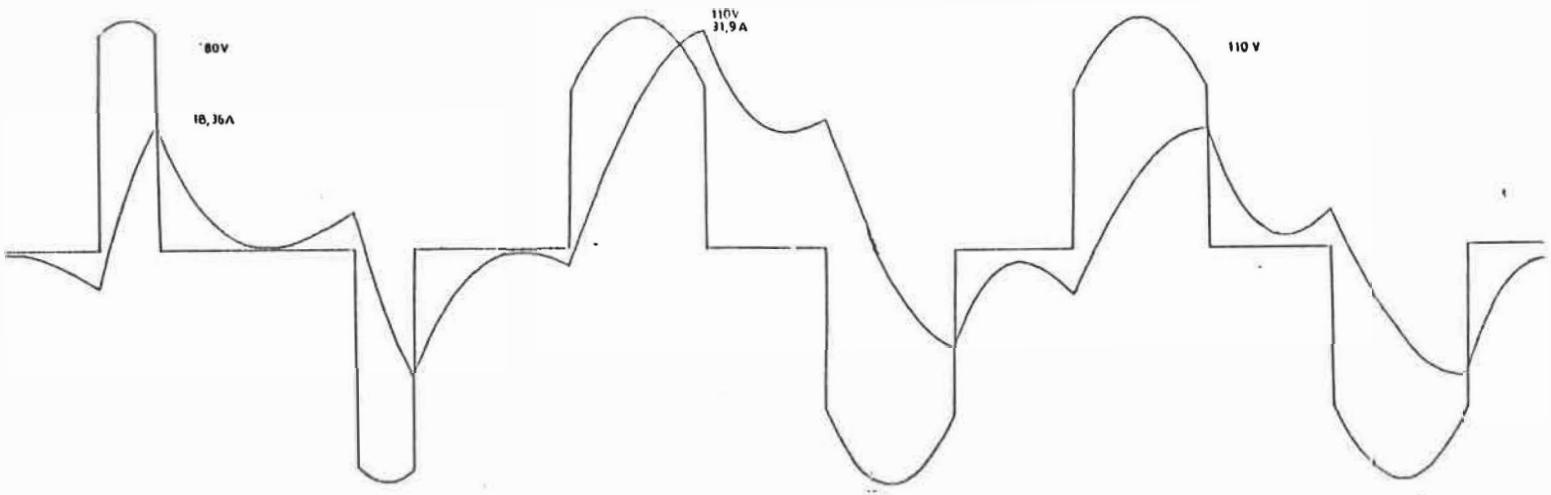


Figure 1.1-3 Variation de 80V à 110V en un cycle.

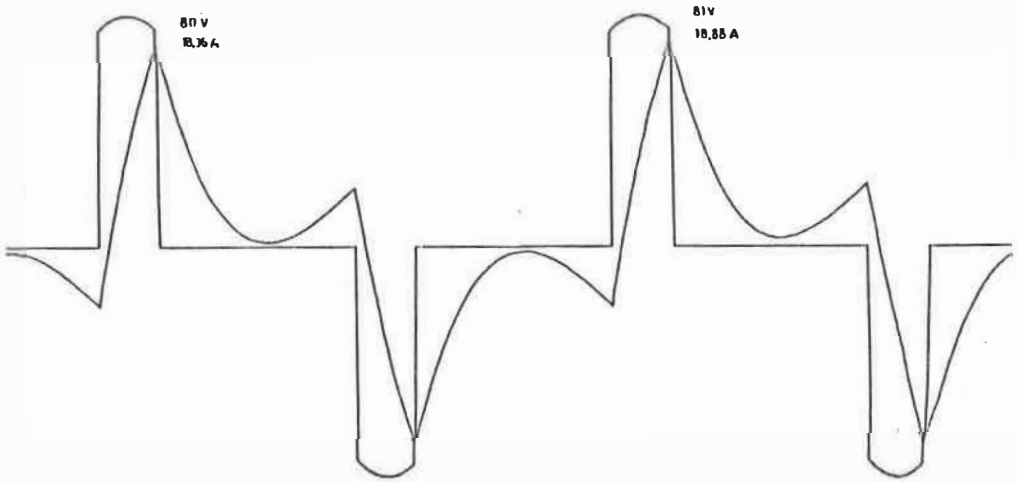
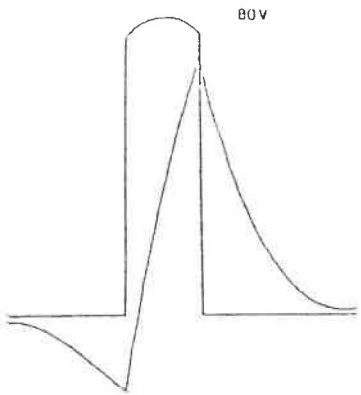


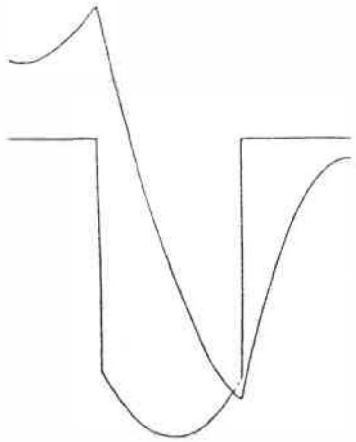
Figure 1.1-4 Progression de 1V par cycle.

essai où la consigne de tension passe de 80V à 110V au cycle suivant. Le courant maximal passe de 18.4A à 31.9A; une augmentation de 73% ! Cela pourrait signifier la destruction d'un interrupteur. On notera qu'à 120V, la valeur maximale du courant ne dépasse pas 23A. Lorsqu'on progresse lentement pour augmenter la tension, soit une augmentation de 1V par cycle, on obtient une augmentation de la valeur maximale du courant de 0.5A lors d'un essai simulé du passage de 80V à 81V le cycle suivant. On peut observer cette variation à la figure 1.1-4.

Il n'y a pas que la valeur maximale du courant à considérer; la durée de la transitoire aussi est importante. Pour le premier essai (figure 1.1-3), deux cycles après la variation de la consigne de tension, la transitoire du courant n'est pas encore terminée. Quatre cycles sont nécessaires pour atteindre le régime permanent. Il faut donc considérer qu'il s'est produit une une variation de 30V répartie sur 5 cycles, soit 6V par cycles. Si maintenant au lieu de varier de 30V en un cycle on variait plutôt la tension de 7.5V par cycle, 4 cycles successifs, on obtiendrait probablement une variation du courant maximal moindre que dans le premier cas ainsi qu'une transitoire beaucoup plus courte à la suite de cette progression sur 4 cycles. La figure 1.1-5 nous montre les courant et tension résultant de cet essai. On note que le courant maximal ne dépasse pas 24A durant ces 4 cycles. De plus, la transitoire qui suit est très courte: deux cycles après la dernière variation de 7.5V, le régime permanent est atteint.

Ces différentes simulations nous indiquent qu'une progression lente vers la consigne nous permet de contrôler les pointes de courant survenant lors de changements importants de consignes. De plus, le temps supplémentaire


80V



nécessaire à l'atteinte de la consigne n'est pas très important car, comme nous l'avons vu précédemment la différence entre les deux cas simulés est d'environ 1 cycle. On préférera prendre un peu plus de temps pour atteindre la consigne de tension plutôt que de détruire les interrupteurs lors de grandes transitoires.

On obtient donc des indications utiles sur la façon d'opérer les interrupteurs:

- il faut toujours prévoir un chemin pour le courant de charge lorsqu'on commute;
- répartir la variation de consigne de tension sur plusieurs cycles.

Il faudra introduire ces contraintes dans le programme de commande du convertisseur. En ce qui concerne la première contrainte, le programmeur devra aussi veiller à son respect lors de l'élaboration des stratégies d'opération du convertisseur.

On remarquera qu'une variation brusque de la charge entraîne aussi des variations sur le courant demandé par le moteur. Il y a aussi des pointes de courant qui en résultent. Cela est, en fait, indépendant du convertisseur; même en régime sinusoïdal on noterait ces pointes de courant. Mais il faudra quand même en tenir compte lors du choix des stratégies d'opération des interrupteurs du convertisseur, car certaines stratégies permettent, en temps normal, des courants de pointe plus élevés que d'autres. Si une variation de charge les amplifie encore, il y a risque de destruction des interrupteurs par courant instantanné trop élevé. Nous verrons cela au chapitre 2 qui traitera en détail le choix des différentes stratégies retenues.

1.2 définition du problème.

Le convertisseur est commandé par microprocesseur. Il doit permettre de générer une tension de sortie d'amplitude et de déphasage ajustables par rapport à la tension de source, selon une certaine stratégie d'opération des interrupteurs. De plus, comme nous l'avons vu plus tôt, le convertisseur doit respecter certaines contraintes tout en répondant à nos besoins. On peut résumer ces principaux points à l'aide de la liste suivante:

- commander les interrupteurs selon une stratégie programmée
  et en fonction des consignes de tension et de déphasage,
- progresser sans heurts (à un faible taux de variation de la consigne par cycle) vers les consignes en évitant les grandes transitoires de courant aux interrupteurs,
- les consignes sont fournies par un second microprocesseur,
- être peu sensible à une perte temporaire de synchronisation au réseau,
- permettre un faible nombre de commutations par cycle (moins de 20),
- permettre une programmation simple des stratégies d'opération des interrupteurs du convertisseur,
- pouvoir changer de stratégie en fonctionnement (doit donc permettre le choix entre différentes stratégies et prévoir une façon d'en changer).

Les consignes de tension et de déphasage sont fournies par un second microordinateur (le #2). Ce dernier indique aussi la stratégie à utiliser parmis une certaine banque disponible au microordinateur commandant le convertisseur

(le #1). Le schéma de principe du système à multi-processeur est donné à la figure 1.2-1. On remarquera que varier le déphasage fait aussi varier la tension efficace de sortie du convertisseur pour de mêmes angles d'arrivée des événements d'une stratégie. On ne peut donc pas qu'utiliser des tables donnant des instants d'arrivée des événements; on doit fermer la boucle en mesurant la tension de sortie et en la comparant à la consigne de tension et agir de façon à réduire la différence. Afin de déphaser la tension de sortie et de se synchroniser à celle d'entrée, on a besoin d'une détection du passage par zéro de la tension d'entrée. Le déphasage entre la sortie et l'entrée est obtenu à l'aide d'horloges internes verrouillées au réseau mais déphasées. La mesure du courant de sortie est essentielle pour l'opération des interrupteurs. Celle de la tension de sortie l'est pour réguler la tension. Celle de la source est nécessaire pour la détection du passage par zéro servant de signal de synchronisation. Les horloges internes le sont pour la synchronisation, le déphasage et l'exécution des événements. Le port de communication entre les deux microordinateurs permet le passage des paramètres nécessaires à la commande de l'ensemble du système. Le schéma-bloc du microordinateur de commande et ses interfaces sont donnés à la figure 1.2-2.

Les événements (opérations sur les interrupteurs arrivant à un moment précis dans le cycle) sont exécutés en temps réel. Ils sont référés à une horloge interne synchronisée au réseau et non les uns par rapport aux autres. Ils sont prioritaires.

La progression vers les consignes se fait cycle par cycle: tous les événements arrivant dans un cycle sont modifiés et utilisés au début d'un nouveau cycle. La

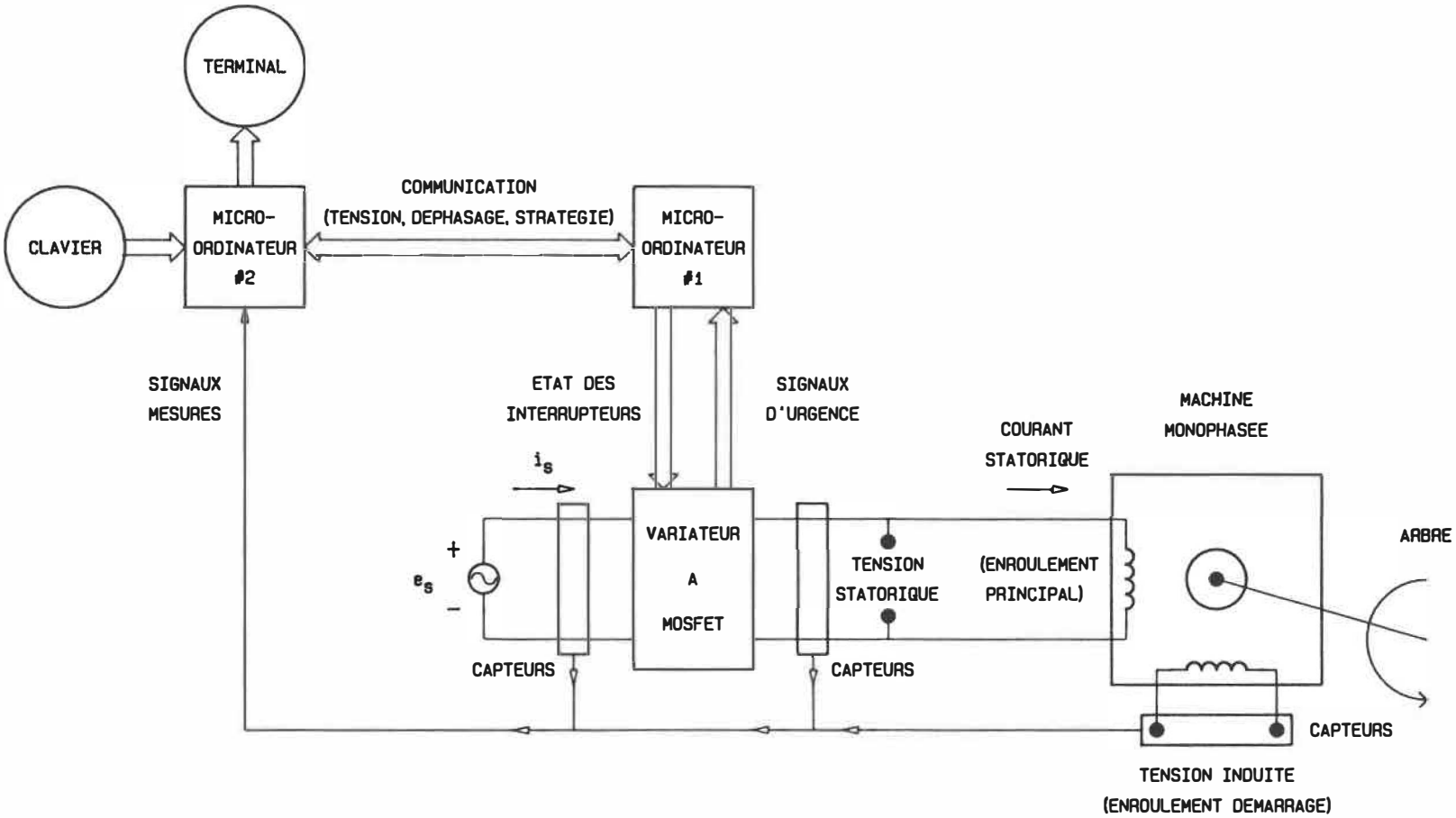


fig 1.2-1 SCHEMA DE PRINCIPES DE L'ENTRAINEMENT

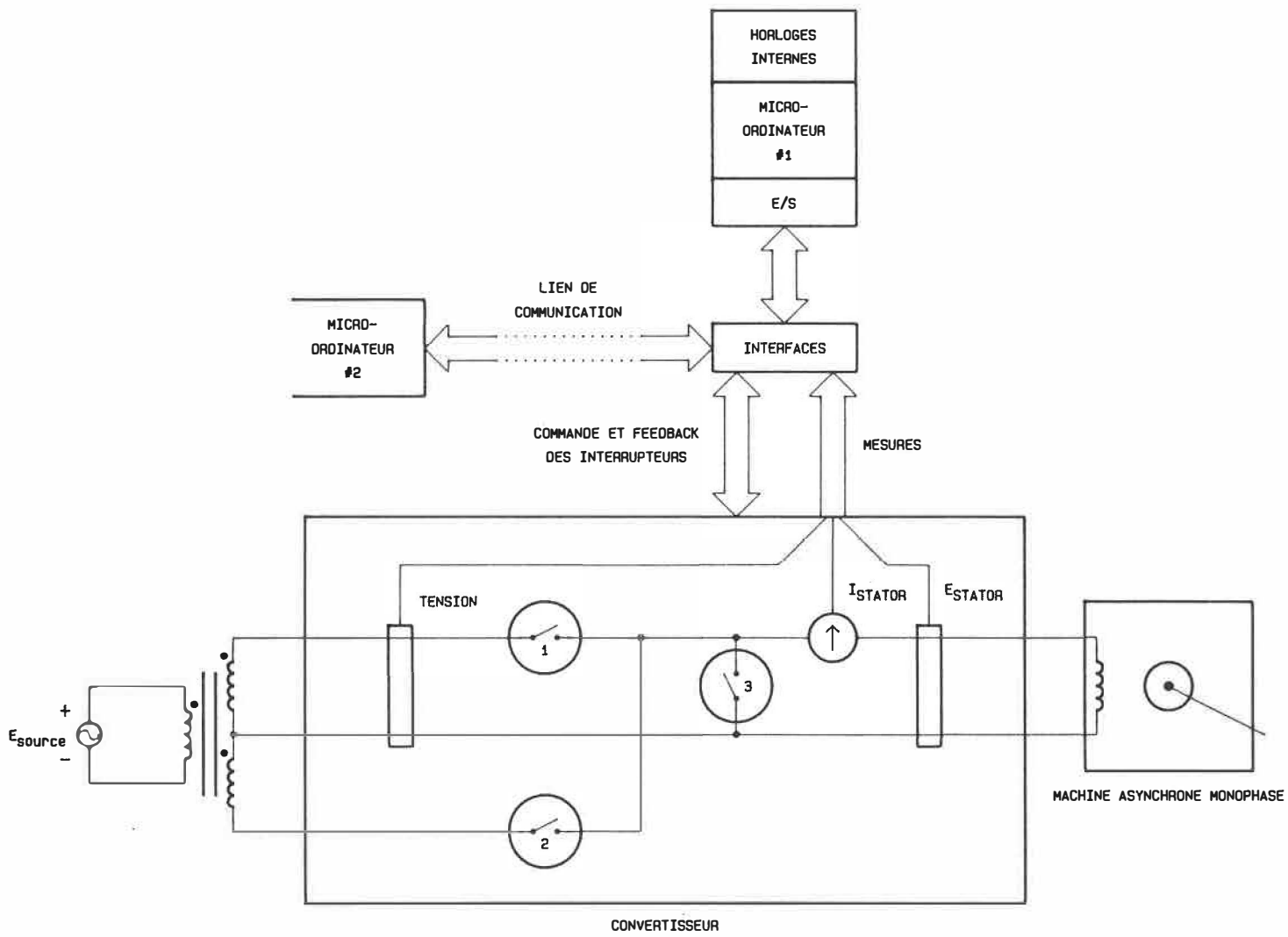


fig 1.2-2 LE MICROORDINATEUR DE COMMANDE DU CONVERTISSEUR

consigne reste donc constante un nombre entier de cycles, assurant ainsi une tension moyenne nulle. La progression vers les consignes peut être ralentie afin de répondre aux besoins prioritaires du programme, c'est-à-dire à la synchronisation et à l'exécution des événements .

Il faut aussi se synchroniser au réseau. On devra le faire sans que cela ne crée de sous-harmoniques ou encore de débalancement entre les alternances. On privilégie l'approche à verrouillage numérique (P.L.L.) avec maintien de symétrie: même si le signal de synchronisation (provenant du détecteur de passage par zéro) était perdu, la boucle continuerait à osciller à la fréquence qu'avait le signal avant sa disparition et se reverrouillera à ce dernier lorsqu'il réapparaîtra. De plus, la symétrie évite intrinsèquement le débalancement entre les alternances.

En ce qui concerne la programmation des stratégies, on veut se limiter à quelques types d'opérations de bases pouvant les former. On retient les opérations suivantes: tension directe, tension inverse et roue libre. Eventuellement, d'autres opérations pourraient être permises. L'état, ouvert ou fermé, d'un interrupteur dépend d'un signal logique envoyé vers les interrupteurs du convertisseur. Les stratégies et leurs paramètres de définition sont mis dans des tables à la fin du programme (dans un ROM contenant les données). Le nombre de octets requis pour définir une stratégie est alors proportionnel au nombre d'événements de la stratégie.

Les opérations prioritaires, c'est-à-dire la synchronisation et l'exécution des événements, devraient être appelés par interruption: ces opérations ne durent qu'un court moment et arrive à des instants précis et prévisibles dans le cycle (mais variables selon la consigne de tension pour les événements). On a besoin alors d'un

serveur pour ces opérations. Les autres parties du logiciel qui ne servent, en fin de compte, qu'à déterminer les instants d'arrivées et le type d'opération des événements, peuvent être traitées à temps "perdu", dans les fenêtres de temps laissées libres entre l'exécution des opérations prioritaires. On ne veut donc pas rester en attente des instants où doivent se produire les événements; alors, on programme (une horloge) un générateur d'intervalle qui générera une interruption au bon moment; les instants d'attente peuvent être utilisés judicieusement par les autres parties du logiciel. On se retrouve alors avec deux niveaux dans le logiciel: un pour exécuter, aux instants requis, les événements et assurer la synchronisation, et un second niveau assurant l'atteinte des consignes en modifiant les instants d'arrivée demandés pour chacun des événements et fournir la consigne de déphasage. Il y a donc deux volets: un "d'exécution" et un second de "gestion-préparation". Cela déterminera les grandes divisions de la structure hiérarchique du logiciel.

Précisons un peu la façon d'obtenir les événements et surtout leur séquencement. On ne voudrait pas que l'événement suivant soit affecté par l'imprécision sur l'instant d'arrivée du premier et ainsi de suite jusqu'à la fin. On a donc besoin d'une référence servant au calcul de l'instant d'arrivée de chacun des événements de façon indépendante (et non pas lancer un événement tant de microsecondes après le précédent). Ainsi, l'incertitude sur l'instant d'arrivée reste constante pour chacun des événements et n'est pas cumulative comme lorsqu'on calcul des instants relatifs d'un événement par rapport à l'autre (le deuxième événement ayant une incertitude deux fois plus grande que le premier, le troisième, trois fois plus grande que le premier et ainsi de suite). On se retrouve avec la

notion d'horloge du cycle (la référence) et une horloge d'événement contenant le temps d'attente entre le moment où on a calculé et le temps d'arrivée prévu (à l'horloge du cycle).

Dans cette section, nous avons précisé les fonctions et contraintes de base du système au point de vue du logiciel. Nous avons établi la nécessité d'une structure à deux volets dont un sert à l'exécution des événements et la synchronisation au réseau et s'exécute en temps réel, tandis que l'autre assure l'atteinte des consignes en occupant les fenêtres de temps laissées libres par le précédent. Nous avons déterminé la façon de séquencer les événements en se référant à une horloge interne minimisant ainsi les incertitudes sur l'instant d'arrivée des événements. Nous avons aussi arrêté la manière de programmer les stratégies: à l'aide de tables donnant la séquence des opérations sur les interrupteurs formant les événements de la stratégie. Nous avons donné un schéma de principe du convertisseur et de son système de commande à microprocesseur. Dans la section suivante, nous verrons le système beaucoup plus en détails au point de vue du microprocesseur.

1.3 Le système à microprocesseur.

Nous allons disséquer le système à microprocesseur commandant le convertisseur. Tout d'abord, nous commencerons par définir les entrées/sorties, la structure des données, le diagramme de flux des données puis le diagramme hiérarchique. Suivra une description du matériel; des capteurs jusqu'au microprocesseur. Une description détaillée des procédures importantes du logiciel et des

détails d'intérêts sur certaines parties du logiciel termineront la section.

1.3.1 Eléments de base du logiciel.

Commençons par préciser les entrées et les sorties du système à microprocesseur. Du schéma donné à la figure 1.2-2, on trouve:

ENTREES:

- Horloge interne (lecture)
- Lien de communication
- Chaîne de mesure
- Signal de synchronisation (détection du passage par zéro de la tension de source: feedback des interrupteurs

SORTIES:

- Horloges internes (écriture)
- Commande des interrupteurs du convertisseur
- Commandes d'acquisition et de sélection (agit sur le matériel de la chaîne de mesure)

Comme nous l'avons vu à la section précédente, il y a des opérations prioritaires qu'il faut traiter avant les autres, c'est-à-dire en temps réel et au bon moment. Ces opérations sont reliées aux événements à exécuter au convertisseur et à la synchronisation. Ils forment la partie "EXECUTION" du logiciel; le reste ne servant qu'à modifier les "événements". Cette dernière partie en est une de gestion, de préparation. On préconise une structure de donnée formée de jeux de tables dont se servirait la partie "exécution". La partie "gestion" modifierait ces jeux de tables, en fonction des consignes reçues et de l'état du système, et le signalerait à l'aide de drapeaux. En fait,

un double jeux de tables est envisagé: un en utilisation et un second en préparation ou en attente. Cela évite toute confusion pouvant survenir lorsque deux parties de logiciel utilisent la même mémoire: un événement étant modifié avant qu'il ne soit exécuté entraînant par la suite un débalancement dans le cycle par exemple. C'est pourquoi on travaille sur un jeu de table, pendant qu'un second est en exécution; le changement de jeux de tables se faisant à la fin du cycle en vue du suivant: on vérifie si un jeux de tables est prêt (à l'aide de drapeaux ou d'index), si c'est le cas, on le prend comme jeux de tables pour exécution et on le signale. Les jeux de tables constituent l'information sur le nombre d'événement par cycle, l'instant d'arrivée des événements et le type d'opération à faire pour un événement donné, c'est-à-dire la stratégie d'opération des interrupteurs. La figure 1.3.1-1 illustre l'organisation en deux volets. On notera que le volet "gestion-préparation" occupe les fenêtres laissées libres par le volet "exécution". De plus, on ne peut modifier les jeux de tables qu'une fois par cycle afin de pouvoir exécuter les modifications préparées au cycle suivant. Le volet gestion doit donc vérifier si un nouveau cycle au moins est arrivé depuis la dernière modification aux jeux de tables. On y montre le lien entre les deux parties "quasie indépendantes" du logiciel, c'est-à-dire un lien par la mémoire.

On peut maintenant mieux comprendre le diagramme de flux des données présentés à la figure 1.3.1-2. Différentes horloges y sont nommées; ce sont les "horloges de cycle", l'horloge d'évévement, et le compteur de périodes. Leur rôle est fonction des contraintes amenées par le séquencement des événements et la synchronisation. Autant d'horloges de cycles sont requises à cause de la

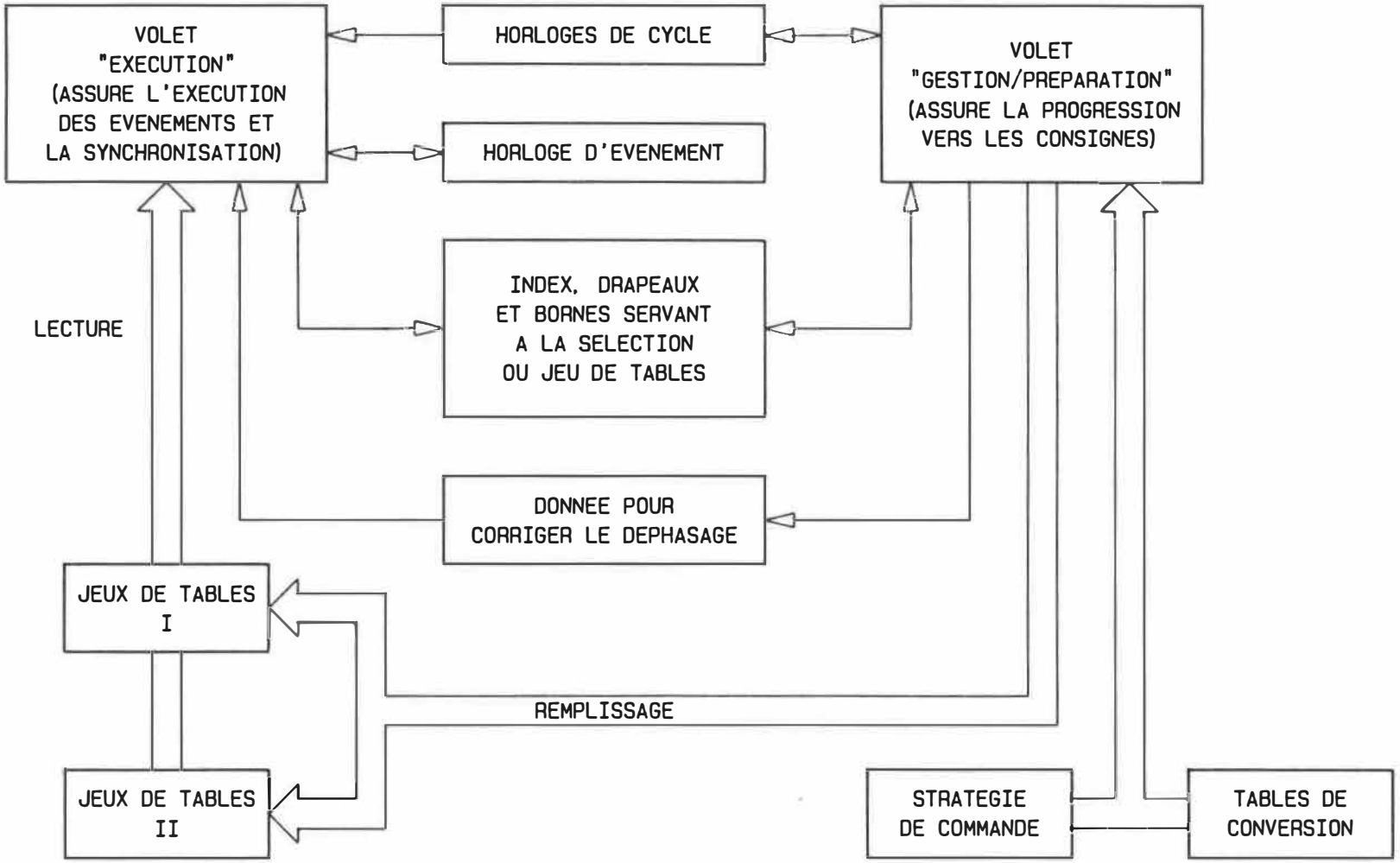


fig 1.3-1 LES DEUX VOLETS DU LOGICIEL

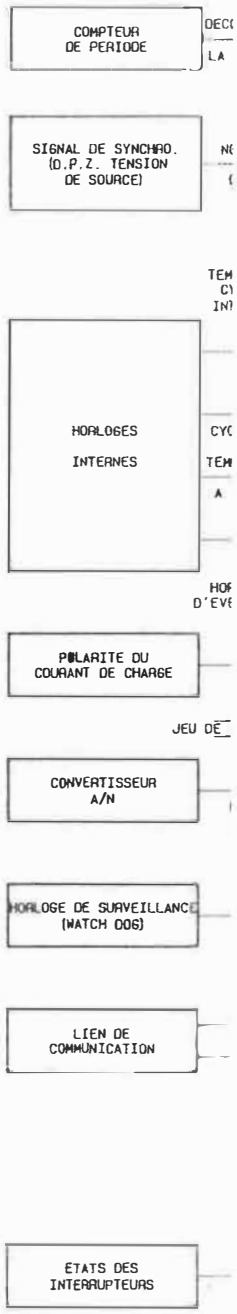

COMPTEUR
DE PERIODE
SIGNAL DE SYNCHRO.
(D.P.Z. TENSION
DE SOURCE)
HORLOGES
INTERNES
POLARITE DU
COURANT DE CHARGE
CONVERTISSEUR
A/N
HORLOGE DE SURVEILLANCE
(WATCH DOG)
LIEN DE
COMMUNICATION
ETATS DES
INTERRUPTEURS

FIGURE 1

possibilité de déphaser la sortie par rapport à la tension de source.

On se sert du détecteur de passage par zéro pour déterminer la durée d'un cycle réseau. Cette valeur nous sert à réajuster la période d'une horloge interne cyclique, appelée "horloge du cycle". Cette horloge nous sert de "PLL (numérique)": elle est verrouillée au réseau. Une perte temporaire du signal de synchronisation provenant du D.P.Z. n'est pas catastrophique car l'horloge du cycle garde la dernière période mesurée jusqu'à ce qu'on la renouvelle, c'est-à-dire que la synchronisation revienne. Afin d'obtenir une mesure précise du cycle réseau, le D.P.Z. provoque une interruption (non masquable) au microprocesseur. Le microprocesseur lit et mémorise le contenu d'un "compteur de période", puis le relance. L'information mémorisée servira, à un moment opportun, au calcul de la période du cycle et à l'ajustement de l'horloge du cycle. Afin de permettre le déphasage par rapport au réseau, on utilise une autre "horloge", dite "déphasée", synchronisée sur l'horloge du cycle mais dont la phase est réglable (on peut l'avancer seulement). Cette horloge déphasée nous sert de référence de temps pour calculer le délai avant le prochain événement, délai dont on charge l'horloge d'événement. Cette dernière provoquera une interruption (de type masquable) lorsque le délai sera écoulé permettant ainsi au microprocesseur d'exécuter l'événement au bon moment. Finalement, une horloge déphasée de 180° par rapport à notre horloge de cycle, qu'on appelera "horloge à 180°", nous sert à respecter les alternances positives et négatives: on peut alors sélectionner le bon groupe d'interrupteurs et maintenir la symétrie entre les alternances.

Nous verrons plus loin quelques précisions sur le

fonctionnement des horloges et leur relation avec le déphasage.

1.3.2 Le diagramme hiérarchique.

Maintenant que les données, leur traitements ainsi que les contraintes sont mieux connues, on peut montrer le diagramme hiérarchique du logiciel (l'ensemble des modules prévus pour le programme dans une structure arborescente allant du général au détaillé).

On peut scinder le diagramme en deux parties: celle s'occupant de l'exécution des événements et de la synchronisation et celle s'occupant de la "gestion". C'est une séparation selon les deux volets présentés plus tôt: Le volet "exécution" formé par les deux modules appelés par interruption et le principal et le volet "gestion/préparation" formé par le reste et le principal. Cela est très logique: la partie "exécution" n'intervenant qu'à des instants précis dans le cycle et la partie "gestion" occupant le reste du temps; le lien entre les deux se faisant par des drapeaux, index et jeux de tables.

La figure 1.3.2-1 nous montre le diagramme. Une interruption non masquable est ultra prioritaire et on ne peut pas empêcher son exécution (en temps normal si on ne modifie pas l'adresse dans le vecteur d'interruption ...) on s'en sert donc pour la synchronisation et pour les cas de défaut qui ne peuvent attendre. Le type masquable nous sert à l'exécution des événements; exécution que le module principal ou celui détectant les défauts peut interdire ou permettre selon les conditions au convertisseur.

Dans la partie "gestion", on retrouve les modules principaux dont deux ne servant que rarement soient: celui d'initialisation du système et celui de réglage de tension

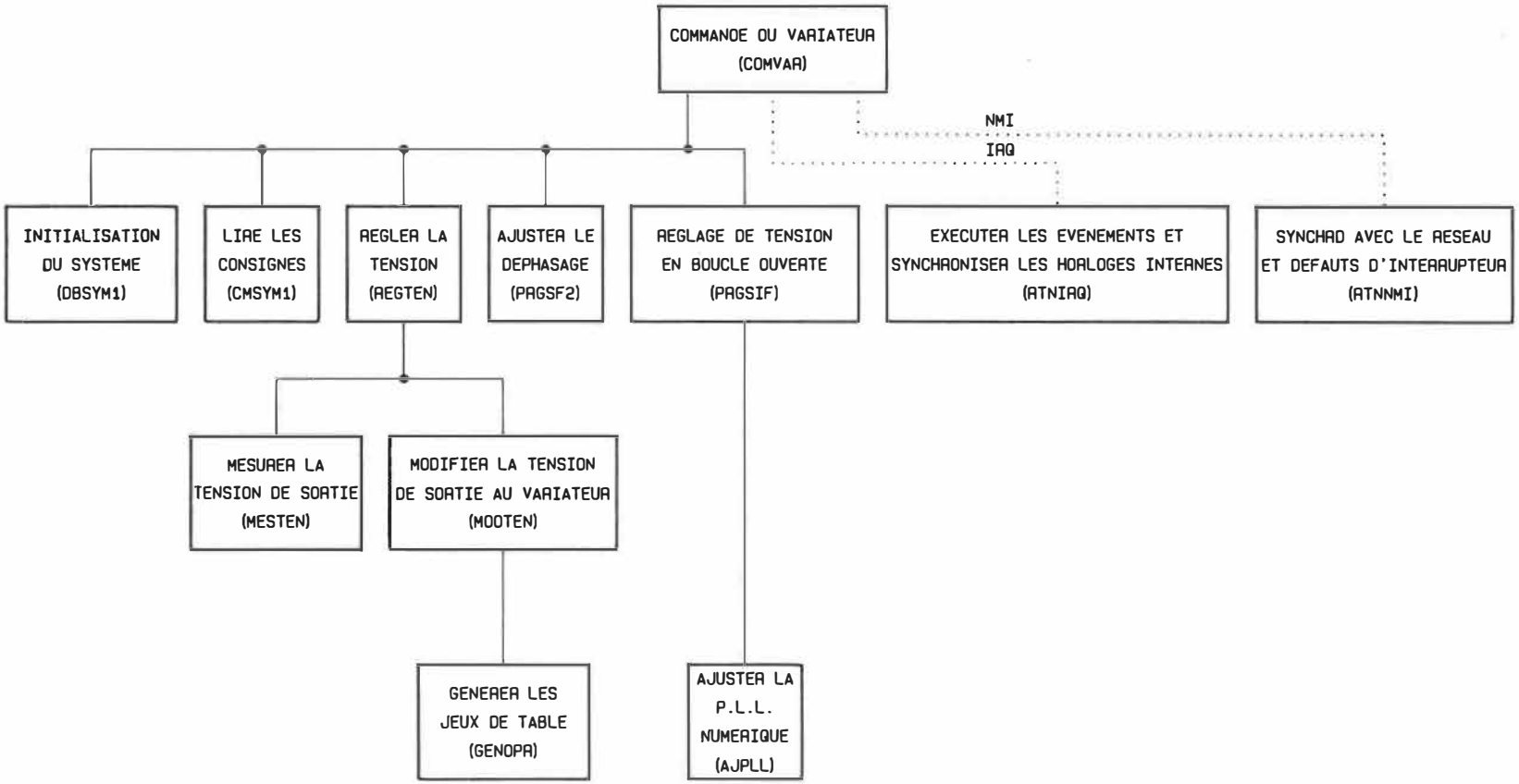


fig 1.3.2-1 DIAGRAMME HIERARCHIQUE DU LOGICIEL DE COMMANDE DU CONVERTISSEUR

en boucle ouverte. Le premier ne sert qu'au début et le second ne sert que lorsqu'on utilise une nouvelle stratégie: il permet les transitions. Les quatre autres modules forment le noyau de la partie "gestion": la lecture des consignes, le réglage du déphasage, le réglage de la tension et l'ajustement de la période du cycle de la PLL numérique. Ce sont les fonctions qui s'assurent de satisfaire aux besoins de base du convertisseur: tension et déphasage de sorties réglés selon les consignes.

Les procédures des principaux modules apparaissant dans le diagramme hiérarchique seront présentées après les schéma-bloc du matériel les interfaces nécessaires pour satisfaire aux entrées et sorties du système et à la réalisation des fonctions des différents modules.

1.3.3 le matériel.

On peut regarder le matériel associé à un système à microprocesseur de façon modulaire tel que montré à la figure 1.3.3-1. Dans notre cas, on regarde les modules d'interface d'entrée et de sortie au microprocesseur en même temps car ils sont reliés à deux VIA 6522; l'interfaçage au microprocesseur est assez direct. Les modules de conditionnement des signaux seront vus de façon séparée. On n'insistera pas sur les détails. Nous séparons la section en trois: le conditionnement des signaux, l'interfaçage au microprocesseur et le module microprocesseur au sens large.

1.3.3.1 le conditionnement des signaux.

Deux modules sont visés: celui de conditionnement des signaux d'entrée et celui de conditionnement des signaux de

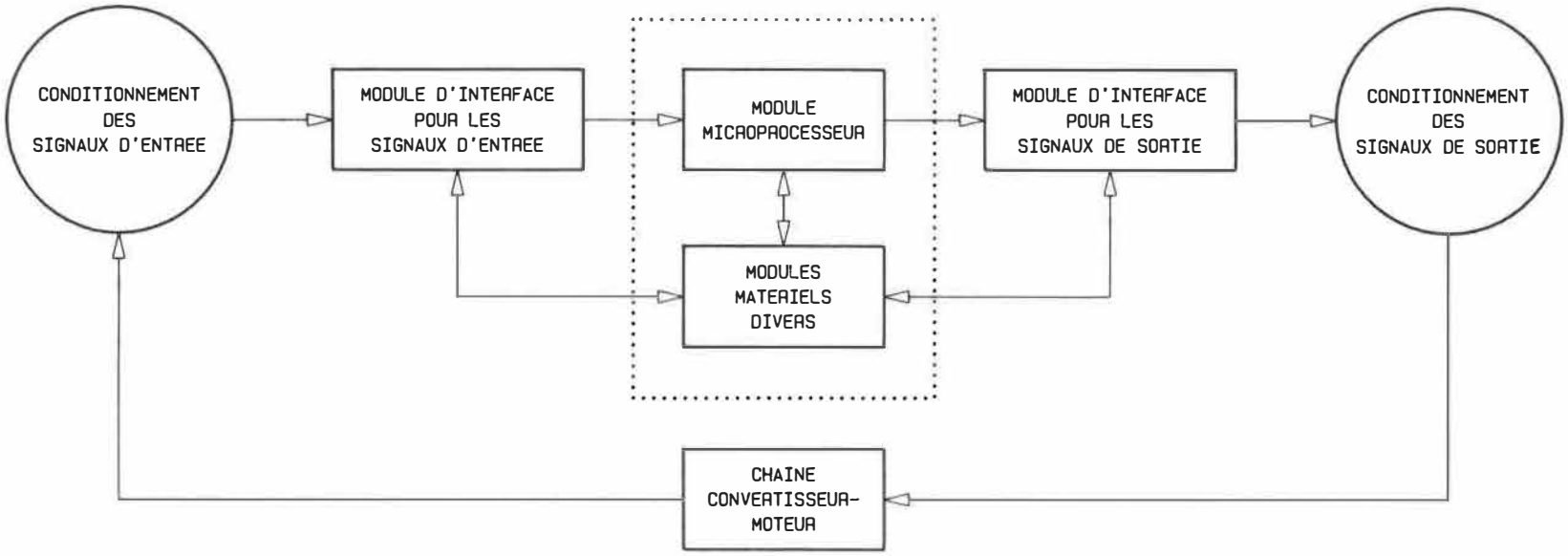


fig 1.3.3-1 LES MODULES MATERIELS DU SYSTEME DE COMMANDE DU CONVERTISSEUR

sortie. Le module de conditionnement des signaux d'entrée permet la mesure des signaux d'intérêts et leur différentes conversions. La figure 1.3.3.1-1 nous illustre les différents éléments du conditionnement requis; les entrées sont le courant et la tension statorique de la machine asynchrone (ou d'une autre charge). A partir d'eux, on obtient le signal de synchronisation, le signal du courant, la mesure efficace (RMS) de la tension, ainsi que d'autres signaux nécessaires au matériel pour la commande des interrupteurs. Il y a aussi le lien de communication à considérer comme entrée. Il est relié au module d'interfaçage.

Dans le module de conditionnement des signaux d'entrée,un capteur à effet Hall est utilisé pour la mesure du courant statorique. Un amplificateur d'isolation permet la mesure de la tension, sans problèmes de référence, tout comme la sonde à effet Hall. Un détecteur de passage par zéro est branché sur la tension de source, il fournit le signal nécessaire à la synchronisation au réseau. Les signaux de courant et de tension sont ensuite amplifiés puis dirigés vers un multiplexeur suivi d'un convertisseur donnant la vraie valeur efficace, d'un échantillonneur bloqueur d'un filtre limiteur et d'un convertisseur analogique/numérique. En plus, la sortie de l'amplificateur différentiel du courant sert d'entrée à deux réseaux de comparaison fournissant le signe du courant (CRNPOS) et une indication lorsque son amplitude est près de zéro (epsilon). Le feedback des interrupteurs (signalant les défauts) fournit un second signal pouvant générer une interruption non masquable. En plus, l'état de chacun des interrupteurs est envoyé au microprocesseur. Le lien de communication est un lien parallèle de 8 bits de donnée et est mis en parallèle avec la sortie du convertisseur analogique/numérique; le

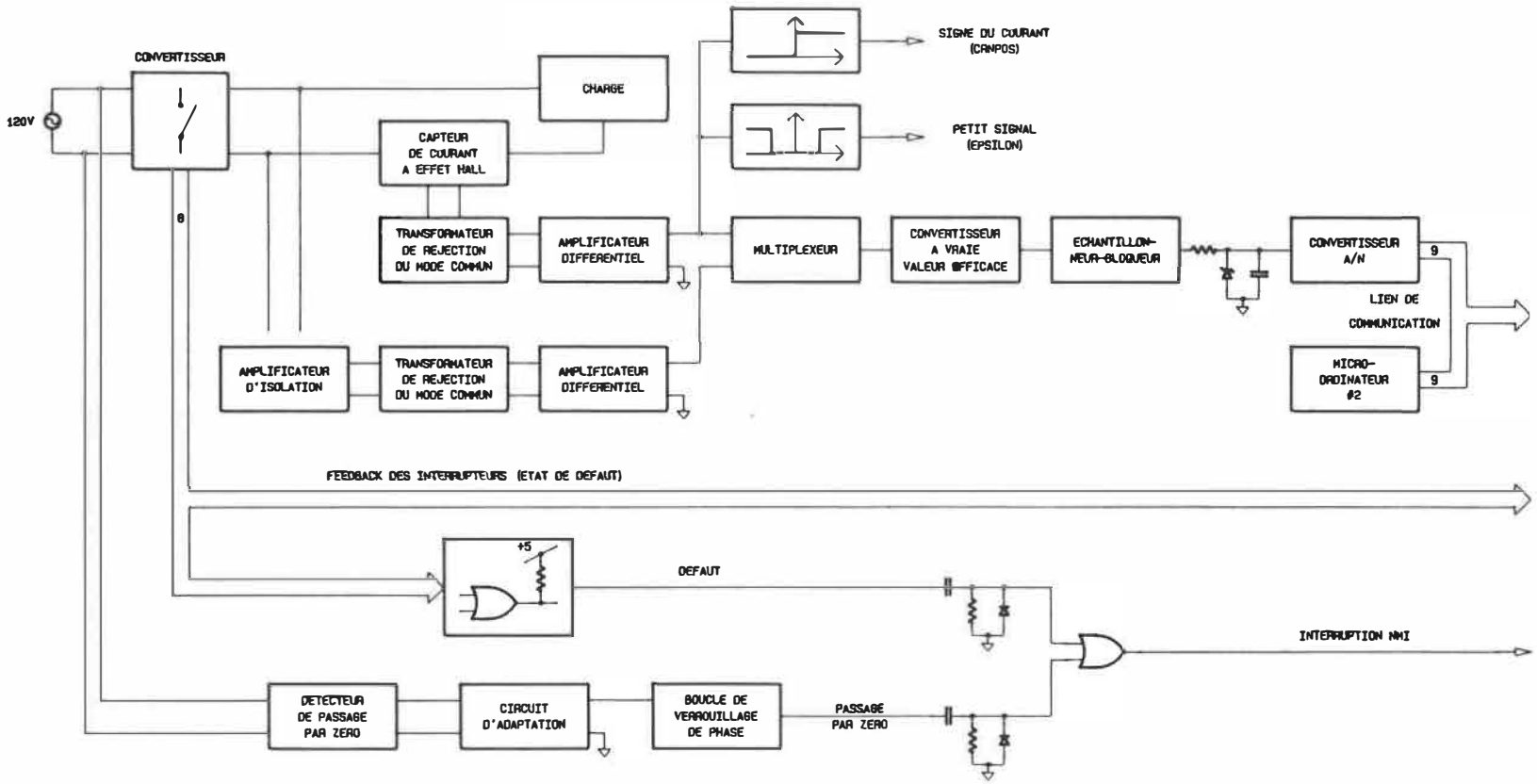


fig 1.3.3.1-1 MODULE DE CONDITIONNEMENT DES SIGNAUX D'ENTREE

microprocesseur les multiplexera.

Le module de conditionnement des signaux de sortie, quant à lui, permet principalement la commande des interrupteurs; nous ne reparlerons pas du lien de communication qui est bidirectionnel. La figure 1.3.3.1-2 nous illustre le transfert de l'information servant à commander les interrupteurs, du VIA aux interrupteurs eux-même. Les interrupteurs sont divisés en trois groupes: direct, d'inversion et de roue libre. Ils sont aussi pairés de façon à réaliser des interrupteurs bipolaires. On se sert des signaux "Epsilon" et "CRNPOS" fournis par le module de conditionnement des signaux d'entrée pour sélectionner la direction de conduction des interrupteurs bipolaires, c'est-à-dire de sélectionner lequel des deux interrupteurs doit conduire en fonction du courant à la charge. La logique s'assure d'un certain chevauchement autour de zéro (signal "Epsilon") pour la sélection des interrupteurs positif et négatif dans une zone d'incertitude des mesures autour d'un courant nul. Nous verrons plus loin le format de la commande associée à ce matériel dans la section 1.3.4.

1.3.3.2 L'interfaçage des entrées et des sorties au microprocesseur.

Principalement réalisé à l'aide des deux VIA disponibles au module "microprocesseur" on s'en sert pour recevoir et transmettre de l'information dans une forme compréhensible au microprocesseur et son environnement.

Les VIA 6522 contiennent, entres autres, des ports d'entrée/sortie et des lignes dites de commande appelées CA1 et CA2. A l'aide de ces lignes et d'un port d'entrée/sortie, on peut réaliser facilement un lien de

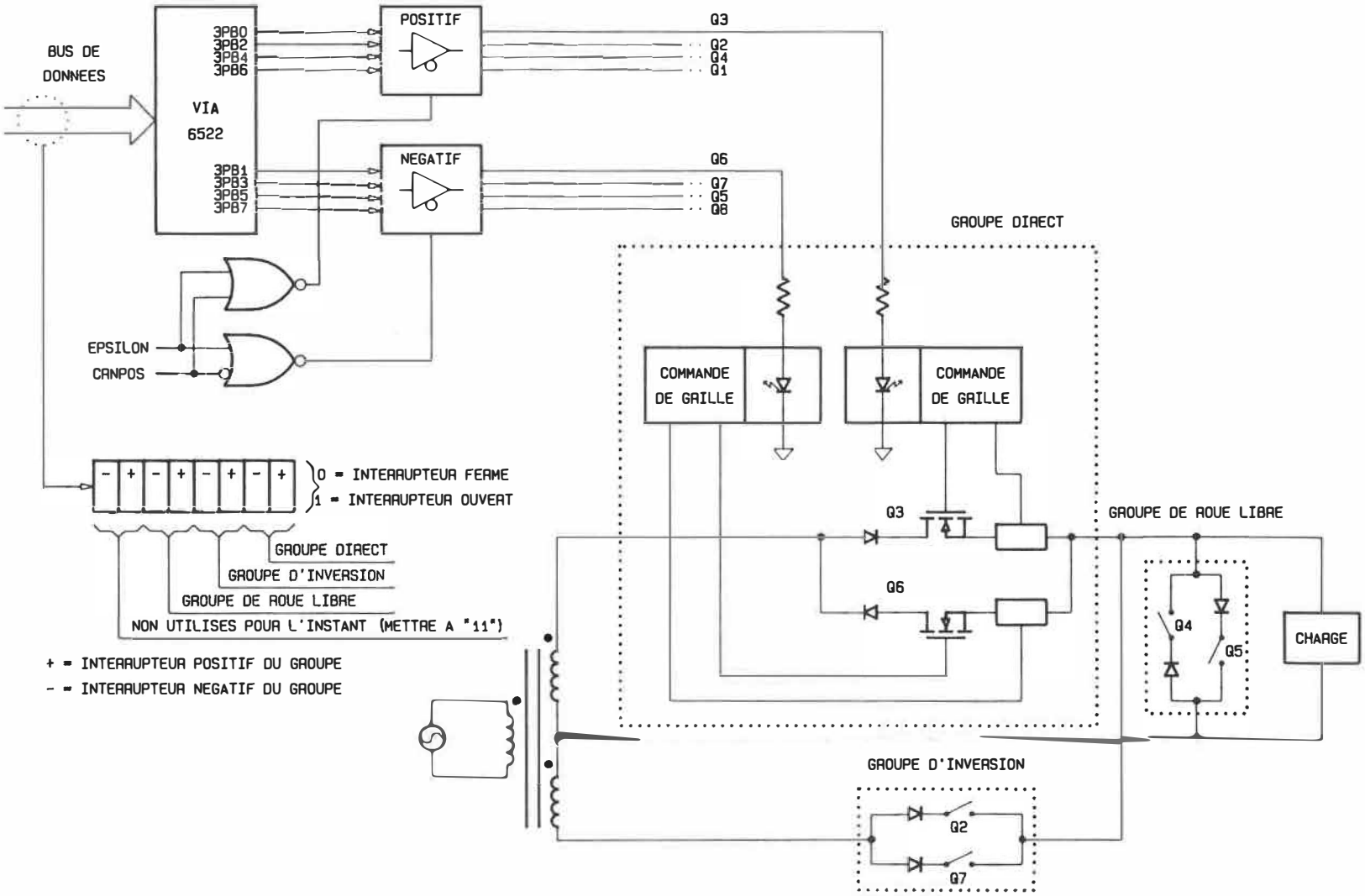


fig 1.3.3.1-2 COMMANDE DES INTERRUPTEURS

communication parallèle 8 bits. Sans entrer en détails dans les nombreuses possibilités des VIA 6522, disons que nous utilisons ces ports d'entrée/sortie et ces lignes de commandes pour réaliser le lien de communication bidirectionnel 8 bits et pour recevoir ou émettre de l'information. C'est-à-dire: la commande et le feedback des interrupteurs, la commande de la chaîne de mesure, le lien de communication. La figure 1.3.3.2-1 illustre bien l'interfaçage des signaux.

1.3.3.3 Module microprocesseur et module matériel divers.

Le Module microprocesseur est constitué du microordinateur SYM dont le microprocesseur est un 6502. Le SYM (voir annexe A) contient entre autre un clavier et une sortie qu'on peut relier à un terminal qui nous serviront à entrer des données et à voir les sorties. De plus, le SYM contient les VIA nécessaire pour le module d'interface. L'éditeur-assembleur et l'interface cassette du SYM ne servent plus car la longueur des programmes le rendaient d'utilisation très laborieuse. On utilise le système ORION pour le développement et le moniteur du SYM par la suite.

1.3.4 Les procédures.

Dans cette section, nous verrons les principes de base des modules importants du diagramme hiérarchique et leurs procédure. En particulier, nous regarderons les divisions logicielles suivantes:

-l'exécution des événements et la synchronisation,

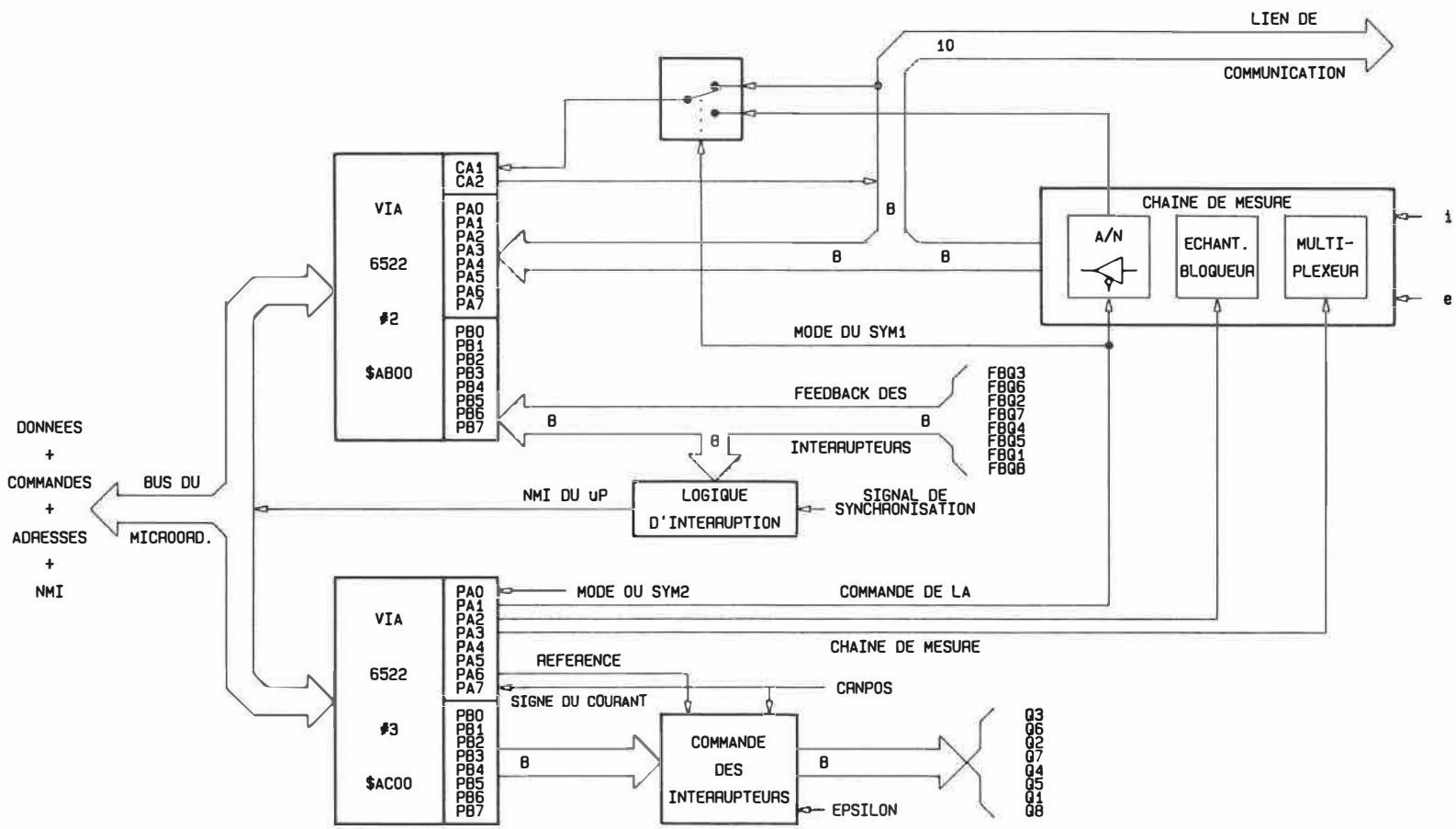


fig 1.3.3.2-1 LE MODULE D'INTERFACE DU MICROPROCESSEUR

-la façon de progresser vers les consignes,
-la communication entre les microprocesseurs,
-le module principal commandant le convertisseur.

1.3.4.1 L'exécution des événements et la synchronisation.

Cette division logicielle regroupe les modules de synchronisation et d'exécution des événements. Ces modules s'exécutent en temps réel à l'aide des sous-programmes d'interruption permis avec le microprocesseur 6502 utilisé par les microordinateurs SYM, c'est-à-dire un sous-programme appelé par interruption masquable (IRQ) et un autre par interruption non masquable (NMI). Comme nous l'avons vu un peu plus tôt, les horloges internes tirées des VIA des 6522 des SYM génèrent des interruptions qui permettent l'exécution du module s'occupant de l'exécution des événements ainsi que de la synchronisation de l'horloge du cycle déphasé. Ces interruptions sont moins prioritaires que celles qui originent des défauts d'interrupteur ainsi que du signal de synchronisation au réseau. C'est pourquoi ces différents signaux externes agiront sur l'interruption non masquable.

Nous avons séparé cette section de façon à bien expliquer le fonctionnement des horloges internes en fonction de leur rôle et ainsi mieux comprendre les choix faits pour la procédure des deux modules d'intérêts. Comme nous le verrons plus loin, ces choix ont surtout été influencés par les contraintes de temps d'exécution et par l'horloge du cycle déphasé.

Nous verrons qu'elle est la précision du réglage du déphasage ainsi que le nombre d'événements qu'on peut exécuter par cycle. Pour ce faire, nous élaborerons d'abord

sur les principes du fonctionnement de l'exécution des événements puis nous verrons la procédure des sous-programmes réalisant la partie "exécution" du logiciel.

1.3.4.1.1 Les horloges.

Rappelons que nous avons besoin de quatre horloges internes et un compteur (pour mesurer la durée d'un cycle du réseau):

-horloge de référence,
-horloge de référence déphasée de 180°,
-horloge du cycle dephasé,
-horloge d'événement,
-compteur pour mesurer la période,

On aura besoin de 5 horloges internes. Les trois VIA du SYM nous en fournissent six, dont cinq peuvent être utilisées (l'autre sert au nettoyage du clavier et autres trucs du genre). Les deux horloges de référence et celle du cycle déphasé utilisent des "timer 1" car on peut les faire fonctionner dans un mode intéressant afin de réaliser les P.L.L. numériques: on peut les utiliser en mode auto-oscillant avec la période (d'oscillation) rafraîchie à chaque fin de cycle par le contenu de registre (latches), eux-mêmes ayant leur valeur rafraichie lorsqu'on veut modifier la période. En tout temps, on peut lire le contenu des compteurs de ces horloges et savoir où on est dans le cycle. La fin d'un cycle est signalée par la levée de fanion et d'une interruption, si elle est permise. L'horloge d'événement fonctionne en décompteur: on la charge du temps d'attente avant le prochain événement; à la fin du décompte, l'horloge génère une interruption demandant l'exécution de l'événement prévu. Le compteur de

période est lancé au début d'un cycle (signalé par une NMI provoquée par le signal de synchronisation du D.P.Z. de tension) lu au début du suivant puis relancé. La durée du cycle est alors connue. On la valide . Si elle est valable, on l'utilise pour réajuster la période. Cette valeur de la période est utilisée pour les horloges de référence et du cycle déphasé.

On va maintenant parler plus en détails du réglage de l'horloge du cycle déphasé puis de ce qu'on entend par "P.L.L. numérique".

1.3.4.1.1.1 P.L.L. numérique.

La période est mesurée dans la partie "exécution". Elle se fait valider à un moment opportun dans le programme de "gestion/préparation". Si la valeur est acceptée, on vient modifier la valeur actuelle par un gain qui multiplie la différence entre la valeur actuelle et celle mesurée. On additionne cette valeur à la valeur actuelle et on obtient la nouvelle valeur pour la période. Cette nouvelle valeur sera utilisée par les horloges internes au début du cycle suivant après les calculs. De façon schématique, on l'illustre à la figure 1.3.4.1.1.1-1a.

Le bloqueur est constitué par les "latches". La période de rafraîchissement dépendra beaucoup de la durée du programme de gestion et du temps qu'il a pour s'exécuter dans un cycle; donc du temps occupé par la partie exécution et du temps requis pour boucler dans le programme de gestion. Supposons que la période de rafraîchissement est de quelques cycles, cela veut dire que le gain doit être faible car la correction se fait avec une grande période de rafraîchissement. Un grand gain entraînerait de grandes variations qui seraient corrigées plusieurs cycles plus

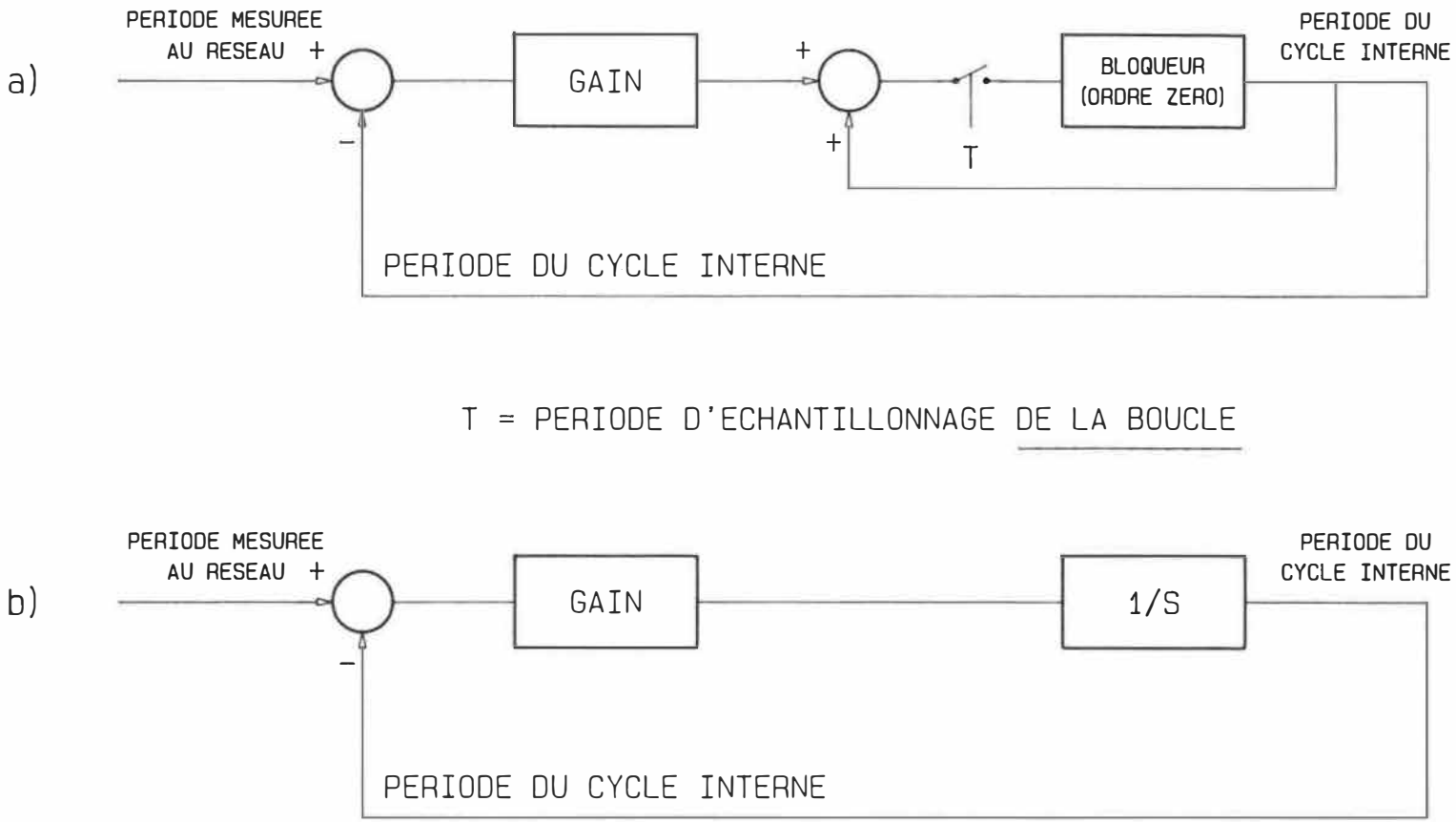


fig 1.3.4.1.1.1-1 LE MODELE DE LA P.L.L. NUMERIQUE

a) MODELE ECHANTILLONNE

b) MODELE EQUIVALENT A BASSE FREQUENCE

tard; une instabilité de la période en résulterait. Pour des durées beaucoup plus grandes que cette période de rafraîchissement, on a le système de la figure 1.3.4.1.1.1-1b. On obtient donc une période moyenne. Période moyenne dont la valeur s'approche de celle d'un cycle moyen. Si on perdait la synchronisation au réseau, les horloges continueraient avec cette valeur, comme le ferait une boucle à verrouillage de phase perdant temporairement son signal d'entrée. C'est pourquoi on appelle cela une P.L.L. numérique par analogie à l'action d'une "vraie" P.L.L..

1.3.4.1.1.2 L'horloge du cycle déphasé.

Permet d'avancer le cycle déphasé par rapport au cycle interne (<=75°). Elle fonctionne en mode auto-oscillant avec une période ajustée de façon à maintenir un certain déphasage par rapport à l'horloge du cycle interne. L'horloge du cycle est un genre de P.L.L.. Si la synchronisation au réseau disparaît, elle continuera à osciller avec la période moyenne d'un cycle. Elle verrouillera aussi l'horloge du cycle déphasé à son déphasage actuel en lui imposant la même période de cycle qu'elle utilise. A ce moment là, on ne peut plus régler le déphasage; on doit attendre que le signal de synchronisation revienne.

Etudions la façon de verrouiller l'horloge déphasée au réseau tout en commandant sa phase. Le schéma de la figure 1.3.4.1.1.2-1 nous fera comprendre le principe de base:

La correction se fait en plusieurs étapes:

1) On mesure l'erreur entre la valeur à l'horloge qu'on devrait avoir avec un certain déphasage au début d'un cycle réseau et celle qu'on a lue à ce moment. Cette

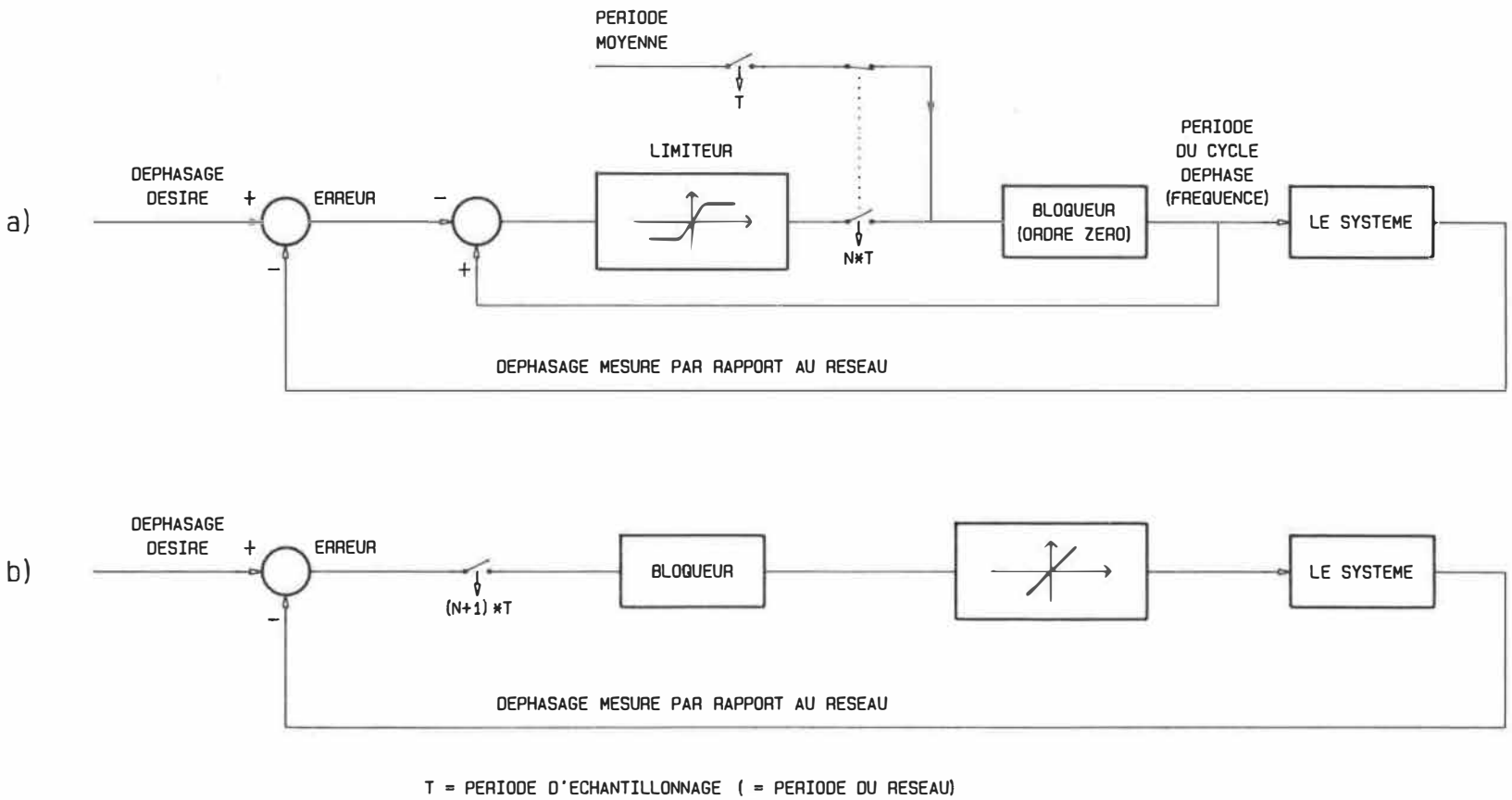


fig 1.3.4.1.1.2-1 REGLAGE DE L'HORLOGE DEPHASEE

a) MODELE DISCRET REEL

b) MODELE DISCRET SIMPLIFIE

mesure de "l'erreur" se fait dans le module synchronisation; c'est-à-dire au début d'un cycle.

2) 180° plus tard, au début d'un cycle de 180°:
On calcule la nouvelle période pour l'horloge déphasée. Cette valeur est ensuite portée dans les latches de l'horloge. On abaisse un drapeau (ETAP3) signalant qu'on veut un certain temps de repos avant qu'on ne refasse une correction. Pendant ce temps de repos (ETAP3 est bas), on utitise la période moyenne comme valeur pour l'horloge du cycle déphasé.

3) Au début du cycle déphasé, l'horloge charge ses compteurs de la période contenue dans ses latches. Si ETAP3 est bas, c'est-à-dire qu'on est dans un temps de repos, on incrémente le compteur de cycles de repos. Si sa valeur atteint une certaine limite ("3" actuellement) on lève le drapeau "ETAP3", signalant la fin du temps de repos.

Le "temps de repos" permet, en fait, de régler la période d'échantillonnage. On fait une correction de fréquence à un cycle, puis on revient à la fréquence "moyenne" pour un certain nombre de cycle (temps de repos). On recommence s'il y a lieu. En fait, c'est une correction de phase qu'on fait à tous les "N" cycles (N= nombre de cycle de repos +1): la variation de fréquence à un cycle, permet de déplacer le cycle déphasé par rapport au cyle interne (du réseau). Le retour à la période normale fixe le cycle déphasé à celui du réseau, maintenant ainsi la différence de phase obtenue lors de la variation de fréquence. C'est comme pour une P.L.L. avec un oscillateur à fréquence commandée (VCO). Le système équivalent est celui de la figure 1.3.4.1.1.2-1b.

Dans les sections qui suivent, nous verrons quelle est la précision du réglage du déphasage ainsi que certain

problèmes reliés aux horloges internes qui affectent les procédures des modules qui ont des exigences sévères reliés à leur exécution en temps réel.

1.3.4.1.1.3 La précision du réglage.

Elle dépend de la durée du "temps de repos" et de la précision de la période moyenne par rapport à la période réelle. En effet, la différence entre la période moyenne et l'actuelle entraîne une dérive de l'horloge déphasée par rapport au réseau. Plus le "temps mort" est long, plus cette dérive devient significative.

On doit aussi considérer le gain utilisé lors de la correction: petit, on aurait besoin de plusieurs actions correctives consécutives pour atteindre le bon déphasage. Mais, la dérive lors du temps de repos intervient aussi: avec un faible gain, la correction peut être "perdue" par dérive lors du temps de repos. De façon expérimentale, on a remarqué une différence de plus ou moins 6 micro-secondes entre la période moyenne et celle du réseau. Si on considère que la dérive ne devrait pas dépasser un demi degré entre deux ajustements de la période de l'horloge déphasée, on ne devrait donc pas attendre plus de quatre cycles entre deux corrections. Cela requiert donc un temps de repos maximal de trois cycles. De plus, le limiteur agissant sur la période devra être plus grand que la valeur nécessaire à la correction de la dérive. Le limiteur devra permettre une variation supérieure à plus ou moins 24 micro-secondes. Nous avons fixé cette valeur à environ trois fois cette valeur. On pourra alors corriger effectivement de 1° par "cycle de correction-période d'échantillonnage" c'est-à-dire un quart de degré par cycle

réseau.

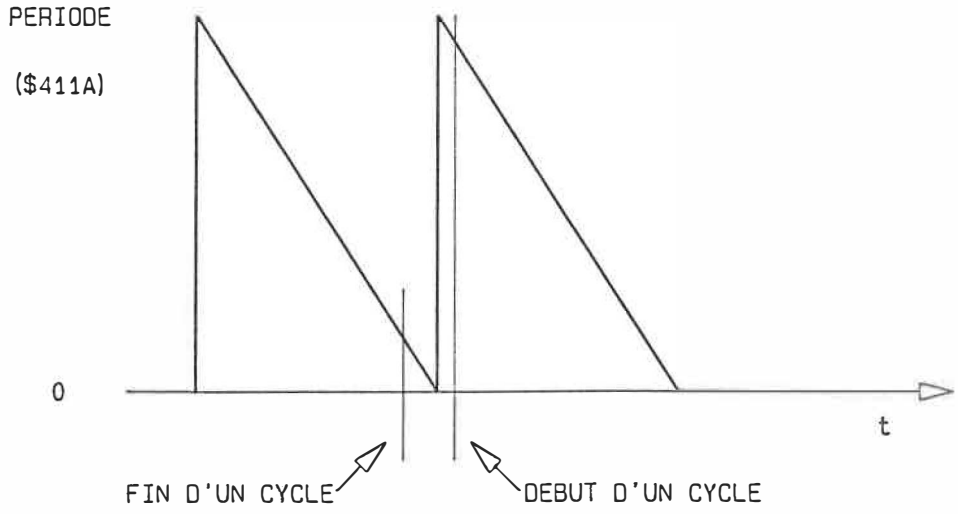
On a regardé le cas où la "dérive" nous éloigne de l'atteinte de la consigne. Les autres cas ne sont pas problématiques car ils aident ou facilitent le réglage.

1.3.4.1.1.4 Les problèmes reliés à la discontinuité dans les horloges.

Les horloges fonctionnent en décompteur. Une fois le décompte atteint, (on est à "zéro"), automatiquement le décompteur est rechargé à partir de la valeur de départ. De plus, à cet instant, des fanions indiquent la fin du cycle et génèrent une interruption si on la permet. La valeur de départ du décompte c'est la période. On peut illustrer le fonctionnement de l'horloge comme à la figure 1.3.4.1.1.4-1. On notera que les valeurs numériques données sont théoriques; en pratique la période variera.

On y note la discontinuité numérique entre la fin du cycle précédent au début du cycle suivant: on passe brusquement de $0+\varepsilon$ à $\$411A-\varepsilon$. Cela complique un peu les calculs qui doivent permettre d'identifier les conséquences "numériques"des calculs.

En ce qui concerne l'horloge déphasée, la région autour de "0" est critique. Pour permettre l'identification certaine de la position d'une horloge par rapport à une autre, on doit alors limiter la plage effective du déphasage permis afin d'assurer qu'on règle correctement. On voit qu'un traitement, plus long qu'une simple addition ou soustraction, s'en suit. Ce traitement doit être localisé "temporellement" là où ça ne dérange pas, donc bien avant une prise de décision qui en dépend. C'est pourquoi on retrouve l'étalement en trois temps du réglage



DISCONTINUITE DES HORLOGES DE CYCLE

fig 1.3.4.1.1.4-1

de l'horloge déphasée.

Dans la partie "exécution", pour le calcul du temps d'attente avant le prochain événement, la discontinuité des horloges cause aussi certains problèmes. Entre autres lorsqu'on se trouve à calculer le temps d'attente entre le dernier événement d'un cycle et le premier du suivant; le dernier événement est près de "0", le premier du cycle suivant est près de $411A (valeur de la période). Un sous-programme vérifiera que le délai calculé est bien dans les plages normales ou dans les plages "fantômes" crées par la discontinuité et veillera à compenser s'il y a lieu.

1.3.4.1.2 L'exécution des événements.

Comme nous l'avons vu plus tôt, la commande des événements se fait par le choix de l'état d'un interrupteur bipolaire. On fonctionne par groupe d'interrupteurs. Le convertisseur en a trois: le groupe direct, le groupe d'inversion et celui de roue libre. En tout temps, le programme de commande détermine le groupe qu'on devrait normalement avoir de façon à respecter les alternances positives et négatives: lorsqu'on déphase le cycle, on doit sélectionner entre le groupe direct ou d'inversion. Si on est dans la partie en avance, et qu'on doit avoir l'alternance positive, on devra sélectionner le groupe d'inversion jusqu'à ce qu'on atteigne le début du nouveau cycle réseau; à ce moment-là, on doit utiliser le groupe direct de façon à respecter l'alternance positive dans le cycle déphasé (avancé).

La commande que le programme lit, doit définir le groupe qu'on veut utiliser en le spécifiant par rapport au groupe "normal": roue libre ou non, groupe normal ou non.

Une stratégie est constituée d'une séquence de commandes

(ou d'opérations) qui imposent chacune un état donné aux interrupteurs du convertisseur pendant un certain temps. Deux tables importantes à considérer pour l'exécution des événements: celle contenant la séquence des commandes (TBSEQ) et celle contenant la durée de chacune des commandes de la stratégie. Il y a évidemment toute une série de paramètres constituant une partie de la définition des stratégies qui permettent de définir les tables, les sélectionner et les gérer correctement. Mais cela est de peu d'intérêt pour la compréhension de l'exécution des commandes d'une stratégie; cela a été vu de façon général dans la section 1.3.1 donnant la structure des données.

En se référant à la figure 1.3.4.1.2-1, on peut voir l'organisation générale des deux tables. De plus, on y montre les détails du format d'une commande de la table donnant la séquence des événements. Ce format peut sembler farfelu mais il nous permet d'accélérer l'exécution du traitement car l'information est contenue dans les bits 7 et 0 de la commande et que ces bits font l'objet d'une attention particulière lors de l'exécution de plusieurs instructions du microprocesseur 6502.

Les détails nécessaires à la spécification du type d'opération dans une commande sont donnés sur la figure 1.3.4.1.2-1. Comme on le voit, ces commandes ne constituent pas directement l'information de l'état désiré de chacun des interrupteurs (pas possible avec le déphasage et le respect de l'alternance) mais bien l'opération à faire au convertisseur. C'est lors de l'exécution qu'on donne, en fonction de la commande, l'état désiré des interrupteurs des trois groupes du convertisseur. Ces états seront modifiés si, à un moment ou à un autre, le programme détermine que le groupe "normal" change, de façon à respecter les alternances. Nous verrons cela un peu plus

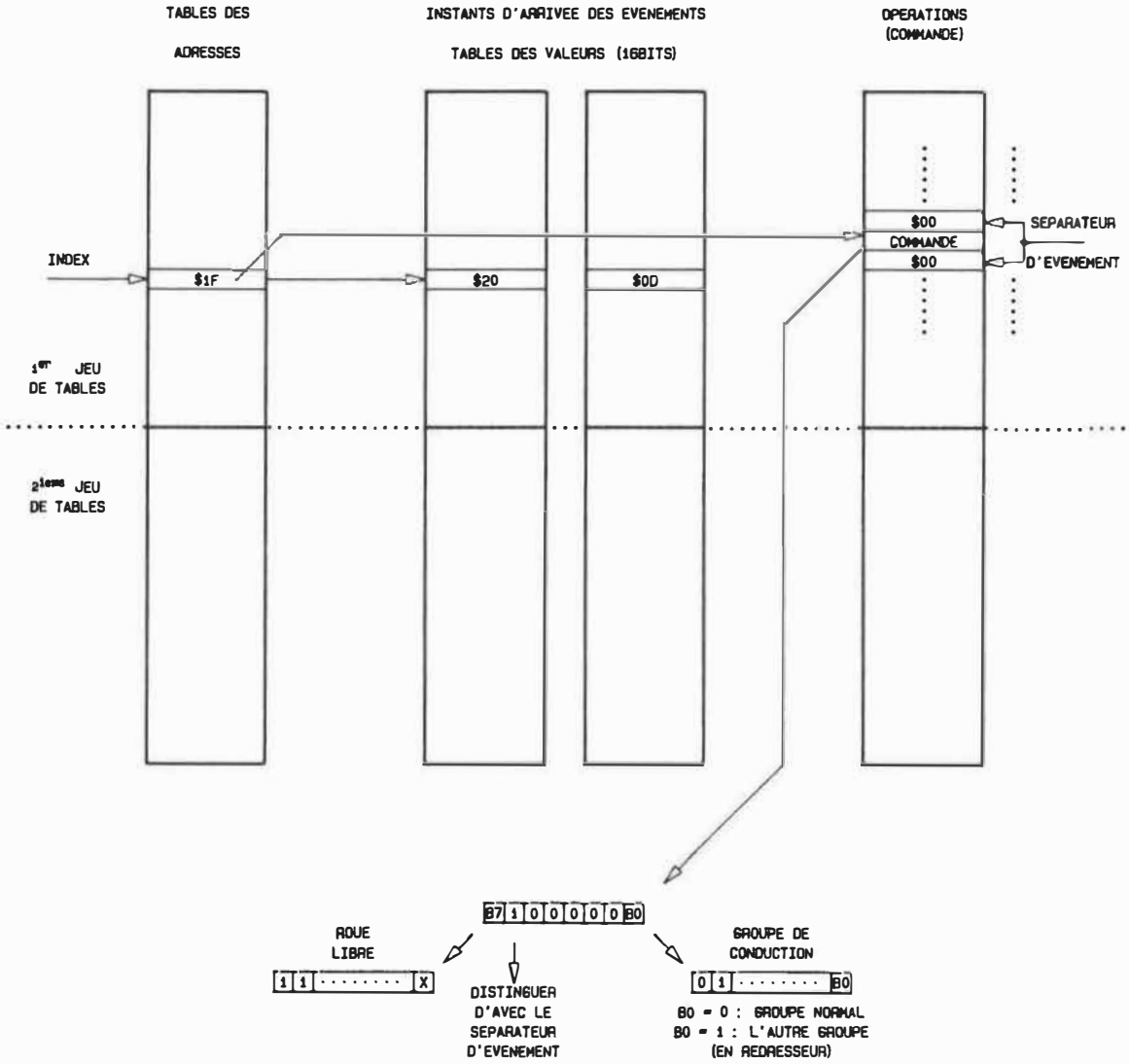


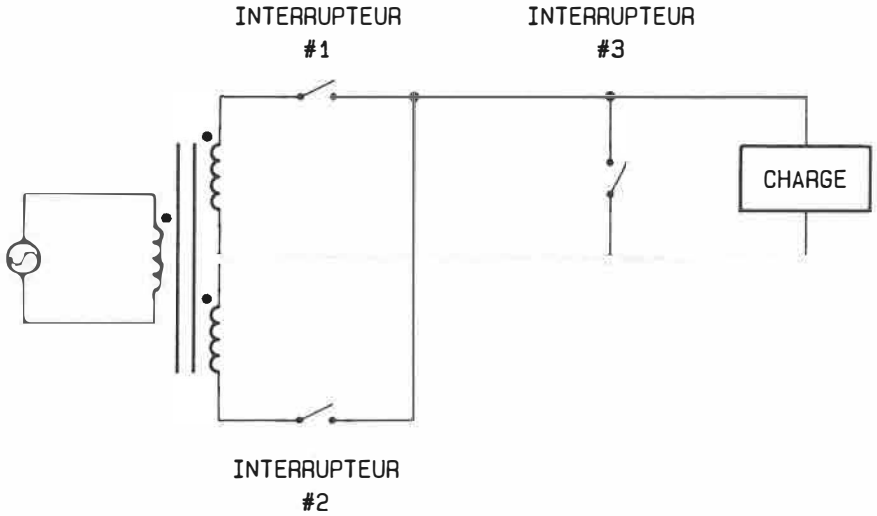
fig 1.3.4.1.2-1 LES TABLES ET LE FORMAT DE LA COMMANDE

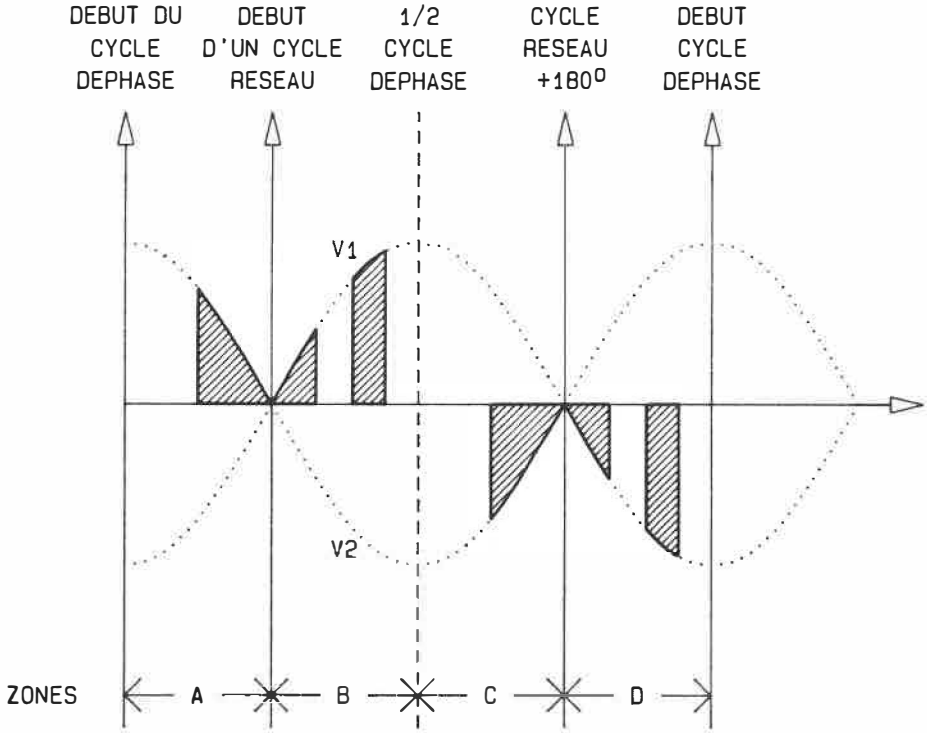
loin lorsque nous parlerons du séquencement des événements dans le cycle déphasé et à l'aide des pseudocodes schématiques. Pour l'instant, voyons à l'aide de la figure 1.3.3.1-2 comment on commande l'état des interrupteurs du convertisseur.

Le format de la commande des interrupteurs est donné à la figure 1.3.3.1-2. On utilise deux bits par groupe d'interrupteurs: un pour l'interrupteur portant le courant positif et l'autre pour celui portant le courant négatif. On notera que le dernier groupe ("4") n'est pas utilisé; on laisse en tout temps les interrupteurs ouverts (B7=B6="1"). Ainsi :BØ et B1 servent à commander le groupe direct, B2 et B3 le groupe d'inversion et B4 et B5 le groupe de roue libre. Dans notre cas, on commande par groupe (ce n'est pas une obligation physique), c'est-à-dire que le programme ferme les deux interrupteurs d'un groupe à la fois: c'est le matériel, selon le sens du courant, qui s'occupe de permettre à un ou à l'autre des innterrupteurs de se fermer. On se facilite ainsi la tâche: pas besoin de vérifier logiciellement le signe du courant à chaque fois (on ne peut pas prédire en tout temps le signe du courant à cause du déphasage qu'on peut avoir par rapport au réseau et selon la stratégie d'opération des interrupteurs utilisée).

Le programme s'assure de façon logicielle lors du passage d'un groupe de conduction à un autre d'éviter les courts-circuits en passant par l'état de roue libre avec l'interrupteur sélectionné selon le signe du courant à la charge (autrement on créerait un court-circuit aussi!).

Voyons maintenant le découpage par zones du cycle. Sur la figure 1.3.4.1.2-2, on illustre l'enveloppe dans laquelle on découpe afin d'obtenir la sortie du convertisseur. Cette enveloppe est formée par les deux sources de tension





V1 : TENSION DIRECTE

V2 : TENSION D'INVERSION

FIG 1.3.4.1.2-2 LES QUATRE ZONES DU CONVERTISSEUR

déphasée utilisées au convertisseur; les deux sources sont déphasées de 180°. On y voit aussi leur identification par rapport aux groupes direct et d'inversion; la source de référence est branchée sur le groupe direct et est appelée "V1". L'autre est branchée sur le groupe d'inversion et est appelée "V2". On y a aussi inscrit les débuts de cycle déphasé interne (verrouillé sur la tension de référence V1) et déphasé de 180° (180° par rapport à la tension V1). On illustre une stratégie à huit événements.

Dans la zone "A", on doit normalement utiliser V2 pour obtenir une alternance positive; c'est-à-dire qu'on doit utiliser le groupe d'inversion. Dans la zone "B", on utilise la tension V1 et donc le groupe direct et ce jusqu'à la fin de l'alternance positive du déphasage. La zone "C" débute avec l'alternance négative du cycle déphasé et se termine à la fin de l'alternance positive du cycle de référence V1; le groupe utilisé normalement sera le groupe d'inversion car l'alternance négative du cycle de référence n'est pas encore commencée et qu'on veut une tension négative. La zone "D" débute avec l'alternance négative du cycle de référence et se termine à la fin du cycle déphasé; comme on est sur l'alternance négative du cycle de référence, on utilisera V1, le groupe direct pour compléter l'alternance négative du cycle déphasé. Ces zones sont crées par le déphasage. On peut résumer de la façon suivante:

zone:	groupe normal à utiliser
A	2
B	1
C	2
D	1

On détermine la zone et le groupe normal à l'aide des horloges déphasées, du cycle, du cycle à 180° et de la

symétrie entre les deux alternances dans les stratégies utilisées.

On doit "rafraîchir" le groupe actif à chaque interruption provoquée par une des horloges afin de s'assurer de toujours respecter et la commande et le groupe normal dans la zone. Par exemple, dans le cas du premier événement: il a lancé une conduction sur le groupe d'inversion qui doit être changé au début du cycle de référence afin d'avoir une tension de sortie positive (on est sur l'alternance positive du cycle déphasé). Ainsi, à chaque début d'un des trois cycles internes, on vérifie si le groupe actif est bien celui qu'on doit avoir. En ce qui concerne le demi cycle de l'horloge déphasé, on ne le signale pas comme tel mais on utilise la principale symétrie entre les alternances afin de s'assurer de l'exécution correcte dans la zone "C".

On regroupe les opérations ultra prioritaires, c'est-à-dire reliées à la synchronisation au réseau et au traitement des défauts aux interrupteurs du convertisseur, dans le sous-programme RTNNMI. De plus, on y fait les opérations nécessaires afin de veiller au bon fonctionnement du convertisseur au début d'un cycle réseau: la signalisation et la sélection du groupe normal et la modification du groupe actif s'il y a lieu. Ce sous-programme constitue le module "synchronisation" dans le diagramme hiérarchique.

Dans le sous-programme RTNIRQ, on s'occupe de l'exécution des événements et du réglage de l'horloge déphasée. Il y a quatre sources d'interruptions pouvant provoquer l'exécution de ce sous-programme: l'horloge interne du cycle et celle de 180° ainsi que l'horloge déphasée et finalement l'horloge d'événement lançant l'exécution au bon moment des événements.

Pour plus de détails sur les procédure, on peut consulter le cahier annexe [3] qui contient, entre autres, le pseudocode schématique et le code des procédures. On y remarquerait que les procédures sont quand même assez longues. Malgré toutes les simplifications et un choix éclairé pour le traitement et la structure des données en fonction d'une minimisation du temps d'exécution des interruptions, on doit respecter certaines contraintes qui dépendent directement de la lenteur du microprocesseur.

1.3.4.2 La progression vers les consignes.

Nous allons en premier lieu voir le principe du réglage de la tension. En regardant le diagramme hiérarchique, on note plusieurs modules reliés à cette régulation:

- -REGTEN, le module servant à réguler la tension ,
- -MESTEN, le module servant à mesurer la sortie du convertisseur,
- -MODTEN, le module servant à modifier les instants d'arrivée des événements de façon à augmenter ou diminuer la tension de sortie selon le cas.

En plus de ces trois modules directement reliés à la boucle de régulation de la tension, il y a les modules servant à générer les tables (dans le jeu de table en attente) en fonction de la commande de tension demandée (par la boucle de régulation) et celui servant lors des changements de stratégie et au tout début. Nous les regarderons sommairement après avoir expliqué la régulation. Le schéma de la figure 1.3.4.2-1, nous montre le système bouclé réalisé par le module REGTEN.

De façon cyclique, le module réalisera la boucle jusqu'à

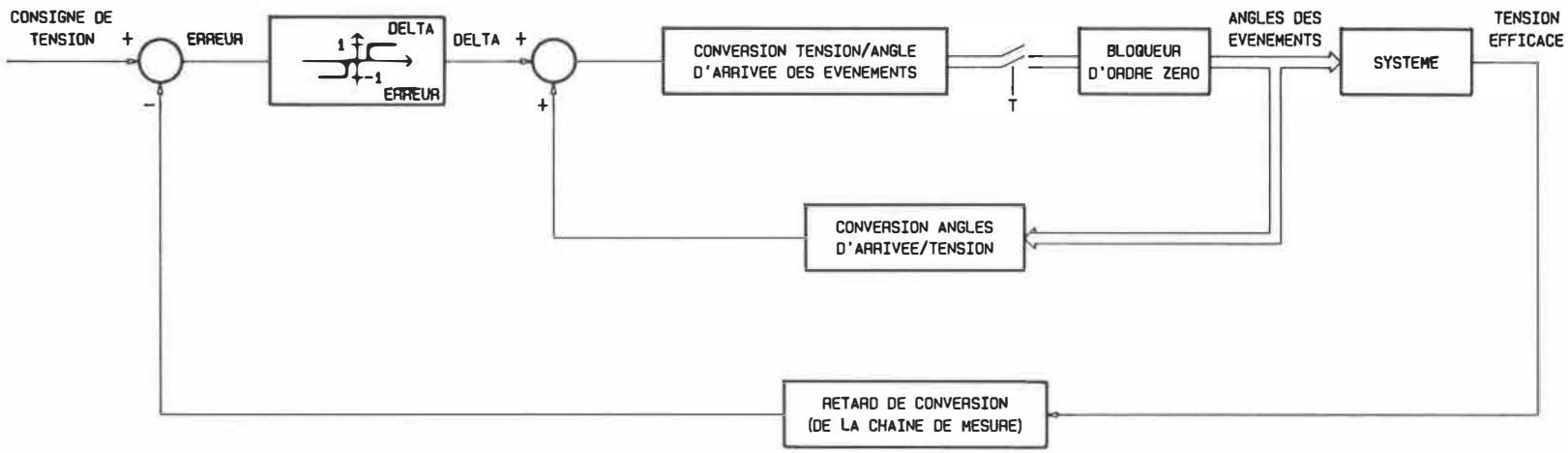


fig 1.3.4.2-1 BOUCLE DE REGLAGE DE LA TENSION DE SORTIE DU CONVERTISSEUR

ce que l'erreur mesurée soit contenue dans la bande morte du régulateur à trois positions. A ce moment là, le module considère la consigne atteinte et retourne au module principal. Ce dernier exécute alors ses autres fonctions puis, avec les nouvelles consignes, il retournera au programme de régulation de tension. Encore une fois, on s'aperçoit que la boucle n'est pas celle d'un système linéaire avec un grand gain pour diminuer l'erreur en régime permanent. Nous avons à prendre en considération que la mesure d'une valeur RMS nécessite plusieurs cycles pour être précise (la forme d'onde n'étant pas sinusoïdale, on ne peut pas mesurer la valeur moyenne et obtenir, à l'aide d'un facteur de conversion fixe la valeur efficace).

De plus, on doit modifier les tables des instants d'arrivée des événements à chaque fois qu'on veut modifier la tension de sortie. Encore là, on doit compter plusieurs cycles: il n'est pas dit qu'on ait le temps de modifier toutes les tables dans les périodes de temps laissées libres dans un cycle par la partie "exécution" et même si c'était le cas, on doit toujours attendre le nouveau cycle pour changer de jeu de tables. Donc, dans le meilleur des cas, on doit compter un cycle pour la modification des tables et son utilisation, et environ cinq cycles (pour notre convertisseur) afin de mesurer; donc six cycles comme période de rafraîchissement de la consigne au convertisseur. C'est très lent; on a un grand retard dans la chaîne de mesure. La période doit être plus grande que le retard et le gain faible. On utilisera plutôt un système bouclé avec une faible variation de la commande (un faible "pas") et ayant un effet intégrateur. En plus, il faut tenir compte des incertitudes sur la valeur mesurée; c'est pourquoi on utilise une bande morte dans le régulateur: l'erreur devra être au moins supérieure à deux fois

l'incertitude prévue pour le calcul de l'erreur. Ainsi, on ne restera pas prisonnier de la boucle.

Lorsqu'on utilise le module de transition, on doit d'abord avoir le déphasage à 0°. Après, il réinitialisera le vecteur des paramètres contenant l'information nécessaire pour accéder aux tables définissant la nouvelle stratégie, les valeurs maximales et minimales des instants d'arrivée des événements et le nombre d'événement. Puis, il imposera les valeurs donnant la tension minimale puis progressera en modifiant par 1° par cycle (au maximum) les événements jusqu'à ce qu'on atteigne les valeurs cibles pour obtenir la tension de consigne selon les tables. Après, on se sert du module REGTEN qui fonctionne en boucle fermée.

Les deux modules assurant l'atteinte de la consigne de tension, soit REGTEN et PRGSIF utilisent un module appelé "GENOPR" qui remplit le jeu de table en attente à partir des "cibles" qu'on lui donne pour chacun des événements: les cibles sont les angles d'arrivée de chacun des événements (on ne donne que les cibles pour la première alternance, la seconde étant symétrique, 180° plus loin). Pour REGTEN, c'est le module inférieur MODTEN qui fournit les cibles; PRGSIF le fait directement lui.

Le module PRGSF2 NOUS sert à atteindre la consigne de déphasage. Celui-ci se trouve a fournir la consigne de déphasage aux modules d'exécution et de synchronisation. Comme nous l'avons vu plus tôt, ces derniers réalisent, entre autres, le réglage de l'horloge déphasée; réglage fait en trois temps. Nous avions vu que l'on pouvait s'attendre à un taux de progression moyen du déphasage de l'ordre de 1/4° par cycle. Dans le module PRGSF2, on fournit la consigne. On variera lentement cette consigne de façon à être certain que l'horloge déphasée puisse

l'atteindre bien avant qu'elle ne soit de nouveau modifiée. Donc, le module PRGSF2 fournit la consigne à la boucle de réglage de l'horloge déphasée et, la modifie de nouveau, s'il y a lieu, bien après l'atteinte (d'un régime permanent) de la consigne. Pour ce faire, on devra attendre un certain nombre de cycles entre deux nouvelles consignes et modifier d'un faible incrément cette consigne. Nous avons choisi l'incrément à 1/4° environ et le temps d'attente à 10 cycles, soit plus de deux fois la période de rafraîchissement dans la boucle de réglage de l'horloge déphasée.

On sort du module une fois que la consigne fournie à la boucle de réglage est égale au déphasage demandé pour le convertisseur.

Rappelons que les sous-programmes d'interruption signalent les nouveaux cycles à l'aide de la levée d'un drapeau; ainsi la prochaine modification du drapeau nous indique un début de cycle. On se sert de cela pour attendre un certain nombre de cycles dans les programmes où on varie les consignes.

Le pseudocode des modules décrits ci-haut et ceux des modules inférieurs qui leur sont attachés (voir le diagramme hiérarchique) GENOPR, REGTEN, MESTEN, MODTEN, PRGSIF et PRGSF2 sont donnés en [3].

1.3.4.3 Programme de communication.

On doit recevoir les consignes d'un second microprocesseur. Nous avons un lien parallèle de 8 bits de données et 4 de signaux de commande pour le faire. Comme on peut le voir à la figure 1.3.3.2-1, ce lien de communication est multiplexé avec la sortie du CA/N; et ce

aux deux microprocesseurs. Seules deux lignes ne le sont pas: elles permettent de connaître le mode utilisé à chacun des microprocesseur (en communication ou en CA/N).

Nous avons décidé d'imposer un "maître" dans la communication; c'est lui qui demande la communication, qui transmet son état actuel puis demande au second microprocesseur de transmettre à son tour. En plus, les donnés reçues ne seront acceptées que si la communication a été jugée valide. Précisons que des caractères de synchronisation sont envoyés en début et en fin de transmission par chacun des microprocesseurs et que la durée de transmission d'une donnée est bornée par une horloge de surveillance. Si un des caractères de synchronisation n'est pas reçu au moment attendu, la communication est male cadrée; si l'horloge de surveillance nous informe que le temps d'attente est trop long, on suppose un problème dans le lien. Dans ces cas, on considère les informations et la communication invalides. Pour éviter de modifier les variables avec l'information reçue, sans être certain de la validité, on remplit plutôt une pile et lorsque la communication aura été jugée valide, on transfèrera les valeurs de la pile dans les variables.

La communication se passe en plusieurs étapes: il faut d'abord vérifier que le second microprocesseur peut communiquer. Pour ce faire, on vérifie le mode de définition de son bus et s'il n'est pas en attente de communication, on attend jusqu'à $808 micro-secondes puis on considère une communication invalide. S'il est prêt, on définit correctement les valeurs de départ des signaux de commande sur le lien puis on lance un appel au deuxième microprocesseur. Chaque fois qu'on attend une réponse, l'horloge de surveillance vérifie que le temps d'attente n'est pas trop long. S'il nous répond, on procède en

envoyant les valeurs, encadrées par les caractères de synchronisation et sous-surveillance de l'horloge. Si la transmission est réussie, on passe en mode réception pour attendre les données du deuxième microprocesseur sous surveillance de l'horloge. Les données reçues sont mises sur une pile. Si elles sont correctement encadrées, elles sont acceptées et transférées dans les variables correspondantes.

L'inconvénient majeur de cette méthode est lié à la difficulté d'initier la communication; une fois lancé, il est très rare d'obtenir une communication invalide. Mais les deux microprocesseurs n'étant pas synchronisés, et ayant souvent des mesures à faire, le taux de communication reste faible. Tout cela doit se faire dans une fenêtre de temps restreinte. On doit faire attention à la durée d'une communication car le microprocesseur est très occupé avec toutes ses opérations prioritaires lancées par interruption: la communication est souvent interrompue et l'horloge de surveillance, elle, continue de compter, ce qui provoque souvent l'invalidité de la communication. On notera que le deuxième microprocesseur ne retarde pas la communication lorsque son bus est en mode communication car un des signaux de handshake provoque une interruption appelant le programme frère à ce microprocesseur. Le programme principal relancera la communication tant que de nouvelles consignes n'auront pas été reçues. Il serait de loin préférable d'utiliser du matériel dédié pour réaliser la communication: un ACIA par exemple. Ainsi, même avec de petites fenêtres de temps, le taux de communication resterait très élevé. Son pseudocode schématique est donné en [3].

1.3.4.4 La commande du variateur.

Le module principal du logiciel s'assure de bien commander le convertisseur. il est constitué de trois "blocs" principaux: le premier servant à initialiser et lancer le programme correctement, le second étant la boucle principale (le coeur du programme) et le troisième servant à arrêter le tout. La boucle principale doit demander les consignes, ajuster le déphasage selon la consigne, puis la tension si la consigne ou le déphasage changé, ajuster la P.L.L. numérique, changer de stratégie si c'est requis. En parallèle, le volet "exécution" est toujours en action. On sort de la boucle lorsque c'est demandé par le deuxième microprocesseur ou s'il y a un défaut (drapeau d'arrêt levé ou de recouvrement levé).

Pour arrêter le tout, on passe en roue libre aux interrupteurs et on interdit les interruptions (lançant les événements). Finalement, on redonne le contrôle au moniteur du système. Pour plus de détails sur la procédure, voir [3].

1.3.5 Le système en pratique.

La figure 1.3.5-1, nous montre l'organisation physique du système complet. On remarque trois cartes d'interfaçage autour des deux microordinateurs du système: mesures, commandes des interrupteurs et pré-traitement. Comme les deux systèmes à microprocesseurs ont besoins de la mesure du courant et de la tension instantannée au moteur, on utilise alors la carte de pré-traitement qui reçoit tous les signaux des différents capteurs et réalise le conditionnement des signaux d'entrée; ainsi, on ne dédouble pas le matériel. Alors, une partie du matériel apparaissant

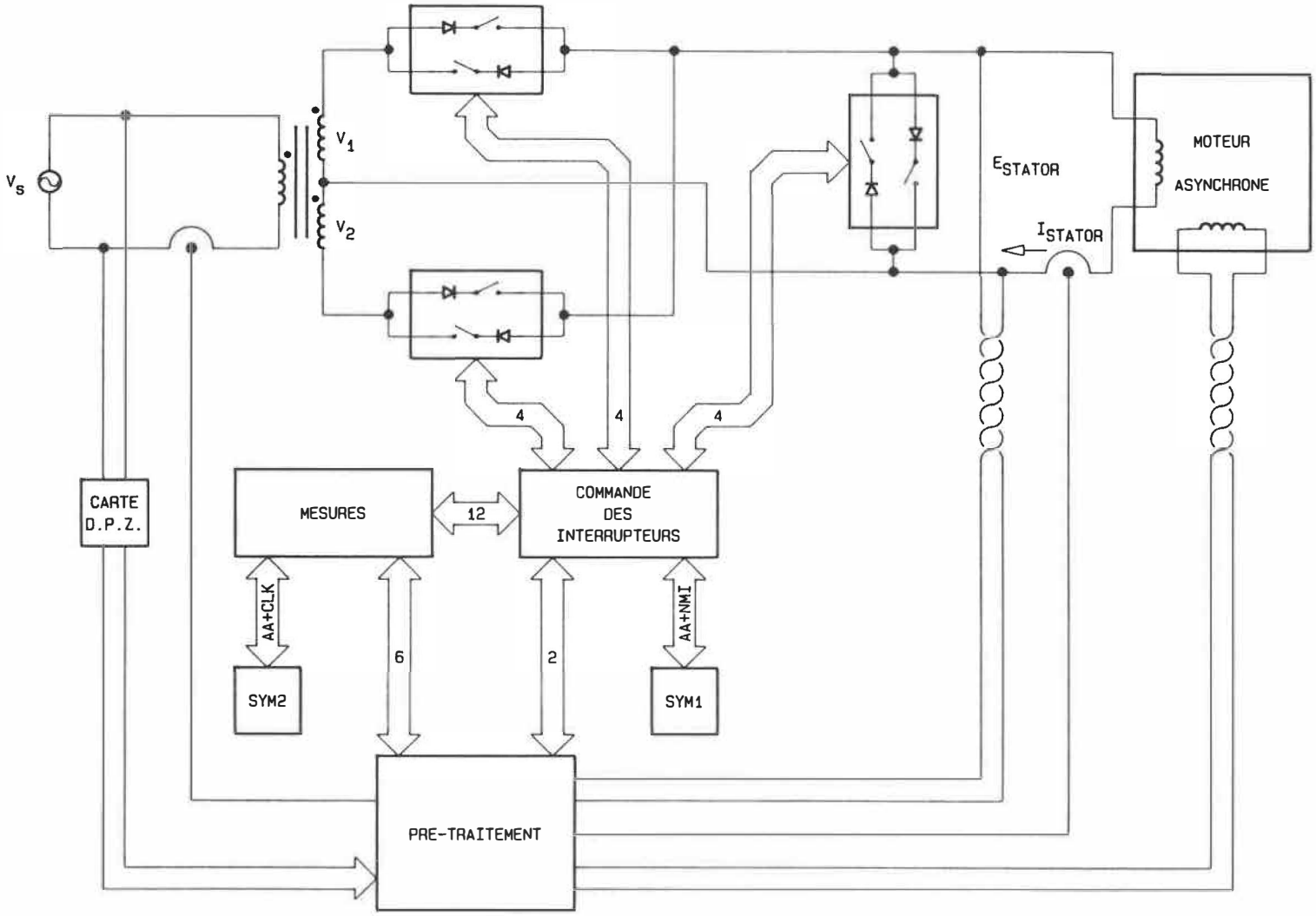


fig 1.3.5-1 ORGANISATION PHYSIQUE DES ELEMENTS DE L'ENTRAINEMENT

dans les schéma blocs vus précédemment est utilisée sur la carte de pré-traitement et est commune. les cartes de "commande des interrupteurs" et des interrupteurs sont propres au système de commande du convertisseur. Des renseignements détaillés sur la carte de commande des interrupteurs sont disponibles dans le mémoire de François Beauregard [1]. Quant au reste, il sera vu au chapitre trois qui traite de l'autre logiciel. Nous y donnerons quelques précisions sur les circuits moins communs ou sur le soin particulier a leur porter; le reste est facilement compréhensible en se servant des diagrammes blocs donnés.

CHAPITRE II

L'asservissement de l'entraînement.

Au premier chapitre, nous avons vu l'élément principal de l'entraînement: le convertisseur. Dans ce chapitre, nous analysons différentes possibilités de générer ses consignes de tension, de déphasage et de stratégie d'opération des interrupteurs.

Différentes possibilités d'asservissements sont dérivées d'une étude de la machine asynchrone. Deux classes d'asservissements y sont décrites: la régulation d'un paramètre et l'évolution d'optimisation.

Nous voyons aussi différentes stratégies d'opération des interrupteurs. Leurs impacts sur le moteur, le réseau et les interrupteurs sont mis en évidence. Les stratégies ont aussi un impact sur l'asservissement car elles influent sur les performances de la machine.

Les différents asservissements présentés sont mis en pratique à l'aide du logiciel de commande de l'entraînement décrit au troisième chapitre.

2.1 La commande du moteur asynchrone.

Le moteur asynchrone fonctionnant à tension nominale a des caractéristiques (rendement, impédance vue par sa source, etc) qui dépendent de la charge qu'il entraîne. Entre autres, le rendement et le facteur de puissance sont fonction du couple demandé. Pour ces deux paramètres, le "meilleur" point d'opération est situé près de la charge nominale prévue pour le moteur (ce qui est normal au point

de vue design de machine). Donc, en charge nominale, pas de problèmes. Mais le moteur ne fonctionne généralement pas constamment à sa charge nominale: pour la plupart des machines de faible puissance, il passe de façon cyclique de presque à vide à un état de charge presque nominale. Et la fréquence de ces cycles est faible car elle est liée à des opérations humaines et donc de l'ordre du hertz. Plusieurs auteurs [4,5,6,7] affirment, et nous l'avons vérifié à l'école [8,9], qu'on peut améliorer le rendement et le facteur de puissance d'une machine asynchrone à faible charge en variant l'amplitude de sa tension statorique. la figure 2.1-1 (à partir de [8], pp.51 et 76) nous montre le facteur de puissance et le rendement de la machine en fonction de la puissance de sortie pour les cas de:

- fonctionnement à tension statorique nominale (indice "1");
- fonctionnement à tension alternative ajustée pour maximiser le paramètre considéré (un paramètre seul est maximisé).

Ainsi, la différence entre les deux courbes d'un paramètre donné représente l'amélioration possible de ce dernier obtenu par un entraînement variant l'amplitude de la tension statorique pour le maximiser, à une condition de charge donnée. On note immédiatement que les gains relatifs sont plus grand à faible charge, là où les valeurs de puissance (charge) impliquées sont moins importantes. De là ressort l'importance d'un faible coût pour le système complet car les gains absolus ne sont pas nécessairement très grand. On notera aussi que la maximisation des différents paramètres illustrés, en fonction de la puissance de sortie, n'est pas obtenue pour une même amplitude de tension: lorsqu'on en maximise un, les autres ne le sont généralement pas. Donc, si on veut améliorer

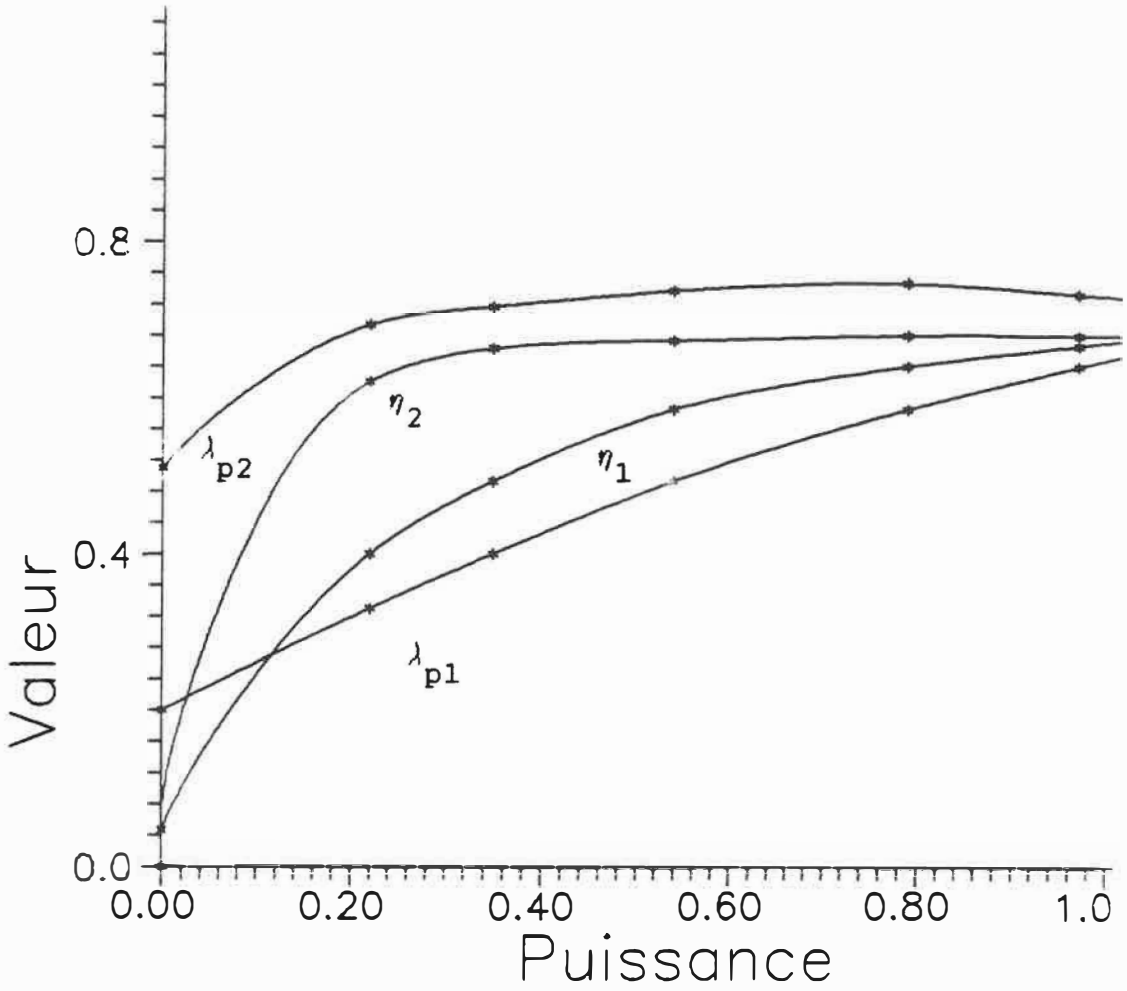


Figure 2.1-1 : Rendement (η) et facteur de puissance (λ_p) du moteur pour les cas de:

1) tension nominale: avec indice "1";
2) tension ajustée pour maximiser le paramètre: avec indice "2".

en fonction de la puissance de sortie normalisée.

l'ensemble des caractéristiques, on devra les combiner, de façon pondérées, dans une fonction globale exprimant notre **mesure de performance.**

Les diagrammes de phaseurs de la figure 2.1-2 nous éclairent sur le comportement de la machine en fonction de l'amplitude de la tension statorique. Ils montrent l'effet d'une variation de l'amplitude de la tension statorique sur les courants du circuit équivalent de la machine en régime permanent sinusoïdal. Du premier schéma de phaseurs, on note qu'en fonctionnant à tension nominale le courant de magnétisation (Im) est grand et, par conséquent, les pertes dans le fer sont élevées. Si on réduit cette tension, on réduit le courant Im et les pertes dans le fer. Une réduction excessive, telle que montrée sur le dernier schéma de phaseurs, entraînerait une très grande augmentation du courant statorique et, par conséquent, des pertes dans le cuivre. Il y aurait donc une tension qui nous donnerait le "meilleur" rendement ou autre critère de performance.

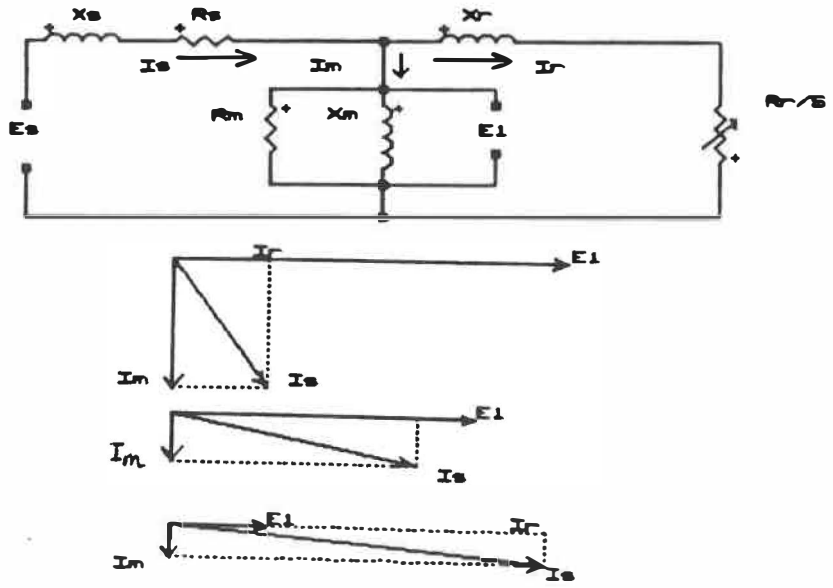


Figure 2.1-2 Effets d'une diminution de tension statorique sur les courants.

Comme on le voit à la figure 2.1-3, une diminution de la tension amène aussi une augmentation du glissement (charge constante). Il y aurait donc un glissement qui donnerait la meilleure performance (selon un critère donné) et obtenu par le réglage correct de la tension. Le glissement est le seul paramètre variable dans le circuit équivalent de la figure 2.1-2; il varie selon la charge du moteur. Le glissement détermine alors l'impédance que voit la source et les différents paramètres électriques de la machine (impédance d'entrée, facteur de puissance, rendement, etc) peuvent être vus comme des fonctions du glissement.

Donc, on peut améliorer des performances par la commande de l'amplitude de la tension d'alimentation du moteur en la réduisant lorsque le couple demandé peut-être atteint avec moins que le flux nominal de magnétisation; la grandeur de la réduction de tension est limitée par le couple demandé: une réduction trop grande amènerait un grand glissement, de grandes pertes dans le cuivre au stator et au rotor et peut-être même le décrochage du moteur.

On peut voir aussi que le moteur est une charge inductive, et qu'elle l'est de moins en moins selon l'importance du terme "Rr/s": plus le glissement est grand, plus le terme est important et plus le moteur présente une impédance résistive. Mais, même avec un glissement important (qui est plus grand que celui de décrochage à une charge et une tension données) l'impédance complexe est en partie inductive entraînant un retard dans le courant par rapport à la tension. On peut améliorer le facteur de puissance si:

- la tension de sortie du convertisseur est générée en avance de phase par rapport à celle de source,

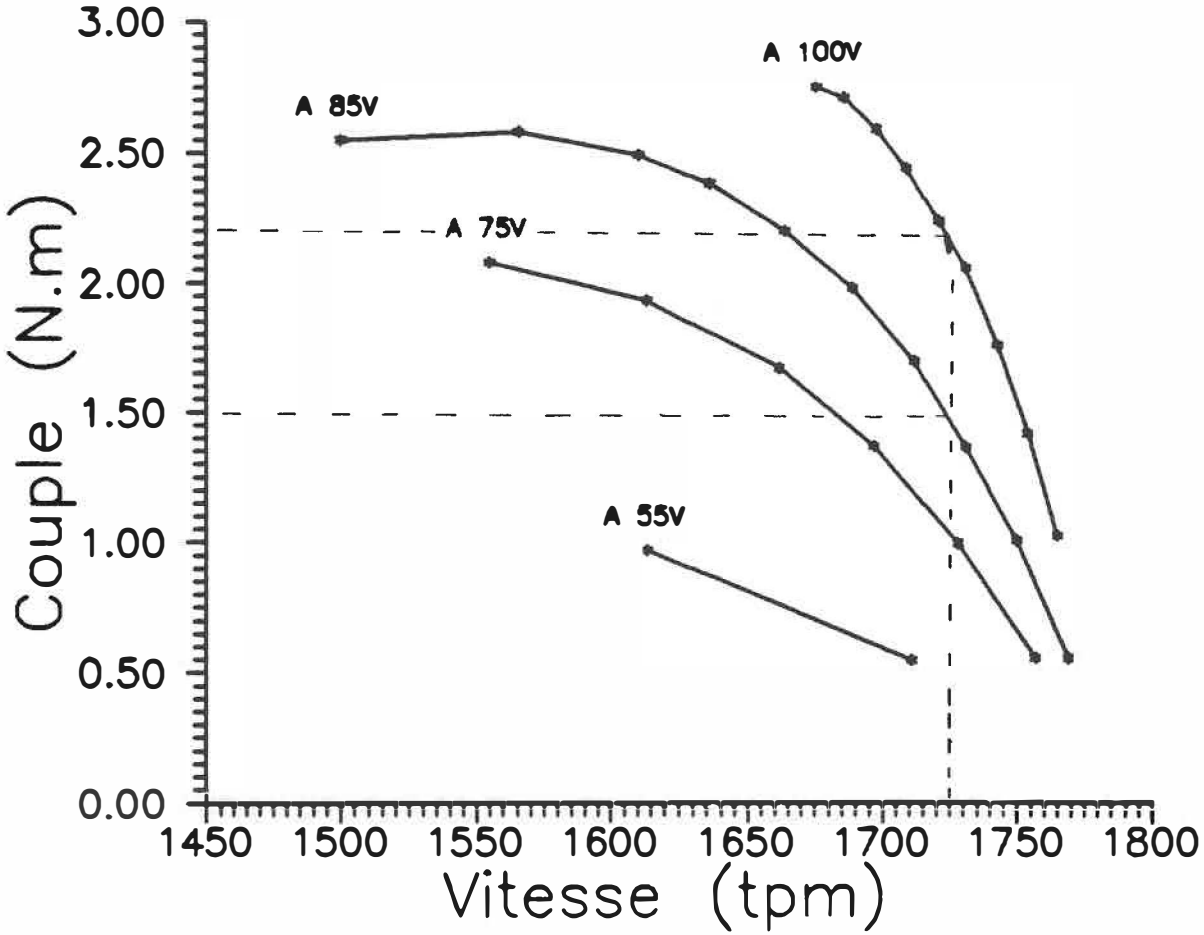


Figure 2.1-3 Couple en fonction de la vitesse pour differentes tensions statoriques.

permettant ainsi d'avancer en phase le courant du moteur et, par conséquent, de diminuer le retard de phase entre le courant et la tension de source. On peut aller jusqu'à obtenir un facteur de déplacement nul.

- l'impact des harmoniques de tension produites par le convertisseur reste faible; on a une faible distorsion harmonique.

Afin de fournir des consignes adéquates pour améliorer les performances du moteur, il nous faut déterminer une certaine loi de commande basée sur la mesure de valeurs à l'entraînement. Pour ce faire, nous devons, en premier lieu, identifier les buts de l'asservissement, les mesures permises pour établir la contre-réaction, ainsi qu'évaluer l'effet des contraintes sur la réalisabilité des algorithmes qu'on retrouve dans certaines approches d'asservissement.

- Le but de l'asservissement est d'améliorer les performances d'une machine asynchrone monophasée à un faible coût d'investissement plutôt que d'obtenir un réglage optimal à un coût plus élevé. De plus, l'asservissement ne doit pas requérir la connaissance précise du moteur (son modèle): on désire qu'il se greffe à toute machine asynchrone monophasée (d'une puissance maximale d'un hp) et améliore ses performances et ce qu'on en voit du réseau. Au point de vue de sa réalisation, le système a microprocesseur utilisé doit être très simple et peu coûteux; on utilise un SYM. Parce que le convertisseur (chapitre 1) ne nous permet pas d'agir très rapidement sur sa sortie (on ne peut agir que cycle par cycle et non événement par événement), l'asservissement est de type "régime permanent" au point de vue électrique.

- On veut éviter de mesurer le couple et la vitesse à l'aide de capteurs sur l'arbre du moteur car cela

encombre l'interface mécanique entre le moteur et sa charge et rend la greffe de l'asservissement à un moteur déjà installé coûteuse. En plus, les capteurs sur l'arbre sont considérés comme un lien de faible fiabilité et demandant de l'entretien régulier en milieu hostile. On se limite donc aux quantités mesurables aux bornes de la machine afin d'évaluer son état de fonctionnement (point d'opération); ne connaissant pas le système mécanique, seuls les courants et tensions aux enroulements statoriques, ainsi que leurs paramètres dérivés, sont utilisés.

- Le système réalisant l'asservissement est représenté à la figure 2.1-4. Comme ce système a des temps d'exécution relativement longs, les possibilités du logiciel qu'il peut exécuter en sont d'autant réduites; on peut donc oublier l'identification "on line" et l'utilisation de gros modèles (le moteur asynchrone, avec sa charge, présente un modèle du cinquième ordre ayant des paramètres qui dépendent de sa vitesse). On sait que certains éléments du modèle équivalent sont dépendants de la température et des conséquences de la stratégie d'opération des interrupteurs (harmoniques, etc). Si on faisait de l'identification, le modèle du moteur aurait à être réajusté régulièrement, et à chaque fois qu'on modifie la stratégie d'opération des interrupteurs. On sacrifie donc la dynamique de l'entraînement et on vise un entraînement améliorant les performances en régime permanent (il n'aura donc pas le temps d'agir de façon efficace lors de variation rapide de charge). On identifiera les éléments qui établissent le coût d'opération du moteur pour le minimiser.

Les approches considérées sont:

- la régulation d'un paramètre qui puisse constituer une bonne mesure du point d'opération du moteur;

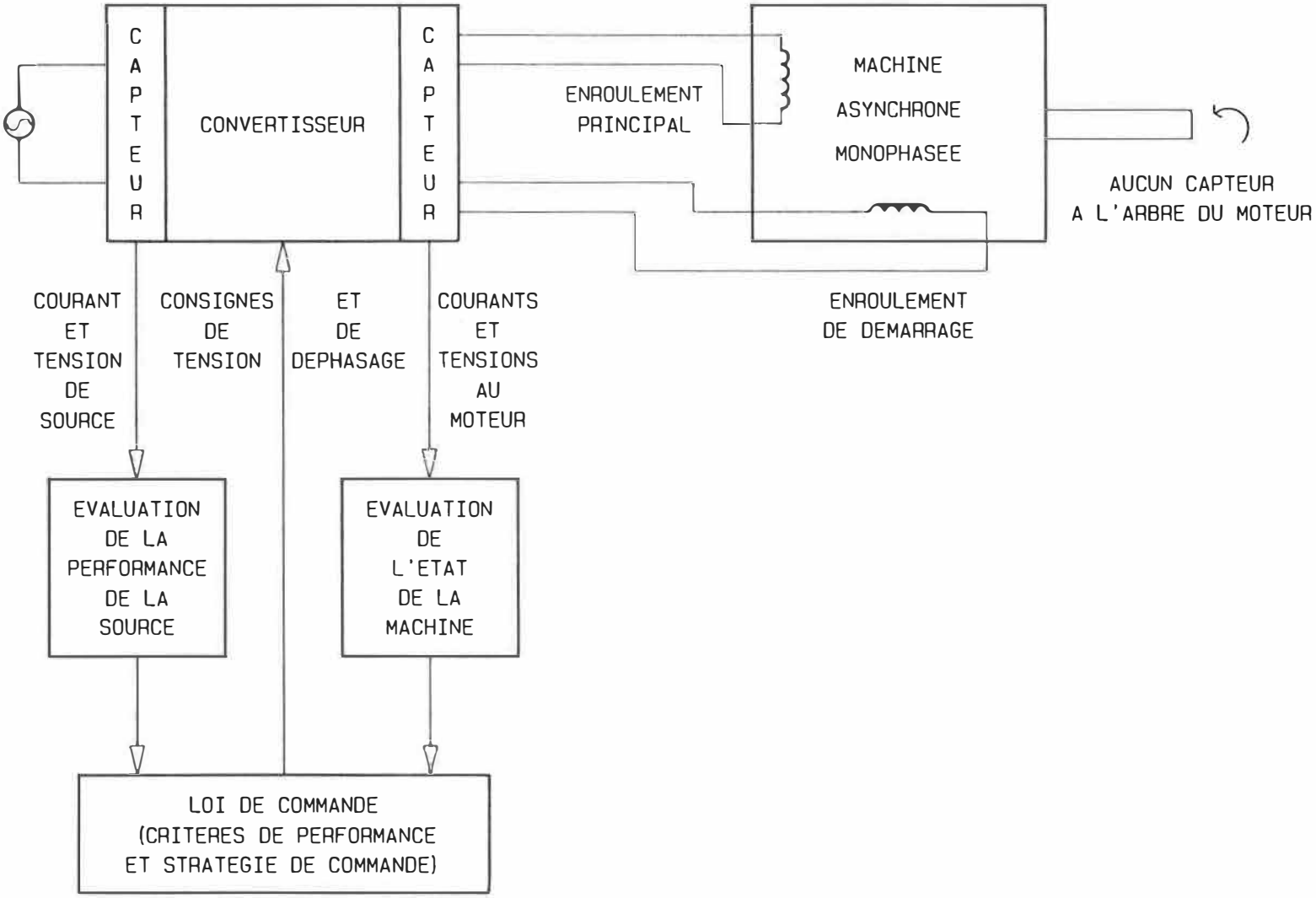


fig 2.1-4 SYSTEME REGLE RECHERCHE

- une stratégie de recherche itérative du point d'opération optimisant un critère mesurant la performance en régime permanent.

Pour la régulation, il faut trouver le paramètre. Nous avons vu plus tôt que le glissement était un paramètre important, mais comme on ne mesure pas la vitesse du moteur et qu'aucun capteur n'est inséré au rotor, il faudrait déduire la vitesse autrement. Deux possibilitées sont envisagées:

- la mesure du facteur de déplacement, ou de l'angle de déplacement (déphasage entre les fondamentales de courant et de tension au moteur): il permet d'évaluer l'impédance que représente le moteur, donc d'estimer le glissement ou les paramètres qui en dépendent.
- Une valeur déduite du modèle équivalent du moteur et obtenu par l'utilisation de la tension induite à l'enroulement de démarrage.

On reparle de ces paramètres dans les sections qui suivent.

En ce qui concerne l'approche itérative, une section complète lui est dévolue. Disons qu'on peut fixer le critère désiré, tenant compte de valeurs à la source ou à la machine, et le faire minimiser selon une certaine stratégie d'évolution. C'est un cas particulier de la commande optimale qui constitue une "méthode d'évolution **par une succession d'étapes vers l'optimisation**". C'est une méthode très avantageuse pour des procédés mal définis et pour une optimisation en régime permanent car elle ne requiert qu'une connaissance des réactions générales du système aux modifications des variables de commande.

Nous avons spécifié pourquoi la commande était nécessaire, quels sont les variables de commandes (la tension et le déphasage) et les types d'asservissements

proposés. Nous avons aussi entrevu le genre de paramètres ou critères utilisés par l'asservissement comme loi de commande. Ces lois de commandes seront précisées dans les sections qui suivent. (Le programme d'asservissement réalisera ces lois de commande. Même si un critère est optimisé, cela ne veut pas dire que l'entraînement réalise le "meilleur" compromis possible mais bien qu'il minimise une certaine fonction de coût).

2.1.1 Régulation d'un paramètre représentatif du point d'opération.

Pour améliorer les performances de la machine, on veut réguler un paramètre mesuré au moteur. Les paramètres d'intérêt sont des valeurs qui, maintenues constantes, nous assurent un point d'opération à grande performance pour la machine. Nous disposons de deux variables de commande: la tension statorique au moteur et le déphasage de cette dernière par rapport à la tension du réseau. Nous désirons utiliser la tension pour commander le point d'opération à la machine, et le déphasage en vue d'améliorer la performance du système vu de la source. Nous sommes donc en présence d'un système multivariable à deux variables de commandes (entrées) et deux sorties: la performance vue de la source (Ps) et celle de la machine (Pm). Le système équivalent est illustré à la figure 2.1.1-1. On notera la présence de G_{12} et G_{21} qui représentent le couplage entre les deux entrées de commande et les deux sorties. Ce couplage pose un problème pour le design des régulateurs: on n'a pas deux boucles indépendantes ayant chacune leur propre variable de commande et n'agissant que sur la sortie à réguler.

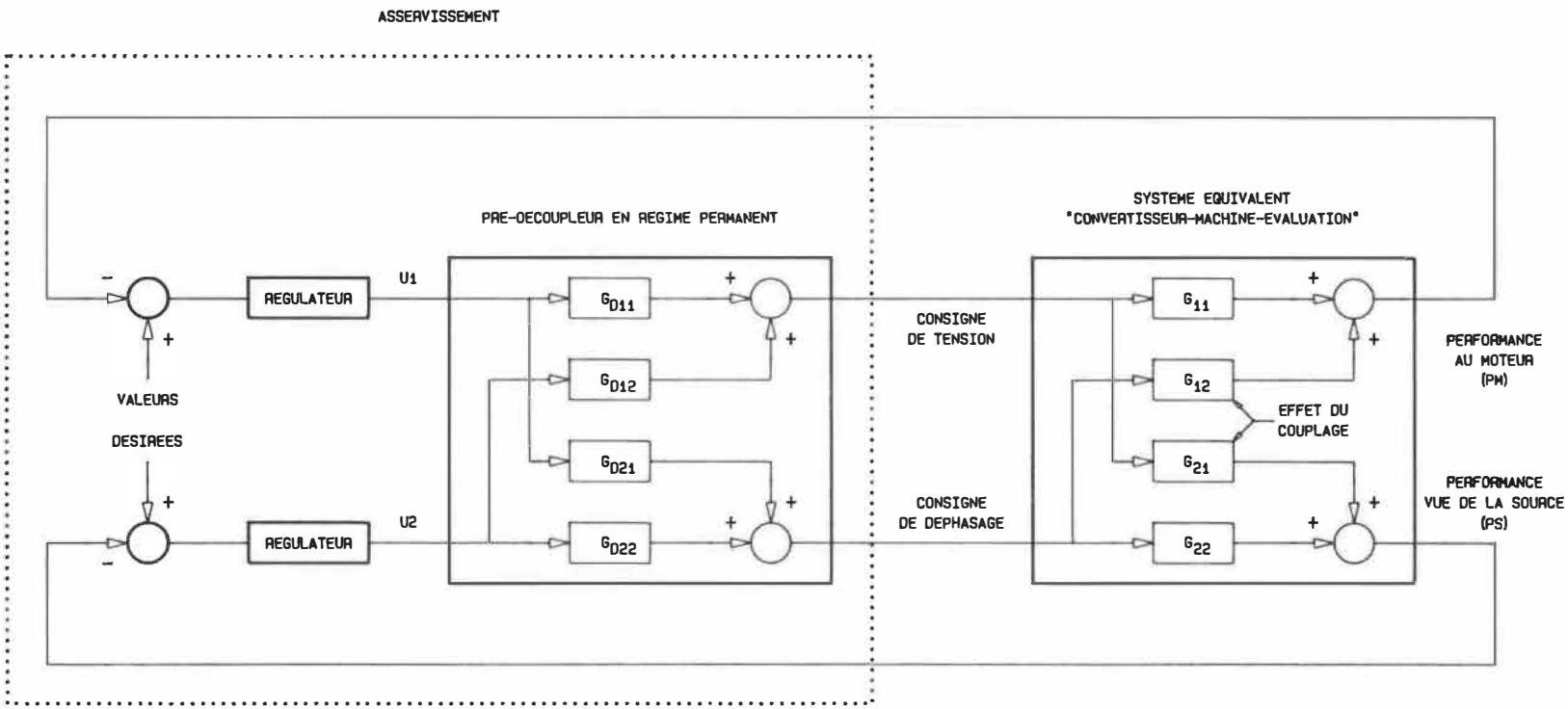


fig 2.1.1-1 LE SYSTEME MULTIVARIABLE EQUIVALENT DE L'ENTRAINEMENT

Le problème fondamental dans le design du régulateur d'un système multivariable à contre-réaction réside dans l'interaction entre les différentes variables d'entrées et de sorties. S'il n'y avait pas de couplage entre les différentes variables et que le nombre de variables d'entrée était égal au nombre de sorties à commander, alors on aurait une matrice de tansfert diagonale multipliant le vecteur des variables d'entrées; on pourrait alors considérer le système comme formé de deux boucles indépendantes. Pour obtenir deux boucles indépendantes, on fait précéder les entrées du système d'un pré-découpleur qui diagonalise la matrice de transfert en régime permanent (une matrice de gain). Cela est illustré sur la figure 2.1.1-1. De façon mathématique, on obtient alors le problème suivant:

$$\begin{vmatrix} Pm \\ Ps \end{vmatrix} = \begin{vmatrix} G_{11} & G_{12} \\ G_{21} & G_{22} \end{vmatrix} \cdot \begin{vmatrix} G_{D11} & G_{D12} \\ G_{D21} & G_{D22} \end{vmatrix} \cdot \begin{vmatrix} U_1 \\ U_2 \end{vmatrix}$$

d'où:

$$G_{11} \cdot G_{D11} + G_{12} \cdot G_{D21} = 1$$
$$G_{11} \cdot G_{D12} + G_{12} \cdot G_{D22} = 0$$
$$G_{21} \cdot G_{D11} + G_{22} \cdot G_{D21} = 0$$
$$G_{21} \cdot G_{D12} + G_{22} \cdot G_{D22} = 1$$

et: G_{ij} connus.

Connaissant le modèle du système, le pré-découpleur peut être calculé à l'aide des quatres équations données ci-haut. Le modèle dépend des paramètres utilisés comme variables de sorties; les entrées étant fixées par les variables commandables au convertisseur soient: l'amplitude

de la tension et son déphasage par rapport au réseau alimentant le convertisseur. Une fois les valeurs du pré-découpleur calculées, on peut alors déterminer les régulateurs des deux boucles indépendamment comme pour deux systèmes différents. Le réglage en régime permanent ne devrait plus montrer de couplage ou si peu qu'il peut être considéré comme une perturbation.

Nous avons décidé de dédier la consigne de déphasage au réglage du critère de performance vu de la source et la consigne de tension (amplitude) au réglage du point d'opération du moteur selon un certain critère de performance. Théoriquement, si on arrivait à déphaser la sortie du convertisseur tout en conservant une forme d'onde sinusoïdale, la performance du moteur n'en serait pas du tout affectée. Cela nous donnerait un couplage nul entre la consigne de déphasage et la performance du moteur. En pratique, le couplage de la consigne de déphasage sur la performance du moteur n'est pas nul car, lorsqu'on déphase la sortie du convertisseur, on change le contenu harmonique de la tension générée. Hors, le comportement du moteur est affecté par les harmoniques: une variation de l'amplitude des composantes harmoniques de l'onde de tension fera varier les pertes dues à ces harmoniques et donc le rendement de la machine et sa performance. Mais, nous l'avons quand même considéré nul, car la valeur du couplage est trop dépendante de la stratégie d'opération des interrupteurs du convertisseur et varie aussi selon l'amplitude de la tension générée par ce dernier. De plus, le couplage dépend du critère de performance de la machine et de la connaissance précise du modèle de cette dernière, ce qu'on veut éviter d'avoir à faire (on veut pouvoir changer de machine sans avoir à changer tous les paramètres des boucles de commande...). Comme la valeur de ce couplage

reste faible comparativement aux gains de la diagonale de la matrice de transfert, on a décidé de le considérer nul à priori; la boucle de réglage du critère de performance du moteur le verra comme une perturbation. Donc, on considère G_{12} nul. Mais, G_{21} ne l'est pas. En effet, le moteur peut être vu comme une impédance variable selon son glissement; le glissement pouvant être modifié par une modification d'amplitude de la tension, l'impédance l'est aussi et donc le déphasage du courant tiré de la source. La performance vue de la source est donc couplée à la consigne de tension. Pour le pré-découpleur, on se retrouve avec les conditions suivantes:

$$G_{D11} = 1/G_{11}$$
$$G_{D12} = \emptyset$$
$$G_{D21} = -G_{21}/(G_{11}.G_{22})$$
$$G_{D22} = 1/G_{22}$$

On constate que la boucle utilisant la consigne de tension est indépendante de celle utilisant le déphasage, même sans pré-découpleur. Les valeurs de gains sont calculées lorsque la matrice de transfert a été calculée ou mesurée pour les critères de performances désirés.

Pour simplifier le réglage et limiter les possibilités, on a décidé d'utiliser le facteur de déplacement comme paramètre à réguler du côté source; l'autre paramètre, du côté moteur, assurera un fonctionnement de la machine à grande performance par le réglage de son point d'opération en variant la tension statorique. Finalement, le choix de la statégie d'opération des interrupteurs déterminera la distorsion harmonique reliée à la commutation. Ainsi, le paramètre important qu'il faut surveiller du côté réseau est le facteur de

déplacement; l'utilisation efficace de l'énergie étant contrôlée par le bon réglage du moteur et la ditorsion minimisée par un choix de stratégie d'opération des interrupteurs.

Nous allons voir, dans les deux sections suivantes, les critères de performance à la machine (Pm) proposés et les boucles de réglage associées pour le point d'opération de la machine.

2.1.1.1 Le facteur de déplacement au moteur.

D'après Jian, Schmitz et Novotny [4] on peut établir une valeur optimale de glissement pour un critère de performance donné. Dans l'article de référence, ils dérivent cette valeur théoriquement du modèle de la machine en régime permanent et sinusoïdal pour différents critères (courant statorique minimal, puissance d'entrée minimale, facteur de puissance nominal, etc.) puis, ils font remarquer l'effet des non linéarités du moteur sur ces valeurs. On s'attend aussi à une dérive, par rapport au point optimal d'un critère, à cause du convertisseur non idéal générant la tension statorique.

On sait que le déplacement est une fonction du glissement, car il représente la phase de l'impédance équivalente de la machine. D'après Beck et Naunin [10], on peut déduire le glissement à partir du déplacement de façon précise. Sans aller jusqu'à la détermination du point d'opération de la machine par le déplacement, on peut le maintenir constant à une valeur correspondante à un glissement optimal pour un critère donné. On devrait alors être toujours assez près de cette valeur optimale de glissement. Malheureusement, cette valeur de glissement

optimal varie en fonction de la tension et aussi selon la stratégie d'opération des interrupteurs utilisée au convertisseur. Cela est dû à la non linéarité du circuit magnétique de la machine ainsi que des harmoniques produites par le convertisseur. Comme la valeur des variations dépendra du moteur utilisé, de la charge qu'il entraîne ainsi que de la stratégie d'opération des interrupteurs utilisée, on ne s'attardera pas à la quantifier car un des objectifs de l'entraînement est d'améliorer la performance globale peu importe la machine. On se limitera à la connaissance du facteur de déplacement à une valeur correspondant à un glissement optimal pour un critère donné: une seule valeur à tenir en compte.

A l'aide d'un logiciel de simulation basé sur le modèle DQ stationnaire par rapport au stator, nous avons déterminé le facteur de déplacement à la machine ainsi que le couple électromagnétique qu'elle génère en fonction de sa vitesse. Le moteur simulé est un Wagner-Leland de type SPBKJ10 monophasé de 1/2 hp, fonctionnant à 60Hz, 115V, et dont la vitesse nominale est de 1725 tpm. Les résultats sont illustrés au graphique de la figure 2.1.1.1-1. On obtient des résultats très semblables à ceux de Beck. A l'aide de ces courbes et de la figure 2.1-2, on peut évaluer le couplage de la consigne de tension statorique au moteur sur le déphasage (du courant) vu de la source. En effet, en utilisant une approximation linéaire pour relier le couple et le glissement dans la région de faible glissement, et en utilisant le modèle en régime permanent sinusoïdal, on peut écrire:

$$Temg = Knom.(E/Enom)^2.S$$

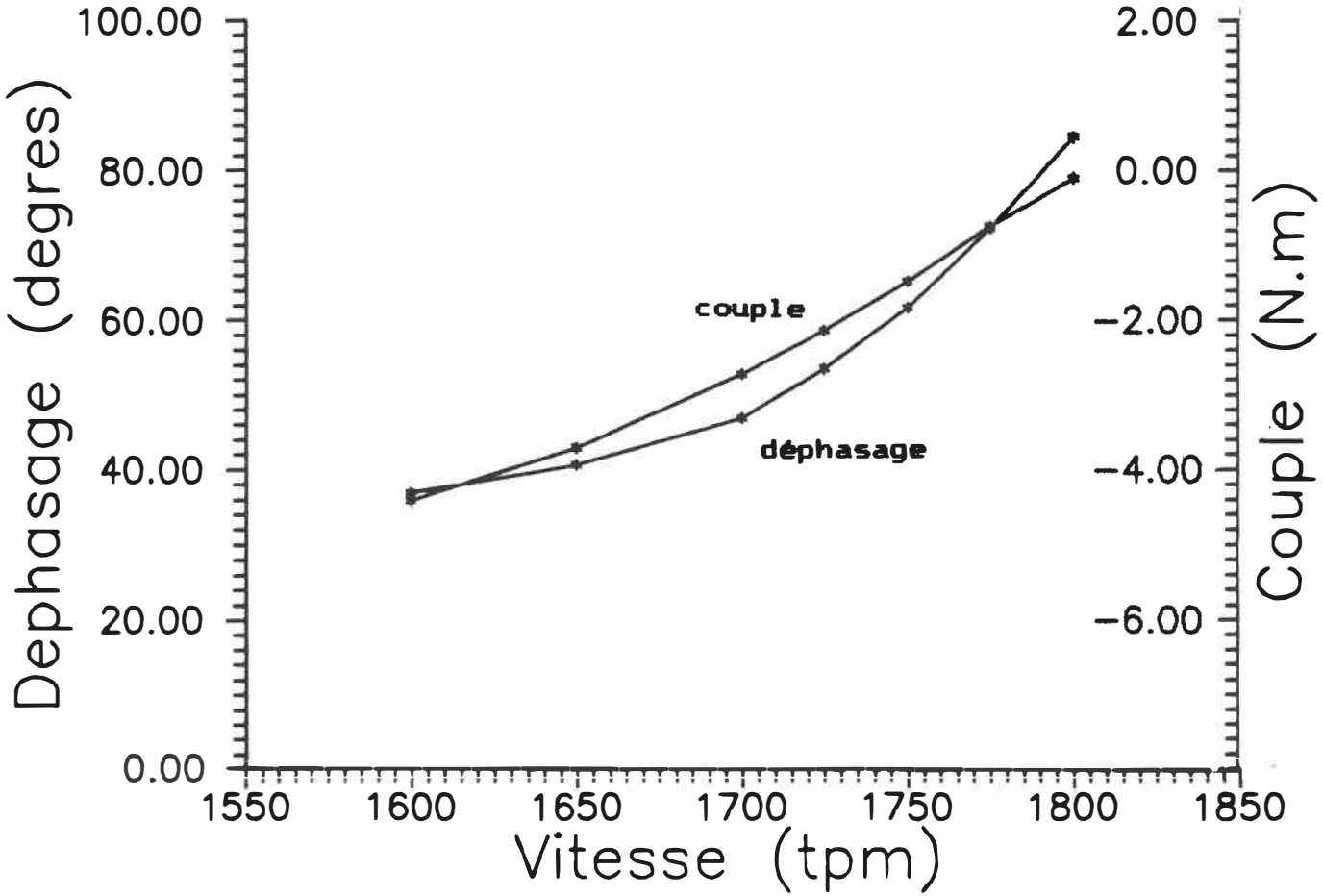


Figure 2.1.1.1-1 Courbes du déphasage et couple en fonction de la vitesse.

$$\frac{dS}{dE} = \frac{(-2.Enom^2.Temg).}{Knom} \frac{1}{E}$$

et à partir du graphique du déphasage au moteur (égal au déplacement en régime sinusoïdal) en fonction du glissement à couple constant, on note une ressemblance à une demi parabole, donc une fonction quadratique. On obtiendrait une courbe similaire en l'exprimant en fonction du glissement. On peut donc écrire:

$$\frac{dDEPLACEMENT}{dS} = \frac{d\theta}{dS} = K.S + K_2$$

ce qui nous donne une relation pour G_{21}:

$$\frac{d\theta}{dE} = (K.S + K_2).\frac{(-2Enom^2.Temg).}{Knom} \frac{1}{E}$$

Cette relation, en plus de ne pas être linéaire, dépend d'approximations et de paramètres difficilement contrôlables tels le couple de charge et la constante liant le couple et le glissement. Autour de la vitesse de 1725 tpm, on obtient un gain entre la consigne de tension et le déphasage vu de la source de 0,45 degré/volt.

Le gain direct liant la consigne de déphasage de la tension de sortie du convertisseur et le déplacement mesuré à la source (entre la tension source et le courant de source) est de -1: si on augmente le déphasage (avancer la tension de sortie par rapport à la tension du réseau) de un degré, le facteur de déplacement diminue de un degré car le courant au moteur est avancé de un degré et celui de source aussi, mais pas la tension de source.

En ce qui concerne le gain direct liant la consigne de tension au moteur et le déplacement entre le courant et la tension au moteur, on ne cherche pas à le quantifier car

il est dépendant du point d'opération, du moteur et de la charge. La relation $d\theta/dE$ donne une idée de la variation de ce gain (ou gain en petit signal) à couple constant, mais non de sa grandeur absolue. Comme la boucle de déplacement au moteur est indépendante de celle du déplacement à la source, la valeur exacte de ce gain est inutile pour le calcul du pré-découpleur.

En se référant au schéma de la figure 2.1.1-1 et aux équations de la section précédente, on obtient donc les valeurs suivantes pour obtenir une matrice de gains diagonale, et donc des boucles indépendante:

$G_{D11} = 1$
$G_{D12} = 0$
$G_{D21} = 0,4$
$G_{D22} = -1$

En ce qui concerne le régulateur des boucles découplées, il aura une composante de type intégrateur car les gains des boucles sont trop variables.

2.1.1.2 Un autre paramètre obtenu à l'aide de la tension induite à l'enroulement de démarrage

Afin de trouver un paramètre facilement mesurable qui nous indiquerait l'état de charge de la machine ou qui nous donnerait une idée de sa réserve par rapport à un fonctionnement à haute performance, une étude du système équivalent du moteur s'est avérée nécessaire. La figure 2.1.1.2-1 illustre le modèle général d'une machine asynchrone; nous le décrivons à l'annexe B. Il nous permet d'identifier les valeurs qui nous seraient utiles de mesurer pour établir la rétroaction nécessaire pour

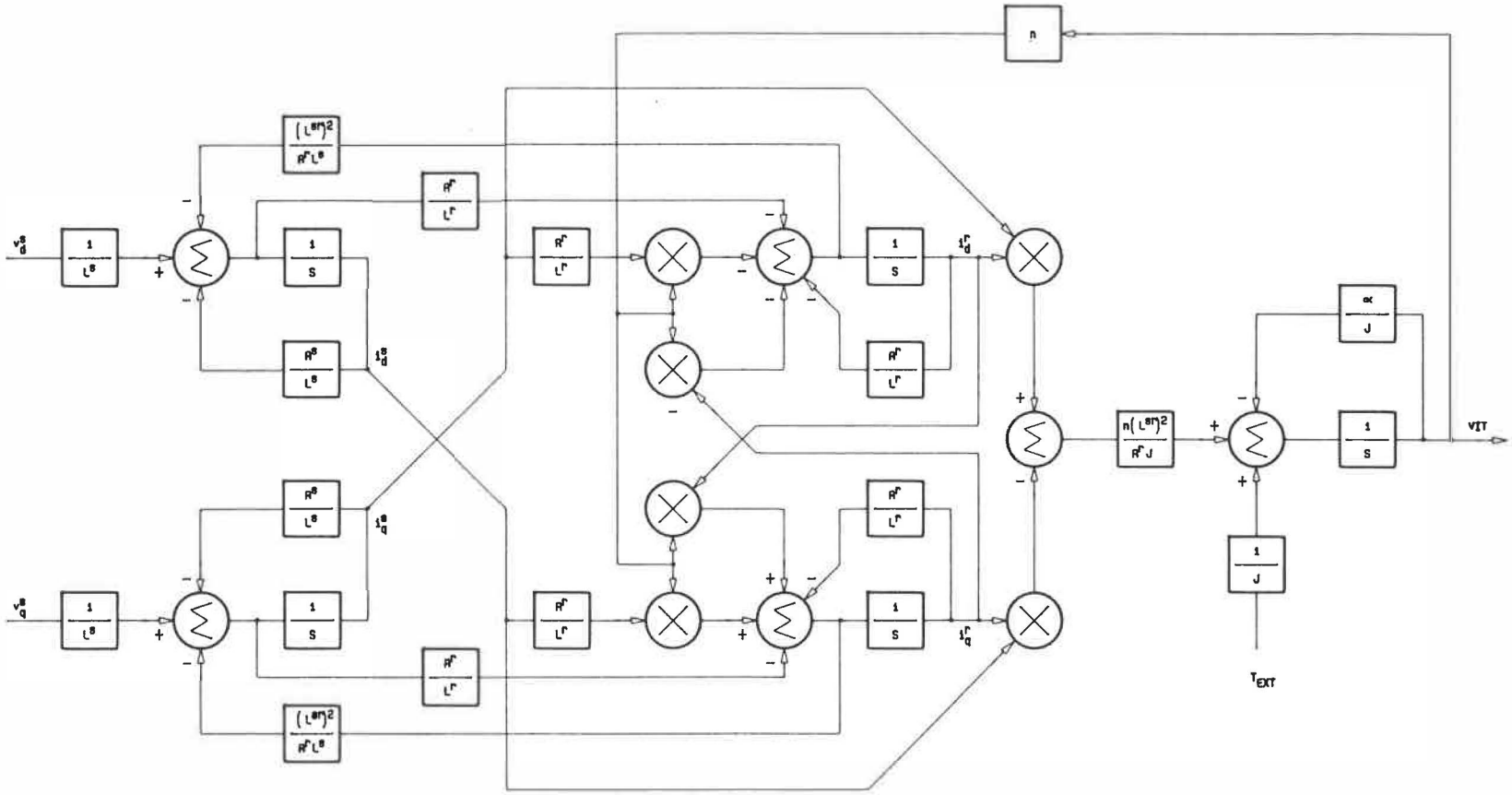


fig 2.1.1.2-1 MODELE EQUIVALENT DE LA MACHINE ASYNCHRONE

commander la machine.

Le choix d'une tension d'opération serait facilité si on connaissait ce qui se passe à l'arbre du moteur, c'est-à-dire: le couple demandé, l'inertie de la charge, la vitesse de rotation. Mais nous n'y permettons aucun capteur; seuls les tensions et courants au moteur doivent nous donner les renseignements nécessaires pour le réglage en régime permanent. Notre réglage améliore les performances en régime permanent; nous pouvons nous passer de tout ce qui a trait à la dynamique et regarder l'ensemble à l'équilibre d'un régime permanent.

Simplifions d'abord le modèle général du moteur en considérant le démarrage terminé. Le courant en quadrature est alors nul, éliminant bien des blocs l'impliquant lui et sa dérivée. Le modèle simplifié est donné à la figure 2.1.1.2-2 (ne pas considérer ce qui est entre parenthèses). En régime permanent, on peut faire plusieurs constatations en imposant l'équilibre à certains sommateurs. Ainsi, à l'équilibre, on a que:

- le couple électromagnétique demandé à la machine est égal à celui demandé par la charge plus l'effet des frictions au moteur, car la variation de vitesse est nulle;
- la tension induite à l'enroulement en quadrature est proportionnelle à la dérivée temporelle du courant rotorique en quadrature.

Cela est montré en a et b de la figure 2.1.1.2-3. Sachant que le couple électromagnétique est proportionnel au produit des courants statorique direct et du courant rotorique en quadrature (qui n'est pas accessible dans une machine à cage), on constate toute l'importance de la dernière remarque faite plus haut. En effet, on peut déterminer le couple électromagnétique à l'aide de la

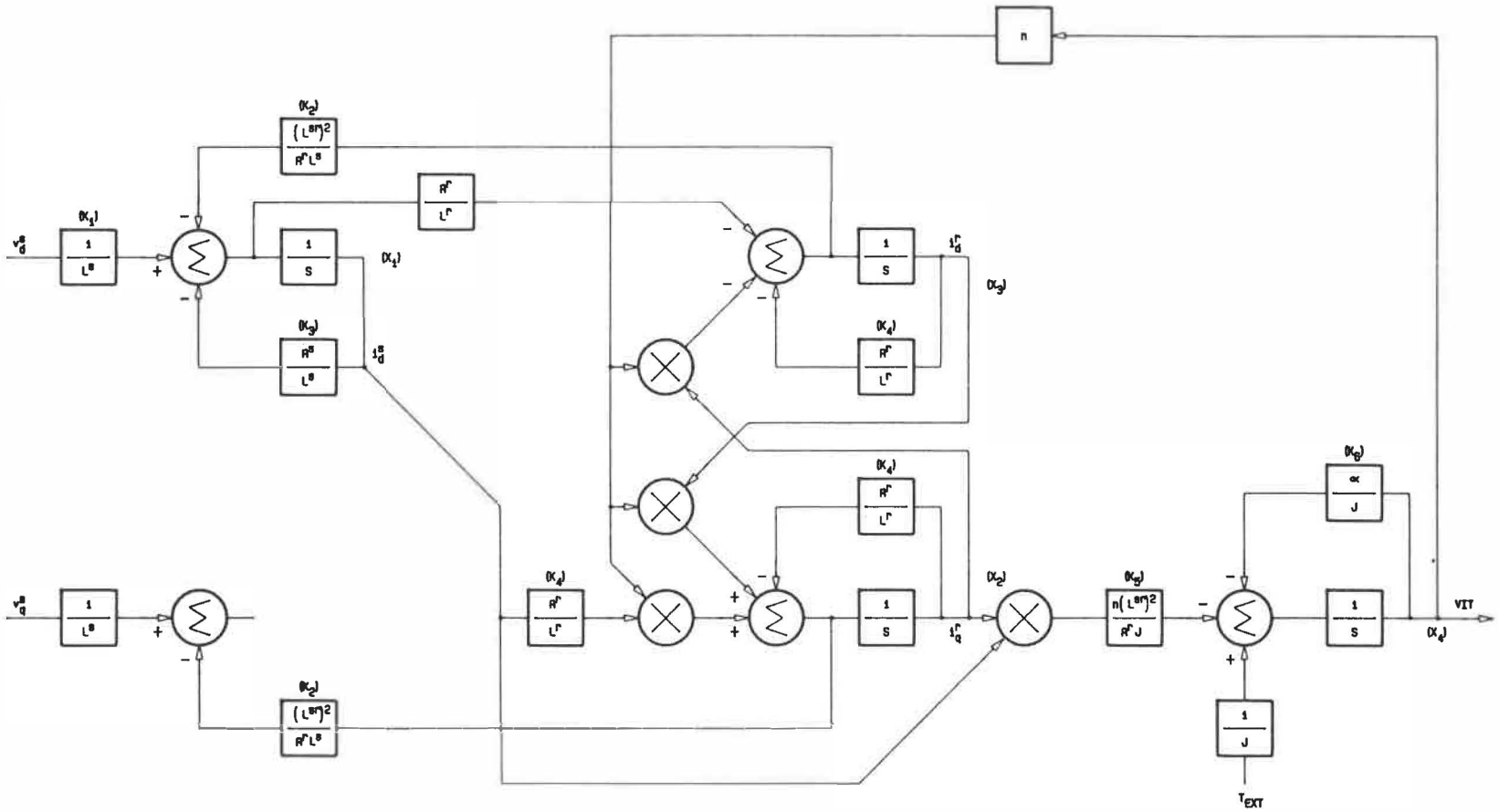


fig 2.1.1.2-2 MODELE APRES DEMARRAGE

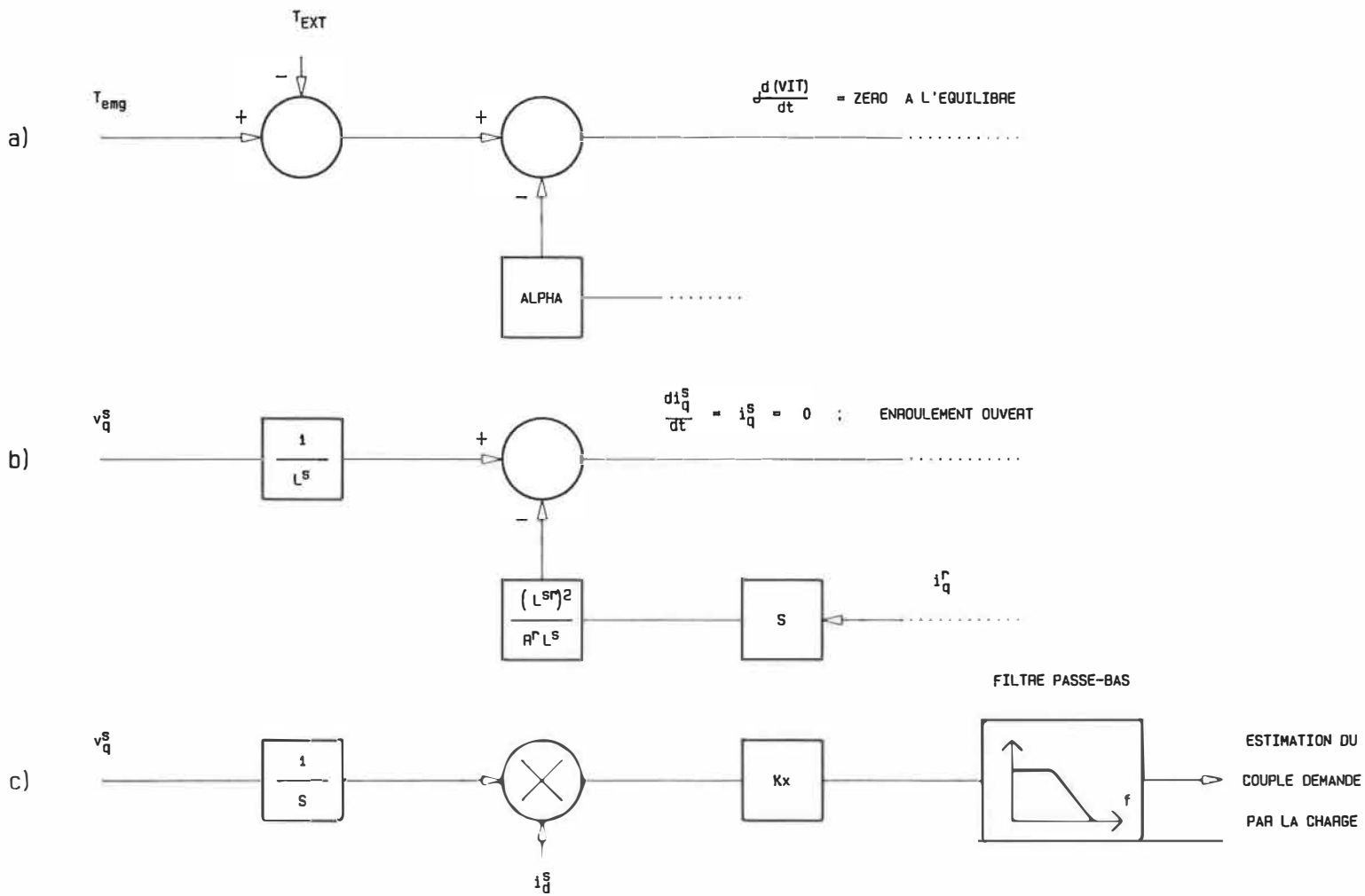


fig 2.1.1.2-3 SIMPLIFICATIONS EN REGIME PERMANENT

tension induite à l'enroulement statorique en quadrature et du courant de l'enroulement statorique direct d'une machine généralisée:

$$T_{emg} = K.i_d^s.\int(v_q^s.dt)$$

et on a que:

$$T_{emg} = T_{EXT} + T_{friction}$$

en régime permanent pour le système mécanique . Donc, on peut évaluer l'état de charge du moteur en régime permanent. Le régime permanent mécanique est à considérer car c'est lui qui présente la constante de temps la plus grande du système, le système électrique réagissant presque instantannément. La figure 2.1.1.2-3c illustre ce concept.

Mais, l'enroulement en quadrature, qu'est-ce que c'est pour une machine asynchrone monophasée à cage d'écureuil? Sur une machine monophasée, on a l'enroulement principal que j'associe à l'enroulement direct et on a l'enroulement de démarrage qui, comme son nom l'indique, permet, en générerant un couple non nul lorsque la machine est au repos, le démarrage. Ce dernier enroulement doit agir, en partie au moins, comme l'enroulement statorique en quadrature d'une machine généralisée car, autrement, il n'y aurait aucun couple de produit à vitesse nulle. On remarquera que, même si ce n'est pas un enroulement en parfaite quadrature, le fait qu'il nous fournisse une tension induite par les courants rotoriques et la rotation du moteur en fait un capteur potentiel qu'on se doit de considérer. Pour cette raison, nous sommes intéressé à son utilisation.

Nous avons donc considéré que l'enroulement de démarrage en était un en quadrature. Nous avons utilisé ce dernier dans un circuit réalisant le schéma de principes de la figure 2.1.1.2-3c et vérifié sa sortie en fonction de l'état de charge de la machine asynchrone. Pour ce faire, nous avons réalisé un montage correspondant au schéma présenté à la figure 2.1.1.2-4. La tension provenait d'un auto-transformateur et était donc sinusoïdale. Des résultats similaires ont été observés pour les différentes stratégies d'opération des interrupteurs du convertisseur. Pour conserver l'indépendance du choix de la stratégie, nous n'utiliserons que les résultats obtenus en régime sinusoïdal. De ces mesures, on peut tracer les graphiques donnant la tension de sortie du circuit se servant de l'enroulement de démarrage , soit $V_{circuit}$, en fonction de la vitesse de rotation du moteur ainsi qu'en fonction du couple au moteur, et cela pour différentes amplitudes de tension statorique (enroulement direct...). Ces graphiques sont donnés aux figures 2.1.1.2-5 et 2.1.1.2-6 respectivement.

Du deuxième graphique, on note que $V_{circuit}$ est nul, quelle que soit la tension statorique, autour de 1727 tpm; Cette vitesse est tout près de la valeur nominale pour la machine utilisée (1725 tpm). Pour des vitesses supérieures à la vitesse nominale, $V_{circuit}$ est négatif et l'inverse pour des vitesses inférieures. Comme on le sait, si la vitesse est supérieure à la vitesse nominale, c'est que le moteur peut générer efficacement plus de couple que celui requis par la charge; conclusion inverse pour le cas où la vitesse est inférieure à la nominale. On se retrouve donc avec une mesure de la réserve de couple qu'on a par rapport à celui nécessaire pour fonctionner à vitesse nominale (ou très près). On peut simplement utiliser cette indication

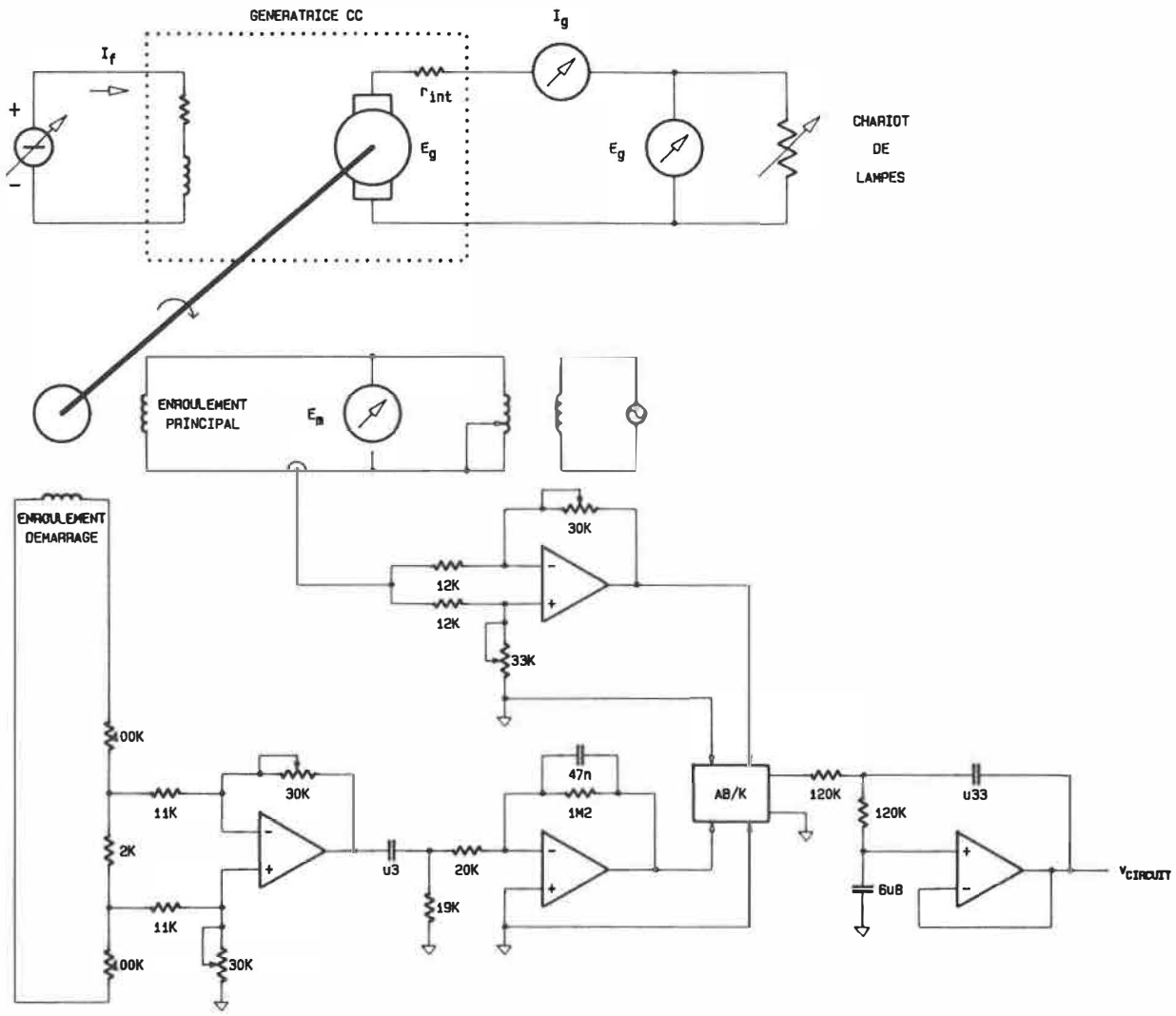
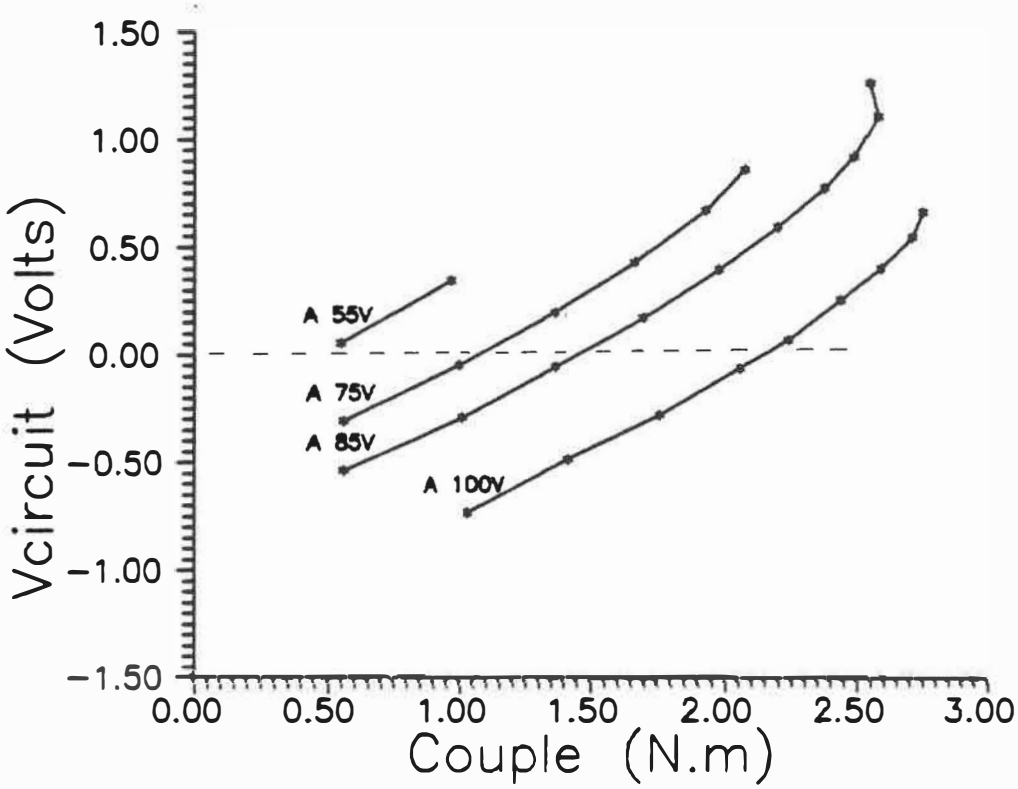


fig 2.1.1.2-4 MONTAGE DU CIRCUIT D'ESTIMATION DU COUPLE



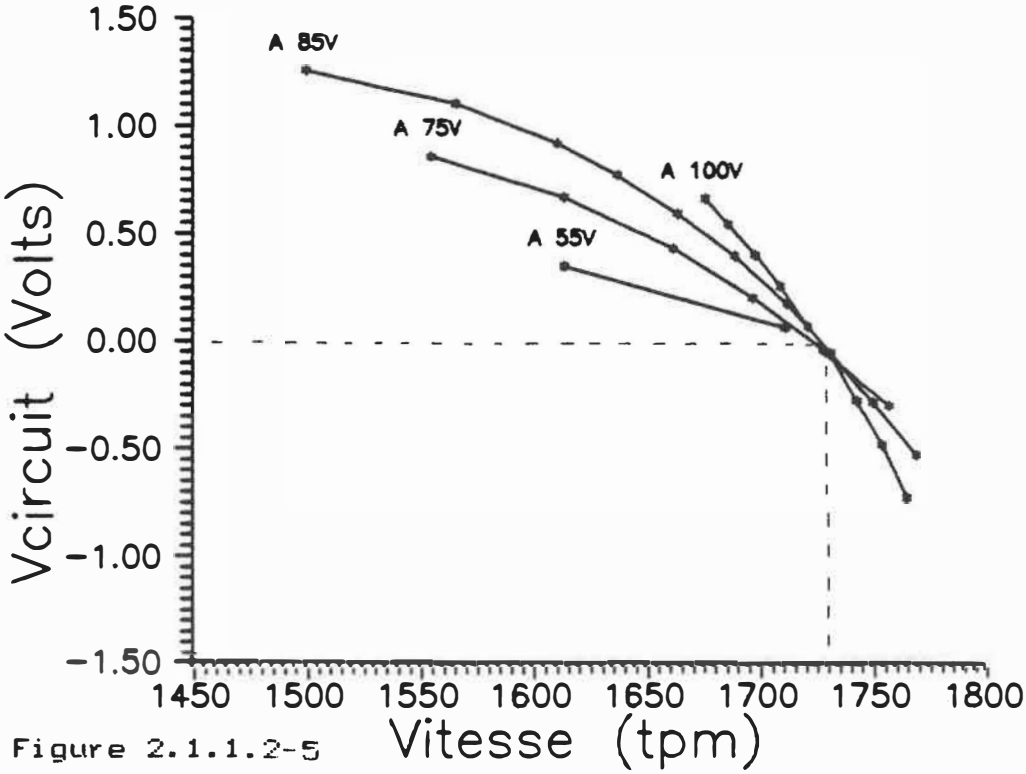


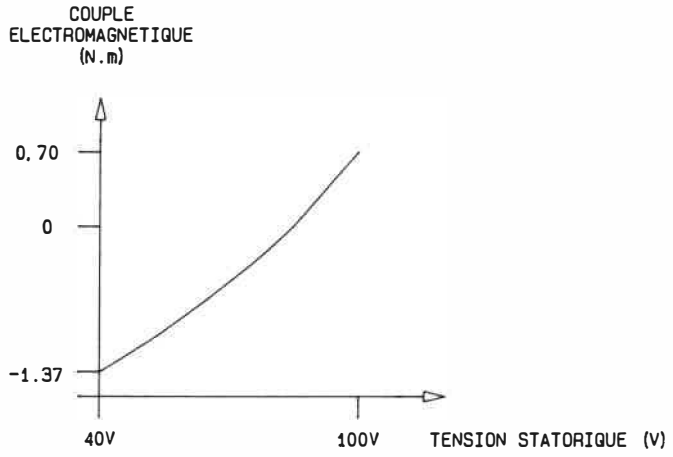
Figure 2.1.1.2-5 et 6
Graphiques du couple et de la vitesse en fonction de Vcircuit pour différentes tensions.

pour maintenir des condition similaires aux nominales à la machine en régulant de façon à annuler $V_{circuit}$. C'est déjà un résultat intéressant.

Du premier graphique, on note que la relation couple de charge en fonction de $V_{circuit}$ pour différentes tensions nous donne une famille de courbes régulièrement distancées les unes des autres sur toute leur plage et montrant une relation assez linéaire, du même genre qu'une courbe de couple en fonction du glissement. Si on connaît une courbe du couple en fonction de $V_{circuit}$ pour une amplitude de tension statorique donnée, on peut évaluer le couple de charge au moteur connaissant $V_{circuit}$ et la compensation de couple nécessaire pour déplacer la courbe de référence sur celle d'une tension statorique donnée. Cette valeur de compensation de couple varie un peu. Nous l'avons choisie à $V_{circuit}$=0 car à cette valeur, le moteur tourne près de sa vitesse nominale et donc très près de son meilleur point d'opération. On devrait être près de cette vitesse si on régule correctement... La courbe de référence devrait être assez centrée dans la plage de variation de l'amplitude de tension. C'est pourquoi nous avons choisi la courbe à 85V: la tension maximale étant 120V et la minimale autour de 50V. La courbe de compensation est donnée à la figure 2.1.1.2-7a. Comme on l'illustre à la figure 2.1.1.2-7b, on peut quantifier le couple demandé au moteur; on a donc un estimateur de couple.

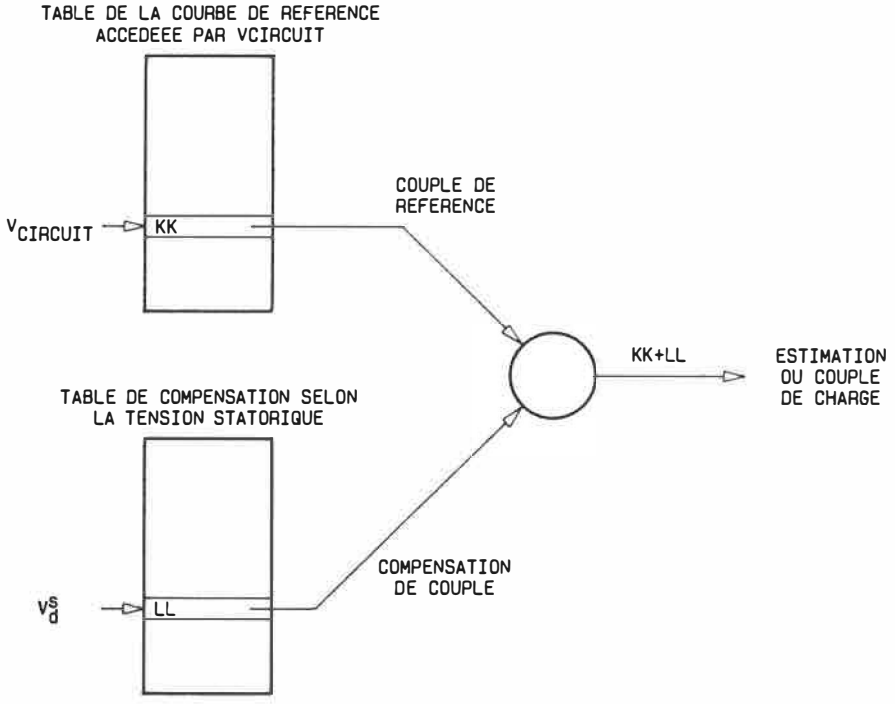
Ainsi, nous pouvons utiliser l'enroulement de démarrage pour dériver une valeur nous permettant de réaliser un estimateur de couple, ou encore d'évaluer directement la "réserve" qu'on a par rapport à un point d'opération similaire au point d'opération nominal de la machine. En ce qui concerne la régulation de paramètre, on utilisera directement $V_{circuit}$ comme indication de

a)



COMPENSATION DE COUPLE P/R A LA VALEUR DE REFERENCE A 85V (POUR Vcircuit = 0V)

b)



UTILISATION DES TABLES POUR L'ESTIMATION DU COUPLE

fig 2.1.1.2-7 ESTIMATION DU COUPLE

performance à la machine; on tentera d'annuler cette tension. Pour ce faire, on utilise $V_{circuit}$ comme feedback et ØV comme valeur désirée; le régulateur étant du même type que celui utilisé pour réguler le déplacement au moteur. La possibilité d'estimation de couple nous ouvre une autre avenue dont nous parlerons en 2.1.3.

2.1.2 L'évolution d'optimisation en régime permanent.

C'est une méthode qui ne requiert pas la connaissance du modèle mathématique du procédé à optimiser: c'est le procédé lui-même qui est utilisé. On utilise une procédure itérative pour ajuster les variables de commande dans le but d'optimiser un critère de performance donné; chacune des itérations nous rapproche du point d'opération optimum. C'est une approche de feedback car la procédure d'évolution se sert du calcul de la fonction à optimiser au point d'opération actuel du procédé afin de déterminer les ajustements requis aux variables de commande.

La fonction d'optimisation s'écrit:

$$F = f(X_i, Y_i)$$

X_i = les variables de commande du procédé;
Y_i = les sorties mesurées du procédé.

notons qu'un modèle du procédé se représente comme:

$$Y_i = f_2(X_i)$$

et, dans notre cas, il n'est pas utilisable ou connu. Ainsi, la relation générale qui aurait permis de relier F à X_i ne peut être obtenue. On se retrouve avec:

$$F = f(Y_i(X_i), X_i)$$

dont les Y_i ne sont connus qu'après le réglage des X_i. La relation f permet de combiner, par l'utilisation de facteurs de pondération, l'effet des différents X_i et Y_i obtenus en une mesure de coût (ou d'optimalité). Les facteurs de pondération appliqués aux entrées X_i peuvent aussi servir de protection en attribuant un coût élevé lorsque les X_i sortent des plages permises pour le bon fonctionnement de l'entraînement. Enfin, disons qu'une forme quadratique pour f est intéressante car elle présente toujours un minimum.

Cette méthode peut être utilisée avantageusement si une ou plusieurs des conditions suivantes sont rencontrées au niveau du modèle du procédé:

- il est techniquement impossible à développer;
- il est non fiable ou imprécis pour la plage d'opération désirée;
- il est coûteux à développer ou à utiliser;
- le but de l'asservissement est d'améliorer les performances à faible coût plutôt qu'optimiser parfaitement à grand coût.

En ce qui nous concerne, un modèle précis du moteur asynchrone est techniquement réalisable mais difficilement utilisable: il faut tenir compte de tous les paramètres variables avec la température, de l'effet des harmoniques, ou encore de la saturation de certaines réactances. La gestion de ce genre de modèle demande beaucoup de calculs, des mesures rapides et précises et de faibles temps d'exécution; cela requiert un matériel informatique (microprocesseurs, convertisseur analogique/numérique, mémoires) très rapide (beaucoup plus qu'un 65Ø2). Evidemment, cela dépasse largement le coût d'investissement

qu'on peut assumer en vue d'améliorer les performances d'une machine monophasée de faible puissance. De par cela, et parce que notre but est l'amélioration des performances en régime permanent, la méthode d'évolution nous apparaît très avantageuse.

La figure 2.1.2-1 illustre le principe de la méthode et de sa procédure générale simplifiée. Elle est constituée de quatres étapes successives répétées tant qu'on n'atteint pas le maximum de notre fonction de performance (ou le minimum d'une fonction de coût). On s'en doute, il y a plusieurs façons d'ajuster les variables de commandes et il est possible d'avoir plusieurs points d'opération maximisant la performance du système. La logique utilisée pour déterminer les ajustement à faire est appelée "stratégie". Pour deux variables de commande, la fonction évaluant la performance à une valeur donnée se représente dans un espace tri-dimensionnel ou encore comme des courbes de niveaux dans le plan formé par les variables de commande. Nous appelerons ce plan une surface de réponse. La série de pas effectués pour atteindre le niveau maximal de la surface de réponse constitue la trajectoire. Cette trajectoire est indépendante du temps car on fonctionne en régime permanent: après chaque pas, on doit attendre d'être en régime permanent pour calculer les prochains ajustements. En pratique, on devra laisser suffisamment de temps entre chaque pas pour atteindre le régime permanent, c'est-à-dire: attendre que les variations des valeurs mesurées soient faibles (fin des variations dues aux transitoires). Les variations de charge viendront relancer le processus dans une nouvelle progression vers un nouvel optimum pour la nouvelle valeur de charge: si , lors d'une comparaison entre la valeur de **F** mémorisée en fin d'évolution et la valeur courante, la différence se révèle

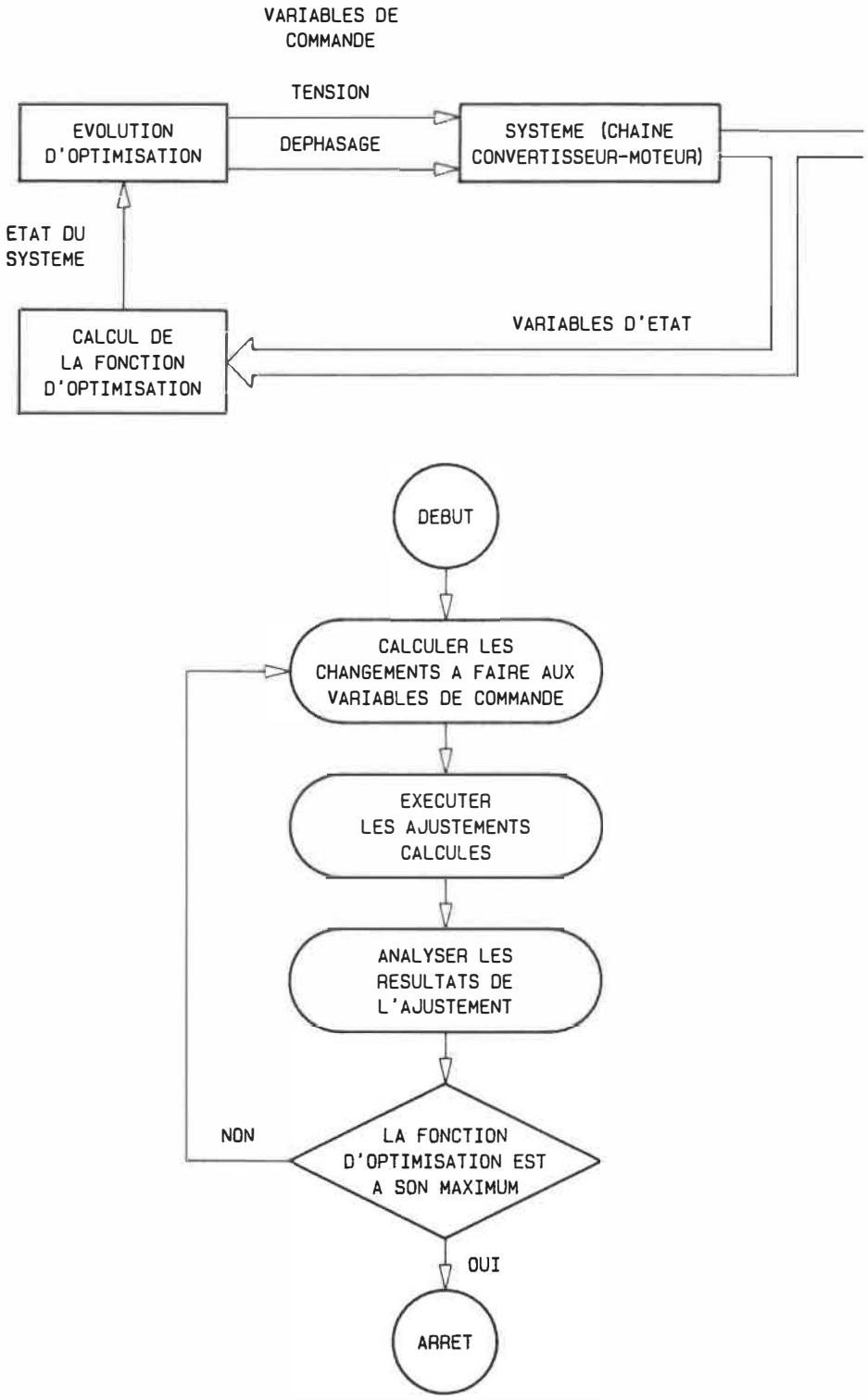


fig 2.1.2-1 LE PRINCIPE ET LA PROCEDURE

supérieure à l'effet d'un déplacement d'un pas des variables de commande, on relance le processus d'optimisation. La figure 2.1.2-2 illustre l'idée de la mesure de performance d'un système en fonction des variables de commande pour un état de charge donné, ainsi que la surface de réponse correspondante.

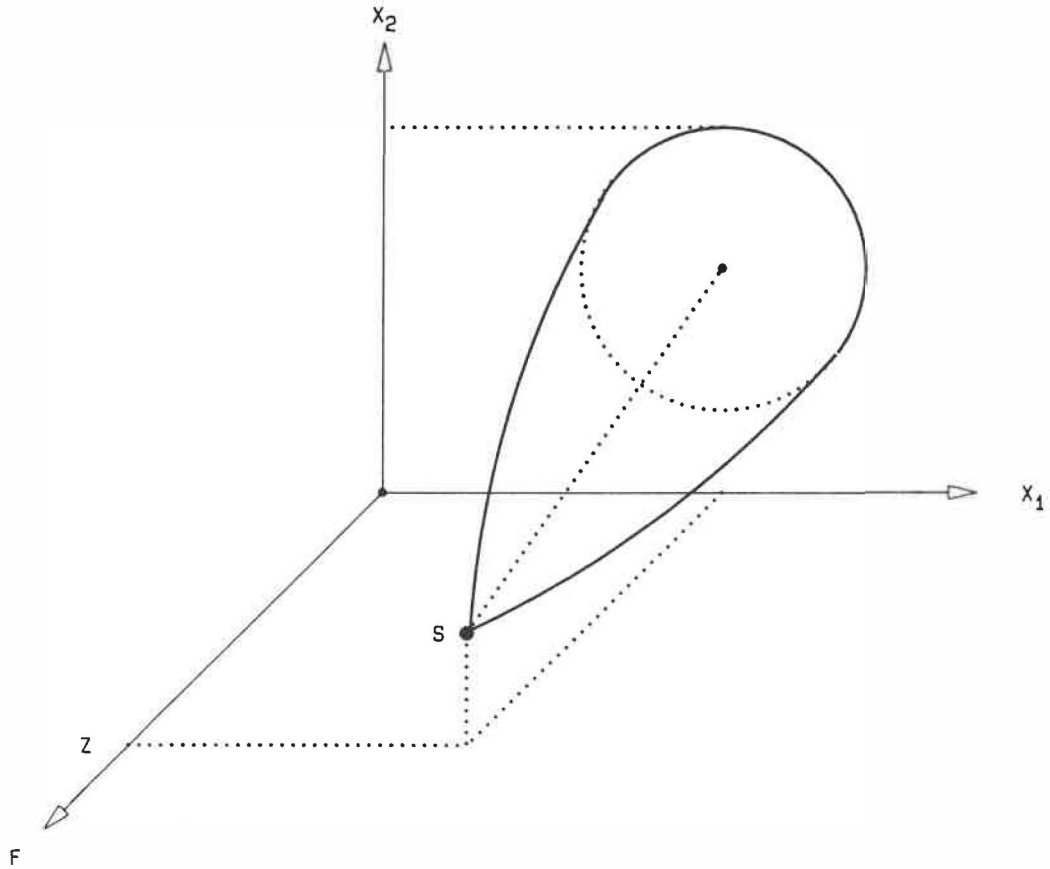
Dans la section suivante, nous décrivons la stratégie d'évolution que nous avons choisie. Ce choix est surtout relié aux possibilités restreintes de notre système de commande, qui doit rester modeste, ainsi qu'à la rapidité et la simplicité avec lesquelles on se dirige vers le point optimal (point qui change avec la charge entre autres).

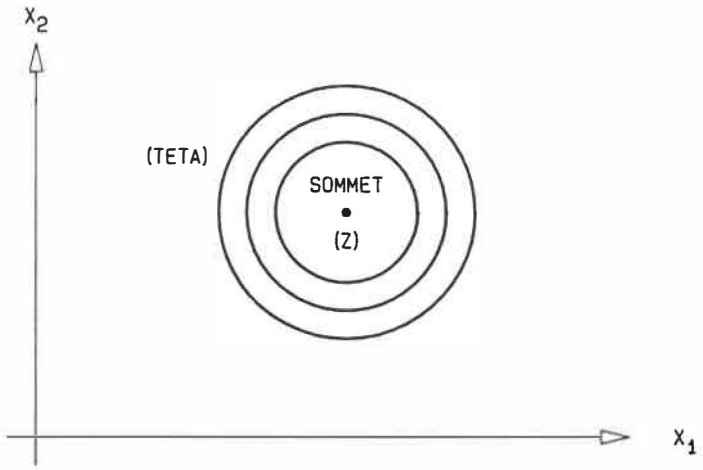
2.1.2.1 La stratégie d'évolution.

C'est la procédure pour déterminer les ajustements aux variables de commande pour le prochain pas de la trajectoire nous menant au point optimal. L'efficacité d'une stratégie se mesure par:

- la vitesse pour atteindre le sommet;
- sa robustesse face aux abérations d'une surface de réponse (faux maximum, ressèrement des courbes de niveaux, discontinuités);
- la certitude d'atteindre le "vrai maximum";
- sa simplicité.

Les ajustements à faire sont déterminés à l'aide du gradient de la fonction à optimiser. Le gradient est un vecteur dont les composantes sont orientées selon les axes des variables de commande. Chacune des composantes est égale à la dérivée partielle de la fonction à optimiser par





PROJECTION DANS LE PLAN DES VARIABLES DE COMMANDE

fig 2.1.2-2 INDICE DE PERFORMANCE DU SYSTEME

rapport à la variable de commande considérée à un point d'opération p donné:

$$\textbf{GRADIENT AU POINT P} = G^p = \begin{vmatrix} G_1^p \\ \\ G_2^p \end{vmatrix} = \begin{vmatrix} \dfrac{dF^p}{dX_1} \\ \\ \dfrac{dF^p}{dX_2} \end{vmatrix}$$

Où:

F^p = valeur de la fonction à optimiser au point p.

X_i^p = valeur de la "ième" variable de commande au point p.

G_i^p = composante du gradient reliée à la "ième" variable de commande.

Les composantes du gradient selon les axes des variables de commande forment un vecteur qui indique la direction de la plus grande augmentation de la fonction d'optimisation "F" au point p. Le gradient pointe vers un maximum et s'éloigne d'un minimum. On peut se servir de ce dernier pour déterminer les ajustements à faire aux variables de commandes de façon à évoluer dans la direction du gradient, donc vers le point maximal (sommet). On notera que pour minimiser une fonction de coût, on n'aurait qu'à se diriger dans la direction inverse du gradient. Dans ce qui suit, nous considérons une maximisation de la fonction; les résultats sont directement transposables au cas d'une minimisation en remplaçant G^p par $-G^p$. L'équation d'un pas

de la trajectoire se calcule donc de la façon suivante:

$$X_i^{p+1} = X_i^p + K_i^p . D_i^p$$

où:

D_i^p = la "ième" composante du vecteur normalisé (norme=1) en direction du gradient au point p.

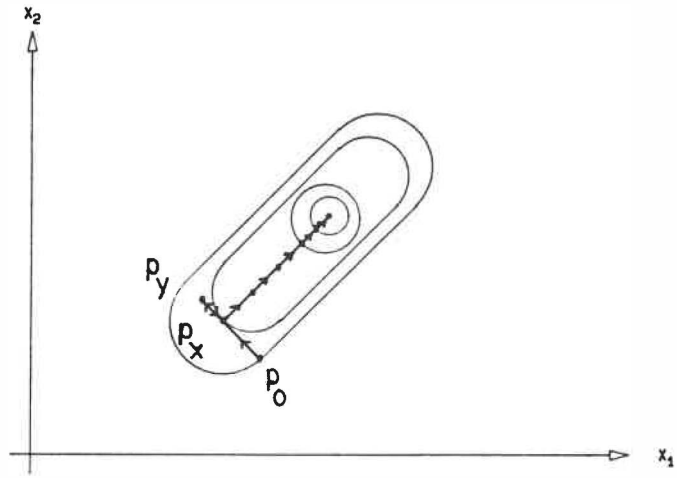
K_i^p = le facteur d'échelle appliqué sur la "ième" composante du gradient au poit p.

Voyons voir comment on peut utiliser cela dans une stratégie simple et assez rapide pour le système. Supposons que nous sommes à un point de départ p_0. La première chose à faire est d'y évaluer le gradient et d'évaluer la valeur initiale de la fonction d'optimisation à ce point. Ensuite, de calculer les nouvelles valeurs des variables de commande et de faire le pas. On supposera que le facteur d'échelle est unitaire au départ. Puis, on mesure les valeurs nécessaires au calcul de notre fonction d'optimisation, on l'évalue et, si elle s'accroît, on ne remesure pas le gradient. On se sert de la dernière valeur mesurée du gradient tant que la fonction d'optimisation s'accroît. Si on note une diminution, on revient au point précédent et on refait la mesure du gradient. Cela nous permet d'accélérer le processus de convergence de beaucoup car le calcul du gradient est long en pratique; nous regarderons comment on le fait un peu plus loin. On tient compte aussi du taux de variation de la fonction d'optimisation entre deux pas successifs; cette mesure nous renseigne sur le resserrement ou l'étalement des courbes de niveaux de la surface de réponse et nous permet d'ajuster le facteur d'échelle pour

mieux approcher un maximum. On se sert du fait que près d'un sommet, il y a généralement une diminution de la pente et un gradient qui tend vers zéro. La figure 2.1.2.1-1 nous donne l'algorithme simplifié et un exemple d'évolution sur une surface de réponse arbitraire. On peut voir le départ de p_0 et le maintien de la direction initiale jusqu'en p_y où on note une diminution de la fonction d'optimisation; on revient alors en p_x où on calcul le gradient et on repart dans sa direction. On notera que le pas s'aggrandit dans la région où les courbes de niveaux se resserent et diminue lorsqu'elles s'éloignent.

Il reste à déterminer comment on peut mesurer le gradient en pratique. On ne connaît pas la surface de réponse. On pourrait y aller par une approximation de la surface de réponse à l'aide de régression linéaire ou bien à l'aide d'équations aux différences pour approximer les dérivées partielles de "F" selon les variables de commande "X_i". La première façon est trop exigeante au niveau de la précision et de la rapidité de calcul. En plus, il est peu utile de rechercher une méthode précise pour obtenir le gradient à un point d'opération donné lorsque la mesure des différentes variables d'état du système est elle-même peu précise (car le système de mesure est très simple et peu coûteux). En plus, comme on corrigera si on se trompe de direction, il est acceptable de prendre une décision plus rapide et basée sur moins de données. Et, puisque la charge varie, l'évolution est souvent à recommencer; il vaut donc mieux se rendre près d'un point optimal rapidement car la surface de réponse risque d'être modifiée par une variation de charge... Pour ces raisons, nous n'avons retenu que l'utilisation des différences finies. Nous disions donc:

$$\frac{dF^p}{dX_i} \approx (DF_i/DX_i)|_p$$



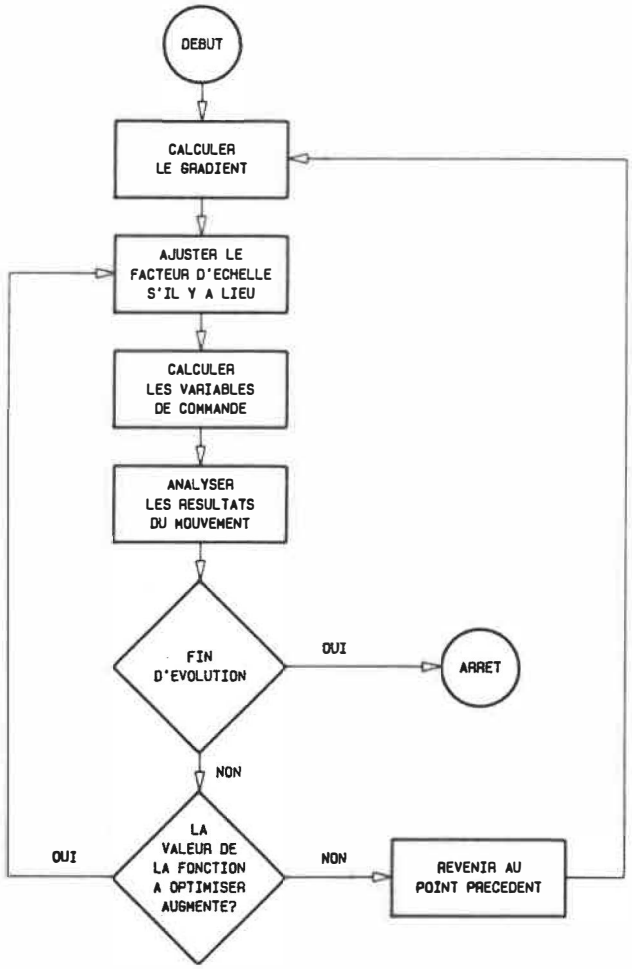


fig 2.1.2.1-1 TRAJECTOIRE D'EVOLUTION

où:

$\frac{dF^p}{dX_i}$ = la composante du gradient selon X_i au point p.

DF_i = la variation de la fonction "F" lors du changement de la variable de commande X_i.

DX_i = petite variation de la variable X_i.

Dans le cas où on a deux variables de commande, on se déplacera de façon à mesurer la valeur de la fonction aux quatre points d'un carré ayant comme centre le point p et comme côtés deux incréments de variable. Ce carré est illustré à la figure 2.1.2.1-2. On obtiendra la valeur des composantes du gradient au point p de la façon suivante:

$$\frac{dF^p}{dX_1} \approx \frac{(F^3 + F^2 - F^1 - F^4)}{2.DX_1}$$

$$\frac{dF^p}{dX_2} \approx \frac{(F^4 + F^3 - F^1 - F^2)}{2.DX_2}$$

C'est une approximation du gradient du premier ordre. Le désavantage de cette méthode est liée au fait qu'il n'y a pas de moyennage et qu'elle est donc sujette aux bruits ou aux perturbations. Pour pallier à cela, nous filtrons les mesures et nous nous assurons de ne rester à chacun des points que le temps nécessaire pour faire les mesures en régime permanent.

On a aussi décidé d'imposer des contraintes sur la plage d'opération des variables de commande car une erreur dans le choix de la fonction d'optimisation pourrait bien amener le moteur au décrochage; des contraintes sur les variables de commande permettront au système de se rattrapper ou du moins de ne pas s'arrêter. Deux types de

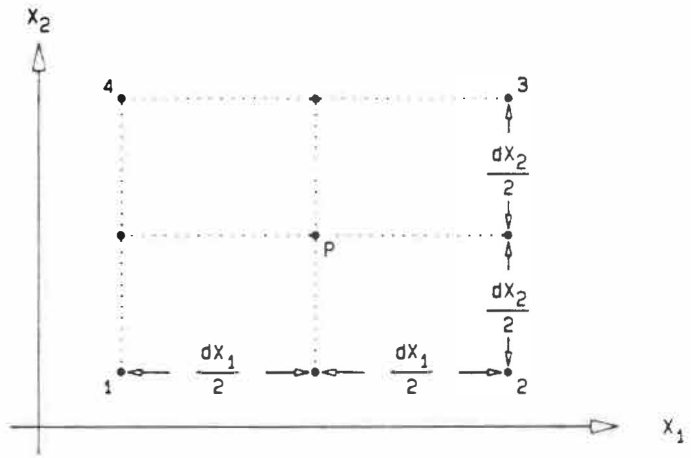


fig 2.1.2.1-2 MESURE DU GRADIENT AU POINT P

contraintes sont prévues: un type dit "doux" et agissant dans le calcul de la fonction d'optimisation de façon progressive plus on s'approche de la limite, et un autre type dit "dur" qui constitue une limite infranchissable. Le type "doux" permet au système de choisir un maximum qui ne pénalise pas sa fonction d'optimisation; le type "dur" est un signal d'alarme et une protection de dernier recours pour le système.

2.1.3 Utilisation d'un estimateur en boucle ouverte.

Comme nous l'avons vu à la section 2.1.1.2 et à la figure 2.1.1.2-7, on peut se servir de $V_{circuit}$ pour obtenir une estimation du couple de charge. Cette estimation peut être utilisée en boucle ouverte (sans vérification de la valeur réelle du couple et ajustements subséquents) avec des tables, pour nous donner la valeur de consigne de tension à imposer pour s'approcher d'un point d'opération performant (défini par les tables).

Par exemple, on pourrait déterminer expérimentalement la sortie $V_{circuit}$ à tension et charge nominales et l'ajuster à une certaine valeur normalisée à l'aide d'un gain réglable. On bâtirait aussi une table donnant la tension normalisée nécessaire pour maintenir la vitesse nominale avec la charge. Les consignes seraient ajustées de façon à s'approcher de ce point. On préconise une approche de type binaire: on corrige la consigne de la moitié de l'erreur entre cette dernière et la valeur calculée à partir de l'information des tables; cela permet une correction rapide au début puis de moins en moins forte plus on s'approche du point d'équilibre.

Une autre façon de faire serait d'employer les résultats d'évolution d'optimisation donnant la consigne de tension et de déphasage à un certain couple de charge donné. Ces valeurs seraient tabulées lors d'un fonctionnement en mode d'évolution d'optimisation; en pratique, pour obtenir une table bien garnie, il faut que le système soit excité par une série de variations de charge qui lui fait voir la plage complète de couples de charge et qui lui permet de tabuler les consignes pour la plage complète. Cela est rarement le cas. On devra imposer cette série de charge au système ou encore utiliser une approximation pour déterminer les consignes pour un couple non tabulé. Nous avons décidé d'utiliser une approximation linéaire pour combler les vides entre les valeurs tabulées.

Nous avons prévu l'emploi de la dernière méthode dans le logiciel de commande de l'entraînement; les détails pratiques sont donnés au chapitre trois. On notera que l'estimation nous permettra de régler selon le critère utilisé pour l'évolution et ne requiert plus de mesures à la source même si l'évolution en avait besoin; on peut réduire de beaucoup le système initial. Cette méthode de

réglage, quoique basée sur les résultats d'une évolution maximisant un certain critère de performance est limitée par la précision de l'estimation du couple. On devrait s'assurer de toujours se garder une certaine réserve, c'est-à-dire de conserver $V_{circuit}$ toujours un peu négatif. Une approche progressive des consignes tabulées aide aussi à stabiliser l'ensemble et permet de refaire plusieurs mesures avant d'atteindre un équilibre.

2.2 Les stratégies d'opération des interrupteurs.

Nous allons faire une brève étude de différentes stratégies que nous utilisons au convertisseur. Nous rappelons que le convertisseur n'utilise que des stratégies aux alternances symétriques et ayant un faible nombre d'événements. L'étude n'est pas exhaustive et constitue bien plus une présentation qu'une analyse détaillée.

Dans la dernière version du logiciel de commande du convertisseur nous avons mis en mémoire trois stratégies. Notons que cela n'est pas une limite du logiciel qui est très général et qui pourrait aisément gérer une banque de 256 stratégies. Mais, il est pratique de limiter la définition des stratégie à un seul ROM, car il n'y a qu'un seul élément à reprogrammer lorsqu'on apporte une modification quelconque aux tables des stratégies. En plus, trois stratégies semble être le nombre idéal: une pour chaque extrémité de la plage d'opération du convertisseur et l'autre en régime normal. Les stratégies en question permettent la réalisation de: un variateur symétrique, un variateur asymétrique à avance de phase, et un variateur symétrique à plusieurs commutations (16). Les deux premières stratégies sont bien décrite dans [9]. La

dernière stratégie est un essai pour réduire les pointes de courants observées dans la stratégie symétrique à faible tension. On part du fait qu'un étalement des zones de conduction sur l'ensemble du cycle, dans des régions à plus faible tension qu'autour de 90 et de 270 degrés, permettra de réduire la pointe de courant qu'on observe dans un symétrique à quatre événements.

Les figures 2.2-1 et 2.2-2 nous montrent le résultat de simulation de chacune des trois stratégies, à différentes amplitudes de tension statorique, alimentant un moteur asynchrone; la charge est la même dans les trois cas. A 90V, la stratégie symétrique à un courant de pointe de 20,6A tandis que pour celle à plusieurs commutations, le courant maximal est de 15,5A. Par contre, on y retrouve quatre de ces pointes comparativement à deux dans un symétrique. L'asymétrique par avance de phase a un courant de pointe de 21A mais présente la courbe de courant la plus douce (la plus uniforme et celle ayant des $\delta i/\delta t$ maximum les plus faibles).

Dans les sections qui suivent, nous voyons les logiciels de simulation utilisés et les critères de sélection minimaux pour une stratégie d'opération des interrupteurs du convertisseur.

2.2.1 La simulation.

Nous nous servons du modèle de la machine asynchrone donné à la figure 2.1.1.2-1. On établit le modèle d'état correspondant; les variables utilisées (les différents courants des enroulements du modèle) sont nommées "X_i" et l'entrée (la tension statorique directe) est appelée "U". On dérive un modèle en utilisant les valeurs entre

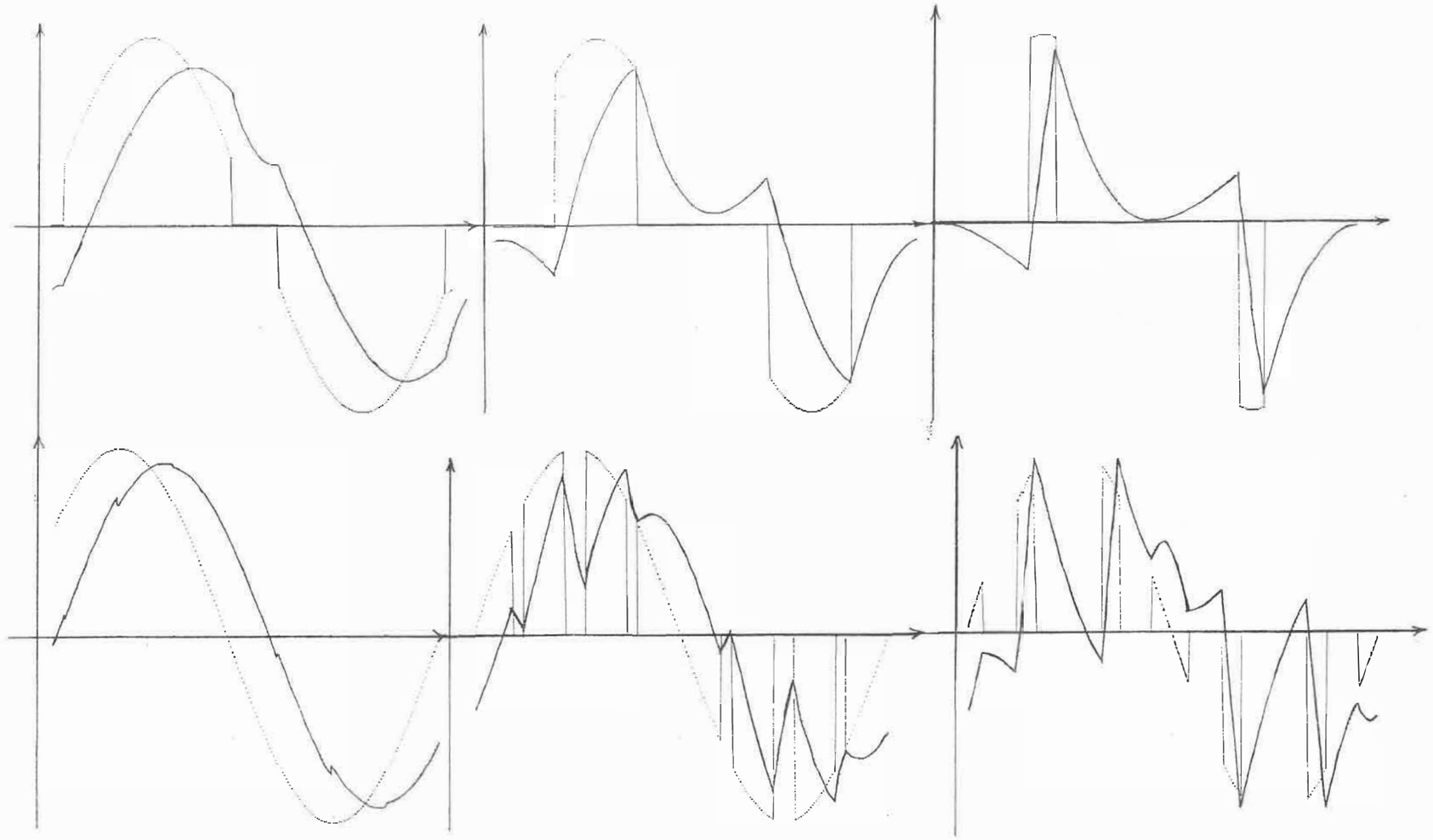


Figure 2.2-1 Formes d'ondes du courant et de la tension au moteur à 119V, 90V et 60V pour les stratégies symétrique et à plusieurs commutations

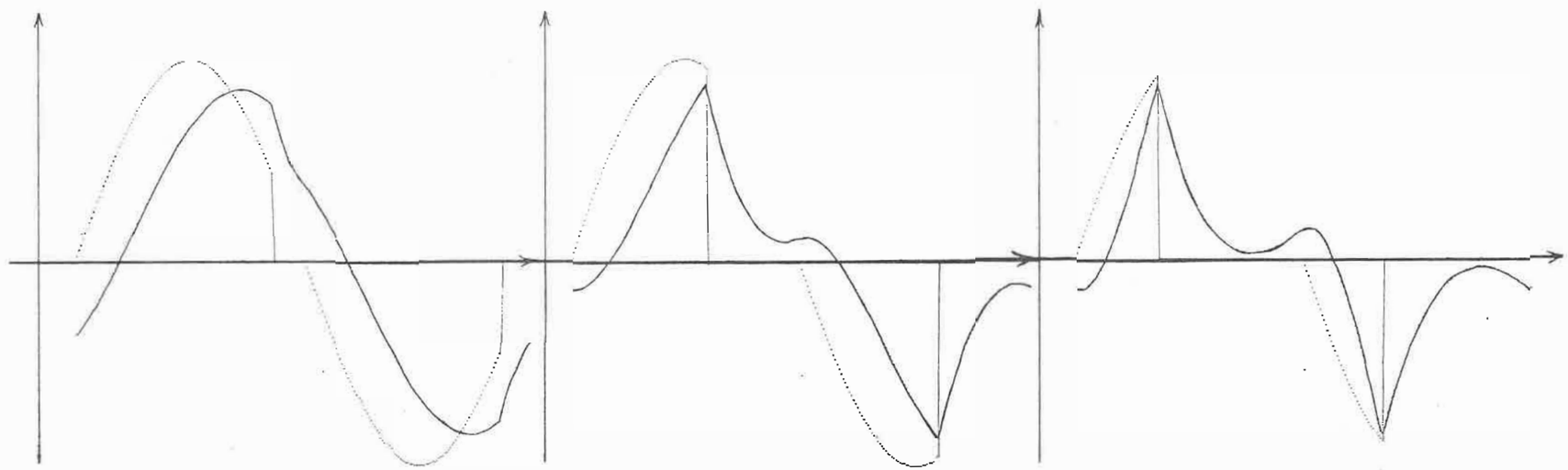


Figure 2.2-2 Formes d'ondes du courant et de la tension au moteur à 119V, 90V et 60V pour une stratégie asymétrique.

parenthèses sur la figure 2.1.1.2-2. Pour obtenir un modèle classique de la forme:

$$\dot{X} = A.X + B.U \ ,$$

les éléments de la matrice A doivent inclure l'effet des multiplicateurs et de la vitesse de rotation (voir la figure 2.1.1.2-2); certains des éléments sont alors dépendants de la vitesse. Une équation supplémentaire, non linéaire, permet de calculer la vitesse au moteur:

$$\dot{X}_4 = -K_6.X_4 - K_5.X_1.X_2 + T_{EXT}/J$$

X_4: la vitesse angulaire du moteur (ω);
T_{EXT}: un couple extérieur imposé par la charge;

Connaissant la vitesse, on peut rafraîchir régulièrement les éléments de la matrice A lors des simulations. Les matrices A et B du modèle sont:

$$A = \begin{vmatrix} -K_3/D & \omega.N.K_2/D & K_2.K_4/D \\ \omega.N.K_4 & -K_4 & \omega.N \\ K_3.K_4/D & -\omega.N/D & -K_4/D \end{vmatrix}$$

$$B = \begin{vmatrix} K_1/D \\ \emptyset \\ -K_1.K_4/D \end{vmatrix}$$

où:

ω = vitesse du moteur en radians par seconde.

$D = 1 - K_2.K_4$

K_i = définis sur la figure 2.1.1.2-2.

N = nombre de paires de pôles du moteur considéré.

X = vecteur d'état formé des X_i donné sur la figure 2.1.1.2-2.

U = Tension statorique directe au moteur.

A l'aide de ce dernier, nous évaluons l'évolution des variables d'état en fonction du temps avec l'entrée de tension telle que requise pour une certaine stratégie et une certaine amplitude désirées. Le logiciel s'appelle Simot*.bas et est donné en annexe B (*=1 pour le symétrique, *=2 pour l'asymétrique, *=3 pour celui à plusieurs commutations) La procédure très simplifiée du logiciel est la suivante:

- on sépare le cycle selon les intervalles définis par les instants d'arrivée des événements; on a "n+1" parties où "n" représente le nombre d'événements dans le cycle.
- pour chacune des parties, on sépare en divisions égales (et en nombre suffisants pour obtenir une bonne simulation) qui correspondent aux instants où on évalue les variables d'état et les sorties du système. Pour ce faire, on calcule l'intégrale de convolution entre la matrice de transition et le terme "B.U(t)"; cela nous permet de calculer les X_i:

$$X_i(t) = X_i(0) + \int_0^t \phi(t-\gamma).B_i.U(\gamma).d\gamma$$

On fait l'intégration numérique en séparant la division en 10 intervalles égaux et en évaluant le produit sous l'intégrale. Cela demande la connaissance de la matrice de

transition d'état $\phi(t)$ qui est calculée de la façon suivante:

$$\phi(t) = e^{A.t} \approx I + (A.t + \frac{A^2}{2!}.t^2 + ... + \frac{A^n}{n!}.t^n)$$

- A la fin du cycle, on recommence. Au bout d'un certain nombre de cycles (3), on réévalue la matrice A qui est fonction de la vitesse; la matrice aux valeurs rafraîchies nous permettra de toujours avoir un modèle correct lorsqu'on simule avec l'équation mécanique (couple extérieur, vitesse,etc.). Les résultats pour chaque division sont mémorisés et portés dans un fichier de sortie lorsqu'ils deviennent trop nombreux ou lorsqu'on arrête la simulation. A partir du fichier de données, on peut utiliser deux programmes post-traitant les résultats: un premier nous permet de tracer des formes d'ondes et le second de calculer la série de Fourier des variables de tension ou de courant.

2.2.2 les critères de sélection.

La stratégie d'opération des interrupteurs a un impact sur la sortie produite au convertisseur, sur les éléments commutants, sur la source et sur la charge. On peut grouper les critères de sélection d'une stratégie en deux types: ceux reliés au convertisseur et ceux reliés à la source et à la charge.

Les critères de sélection reliés au fonctionnement du convertisseur tiennent compte du stress subit par les interrupteurs et de la précision de l'amplitude de tension de sortie du convertisseur. Ce dernier point est influencé

par le comportement particulier du logiciel de commande ainsi que par le nombre d'événements d'une stratégie. Nous avons retenu comme critères: le courant maximal circulant dans un interrupteur et la précision du réglage de la tension. Les critères reliés à la source et à la charge sont basés sur le contenu harmonique de la tension et du courant. En fait, nous n'avons regardé que les valeurs à la charge.

Nous avons simulé pour des tensions de sortie de 60, 90, et 119V chacune des trois stratégies alimentant la même machine à la même charge. Le logiciel a aussi calculé les composantes harmoniques des courants et tensions de sortie du convertisseur. A partir de ces résultats et de la connaissance du fonctionnement du convertisseur, nous avons établi le tableau 2.2.2-1. On notera qu'au niveau de la précision du réglage, le logiciel de commande calcule ses événements en multiple de degré; plus il y a d'événements, plus l'effet de cette discrétisation se fait sentir. En plus, on ne peut pas exécuter deux événements successifs rapprochés de plus de 8°. Les stratégies qui répartissent les volts-seconde sur l'ensemble du cycle vont tomber dans cette limite aux extrémités des plages de réglage de la tension. Dans le cas de la stratégie symétrique, on note une difficulté d'obtenir de faibles valeurs de tension car on commute alors autour de 90° , donc près de la tension maximale, et qu'une imprécision ou un effet de la discrétisation entraînant une variation dans l'instant d'arrivée d'un événement entraîne aussi une variation très notable de la tension efficace de sortie du convertisseur. Au tableau 2.2.2-2, on présente les résultats d'une analyse fréquentielle à 90V. On peut observer une diminition des harmoniques de courant par rapport à celles de tension pour les harmoniques ≥5; le moteur agit comme un filtre. Ainsi,

	Symetrique	Asymetrique	Symetrique a plusieurs commutations
Courant maximal	Eleve	Moyen	Faible
Precision du reglage	Faible si tension < 50V Bonne si tension > 70V	Tres bonne aux extremites Moyenne en general	Moyenne aux extremites Bonne entre 05 et 50 volts
Harmoniques	Forte 3ieme	Moyenne 3ieme	Faible 3ieme
Facteur de deplacement	Nul	Avance de phase diminue avec l'augmentation de la tension	Nul

Tableau 2.2.2-1 Evaluation qualitative des 3 stratégies.

Strategie de commutation	Symetrique	Asymetrique	Symetrique a plusieurs commutations
Amplitude de la fondamentale			
1) de tension	96,6	108,9	96,33
2) de courant	9,97	11,40	9,92
Amplitude des harmoniques			
1) de tension			
Ordre 1	1,00	1,00	1,00
2	0,00	0,00	0,00
3	0,72	0,49	0,21
4	0,00	0,00	0,00
5	0,31	0,17	0,37
6	0,00	0,00	0,00
7	0,04	0,16	0,62
8	0,00	0,00	0,00
9	0,20	0,10	0,10
2) de courant			
Ordre 1	1,00	1,00	1,00
2	0,00	0,00	0,00
3	0,71	0,48	0,20
4	0,00	0,00	0,00
5	0,19	0,10	0,23
6	0,00	0,00	0,00
7	0,02	0,07	0,28
8	0,00	0,00	0,00
9	0,07	0,04	0,04

Tableau 2.2.2-2 Analyse fréquentielle à 90V pour les 3 stratégies.

on doit porter notre attention sur l'amplitude relative de la troisième harmonique car elle n'est pas atténuée et elle est assez élévée; la stratégie qui en génère le moins est plus intéressante car elle requiert un moins gros filtre. Le tableau 2.2.2-2 résume les constatations qu'on peut faire à partir des simulations et de la connaissance du système; on y donne aussi l'appréciation des stratégies en fonction des critères retenus.

Pour résumer, disons qu'on préférera utiliser la stratégie asymétrique lorsqu'on est aux extrémités de la plage de tension et surtout en basse tension à cause de son avance de phase intrinsèque qui compense en partie le retard du courant vu au primaire. La tension symétrique à plusieurs commutations est intéressante dans des tension moyennes et lorsqu'on a une charge qui varie brusquement, car cela provoque une augmentation du courant maximal aux interrupteurs; augmentation plus tolérable si la stratégie est efficace pour les maintenir faibles. La symétrique reste très intéressante à cause de sa simplicité et parce qu'elle a un déplacement intrinsèque nul; utilisée dans la demie plage supérieure de tension (60-120V), la pénalité introduite par son courant maximal élevé et sa grande troisième harmonique est atténuée. Dans un système simple, on n'aurait pas le choix entre plusieurs stratégies; dans cette optique, le choix doit se faire aussi en fonction du travail exigé normalement du moteur et du type d'interrupteur utilisé pour la commutation forçée.

CHAPITRE III

Description de l'entraînement.

Le but de ce chapitre est de présenter le système à microprocesseur qui asservit l'entraînement. Le système à microprocesseur regarde ce qui se passe au réseau et à la machine afin de déterminer les consignes requises pour réaliser le type de commande choisi par l'utilisateur. En plus, le microprocesseur gèrera l'interface usager utilisé pour communiquer de l'information, vers et du système, avec l'utilisateur; le convertisseur vu au premier chapitre est un élément de l'entraînement et est commandé par ce système pour réaliser un certain asservissement ou faire des tests sur la machine asynchrone lorsqu'on utilise le mode "manuel".

La caractéristique première de ce logiciel est la flexibilité: l'utilisateur choisit la structure de l'entraînement par l'intermédiaire d'un registre de commande et des fonctions qu'il entre au clavier. En plus, chacun des modules de haut niveau est contrôlé par un registre de commande qui définit les valeurs qu'il doit considérer lors de son exécution. L'exécution des modules est vérifiée et validée par un module de surveillance qui permettra d'avertir l'utilisateur de malfonctions et, dans les cas graves, de restructurer l'entraînement, tout ça de façon à permettre au progamme de continuer la commande en toute sécurité. Les asservissements possibles sont les trois types discutés au chapitre deux soient: la régulation de paramètre, l'évolution d'optimisation et le

fonctionnement à l'aide de l'estimateur en boucle ouverte. On a aussi implanté les fonctions nécessaires pour créer les tables pour l'estimateur; elles sont remplies (créées) à la fin de l'évolution d'optimisation pour une condition de charge donnée. Un mode de fonctionnment manuel est prévu: l'utilisateur entre les consignes qu'il désire au clavier.

Nous allons définir de façon précise le problème au point de vue du système de commande et en fonction des besoins et contraintes auxquels il doit répondre. Dans les sections qui suivent, nous donnons une description générale du logiciel et du matériel puis nous voyons les procédures plus en détails.

3.1 La définition du problème.

Le système à microprocesseur décrit ici fournit les consignes de tension de sortie, de déphasage par rapport au réseau et de stratégie d'opération des interrupteurs à utiliser au convertisseur. Les consignes proviennent directement de l'utilisateur (entrées au clavier) ou d'un asservissement; elles sont envoyées au microprocesseur de commande du convertisseur à l'aide d'un lien de communication parallèle (8 bits de données). L'interface entre l'utilisateur et le système permet la visualisation de l'information, relative à l'asservissement et à la chaîne convertisseur-moteur, ainsi que le transfert de commande de l'entraînement et sa programmation, c'est-à-dire l'établissement de sa structure: le mode de fonctionnement, la stratégie d'opération des interrupteurs, les valeurs à afficher et l'asservissement à utiliser s'il y a lieu. Les différents asservissemnets prévus sont:

- la régulation d'un paramètre,

- l'évolution d'optimisation d'un critère,
- l'utilisation de l'estimateur en boucle ouverte.

Pour les réaliser, on a besoin de mesurer certaines valeurs et de les traiter (facteur de conversion et filtrage) selon les besoins du logiciel et du matériel de conditionnement et d'interfaçage utilisés.

La communication entre l'utilisateur et l'entraînement se fait par l'intermédiaire du clavier HEX du SYM (pour les entrées) et par le terminal (pour les sorties); le tout étant géré par le logiciel de commande de l'entraînement qui vérifie régulièrement le clavier (utilisateur) et affiche les valeurs que l'utilisateur a demandées lors de la programmation de la structure de l'entraînement.

Les fonctions disponibles à l'utilisateur sont:

- arrêt immédiat,
- demande de restructuration,
- modification des consignes au convertisseur (donner une valeur ou bien l'augmenter ou la diminuer).

A l'écran, on affiche de façon régulière les valeurs demandées et on donne les messages d'erreurs ou d'avertissement prévus dans le logiciel.

La communication d'information entre le logiciel de commande de l'entraînement et celui du convertisseur a été décrite au chapitre premier. On se sert du lien de communication parallèle de huit bits de données. La communication est initiée par une interruption; le SYM1 est maître.

On a donc les fonctions suivantes à réaliser:

- communiquer les consignes au SYM,
- mesurer les valeurs,
- convertir les valeurs en fonction des gains dans la chaîne de mesure,
- filtrer les valeurs utilisées par l'asservissement,

- afficher les valeurs (mesurées, traitées ou de l'asservissement) demandées par l'utilisateur,
- vérifier les demandes de l'utilisateur (arrêt, modification à la structure de l'entraînement, modifications aux consignes envoyées au convertisseur),
- réaliser l'asservissement choisi par l'utilisateur,
- surveiller le fonctionnement du programme: valider les valeurs mesurées, filtrées et calculées, signaler les erreurs à l'utilisateur et restructurer l'entraînement pour permettre de continuer l'asservissement en attendant les réactions de l'utilisateurs.

Les entrées sont:
- le lien de communication,
- la sortie de la chaîne de mesure,
- les sorties des détecteurs de passage par zéro des fondamentales (de courant et de tension au moteur),
- le clavier hexadécimal du SYM.
- les signaux de dépassement de seuil des amplificateurs à gain programmable

Les sorties sont:
- le lien de communication,
- la commande de la chaîne de mesure et l'ajustement du gain programmable des amplificateurs,
- l'écran.

Les contraintes particulières:
- utiliser un SYM,
- commander le convertisseur du premier chapitre: il faut tenir compte du fonctionnement du programme de communication et de l'exécution du logiciel de réglage des consignes (temps de réponse, précision).
- matériel de faible coût,

- flexibilité; programmable.

Dans la section suivante, nous décrivons le système à microprocesseur répondant aux spécifications.

3.2 Description du système à microprocesseur.

Nous procédons en quatre parties: deux pour le matériel et deux pour le logiciel. Nous passons du général au détaillé.

Pour le matériel, la partie générale introduit les différents modules requis et spécifie les différentes fonctions matérielles requises pour les réaliser: on donne le schema bloc de chacune des grandes divisions du matériel. La partie plus détaillée entre dans la réalisation des circuits décrits plus tôt.

Pour le logiciel, on procède d'une façon semblable: une description générale décrivant les différents modules requis et leur organisation hiérarchique et une plus détaillée décrivant les fonctions logicielles et les procédures pour les réaliser.

On commence par la présentation des descriptions générales (du logiciel puis matériel) et puis on passe aux descriptions détaillées. Le but évident est de permettre au lecteur de se faire rapidement une idée du fonctionnement général et d'approfondir certains points s'il le désire.

3.2.1 Le logiciel.

Lors de la définition du problème, nous avons établi les fonctions générales que le système doit posséder. Nous irons plus loin en précisant les différentes fonctions, au point de vue du microprocesseur, ainsi que la philosophie d'ensemble.

Le module principal demande l'exécution séquentielle de modules supérieurs chargés d'une fonction de base dans le logiciel. Ces fonctions sont:

-les mesures à faire sur l'entraînement,

-la conversion de ces valeurs et leur filtrage,

-l'asservissement de l'entraînement,

-la gestion de l'interface de communication avec l'utilisateur,

-la communication entre les deux microprocesseurs.

Chacune des fonctions est exécutée par un module. Ces modules ont la caractéristique de faire un auto-diagnostique de leur fonctionnement; cela sert à un module de correction que le module principal appelle après le retour de chacun des modules supérieurs. Son rôle est de modifier la définition des registres de commande du logiciel selon le défaut rencontré. La définition de l'entraînement est basée sur celle des registres de commande; on modifie ces derniers pour permettre de continuer l'asservissement qui, autrement, devrait être arrêté. La communication entre les deux microprocesseurs est appelée par interruption. La signalisation entre le module de communication et le principal se fait à l'aide de drapeaux.

La structure des données est principalement basée sur l'emploie de vecteurs de valeurs mesurées, converties et filtrées, de drapeaux, de tableaux de données et de pointeurs. Des variables de travail dédiées par niveaux hiérarchiques et les registres A, X, Y à l'appel et au retour pour passer de l'information (le diagnostique au retour...) constituent le gros des moyens utilisés pour la communication inter-modules et la gestion des espaces de travail partagés. Ainsi, plusieurs modules de mêmes niveaux se voient associés les mêmes variables de travail mais ne

peuvent réécrire sur des registres utilisés par des modules inférieurs qui ont eux aussi leurs registres réservés; il faut revenir au module de niveau supérieur pour que les registres de travails inférieurs soient réutilisés par un autre module inférieur.

La procédure du logiciel est basée sur la définition des registres de commande suivant:

-RCDRIV,

-RCRITA,

-RCOMES,

-RCOMFL,

-RCAFFI.

Ils sont utilisés pour fixer les fonctions à exécuter ou les valeurs à utiliser. L'avantage évident est la généralisation du programme et la facilité de restructurer l'entraînement en redéfinissant quelques registres seulement (on peut changer l'ensemble des fonctions de l'entraînement par la modification de ces registres). Le microprocesseur peut reprogrammer l'entraînement en cas d'erreur: à l'aide d'auto-diagnostique au niveau des modules et d'un module de correction, on s'assure de modifier dynamiquement de ces registres pour permettre la survie de l'entraînement (on tient l'utilisateur au courant des modifications). L'utilisateur peut aussi le faire en tout temps; il peut restructurer l'entraînement à volonté: c'est une des fonctions prévues à l'interface avec l'utilisateur.

Trois asservissements sont prévus:

-Régulation d'un paramètre,

-Evolution d'optimisation,

-Avec l'estimateur de couple en boucle ouverte.

Ce sont les asservissements vus au chapitre 2.

Pour mesurer les signaux, on utilise une chaîne de

mesure avec convertisseur analogique-numérique en bout de chaîne. Pour certains signaux, on utilise des amplificateurs à gain programmable. Ces derniers devront être bien ajustés. De plus, des mesures de délais sont requises: cela demande du comptage. Une fois le matériel ajusté (par le logiciel) et les mesures faites, on doit les convertir pour se débarasser des différents gains dans la chaîne de mesure. Pour certaines de ces valeurs, du filtrage est nécessaire et réalisé. L'interface avec l'utilisateur est assez exigeant: le microprocesseur doit vérifier si l'utilisateur veut entrer une donnée ou un ordre au clavier et prévoir les différentes séquences d'interrogation et d'affichage pour bien orienter le transfert d'information. On doit aussi afficher régulièrement les valeurs demandées par l'utilisateur et les différents messages requis en opération.

Le diagramme hiérarchique de la figure 3.2.1-1 présente les différents modules logiciel. On note 6 modules supérieurs permettant de mesurer (MESURE), de traiter les valeurs (TRTSGN), d'exécuter l'asservissement demandé (ASSERV), de communiquer avec l'autre micro (CMSYM2), de gérer l'interface avec l'utilisateur (COMUSR) et de corriger les registres de commande pour prévenir la perte de l'entraînement (CORDRV). Pour chacun d'eux, une série de modules inférieurs permettre d'atteindre leur but respectif. Leur rôle sera précisé en 3.2.3.

3.2.2 Le matériel.

Le système à microprocesseur est construit autour d'un SYM. A cela se greffe le matériel nécessaire aux entrées/sorties de pour communiquer avec l'utilisateur (clavier et terminal). Il y a aussi le conditionnement des

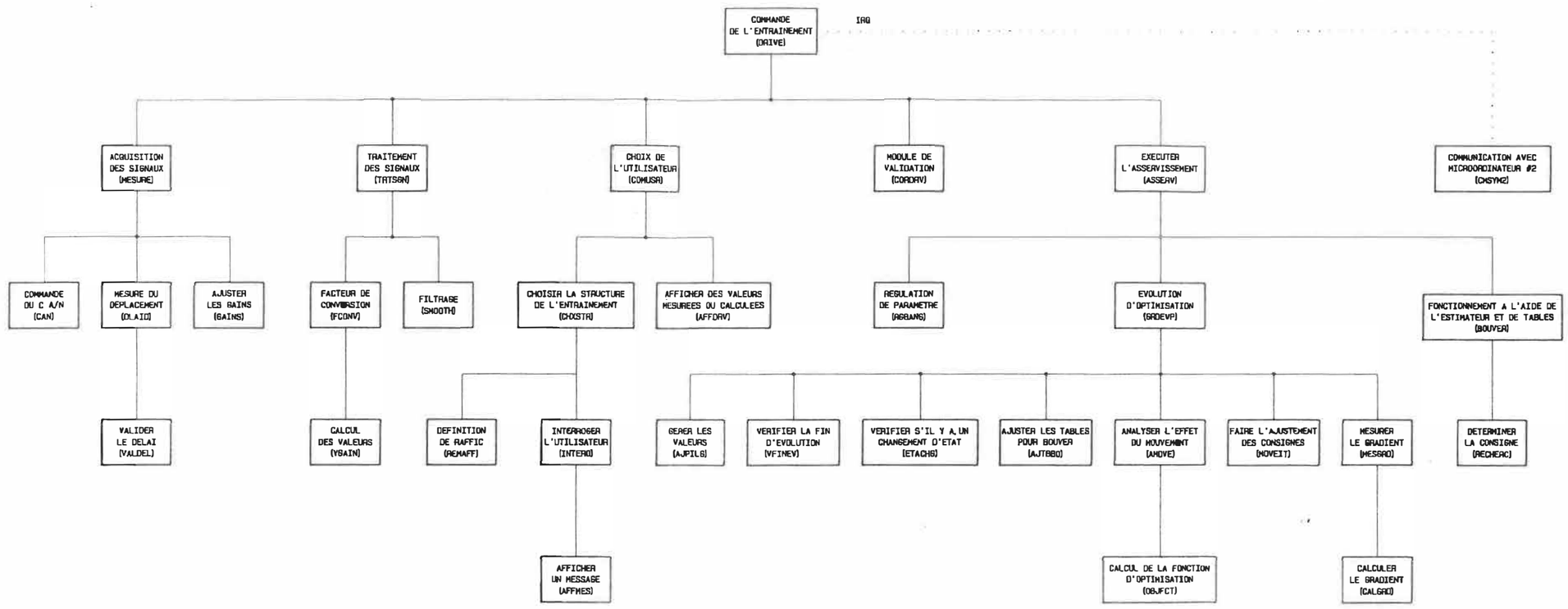


fig 3.2.1-1 DIAGRAMME HIERARCHIQUE DU LOGICIEL DE COMMANDE DE L'ENTRAINEMENT

signaux mesurés à l'entraînement et le matériel associé pour les rendre accessibles au microprocesseur. D'autres signaux voyagent sur le lien de communication de 8 bits de données.

3.2.2.1 Le module microprocesseur au sens large.

Cela regroupe le microordinateur SYM et son clavier hexadécimal, et l'écran. On en exclue les VIA utilisés pour l'acquisition externe au microordinateur.

3.2.2.2 Le module de conditionnement des signaux.

Les signaux qu'on capte à l'entraînement: le courant et la tension de source, le courant et la tension au moteur, la tension à l'enroulement de démarrage. Comme on peut le voir à la figure 3.2.2.2-1, de ces signaux on retire plusieurs valeurs qui sont amenées au multiplexeur anologique d'une chaîne de mesure et à des entrées de commande des VIA du système. La chaîne de mesure permet une mesure instantannée ou en vraie valeur efficace de la valeur sélectionnée au multiplexeur. Cette mesure est ensuite convertie en 8 bits et lue par le microprocesseur au moment désiré. La détection des passages par zéro des fondamentales de tension et courant moteur est envoyée aux entrées de commande pour permettre la mesure du délai (déphasage) entre leur transition montante respective. Regardons plus en détails comment on obtient les différentes valeurs au multiplexeur et aux entrées de commande à partir des signaux captés:

1)le courant Im: le signal de courant au moteur est capté à l'aide d'une sonde à effet Hall puis envoyé à un amplificateur différentiel (qui tiendra compte des légères

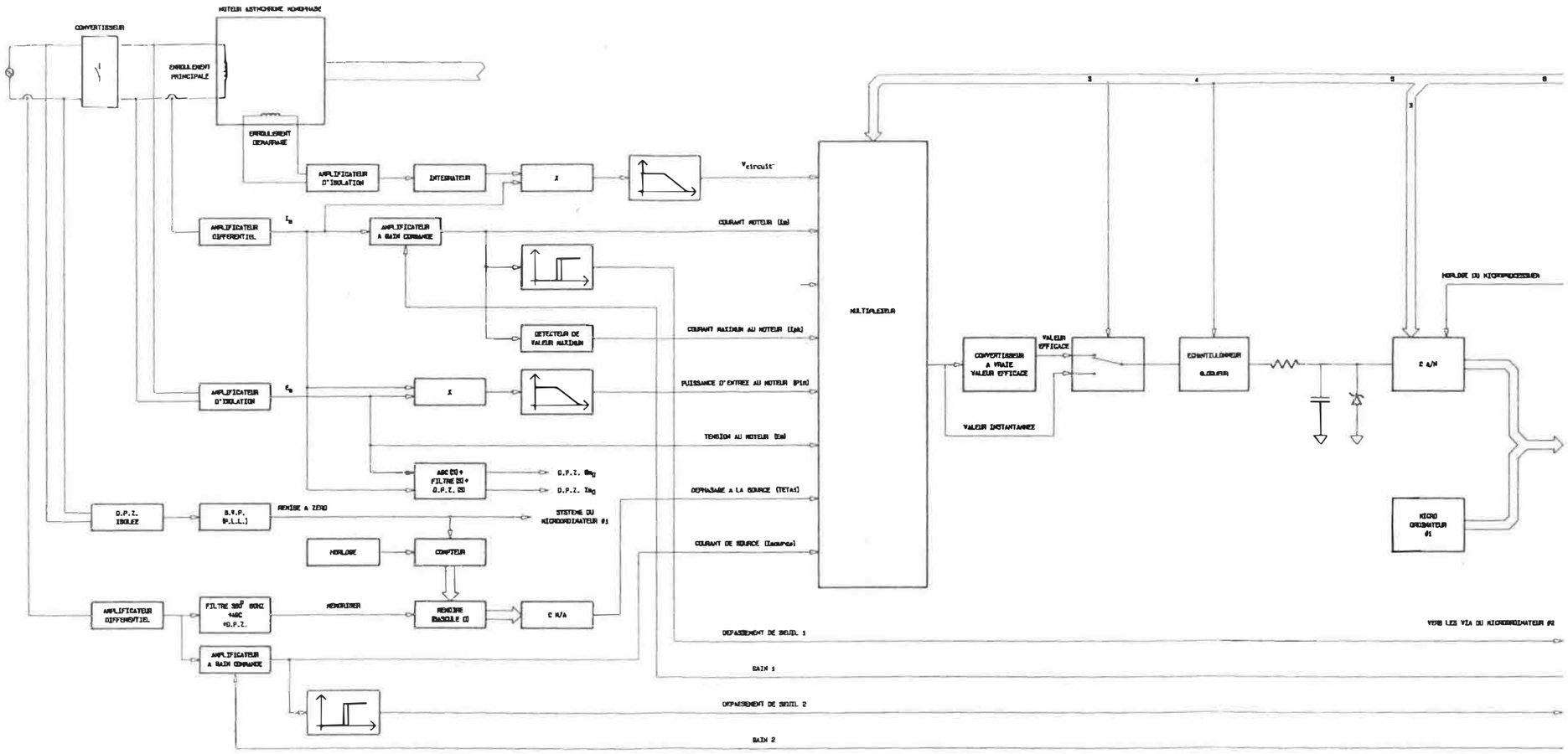


fig 3.2.2.2-1 CONDITIONNEMENT DES SIGNAUX D'ENTREE

différences entre les tensions de référence). Le signal Im est une image du courant moteur. Il sert à plusieurs endroits:

-à la mesure du courant maximal Ipk par un circuit de détection de la tension crête,

-à la mesure de puissance d'entrée Pin au moteur en calculant la valeur moyenne du produit du courant et de la tension d'entrées au moteur,

-à une détection du passage par zéro de la fondamentale de ce signal pour mesurer le déphasage au moteur par rapport à la fondamentale de la tension l'alimentant,

-à la mesure du courant au moteur de façon précise en envoyant le signal dans un amplificateur à gain programmable dont le but est de permettre une mesure dans les meilleures conditions au convertisseur analogique-numérique en maintenant l'amplitude de l'image du courant moteur entre 2 et 4 volts; on tiendra compte du réglage de cet amplificateur de façon logicielle pour obtenir la vraie amplitude de Im. Pour régler correctement le gain, on doit utiliser une détection de dépassement de seuil.

2) La mesure de la tension au moteur Em. On échantillonne la tension au moteur à l'aide d'un diviseur dont l'entrée légèrement filtrée est envoyée à l'amplificateur d'isolation. Comme le moteur n'est pas référé au même point commun que les circuits de mesure, on utilise un amplificateur d'isolation (utilisant une modulation AM à haute fréquence du signal mesuré à la référence du moteur qu'on couple magnétiquement du côté référé au point commun des circuits électronique à l'aide d'un transformateur d'isolation; le signal de sortie est démodulé) donnant l'image de la tension au moteur. Ce signal est utilisé à 3 endroits:

-à une entrée d'un multiplicateur analogique pour mesurer la puissance d'entrée au moteur,

-au multiplexeur analogique de la chaîne de mesure,

-à un circuit donnant le passage par zéro de la fondamentale du signal de tension au moteur. On passe en premier par un filtre passe-bas d'ordre élevé récupérant la fondamentale, puis par un AGC suivi d'un détecteur de passage par zéro. L'AGC est requis pour diminuer l'imprécision sur la détection du passage par zéro en éliminant les variations d'amplitude du signal sortant du filtre; on détecte avec la même erreur toujours. Le filtre est à retard connu.

3) La tension de source. On en retire une détection de passage par zéro isolée qui est envoyée à une boucle à verrouillage de phase (PLL) qui permet une perte instantannée du signal d'entrée sans affecter instantannément sa sortie; la PLL sert de mémoire temporaire en cas de perte de signal. La sortie de la PLL est envoyée au SYM1 et à un circuit de comptage pour mesurer le déphasage entre les transitions montantes des détecteurs de passage par zéro de la tension et du courant de source: ce qui vient de la tension relance le comptage et ce qui vient du courant de source permet de mesurer le compte. Le compte (8 bits) est renouvelé à chaque cycle et convertit analogiquement entre Ø et 5 volts puis envoyé au multiplexeur analogique comme signal proportionnel au déphasage à la source.

4)La mesure du courant de source Is: se fait comme la mesure du courant moteur. Ce signal sert à deux endroits:

-la mesure du courant Is lui-même en passant par un amplificateur à gain programmable associé à un détecteur de dépassement de seuil (voir la mesure de Im). La sortie de l'amplificateur à gain programmable est envoyée au

multiplexeur de la chaîne de mesure,

-à un filtre-AGC-DPZ qui sert à mesurer le déphasage entre les fondamentales de courant et de tension de source. La détection du passage par zéro de la fondamentale de courant fournit le signal de mémorisation du compte fait depuis la détection du passage par zéro de la tension.

5)La mesure de la tension induite à l'enroulement de démarrage. La tension passe par un amplificateur d'isolation (comme celui mesurant la tension au moteur) puis elle est intégrée et envoyée à un multiplicateur analogique (l'autre entrée est Im); la sortie est filtrée par un basse-bas dont la fréquence de coupure est très faible. La sortie du filtre est appelée Vcircuit et est envoyée à une entrée du multiplexeur anologique. Ce signal permet, entre autres, l'estimation du couple au moteur.

Les différents signaux ainsi recueillis sont envoyés au multiplexeur de la chaîne de mesure. Nous en reparlerons lors de la description du module d'interfaçage des entrées et sorties du système à microprocesseur.

3.2.2.3 L'interfaçage des entrées et sorties.

C'est essentiellement basé sur l'utilisation de 2 VIA dont les trois quarts sont disponibles sur le SYM. La figure 3.2.2.3-1 illustre les différentes entrées des VIA. L'utilisation des VIA est la suivante:

-un port pour les données du lien de communication et la sortie (8 bits) du convertisseur analogique-numérique (CA/N) de la chaîne de mesure;

-un port pour commander la chaîne de mesure;

-quelques lignes formant un bus de commande pour le lien de communication; le bus de communication comprend ces

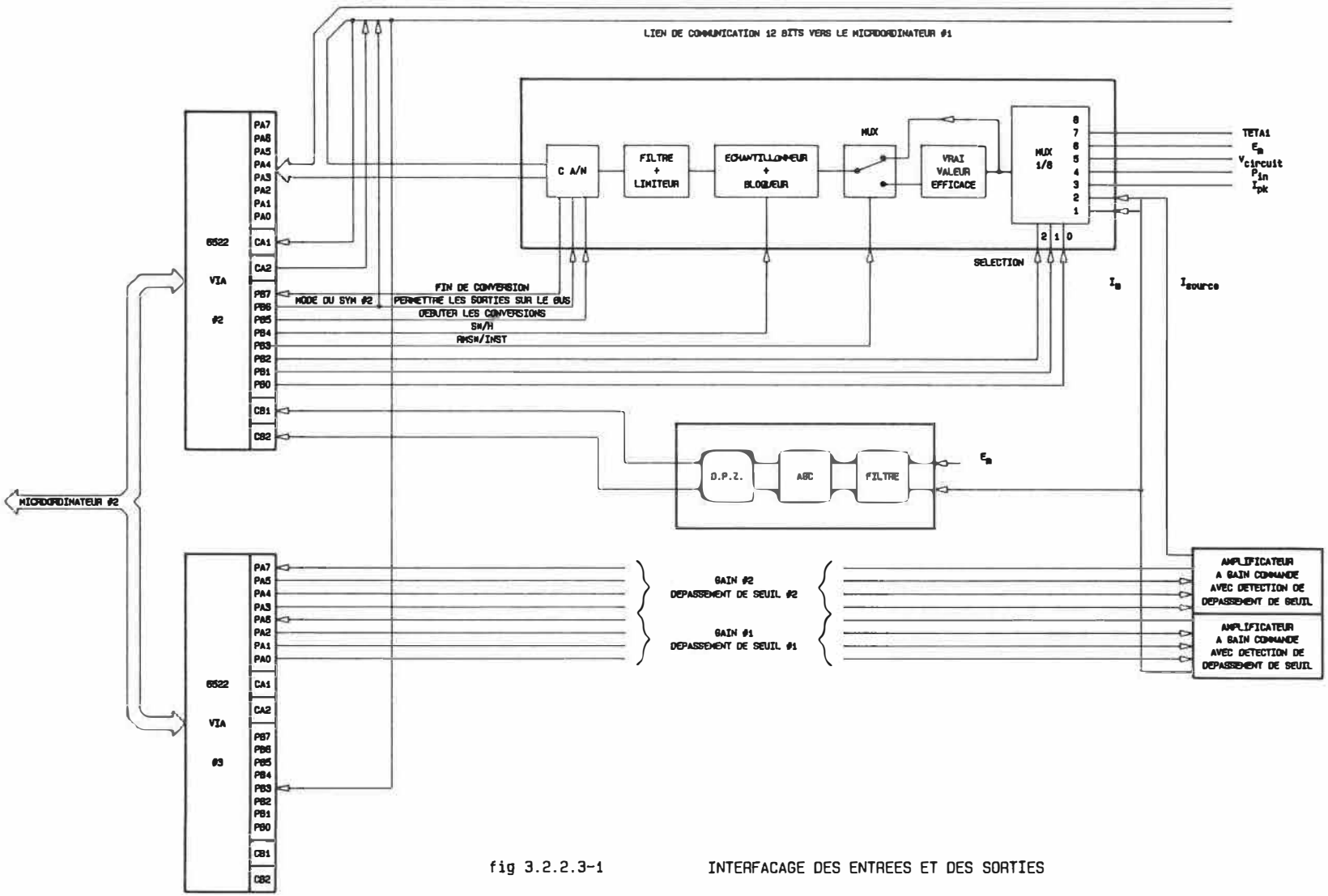


fig 3.2.2.3-1 INTERFACAGE DES ENTREES ET DES SORTIES

lignes et le bus de données relié à un port;

-un port pour la commande de deux amplificateurs à gain programmable;

-deux lignes de commande (CB1 et CB2 du VIA2) utilisées pour détecter les transitions montantes des sorties des détecteurs de passage par zéro (DPZ) des fondamentales du courant et de la tension au moteur; une horloge compte le délai entre les transitions montantes.

La chaîne de mesure est constituée d'un multiplexeur analogique, d'un convertisseur à vraie valeur efficace (RMS), d'un interrupteur sélectionnant entre la valeur RMS ou instantannée, d'un échantillonneur-bloqueur, d'un filtre-limiteur et d'un CA/N tels qu'illustrés en 3.2.2.3-1. Huit entrées sont disponibles au multiplexeur analogique; 7 sont utilisées: le courant au moteur (Im), le courant de source (Is), le courant maximal au moteur (Ipk), la puissance d'entrée (Pin) au moteur, le signal Vcircuit, la tension au moteur (Em) et le déphasage à la source (TETAls). Les signaux Im, Is et Em doivent être convertis en RMS, les autres valeurs sont des tensions continues. On utilise 3 bits (PB0, 1, 2 du VIA2) pour sélectionner une des huits entrées possibles du multiplexeur; la valeur sélectionnée correspond au nombre formé par les 3 bits. La sortie du multiplexeur est envoyée au convertisseur à vraie valeur efficace et à un sélecteur permettant de choisir entre la valeur RMS et la valeur instantannée provenant directement de la sortie du multiplexeur analogique. Une conversion RMS demande quelques périodes du signal d'entrée pour être précise (et au moins une pour la faire). On utilise 1 bit (PB3 du VIA2) pour faire la sélection entre RMS ou instantanné: 0 pour RMS et 5 volts pour une valeur instantannée. Pour éviter les fluctuations du signal qui peuvent entraîner une mauvaise conversion au CA/N, on

l'échantillonne et on bloque la valeur acquise. La valeur est "gelée" et envoyée vers le CA/N. Une ligne de commande permet de passer du mode échantillonnage (la sortie est égale à l'entrée de l'échantillonneur-bloqueur) au mode bloquage (la sortie n'est plus influencée par l'entrée et reste à la valeur où on a bloqué) en passant de 0 volt à 5 volts sur la ligne S*/H en PB4 du VIA2. La sortie passe par un filtre limiteur (limite à 5 volts et 0 volt) et entre dans un CA/N. Ce dernier utilise 3 lignes de commande:

-PB5 du VIA2 pour lancer la conversion (pulse 0-5-0 volt),

-PB7 du VIA2 pour signaler la fin de conversion (transition montante: 0-5volts),

-PB6 du VIA2 pour mettre la sortie sur le bus de huits bits (5 volts). Le bus du CA/N est relié au port A du VIA2.

Le lien de communication bidirectionnel utilise le port A du VIA2 pour recevoir ou envoyer les données. Les lignes de commande (handshake...) sont constituées de CA1 et CA2 (gèrent le transfert d'information en utilisant le mode de communication avec handshake du VIA) et de 2PB6 pour signaler le mode du bus au SYM2 (communication ou mesure) et de 3PB3 pour signaler le mode du SYM1 (communication ou mesure). Le SYM1 est maître de la communication; on ne peut mesurer au SYM2 que si le SYM1 ne demande pas la communication. Il ne le fera que si le bus du SYM2 est en mode communication (état normal). Nous verrons cela plus en détails dans la partie"logiciel".

Les deux amplificateurs à gain programmable reçoivent chacun 3 bits pour commander leur gain (8 valeurs) et émettent un bit par l'intermédiaire d'un détecteur de dépassement de seuil nécessaire à leur réglage. Le premier amplificateur se sert des lignes PA0, 1, 2 du VIA3 pour le

réglage du gain et de PA6 pour le détecteur de seuil. Le second amplificateur se sert des lignes PA3, 4, 5 pour le gain et PA7 pour détecter le dépassement de seuil associé. Les gains sont: 1/8, 1/4, 1/2, 1, 2, 4, 8 et 16 pour un réglage de: 0, 1, 2, 3, 4, 5, 6, 7 formé sur les bits 0, 1, 2 (3, 4 et 5) des lignes de réglage du gain.

3.2.3 Les modules logiciels.

Dans cette section, nous approfondissons la description du logiciel fonction par fonction. On décrit la fonction à réaliser , l'interaction avec le matériel, la structure des données, les principes et les approximations sur lesquelles repose la procédure. On donnera une description de la procédure des modules utilisés pour la fonction. Tout cela permettra de bien comprendre le code du logiciel.

3.2.3.1 La mesure des signaux.

On veut mesurer les valeurs indiquées par le contenu du registre RCOMES, le registre de commande des mesures. Ce registre est de 8 bits; chacun d'eux est associé à une valeur (Im, Em, Ipk, Pin, Vcircuit, Is, Tetals, Tetalm). Selon l'état (1 ou 0) du bit, la valeur sera mesurée (bit=1) ou non (bit=0). Pour mesurer la valeur, on doit en commander la sélection (au multiplexeur analogique) et l'acquisition (au reste de la chaîne de mesure). Une fois la valeur mesurée, on la mémorise dans la position correspondante du vecteur des valeurs mesurées VALMES. Pour la sélection, on utilise le tableau de multiplexage TABMUX donnant le code pour sélectionner la valeur au multiplexeur. Cette valeur est portée aux trois premiers bits du port B; le port commandant le multiplexeur et la

chaîne de mesure. L'acquisition se fera en modifiant les 5 MSB de ce port. On obtient l'ensemble des valeurs désirées en vérifiant l'état des différents bits de RCOMES. La figure 3.2.3.1-1 illustre l'ensemble de ce processus en mettant en évidence le rôle du RCOMES, des tables nécessaires et des ports de commande (port B) et de donnée (port A) de la chaîne de mesure. Pour faire les mesures, plusieurs modules, tel qu'illustré par le diagramme hiérarchique, sont utilisés:

-MESURE,

-CAN,

-DLAID et VALDEL,

-GAINS.

MESURE est le module principal se servant des autres pour réaliser la fonction avec le circuit d'interfaçage du système. Sa procédure est résumée par le figure 3.2.3.1-1. Ajoutons que lorsque des mesures RMS sont demandées, on lance la mesure du délai si cette mesure est à faire (selon RCOMES) ou une boucle d'attente dans le cas inverse. On mémorise les valeurs de délai dans une pile. Lorsqu'on a finit toutes les mesures, on vérifie si une mesure du délai est requise: si oui, on utilise les valeurs sur la pile s'il y en a (on fait la moyenne des mesures) ou on mesure le délai; si on n'arrive pas à la faire, on considère qu'on est en erreur et on le signalera dans l'auto-diagnostique. Le délai est compté. On se sert des lignes CB1 et CB2 (voir figure 3.2.2.3-1). Le VIA est programmé pour signaler les transitions montantes sur ces lignes; on compte entre deux transitions successives. Le pseudocode du module nous montre comment on réalise la mesure. Les autres mesures sont obtenus directement de la chaîne de mesure; le module CAN est utilisé. Le module GAINS sert à ajuster les amplificateurs à gain ajustable; on le fait s'il y a de

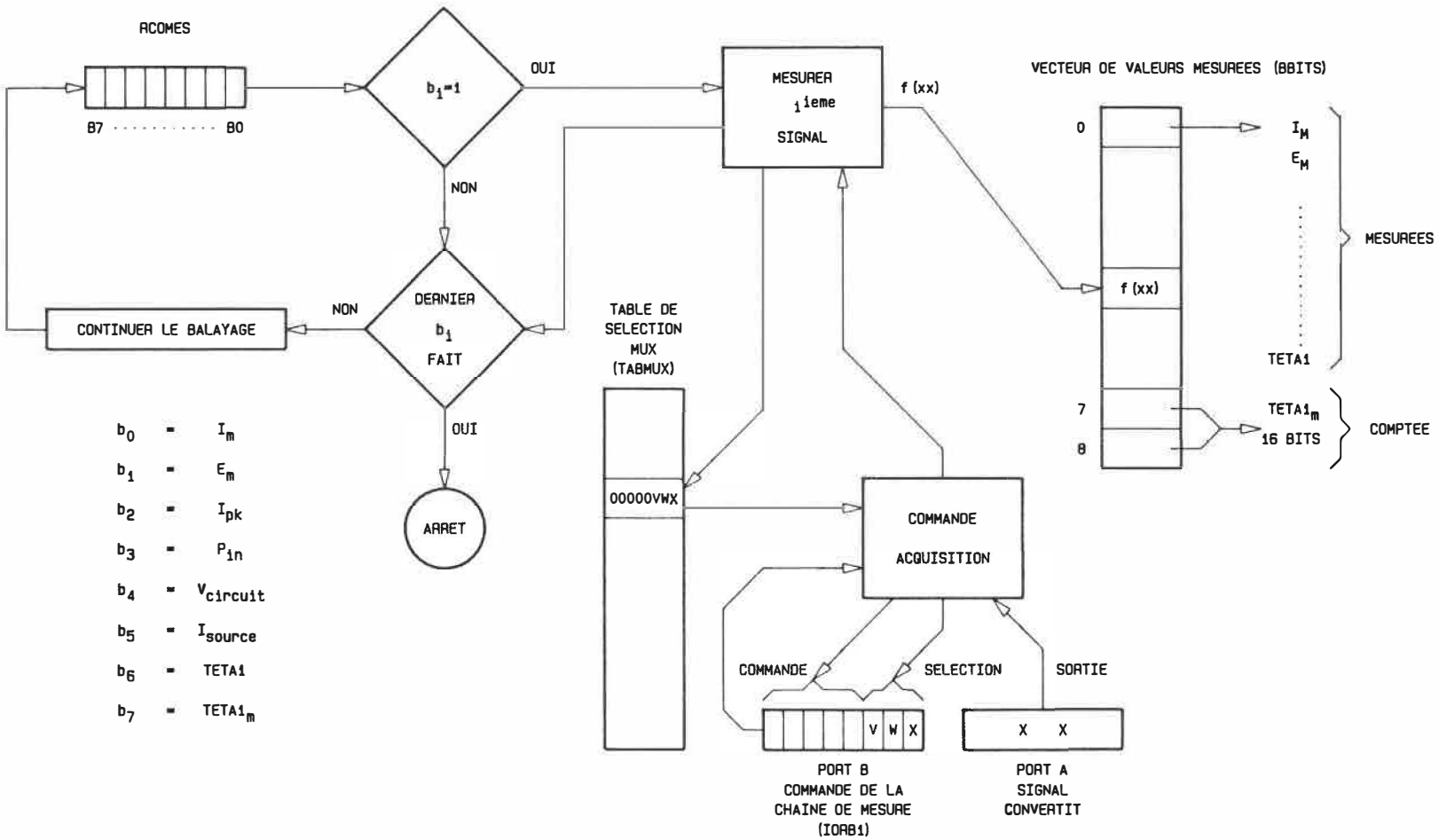


fig 3.2.3.1-1 LA MESURE DES SIGNAUX ET "RCOMES"

grandes variations ou un changement au détecteur de dépassement de seuil. Le pseudocode schématique et le code de chacun des modules abordés ici sont donnés en [3].

3.2.3.2 Le traitement des valeurs mesurées.

Si on regarde le diagramme hiérarchique (figure 3.2.1-1), on note qu'il y a deux modules importants définissant les deux parties du traitement des signaux soient: l'application de facteur de conversion aux valeurs mesurées et le filtrage de valeurs converties. En effet, la première partie permet de tenir compte des gains dans les chaînes de mesures de chacune des valeurs et la seconde permet de filtrer les valeurs qui subissent des perturbations cycliques ou aléatoires ainsi que l'effet de discrétisation lors de la conversion analogique-numérique. Le filtrage est de la forme:

$$\text{Sortie} = \Sigma(Ci.Vi)/\Sigma Ci, \quad i = 0 \text{ à } \text{ORDRE} - 1$$

où:

Ci = le coefficient associé à la valeur "i"

Vi = les différentes valeurs accumulées; les "ORDRE" dernières valeurs sont conservées dans une pile LILO.

On peut choisir l'ordre du filtre associé à chacune des différentes valeurs traitées. On notera que ce type de filtre devient une moyenne si tous les coefficients Ci sont unitaires. En fait, on obtient:

$$\text{Sortie}(z) = V(z).C(z)$$

$$\text{Sortie}(n) = Cn.Vn + Cn\text{-}1.Vn\text{-}1 + \ldots + C1.V1$$

Et on choisit la fonction de transfert pulsée du filtre numérique C(z) en déterminant les Ci. On doit tenir compte de l'effet de ce filtre sur la boucle de régulation; les Ci et l'ordre du filtre dépendront des constantes de temps du

système asservi et de la fréquence d'échantillonnage. Les signaux à traiter sont identifiés par le contenu du registre de commande de filtrage RCOMFL. Ce registre sert à identifier les valeurs qu'on doit convertir et filtrer. On commence par convertir les valeurs mesurées, puis on les filtre. La figure 3.2.3.2-1 nous montre le principe utilisé (un balayage des bits du registre et une conversion de la valeur correspondante pour ceux qui sont à "1" et les différents tableaux servant au processus. On voit bien la partie balayant le registre; si on trouve un bit = 1, on applique le facteur de conversion: on multiplie la "ième" valeur mesurée par le facteur de conversion fixe de la chaîne de mesure (dans le tableau FCFIXE en position i) puis, s'il y a un amplificateur à gain commandé dans la chaîne, on en tient compte par l'exécution du module YGAIN (multiplie le résultat obtenu jusqu'ici par le gain commandé obtenu à l'aide du contenu de GAINSX et du tableau de conversion CVGAIN). La valeur obtenue est mémorisée dans le vecteur des valeurs vraies VALTRU comme "ième" valeur.

La figure 3.2.3.2-2 nous montre le déroulement de l'étape suivante qui est le filtrage. On procède encore par un balayage du RCOMFL. Lorsqu'on trouve qu'une valeur est à filtrer, on remplit le vecteur des paramètres pour le filtre associé à l'aide du contenu du tableau de paramètres des filtres TBPARF. On a alors l'ordre du filtre, les adresses de la table des coefficients TBCOEFi et où mémoriser les valeurs converties VALUEi afin d'effectuer le calcul de la sortie du filtre. Le résultat devient la valeur "i" du vecteur des valeurs filtrées VALSMT.

Le pseudocode et le code des modules TRTSGN, SMOOTH, FCONV et YGAIN sont donnés en [3].

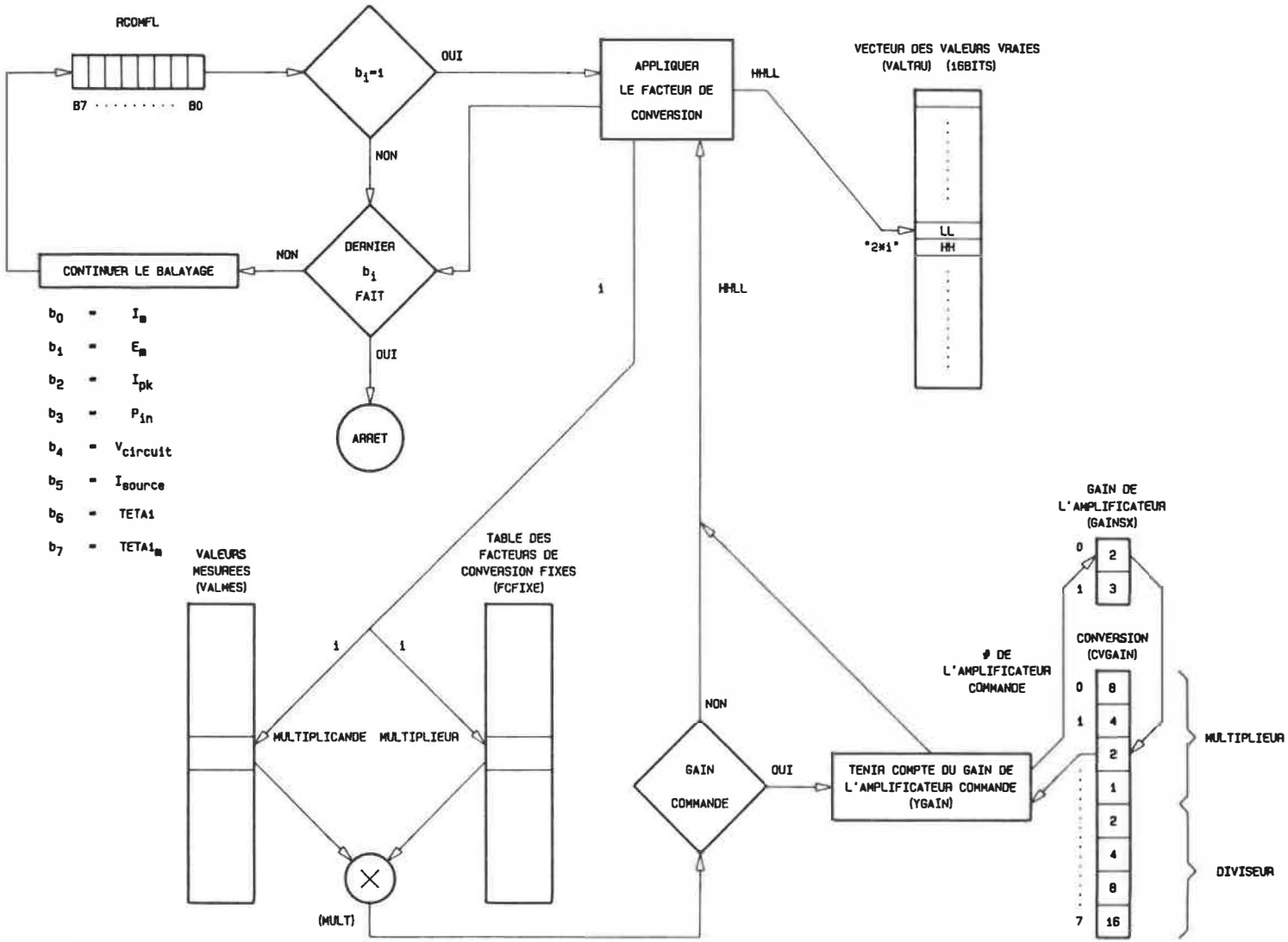


fig 3.2.3.2-1 LES FACTEURS DE CONVERSION ET "RCOMFL"

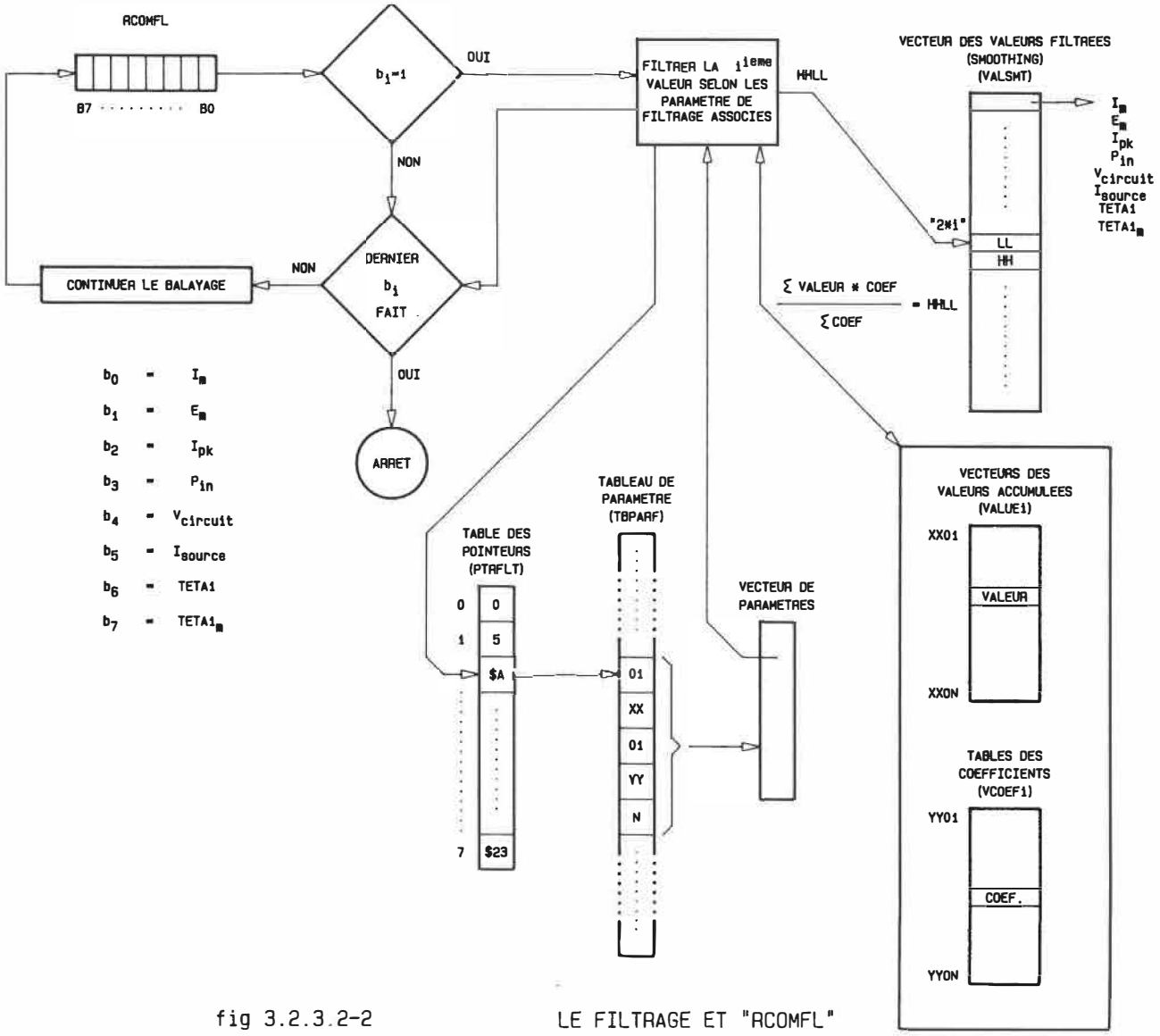


fig 3.2.3.2-2 LE FILTRAGE ET "RCOMFL"

3.2.3.3 L'asservissement.

Comme on peut le voir au diagramme hiérarchique à la figure 3.2.1-1, le module ASSERV exéxutant l'asservissement a le choix entre trois types:

-l'évolution d'optimisation avec le module GRDEVP,

-la régulation de paramètre avec le module RGBANG,

-une opération en boucle ouverte, à l'aide de l'estimateur de couple et de tables, avec le module BOUVER.

On pourrait aussi fonctionner en mode manuel (pas d'asservissement): le module ASSERV n'exécute aucun des module d'asservissement. Ce mode est utilisé pour mettre au point certaines fonctions logicielles ou pour tester certaines stratégies d'opération par exemple.

Le choix de l'asservissement se fait à l'aide du registre de commande de l'entraînement RCDRIV. Nous verrons comment à la section 3.2.3.4 où on traite du choix de la structure de l'entraînement entre autres. Le pseudocode [ ] du module ASSERV nous montre sa procédure: si on fonctionne avec un asservissement, on appele le module correspondant (GRDEVP, RGBANG, BOUVER) qui l'exécutera. Dans le cas particulier d'une régulation (RGBANG), on vérifie si on tient compte de ce que voit la source (déplacement, facteur de puissance...); si oui, on régule le déplacement à la source avec une boucle très lente. Les trois asservissements ont été vus théoriquement au chapitre deux; nous allons voir comment ils fonctionnent réellement.

La partie de l'auto-diagnostique du module ASSERV donne toujours "aucune erreur" pour l'instant; c'est encore à faire.

3.2.3.3.1 Régulation d'un paramètre.

Pour régler un paramètre, on se sert du module RGBANG. Le paramètre est calculé à partir d'une ou plusieurs mesures et est choisi par la sélection de coefficients associés à la somme d'une série de valeurs. Nous verrons cela dans la section traitant de l'interface avec l'utilisateur et plus particulièrement du module CHXSTR qui permet le choix du critère ou paramètre de régulation.

La régulation se fait à peu près comme on l'a dit à section 2.1.1 mais au lieu de découpler l'effet de la consigne de déphasage sur la tension de sortie, on ralentit la boucle utilisant le déphasage pour régler le facteur de déplacement à la source et on n'utilise pas le pré-découpleur. Cela fait apparaître une perturbation dans la boucle de régulation de tension. Nous avons donc ralenti la boucle de régulation du facteur de déplacement à la source permettant à celle régulant à l'aide de la tension de combattre cette perturbation à basse fréquence. On considère l'amélioration du facteur de déplacement comme complémentaire; l'important c'est d'abord d'améliorer les performances de la machine par l'ajustement de la tension puis, améliorer ce qu'on en voit de la source. Comme le couplage est généralement plus faible que 1, on arrive à compenser son effet dans les différentes boucles de régulation.

Le régulateur utilisé dans les boucles est de la forme illustrée à la figure 3.2.3.3.1-1. Il s'agit d'un régulateur de type intégrateur à gain variable: tant que l'erreur n'est pas dans la zone morte, la consigne est modifiée par le delta consigne calculé selon l'erreur et limité par une valeur maximale permise DCSNMX; dans la zone morte, le gain tombe à 0 et la consigne n'est plus

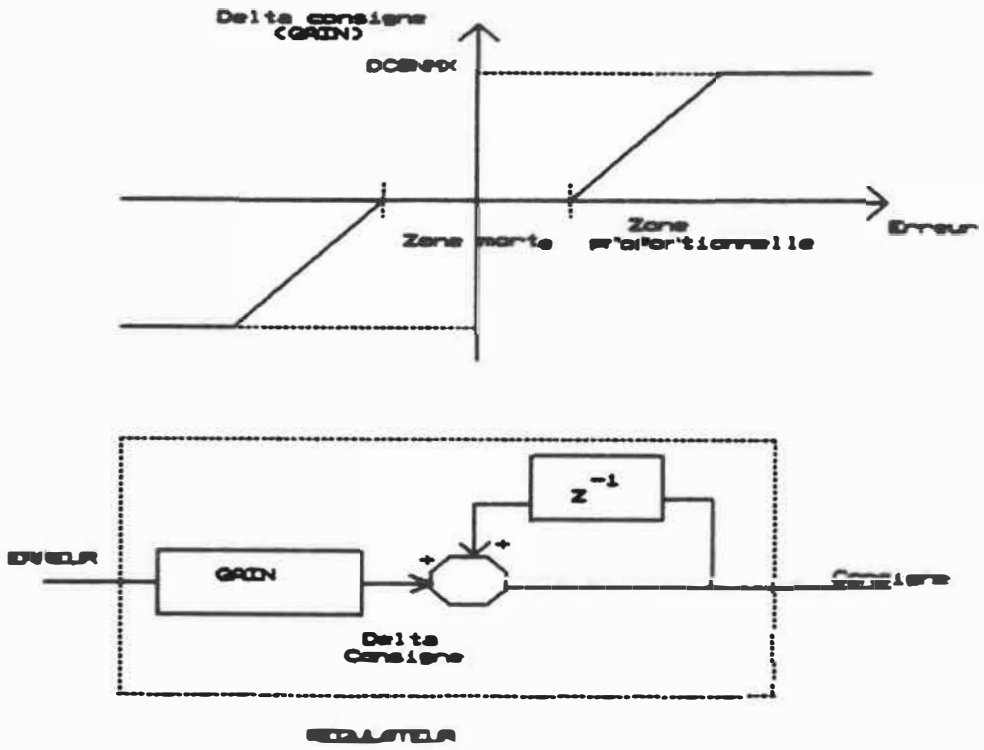


Figure 3.2.3.3.1-1 Le régulateur.

modifiée. La grandeur de la zone morte est choisie pour tenir compte des imprécisions dues à la discrétisation ainsi que des erreurs d'arrondis dans les différents calculs nécessaires à l'asservissement. Le module RGBANG utilise un vecteur dit de régulation VECREG qui contient l'information relative la boucle réalisée:

-la zone morte,
-le gain dans la zone proportionnelle,
-le delta consigne maximal,
-le signe du gain du procédé (signe de out/in),
-la valeur de consigne,
-le paramètre à réguler,
-la consigne à modifier (tension ou déphasage).

Ces valeurs sont obtenues d'un tableau de paramètres de régulation; il y a un jeu pour chacun des paramètres prévus actuellement.

Pour choisir le delta consigne maximal et les autres paramètres du régulateur, on doit considérer la boucle complète du système: du calcul des consignes en passant au microordinateur les exécutant jusqu'à l'acquisition dans la chaîne de mesure des valeurs. On doit faire attention à:

-le délai dans la chaîne de mesure,
-la précision et la conversion analogique-numérique dans la chaîne de mesure,
-le délai rajouté à cause du filtrage des valeurs,
-le temps de réponse du microordinateur commandant le convertisseur pour atteindre les consignes demandées,
-le temps de réponse du moteur à une variation de consigne.

La figure 3.2.3.3.1-2 nous montre ces différents éléments dans la boucle de régulation.

La mesure d'une valeur efficace (RMS) demande plusieurs périodes du signal à mesurer lorsque sa forme est inconnue.

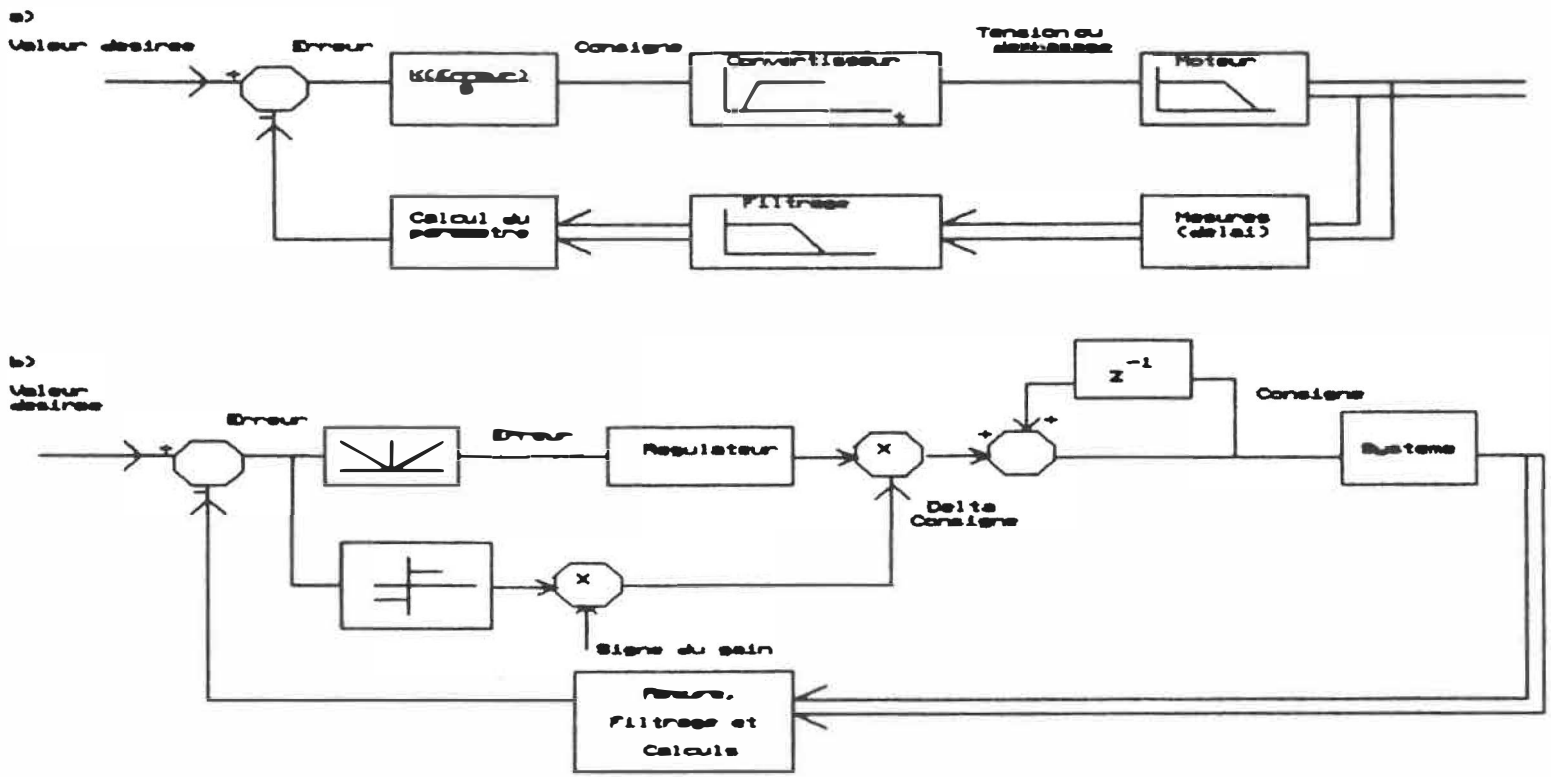


Figure 3.2.3.3.1-2 La boucle de régulation
a)Eléments à considérer.
b)En pratique.

Pour une mesure précise, avec le AD536 (convertisseur à vraie valeur efficace) dans la chaîne de mesure, on doit attendre près de 10 cycles. La mesure du délai requiert 2 cycles au maximum (attente de la bonne transition et comptage sur le cycle suivant). Dans le pire des cas, on a 7 mesures à faire avec la chaîne: 7x10 cycles + 2 cycles soit 72 cycles à 60Hz. C'est environ 1 seconde. Ces valeurs doivent être filtrées. Si on utilise un filtre du 8 ième ordre, on doit prendre 8 jeux de mesures pour obtenir la valeur à l'instant de sortie; cela demande près de 9 secondes. Par rapport à cela, le temps de calcul au régulateur est très négligeable.

Le microordinateur commandant le convertisseur (SYM1) progresse d'environ 1Volt/cycle; ce n'est pas rapide, mais c'est largement suffisant pour passer de 0 à 120V entre deux commandes successives fournies par le microordinateur calculant les consignes à l'aide des mesures.

Le moteur à une constante de temps mécanique qui dépend surtout de la charge qu'il entraîne; on peut supposer que c'est de l'ordre de la seconde pour un moteur d'une fraction de HP. Le SYM1 et le moteur atteignent donc leur valeur finale (déterminées par leurs consignes) avant que le second microordinateur ait recalculé de nouvelles consignes. On ne peut pas utiliser un régulateur linéaire à grand gain pour minimiser l'erreur en régime permanent car les délais de mesure sont trop grands: le gain fois l'erreur demanderait la consigne maximale (ou minimale) dès que l'erreur ne serait plus nulle et le moteur aurait le temps d'atteindre la valeur finale avec cette consigne avant qu'on ait un jeu pour calculer une nouvelle consigne qui demanderait de diminuer la consigne; on oscillerait entre la valeur maximale et la valeur minimale. Il y a deux façon d'y remédier:

-retarder le processus (chaîne convertisseur-moteur) qui exigerait de nouvelles consignes, donc de mesurer-filtrer pour calculer une nouvelle consigne, après chaque mouvement incrémental;
-utiliser un très faible gain pour un régulateur de type intégrateur et lui imposer un delta consigne maximal. Cela permet d'imposer des variations de consignes faibles qui ne provoqueront pas de grandes oscillations, tout au plus un genre de cycle limite de faible amplitude.

Nous avons utilisé la deuxième possibilité car elle minimise les interactions entre les deux microordinateurs et donc le nombre de communications entre eux. Le SYM1 doit rester maître des communications car il ne peut communiquer que dans les fenêtres de temps laissées libres par la commande en temps réel du convertisseur; cela ne se produit pas à tous les cycles. En plus, si on occupe beaucoup le lien de communications, on ne peut pas effectuer les mesures car le bus de données pour la communication est partagé avec celui de mesure (multiplexé) et ce aux deux microordinateurs.

Le module RGBANG exécute donc l'asservissement de la figure 3.2.3.3.1-1. Le pseudocode schématique et le code du module sont donnés en [3].

3.2.3.3.2 Evolution d'optimisation.

Le module GRDEVP exécute l'asservissement d'optimisation par évolution selon la méthode du gradient donnée à la section 2.1.2.1. La figure 2.1.2.1-1 donne un aperçu de l'évolution. Sur le diagramme hiérarchique à la figure 3.2.1-1, on remarque plusieurs modules reliés à GRDEVP qui correspondent aux différentes étapes de la figure 2.1.2.1-1:

-OBJFCT: sert à calculer la valeur de la fonction à optimiser au point d'opération courant.
-AMOVE: se sert du modula OBJFCT; il analyse l'effet du pas précédent (optimise-t-on?).
-VFINEV: vérifie si l'évolution est finie.
-MESGRD: permet de mesurer le gradient au point d'opération courant; se sert de CALGRD.
-CALGRD: calcule le gradient à partir des mesures faites autour du point d'opération.
-AJPILE: gère une pile LILO qui retient dans un registre à décalage les n dernières valeurs; cela servira à détecter s'il y a fin d'évolution.
-ETACHG: lorsque l'évolution est terminée, on opère à un point optimal selon le critère. Ce module sert à vérifier si on doit recommencer l'évolution en comparant l'état actuel à l'état de fin d'évolution.

La mesure du gradient nécessite 5 étapes qui permettent de se déplacer aux 4 points formant un carré dont le centre est le point d'opération actuel. Une dernière étape permet le retour au point original. A chacun des 4 points, on mesure la valeur de la fonction à optimiser (F) et donc on obtient les variations de F en fonction de chacune des consignes; on trouve le gradient.

La plupart des états (en évolution ou non, en mesure du gradient ou non, en début d'évolution ou non) sont signalés à l'aide de drapeaux. Des compteurs signalent l'étape de mesure du gradient, si les dernières consignes ont servi et si le filtre a accumulé au moins ORDRE valeurs.

Le pseudocode de GRDEVP est donné en [3]. Pour les autres modules impliqués dans cet asservissement, nous allons donner quelques précisions qui permettront de mieux apprécier leur procédure décrite par leur pseudocode.

La vérification de la fin d'évolution (VFINEV) se sert

des piles LILO. A la fin d'une évolution, les consignes dans les piles devraient être à peu près constantes, oscillant à ±pas de la valeur optimale. La valeur moyenne des valeurs accumulées sera donc à ±pas de la consigne actuelle; c'est la condition de fin d'évolution. Cela permet d'arrêter l'évolution (éviter les cycles limites inutiles) et bâtir les tables donnant les consignes optimales selon un critère donné en fonction de la charge estimée pour fonctionner en boucle ouverte (BOUVER). Si la vérification s'avère positive, on mémorise le point d'opération actuel (consignes et valeur de la fonction à optimiser) qui servira à vérifier si on doit recommencer l'évolution à cause d'un changement du point d'opération.

C'est le module ETACHG qui fait cela: il vérifie si les consignes ont varié de plus d'un pas ou si la valeur de la fonction à optimiser calculée au point d'opération courant a changé de plus de la somme des composantes du gradient par rapport à sa valeur de fin d'évolution. Dans le cas affirmatif, on considère que l'état a changé: c'est comme si on s'était déplacé d'un pas ou plus par rapport au point de fin d'évolution.

Le gradient est mesuré quand on recommence l'évolution ou si on note un changement dans la progression de la valeur de la fonction à optimiser (changement du signe de la variation de cette fonction). Le gradient a autant de composantes que de variables de commandes sont utilisées pour l'évolution. La méthode est bien expliquée au chapitre 2 et par le pseudocode du module MESGRD. Notons que lors de l'évolution avec une seule variable de commande (la tension), on mesure quand même en 5 étapes qui nous donnerons deux valeurs à ±pas unitaire du point d'opération actuel.

Le calcul du gradient se fait à l'aide des valeurs de la

fonction à optimiser mesurées à chacun des points tel que décrit au chapitre 2 et au pseudocode du module CALGRD.

L'ensemble des pseudocodes de modules reliés à l'évolution d'optimisation permet une compréhension plus détaillée de la mise en pratique de cet asservissement; ils sont en [3].

3.2.3.3.3 Fonctionnement en boucle ouverte à l'aide de l'estimateur de couple et de tables.

Le diagramme hiérarchique nous montre deux modules impliqués dans le fonctionnement en boucle ouverte: BOUVER, le module principal et RECHER le module servant à déterminer les consignes à partir de l'estimation du couple. On doit aussi rajouter AJTBBO qu'on voit dans les ramifications de GRDEVP. Ce module bâtit les tables dont se sert RECHER à partir des données de fin d'évolution.

On se sert de Vcicuit et de la tension au moteur pour obtenir, à l'aide des tables de l'estimateur, le couple estimé. Ces tables sont:

-TBCOUR qui mémorise la courbe de référence à 85V; on y accède à l'aide de Vcircuit discrétisé 8 bits et on obtient le couple correspondant pour une tension de 85V.

-TBCOMP qui donne la compensation de couple par rapport à la courbe à 85V en fonction de la tension au moteur.

Les fondements de la méthode sont donnés à la section 2.1.1.2 et à la figure 2.1.1.2-7.

A l'aide du couple estimé, on obtient les consignes nécessaires (tabulées pour chacune des variables de commande en fonction du couple et des différents critères) pour fonctionner au point optimal à un critère donné. On notera que ces tables sont bâties et améliorées par le processus d'évolution. Si les données n'ont pas été

tabulées pour un couple donné, on fera une recherche (module RECHER) dans l'environnement immédiat (autour de cette valeur de couple); si on ne trouve pas une consigne, on utilise la valeur prévue à l'initialisation.

La valeur trouvée est moyennée avec la valeur actuelle de consigne: on a une approche binaire vers la supposée valeur optimale. Cela nous rend plus tolérant à une erreur d'estimation. Cela nous ralentit aussi, mais comme le processus d'estimation, se basant sur une mesure et des tables, est rapide en lui-même, on le fait.

Le fonctionnement en boucle ouverte se sert du travail fait en évolution d'optimisation; travail qui n'est pas perdu. En plus, si on retourne en évolution d'optimisation, on améliore constamment le contenu actuel des tables (voir le module AJTBBO) en moyennant son contenu pour un couple donné avec la nouvelle valeur trouvée.

Le pseudocode des modules sont présentés [3]; ils sont assez détaillés pour permettre de saisir les idées de bases de la procédure ainsi que la compréhension de leur code [3].

3.2.3.4 L'interface avec l'utilisateur.

Le diagramme hiérarchique nous montre le module COMUSR et ses modules inférieurs CHXSTR, AFFDRV, INTERO, REMAFF et AFFMES qui réalisent l'interface. Ces modules se servent de l'écran et du clavier hexadécimal reliés au SYM. Cet interface permet:

-d'arrêter l'entraînement,

-de choisir sa structure (définition des registres de commande),

-de définir les valeurs qu'on veut faire afficher (RCAFFI),

-d'afficher les valeurs demandées (selon le registre RCAFFI),
-d'afficher le contenu des registres de commande de l'entraînement,
-de modifier le point d'opération du convertisseur.
Selon le code entré au clavier hexadécimal, COMUSR exécutera une ou l'autre des fonctions énoncées ci-haut. Notons entre autres que pour modifier le point d'opération, on peut soit incrémenter ou décrémenter les valeurs actuelles, soit en entrer de nouvelles. Pour demander des valeurs, le module se sert de INTERO (module d'interrogation) qui utilise le module AFFMES pour envoyer l'interrogation et de sous-routines du SYM pour gérer le clavier.

Pour définir la structure de l'entraînement, on se sert du module CHXSTR. Ce dernier interroge d'abord pour connaître la définition désirée de RCDRIV (registre de commande de la "drive"); le message d'interrogation explique le rôle de chacun des bits de ce registre. Si un asservissement est désiré, (BØ=1 dans RCDRIV), on interroge pour connaître le critère de l'asservissement, soit RCRITA. Cette valeur nous donne, à l'aide des tables TBRCME et TBRCFL les registres de commande des mesures et de filtrage RCOMES et RCOMFL. On notera que le type d'asservissement est spécifié dans RCDRIV (voir le pseudocode de COMUSR et CHXSTR). Puis, pour les cas d'asservissement par évolution d'optimisation et de fonctionnement en boucle ouverte, on rajoute Vcircuit comme valeur à mesurer et à traiter (modifier RCOMES et RCOMFL en conséquence) et, selon le critère choisi (RCRITA), on détermine le nombre de variable de commande à utiliser.

Après cela, on s'occupe du choix des valeurs à afficher en définissant le registre d'affichage RCAFFI; on se sert

du module REMAFF pour le faire. Si l'utilisateur demande d'afficher une valeur qui n'est pas mesurée ou obtenue par différents calculs pour l'asservissement en cours, le module AFFDRV, affichant les valeurs demandées par RCAFFI lorsque c'est prévu dans RCDRIV, affichera des points d'interrogations à la place. La figure 3.2.3.4-1 illustre la procédure de ce module.

Pour modifier les consignes, on se sert de module de modification des consignes MODCSN. Pour changer directement les valeurs, on agit sur les consignes directement.

Les pseudocodes des modules ci-haut nommés sont tous en [3]; ils sont très explicites et c'est pourquoi nous n'irons pas plus en détail dans la description.

3.2.3.5 La commande de l'entraînement.

Elle est réalisée par le module DRIVE. Comme nous l'avons déjà bien décrit en 3.2.1, nous ne le reprendrons pas. Le pseudocode [3] est très suffisant pour comprendre le reste.

On y notera l'utilisation du module CORDRV vérifiant l'auto-diagnostique de chacun des modules supérieurs (TRTSGN, MESURE,...) et corrigeant les registres de commande pour permettre à l'entraînement de poursuivre en cas d'erreur mineure.

On utilise aussi le module de communication CMSYM2 appelé par interruption masquable IRQ: ce module permet entre autres le transfert des consignes de tension, de déphasage et de stratégie vers le microordinateur commandant le convertisseur; la communication est bidirectionnelle et emprunte le même bus que la sortie du convertisseur analogique-numérique, et ce pour les deux

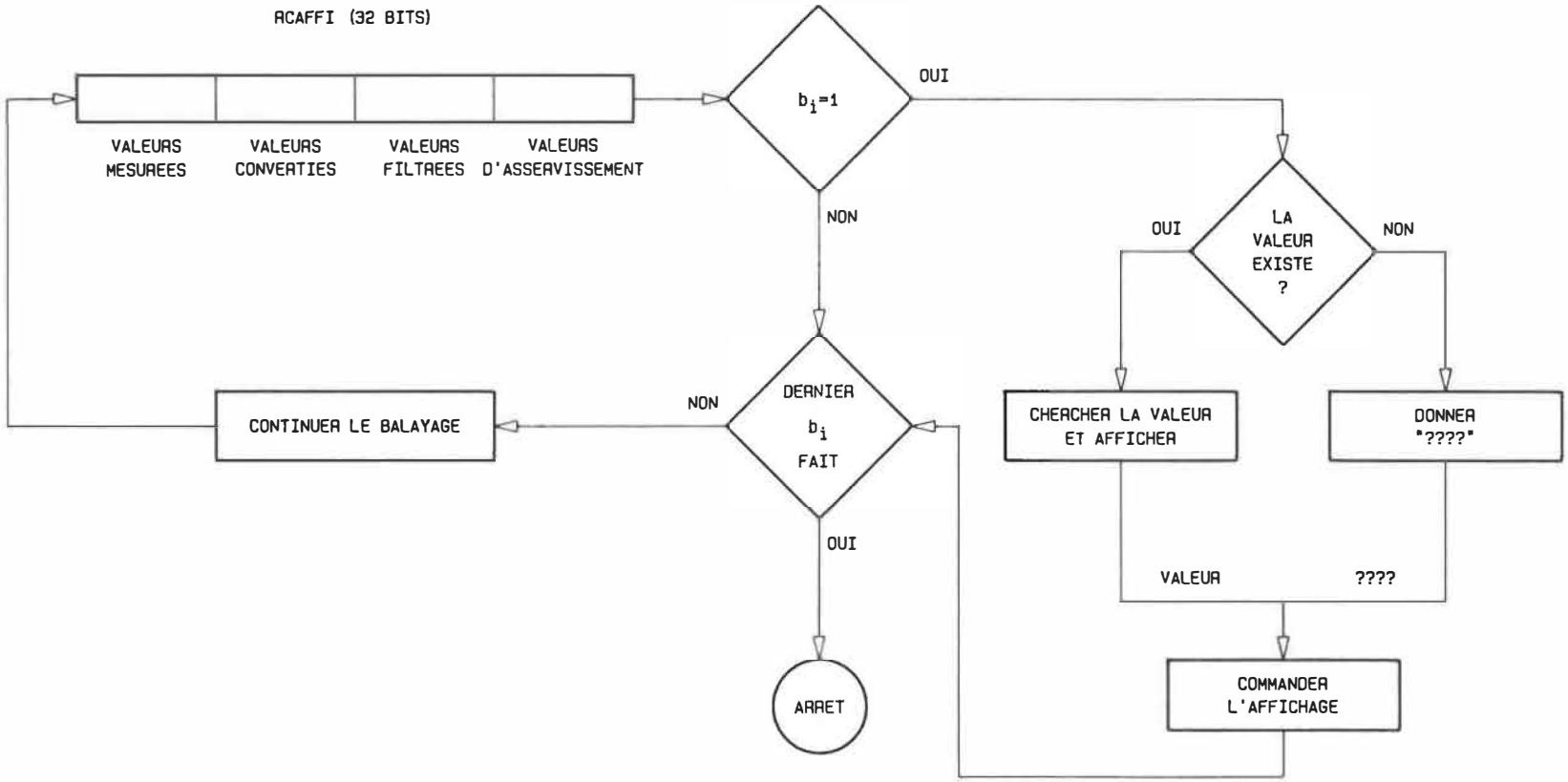


fig 3.2.3.4-1 AFFICHAGE ET "RCAFFI"

microordinateurs.

Les différents pseudocodes [3] expliquent clairement la procédure des modules ci-haut nommés.

3.2.4 Le matériel.

On décrit brièvement quelques circuits, moins courants ou plus importants, réalisant des fonctions des schéma-blocs illustrant le matériel. Ce sont les circuits de:

-l'amplificateur à gain programmable;

-les "filtres-AGC-DPZ" numériques et analogiques;

-mesure du déplacement (TETA1).

L'amplificateur à gain programmable est présenté à la figure 3.2.4-1. Le signal d'entrée est reçu d'un amplificateur différentiel puis passe par un second amplificateur opérationnel dont le gain est commandé par la sélection de la résistance de feedback au multiplexeur analogique (LF13508). La commande du multiplexeur vient du microprocesseur (par l'intermédiaire d'un VIA). Un détecteur de seuil aide à l'ajustement du gain en signalant le dépassement d'un certain seuil par la tension crête de sortie de l'amplificateur à gain commandé. Le seuil est réglé par le réseau de résistances dont la sortie est dirigée vers l'entrée inverseuse d'un comparateur; son entrée positive reçoit la mesure de tension crête. On a prévu une petite hystérèse pour éviter les oscillations possibles autour du seuil. La sortie du comparateur est reliée à une ligne d'un port d'entrée/sortie du microordinateur.

Le "filtre-AGC-DPZ" numérique est illustré à la figure 3.2.4-2. La sortie de l'amplificateur différentiel recevant une tension proportionnelle au courant de source (d'une

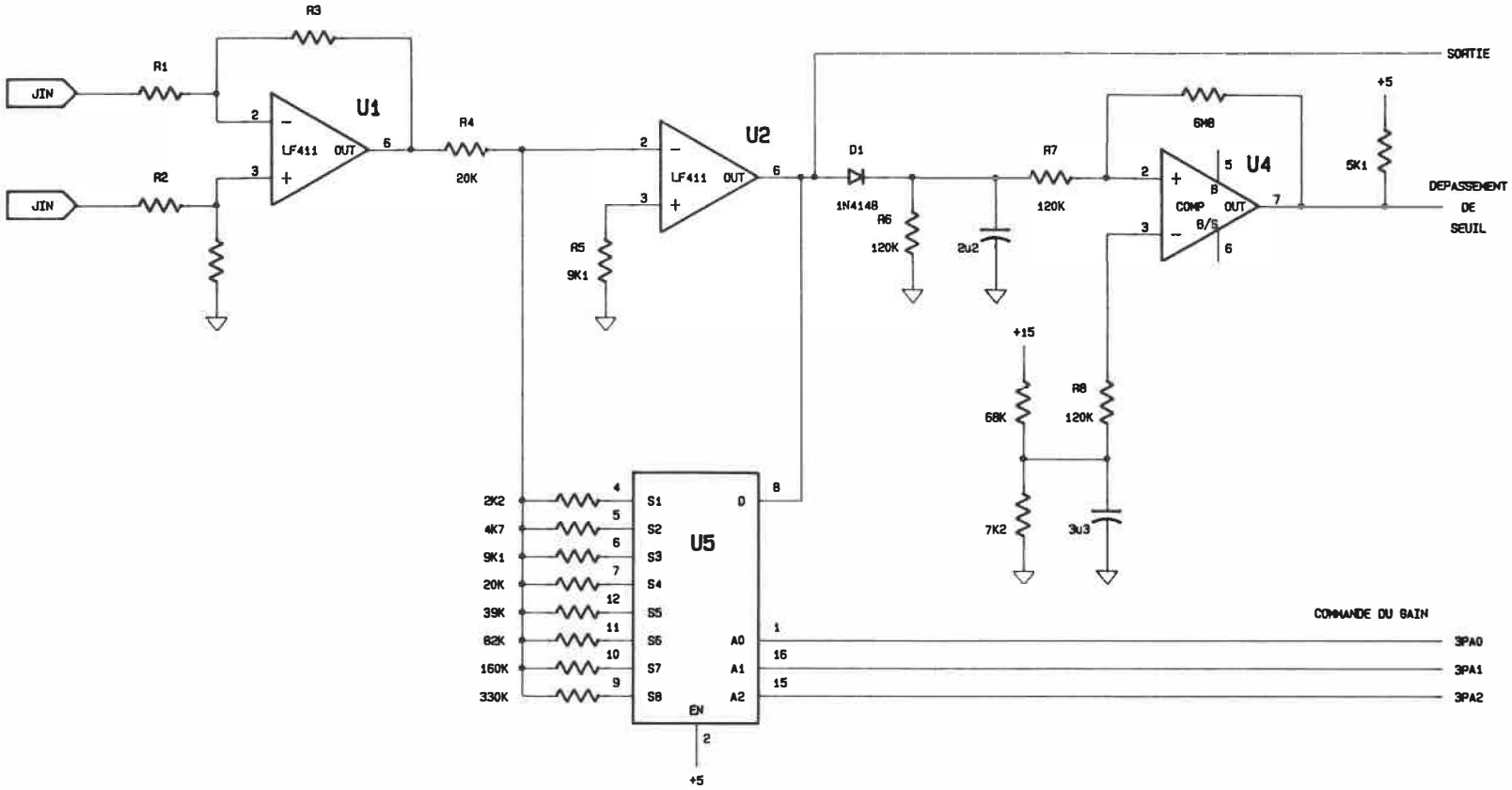


FIGURE 3.2.4-1 AMPLIFICATEUR A GAIN PROGRAMMABLE

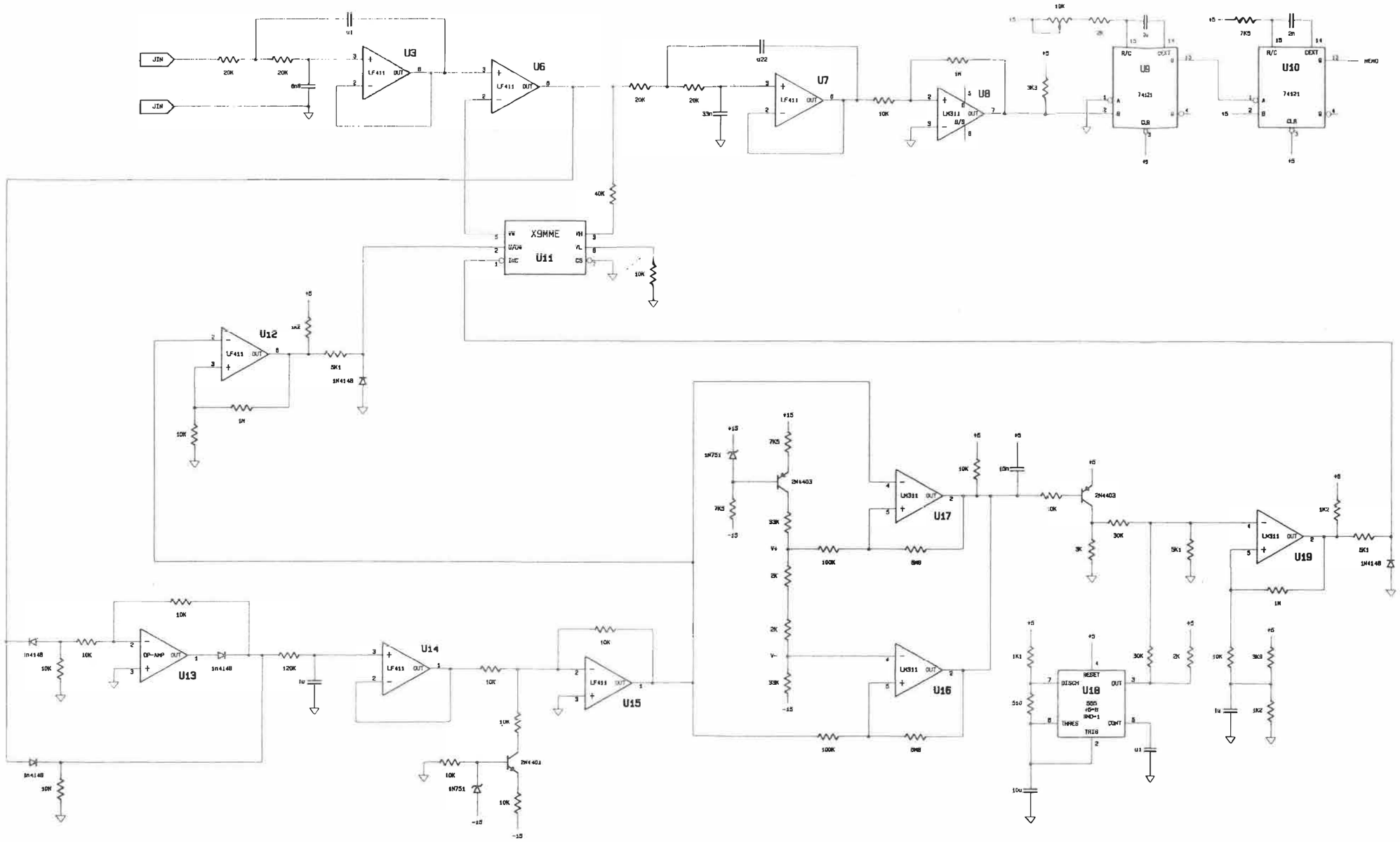


FIGURE 3.2.4-2 "FILTRE - AGC - DPZ" NUMERIQUE

sonde à effet Hall) passe par un filtre passe-bas du second ordre. Puis le signal filtré passe par un amplificateur à gain commandé (élément de base de l'AGC). La sortie de cet amplificateur (à amplitude maximale régulée) est envoyé au circuit asservissant son gain ainsi que vers un second filtre passe-bas du second ordre. Finalement, la sortie du deuxième filtre passe par un comparateur, puis un déphaseur pour compenser le retard dû aux deux filtres dans le parcours du signal. La sortie du déphaseur est une impulsion générée en début du cycle de la fondamentale du courant de source. La régulation du gain se fait par l'ajustement du diviseur de feedback fournissant l'échantillon de tension a l'entrée inverseuse de U6. Le circuit intégré U11 est un potentiomètre de 50K dont on commande la position du curseur à l'aide de ses entrées "U/D*" (sens de correction) et "INC" (ordre de changer la position du curseur); il y a 200 positions également reparties pour le curseur. Les résistances de 40K et 10K qui sont reliées aux entrées de ce potentiomètres fixent les gains maximal et minimal. On relie le curseur du potentiomètre à l'entrée inverseuse de U3, permettant de commander le gain de l'amplificateur non inverseur ainsi formé. Le gain peut varier entre 1,67 et 10. La circuiterie autour des amplificateurs U13 et U14 permet la mesure de la tension maximale de sortie de l'amplificateur à gain commandé. Cette valeur est comparée à une référence (environ 4,3V) dont le signal d'erreur est à la sortie de U15. A l'aide du signe de l'erreur, on détermine le sens de la correction à U12 et on l'envoie à la broche "U/D*" du potentiomètre commandé U11. L'erreur est aussi comparée à une certaine fenêtre déterminée par "V+" et "V-" respectivement reliés aux comparateur U17 et U16. Une horloge vient échantilloner régulièrement la sortie du

comparateur en fenêtre permettant la génération d'une impulsion de commande sur INC de U11 lorsque l'erreur est à l'extérieur de la fenêtre. Cette impulsion provoque le changement d'un incrément du curseur dans le sens de correction déterminé par la tension à la broche U/D* de U11. Normalement, la période d'échantillonnage doit être supérieure au délai dans la mesure de la tension maximale de sortie de l'amplificateur à gain ajusté. Pour ne pas trop ralentir la boucle de réglage, nous les avons pris de environ de même valeur. Par un réglage convenable de la fenêtre (plage morte), on évite les cycle limites (oscillations de faible amplitude) autour du point d'équilibre.

La figure 3.2.4-3 illustre le "filtre-AGC-DPZ" analogique utilisés pour la détection des passage par zéro des fondamentales de tension et courant au moteur. Les circuits U20 et U21 réalisent un filtre passe-bas du 4 ième ordre de type Butterworth à fréquence de coupure près de 60Hz. La composante alternative du signal passe par un diviseur asservi pour réguler la tension maximale de sortie de l'amplificateur non inverseur à gain fixe le suivant. La tension de référence "Vref." (environ 4,3V) est comparée à la mesure de la tension maximale de sortie de U22; l'erreur passe par un régulateur de type intégrateur dont la sortie (limitée à des valeurs positives) commande la tension de gâchette d'un JFET (2n5462) agissant en résistance commandée. Finalement, la tension de sortie de U22 est dirigée vers un DPZ à niveaux. Qu'importe la tension à l'entrée du filtre (connecteur JIN), la détection du passage par zéro se fait avec des signaux d'amplitude maximale constante et de forme sinusoïdale (on n'a que la fondamentale à 60Hz après le filtre). Cela permet de conserver une incertitude constante sur la mesure du DPZ.

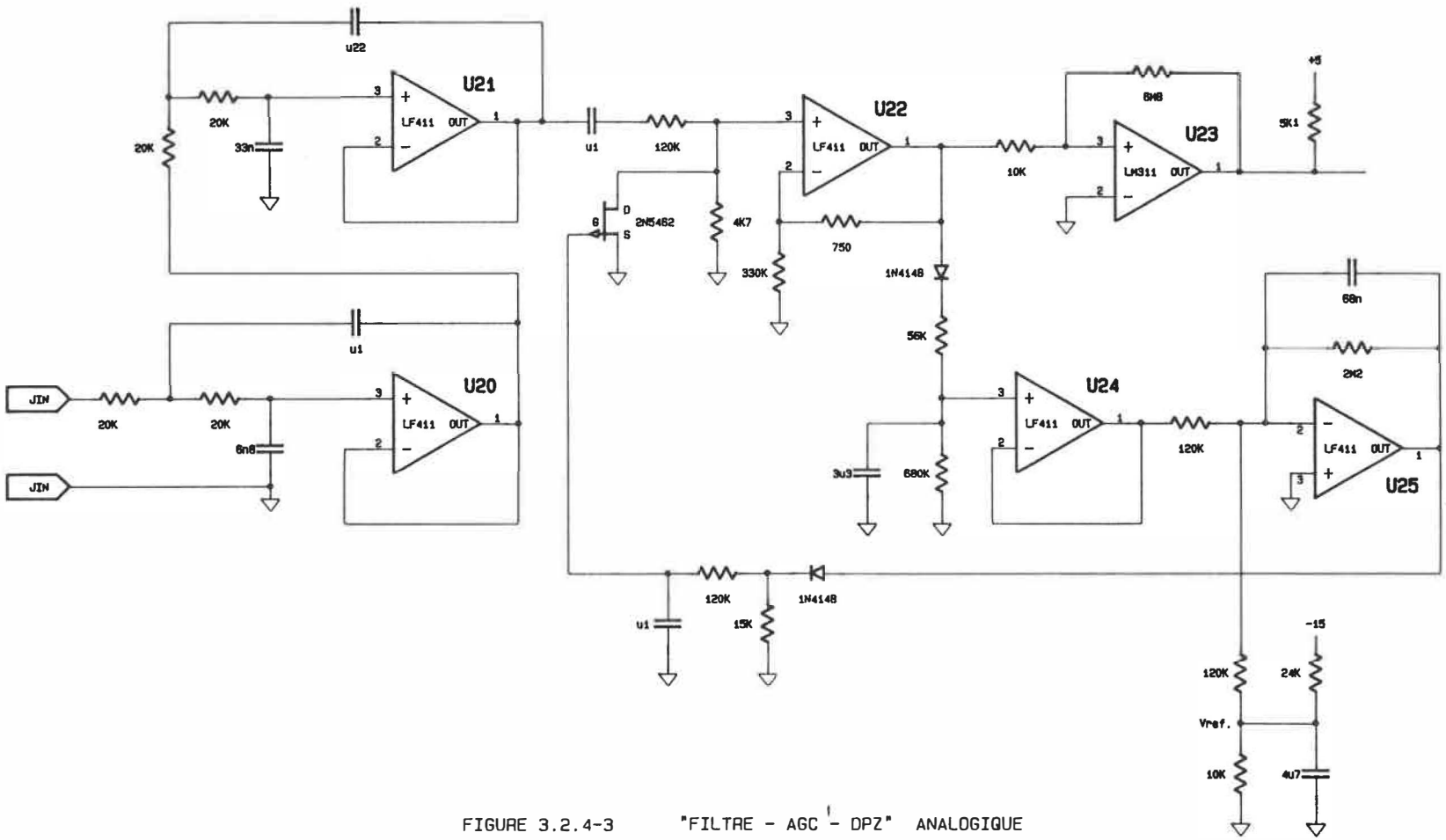


FIGURE 3.2.4-3 "FILTRE - AGC - DPZ" ANALOGIQUE

On notera que les délais de filtrage ont été égalisés entre les deux circuits servant respectivement à la tension et au courant; on peut donc mesurer le déplacement entre les fondamentales de ces ondes.

Le circuit de mesure du déplacement à la source est illustré à la figure 3.2.4-4. Il s'agit d'un compteur 8 bits (les 2 7493) lancé au début d'un cycle réseau et incrémenté au rythme des impulsions de l'horloge à 61,4KHz formée avec le 555 dont on mémorise (74374) la valeur au passage par zéro du courant.On obtient une mesure du délai entre les passages par zéro du courant et de la tension. La sortie numérique de la mémoire (le délai) est convertie analogiquement par le convertisseur MC1408 (U30) et l'amplificateur U31. Avec l'horloge à 61,4 KHz et le compteur 8 bits, on obtient une plage mesure de 0 à 90° à 60Hz.

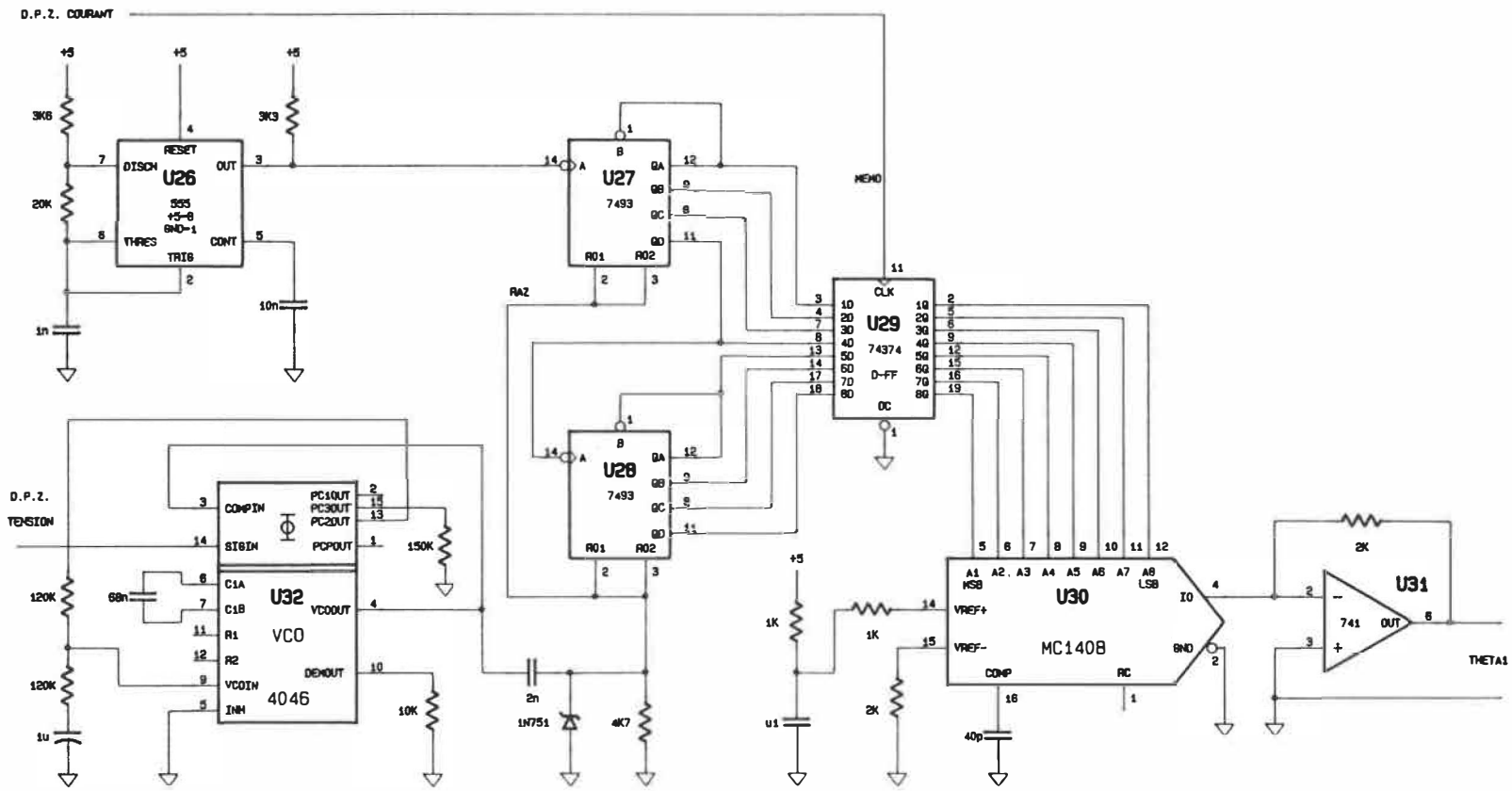


FIGURE 3.2.4-4 MESURE DU DEPHASAGE

CONCLUSION

L'entraînement présenté au chapitre 3 est un système multi-processeurs permettant beaucoup de flexibilité, ce qui en fait un outil intéressant pour le développement ou pour le test de:

-nouvelles stratégies de commutation des interrupteurs;

-paramètres de régulation ou de critères d'optimisation.

Les asservissements proposés sont de type "régime permanent" et n'utilisent que les valeurs dérivées des tensions et courants au moteur. Les variables de commandes sont l'amplitude (valeur efficace) et la phase de la tension de sortie du convertisseur.

L'étude de la machine nous a donné une méthode pour obtenir une mesure de la réserve énergitique du moteur par rapport à un point d'opération très performant (vitesse nominale) en se servant de l'enroulement de démarrage. A partir de cette mesure, on peut commander la convertisseur de façon à maintenir la réserve à un niveau donné. On peut aussi s'en servir pour estimer le couple au moteur en régime permanent. Cette mesure est donc très intéressante, mais reste en partie dépendante de la construction et de l'utilisation de l'enroulement de démarrage du moteur.

L'évolution d'optimisation est une méthode tout à fait générale de réglage du point d'opération du moteur. C'est une méthode itérative qui passe par une succession de régimes permanents. A chacun d'eux, on évalue la performance du système pour déterminer les prochaines commandes. La performance est définie par une fonction de coût; on doit la minimiser. On évolue dans la direction

(inverse) du gradient pour maximiser (minimiser) la performance (le coût). Cet asservissement est idéal pour les systèmes mals définis car il n'en requiert pas le modèle. Un entraînement ne possédant pas d'information directe sur l'évolution de son système mécanique, ne possède pas d'information sur deux de ses variables d'état; c'est donc un système mal défini. La méthode de calcul du gradient et la stratégie d'évolution permettent une évolution relativement rapide vers la point "optimal". Une approche rapide permet de passer plus de temps près du point optimal; point changeant avec les variations de charge au moteur.

Le système de commande de l'entraînement permet de bâtir dynamiquement des tables en mémorisant les consignes de fin d'évolution en fonction du couple estimé. Avec un système suffisamment excité, on peut obtenir une table bien garnie des consignes nécessaires pour opérer à un point quasie optimal, selon un critère donné. On pourrait donc utiliser l'estimation du couple et ce jeu de tables pour fonctionner en "boucle ouverte" (pas de calcul de fonction d'optimisation) et obtenir un point d'opération quasie optimal rapidement car on n'a pas besoin d'évoluer vers ce point qu'on connaît déjà.

Des études supplémentaires sur l'effet de la stratégie de commutation des interrupteurs sur Vcircuit devront être menées. De même, on devra étudier rigoureusement les différents types d'enroulements de démarrage existants sur les moteurs pour vérifier lesquels permettent d'obtenir Vcircuit. Enfin, on devra concevoir un nouveau système à microprocesseur beaucoup plus puissant (plus rapide et avec un jeu d'instructions plus grand), le système actuel étant à la limite de ses possibilités (surtout le système de commande du convertisseur).

BIBLIOGRAPHIE

Liste des références.

1. BEAUREGARD, François, "Commande polyvalente d'une chaîne variateur-moteur asynchrone", mémoire de maîtrise, Département de Génie Electrique, Ecole Polytechnique, Montréal 1985, 156p.

2. BOOSE, B.K., "Power electronics and AC drives", Prentice Hall, New-Jersey 1986, pp.28-52.

3. D'ULISSE, Richard, "CAHIER ANNEXE AU MEMOIRE SYSTEME DE REGLAGE DE MACHINE ASYNCHRONE MONOPHASEE", Département de Génie Electrique, Ecole Polytechnique, Montréal 1989.

4. JIAN, T.W., SCHMITZ, N.L., NOVOTNY, D.W., "Characteristic induction motor slip values for variable voltage part load performance optimization", IEEE trans. on Power Apparatus and Systems, vol. PAS-102, no. 1, Jan. 1983, pp.38-46.

5. MOHAN, N., "Improvement in energy efficiency of induction motors by means of voltage control", IEEE trans. on Power Apparatus and Systems, vol. PAS-99, no. 4 July/Aug 1980, pp.1466-1471.

6. ROWAN, T.M., LIPO, T.A., "A quantitative analysis of induction motor performance improvement by SCR voltage control", IEEE Power Electronics Specialists Conference, PESC'82 Record, 1982, pp.312-320.

7. MURPHY, J.M.D., HOSINGER, V.B., "Efficiency optimization of inverter-fed induction motor drives", IEEE, IAS Juin 1982, pp.544-552.

8. POITEVIN, P., "Commande de moteur asynchrone par variateur programmable", mémoire de maîtrise, Département de Génie Electrique, Ecole Polytechnique, Montréal 1985, 136p.

9. ROY, G., POITEVIN, P., OLIVIER, G., "Characteristics of chopper fed single phase induction motors", IEEE IAS Annual Meeting 1985, Toronto, 6-11 oct. 1985, pp.1075-1080.

10. BECK, M., DIETRICH, N., "A new method for the calculation of the slip frequency for a sensorless control of a squirrel-cage induction motor", IEEE 1985, pp.678-683.

Ouvrages consultés.

11. ABBONDANTI, A., BRENNEN, M., "Variable speed induction motor drives use electronic slip calculator based on motor voltages and currents", IEEE trans. on Industry Applications, vol. IA-11, no. 5, Sep/Oct 1975, pp.483-488.

12. MIN HO PARK, SEUNG KI SUL, "Microprocessor-based optimal efficiency drive of an induction motor", IEEE trans. on Industrial Electronics, vol. IE-31, no. 1, Feb 1984, pp.69-73.

13. SYM-1 REFERENCE MANUAL, Synertek Systems Corp. Pub. MAN-A-260006-D, Santa Clara Ca., 1981, approx. 200p.

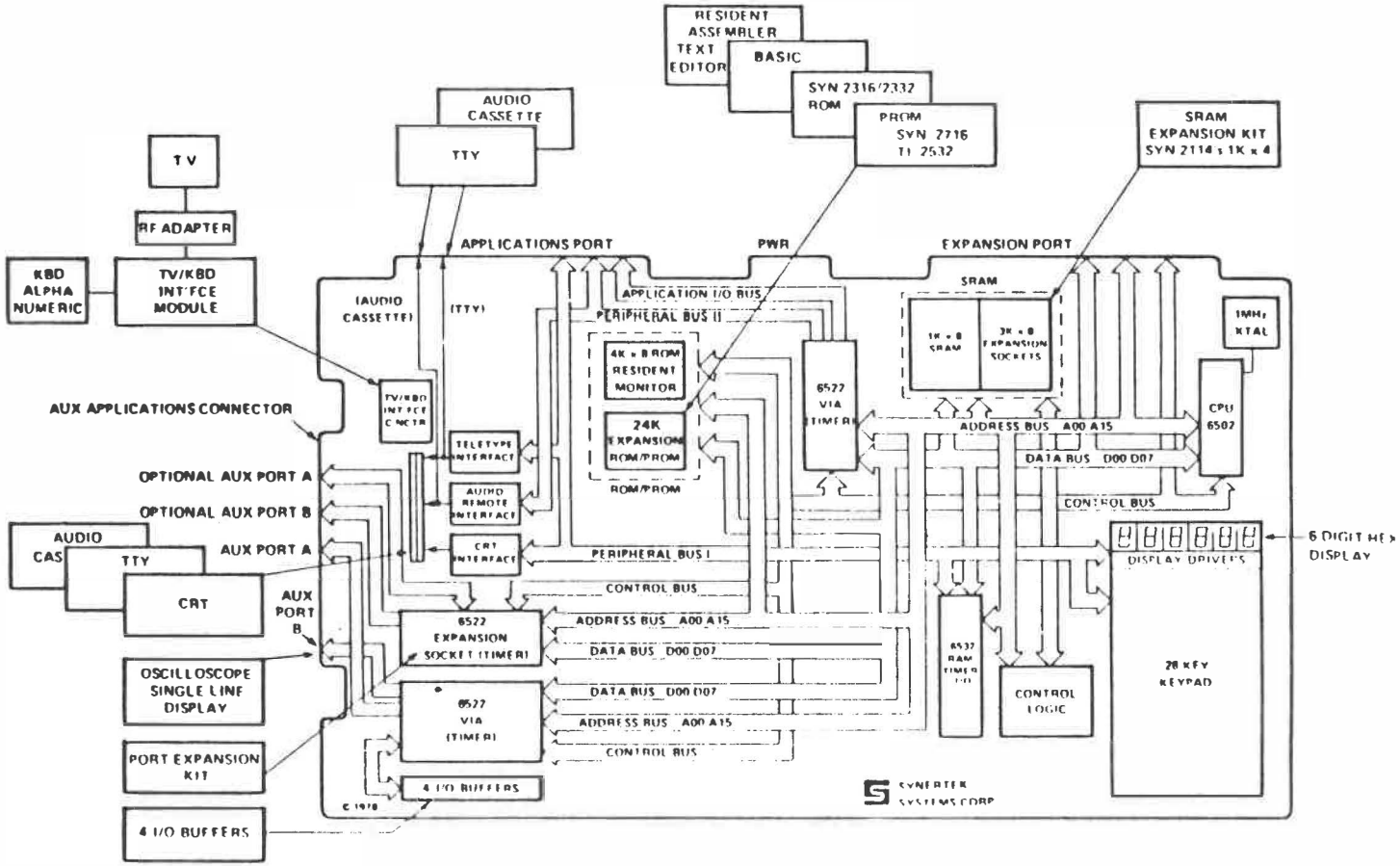
14. WHITE, D.C., WOODSON, H.H., "Electromechanical Energy Conversion", John Wiley and Sons inc., New-York 1959, 646p.

15. OGATA, K., "Modern control engineering", Prentice Hall inc., New-Jersey 1970, 836p.

16. SEVELY, Y., "Systèmes et asservissements linéaires échantillonnés", Dunod Université, Dunod, Paris 1973, 210p.

ANNEXE A:

Schéma simplifié d'un microordinateur SYM.


RESIDENT ASSEMBLER TEXT EDITOR
BASIC
SYN 2316/2332 ROM
PROM SYN 2716 TI 2532
SRAM EXPANSION KIT SYN 2114 x 1K x 4
AUDIO CASSETTE
TTY
TV
RF ADAPTER
KBD ALPHA NUMERIC
TV/KBD INT'FCE MODULE
APPLICATIONS PORT
PWR
EXPANSION PORT
(AUDIO CASSETTE)
(TTY)
APPLICATION I/O BUS
PERIPHERAL BUS II
4K x 8 ROM RESIDENT MONITOR
24K EXPANSION ROM/PROM
ROM/PROM
6522 VIA (TIMER)
SRAM
1K x 8 SRAM
3K x 8 EXPANSION SOCKETS
1MHz XTAL
CPU 6502
ADDRESS BUS A00 A15
DATA BUS D00 D07
CONTROL BUS
AUX APPLICATIONS CONNECTOR
TV/KBD INTFCE CNCTR
TELETYPE INTERFACE
AUDIO REMOTE INTERFACE
CRT INTERFACE
OPTIONAL AUX PORT A
OPTIONAL AUX PORT B
AUX PORT A
AUX PORT B
AUDIO CAS
TTY
CRT
PERIPHERAL BUS I
CONTROL BUS
6522 EXPANSION SOCKET (TIMER)
ADDRESS BUS A00 A15
DATA BUS D00 D07
6522 VIA (TIMER)
DATA BUS D00 D07
ADDRESS BUS A00 A15
CONTROL BUS
4 I/O BUFFERS
OSCILLOSCOPE SINGLE LINE DISPLAY
PORT EXPANSION KIT
4 I/O BUFFERS
6537 RAM TIMER I/O
CONTROL LOGIC
DISPLAY DRIVERS
6 DIGIT HEX DISPLAY
28 KEY KEYPAD
SYNERTEK SYSTEMS CORP
© 1978

ANNEXE B:

Le modèle DQ du moteur.

Une machine peut se décrire par un système d'équations reliant les tensions de chaque enroulement du stator et du rotor aux courants qui circulent dans tous les enroulements. Ainsi, pour les différents enroulements:

$$v = R.i + \frac{d(\text{flux embrassé})}{dt}$$

Et, pour le premier enroulement:

$$v_1 = R.i_1 + \frac{d(L_{11}.i_1 + \sum_{j=2}^{n} M_{1j}.i_j)}{dt}$$

Dans cette équation, L représente une valeur d'inductance de fuite et M représente une mutuelle entre deux enroulements.
La valeur des inductances dépend de la position angulaire du rotor par rapport au stator. Ce modèle ne tient pas compte de la possible saturation du circuit magnétique de la machine.

De façon générale, on peut décrire une machine à l'aide d'une machine dite **généralisée** ayant deux phases en quadrature au stator et au rotor. Dans le cas du moteur asynchrone monophasé, les composantes D et Q représentent des tensions et courants directement mesurables au enroulements du stator, et les composantes α et β , des tensions et courants rotoriques. Dans le cas particulier d'un moteur à cage, les tensions selon α et β sont nulles,

et les courants (des équivalents fictifs) sont généralement non mesurables. En considérant les hypothèses suivantes:

- les enroulements rotoriques équivalents sont similaires et les enroulements statoriques sont similaires ;
- l'entrefer est uniforme;
- l'inductance mutuelle (fonction de l'angle) varie de façon sinusoïdale avec l'angle (θ) entre les enroulements concernés;

Et:

- qu'il n'y a pas de flux embrassé entre des enroulements à angle droit;
- que les inductances mutuelles sont égales;

On obtient le système d'équation suivant:

$$\begin{vmatrix} v_\alpha \\ v_\beta \\ v_D \\ v_Q \end{vmatrix} = \begin{vmatrix} R_r + L_r.p & & M.p.\cos\theta & M.p.\sin\theta \\ & R_r + L_r.p & M.p.\sin\theta & -M.p.\cos\theta \\ M.p.\cos\theta & M.p.\sin\theta & R_s + L_s.p & \\ M.p.\sin\theta & -M.p.\cos\theta & & R_s + L_s.p \end{vmatrix} \cdot \begin{vmatrix} i_\alpha \\ i_\beta \\ i_D \\ i_Q \end{vmatrix}$$

M : valeur maximale de l'inductance mutuelle entre deux enroulements (appelée L_{sr} à la figure 2.1.2-1).

L_s, L_r : inductance de fuite d'un enroulement du stator et du rotor respectivement

R_r, R_s : résistance propre d'un enroulement du rotor et du stator respectivement

p : opérateur de dérivation par rapport au temps; agit sur le produit des éléments de la matrice d'impédance et des courants associés (l'expression qui suit le terme p).

Le modèle DQ est obtenu en appliquant une transformation sur le rotor de façon à obtenir des composantes rotatives par rapport au rotor, mais fixe par rapport au stator sur des axes fictifs d et q. Ce système d'équations présente l'avantage de ne pas dépendre de la position du rotor, mais de sa vitesse. Comme la vitesse est relativement constante, ou de variation relativement lente (le système mécanique est relativement lent par rapport aux constantes de temps électriques), du point de vue électrique, les équations différentielles du système d'équations ont alors des coefficients constants. Le modèle obtenu est de la forme suivante:

$$\begin{vmatrix} v_d \\ v_q \\ v_D \\ v_Q \end{vmatrix} = \begin{vmatrix} R_r + L_r.p & L_r.\dot{\theta} & M.p & M.\dot{\theta} \\ -L_r.\dot{\theta} & R_r + L_r.p & -M.\dot{\theta} & M.p \\ M.p & \emptyset & R_s + L_s.p & \emptyset \\ \emptyset & M.p & \emptyset & R_s + L_s.p \end{vmatrix} . \begin{vmatrix} i_d \\ i_q \\ i_D \\ i_Q \end{vmatrix}$$

Au point de vue de la détermination des paramètres, les méthodes de mesure reposent sur la séparation des composantes: on essai de déterminer les paramètres en ne permettant que le passage d'un seul courant à la fois, ce qui réduit les inconnues dans les équations de tension impliquées. Puis, avec des tests à rotor calé ($\dot{\theta} = \emptyset$) et des mesures en CC, on peut alors identifier une autre partie des inconnues.

Pour la cage d'écureuil, il est impossible d'annuler le courant i_α ou i_β indépendamment. Aussi, la mesure des paramètres d'inductance de fuite et de résistances sont difficiles. Néanmoins, le principe de base reste le même:

annuler le plus de courants à la fois pour simplifier la mesure. Par la suite, en utilisant des corrections empiriques et des corroborations par simulation, on arrive à estimer assez bien ces différentes valeurs.

Le modèle d'état du moteur utilisé pour la simulation est directement dérivé du modèle DQ. La figure 2.1.2-1 représente schématiquement le système d'équations du modèle DQ. Pour plus de clarté, on a utilisé des exposant s ou r (au lieu de majuscule ou minuscule) pour distinguer entre les **D et Q** du stator et les **d et q** du rotor.

ANNEXE C:

Programme de simulation de la chaîne convertisseur-moteur.

Le programme demande le temps de simulation et certains autres paramètres fondamentaux: intervalles d'intégration, nombre de pas dans un cycle 60Hz -précision de l'analyse-, et la consigne de tension. Puis, il commence la simulation. Le programme utilise le modèle d'état du moteur présenté au chapitre 2. Pour obtenir la tension alimentant le moteur (sortie du convertisseur), on utilise le fichier SYMTAB contenant les instants d'arrivée des événements d'une stratégie en fonction de la tension. L'analyse se fait pour chacune des parties entre deux événements successifs car cela défini des plages où l'entrée est nulle ou égale à la tension du réseau (directe ou inversée). Dans chacune de ces régions, on calcule l'intégrale de convolution sur un pas; les résultats sont mémorisés et affichés. DATAM, le fichier de sortie, reçoit les résultats de la simulation sur chacun des pas. Ce fichier est traité, par d'autres programmes, pour obtenir la forme d'onde ou l'analyse harmonique d'un signal.

SYMOT1.BAS

```
10 DEFDBL T,A,Y,X,V,P,H,B,S
11 INPUT "NBRE D'INTERVALLE POUR INTEGRATION?";NINT
12 PERIODE=1/60
13 INPUT "NBE CYCLE DE SIMULATION?";NBCYCSIM
14 REM Attention aux "PHI(t)" NEGATIF OU >1
15 IMPOSSIBLE POUR SYSTEME STABLE...
16 REM période D'ANALYSE DOIT ETRE <TO.
17 REM NDIVISIONS TOTAL DANS T > MAX# * T
18 REM MAX# approxime 1/TOM lorsque inconnu
19 REM ndivision recalculé en meme temps que A(I,J,K).
20 DIM T(5,120),A(3,3,5),ID(3,3),PHI(3,3,100),
21 DIM PHIPRE(3,3),IRMS(400),VITESSE(400),PMOY(400)
22 DIM Y(3,100),X(3),VECVAL(2000)
23 INPUT "NBE DE DIVISIONS(>360)?";DIVMIN
24 INPUT "CONSIGNE DE TENSION?";CSN
34 OPEN "SYMTAB" FOR INPUT AS #2
35 FOR N=0 TO 120
37 INPUT #2,T(0,N),T(1,N),T(2,N),T(3,N),T(4,N),T(5,N)
39 NEXT N
41 CLOSE #2
45 PI=3.1415296#
50 LS=.0917:LR=LS:LSR=.0875
60 RR=1.91:RS=1.12
70 J=.1:ALPHA=0!:NP=2
80 K1=1/LS:K2=LSR*LSR/(RR*LS):K3=RS/LS:K4=RR/LR
82 K5=NP*LSR*LSR/(RR*J):K6=ALPHA/J
90 A11=-K3/(1-K2*K4):A12=NP*K2/(1-K2*K4)
92 A13=K2*K4/(1-K2*K4)
100 A21=NP*K4:A22=-K4:A23=NP
110 A31=K3*K4/(1-K2*K4):A32=-NP/(1-K2*K4)
112 A33=-K4/(1-K2*K4)
120 B1=K1/(1-K2*K4):B2=0:B3=-K1*K4/(1-K2*K4)
126 CLS:LOCATE 1,1 : PRINT "NB CYC.: ";NBCYCSIM
127 LOCATE 1,20: PRINT "PERIODE: ";PERIODE
128 LOCATE 1,52: PRINT "NB INTERV. INT.: ";NINT
129 LOCATE 3,1: PRINT "------------------------"
130 LOCATE 5,1:PRINT "NUMERO CYCLE: ";NUMCYC
131 LOCATE 5,20:PRINT "TEMPS: ";TEMPS
132 LOCATE 5,50:PRINT "X1=";X(1)
133 LOCATE 6,50:PRINT "X2=";X(2)
134 LOCATE 7,50:PRINT "X3=";X(3)
135 LOCATE 8,50:PRINT "Y="
136 LOCATE 6,1:PRINT "NB MAX DANS A:"
137 LOCATE 8,1:PRINT "NDIVISION=";NDIVISION
138 LOCATE 7,5:PRINT MAX#
139 LOCATE 9,1:PRINT "VITESSE=";DTETA*60/(2*PI)
140 LOCATE 11,1: PRINT "--------------------------"
141 LOCATE 13,1:PRINT "ELEMENTS DE MATRICE DE TRANSITION"
142 LOCATE 14,1:PRINT "PHI(1,1)=";PHI(1,1,0)
143 LOCATE 15,1:PRINT "PHI(1,2)=";PHI(1,2,0)
144 LOCATE 16,1:PRINT "PHI(1,3)=";PHI(1,3,0)
145 LOCATE 17,1:PRINT "FLAG ERREUR:";FLAGERREUR
150 LOCATE 14,50:PRINT "VITESSE= "
151 LOCATE 15,50:PRINT "IRMS= "
190 DTETA=185*.9
210 FOR I=1 TO 3
230 FOR J=1 TO 3
240 ID(I,J)=0
250 NEXT J
260 ID(I,I)=1:X(I)=0
270 NEXT I
280 CPTDTA=0:SUMDTETA=0:TEMPS=0:INDEXVAL=0
```

```
290 GOSUB 1000 'CALCULER MATRICES A
300 OPEN "DATAM" FOR OUTPUT AS #1
310 REM: LE CYCLE EST DIVISE EN 5 PARTIES
315 REM CAR IL Y A 5 INTERV. DEFINIS PAR
320 REM :4 EVENEMENTS DE LA STRATEGIE
321 REM ALPHA/BETA SYMETRIQUE
330 FOR NUMCYC=0 TO NBCYCSIM-1
340 REM de t=0 a t(alpha1)
350 DELTAT=T(1,CSN)-0
360 UT=0
361 NDIVTYP=PERIODE*DIVMIN/DELTAT
362 NDIVMIN=MAX#*2*DELTAT
363 IF NDIVMIN<NDIVTYP
    THEN LET NDIVISION=NDIVTYP
    ELSE LET NDIVISION=NDIVMIN
366 NDIVISION=INT(NDIVISION)
380 IF NDIVISION<3 THEN LET NDIVISION=3
381 NDIVISION=INT(NDIVMIN)
383 LOCATE 8,12:PRINT "              "
384 LOCATE 8,12:PRINT NDIVISION
390 PASDIV=DELTAT/NDIVISION
400 PARTIE=1
410 GOSUB 2000 'CALCUL SUR PARTIE X
420 REM de t(ALPHA1) A t(BETA1)
430 DELTAT=T(2,CSN)-T(1,CSN)
440 UT=1
445 NDIVTYP=PERIODE*DIVMIN/DELTAT
450 NDIVMIN=MAX#*2*DELTAT
455 IF NDIVMIN<NDIVTYP
    THEN LET NDIVISION=NDIVTYP
    ELSE LET NDIVISION=NDIVMIN
458 NDIVISION=INT(NDIVISION)
460 IF NDIVISION<3 THEN LET NDIVISION=3
461 NDIVISION=INT(NDIVMIN)
463 LOCATE 8,12:PRINT "                 "
464 LOCATE 8,12:PRINT NDIVISION
470 PASDIV=DELTAT/NDIVISION
480 PARTIE=2
490 GOSUB 2000 'CALCUL SUR PARTIE X
500 REM de t(BETA1) A t(ALPHA2)
510 DELTAT=T(3,CSN)-T(2,CSN)
520 UT=0
525 NDIVTYP=PERIODE*DIVMIN/DELTAT
530 NDIVMIN=MAX#*2*DELTAT
535 IF NDIVMIN<NDIVTYP
    THEN LET NDIVISION=NDIVTYP
    ELSE LET NDIVISION=NDIVMIN
540 NDIVISION=INT(NDIVISION)
545 IF NDIVISION<3 THEN LET NDIVISION=3
546 NDIVISION=INT(NDIVMIN)
547 LOCATE 8,12:PRINT "              "
548 LOCATE 8,12:PRINT NDIVISION
550 PASDIV=DELTAT/NDIVISION
560 PARTIE=3
565 GOSUB 2000 'CALCUL SUR PARTIE X
570 REM de t(BETA1) A t(ALPHA2)
580 REM de t(ALPHA2) A t(BETA2)
590 DELTAT=T(4,CSN)-T(3,CSN)
600 UT=1
605 NDIVTYP=PERIODE*DIVMIN/DELTAT
610 NDIVMIN=MAX#*2*DELTAT
615 IF NDIVMIN<NDIVTYP
    THEN LET NDIVISION=NDIVTYP
    ELSE LET NDIVISION=NDIVMIN
620 NDIVISION=INT(NDIVISION)
625 IF NDIVISION<3 THEN LET NDIVISION=3
```

```
626 NDIVISION=INT(NDIVMIN)
627 LOCATE 8,12:PRINT "                    "
628 LOCATE 8,12:PRINT NDIVISION
630 PASDIV=DELTAT/NDIVISION
640 PARTIE=4
650 GOSUB 2000 'CALCUL SUR PARTIE X
660 REM de t(BETA2) A t=T
670 DELTAT=PERIODE-T(4,CSN)
680 UT=0
685 NDIVTYP=PERIODE*DIVMIN/DELTAT
690 NDIVMIN=MAX#*2*DELTAT
695 IF NDIVMIN<NDIVTYP
    THEN LET NDIVISION=NDIVTYP
    ELSE LET NDIVISION=NDIVMIN
700 NDIVISION=INT(NDIVISION)
705 IF NDIVISION<3 THEN LET NDIVISION=3
706 NDIVISION=INT(NDIVMIN)
707 LOCATE 8,12:PRINT "                    "
708 LOCATE 8,12:PRINT NDIVISION
710 PASDIV=DELTAT/NDIVISION
720 PARTIE=5
730 GOSUB 2000 'CALCUL SUR PARTIE X
740 REM: CALCULS..........
745 I2T=ABS(60*SUMCAR)
750 IRMS(NUMCYC)=SQR(I2T):SUMCAR=0
760 PMOY(NUMCYC)=60*SUMPMOY:SUMPMOY=0
770 VITESSE(NUMCYC)=DTETA*60/1800
771 LOCATE 14,60:PRINT "                    "
772 LOCATE 14,60:PRINT VITESSE(NUMCYC)
773 LOCATE 15,60:PRINT "                    "
774 LOCATE 15,60:PRINT IRMS(NUMCYC)
775 LOCATE 7,5:PRINT NUMCYC
780 TEMPS=0:LOCATE 5,27:PRINT TEMPS
790 CPTDTA=CPTDTA+1
800 IF CPTDTA=3 THEN GOTO 830
810 DTETA=DTETA+SUMDTETA:SUMDTETA=0:CPTDTA=0
815 DTETA=.9*185
820 GOSUB 1000 'CALCULER MATRICES A
822 LOCATE 5,14:PRINT NUMCYC
830 NEXT NUMCYC
840 FOR N=0 TO INDEXVAL-1
850 WRITE #1,VECVAL(N)
860 NEXT N
870 WRITE #1,1000000!,1000000!,1000000!,1000000!
880 WRITE #1,NBCYCSIM
890 FOR N=0 TO NBCYCSIM-1
900 WRITE #1,N,IRMS(N),PMOY(N),VITESSE(N)
910 NEXT N
920 PRINT "FIN DE SIMULATION; FICHIER DE SORTIE: DATAM"
925 CLOSE
930 END

1000 REM CALMATA ; CALCULER LES MATRICES A

1010 A(1,2,1)=A12*DTETA
1020 A(2,1,1)=A21*DTETA
1030 A(2,3,1)=A23*DTETA
1040 A(3,2,1)=A32*DTETA
1050 A(1,1,1)=A11
1060 A(1,3,1)=A13
1070 A(2,2,1)=A22
1080 A(3,1,1)=A31
1090 A(3,3,1)=A33
1100 MAX#=ABS(A(1,1,1))
1110 FOR I=1 TO 3
```

```
1120 FOR J=1 TO 3
1130 IF ABS(A(I,J,1))>ABS(MAX#)
     THEN LET MAX#=ABS(A(I,J,1))
1140 NEXT J
1150 NEXT I
1155 IF MAX#=0 THEN LET MAX#=1
1156 LOCATE 7,5:PRINT "                    "
1157 LOCATE 7,5:PRINT MAX#
1160 FOR I=1 TO 3
1170 FOR J=1 TO 3
1180 A(I,J,1)=A(I,J,1)/MAX#
1185 LOCATE 20,1:PRINT A(I,J,1)
1190 NEXT J
1200 NEXT I
1300 FOR K=2 TO 5
1310 FOR I=1 TO 3
1320 FOR J=1 TO 3
1330 A(I,J,K)=A(I,1,1)*A(1,J,K-1)+A(I,2,1)*A(2,J,K-1)
1331 A(I,J,K)=A(I,J,K)+A(I,3,1)*A(3,J,K-1)
1335 REM PRINT I,J,K,A(I,J,K)
1340 NEXT J
1350 NEXT I
1360 NEXT K
1370 RETURN


2000 REM CALCUL DE LA PARTIE X........
2010 IF UT=0 THEN LET AMPLITUDE=0
     ELSE AMPLITUDE=SQR(2)*120
2020 T(0,CSN)=0:T(5,CSN)=PERIODE
2030 REM T(i,csn) CONTIENT LES INSTANTS D'ARRIVEE
2040 REM DES EVENEMENTS NECESSAIRES POUR OBTENIR CSN
2041 VALIDER=0:DPCALPHI=1
2050 FOR DIV=1 TO NDIVISION
2051 VALIDER=0
2060 I=DIV
2070 T0=(I-1)*PASDIV+T(PARTIE-1,CSN)
2080 TF=I*PASDIV+T(PARTIE-1,CSN)

2090 REM====CALCULER X(i) A TF POUR i=1 A 3========

2100 H=(TF-T0)/NINT 'ON CONSIDERE 10 INTERV. D'INTEGR.
2110 FOR N=0 TO NINT
2120 TEMPS=T0+N*H
2130 U=AMPLITUDE*SIN(2*PI*(TF-N*H)/PERIODE)
2140 IF DPCALPHI=1 THEN GOSUB 3000 ' CALCULER PHI(TEMPS)
2150 Y(1,N)=(PHI(1,1,N)*B1+PHI(1,2,N)*B2+PHI(1,3,N)*B3)*U
2160 Y(2,N)=(PHI(2,1,N)*B1+PHI(2,2,N)*B2+PHI(2,3,N)*B3)*U
2170 Y(3,N)=(PHI(3,1,N)*B1+PHI(3,2,N)*B2+PHI(3,3,N)*B3)*U
2175 VALIDER=0'NORMALEMENT DOIT ÊTRE A 1
2176 IF FLAGERREUR=1 THEN PRINT
     "ERREUR DANS CALCUL DE PHI (AUGM. NDIVISION)"
2180 NEXT N
2190 X1=X(1):X2=X(2):X3=X(3) ' VALEURS PRECEDENTES.
2200 FOR N=1 TO 3
2210 XIPRIME=(H/3)*(Y(N,0)+4*(Y(N,1)+Y(N,3)+Y(N,5)
             +Y(N,7)+Y(N,9))+2*(Y(N,2)+Y(N,4)+Y(N,6)
             +Y(N,8))+Y(N,10))
2220 X(N)=PHI(N,1,NINT)*X1+PHI(N,2,NINT)*X2
        +PHI(N,3,NINT)*X3+XIPRIME
2225 LOCATE 4+N,54:PRINT "                 "
2226 LOCATE 4+N,54:PRINT X(N)
2230 NEXT N
2235 LOCATE 5,27:PRINT "                   "
2236 LOCATE 5,27:PRINT TEMPS
2237 LOCATE 8,54:PRINT "                   "
```

```
2238 LOCATE 8,54:PRINT U
2240 DPCALPHI=0:VALIDER=0
2242 VECVAL(INDEXVAL)=NUMCYC
2243 INDEXVAL=INDEXVAL+1
2245 VECVAL(INDEXVAL)=U
2250 INDEXVAL=INDEXVAL+1
2260 VECVAL(INDEXVAL)=TEMPS
2270 INDEXVAL=INDEXVAL+1
2280 VECVAL(INDEXVAL)=X(1)
2290 INDEXVAL=INDEXVAL+1
2300 REM VECVAL(INDEXVAL)=X(2)
2310 REM INDEXVAL=INDEXVAL+1
2320 REM VECVAL(INDEXVAL)=X(3)
2330 REM INDEXVAL=INDEXVAL+1
2340 IF INDEXVAL<1995 THEN GOTO 2390
2350 FOR N=0 TO INDEXVAL-1
2360 WRITE #1,VECVAL(N)
2370 NEXT N
2380 INDEXVAL=0
2390 DDTETA=-K6*DTETA+TEXT/J-K5*X(1)*X(2)
2400 SUMDTETA=SUMDTETA+DTETA*PASDIV
2410 SUMPMOY=U*X(1)*PASDIV+SUMPMOY
2420 SUMCAR=ABS(X(1)*X(1)*PASDIV+SUMCAR)
2430 LOCATE 10,1:PRINT "                    "
2433 LOCATE 10,1:PRINT DIV
2450 NEXT DIV
2460 RETURN


3000 REM CALCULER PHI DE T
     LA MATRICE DE TRANSITION D'ETAT

3001 FLAGERREUR=0
3010 FOR I=1 TO 3
3020 FOR J=1 TO 3
3025 TR=TEMPS-T0
3026 PART1=A(I,J,1)*MAX#*TR
3027 PART2=A(I,J,2)*(MAX#^2)*(TR^2)/2
3028 PART3=A(I,J,3)*(MAX#^3)*(TR^3)/(3*2)
3029 PART4=A(I,J,4)*(MAX#^4)*(TR^4)/(4*3*2)
3030 PART5=A(I,J,5)*(MAX#^5)*(TR^5)/(5*4*3*2)
3031 IF PHI(I,J,N)>=0 THEN LET PHIPRE(I,J)=PHI(I,J,N)
     ELSE PHIPRE(I,J)=1
3032 PHI(I,J,N)=ID(I,J)+PART1+PART2+PART3+PART5
3040 REM .......APPROXIMATION DE EXP(A*TEMPS) ..
3050 NEXT J
3051 NEXT I
3052 IF VALIDER=0 THEN GOTO 3079
3061 FOR I=1 TO 3
3062 FOR J=1 TO 3
3063 QQ=((PHI(I,J,N)>0 AND PHI(I,J,N)>PHIPRE(I,J))
        OR (PHI(I,J,N)<0))
3064 IF QQ=0 THEN GOTO 3074
3065 PHI(I,J,N)=PHIPRE(I,J)*((NINT-N)/(NINT+1-N))
3066 FLAGERREUR=1
3074 PHIPRE(I,J)=PHI(I,J,N)
3075 NEXT J
3076 NEXT I
3079 FOR J=1 TO 3
3080 LOCATE (14-1+J),11:PRINT "        "
3081 LOCATE (14-1+J),11:PRINT PHI(1,J,N)
3091 NEXT J
3092 LOCATE 17,14:PRINT "                "
3093 LOCATE 17,14:PRINT FLAGERREUR
3094 RETURN
```

ÉCOLE POLYTECHNIQUE DE MONTRÉAL
3 9334 00290805 9

CA2P
UP 8
1989
D878