

50376
1994
149

50376 1994 149

50376
1994
149

N° d'ordre : 1310

THESE

présentée à

L'UNIVERSITE DES SCIENCES ET TECHNIQUES DE LILLE

Spécialité : Productique : Automatique et Informatique Industrielle

par Vincent ABADIE

Ingénieur IDN

COMMANDE DES SYSTEMES CONTINUS
A ENTREES BINAIRES.

APPLICATION AUX MACHINES ELECTRIQUES



Soutenue le 17/6/1994 devant le jury d'examen :

M. BORNE

M. BERTRAND

M. FOSSARD

Mme DAUPHIN-TANGUY

M. DESCUSSE

M. ROMBAUT

M. STAROSWIECKI

M. VIDAL

Président du Jury

Rapporteur

Rapporteur

Examineur, directeur de thèse

Examineur

Examineur

Examineur

Examineur

AVANT-PROPOS

AVANT-PROPOS

Le travail présenté dans ce mémoire a été effectué au Laboratoire d'Automatique et d'Informatique Industrielle de Lille à l'Ecole Centrale de Lille sous la direction du Professeur Geneviève DAUPHIN-TANGUY

Nous lui en sommes particulièrement reconnaissants. Il a toujours, et en toutes circonstances, été le soutien qui nous a permis de mener à bien cette entreprise et toute l'aide qu'il a pu nous apporter a été une condition essentielle pour la réalisation de ce travail.

Qu'il nous soit permis de remercier Monsieur le Professeur Pierre BORNE, Directeur Scientifique de l'Ecole Centrale de Lille, pour son amitié, ses précieux conseils, son intérêt pour nos travaux de recherche et pour nous avoir fait l'honneur d'être président de notre jury de Thèse.

Nous tenons à remercier très vivement M. André FOSSARD, Professeur à l'ENSAE de Toulouse, de l'honneur qu'il nous a fait en acceptant d'être Rapporteur de ce mémoire, ainsi que pour les conseils qu'il a su nous prodiguer lors de notre entrevue.

Nous tenons à remercier très chaleureusement M. Pierre BERTRAND, Directeur de Recherche au CNRS, de l'honneur qu'il nous a fait en acceptant d'être Rapporteur de ce mémoire ainsi que pour ses précieux conseils.

Qu'il nous soit permis de remercier également M. DESCUSSE pour l'honneur qu'il nous a fait en acceptant d'être membre de notre jury.

Nous sommes également très reconnaissants au Professeur ROMBAUT, d'une part pour son amitié, son intérêt pour nos recherches et pour l'honneur qu'il nous a fait en acceptant d'être membre de notre jury de Thèse.

Nous tenons également à assurer toute notre reconnaissance aux Professeurs VIDAL et STAROWIECKI pour avoir accepté d'être membres de notre jury.

Enfin, qu'il nous soit permis de remercier les membres du LAII dans leur totalité pour leur soutien et leurs aides diverses dans toutes nos activités de recherche.

SOMMAIRE

INTRODUCTION GENERALE

CHAPITRE I.....
SYSTEMES A COMMUTATION.....	2
CHAPITRE I.....	3
INTRODUCTION.....	3
I.1 SYSTEMES A COMMUTATION.....	3
1.1.1 Modélisation par bond-graphs.....	3
I.1.1.1 Préliminaires.....	3
I.1.1.2 Modélisation bond-graph d'un élément de commutation.....	4
I.1.1.2.1 Causalité variable.....	4
I.1.1.2.2 Causalité unique.....	6
I.1.1.3 Exemple d'application.....	8
1.1.2 Commande booléenne.....	11
I.2 COMMANDE PAR VALEUR MOYENNE.....	13
1.2.1 Préliminaires.....	13
1.2.2 Commande rapprochée d'un convertisseur.....	13
I.2.2.1 Position du problème.....	13
I.2.2.2 Commande par hystérésis.....	14
I.2.2.3 Commande par M.L.I.....	15
I.3 COMMANDE PAR MODES GLISSANTS.....	16
1.3.1 Systèmes à structure variable.....	17
1.3.2 Elaboration de la loi de commande.....	17
I.3.2.1 Cas linéaire.....	17
I.3.2.1.1 Cas particulier : systèmes linéaires monovariabiles.....	18
I.3.2.1.2 Exemple d'application.....	19
I.3.2.2 Cas des systèmes non-linéaires affines en la commande.....	21
I.4 MODES GLISSANTS POUR LES SYSTEMES A ENTREES BOOLEENNES.....	22
1.4.1 Cas linéaire.....	23
Cas mono-entrée.....	24
1.4.2 Exemple d'application.....	24
I.4.2.1 Exemple 1.....	24
I.4.2.2 Exemple 2.....	26
1.4.3 Cas bilinéaire.....	27
I.4.3.1 Cas monoentrée.....	28
I.4.3.2 Cas multi-entrées.....	30
CONCLUSION.....	32
CHAPITRE II.....
INTRODUCTION.....	33
II.1 POSITION DU PROBLEME.....	33
II.2 DOMAINE DE CONVERGENCE.....	34
11.2.1 Définitions.....	34
1) Atteignabilité.....	34
2) Commandabilité d'un système.....	34
3) Accessibilité.....	35
4) Domaine de convergence.....	35
Remarques :.....	35
Remarque 1 :.....	35
Remarque 2 :.....	36
11.2.2 Résolution de l'équation d'état.....	36
II.2.2.1 Cas général.....	36
II.2.2.2 Hypothèse de stabilité :.....	37
11.2.3 Domaine de convergence de l'état.....	37
II.3 COMMANDE EN BOUCLE OUVERTE.....	38
11.3.1 Commande vers un objectif fixe.....	38
Algorithme de commande 1 :.....	38
Démonstration :.....	39
11.3.2 Domaine de commandabilité.....	43
11.3.3 Commande en sortie.....	45

II.3.3.1	Domaine de convergence	45
II.3.3.2	Algorithme de commande	46
II.3.4	Choix de la commande	46
II.3.4.1	Degrés de liberté	46
II.3.4.2	Contraintes	47
II.3.4.3	Critères de choix	50
a)	Exclusion d'interrupteurs	50
b)	Optimisation du choix	52
→	Energie en entrée	52
Exemple :		54
Remarque concernant l'optimisation en sortie :		55
II.3.5	Application à la poursuite de trajectoires	56
II.4	COMMANDE EN BOUCLE FERMEE	57
II.4.1	Elaboration d'un algorithme de régulation	57
II.4.2	Poursuite de trajectoire	59
II.5	EXEMPLES D'APPLICATION	61
II.5.1	Exemple 1	61
II.5.2	Exemple 2	64
CONCLUSION		68

CHAPITRE III.....

INTRODUCTION	69
III.1 COMMANDE DES SYSTEMES BILINEAIRES A COMMUTATION	69
III.1.1 Préliminaires	69
III.1.2 Elaboration de l'algorithme	70
III.1.2.1 Cas où toutes les matrices $(\mathbf{A} + \mathcal{N}_k)$ sont asymptotiquement stables	70
III.1.2.2 Cas général	72
a) Cas où toutes les matrices $(\mathbf{A} + \mathcal{N}_i)$ $i \in \{1, \dots, 2^m\}$ sont inversibles	72
b) Cas où il existe un i tel que $(\mathbf{A} + \mathcal{N}_i)$ est non inversible	73
Remarques concernant le domaine de convergence	73
III.1.3 Exemples d'application	73
III.1.3.1 Cas où tous les $(\mathbf{A} + \mathcal{N}_i)$ sont asymptotiquement stables	73
a) Exemple n°1	73
b) Exemple n°2	74
III.1.3.2 Cas général	76
III.1.4 Conclusion du cas bilinéaire	78
III.2 SYSTEMES NON-LINEAIRES : LINEARISATIONS PAR MORCEAUX	79
III.2.1 Linéarisation du système étudié	79
III.2.2 Commande en boucle ouverte	79
III.2.2.1 Régulation	79
III.2.2.2 Poursuite de trajectoires	81
III.2.3 Commande en boucle fermée	82
III.2.3.1 Elaboration de l'algorithme	82
III.2.3.2 Convergence de l'algorithme	84
III.2.4 Exemple d'application	85
III.3 SYSTEMES NON-LINEAIRES : LINEARISATION PAR RETOUR D'ETAT	87
III.3.1 Position du problème	87
III.3.2 Loi de commande continue	88
III.3.2.1 Commandes à valeurs dans \mathbb{R}^m	88
III.3.2.1.1 Systèmes carrés	89
a) Cas où la somme des degrés relatifs vaut n :	90
b) Cas où la somme des degrés relatifs est inférieure à n :	90
III.3.2.1.2 Systèmes non carrés	90
III.3.2.1.3 Dynamique de l'erreur	91
III.3.2.2 Commandes à valeurs dans $[0, 1]^m$	91
III.3.2.2.1 Position du problème	91
III.3.2.2.1.1 Exemple introductif : cas scalaire	91
III.3.2.2.1.2 Adaptation de la dynamique	92
III.3.2.2.1.3 Exemples d'application	93
III.3.2.2.2 Cas général	94
III.3.2.2.2.1 Expression de $u_c(t)$	95

a) Cas où l'on peut exprimer $x(t)$	95
b) Cas où l'on ne peut pas exprimer directement $x(t)$	96
III.3.3.2 Exemples d'application	96
Exemple 1 : $x(t)$ directement exprimable	96
Exemple 2 : $x(t)$ exprimable par intégration	97
Exemple 3 : $x(t)$ non directement exprimable	100
III.3.4 Binarisation de la commande	102
III.3.4.1 Valeur moyenne	102
III.3.4.2 Commande booléenne	103
III.3.4.3 Convergence de l'algorithme	104
III.3.4.3.1 Cas scalaire	104
III.3.4.3.2 Cas multivariable	105
III.3.5 Mise en place de l'algorithme	106
III.3.6 Exemple d'application	109
III.3.6.1 Exemple 1	109
III.3.6.2 Exemple 2	112
III.3.6.3 Exemple 3	114
III.3.6.3.1 Zéros dynamiques	115
III.3.6.3.2 Simulation du système à entrées binaires	116
CONCLUSION	117

CHAPITRE IV

INTRODUCTION	118
IV.1 CAS DE LA MACHINE SYNCHRONE	118
IV.1.1 Modélisation par bond-graph	118
IV.1.1.1 Schéma électrique de l'onduleur	118
IV.1.1.2 Représentation bond-graph de l'onduleur	119
IV.1.1.3 Association onduleur-machine synchrone	120
IV.1.1.4 Modèle bond-graph de l'ensemble onduleur-moteur synchrone	120
IV.1.2 Equation d'état du système	122
IV.1.3 Commande par retour d'état	124
IV.1.3.1 Commande continue	124
IV.1.3.2 Etude des zéros dynamiques	125
IV.1.3.3 Binarisation de la commande	126
IV.1.3.3 Application numérique et simulation	128
IV.1.3.4 Conclusion	130
IV.1.4 Commande booléenne par modes glissants	131
IV.1.4.1 Elaboration de l'algorithme	131
IV.1.4.2 Simulations	132
IV.1.4.2.1 Système linéarisé :	132
IV.1.4.2.2 Système non-linéaire :	134
IV.1.4.3 Conclusion	134
IV.1.5 Commande par linéarisations par morceaux	135
IV.1.5.1 Elaboration de l'algorithme	135
IV.1.5.2 Amélioration de l'algorithme	138
IV.1.5.3 Simulations	139
IV.1.5.3.1 Simulation 1	139
IV.1.5.3.2 Simulation 2	140
IV.1.5.4 Conclusion	141
IV.2 COMMANDE DE LA MACHINE ASYNCHRONE	141
IV.2.1 Modélisation par bond-graph	142
IV.2.1.1 Bond-graph du système	142
IV.2.1.2 Equation d'état	143
IV.2.2 Commande par linéarisations successives	144
IV.2.2.1 Modèles linéarisés	144
IV.2.2.2 Commande en vitesse	145
IV.2.2.3 Calcul des points limites	145
IV.2.2.4 Problème du glissement	147
IV.2.3 Commande par MLI	147
IV.2.4 Simulations	149
IV.2.4.1 Méthode de linéarisations successives	149
IV.2.4.1.1 Régulation	149
IV.2.4.1.2 Poursuite de trajectoires	151
IV.2.4.2 Commande par MLI	151

IV.2.4.2.1 Régulation	151
IV.2.4.2.2 Poursuite de trajectoires.....	153
IV.2.4.3 Effet d'une variation des paramètres.....	153
IV.3 COMPARAISON DES RESULTATS	154
IV.3.1 Moteur synchrone	155
IV.3.2 Moteur asynchrone	155
IV.4 IMPLANTATION SUR SITE	156
CONCLUSION.....	157

CONCLUSION GENERALE

BIBLIOGRAPHIE

ANNEXES

INTRODUCTION GENERALE

INTRODUCTION GENERALE

Les travaux menés à l'Ecole Centrale de Lille issus d'une collaboration entre l'équipe du Professeur Rombaut du LEEP et l'équipe du Professeur Dauphin-Tanguy du LAIL, ont abouti à des résultats concernant la modélisation par bond-graphs de l'association convertisseur de puissance-machine électrique [Dauphin-Tanguy et al, 1989]. Les modèles d'état obtenus à partir de cette modélisation conduisent à des équations continues en l'état à commandes booléennes. L'obtention de ces modèles intégrant directement l'état des commutateurs nous a conduit à nous intéresser au problème de l'élaboration de lois de commande pour les systèmes continus à commandes binaires.

Une étude bibliographique a fait apparaître que peu de travaux existaient dans ce domaine. D'une part, la commande par Modulation de Largeur d'Impulsion (MLI ou PWM) permet de piloter les systèmes à commutation, mais elle ne considère pas la commande d'un point de vue global. D'autre part, quelques approches ont été menées [Sira-Ramirez, 1988, 1989], [Sabanovic, 1993] en considérant la structure variable de la commande par une application de la technique des modes glissants. Cependant les travaux effectués présentent l'inconvénient de pas traiter de manière générale le cas multivariable et de ne pas fournir des conditions d'existence de surfaces de glissement directement exploitables.

Dans le premier chapitre, nous rappelons tout d'abord la méthode de modélisation des éléments de commutation utilisée. Les techniques de commande existantes des systèmes à commutation sont ensuite évoquées. Nous exposons dans un premier temps les méthodes classiques de commande des commutateurs en électronique de puissance. La méthode générale de commande par modes glissants est ensuite présentée afin d'introduire une méthode d'application de cette technique au cas des systèmes à commandes booléennes. Nous proposons alors une étude de l'existence de régimes de glissement pour certaines logiques de commutation.

Au cours du deuxième chapitre, après avoir introduit la notion de domaine de convergence, nous élaborons un algorithme nouveau de commande en boucle ouverte pour les systèmes linéaires booléens en la commande fondé sur un principe de calcul de coordonnées barycentriques de la consigne à atteindre. Ensuite, nous définissons un algorithme en boucle fermée qui nous permet de faire de la poursuite de trajectoires.

Le chapitre III est consacré à l'étude des systèmes non-linéaires booléens en la commande. Nous traitons d'abord le cas des systèmes bilinéaires en élaborant un algorithme analogue à

celui du linéaire. Nous élaborons, dans la deuxième partie du chapitre, un algorithme fondé sur une technique de linéarisation par morceaux du système non-linéaire et une application de l'algorithme en boucle fermée du chapitre II sur les systèmes linéarisés.

Dans la troisième partie du chapitre, le cas des systèmes non-linéaires affines en la commande est envisagé et nous proposons un algorithme faisant intervenir une commande continue équivalente calculée à partir d'une linéarisation par retour d'état et une binarisation de cette commande.

Dans le quatrième chapitre, ces algorithmes sont appliqués à la commande de moteurs électriques. Après une brève évocation du problème de la modélisation, diverses techniques de commande en vitesse de la machine synchrone (MLI, Modes Glissants) sont utilisées en simulation, ainsi que les algorithmes du chapitre III (linéarisation par retour d'état, linéarisations par morceaux), afin d'effectuer une comparaison entre ces différentes lois de commande.

Nous appliquons ensuite nos méthodes en simulation à la commande en vitesse d'une machine asynchrone, et les résultats sont comparés à ceux obtenus avec une commande classique par MLI.

Une implantation des lois de commande que nous avons élaborées est menée actuellement au LEEP, en collaboration avec l'équipe du Professeur Rombaut sur la commande en vitesse et la commande en couple d'un moteur asynchrone associé à un onduleur de tension.

CHAPITRE I

SYSTEMES A COMMUTATION

CHAPITRE I

INTRODUCTION

Dans ce premier chapitre nous allons présenter dans un premier temps les raisons qui nous ont amené à étudier les systèmes continus à structure booléenne en entrée. Pour cela, nous allons montrer, en prenant l'exemple des convertisseurs en électronique, que l'outil bond-graph permet une analyse des systèmes comportant des éléments de commutation utilisés pour la commande, en donnant une équation d'état continue à entrées booléennes.

Nous allons ensuite évoquer les méthodes existantes pouvant se rapprocher de notre problème, car faisant intervenir des commutations dans la commande. Nous rappellerons d'abord quelques techniques de commande de ces systèmes en valeur moyenne (MLI, hystérésis) utilisées en électronique de puissance. Nous exposerons ensuite les principes généraux de la commande par modes glissants des systèmes à structure variable et nous proposerons une méthode permettant d'appliquer cette technique à la commande des systèmes à entrées booléennes.

I.1 SYSTEMES A COMMUTATION

I.1.1 Modélisation par bond-graphs

I.1.1.1 Préliminaires

La modélisation des systèmes dis-continus, c'est à dire présentant des discontinuités dans leur évolution dans le temps, (vannes ou valves en hydraulique, butées ou chocs en mécanique, convertisseurs en électronique de puissance...) n'est pas simple et a fait l'objet de nombreuses discussions et travaux.

Des séminaires de réflexion sur l'intérêt de l'approche bond-graph pour résoudre ce problème sont organisés (Journées AFCET/SEE, Janvier 1994-Sessions spécialisées aux congrès SCS de San Diego, Janvier 1993, IEEE/SMC-Le Touquet-Octobre 1994, SCS-Las Vegas-Janvier 1995...). Les différents travaux trouvés dans la littérature font apparaître deux classes de résultats : les méthodes conduisant à une représentation multi-modèles [Asher et al, 1989,

1991] , [Michel et al, 1993], [Strömberg et al, 1993] et les méthodes conduisant à un modèle unique valable quel que soit l'état des commutateurs [Lorenz, 1993].

Les travaux effectués à l'Ecole Centrale de Lille, en collaboration entre l'équipe du Professeur Rombaut du LEEP et l'équipe du Professeur Dauphin-Tanguy du LAIL, concernent les convertisseurs de l'électronique de puissance et appartiennent à la deuxième classe de méthodes. Pour plus de précisions concernant les convertisseurs de puissance le lecteur pourra se référer aux articles et ouvrages suivants [Middlebrock et al, 1977], [McMurray, 1988], [Rashid, 1988], [Mohan et al, 1989], [Seguier et al, 1989], [Bühler et al, 1991].

Le modèle bond-graph [Paynter, 1961] est un intermédiaire entre le système physique et les modèles mathématiques habituellement utilisés par les automaticiens (matrices de transfert en linéaire, équations d'état en linéaire et non-linéaire). Cet outil permet de représenter et de coupler des éléments pouvant appartenir à différents domaines de la physique (mécanique, hydraulique, électronique,...) et permet une déduction immédiate [Rosenberg et al, 1983] des équations mathématiques du système.

Le bond-graph est un modèle de connaissance permettant de fournir les réponses d'un système à des entrées connues, directement si on dispose d'un outil de simulation intégrant le formalisme bond-graph du type ARCHER/MATLAB [Azmani et al, 1992], ENPORT [Rosenberg, 1990], CAMPG/ACSL [Granda, 1985, 1993], TUTSIM [Univ. of Twente, 1985-1992], DESIS [Delgado, 1990], CAMAS [Univ. of Twente, 1991] ou en simulant les équations d'état déterminées à partir du bond-graph en utilisant un outil de simulation "classique" du type MATLAB [Mathworks, 1984-1993], SIMNON [Elmqvist, 1990], ACSL [Mitchell Associates, 1986-1993], NEPTUNIX [Nahklé, 1986].

I.1.1.2 Modélisation bond-graph d'un élément de commutation

I.1.1.2.1 Causalité variable

En électronique de puissance, il existe deux démarches classiques reposant sur la même hypothèse de composant idéal (sans perte en fonctionnement) pour modéliser les éléments de commutation.

La première, "à topologie variable" conduit à un modèle différent pour chaque état du système (2^r modèles, si r composants). La seconde consiste à représenter l'élément de commutation comme une résistance nulle si le composant est passant et infinie s'il est bloqué.

Ces deux techniques présentent chacune un inconvénient majeur : la première n'est pas utilisable pour la commande, la deuxième introduit de très petites constantes de temps rendant la simulation difficile.

La méthode développée à l'Ecole Centrale de Lille, consiste à représenter le composant discontinu sous forme de l'association de deux éléments bond-graph. Le premier est un élément indiquant la perte de puissance dans le composant de commutation lorsqu'il est en fonctionnement, le second représente la logique de commutation et est caractérisé par un booléen m [Dauphin-Tanguy et al, 1989, 1993], [Castelain et al, 1990], [Ducreux et al, 1992, 1993].

Considérons un élément de commutation dont la caractéristique est donnée par la fig.1.1 :

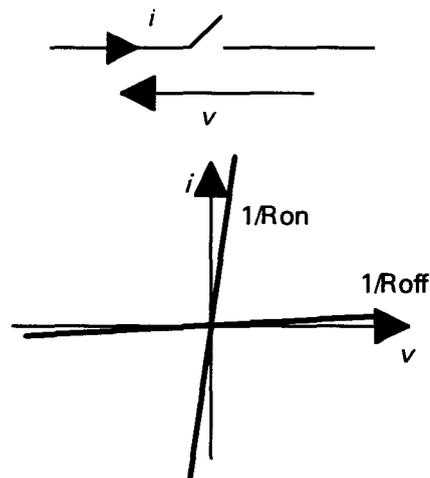


Fig.1.1 : Caractéristique d'un élément de commutation non-idéal.

R_{on} et R_{off} sont les résistances internes de l'élément de commutation dont les valeurs sont respectivement faibles et grandes.

L'interrupteur ainsi considéré agit alors comme une résistance ou une conductance et impose au reste du circuit une faible tension ou une faible intensité. Pour chaque cas de fonctionnement, nous pouvons donner une représentation bond-graph (fig.1.2a, 1.2b) :

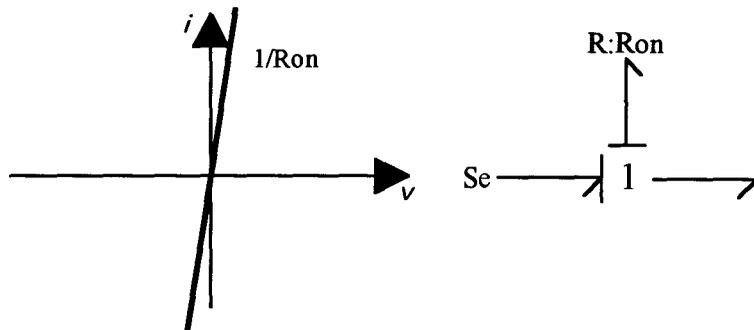


Fig.1.2a : Interrupteur passant.

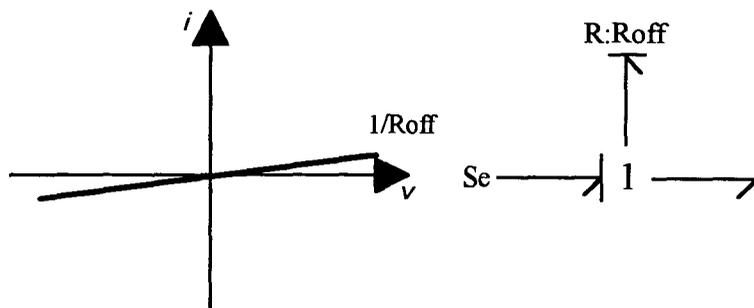


Fig.1.2b : Interrupteur bloqué.

Cette démarche présente l'inconvénient de donner 2^r bond-graphs si nous disposons de r éléments de commutation dans le circuit.

1.1.1.2.2 Causalité unique

En général, la conductance interne de l'interrupteur lorsque celui-ci est bloqué ($1/R_{off}$) est très petite, ce qui n'est pas le cas de la résistance (R_{on}) lorsqu'il est passant. La caractéristique de la fig.1.1 devient alors (fig.1.3) :

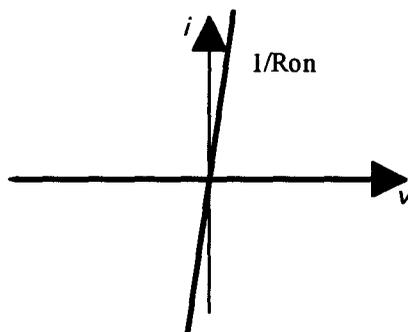


Fig.1.3 : Caractéristique d'un élément de commutation (état bloqué supposé idéal).

Une méthode pour modéliser ce composant à l'aide du bond-graph a été proposée dans [Dauphin-Tanguy et al, 1989], (fig.1.4) :

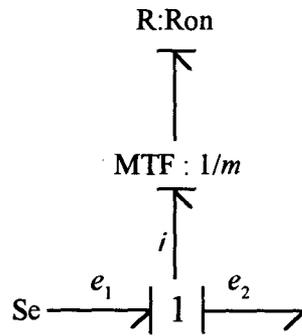


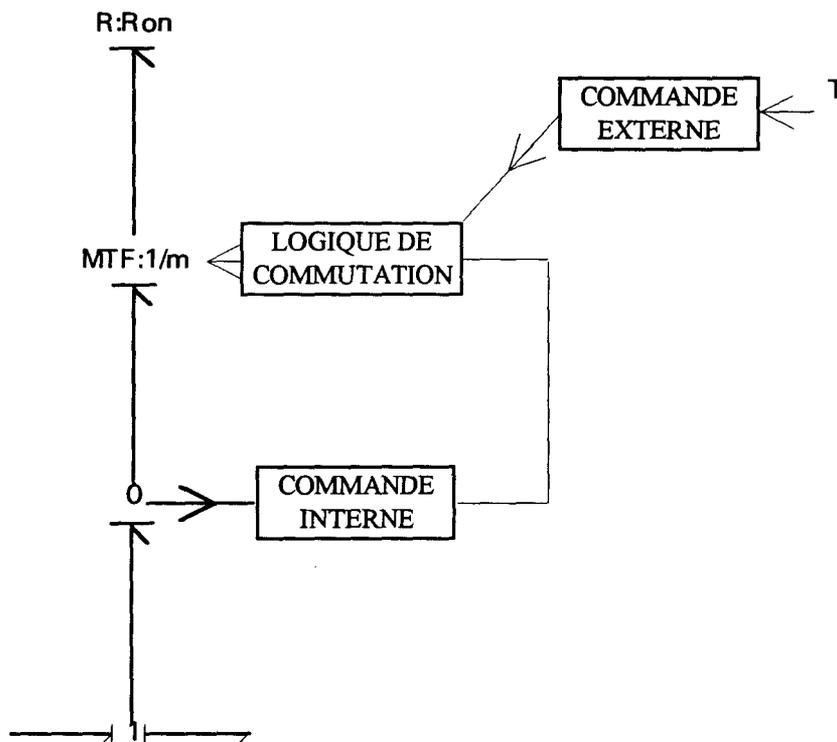
Fig. 1.4 : Modèle bond-graph d'un interrupteur (causalité unique).

m est un booléen qui vaut 0 si l'élément de commutation est bloqué et 1 dans le cas contraire. Ce modèle nous permet de déduire l'équation :

$$i = \frac{m^2}{R_{on}}(e_1 - e_2) \quad (1.1)$$

où $e_1 - e_2$ est la tension aux bornes du composant. La conductance lorsque l'interrupteur est passant vaut $\frac{1}{R_{on}}$ et 0 lorsqu'il est bloqué ce qui correspond bien à la caractéristique fig.1.3.

Ce modèle est valable quel que soit le type de composant (diode, thyristor, transistor de commutation). La différence intervient dans la logique de commutation, comme indiqué fig.1.5, donc sur les transitions du booléen m :



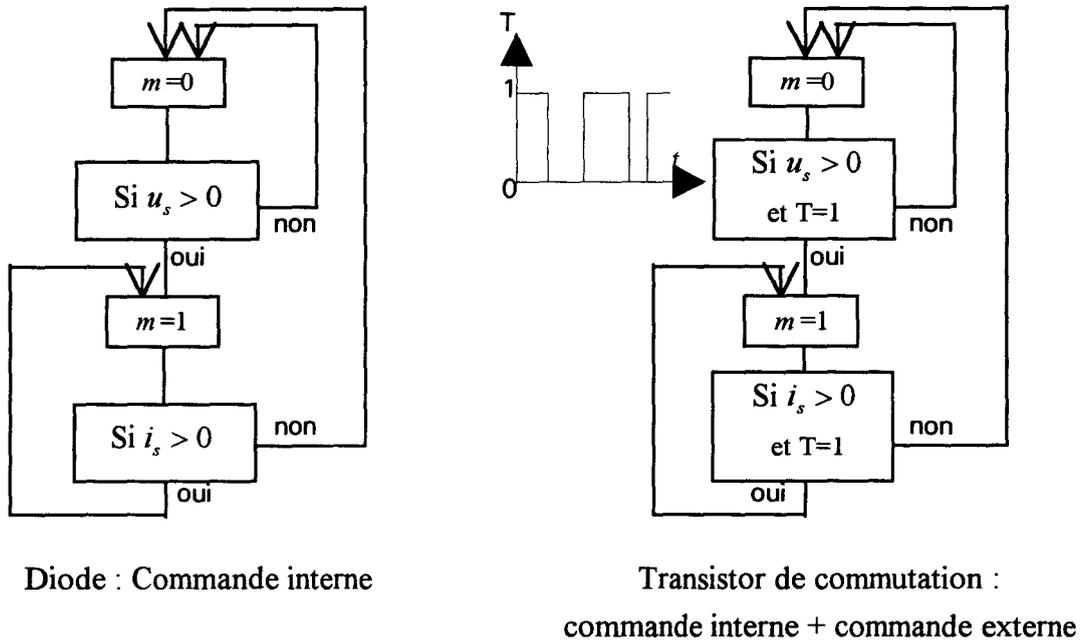


Fig.1.5 : Différentes logiques de commutation en électronique.

Cette démarche peut être étendue à d'autres domaines de la physique : une valve ou une vanne en hydraulique auront la même représentation bond-graph que ci-dessus. Pour une butée mécanique ou un relais de pression en mécanique, on associera le même élément conservatif de puissance qui introduira la logique de commutation (MTF) et un élément de stockage d'énergie potentielle (C).

Le choix d'une affectation unique de la causalité [Dauphin-Tanguy et al, 1993], valable quels que soient les états des commutateurs conduit à un bond-graph unique, et donc à un modèle d'état unique, valable quelle que soit la configuration du système. Ce modèle s'exprime sous la forme suivante :

$$\dot{x} = f(x, E, m_1, \dots, m_r) \tag{1.2}$$

où $x = (\phi^T, q^T)^T$ est le vecteur d'état issu du modèle bond-graph composé des variables d'état flux magnétiques dans les bobines et charges dans les condensateurs. E est le vecteur entrée et $m_i, i=1, \dots, r$, sont les booléens associés aux composants de commutation.

1.1.1.3 Exemple d'application

Considérons le système physique composé d'une association d'un convertisseur de puissance associé à une machine électrique (fig. 1.6) :

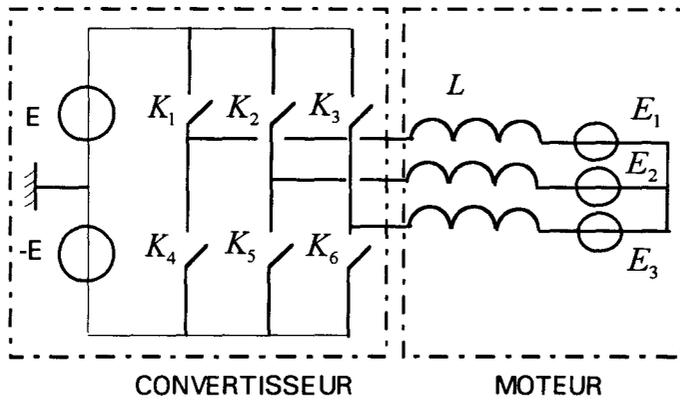


Fig 1.6 : Association convertisseur-machine électrique.

E et $-E$ sont deux sources de tensions continues en opposition et E_1, E_2, E_3 sont trois sources de tensions alternatives. $K_i, i=1, \dots, 6$ sont des éléments de commutation composés d'un transistor et d'une diode montés en anti-parallèle. Le bond-graph associé à ce système est représenté fig1.7 :

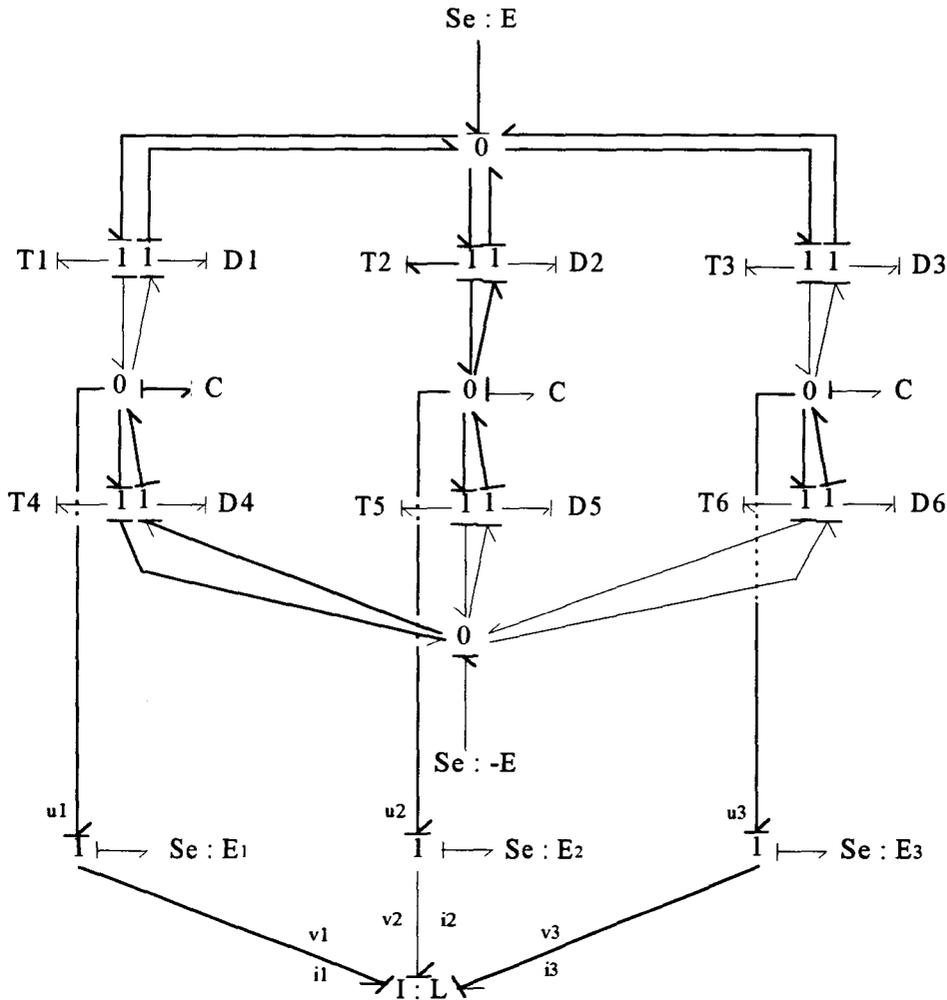


Fig 1.7 : Modèle bond-graph associé à la fig. 1.6.

Les transistors et diodes ont été représentés par D_i et T_i , $i=1,\dots,6$, mais ils sont en fait modélisés en accord avec le formalisme défini précédemment (fig.1.8a,b) :

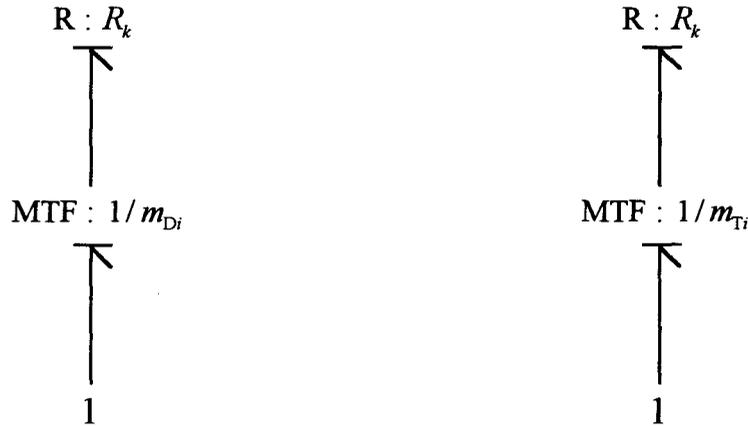


Fig.1.8a : Représentation d'une diode. Fig.1.8b : Représentation d'un transistor.

Les booléens m_{T_i} et m_{D_i} représentent l'état (passant, bloqué) des transistors T_i et des diodes D_i . Nous avons supposé que tous les éléments de commutation avaient la même résistance interne R_k .

Trois condensateurs C ont été ajoutés au circuit pour permettre une mesure des tensions simples aux bornes du moteur.

Le bloc multi-port $I : L$ est défini si on suppose le moteur équilibré ($i_1+i_2+i_3=0$), par les équations suivantes :

$$\begin{pmatrix} i_1 \\ i_2 \\ i_3 \end{pmatrix} = \frac{1}{3L} \begin{pmatrix} 2 & -1 & -1 \\ -1 & 2 & -1 \\ -1 & -1 & 2 \end{pmatrix} \begin{pmatrix} \phi_1 \\ \phi_2 \\ \phi_3 \end{pmatrix} \quad \text{et} \quad \begin{pmatrix} \phi_1 \\ \phi_2 \\ \phi_3 \end{pmatrix} = \frac{1}{3} \begin{pmatrix} 2 & -1 & -1 \\ -1 & 2 & -1 \\ -1 & -1 & 2 \end{pmatrix} \begin{pmatrix} v_1 \\ v_2 \\ v_3 \end{pmatrix} \quad (1.3)$$

avec $v_i = u_i - E_i$, $i=1,2,3$.

Les équations d'état peuvent être déduites du bond-graph :

$$\left\{ \begin{array}{l}
 \phi_1 = \frac{1}{3C}(2q_1 - q_2 - q_3) - E_1 \\
 \phi_2 = \frac{1}{3C}(-q_1 + 2q_2 - q_3) - E_2 \\
 \phi_3 = \frac{1}{3C}(-q_1 - q_2 + 2q_3) - E_3 \\
 \dot{q}_1 = -\left(\frac{m_1^2 + m_4^2}{R_k C}\right)q_1 - \frac{\phi_1}{L} + \left(\frac{m_1^2 - m_4^2}{R_k}\right)E \\
 \dot{q}_2 = -\left(\frac{m_2^2 + m_5^2}{R_k C}\right)q_2 - \frac{\phi_2}{L} + \left(\frac{m_2^2 - m_5^2}{R_k}\right)E \\
 \dot{q}_3 = -\left(\frac{m_3^2 + m_6^2}{R_k C}\right)q_3 - \frac{\phi_3}{L} + \left(\frac{m_3^2 - m_6^2}{R_k}\right)E \\
 \text{avec } m_i^2 = m_{Di}^2 + m_{Ti}^2, i = 1, \dots, 6.
 \end{array} \right. \quad (1.4)$$

I.1.2 Commande booléenne

Les booléens m_i peuvent être des grandeurs de commande, comme dans le cas des associations machines électriques-convertisseurs d'Electronique de Puissance [Ducreux et al, 1992, 1993].

On obtient alors des modèles d'état de la forme :

$$\dot{x} = f(x, u, E) \quad (1.5)$$

où u , vecteur de commande est à composantes booléennes.

Ce résultat obtenu au laboratoire est à l'origine de nos travaux et justifie l'intérêt que nous avons porté à cette classe particulière de modèles.

L'étude bibliographique que nous avons alors effectuée nous a montré que les travaux existant pouvant se rapprocher du problème étaient peu nombreux et pouvaient être séparés en deux catégories :

- Les méthodes utilisées pour la commande de machines électriques qui consistent à commander le convertisseur par valeur moyenne d'une commande continue.
- Les méthodes déterminant directement une commande booléenne.

Les méthodes appartenant à la première catégorie, la plus habituellement utilisée étant la commande par Modulation de Largeur d'Impulsion (MLI) [Skoog et al, 1970], [Capel et al, 1973], [Chauprade et al, 1980], [Leonhard, 1985], [Bose, 1986], [Kiel et al, 1987], [De Carli, 1990], [Cocquerelle, 1993] sont fondées sur le principe d'une commande rapprochée des éléments de commutation en valeur moyenne de l'entrée. Le système est alors assimilé à un système à entrées continues ce qui est représenté par la fig.1.9 :

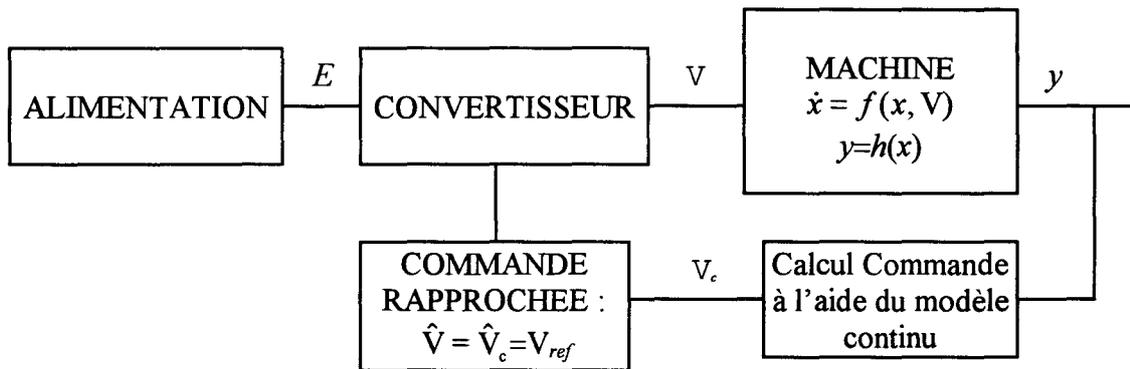


Fig.1.9 : Principe classique de commande des systèmes à commutation.

où E est la tension d'alimentation (généralement continue) du convertisseur, V l'entrée du système et V_c la commande continue calculée de valeurs moyennes respectives \hat{V} et \hat{V}_c .

Nous allons présenter dans la partie II.2 de ce chapitre les méthodes utilisées usuellement en électronique de puissance pour effectuer la commande rapprochée sous leur forme générale.

Les méthodes appartenant à la deuxième catégorie sont fondées sur un modèle à entrées booléennes comme présenté dans le cas de l'association machine-convertisseur (fig.1.10) :

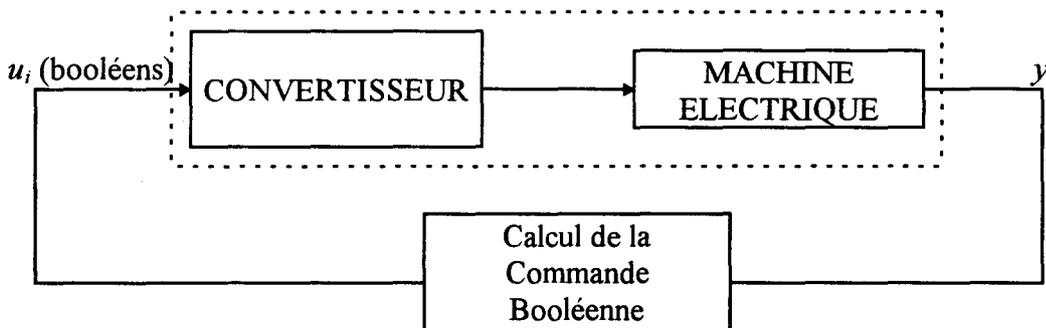


Fig.1.10 : Commande booléenne du système global.

Parmi ces dernières, on trouve la commande par modes glissants [Emelyanov, 1964], [Utkin, 1974], qui permet de commander les systèmes à structure variable (SSV). Les systèmes que nous étudions représentent un cas particulier de SSV où les composantes de l'entrée sont obligatoirement booléennes, et des méthodes d'application de la technique de commande par modes glissants peuvent être alors élaborées [Sira-Ramirez, 1987a, 1988], [Sabanovic et al, 1993]. Nous allons, dans la partie III de ce chapitre présenter le cas général de la commande par modes glissants puis mettre en évidence les techniques permettant de l'adapter aux systèmes à entrées booléennes.

I.2 COMMANDE PAR VALEUR MOYENNE

I.2.1 Préliminaires

Nous allons décrire dans ce paragraphe deux types de méthodes de commande rapprochée des convertisseurs de puissance mises en oeuvre dans le cas de structures de commande telles que celle présentée fig.1.9. La première dite "tout ou rien" ou "commande par hystérésis" présente l'avantage d'offrir une mise en oeuvre simple. La seconde, la technique de commande par modulation de largeur d'impulsion (MLI ou PWM), est la plus largement utilisée [Plunkett, 1977], [Holtz et al, 1983], [Kusho et al, 1983], [Sabanovic et al, 1981, 1984, 1988], [Shörner, 1986], [Nonaka, 1987], [Vorpérian, 1990], [Silva, 1992] et offre des performances intéressantes.

I.2.2 Commande rapprochée d'un convertisseur

I.2.2.1 Position du problème

Nous allons considérer pour cette étude une branche d'un convertisseur électrique du même type que celui représenté fig.1.6 comportant deux interrupteurs T1 et T4 qui seront supposés fonctionner de manière complémentaire, c'est à dire que lorsqu'un est bloqué, l'autre est passant et vice-versa.

La branche du convertisseur est représentée par la fig.1.11 :

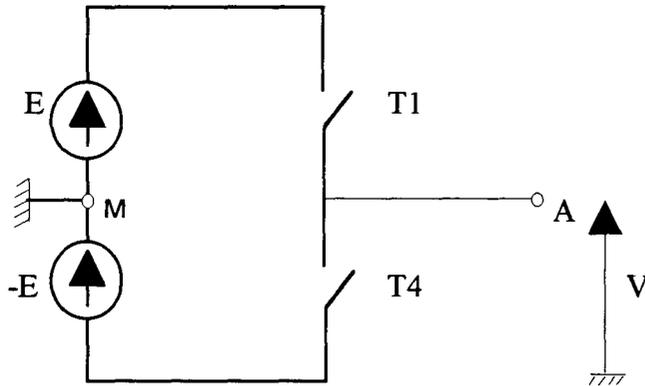


Fig.1.11 : Schéma d'une branche du convertisseur.

avec $V=V_A-V_M$. E est une tension continue. $T1$ et $T4$ représentent deux transistors de commutation. On désire alors commander V à partir de l'ouverture et la fermeture de $T1$ ($T4$ étant complémentaire). Si $T1$ est ouvert alors $T4$ est fermé et $V=-E$. Si $T1$ est fermé alors $V=E$. Nous allons voir qu'en choisissant des durées d'ouvertures et de fermetures de manières appropriées, il est possible de donner à V n'importe quelle valeur (en valeur moyenne) entre $-E$ et E .

Pour cela, dans chacune des méthodes que nous allons envisager, le domaine temporel sera découpé en intervalles $I_n=[nT_{mod}, (n+1)T_{mod}]$, pour n entier positif, et T_{mod} réel fixé. La référence en tension que l'on désire atteindre est notée V_{ref}

1.2.2.2 Commande par hystérésis

La première méthode consiste à commander le transistor $T1$ en tout ou rien en respectant une logique de commutation par hystérésis [Leonhard, 1985], [Semail, 1990].

Le principe est le suivant : soit e_{ref} une valeur positive fixée. L'erreur en sortie $V_{ref}-V$ est notée ε .

Considérons le cas où ε est négatif. Si ε est inférieur à $-e_{ref}$, alors $T1$ est bloqué sur I_n ; si ε est supérieur à $-e_{ref}$, $T1$ est bloqué sur I_n si ε diminue et passant dans le cas contraire.

Le même type de raisonnement est appliqué dans le cas où l'erreur est positive. On obtient un comportement global en hystérésis que l'on peut schématiser (fig.1.12) :

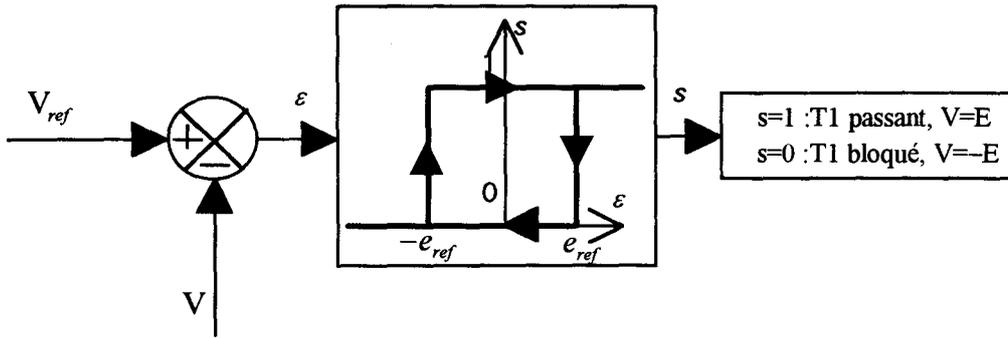


Fig.1.12 : Commande d'un convertisseur par hystérésis.

s représente la commande externe du convertisseur.

La commande ainsi définie entraîne que l'erreur ε converge en valeur moyenne vers un cycle limite d'amplitude inférieure ou égale à e_{ref} . Cette grandeur est donc choisie la plus petite possible en fonction des contraintes matérielles imposées par le système.

I.2.2.3 Commande par M.L.I.

La commande par modulation de largeur d'impulsion peut être décrite par le principe suivant : le calcul de l'erreur en sortie $V_{ref} - V$ permet de déterminer un temps de conduction à l'intérieur de I_n , que l'on notera α_n , appelé rapport cyclique.

Sur I_n , V peut alors être représenté (fig.1.13) :

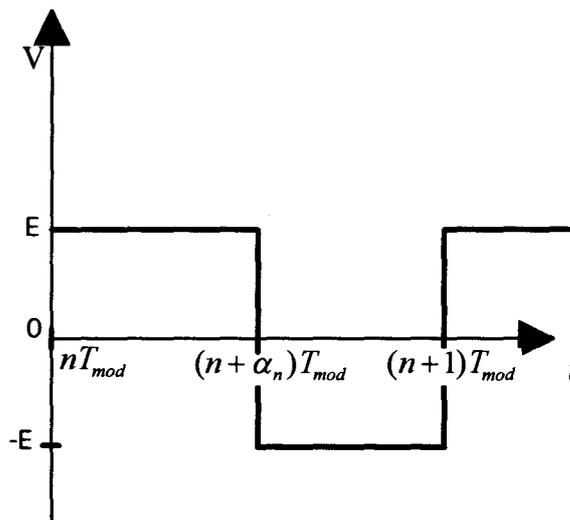


Fig.1.13 : Valeur de V sur I_n .

V est alors égal, en valeur moyenne sur I_n à : $(2\alpha_n - 1)E$

La procédure de calcul de α_n est décrite par le schéma suivant (fig. 1.14) :

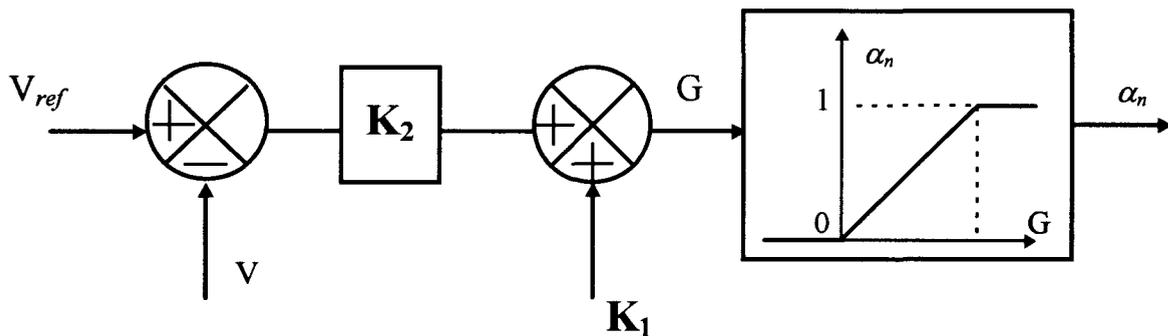


Fig. 1.14 : Schéma de commande d'une branche de convertisseur en M.L.I.

La grandeur $G = K_1 + K_2 (V_{ref} - V)$ permet de calculer le rapport cyclique. Si G est dans $[0, 1]$, on choisit le rapport cyclique égal à G . Si G est négatif on choisit $\alpha_n = 0$ et si G est plus grand que 1, on prend $\alpha_n = 1$.

Ce type d'algorithme génère une erreur statique qu'il est possible de compenser dans une boucle d'asservissement supérieure, qui permet de régler K_1 et K_2 de façon à l'éliminer.

I.3 COMMANDE PAR MODES GLISSANTS

Les techniques de commande par modes glissants peuvent être utilisées dans bon nombre de domaines pour améliorer les performances des systèmes car elles permettent, entre autres, de faire du placement de pôle, de l'optimisation, de l'adaptation de paramètres, et de manière plus générale toutes les techniques usuelles de l'automatique. Les commandes de chaque côté du plan de commutation sont continues et dépendent de la variable d'état x , ce qui donne des propriétés intéressantes au système commandé.

Cette technique a été appliquée à de nombreuses reprises en électronique de puissance pour la commande de moteur en utilisant la MLI pour piloter le convertisseur statique associé [Sabanovic et al, 1981, 1984, 1988], [Bose, 1985], [Wrong, 1987]. Nous verrons au paragraphe suivant qu'il est possible d'appliquer cette méthode directement sur le système à entrées booléennes sans utiliser de technique rapprochée pour la commande du convertisseur.

Au cours de ce paragraphe nous allons présenter brièvement la méthode générale de commande par modes glissants, dans un premier temps pour les systèmes linéaires, puis dans le cas non-linéaire affine.

I.3.1 Systèmes à structure variable

Les systèmes à structure variable présentent un intérêt considérable pour l'étude de la commande. Les premières utilisations de l'idée de changement de structure apparaissent dans des publications datant des années soixante [Emelyanov et al, 1959, 1963(a), 1963(b)], [Garrett, 1961], [Clegg, 1956], [Neimark, 1957], [Ostrovsky, 1960]. Le fil conducteur d'une telle démarche était de combiner les propriétés intéressantes de chacune des structures utilisées. Le système discontinu ainsi commandé peut posséder des propriétés qui n'existent pas dans les sous-systèmes qui le composent.

La notion de modes glissants introduite par de nombreux auteurs soviétiques fut développée au cours des années soixante-dix [Emelyanov et al, 1964, 1966, 1967], [Utkin, 1971, 1974, 1977]. L'idée générale est de déterminer une surface de glissement, généralement des hyperplans fixés dans l'espace d'état, utilisée comme surface de commutation. Le système dans le mode glissant converge alors asymptotiquement vers l'origine.

I.3.2 Elaboration de la loi de commande

I.3.2.1 Cas linéaire

Considérons le système linéaire multivariable décrit par :

$$\dot{x} = \mathbf{A}x + \mathbf{B}u \quad (1.6)$$

où $u=(u_1, \dots, u_m)^T$ est le vecteur de commande, $x=(x_1, \dots, x_n)^T$, le vecteur d'état, et \mathbf{A} , \mathbf{B} des matrices de dimensions appropriées.

Soit $s(x)=(s_1(x), \dots, s_m(x))=0$ un ensemble de m surfaces de commutation définies par :

$$s(x) = \mathbf{C}x \quad (1.7)$$

Supposons que u_i est discontinue sur $s_i=0$:

$$\begin{cases} u_i = u_i^+(x) \text{ pour } s_i(x) > 0 \\ u_i = u_i^-(x) \text{ pour } s_i(x) < 0 \end{cases} \quad (1.8)$$

Le vecteur \dot{x} est discontinu selon le même plan. Ceci entraîne que si les trajectoires sont dirigées vers $s=0$, un mode glissant va apparaître sur ce plan. Une condition suffisante pour qu'un mode glissant existe est, [Barbashin, 1967/1970] :

$$\begin{cases} \lim_{s_i \rightarrow 0^-} \dot{s}_i \geq 0 \\ \lim_{s_i \rightarrow 0^+} \dot{s}_i \leq 0 \end{cases} \quad (1.9)$$

ou de manière équivalente :

$$s_i \dot{s}_i \leq 0$$

au voisinage de $s_i(x)=0$.

Si le système est en glissement sur l'intersection des surfaces $s(x)=Cx=0$ alors, on a $C\dot{x} = 0$, ce qui entraîne :

$$0 = CAx + CBu$$

Si nous faisons les hypothèses $\text{rang}(\mathbf{B})=m$ et $\text{rang}(\mathbf{C})=m$, et \mathbf{CB} inversible, on définit une commande équivalente u_e telle que :

$$u_e = -(\mathbf{CB})^{-1} \mathbf{CA}x \quad (1.10)$$

Le mouvement est alors décrit par :

$$\dot{x} = \underbrace{[\mathbf{A} - \mathbf{B}(\mathbf{CB})^{-1} \mathbf{CA}]}_{A_e} x \quad (1.11)$$

Le comportement du système ne dépend pas des valeurs des commandes u_i^+ et u_i^- , qui ne servent qu'à assurer les conditions de stabilité autour des surfaces $s_i=0$.

La matrice A_e a m valeurs propres nulles et $n-m$ non nulles. Le mode de glissement vers l'origine existe alors si et seulement si ces $n-m$ valeurs propres ont une partie réelle négative [Utkin, 1977].

1.3.2.1.1 Cas particulier : systèmes linéaires monovariables

Supposons que u est discontinue sur $s(x)=Cx=0$:

$$\begin{cases} u = u^+(x) \text{ pour } s(x) > 0 \\ u = u^-(x) \text{ pour } s(x) < 0 \end{cases} \quad (1.12)$$

Une condition suffisante pour qu'un mode glissant existe est :

$$\begin{cases} \lim_{s \rightarrow 0^-} \dot{s} \geq 0 \\ \lim_{s \rightarrow 0^+} \dot{s} \leq 0 \end{cases} \quad (1.13)$$

Si le système est en glissement sur la surface $s(x)=Cx=0$ alors, on définit une commande équivalente u_e (si CB non nul) telle que :

$$u_e = -(CB)^{-1} CAx \quad (1.14)$$

et le mouvement est décrit par :

$$\dot{x} = \underbrace{[A - B(CB)^{-1} CA]}_{A_e} x \quad (1.15)$$

Dans le cas monovariable, une condition nécessaire et suffisante d'existence d'un mode de glissement, énoncée par [Emelyanov, 1963] est que la commande équivalente ainsi définie soit comprise entre les valeurs minimales et maximales de u soit, si $u^+ > u^-$:

$$u_e \text{ existe et } u^- < u_e < u^+ \quad (1.16)$$

Une méthode simple de conception d'une loi de commande vérifiant cette condition peut alors être envisagée :

On choisit u telle que $u = u_e + (CB)^{-1} u_d$ avec u_d discontinue, ce qui donne $C\dot{x} = CAx + CBu_e + u_d = u_d$.

La condition (1.16) entraîne donc que l'on doit choisir $u_d > 0$ pour $s < 0$ et $u_d < 0$ pour $s > 0$. On peut alors prendre :

$$u_d = -k \operatorname{sgn}(s) \quad (1.17)$$

avec $k > 0$ et $\operatorname{sgn}(s)$ représente la fonction signe de s : $\begin{cases} \operatorname{sgn}(s) = +1 \text{ si } s > 0 \\ \operatorname{sgn}(s) = -1 \text{ si } s < 0 \end{cases}$

et u_e est bien comprise entre les valeurs minimales et maximales de u .

1.3.2.1.2 Exemple d'application

Considérons l'exemple suivant (monovariable en entrée) [Fossard, 1991] :

$$\begin{pmatrix} \dot{x}_1 \\ \dot{x}_2 \end{pmatrix} = \begin{pmatrix} 0 & 1 \\ -a_0 & -a_1 \end{pmatrix} x + \begin{pmatrix} 0 \\ 1 \end{pmatrix} u$$

Soit $s(x)=0$ un plan quelconque défini par : $s(x) = (c \ 1)x = cx_1 + x_2$

La commande équivalente u_e est définie par :

$$u_e = (a_0 \ a_1 - c)x = a_0 x_1 + (a_1 - c)x_2$$

On suppose que u commute selon ce plan tel que :

$$u = u_e - k \operatorname{sgn}(s)$$

Le système commandé peut s'exprimer :

$$\begin{cases} \dot{x}_1 = x_2 \\ \dot{x}_2 = -cx_2 - k \operatorname{sgn}(s) \end{cases}$$

et le système en mode glissant se comporte :

$$\begin{cases} \dot{x}_1 = x_2 \\ \dot{x}_2 = -cx_2 \end{cases}$$

ce qui donne sur le plan de glissement :

$$\begin{cases} x_1(t) = x_{10} + \frac{x_{20}}{c} (1 - e^{-c(t-t_0)}) \\ x_2(t) = x_{20} e^{-c(t-t_0)} \end{cases}$$

Ceci peut être représenté pour $c=2$, $x_{10}>0$, $x_{20}<0$ par (fig. 1.15) :

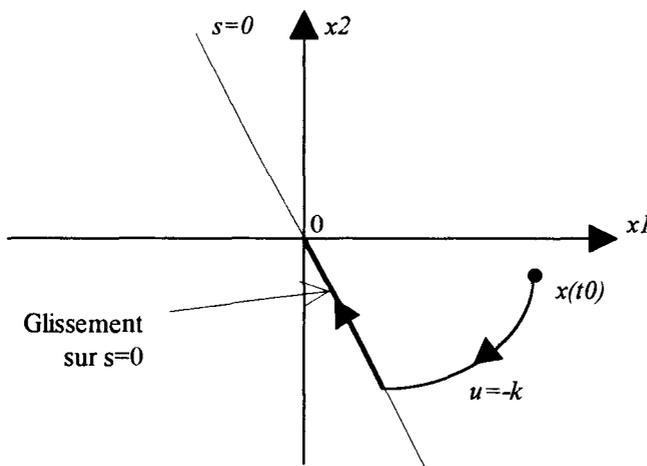


Fig. 1.15 : Mode glissant pour un système de dimension 2.

I.3.2.2 Cas des systèmes non-linéaires affines en la commande

Dans ce paragraphe, nous allons décrire succinctement la méthode pour appréhender le cas non-linéaire multientrées, en traitant le cas des systèmes affines en la commande.

Considérons le système décrit par :

$$\dot{x}(t) = f(x) + B(x)u \quad (1.18)$$

où $x \in \mathbf{R}^n$ est le vecteur état du système, $u \in \mathbf{R}^m$ est le vecteur de commande du système, $f(x)$ est une fonction de l'état à valeurs dans \mathbf{R}^n et $B(x)$ une fonction de l'état à valeurs dans $\mathbf{R}^n \times \mathbf{R}^m$.

La commande discontinue est supposée telle que :

$$\begin{cases} u_i = u_i^+(x) & \text{si } s_i(x) > 0 \\ u_i = u_i^-(x) & \text{si } s_i(x) < 0 \end{cases} \quad (1.19)$$

où u_i est la i -ème composante de u , et $u_i^+ \neq u_i^-$ pour tout $i = \{1, \dots, m\}$. $s_i(x)$ est la i -ème composante du vecteur s .

On notera $G(x)$ la fonction définie par :

$$G(x) = \frac{\partial s}{\partial x} \quad (1.20)$$

Une condition suffisante pour que le mode glissant apparaisse sur la i -ème composante est [Utkin, 1971] :

$$\begin{cases} \lim_{s_i \rightarrow 0^+} \dot{s}_i \leq 0 \\ \lim_{s_i \rightarrow 0^-} \dot{s}_i \geq 0 \end{cases} \quad (1.21)$$

ou

$$s_i \dot{s}_i \leq 0$$

au voisinage de $s_i(x)=0$.

Dans le mode glissant, on a $s(x)=0$ et $\dot{s}(x) = 0$, ce qui entraîne que la commande équivalente u_e est définie, si $G(x)B(x)$ inversible par :

$$u_e = -(G(x)B(x))^{-1}G(x)f(x) \quad (1.22)$$

Le comportement en mode glissant est alors décrit par [Utkin, 1971] :

$$\dot{x} = f(x) - B(x)(G(x)B(x))^{-1}G(x)f(x) \quad (1.23)$$

Si un régime de glissement apparaît sur la famille de surfaces de dimension $n-m$, intersection des surfaces de discontinuités $s_i(x)=0$, l'équation $s(x)=0$ nous permet d'éliminer m variables d'état et la description du système est réduite à un système de dimension $(n-m)$.

Dans le cas non-linéaire monovarié une condition nécessaire et suffisante d'existence de régime de glissement est que l'on ait, comme dans le cas linéaire, u_e comprise entre les valeurs minimales et maximales de l'entrée, soit si $u^+ > u^-$:

$$u^- \leq u_e \leq u^+ \quad (1.24)$$

Remarque :

Dans le cas multivariable le mode de glissement existe s'il y a un régime de commutation sur toutes les intersections de toutes les surfaces. Plusieurs méthodes pour élaborer les lois de commande sont données dans [Fossard, 1991]. Le lecteur pourra se référer pour plus de détails sur les conditions d'existence de surface de glissement, ainsi que de stabilité, à cet ouvrage ainsi qu'à ceux cités dans cette partie.

I.4 MODES GLISSANTS POUR LES SYSTEMES A ENTREES BOOLEENNES

Il est possible d'appliquer directement la technique des modes glissants aux systèmes continus à entrées booléennes en prenant $u_i^+ = 1$ et $u_i^- = 0$ de part et d'autre d'hyperplans de commutation $s_i(x)=0$. Cette méthode consiste donc à appliquer l'algorithme de commande sur le système global intégrant les interrupteurs (fig.1.16) :

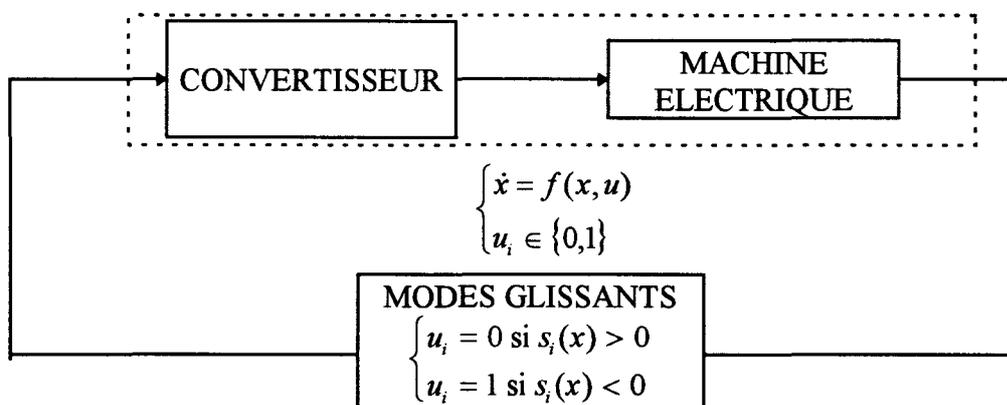


Fig.1.16 : Commande par modes glissant du système global.

Quelques auteurs ont envisagé d'utiliser ce type de démarche. D'une part, H. Sira-Ramirez dans [Sira-Ramirez, 1988, 1989] propose pour les systèmes à commutation monovariables une technique qui consiste à élaborer une loi de commande par modes glissants équivalente à haute fréquence à la MLI. Cette approche ne traite pas le problème multivariable et l'auteur l'applique sur un circuit électrique qui ne comporte qu'un élément de commutation.

D'autre part, A. Sabanovic dans [Sabanovic, 1993] propose de commander une association onduleur-moteur synchrone en utilisant un modèle multivariable simplifié de l'ensemble. Cependant, aucune méthode générale n'est donnée, en particulier sur les conditions que doivent satisfaire les surfaces de glissement dans le cas de systèmes à commandes booléennes. Nous allons proposer dans ce qui suit une méthode générale pour l'élaboration de lois de commande par modes glissants des systèmes à entrées booléennes.

I.4.1 Cas linéaire

Considérons le système linéaire multivariable :

$$\dot{x} = \mathbf{A}x + \mathbf{B}u \quad (1.25)$$

avec x dans \mathbf{R}^n , u_i dans $\{0,1\}$ pour $i=1,\dots,m$ et \mathbf{A} et \mathbf{B} des matrices constantes de dimensions appropriées, avec \mathbf{A} inversible.

Construisons $U=2u-1$ à valeur dans $\{-1,1\}^m$.

Le changement de variable $X = x + \Lambda$ nous permet de nous ramener à :

$$\dot{X} = \mathbf{A}X + \mathbf{B}^*U \quad (1.26)$$

avec U dans $\{-1,1\}^m$, $\mathbf{B}^* = \frac{1}{2}\mathbf{B}$ et $\Lambda = \frac{\mathbf{A}^{-1}\mathbf{B}}{2}$.

Soit $s_i=0$, $i=1,\dots,m$ les m plans définis par $s(X)=CX=0$ où \mathbf{C} est une matrice constante de $\mathbf{R}^m \times \mathbf{R}^n$.

Nous choisissons une logique de commutation définie, par analogie avec le cas monoentrée, par :

$$\begin{cases} U_i = -\text{sgn}(s_i) \\ i = 1, \dots, m \end{cases} \quad (1.27)$$

La commande équivalente est définie, comme précédemment, ce qui donne :

$$U_e = -(\mathbf{CB}^*)^{-1} \mathbf{CAX}, \quad (1.28)$$

en supposant \mathbf{CB}^* non nul et le comportement du système en mode glissant est défini par :

$$\dot{X} = (\mathbf{I} - \mathbf{B}^*(\mathbf{CB}^*)^{-1} \mathbf{C}) \mathbf{AX} \quad (1.29)$$

Cas mono-entrée

U est discontinue sur $s(x)=Cx=0$ avec $U=-\text{sgn}(s)$, avec $U=2u-1$. U_e est définie par (1.28). La condition d'existence du glissement est vérifiée si et seulement si la commande équivalente est comprise entre les bornes minimales et maximales de l'entrée -1 et 1, soit :

$$-1 \leq U_e \leq 1 \quad (1.30)$$

Cette condition indique que le glissement ne pourra pas avoir lieu pour toutes les valeurs de l'état, mais que l'on est restreint à un domaine.

1.4.2 Exemple d'application

1.4.2.1 Exemple 1

Considérons le système linéaire monoentrée décrit par :

$$\begin{pmatrix} \dot{x}_1 \\ \dot{x}_2 \end{pmatrix} = \begin{pmatrix} 0 & 1 \\ -1 & 0 \end{pmatrix} \begin{pmatrix} x_1 \\ x_2 \end{pmatrix} + \begin{pmatrix} 0 \\ -2K \end{pmatrix} u \quad (1.31)$$

avec u booléen.

On effectue le changement de variable décrit précédemment pour se ramener à une commande dans $\{-1,1\}$:

$$\begin{cases} X_1 = x_1 + K \\ X_2 = x_2 \end{cases}$$

et $U=2u-1$.

On obtient le système :

$$\begin{pmatrix} \dot{X}_1 \\ \dot{X}_2 \end{pmatrix} = \begin{pmatrix} 0 & 1 \\ -1 & 0 \end{pmatrix} \begin{pmatrix} X_1 \\ X_2 \end{pmatrix} + \begin{pmatrix} 0 \\ -K \end{pmatrix} U$$

La logique de commutation est définie par :

$$\begin{cases} U = 1 & \text{si } s(X) = cX_1 + X_2 > 0 \\ U = -1 & \text{si } s(X) = cX_1 + X_2 < 0 \end{cases}$$

La commande équivalente est :

$$U_e = \frac{-1}{2K} (c^2 + 1) X_1$$

et le glissement ne pourra avoir lieu que si

$$-1 \leq \frac{-1}{2K} (c^2 + 1) X_1 \leq 1$$

c'est à dire sur le segment AB représenté fig1.17, avec $c=K=1$:

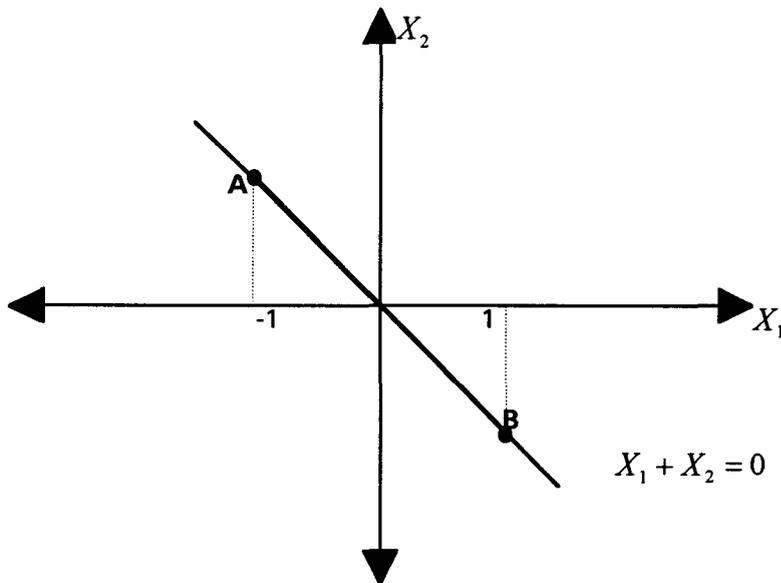


Fig.1.17 : Représentation du segment de glissement, dans le plan (X_1, X_2) .

Ce qui donne, pour le système initial, dans le plan (x_1, x_2) :

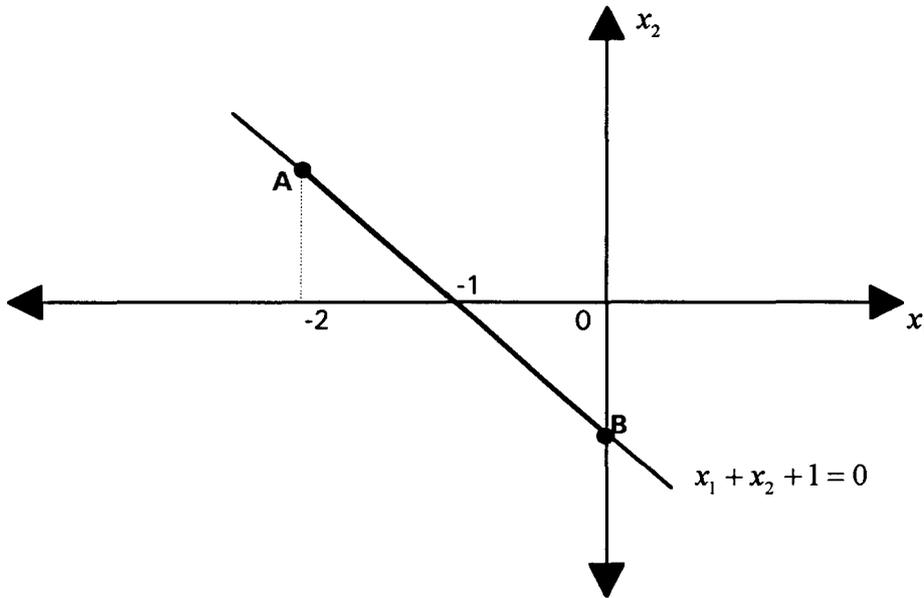


Fig. 1.18 : Représentation du segment de glissement, dans le plan (x_1, x_2) .

I.4.2.2 Exemple 2

Pour l'exemple suivant, nous considérons que le système a déjà été mis sous la forme (1.26). Considérons le système de dimension 3 :

$$\dot{X} = \begin{pmatrix} 0 & 1 & -1 \\ 0 & 0 & 1 \\ -1 & -2 & -1 \end{pmatrix} X + \begin{pmatrix} 1 & -1 \\ 0 & 0 \\ 0 & 1 \end{pmatrix} U \quad (1.32)$$

avec $U_i \in \{-1, 1\}$, pour $i = 1, 2$.

Soit $s_1=0$ le plan tel que $s_1(X) = c_1 X_1 + c_2 X_2 + X_3 = 0$ et $s_2=0$ le plan tel que $s_2(X) = c_3 X_1 + c_4 X_2 + X_3 = 0$.

U_i est supposée telle que : $U_i = -\text{sgn}(s_i)$.

La commande équivalente est définie alors sur l'intersection des deux plans par:

$$U_e = -(\mathbf{CB}^*)^{-1} \mathbf{CA}X \quad (1.33)$$

Soit :

$$U_e = \begin{pmatrix} U_{e_1} \\ U_{e_2} \end{pmatrix} = \begin{pmatrix} 1 & 1 & 2 + \frac{c_2(1-c_3) + c_4(c_1-1)}{c_3-c_1} \\ 1 & 2 & 1 + \frac{-c_2 c_3 + c_4 c_1}{c_3-c_1} \end{pmatrix} x \quad (1.34)$$

avec $c_1 \neq c_3$.

Le comportement du système en mode glissant est déterminé, en appliquant l'équation (1.29), par :

$$\begin{cases} \dot{X}_1 \\ \dot{X}_2 \\ \dot{X}_3 \end{cases} = \begin{pmatrix} 0 & 0 & \alpha \\ 0 & 0 & 1 \\ 0 & 0 & \beta \end{pmatrix} \begin{pmatrix} X_1 \\ X_2 \\ X_3 \end{pmatrix} \quad (1.35)$$

$$\text{avec } \alpha = \frac{c_2 - c_4}{c_3 - c_1} \text{ et } \beta = \frac{-c_2 c_3 + c_4 c_1}{c_3 - c_1}.$$

D'autre part, on a, sur l'intersection des surfaces de glissement :

$$\begin{cases} \dot{s}_1(X) = (-c_2^2 + c_2 + c_1 + c_1 c_2 - 2)X_2 + (c_1^2 - c_1 c_2 + c_1 - 1)X_1 + c_1(U_1 - U_2) + U_2 \\ \dot{s}_2(X) = (-c_4^2 + c_4 + c_3 + c_3 c_4 - 2)X_2 + (c_3^2 - c_3 c_4 + c_3 - 1)X_1 + c_3(U_1 - U_2) + U_2 \end{cases}$$

Les conditions de glissement s'écrivent alors :

$$\begin{cases} \begin{cases} -1 \\ 1 - 2c_1 \end{cases} \leq (-c_2^2 + c_2 + c_1 + c_1 c_2 - 2)X_2 + (c_1^2 - c_1 c_2 + c_1 - 1)X_1 \leq \begin{cases} 1 \\ 2c_1 - 1 \end{cases} \\ \begin{cases} -1 \\ 2c_3 - 1 \end{cases} \leq (-c_4^2 + c_4 + c_3 + c_3 c_4 - 2)X_2 + (c_3^2 - c_3 c_4 + c_3 - 1)X_1 \leq \begin{cases} 1 \\ 1 - 2c_3 \end{cases} \end{cases} \quad (1.36)$$

qui impose nécessairement $c_3 \leq \frac{1}{2}$ et $c_1 \geq \frac{1}{2}$.

Les inégalités (1.36) délimitent le domaine sur l'intersection de $s_1=0$ et $s_2=0$ sur lequel le glissement est possible. Il faut de plus, pour pouvoir assurer une convergence vers l'origine que la valeur propre de (1.35) non nulle soit négative :

$$\beta = \frac{-c_2 c_3 + c_4 c_1}{c_3 - c_1} < 0 \quad (1.37)$$

L4.3 Cas bilinéaire

L'application de commande à modes glissants pour les systèmes bilinéaires à entrées booléennes dont une approche est donnée dans le cas monovorable dans [Sira-Ramirez, 1988]

présente un intérêt pratique car ces modèles représentent un certain nombre de systèmes électriques à commutation.

I.4.3.1 Cas monoentrée

Considérons un système bilinéaire monoentrée à entrée booléenne décrit par :

$$\dot{x} = Ax + Nxu \quad (1.38)$$

avec $x \in \mathbf{R}^n$; $A, N \in \mathbf{R}^n \times \mathbf{R}^n$; $u \in \{0,1\}$.

Dans le cas linéaire, le plan de glissement du système à entrées dans $\{-1,1\}$ était choisi de la forme $s(X)=CX$, ce qui entraîne que le plan de glissement du système à entrée booléenne est de la forme $s(x)=Cx+d$ avec d scalaire.

Par analogie avec le cas linéaire, choisissons $s(x)=Cx+d=0$ comme hyperplan de commutation ; la logique de commutation peut alors être définie par :

$$u = \frac{1 - \text{sgn}(s)}{2} \quad (1.39)$$

Les conditions d'existence d'un plan de glissement sont :

$$\lim_{s \rightarrow 0^+} \dot{s} > 0 \text{ et } \lim_{s \rightarrow 0^-} \dot{s} < 0$$

On a, sur le plan de glissement, $Cx+d=0$, ce qui entraîne $C\dot{x}=0$, soit :

$$CAx + CNxu = 0 \quad (1.40)$$

ce qui définit une commande équivalente :

$$u_e = -(CNx)^{-1} (CAx) \quad (1.41)$$

définie si CNx est non nul.

Il faut alors vérifier, d'après (1.24) que :

$$0 \leq -(CNx)^{-1} (CAx) \leq 1 \quad (1.42)$$

Nous allons montrer sur l'exemple suivant le principe de la démarche :

Soit le système, exprimé par :

$$\dot{x} = \begin{pmatrix} \dot{x}_1 \\ \dot{x}_2 \end{pmatrix} = \begin{pmatrix} 0 & 1 \\ -a_1 & -a_2 \end{pmatrix} x + \begin{pmatrix} b_1 \\ 0 \end{pmatrix} x_1 u + \begin{pmatrix} 0 \\ b_2 \end{pmatrix} x_2 u \quad (1.43)$$

avec $u \in \{0,1\}$.

Le plan de glissement est défini par : $s(x)=Cx+d=cx_1+x_2+d=0$.

On utilise la logique de commutation définie en (1.39). La commande équivalente est calculée, en utilisant les résultats précédents :

$$u_e = \frac{1}{cb_1x_1 + b_2x_2} (a_1x_1 + (-c + a_2)x_2) \quad (1.44)$$

Pour cet exemple, trois possibilités peuvent se présenter :

- Il existe un mode glissant en tout point de $s=0$.
- Il n'existe pas de mode glissant sur $s=0$.
- Il existe un mode glissant sur une partie de $s=0$.

On a :

$$u_e = \frac{-c + a_2}{b_2} \frac{\left(\frac{a_1}{-c + a_2} x_1 + x_2\right)}{\frac{cb_1}{b_2} x_1 + x_2} \quad (1.45)$$

Le choix de c tel que $a_1b_2 = cb_1(a_2 - c)$ entraîne que la condition (1.24) est vérifiée sur $s=0$ si et seulement si :

$$0 \leq \frac{a_2 - c}{b_2} \leq 1 \quad (1.46)$$

Si cette condition est vérifiée un mode glissant existe en tout point de $s=0$. Dans le cas contraire il n'existe pas de mode glissant sur $s=0$.

D'autre part, si on choisit par exemple $b_1=b_2=a_1=a_2=1$, et $c=2$ la condition (1.24) entraîne :

$$0 \leq \frac{(x_1 - x_2)}{2x_1 + x_2} \leq 1 \quad (1.47)$$

soit, sur $s=0$, avec $d=3$:

$$-1 \leq x_1 \leq 0 \quad (1.48)$$

Remarque :

Cette exemple montre que l'existence d'un plan de glissement vérifiant les conditions (1.42)) n'est pas toujours possible, quelle que soit la logique de commutation choisie.

I.4.3.2 Cas multi-entrées

Considérons un système bilinéaire multientrées à commandes booléennes décrit par :

$$\dot{x} = \mathbf{A}x + \left(\sum_{p=1}^m \mathbf{N}_p u_p \right) x \quad (1.49)$$

avec $x \in \mathbf{R}^n$; $\mathbf{A}, \mathbf{N}_p \in \mathbf{R}^n \times \mathbf{R}^n$; $u_p \in \{0,1\}, p=1,\dots,m$.

Soit $s_i=0, i=1,\dots,m$ les m plans définis par $s(x)=\mathbf{C}x+\mathbf{D}=0$ où \mathbf{C} est une matrice constante de $\mathbf{R}^m \times \mathbf{R}^n$, et \mathbf{D} un vecteur constant. Les coefficients de \mathbf{C} et \mathbf{D} sont notés respectivement c_{kj} , pour $k \in \{1,\dots,m\}, j \in \{1,\dots,n\}$, et $d_i, i \in \{1,\dots,m\}$

La logique de commutation est définie, par analogie avec le cas linéaire telle que :

$$\begin{cases} u_i = \frac{1 - \text{sgn}(s_i)}{2} \\ i = 1, \dots, m \end{cases} \quad (1.50)$$

La commande équivalente u_e , peut être exprimée par :

$$u_e = - \left(\mathbf{C} \sum_{p=1}^m \tilde{\mathbf{N}}_p x_p \right)^{-1} (\mathbf{C} \mathbf{A} x) \quad (1.51)$$

avec $\tilde{n}_{ij}(p)$ les coefficients de $\tilde{\mathbf{N}}_p$ définis par : $\begin{cases} \tilde{n}_{ij}(p) = n_{ip}(j) \\ i \in \{1,\dots,n\}; (j,p) \in \{1,\dots,m\}^2 \end{cases}$

Les conditions de glissements sont obtenues en dérivant s_k , pour $k=1,\dots,m$:

$$\dot{s}_k(x) = \sum_{i=1}^n c_{ki} \dot{x}_i = \sum_{i=1}^n c_{ki} \left(\sum_{j=1}^n a_{ij} x_j + \sum_{p=1}^m \left(\sum_{j=1}^n \tilde{n}_{ij}(p) x_j \right) u_p \right) \quad (1.52)$$

avec $c_{kn}=1$.

soit :

$$\dot{s}_k(x) = \sum_{j=1}^m \left(\sum_{i=1}^n c_{ki} a_{ij} + \sum_{p=1}^m \tilde{n}_{ij}(p) u_p \right) x_j \quad (1.53)$$

et en réinjectant x_n dans $\dot{s}_k(x)$ en utilisant $s_k(x)=0$:

$$\dot{s}_k(x) = \sum_{j=1}^{n-1} \left(\sum_{i=1}^n c_{ki} a_{ij} + \sum_{p=1}^m \tilde{n}_{ij}(p) u_p \right) x_j - \left(\sum_{i=1}^n c_{ki} a_{ij} + \sum_{p=1}^m \tilde{n}_{ij}(p) u_p \right) \left(\sum_{i=1}^{n-1} c_{ki} x_i + d_k \right) \quad (1.54)$$

Ce que l'on écrira par la suite :

$$\dot{s}_k(x) = \sum_{j=1}^{n-1} \lambda_{jk}(u, C_k) x_j + \mu(u, C_k, d_k) \quad (1.55)$$

avec $C_k = (c_{k1}, \dots, c_{kn})$, où $\lambda_{jk}(u, C_k)$ sont des grandeurs qui dépendent de l'entrée et de s_k .

Si $s_k(x) > 0$ alors $u_k = 0$ et quels que soient $u_i \in \{0, 1\}$, $i=1, \dots, m$, $i \neq k$, il faut $\dot{s}_k(x) \leq 0$. On a dans cette configuration $2^m - 1$ possibilités pour le vecteur u (u_k étant fixé à 0). Si on note $\lambda_{jk}(r)$ et $\mu_k(r)$, $r=1, \dots, 2^{m-1}$, les valeurs correspondantes de $\lambda_{jk}(u, C_k)$ et $\mu(u, C_k, d_k)$, les valeurs la condition de glissement entraîne :

$$\begin{cases} \sum_{j=1}^{n-1} \lambda_{jk}(r) x_j + \mu_k(r) \leq 0 & \text{pour } k = 1, \dots, m \\ \text{pour } r = 1, \dots, 2^{m-1} \end{cases} \quad (1.56)$$

De la même manière si $s_k(x) < 0$ alors $u_k = 1$ et quels que soient $u_i \in \{0, 1\}$, $i=1, \dots, m$, $i \neq k$, il faut $\dot{s}_k(x) \geq 0$. On a $2^m - 1$ possibilités pour le vecteur u (u_k étant fixé à 1). Si on note $\lambda'_{jk}(r)$ et $\mu'_k(r)$, $r=1, \dots, 2^{m-1}$, les valeurs correspondantes de $\lambda_{jk}(u, C_k)$ et $\mu(u, C_k, d_k)$, la condition de glissement entraîne :

$$\begin{cases} \sum_{j=1}^{n-1} \lambda'_{kj}(r) x_j + \mu'_k(r) \geq 0 & \text{pour } k = 1, \dots, m \\ \text{pour } r = 1, \dots, 2^{m-1} \end{cases} \quad (1.57)$$

On doit alors imposer les coefficients c_{kj} et d_k pour $(k, j) \in \{1, \dots, m\} \times \{1, \dots, n\}$ tels que pour chacune des surfaces $s_k=0$, les équations (1.56) et (1.57) ne sont pas incompatibles.

Remarque :

Les résultats concernant l'existence du plan de glissement, que nous avons obtenus dans cette partie sont assez complexes à manipuler pour les systèmes de dimension élevée, car on doit vérifier la compatibilité de 2^m équations. Pour ce qui est de la condition de stabilité, le système (1.49) avec $u=u_e$ est difficile à analyser car il est non-linéaire dans le cas général.

CONCLUSION

Nous avons vu dans ce chapitre que les systèmes physiques comportant des commutateurs modélisés par bond-graph peuvent être représentés par une équation d'état à structure booléenne en entrée. L'étude de la commande des systèmes en électronique de puissance a mis en évidence l'existence de deux classes d'algorithmes utilisés : d'une part les méthodes de commande par valeurs moyennes des entrées (généralement la MLI) qui ne donnent pas directement accès à l'état des commutateurs et d'autre part certaines applications de la commande par modes glissants (cas monovarié, exemples particuliers). L'étude de cette dernière a montré pour les systèmes étudiés que dans le cas monovarié, les méthodes d'élaboration de lois de commande sont réalisables et simples à mettre en oeuvre et que dans le cas multientrées, cette élaboration peut se révéler complexe quant aux méthodologies à utiliser. Nous allons donc proposer au cours des deux chapitres suivants des algorithmes de commande applicables directement aux systèmes multivariés en entrées.

CHAPITRE II

COMMANDE BOOLEENNE DES SYSTEMES LINEAIRES

CHAPITRE II

INTRODUCTION

Le but de ce chapitre est de proposer de nouveaux algorithmes de commande pour des systèmes linéaires à commutation. Pour cela, nous allons utiliser un modèle global du système issu d'une représentation bond-graph. Nous avons vu au premier chapitre que cette représentation permettait d'obtenir un modèle d'état dont les entrées booléennes représentent l'état des éléments de commutation du système. Nous allons donc proposer une étude générale des équations d'état linéaires à commandes booléennes pour faire de la régulation, puis du suivi de trajectoire.

Nous allons élaborer dans un premier temps une loi de commande en boucle ouverte qui va permettre de déterminer des domaines de convergence pour les vecteurs état et sortie du système. Nous proposerons ensuite un algorithme de commande en boucle fermée permettant d'améliorer les performances du système.

II.1 POSITION DU PROBLEME

Le vecteur d'état du système est noté $x = (x_1, \dots, x_n)^T \in \mathbf{R}^n$, le vecteur entrée $u = (u_1, \dots, u_m)^T \in \mathbf{R}^m$, et $y = (y_1, \dots, y_s)^T \in \mathbf{R}^s$ représente le vecteur sortie. \mathbf{A} , \mathbf{B} et \mathbf{C} sont des matrices constantes telles que : $\mathbf{A} \in \mathbf{R}^n \times \mathbf{R}^n$, $\mathbf{B} \in \mathbf{R}^n \times \mathbf{R}^m$ et $\mathbf{C} \in \mathbf{R}^s \times \mathbf{R}^n$. Les composantes de u ont des valeurs booléennes.

Le modèle du système peut être exprimé par :

$$\begin{cases} \dot{x} = \mathbf{A}x + \mathbf{B}u \\ y = \mathbf{C}x \\ u_i \in \{0,1\}, i = 1, \dots, m. \end{cases} \quad (2.1)$$

Le but de l'étude est de déterminer la séquence de vecteurs u permettant à l'état x ou à la sortie y d'atteindre une valeur désirée constante x_d (resp. y_d), ou de poursuivre une trajectoire $x_d(t)$ (resp. $y_d(t)$).

Le vecteur d'entrée ne peut prendre qu'un nombre fini de valeurs, égal à 2^m , car chacune de ses composantes est booléenne. L'ensemble de ces vecteurs est noté $config(u)$, et on a :

$$config(u) = \left\{ \begin{pmatrix} 0 \\ 0 \\ 0 \\ \vdots \\ 0 \\ 0 \end{pmatrix}, \begin{pmatrix} 1 \\ 0 \\ 0 \\ \vdots \\ 0 \\ 0 \end{pmatrix}, \begin{pmatrix} 0 \\ 1 \\ 0 \\ \vdots \\ 0 \\ 0 \end{pmatrix}, \begin{pmatrix} 0 \\ 0 \\ 1 \\ \vdots \\ 0 \\ 0 \end{pmatrix}, \dots, \begin{pmatrix} 0 \\ 0 \\ 0 \\ \vdots \\ 0 \\ 1 \end{pmatrix}, \begin{pmatrix} 1 \\ 1 \\ 0 \\ \vdots \\ 0 \\ 0 \end{pmatrix}, \begin{pmatrix} 1 \\ 0 \\ 1 \\ \vdots \\ 0 \\ 0 \end{pmatrix}, \begin{pmatrix} 1 \\ 0 \\ 0 \\ \vdots \\ 0 \\ 1 \end{pmatrix}, \begin{pmatrix} 1 \\ 1 \\ 1 \\ \vdots \\ 0 \\ 1 \end{pmatrix} \right\} \quad (2.2)$$

Les vecteurs de cet ensemble seront notés $[config(u)]_p$ avec p appartenant à $\{1, \dots, 2^m\}$.

Chaque élément de $config(u)$ peut être défini à partir des vecteurs de la base canonique de \mathbf{R}^m . Les polyèdres formés par les éléments de l'ensemble $config(u)$ sont donc inclus dans \mathbf{R}^m .

II.2 DOMAINE DE CONVERGENCE

II.2.1 Définitions

Dans ce paragraphe, nous allons donner les définitions de base concernant la commandabilité des systèmes continus. Pour plus de détails, le lecteur pourra se référer à [Nijmeijer et al, 1990].

1) Atteignabilité

Une composante du vecteur état est dit u -atteignable si et seulement si la commande u a une influence sur son comportement.

2) Commandabilité d'un système

Un système est dit commandable en l'état si et seulement si quels que soient $x(t_1)$ et $x(t_2)$ appartenant à l'espace d'état, il existe une commande $u(t)$ permettant au vecteur état de passer de $x(t_1)$ à $x(t_2)$ en un temps fini.

Par extension, un domaine de l'espace d'état est dit commandable si et seulement si quels que soient $x(t_1)$ et $x(t_2)$ appartenant à ce domaine, il existe une commande $u(t)$ permettant au vecteur état de passer de $x(t_1)$ à $x(t_2)$ en un temps fini.

3) Accessibilité

Une valeur x_1 de l'état est dite u -accessible à partir de la valeur x_0 si et seulement si il existe une commande $u(t)$ et des réels positifs t_0 et t_1 tels que la solution de l'équation d'état vérifie :

$$\begin{cases} x(t_0) = x_0 \\ x(t_1) = x_1 \end{cases} \quad (2.3)$$

4) Domaine de convergence

Nous appellerons domaine de convergence en l'état (resp. en sortie) d'un système, un domaine fermé dont tous les points peuvent être atteints par le vecteur état (resp. sortie) du système en un temps infini.

Remarques :

Remarque 1 :

Les notions de commandabilité et d'accessibilité diffèrent essentiellement sur un point : l'accessibilité ne fait pas d'hypothèse quant au sens de variation du temps. La commandabilité représente donc une condition plus "dure" que l'accessibilité, ce qui peut être illustré par l'exemple suivant :

Considérons le système non-linéaire décrit par l'équations d'état suivante :

$$\begin{cases} \dot{x}_1 = x_2^2 \\ \dot{x}_2 = u \end{cases} \quad (2.4)$$

Les solutions de ce système appartiennent obligatoirement à la partie située "à droite" de $x_1(t_0)$ dans l'espace d'état (fig.2.1) car \dot{x}_1 est toujours positive :

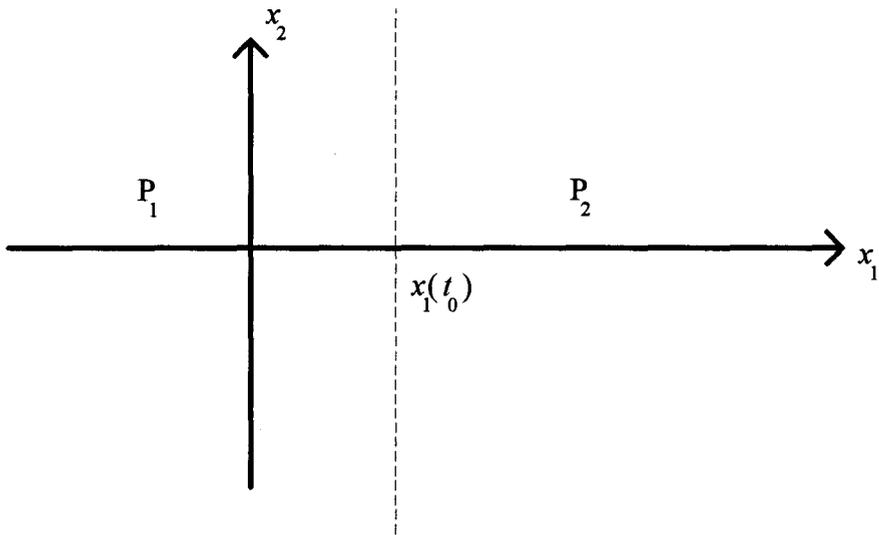


Fig.2.1 : Domaines d'accessibilité et de commandabilité.

Tout point de \mathbf{R}^2 est donc accessible alors que seul le domaine P_2 est commandable.

Remarque 2 :

La notion d'atteignabilité n'est en fait qu'une pré-condition à la commandabilité d'un système. La gouvernabilité n'a pas été évoquée car elle fait apparaître la notion d'observabilité.

L'accessibilité et la commandabilité ont été introduites ici dans le but de clarifier l'utilisation du terme de domaine de convergence qui sera faite au cours de cette étude, et nous montrerons au §II.3.2 sur un exemple de système à entrées booléennes la différence entre la notion de domaine de convergence et commandabilité.

II.2.2 Résolution de l'équation d'état

II.2.2.1 Cas général

La théorie des équations différentielles linéaires nous permet d'exprimer $x(t)$ en fonction du temps :

$$x(t) = e^{A(t-t_0)} x(t_0) + \int_{t_0}^t e^{A(t-\theta)} \mathbf{B}u(\theta) d\theta \quad (2.5)$$

en prenant t_0 comme origine des temps.

Les composantes u_i sont booléennes, et le vecteur u a 2^m valeurs distinctes. Les valeurs de t pour lesquelles les entrées commutent sont notées $\{t_1, t_2, \dots\}$. Sur $[t_k, t_{k+1}]$, u est un vecteur constant dont la valeur est notée :

$$u = [\text{config}(u)]_{f(k)}$$

avec $f(k) \in \{1, \dots, 2^m\}$, pour tout k entier, ce qui conduit à, pour $\tau \in [t_k, t_{k+1}]$:

$$x(\tau) = e^{A(\tau-t_k)} x(t_k) + \left(\int_{t_k}^{\tau} e^{A(\tau-\theta)} d\theta \right) \mathbf{B} [\text{config}(u)]_{f(k)} \quad (2.6)$$

II.2.2.2 Hypothèse de stabilité :

Nous allons supposer dans tout ce qui suit que le modèle exprimé par l'équation (2.1) est asymptotiquement stable, ce qui entraîne que la matrice \mathbf{A} est non singulière. L'expression (2.6) s'exprime alors, pour $\tau \in [t_k, t_{k+1}]$, par :

$$x(\tau) = e^{A(\tau-t_k)} x(t_k) - \mathbf{A}^{-1} (\mathbf{I} - e^{A(\tau-t_k)}) \mathbf{B} [\text{config}(u)]_{f(k)} \quad (2.7)$$

ce qui s'écrit,

$$x(\tau) = e^{A(\tau-t_k)} x(t_k) - \mathbf{A}^{-1} (\mathbf{I} - e^{A(\tau-t_k)}) \mathbf{B}_k \quad (2.8)$$

en posant $\mathbf{B}_k = \mathbf{B} [\text{config}(u)]_{f(k)}$. \mathbf{I} est la matrice identité de dimension n .

II.2.3 Domaine de convergence de l'état

On pose $x_{i\infty} = -\mathbf{A}^{-1} \mathbf{B}_i$, pour $i \in \{1, \dots, 2^m\}$. Si on impose la valeur $[\text{config}(u)]_i$, $i \in \{1, \dots, 2^m\}$ pour tout $t \in [0, +\infty[$, en prenant $t_k = 0$, $t_{k+1} = +\infty$ et $\tau = t$ dans (2.8), il vient :

$$x(t) = e^{At} x(0) - \mathbf{A}^{-1} [\mathbf{I} - e^{At}] \mathbf{B}_i \quad (2.9)$$

Le modèle est supposé asymptotiquement stable, ce qui implique $\lim_{t \rightarrow \infty} (e^{At}) = 0$, soit :

$$\lim_{t \rightarrow \infty} (x(t)) = -A^{-1}B_i = x_{i\infty}.$$

Proposition 1 :

Considérons ρ_x le polyèdre convexe formé par l'ensemble de points $\{x_{1\infty}, \dots, x_{2^m\infty}\}$. ρ_x représente le domaine de convergence du vecteur d'état du système défini par l'équation (2.1).

Preuve :

Nous allons démontrer dans le paragraphe §II.3 de ce chapitre que l'on peut construire un algorithme permettant d'atteindre tout point quelconque de ρ_x . Cela démontrera alors que ce polyèdre est un domaine de convergence pour l'état du système.

II.3 COMMANDE EN BOUCLE OUVERTE

Dans cette partie, nous allons élaborer une loi de commande en boucle ouverte permettant de commander l'état d'un système linéaire à entrées booléennes à l'intérieur du domaine de convergence défini précédemment.

II.3.1 Commande vers un objectif fixe

Soit $x_d \in \rho_x$, la consigne désirée pour le vecteur état et t_0 représente l'instant initial.

Algorithme de commande 1 :

Soient $\{a_1, \dots, a_{2^m}\}$ les coordonnées barycentriques de x_d dans le polyèdre ρ_x formé des points $\{x_{1\infty}, \dots, x_{2^m\infty}\}$, telles que :

$$\begin{cases} x_d = \sum_{i=1}^{2^m} a_i x_{i\infty} \\ \sum_{i=1}^{2^m} a_i = 1 \\ a_i \in [0,1], i \in \{1, \dots, 2^m\} \end{cases} \quad (2.10)$$

La commande proposée pour commander l'état du système vers l'objectif fixe x_d est une commande périodique décrite par (fig.2.2) :

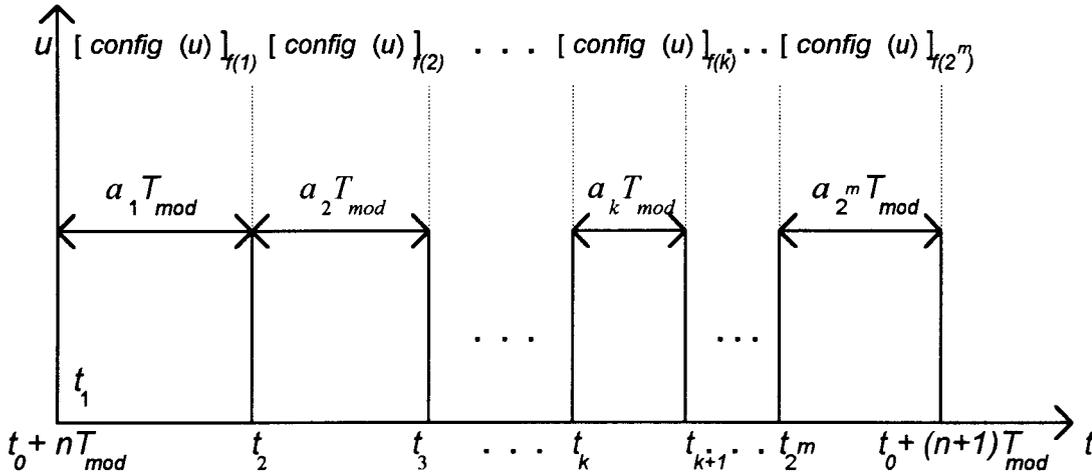


Fig.2.2 : Représentation des commandes durant une période $[t_0+nT_{mod}, t_0+(n+1)T_{mod}]$.

pour $n=0,1,\dots$ et T_{mod} intervalle de temps à définir pour que :

$$\|\mathbf{A}\| \ll \frac{1}{T_{mod}} \quad (2.11)$$

où $\|\mathbf{A}\|$ représente une norme de la matrice \mathbf{A} .

Chaque $[config(u)]_{r(k)}$ appartient à l'ensemble $config(u)$ défini par l'équation (2.2). Le choix n'est pas unique et sera fait en fonction de critères à définir.

Avec cette stratégie de commande, le système modélisé par (2.1) converge vers x_d .

Démonstration :

Les x_{i_∞} pour $i \in \{1, \dots, 2^m\}$ sont déterminés en utilisant la proposition 1, et l'équation (2.10) nous permet de déterminer les valeurs de α_i en résolvant le système de $n+1$ équations et 2^m inconnues.

Chaque intervalle $[t_0+nT_{mod}, t_0+(n+1)T_{mod}]$, $n \in \mathbb{N}$ est subdivisé en intervalles $[t_k, t_{k+1}]$, $k \in \mathbb{N}$ tels $t_{k+1} - t_k = \alpha_k T_{mod}$ avec $\alpha_i \in [0,1]$ pour tout $i \in \{1, \dots, 2^m\}$. Nous allons montrer que le choix de $\alpha_i = \alpha_i$, pour tout $i \in \{1, \dots, 2^m\}$ entraîne une convergence au premier ordre du vecteur état vers x_d .

On applique sur chaque intervalle $[t_0+nT_{mod}, t_0+(n+1)T_{mod}]$ la même séquence pour u : la valeur $[config(u)]_{f(k)}$ est appliquée au système durant chaque intervalle de temps : $t_{k+1}-t_k=\alpha_k T_{mod}$, (fig.2.3) :

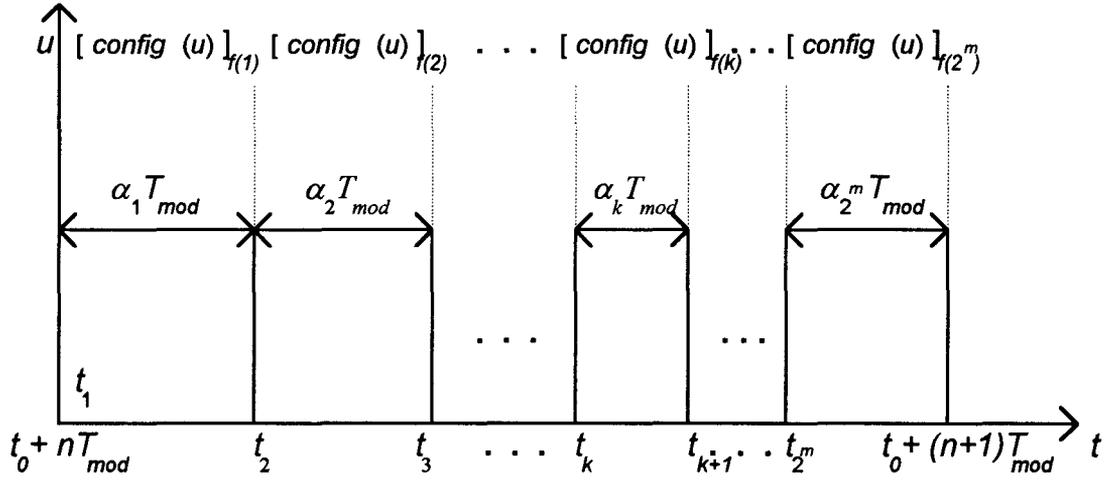


Fig.2.3 : Représentation des entrées durant une période $[t_0+nT_{mod}, t_0+(n+1)T_{mod}]$.

Dans l'intervalle $[t_0, t_0+T_{mod}]$, l'équation (2.8) s'écrit :

$$x(t_0 + \alpha_1 T_{mod}) = e^{A\alpha_1 T_{mod}} x(t_0) - \mathbf{A}^{-1} [\mathbf{I} - e^{A\alpha_1 T_{mod}}] \mathbf{B}_1$$

$$x(t_0 + (\alpha_1 + \alpha_2) T_{mod}) = e^{A(\alpha_2 + \alpha_1) T_{mod}} x(t_0) - e^{A\alpha_2 T_{mod}} \mathbf{A}^{-1} [\mathbf{I} - e^{A\alpha_1 T_{mod}}] \mathbf{B}_1 - \mathbf{A}^{-1} [\mathbf{I} - e^{A\alpha_2 T_{mod}}] \mathbf{B}_2$$

$$x(t_0 + T_{mod}) = e^{AT_{mod}} x(t_0) - \sum_{\substack{j=1 \\ i=j+1 \\ i \leq 2^m}}^{2^m} \prod e^{A\alpha_i T_{mod}} \mathbf{A}^{-1} [\mathbf{I} - e^{A\alpha_j T_{mod}}] \mathbf{B}_j$$

De même,

$$x(t_0 + 2T_{mod}) = e^{2AT_{mod}} x(t_0 + T_{mod}) - \sum_{\substack{j=1 \\ i=j+1 \\ i \leq 2^m}}^{2^m} \prod e^{A\alpha_i T_{mod}} \mathbf{A}^{-1} [\mathbf{I} - e^{A\alpha_j T_{mod}}] \mathbf{B}_j$$

Par récurrence dans l'intervalle $[nT_{mod}, (n+1)T_{mod}]$, cela donne :

$$\underbrace{x(t_0 + (n+1)T_{mod})}_{x_{n+1}} = e^{AT_{mod}} \underbrace{x(t_0 + nT_{mod})}_{x_n} - \underbrace{\sum_{\substack{j=1 \\ i=j+1 \\ i \leq 2^m}}^{2^m} \prod e^{A\alpha_i T_{mod}} \mathbf{A}^{-1} [\mathbf{I} - e^{A\alpha_j T_{mod}}] \mathbf{B}_j}_K \quad (2.12)$$

soit :

$$x_{n+1} = e^{AT_{mod}} x_n + K, \text{ pour } n=0,1,\dots$$

Après quelques calculs développés en Annexe 1, cela conduit à :

$$x_n = \left(\mathbf{I} - (e^{AT_{mod}})^n \right) \left(\left[\mathbf{I} - e^{AT_{mod}} \right]^{-1} K \right) + (e^{AT_{mod}})^n x(t_0) \quad (2.13)$$

A est supposée asymptotiquement stable, d'où $\lim_{t \rightarrow \infty} x(t) = \lim_{n \rightarrow \infty} x_n = \left[\mathbf{I} - e^{AT_{mod}} \right]^{-1} K$. Si la limite est notée x_∞ , en utilisant l'équation (2.12), on a :

$$x_\infty = - \left(\mathbf{I} - e^{AT_{mod}} \right)^{-1} \sum_{j=1}^{2^m} \prod_{\substack{i=j+1 \\ i \leq 2^m}} e^{A\alpha_i T_{mod}} \mathbf{A}^{-1} \left[\mathbf{I} - e^{A\alpha_j T_{mod}} \right] \mathbf{B}_j \quad (2.14)$$

Le but est de calculer $\alpha_i, i \in \{1, \dots, 2^m\}$ qui vérifie $x_\infty = x_d = - \sum_{i=1}^{2^m} \alpha_i \mathbf{A}^{-1} \mathbf{B}_i$, ce qui donne, par identification (Annexe 2) :

$$\begin{cases} e^{\alpha_{2^m} AT_{mod}} \dots e^{\alpha_{i+1} AT_{mod}} \left(\mathbf{I} - e^{\alpha_i AT_{mod}} \right) = \left(\mathbf{I} - e^{AT_{mod}} \right) \alpha_i \\ i \in \{1, \dots, 2^m\} \end{cases} \quad (2.15)$$

Par une utilisation du développement en série entière de l'exponentielle il vient :

$$\sum_{k=1}^{\infty} \frac{(\alpha_i AT_{mod})^k}{k!} = \sum_{k=1}^{\infty} \alpha_i \frac{(AT_{mod})^k}{k!} \sum_{k=0}^{\infty} \frac{(-(\alpha_{i+1} + \dots + \alpha_{2^m}) AT_{mod})^k}{k!} \quad (2.16)$$

ce qui peut être écrit :

$$\alpha_i \mathbf{I} = \alpha_i \mathbf{I} + \mathbf{A} T_{mod} \phi(\mathbf{A} T_{mod}) \quad (2.17)$$

avec :

$$\phi(\mathbf{A} T_{mod}) = \frac{\alpha_i - \alpha_i^2}{2} \mathbf{I} - \alpha \alpha_i \mathbf{I} - \alpha \frac{\mathbf{A} T_{mod}}{2} - \sum_{k=2}^{\infty} \frac{(\alpha_i + \alpha_i)(\mathbf{I} - \alpha \mathbf{A} T_{mod})(\mathbf{A} T_{mod})^{k-1}}{(k+1)!} - \alpha \alpha_i \sum_{k=0}^{\infty} \frac{(\mathbf{A} T_{mod})^{k-1}}{(k+1)!} \sum_{k=2}^{\infty} \frac{(\alpha \mathbf{A} T_{mod})^{k-1}}{k!}$$

et

$$\alpha = \alpha_{i+1} + \dots + \alpha_{2^m} \quad (2.18)$$

Si on suppose $\|\mathbf{A} T_{mod}\| \ll 1$, on peut assimiler ϕ à son terme de premier ordre, et on obtient (Annexe 3) :

$$\|AT_{mod}\phi(AT_{mod})\| \ll \alpha_i$$

Nous pouvons conclure, si on suppose $\|AT_{mod}\| \ll 1$, que nous pouvons faire une approximation de l'égalité (2.17) au premier ordre, ce qui donne:

$$\begin{cases} \alpha_i = a_i \\ i \in \{1, \dots, 2^m\} \end{cases} \quad (2.19)$$

■

Remarques :

L'algorithme de commande que nous venons de définir nous permet d'atteindre tout point x_d appartenant à \mathcal{P}_x . \mathcal{P}_x représente donc un domaine de convergence en l'état du système.

D'autre part, il est bien connu que $|\lambda(\mathbf{A})|_{\max} \leq \|\mathbf{A}\|$. L'hypothèse $\|\mathbf{A}\| \ll 1/T_{mod}$ est équivalente à $|\lambda(\mathbf{A})|_{\max} \ll 1/T_{mod}$. Ceci représente une hypothèse usuelle pour les systèmes dont la période d'échantillonnage doit être choisie petite par rapport aux dynamiques du système.

L'algorithme peut être représenté (fig.2.4) :

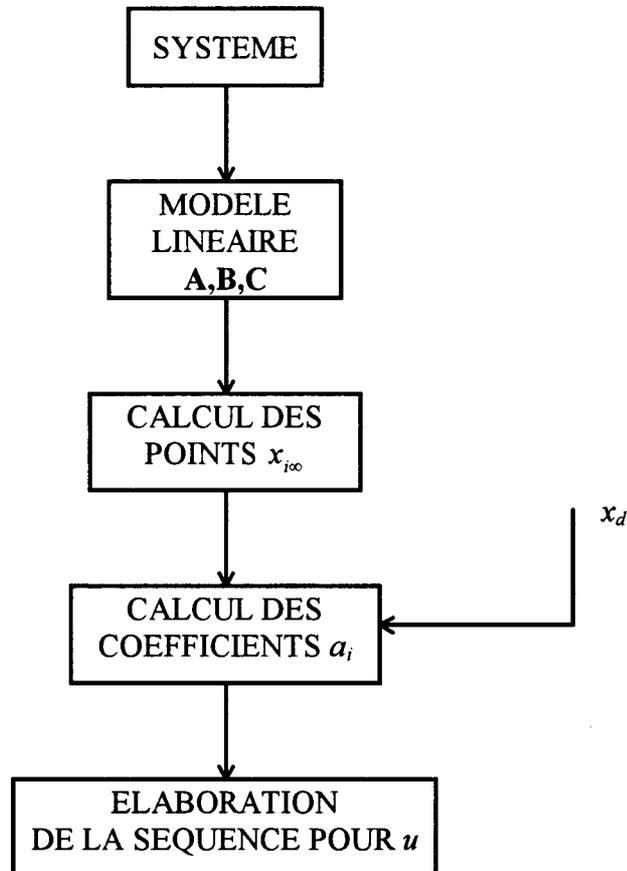


Fig.2.4 : Algorithme de commande en boucle ouverte.

II.3.2 Domaine de commandabilité

Dans ce qui précède, nous avons mis en évidence l'existence d'un domaine que l'on a nommé domaine de convergence, représentant l'ensemble des points accessibles avec l'algorithme proposé par le vecteur état du système. Il est très important de le distinguer du domaine de commandabilité qui représente l'ensemble des points accessibles en un temps fini par l'état.

L'exemple suivant nous permet de mettre en évidence la différence entre les deux notions :

$$\begin{pmatrix} \dot{x}_1 \\ \dot{x}_2 \end{pmatrix} = \begin{pmatrix} -2 & 1 \\ 0 & -4 \end{pmatrix} \begin{pmatrix} x_1 \\ x_2 \end{pmatrix} + \begin{pmatrix} 0 & 0 \\ 2 & 2 \end{pmatrix} \begin{pmatrix} u_1 \\ u_2 \end{pmatrix} \quad (2.20)$$

On peut calculer les sommets du domaine de convergence en utilisant les résultats précédents:

$$x_{1\infty} = \begin{pmatrix} 0 \\ 0 \end{pmatrix} \quad x_{2\infty} = \begin{pmatrix} \frac{1}{4} \\ 1 \end{pmatrix} \quad x_{3\infty} = \begin{pmatrix} \frac{1}{4} \\ \frac{1}{2} \end{pmatrix} \quad x_{4\infty} = \begin{pmatrix} \frac{1}{2} \\ 1 \end{pmatrix} \quad (2.21)$$

Donc le domaine de convergence du vecteur état ρ_x est le segment $\overline{Ox_{4\infty}}$ et peut être représenté (fig.2.5) :

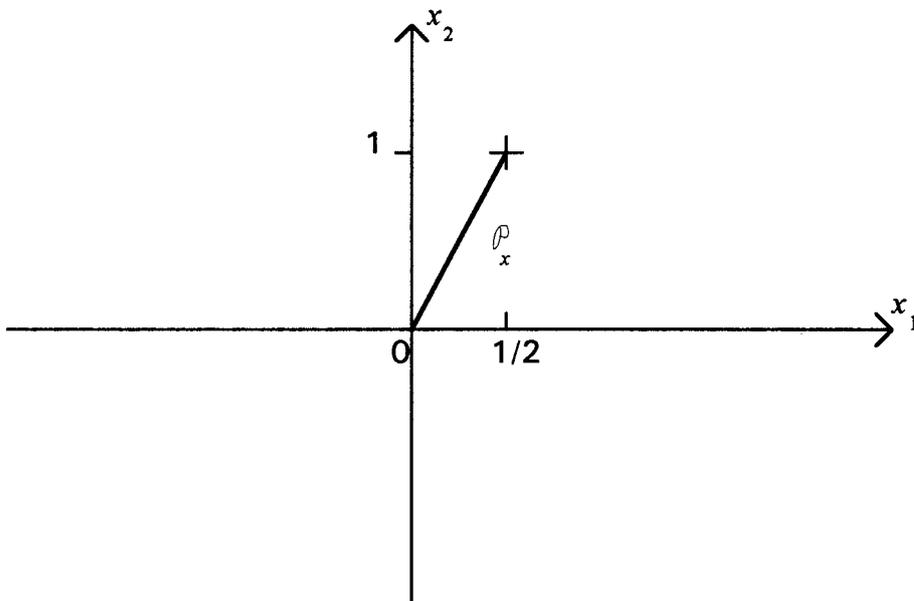


Fig.2.5 : Domaine de convergence du système.

Si on impose sur tout le domaine temporel, $u=[\text{config}(u)]_4$, i.e. $u=(1,1)^T$ alors on a montré que le système convergeait vers $x_{4\infty}$. Pour représenter dans ce cas le domaine de commandabilité, il faut résoudre le système d'équations différentielles suivant :

$$\begin{cases} \dot{x}_1 = -2x_1 + x_2 \\ \dot{x}_2 = -4x_2 + 4 \end{cases} \quad (2.22)$$

On obtient alors, à partir de l'origine :

$$\begin{cases} x_1 = \frac{1}{2} - e^{-2t} + \frac{e^{-4t}}{2} \\ x_2 = 1 - e^{-4t} \end{cases} \quad (2.23)$$

ce qui donne dans l'espace d'état la courbe \mathcal{C} d'équation :

$$x_1 = \frac{1-x_2}{2} - \sqrt{1-x_2} + \frac{1}{2} \quad (2.24)$$

qui peut être représentée (fig.2.6) :

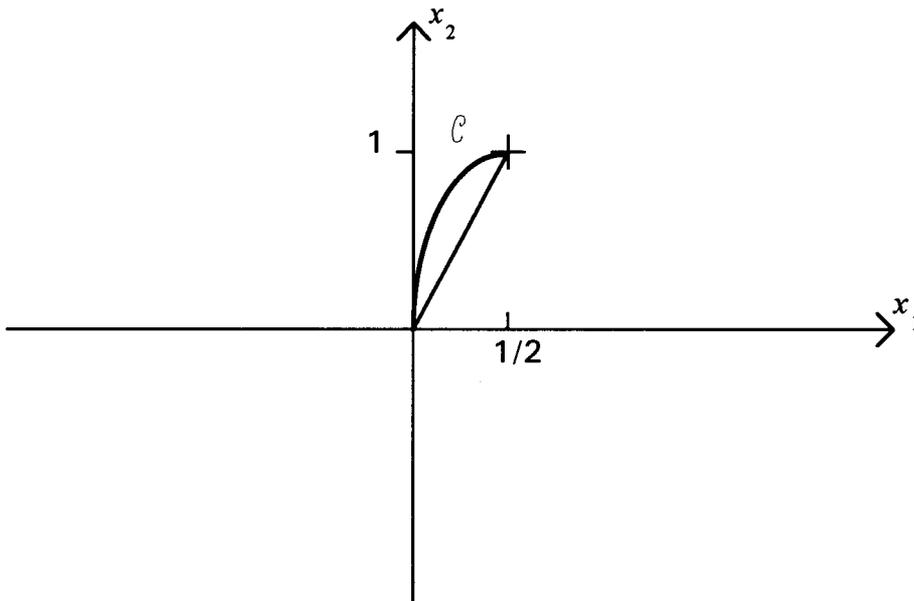


Fig.2.6 : Trajectoire dans l'espace d'état.

Les points de la courbe \mathcal{C} sont commandables depuis $x_0=0$ puisque chacun d'entre eux peut être pris par l'état en un temps fini, alors que le domaine de convergence est le segment $\overline{Ox_{4\infty}}$.

II.3.3 Commande en sortie

L'algorithme de commande présenté précédemment consiste en une commande de l'état du système. Nous allons voir qu'il est applicable à la commande en sortie des systèmes linéaires à commutations.

II.3.3.1 Domaine de convergence

On a vu au §II.2 que le domaine de convergence pour le vecteur état du système était limité par l'ensemble $\{x_{1\infty}, \dots, x_{2^m\infty}\}$ avec $x_{i\infty} = -\mathbf{A}^{-1}\mathbf{B}_i$, pour $i=1, \dots, 2^m$.

De la même manière, on peut définir, pour $i=1, \dots, 2^m$:

$$y_{i\infty} = \mathbf{C}x_{i\infty} \quad (2.25)$$

Proposition 2 :

Le polyèdre convexe formé par l'ensemble des $y_{i\infty}$ représente le domaine de convergence du vecteur sortie du système décrit par (2.1). Ce polyèdre est noté ρ_y .

Preuve :

Considérons y_d appartenant à ρ_y . Il existe au moins un x_d tel que $y_d = \mathbf{C}x_d$ (voir Annexe 4).

Comme défini précédemment, x_d peut être écrit sous la forme :

$$\left\{ \begin{array}{l} x_d = \sum_{i=1}^{2^m} a_i x_{i\infty} \\ \sum_{i=1}^{2^m} a_i = 1 \\ a_i \in [0,1], \text{ pour } i \in \{1, \dots, 2^m\}. \end{array} \right. \quad (2.26)$$

En multipliant par \mathbf{C} , il vient :

$$\begin{cases} y_d = \sum_{i=1}^{2^m} \alpha_i y_{i\infty} \\ \sum_{i=1}^{2^m} \alpha_i = 1 \\ \alpha_i \in [0,1], i = 1, \dots, 2^m \end{cases} \quad (2.27)$$

Ainsi une application de l'algorithme décrit au §II.3.1 conduit à une convergence au premier ordre vers x_d . Ceci entraîne donc que la sortie converge vers y_d au premier ordre.

II.3.3.2 Algorithme de commande

Dans la pratique, l'élaboration de l'algorithme de commande de la sortie n'est pas la même que celle utilisée pour la démonstration. La démarche à utiliser est de calculer les $y_{i\infty} = Cx_{i\infty}$ et de calculer directement les α_i en utilisant la relation (2.27). La mise en place de l'algorithme est alors identique à celle utilisée pour le vecteur d'état.

II.3.4 Choix de la commande

II.3.4.1 Degrés de liberté

L'espace engendré par les vecteurs $\{x_{1\infty}, \dots, x_{2^m\infty}\}$ est noté H_x et sa dimension m_x . De même, l'espace engendré par les vecteurs $\{y_{1\infty}, \dots, y_{2^m\infty}\}$ est noté H_y et sa dimension m_y .

Nous avons vu que l'ensemble $config(u)$ engendre un espace de dimension m . D'après les définitions des $x_{i\infty}$ et $y_{i\infty}$, cela entraîne :

$$m_x \leq m \text{ et } m_x \leq n \quad (2.28)$$

$$m_y \leq m \text{ et } m_y \leq s \quad (2.29)$$

car $\dim(x_{i\infty})=n$ et $\dim(y_{i\infty})=s$.

Considérons les systèmes linéaires associés aux équations (2.10) et (2.27). Ces systèmes peuvent être décrits comme suit :

$$\begin{cases} \sum_{i=1}^{2^m} x_{i\infty} a_i = x_d \\ \sum_{i=1}^{2^m} a_i = 1 \end{cases} \quad (2.30)$$

et

$$\begin{cases} \sum_{i=1}^{2^m} a_i y_{i\infty} = y_d \\ \sum_{i=1}^{2^m} a_i = 1 \end{cases} \quad (2.31)$$

Le système (2.30) n'est en général pas directement soluble car il peut exister des degrés de liberté dans le choix des coefficients a_i .

Si on choisit x_d appartenant à \mathcal{P}_x , les coordonnées de x_d sont liées entre elles par $n-m_x$ équations. Donc, (2.30) représente un système de $n+1-(n-m_x)=m_x+1$ équations indépendantes. On a donc: 2^m-m_x-1 degrés de liberté pour le choix des a_i .

Le même raisonnement peut être appliqué pour le choix des coefficients de commande en sortie. (2.31) représente alors un système de $n+1-(n-m_y)=m_y+1$ équations indépendantes. Nous avons donc : 2^m-m_y-1 degrés de liberté pour le choix des a_i .

II.3.4.2 Contraintes

Les équations (2.10) et (2.27) ne sont cependant pas équivalentes aux systèmes linéaires (2.30) et (2.31). En effet il faut tenir compte pour les premières de l'existence de la contrainte $a_i \in [0, 1]$ pour $i=1, \dots, 2^m$.

On peut démontrer que les degrés de libertés qui apparaissent dans la résolution de (2.30) et (2.31) entraînent, excepté si x_d (resp. y_d) est un des sommets de \mathcal{P}_x (resp. \mathcal{P}_y), que l'on a une infinité de possibilités dans le choix des coefficients a_i vérifiant la contrainte $a_i \in [0, 1]$ pour $i=1, \dots, 2^m$.

L'exemple suivant permet d'illustrer cela.

Considérons le système suivant :

$$\dot{x} = \begin{pmatrix} -1 & 2 \\ 0 & -4 \end{pmatrix} x + \begin{pmatrix} 1 & 2 \\ -2 & 2 \end{pmatrix} \begin{pmatrix} u_1 \\ u_2 \end{pmatrix} \quad (2.32)$$

on a :

$$\begin{aligned} x_{1\infty} &= \begin{pmatrix} 0 \\ 0 \end{pmatrix} & x_{2\infty} &= -\mathbf{A}^{-1}\mathbf{B} \begin{pmatrix} 1 \\ 0 \end{pmatrix} = \begin{pmatrix} 0 \\ -1/2 \end{pmatrix} \\ x_{3\infty} &= -\mathbf{A}^{-1}\mathbf{B} \begin{pmatrix} 0 \\ 1 \end{pmatrix} = \begin{pmatrix} 3 \\ 1/2 \end{pmatrix} & x_{4\infty} &= -\mathbf{A}^{-1}\mathbf{B} \begin{pmatrix} 1 \\ 1 \end{pmatrix} = \begin{pmatrix} 3 \\ 0 \end{pmatrix} \end{aligned} \quad (2.33)$$

Le domaine de convergence est un quadrilatère que l'on peut représenter dans \mathbf{R}^2 (fig.2.7) :

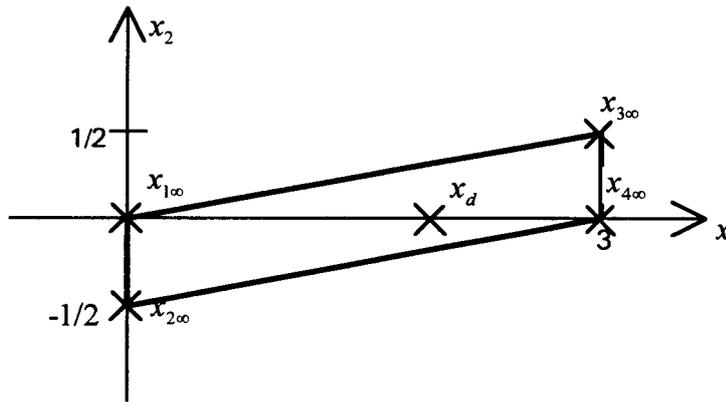


Fig.2.7 : Domaine de convergence.

Soit $x_d = (x_{d1}, x_{d2})^T$ appartenant à \mathcal{P}_x . x_d peut être exprimé en utilisant l'équation (2.10), ce qui donne :

$$\begin{cases} x_{d1} = 3a_3 + 3a_4 \\ x_{d2} = \frac{-1}{2}a_2 + \frac{1}{2}a_3 \\ a_1 = 1 - a_2 - a_3 - a_4 \end{cases} \quad (2.34)$$

Si on exprime le système précédent en utilisant a_3 comme paramètre, on obtient :

$$\begin{cases} a_2 = -2x_{d_2} + a_3 \\ a_4 = \frac{x_{d_1}}{3} - a_3 \\ a_1 = 1 - \frac{x_{d_1}}{3} + 2x_{d_2} - a_3 \end{cases} \quad (2.35)$$

La contrainte $a_i \in [0,1]$ pour $i=1,2,3,4$ implique les inégalités suivantes :

$$\begin{cases} 0 < a_3 < 1 \\ 0 < a_3 - 2x_{d_2} < 1 \\ 0 < -a_3 + \frac{x_{d_1}}{3} < 1 \\ 0 < a_3 - 2x_{d_2} + \frac{x_{d_1}}{3} < 1 \end{cases} \quad (2.36)$$

Si x_d ne représente pas un des sommets du domaine \mathcal{P}_{x^*} , alors il existe une infinité de valeurs possibles pour les coefficients a_i , $i=1,2,3,4$.

Choisissons $x_d = (2,0)^T$, par exemple. L'équation (2.35) entraîne :

$$\begin{cases} a_1 = \frac{1}{3} - a_3 \\ a_2 = a_3 \\ a_4 = \frac{2}{3} - a_3 \end{cases} \quad (2.37)$$

On voit donc que toute valeur de a_3 appartenant à $[0,1/3]$ convient et permet aux coefficients de satisfaire les conditions (2.36).

Si on choisit x_d comme étant un des sommets de \mathcal{P}_{x^*} , $x_d = x_{3^{\infty}} = (3,0.5)^T$ par exemple, l'équation (2.35) entraîne :

$$\begin{cases} a_1 = 1 - a_3 \\ a_2 = a_3 - 1 \\ a_4 = 1 - a_3 \end{cases} \quad (2.38)$$

Les conditions $a_2 \in [0,1]$ et $a_3 \in [0,1]$ entraînent $a_3=1$. On a alors une solution unique :

$$\begin{aligned} a_1 = a_2 = a_4 = 0 \\ a_3 = 1 \end{aligned} \quad (2.39)$$

On peut remarquer que cette solution revient pour l'algorithme de commande à imposer au système une unique configuration, celle qui correspond à $x_{3\infty}$, sur tout le domaine temporel, qui représente le seul moyen d'atteindre ce sommet.

II.3.4.3 Critères de choix

Nous avons donc mis en évidence l'existence dans ce qui précède de degrés de liberté pour le choix de l'algorithme de commande, cette latitude peut être utilisée de différentes manières pour la commande.

a) Exclusion d'interrupteurs

Les degrés de liberté relatifs aux choix de la commande peuvent être utilisés pour pallier une panne éventuelle d'un élément de commutation, si celle-ci est détectée.

Considérons l'exemple suivant :

$$\begin{pmatrix} \dot{x}_1 \\ \dot{x}_2 \end{pmatrix} = \begin{pmatrix} -2 & 2 \\ 0 & -1 \end{pmatrix} \begin{pmatrix} x_1 \\ x_2 \end{pmatrix} + \begin{pmatrix} 2 & 1 & 1 \\ 0 & 2 & 1 \end{pmatrix} \begin{pmatrix} u_1 \\ u_2 \\ u_3 \end{pmatrix} \quad (2.40)$$

Les $x_{i\infty}$ pour $i=1, \dots, 8$ peuvent être calculés :

$$\begin{aligned} x_{1\infty} &= \begin{pmatrix} 0 \\ 0 \end{pmatrix} & x_{2\infty} &= -\mathbf{A}^{-1}\mathbf{B} \begin{pmatrix} 1 \\ 0 \\ 0 \end{pmatrix} = \begin{pmatrix} 1 \\ 0 \end{pmatrix} \\ x_{3\infty} &= -\mathbf{A}^{-1}\mathbf{B} \begin{pmatrix} 0 \\ 1 \\ 0 \end{pmatrix} = \begin{pmatrix} 5 \\ 2 \\ 2 \end{pmatrix} & x_{4\infty} &= -\mathbf{A}^{-1}\mathbf{B} \begin{pmatrix} 0 \\ 0 \\ 1 \end{pmatrix} = \begin{pmatrix} 3 \\ 2 \\ 1 \end{pmatrix} \\ x_{5\infty} &= -\mathbf{A}^{-1}\mathbf{B} \begin{pmatrix} 1 \\ 1 \\ 0 \end{pmatrix} = \begin{pmatrix} 7 \\ 2 \\ 2 \end{pmatrix} & x_{6\infty} &= -\mathbf{A}^{-1}\mathbf{B} \begin{pmatrix} 1 \\ 0 \\ 1 \end{pmatrix} = \begin{pmatrix} 5 \\ 2 \\ 1 \end{pmatrix} \\ x_{7\infty} &= -\mathbf{A}^{-1}\mathbf{B} \begin{pmatrix} 0 \\ 1 \\ 1 \end{pmatrix} = \begin{pmatrix} 4 \\ 3 \\ 3 \end{pmatrix} & x_{8\infty} &= -\mathbf{A}^{-1}\mathbf{B} \begin{pmatrix} 1 \\ 1 \\ 1 \end{pmatrix} = \begin{pmatrix} 5 \\ 3 \\ 3 \end{pmatrix} \end{aligned} \quad (2.41)$$

On peut représenter le domaine de convergence (fig.2.8) :

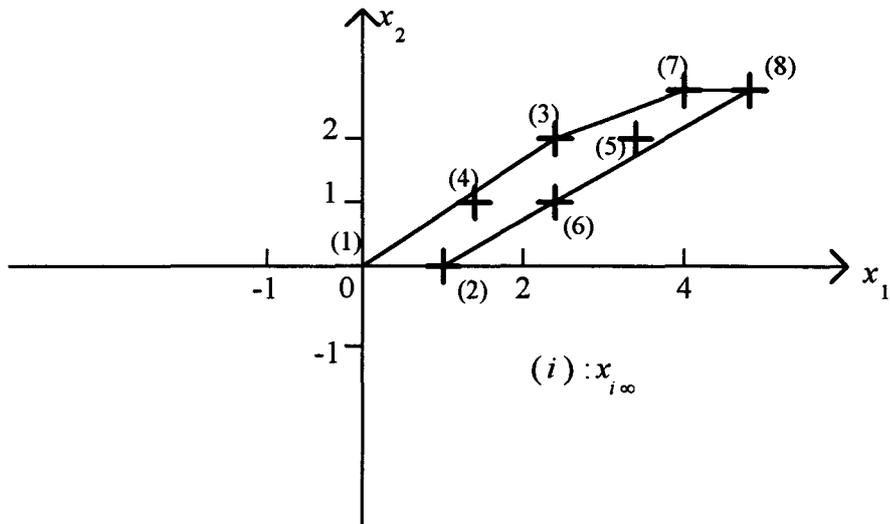


Fig.2.8 : Domaine de convergence.

Si on choisit $x_d = (1/2, 1/8)^T$, le choix des a_i est conditionné par la résolution du système linéaire suivant :

$$\begin{cases} a_2 + \frac{5}{2}a_3 + \frac{3}{2}a_4 + \frac{7}{2}a_5 + \frac{5}{2}a_6 + 4a_7 + 5a_8 = \frac{1}{2} \\ 2a_3 + a_4 + 2a_5 + a_6 + 3a_7 + 3a_8 = \frac{1}{8} \\ a_1 + a_2 + a_3 + a_4 + a_5 + a_6 + a_7 + a_8 = 1 \end{cases} \quad (2.42)$$

Parmi les solutions possibles, détaillons les deux cas suivants :

$$S1: \begin{cases} a_1 = \frac{9}{16} \\ a_2 = \frac{5}{16} \\ a_3 = 0 \\ a_4 = \frac{2}{16} \\ a_5 = a_6 = a_7 = a_8 = 0 \end{cases}$$

$$S2: \begin{cases} a_1 = \frac{19}{32} \\ a_2 = \frac{11}{32} \\ a_3 = \frac{1}{16} \\ a_4 = 0 \\ a_5 = a_6 = a_7 = a_8 = 0 \end{cases}$$

Si on considère que le coefficient a_i est associé à la configuration $[config(u)]_i$ de l'ensemble (2.2), on remarque alors que pour appliquer S1, il n'est pas nécessaire que l'élément de commutation associé à l'entrée 2 fonctionne ($a_3=0$) et pour appliquer S2, le commutateur associé à l'entrée 3 ne fonctionne pas obligatoirement ($a_4=0$).

Ainsi, il est possible d'envisager pallier une panne éventuelle du deuxième ou du troisième commutateur, à condition que celle-ci soit détectée.

De plus, une panne détectée sur l'interrupteur i permet d'éliminer 2^{m-1} configurations de commande de l'ensemble $config(u)$ et de continuer à faire fonctionner le système. Cependant, une diminution du nombre de commutateurs en fonctionnement diminuera l'étendue du domaine de convergence.

b) Optimisation du choix

Le but de ce paragraphe est de poser le problème quant aux critères de choix sur les coefficients a_i pour optimiser le fonctionnement du système.

→ Energie en entrée

L'énergie E_i consommée par l'entrée u_i peut être calculée, si on applique l'algorithme décrit précédemment, sur $[nT_{mod}, (n+1)T_{mod}]$ par :

$$E_i = \int_{nT_{mod}}^{(n+1)T_{mod}} u_i(t) dt \quad (2.43)$$

On définit E l'énergie totale en entrée par :

$$E = \sum_{i=1}^{2^m} E_i \quad (2.44)$$

Si l'ensemble $config(u)$ est construit en respectant l'ordre décrit dans l'équation (2.2), l'entrée u_i est égale à 1 pendant $a_i T_{mod}$ mais aussi pendant $a_k T_{mod}$ pour lequel k est tel que $[config(u)]_k$ ait sa i -ème composante égale à 1.

Dans le cas, par exemple, d'un système de dimension 4, on a $2^4=16$ configurations possibles de l'entrée. $config(u)$ est tel que :

$$config(u) = \left(\begin{pmatrix} 0 \\ 0 \\ 0 \\ 0 \end{pmatrix}, \begin{pmatrix} 1 \\ 0 \\ 0 \\ 0 \end{pmatrix}, \begin{pmatrix} 0 \\ 1 \\ 0 \\ 0 \end{pmatrix}, \begin{pmatrix} 0 \\ 0 \\ 1 \\ 0 \end{pmatrix}, \begin{pmatrix} 0 \\ 0 \\ 0 \\ 1 \end{pmatrix}, \begin{pmatrix} 1 \\ 1 \\ 0 \\ 0 \end{pmatrix}, \begin{pmatrix} 1 \\ 0 \\ 1 \\ 0 \end{pmatrix}, \begin{pmatrix} 1 \\ 0 \\ 0 \\ 1 \end{pmatrix}, \begin{pmatrix} 0 \\ 1 \\ 1 \\ 0 \end{pmatrix}, \begin{pmatrix} 0 \\ 1 \\ 0 \\ 1 \end{pmatrix}, \begin{pmatrix} 0 \\ 0 \\ 1 \\ 1 \end{pmatrix}, \begin{pmatrix} 1 \\ 1 \\ 1 \\ 0 \end{pmatrix}, \begin{pmatrix} 1 \\ 1 \\ 0 \\ 1 \end{pmatrix}, \begin{pmatrix} 1 \\ 0 \\ 1 \\ 1 \end{pmatrix}, \begin{pmatrix} 0 \\ 1 \\ 1 \\ 1 \end{pmatrix} \right)$$

On a alors pour la commande u_1 :

$$E_1 = (a_2 + a_6 + a_7 + a_8 + a_{12} + a_{13} + a_{14} + a_{16}) T_{mod} \quad (2.45)$$

et l'énergie totale est :

$$E = [0.a_1 + 1.(a_2 + a_3 + a_4 + a_5) + 2(a_6 + a_7 + a_8 + a_9 + a_{10} + a_{11}) + 3(a_{12} + a_{13} + a_{14} + a_{15}) + 4a_{16}] T_{mod}$$

ce qui peut s'écrire,

$$E = \left(\sum_{p=0}^4 p \sum_{i=j(p)+1}^{i=j(p)} a_i \right) T_{mod} \quad (2.46)$$

avec : $j(p) = \sum_{i=0}^{p-1} C_4^i$, $j(0) = 0$, où C représente la fonction combinatoire définie pour tout m et i

entiers par :

$$C_m^i = \frac{m!}{(m-i)!i!}$$

Si on généralise au cas de m entrées l'énergie est alors égale à :

$$E = \left(\sum_{p=0}^m p \sum_{i=j(p)+1}^{i=j(p)} a_i \right) T_{mod} \quad (2.47)$$

avec, cette fois : $j(p) = \sum_{i=0}^{p-1} C_m^i$, $j(0) = 0$.

D'autre part, l'équation de détermination des a_i peut s'écrire :

$$\begin{cases} a_1(x_{1\infty})_1 + a_2(x_{2\infty})_1 + \dots + a_{2^m}(x_{2^m\infty})_1 = x_{d_1} \\ a_1(x_{1\infty})_2 + a_2(x_{2\infty})_2 + \dots + a_{2^m}(x_{2^m\infty})_2 = x_{d_2} \\ \vdots \\ a_1(x_{1\infty})_n + a_2(x_{2\infty})_n + \dots + a_{2^m}(x_{2^m\infty})_n = x_{d_n} \end{cases} \quad (2.48)$$

Les $x_{j\infty}$, $j > m+1$ sont liés aux $x_{i\infty}$, $i \leq m$ de la même manière que les $[config(u)]_j$, sont liés aux $[config(u)]_i$, qui forment une base. Ceci entraîne donc que (2.48) peut s'écrire :

$$\begin{cases} \frac{E_1}{T_{mod}}(x_{1\infty})_1 + \frac{E_2}{T_{mod}}(x_{2\infty})_1 + \dots + \frac{E_m}{T_{mod}}(x_{m\infty})_1 = x_{d_1} \\ \frac{E_1}{T_{mod}}(x_{1\infty})_2 + \frac{E_2}{T_{mod}}(x_{2\infty})_2 + \dots + \frac{E_m}{T_{mod}}(x_{m\infty})_2 = x_{d_2} \\ \vdots \\ \frac{E_1}{T_{mod}}(x_{1\infty})_n + \frac{E_2}{T_{mod}}(x_{2\infty})_n + \dots + \frac{E_m}{T_{mod}}(x_{m\infty})_n = x_{d_n} \end{cases} \quad (2.49)$$

Dans le cas où $m \leq n$, les E_i sont donc déterminés (sauf singularités). L'énergie en entrée est alors constante.

Dans le cas contraire, le problème de la minimisation de l'énergie en entrée est donc résolu si on peut calculer les a_i qui minimisent (2.47).

Exemple :

Reprenons l'exemple défini par l'équation (2.32) avec $x_d = (2,0)^T$. Les coefficients a_i vérifient l'équation (2.37) :

$$\begin{cases} a_1 = \frac{1}{3} - a_3 \\ a_2 = a_3 \\ a_4 = \frac{2}{3} - a_3 \end{cases}$$

L'énergie vaut, sur $[nT_{mod}, (n+1)T_{mod}]$, en utilisant (2.47) :

$$E = (a_2 + a_3 + 2a_4)T_{mod} \quad (2.50)$$

Soit $E = cste = 4/3 T_{mod}$. Dans ce cas le choix des a_i n'influe pas sur l'énergie en entrée.

Remarque concernant l'optimisation en sortie :

Nous pouvons poser les équations permettant d'optimiser en sortie un système linéaire à entrées booléennes. Une manière de mesurer les performances d'un système commandé est d'évaluer l'intégrale suivante :

$$I = \int_0^{\infty} (x(t) - x_d)^2 dt \quad (2.51)$$

On ne connaît pas d'expression de $x(t)$ valable pour tout t , mais seulement sur $[t_0 + nT_{mod} + a_k T_{mod}, t_0 + nT_{mod} + a_{k+1} T_{mod}]$. On décompose I de sorte que :

$$I = \sum_{n=0}^{\infty} \sum_{i=1}^{2^m} \int_{t_0 + nT_{mod} + \sum_{j=1}^{i-1} a_j T_{mod}}^{t_0 + nT_{mod} + \sum_{j=1}^i a_j T_{mod}} (x(t) - x_d)^2 dt = \sum_{n=0}^{\infty} \sum_{i=1}^{2^m} I_{n,i} \quad (2.52)$$

avec $a_0 = 0$.

Il faut alors calculer $I_{n,i}$ en remplaçant $x(t)$ par sa valeur, soit (voir II.II.3.2) :

$$\begin{cases} x(t) = e^{A(t-t_{n,i})} x(t_{n,i}) - \mathbf{A}^{-1} (\mathbf{I} - e^{A(t-t_{n,i})}) \mathbf{B}_i \\ \text{avec } t_{n,i} = t_0 + nT_{mod} + \sum_{j=1}^{i-1} a_j T_{mod} \\ \text{et } x(t_{n,i}) = e^{\sum_{j=1}^{i-1} a_j A T_{mod}} x(t_0 + nT_{mod}) - \sum_{j=1}^{i-1} e^{\sum_{k=j+1}^{i-1} a_k A T_{mod}} \mathbf{A}^{-1} (\mathbf{I} - e^{A a_j T_{mod}}) \mathbf{B}_j \\ \text{et } x(t_0 + nT_{mod}) = (\mathbf{I} - e^{nA T_{mod}}) (\mathbf{I} - e^{A T_{mod}}) \mathbf{K} + e^{nA T_{mod}} x(t_0) \end{cases} \quad (2.53)$$

avec \mathbf{K} déterminé par l'équation (2.12).

On voit que la résolution d'une telle équation est assez complexe même si on connaît **A** et **B**. Une méthode de simplification peut être apportée en utilisant un développement en série entière des termes exponentiels.

II.3.5 Application à la poursuite de trajectoires

Le but est ici de poursuivre une trajectoire désirée $y_d(t)$ en sortie. Pour cela, nous ferons l'hypothèse que cette trajectoire appartient au domaine \mathcal{P}_y défini au §II.3.3.1.

Algorithme :

- 1) Discrétisation de la trajectoire $y_d(t)$ à la période T_d telle que $T_d \gg T_{mod}$.
- 2) Application de l'algorithme précédent à chaque $y_d(kT_d)$ et détermination des $a_i(k)$ pour $i=1, \dots, 2^m$ et k entier en utilisant :

$$\left\{ \begin{array}{l} y_d(kT_d) = \sum_{i=1}^{2^m} a_i(k) y_{i\infty} \\ \sum_{i=1}^{2^m} a_i(k) = 1 \\ a_i(k) \in [0,1], i = 1, \dots, 2^m \end{array} \right. \quad (2.54)$$

Remarques :

Remarque 1 :

La solution de (2.54) n'est pas unique à chaque pas de calcul. Comme pour la régulation, il faut imposer des contraintes supplémentaires sur les $a_i(k)$ pour pouvoir résoudre le système.

Remarque 2 :

L'hypothèse $T_d \gg T_{mod}$ peut être comparée avec celles faites usuellement sur les valeurs des périodes d'échantillonnage par rapport aux périodes de discrétisation des trajectoires-consignes.

Pour que les performances de l'algorithme soient satisfaisantes, il est nécessaire que les variations de la trajectoire de référence soient lentes par rapport aux dynamiques du système, ce qui est une hypothèse usuelle pour effectuer de la poursuite de trajectoires.

II.4 COMMANDE EN BOUCLE FERMEE

L'algorithme proposé précédemment est fondé sur le calcul des coefficients a_i calculés avant l'implantation de la commande, à partir d'un modèle linéaire du système à commander. Ceci présente l'avantage de ne nécessiter aucun calcul en temps réel. Cependant ce type d'algorithme en boucle ouverte présente des inconvénients en présence de perturbations, les coefficients a_i étant déterminés a priori.

Un moyen d'améliorer la loi de commande est d'élaborer un algorithme de commande en boucle fermée. Pour cela, nous allons conserver le principe de calcul des coefficients a_i mais ces derniers seront réactualisés périodiquement en fonction du comportement du système réel. Nous traiterons d'abord la régulation autour d'une constante notée y_d puis nous envisagerons la poursuite de trajectoire. Le principe est de déterminer une nouvelle trajectoire de référence qui sera notée $y_r(t)$. Cette trajectoire sera élaborée de manière à ce que si le modèle (linéaire) du système poursuit $y_r(t)$ alors le système réel converge vers la consigne y_d .

II.4.1 Elaboration d'un algorithme de régulation

La trajectoire désirée est discrétisée à une période T_d de la même manière qu'au paragraphe précédent. Considérons $e=(e_1, \dots, e_s)^T$ un vecteur constant pré-défini et $\mathcal{D}_i, i=1, \dots, s$ des domaines bornés autour de y_d , définis par (fig.2.9) :

$$(y_i \in \mathcal{D}_i) \text{ ssi } (\forall i \in \{1, \dots, s\}, |y_i - y_d| \leq e_i) \tag{2.55}$$

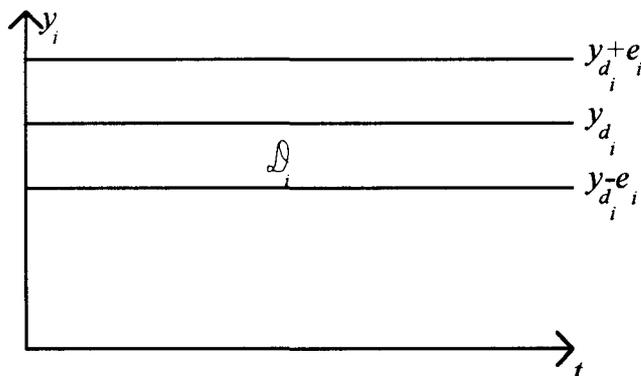


Fig.2.9 : Domaines autour de la consigne dans le cas constant.

Algorithme de régulation :

1) Détermination de $\varepsilon=(\varepsilon_1,\dots,\varepsilon_s)^T$ défini pour tout i , à l'instant kT_d par :

$$\varepsilon_i(kT_d) = |y_i(kT_d) - y_{d_i}| \quad (2.56)$$

2) Détermination de la trajectoire de référence y_r définie par :

Si, pour tout $i \in \{1, \dots, s\}$, $\varepsilon_i(kT_d) \leq e_i$, alors :

$$y_{r_i}(kT_d) = y_{d_i}$$

(commande en boucle ouverte)

sinon :

$$y_{r_i}(kT_d) = 2y_{d_i} - y_i((k-1)T_d)$$

(point symétrique de y à l'instant précédent par rapport à y_{d_i}) (2.57)

3) Détermination des coefficients $a_j(k)$ définis par :

$$\begin{cases} y_r(kT_d) = \sum_{j=1}^{2^m} a_j(k) y_{j\infty} \\ \sum_{j=1}^{2^m} a_j(k) = 1 \\ a_j(k) \in [0,1], j = 1, \dots, 2^m \end{cases} \quad (2.58)$$

en fonction de critères de choix déterministes conduisant à une solution unique définie.

La figure suivante donne un exemple de construction d'une trajectoire de référence (fig.2.10).

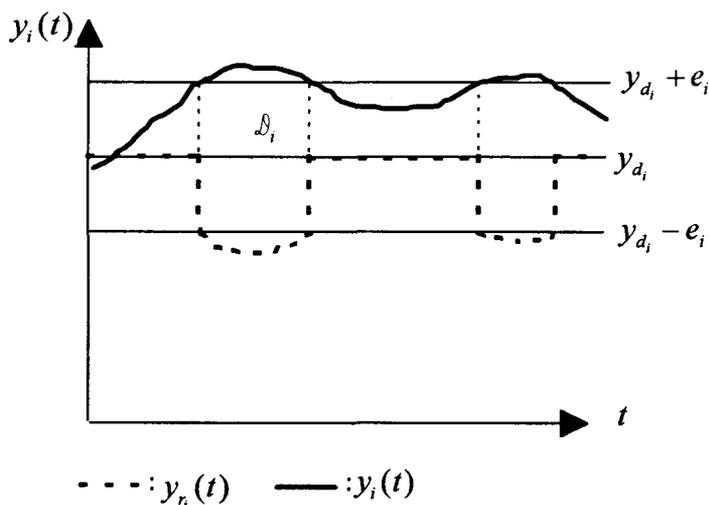


Fig.2.10 : Construction d'une trajectoire de référence.

II.4.2 Poursuite de trajectoire

Le but est ici d'élaborer un algorithme en boucle fermée permettant de poursuivre une trajectoire en sortie, $y_d(t)$. Nous allons, dans un premier temps discrétiser $y_d(t)$ en faisant les mêmes hypothèses sur la période de discrétisation T_d que dans le §II.3.5.

On construit ensuite autour de $y_{d_i}(t)$ des domaines \mathcal{D}_i définis de manière analogue que pour la régulation (fig.2.11):

$$(y_i \in \mathcal{D}_i) \text{ ssi } (\forall i \in \{1, \dots, s\}, |y_i(t) - y_{d_i}(t)| \leq e_i) \quad (2.59)$$

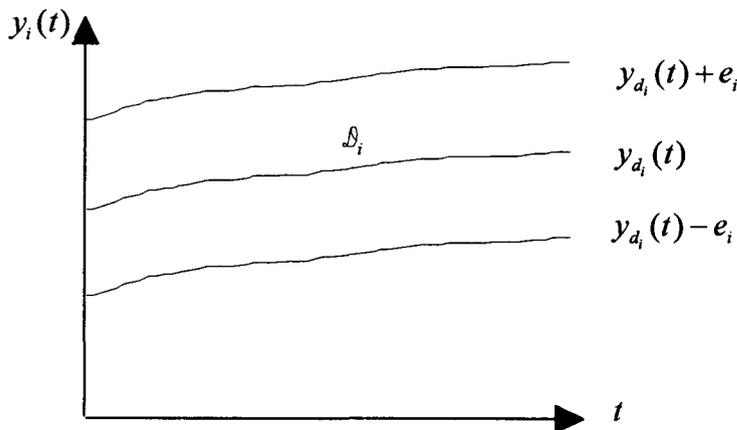


Fig.2.11. Domaines autour de la trajectoire à poursuivre.

Algorithme de poursuite de trajectoire :

1) Détermination du vecteur $\varepsilon = (\varepsilon_1, \dots, \varepsilon_s)^T$ défini pour tout i , à l'instant kT_d par :

$$\varepsilon_i(kT_d) = |y_i(kT_d) - y_{d_i}(kT_d)| \quad (2.60)$$

2) Détermination des points de la trajectoire de référence $y_r(kT_d)$ par la procédure suivante :

Si $\varepsilon_i((k-1)T_d) \leq e_i((k-1)T_d)$, alors :

$$y_r(kT_d) = y_{d_i}(kT_d)$$

(commande en boucle ouverte)

sinon :

$$y_r(kT_d) = 2y_{d_i}(kT_d) - y_i((k-1)T_d)$$

(point symétrique de y par rapport à y_d) (2.61)

3) Détermination des coefficients $a_j(k)$ en résolvant l'ensemble d'équations (2.58) associés à des critères de choix déterministes.

Un exemple de construction d'une trajectoire de référence peut être donné par la figure suivante (fig.2.12) :

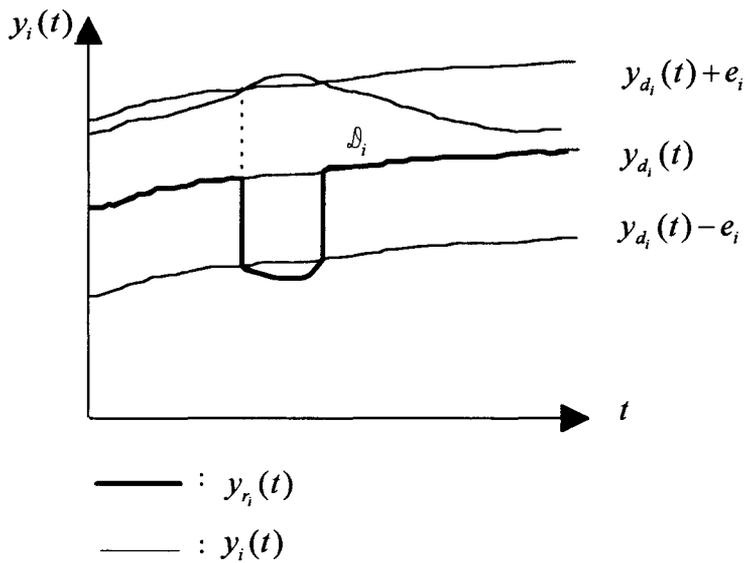


Fig.2.12 : Construction d'une trajectoire de référence.

L'algorithme peut alors être représenté sous forme de schéma-bloc (fig.2.13) :

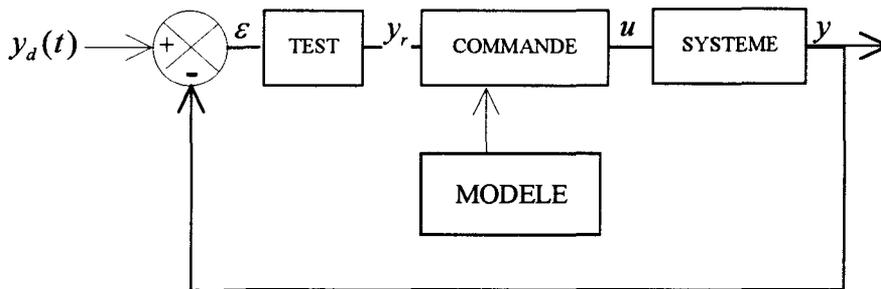


Fig.2.13 : Schéma-bloc de l'algorithme de poursuite de trajectoire.

Le bloc TEST détermine $y_r(t)$ à partir de $\varepsilon(t)$.

Le bloc COMMANDE élabore la séquence du vecteur u à partir de y_r .

Le bloc MODELE permet de calculer les coordonnées barycentriques $a_j(k)$ à partir de l'équation (2.58).

II.5 EXEMPLES D'APPLICATION

II.5.1 Exemple 1

Considérons le système linéaire à commutation représenté par l'équation d'état suivante:

$$\begin{cases} \dot{x} = \begin{pmatrix} -2 & 0 \\ 1 & -4 \end{pmatrix} x + \begin{pmatrix} -1 & 2 \\ 1 & 1 \end{pmatrix} u \\ y = x \end{cases} \quad (2.62)$$

avec $x = (x_1, x_2)^T$ et $u = (u_1, u_2)^T$. Nous avons, en utilisant les résultats précédents :

$$\begin{cases} x_{1\infty} = 0 \\ x_{2\infty} = -\mathbf{A}^{-1}\mathbf{B} \begin{pmatrix} 1 \\ 0 \end{pmatrix} = \begin{pmatrix} -1/2 \\ 1/8 \end{pmatrix} \\ x_{3\infty} = -\mathbf{A}^{-1}\mathbf{B} \begin{pmatrix} 0 \\ 1 \end{pmatrix} = \begin{pmatrix} 1 \\ 1/2 \end{pmatrix} \\ x_{4\infty} = -\mathbf{A}^{-1}\mathbf{B} \begin{pmatrix} 1 \\ 1 \end{pmatrix} = \begin{pmatrix} 1/2 \\ 5/8 \end{pmatrix} \end{cases} \quad (2.63)$$

Le domaine de convergence peut être représenté (fig.2.14) :

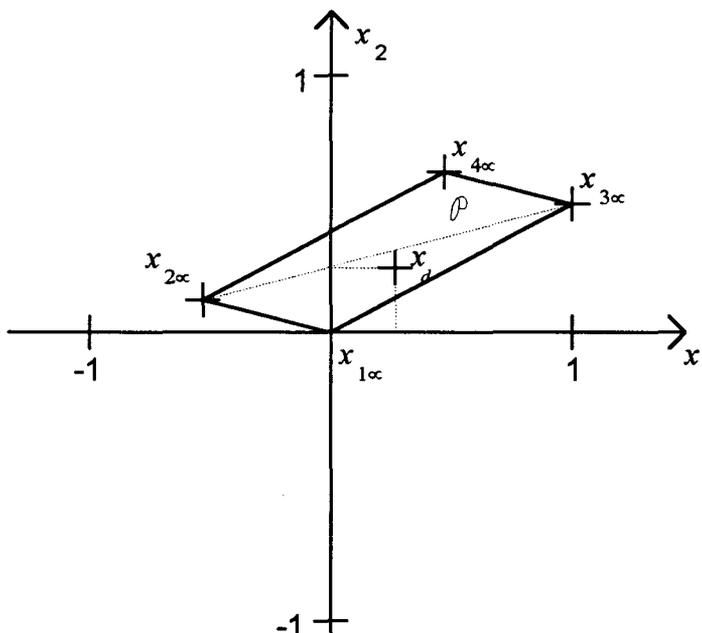


Fig.2.14 : Domaine de convergence du système à commutation (2.62).

Si on désire que le système converge, par exemple, vers $x_d = \left(\frac{1}{4}, \frac{1}{4}\right)^T$, ceci entraîne la résolution de l'ensemble d'équations :

$$\begin{cases} \frac{1}{4} = -\frac{1}{2}a_2 + a_3 + \frac{1}{2}a_4 \\ \frac{1}{4} = \frac{1}{8}a_2 + \frac{1}{2}a_3 + \frac{5}{8}a_4 \\ 1 = a_1 + a_2 + a_3 + a_4 \end{cases} \quad (2.64)$$

ce qui donne, avec a_4 comme paramètre :

$$\begin{cases} a_1 = \frac{1}{4} + a_4 \\ a_2 = \frac{1}{3} - a_4 \\ a_3 = \frac{5}{12} - a_4 \end{cases} \quad (2.65)$$

Toute valeur de a_4 comprise entre 0 et $\frac{1}{3}$ convient. On peut choisir, par exemple, $a_4=0$ ce qui donne les valeurs :

$$\begin{cases} a_1 = \frac{1}{4} \\ a_2 = \frac{1}{3} \\ a_3 = \frac{5}{12} \\ a_4 = 0 \end{cases} \quad (2.66)$$

Si nous choisissons pour norme de la matrice \mathbf{A} : $\|\mathbf{A}\| = \max_{i=1,2} (\|\mathbf{A}(i,j)\|, j=1,2)$ où $\mathbf{A}(i,j)$, $i, j=1,2$ représentent les coefficients de la matrice \mathbf{A} , on a : $\|\mathbf{A}\| = 4$. D'après les résultats présentés précédemment, nous devons choisir T_{mod} tel que : $\|\mathbf{A}T_{mod}\| \ll 1$, soit :

$$T_{mod} \ll \frac{1}{4} \quad (2.67)$$

Remarque :

Les valeurs-propres $\lambda_i(\mathbf{A})$, $i=1,2$ de \mathbf{A} sont : -2 et -4 . L'équation (2.67) est équivalente à :

$$T_{mod} \ll 1/|\lambda(\mathbf{A})|_{\max}$$

Les composantes de l'entrée u peuvent être représentées en fonction du temps par la figure suivante (fig.2.15) :

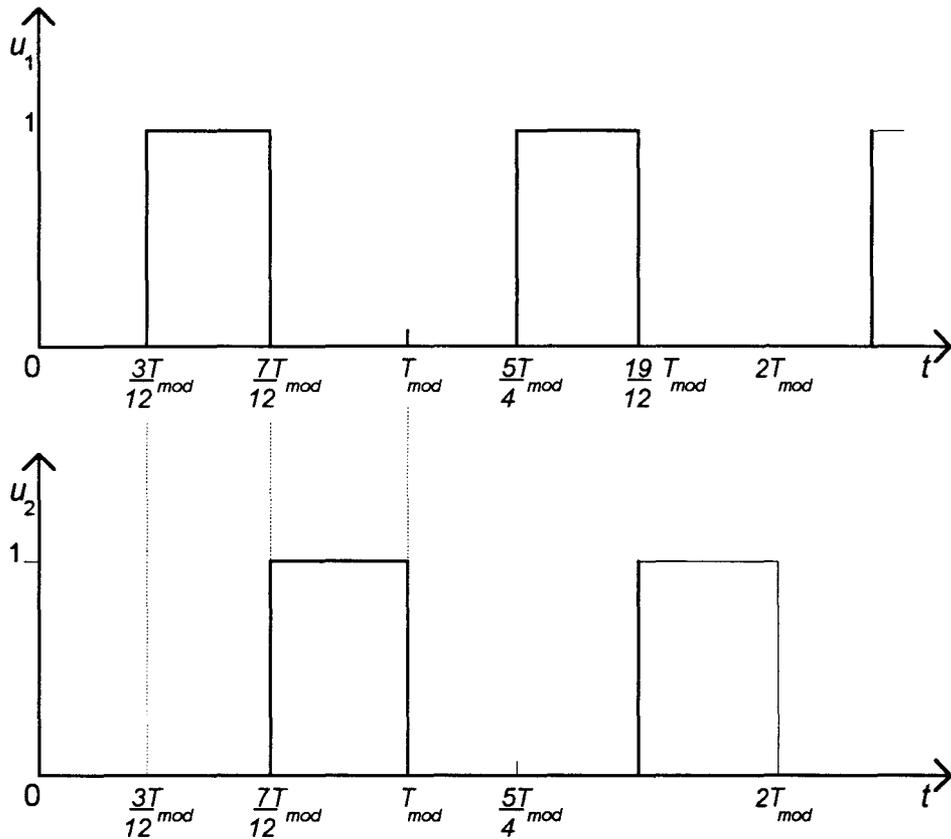


Fig.2.15 : Représentation des entrées du système linéaire à commutation.

Les réponses temporelles (fig.2.16) de l'état de ce modèle, ont été simulées en utilisant le logiciel de simulation A.C.S.L., avec $T_{mod}=10^{-3}$ s :

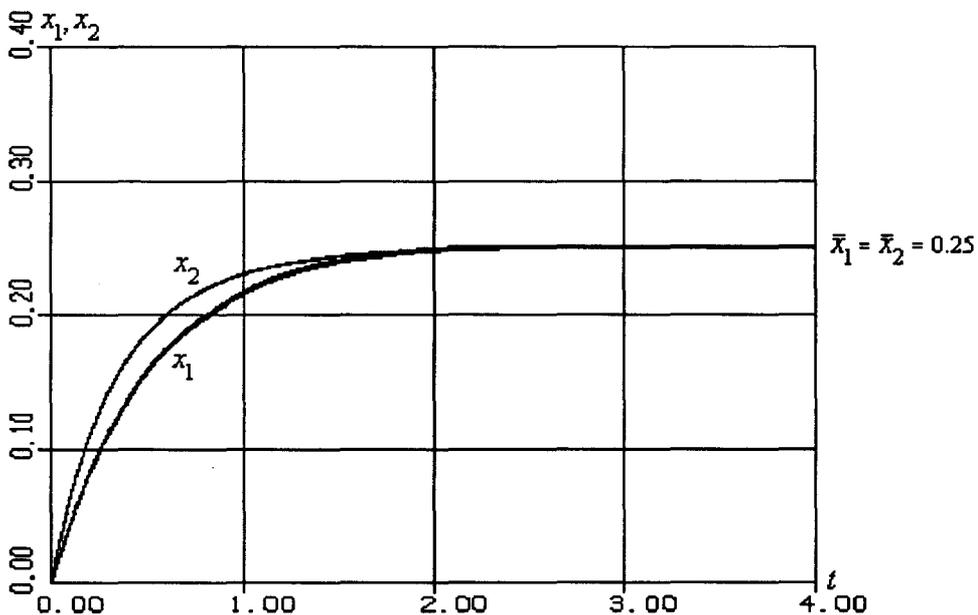


Fig.2.16 : Réponse temporelle des variables d'état de l'exemple 1.

II.5.2 Exemple 2

Considérons le circuit électrique défini comme suit (fig.2.17) :

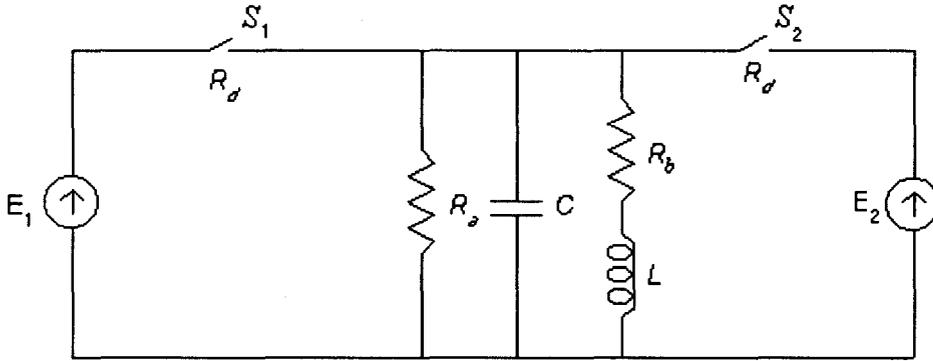


Fig.2.17 : Représentation d'un circuit électrique avec interrupteurs.

Les S_i , pour $i=1,2$ représentent des éléments de commutation dont les états bloqués et passants sont commandables. Ce type de circuits est modélisable par bond-graph (chapitre I) et on obtient une représentation unique valable quels que soient les états des commutateurs. Le bond-graph correspondant est présenté fig.2.18 :

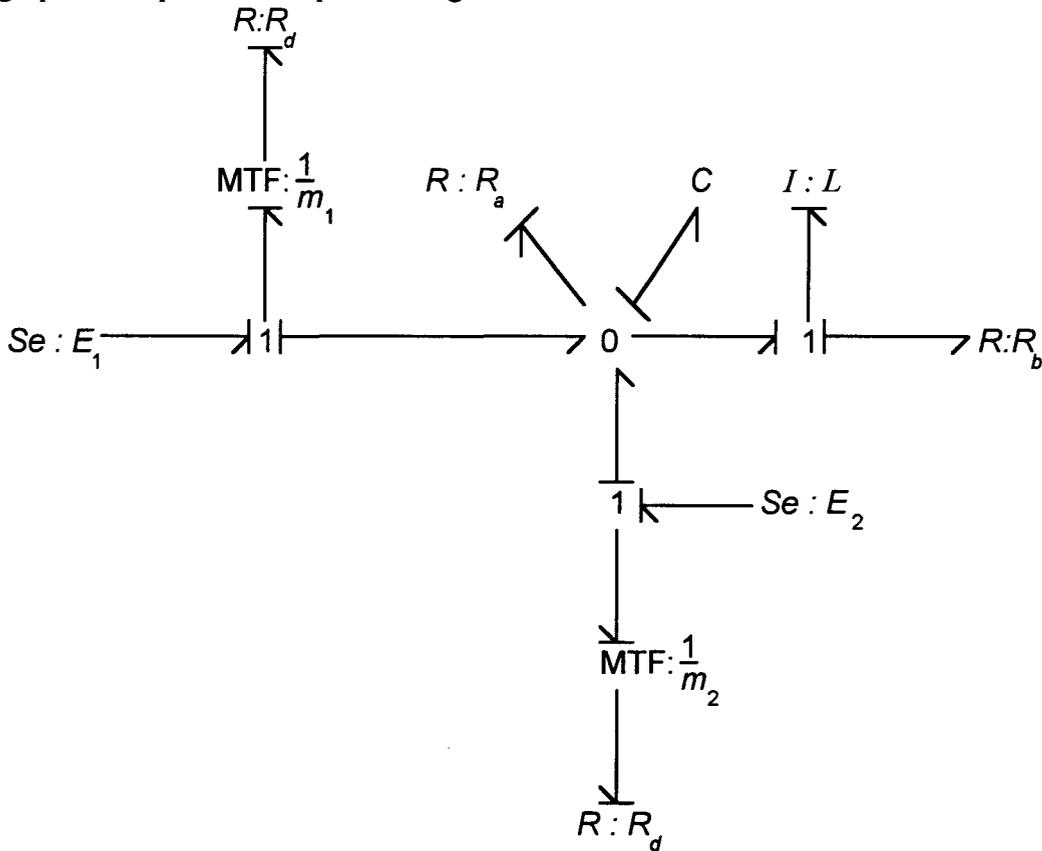


Fig.2.18 : Bond-graph du circuit électrique fig.2.17.

La théorie des bond-graphs, nous permet alors de déduire l'équation d'état :

$$\dot{x} = \begin{pmatrix} \dot{p} \\ \dot{q} \end{pmatrix} = \begin{pmatrix} -R_b / L & 1 / C \\ -1 / L & -1 / R_a C - \sum_{i=1}^2 m_i^2 / R_d C \end{pmatrix} x + \begin{pmatrix} 0 & 0 \\ E_1 / R_d & E_2 / R_d \end{pmatrix} \begin{pmatrix} m_1^2 \\ m_2^2 \end{pmatrix} \quad (2.68)$$

avec $x=(p,q)^T$ où p est le flux magnétique à travers l'inductance L et q la charge du condensateur C .

Si on pose $u = (m_1^2, m_2^2)^T$, nous obtenons l'équation d'état suivante :

$$\dot{x} = \mathbf{A}(u_i)x + \mathbf{B}u \quad (2.69)$$

Les configurations où S_1 et S_2 sont passants simultanément et S_1 et S_2 sont bloqués simultanément sont éliminées (si la configuration S_1, S_2 bloqués est autorisée, on obtient un modèle bilinéaire qui sera analysé dans le chapitre III : §III.1.3.1b).

Il en résulte que : $\sum_i m_i^2 = 1$ car $u \neq (1,1)^T$ et $u \neq (0,0)^T$.

On peut alors simplifier l'équation d'état :

$$\dot{x} = \begin{pmatrix} \dot{p} \\ \dot{q} \end{pmatrix} = \begin{pmatrix} -R_b / L & 1 / C \\ -1 / L & -1 / R_a C - 1 / R_d C \end{pmatrix} x + \begin{pmatrix} 0 & 0 \\ E_1 / R_d & E_2 / R_d \end{pmatrix} u \quad (2.70)$$

Considérons les paramètres électriques suivants :

$$\rightarrow R_b = 2\Omega, R_a = 4\Omega, R_d = 1\Omega$$

$$\rightarrow L = 200\text{mH}, C = 10\text{mF}$$

$$\rightarrow E_1 = +15\text{V}, E_2 = -15\text{V}$$

L'équation (2.70) devient :

$$\dot{x} = \begin{pmatrix} -10 & 100 \\ -5 & -125 \end{pmatrix} x + \begin{pmatrix} 0 & 0 \\ 15 & -15 \end{pmatrix} u$$

Nous appliquons alors les résultats de la partie II :

$$\begin{cases} x_{2\infty} = \begin{pmatrix} 0.857 \\ 0.085 \end{pmatrix} \\ x_{3\infty} = \begin{pmatrix} -0.857 \\ -0.085 \end{pmatrix} \end{cases} \quad (2.71)$$

L'égalité $x_{3\infty} = -x_{2\infty}$, entraîne que le polyèdre qui représente le domaine de convergence est le segment $[x_{3\infty}, x_{2\infty}]$ représenté fig.2.19 :

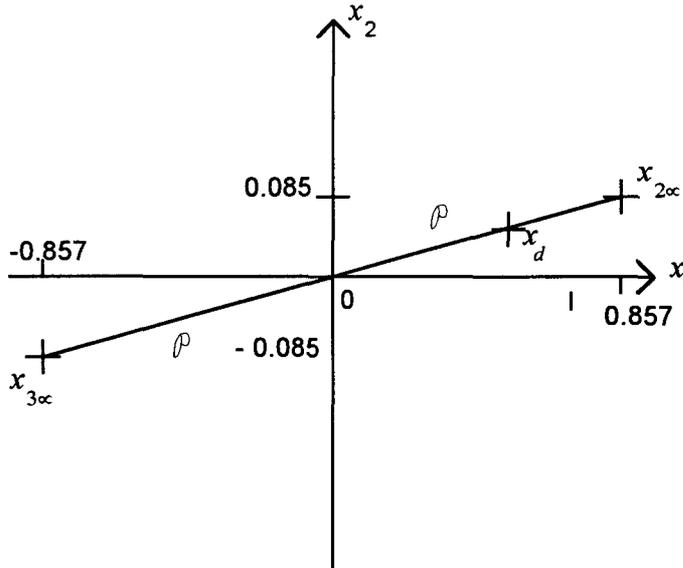


Fig.2.19 : Domaine de convergence du vecteur état du circuit électrique.

Considérons par exemple, $x_d = \begin{pmatrix} 0.50 \\ 0.05 \end{pmatrix}$. On doit alors résoudre le système suivant :

$$\begin{cases} 0.857a_2 - 0.857a_3 = 0.5 \\ 0.085a_2 - 0.085a_3 = 0.05 \\ a_2 + a_3 = 1 \end{cases} \quad (2.72)$$

soit :

$$\begin{cases} a_2 = 19/24 \\ a_3 = 5/24 \end{cases} \quad (2.73)$$

D'autre part, on a :

$$\|\mathbf{A}\| = 125,$$

ce qui nous impose le choix de T_{mod} de sorte que :

$$T_{mod} \ll 8 \cdot 10^{-3} \text{ s}$$

Les composantes de u peuvent alors être représentées (fig.2.20) :

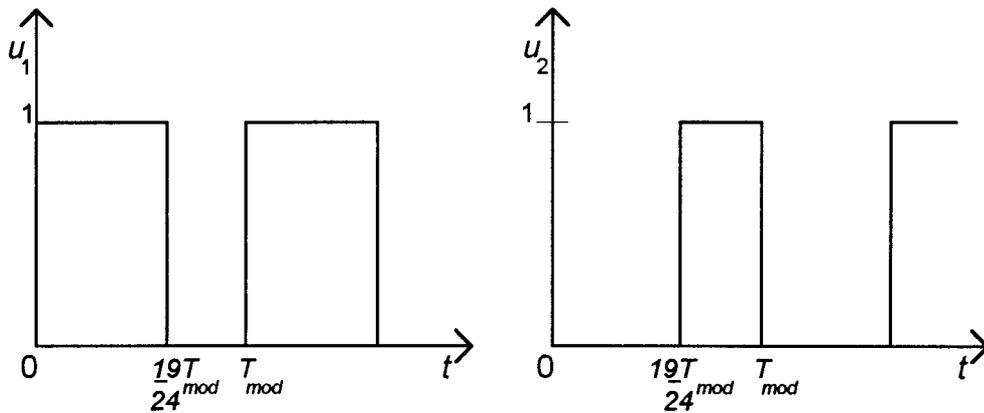


Fig.2.20 : Représentation des entrées du circuit électrique fig.2.17.

Les réponses temporelles du vecteur état de ce modèle ont été simulés avec A.C.S.L. sur un PC(486), avec $T_{mod}=10^{-4} \text{ s}$ (fig.2.21), à partir de conditions initiales nulles.

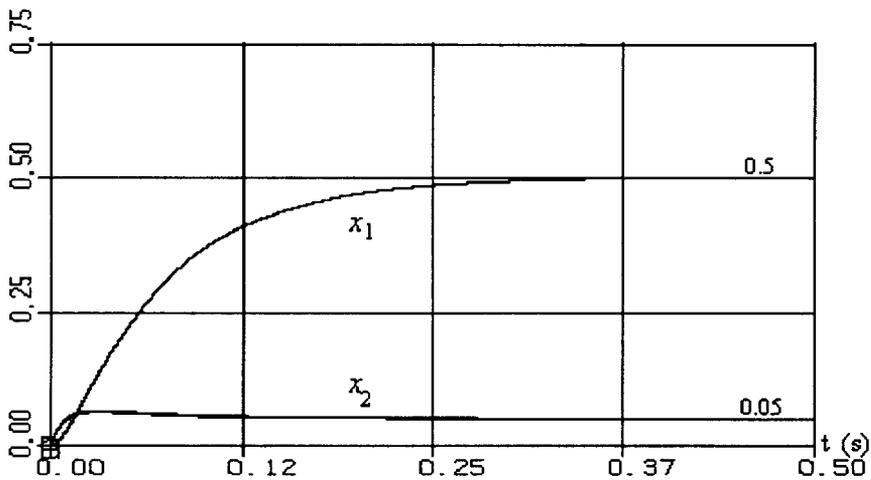


Fig.2.21 : Réponse temporelle des variables d'état du circuit fig.2.17.

CONCLUSION

Nous avons élaboré dans ce chapitre, une méthode pour commander les systèmes linéaires à entrées booléennes. Les principaux résultats sont la définition d'un domaine de convergence à l'intérieur duquel l'état ou la sortie peut converger et la détermination de lois de commande en boucle ouverte et en boucle fermée fondées sur l'élaboration de séquences périodiques des entrées. Nous avons démontré que si la matrice d'état est asymptotiquement stable, ou peut être rendue stable au préalable par une boucle de stabilisation, et si la période T_{mod} est choisie de manière adéquate, l'algorithme ainsi défini converge. Les commandes que nous avons élaborées dépendent du modèle d'état et sont soumises à sa validité. Cependant, le bond-graph représente un modèle de connaissance physique assez fiable, du moins pour la structure. Nous n'avons pas présenté dans ce chapitre d'étude de la sensibilité des algorithmes par rapport aux paramètres.

Il existe pour ces lois de commande, une contrainte portant sur le choix de T_{mod} par rapport aux dynamiques du système. Il apparaît que des dynamiques trop élevées pour le système à commander entraîneront le choix d'un T_{mod} petit, ce qui peut s'avérer gênant, surtout pour la poursuite de trajectoires en boucle fermée (volume de calcul important). D'autre part, il est à noter que le choix de la périodicité de la commande n'est pas obligatoire ; il serait intéressant d'envisager des types de séquences de commutation non-périodiques. Nous avons montré que le choix d'une séquence de commande n'est pas unique. Des critères d'optimisation ont été proposés et d'autres, comme l'optimisation des temps de réponse ou l'amélioration de la robustesse pourraient également être envisagés. Dans le chapitre suivant, nous allons montrer comment les algorithmes développés dans le cas linéaire peuvent s'appliquer au non-linéaire.

CHAPITRE III

COMMANDE BOOLEENNE DES SYSTEMES NON-LINEAIRES

CHAPITRE III

INTRODUCTION

En électronique de puissance les systèmes commandés par des éléments de commutation peuvent donner lieu après modélisation par bond-graph (chapitre I) à des équations d'état linéaires (hacheurs), bilinéaires (transformateurs) ou non-linéaires (associations moteurs électriques-convertisseurs de puissance) dont les composantes de l'entrée sont booléennes. Dans le chapitre précédent nous avons élaboré une méthode permettant de commander les systèmes linéaires à commutation.

Au cours de ce chapitre, nous traiterons dans un premier temps, en utilisant une technique analogue à celle développée au cours du chapitre II, le cas des systèmes bilinéaires. Le cas des systèmes non-linéaires sera ensuite envisagé. Nous proposerons tout d'abord d'utiliser les résultats du chapitre II pour une certaine classe de systèmes linéarisables par morceaux. Enfin, une méthode de découplage-linéarisation par retour d'état statique sera étudiée pour la commande des systèmes non-linéaires affines.

III.1 COMMANDE DES SYSTEMES BILINEAIRES A COMMUTATION

III.1.1 Préliminaires

Nous allons dans cette partie étudier l'élaboration d'une loi de commande pour les système bilinéaires à commandes booléennes [Rimaux, Abadie et Dauphin-Tanguy, 1994(a)].

Les systèmes bilinéaires peuvent être représentés par une équation différentielle d'état linéaire en l'état, linéaire en la commande, mais non linéaire conjointement, sous la forme :

$$\frac{dx}{dt} = \mathbf{A}x + \sum_{i=1}^m \mathbf{N}_i x u_i + \sum_{i=1}^m \mathbf{B}_i u_i \quad (3.1)$$

avec $x \in \mathbf{R}^n$; $u = (u_1, \dots, u_m)^T \in \mathbf{R}^m$; \mathbf{A}, \mathbf{N}_i , et \mathbf{B}_i , $i = 1, \dots, m$ sont des matrices constantes avec les dimensions appropriées. Chaque composante u_i est booléenne.

Pour commander de tels systèmes, nous allons utiliser la même loi de commande périodique que dans le cas linéaire. Notre but est alors de déterminer les 2^m inconnues α_i dans le but de

faire converger le système bilinéaire vers une consigne constante x_d . Nous avons montré au chapitre précédent que les α_i étaient égaux aux a_i (coordonnées barycentriques de x_d). Nous allons montrer que pour les systèmes bilinéaires, les coefficients α_i peuvent être déterminés en fonction de x_d par des relations linéaires.

Nous ne traiterons pas le cas de la commande en sortie, ni celui de la poursuite de trajectoires car ils peuvent être aisément déduits de l'étude menée au chapitre II.

Entre 2 commutations du système, le vecteur u est, comme on l'a vu précédemment, constant.

Si on suppose dans l'intervalle $[t_k, t_{k+1}[$ l'entrée égale à $[config(u)]_k$, alors l'intégration de l'équation (3.1) conduit à, pour $\tau \in [t_k, t_{k+1}[$:

$$x(\tau) = e^{(A+\mathcal{N}_k)(\tau-t_k)}x(t_k) + \int_{t_k}^{\tau} e^{(A+\mathcal{N}_k)(\tau-\theta)} \beta_k d\theta \quad (3.2)$$

avec $\mathcal{N}_k = \sum_{j=1}^m \mathbf{N}_j u_j^k$, $\beta_k = \sum_{j=1}^m \mathbf{B}_j u_j^k$, où u_j^k représente la j -ème composante de $[config(u)]_k$ pour tout $k \in \{1, \dots, 2^m\}$.

III.1.2 Elaboration de l'algorithme

III.1.2.1 Cas où toutes les matrices $(A + \mathcal{N}_k)$ sont asymptotiquement stables

Nous allons supposer dans ce paragraphe que les matrices $(A + \mathcal{N}_k)$ sont asymptotiquement stables ce qui entraîne que $(A + \mathcal{N}_k)^{-1}$ existe.

L'équation (3.2) devient alors pour $\tau \in [t_k, t_{k+1}[$:

$$x(\tau) = e^{(A+\mathcal{N}_k)(\tau-t_k)}x(t_k) - (A + \mathcal{N}_k)^{-1} [I - e^{(A+\mathcal{N}_k)(\tau-t_k)}] \beta_k$$

Le domaine temporel est de la même manière que dans le chapitre II (§II.3.1) divisé en intervalles $[t_0 + nT_{mod}, t_0 + (n+1)T_{mod}]$ subdivisé en intervalles $[t_k, t_{k+1}[$ tels que $t_{k+1} - t_k = \alpha_k T_{mod}$.

Nous pouvons écrire à chaque instant de commutation $t_0 + nT_{mod} + \sum_{i=1}^j \alpha_i T_{mod}$, $j \leq 2^m$, $n \in \mathbf{N}$:

$$x(t_0 + nT_{mod} + \alpha_1 T_{mod}) = e^{(A+\mathcal{N}_1)(\alpha_1 T_{mod})}x(t_0 + nT_{mod}) - (A + \mathcal{N}_1)^{-1} [I - e^{(A+\mathcal{N}_1)(\alpha_1 T_{mod})}] \beta_1$$

et :

$$x(t_0 + nT_{mod} + (\alpha_1 + \alpha_2)T_{mod}) = e^{(A+\mathcal{N}_2)(\alpha_2 T_{mod})}x(t_0 + nT_{mod} + \alpha_1 T_{mod}) - (A + \mathcal{N}_2)^{-1} [I - e^{(A+\mathcal{N}_2)(\alpha_2 T_{mod})}] \beta_2$$

et, à l'instant $t_0 + (n+1)T_{mod}$:

$$x(t_0 + (n+1)T_{mod}) = e^{\sum_{k=1}^{2^n} (A + \eta_k) \alpha_k T_{mod}} x(t_0 + nT_{mod}) - \sum_{k=1}^{2^n} \prod_{j=k+1}^{2^n} e^{(A + \eta_j) \alpha_j T_{mod}} (A + \eta_k)^{-1} (\mathbf{I} - e^{(A + \eta_k) \alpha_k T_{mod}}) \beta_k \quad (3.3)$$

Ce qui donne à l'infini : $\lim_{k \rightarrow \infty} x(t_k) = x_\infty$ avec,

$$x_\infty = - \left[\mathbf{I} - e^{\sum_{k=1}^{2^n} (A + \eta_k) \alpha_k T_{mod}} \right]^{-1} \sum_{k=1}^{2^n} e^{\sum_{j=k+1}^{2^n} (A + \eta_j) \alpha_j T_{mod}} (A + \eta_k)^{-1} (\mathbf{I} - e^{(A + \eta_k) \alpha_k T_{mod}}) \beta_k \quad (3.4)$$

Pour simplifier cette expression nous devons effectuer un développement du premier ordre, ce qui impose un choix de T_{mod} tel que :

$$\left\| \sum_{k=1}^{2^n} (A + \eta_k) \alpha_k \right\| T_{mod} \ll 1 \quad (3.5)$$

Après quelques calculs semblables à ceux présentés dans le chapitre II, nous obtenons l'approximation de x_∞ sous la forme :

$$x_\infty = - \left[\mathbf{A} + \sum_{i=1}^{2^m} \alpha_i \eta_i \right]^{-1} \sum_{i=1}^{2^m} \alpha_i \beta_i \quad (3.6)$$

On désire que le système converge vers x_d , soit $x_\infty = x_d$, ce qui entraîne :

$$\begin{cases} \left[\mathbf{A} + \sum_{i=1}^{2^m} \alpha_i \eta_i \right] x_d = - \sum_{i=1}^{2^m} \alpha_i \beta_i \\ \sum_{i=1}^{2^m} \alpha_i = 1 \\ 0 \leq \alpha_i \leq 1 \quad \forall i \in \{1, \dots, 2^m\} \end{cases} \quad (3.7)$$

ce qui équivaut à résoudre le système d'équations suivant :

$$\left\{ \begin{array}{l} \sum_{i=1}^{2^m} \alpha_i (\eta_i x_d + \beta_i) = -\mathbf{A}x_d \\ \sum_{i=1}^{2^m} \alpha_i = 1 \\ 0 \leq \alpha_i \leq 1 \quad \forall i \in \{1, \dots, 2^m\} \end{array} \right. \quad (3.8)$$

III.1.2.2 Cas général

Si pour tout i , $(\mathbf{A} + \eta_i)$ est instable, il est impossible de stabiliser le système avec une loi de commande booléenne. Nous allons donc supposer qu'il existe i_s tel que $(\mathbf{A} + \eta_{i_s})$ est asymptotiquement stable.

a) *Cas où toutes les matrices $(\mathbf{A} + \eta_i)$ $i \in \{1, \dots, 2^m\}$ sont inversibles*

La résolution de l'équation (3.3) donne :

$$\left\{ \begin{array}{l} \mathbf{x}(t_0 + nT_{mod}) = e^{\mathbf{M}T_{mod}} \mathbf{x}(t_0) - \left(\mathbf{I} - e^{\mathbf{M}T_{mod}} \right)^{-1} \sum_{k=1}^{2^m} e^{j=k+1} \sum_{j=k+1}^{2^m} (\mathbf{A} + \eta_j)^{\alpha_j T_{mod}} (\mathbf{A} + \eta_k)^{-1} (\mathbf{I} - e^{(\mathbf{A} + \eta_k) \alpha_k T_{mod}}) \beta_k \\ \text{avec } \mathbf{M} = \sum_{k=1}^{2^m} (\mathbf{A} + \eta_k) \alpha_k \end{array} \right. \quad (3.9)$$

Pour que cette équation converge, il faut que la matrice $\mathbf{M} = \sum_{k=1}^{2^m} (\mathbf{A} + \eta_k) \alpha_k$ soit asymptotiquement stable.

Pour assurer la stabilité du système et pour que le système converge vers x_d , il faut donc choisir les α_i qui vérifient :

$$\left\{ \begin{array}{l} -\mathbf{A}x_d = \sum_{i=1}^{2^m} \alpha_i (\eta_i x_d + \beta_i) \\ \sum_{i=1}^{2^m} \alpha_i = 1 \\ \alpha_i \in [0, 1], i = 1, \dots, 2^m \\ \sum_{j=1}^{2^m} (\mathbf{A} + \eta_j) \alpha_j \text{ stable.} \end{array} \right. \quad (3.10)$$

Si nous supposons que $i_s=1$, nous pouvons représenter la commande (fig.3.1) :

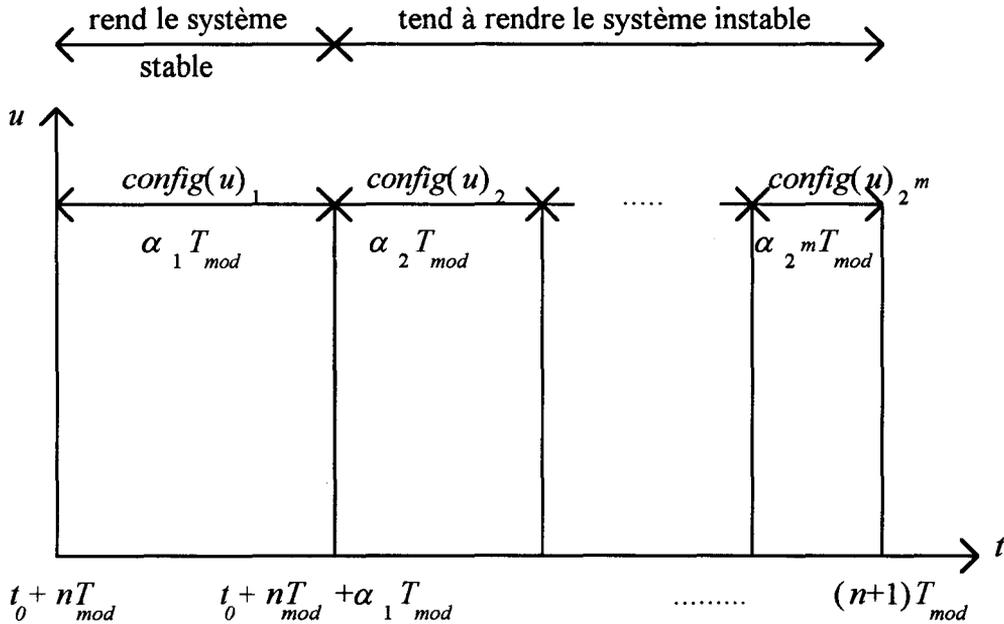


Fig.3.1 : Implantation de l'algorithme de commande.

b) Cas où il existe un i tel que $(A + \eta_i)$ est non inversible

Ce cas représente en fait le cas limite entre les deux cas précédents. Si nous introduisons un terme ε réel positif dans la matrice A de telle sorte que 0 ne soit plus une valeur propre de $(A + \eta_i)$, le calcul du paragraphe précédent peut être appliqué, et pour $\varepsilon \rightarrow 0$ nous retrouvons la même condition (voir annexe 5).

Remarques concernant le domaine de convergence :

Dans le cas où tous les $(A + \eta_i)$ sont stables, la notion de domaine de convergence associée à notre algorithme peut être définie de la même manière que dans le cas linéaire, et on peut construire un polyèdre à partir des valeurs limites obtenues en appliquant la même configuration sur tout le domaine temporel. Dans le cas où il existe un $(A + \eta_i)$ instable, le domaine de convergence n'est plus un espace borné, mais un cône "ouvert sur l'infini".

III.1.3 Exemples d'application

III.1.3.1 Cas où tous les $(A + \eta_i)$ sont asymptotiquement stables

a) Exemple n°1

Considérons le système bilinéaire mono-entrée :

$$\dot{x} = \mathbf{A}x + \mathbf{N}xu + \mathbf{B}u.$$

La loi de commande est $u=0$ pendant αT_{mod} et $u=1$ durant le reste de la période $(1-\alpha)T_{mod}$

On a, avec les notations précédentes :

$$\begin{aligned} \eta_1 &= \mathbf{0} & \eta_2 &= \mathbf{N} \\ \beta_1 &= \mathbf{0} & \beta_2 &= \mathbf{B} \end{aligned}$$

Nous obtenons avec l'équation (3.8) :

$$0 + (1 - \alpha)(\mathbf{N}x_d + \mathbf{B}) = -\mathbf{A}x_d$$

ce qui équivaut à :

$$(\mathbf{A} + \mathbf{N})x_d + \mathbf{B} = \alpha(\mathbf{N}x_d + \mathbf{B})$$

Nous pouvons voir dans cet exemple que si la dimension de x est plus grande que 1 nous aurons plus d'équations que d'inconnues. Les valeurs possibles de x_d appartiendront alors à un sous-espace de dimension égale à la différence entre le nombre d'équations et le nombre de paramètre qui vaut 1 dans le cas présent.

b) Exemple n°2

Reprenons le système électrique du chapitre II (§II.5.2, fig.2.17). Si la configuration où S1 et S2 sont bloqués simultanément est retenue comme possible, alors $u=(u_1, u_2)^T$ peut prendre la valeur $(0,0)^T$ (la configuration $u=(1,1)^T$ n'est pas retenue car elle conduirait à un court-circuit entre les sources E_1 et E_2). Le modèle est alors bilinéaire et s'écrit :

$$\dot{x} = \begin{pmatrix} -10 & +100 \\ -5 & -25 \end{pmatrix} x + \begin{pmatrix} 0 & 0 \\ 0 & -100 \end{pmatrix} x u_1 + \begin{pmatrix} 0 & 0 \\ 0 & -100 \end{pmatrix} x u_2 + \begin{pmatrix} 0 \\ 15 \end{pmatrix} u_1 + \begin{pmatrix} 0 \\ -15 \end{pmatrix} u_2$$

Nous avons :

$$[\text{config}(u)]_1 = \begin{pmatrix} 0 \\ 0 \end{pmatrix} \quad \eta_1 = \begin{pmatrix} 0 & 0 \\ 0 & 0 \end{pmatrix} \quad \beta_1 = \begin{pmatrix} 0 \\ 0 \end{pmatrix}$$

$$[\text{config}(u)]_2 = \begin{pmatrix} 1 \\ 0 \end{pmatrix} \quad \eta_2 = \begin{pmatrix} 0 & 0 \\ 0 & -100 \end{pmatrix} \quad \beta_2 = \begin{pmatrix} 0 \\ 15 \end{pmatrix}$$

$$[config(u)]_3 = \begin{pmatrix} 0 \\ 1 \end{pmatrix} \quad \eta_3 = \begin{pmatrix} 0 & 0 \\ 0 & -100 \end{pmatrix} \quad \beta_3 = \begin{pmatrix} 0 \\ -15 \end{pmatrix}$$

$[config(u)]_4$ n'est pas calculée car elle correspond à $u=(1,1)^T$ (configuration éliminée). On est ici dans le cas où tous les $A+\eta_i$ sont stables. On peut donc utiliser l'équation (3.8) pour calculer les α_i , ce qui entraîne :

$$\alpha_2 \begin{pmatrix} 0 & 0 \\ 0 & -100 \end{pmatrix} x_d + \begin{pmatrix} 0 \\ 15 \end{pmatrix} + \alpha_3 \left[\begin{pmatrix} 0 & 0 \\ 0 & -100 \end{pmatrix} x_d + \begin{pmatrix} 0 \\ -15 \end{pmatrix} \right] = \begin{pmatrix} 10 & -100 \\ 5 & 25 \end{pmatrix} x_d$$

ce qui donne avec $x_d = \begin{pmatrix} x_{d_1} \\ x_{d_2} \end{pmatrix}$:

$$\begin{cases} 10x_{d_1} - 100x_{d_2} = 0 \\ -100\alpha_2 x_{d_2} + 15\alpha_2 - 100\alpha_3 x_{d_2} - 15\alpha_3 = 5x_{d_1} + 25x_{d_2} \end{cases}$$

Si nous choisissons comme au §II.5.2 $x_d = \begin{pmatrix} 0.5 \\ 0.05 \end{pmatrix}$, nous obtenons alors, en prenant α_3 comme paramètre :

$$\begin{cases} \alpha_1 = \frac{5}{8} - 3\alpha_3 \\ \alpha_2 = \frac{3}{8} + 2\alpha_3 \end{cases}$$

Les contraintes sur les coefficients α_i entraînent que l'on a :

$$\begin{cases} 0 \leq \alpha_1 \leq \frac{5}{8} \\ \frac{3}{8} \leq \alpha_2 \leq 1 \\ 0 \leq \alpha_3 \leq \frac{5}{24} \end{cases}$$

Le choix de $\alpha_1=0$ (configuration $u=(0,0)^T$ non autorisée) entraîne $(\alpha_2, \alpha_3) = \left(\frac{19}{24}, \frac{5}{24}\right)$, ce qui donne le résultat du linéaire (§II.5.2).

Le calcul de l'énergie en entrée E donne sur $[t_0+nT_{mod}, t_0+(n+1)T_{mod}]$:

$$E = (\alpha_2 + \alpha_3)T_{mod} = (1 - \alpha_1)T_{mod}$$

Cette énergie est donc minimale si α_1 est maximum, ce qui correspond ici à $\alpha_1 = \frac{5}{8}$. L'énergie en entrée vaut alors $\frac{3}{8}T_{mod}$.

Le système correspondant à $\alpha_1 = \frac{5}{8}, \alpha_2 = \frac{3}{8}, \alpha_3 = 0$ (cas "bilinéaire" avec énergie en entrée minimale) a été simulé avec $T_{mod} = 8 \cdot 10^{-5} s$ et représenté (fig.3.2) avec les courbes obtenues pour $\alpha_1 = 0, \alpha_2 = \frac{19}{24}, \alpha_3 = \frac{5}{24}$ (cas "linéaire") :

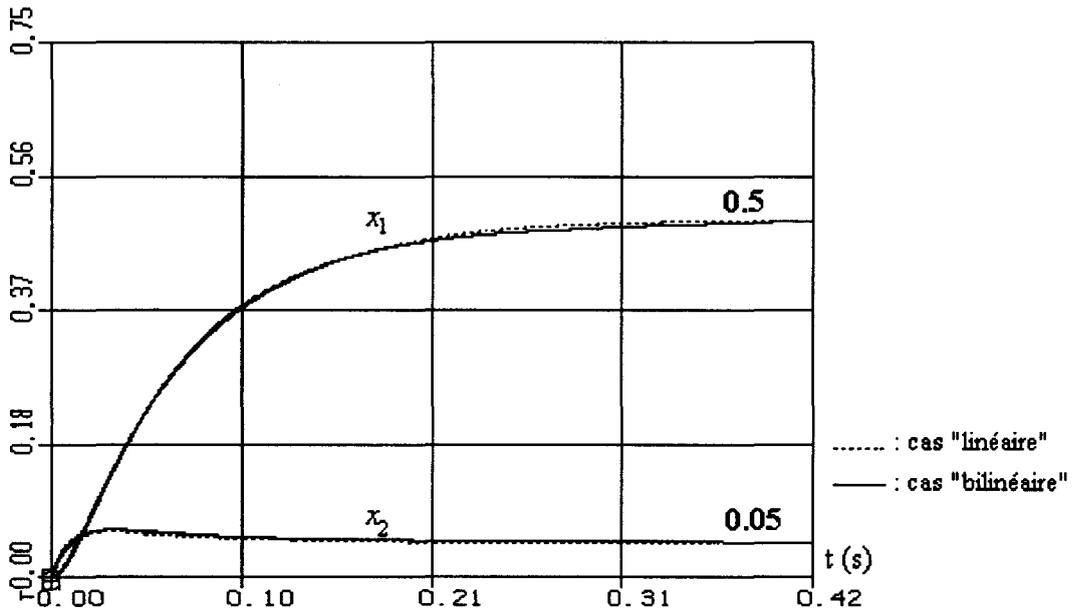


Fig.3.2 : Comportement du système électrique fig.2.17.

La commande minimisant l'énergie en entrée élaborée dans ce paragraphe donne une dynamique de convergence légèrement moins rapide que la loi de commande élaborée dans le cadre linéaire en §II.5.2.

III.1.3.2 Cas général

Considérons le système décrit par :

$$\dot{x} = \begin{pmatrix} 0 & 1 \\ -2 & -4 \end{pmatrix} x + \begin{pmatrix} 0 & 0 \\ 4 & 6 \end{pmatrix} x u_1 + \begin{pmatrix} 0 & 0 \\ 0 & -1 \end{pmatrix} x u_2 + \begin{pmatrix} 1 \\ 0 \end{pmatrix} u_1 + \begin{pmatrix} 1 \\ 1 \end{pmatrix} u_2$$

Les η_i et β_i peuvent être calculés :

$$\eta_1 = \begin{pmatrix} 0 & 0 \\ 0 & 0 \end{pmatrix} \quad \beta_1 = \begin{pmatrix} 0 \\ 0 \end{pmatrix} \quad \eta_2 = \begin{pmatrix} 0 & 0 \\ 4 & 6 \end{pmatrix} \quad \beta_2 = \begin{pmatrix} 1 \\ 0 \end{pmatrix}$$

$$\eta_3 = \begin{pmatrix} 0 & 0 \\ 0 & -1 \end{pmatrix} \quad \beta_3 = \begin{pmatrix} 1 \\ 1 \end{pmatrix} \quad \eta_4 = \begin{pmatrix} 0 & 0 \\ 4 & 5 \end{pmatrix} \quad \beta_4 = \begin{pmatrix} 2 \\ 1 \end{pmatrix}$$

On a :

$$\mathbf{A} + \eta_1 = \begin{pmatrix} 0 & 1 \\ -2 & -4 \end{pmatrix} \quad \mathbf{A} + \eta_2 = \begin{pmatrix} 0 & 1 \\ 2 & 2 \end{pmatrix} \quad \mathbf{A} + \eta_3 = \begin{pmatrix} 0 & 0 \\ 0 & -1 \end{pmatrix} \quad \mathbf{A} + \eta_4 = \begin{pmatrix} 0 & 1 \\ 2 & 1 \end{pmatrix}$$

$\mathbf{A} + \eta_1$ est la seule matrice asymptotiquement stable.

Nous devons donc appliquer les conditions établies en (3.10) pour faire converger le système vers x_d :

$$\left\{ \begin{array}{l} x_d = - \left(\begin{array}{cc} 0 & 1 \\ -2 + 4\alpha_2 + 4\alpha_4 & -4 + 6\alpha_2 + 5\alpha_4 - \alpha_3 \end{array} \right)^{-1} \sum_{i=1}^{2m} \alpha_i \beta_i \\ \sum_{i=1}^{2m} \alpha_i = 1 \\ \left(\begin{array}{cc} 0 & 1 \\ -2 + 4\alpha_2 + 4\alpha_4 & -4 + 6\alpha_2 + 5\alpha_4 - \alpha_3 \end{array} \right) \text{ stable} \end{array} \right.$$

De plus, on a :

$$\det \left(\begin{array}{cc} 0 & 1 \\ -2 + 4\alpha_2 + 4\alpha_4 & -4 + 6\alpha_2 + 5\alpha_4 - \alpha_3 \end{array} \right) = -(-2 + 4\alpha_2 + 4\alpha_4)$$

L'annulation du déterminant est obtenue pour $\alpha_2 + \alpha_4 = 1/2$ et une configuration stable est $\alpha_1 = 1$. La matrice \mathbf{A} est stable, ce qui implique que nous devons choisir les coefficients tels que $\alpha_2 + \alpha_4 < 1/2$.

Si nous prenons, par exemple $x_d = (1, -1/3)^T$ et $\alpha_3 = \alpha_4 = 0$, la résolution du système précédent donne $\alpha_1 = 2/3$ et $\alpha_2 = 1/3$. Nous devons alors choisir T_{mod} tel que (3.5) soit vérifiée, i.e :

$$\left\| \begin{array}{cc} 0 & 1 \\ -2/3 & -2 \end{array} \right\|_{T_{mod}} \ll 1$$

soit en reprenant la définition utilisée au §II.5.1 pour la norme de matrice : $T_{mod} \ll \frac{1}{2}$.

Le système précédent a été simulé (logiciel ACSL) en prenant $T_{mod} = 10^{-2}$ s (fig.3.3) :

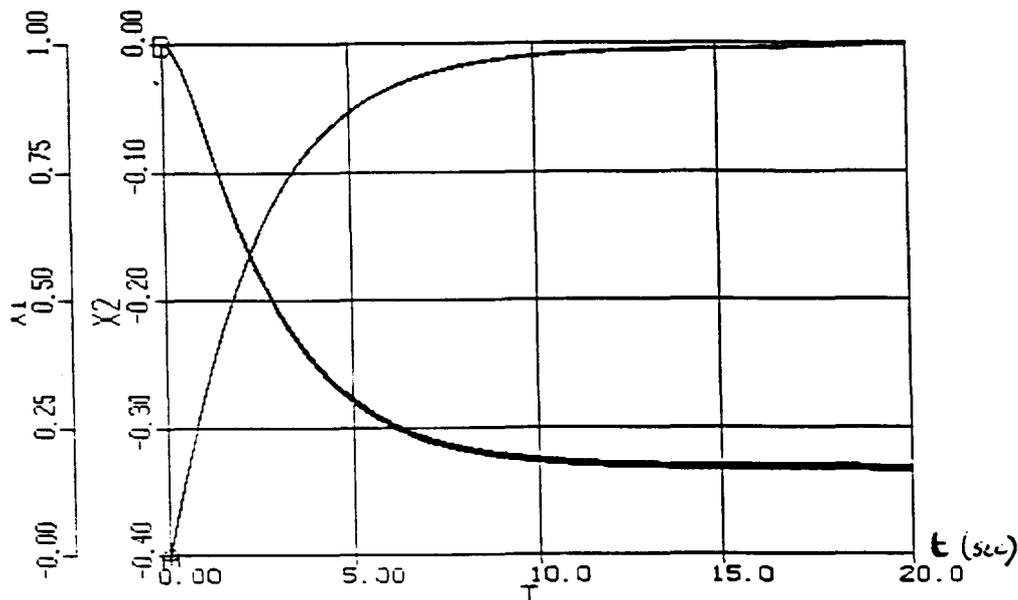


Fig.3.3 : Réponse temporelle de l'état du système.

III.1.4 Conclusion du cas bilinéaire

Nous avons élaboré un algorithme permettant de commander les systèmes bilinéaires à commutation. L'étude qui a été menée nous a montré que l'on pouvait établir une loi de commande cyclique en boucle ouverte analogue à celle élaborée pour les systèmes linéaires.

De plus, nous avons vu qu'il était possible d'utiliser des configurations du système a priori instables à condition que le temps d'application de ces configurations vérifie certaines contraintes, l'intérêt d'une telle démarche étant de donner la possibilité à l'état de converger à l'intérieur d'un domaine non-borné (contrairement au linéaire).

D'autre part, l'étude de l'exemple électrique du chapitre II (§II.5.2) a montré que le modèle bilinéaire permettait l'utilisation des degrés de liberté apparaissant dans le choix des coefficients α_i en fonction des performances requises (dynamique de convergence plus élevée ou minimisation de l'énergie en entrée).

III.2 SYSTEMES NON-LINEAIRES : LINEARISATIONS PAR MORCEAUX

Dans cette partie, nous allons calculer les lois de commande des modèles linéaires issus d'une linéarisation par morceaux du système étudié, en utilisant les résultats du chapitre II. Les algorithmes ainsi définis seront alors appliqués sur le système non-linéaire. Les performances du système seront ensuite améliorées, en appliquant un algorithme en boucle fermée calculé à partir de l'erreur en sortie du système.

III.2.1 Linéarisation du système étudié

Considérons le système défini par :

$$(\Sigma) : \begin{cases} \dot{x} = f(x, u) \\ y = h(x) \end{cases} \quad (3.11)$$

avec x , y et u définis comme précédemment, $f(x, u)$ une fonction continue de $\mathbf{R}^n \times \mathbf{R}^m$ à valeurs dans \mathbf{R}^n et $h(x)$ une fonction continue de \mathbf{R}^n vers \mathbf{R}^s .

On suppose qu'il existe T_d tel que (Σ) admet un modèle linéarisé sur chaque intervalle $[kT_d, (k+1)T_d[$, pour tout k entier tel que :

$$(\Sigma_k) : \begin{cases} \dot{x}(t) = \mathbf{A}(kT_d)x(t) + \mathbf{B}(kT_d)u(t) \\ y(t) = \mathbf{C}(kT_d)x(t) \end{cases} \quad (3.12)$$

avec pour tout $k \in \mathbf{N}$, $\mathbf{A}(kT_d) \in \mathbf{R}^n \times \mathbf{R}^n$, $\mathbf{B}(kT_d) \in \mathbf{R}^n \times \mathbf{R}^m$ et $\mathbf{C}(kT_d) \in \mathbf{R}^s \times \mathbf{R}^n$.

(Σ') est défini comme le système linéaire par morceaux décrit par (Σ_k) sur chaque intervalle $[kT_d, (k+1)T_d[$, pour k entier.

Le but est d'élaborer un algorithme pour commander en sortie (régulation, poursuite de trajectoire) le système décrit par (3.11).

III.2.2 Commande en boucle ouverte

III.2.2.1 Régulation

Le but est ici de faire converger la sortie du système étudié vers une consigne constante que l'on notera y_d .

Si nous définissons $x_{i\infty}(kT_d)$ et $y_{i\infty}(kT_d)$, pour $i=1, \dots, 2^m$ et $k \in \mathbb{N}$ tels que :

$$\begin{cases} x_{i\infty}(kT_d) = -\mathbf{A}^{-1}(kT_d)\mathbf{B}(kT_d)[\text{config}(u)]_i \\ y_{i\infty}(kT_d) = -\mathbf{C}\mathbf{A}^{-1}(kT_d)\mathbf{B}(kT_d)[\text{config}(u)]_i \end{cases} \quad (3.13)$$

$\rho_x(k)$ et $\rho_y(k)$ représentent les polyèdres convexes formés respectivement par les sommets $\{x_{1\infty}(kT_d), x_{2\infty}(kT_d), \dots, x_{2^m\infty}(kT_d)\}$ et $\{y_{1\infty}(kT_d), y_{2\infty}(kT_d), \dots, y_{2^m\infty}(kT_d)\}$.

Nous supposons dans ce qui va suivre que, quel que soit $k \in \mathbb{N}$, on a $y_d \in \rho_y(k)$.

Par analogie avec le chapitre II, nous pouvons définir les coefficients $a_i(kT_d)$ pour $i=1, \dots, 2^m$ comme étant les coordonnées barycentriques de y_d par rapport aux points $\{y_{1\infty}(kT_d), y_{2\infty}(kT_d), \dots, y_{2^m\infty}(kT_d)\}$.

On a alors :

$$\begin{cases} y_d(kT_d) = \sum_{i=1}^{2^m} a_i(kT_d) y_{i\infty}(kT_d) \\ \sum_{i=1}^{2^m} a_i(kT_d) = 1 \\ a_i(kT_d) \in [0,1], \quad i = 1, \dots, 2^m \end{cases} \quad (3.14)$$

Nous pouvons maintenant définir l'algorithme en utilisant les coefficients $a_i(kT_d)$ pour commander les entrées booléennes, déterminés en résolvant l'équation (3.14) associée à des critères déterministes.

Chaque intervalle $[kT_d, (k+1)T_d[$ est subdivisé en intervalles de longueur T_{mod} et sur chaque $[nT_{mod}, (n+1)T_{mod}[$, on applique la même séquence comme dans le cas du linéaire (voir chapitre II, §II.3.1), i.e. $u = [\text{config}(u)]_p$ pendant $a_p(kT_d)T_{mod}$.

Ceci peut être représenté (fig.3.4) :

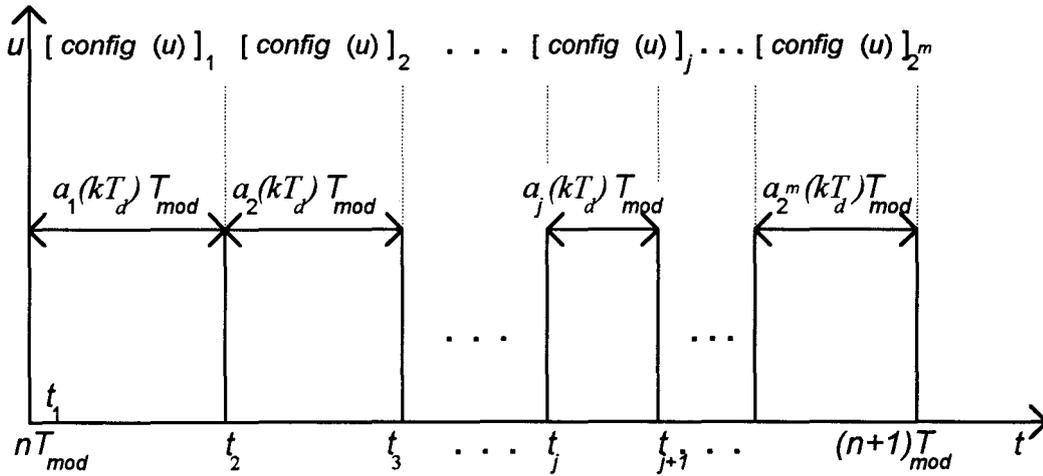


Fig.3.4 : Représentation de l'entrée dans le cas non-linéaire.

La commande ainsi définie est appliquée sur le système non-linéaire décrit par l'équation (3.11).

La procédure de détermination de l'algorithme de commande peut alors être représentée (fig.3.5) :

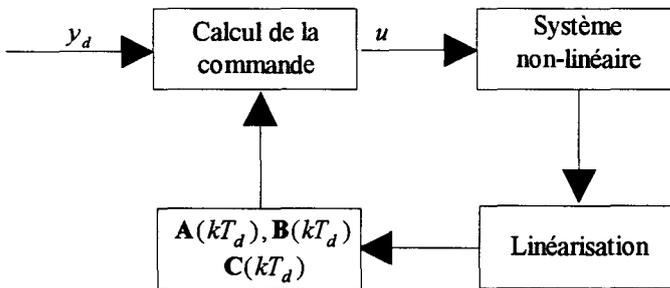


Fig. 3.5 : Schéma-bloc de l'algorithme en boucle ouverte.

III.2.2.2 Poursuite de trajectoires

Nous pouvons envisager d'appliquer la méthode décrite dans le paragraphe précédent à la poursuite de trajectoires. Pour cela, une méthode simple est de discrétiser la trajectoire consigne y_d avec la période T_d et de supposer que sur chaque intervalle $[kT_d, (k+1)T_d]$, on a $y_d(t) \cong y_d(kT_d)$.

Dans ce cas, il faut déterminer les coordonnées barycentriques de $y_d(kT_d)$:

$$\begin{cases} y_d(kT_d) = \sum_{i=1}^{2^m} a_i(kT_d) y_{i0}(kT_d) \\ \sum_{i=1}^{2^m} a_i(kT_d) = 1 \end{cases} \quad (3.15)$$

avec pour $i=1, \dots, 2^m$, $a_i(kT_d) \in [0, 1]$.

On applique alors pour le vecteur u la même séquence de commande que celle décrite au paragraphe §III.2.2.1.

L'algorithme ainsi défini peut être schématisé (fig.3.6) :

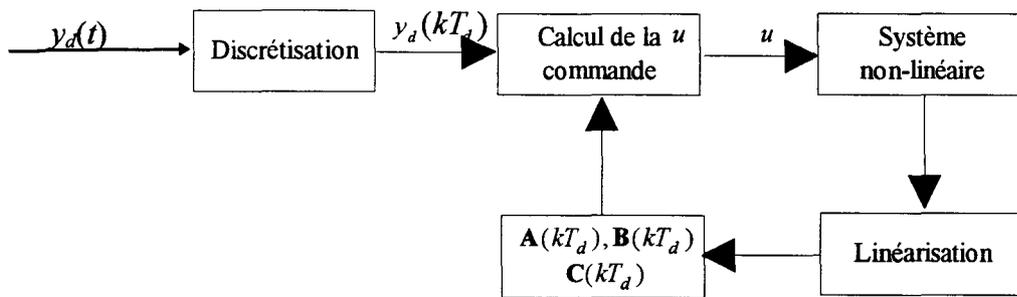


Fig.3.6 : Algorithme en boucle ouverte de poursuite de trajectoire.

III.2.3 Commande en boucle fermée

III.2.3.1 Elaboration de l'algorithme

Pour élaborer une loi de commande en boucle fermée afin d'améliorer le comportement du système, nous allons appliquer les résultats du chapitre II concernant la commande en boucle fermée des systèmes linéaires sur les modèles linéarisés (Σ_k) , la mesure de l'erreur étant effectuée sur le système non-linéaire (Σ) .

La trajectoire consignée à poursuivre est notée $y_d(t)$ et la période de discrétisation est notée T_d .

Le vecteur erreur en sortie $y(kT_d) - y_d(kT_d)$ est noté $\mathcal{E}(kT_d) = (\mathcal{E}_1, \mathcal{E}_2, \dots, \mathcal{E}_s)^T$. De la même manière que dans le deuxième chapitre, nous allons construire une trajectoire de référence $y_r(t)$, que nous allons utiliser pour que le système converge vers $y_d(t)$.

On considère $e = (e_1, \dots, e_s)^T$ un vecteur constant prédéfini et \mathcal{D}_i , $i=1, \dots, s$ des domaines bornés autour de y_{d_i} définis par :

$$(y_i \in \mathcal{D}_i) \text{ ssi } (\forall i \in \{1, \dots, s\}, |y_i - y_{d_i}| \leq e_i) \quad (3.16)$$

Ceci peut être schématisé par (fig.3.7) :

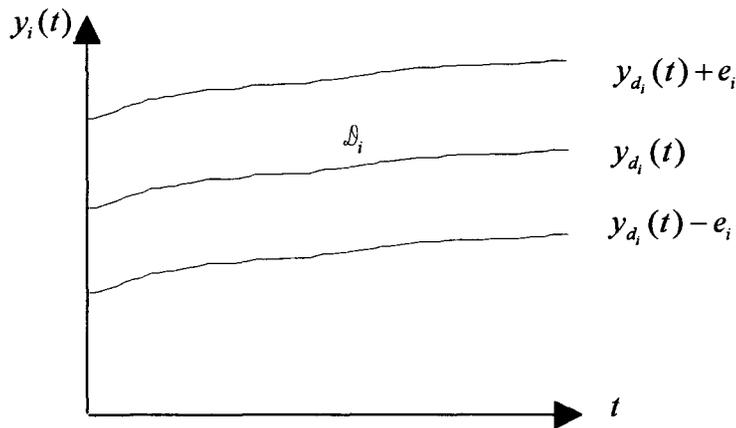


Fig.3.7 : Domaines autour de la consigne.

Algorithme :

1) Détermination de la trajectoire de référence :

Si, pour tout $i \in \{1, \dots, s\}$, $\varepsilon_i((k-1)T_d) \leq e_i((k-1)T_d)$, alors :

$$y_r(kT_d) = y_d(kT_d)$$

(commande en boucle ouverte)

sinon :

$$y_r(kT_d) = 2y_d(kT_d) - y(kT_d)$$

(point symétrique de y par rapport à y_d) (3.17)

2) Calcul des coefficients $\alpha_j(kT_d)$ à partir de l'équation (Chap.II : 2.58) que nous rappelons :

$$\left\{ \begin{array}{l} y_r(kT_d) = \sum_{j=1}^{2^m} \alpha_j(k) y_{j\infty} \\ \sum_{j=1}^{2^m} \alpha_j(k) = 1 \\ \alpha_j(k) \in [0,1], j = 1, \dots, 2^m \end{array} \right. \quad (3.18)$$

La figure suivante donne un exemple de construction d'une trajectoire de référence (fig.3.8).

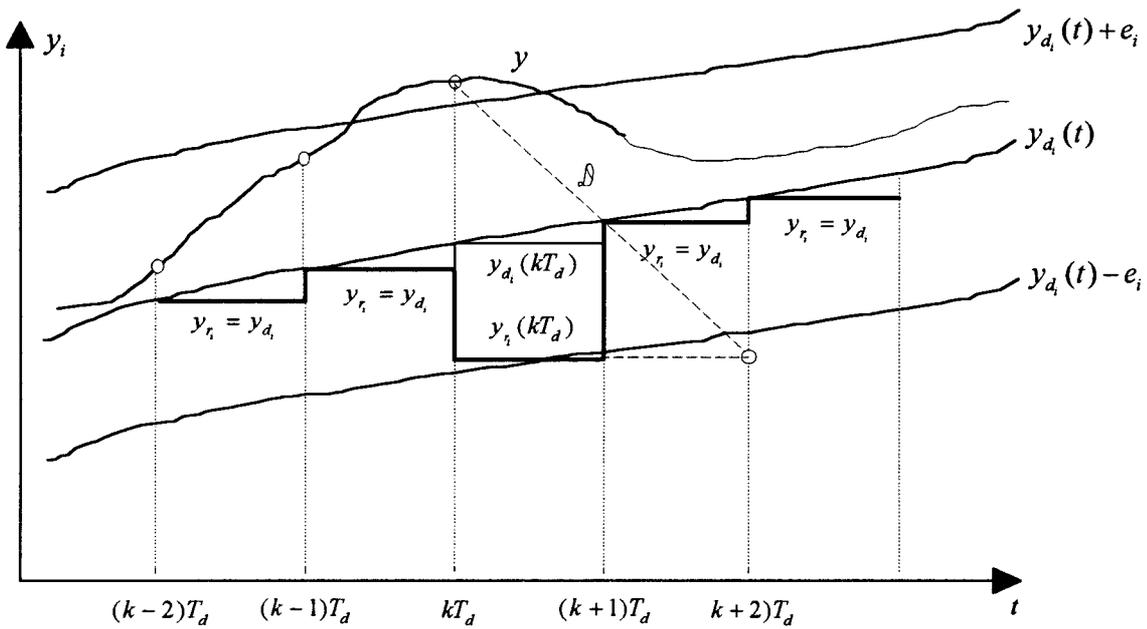


Fig.3.8 : Construction d'une trajectoire de référence.

L'algorithme précédent peut être représenté sous forme de schéma-bloc (fig.3.9) :

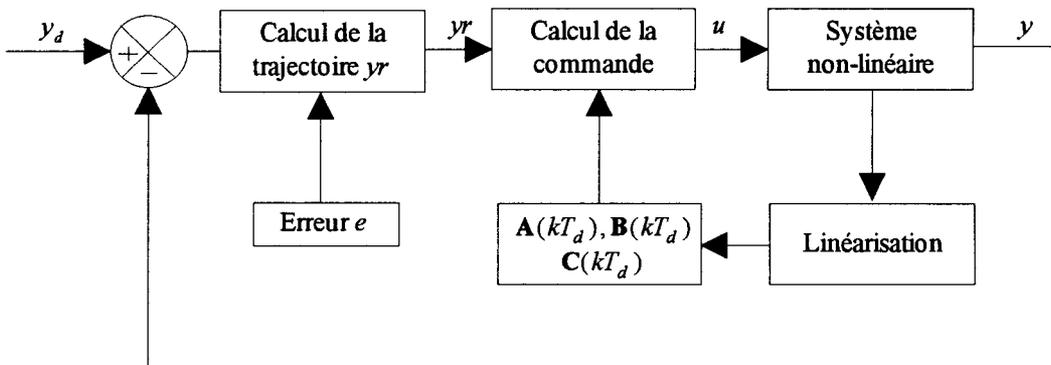


Fig.3.9 : Schéma-bloc de l'algorithme.

Le bloc "calcul de la trajectoire" détermine $y_r(kT_d)$ à partir de $e(kT_d)$ et e .

Le bloc "calcul de la commande" calcule les coefficients $\alpha_i(kT_d)$ à partir des matrices $\mathbf{A}(kT_d)$, $\mathbf{B}(kT_d)$ et $\mathbf{C}(kT_d)$ et $y_r(kT_d)$ en utilisant la formule (3.18) ce qui permet d'élaborer la séquence du vecteur u .

III.2.3.2 Convergence de l'algorithme

Plusieurs hypothèses sont nécessaires pour pouvoir implanter l'algorithme. Le système non-linéaire (Σ) doit être linéarisable par morceaux. De plus, la dynamique du système doit être

plus rapide que les variations de la trajectoire à poursuivre. Il faut donc que l'on puisse choisir $T_d \gg T_{mod}$.

Ces hypothèses étant satisfaites, la démonstration de la convergence de l'algorithme ne peut pas être effectuée dans le cas général. Il faut, au cas par cas, démontrer la convergence en utilisant entre autres des méthodes de transformation de matrices pour mettre en évidence les termes dominants (majoration, changements de coordonnées, méthode des pôles dominants), ce qui peut s'avérer complexe dans le cas de systèmes dont les dimensions sont élevées.

III.2.4 Exemple d'application

Considérons le système suivant :

$$\begin{pmatrix} \dot{x}_1 \\ \dot{x}_2 \end{pmatrix} = \begin{pmatrix} -x_1 x_2 \\ x_1 \end{pmatrix} + \begin{pmatrix} 1 & 2 \\ 0 & 1 \end{pmatrix} u$$

On linéarise sur $[kT_d, (k+1)T_d]$:

$$\begin{cases} \dot{x} = \begin{pmatrix} 0 & -x_1(kT_d) \\ 1 & 0 \end{pmatrix} x + \begin{pmatrix} 1 & 2 \\ 0 & 1 \end{pmatrix} u \\ y = x \end{cases}$$

On peut alors calculer les points-limites :

$$\begin{cases} y_{1\infty}(kT_d) = -\mathbf{A}^{-1}(kT_d)\mathbf{B} \begin{pmatrix} 0 \\ 0 \end{pmatrix} = \begin{pmatrix} 0 \\ 0 \end{pmatrix} \\ y_{2\infty}(kT_d) = -\mathbf{A}^{-1}(kT_d)\mathbf{B} \begin{pmatrix} 1 \\ 0 \end{pmatrix} = \begin{pmatrix} 0 \\ 1 \\ x_1(kT_d) \end{pmatrix} \\ y_{3\infty}(kT_d) = -\mathbf{A}^{-1}(kT_d)\mathbf{B} \begin{pmatrix} 0 \\ 1 \end{pmatrix} = \begin{pmatrix} -1 \\ 2 \\ x_1(kT_d) \end{pmatrix} \\ y_{4\infty}(kT_d) = -\mathbf{A}^{-1}(kT_d)\mathbf{B} \begin{pmatrix} 1 \\ 1 \end{pmatrix} = \begin{pmatrix} -1 \\ 3 \\ x_1(kT_d) \end{pmatrix} \end{cases}$$

On voit que la première composante des $y_{i\infty}(kT_d)$ est indépendante de kT_d , ce qui entraîne que $x_1(kT_d)$ est compris entre 0 et -1.

On désire faire passer le système d'une condition initiale $x(0)=(0,0)^T$ à $y_d=x_d=(-1/2,-3)^T$.

On va prendre un seul système linéarisé dans tout le domaine, en choisissant $x_1(kT_d)=-1/2$, et nous allons appliquer pour cet exemple l'algorithme en boucle ouverte.

Le calcul des coefficients a_i , pour $i=1,2,3,4$ peut être effectué en résolvant :

$$\begin{cases} a_3 + a_4 = \frac{1}{2} \\ a_2 + 2a_3 + 3a_4 = \frac{3}{2} \\ a_1 = 1 - a_2 - a_3 - a_4 \end{cases}$$

Ce qui donne en fixant arbitrairement $a_3=a_4$:

$$a_1 = a_2 = a_3 = a_4 = \frac{1}{4}$$

L'axe des temps est découpé en intervalles de longueur fixe notée $T_{mod}=10^{-3}s$. L'allure des entrées que l'on va appliquer au système peut être alors représenté (fig.3.10) :

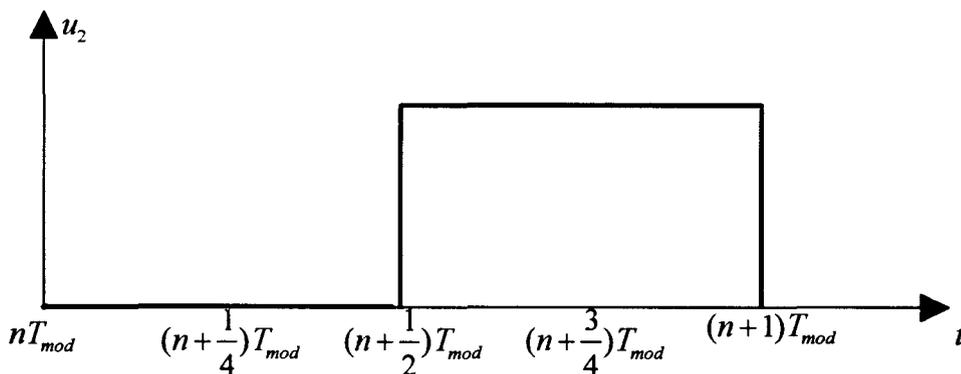
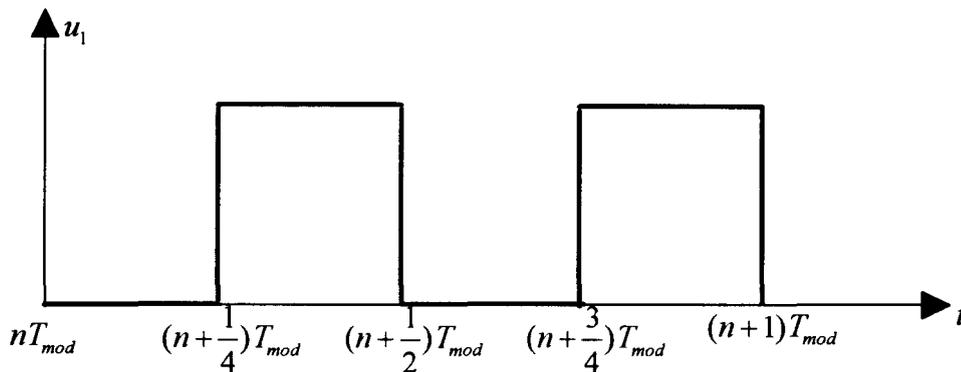


Fig.3.10 : Représentation temporelle des entrées.

Si on applique cet algorithme au système non-linéaire, nous obtenons les résultats de simulation suivants (fig.3.11) :

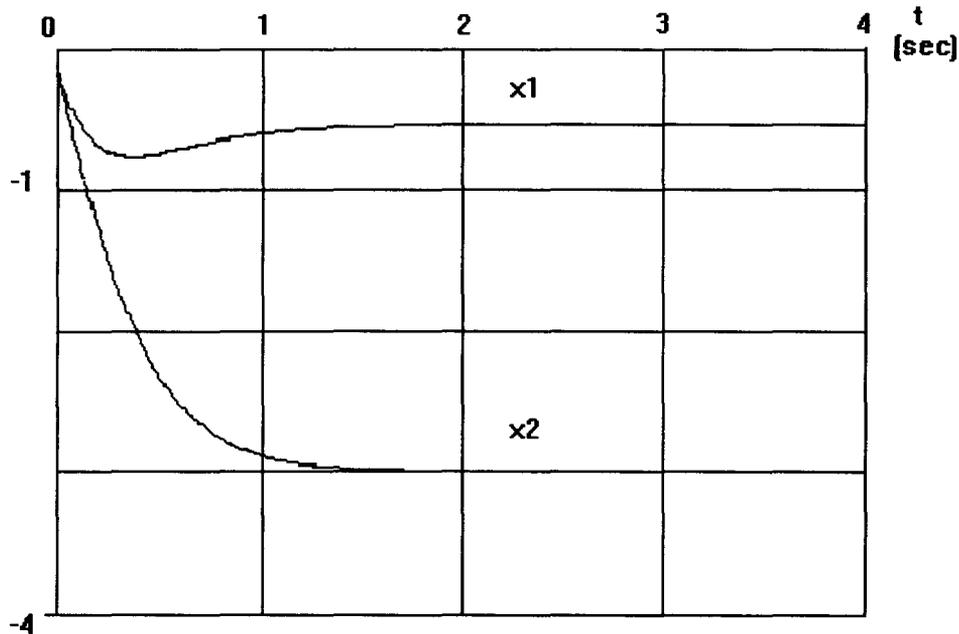


Fig.3.11 : Réponse temporelle de l'état du système non-linéaire.

III.3 SYSTEMES NON-LINEAIRES : LINEARISATION PAR RETOUR D'ETAT

III.3.1 Position du problème

L'objet de l'étude qui va suivre est d'élaborer une loi de commande en sortie pour certaines classes de systèmes non-linéaires à commandes booléennes. Pour plus de simplicité, nous nous limiterons à l'étude des systèmes non-linéaires en l'état et affines en la commande.

L'élaboration de l'algorithme de commande peut être détaillée comme suit : dans un premier temps, nous élaborerons une loi de commande par une linéarisation par retour d'état en construisant un système à commandes continues associé au modèle à commandes booléennes. La deuxième étape sera de borner les composantes de l'entrée dans l'intervalle $[0,1]$ puis la commande booléenne sera déterminée par binarisation de la commande du système continu associé.

Le modèle du système que nous allons étudier peut s'exprimer sous la forme:

$$\begin{cases} \frac{dx}{dt} = f(x) + g(x)u \\ y = h(x) \end{cases} \quad (3.19)$$

Les vecteurs x , y et u sont définis de la même manière que dans le deuxième chapitre. h est un vecteur de \mathbf{R}^s dont les composantes dépendent de x . $f(x)$ et $g(x)$ sont des matrices dont les coefficients peuvent dépendre de l'état x appartenant respectivement à \mathbf{R}^n et $\mathbf{R}^n \times \mathbf{R}^m$. Nous supposons en outre que pour tout $i=1, \dots, m$, les coefficients u_i sont booléens.

Le but de cette partie va être d'élaborer une loi de commande pour réguler le système puis pour faire du suivi de trajectoire. Les résultats obtenus seront ensuite appliqués sur plusieurs exemples en simulation [Abadie et al, 1993(a)].

III.3.2 Loi de commande continue

III.3.2.1 Commandes à valeurs dans \mathbf{R}^m

Le but est tout d'abord de déterminer une loi de commande idéale, en faisant l'hypothèse que l'entrée est une variable continue à valeurs dans \mathbf{R}^m . Dans le paragraphe suivant, une binarisation de la loi de commande sera envisagée.

Considérons $u_c = (u_{c_1}, \dots, u_{c_m})^T$ un vecteur dont les composantes sont continues. Le système à entrées continues associé à l'équation (3.19) peut être exprimé, en boucle ouverte par :

$$\begin{cases} \frac{dx}{dt} = f(x) + g(x)u_c \\ y = h(x) \end{cases} \quad (3.20)$$

Pour élaborer une loi de commande pour le système représenté par l'équation (3.20) nous allons utiliser une méthode de découplage-linéarisation entrées-sorties par difféomorphisme et bouclage des systèmes non-linéaires affines.

Pour plus de détails sur cette méthode le lecteur pourra consulter les ouvrages [Isidori, 1989], [Bournonville, 1984] et [Descusse et al, 1993].

III.3.2.1.1 Systèmes carrés

Nous supposons dans ce paragraphe que le nombre d'entrées est égal au nombre de sorties ($s=m$). Soient (r_1, \dots, r_m) les degrés relatifs du système (3.20) définis par :

$$r_i = \inf_{l \in \mathbb{N}} (l) \text{ tq } \exists j \in \{1, \dots, m\}, L_{g_j} L_f^{l-1} h_i \neq 0$$

où $L_g h$ désigne la dérivée de Lie de h dans la direction de champs de vecteurs f , c'est à dire :

$$L_f h(x) = \langle dh, f \rangle = \sum_{j=1}^n \frac{\partial h}{\partial x_j} f_j(x) \quad (3.21)$$

Cette opération peut être réitérée, ce qui donne pour les dérivées successives :

$$L_f^k h(x) = \sum_{j=1}^n \frac{\partial^k h}{\partial x_j} f_j(x) \quad (3.22)$$

avec $L_f^0 h = h$.

Les degrés relatifs correspondent à l'ordre de la première dérivée des composantes de la sortie qui fait apparaître l'entrées u_c , soit :

$$y_i^{(r_i)} = L_f^{r_i} h_i(x) + L_g L_f^{r_i-1} h_i(x) u_c \quad (3.23)$$

On définit alors la matrice de découplage $\Delta(x) \in \mathbb{R}^m \times \mathbb{R}^m$ par :

$$\Delta(x) = \begin{pmatrix} L_{g_1} L_f^{r_1-1} h_1(x) & \dots & L_{g_m} L_f^{r_1-1} h_1(x) \\ \vdots & & \vdots \\ L_{g_1} L_f^{r_m-1} h_m(x) & \dots & L_{g_m} L_f^{r_m-1} h_m(x) \end{pmatrix} \quad (3.24)$$

et le vecteur $\Delta_0(x) = (L_f^{r_1} h_1(x), \dots, L_f^{r_m} h_m(x))^T$.

Le système (3.20) est découplable statiquement si et seulement si [Isidori, 1989] :

$$\text{rang}(\Delta(x)) = m \quad (3.25)$$

Si cette condition est vérifiée, on définit le retour d'état par :

$$u_c = -\Delta^{-1}(x)\Delta_0(x) + \Delta^{-1}(x)v \quad (3.26)$$

Le comportement entrées-sorties du système bouclé par ce retour d'état est alors linéaire et peut être défini par :

$$\begin{cases} y_i^{(r_i)} = v_i \\ i = 1, \dots, m \end{cases} \quad (3.27)$$

a) Cas où la somme des degrés relatifs vaut n :

Si on a $\sum_{i=1}^m r_i = n$, il existe alors un changement de coordonnées pour l'état $\phi : x \rightarrow z$ tel que :

$$\begin{cases} \phi = (\phi_1^1, \dots, \phi_{r_1}^1, \dots, \phi_1^m, \dots, \phi_{r_m}^m)^T \\ \text{avec } \phi_i^k(x) = L_f^{k-1} h_i(x), i \in \{1, \dots, m\}, k \in \{1, \dots, r_i\} \end{cases} \quad (3.28)$$

ϕ ainsi défini est un difféomorphisme et le système bouclé est linéaire, commandable et observable.

b) Cas où la somme des degrés relatifs est inférieure à n :

Soit $r = \sum_{i=1}^m r_i$. L'ensemble des $\phi_i^k(x)$ doit être complété par les fonctions arbitraires $\phi_{r+1}, \dots, \phi_n$ telles que $\phi = (\phi_1^1, \dots, \phi_{r_1}^1, \dots, \phi_1^m, \dots, \phi_{r_m}^m, \phi_{r+1}, \dots, \phi_n)^T$ ait un rang égal à n . Après ce changement de coordonnées, il reste une partie non-linéaire dans l'équation d'état appelée le système des zéros dynamiques sans action sur la sortie, mais dont il faut étudier la stabilité.

III.3.2.1.2 Systèmes non carrés

On suppose $s < m$ et $\Delta(x)$ est définie par :

$$\Delta(x) = \begin{pmatrix} L_{g_1} L_f^{r_1-1} h_1(x) & \dots & L_{g_m} L_f^{r_1-1} h_1(x) \\ \vdots & & \vdots \\ L_{g_1} L_f^{r_m-1} h_m(x) & \dots & L_{g_m} L_f^{r_m-1} h_m(x) \end{pmatrix} \quad (3.29)$$

Si $\text{rang}(\Delta(x))=s$, le système défini par (3.20) est découplable statiquement par le retour d'état $u_c = -\Delta^*(x)\Delta_0(x) + \Delta^*(x)v$, $\Delta^*(x)$ désignant la pseudo-inverse de $\Delta(x)$.

Le comportement entrées-sorties du système bouclé est linéaire et décrit par :

$$\begin{cases} y_i^{(\eta_i)} = v_i \\ i = 1, \dots, s \end{cases} \quad (3.30)$$

De la même manière que pour les systèmes carrés, si la somme des degrés relatifs vaut n le système bouclé sera linéaire, commandable et observable et si $r < n$, il sera nécessaire d'étudier la dynamique des zéros.

III.3.2.1.3 Dynamique de l'erreur

On souhaite que la sortie converge vers une consigne $y_d(t) = (y_{d_1}(t), \dots, y_{d_s}(t))^T$. Il faut alors imposer, pour $i=1, \dots, s$:

$$v_i = v_i(y_i, y_{d_i}) = y_{d_i}^{(\eta_i)} + \sum_{j=0}^{\eta_i-1} \lambda_{i,j} (y_i^{(j)} - y_{d_i}^{(j)}) \quad (3.31)$$

avec $\lambda_{i,j}$ scalaires tels que les polynômes $X^{\eta_i} - \sum_{j=0}^{\eta_i-1} \lambda_{i,j} X^j$ aient leurs racines à parties réelles négatives.

III.3.3 Commandes à valeurs dans $[0,1]^m$

Nous avons supposé dans ce qui précède que les composantes de la commande u étaient à valeurs dans \mathbf{R} . Pour pouvoir appliquer cet algorithme au système décrit par (3.19), la seconde étape est de déterminer des composantes de l'entrée appartenant à l'intervalle $[0,1]$ à partir des composantes continues obtenues précédemment.

III.3.3.1 Position du problème

III.3.3.1.1 Exemple introductif : cas scalaire

Nous allons envisager pour permettre de mieux appréhender le problème, le cas scalaire, avec y, x , et u_c appartenant à \mathbf{R} . On supposera en outre que $y=cx$ et que les scalaires c

et $g(x)$ sont non-nuls. On peut donc déduire de ce qui précède que le problème de la linéarisation sera résolu si on choisit u_c tel que :

$$u_c = \frac{\dot{y} - cf(x)}{cg(x)} \quad (3.32)$$

On choisit une dynamique du premier ordre, telle que :

$$\frac{dy}{dt} = \frac{dy_d}{dt} - k_p(y - y_d) \quad (3.33)$$

k_p supposé positif. L'équation d'état en boucle fermée est définie par :

$$\dot{x} = -k_p x + \frac{1}{c} \frac{dy_d}{dt} + \frac{k_p}{c} y_d \quad (3.34)$$

dont la solution est :

$$x(t) = \left(x(t_0) - \frac{y_d(t_0)}{c} \right) e^{-k_p(t-t_0)} + \frac{y_d(t)}{c} \quad (3.35)$$

On souhaite que $u_c \in [0, 1]$, ce qui entraîne :

$$0 \leq \frac{\left[\frac{dy_d}{dt} - k_p(cx - y_d) - cf(x) \right]}{cg(x)} \leq 1 \quad (3.36)$$

avec $x(t)$ définie par l'équation (3.35).

Il faut donc ajuster k_p pour que les inégalités de l'équation (3.36) soient vérifiées.

III.3.3.1.2 Adaptation de la dynamique

La contrainte (3.36) peut ne pas être satisfaite pour toutes les valeurs de t . On peut donc envisager de choisir k_p constant par morceaux pour pouvoir adapter la dynamique du système permettant de satisfaire la contrainte $u_c \in [0, 1]$.

On doit alors, sur chaque intervalle $[nT_k, (n+1)T_k]$ choisir $k_p = k_p(nT_k)$ tel que :

$$0 \leq \frac{\left[\frac{dy_d}{dt} - k_p(nT_k)(cx - y_d) - cf(x) \right]}{cg(x)} \leq 1 \quad (3.37)$$

III.3.3.1.3 Exemples d'application

Considérons le système décrit par l'équation d'état :

$$\begin{cases} \dot{x} = -x^2 + u \\ y = x \end{cases}$$

On désire poursuivre la trajectoire $y_d(t) = \frac{1-e^{-t}}{2}$. Il faut donc choisir pour effectuer la linéarisation par retour d'état :

$$u_c = \frac{dy_d}{dt} - k_p(x - y_d) + x^2 = \frac{1}{2}e^{-t} - k_p\left(x - \frac{1-e^{-t}}{2}\right) + x^2$$

Le système en boucle fermée est régi par :

$$\dot{x} = -k_p x + \frac{1}{2}e^{-t} + \frac{k_p}{2}(1 - e^{-t})$$

qui entraîne :

$$x(t) = x(0)e^{-k_p t} + \frac{1 - e^{-t}}{2}$$

La condition (3.36) s'écrit :

$$0 \leq \frac{1}{2}e^{-t} - k_p x(0)e^{-k_p t} + \left(x(0)e^{-k_p t} + \frac{1 - e^{-t}}{2}\right)^2 \leq 1$$

Cette condition dépend des conditions initiales, et par conséquent le choix de k_p aussi.

Si on a $x(t_0)=0.5$ par exemple, une représentation graphique (fig.3.12) de $u_c(t)$ nous permet alors de déterminer les valeurs possibles du paramètre k_p :

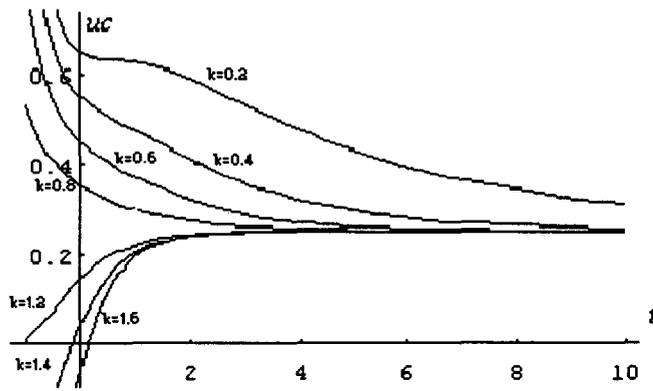


Fig.3.12 : Représentation de u_c pour quelques valeurs de k_p .

On voit que les valeurs de k_p entre 0.2 et 1.4 conviennent et que 1.6 ne convient pas.

Un autre exemple nous montre qu'il n'est pas toujours possible que la commande u_c appartienne à $[0,1]$:

Considérons le même système monovarié que précédemment :

$$\begin{cases} \dot{x} = -x^2 + u \\ y = x \end{cases}$$

pour lequel on modifie la trajectoire désirée, par exemple $y_d=2$, il vient :

$$u_c = -k_p(x-2) + x^2$$

ce qui entraîne :

$$u_c(t) = -k_p((x(0)-2)e^{-k_p t}) + (x(0) + 2(1 - e^{-k_p t}))^2$$

On a donc :

$$\lim_{t \rightarrow \infty} (u_c(t)) = 4$$

et comme u_c est une fonction continue, il existe t_1 tel que pour tout $t > t_1$ l'inégalité $u_c(t) < 1$ est impossible à réaliser, quelle que soit la valeur de k_p .

III.3.3.2 Cas général

La dynamique de l'erreur étant imposée, la commande u_c est définie par :

$$\begin{cases} u_c = -\Delta'(x)\Delta_0(x) + \Delta'(x)v \\ v_i = v_i(y_i, y_{d_i}) = y_{d_i}^{(r_i)} + \sum_{j=0}^{r_i-1} \lambda_{i,j} (y_i^{(j)} - y_{d_i}^{(j)}) \\ \Delta'(x) = \Delta^{-1}(x) \text{ si } s = m \\ \Delta'(x) = \Delta^*(x) \text{ si } s < m \end{cases} \quad (3.38)$$

Ce système nous permet de déduire les composantes de u_c en fonction de l'état x , des dérivées successives des sorties et des consignes .

Il faut adapter les coefficients $\lambda_{i,j}$ pour que les composantes de u_c soient à valeurs dans $[0,1]$. Cette opération est plus difficile que dans le cas scalaire car on ne peut pas toujours exprimer x en fonction de t .

L'équation d'état en boucle fermée est donnée par :

$$\begin{cases} \dot{x} = f(x) + g(x)(-\Delta'(x)\Delta_0(x) + \Delta'(x)v) \\ y_i^{(r_i)} = v_i \\ v_i = v_i(y_i, y_{d_i}) = y_{d_i}^{(r_i)} + \sum_{j=0}^{r_i-1} \lambda_{i,j} (y_i^{(j)} - y_{d_i}^{(j)}) \end{cases} \quad (3.39)$$

III.3.3.2.1 Expression de $u_c(t)$

a) Cas où l'on peut exprimer $x(t)$

Si on peut résoudre l'équation (3.39), alors on peut exprimer x en fonction de t et par conséquent $u_c(t)$ est calculable. Nous pouvons alors appliquer la même méthode que dans le cas scalaire pour adapter les coefficients $\lambda_{i,j}$ de manière à ce que les composantes de u_c soient à valeurs dans $[0,1]$.

Remarque :

Si $s=n$ et $h^{-1}(y)$ existe, alors x est exprimable en fonction de t par $x(t)=h^{-1}(y(t))$ et par conséquent u_c aussi.

b) Cas où l'on ne peut pas exprimer directement $x(t)$

Si on ne peut pas résoudre l'équation (3.39), alors on ne peut pas exprimer $u_{c_i}(t)$ de manière exacte. La méthode que l'on peut alors utiliser est une technique numérique qui consiste à simuler le comportement du système continu (3.39) pour différentes valeurs des paramètres $\lambda_{i,j}$ et de retenir les valeurs des paramètres pour lesquelles les composantes de u_c sont dans $[0,1]$.

III.3.3.2.2 Exemples d'application

Exemple 1 : $x(t)$ directement exprimable

On considère le système décrit par :

$$\begin{cases} \dot{x}_1 = -x_2 + u_{c_1} \\ \dot{x}_2 = -2x_1^2 - u_1 + 4u_{c_2} \\ y = x = (x_1, x_2)^T \end{cases}$$

La linéarisation du système continu associé est obtenue par le bouclage suivant :

$$\begin{cases} u_{c_1} = \frac{dy_{d_1}}{dt} - k_{p_1}(y_1 - y_{d_1}) + x_2 \\ u_{c_2} = \frac{1}{4} \left(\frac{dy_{d_2}}{dt} - k_{p_2}(y_2 - y_{d_2}) + u_{c_1} + 2x_1^2 \right) \end{cases}$$

On choisit : $y_{d_1} = 0.5$ et $y_{d_2} = 0.5(1 - e^{-t})$:

La solution de l'équation d'état en boucle fermée s'écrit :

$$\begin{cases} x_1(t) = 0.5 + x_1(0)e^{-k_{p_1}t} \\ x_2(t) = x_2(0)e^{-k_{p_2}t} + 0.5(1 - e^{-t}) \end{cases}$$

Soit :

$$\begin{cases} u_{c_1} = -x_1(0)k_{p_1}e^{-k_{p_1}t} + x_2(0)e^{-k_{p_2}t} + 0.5(1 - e^{-t}) \\ u_{c_2} = \frac{1}{4}(u_{c_1} + 2(0.5 + x_1(0)e^{-k_{p_1}t})^2 + 0.5e^{-t} - k_{p_2}x_2(0)e^{-k_{p_2}t}) \end{cases}$$

On trace u_{c_1} et u_{c_2} pour quelques valeurs des paramètres k_{p_1} et k_{p_2} , avec $x_1(0)=0.5$, $x_2(0)=1$ et $k_{p_1} = k_{p_2} = k$ (fig.3.13, 3.14) :

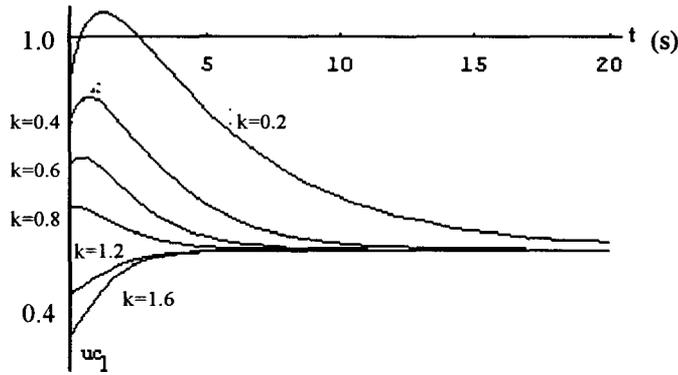


Fig.3.13 : Commande u_{c_1} en fonction des paramètres.

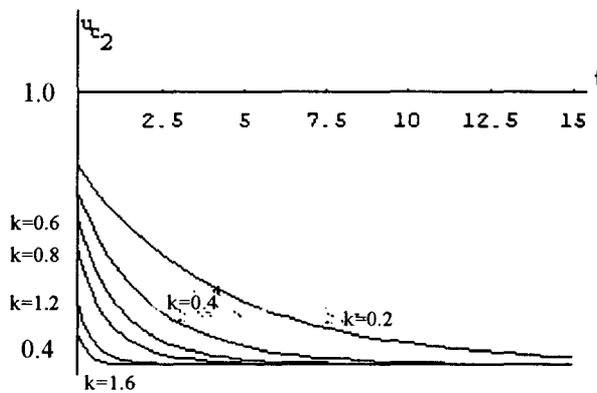


Fig.3.14 : Commande u_{c_2} en fonction des paramètres.

On voit que par exemple, les valeurs de k_{p_1} et k_{p_2} telles que $k_{p_1}=k_{p_2}$ comprises entre 0.6 et 1.4 conviennent. alors que $k_{p_1}=k_{p_2}=0.2$ ne convient pas.

Exemple 2 : $x(t)$ exprimable par intégration

Considérons le système décrit par :

$$\begin{cases} \dot{x} = \begin{pmatrix} -x_3^2 \\ -x_1^2 \\ x_2 \end{pmatrix} + \begin{pmatrix} 5 & 0 \\ 0 & 3 \\ 0 & 0 \end{pmatrix} \begin{pmatrix} u_{c_1} \\ u_{c_2} \end{pmatrix} \\ y = \begin{pmatrix} 1 & 0 & 0 \\ 0 & 0 & 1 \end{pmatrix} x \end{cases}$$

Les indices caractéristiques du système sont 1 et 2. On souhaite faire converger le système vers $y_d=(1,2)^T$.

La linéarisation entrées-sorties est obtenue si :

$$\begin{pmatrix} u_{c_1} \\ u_{c_2} \end{pmatrix} = \begin{pmatrix} \frac{1}{5}(v_1 + x_3^2) \\ \frac{1}{3}(v_2 + x_1^2) \end{pmatrix} = \begin{pmatrix} \frac{1}{5}\left(\frac{dy_1}{dt} + x_3^2\right) \\ \frac{1}{3}\left(\frac{d^2y_2}{dt^2} + x_1^2\right) \end{pmatrix}$$

On choisit des dynamiques pour les sorties telles que :

$$\begin{cases} v_1 = \frac{dy_1}{dt} = -k(y_1 - 1) \\ v_2 = \frac{d^2y_2}{dt^2} = -\lambda \frac{dy_2}{dt} - \mu(y_2 - 2) \end{cases}$$

avec $k > 0$, $\lambda^2 - 4\mu > 0$ et $\mu > 0$.

On pose :

$$\rho_{1,2} = \frac{\lambda \pm \sqrt{\lambda^2 - 4\mu}}{2}, \quad A = \frac{1}{\rho_2 - \rho_1}(\rho_2 y_{20} + y'_{20} - 2\rho_2) \quad \text{et} \quad B = \frac{-1}{\rho_2 - \rho_1}(\rho_1 y_{20} + y'_{20} - 2\rho_1)$$

avec $y_{20} = x_{30}$, $y'_{20} = \dot{x}_{30} = x_{20}$.

y est alors exprimable en boucle fermée par :

$$\begin{cases} x_1(t) = y_1(t) = 1 + (y_{10} - 1)e^{-kt} \\ x_3(t) = y_2(t) = 2 + Ae^{-\rho_1 t} + Be^{-\rho_2 t} \end{cases}$$

ce qui donne pour v :

$$\begin{cases} v_1(t) = -k(y_{10} - 1)e^{-kt} \\ v_2(t) = (\lambda\rho_1 - \mu)Ae^{-\rho_1 t} + (\lambda\rho_2 - \mu)Be^{-\rho_2 t} \end{cases}$$

ce qui entraîne pour x_2 :

$$x_2(t) = \eta(e^{-\rho_1 t} - e^{-\rho_2 t}) + y'_{20}e^{-\rho_2 t}$$

avec $\eta = \frac{\mu - \lambda\rho_1}{\rho_1} A$.

On peut donc exprimer u_{c_1} et u_{c_2} en fonction de t :

$$\begin{cases} u_{c_1}(t) = \frac{1}{5}(-k_1(y_{10} - 1)e^{-k_1 t} + (2 + Ae^{-\rho_1 t} + Be^{-\rho_2 t})^2) \\ u_{c_2}(t) = \frac{1}{3}(A\rho_1^2 e^{-\rho_1 t} + B\rho_2^2 e^{-\rho_2 t} + (1 + (y_{10} - 1)e^{-k_1 t})^2) \end{cases}$$

Les composantes de u_c ont été simulées pour quelques valeurs des paramètres ρ_1, ρ_2 et k_1 (fig.3.15a,b) avec les conditions initiales $y_{10} = x_{10} = 1.5, y_{20} = x_{30} = 2, y'_{20} = x_{20} = 0.25$:

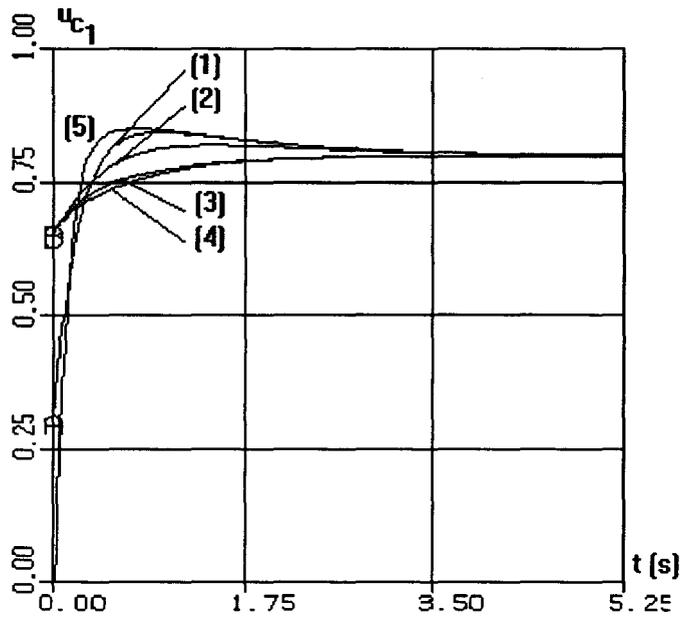


Fig.3.15a : Représentation temporelle de u_{c1} .

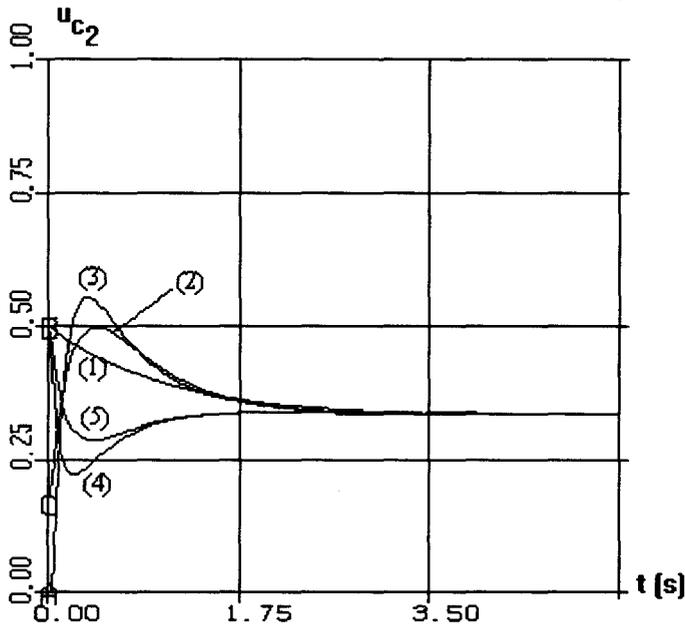


Fig.3.15b : Représentation temporelle de u_{c2} .

- avec :
- (1): $k_1 = 1.5, \rho_1 = 1, \rho_2 = 2$
 - (2): $k_1 = 1.5, \rho_1 = 3, \rho_2 = 4$
 - (3): $k_1 = 1.5, \rho_1 = 5, \rho_2 = 6$. Les configuration (1), (2) et (4) conviennent.
 - (4): $k_1 = 5, \rho_1 = 1, \rho_2 = 2$
 - (5): $k_1 = 10, \rho_1 = 1, \rho_2 = 2$

Exemple 3 : $x(t)$ non directement exprimable

Considérons le système défini par :

$$\begin{cases} \dot{x}_1 = -\alpha x_3 + u_{c_1} \\ \dot{x}_2 = -\beta x_1 + u_{c_2} \\ \dot{x}_3 = f(x_3) + \eta x_1 \\ y_1 = x_1 \\ y_2 = x_2 \end{cases}$$

pour lequel les indices caractéristiques sont égaux à 1.

Le retour linéarisant est donné par :

$$\begin{cases} u_{c_1} = v_1 + \alpha x_3 = \frac{dy_1}{dt} + \alpha x_3 \\ u_{c_2} = v_2 + \beta x_1 = \frac{dy_2}{dt} + \beta x_1 \end{cases}$$

On obtient en boucle fermée :

$$\begin{cases} \dot{x}_1 = v_1 \\ \dot{x}_2 = v_2 \\ \dot{x}_3 = f(x_3) + \eta x_1 \end{cases}$$

ce qui donne après résolution, pour une dynamique de l'erreur du premier ordre

$$\begin{cases} x_1(t) = y_{d_1}(t) + (y_{10} - y_{d_{10}})e^{-k_{p1}t} \\ x_2(t) = y_{d_2}(t) + (y_{20} - y_{d_{20}})e^{-k_{p2}t} \end{cases}$$

et x_3 est régie par : $\dot{x}_3 = f(x_3) + \eta (y_{d_1}(t) + (y_{10} - y_{d_{10}})e^{-k_{p1}t})$.

Cette dernière équation n'est pas soluble en général. u_c ne peut donc pas être exprimée en fonction de t . Si nous prenons, par exemple, $f(x_3)=\sin(x_3)$, $\alpha=1/5$, $\beta=1$, $\eta=1$, $x_{10}=x_{20}=1$ et $y_d=(1-e^{-t}, (1-e^{-t})/2)^T$, nous avons alors :

$$\dot{x}_3 = \sin(x_3) + 1 - e^{-t} + e^{-k_{p1}t}$$

La somme des degrés relatifs est strictement inférieure à $n=3$. Il est donc nécessaire d'étudier la stabilité des zéros dynamiques du systèmes, ce que nous ferons au §III.3.5.2. Nous avons simulé pour différentes valeurs de k_{p1} et k_{p2} (nous avons choisi la même dynamique pour les deux sorties soit $k_{p1}=k_{p2}=k$) le système suivant avec $x_{30}=1$:

$$\begin{cases} u_{c_1}(t) = e^{-t} - k_{p_1} e^{-k_{p_1} t} + \frac{1}{5} x_3 \\ u_{c_2}(t) = 1 - \frac{1}{2} e^{-t} - k_{p_2} e^{-k_{p_2} t} + e^{-k_{p_1} t} \\ \dot{x}_3 = \sin(x_3) + 1 - e^{-t} + e^{-k_{p_1} t} \end{cases}$$

Les entrées sont représentées fig.3.16a,b :

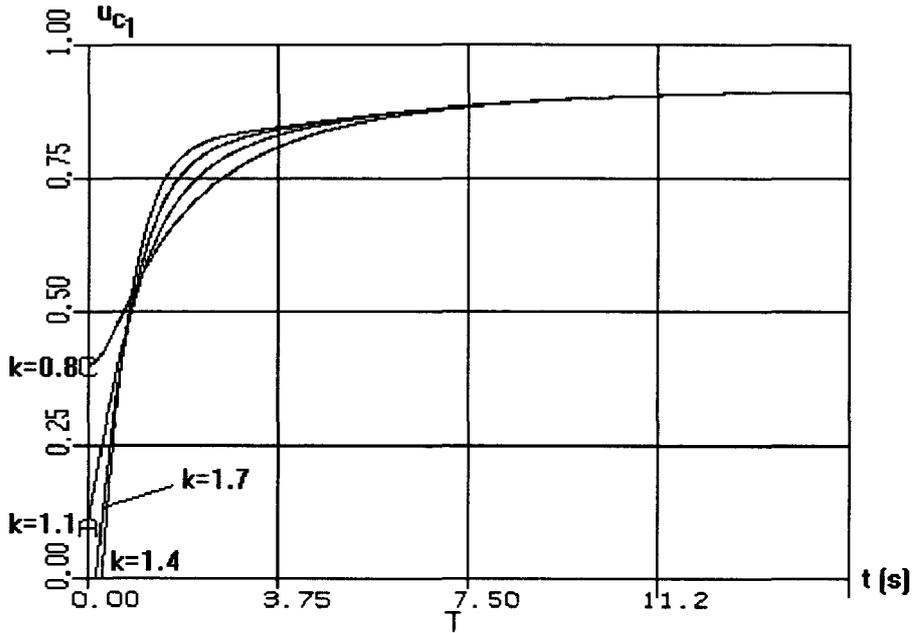


Fig.3.16a : Entrée u_{c_1} pour quelques valeurs de k .

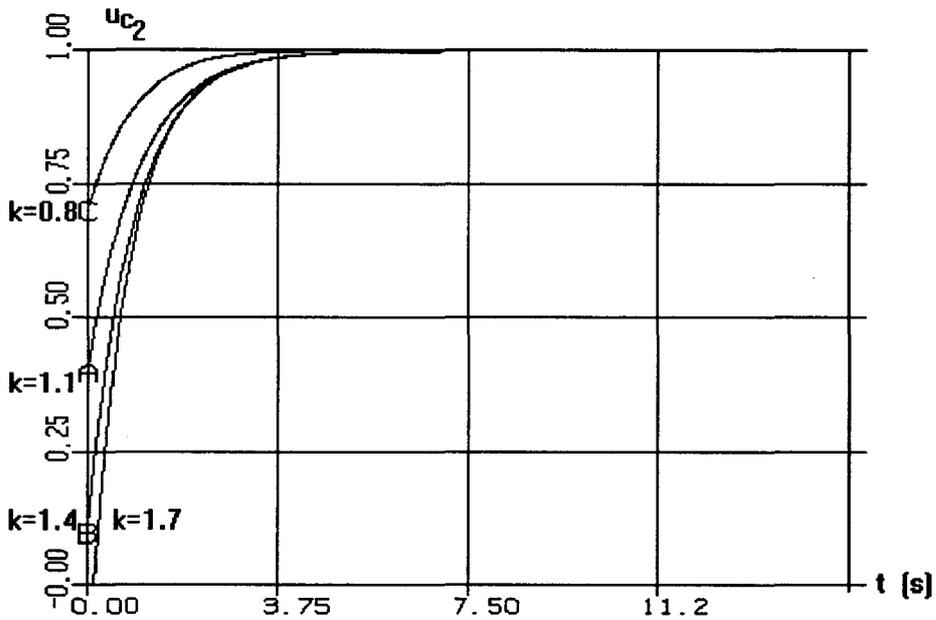


Fig.3.16b : Entrée u_{c_2} pour quelques valeurs de k .



On voit que les valeurs $k_{p1}=k_{p2}=0.8$, ou $k_{p1}=k_{p2}=1.1$ conviennent alors que les valeurs 1.4 et 1.7 ne conviennent pas.

III.3.4 Binarisation de la commande

Nous avons étudié dans ce qui précède une méthode permettant d'élaborer une loi de commande continue à valeurs dans $[0,1]^m$ pour le système (3.20). Le but est ici d'élaborer et de justifier une méthode permettant de construire une commande à valeurs dans $\{0,1\}^m$ à partir des grandeurs d'entrée continues pour pouvoir appliquer notre étude au système décrit par l'équation (3.19). Pour cela, nous allons binariser les composantes de la commande continue.

III.3.4.1 Valeur moyenne

Proposition 3.1

Soit ε un réel positif fixé et $\Phi(t)=(\Phi_1(t), \dots, \Phi_m(t))^T$ une fonction à valeurs dans $[0,1]^m$. Il est possible de construire la fonction $u(t)$ à composantes booléennes telle que, pour tout t réel, u et Φ aient la même valeur moyenne sur tout intervalle $]t, t+\varepsilon[$.

Preuve

Soient t et ε réels positifs. La valeur moyenne de $\Phi(t)$ sur $]t, t+\varepsilon[$ est notée $\hat{\Phi}(t)=(\hat{\Phi}_1(t), \dots, \hat{\Phi}_m(t))^T$. Si on construit u de sorte que :

$$\begin{cases} u_i = 1 \text{ sur }]t, t + \varepsilon \hat{\Phi}_i(t)[\\ u_i = 0 \text{ sur }]t + \varepsilon \hat{\Phi}_i(t), t + \varepsilon[\end{cases} \quad (3.40)$$

La valeur moyenne de u_i sur $]t, t+\varepsilon[$ est alors égale à :

$$\hat{u}_i(t) = \frac{1}{\varepsilon} \left(\int_t^{t+\varepsilon \hat{\Phi}_i(t)} 1 \cdot dt + \int_{t+\varepsilon \hat{\Phi}_i(t)}^{t+\varepsilon} 0 \cdot dt \right) \quad (3.41)$$

soit :

$$\hat{u}_i(t) = \hat{\Phi}_i(t)$$

pour $i=1, \dots, m$.

Par conséquent si u est construite de cette manière on a bien :

$$\hat{u}(t) = \hat{\Phi}(t)$$

ce qui démontre la proposition 3.1.

III.3.4.2 Commande booléenne

En utilisant ce qui précède, nous pouvons construire une commande booléenne à partir de la valeur moyenne de $u_c(t)$. Dans la pratique, l'axe des temps est discrétisé avec une période notée T_k et on note $\hat{u}_c(nT_k)$ sa valeur moyenne sur l'intervalle $[nT_k, (n+1)T_k[$, soit :

$$\hat{u}_c(nT_k) = \frac{1}{T_k} \int_{nT_k}^{(n+1)T_k} u_c(t) dt \quad (3.42)$$

On construit alors la commande u en appliquant la formule (3.40) à $\Phi(t) = u_c(t)$ avec $t=t_0+nT_k$ et $\varepsilon=T_k$.

On a alors :

$$\begin{cases} u_i = 1 \text{ sur }]nT_k, nT_k + T_k \hat{u}_c(nT_k)[\\ u_i = 0 \text{ sur }]nT_k + T_k \hat{u}_c(nT_k), (n+1)T_k[\end{cases} \quad (3.43)$$

qui entraîne, sur $[nT_k, (n+1)T_k[$:

$$\hat{u}(nT_k) = \hat{u}_c(nT_k) = \frac{1}{T_k} \int_{nT_k}^{(n+1)T_k} u_c(t) dt \quad (3.44)$$

La procédure de construction de la loi de commande peut être résumée par le schéma suivant (fig.3.17) :

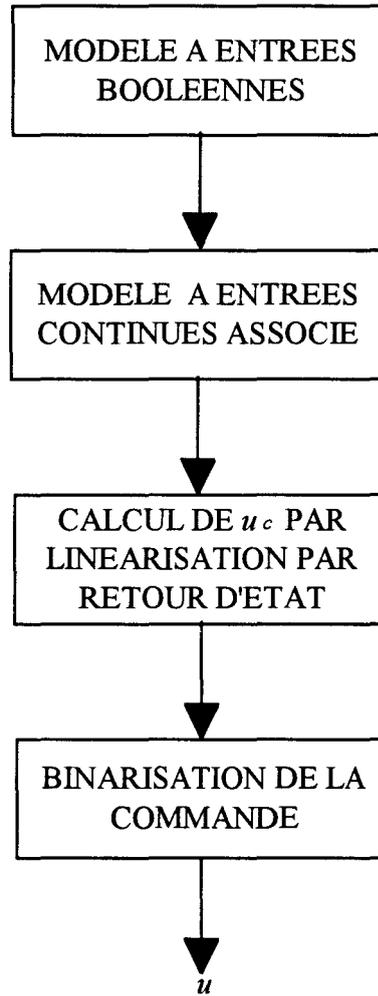


Fig.3.17 : Procédure de construction de l'algorithme de commande.

III.3.4.3 Convergence de l'algorithme

III.3.4.3.1 Cas scalaire

Nous supposons dans ce paragraphe que les systèmes étudiés (3.19) et (3.20) sont tels que x , y sont scalaires, u_c appartient à $[0,1]$ et u à $\{0,1\}$. Nous supposons en outre que $y=cx$, $c \neq 0$.

Si nous faisons apparaître la différence $u-u_c$ dans l'équation d'état, il vient :

$$\begin{cases} \dot{x} = f(x) + g(x)u_c + g(x)(u - u_c) \\ y = cx \end{cases} \quad (3.45)$$

En choisissant une dynamique de l'erreur du premier ordre, l'équation d'état en boucle fermée s'exprime :

$$\dot{x} = -k_p x + \frac{1}{c} \frac{dy_d}{dt} + \frac{k_p}{c} y_d + g(x)(u - u_c) \quad (3.46)$$

et peut être résolue :

$$x(t) = \left(x(0) - \frac{1}{c} y_d(0) \right) e^{-k_p t} + \frac{1}{c} y_d(t) + \int_0^t e^{k_p(\tau-t)} g(x(\tau))(u - u_c) d\tau$$

ou :

$$y(t) = (y(0) - y_d(0)) e^{-k_p t} + y_d(t) + c \int_0^t e^{k_p(\tau-t)} g(x(\tau))(u - u_c) d\tau$$

On veut que $y(t)$ poursuive $y_d(t)$, ce qui entraîne :

$$\lim_{t \rightarrow \infty} (y(t) - y_d(t)) = 0$$

Il faut donc démontrer que :

$$\lim_{t \rightarrow \infty} \left(\int_0^t e^{k_p(\tau-t)} g(x(\tau))(u - u_c) d\tau \right) = 0 \quad (3.47)$$

On peut montrer que, si $g(x)$ est bornée, cette limite est du même ordre que T_k (Annexe 7).

III.3.4.3.2 Cas multivariable

Pour plus de simplicité, nous nous restreindrons aux cas où les degrés relatifs valent 1 avec $s=n$. En procédant de la même manière que pour le cas monovarié, il vient :

$$\begin{cases} \dot{x} = f(x) + g(x)u_c + g(x)(u - u_c) \\ y = h(x) \end{cases} \quad (3.48)$$

on pose $w(x) = g(x)(u - u_c)$

Le système entrées-sorties en boucle fermée s'exprime alors :

$$\begin{cases} \dot{y}_i = L_f h_i(x) + L_w h_i(x) + L_g h_i(x) u_c = v_i + L_w h_i(x) \\ i = 1, \dots, s \end{cases} \quad (3.49)$$

Si on choisit v_i de sorte que le système continu converge vers une consigne $y_d(t)$ avec une dynamique de l'erreur du premier ordre, on obtient :

$$y_i(t) = (y_{i0} - y_{d_{i0}})e^{-k_i t} + y_{d_i}(t) + \underbrace{e^{-k_i t} \int_0^t e^{k_i \tau} L_w h_i(x(\tau)) d\tau}_{R_i} \quad (3.50)$$

Les termes résiduels R_i peuvent s'exprimer par :

$$R_i = \int_0^t e^{k_i(\tau-t)} L_w h_i(x(\tau)) d\tau = \int_0^t e^{k_i(\tau-t)} \sum_{j=1}^n \frac{\partial h_i}{\partial x_j} \sum_{k=1}^m g_{jk}(u_k - u_{c_k}) d\tau \quad (3.51)$$

Si les $\frac{\partial h_i}{\partial x_j}, (i, j) \in \{1, \dots, n\}^2$ sont bornés on montre, de la même manière que pour le cas scalaire, que si les $g_{jk}(i, k) \in \{1, \dots, n\} \times \{1, \dots, m\}$ sont bornées, les termes R_i sont du même ordre que T_k .

III.3.5 Mise en place de l'algorithme

La procédure de mise en place de la loi de commande peut être représentée (fig.3.18):

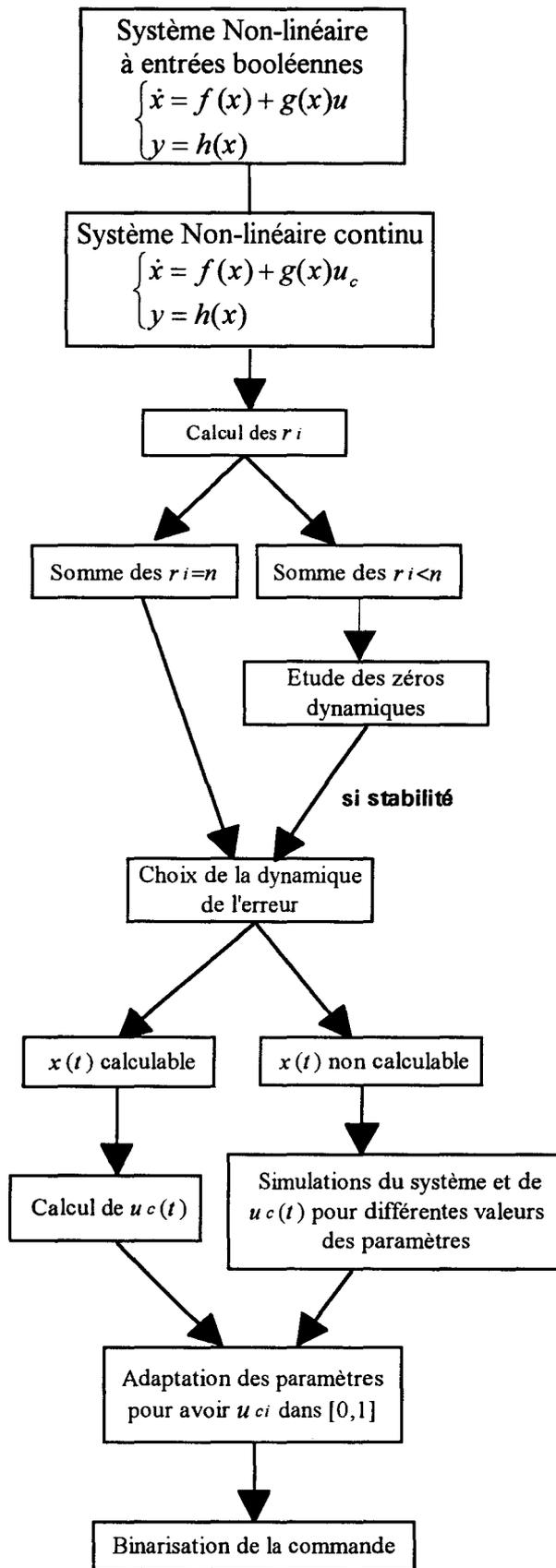


Fig.3.18 : Algorithme de commande.

Le système en boucle fermée peut être représenté par le schéma-bloc suivant (fig.3.19) :

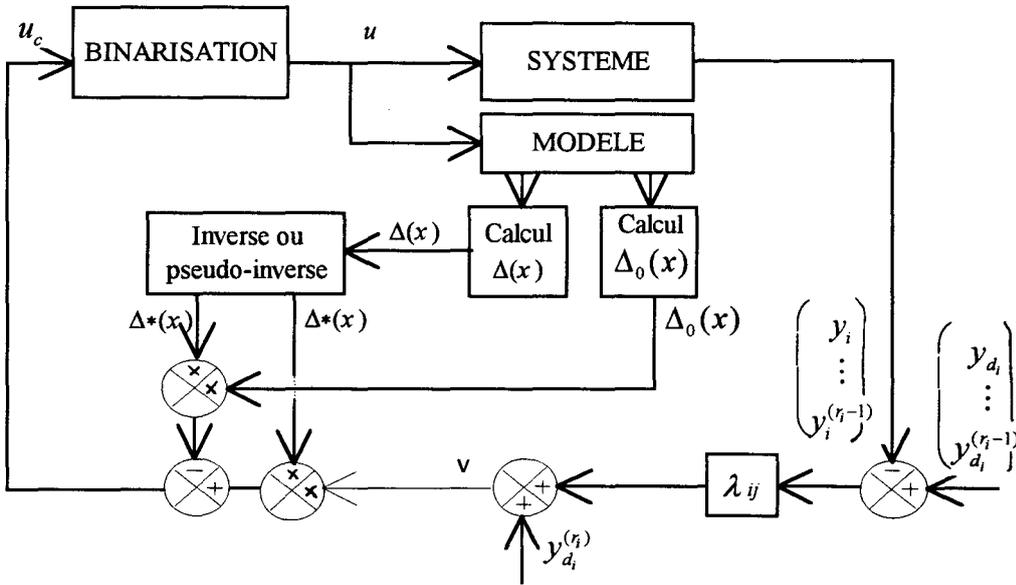


Fig.3.19 : Schéma-bloc du système en boucle fermée.

Remarques concernant l'implantation de l'algorithme :

Remarque 1 :

Si la dynamique des zéros est instable on ne peut pas utiliser cette technique de commande pour le système.

Remarque 2 :

Ces schéma-blocs nous montrent que, pour mettre en place cet algorithme, il est nécessaire que les fonctions $f(x)$ et $g(x)$ soient parfaitement connues et aussi que toutes les composantes du vecteur d'état soient mesurables. Si ce n'est pas le cas, il faut faire appel à des estimateurs ce qui soulève un certain nombre de problèmes dans le cas non-linéaire.

Remarque 3 :

L'inverse (ou pseudo-inverse) de la matrice de découplage peut ne pas exister en certains points. Il faut alors vérifier que l'état ne prend pas ces valeurs. De plus, la matrice Δ peut posséder elle-même certains points singuliers ce qui pose des problèmes car on ne dispose pas de techniques rigoureuses pour franchir ces singularités.

III.3.6 Exemple d'application

III.3.6.1 Exemple 1

Considérons le système décrit par :

$$\begin{cases} \dot{x}_1 \\ \dot{x}_2 \\ y = x \end{cases} = \begin{pmatrix} x_2^2 \\ -x_1 \end{pmatrix} + \begin{pmatrix} -8 & 0 \\ 0 & +2 \end{pmatrix} \begin{pmatrix} u_1 \\ u_2 \end{pmatrix}$$

avec u_1 et u_2 booléens.

Le système continu associé s'exprime :

$$\begin{cases} \dot{x}_1 \\ \dot{x}_2 \\ y = x \end{cases} = \begin{pmatrix} x_2^2 \\ -x_1 \end{pmatrix} + \begin{pmatrix} -8 & 0 \\ 0 & 2 \end{pmatrix} \begin{pmatrix} u_{c_1} \\ u_{c_2} \end{pmatrix}$$

avec $u_c = (u_{c_1}, u_{c_2})^T$ continue.

Les degrés relatifs des sorties sont égaux à 1, ce qui entraîne $\sum r_i = n = 2$ (pas de zéros dynamiques).

Le découplage est réalisé pour le système continu si :

$$\begin{pmatrix} u_{c_1} \\ u_{c_2} \end{pmatrix} = \begin{pmatrix} -1/8 & 0 \\ 0 & 1/2 \end{pmatrix} \left[\frac{dy_d}{dt} - k_p(y - y_d) - \begin{pmatrix} x_2^2 \\ x_1 \end{pmatrix} \right]$$

et la sortie est régie par :

$$\frac{dy}{dt} = \frac{dy_d}{dt} - \begin{pmatrix} k_{p_1} & 0 \\ 0 & k_{p_2} \end{pmatrix} (y - y_d)$$

Si on choisit :

$$y_d = \begin{pmatrix} 1 - e^{-t/2} \\ 2(1 - e^{-t/2}) \end{pmatrix}$$

La résolution de l'équation différentielle en boucle fermée donne :

$$\begin{pmatrix} x_1 \\ x_2 \end{pmatrix} = \begin{pmatrix} 1 - e^{-t/2} + e^{-k_{p_1}t} \\ 2(1 - e^{-t/2}) + 2e^{-k_{p_2}t} \end{pmatrix}$$

en choisissant $x_2(0)=2$ et $x_1(0)=1$.

On peut alors exprimer u_c en fonction de t :

$$\begin{pmatrix} u_{c_1} \\ u_{c_2} \end{pmatrix} = \begin{pmatrix} -\frac{1}{8}(-k_{p_1} e^{-k_{p_1} t} + \frac{1}{2} e^{-t/2} - (2e^{-k_{p_2} t} + 2(1 - e^{-t/2}))^2) \\ \frac{1}{2}(e^{-t/2} - 2k_{p_2} e^{-k_{p_2} t} + 1 - e^{-t/2} + e^{-k_{p_1} t}) \end{pmatrix}$$

Il faut que pour tout t , on ait $u_c \in [0,1]^2$. Si on trace les courbes représentant les composantes de u_c en fonction du temps avec k_{p_1} et k_{p_2} comme paramètres (fig.3.20a,b) :

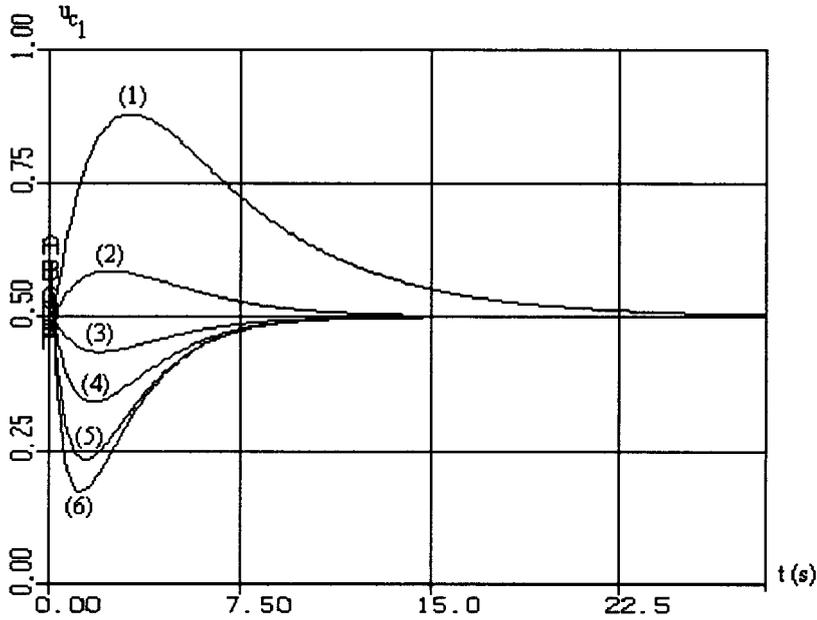


Fig.3.20a : Entrée u_{c_1} en fonction des paramètres.

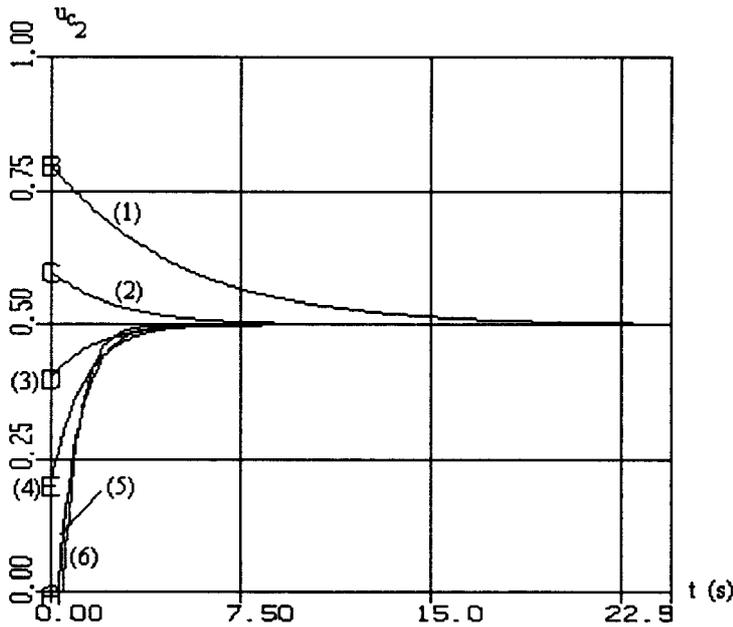


Fig.3.20b : Entrée u_{c_2} en fonction des paramètres.

avec : (1) : $k_{p_1}=k_{p_2}=0.2$; (2) : $k_{p_1}=k_{p_2}=0.4$; (3) : $k_{p_1}=k_{p_2}=0.6$; (4) : $k_{p_1}=k_{p_2}=0.8$; (5) : $k_{p_1}=k_{p_2}=1.2$; (6) : $k_{p_1}=k_{p_2}=1.6$.

On voit que les valeurs $k_{p_1}=k_{p_2}=0.8$ conviennent. Il faut maintenant binariser la commande continue que l'on vient de déterminer. Pour cela, on utilise la procédure décrite précédemment en découpant l'axe des temps avec une période $T_k=10^{-3}$ s.

Le système à entrées booléennes en boucle fermée a été simulé (fig3.21a,b,c,d) :

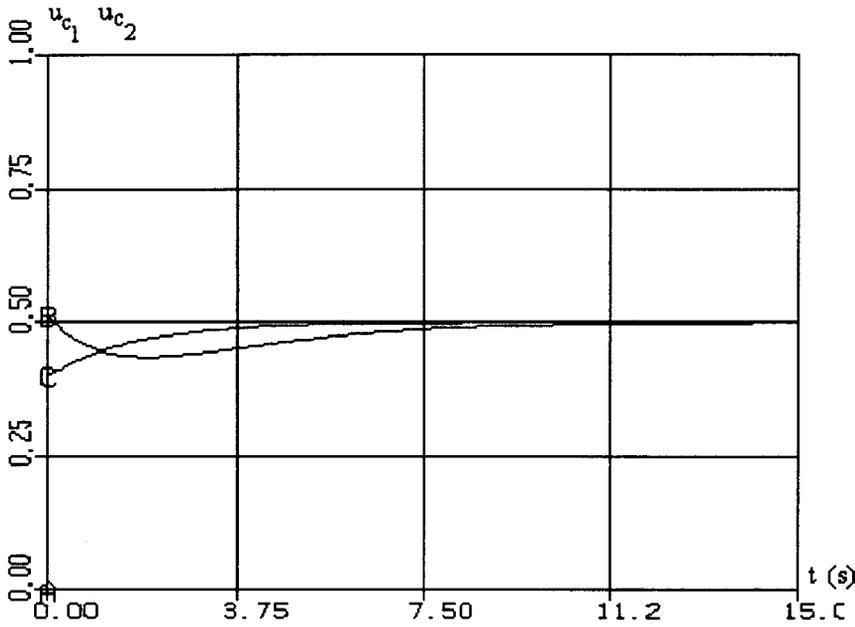


Fig. 3.21a : Commandes continues.

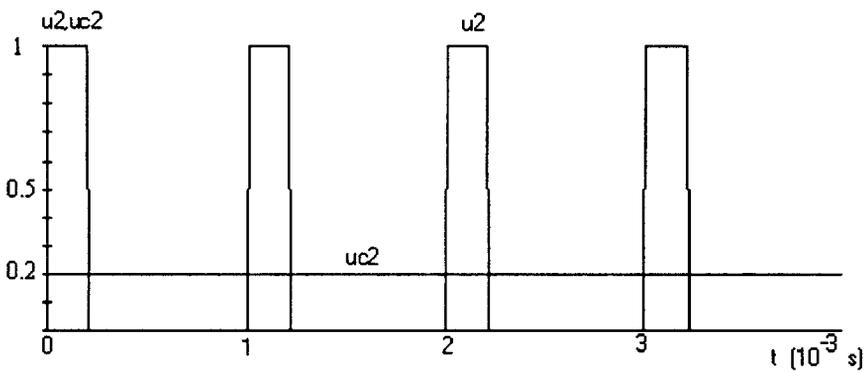


Fig.3.21b : Commande continue u_{c2} et commande binaire u_2 .

Remarque :

La variation de u_{c2} sur le pas choisi est très faible et n'est pas perceptible sur cette figure.

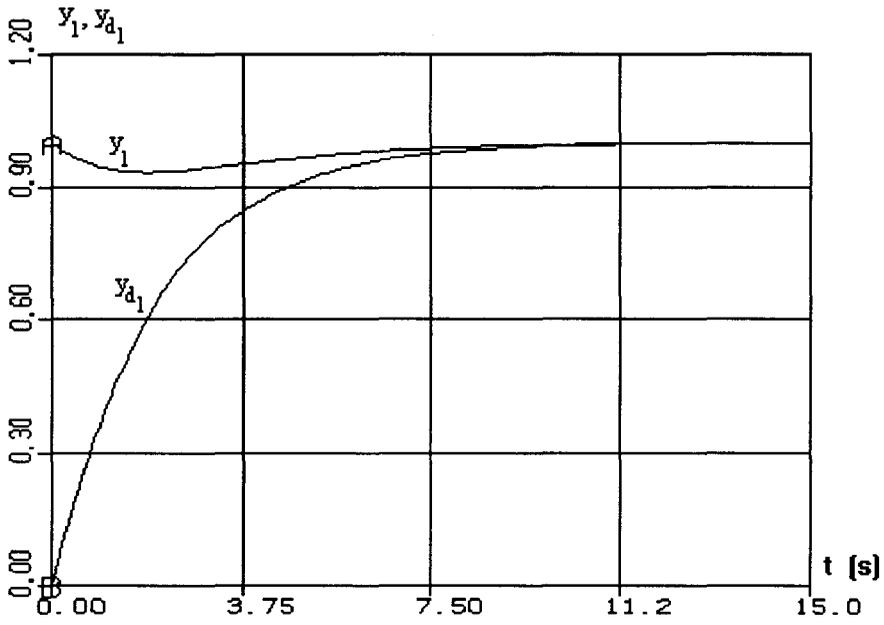


Fig.3.21c : Réponse temporelle de la première sortie.

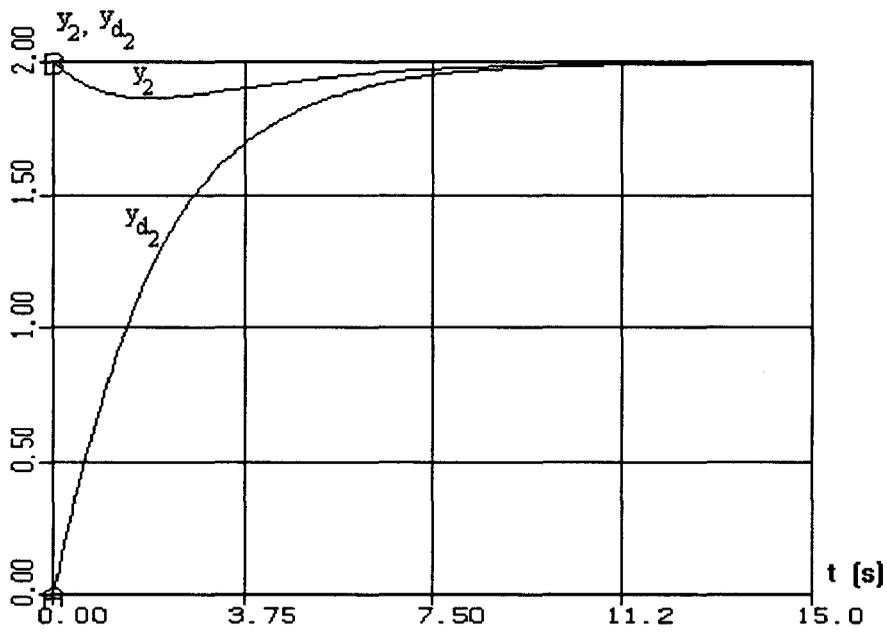
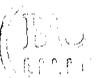


Fig.3.21d : Réponse temporelle de la deuxième sortie.

III.3.6.2 Exemple 2

Reprenons l'exemple 2 du §III.3.3.2 décrit par :



$$\begin{cases} \dot{x} = \begin{pmatrix} -x_3^2 \\ -x_1^2 \\ x_2 \end{pmatrix} + \begin{pmatrix} 5 & 0 \\ 0 & 3 \\ 0 & 0 \end{pmatrix} \begin{pmatrix} u_{c_1} \\ u_{c_2} \end{pmatrix} \\ y = \begin{pmatrix} 1 & 0 & 0 \\ 0 & 0 & 1 \end{pmatrix} x \end{cases}$$

et que l'on veut faire converger vers $y_d=(1,2)^T$.

Nous avons vu que u_{c_1} et u_{c_2} pouvaient être exprimées en fonction de t par :

$$\begin{cases} u_{c_1}(t) = \frac{1}{5}(-k_1(y_{10}-1)e^{-k_1 t} + (2 + Ae^{-\rho_1 t} + Be^{-\rho_2 t})^2) \\ u_{c_2}(t) = \frac{1}{3}(A\rho_1^2 e^{-\rho_1 t} + B\rho_2^2 e^{-\rho_2 t} + (1 + (y_{10}-1)e^{-k_1 t})^2) \end{cases}$$

Nous avons vu au §III.3.3.2 que les valeurs $\rho_1=1$, $\rho_2=2$, $k_1=5$ donnaient des commandes continues à valeurs dans $[0,1]$. u_c a alors l'allure suivante (fig.3.22) :

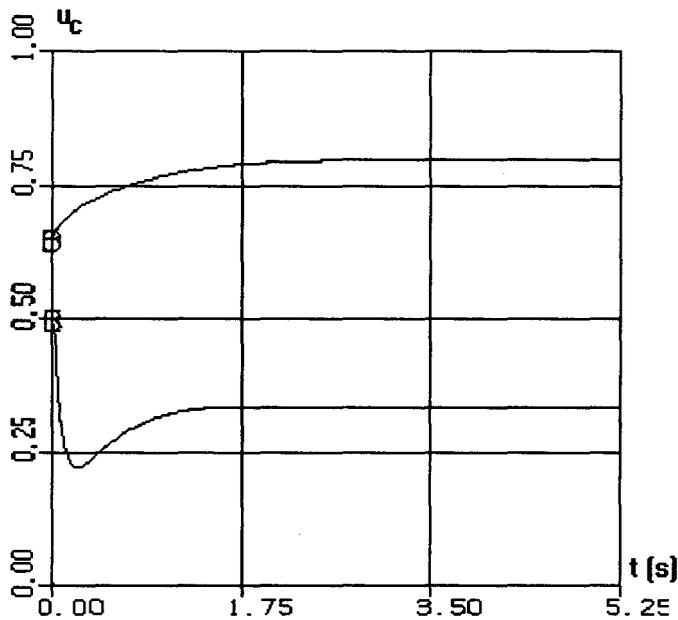


Fig.3.22 : Représentation temporelle de u_c .

Les commandes continues sont binarisées en utilisant la procédure décrite au §III.3.4.2, en choisissant comme période $T_k=10^{-3}s$. Une simulation du système commandé par des entrées binaires est donnée fig.3.23 :

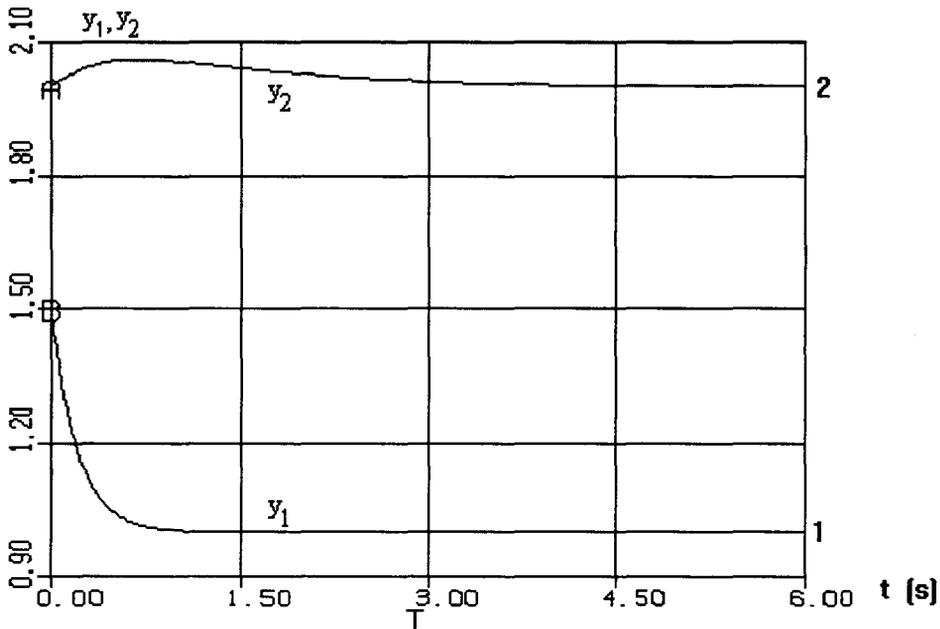


Fig.3.23 : Réponse temporelle des sorties du système.

Remarque :

La somme des degrés relatifs du système vaut $n=3$ ce qui assure la commandabilité du système continu (pas de zéros dynamiques). D'autre part, $\partial h_i / \partial x_j$ et g_i sont constants (donc bornés) ce qui nous permet de démontrer la convergence du système à commandes binaires (voir §III.3.4.3).

III.3.6.3 Exemple 3

Reprenons l'exemple 3 du §III.3.3.2.2 décrit par :

$$\begin{cases} \dot{x}_1 = -\frac{1}{5}x_3 + u_1 \\ \dot{x}_2 = -x_1 + u_2 \\ \dot{x}_3 = \sin(x_3) + x_1 \\ y_1 = x_1 \\ y_2 = x_2 \end{cases}$$

que l'on veut faire converger vers $y_d = (1 - e^{-t}, (1 - e^{-t})/2)^T$.

Nous avons vu que le retour linéarisant pour le système continu associé est donné par :

$$\begin{cases} u_{c_1} = v_1 + \frac{1}{5}x_3 = \frac{dy_1}{dt} + \frac{1}{5}x_3 \\ u_{c_2} = v_2 + x_1 = \frac{dy_2}{dt} + x_1 \end{cases}$$

III.3.6.3.1 Zéros dynamiques

L'étude de la stabilité des zéros dynamiques doit être menée car la somme des degrés relatifs vaut 2. Ceci revient à étudier la stabilité autour de l'équilibre du système :

$$\begin{cases} \dot{x}_1 = 1 - y_{d_1} - k_{p_1}(x_1 - y_{d_1}) \\ \frac{dy_{d_1}}{dt} = 1 - y_{d_1} \\ \dot{x}_3 = \sin(x_3) + x_1 \\ x_{10} = 1, x_{30} = 1, y_{d_{10}} = 0 \end{cases}$$

Les points d'équilibres du système sont $(x_1, y_{d_1}, x_3) = (1, 1, 3\pi/2 + 2k\pi)$, pour $k \in \mathbf{Z}$. On pose donc $Y_1 = x_1 - 1, Y_2 = y_{d_1} - 1, Z = x_3 - (3\pi/2 + 2k\pi)$. Le système décrit par :

$$\begin{cases} \dot{Y}_1 = -k_{p_1}Y_1 + (k_{p_1} - 1)Y_2 \\ \frac{dY_2}{dt} = -Y_2 \\ \dot{Z} = -\cos(Z) + Y_1 + 1 \\ Y_{10} = 0, Y_{20} = -1, Z_0 = 1 - (3\pi/2 + 2k\pi) \end{cases}$$

converge alors vers l'équilibre $(Y_1, Y_2, Z) = (0, 0, 0)$.

Rappel de stabilité :

Soit le système (Σ_{yz}) décrit par :

$$\begin{cases} \dot{y} = \mathbf{A}y + p(z, y) \\ \dot{z} = f(z, y) \end{cases}$$

tel que : $p(z, 0) = 0, \frac{\partial p}{\partial y}(0, 0) = 0, \mathbf{A}$ matrice de Hurwitz. Si $z=0$ est équilibre asymptotiquement stable de $\dot{z} = f(z, 0)$, alors $(0, 0)$ est un équilibre asymptotiquement stable pour (Σ_{yz}) .

En appliquant les notations du théorème à notre étude, nous avons :

$$A = \begin{pmatrix} -k_{p1} & k_{p1} - 1 \\ 0 & -1 \end{pmatrix}, p(Z, Y) = 0, f(Z, Y) = 1 - \cos(Z) + Y_1$$

Les hypothèses du théorème sont donc vérifiées et il suffit de montrer que $z=0$ est un équilibre asymptotiquement stable pour $\dot{Z} = f(Z, 0) = 1 - \cos(Z)$.

L'équation ci-dessus peut être intégrée (variables séparables), et on obtient : $Z = 2 \operatorname{atan}\left(\frac{1}{K_0 - t}\right)$, avec $K_0 = \tan(Z_0/2)$, pour lequel $Z=0$ est un équilibre asymptotiquement stable.

III.3.6.3.2 Simulation du système à entrées binaires

Nous avons choisi $k_1 = k_2 = 1.1$, ce qui nous permet d'avoir des composantes de l'entrée continue équivalente dans $[0, 1]$. Nous avons utilisé la procédure de binarisation décrite au §III.3.4.2 avec $T_k = 10^{-3} s$.

Les sorties du système sont représentées fig.3.24a,b :

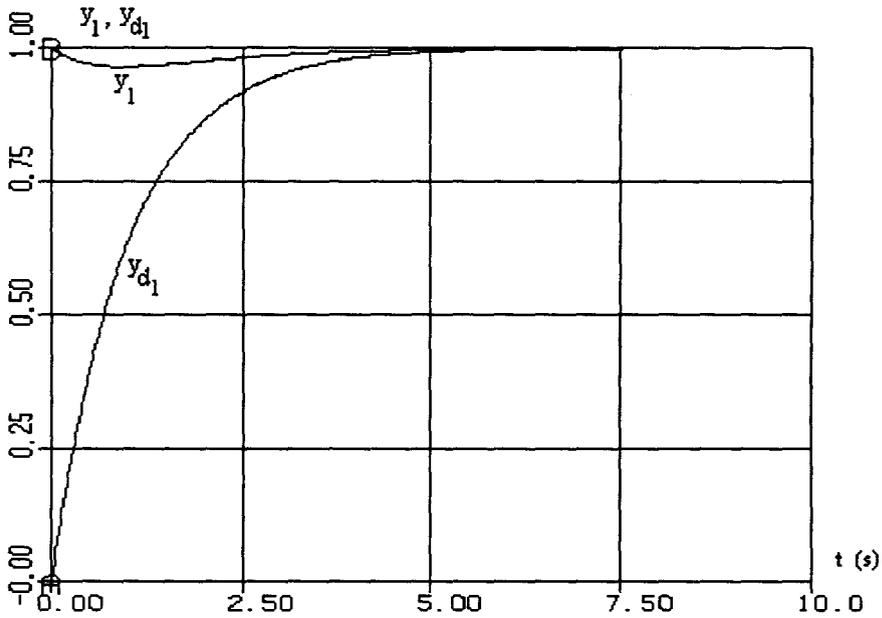
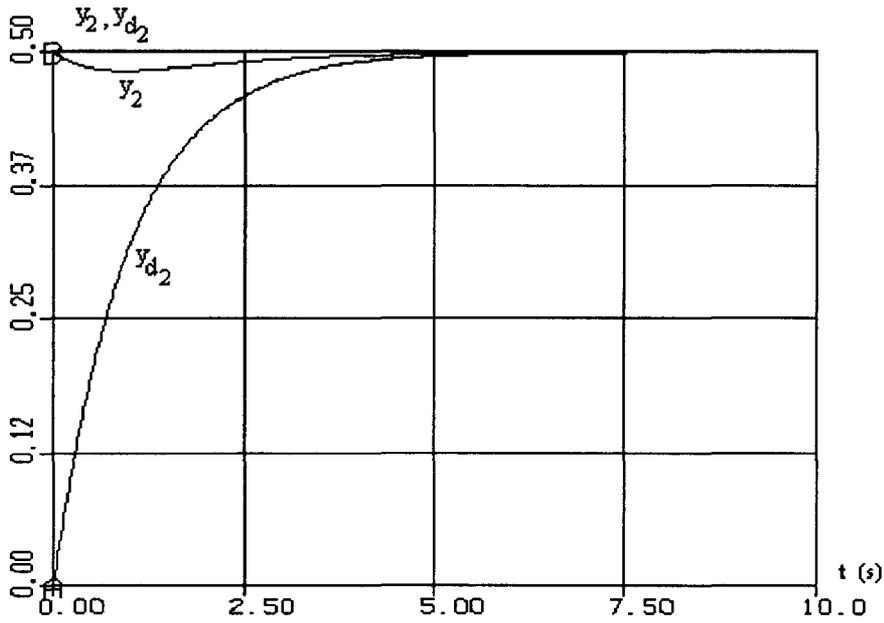


Fig.3.24a : Représentation temporelle de la sortie y_1 .

Fig.3.24b : Représentation temporelle de la sortie y_2 .

CONCLUSION

La première partie du chapitre a été consacrée à l'étude des systèmes bilinéaires à entrées booléennes. Après avoir élaboré un algorithme de commande analogue au cas linéaire, les résultats obtenus ont été appliqués sur plusieurs exemples. Nous avons ensuite élaboré des lois de contrôle pour les systèmes non-linéaires à entrées booléennes. Nous avons tout d'abord proposé des lois de commande en boucle ouverte et en boucle fermée pour les systèmes linéarisables par morceaux, en appliquant les méthodes de calcul élaborées dans le chapitre II sur les systèmes linéaires.

Le cas des systèmes non-linéaires affines a été ensuite envisagé pour lesquels nous avons proposé une méthode de construction d'un algorithme de découplage-linéarisation entrées-sorties par retour d'état statique en utilisant un modèle continu construit à partir du modèle à entrées booléennes. Cette technique a été appliquée à la régulation et à la poursuite de trajectoire et validée par simulation sur des exemples.

Les différentes méthodes évoquées au cours de ce chapitre seront comparées entre elles au cours du chapitre IV, en simulation, pour la commande de machines électriques associées à des convertisseurs de puissance.

CHAPITRE IV

COMMANDE DE MACHINES ELECTRIQUES

CHAPITRE IV

INTRODUCTION

En électronique de puissance, la plupart des systèmes sont commandés par des éléments de commutation pour permettre une répartition optimale de l'énergie. Ces éléments ne sont habituellement pas pris en compte dans la modélisation et on utilise en général, comme on a vu au premier chapitre, des méthodes de commande qui ne tiennent pas compte du caractère booléen des grandeurs d'entrée [MLI].

Dans ce chapitre, nous allons appliquer les résultats que nous avons élaborés dans les chapitres II et III à la commande de certains de ces systèmes. Nous traiterons tout d'abord, le cas d'un système électrique composé d'une machine synchrone associée à un onduleur de tension pour lequel nous effectuerons une comparaison entre les algorithmes classiques et ceux élaborés au cours des chapitres précédents. Pour cela nous utiliserons un modèle bond-graph de ce système et nous effectuerons une transformation de Park et Concordia [Huang et al, 1987] pour en réduire la dimension. Nous envisagerons ensuite le cas de la machine asynchrone pour lequel nous appliquerons l'algorithme par linéarisation par morceaux élaboré au chapitre III, et un algorithme "classique" de commande par MLI. Nous envisagerons alors à travers un exemple numérique l'effet des variations de certains paramètres du système sur la robustesse de l'algorithme.

IV.1 CAS DE LA MACHINE SYNCHRONE

IV.1.1 Modélisation par bond-graph

Nous allons étudier ici un ensemble comportant un convertisseur de puissance associé à un moteur synchrone. Nous considérerons que le convertisseur est un onduleur de tension alimenté par une tension continue.

IV.1.1.1 Schéma électrique de l'onduleur

On peut donner le schéma électrique de l'onduleur (fig. 4.1) :

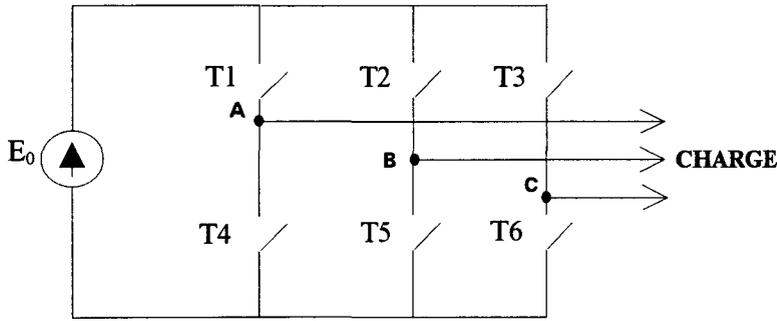


Fig.4.1 : Schéma électrique d'un onduleur de tension alimenté par une tension continue.

T1,T2,...,T6 représentent des interrupteurs dont on peut commander l'état (bloquant ou passant). E_0 représente la tension continue d'alimentation de l'onduleur.

IV.1.1.2 Représentation bond-graph de l'onduleur

Les éléments de commutation T1,T2,...,T6, sont considérés comme non-idéaux avec des résistances internes $R_{t1}, R_{t2}, \dots, R_{t6}$. En utilisant la méthode de modélisation des interrupteurs du chapitre I, on peut construire un modèle bond-graph de l'onduleur.

Le modèle bond-graph est alors (fig.4.2) :

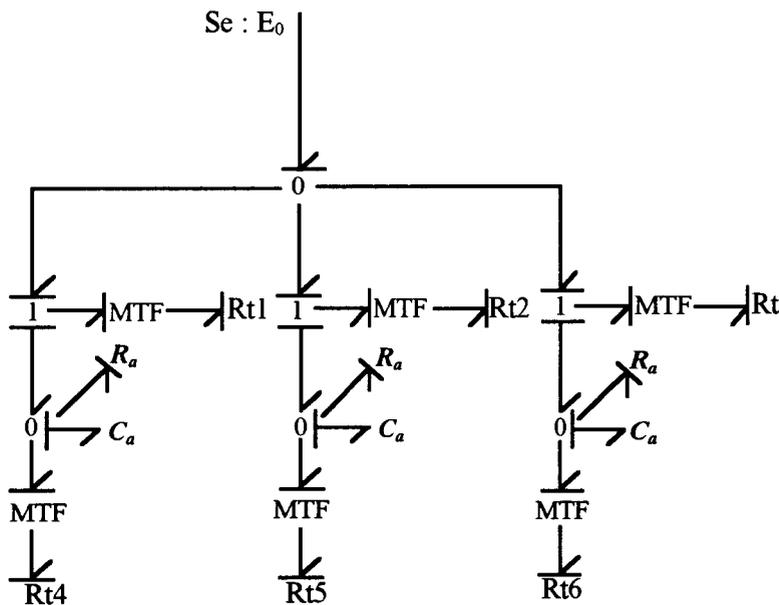


Fig.4.2 : Bond-graph d'un onduleur de tension alimenté par une source de tension continue.

Les transformateurs MTF correspondant aux interrupteurs T1,T2,...,T6 sont modulés respectivement par $\{1/m_1, \dots, 1/m_6\}$ où les $m_i, i=1, \dots, 6$ sont des booléens tels que :

$$\begin{cases} m_i = 0 \text{ ssi } T_i \text{ est bloqué} \\ m_i = 1 \text{ ssi } T_i \text{ est passant} \end{cases}$$

Les blocs R_a, C_a en parallèles sont ajoutés entre les points A, B, C et la masse pour obtenir une mesure des tensions simples dans chacune des branches. Leurs valeurs numériques doivent être choisies pour ne pas perturber la dynamique du modèle (petites constantes de temps).

IV.1.1.3 Association onduleur-machine synchrone

Cet onduleur est associé à un moteur synchrone à rotor bobiné, ce que l'on peut représenter par un schéma électrique (fig.4.3) :

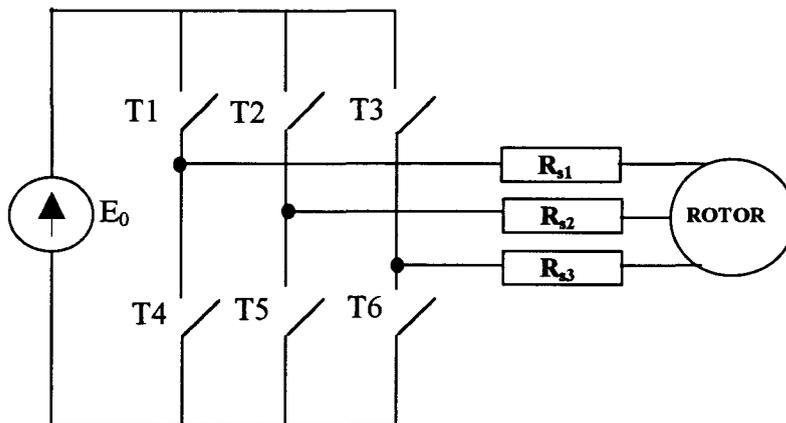


Fig. 4.3 : Schéma électrique d'une association onduleur-moteur synchrone.

$R_{s_i}, i=1,2,3$ représentent les résistances statoriques et nous supposons par la suite que ces grandeurs sont les mêmes sur les trois branches du moteur synchrone. Le rotor est supposé bobiné lisse avec une seule paire de pôle. Pour plus de détails sur la machine synchrone, on pourra se référer aux ouvrages suivants : [Bergman et al, 1986], [Le Piouffle et al, 1990].

IV.1.1.4 Modèle bond-graph de l'ensemble onduleur-moteur synchrone

Le modèle bond-graph de l'ensemble [Ducreux et al, 1992] est présenté (fig.4.4) :

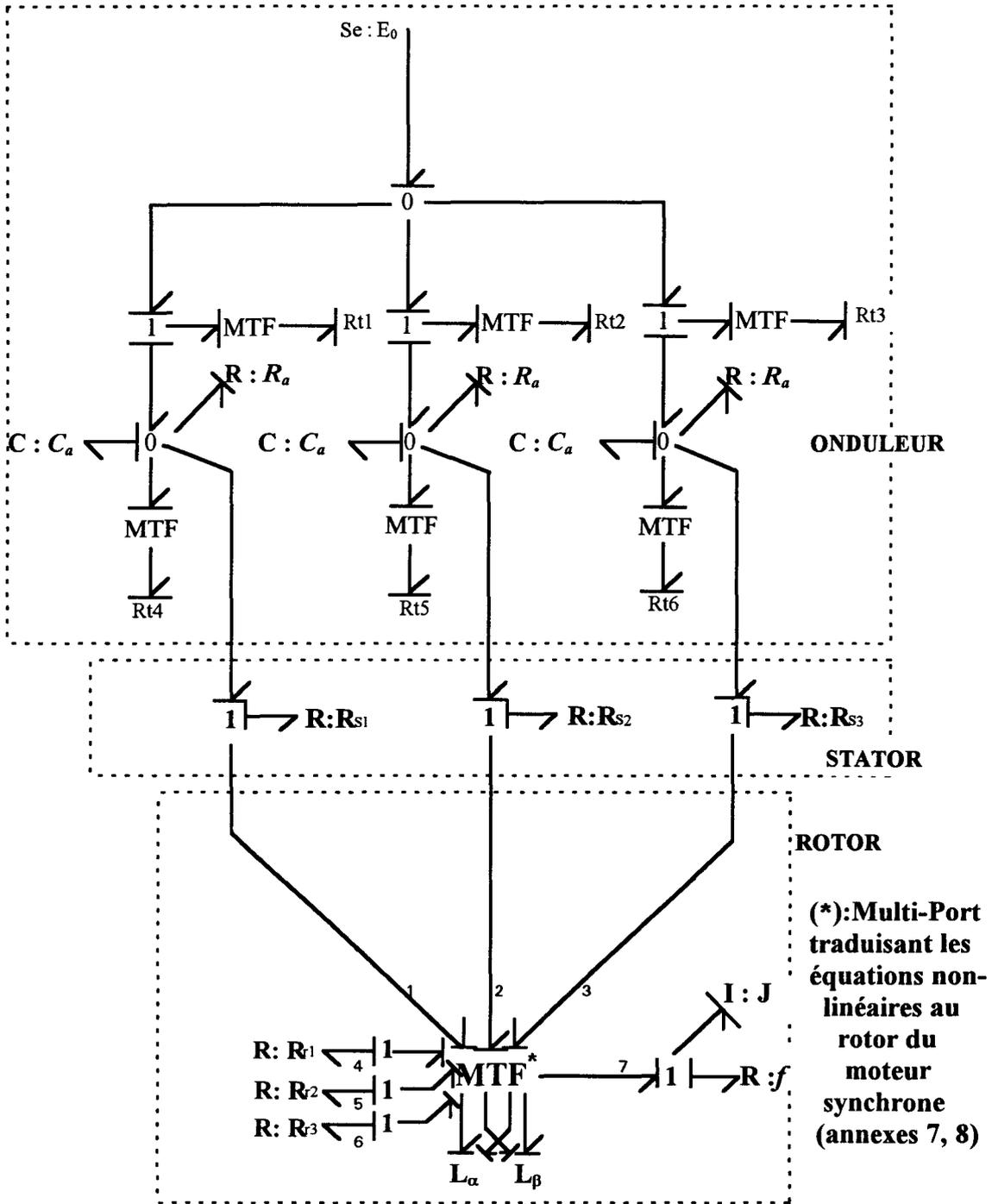


Fig.4.4 : Bond-graph d'un moteur synchrone associé à un onduleur de tension.

R_{r1} , R_{r2} , R_{r3} représentent les résistances rotoriques. J et f représentent l'inertie et le frottement rotorique. L'élément MTF multi-ports traduit les relations non linéaires qui lient les efforts et les flux des branches 1 à 7 : $i = L^{-1} \phi$, avec $\phi = (\phi_1, \phi_2, \phi_3, \phi_4, \phi_5, \phi_6)^T$, $i = (i_1, i_2, i_3, i_4, i_5, i_6)^T$, (matrice inductance L définie dans l'annexe 7) et la transformation de Park (annexe 8).



IV.1.2 Equation d'état du système

Le bond-graph (fig.4.4) nous permet d'obtenir les équations reliant les variables efforts et flux aux différentes jonctions (annexe 7). On considère que les constantes de temps liées aux blocs $R_a C_a$ sont petites et que la machine synchrone est équilibrée au stator et au rotor. D'autre part, on considère que la commande de l'onduleur de tension est complémentaire c'est à dire que T1 et T4 (resp. T2 et T5, T3 et T6) ne sont ni bloqués ni passants simultanément. Cette hypothèse entraîne :

$$\begin{cases} m_i^2 + m_{i+3}^2 = 1 \\ i = 1,2,3 \end{cases} \quad (4.1)$$

Après simplifications (voir annexes 7 et 8) les équations du système peuvent être décrites :

$$\begin{cases} \frac{di_\alpha}{dt} = \frac{-R}{L} i_\alpha + \frac{\phi_f}{L} \omega \sin(\theta) + \frac{1}{L} u_\alpha \\ \frac{di_\beta}{dt} = \frac{-R}{L} i_\beta - \frac{\phi_f}{L} \omega \cos(\theta) + \frac{1}{L} u_\beta \\ \frac{d\theta}{dt} = \omega \\ \frac{d\omega}{dt} = \frac{-f}{J} \omega + \frac{\phi_f}{J} (i_\beta \cos(\theta) - i_\alpha \sin(\theta)) - \frac{C_r}{J} \end{cases} \quad (4.2)$$

i_α et i_β sont les intensités statoriques équivalentes après réduction du modèle. θ représente la position du rotor par rapport au référentiel fixe statorique. ω est la vitesse du moteur, ϕ_f représente l'amplitude des flux rotoriques.

R et L représentent la résistance et l'inductance équivalente au stator.

u_α et u_β sont les tensions appliquées au stator. Elles peuvent être exprimées en fonction des grandeurs booléennes de commutation :

$$\begin{cases} u_\alpha = \frac{E_0}{\sqrt{6}} (2m_1^2 - m_2^2 - m_3^2) \\ u_\beta = \frac{E_0}{\sqrt{2}} (m_2^2 - m_3^2) \end{cases} \quad (4.3)$$

Les grandeurs m_1, m_2, m_3 représentent les booléens associés aux éléments de commutation T1, T2, T3. En posant $u=(u_1, u_2, u_3)=(m_1^2, m_2^2, m_3^2)$, l'équation (4.3) s'écrit :

$$\begin{cases} u_\alpha = \frac{E_0}{\sqrt{6}}(2u_1 - u_2 - u_3) \\ u_\beta = \frac{E_0}{\sqrt{2}}(u_2 - u_3) \end{cases}$$

Pour plus de clarté nous allons utiliser les variables suivantes. On posera x , le vecteur état du système tel que :

$$x(t) = (x_1, x_2, x_3, x_4)^T = (i_\alpha, i_\beta, \theta, \omega)^T$$

Nous utiliserons les grandeurs réduites suivantes :

$$\begin{cases} \tau_m = \frac{J}{f} \\ \tau = \frac{L}{R} \\ a = \frac{\phi_f}{L} \\ b = \frac{\phi_f}{J} \\ \gamma_r = \frac{C_r}{J} \end{cases} \quad (4.4)$$

Le système représenté par (4.2) peut alors être décrit par l'équation d'état suivante :

$$\frac{dx}{dt} = \begin{pmatrix} -\frac{1}{\tau} x_1 + a x_4 \sin(x_3) \\ -\frac{1}{\tau} x_2 - a x_4 \cos(x_3) \\ x_4 \\ -\frac{1}{\tau_m} x_4 + b(x_2 \cos(x_3) - x_1 \sin(x_3)) - \gamma_r \end{pmatrix} + \frac{E_0}{L\sqrt{6}} \begin{pmatrix} 2 & -1 & -1 \\ 0 & \sqrt{3} & \sqrt{3} \\ 0 & 0 & 0 \\ 0 & 0 & 0 \end{pmatrix} \begin{pmatrix} u_1 \\ u_2 \\ u_3 \end{pmatrix} \quad (4.5)$$

Cette équation d'état est non-linéaire et peut être exprimée sous la forme :

$$\dot{x} = f(x) + Gu$$

Nous allons appliquer plusieurs algorithmes de commande sur le même système afin d'effectuer une comparaison de leurs performances.

IV.1.3 Commande par retour d'état

IV.1.3.1 Commande continue

Dans cette partie, nous allons appliquer les résultats du chapitre III concernant la commande linéarisante par bouclage statique.

Pour cela, on considère le système continu décrit par l'équation (4.5) avec une entrée continue notée u_c . Nous choisissons de plus comme grandeurs de sorties le courant $i_\alpha \cos(\theta) + i_\beta \sin(\theta)$ (courant statorique dans le repère tournant) ainsi que la vitesse $\omega = x_4$, soit :

$$y = \begin{pmatrix} y_1 \\ y_2 \end{pmatrix} = \begin{pmatrix} x_1 \cos(x_3) + x_2 \sin(x_3) \\ x_4 \end{pmatrix} \quad (4.6)$$

Le système (4.5) avec une commande continue est décrit par :

$$\begin{cases} \dot{x} = f(x) + Gu_c \\ y = h(x) \end{cases}$$

ou :

$$\begin{cases} \dot{x} = f(x) + \begin{pmatrix} 1 & 0 \\ 0 & 1 \\ 0 & 0 \\ 0 & 0 \end{pmatrix} U_c \\ y = h(x) \end{cases} \quad (4.7)$$

avec $U_c = (U_{c1}, U_{c2})^T$, les grandeurs continues associées à $(u_\alpha/L, u_\beta/L)^T$.

Les entrées apparaissent dès la première dérivation pour y_1 (degré relatif égal à 1), et à la deuxième dérivation pour y_2 . La somme des degrés relatifs du système est alors égale à 3. Il reste donc une partie non-linéaire après bouclage dont nous étudierons la stabilité au paragraphe suivant. On a, en reprenant les notations du chapitre III :

$$\begin{pmatrix} \frac{dy_1}{dt} \\ \frac{d^2 y_2}{dt^2} \end{pmatrix} = \Delta_0(x) + \Delta(x)U_c \quad (4.8)$$

avec :

$$\Delta_0(x) = \begin{pmatrix} \Delta_{01}(x) \\ \Delta_{02}(x) \end{pmatrix} = \begin{pmatrix} (f_1(x) + x_2 x_4) \cos(x_3) + (f_2(x) - x_1 x_4) \sin(x_3) \\ \frac{-1}{\tau_m} f_4(x) - (f_1(x) + x_2 x_4) b \sin(x_3) + (f_2(x) - x_1 x_4) b \cos(x_3) \end{pmatrix}$$

et

$$\Delta(x) = \begin{pmatrix} \cos(x_3) & \sin(x_3) \\ -b \sin(x_3) & b \cos(x_3) \end{pmatrix} \quad (4.9)$$

L'inversion de la matrice de découplage donne :

$$\Delta^{-1}(x) = \frac{1}{b} \begin{pmatrix} b \cos(x_3) & -\sin(x_3) \\ b \sin(x_3) & \cos(x_3) \end{pmatrix} \quad (4.10)$$

On souhaite réguler la vitesse y_2 vers ω_d (constante) et y_1 vers 0. Si on choisit des dynamiques pour les sorties telles que :

$$\begin{cases} v_1 = \frac{dy_1}{dt} = -k_1 y_1 \\ v_2 = \frac{d^2 y_2}{dt^2} = -\eta \frac{dy_2}{dt} - k_2 (y_2 - \omega_d) \end{cases} \quad (4.11)$$

avec $k_1 > 0$ et les solutions de $X^2 + \eta X + k_2 = 0$ à parties réelles négatives, il faut alors choisir U_c telle que :

$$U_c = -\Delta^{-1}(x)\Delta_0(x) + \Delta^{-1}(x)v = \begin{pmatrix} \frac{1}{b} [b \cos(x_3)(v_1 - \Delta_{01}(x)) - \sin(x_3)(v_2 - \Delta_{02}(x))] \\ \frac{1}{b} [b \sin(x_3)(v_1 - \Delta_{01}(x)) + \cos(x_3)(v_2 - \Delta_{02}(x))] \end{pmatrix} \quad (4.12)$$

avec v_1 et v_2 définis par (4.11).

IV.1.3.2 Etude des zéros dynamiques

La non-singularité de la matrice de découplage assure l'existence d'un difféomorphisme $\phi : x \rightarrow z$ dont les trois premières composantes sont $(h_1(x), h_2(x), L_f h_2(x))$. Si nous choisissons comme quatrième composante $\phi_4(x) = x_3 = \theta$, on obtient :

$$\phi = \begin{pmatrix} z_1 \\ z_2 \\ z_3 \\ z_4 \end{pmatrix} = \begin{pmatrix} h_1(x) \\ h_2(x) \\ L_f h_2(x) \\ x_3 \end{pmatrix} = \begin{pmatrix} x_1 \cos(x_3) + x_2 \sin(x_3) \\ x_4 \\ -\frac{1}{\tau_m} x_4 + b(x_2 \cos(x_3) - x_1 \sin(x_3)) - \gamma_r \\ x_3 \end{pmatrix} \quad (4.13)$$

Le système en boucle fermée est alors régi par :

$$\begin{cases} \dot{z}_1 = v_1 \\ \dot{z}_2 = z_3 \\ \dot{z}_3 = v_2 \\ \dot{z}_4 = z_2 \\ y_1 = z_1 \\ y_2 = z_2 \end{cases} \quad (4.14)$$

Ce système est linéaire, découplé et commandable et il reste une partie dynamique non observable z_4 . Cette dynamique ne pose pas de problème quant à sa stabilité puisqu'elle correspond au lien entre la position du rotor et la vitesse.

IV.1.3.3 Binarisation de la commande

D'après les relation (4.2), on a :

$$\begin{cases} u_\alpha \in \left\{ \frac{-2E_0}{\sqrt{6}}, \frac{-E_0}{\sqrt{6}}, 0, \frac{E_0}{\sqrt{6}}, \frac{2E_0}{\sqrt{6}} \right\} \\ u_\beta \in \left\{ \frac{-E_0}{\sqrt{2}}, 0, \frac{E_0}{\sqrt{2}} \right\} \end{cases} \quad (4.15)$$

Seules les valeurs $\frac{\pm E_0}{\sqrt{6}}$ pour u_α et $\frac{\pm E_0}{\sqrt{2}}$ pour u_β peuvent être obtenues indépendamment. En effet, pour obtenir par exemple $u_\alpha = \frac{-2E_0}{\sqrt{6}}$, il faut obligatoirement appliquer $(u_1, u_2, u_3) = (0, 1, 1)$ ce qui fige u_β à 0. Dans ce cas (comme pour $u_\alpha = \frac{2E_0}{\sqrt{6}}$) le choix de u_α impose la valeur de u_β . Les quatre possibilités pour le couple (u_α, u_β) ,

$$\left\{ \left(\frac{-E_0}{\sqrt{6}}, \frac{-E_0}{\sqrt{2}} \right), \left(\frac{-E_0}{\sqrt{6}}, \frac{E_0}{\sqrt{2}} \right), \left(\frac{E_0}{\sqrt{6}}, \frac{-E_0}{\sqrt{2}} \right), \left(\frac{E_0}{\sqrt{6}}, \frac{E_0}{\sqrt{2}} \right) \right\} \quad (4.16)$$

sont obtenues à partir des configurations de u :

$$\left\{ \begin{pmatrix} 0 \\ 0 \\ 1 \end{pmatrix}, \begin{pmatrix} 0 \\ 1 \\ 0 \end{pmatrix}, \begin{pmatrix} 1 \\ 0 \\ 1 \end{pmatrix}, \begin{pmatrix} 1 \\ 1 \\ 0 \end{pmatrix} \right\}$$

Le domaine temporel est découpé en intervalles de période T_k , $I_k=[nT_k, (n+1)T_k]$. Soient $\hat{U}_{c_1}(nT_k)$ et $\hat{U}_{c_2}(nT_k)$ les valeurs moyennes de U_{c1} et U_{c2} sur I_k .

On construit u_α et u_β tels que :

$$\left\{ \begin{array}{l} u_\alpha = \frac{E_0}{\sqrt{6}} \text{ sur } \left[nT_k, nT_k + \left(\frac{1}{2} + \frac{L\sqrt{6}}{2E_0} \hat{U}_{c_1}(nT_k) \right) T_k \right] \\ u_\alpha = \frac{-E_0}{\sqrt{6}} \text{ sur } \left[nT_k + \left(\frac{1}{2} + \frac{L\sqrt{6}}{2E_0} \hat{U}_{c_1}(nT_k) \right) T_k, (n+1)T_k \right] \\ u_\beta = \frac{E_0}{\sqrt{2}} \text{ sur } \left[nT_k, nT_k + \left(\frac{1}{2} + \frac{L\sqrt{2}}{2E_0} \hat{U}_{c_2}(nT_k) \right) T_k \right] \\ u_\beta = \frac{-E_0}{\sqrt{2}} \text{ sur } \left[nT_k + \left(\frac{1}{2} + \frac{L\sqrt{2}}{2E_0} \hat{U}_{c_2}(nT_k) \right) T_k, (n+1)T_k \right] \end{array} \right. \quad (4.17)$$

L'algorithme de commande peut être schématisé (fig.4.5) :

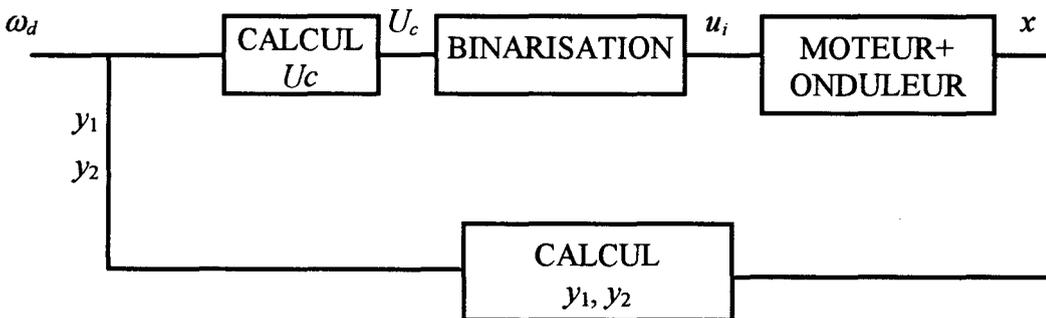


Fig.4.5 : Algorithme de commande de la machine synchrone.

Le bloc commande comporte une phase de calcul de la variable continue U_c décrite au paragraphe précédent et des phases de binarisation de U_c et de calcul de u décrites précédemment dans ce paragraphe.

IV.1.3.3 Application numérique et simulation

On a choisi pour la simulation une machine synchrone ayant les caractéristiques suivantes :

$$\begin{aligned} R &= 0.1 \, \Omega \\ L &= 100 \, \text{mH} \\ f &= 2 \cdot 10^{-3} \, \text{N.s} \\ J &= 8 \cdot 10^{-3} \, \text{kg.m}^2 \\ E_0 &= 380 \, \text{V} \\ \gamma &= 10 \end{aligned}$$

On choisit $\omega_d = 100 \, \text{rad/s}$ pour la consigne en vitesse et $\phi_f = 1$.

Les dynamiques de l'erreur en sortie sont choisies telles que :

$$U_c \in \left[\frac{-E_0}{L\sqrt{6}}, \frac{E_0}{L\sqrt{6}} \right] \times \left[\frac{-E_0}{L\sqrt{2}}, \frac{E_0}{L\sqrt{2}} \right]$$

Les entrées continues ont été simulées pour différentes valeurs des paramètres η , k_1 , k_2 . Nous ne présentons pas les différentes courbes obtenues pour U_c afin de ne pas alourdir le manuscrit. Une telle étude montre que les valeurs des paramètres telles que $\eta < 4$, $k_1 < 2$, $k_2 < 3$ donnent bien des composantes de U_c dans les intervalles désirés.

Pour la suite nous avons donc choisi $\eta = 3$, $k_2 = 2$, $k_1 = 1$. Nous pouvons alors représenter les composantes de U_c pour ces valeurs (fig.4.6a,b) :

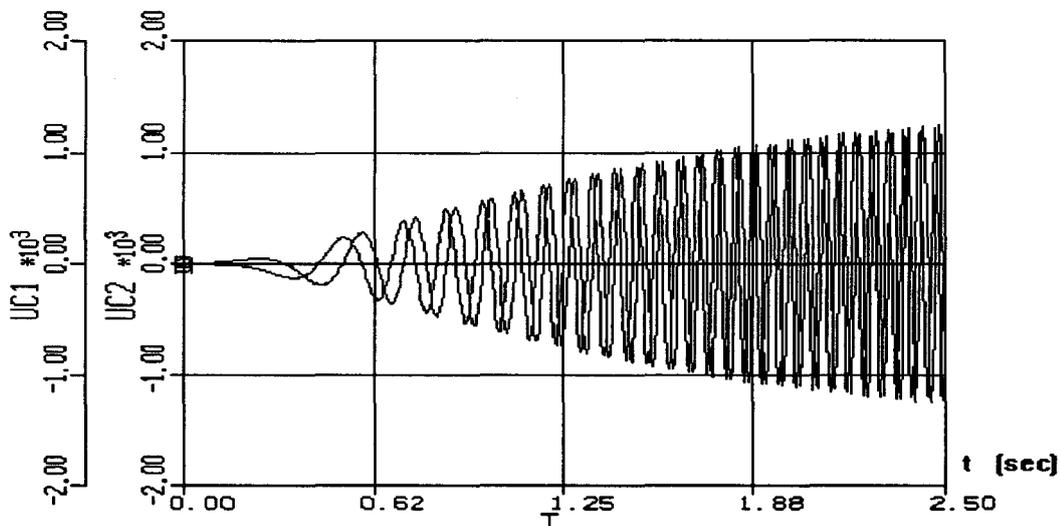


Fig.4.6a : Commandes continue : U_{c1} et U_{c2} .

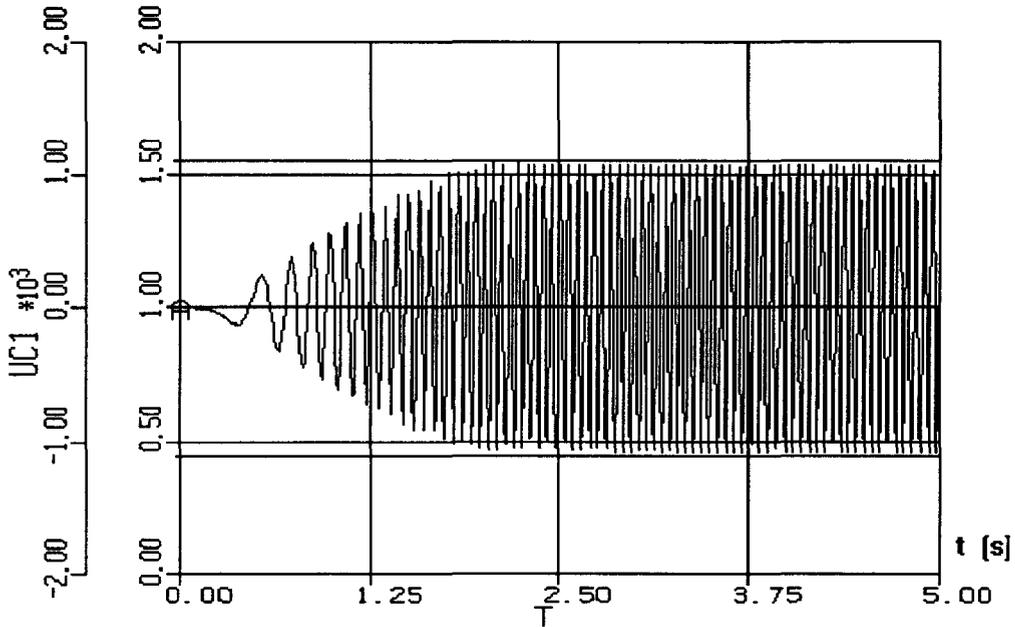


Fig.4.6b : Commande continue : U_{c1} (horizon plus large).

La période de discrétisation de l'entrée est choisie telle que $T_k=10^{-3}s$ et la période d'échantillonnage $T =10^{-5}s$. Les réponses temporelles de l'entrée u_1 et des sorties sont représentées fig.4.7a,b,c :

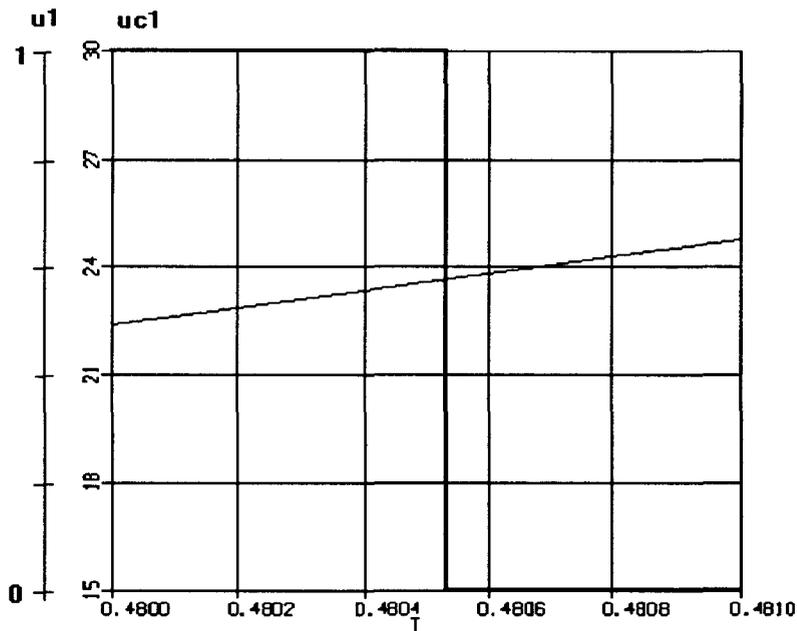


Fig.4.7a : Représentation de l'entrée u_1 .

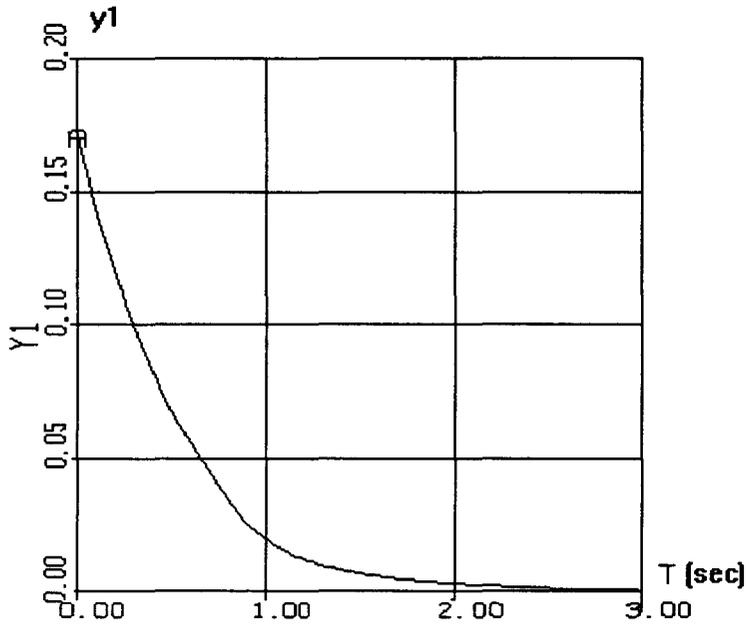


Fig.4.7b : Réponse temporelle de y_1 .

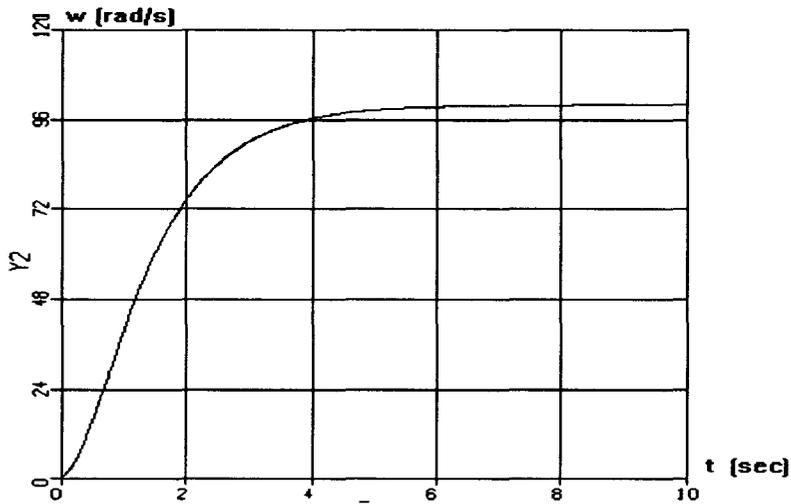


Fig.4.7c : Réponse temporelle de la vitesse du moteur.

IV.1.3.4 Conclusion

L'algorithme que nous venons d'établir peut poser certaines difficultés quant à sa mise en place pratique :

- 1) Nous ne disposons d'aucune information concernant les valeurs admissibles pour la consigne en vitesse (contrairement à la méthode de linéarisation par morceaux qui donne un domaine de convergence).

2) Le volume de calcul "ON LINE" est important et cela peut poser des problèmes pour une implantation sur le système réel.

Remarque :

L'algorithme de poursuite de trajectoire aurait pu être implanté en utilisant exactement la même procédure, seule la régulation a été présentée pour ne pas alourdir les calculs.

IV.1.4 Commande booléenne par modes glissants

IV.1.4.1 Elaboration de l'algorithme

Pour cette étude, nous élaborerons la loi de commande à partir du système linéarisé localement autour de $\omega=\omega_d$. Ceci entraîne $\theta=\omega_d t$, et en posant $x=(x_1, x_2, x_4)^T=(i_\alpha, i_\beta, \omega)^T$, l'équation (4.4) donne :

$$\dot{x} = \begin{pmatrix} -\frac{1}{\tau} x_1 + a\omega_d \sin(\omega_d t) + \frac{1}{L} u_\alpha \\ -\frac{1}{\tau} x_2 - a\omega_d \cos(\omega_d t) + \frac{1}{L} u_\beta \\ -\frac{1}{\tau_m} x_4 + b(x_2 \cos(\omega_d t) - x_1 \sin(\omega_d t)) - \gamma_r \end{pmatrix} \quad (4.18)$$

Si les courants statoriques convergent vers $(-i_0 \sin(\omega_d t), i_0 \cos(\omega_d t))$ alors la vitesse converge vers ω_d (voir annexe 9) avec $\omega_d = b \tau_m i_0 - \gamma_r \tau_m$

Considérons les surfaces de glissement non-stationnaires définies par :

$$s(x) = \begin{pmatrix} s_1(x) \\ s_2(x) \end{pmatrix} = \begin{pmatrix} x_1 + i_0 \sin(\omega_d t) \\ x_2 - i_0 \cos(\omega_d t) \end{pmatrix} = x + \mathbf{D}(t) \quad (4.19)$$

Nous pouvons alors définir la commande équivalente (voir chapitre I), u_{de} en dérivant s :

$$\begin{pmatrix} \dot{s}_1(x) \\ \dot{s}_2(x) \end{pmatrix} = \begin{pmatrix} -\frac{1}{\tau} x_1 + a\omega_d \sin(\omega_d t) + i_0 \omega_d \cos(\omega_d t) \\ -\frac{1}{\tau} x_2 - a\omega_d \cos(\omega_d t) + i_0 \omega_d \sin(\omega_d t) \end{pmatrix} + \frac{1}{L} u_{de} \quad (4.20)$$

soit sur l'intersection des surfaces de glissement :

$$u_{de} = -L \begin{pmatrix} \frac{1}{\tau} i_0 \sin(\omega_d t) + a \omega_d \sin(\omega_d t) + i_0 \omega_d \cos(\omega_d t) \\ -\frac{1}{\tau} i_0 \cos(\omega_d t) - a \omega_d \cos(\omega_d t) + i_0 \omega_d \sin(\omega_d t) \end{pmatrix} \quad (4.21)$$

Le système est donc décrit sur l'intersection des surfaces de glissement par :

$$\dot{x} = \begin{pmatrix} -\frac{1}{\tau} x_1 - \frac{1}{\tau} i_0 \sin(\omega_d t) - i_0 \omega_d \cos(\omega_d t) \\ -\frac{1}{\tau} x_2 + \frac{1}{\tau} i_0 \cos(\omega_d t) - i_0 \omega_d \sin(\omega_d t) \\ -\frac{1}{\tau_m} x_4 + b(x_2 \cos(\omega_d t) - x_1 \sin(\omega_d t)) - \gamma_r \end{pmatrix} \quad (4.22)$$

Nous avons défini alors la logique de commutation suivante :

$$\begin{cases} \left\{ \begin{array}{l} u_\alpha = \frac{E_0}{\sqrt{6}} \text{ si } s_1(x) < 0 \\ u_\alpha = -\frac{E_0}{\sqrt{6}} \text{ si } s_1(x) > 0 \end{array} \right. \\ \left\{ \begin{array}{l} u_\beta = \frac{E_0}{\sqrt{2}} \text{ si } s_2(x) < 0 \\ u_\beta = -\frac{E_0}{\sqrt{2}} \text{ si } s_2(x) > 0 \end{array} \right. \end{cases} \quad (4.23)$$

La démonstration de l'existence du régime de glissement avec cette logique de commutation est donnée en annexe 10.

IV.1.4.2 Simulations

Les réponses temporelles de l'entrée u_α et des courants statoriques du système sont donnés (fig.4.8a,b) avec $\omega_d=100$ rad/s et $T=10^{-5}$ s (période d'échantillonnage) :

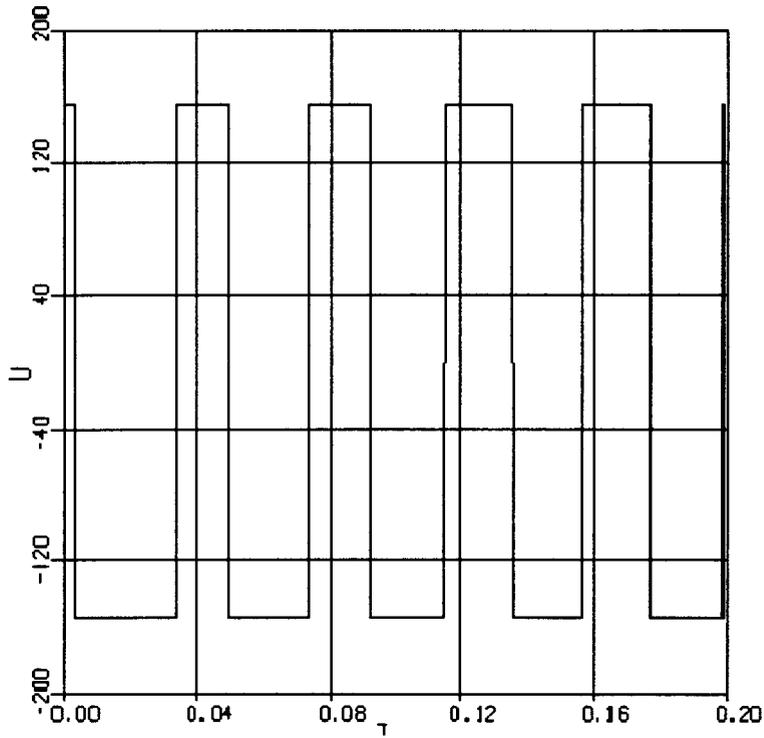


Fig.4.8a : Réponse temporelle de l'entrée u_α

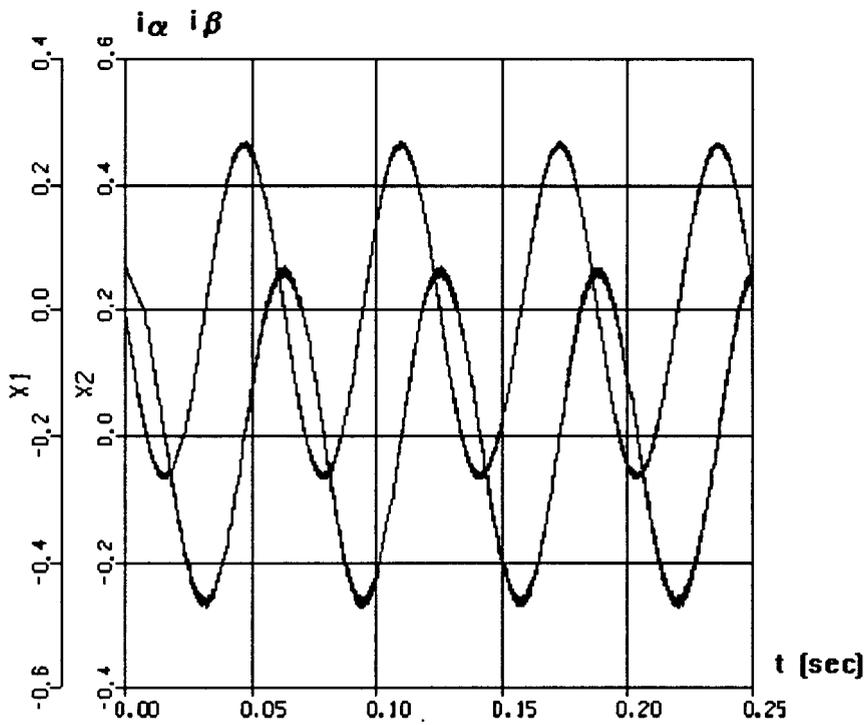


Fig.4.8b : Courants statoriques

La vitesse ainsi que l'erreur sont présentées (fig.4.9 et fig.4.10) :

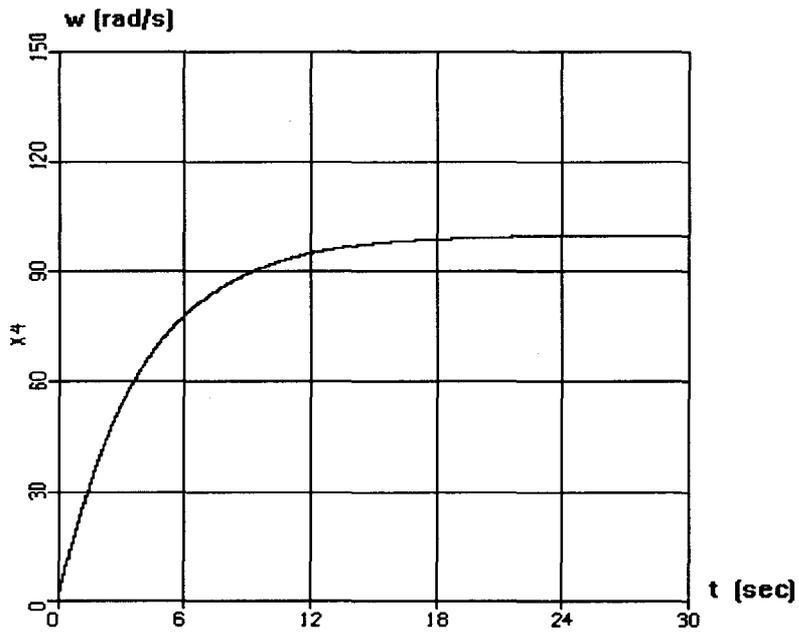


Fig.4.9 : Réponse temporelle de la vitesse.

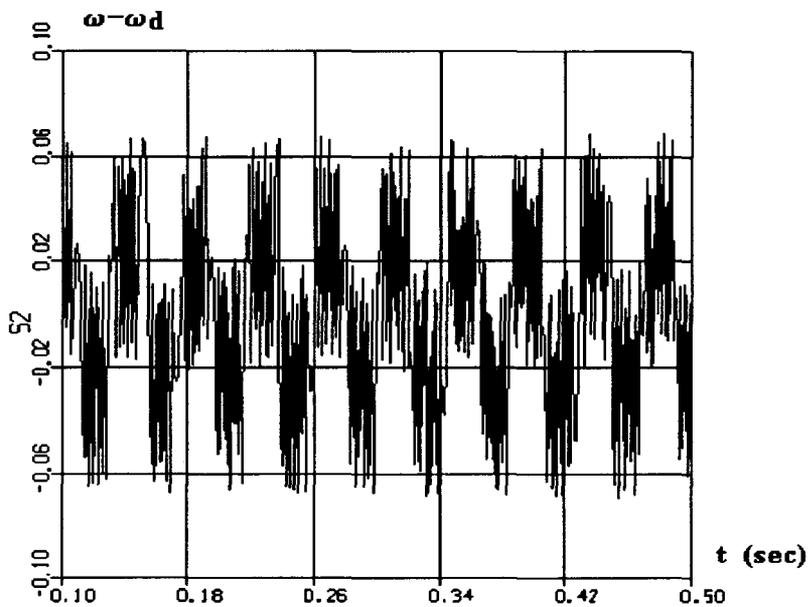


Fig.4.10 : Erreur en vitesse (régime établi).

IV.1.4.3 Conclusion

Cet algorithme appelle plusieurs remarques :



1) L'hypothèse que la vitesse est lentement variable est nécessaire pour pouvoir garder expressions simples dans les calculs de la commande.

2) La commande appliquée sur le système non-linéaire donne des performances acceptables, mais on ne peut pas envisager de poursuite de trajectoire en utilisant la même procédure. Pour cela il faudrait utiliser le modèle non-linéaire de départ (dimension 4 non-linéaire multivariable) pour lequel les surfaces de glissement sont plus difficiles à déterminer.

IV.1.5 Commande par linéarisations par morceaux

IV.1.5.1 Elaboration de l'algorithme

Le système non-linéaire est décrit (voir (4.5)) par :

$$\frac{dx}{dt} = \begin{pmatrix} -\frac{1}{\tau} x_1 + ax_4 \sin x_3 \\ -\frac{1}{\tau} x_2 - ax_4 \cos x_3 \\ x_4 \\ -\frac{1}{\tau_m} x_4 + bx_2 \cos x_3 - bx_1 \sin x_3 - \gamma_r \end{pmatrix} + \begin{pmatrix} \frac{2E_0}{L\sqrt{6}} & \frac{-E_0}{L\sqrt{6}} & \frac{-E_0}{L\sqrt{6}} \\ 0 & \frac{E_0}{L\sqrt{2}} & \frac{-E_0}{L\sqrt{2}} \\ 0 & 0 & 0 \\ 0 & 0 & 0 \end{pmatrix} \begin{pmatrix} u_1 \\ u_2 \\ u_3 \end{pmatrix} \quad (4.24)$$

Nous allons supposer que nous voulons commander la machine en vitesse vers une consigne variable $\omega_d(t)$.

Pour linéariser le modèle, nous allons découper le domaine temporel en intervalles $I_k = [kT_d, (k+1)T_d]$, k entier, et nous allons faire l'hypothèse que la vitesse est constante sur cet intervalle, égale à $\omega_d(kT_d)$.

On obtient les modèles, valables sur I_k :

$$\frac{dX}{dt} = \begin{pmatrix} -\frac{1}{\tau} X_1 \\ -\frac{1}{\tau} X_2 \end{pmatrix} + \begin{pmatrix} a\omega_d(kT_d) \sin \omega_d(kT_d)t \\ -a\omega_d(kT_d) \cos \omega_d(kT_d)t \end{pmatrix} + \begin{pmatrix} \frac{2E_0}{L\sqrt{6}} & \frac{-E_0}{L\sqrt{6}} & \frac{-E_0}{L\sqrt{6}} \\ 0 & \frac{E_0}{L\sqrt{2}} & \frac{-E_0}{L\sqrt{2}} \end{pmatrix} \begin{pmatrix} u_1 \\ u_2 \\ u_3 \end{pmatrix} \quad (4.25)$$

avec $X = (X_1, X_2)^T = (x_1, x_2)^T$.

On remarque que le modèle obtenu n'est pas stationnaire.

Considérons alors les courants rotoriques sinusoïdaux solutions du système:

$$\begin{cases} \frac{dz_3}{dt} = \omega_d(kT_d)z_4 \\ \frac{dz_4}{dt} = -\omega_d(kT_d)z_3 \end{cases} \quad (4.26)$$

avec $z_3(0)=0$ et $z_4(0)=a$.

En considérant $Z=(X_1, X_2, z_3, z_4)^T$ comme vecteur d'état du système, il vient :

$$\frac{dZ}{dt} = \begin{pmatrix} \frac{-1}{\tau} & 0 & \omega_d(kT_d) & 0 \\ 0 & \frac{-1}{\tau} & 0 & -\omega_d(kT_d) \\ 0 & 0 & 0 & \omega_d(kT_d) \\ 0 & 0 & -\omega_d(kT_d) & 0 \end{pmatrix} Z + \begin{pmatrix} \frac{2E_0}{L\sqrt{6}} & \frac{-E_0}{L\sqrt{6}} & \frac{-E_0}{L\sqrt{6}} \\ 0 & \frac{E_0}{L\sqrt{2}} & \frac{-E_0}{L\sqrt{2}} \\ 0 & 0 & 0 \\ 0 & 0 & 0 \end{pmatrix} \begin{pmatrix} u_1 \\ u_2 \\ u_3 \end{pmatrix} \quad (4.27)$$

qui peut être écrit :

$$\frac{dZ}{dt} = \mathbf{A}(kT_d)Z + \mathbf{B}u \quad (4.28)$$

Nous avons vu précédemment que pour faire converger la vitesse du moteur vers $\omega_d(kT_d)$, il était suffisant de faire converger les courants statoriques vers :

$$\begin{cases} x_{1d} = -i_0 \sin(\omega_d(kT_d)t) \\ x_{2d} = i_0 \cos(\omega_d(kT_d)t) \end{cases}$$

avec $\omega_d = b\tau_m i_0 - \gamma_r \tau_m$.

La sortie du système est donc choisie telle que :

$$y = \begin{pmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \end{pmatrix} X \quad (4.29)$$

et la consigne désirée :

$$y_d = \begin{pmatrix} y_{d1} \\ y_{d2} \end{pmatrix} = \begin{pmatrix} -i_0 \sin(\omega_d(kT_d)t) \\ i_0 \cos(\omega_d(kT_d)t) \end{pmatrix}$$

Il faut alors calculer $y_{i\infty}$, pour $i=1, \dots, 8$, tels que

$$y_{i\infty} = -\mathbf{CA}^{-1}\mathbf{B}[\text{config}(u)]_i$$

On a alors :

$$\begin{aligned}
 y_{1\infty}(kT_d) &= -\mathbf{CA}^{-1}(kT_d)\mathbf{B} \begin{pmatrix} 0 \\ 0 \\ 0 \end{pmatrix} = \begin{pmatrix} 0 \\ 0 \end{pmatrix} & y_{2\infty}(kT_d) &= -\mathbf{CA}^{-1}(kT_d)\mathbf{B} \begin{pmatrix} 1 \\ 0 \\ 0 \end{pmatrix} = \frac{E_0}{L\sqrt{6}} \begin{pmatrix} 2\tau \\ 0 \end{pmatrix} \\
 y_{3\infty}(kT_d) &= -\mathbf{CA}^{-1}(kT_d)\mathbf{B} \begin{pmatrix} 0 \\ 1 \\ 0 \end{pmatrix} = \frac{E_0}{L\sqrt{6}} \begin{pmatrix} -\tau \\ \sqrt{3}\tau \end{pmatrix} & y_{4\infty}(kT_d) &= -\mathbf{CA}^{-1}(kT_d)\mathbf{B} \begin{pmatrix} 0 \\ 0 \\ 1 \end{pmatrix} = \frac{-E_0}{L\sqrt{6}} \begin{pmatrix} \tau \\ \sqrt{3}\tau \end{pmatrix} \\
 y_{5\infty}(kT_d) &= -\mathbf{CA}^{-1}(kT_d)\mathbf{B} \begin{pmatrix} 1 \\ 1 \\ 0 \end{pmatrix} = \frac{E_0}{L\sqrt{6}} \begin{pmatrix} \tau \\ \sqrt{3}\tau \end{pmatrix} & y_{6\infty}(kT_d) &= -\mathbf{CA}^{-1}(kT_d)\mathbf{B} \begin{pmatrix} 1 \\ 0 \\ 1 \end{pmatrix} = \frac{-E_0}{L\sqrt{6}} \begin{pmatrix} -\tau \\ \sqrt{3}\tau \end{pmatrix} \\
 y_{7\infty}(kT_d) &= -\mathbf{CA}^{-1}(kT_d)\mathbf{B} \begin{pmatrix} 0 \\ 1 \\ 1 \end{pmatrix} = \frac{-E_0}{L\sqrt{6}} \begin{pmatrix} 2\tau \\ 0 \end{pmatrix} & y_{8\infty}(kT_d) &= -\mathbf{CA}^{-1}(kT_d)\mathbf{B} \begin{pmatrix} 1 \\ 1 \\ 1 \end{pmatrix} = \begin{pmatrix} 0 \\ 0 \end{pmatrix}
 \end{aligned} \tag{4.30}$$

On remarque que les points-limites des systèmes linéarisés ne dépendent pas de kT_d . On a donc un domaine de convergence fixe, comme dans le cas du linéaire qui peut être schématisé par (fig.4.11):

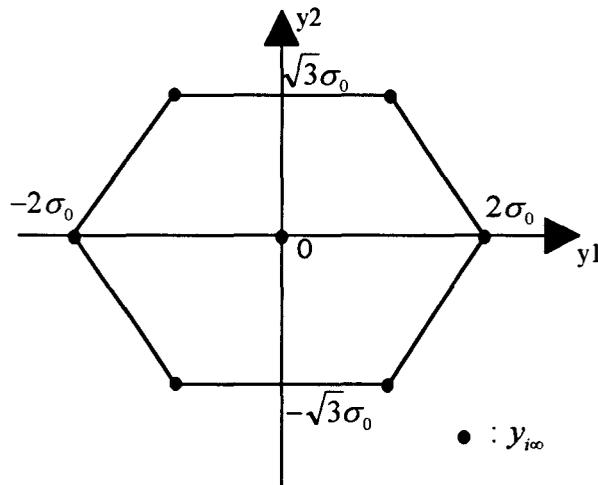


Fig.4.11 : Domaine de convergence du vecteur sortie.

$$\text{en posant } \sigma_0 = \frac{E_0 \tau}{L\sqrt{6}}.$$

On applique l'algorithme de poursuite de trajectoire en boucle fermée décrit dans le chapitre III, à la commande de la machine synchrone.

Les trajectoires de référence $y_r=(y_{r1},y_{r2})^T$ sont construites en utilisant la procédure décrite en 3.10 du chapitre III. Les coefficients $a_i(kT_d)$, $i=1,\dots,2^m$, k entiers, sont calculés sur chaque intervalle $[kT_d,(k+1)T_d]$ en utilisant (voir chapitre III) :

$$\begin{cases} y_r(kT_d) = \sum_{i=1}^{2^m} a_i(kT_d) y_{i\infty}(kT_d) \\ \sum_{i=1}^{2^m} a_i(kT_d) = 1 \\ a_i(kT_d) \in [0,1] \end{cases}$$

L'algorithme de commande que l'on applique peut alors être schématisé (fig.4.12) :

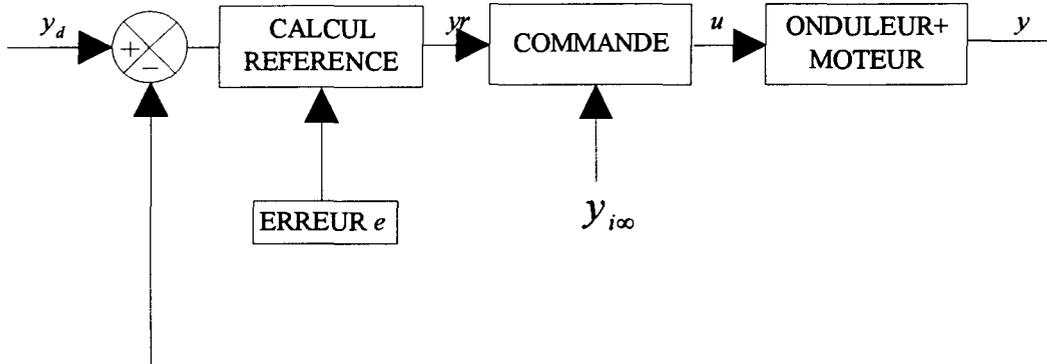


Fig.4.12 : Algorithme de commande d'un moteur synchrone

IV.1.5.2 Amélioration de l'algorithme

La représentation précédente montre que l'on contrôle l'erreur sur les courants mais pas directement sur la vitesse. Une méthode permettant de remédier à cela est d'utiliser l'erreur en vitesse dans le schéma suivant (fig.4.13) :

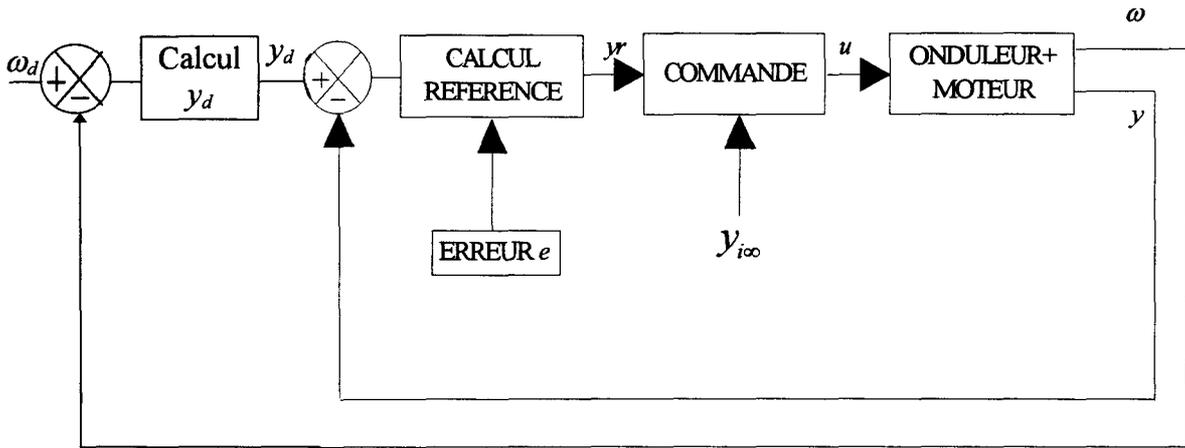


Fig.4.13 : Algorithme de commande d'un moteur synchrone avec compensation.

avec le bloc calcul de y_d tel que :

$$y_d = \begin{pmatrix} y_{d_1} \\ y_{d_2} \end{pmatrix} = \begin{pmatrix} -i_0 \sin((\omega_d(kT_d) - \omega)t) \\ i_0 \cos((\omega_d(kT_d) - \omega)t) \end{pmatrix}$$

IV.1.5.3 Simulations

Nous avons simulé le système électrique composé d'un onduleur de tension associé à un moteur synchrone avec les paramètres du §IV.1.3.3.

IV.1.5.3.1 Simulation 1

On impose une consigne $\omega_d=100$ rad/s avec $T_d=10^{-1}s$, $T_{mod}=10^{-3}s$, $T=10^{-5}s$. Les courants et la vitesse ont été simulés en utilisant l'algorithme avec compensation (fig.4.14a,b) :

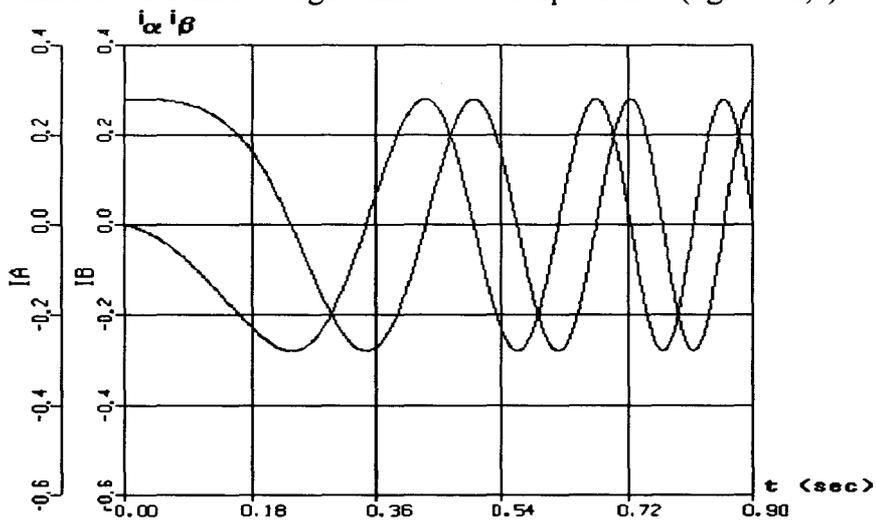


Fig.4.14a : Courants de la machine synchrone

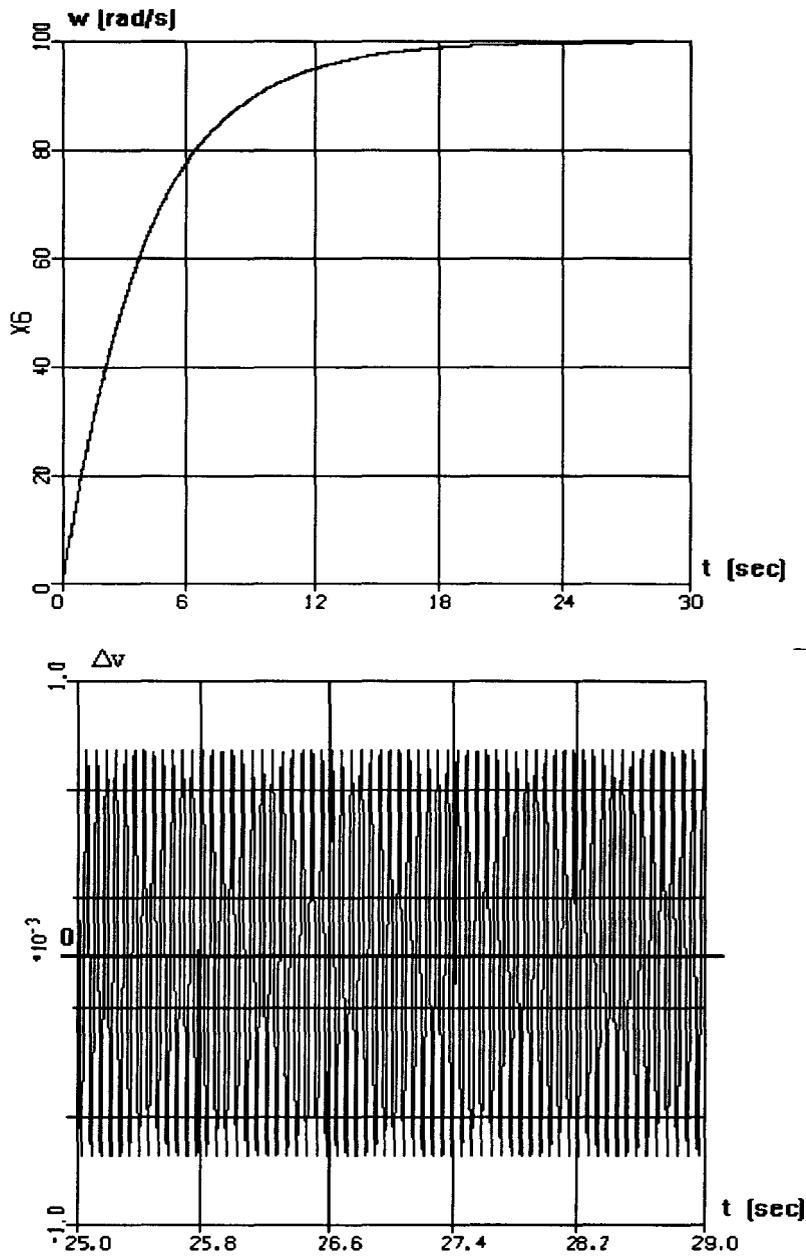


Fig.4.14b : Vitesse et erreur en vitesse en régime établi.

IV.1.5.3.2 Simulation 2

On impose une consigne variable telle que :

$\omega_d(t)=t$ rad/s pour $t < 100s$, $\omega_d(t)=100$ rad/s pour $100 < t < 150s$, $\omega_d(t)=-t+250$ rad/s pour $150 < t < 200s$, $\omega_d(t)=50$ rad/s pour $t > 200s$, avec les mêmes valeurs pour les paramètres d'échantillonnage.

Le système a été simulé en utilisant l'algorithme avec compensation (fig.4.15) :

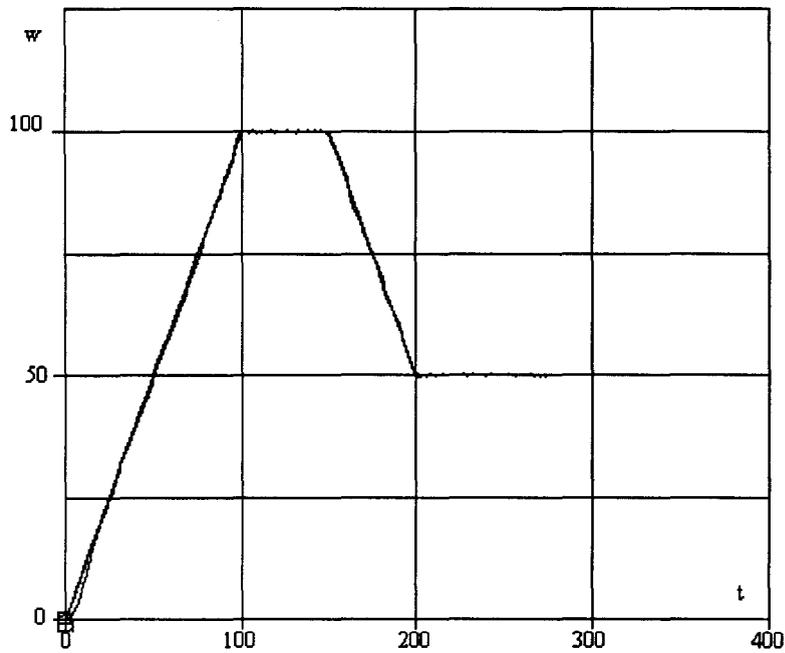


Fig.4.15 : Poursuite de trajectoire

IV.1.5.4 Conclusion

Ces simulations appellent plusieurs remarques :

- 1) Le domaine de convergence de la sortie est connu ce qui donne des informations sur les consignes que l'on peut appliquer au système.
- 2) Contrairement à l'algorithme de linéarisation par retour d'état, la commande est toujours définie et bornée ce qui facilite l'implantation de la partie calcul de la commande.
- 3) La poursuite de trajectoire peut être effectuée directement en utilisant la même forme d'algorithme.

Un tableau de comparaison des différents algorithmes est donné dans la partie III de ce chapitre.

IV.2 COMMANDE DE LA MACHINE ASYNCHRONE

Nous allons étudier dans cette partie la commande en vitesse d'une machine asynchrone bipolaire associée, comme précédemment, à un onduleur de tension alimenté par une source de tension continue.

IV.2.1 Modélisation par bond-graph

IV.2.1.1 Bond-graph du système

L'association onduleur-moteur asynchrone que nous allons étudier peut être modélisée par bond-graph [Ducreux et al, 1992], (fig.4.16) :

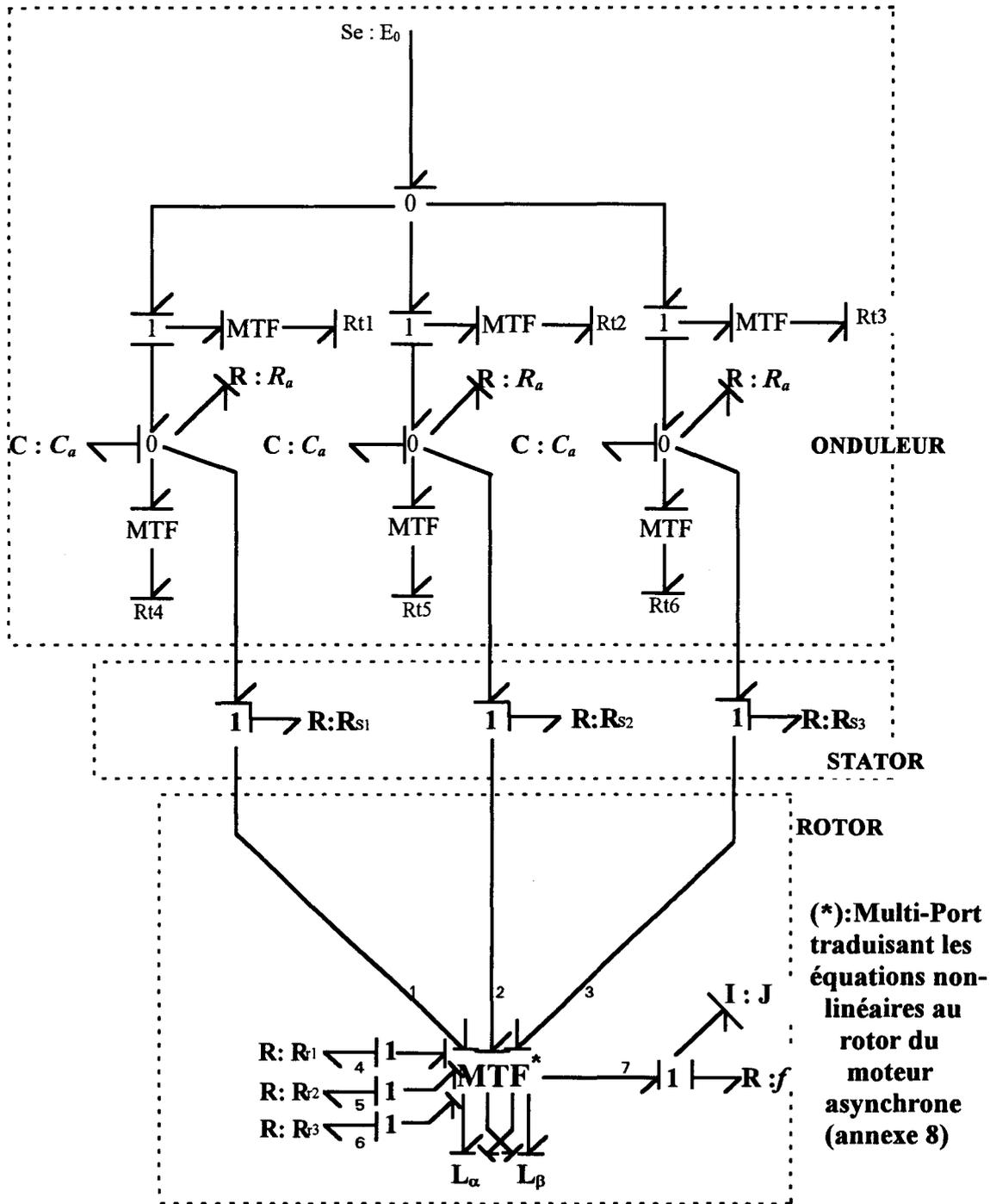


Fig.4.16 : Bond-graph d'une association onduleur moteur asynchrone.

Remarque :

Les bond-graphs pour la machine asynchrone et synchrone diffèrent par les équations non-linéaires qui décrivent les MTF(*), et sont identiques pour le reste du modèle.

IV.2.1.2 Equation d'état

Comme pour la machine synchrone, la constante de temps associée aux blocs R_a , C_a est supposée petite, ce qui permet d'éliminer des équations d'état la charge de ces condensateurs (voir machine synchrone, annexe 7). Le bloc MTF décrit la transformation de Park effectuée au stator. Les équations sont décrites dans l'annexe 8. Nous avons pris les flux statoriques et rotoriques et la vitesse du moteur comme variables d'état pour obtenir un modèle plus simple :

$$\begin{cases} \frac{d\phi_{s\alpha}}{dt} = -\alpha_s \phi_{s\alpha} - \beta_s \phi_{r\alpha} + u_\alpha \\ \frac{d\phi_{s\beta}}{dt} = -\alpha_s \phi_{s\beta} - \beta_s \phi_{r\beta} + u_\beta \\ \frac{d\phi_{r\alpha}}{dt} = -\alpha_r \phi_{r\alpha} + \beta_r \phi_{s\alpha} + \omega \phi_{r\beta} \\ \frac{d\phi_{r\beta}}{dt} = -\alpha_r \phi_{r\beta} + \beta_r \phi_{s\beta} - \omega \phi_{r\alpha} \\ \frac{d\omega}{dt} = -\frac{f}{J} \omega + \frac{K}{J} (\phi_{r\alpha} \phi_{s\beta} - \phi_{s\alpha} \phi_{r\beta}) \end{cases} \quad (4.31)$$

$\phi_{s\alpha}$, $\phi_{s\beta}$ sont les flux statoriques, d'axes $(\alpha-\beta)$, $\phi_{r\alpha}$, $\phi_{r\beta}$ sont les flux rotoriques d'axe $(\alpha-\beta)$ et ω est la vitesse de la machine asynchrone et u_α , u_β telles que :

$$\begin{cases} u_\alpha = \frac{E_0}{\sqrt{6}} (2u_1 - u_2 - u_3) \\ u_\beta = \frac{E_0}{\sqrt{2}} (u_2 - u_3) \end{cases} \quad (4.32)$$

Les paramètres peuvent s'exprimer en fonction des grandeurs électriques :

$$\alpha_s = \frac{R_s L}{L^2 - M^2}, \quad \alpha_r = \frac{R_r L}{L^2 - M^2}$$

$$\beta_s = \frac{R_s M}{L^2 - M^2}, \quad \beta_r = \frac{R_r M}{L^2 - M^2}$$

$$K = M(L^2 - M^2)$$

$$M = \frac{3}{2} m_{sr}, L = L_s - M_s, L = L_r - M_r \quad (4.33)$$

Signification des paramètres électriques:

R_s, R_r : résistances statoriques et rotoriques ; L_s, L_r : Self-inductances statoriques et rotoriques
 M_s, M_r : Inductances mutuelles statoriques et rotoriques ; m_{sr} : Inductance mutuelle entre stator et rotor ; E_0 : Tension d'alimentation de l'onduleur ; f, J : Frottement et inertie du rotor.

IV.2.2 Commande par linéarisations successives

IV.2.2.1 Modèles linéarisés

Pour linéariser le système, nous allons supposer que la vitesse du moteur est telle que ω est constant par morceaux avec, sur $[kT_d, (k+1)T_d]$: $\omega(kT_d) = \omega_d(kT_d)$.

alors :

$$\begin{cases} \frac{d\phi_{s\alpha}}{dt} = -\alpha_s \phi_{s\alpha} - \beta_s \phi_{r\alpha} + \frac{E_0}{\sqrt{6}} (2u_1 - u_2 - u_3) \\ \frac{d\phi_{s\beta}}{dt} = -\alpha_s \phi_{s\beta} - \beta_s \phi_{r\beta} + \frac{E_0}{\sqrt{2}} (u_2 - u_3) \\ \frac{d\phi_{r\alpha}}{dt} = -\alpha_r \phi_{r\alpha} + \beta_r \phi_{s\alpha} + \omega_d(kT_d) \phi_{r\beta} \\ \frac{d\phi_{r\beta}}{dt} = -\alpha_r \phi_{r\beta} + \beta_r \phi_{s\beta} - \omega_d(kT_d) \phi_{r\alpha} \end{cases} \quad (4.34)$$

et la vitesse est régie indépendamment par :

$$0 = -\frac{f}{J} \omega_d(kT_d) + \frac{K}{J} (\phi_{r\alpha} \phi_{s\beta} - \phi_{s\alpha} \phi_{r\beta}) \quad (4.35)$$

L'équation d'état linéarisée peut donc s'exprimer par :

$$\dot{x}(t + kT_d) = \mathbf{A}(kT_d)x(t + kT_d) + \mathbf{B}u \quad (4.36)$$

pour $t \in [0, T_d]$, et $x = (\phi_{s\alpha}, \phi_{s\beta}, \phi_{r\alpha}, \phi_{r\beta})^T$ et $u = (u_1, u_2, u_3)^T$

avec :

$$\mathbf{A}(kT_d) = \begin{pmatrix} -\alpha_s & 0 & -\beta_s & 0 \\ 0 & -\alpha_s & 0 & -\beta_s \\ \beta_r & 0 & -\alpha_r & \omega_d(kT_d) \\ 0 & \beta_r & -\omega_d(kT_d) & -\alpha_r \end{pmatrix} \quad (4.37)$$

et

$$\mathbf{B} = \frac{E_0}{\sqrt{6}} \begin{pmatrix} 2 & -1 & -1 \\ 0 & \sqrt{3} & -\sqrt{3} \\ 0 & 0 & 0 \\ 0 & 0 & 0 \end{pmatrix} \quad (4.38)$$

Sur chaque $[kT_d, (k+1)T_d]$, la vitesse est constante, soit :

$$0 = -\frac{f}{J} \omega_d(kT_d) + \frac{K}{J} (\phi_{r\alpha} \phi_{s\beta} - \phi_{s\alpha} \phi_{r\beta}) \quad (4.39)$$

IV.2.2.2 Commande en vitesse

On peut démontrer (voir Annexe 8), que pour commander la vitesse du système vers ω_d , il est suffisant de commander les flux statoriques de sorte que :

$$\begin{aligned} \phi_{s\alpha} &= \phi_N \sin(\omega_d t) \\ \phi_{s\beta} &= \phi_N \cos(\omega_d t) \end{aligned} \quad (4.40)$$

ce qui entraîne au rotor :

$$\begin{aligned} \phi_{r\alpha} &= \phi_f \sin(\omega_d t) \\ \phi_{r\beta} &= -\phi_f \cos(\omega_d t) \end{aligned} \quad (4.41)$$

avec : $\phi_f = \frac{\beta_r \phi_N}{\alpha_r}$, et $\phi_N = \sqrt{\frac{f \alpha_r}{K \beta_r}} \omega_d$

Nous devons donc commander, comme pour la machine synchrone les flux statoriques de manière sinusoïdale. On choisit alors :

$$y = \begin{pmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \end{pmatrix} x$$

IV.2.2.3 Calcul des points limites

L'étape suivante est de calculer les $y_{i\infty}(kT_d)$ pour $i=1,\dots,2^m$. Pour cela, on utilise la formule élaborée au chapitre III :

$$y_{i\infty}(kT_d) = -\mathbf{CA}^{-1}(kT_d)\mathbf{B}[\text{config}(u)]_i$$

avec pour tout i , $[\text{config}(u)]_i \in \text{config}(u)$.

Il vient alors :

$$y_{2\infty}(kT_d) = -\mathbf{CA}^{-1}\mathbf{B} \begin{pmatrix} 1 \\ 0 \\ 0 \end{pmatrix} = \frac{2E_0}{d\sqrt{6}} \begin{pmatrix} \alpha_r\delta + \alpha_s\omega_d^2(kT_d) \\ \beta_r\beta_s\omega_d(kT_d) \end{pmatrix}$$

$$y_{3\infty}(kT_d) = -\mathbf{CA}^{-1}\mathbf{B} \begin{pmatrix} 0 \\ 1 \\ 0 \end{pmatrix} = \frac{E_0}{d\sqrt{6}} \begin{pmatrix} -\beta_r\beta_s\sqrt{3}\omega_d(kT_d) - \alpha_r\delta - \alpha_s\omega_d^2(kT_d) \\ -\beta_r\beta_s\omega_d(kT_d) + \sqrt{3}(\alpha_r\delta + \alpha_s\omega_d^2(kT_d)) \end{pmatrix}$$

$$y_{4\infty}(kT_d) = -\mathbf{CA}^{-1}\mathbf{B} \begin{pmatrix} 0 \\ 0 \\ 1 \end{pmatrix} = \frac{E_0}{d\sqrt{6}} \begin{pmatrix} \beta_r\beta_s\sqrt{3}\omega_d(kT_d) - (\alpha_r\delta + \alpha_s\omega_d^2(kT_d)) \\ -\beta_r\beta_s\omega_d(kT_d) - \sqrt{3}(\alpha_r\delta + \alpha_s\omega_d^2(kT_d)) \end{pmatrix}$$

et :

$$\begin{cases} y_{5\infty} = y_{2\infty} + y_{3\infty} \\ y_{6\infty} = y_{2\infty} + y_{4\infty} \\ y_{7\infty} = y_{3\infty} + y_{4\infty} \\ y_{8\infty} = y_{1\infty} = \begin{pmatrix} 0 \\ 0 \end{pmatrix} \end{cases}$$

(4.42)

avec $\delta = \alpha_s\alpha_r + \beta_s\beta_r$ et $d = \det(\mathbf{A}) = \delta^2 + \omega_d^2(kT_d)$.

L'algorithme ainsi mis en place peut alors être schématisé (fig.4.17) :

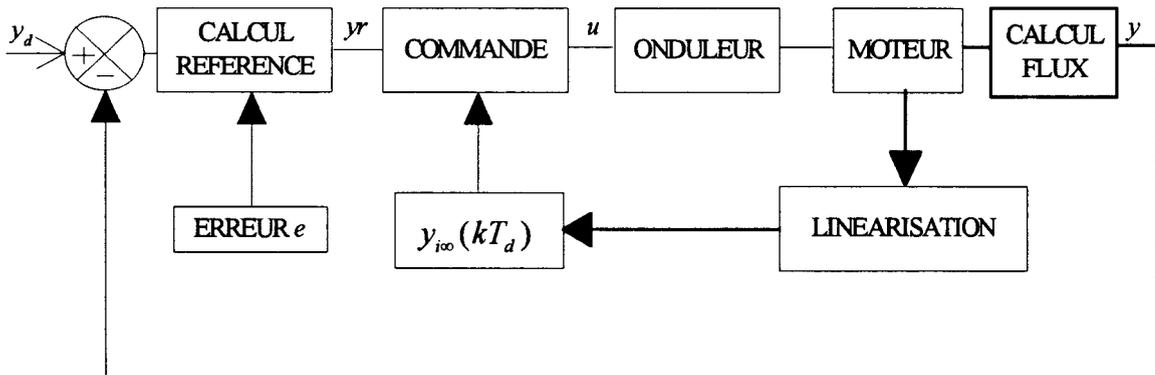


Fig.4.17 : Algorithme de commande d'un moteur asynchrone (sans compensation).

IV.2.2.4 Problème du glissement

Pour le système réel, on observe l'apparition d'une pulsation de glissement qui nous oblige à définir un algorithme de commande avec compensation.

Il faut donc imposer comme consigne pour les flux statoriques :

$$\begin{cases} \phi_{s\alpha} = y_{d1} = \phi_N \sin((\omega_d(kT_d) + \bar{\omega}_r(kT_d))t) \\ \phi_{s\beta} = y_{d2} = \phi_N \cos((\omega_d(kT_d) + \bar{\omega}_r(kT_d))t) \end{cases}$$

avec $\bar{\omega}_r(kT_d)$, pulsation rotorique de glissement estimée sur le modèle du système.

L'algorithme ainsi mis en place peut alors être schématisé (fig.4.18) :

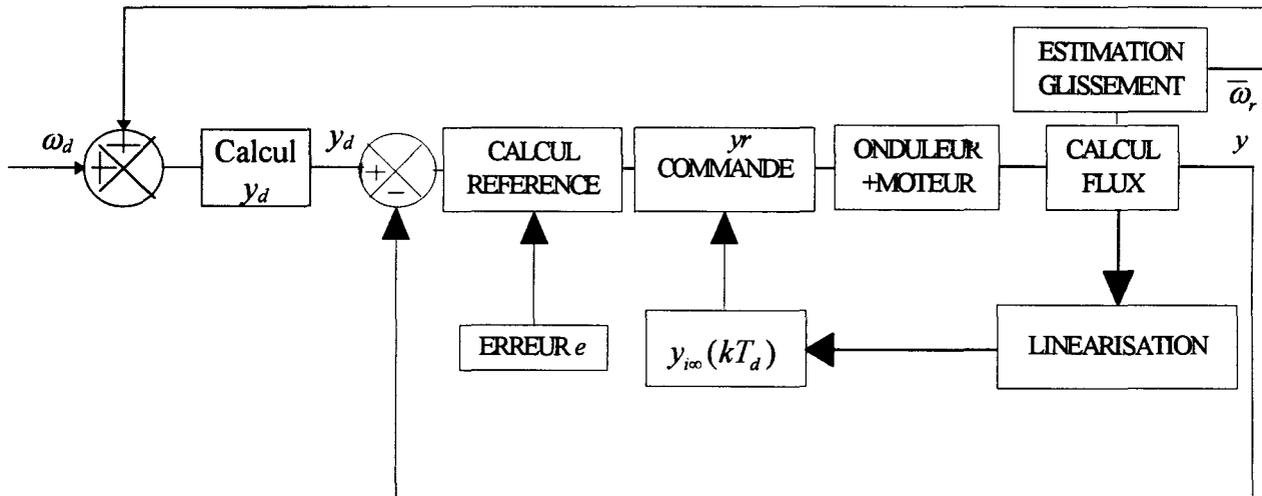


Fig.4.18 : Algorithme avec compensation de commande d'un moteur asynchrone .

Les simulations concernant cette partie sont présentées dans la partie IV.2.4.

IV.2.3 Commande par MLI

On applique sur le même système le même algorithme de commande en appliquant un mode de calcul de la commande booléenne par MLI. L'algorithme (avec compensation du glissement) devient (fig.4.19) :

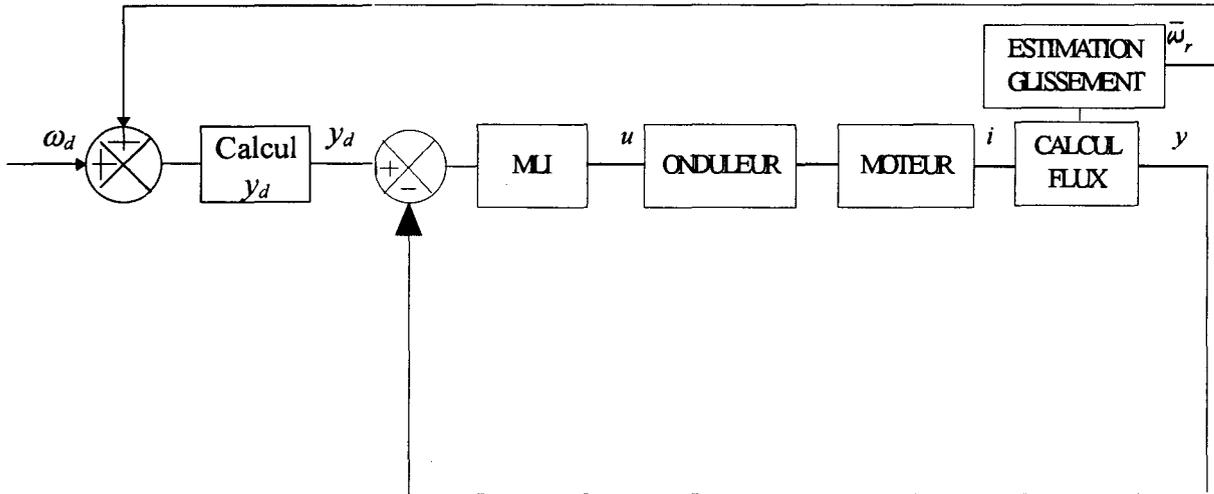


Fig.4.19 : Commande d'un moteur synchrone par MLI.

Le bloc MLI est décrit par la fig.4.20 :

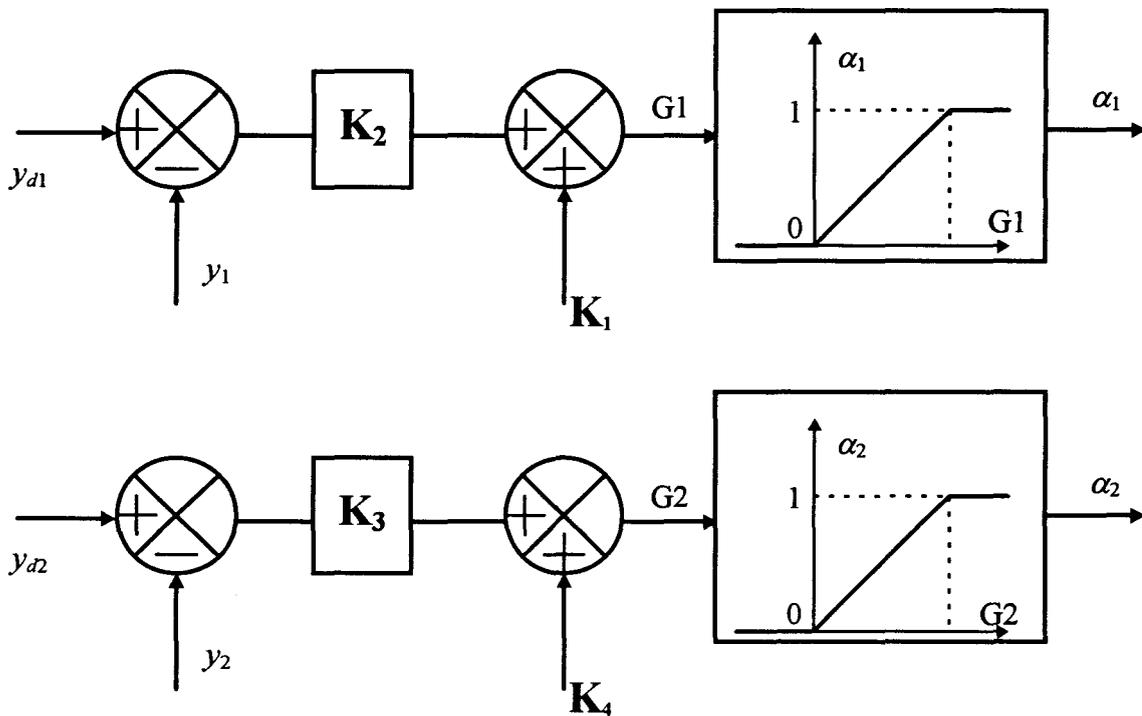


Fig.4.20 : Bloc de commande de l'onduleur par MLI.

α_1 et α_2 représentent les rapports cycliques pour les commutateurs T1 et T2 (voir chapitre I).

IV.2.4 Simulations

Nous avons considéré une machine asynchrone avec les paramètres suivants :

$$\begin{array}{ll}
 E_0=380 \text{ V} & M_{sr}=80 \text{ mH} \\
 R_s=R_r=0.1 \ \Omega & f=0.001 \text{ N.s} \\
 L_s=L_r=100 \text{ mH} & J=0.0025 \text{ kg.m}^2 \\
 & M_s=M_r=-50 \text{ mH}
 \end{array}$$

Les réponses du système électrique ont été simulées avec logiciel de simulation A.C.S.L.

IV.2.4.1 Méthode de linéarisations successives

Dans cette partie nous avons effectué des simulations en appliquant l'algorithme "avec compensation" décrit fig.4.19.

Les paramètres d'échantillonnage suivants ont été utilisés : $T_{mod}=10^{-4}$ s, et $T_d=10^{-2}$ s.

IV.2.4.1.1 Régulation

On désire commander le système en régulation vers $\omega_d=100$ rad/s. Les flux, la vitesse, et l'erreur en vitesse sont représentés fig.4.21a,b,c :

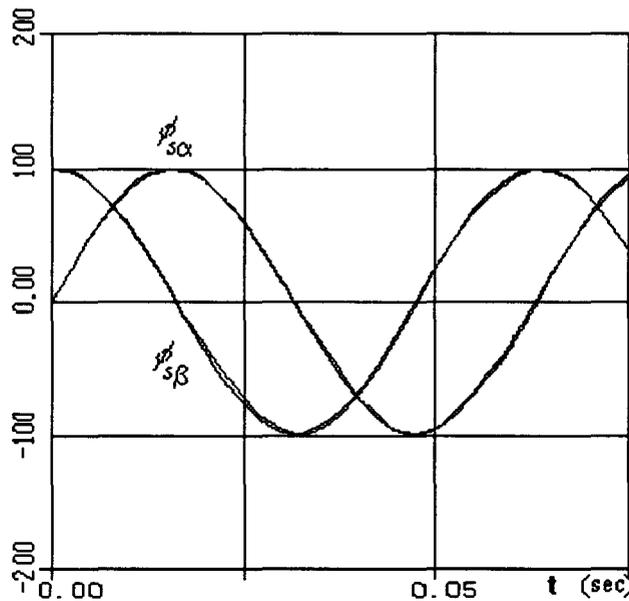


Fig.4.21a : Réponse temporelle des flux statoriques.

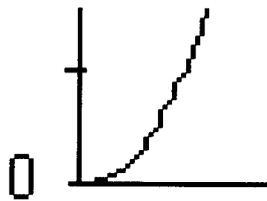
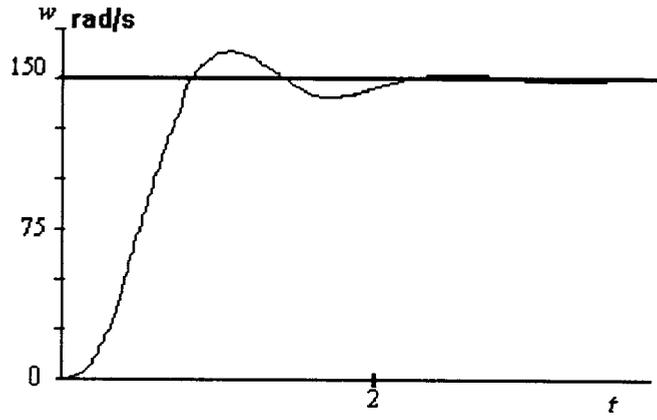


Fig.4.21b : Réponse temporelle de la vitesse et allure en 0.

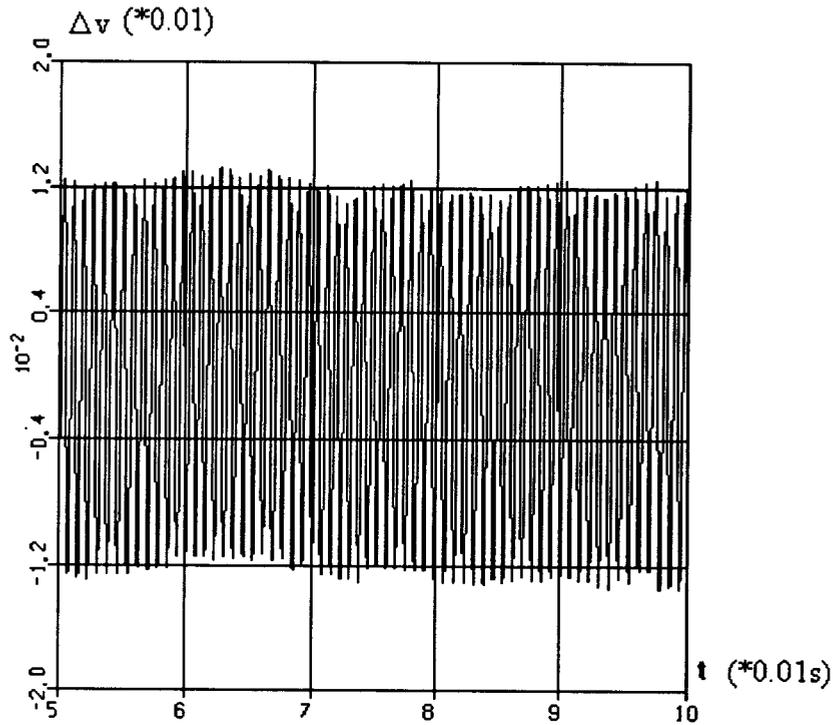


Fig.4.21c : Erreur en vitesse.



IV.2.4.1.2 Poursuite de trajectoires

On souhaite effectuer une poursuite de la trajectoire en vitesse définie par :

$$\begin{aligned} \omega_d(t) &= 10t \text{ rad/s pour } t < 10\text{s}, & \omega_d(t) &= 100 \text{ rad/s pour } 10 < t < 15\text{s}, \\ \omega_d(t) &= -10t + 250 \text{ rad/s pour } 15 < t < 20 \text{ s}, \\ \omega_d(t) &= 50 \text{ rad/s pour } t > 20\text{s}. \end{aligned}$$

La vitesse est représentée fig.4.22 :

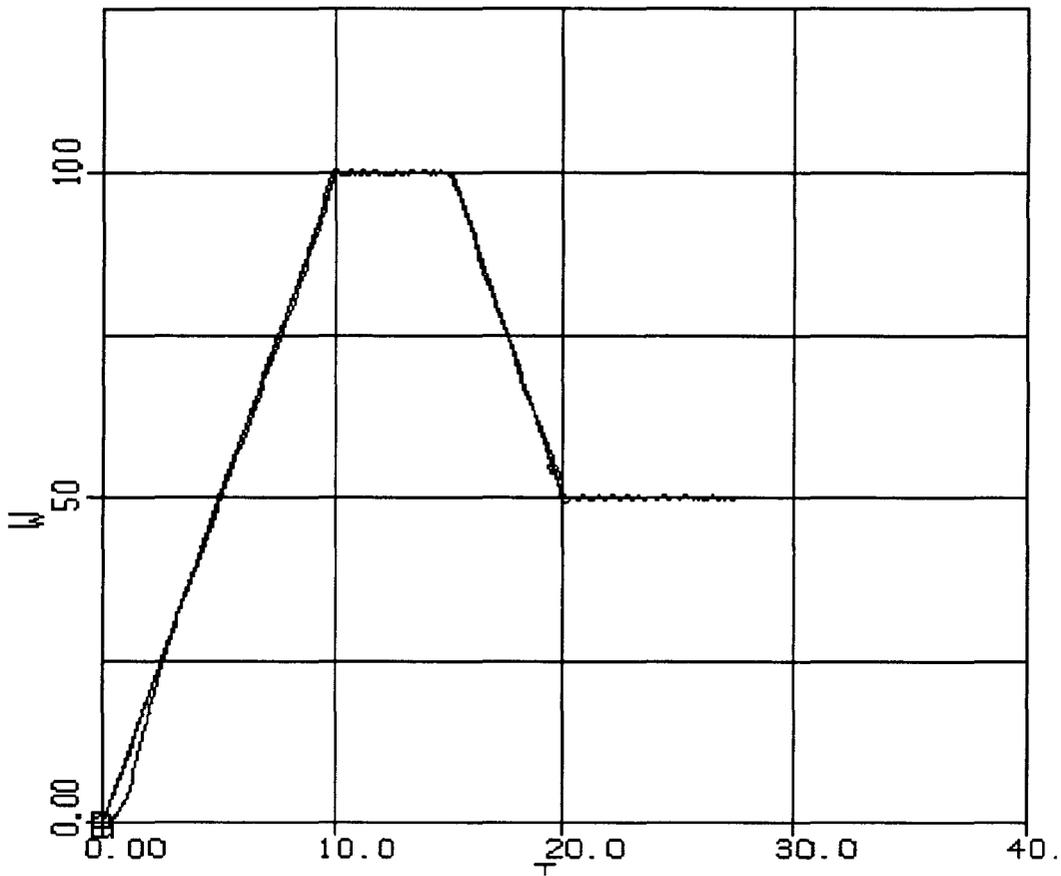


Fig.4.22 : Variation temporelle de la vitesse.

IV.2.4.2 Commande par MLI

IV.2.4.2.1 Régulation

On impose la même consigne constante $\omega_d=100$ rad/s. Les flux et la vitesse sont représentés fig.4.23 et fig.4.24. Les paramètres d'échantillonnage suivants ont été utilisés : $T_{mod}=10^{-5}$ s, et $T_d=10^{-3}$ s.

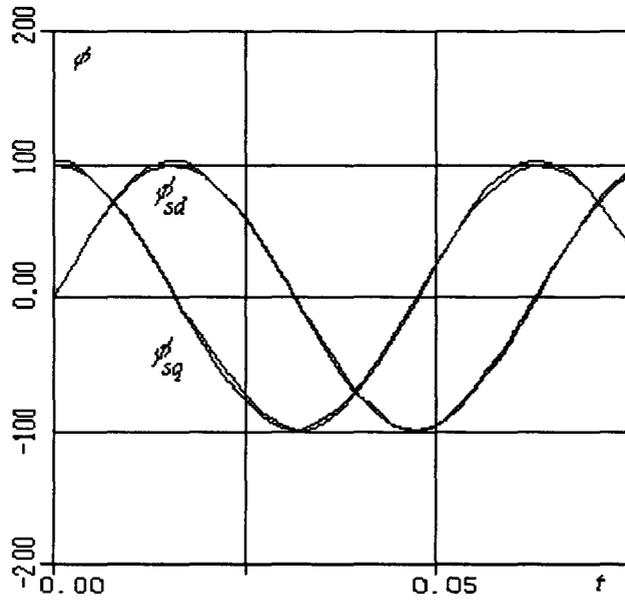


Fig.4.23 : Réponse temporelle des flux statoriques.

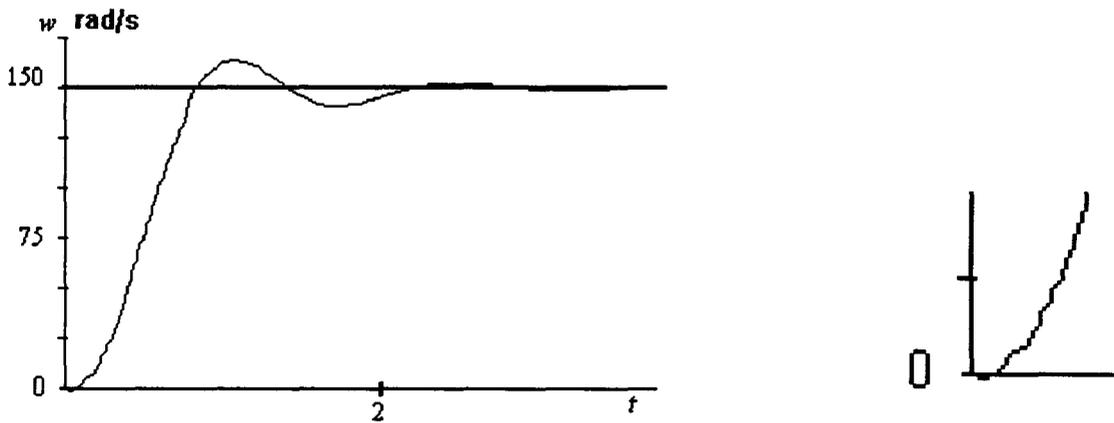


Fig.4.24 : Réponse temporelle de la vitesse et allure en 0.

L'erreur en vitesse est représentée fig.4.25 :

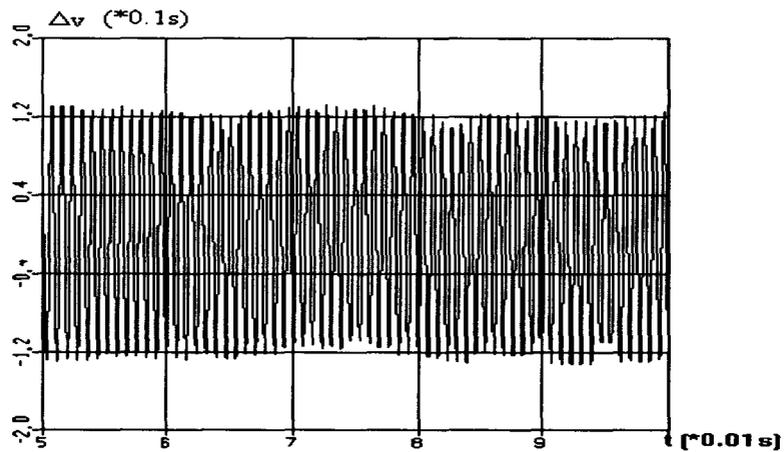


Fig. 4.25 : Erreur en vitesse.

IV.2.4.2 Poursuite de trajectoires

La trajectoire en vitesse est comme précédemment définie par : $\omega_d(t)=10t$ rad/s pour $t<10s$, $\omega_d(t)=100$ rad/s pour $10<t<15s$, $\omega_d(t)=-10t+250$ rad/s pour $15<t<20$ s, $\omega_d(t)=50$ rad/s pour $t>20s$.

La vitesse est représentée fig.4.26 :

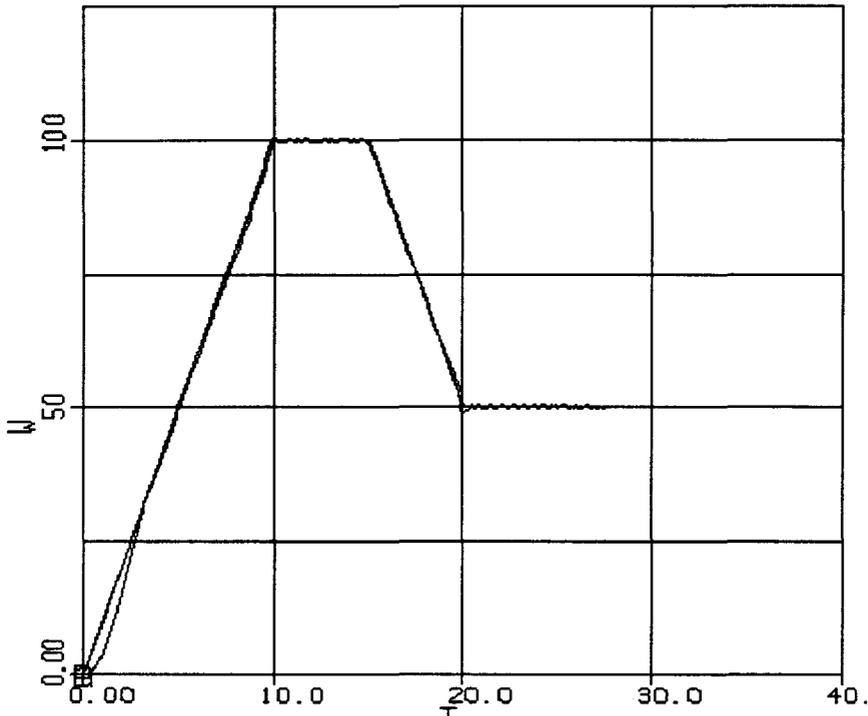


Fig.4.26 : Variation temporelle de la vitesse.

IV.2.4.3 Effet d'une variation des paramètres

Pour les machines asynchrones certains paramètres varient, lors de la phase de fonctionnement. C'est le cas, par exemple, de la résistance rotorique qui en chauffant, peut varier quelquefois de 50% de sa valeur nominale. Nous avons donc étudié l'effet des variations de certains paramètres sur le comportement du système commandé par l'algorithme que nous avons élaboré au chapitre III.

Nous avons étudié le cas de la régulation en reprenant les grandeurs identiques aux précédentes, et nous présentons les courbes de l'erreur en vitesse (régime établi) obtenues pour $R_r=(R_r)_{nom}+50\%=0.15\Omega$ (Fig.4.27) et $R_r=(R_r)_{nom}+100\%=0.20\Omega$ (Fig.4.28) :

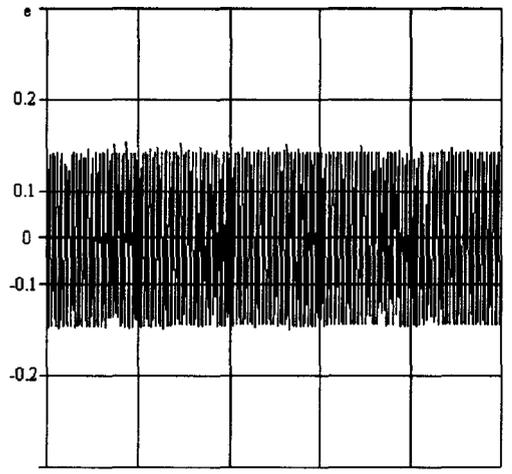


Fig.4.27 : Erreur en vitesse pour $R_r=0.15\Omega$.

Les oscillations sont dans ce cas plus importantes que dans le fonctionnement nominal, mais sont centrées sur 0 (pas d'erreur statique).

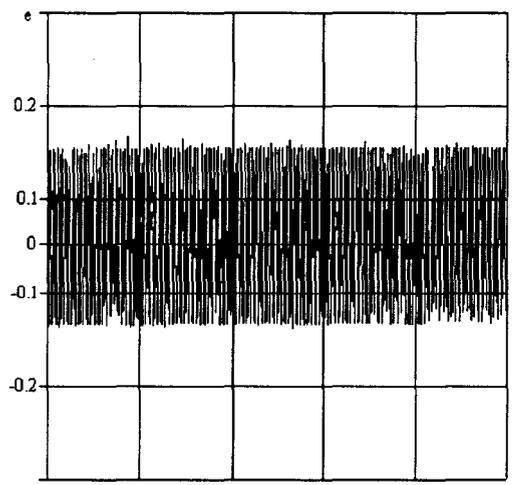


Fig.4.28 : Erreur en vitesse pour $R_r=0.20\Omega$.

Cette courbe montre que si le paramètre R_r est trop éloigné de sa valeur nominale, les performances du système se dégradent : l'amplitude des oscillations croît et celles-ci ne sont plus centrées sur l'origine.

IV.3 COMPARAISON DES RESULTATS

Dans cette partie nous élaborons un tableau comparatif des résultats fournis par les différents algorithmes pour la machine synchrone, puis pour la machine asynchrone.

IV.3.1 Moteur synchrone

MACHINE SYNCHRONE			
	Méthode 1 : Linéarisation par retour d'état	Méthode 2 : Modes glissants booléens	Méthode 3 : Linéarisations par morceaux
Régulation wd=cste			
Domaine de convergence	NON	NON	OUI
Période maximale d'échantillonnage (*)	0.01 ms	0.01ms	1 ms
Temps de réponse	Choisi arbitrairement (sans altérer la convergence)	Imposé par le système	Imposé par le système
Période maximale de discrétisation	1 ms		100 ms
Volume de calcul OFF Line	MOYEN	MOYEN	IMPORTANT
Volume de calcul ON LINE	IMPORTANT	FAIBLE	MOYEN
Nécessité d'estimateur	OUI	NON	NON
Poursuite de trajectoire : wd(t)			
Période maximale d'échantillonnage (*)	NON IMPLANTEE	DIFFICILE A IMPLANTER	0.1 ms
Temps de réponse	Choisi arbitrairement		Imposé par le système
Période maximale de discrétisation	NON IMPLANTEE		10 ms
Volume de calcul OFF Line	MOYEN		IMPORTANT
Volume de calcul ON LINE	IMPORTANT		MOYEN
Nécessité d'estimateur	OUI	NON	NON

(*) : période au delà de laquelle les performances du système sont altérées.

IV.3.2 Moteur asynchrone

Nous présentons le même type de tableau pour la machine asynchrone :

MACHINE ASYNCHRONE		
	Méthode 1 : Commande par MLI	Méthode 2: Linéarisations par morceaux
Régulation $wd=cste$		
Domaine de convergence	NON	OUI
Période maximale d'échantillonnage (*)	0.1ms	1 ms
Période maximale de discrétisation	10 ms	100 ms
Temps de réponse	Imposé par le système	Imposé par le système
Volume de calcul OFF Line	FAIBLE	IMPORTANT
Volume de calcul ON LINE	MOYEN	MOYEN
Nécessité d'estimateur	NON	NON
Poursuite de trajectoire : $wd(t)$		
Période maximale d'échantillonnage (*)	0.01ms	0.1 ms
Temps de réponse	Imposé par le système	Imposé par le système
Période maximale de discrétisation	1ms	10 ms
Volume de calcul OFF Line	FAIBLE	IMPORTANT
Volume de calcul ON LINE	MOYEN	MOYEN
Nécessité d'estimateur	NON	NON

(*) : période au delà de laquelle les performances du système sont altérées.

IV.4 IMPLANTATION SUR SITE

Des travaux sont en cours pour implanter l'algorithme de linéarisation par morceaux sur un moteur asynchrone. Cette entreprise est menée en collaboration avec l'équipe du Professeur Rombaut du LEEP à l'Ecole Centrale de Lille.

Le système est composé d'un onduleur de tension tel qu'il a été décrit au cours de ce chapitre. Les éléments de commutation sont des transistors.

Le système peut être représenté par (fig.4.29) :

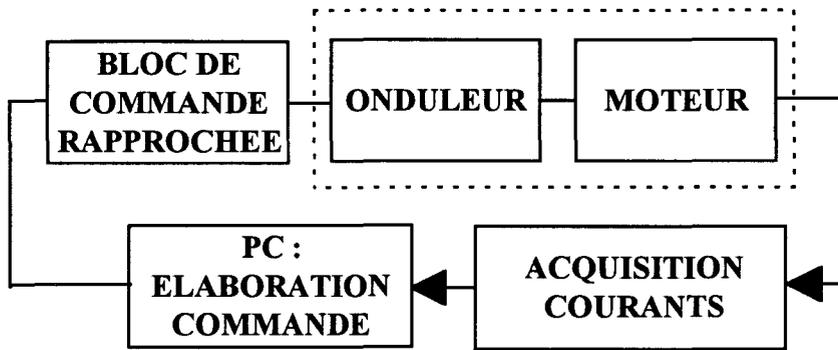


Fig.4.29 : Représentation du système réel.

Les paramètres du moteur asynchrone que nous étudions sont les suivants :

$$\begin{cases} L_s = L_r = 100\text{mH} ; M_{sr} = 80\text{mH} ; M_s = M_r = -50\text{mH} \\ R_s = R_r = 0.1\Omega ; R_t = 0.1\Omega ; R_a = 10k\Omega ; C_a < 1\mu F \\ f = 0.008\text{N.s} ; J = 0.02\text{kg.m}^2 \end{cases}$$

Les grandeurs mesurées sont les courants en ligne qu'il faut donc transformer par la relation de Park, pour pouvoir appliquer notre loi de commande.

L'algorithme de commande est implanté directement sur PC (langage C). Le bloc de commande rapprochée permet de commander à l'aide de compteurs l'ouverture et la fermeture des éléments de commutation.

Il est à noter que chaque élément de commutation dispose d'un temps minimal de conduction (différent de 0) et maximal (différent de 1). Notre algorithme doit donc être adapté en fonction de cela, ce qui fournit une contrainte supplémentaire dans le choix des coefficients α_i .

CONCLUSION

Nous avons appliqué en simulation les résultats des précédents chapitres à la commande de système électriques comportant des éléments de commutation. Nous avons tout d'abord traité le cas de la commande en vitesse d'un moteur synchrone associé à un onduleur de tension alimenté par une tension continue. Trois algorithmes différents ont été implantés : nous avons tout d'abord appliqué une commande fondée sur le principe de la linéarisation par retour d'état et sur une procédure de binarisation que nous avons élaboré au chapitre III. Nous avons ensuite appliqué une commande booléenne par modes glissants (chapitre I) puis l'algorithme de linéarisation par morceaux établi au chapitre III a été implanté. Les résultats ont révélé essentiellement que le dernier algorithme permettait un choix plus grand de la période

d'échantillonnage, qu'il était le seul des trois qui donnait des informations sur le domaine de convergence et qui permettait d'envisager d'effectuer de la poursuite de trajectoire.

Le cas de la commande en vitesse d'un moteur asynchrone a été ensuite envisagé. Nous avons implanté deux algorithmes différents : une commande par linéarisations par morceaux décrite dans le chapitre III, et une commande "classique" par MLI. Dans les deux cas, nous avons commandé le système vers une consigne constante (régulation), puis variable (poursuite de trajectoire). Les résultats ont montré que les deux algorithmes avaient des performances comparables quant aux volumes de calcul et les temps de réponses, mais que le premier permettait là encore, un choix plus élevé de la période d'échantillonnage et donnait des informations sur les valeurs admissibles de la consigne (domaine de convergence).

CONCLUSION GENERALE

CONCLUSION GENERALE

Nous avons traité au cours de ce mémoire, le problème de la commande booléenne des systèmes continus. Au cours du premier chapitre quelques rappels sur les méthodes pouvant se rapprocher de notre étude (MLI, hystérésis) ont été effectués. Nous avons ensuite étudié la possibilité d'adapter à notre cas une méthode plus générale de commande des systèmes à structure variable, la commande par modes glissants. Le cas des systèmes linéaires et bilinéaires a été envisagé dans le cas monoentrée et le cas multientrées. Cette étude a révélé qu'il était possible d'appliquer cette technique dans certains cas (systèmes de dimensions pas trop importantes).

Au cours du deuxième et du troisième chapitre nous avons élaboré de nouveaux algorithmes de commande pour les systèmes à entrées booléennes. Le cas linéaire a d'abord été envisagé par l'élaboration d'une loi de commande en boucle ouverte à l'intérieur d'un domaine déterminé (domaine de convergence). Nous avons ensuite proposer une utilisation des degrés de liberté apparaissant dans l'élaboration de l'algorithme de commande par rapport à des critères énergétiques ou pour éliminer des commutateurs. Un algorithme en boucle fermée permettant de réguler le système et de faire du suivi de trajectoire a alors été élaboré. Ces résultats ont ensuite été étendus aux systèmes bilinéaires par l'élaboration d'algorithmes analogues à ceux du linéaire. Un algorithme de linéarisations par morceaux permettant d'utiliser les résultats du linéaire à la commande de systèmes non-linéaires a ensuite été proposé. Le cas des systèmes non-linéaires affines en la commande a été ensuite traité. Pour cela, nous avons proposé tout d'abord un algorithme de commande faisant intervenir une linéarisation par bouclage statique sur le système à entrées continues équivalent associée à une procédure de binarisation des entrées.

Ces méthodes ont été appliquées et comparées entre elles en simulation, dans le chapitre IV, à la commande dans un premier temps, d'un moteur synchrone associé à un onduleur de tension, puis à la commande d'une machine asynchrone associée au même convertisseur de puissance. L'analyse des résultats obtenus montre que l'algorithme de linéarisations par morceaux est le seul qui permet d'avoir des informations sur les valeurs admissibles pour les consignes et qui permet une implantation immédiate de la poursuite de trajectoire. En outre, cet algorithme nécessite des périodes d'échantillonnage moins faibles que les autres algorithmes que nous avons testé. Une implantation sur site de l'algorithme de commande par linéarisations par morceaux est en cours au LEEP de l'Ecole Centrale de Lille dirigé par le Prof. Rombaut.

Nous avons traité dans ce mémoire le cas des systèmes dont les composantes de la commande sont booléennes. Le cas où ces composantes peuvent prendre deux valeurs distinctes quelconques (commandes binaires) peut être déduit aisément de cette étude par des changements de variable du même type que ceux effectués au chapitre I.

Ce travail va nous permettre de développer dans le futur des perspectives de recherche qui peuvent se décomposer en quatre axes :

- La possibilité d'utiliser les degrés de liberté dans le choix des coordonnées barycentriques pour faire de la commande optimale n'a été étudiée que partiellement au chapitre II. Il serait intéressant d'effectuer une étude approfondie pour élaborer des critères de choix permettant d'améliorer les performances des systèmes à entrées booléennes (linéaires ou non).

- Un autre axe à développer consiste en l'étude de la possibilité d'implanter des lois de commande non-périodiques, en vue, là encore d'obtenir une amélioration des performances du système (temps de réponse, robustesse, optimisation...).

- Une autre perspective est d'élargir les lois de commande que nous avons élaborées à une famille plus vaste de systèmes. D'une part, aux systèmes à entrées booléennes non-linéaires présentant des caractéristiques particulières (systèmes quadratiques, quadratiques bilinéaires, non-linéaires majorés...), et d'autre part aux systèmes hybrides en la commande, c'est à dire dont un certain nombre d'entrées sont booléennes et un certain nombre continues. Ces études peuvent être justifiées d'un point de vue applicatif par le fait qu'un certain nombre de systèmes d'électronique de puissance peuvent être modélisés par des équations de la forme de celles que nous venons de citer.

- Enfin, le fait d'avoir directement accès à l'état des commutateurs dans le modèle du système peut nous permettre d'envisager d'étudier l'effet de commutations imparfaites (pente non infinie entre les états 0 et 1).

BIBLIOGRAPHIE

BIBLIOGRAPHIE

V. Abadie et G. Dauphin-Tanguy, "Elaboration of a control law for switching nonlinear systems", *Journal of the Franklin Institute*, Vol 330, No 4, pp. 685-693, 1993(a).

V. Abadie et G. Dauphin-Tanguy, "Opened loop control of switching linear systems", *Journal of the Franklin Institute*, Vol. 330, No 5, pp. 799-813, 1993(b).

V. Abadie et G. Dauphin-Tanguy, "Control of switching continuous systems", *IEEE Systems, Man and Cybernetics Conference Proceedings*, Vol.4, pp.595-600, Le Touquet, Octobre 1993(c).

V. Abadie et G. Dauphin-Tanguy, "Commande de systèmes à commutation", communication au GR d'automatique, pôle non-linéaire, Mars 1993(d).

G.M. Asher et V. Eslamdoost, "Power electronics circuit simulation using bond-graph techniques", *EPE Congress Aachen*, pp.807-812, 1989.

G.M. Asher et V. Eslamdoost, "A novel causality changing method for the bond-graph modelling of variable topology switching circuits", *Proceedings IMACS Conference MCTS'91* pp. 371-376, Lille 1991.

A. Azmani et G. Dauphin-Tanguy, "ARCHER : a Processor for Computer Aided modelling", North Holland/Elsevier 1992.

E.A. Barbashin, "Introduction to the theory of stability", en russe : Moscow : Nauka , 1967, aussi en anglais : Wolters-Noordhoff Publishing, Groningen, Pays-bas, 1970.

C. Bergmann, "Contribution à l'étude de la commande numérique directe de moteurs synchrones à aimants"-Modélisation, réalisation, expérimentations", Doctorat, Université Paris 6, 1986.

P. Borne, G. Dauphin-Tanguy, J.P. Richard, F. Rotella, I. Zambettakis, "Modélisation et identification des processus Tome 1. Collection: méthodes et pratiques de l'ingénieur. Editions Technip, 1992.

B.K. Bose, "Sliding mode control of induction motor", Proceedings IEEE Appl. Soc. Toronto (Canada), pp.479-486, 1985.

B.K. Bose, "Power electronics and AC drives", Englewood Cliffs, NJ : Prentice-Hall, 1986.

F. Bournonville, "Découplage des systèmes non-linéaires affines", Thèse E.N.S.M., 1984.

H. Bühler, "Convertisseurs statiques", Presses polytechniques et universitaires romandes, 1991.

A. Capel, J.G. Ferrante et R. Prajoux, "Dynamic behaviour and 2 transform stability analysis of DC/DC regulators with a nonlinear PWM controlled loop", Proceedings IEEE transactions on power electronics Vol. PE-1, pp. 76-89, 1986.

A. Castelain, J.P. Ducreux, G. Dauphin-Tanguy et C. Rombaut, "Modelling and analysis of power electronics network by bond-graphs", Proceedings IMACS TC1 Symposium, Vol II pp. 405-410, 1990.

R. Chauprade, F. Milsant, "Commande électronique des moteurs à courants alternatifs", Ed. Eyrolles, 1980.

J. Clegg, "Time-dependant nonlinear compensation network", Elec. Eng., no. 10, 1956.

J.L. Cocquerelle, "Electronique de commutation", Editions Technip Paris, 1993.

G. Dauphin-Tanguy, V. Abadie, "Bond-Graphs et discontinuités", Journée AFCET/SEE, Paris, Janvier 1994.

G. Dauphin-Tanguy, C. Sueur et C. Rombaut, "Bond-graph approach of commutation phenomena", Symposium IFAC/IMACS/IFORS "AIPAC'89", Nancy, Tome 1, pp.297-301, 1989.

G. Dauphin-Tanguy et C. Rombaut, "Why a unique causality in the elementary commutation cell bond-graph model of a power electronics converter", Proceedings IEE SMC Conference, Vol.1 pp.257-263, Le Touquet 1993.

A. De Carli, "Requirements and Control Strategies in Electrical Drives", 1990.

M. Delgado et C. Brie, "DESIS, a Modelling and Simulation package based on bond-graph", 5° IFAC/IMACS Symposium on Computer Aided Design in Control Systems, Vol.1 pp.345-350, Swansea (GB), 1990.

J. Descusse, "Systèmes Non Linéaires, commande, La linéarisation entrées-sorties par difféomorphisme et bouclage", pp.45-69, ouvrage collectif réalisé dans le cadre du Groupe Non-Linéaire DRET/AFCET, coordonné par A. Fossard et D. Normand-Cyrot, Editions MASSON, 1993.

J.P. Ducreux, A. Castelain, G. Dauphin-Tanguy, C. Rombaut, "Power electronics and electrical machines modelling using bond-graphs", IMACS Transactions on "Bond-Graphs for engineers", G.Dauphin-Tanguy et P.Breedveld editors, Elsevier Sc. Publisher, 1992.

J.P. Ducreux, G. Dauphin-Tanguy, C. Rombaut, "Bond-graph modelling of commutation phenomena in power electronics circuits", Proceedings SCS Conference ICBGM'93 San Diego (USA), pp.132-136, 1993.

J.P. Ducreux et G. Dauphin-Tanguy, "Communication au GR. Automatique et électrotechnique", 1991.

H. Elmqvist, "SIMNON's users guide for MSDOS computers", Suède, 1990.

S.V. Emelyanov, "The use of nonlinear devices of the key type to improve the behavior of second order control system", Automatic Remote Control, no. 7, pp.844-859, 1959.

S.V. Emelyanov and V.A. Taran, "Use of passives switched filters in variable structure control system", Eng. Cybern, no. 5, pp.143-149, 1963(a).

S.V. Emelyanov and V.A. Taran, "On design of variable structure system for control of linear plants", Eng. Cybern, no. 2, pp.69-78, 1963(b).

S.V. Emelyanov et al, "Theory of Variable Structure System for control of linear plants", Eng. Cybern, no. 2, pp.69-78, 1963(c).

S.V. Emelyanov, "Design of variable structure control systems with discontinuous switching functions", Eng. Cybern. vol. 1 pp156-160, 1964.

S.V. Emelyanov et al, "Design principles for variable structure control systems", Proceedings 3rd IFAC Congress, vol. 1, no. 3, pp.40C.1-40C.6, 1966.

S.V. Emelyanov et V.I. Utkin, "Invariant solutions of differential equations with discontinuous coefficients. Design principles of variable structure control systems", dans *Mathematical Theory of Control*. New York : Academic press, 1967.

A. Fossard, "Stabilisation, Commande et Poursuite par régime glissant", Rapport ONERA-CERT, contrat DRET, Mars 1991.

S.J.Garrett, "Linear switching conditions for a third order positive-negative feedback control", *Appl. Ind.*, no 54, 1961.

J.J. Granda et F. Pourrahimi, "Computer graphics techniques for the generation and analysis of physical system models", proceedings of the multiconference on artificial intelligence, graphics and simulation, San Diego, 1985.

J.J. Granda, "Computer-aided modelling of multiport elements and large bond-graph models using CAMP/G, proceedings of the SCS conference ICBGM'93, San Diego, 1993.

J. Holtz et S. Stadtfeld, "A predictive controller for stator current vector of AC machines fed from a switched voltage source", Proceedings IPEC, Tokyo, pp.1665-1673, 1983.

A.Isidori, "Nonlinear control systems (2nd Edition), Springer, Berlin, 1989.

E.I. Jury, "Inners and stability of dynamic systems", Wiley, New-York, 1974.

D.C. Karnopp, R.C. Rosenberg, "Introduction to physical system dynamics", Mc Graw hill book company, 1983.

E. Kiel, W. Schumacher, et R. Gabriel, "PWM gate array for AC-drives", Proceedings EPE, pp.653-658, 1987.

P.Kokotovic, H.K. Khalil et J. O'Reilly, "Singular perturbation method in control : analysis and design", Academic press, New York, 1986.

A. Kusho et D. Galler, "Control means for minimization of losses in AC and DC motor drives", Proceedings IEEE Transactions on Industrial Applications, Vol IA-19, no.4, pp.561-570, 1983.

B. Le Pioufle, G. Georgiou et J.P. Louis, "Application des commandes non-linéaires pour la régulation en vitesse et en position de la machine synchrone Autopilotée.", Revue de Physique Appliquée, Juin 1990, pp. 517-526.

B. Le Pioufle, Thèse de doctorat de l'Université de Paris Sud, 1991.

J. Lesenne, F. Notelet, G. Segulier, "Introduction à l'électrotechnique approfondie", technique et documentation, 1982.

W. Leonhard, "Control of electrical drives", Springer, Berlin, 1985.

F. Lorenz, "Discontinuities in bond-graphs : what is required ?", Proceedings SCS Conference ICBGM'93 San Diego (USA), pp. 137-142, 1993.

Mathworks, "MATLAB : Users Reference Guide", 1984-1993.

W. McMurray, "Power Electronic circuit topology", in IEEE, Proceedings on Power Electronics, pp.428-437, 1988.

B. Michel, B. Maschke, G.Manesse, "Bond-graph enumeration of the configurations of power static converters", Proceedings IEEE SMC Conference Vol.1 pp. 251-256, Le Touquet, 1993.

R.D. Middlebrook et S. Cuk, "A general unified approach to modelling switching DC-DC converters in discontinuous conduction modes", Proceedings IEEE PESC, pp.36-57, 1977.

R. Mitchell et P. Gauthier, "Users guide to the Advanced Continuous Simulation Language (ACSL) Concord, MA, USA, 1986, 1989, 1993.

N. Mohan, T.M. Undeland et W.P. Robbins, "Power Electronic Converters, Applications and Design", Wiley, New York, 1989.

R. Mohler, "Nonlinear systems Vol. II, Application to bilinear control", Prentice Hall, Englewood Cliffs, New Jersey, 1991.

M. Nahklé, "NEPTUNIX, an efficient tool for large size system simulations", Proceedings 2^o International Conference on System Simulations in building, Lièges (Belgique), 1986.

I. Neimark, "On sliding modes in relay control systems", Automatic Remote Control, no.1 pp.29-35, 1957.

H. Nijmeijer, A. Van der Schaft, "Nonlinear dynamical control system dynamics", Springer Verlag, New York Inc.

S. Nonaka et Y. Neba, "Analysis of a PWM GTO current source inverter fed induction motor drive system", IEEE Transactions on Industry Applications, Vol.23 No.2, 1987.

H.M. Paynter, "Analysis and design of engineering systems", MIT Press, Cambridge, Mass, 1961.

A.B. Plunkett, "Direct flux and torque regulation in a PWM induction motor drive", IEEE Transactions on Industry Applications, pp.139-147, 1977.

G.M. Ostrovsky, "Application of nonlinear correction to control systems of the second order", Automatic Remote Control, no.11, 1956.

M.H. Rashid, "Power Electronics, Circuits, Devices and Applications", Prentice-Hall, Englewood Cliffs, 1988.

S. Rimaux, V. Abadie and G. Dauphin-Tanguy, "Control law for switching bilinear systems", soumis pour publication dans IEEE Circuits and Systems.

S. Rimaux, V. Abadie and G. Dauphin-Tanguy, "Structural control analysis of switching bilinear systems modelled by Bond-Graph", actuellement en préparation.

R.C. Rosenberg, "The ENPORT Reference Manual ", Lausing-Michigan, USA, Rosencode, 1990.

A. Sabanovic et D.B. Izosimov, "Application of sliding modes to induction motor control", IEEE Transactions on Industry Applications, Vol 17, no.1 pp. 41-49, 1981.

A. Sabanovic et M. Vujovic, "Sliding mode control of DC drives", Proceedings IEEE Conference on Power Electronics and Variable Speed Drives, Londres (GB), 1984.

A. Sabanovic, R. Benitez, H. Hashimoto et F. Harashima, "VSS approach to DC drives control", IEEE PESC, pp.235-242, Kyoto (Japan), 1988.

A. Sabanovic, N. Sabanovic et K. Ohnishi, "Sliding modes in power converters and motion control systems", International Journal of Control, Vol. 57, No5, pp.1237-1259, 1993.

M.G. Say, "Alternating current machine", Edit. Pitman, 1986.

G. Seguier et F. Labrique, "Les convertisseurs de l'électronique de puissance", Editions techniques et documentation Lavoisier, 1989.

B. Semail, E. Mendes, F. Bouillault, A. Razek, "Vector controlled induction machine simulation- parameter sensitivity analysis", IMACS TC1, pp.291-296, Nancy, 1990

B. Semail, Thèse de doctorat de l'Université de Paris Sud, 1991.

B. Semail, C. Rombaut, G. Dauphin-Tanguy, V. Abadie, "Réunions LEEP-LAIL", Ecole Centrale de Lille, 1991-1994.

J. Shörner, "Speed controlled industrial drives", in Proceedings ICEM'86, pp.40-45, Munich, 1986.

J.F. Silva, "Sliding mode voltage control in current mode PWM inverters", PESC'92 Proceedings, Vol. II pp.762-769, Toledo (Espagne), 1992.

H. Sira-Ramirez, "A geometric approach to pulse-width-modulated control design", Proceedings IEEE Conference Decision and Control, Los Angeles, Vol. 3, pp. 1771-1776, 1987 (a).

H. Sira-Ramirez, "Sliding motions in bilinear switched networks", IEEE Transaction on Circuits and Systems, Vol CAS-34, pp. 919-933, 1987(b).

H. Sira-Ramirez, "Sliding-mode control of slow manifolds of DC-to-DC power converters", International Journal of Control, Vol. 47, pp. 1323-1340, 1988.

H. Sira-Ramirez, "Variable Structure Control in nonlinear Dynamical Systems", IEEE Transactions on Automatic Control, Vol.34, no.2, pp1184-187, 1989.

R.A. Skoog et G.L. Blankenship, "Generalized pulse-modulated feedback systems : norms, gains, Lipschitz constants and stability", IEEE Trans. Automat. Contr., vol. AC-15, pp. 300-315, 1970.

J.E. Strömberg, J. Top et U. Söderman, "Variable causality in bond-graph caused by discrete effects", Proceedings SCS Conference ICBGM'93, San Diego (USA), pp.115-119, 1993.

V.A. Taran, "Design of control systems applying switched filters", Engineering and Cybernetics, no.4, pp.172-180, 1965.

University of Twente, "CAMAS : Computer Aided Modelling, Analysis and Simulation Reference Manual ", 1991.

University of Twente, "TUTSIM : Users Reference Manual ", 1985-1992.

V.I. Utkin, "Equations of sliding mode in discontinuous systems", Automatic Remote Control no .21 pp1897-1907, 1971.

V.I. Utkin, "Sliding modes and their Application in Variable Structure Systems, MIR Moscow, 1974.

V.I. Utkin, "Variable structure systems with sliding modes", IEE Transaction on Automatic Control, pp.212-222, 1977.

V. Vorpérian, "Simplified analysis of PWM converters using models of PWM switches", Part I et II, IEEE Trans. on Aero. and Elec. Syst. Vol.26, no.3, pp.490-505, 1990.

J.H. Wilkinson, "The algebraic eigenvaleurs problem", Oxford University Press, London, 1965.

R.C. Wrong, "Accelerated convergence to the steady state solution of closed loop regulated switching mode systems as obtained through simulations", Proceedings IEEE PESC, no.87 pp.682-692, 1987.

ANNEXES

ANNEXES

ANNEXE 1 :

On choisit :
$$z_n = x_n - [\mathbf{I} - e^{A T_{mod}}]^{-1} K,$$

Il faut justifier l'inversibilité de $\mathbf{I} - e^{A T_{mod}}$: Considérons $\{\lambda_1, \dots, \lambda_q\}$ l'ensemble des valeurs propres de \mathbf{A} . On a $\mathbf{A} = \mathbf{P}^{-1} \mathbf{A}' \mathbf{P}$ où \mathbf{P} est une matrice inversible de $\mathbf{C}^{q \times q}$ et \mathbf{A}' une matrice triangulaire telle que :

$$\mathbf{A}' = \begin{pmatrix} \lambda_1 & 0 & \dots & 0 \\ a'_{2,1}(1) & \lambda_2 & \ddots & \vdots \\ \vdots & \ddots & \ddots & 0 \\ a'_{q,1}(1) & \dots & a'_{q,q-1}(1) & \lambda_q \end{pmatrix}$$

où $a'_{i,j}$, pour $i \in \{2, \dots, q\}$ et $j \in \{1, \dots, q-1\}$ sont les coefficients a priori complexes de \mathbf{A}' .

Alors :

$$e^{A T_{mod}} = \sum_{k=0}^{\infty} \frac{(\mathbf{A} T_{mod})^k}{k!} = \sum_{k=0}^{\infty} \frac{\mathbf{P}^{-1} (\mathbf{A}' T_{mod})^k \mathbf{P}}{k!} = \mathbf{P}^{-1} \left(\sum_{k=0}^{\infty} \frac{(\mathbf{A}' T_{mod})^k}{k!} \right) \mathbf{P}.$$

Or :

$$(\mathbf{A}')^k = \begin{pmatrix} (\lambda_1)^k & 0 & \dots & 0 \\ a'_{2,1}(k) & (\lambda_2)^k & \ddots & \vdots \\ \vdots & \ddots & \ddots & 0 \\ a'_{q,1}(k) & \dots & a'_{q,q-1}(k) & (\lambda_q)^k \end{pmatrix}$$

Alors :

$$e^{A T_{mod}} = \mathbf{P}^{-1} \left(\sum_{k=0}^{\infty} \frac{(\mathbf{A}' T_{mod})^k}{k!} \right) \mathbf{P} = \mathbf{P}^{-1} \begin{pmatrix} e^{\lambda_1 T_{mod}} & 0 & \dots & 0 \\ s_{2,1} & e^{\lambda_2 T_{mod}} & \ddots & \vdots \\ \vdots & \ddots & \ddots & 0 \\ s_{q,1} & \dots & s_{q,q-1} & e^{\lambda_q T_{mod}} \end{pmatrix} \mathbf{P}$$

avec $s_{i,j}$, $i \in \{2, \dots, q\}$, $j \in \{1, \dots, q-1\}$ coefficients a priori complexes.

Alors :

$$\det(\mathbf{I} - e^{A_{T_{mod}}}) = \det \mathbf{P} \cdot \det \left(\mathbf{I} - \begin{pmatrix} e^{\lambda_1 T_{mod}} & 0 & \dots & 0 \\ s_{2,1} & e^{\lambda_2 T_{mod}} & \ddots & \vdots \\ \vdots & \ddots & \ddots & 0 \\ s_{q,1} & \dots & s_{q,q-1} & e^{\lambda_q T_{mod}} \end{pmatrix} \right) \det \mathbf{P}^{-1}$$

Alors :

$$\det(\mathbf{I} - e^{A_{T_{mod}}}) = \det \begin{pmatrix} 1 - e^{\lambda_1 T_{mod}} & 0 & \dots & 0 \\ -s_{2,1} & 1 - e^{\lambda_2 T_{mod}} & \ddots & \vdots \\ \vdots & \ddots & \ddots & 0 \\ -s_{q,1} & \dots & -s_{q,q-1} & 1 - e^{\lambda_q T_{mod}} \end{pmatrix}$$

ce qui donne,

$$\det(\mathbf{I} - e^{A_{T_{mod}}}) = \prod_{k=1}^q (1 - e^{\lambda_k T_{mod}})$$

Ce déterminant est égal à zéro si et seulement si il existe λ_k tel que $\lambda_k = 0$ ce qui est impossible d'après l'hypothèse de stabilité asymptotique de \mathbf{A} . Alors $\mathbf{I} - e^{A_{T_{mod}}}$ est une matrice inversible.

Nous pouvons donc définir z_n : $z_n = x_n - (\mathbf{I} - e^{A_{T_{mod}}})K$

où x_n est défini par : $x_{n+1} = e^{A_{T_{mod}}}x_n + K$.

Alors, il vient : $z_{n+1} + (\mathbf{I} - e^{A_{T_{mod}}})^{-1}K = e^{A_{T_{mod}}}(z_n + (\mathbf{I} - e^{A_{T_{mod}}})^{-1}K) + K$

ce qui donne : $z_{n+1} = e^{A_{T_{mod}}}z_n$

soit :

$$z_n = (e^{A_{T_{mod}}})^n z_0 = (e^{A_{T_{mod}}})^n \left(-[\mathbf{I} - e^{A_{T_{mod}}}]^{-1} K \right),$$

et en remplaçant z_n par sa valeur :

$$x_n = \left(\mathbf{I} - (e^{A_{T_{mod}}})^n \right) \left([\mathbf{I} - e^{A_{T_{mod}}}]^{-1} K \right) + (e^{A_{T_{mod}}})^n x(t_0)$$

ANNEXE 2 :

L'égalité $x_\infty = x_d$ entraîne :

$$\left(\mathbf{I} - e^{A T_{mod}} \right)^{-1} \left(\sum_{i=1}^{2^m} e^{\alpha_{2^m} A T_{mod}} \dots e^{\alpha_{i+1} A T_{mod}} \mathbf{A}^{-1} \left(\mathbf{I} - e^{\alpha_i A T_{mod}} \right) \mathbf{B}_i \right) = \sum_{i=1}^{2^m} a_i \mathbf{A}^{-1} \mathbf{B}_i$$

Alors, il vient :

$$\left(\mathbf{I} - e^{A T_{mod}} \right)^{-1} \left(\sum_{i=1}^{2^m} \mathbf{A}^{-1} e^{\alpha_{2^m} A T_{mod}} \dots e^{\alpha_{i+1} A T_{mod}} \left(\mathbf{I} - e^{\alpha_i A T_{mod}} \right) \mathbf{B}_i \right) = \sum_{i=1}^{2^m} a_i \mathbf{A}^{-1} \mathbf{B}_i,$$

parce que :

$$\mathbf{A} e^{\alpha_i A T_{mod}} = \mathbf{A} \left(\sum_{k=0}^{\infty} \frac{\mathbf{A}^k (\alpha_i T_{mod})^k}{k!} \right) = \sum_{k=0}^{\infty} \frac{\mathbf{A}^{k+1} (\alpha_i T_{mod})^k}{k!} = e^{\alpha_i A T_{mod}} \mathbf{A}$$

alors :

$$\left(\mathbf{I} - e^{A T_{mod}} \right)^{-1} \mathbf{A}^{-1} \left(\sum_{i=1}^{2^m} e^{\alpha_{2^m} A T_{mod}} \dots e^{\alpha_{i+1} A T_{mod}} \left(\mathbf{I} - e^{\alpha_i A T_{mod}} \right) \mathbf{B}_i \right) = \mathbf{A}^{-1} \sum_{i=1}^{2^m} a_i \mathbf{B}_i$$

Nous avons :

$$\left(\mathbf{I} - e^{A T_{mod}} \right)^{-1} \mathbf{A}^{-1} = \left(\mathbf{A} \left(\mathbf{I} - e^{A T_{mod}} \right) \right)^{-1} = \left(\mathbf{A} - \mathbf{A} e^{A T_{mod}} \right)^{-1} = \left(\mathbf{A} - e^{A T_{mod}} \mathbf{A} \right)^{-1} = \mathbf{A}^{-1} \left(\mathbf{I} - e^{A T_{mod}} \right)^{-1}$$

Alors en multipliant chaque terme par \mathbf{A} , nous avons :

$$\sum_{i=1}^{2^m} e^{\alpha_{2^m} A T_{mod}} \dots e^{\alpha_{i+1} A T_{mod}} \left(\mathbf{I} - e^{\alpha_i A T_{mod}} \right) \mathbf{B}_i = \sum_{i=1}^{2^m} \left(\mathbf{I} - e^{A T_{mod}} \right) a_i \mathbf{B}_i$$

En identifiant terme à terme, on obtient le résultat énoncé.

ANNEXE 3 :

On a, en effectuant un développement limité :

$$\phi(A T_{mod}) \cong \frac{(a_i - \alpha_i^2)}{2} I - a_i (\alpha_{i+1} + \dots + \alpha_{2^m}) I.$$

Nous avons :

$$\frac{a_i - \alpha_i^2}{2} - (\alpha_{i+1} + \dots + \alpha_{2^m})a_i < \frac{a_i - \alpha_i^2}{2}$$

car
$$(\alpha_{i+1} + \dots + \alpha_{2^m})a_i > 0$$

et :
$$\frac{a_i - \alpha_i^2}{2} < \frac{a_i}{2} < a_i$$

ce qui entraîne :
$$\frac{a_i - \alpha_i^2}{2} - (\alpha_{i+1} + \dots + \alpha_{2^m})a_i < a_i.$$

ANNEXE 4 :

On a :

$$y_d = \sum_{i=1}^{2^m} a_i y_{i\infty}$$

or $y_{i\infty} = \mathbf{C}x_{i\infty}$, donc :

$$y_d = \sum_{i=1}^{2^m} a_i \mathbf{C}x_{i\infty} = \mathbf{C} \sum_{i=1}^{2^m} a_i x_{i\infty}$$

en posant :

$$x_d = \sum_{i=1}^{2^m} a_i x_{i\infty}$$

on obtient :

$$y_d = \mathbf{C}x_d.$$

ANNEXE 5 :

S'il existe i tel que $\mathbf{A} + \eta_i$ est non inversible, on construit \mathbf{A}^* en ajoutant ε_i un réel positif à un terme de la matrice \mathbf{A} . Du fait de la continuité du déterminant, il existe un réel r tel que

\mathbf{A}^* soit inversible pour toute valeur de ε_i strictement comprise entre 0 et r . On considère alors l'équation :

$$\frac{dx}{dt} = \mathbf{A}^* x + \sum_{i=1}^m \mathbf{N}_i x u_i + \sum_{i=1}^m \mathbf{B}_i u_i$$

En effectuant les mêmes calculs qu'au §III.1.2.2 le système converge vers :

$$x_{\infty}^* = \left[\mathbf{I} - e^{\sum_{k=1}^{2^m} (\mathbf{A}^* + \eta_k) \alpha_k T_{mod}} \right]^{-1} \sum_{k=1}^{2^m} e^{\sum_{j=k+1}^{2^m} (\mathbf{A}^* + \eta_j) \alpha_j T_{mod}} (\mathbf{A}^* + \eta_k)^{-1} (\mathbf{I} - e^{(\mathbf{A}^* + \eta_k) \alpha_k T_{mod}}) \beta_k$$

Si T_{mod} vérifie $\sum_{k=1}^{2^m} (\mathbf{A}^* + \eta_k) \alpha_k T_{mod} \ll 1$, les α_i doivent être choisis tels que :

$$\begin{cases} -\mathbf{A}^* x_d = \sum_{i=1}^{2^m} \alpha_i (\eta_i x_d + \beta_i) \\ \sum_{i=1}^{2^m} \alpha_i = 1 \\ \alpha_i \in [0,1], i = 1, \dots, 2^m \\ \sum_{k=1}^{2^m} (\mathbf{A}^* + \eta_k) \alpha_k \text{ stable.} \end{cases}$$

En faisant tendre ε_i vers 0 dans ces équations, on retrouve le système (3.10).

ANNEXE 6 :

On pose :

$$I_N = \int_0^{(N+1)T_k} e^{-((N+1)T_k - \tau)} g(x)(u - u_c) d\tau$$

On a :

$$I_N = \sum_0^N \int_{nT_k}^{(n+1)T_k} e^{-((N+1)T_k - \tau)} g(x)(u - u_c) d\tau$$

Si $|g(x)| < A$, alors :

$$|I_N| \leq \sum_0^N A \int_{nT_k}^{(n+1)T_k} e^{-((N+1)T_k - \tau)} |u - u_c| d\tau$$

On a sur $[nT_k, (n+1)T_k]$,

$$\int_{nT_k}^{(n+1)T_k} |u - u_c| d\tau = \int_{nT_k}^{(n+\hat{u}_c(nT_k))T_k} (1 - u_c) d\tau + \int_{(n+\hat{u}_c(nT_k))T_k}^{(n+1)T_k} u_c d\tau$$

donc en majorant $1 - u_c$ par $1 + u_c$:

$$\int_{nT_k}^{(n+1)T_k} |u - u_c| d\tau \leq 2T_k \hat{u}_c(nT_k) \leq 2T_k$$

Il en résulte que :

$$|I_N| \leq \sum_0^N 2AT_k \int_{nT_k}^{(n+1)T_k} e^{-((N+1)T_k - \tau)} d\tau$$

soit :

$$|I_N| \leq 2AT_k (1 - e^{-(N+1)T_k})$$

Par conséquent :

$$\lim_{t \rightarrow \infty} \left| \int_0^t e^{-(t-\tau)} g(x)(u - u_c) d\tau \right| \leq 2AT_k$$

On peut donc choisir T_k suffisamment petit pour que cette limite soit considérée comme nulle, ce qui démontre l'affirmation p.87. L'étude peut être menée de la même manière dans le cas multientrées (projection des majorants sur chaque axe).

ANNEXE 7 :

Les flux et les courants au stator sont liés par la relation :

$$\phi = Li$$

avec $\phi = (\phi_1, \phi_2, \phi_3)^T$, $i = (i_1, i_2, i_3)^T$ et la matrice inductance L définie par :

$$L = \begin{pmatrix} L & M & 0 \\ M & L & 0 \\ 0 & 0 & L_Q \end{pmatrix}$$

Les variables flux (courants) au sens du bond-graph seront notées f_i et les variables effort (tensions) e_i .

D'autre part les courants et le couple τ sont liés par :

$$\tau = \frac{1}{2} i' \frac{dL}{dt} i = (f_2 f_4 - f_1 f_3)$$

Les équations du bond-graph donnent pour les condensateurs :

$$\begin{cases} \frac{dq_1}{dt} = \left(-\frac{1}{R_a C_a} - \frac{m_1^2}{R_{t1} C_a} - \frac{m_4^2}{R_{t4} C_a} \right) q_1 - f_1 + \frac{m_1^2}{R_{t1}} E \\ \frac{dq_2}{dt} = \left(-\frac{1}{R_a C_a} - \frac{m_2^2}{R_{t2} C_a} - \frac{m_5^2}{R_{t5} C_a} \right) q_2 - f_2 + \frac{m_2^2}{R_{t2}} E \\ \frac{dq_3}{dt} = \left(-\frac{1}{R_a C_a} - \frac{m_3^2}{R_{t3} C_a} - \frac{m_6^2}{R_{t6} C_a} \right) q_3 - f_3 + \frac{m_3^2}{R_{t3}} E \end{cases}$$

et pour l'inertie du rotor :

$$\begin{cases} \frac{d\theta}{dt} = \omega \\ \frac{d\omega}{dt} = \frac{-f}{J} \omega + \frac{\tau}{J} \end{cases}$$

Les flux relatifs au stator sont décrits par :

$$\begin{cases} \frac{d\phi_1}{dt} = \frac{-2}{3} R_{s1} f_1 + \frac{1}{3} R_{s2} f_2 + \frac{1}{3} R_{s3} f_3 + \frac{2}{3} \frac{q_1}{C_a} - \frac{1}{3} \frac{q_2}{C_a} - \frac{1}{3} \frac{q_3}{C_a} \\ \frac{d\phi_2}{dt} = \frac{-2}{3} R_{s2} f_2 + \frac{1}{3} R_{s1} f_1 + \frac{1}{3} R_{s3} f_3 + \frac{2}{3} \frac{q_2}{C_a} - \frac{1}{3} \frac{q_1}{C_a} - \frac{1}{3} \frac{q_3}{C_a} \\ \frac{d\phi_3}{dt} = \frac{-2}{3} R_{s3} f_3 + \frac{1}{3} R_{s2} f_2 + \frac{1}{3} R_{s1} f_1 + \frac{2}{3} \frac{q_3}{C_a} - \frac{1}{3} \frac{q_2}{C_a} - \frac{1}{3} \frac{q_1}{C_a} \end{cases}$$

et, au rotor :

$$\begin{cases} \frac{d\phi_4}{dt} = -R_r f_4 \\ \frac{d\phi_5}{dt} = -R_r f_5 \\ \frac{d\phi_6}{dt} = -R_r f_6 \end{cases}$$

ANNEXE 8

En plus de ces relations, le bloc MTF traduit la transformation de Park (au stator) et Concordia . La matrice de Park permettant d'effectuer ce changement de base pour les flux est définie par :

$$T(\theta) = \sqrt{\frac{2}{3}} \begin{pmatrix} \frac{1}{\sqrt{2}} & \frac{1}{\sqrt{2}} & \frac{1}{\sqrt{2}} \\ \cos(\theta) & \cos(\theta - \frac{2\pi}{3}) & \cos(\theta - \frac{4\pi}{3}) \\ -\sin(\theta) & -\sin(\theta - \frac{2\pi}{3}) & -\sin(\theta - \frac{4\pi}{3}) \end{pmatrix}$$

On effectue la même rotation pour les courants. On a alors une nouvelle matrice d'inductance après transformation de Park (machine équilibrée):

$$L_0 = \begin{pmatrix} L & 0 & 0 & M & 0 & 0 \\ 0 & L & 0 & 0 & M & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & -M\omega & 0 \\ 0 & 0 & 0 & M\omega & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 \end{pmatrix}$$

Si on suppose que $\frac{dq_i}{dt} \cong 0$, il vient :

$$\left\{ \begin{array}{l} \frac{d\phi_{s\alpha}}{dt} = -R_s j_{s\alpha} + \frac{E}{\sqrt{6}} (2m_1^2 - m_2^2 - m_3^2) \\ \frac{d\phi_{s\beta}}{dt} = -R_s j_{s\beta} + \frac{E}{\sqrt{2}} (2m_1^2 - m_2^2 - m_3^2) \\ \frac{d\phi_{r\alpha}}{dt} = -\omega \phi_{r\beta} \\ \frac{d\phi_{r\beta}}{dt} = \omega \phi_{r\alpha} \\ \frac{d\omega}{dt} = \frac{-f}{J} \omega + \frac{1}{J} (i_{s\beta} \phi_{r\alpha} - i_{s\alpha} \phi_{r\beta}) \end{array} \right.$$

avec $R_{t1}=\dots=R_{t6}=R_t$ et $R_t \ll R_a$, $R_t \ll R_s$. Ces hypothèses sont justifiées par le fait que les éléments R_a et C_a sont utilisés pour mesurer les courants et n'interviennent pas sur le comportement du système.

avec en plus :

$$\left\{ \begin{array}{l} \phi_{sd} = \phi_{rd} + L i_{sd} \\ \phi_{sq} = \phi_{rq} + L i_{sq} \end{array} \right.$$

et $L=L_s-M_s$.

Machine asynchrone

équations du bond-graph sont les mêmes, seuls la matrice inductance change.

$\phi = (\phi_1, \phi_2, \phi_3, \phi_4, \phi_5, \phi_6)^T$, $i = (i_1, i_2, i_3, i_4, i_5, i_6)^T$ et la matrice inductance L est définie par :

$$L = \begin{pmatrix} l_s & m_s & m_s & m_1 & m_3 & m_2 \\ m_s & l_s & m_s & m_2 & m_1 & m_3 \\ m_s & m_s & l_s & m_3 & m_2 & m_1 \\ m_1 & m_2 & m_3 & l_r & m_r & m_r \\ m_3 & m_1 & m_2 & m_r & l_r & m_r \\ m_2 & m_3 & m_1 & m_r & m_r & l_r \end{pmatrix}$$

avec l_s, m_s, l_r, m_r constants et $m_1 = m \cos(\theta)$, $m_2 = m \cos(\theta + 2\pi/3)$, $m_3 = m \cos(\theta + 4\pi/3)$.

Une transformation de Park au stator permet d'aboutir au résultat énoncé dans la partie II du chapitre IV (voir Ducreux et al, 1991, 1992).

ANNEXE 9 :

Supposons que :

$$\begin{aligned}i_{\alpha} &= -i_0 \sin(\omega_d t) \\i_{\beta} &= i_0 \cos(\omega_d t)\end{aligned}$$

on a alors, en régime permanent, la vitesse qui est régie par :

$$\frac{d\omega}{dt} = \frac{-f}{J} \omega + \frac{\phi_f}{J} (i_0) - \gamma_r$$

ce qui donne en reprenant les notations du IV.1.2 :

$$= \omega_d = b \tau_m i_0 - \gamma_r \tau_m$$

La même démonstration peut être appliquée pour la machine asynchrone.

ANNEXE 10 :

Nous avons sur l'intersection des surfaces de glissement :

$$\begin{pmatrix} \dot{s}_1(x) \\ \dot{s}_2(x) \end{pmatrix} = \begin{pmatrix} \frac{1}{\tau} i_0 \sin(\omega_d t) + a \omega_d \sin(\omega_d t) + i_0 \omega_d \cos(\omega_d t) \\ -\frac{1}{\tau} i_0 \cos(\omega_d t) - a \omega_d \cos(\omega_d t) + i_0 \omega_d \sin(\omega_d t) \end{pmatrix} + \frac{1}{L} \begin{pmatrix} u_{\alpha} \\ u_{\beta} \end{pmatrix}$$

Nous avons :

$$-\frac{1}{\tau} i_0 - a |\omega_d| - i_0 |\omega_d| \leq \frac{1}{\tau} i_0 \sin(\omega_d t) + a \omega_d \sin(\omega_d t) + i_0 \omega_d \cos(\omega_d t) \leq \frac{1}{\tau} i_0 + a |\omega_d| + i_0 |\omega_d|$$

et :

$$-\frac{1}{\tau} i_0 - a |\omega_d| - i_0 |\omega_d| \leq -\frac{1}{\tau} i_0 \cos(\omega_d t) - a \omega_d \cos(\omega_d t) + i_0 \omega_d \sin(\omega_d t) \leq \frac{1}{\tau} i_0 + a |\omega_d| + i_0 |\omega_d|$$

Si $\frac{E_0}{L\sqrt{6}} \geq \frac{1}{\tau} i_0 + a |\omega_d| + i_0 |\omega_d|$ cela entraîne que $\frac{E_0}{L\sqrt{2}} \geq \frac{1}{\tau} i_0 + a |\omega_d| + i_0 |\omega_d|$ qui entraîne

que les conditions de glissement sur l'intersection des deux surfaces sont vérifiées. Cette conditions est vérifiée pour les valeurs numériques données en IV.1.3.3.

