

Année : 1999

|2|5|4|0|

**THESE**

pour obtenir le grade de

**DOCTEUR DE L'UNIVERSITE**

*Discipline : PRODUCTIQUE,  
AUTOMATIQUE et INFORMATIQUE INDUSTRIELLE*

présentée et soutenue publiquement

le 28 Juin 1999

par

**William HOLDERBAUM**



**COMMANDE DES SYSTEMES A ENTREES  
BOOLEENNES**

*directeurs de thèse :*

P. BORNE	Professeur à l'Ecole Centrale de Lille
G. DAUPHIN-TANGUY	Professeur à l'Ecole Centrale de Lille

**JURY**

M.	J. P. RICHARD	Président	Professeur à l'Ecole Centrale de Lille (LAIL)
M.	J. FAUCHER	Rapporteur	Professeur à l'ENSEEIH (LEEI) de Toulouse
M.	D. MEIZEL	Rapporteur	Professeur à l'UTC (HEUDIASYC)
N.	J. BUISSON	Examineur	Professeur à Supélec (ASCH) de Rennes
Mme.	B. SEMAIL	Examinatrice	Professeur à l'Université de Lille I (L2EP)

Thèse préparée au Laboratoire d'Automatique et Informatique Industrielle de Lille  
L.A.I.L. UPRESA CNRS 8021 - Ecole Centrale de Lille.

# Sommaire

<i>Introduction Générale</i> .....	1
<b>I. Les Systèmes à Entrées Binaires</b> .....	5
<b>I.1 Introduction</b> .....	5
<b>I.2 Les systèmes à commandes booléennes</b> .....	6
<i>I.2.1 Systèmes Hybrides</i> .....	6
<i>I.2.2 Modélisation des systèmes à commutation</i> .....	7
<i>I.2.3 Exemple de modélisation par bond-graph</i> .....	7
<i>I.2.4 Conclusion</i> .....	10
<b>I.3 Les différentes commandes booléennes</b> .....	11
<i>I.3.1 Détermination directe de la commande booléenne</i> .....	11
<b>I.3.1.1 Présentation</b> .....	12
<b>I.3.1.2 Domaine convergence</b> .....	12
<b>I.3.1.3 Commande en boucle ouverte</b> .....	13
<b>I.3.1.4 Commande en boucle fermée</b> .....	18
<i>I.3.2 commande binaire à ramener dans <math>\{0,1\}</math></i> .....	20
<b>I.3.2.1 Système à structure variable</b> .....	20
<b>I.3.2.2 Cas général du mode glissant</b> .....	21
I.3.2.2.1 Systèmes Multivariables.....	21
I.3.2.2.2 Système mono-entrée .....	22
<b>I.3.2.3 Application au cas particulier des systèmes à entrées booléennes</b> .....	24
<b>I.3.2.4 Exemple</b> .....	25
<i>I.3.3 Commande par valeur moyenne</i> .....	26
<b>I.3.3.1 Commande en électronique de puissance</b> .....	27
<b>I.3.3.2 Commande par M.L.I</b> .....	28
<b>I.3.3.3 Commande par hystérésis</b> .....	30
<i>I.3.4 Binarisation d'une commande continue</i> .....	31
<b>I.3.4.1 Valeur moyenne</b> .....	32
<b>I.3.4.2 Commande Booléenne</b> .....	33
<b>I.4 Conclusion</b> .....	34

---

<b>II. Nouvelle méthode de calcul d'une commande booléenne</b> .....	<b>39</b>
<b>II.1 Introduction</b> .....	<b>39</b>
<b>II.2 Présentation des systèmes utilisés</b> .....	<b>39</b>
<b>II.3 Régulation autour d'une consigne constante</b> .....	<b>40</b>
<i>II.3.1 Principe</i> .....	40
<i>II.3.2 Etude de la stabilité</i> .....	42
<i>II.3.3 Cas particulier</i> .....	49
<i>II.3.4 Exemple</i> .....	50
<b>II.4 Poursuite de trajectoire</b> .....	<b>56</b>
<i>II.4.1 Algorithme</i> .....	56
<i>II.4.2 Exemple</i> .....	57
<i>II.4.3 Résultat</i> .....	61
<b>II.5 Minimisation de l'énergie dissipée dans les commutations</b> .....	<b>63</b>
<i>II.5.1 Commande booléenne et réduction des commutations</i> .....	64
<b>II.5.1.1 Méthode de la distance de Hamming</b> .....	<b>64</b>
<b>II.5.1.2 Méthode de l'espace de dimension <math>n</math></b> .....	<b>66</b>
<i>II.5.2 Application</i> .....	66
<b>II.5.2.1 Exemple 1</b> .....	<b>67</b>
<b>II.5.2.2 Exemple 2</b> .....	<b>70</b>
<b>II.5.2.3 Conclusion</b> .....	<b>72</b>
<b>II.6 Conclusion</b> .....	<b>73</b>
<b>III. Approche Neuromimétique de la Commande Booléenne</b> .....	<b>77</b>
<b>III.1 Introduction</b> .....	<b>77</b>
<b>III.2 Présentation</b> .....	<b>78</b>
<b>III.3 Les Réseaux de Neurones</b> .....	<b>80</b>
<i>III.3.1 Structure d'un réseau neuromimétique</i> .....	80
<b>III.4 Utilisation en Commande Booléenne</b> .....	<b>83</b>
<i>III.4.1 Initialisation</i> .....	84
<i>III.4.2 Traitement des données</i> .....	84
<i>III.4.3 Algorithme d'apprentissage</i> .....	86

---

III.4.4 <i>Algorithme de commande</i> .....	89
<b>III.5 Application au moteur synchrone</b> .....	<b>91</b>
III.5.1 <i>Présentation de la machine synchrone</i> .....	91
III.5.2 <i>Simulation</i> .....	95
<b>III.6 Conclusion</b> .....	<b>102</b>
<b>IV. Mise en oeuvre et comparaison des diverses commandes présentées</b>	<b>105</b>
<b>IV.1 Introduction</b> .....	<b>105</b>
<b>IV.2 Structure du système non linéaire</b> .....	<b>105</b>
IV.2.1 <i>Modélisation par Bond-Graph</i> .....	106
IV.2.2 <i>Principe de commande</i> .....	111
<b>IV.2.2.1 Contrôleur de la vitesse du moteur</b> .....	<b>111</b>
<b>IV.2.2.2 schéma de commande</b> .....	<b>114</b>
<b>IV.3 Mise en oeuvre des différentes commandes</b> .....	<b>115</b>
IV.3.1 <i>Critère de comparaison</i> .....	115
IV.3.2 <i>Commande M.L.I</i> .....	116
<b>IV.3.2.1 Elaboration de l'algorithme</b> .....	<b>116</b>
<b>IV.3.2.2 Simulation</b> .....	<b>118</b>
IV.3.3 <i>Commande par Modes Glissants</i> .....	119
<b>IV.3.3.1 Elaboration de l'algorithme</b> .....	<b>119</b>
<b>IV.3.3.2 Simulation</b> .....	<b>121</b>
IV.3.4 <i>Commande Booléenne</i> .....	123
<b>IV.3.4.1 Elaboration de l'algorithme</b> .....	<b>123</b>
<b>IV.3.4.2 Simulation</b> .....	<b>125</b>
IV.3.5 <i>Commande neuronale</i> .....	126
<b>IV.3.5.1 Elaboration de l'algorithme</b> .....	<b>126</b>
<b>IV.3.5.2 Simulation</b> .....	<b>128</b>
<b>IV.4 Comparaisons</b> .....	<b>131</b>
IV.4.1 <i>comparaison des structures de commande</i> .....	131
IV.4.2 <i>Comparaisons des indicateurs</i> .....	132
IV.4.3 <i>Comparaison des stratégies de mise en oeuvre</i> .....	134
<b>IV.5 Conclusion</b> .....	<b>134</b>

<i>Conclusion Générale.....</i>	<i>139</i>
<i>Perspectives.....</i>	<i>140</i>
<i>Bibliographie.....</i>	<i>145</i>

*Annexe*

<i>Annexe A.....</i>	<i>155</i>
<i>Annexe B.....</i>	<i>161</i>
<i>Annexe C.....</i>	<i>169</i>
<i>Annexe D.....</i>	<i>177</i>
<i>Annexe E.....</i>	<i>187</i>

# Sommaire

*A mes parents*

# Avant Propos

Le travail que nous présentons dans ce mémoire a été effectué au laboratoire d'automatique et d'informatique industrielle de Lille (LAIL) à l'Ecole Centrale de Lille sous la direction de Madame le Professeur **G. DAUPHIN-TANGUY** et de Monsieur le Professeur **P. BORNE**.

Je tiens à exprimer toute ma gratitude à Madame **G. DAUPHIN-TANGUY**, Professeur et directeur adjoint du LAIL, pour son aide précieuse, sa disponibilité, et la qualité de ses rapports humains qui m'ont permis d'achever dans les meilleures conditions les travaux entrepris

Je tiens à remercier très vivement Monsieur le Professeur **P. BORNE**, Directeur scientifique de l'Ecole Centrale de Lille, pour ses conseils et son enthousiasme qu'il a su me communiquer durant ces dernières années. En outre je n'oublierai jamais qu'il n'a pas hésité à m'accueillir dans son équipe alors que je sortais du service militaire.

Je suis très honoré que Monsieur **J. FAUCHER**, Professeur à L'ENSEEIHHT ait accepté de juger ce travail en tant que rapporteur.

Je tiens à assurer de ma reconnaissance le Professeur **D. MEIZEL** de L'UTC de Compiègne pour m'avoir fait l'honneur d'être rapporteur de ce mémoire.

Qu'il me soit permis de remercier Monsieur **J. BUISSON**, Professeur à SUPELEC de Rennes, pour l'honneur qu'il me fait de bien vouloir juger cette thèse.

Je voudrais remercier également Monsieur **Jean-Pierre RICHARD**, Professeur à l'école centrale de Lille, pour l'honneur qu'il me fait en acceptant d'examiner ce travail, et aussi de ses discussions fructueuses que nous avons eu.

Je suis très heureux de la présence de Madame le Professeur **B. SEMAIL** de l'université de Lille I, qui nous honore de sa présence et de sa participation dans ce jury.

Je tiens aussi à remercier toute **l'équipe de LAIL**, et plus particulièrement l'ensemble des thésards pour l'ambiance de travail chaleureuse et amicale qui se renouvelle chaque année et pour les nombreux coups de main de chacun. Mes remerciements s'adressent aussi à tous mes amis et à tous ceux qui ont contribué à ce travail d'une façon direct ou indirecte, continue ou discontinuée.

Enfin je voudrais remercier les membres de l'Ecole Centrale de Lille qui m'ont aidé dans mes activités d'enseignant et de chercheur.

# **Introduction Générale**

# Introduction Générale

Dans de nombreux domaines d'application, tels que l'électrotechnique (convertisseurs de puissance) ou l'hydraulique (commandes de vanne en tout ou rien), les processus physiques peuvent être représentés par des systèmes à entrées booléennes. Ceux-ci constituent la classe particulière des systèmes hybrides qui évoluent sous l'effet de l'interaction d'un système continu (à temps continu ou à temps discret) et d'un système à événements discrets. Dans notre cas, la partie continue est représentée par une équation différentielle alors que la partie discrète intervient par la nature des commandes, binaire et de plus booléenne. Le modèle considéré s'écrit sous la forme générale  $\dot{x} = f(x, u_1, u_2, \dots, u_m)$  avec  $u_i \in \{0,1\}$ .

Peu de travaux existent dans la littérature sur la détermination de lois de commande à partir de tels modèles [Sira-Ramirez 1987,1989], [Abadie 1994]. Une des raisons évoquées pourrait être la difficulté rencontrée pour construire un modèle sous cette forme, qui regroupe sous un modèle unique toutes les configurations possibles du système, quel que soit l'état des éléments de commutations. La démarche la plus classique est de construire une représentation multi-modèle avec un modèle par état, ce qui donne  $2^m$  modèles s'il y a  $m$  éléments de commutation. Cette dernière approche est tout à fait satisfaisante pour la simulation, mais ne peut être utilisée directement pour le calcul de lois de commande, si ce n'est par des approches type MLI en électronique de puissance ou commande multi-modèle [Dubois 1995].

Les travaux réalisés au LAIL sur la modélisation par bond graph des éléments de commutation ont permis d'obtenir de façon systématique un modèle unique. Ils reposent principalement sur le principe qu'un élément de commutation n'est pas seulement un élément logique tout-ou-rien, mais qu'il possède aussi un comportement énergétique propre (dissipation ou stockage d'énergie propre), ce qui permet de le prendre en compte dans l'écriture des lois de comportement dynamique.

Ce mémoire se décompose en quatre chapitres :

Dans le **Chapitre I** nous présentons la classe particulière des systèmes à entrées booléennes ainsi que les méthodes de modélisation associées. Les différentes techniques de commande applicables aux systèmes étudiés, existant dans la littérature, sont ensuite rappelées.

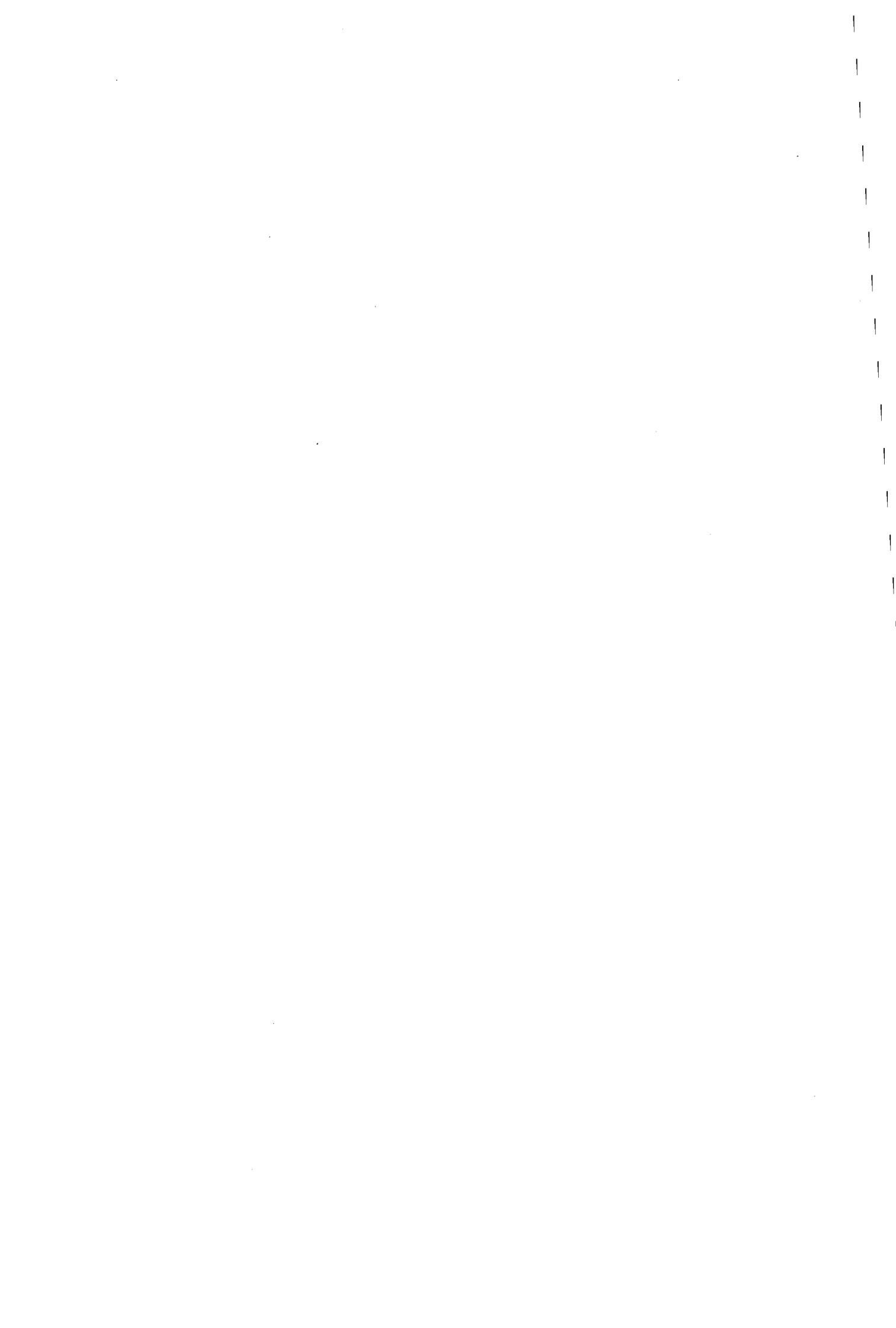
Le **Chapitre II** concerne l'élaboration d'une loi de commande originale pour faire de la régulation et du suivi de trajectoire dans l'espace d'état. Puis, nous étudions la stabilité de cette loi de commande dans le cas des systèmes linéaires en l'état. Enfin, nous abordons le problème des pertes de puissance par commutation dans les interrupteurs. Pour pallier à cette difficulté, nous définissons un algorithme minimisant ces pertes, fondé sur la distance de Hamming. Cet algorithme est appliqué à deux exemples de circuits électroniques.

Au cours du **Chapitre III**, nous montrons comment le principe de la loi de commande proposé au chapitre II peut être transposé aux réseaux de neurones par une méthode de classification. Nous rappelons tout d'abord l'architecture et le fonctionnement d'un réseau de neurones. Puis, la mise en oeuvre d'une commande booléenne par réseau de neurones sera abordée : nous examinons la structure, l'initialisation, l'apprentissage et la commande du réseau de neurones. Finalement, le réseau de neurones est implanté en simulation sur un modèle de moteur synchrone avec son convertisseur de puissance associé.

Dans le **Chapitre IV**, une comparaison en simulation de différents algorithmes de commande booléenne sur un exemple d'association machine asynchrone-onduleur est effectuée. Nous comparons les résultats obtenus par les commandes que nous proposons à ceux obtenus par les commandes classiques par MLI et modes glissants. Pour cela des critères de comparaison spécifiques sont mis en place.

En conclusion générale, nous dressons un bilan des différents résultats proposés dans ce mémoire, puis nous présentons quelques perspectives qui peuvent être envisagées pour la suite de ce travail. Dans le souci de ne pas alourdir la présentation de ce mémoire, nous avons préféré reporter en annexes certaines démonstrations du chapitre I et du chapitre IV.

# Chapitre I



# I. Les Systèmes à Entrées Binaires

## I.1 Introduction

De nombreux processus dynamiques présentent des discontinuités dans leur évolution dans le temps : les vannes ou valves en hydraulique, les butées en mécanique ou les convertisseurs de puissance en électronique. Ces domaines d'application variés témoignent d'une utilisation très répandue. Ces processus non linéaires sont caractérisés de façon générale par les équations différentielles ordinaires.

$$\dot{x} = f(x, u); \quad (\text{I-1})$$

avec

$$x \in \mathcal{R}^n \text{ et } u = [u_1, u_2, \dots, u_m]^T, \quad (\text{I-2})$$

$$u_i \in \{0, 1\} \text{ ou } \{u_{i, \min}, u_{i, \max}\}. \quad (\text{I-3})$$

La commande  $u$  de ces systèmes non linéaires est très restrictive. En effet elle ne peut prendre que deux valeurs. Nous nous intéressons ici aux commandes à valeurs booléennes 0 et 1, symbolisant par exemple l'ouverture ou la fermeture des organes de commande tels que les vannes ou les interrupteurs.

L'objectif du travail présenté dans ce mémoire est de déterminer la commande  $u$  permettant de piloter le système pour une poursuite de trajectoire ou une régulation autour d'une consigne fixée.

La commande peut prendre une des configurations suivantes :

$$\{Config_i(u) / i = 1 \dots 2^m\} = \left\{ \begin{array}{ccccccc} \begin{pmatrix} 0 \\ 0 \\ 0 \\ \vdots \\ 0 \end{pmatrix} \begin{pmatrix} 1 \\ 0 \\ 0 \\ \vdots \\ 0 \end{pmatrix} \begin{pmatrix} 0 \\ 1 \\ 0 \\ \vdots \\ 0 \end{pmatrix} & \dots & \begin{pmatrix} 0 \\ 0 \\ 0 \\ \vdots \\ 1 \end{pmatrix} \begin{pmatrix} 1 \\ 1 \\ 0 \\ \vdots \\ 0 \end{pmatrix} \begin{pmatrix} 1 \\ 0 \\ 1 \\ \vdots \\ 0 \end{pmatrix} & \dots & \begin{pmatrix} 1 \\ 0 \\ 0 \\ \vdots \\ 1 \end{pmatrix} & \dots & \begin{pmatrix} 1 \\ 1 \\ 1 \\ \vdots \\ 1 \end{pmatrix} \\ \text{1} & \text{2} & \text{3} & \dots & \text{m+1} & \text{m+2} & \text{m+3} & \dots & \text{i} & \dots & \text{2}^m \end{array} \right\}; \quad (\text{I-4})$$

Il y a donc  $2^m$  possibilités de vecteurs de commande différents avec  $m$  représentant le nombre d'interrupteurs.

La modélisation sous forme de l'équation (I-1) de ce type de processus physique présentant des commutations est une tâche délicate. Des travaux pour modéliser ces systèmes à entrées binaires sont menés au sein du LAIL en collaboration avec le laboratoire L2EP. Ces travaux concernent la modélisation des convertisseurs de l'électronique de puissance par l'approche Bond-Graph.

Au cours de ce chapitre, nous présenterons la classe de systèmes à étudier ainsi que les méthodes de modélisation associées. Ensuite, nous exposerons les différents travaux qui existent sur la commande binaire dans la littérature. Parmi ces techniques nous présenterons tout d'abord une approche de la commande booléenne étudiée dans [ABADIE 1994], puis nous exposerons les commandes classiques, telles que la commande par mode glissant des systèmes à structure variable, ou les commandes d'électronique de puissance comme la Modulation de Largeur d'Impulsion ou MLI (PWM en anglais).

## I.2 Les systèmes à commandes booléennes

### I.2.1 Systèmes Hybrides

Les processus à entrées discontinues, que nous considérons ici, peuvent entrer dans le contexte des systèmes hybrides. Les systèmes dynamiques hybrides sont des systèmes composés de parties dynamiques continues et de parties à événements discrets [Buisson 1993][Buisson 1998], [Iqbal 1996]. Les systèmes continus sont modélisés par des équations différentielles alors que les systèmes à événements discrets sont généralement modélisés par automates ou réseaux de Pétri. On considère dans notre mémoire un type particulier de systèmes dynamiques hybrides : **les systèmes à entrées binaires**. Prenons un exemple en électronique de puissance pour illustrer le type de système hybride que nous étudierons dans ce mémoire. Considérons l'ensemble convertisseur de puissance et machine électrique comme processus hybride constitué de l'association entre :

- un processus continu : l'alimentation et le processus électromécanique (machine électrique),
- un processus à événements discrets : un ensemble d'interrupteurs statiques.

L'ensemble doit permettre la maîtrise des grandeurs continues (courant et éventuellement vitesse et position) à travers des séquences de reconfiguration des circuits

---

électriques grâce à la fermeture ou à l'ouverture d'interrupteurs judicieusement choisis en fonction des états de l'alimentation et / ou du processus électromécanique.

### **I.2.2 Modélisation des systèmes à commutation**

De nombreux travaux ont été menés sur la modélisation des systèmes à commutation. Ce type de système discontinu n'est pas simple à modéliser. Les travaux existant sur la modélisation de ces systèmes dans la littérature font apparaître deux classes : les méthodes conduisant à une représentation multi-modèles, et les méthodes conduisant à une représentation à modèle unique valable quelque soit l'état des commutateurs.

La modélisation multi-modèles fournit une équation d'état pour chaque topologie du circuit, c'est à dire pour chaque configuration du système (position des interrupteurs). L'inconvénient de cette approche est d'avoir  $2^m$  modèles si  $m$  est le nombre d'interrupteurs. La difficulté réside alors dans la détermination de la commande.

Les travaux menés au LAIL appartiennent à la deuxième classe de méthodes et utilisent l'approche Bond-Graph. Cette technique de modélisation a été créée par Paynter [Paynter 61] puis développée par Rosenberg et Karnopp [Rosenberg, Karnopp 83]. Elle permet une représentation graphique concise et complète des mécanismes d'échange de puissance entre deux sous-systèmes interconnectés, utilisant un langage unique quel que soit le domaine physique à considérer. De plus, ce formalisme est valable en linéaire et en non-linéaire. En conséquence le Bond-Graph est un outil privilégié pour l'analyse des systèmes pluridisciplinaires et permet une déduction immédiate des équations mathématiques du système.

A partir de ce modèle de connaissance, on peut directement implanter les équations en simulation avec les logiciels tel que MATLAB SIMULINK ou, directement en utilisant la représentation du graphe avec 20-SIM. Examinons à titre d'exemple comment la modélisation Bond-Graph nous donne un modèle d'un composant en électronique de puissance.

### **I.2.3 Exemple de modélisation par bond-graph**

Nous avons choisi de représenter les commutateurs d'électronique de puissance à partir des éléments bond-graph (voir annexe A). Ce type de composant peut être décrit par

l'association de deux éléments; le premier élément indique la perte de puissance en conduction, le second caractérise la logique de commutation représentée par une variable booléenne [Dauphin-Tanguy et al 1993], [Ducreux et al, 1992].

Considérons l'élément de commutation représenté (Figure I-1 ) par :

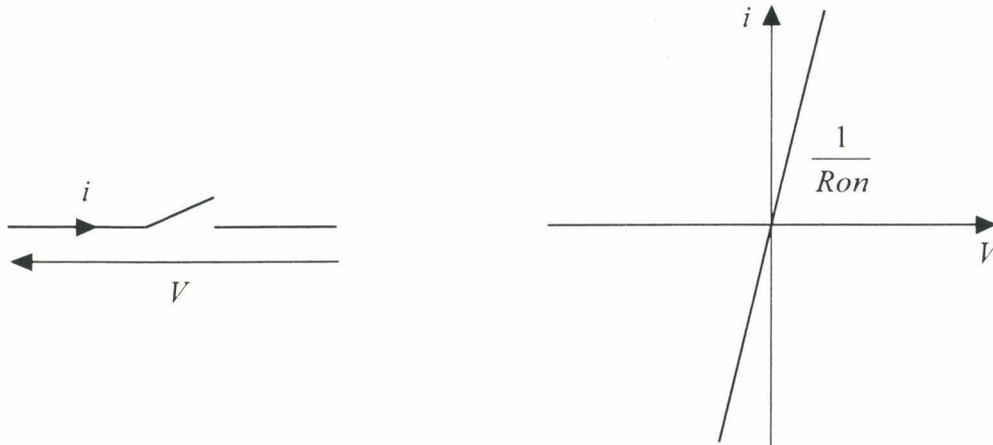


Figure I-1: Caractéristique d'un élément de commutation non idéal.

$R_{on}$  représente la résistance interne de l'élément de commutation à l'état passant.

La conductance interne de l'interrupteur, lorsqu'il est bloqué, est supposée négligeable, ce qui n'est pas le cas de la résistance lorsque le composant est à l'état passant.

La méthode Bond-Graph qui a été proposée dans [Dauphin-Tanguy 1993] est schématisée Figure I-2.

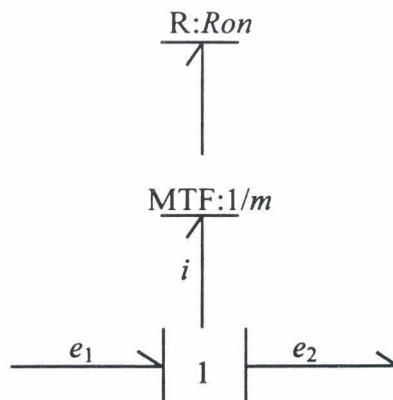


Figure I-2 : Caractéristique d'un élément de commutation.

$m$  est un booléen, prenant la valeur 0 lorsque le composant est bloqué et la valeur 1 lorsque celui-ci est passant. On peut en déduire d'après la méthodologie bond-graph l'équation de ce modèle :

$$i = \frac{m^2}{R_{on}}(e_1 - e_2). \quad (I-5)$$

ce qui correspond à la caractéristique statique de la Figure I-1.

La différence entre  $e_1$  et  $e_2$  représente la tension aux bornes du composant. Le modèle présenté reste valide quels que soient les composants électroniques utilisés, tels que les diodes, transistors, ou les thyristors, la distinction se faisant au niveau de la logique de commutation (cf. Figure I-3), et donc sur les transitions de  $m$ . On peut donc ajouter des liens permettant de représenter les informations nécessaires à la commande de l'interrupteur représentée par le booléen  $m$ .

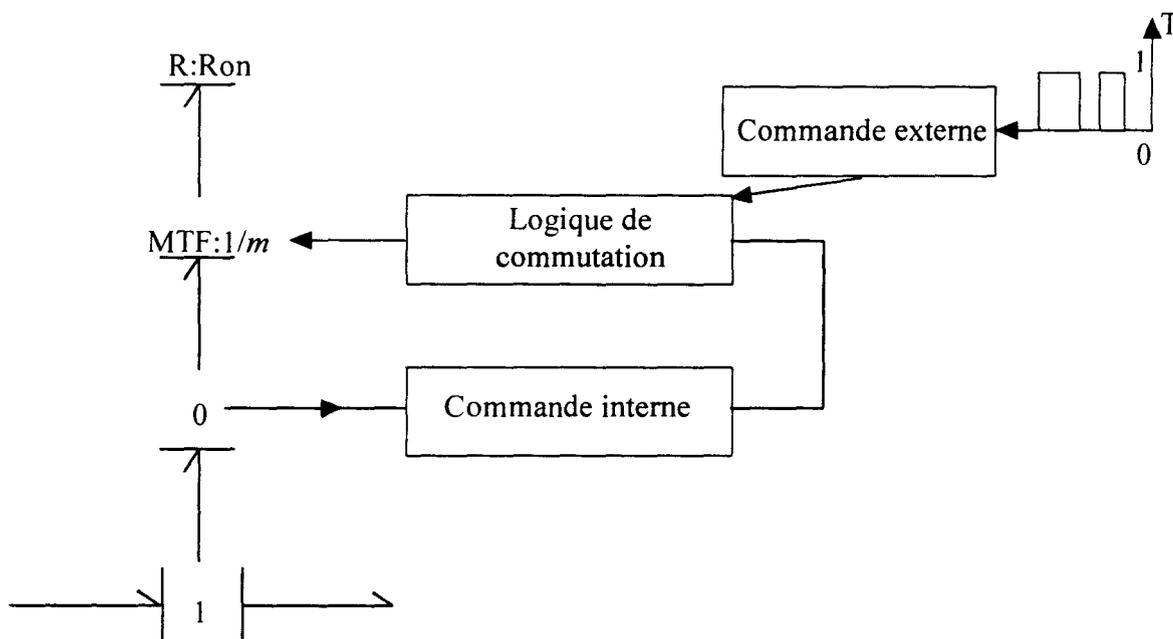


Figure I-3 : Représentation de la commande d'un composant de commutation.

On peut citer en exemple la diode avec son fonctionnement interne (diode passante si la tension à ses bornes est positive) et le transistor de commutation avec la commande interne (tension à ses bornes positive) et la commande externe ( $T=1$ ). Cette représentation bond graph peut être étendue à d'autres domaines de la physique tels que les vannes ou valves en hydraulique.

Le choix d'une affectation unique de la causalité, valable quels que soient les états des interrupteurs conduit à un bond graph unique, et donc un modèle d'état unique, valable quelle que soit la configuration du système. Dans le cas des circuits électroniques le modèle s'exprime sous la forme de l'équation d'état suivante :

$$\dot{x} = f(x, E, m_1 \cdots m_i \cdots m_m); \quad (I-6)$$

où le vecteur d'état  $x$  est composé des flux magnétiques dans les bobines et des charges dans les condensateurs.  $E$  est un vecteur regroupant les sources de puissance et les  $m_i$  ( $i=1\dots m$ ) sont les variables associées à la logique des composants de commutation et servent à commander le système.

#### I.2.4 Conclusion

Cette partie a montré qu'il est possible de modéliser simplement les interrupteurs avec les bond graphs malgré les difficultés dues aux problèmes de commutation. Tout comme la représentation usuelle des circuits électriques, cette modélisation permet de voir la structure d'un système. Le modèle ainsi obtenu permet d'aboutir à une équation d'état grâce à une procédure automatique qui fournit les variables d'état et la mise en équation. De plus, cette équation d'état est unique pour toutes les topologies du circuit c'est à dire pour chaque état différent du système (position des interrupteurs). Cet aspect d'unicité est très important puisqu'il nous permet l'étude de différentes commandes à appliquer aux interrupteurs.

La commande se fera par l'intermédiaire des booléens  $m_i$ , on obtient alors des modèles d'état de la forme suivante :

$$\dot{x} = f(x, u, E) \quad (I-7)$$

où  $u$ , est le vecteur de commande à composantes booléennes ( $m_i$ ).

## I.3 Les différentes commandes booléennes

Après ce rappel de la modélisation bond-graph, nous allons effectuer un inventaire des travaux existant dans le domaine de la commande booléenne pour la régulation autour d'un état désiré  $x_{ref}$  ou le suivi de trajectoire  $x_{ref}(t)$ . L'étude bibliographique qui a été réalisée a montré qu'il y avait peu de travaux effectués dans ce domaine. On peut cependant dégager quatre grandes catégories :

- *Les méthodes qui déterminent la commande sous forme booléenne directement.*
- *Les méthodes conduisant à une commande binaire à ramener dans  $\{0,1\}$ .*
- *Les méthodes de commande par valeur moyenne d'une commande continue.*
- *Les méthodes reposant sur la binarisation d'une commande continue.*

Dans les paragraphes suivant nous allons présenter ces différentes catégories d'approche.

### I.3.1 Détermination directe de la commande booléenne

Les méthodes qui constituent cette première catégorie sont fondées sur un calcul direct de la commande à partir du modèle à entrées booléennes (I-1). La commande directe de couple (DTC) peut être répertoriée dans cette classe. Celle-ci est exclusive à un domaine bien particulier d'application puisqu'elle est utilisée en électrotechnique pour la commande de machines à induction. Cette méthode est définie à partir de l'erreur en couple et du flux statorique de la machine. Le flux statorique est représenté dans l'espace d'état formé du repère  $d-q$  (repère lié au stator de la machine). A l'intérieur de ce repère un cercle de flux de référence a été divisé en 6 secteurs. Ces secteurs sont définis à partir du modèle et des configurations  $Config_i(u)$  (I-4). Il s'agit alors de déterminer la position du vecteur flux statorique dans ce repère et le secteur dans lequel il se trouve. Un algorithme donne l'état des interrupteurs qui sont répertoriés dans une table suivant la valeur du secteur et les valeurs des signaux erreurs (couple et flux statoriques). Nous ne détaillerons pas cette commande dans ce mémoire mais pour plus de détails le lecteur pourra se référer aux ouvrages suivants : [Grellet 1996][Casadei 1997][Depenbrock 1988]. Une autre technique originale qui fait partie de cette catégorie a été développée par V.Abadie [Abadie 1994].

Cette méthode repose sur l'élaboration de séquences périodiques des entrées à partir d'une loi de commande en boucle ouverte et en boucle fermée.

Nous rappellerons la méthodologie proposée [Abadie 1994] pour élaborer l'algorithme de commande pour les systèmes à entrées booléennes. Le cas des systèmes linéaires sera envisagé. Ce cas linéaire peut être transposé aux systèmes non-linéaires par le biais d'une linéarisation par morceaux du système étudié. Cette linéarisation se fait sur chaque intervalle de temps autour du point de fonctionnement courant.

Dans ce paragraphe nous rappellerons tout d'abord, le concept de domaine de convergence du système défini dans [Abadie 1994]. Nous présenterons ensuite l'élaboration de la loi de commande en boucle ouverte à l'intérieur de ce même domaine. Enfin, nous verrons la mise en œuvre de la régulation en boucle fermée.

**1.3.1.1 Présentation**

Le modèle du système est donné par l'équation d'état suivante :

$$\dot{x} = Ax + Bu . \tag{I-8}$$

où  $x = (x_1 \dots x_n)^T \in \mathbb{R}^n$  est le vecteur d'état et  $u = (u_1 \dots u_m)^T \in \{0,1\}^m$  est le vecteur entrée composé de variables booléennes.

**1.3.1.2 Domaine de convergence**

Nous appellerons de manière équivalente le domaine de convergence du système, ou le domaine de consigne admissible de l'état d'un modèle, un domaine fermé dont tous les points peuvent être atteints par le vecteur état du modèle en un temps infini.

Evaluons ce domaine :

Dans la théorie des équations différentielles linéaires,  $x(t)$  est exprimé, en fonction du temps par :

$$x(t) = \underbrace{e^{A(t-t_0)} x(t_0)}_{\substack{\text{régime libre à la} \\ \text{condition initiale par } x(t_0)}} + \underbrace{\int_{t_0}^t (e^{A(t-\theta)} Bu(\theta)) d\theta}_{\text{régime forcé par } u(t)} \quad \text{pour } t > t_0 . \tag{I-9}$$

Le vecteur  $u$  peut admettre  $2^m$  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 :

$$u = [Config_i(u)]; \quad (I-10)$$

avec  $i$  valeur quelconque de  $\{1 \dots 2^m\}$ . On exprime la solution sur cette période  $\tau \in [t_k; t_{k+1}]$  par :

$$x(\tau) = e^{A(\tau-t_k)}x(t_k) + \left[ \int_{t_k}^{\tau} (e^{A(\tau-\theta)} d\theta) \right] B [Config_i(u)]. \quad (I-11)$$

Nous allons supposer dans tout ce qui suit que la matrice  $A$  est non singulière. L'expression (I-11) s'exprime alors, pour  $\tau \in [t_k; t_{k+1}]$ , par :

$$x(\tau) = e^{A(\tau-t_k)}x(t_k) - A^{-1}(I - e^{A(\tau-t_k)})B [Config_i(u)]. \quad (I-12)$$

Ce qui s'écrit :

$$x(\tau) = e^{A(\tau-t_k)}x(t_k) - A^{-1}(I - e^{A(\tau-t_k)})B_i; \quad (I-13)$$

en posant  $B_i = B [Config_i(u)]$ .  $I$  est la matrice identité de dimension  $n$ .

Si  $A$  est supposé asymptotiquement stable, alors quand  $t \rightarrow +\infty$ , on a  $\dot{x}_i = 0$  ce qui entraîne :

$$x_{i\infty} = -A^{-1}B_i. \quad (I-14)$$

L'ensemble des points  $\{x_{1\infty} \dots x_{2^m\infty}\}$  forme les sommets d'un polyèdre  $\mathcal{P}_x$  convexe, défini par le système. Ce polyèdre  $\mathcal{P}_x$  convexe peut être aussi appelé **le domaine de convergence du système**.

### I.3.1.3 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 stabilité défini précédemment.

Soient  $x_d \in \mathcal{P}_x$  la consigne désirée pour le vecteur état et  $t_0$  l'instant initial.

Soient  $\{a_1, \dots, a_{2^m}\}$  les coordonnées barycentriques de  $x_d$  dans le polyèdre  $\mathcal{P}_x$  délimité par les points  $\{x_{1\infty}, \dots, x_{2^m\infty}\}$ , telles que :

$$\begin{cases} x_d = \sum_{i=1}^{2^m} a_i x_{i\infty}; \\ \sum_{i=1}^{2^m} a_i = 1; \\ a_i \in [0,1], i \in \{1, \dots, 2^m\}. \end{cases} \quad (I-15)$$

La commande proposée pour commander l'état du système vers l'objectif fixe  $x_d$  est une commande périodique. En effet la durée d'application de la configuration  $Config_i(u)$  est  $a_i T_{mod}$  comme le montre la Figure I-4.

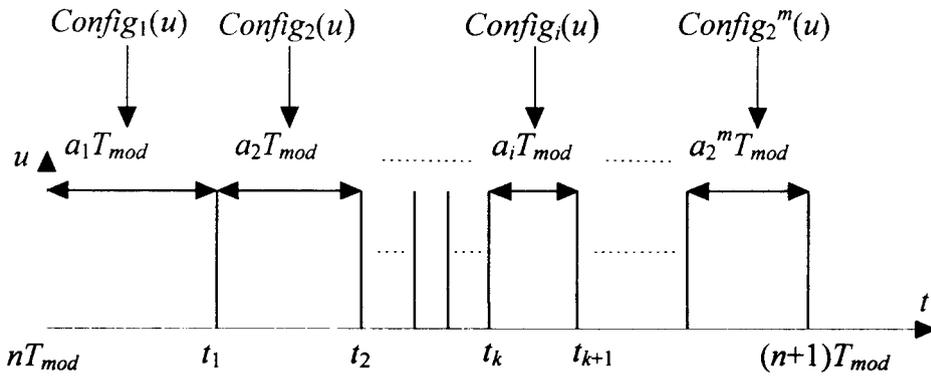


Figure I-4: Représentation de commandes durant une période  $[nT_{mod}, (n+1)T_{mod}]$  la période de modulation  $T_{mod}$  doit être choisie telle que :

$$\|A\| \ll \frac{1}{T_{mod}}; \quad (I-16).$$

où  $\|A\|$  représente une norme de la matrice  $A$ . En effet, puisque  $|\lambda(A)|_{max} \leq \|A\|$ , la condition  $\|A\| \ll 1 / T_{mod}$  est équivalente à  $|\lambda(A)|_{max} \ll 1 / T_{mod}$ .

Chaque  $Config_i(u)$  appartient à l'ensemble  $\{Config_i(u) / i = 1 \dots 2^m\}$  défini par l'équation (I-4). Le choix n'est pas unique et sera fait en fonction de critères à définir.

Avec cette stratégie de commande, le système modélisé par (I-8) converge vers  $x_d$ . Illustrons cette technique de commande par un exemple.

Considérons le système modélisé par l'équation d'état suivante :

$$\begin{cases} \dot{x} = \begin{pmatrix} -2 & 0 \\ 1 & -4 \end{pmatrix} x + \begin{pmatrix} -1 & 2 \\ 1 & 1 \end{pmatrix} u; \end{cases} \quad (\text{I-17})$$

Nous obtenons alors les résultats :

$$\begin{cases} x_{1\infty} = -A^{-1}B \begin{pmatrix} 0 \\ 0 \end{pmatrix} = \begin{pmatrix} 0 \\ 0 \end{pmatrix}; \\ x_{2\infty} = -A^{-1}B \begin{pmatrix} 1 \\ 0 \end{pmatrix} = \begin{pmatrix} -\frac{1}{2} \\ \frac{1}{8} \end{pmatrix}; \\ x_{3\infty} = -A^{-1}B \begin{pmatrix} 0 \\ 1 \end{pmatrix} = \begin{pmatrix} 1 \\ \frac{1}{2} \end{pmatrix}; \\ x_{4\infty} = -A^{-1}B \begin{pmatrix} 1 \\ 1 \end{pmatrix} = \begin{pmatrix} \frac{1}{2} \\ \frac{5}{8} \end{pmatrix}. \end{cases} \quad (\text{I-18})$$

Le domaine de convergence  $\mathcal{P}_x$  peut être représenté par :

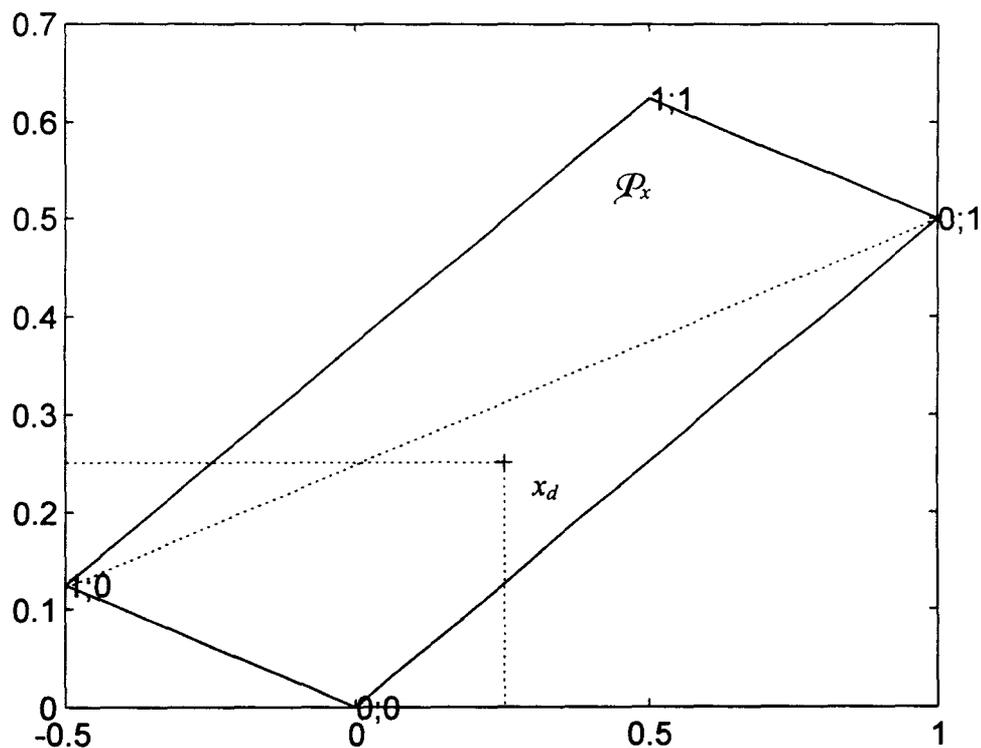


Figure I-5 : Domaine de convergence du système (I-17)

On souhaite que le système converge vers  $x_d = (1/4, 1/4)^T$ .  $x_d$  peut être exprimé en utilisant l'équation (I-15), ce qui donne :

$$\begin{cases} \frac{1}{4} = \frac{-1}{2}a_2 + a_3 + \frac{1}{2}a_4; \\ \frac{1}{4} = \frac{1}{8}a_2 + \frac{1}{2}a_3 + \frac{5}{8}a_4; \\ 1 = a_1 + a_2 + a_3 + a_4. \end{cases} \quad (\text{I-19})$$

Si on exprime les équations précédentes en utilisant  $a_4$  comme paramètre, on obtient :

$$\begin{cases} a_1 = \frac{1}{4} + a_4; \\ a_2 = \frac{1}{3} - a_4; \\ a_3 = \frac{5}{12} - a_4. \end{cases} \quad (\text{I-20})$$

La contrainte  $a_i \in [0,1]$  pour  $i=1,2,3,4$  implique que l'on peut prendre toute valeur de  $a_4$  comprise entre 0 et 1/3. On peut choisir, par exemple,  $a_4=0$  ce qui donne les valeurs suivantes :

$$\begin{cases} a_1 = \frac{1}{4} \\ a_2 = \frac{1}{3} \\ a_3 = \frac{5}{12} \\ a_4 = 0 \end{cases} \quad (\text{I-21})$$

D'après les résultats présentés précédemment, nous devons choisir  $T_{mod}$  tel que  $\|A\| \ll 1/T_{mod}$ . En conséquence si  $\|A\|$  représente la norme duale du  $\max \|A\| = 4$ , alors :

$$T_{mod} \ll \frac{1}{4}. \quad (\text{I-22})$$

L'entrée de commande  $u$  est représentée en fonction du temps par la figure ci-dessous :

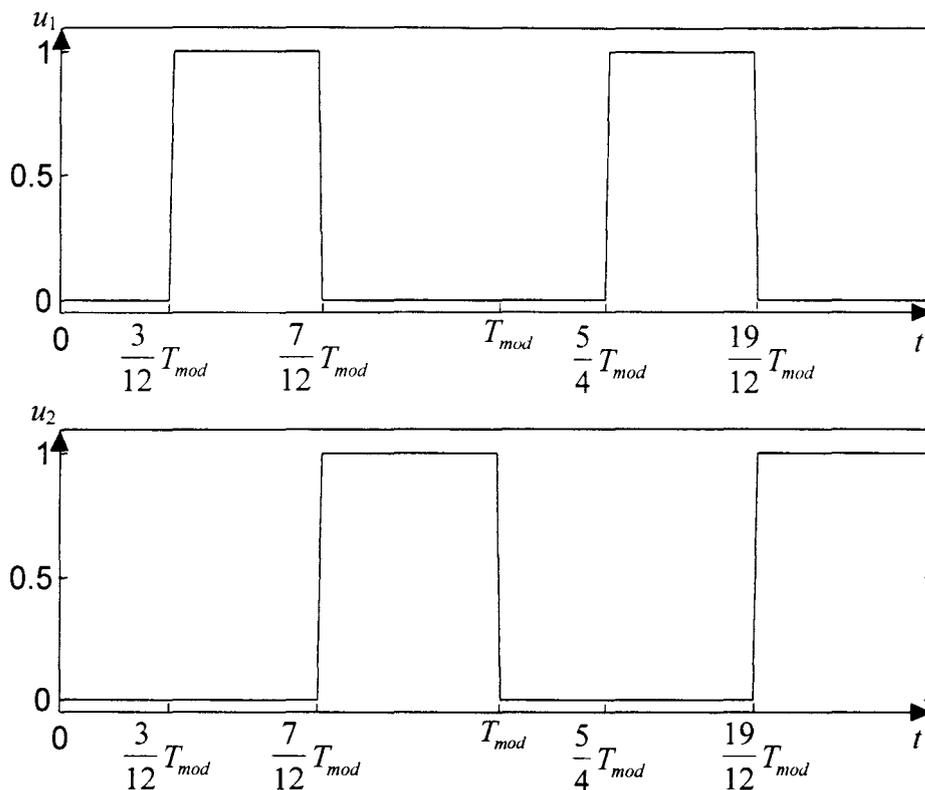


Figure I-6 : Représentation des entrées du système linéaire à commutation.

Les réponses temporelles du modèle ont été simulées en utilisant le logiciel MATLAB-SIMULINK, avec  $T_{mod} = 10^{-3}$  sec :

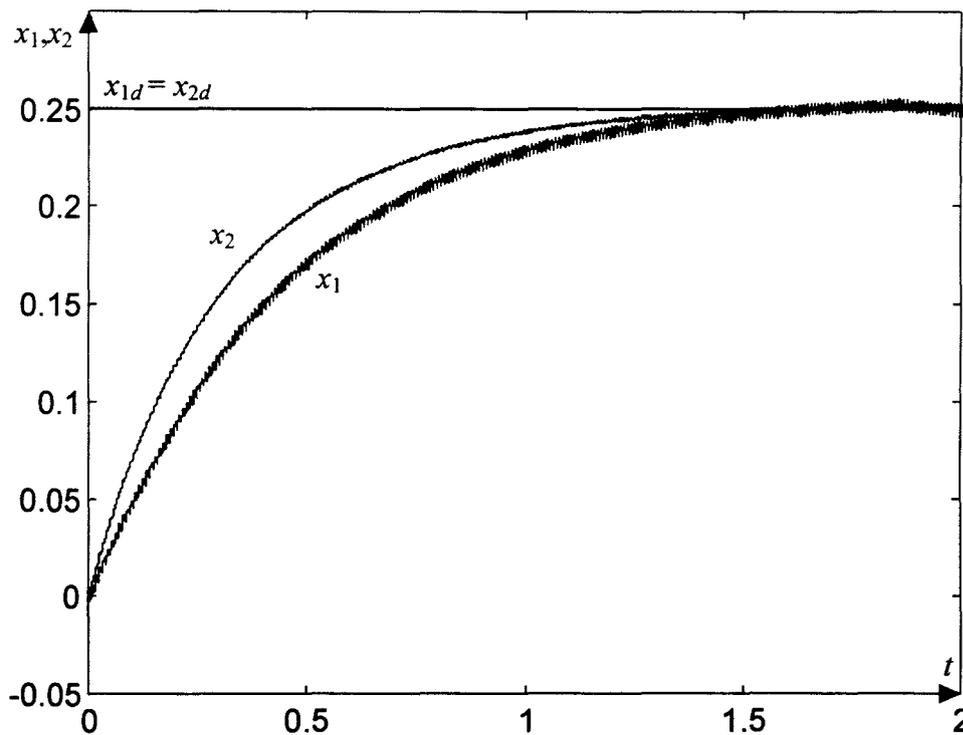


Figure I-7 : Réponse temporelle des variables d'état

### I.3.1.4 Commande en boucle fermée

Le but ici est d'élaborer un algorithme en boucle fermée permettant de poursuivre une trajectoire  $x_d(t)$ . Nous allons échantillonner la trajectoire  $x_d$  à une période  $T_d$ . Considérons  $e = (e_1, \dots, e_n)^T$  un vecteur constant pré-défini et  $D_j$  ( $j = 1, \dots, n$ ) des domaines bornés autour de  $x_{d_j}(t)$  définis par :

$$x_j \in D_j \text{ ssi } \left( \forall j \in \{1, \dots, n\}, |x_j(t) - x_{d_j}(t)| \leq e_j \right). \quad (\text{I-23})$$

L'algorithme de poursuite peut s'écrire de la manière suivante :

1. Détermination de  $\varepsilon_j = (\varepsilon_{d_1}, \dots, \varepsilon_{d_n})^T$  défini pour tout  $j$ , à l'instant  $kT_d$  par :

$$\varepsilon_j(kT_d) = |x_j(kT_d) - x_{d_j}(kT_d)|. \quad (\text{I-24})$$

2. Détermination de la trajectoire de référence  $x_{ref_j}(kT_d)$  (Figure I-8) définie par :

si pour tout  $j \in \{1, \dots, n\}$ ,  $\varepsilon_j((k-1)T_d) \leq e_j$ , alors :

$$x_{ref_j}(kT_d) = x_{d_j}(kT_d);$$

sinon :

$$x_{ref_j}(kT_d) = 2x_{d_j}(kT_d) - x_j((k-1)T_d).$$

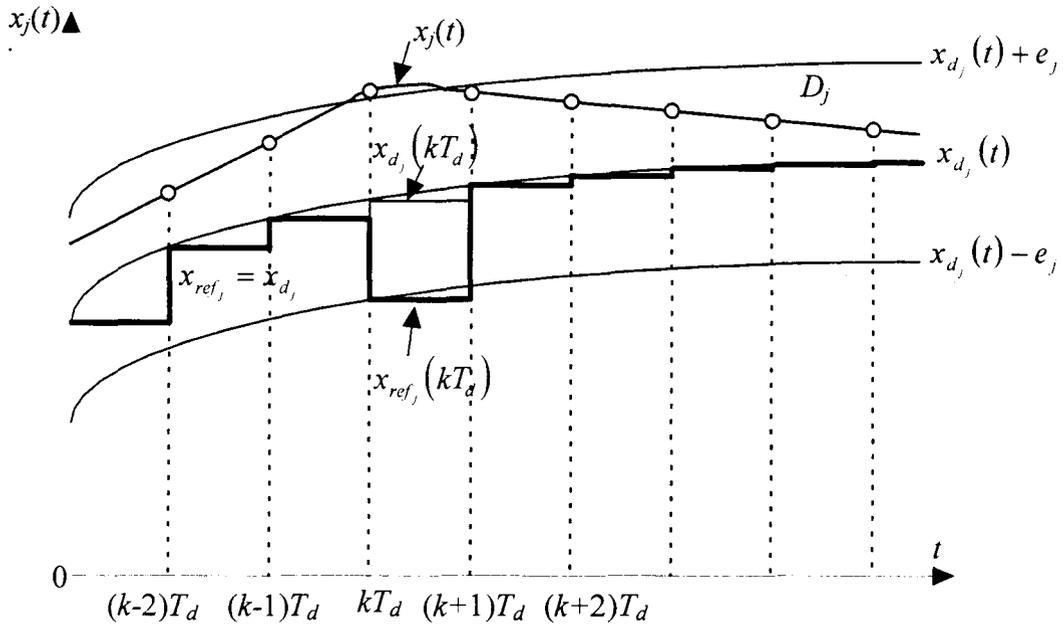


Figure I-8 : Construction d'une trajectoire de référence.

3. Détermination des coefficients  $a_i(k)$  définis par :

$$\begin{cases} x_{ref}(kT_d) = \sum_{i=1}^{2^m} a_i(k)x_{i\infty} \\ \sum_{i=1}^{2^m} a_i(k) = 1 \\ a_i \in [0,1], i \in \{1, \dots, 2^m\} \end{cases} \quad (I-25)$$

Ce système (I-25) admet plusieurs solutions, et donc des critères de choix sont possibles conduisant à une solution unique.

L'algorithme peut se mettre sous forme de schéma-bloc :

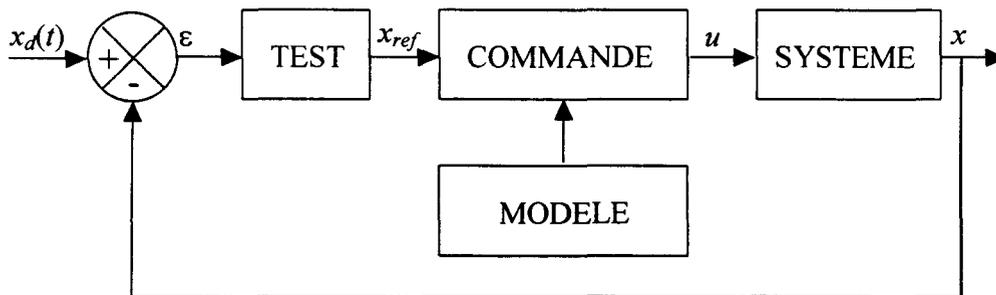


Figure I-9 : Algorithme en boucle fermée

Le bloc TEST détermine  $x_{ref}(t)$  à partir de  $\varepsilon(t)$ .

Le bloc COMMANDE calcule la séquence du vecteur  $u$  à partir de  $x_{ref}(t)$ .

Le bloc MODELE permet de calculer les coordonnées barycentriques  $a_i(k)$  à partir de l'équation (I-25)

### **I.3.2 commande binaire à ramener dans $\{0,1\}$**

La commande par modes glissants [Utkin 1974][Emelyanov 1974][Andronov 1966][Emelyanov 1967], et [Utkin 1977] constitue cette deuxième partie de méthodes employées pour piloter les systèmes à entrées booléennes. Le contrôleur utilisé dans cette commande peut commuter entre plusieurs structures. C'est le cas, par exemple des circuits d'électronique de puissance, où le système est gouverné par une équation différentielle différente pour chaque position de l'interrupteur [Sira-ramirez 1987]. Cette théorie de la commande "tout-ou-rien" constitue un cas particulier de la théorie des Systèmes à Structure Variable (SSV), puisqu'elle a donné naissance aux notions de régimes glissants qui se sont par la suite étendus aux systèmes discontinus réalisés artificiellement. Ces systèmes sont de la forme  $\dot{x} = f(x, u)$  avec  $f$  une fonction non linéaire continue, et pour lesquels on définit un retour d'état discontinu. Cette commande sera abordée tout d'abord dans un cas général, c'est à dire pour les systèmes à entrées continues puis nous l'étendrons aux systèmes à entrées booléennes.

#### ***1.3.2.1 Système à structure variable***

De nombreux processus mécaniques, électriques ou chimiques sont caractérisés par des équations différentielles à second membre discontinu. Ainsi ces problèmes sont caractéristiques de certaines catégories de commandes en automatique. Effectivement, pour certains domaines d'applications, de nombreux actionneurs sont binaires. Les deux seules valeurs possibles sont  $\{0,1\}$ .

D'une manière générale les systèmes à structure variable apparaissent dans deux domaines différents.

Dans le premier cas, le système continu est régi par une équation différentielle continue. Le but alors est de définir une loi de commande sous forme de retour d'état possédant des discontinuités sur une (des) surface(s) donnée(s) dans l'espace d'état. On espère par cette commande rendre la surface attractive. Sous certaines conditions, l'état du système se

maintient sur la surface  $S = \{x \in \mathcal{X} : s(x) = 0\}$ , le régime est alors en mode glissant. Sa dynamique ne dépend que du choix de la variété  $S$ .

Dans le deuxième cas, le système est discontinu par nature, c'est le cas par exemple des circuits électriques à commutateurs, qui prennent les valeurs 0 ou 1. L'objectif de la loi de commande est alors défini comme dans le premier cas où la dynamique dépend des coefficients de la variété.

Dans les deux cas, on cherchera à définir l'équation quand l'état évolue sur la surface  $S$ . Ce problème ressort de la théorie des équations différentielles à second membre discontinu dont la théorie des régimes glissants constitue un "champ d'application dans la théorie de la commande". Nous allons décrire cette commande sur un système linéaire au prochain paragraphe. Pour de plus amples renseignements sur la théorie utilisée, et l'extension aux systèmes non linéaires le lecteur pourra se référer à l'annexe B et aux documents suivants [Utkin 1977][Fossart 1991] [Filipov 1960][Filipov 1988].

### ***1.3.2.2 Cas général du mode glissant***

Dans cette partie nous présenterons la commande par mode glissant sur les systèmes linéaires artificiellement discontinus. Par ce terme, nous désignons les systèmes dont la commande est continue par nature et qui sont rendus artificiellement discontinus par le contrôleur.

#### *1.3.2.2.1 Systèmes Multivariables*

Considérons le système linéaire multivariable suivant :

$$\dot{x} = Ax + Bu; \quad (\text{I-26})$$

où  $u = [u_1, \dots, u_m]^T$  est le vecteur de commande,  $x = [x_1, \dots, x_n]^T$ , le vecteur d'état. Prenons un ensemble de  $m$  surfaces de commutation  $s(x) = (s_1(x), \dots, s_m(x))$  définies par :

$$s(x) = Gx; \quad (\text{I-27})$$

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

$$u_i = \begin{cases} u_i^+ & \text{si } s_i(x) > 0; \\ u_i^- & \text{si } s_i(x) < 0. \end{cases} \quad (\text{I-28})$$

Le vecteur  $\dot{x}$  est discontinu sur le plan. Par conséquent si les trajectoires sont dirigées vers  $s=0$ , le mode de glissement apparaît sur ce plan. La condition suffisante pour que le mode glissant [Filipov 1960] existe est donné par :

$$s_i \dot{s}_i \leq 0. \tag{I-29}$$

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

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

$$0 = GAx + GBu. \tag{I-30}$$

On définit alors une commande équivalente  $u_e$  telle que :

$$\begin{cases} u_e = -(GB)^{-1}GAx; \\ \text{Avec } (GB)^{-1} \text{ inversible.} \end{cases} \tag{I-31}$$

Le mouvement obéit alors à l'équation :

$$\dot{x} = [A - B(GB)^{-1}GA]x. \tag{I-32}$$

Le comportement du système ne dépend pas des valeurs des commandes  $u_i^+$  et  $u_i^-$ . Celles-ci ne servent qu'à assurer les conditions de stabilité autour des surfaces  $s_i = 0$ . La matrice définie par  $[A - B(GB)^{-1}GA]$  a  $m$  valeurs propres nulles et  $n-m$  non nulles. La condition de régime de glissement vers le point d'équilibre existe si et seulement si les  $n-m$  valeurs propres associées ont des parties réelles strictement négatives [Utkin 1977]. A titre d'exemple examinons comment la commande par mode glissant s'effectue sur un système mono-entrée.

### 1.3.2.2.2 Système mono-entrée

Reprenons le système modélisé par une équation de la forme (I-26), mais dans le cas monovariante en entrée, c'est-à-dire avec  $u \in \mathbb{R}$ . Supposons que  $u$  est discontinue sur  $s(x)=Gx=0$  :

$$u = \begin{cases} u^+ & \text{si } s(x) > 0; \\ u^- & \text{si } s(x) < 0. \end{cases} \tag{I-33}$$

Le système est en glissement sur la surface  $s(x)=Gx=0$ ; alors, on définit une commande équivalente  $u_e$  telle que  $u_e = -(GB)^{-1}GAx$  (avec  $GB$  inversible). Le mouvement est décrit alors par  $\dot{x} = [A - B(GB)^{-1}GA]x$ , et peut se représenter par la figure suivante :

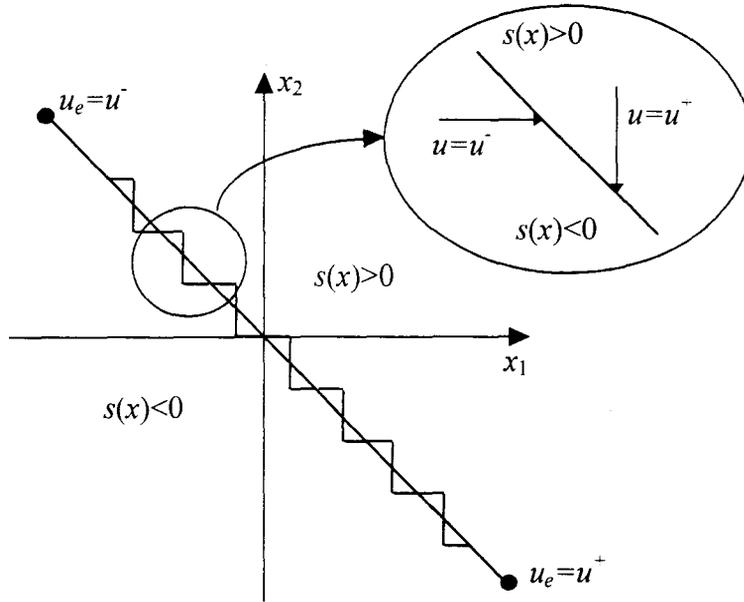


Figure I-10 : Trajectoire de glissement.

On peut interpréter la grandeur de la commande équivalente  $u_e$  comme la valeur moyenne que prend la grandeur de commande lors de la commutation rapide entre  $u^-$  et  $u^+$ .

Dans ce cas monovariante, 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 minimale et maximale de  $u$  (si  $u^+ > u^-$ ) :

$$u^- < u_e < u^+ ; \tag{I-34}$$

Une loi de commande peut être alors construite en vérifiant cette condition. On choisit  $u$  telle que  $u = u_e + (GB)^{-1}u_d$  avec  $u_d$  discontinue ce qui donne  $G\dot{x} = GAx + GBu_e + u_d = u_d$ . La condition  $s\dot{s} \leq 0$  entraîne donc que l'on choisit  $u_d > 0$  pour  $s(x) < 0$  et  $u_d < 0$  pour  $s(x) > 0$ . On peut prendre alors :

$$u_d = -ksign(s(x)). \tag{I-35}$$

avec  $k \in \mathbb{R}^{**}$  et  $sign(s)$  représente la fonction signe de  $s = \begin{cases} sign(s) = +1 & \text{si } s(x) > 0 \\ sign(s) = -1 & \text{si } s(x) < 0 \end{cases}$ ,

et  $u_e$  est bien comprise entre les valeurs minimale et maximale de  $u$ . Finalement la loi de commande est donnée par la relation suivante :

$$u = \begin{cases} u_c + (GB)^{-1} & \text{si } s(x) > 0; \\ u_c - (GB)^{-1} & \text{si } s(x) < 0. \end{cases} \tag{I-36}$$

Le comportement dynamique du système en mode de glissement sera réglé par les coefficients de la surface  $s$ .

### I.3.2.3 Application au cas particulier des systèmes à entrées booléennes

La technique exposée précédemment a été appliquée à de nombreuses reprises en électronique de puissance pour la commande de moteurs en utilisant la MLI pour piloter le convertisseur statique associé [Sabanovic et al 1981][Bose 1985]. D'autre part, A Sabanovic dans [Sabanovic, 1993] propose de commander une association onduleur-moteur en utilisant un modèle multivariable simplifié.

Nous verrons au cours de ce paragraphe qu'il est possible d'appliquer cette méthode directement ( $u^+ = 1, u^- = 0$ ) sur le système à entrées booléennes sans utiliser de technique de commande rapprochée du convertisseur. Cette méthode consistera donc à appliquer l'algorithme de commande sur le système global intégrant les interrupteurs (Figure I-11).

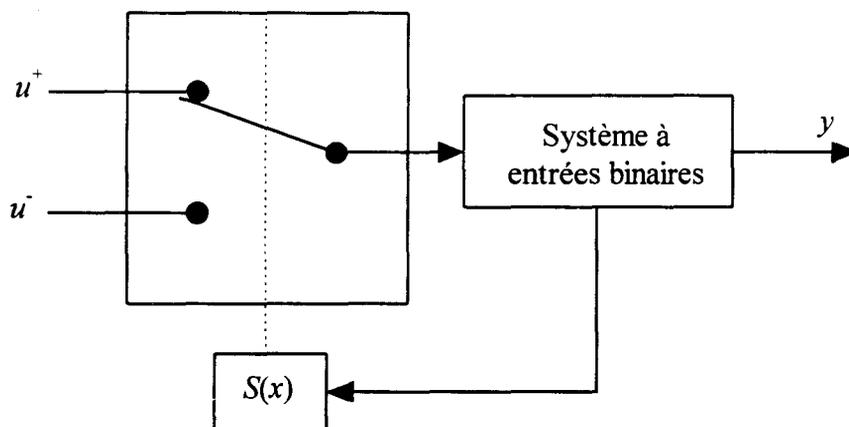


Figure I-11 : Commutation de type relais

Considérons le système linéaire multivariable suivant :

$$\dot{x} = Ax + Bu; \tag{I-37}$$

Avec  $x$  le vecteur d'état dans  $\mathfrak{R}^n$ ,  $u_i \in \{0,1\}$  pour  $i = 1, \dots, m$  et  $A \in \mathfrak{R}^{n \times n}$ ,  $B \in \mathfrak{R}^{n \times m}$  des matrices constantes, avec  $A$  inversible.

La commande est conçue afin que la grandeur  $u_i$  ne prennent que deux valeurs constantes 0 ou 1. La commutation entre ces deux valeurs est imposée par la loi de commutation selon :

$$u_i = \begin{cases} 0 & \text{si } s_i(x) > 0; \\ 1 & \text{si } s_i(x) < 0. \end{cases} \quad (\text{I-38})$$

Dans ce cas aussi, il est possible que la commutation ait lieu à une fréquence très élevée de sorte que le système de réglage travaille en mode de glissement. De plus il faut imposer la condition  $s_i(x)=0$  pour que le comportement dynamique du système en mode glissant existe. La commande équivalente est donc définie, comme précédemment, ce qui donne :

$$u_e = -(GB)^{-1}GAx. \quad (\text{I-39})$$

La condition d'existence du glissement est vérifiée si et seulement si la commande équivalente est comprise entre les bornes minimale et maximale de l'entrée 0 et 1, soit :

$$0 \leq u_e \leq 1. \quad (\text{I-40})$$

Nous remarquons que cette condition est très contraignante. En effet celle-ci indique que le glissement ne pourra avoir lieu sur toutes les valeurs de l'état, mais que l'on est restreint sur un domaine. Le comportement dynamique sera ajusté par les coefficients de la surface de glissement.

Illustrons cette stratégie sur un exemple monoentrée dans le prochain paragraphe.

#### ***1.3.2.4 Exemple***

Considérons le système linéaire monoentrée décrit par :

$$\begin{bmatrix} \dot{x}_1 \\ \dot{x}_2 \end{bmatrix} = \begin{bmatrix} 0 & 1 \\ -1 & 0 \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \end{bmatrix} + \begin{bmatrix} 0 \\ -2 \end{bmatrix} u; \quad (\text{I-41})$$

avec  $u$  booléen.

La logique de commutation est définie par :

$$\begin{cases} u = 0 & \text{si } s(x) = gx_1 + x_2 > 0; \\ u = 1 & \text{si } s(x) = gx_1 + x_2 < 0. \end{cases} \quad (\text{I-42})$$

La commande équivalente (en considérant  $s(x)=0$ ) est :

$$u_c = -\frac{1}{2}(g^2 + 1)x_1; \quad (\text{I-43})$$

et le glissement ne pourra avoir lieu que si

$$0 \leq -\frac{1}{2}(g^2 + 1)x_1 \leq 1 \quad (\text{I-44})$$

c'est-à-dire sur le segment AB représenté sur la figure ci-dessous, avec  $g=1$  :

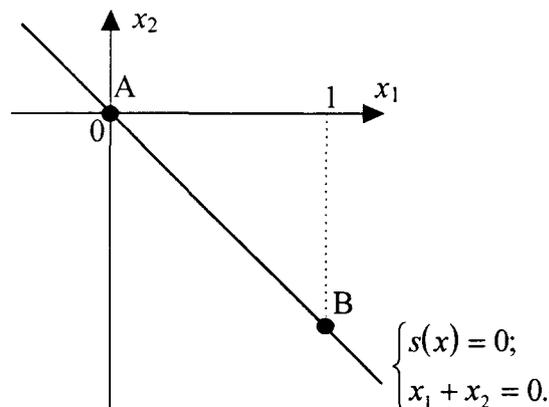


Figure I-12 : Représentation du segment de glissement, dans le plan  $(x_1, x_2)$ .

### I.3.3 Commande par valeur moyenne

Nous présenterons ici deux techniques de commande par valeur moyenne. Ces deux méthodes sont issues des techniques de l'électronique de puissance. Elles peuvent être assimilées par analogie mais pas directement à la commande booléenne. Le but est de piloter des convertisseurs de puissance afin d'obtenir une tension moyenne variable à partir d'une ou plusieurs tensions continues. Ces convertisseurs permettent de convertir comme leurs noms l'indiquent, un signal de basse puissance en un signal de haute puissance exigée par le système électrique.

Parmi ces deux commandes électroniques, nous trouvons tout d'abord la méthode la plus couramment utilisée qui est la MLI (PWM Pulse Width Modulation en anglais) [Leonhard 1985][Bose 1986][Ismail, Erickson 1996][Nonaka, Neba 1987][Faucher 1993].

Le principe est fondé sur la commande rapprochée des éléments de commutation en valeur moyenne de l'entrée de commande. Cette entrée de commande est issue généralement d'un proportionnel-intégral définie à partir du modèle sans convertisseur. Cette stratégie assure un bon suivi de consigne et donne des performances satisfaisantes. La seconde approche est la commande par hystérésis. Celle-ci, par nature allie robustesse du réglage et simplification de l'architecture.

### I.3.3.1 Commande en électronique de puissance.

Nous allons présenter les deux commandes par valeur moyenne utilisées fréquemment en électronique de puissance. Ces méthodes sont un cas particulier des systèmes à entrées booléennes puisqu'elles ne prennent pas en compte dans le modèle le convertisseur proprement dit. Le calcul de la commande se fait de manière classique, c'est-à-dire en considérant l'entrée du système continue.

Pour illustrer ces deux techniques, nous allons considérer l'exemple représenté par la Figure I-13. Cet exemple est un convertisseur de puissance de type hacheur avec une charge de type inductif (comme dans les machines électriques).

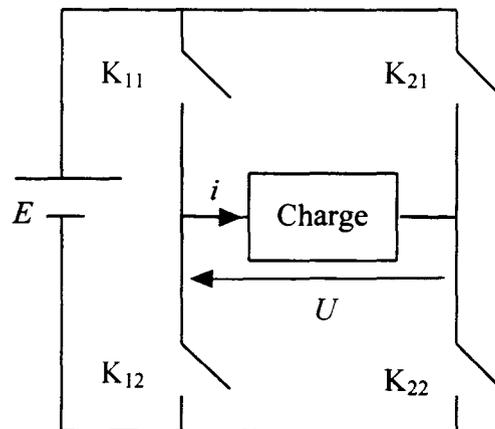


Figure I-13 : Hacheur et charge

Dans ce schéma  $E$  est une tension continue, et les  $K_{ij}$  représentent les transistors de commutation. Le but est de commander la grandeur de sortie qui est représentée par le courant  $i$  dans la charge. La tension  $U$  est le signal de commande de la charge. Ce signal de commande est évalué en valeur moyenne à partir de l'ouverture et la fermeture des interrupteurs. Pour une commande bipolaire (c'est-à-dire telle que la tension ne peut prendre que deux valeurs  $+E$  et  $-E$ ), il n'y a que deux états possibles (les interrupteurs sur la même branche c'est-à-dire  $K_{11}$ ,  $K_{12}$  ou  $K_{21}$ ,  $K_{22}$  sont complémentaires) commandés par

paires d'interrupteurs ( $K_{11}, K_{22}$ ) ; ( $K_{12}, K_{21}$ ). Un exemple de fonctionnement est donné sur la Figure I-14.

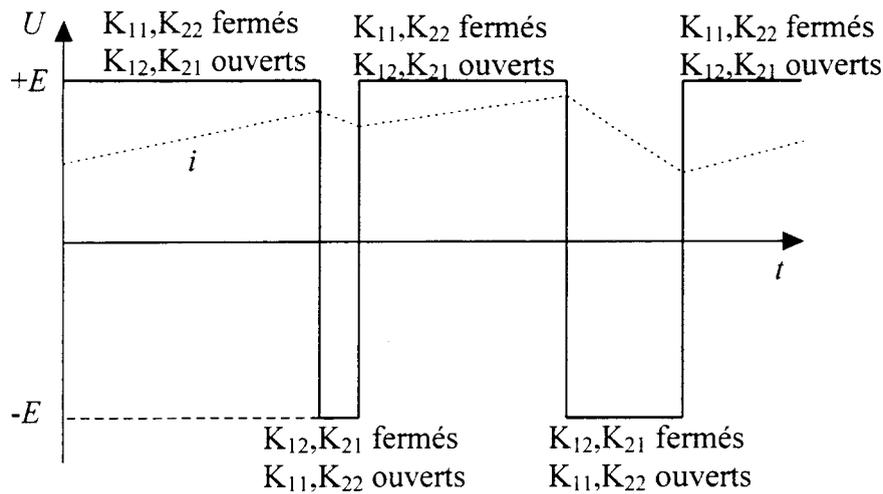


Figure I-14 : Exemple

On peut faire deux remarques à l'examen de cette figure :

- La tension  $U$  est formée par une succession de crêteaux d'amplitude égale à la tension continue d'alimentation et de fréquence variable suivant la méthode de calcul choisie.
- Lorsque la charge présente une constante de temps élevée, l'ondulation du courant de charge est faible.

### I.3.3.2 Commande par M.L.I

Les lois de Modulation de Largeur d'Impulsion sont nombreuses, les techniques de modulation les plus répandues comparent des tensions de référence à un signal triangulaire symétrique de haute fréquence. Aux instants d'égalité on obtient le basculement du comparateur, donc le changement (commutation) d'état logique du signal de commande (Figure I-15). Ce signal triangulaire est appelé porteuse  $U_p$ . Le signal de référence est appelé modulatrice (signal de commande).

Le signal de commande  $U_0$  est calculé en considérant le modèle sans convertisseur. Généralement un proportionnel-intégral (PI) est utilisé pour ce genre de commande.

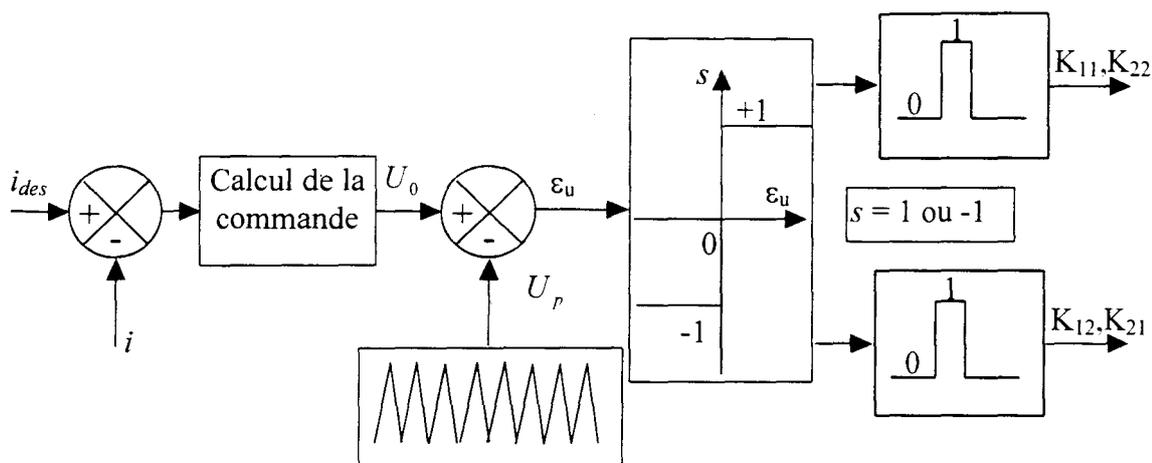


Figure I-15 : Commande par MLI

Regardons maintenant ce qui se passe au cours d'une période du signal triangulaire  $U_p$  (Figure I-16) auquel on superpose une tension de référence  $U_0$ , ces deux signaux se coupent aux points d'abscisses  $\alpha$  et  $2\pi - \alpha$ . Leurs intersections déterminent la valeur moyenne de la tension découpée (Figure I-17). La loi de calcul est décrite de la manière suivante :

- si  $U_p - U_0 < 0$  alors  $U = +E$ ;
- si  $U_p - U_0 > 0$  alors  $U = -E$ .

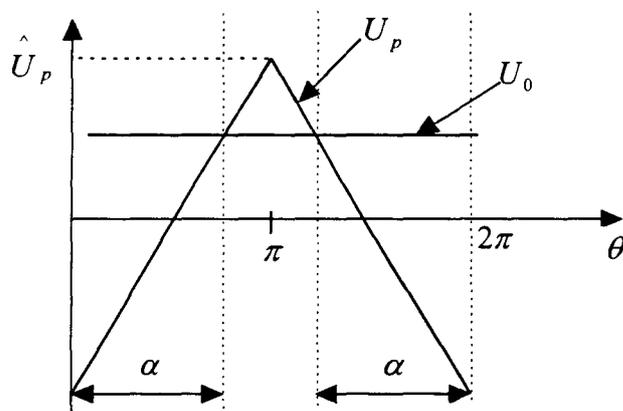


Figure I-16 : Superposition du signal triangulaire et de la tension de référence

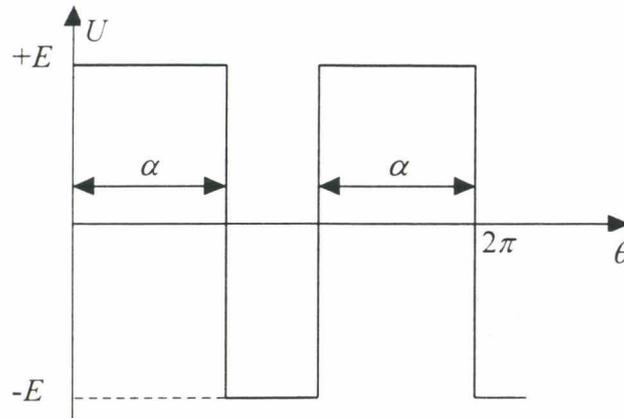


Figure I-17 : Tension découpée

L'expression de la valeur moyenne est  $U_{moy} = E \left( \frac{2\alpha}{\pi} - 1 \right)$ ; d'autre part

$\alpha = \frac{\pi}{2} \left( 1 + \frac{U_0}{\hat{U}_p} \right)$  avec  $\hat{U}_p$  valeur de crête de la porteuse. Par conséquent la valeur

moyenne du signal modulé sur une période de la porteuse est égale à la valeur instantanée au point d'intersection entre la modulante (signal de commande) et la porteuse. On définit

ainsi la valeur moyenne instantanée représentée par :  $\frac{U_{moy}}{E} = \frac{U_0}{\hat{U}_p}$ .

Dans le choix de la porteuse, il y a deux conditions essentielles :

1. La fréquence du signal de la porteuse doit être très supérieure à la fréquence de référence  $f_{u_p} \gg f_{u_0}$  afin d'avoir un bon rendement du convertisseur (c'est à dire une bonne recopie du signal de référence).

2. La valeur crête de la porteuse doit être supérieure au maximum du signal de référence  $\hat{U}_p \geq U_0 \max$ .

### 1.3.3.3 Commande par hystérésis

Cette méthode consiste à commander les interrupteurs tout en respectant une logique de commutation par hystérésis [Leonhard 1985],[Semail 1990]. On peut représenter le fonctionnement par le schéma suivant :

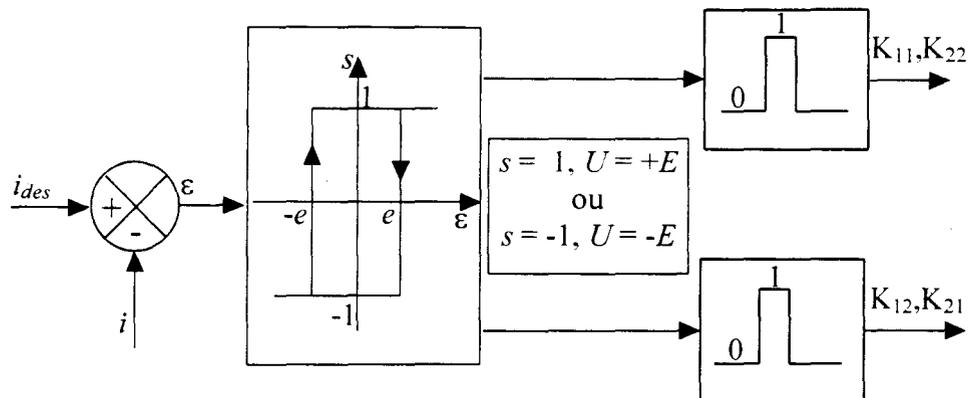


Figure I-18 : Commande par hystérésis

Le principe de commande est le suivant : soit  $\varepsilon$  l'erreur en sortie définie par la différence entre le courant  $i_{des}$  et le courant  $i$ , et  $e$  une tension fixée pour les bornes de l'hystérésis. On obtient alors cet algorithme pour les deux cas :

1. la valeur de l'erreur est négative et croissante :

- si  $\varepsilon < -e$  alors  $s = -1$ , et les interrupteurs  $K_{12}, K_{21}$  ont l'état fermé.
- si  $\varepsilon > -e$  alors  $s = 1$  et les interrupteurs  $K_{11}, K_{22}$  ont l'état fermé.

2. la valeur de l'erreur est positive et décroissante :

- si  $\varepsilon > e$  alors  $s = 1$  et les interrupteurs  $K_{11}, K_{22}$  ont l'état fermé.
- si  $\varepsilon < e$  alors  $s = -1$  et les interrupteurs  $K_{12}, K_{21}$  ont l'état fermé.

Remarque : les interrupteurs sont complémentaires sur la même branche.

On remarque que cette commande entraîne la convergence de l'erreur  $\varepsilon$  en valeur moyenne vers une amplitude inférieure ou égale à  $e$ . En conséquence la valeur  $e$  sera choisie la plus petite possible en fonction des contraintes technologiques du système (fréquence de commutation des composants).

### I.3.4 Binarisation d'une commande continue

La quatrième catégorie est composée de la méthode par binarisation d'une commande continue. Ce principe de commande binarisée a été développé par [Sira-Ramirez 1989] [Abadie 1994] pour certaines classes de systèmes non linéaires en l'état et affines en la commande.

L'élaboration de l'algorithme peut être détaillé comme suit : Dans un premier temps une loi de commande classique (Mode glissant, commande optimal, linéarisation et bouclage etc....) est déterminée en faisant l'hypothèse que l'entrée de commande du système est une variable continue à valeurs dans  $\mathfrak{R}^m$ . La deuxième étape consiste à borner les composantes de l'entrée dans l'intervalle  $[0,1]$  à partir des composantes continues obtenues dans la première étape. Enfin il faut construire les commandes à valeurs dans  $\{0,1\}^m$  à partir des grandeurs d'entrées continues bornées.

Deux méthodes peuvent être utilisées pour borner la commande entre  $[0,1]$  : soit il y a une prise en compte interne dans le calcul de la commande continue, soit elle se fait en externe, c'est-à-dire après avoir élaboré la commande à valeur dans  $\mathfrak{R}^m$ . Par exemple dans le premier cas si on considère que notre modèle est un processus rendu discontinu artificiellement, la commande à mode glissant détermine directement une loi de commande continue à valeurs dans  $[0,1]^m$ . Pour le second cas nous avons par exemple la technique réalisée dans [Abadie 1994] qui propose une linéarisation par retour d'état ( $u \in \mathfrak{R}^m$ ), puis de ramener la commande dans  $[0,1]^m$ . Cette méthode ne sera pas développée dans ce chapitre, mais le lecteur pourra consulter les références données ci-dessus pour plus de détails.

Cette partie se décompose en deux parties : La première étudie une méthode qui permet le passage d'une valeur moyenne d'un signal borné  $[0,1]^m$  vers un signal  $u$  booléen ( $u \in \{0,1\}^m$ ). La seconde partie applique cette méthode à la commande d'un système à entrées booléennes.

#### 1.3.4.1 Valeur moyenne

Posons  $\Delta$  un réel positif fixé et  $\Phi(t) = [\Phi_1(t), \dots, \Phi_m(t)]^T$  une fonction à valeurs dans  $[0,1]^m$ .

Nous allons montrer qu'il est possible de construire une fonction  $u(t)$  à composantes booléennes telle que, pour  $t$  réel,  $u$  et  $\Phi(t)$  aient la même moyenne sur tout l'intervalle  $]t, t + \Delta[$ .

Soient  $t$  et  $\Delta$  réels positifs. La valeur moyenne de  $\Phi(t)$  sur  $]t, t + \Delta[$  est notée  $\hat{\Phi}(t) = [\hat{\Phi}_1(t), \dots, \hat{\Phi}_m(t)]^T$ . Si on construit  $u$  de sorte que :

$$\begin{cases} u_i = 1 & \text{sur } ]t, t + \Delta\hat{\Phi}_i(t)[; \\ u_i = 0 & \text{sur } ]t + \Delta\hat{\Phi}_i(t), t + \Delta[. \end{cases} \quad (\text{I-45})$$

La valeur moyenne de  $u_i$  sur  $]t, t + \Delta[$  est alors égale à :

$$\hat{u}_i(t) = \frac{1}{\Delta} \int_t^{t + \Delta\hat{\Phi}_i(t)} 1 dt + \int_{t + \Delta\hat{\Phi}_i(t)}^{t + \Delta} 0 dt ; \quad (\text{I-46})$$

soit :

$$\hat{u}_i(t) = \hat{\Phi}_i(t). \quad (\text{I-47})$$

pour  $i=1, \dots, m$ .

#### I.3.4.2 Commande Booléenne

En utilisant ce qui précède, nous pouvons construire une commande booléenne à partir de la valeur moyenne d'une commande  $u(t)$  définie dans  $[0, 1]^m$ . Dans la pratique, l'axe des temps est discrétisé avec une période notée  $T_k$  et on note  $\hat{u}(nT_k)$  la valeur moyenne de  $u(t)$  sur l'intervalle  $[nT_k, (n+1)T_k[$  soit :

$$\hat{u}(nT_k) = \frac{1}{T_k} \int_{nT_k}^{(n+1)T_k} u(t) dt \quad (\text{I-48})$$

On construit alors la commande  $u$  en appliquant l'expression (I-45) à  $\Phi(t) = u(t)$  avec  $t=nT_k$  et  $\Delta=T_k$ . On a alors :

$$\begin{cases} u_i = 1 & \text{sur } ]nT_k, nT_k + T_k\hat{u}_i(nT_k)[; \\ u_i = 0 & \text{sur } ]nT_k + T_k\hat{u}_i(nT_k), (n+1)T_k[. \end{cases} \quad (\text{I-49})$$

Prenons pour exemple la commande à mode glissant que nous voulons binariser. Cette commande a l'avantage dans sa conception (commande équivalente  $u_e(t) \in [0, 1]^m$ ) d'être déjà bornée par l'équation (I-34). On obtient sur  $[nT_k, (n+1)T_k[$  :

$$\hat{u}(nT_k) = \hat{u}_e(nT_k) = \frac{1}{T_k} \int_{nT_k}^{(n+1)T_k} u_e(t) dt \quad (\text{I-50})$$

ce qui entraîne alors la commande booléenne définie par l'équation (I-49).

Le principale inconvénient de cette méthode est qu'elle est très contraignante, puisqu'il faut ramener une commande continue entre 0 et 1. De plus avec cette méthodologie il est probable que certaines régions du plan (espace d'état) ne soient pas atteignables par le système.

## I.4 Conclusion

Dans ce premier chapitre nous avons présenté brièvement la classe particulière des systèmes étudiés. Ces systèmes ont été situés, comme étant une sous-classe des systèmes hybrides. La modélisation de ce genre de système n'est pas simple à réaliser. A titre d'exemple nous avons examiné comment la modélisation par bond-graph permet d'obtenir une équation d'état.

La synthèse des principaux résultats sur les commandes des systèmes à entrées binaires fait apparaître quatre classes de méthodes :

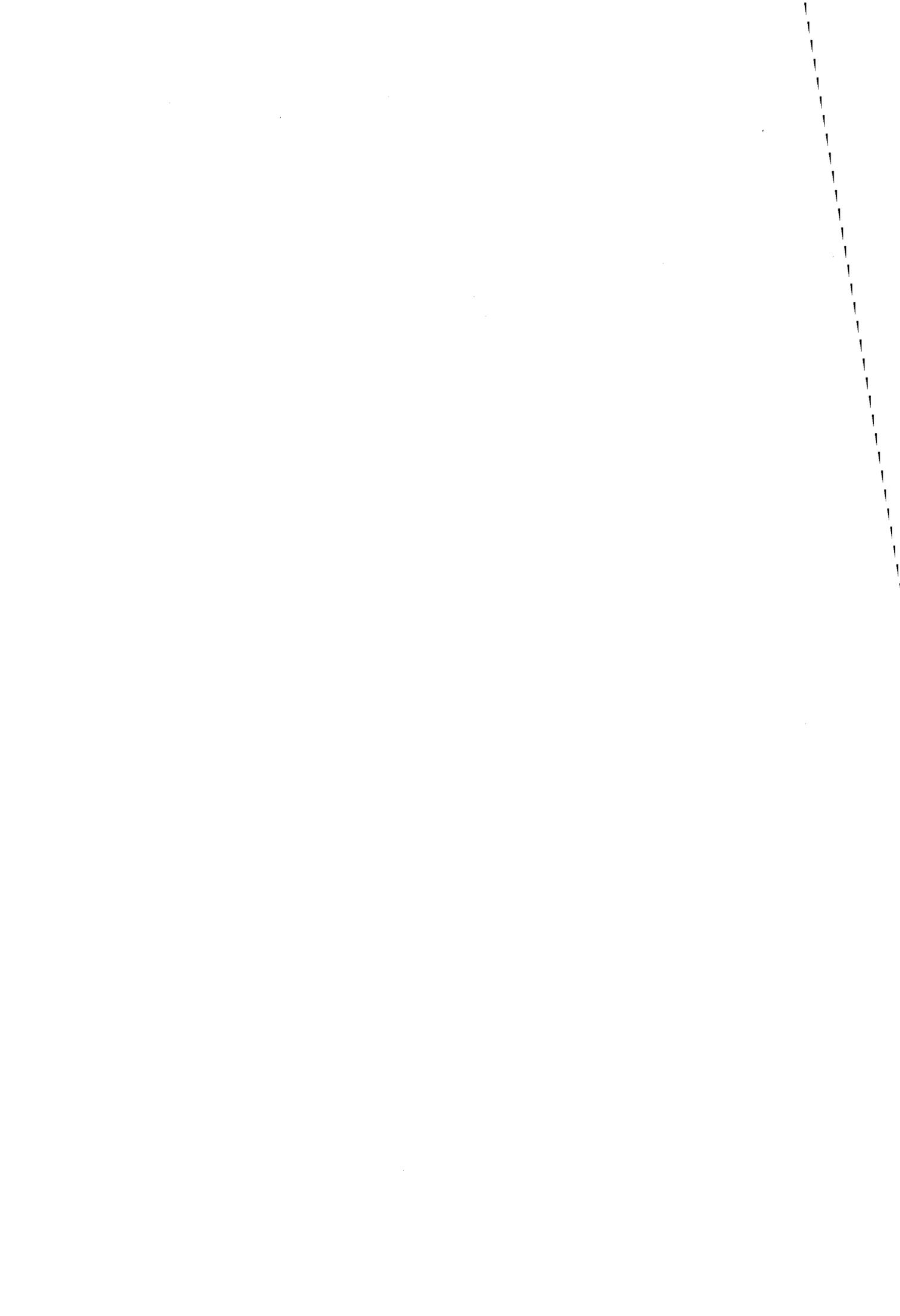
- Nous avons en premier lieu des approches qui ont été menées en considérant l'accès direct aux commutateurs. Dans cette catégorie une méthode originale a été développée par V. Abadie en 1994 au LAIL. Ces principaux résultats sont la définition d'un domaine de convergence à l'intérieur duquel l'état peut converger et la détermination de lois de commande en boucle ouverte et en boucle fermée. Cependant il apparaît de fortes contraintes sur le choix de la période de modulation lorsque les dynamiques du système sont rapides. En effet, la technologie actuelle ne nous permet pas de prendre des périodes trop petites. De plus, la méthode est très sensible aux variations de paramètres.
- Dans la seconde classe d'algorithmes, nous avons les méthodes conduisant à une commande binaire à ramener dans  $\{0,1\}$ . La commande à mode glissant qui a été appliquée à la structure variable du système constitue cette seconde classe. Ces travaux ont l'avantage de s'adapter aux changements de paramètres du procédé (robustesse). Cependant l'inconvénient est de ne pas traiter de manière générale les

---

cas multivariables et de ne pas fournir des conditions suffisantes sur les surfaces de glissement.

- Dans une troisième partie nous trouvons les méthodes par valeur moyenne. La commande par modulation de largeur d'impulsion (MLI ou PWM) fait partie de cette classe d'algorithmes. Celle-ci est souvent utilisée et rencontrée dans la littérature. L'idée est de calculer une commande (avec un PI par exemple) sans modèle du convertisseur puis de la moduler pour obtenir des entrées booléennes sur le système réel. Cette méthode est un cas particulier des systèmes à entrées booléennes utilisés en électronique de puissance. L'intérêt majeur de cette approche se situe dans la simplicité de mise en oeuvre mais elle ne considère pas la commande d'un point de vue global.
- Enfin nous avons la commande par binarisation d'une commande continue. Le principe repose sur le calcul d'une commande par des méthodes classiques en considérant le système à entrées continues. Cette commande continue est dans un premier temps ramenée dans l'intervalle  $[0,1]$ , puis binarisée pour obtenir les valeurs booléennes de la commande  $\{0,1\}$ . Cette approche est très contraignante puisqu'elle nécessite de pouvoir ramener toute commande classique issue du continu dans l'intervalle  $[0,1]$ . De plus, cela implique que certaines régions de l'espace d'état ne soient pas atteignables par le système.

Nous allons donc proposer au cours des deux chapitres suivants de nouvelles lois de commande booléenne assurant performance et simplicité.



## **Chapitre II**



## II. Nouvelle méthode de calcul d'une commande booléenne

### II.1 Introduction

Dans le chapitre précédent nous avons exposé les différentes techniques de commande existant dans la littérature pour les systèmes à entrées booléennes. Nous y avons montré en conclusion les avantages et les inconvénients que peuvent induire ces commandes. Au cours de ce chapitre, nous allons proposer de nouveaux algorithmes de commande pour des systèmes non-linéaires à commutation et tenter de pallier à certains de leurs inconvénients.

Nous présenterons tout d'abord le type de système auquel nous nous sommes intéressés, puis une loi de commande en boucle fermée sera déterminée pour ces systèmes. Cette loi de commande va permettre par la suite, de faire du suivi de trajectoire et de la régulation dans l'espace d'état. Une étude de stabilité de cette loi de commande sera effectuée sur un système linéaire en l'état. La méthode proposée sera illustrée par un exemple de système à entrées booléennes particulier. Enfin, un algorithme de commande pour la minimisation des pertes de puissance par commutation sera proposé. Il permettra également de faire du suivi de trajectoire ainsi qu'il sera montré sur un exemple de circuit électronique.

### II.2 Présentation des systèmes utilisés

Considérons le modèle d'équation d'état :

$$\dot{x} = f(x, u) \tag{II-1}$$

où  $x = (x_1 \dots x_n)^T \in \mathfrak{R}^n$  est le vecteur d'état, et  $u = (u_1 \dots u_m)^T \in \{0,1\}^m$  est le vecteur d'entrée composé de variables booléennes.

Le vecteur d'entrée peut prendre toutes les configurations parmi les  $2^m$  vecteurs différents  $Config(u)$ , contenant des valeurs booléennes tels que :

$$\{Config_i(u) / i = 1 \dots 2^m\} = \left\{ \begin{array}{cccc} \begin{pmatrix} 0 \\ 0 \\ 0 \\ \vdots \\ 0 \end{pmatrix} & \begin{pmatrix} 1 \\ 0 \\ 0 \\ \vdots \\ 0 \end{pmatrix} & \begin{pmatrix} 0 \\ 1 \\ 0 \\ \vdots \\ 0 \end{pmatrix} & \dots & \begin{pmatrix} 0 \\ 1 \\ 0 \\ \vdots \\ 0 \end{pmatrix} & \begin{pmatrix} 1 \\ 1 \\ 1 \\ \vdots \\ 1 \end{pmatrix} & \begin{pmatrix} 1 \\ 0 \\ 1 \\ \vdots \\ 0 \end{pmatrix} & \dots & \begin{pmatrix} 1 \\ 0 \\ 0 \\ \vdots \\ 1 \end{pmatrix} & \begin{pmatrix} 1 \\ 1 \\ 1 \\ \vdots \\ 1 \end{pmatrix} \\ \text{1} & \text{2} & \text{3} & & \text{m+1} & \text{m+2} & \text{m+3} & & \text{i} & \text{2}^m \end{array} \right\} \quad (\text{II-2})$$

Entre deux valeurs de la période d'échantillonnage, l'entrée  $u$  est un vecteur constant dénoté  $Config_i(u)$ . Par exemple, si le nombre d'entrées est  $m = 2$ , alors l'ensemble des  $Config_i(u)$  devient :

$$\{Config_i(u) / i = 1 \dots 4\} = \left\{ \begin{array}{cccc} \begin{pmatrix} 0 \\ 0 \end{pmatrix} & \begin{pmatrix} 1 \\ 0 \end{pmatrix} & \begin{pmatrix} 0 \\ 1 \end{pmatrix} & \begin{pmatrix} 1 \\ 1 \end{pmatrix} \\ \text{1} & \text{2} & \text{3} & \text{4} \end{array} \right\} \quad (\text{II-3})$$

Hypothèse de stabilité : Nous allons supposer dans tout ce qui suit, que le modèle exprimé par l'équation (II-1) est globalement asymptotiquement stable pour les valeurs de  $u$  admissibles.

## II.3 Régulation autour d'une consigne constante

Le but de cette étude est de déterminer la séquence  $Config_i(u)$  pour le vecteur  $u$  afin de réguler le vecteur d'état du système autour d'une position fixée notée  $x_d$  [Holderbaum 98a].

### II.3.1 Principe

Considérons un vecteur  $\varepsilon$  qui représente le vecteur erreur de position associé à la relation :

$$\varepsilon = x_d - x \quad (\text{II-4})$$

où  $x$  est la position courante du modèle dans l'hyperplan associé à l'espace d'état, et  $x_d$  le vecteur objectif. Calculons le vecteur  $\vec{V}_i$  associé à  $(\dot{x})_i$ , tel que  $\vec{V}_i \stackrel{\Delta}{=} (\dot{x})_i$  pour chaque configuration  $Config_i(u)$  associée à la position courante  $x$ .

$$\begin{cases} \vec{V}_i = \Delta(\dot{x})_i = f(x, \text{Config}_i(u)) \\ i = \{1 \dots 2^m\} \end{cases} \quad (\text{II-5})$$

Notre but est de déterminer la configuration  $\text{Config}_i(u)$  minimisant l'angle  $\alpha$  (sans convention de signe pour cet angle) entre  $\varepsilon$  et  $\vec{V}_i$ . Nous pouvons illustrer cette procédure avec l'exemple présenté sur la figure ci-dessous, où  $x \in \mathfrak{R}^2$  et  $u \in \{0,1\}^2$ .

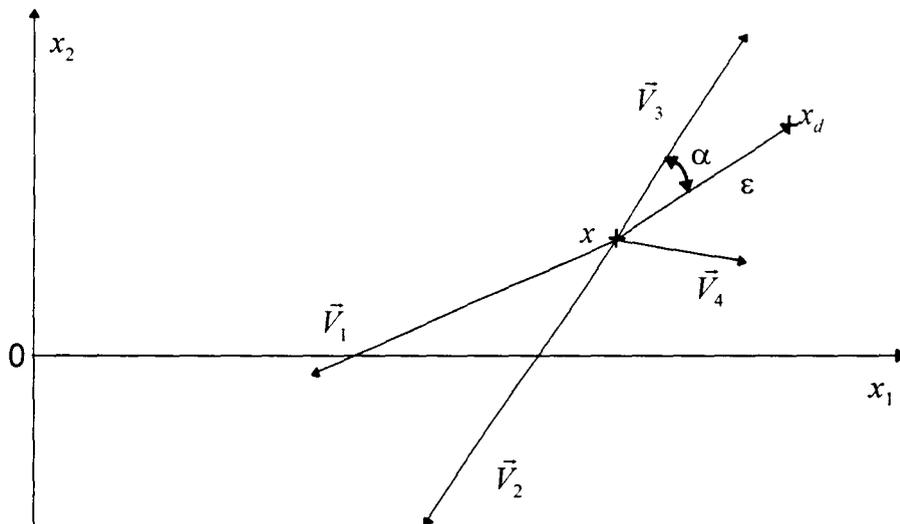


Figure II-1 : Commande dans l'espace d'état.

Dans ce cas, le vecteur  $\vec{V}_3$  forme le plus petit angle (en valeur absolue) avec le vecteur  $\varepsilon$ . La commande  $\text{Config}_3(u) = \begin{bmatrix} 1 \\ 0 \end{bmatrix}$  sera choisie et appliquée au système.

**Interprétation :** Le produit scalaire entre  $\varepsilon$  et  $\vec{V}_i$  ( $\vec{V}_i = \Delta(\dot{x})_i$ ) s'exprime par :

$$\langle \varepsilon, \vec{V}_i \rangle = \|\varepsilon\| \cdot \|\vec{V}_i\| \cdot \cos(\alpha) \quad (\text{II-6})$$

Le cosinus de l'angle  $\alpha$  est donné par :

$$\cos(\alpha) = \frac{\langle \varepsilon, \vec{V}_i \rangle}{\|\varepsilon\| \cdot \|\vec{V}_i\|} \quad (\text{II-7})$$

Nous choisissons la configuration  $Config_i(u)$  minimisant l'angle  $\alpha$ , elle est obtenue en prenant le cosinus maximum :

$$\max_{Config_i(u), i \in \{1, \dots, 2^m\}} (\cos(\alpha)) \quad (\text{II-8})$$

alors le critère est donné par :

$$J = \max_{Config_i(u), i \in \{1, \dots, 2^m\}} \left( \frac{\langle \varepsilon, \vec{V}_i \rangle}{\|\varepsilon\| \|\vec{V}_i\|} \right) \quad (\text{II-9})$$

Ainsi, le critère de commande est exprimé par l'équation (II-9). A présent, nous allons nous intéresser à la stabilité de cette commande.

### II.3.2 Etude de la stabilité

Les résultats portant sur l'étude de la stabilité des systèmes hybrides que l'on trouve dans la littérature sont obtenus pour des représentations multi-modèle [Ye 1998] [Branicky 1998]. Pour le type de modèle manipulé, il s'est avéré **difficile** de montrer la **stabilité de cette commande** dans le cas général. Nous tenterons ici de le faire sur un **modèle linéaire**. Nous supposons que le vecteur d'état  $x$  appartient au domaine de convergence défini chapitre I [Abadie 1994]. Ce domaine est défini par un polyèdre convexe dont les points  $\{x_{1\infty}, \dots, x_{2^m\infty}\}$  forment les sommets (cf chapitre I, équation I-14) .

Par définition ce domaine est tel que, tout point à l'intérieur est atteignable.

Pour des raisons d'analyse et de simplicité dans l'interprétation, nous discrétisons le système linéaire à période d'échantillonnage constante  $T_e$ , sous la forme suivante :

$$x_{k+1} = Mx_k + Nu_k \quad (\text{II-10})$$

La condition d'après la loi de commande discrétisée donne :

$$\max_i \left\{ \frac{\left( (x_{k+1})_i - x_k \right)^T (x_d - x_k)}{\left\| (x_{k+1})_i - x_k \right\| \left\| x_d - x_k \right\|} \right\} > 0 \quad (\text{II-11})$$

expression obtenue pour les configurations  $i$  sélectionnées rendant le critère positif. Cette condition est une condition nécessaire de convergence ainsi qu'il apparaît sur la figure suivante.

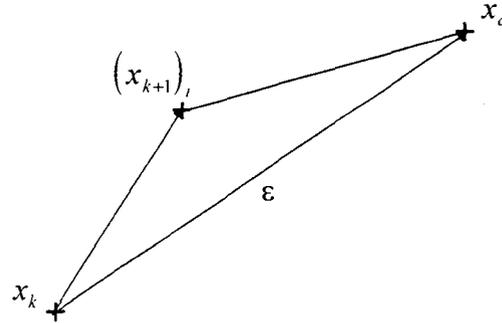


Figure II-2 : Condition de convergence.

Dans ce cas de figure, on a bien  $\|(x_{k+1})_i - x_k\| < \|x_d - x_k\|$ . Toutefois cette condition de positivité du cosinus n'est pas suffisante pour assurer la convergence vers  $x_d$  ainsi qu'il apparaît ci-dessous où l'on a  $\|(x_{k+1})_i - x_k\| > \|x_d - x_k\|$  :

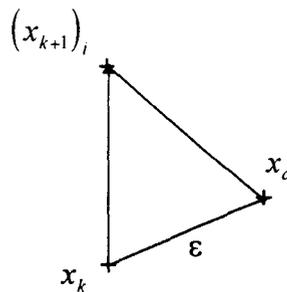


Figure II-3 : Contre exemple

On peut donc au mieux obtenir une oscillation autour de l'objectif. Une majoration  $\delta$  des oscillations pour  $\cos(\alpha) > 0$  peut être donnée.

On suppose que  $\|M\| < 1 \Leftrightarrow 1 - \|M\| > 0$ . Soit :

$$\delta = \frac{\rho}{1 - \|M\|} \tag{II-12}$$

où

$$\rho = \max_{i \in \{1, \dots, 2^n\}} \{ \|Mx_d + NConfig_i(u) - x_d\| \} \tag{II-13}$$

Considérons  $x_k \in B(x_d, \delta)$  où  $B(x_d, \delta)$  est une boule de centre  $x_d$  et de rayon  $\delta$ .

Démontrons l'invariance de la boule, c'est-à-dire :

$$\forall k \geq k_0 \quad \|x_d - x_k\| \leq \delta \Rightarrow \|(x_{k+1})_i - x_d\| \leq \delta \quad (\text{II-14})$$

Nous avons :

$$\|(x_{k+1})_i - x_d\| = \|Mx_k + N\text{Config}_i(u) - x_d\| \quad (\text{II-15})$$

$$\|(x_{k+1})_i - x_d\| = \|M(x_k - x_d) + N\text{Config}_i(u) + (M - I)x_d\| \quad (\text{II-16})$$

$$\|(x_{k+1})_i - x_d\| \leq \|M(x_k - x_d)\| + \|N\text{Config}_i(u) + (M - I)x_d\| \quad (\text{II-17})$$

$$\|(x_{k+1})_i - x_d\| \leq \|M\|\delta + (1 - \|M\|)\delta \quad (\text{II-18})$$

d'où

$$\|(x_{k+1})_i - x_d\| \leq \delta \quad (\text{II-19})$$

En conclusion la boule  $B(x_d, \delta)$  est invariante. La question légitime que l'on peut se poser est : la commande optimisée ramène-t-elle  $x_k$  dans  $B(x_d, \delta)$  pour  $k \geq k_1$  comme le montre la figure suivante :

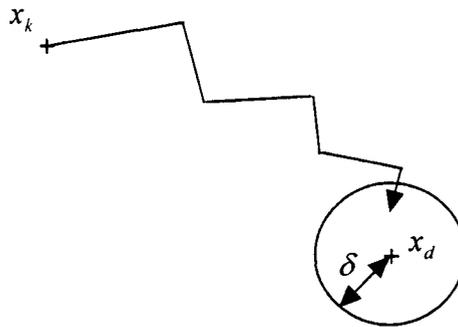


Figure II-4 : Commande optimisée

Pour des conditions initiales à l'extérieur de la boule  $B(x_d, \delta)$ , une condition suffisante de décroissance d'un pas à l'autre de l'écart par rapport à l'objectif final peut-être recherchée en utilisant la fonction candidate à Lyapunov :

$$V(x_d - x_k) = (x_d - x_k)^T (x_d - x_k) = \|x_d - x_k\|^2 \quad (\text{II-20})$$

Il vient :

$$\Delta V = V_{k+1} - V_k = \left(x_d - (x_{k+1})_i\right)^T \left(x_d - (x_{k+1})_i\right) - (x_d - x_k)^T (x_d - x_k) \quad (\text{II-21})$$

soit :

$$\Delta V = \left(x_d - (x_{k+1})_i - x_k + x_k\right)^T \left(x_d - (x_{k+1})_i - x_k + x_k\right) - (x_d - x_k)^T (x_d - x_k) \quad (\text{II-22})$$

$$\Delta V = \left(- (x_{k+1})_i + x_k\right)^T \left(- (x_{k+1})_i + x_k\right) - 2 \left((x_{k+1})_i - x_k\right)^T (x_d - x_k) \quad (\text{II-23})$$

or  $\left((x_{k+1})_i - x_k\right)^T (x_d - x_k) = \cos(\alpha) \left\| (x_{k+1})_i - x_k \right\| \left\| x_d - x_k \right\|$ , par conséquent :

$$\Delta V = \left\| (x_{k+1})_i - x_k \right\|^2 - 2 \cos(\alpha) \left\| (x_{k+1})_i - x_k \right\| \left\| x_d - x_k \right\| \quad (\text{II-24})$$

La condition  $\Delta V < 0$  est vraie si et seulement si :

$$\left\| (x_{k+1})_i - x_k \right\|^2 < 2 \cos(\alpha) \left\| (x_{k+1})_i - x_k \right\| \left\| x_d - x_k \right\| \quad (\text{II-25})$$

$$\left\| (x_{k+1})_i - x_k \right\| < 2 \cos(\alpha) \left\| x_d - x_k \right\| \quad (\text{II-26})$$

Interprétons ce résultat de manière géométrique pour l'ensemble des configurations  $i$ . Rappelons la formule qui lie les cotés d'un triangle par cet exemple :

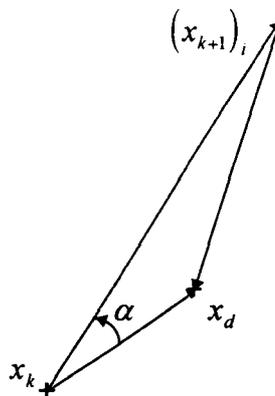


Figure II-5 : Exemple de triangle.

Il vient l'équation suivante :

$$\|(x_{k+1})_i - x_d\|^2 = \|(x_{k+1})_i - x_k\|^2 + \|x_d - x_k\|^2 - 2\|(x_{k+1})_i - x_k\|\|x_d - x_k\|\cos(\alpha) \quad (\text{II-27})$$

On voit que si l'on remplace l'inégalité par l'égalité  $(\|(x_{k+1})_i - x_k\| = 2 \cos(\alpha)\|x_d - x_k\|)$  on se retrouve dans la situation suivante  $\|(x_{k+1})_i - x_d\| = \|x_k - x_d\|$  (Figure II-6) :

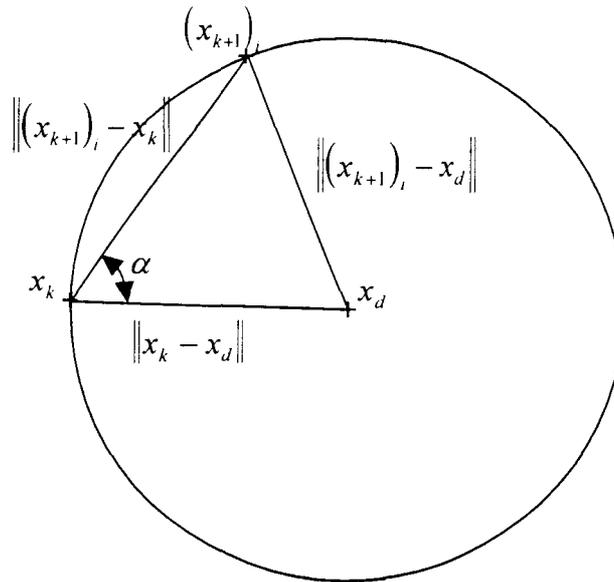


Figure II-6 : Interprétation au sens de l'égalité de la condition.

A présent, revenons sur le sens de l'inégalité dans le schéma ci-dessous. Pour une configuration  $i$  nous avons alors :

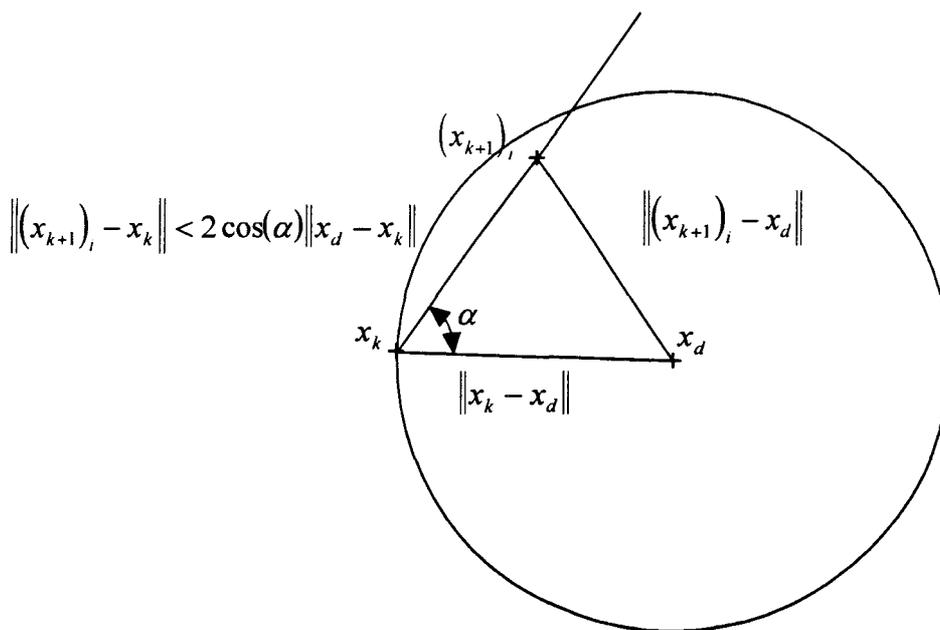


Figure II-7 : Interprétation au sens de l'inégalité pour la convergence.

Dans ce cas on a  $\|(x_{k+1})_i - x_d\| < \|x_k - x_d\|$ , et par conséquent le système converge vers  $B(x_d, \delta)$ . Si le sens de l'inégalité est inversé, le point  $(x_{k+1})_i$  est en dehors du cercle (Figure II-8) de rayon  $\|x_k - x_d\|$ .

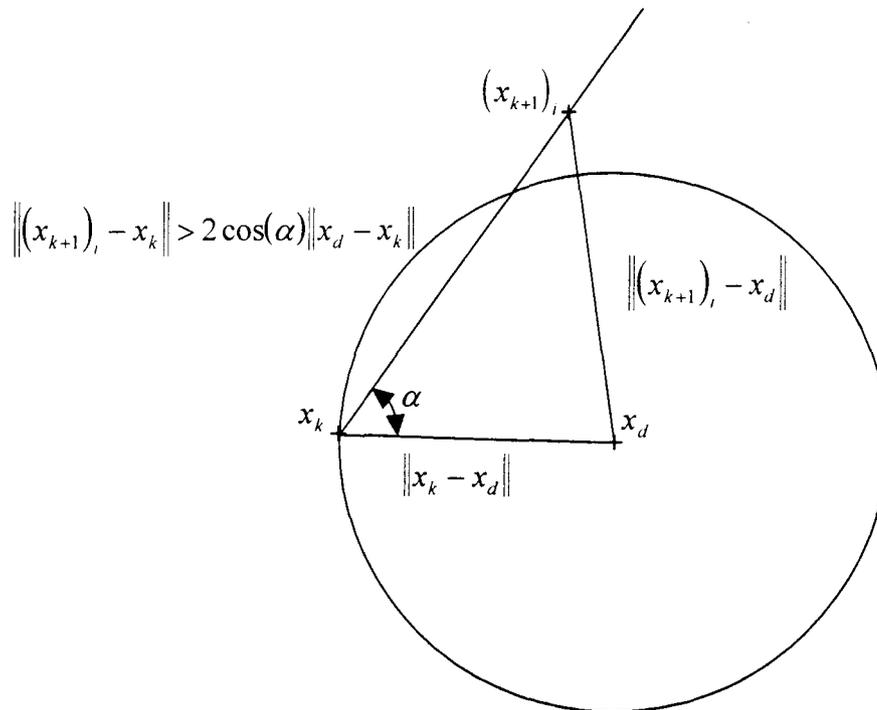


Figure II-8 : Interprétation au sens de l'inégalité pour l'éloignement.

Dans ce cas on obtient  $\|(x_{k+1})_i - x_d\| > \|x_k - x_d\|$ , et donc le système s'éloigne de  $B(x_d, \delta)$ . Du point de vue pratique, pour trouver le point  $(x_{k+1})_i$  qui se rapproche le plus de  $B(x_d, \delta)$ , il est équivalent de maximiser le critère suivant :

$$\cos(\alpha) = \frac{\left( (x_{k+1})_i - x_k \right)^T (x_d - x_k)}{\| (x_{k+1})_i - x_k \| \| x_d - x_k \|} \quad (\text{II-28})$$

Ayant calculé  $\cos(\alpha)$ , la condition suffisante de convergence II-26 n'est pas nécessairement vérifiée. En effet une condition sur la dynamique apparaît, c'est à dire l'appartenance de  $(x_{k+1})_i$  à la boule de rayon  $\|x_k - x_d\|$  dépend à  $\cos(\alpha)$  fixé de la période d'échantillonnage.

Dans une deuxième partie essayons de trouver une commande,  $Config_i(u)$  si elle existe qui respecte la condition  $\Delta V < 0$ . Reprenons l'équation (II-26) et exprimons  $(x_{k+1})_i$  :

$$\|Mx_k + NConfig_i(u) - x_k\| < 2 \cos(\alpha) \|x_d - x_k\| \quad (\text{II-29})$$

$$\|(M - I)x_k + NConfig_i(u)\| < 2 \cos(\alpha) \|x_d - x_k\| \quad (\text{II-30})$$

Majorons le 1<sup>er</sup> terme de l'inéquation (II-30) :

$$\|(M - I)x_k + NConfig_i(u)\| \leq \|NConfig_i(u) + (M - I)x_d\| + \|(M - I)(x_k - x_d)\| \quad (\text{II-31})$$

$$\|(M - I)x_k + NConfig_i(u)\| \leq \rho + \|(M - I)\|(x_k - x_d)\| \quad (\text{II-32})$$

Pour que l'inégalité (II-30) soit vérifiée, il suffit alors que :

$$\rho + \|(M - I)\|(x_k - x_d)\| < 2 \cos(\alpha) \|x_d - x_k\| \quad (\text{II-33})$$

$$\rho < (2 \cos(\alpha) - \|(M - I)\|) \|x_d - x_k\| \quad (\text{II-34})$$

Or le point  $x_k$  est supposé ne pas appartenir à la boule  $B(x_d, \delta)$ , il vient :

$$\|x_d - x_k\| \geq \delta \quad (\text{II-35})$$

d'où

$$\Rightarrow \|x_d - x_k\| \geq \frac{\rho}{1 - \|M\|} \quad (\text{II-36})$$

$$\Rightarrow \frac{1}{\|x_d - x_k\|} \leq \frac{1 - \|M\|}{\rho} \quad (\text{II-37})$$

En remplaçant dans l'équation (II-34), on obtient :

$$2 \cos(\alpha) - \|(M - I)\| > 1 - \|M\| \quad (\text{II-38})$$

$$\cos(\alpha) > \frac{1 - \|M\| + \|(M - I)\|}{2} \quad (\text{II-39})$$

En conséquence si il existe une configuration  $Config_i(u)$  (et donc un  $\alpha_i$ ) qui respecte :

$$\cos(\alpha_i) > \frac{1 - \|M\| + \|(M - I)\|}{2} \text{ on a bien : } \Delta V < 0 \quad (\text{II-40})$$

Or  $M$  dépend de la période d'échantillonnage  $T_e$ . Si  $T_e$  est suffisamment petite, on s'aperçoit en pratique que  $\|M\|$  est proche de 1, i.e  $\cos(\alpha) > 0$  est alors réalisé. Le choix de  $T_e$  telle que  $\|(M - I)\| \geq \|M\| - 1$  assure donc l'existence d'une solution impliquant l'attractivité de la boule  $B(x_d, \delta)$ .

En **conclusion**, nous avons explicité une condition suffisante d'existence d'une boule invariante, néanmoins le rayon de celle-ci est important. Il convient de s'intéresser à l'influence de la période d'échantillonnage sur ce genre de commande. Dans une autre approche nous pourrions rechercher une solution moins conservatrice.

### II.3.3 Cas particulier

Nous devons prendre en compte dans cet algorithme deux cas particuliers qui peuvent se produire. Le premier cas apparaît lorsque les points courants et désirés se confondent ( $x = x_d$ ), ou se trouve dans un voisinage de  $x_d$  tel que  $\|x - x_d\| \leq \gamma$  avec  $\gamma \in \mathfrak{R}$  de valeur très petite. Dans cette situation, la solution est de prendre la commande identique à la précédente dans le but de réduire les commutations.

Illustrons le cas où  $x = x_d$  par la figure ci-dessous :

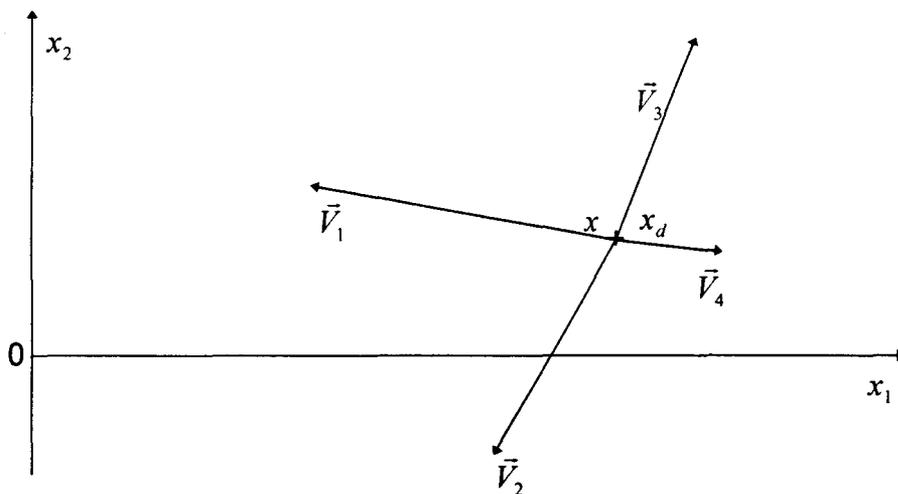


Figure II-9 : Cas où  $x = x_d$ .

Le second cas particulier se produit lorsque les angles sont égaux. Prenons l'exemple de la figure suivante (Figure II-10), où  $x \in \mathbb{R}^2$  et  $u \in \{0,1\}^2$ , pour illustrer cette situation.

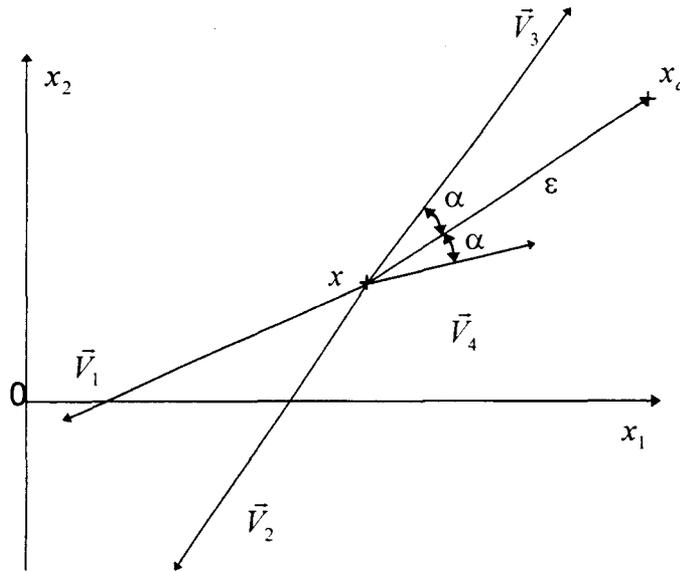


Figure II-10 : Commande avec égalité des angles

Dans cette circonstance, la stratégie adoptée est de prendre la configuration  $i$  qui minimisera les commutations par rapport à la précédente. L'algorithme de minimisation des commutations est fondé sur les distances de Hamming. Ce principe sera développé dans le paragraphe II.5.1.1.

### II.3.4 Exemple

La commande étudiée précédemment est illustrée par l'exemple linéaire factice suivant :

Considérons un système du second ordre composé de deux condensateurs  $C_1$  et  $C_2$ , de résistances et d'un convertisseur de puissance (Figure II-11). Le convertisseur de puissance est alimenté par deux sources de tension continue  $E_1$  et  $E_2$ .  $S_1, S_2, S_3$  et  $S_4$  représentent les interrupteurs dont on peut commander l'état (bloqué ou passant).

Le but dans ce système est de commander les charges électriques  $q_{C1}$  et  $q_{C2}$  des condensateurs.

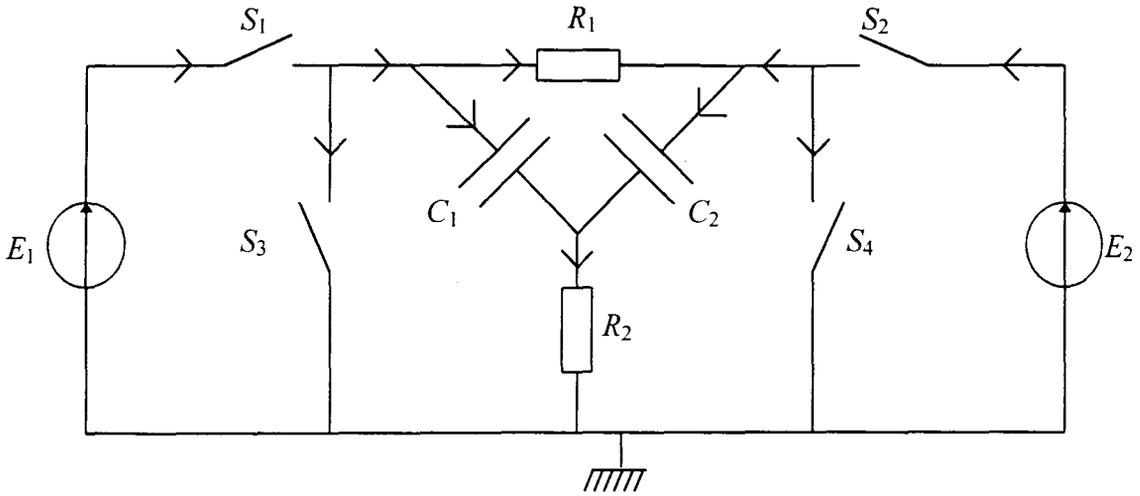


Figure II-11 : Schéma d'électronique de puissance

L'équation d'état est déduite de la modélisation bond-graph du circuit, qui est représentée ci dessous (Figure II-12).

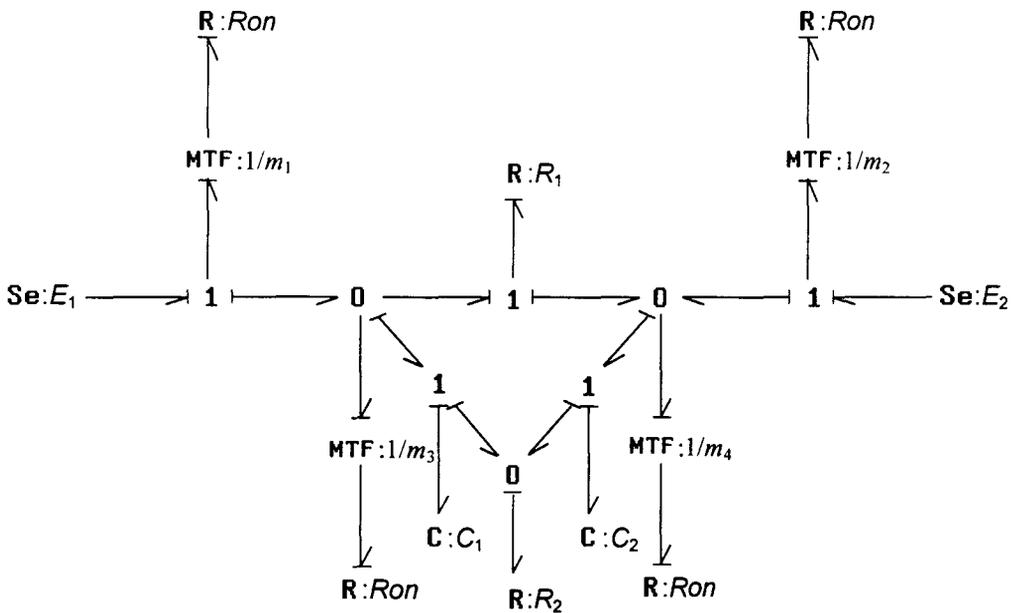


Figure II-12 : Modèle Bond-Graph.

$R_{on}$  est la résistance interne des éléments de commutation quand ils sont en position ON, et les  $m_i$  ( $i = 1,2,3,4$ ) sont les variables booléennes de commande représentant la règle de commutation logique choisie. Les états des interrupteurs  $S_1$  et  $S_3$  ainsi que  $S_2$  et  $S_4$  sont complémentaires, ce qui s'écrit :

$$\begin{cases} m_i + m_{i+2} = 1 \\ i = 1,2 \end{cases} \quad (II-41)$$

Le modèle mathématique du circuit peut-être représenté sous la forme :

$$\{\dot{x} = Ax + Bu \quad (II-42)$$

$$\text{avec } x = \begin{pmatrix} q_{C_1} \\ q_{C_2} \end{pmatrix}.$$

Le vecteur d'entrée  $u$  est composé des variables booléennes telles que  $u = \begin{bmatrix} u_1 \\ u_2 \end{bmatrix} = \begin{bmatrix} m_1 \\ m_2 \end{bmatrix}$ . les matrices  $A$  and  $B$  sont exprimées part :  $A = \frac{1}{\Delta_r} \begin{bmatrix} a_{11} & a_{12} \\ a_{21} & a_{22} \end{bmatrix}$  ;  
 $B = \frac{1}{\Delta_r} \begin{bmatrix} b_{11} & b_{12} \\ b_{21} & b_{22} \end{bmatrix}$  avec :

$$\left\{ \begin{array}{l} \Delta_r = 1 + 2 \frac{R_2}{Ron} \\ a_{11} = -\frac{1}{C_1 Ron} \left( 1 + 2 \frac{R_2}{Ron} \right) - \frac{1}{C_1 R_1} \left( 1 + \frac{R_2}{Ron} \right) \\ a_{12} = -\frac{R_2}{C_2 Ron} \left( \frac{1}{R_1} + \frac{1}{Ron} \right) - \frac{1}{C_2 R_1} \left( 1 + \frac{R_2}{Ron} \right) \\ a_{21} = -\frac{R_2}{C_1 Ron} \left( \frac{1}{R_1} + \frac{1}{Ron} \right) - \frac{1}{C_1 R_1} \left( 1 + \frac{R_2}{Ron} \right) \\ a_{22} = -\frac{1}{C_2 Ron} \left( 1 + 2 \frac{R_2}{Ron} \right) - \frac{1}{C_2 R_1} \left( 1 + \frac{R_2}{Ron} \right) \\ b_{11} = -\frac{E_1}{Ron} \left( 1 + 2 \frac{R_2}{Ron} \right) \\ b_{12} = E_2 \frac{R_2}{Ron^2} \\ b_{21} = -E_1 \frac{R_2}{Ron^2} \\ b_{22} = \frac{E_2}{Ron} \left( 1 + 2 \frac{R_2}{Ron} \right) \end{array} \right. \quad (II-43)$$

Considérons les paramètres électriques définis par :

$$\begin{cases} E_1 = -2 \text{ volts} \\ E_2 = 5 \text{ volts} \\ R_{on} = 0.1 \Omega \\ R_1 = 2.7 \Omega \\ R_2 = 1 \Omega \\ C_1 = 100 \text{ mF} \\ C_2 = 47 \text{ mF} \end{cases} \quad (\text{II-44})$$

alors l'équation d'état devient :

$$\dot{x} = \begin{bmatrix} -101.94 & -109.2 \\ -51.32 & -216.9 \end{bmatrix} \cdot x + \begin{bmatrix} 420 & 500 \\ 200 & 1050 \end{bmatrix} \cdot u \quad (\text{II-45})$$

Avec ce modèle on peut calculer les sommets du domaine de convergence en utilisant les résultats précédents

$$\{Config_i(u) / i = 1 \dots 4\} = \left\{ \begin{pmatrix} 0 \\ 0 \end{pmatrix}_1 \begin{pmatrix} 1 \\ 0 \end{pmatrix}_2 \begin{pmatrix} 0 \\ 1 \end{pmatrix}_3 \begin{pmatrix} 1 \\ 1 \end{pmatrix}_4 \right\} \quad (\text{II-46})$$

$$x_{1\infty} = \begin{pmatrix} 0 \\ 0 \end{pmatrix} \quad x_{2\infty} = \begin{pmatrix} 4.2 \\ -0.07 \end{pmatrix} \quad x_{3\infty} = \begin{pmatrix} -0.37 \\ 4.93 \end{pmatrix} \quad x_{4\infty} = \begin{pmatrix} 3.81 \\ 4.85 \end{pmatrix} \quad (\text{II-47})$$

On souhaite que le système évolue vers le point  $x_d = [2; 2]^T$ . Pour mettre en oeuvre l'algorithme de commande, on choisit une période d'échantillonnage d'après les valeurs propres de  $A$ .

Les valeurs propres  $\lambda_i(A)$ ,  $i = 1, 2$  de  $A$  sont : -65 et -250. La période est choisie par rapport à la plus grande valeur propre en respectant  $T_e \ll \frac{1}{|\lambda(A)|_{\max}} = 0.004$ . Nous prenons par exemple comme valeur de  $T_e$  égale à 0.0001 sec.

En utilisant le logiciel de simulation MATLAB-SIMULINK, on obtient les réponses temporelles des variables d'état et de la commande suivantes :

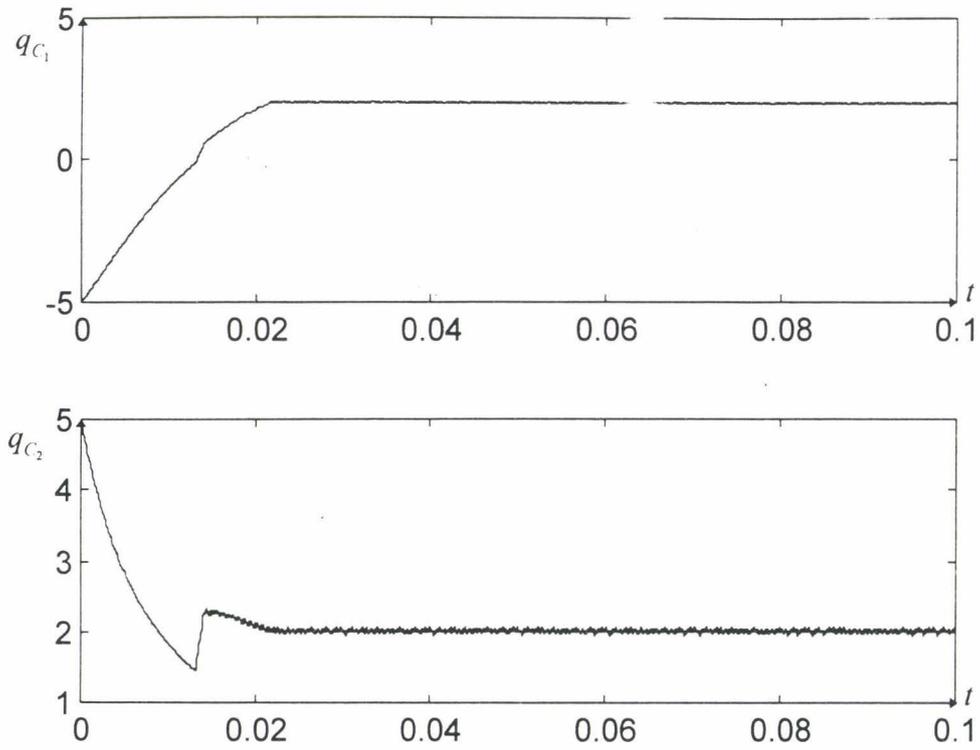


Figure II-13 : Réponse temporelle des variables d'état.

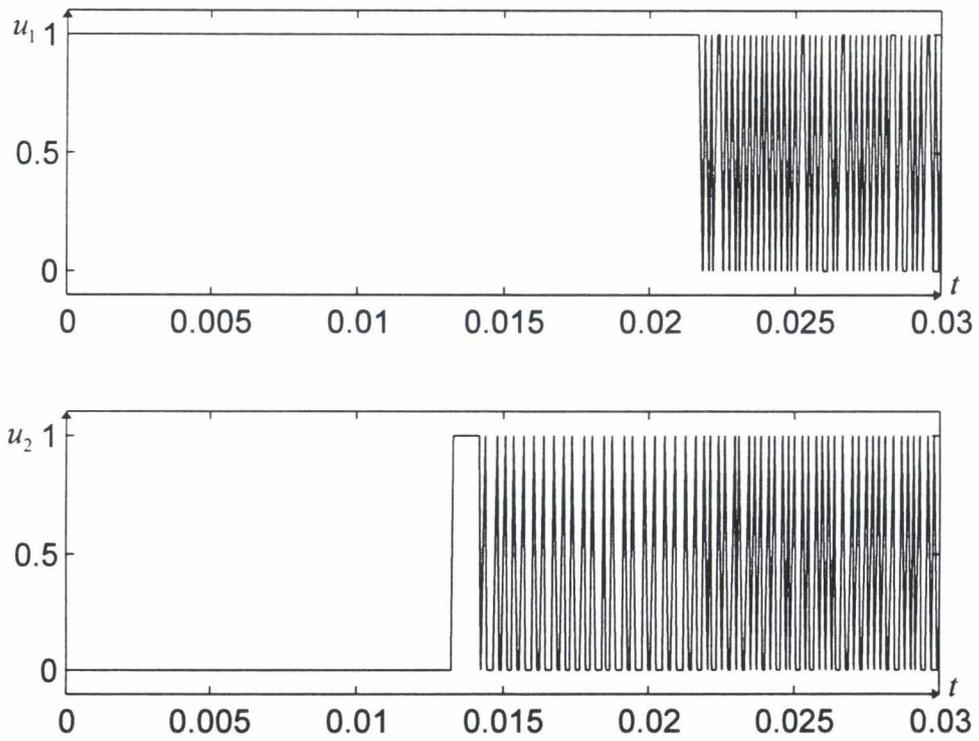


Figure II-14 : Entrées de commande.

La représentation dans l'espace d'état est donnée par la figure suivante :

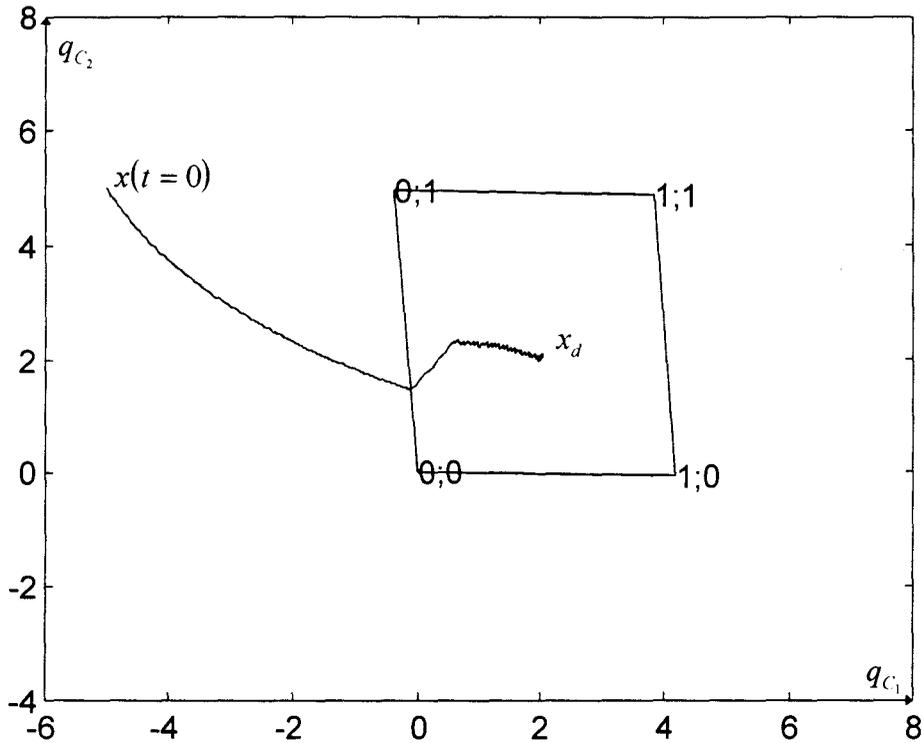


Figure II-15 : Trajectoire dans l'espace d'état.

Cet exemple montre la validité de l'algorithme de commande puisque l'état du système converge bien vers la valeur désirée  $x_d$ .

A présent nous allons révéler sur cet exemple l'étude de la stabilité et la détermination de l'attracteur à consigne constante. Tout d'abord discrétisons notre modèle :

$$x_{k+1} = \begin{bmatrix} 0.9899 & -0.0107 \\ -0.0051 & 0.9786 \end{bmatrix} x_k + \begin{bmatrix} 0.0417 & 0.0492 \\ 0.0197 & 0.1037 \end{bmatrix} u \tag{II-48}$$

avec  $M = \begin{bmatrix} 0.9899 & -0.0107 \\ -0.0051 & 0.9786 \end{bmatrix}$  et  $N = \begin{bmatrix} 0.0417 & 0.0492 \\ 0.0197 & 0.1037 \end{bmatrix}$

La majoration  $\delta$  de l'écart final est donnée par la valeur suivante :

$$\delta = \frac{\rho}{1 - \|M\|} = 14.2 \tag{II-49}$$

En dehors de la boule  $B(x_d, \delta)$  nous pouvons expliciter la condition II-40 :

$$\cos(\alpha_i) > \frac{1 - \|M\| + \|(M - I)\|}{2} = 0.0159 \quad (\text{II-50})$$

Cette condition a pour conséquence que la convergence de l'état du système vers le point désiré existe si il y a une configuration  $Config_i(u)$  (et donc un  $\alpha_i$ ) qui respecte un angle inférieur à  $89^\circ$ .

## II.4 Poursuite de trajectoire

Notre objectif ici, est de permettre au système de réaliser la poursuite d'une trajectoire non-prédéfinie [Holderbaum 98b] dans l'espace d'état. La méthode pour réaliser cette poursuite de trajectoire repose sur la loi de commande définie précédemment dans la section II-3. Les détails de cette procédure sont décrits ci-dessous.

### II.4.1 Algorithme

La stratégie de commande en boucle fermée consiste à définir la configuration de commande  $u$  à chaque période d'échantillonnage  $T_e$ . On détermine alors cette prochaine configuration  $Config_i(u)$  du vecteur d'entrée  $u$  par la méthodologie présentée précédemment dans le paragraphe II-3.

Cette procédure est illustrée par l'exemple de la Figure II-16. Comme on peut le voir dans cet exemple le vecteur  $\vec{V}_1(t)$  minimise l'angle entre  $\left( (x_{k+1})_i - x_k \right)$  et  $(x_d - x_k)$  à l'instant  $t$  donc  $Config_1(u)$  sera appliquée, et le vecteur  $\vec{V}_2(t + T_e)$  associé à  $Config_2(u)$  sera sélectionné à l'instant  $t + T_e$ .

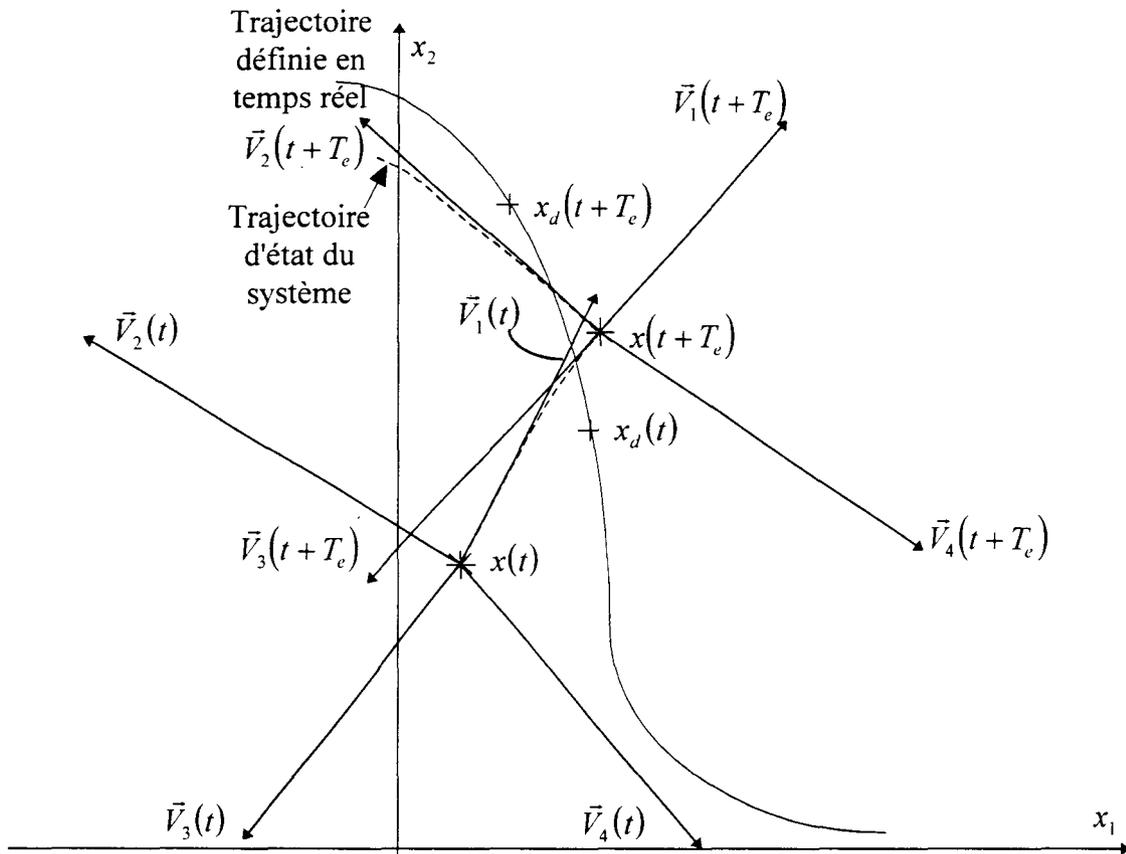


Figure II-16 : Poursuite de trajectoire dans l'espace d'état.

### II.4.2 Exemple

Cette méthode est mise en oeuvre sur un circuit d'électronique de puissance (Figure II-17). Ce circuit est un système du second ordre composé d'une bobine d'inductance  $L_1$  et d'un condensateur de capacité  $C_1$ , de résistances ainsi que d'interrupteurs. Le montage électronique est alimenté par deux tensions continues  $E_1$  et  $E_2$ . Le but est de commander le flux magnétique  $\phi_{L_1}$  dans la bobine, et la charge  $q_{C_1}$  dans le condensateur.

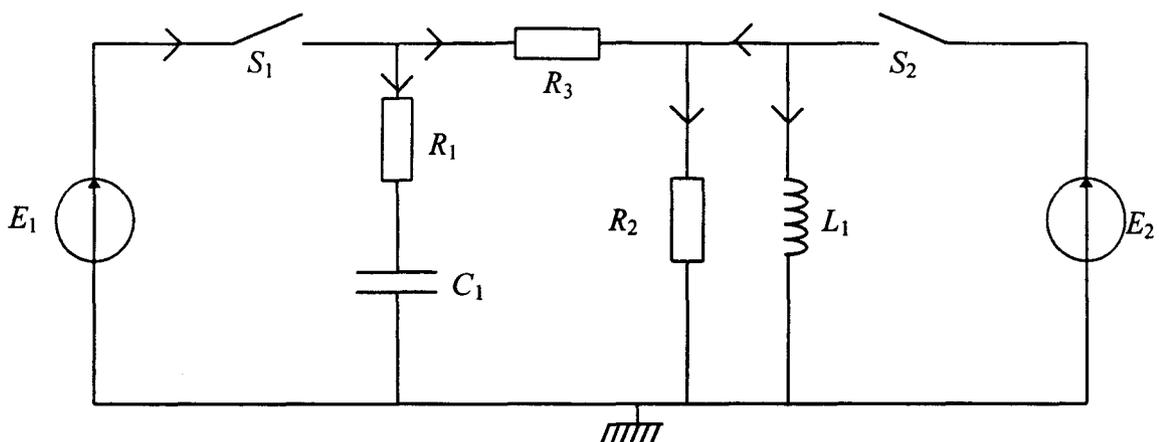


Figure II-17 : Schéma d'électronique de puissance.

### Modélisation

L'approche classique pour étudier un circuit électrique avec des interrupteurs est fondée sur la détermination de toutes les topologies possibles du circuit. En fait, il y a  $2^m$  topologies pour le circuit électronique, si  $m$  est le nombre d'interrupteurs. Cette approche devient très compliquée quand le nombre d'éléments de commutation augmente. La méthode proposée pour obtenir le modèle est d'utiliser la méthodologie bond-graph exposée chapitre I [Borne 92][Ducreux 92]. Cette approche nous permet de déterminer une équation d'état unique du circuit électronique valable quel que soit l'état des interrupteurs (passant ou bloqué).

Le modèle bond-graph associé au modèle physique Figure II-17 est donné Figure II-18 :

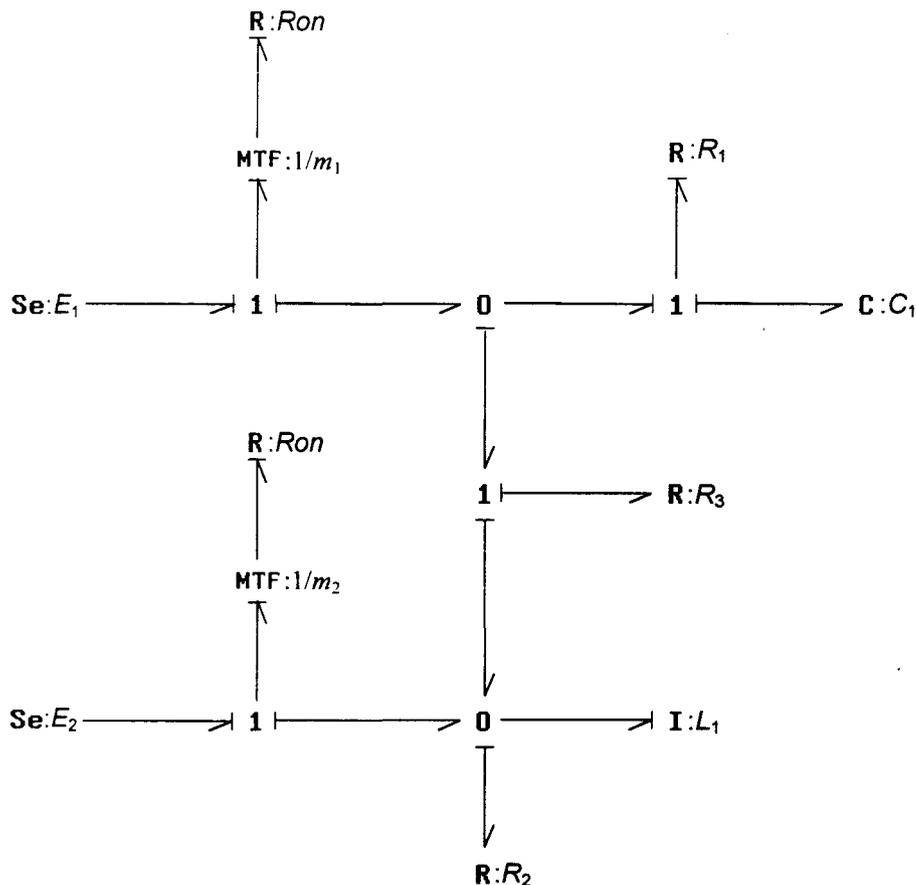


Figure II-18 : Modèle Bond-Graph.

$R_{on}$  est la résistance interne des interrupteurs quand ils sont passants, et les  $m_i$  ( $i = 1,2$ ) sont les variables booléennes représentant la logique de commutation, c'est-à-dire les variables d'entrées booléennes de commande, de l'état des deux commutateurs. Nous

représentons alors l'état des commutateurs par  $m_i = 0$  pour faire passer le  $i^{\text{ème}}$  transistor de l'état ON à l'état OFF, et  $m_i = 1$  pour le faire passer de l'état OFF à l'état ON.

Le modèle mathématique du circuit peut être représenté sous la forme :

$$\dot{x} = G(u)x + F(u)u \quad (\text{II-51})$$

avec  $x = \begin{pmatrix} \phi_{L_1} \\ q_{C_1} \end{pmatrix}$ .

Le vecteur d'entrée  $u$  est composé des variables booléennes telles que

$$u = \begin{bmatrix} u_1 \\ u_2 \end{bmatrix} = \begin{bmatrix} m_1 \\ m_2 \end{bmatrix}. \text{ les matrices } G \text{ et } F \text{ s'écrivent : } G = \frac{1}{\Delta_r} \begin{bmatrix} g_{11} & g_{12} \\ g_{21} & g_{22} \end{bmatrix}; F = \frac{1}{\Delta_r} \begin{bmatrix} f_{11} & f_{12} \\ f_{21} & f_{22} \end{bmatrix}$$

avec :

$$\left\{ \begin{array}{l} \Delta_r = 1 + \frac{R_3}{Ron} u_1 + \frac{R_3}{R_1} + \frac{R_2}{Ron} u_2 + \frac{R_2}{Ron} u_1 + \frac{R_2}{R_1} \\ g_{11} = -\frac{R_2}{L_1} \left( 1 + \frac{R_3}{Ron} u_1 + \frac{R_3}{R_1} \right) \\ g_{12} = \frac{R_2}{C_1 R_1} \\ g_{21} = -\frac{R_2}{L_1 R_1} \\ g_{22} = -\frac{1}{C_1 R_1} \left( 1 + \frac{R_3}{Ron} u_1 + \frac{R_2}{Ron} u_2 + \frac{R_2}{Ron} u_1 \right) \\ f_{11} = E_1 \frac{R_2}{Ron} \\ f_{12} = E_2 \frac{R_2}{Ron} \left( 1 + \frac{R_3}{Ron} u_1 + \frac{R_3}{R_1} \right) \\ f_{21} = E_1 \frac{R_3}{Ron R_1} \left( 1 + \frac{R_2}{Ron} u_2 \right) + E_1 \frac{R_2}{Ron R_1} \\ f_{22} = E_2 \frac{R_2}{Ron R_1} \end{array} \right. \quad (\text{II-52})$$

Considérons les paramètres électriques définis par :

$$\begin{cases} E_1 = 12 \text{ volts} \\ E_2 = 4 \text{ volts} \\ R_{on} = 0.1 \Omega \\ R_1 = 12 \Omega \\ R_2 = 2 \Omega \\ R_3 = 40 \Omega \\ L_1 = 100 \text{mH} \\ C_1 = 4.7 \text{mF} \end{cases} \quad (\text{II-53})$$

Alors l'équation d'état devient :

$$\dot{x} = \frac{1}{\Delta_r} \begin{bmatrix} -86.6 - 8000u_1 & 35.46 \\ -1.66 & -17.73(1 + 420u_1 + 20u_2) \end{bmatrix} \cdot x + \frac{1}{\Delta_r} \begin{bmatrix} 240 & 346.4 + 32000u_1 \\ 420 + 8000u_2 & 6.66 \end{bmatrix} \cdot u \quad (\text{II-54})$$

$$\text{avec } \frac{1}{\Delta_r} = \frac{1}{4.5 + 420u_1 + 20u_2}$$

Le logiciel de simulation MATLAB-SIMULINK a été utilisé pour l'étude de la réponse du système électrique. La période d'échantillonnage utilisée pour la simulation est  $T_e = 0.001$  sec. Le modèle en boucle fermée est représenté Figure II-19 :

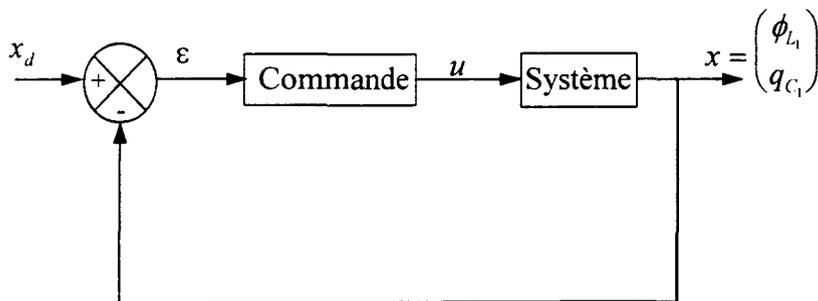


Figure II-19 : Schéma de commande du système électrique.

### II.4.3 Résultat

La loi de commande proposée est illustrée sur le circuit électronique décrit précédemment (Figure II-17).

Le vecteur désiré  $x_d$  a été simulé par deux formes d'ondes sinusoïdales telles que :

$$x_d = \begin{pmatrix} x_{d1} = 3 + \frac{1}{2} \sin(6.t) \\ x_{d2} = 0.4 + \frac{1}{5} \sin(3.t) \end{pmatrix} \quad (\text{II-55})$$

Ces équations simulent le vecteur  $x_d$  en temps réel. Nous pouvons noter que la fréquence des signaux désirés doit respecter la bande passante du système.

La Figure II-20 représente la trajectoire d'état dans l'espace d'état :

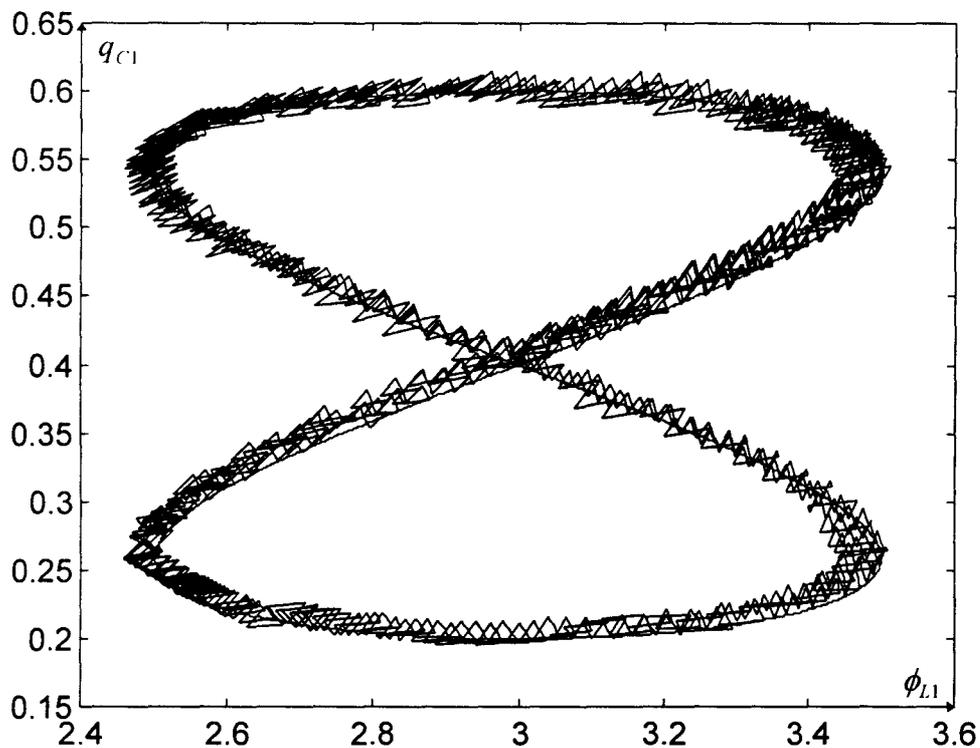


Figure II-20 : Trajectoire dans l'espace d'état.

Les réponses temporelles de la charge électrique  $q_C$  dans le condensateur et du flux magnétique  $\phi_{L_1}$  dans la bobine sont représentées Figure II-21 et Figure II-22 avec leurs trajectoires désirées.

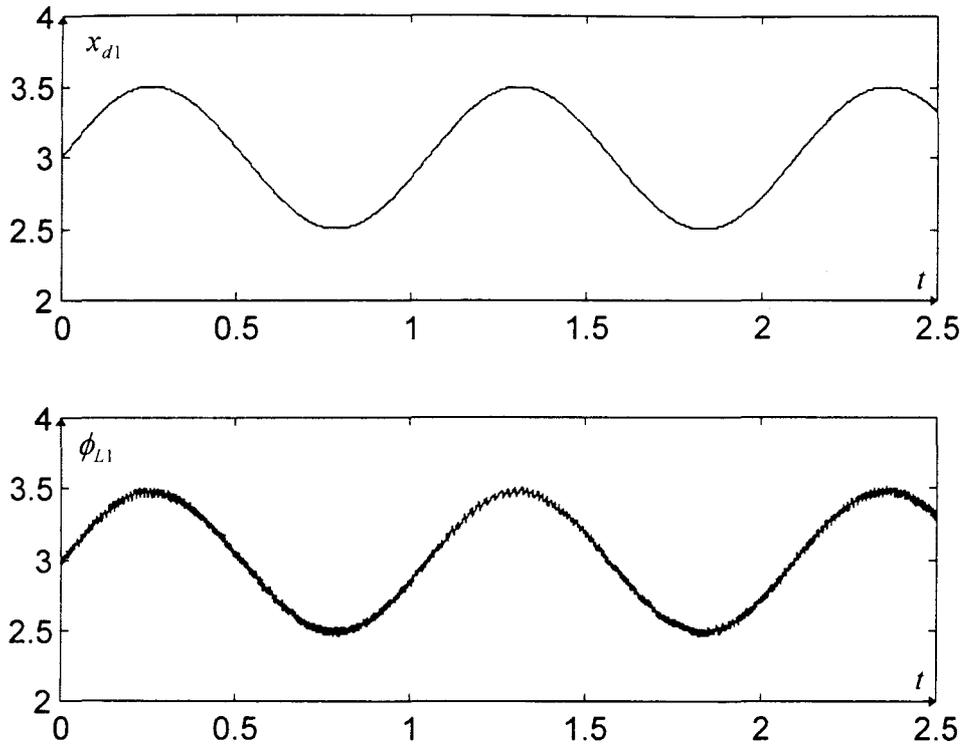


Figure II-21 : Réponses temporelles du flux magnétique  $\phi_{L1}$  et de la trajectoire désirée  $x_{d1}$ .

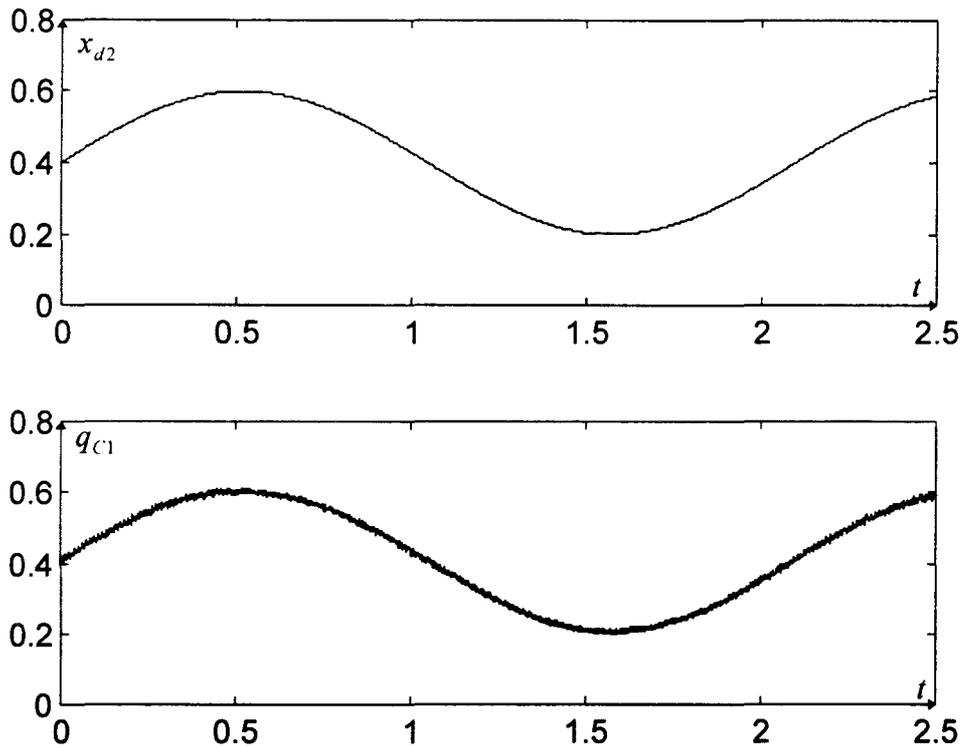


Figure II-22 : Réponses temporelles de la charge  $q_{C1}$  et de la trajectoire désirée  $x_{d2}$ .

La figure suivante représente les commandes booléennes :

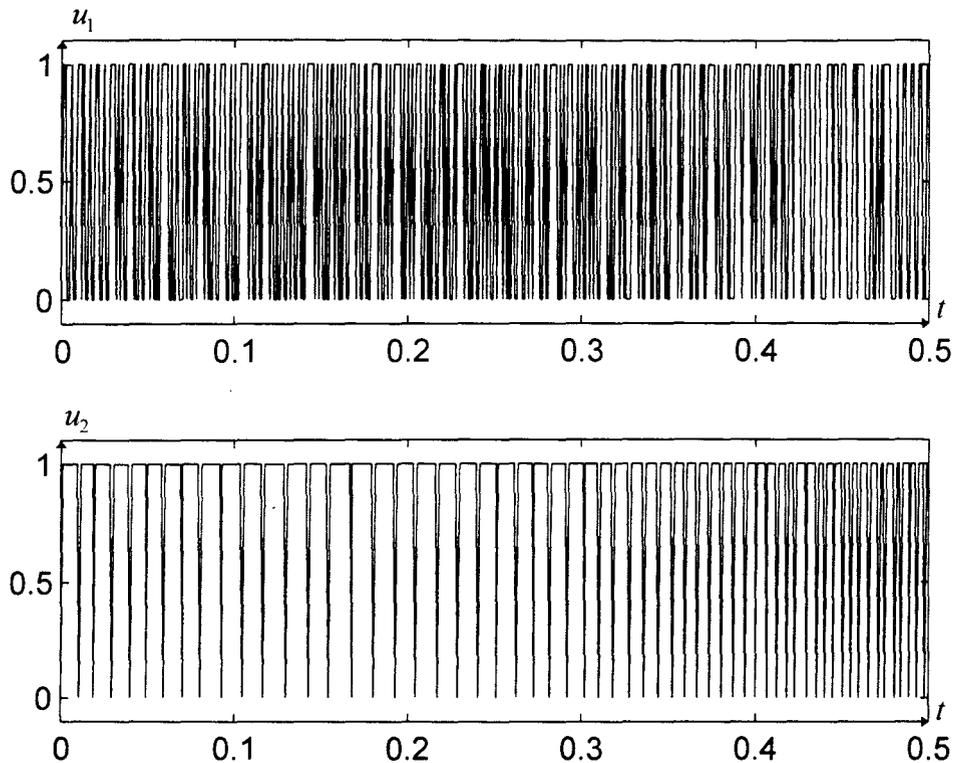


Figure II-23 : Entrées booléennes.

Les résultats de simulation valident la stratégie de commande booléenne obtenue pour la poursuite de trajectoire définie dans l'espace d'état.

## II.5 Minimisation de l'énergie dissipée dans les commutations

Notre objectif ici est d'élaborer une loi de commande qui permette de suivre une trajectoire en temps réel tout en minimisant les pertes par commutations. Cette réduction des commutations permet en fait de diminuer les pertes de puissance dues aux interrupteurs. Ces pertes se produisent quand les composants (par exemple, les transistors) sont dans les phases de transition ou de commutation. Elles sont dues au temps de transition pour passer de l'état passant (ON) à l'état bloqué OFF ou l'inverse. Par exemple, pour un transistor de technologie MOS-FET [Garcia 1997] associé à une charge résistive, l'état ON du composant est représenté sur la Figure II-24 (a) : nous pouvons y voir la transition de la valeur 0 à la valeur  $I_{ON}$  pour le courant, et la transition de la valeur  $V_{ON}$  à la valeur  $V_{OFF}$  pour la tension.

La perte de puissance dans le composant  $P_e[W]$  est également caractérisée par la Figure II-24(b).  $P_e[W]$  est défini comme le produit entre la tension et le courant:  $P_e(t) = V_T(t)I_T(t)$ . L'énergie totale dissipée dans le composant est déterminée par :

$$E = \int_{t=0}^{\infty} P_e(t)dt \quad (\text{II-56})$$

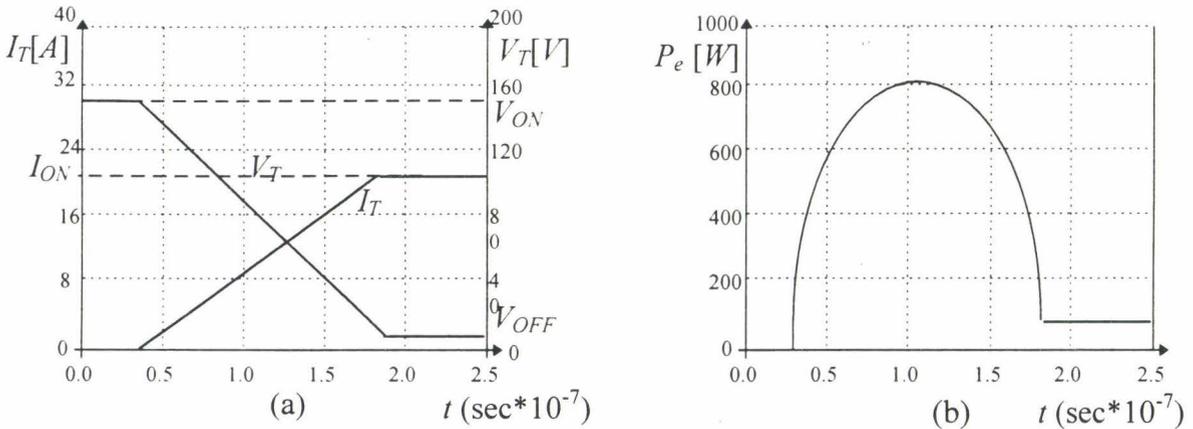


Figure II-24 : (a) Simulation à l'allumage du courant et de la tension dans le composant  
(b) dissipation de puissance électrique dans le composant

L'énergie totale consommée par le système durant la commutation est la somme de toutes les énergies dissipées dans les différents composants.

## II.5.1 Commande booléenne et réduction des commutations

Pour résoudre ce problème de réduction des commutations, nous proposons d'utiliser la distance de Hamming associée à la définition d'un espace de dimension  $n$  tout autour du point désirée  $x_d(t)$ .

### II.5.1.1 Méthode de la distance de Hamming

Considérons deux vecteurs binaires  $X$  et  $Z$  représentés Figure II-25. On appelle "distance de Hamming  $D(X,Z)$ " le nombre de bits pour lesquels les vecteurs  $X$  et  $Z$  sont différents.

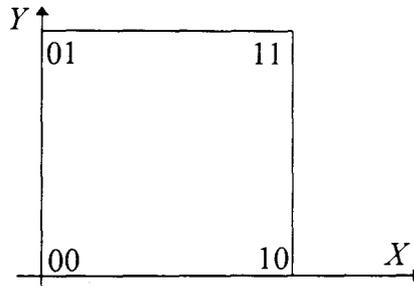


Figure II-25 : Représentation géométrique des combinaisons booléennes pour  $m=2$

C'est également le nombre de 1 dans le vecteur  $X \oplus Z$ , où  $\oplus$  représente la fonction logique XOR.

Exemple : Considérons les deux vecteurs binaires  $\begin{cases} X = 01 \\ Z = 11 \end{cases}$  alors  $X \oplus Z = 10$  et donc

$$D(X, Z) = 1 + 0 = 1.$$

Dans cet exemple il n'y a qu'un digit binaire qui a changé. Pour choisir la configuration  $Config_i(u)$  à appliquer au temps  $t$ , nous calculons les distances de Hamming entre  $[Config_j(u)](t - T_e)$  et tous les cas possibles de  $Config_i(u)$ ,  $i = 1..2^m$ . Les  $p$  configurations menant vers une distance  $D([Config_j(u)](t - T_e), [Config_i(u)](t)) = 0$  ou 1 sont conservées. Alors la configuration  $Config_i(u)$  est déterminée, parmi les  $p$  configurations sélectionnées, en minimisant le critère  $J$  défini paragraphe II-3.

Par exemple, si nous affectons au temps  $t - T_e$  la commande  $Config_2(u) = \begin{bmatrix} 1 \\ 0 \end{bmatrix}$ , alors

nous avons le choix pour la commande au temps  $t$  entre  $Config_2(u) = \begin{bmatrix} 1 \\ 0 \end{bmatrix}$  ( $D(.)=0$ ),

$$Config_1(u) = \begin{bmatrix} 0 \\ 0 \end{bmatrix} \text{ (D(.)=1) et } Config_4(u) = \begin{bmatrix} 1 \\ 1 \end{bmatrix} \text{ (D(.)=1).}$$

Il est à noter que dans ce cas il n'y a pas de contradiction avec la condition de stabilité qui a été exposée précédemment. En effet, le principe de base restant le même c'est-à-dire de choisir le vecteur qui se dirige dans la position du point désiré (angle minimum) parmi les configurations restantes qui minimise l'énergie.

### II.5.1.2 Méthode de l'espace de dimension $n$

Cette minimisation des commutations est également réalisée, en définissant un espace de dimension  $n$  autour du point courant désiré  $x_d(t)$ . La stratégie de commande en boucle fermée consiste à déterminer la configuration de commande  $u$  à chaque instant d'intersection de la trajectoire du système avec le domaine défini autour du point  $x_d(t)$ . La contrainte de minimisation des commutations est satisfaite puisque la même commande est conservée dans le domaine. Les détails de cette procédure sont décrits ci-dessous.

#### **Procédure**

L'instant de commutation est déterminé quand la position courante  $x(t)$  sort du domaine défini autour du point désiré. On détermine alors la prochaine configuration  $Config_i(u)$  du vecteur d'entrée  $u$  par la méthodologie présentée précédemment dans le paragraphe II.5.1.1. L'algorithme est expliqué ci-après.

Dans un premier temps, un domaine  $D_s$  est défini autour du point désiré courant  $x_d(t)$ . La taille de ce domaine est choisie arbitrairement par l'utilisateur, de façon à ce que le compromis entre l'efficacité sur la réduction du nombre des commutations et la précision sur le suivi de trajectoire soit respecté.

Nous examinons si le point  $x_d(t)$  est l'intérieur ou à l'extérieur de  $D_s$  :

- Si  $x \notin D_s$ , alors la détermination de la séquence  $u$  est réalisée en utilisant la méthode présentée dans la section précédente II.5.1.1.
- Si  $x \in D_s$ , alors la commande calculée précédemment, lorsque  $x$  était en dehors du domaine  $D_s$ , est appliquée.

## II.5.2 Application

La méthode présentée a été mise en oeuvre sur deux exemples de circuits d'électronique de puissance. Nous prendrons, pour ces deux exemples, comme domaine  $D_s$  une forme rectangulaire avec une valeur de 5% par rapport à la valeur  $x_d(t)$  pour définir ses bords. Il est à noter que cette définition des bords et la commande sont recalculés à chaque période d'échantillonnage.

Les détails et les résultats obtenus pour ces deux exemples sont présentés dans les paragraphes qui suivent.

### II.5.2.1 Exemple 1

Le premier exemple est composé du circuit présenté précédemment (Figure II-11). Le but dans ce système est de commander la charge électrique  $q_{C1}$  et  $q_{C2}$  dans les condensateurs. L'étude de la réponse du système électrique a été effectuée à partir du schéma du système bouclé (Figure II-19). La période d'échantillonnage utilisée est de  $T_e = 0.0001$  sec. Le vecteur désiré  $(x)_d$  a été simulé par deux formes d'ondes sinusoïdales telles que :

$$x_d = \begin{pmatrix} x_{d1} = 2 + \sin(30t) \\ x_{d2} = 2 + \sin(60t) \end{pmatrix} \quad (\text{II-57})$$

Nous illustrons sur cet exemple l'étape transitoire, c'est-à-dire si on part de conditions initiales éloignées de la trajectoire. Nous prendrons comme conditions initiales les valeurs suivantes pour le système :  $q_{C1} = -5C$  et  $q_{C2} = 5C$

Nous obtenons alors les résultats suivants pour l'étape transitoire :

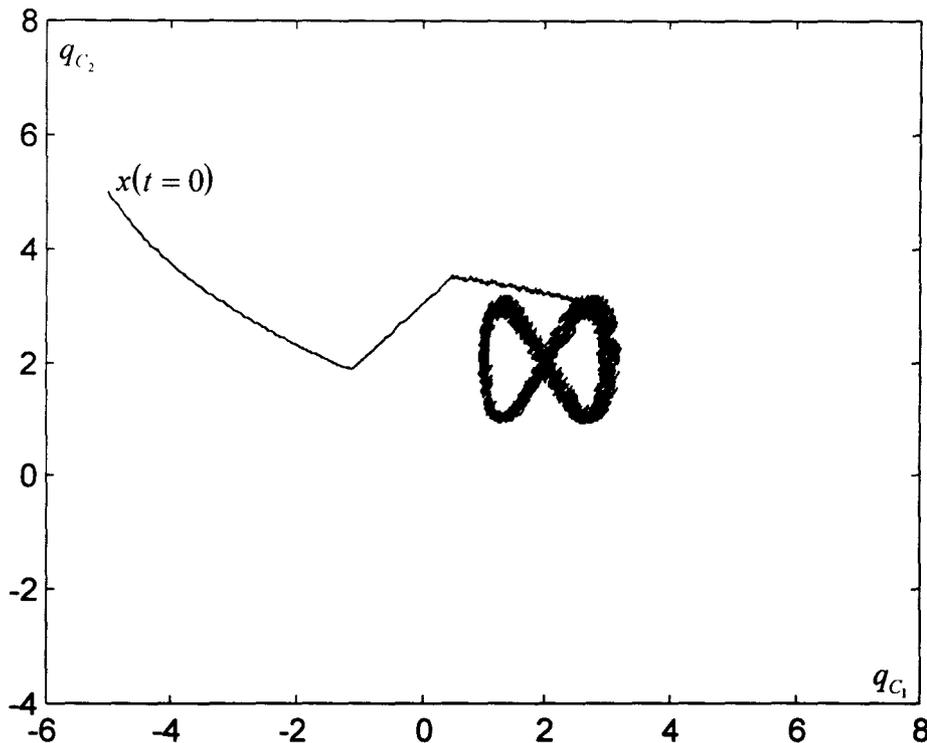


Figure II-26 : Etape transitoire de la trajectoire dans l'espace d'état.

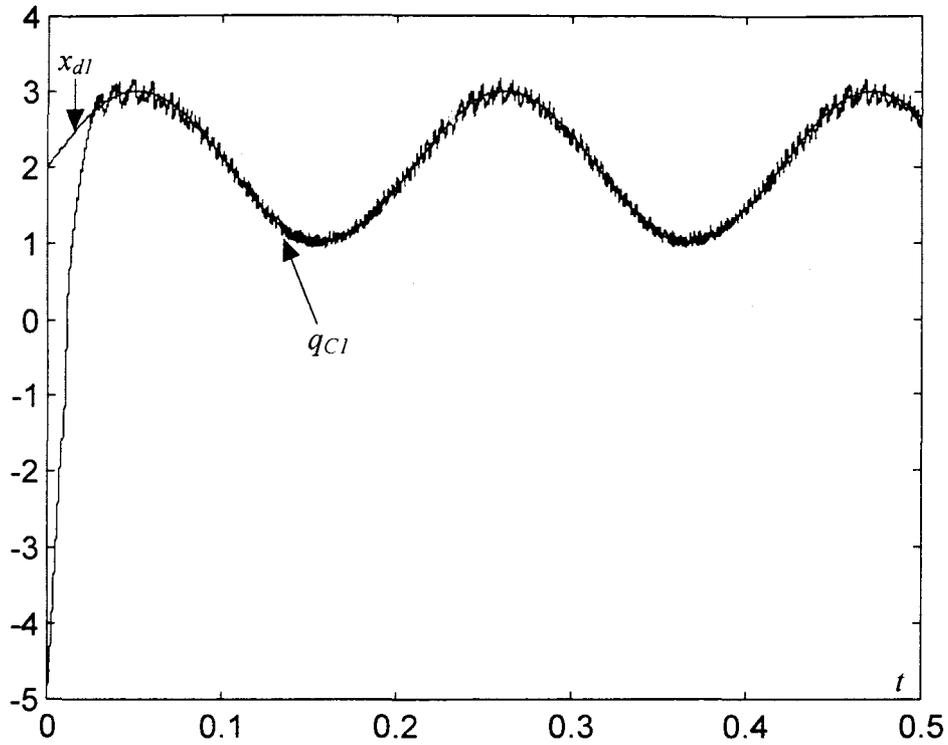


Figure II-27 : Etape transitoire d'évolution de la charge électrique  $q_{C1}$  comparée à sa trajectoire désirée.

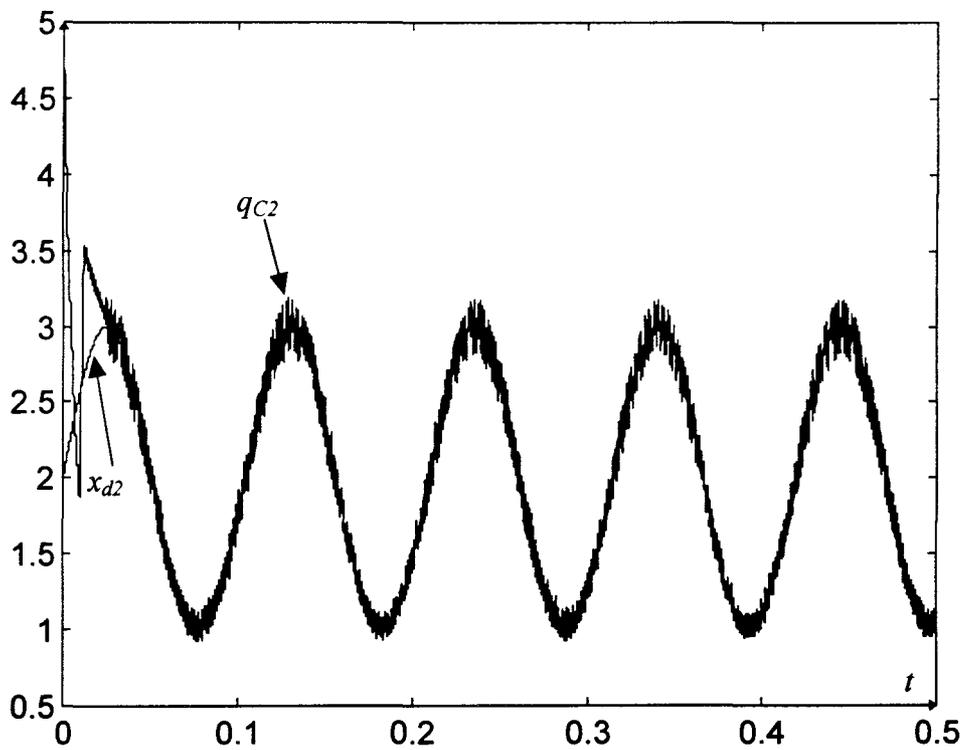


Figure II-28 : Etape transitoire d'évolution de la charge électrique  $q_{C2}$  comparée à sa trajectoire désirée.

A présent examinons les courbes obtenues après l'étape transitoire. La figure suivante représente la trajectoire d'évolution du système dans l'espace d'état (entre 0.7 et 1.5 sec) :

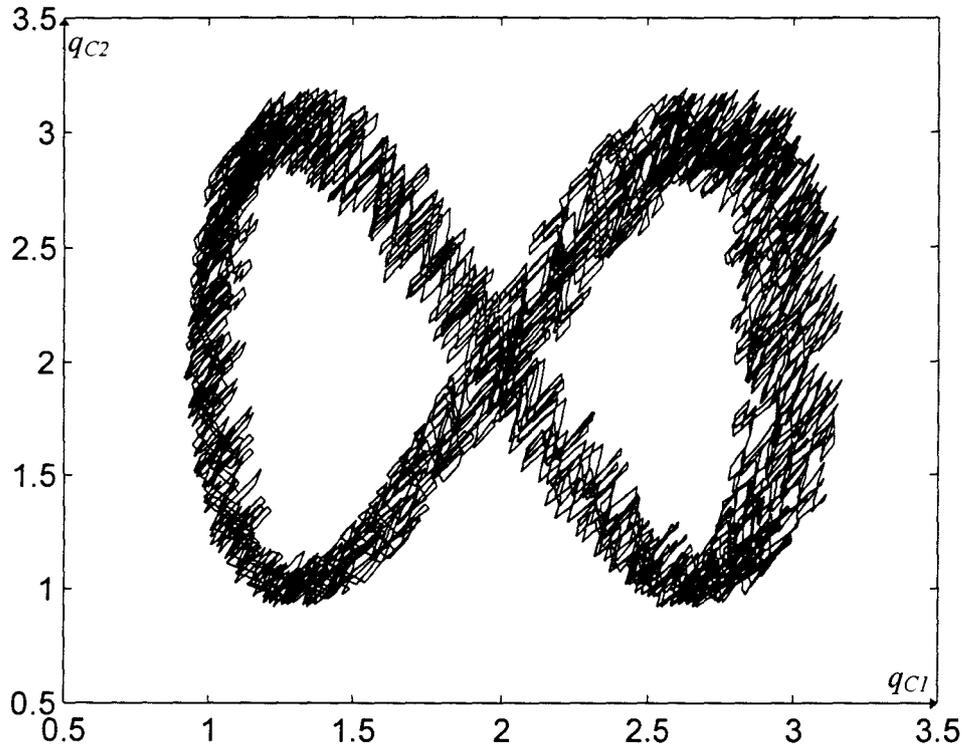


Figure II-29 : Trajectoire dans l'espace d'état.

Les charges électriques  $q_{C1}$  et  $q_{C2}$  dans les condensateurs C1 et C2 sont représentées sur les Figure II-30 et Figure II-31 avec leurs trajectoires désirées respectives.

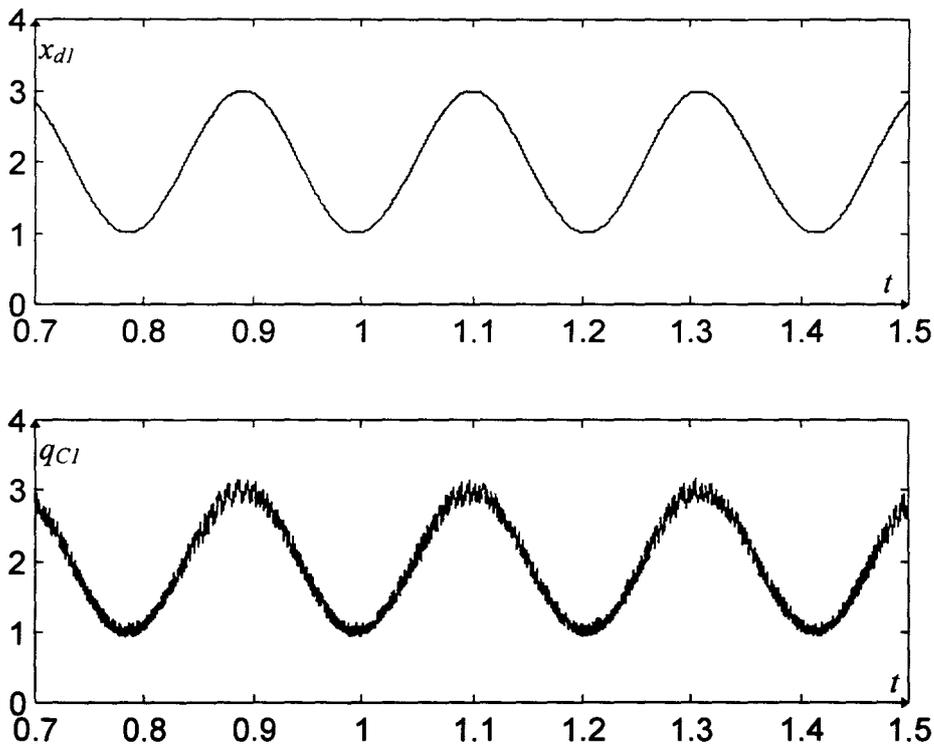


Figure II-30 : La charge électrique  $q_{C1}$  comparée à sa trajectoire désirée.

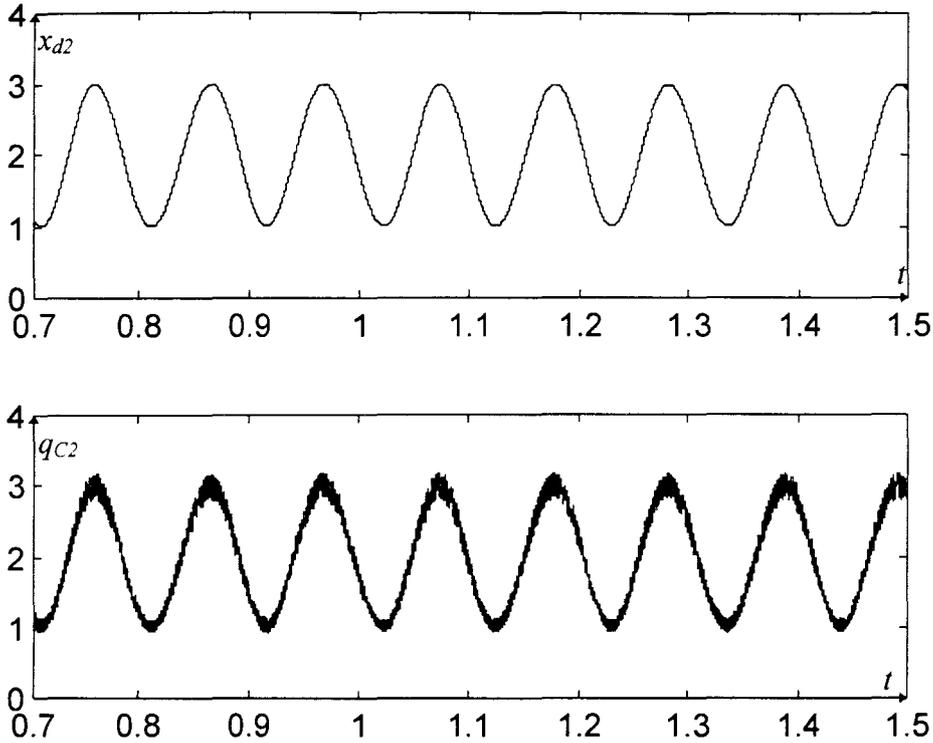


Figure II-31 : La charge électrique  $q_{c2}$  comparée à sa trajectoire désirée.

Dans cette simulation, le nombre de commutations a été compté durant 1 sec (0.5 à 1.5 sec). Il est à noter que pendant ce comptage nous n'avons pas pris en compte la période transitoire. Ce nombre a été comparé avec le nombre de commutations obtenu à partir de la commande sans la minimisation des commutations. De plus, nous avons comparé l'erreur moyenne absolue et l'erreur maximum sur le même temps de simulation. Les résultats sont alors les suivants :

- 2516 commutations pour la commande avec la méthode de minimisation. Les erreurs moyennes absolues pour  $q_{c1}$  et  $q_{c2}$  sont respectivement de 0.0579C et de 0.0638C, et les erreurs maximales sont  $q_{c1}= 0.2113C$  et  $q_{c2}= 0.2183C$ .
- 10731 commutations pour la commande sans la méthode de minimisation. Les erreurs moyennes absolues pour  $q_{c1}$  et  $q_{c2}$  sont respectivement de 0.0184C et de 0.0313C, et les erreurs maximales sont  $q_{c1}= 0.0652C$  et  $q_{c2}= 0.0971C$ .

### II.5.2.2 Exemple 2

Le second exemple est composé du circuit présenté Figure II-17. La simulation de ce système est réalisée avec les mêmes spécifications (forme de  $x_d$ , valeur des composants, période d'échantillonnage  $T_e$ ). Les résultats de simulation obtenus pour la commande sont

alors présentés ci-dessous. La trajectoire d'état dans l'espace d'état est représentée Figure II-32.

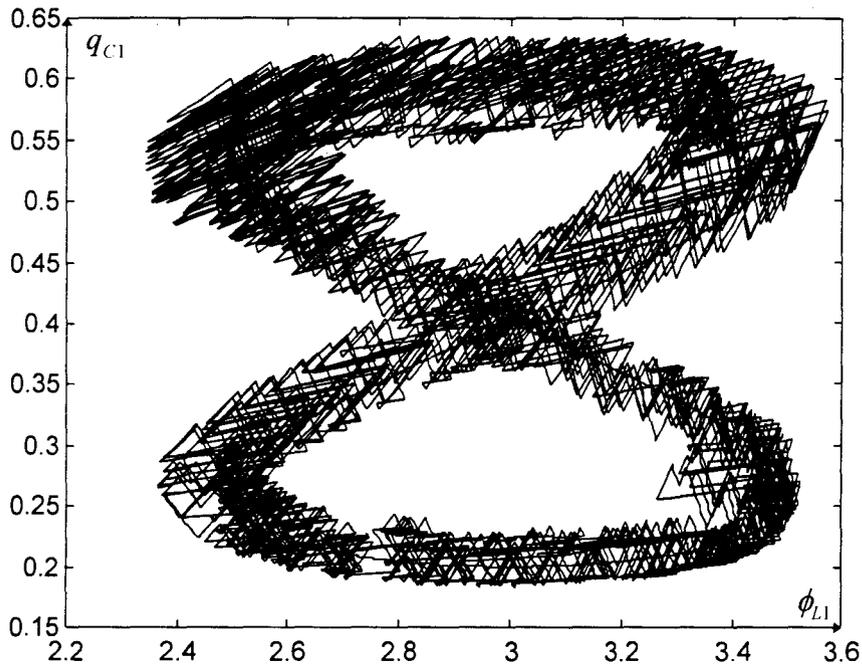


Figure II-32 : Trajectoire dans l'espace d'état.

Les réponses temporelles de la charge électrique  $q_{C1}$  dans le condensateur et du flux magnétique  $\phi_{L1}$  dans la bobine sont représentées respectivement Figure II-33 et Figure II-34.

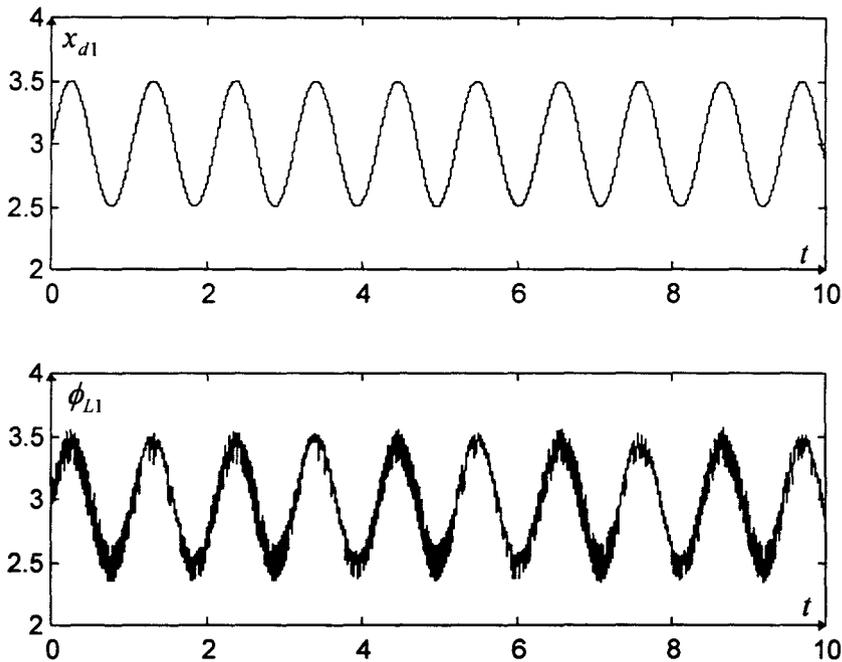


Figure II-33 : Réponses temporelles du flux magnétique  $\phi_{L1}$  et de la trajectoire désirée  $x_{d1}$

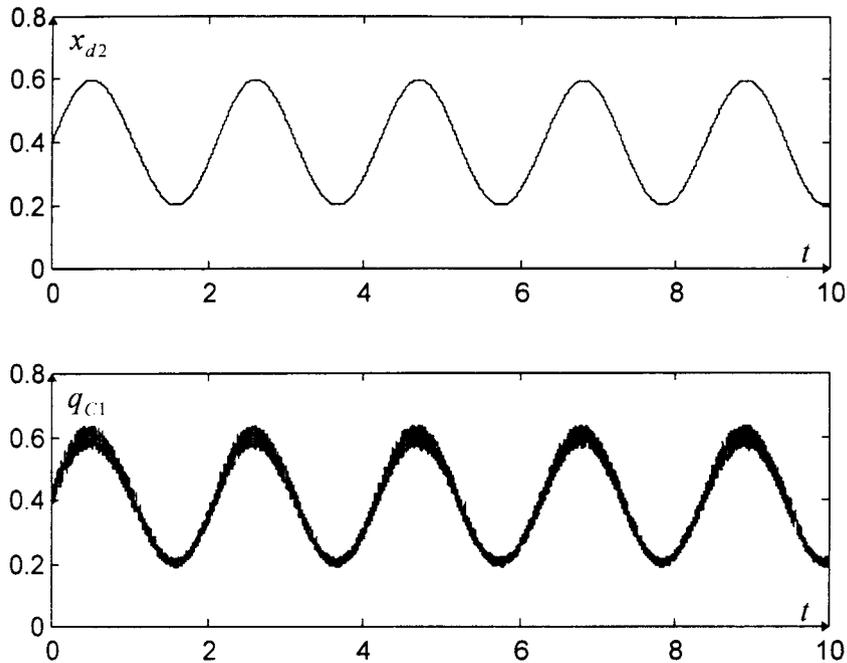


Figure II-34 : Réponses temporelles de la charge  $q_{C1}$  et de la trajectoire désirée  $x_{d2}$ .

Le nombre de commutations a été compté durant 10 secondes pendant la simulation. Ce nombre a été comparé avec le nombre de commutations obtenu à partir de la commande sans la minimisation des commutations. Dans notre cas les méthodes de minimisation comprennent la technique de la distance de Hamming et la définition d'un domaine  $D_s$ . Nous avons également comparé l'erreur moyenne absolue et l'erreur maximum sur le même temps de simulation. Les résultats sont alors les suivants :

- 2597 commutations pour la commande avec la méthode de minimisation. Les erreurs moyennes absolues pour  $\phi_{L1}$  et  $q_{C1}$  sont respectivement de  $0.0579Wb$  et de  $0.0153C$ , et les erreurs maximales sont  $\phi_{L1} = 0.2326Wb$  et  $q_{C1} = 0.0574C$ .
- 7995 commutations pour la commande sans la méthode de minimisation. Les erreurs moyennes absolues pour  $\phi_{L1}$  et  $q_{C1}$  sont respectivement de  $0.0209Wb$  et de  $0.007C$ , et les erreurs maximales sont  $\phi_{L1} = 0.0662Wb$  et  $q_{C1} = 0.0220C$ .

### II.5.2.3 Conclusion

Comme nous l'avons remarqué, la méthode de minimisation diminue le nombre de commutations, et donc les pertes de puissance par commutation. Toutefois, l'erreur moyenne relative et l'erreur maximum sont un peu plus importantes avec la commande minimisant l'énergie de commutation. Ceci résulte du fait qu'il faut choisir un bon

---

compromis entre la précision et la minimisation des pertes de puissance. Ce réglage s'effectuera par l'intermédiaire de la définition de la taille du domaine  $D_s$ .

## II.6 Conclusion

Au cours de ce chapitre, différents algorithmes ont été proposés. Dans une première partie, une loi de commande a été proposée pour la régulation autour d'une consigne constante. Nous avons effectué alors une étude de la stabilité dans le cas d'un système linéaire. Dans un second temps, la loi de commande a été étendue pour le suivi de trajectoires définies en temps réel. Enfin une méthode de minimisation de l'énergie dissipée dans les commutations a été présentée. L'objectif est de réduire le nombre des commutations dans le but d'avoir de faibles pertes de puissance dans les composants de commutation. Pour cela nous avons utilisé la notion de distance de Hamming et défini un domaine de tolérance. La sélection de la commande se fera alors selon la méthode exposée précédemment. Pour illustrer ces algorithmes nous les avons appliqués à des circuits électroniques. Les courbes de simulation montrent la validité de la méthode.

Quelques remarques sur la taille des domaines définis peuvent être présentées dans les cas de minimisation des commutations : Si ces domaines sont très grands alors nous réduisons le nombre de commutations, mais nous avons une précision faible. A l'inverse si ces domaines sont très petits, le nombre de commutations augmentera mais par contre nous aurons une grande précision. Les domaines seront choisis de manière à avoir un "bon" compromis entre la réduction du nombre de commutation et la précision.

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 vis à vis des paramètres.

Dans le chapitre suivant, nous allons développer une autre stratégie de commande booléenne en utilisant les réseaux de neurones et le principe de loi de commande présenté dans le paragraphe II-3 de ce chapitre.



## **Chapitre III**



# III. Approche Neuromimétique de la Commande Booléenne

## III.1 Introduction

Les premiers travaux sur les réseaux de neurones datent de 1943. A cette époque, McCulloch et Pitts [McCulloch 1943] inventèrent le premier neurone formel. Ils supposèrent que l'impulsion nerveuse était le résultat d'un calcul simple effectué par chaque neurone. La pensée naissait alors grâce à un effet collectif de tous les neurones interconnectés. Le modèle associé au premier neurone formel consistait en un produit scalaire entre un vecteur d'entrée et un vecteur poids, suivi d'un élément à seuil répondant 0 ou 1 selon que le résultat est supérieur ou non à un seuil prédéfini.

Il faut attendre 1958 pour voir la naissance d'une méthode analytique d'adaptation des poids au sein d'un modèle multicouche appelé perceptron [Rosenblatt 1962]. De nombreuses études théoriques furent alors menées sur les capacités d'apprentissage des réseaux de neurones artificiels jusqu'en 1969, date à laquelle Minsky et Papert publièrent [Minsky 1969] une analyse théorique mettant en évidence les limites des réseaux de neurones (problème de la modélisation du OU exclusif). Les chercheurs se tournèrent alors vers un nouveau courant de pensée, celui de la représentation symbolique, à la base des systèmes experts.

Quelques décennies plus tard certains chercheurs, fidèles aux réseaux de neurones, ont permis la reformulation de ces réseaux neuromimétiques. Le développement le plus significatif est l'écriture en 1986 par Rumelhart [Rumelhart 1986] d'un algorithme d'apprentissage purement mathématique pour des réseaux comportant plusieurs couches de neurones (algorithme fondé sur la rétropropagation du gradient).

Depuis, les travaux se succèdent à vive allure au niveau de l'architecture et de l'algorithme d'apprentissage, et aujourd'hui, les réseaux de neurones apparaissent comme des outils pratiques pour le traitement de l'information. En effet, ils sont très utilisés dans les domaines de la reconnaissance de formes et de la modélisation des phénomènes non-linéaires pour l'automatique. L'utilisation des méthodes par modèle inverse

[Kuperstein 1988][Funashi 1989][Narendra 1990][Levin 1991], sont couramment utilisées en automatique. Celle-ci reposent sur une approche qui procède en deux étapes successives : dans un premier temps, on cherche à découvrir un modèle inverse du processus en utilisant sa sortie. Dans un second temps, le réseau est placé devant un processus pour le conduire en boucle ouverte, avec tous les problèmes de robustesse que cela engendre. Nous tenterons ici de nous démarquer de cette méthode.

Au cours de ce chapitre, nous verrons comment la commande booléenne exposée dans le chapitre II peut être déterminée par le réseau de neurones artificiels à l'aide d'une méthode de classification. Pour cela, nous effectuerons tout d'abord quelques rappels sur l'architecture et le fonctionnement d'un réseau de neurones multicouche. Nous examinerons alors comment celui-ci est utilisé pour la commande des systèmes à entrées booléennes.

La méthodologie mise en oeuvre est la suivante : premièrement, nous initialisons le réseau de neurones artificiels. Dans une seconde partie, le traitement des données et le filtrage sont étudiés afin de réduire les perturbations. Enfin la troisième partie de la méthode traite de la structure du réseau de neurones et de son apprentissage en ligne afin d'obtenir une commande convenable à appliquer au système à entrées booléennes. Ce processus d'apprentissage en ligne implique que le réseau apprend en même temps qu'il commande le système.

Notons que la méthode proposée a été simulée pour la commande d'un modèle de moteur synchrone, associé à son convertisseur de puissance, pour un suivi de trajectoire de la vitesse du moteur.

## III.2 Présentation

Le désavantage de la méthode présentée au chapitre II est que la connaissance du modèle est exigée pour déterminer les vecteurs variations  $\vec{V}_i$ . En effet, le vecteur  $\vec{V}_i$  (associé à  $(\dot{x})_i$ ) est calculé à partir du modèle pour chaque configuration  $Config_i(u)$ . Rappelons que les configurations  $Config_i(u)$  sont les vecteurs de commande possible ( $2^m$ ) contenant des valeurs booléennes tels que :

$$\{Config_i(u) / i = 1, \dots, 2^m\} = \left\{ \begin{pmatrix} 0 \\ 0 \\ 0 \\ \vdots \\ 0 \end{pmatrix} \begin{pmatrix} 1 \\ 0 \\ 0 \\ \vdots \\ 0 \end{pmatrix} \begin{pmatrix} 0 \\ 1 \\ 0 \\ \vdots \\ 0 \end{pmatrix} \dots \begin{pmatrix} 0 \\ 0 \\ 0 \\ \vdots \\ 1 \end{pmatrix} \begin{pmatrix} 1 \\ 1 \\ 0 \\ \vdots \\ 0 \end{pmatrix} \begin{pmatrix} 1 \\ 0 \\ 1 \\ \vdots \\ 0 \end{pmatrix} \dots \begin{pmatrix} 1 \\ 0 \\ \dots \\ \vdots \\ 1 \end{pmatrix} \begin{pmatrix} 1 \\ 1 \\ \dots \\ \vdots \\ 1 \end{pmatrix} \right\} \quad (III-1)$$

Ce problème de connaissance du modèle se pose d'autant plus que les variations paramétriques du système sont importantes. Nous proposons de résoudre ce problème en utilisant les réseaux de neurones, du fait de leur aptitude à commander le système sans connaissance explicite du modèle associé. L'objectif du réseau est de trouver la configuration  $i$  telle que l'état courant du système converge vers l'état désiré. Ce réseau ainsi défini effectue une association entre la commande et les directions d'évolution du système. Le schéma de la commande par réseau de neurones est alors réalisé de la manière suivante :

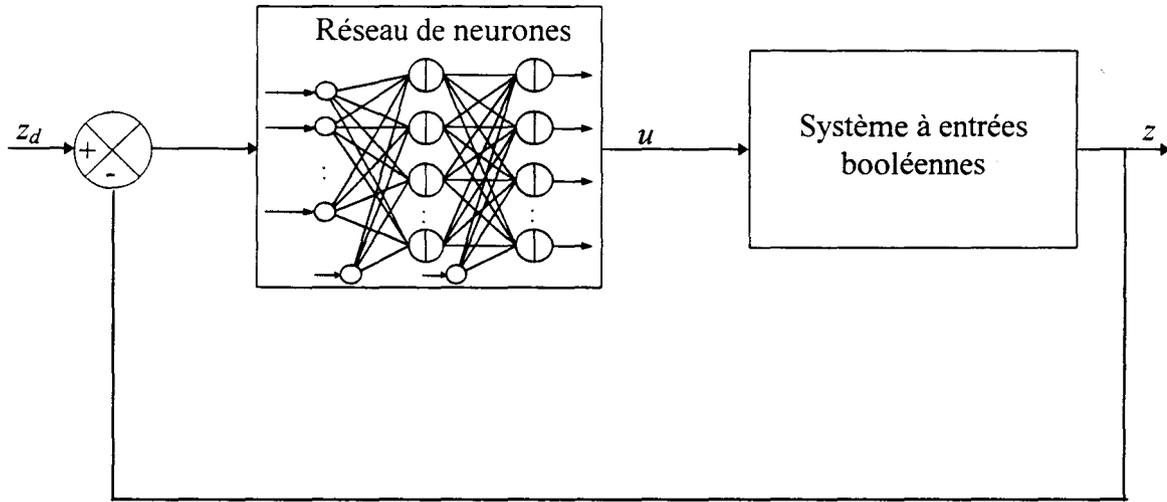


Figure III-1 : Commande par réseau de neurones.

avec  $z$  la sortie mesurée du processus et  $z_d$  la consigne désirée.

Examinons dans le prochain paragraphe comment le réseau de neurones est défini pour réaliser la commande booléenne.

### III.3 Les Réseaux de Neurones

Dans cette partie nous nous intéresserons seulement aux aspects théoriques de la commande booléenne par réseau de neurones. En premier lieu nous définirons la structure de notre réseau neuromimétique puis grâce à ce réseau, nous verrons comment la commande booléenne peut être réalisée.

Pour plus de détails sur les réseaux de neurones le lecteur pourra se référer aux ouvrages suivants [Davallo 1993][[Abdi 1994][Jodouin 1994][Patterson 1996] dans lesquels plusieurs structures de réseaux neuromimétiques et d'algorithmes d'apprentissage, ainsi que leurs utilisations sont détaillées.

#### III.3.1 Structure d'un réseau neuromimétique

Les réseaux neuromimétiques peuvent être définis comme étant une interconnexion dense entre plusieurs neurones formels, possédant des fonctions d'entrées-sorties généralement non linéaires et agissant en parallèle [Patterson 1996]. Un modèle neuronal est donc complètement défini par : le type de ses neurones (fonction d'entrées-sorties), l'architecture de ses connexions et la règle d'apprentissage (règle de modification des poids des connexions).

La structure interne d'un neurone est définie ainsi :

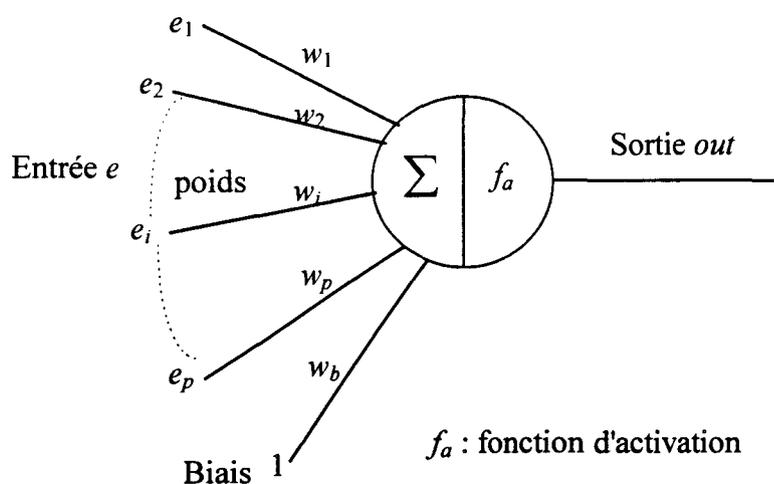


Figure III-2 : Modèle du neurone

L'activité interne d'un simple neurone par rapport à ses entrées  $e_i$  est calculée de la manière suivante :

$$\begin{cases} net = \sum_{i=1}^p w_i e_i + w_b \\ out = f_a(net) \end{cases} \quad (\text{III-2})$$

Un terme  $w_b$  appelé le terme de biais est connecté aux neurones. Ce biais est pris en compte dans la somme définie précédemment. La fonction  $f_a$  ou fonction d'activation définit l'état interne du neurone en fonction de l'entrée totale. Elle joue un rôle prépondérant dans le comportement du neurone. Il existe dans la littérature de nombreuses manières de définir ces fonctions d'activation. Néanmoins, on distingue deux catégories principales :

- les fonctions non dérivables (fonction seuil)
- les fonctions dérivables (fonction linéaire, sigmoïde ou gaussienne).

Nous utiliserons par la suite des fonctions sigmoïdes ayant pour expression :

$$f_a(net) = \frac{1}{1 + e^{-net}} \quad (\text{III-3})$$

L'architecture d'un réseau peut être à connectivité totale (tous les neurones du réseau sont reliés les uns aux autres) ou à connectivité locale (les neurones ne sont reliés qu'à leurs plus proches voisins). Nous préférons de manière générale cette deuxième structure (réseau à structure régulière) permettant une simplification de l'algorithme d'apprentissage.

Dans notre cas nous avons choisi un réseau multicouche unidirectionnel qui est le plus fréquemment rencontré dans les problèmes de modélisation ou de reconnaissance des formes. Les neurones sont disposés en ensembles élémentaires appelés communément couches.

Le réseau de neurones utilisé ici est constitué de trois parties. Une première couche est une couche tampon chargée de distribuer les entrées. La seconde partie est formée d'une ou plusieurs couches intermédiaires, ce sont les couches cachées. Enfin, il existe une troisième couche qui assure la redistribution des sorties. Il est à noter que chaque neurone d'une couche est connecté à un neurone d'une couche adjacente par l'intermédiaire d'un arc

pondéré. Le calcul de ces poids est réalisé par un algorithme d'apprentissage qui sera présenté paragraphe III.4.3.

De plus nous avons besoin de trois couches pour obtenir les relations entre les différentes configurations et l'évolution de l'état du système à entrées booléennes. Le nombre de couches cachées est donc réduit à un.

La dimension de la couche d'entrée est égale au nombre de variables d'état à commander. La dimension de la couche de sortie est égale aux  $2^m$  configurations du vecteur de commande. Le choix du nombre de neurones de la couche cachée résulte du bon compromis entre l'efficacité et la précision désirées. Des résultats satisfaisants sont obtenus avec le nombre de neurones de la couche cachée égale aux nombres de neurones de la couche de sortie. Le réseau ainsi défini, permet la classification des  $2^m$  configurations. Sa structure de base représentée ci-dessous (Figure III-3), permet une commande booléenne satisfaisante quant aux objectifs que nous nous sommes fixés au paragraphe précédent.

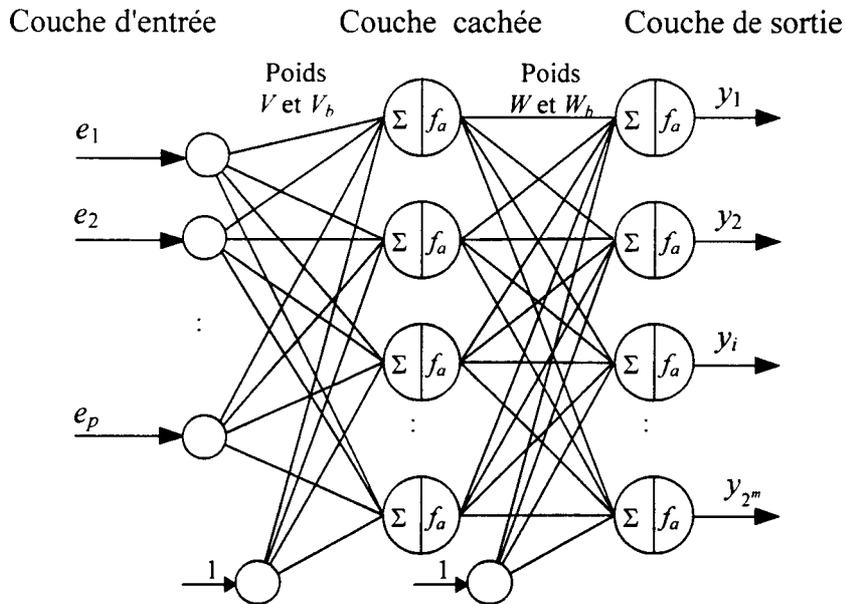


Figure III-3 : Représentation du réseau de neurones artificiel choisi

La propagation des données à travers un neurone  $i$  (valeur de sortie  $y_i$ ) de la couche de sortie est réalisée de la manière suivante :

$$y_i = f_a \left( \sum_{j=1}^{2^m} \left( w_{ij} f_a \left( \sum_{k=1}^p v_{jk} e_k + v_{b,j} \right) + w_{b,i} \right) \right) \quad (\text{III-4})$$

avec :

$e_i$  :  $i^{\text{ème}}$  entrée du réseau.

$v_{ji}$  : poids de la connexion entre le  $i^{\text{ème}}$  neurone de la couche d'entrée et le  $j^{\text{ème}}$  neurone de la couche cachée.

$v_{b,j}$  : poids du biais du  $j^{\text{ème}}$  neurone de la couche cachée.

$w_{ij}$  : poids de la connexion entre le  $j^{\text{ème}}$  neurone de la couche cachée et le  $i^{\text{ème}}$  neurone de la couche de sortie.

$w_{b,i}$  : poids du biais du  $i^{\text{ème}}$  neurone de la couche de sortie.

La sortie du réseau d'une manière générale peut s'écrire sous forme matricielle telle que :

$$y = F_a \left( W \left( F_a (Ve + V_b) \right) + W_b \right) \quad \text{(III-5)}$$

avec  $F_a(NET) = \left[ f_a(net_1) \quad f_a(net_2) \quad \dots \quad f_a(net_{2^m}) \right]^T$

ou  $net_q$  est la somme pondérée du  $q^{\text{ème}}$  neurone, et  $NET = \left[ net_1 \quad net_2 \quad \dots \quad net_{2^m} \right]^T$

$$y = \begin{bmatrix} y_1 \\ y_2 \\ \vdots \\ y_{2^m} \end{bmatrix}, e = \begin{bmatrix} e_1 \\ e_2 \\ \vdots \\ e_p \end{bmatrix}, W = [w_{ij}], V = [v_{ji}], W_b = [w_{b,i}] \text{ et } V_b = [v_{b,j}]. \quad \text{(III-6)}$$

### III.4 Utilisation en Commande Booléenne

Le but de cette partie est de définir l'algorithme général à partir duquel le réseau de neurones réalisera une commande booléenne. Cet algorithme peut être décomposé en quatre étapes : l'initialisation, le traitement des données d'entrée, l'apprentissage et la commande.

### III.4.1 Initialisation

Au début de l'algorithme, lorsque aucun traitement n'a été effectué, l'initialisation des poids du réseau est nécessaire. Deux situations peuvent se présenter :

- Si l'apprentissage du réseau a déjà été réalisé une fois, alors nous assignons aux poids du réseau les dernières valeurs utilisées.
- Si aucun apprentissage a été exécuté, les poids initiaux sont fixés aléatoirement.

Dans un second temps, après l'étape d'initialisation des poids, il est nécessaire d'acquérir un ensemble de mesures pour obtenir les données d'apprentissage initiales, qui seront utilisées dans la prochaine étape. Cette acquisition de mesure est réalisée en appliquant  $r$  fois chaque configuration  $Config_i(u)$  avec  $i = \{1 \dots 2^m\}$  au système.  $r$  est choisi pour éviter les erreurs dues aux bruits de mesure (nous expliquerons ce filtrage des données dans le prochain paragraphe). L'acquisition des mesures est accompli à chaque période d'échantillonnage  $T_e$ . En conséquence cette étape d'initialisation a une durée de  $2^m r T_e$ . L'apprentissage du réseau et la commande du processus ne peuvent commencer qu'après cette étape préliminaire.

### III.4.2 Traitement des données

Le paragraphe suivant présente le traitement de données, qui consiste en une définition convenable des données d'apprentissage entrée-sortie du réseau de neurones. En effet, le but du réseau est de fournir une association entre les variations d'état et les configurations pour commander le processus.

Définissons pour chaque configuration  $i$ , un vecteur variation de la manière suivante :

$$(\Delta z_k)_i = (z_k)_i - z_{k-1} \quad (\text{III-7})$$

où

- $z_k$  est le vecteur des variables mesurées du système discrétisé.
- $(\Delta z_k)_i$  désigne le vecteur variation à l'instant  $k$  quand la configuration  $i$  est appliquée au système binaire à l'instant  $k - 1$

Ces **vecteurs variations** sont alors définis comme les **données d'apprentissage** du réseau de neurones.

Notons que cette opération est sensible aux bruits de mesure. En conséquence pour réduire l'influence de ces bruits de perturbations, un filtrage des données est nécessaire. La méthode la plus simple et la plus efficace consiste à effectuer pour chaque configuration  $i$  la moyenne sur les  $r$  derniers vecteurs variations. Pour des raisons de commodité, nous désignons par  $\{\Delta z\}_i$ , l'ensemble des  $r$  derniers vecteurs variations de la configuration  $i$ , et  $\overline{\{\Delta z\}}_i$ , sa moyenne associée pour le filtrage. Cette procédure est exécutée en ligne. Par conséquent, à chaque période d'échantillonnage un nouveau vecteur variation  $(\Delta z_k)_i$ , est calculé. L'ensemble  $\{\Delta z\}_i$ , et sa moyenne sont alors mis à jour.

Une illustration de ce traitement est présentée Figure III-4, dans laquelle nous considérons  $z \in \mathbb{R}^2$ ,  $u \in \{0,1\}^2$  et  $r = 3$ .

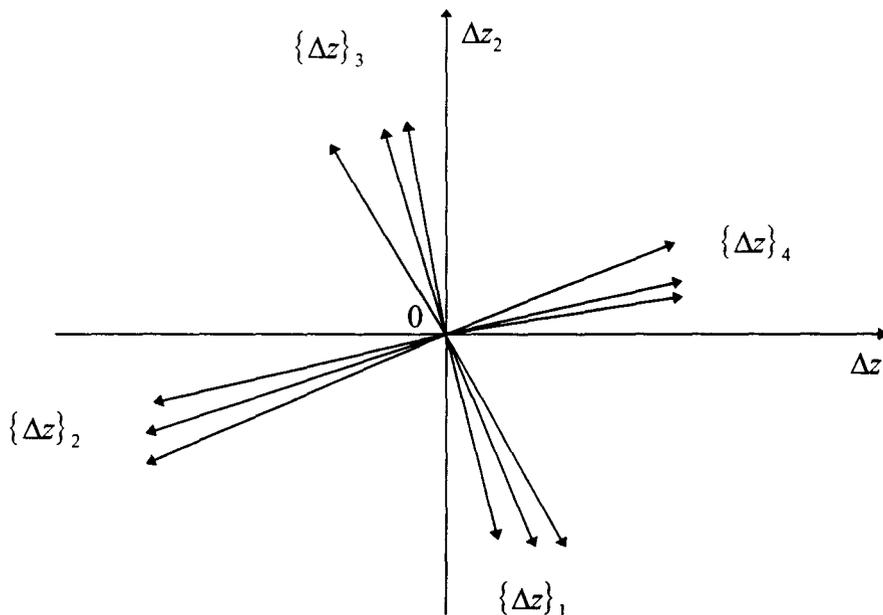


Figure III-4 : Vecteurs variations pour chaque configuration  $i = \{1 \dots 4\}$ .

Les résultats de la moyenne  $\overline{\{\Delta z\}}_i$ , sont les données d'entrée du réseau de neurones. A présent nous devons également déterminer les données de sortie, qui seront utilisées pour l'apprentissage du réseau de neurones. Dans le cas de notre réseau la  $i^{\text{ème}}$  sortie est associée à la configuration  $i$ . En conséquence le vecteur de sortie désiré, appelé également vecteur cible  $(y^d)_i$ , est choisi de la manière suivante :

$$(y^d)_i = \begin{bmatrix} y_1^d \\ y_2^d \\ \vdots \\ y_j^d \\ \vdots \\ y_{2^m}^d \end{bmatrix} \text{ avec } \begin{cases} y_j^d = 1 \text{ pour } j = i, \\ y_j^d = 0 \text{ autrement.} \end{cases} \quad (\text{III-8})$$

Finalement, nous avons déduit des paires de données d'entrée-sortie pour chaque configuration  $i$ . On définit alors l'ensemble  $S$  d'échantillons de la forme suivante :

$$S = \left\{ \left( \{\overline{\Delta z}\}_1, (y^d)_1 \right), \left( \{\overline{\Delta z}\}_2, (y^d)_2 \right), \dots, \left( \{\overline{\Delta z}\}_i, (y^d)_i \right), \dots, \left( \{\overline{\Delta z}\}_{2^m}, (y^d)_{2^m} \right) \right\} \quad (\text{III-9})$$

### III.4.3 Algorithme d'apprentissage

Le problème de l'apprentissage consiste à ajuster les poids des connexions du réseau en ligne, en utilisant l'ensemble des données de  $S$ . La stratégie d'apprentissage la plus fréquemment rencontrée est celle de l'algorithme de la rétropropagation du gradient [8]. Le principe est de minimiser pour toutes les paires d'entrée-sortie dénotées  $(e, y^d)$  de l'ensemble  $S$ , le critère quadratique  $J$  suivant :

$$J = \frac{1}{2} \varepsilon_{nn}^T \varepsilon_{nn} \quad (\text{III-10})$$

où le vecteur erreur  $\varepsilon_{nn}$  (indice  $nn$  pour "neural network") est donné par la différence entre la sortie désirée  $y^d$  et la sortie  $y$  du réseau de neurones obtenue pour l'entrée  $e$ .

$$\varepsilon_{nn} = y^d - y \quad (\text{III-11})$$

L'algorithme utilisé pour minimiser ce critère, est généralement la méthode de la descente du gradient. Cette méthode d'optimisation entraîne la modification de chaque poids  $w$  (poids entre les couches et poids des biais) par la loi d'adaptation ci-dessous :

$$\Delta w = -\eta \frac{\partial J}{\partial w} \quad (\text{III-12})$$

$\eta$  étant le coefficient d'apprentissage qui influence la vitesse de convergence des poids.

En appliquant cet algorithme aux poids du réseau (annexe C), nous obtenons les vecteurs gradients dénotés  $\delta_w$  entre la couche de sortie et la couche cachée et  $\delta_i$  entre la couche cachée et la couche d'entrée :

$$\begin{cases} \delta_w = (y^d - y) * F_a' (W(F_a(Ve + V_b)) + W_b) \\ \delta_i = (W^T \delta_w) * F_a' (Ve + V_b) \end{cases} \quad (III-13)$$

avec  $F_a'(NET) = [f_a'(net_1) \quad f_a'(net_2) \quad \dots \quad f_a'(net_{2^m})]^T$  la dérivée de  $F_a(NET)$ , ou  $f_a'(net_q)$  est elle même la dérivée de  $f_a$  par rapport à  $net_q$  :

$$f_a'(net_q) = \frac{\partial f_a(net_q)}{\partial net_q} = \frac{e^{-net_q}}{(1 + e^{-net_q})^2} \quad (III-14)$$

Dans cette écriture \* correspond au produit de HADAMAR. Ce produit correspond aux produit terme à terme de deux vecteurs. Par exemple, le produit de HADAMAR de deux vecteurs  $x^1, x^2$  est défini comme :

$$x^1 * x^2 = \begin{bmatrix} x_1^1 \\ x_2^1 \\ x_3^1 \end{bmatrix} * \begin{bmatrix} x_1^2 \\ x_2^2 \\ x_3^2 \end{bmatrix} = \begin{bmatrix} x_1^1 x_1^2 \\ x_2^1 x_2^2 \\ x_3^1 x_3^2 \end{bmatrix} \quad (III-15)$$

Par conséquent, les lois d'adaptation sont :

$$\begin{cases} W_{new} = W_{old} + \eta \delta_w [F_a(Ve + V_b)]^T \\ W_{b,new} = W_{b,old} + \eta_b \delta_w \\ V_{new} = V_{old} + \eta \delta_i [e]^T \\ V_{b,new} = V_{b,old} + \eta_b \delta_i \end{cases} \quad (III-16)$$

Le gain d'apprentissage pour les poids des biais est noté  $\eta_b$ . Nous considérons ici que  $\eta_b < \eta$ , et que leurs valeurs respectives ne sont pas trop grandes par rapport aux variations des poids  $W$  et  $V$ . En effet, des valeurs trop grande des coefficients  $\eta_b$  et  $\eta$  entraînerait l'instabilité du réseau de neurones.

A chaque période d'échantillonnage  $k$ , les poids des matrices  $W, W_b, V$  et  $V_b$  sont mis à jour grâce aux lois d'adaptation (III-16) et (III-18). Cette mise à jour est réalisée avec toutes

les paires d'entrées-sorties de l'ensemble  $S$ , même si une seule paire  $(\overline{\{\Delta z\}}_i, (y^d)_i)$  de  $S$  est modifiée au  $k^{\text{ème}}$  instant. L'intérêt d'utiliser toutes les données d'apprentissage est d'accélérer l'adaptation des poids pour permettre l'amélioration du taux de convergence de l'apprentissage. De plus, cette technique renforce l'apprentissage du réseau et donc la classification des  $2^m$  configurations. En outre, la disparition ou la réduction d'une classe est évitée. Grâce à cette stratégie d'apprentissage, le réseau peut être parfaitement utilisé pour la commande des systèmes à entrées booléennes.

### Initialisation et procédure d'apprentissage

*Etape 1* : Pour l'instant  $k = 0$ .

- ◆ Si l'apprentissage n'a jamais été effectué, alors initialiser les poids aléatoirement. Autrement reprendre les poids précédents,
- ◆ Mesurer  $z_0$ ,
- ◆ Affecter la valeur  $i=1$  et appliquer  $Config_i(u)$  au système pendant la période d'échantillonnage  $T_e$ .

*Etape 2* : Pour  $k = 1$  à  $2^m r - 1$ .

- ◆ Mesurer  $z_k$  et calculer les variations  $(\Delta z_k)_i$ ,
- ◆ Mettre à jour la moyenne  $\overline{\{\Delta z\}}_i$ , et donc l'ensemble  $S$  avec la nouvelle paire d'entrée-sortie  $(\overline{\{\Delta z\}}_i, (y^d)_i)$ ,
- ◆ Affecter la valeur de  $i = k \bmod 2^m + 1$  et appliquer la  $Config_i(u)$  au système.

*Etape 3* : Pour  $k \geq r2^m$

- ◆ Mesurer  $z_k$  et calculer les variations  $(\Delta z_k)_i$ ,
- ◆ Mettre à jour la moyenne  $\overline{\{\Delta z\}}_i$ , et donc l'ensemble  $S$  avec la nouvelle paire d'entrée-sortie  $(\overline{\{\Delta z\}}_i, (y^d)_i)$ ,

- ◆ Mettre à jour les poids du réseau en utilisant les équations (III-13) et (III-16) pour les  $2^m$  paires d'apprentissage de l'ensemble  $S$ ,
- ◆ Aller à la procédure de commande pour déterminer la configuration  $Config_i(u)$  à appliquer au système.

### III.4.4 Algorithme de commande

Dans ce paragraphe nous présenterons la loi de commande, dont le but est de réduire l'erreur entre la position courante du système et la position désirée, en trouvant la "bonne" configuration à appliquer au système. Comme il a été établi précédemment, le réseau de neurones est défini pour permettre l'association entre toutes les configurations et les vecteurs variations des variables mesurées du système. Dès lors, le réseau de neurones effectue la classification par apprentissage des  $2^m$  configurations, qui dépendent du vecteur d'entrée du réseau  $e$ .

Définition : Soit un vecteur d'entrée  $e$  quelconque, soit  $i \in \{1..2^m\}$  alors  $e$  appartient à la classe dénoté  $C_i$  si et seulement si  $y_i$  est la composante de valeur maximale du vecteur de sortie  $y$ .

$$e \in C_i \quad \text{ssi} \quad y_i > y_j, \forall i \neq j \quad (\text{III-17})$$

avec

$$y = F(W(F(Ve + V_b)) + W_b) \quad (\text{III-18})$$

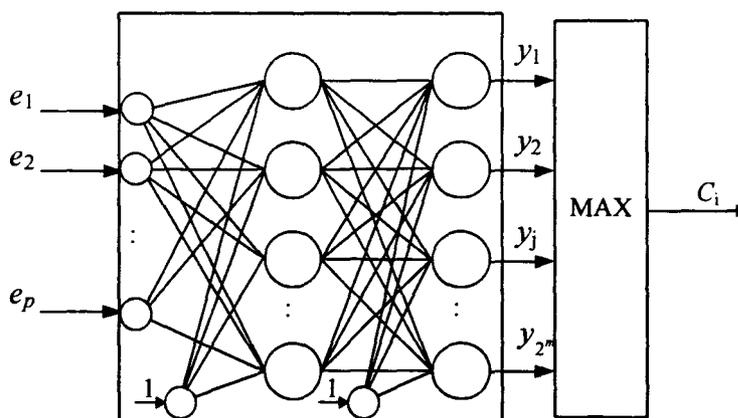


Figure III-5 : Détermination de la  $i^{\text{ème}}$  Classe

Un exemple de classification par un réseau de neurones avec deux entrées et quatre classes est présenté ci-dessous :

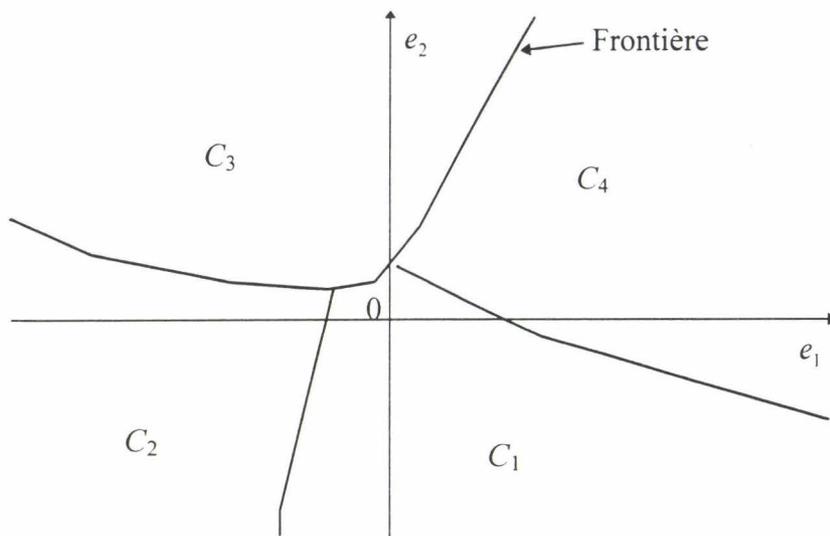


Figure III-6 : Classification du réseau de neurones  $i = \{1 \dots 4\}$

Le but de cette classification est de choisir la configuration qui donne la valeur minimale de l'erreur en poursuite. L'expression de cette erreur est écrite comme :

$$\varepsilon = z_d - z \quad (\text{III-19})$$

où  $z_d$  est l'état désiré et  $z$  l'état mesuré du processus.

En appliquant alors ce vecteur  $\varepsilon$  à l'entrée du réseau, nous obtenons un vecteur de sortie  $y$  dont la composante maximum donne la valeur de la classe d'appartenance. A partir de cette classe, nous déduisons la configuration correspondante et donc le vecteur de commande associé.

### Procédure de commande

*Etape 1:* Calculer l'erreur de position  $\varepsilon$

*Etape 2:* Calculer la sortie du réseau  $y$  avec  $\varepsilon$  en entrée.

*Etape 3:* Déterminer l'élément maximum  $y_i$  de  $y$  pour déduire la classe exacte  $i$  à choisir.

*Etape 4:* Sélectionner le vecteur de commande  $Config_i(u)$

*Etape 5:* Appliquer le vecteur commande au système pendant la période

d'échantillonnage  $T_e$ .

## III.5 Application au moteur synchrone

### III.5.1 Présentation de la machine synchrone

Une application a été réalisée, dans le but d'évaluer l'efficacité de l'algorithme de commande booléenne proposé. Le système utilisé pour obtenir les résultats de simulation est composé d'une machine synchrone et d'un convertisseur de puissance (Figure III-7). Le convertisseur de puissance est composé de six transistors de commutation ( $S_1, S_2, S_3, S_4, S_5, S_6$ ) et alimenté par une tension continue  $E_o$ . Le moteur synchrone comprend trois enroulements au stator et trois enroulements au rotor. Pour plus de détails sur le moteur synchrone, on pourra se référer aux ouvrages suivants : [Le Pioufle 1990] [Bergmann 1986][ Leonhard 1985].

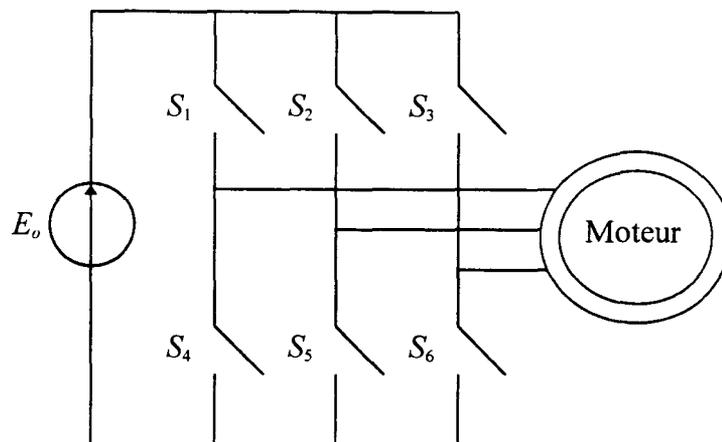


Figure III-7 : Schéma électrique du moteur synchrone et du convertisseur de puissance.

Dans cette approche du problème de commande de moteur, un modèle mathématique est exigé pour simuler son comportement. Ce modèle mathématique du système [Ducreux 1992] est donné ci-dessous, en prenant les courants statoriques, la position angulaire et la vitesse comme variable d'état.

$$\begin{cases} \frac{di_\alpha}{dt} = -\frac{R}{L}i_\alpha + \frac{\phi_f}{L}\omega \sin(\theta) + \frac{1}{L}u_\alpha \\ \frac{di_\beta}{dt} = -\frac{R}{L}i_\beta - \frac{\phi_f}{L}\omega \cos(\theta) + \frac{1}{L}u_\beta \\ \frac{d\theta}{dt} = \omega \\ \frac{d\omega}{dt} = -\frac{f}{J}\omega + \frac{\phi_f}{J}(i_\beta \cos(\theta) - i_\alpha \sin(\theta)) - \frac{C_r}{J} \end{cases} \quad (\text{III-20})$$

$i_\alpha, i_\beta$  sont les intensités statoriques équivalentes après la réduction du modèle (Transformation T32).  $\theta$  représente l'angle du rotor par rapport au référentiel fixe statorique.  $\omega$  est la vitesse de la machine synchrone.  $\phi_f$  représente l'amplitude de flux rotorique.

$u_\alpha$  et  $u_\beta$  sont les tensions appliquées. Elles peuvent être exprimées en fonction des grandeurs booléennes de commutation :

$$\begin{cases} u_\alpha = \frac{E_o}{\sqrt{6}}(2u_1 - u_2 - u_3) \\ u_\beta = \frac{E_o}{\sqrt{2}}(u_2 + u_3) \end{cases} \quad (\text{III-21})$$

Les grandeurs  $u_i$  (avec  $i = 1, 2, \dots, 6$ ) représentent les booléens associés aux 6 transistors de commutation ( $S_1, S_2, \dots, S_6$ ). Si  $u_i = 0$  alors le  $i^{\text{ème}}$  transistor  $S_i$  passe de l'état ON à OFF sinon  $u_i = 1$  et le transistor passe de l'état OFF à l'état ON. Pour des raisons physiques (complémentarité des interrupteurs) les grandeurs  $u_i$  sont liées par la relation suivante :

$$u_i + u_{i+3} = 1 \text{ pour } i = 1, 2, 3 \quad (\text{III-22})$$

Nous posons  $u = [u_1 \quad u_2 \quad u_3]^T$  comme étant le vecteur de commande du système.

Les autres paramètres sont définis comme suit:

$R$  : Résistance statorique.

$L$  : Inductance statorique.

$E_o$  : Tension appliquée au convertisseur de puissance.

$f, J$  : Frottement et inertie du rotor.

$C_r$  : Couple résistant.

Notre but est de commander la machine synchrone afin d'obtenir une vitesse désirée  $\omega_d$  [Leonhard 1985]. Le réglage de la vitesse  $\omega$  de la machine synchrone est effectué en utilisant l'équation de vitesse non-linéaire.

$$\frac{d\omega}{dt} = -\frac{f}{J}\omega + \frac{\phi_f}{J}(i_\beta \cos(\theta) - i_\alpha \sin(\theta)) - \frac{C_r}{J} \quad (\text{III-23})$$

Pour piloter le système vers une vitesse désirée  $\omega_d$ , nous pouvons montrer qu'il est suffisant de commander les courants statoriques, et de lui fournir deux formes d'ondes sinusoïdales comme courants de référence. Ces courants de référence sont définis par les relations suivantes :

$$\begin{cases} i_{\alpha ref} = -i_o \sin(\theta) \\ i_{\beta ref} = i_o \cos(\theta) \end{cases} \quad (\text{III-24})$$

En introduisant ces expressions à l'intérieur de l'équation (III-23), nous obtenons :

$$\frac{d\omega}{dt} = -\frac{f}{J}\omega + \frac{\phi_f}{J}i_o - \frac{C_r}{J} \quad (\text{III-25})$$

A partir de cette équation, on peut clairement déduire que la régulation de la vitesse dépend de l'amplitude  $i_o$  des courants de référence. Par exemple pour une vitesse désirée constante ( $\omega = \omega_d$  and  $\frac{d\omega}{dt} = 0$ ), l'amplitude  $i_o$  devient :

$$i_o = \frac{f}{\phi_f}\omega_d + \frac{C_r}{\phi_f} \quad (\text{III-26})$$

Le schéma proposé pour la commande de vitesse est présenté Figure III-8. La régulation du moteur est réalisée simplement par une action directe (feedforward) combinée avec un

contrôleur proportionnel-intégral (PI). L'intérêt de ce feedforward est d'améliorer les performances dynamiques de la commande. Ce compensateur ainsi défini donne alors des performances satisfaisantes que l'on peut obtenir pour cette application. Les gains du PI sont choisis de telle manière que la réponse du système soit rapide et avec peu de dépassement pour les conditions nominales. Avec ce contrôleur qui réduit l'erreur entre le signal de référence  $\omega_d$  et la sortie du système  $\omega$ , on obtient alors la loi de commande du courant  $i_o$  exprimée par :

$$i_o = \frac{f}{\phi_f} \left( \omega_d + K(\omega_d - \omega) + \frac{1}{T_i} \int_0^t (\omega_d - \omega) dt \right) + \frac{C_r}{\phi_f} \quad (\text{III-27})$$

Pour cette application, le contrôleur PI donne des résultats satisfaisants avec les gains choisis suivants :

$$K = 5 \text{ et } T_i = 2.$$

Les courants de références  $i_{\alpha}ref$  et  $i_{\beta}ref$  d'enroulement statorique sont générés par les relations (III-26) et (III-29), décrites précédemment.

Les courants de phase  $i_a$ ,  $i_b$  et  $i_c$  sont mesurés et réduits en deux composantes de courant (transformation de Park)  $i_{\alpha}$  et  $i_{\beta}$  d'une machine équivalente biphasée. Ces composantes sont comparées aux courants de référence  $i_{\alpha}ref$  et  $i_{\beta}ref$ . Les sorties de ces comparateurs sont les entrées de commande  $\varepsilon$  du réseau de neurones, et les courants  $i_{\alpha}$  et  $i_{\beta}$  sont les entrées pour l'apprentissage de ce réseau. Le processus de commande est maintenant constitué par deux sous-systèmes de régulation, l'un pour la commande de vitesse et l'autre pour le réglage des courants.

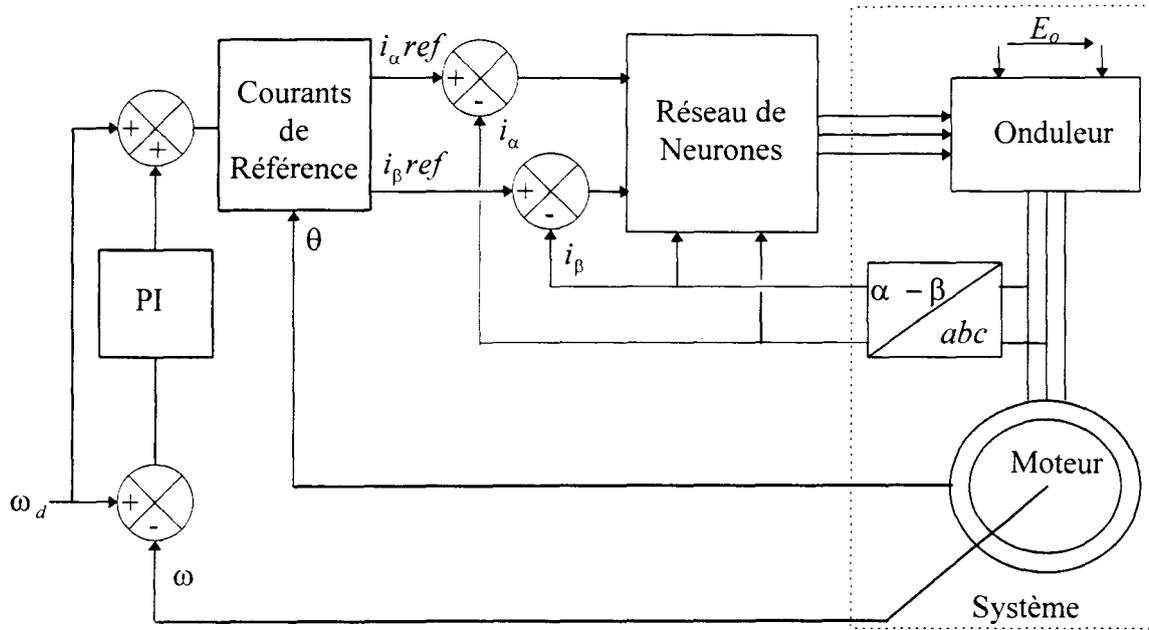


Figure III-8 : Schéma de commande

Notre objectif est la **poursuite d'une trajectoire dans l'espace d'état** par ce système. La trajectoire désirée est imposée ici par les courants de référence  $i_{\alpha.ref}$  et  $i_{\beta.ref}$ . Cet objectif est réalisé en utilisant le réseau de neurones décrit précédemment.

### III.5.2 Simulation

Le logiciel de simulation MATLAB-SIMULINK a été utilisé pour étudier la réponse du système électrique. L'ensemble moteur et commande a été programmé en langage C. La simulation des dynamiques de cet ensemble était de  $100\mu s$  d'unité de temps. Les équations de cette association moteur-commande ont été résolues par la méthode d'intégration numérique Runge-Kutta d'ordre cinq. Le convertisseur trois-phases a été simulé en supposant les éléments de commutation idéaux. La simulation de la machine synchrone a été réalisée par l'équation d'état (III-22) avec les paramètres suivants :

$$\left\{ \begin{array}{l} R : 2 \Omega \\ L : 200 \text{ mH} \\ E_o : 380 \text{ V} \\ f : 0.08 \text{ N.s} \\ J : 0.02 \text{ kg.m}^2 \\ Cr : 0.8 \text{ N.m} \\ \phi_f : 1 \text{ Wb} \end{array} \right. \quad \text{(III-28)}$$

La structure du réseau de neurones utilisé est une structure 2-8-8 (deux entrées, huit neurones dans la couche cachée, et huit neurones dans la couche de sortie). Le choix du nombre de neurones dans la couche cachée est un compromis entre précision et coût de calcul. La période d'échantillonnage du réseau de neurones est de 0.2 ms, qui correspond à une fréquence de commutation de 5 kHz. Cette période d'échantillonnage est choisie en fonction des dynamiques du système et des limitations en fréquence des composants.

A cause de la structure du convertisseur (Figure III-7), le comportement moteur pour les configurations 1 et 8 sont les mêmes, ce qui signifie que les données des classes 1 et 8 se chevauchent. En général, les classes qui se chevauchent l'une et l'autre ou qui englobe les autres ne peuvent être séparées. Par conséquent les classes 1 et 8 doivent être considérées identiques. Pour palier ce problème, les vecteurs désirés  $(y^d)_1$  et  $(y^d)_8$  sont choisis de la manière suivante :

$$(y^d)_1 = (y^d)_8 = \begin{bmatrix} y_1^d \\ 0 \\ \vdots \\ 0 \\ y_8^d \end{bmatrix} \text{ avec } y_1^d = y_8^d = 1 \quad (\text{III-29})$$

Cette nouvelle définition du vecteur-cible n'a pas d'influence sur l'algorithme de commande neuronale puisque les configurations 1 et 8 sont équivalentes.

Le réseau de neurones est entraîné à apprendre grâce à l'algorithme d'apprentissage présenté précédemment, et les vecteur-cibles ci-dessus. Les taux d'apprentissage  $\eta$  et  $\eta_b$  sont respectivement de 0.1 et de 0.01. Ils assurent une adaptation rapide et stable des poids.

De plus, nous avons pris en compte également le bruit de mesure dont les effets ne peuvent être négligés en électronique de puissance. En effet, les mesures données par un capteur classique sont affectées par le bruit inhérent au système, qui est généralement présumé être blanc. Les amplitudes des bruits pour les capteurs de courant, le capteur de position et le capteur de vitesse sont supposées être égales respectivement à 0.1 A , 0.005 rad et 0.05 rad/s. Pour réduire leur effet, nous utilisons la méthode de filtrage définie dans le paragraphe sur le traitement des données doté du coefficient  $r=10$ .

Pour illustrer la procédure d'apprentissage, déterminons le temps exigé par le réseaux pour effectuer une classification correcte des  $2^m$  configurations. Sous les conditions de simulation définie précédemment, l'évolution des poids initiaux aléatoires vers des poids satisfaisants (qui assurent une convergence en poursuite du système) se fait en moins de 0.4 s. Ainsi, dans cet exemple, environs 2000 itérations complètes sont nécessaires, quand l'apprentissage est effectué pour la première fois. Cet étape se produit seulement une fois puisque les initialisations suivantes du réseau se font avec les poids sauvegardés. En conséquence, comme le montre la figure suivante, la procédure d'apprentissage fondée sur l'algorithme de la rétropropagation et la commande dérivée du réseau mène à de bons résultats.

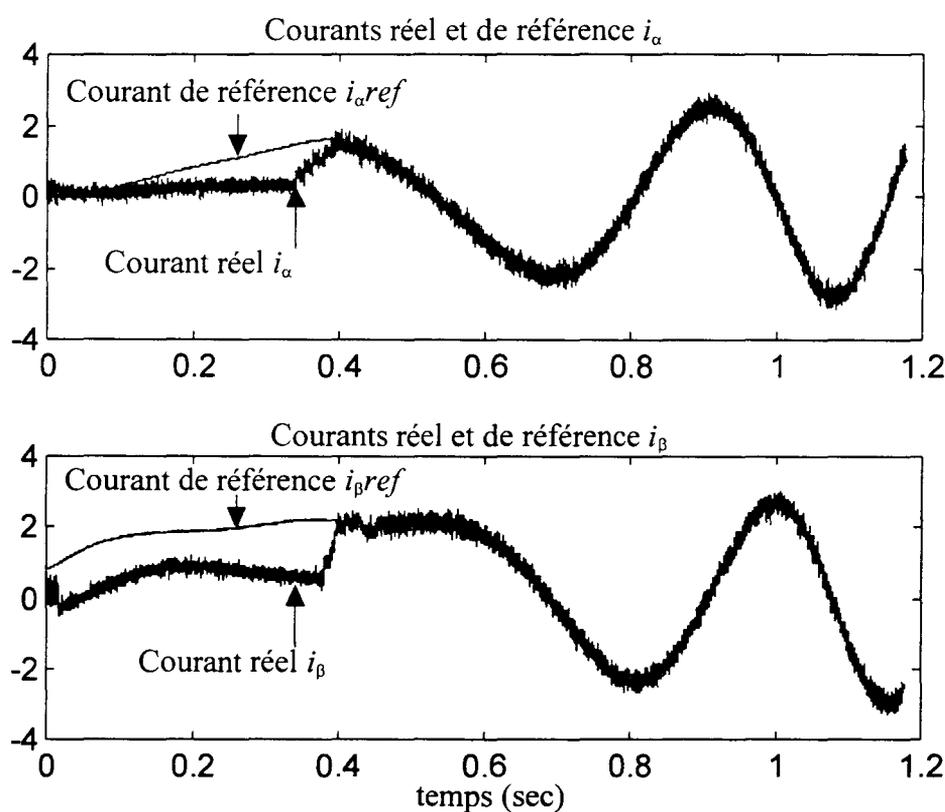


Figure III-9 : Courants réels et désirés pour le premier apprentissage.

Après cette étape préliminaire de l'apprentissage, le réseau de neurone et l'ensemble des contrôleurs sont prêts à commander les courants et la vitesse du moteur. Les figures suivantes montrent les courants réels et désirés obtenus pour une vitesse de référence croissant de 0 jusqu'à 40 rad/s en 2 secondes.

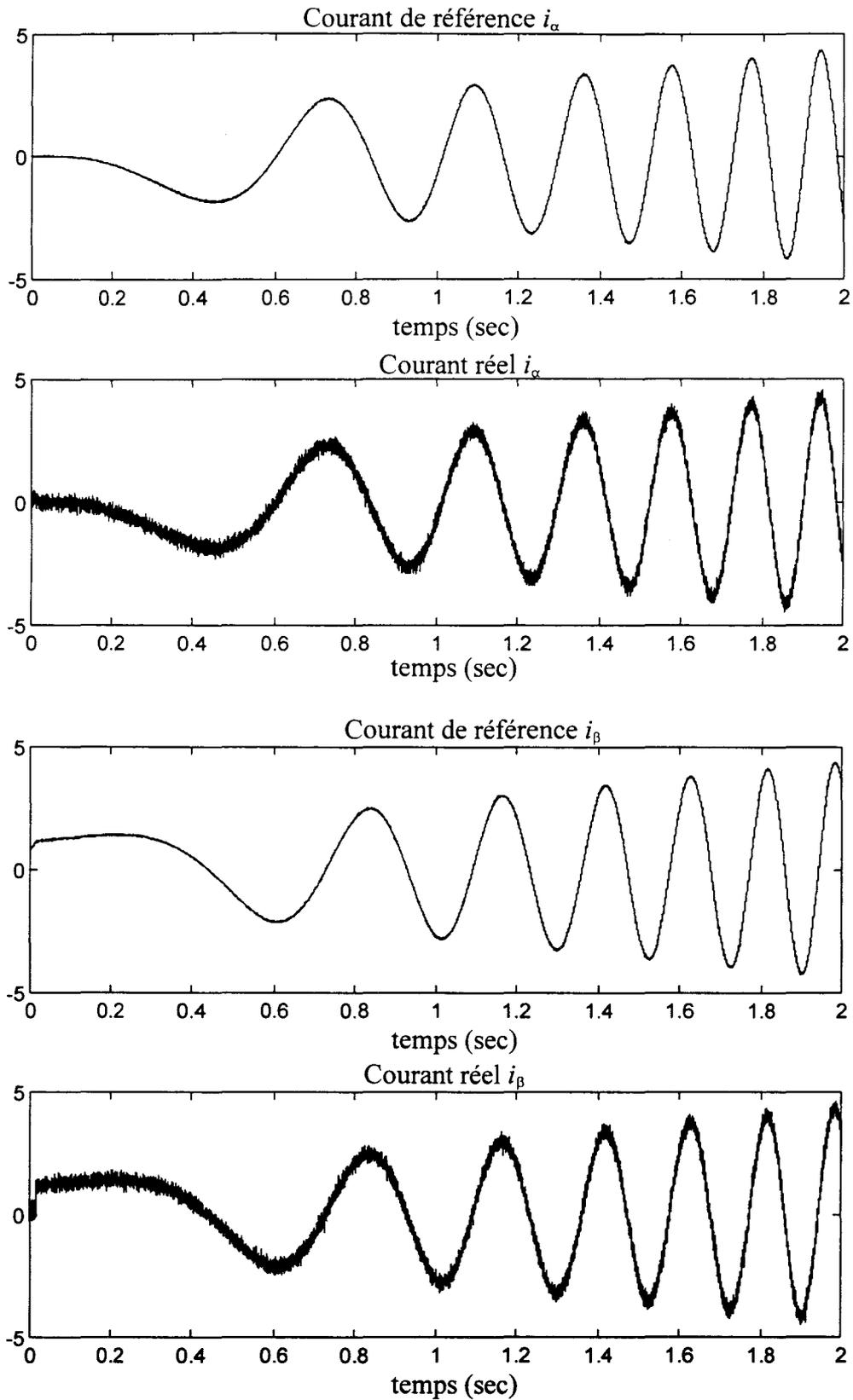


Figure III-10 : Courants réels et désirés pour une consigne de vitesse en rampe.

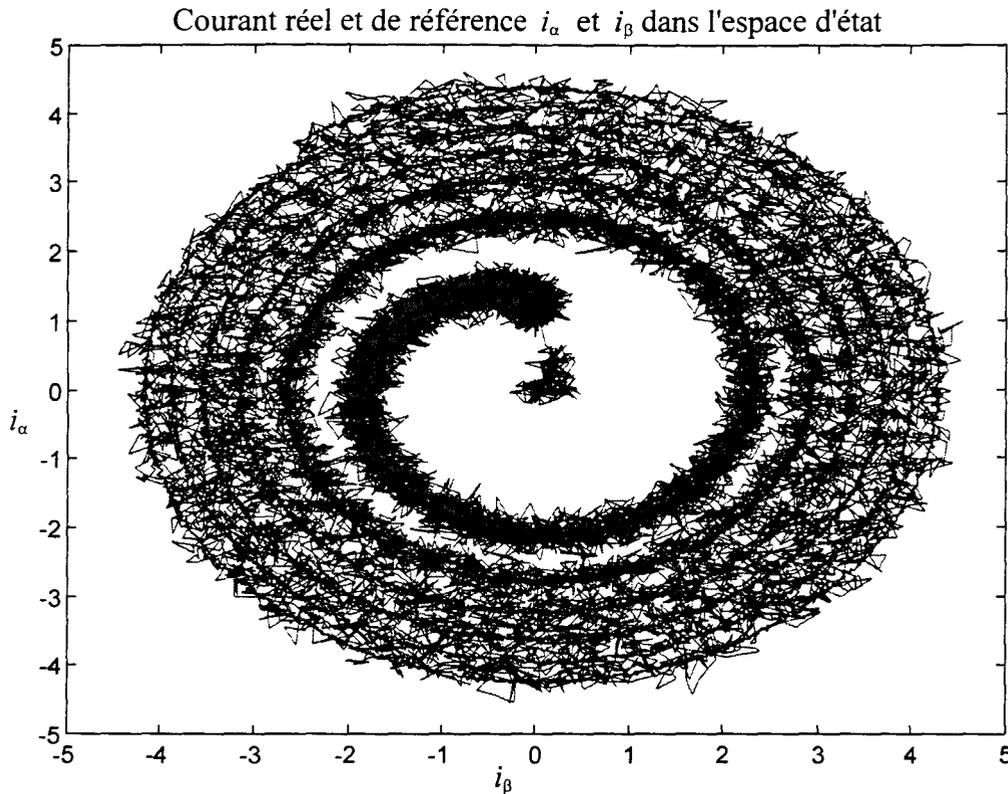


Figure III-11 : Courants réels et désirés dans l'espace d'état pour une consigne de vitesse en rampe.

Dans le but de fournir une évaluation quantitative des performances et de la précision de la commande proposée, nous calculons le critère défini comme la moyenne des erreurs absolue entre les valeurs réelles et désirées. Malgré les bruits de mesure et les commutations booléennes, les courants convergent vers les courants de référence avec précision. Les moyennes des erreurs absolues pour  $i_\alpha$  et  $i_\beta$  sont respectivement de 0.12 A et 0.13 A et les maximums des erreurs absolues sont de 0.62 A pour  $i_\alpha$  et de 1.32 A pour  $i_\beta$ . Ces valeurs importantes des maximums sont dues à l'apprentissage de départ du réseau ( évolution des poids initiaux vers des poids qui font convergé le système), toutefois celle-ci deviennent négligeable par la suite.

Une régulation précise de ces courants est absolument nécessaire pour obtenir une bonne régulation de vitesse. Nous avons vérifié l'aptitude de la commande proposée à poursuivre une référence spécifique de vitesse désirée. Les résultats sont montrés en Figure III-12.

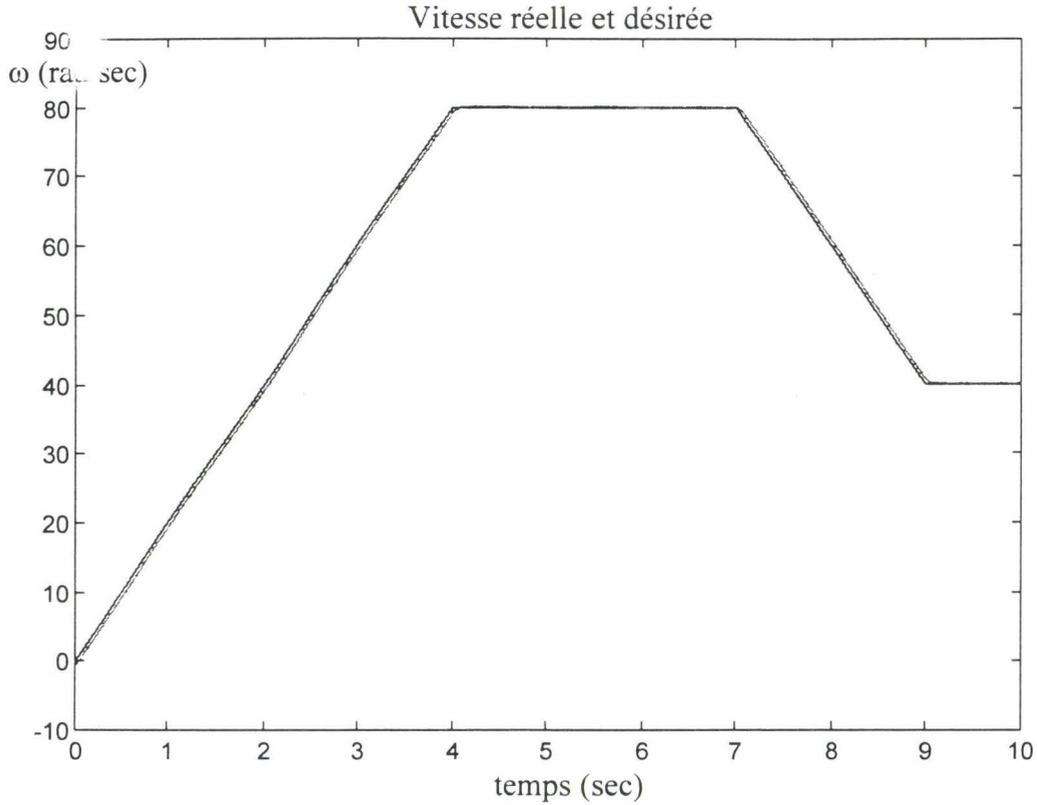


Figure III-12 : Vitesses réelle et désirée.

La vitesse désirée  $\omega_d$  est suivie avec une erreur absolue maximum de 1.1 rad/s. L'erreur moyenne absolue sur l'intervalle de simulation est de 0.51 rad/s. Toutefois, entre la 4<sup>ème</sup> et la 7<sup>ème</sup> seconde ( $\omega_d = 80$  rad/s), l'erreur moyenne absolue est de 0.05 rad/s. Il est donc possible de vérifier que la poursuite de vitesse est efficace et précise.

**L'analyse du réseau** est réalisée en traçant les classes par rapport aux variables d'entrée  $e$  du réseau de neurones. Cette représentation est une image de la classification à un instant donné pour des poids fixés. Dans le centre de la Figure III-13, nous notons que les classes 1 et 8 se chevauchent comme nous l'avons noté précédemment. Les 6 autres classes sont bien séparées et, comme le montre la Figure III-14, les limites des classes apparaissent clairement. En effet les sorties du réseau  $y_i$  varient rapidement entre les niveaux d'activation bas 0 et haut 1.

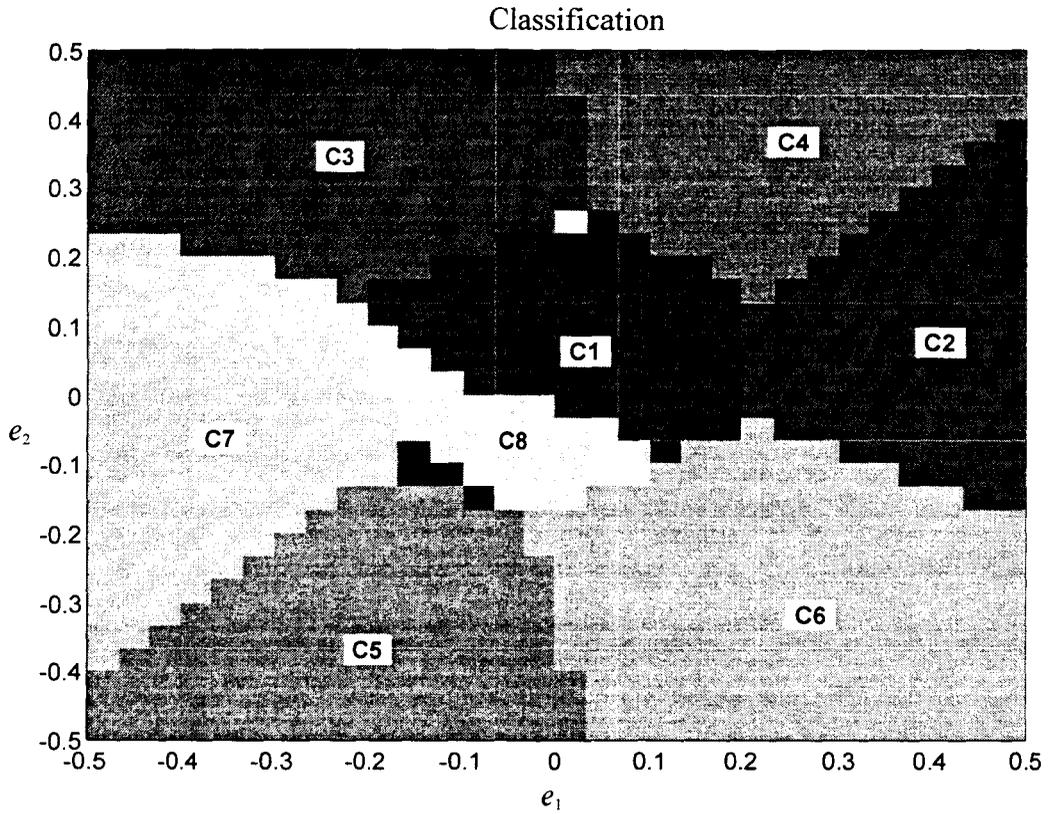


Figure III-13 : Classification neuronale.

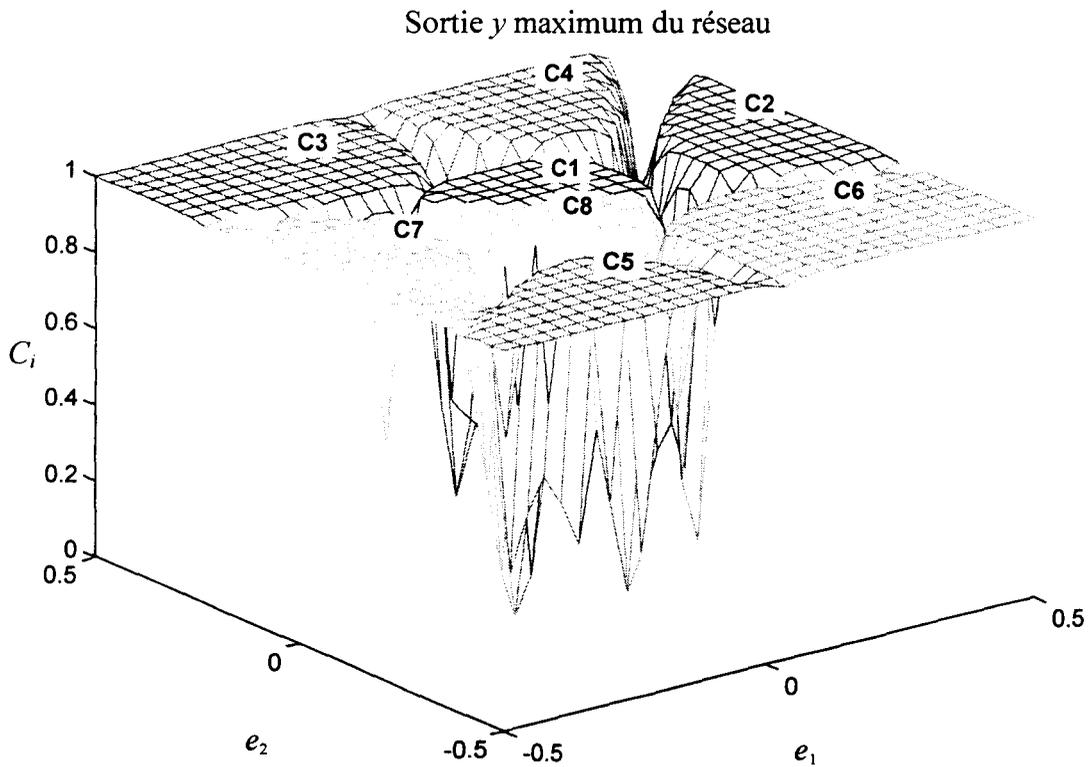


Figure III-14 : Les sorties maximum du réseau.

Ces résultats montrent l'aptitude d'un réseau de neurones à effectuer une bonne classification de  $2^m$  configurations. La loi de commande booléenne fondée sur ce réseau réalise donc de bonnes performances.

## III.6 Conclusion

Dans ce troisième chapitre, nous avons proposé d'utiliser les réseaux de neurones artificiels pour commander les systèmes à entrées booléennes. Une structure de réseau de neurones appropriée avec une méthode d'initialisation adéquate, et un traitement efficace des données ont été développés pour traiter le problème de la commande des systèmes non linéaire à commutation.

L'étape d'initialisation et le filtrage des données donnent un ensemble (de données) d'apprentissage satisfaisant malgré les bruits de mesure. Toutes ces procédures sont nécessaires pour améliorer l'apprentissage du réseau. Ce dernier est basé sur l'algorithme de la rétropropagation du gradient, qui est une méthode relativement faible en coût de calcul, et facile à implanter en temps réel. Il est réalisé en ligne à chaque période d'échantillonnage pour l'ensemble de toutes les données d'apprentissage. L'avantage résultant est d'accroître le taux de convergence des poids et de renforcer la classification. Enfin, celle-ci est utilisée pour déterminer la loi de commande booléenne.

L'efficacité de cette approche est confirmée par les résultats obtenus en simulation sur le moteur synchrone (utilisé comme exemple d'application). Le point fort des réseaux de neurones est de commander le système sans le connaître précisément. L'adaptation en ligne améliore considérablement la robustesse du système par rapport aux variations paramétriques. Cependant, pour notre exemple un régulateur PI exigeant la connaissance du modèle est utilisé. Mais celui-ci n'intervient pas directement sur le système booléen qui est l'objet de notre étude. Il est à noter que le critère de choix entre les configurations 1 et 8 aurait pu être amélioré ; il serait intéressant d'utiliser la méthode de minimisation des commutations présentée au chapitre II.

En conclusion, le réseau de neurones proposé réalise de bonnes performances et permet une commande précise pour les systèmes à commutation.

## **Chapitre IV**



## **IV. Mise en oeuvre et comparaison des diverses commandes présentées**

### **IV.1 Introduction**

Dans ce chapitre, nous allons appliquer les méthodes exposées précédemment pour la commande d'un système à entrées booléennes. Nous comparerons nos algorithmes aux algorithmes classiques sur un modèle composé d'un moteur asynchrone associé à un onduleur de tension. Le modèle est déterminé par l'approche Bond-Graph et une transformation de Park et de Concordia [Caron 1995] pour en réduire la dimension. Le schéma de commande en vitesse du moteur restera identique pour toutes les méthodes, alors que la commande du convertisseur sera associée aux différents algorithmes proposés.

Nous appliquerons en simulation sur notre exemple les algorithmes classiques présentés au chapitre I comme la commande par M.L.I (Modulation de Largeur d'Impulsion) et la commande par modes glissants, ainsi que les commandes proposées dans les chapitres II et III. La comparaison de ces différentes techniques sera réalisée à l'aide de critères spécifiques portant notamment sur l'effet des variations de certains paramètres du système, et sur la robustesse vis-à-vis des bruits de mesure. Il est à noter que pour aucune de ces méthodes, nous n'envisagerons de techniques de réduction du nombre de commutations. Un tableau récapitulatif ainsi qu'une analyse des différents résultats concluront ce chapitre.

### **IV.2 Structure du système non linéaire**

Nous allons définir tout d'abord un modèle Bond-Graph de l'association moteur asynchrone et convertisseur de puissance. Ce modèle sera utilisé pour la simulation du système et pour calculer certaines commandes. Nous verrons également au cours de ce paragraphe le principe de la commande en vitesse mise en oeuvre sur ce système à partir de l'élaboration des flux de référence.

### IV.2.1 Modélisation par Bond-Graph

Nous considérons que le convertisseur de puissance est un onduleur de tension alimenté par une tension continue. Le schéma électrique de l'onduleur est représenté par la figure ci-dessous :

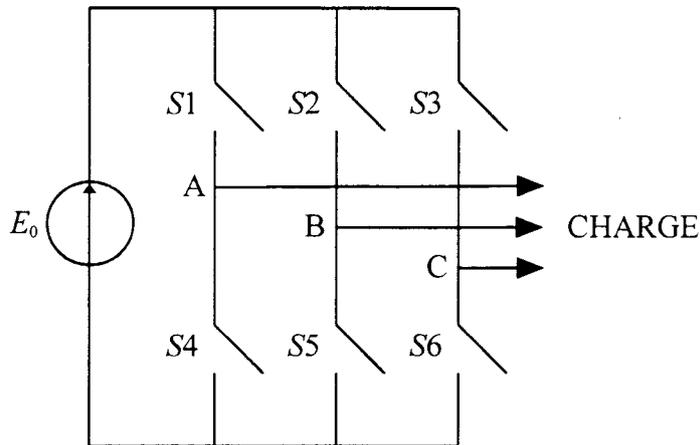


Figure IV-1 : Schéma électrique de l'onduleur.

Les interrupteurs sont représentés par les éléments  $S1, S2, \dots, S6$  dont on peut commander l'état bloquant ou passant.  $E_0$  représente la tension continue alimentant l'onduleur. A présent, nous représentons le modèle bond-graph de l'onduleur (Figure IV-2) :

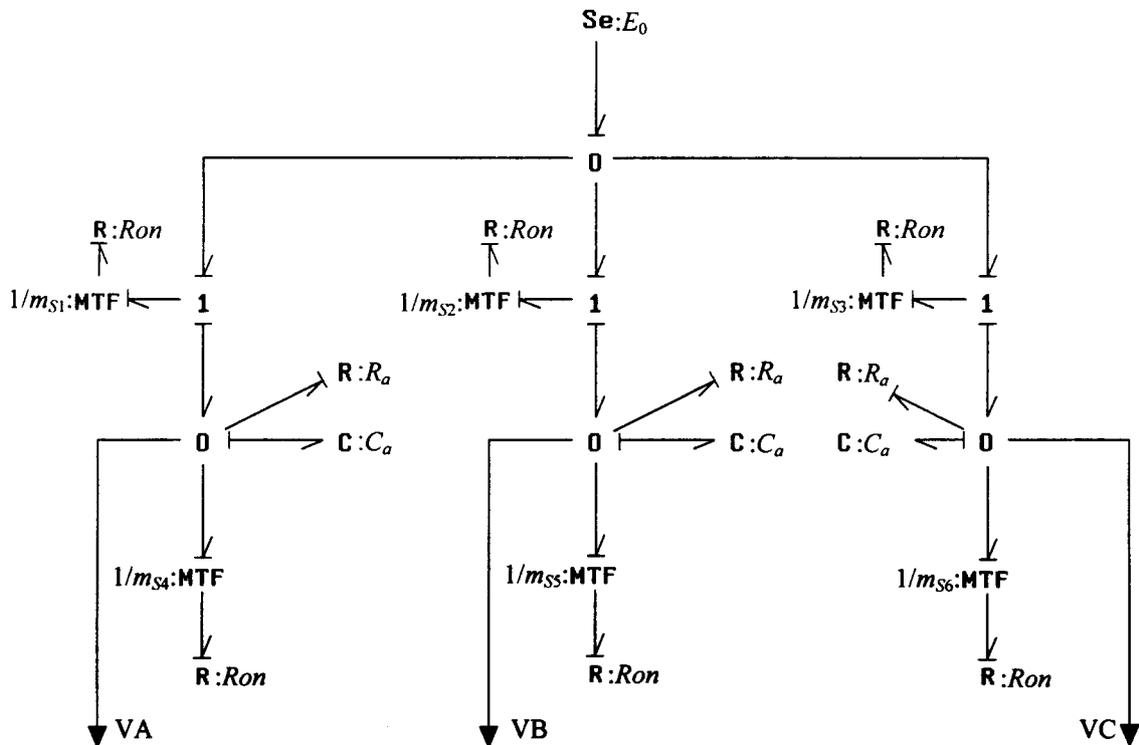


Figure IV-2 : Bond-Graph d'un onduleur de tension.

Les éléments de commutation  $S_1, S_2, \dots, S_6$ , sont considérés non idéaux avec des résistances internes identiques  $R_{on}$ . Nous avons utilisé la modélisation Bond-Graph des interrupteurs définie au chapitre I.

Les transformateurs MTF correspondant aux interrupteurs  $S_1, S_2, \dots, S_6$  sont représentés respectivement par  $\{1/m_1, \dots, 1/m_6\}$  où les  $m_i, i = 1, \dots, 6$  sont des booléens tels que :

$$\begin{cases} m_i = 0 \text{ pour } S_i \text{ bloqué;} \\ m_i = 1 \text{ pour } S_i \text{ passant.} \end{cases} \quad (\text{IV-1})$$

Les blocs  $R_a, C_a$  sont ajoutés en parallèle entre les points A, B, C et la masse pour obtenir une mesure des tensions simples de chaque branche. Leurs valeurs numériques doivent être choisies pour ne pas perturber la dynamique du modèle (petite constante de temps).

Cet onduleur est associé à un moteur asynchrone, que l'on peut représenter par la figure suivante :

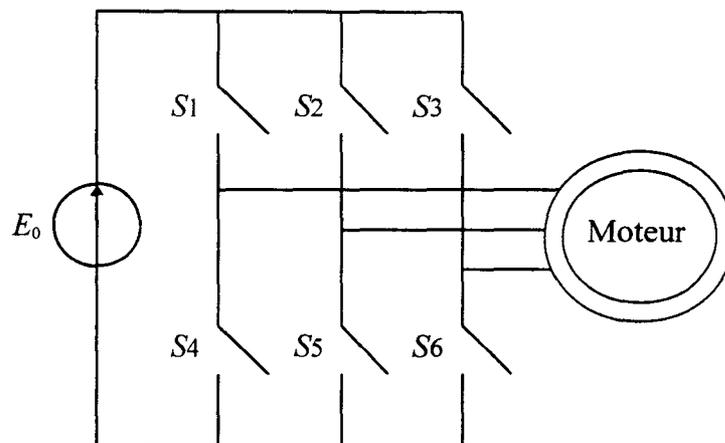


Figure IV-3 : Schéma électrique d'une association onduleur-moteur asynchrone.

La machine asynchrone est composée de trois enroulements au stator et de trois enroulements au rotor (pour plus de détails sur la machine asynchrone, on pourra se référer aux ouvrages suivants : [Caron 1995][Lesenne 1981]).

L'association onduleur-moteur asynchrone que nous étudions peut être modélisée par Bond-Graph [Ducreux 1992], (Figure IV-4) :

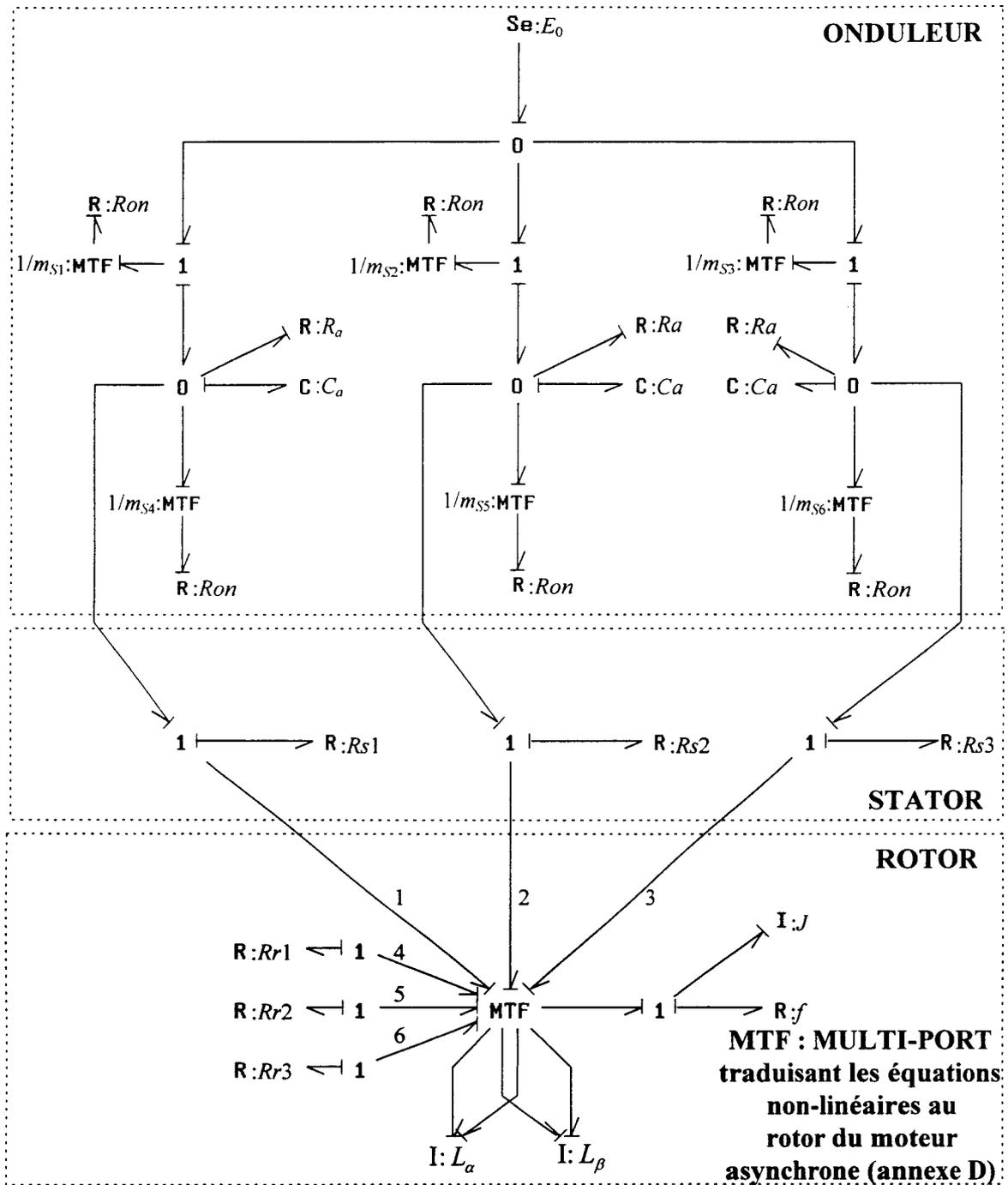


Figure IV-4 : Bond-Graph d'une Association onduleur-moteur asynchrone.

Les  $R_{si}$   $\{i = 1,2,3\}$  représentent les résistances statoriques et nous supposons par la suite que ces grandeurs sont les mêmes sur les trois branches du moteur asynchrone. Le rotor est supposé bobiné lisse avec une seule paire de pôles.  $R_{r1}$ ,  $R_{r2}$ ,  $R_{r3}$  sont les résistances rotoriques.  $J$  et  $f$  représentent respectivement l'inertie et le frottement rotorique. L'élément **MTF** multi-port traduit les relations non linéaires qui lient les tensions et les courants des branches qui l'entourent :  $i = L^{-1} \phi$ , avec  $\phi = (\phi_1, \phi_2, \phi_3, \phi_4, \phi_5, \phi_6)^T$ ,  $i = (i_1, i_2, i_3, i_4, i_5, i_6)^T$ ,

(la matrice  $L$  est définie dans l'annexe D) et la transformation de Park effectuée au stator (annexe D) qui consiste en un passage d'un système triphasé en composantes diphasées.

Le Bond-Graph (Figure IV-4) nous permet d'obtenir les équations reliant les variables de tension et de courant aux différentes jonctions (annexe D). On considère que les constantes de temps liées aux blocs  $R_a$ ,  $C_a$  sont petites et que le système des courants n'a pas de composante homopolaire. D'autre part, on considère que les commandes des interrupteurs sont complémentaires c'est-à-dire que  $S1$  et  $S4$  (resp.  $S2$  et  $S5$ ,  $S3$  et  $S6$ ) ne sont ni bloqués ni passant simultanément. Cette hypothèse entraîne :

$$\begin{cases} m_i^2 + m_{i+3}^2 = 1 \\ i = 1,2,3 \end{cases} \quad (\text{IV-2})$$

Nous avons pris les flux statoriques et rotoriques et la vitesse comme variables d'état pour obtenir un modèle plus simple [Rivera 1996].:

$$\begin{cases} \frac{d\phi_{\alpha s}}{dt} = -R_s \alpha_r \phi_{\alpha s} + R_s \beta \phi_{\alpha r} + u_\alpha \\ \frac{d\phi_{\beta s}}{dt} = -R_s \alpha_r \phi_{\beta s} + R_s \beta \phi_{\beta r} + u_\beta \\ \frac{d\phi_{\alpha r}}{dt} = -R_r \alpha_s \phi_{\alpha r} + R_r \beta \phi_{\alpha s} - p_o \omega \phi_{\beta r} \\ \frac{d\phi_{\beta r}}{dt} = -R_r \alpha_s \phi_{\beta r} + R_r \beta \phi_{\beta s} + p_o \omega \phi_{\alpha r} \\ \frac{d\omega}{dt} = -\frac{f}{J} \omega + \frac{p_o \beta}{J} (\phi_{\beta s} \phi_{\alpha r} - \phi_{\alpha s} \phi_{\beta r}) \end{cases} \quad (\text{IV-3})$$

$\phi_{\alpha s}$  et  $\phi_{\beta s}$  sont les flux statoriques d'axes  $(\alpha - \beta)$ ,  $\phi_{\alpha r}$  et  $\phi_{\beta r}$  sont les flux rotoriques d'axes  $(\alpha - \beta)$ , et  $\omega$  est la vitesse de la machine asynchrone.  $u_\alpha$  et  $u_\beta$  sont les tensions appliquées au stator, elles peuvent être exprimées en fonction des grandeurs booléennes de commutation :

$$\begin{cases} u_\alpha = E_o (2m_1^2 - m_2^2 - m_3^2) \\ u_\beta = E_o (m_2^2 \sqrt{3} - m_3^2 \sqrt{3}) \end{cases} \quad (\text{IV-4})$$

Les grandeurs  $m_1$ ,  $m_2$ ,  $m_3$  représentent les booléens associés aux éléments de commutation  $S1$ ,  $S2$ ,  $S3$ .

En posant  $u = (u_1, u_2, u_3)^T = (m_1^2, m_2^2, m_3^2)^T$ , l'équation (IV-4) s'écrit :

$$\begin{cases} u_\alpha = E_0(2u_1 - u_2 - u_3) \\ u_\beta = E_0(u_2\sqrt{3} - u_3\sqrt{3}) \end{cases} \quad (IV-5)$$

Les autres paramètres sont définis comme :

$$\alpha_s = \frac{L_s}{L_r L_s - M_{sr}^2} \quad \alpha_r = \frac{L_r}{L_r L_s - M_{sr}^2} \quad \beta = \frac{M_{sr}}{L_r L_s - M_{sr}^2} \quad (IV-6)$$

$$L_s = l_s - M_s, \quad L_r = l_r - M_r \quad (IV-7)$$

$$M_{sr} = \frac{3}{2} m_{sr} \quad (IV-8)$$

$R_s$  : Résistance statorique.

$l_s$  : Inductance statorique.

$R_r$  : Résistance rotorique.

$l_r$  : Inductance rotorique.

$M_s$  : Inductance mutuelle statorique.

$M_r$  : Inductance mutuelle rotorique.

$E_0$  : Tension d'alimentation de l'onduleur.

$f, J$  : Frottement et inertie du rotor.

$p_o$  : Nombre de paires de pôles.

$m_{sr}$  : Inductance mutuelle entre stator et rotor.

Les valeurs de ces paramètres identifiés pour cette application sont les suivantes :

$$\begin{cases} R_s : 1.845 \Omega \\ R_r : 1.6 \Omega \\ M_{sr} : 0.227 H \\ L_s : 236 mH \\ L_r : 236.4 mH \\ E_0 : 300 V \\ f : 0.07 N.s \\ J : 0.04 kg.m^2 \\ p_o : 2 \end{cases} \quad (IV-9)$$

## IV.2.2 Principe de commande

### IV.2.2.1 Contrôleur de la vitesse du moteur

Dans cette partie, nous définissons le principe de commande du moteur. Le but est de commander la machine asynchrone pour obtenir une vitesse désirée  $\omega_d$ . Montrons que pour diriger la vitesse du système vers  $\omega_d$ , il est suffisant de commander les flux statoriques de sorte que :

$$\begin{cases} \phi_{\alpha s} ref = \phi_1 \cos(\theta_s) - \phi_2 \sin(\theta_s) \\ \phi_{\beta s} ref = \phi_1 \sin(\theta_s) + \phi_2 \cos(\theta_s) \end{cases} \quad (IV-10)$$

Avec :

$\phi_1$  et  $\phi_2$  les flux à déterminer.

$\theta_s$  l'angle de rotation à imposer au rotor.

Les flux  $\phi_{\alpha s} ref$  et  $\phi_{\beta s} ref$  sont déduits à partir de la matrice de transformation d'angle ou matrice de Concordia. Cette matrice de transformation est utilisée en électrotechnique (Annexe D) pour effectuer un changement de repère d'axe ( $\alpha - \beta$ ) vers le repère tournant ( $d - q$ ). Nous nous basons sur le principe de la commande vectorielle [Delmotte 1997] , [Leonhard 1985] pour construire ces signaux.

Classiquement dans une machine asynchrone il est nécessaire de commander un flux de référence  $\phi_{ref}$  qui magnétise la machine. En conséquence nous considérons ici que le module de flux de magnétisation est fixe et constant  $\phi_1 = \phi_{ref} = 0.8Wb$ .

Pour trouver  $\theta_s$ , il faut reprendre le modèle du système dans le repère  $d-q$  (repère fixe tournant). Prenons l'expression du couple dans l'axe de référence  $d-q$  :

$$C_{em} = p_o \frac{M_{sr}}{L_r} (\phi_{dr} i_{qs} - \phi_{qr} i_{ds}) \quad (IV-11)$$

Comme nous pouvons le voir, le couple (et par conséquent la vitesse) dépend de 4 variables ( $\phi_{dr}; i_{qs}; \phi_{qr}; i_{ds}$ ). La stratégie (en commande vectorielle) est de trouver une expression de la forme  $C = k\phi i$ , qui permet de commander le couple directement en

commandant seulement l'un des deux courants (dans le repère  $d-q$ ). Pour cela, il suffit d'imposer une position angulaire  $\theta_s$  afin que le flux du rotor soit sur l'axe  $d$ . Dès lors, le flux  $\phi_{qr}$  devient nul. Le flux rotorique  $\phi_{dr}$  est fixé par le flux de magnétisation constant  $\phi_{ref}$ . Par conséquent, l'équation du couple devient :

$$C_{em} = p_o \frac{M_{sr}}{L_r} (\phi_{ref} i_{qs}) \quad (IV-12)$$

En utilisant le modèle dans le repère  $d-q$ , et en annulant  $\phi_{qr}$ , on obtient alors (voir annexe D) pour une vitesse désirée  $\omega_d$ , l'équation dérivée de la position angulaire :

$$\frac{d\theta_s}{dt} = p_o \omega_d + \frac{M_{sr}}{T_r \phi_{ref}} i_{qs} \quad (IV-13)$$

avec  $T_r = \frac{L_r}{R_r}$  la constante de temps mécanique. On peut alors remplacer  $i_{qs}$  (équation

IV-12) par:

$$i_{qs} = \frac{L_r}{\phi_{ref} p_o M_{sr}} C_{em}, \quad (IV-14)$$

dans l'équation (IV-13).

$$\frac{d\theta_s}{dt} = p_o \omega_d + \frac{R_r}{p_o \phi_{ref}^2} C_{em} \quad (IV-15)$$

Le second terme de cette expression peut être vu comme la pulsation rotorique de glissement. Le glissement du rotor est donc compensé car il est pris en compte dans les flux statoriques de référence définis par l'équation (IV-10).

D'après l'équation fondamentale de la dynamique, la relation du couple est exprimée par  $C_{em} = J \frac{d\omega}{dt} + f\omega$ , avec  $\omega$  la vitesse de la machine. On obtient :

$$\frac{d\theta_s}{dt} = p_o \omega_d + \frac{R_r}{p_o \phi_{ref}^2} \left( J \frac{d\omega}{dt} + f\omega \right) \quad (IV-16)$$

En définissant le glissement du rotor par  $\omega_{sr} = \frac{R_r}{p_o \phi_{ref}^2} \left( J \frac{d\omega}{dt} + f\omega \right)$ , l'équation ci-dessus devient :

$$\frac{d\theta_s}{dt} = p_o \omega_d + w_{sr} \quad (IV-17)$$

Finalement la valeur de  $\theta_s$  est obtenue par l'expression suivante :

$$\theta_s = \int_0^t (p_o \omega_d + w_{sr}) \quad (IV-18)$$

A présent intéressons nous au flux  $\phi_2$  associé aux flux de références  $\phi_{\alpha,ref}$  et  $\phi_{\beta,ref}$ . Ce flux est proportionnel au couple à imposer. Il sera donc utilisé pour régler la dynamique et le temps de réponse de la machine électrique. Pour ce faire, nous allons contrôler le couple (et donc la vitesse) par une boucle fermée. Nous prendrons simplement un correcteur proportionnel intégral (PI). Il permet d'obtenir pour cette application des performances satisfaisantes sur la réponse en vitesse. Il est à noter que cette boucle de régulation ne fait pas l'objet de notre étude, et de nombreux travaux en électrotechnique (dont [Forgez 1998], [Robyns 1996], [Pierlot 1996]) sont menés pour améliorer les performances de cette boucle de commande. Nous rappelons que notre but ici est de suivre les flux de références.

Les gains de ce PI sont choisis tels que la réponse du système soit rapide avec un faible dépassement pour les conditions nominales. Ce contrôleur réduit l'erreur entre la vitesse de référence  $\omega_d$  et la sortie du processus  $\omega$  en utilisant l'expression suivante :

$$\left\{ \phi_2 = K_1 (\omega_d - \omega) + \frac{1}{T_i} \int_0^t (\omega_d - \omega) dt \right. \quad (IV-19)$$

Pour cette application, le contrôleur PI donne des résultats satisfaisants avec les valeurs suivantes :

$$K_1 = 0.5 \text{ et } T_i = 0.4. \quad (IV-20)$$

La période d'échantillonnage du PI sera choisie selon les dynamiques du système, en conséquence pour le moteur asynchrone nous prendrons pour la simulation une période de  $2.10^{-4}$  secondes.

#### IV.2.2.2 schéma de commande

Les flux du système diphasé  $\phi_{\alpha}$  et  $\phi_{\beta}$  sont mesurés et comparés aux flux de référence  $\phi_{\alpha,ref}$  et  $\phi_{\beta,ref}$ . Les sorties de ces comparateurs sont les entrées  $\varepsilon$  des différentes commandes booléennes puisque l'on désire réduire l'erreur entre les valeurs désirées et les valeurs mesurées. La commande du système est maintenant constituée par deux sous-systèmes de régulation, un pour la vitesse, et l'autre pour régler les flux qui font l'objet de notre étude.

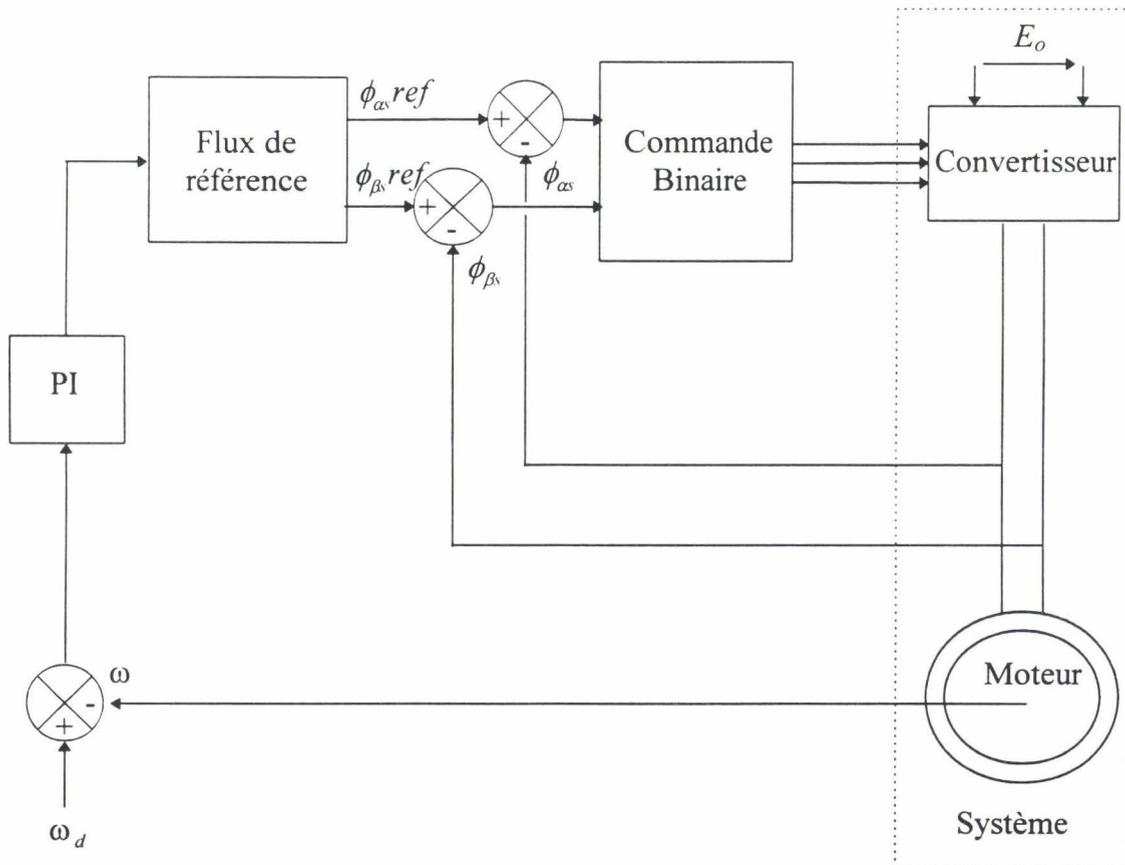


Figure IV-5 : Schéma de commande de l'association machine asynchrone-onduleur

Le principal objectif sur cet exemple est de réaliser la poursuite de trajectoire. Les trajectoires désirées sont imposées par les flux de référence  $\phi_{\alpha,ref}$  et  $\phi_{\beta,ref}$ . Cet objectif est réalisé en utilisant différentes commandes binaires.

## IV.3 Mise en oeuvre des différentes commandes

Nous allons au cours de ce paragraphe mettre en oeuvre en simulation les différentes commandes booléennes que nous avons décrites dans les chapitres précédents. Le logiciel de simulation MATLAB-SIMULINK a été utilisé pour l'étude de la réponse du système électrique. Le système complet comprenant la commande et le système a été écrit en langage C, et avec une période d'échantillonnage de  $1\mu s$  pour la modélisation des dynamiques. Les équations du système complet ont été résolues par la méthode d'EULER pour l'intégration numérique. La période d'échantillonnage pour les différentes commandes booléennes est choisie selon les dynamiques du système. Par conséquent nous prenons une fréquence d'échantillonnage de 10kHz pour assurer de bonne performance en réponse et également pour ne pas trop fatiguer les transistors de commutation. Le convertisseur a été simulé en considérant les interrupteurs non-idéaux et la machine asynchrone a été modélisée par l'équation d'état (IV.3) avec un écart de 20% sur certains paramètres ( $\Delta R_s = \Delta R_r = \Delta L_s = \Delta L_r = 20\%$ ) pour représenter les erreurs de modélisation.

### IV.3.1 Critère de comparaison

Le profil de vitesse désirée présenté Figure IV-6 (a) sera le même pour toutes les commandes considérées. De même, comme nous l'avons vu la vitesse désirée est induite par les flux de référence. Une partie de ces courbes en temporel est montrée Figure IV-6 (b).

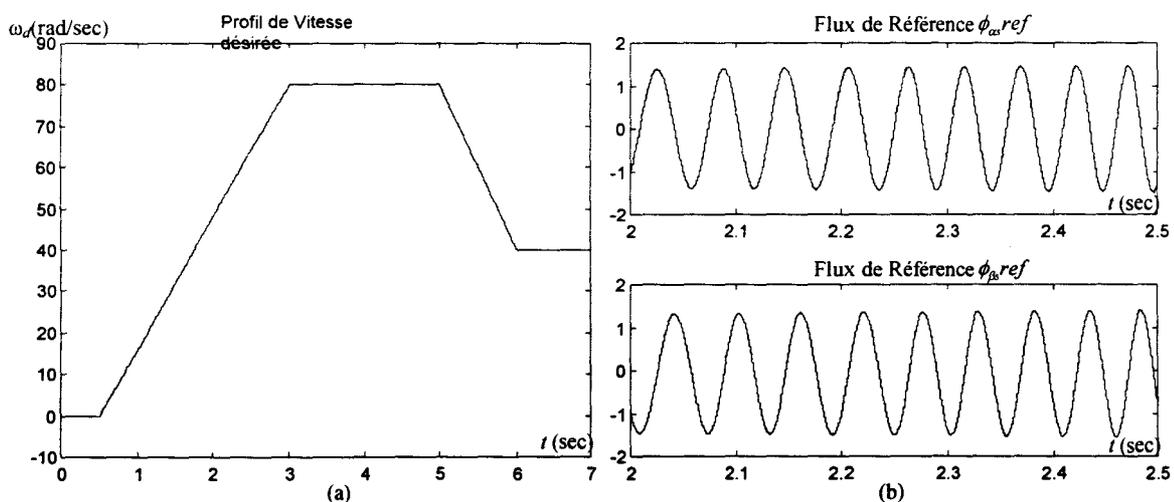


Figure IV-6 : Vitesse désirée (a) et flux de référence (b).

La comparaison et l'évaluation des méthodes se fera sur les indicateurs suivants :

- le calcul du critère défini comme la moyenne des erreurs absolues entre les valeurs réelles et désirées sur la vitesse  $\bar{\varepsilon}_\omega$  en (rad/sec) et sur les flux  $\bar{\varepsilon}_{\phi_\alpha}$ ,  $\bar{\varepsilon}_{\phi_\beta}$  en (Wb);
- l'erreur maximum sur la vitesse  $\varepsilon_{M\omega}$  et sur le flux (Wb)  $\varepsilon_{M\phi_\alpha}$ ,  $\varepsilon_{M\phi_\beta}$ ;
- le nombre de commutations des interrupteurs  $nc$ .

Nous allons maintenant définir plusieurs modes de fonctionnement (identiques pour toutes les méthodes) pour effectuer les mesures. Dans un premier temps, nous ferons fonctionner les méthodes avec une charge  $J$  nominale, c'est à dire sans aucune perturbation externe ou variation paramétrique sur la charge. Ensuite, un test de robustesse par rapport aux variations paramétriques est réalisé. Ce test de robustesse correspond à une variation de la charge de 100%. Pour terminer, nous testerons l'influence des bruits de mesure sur les capteurs; ils seront supposés blancs. Les amplitudes du bruit pour le capteur de flux et le capteur de vitesse sont prises respectivement égales à 0.1 Wb et 0.05 rad/sec.

### IV.3.2 Commande M.L.I

#### IV.3.2.1 Elaboration de l'algorithme

Nous allons ici appliquer la commande par M.L.I développée au chapitre I. Le schéma de commande simplifié devient alors :

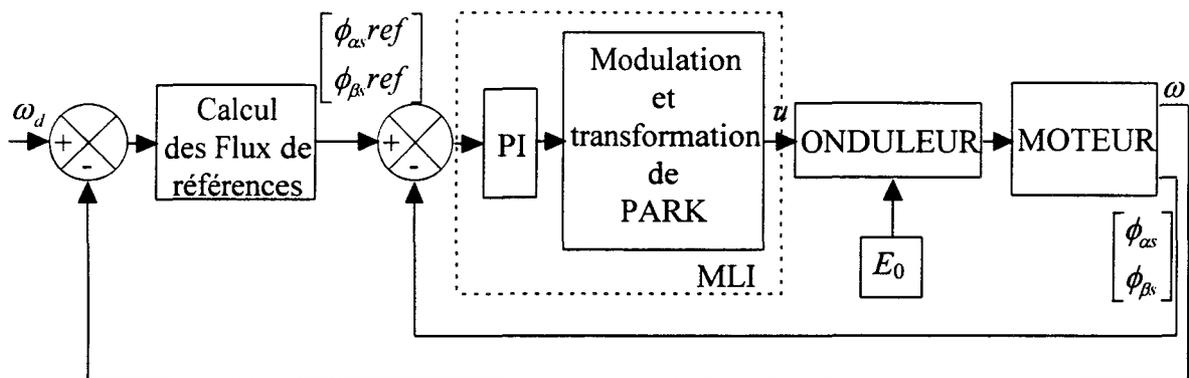


Figure IV-7 : Commande d'un moteur asynchrone par MLI.

Le contrôleur PI est calculé et dimensionné généralement à partir des équations du moteur sans prise en compte de l'onduleur. Ces gains (identiques sur les axes  $\alpha$  et  $\beta$ ) sont respectivement  $K=1845$  et  $T_i=0.002$ . Les équations du PI sont présentées ci-dessous :

$$\begin{cases} \phi_{\alpha s} cor = K(\phi_{\alpha s} ref - \phi_{\alpha s}) + \frac{1}{T_i} \int_0^i (\phi_{\alpha s} ref - \phi_{\alpha s}) dt \\ \phi_{\beta s} cor = K(\phi_{\beta s} ref - \phi_{\beta s}) + \frac{1}{T_i} \int_0^i (\phi_{\beta s} ref - \phi_{\beta s}) dt \end{cases} \quad (IV-21)$$

Le bloc MLI est décrit par la figure suivante :

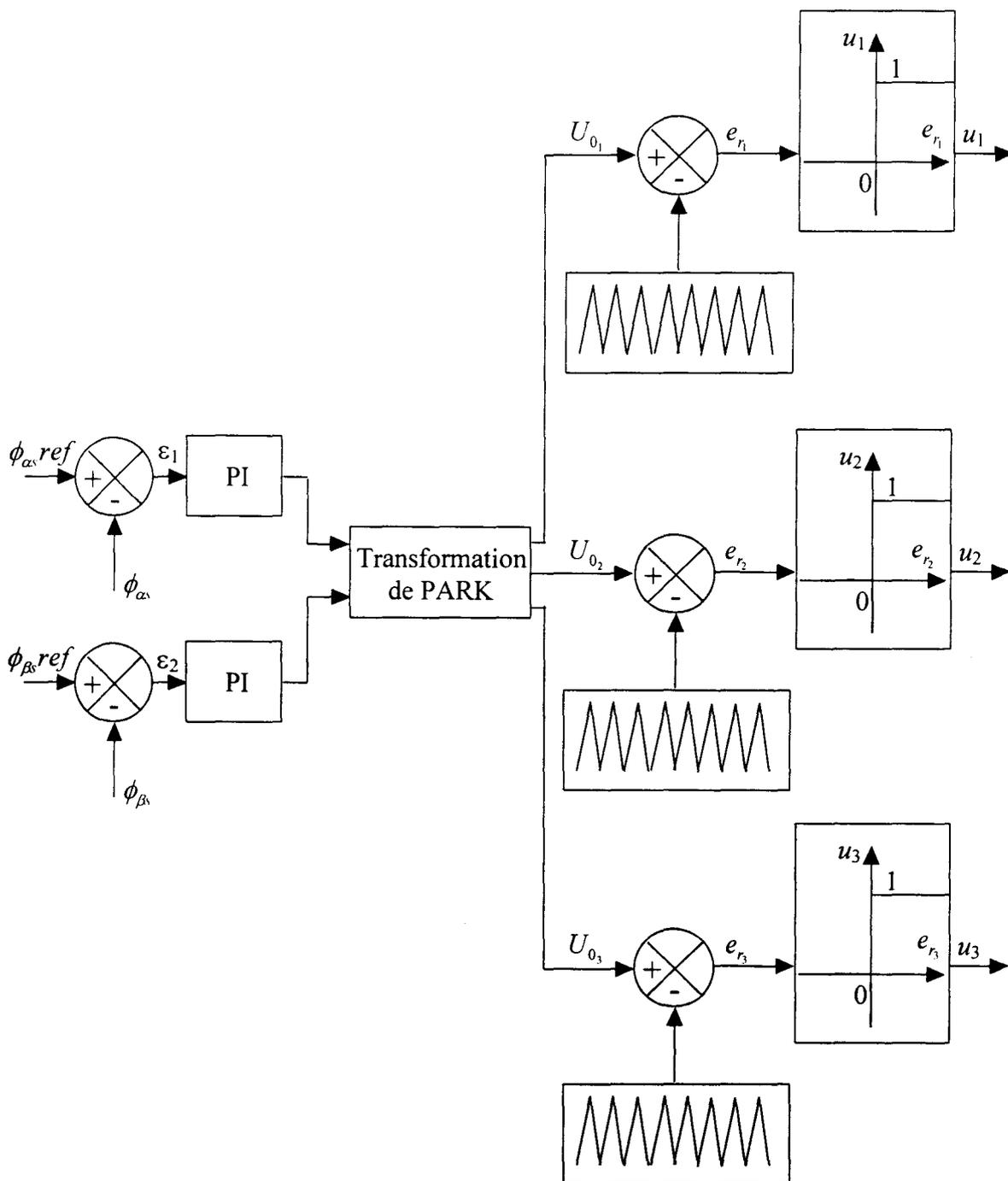


Figure IV-8 : Bloc de commande de l'onduleur MLI.

**IV.3.2.2 Simulation**

Nous obtenons en simulation pour les trois modes de fonctionnement les résultats suivants :

**Fonctionnement à charge nominale ( $J_{nom}$ ) :**

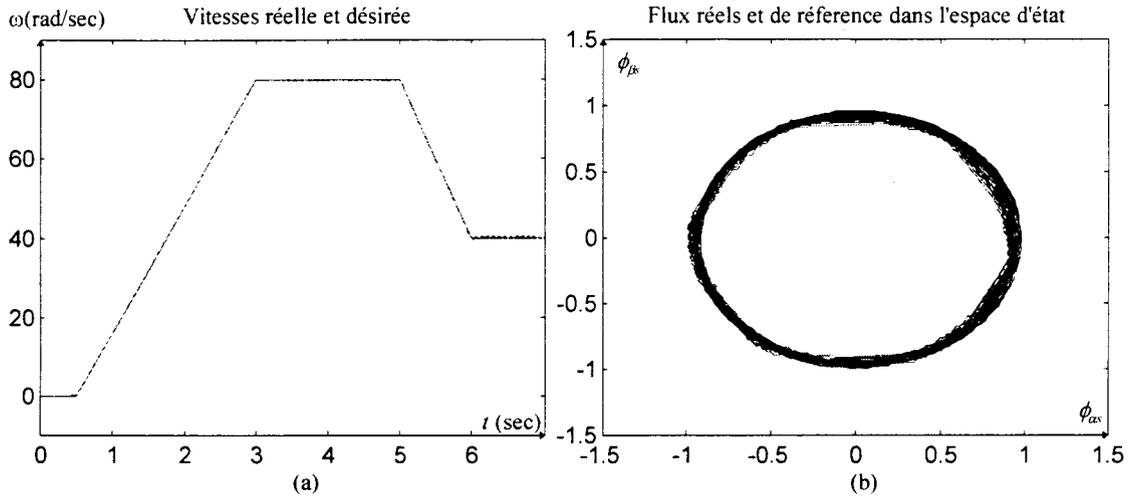


Figure IV-9 : Fonctionnement à charge nominale pour les vitesses réelle et désirée (a) et les flux réels et désirés dans l'espace d'état (b)

**Variation de la charge ( $\Delta J = 100\%$ ) :**

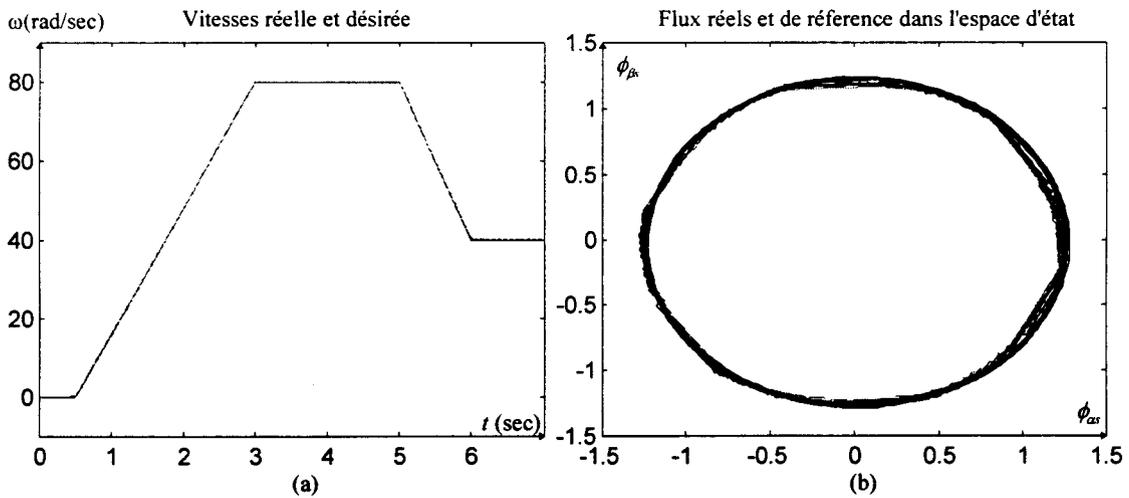


Figure IV-10 : Fonctionnement avec variations sur la charge pour les vitesses réelle et désirée (a) et les flux réels et désirés dans l'espace d'état (b).

**Influence des bruits de mesure :**

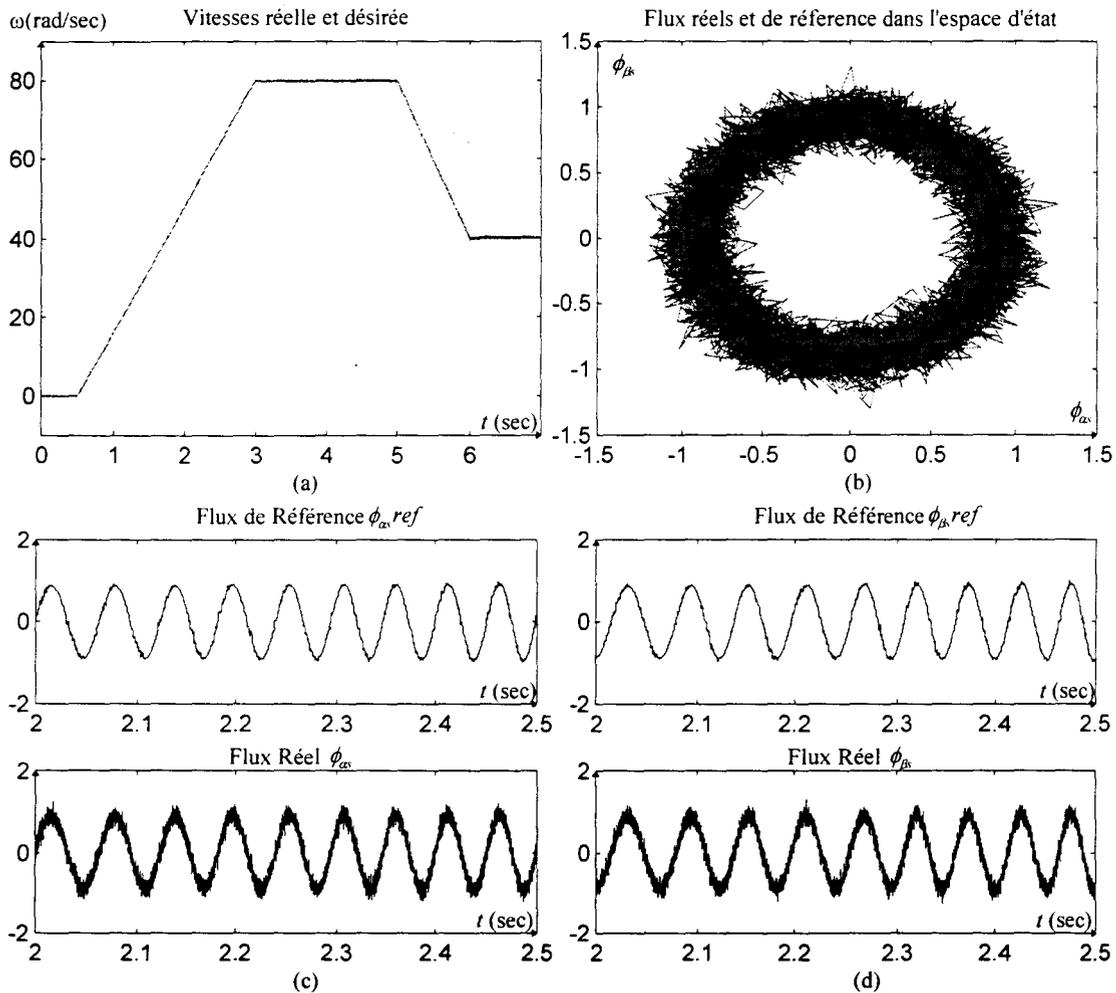


Figure IV-11 : Fonctionnement avec bruits de mesure pour les vitesses réelle et désirée (a) et les flux réels et désirés dans l'espace d'état (b) et en temporel (c)(d).

La simulation des flux dans cette figure s'effectue entre 2 et 2.5 secondes pour avoir une meilleure lisibilité des graphiques. Nous avons choisi de représenter les flux temporellement dans ce cas pour montrer l'influence des bruits de mesure.

**IV.3.3 Commande par Modes Glissants**

**IV.3.3.1 Elaboration de l'algorithme**

Pour cette étude, reprenons le système (équation IV-3) sous la forme d'équation d'état en posant  $x = (x_1, x_2, x_3, x_4, x_5)^T = (\phi_\alpha, \phi_\beta, \phi_{\alpha r}, \phi_{\beta r}, \omega)$  ce qui donne :

$$\dot{x} = \begin{pmatrix} -R_s \alpha_r x_1 + R_s \beta x_3 + u_\alpha \\ -R_s \alpha_r x_2 + R_s \beta x_4 + u_\beta \\ -R_r \alpha_s x_3 + R_r \beta x_1 - p_o \omega x_4 \\ -R_r \alpha_s x_4 + R_r \beta x_2 + p_o \omega x_3 \\ -\frac{f}{J} x_5 + \frac{p_o \beta}{J} (x_2 x_3 - x_1 x_4) \end{pmatrix} \quad (IV-22)$$

La loi de commande est conduite à partir des surfaces de glissement définies par :

$$s(x) = \begin{pmatrix} s_1(x_1) \\ s_2(x_2) \end{pmatrix} = \begin{pmatrix} x_1 - \phi_1 \cos(\theta_s(t)) + \phi_2 \sin(\theta_s(t)) \\ x_2 - \phi_1 \sin(\theta_s(t)) - \phi_2 \cos(\theta_s(t)) \end{pmatrix} \quad (IV-23)$$

Nous pouvons alors définir la commande équivalente (voir chapitre I),  $u_{de} = \begin{pmatrix} u_\alpha \\ u_\beta \end{pmatrix}$  en dérivant  $s$  :

$$\dot{s}(x) = \begin{pmatrix} \dot{s}_1(x_1) \\ \dot{s}_2(x_2) \end{pmatrix} = \begin{pmatrix} -R_s \alpha_r x_1 + R_s \beta x_3 + \phi_1 \dot{\theta}_s(t) \sin(\theta_s(t)) + \phi_2 \dot{\theta}_s(t) \cos(\theta_s(t)) \\ -R_s \alpha_r x_2 + R_s \beta x_4 - \phi_1 \dot{\theta}_s(t) \cos(\theta_s(t)) + \phi_2 \dot{\theta}_s(t) \sin(\theta_s(t)) \end{pmatrix} + u_{de} \quad (IV-24)$$

soit sur l'intersection des surfaces de glissement :

$$u_{de} = \begin{pmatrix} -R_s \beta x_3 - \sin(\theta_s(t))(R_s \alpha_r \phi_2 + \phi_1 \dot{\theta}_s(t)) + \cos(\theta_s(t))(R_s \alpha_r \phi_1 - \phi_2 \dot{\theta}_s(t)) \\ -R_s \beta x_4 + \sin(\theta_s(t))(R_s \alpha_r \phi_1 - \phi_2 \dot{\theta}_s(t)) + \cos(\theta_s(t))(R_s \alpha_r \phi_2 + \phi_1 \dot{\theta}_s(t)) \end{pmatrix} \quad (IV-25)$$

Le système est donc décrit sur l'intersection des surfaces de glissement par :

$$\dot{x} = \begin{pmatrix} -R_s \alpha_r x_1 - \sin(\theta_s(t))(R_s \alpha_r \phi_2 + \phi_1 \dot{\theta}_s(t)) + \cos(\theta_s(t))(R_s \alpha_r \phi_1 - \phi_2 \dot{\theta}_s(t)) \\ -R_s \alpha_r x_2 + \sin(\theta_s(t))(R_s \alpha_r \phi_1 - \phi_2 \dot{\theta}_s(t)) + \cos(\theta_s(t))(R_s \alpha_r \phi_2 + \phi_1 \dot{\theta}_s(t)) \\ -R_r \alpha_s x_3 + R_r \beta x_1 - p_o \omega x_4 \\ -R_r \alpha_s x_4 + R_r \beta x_2 + p_o \omega x_3 \\ -\frac{f}{J} x_5 + \frac{p_o \beta}{J} (x_2 x_3 - x_1 x_4) \end{pmatrix} \quad (IV-26)$$

Nous définissons alors la logique de commutation suivante :

$$\begin{cases} u_\alpha = E_0 \text{ si } s_1(x_1) < 0 \\ u_\alpha = -E_0 \text{ si } s_1(x_1) > 0 \\ u_\beta = E_0\sqrt{3} \text{ si } s_2(x_2) < 0 \\ u_\beta = -E_0\sqrt{3} \text{ si } s_2(x_2) > 0 \end{cases} \quad (IV-27)$$

La démonstration de l'existence du régime de glissement avec cette logique de commutation est donnée en annexe E. Cette logique nous permet donc de déduire les variables Booléennes et de les représenter pour chaque valeur de la surface de glissement dans le tableau suivant :

	$s_1 < 0 \ \& \ s_2 < 0$	$s_1 < 0 \ \& \ s_2 > 0$	$s_1 > 0 \ \& \ s_2 < 0$	$s_1 > 0 \ \& \ s_2 > 0$
$u_1$	1	1	0	0
$u_2$	1	0	1	0
$u_3$	0	1	0	1

Tableau IV-1 : Commandes pour différentes valeurs des surfaces.

Le schéma pour cette commande est représenté ci-dessous :

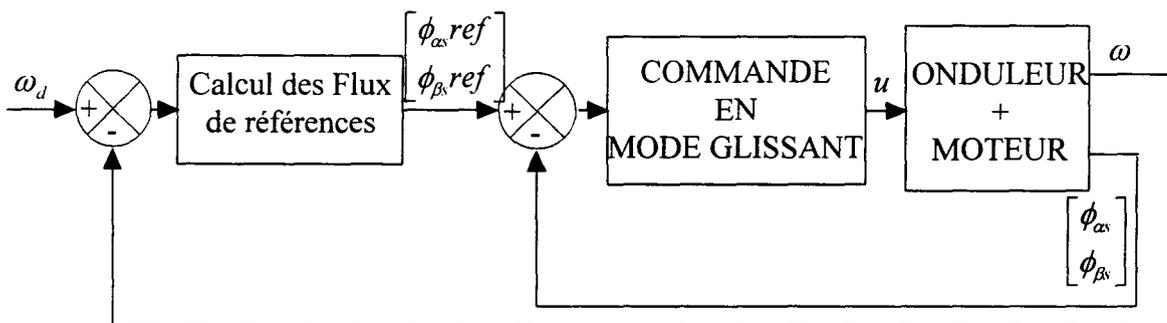


Figure IV-12 : Commande par mode glissant.

### IV.3.3.2 Simulation

Nous obtenons en simulation pour les trois modes de fonctionnement les résultats suivants :

**Fonctionnement à charge nominale ( $J_{nom}$ ) :**

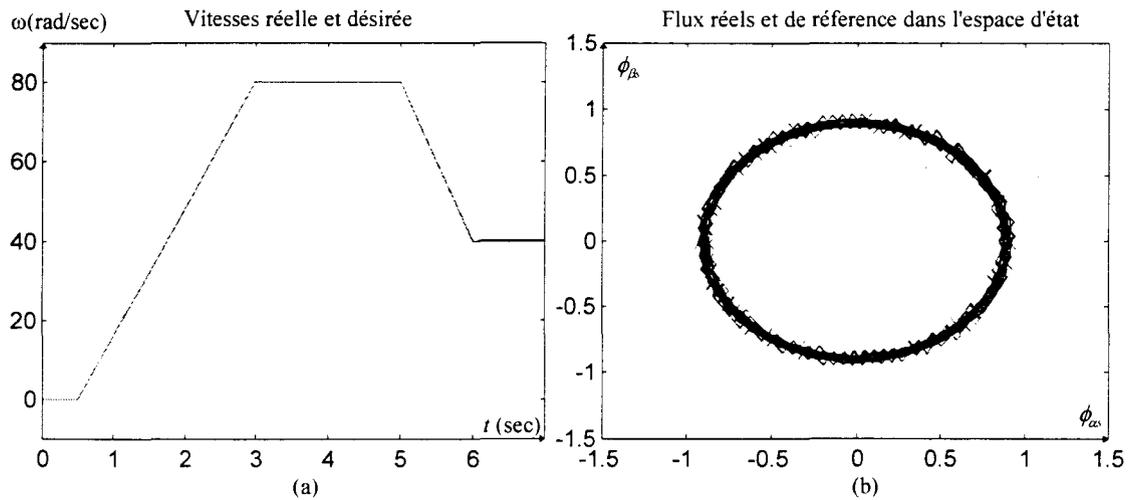


Figure IV-13 : Fonctionnement à charge nominale pour les vitesses réelle et désirée (a) et les flux réels et désirés dans l'espace d'état (b).

**Variation de la charge ( $\Delta J = 100\%$ ) :**

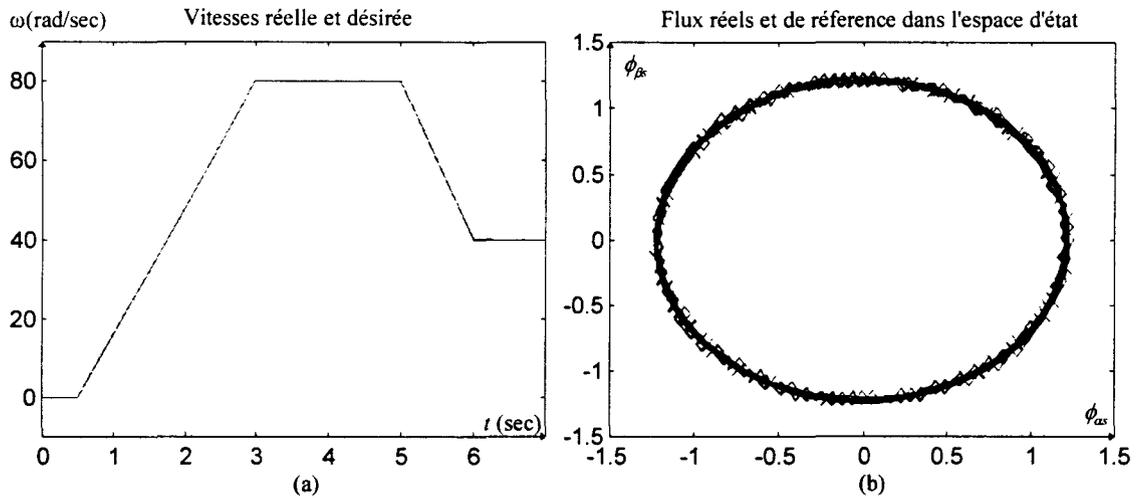


Figure IV-14 : Fonctionnement avec variations sur la charge pour les vitesses réelle et désirée (a) et les flux réels et désirés dans l'espace d'état (b).

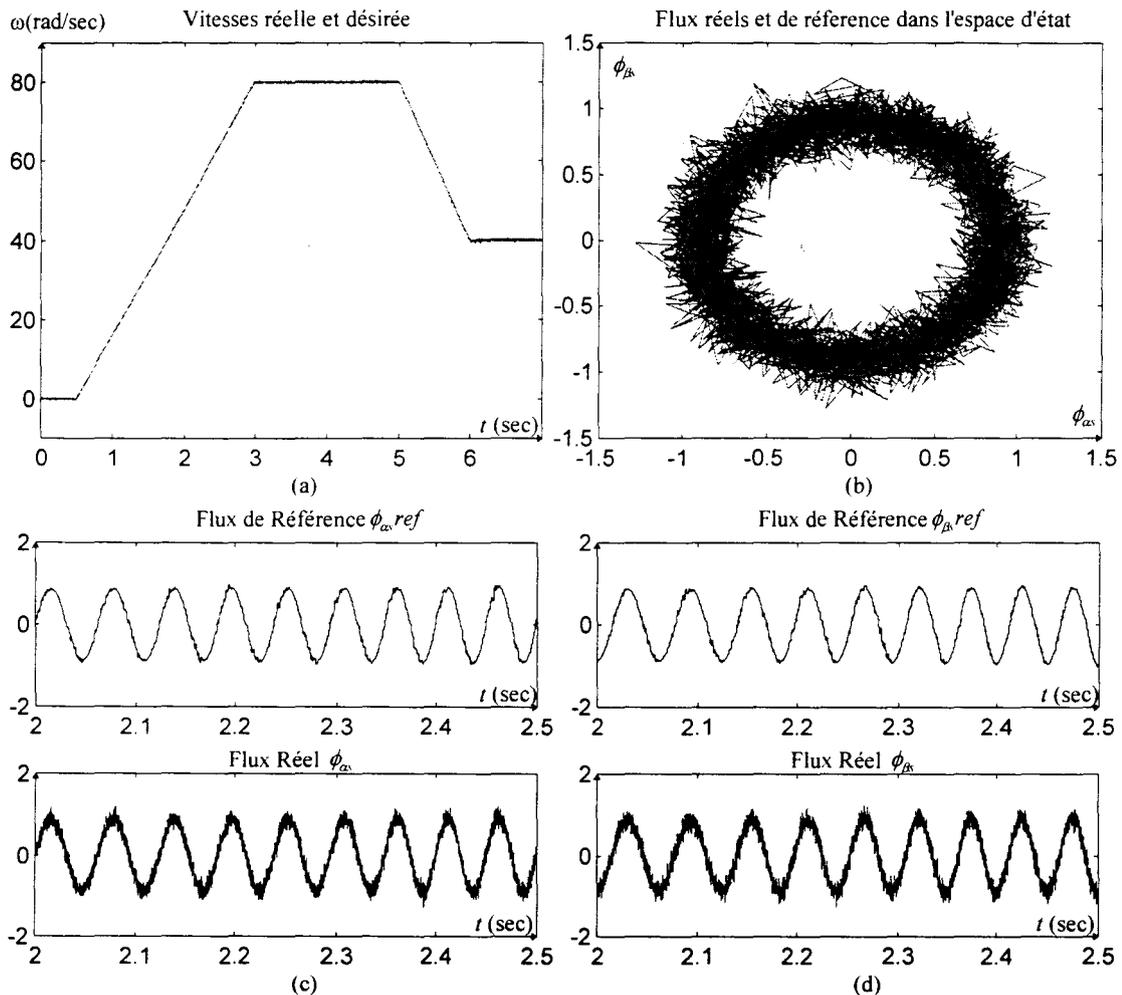
**Influence des bruits de mesure :**

Figure IV-15 : Fonctionnement avec bruits de mesure pour les vitesses réelle et désirée (a) et les flux réels et désirés dans l'espace d'état (b) et en temporel (c)(d).

**IV.3.4 Commande Booléenne****IV.3.4.1 Elaboration de l'algorithme**

L'algorithme ici est basé sur la commande développée au chapitre II. Il est réalisé sans minimisation des commutations, et ceci afin de comparer toutes les autres commandes sur le nombre de commutations. Le schéma de réalisation de cette commande est représenté ci-dessous.

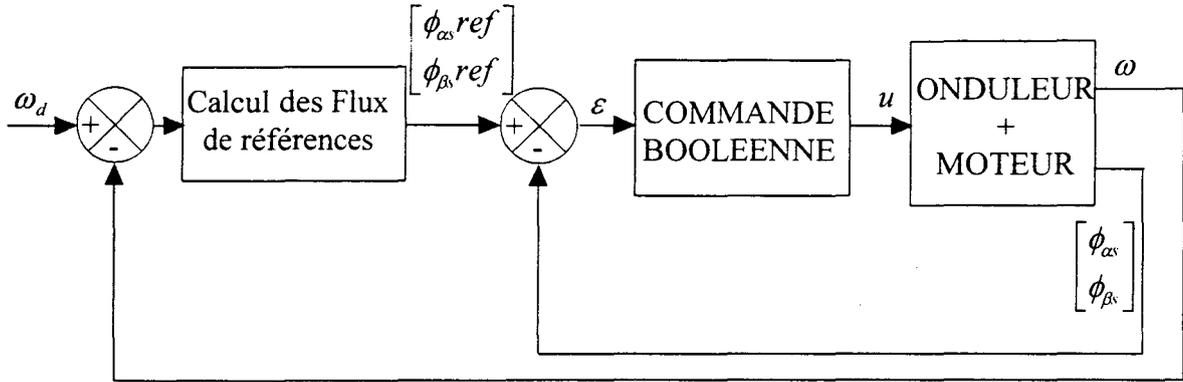


Figure IV-16: Commande Booléenne.

Notons pour ce schéma  $\phi_s ref = \begin{bmatrix} \phi_{\alpha s} ref \\ \phi_{\beta s} ref \end{bmatrix}$ , et  $\phi_s = \begin{bmatrix} \phi_{\alpha s} \\ \phi_{\beta s} \end{bmatrix}$  les vecteurs des flux désirés et mesurés respectivement.

La comparaison des angles entre vecteurs (produit scalaire  $\langle \varepsilon, (\dot{\phi}_s)_i \rangle$ ) se fera entre les dynamiques possibles du système (équation IV-3) des flux statoriques pour chaque configuration  $i$  et le vecteur erreur  $\varepsilon$ , défini par la relation suivante :

$$\varepsilon = \phi_s ref - \phi_s \tag{IV-29}$$

Pour le cas du moteur asynchrone, l'ensemble des configurations de commande est :

$$\{Config_i(u) / i = 1 \dots 8\} = \left\{ \begin{pmatrix} (0) & (1) & (0) & (1) & (0) & (1) & (0) & (1) \\ (0) & (0) & (1) & (1) & (0) & (0) & (1) & (1) \\ (0) & (0) & (0) & (0) & (1) & (1) & (1) & (1) \end{pmatrix} \right\} \tag{IV-30}$$

L'algorithme consiste à trouver le "bon" vecteur qui dirige  $\phi_s$  vers le point désiré  $\phi_s ref$  défini dans l'espace d'état. Pour cela, maximisons à chaque période d'échantillonnage le critère suivant (chapitre II) :

$$\max \left\{ \frac{\langle \varepsilon, (\dot{\phi}_s)_i \rangle}{\|\varepsilon\| \|(\dot{\phi}_s)_i\|} \right\} \tag{IV-31}$$

La valeur de la configuration  $i$ , qui maximise le critère, nous donnera la commande  $u$  à appliquer au moteur asynchrone.

#### IV.3.4.2 Simulation

Nous obtenons pour la simulation les résultats suivants :

**Fonctionnement à charge nominale ( $J_{nom}$ ) :**

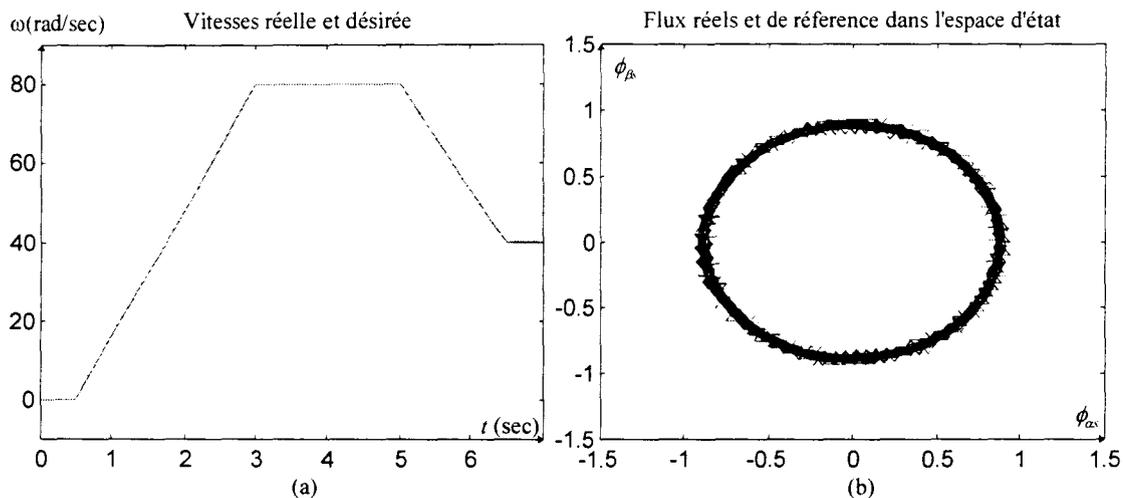


Figure IV-17 : Fonctionnement à charge nominale pour les vitesses réelle et désirée (a) et les flux réels et désirés dans l'espace d'état (b).

**Variation de la charge ( $\Delta J = 100\%$ ) :**

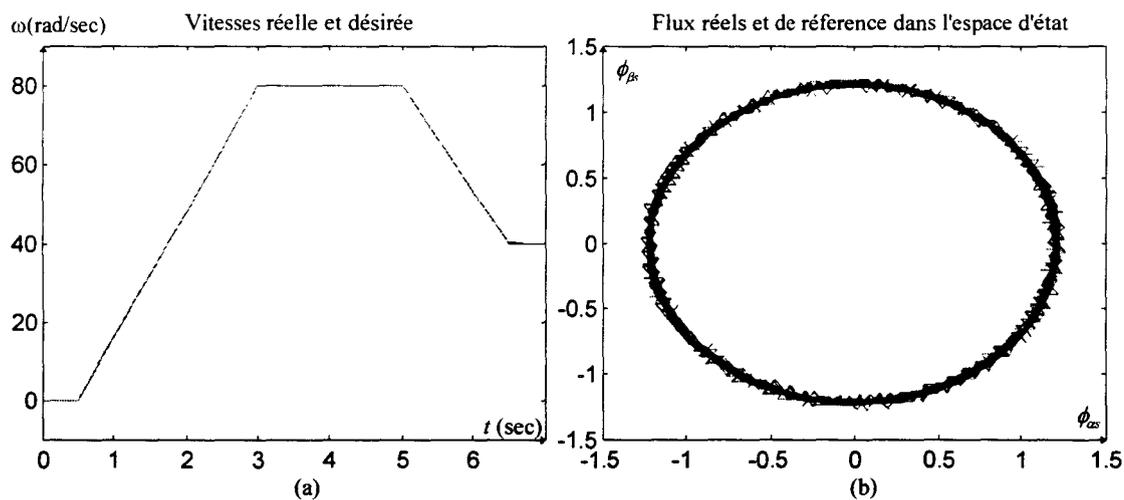


Figure IV-18 : Fonctionnement avec variations sur la charge pour les vitesses réelle et désirée (a) et les flux réels et désirés dans l'espace d'état (b) .

**Influence des bruits de mesure :**

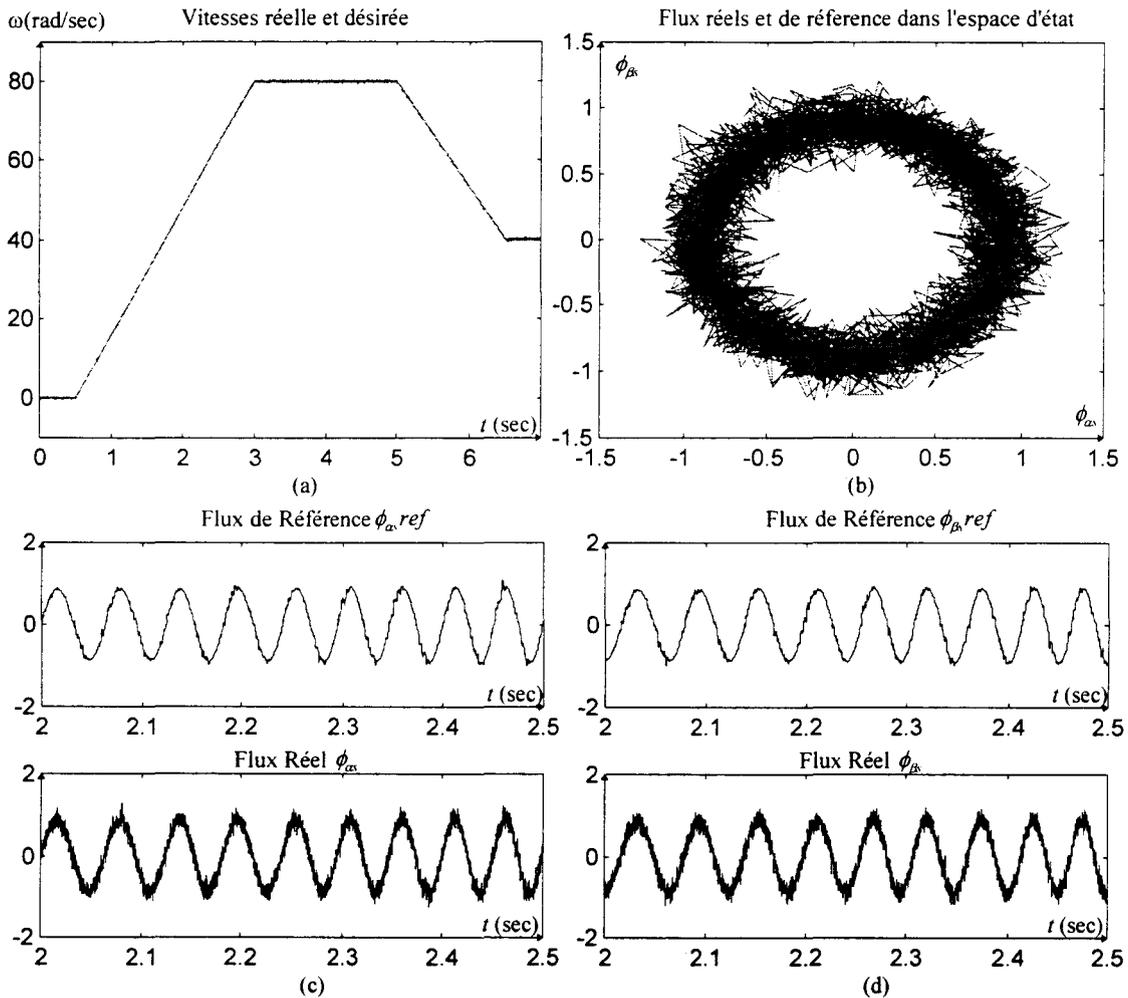


Figure IV-19 : Fonctionnement avec bruits de mesure pour les vitesses réelle et désirée (a) et les flux réels et désirés dans l'espace d'état (b) et en temporel (c)(d).

**IV.3.5 Commande neuronale**

**IV.3.5.1 Elaboration de l'algorithme**

L'élaboration de cette commande repose sur le principe développé dans le chapitre III. Les flux de référence  $\phi_{\alpha s,ref}$  et  $\phi_{\beta s,ref}$  sont comparés aux flux du moteur  $\phi_{\alpha s}$  et  $\phi_{\beta s}$ . Les sorties de ces comparateurs sont les entrées de commande du réseau de neurones, et les flux  $\phi_{\alpha s}$  et  $\phi_{\beta s}$  sont les entrées pour l'apprentissage de ce réseau. Le processus de commande est représenté par le schéma Figure IV-20 :

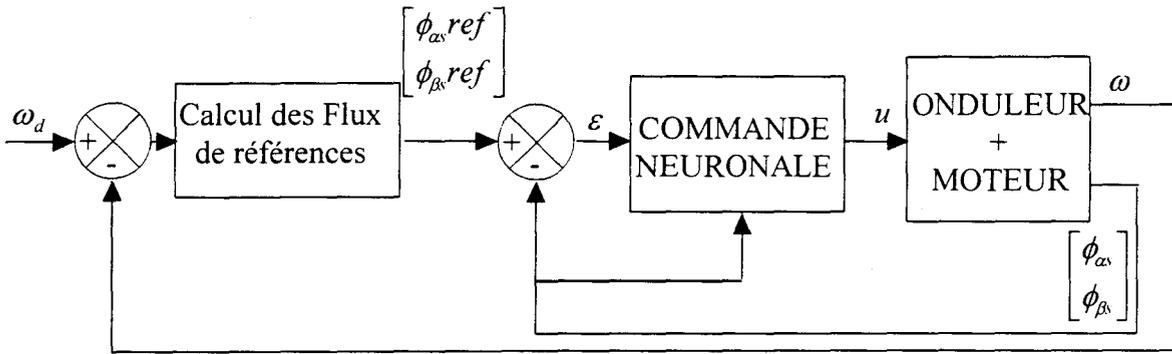


Figure IV-20 : Commande Neuronale.

La structure du réseau de neurones utilisé est une structure 2-8-8 (deux entrées  $e_1$  et  $e_2$ , huit neurones dans la couche cachée, et huit neurones  $y_i$  avec  $i \in \{1, \dots, 2^m\}$  dans la couche de sortie). Chaque sortie correspondant à une classe, et donc à une configuration associée (définition identique pour le moteur synchrone du chapitre III). L'ensemble de ces configurations est défini par :

$$\{Config_i(u) / i = 1 \dots 8\} = \left\{ \begin{pmatrix} 0 & 1 & 0 & 1 & 0 & 1 & 0 & 1 \\ 0 & 0 & 1 & 1 & 0 & 0 & 1 & 1 \\ 0 & 0 & 0 & 0 & 1 & 1 & 1 & 1 \end{pmatrix} \right\} \quad (IV-32)$$

Nous rappelons que le principe d'apprentissage est basé sur la classification des variations d'état des  $2^m$  configurations.

Le comportement moteur pour les configurations 1 et 8 sont les mêmes, ceci étant dû à la structure du convertisseur (Figure IV-1). Par conséquent, cela signifie que les données des classes 1 et 8 se chevauchent. En général, les classes qui se chevauchent ou qui entourent les autres ne peuvent être séparées. Par conséquent, les classes 1 et 8 doivent être considérées identiques. Les vecteurs désirés pour l'apprentissage  $(y^d)_1$  et  $(y^d)_8$  sont alors choisis de la manière suivante:

$$(y^d)_1 = (y^d)_8 = \begin{bmatrix} y_1^d \\ 0 \\ \vdots \\ 0 \\ y_8^d \end{bmatrix} \text{ avec } y_1^d = y_8^d = 1$$

Cette nouvelle définition du vecteur cible n'a pas d'influence sur l'algorithme de commande neuronale puisque les configurations 1 et 8 sont équivalentes par rapport au comportement moteur.

Le réseau de neurones est entraîné à apprendre avec l'algorithme d'apprentissage présenté chapitre III, et les vecteurs cibles ci-dessus. Les taux d'apprentissage  $\eta$  et  $\eta_b$  sont choisis égaux à 0.2 et 0.1, et assurent une adaptation rapide et stable des poids.

#### IV.3.5.2 Simulation

Le processus d'apprentissage du réseau de neurones réalisé la première fois pour faire évoluer les poids initiaux choisis aléatoirement vers des poids satisfaisants se fait en 0.56 secondes. Nous obtenons alors avec ces poids et pour les trois modes de fonctionnement les résultats de simulation suivants :

##### Fonctionnement à charge nominale ( $J_{nom}$ ) :

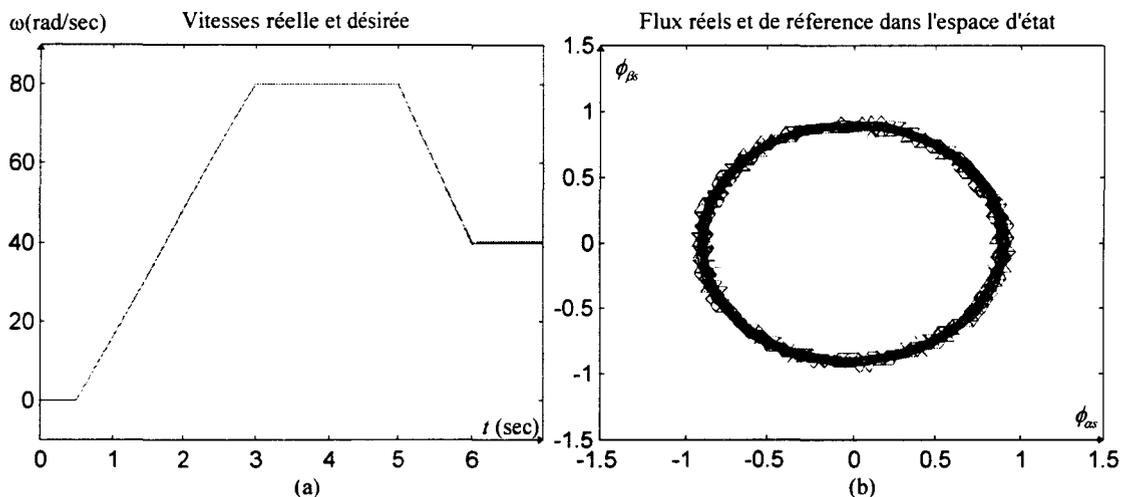


Figure IV-21 : Fonctionnement à charge nominale pour les vitesses réelle et désirée (a) et les flux réels et désirés dans l'espace d'état (b).

##### Variation de la charge ( $\Delta J = 100\%$ ) :

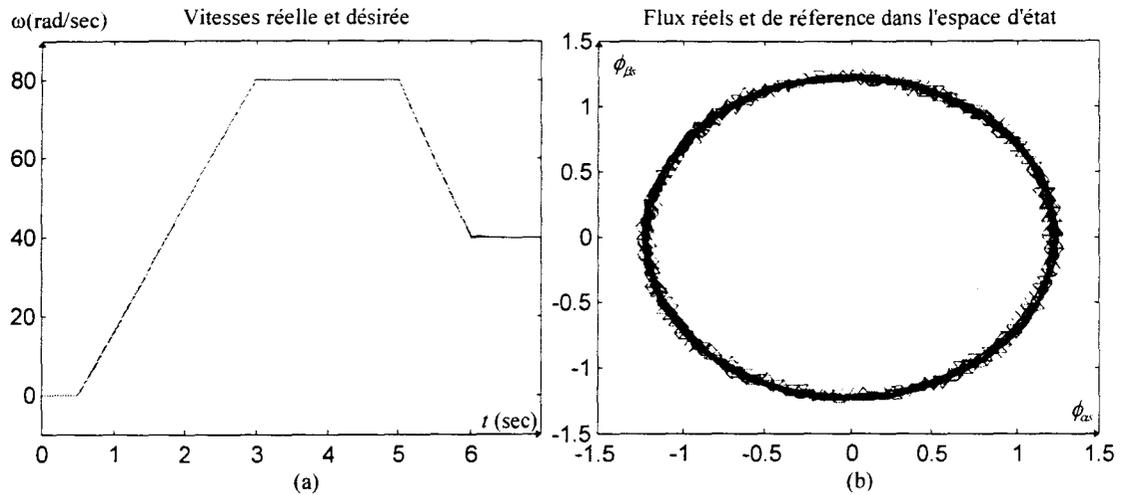


Figure IV-22 : Fonctionnement avec variations sur la charge pour les vitesses réelle et désirée (a) et les flux réels et désirés dans l'espace d'état (b).

**Influence des bruits de mesure :**

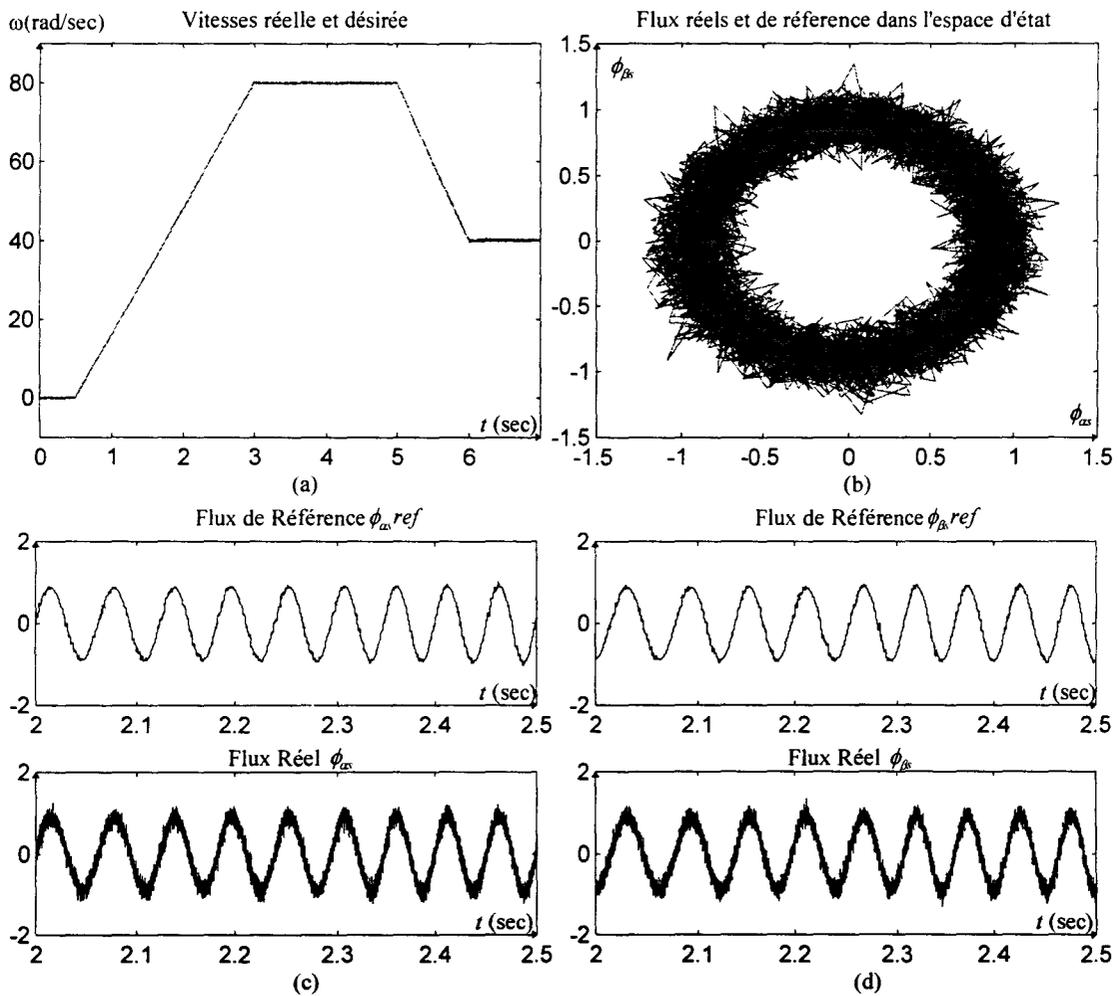


Figure IV-23 : Fonctionnement avec bruits de mesure pour les vitesses réelle et désirée (a) et les flux réels et désirés dans l'espace d'état (b) et en temporel (c)(d).

L'analyse du réseau est réalisée en traçant les classes par rapport aux variables d'entrée  $e$ . Dans le centre de la Figure IV-24, nous notons que les classes 1 et 8 se chevauchent comme nous l'avons décrit précédemment. Les 6 autres classes sont bien séparées et, comme le montre la Figure IV-25, les limites des classes apparaissent clairement. En effet les sorties  $y_i$  du réseaux varient rapidement entre les niveaux d'activation bas 0 et haut 1.

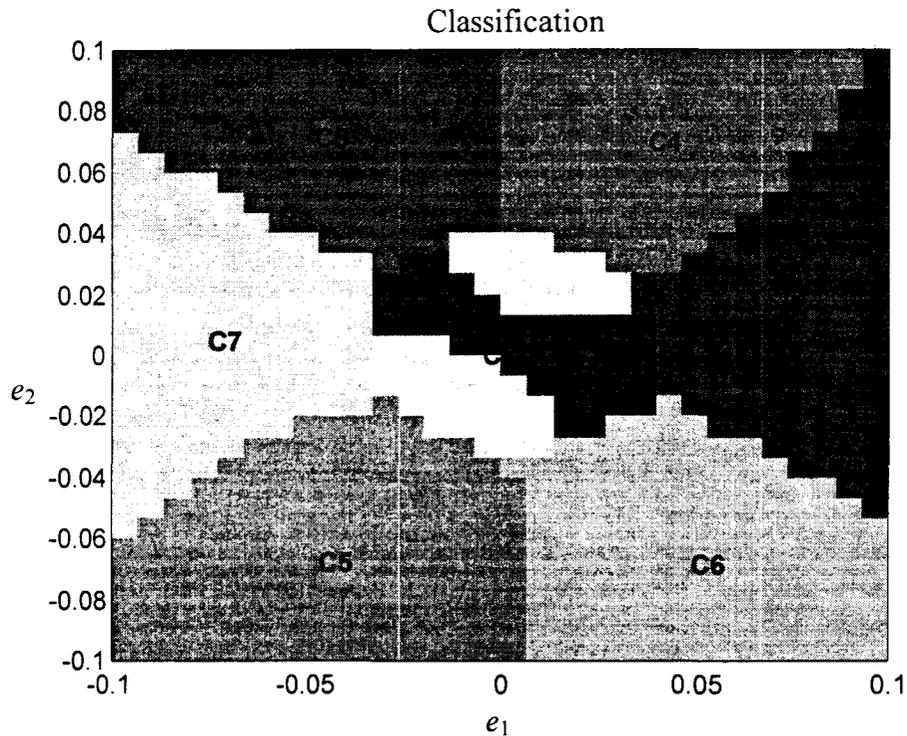


Figure IV-24 : Classification Neuronale (une image à un instant donné).

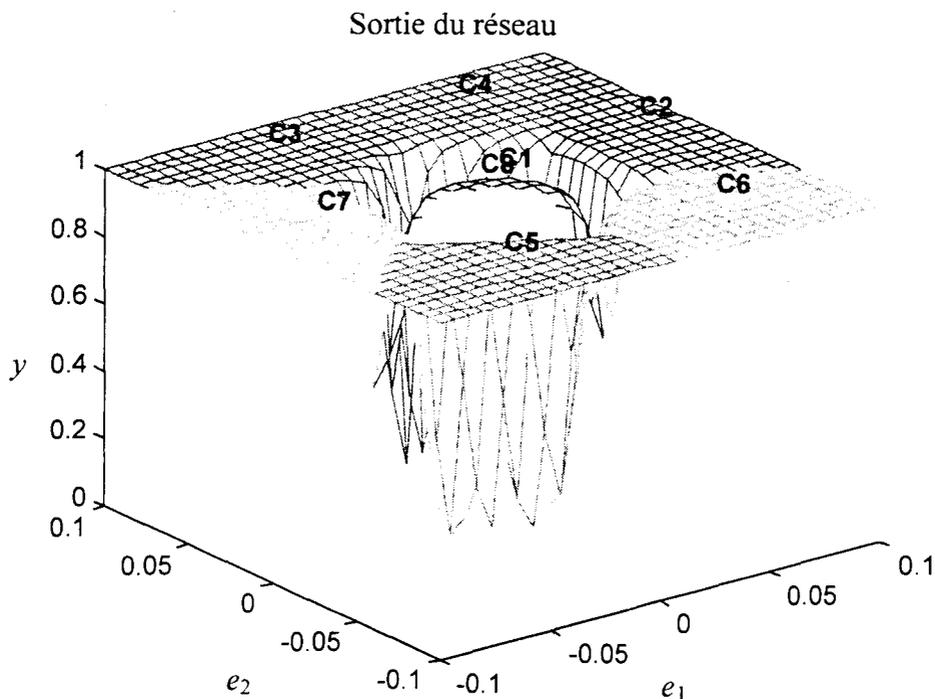


Figure IV-25 : Sortie du réseau neuronal.

## IV.4 Comparaisons

Au cours de ce paragraphe nous allons faire quelques commentaires sur les différentes méthodes qui ont été implantées en simulation.

### IV.4.1 comparaison des structures de commande

Les structures de commande utilisées dans les diverses méthodes comportent des différences. En effet, la structure de la méthode MLI calcule un contrôleur PI ou une autre commande sans prendre en compte le convertisseur dans le modèle. La commande continue est alors modulée par un signal triangulaire de haute fréquence. Par conséquent les instants de commutation ne sont pas contrôlés. L'avantage d'utiliser les autres méthodes ( commande à mode glissant, commande booléenne et commande neuronale) est que la commutation est contrôlée par la période d'échantillonnage, celle-ci étant fixe. Pour ce qui concerne la commande à mode glissant, la commande est binaire mais elle n'est pas définie en 0 et 1, et doit être déterminée à l'aide d'une table (voir Tableau IV-1). Le calcul de la commande booléenne utilise le modèle global, en intégrant le modèle du convertisseur. Enfin, la commande neuronale ne demande pas la connaissance du modèle pour élaborer la commande, les signaux délivrés par cette méthode sont booléens.

### IV.4.2 Comparaisons des indicateurs

Nous présentons sous forme de tableaux quelques éléments de comparaison portant sur les mesures d'erreurs définies au paragraphe IV.3.1 (l'erreur moyenne absolue et l'erreur maximum sur la vitesse notées respectivement  $\bar{\varepsilon}_\omega$  en (rad/sec)  $\varepsilon_{M\omega}$ ; l'erreur moyenne absolue sur les flux  $\bar{\varepsilon}_{\phi_\alpha}$ ,  $\bar{\varepsilon}_{\phi_\beta}$  en (Wb); l'erreur maximum sur le flux (Wb)  $\varepsilon_{M\phi_\alpha}$ ,  $\varepsilon_{M\phi_\beta}$ ; le nombre de commutations  $nc$ ).

Ces erreurs doivent être interprétées en fonction des ordres de grandeur des variables concernées : soit  $\omega_d$  variant de 0 à 80 rad/sec, et  $\left[ \begin{matrix} \phi_{\alpha,ref} \\ \phi_{\beta,ref} \end{matrix} \right]$  entre -1.5 et 1.5 Wb (Figure IV-6)

Mesures	Charge Nominale		Variation de la charge		Bruits de Mesure	
	$\bar{\varepsilon}_\omega$ (rad/sec)	$\varepsilon_{M\omega}$ (rad/sec)	$\bar{\varepsilon}_\omega$ (rad/sec)	$\varepsilon_{M\omega}$ (rad/sec)	$\bar{\varepsilon}_\omega$ (rad/sec)	$\varepsilon_{M\omega}$ (rad/sec)
Commande PI+MLI	<b>0.1074</b>	<b>0.5780</b>	<b>0.2007</b>	<b>0.8151</b>	<b>0.1010</b>	<b>0.7756</b>
Mode Glissant	<b>0.0624</b>	<b>0.37</b>	<b>0.0620</b>	<b>0.3395</b>	<b>0.1218</b>	<b>0.9302</b>
Commande Booléenne	<b>0.0476</b>	<b>0.1720</b>	<b>0.0562</b>	<b>0.2177</b>	<b>0.1268</b>	<b>0.9853</b>
Commande Neuronale	<b>0.0645</b>	<b>0.4136</b>	<b>0.0620</b>	<b>0.3456</b>	<b>0.1110</b>	<b>0.7041</b>

Tableau IV-2 : Erreur moyenne absolue et erreur maximum de la vitesse.

Mesures	Charge Nominale		Variation de la charge		Bruits de Mesure	
	$\bar{\varepsilon}_{\phi_\alpha}$ (Wb)	$\bar{\varepsilon}_{\phi_\beta}$ (Wb)	$\bar{\varepsilon}_{\phi_\alpha}$ (Wb)	$\bar{\varepsilon}_{\phi_\beta}$ (Wb)	$\bar{\varepsilon}_{\phi_\alpha}$ (Wb)	$\bar{\varepsilon}_{\phi_\beta}$ (Wb)
Commande PI+MLI	<b>0.0712</b>	<b>0.0626</b>	<b>0.0708</b>	<b>0.0636</b>	<b>0.0988</b>	<b>0.0992</b>
Mode Glissant	<b>0.0180</b>	<b>0.0280</b>	<b>0.0180</b>	<b>0.0279</b>	<b>0.1007</b>	<b>0.1012</b>
Commande Booléenne	<b>0.0221</b>	<b>0.0214</b>	<b>0.0222</b>	<b>0.0216</b>	<b>0.1027</b>	<b>0.1009</b>
Commande Neuronale	<b>0.0178</b>	<b>0.0177</b>	<b>0.0182</b>	<b>0.0185</b>	<b>0.0979</b>	<b>0.0984</b>

Tableau IV-3 : Erreur moyenne absolue des flux.

Mesures	Charge Nominale		Variation de la charge		Bruits de Mesure	
	$\varepsilon_{M\phi_\alpha}$ (Wb)	$\varepsilon_{M\phi_\beta}$ (Wb)	$\varepsilon_{M\phi_\alpha}$ (Wb)	$\varepsilon_{M\phi_\beta}$ (Wb)	$\varepsilon_{M\phi_\alpha}$ (Wb)	$\varepsilon_{M\phi_\beta}$ (Wb)
Commande PI+MLI	<b>0.2165</b>	<b>0.1876</b>	<b>0.2002</b>	<b>0.1772</b>	<b>0.6511</b>	<b>0.6738</b>
Mode Glissant	<b>0.0892</b>	<b>0.0985</b>	<b>0.0895</b>	<b>0.0964</b>	<b>0.52</b>	<b>0.5391</b>
Commande Booléenne	<b>0.1011</b>	<b>0.0953</b>	<b>0.0984</b>	<b>0.0921</b>	<b>0.5203</b>	<b>0.5254</b>
Commande Neuronale	<b>0.3572</b>	<b>0.1031</b>	<b>0.3572</b>	<b>0.1032</b>	<b>0.5126</b>	<b>0.5246</b>

Tableau IV-4 : Erreur maximum des flux.

Mesures	Mode Normal	Variations de la charge	Bruits de Mesure
	<i>nc</i>	<i>nc</i>	<i>nc</i>
Commande PI+MLI	<b>240176</b>	<b>240165</b>	<b>260532</b>
Mode Glissant	<b>110123</b>	<b>111666</b>	<b>120970</b>
Commande Booléenne	<b>105806</b>	<b>104582</b>	<b>112646</b>
Commande Neuronale	<b>56542</b>	<b>59155</b>	<b>62492</b>

Tableau IV-5 : Nombre de commutations.

Quelques remarques peuvent être déduites de l'observation de ces résultats :

- Tout d'abord, les mesures des erreurs moyennes absolues sur le flux et la vitesse sont plus faibles pour les trois dernières méthodes de ces tableaux,
- toutes les commandes sont robustes vis-à-vis des variations paramétriques et des bruits de mesure,
- l'observation du tableau sur les erreurs maximums fait apparaître de fortes valeurs dans l'approche neuronale (charge nominale et variation de charge). Elles sont le résultat de la phase d'initialisation du réseau de neurones,

- en présence de bruits de mesure, les erreurs maximums sont importantes pour toutes les méthodes.
- l'étude du nombre de commutations montre une réduction sur les stratégies de commande booléenne et de commande neuronale par rapport aux autres méthodes. Cela apparaît plus particulièrement sur la commande neuronale. Ce phénomène peut être interprété par l'évolution dans le temps de la taille des classes.

### **IV.4.3 Comparaison des stratégies de mise en oeuvre**

La principale difficulté de mise en oeuvre pratique se situe au niveau du temps d'exécution des algorithmes. Il faut que celui-ci soit très inférieur à la période d'échantillonnage de la commande. En effet, les constantes de temps électriques sur ce système sont très faibles et donc la période d'échantillonnage doit être très petite. Des problèmes seraient certainement rencontrés sur la commande neuronale qui est plus lourde en temps de calcul, les autres méthodes ayant un temps d'exécution algorithmique beaucoup plus faible.

Pour toutes les méthodes, un estimateur ou un observateur de flux est nécessaire. Une mauvaise estimation altérerait les différentes commandes.

## **IV.5 Conclusion**

Nous avons testé en simulation les deux commandes originales développées aux chapitres II et III. Après avoir présenté l'exemple choisi d'une machine asynchrone associée à un convertisseur de puissance (onduleur) alimenté par une tension continue, nous avons implanté quatre algorithmes différents pour la commande du système. Nous avons, pour la boucle de flux, appliqué la commande "classique" MLI avec un correcteur proportionnel intégral (PI), puis la commande en mode glissant et la commande booléenne développée au chapitre II et enfin la commande neuronale.

Pour la comparaison, un profil de vitesse identique a été utilisé pour toutes les commandes. Des indicateurs ont été définis pour comparer numériquement les résultats.

Ceux-ci ont révélé que les différentes commandes booléennes avaient des performances comparables quant à la moyenne des erreurs absolues pour les flux, la vitesse, et le

---

maximum des erreurs obtenues. Toutefois, des différences apparaissent sur le nombre de commutations. En effet, les deux dernières méthodes améliorent les performances (commande booléenne et commande neuronale) puisqu'elles réduisent le nombre de commutations. La différence pour la MLI vient du fait que les instants de commutation ne sont pas contrôlés directement (modulation d'un signal haute fréquence).

Deux autres méthodes de commande booléenne auraient pu être utilisées : la commande directe de couple (DTC) et la "commande de Abadie" (cf. chapitre I). La première méthode n'a pas été retenue ici, car elle ne permet pas de traiter tout type de système à entrées booléennes. Quand à la deuxième, elle s'est révélée très difficile d'implémentation : les périodes d'échantillonnage qu'elle requiert sont très faibles.

L'avantage de nos approches est de prendre le système dans sa globalité, d'une part avec la connaissance du modèle (commande booléenne) et d'autre part sans la connaissance exacte de celui-ci (réseau de neurones). Cette dernière a l'avantage de supporter de plus larges variations paramétriques, et des bruits de mesure plus important (adaptation de la commande au modèle évoluant).



## **Conclusion Générale**



## Conclusion Générale

Au cours de ce mémoire, nous nous sommes intéressés au problème de la commande booléenne pour les systèmes continus modélisés sous la forme  $\dot{x} = f(x, u_1, u_2, \dots, u_m)$  avec  $u_i \in \{0,1\}$ .

Au cours du premier chapitre, quelques rappels sur les différentes commandes booléennes existant dans la littérature (MLI, Mode glissant) ont été effectuées. Nous avons ensuite rappelé une approche de commande booléenne développée dans une thèse de doctorat par V. Abadie au sein du LAIL.

Dans le second chapitre, une loi de commande en boucle fermée pour les systèmes à entrées booléennes a été proposée pour faire du suivi de trajectoire et de la régulation dans l'espace d'état. L'étude de la stabilité de cette loi de commande a été effectuée dans le cas linéaire. Les résultats obtenus ne sont pas tout à fait convaincants et demandent à être développés. Le problème des pertes de puissance par commutation dans les interrupteurs a été ensuite traité. Pour cela, nous avons proposé un algorithme de minimisation faisant intervenir la loi de commande booléenne développée au début de ce chapitre associée à une procédure de minimisation fondée sur la distance de Hamming. Enfin, ces résultats ont été validés et illustrés à travers des exemples en simulation.

Dans le troisième chapitre, nous avons proposé un algorithme de commande booléenne par réseau neurones faisant intervenir le principe de la commande précédente. Cette technique est fondée sur la classification des sorties mesurées pour chaque configuration possible. La structure du réseau de neurones, l'initialisation, l'apprentissage par rétropropagation du gradient et la commande ont été développés tout au long de ce chapitre. Un exemple de commande de machine synchrone nous a permis de valider les résultats obtenus.

Dans le chapitre IV, nous avons choisi d'étudier la commande d'une machine asynchrone associée à un onduleur de tension. Notre choix s'est porté plus particulièrement sur quatre commandes : la commande par correcteur PI (Proportionnel Intégral) associé à une MLI, la commande par modes glissants, et les méthodes développées dans le chapitre II et le chapitre III. L'analyse des résultats obtenus montre que les trois derniers algorithmes sont

assez robustes vis-à-vis des perturbations (variations paramétriques et bruits de mesures) par rapport à la commande PI+MLI. En outre, ces trois derniers algorithmes nécessitent des périodes d'échantillonnage plus faibles que l'algorithme PI+MLI. Il apparaît que nos approches ont l'avantage de moins solliciter les actionneurs électriques que les approches plus classiques. C'est notamment le cas pour la commande neuronale dont le nombre de commutations est le plus faible.

Nos approches ont été validées en simulation. Une implantation sur site réel permettrait de mettre en évidence des problèmes de mise en oeuvre impossibles à détecter par simulation

## Perspectives

Les suites que nous envisageons pour ces travaux sont les suivantes :

- Nous avons abordé l'étude de la stabilité de la commande au chapitre II. Il faut améliorer le résultat relatif à la stabilité, c'est à dire à trouver un domaine d'attraction autour du point désiré le plus petit possible. Il serait souhaitable également de quantifier l'énergie de la commande minimisant les commutations. Une possibilité serait d'utiliser par exemple un calcul de critère de performance énergétique (en référence à la commande optimale). L'influence de la période d'échantillonnage devra être prise en compte dans cette étude.
- 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 peut-être envisagé aisément pour les commandes proposées dans les chapitres II et III.
- Lors du développement de nos commandes nous avons émis l'hypothèse que l'état est mesurable ou reconstituable. Nos approches permettent une commande par retour de sortie, à la seule condition que la sortie du système et sa dérivée soient mesurables.

- 
- Une implantation sur machine réelle des algorithmes de commande doit être envisagée. Ceci pourrait se faire avec l'aide du LEEP de l'Ecole Centrale de Lille qui possède une installation de ce type, mais cela nécessitera de revoir les choix technologiques qui ont été effectués
  - La dernière piste concerne la conception du réseau de neurones. En effet le temps de convergence pour l'apprentissage pourrait être amélioré en utilisant soit une structure de réseau de neurones différente soit d'autres algorithmes d'apprentissage (Hopfield, Fonction à Radial de Base .....). Cette étude permettrait certainement de pallier à certains problèmes tels que l'initialisation des poids du réseau.



# **Bibliographie**



## Bibliographie

V. Abadie : "Commande des systèmes continus a entrées binaire. Application aux machines électriques", Thèse de l'UNIVERSITE DES SCIENCES ET TECHNIQUE DE LILLE No 1310 1994.

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 system", 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 Proceedings, Vol. 4, pp.595-600, Le Touquet Octobre 1993(c).

H. Abdi : "Les Réseaux de neurones" collection sciences et technologies de la connaissance, édition Plug 1994.

A.A. Andronov, S.E Vitt, S.E. Khaiken : "Theory of oscillator", Oxford : Pergamon press 1966.

C. Berger : "Graphes." Editions Bordas, Paris 1983.

C. Bergmann, : "contribution à l'étude de la commande numérique directe de moteurs synchrones à aimant- Modélisation, réalisation, expérimentations", Doctorat, Université Paris 6 1986.

P. Borne, G. Dauphin-Tanguy, J.P. Richard, F. I. Rotella, Zambettakis : "Modelisation et identification des processus" Tome 1. Collection : méthode et pratique de l'ingénieur. Edition Technip. 1992.

M. Branicky : "Multiple Lyapunov Functions and other analysis tools for switched and Hybrid Systems" IEEE Transaction on Automatic Control Vol 43 No 4, pp 475-482 1998.

P.C. Breedveld : "Physical systems theory in terms of bond-graph." Ph. D. Thesis Department of Electrical Engineering, Twente University of technology. P.O Box 217, 7500 AE Enschede, The Netherlands Feb 1984.

H. Buhler : "Réglage par mode de glissement", complément au traité d'électricité. Lausanne : presse polytechnique romande 1986.

J. Buisson : " Analysis and characterisation of hybrid system with bond graph" IEEE conference on SMC Le Touquet (France) pp 264-269 1993.

J. Buisson H. Cormerais : "Descriptor systems for the knowledge modelling and simulation of hybrid physical systems" APII-JESA Vol 32 No 9-10, pp 1047-1072 1998.

J.P Caron, J.P Hautier, J Faucher : "Électrotechnique : modélisation et commande de la machine asynchrone". Collection : méthode et pratique de l'ingénieur. Edition Technip. 1995.

D. Casadei, G. Serra, A. Tani : "Performances analysis of a DTC control scheme for induction motor in the low speed range". EPE97 Conference Trondheim (Norway), proceeding Vol. 3, pp 700-704 1997.

F. Cellier : "Continuous Systems modeling." Springer Verlag. 1991.

F.R.E Crossley : " The permutation of kinematic chains of eight members or less from the graph-theoretic viewpoint" Development in Theoretical and Applied Mechanics, Pergamon Press, Atlanta 1965.

G. Dauphin-Tanguy et S. Scavarda : "Modélisation des systèmes physiques par bond-graphs" Systèmes linéaires, Ouvrage collectif, Edition Masson, Paris. Vol 1, Chap. 2, pp. 35-112 1993

E. Davalo, P. Naim : "Des réseaux de neurones", Edition Eyrolles, seconde édition, 1993.

E. Delmotte : "Observateur robuste de flux pour la commande vectorielle d'une machine asynchrone", Thèse de l'UNIVERSITE DES SCIENCES ET TECHNIQUE DE LILLE No 2138, 1997.

M. Depenbrock : "Direct self control (DSC) of inverter-fed induction machines" IEEE transaction Power electronics , Vol. 3, no 4, pp 420-429, octobre 1988.

L. Dubois : "Utilisation de la logique floue dans la commande des systèmes complexes", Thèse de l'UNIVERSITE DES SCIENCES ET TECHNIQUE DE LILLE No 1588 1995.

---

J.P. Ducreux, A. Castelain, G. Dauphin-Tanguy and C. Rombaut: "Power electronics and electrical machines modelling using Bond-Graph ", IMACS Transactions on "Bond-Graph for Engineers "(eds. G. Dauphin-Tanguy and P. Breedveld) Elsevier, NewYork, 1992.

J.P. Ducreux,: "Modélisation des associations convertisseurs-machines pour le calcul des pertes électromagnétiques dans les moteurs à grande vitesse", Thèse de l'université des sciences et technique de Lille 1994.

S.V. Emelyanov, V.A. Taran : "Use of passives switched filters in variable structure control system",Engineering Cybernetics. No. 5,pp.143-149 1963(a).

S.V. Emelyanov, V.A. Taran : "On design of variable structure system for control of linear plants",Engineering Cybernetics. No. 2,pp.69-78 1963(b).

S.V. Emelyanov : "Invariant solutions of differential equations with discontinuous coefficient. Design principle of variable structure control systems", Mathematical Theory of Control. New York : Academic press 1967.

J. Faucher : "Quelques aspects de la Modulation de Largeur d'Impulsion" Journées 3EI, 1993.

A.F Filipov : "Differential equations with discontinuous right hand sides". Boston : Kluwer academic publisher, 1988.

A.F Filipov : "Application of the theory of differential equations with discontinuous right hand sides to non-linear problems in automatic control". Proceedings IFAC Congress 1960 p.923-927.

C. Forgez : "Méthodologie de modélisation et de commande par réseaux de neurones pour des dispositifs électrotechniques non-linéaires", Thèse de l'UNIVERSITE DES SCIENCES ET TECHNIQUE DE LILLE No 2400 1998.

A. Fossart : "Stabilisation, Commande et Poursuite par régime glissant", Rapport ONERA-CERT, contrat DRET, mars 1991.

K. Funashi : "On the approximate realization of continuous mappings by neural networks. Neural Networks, Vol. 2, pp. 183-192, 1989.

J. Garcia : "Approche bon-graph pour la modélisation des effets thermiques dans les composants de commutation en électronique de puissance", Thèse de l'UNIVERSITE DES SCIENCES ET TECHNIQUE DE LILLE 1997.

P. Gawthrop et L. Smith : "METAMODELLING : Bond-Graph and Dynamic systems." Prentice Hall International Series in Systems and Control Engineering. M.J. Grimble, Series Editor. 1996.

R. Genesio and A. Tesi : "Feedback of SISO bilinear systems", Int. J. Control, Vol. 50, No 4, pp.1487-1506, 1989.

G. Grellet, Guy Clerc : "Actionneurs électriques Principes modèles commande", Edition Eyrolles, 1996.

S. Hajry : " Commande dynamique par mode glissant application à la robustification des processus complexes" Thèse de l'UNIVERSITE DES SCIENCES ET TECHNIQUE DE LILLE 1997.

F. Hembert : "Etude et réalisation d'un filtre actif monophasé à structure multiniveaux obtenue par mise en série de ponts à quatre interrupteurs", Thèse de l'UNIVERSITE DES SCIENCES ET TECHNIQUE DE LILLE No 1929 1996.

W. Holderbaum, G. Dauphin-Tanguy, P. Borne : "Boolean control for linear system", I.S.I.A.C International Symposium on Intelligent Automation and control, Wac'98 Anchorage , USA, proceeding pp 212.1-212.6, 1998 .

W. Holderbaum, G. Dauphin-Tanguy, P. Borne: "Tracking control problem for switching linear system", CESA'98 IMACS-IEEE/SMC Conference Hammamet (Tunisia), proceeding Vol. 1, pp 935-940 1998.

W. Holderbaum, G. Dauphin-Tanguy, P. Borne : "Minimization of switching losses in Boolean control of linear systems", Accepté dans le congrès ELECTRIMACS 99 Lisbonne (Portugal) 1999.

W. Holderbaum, G. Dauphin-Tanguy, P. Borne : "A comparative study of different methods for Boolean control designing", Accepté dans le congrès CSCC 99 Grèce (Athènes) 1999.

---

W. Holderbaum, R. Canart, P. Borne : " Artificial Neural Networks Application to Boolean Input Systems Control ", dans la revue Studies in Informatics and Control Vol.8, No.2/June 1999.

N. Iqbal, J. Buisson, Y. Quenec'hdu : "Constrained Optimal Control of Descriptor systems : State feedback case" Symposium on Control, Optimization and Supervision, CESA'96, Lille 9-12, Juillet 1996.

Itkis : "Control system of variable structures ", J. Wiley : New York 1976.

J.F. Jodouin : "Les réseaux neuromimétiques", collection informatique Hermes 1994.

D. Karnopp, D. Margolis et R. Rosenberg. "System dynamics : A unified approach 2<sup>nd</sup> ed." John Wiley & Sons, Inc 1990.

M. Kuperstein : "Neural network model for adaptive hand eye coordination for single postures", Science, Vol. 239, pp 1308-1311, 1988.

W .Leonhard. : "Control of electrical drives", Springer, Berlin , 1985.

J. Lesenne, F. Notelet, et G. Séguier : "Introduction à l'électrotechnique approfondie", Technique et documentation, Paris 1981.

E. Levin, R. Gewirtzman, and G. Inbar : "Neural network architecture for adaptive system modeling and control", Neural Networks, Vol. 4, pp. 185-191 1991.

W. Mc Culloch, W. Pitts : " A logical calculus of the ideas immanent in nervous activity", Bulletin of mathematical Bio-physics, Vol 5, pp115-133 1943.

M. Minsky, S. Papert : " Perceptrons : A introduction to computational geometry", Cambridge Massachusetts, MIT Press 1969.

K S. Narendra and K . Parthasarathy. "Identification and Control of Dynamical Systems Using Neural Networks" IEEE Transaction on Neural Networks Vol 1 No 1, 1990.

H. Paynter : " Analysis and design of engineering systems" M.I.T. Press, Cambridge, Mass. 1961.

D.W Patterson : "Artificial neural networks, Theory and applications", Prentice Hall Singapore 1996.

N. Pierlot : "Application des réseaux neuronaux à la commande en vitesse d'une charge mécanique entraînée par machine asynchrones", Thèse de l'UNIVERSITE DES SCIENCES ET TECHNIQUE DE LILLE L2EP 1996.

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.

R. Rivera : "Modélisation par Bond-Graph et commande Booléenne d'un machine asynchrone", Rapport de DEA de l'UNIVERSITE DES SCIENCES ET TECHNIQUE DE LILLE 1996(a).

R. Rivera, E. Delmotte, A. Kamel, B. Semail, G. Dauphin-Tanguy : "Induction motor control using a Bond-Graph model of inverter fed machine", ELECTRIMACS 96 Proceedings, Vol. 2, pp.769-774, Saint-Nazaire Septembre 1996(b).

B. Robyns, H. Buyse, F. Labrique : "Fuzzy logic based field orientation in an indirect FOC strategy of an induction actuator", ELECTRIMACS 96 Proceedings, Vol. 1, pp.239-244, Saint-Nazaire Septembre 1996.

R. Rosenblatt : " Principles of neurodynamics", New York, Spatan Books 1962.

D.E. Rumelhart, G.E. Hinton, and R.J. Williams : "Learning internal representation by error propagation", Parallel Distributed Processing, Editors D.E. Rumelhart, J.L. McGelland, and the PDP Research group (Eds), Chapitre 8, pp.318-364 1986

A . Sabanovic and D.B Izosimov. "Application of Sliding modes to induction motor control", IEEE Transaction on Industry Applications, Vol. 17, No 1 pp 41-49, 1981.

G. Segquier, F. Labrique: "Les convertisseur de l'électronique de puissance", Editions technique et documentation Lavoisier. 1989.

B. Semail, E. Mendes, F. Bouillault, A. Razek : "Vector controlled induction machine simulation-parameter sensivity analysis", IMACS TC1, pp.291-296, Nancy, 1990

H Sira-Ramirez.: "Sliding regimes in general non linear systems : a relative degree approach", Int. J. Control, Vol. 50, No 4, pp.1487-1506, 1989.

---

H Sira-Ramirez.: "Sliding motions in bilinear switched networks", IEEE Transaction on Circuits and Systems, Vol. 34, No 8, pp.919-932, 1987.

H Sira-Ramirez.: "A geometric approach to pulse-Width Modulated control in nonlinear dynamical systems", IEEE Transaction on Circuits and Systems, Vol. 34, No 2, pp.184-187, 1989.

J E.Slotine. : "Sliding controller for non-linear systems", Int. J. Control. Vol. 40, No 2, pp.421-434, 1984.

J.U Thoma : "Simulation by bond-graph". Springer Verlag 1990.

H.M. Trent. "Isomorphisms between oriented linear graphs and lumped physical systems" J. of the Acoustical Society of America. Vol. 27, N°3, pp 500-527, 1954.

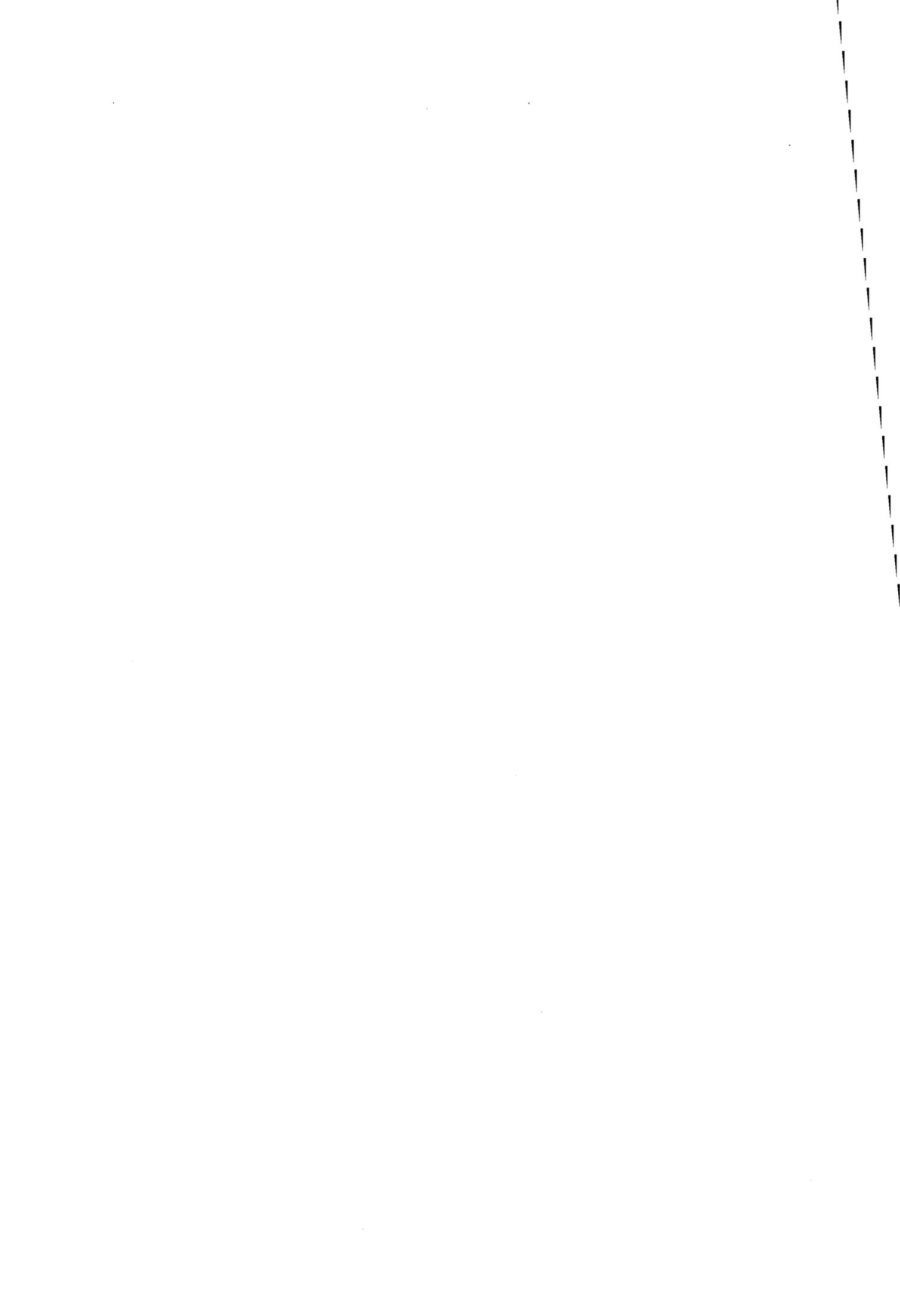
V.I. Utkin : "Variable structure systems with sliding mode control", IEEE Transaction on Automatic Control, Vol 22, pp. 212-222 1977.

V.I. Utkin : "Sliding mode in control optimisation", Springer Verlag 1978.

H. Ye A N. Michel and L. Hou : "Stability Theory for Hybrid Dynamical Systems" IEEE Transaction on Automatic Control Vol 43 No 4, pp 461-474 1998.



# **Annexes**



# Annexe A

## Principes de la méthodologie bond-graph

### A.1 Introduction

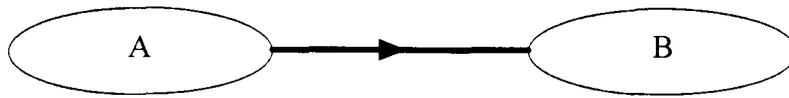
La modélisation par bond-graph permet une représentation graphique des systèmes physiques à travers un formalisme de type réseau [Paynter 1961]. L'origine de cette représentation graphique sont les graphes [Berger 1983], les graphes linéaires pour les circuits [Trent 1954] ou les graphes structurels des mécanismes [Crossley 1965]. La notion de bond-graph ne constitue qu'une étape d'abstraction supplémentaire de ces graphes où l'on a unifié différents domaines physiques [Breedveld 1984]. Nous ne présentons ici qu'un bref rappel de la modélisation bond-graph, pour plus d'informations le lecteur pourra se reporter à l'un des nombreux ouvrages de synthèse existants [Cellier 1991, Karnopp et *al.* 1990, Thoma 1990, Borne et *al.* 1992, Dauphin-Tanguy et Scavarda 1993, Gawthrop et Smith 1996]

### A.2 Présentation de l'outil bond-graph

C'est une technique de modélisation basée sur les échanges de puissance entre systèmes. Le postulat de départ est que la puissance instantanée  $P(t)$  d'un système peut être calculée à partir de deux variables qui sont l'effort  $e(t)$  et le flux  $f(t)$  tels que :

$$P(t) = e(t) f(t) \quad (\text{A-1})$$

Le sens de l'échange de puissance est représenté implicitement sur le bond-graph par une demi-flèche, comme indiqué Figure A-1 entre les deux sous-systèmes A et B interconnectés par une liaison physique.



$$A \xrightarrow[\textit{f}]{\textit{e}} B$$

Figure A-1 : Représentation de l'échange de puissance entre A et B.

Aux variables de puissance  $e$  et  $f$  sont associées les variables d'énergie  $p$  moment généralisé et  $q$  déplacement généralisé, obtenu par :

$$p(t) = \int_0^t e(\tau) d\tau + p(0)$$

$$q(t) = \int_0^t f(\tau) d\tau + q(0)$$
(A-2)

Le Tableau A-1 présente les variables généralisées  $e, f, p, q$  dans différents domaines de la physique.

	<b>Effort</b> $e(t)$	<b>flux</b> $f(t)$	<b>moment</b> $p(t)$	<b>déplacement</b> $q(t)$
<b>translation</b>	force $F$ (N)	vitesse $v$ (m/s)	impulsion $p$ (N.s)	déplacement $x$ (m)
<b>rotation</b>	couple $\tau$ (N.m)	vitesse ang. $\omega$ (rad/s)	impuls. ang. $H$ (N.m.s)	angle $\theta$ (rad)
<b>hydraulique</b>	pression $P$ (N/m <sup>2</sup> )	débit vol. $Q$ (m <sup>3</sup> /s)	impuls. press. $\Gamma$ (N.s/m <sup>2</sup> )	volume $V$ (m <sup>3</sup> )
<b>acoustique</b>	pression $P$ (N/m <sup>2</sup> )	vitesse vol. $\phi_v$ (m <sup>3</sup> /s)	impulsion $\Gamma$ (N.s/m <sup>2</sup> )	volume $V$ (m <sup>3</sup> )
<b>électrique</b>	tension $u$ (V)	intensité $i$ (A)	impuls. tension $\Phi$ (V.s)	charge $q$ (Cb)
<b>magnétique</b>	force magnétomot. $\lambda$ (A)	dérivée du flux magn. $\dot{\Phi}$ (wb/s)		flux magn. $\Phi$ (wb)
<b>chimique</b>	pot. Chim. $\mu$ (J/mole)	flux molaire $\dot{N}$ (mol/s)		masse mol. $N$ (mol)
<b>thermodynamique</b>	température $T$ (K)	flux d'entropie $\dot{S}$ (wb/K)		entropie $S$ (J/K)

Tableau A-1 : Variables des puissance ( $e, f$ ) et d'énergie ( $p, q$ )

On peut remarquer que dans les domaines magnétique, chimique et thermodynamique la notion de moment n'est pas définie. Cela signifie tout simplement que dans ces domaines il n'existe pas de phénomène physique correspondant analogue à une inertie ou une bobine.

### A.3 Variables mises en jeu

Les éléments bond-graphs peuvent se classier de la façon suivante :

- *éléments passifs* (reçoivent de la puissance) :  $R$  : dissipation d'énergie  
 $C, I$  : stockage d'énergie
- *éléments actifs* (fournissent la puissance) :  $Se, Sf$  (sources)
- *éléments de jonction multi-ports* :  $0, 1, TF, GY$  (conservatifs de puissance)

### A.4 Notion de causalité

Les bond-graphs représentent l'architecture du système, où apparaissent les échanges de puissance entre éléments. De plus, ils permettent de définir la structure de calcul et de mettre en évidence des relations de cause à effet au sein du système. Ceci est un avantage évident sur les représentations graphiques comme les graphes de fluence.

Lorsque deux sous-systèmes  $A$  et  $B$  sont couplés, tels que  $A$  transmet à  $B$  la puissance  $P = ef$ , nous avons deux situations possibles :

$A$  applique à  $B$  un "effort"  $e$ , qui réagit en envoyant à  $A$  un "flux"  $f$ ;

$A$  envoie à  $B$  un "flux"  $f$ , qui répond par un "effort"  $e$ .

Ces deux cas conduisent à deux schémas-blocs différents :



Figure A-2 : Deux situations de causalité

Ces relations de cause à effet sont représentées sur le modèle bond-graph, par le *trait causal*.



**LES ELEMENTS DE BASE BOND-GRAPH ET LEURS PROPRIETES**

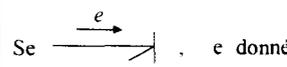
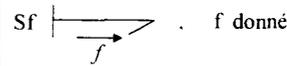
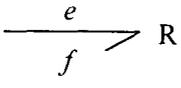
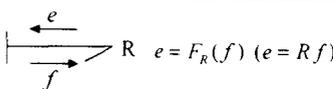
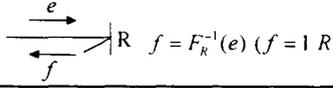
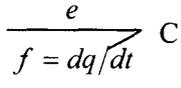
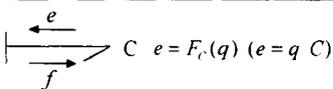
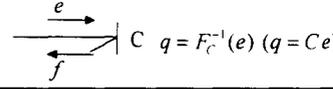
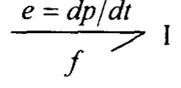
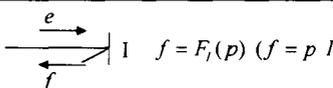
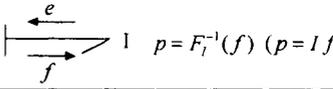
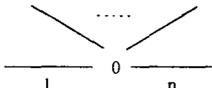
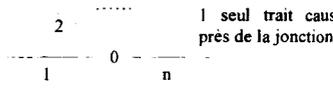
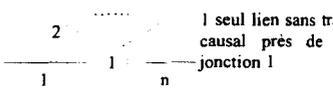
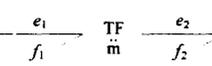
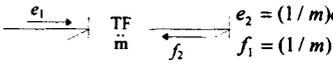
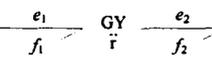
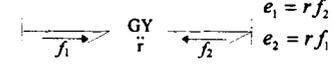
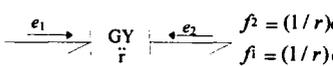
Éléments bond-graph	Symbole	Loi générique	Exemples physiques	Causalités
<b>Variables de puissance</b>	effort $e$ flux $f$		force, couple, tension, pression ... vitesse, vitesse angulaire, courant, débit, volume	
<b>Variables d'énergie</b>	moment $p$ déplacement $q$	$p = \int e dt$ $q = \int f dt$	impulsion, flux magnétique, ... déplacement, charge, volume, ...	
<b>Éléments actifs (sources)</b>	Se  Sf 	$e$ indépendant de $f$ $f$ indépendant de $e$	pesanteur, générateur de tension, ... générateur de courant, pompe, ...	Se  , $e$ donné Sf  , $f$ donné
<b>Éléments passifs 1-port</b>	 R	$\Phi_R(e, f) = 0$	amortisseur, résistance, restriction hydraulique, frottement,	 $e = F_R(f)$ ( $e = Rf$ )  $f = F_R^{-1}(e)$ ( $f = 1/R e$ )
	 C	$\Phi_C(e, q) = 0$	ressort, condensateur, réservoir, élasticité, ...	 $e = F_C(q)$ ( $e = q C$ )  $q = F_C^{-1}(e)$ ( $q = C e$ )
	 I	$\Phi_I(p, f) = 0$	masse, inertie, self, ...	 $f = F_I(p)$ ( $f = p I$ )  $p = F_I^{-1}(f)$ ( $p = I f$ )
<b>Éléments de jonction (conservatifs de puissance)</b>		$e_1 = \dots = e_n$ $\sum_{i=1}^n a_i f_i = 0$ ( $a_i = \pm 1$ )	connection série en mécanique connection parallèle en électrique	 I seul trait causal près de la jonction 0 ex. : $e_2 = e_1, \dots, e_n = e_1, f_1 = a_1 \sum_{i=1, i \neq 1}^n a_i f_i$
		$f_1 = \dots = f_n$ $\sum_{i=1}^n a_i e_i = 0$ ( $a_i = \pm 1$ )	connection parallèle en mécanique connection série en électrique	 I seul lien sans trait causal près de la jonction 1 ex. : $f_2 = f_1, \dots, f_n = f_1, e_1 = a_1 \sum_{i=2}^n a_i e_i$
<b>Éléments de jonction (conservatifs de puissance)</b>		$e_1 = m e_2$ $f_2 = m f_1$	levier, système d'engrenages ou de poulies, réducteur, transformateur électrique, changement de domaine physique	 $e_1 = m e_2$ $f_2 = m f_1$  $e_2 = (1/m) e_1$ $f_1 = (1/m) f_2$
		$e_1 = r f_2$ $e_2 = r f_1$	moteur à courant continu, capteur à effet Hall, changement de domaine physique	 $e_1 = r f_2$ $e_2 = r f_1$  $f_2 = (1/r) e_1$ $f_1 = (1/r) e_2$

Tableau A-2: récapitulatif des éléments de base bond-graph et de leurs propriétés.



## Annexe B

# Commande par Mode Glissant

### B.1 Introduction

Les systèmes travaillant en mode de glissement sont des systèmes discontinus. On désigne par système discontinu, un système décrit par une équation différentielle dont le second membre n'est pas continu en l'état. Ce système présente des discontinuités sur une ou des surfaces (ou des variétés) dans l'espace d'état. Ce dernier présente alors un comportement différent (une équation différentielle différente) selon qu'on est d'un côté ou de l'autre de la surface de discontinuité. On dit que le système change de structure, c'est pour cette raison que dans la littérature ces systèmes sont connus sous le nom des Systèmes à Structure Variable.

### B.2 Systèmes à structure variable

La théorie des systèmes à structure variable a pris son essor dans les années 60 grâce à Emelyanov [Emelyanov 1963] et [Itkis 1976], à cause de la structure du contrôleur utilisé qui peut changer d'une façon discontinue entre deux ou plusieurs structures. C'est le cas, par exemple des circuits de conversion de puissance, où le système est gouverné par une équation différentielle différente pour chaque position de l'interrupteur. Un modèle serait du type  $\dot{x} = f(x, t)$  où le champ de vecteur  $f$  prend diverses valeurs selon les régions de l'espace d'état où l'on se trouve.

La théorie de la commande "bang-bang" des systèmes à relais constitue un cas particulier de la théorie des systèmes à structure variable. Elle a donné naissance aux notions de régime glissant qui se sont étendus par la suite aux systèmes discontinus réalisés artificiellement [Utkin 1977], c'est à dire des systèmes du type  $\dot{x} = f(x, t)$ , avec  $f$  fonction non linéaire continue, et pour lesquels on définit un retour d'état discontinu. Cette approche a pour but de maintenir l'état représentatif de l'évolution du système sur une variété  $S$

définie au préalable, appelée surface de glissement. Dans ce contexte et sous certaines conditions, le système est dit en régime glissant.

La dynamique du système et sa stabilité sont indépendantes de la fonction  $f(x, t)$  et dépendent seulement des paramètres de l'hypersurface choisie, ce qui explique la robustesse de la loi de commande vis-à-vis des variations paramétriques.

Avant de s'intéresser aux régimes glissants, nous allons rappeler quelques notions relatives aux systèmes discontinus.

### **B.3 Le formalisme des équations différentielles à second membre discontinu**

Ce bref rappel ne prétend pas à l'exhaustivité. Pour plus de détails, se reporter à [UTKIN 1977]

#### **B.3.1 Le problème de l'existence et de l'unicité de la solution.**

Considérons un système dont l'évolution des variables d'état est régie par l'équation différentielle suivante :

$$\dot{x} = f(x, t) \tag{B-1}$$

où  $x = [x_1, x_2, \dots, x_n]^T \in \mathcal{X}$  (avec  $\mathcal{X} \subset \mathbb{R}^n$ ) représente le vecteur état et  $f = [f_1, f_2, \dots, f_n]^T$  est un champs de vecteurs dépendant de  $x$  et  $t$ .

Le champ de vecteurs  $f$  n'est pas défini sur une variété de codimension 1, ( $s(x) = 0$ ). Du fait de la discontinuité de  $f$ , il n'est pas possible d'appliquer le théorème de Cauchy-Lipschitz, sur l'existence et l'unicité d'une solution à l'équation (B-1).

De nombreux mathématiciens et physiciens (Utkin, Filipov) se sont intéressés à l'existence de solutions sur la surface  $s(x) = 0$ , en imaginant prolonger le champ  $f$  sur la surface  $s$ .

La manière dont ils ont procédé est la suivante : dans la réalité, les systèmes ne peuvent pas être discontinus. De nombreux phénomènes, (retards, hystérésis, saturations) ne sont pas pris en compte pour l'établissement de la loi d'évolution du système.

On imagine donc un processus qui considère un voisinage tubulaire autour de la surface  $s$  (dû aux phénomènes négligés), et l'on fait tendre le diamètre du tube vers 0, pour retrouver le cas idéal.

Aborder le problème sous cet angle permet d'utiliser tous les théorèmes habituellement à notre disposition pour la résolution d'équations différentielles à second membre continu. Le mode glissant apparaît alors comme une limite d'une solution classique.

Dans ce qui suit, on présente la méthode de Filipov, utilisée pour les systèmes "naturellement" discontinus.

### B.3.2 Méthode de Filipov [FILIPOV 60].

Considérons un système régi par l'équation (B-1), on suppose que le champ  $f$  rend attractive la variété  $s(x)=0$ . Les vecteurs vitesses sont donc orientés vers la surface :

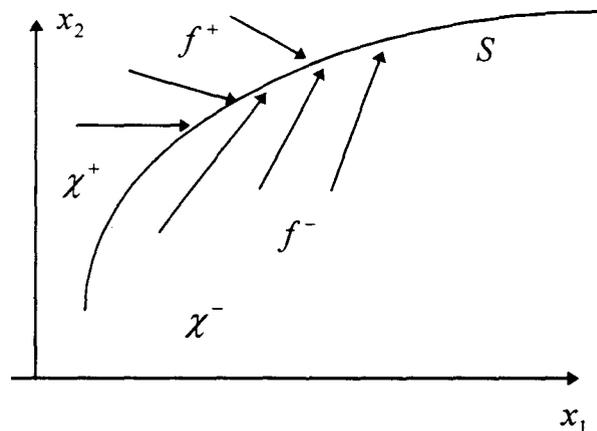


Figure C-1: Attractivité de la surface  $S$

On a alors :

$$\chi = \begin{cases} \chi^+ & \text{si } s(x) > 0 \\ \chi^- & \text{si } s(x) < 0 \end{cases} \quad (\text{B-2})$$

L'espace d'état est séparé en trois partitions :

$$\chi = \chi^+ + \chi^- + S \quad (\text{B-3})$$

Le champ  $f$  est alors défini par :

$$f(x) = \begin{cases} f^+ & \text{si } s(x) > 0 \\ f^- & \text{si } s(x) < 0 \end{cases} \quad (\text{B-4})$$

Soient  $f_n^+$  et  $f_n^-$  les projections respectives de  $f^+$  et  $f^-$  sur la normale à la surface  $s(x)=0$ . L'existence et l'unicité de solutions de l'équation ont été démontrées par Filipov [FILIPOV 60]. Ces théorèmes concluent que pour prolonger une solution  $x(t)$  de (B-1) sur la surface  $s$ , il faut que cette dernière soit **attractive** au moins dans un voisinage de cette surface. En résumé, on aboutit à une condition de glissement.

### B.3.3 Condition de glissement

La condition d'existence des solutions données par le théorème de Filipov [FILIPOV 60] peut s'exprimer plus aisément de la manière suivante :

$$\boxed{\{f_n^+ < 0 \text{ et } f_n^- > 0\}} \Leftrightarrow ss' < 0 \quad (\text{B-5})$$

Cette condition, essentielle pour la régulation d'un système par modes glissants, est appelée condition de glissement.

## B.4 Méthode de la commande équivalente

Nous allons présenter dans ce paragraphe la commande équivalente pour les systèmes non linéaire de la forme  $\dot{x} = f(x) + g(x)u$ .

### B.4.1 Systèmes non-linéaires et monovariables en entrée

Le principe de la commande équivalente est utilisé pour un système "**artificiellement discontinus**". La dynamique glissante résultant de l'action du champ équivalent, donné par la commande, est appelée commande équivalente. Considérons le système non linéaire décrit par (B-6), dont l'espace d'état  $x$  est de dimension  $n$ .

$$\dot{x} = f(x) + g(x)u \quad (\text{B-6})$$

avec  $g = [g_1, \dots, g_n]^T$  un champ de vecteurs définissant les gains de commande et  $u \in \mathfrak{R}$  une loi de commande vérifiant :

$$u = \begin{cases} u^+ & \text{si } s(x) > 0 \\ u^- & \text{si } s(x) < 0 \end{cases} \quad (\text{B-7})$$

On cherche une loi de commande  $u_{eq}$ , sur la surface  $s$ , qui vérifie :

$$f_{eq}(x) = f(x) + g(x)u_{eq} \quad (\text{B-8})$$

dans laquelle,  $u_{eq}$  est la commande équivalente qui vérifie, pour  $x \in S$ , l'inégalité suivante :

$$\min(u^-(x), u^+(x)) < u_{eq}(x) < \max(u^-(x), u^+(x)) \quad (\text{B-9})$$

On peut interpréter  $u_{eq}$  comme étant la valeur moyenne instantanée de  $u(t)$  lors de la commutation rapide entre  $\min(u^-(x), u^+(x))$  et  $\max(u^-(x), u^+(x))$ , [UTKIN 92].

En outre la commande équivalente doit aussi permettre l'invariance de la surface  $s$  [EMELYONOV 1963], c'est à dire :

$$\begin{cases} \dot{s}(x) = 0 \\ \dot{\dot{s}}(x) = 0 \end{cases} \quad (\text{B-10})$$

Il vient alors, en développant les calculs :

$$\dot{s} = \langle \nabla s, f \rangle = 0 \quad (\text{B-11})$$

ou dans notre cas,  $\dot{s}(x) = \frac{\partial s}{\partial x} \frac{dx}{dt} = 0$

d'où :

$$\begin{cases} u_{eq} = -\frac{\langle \nabla s, f \rangle}{\langle \nabla s, g \rangle} = -\left(\frac{\partial s}{\partial x} g(x)\right)^{-1} \frac{\partial s}{\partial x} f(x) \\ \text{sous la condition } \langle \nabla s, g \rangle \neq 0 \end{cases} \quad (\text{B-12})$$

En portant l'expression de  $u_{eq}$  dans (B-8), on obtient alors la solution de l'équation différentielle sur la surface :

$$\dot{x} = -f(x) - g(x)\left(\langle \nabla s, g \rangle\right)^{-1} \langle \nabla s, f \rangle = f(x) - g(x)\left(\frac{\partial s}{\partial x} g(x)\right)^{-1} \frac{\partial s}{\partial x} f(x) \quad (\text{B-13})$$

### B.4.2 Systèmes non linéaires, multivariables en entrée

Dans cette section, on s'intéresse aux cas des systèmes non linéaires multivariables en entrée [Utkin 1977, Sira Ramirez 1988,1990, Decarlo 1988]. Ces systèmes présentent un intérêt particulier, du fait que la plupart des systèmes physiques réels peuvent s'écrire sous cette forme (comme les machines électriques). Reprenons le système décrit par (B-6), mais avec  $g$  une matrice de commande de dimension  $n \times m$ .

$u_i$  est la  $i$ -ème composante ( $i \in \{1, \dots, m\}$ ) du vecteur commande défini sur  $\mathcal{R}^m$ , et subissant une discontinuité du type :

$$u_i = \begin{cases} u_i^+ & \text{si } s_i(x) > 0 \\ u_i^- & \text{si } s_i(x) < 0 \end{cases} \quad (\text{B-14})$$

sur une surface  $s_i(x)=0$  ( $i = \{1, \dots, m\}$ ).

Le régime glissant idéal obtenu grâce au vecteur commande équivalente  $u_{eq}$ , s'effectue en satisfaisant la condition (B-5) ou  $s_i \dot{s}_i \leq 0$  au voisinage de  $s_i(x)=0$ . Cette dernière devient, en utilisant le modèle ([Utkin 1977]) :

$$J_x(x)f(x) + J_x(x)g(x)u_{eq} = 0 \quad (\text{B-15})$$

avec  $J_x(x)$  une matrice de dimension  $m \times n$ , dont les lignes correspondent aux vecteurs gradient de  $s_i(x)$  :

$$J_x(x) = \begin{bmatrix} \frac{\partial s_1}{\partial x_1} & \dots & \frac{\partial s_1}{\partial x_n} \\ \vdots & & \vdots \\ \frac{\partial s_m}{\partial x_1} & \dots & \frac{\partial s_m}{\partial x_n} \end{bmatrix} \quad (\text{B-16})$$

On définit alors la commande équivalente :

$$u_{eq} = -(J_x(x)g(x))^{-1} J_x(x)f(x) \quad (\text{B-17})$$

sous la condition que  $(J_x(x)g(x))$  soit inversible.

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

$$\begin{cases} \dot{x} = f(x) - g(x)(J_x(x)g(x))^{-1} J_x(x)f(x) \\ \text{Condition : } (J_x(x)g(x))^{-1} \text{ inversible} \end{cases} \quad (\text{B-18})$$

Si un régime de glissement apparaît sur la surface 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 multivariable, le mode de glissement existe s'il y a un régime de commutation sur toutes les intersections de toutes les surfaces. Il existe dans la littérature [HARASHIMA 1985] [FOSSARD 1991] d'autres manières pour commuter . Le lecteur pourra s'y référer pour plus de détails sur les conditions d'existence de surface de glissement, sur les différentes méthodes pour rejoindre l'intersection des surfaces de commutation ainsi que sur l'étude de la stabilité de la commande.



## Annexe C

# Calcul de la rétropropagation du gradient

### C.1 Introduction

Au cours de cette annexe nous allons développer les calculs de la rétropropagation du gradient. Dans un premier temps, nous définirons les variables utilisées par notre réseau de neurones (chapitre III). Puis nous rappellerons l'algorithme du gradient utilisé pour minimiser une fonction quadratique. Enfin le calcul du gradient sur chaque couche du réseau de neurones (couche cachée et couche de sortie) sera exposé.

### C.2 Définition des variables

Dans ce paragraphe nous allons définir sous forme matricielle les variables et les fonctions d'activation qui seront nécessaires par la suite. On note :

$e$  : Le vecteur représentant le stimulus pour l'entrée du réseau.

$y$  : Le vecteur représentant la réponse des cellules de la couche de sortie lorsque un stimulus  $e$  est présenté en entrée.

$h$  : Le vecteur représentant la réponse des cellules de la couche cachée lorsque un stimulus  $e$  est présenté en entrée.

$y^d$  : Le vecteur représentant la réponse désirée des cellules de la couche de sortie pour le stimulus  $e$  présenté en entrée.

$W$  : La matrice des valeurs des connexions reliant les cellules de la couche cachée aux cellules de la couche de sortie;  $w_{ij}$  donne la valeur de la connexion entre le  $j^{\text{ème}}$  neurone de la couche cachée et le  $i^{\text{ème}}$  neurone de la couche de sortie.

$W_b$  : La matrice des valeurs des biais des cellules de la couche de sortie;  $w_{b,i}$  donne la valeur du poids de biais du  $i^{\text{ème}}$  neurone de la couche de sortie.

$V$  : La matrice des valeurs des connexions reliant les cellules de la couche d'entrée aux cellules de la couche cachée;  $v_{jl}$  donne la valeur de la connexion entre le  $l^{\text{ème}}$  neurone de la couche d'entrée et le  $j^{\text{ème}}$  neurone de la couche cachée.

$V_b$  : La matrice des valeurs des biais des cellules de la couche cachée;  $v_{b,j}$  donne la valeur du poids de biais du  $j^{\text{ème}}$  neurone de la couche cachée.

Pour utiliser la technique de la rétro-propagation, la réponse des cellules doit être une fonction continue de l'état d'activation de la cellule. En notant  $net$  l'état d'activation de la cellule  $q$  (qui peut être une cellule de la couche cachée  $net_q^1$  ou de la couche de sortie  $net_q^2$ ), la réponse de la cellule (notée  $y_q$ ) sera :

$$y_q = f_a(net_q) \tag{C-1}$$

Il existe plusieurs fonctions d'activation  $f$  satisfaisantes pour un neurone. Dans notre cas, nous avons pris la fonction sigmoïde :

$$f_a(net_q) = \frac{1}{1 + e^{-net_q}} \tag{C-2}$$

en grande partie parce que le calcul de sa dérivée est simple :

$$\frac{df_a(net_q)}{d(net_q)} = \frac{e^{-net_q}}{(1 + e^{-net_q})^2} \tag{C-3}$$

La notation matricielle permet une expression simple des opérations successives à effectuer. Ainsi lorsqu'un stimulus est présenté en entrée, le vecteur de réponse des cellules de la couche cachée  $h$  est donné par :

$$h = F_a(NET^1) \tag{C-4}$$

avec  $F_a(NET^1) = [f_a(net_1^1) \quad f_a(net_2^1) \quad \dots \quad f_a(net_{2^m}^1)]^T$  et  $NET^1 = Ve + V_b$

La réponse de la couche de sortie  $y$  est donnée par :

$$y = F_a(NET^2) \quad (C-5)$$

avec  $F_a(NET^2) = \left[ f_a(net_1^2) \quad f_a(net_2^2) \quad \dots \quad f_a(net_m^2) \right]^T$  et  $NET^2 = Wh + W_b$

### C.3 La méthode du gradient

L'idée générale de cette technique consiste à chercher le minimum d'une fonction de plusieurs variables ou paramètres par itérations successives. Le gradient d'une fonction matricielle est défini comme la matrice des dérivées partielles de cette fonction. Les valeurs des paramètres sont dans notre cas les poids du réseau et la fonction à minimiser est une fonction quadratique de l'erreur. Cette fonction quadratique est définie comme :

$$J = \frac{1}{2} \varepsilon_m^T \varepsilon_m \quad (C-6)$$

avec pour erreur :

$$\varepsilon_m = y^d - y \quad (C-7)$$

L'algorithme se déroule comme suit :

1. Choisir les valeurs initiales des poids ( $W, V, W_b, V_b$ ), de manière aléatoire en règle générale.

2. Calculer le gradient de  $J$  pour chaque matrice de poids :

$$\nabla_W J = \frac{\partial J}{\partial W}; \quad \nabla_V J = \frac{\partial J}{\partial V}; \quad \nabla_{W_b} J = \frac{\partial J}{\partial W_b}; \quad \nabla_{V_b} J = \frac{\partial J}{\partial V_b} \quad (C-8)$$

3. Corriger les poids en direction inverse du gradient de  $W, V, W_b$  et  $V_b$  (avec  $\eta$  et  $\eta_b$  dénotant les constantes de proportionnalité)

$$\begin{cases} W_{new} = W_{old} - \eta \frac{\partial J}{\partial W} \\ W_{b,new} = W_{b,old} - \eta_b \frac{\partial J}{\partial W_b} \\ V_{new} = V_{old} - \eta \frac{\partial J}{\partial V} \\ V_{b,new} = V_{b,old} - \eta_b \frac{\partial J}{\partial V_b} \end{cases} \quad (C-9)$$

4. Continuer les étapes 2 et 3, tant que l'écart entre les nouveaux poids ( $W_{new}$ ,  $V_{new}$ ,  $W_{b,new}$ ,  $V_{b,new}$ ) et les anciens poids ( $W_{old}$ ,  $V_{old}$ ,  $W_{b,old}$ ,  $V_{b,old}$ ) est jugé important.

Dans les deux prochains paragraphes nous allons calculer les gradients de chaque matrice de poids puis nous exprimerons dans chaque cas la mise à jour (ou correction) de ces poids.

## C.4 Correction pour la couche de sortie

Pour la couche de sortie, on cherche à modifier les valeurs de  $W$  et  $W_b$ .

### C.4.1 Poids $W$

Le gradient de  $J$  par rapport à un poids  $w_{ij}$  se calcule en utilisant la règle de dérivation des fonctions composées :

$$\frac{\partial J}{\partial w_{ij}} = \frac{\partial J}{\partial y_i} \frac{\partial y_i}{\partial (net_i^2)} \frac{\partial (net_i^2)}{\partial w_{ij}} \quad (C-10)$$

Evaluons le premier terme de cette expression :

$$\frac{\partial J}{\partial y} = \frac{1}{2} \frac{\partial}{\partial y_i} \left( (y^d - y_i)^T (y^d - y_i) \right) \quad (C-11)$$

On déduit alors le premier terme :

$$\frac{\partial J}{\partial y} = -(y^d - y_i) \quad (C-12)$$

A présent, calculons le second terme de l'expression (C-10)

$$\frac{\partial y_i}{\partial (net_i^2)} = \frac{\partial_a (net_i^2)}{\partial (net_i^2)} = f'_a (net_i^2) \quad (C-13)$$

Puis le dernier terme :

$$\frac{\partial (net_i^2)}{\partial w_{ij}} = \frac{\partial \left( \sum_{j=1}^{2^n} w_{ij} h_j + w_{b,i} \right)}{\partial w_{ij}} = h_j \quad (C-14)$$

Posons à présent :

$$\delta_{w_{ij}} = - \frac{\partial J}{\partial y_i} \frac{\partial y_i}{\partial (net_i^2)} \quad (C-15)$$

$$\delta_{w_{ij}} = (y^d - y_i) f'_a (net_i^2) \quad (C-16)$$

$$\delta_{w_{ij}} = (y^d - y_i) \cdot f'_a \left( \sum_{j=1}^{2^n} w_{ij} h_j + w_{b,i} \right) \quad (C-17)$$

Ecrivons cette expression sous forme matricielle :

$$\delta_w = (y^d - y) * F'_a (W \cdot F_a (V_e + V_b) + W_b) \quad (C-18)$$

\* désigne le produit de HADAMAR. Par exemple le produit de HADAMAR de deux matrices A, B se note  $[A * B] = [a_{i,j} \cdot b_{i,j}]$

La correction à apporter à  $W$  à chaque étape est :

$$\Delta W = -\eta \frac{\partial J}{\partial W} = -\eta (-\delta_w h^T) \quad (C-19)$$

$$\Delta W = \eta \delta_w \cdot [F_a (V_e + V_b)]^T \quad (C-20)$$

Avec  $\eta$  la constante de proportionnalité sur la matrice de poids  $W$ .

On obtient alors la mise à jour des poids par la relation suivante :

$$W_{new} = W_{old} + \eta \delta_W [F_a (Ve + V_b)]^T \quad (C-21)$$

### C.4.2 Poids $W_b$

Le gradient de  $J$  par rapport à  $w_{b,j}$  se calcule en utilisant la règle de dérivation des fonctions composées :

$$\frac{\partial J}{\partial w_{b,j}} = \frac{\partial J}{\partial y_i} \frac{\partial y_i}{\partial \text{net}_i^2} \frac{\partial \text{net}_i^2}{\partial w_{b,j}} \quad (C-22)$$

Les deux premiers termes ont été calculés précédemment ( $\delta_{w_j}$ ), calculons le dernier terme de cette expression :

$$\frac{\partial \text{net}_i^2}{\partial w_{b,j}} = \frac{\partial \left( \sum_{j=1}^{2^m} w_{ij} h_j + w_{b,j} \right)}{\partial w_{b,j}} = 1 \quad (C-23)$$

En utilisant le coefficient de proportionnalité  $\eta_b$ , on obtient alors la mise à jour des poids de biais par la relation sous forme matricielle suivante :

$$W_{b,new} = W_{b,old} + \eta_b \delta_W \quad (C-24)$$

## C.5 Correction pour la couche cachée

Pour la couche cachée, on cherche à modifier les valeurs de  $V$ .

### C.5.1 Poids $V$

Le gradient de  $J$  par rapport à  $v_{jl}$  se calcule en utilisant la règle de dérivation des fonctions composées :

$$\frac{\partial J}{\partial v_{jl}} = \frac{\partial J}{\partial y_i} \frac{\partial y_i}{\partial \text{net}_i^2} \frac{\partial \text{net}_i^2}{\partial h_j} \frac{\partial h_j}{\partial \text{net}_i^1} \frac{\partial \text{net}_i^1}{\partial v_{jl}} \quad (C-25)$$

Les deux premiers termes sont déjà connus, ils correspondent à  $-\delta_{w_y}$  :

$$\frac{\partial J}{\partial v_{jl}} = -\delta_{w_y} \frac{\partial(\text{net}_i^2)}{\partial h_j} \frac{\partial h_j}{\partial(\text{net}_i^1)} \frac{\partial(\text{net}_i^1)}{\partial v_{jl}} \quad (\text{C-26})$$

En évaluant chacun des termes restants, on trouve pour le premier :

$$\frac{\partial(\text{net}_i^2)}{\partial h_j} = \frac{\partial\left(\sum_{j=1}^{2^m} w_{yj} h_j + w_{b,j}\right)}{\partial h_j} = w_{yj} \quad (\text{C-27})$$

Le second

$$\frac{\partial h_j}{\partial(\text{net}_i^1)} = \frac{\partial_a(\text{net}_i^1)}{\partial(\text{net}_i^1)} = f'_a(\text{net}_i^1) \quad (\text{C-28})$$

puis finalement :

$$\frac{\partial(\text{net}_i^1)}{\partial v_{jl}} = \frac{\partial\left(\sum_{l=1}^n v_{jl} e_l + v_{b,j}\right)}{\partial v_{jl}} = e_l \quad (\text{C-29})$$

On obtient alors pour l'expression (C-25) le résultat suivant :

$$\frac{\partial J}{\partial v_{jl}} = -\delta_{w_y} w_{yj} f'_a(\text{net}_i^1) e_l \quad (\text{C-30})$$

Ecrivons cette expression sous forme matricielle :

$$\frac{dJ}{dV} = -\left((W^T \delta_w) * F'_a(Ve + V_b)\right) e^T \quad (\text{C-31})$$

Posons à présent :

$$\delta_v = (W^T \delta_w) * F'_a(Ve + V_b) \quad (\text{C-32})$$

La correction à apporter à  $V$  à chaque étape est :

$$\Delta V = -\eta \frac{\partial J}{\partial V} = \eta \delta_v [e]^T \quad (\text{C-33})$$

On obtient alors la mise à jour des poids par la relation suivante :

$$V_{new} = V_{old} + \eta \delta_v [e]^T \quad (C-34)$$

### C.5.2 Poids $V_b$

Le gradient de  $J$  par rapport à  $v_{b,j}$  se calcule en utilisant la règle de dérivation des fonctions composées :

$$\frac{\partial J}{\partial v_{b,j}} = \frac{\partial J}{\partial y_i} \frac{\partial y_i}{\partial (net_i^2)} \frac{\partial (net_i^2)}{\partial x_j} \frac{\partial x_j}{\partial (net_i^1)} \frac{\partial (net_i^1)}{\partial v_{b,j}} \quad (C-35)$$

Les quatre premiers termes sont déjà connus, ils correspondent à  $-\delta_{v_j}$  (une composante de  $-\delta_v$ ). Evaluons le dernier terme de cette expression :

$$\frac{\partial (net_i^1)}{\partial v_{b,j}} = \frac{\partial \left( \sum_{l=1}^n v_{jl} e_l + v_{b,j} \right)}{\partial v_{b,j}} = 1 \quad (C-36)$$

En utilisant le coefficient de proportionnalité  $\eta_b$ , on obtient alors la mise à jour des poids de biais par la relation sous forme matricielle suivante :

$$V_{b,new} = V_{b,old} + \eta_b \delta_v \quad (C-37)$$

## C.6 Conclusion

En résumé, la mise à jour des poids à utiliser dans l'algorithme est définie par :

$$\begin{cases} W_{new} = W_{old} + \eta \delta_W [F_a (Ve + V_b)]^T \\ W_{b,new} = W_{b,old} + \eta_b \delta_W \\ V_{new} = V_{old} + \eta \delta_v [e]^T \\ V_{b,new} = V_{b,old} + \eta_b \delta_v \end{cases} \quad (C-38)$$

## Annexe D

### Modélisation de la Machine Asynchrone

Cette annexe est composée en deux parties : dans un premier temps nous rappellerons les équations générales de la machine asynchrone dans le repère  $d-q$  et dans le repère  $\alpha-\beta$ , puis dans un second temps nous examinerons la modélisation bond-graph de la machine.

#### D.1 Equations Générales de la Machine Asynchrone

L'étude physique d'une machine asynchrone conduit aux relations suivantes, écrites sous forme matricielle :

$$\begin{pmatrix} V_a \\ V_b \\ V_c \end{pmatrix}_s = R_s \begin{pmatrix} I_a \\ I_b \\ I_c \end{pmatrix}_s + \frac{d}{dt} \begin{pmatrix} \phi_a \\ \phi_b \\ \phi_c \end{pmatrix}_s \quad \text{et} \quad \begin{pmatrix} V_a \\ V_b \\ V_c \end{pmatrix}_r = R_r \begin{pmatrix} I_a \\ I_b \\ I_c \end{pmatrix}_r + \frac{d}{dt} \begin{pmatrix} \phi_a \\ \phi_b \\ \phi_c \end{pmatrix}_r \quad (\text{D-1})$$

$$\begin{pmatrix} \phi_a \\ \phi_b \\ \phi_c \end{pmatrix}_s = \begin{bmatrix} l_s & M_s & M_s \\ M_s & l_s & M_s \\ M_s & M_s & l_s \end{bmatrix} \begin{pmatrix} I_a \\ I_b \\ I_c \end{pmatrix}_s + m_{sr} \begin{bmatrix} \cos(p_0\theta) & \cos\left(p_0\theta + \frac{2\pi}{3}\right) & \cos\left(p_0\theta - \frac{2\pi}{3}\right) \\ \cos\left(p_0\theta - \frac{2\pi}{3}\right) & \cos(p_0\theta) & \cos\left(p_0\theta + \frac{2\pi}{3}\right) \\ \cos\left(p_0\theta + \frac{2\pi}{3}\right) & \cos\left(p_0\theta - \frac{2\pi}{3}\right) & \cos(p_0\theta) \end{bmatrix} \begin{pmatrix} I_a \\ I_b \\ I_c \end{pmatrix}_r \quad (\text{D-2})$$

$$\begin{pmatrix} \phi_a \\ \phi_b \\ \phi_c \end{pmatrix}_r = \begin{bmatrix} l_r & M_r & M_r \\ M_r & l_r & M_r \\ M_r & M_r & l_r \end{bmatrix} \begin{pmatrix} I_a \\ I_b \\ I_c \end{pmatrix}_r + m_{sr} \begin{bmatrix} \cos(p_0\theta) & \cos\left(p_0\theta - \frac{2\pi}{3}\right) & \cos\left(p_0\theta + \frac{2\pi}{3}\right) \\ \cos\left(p_0\theta + \frac{2\pi}{3}\right) & \cos(p_0\theta) & \cos\left(p_0\theta - \frac{2\pi}{3}\right) \\ \cos\left(p_0\theta - \frac{2\pi}{3}\right) & \cos\left(p_0\theta + \frac{2\pi}{3}\right) & \cos(p_0\theta) \end{bmatrix} \begin{pmatrix} I_a \\ I_b \\ I_c \end{pmatrix}_s \quad (\text{D-3})$$

$(V_a \ V_b \ V_c)$  : Tensions aux bornes des phases statoriques.

$(I_a \ I_b \ I_c)_s$  : Courants dans les enroulements triphasés statoriques.

$(\phi_a \ \phi_b \ \phi_c)_s$  : Flux magnétiques captés par les phases statoriques.

$(I_a \ I_b \ I_c)_r$  : Courants dans les enroulements triphasés rotoriques.

$(\phi_a \ \phi_b \ \phi_c)_r$  : Flux magnétiques captés par les phases rotoriques.

$R_s$  : Résistance statorique.

$l_s$  : Inductance statorique.

$R_r$  : Résistance rotorique.

$l_r$  : Inductance rotorique.

$M_s$  : Inductance mutuelle statorique.

$M_r$  : Inductance mutuelle rotorique.

$p_o$  : Nombre de paires de pôles.

$m_{sr}$  : Inductance mutuelle entre stator et rotor.

### D.1.1 Transformation de Park

La transformation de Park, appelée souvent transformation des deux axes, fait correspondre aux variables réelles leurs composantes homopolaires (indice 0), d'axe direct (indice  $d$ ) et d'axe en quadrature (indice  $q$ ). La matrice de transformation de Park sous sa forme orthogonale s'écrit :

$$P(\theta) = \sqrt{\frac{2}{3}} \begin{bmatrix} \frac{1}{\sqrt{2}} & \cos(\theta) & -\sin(\theta) \\ \frac{1}{\sqrt{2}} & \cos\left(\theta - \frac{2\pi}{3}\right) & -\sin\left(\theta - \frac{2\pi}{3}\right) \\ \frac{1}{\sqrt{2}} & \cos\left(\theta - \frac{4\pi}{3}\right) & -\sin\left(\theta - \frac{2\pi}{3}\right) \end{bmatrix} \quad (D-4)$$

Dans le cas d'un système de courants, elle s'écrit :

$$\begin{pmatrix} I_a \\ I_b \\ I_c \end{pmatrix} = P(\theta) \begin{pmatrix} I_0 \\ I_d \\ I_q \end{pmatrix} \quad (D-5)$$

Il est à noter que dans un système triphasé équilibré, la composante  $I_0$  est nulle.

La transformation de Park est appliquée sur le modèle mathématique de la machine asynchrone, avec un angle  $\theta_s$  sur les composantes du stator et un angle  $\theta_r$  sur les composantes du rotor. De plus, les angles  $\theta_s$ ,  $\theta_r$  sont liés par la relation suivante :

$$\theta_s = p_0 \theta + \theta_r \quad (\text{D-6})$$

on obtient alors pour le modèle :

$$\begin{aligned} \begin{pmatrix} V_d \\ V_q \end{pmatrix}_s &= R_s \begin{pmatrix} I_d \\ I_q \end{pmatrix}_s + \frac{d\theta_s}{dt} \begin{pmatrix} 0 & 1 \\ -1 & 0 \end{pmatrix} \begin{pmatrix} \phi_d \\ \phi_q \end{pmatrix}_s + \frac{d}{dt} \begin{pmatrix} \phi_d \\ \phi_q \end{pmatrix}_s \\ \begin{pmatrix} 0 \\ 0 \end{pmatrix}_r &= R_r \begin{pmatrix} I_d \\ I_q \end{pmatrix}_r + \frac{d\theta_r}{dt} \begin{pmatrix} 0 & -1 \\ 1 & 0 \end{pmatrix} \begin{pmatrix} \phi_d \\ \phi_q \end{pmatrix}_r + \frac{d}{dt} \begin{pmatrix} \phi_d \\ \phi_q \end{pmatrix}_r \\ \begin{pmatrix} \phi_d \\ \phi_q \end{pmatrix}_s &= L_s \begin{pmatrix} I_d \\ I_q \end{pmatrix}_s + M_{sr} \begin{pmatrix} I_d \\ I_q \end{pmatrix}_r \\ \begin{pmatrix} \phi_d \\ \phi_q \end{pmatrix}_r &= L_r \begin{pmatrix} I_d \\ I_q \end{pmatrix}_r + M_{sr} \begin{pmatrix} I_d \\ I_q \end{pmatrix}_s \end{aligned} \quad (\text{D-7})$$

avec :

$L_s$  : Inductance cyclique propre du stator définie par  $L_s = l_s - M_s$ .

$L_r$  : Inductance cyclique propre du rotor définie par  $L_r = l_r - M_r$ .

$M_{sr}$  : Inductance mutuelle cyclique propre du stator-rotor définie par  $M_{sr} = \frac{3}{2} m_{sr}$

### D.1.2 Equation d'état

Le modèle mathématique transformé de la machine, se déduit des relations suivantes:

$$\begin{aligned} I_{ds} &= \frac{L_r}{L_r L_s - M_{sr}^2} \phi_{ds} - \frac{M_{sr}}{L_r L_s - M_{sr}^2} \phi_{dr} \\ I_{qs} &= \frac{L_r}{L_r L_s - M_{sr}^2} \phi_{qs} - \frac{M_{sr}}{L_r L_s - M_{sr}^2} \phi_{qr} \\ I_{dr} &= \frac{L_s}{L_r L_s - M_{sr}^2} \phi_{dr} - \frac{M_{sr}}{L_r L_s - M_{sr}^2} \phi_{ds} \\ I_{qr} &= \frac{L_s}{L_r L_s - M_{sr}^2} \phi_{qr} - \frac{M_{sr}}{L_r L_s - M_{sr}^2} \phi_{qs} \end{aligned} \quad (\text{D-8})$$

en notant :

$$\alpha_s = \frac{L_s}{L_r L_s - M_{sr}^2} \quad \alpha_r = \frac{L_r}{L_r L_s - M_{sr}^2} \quad \beta = \frac{L_s}{L_r L_s - M_{sr}^2} \quad (D-9)$$

On obtient alors en prenant les flux comme variable d'état l'équation suivante :

$$\begin{cases} \frac{d\phi_{ds}}{dt} = -R_s \alpha_r \phi_{ds} + R_s \beta \phi_{dr} - \frac{d\theta_s}{dt} \phi_{qs} + u_d \\ \frac{d\phi_{qs}}{dt} = -R_s \alpha_r \phi_{qs} + R_s \beta \phi_{qr} + \frac{d\theta_s}{dt} \phi_{ds} + u_q \\ \frac{d\phi_{dr}}{dt} = -R_r \alpha_s \phi_{dr} + R_r \beta \phi_{ds} + \frac{d\theta_r}{dt} \phi_{qr} \\ \frac{d\phi_{qr}}{dt} = -R_r \alpha_s \phi_{qr} + R_r \beta \phi_{qs} - \frac{d\theta_r}{dt} \phi_{dr} \end{cases} \quad (D-10)$$

A présent, il nous faut résoudre  $\theta_s = p_0 \theta + \theta_r$ . Parmi toutes les solutions satisfaisant cette équation, deux sont particulièrement intéressantes.

#### D.1.2.1 Axes de référence fixes par rapport au rotor

Ce repère est appelé  $\alpha$ - $\beta$ . L'équation d'état est obtenu en considérant  $\theta_s = 0$  et  $\theta_r = -p_0 \theta$ .

$$\begin{cases} \frac{d\phi_{\alpha s}}{dt} = -R_s \alpha_r \phi_{\alpha s} + R_s \beta \phi_{\alpha r} + u_\alpha \\ \frac{d\phi_{\beta s}}{dt} = -R_s \alpha_r \phi_{\beta s} + R_s \beta \phi_{\beta r} + u_\beta \\ \frac{d\phi_{\alpha r}}{dt} = -R_r \alpha_s \phi_{\alpha r} + R_r \beta \phi_{\alpha s} - p_o \omega \phi_{\beta r} \\ \frac{d\phi_{\beta r}}{dt} = -R_r \alpha_s \phi_{\beta r} + R_r \beta \phi_{\beta s} + p_o \omega \phi_{\alpha r} \end{cases} \quad (D-11)$$

Avec  $\omega = \frac{d\theta}{dt}$  la vitesse du rotor.

### D.1.2.2 Axes de référence tournant au synchronisme

Ce repère est appelé  $d$ - $q$ . Dans ce cas, on considère  $\theta_s = \omega_s t$  et  $\theta_r = \omega_s t - p_0 \theta$ , avec  $\omega_s$  la vitesse synchrone de la machine. Cette vitesse est définie par  $\omega_s = \frac{\omega_c}{p_0}$ , avec  $\omega_c$  pulsation des tensions d'alimentation du stator. L'équation d'état est alors la suivante :

$$\begin{cases} \frac{d\phi_{ds}}{dt} = -R_s \alpha_r \phi_{ds} + R_s \beta \phi_{dr} - \omega_s \phi_{qs} + u_d \\ \frac{d\phi_{qs}}{dt} = -R_s \alpha_r \phi_{qs} + R_s \beta \phi_{qr} + \omega_s \phi_{ds} + u_q \\ \frac{d\phi_{dr}}{dt} = -R_r \alpha_s \phi_{dr} + R_r \beta \phi_{ds} - \omega_s \phi_{qr} - p_0 \omega \phi_{qr} \\ \frac{d\phi_{qr}}{dt} = -R_r \alpha_s \phi_{qr} + R_r \beta \phi_{qs} + \frac{d\theta_r}{dt} \phi_{dr} + p_0 \omega \phi_{dr} \end{cases} \quad (D-12)$$

### D.1.3 Relation entre les composantes de Concordia et celles de Park

Désignons par  $I_\alpha$  et  $I_\beta$  les composantes symétriques en courant de Concordia dans le repère  $\alpha$ - $\beta$ . La relation entre les composantes du repère  $d$ - $q$  et le repère  $\alpha$ - $\beta$  est définie par la relation suivante :

$$\begin{bmatrix} I_d \\ I_q \end{bmatrix} = \begin{bmatrix} \cos(\theta) & \sin(\theta) \\ -\sin(\theta) & \cos(\theta) \end{bmatrix} \begin{bmatrix} I_\alpha \\ I_\beta \end{bmatrix} \quad (D-13)$$

### D.1.4 Calcul du couple

La relation générale du couple pour la machine asynchrone s'écrit sous la forme :

$$C_{em} = \frac{1}{2} \begin{bmatrix} I_s^T & I_r^T \end{bmatrix} \left( \frac{d}{d\theta} \begin{bmatrix} L_s & M \\ M^T & L_r \end{bmatrix} \right) \begin{bmatrix} I_s \\ I_r \end{bmatrix} \quad (D-14)$$

avec

$$I_s = \begin{pmatrix} I_a \\ I_b \\ I_c \end{pmatrix}_s, \quad I_r = \begin{pmatrix} I_a \\ I_b \\ I_c \end{pmatrix}_r, \quad L_s = \begin{bmatrix} l_s & M_s & M_s \\ M_s & l_s & M_s \\ M_s & M_s & l_s \end{bmatrix}, \quad L_r = \begin{bmatrix} l_r & M_r & M_r \\ M_r & l_r & M_r \\ M_r & M_r & l_r \end{bmatrix} \quad (D-15)$$

$$M = \begin{bmatrix} \cos(p_0\theta) & \cos\left(p_0\theta + \frac{2\pi}{3}\right) & \cos\left(p_0\theta - \frac{2\pi}{3}\right) \\ \cos\left(p_0\theta - \frac{2\pi}{3}\right) & \cos(p_0\theta) & \cos\left(p_0\theta + \frac{2\pi}{3}\right) \\ \cos\left(p_0\theta + \frac{2\pi}{3}\right) & \cos\left(p_0\theta - \frac{2\pi}{3}\right) & \cos(p_0\theta) \end{bmatrix} \quad (\text{D-16})$$

Comme les matrices  $L_s$  et  $L_r$  sont constantes, et  $M$  est fonction de  $\theta$ , il vient :

$$C_{em} = \frac{1}{2} I_s^T \left( \frac{d}{d\theta} M \right) I_r \quad (\text{D-17})$$

En effectuant un changement de variable, avec les transformations de Park aux axes  $d$ - $q$ , on obtient alors :

$$C_{em} = p_0 M \left( I_{dr} I_{qs} - I_{qr} I_{ds} \right) \quad (\text{D-18})$$

En terme de variables associées au rotor, cette équation peut s'écrire de la manière suivante:

$$C_{em} = p_0 \left( I_{dr} \phi_{qr} - I_{qr} \phi_{dr} \right) \quad (\text{D-19})$$

## D.2 Analyse de la Modélisation Bond Graph

Rappelons la représentation bond graph du chapitre IV de l'association onduleur moteur asynchrone :

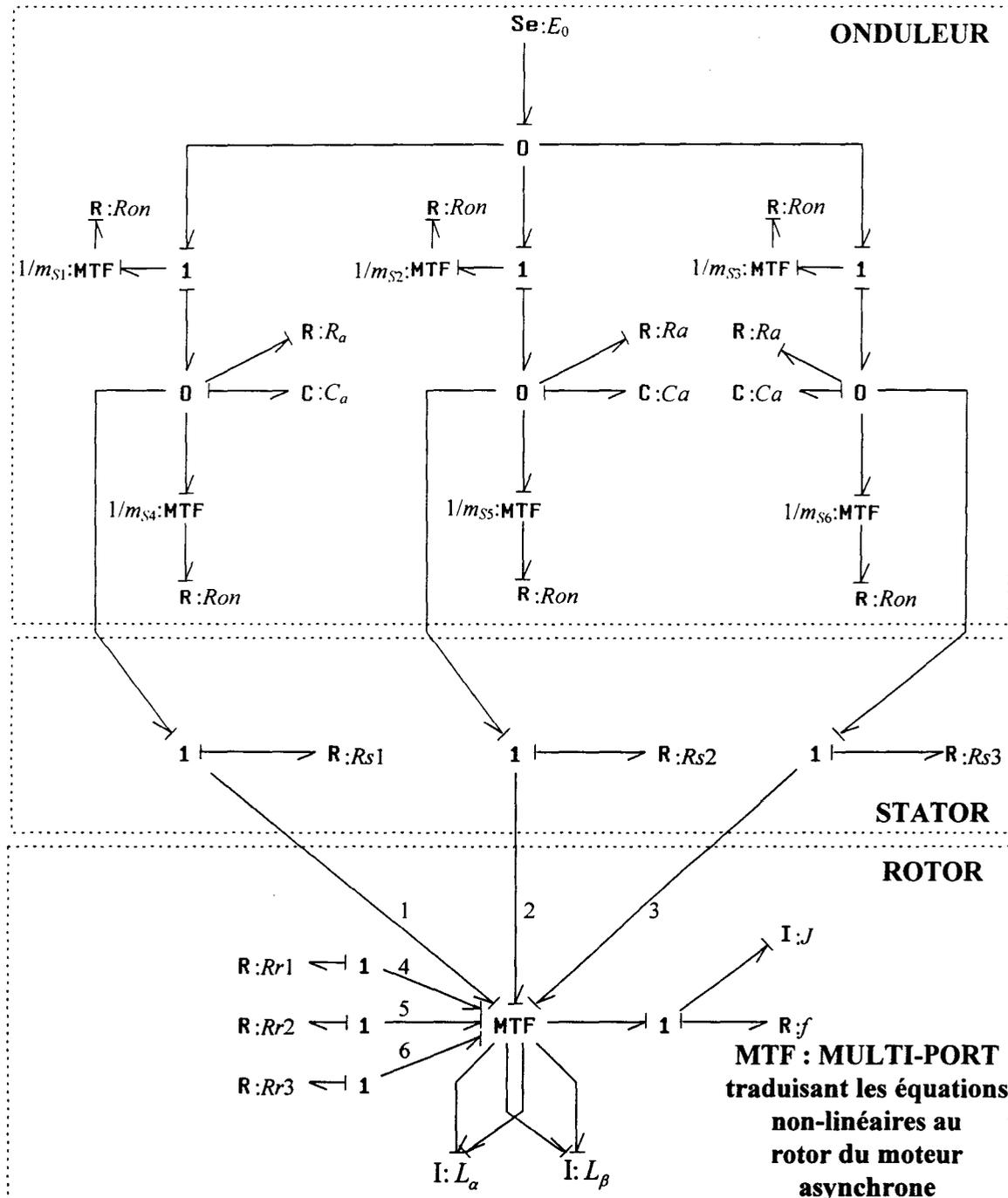


Figure D-1 : Bond-Graph d'une Association onduleur-moteur asynchrone.

Les flux et les courants au stator et au rotor sont liés par la relation :

$$\phi = Li \quad (D-20)$$

avec  $\phi = (\phi_1, \phi_2, \phi_3, \phi_4, \phi_5, \phi_6)^T$ ,  $i = (i_1, i_2, i_3, i_4, i_5, i_6)^T$ , et la matrice d'inductance

$$L = \begin{bmatrix} l_s & m_s & m_s & m_{sr1} & m_{sr3} & m_{sr2} \\ m_s & l_s & m_s & m_{sr2} & m_{sr1} & m_{sr3} \\ m_s & m_s & l_s & m_{sr3} & m_{sr2} & m_{sr1} \\ m_{rs1} & m_{rs2} & m_{rs3} & l_r & m_r & m_r \\ m_{rs3} & m_{rs1} & m_{rs2} & m_r & l_r & m_r \\ m_{rs2} & m_{rs3} & m_{rs1} & m_r & m_r & l_r \end{bmatrix} \quad (D-21)$$

$$l_s, m_s, l_r, m_r \text{ constant et } m_{sr1} = m_{rs1} = m \cos(\theta), m_{sr2} = m_{rs2} = m \cos\left(\theta + \frac{2\pi}{3}\right), m_{sr3} = m_{rs3} = m \cos\left(\theta + \frac{4\pi}{3}\right).$$

Les variables flux (courants) au sens bond-graph seront notées  $f_i$  et les variables effort (tensions)  $e_i$ .

D'autre part les courants et le couple  $C_{em}$  sont liés par :

$$C_{em} = \frac{1}{2} i^T \frac{dL}{d\theta} i = (f_2 f_4 - f_1 f_3) \quad (D-22)$$

Les équations du bond-graph donnent pour les condensateurs :

$$\begin{cases} \frac{dq_1}{dt} = \left( -\frac{1}{R_a C_a} - \frac{m_1^2}{Ron C_a} - \frac{m_4^2}{Ron C_a} \right) q_1 - f_1 + \frac{m_1^2}{Ron} E_0 \\ \frac{dq_2}{dt} = \left( -\frac{1}{R_a C_a} - \frac{m_2^2}{Ron C_a} - \frac{m_5^2}{Ron C_a} \right) q_2 - f_2 + \frac{m_2^2}{Ron} E_0 \\ \frac{dq_3}{dt} = \left( -\frac{1}{R_a C_a} - \frac{m_3^2}{Ron C_a} - \frac{m_6^2}{Ron C_a} \right) q_3 - f_3 + \frac{m_3^2}{Ron} E_0 \end{cases} \quad (D-23)$$

et pour l'inertie du rotor :

$$\begin{cases} \frac{d\theta}{dt} = \omega \\ \frac{d\omega}{dt} = -\frac{f}{J} \omega + \frac{C_{em}}{J} \end{cases} \quad (D-24)$$

Les flux relatifs au stator sont décrits par :

$$\begin{cases} \frac{d\phi}{dt} = -\frac{2}{3} R_{s1} f_1 + \frac{1}{3} R_{s2} f_2 + \frac{1}{3} R_{s3} f_3 + \frac{2}{3} \frac{q_1}{C_a} - \frac{1}{3} \frac{q_2}{C_a} - \frac{1}{3} \frac{q_3}{C_a} \\ \frac{d\phi_2}{dt} = -\frac{2}{3} R_{s2} f_2 + \frac{1}{3} R_{s1} f_1 + \frac{1}{3} R_{s3} f_3 + \frac{2}{3} \frac{q_2}{C_a} - \frac{1}{3} \frac{q_1}{C_a} - \frac{1}{3} \frac{q_3}{C_a} \\ \frac{d\phi_3}{dt} = -\frac{2}{3} R_{s3} f_3 + \frac{1}{3} R_{s2} f_2 + \frac{1}{3} R_{s1} f_1 + \frac{2}{3} \frac{q_3}{C_a} - \frac{1}{3} \frac{q_2}{C_a} - \frac{1}{3} \frac{q_1}{C_a} \end{cases} \quad (\text{D-25})$$

et, au rotor :

$$\begin{cases} \frac{d\phi_4}{dt} = -R_r f_4 \\ \frac{d\phi_5}{dt} = -R_r f_5 \\ \frac{d\phi_6}{dt} = -R_r f_6 \end{cases} \quad (\text{D-26})$$

Les équations d'état simplifiées sont obtenues après l'application de la méthode des perturbations singulières ( $\frac{dq_i}{dt} = 0$ ) pour éliminer les dynamiques rapides des éléments  $C_a$ .



## Annexe E

### Existence du Régime de Glissement

La commande équivalente est déterminée à partir de la dérivée de la surface  $s$  :

$$\dot{s}(x) = \begin{pmatrix} \dot{s}_1(x_1) \\ \dot{s}_2(x_2) \end{pmatrix} = \begin{pmatrix} -R_s \alpha_r x_1 + R_s \beta x_3 + \phi_1 \dot{\theta}_s(t) \sin(\theta_s(t)) + \phi_2 \dot{\theta}_s(t) \cos(\theta_s(t)) \\ -R_s \alpha_r x_2 + R_s \beta x_4 - \phi_1 \dot{\theta}_s(t) \cos(\theta_s(t)) + \phi_2 \dot{\theta}_s(t) \sin(\theta_s(t)) \end{pmatrix} + u_{de} \quad (\text{E-1})$$

soit sur l'intersection des surfaces de glissement (cf. chapitre IV) :

$$u_{de} = \begin{pmatrix} -R_s \beta x_3 - \sin(\theta_s(t))(R_s \alpha_r \phi_2 + \phi_1 \dot{\theta}_s(t)) + \cos(\theta_s(t))(R_s \alpha_r \phi_1 - \phi_2 \dot{\theta}_s(t)) \\ -R_s \beta x_4 + \sin(\theta_s(t))(R_s \alpha_r \phi_1 - \phi_2 \dot{\theta}_s(t)) + \cos(\theta_s(t))(R_s \alpha_r \phi_2 + \phi_1 \dot{\theta}_s(t)) \end{pmatrix} \quad (\text{E-2})$$

La définition de la commande équivalente  $u_{de} = \begin{pmatrix} u_\alpha \\ u_\beta \end{pmatrix}$  vérifie, pour  $x \in s$ , l'inégalité

suivante (Annexe B) :

$$\min(u^-(x), u^+(x)) < u_{de}(x) < \max(u^-(x), u^+(x)) \quad (\text{E-3})$$

nous avons alors :

$$-R_s \beta x_3 - (R_s \alpha_r |\phi_2| + \phi_1 \dot{\theta}_s(t)) - (R_s \alpha_r \phi_1 - |\phi_2| \dot{\theta}_s(t)) < u_\alpha < -R_s \beta x_3 + (R_s \alpha_r |\phi_2| + \phi_1 \dot{\theta}_s(t)) + (R_s \alpha_r \phi_1 - |\phi_2| \dot{\theta}_s(t)) \quad (\text{E-4})$$

et :

$$-R_s \beta x_4 - (R_s \alpha_r \phi_1 - \phi_2 \dot{\theta}_s(t)) - (R_s \alpha_r \phi_2 + \phi_1 \dot{\theta}_s(t)) < u_\beta < -R_s \beta x_4 + (R_s \alpha_r \phi_1 - \phi_2 \dot{\theta}_s(t)) + (R_s \alpha_r \phi_2 + \phi_1 \dot{\theta}_s(t)) \quad (\text{E-5})$$

