

10/99

REPUBLIQUE ALGERIENNE DEMOCRATIQUE ET POPULAIRE

MINISTERE DE L'ENSEIGNEMENT SUPERIEUR
ET DE LA RECHERCHE SCIENTIFIQUE

ECOLE NATIONALE POLYTECHNIQUE

DER de : GENIE ELECTRIQUE ET INFORMATIQUE

Filière : AUTOMATIQUE

المدرسة الوطنية المتعددة التقنيات
BIBLIOTHEQUE — المكتبة
Ecole Nationale Polytechnique

Thèse de projet de
fin d'étude

THEME

*SUR LES DIFFERENTS TYPES DE
COMMANDES ET OBSERVATEURS D'UN
SIMULATEUR DE VOL D'HELICOPTERE*

Etudié par :

Djeridane Badis
Djeddou Mohamed

Dirigé par :

M. Tadjine
F. Boudjema

Promotion : juin 1999

E.N.P. 10 avenue Hassen Badi EL-HARRACH ALGER



المدرسة الوطنية المتعددة التقنيات
المكتبة — BIBLIOTHEQUE
Ecole Nationale Polytechnique



Dédicace

- *A mes très chers parents.*
- *A mon frère Yassine.*
- *A mon cousin Adel.*
- *A mes chers amis.*

Dédicace

- *A ma mère.*
- *A mon père.*
- *A mes frères et mes soeurs.*
- *A tous mes amis.*

Mohamed

Remerciement

Nous aimerons remercier Mr M.Tadjine et Mr F.Boudjema qui ont toujours été prêts à donner de leurs temps généreusement pour discuter sur les idées et les approches développées, et aussi pour leurs conseils précieux. Nous aimerons aussi remercier tous les enseignants du Département d'Automatique, et sans oublier Mr. SALAH pour l'aide qu'il nous a fourni dans la recherche bibliographique.

SOMMAIRE



Introduction Générale	2
Chapitre 1 Modélisation	
1.1 Introduction	5
1.2 Modélisation de la dynamique d'élévation	5
1.3 Modélisation de la dynamique azimutale	6
1.4 Modélisation du MCC et de l'hélice	7
1.5 Schéma complet du système	9
1.6 Simplification du modèle	10
1.7 Modèle d'état du simulateur d'hélicoptère	11
1.8 Essai en boucle ouverte	12
Chapitre 2 Commandes quantitatives	
2.1 Commande avec linéarisation par retour d'état	14
2.1.1 Introduction	14
2.1.2 Concepts intuitifs	14
2.1.2.1 Linéarisation par retour d'état et forme canonique	14
2.1.2.2 Cas d'un système sous forme canonique	16
2.1.2.3 Linéarisation entrée-sortie	17
2.1.3 Génération d'une relation entrée-sortie	19
2.1.4 Systèmes à plusieurs entrées	20
2.1.5 Linéarisation par retour d'état pour les systèmes MIMO	20
2.2 Commande à structure variable	21
2.2.1 Introduction	21
2.2.2 Condition d'existence du mode de glissement	22
2.2.3 Surface de commutation	23
2.2.4 Détails sur la CSV pour les système mono-entrée	23
2.2.4.1 Forme canonique de commande	23
2.2.4.2 Détails sur la surface de commutation	24
2.2.5 Procédure de conception	24
2.2.5.1 Détermination de la surface de commutation	24
2.2.5.1.2 Commande équivalente	26

2.2.5.2 Développement de la loi de commande	26
2.3 Commande par « Backstepping »	28
2.3.1 Introduction	28
2.3.2 Présentation de la méthode	28
2.3.3 « Backstepping » pour les systèmes avec termes incertains	34
Chapitre 3	Commande qualitative
3.1 Introduction	39
3.2 Ensembles flous	40
3.2.1 Variable linguistique	40
3.2.2 Fonctions d'appartenance	41
3.2.3 Opérations sur les ensembles flous	42
3.3 Logique floue	44
3.4 Règles Si-Alors	45
3.5 Raisonnement flou	46
3.6 Réglage par logique floue	48
3.6.1 Structure d'un réglage par logique floue	48
3.6.2 Configuration interne d'un régulateur par logique floue	49
Chapitre 4	Observateurs non linéaires
4.1 Eléments de la théorie de l'observation	52
4.2 Systemes uniformément observables	53
4.3 Synthèses d'observateurs pour les systèmes non linéaires	54
4.3.1 Position du problème	54
4.3.2 Observateur de Thau	55
4.3.3 Observateur glissant	56
4.3.3.1 Extension au cas général des systèmes non linéaires	57
4.3.4 Observateur basé sur la méthode du « Backstepping »	58
4.4 Stabilisation via un observateur	60
4.4.1 Position du problème	60
4.4.2 Stabilisation exponentielle globale	61
Chapitre 5	Application au simulateur d'hélicoptère
5.1 Commande avec linéarisation par retour d'état	63
5.1.1 Application au premier sous système	64

5.1.2 Application au second sous système	65
5.1.3 Résultats de simulation	66
5.2 Commande à structure variable	70
5.2.1 Application	70
5.2.2 Résultats de simulation	73
5.3 Commande par la méthode du «Backstepping »	79
5.3.1 Application	79
5.3.2 Résultats de simulation	81
5.4 Commande par logique floue	85
5.4.1 Choix des variables d'entrée et de sortie	85
5.4.2 Normalisation	85
5.4.3 Régulateur du type Mamdani	86
5.4.4 Matrice de Macvicar Whelan	86
5.4.5 Synthèse des régulateurs	87
5.4.5.1 Régulateur à 3 classes	87
5.4.5.2 Régulateur à 5 classes	87
5.4.6 Résultats de simulation	88
5.5 Observateur de Thau	94
5.5.1 Résultats de simulation	95
5.6 Observateur glissant	98
5.6.1 Résultats de simulation	100
5.7 Observateur à base de « Backstepping »	103
5.7.1 Dynamique de l'erreur	103
5.7.2 Résultats de simulation	105
5.8 Stabilité du compensateur dynamique	107
5.8.1 Résultats de simulation	107
5.9 Conclusion	107
5.10 Etude comparative	110
5.10.1 Etude comparative entre les commandes appliquées	110
5.10.2 Etude comparative entre les observateurs appliqués	111
5.11 Animation	112
Conclusion générale	115
Références bibliographiques	118
Annexe	120

المدرسة الوطنية المتعددة التقنيات
BIBLIOTHEQUE — المكتبة
Ecole Nationale Polytechnique

*INTRODUCTION
GENERALE*

Introduction Générale

Pendant des décennies, des chercheurs de tous les domaines s'intéressaient aux systèmes aéronautiques. Comme l'automatique est une science multidisciplinaire par excellence, les chercheurs automaticiens se sont penchés sur le problème de la commande des avions et des hélicoptères. Mais cette attirance vers l'aéronautique peut s'expliquer par plusieurs raisons, l'une de ces raisons est que les systèmes aéronautiques sont devenus très complexes et donc difficiles à commander manuellement.

Souvent, les lois de commande des systèmes dynamiques nécessitent la connaissance totale ou partielle de l'état du système à commander. D'un point de vue pratique, la connaissance de l'état s'obtient grâce à des mesures effectuées avec des capteurs. Ces mesures sont souvent entachées de bruit. Cet inconvénient limite les performances d'une boucle de commande. Il est aussi parfois difficile de reconstituer l'état du système par déduction algébrique. Pour ces différentes raisons, nous faisons alors appel à la théorie des observateurs pour reconstituer l'état des systèmes dynamiques.

Dans ce travail, nous nous sommes intéressés à la synthèse de différentes stratégies de commande à base d'observateur pour stabiliser un simulateur de vol d'hélicoptère. Ce simulateur est un système multivariable à deux entrées, deux sorties caractérisé par une forte non linéarité, un couplage important et une dynamique instable en boucle ouverte.

Tout d'abord, les techniques de synthèse quantitatives, telles que Linéarisation exacte, Mode glissant et le « Backstepping » sont utilisées. Ensuite la technique de synthèse qualitative, à savoir la commande par logique floue, est utilisée pour montrer les différences existantes entre les approches de synthèse. Par ailleurs, plusieurs structures d'observateurs sont développées pour l'estimation des variables non mesurables dans notre application. Ces observateurs sont tous non linéaires, observateur de Thau, observateur à mode glissant et observateur à base de Backstepping.

Ce mémoire est décomposé comme suit :

Le premier chapitre est consacré à la modélisation de notre système à l'aide de la mécanique analytique, et la mise sous forme d'équations d'état.

Dans le deuxième chapitre, nous allons présenter les notions théoriques sur lesquelles se basent les différentes commandes quantitatives utilisées dans notre étude. On présente alors, la commande avec linéarisation par retour d'état, la commande à structure variable et la commande par Backstepping, avec des exemples illustratifs.

Le chapitre trois donne une base théorique (logique floue) pour l'unique commande qualitative utilisée dans ce mémoire.

Dans le quatrième chapitre, nous allons exposer la base théorique des observateurs utilisés, tels que l'observateur de Thau, l'observateur glissant et l'observateur Backstepping. A noter que toutes ces approches sont basées sur la recherche d'une fonction de Lyapunov pour le calcul des gains de l'observateur et l'étude de la stabilité.

Le cinquième chapitre présente les résultats des simulations effectuées à l'aide de d'un calculateur numérique. Les calculs analytiques des commandes et des observateurs ainsi que leurs commentaires sont aussi exposés. Une étude comparative entre les différentes structures de commandes est alors faite.

Le mémoire se termine par une conclusion générale et des perspectives.



Chapitre 1

MODELISATION

1.1 Introduction

La modélisation de notre système en détail conduit à un modèle compliqué et difficile à utiliser. Nous devons décider pour quel fin ce modèle sera utilisé et dans quelles conditions il va fonctionner. Dans notre cas le modèle doit être utilisé pour décrire les dynamiques du système en respectant les tâches de commande. Le système doit fonctionner dans des conditions précises et donc les dynamiques du système ne vont pas être toutes invoquées. Ce qui conduit à une simplification du modèle. [6]

1.2 Modélisation de la dynamique d'élévation ψ

La figure (1.1) est une représentation graphique du simulateur de vol avec les forces agissantes sur chaque partie.

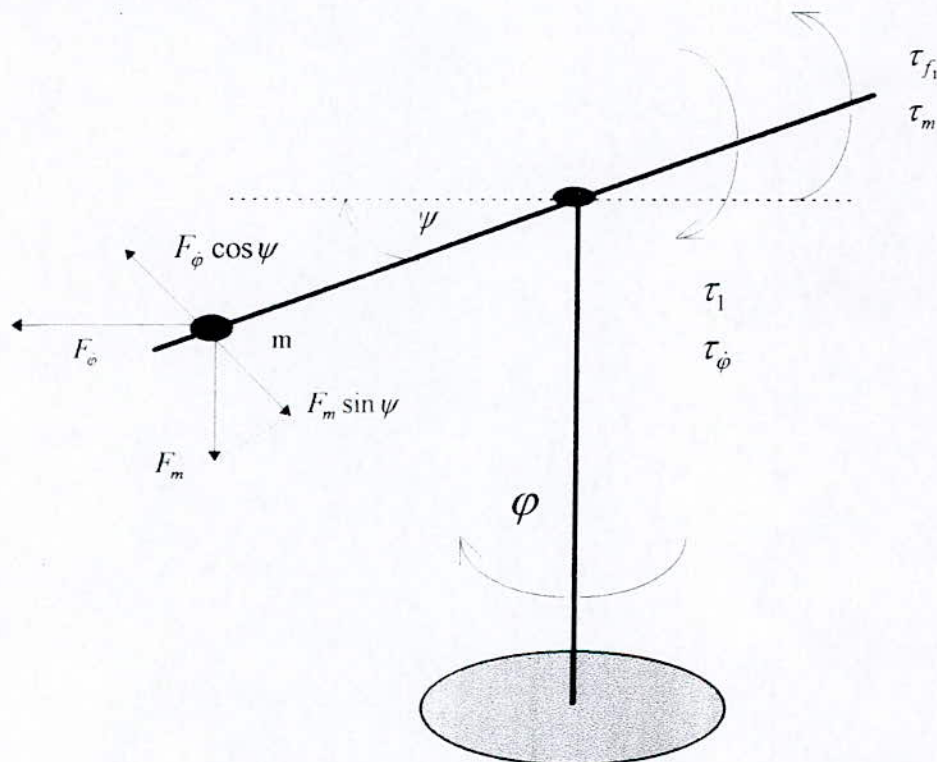


Figure 1.1 : Représentation des forces agissantes sur le système.

En écrivant dans le plan vertical, le bilan des moments agissants sur le corps de l'hélicoptère et d'après la loi de Newton, on a :

$$I_{\psi} \cdot \ddot{\psi} = \tau_1 + \tau_{\phi} - \tau_{f_1} - \tau_m + \tau_G \quad (1.1)$$

avec :

τ_G	Moment gyroscopique.	[N.m]
τ_m	Moment gravitationnel.	[N.m]
τ_{f_1}	Moment de viscosité.	[N.m]
τ_1	Moment d'élévation.	[N.m]
I_ψ	Moment d'inertie de l'hélicoptère suivant l'axe horizontal.	[N.m]

Les différents moments sont donnés par :

$$\tau_m = F_m \cdot l \cdot \sin \psi = m \cdot g \cdot l \cdot \sin \psi = \tau_1 \cdot \sin \psi \quad (1.2)$$

$$\tau_\phi = m \cdot l \cdot \dot{\phi}^2 \cdot \sin \psi \cdot \cos \psi = (1/2) \cdot m \cdot l \cdot \dot{\phi}^2 \cdot \sin 2\psi \quad (1.3)$$

$$\tau_1 = k_{\omega_1} \cdot \omega_1^2 \quad (1.4)$$

$$\tau_{f_1} = C_\psi \cdot \text{sign} \psi + B_\psi \cdot \psi \quad (1.5)$$

$$\tau_G = k_G \cdot \dot{\phi} \cdot \omega_1 \cdot \cos \psi \quad \dot{\phi} \ll \omega_1 \quad (1.6)$$

Quelques dynamiques sont négligées, par exemple : la variation de la résistance de l'air qui dépend de la vitesse de rotation de l'hélice principale, l'influence du moteur secondaire sur l'angle d'élévation, et l'influence de la vitesse du moteur principal sur le moment de friction qui est très difficile à modéliser analytiquement. Le couple gyroscopique est considéré en supposant que $\dot{\phi} \ll \omega_1$.

1.3 Modélisation de la dynamique azimutale ϕ

La figure 1.2 présente une vue de dessus du système avec les moments agissants sur le corps principal et leurs sens d'action. Ces moments sont :

τ_2	Moment du moteur stabilisateur.	[N.m]
τ_r	Moment de réaction du moteur principal.	[N.m]
τ_{f_2}	Moment de viscosité.	[N.m]

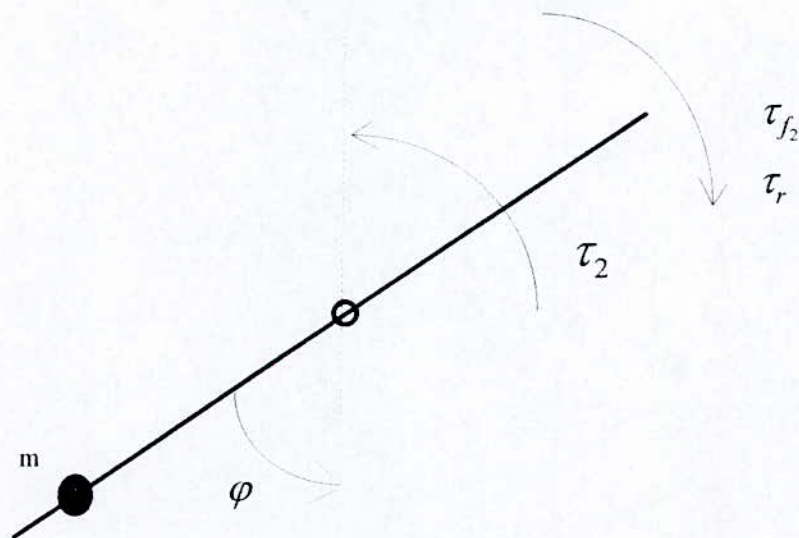


Figure 1.2 : Vue de dessus du système

L'équation (1.7) donne le bilan des moments suivant le plan horizontal, en tenant compte des forces les plus importantes agissant sur l'hélicoptère dans la direction de l'angle φ :

$$I_{\varphi} \cdot \ddot{\varphi} = \tau_2 - \tau_{f_2} - \tau_r \quad (1.7)$$

avec :

$$I_{\varphi} = I \cdot \sin \psi \quad (1.8)$$

$$\tau_2 = k_2 \cdot I_2 \cdot \sin \psi \cdot \omega_2^2 \quad (1.9)$$

$$\tau_{f_2} = C_{\varphi} \cdot \text{sign} \varphi + B_{\varphi} \cdot \dot{\varphi} \quad (1.10)$$

1.4 Modélisation du moteur à courant continu MCC et de l'hélice

Le modèle du MCC va être établi en tenant compte des considérations suivantes :

- Les inductances des armatures sont faibles.
- Le frottement de Coulomb et le moment résistif généré par l'hélice dans l'air sont pris en considération.
- Le moment généré par l'hélice dépend de ω et ω^2 .

On a les équations suivantes :

$$i = (u - K_b \cdot \omega) / R \quad (1.11.a)$$

$$\tau = K_t \cdot i \quad (1.11.b)$$

$$\tau_c = C \cdot \text{sign}(\omega) \quad (1.11.c)$$

$$\tau_p = B_p \cdot \omega + D_p \cdot \omega^2 \quad (1.11.d)$$

$$I \cdot \dot{\omega} = \tau - \tau_c - B \cdot \omega - \tau_p \quad (1.11.e)$$

où :

u	tension d'entrée (commande).	R	résistance d'armature.
i	courant d'armature.	K_b	constante de la fem de retour.
u_b	fem de retour.	K_t	constante de moment électrique.
ω	vitesse angulaire du rotor.	I	moment d'inertie du rotor avec l'hélice
τ	moment d'inertie.	B	coefficient de viscosité.
τ_c	moment de frottement.	C	coefficient de Coulomb
τ_p	moment de résistance de l'air.	B_p	coefficient de la résistance de l'air.

Les sorties sont : l'angle d'élévation ψ et l'angle d'azimut φ , et les entrées sont u_1 et u_2 qui sont respectivement la tension de commande du moteur principal et du moteur secondaire. Le problème de commande consiste à stabiliser les angles ψ et φ autour de leurs références ψ_d, φ_d . Notre modèle est multivariable à deux entrées, et deux sorties MIMO 2×2 non linéaire et fortement couplé. La figure (1.3) donne une représentation du système.

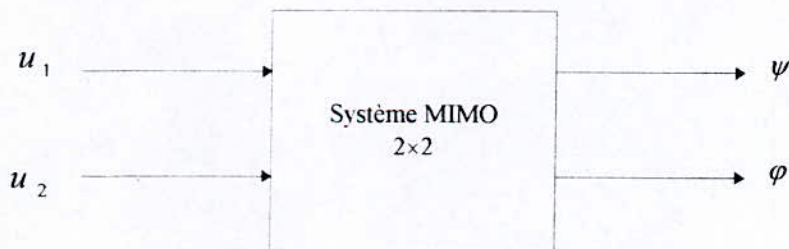


Figure 1.3 : Représentation du Simulateur

1.6 Simplification du modèle

Il est plus judicieux de simplifier notre modèle, à cet effet on a recouru à la technique d'identification block par block, qui est exposée dans le manuel du modèle du simulateur de vol de l'hélicoptère.

Dans [6], des simplifications d'ordre structurel, ainsi que l'identification de certains paramètres du simulateur d'hélicoptère, qui peut être fidèlement représenté par le modèle suivant :

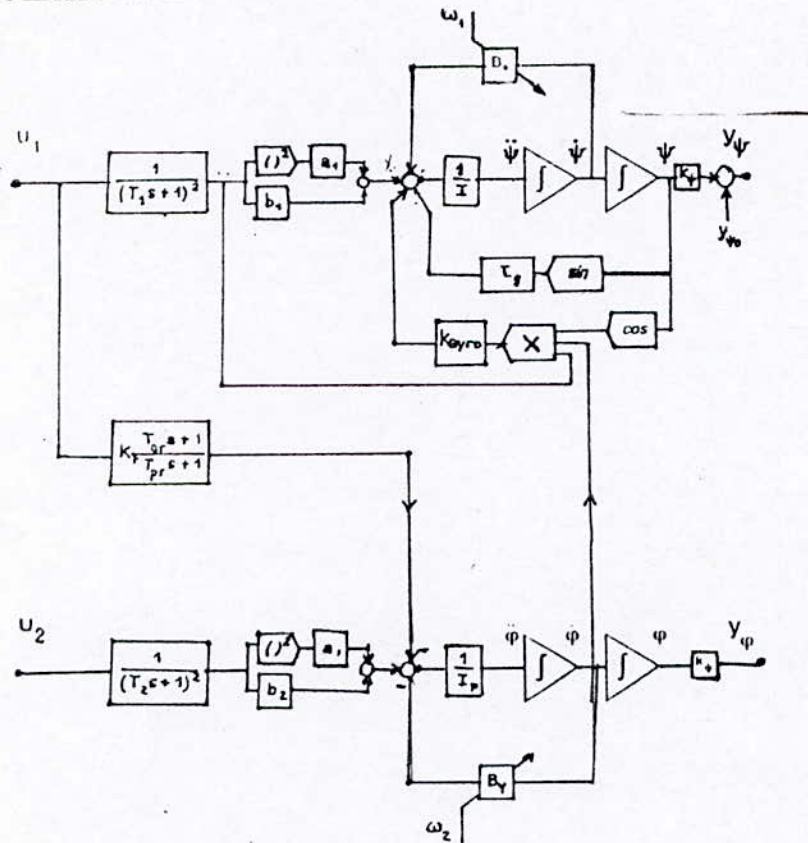


Figure 1.5 : Modèle empirique

Ce modèle est caractérisé par des parties linéaires et une partie non linéaire. Par suite, ce modèle est utilisé pour la synthèse des différentes stratégies de commande.

1.7 Modèle d'état du simulateur d'hélicoptère [6]

En fonction du choix des variables d'état, plusieurs modèles d'état du simulateur d'hélicoptère, peuvent être développés. Dans notre cas, voici les grandeurs physiques choisies comme variables d'état :

$$x_1 = \psi, \quad x_3 = \omega_1, \quad x_5 = \varphi, \quad x_7 = \omega_2 \quad (1.12.a)$$

$$x_2 = \dot{\psi}, \quad x_4 = \dot{\omega}_1, \quad x_6 = \dot{\varphi}, \quad x_8 = \dot{\omega}_2 \quad (1.12.b)$$

En faisant quelques substitutions, et en supposant que la fonction de transfert du couplage est réduite à un coefficient constant, on aboutit au système d'équations suivant :

$$\begin{aligned} \dot{x}_1 &= x_2 \\ \dot{x}_2 &= -\alpha_1 x_2 - \alpha_2 \sin x_1 + \alpha_3 x_3^2 + \alpha_4 x_3 - \beta x_3 x_6 \cos x_1 \\ \dot{x}_3 &= x_4 \\ \dot{x}_4 &= -\alpha_6 x_3 - \alpha_5 x_4 + \alpha_6 u_1 \\ \dot{x}_5 &= x_6 \\ \dot{x}_6 &= -\alpha_9 x_6 + \alpha_7 x_7^2 + \alpha_8 x_7 - \alpha_{10} u_1 \\ \dot{x}_7 &= x_8 \\ \dot{x}_8 &= -\alpha_{11} x_8 - \alpha_{12} x_7 + \alpha_{12} u_2 \end{aligned} \quad (1.13)$$

avec :

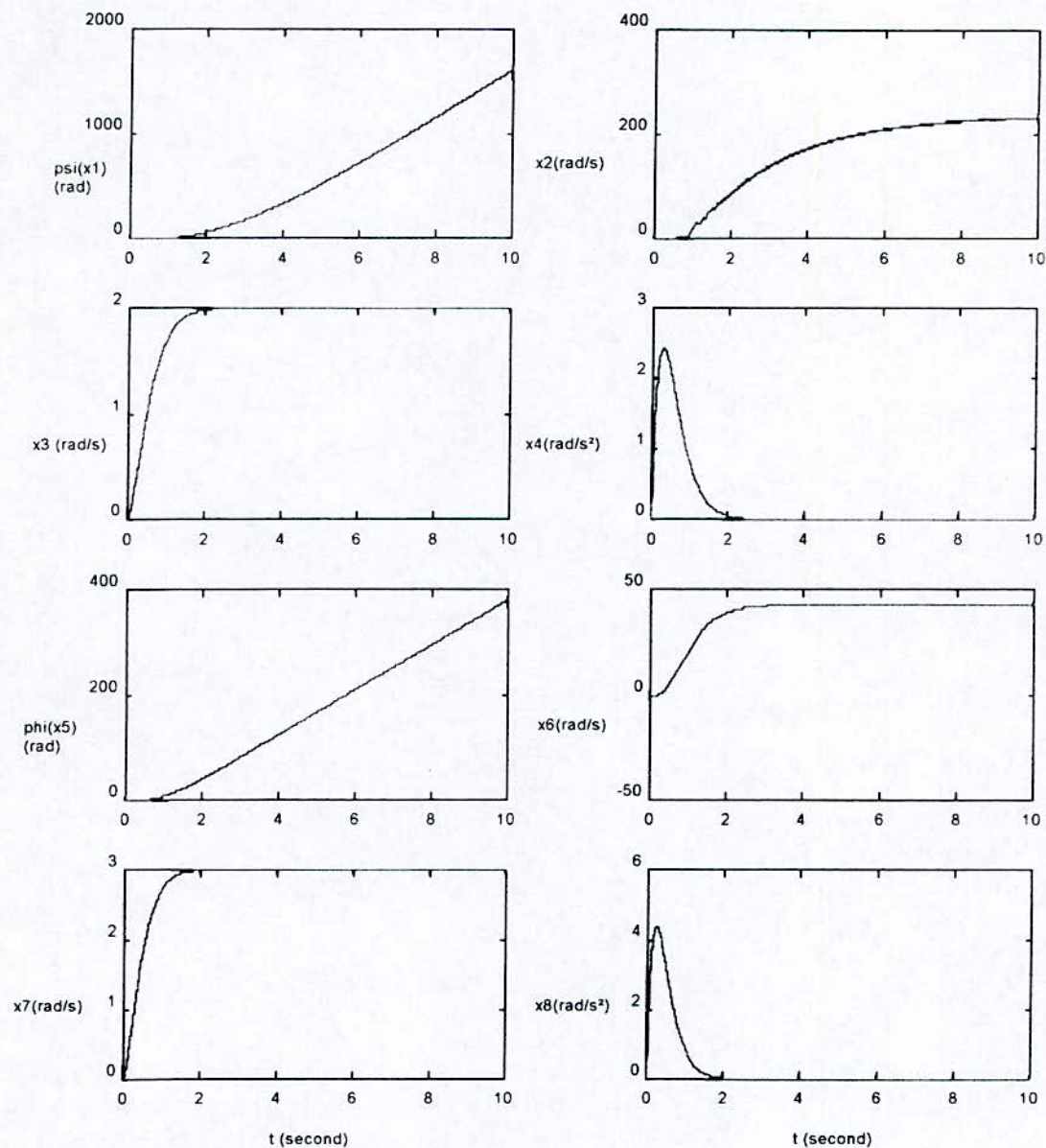
$$\begin{aligned} \alpha_1 &= \frac{B_\psi}{I_1}; & \alpha_2 &= \frac{\tau_g}{I_1}; & \alpha_3 &= \frac{a_1}{I_1}; & \alpha_4 &= \frac{b_1}{I_1}; & \alpha_5 &= \frac{2}{T_1}; & \alpha_6 &= \frac{1}{T_1^2}; \\ \alpha_7 &= \frac{a_2}{I_2}; & \alpha_8 &= \frac{b_2}{I_2}; & \alpha_9 &= \frac{B_\varphi}{I_2}; & \alpha_{10} &= \frac{k_r T_{or}}{I_2 T_{pr}}; & \alpha_{11} &= \frac{2}{T_2}; & \alpha_{12} &= \frac{1}{T_2^2}. \end{aligned}$$

et :

$$\begin{aligned} B_\psi &= 1.84 \cdot 10^{-3} \text{ kg.m}^2 / \text{s} & I_1 &= 4.37 \cdot 10^{-3} \text{ kg.m}^2 & \tau_g &= 3.83 \cdot 10^{-2} \text{ N.m} \\ \alpha_1 &= 0.105 \text{ N.m} / \text{MU}^2 & b_1 &= 0.00936 \text{ N.m} / \text{MU} & T_1 &= 0.3 \text{ s} \\ \alpha_2 &= 0.033 \text{ N.m} / \text{MU}^2 & I_2 &= 4.14 \cdot 10^{-3} \text{ kg.m}^2 & b_2 &= 0.0294 \text{ N.m} / \text{MU} \\ B_\varphi &= 8.69 \cdot 10^{-3} \text{ kg.m}^2 / \text{s} & k_r &= 0.00162 \text{ N.m} / \text{MU} & T_{or} &= 2.7 \text{ s} \\ T_{pr} &= 0.75 \text{ s} & T_2 &= 0.25 \text{ s} & k_{gyro} &= 0.015 \text{ N.m} / \text{s} \end{aligned}$$

1.8 Essai en boucle ouvert

On fait un essai en boucle ouverte avec les entrées $u_1 = 2 \text{ volts}$, $u_2 = 3 \text{ volts}$.



Conclusion

Dans ce chapitre, nous avons développé un modèle de commande pour le simulateur d'hélicoptère.

Chapitre 2

*COMMANDES
QUANTITATIVES*

2.1 Commande avec linéarisation par retour d'état

2.1.1 Introduction

La linéarisation par retour d'état est une approche de commande non linéaire qui a attiré un grand nombre de chercheurs ces dernières années. L'idée principale de l'approche est de transformer algébriquement le système non linéaire, (complètement ou partiellement) en un autre système linéaire ; tel que les techniques de commande linéaire peuvent être appliquées. Ceci est entièrement différent de la linéarisation conventionnelle (i.e. linéarisation Jacobienne), du fait que la linéarisation par retour d'état est basée sur une transformation exacte des états du système, et non pas en une approximation linéaire de la dynamique. [18]

L'idée de simplifier la forme de la dynamique d'un système en choisissant différentes représentations n'est pas entièrement inhabituelle. En mécanique, il est bien connu que la forme et la complexité d'un système dépendent considérablement du choix du repère et des coordonnées. La méthode de linéarisation par retour d'état peut être vue comme une *' technique de transformation du système original en un modèle plus simple '*.

La linéarisation par retour d'état a été utilisée avec succès pour résoudre quelques problèmes de commande. Par exemple dans la commande des hélicoptères, avions de haute performance, robots industriels et des systèmes biomédicales. Beaucoup d'applications de la méthode se développent actuellement dans l'industrie. Cependant, il existe un nombre de limitations associé à la linéarisation par retour d'état ; et d'autres problèmes persistent et font l'objet de sujets de recherches.

2.1.2 Concepts intuitifs

Cette section décrit les concepts de base de la linéarisation par retour d'état en utilisant des exemples simples.

2.1.2.1 Linéarisation par retour d'état et forme canonique [18]

Dans sa forme la plus simple, la linéarisation par retour d'état se base sur une transformation telle que la dynamique en boucle fermée est sous forme linéaire. Cette idée très simple est illustrée dans l'exemple suivant :

Exemple : (Régulation du niveau de fluide dans un réservoir).

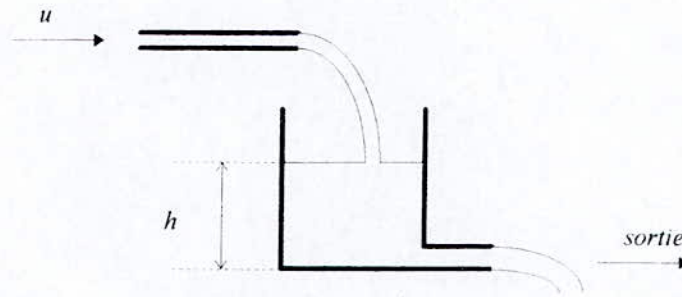


Figure 2.1 : représentation du réservoir

Soit le problème de régulation du niveau h dans un réservoir à un niveau h_d . L'entrée de commande est le débit u dans le réservoir, et le niveau initial est h_0 .

La dynamique du modèle est :

$$\frac{d}{dt} \left[\int_{h_0}^h A(h) dh \right] = u(t) - a \sqrt{2gh} \quad (2.1.1)$$

où $A(h)$ est la surface du réservoir et a est la surface de la section de la sortie du réservoir. Si le niveau initial h_0 est différent de celui désiré, la régulation de h nécessite une analyse d'un problème non linéaire. La dynamique (2.1.1) peut être écrite par :

$$A(h) \dot{h} = u(t) - a \sqrt{2gh} \quad (2.1.2)$$

si $u(t)$ est choisie telle que :

$$u(t) = a \sqrt{2gh} + A(h) v \quad (2.1.3)$$

avec v est une 'entrée équivalente' à spécifier, la dynamique résultante est linéaire.

$$\dot{h} = v \quad (2.1.4)$$

en choisissant v comme :

$$v = -\alpha \tilde{h} \quad (2.1.5)$$

avec $\tilde{h} = h(t) - h_d$ est l'erreur de niveau, et α est une constante strictement positive, la dynamique résultante en boucle fermée est :

$$\dot{h} + \alpha \tilde{h} = 0 \quad (2.1.6)$$

ceci implique que $\tilde{h}(t) \rightarrow 0$ quand $t \rightarrow \infty$. En se basant sur (2.1.3) et (2.1.5), l'entrée de commande est donnée par la loi non linéaire suivante :

$$u(t) = a\sqrt{2gh} - A(h)\alpha\tilde{h} \quad (2.1.7)$$

On note que, dans la loi de commande (2.1.7), la première partie de la droite est utilisée pour produire le débit de sortie $a\sqrt{2gh}$, cependant la deuxième partie est utilisée pour réduire le niveau du fluide en relation avec le niveau désiré (2.1.6).

De même, si le niveau désiré est donné par une fonction variable dans le temps $h_d(t)$, l'entrée équivalente v peut être choisie comme :

$$v = \dot{h}_d(t) - \alpha\tilde{h} \quad (2.1.8)$$

tel que $\tilde{h}(t) \rightarrow 0$ quand $t \rightarrow \infty$ reste valable.

2.1.2.2 Cas d'un système sous forme canonique

L'idée de linéarisation par retour d'état, i.e., de compenser les non linéarités et d'imposer une dynamique linéaire, peut être simplement appliquée à une classe de systèmes non linéaires décrits par la *forme compagne*, ou *forme canonique de commande*. Un système est dit qu'il est sous forme compagne, si sa dynamique est représentée par :

$$\dot{x}^{(n)} = f(\mathbf{x}) + b(\mathbf{x})u \quad (2.1.9)$$

où u est une entrée scalaire, x est la sortie scalaire qui nous intéresse, $\mathbf{x} = [x, \dot{x}, \dots, x^{(n-1)}]^T$ est le vecteur d'état, $f(x)$ et $b(x)$ sont des fonctions d'état non linéaires. Cette forme est unique du fait qu'on ne trouve que des dérivées de x et non pas celles de u . Notons qu'en représentation d'état, (2.1.9) peut être écrite :

$$\frac{d}{dt} \begin{bmatrix} x_1 \\ \dots \\ x_{n-1} \\ x_n \end{bmatrix} = \begin{bmatrix} x_2 \\ \dots \\ x_n \\ f(\mathbf{x}) + b(\mathbf{x})u \end{bmatrix} \quad (2.1.10)$$

Pour les systèmes pouvant être écrits sous la forme canonique de commande, en utilisant la commande :

$$u = \frac{1}{b} [v - f] \quad (2.1.11)$$

on peut compenser les non linéarités et obtenir la relation simple entrée-sortie (forme d'intégrateurs en cascade) :

$$x^{(n)} = v \quad (2.1.12)$$

donc, on peut prendre :

$$v = -k_0 x - k_1 \dot{x} - \dots - k_{n-1} x^{(n-1)} \quad (2.1.13)$$

avec : k_i choisis tel que le polynôme $p^n + k_{n-1} p^{n-1} + \dots + k_0$ a toutes ces racines dans la partie gauche du plan complexe, ce qui mène à une dynamique exponentiellement stable :

$$k_0 x + k_1 \dot{x} + \dots + k_{n-1} x^{(n-1)} + x^{(n)} = 0 \quad (2.1.14)$$

qui implique que $x(t) \rightarrow 0$. Pour les problèmes de poursuite, on choisit la commande de la forme :

$$k_0 e + k_1 \dot{e} + \dots + k_{n-1} e^{(n-1)} + x_d^{(n)} = 0 \quad (2.1.15)$$

(où $e(t) = x(t) - x_d(t)$ est l'erreur de poursuite) ce qui conduit à une convergence exponentielle. On note que les mêmes résultats sont obtenus en remplaçant le scalaire x par un vecteur et le scalaire b par une matrice inversible.

Une application très intéressante de la commande précédente est en robotique. L'exemple est traité dans [18].

2.1.2.3 Linéarisation 'entrée-sortie'

Soit le système non linéaire :

$$\dot{\mathbf{x}} = \mathbf{f}(\mathbf{x}, u) \quad (2.1.16.a)$$

$$y = h(\mathbf{x}) \quad (2.1.16.b)$$

et supposons que notre objectif est de faire suivre à $y(t)$ la trajectoire désirée $y_d(t)$, où $y_d(t)$ est continuellement dérivable. Une difficulté qui apparaît, est que la sortie est reliée indirectement à l'entrée u , à travers les variables d'états (2.1.16). Mais, il n'est pas facile de trouver l'entrée u qui peut commander la sortie y . De (2.1.2), on peut voir que la difficulté peut être réduite, en effet '*si on trouve une relation directe et simple entre la sortie y du système et l'entrée u* '. Cette idée constitue la base intuitive d'une approche dite 'linéarisation entrée-sortie'. Illustrant cette approche par l'exemple suivant :

Considérant le système d'ordre trois :

$$\dot{x}_1 = \sin x_2 + (x_2 + 1)x_3 \quad (2.1.17.a)$$

$$\dot{x}_2 = x_1^5 + x_3 \quad (2.1.17.b)$$

$$\dot{x}_3 = x_1^2 + u \quad (2.1.17.c)$$

pour générer une relation directe entre la sortie $y=x_1$ et l'entrée u , dérivant la sortie y :

$$\dot{y} = \dot{x}_1 = \sin x_2 + (1 + x_2)x_3$$

puisque \dot{y} n'est pas directement reliée à l'entrée u , en dérivant une autre fois, on obtient :

$$\ddot{y} = (1 + x_2)u + f_1(x) \quad (2.1.18)$$

où $f_1(x)$ est une fonction définie par :

$$f_1(x) = (x_1^5 + x_3)(x_3 + \cos x_2) + (x_2 + 1)x_1^2 \quad (2.1.19)$$

l'équation (2.1.18) représente une relation explicite entre y et u . Si nous choisissons l'entrée de commande de la forme :

$$u = \frac{1}{1 + x_2}(v - f_1) \quad (2.1.20)$$

où v est une nouvelle entrée à déterminer, la non linéarité dans (2.1.18) est compensée, et on obtient une relation simple linéaire, double intégrateur entre la sortie et la nouvelle entrée v .

$$\ddot{y} = v \quad (2.1.21)$$

La synthèse d'une commande de poursuite de ce système est simple. Soit $e(t) = y(t) - y_d(t)$ l'erreur de poursuite, et en choisissant la nouvelle commande v :

$$v = \ddot{y}_d - k_1 \dot{e} - k_2 e \quad (2.1.22)$$

avec k_1 et k_2 sont des constantes positives, l'erreur de poursuite du système en boucle fermée est donnée par :

$$\ddot{e} + k_1 \dot{e} + k_2 e = 0 \quad (2.1.23)$$

qui représente une dynamique exponentiellement stable. Notons que :

- La commande est définie partout, à l'exception du point singulier $x_2 = -1$.
- Tous les états sont nécessaires pour implementer la commande, parce que le calcul de \dot{y} et la transformation de l'entrée nécessitent la valeur de \mathbf{x} .

Remarque : *Dynamique interne d'un système non linéaire*

En général, il est difficile de déterminer la stabilité de la dynamique interne car elle est non linéaire, non autonome, et couplée à la dynamique externe (boucle fermée). Une analyse basée sur la fonction de Lyapunov peut être utile pour quelques systèmes.

2.1.3 Génération d'une relation 'entrée-sortie' [7]

Soit le système :

$$\dot{x} = f(x) + g(x)u$$

l'approche de base de la linéarisation entrée-sortie est simplement de dériver la sortie plusieurs fois jusqu'à ce que l'entrée apparaisse, puis de synthétiser la commande u qui peut compenser la non linéarité. En se plaçant dans une région Ω_x dans l'espace d'état et en utilisant les notions de la géométrie différentielle, la différentiation répétitive nous donne :

$$\dot{y} = \nabla h(\mathbf{f} + \mathbf{g}u) = L_f h(\mathbf{x}) + L_g h(\mathbf{x})u \quad (2.1.24)$$

Si $L_g h(\mathbf{x}) \neq 0$ pour $\mathbf{x} = \mathbf{x}_0$ dans Ω_x , donc par continuité, si cette relation est vérifiée dans un voisinage Ω de \mathbf{x}_0 . Dans Ω , la transformation :

$$u = \frac{1}{L_g h} (-L_f h + v) \quad (2.1.25)$$

résulte une relation linéaire entre y et v , où $\dot{y} = v$. Si $L_g h(\mathbf{x}) = 0$ pour \mathbf{x} dans Ω , on peut dériver pour obtenir :

$$\ddot{y} = L_f^2 h(\mathbf{x}) + L_g L_f h(\mathbf{x})u \quad (2.1.26)$$

si $L_g L_f h(\mathbf{x})$ est nul, on doit dériver encore :

$$y^{(i)} = L_f^i h(\mathbf{x}) + L_g L_f^{i-1} h(\mathbf{x})u \quad (2.1.27)$$

jusqu'à un ordre r

$$L_g L_f^{r-1} h(\mathbf{x}) \neq 0 \quad (2.1.28)$$

donc la commande :

$$u = \frac{1}{L_g L_f^{r-1} h} (-L_f^r h + v) \quad (2.1.29)$$

appliquée à :

$$y^{(r)} = L_f^r h(\mathbf{x}) + L_g L_f^{r-1} h(\mathbf{x})u \quad (2.1.30)$$

donne la relation linéaire simple :

$$y^{(r)} = v \quad (2.1.31)$$

Le nombre r de dérivations requises pour faire apparaître la commande u est appelé 'degré relatif' du système, une extension de la définition usuelle du degré relatif pour les systèmes linéaires.

2.1.4 Systèmes à plusieurs entrées [7]

Dans le cas MIMO, on considère, dans un voisinage d'un point \mathbf{x}_0 , les systèmes carrés (i.e., les systèmes ayant le même nombre d'entrées que de sorties) de la forme :

$$\dot{\mathbf{x}} = \mathbf{f}(\mathbf{x}) + \mathbf{G}(\mathbf{x})\mathbf{u} \quad \mathbf{y} = \mathbf{h}(\mathbf{x}) \quad (2.1.32)$$

où $\mathbf{x} \in \mathbf{R}^n$ est le vecteur d'état, $\mathbf{u} \in \mathbf{R}^m$ le vecteur d'entrée, $\mathbf{y} \in \mathbf{R}^m$ le vecteur de sortie, \mathbf{f} et \mathbf{h} sont des champs de vecteurs infiniment dérivables et $\mathbf{G} \in \mathbf{R}^{n \times m}$ est une matrice qui a des champs de vecteurs \mathbf{g}_i infiniment dérivables.

2.1.5 Linéarisation par retour d'état pour les systèmes MIMO

La linéarisation entrée-sortie pour les systèmes MIMO est obtenue en dérivant les sorties y_i jusqu'à ce que les entrées apparaissent. Supposons que r_i est le plus petit entier tel qu'au moins une des commandes apparaisse dans $y_i^{(r_i)}$.

donc :

$$y_i^{(r_i)} = L_{\mathbf{r}}^{r_i} h_i + \sum_{j=1}^m L_{\mathbf{g}_j} L_{\mathbf{r}}^{r_i-1} h_i u_j \quad (2.1.33)$$

avec $L_{\mathbf{g}_j} L_{\mathbf{r}}^{r_i-1} h_i(\mathbf{x}) \neq 0$ au moins pour un j , dans un voisinage Ω_i du point \mathbf{x}_0 . En généralisant cette procédure pour chaque sortie y_i :

$$\begin{bmatrix} y_1^{(r_1)} \\ \dots \\ \dots \\ y_m^{(r_m)} \end{bmatrix} = \begin{bmatrix} L_{\mathbf{r}}^{r_1} h_1(\mathbf{x}) \\ \dots \\ \dots \\ L_{\mathbf{r}}^{r_m} h_m(\mathbf{x}) \end{bmatrix} + \mathbf{E}(\mathbf{x}) \mathbf{u} \quad (2.1.34)$$

où la matrice $\mathbf{E}(\mathbf{x})$ est définie automatiquement.

Définissons donc Ω comme l'intersection des Ω_i . Si, les degrés relatifs sont tous définis, donc Ω est lui même un voisinage de \mathbf{x}_0 .

En plus, si $\mathbf{E}(\mathbf{x})$ est inversible sur une région Ω , donc, comme dans le cas SISO, la transformation de l'entrée est :

$$u = E^{-1} \begin{bmatrix} v_1 - L_f^{r_1} h_1 \\ \dots \\ \dots \\ v_m - L_f^{r_m} h_m \end{bmatrix} \quad (2.1.35)$$

donne m équations de la forme :

$$y_i^{(r_i)} = v_i$$

Puisque l'entrée v_i agit seulement sur y_i , (2.1.35) est appelée 'commande de découplage', et la matrice inversible $E(\mathbf{x})$ est appelée 'matrice de découplage' du système. Le système (2.1.32) est dit d'avoir un 'degré relatif' (r_1, \dots, r_m) à x_0 , et le scalaire $r = r_1 + \dots + r_m$ est dit 'degré relatif total' du système à x_0 .

Un cas intéressant correspond à un degré relatif total égal à n . Dans ce cas, on n'a pas de dynamiques internes. Avec une loi de commande de la forme (2.1.35), on obtient une linéarisation entrée-état du système non linéaire. Avec les entrées équivalentes v_i déterminées de la même façon que dans le cas SISO, on peut réaliser une stabilisation ou une poursuite sans aucun souci sur la stabilité des dynamiques internes.

2.2 Commande à structure variable

2.2.1 Introduction

Pour discuter les avantages et les inconvénients de la commande à structure variable (CSV), il faut répondre en premier lieu à la question suivante : Pourquoi utilise-t-on cette commande ?

L'idée fondamentale de la CSV est de conduire les états du système à la surface de commutation et par un choix judicieux des gains de la commande on fait maintenir les variables d'état sur la surface de commutation pour toutes les séquences du temps suivantes, la surface de commutation est choisie de telle façon que tous les états du système se dirigent vers la surface de commutation, et la dynamique du nouveau système est équivalente à la dynamique désirée.

La CSV a été étudiée en détail par plusieurs chercheurs tels que UTKIN, FILLIPOV et EMELY'NANOV dans l'ex Union Soviétique, on voit que la CSV appliquée pour un système linéaire donne un résultat souvent indépendant des paramètres du système donc notre commande est robuste, la conception mathématique de la commande pour le cas linéaire est plus facile que d'autres méthodes.

Pour les systèmes non linéaires les questions de robustesse et de bruits sont des problèmes considérablement difficiles à résoudre, cependant pour une large classe de systèmes non linéaires le résultat linéaire peut être appliqué et la CSV fournit une méthode qui force le système à se comporter comme un système linéaire invariant dans le temps.

2.2.2 Conditions d'existence du mode de glissement [20]

Intuitivement, la commande par mode glissant fait tendre la trajectoire vers la surface de glissement. Quand la trajectoire sera sur la surface de glissement, on dit qu'on est en régime glissant.

si on suppose un système à structure variable tel que :

$$\dot{x} = f(x) = \begin{cases} f^+(x) & \text{si } \sigma(x) > 0 \\ f^-(x) & \text{si } \sigma(x) < 0 \end{cases} \quad (2.2.1)$$

On voit que le système n'est pas défini sur la surface $\sigma = 0$, mais on peut définir :

$$\lim_{\sigma \mapsto 0^-} f(x) = f^-(x) \qquad \lim_{\sigma \mapsto 0^+} f(x) = f^+(x)$$

où le vecteur de champ $f^+(x)$ et $f^-(x)$ est bien défini. On a $\frac{d\sigma}{dt} = \langle d\sigma, f(x) \rangle$ et on désire que la trajectoire du système tend vers la surface de glissement.

$$\lim_{\sigma \mapsto 0^-} \frac{d\sigma}{dt} > 0 \quad ; \quad \lim_{\sigma \mapsto 0^+} \frac{d\sigma}{dt} < 0$$

Les deux conditions précédentes seront équivalentes à :

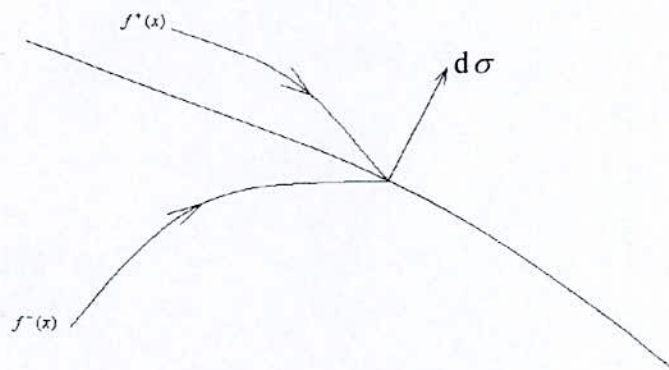


Figure 2.2.1 : représentation graphique de la condition d'existence

$$\lim_{\sigma \mapsto 0} \sigma \cdot \frac{d\sigma}{dt} < 0 \Rightarrow \lim_{\sigma \mapsto 0} \frac{1}{2} \cdot \frac{d}{dt} (\sigma^2) < 0$$

ici on peut parler d'une fonction de Lyapunov $V = \frac{1}{2} \cdot \sigma^2$, qui est candidate au système.

2.2.3 Surface de commutation

La surface de commutation est une loi dans R^n . La loi de commande utilise la surface comme une référence pour déterminer l'instant de commutation entre les structures du retour d'état. Elle est construite pour obtenir une réponse désirée. Par exemple, on peut choisir les paramètres de la surface pour avoir le comportement du système équivalent voulu. Nous pouvons utiliser la commande optimale, par exemple, pour trouver les paramètres qui minimisent quelques performances.

2.2.4 Détails sur la CSV pour les systèmes mono-entrée [13]

Généralement pour la plupart des systèmes non linéaire mono-entrée affines en entrée dont le modèle est donné par :

$$\dot{x} = f(x, t) + B(x, t) \cdot u(t) \quad (2.2.2)$$

où $f(x, t) \in \mathfrak{R}^n$; $B(x, t) \in \mathfrak{R}^n$, le vecteur d'état $x(t) \in \mathfrak{R}^n$ et la commande d'entrée $u(t) \in \mathfrak{R}$; la loi de commande $u(t) = u(x, t)$ a la structure générale suivante :

$$u(x, t) = \begin{cases} u(x, t)^+ \text{ avec } \sigma(x) > 0 \\ u(x, t)^- \text{ avec } \sigma(x) < 0 \end{cases} \quad (2.2.3)$$

2.2.4.1 Forme canonique de commande

Il est relativement facile de concevoir une loi de commande qui stabilise notre système non linéaire pour uniquement quelques structures spéciales qui peuvent être équivalentes à un système linéaire. Une de ces structures spéciales est la forme canonique de commande (2.2.4), cette forme est donnée par :

$$\dot{x}(t) = A(x, t) \cdot x(t) + B \cdot u(t) \quad (2.2.4)$$

où $x(t) = [x_1(t), \dots, x_n(t)] \in \mathfrak{R}^n$, et $u(t) \in \mathfrak{R}$; $B^T = [0, \dots, 1] \in \mathfrak{R}^n$, $A(x, t) \in \mathfrak{R}^{n \times n}$ telle que :

$$A(x, t) = \begin{bmatrix} 0 & 1 & \dots & 0 \\ \cdot & \cdot & \cdot & \cdot \\ \cdot & \cdot & \cdot & \cdot \\ \cdot & \cdot & \cdot & \cdot \\ 0 & 0 & \cdot & 1 \\ a_1(x, t) & a_2(x, t) & \dots & a_n(x, t) \end{bmatrix} \quad (2.2.5)$$

on suppose que $a_i(x, t)$ sont continuellement dérivables .

2.2.4.2 Détails sur la surface de commutation

Intuitivement, on peut utiliser une surface de commutation non linéaire, mais il est plus judicieux pour que le comportement dynamique de notre système soit linéaire d'utiliser une surface linéaire. La forme générale de la surface est la suivante :

$$\sigma(x, t) = S \cdot x + u_r(t) = \sum_{i=1}^n s_i \cdot x_i + u_r(t) = 0 \quad (2.2.6)$$

où $s_i \in \mathfrak{R}$ est un scalaire , $u_r(t)$ est une entrée de référence continuellement dérivable.

2.2.5 Procédure de Conception

La conception de la CSV comporte deux étapes ; la première est de choisir la surface de commutation donc limiter notre système à la surface de commutation pour avoir la réponse désirée, la deuxième étape est de déterminer le gain de la loi de commande qui garantie la condition l'existence du mode glissant .

2.2.5.1 Détermination de la surface de commutation

Pour un système non linéaire sous forme canonique de commande, il existe en particulier une méthode de détermination des paramètres de la surface de commutation, cette méthode n'est pas générale pour les systèmes de forme quelconque, nous étudierons :

- 1) conception des surfaces de commutation pour les systèmes non linéaires de la forme canonique de commande .
- 2) description de la méthode de la commande équivalente .

Pour les systèmes du type (2.2.4), on a la surface de commutation de la forme (2.2.6), où $S \in \mathcal{R}^{1 \times n}$ est une matrice constante de gains $S = [s_1, \dots, s_n]$ et on suppose que $s_n \neq 0$ et $u_r(t)$ est continuellement dérivable.

Théorème [13]

Supposant qu'on a un système non linéaire de la forme canonique de commande, si la trajectoire d'état est limitée à la surface de commutation, alors le mouvement de la trajectoire d'état est décrit par un système linéaire équivalent d'ordre $(n-1)$ de la forme :

$$\dot{\tilde{x}} = \tilde{A} \cdot \tilde{x}(t) + \tilde{B} \cdot \tilde{u}_r(t)$$

où $\tilde{x}(t) \in \mathcal{R}^{n-1}$; $\tilde{A} \in \mathcal{R}^{(n-1) \times (n-1)}$, $\tilde{B} \in \mathcal{R}^{n-1}$, tel que $\tilde{x}(t) = [x_1(t), \dots, x_{n-1}(t)]^T$ est le vecteur d'état :

$$\tilde{A} = \begin{bmatrix} 0 & 1 & \dots & 0 \\ 0 & 0 & \dots & 1 \\ \tilde{s}_1 & \tilde{s}_2 & \dots & \tilde{s}_{n-1} \end{bmatrix}, \tilde{B} = \begin{bmatrix} 0 \\ \cdot \\ \cdot \\ 0 \\ 1 \end{bmatrix} \quad (2.2.7)$$

et $\tilde{u}_r(t) = -\frac{1}{s_n} \cdot u_r(t)$, $\tilde{s}_i = -\frac{s_i}{s_n}$ pour $i = 1, \dots, n-1$

Exemple

Pour un système d'ordre trois sous forme canonique de commande, on veut concevoir une surface de commutation :

$$A = \begin{bmatrix} 0 & 1 & 0 \\ 0 & 0 & 1 \\ a_1(x,t) & a_2(x,t) & a_3(x,t) \end{bmatrix}; B = \begin{bmatrix} 0 \\ 0 \\ 1 \end{bmatrix} \quad (2.2.8)$$

On désire obtenir une réponse asymptotiquement stable, en plaçant dans l'équation caractéristique les valeurs propres -1 et -2, la surface de commutation est :

$$\sigma(x_1, x_2, x_3) = s_1 \cdot x_1 + s_2 \cdot x_2 + s_3 \cdot x_3 \quad (2.2.9)$$

le système équivalent linéaire sera donc :

$$\dot{\tilde{x}}(t) = \begin{bmatrix} 0 & 1 \\ \tilde{s}_1 & \tilde{s}_2 \end{bmatrix} \cdot \tilde{x}(t) + \begin{bmatrix} 0 \\ 1 \end{bmatrix} \cdot u(t)$$

le polynôme caractéristique associé : $P(\lambda) = \lambda^2 - \tilde{s}_2 \cdot \lambda - \tilde{s}_1$

si on choisit les racines -1, -2 c'est à dire : $\tilde{s}_2 = -3, \tilde{s}_1 = -2$, donc on a :

$\tilde{s}_i = -\frac{s_i}{s_n}$ pour $i = 1, 2$ et $n = 3$, d'après le théorème si dessus, en choisissant $s_3 = 1, s_2 = 3, s_1 = 2$

donc la surface de commutation résultante est :

$$\sigma(x_1, x_2, x_3) = 2 \cdot x_1 + 3 \cdot x_2 + x_3 = 0 \quad (2.2.10)$$

2.2.5.1.2 La commande équivalente

Notre but est de trouver une expression explicite de la commande équivalente $u_{eq}(t)$ qui est calculée depuis le système équivalent, et qui fait déplacer le système équivalent sur la surface de commutation .

Supposons que le mode glissant existe $\sigma(x, t) = 0$. Si la trajectoire d'état coupe la surface de commutation à l'instant $t = t_0$, alors pour tout $t \geq t_0$, $\sigma(x, t) = 0$. Cela implique que $\frac{d\sigma(x, t)}{dt} = 0$ pour tout $t \geq t_0$, depuis la règle de chaîne $\frac{d\sigma}{dt} = \frac{\partial \sigma}{\partial x} \cdot \frac{\partial x}{\partial t} + \frac{\partial \sigma}{\partial t}$, si on prend $\sigma(x, t) = S \cdot x + u_{ref} = 0 \Rightarrow S \cdot \dot{x} + \dot{u}_{ref} = 0$, pour $u(t) = u_{eq}(t)$ une commande continuellement dérivable, on observe que :

$$S \cdot \dot{x} = S \cdot f(x, t) + s \cdot B(x, t) \cdot u_{eq}(t) \quad (2.2.11)$$

supposons que $S \cdot B(x, t)$ est non singulière pour tout x et t , résolvant pour $u_{eq}(t)$:

$$u_{eq}(t) = -[SB(x, t)]^{-1} \cdot S \cdot f(x, t) - [SB(x, t)]^{-1} \cdot \dot{u}_{ref}(t) \quad (2.2.12)$$

le système sera régi par l'équation suivante :

$$\dot{x} = [1 - B(x, t) \cdot [S \cdot B(x, t)]^{-1} \cdot S] \cdot f(x, t) - B(x, t) \cdot [S \cdot B(x, t)]^{-1} \cdot \dot{u}_{ref} \quad (2.2.13)$$

2.2.5.2 Développement de la loi de commande [18] [13]

Rappelons d'abord que lorsqu'il y a régime glissant, la dynamique est indépendante de la loi de commande qui n'a pour seul but que de maintenir les conditions de glissement, c'est-à-dire l'attractivité des surfaces de commutation, c'est pourquoi la surface de glissement peut être déterminée indépendamment de la commande, sur la base du système et des performances désirées.

Pour obtenir un régime glissant, on suppose une commande discontinue, où la surface de glissement doit être attractive.

Supposons qu'on a un système non linéaire

$$\dot{x} = f(x, t) + B(x, t)u \quad (2.2.14)$$

auquel on associe la surface

$$\sigma(x, t) = S \cdot x \quad (2.2.15)$$

La commande équivalente est calculée, sachant que le comportement dynamique durant le mode glissant est décrit par

$$\dot{\sigma}(x, t) = 0 \quad (2.2.16)$$

donc

$$u_{eq} = -(SB)^{-1}Sf(x) \quad (2.2.17)$$

Si on applique une commande constituée d'une partie continue

$$u_c = u_{eq} \quad (2.2.18)$$

et d'une partie discontinue

$$u_d = -k(SB)^{-1}sign\sigma \quad (2.2.19)$$

On peut facilement démontrer que la condition d'attractivité est vérifiée quel que soit k positif et aussi petit que possible.

On a

$$u = u_{eq} + u_d \quad (2.2.20)$$

Démonstration [13]

$$\begin{aligned} \dot{\sigma}(x, t) &= S\dot{x} \\ &= S(f(x, t) + B(x, t)u(t)) \\ &= S(f(x, t) + B(x, t)[u_{eq} + u_d]) \\ &= S(f(x, t) + B(x, t)[-(SB)^{-1}Sf(x) - k(SB)^{-1}sign\sigma]) \\ &= S(f(x, t) - f(x, t) - kS^{-1}sign\sigma) \\ &= -ksign\sigma \end{aligned} \quad (2.2.21)$$

donc la condition d'existence du mode glissant est garantie.

2.3 Commande par « Backstepping »

2.3.1 Introduction

La technique de 'Backstepping' est une procédure récursive qui nous aide à construire une fonction de Lyapunov pour la synthèse d'une loi de commande. Elle décompose le problème de synthèse d'un système complet en une séquence de problèmes de synthèse de systèmes d'ordre inférieur.

En exploitant l'extra flexibilité qu'existe avec les systèmes d'ordre inférieur, la méthode de « Backstepping » peut souvent résoudre les problèmes de stabilisation, de poursuite (*tracking*) et de la commande robuste.

2.3.2 Présentation de la méthode [10]

On commence par le cas spécial de l'intégrateur « Backstepping ». Considérons le système suivant

$$\dot{\eta} = f(\eta) + g(\eta) \cdot \xi \quad (2.3.1)$$

$$\dot{\xi} = u \quad (2.3.2)$$

où $[\eta^T, \xi^T]^T \in \mathfrak{R}^{n+1}$ est le vecteur d'état et $u \in \mathfrak{R}$ est l'entrée de commande. Les fonctions $f : D \rightarrow \mathfrak{R}^n$ et $g : D \rightarrow \mathfrak{R}^n$ sont infiniment dérivables dans un domaine $D \subset \mathfrak{R}^n$ qui contient $\eta = 0$ et $f(0) = 0$.

On veut synthétiser une commande par retour d'état (state feedback control) qui stabilise le système suscité à l'origine ($\eta=0, \xi=0$). On suppose que les deux fonctions g et f sont connues. Ce système peut être vu comme une connexion en cascade de deux composantes, comme il est montré sur la figure 2.3.1(a) ; la première composante est (2.3.1), avec ξ comme entrée, et la deuxième est l'intégrateur (2.3.2).

Supposons que le sous système (2.3.1) peut être stabilisé par une commande par retour d'état $\xi = \phi(\eta)$ avec $\phi(0) = 0$; ce qui donne :

$$\frac{d}{dt} \eta = f(\eta) + g(\eta)\phi(\eta) \tag{2.3.3}$$

est asymptotiquement stable. Supposant aussi qu'on connaît une fonction de Lyapunov (Infiniment dérivable et définie positive) $V(\eta)$ qui satisfait l'inégalité :

$$\left(\frac{d}{d\eta}V\right)(f(\eta) + g(\eta)\phi(\eta)) \leq -W(\eta) \tag{2.3.4}$$

où $W(\eta)$ est définie positive.

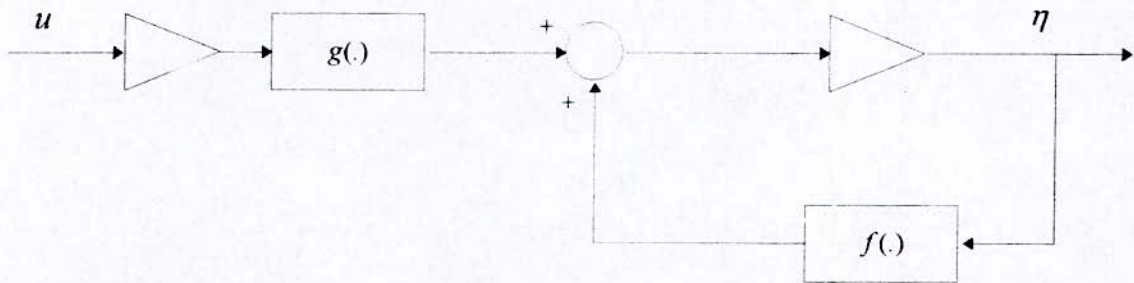


Figure 2.3.1.a

En rajoutant et soustrayant $g(\eta)\phi(\eta)$ de l'équation (2.3.1), on obtient la représentation équivalente dans la figure (2.3.1.b) et l'équation (2.3.5):

$$\frac{d}{dt} \eta = (f(\eta) + g(\eta)\phi(\eta)) + g(\eta)(\xi - \phi(\eta)) \tag{2.3.5.a}$$

$$\frac{d}{dt} \xi = u \tag{2.3.5.b}$$

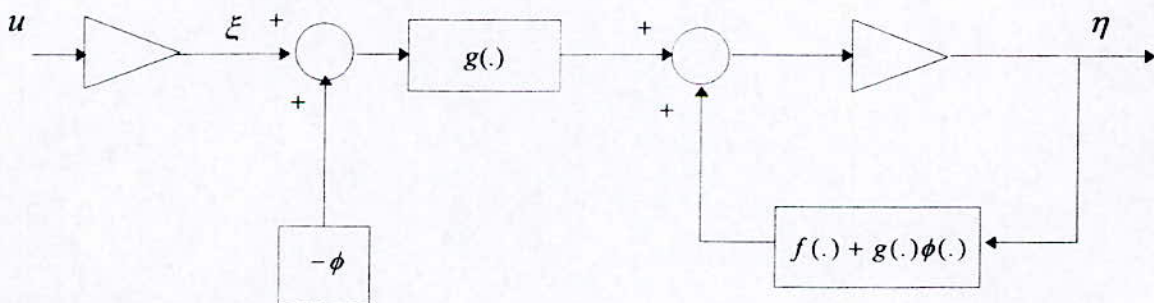


Figure 2.3.1.b

Soit le changement de variable suivant : $z = \xi - \phi(\eta)$, alors le système devient :

$$\frac{d}{dt} \eta = (f(\eta) + g(\eta)\phi(\eta)) + g(\eta)z \quad (2.3.6.a)$$

$$\frac{d}{dt} z = u - \frac{d}{dt} \phi \quad (2.3.6.b)$$

Ce système est représenté à la figure (2.3.1.c). Le passage de la figure (2.3.1.b) à (2.3.1.c) peut être considéré comme un « Backstepping » de $-\phi(\eta)$ à travers l'intégrateur.

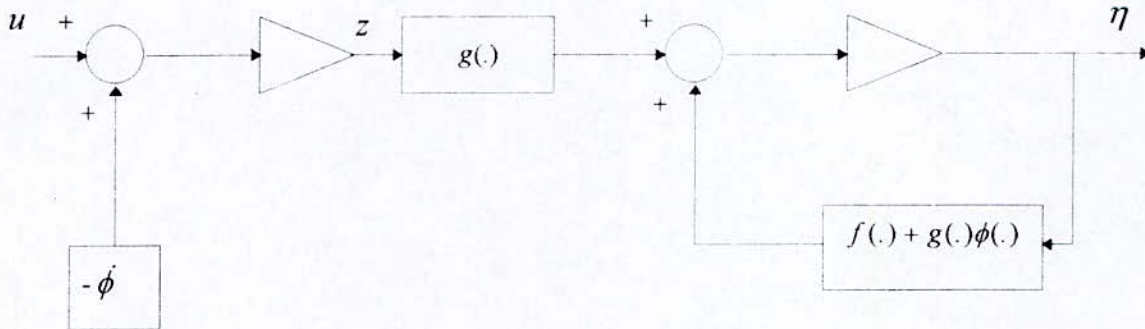


Figure 2.3.1.c

Puisque f , g et ϕ sont connues, la dérivée $\dot{\phi}$ peut être calculée en utilisant l'expression :

$$\frac{d}{dt} \phi = \frac{d}{d\eta} \phi (f(\eta) + g(\eta)\phi(\eta)) + g(\eta)z \quad (2.3.7)$$

en prenant $v = (u - \dot{\phi})$; on réduit le système à une connexion en cascade :

$$\frac{d}{dt} \eta = (f(\eta) + g(\eta)\phi(\eta)) + g(\eta)z \quad (2.3.8.a)$$

$$\frac{d}{dt} z = v \quad (2.3.8.b)$$

qui est similaire au système de démarrage à l'exception que cette fois le premier terme est asymptotiquement stable quand l'entrée est nulle. Cette caractéristique va être exploitée dans la synthèse de v pour stabiliser le système global. En utilisant :

$$V_a = V(\eta) + \frac{1}{2} z^2 \quad (2.3.9)$$

comme fonction de Lyapunov, on obtient :

$$\left[\frac{d}{dt} V_a = \frac{d}{d\eta} V \cdot (f(\eta) + g(\eta)\phi(\eta)) + \frac{d}{d\eta} V \cdot g(\eta)z + zv \right] \leq -W(\eta) + \frac{d}{d\eta} V \cdot g(\eta)z + zv \quad (2.3.10)$$

en choisissant :

$$v = -\frac{d}{d\eta} V \cdot g(\eta) - kz \quad (2.3.11)$$

$$\frac{d}{dt} V_a \leq -W(\eta) - kz^2 \quad k > 0 \quad (2.3.12)$$

qui montre que l'origine ($\eta=0$, $z=0$) est asymptotiquement stable. Puisque $\phi(0)=0$, on conclut que l'origine ($\eta=0$, $\xi=0$) est asymptotiquement stable. En utilisant v , z , et $\dot{\phi}$ on obtient la commande :

$$u = \frac{\partial \phi}{\partial \eta} [f(\eta) + g(\eta) \cdot \xi] - \frac{\partial V}{\partial \eta} \cdot g(\eta) - k \cdot [\xi - \phi(\eta)] \quad (2.3.13)$$

Si toutes les suppositions prises restent valables et $V(\eta)$ est radialement non bornée, on peut conclure que le système à un origine stable.

Lemme [10] : considérons le système (2.3.1) et (2.3.2), avec $\phi(\eta)$, une entrée qui stabilise le système (2.3.1) avec $\phi(0)=0$, et $V(\eta)$ la fonction de Lyapunov qui satisfait (2.3.9) avec une fonction définie positive $W(\eta)$. Alors, la commande par retour d'état (2.3.13) stabilisant le système à l'origine de (2.3.1), (2.3.2) ; avec $V(\eta) + 1/2 \cdot [\xi - \phi(\eta)]^2$ comme fonction de Lyapunov augmentée. En plus, si toutes les suppositions sont maintenues et $V(\eta)$ radialement non bornée, alors l'origine doit être asymptotiquement stable. [6]

Considérons maintenant au lieu du système (2.3.1) ; (2.3.2), un système plus général dont les équations sont données par :

$$\frac{d}{dt} \eta = f(\eta) + g(\eta)\xi \quad (2.3.14.a)$$

$$\frac{d}{dt} \xi = f_a(\eta, \xi) + g_a(\eta, \xi)u \quad (2.3.14.b)$$

où f_a et g_a sont infiniment dérivables. Si $g_a(\eta, \xi) \neq 0$ sur le domaine qui nous intéresse alors la transformation suivante :

$$\frac{d}{dt} \xi = u_a \quad (2.3.15)$$

réduit (2.3.14.b) à $\dot{\xi} = u_a$. Donc, il existe une commande par retour d'état $\phi(\eta)$ et une fonction de Lyapunov $V(\eta)$ qui satisfait les conditions du lemme pour (2.3.14.a), donc le lemme et l'équation (2.3.15) donnent :

$$u = \phi(\eta, \xi)_a = \frac{1}{g(\eta, \xi)_a} \left[\frac{d}{d\eta} \phi \cdot (f(\eta) + g(\eta)\xi) - \frac{d}{d\eta} V \cdot g(\eta) - k(\xi - \phi(\eta)) - f(\eta, \xi)_a \right] \quad (2.3.16.a)$$

$$k > 0$$

$$V(\eta, \xi) = V(\eta) + \frac{1}{2}(\xi - \phi(\eta))^2 \quad (2.3.16.b)$$

qui représentent la commande et la fonction de Lyapunov, respectivement, pour le système global (2.3.16.a), (2.3.16.b). Par l'application récursive du « Backstepping » on peut stabiliser la classe de systèmes dits « *strict-feedback systems* » suivants :

$$\begin{aligned} \dot{x} &= f_0(x) + g_0(x) \cdot z_1 \\ \dot{z}_1 &= f_1(x, z_1) + g_1(x, z_1) \cdot z_2 \\ &\vdots \\ \dot{z}_k &= f_k(x_1, z_2, \dots, z_k) + g_k(x_1, z_2, \dots, z_k) \cdot u \end{aligned} \quad (2.3.17)$$

où $x \in \mathfrak{R}^n$, $z_1 \dots z_k$ sont scalaires, et f_0 et f_k , s'annulent en prenant la raison pour laquelle on a dit que les systèmes précédents sont des « strict-feedback » et que les deux fonctions non linéaires f_i et g_i , dans les équations \dot{z}_i ($i=1, \dots, k$) ne dépendent que de x, z_1, \dots, z_i qui sont des « feed back » c'est à dire que se sont des états précédents par rapport à l'état z_i .

On suppose que : $g_i(x, z_1, \dots, z_i) \neq 0$, pour $k \geq i \geq 1$ sur le domaine qui nous intéresse. La procédure récursive commence avec le système : $\dot{x}_1 = f_0(x) + g_0(x) \cdot z_1$ où z_1 est considéré comme entrée de commande. On suppose qu'il est possible de déterminer une commande par retour d'état $z_1 = \phi_0(x)$ avec $\phi_0(0) = 0$, et une fonction de Lyapunov $V_0(x)$ telle que :

$$\left(\frac{dV_0}{dx} \right) \cdot [f_0(x) + g_0(x) \cdot \phi_0(x)] \leq -W(x) \quad (2.3.18)$$

$W(x)$ est définie positive sur le domaine qui nous intéresse.

Dans beaucoup d'applications de « Backstepping », la variable x est scalaire, ce qui simplifie le problème de stabilisation. Si on dispose de $\phi_0(0)$ et $V_0(x)$ on procède le Backstepping d'une façon systématique. En premier lieu, on considère le sous système :

$$\begin{aligned}\dot{x} &= f_0(x) + g_0(x)z_1 \\ \dot{z}_1 &= f_1(x, z_1) + g_1(x, z_1)z_2\end{aligned}\quad (2.3.19)$$

ce sous système est un cas spécial de (2.3.16.a), (2.3.16.b) avec :

$$\eta = x, \xi = z_1, u = z_2, f = f_0, g = g_0, f_a = f_1, g_a = g_1$$

on utilise (2.3.16.a), (2.3.16.b) pour obtenir la commande qui stabilise le système et la fonction de Lyapunov comme :

$$\begin{aligned}\phi_1(x, z_1) &= \frac{1}{g_1} \cdot \left[\frac{d\phi_0}{dx} (f_0 + g_0 \cdot z_1) - \frac{dV_0}{dx} \cdot g_0 - k_1(z_1 - \phi) - f_1 \right], k_1 > 0 \\ V_1(x, z_1) &= V_0(x) + \frac{1}{2} \cdot (z_1 - \phi(x))^2\end{aligned}\quad (2.3.20)$$

après, on considère le système augmenté suivant :

$$\begin{aligned}\dot{x} &= f_0(x) + g_0(x) \cdot z_1 \\ \dot{z}_1 &= f_1(x, z_1) + g_1(x, z_1) \cdot z_2 \\ \dot{z}_2 &= f_2(x, z_1, z_2) + g_2(x, z_1, z_2) \cdot z_3\end{aligned}\quad (2.3.21)$$

c'est aussi le cas spécial de (2.3.16.a), (2.3.16.b) avec :

$$\eta = \begin{bmatrix} x \\ z_1 \end{bmatrix}; \xi = z_2; u = z_3; f = \begin{bmatrix} f_0 + g_0 \cdot z_1 \\ f_1 \end{bmatrix}; g = \begin{bmatrix} 0 \\ g_1 \end{bmatrix}; f_a = f_2, g_a = g_2$$

En utilisant (2.3.16.a) et (2.3.16.b), on obtient la commande ϕ_2 et la fonction de Lyapunov :

$$\begin{aligned}\phi_2(x, z_1, z_2) &= \frac{1}{g_2} \left[\frac{d\phi_1}{dx} (f_0 + g_0 \cdot z_1) + \frac{d\phi_1}{dz_1} (f_1 + g_1 \cdot z_2) - \frac{dV_1}{dz_1} g_1 - k_2 \cdot (z_2 - \phi_1) - f_2 \right], k_2 > 0 \\ V_2(x, z_1, z_2) &= \phi_2(x, z_1) + \frac{1}{2} \cdot [z_2 - \phi_2(x, z_1)]^2\end{aligned}$$

Cette procédure est répétée k fois pour obtenir la commande par retour d'état qui stabilise le système global $u = \phi_k(x, z_1, \dots, z_k)$ et une fonction de Lyapunov $V_k(x, z_1, \dots, z_k)$.

2.3.3 « Backstepping » pour les systèmes avec termes incertains

On va introduire la méthode du « *nonlinear damping* », qui va être d'un grand secours pour la majoration des perturbations dans la fonction de Lyapunov.

Supposition [10] : Considérons le système :

$$\dot{x} = f(x) + g(x)u \quad (2.3.22)$$

où $x \in \mathfrak{R}^n$ est le vecteur d'état, et $u \in \mathfrak{R}$ est la commande, donc il existe une commande continue, infiniment dérivable :

$$u = \alpha(x), \quad \alpha(0) = 0$$

et une fonction radialement non bornée, définie positive, infiniment dérivable $V: \mathfrak{R}^n \rightarrow \mathfrak{R}$ telle que :

$$\frac{\partial V}{\partial x}(x)[f(x) + g(x)\alpha(x)] \leq -W(x) \leq 0 \quad (2.3.23)$$

où $W: \mathfrak{R}^n \rightarrow \mathfrak{R}$ est une fonction semi-définie positive.

Lemme 1 [12]: soit le système perturbé suivant :

$$\dot{x} = f(x) + g(x)[u + \varphi(x)^T \Delta(x, u, t)] \quad (2.3.24)$$

où $\varphi(x) \in R^{p \times 1}$ un vecteur de fonctions non linaires connues, et $\Delta(x, u, t) \in R^{p \times 1}$ un vecteur de non linéarités incertaines qui sont uniformément bornées pour toutes les valeurs de x, u, t .

Si la supposition ci-dessus est satisfaite, avec $W(x)$ définie positive et radialement non bornée, alors la commande u égale à :

$$u = \alpha(x) - k \frac{\partial V}{\partial x}(x)g(x)|\varphi(x)|^2, \quad k > 0 \quad (2.3.25)$$

quand on applique la commande au système perturbé, et on calcule le système en boucle fermée, on aura un domaine, qui nous garantie la stabilité du système. En dehors de ce domaine, la stabilité n'est pas garantie.

$$R = \left\{ x: |x| \leq \gamma_1^{-1} \circ \gamma_2 \circ \gamma_3^{-1} \left(\frac{\|\Delta\|_{\infty}^2}{4k} \right) \right\} \quad (2.3.26)$$

avec $\gamma_1, \gamma_2, \gamma_3$ sont des fonctions de classe- K_∞ tel que :

$$\begin{aligned}\gamma_1(|x|) &\leq V(x) \leq \gamma_2(|x|) \\ \gamma_3(|x|) &\leq W(x)\end{aligned}$$

Lemme 2 : [12] (majoration via backstepping)

considérons le système :

$$\dot{x} = f(x) + g(x)u + F(x)\Delta_1(x, u, t) \quad (2.3.27)$$

où $x \in \mathfrak{R}^n, u \in \mathfrak{R}, F(x)$ est une matrice des fonctions non linéaires infiniment dérivables $\in R^{n \times q}$, et $\Delta_1(x, u, t)$ est un vecteur $\in R^{q \times 1}$ de non linéarités incertaines qui sont uniformément bornées pour toutes les valeurs x, u, t . On suppose qu'il existe une loi de commande $u = \alpha(x)$ qui rends $x(t)$ globalement borné, et établit la fonction suivante :

$$\frac{\partial V}{\partial x}(x)[f(x) + g(x)\alpha(x) + F(x)\Delta_1(x, u, t)] \leq -W(x) + b \quad (2.3.28)$$

telle que $W(x)$ est une fonction définie positive. Maintenant considérons le système augmenté :

$$\dot{x} = f(x) + g(x)\xi + F(x)\Delta_1(x, u, t) \quad (2.3.29.a)$$

$$\dot{\xi} = u + \varphi(x, \xi)^T \Delta_2(x, \xi, u, t) \quad (2.3.29.b)$$

où $\varphi(x, \xi)$ est un vecteur $\in R^{p \times 1}$ des fonctions non linéaires infiniment dérivables, et $\Delta_2(x, \xi, u, t)$ est un vecteur $\in R^{p \times 1}$ des non linéarités incertaines qui sont uniformément bornées pour toutes les valeurs de x, ξ, u, t . Pour ce système la loi de commande suivante :

$$u = -c[\xi - \alpha(x)] + \frac{\partial \alpha}{\partial x}(x)[f(x) + g(x)\xi] - \frac{\partial V}{\partial x}(x)g(x) - k[\xi - \alpha(x)] \left\{ \left| \varphi(x, \xi) \right|^2 + \left| \frac{\partial \alpha}{\partial x}(x)F(x) \right|^2 \right\} \quad (2.3.30)$$

garantie une bornitude uniforme globale de $x(t)$ et $\xi(t)$ avec $c > 0$ et $k > 0$.

Exemple [12]: Soit le système suivant :

$$\dot{x} = \xi + x^2 \arctan \xi \Delta_0(t) \quad (2.3.31.a)$$

$$\dot{\xi} = (1 + \xi)u + e^{x\xi} \Delta_0(t) \quad (2.3.31.b)$$

où $\Delta_0(t)$ est une perturbation variable dans le temps, alors la méthode du « Backstepping » est utilisée avec une technique de « NonLinear Damping », pour déterminer la loi de commande.

Etape 1 : le point de commencement est l'équation (2.3.31.a) et le choix de la commande virtuelle est clairement ξ , dans la notation :

$$x^2 \arctan \xi \Delta_0(t) = x^2 \Delta_1(\xi, t) = \varphi_1(x) \Delta_1(\xi, t) \quad (2.3.32)$$

la non linéarité incertaine $\Delta_1(\xi, t)$ est bornée :

$$\|\Delta_1(\xi, t)\|_{\infty} = \|\Delta_0 \arctan \xi\|_{\infty} \leq \frac{\pi}{2} \|\Delta_0\|_{\infty} \quad (2.3.33)$$

On peut construire une fonction qui stabilise ξ avec une fonction de Lyapunov $V(x) = \frac{1}{2} x^2$, et la commande correspondante :

$$\alpha_1(x) = -c_1 x - k_1 x \varphi_1^2(x) \quad (2.3.34)$$

si on définit une variable de l'erreur :

$$z = \xi - \alpha_1(x)$$

donc :

$$\dot{x} = -c_1 x + z - k_1 x \varphi_1^2(x) + \varphi_1(x) \Delta_1(\xi, t) \quad (2.3.35)$$

la dérivée de la fonction de Lyapunov devient :

$$\dot{V} = zx - c_1 x^2 - k_1 x^2 \varphi_1^2 + x^3 \arctan \xi \Delta_0(t) \quad (2.3.36)$$

de l'équation (2.3.32) :

$$\dot{V} \leq zx - c_1 x^2 - k_1 x^2 \varphi_1^2 + |x \varphi_1(x)| \|\Delta_1\|_{\infty} \quad (2.3.37)$$

et donc :

$$\dot{V} = zx - c_1 x^2 + \frac{\|\Delta_1\|_{\infty}^2}{4k_1} \quad (2.3.38)$$

Etape 2 : on construit la fonction augmentée :

$$V_2(x, \xi) = \frac{1}{2} x^2 + \frac{1}{2} z^2 \quad (2.3.39)$$

la dérivée de cette fonction est :

$$\dot{V}_2(x, \xi) = x\dot{x} + z\dot{z}$$

avec :

$$\dot{z} = (1 + \xi^2)u + e^{x\xi} \Delta_0(t) - \frac{\partial \alpha_1}{\partial x} [\xi + x^2 \arctan \xi \Delta_0(t)] \quad (2.3.40)$$

$$= (1 + \xi^2)u - \frac{\partial \alpha_1}{\partial x} \xi + [e^{x\xi} - \frac{\partial \alpha_1}{\partial x} x^2 \arctan \xi] \Delta_0(t)$$

où :

$$\frac{\partial \alpha_1}{\partial x} = -c_1 - k_1 \frac{\partial}{\partial x} [x \varphi_1^2(x)] = -c_1 - 5k_1 x^4 \quad (2.3.41)$$

on a :

$$\dot{V}_2 = \dot{V} + z\dot{z}$$

de l'équation (2.3.38) :

$$\dot{V} \leq zx - c_1 x^2 + \frac{\|\Delta_1\|_\infty^2}{4k_1} + z\dot{z} \quad (2.3.42)$$

de (2.3.40) :

$$\dot{V} = -c_1 x^2 + \frac{\|\Delta_1\|_\infty^2}{4k_1} + z \left\{ x + (1 + \xi^2)u - \frac{\partial \alpha_1}{\partial x} \xi + [e^{x\xi} - \frac{\partial \alpha_1}{\partial x} x^2 \arctan \xi] \Delta_0(t) \right\}$$

donc la commande :

$$u = \frac{1}{(1 + \xi^2)} [-c_2 z + \frac{\partial \alpha_1}{\partial x} \xi - x] \quad (2.3.43)$$

pour compenser le terme de perturbation $\Delta_0(t)$, le lemme (1) sera utilisé encore :

$$u = \frac{1}{(1 + \xi^2)} \left\{ -c_2 z + \frac{\partial \alpha_1}{\partial x} \xi - x - k_2 z [e^{x\xi} - \frac{\partial \alpha_1}{\partial x} x^2 \arctan \xi]^2 \right\} \quad (2.3.44)$$

donc :

$$\dot{V} \leq -c_1 x^2 - c_2 z^2 + \frac{\|\Delta_1\|_\infty^2}{4k_1} - k_2 z^2 [e^{x\xi} - \frac{\partial \alpha_1}{\partial x} x^2 \arctan \xi]^2 + |z| \left| e^{x\xi} - \frac{\partial \alpha_1}{\partial x} x^2 \arctan \xi \right| \|\Delta_0\|_\infty$$

alors

$$\dot{V} \leq -c_1 x^2 - c_2 z^2 + \frac{\|\Delta_1\|_\infty^2}{4k_1} + \frac{\|\Delta_0\|_\infty^2}{4k_2} \quad (2.3.45)$$

enfin $x(t)$ est uniformément borné dans un domaine bien défini par l'équation (2.3.45).



Chapitre 3

COMMANDE QUALITATIVE

3.1 Introduction [16]

Durant les dernières années, la logique floue et son application dans différents domaines ont connu un grand essor. On peut distinguer deux périodes principales : la première concerne les expérimentations dans les laboratoires et les prototypes avec relativement des applications industrielles très rares et cela à cause de la dominance de la logique classique. La seconde période n'a commencé que tardivement, dans les années quatre-vingt avec une orientation commerciale qui a inclut une variété d'applications domestiques telles que la machine à laver, la caméra, l'aspirateur...

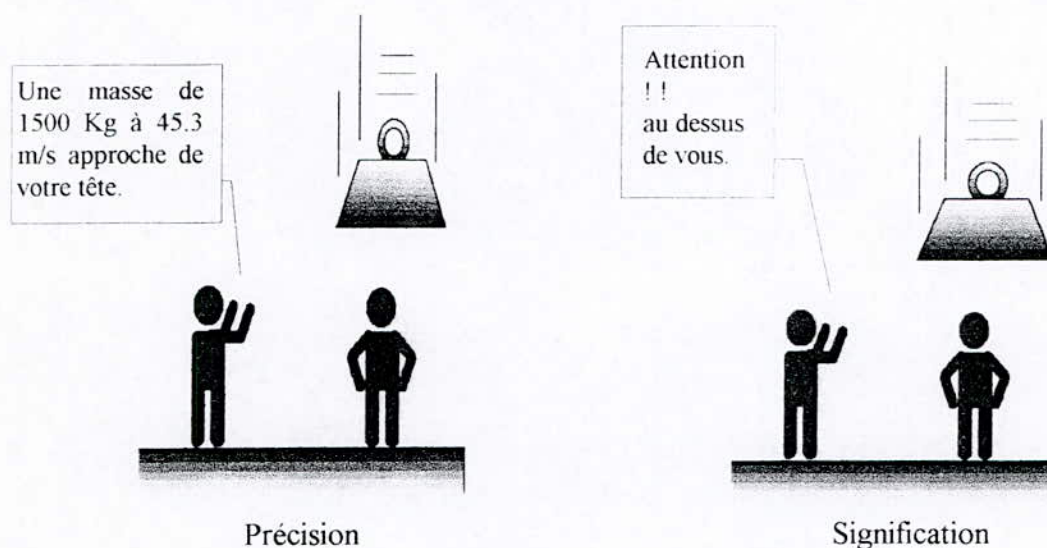


Figure 3.1 : La précision et la signification. [21]

On peut se demander, c'est quoi la logique floue ? la réponse à cette question peut se trouver dans la phrase suivante : ' la précision n'est pas la vérité '. Dans ce contexte on va donner quelques citations :

henri Matisse : « quelques fois beaucoup de mesures nous conduisent de hors l'essentielle. »

lotfi Zadeh : « ne pas perdre de vue la forêt pour voir un arbre. »

Pourquoi utilise-t-on la logique floue ?

Voici quelques raisons pour l'utilisation de la logique floue :

- ☞ Le concept de la logique floue est facile à comprendre, les fondements mathématiques qui sont utilisés dans le raisonnement sont assez simples, ce qui fait que la logique floue est agréable à utiliser.
- ☞ La logique floue tolère l'imprécision des données.
- ☞ La logique floue peut modéliser une fonction non linéaire d'une complexité arbitraire en recourant par exemple aux Systèmes d'Inférence Adaptative Neuro Floue ANFIS.
- ☞ La logique floue est basée sur le langage humain.

Ce dernier argument peut être le plus important parce que le langage humain est utilisé par des gens ordinaires chaque jour, et ce le cumul d'une histoire humaine et d'expériences de milliers d'années ce qui rend la logique floue efficace.

3.2 Ensembles Flous

3.2.1 Variable linguistique

La description d'une certaine situation, d'un phénomène ou d'un procédé, contient en général des expressions floues comme : quelques, beaucoup, souvent, des expressions de ce genre forment des variables linguistiques, par exemple 'les jours de week-end' est une variable linguistique parce que sur mercredi il y a une discussion et ceci varie d'une personne à une autre selon la profession et l'habitude.

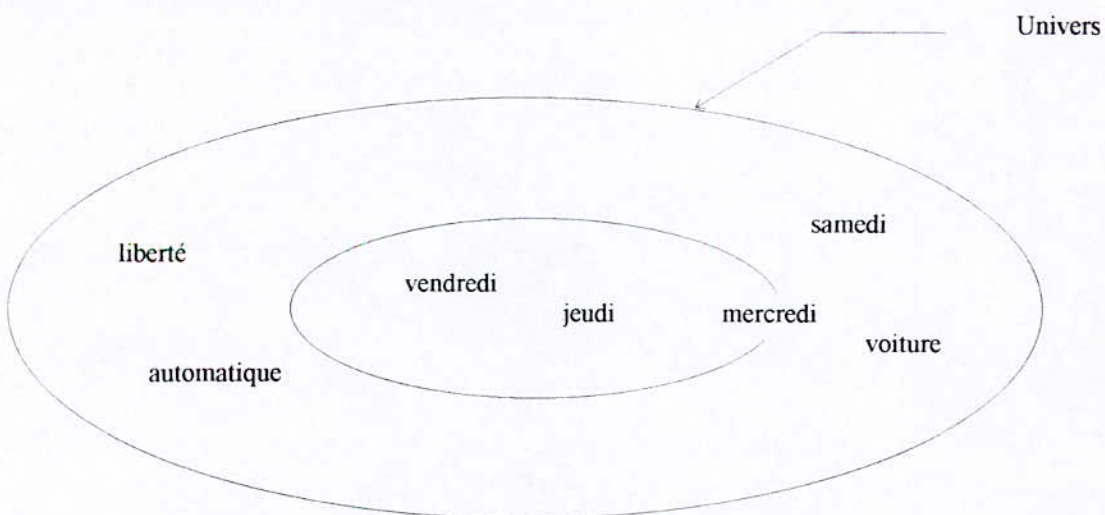


Figure 3.2 : Ensemble flou représentant les jours de week-end.

Mais dans l'approche classique (logique binaire) ce raisonnement n'existe pas, donc on peut conclure que la vérité dans la logique floue n'est pas absolue mais à un degré bien défini.

3.2.2 Fonctions d'appartenances [16] [17]

Afin de permettre un traitement numérique, il est indispensable de soumettre les variables linguistiques à une définition à l'aide de la fonction d'appartenance.

On peut présenter cette définition par un exemple (figure 3.3). On considère que la variable floue est 'les gens de grande taille', dans ce cas on pose les gens qui ont une opportunité d'être considérés comme de grande taille sont compris entre 1.60 et 2.50 m, et on désigne par le mot « grande taille » toute personne ayant une taille supérieure à 1.80 m, mais si on parle des gens, ce n'est pas raisonnable de considérer toute personne au dessous de 1.80 m comme petite quand elle diffère dans la taille à cause de la longueur de ses cheveux, c'est pourquoi on fait recours aux fonctions d'appartenances, donc on peut dire que cette personne est grande à 95% et celle-là à 60%.

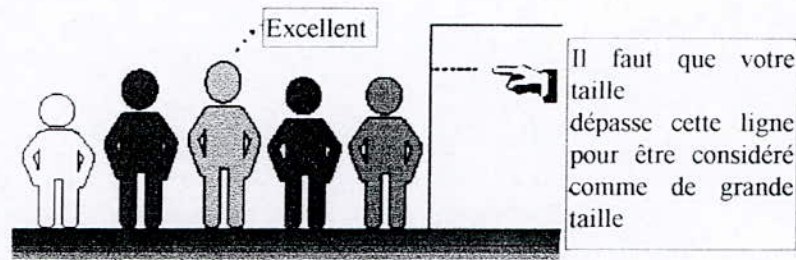


Figure 3.3 : Répartition des gens suivant leurs tailles. [21]

On présente la forme générale des ensembles flous :

$$A = \{x, \mu_A(x) / x \in U\} \quad (3.1)$$

où :

- $\mu_A(x)$ fonction d'appartenance.
- U univers de discussion.
- x élément de U .
- A variable floue.

Les plus simples fonctions d'appartenances sont celles qui utilisent des segments de droites, parmi eux on a : la fonction d'appartenance triangulaire et trapézoïdale. Il y a d'autres fonctions d'appartenances telles que : la fonction gaussienne et la fonction Bell, qui sont les plus populaires pour la spécification des variables floues parce qu'elles ont l'avantage d'être infiniment dérivables et non nulles dans tous les points. Ces fonctions sont présentées dans la figure (3.4)

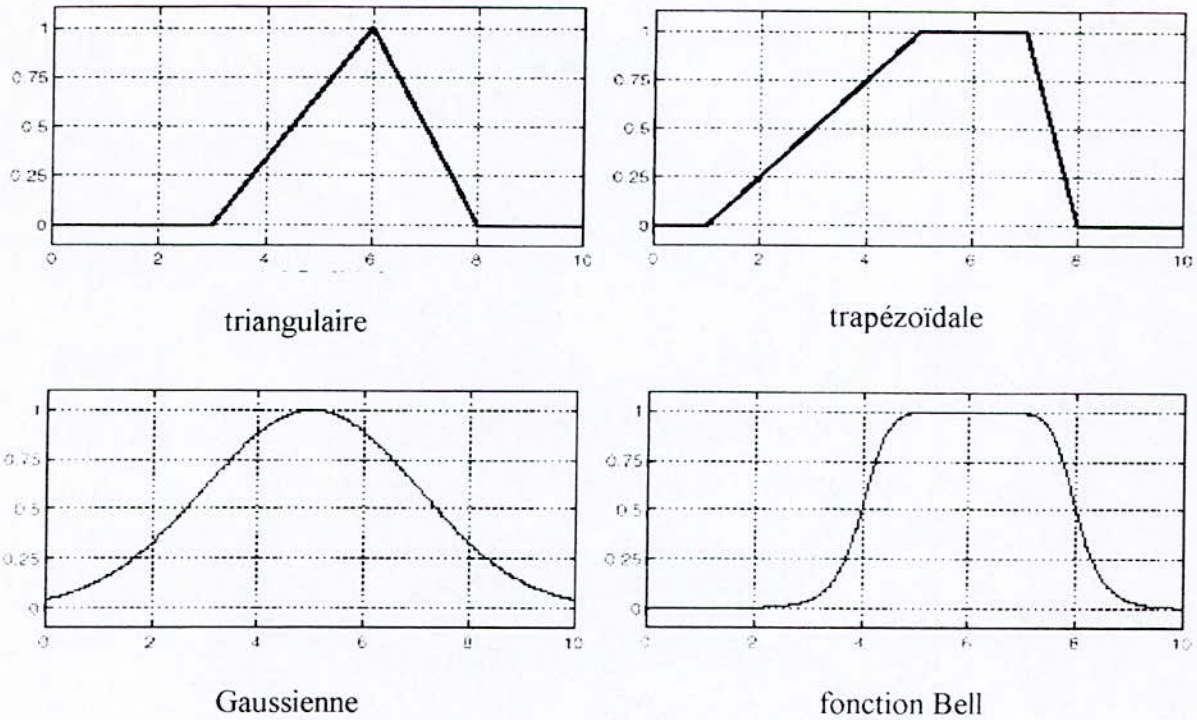


Figure 3.4 : Les fonctions d'appartenance les plus connues

3.2.3 Opérations sur les sous ensembles flous [16]

Le fait d'utiliser des sous ensembles flous pour décrire des classes vaguement et imparfaitement localisées conduit dans U à caractériser les éléments de U communs ou étrangers à ces différentes classes.

Comme la notion de sous-ensembles flous est une généralisation de la notion de sous ensembles classiques, on est amené à définir les différentes opérations entre ces derniers..

(i). Egalité et inclusion de sous ensembles :

Soient A , B deux sous ensembles flous de U avec μ_A , μ_B leurs fonctions d'appartenances respectives. On dit que les deux sous ensembles flous A et B sont égaux si leurs fonctions d'appartenance prennent la même valeur pour tout élément x de U :

$$\forall x \in U: \mu_A(x) = \mu_B(x) \quad (3.2)$$

On dit aussi que A est incluse dans B , si tout élément x de U appartenant à A appartient à B avec un degré au moins aussi grand :

$$\forall x \in U: \mu_A(x) \leq \mu_B(x) \quad (3.3)$$

(ii). Produit cartésien de sous ensembles flous :

A_1, \dots, A_n sont des sous ensembles flous dans U_1, \dots, U_n respectivement, le produit cartésien de A_1, \dots, A_n est un sous ensembles flous dans l'espace produit $U_1 \times \dots \times U_n$ avec la fonction d'appartenance :

$$\mu_{A_1 \times A_2 \times \dots \times A_n}(x_1, \dots, x_n) = \min(\mu_{A_1}(x_1), \dots, \mu_{A_n}(x_n)) \quad (3.4)$$

où :

$$\mu_{A_1 \times A_2 \times \dots \times A_n}(x_1, \dots, x_n) = \mu_{A_1}(x_1) \times \dots \times \mu_{A_n}(x_n) \quad (3.5)$$

(iii). Relation floue :

Soient U_1, \dots, U_n des univers de discussion, une relation floue d'ordre n est un sous ensemble flou dans $U_1 \times \dots \times U_n$ exprimé par :

$$R_{U_1 \times \dots \times U_n} = \{((x_1, \dots, x_n), (\mu_R(x_1, \dots, x_n))) / (x_1, \dots, x_n) \in U_1 \times \dots \times U_n\} \quad (3.6)$$

(iv). Intersection :

L'intersection de A et B est définie par :

$$\mu_{A \cap B} = T(\mu_A(x), \mu_B(x)) \quad (3.7)$$

où :

T : T-norm ou norme triangulaire .

T-norm satisfaisant les propriétés suivantes :

1. bornitude : $T(0,0) = 0, T(a,1) = T(1,a) = a$
2. monotonie : $T(a,b) \leq T(c,d)$ si $a \leq c$ et $b \leq d$
3. commutativité : $T(a,b) = T(b,a)$
4. associativité : $T(a, T(b,c)) = T(T(a,b),c)$

(v). Union :

L'union de A et B est définie par :

$$\mu_{A \cup B} = S(\mu_A(x), \mu_B(x)) \quad (3.8)$$

où :

S : S-norm ou T-conorm.

T-conorm satisfait les propriétés suivantes :

1. bornitude : $S(1,1) = 1$, $S(a,0) = S(0,a) = a$
2. monotonie : $S(a,b) \leq S(c,d)$ si $a \leq c$ et $b \leq d$
3. commutativité : $S(a,b) = S(b,a)$
4. associativité : $S(a, S(b,c)) = S(S(a,b), c)$

3.3 Logique floue [8]

On a vu la notion floue, mais pas sa logique jusqu'à maintenant. Cette section présente les règles de calcul pour la logique floue, il y a alors une certaine affinité avec les règles de la logique classique, par exemple :

☞ Opérateur ET, OU et NON :

Dans le cas de la logique classique où on trouve les tableaux de vérité suivants :

A	B	A ET B
0	0	0
0	1	0
1	0	0
1	1	1

ET

A	B	A OU B
0	0	0
0	1	1
1	0	1
1	1	1

OU

A	NON A
0	1
1	0

NON

Figure 3.5 : Tableaux de vérité des fonctions ET, OU et NON.

Et n'oublions pas que la logique floue considère que la vérité n'est pas absolue, mais à un degré bien défini, donc comment vont être affectés ces tableaux de vérité ?

Quelle fonction préservera les résultats de ET, OU et NON classiques de plus sera étendu à tous les nombres réels compris entre 0 et 1 ?

Une réponse est, l'opérateur 'min'. Ceci résoudra A ET B, quand A et B sont dans le rang (0,1), et le même raisonnement pour l'opérateur OU qui sera remplacé par 'max', et finalement l'opérateur NON par 1-A.

A	B	min(A,B)
0	0	0
0	1	0
1	0	0
1	1	1

ET

A	B	max(A,B)
0	0	0
0	1	1
1	0	1
1	1	1

OU

A	1-A
0	1
1	0

NON

Figure 3.6 : Tableaux de vérité des fonctions ET, OU et NON.

☞ L'implication :

Il y a plusieurs techniques pour obtenir une relation floue **R** basée sur la règle si-alors, ceci est connu comme l'implication. Dans ce qui suit nous allons mentionner les différentes techniques de calcul définies par :

lotfi Zadeh

$$\mu_R(x, y) = \max\{\min[\mu_A(x), \mu_B(y)], 1 - \mu_A(x)\} \tag{3.9}$$

Mamdani

$$\mu_R(x, y) = \min\{\mu_A(x), \mu_B(y)\} \tag{3.10}$$

Luckawic

$$\mu_R(x, y) = \min\{1, [1 - \mu_A(x) + \mu_B(y)]\} \tag{3.11}$$

et par G-del :

$$\mu_R(x, y) = \begin{cases} \mu_B(y) & \text{pour } \mu_B(y) < \mu_A(x) \\ 1 & \text{autrement} \end{cases} \tag{3.12}$$

3.4 Règles Si et Alors [8]

La variable floue et l'opérateur flou sont considérés comme le sujet et le verbe, la règle si-alors est utilisée pour formuler leur relation, on parle alors de déduction floue ou inférence, dans ce contexte on peut distinguer deux genres de règles d'inférences :

☞ inférence avec une seule règle :

si x est A alors y est B

où A et B sont des variables linguistiques définies dans le domaine (univers de discussion) X et Y respectivement, la partie 'si x est A' est appelée *antécédent*, et la partie 'alors y' est B est appelée *conséquence*, par exemple :

si le travail est bon alors la note est bonne

L'antécédent d'une règle peut prendre multiples parties :

si le ciel est gris ET le vent est fort alors la pluie peut tomber

Dans ce cas, toute la partie antécédent sera calculé simultanément, la conséquence d'une règle peut aussi avoir multiples parties comme suit :

si la température est basse alors la vanne de l'eau chaud est ouverte ET la vanne de l'eau froid est fermée.

dans ce cas, les conséquences seront toute actionnées par le résultat de l'antécédent.

☞ Inférence avec plusieurs règles :

Le cas d'une inférence avec plusieurs règles se présente lorsqu'une ou plusieurs variables nécessitent une prise de décision différente suivant les valeurs qu'atteignent ces variables. Ces dernières sont des variables linguistiques, donc des variables floues. Les règles peuvent alors être exprimées sous la forme générale :

si condition 1, alors opération 1, OU

si condition 2, alors opération 2, OU

si condition n, alors opération n.

3.5 Raisonnement flou [8]

Pour raisonner sur des connaissances vagues et imprécises, la logique classique ne suffit pas et on utilise par conséquent une logique floue, deux propositions floues x est A et x est A' sont considérées comme proches dans la mesure où les sous ensembles flous A et A' ont des fonctions d'appartenance peu différentes. Le raisonnement approximatif est un raisonnement sur des propositions floues qui tient compte de cette idée de proximité. Le *modus ponens*

défini dans la logique classique trouve son équivalent dans le cadre du raisonnement approximatif sous la forme du modus ponens généralisé.

De façon générale, les conditions d'utilisations du *modus ponens* généralisé sont les suivantes :

règle floue :	si x est A alors y est B
fait observé :	x est A'
Conséquence :	Y est B'

Les informations disponibles pour déterminer la conséquence sont donc d'une part celles relatives à la règle, quantifiées par l'implication floue, d'autre part celles relatives au fait observé, quantifiées par la fonction d'appartenance $\mu_{A'}$. Le modus ponens généralisé est particulièrement utilisé dans la commande par la logique floue, il peut être mis sous la forme suivante :

$$B' = A' \circ R \text{ où } R : \text{relation entre } A \text{ et } B. \quad (3.13)$$

\vee_x : étant le sup.

\vee : étant T-norm

☞ une seule règle avec un antécédent à un ensemble flou :

fait observé :	x est A'
règle floue :	si x est A alors y est B
Conséquence :	Y est B'

Cette règle est mise sous la forme :

$$\begin{aligned} \mu_{B'}(y) &= \vee_x [\mu_{A'}(x) \wedge \mu_R(x, y)] \\ &= \vee_x [\mu_{A'}(x) \wedge \mu_A(x) \wedge \mu_B(y)] \\ &= \vee_x (\mu_{A'}(x), \mu_A(x)) \wedge \mu_B(y) \\ &= W \wedge \mu_B(y) \end{aligned}$$

où W est le degré de l'occurrence.

☞ Une seule règle avec un antécédent à deux ensembles flous :

fait observé :	x est A' et y est B'
règle floue :	si x est A et y est B alors Z est C
conséquence :	Z est C'

La règle floue est interprétée par $(A \times B \rightarrow C)$, cette règle peut être transformée en une relation floue qui a la fonction d'appartenance suivante :

$$\mu_R(x, y, z) = \mu_{(A \times B) \times C}(x, y, z) = \mu_A(x) \wedge \mu_B(y) \wedge \mu_C(z) \quad (3.14)$$

et C' est : $C' = (A' \times B') \circ (A \times B \rightarrow C)$

donc :

$$\mu_{C'}(z) = W_1 \wedge W_2 \wedge \mu_C(z)$$

avec : W_1 est le degré d'occurrence entre A' et A , W_2 est le degré d'occurrence entre B' et B et $W_1 \wedge W_2$ est appelé degré d'accomplissement.

3.6 Réglage par logique floue

3.6.1 Structure d'un réglage par logique floue [3]

Le régulateur par logique floue fournit le signal à l'organe de commande, et reçoit à son entrée la consigne et un vecteur de mesure de sorties.

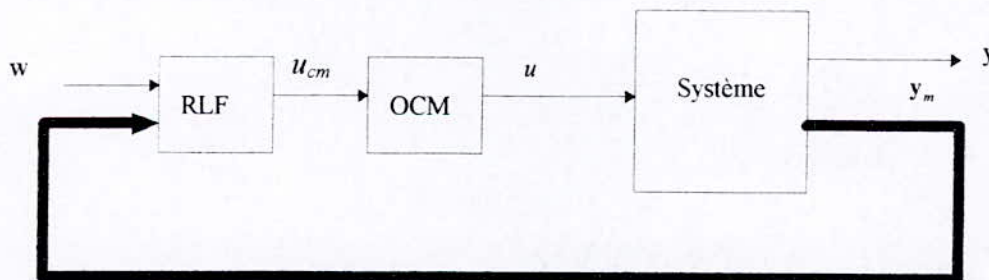


Figure 3.7 : Schéma de réglage par logique floue.

y_m : vecteur de mesure.

y : grandeur à régler.

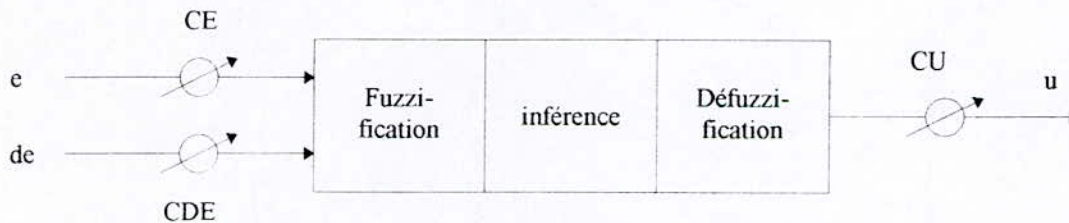
w : grandeur de consigne.

u : grandeur de commande est fournie par l'organe de commande.

u_{cm} : signal de commande.

3.6.2 Configuration interne d'un régulateur par logique floue [16]

Le schéma suivant montre la configuration interne d'un régulateur par logique floue, qui contient des gains d'implémentation, on peut distinguer trois parties :



- ☞ fuzzification
- ☞ inférence
- ☞ défuzzification

a. Fuzzification

Le bloc fuzzification contient en général un traitement de données préliminaire, par exemple la formation de l'écart de réglage ou la détermination de la variation d'une certaine grandeur, les grandeurs sont alors traitées comme variables linguistiques ce qui nécessite leur définition par des fonctions d'appartenances.

b. Inférences

La stratégie de réglage a des inférences adoptées, la formulation concrète des inférences dépend évidemment du comportement statique et dynamique du système à régler et de l'organe de commande, ainsi que des buts de réglages envisagés. Il existe plusieurs stratégies d'inférence, par exemple : Mamdani, Sugeno ...

c. Défuzzification

La sortie du contrôleur flou est aussi une variable floue, comme notre processus a besoin d'un signal de commande réel, il faut donc faire recourt à la défuzzification, il y a quelques approches dans ce contexte :

(i) Méthode de la valeur maximale

Le principe est de caractériser la variable floue de la commande avec une seule représentation :

$$\max U(u) = U(u_0) \quad \text{tel que } u \in U \quad (3.15)$$

la difficulté augmente quand il y a plus qu'un seul élément de U possédant une valeur maximale, donc on a plusieurs choix à faire, pour l'instant on peut prendre arbitrairement un des éléments qui a le plus grand grade d'appartenance.

(ii) Méthode de la moyenne des maximas

La première approche est un peu hasardeuse, donc on préfère une technique systématique qui se base sur le calcul de la moyenne des maximums. On dénote par u_j la variable pour laquelle la commande atteint un maximum.

$$U = \left\{ u_j, \max U(u_j) = U(u_j) / j = 1, 2, \dots \right\} \quad (3.16)$$

on suppose que la cardinalité est égale à r , $\text{card}(U) = r$ alors u_0 est calculée comme suit :

$$u_0 = \sum_{u_j \in U} \frac{u_j}{r}$$

(iii) Méthode du centre de gravité

Dans la technique de défuzzification par centre de gravité, u_0 est calculée comme suit :

$$u_0 = \frac{\int_U uU(u)du}{\int_U U(u)du} \quad (3.17)$$

nous supposons que les deux intégrales existe, on voit que u_0 détermine le centre de gravité de la surface de la commande floue, et en faisant une comparaison avec la première technique, nous observons que le résultat entrée-sortie utilisant la M.C.G ne contient pas de points de discontinuité.

Chapitre 4

*OBSERVATEURS NON
LINEAIRES*

4.1 Éléments de la théorie de l'observation [12]

L'observabilité des systèmes non linéaires constitue une notion plus complexe que celle des systèmes linéaires. En effet, pour les systèmes non linéaires, la notion d'observabilité dépend des entrées appliquées au système, et des états initiaux. Soit le système Σ_1 :

$$\dot{x} = f(x) + \sum_{i=1}^m u_i g_i(x) \quad (4.1.1.a)$$

$$y = h(x) \quad (4.1.1.b)$$

Définition 1 (Discernabilité - Indiscernabilité)

Soit $\mathfrak{N} \subset \mathfrak{R}^n$ un ouvert de \mathfrak{R}^n et soient deux états initiaux distincts x_1 et x_2 dans \mathfrak{N} nous pouvons dire que :

- ☞ x_1 est \mathfrak{N} -indiscernable de x_2 si pour tout $t \geq 0$ et toute entrée admissible u , les sorties $y(t, x_1, 0, u)$ et $y(t, x_2, 0, u)$ sont égales tant que les trajectoires $x_1(t)$ et $x_2(t)$ restent dans \mathfrak{N} .
- ☞ x_1 est \mathfrak{N} -discernable de x_2 s'il existe un temps $t_d \geq 0$ et une entrée u pour laquelle les fonctions $y(t, x_1, 0, u)$ et $y(t, x_2, 0, u)$ sont distinctes avec $x_1(t_d)$ et $x_2(t_d)$ restent dans \mathfrak{N} . Dans ce cas, on dit que u discerne x_1 et x_2 dans \mathfrak{N} .

Définition 2 (Observabilité locale)

Le système Σ_1 est dit observable localement en $x_0 \in M$ s'il existe un voisinage Λ de x_0 tel que, tout $x \in \Lambda (x \neq x_0)$ est Λ -discernable de x_0 .

Le système Σ_1 est dit localement observable s'il l'est en tout $x_0 \in M$.

Théorème [11]

Soit l'espace O , appelé espace d'observation, défini par le plus petit espace vectoriel contenant h_1, \dots, h_p et toutes leurs dérivées de Lie successives :

$$O = \text{span} \left\{ h_1, \dots, h_p, \mathcal{L}_{f+u_1g}(h_1), \dots, \mathcal{L}_{f+u_1g}(h_p), \dots, \mathcal{L}_{f+u_mg}(h_1), \dots, \mathcal{L}_{f+u_mg}(h_p), \dots, \mathcal{L}_{f+u_2g}(\mathcal{L}_{f+u_2g}(h_1)), \dots \right\}$$

Si $\dim O(x_0) = n$, alors le système Σ_1 est localement observable au voisinage de x_0 .

Définition 3 (Observabilité globale)

Le système Σ_1 est dit observable globalement sur M si tout couple d'états initiaux distincts $\{x_1, x_2\}$ est M -discernable.

Définition 4 (Entrée universelle)

Une entrée $u(t)$ est dite universelle pour le système Σ_1 sur un intervalle $[0, T]$ si tout couple d'états initiaux distincts $\{x_1, x_2\}$ peut être discerné par les sorties sur l'intervalle $[0, T]$, le système étant soumis à l'entrée $u(t)$. C'est à dire s'il existe un temps $t \in [0, T]$ tel que $y(t, x_1, 0, u) \neq y(t, x_2, 0, u)$. Une entrée est dite singulière si elle n'est pas universelle.

Voici un exemple illustrant les entrées universelles et singulières :

Exemple [10]: Considérons le système non linéaire suivant :

$$\begin{cases} \dot{\mathbf{x}} = \begin{bmatrix} 0 & u \\ -u & 0 \end{bmatrix} \mathbf{x} \\ y = x_1 \end{cases} \quad u \in \{0, 1\} \quad (4.1.2)$$

Il est facile de voir que tout couple d'états initiaux distincts est discernable pour le système soumis à l'entrée $u = 1$, par conséquent le système est observable. Cependant, l'entrée $u = 0$ ne permet pas de discerner deux valeurs différentes de x_2 , cette entrée est dite singulière.

4.2 Systèmes uniformément observables [5]

L'exemple précédent a mis en évidence des systèmes non linéaires observables dont des états sont indiscernables pour certaines entrées dites singulières. Nous allons caractériser les systèmes observables qui ne possèdent pas d'entrée singulière, appelés uniformément observables. Nous nous limitons au cas mono-sortie pour raison de simplicité.

Considérons le système non linéaire mono-sortie affine par rapport à la commande décrit par :

$$\begin{cases} \dot{x}(t) = f(x(t)) + \sum_{i=1}^m u_i g_i(x(t)) \\ y(t) = h(x(t)) \end{cases} \quad (4.2.1)$$

où $x(t) \in \mathfrak{R}^n$, $u(t) \in \mathfrak{R}^m$, $y(t) \in \mathfrak{R}$, soit l'hypothèse suivante :

$$\psi : \begin{cases} \mathbb{R}^n \rightarrow \mathbb{R}^n \\ x \rightarrow z = \psi(x) = [h(x), \mathcal{L}_f h(x), \dots, \mathcal{L}_f^{n-1} h(x)]^T \end{cases} \quad (4.2.2)$$

Constitue un difféomorphisme local ($\det(\frac{\partial \psi}{\partial x}) \neq 0$ au voisinage du point considéré).

Théorème : Considérons le système (4.2.1). Sous l'hypothèse précédente, et si (4.2.1) est uniformément observable pour toute entrée, alors ψ transforme le système (4.2.1) sous la forme :

$$\left\{ \begin{array}{l} \dot{z}(t) = \begin{bmatrix} \dot{z}_1(t) \\ \dot{z}_2(t) \\ \vdots \\ \dot{z}_{n-1}(t) \\ \dot{z}_n(t) \end{bmatrix} = \begin{bmatrix} z_2(t) \\ z_3(t) \\ \vdots \\ z_n(t) \\ 0 \end{bmatrix} + \begin{bmatrix} 0 \\ 0 \\ \vdots \\ 0 \\ \varphi(z(t)) \end{bmatrix} + \sum_{i=1}^m u_i(t) \begin{bmatrix} g_{i,1}(z_1(t)) \\ g_{i,2}(z_1(t), z_2(t)) \\ \vdots \\ g_{i,n}(z_1(t), \dots, z_n(t)) \end{bmatrix} \\ y = [1 \ 0 \ 0 \ 0 \ 0 \ 0]z(t) = z_1(t) \end{array} \right. \quad (4.2.3)$$

Pour la démonstration du théorème, le lecteur peut se référer à [5].

4.3 Synthèses d'observateurs pour les systèmes non linéaires

4.3.1 Position du problème

Après avoir exposé succinctement les principaux concepts relatifs à l'observabilité des systèmes non linéaires, nous allons présenter différentes approches de synthèse d'observateurs pour ces systèmes. Soit le système non linéaire suivant :

$$\Sigma : \begin{cases} \dot{x}(t) = f(x(t), u(t)) \\ y(t) = h(x(t)) \end{cases} \quad (4.3.1)$$

On appelle observateur d'état, un système dynamique Σ_0 qui permet d'estimer les variables non mesurées d'un système dynamique Σ à partir des entrées et des sorties de ce système, les entrées de l'observateur sont constituées des entrées et des sorties du système Σ , et la sortie de l'observateur constitue l'état estimé. La figure 4.3.1 donne le schéma de principe d'un tel observateur.

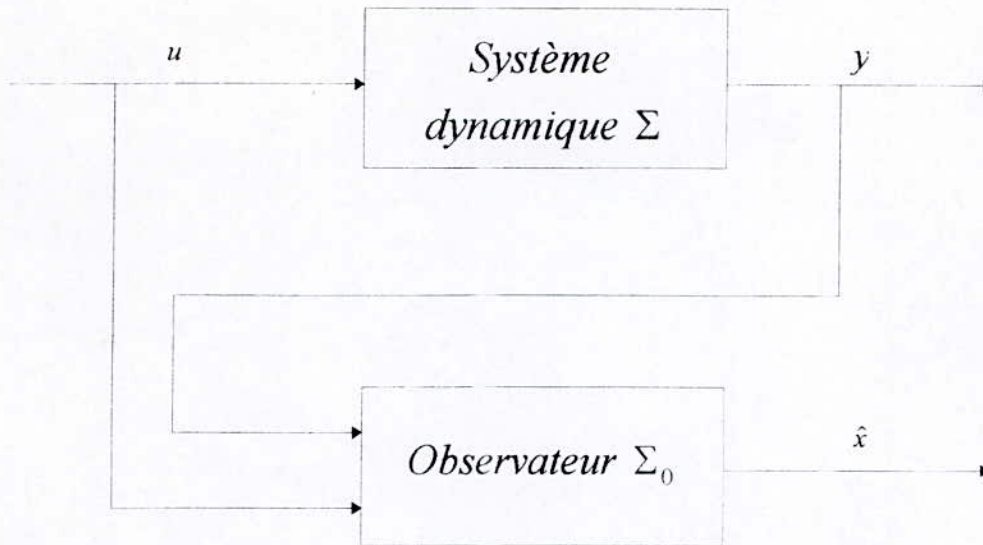


Figure 4.3.1 : Schéma global de l'observateur

4.3.2 Observateur de Thau

Les résultats obtenus par Thau ont été généralisés par Kou et al, et Banks. Cette méthode ne constitue pas une technique systématique pour la synthèse d'un observateur, mais donne une condition suffisante de stabilité exponentielle de l'erreur d'observation.

Considérons un système non linéaire pouvant se mettre sous la forme :

$$\begin{cases} \dot{x} = Ax + Bu + f(x) \\ y = Cx \end{cases} \quad (4.3.2)$$

où : $x(t) \in \mathfrak{R}^n$ représentent l'état, le champ $f: \mathfrak{R}^n \rightarrow \mathfrak{R}^n$ est différentiable, $u(t) \in \mathfrak{R}^m$ est le vecteur des entrées, $y(t) \in \mathfrak{R}^p$ est le vecteur des mesures.

Théorème [11] : Considérons le système (4.3.2), pour lequel :

☞ la fonction f est globalement Lipschitzienne : $\|f(x_1) - f(x_2)\| \leq k\|x_1 - x_2\|$ pour tout $x_1, x_2 \in \mathfrak{R}^n$, avec $k > 0$.

☞ la paire (C, A) est détectable : Il existe une matrice $L \in \mathfrak{R}^{n \times p}$ telle que la matrice $A_0 = A - LC$ ait un spectre contenu dans le demi plan gauche, par conséquent pour toute matrice Q symétrique définie positive, il existe une matrice $P \in \mathfrak{R}^{n \times n}$ symétrique définie positive solution de l'équation algébrique de Lyapunov :

$$A_0^T P + P A_0 = -2Q \quad (4.3.3)$$

Les valeurs propres des matrices P et Q vérifient l'inégalité suivante : $\frac{\lambda_m(Q)}{\lambda_M(P)} > k$

où : $\lambda_m(\cdot)$ et $\lambda_M(\cdot)$ désignent respectivement la plus petite et la plus grande valeur propre.

Alors le système :

$$\dot{\hat{x}} = A\hat{x} + Bu + f(\hat{x}) + L(y - C\hat{x}) \quad (4.3.4)$$

constitue un observateur exponentiel de l'état du système.

4.3.3 Observateur glissant [15] [19]

On considère le système suivant :

$$\ddot{x}_1 = f \quad (4.3.5)$$

où f est une fonction non linéaire de $\mathbf{x} = [x_1 \quad x_2]^T$, construisant maintenant un observateur glissant sachant qu'on peut mesurer que x_1 , donc sa structure est de la forme suivante :

$$\dot{\hat{x}}_1 = -\alpha_1 e_1 + \hat{x}_2 - k_1 \text{sign}(e_1) \quad (4.3.6.a)$$

$$\dot{\hat{x}}_2 = -\alpha_2 e_1 + \hat{f} - k_2 \text{sign}(e_1) \quad (4.3.6.b)$$

où $e_1 = \hat{x}_1 - x_1$, \hat{f} est l'estimation de f , et les constantes α_i sont choisis comme dans le cas de l'observateur de Luenberger (qui correspond à $k_1 = 0, k_2 = 0$). Ainsi on fait un placement de pôles pour le système linéarisé. La dynamique de l'erreur s'écrit comme suit :

$$\dot{e}_1 = -\alpha_1 e_1 + e_2 - k_1 \text{sign}(e_1) \quad (4.3.7.a)$$

$$\dot{e}_2 = -\alpha_2 e_1 + \Delta f - k_2 \text{sign}(e_1) \quad (4.3.7.b)$$

On suppose que $\Delta f = \hat{f} - f$ est explicitement bornée, par le biais de la technique de majoration des non linéarités.

Si on est sur la surface de glissement le système devient :

$$\begin{aligned} \dot{e}_1 &= 0 \\ \dot{e}_2 &= \frac{k_2}{k_1} e_2 + \Delta f \end{aligned} \quad (4.3.8)$$

avec un placement de pôle adéquat on a :

$$k_2 > \Delta f \quad (4.3.9)$$

$$k_2 = p \cdot k_1 \quad \text{avec } p \text{ est un pôle qui appartient au demi plan gauche} \quad (4.3.10)$$

Cette méthodologie peut être étendue au système d'ordre n sous la forme compagne de commande :

$$x_1^{(n)} = f \quad (4.3.11)$$

où x_1 est la seule mesure disponible, donc la structure de l'observateur est de la forme :

$$\begin{aligned} \dot{\hat{x}}_1 &= -\alpha_1 e_1 + \hat{x}_2 - k_1 \text{sign}(e_1) \\ &\dots \\ \dot{\hat{x}}_n &= -\alpha_n e_1 + \hat{f} - k_n \text{sign}(e_1) \end{aligned} \quad (4.3.12)$$

avec :

$$\det(p\mathbf{I}_{n-1} - \begin{bmatrix} -k_2/k_1 & 1 & 0 & \dots & 0 \\ & & & & 1 \\ -k_n/k_1 & 0 & 0 & \dots & 0 \end{bmatrix}) = 0 \quad (4.3.13)$$

où \mathbf{I}_{n-1} est une matrice d'identité d'ordre $n-1$, avec :

$$k_n \geq |\Delta f|$$

Et maintenant nous allons exploiter la structure de l'observateur glissant à l'aide des exemples suivants :

4.3.3.1 Extension au cas général des systèmes non linéaires [19]

Soit le système :

$$\dot{x} = f(x, t) \quad x \in \mathfrak{R}^n \quad (4.3.14.a)$$

$$y = Cx \quad y \in \mathfrak{R}^p \quad (4.3.14.b)$$

on définit l'observateur par la structure :

$$\dot{\hat{x}} = f(\hat{x}, t) - H(y - \hat{y}) - kI_s \quad (4.3.15)$$

où :

k : matrice de gain de dimension $n \times p$

I_s : vecteur de dimension p

\hat{f} : représente notre modèle

H : matrice de gain de dimension $n \times p$

avec :

$$I_s = [\text{sign}(\tilde{y}_1) \quad \dots \quad \text{sign}(\tilde{y}_p)]^T$$

où :

$$\tilde{y}_i = c_i \hat{x} - y_i \quad (4.3.16)$$

On définit la surface comme étant le vecteur de l'erreur :

$$s = \tilde{y} = C(\hat{x} - x) \quad (4.3.17)$$

$$\tilde{x} = \hat{x} - x$$

nous avons :

$$\dot{\tilde{x}} = \Delta f - H\tilde{y} - K\tilde{s} \quad (4.3.18)$$

la surface de dimension p doit être attractive si : $s_i \dot{s}_i < 0, i = 1, \dots, p$. Les matrices H et K sont choisies de manière à satisfaire cette condition. La méthode de synthèse de H et K dépend du cas traité, c'est-à-dire de la forme de f .

En utilisant le principe de la commande équivalente on a : $s = 0, \dot{s} = 0$, on peut exprimer cette condition comme suit :

$$\text{grad}(s)\tilde{f}(\tilde{x}, \tilde{I}_s) = 0 \quad (4.3.19.a)$$

$$\tilde{f} = \Delta f - H\tilde{y} - K\tilde{I}_s \quad (4.3.19.b)$$

\tilde{I}_s est le vecteur de glissement équivalent. on a :

$$\tilde{I}_s = (CK)^{-1}C\Delta f \quad (4.3.20)$$

donc la dynamique sur la surface de glissement est :

$$\dot{\tilde{x}} = (I - K(CK)^{-1}C)\Delta f \quad (4.3.21)$$

4.3.4 Observateur basé sur la méthode du Backstepping [2]

La méthode de backstepping peut être utilisée comme une méthode de construction systématique d'un observateur pour les systèmes non linéaires, qui converge exponentiellement.

A chaque étape de la procédure de conception de cet observateur, les erreurs d'observation non mesurables sont considérées comme perturbations et on contourne le problème par l'utilisation du compensation non linéaire « *nonlinear damping* ». Voici la procédure de construction :

1. Choix de la fonction de Lyapunov candidate, en fonction de l'état mesuré.

$$V = \frac{1}{2}e^2 \quad \text{avec : } e = y - \hat{y}$$

2. Procéder par une majoration où la compensation non linéaire sera utilisée pour contourner le problème des états inaccessibles.

Exemple : Soit le système :

$$\dot{\xi} = x \quad (4.3.22.a)$$

$$\dot{x} = -x + \sin \xi + u \quad (4.3.22.b)$$

notre observateur a la forme suivante :

$$\dot{\hat{\zeta}} = \hat{x} + \Lambda_1 \quad (4.3.23.a)$$

$$\dot{\hat{x}} = -\hat{x} + \sin \hat{\zeta} + u + \Lambda_2 \quad (4.3.23.b)$$

la dynamique de l'erreur :

$$\dot{e}_1 = e_2 - \Lambda_1 \quad (4.3.24.a)$$

$$\dot{e}_2 = -e_2 + \sin \zeta - \sin \hat{\zeta} - \Lambda_2 \quad (4.3.24.b)$$

on suppose que :

$$v = \sin \zeta - \sin \hat{\zeta}$$

avec :

$$e_1 = \zeta - \hat{\zeta}$$

$$e_2 = x - \hat{x}$$

donc on a :

$$\dot{e}_1 = e_2 - \Lambda_1 \quad (4.3.25.a)$$

$$\dot{e}_2 = -e_2 + v - \Lambda_2 \quad \text{avec } v < 2 \quad (4.3.25.b)$$

Etape 1

$$V(e_1) = \frac{1}{2}e_1^2$$

La dérivée de cette fonction est :

$$\begin{aligned} \dot{V}(e) &= \dot{e}_1 e_1 \\ &= e_1 (e_2 - \Lambda_1) \\ &< a |e_1| - e_1 \Lambda_1 \end{aligned}$$

avec :

$$|e_1| = e_1 \cdot \text{sign}(e_1) \quad \text{et} \quad |e_2| < a \quad (4.3.26)$$

donc :

$$\dot{V}(e_1) < e_1 (a \text{sgn} e_1 - \Lambda_1) \quad (4.3.27)$$

si on prend :

$$\Lambda_1 = k_1 e_1 + a \text{sgn} e_1 \quad \text{avec : } k_1 > 0 \quad (4.3.28)$$

enfin :

$$\dot{V}(e) < -k_1 e_1^2$$

Etape 2

$$V(e_1) = \frac{1}{2} e_1^2 + \frac{1}{2} e_2^2$$

La dérivée de cette fonction :

$$\begin{aligned} \dot{V}(e_1, e_2) &= \dot{e}_1 e_1 + \dot{e}_2 e_2 \\ &< -k_1 e_1^2 + e_2(-e_2 + v - \Lambda_2) \\ &< -k_1 e_1^2 - e_2^2 + 2|e_2| + |e_2| \Lambda_2 \end{aligned}$$

donc :

$$\dot{V}(e_1, e_2) < -k_1 e_1^2 - e_2^2 + 2a + a\Lambda_2 \quad \text{et} \quad \Lambda_2 > 0$$

Si on prend :

$$\Lambda_2 = \frac{k_2 e_1^2 - 2a}{a} \quad \text{avec } k_2 > 0 \quad (4.3.29)$$

enfin :

$$\dot{V}(e_1, e_2) < -(k_1 - k_2) e_1^2 - e_2^2 \quad \text{et} \quad k_1 > k_2$$

On peut agir sur la dynamique de l'observateur en changeant les gains k_1 et k_2 .

4.4 Stabilisation via un observateur**4.4.1 Position du problème**

Supposant :

- un système non linéaire :

$$\dot{x} = f(x, u) \quad (4.4.1.a)$$

$$y = Cx \quad (4.4.1.b)$$

où l'état $x \in \mathfrak{R}^n$, la sortie $y \in \mathfrak{R}^p$ et l'entrée $u \in \mathfrak{R}^m$.

- un observateur pour ce système :

$$\dot{\hat{x}} = f_0(\hat{x}, y, u) \quad (4.4.2)$$

- une commande stabilisante :

$$u = u(x)$$

On considère le compensateur dynamique constitué par l'association de l'observateur, et de la loi de commande utilisant l'état estimé :

$$\dot{\hat{x}} = f_0(\hat{x}, y, u(\hat{x})) \quad (4.4.3.a)$$

$$u = u(\hat{x}) \quad (4.4.3.a)$$

Contrairement au cas des systèmes linéaires, la stabilité du système bouclé n'est pas garantie, donc des conditions supplémentaires doivent être ajoutées.

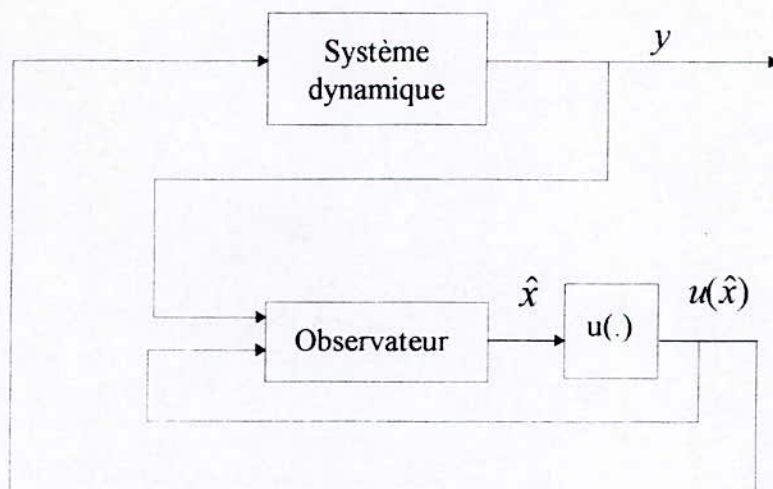


Figure 4.4.1 : Schéma du compensateur dynamique

4.4.2 Stabilisation exponentielle globale

Théorème : [11] Soit le système non linéaire (4.4.1), si les hypothèses suivantes sont vérifiées :

- ☛ L'observateur est à erreur globalement, uniformément et exponentiellement stable de l'état du système.
- ☛ Il existe une loi de commande telle que le système sans observateur soit globalement et exponentiellement stable.

Alors le système bouclé est globalement et exponentiellement stable. pour la démonstration le lecteur peut se référer à la bibliographie mentionnée au-dessus.



Chapitre 5

*APPLICATION AU
SIMULATEUR
D'HELICOPTERE*

A. Commande par retour d'état non linéaire

5.1 Commande avec linéarisation par retour d'état

Pour bien découvrir les caractéristiques de notre système, telles que : degré relatif, types de non linéarités (polynomiales, trigonométriques, ...) et termes de couplage, on va utiliser la commande avec linéarisation par retour d'état. Cette commande permet de bénéficier de la théorie des systèmes linéaires avec une rigoureuse mathématique.

Le fondement théorique de cette méthode a été détaillé au chapitre 2. Puisque notre système s'écrit sous la forme générale, on utilise la linéarisation entrée-sortie. Notre système d'application est modélisé dans le chapitre 1, il est composé de deux sous système (u_1, ψ) , (u_2, φ) avec des termes de couplage. Notre objectif est de stabiliser le système.

Maintenant voici les équations d'état qui représentent la décompositions :

1^{er} sous système :

$$\begin{aligned}
 \dot{x}_1 &= x_2 \\
 \dot{x}_2 &= -\alpha_1 x_2 - \alpha_2 \sin x_1 + \alpha_3 x_3^2 + \alpha_4 x_3 - \beta x_3 x_6 \cos x_1 \\
 \dot{x}_3 &= x_4 \\
 \dot{x}_4 &= -\alpha_6 x_3 - \alpha_5 x_4 + \alpha_6 u_1
 \end{aligned} \tag{5.1.1}$$

2^{ème} sous système :

$$\begin{aligned}
 \dot{x}_5 &= x_6 \\
 \dot{x}_6 &= -\alpha_9 x_6 + \alpha_7 x_7^2 + \alpha_8 x_7 - \alpha_{10} u_1 \\
 \dot{x}_7 &= x_8 \\
 \dot{x}_8 &= -\alpha_{11} x_8 - \alpha_{12} x_7 + \alpha_{12} u_2
 \end{aligned} \tag{5.1.2}$$

La figure (5.1.1) montre mieux cette décomposition qui est inspirée de la construction physique du simulateur d'hélicoptère.

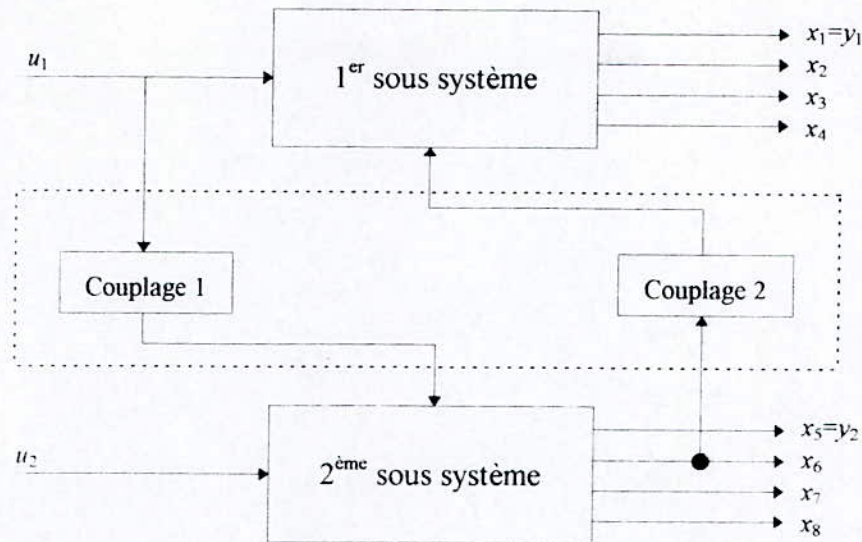


Figure 5.1.1 : Représentation du système décomposé

5.1.1 Application au 1^{er} sous système

On prend la première sortie et on fait la dérivation jusqu'à trouver une relation de la forme :

$$y^{(r)} = f(y, \dot{y}, \dots, y^{(r-1)}, u_1) \quad (5.1.3)$$

en effet :

$$\begin{aligned} y_1 &= x_1 \\ \dot{y}_1 &= x_2 \\ \ddot{y}_1 &= \dot{x}_2 \\ \ddot{y}_1 &= -\alpha_1 x_2 - \alpha_2 \sin(x_1) + \alpha_3 x_3^2 + \alpha_4 x_3 - \beta x_3 x_6 \cos(x_1) \\ \ddot{y}_1 &= f_{11}(x, u_1) \end{aligned} \quad (5.1.4)$$

La commande apparaît à cette étape, le système est donc de degré relatif trois. Mais le terme $\beta x_3 x_6 \cos(x_1)$ où apparaît la commande est un terme de couplage (x_6) qui est en fait faible, et qui peut influencer la stabilité du 2^{ème} sous système ; on continue donc à dériver jusqu'à l'ordre complet du système ($n = 4$) :

$$y_1^{(4)} = f_{12}(x, u_1) \quad (5.1.5)$$

suivant le chapitre 2 :

$$\begin{aligned} y_1^{(4)} &= v_1 \\ y_1^{(4)} &= -k_{11} y_1 - k_{12} \dot{y}_1 - k_{13} \ddot{y}_1 - k_{14} \ddot{y}_1 \end{aligned} \quad (5.1.6)$$

les gains $k_i, i = 1, 4$ sont choisis suivant un placement de pôles approprié. La commande u_1 résultante est :

$$u_1 = f_1(x) / \alpha_6(2\alpha_3x_3 + \alpha_4) \quad (5.1.7)$$

Si on choisit le polynôme caractéristique $p^4 + 8.p^3 + 24.p^2 + 32.p + 16$ la fonction f_1 sera :

$$f_1 = -[23x_2 - 182 \sin x_1 - 34x_3^2 + 21x_3 - 63x_2 \cos x_1 + 44x_3x_4 + 1.95x_4 + 9x_2^2 \sin x_1 + 38 \sin 2x_1 - 210x_3 \cos x_1 + 48x_4^2 + 16x_1] \quad (5.1.8)$$

5.1.2 Application au 2^{ème} sous système

En suivant les mêmes étapes précédentes, on aura :

$$\begin{aligned} y_2 &= x_5 \\ \dot{y}_2 &= x_6 \\ \ddot{y}_2 &= -\alpha_9x_6 + \alpha_7x_7^2 + \alpha_8x_7 - \alpha_{10}u_1 \\ \ddot{\ddot{y}}_2 &= f_{21}(x, u_1, \dot{u}_1) \\ y_2^{(4)} &= f_{22}(x, u_1, \dot{u}_1, \ddot{u}_1, u_2) \end{aligned} \quad (5.1.9)$$

avec :

$$f_{21}(x, u_1, \dot{u}_1) = 2\alpha_7x_7x_8 + \alpha_8x_8 - \alpha_9\alpha_7x_7^2 - \alpha_8\alpha_9x_7 + \alpha_9^2x_6 + \alpha_9\alpha_{10}u_1 - \alpha_{10}\dot{u}_1$$

$$\begin{aligned} f_{22}(x, u_1, \dot{u}_1, \ddot{u}_1, u_2) &= [-2\alpha_7(\alpha_9 + \alpha_{11})x_7x_8 + 2\alpha_7x_8^2 + \alpha_7(\alpha_9^2 - 2\alpha_{12})x_7^2 - \alpha_8(\alpha_{11} + \alpha_9)x_8 \\ &\quad - \alpha_9^3x_6 + \alpha_8(\alpha_9^2 - \alpha_{12})x_7 - \alpha_9^2\alpha_{10}u_1 + \alpha_9^2\dot{u}_1 - \alpha_9\ddot{u}_1 + \alpha_{12}(2\alpha_7x_7 + \alpha_8)u_2] \end{aligned}$$

suitant la méthode :

$$y_2^{(4)} = -k_{21}y_{21} - k_{22}\dot{y}_2 - k_{23}\ddot{y}_2 - k_{24}\ddot{\ddot{y}}_2 \quad (5.1.10)$$

en choisissant le polynôme caractéristique $p^4 + 8.p^3 + 24.p^2 + 32.p + 16$ la commande résultante est :

$$u_2 = - \frac{[-33.5x_7x_8 + 16x_8^2 - 162x_7^2 - 15x_8 + 7.6x_6 - 31x_7 - 16.2u_1 - 8.3\dot{u}_1 - 1.4\ddot{u}_1 + 16x_5]}{(255x_7 + 114)}$$

Cette commande est en fonction de u_1, \dot{u}_1 et \ddot{u}_1 à cause du terme de couplage de u_1 vers le 2^{ème} sous système.

5.1.3 Résultats de simulation :

Les figures (5.1, 2,3,4) représentent les simulations Pour la commande linéarisation par retour d'états. La figure 5.1.2 donne des résultats satisfaisants, car les états du système n'ont rencontré aucun point singulier ($x_3 = -0.045$ rd/s, $x_7 = -0.43$ rd/s) à cause du choix des conditions initiales.

Par contre dans la figure 5.1.3, le système diverge sous l'effet de la perturbation paramétrique (de 20 %) au point singulier correspondant à ($x_7 = -0.43$ rd/s).

Cependant dans la figure 5.1.4, une dégradation de la dynamique est observée, ceci revient au termes de couplage.

Simulation du modèle avec la commande de Linéarisation par retour d'état

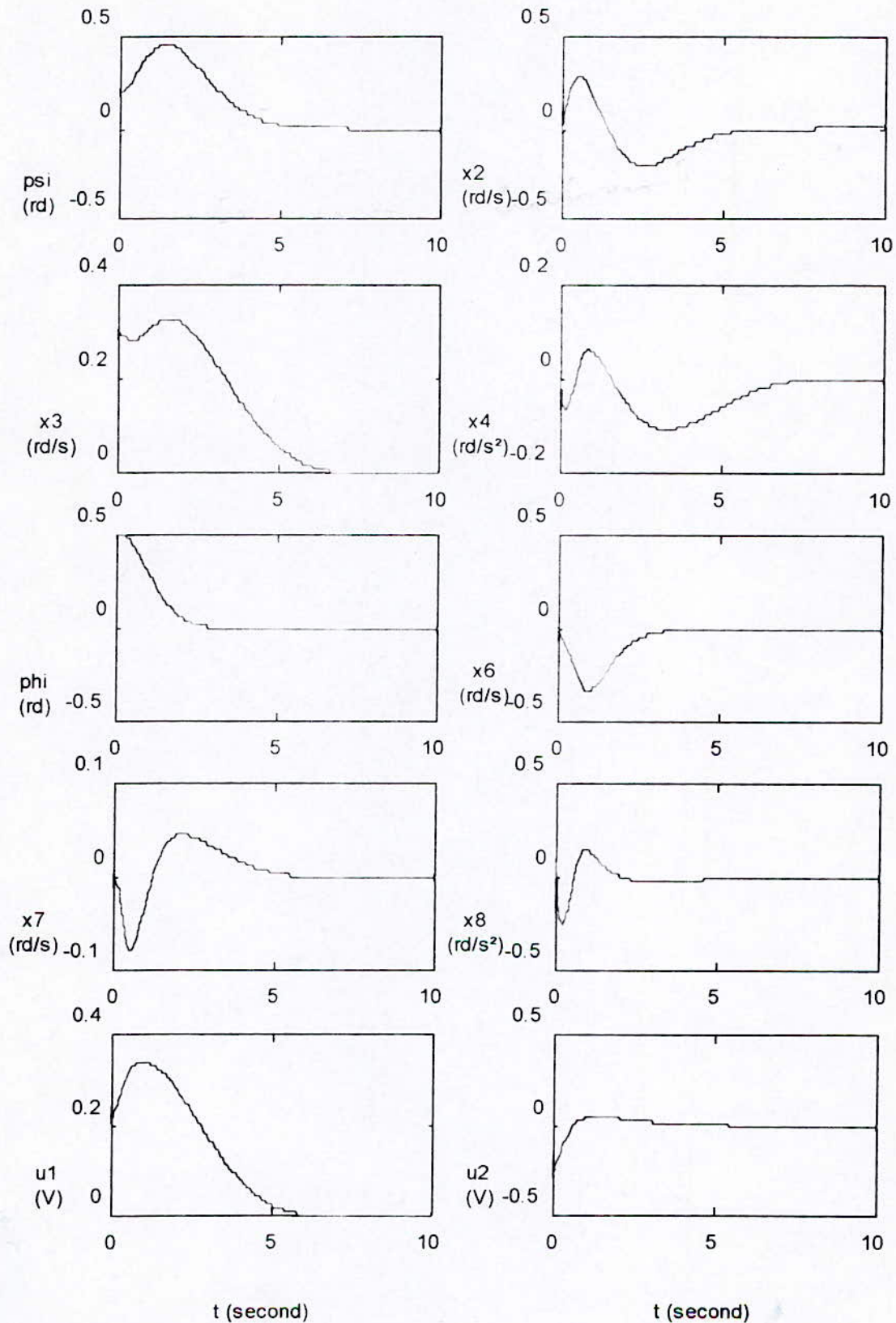


Figure 5.1.2 : Simulation avec : $p_1=-2$, $p_2=-3$, $\psi(0)=0.2$, $\phi(0)=0.5$, $x_3(0)=0.3$

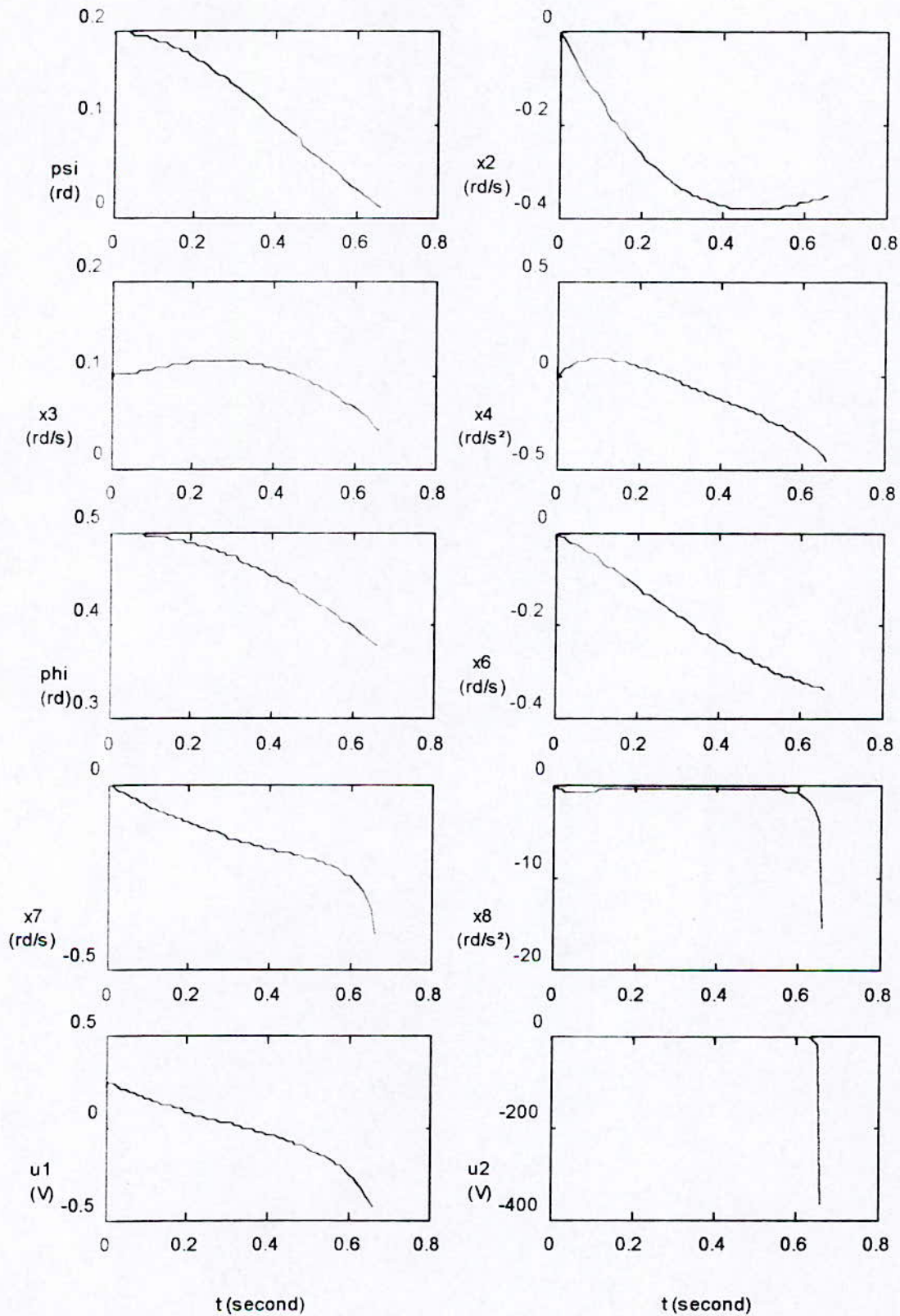


Figure 5.1.3 : Simulation avec : $p_1=-2$, $p_2=-3$, $\psi(0)=0.2$, $\phi(0)=0.5$, $x_3(0)=0.1$ et une perturbation de 20%

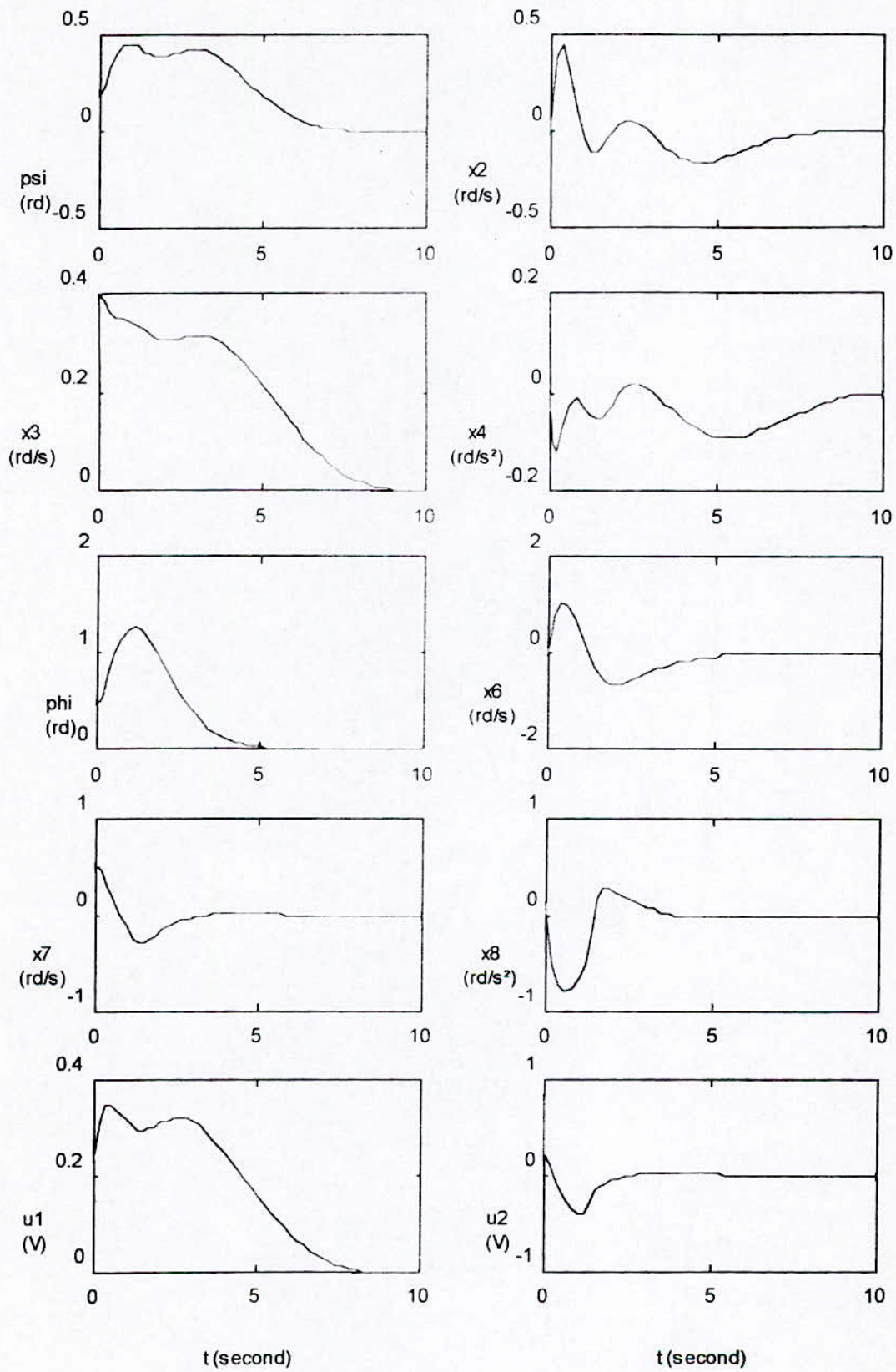


Figure 5.1.4 : Simulation avec : $p_1=-2$, $p_2=-2$, $\psi(0)=0.2$, $\phi(0)=0.5$, $x_3(0)=0.4$, $x_7(0)=0.5$

5.2 Commande à structure variable [4]

La commande avec linéarisation par retour d'état comporte des insuffisances majeures, telles que : points singuliers, non robustesse vis à vis des perturbations et elle exige la connaissance exacte du modèle mathématique. Pour toutes ces raisons, on a fait appel à la commande à structure variable.

Pour cela, nous allons synthétiser la commande à l'aide de la théorie exposée dans le chapitre 2.

5.2.1 Application

On considère toutes les non linéarités de notre système comme perturbations ; ainsi le nouveau modèle proposé dans ce travail en vue de la commande est :

$$\begin{aligned}
 \dot{x}_1 &= x_2 \\
 \dot{x}_2 &= -\alpha_1 x_2 + \alpha_4 x_3 + v_1 \\
 \dot{x}_3 &= x_4 \\
 \dot{x}_4 &= -\alpha_6 x_3 - \alpha_5 x_4 + \alpha_6 u_1 \\
 \dot{x}_5 &= x_6 \\
 \dot{x}_6 &= -\alpha_9 x_6 + \alpha_8 x_7 + v_2 \\
 \dot{x}_7 &= x_8 \\
 \dot{x}_8 &= -\alpha_{11} x_8 - \alpha_{12} x_7 + \alpha_{12} u_2
 \end{aligned} \tag{5.2.1}$$

avec :

$$\begin{aligned}
 v_1 &= -\alpha_2 \sin x_1 + \alpha_3 x_3^2 - \beta x_3 x_6 \cos x_1 \\
 v_2 &= \alpha_7 x_7^2 - \alpha_{10} u_1
 \end{aligned} \tag{5.2.2}$$

donc on peut majorer les perturbations v_1 et v_2 de telle sorte à avoir les inégalités suivantes :

$$\begin{aligned}
 v_1 &\leq a \\
 v_2 &\leq b
 \end{aligned} \tag{5.2.3}$$

où :

$$\begin{aligned}
 a &= \alpha_2 + \alpha_3 x_{3_{\max}}^2 + \beta x_{3_{\max}} x_{6_{\max}} \\
 b &= \alpha_7 x_{7_{\max}}^2 + \alpha_{10} u_{1_{\max}}
 \end{aligned} \tag{5.2.4}$$

On revient au système, avec une écriture sous forme matricielle en deux sous systèmes indépendants :

$$\dot{\mathbf{x}}_s = \mathbf{A}_s \mathbf{x}_s + \mathbf{B}_s u_1 + \mathbf{D}_s v_1 \tag{5.2.5.a}$$

$$\dot{\mathbf{x}}_p = \mathbf{A}_p \mathbf{x}_p + \mathbf{B}_p u_2 + \mathbf{D}_p v_2 \tag{5.2.5.b}$$

où :

$$\mathbf{x}_s^T = [x_1 \quad x_2 \quad x_3 \quad x_4]$$

$$\mathbf{x}_p^T = [x_5 \quad x_6 \quad x_7 \quad x_8]$$

$$\mathbf{B}_s^T = [0 \quad 0 \quad 0 \quad \alpha_6] ; \quad \mathbf{B}_p^T = [0 \quad 0 \quad 0 \quad \alpha_{12}]$$

$$\mathbf{A}_s = \begin{bmatrix} 0 & 1 & 0 & 0 \\ 0 & -\alpha_2 & \alpha_4 & 0 \\ 0 & 0 & 0 & 1 \\ 0 & 0 & -\alpha_6 & -\alpha_5 \end{bmatrix} ; \quad \mathbf{D}_s = \begin{bmatrix} 0 \\ 1 \\ 0 \\ 0 \end{bmatrix}$$

$$\mathbf{A}_p = \begin{bmatrix} 0 & 1 & 0 & 0 \\ 0 & -\alpha_8 & \alpha_8 & 0 \\ 0 & 0 & 0 & 1 \\ 0 & 0 & -\alpha_{12} & -\alpha_{11} \end{bmatrix} ; \quad \mathbf{D}_p = \begin{bmatrix} 0 \\ 1 \\ 0 \\ 0 \end{bmatrix}$$

On suppose que la surface de glissement est:

$$S_1 = -\mathbf{K}_s^T \mathbf{x}_s = -k_1 x_1 - k_2 x_2 - k_3 x_3 - k_4 x_4 \quad (5.2.6.a)$$

$$S_2 = -\mathbf{K}_p^T \mathbf{x}_p = -k_5 x_5 - k_6 x_6 - k_7 x_7 - k_8 x_8 \quad (5.2.6.b)$$

La détermination des paramètres k_i se fait par la théorie exposée dans le chapitre 2, donc il faut calculer \dot{S}_1 et \dot{S}_2 :

$$\dot{S}_1 = -\mathbf{K}_s^T \dot{\mathbf{x}}_s = -k_1 \dot{x}_1 - k_2 \dot{x}_2 - k_3 \dot{x}_3 - k_4 \dot{x}_4 \quad (5.2.7.a)$$

$$\dot{S}_2 = -\mathbf{K}_p^T \dot{\mathbf{x}}_p = -k_5 \dot{x}_5 - k_6 \dot{x}_6 - k_7 \dot{x}_7 - k_8 \dot{x}_8 \quad (5.2.7.b)$$

donc :

$$\dot{S}_1 = -\mathbf{K}_s^T \dot{\mathbf{x}}_s = -\mathbf{K}_s^T (\mathbf{A}_s \mathbf{x}_s + \mathbf{B}_s u_1) \quad (5.2.8.a)$$

$$\dot{S}_2 = -\mathbf{K}_p^T \dot{\mathbf{x}}_p = -\mathbf{K}_p^T (\mathbf{A}_p \mathbf{x}_p + \mathbf{B}_p u_2) \quad (5.2.8.b)$$

sur la surface de glissement :

$$\dot{S}_1 = 0 \Rightarrow u_{1eq} = -\frac{1}{\mathbf{K}_s^T \mathbf{B}_s} (\mathbf{K}_s^T \mathbf{A}_s \mathbf{x}_s + \mathbf{K}_s^T \mathbf{D}_s v_1) \quad (5.2.9.a)$$

$$\dot{S}_2 = 0 \Rightarrow u_{2eq} = -\frac{1}{\mathbf{K}_p^T \mathbf{B}_p} (\mathbf{K}_p^T \mathbf{A}_p \mathbf{x}_p + \mathbf{K}_p^T \mathbf{D}_p v_2) \quad (5.2.9.b)$$

on remplace u_{1eq} et u_{2eq} dans les équations (5.2.2) et (5.2.3), on obtient le système équivalent sur la surface de glissement :

$$\dot{\mathbf{x}}_s = \left(\mathbf{A}_s - \frac{\mathbf{B}_s \mathbf{K}_s^T \mathbf{A}_s}{\mathbf{K}_s^T \mathbf{B}_s} \right) \mathbf{x}_s + \left(\mathbf{D}_s - \frac{\mathbf{B}_s \mathbf{K}_s^T \mathbf{D}_s}{\mathbf{K}_s^T \mathbf{B}_s} \right) v_1 = \mathbf{A}_{cs} \mathbf{x}_s + \mathbf{D}_{cs} v_1 \quad (5.2.10.a)$$

$$(5.2.10.b)$$

$$\dot{\mathbf{x}}_p = \left(\mathbf{A}_p - \frac{\mathbf{B}_p \mathbf{K}_p^T \mathbf{A}_p}{\mathbf{K}_p^T \mathbf{B}_p} \right) \mathbf{x}_p + \left(\mathbf{D}_p - \frac{\mathbf{B}_p \mathbf{K}_p^T \mathbf{D}_p}{\mathbf{K}_p^T \mathbf{B}_p} \right) \mathbf{v}_2 = \mathbf{A}_{cp} \mathbf{x}_p + \mathbf{D}_{cp} \mathbf{v}_2$$

On choisit les pôles suivants $-p_1, -p_2, -p_3$ pour le premier sous système et $-p_4, -p_5, -p_6$ pour le second sous système, puis on détermine les coefficients k_i par identification, on aura donc :

$$\det(\mathbf{A}_{cs} - s\mathbf{I}) = s(s + p_1)(s + p_2)(s + p_3) \quad (5.2.11.a)$$

$$\det(\mathbf{A}_{cp} - s\mathbf{I}) = s(s + p_4)(s + p_5)(s + p_6) \quad (5.2.11.a)$$

on pose $k_4 = 1$ et $k_8 = 1$ et $p_1 = 1, p_2 = 2, p_3 = 3, p_4 = 4, p_5 = 5, p_6 = 6$ donc on a :
 $k_1 = 2.86, k_2 = 4.07, k_3 = 5.61, k_5 = 16.9, k_6 = 8.08, k_7 = 7.9$:

$$S_1 = -2.86x_1 - 4.07x_2 - 5.61x_3 - x_4$$

$$S_2 = -16.9x_5 - 8.08x_6 - 7.9x_7 - x_8$$

En cas de variation de la grandeur de perturbation, on peut distinguer deux cas, si la variation est assez brutale, il est possible que $u_{eq} > u_{max}$ (ou $u_{eq} < u_{min}$), le réglage sort du mode glissant. Par contre, si la variation est moins brutale, le réglage ne quitte pas le mode glissant. Donc pour dépasser ce problème majeur, on utilise une fonction $V = \frac{1}{2} S^2$ qui vérifie les trois conditions d'existence de la fonction de Lyapunov qui sont les suivantes :

1. $S = 0, V = 0$.
2. V est quadratique en S .
3. $\dot{V} < 0 \Rightarrow S\dot{S} < 0$, c'est aussi la condition d'existence du mode glissant.

si on choisit en premier lieu :

$$\begin{aligned} \dot{S}_1 &= -k_s \text{sign} S_1 & k_s > 0 \\ \dot{S}_2 &= -k_p \text{sign} S_2 & k_p > 0 \end{aligned} \quad (5.2.12)$$

donc :

$$S_1 \dot{S}_1 < 0 \Rightarrow S_1 (-1.2x_2 + 2.4x_3 + 1.1x_4 - 4.1v_1 - 11.12u_1) = -k_s S_1 \text{sign} S_1$$

$$S_2 \dot{S}_2 < 0 \Rightarrow S_2 (-0.07x_6 + 41x_7 + 8.1v_2 - 0.1x_8 + 16u_2) = -k_p S_2 \text{sign} S_2$$

alors :

$$\begin{aligned} u_1 &= \frac{1}{11.12} (-1.2x_2 + 2.4x_3 + 1.1x_4 + k_s \text{sign} S_1) \\ u_2 &= \frac{1}{16} (0.07x_6 - 41x_7 + 0.1x_8 - k_p \text{sign} S_2) \end{aligned} \quad (5.2.13)$$

mais :

$$S_1 \dot{S}_1 < 0 \Rightarrow S_1 (-k_s \text{sign} S_1 - 4.1v_1)$$

$$S_2 \dot{S}_2 < 0 \Rightarrow S_2 (-k_p \text{sign} S_2 + 8.1v_2)$$

donc il suffit de prendre :

$$k_s > 4.1a$$

$$k_p > 8.1b$$

On peut changer la fonction d'attractivité :

$$\dot{S}_1 = -k_s S_1$$

$$\dot{S}_2 = -k_p S_2$$

pour éliminer le phénomène de « *chattering* », on utilise la fonction *sat* au lieu de *sign*.

5.2.2 Résultats de simulation

Les figures (5.2.1,2,3,4,5) présentent les simulations de la commande à structure variable appliquée à notre système. Dans la figure 5.2.1, la simulation est faite avec une fonction d'attractivité $-k S$ qui donne une dynamique satisfaisante.

Cependant, dans la figure 5.2.2, la fonction d'attractivité $-k \text{sign} S$ est utilisée, cette fonction a l'inconvénient d'introduire le phénomène de « *chattering* » qui pose le problème d'implémentation, d'où l'utilisation de la fonction *sat*, dont la simulation est représentée dans la figure 5.2.3. En effet le phénomène de « *chattering* » est diminué, mais il reste persistant, ceci revient aux gains d'attractivité utilisés qui sont très élevés.

Enfin, les figures 5.2.4 et 5.2.5, montrent les simulations avec une perturbation paramétrique de 25 %, on constate que les dynamiques ne sont pas beaucoup affectées.

Simulation avec la commande par mode glissant

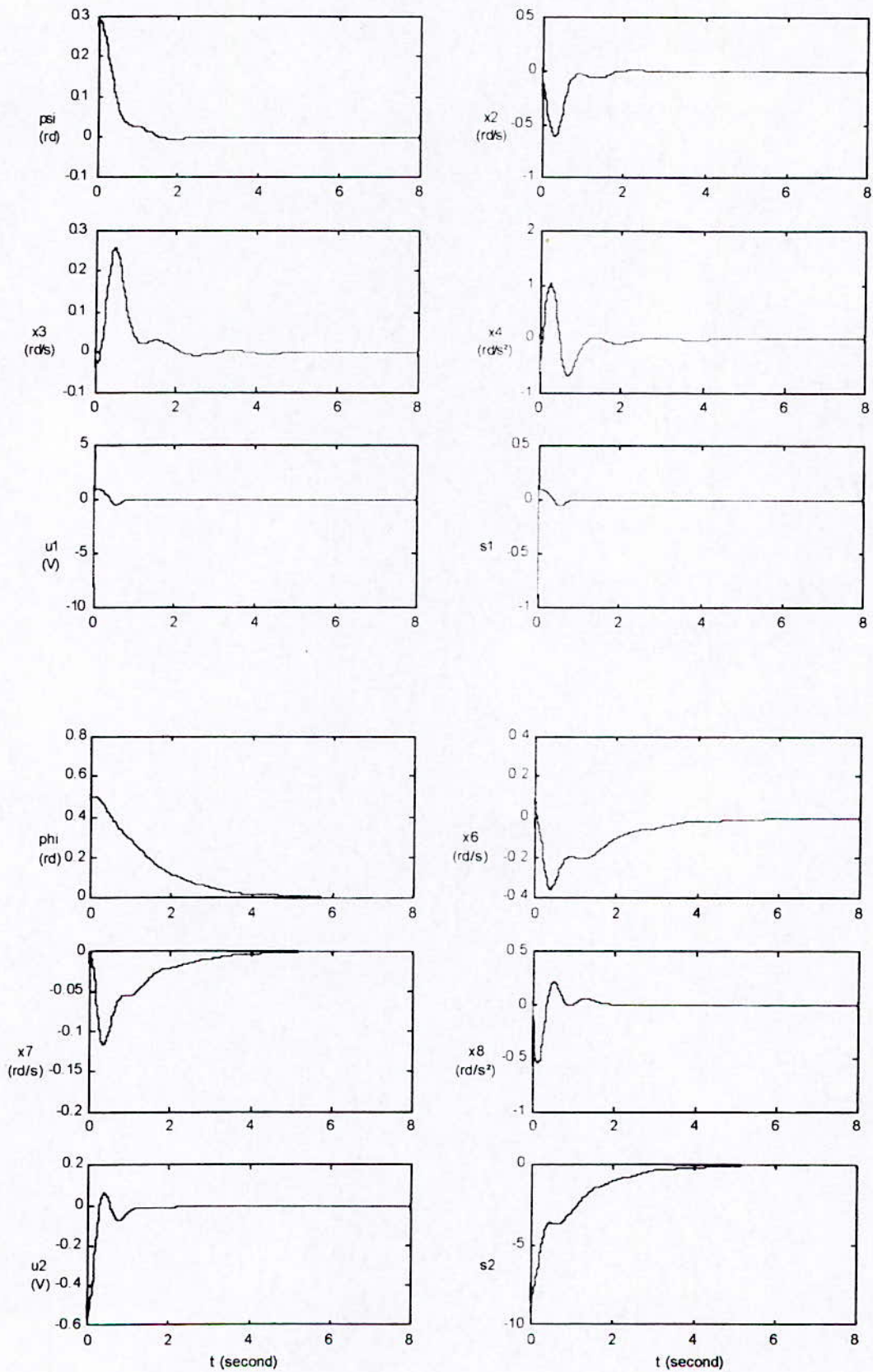


Figure 5.2.1 : Simulation avec une fonction d'attractivité : $-kS$

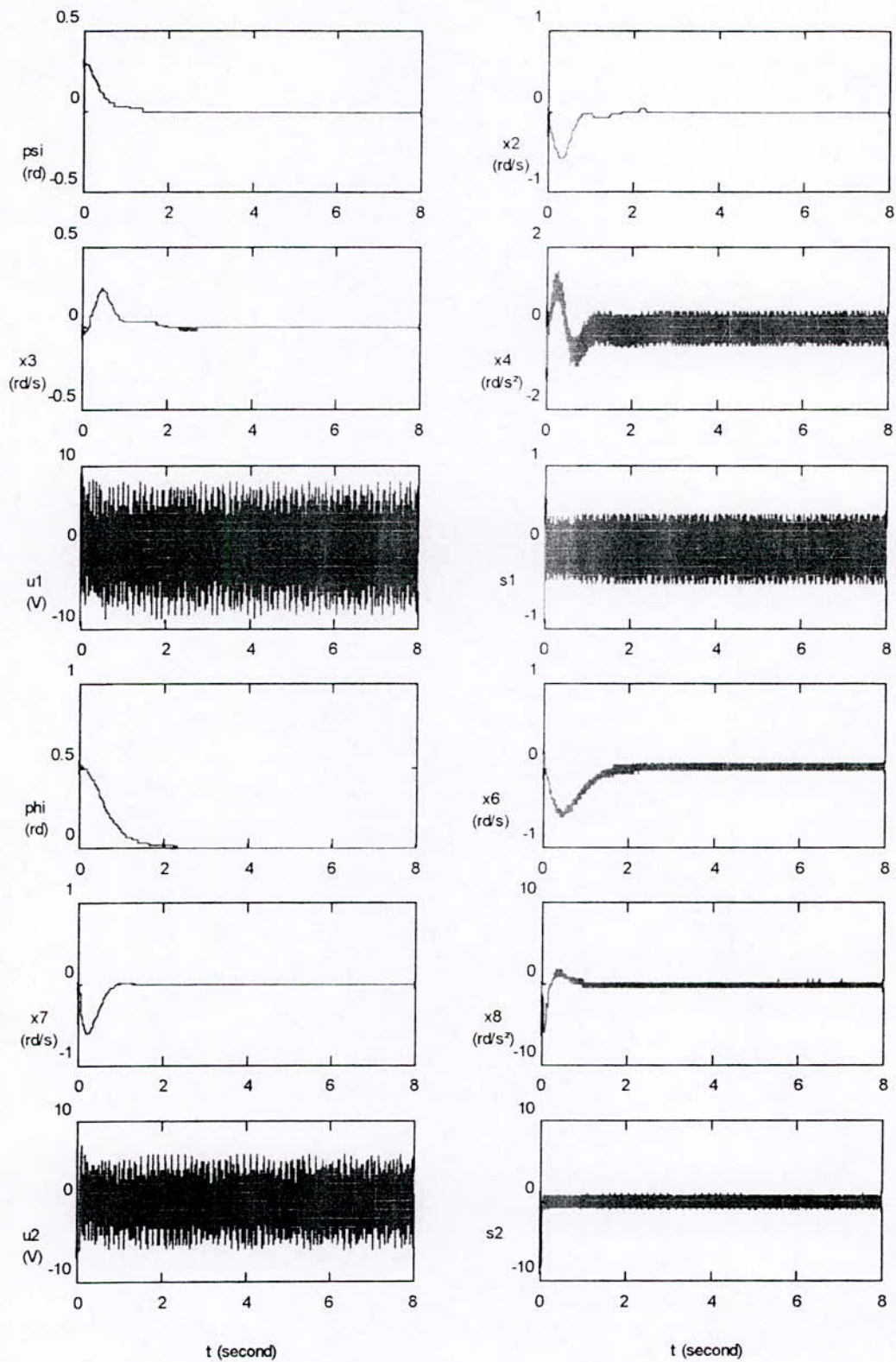


Figure 5.2.2 : Simulation avec fonction d'attractivité : $-k.signS$

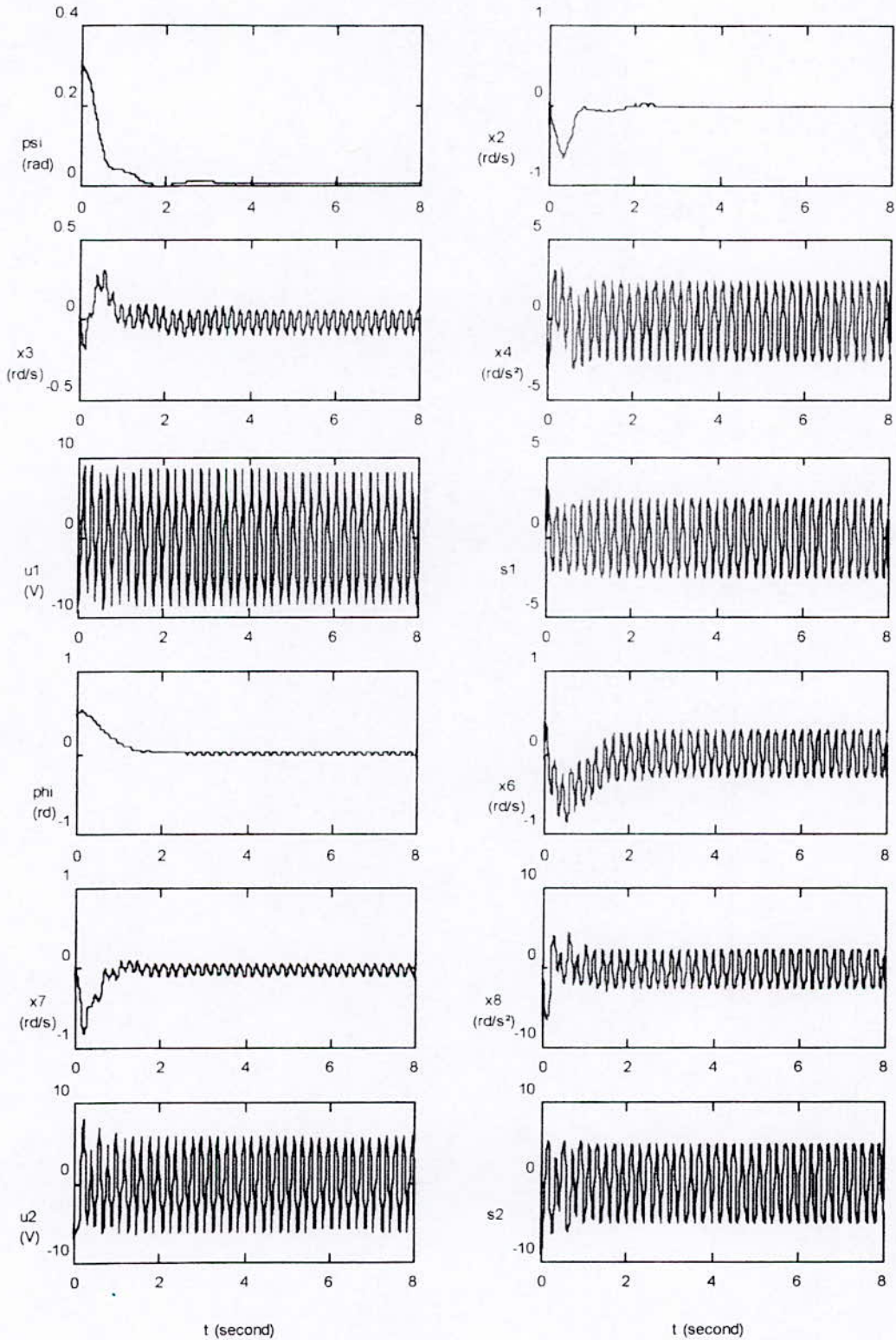


Figure 5.2.3 : Simulation avec fonction d'attractivité : $-k.satS$ avec $\epsilon=0.05$

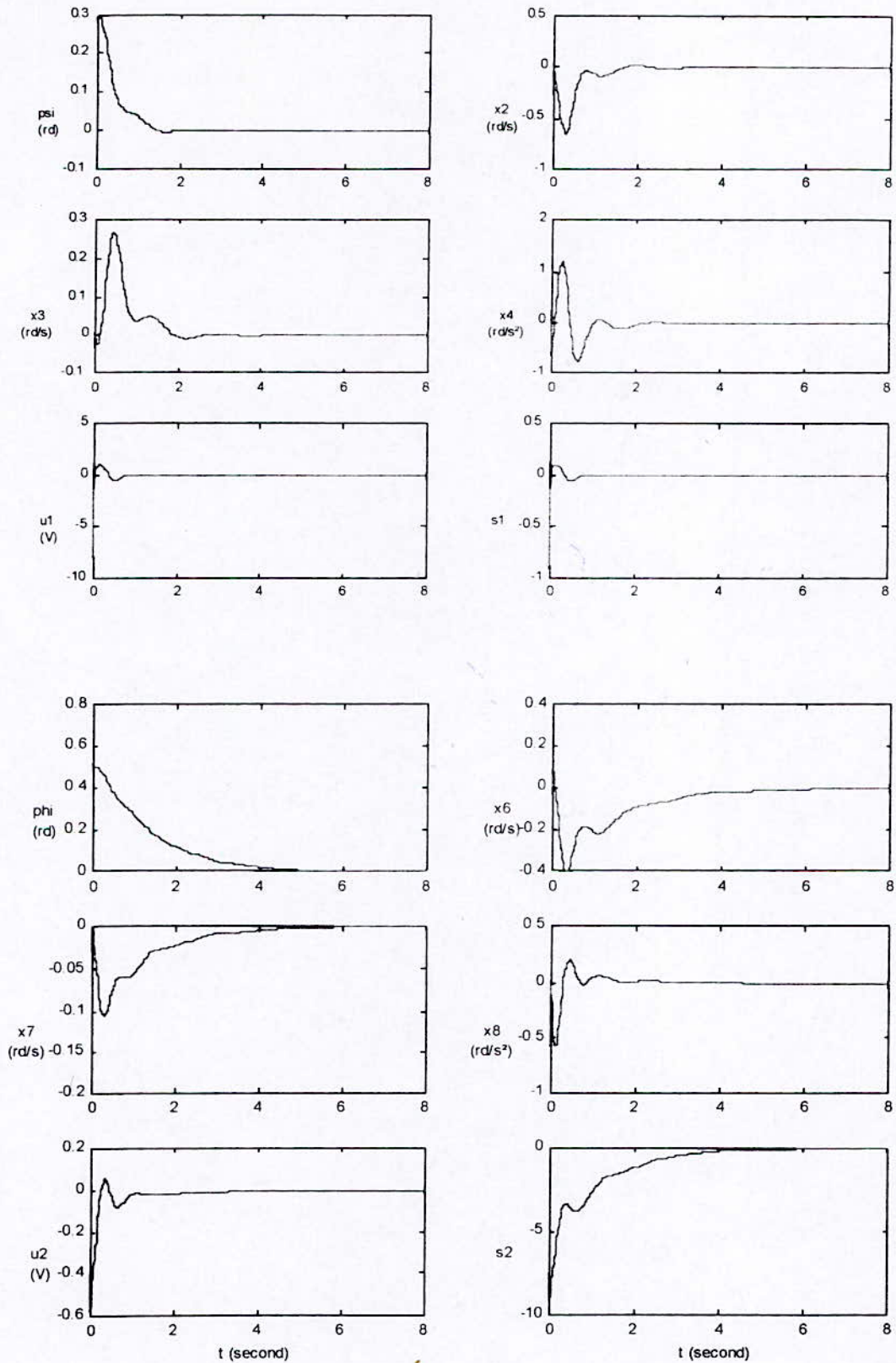


Figure 5.2.4 : Simulation avec perturbation paramétrique de 25% et une fonction d'attractivité : $-kS$

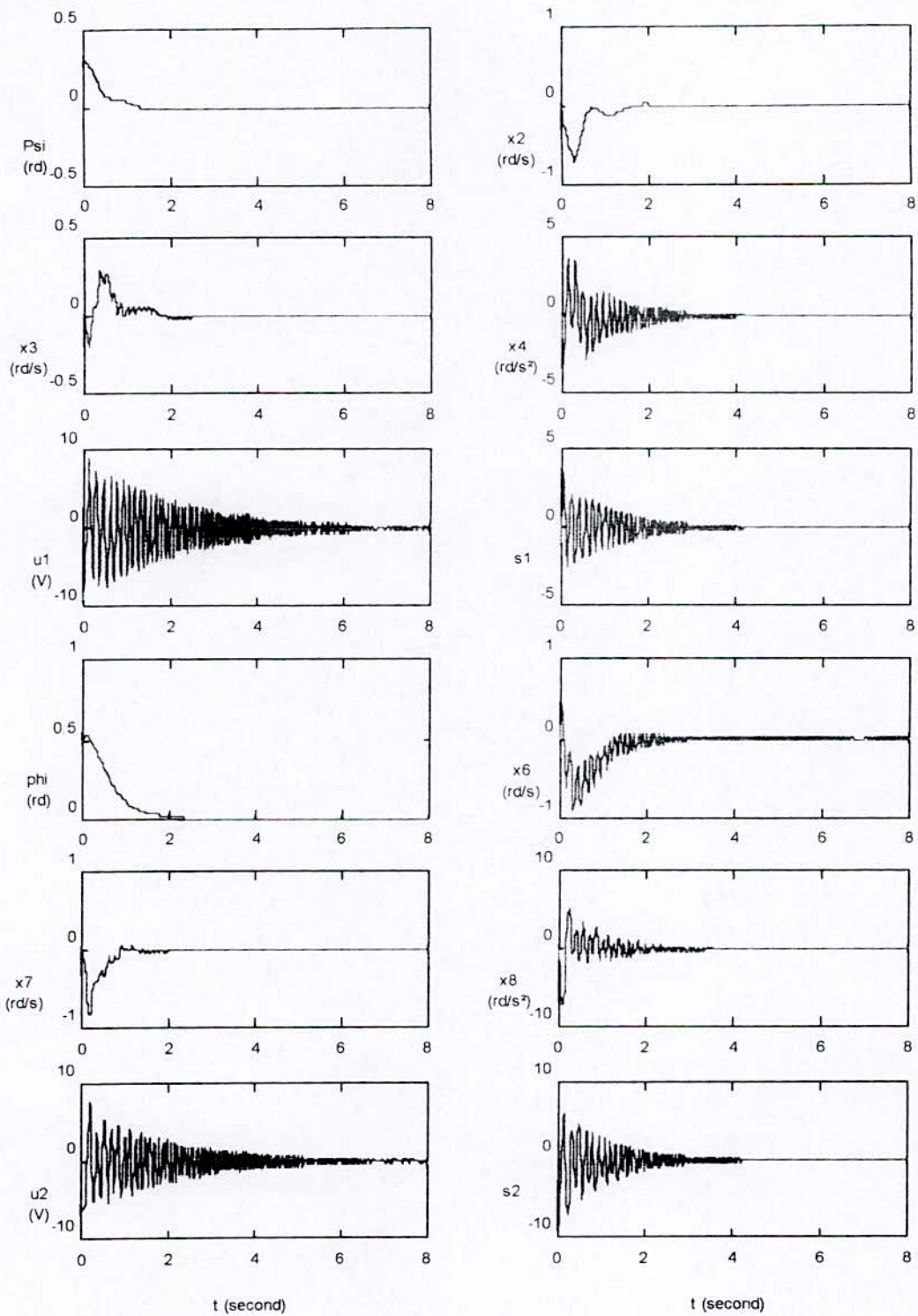


Figure 5.2.5 : Simulation avec perturbation paramétrique de 25% et une fonction d'attractivité : $-ksatS$

5.3 Commande par la méthode du « Backstepping »

Cette méthode a été présentée dans le chapitre 2, elle a l'avantage d'avoir la possibilité de commander les systèmes avec perturbations en utilisant des majorations dans la fonction de Lyapunov. Mais l'application directe de cette méthode à notre modèle, nous conduit aux mêmes problèmes rencontrés dans la méthode de linéarisation par retour d'état car les deux approches sont basées sur la fonction de Lyapunov. On prend donc le modèle (5.2) où v contient toutes les non linéarités et les couplages.

5.3.1 Application

Le modèle choisi (5.2.1) peut être considéré comme deux sous systèmes s'écrivant sous la même forme. Appliquons donc la méthode de « Backstepping » sur le premier sous système.

Etape 1 : Supposons que notre objectif est de faire la sortie x_1 suivre une trajectoire x_{1d} , on prend donc : $z_1 = x_1 - x_{1d}$, où z_1 représente l'erreur de poursuite. Pour stabiliser z_1 choisissons comme commande virtuelle l'état x_2 . Avec une fonction candidate donnée $V(z_1) = \frac{1}{2} z_1^2$, pour obtenir une dérivée définie négative de cette fonction, la commande virtuelle doit être choisie comme suit :

$$x_2 = -k_1 z_1 \quad (5.3.1)$$

Dans ce cas $\dot{V}(z_1) = -k_1 z_1^2$, avec $k_1 > 0$. Donc $x_1 \rightarrow x_{1d}$.

Pour l'étape suivante, les variables à stabiliser sont z_1 et $z_2 = x_2 - k_1 z_1$.

Etape 2 : La fonction choisie est $V(z_1, z_2) = \frac{1}{2}(z_1^2 + z_2^2)$, sa dérivée en fonction du temps est :

$$\dot{V}(z_1, z_2) = z_1 \dot{z}_1 + z_2 \dot{z}_2 \quad \Rightarrow \quad \dot{V} = -k_1 z_1^2 + z_2 (z_1 + \dot{z}_2) \quad (5.3.2)$$

Donc :

$$\dot{V} = -k_1 z_1^2 + z_2 (z_1 + (k_1 - \alpha_1)x_2 + \alpha_4 x_3) + z_2 v_1 \quad (5.3.3)$$

On voit que la fonction \dot{V} comporte le terme v_1 qu'on a supposé comme perturbation. Elle doit être donc majorée par :

$$\dot{V} \leq -k_1 z_1^2 + z_2 [z_1 + (k_1 - \alpha_1)x_2 + \alpha_4 x_3 + v_{1\max} \text{sign}(z_2)]$$

où $v_{1\max}$ vérifie : $|v_1| < v_{1\max}$

on pose :

$$z_1 + (k_1 - \alpha_1)x_2 + \alpha_4 x_3 + v_{1\max} \text{sign}(z_2) = -k_2 z_2 \quad (5.3.4)$$

La dérivée de cette fonction est :

$$\dot{V}(z_1, z_2) = -(k_1 z_1^2 + k_2 z_2^2) \quad (5.3.5)$$

Les variables à stabiliser dans l'étape suivante sont z_1, z_2 et z_3 avec :

$$z_3 = z_1 + (k_1 - \alpha_1)x_2 + \alpha_4 x_3 + v_{1\max} \text{sign}(z_2) + k_2 z_2 \quad (5.3.6)$$

Etape 3 : La fonction choisie est $V(z_1, z_2, z_3) = \frac{1}{2}(z_1^2 + z_2^2 + z_3^2)$, en suivant la même procédure on aboutit à :

$$z_4 = k_3 z_3 + z_2 + \theta_1 x_2 + \theta_2 x_3 + \alpha_4 x_4 + v_{1\max} \theta_3 \text{sign}(z_3) \quad (5.3.7)$$

avec :

$$\begin{aligned} \theta_1 &= 1 - \alpha_1 k_1 + k_1 k_2 - \alpha_1 k_2 + \alpha_1^2 \\ \theta_2 &= \alpha_4 k_2 + k_1 \alpha_4 - \alpha_1 \alpha_4 \\ \theta_3 &= k_2 + k_1 - \alpha_1 \end{aligned}$$

Etape 4 : De même, $V(z_1, z_2, z_3, z_4) = \frac{1}{2}(z_1^2 + z_2^2 + z_3^2 + z_4^2)$, en dérivant cette fonction, la commande apparaît :

$$u_1 = [(-k_4 z_4 - z_3 - \gamma_1 x_2 - \gamma_2 x_3 - \gamma_3 x_4 + v_{1\max} \gamma_4 \text{sign}(z_4))] / \alpha_4 \alpha_6 \quad (5.3.8)$$

avec :

$$\begin{aligned} \gamma_1 &= k_3 \theta_1 - \alpha_1 + k_1 - \theta_1 \alpha_1 \\ \gamma_2 &= k_3 \theta_2 + \alpha_4 + \theta_1 \alpha_4 - \alpha_4 \alpha_6 \\ \gamma_3 &= k_3 \alpha_4 + \theta_2 - \alpha_4 \alpha_5 \\ \gamma_4 &= k_3 \theta_3 + 1 + \theta_1 \end{aligned}$$

Pour le deuxième sous système, les mêmes étapes sont suivies et on obtient :

$$\begin{aligned} z'_1 &= x_5 - x_{5d} \\ z'_2 &= x_6 - k'_1 z'_1 \\ z'_3 &= z'_1 + (k'_1 - \alpha_9)x_6 + \alpha_8 x_7 + v_{2\max} \text{sign}(z'_2) + k'_2 z' \\ z'_4 &= k'_3 z'_3 + z'_2 + \theta'_1 x_6 + \theta'_2 x_7 + \alpha_8 x_8 + v_{2\max} \theta'_3 \text{sign}(z'_3) \end{aligned}$$

où $v_{2\max}$ vérifie : $|v_2| < v_{2\max}$

avec :

$$\begin{aligned} \theta'_1 &= 1 - \alpha_9 k'_1 + k'_1 k'_2 - \alpha_9 k'_2 + \alpha_9^2 \\ \theta'_2 &= \alpha_8 k'_2 + k'_1 \alpha_8 - \alpha_9 \alpha_8 \\ \theta'_3 &= k'_2 + k'_1 - \alpha_9 \end{aligned}$$

enfin :

$$u_2 = \left[(-k_4' z_4' - z_3' - \gamma_1' x_2 - \gamma_2' x_7 - \gamma_3' x_8 + v_{2\max} \gamma_4' \text{sign}(z_4')) \right] / \alpha_8 \alpha_{12} \quad (5.3.9)$$

et :

$$\begin{aligned} \gamma_1' &= k_3' \theta_1' - \alpha_9 + k_1' - \theta_1' \alpha_9 \\ \gamma_2' &= k_3' \theta_2' + \alpha_8 + \theta_1' \alpha_8 - \alpha_8 \alpha_{12} \\ \gamma_3' &= k_3' \alpha_8 + \theta_2' - \alpha_8 \alpha_{11} \\ \gamma_4' &= k_3' \theta_3' + 1 + \theta_1' \end{aligned}$$

5.3.2 Résultats de simulation

Les Figures (5.3.1,2,3) représentent les simulations du système commandé avec la méthode du « Backstepping ». Dans la figure 5.3.1, on utilise la fonction *sign*. La dynamique des sorties est bonne, mais on observe le phénomène de « chattering » dans certains états et surtout dans les commandes u_1 et u_2 .

Pour remédier à ce problème, on introduit la fonction *sat* dont la simulation est donnée dans la figure 5.3.2 . On voit que le phénomène de « chattering » diminue mais il ne s'élimine pas.

Afin de tester la robustesse de cette commande, on fait la simulation (5.3.3) pour une perturbation paramétrique de 25 %. En effet, on voit que la commande synthétisée rejette cette perturbation, et le système se stabilise.

Conclusion

Dans cette première partie, on a introduit les commandes quantitatives, ces commandes exigent la connaissance de tous les états du système. Par la suite nous allons mettre en application une commande qualitative, qui ne demande qu'une partie des états du système.

Simulation du modèle avec commande par Backstepping

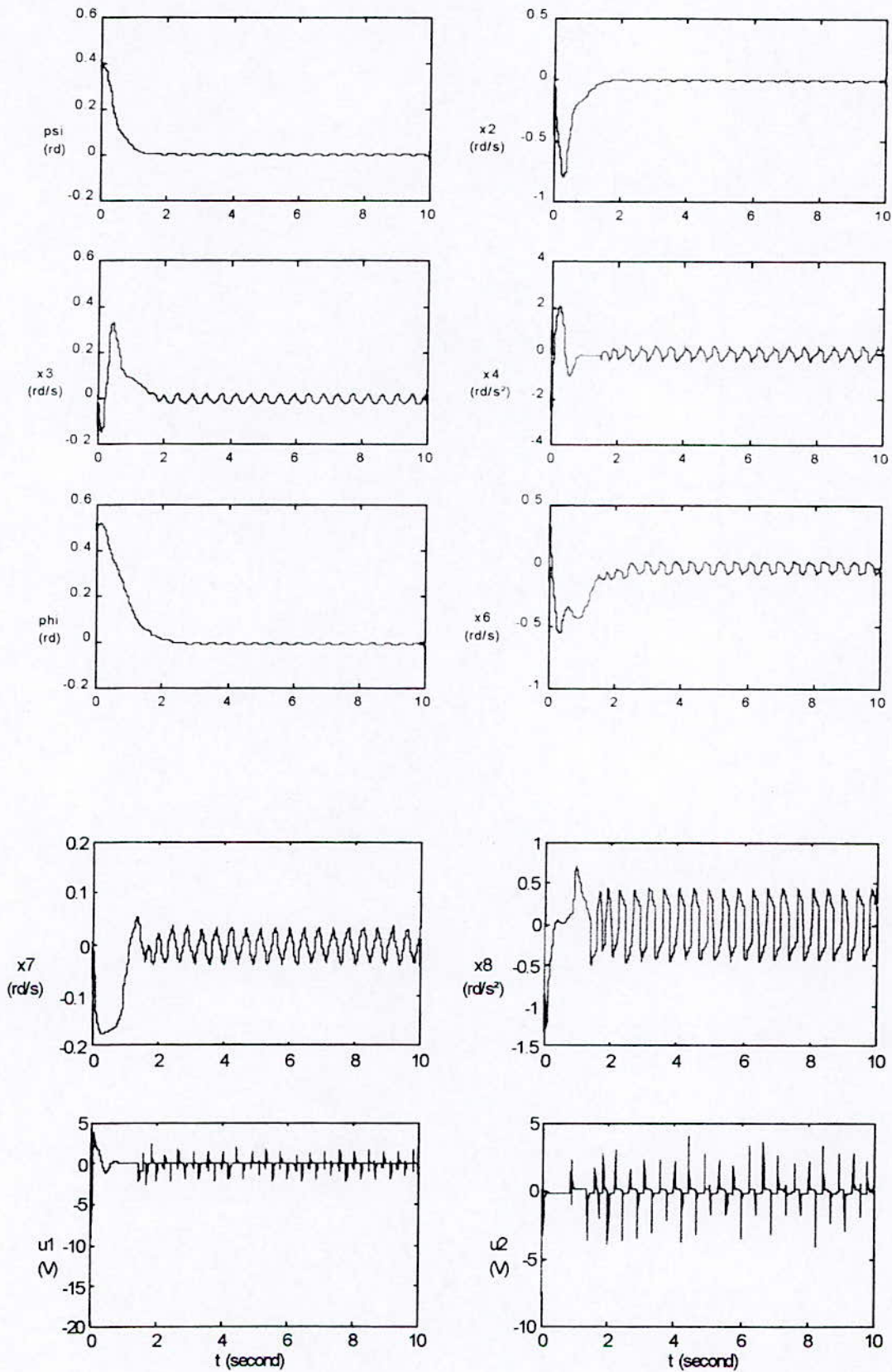


Figure 5.3.1 : Simulation avec la fonction *sign*

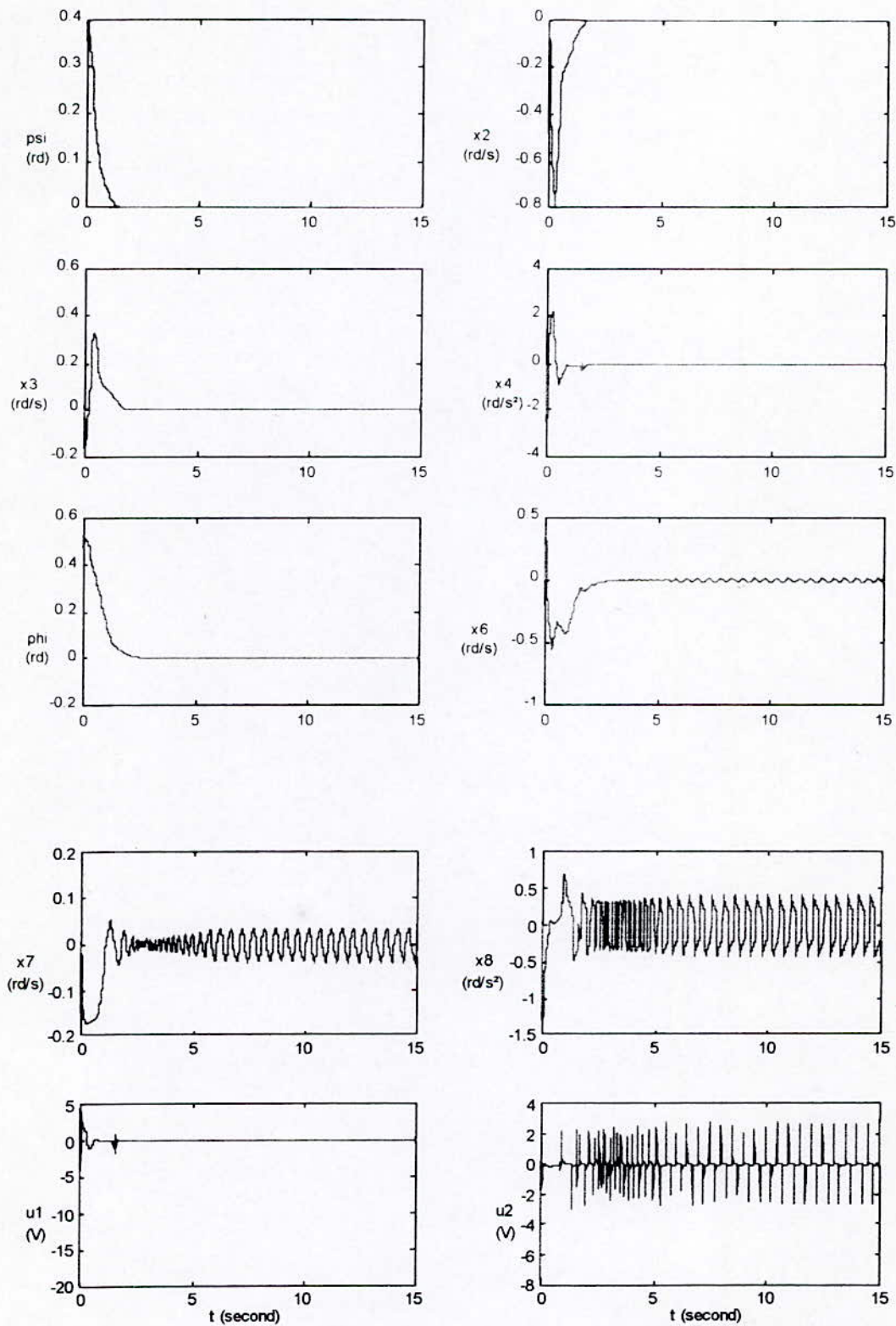


Figure 5.3.2 : Simulation avec la fonction *Sat* $\mathcal{E}=0.1$

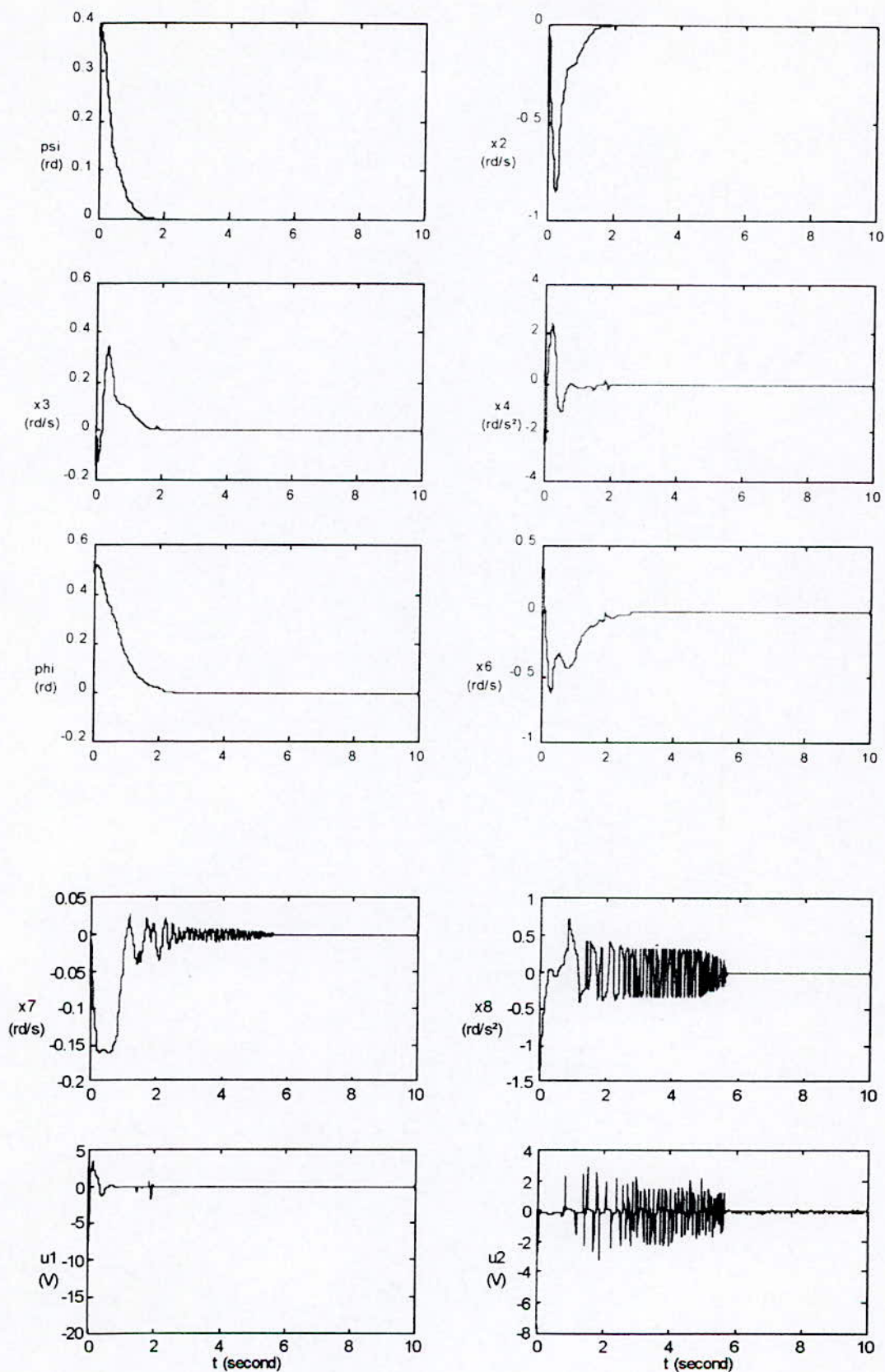


Figure 5.3.3 : Simulation avec la fonction Sat $\mathcal{E}=0.1$ et une perturbation paramétrique de 25%

5.4 Commande par logique floue

Après avoir appliqué différentes commandes quantitatives (Linéarisation par retour d'état, mode glissant et « Backstepping »), on essaye dans cette étape de synthétiser une commande qualitative qui est la logique floue. Cette commande a l'avantage de ne pas avoir besoin du modèle mathématique exacte du système, ni de tous ses états. Elle se caractérise aussi par la simplicité de synthèse de la loi de commande. La théorie de la logique floue est détaillée dans le chapitre 3.

Sur la figure (5.4.1) on présente le schéma de nos deux régulateurs, qui sont indépendants l'un de l'autre :

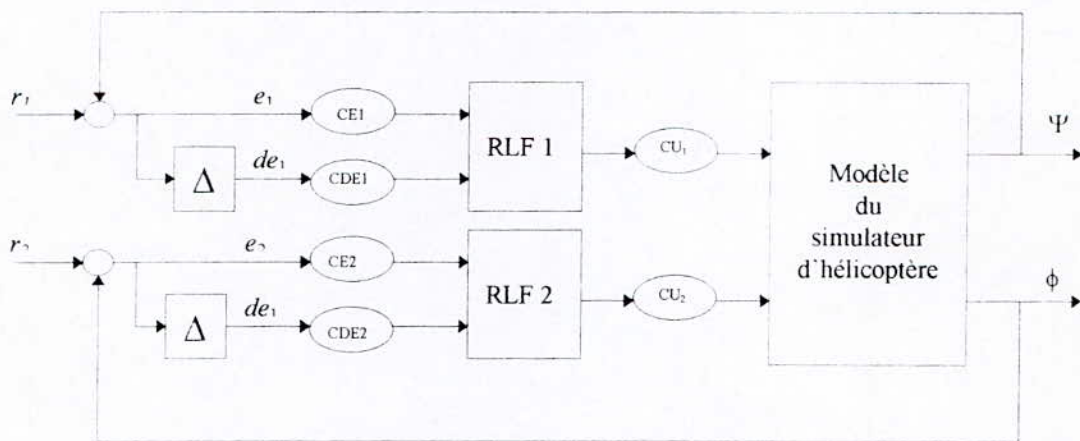


Figure 5.4.1 :Schéma global du système réglé

5.4.1 Choix des variables d'entrée et de sortie

Le choix des variables linguistiques et leurs fonctions d'appartenance est très important, en effet c'est de ce choix que vont dépendre les performances de la commande, dans notre application on a choisi comme variable linguistique l'erreur entre la consigne et la grandeur à régler, et la variation de l'erreur. Ce choix peut être justifié par une analogie au régulateur PID, et aussi par l'inaccessibilité à tous les états du système.

5.4.2 Normalisation

La normalisation consiste à ramener notre centre d'intérêt à un intervalle déterminé entre $[-1, 1]$, pour avoir une flexibilité dans l'implémentation du régulateur, donc il faut avoir des gains d'adaptation pour avoir la dynamique voulue, mais il n'y a pas une technique systématique pour la détermination des gains d'adaptation, donc on procède par tâtonnement.

5.4.3 Régulateur du type Mamdani

Le régulateur du type Mamdani est composé des éléments suivants :

- ☞ un bloc de calcul de l'erreur (e) et de la variation de l'erreur (de).
- ☞ des gains d'adaptation dans l'implémentation du régulateur.
- ☞ un bloc de fuzzification de l'erreur et de sa variation.
- ☞ une base de règles pour le régulateur flou, tiré de la matrice de Macvicar Whelan.
- ☞ une logique floue utilisée pour évaluation des règles de commande floue (inférence), dans notre cas c'est min-max qui est utilisée.
- ☞ un bloc de déffuzzification servant à convertir la commande floue en une valeur numérique, dans notre cas on a utilisé M.C.G.

5.4.4 Matrice de Macvicar Whelan

Une simplification de la description des inférences s'obtient à l'aide d'une représentation graphique appelée matrice d'inférence, à l'intersection d'une colonne et d'une ligne se trouve l'ensemble correspondant de la variable de sortie définie pour une règle d'inférence.

La logique avec laquelle a été obtenue la matrice de Macvicar est proche du raisonnement humain, cette matrice est basée sur les principes suivants :

- ☞ Si la sortie est égale à la valeur désiré et la variation de l'erreur est nulle, la commande sera maintenue constante.
- ☞ Si la sortie diverge de sa valeur désirée, l'action sera dépendante du signe et de la valeur de l'erreur et de sa variation, si l'erreur peut être corrigé rapidement par elle même la commande sera maintenue, sinon la commande sera changée jusqu'à atteindre les résultats satisfaisants.

5.4.5 Synthèse des régulateurs

La base des règles est obtenue à partir de la matrice de Macvicar-whelan, les régulateurs pour les deux angles (élévation, azimutale) se différencient seulement par les gains d'adaptations, les fonctions d'appartenance sont triangulaires pour les entrées (erreur, variation de l'erreur) et pour la sortie (commande).

5.4.5.1 Régulateur à trois classes

Les figures (5.4.2) et (5.4.3) représentent la base des règles et la fonction d'appartenance du régulateur à trois niveaux.

e \ de	N	Z	P
N	N	N	Z
Z	N	Z	P
P	Z	P	P

Figure 5.4.2 : Base règles, RLF(3×3)

(N: négatif, Z: zéro, P: positif)

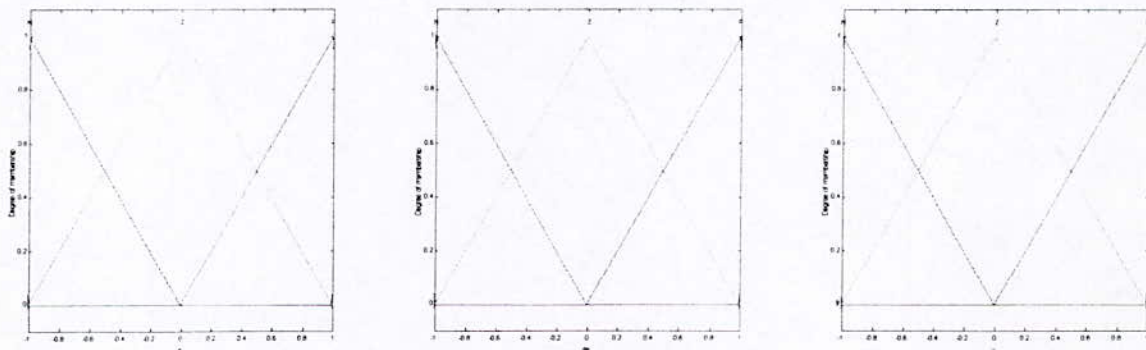


Figure 5.4.3 : Fonctions d'appartenance

5.4.5.2 Régulateur flou à cinq classes

Les figures (5.4.4) et (5.4.5) représentent la base des règles et la fonction d'appartenance du régulateur à trois niveaux.

e \ de	NB	NS	ZE	PS	PB
NB	NB	NB	NB	NS	ZE
NS	NB	NB	NS	ZE	PB
ZE	NB	NS	ZE	PS	PB
PS	NS	ZE	PS	PB	PB
PB	ZE	PS	PB	PB	PB

Figure 5.4.4 : Base des règles, RLF(5×5)

(NB: négatif grand, NS: négative petit, ZE: zéro, PS: positif grand, PB: positif grand)

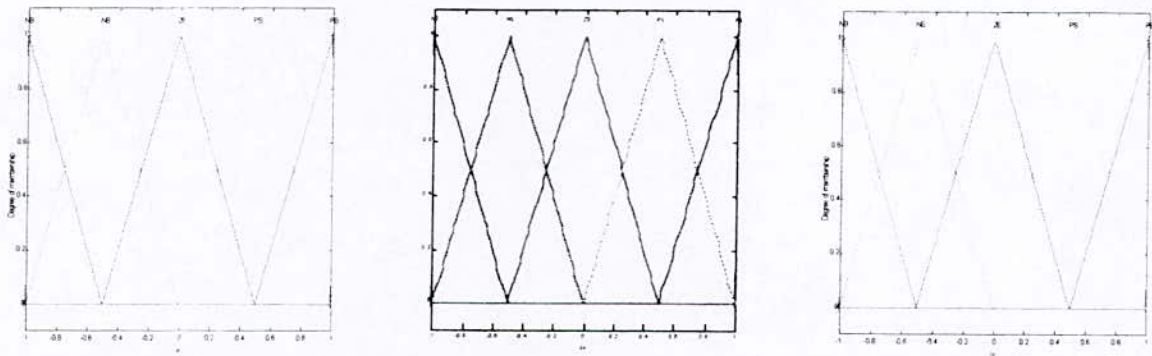


Figure 5.4.5 : Fonctions d'appartenance

5.4.6 Résultats de simulation

Les figures (5.4.6,...,9) concernent le régulateur flou, dans la figure 5.4.6, on donne la simulation pour le régulateur à 3 classes (RLF 3×3). On voit que la dynamique du système est rapide, les états ne sont pas de fréquences très élevées et les commandes n'atteignent pas de grandes valeurs.

La figure 5.4.7 représente la simulation avec le même régulateur (RLF 3×3) mais en introduisant une perturbation paramétrique de 25 %. Avec les mêmes gains précédents, on voit une dégradation de la dynamique du système ; en effet, pour ($\phi = x_5$) on voit une erreur statique de (0.2 rd) et pour ($\psi = x_1$) il y a plus d'oscillations.

Dans la figure 5.4.8, on trouve une simulation pour un (RLF 5×5). On n'est pas arrivé à des gains qui donnent une meilleure dynamique que le régulateur (RLF 3×3), cependant le système se stabilise avec un petit dépassement de 10 %.

Enfin, la figure 5.4.9 représente une simulation avec une perturbation paramétrique de 25 %. Pour ce cas on voit que les sorties se stabilisent mais pas tous les états.

En conclusion, on peut dire que la logique floue est le meilleur choix lorsqu'on trouve les gains de commande appropriés. Pour cela, le seul moyen disponible est le tâtonnement, ceci représente l'inconvénient majeur de la commande par logique floue.

Simulation du modèle avec commande par logique floue

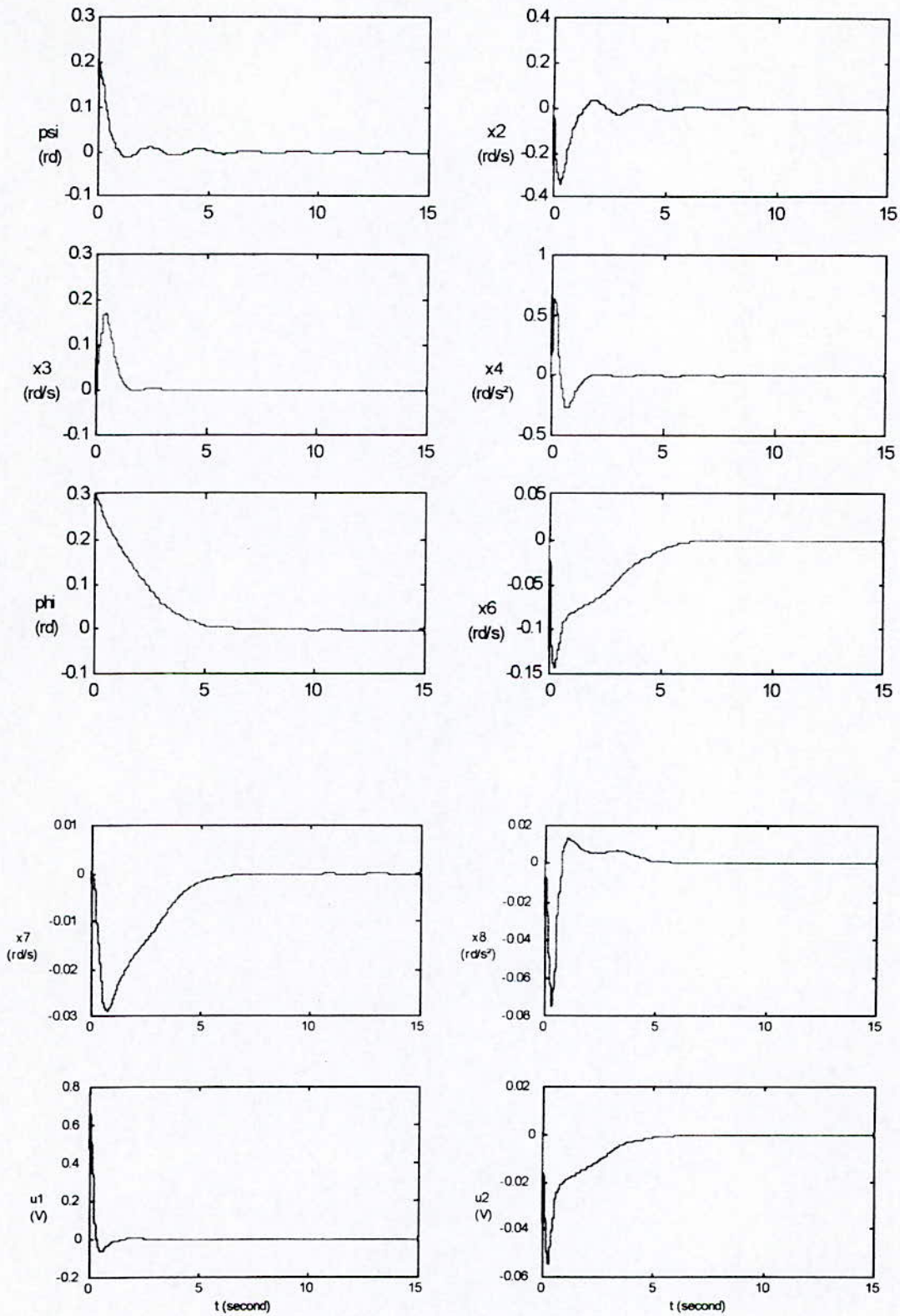


Figure 5.4.6 : Simulation avec un RLF (3x3)

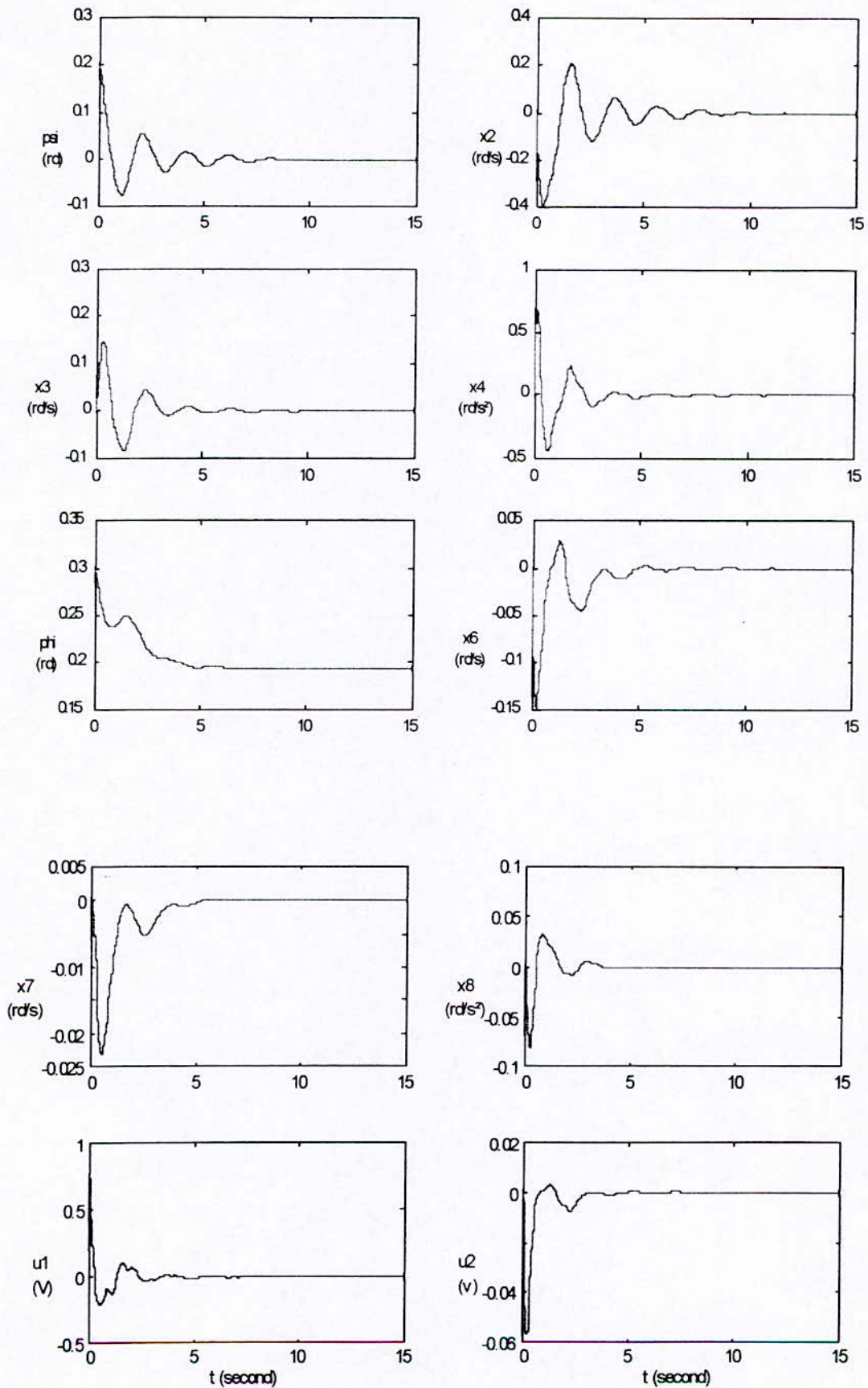


Figure 5.4.7 : Simulation avec un RLF (3×3) et perturbation paramétrique de 25%

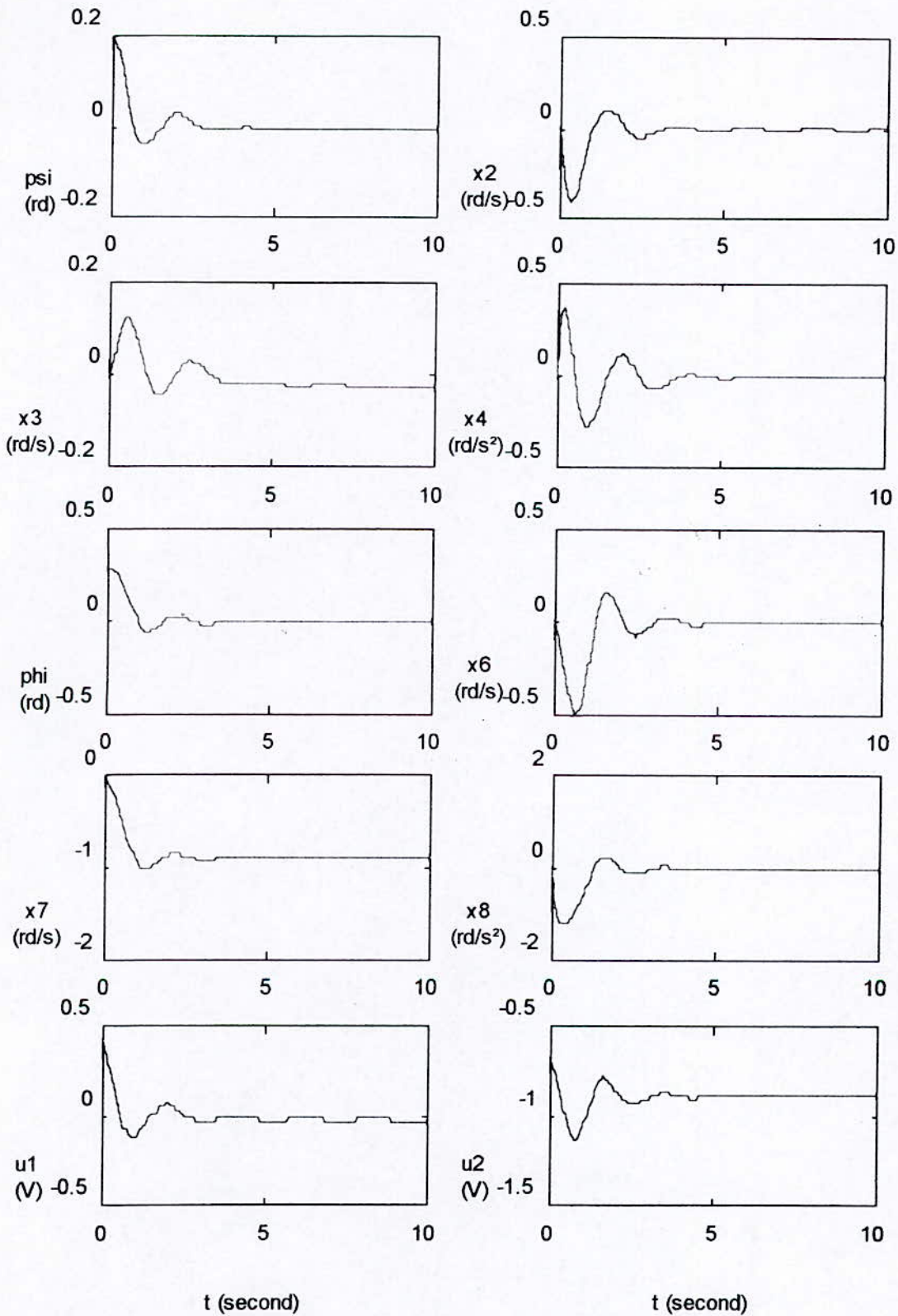


Figure 5.4.8 : Simulation avec un RLF (5×5)

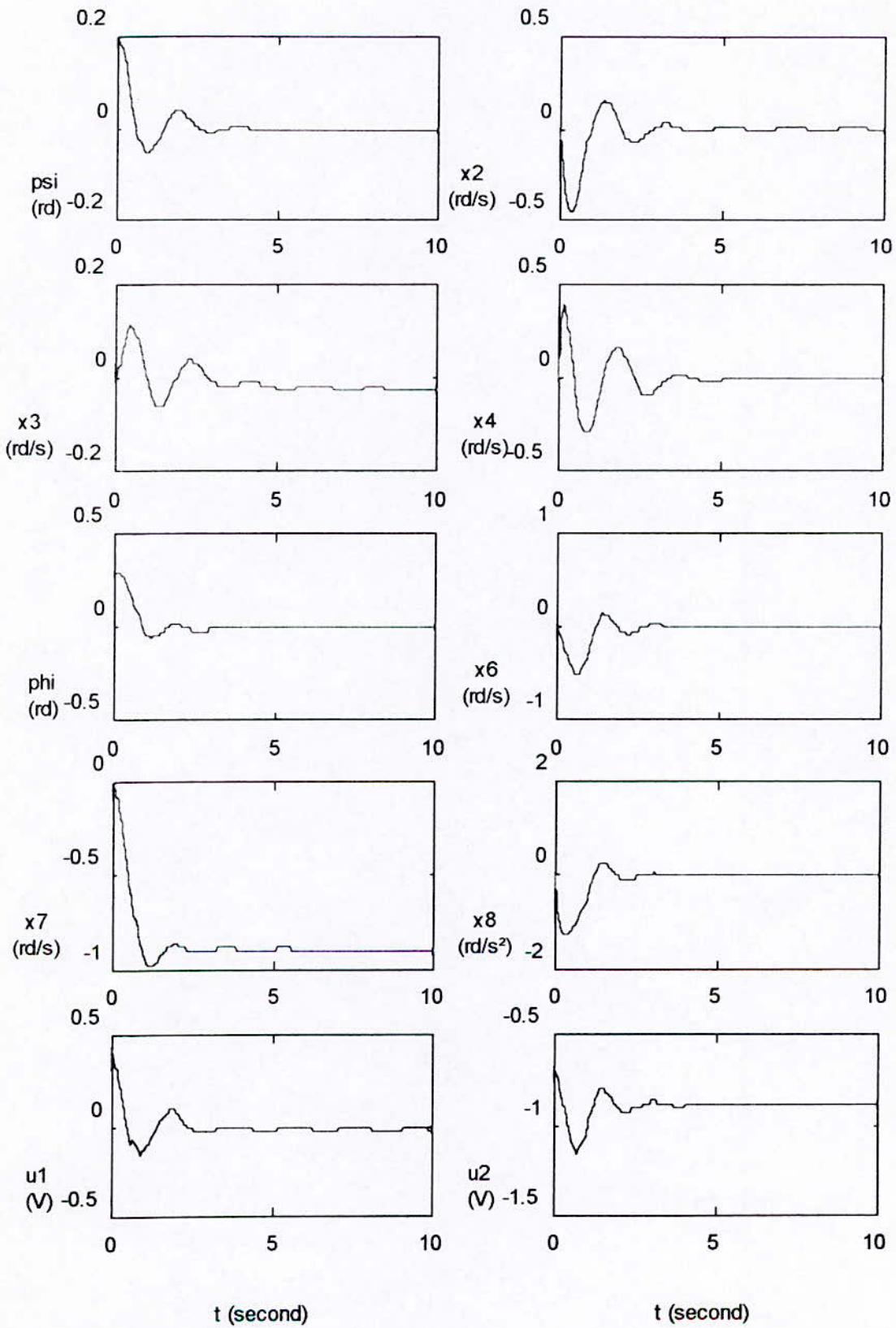


Figure 5.4.9 : Simulation avec un RLF (5×5) et perturbation paramétrique de 25%

B. Commande par retour de sorties non linéaire

Dans cette partie, nous allons synthétiser un certain nombre d'observateurs non linéaires afin de reconstituer les états de notre système d'application ; ces observateurs sont : l'observateur de Thau, l'observateur glissant et l'observateur « Backstepping ». A la fin du chapitre une étude comparative permet de voir les avantages et les inconvénients de chaque observateur.

5.5 Observateur de Thau

Ce type d'observateur est introduit en 1^{er} lieu car il convient à notre système et il est simple à synthétiser. Il a aussi l'avantage d'avoir des termes de correction de type linéaire.

On écrit notre modèle (non linéaire) sous la forme :

$$\dot{x} = Ax + Bu + f(x) \quad (5.5.1.a)$$

$$y = Cx \quad (5.5.1.b)$$

avec :

$$A = \begin{bmatrix} 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & -\alpha_1 & \alpha_4 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & -\alpha_6 & -\alpha_5 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & -\alpha_9 & \alpha_8 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 \\ 0 & 0 & 0 & 0 & 0 & 0 & -\alpha_{12} & -\alpha_{11} \end{bmatrix} ; B = \begin{bmatrix} 0 & 0 \\ 0 & 0 \\ 0 & 0 \\ \alpha_6 & 0 \\ 0 & 0 \\ -\alpha_{10} & 0 \\ 0 & 0 \\ 0 & \alpha_{12} \end{bmatrix}$$

$$C = \begin{bmatrix} 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 \end{bmatrix} ; f(x) = \begin{bmatrix} 0 \\ -\alpha_1 \sin x_1 + \alpha_3 x_3^2 - \beta x_3 x_6 \cos x_1 \\ 0 \\ 0 \\ 0 \\ \alpha_7 x_7^2 \\ 0 \\ 0 \end{bmatrix}$$

En prenant la constante positive de Lipschitz égale à 0.25 pour garantir que la fonction f soit globalement Lipschitzienne.

On calcule la matrice L (gain de correction) telle que les valeurs propres de la matrice A_0 soient :

$$\lambda_1 = -15, \lambda_2 = -1.5, \lambda_3 = -3, \lambda_4 = -1.4, \lambda_5 = -5, \lambda_6 = -6, \lambda_7 = -7, \lambda_8 = -8.$$

et $Q=I(8)$; où $I(8)$ est la matrice identité de dimension 8. avec une condition :

$$\lambda_m = 1 \Rightarrow \lambda_M < \frac{1}{k} \quad \left(\frac{1}{k} = 4 \right)$$

on aura donc :

$$L^T = \begin{bmatrix} 1.63 & 6.86 & -5.06 & 16.19 & 19.02 & 48.57 & -12.07 & 25.65 \\ 10.69 & 16.49 & -7.85 & 28.4 & 2.29 & 12.43 & -2.32 & 2.21 \end{bmatrix}$$

donc on a construit notre observateur :

$$\dot{\hat{x}} = A\hat{x} + Bu + f(\hat{x}) + L(y - C\hat{x}) \quad (5.5.2)$$

5.5.1 Résultats de simulation

Les figure 5.5.1 et 5.5.2 donnent les simulations de l'observateur de Thau pour le simulateur de vol d'hélicoptère. Dans la figure 5.5.1, la simulation est réalisée sans perturbation paramétrique, on voit que le convergence de l'observateur est très rapide (0.3 sec) et elle est exponentielle.

La figure 5.5.2, représente la dynamique de l'observateur avec une perturbation paramétrique de 25 %. On ne voit aucune dégradation des performances.

En conclusion, on peut dire que l'observateur de Thau donne des résultats satisfaisants, mais reste les problèmes de : vérifier la condition de Lipschitz qui n'est pas évidente, décomposer le système sous la forme de Thau et choisir les matrices L et Q . Pour surmonter ces inconvénients, on fait appel à d'autres types d'observateurs.

Simulation avec l'observateur de Thau en boucle ouvert

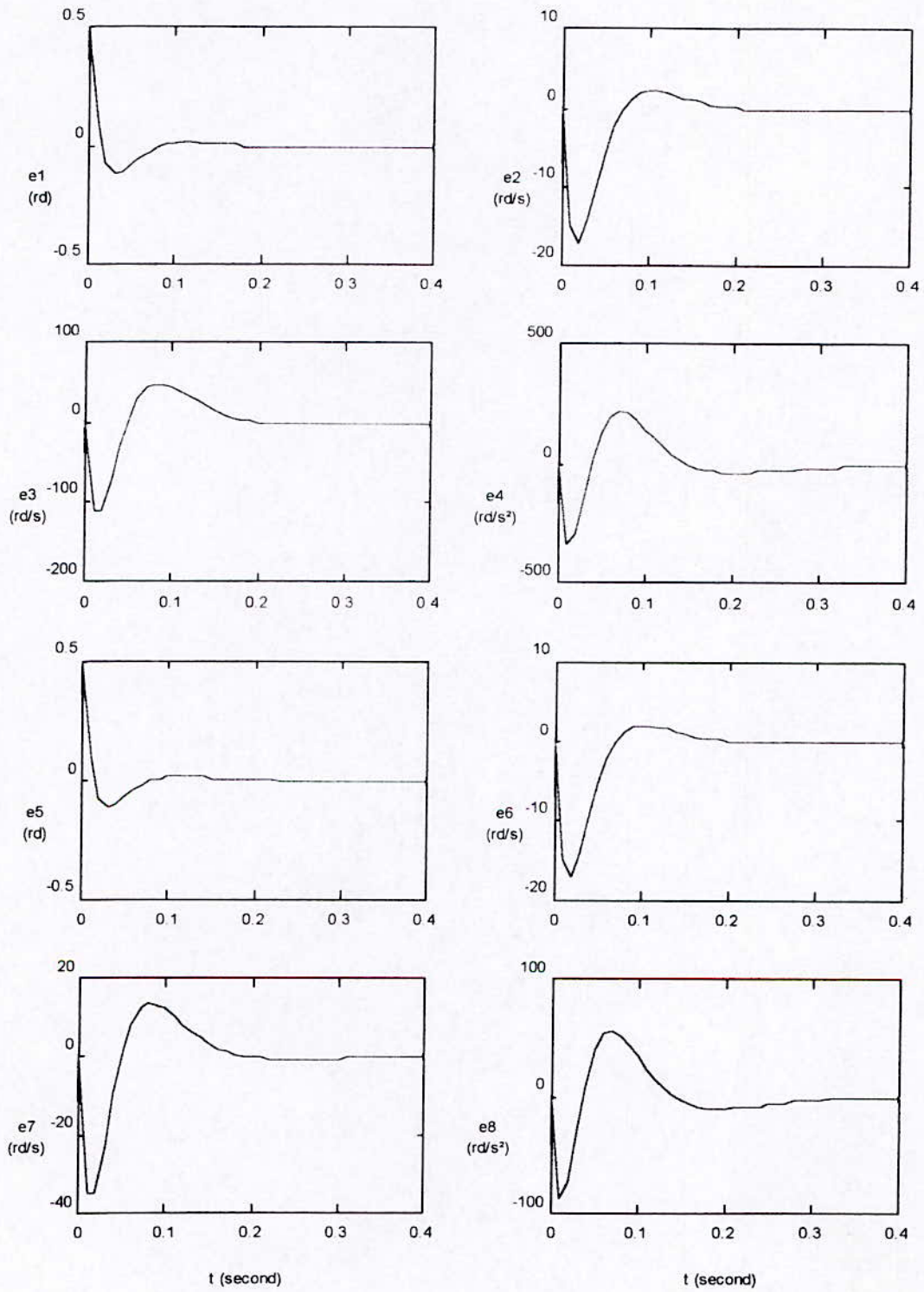


Figure 5.5.1 : Simulation sans perturbation paramétrique

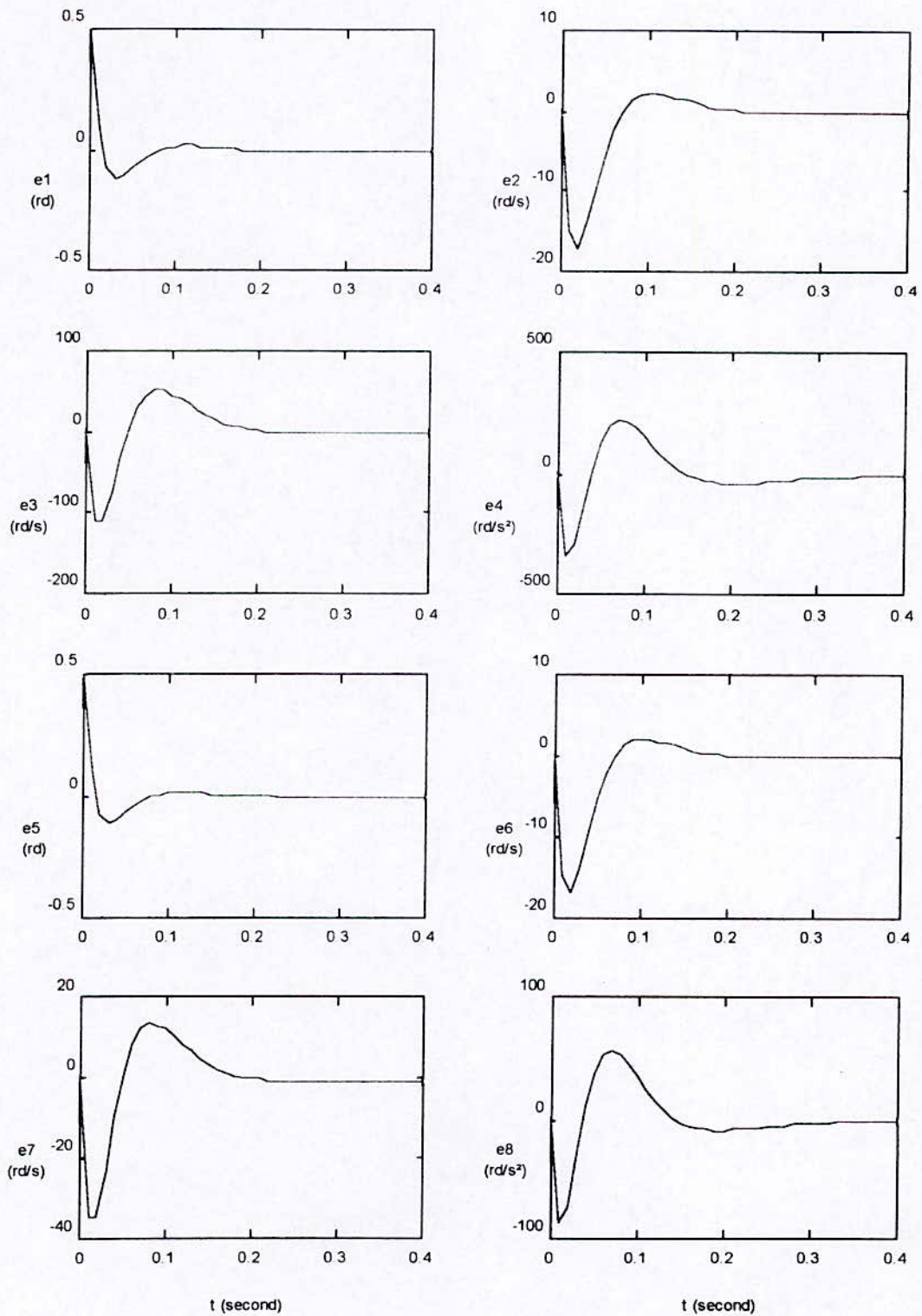


Figure 5.5.2 : Simulation avec perturbation paramétrique de 25%

5.6 Observateur glissant

La théorie du mode glissant a montré son efficacité dans le domaine de la commande, pour cela elle a été utilisée pour la synthèse d'un observateur.

L'observateur glissant les avantages suivants : robustesse, possibilité d'imposition de la dynamique, ceci est important car un observateur doit toujours être plus rapide que la dynamique du système commandé.

On décentralise notre système en deux sous systèmes comme on a fait pour le cas de la commande par mode glissant.

Posons notre observateur de la forme suivante :

$$\dot{\hat{x}}_s = A_s \hat{x}_s + B_s u_1 + I_s \text{signs}_1 \quad (5.6.1.a)$$

$$\dot{\hat{x}}_p = A_p \hat{x}_p + B_p u_2 + I_p \text{signs}_2 \quad (5.6.1.b)$$

avec :

$$I_s = \begin{bmatrix} k_1 \\ k_2 \\ k_3 \\ k_4 \end{bmatrix} ; \quad I_p = \begin{bmatrix} k_5 \\ k_6 \\ k_7 \\ k_8 \end{bmatrix}$$

et :

$$s_1 = e_1 = x_1 - \hat{x}_1 \quad (5.6.2.a)$$

$$s_2 = e_5 = x_5 - \hat{x}_5 \quad (5.6.2.b)$$

la dynamique de l'erreur est donnée par :

$$\dot{e}_s = A_s e_s + D_s v_1 - I_s \text{signs}_1 \quad (5.6.3.a)$$

$$\dot{e}_p = A_p e_p + D_p v_2 - I_p \text{signs}_2 \quad (4.6.3.b)$$

avec :

$$e_s = \begin{bmatrix} x_1 - \hat{x}_1 \\ x_2 - \hat{x}_2 \\ x_3 - \hat{x}_3 \\ x_4 - \hat{x}_4 \end{bmatrix} ; \quad e_p = \begin{bmatrix} x_5 - \hat{x}_5 \\ x_6 - \hat{x}_6 \\ x_7 - \hat{x}_7 \\ x_8 - \hat{x}_8 \end{bmatrix}$$

Nous déterminons les gains k_i par la méthodologie suivante :

1. Pour le premier sous système, on prend une fonction de Lyapunov candidate :

$$V(s_1) = \frac{1}{2} s_1^2 \Rightarrow \dot{V}(s_1) = s_1 \dot{s}_1 = s_1 (e_2 - k_1 \text{sign} s_1) \quad (5.6.4)$$

avec $k_1 > |e_2|$ on garantit la condition d'attractivité. donc quand $s_1 = 0$ et $\dot{s}_1 = 0$: $\text{sign} s_1 = \frac{e_2}{k_1}$. En remplaçant dans les équations de la dynamique de l'erreur pour le premier sous système, on trouve les équations suivantes :

$$\begin{aligned} \dot{e}_2 &= -\left(\alpha_1 + \frac{k_2}{k_1}\right)e_2 + \alpha_4 e_3 + v_1 \\ \dot{e}_3 &= -\frac{k_3}{k_1}e_2 + e_4 \\ \dot{e}_4 &= -\frac{k_4}{k_1}e_2 - \alpha_6 e_3 - \alpha_5 e_4 \end{aligned} \quad (5.6.5)$$

finalement, on fait un placement de pôles pour notre sous système :

$$P(s) = (s+10)^3 \quad k_2 > |v_1| \quad (5.6.6)$$

2. Pour le deuxième sous système, on suit les mêmes étapes. On prend une fonction de Lyapunov candidate :

$$V(s_2) = \frac{1}{2} s_2^2 \Rightarrow \dot{V}(s_2) = s_2 \dot{s}_2 = s_2 (e_6 - k_5 \text{sign} s_2) \quad (5.6.7)$$

avec $k_5 > |e_6|$ on garantit la condition d'attractivité. Donc quand $s_2 = 0$ et $\dot{s}_2 = 0$: $\text{sign} s_2 = \frac{e_6}{k_5}$. En remplaçant dans les équations de la dynamique de l'erreur pour le deuxième sous système, on trouve :

$$\begin{aligned} \dot{e}_6 &= -\left(\alpha_9 + \frac{k_6}{k_5}\right)e_6 + \alpha_8 e_7 + v_2 \\ \dot{e}_7 &= -\frac{k_7}{k_5}e_6 + e_8 \\ \dot{e}_8 &= -\frac{k_8}{k_5}e_6 - \alpha_{12}e_7 - \alpha_{11}e_8 \end{aligned} \quad (5.6.8)$$

les gains trouvés en suivant le même chemin sont :

$$k_1 = 8, \quad k_2 = 169.5, \quad k_3 = 2538.5, \quad k_4 = -2825.4, \quad k_5 = 2, \quad k_6 = 39.8, \quad k_7 = 30.42, \quad k_8 = -60.84$$

la synthèse de l'observateur est donc terminée.

5.6.1 Résultats de simulation

Les figures 5.6.1 et 5.6.2, donnent les graphes de simulations de l'observateur glissant appliqué au simulateur d'hélicoptère. Dans la figure 5.6.2, une perturbation paramétrique de 25% affecte le système. On voit que notre observateur rejette cette perturbation et la dynamique reste pratiquement inchangée. Les résultats sont donc satisfaisants.

Simulation avec observateur glissant en boucle ouverte

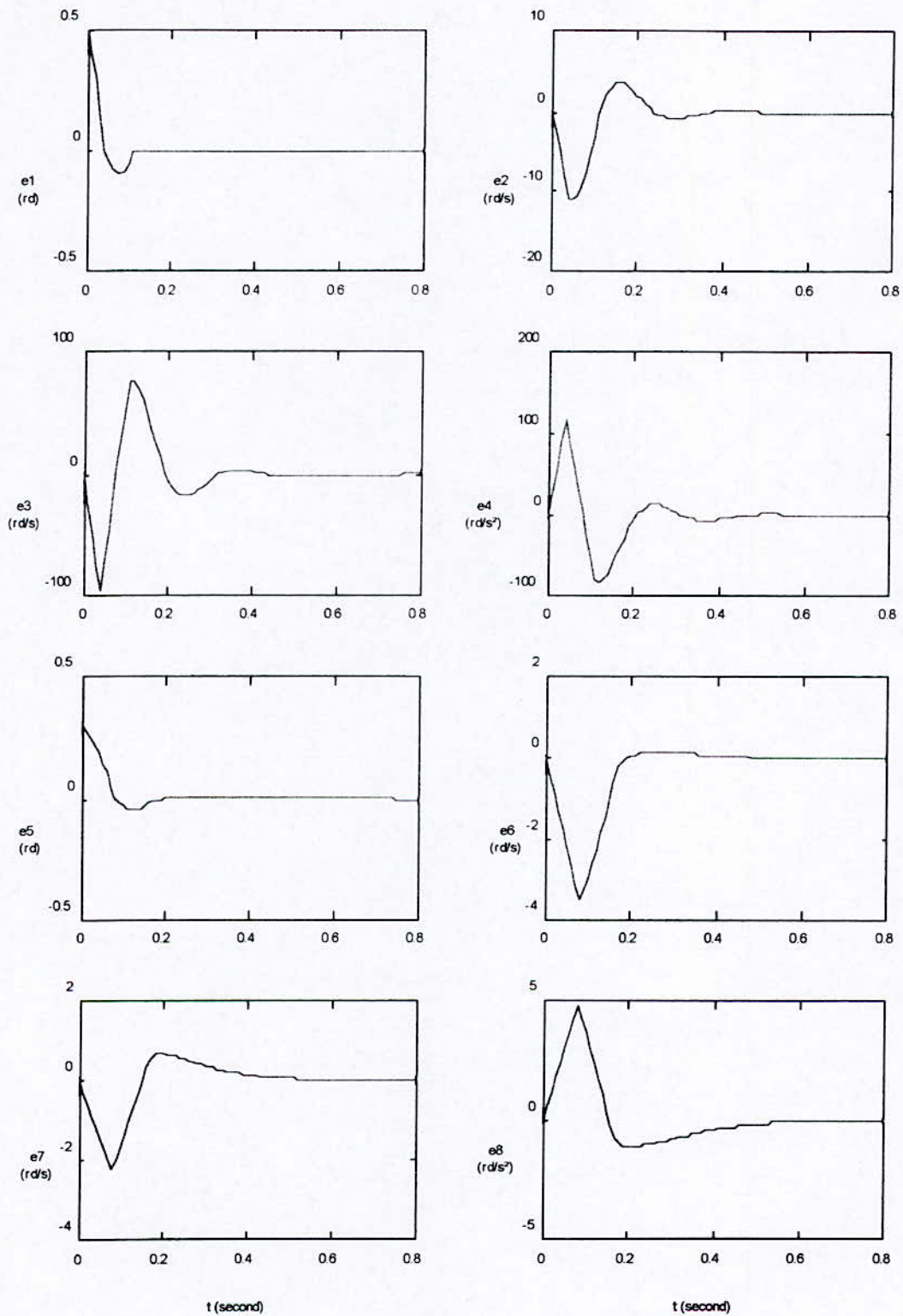


Figure 5.6.1 : Simulation sans perturbation paramétrique

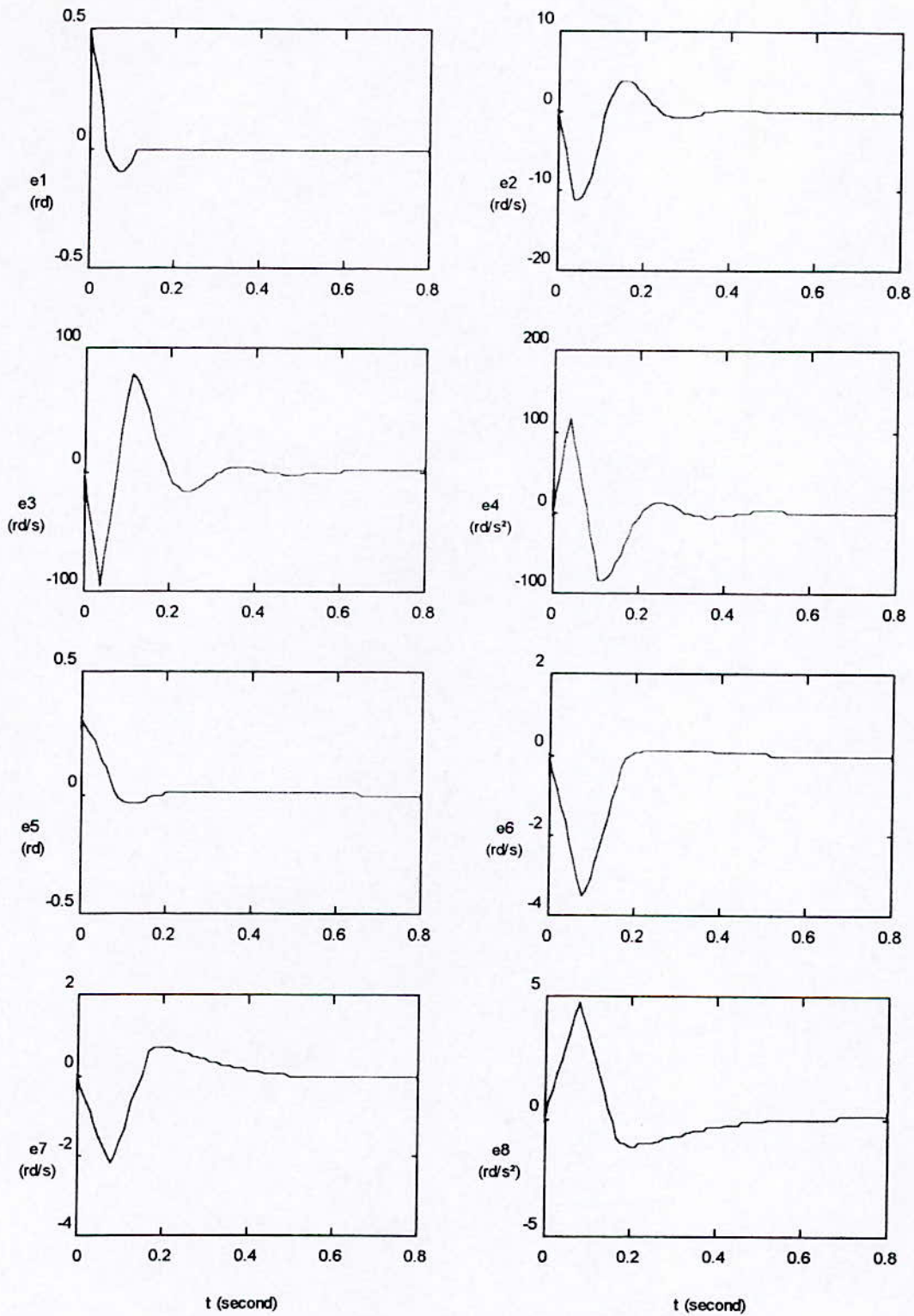


Figure 5.6.2 : Simulation avec perturbation paramétrique de 25%

5.7 Observateur à base de Backstepping

Vu l'efficacité de la méthode du « Backstepping », on l'a utilisé afin de synthétiser un observateur pour notre modèle. Cet observateur a l'avantage d'être robuste devant les perturbations et sa synthèse est systématique.

On synthétise un observateur à base de Backstepping pour notre modèle du simulateur d'hélicoptère écrite sous la forme (5.2.1). On suppose que x_3 et x_7 sont mesurables, cette contrainte est pratiquement réalisable parce que x_7 et x_3 représentent respectivement la vitesse du moteur principale et la vitesse du moteur secondaire.

Donc l'observateur « Backstepping » sera de la forme suivante :

$$\begin{aligned}
 \dot{\hat{x}}_1 &= \hat{x}_2 + \Lambda_1 \\
 \dot{\hat{x}}_2 &= -\alpha_1 \hat{x}_2 + \alpha_4 x_3 + \Lambda_2 \Lambda_1 \\
 \dot{\hat{x}}_3 &= \hat{x}_4 + \Lambda_3 \\
 \dot{\hat{x}}_4 &= -\alpha_6 x_3 - \alpha_5 \hat{x}_4 + \alpha_6 u_1 + \Lambda_4 \\
 \dot{\hat{x}}_5 &= \hat{x}_6 + \Lambda_5 \\
 \dot{\hat{x}}_6 &= -\alpha_9 \hat{x}_6 + \alpha_8 x_7 + \Lambda_6 \Lambda_5 \\
 \dot{\hat{x}}_7 &= \hat{x}_8 + \Lambda_7 \\
 \dot{\hat{x}}_8 &= -\alpha_{12} x_7 - \alpha_{11} \hat{x}_8 + \alpha_{12} u_2 + \Lambda_8
 \end{aligned} \tag{5.7.1}$$

5.7.1 Dynamique de l'erreur ($e = x - \hat{x}$)

La dynamique de l'erreur est donnée par :

$$\begin{aligned}
 \dot{e}_1 &= e_2 - \Lambda_1 \\
 \dot{e}_2 &= -\alpha_1 e_2 + v_1 - \Lambda_2 \Lambda_1 \\
 \dot{e}_3 &= e_4 - \Lambda_3 \\
 \dot{e}_4 &= -\alpha_5 e_4 - \Lambda_4 \\
 \dot{e}_5 &= e_6 - \Lambda_5 \\
 \dot{e}_6 &= -\alpha_9 e_6 + v_2 - \Lambda_6 \Lambda_5 \\
 \dot{e}_7 &= e_8 - \Lambda_7 \\
 \dot{e}_8 &= -\alpha_{11} e_8 - \Lambda_8
 \end{aligned} \tag{5.7.2}$$

Pour la synthèse des gains Λ_i , on fait la procédure suivante :

☞ premier sous système :

Etape 1 : Supposons que la surface de glissement est statique, donc $s_1 = e_1 = x_1 - \hat{x}_1$, avec une fonction de Lyapunov candidate donnée $V(s_1) = \frac{1}{2} s_1^2$ sa dérivée en fonction du temps est :

$$\begin{aligned} \dot{V}(s_1) &= s_1 \dot{s}_1 \Rightarrow \dot{V}(s_1) = s_1(e_2 - \Lambda_1) \\ \dot{V} &< c|s_1| - s_1 \Lambda_1 \quad \text{avec} \quad |e_2| < c \\ \dot{V} &< s_1(c \operatorname{sign} s_1 - \Lambda_1) \quad \text{avec} \quad |s_1| = s_1 \operatorname{sign} s_1 \\ \Lambda_1 &= c \operatorname{sign} s_1 + k_1 s_1 \end{aligned} \quad (5.7.3)$$

dans ce cas $\dot{V} < -k_1 s_1^2$, avec $k_1 > 0$, donc V est une fonction de Lyapunov.

Etape 2 : La fonction candidate choisie est : $V(s_1, e_2) = \frac{1}{2}(s_1^2 + e_2^2)$, sa dérivée en fonction du temps :

$$\dot{V}(s_1, e_2) = s_1 \dot{s}_1 + e_2 \dot{e}_2 \Rightarrow \dot{V} < -k_1 s_1^2 + e_2 \dot{e}_2$$

Sur la surface de glissement où :

$$\Lambda_1 = \Lambda_{1eq} = c \operatorname{sign} s_1$$

et :

$$\begin{aligned} \dot{V} &< -k_1 s_1^2 + e_2(-\alpha_1 e_2 + v_1 - \Lambda_2 c \operatorname{sign} s_1) \\ \dot{V} &< -k_1 s_1^2 - \alpha_1 e_2^2 + a|e_2| - e_2 \Lambda_2 c \operatorname{sign} s_1 \\ \dot{V} &< -k_1 s_1^2 - \alpha_1 e_2^2 + e_2(a \operatorname{sgn} e_2 - \Lambda_2 c \operatorname{sign} s_1) \end{aligned}$$

donc :

$$\Lambda_2 = \frac{a \operatorname{sign} e_2}{c \operatorname{sign} s_1} = -\frac{a}{c} \quad \text{avec : } a > 0, c > 0$$

On voit bien que $\dot{V} < -k_1 s_1^2 - \alpha_1 e_2^2$, donc V est une fonction de Lyapunov.

Etape 3 : On remarque qu'on peut utiliser un observateur linéaire pour les états e_3 et e_4 . Donc, supposons que $\Lambda_3 = l_1 e_3$ et $\Lambda_4 = l_2 e_3$. Avec un placement de pôles p_1, p_2 on a :

$$\begin{aligned} l_1 &= p_1 + p_2 - \alpha_5 \\ l_2 &= p_1 p_2 + \alpha_5^2 - \alpha_5(p_1 + p_2) \end{aligned}$$

☞ deuxième sous système

Pour le deuxième sous système, les mêmes étapes sont suivies et on obtient :

$$\Lambda_5 = d \operatorname{sign} s_2 + k_2 s_2$$

$$\Lambda_6 = -\frac{b}{d}$$

$$\Lambda_7 = l_3 e_7$$

$$\Lambda_8 = l_4 e_7$$

donc :

$$l_3 = p_3 + p_4 - \alpha_{11}$$

$$l_4 = p_3 p_4 + \alpha_{11}^2 - \alpha_{11}(p_3 + p_4)$$

et : $|e_6| < d, k_2 > 0, p_3, p_4$ sont des pôles pour les états e_7, e_8

5.7.2 Résultats de simulation

Les simulations de l'observateur par « Backstepping » sont données dans les figures (5.7.1.a, 5.7.1.b).

La figure 5.7.1.a, représente la dynamique de l'observateur sans l'introduction d'une perturbation, cependant dans la figure 5.7.1.b, une perturbation paramétrique de 25 % est ajoutée, on remarque une insensibilité de l'observateur devant ces perturbations. L'erreur d'observation converge exponentiellement mais avec une réponse un peu lente (pour e_2).

Simulation avec l'observateur backstepping

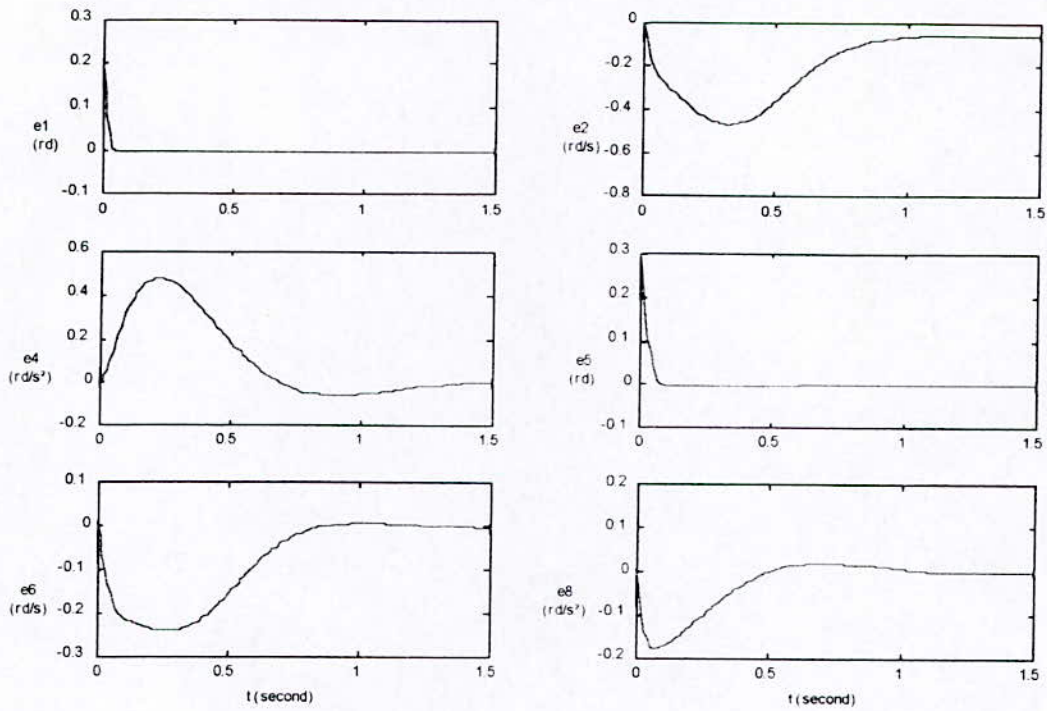


Figure 5.7.1.a : Simulation sans perturbation paramétrique

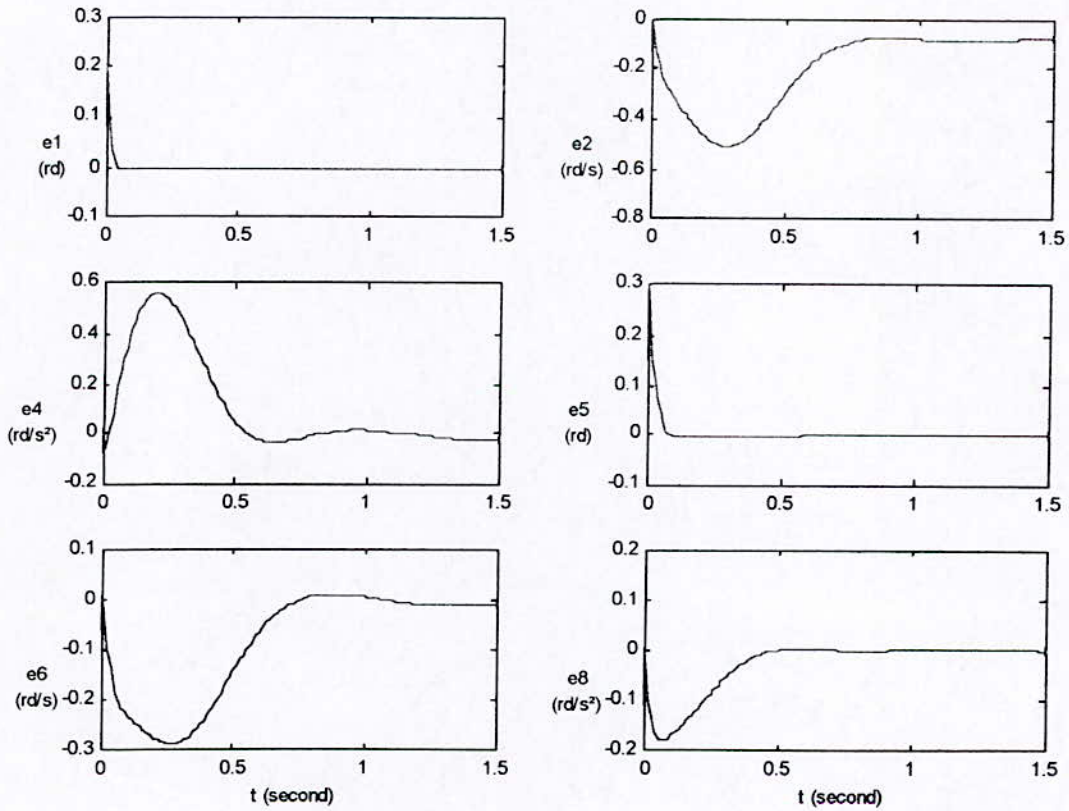


Figure 5.7.1.b : Simulation avec perturbation paramétrique de 25%

5.8 Stabilité du compensateur dynamique

Après avoir synthétiser plusieurs types d'observateurs, on doit maintenant assurer la stabilité du système via un observateur. Ceci est très important, car la synthèse séparée de la commande et de l'observateur n'assure pas toujours une stabilité globale du système bouclé (ce problème ne se pose pas dans le cas linéaire, car on dispose du principe de séparation).

On choisit l'observateur de Thau, et la commande à base du « Backstepping » pour démontrer la stabilité globale. En appliquant le théorème exposé dans la partie théorique des observateurs non linéaires, on déduit facilement que la stabilité globale est assurée.

La commande à base de « Backstepping » satisfait les conditions de la convergence exponentielle, et l'observateur de Thau le fait aussi. Ces conditions sont mentionnées dans l'annexe B.

5.8.1 Résultats de simulation

Les figures 5.8.1 et 5.8.2, représentent les graphes de simulation en boucle fermée en utilisant l'observateur de Thau et la commande par « Backstepping ». Une perturbation paramétrique de 25 % est introduite dans la figure 5.8.2. A travers ces simulations, on voit que la stabilité est garantie, la dynamique de l'observateur est acceptable et qu'aucune dégradation des performances n'est arrivée.

5.9 Conclusion

Dans ce chapitre, on a appliqué les différentes stratégies de commandes et d'observateurs au simulateur d'hélicoptère, on a donné brièvement les calculs de synthèse. Après chaque type de commande et observateur, des graphes de simulation sont insérés pour vérifier les résultats.

Simulation en boucle fermée en utilisant l'observateur de Thau et la commande par backstepping

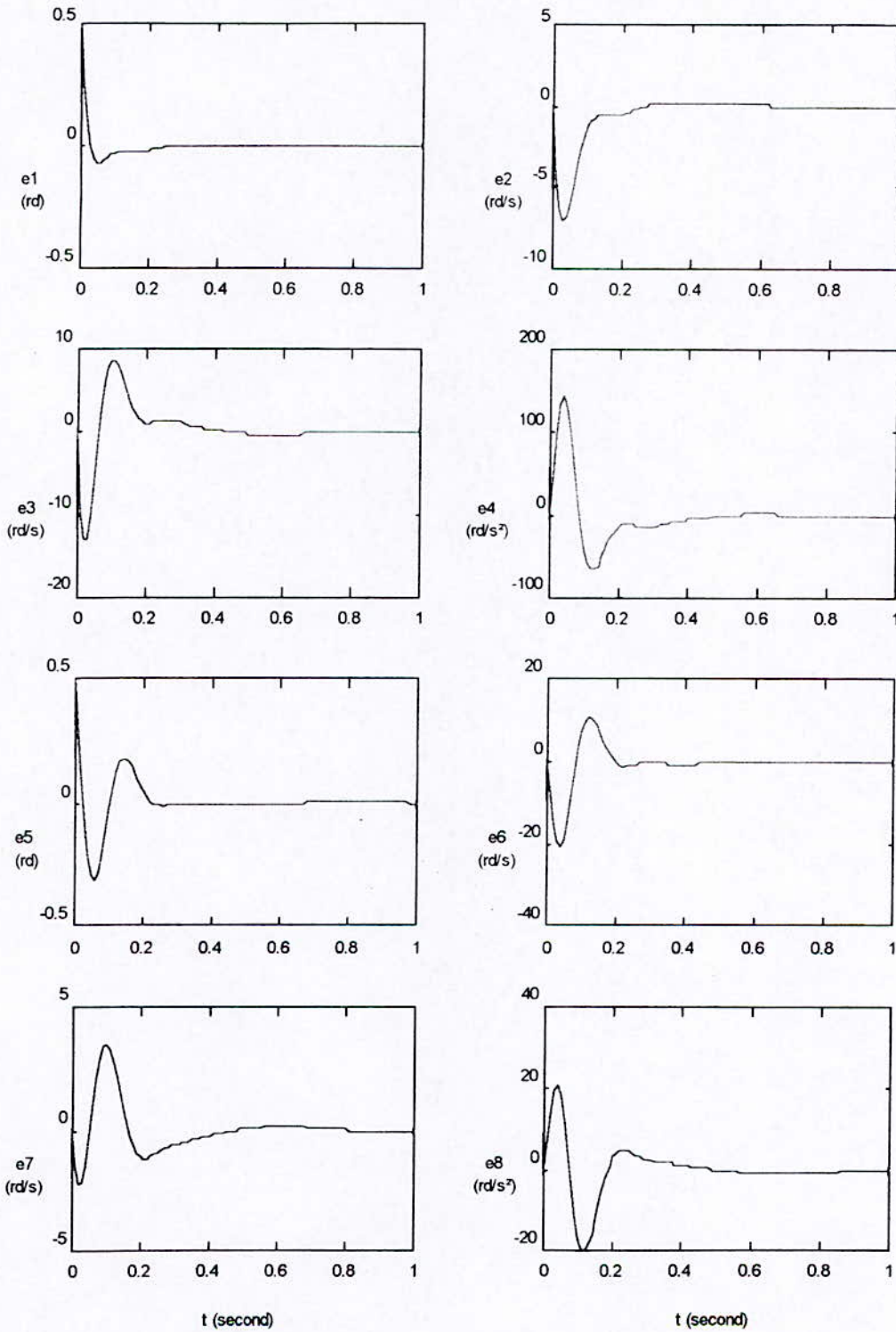


Figure 5.8.1 : Simulation sans perturbation paramétrique

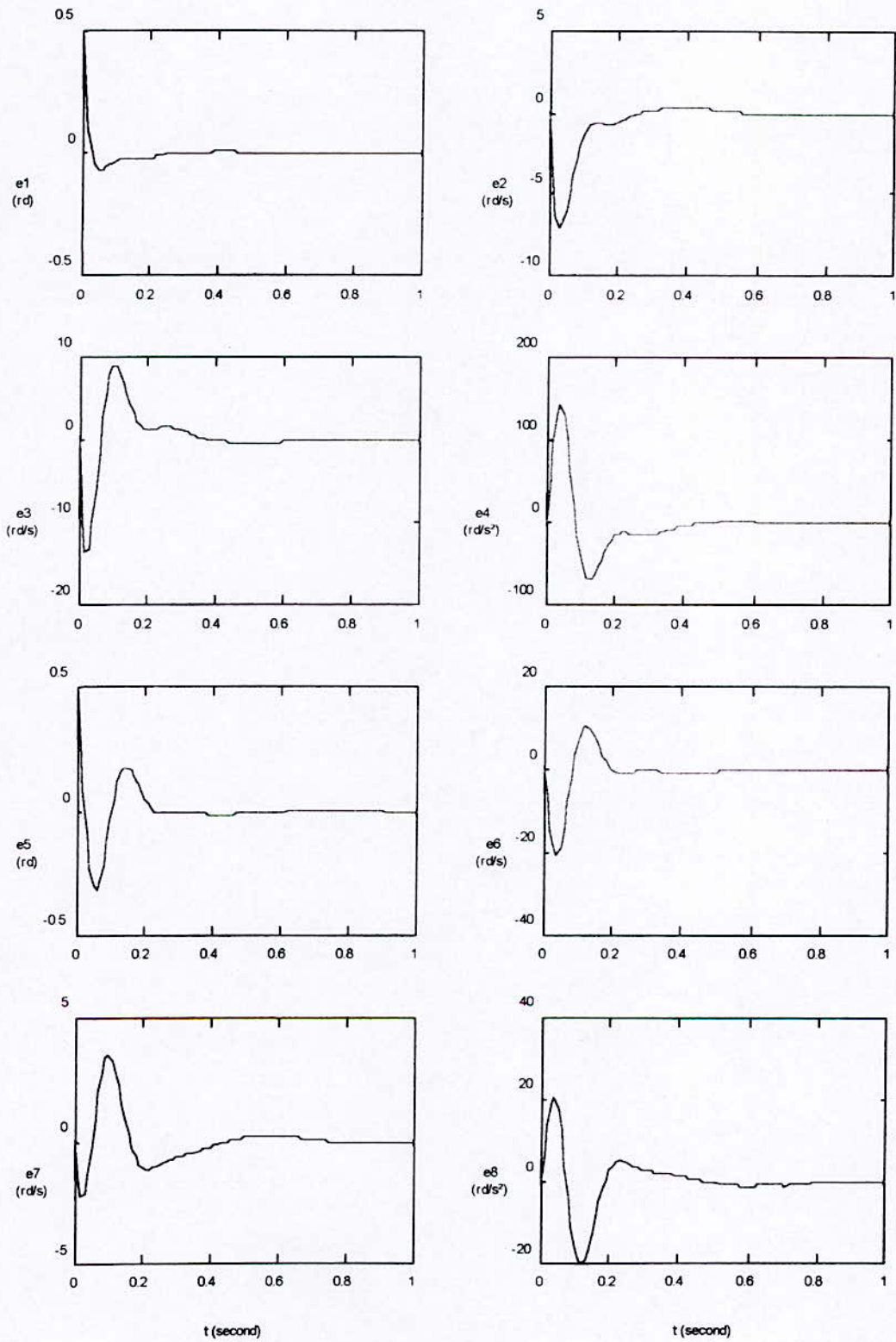


Figure 5.8.2 : Simulation avec perturbation paramétrique de 25 %

5.10 Etude comparative

5.10.1 Etude comparative entre les commandes appliquées

Commande avec linéarisation par retour d'état	
Avantages	Inconvénients
<ul style="list-style-type: none"> • Possibilité d'utilisation des techniques de commande linéaire (t.q. placement de pôles) ce qui facilite l'imposition des dynamiques désirées. • Rigoureuse mathématique. 	<ul style="list-style-type: none"> • Risque de rencontrer des points singuliers, ce qui rend la commande non valide dans certains points du domaine. • Sensibilité : en effet, elle doit être appliquée pour les modèles exactes.

Commande par logique floue	
Avantages	Inconvénients
<ul style="list-style-type: none"> • On n'a pas besoins de tous les états du système. • Le temps de réponse est imposable. • La synthèse de la commande est simple. 	<ul style="list-style-type: none"> • Utilisation du tâtonnement pour trouver les gains d'adaptations.

Commande à structure variable	
Avantages	Inconvénients
<ul style="list-style-type: none"> • Très robuste devant les perturbations. • Possibilité de l'utiliser avec d'autres commandes (Backstepping, linéarisation par retour d'état et logique floue...). • Si le système équivalent est linéaire, donc il y a possibilité d'utiliser les techniques de la commande linéaire (imposition de la dynamique) 	<ul style="list-style-type: none"> • Problème de trouver la surface de commutation pour certains classes de systèmes. • Elle nécessite des commutateurs à hautes fréquences pour l'implémentation. • Exigeante en calcul pour l'ordinateur surtout avec la fonction <i>sign</i>.

Commande par Backstepping	
Avantages	Inconvénients
<ul style="list-style-type: none"> • Possibilité de majoration des perturbations et des nonlinéarités. • Synthèse systématique de la surface de glissement. 	<ul style="list-style-type: none"> • Elle ne garantie que la stabilité et non pas les performances du système. • Beaucoup de calcul pour la synthèse d'une commande (par rapport aux autres commandes).

5.10.2 Etude comparative entre les observateurs appliqués

Observateur de Thau	
Avantages	Inconvénients
<ul style="list-style-type: none"> • Convergence exponentielle de l'observateur. • Rigoureuse mathématique. • Utilisation d'un terme de correction de type linéaire. 	<ul style="list-style-type: none"> • Il n'offre pas une méthode systématique pour la décomposition du système sous la forme $\dot{x} = Ax + Bu + f(x)$. • On doit vérifier la condition de Lipschitz. • Problème du choix des matrices L et Q. • Dégradation des performances avec des perturbations paramétriques.

Observateur glissant	
Avantages	Inconvénients
<ul style="list-style-type: none"> • Possibilité d'imposition de la dynamique. • Robustesse devant les perturbations paramétriques. • Il ne nécessite pas une connaissance exacte des non linéarités du système mais seulement leurs bornes. 	<ul style="list-style-type: none"> • Exige des commutateurs à haute fréquence donc problème d'implémentation. • Exige des gains de commutation assez élevés. • Pour certains systèmes, la majoration n'est pas évidente. • Si on utilise la fonction <i>sat</i> au lieu de <i>sign</i>, le phénomène de <i>chattering</i> sera diminué, mais la convergence n'est pas asymptotique.

Observateur à base de Backstepping	
Avantages	Inconvénients
<ul style="list-style-type: none"> • Possibilité de dépasser le problème des perturbations et des non linéarités. • Synthèse systématique de l'observateur. • Robustesse vis à vis des perturbations paramétriques. 	<ul style="list-style-type: none"> • Elle ne garantit que la stabilité et non pas les performances. • On ne peut pas imposer les performances de toutes les erreurs d'observation. • La majoration de la fonction de Lyapunov n'est pas toujours possible. • Elle nécessite des outils mathématiques puissants.

5.11 Animation

On a développé un programme d'animation graphique du simulateur d'hélicoptère en deux plans, horizontal et latéral. Et voici un schéma représentatif de cette animation avec une commande par mode glissant :

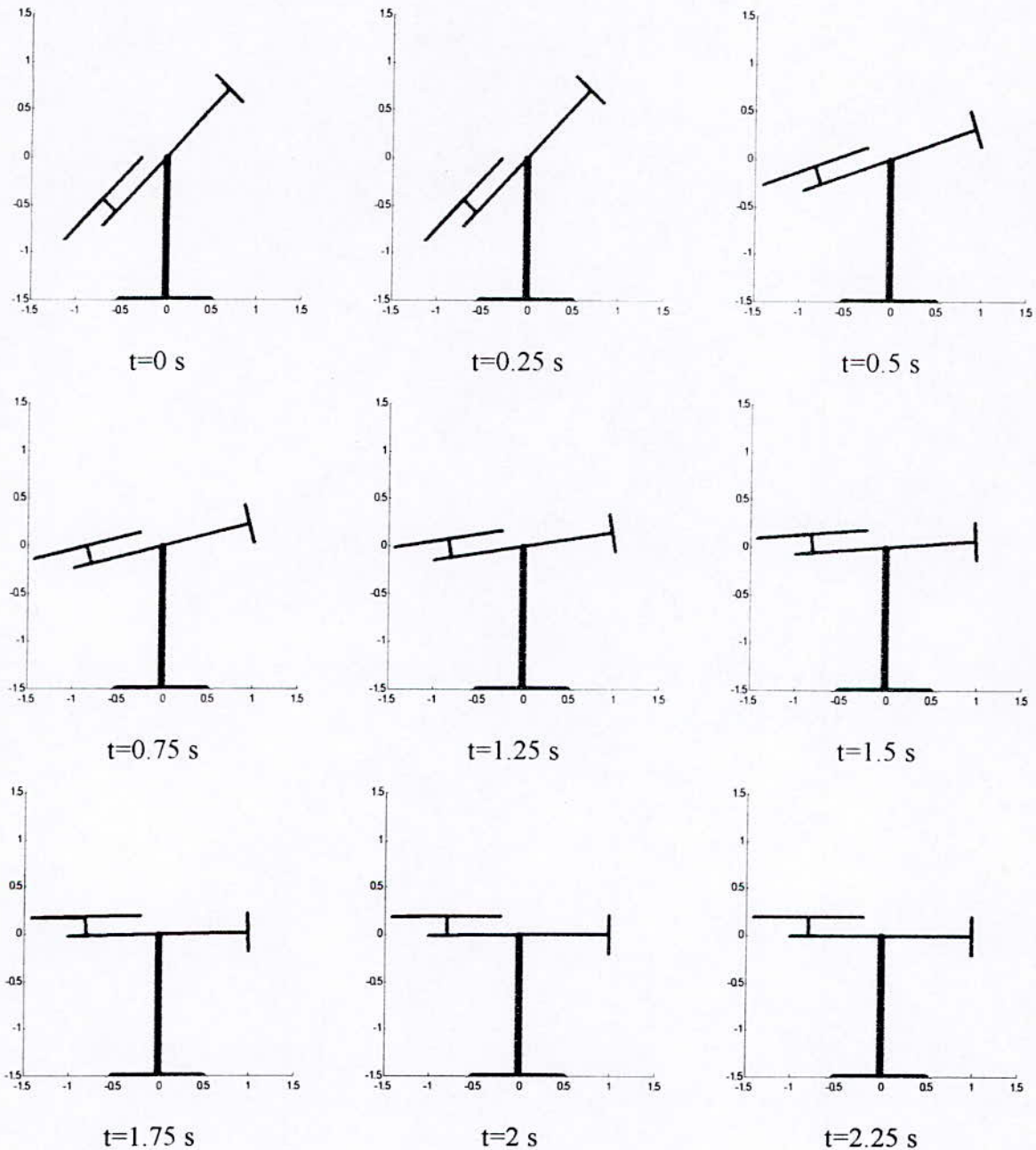


Figure 5.11.1 : Vue sur le plan latéral

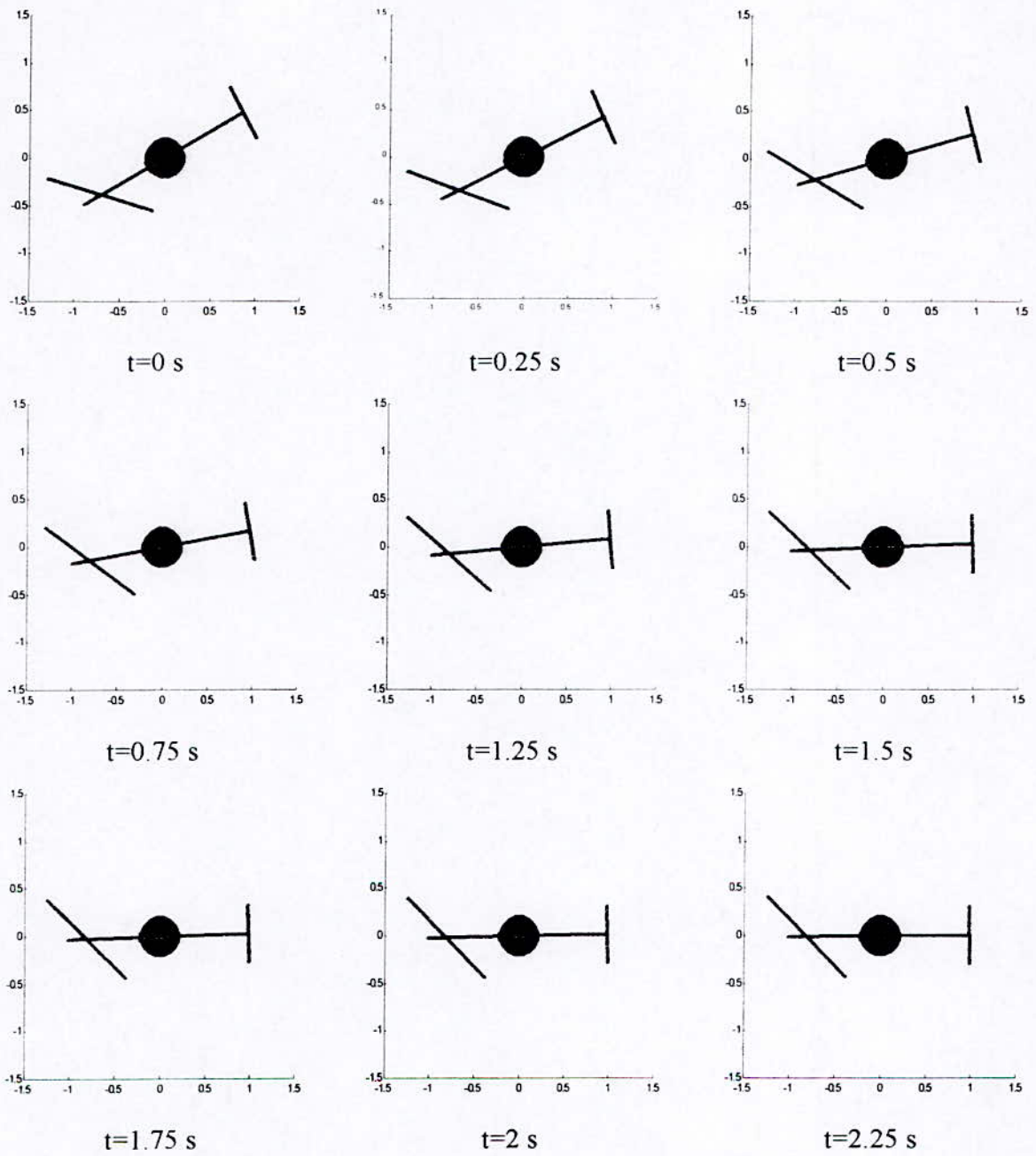


Figure 5.11.2 : Vue sur le plan horizontal



CONCLUSION
GENERALE

Conclusion générale

Dans ce travail, quatre types de commandes et trois types d'observateurs sont présentés avec leurs applications pour la stabilisation du simulateur d'hélicoptère. Ce dernier est un processus de laboratoire, il est à deux entrées et deux sorties avec un couplage important, instable en boucle ouverte.

Notre objectif était de synthétiser le plus grand nombre possible de commandes et d'observateurs non linéaires, et de les implémenter sur le système réel pour pouvoir comparer. Mais les conditions n'ont pas permis de réaliser ce but. Donc on s'est arrêté à l'étape de simulation par ordinateur. Ces simulations ont montré les avantages et les inconvénients de chaque type de commande une fois appliquée à notre système. Une étude comparative entre les commandes et les observateurs est présentée à la fin du chapitre cinq consacré à l'application et aux simulations.

Concernant les types de commandes appliquées, on trouve en premier lieu la linéarisation par retour d'état, cette commande est conseillée au début de chaque étude, car elle permet à l'ingénieur de bien découvrir les caractéristiques de son système (parties non linéaires, couplage et degré relatif). En effet, cette méthode prend le modèle du système sous sa forme initiale et ne fait aucune supposition sur lui, mais cette technique a des insuffisances majeures, telles que les points singuliers et la non robustesse.

La deuxième technique appliquée est la commande à structure variable. Cette méthode produit des commandes robustes et donc permet de pallier le problème d'incertitudes de modélisation. Le problème d'incertitude est toujours rencontré dans la pratique, parce qu'en réalité, aucun système n'est représenté strictement par un modèle donné. La CSV semble très robuste devant ces variations paramétriques.

La méthode du « Backstepping » repose sur la théorie de Lyapunov. Elle permet de faire des majorations et donc on a utilisé cet avantage pour compenser les termes non linéaires et le couplage de notre système. Avec cette technique, on a eu des résultats satisfaisants, mais le temps de calcul était très grand.

La dernière approche utilisée est la logique floue, l'avantage de celle ci est qu'elle n'exige pas une connaissance complète du système et qu'elle est composée d'opérations logiques et arithmétiques simples, ce qui veut dire que le temps de calcul est petit. Mais le tâtonnement reste le seul recours pour trouver les gains du régulateur

Pour les observateurs, on a exposé brièvement la théorie d'observation en utilisant trois types différents. Le premier est l'observateur de Thau, sa méthode ne constitue pas une technique systématique pour la synthèse d'un observateur, mais donne une condition suffisante de stabilité exponentielle de l'erreur d'observation. Son inconvénient est qu'il doit vérifier la condition de Lipshitz pour la partie non linéaire du système, cette condition ne peut pas être vérifiée pour tout le domaine de commande.

L'observateur glissant est aussi appliqué, il a donné de bon résultats, car il est basé sur la théorie du mode glissant qu'on a cité ses avantages. Enfin, l'observateur à base de Backstepping est introduit et appliqué à notre système, il est synthétisé de la même façon que la commande à base de « Backstepping ».

Comme perspective à notre travail, on peut citer les points suivants :

- Application consistant à essayer d'implémenter les techniques développées à travers ce mémoire sur un simulateur réel d'hélicoptère.
- Démonstration théorique de la stabilité globale du compensateur dynamique pour les différentes combinaisons des commandes et observateurs.
- Développement de la théorie de l'observateur du « Backstepping ».
- Essayer d'autre types de commandes et observateurs, afin de faire la comparaison entre eux.



REFERENCES
BIBLIOGRAPHIQUES

- [1] Akesson M., E. Gustafson and K.H. Johansson, "Control design for a helicopter lab process", 13th IFAC World congress, San Francisco, Etats-Unis, 1996.
- [2] Benchaib A., F.Boudjema and A. Rachid, "A new sliding mode flux observer based on backstepping method for induction motor", World Automation Congress WAC'98, Anchorage, Alaska, Etats-Unis, 9-17, 1998.
- [3] Bühler H., "Réglage par logique floue", Lausanne : Presses polytechniques et universitaires romandes, 1994.
- [4] Bühler H., "Réglage par mode de glissement", Lausanne : Presses polytechniques et universitaires romandes, 1986.
- [5] Gauthier J.P. and G. Bornard, "Observability for any $u(t)$ of class of nonlinear systems", IEEE Trans. Automat. Control, AC-26, pp. 922-926, Aug 1981.
- [6] Horacek P., "Technical Manual CE150 Helicopter Model", 1993.
- [7] Isidori A., "Nonlinear control systems-Second Edition ", New York : Springer-Verlag, 1989.
- [8] Jamshidi M., N. Viadiec and J.R. Timothy, "Fuzzy logic and control", New jersey : Prentice-Hall, 1993.
- [9] Khadir T., "Commande robuste par H_∞ théorie et application", Thèse de projet de fin d'étude de l'ENP, 1998.
- [10] Khalil H., "Nonlinear systems", New Jersey : Prentice-Hall, 1996.
- [11] Khelfi M.F., "Observateurs non linéaires : Application à la commande des robots manipulateurs", Thèse de Doctorat de l'Université Henry Poincaré Nancy I, 1995.
- [12] Krstic M., I. Kanellakopoulos and P. Kokotovic, "Nonlinear and adaptive control design", New York : John Wiley & Sons, 1995.
- [13] Matthews G.P., "Variable Structure Control of Interconnected Nonlinear", Etats-Unis, Micro film, 1985.
- [14] Melahi A. et A. Boulahia , "Commande floue et neuro floue décentralisée : Application en robotique", Thèse de projet de fin d'étude de l'ENP, 1997.
- [15] Misawa E.A. and J.K. Hedrick "Nonlinear Observer-A state-of-the- Survey", ASME J. Dynamic Systems, Measurement and Control 111, pp. 344-352, september 1989.
- [16] Pedrycz W., "Fuzzy control and fuzzy systems", New York : Prentice-Hall, 1994.

- [17] Shing JYH., R. Jang and C. Tsai, "Neuro-fuzzy modeling and control", Proceeding of IEEE, MARCH 1995 VOL 83, p 378-387.

- [18] Slotine J.J.E. and W. Li, "Applied Nonlinear Control", New Jersey : Prentice-Hall Englewood,1991 .

- [19] Slotine J.J.E., J.K. Hedrick and E.A. Misawa, "On sliding observers for nonlinear systems", ASME J. Dynamic Systems, Measurement and Control 109, pp. 245-252, september 1987.

- [20] Vydiasagar M., "Robotics Control Systems", New Jersey : Prentice-Hall, 1989.

- [21] "Fuzzy Logic Toolbox User's Guide", Mathworks, Inc,1998



ANNEXES

Annexe A

Soit $\mathbf{f}: \mathbf{R}^n \rightarrow \mathbf{R}^n$, $h(\mathbf{x})$ une fonction scalaire lisse, le gradient de h noté par ∇h :

$$\nabla h = \frac{\partial h}{\partial \mathbf{x}}$$

Le gradient est représenté par un vecteur ligne d'éléments $(\nabla h)_i = \partial h / \partial x_i$. De même, soit un vecteur $\mathbf{f}(\mathbf{x})$, le Jacobien de \mathbf{f} est noté par $\nabla \mathbf{f}$:

$$\nabla \mathbf{f} = \frac{\partial \mathbf{f}}{\partial \mathbf{x}}$$

il est représenté par une matrice $n \times n$ d'éléments $(\nabla \mathbf{f})_{ij} = \partial f_i / \partial x_j$.

Dérivée de Lie & crochet de Lie

Soit une fonction scalaire $h(\mathbf{x})$ et un vecteur $\mathbf{f}(\mathbf{x})$, on définit une nouvelle fonction scalaire $L_{\mathbf{f}}h$, appelée la dérivée de Lie (ou simplement, la dérivée) de h respectant \mathbf{f} .

Définition 1 : Soit $h: \mathbf{R}^n \rightarrow \mathbf{R}$ une fonction scalaire lisse, et $\mathbf{f}: \mathbf{R}^n \rightarrow \mathbf{R}^n$ un vecteur dans \mathbf{R}^n , donc la *dérivée de Lie de h respectant \mathbf{f}* est une fonction scalaire définie par $L_{\mathbf{f}}h = \nabla h \mathbf{f}$.

On définit aussi un autre opérateur mathématique important qui est le crochet de Lie.

Définition 2 : Soit \mathbf{f} et \mathbf{g} deux vecteur de \mathbf{R}^n . Le crochet de Lie est un autre champ de vecteur défini par :

$$[\mathbf{f}, \mathbf{g}] = \nabla \mathbf{g} \mathbf{f} - \nabla \mathbf{f} \mathbf{g}$$

Le crochet de Lie $[\mathbf{f}, \mathbf{g}]$ est écrit généralement $ad_{\mathbf{f}}\mathbf{g}$ (où *ad* signifie 'adjoint'). On a aussi :

$$\begin{aligned} ad_{\mathbf{f}}\mathbf{g} &= \mathbf{g} \\ ad_{\mathbf{f}}\mathbf{g} &= [\mathbf{f}, ad_{\mathbf{f}}\mathbf{g}] \end{aligned}$$

Annexe B

Théorème de la stabilité exponentielle

Soit le système non linéaire suivant :

$$\begin{cases} \dot{x} = f(x, u) \\ y = Cx \end{cases} \quad \text{avec} \quad f(0,0) = 0$$

Le point d'équilibre $x = 0$ est globalement exponentiellement stable s'il existe une fonction de Lyapunov V et des constantes positives $\alpha_1, \alpha_2, \alpha_3$ et α_4 telles que $V(x)$ vérifie pour tout $x \in \mathfrak{R}^n$:

$$\alpha_1 \|x\|^2 \leq V(x) \leq \alpha_2 \|x\|^2$$

$$\left\| \frac{\partial V}{\partial x} \right\| \leq \alpha_3 \|x\|$$

$$\dot{V}(x) \leq -\alpha_4 V(x)$$

في هذا العمل، نهتم بإنشاء عدة أنظمة تحكم (تحويل إلى الشكل الخطي برجع الحالة، نظام الانزلاق، الرجوع المرحلي و المنطق الغامض) ومرصد ("تو"، نظام الانزلاق والرجوع المرحلي)، والهدف هو استقرار جهاز تمثيل الطيران لطائرة عمودية، يتميز هذا النظام : بالخطية، متعدد المداخل والمخارج، ارتباط معتبر وعدم استقرار في الحلقة المفتوحة.

الكلمات الجوهرية : التحويل إلى الشكل الخطي برجع الحالة، نظام الانزلاق، المنطق الغامض، الرجوع المرحلي، مرصد "تو"، الاخطي، نظام تحكم، مرصد، نموذج الطائرة العمودية.

Abstract

In this work, we are interested to design different control laws (feedback linearization, sliding mode, Backstepping, fuzzy logic) and observer (Thau, sliding mode, Backstepping) to stabilise a helicopter flight simulator, this system is MIMO, nonlinear with an important interaction and unstable open loop dynamic.

Keywords : Feedback linearization, sliding mode, fuzzy logic, Backstepping, Thau observer, Nonlinear, Control, Observer, helicopter model.

Résumé

Dans ce travail, nous nous sommes intéressés à la synthèse de différentes stratégies de commandes (Linéarisation par retour d'état, mode glissant, « Backstepping ») et d'observateurs (Thau, glissant, « Backstepping ») pour stabiliser un simulateur de vol d'hélicoptère, ce dernier est un système MIMO, non linéaire avec un couplage important et une dynamique instable en boucle ouverte.

Mots clés : Linéarisation par retour d'état, mode glissement, logique floue, « Backstepping », Observateur de Thau, non linéaire, commande, observateur, modèle d'hélicoptère.