

REPUBLIQUE ALGERIENNE DEMOCRATIQUE ET POPULAIRE
MINISTERE DE L'ENSEIGNEMENT SUPERIEUR ET DE LA RECHERCHE SCIENTIFIQUE
UNIVERSITE MOHAMED BOUDIAF - M'SILA

FACULTE : TECHNOLOGIE
DEPARTEMENT : GENIE ELECTRIQUE
N° :



DOMAINE : TECHNOLOGIES
FILIERE : AUTOMATIQUE
OPTION : AUTOMATIQUE ET SYSTEMES

Mémoire présenté pour l'obtention
Du diplôme de Master Académique

Par: : - *DEBAB nadjib adil*
- *KADI moustafa*

Intitulé

**Commande classique et intelligente appliquée à
un simulateur de vol d'hélicoptère TRMS**

Soutenu devant le jury composé de:

Dr. ZEGHLACHE Samir	Université Mohamed Boudiaf - m'sila	Président
Mr. GHELLAB Mohamed	Université Mohamed Boudiaf - m'sila	Rapporteur
Mr. MEKKI Hemza	Université Mohamed Boudiaf - m'sila	Examineur
Mr. AIB Abdelghani	Université Mohamed Boudiaf - m'sila	Examineur

Année universitaire : 2018 /2019

Remerciements

Tout d'abord nous remercions dieu tout puissant qui nous a gardé en bonne santé afin de faire ce modeste travail et qui nous a permis de vivre ce bonheur.

Tous nos vifs remerciements vont vers notre promoteur Mr GHELLAB Mohamed pour ses conseils chaleureux, son encadrement, sa disponibilité et son aide précieuse pour réaliser ce modeste travail, pour cela on l'assure de notre sincère gratitude et notre respect le plus profond.

Nous remercions les membres du jury qui nous ont fait l'honneur d'examiner notre modeste travail.

Merci à toute personne ayant contribué de près ou de loin à la concrétisation de ce projet.



Dédicaces

A ma chère mère khadra missat

A mes frères

A mes soeurs

A mes amis

Debab Nadjib adil



Dédicaces

*Je dédie ce modeste travail a
MES tres chers parents, MA
future femme, MES frères, MES
soeurs et toute ma famille, et a
tous ceux qui m'ont soutenue*

Moustafa KADI

Tables Des Matières

Table des Matières.....	I
Liste des Figures	II
Liste des Tableaux.....	III
Introduction générale.....	IV

I. Présentation matériel du simulateur de vol d'hélicoptère TRMS

I.1 Introduction	1
I.2 Principe du vol d'un hélicoptère	1
I.2.1 Les forces agissantes sur l'hélicoptère en vol.....	1
I.2.2 Principe du vol de l'hélicoptère.....	2
I.3 Description du Simulateur.....	3
I.3.1 Propulseurs.....	5
I.3.2 Commande des moteurs	5
I.3.3 Les encodeurs optiques.....	5
I.3.4 La base du TRMS.....	6
I.3.5 Boîtier marche/arrêt.....	7
I.3.6 Le (SCSI BOX)	7
I.3.7 Carte d'acquisition : ADVANTECH PCI 1711.....	8
I.4 Environnement de développement.....	9
I.5 Conclusion.....	13

II. Modélisation du simulateur de vol d'hélicoptère

II.1 Introduction	14
II.2 Modèle dynamique.....	14
II.2.1 Sous système d'élévation.....	15
II.2.2 Sous système d'azimut.....	16
II.3 Modèle d'état.....	18
II.4 Paramètres du modèle.....	18

II.5	Simulation.....	21
II.6	Validation du modèle de simulation.....	23
II.7	Identification statique.....	23
II.8	Validation du nouveau modèle du simulation	25
II.9	Conclusion.....	26

III. Régulateur PID et leurs applications en temps réel

III.1	Introduction.....	27
III.2	la commande par le régulateur type PID.....	27
III.3	Structure des correcteurs PID.....	28
III.4	Avantages et inconvénients.....	29
III.5	Domaine d'application.....	29
III.6	Résultat de validation de commande PID en système réel.....	30
III.7	Test de robustesse pour les positions angulaires du TRMS.....	37
III.8	Conclusion.....	39

IV. Commande Intelligente type floue et leur implémentation en temps réel pour TRMS

IV.1	Introduction	40
IV.2	historique.....	40
IV.3	la commande par logique floue.....	41
IV.3.1	Notion de base sur la logique floue	41
IV.3.1.1	Ensembles ordinaires.....	41
IV.3.1.2	Ensemble flou.....	41
IV.3.1.3	Variables linguistiques.....	42
IV.3.1.4	Fonction d'appartenance.....	42
IV.3.2	Raisonnement flou.....	43
IV.3.3	Commande par la logique floue	44
IV.3.3.1	Contrôleur flou.....	44
IV.3.4	Types de régulateurs flous	46
IV.3.4.1	Régulateur de type Mamdani.....	46

IV.3.4.2	Régulateur de type Sugeno.....	46
IV.3.5	Procédé lors de la conception d'un réglage par logique floue.....	47
IV.4	Principe d'un régulateur floue.....	48
IV.5	Avantages et désavantages du réglage par logique floue.....	48
IV.6	Application de la commande floue.....	49
IV.6.1	Synthèse des régulateurs.....	49
IV.6.2	les règles de la fuzzification.....	50
IV-7	Résultat de validation de commande floue	52
IV-7-1	Résultats de simulation de la commande floue.....	53
IV-7-2	Résultats expérimentaux de la commande floue au TRMS.....	59
IV-8	Test de robustesse pour les positions angulaires du TRMS.....	65
IV-9	Conclusion.....	67
<i>V. Commande par réseaux de neurones appliquée au TRMS</i>		
V.1	Introduction	68
V.2	Réseaux de neurones artificiels.....	69
V.2.1	Réseaux Modèle formel d'un neurone.....	69
V.2.1.1	Opérateur de summation.....	70
V.2.1.2	Fonction de seuil (ou d'activation).....	71
V.3	Apprentissage des réseaux de neurones.....	72
V.3.1	Apprentissage supervise.....	72
V.3.2	Apprentissage non supervise.....	72
V.3.3	Surapprentissage.....	73
V.4	Mémoire associative.....	74
V.5	Algorithmes d'apprentissage.....	74
V.5.1	Algorithme de rétropropagation (Back-Propagation BP).....	74
V.5.2	Difficultés et limites de l'algorithme.....	77
V.5.3	Algorithme de back- propagation avec momentum.....	77
V.6	les étapes de la conception d'un réseaux neurone.....	78
V.6.1	Choix des échantillons.....	79

V.6.2	Normalisation des données.....	79
V.6.3	Elaboration de la structure du réseau.....	79
V.6.4	Apprentissage.....	79
V.6.5	Validation.....	79
V.7	Création de la base d'apprentissage et la validation de la commande neuronale	80
V.8	Résultats de simulation de la commande par réseaux de neurones au TRMS.....	81
V.9	Test de robustesse pour les positions angulaires du TRMS.....	87
V.10	Conclusion.....	89

VI. Etude comparative

VI.1	Introduction.....	90
VI.2	L'étude comparative.....	90
VI.3	Conclusion	93
	Conclusion générale.....	94
	Bibliographiques.....	96

Tables Des Figures

CHAPITRE I :

Figure I.1	: forces agissantes sur l'hélicoptère.....	2
Figure I.2	: principe du vol de l'hélicoptère.....	2
Figure I.3	: le simulateur d'hélicoptère TRMS (Twin Rotor Mimo System).....	4
Figure I.4	: les deux vis de fixation	4
Figure I.5	: schéma synoptique du fonctionnement du propulseur.....	5
Figure I.6	: encodeurs optiques.....	6
Figure I.7	: la base du TRMS.....	7
Figure I.8	: branchement des différents connecteurs.....	8
Figure I.9	: rapid prototyping VS processus de développement traditionnel.....	9
Figure I.10	: processus de génération du code exécutable.....	11
Figure I.11	: différentes cibles d'implémentation	11
Figure I.12	: librairie de la carte d'acquisition.....	12
Figure I.13	: exemple d'implémentation d'un contrôleur.....	12

CHAPITRE II :

Figure II.1 : Configuration du TRMS 33-949S.....	15
Figure II.2 : Schéma bloc du système TRMS 33-949S.....	20
Figure II.3 : Réponse libre du système avec : $\alpha_{v0} = 0$ et $\alpha_{h0} = 0$ et $u_v = u_h = 0$	21
Figure II.4 : Réponse du système pour $u_v = u_h = 0.5$ avec $\alpha_{v0} = 0$ et $\alpha_{h0} = 0$	22
Figure II.5 : Réponse du système pour $u_v = u_h = 0.5$ avec $\alpha_{v0} = 0$ et $\alpha_{h0} = 0$	23
Figure II.6 : réponse de système réel VS modèle de simulation en simulateur de TRMS.....	25
Figure II.7 : Réponse du système pour $u_v = u_h = 0.5$ avec $\alpha_{v0} = 0$ et $\alpha_{h0} = 0$	26

CHAPITRE III:

Figure III.1 : Structure série d'un PID.....	28
Figure III.2 : Structure parallèle d'un PID.....	28
Figure III.3 : Structure mixte d'un PID.....	28
Figure III.4 : la commande PID en simulateur de TRMS.....	30
Figure III.5 : Résultats de la commande PID avec une consigne échelon pour sous-système verticale.....	31
Figure III.6 : Résultats de la commande PID avec une consigne échelon pour sous-système horizontale.....	32
Figure III.7 : Résultats de la commande PID avec une consigne carrée pour sous-système verticale.....	33
Figure III.8 : Résultats de la commande PID avec une consigne carrée pour sous-système horizontale.....	34
Figure III.9 : Résultats de la commande PID avec une consigne sinusoïdale sous-système verticale.....	35
Figure III.10 : Résultats de la commande PID avec une consigne sinusoïdale sous-système horizontale.....	36
Figure III.11 : Résultats de la commande PID avec une consigne échelon pour sous-système verticale avec perturbation.....	37
Figure III.12 : Résultats de la commande PID avec une consigne échelon pour sous-système horizontale avec perturbation.....	38

CHAPITRE IV :

Figure IV-1 : Architecture générale d'un contrôleur flou.....	46
Figure IV-2 : Procédé lors de la conception d'un réglage par logique floue.....	47
Figure IV-3 : schéma de commande floue d'un vol d'hélicoptère TRMS.....	49

Figure IV.4 : Fuzzification avec sept fonctions d'appartenance dE.....	50
Figure IV.5 : Fuzzification avec sept fonctions d'appartenance E.....	50
Figure IV.6 : Fuzzification avec sept fonctions d'appartenance de la Sortie.....	50
Figure IV.7 : La surface de le régulateur floue.....	51
Figure IV.8: Implementation de la commande floue au simulateur de TRMS.....	52
Figure IV.9 : Résultats de la commande floue avec une consigne échelon pour sous-système verticale.....	53
Figure IV.10 : Résultats de la commande floue avec une consigne échelon Pour sous-système horizontale.....	54
Figure IV.11 : Résultats de la commande floue avec une consigne carrée pour sous-système verticale.....	55
Figure IV.12 : Résultats de la commande floue avec une consigne carrée Pour sous-système horizontale.....	56
Figure IV.13 : Résultats de la commande floue avec une consigne sinusoïdale pour sous-système verticale.....	57
Figure IV.14 : Résultats de la commande floue avec une consigne sinusoïdale Pour sous-Système horizontale.....	58
Figure IV.15 : Résultats de la commande floue avec une consigne échelon pour sous-système verticale.....	59
Figure IV.16 : Résultats de la commande floue avec une consigne échelon Pour sous-système horizontale.....	60
Figure IV.17 : Résultats de la commande floue avec une consigne carrée pour sous-système verticale.....	61
Figure IV.18 : Résultats de la commande floue avec une consigne carrée Pour sous-système horizontale.....	62
Figure IV.19 : Résultats de la commande floue avec une consigne sinusoïdale pour sous-système verticale.....	63
Figure IV.20 : Résultats de la commande floue avec une consigne sinusoïdale Pour sous-système horizontale.....	64
Figure IV.21 : Résultats de la commande floue avec une consigne échelon pour sous-système verticale.....	65
Figure IV.22 : Résultats de la commande floue avec une consigne échelon Pour sous-système horizontale.....	66

CHAPITRE V:

Figure V.1 : Neurone formel.....	69
Figure V.2 : Mise en correspondance neurone biologique/ neurone artificiel.....	70
Figure V.3 : Modèle formel d'un neurone.....	70
Figure V.4 : Schéma d'un apprentissage supervise.....	72
Figure V.5 : Schéma de l'apprentissage non supervise.....	73
Figure V.6 : Phénomène de surapprentissage.....	73
Figure V.7 : Un réseau multicouche comportant 2 neurones d'entrée, 4 neurones.....	74
Figure V.8 : Organigramme de l'apprentissage pour les réseaux multicouches.....	76
Figure V.9 : Convergence et choix de η	77
Figure V.10 : Effet de momentum pour échapper d'un minimum local.....	78
Figure V.11 : schéma d'apprentissage de réseaux neurone pour la commande de TRMS.....	80
Figure V.12 : schéma de validation la commande réseaux neurone pour TRMS.....	80
Figure V.13 : Résultats de la commande floue avec une consigne échelon pour sous-système vertical.....	81
Figure V.14 : Résultats de la commande floue avec une consigne échelon pour sous-système horizontale.....	82
Figure V.15 : Résultats de la commande floue avec une consigne carre pour sous-système verticale.....	83
Figure V.16 : Résultats de la commande floue avec une consigne carre pour sous-système horizontale.....	84
Figure V.17 : Résultats de la commande neuronale avec une consigne sinusoïdale pour sous-système vertical.....	85
Figure V.18 : Résultats de la commande neuronale avec une consigne sinusoïdale pour sous-système horizontale.....	86
Figure V.19 : Résultats de la commande floue avec une consigne échelon pour sous-système vertical avec perturbation.....	87
Figure V.20 : Résultats de la commande floue avec une consigne échelon pour sous-système horizontale avec perturbation.....	88

Liste Des Tableaux

CHAPITRE II :

Tableau II.1 : paramètres du modèle.....19

CHAPITRE III:

Tableau III-1 : les avantages et les inconvénients régulateur PID.....29

CHAPITRE IV :

Tableau IV-1 : Désignation standard des ensembles flous.....45

Tableau IV.2 : Base de règles (7*7).....49

CHAPITRE VI:

Tableaux VI.1 : la comparaison entre la commande PID et la commande floue dans le plan vertical. (Ψ).....91

Tableaux VI.2 : la comparaison entre la commande PID et la commande floue dans le plan horizontal. (φ).....91

Tableaux VI.3 : la comparaison entre la commande PID et la commande réseaux neurone dans le plan vertical. (Ψ).....92

Tableaux VI.4 : la comparaison entre la commande PID et la commande réseaux neurone dans le plan horizontal. (φ).....92

Introduction générale

Introduction générale

Le pilotage automatique d'avions et d'hélicoptères est né avec l'aviation moderne et a évolué au cours du temps pour satisfaire des besoins de plus en plus contraignants. Il peut être utilisé lorsque la tâche à réaliser est trop répétitive ou trop difficile pour le pilote. La maîtrise de la commande automatique de l'évolution des drones miniatures contrôlés par radio ouvre la voie à des applications dans les domaines de la sécurité (surveillance de l'espace aérien, du trafic urbain et interurbain), de la gestion des risques naturels (surveillance de l'activité des volcans), de l'environnement (mesure de la pollution de l'air, surveillance des forêts), pour l'intervention dans des environnements hostiles (milieux radioactifs, déminage des terrains sans intervention humaine), la gestion des grandes infrastructures (barrages, lignes à haute tension, pipelines), l'agriculture (détection et traitement des cultures infestées) et la prise de vue aérienne dans la production de films [1].

Dans le domaine de l'automatique, l'étude des systèmes non linéaires constitue toujours un champ de recherche très riche. Contrairement aux systèmes linéaires pour lesquels l'automatique fournit une panoplie de méthodes d'analyse et de synthèse de lois de la commande, les systèmes non linéaires ne disposent pas d'outils et de méthodes générales pour l'analyse et la synthèse. Ceci est dû au fait que les systèmes non linéaires possèdent des structures extrêmement variées, des dynamiques complexes et peuvent présenter toutes sortes de comportements étranges.

La plupart des systèmes non linéaires sont caractérisés par des incertitudes structurelles et / ou non structurelles variantes dans le temps, ce qui rend leur commande très délicates et complexes à mettre en œuvre. Pour résoudre ce problème, plusieurs approches ont été développées dans la littérature. Avec le progrès technologique des moyens de calcul numériques, la stabilisation des systèmes aéronautiques constitue à l'heure actuelle un axe de recherche et de développement très privilégié [2],[3],[4],[5],[6],[7],[8], Pour des raisons de sécurité et de coût élevé, il est clair que les lois de commande développées ne peuvent être appliquées sur le système réel. En effet, des plates formes d'expérimentation ont été conçues afin de permettre de les tester sur des prototypes. Parmi ces systèmes, on trouve le simulateur de vol d'hélicoptère TRMS (Twin Rotor MIMO System).

Ce simulateur est un système multi variables à deux entrées et deux sorties, caractérisé par une forte non linéarité, un couplage important et une dynamique instable en

boucle ouverte, ce qui rend sa commande très délicate et complexe à mettre en œuvre. Pour résoudre ce problème, plusieurs approches ont été développées dans la littérature

Les chercheurs dans tous les domaines s'intéressaient aux systèmes aéronautiques. Plus particulièrement, Les chercheurs automaticiens qui se sont penchés sur le problème de la commande des avions et des hélicoptères. En effet, ces engins sont des systèmes complexes non linéaires multivariables, instables dans certaines plages de vol et présentent une dynamique fortement couplée.

L'application des commandes développées en théorie n'est pas toujours systématique. Cependant, les phases de validation dans les conditions réelles s'avèrent nécessaires surtout en l'absence de modèles précis ce qui implique l'utilisation de commandes robustes. On entend par les commandes qui conservent les performances en présence d'erreurs de modélisation et de perturbations extérieures, Cette dynamique complexe implique la conception d'algorithmes de contrôle sophistiqués pouvant traiter ces difficultés. Dans ce mémoire nous avons fait, un contrôleur PID et la commande par logique floue et la commande par Réseaux de Neurone proposé pour TRMS (Twin Rotor Mimo System)

Régulateur PID est le régulateur le plus présent dans l'industrie. Ce régulateur présente l'immense avantage de ne pas nécessiter de modélisation dynamique du processus

La logique floue est une technique intelligente pour le traitement de connaissances imprécises basées, sur des termes linguistiques; elle donne les moyens de convertir une commande linguistique basée sur le raisonnement humain, en une commande automatique, permettant ainsi la commande des systèmes complexes dont les informations sont exprimées d'une façon vague et mal définie.

Dans le domaine du génie électrique, la commande par logique floue a fait l'objet de plusieurs travaux : dans la commande des convertisseurs statiques et dans la commande des machines électriques [9], [10], [11], dans la navigation de robots mobiles [12], [13]. Toutes ces applications ont démontré qu'un régulateur à logique floue est plus robuste qu'un régulateur conventionnel [14], [15] Les performances que la commande floue peut apporter par comparaison avec les commandes classiques, sont essentiellement dues à la méthode de conception de ces régulateurs. En effet, ces derniers ne nécessitent pas la connaissance des modèles mathématiques du système. Par contre ils ont besoin d'un ensemble de règles basées essentiellement sur les connaissances d'un opérateur qualifié manipulant le système

Les réseaux de neurones forment une famille de fonctions non linéaires, permettant de construire, par apprentissage, une très large classe de modèles et de contrôleurs. Un réseau de neurones est un système d'opérateurs non linéaires interconnectés, recevant des signaux de l'extérieur par ses entrées, et délivrant des signaux de sortie, qui sont en fait les activités de certains neurones

Ce mémoire est organisé en une introduction, Six chapitres et une conclusion:

➤ Le premier chapitre est nous présentons le système en question nous avons d'abord donné le principe du vol d'un hélicoptère réel. Ensuite nous avons présenté une brève description du simulateur (TRMS). Ainsi que son principe de fonctionnement. Enfin, nous avons expliqué l'environnement de développement utilisé.

➤ Dans le deuxième chapitre, Nous avons procédé par la modélisation analytique pour avoir un modèle mathématique du TRMS qui permettent de décrire avec précision suffisante les interactions entre le système et son environnement. Lorsque les relations citées auparavant sont issues des équations de la physique. Ensuite, nous validerons le modèle développé, et en fin nous présenterons des simulations en boucle ouverte du modèle.

➤ Le troisième chapitre, est consacré à l'étude et la synthèse de la commande par le régulateur PID et l'application en temps réel nous présentons les résultats de validation de commande PID en système réel

➤ Le quatrième chapitre nous présentons dans ce chapitre un bref rappel théorique sur la commande par logique floue et leur application en temps réel

➤ Le cinquième chapitre nous présentons dans ce chapitre un bref rappel théorique sur la commande par RESEAUX DE NEURONES et faire la simulation de la commande hors ligne

➤ Le sixième chapitre nous avons fait une comparaison entre les commandes classiques et les commandes intelligentes.

Enfin, nous terminons par une conclusion générale et des perspectives

CHAPITRE I

*Présentation matériel du
simulateur de vol d'hélicoptère*

TRMS

CHAPITRE

1

I. Présentation matériel du simulateur de vol d'hélicoptère TRMS

I.1 INTRODUCTION

Les systèmes aéronautiques sont devenus aujourd'hui si difficiles et complexe qu'on ne puisse pas réaliser leurs commandes par des techniques classiques. En effet, des chercheurs automaticiens se sont penchés sur ces problèmes de commande, pour cela Plusieurs prototypes ont été réalisés dans le but de tester de nouvelles techniques de commande. Le simulateur d'hélicoptère (TRMS) est l'un de ces prototypes sur lequel nous allons travailler.

Dans ce chapitre, nous allons d'abord donner un petit aperçu sur le principe du vol des hélicoptères, ensuite nous présentons en détail le simulateur.

I.2 PRINCIPE DU VOL D'UN HÉLICOPTÈRE : [16] [17]

I.2.1 Les forces agissantes sur l'hélicoptère en vol

Nous supposons, pour simplifier la présentation du vol de l'hélicoptère, que les pales du rotor principal décrivent un disque dont l'axe passe par le centre de gravité de l'hélicoptère.

En vol, l'hélicoptère est soumis à trois forces [figure I.1] :

- Le poids, appliqué au centre de gravité.
- La force générée par le rotor (portance).

- La traînée générale de l'appareil provoquée en vol de translation, par résistance de l'air sur la structure.

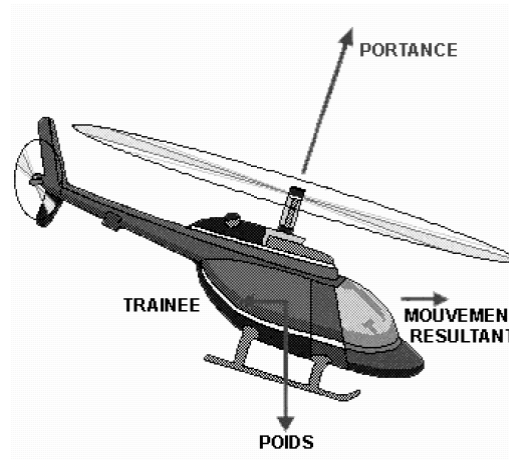


Figure I.1 : forces agissantes sur l'hélicoptère

Pour que l'hélicoptère soit en équilibre, il faut que la résultante du poids et de la traînée soit égale et opposée à la force générée par le rotor. La résultante des forces est alors nulle et le vol est stabilisée : si l'hélicoptère est immobile il le demeure, s'il est soumis à une translation, son mouvement est rectiligne. Le mouvement de l'hélicoptère résulte des variations en intensité et en direction de la portance.

1.2.2 Principe du vol de l'hélicoptère

Le principe du vol est à la fois simple et très compliqué, tout dépend du niveau de sophistication de l'explication. Regardons dans les grandes directions le principe du vol de l'hélicoptère est le suivant :

L'hélice de l'avion assure une traction, elle tire l'avion en se vissant dans l'air comme une vis dans le bois, lorsqu'au décollage l'avion va assez vite son aile génère une portance qui lui permet de voler. [Figure I.2]

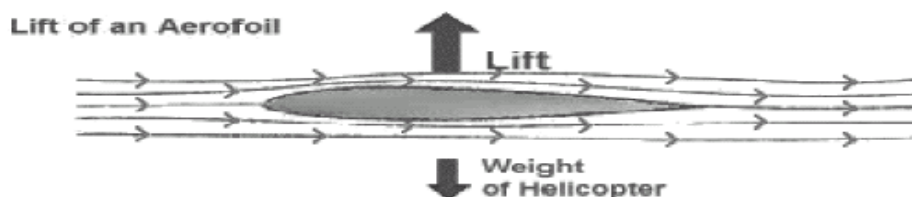


Figure I.2 : principe du vol de l'hélicoptère

Le rotor de l'hélicoptère est à la fois l'aile et l'hélice, il assure traction et portance en tournant d'où son appellation rotor.

1.3 DESCRIPTION DU SIMULATEUR : [18]

Le TRMS (Twin Rotor MIMO System) est un système physique aérodynamique conçu pour le développement et l'implémentation de nouvelles lois de commandes. Ceci inclut, la modélisation de la dynamique du système, l'identification, l'analyse et la conception de divers contrôleurs par des méthodes classiques et modernes. Son comportement ressemble à celui d'un hélicoptère, de point de vue de commande ; il présente un système non linéaire d'ordre supérieur avec des couplages significatifs.

Le TRMS est formé de [figure 1.3] :

- une poutre qui peut pivoter sur sa base de telle manière qu'elle puisse tourner librement dans le plan vertical et horizontal.
- Deux propulseurs (principal et secondaire) fixées aux deux extrémités de la poutre, ils sont formés d'une hélice, un moteur à courant continu ainsi qu'un bouclier pour des raisons de sécurité
- Un contrepoids fixé sur la tige à son pivot, son rôle est de diminuer les vibrations (oscillations) de la poutre
- Une tour pour maintenir la poutre
- Une base comprenant des circuits électroniques pour l'adaptation, synchronisation et filtrage des signaux entrants et sortants
- Un boîtier de marche/arrêt des moteurs

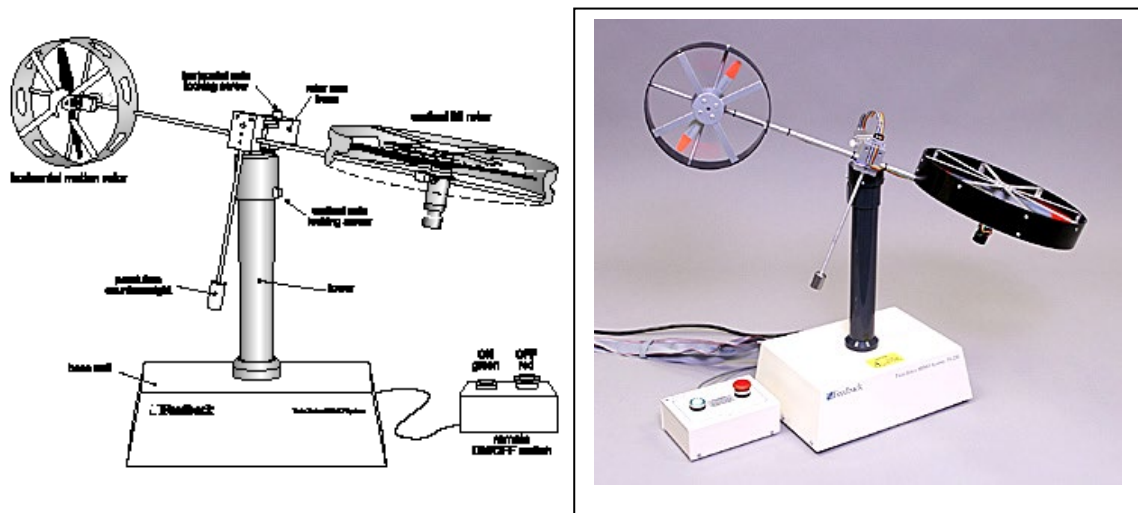


Figure I.3 : le simulateur d'hélicoptère TRMS
(Twin Rotor Mimo System)

Le modèle du TRMS est multi-variable à deux entrées et deux sorties (MIMO 2*2). Les sorties étant l'angle d'élévation et l'angle d'azimut, les entrées étant les tensions appliquées aux deux moteurs. L'articulation sphérique permet à la poutre de pivoter simultanément dans le plan horizontal et vertical, c'est un système à deux degrés de liberté, mais on peut restreindre le mouvement à un degré de liberté par de 2 vis. [Figure I.4]

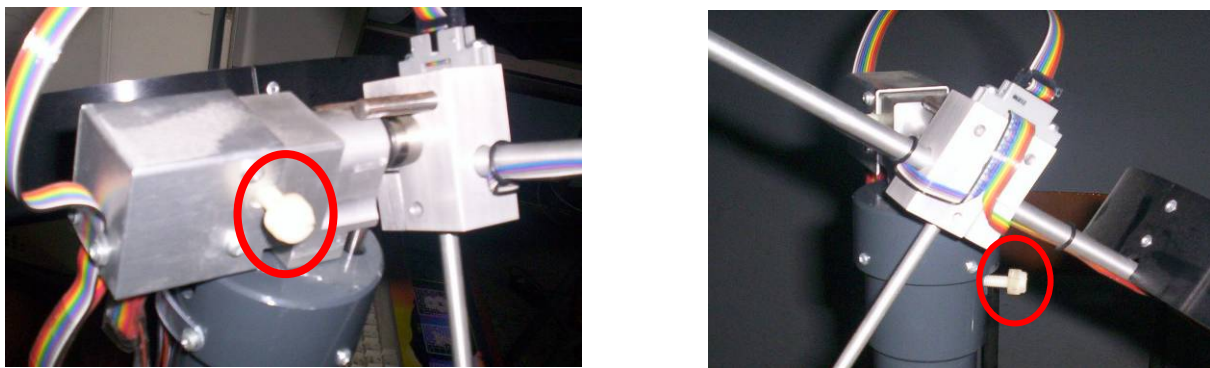


Figure I.4 : les deux vis de fixation

L'objectif de commande est de faire pivoter la poutre autour des deux axes rapidement et avec précision, et cela en variant les forces aérodynamiques des deux propulseurs. Dans un hélicoptère normal la force aérodynamique est commandée en changeant l'angle d'attaque. Néanmoins dans le TRMS cet angle est fixé, et les forces aérodynamiques sont commandées en changeant la vitesse des propulseurs.

1.3.1 Propulseurs

Le TRMS possède deux propulseurs, un principal qui agit dans le plan vertical, et l'autre secondaire qui agit dans le plan horizontal. Chaque propulseur est composé d'une hélice. Ces dernières sont entraînées par des moteurs à courant continu (marque MAXON), couplés avec des tachymètres. La tension de commande varie entre $\pm 2.5V$, le signe de la tension indique le sens de rotation

1.3.2 Commande des moteurs

Le principe des moteurs électriques à courant continu utilisés est que leurs vitesses de rotation sont proportionnelle à la tension qui leur est appliquée. Le mécanisme utilisé est un hacheur, qui est un convertisseur statique alimenté par une source de tension continue qui produit aux bornes du moteur une tension de valeur moyenne réglable. Cet hacheur est commandé par des trains d'impulsions (MLI) qui sont générés à partir du signal de commande provenant du PC à travers la carte d'acquisition.

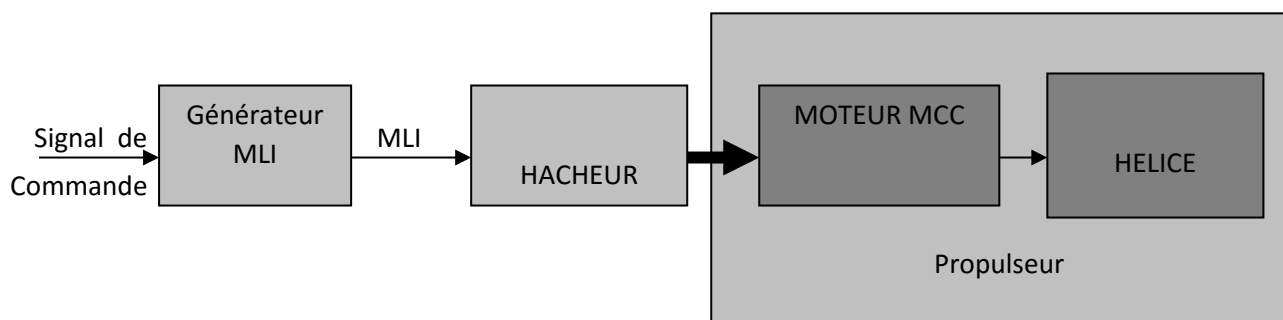


Figure I.5 : schéma synoptique du fonctionnement du propulseur

1.3.3 Les encodeurs optiques

Le TRMS a deux capteurs pour mesurer l'orientation de la poutre dans l'espace, ils se situent à l'intérieur du pivot sphérique de la poutre. Il s'agit de deux encodeurs optiques incrémentaux : le principe de fonctionnement est le suivant [figure I.6]:

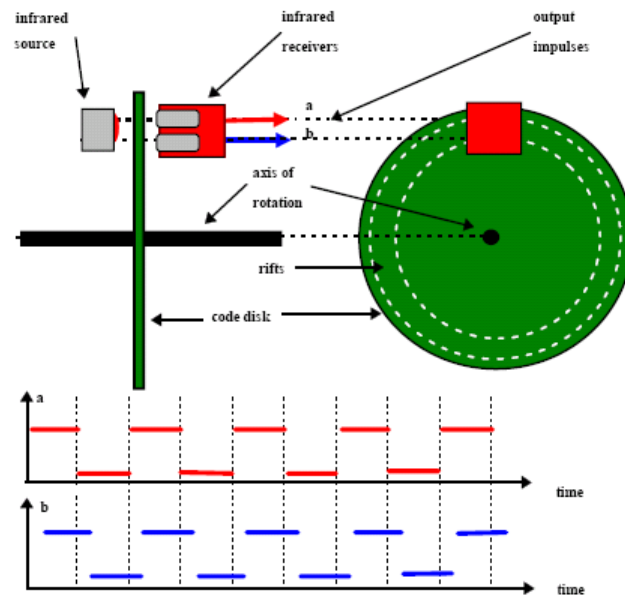


Figure I.6: encodeurs optiques

Les faisceaux lumineux émis par les deux photo diodes (A et B) passent par deux anneaux de fentes sur le disque. Les fentes ont une différence de phase, de sorte que la tension des récepteurs (A et B) soient des ondes rectangulaires avec une différence de phase. Le signe de la différence de phase permet de déterminer la direction de la rotation

I.3.4 La base du TRMS

À l'arrière de la base on trouve les différentes connectiques **figure I.7:**

- **Connecteur ON / Off :** fiche DIN pour connecter la boîte de l'intercepteur ON/OFF des moteurs
- **Sélecteur 110/220v**
- **Fiche d'alimentation 110/220 V**
- **Connecteur CN1 :** il se branche avec le PL1 de la boîte d'adaptation, avec une large nappe 40 pistes. Les signaux sont ceux des deux capteurs de positions, c'est des signaux digitaux sur 16 bits,
- **Connecteur CN2 :** il se branche avec PL3, avec une nappe 20 pistes. Les signaux sont les tensions de commandes des deux moteurs
- **Connecteur CN3 ;** il se branche avec PL2, avec une nappe 20 pistes. Les signaux sont ceux des tachymètres

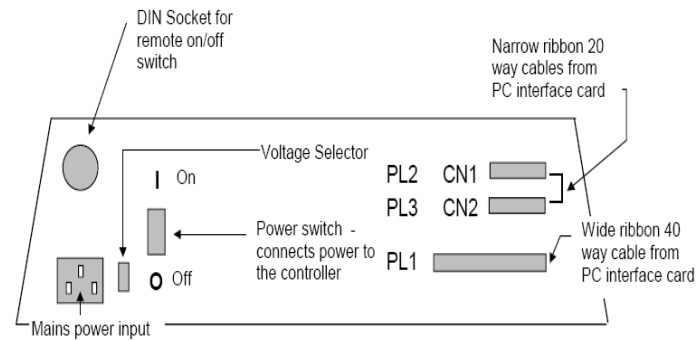
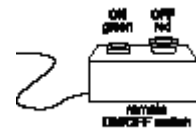


Figure I.7 : la base du TRMS

1.3.5 Boîtier marche/arrêt

C'est un petit boîtier avec deux boutons poussoirs pour, l'un vert pour fermer le circuit d'amplification des moteurs, et l'autre rouge pour l'ouvrir et couper l'alimentation, son rôle principal est de protéger les moteurs en cas de mauvaise manœuvre par l'utilisateur. Il est connecté à l'arrière de la base du TRMS



1.3.6 Le (SCSI BOX)

La (SCSI BOX) figure I.8.b a comme rôle d'adapter les signaux entre la carte d'acquisition et les différents connecteurs de la base du TRMS, et aussi séparer ces différents signaux en fonction de leurs types. La boîte dispose de quatre connecteurs :

- Un connecteur 68 pin pour connecter le câble global 68 pin SCSI de la carte d'acquisition
- Un grand connecteur PL1 pour les entrées digitales des capteurs de positions (deux encodeurs optiques)
- Un petit connecteur PL2 pour les sorties analogiques (deux tensions de commandes des moteurs)
- Un autre petit connecteur PL3 pour les entrées analogiques (retour tachymétrique des deux moteurs)

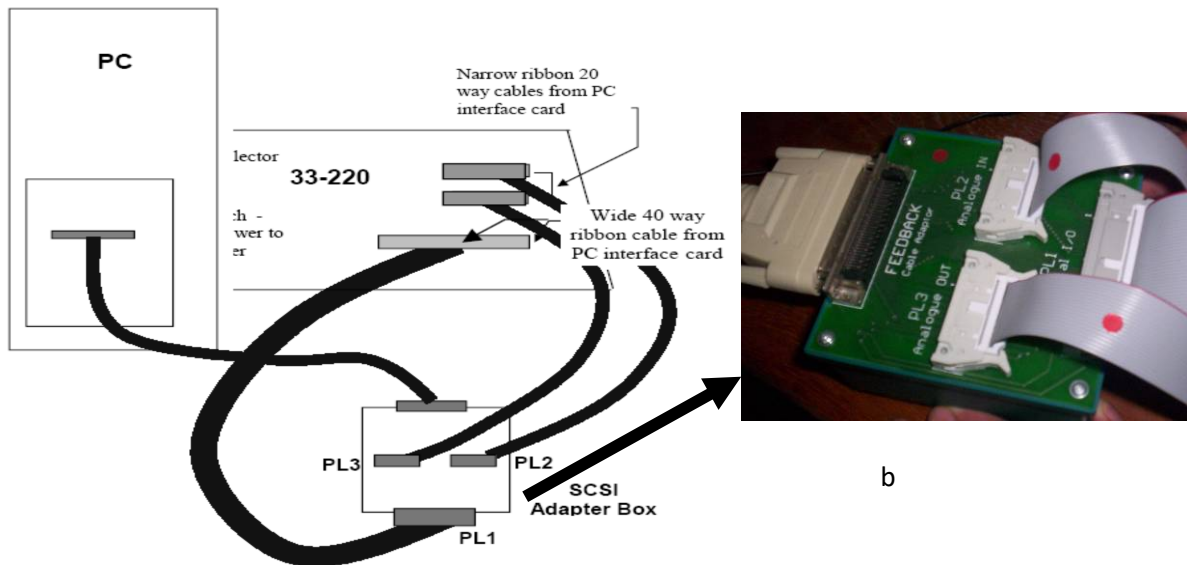


Figure I.8 : branchement des différents connecteurs

I.3.7 Carte d'acquisition : ADVANTECH PCI 1711

C'est une carte d'acquisition universelle qui s'installe sur le port PCI du PC de commande et dispose de connecteurs extérieurs pour des entrées/sorties analogiques et digitaux. Ces principales caractéristiques sont :

- La fonction Plug & Play
- 16 entrées analogiques configurables 16 simples (single-ended) ou source flottante
- Convertisseurs A/D industriels normalisés à approximations successives
- 12 bits utilisés pour la conversion des entrées analogiques
- La fréquence maximale d'échantillonnage est de 100KHz.
- Gamme des entées analogiques est programmables et contrôlable par software *
- Chaque canal à sa gamme individuelle stockée dans la RAM de la carte
- 2 sorties analogiques (convertisseur D/A)
- 16 canaux d'entées digitales
- 16 canaux de sorties digitales
- Un compteur/timer programmable
- Scanne automatique des gains/canaux.

Son principal rôle dans notre application, c'est qu'elle permet la commande digitale d'un système continu à travers un PC, en convertissant les signaux analogiques et numériques et vice-versa. Dans notre cas, on utilise :

- Deux sorties analogiques (convertisseur D/A) pour délivrer les tensions de commandes de références
- Les entrées digitales des deux encodeurs optiques pour mesurer l'angle d'élévation et d'azimut de la poutre
- Deux entrées analogiques (convertisseur A/D) pour récupérer le retour tachymétrique des deux moteurs

1.4 ENVIRONNEMENT DE DEVELOPPEMENT : [19] [20]

Ce banc d'essai utilise une approche novatrice d'implémentation, qui jouit d'une flexibilité accrue et d'une architecture totalement ouverte, cette approche est appelée HIL pour « **Hardware-Int he-Loop** ». Ou bien « **Rapid Prototyping** »

Cette approche élimine l'étape entre la synthèse du régulateur en simulation, et son implémentation réelle sur un système embarqué spécialisé. Cette dernière cause des pertes de temps considérables, en testant et en calibrant à nouveau le régulateur implémenté, voici ci-dessous, un petit comparatif entre cette approche et l'approche classique :

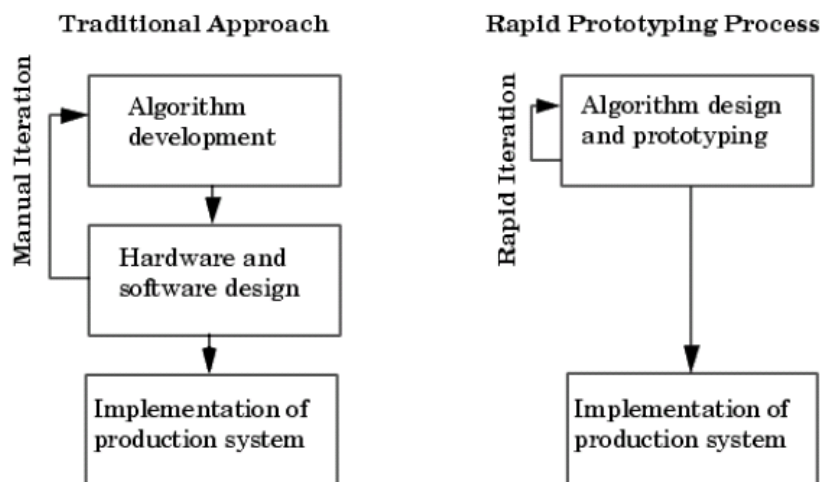


Figure I.9 : rapide prototypage VS processus de développement traditionnel

Avec le « hardware-in-the-loop », on teste directement le contrôleur qu'on a synthétisé en simulation sur le système physique réel, avec un simple PC de développement. En effet tous ce qu'on a besoin est :

- Un PC standard (pour le développement et pour la commande)
- Une carte d'acquisition pour la conversion A/D et D/A
- Le système physique à commander
- Les logiciels nécessaires

Les logiciels utilisés sont :

- **MATLAB** : il joue le rôle d'une plateforme, où tous les autres composants s'exécutent, c'est l'environnement le plus utilisé dans le monde par les ingénieurs, il fournit plusieurs routines sophistiquées de calculs numériques.
- **SIMULINK** : C'est un langage de haut niveau graphique, avec lequel on programme avec des objets ou blocks, ces objets peuvent être soit des blocks standards de bien des blocks spécialisées écrit par l'utilisateur comme des S-fonctions.
- **REAL TIME WORKSHOP** : c'est le programme le plus important, il génère automatiquement du code source C++ optimisé, à partir du modèle simulink. De plus, ce code est paramétrable pour plusieurs targets (cible) d'implémentation. Dans notre cas le target est un processeur Pentium sous WINDOWS comme système d'exploitation
- **Compilateur C++** : il compile le code généré par real time workshop et fait l'édition des liens pour générer un exécutable qui communique avec la cible d'implémentation (target), dans notre cas, le compilateur est VISUAL C++ 6.0 PRO
- **REAL TIME WINDOWS TARGET** : Dans le cas d'un environnement WINDOWS, ce petit kernel (noyau), est essentiel, son rôle est primordial, il assure le temps réel pour notre régulateur. Car WINDOWS est un OS événementiel donc il n'est pas temps réel. Il s'exécute comme un service résident dans le niveau zéro (mode noyau) et intercepte les interruptions matérielles avant WINDOWS, ensuite, Il communique avec l'exécutable du régulateur et interface avec le système physique à travers la carte

d'acquisition. Il contrôle donc le flux de données et de signaux du modèle (maintenant exécutable) vers le système physique et vice versa

- On a besoin aussi, des bibliothèques, ou drivers de la carte d'acquisition, pour les intégrer dans notre programme

➤ Voici ci-dessus un schéma [figure I.10] qui illustre la relation qui existe entre les différents modules :

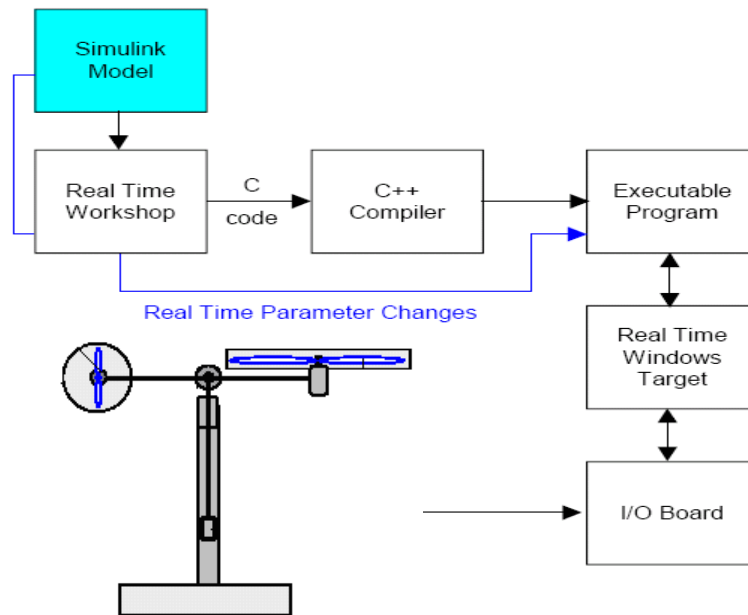


Figure I.10 : processus de génération du code exécutable

➤ Et voici un autre schéma [figure I.11] qui illustre les différentes cibles (targets) et les différentes manières dont on peut implémenter notre régulateur

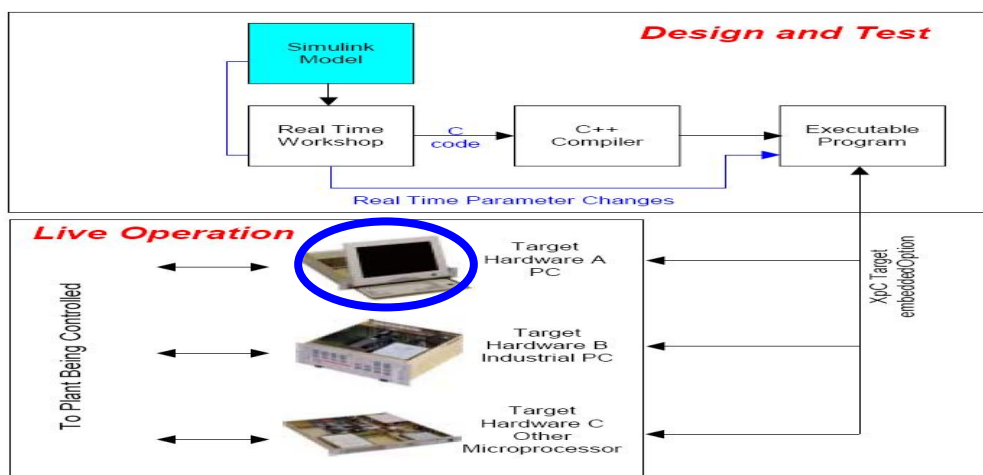


Figure I.11 : différentes cibles d'implémentation

Pour passer d'un régulateur simulé (non temps réel) vers un programme temps réel, il suffit juste d'ajouter dans le modèle de simulation sous SIMULINK, les différents blocs des entrées sorties de la carte d'acquisition (ANALOG INPUT, ANALOG OUTPUT, ENCODER CHANNEL ...etc.)

Pour notre carte d'acquisition voici la librairie utilisée :

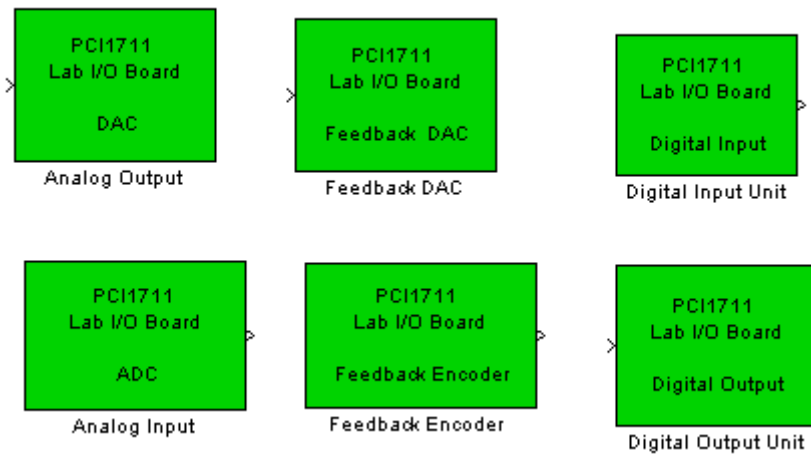


Figure I.12 : librairie de la carte d'acquisition

➤ Et voici ci-dessous un exemple complet de commande en temps réel avec MATLAB :

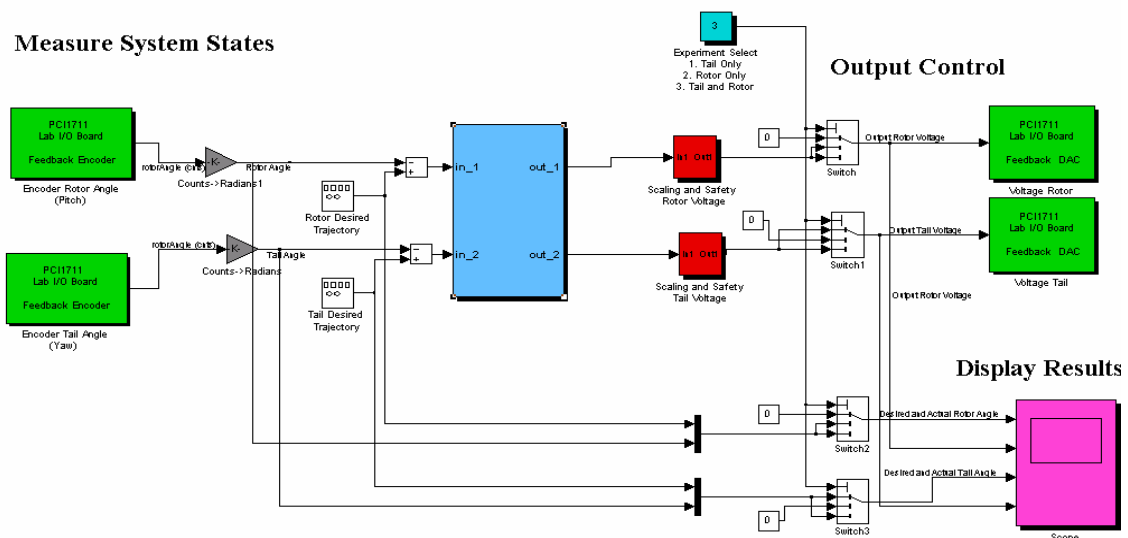


Figure I.13 : exemple d'implémentation d'un contrôleur

En vert : blocs de communications avec la carte d'acquisition, à gauche les encodeurs et à droite des sorties analogiques (tension de commande)

En rouge : blocs de protection des moteurs, car la tension max est de ± 2.5 V

En gris : blocs de conversion, car les encodeurs incrémentaux donnent le nombre d'impulsions qu'il faut convertir en radian

En bleu : est la partie essentielle qui est le bloc de notre régulateur. Ce bloc peut être écrit par n'importe quel langage, on peut même importer un régulateur déjà écrit et l'implémenté, grâce à l'architecture ouverte de cette approche.

Après compilation du programme, il sera indépendant de MATLAB et SIMULINK, néanmoins, on peut utiliser juste SIMULINK comme interface homme machine, pour visualiser et récupérer les signaux.

I.5 CONCLUSION:

Dans ce chapitre, nous avons d'abord donné le principe du vol d'un hélicoptère réel. Ensuite nous avons présenté une brève description du simulateur (TRMS). Ainsi que son principe de fonctionnement. Enfin, nous avons expliqué l'environnement de développement utilisé.

CHAPITRE II

Modélisation du simulateur de vol d'hélicoptère

II . Modélisation du simulateur de vol d'hélicoptère

II.1 INTRODUCTION

Pour effectuer l'analyse et la synthèse dynamique, il est nécessaire de connaître les relations entre les grandeurs d'entrées et de sorties. L'ensemble de ces relations constitue le modèle mathématique du système. L'efficacité de ce modèle repose sur une analogie entre le comportement des objets physiques et celui des formules mathématiques.

Dans l'automatique, modéliser un système consiste à établir un ensemble de relations mathématiques qui permettent de décrire avec précision suffisante les interactions entre le système et son environnement. Lorsque les relations citées auparavant sont issues des équations de la physique, le modèle obtenu est dit modèle de connaissance, ces relations découlent des observations disponibles sur le système. On aboutit aussi au modèle de représentation, en passant par l'identification.

Dans le présent chapitre, nous abordons en premier la modélisation dynamique du système déjà présenté dans le chapitre I. Ensuite, nous validerons le modèle développé, et en fin nous présenterons des simulations en boucle ouverte du modèle.

II.2 Modèle dynamique

Le modèle physique est développé sous certaines hypothèses simplificatrices. En premier lieu, on suppose que les dynamiques du sous-système rotor sont représentées par des équations différentielles du premier ordre. De plus, on suppose que les frottements sont de type visqueux, et que le sous-système hélice-air peut être décrit par les lois d'écoulement aérodynamiques [21] [22]

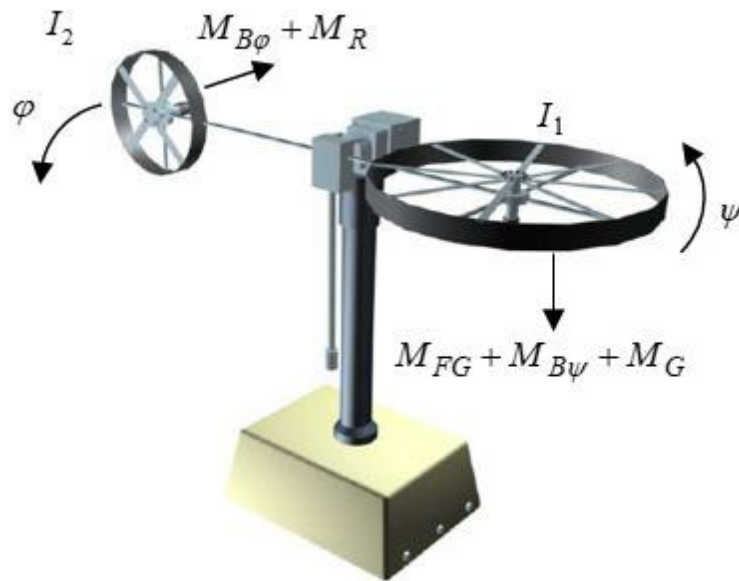


Figure II.1 : Configuration du TRMS 33-949S.

Les deux rotors sont commandés par des moteurs électriques à vitesse variable permettant au système de tourner dans un plan vertical et horizontal (tangage et lacet). La configuration du TRMS est illustrée dans la figure II.1.

II.2.1 Sous système d'élévation

D'abord, considérons la rotation de la poutre dans le plan vertical ; c'est-à-dire autour de l'axe horizontal. En appliquant la seconde loi de Newton on obtient :

$$M_v = I_1 \frac{d^2\psi}{dt^2} \quad (II.1)$$

Avec :

$$M_v = M_1 - M_{FG} - M_{B\psi} - M_G \quad (II.2)$$

Les équations de quantité de mouvement suivantes peuvent être dérivées pour le mouvement vertical [23] [24] :

$$I_1 \cdot \ddot{M} = M_1 - M_{FG} - M_{B\psi} - M_G \quad (II.3)$$

Où :

M_v : Somme des moments dans le plan vertical.

I_1 : Somme des moments d'inertie par rapport à l'axe horizontal.

ψ : Angle d'élévation de la tige.

M_{FG} : Moment de la gravitation.

$M_{B\psi}$: Moment de friction.

M_G : Moment de la force centrifuge.

M_1 : Moment de la force aérodynamique.

Les différents moments de l'équation (2.3) sont donnés par :

$$M_1 = a_1 \tau_1^2 + b_1 \cdot \tau_1 \quad (II.4)$$

$$M_{Fg} = M_g \cdot \sin(\Psi) \quad (II.5)$$

$$M_{B\psi} = B_{1\psi} \cdot \dot{\Psi} \quad (II.6)$$

$$M_G = K_{gy} \cdot M_1 \cdot \phi \cdot \cos(\Psi) - K_{gx} \phi^2 \sin(2\Psi) \quad (II.7)$$

Avec :

τ_1 : Couple délivré par le rotor principal (vertical).

a_1 Et b_1 : Sont des paramètres caractérisant le moment de la force aérodynamique M_1 .

M_g : Dynamique de gravité.

$B_{1\psi}$: Paramètre de la fonction de friction dynamique.

K_{gy} : Paramètre de l'inertie gyroscopique.

ϕ : L'angle d'azimut.

Le moteur (M1) et le circuit électrique de commande sont modélisés par une fonction de transfert du premier ordre :

$$\tau_1(s) = \frac{k_1}{T_{11}s + T_{10}} \cdot u_1(s)$$

(II.8)

Où :

K_1 Est le gain du moteur (M1), T_{11} et T_{10} sont des constantes liées au moteur (M1) et u_1 est l'entrée de commande du moteur (M1).

II.2.2 Sous système d'azimut

De la même façon, on peut décrire le mouvement de la tige autour de l'axe vertical. Le mouvement horizontal peut être décrit comme étant un mouvement de rotation d'un solide :

$$M_h = I_2 \frac{d^2\phi}{dt^2} \quad (II.9)$$

Avec :

$$M_h = M_2 - M_{B\varphi} - M_R \quad (II.10)$$

De même, le mouvement horizontal est décrit par :

$$I_2 \cdot \ddot{\varphi} = M_2 - M_{B\varphi} - M_R \quad (II.11)$$

Avec :

$$M_2 = a_2 \tau_2^2 + b_2 \cdot \tau_2 \quad (II.12)$$

$$M_{B\varphi} = B_{1\varphi} \cdot \dot{\varphi} \quad (II.13)$$

Où :

M_h : Somme des moments dans le plan horizontal.

I_2 : Somme des moments d'inertie par rapport à l'axe vertical.

M_2 : Moment de la force aérodynamique.

$M_{B\varphi}$: Moment de friction.

τ_2 : Couple délivré par le rotor secondaire (horizontal).

a_2 Et b_2 : Sont des paramètres caractérisant le moment de la force aérodynamique M_2 .

$B_{1\varphi}$: Paramètre de la fonction de friction dynamique.

M_R : est la dynamique de réaction croisée, elle est approchée par : $M_R(s) = \frac{K_c(T_0s+1)}{T_p s+1} \cdot M_1$

(II.14)

Où :

K_c Est le gain de la réaction de l'élan et T_0 et T_p sont des paramètres de la réaction de l'élan.

Le moteur à courant continu (M2) et le circuit électrique sont modélisés par une fonction de

transfert du premier ordre :

$$\tau_2(s) = \frac{k_2}{T_{21}s+T_{20}} \cdot u_2(s)$$

(II.15)

Avec :

k_2 Est gain du moteur (M2), T_{21} et T_{20} sont des constantes liées au moteur (M2) et u_2 est l'entrée de commande du moteur (M2).

Le modèle dynamique du TRMS 33-949S est donné par :

$$\begin{cases} \ddot{\Psi} = \frac{1}{I_1} \{-M_g \cdot \sin(\Psi) - B_{1\Psi} \cdot \dot{\Psi} + K_{gx} \dot{\phi}^2 \cdot \sin(2\Psi) + (a_1 \tau^2_1 + b_1 \tau_1)(1 - K_{gy} \cdot \dot{\phi} \cos(\Psi))\} \\ \ddot{\phi} = \frac{1}{I_2} \left\{ B_{1\phi} \dot{\phi} - \frac{K_c(T_0 S + 1)}{T_p S + 1} (a_1 \tau^2_1 + b_1 \tau_1) + a_2 \tau^2_2 + b_2 \tau_2 \right\} \end{cases} \quad (II.16)$$

II.3 Modèle d'état

Le modèle couplé après de l'étude en présente :

$$\begin{cases} \frac{d\Psi}{dt} = \dot{\Psi} \\ \frac{d\dot{\Psi}}{dt} = \frac{1}{I_1} \{-M_g \cdot \sin(\Psi) - B_{1\Psi} \cdot \dot{\Psi} + K_{gx} \dot{\phi}^2 \cdot \sin(2\Psi) + (a_1 \tau^2_1 + b_1 \tau_1)(1 - K_{gy} \cdot \dot{\phi} \cos(\Psi))\} \\ \frac{d\tau_1}{dt} = \frac{K_1}{T_{11}} \cdot U_1(t) - \frac{T_{10}}{T_{11}} \\ \frac{d\phi}{dt} = \dot{\phi} \\ \frac{d\dot{\phi}}{dt} = \frac{1}{I_2} \left\{ B_{1\phi} \dot{\phi} - \frac{K_c(T_0 S + 1)}{T_p S + 1} (a_1 \tau^2_1 + b_1 \tau_1) + a_2 \tau^2_2 + b_2 \tau_2 \right\} \\ \frac{d\tau_2}{dt} = \frac{K_2}{T_{21}} \cdot U_2(t) - \frac{T_{20}}{T_{21}} \end{cases} \quad (II.17)$$

II.4 Paramètres du modèle

Pour simuler les modèles précédents, on doit d'abord déterminer les paramètres du modèle, qu'on peut les deviser en trois catégories :

- Paramètres physiques.
- Caractéristiques non linéaires.
- Les constantes de temps et les gains statiques.

Tableau II.1 : paramètres du modèle

Définition	Symbole	Valeur
Moment d'inertie du rotor vertical	I_1	0.068 Kg.m ²
Moment d'inertie du rotor horizontal	I_2	0.02 Kg.m ²
Paramètre caractéristique statique	a_1	0.0135
Paramètre caractéristique statique	b_1	0.0924
Paramètre caractéristique statique	a_2	0.02
Paramètre caractéristique statique	b_2	0.09
Dynamique de gravité	M_g	0.32 Nm
Paramètre de la fonction de friction dynamique	$B_{1\psi}$	0.006 N.m.S/rad
Paramètre de la fonction de friction dynamique	$B_{1\varphi}$	0.1 N.m.S/rad
Paramètre d'inertie gyroscopique	K_{gy}	0.05 S/rad
Paramètre d'inertie gyroscopique	K_{gx}	1
Le gain du Moteur 1	K_1	1.1
Le gain du Moteur 2	K_2	0.8
Constante liée au moteur 1	T_{11}	1.1
Constante liée au moteur 1	T_{10}	1
Constante liée au moteur 2	T_{21}	1
Constante liée au moteur 2	T_{20}	1
Paramètre de la réaction de l'élan	T_p	2
Paramètre de la réaction de l'élan	T_0	3.5
Le gain de la réaction de l'élan	K_c	-0.2

La limite pour le signal de commande est réglée sur [-2.5V ; 2,5 V].

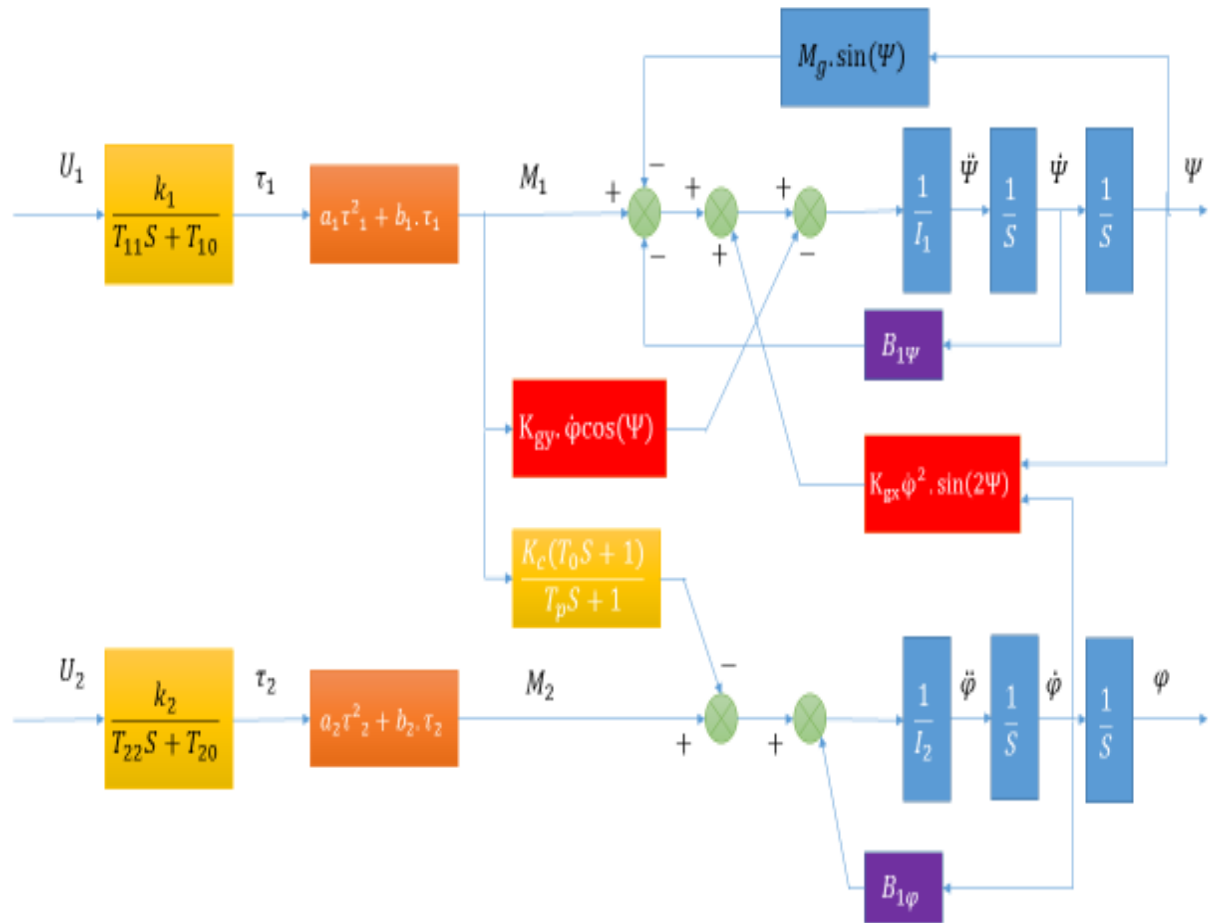


Figure II.2 : Schéma bloc du système TRMS 33-949S [25].

II.5 Simulation

On a : Ψ : alpha V et φ : alpha H

➤ Voici ci-dessous quelques réponses du système en boucle ouverte

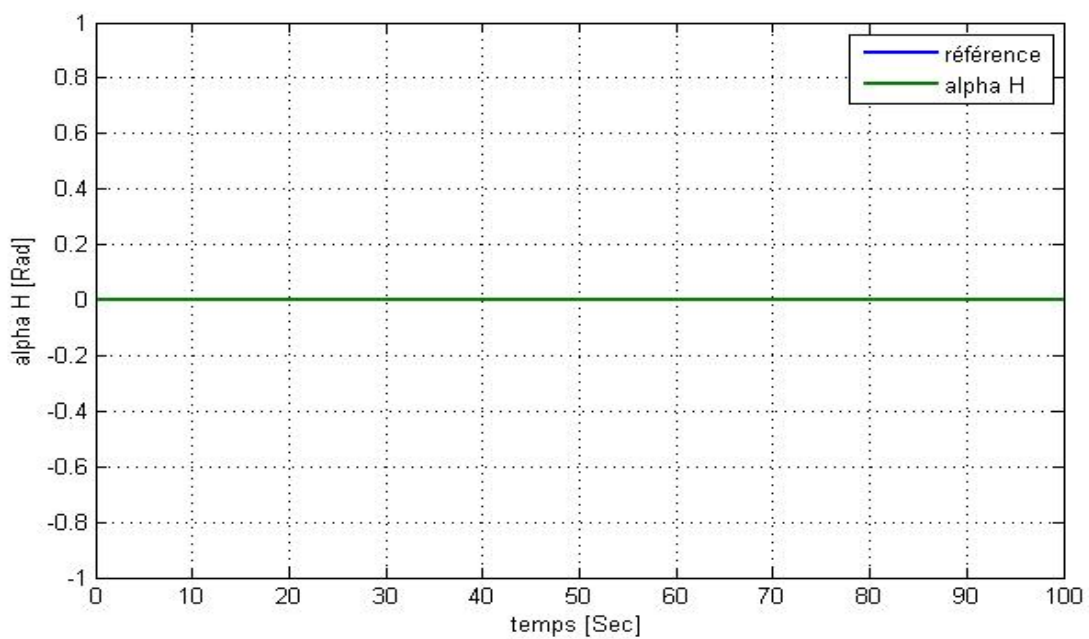
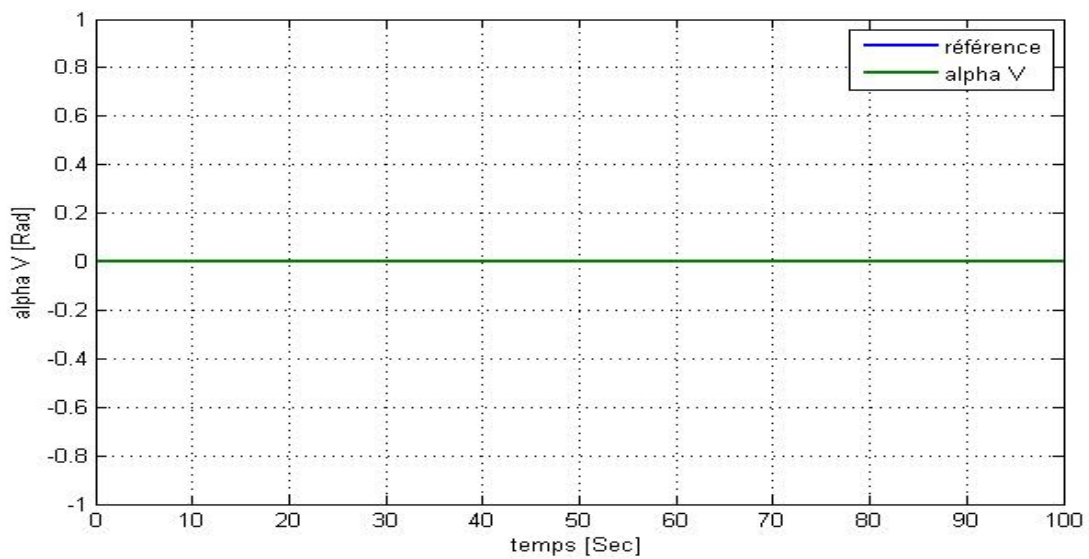


Figure II.3 : Réponse libre du modèle de simulation avec : $u_v = u_h = 0$ et $\Psi_0 = 0$, $\varphi_0 = 0$

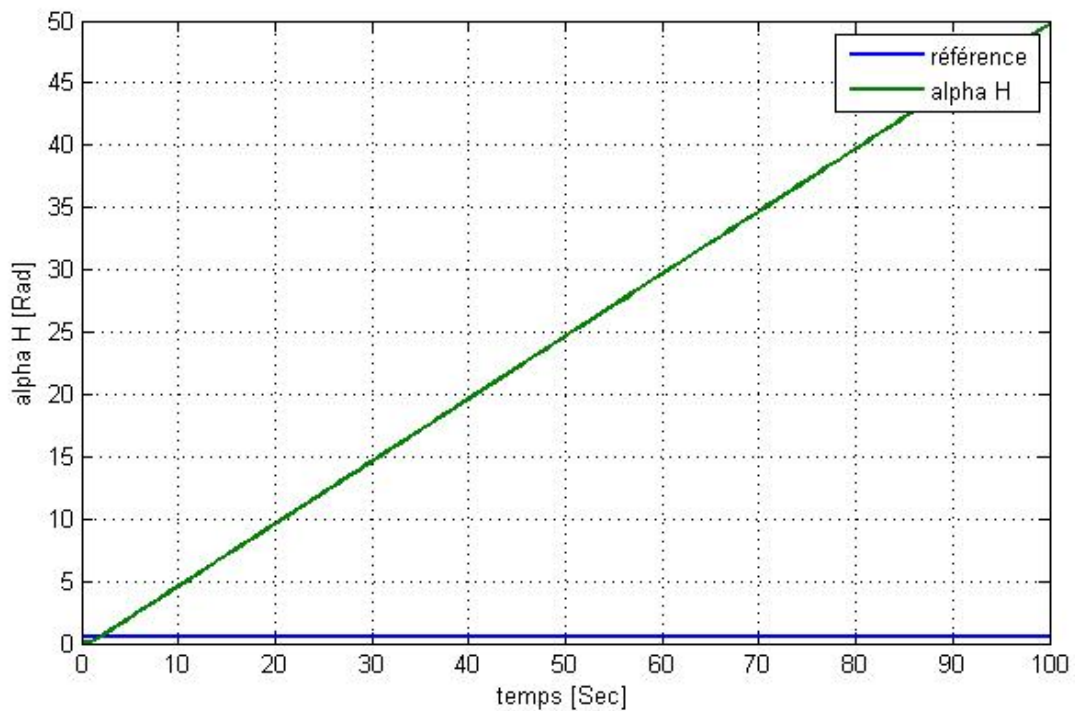
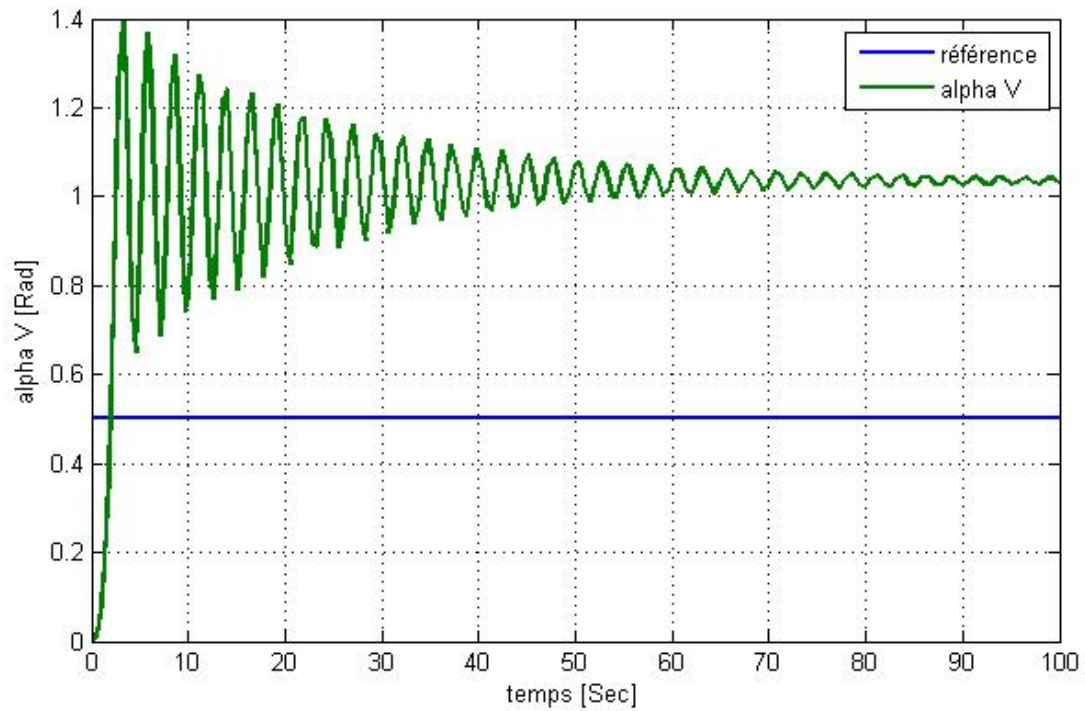


Figure II.4 : Réponse du modèle de simulation pour : $u_v = u_h = 0.5$ et $\Psi_0 = 0$, $\varphi_0 = 0$

Lorsqu’on excite le système, le sous-système vertical tend vers un nouveau point d’équilibre qui dépendra de l’excitation appliquée .Pour une excitation $u_v \in [-2.5, 2.5]$, le

sous-système vertical se comporte comme un système **BIBO**. (Bounded Input Bounded Output).

Le sous-système horizontal a le comportement d’un intégrateur, il a une infinité de points d’équilibres (tout le plan horizontal), et il diverge même pour de petites excitations.

II.6 Validation du modèle de simulation

Pour valider notre modèle de simulation, nous allons comparer les réponses obtenues précédemment avec les réponses du système réel.

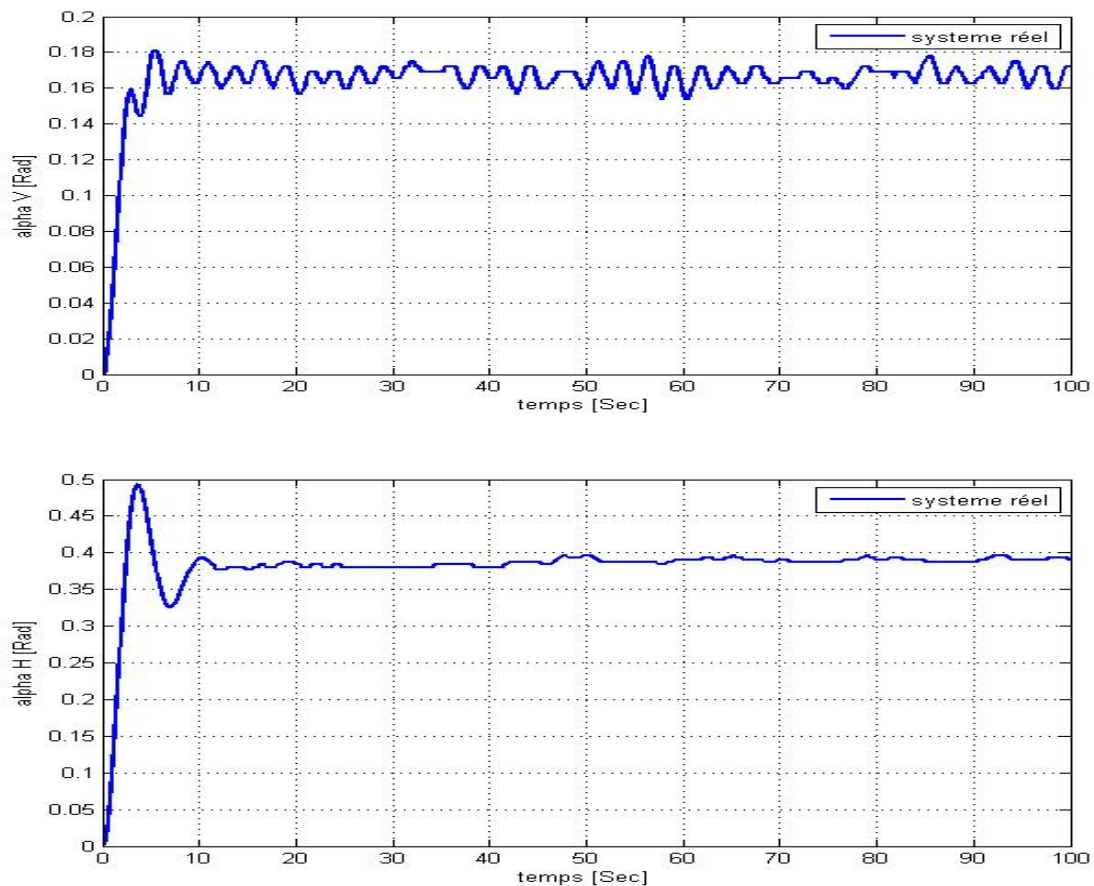


Figure II.5 : Réponse du système réel pour : $u_v = u_h = 0.5$ avec $\Psi_0 = 0$, $\varphi_0 = 0$

On constate une différence entre les deux réponses, qui sera corrigé en précédant par une identification statique des paramètres.

II.7 Identification statique : [21]

- *Modèle vertical :*

Considérons d'abord le modèle vertical :

A l'équilibre nous avons :

$$\begin{cases} \frac{d\Psi}{dt} = \dot{\Psi} = 0 \\ \frac{d\dot{\Psi}}{dt} = \frac{1}{I_1} \{-M_g \cdot \sin(\Psi) - B_{1\Psi} \cdot \dot{\Psi} + K_{gx} \dot{\phi}^2 \cdot \sin(2\Psi) + (a_1 \tau^2 + b_1 \tau)(1 - K_{gy} \cdot \dot{\phi} \cos(\Psi))\} = 0 \\ \frac{d\tau_1}{dt} = \frac{K_1}{T_{11}} \cdot U_1(t) - \frac{T_{10}}{T_{11}} = 0 \end{cases} \quad (II.18)$$

Ce qui donne :

$$\frac{1}{I_1} \{-M_g \cdot \sin(\Psi) - B_{1\Psi} \cdot \dot{\Psi} + K_{gx} \dot{\phi}^2 \cdot \sin(2\Psi) + (a_1 \tau^2 + b_1 \tau)(1 - K_{gy} \cdot \dot{\phi} \cos(\Psi))\} = 0 \quad (II.19)$$

Afin d'atteindre le même régime permanent nous avons ajouté une force de torsion pour déplacer le système au nouveau point d'équilibre.

L'équation devient alors :

$$\frac{1}{I_1} \{-M_g \cdot \sin(\Psi) - B_{1\Psi} \cdot \dot{\Psi} + K_{gx} \dot{\phi}^2 \cdot \sin(2\Psi) + (a_1 \tau^2 + b_1 \tau)(1 - K_{gy} \cdot \dot{\phi} \cos(\Psi))\} + F = 0 \quad (II.20)$$

Avec

$$F = f_v \alpha_{v0} \quad (II.21)$$

Où :

Ψ_0 : Est la position d'équilibre du système réel (dans ce cas $\Psi_0 = -0.5645 \text{ rad}$)

f_v : Est une constante, déterminée à partir des équations (II.20) et (II.21)

$$f_v = \frac{\frac{1}{I_1} \{-M_g \cdot \sin(\Psi) - B_{1\Psi} \cdot \dot{\Psi} + K_{gx} \dot{\phi}^2 \cdot \sin(2\Psi) + (a_1 \tau^2 + b_1 \tau)(1 - K_{gy} \cdot \dot{\phi} \cos(\Psi))\}}{\alpha_{v0}}$$

A.N $f_v = 0.1 \text{ [N.m}^2 \cdot \text{rad}^{-1}]$

• *Modèle horizontal*

De même que pour le modèle vertical, on va adapter le point d'équilibre ainsi que le coefficient de friction.

La force de torsion est de la forme $F_{th} = f_h (\alpha_h - \alpha_{h0})$.

φ_0 : Est la position d'équilibre du système réel. (dans ce cas $\varphi_0 = -0.54 \text{ rad}$)

f_h : Est le coefficient de torsion.

Pour f_h , il est déterminé de la même façon que f_v .

A l'équilibre on a :

$$\frac{1}{I_2} \{ B_1 \phi \dot{\phi} - \frac{K_c(T_0 S + 1)}{T_p S + 1} (a_1 \tau^2_1 + b_1 \tau_1) + a_2 \tau^2_2 + b_2 \tau_2 \} + f_h (\phi_1 - \phi_0) = 0$$

Où :

ϕ_0 : La position d'équilibre du système réel. (dans ce cas $\phi_0 = -0.54 \text{ rad}$).

ϕ_1 : Le nouveaux position d'équilibre ($\phi_1 = -0.35 \text{ rad}$).

Et cela donne : $f_h = 0.1275 \text{ [N.m}^2.\text{rad}^{-1}\text{]}$

II.8 Validation du nouveau modèle de la simulation :

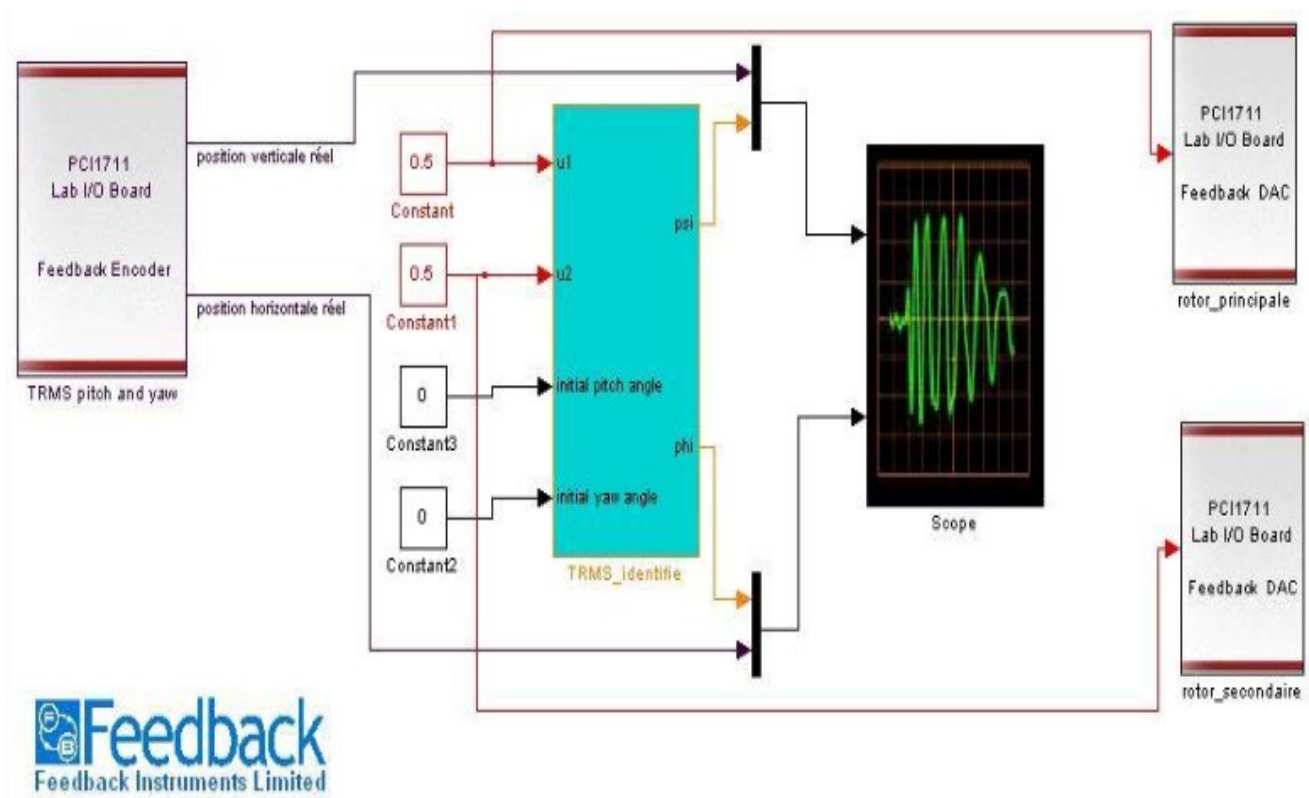


Figure II.6 : schéma bloc de validation du système réel avec modèle de simulation

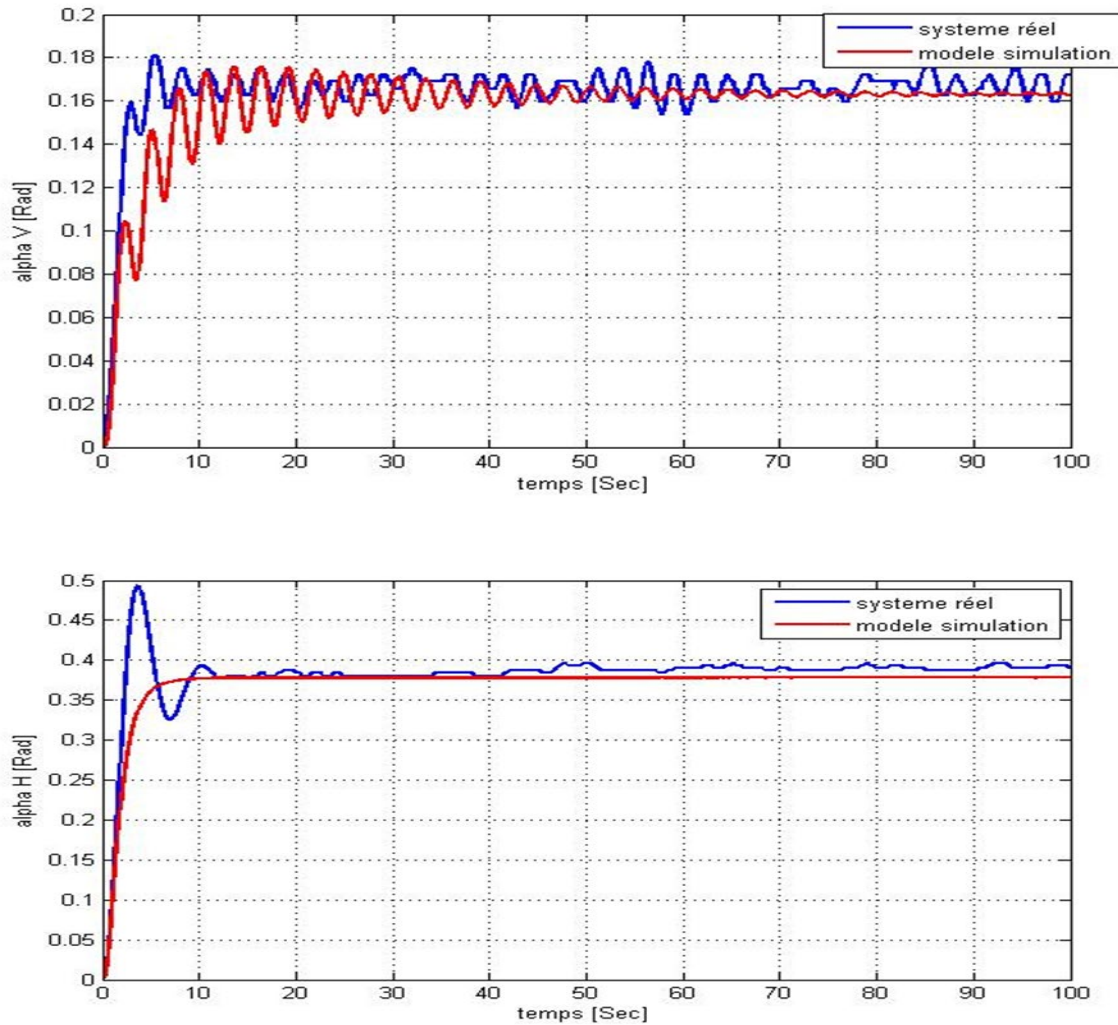


Figure II.7 : Réponse du système pour $u_v = u_h = 0.5$ avec $\Psi_0 = 0$ et $\varphi_0 = 0$

II.9 CONCLUSION

Nous avons procédé par la modélisation analytique pour avoir un modèle mathématique du TRMS, un modèle qui est non linéaire et couplé. Bien que nous avons essayé d'adapter le modèle au système réel, une erreur de modélisation due essentiellement aérodynamique négligées persiste. (Exemple : problème des nappes des capteurs).

CHAPITRE III

Régulateur PID et leurs applications en ligne

III. Régulateur PID et leurs applications en ligne

III.1 Introduction

Ce chapitre est consacré dans un premier temps aux quelques éléments de la théorie et une présentation générale des concepts de base de la régulateur PID, Tout d'abord, nous expliquons brièvement le principe de cette commande, et en fin nous avons valider la commande en temps réelle avec des déférente consigne et faire le test de robustesse de la commande PID avec une consigne échelon.

III.2 la commande par le régulateur type PID

Le régulateur PID (Proportionnel, Intégral, Dérivé) est encore largement utilisé dans le milieu industriel malgré l'émergence d'autres méthodes de régulation. Ce régulateur linéaire est basé sur une structure très simple dont le fonctionnement ne dépend que de trois coefficients, qui sont les gains appliqués sur les signaux proportionnel (K_p), intégral (K_i) et dérivé (K_d). Donc il permet de régler à l'aide de ses trois paramètres les performances (amortissement, temps de réponse) d'une régulation d'un processus modélisé par un deuxième ordre. Nombreux sont les systèmes physiques qui, même en étant complexes, ont un comportement voisin de celui d'un deuxième ordre, dans une certaine échelle de temps. Par conséquent, le régulateur PID est bien adapté à la plupart des processus de type industriel et est relativement robuste par rapport aux variations des paramètres du procédé, quand on n'est pas trop exigeant pour les performances de la boucle fermée par rapport à celles de la boucle ouverte

III.3 Structure des correcteurs PID

En général, le correcteur ne fonctionne pas en action dérivé pure (trop instable). Il fonctionne en correcteur Proportionnel Intégral Dérivé (PID). Le triplet : Bonde Proportionnel, Temps Intégral, Gain Dérivé ; définit trois types de fonctionnement qui sont représentés sur les figures (III.1), (III.2) et (III.3) [26]

a) Structure série : les trois corrections sont branchées en série :

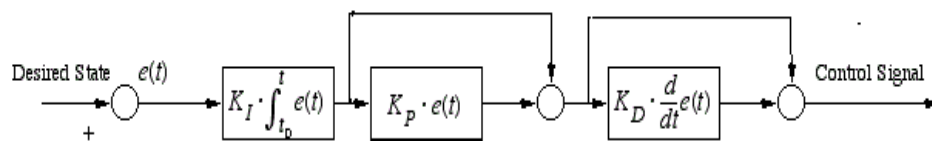


Figure III.1 : Structure série d'un PID

b) Structure parallèle : les deux corrections sont branchées en parallèles :

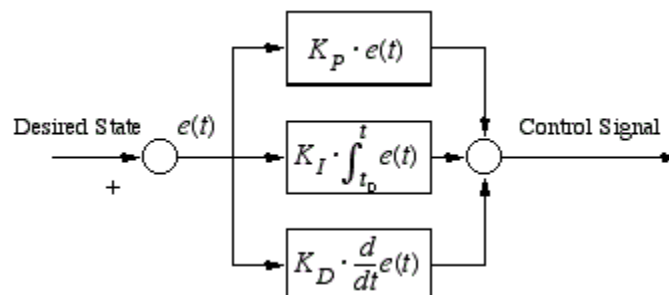


Figure III.2 : Structure parallèle d'un PID

c) Structure mixte :

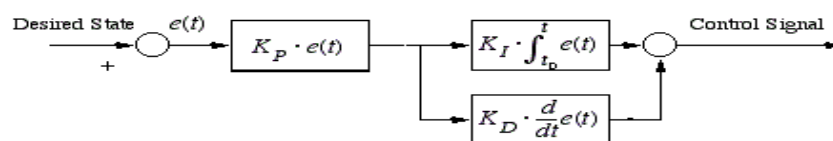


Figure III.3 : Structure mixte d'un PID

III.4 Avantages et inconvénients

Le correcteur PID est le plus utilisé dans l'industrie, mais le réglage de ces paramètres plus difficiles. Dans ce tableau on récapituler les avantages et les inconvénients des correcteurs usuels [27] :

Tableau III-1 : les avantages et les inconvénients régulateur PID

Action	Avantage	Inconvénient
- Proportionnel	- Diminue erreur statique. - Diminue le temps de réponse.	- Argumente dépassement. - Diminue la stabilité.
- Intégral	- Annule l'erreur statique. - Argumente la stabilité.	- Système parfois lent en BF
- Dérivé	- Améliore stabilité et rapidité	- Sensibilité du système aux bruits. - Argumente le temps de réponse.

III.5 Domaine d'application :

C'est le correcteur le plus utilise en industrie à cause de sa simplicité et sans oublier qu'il regroupe

Les trois actions, alors il assure : [28]

L'action proportionnelle (P) :

Est souvent utilisée pour améliorer la rapidité du système bouclé.

L'action intégrale (I) :

Est employée pour améliorer le régime permanent, une précision parfaite, en outre, elle permet de filtrer la variable à régler d'où l'utilité pour le réglage des variables bruitées telles que la pression.

L'action dérive (D) :

assure l'amélioration de la stabilité.

III.6 Résultat de validation de commande PID en système réel

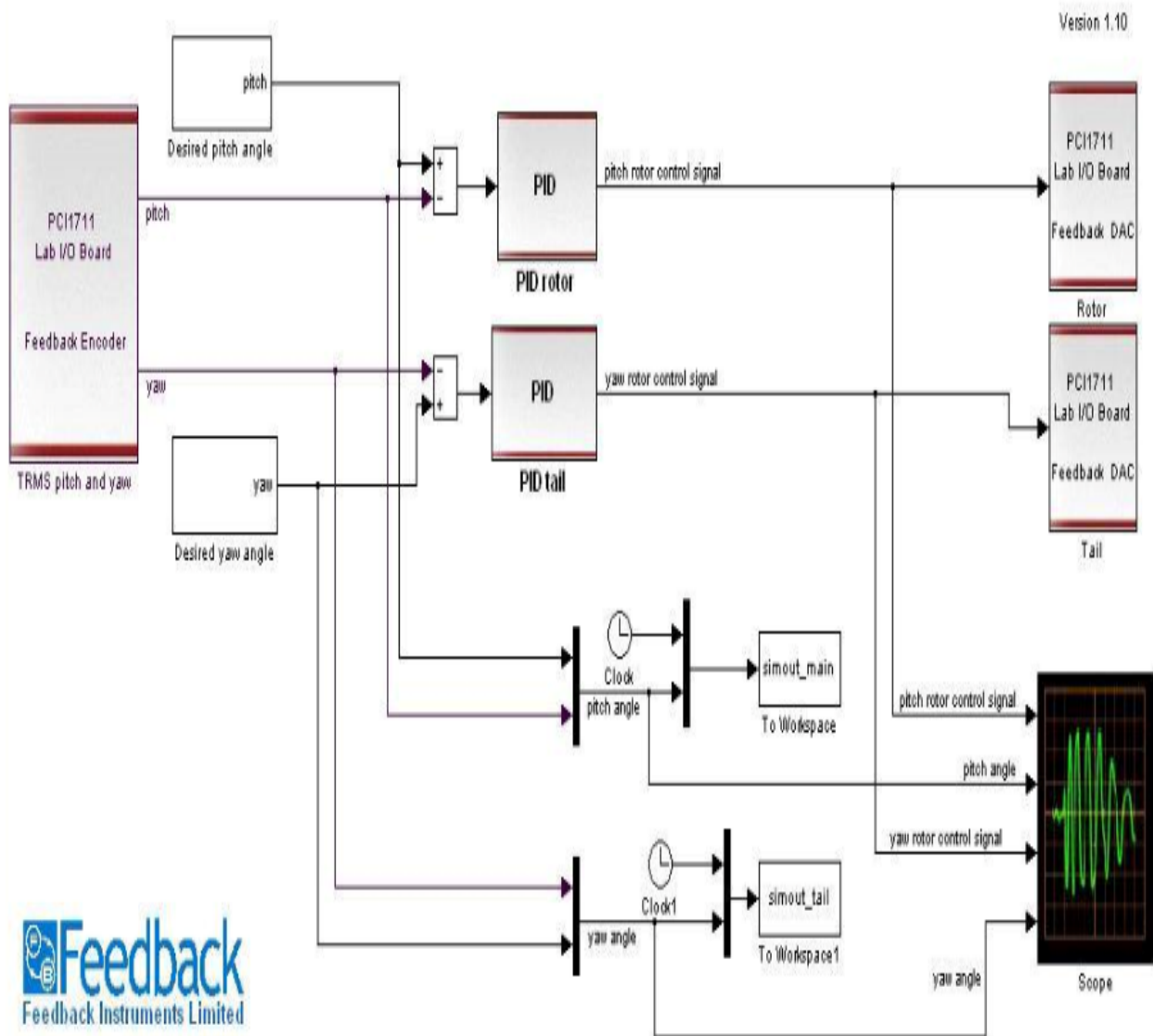


Figure III.4 : la commande PID en simulateur de TRMS

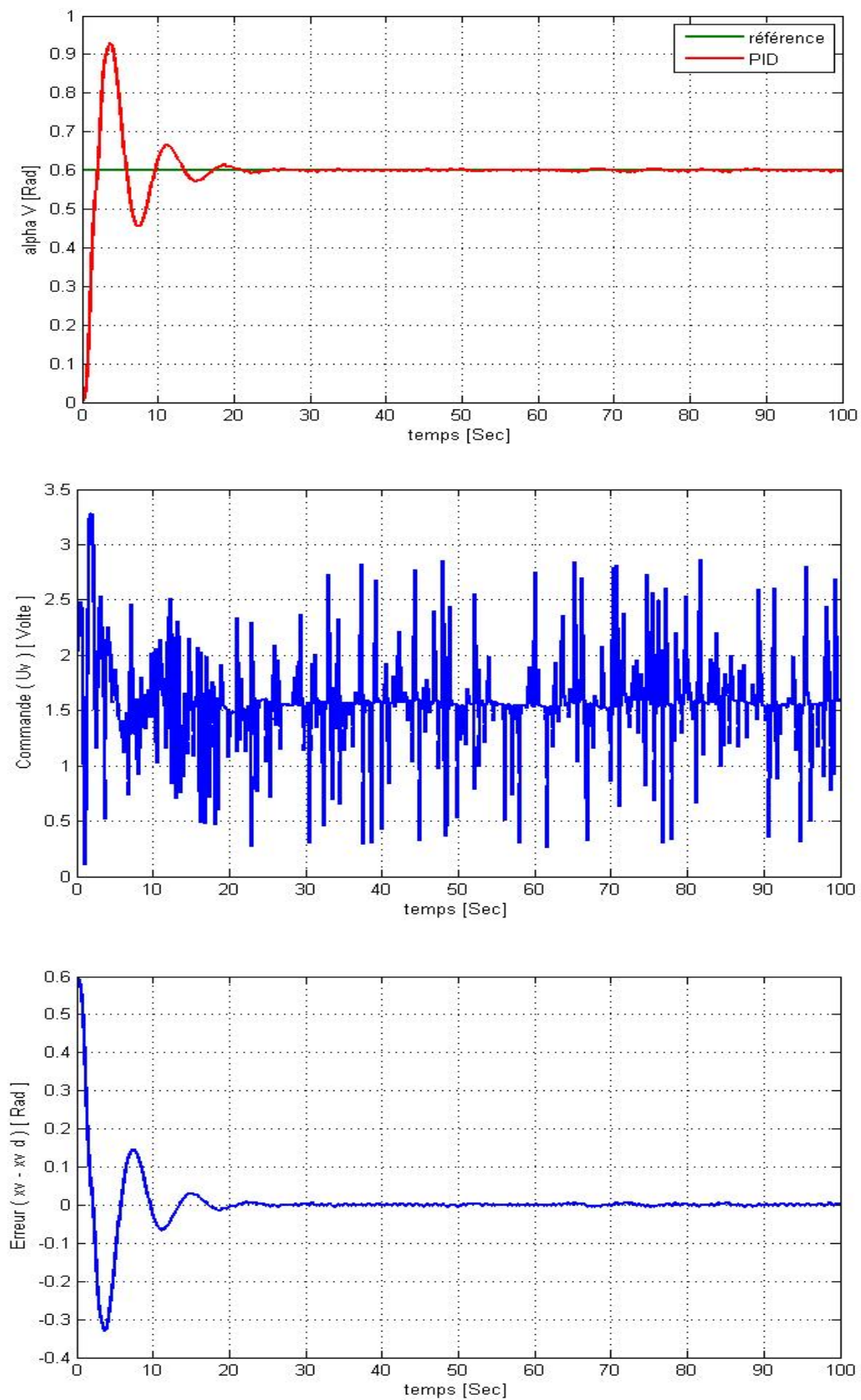


Figure III.5 : Résultats de la commande PID avec une consigne échelon pour sous-système verticale

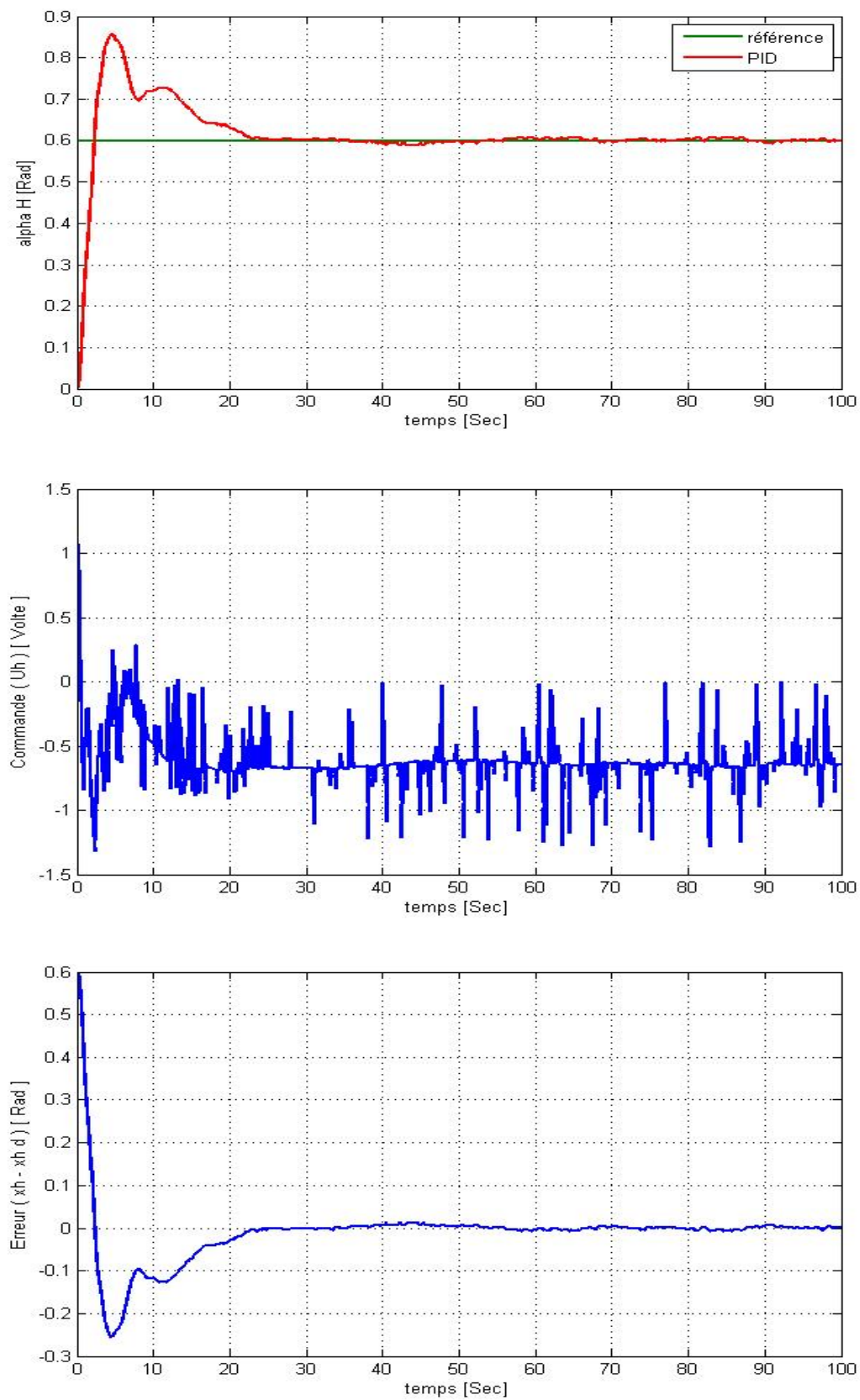


Figure III.6 : Résultats de la commande PID avec une consigne échelon pour sous-système horizontale

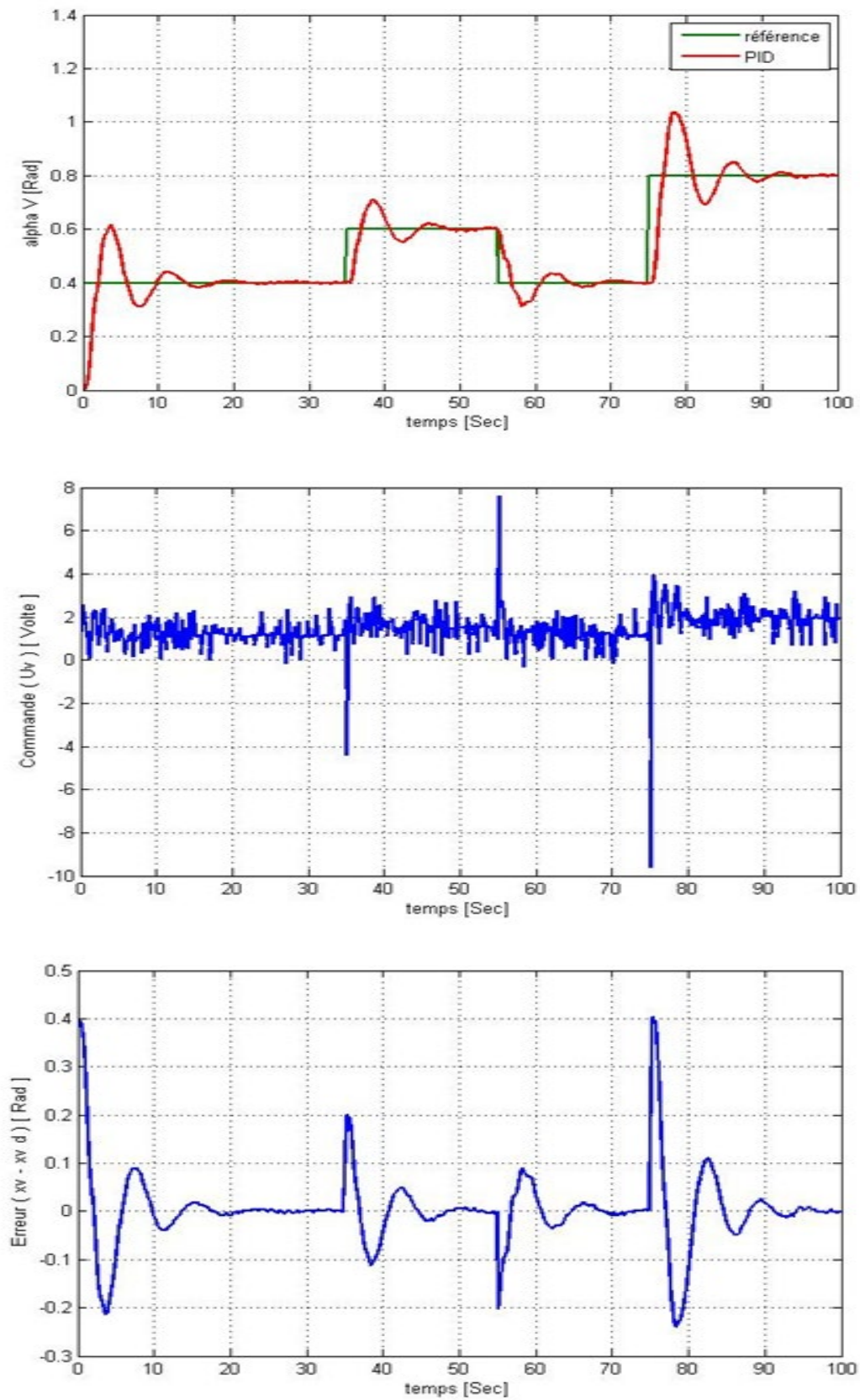


Figure III.7 : Résultats de la commande PID avec une consigne carrée pour sous-système verticale

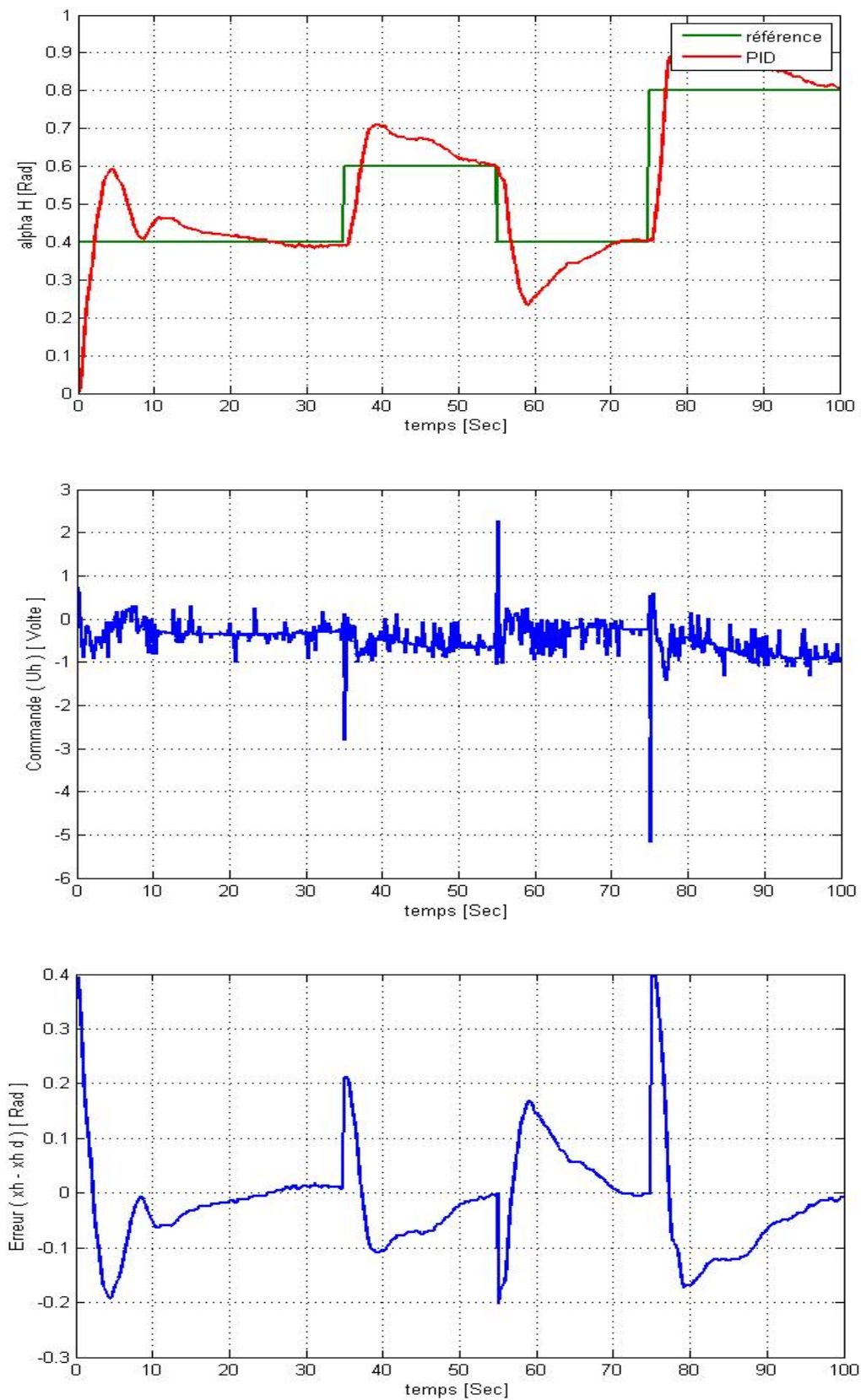


Figure III.8 : Résultats de la commande PID avec une consigne carrée pour sous-système horizontale

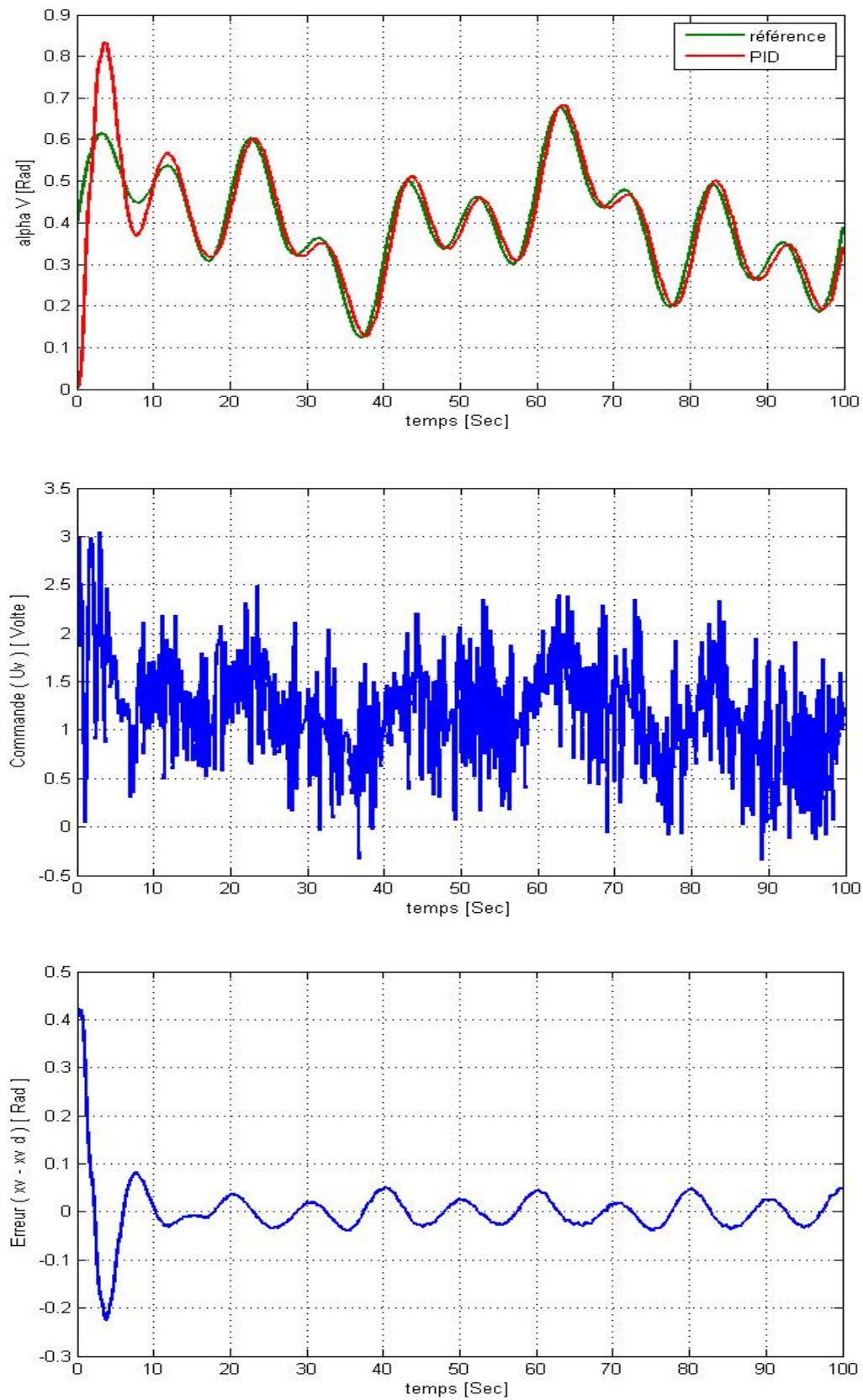


Figure III.9 : Résultats de la commande PID avec une consigne sinusoïdale sous-système verticale

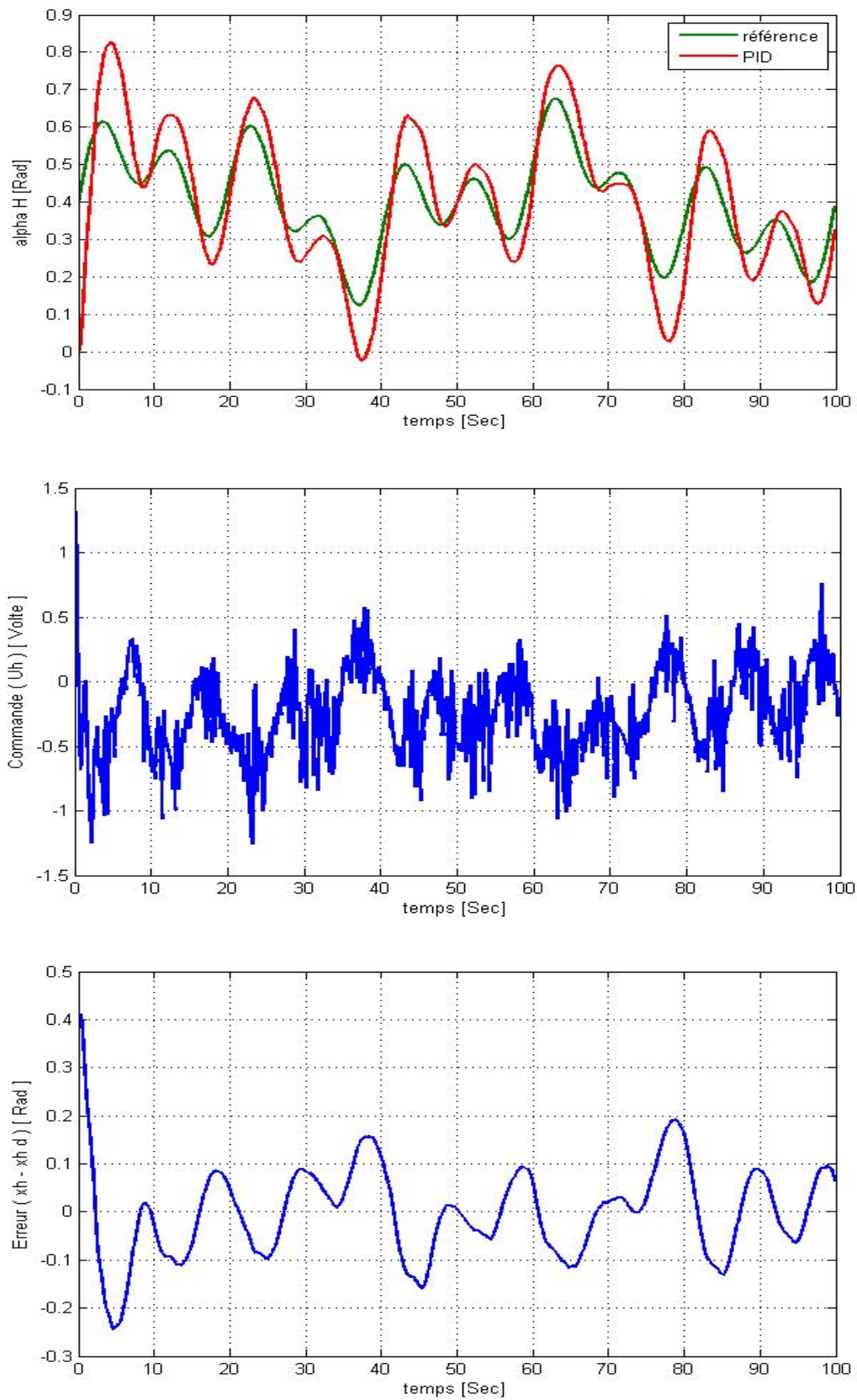


Figure III.10 : Résultats de la commande PID avec une consigne sinusoïdale sous-système horizontale

III.7 Test de robustesse pour les positions angulaires du TRMS

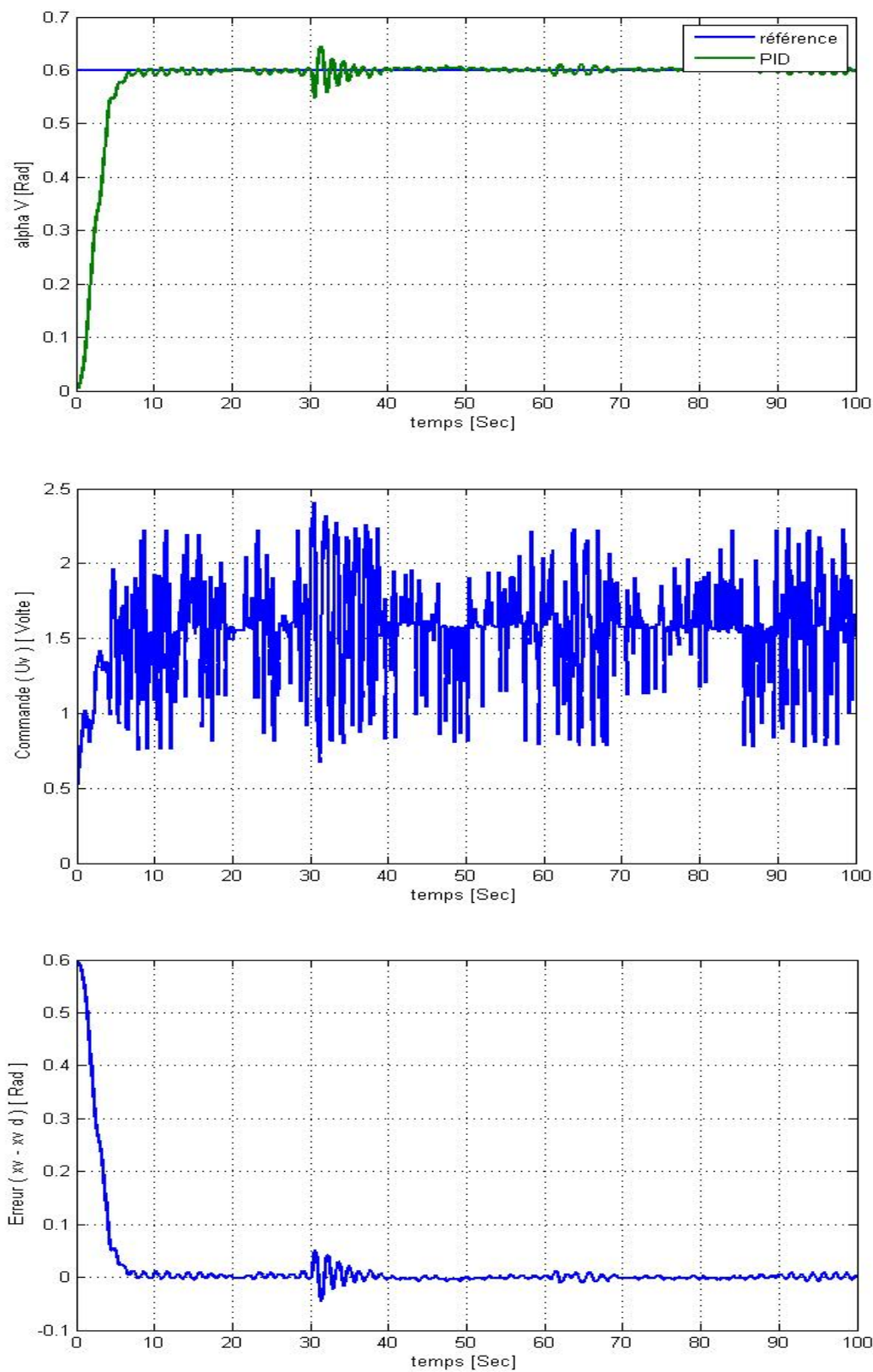


Figure III.11 : Résultats de la commande PID avec une consigne échelon pour sous-système verticale avec perturbation

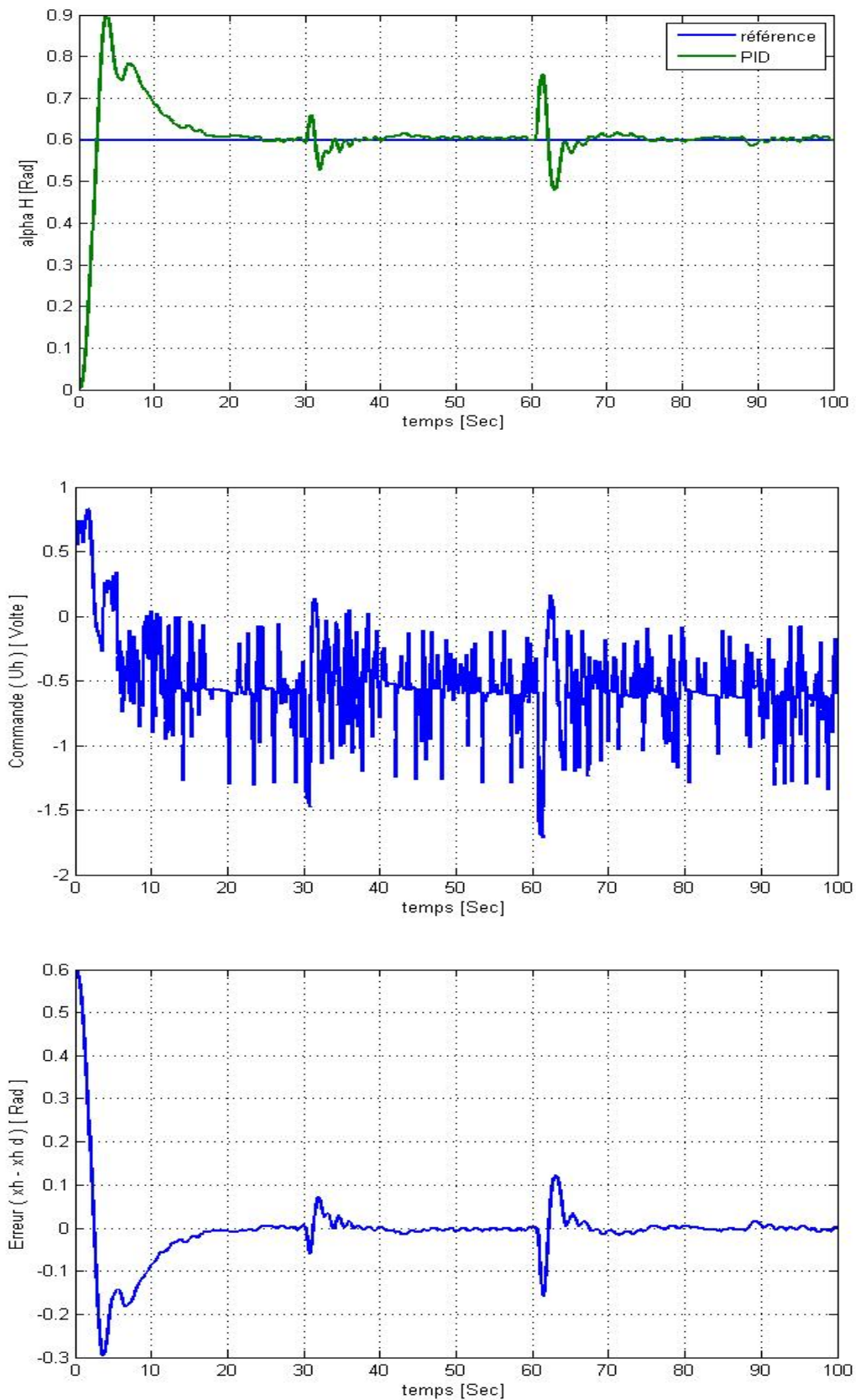


Figure III.12 : Résultats de la commande PID avec une consigne échelon pour sous-système horizontale avec perturbation

III.8 CONCLUSION :

Dans ce chapitre, Nous avons vu la théorie de la commande par le régulateur PID et nous avons faites les résultat de validation de commande PID en système réel avec des différentes consignes, en fin voire le test de robustesse de la commande avec une consigne échelon

CHAPITRE IV

*Commande Intelligente type
floue et leur implémentation en
ligne pour TRMS*

IV. Commande Intelligente type floue et leur implémentation en ligne pour TRMS

IV-1 INTRODUCTION :

La logique floue est une technique pour le traitement de connaissances imprécises basées, sur des termes linguistiques; elle donne les moyens de convertir une commande linguistique basée sur le raisonnement humain, en une commande automatique, permettant ainsi la commande des systèmes complexes dont les informations sont exprimées d'une façon vague et mal définie

Dans ce chapitre nous avons validé la commande en temps réel avec des déférente consigne et faire le test de robustesse de la commande par logique floue avec une consigne échelon

IV-2 historique

C'est à Lotfi Zadeh, spécialiste renommé en automatique et en théorie des systèmes, à l'université de Berkeley, que revient le mérite d'avoir établi les bases théoriques de la logique floue. L'intérêt de la logique floue réside dans sa capacité à traiter l'imprécis, l'incertain et le vague. Elle est issue de la capacité de l'homme à décider et agir de façon pertinente malgré le flou des connaissances disponibles. En effet, la logique floue a été introduite pour approcher le raisonnement humain à l'aide d'une représentation adéquate des connaissances.

IV.3 la commande par logique floue

IV-3-1 Notion de base sur la logique floue [29]

Dans cette section, nous allons voir les concepts de la théorie de la logique floue nécessaires pour la conception des régulateurs flous.

IV-3-1-1 Ensembles ordinaires

Etant donné un ensemble de référence X , on peut distinguer les éléments de X qui appartiennent à une certaine classe de X et ceux qui n'y appartiennent pas. Cette classe est alors un sous ensemble de X (au sens habituel de la théorie des ensembles), on le qualifie de classique ou d'ordinaire. Pour un ensemble ordinaire X , on peut définir la fonction d'appartenance $\mu_X(x)$ telle que :

$$\mu_X(x) = 0 \quad \text{si } x \in X \quad (IV.1)$$

IV-3-1-2 Ensemble flou

Soit X une collection continue ou discrète d'objets dénotés $\{X\}$, X est appelé univers de discours et x représente l'élément générique de X .

Un ensemble flou peut être vu comme une généralisation du concept d'ensemble ordinaire dont la fonction d'appartenance prend seulement deux valeurs $\{0,1\}$ (ensemble booléen). Ainsi sur un univers de discours X , un sous ensemble flou A est caractérisé par une fonction d'appartenance μ_A prenant ses valeurs dans l'intervalle $[0,1]$, alors l'ensemble flou A dans X est défini par l'ensemble des pairs (élément générique, degrés d'appartenance).

$$A = \{(x, \mu_X(x)) / x \in X\} \quad (IV.2)$$

IV-3-1-3 Variables linguistiques

La description d'une certaine situation, d'un phénomène ou d'un procédé contient en général des expressions linguistiques (floues), comme : *chaud, froid, grand, petit, ...etc.*

Les expressions de ce genre forment les valeurs d'une variable linguistique, qu'on appelle valeurs floues qu'on peut représenter par des fonctions d'appartenances.

IV-3-1-4 Fonction d'appartenance [30]

Afin de permettre un traitement numérique des variables linguistiques dans la prise de décisions douces sur ordinateur, une définition des variables linguistiques à l'aide de fonctions d'appartenance s'impose. Dans ce contexte, on associe à chaque valeur de la variable linguistique une fonction d'appartenance désignée par $\mu_A(x)$ où x est la variable linguistique, tandis que A indique l'ensemble concerné. Une valeur précise de $\mu_A(x)$, sera désignée par le degré ou le facteur d'appartenance. Il est à noter que l'ensemble des éléments de x pour lesquels $\mu_A(x) > 0$, est appelé support de A . Le plus souvent, on utilise pour les fonctions d'appartenance les fonctions suivantes :

- **Fonction triangulaire**

Elle est définie par trois paramètres $\{a, b, c\}$, qui déterminent les coordonnées des trois sommets.

$$\mu(x) = \max\left(\min\left(\frac{x-a}{b-a}, \frac{c-x}{c-b}\right), 0\right) \quad (IV.3)$$

- **Fonction trapézoïdale**

Elle est définie par quatre paramètres $\{a, b, c, d\}$

$$\mu(x) = \max\left(\min\left(\frac{x-a}{b-a}, 1, \frac{d-x}{d-c}\right), 0\right) \quad (IV.4)$$

- **Fonction gaussienne**

Elle est définie par deux paramètres $\{\sigma, m\}$

$$\mu(x) = \exp\left(-\frac{(x-m)^2}{2\sigma^2}\right) \quad (IV.5)$$

- *Fonction sigmoïde*

Une fonction sigmoïde est définie par deux paramètres $\{a, c\}$

$$\mu(x) = \frac{1}{1 + \exp(-a(x-c))} \quad (IV.6)$$

IV-3-2 Raisonnement flou

Les ensembles flous et la théorie des possibilités sont des éléments importants de la représentation des connaissances imparfaitement définies. Pour raisonner sur de telles connaissances, la logique classique ne suffit pas. On utilise la logique floue lorsque les connaissances sont imprécises, vagues et éventuellement incertaines. Voici quelques définitions utilisées :

Soit x une variable linguistique et A une caractéristique :

Définition 1 (proposition)

Une proposition floue est définie à partir d'une variable linguistique $(x, T(x), X)$ par la qualification : " x est A ".

Définition 2 (conjonction)

La conjonction de deux propositions floues est réalisée par l'opérateur ET par exemple : " x_1 est A_1 ET x_2 est A_2 "

Définition 3 (disjonction)

La disjonction de deux propositions floues est réalisée par l'opérateur OU par exemple : " x_1 est A_1 OU x_2 est A_2 "

Définition 4 (implication)

L'implication entre deux propositions floues définit aussi une proposition floue que l'on peut exprimer par : " x_1 est A_1 ALORS x_2 est A_2 "

Plus généralement, on peut construire des propositions floues par conjonction, disjonction ou implication portant sur des propositions floues élémentaires.

Définition 5 (règle floue)

Une règle floue est une proposition floue utilisant une implication entre deux propositions floues quelconques. Par exemple :

'' SI x_1 est A_1 ET x_2 est A_2 ALORS x_3 est A_3 ''

Où : *'' x_1 est A_1 ET x_2 est A_2 ''* est la prémisse de la règle

Et *'' x_3 est A_3 ''* est sa conclusion.

En utilisant les règles de composition d'inférence, nous pouvons formaliser une procédure d'inférence, appelé raisonnement flou sur l'ensemble des règles (*Si-Alors*).

IV-3-3 Commande par la logique floue [31]

La commande par logique floue est le domaine dans lequel il existe le plus de réalisation effective, en particulier industrielle. Son but est de traiter des problèmes de commande de processus, le plus souvent à partir des connaissances des experts.

IV-3-3-1 Contrôleur flou

La configuration de base d'un contrôleur flou (*fig. IV.2*) comprend les éléments suivants

Fuzzification

C'est la partie du contrôleur flou chargé de convertir les grandeurs physiques en variables linguistiques.

Base de règles

Elle contient les définitions des termes utilisés dans la commande et l'ensemble des règles caractérisant la cible de la commande et décrivant la conduite de l'expert.

Les différents ensembles flous sont caractérisés par des désignations standards: la signification des symboles est indiquée au tableau (IV-1).

Tableau IV-1 : Désignation standard des ensembles flous.

Symboles	Significations
NG	Négatif Grand
NM	Négatif Moyen
NP	Négatif Petit
EZ	Environ Zéro
PP	Positif Petit
PM	Positif Moyen
PG	Positif Grand

Inférence

L'inférence transforme à l'aide du jeu de règles (en manipulant la base de règles). La partie floue issue de la fuzzification en une nouvelle partie floue qui caractérise la sortie du contrôleur.

Défuzzification

La défuzzification consiste à convertir la partie floue issue de l'inférence en une grandeur physique.

Plusieurs stratégies de défuzzification sont utilisées:

- Méthode du maximum :

La commande est égale à la valeur dont le degré d'appartenance est le plus fort.

- Méthode de la moyenne des maximas :

La commande sera égale à la moyenne des valeurs dont le degrés d'appartenance est maximal.

- Méthode du centre de gravité:

C'est la méthode la plus utilisée dans les contrôleurs flous. Dans celle-ci la commande sera égale au centre de gravité de l'ensemble flou de sortie, on obtient donc pour :

- *Univers de discours discret*

$$C = \frac{\sum_{i=1}^n x_i \mu_A(x_i)}{\sum_{i=1}^n \mu_A(x_i)} \quad (IV.7)$$

- *univers de discours continu*

$$C = \frac{\int x \cdot \mu_A dx}{\int \mu_A dx} \quad (IV.8)$$

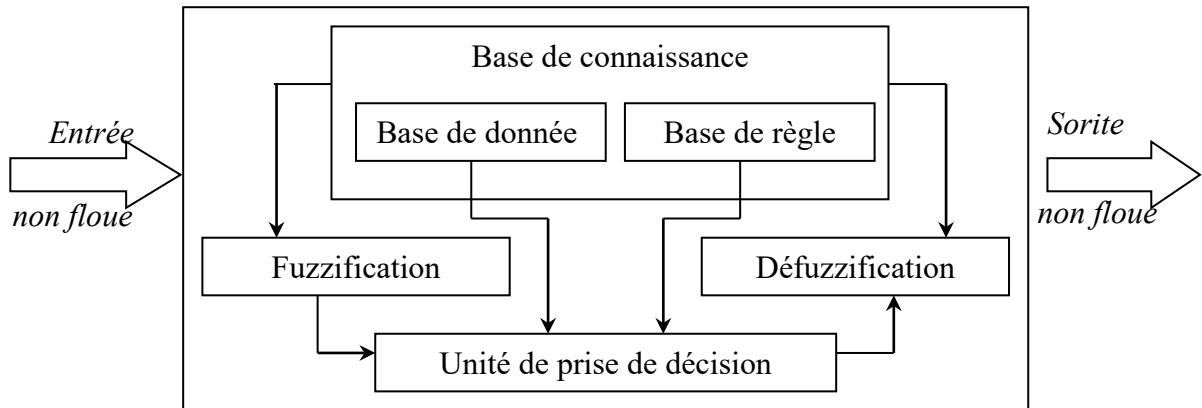


Figure IV.1 : Architecture générale d'un contrôleur flou

IV-3-4 Types de régulateurs flous

Il existe plusieurs types de régulateurs flous, qu'il diffèrent de mécanisme d'inférence utilisé, dont on cite: régulateur de *Mamdani*, de *Sugeno*, ... etc.

IV-3-4-1 Régulateur de type *Mamdani*

Mamdani fut le premier à utiliser la logique floue pour la synthèse de commande. Il utilise le minimum comme opérateur de jonction et l'implication pour représenter le graphe flou associé à chaque règle et l'opérateur maximum pour l'agrégation. Dans la règle i :

$$\text{Si } x_1 \text{ est } A_1 \text{ et } \dots \text{ et } x_n \text{ est } A_n \text{ Alors } y \text{ est } B^i$$

où B^i sont des sous-ensembles flous. Les B^i forment en général une partition de l'univers de sortie.

IV-3-4-2 Régulateur de type *Sugeno*

Dans les régulateurs de ce type, les conclusions des règles ne sont symboliques (i.e. représentées par des sous-ensembles flous) mais une fonction des entrées, par exemple :

$$b^i = f(x_1, \dots, x_n)$$

Où : $f(\cdot)$ est généralement une fonction polynomiale.

Et la sortie du régulateur est donnée par :

$$y = \frac{\sum_{i=1}^n \alpha_i(x) * b^i}{\sum_{j=1}^n \alpha_j(x)} \quad (IV.9)$$

où les α_i sont les valeurs de vérité de chaque règle pour $i=1$ à n .

Notons que la sortie donnée par le régulateur est en effet la variation du signal de commande.

IV-3-5 Procédé lors de la conception d'un réglage par logique floue

Le procédé à suivre lors de la conception d'un réglage par logique floue est assez différent de celui d'un réglage conventionnel. La figure IV-2 en montre les étapes principales.

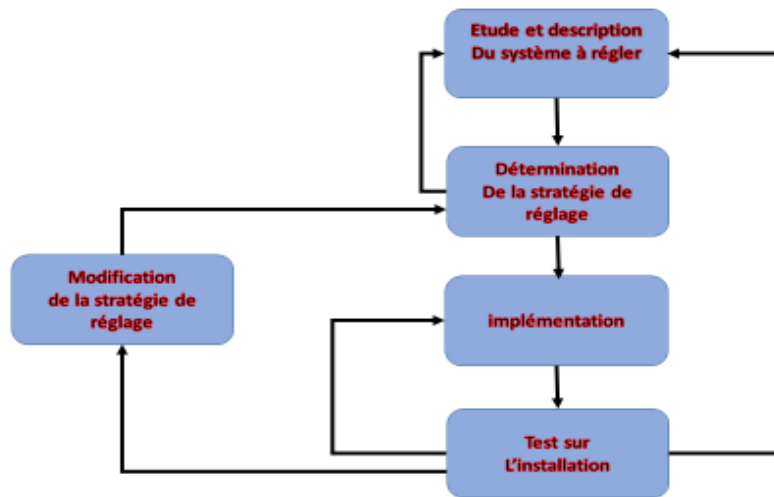


Figure IV.2 : Procédé lors de la conception d'un réglage par logique floue.

IV-4 Principe d'un régulateur floue :

Un régulateur flou permet de déterminer la commande à appliquer à un processus à partir de la valeur s de la variable de sortie de ce contrôleur ; celle-ci est elle-même déterminée à partir des valeurs des variables d'entrée e , Δe du régulateur par des relations floues, ou règles floues.

Théoriquement, le nombre d'entrée n'est pas limité. En pratique, cependant, il n'est pas rationnel d'utiliser plus de trois variables d'entrée puisque la détermination des règles devient trop complexe. Pour le régulateur décrit par la figure I.11, nous n'avons intervenir que deux variables d'entrée :

$e(k) = y_d(k) - y(k)$: Écart entre la consigne $y_d(k)$ et le signal de sortie du processus $y(k)$.

$\Delta e(k) = e(k) - e(k-1)$: Variation du signal d'écart à l'instant k .

IV-5 Avantages et désavantages du réglage par logique floue

Les avantages principaux des régulateurs flous sont les suivants :

- Il n'est pas nécessaire de faire un modèle mathématique du système à régler.
- On peut implémenter des connaissances de l'opérateur du processus.

On peut maîtriser les systèmes non linéaires et difficiles à modéliser.

- Grâce à des processus dédiés, On peut régler des processus rapides.

Les désavantages sont les suivants :

- Manque de directives précises pour la conception d'un régulateur.
- Approche artisanale et non systématique (il est souvent très difficile d'implémenter les connaissances de l'opérateur).
- Impossibilité de démontrer la stabilité du circuit de réglage en toute généralité (en absence d'un modèle valable).
- Précision du réglage en général peu élevée.
- Cohérence des inférences non garantie a priori (possible apparition de règles d'inférence contradictoire).

IV.6 Application de la commande floue

IV.6.1 Synthèse des régulateurs

La base des règles est obtenue à partir de la matrice de Macvicar-whelan, les régulateurs des deux sous-systèmes se différencient seulement par les gains de normalisation et les gains associés aux commandes, les fonctions d'appartenance sont Gaussiennes pour les entrées (erreur et variation de l'erreur) et les sorties. (la commande)

Tableau IV.2 : Base de règles (7*7)

Δe	NB	NM	NS	EZ	PS	PM	PB
e							
NB	NB	NB	NB	NM	NM	NS	EZ
NM	NB	NB	NB	NS	NS	EZ	PS
NS	NB	NB	NM	EZ	EZ	PS	PM
EZ	NB	NS	NS	PS	PS	PM	PB
PS	NM	NS	EZ	PM	PM	PB	PB
PM	NS	EZ	PS	PB	PB	PB	PB
PB	EZ	PS	PM	PB	PB	PB	PB

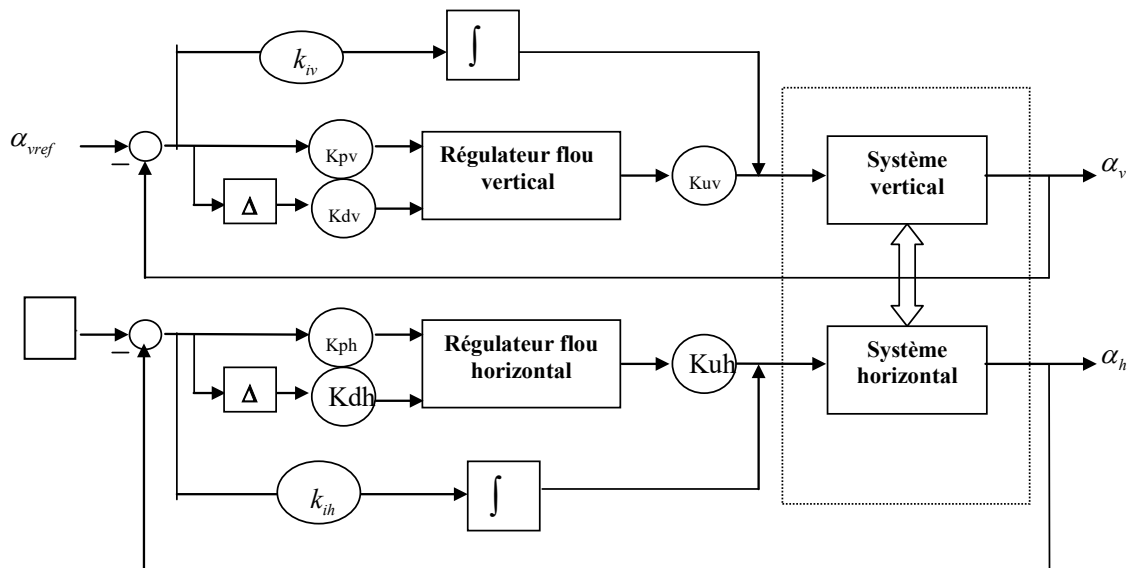


Figure IV-3 : schéma de la commande floue d'un vol d'hélicoptère TRMS

IV.6.2 les règles de la fuzzification

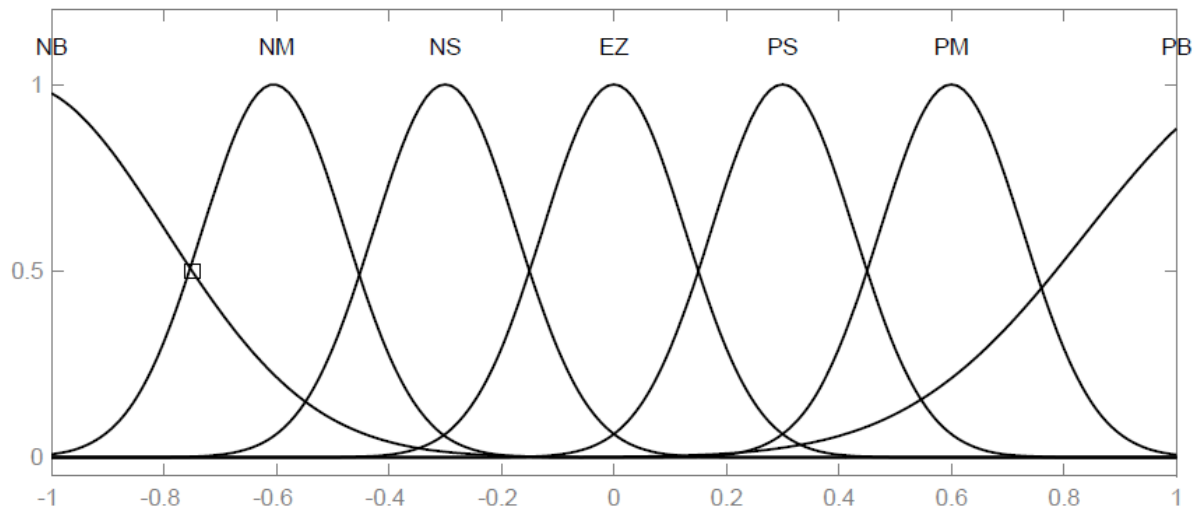


Figure IV.4 : Fuzzification avec sept fonctions d'appartenance dE

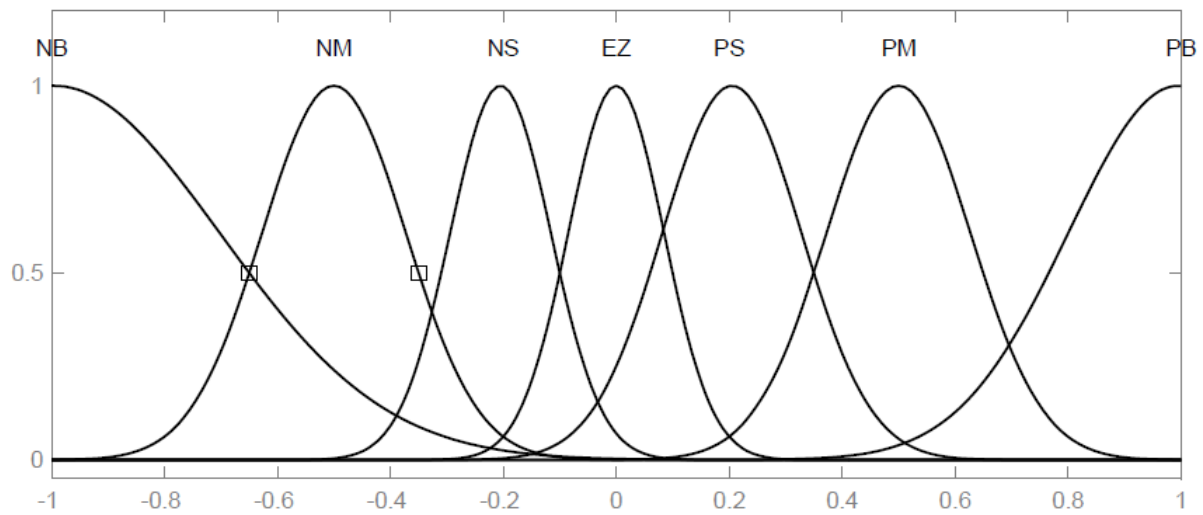


Figure IV.5 : Fuzzification avec sept fonctions d'appartenance E

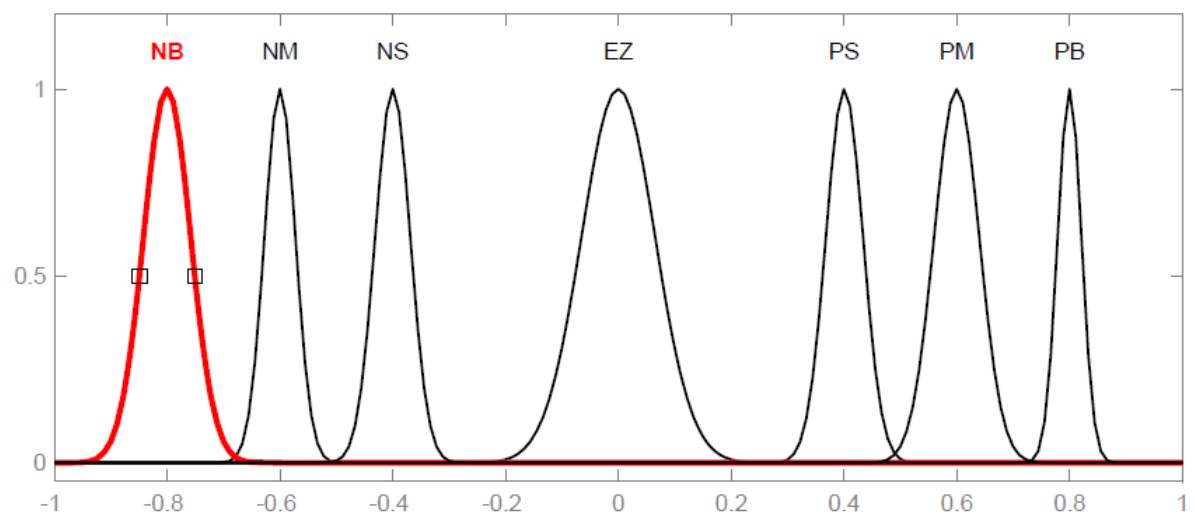


Figure IV.6 : Fuzzification avec sept fonctions d'appartenance de la Sortie

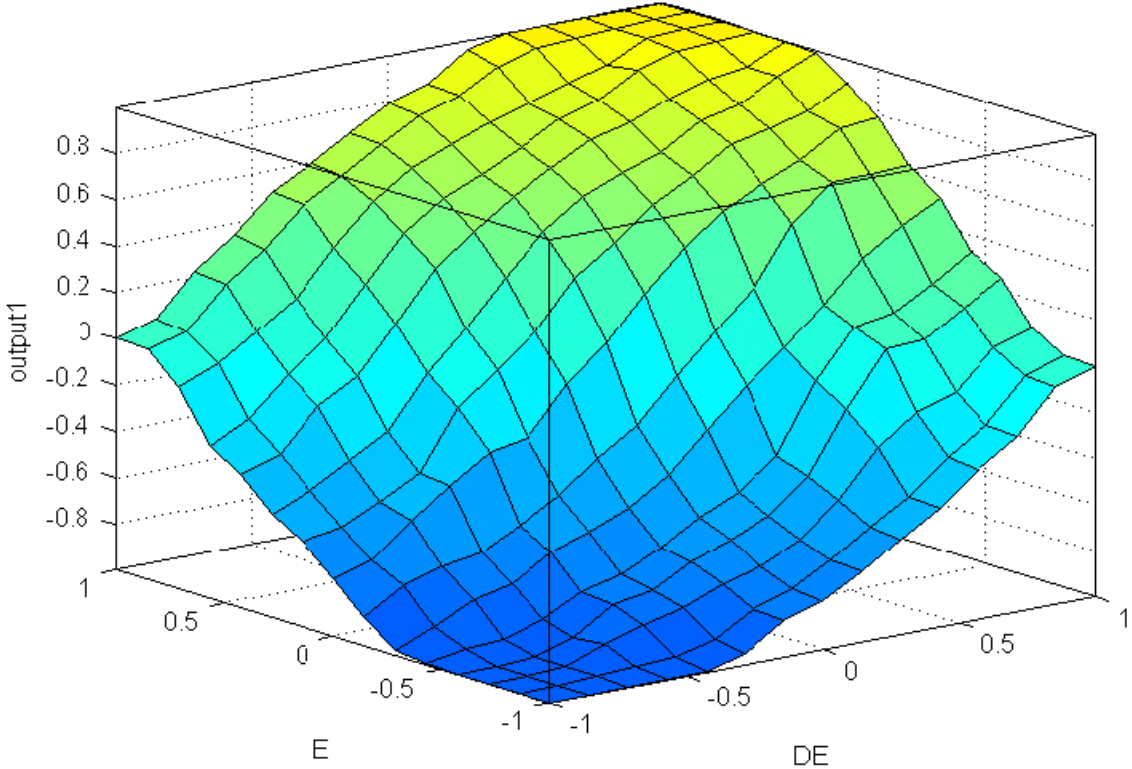


Figure IV.7 : La surface de le régulateur floue

IV-7 Résultat de validation de commande floue :

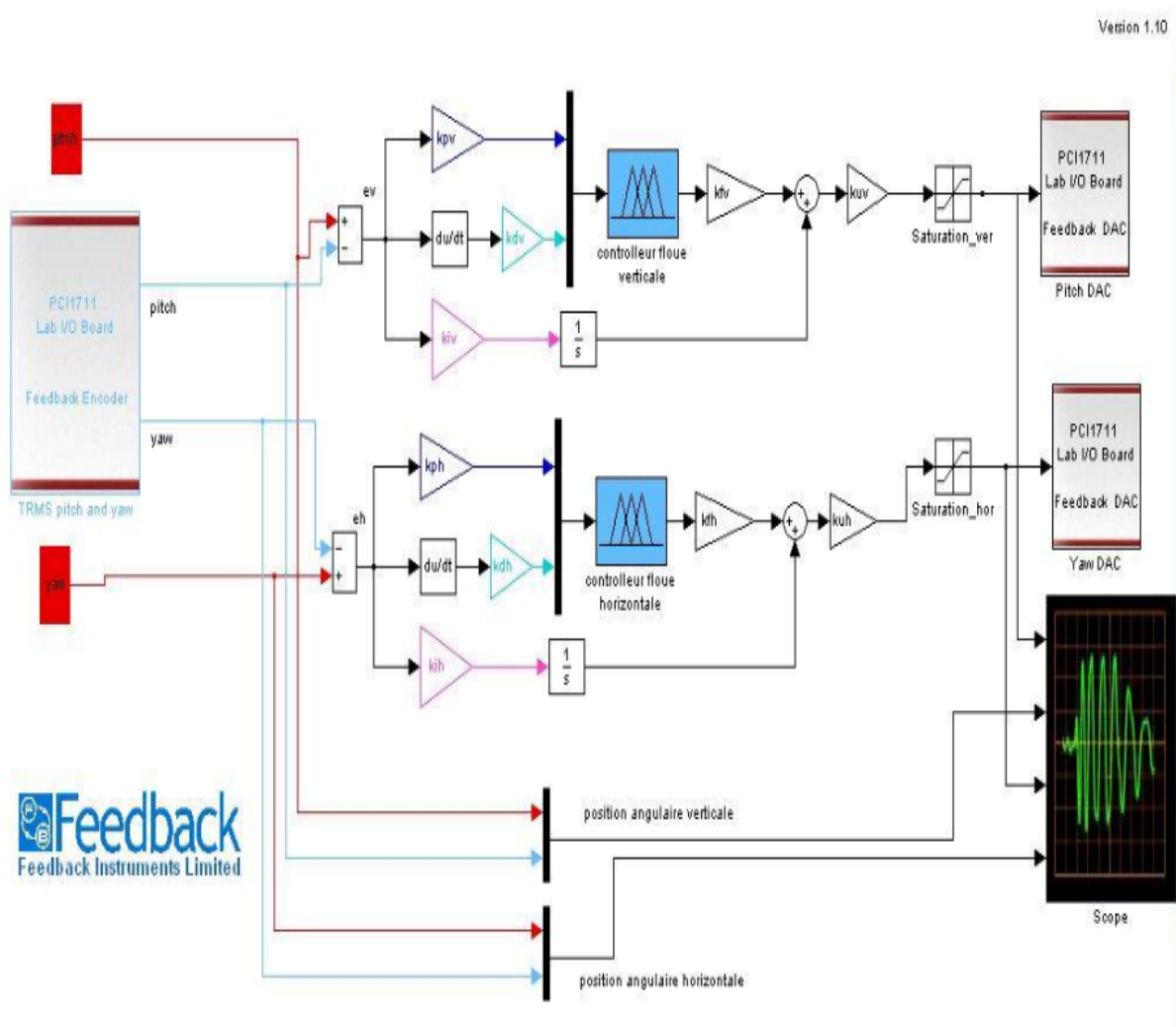


Figure IV.8: Implementation de la commande floue au simulateur de TRMS

III-7-1 Résultats de simulation de la commande floue

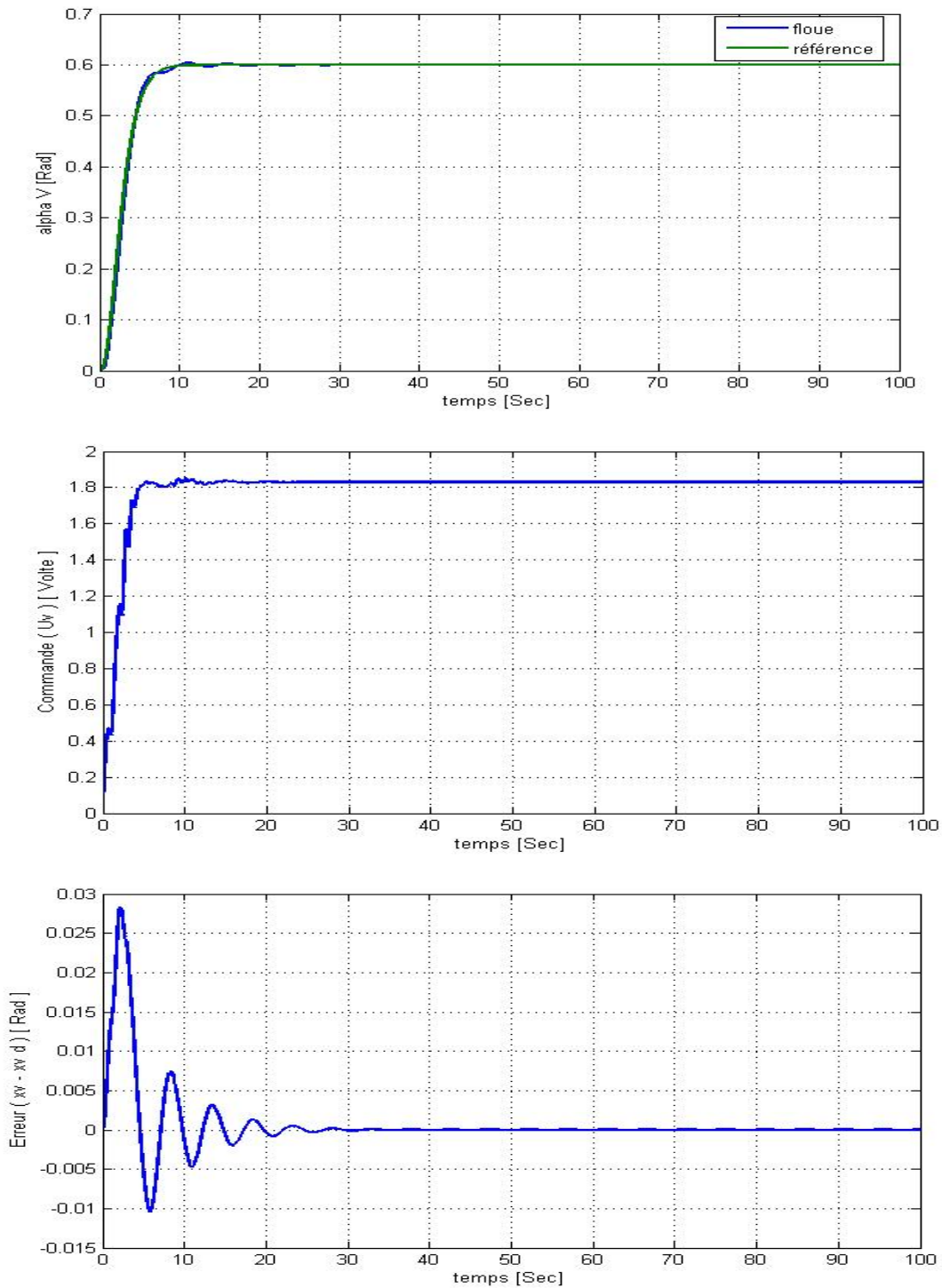


Figure IV.9 : Résultats de la commande floue avec une consigne échelon pour sous-système verticale

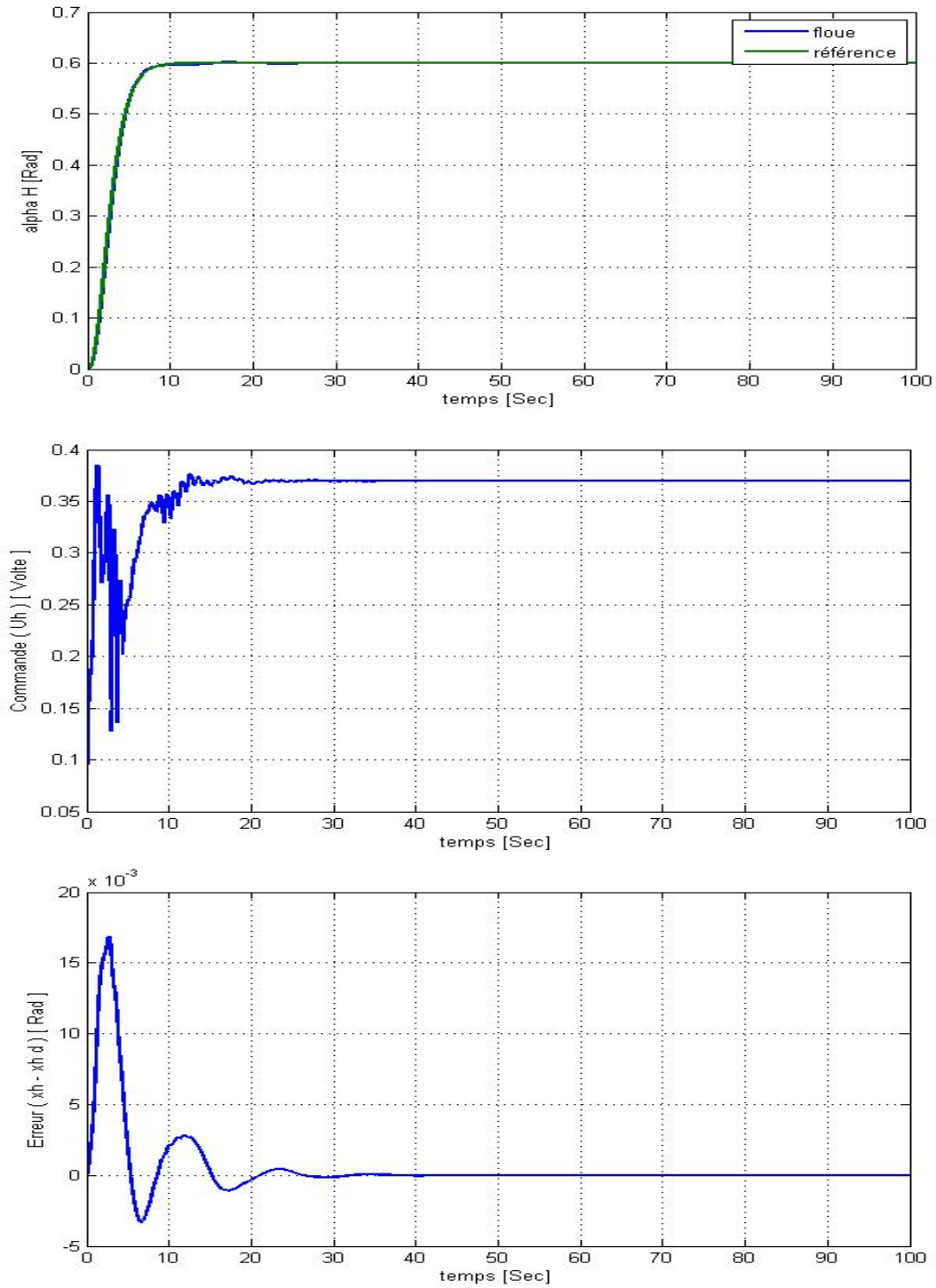


Figure IV.10 : Résultats de la commande floue avec une consigne échelon Pour sous-système horizontale

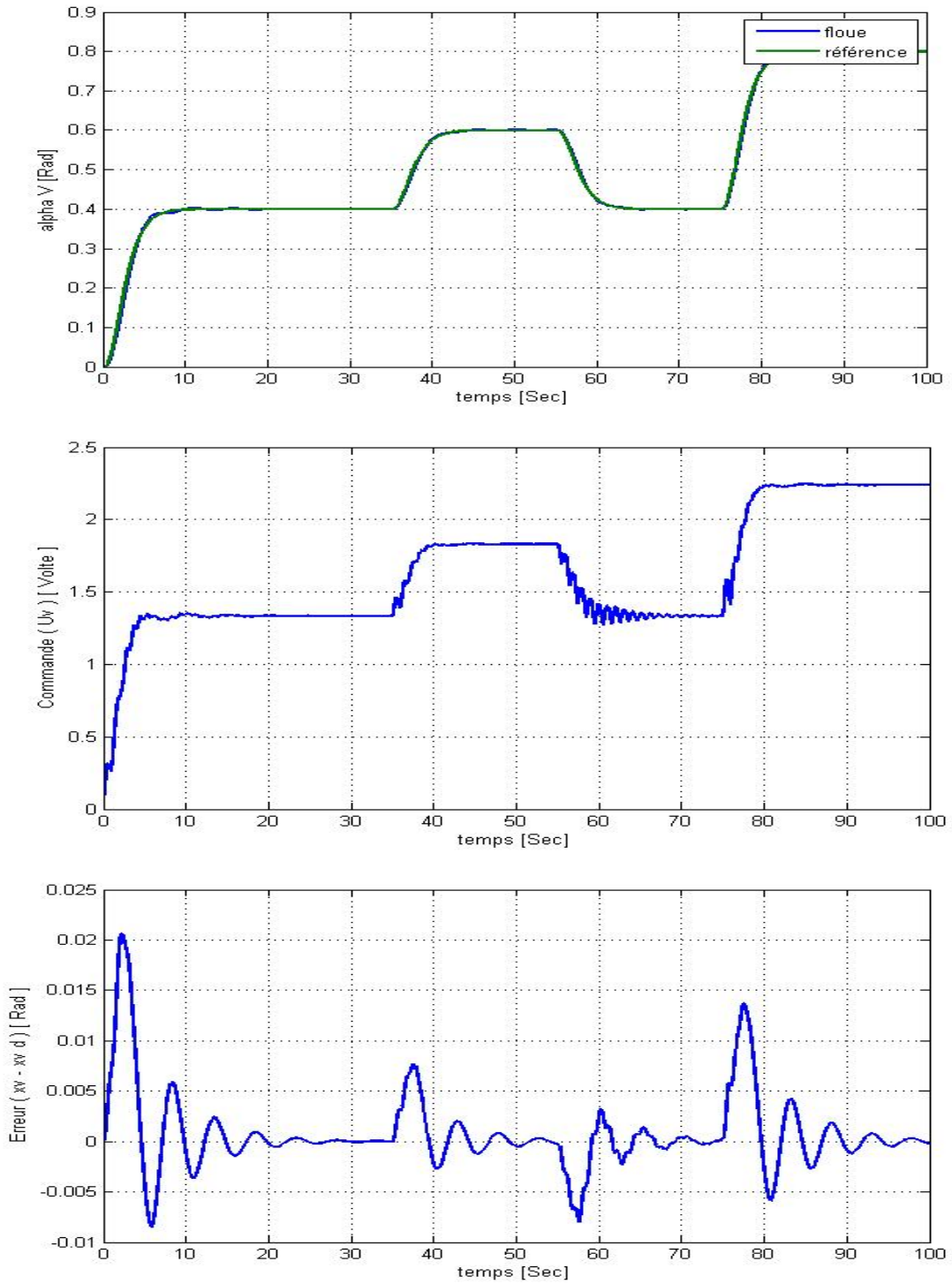


Figure IV.11 : Résultats de la commande floue avec une consigne carrée pour sous-système verticale

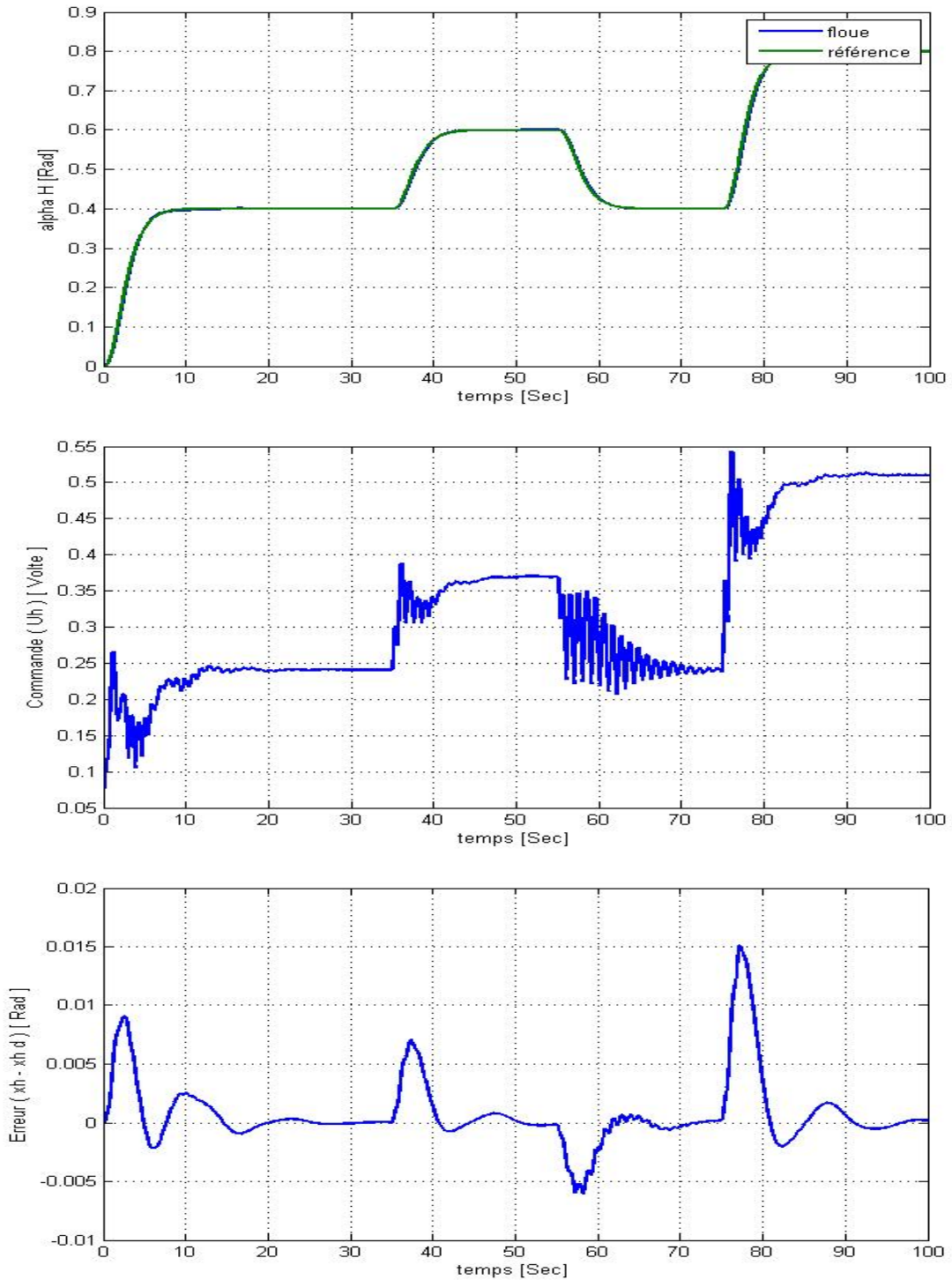


Figure IV.12 : Résultats de la commande floue avec une consigne carrée Pour sous-système horizontale

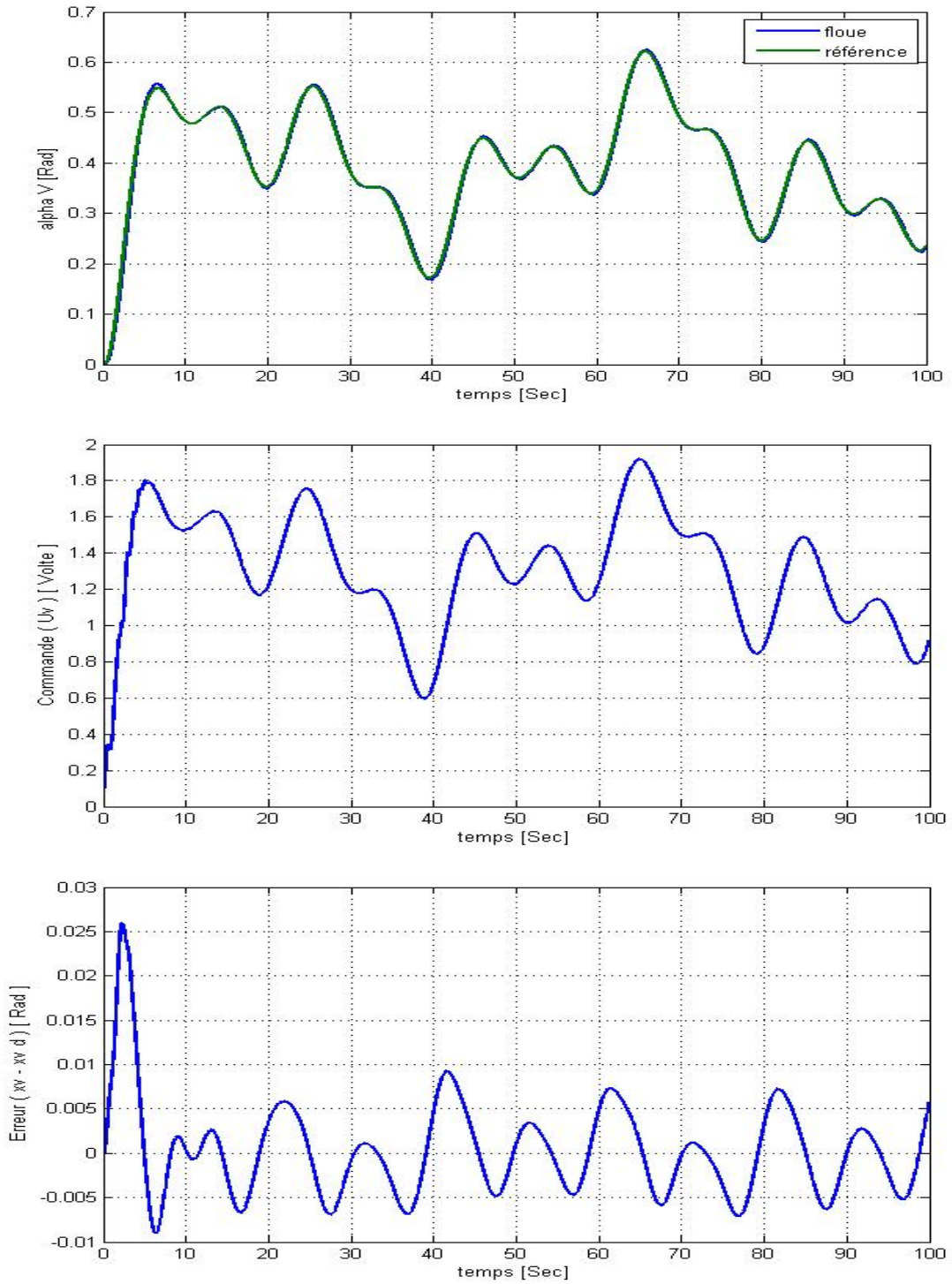


Figure IV.13 : Résultats de la commande floue avec une consigne sinusoïdale pour sous-système verticale

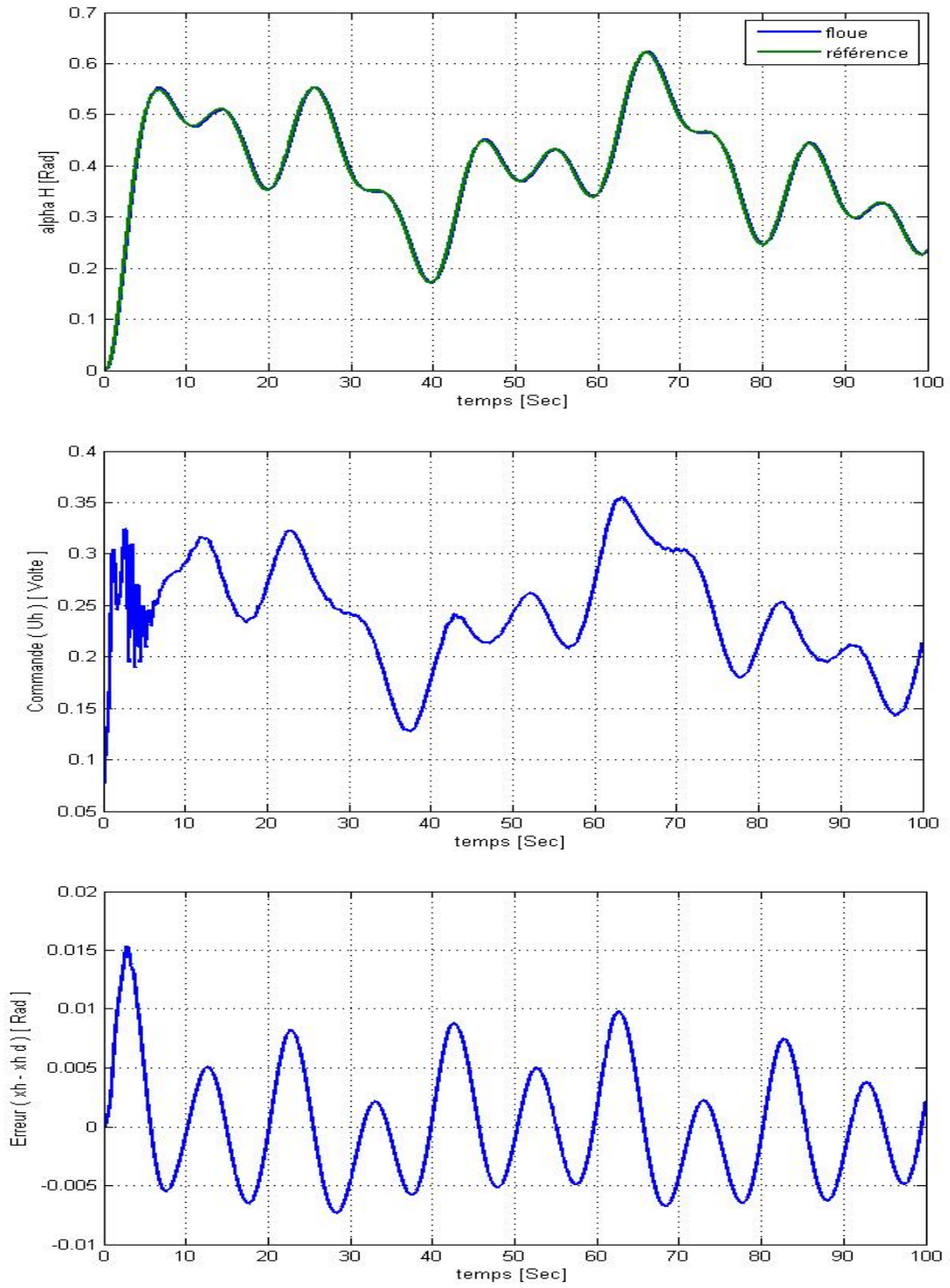


Figure IV.14 : Résultats de la commande floue avec une consigne sinusoïdale Pour sous-Système horizontale

III-7-2 Résultats expérimentaux de la commande floue au TRMS:

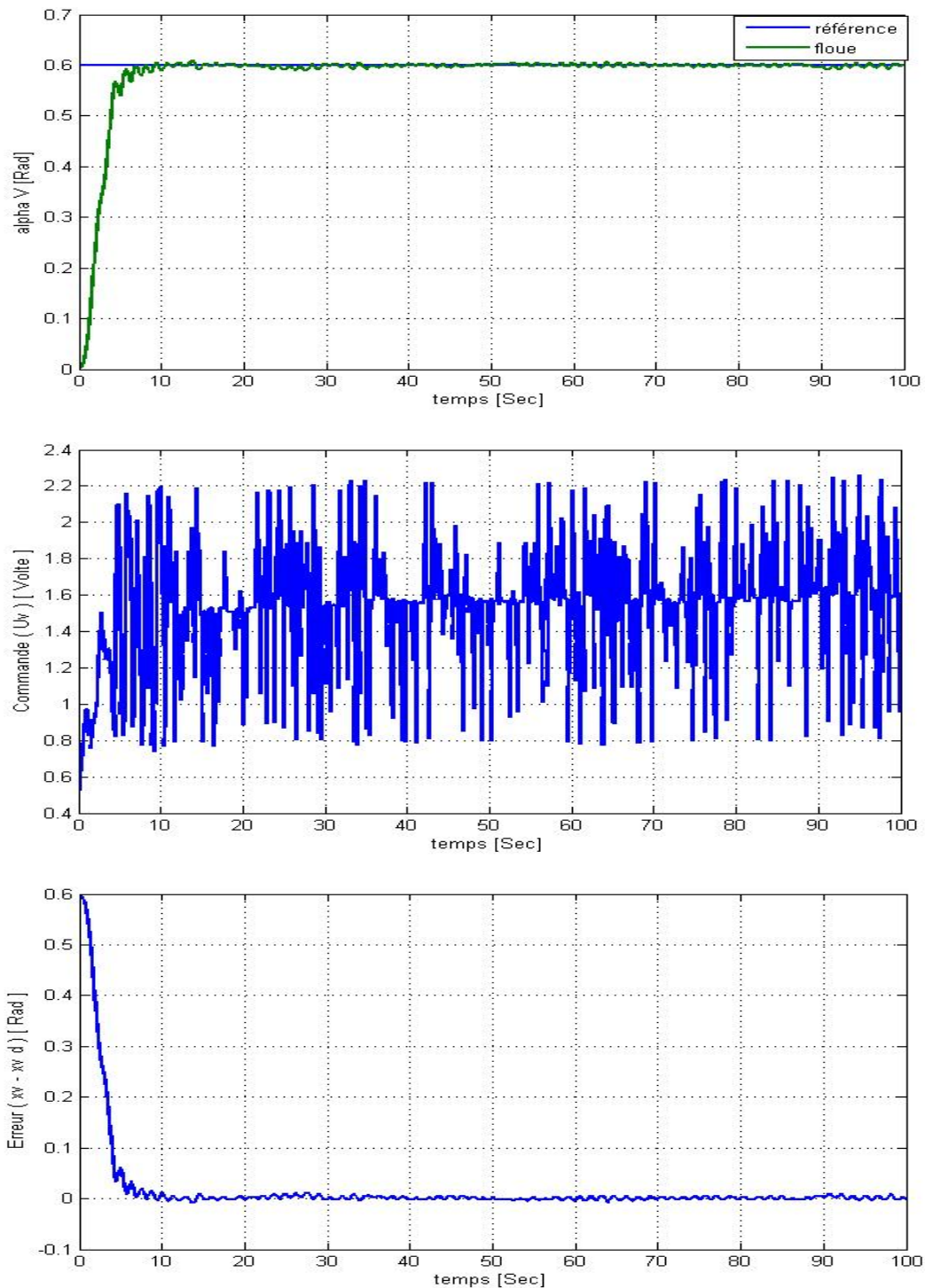


Figure IV.15 : Résultats de la commande floue avec une consigne échelon pour sous-système verticale

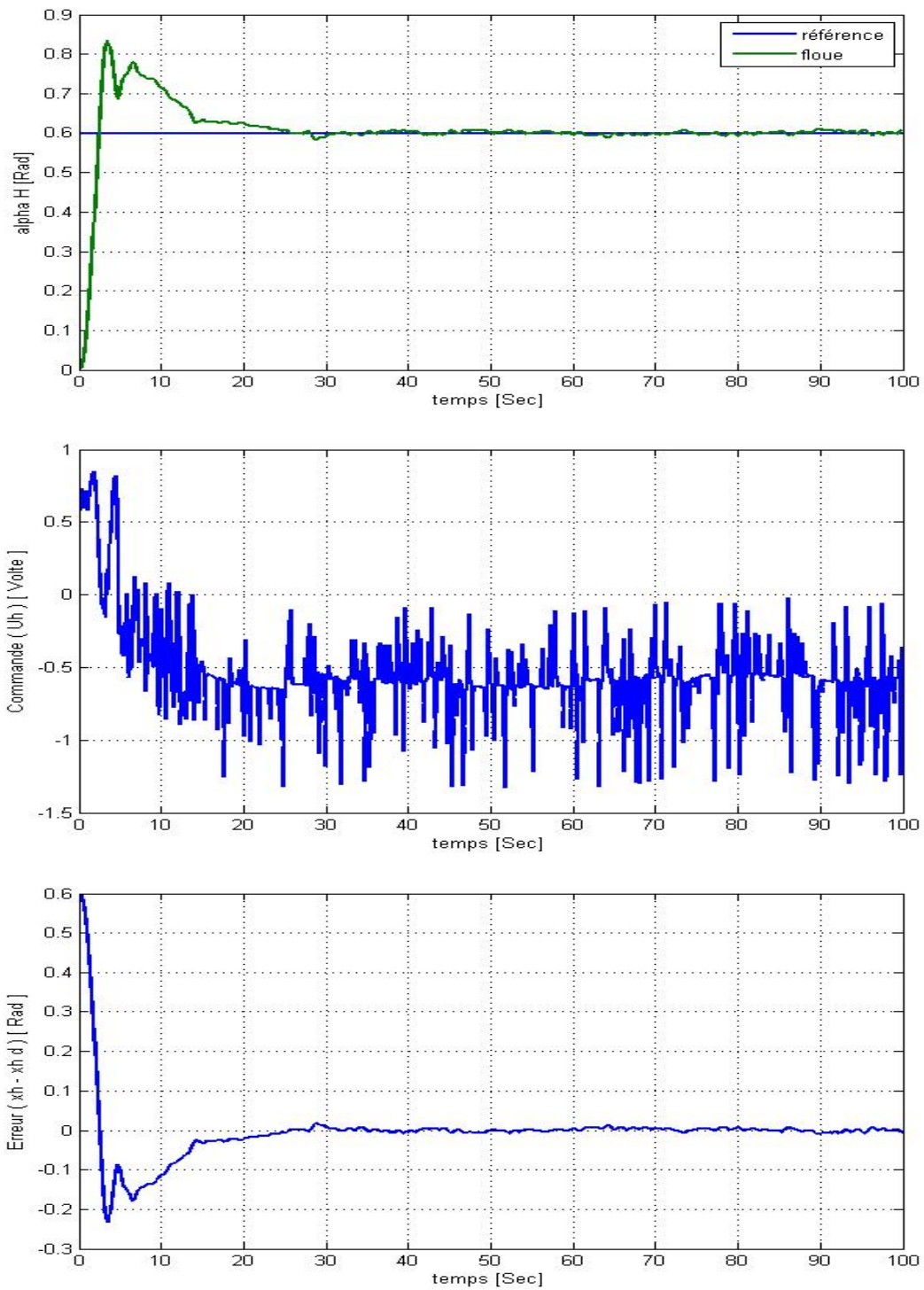


Figure IV.16 : Résultats de la commande floue avec une consigne échelon Pour sous-système horizontale

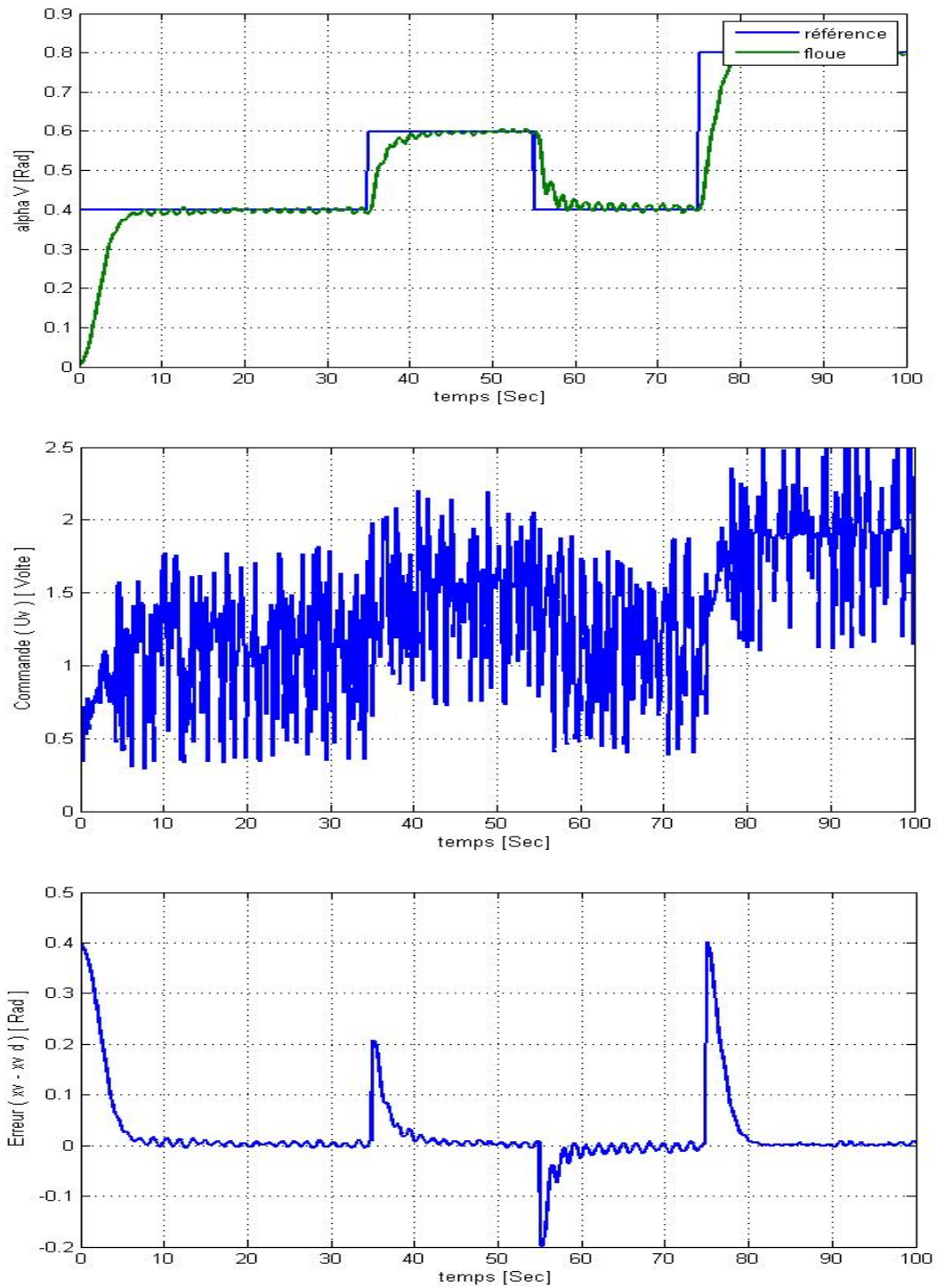


Figure IV.17 : Résultats de la commande floue avec une consigne carrée pour sous-système verticale

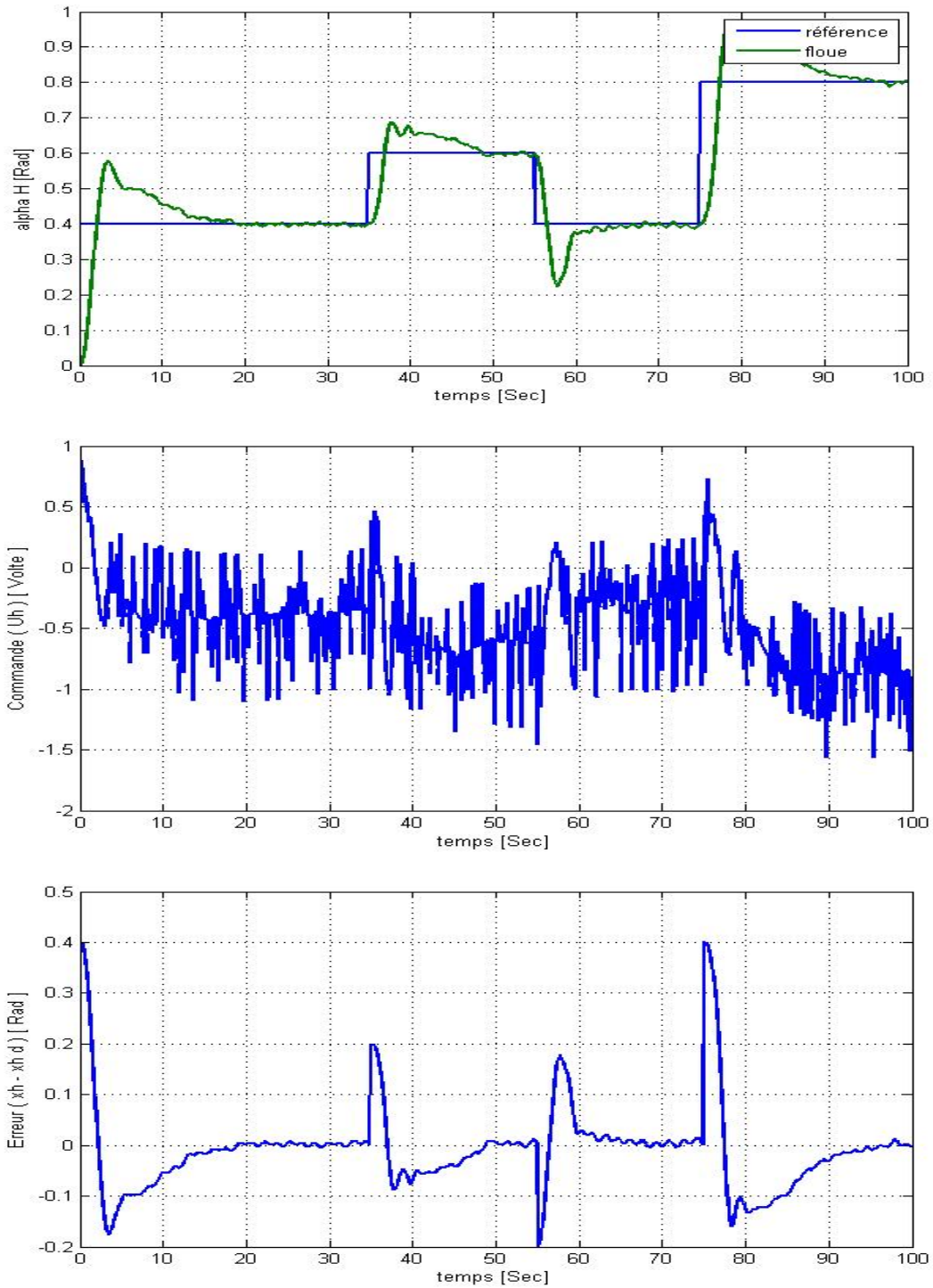


Figure IV.18 : Résultats de la commande floue avec une consigne carrée Pour sous-système horizontale

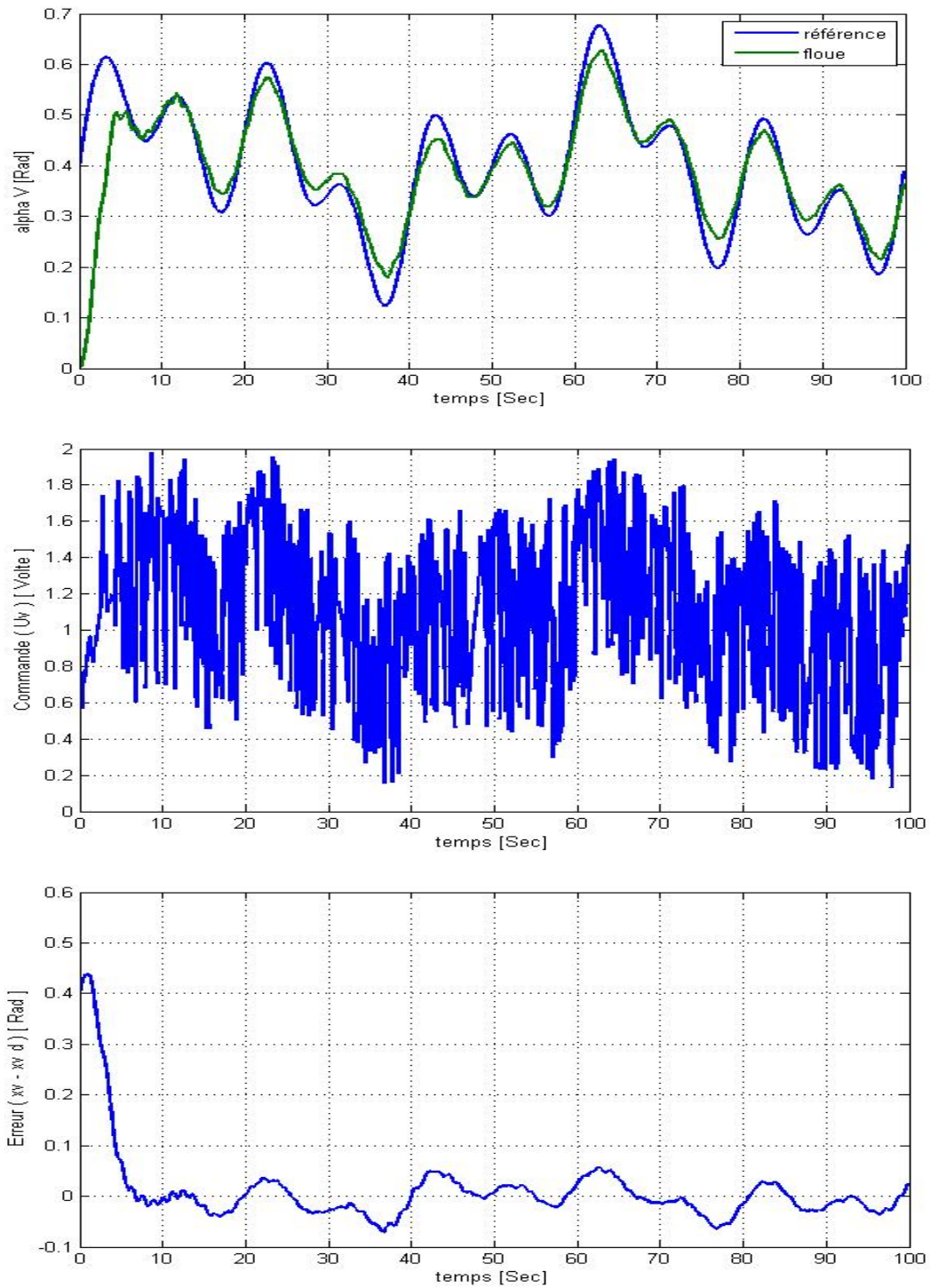


Figure IV.19 : Résultats de la commande floue avec une consigne sinusoïdale pour sous-système verticale

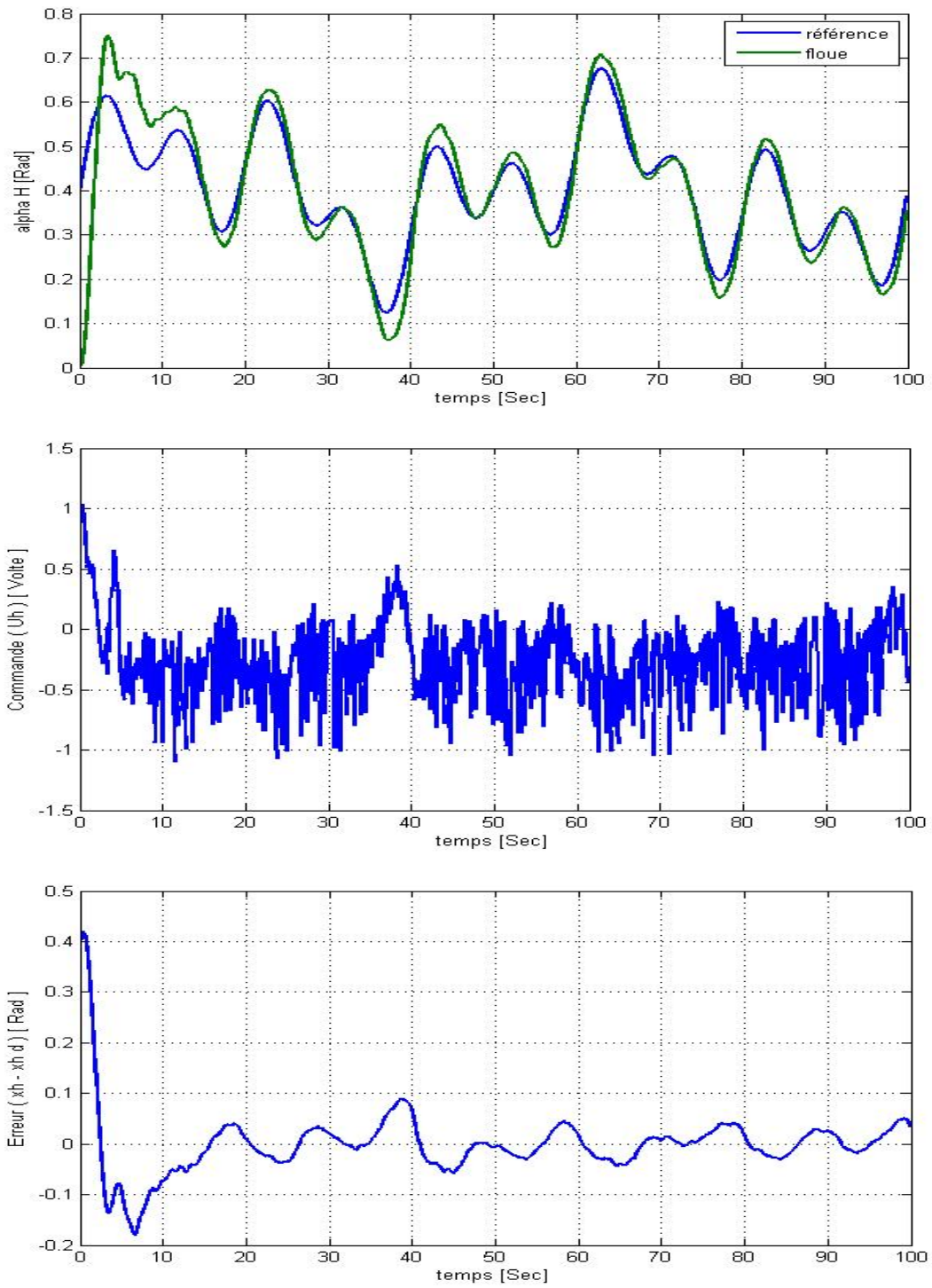


Figure IV.20 : Résultats de la commande floue avec une consigne sinusoïdale Pour sous-système horizontale

IV-8 Test de robustesse pour les positions angulaires du TRMS

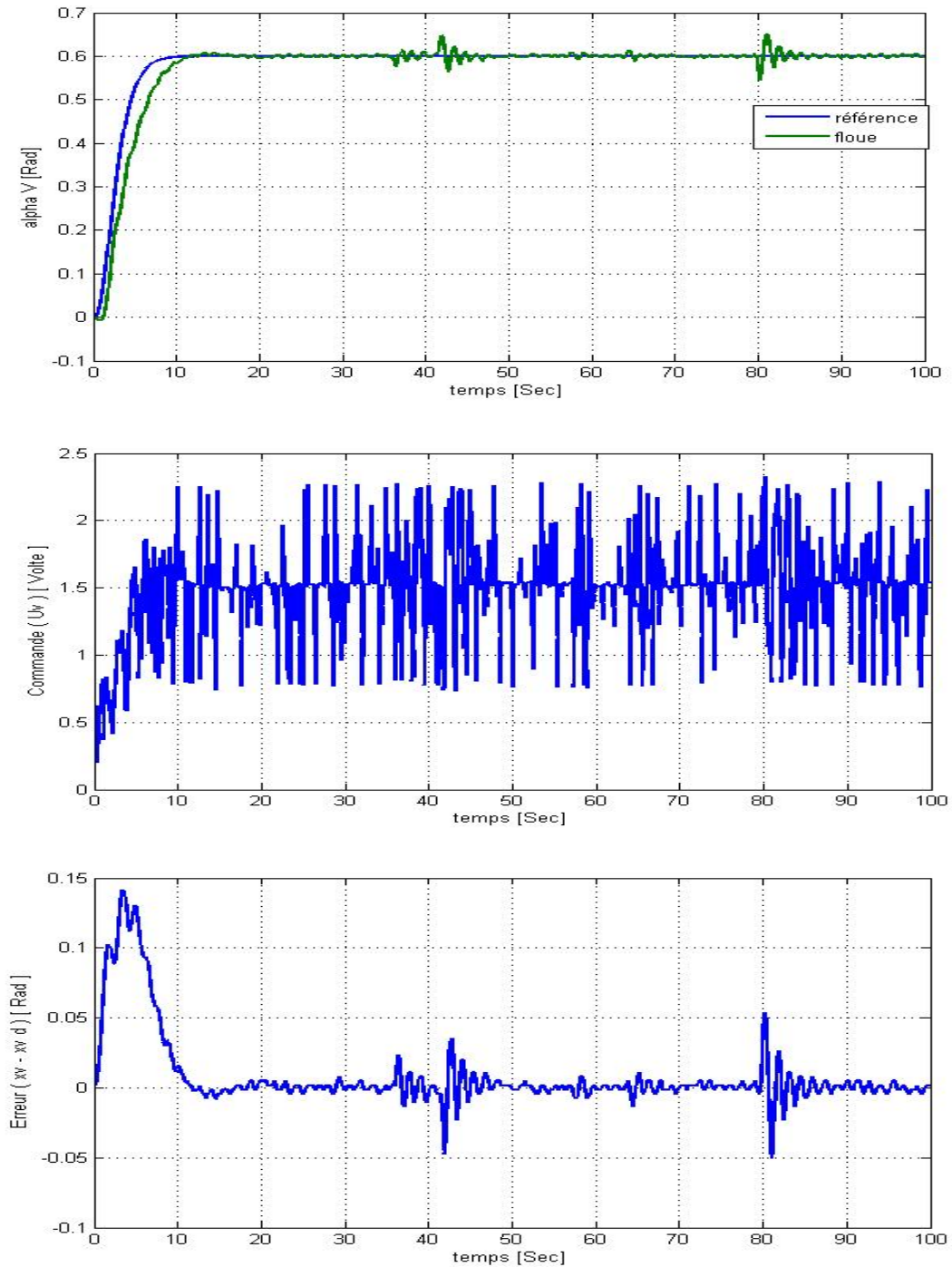


Figure IV.21 : Résultats de la commande floue avec une consigne échelon pour sous-système verticale

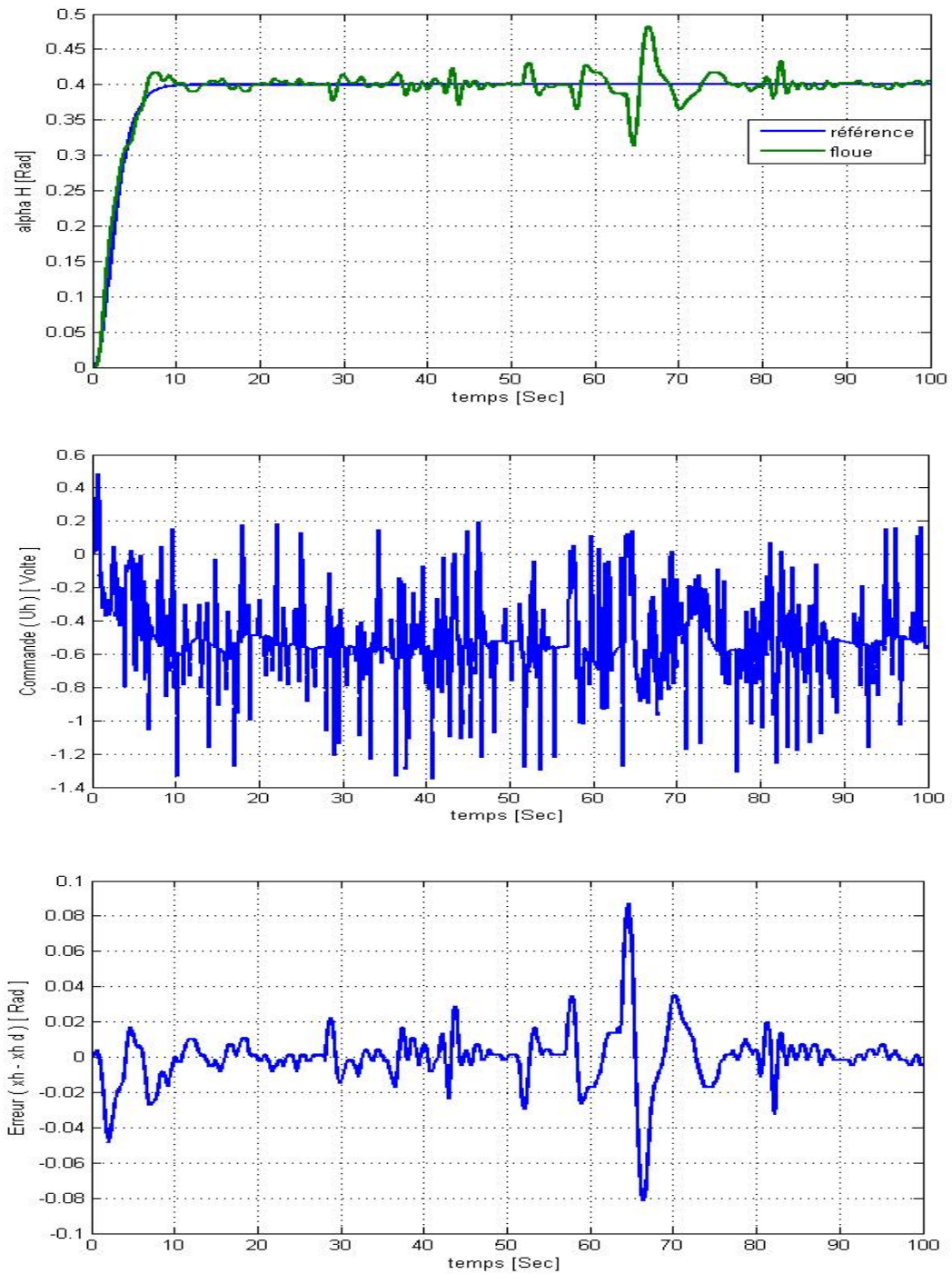


Figure IV.22 : Résultats de la commande floue avec une consigne échelon Pour sous-système horizontale

IV-9 Conclusion :

Dans ce chapitre, Nous avons vu la théorie de la commande floue et le principe de fonctionnement. Ainsi, nous avons cité les avantages et les inconvénients de cette technique de commande, nous avons faites les Résultat de validation de commande par logique floue en système réel avec des déférents consigne, en fin voire le test de robustesse de la commande avec une consigne échelon.

CHAPITRE V

*Commande par réseaux de
neurones appliquée au TRMS*

V. Commande par réseaux de neurones appliquée au TRMS

V-1 INTRODUCTION

Dans ce chapitre, nous avons effectué une présentation générale des réseaux de neurones.

Nous avons introduit des définitions et des notions utiles dans leur application.

Les réseaux de neurones artificiels, ou réseaux neuromémitiques présentent une alternative puissante à tout algorithme de calcul classique car ils ont la capacité de résoudre des problèmes mathématiques non linéaires. Ils sont constitués de cellules de calculs identiques appelées neurones formels, qui fonctionnent en parallèle. La mémorisation des informations est faite dans les poids des connexions de chaque neurone. Les RNA sont aussi caractérisés par leurs capacités d'apprentissage et de généralisation. Ces caractéristiques poussent les chercheurs de toutes les disciplines à étudier les RNA et à les appliquer dans leurs domaines de recherche.

➤ Historique

L'étude des réseaux de neurones date des années 40. Le premier modèle de neurone est dû aux deux neurobiologistes américains, Warren McCulloch et Walter Pitts qui supposèrent que l'impulsion nerveuse était le résultat d'un calcul simple effectué par chaque neurone et c'est grâce à la mise en commun des neurones dans le réseau que naît la pensée. Au même moment

Donald Hebb, psychologue canadien proposait une hypothèse qui associait l'activité des neurones à l'efficacité des synapses qui les relient [32].

Les travaux sur les réseaux de neurones formels ont connu des débuts prometteurs vers la fin des années 50, mais un manque de base théorique, et la concurrence de l'intelligence artificielle, ont conduit à une période de latence dans ce domaine. Il faut attendre la fin des années 70/milieu des années 80 pour que la recherche dans ce domaine redémarre, c'est en fait

la période où l’approche « classique » de l’intelligence artificielle basée sur la logique binaire a montré ses limites. Le physicien John Hopfield est l’un des chefs de file de cette reprise vers la fin des années 80. On peut dire que les réseaux de neurones sont devenus un domaine de recherche attractif, dans différentes disciplines : Psychologie, biologie, informatique, automatique.

Dans ce chapitre, nous présentons brièvement quelques notions générales sur le neurone biologique, ensuite les caractéristiques du neurone artificiel, puis l’algorithme d’apprentissage pour un RNA.

V-2 Réseaux de neurones artificiels

Les réseaux de neurones artificiels sont des modèles informatiques inspirés de la structure et du comportement des neurones biologiques. Ils sont composés d’unités interconnectées que l’on appelle neurones formels ou artificiels, recevant des informations provenant du milieu extérieur ou d’autres neurones. Les neurones sont connectés entre eux au moyen de synapses dont les poids modulent les informations transmises à d’autres neurones ou vers la sortie. Ces poids sont déterminés par apprentissage.

V-2-1 Réseaux Modèle formel d’un neurone

C’est le résultat de l’analogie directe qu’ont faite MacCulloch et Pitts, en 1943, avec le neurone biologique. Ce fut la première modélisation mathématique d’un neurone. Il s’agit d’un corps cellulaire qui exécute une somme pondérée des signaux d’entrée qui lui parviennent (figure 5.1). Si cette somme dépasse un certain seuil, le neurone est activé, ou au niveau haut. Autrement le neurone est dit désactiver, ou au niveau bas [33] [34]

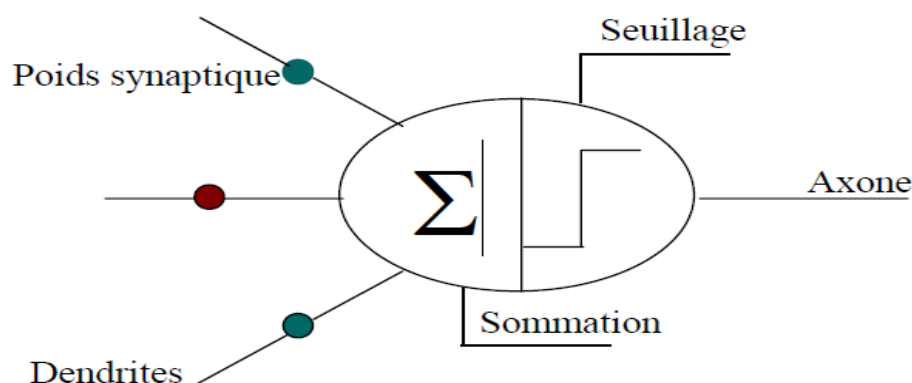


Figure V.1 : Neurone formel

La (figure 5.2) montre la correspondance entre le neurone biologique et le neurone artificiel.

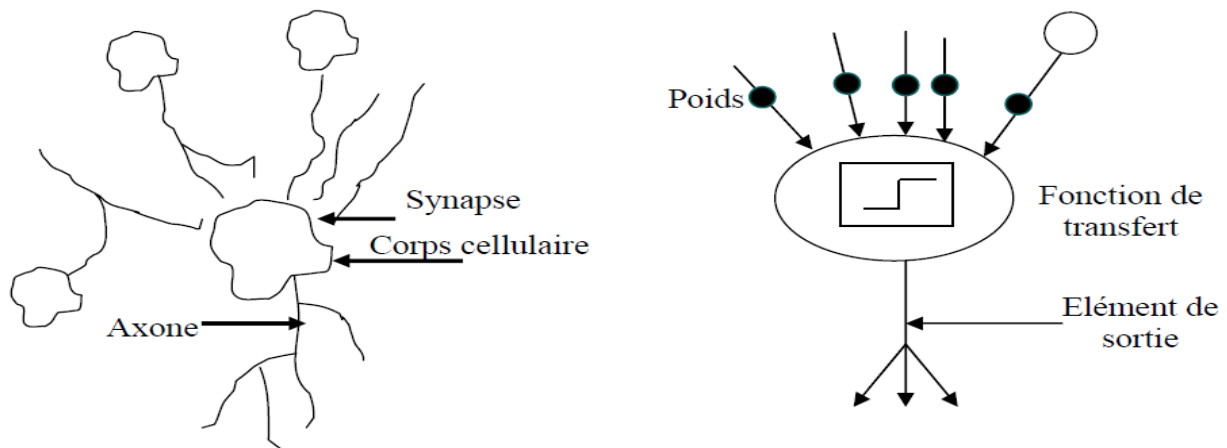


Figure V.2 : Mise en correspondance neurone biologique/ neurone artificiel

Le neurone formel est modélisé par deux opérateurs comme la montre la (figure 5.3).

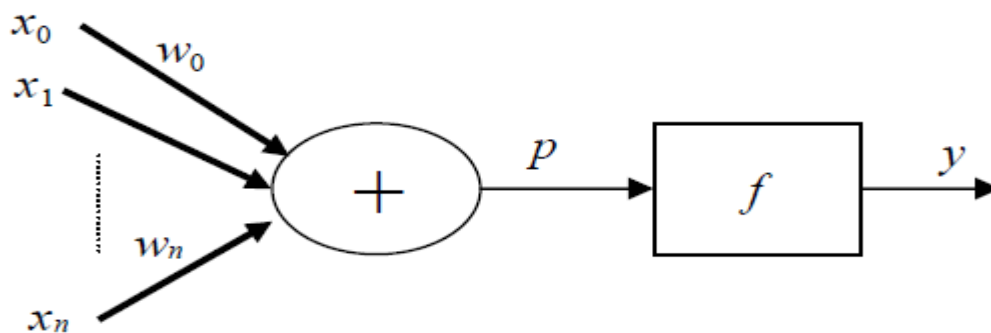


Figure V.3 : Modèle formel d'un neurone

Les paramètres sont attachés aux entrées du neurone : La sortie du neurone est une fonction non linéaire d'une combinaison des entrées $\{x_i\}$ pondérées par les paramètres $\{w_i\}$, qui sont souvent désignés sous le nom de « poids » ou, en raison de l'inspiration biologique des réseaux de neurones « poids synaptiques », conformément à l'usage (également inspiré par la biologie), cette combinaison linéaire sera appelé « potentiel ». Le potentiel p le plus fréquemment utilisé est la somme pondérée, à laquelle s'ajoute un terme constant ou « biais ».

V-2-1-1 Opérateur de sommation

$$p = \sum_{i=1}^n w_i x_i + w_0$$

Avec :

p : potentiel somatique

w_0 : biais

x_i : entrée du neurone i

w_i : poids synaptiques (Les connexions entre neurones ont des poids variables, ce qui rend les neurones plus ou moins influents sur d'autres neurones).

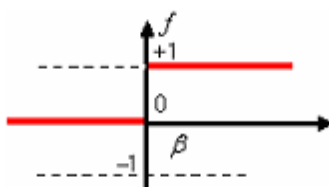
V-2-1-2 Fonction de seuil (ou d'activation)

C'est un opérateur qui calcule la sortie y du neurone en fonction de son potentiel P .

$$y = f(p)$$

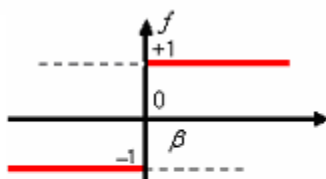
La fonction d'activation f doit être en générale saturable pour éviter des valeurs en sortie trop élevées qui peuvent déstabiliser le réseau. D'autre part, la fonction d'activation est en général à caractère non linéaire, ce qui rend les réseaux de neurones capables de simuler des fonctions complexes.

Les figures suivantes présentent des fonctions d'activation utilisées :



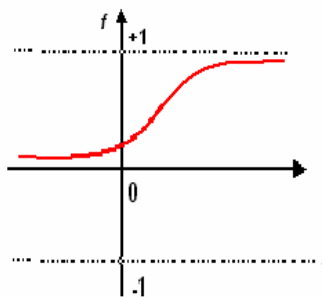
Fonction à seuil

Ici la fonction f est booléenne, si le produit scalaire p dépasse un certain seuil β le neurone répond par 1, mais si p est inférieur ou égal à β , le neurone répond par 0.



Fonction signe

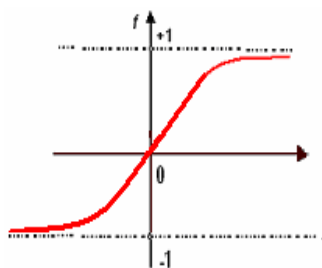
Comme dans le cas précédent la fonction f est une fonction binaire, si p est supérieur à β la sortie est +1, autrement, si p est inférieur ou égal à β la sortie est -1.



Fonction log-sigmoïde

Dans ce cas f est une fonction sigmoïde prenant ses valeurs sur l'intervalle $[0, 1]$.

$$f(p) = \frac{1}{1 + \exp(-p)}$$



Fonction tan-sigmoïde

Il s'agit d'une fonction sigmoïde (tangente hyperbolique) prenant ses valeurs sur

L'intervalle $[-1, 1]$ suivant l'expression :

$$f(p) = \frac{\exp(p) - \exp(-p)}{\exp(p) + \exp(-p)}$$

V-3 Apprentissage des réseaux de neurones

On appelle apprentissage des réseaux de neurones la procédure qui consiste à estimer les paramètres des neurones du réseau, afin que celui-ci remplisse au mieux la tâche qui leur est affectée.

L'apprentissage est la phase qui permet de déterminer les poids et le seuil de chaque neurone dans un RNA. Une fois l'apprentissage terminé, le réseau est prêt à remplir sa fonction. Cette fonction peut être la mémorisation, la classification, ou l'optimisation.

Dans le cadre de cette définition, on peut distinguer deux types d'apprentissage : l'apprentissage supervisé et l'apprentissage non supervisé.

V-3-1 Apprentissage supervisé

Le processus d'apprentissage supervisé est réalisé à l'aide d'un expert qui fournit une sortie que le réseau doit approcher au mieux en réponse aux stimuli provenant de son environnement (figure 5.4). Pour ce faire, les deux réponses sont comparées pour définir un critère d'erreur. Les méthodes classiques d'optimisation sont utilisées pour minimiser ce critère de sorte et atteindre la précision désirée. Parmi les algorithmes qui utilisent ce type d'apprentissage, on trouve la rétropropagation du gradient.

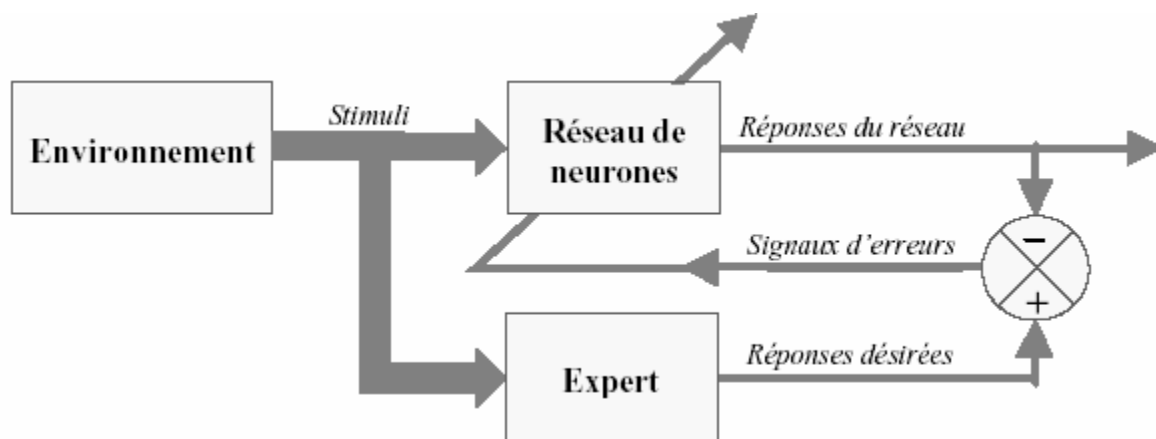


Figure V.4 : Schéma d'un apprentissage supervisé.

V-3-2 Apprentissage non supervisé

L'apprentissage non supervisé se fait en l'absence d'expert. C'est le réseau lui-même qui détermine une classification en s'auto-organisant par association des réponses aux stimuli (figure 5.5). La qualité de représentation de la connaissance au sein du réseau est exprimée par un critère d'erreur mesuré dépendant uniquement des réponses. Le modèle de Kohonen, entre autres, utilise ce type d'apprentissage.

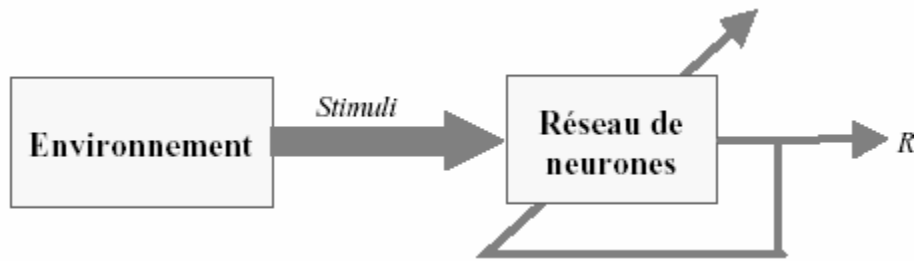


Figure V.5 : Schéma de l'apprentissage non supervisé.

➤ **L'apprentissage non supervisé est un outil à objectifs multiples :**

-Dans le cas de l'optimisation il détermine les paramètres donnant la solution optimale au problème.

-Dans le cas de la classification il divise l'espace de représentation en régions de décisions dont les limites sont définies par les règles de décisions adoptées dans le RNA et les représentants de chaque classe seront codés dans les poids (modèle de Kohonen).

V-3-3 Surapprentissage

Les travaux expérimentaux menés sur les réseaux de neurones ont montré que, si on dépasse un certain nombre d'exemples d'entraînement, le réseau peut perdre sa capacité de généralisation, et donner ainsi des sorties non satisfaisantes pour des nouvelles entrées en gardant des réponses correctes pour les entrées déjà apprises, c'est ce que l'on appelle le surapprentissage . [35]

Pour éviter ce phénomène, on effectue un test de généralisation chaque fois que le réseau aura appris un certain nombre d'exemples. Si le test s'avère être négatif, on arrête l'introduction de nouveaux exemples, pour éviter le surapprentissage. Cette explication est résumée dans un schéma graphique (figure V.6).

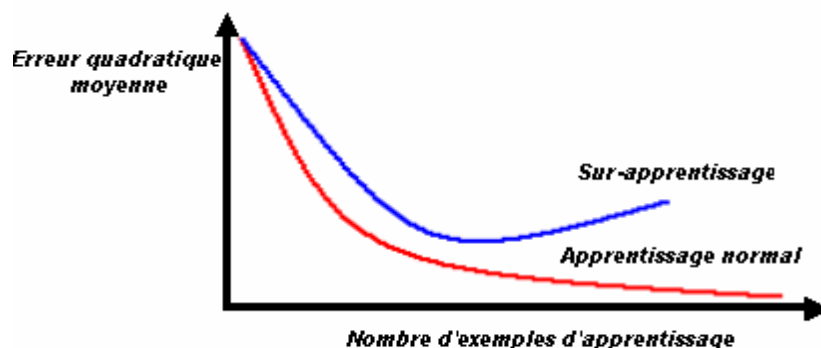


Figure V.6 : Phénomène de surapprentissage

V-4 Mémoire associative

Un réseau neuronal n'est rien d'autre qu'une mémoire associative. Son travail consiste en premier lieu à stocker ou à mémoriser des données sous forme d'entrées/sorties durant la phase d'apprentissage, et en second lieu à se rappeler, un exemple déjà mémorisé durant la phase de généralisation même si l'exemple présenté est bruité. Si les entrées et sorties sont du même type, on parlera d'auto-association. Si les entrées et sorties ne sont pas du même type, on parlera d'hétéro- association.

Dans la phase de rappel, la notion la plus importante est celle de la décision. Le réseau doit décider si la donnée est plus proche de tel exemple appris plutôt que de tel autre. Cette capacité est tout simplement analogue au fonctionnement du cerveau humain, puisque celui-ci peut, après avoir vu le visage d'une personne, l'associer automatiquement au nom de celle-ci.

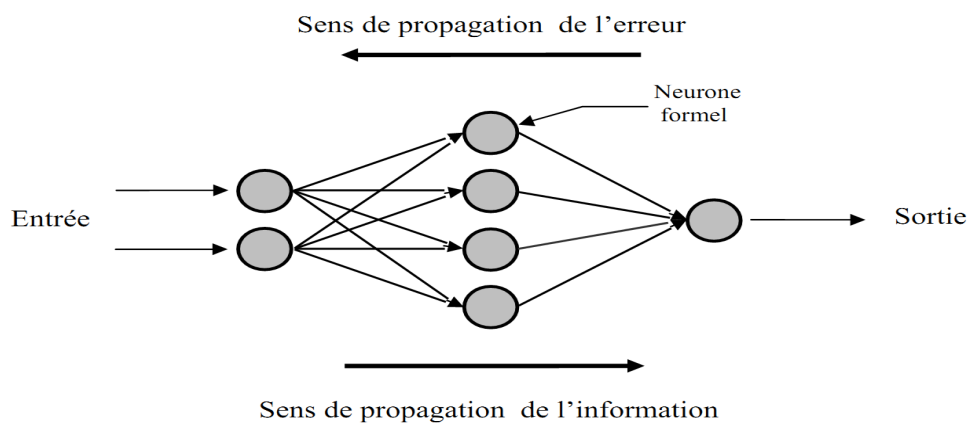


Figure V.7 : Un réseau multicouche comportant 2 neurones d'entrée, 4 neurones cachés et un neurone de sortie

V-5 Algorithmes d'apprentissage

La phase d'apprentissage est une étape déterminante dans la conception du réseau de neurones. Pour cela, des algorithmes appropriés ont été élaborés et développés au fil des années chaque algorithme est développé pour un type spécifique de réseau. Ces méthodes sont basées sur des techniques mathématiques connues qui ont prouvé leurs efficacités.

V-5-1 Algorithme de rétropropagation (Back-Propagation BP)

Nous présentons l'algorithme le plus connu qui est celui de rétropropagation du gradient, découvert par Werbos, dans sa thèse de doctorat, et développé par Rumelhart, Hinton et Williams en 1986. C'est grâce à lui que l'on doit le renouveau d'intérêt pour les RNA au début des années 80. Il s'agit d'une méthode d'apprentissage supervisée.

L'algorithme de rétro propagation se résume par les étapes suivantes :

Etape 1 : initialisation aléatoire des poids et des biais.

Etape 2 : appliquer le vecteur d'entrée de la couche d'entrée.

Etape 3 : calculer le vecteur d'entrée de chaque couche cachée:

$$[H^q] = [W^q]. [Y^{q-1}] + [B^q] , q = 1, 2 \dots \dots \dots , k - 1$$

Etape 4 : calculer le vecteur de sortie de chaque couche cachée:

$$[Y^q] = [F^q(H^q)] , q = 1, 2 \dots \dots \dots , k - 1$$

Etape 5 : calculer le vecteur d'entrée de la couche de sortie :

$$[H^k] = [W^k]. [Y^{k-1}] + [B^k]$$

Etape 6 : calculer le vecteur de sortie du réseau :

$$[Y_c] = [Y^k] = [F^k(H^k)]$$

Etape 7 : ajuster les poids et les biais de la couche de sortie :

$$[W^k] = [W^k] + \eta[\delta^k][Y^{k-1}]$$

$$[B^k] = [B^k] + \eta[\delta^k]$$

Avec : η : pas d'apprentissage

$$[\delta^k] = [(Y_d - Y_c). [F^k(H^k)]]$$

F' : représente la dérivée de la fonction d'activation vectorielle.

Etape 8 : modifier les poids et les biais de chaque couche cachée :

$$[W^q] = [W^q] + \eta[\delta^q][Y^{q-1}]$$

$$[B^q] = [B^q] + \eta[\delta^q]$$

Avec : $[\delta^q] = [W^q]. [\delta^{q+1}]. [F^q(H^q)]$

Etape 9 : calculer la sortie du réseau en utilisant les nouveaux poids.

L'organigramme de l'apprentissage pour les réseaux multicouches est présenté :

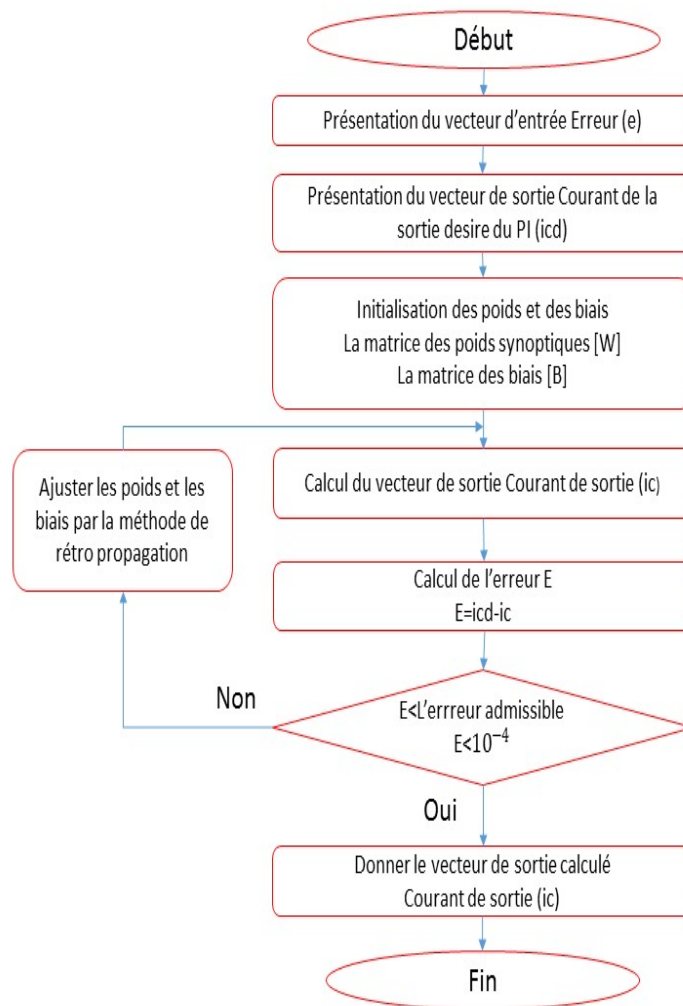


Figure V.8 : Organigramme de l'apprentissage pour les réseaux multicouches. [36]

Remarque :

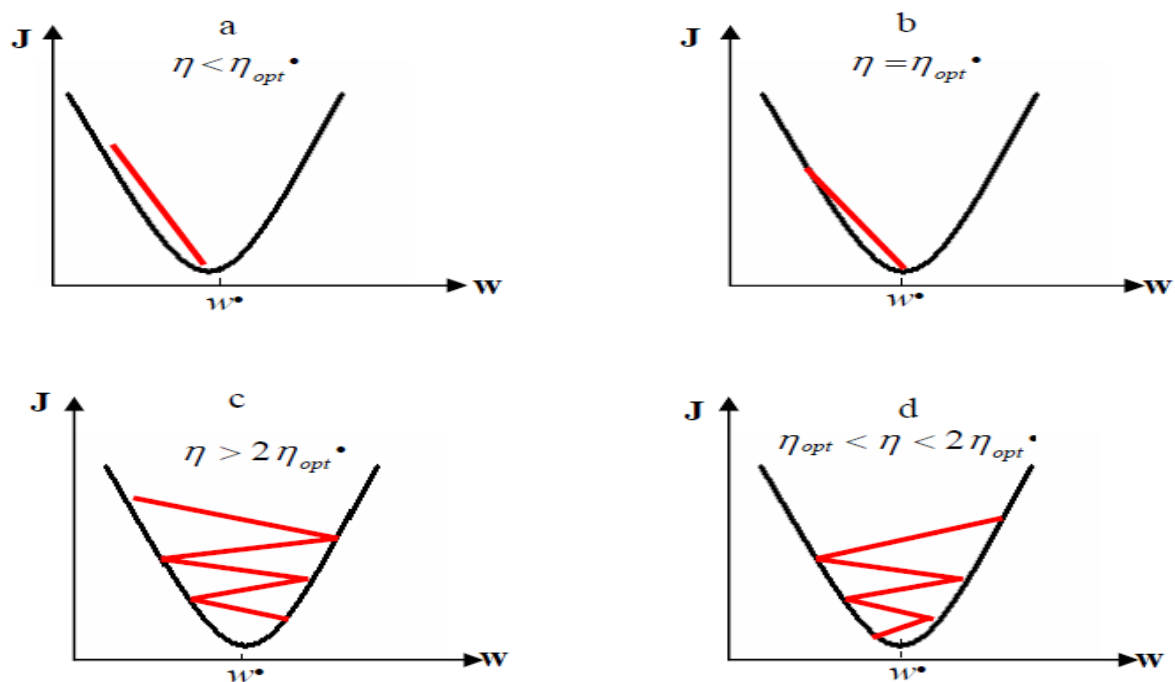
- L'initialisation des poids synaptiques w est aléatoire.
- L'expérience a montré qu'il est généralement, préférable d'initialiser les poids à des valeurs comprises entre 0 et 1 pour éviter de déstabiliser le réseau dès le départ [37] [38].
- Le pas du gradient η est choisi entre 0 et 1, et on le diminuera au fur et à mesure pour atteindre une valeur fixe.
- Le choix des exemples d'apprentissage est très important, un choix judicieux de ces derniers pourra faciliter la généralisation sur des exemples non appris. Ceci a poussé les chercheurs à faire l'étude statistique des données d'entraînement pour trier celles qui sont le plus représentatives du phénomène qu'on souhaite avoir.

- L'algorithme de rétropropagation du gradient a pu être étendu à certaines classes de réseaux récurrents c'est ce que l'on appelle la rétropropagation dynamique.

V-5-2 Difficultés et limites de l'algorithme

Des questions restent posées quant à l'utilisation de cet algorithme, à savoir :

- Combien faut-il utiliser des couches, et combien doit-il y avoir de neurones par couche ?
- Les réponses ne sont données que par expérience, et il n'existe aucune règle théorique précise qui puisse fixer le nombre de neurones ou de couches dans le réseau.
- Le problème de minimisation n'est pas facile à résoudre. En effet, la surface d'erreur peut présenter des caractéristiques peu satisfaisantes, telles que des minima locaux qui empêchent la convergence vers le minimum global, ainsi que des plateaux où les pentes sont très faibles.
- Le pas du gradient (figure 5.9) peut être difficile à choisir. S'il est faible, la convergence peut être très lente. S'il est élevé, l'erreur risque d'osciller sans pour autant converger.



a- Convergence lente b- Convergence rapide c- Divergence d- Convergence oscillatoire

Figure V.9 : Convergence et choix de η

V-5-3 Algorithme de back-propagation avec momentum

Comme nous l'avons mentionné précédemment, le pas du gradient peut être difficile à choisir. S'il est faible, la convergence peut être très lente. S'il est élevé, l'erreur risqué d'osciller sans pour autant converger.

Une solution proposée par D.E.Rumelhart [...], consiste à introduire un terme supplémentaire pour la réadaptation des poids.

L'équation d'adaptation devient alors :

$$\Delta w_{ji}(t) = -\eta E_{rj} x_i + \alpha \Delta w_{ji}(t-1)$$

Le terme ajouté est appelé momentum (élan, quantité de mouvement), en analogie avec la mécanique classique, ou un objet en mouvement garde l'élan acquis grâce à la quantité de mouvement qui lui a été communiquée précédemment. Le paramètre α est utilisé pour pondérer l'effet de ce terme. Sa valeur est généralement prise entre 0.8 et 0.9. Des recherches plus poussées ont montré que l'adaptation de sa valeur, comme c'est le cas pour le taux d'apprentissage η , donne des résultats très intéressants.

L'utilisation de cette méthode permet de faire sortir les poids des minimums locaux, (figure 2.14) afin de chercher d'autres optimums. Ce qui donne beaucoup de chance d'aboutir à un minimum global.

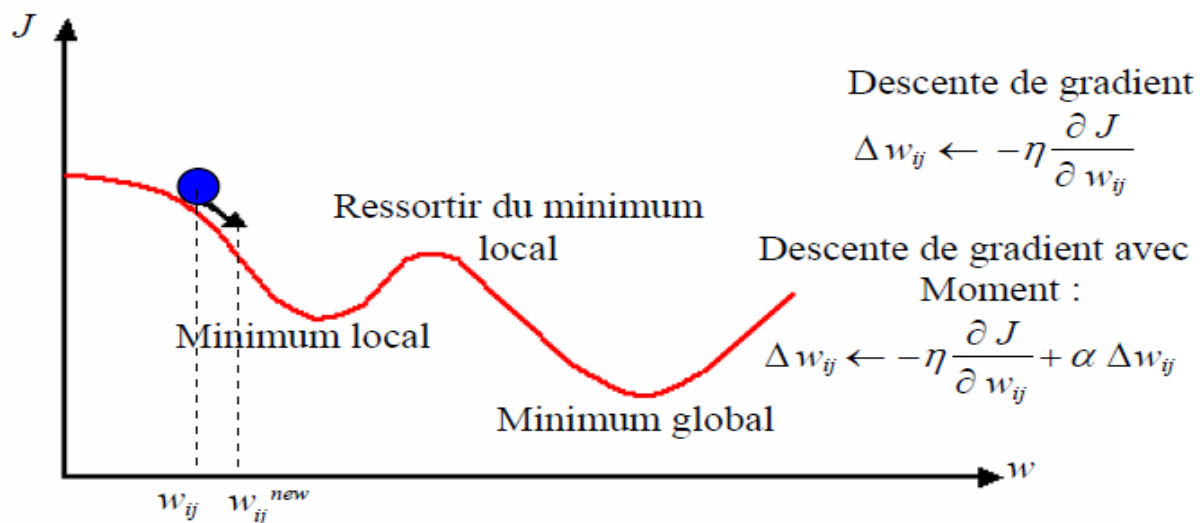


Figure V.10 : Effet de momentum pour échapper d'un minimum local

V.6 les étapes de la conception d'un réseaux neurone:

Pour construire un réseau de neurones, la première chose à faire n'est pas de choisir le type de réseau mais de bien choisir ses données d'apprentissage et ceux du test de validation. Ce n'est qu'ensuite que le choix du type de réseau interviendra. La construction d'un réseau de neurones passe par quatre étapes importantes : [39]

V.6.1 Choix des échantillons

Le choix des échantillons représente la première phase d'élaboration d'un réseau de neurones. Cette étape est déterminante pour déterminer le type de réseau approprié au problème à résoudre. La présentation de l'échantillon conditionne le type de réseau, le nombre de cellules d'entrée, le nombre de cellules de sortie et la manière dont il faudra mener l'apprentissage et la validation.

V.6.2 Normalisation des données

Afin d'améliorer la performance des réseaux neuronaux multicouches, il est important de normaliser les données d'entrée et de sortie de telle sorte qu'elles se trouvent dans l'intervalle $[-1 \ 1]$.

V.6.3 Elaboration de la structure du réseau

La structure du réseau dépend de près du type des échantillons utilisés. Il faut d'abord choisir le type de réseau. Dans le cas du réseau multicouche, il faudra choisir le nombre de neurones dans la couche cachée. Une des méthodes utilisée est de prendre une moyenne du nombre de neurones d'entrée et de sortie, mais il est plus judicieux de tester les différentes possibilités et choisir celle qui offre les meilleurs résultats.

V.6.4 Apprentissage

Cette phase consiste à calculer les pondérations optimales des différentes liaisons réseau de neurones, en utilisant un échantillon. Un algorithme d'entraînement est utilisé. Il consiste à présenter des valeurs à l'entrée et en fonction de l'erreur obtenue à la sortie, les poids accordés aux pondérations sont corrigés. C'est un cycle qui est répété jusqu'à ce que l'erreur du réseau soit minimale. Il faut prendre garde de ne pas surentraîner le réseau de neurones pour ne pas le spécialiser et le rendre moins performant.

V.6.5 Validation

Une fois le réseau calculé, il faut procéder à des tests pour vérifier que le réseau réagit bien comme souhaité, c'est la phase de validation. La méthode la plus simple étant de garder une partie de l'échantillon réservé à l'apprentissage, pour la validation.

V.7 Création de la base d'apprentissage et la validation de la commande neuronale :

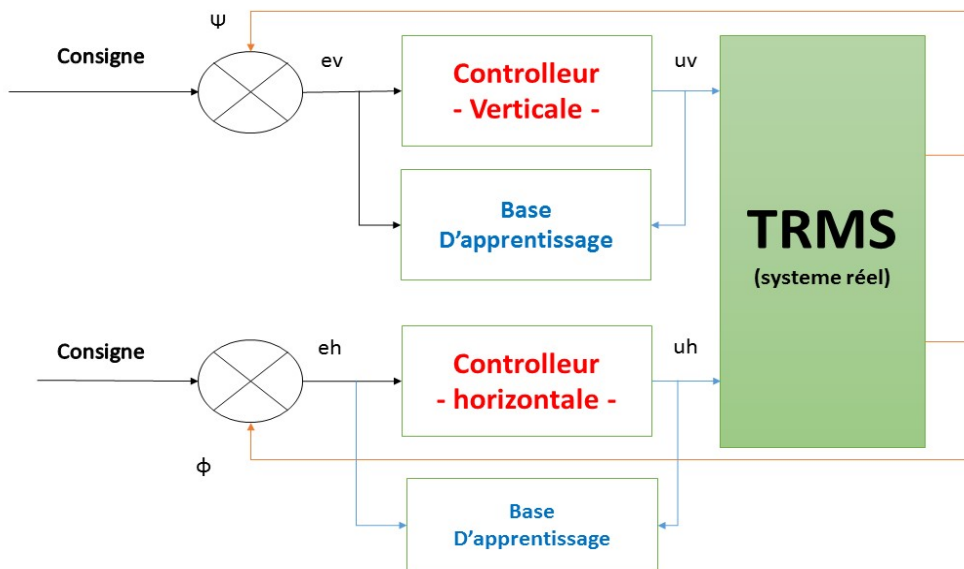


Figure V.11 : schéma d'apprentissage de réseaux neurone pour la commande de TRMS

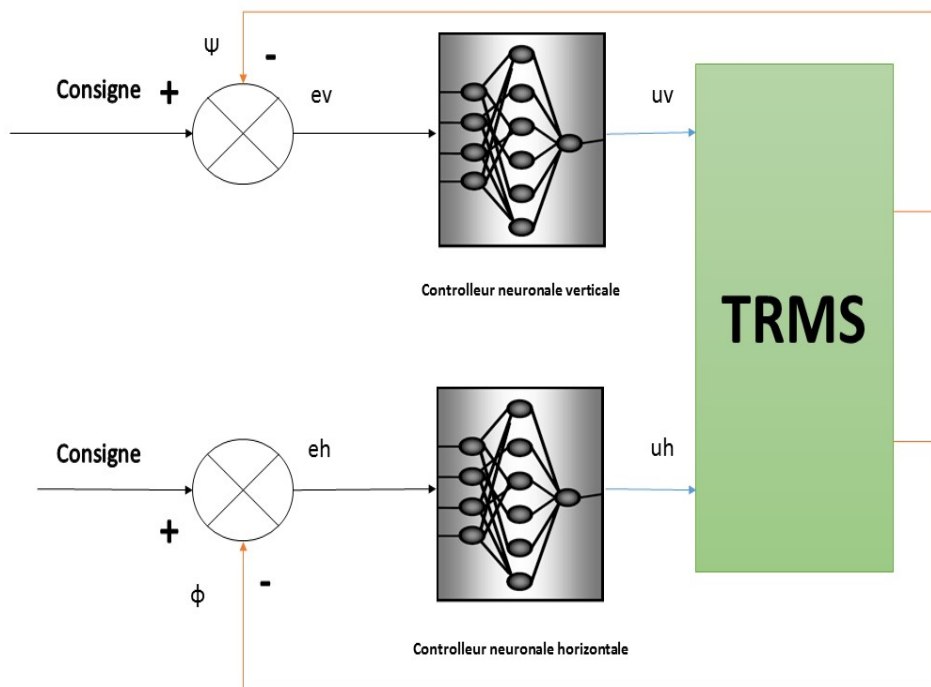


Figure V.12 : schéma de validation la commande réseaux neurone pour TRMS

V.8 Résultats de simulation de la commande par réseaux de neurones au TRMS :

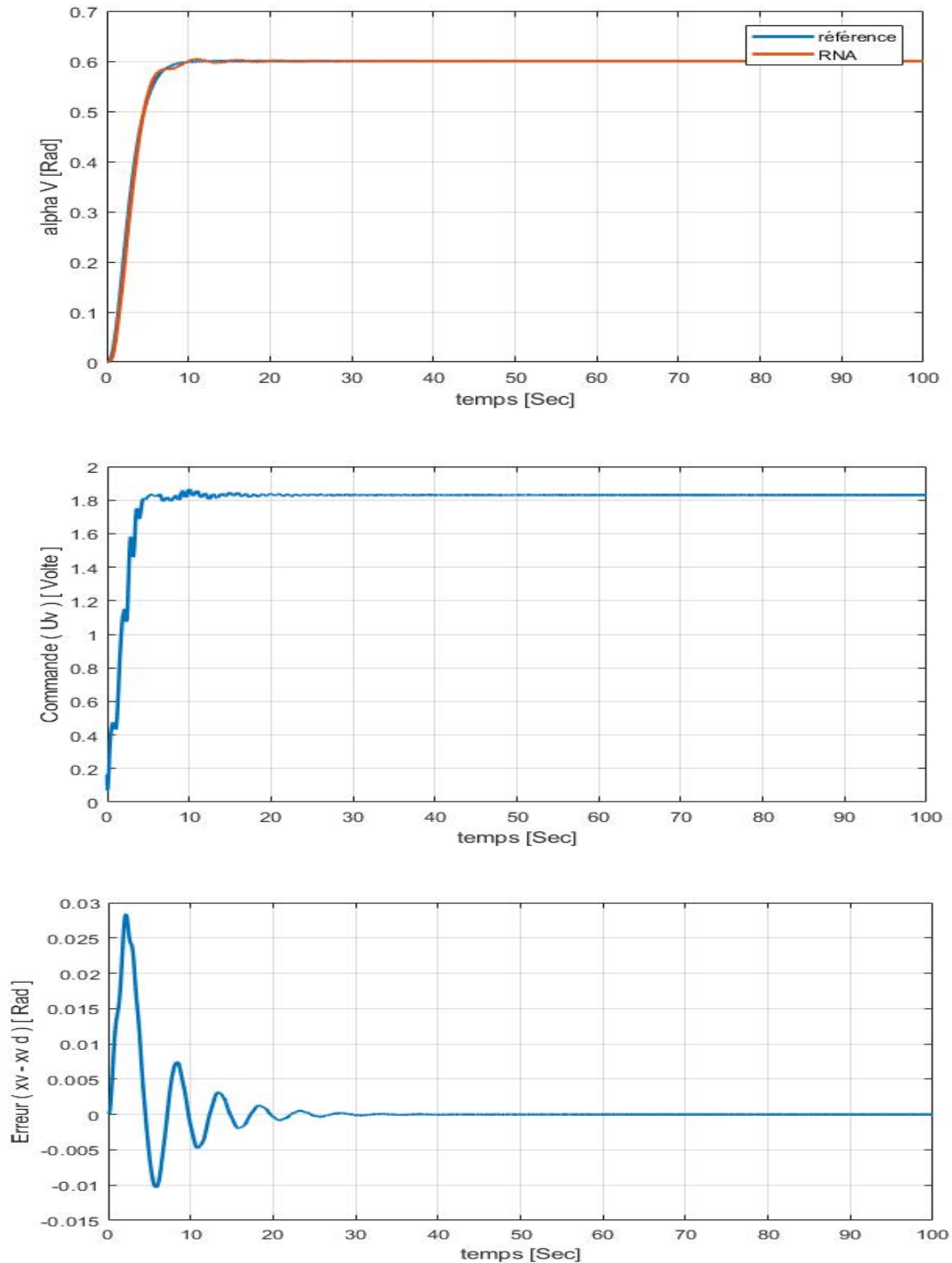


Figure V.13 : Résultats de la commande floue avec une consigne échelon pour sous-système vertical

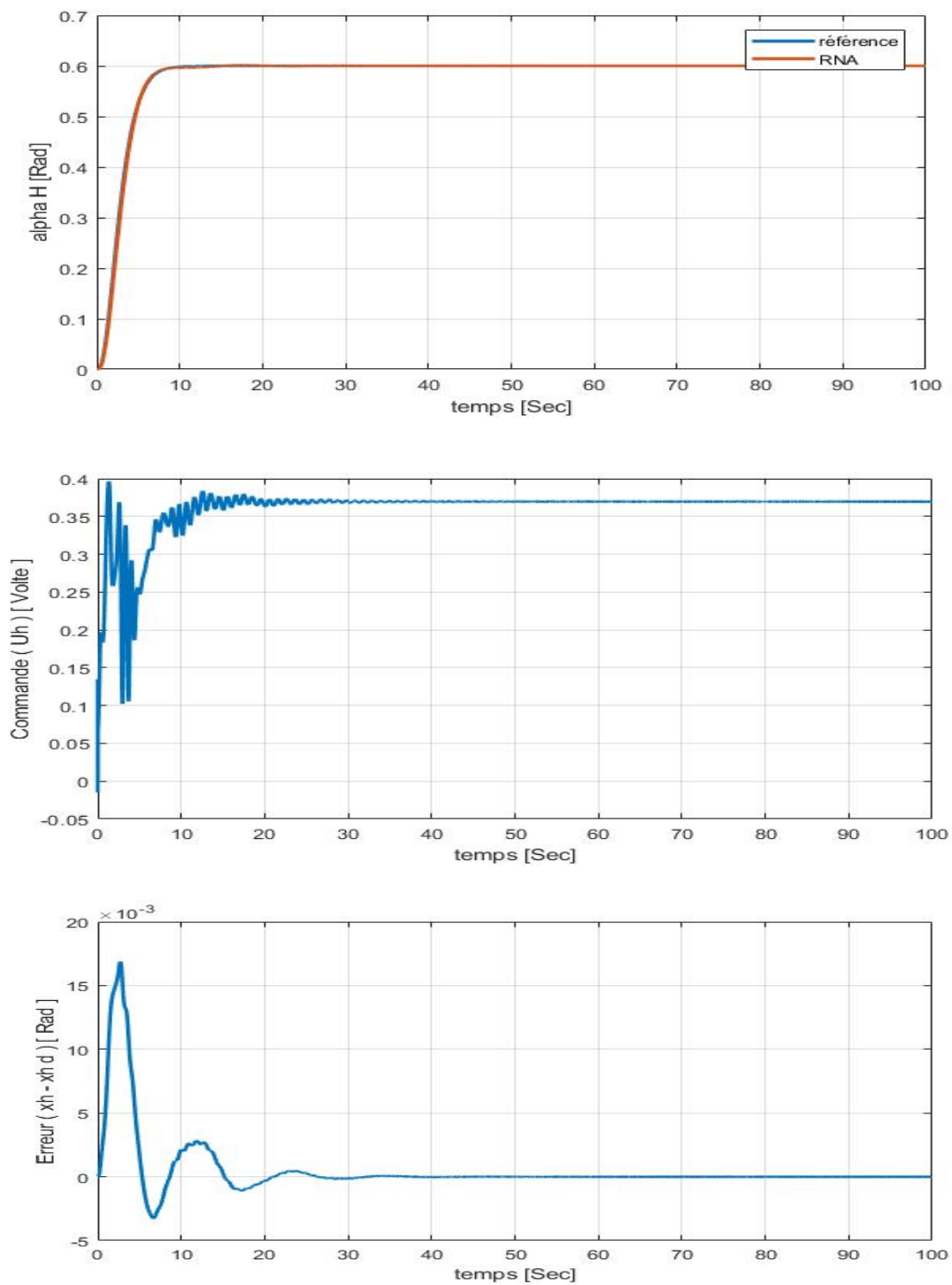


Figure V.14 : Résultats de la commande floue avec une consigne échelon pour sous-système horizontale

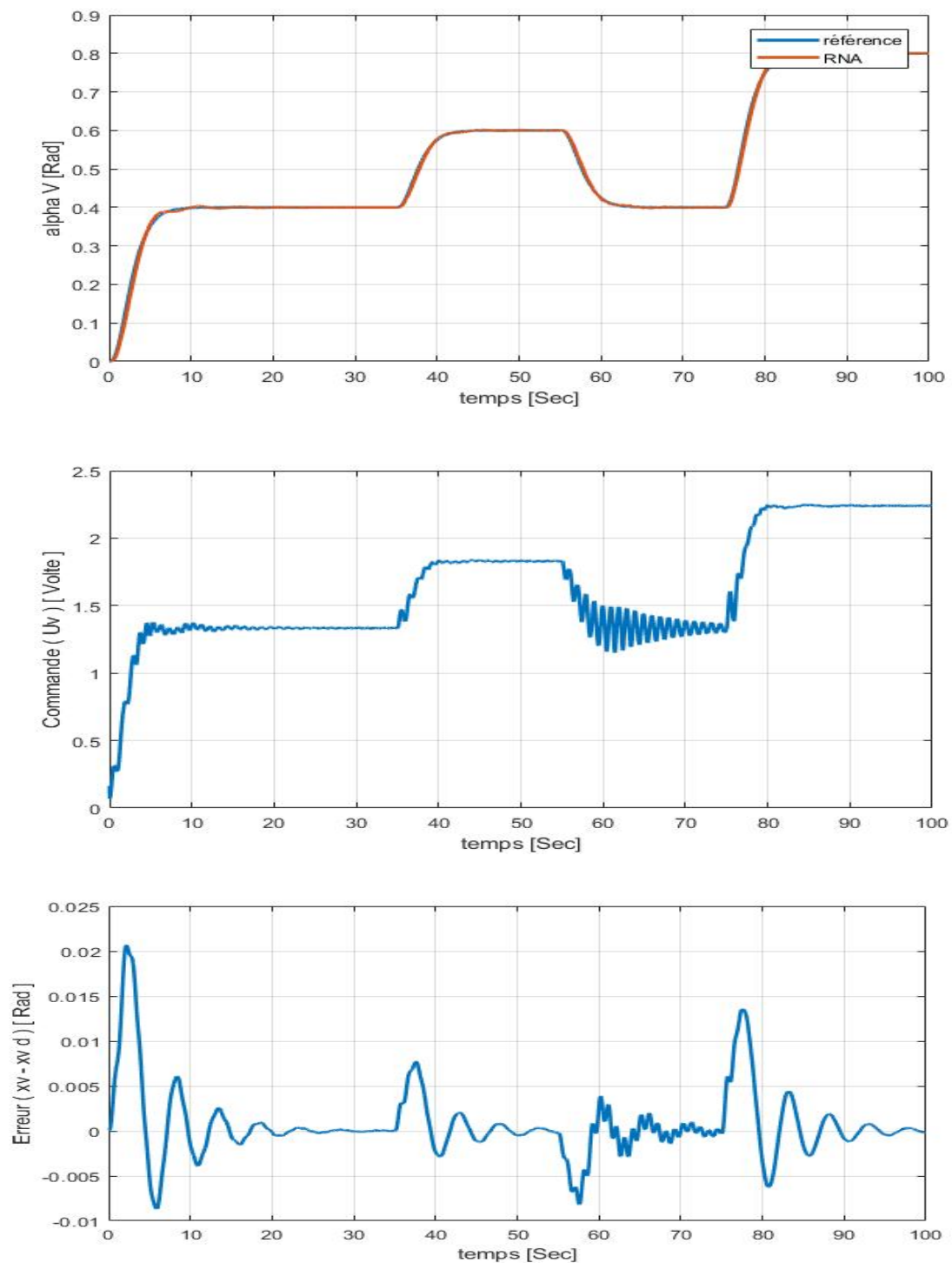


Figure V.15 : Résultats de la commande floue avec une consigne carre pour sous-système verticale

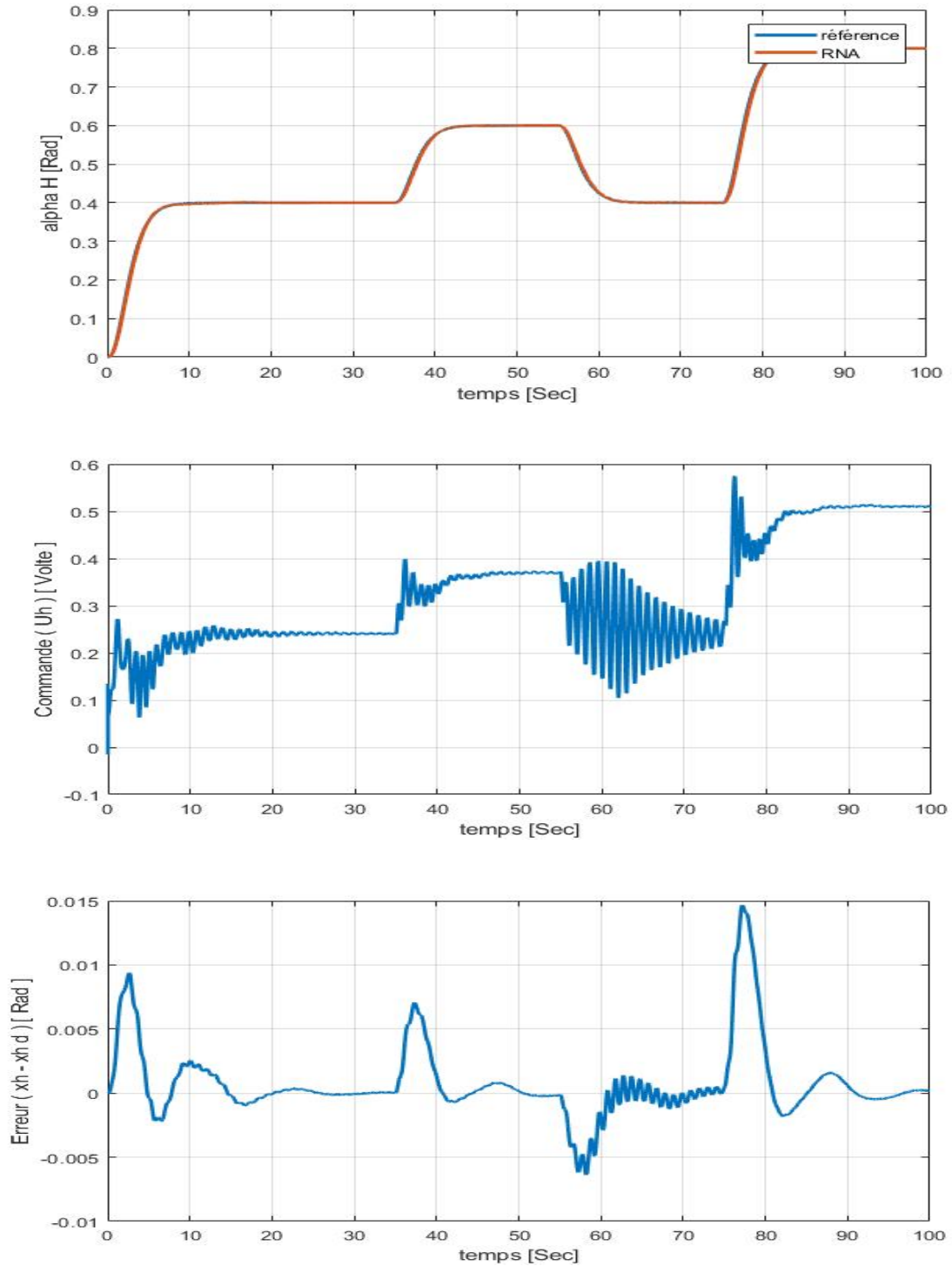


Figure V.16 : Résultats de la commande floue avec une consigne carre pour sous-système horizontale

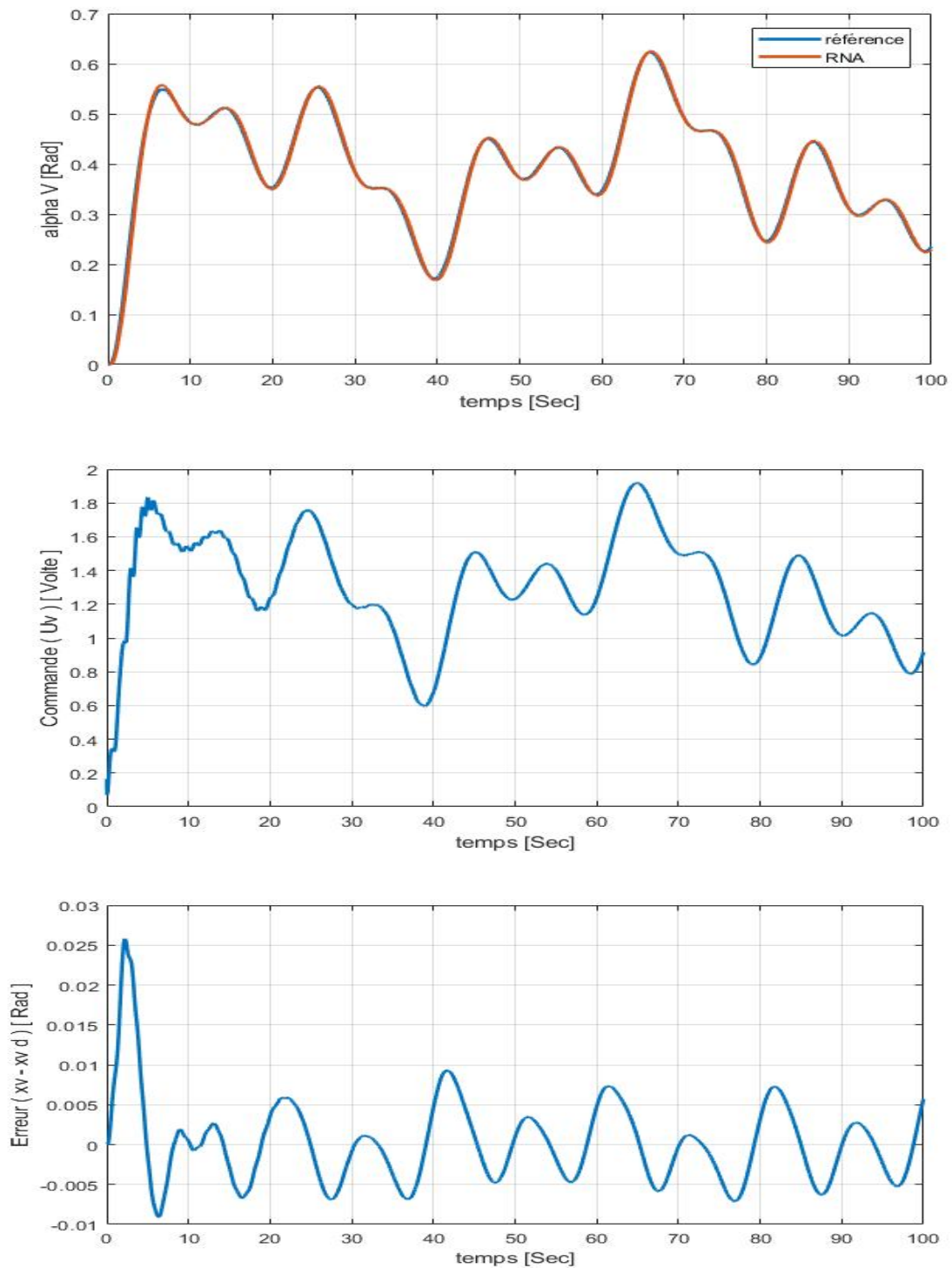


Figure V.17 : Résultats de la commande neuronale avec une consigne sinusoïdale pour sous-système vertical

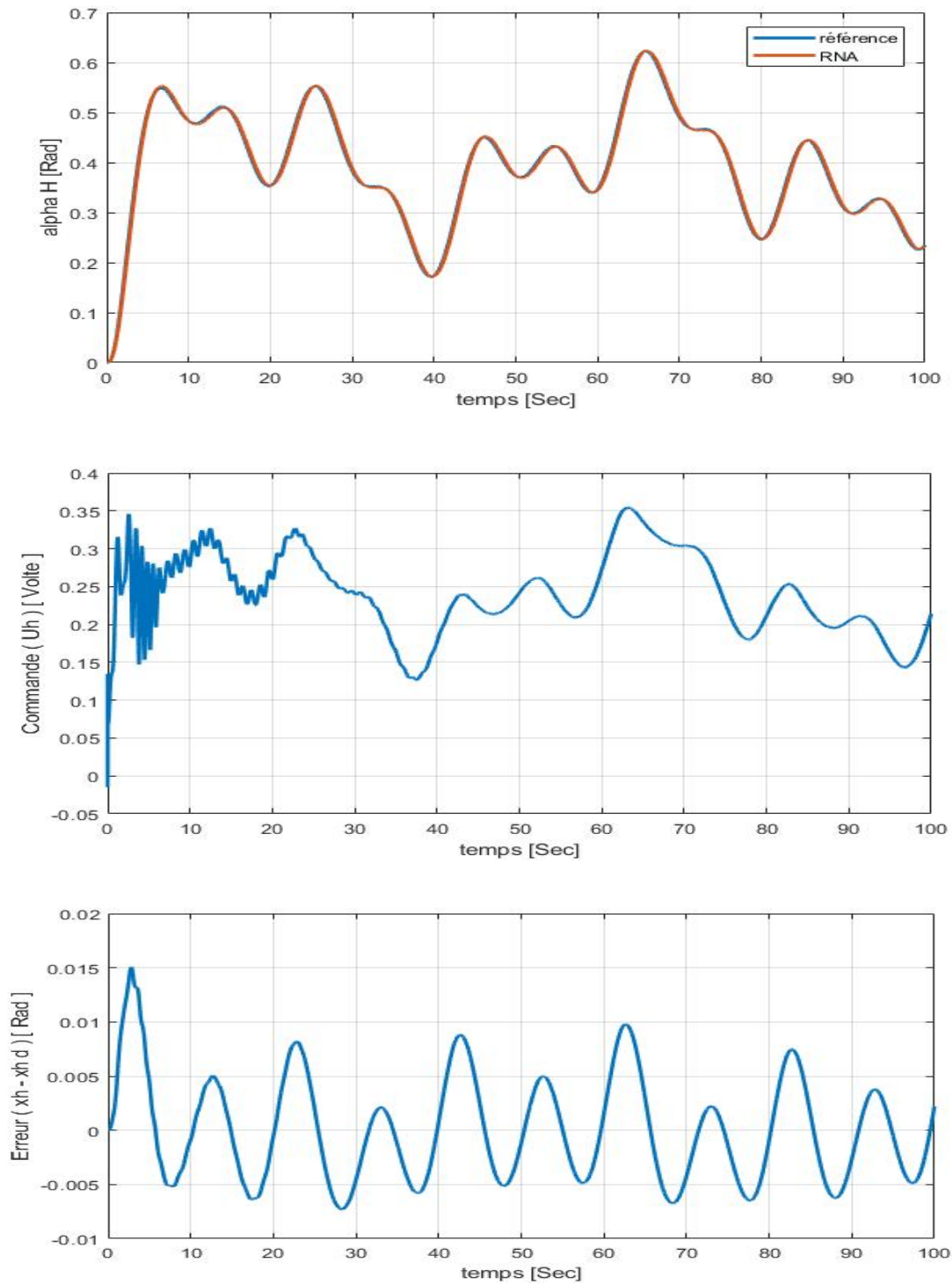


Figure V.18 : Résultats de la commande neuronale avec une consigne sinusoïdale pour sous-système horizontale

V.9 Test de robustesse pour les positions angulaires du TRMS

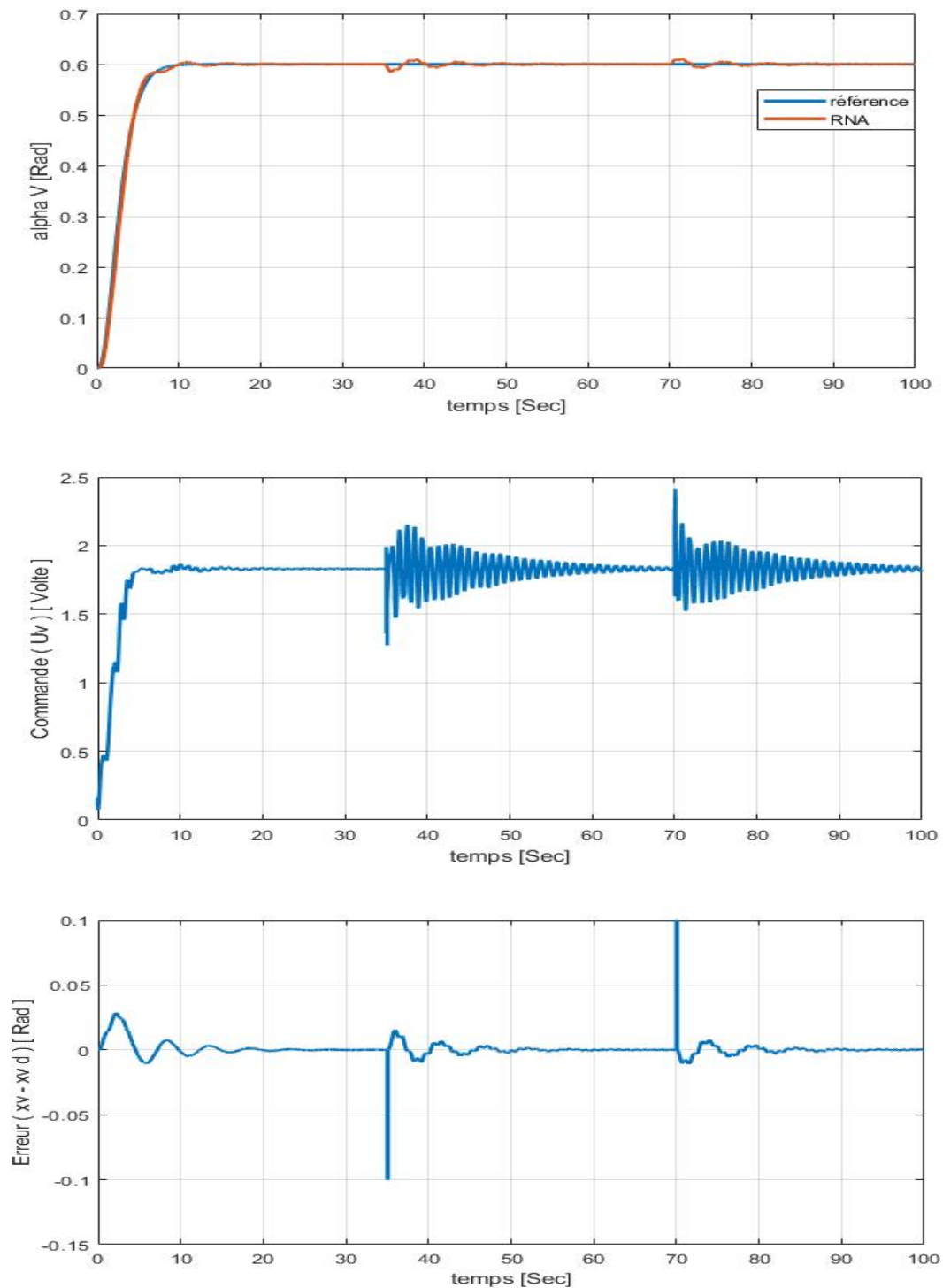


Figure V.19 : Résultats de la commande floue avec une consigne échelon pour sous-système vertical avec perturbation

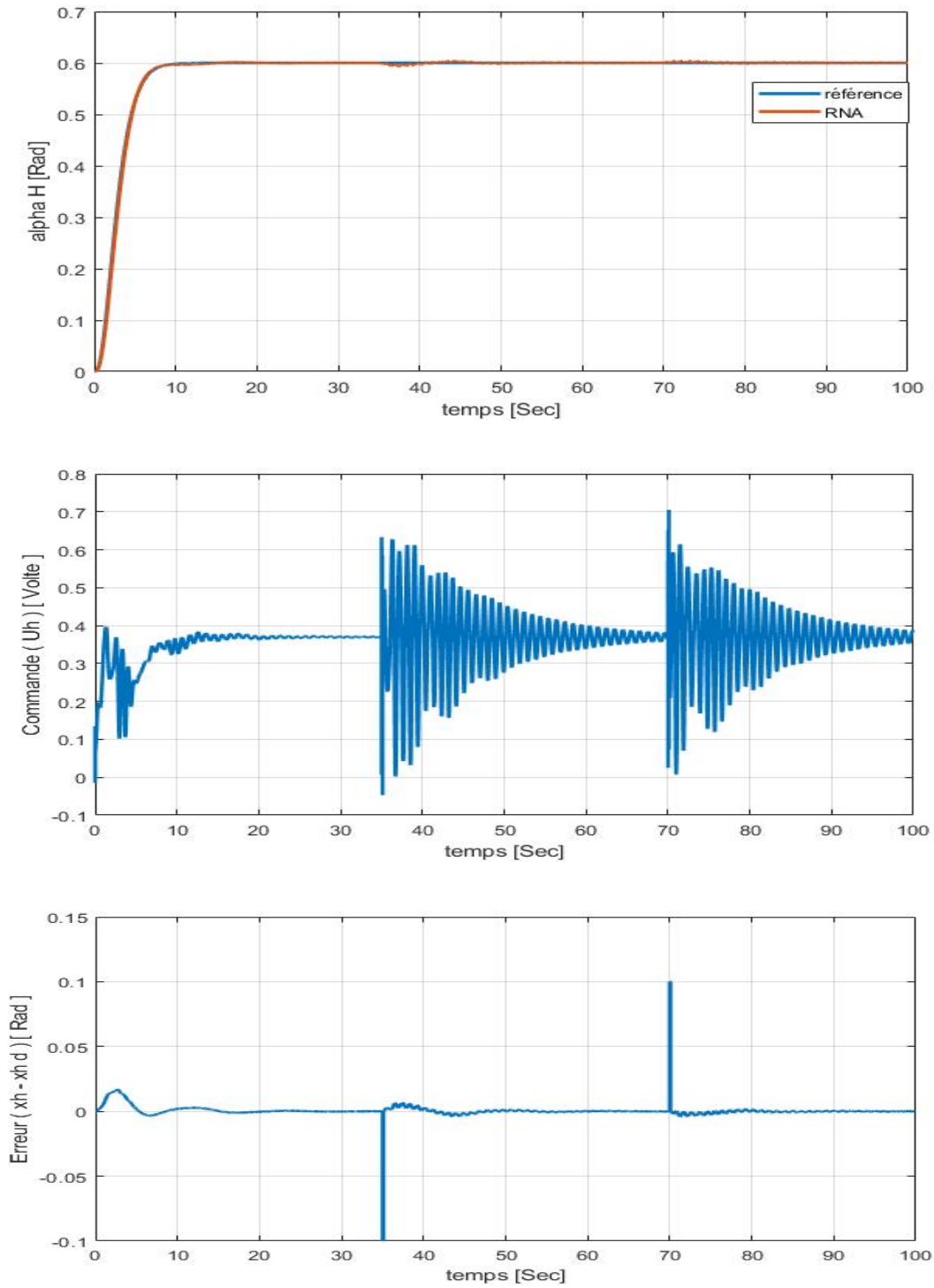


Figure V.20 : Résultats de la commande floue avec une consigne échelon pour sous-système horizontale avec perturbation

V.10 Conclusion :

Dans ce chapitre, nous avons introduit différentes notions théoriques concernant les réseaux de neurones tels que l'architecture, l'activation et l'apprentissage. Chaque réseau est caractérisé par un certain nombre de paramètres (fonction d'activation, connexions des différents neurones du réseau, etc...), en tenant compte de cela, nous aboutissons au réseau de neurones multicouches.

De manière générale les réseaux de neurones présentent un moyen efficace pour résoudre des problèmes pour lesquels, les méthodes classiques ont montré leurs limites, que cela soit en commande, et les résultats obtenus sont appréciables.

CHAPITRE VI

Etude comparative

VI. Etude comparative

VI.1 INTRODUCTION

Dans ce chapitre, Nous étudierons une comparaison entre la commande classique type PID et les commandes intelligentes type logique floue et type réseaux neurone

Cette étude permettra de conclure sur les performances des différentes lois de commande développées pour la commande du TRMS. Elle est basée sur le deux critères suivants : [40]

- L'énergie de la commande.

$$J_1 = \frac{1}{2} \sum_{k=1}^p (\mathbf{u}^T \mathbf{u}) \quad (VI.1)$$

- La somme des carrés des erreurs.

$$J_2 = \frac{1}{2} \sum_{k=1}^p (\mathbf{e}^T \mathbf{e}) \quad (VI.2)$$

VI.2 L'étude comparative

Les mêmes conditions de validation, tels que le pas de validation, la plage temporelle, les références de poursuite, ...etc., ont été utilisées. Les résultats de cette comparaison sont regroupés dans les tableaux VI.1 et VI.2 et VI.3

Tableaux VI.1 : la comparaison entre la commande PID et la commande floue dans le plan vertical. (Ψ)

<i>Critère</i>	<i>Commandes développées pour le TRMS</i>					
	<i>Commande classique</i>			<i>Commande intelligente</i>		
	<i>PID</i>			<i>floue</i>		
	<i>échelon</i>	<i>carré</i>	<i>sinusoïdale</i>	<i>échelon</i>	<i>carré</i>	<i>sinusoïdale</i>
$J_1 = \frac{1}{2} \sum_{k=1}^p (u^T u)$	1283.9	1312.6	806.72	1209.7	1025.6	687.01
$J_2 = \frac{1}{2} \sum_{k=1}^p (e^T e)$	3.0135	3.4250	1.7341	3.4659	2.9880	2.8837

D'après le Tableau VI.1 nous constatons, dans le cas PID VS floue dans le plan vertical, que la valeur la plus faible de critères J_1 est obtenue dans la commande par logique floue, et la valeur la plus faible de critère J_2 est obtenue dans la commande PID.

Tableaux VI.2 : la comparaison entre la commande PID et la commande floue dans le plan horizontal. (φ)

<i>Critère</i>	<i>Commandes développées pour le TRMS</i>					
	<i>Commande classique</i>			<i>Commande intelligente</i>		
	<i>PID</i>			<i>floue</i>		
	<i>échelon</i>	<i>carré</i>	<i>sinusoïdale</i>	<i>échelon</i>	<i>carré</i>	<i>sinusoïdale</i>
$J_1 = \frac{1}{2} \sum_{k=1}^p (u^T u)$	211.30	171.32	88.05	195.32	184.18	94.57
$J_2 = \frac{1}{2} \sum_{k=1}^p (e^T e)$	3.2055	4.6981	4.6784	3.3541	3.9213	2.1240

D'après le Tableau VI.2 nous constatons, dans le cas PID VS floue dans le plan horizontal, que la valeur la plus faible de critères J_1 est obtenue dans la commande PID, et la valeur la plus faible de critère J_2 est obtenue dans la commande par logique floue.

Tableaux VI.3 : la comparaison entre la commande PID et la commande réseaux neurone dans le plan vertical. (Ψ)

<i>Critère</i>	<i>Commandes développées pour le TRMS</i>					
	<i>Commande classique</i>			<i>Commande intelligente</i>		
	<i>PID</i>			<i>Réseaux neurone</i>		
	<i>échelon</i>	<i>carré</i>	<i>sinusoïdale</i>	<i>échelon</i>	<i>carré</i>	<i>sinusoïdale</i>
$J_1 = \frac{1}{2} \sum_{k=1}^p (u^T u)$	1283.9	1312.6	806.72	16329	13936	8919.5
$J_2 = \frac{1}{2} \sum_{k=1}^p (e^T e)$	3.0135	3.4250	1.7341	0.0826	0.0825	0.1453

D'après le Tableau VI.3 nous constatons, dans le cas PID VS floue dans le plan vertical, que la valeur la plus faible de critères J_1 est obtenue dans la commande PID, et la valeur la plus faible de critère J_2 est obtenue dans la commande par réseaux neurone

Tableaux VI.4 : la comparaison entre la commande PID et la commande réseaux neurone dans le plan horizontal. (φ)

<i>Critère</i>	<i>Commandes développées pour le TRMS</i>					
	<i>Commande classique</i>			<i>Commande intelligente</i>		
	<i>PID</i>			<i>Réseaux neurone</i>		
	<i>échelon</i>	<i>carré</i>	<i>sinusoïdale</i>	<i>échelon</i>	<i>carré</i>	<i>sinusoïdale</i>
$J_1 = \frac{1}{2} \sum_{k=1}^p (u^T u)$	211.30	171.32	88.05	657.93	585.93	303.68
$J_2 = \frac{1}{2} \sum_{k=1}^p (e^T e)$	3.2055	4.6981	4.6784	0.0313	0.0450	0.1193

D'après le Tableau VI.4 nous constatons, dans le cas PID VS floue dans le plan horizontal, que la valeur la plus faible de critères J_1 est obtenue dans la commande PID, et la valeur la plus faible de critère J_2 est obtenue dans la commande par réseaux neurone

VI.3 Conclusion :

Dans ce chapitre, Nous avons fait la comparaison entre la commande classique type PID et les commande intelligente type FLOUE et type RESEAUX NEURONE.

À travers les résultats de la comparaison, nous avons trouvé les performances des différentes lois de commande développées pour la commande du TRMS (classique et intelligente).

Enfin, nous pouvons dire que la commandes classique type PID et la commande intelligente type FLOUE contrairement aux caractéristiques (si J_1 faible en PID implique J_2 faible en FLOUE et Le contraire est vrai). Et la commandes classique type PID toujours faible en critère J_1 par rapport la commande RESEAUX NEURONE, mais la commande RESEAUX NEURONE est toujours faible en critère J_2 par rapport la commande PID.

Conclusion generale :

Le travail effectué dans ce mémoire concerne un thème d'actualité dans le domaine de la commande des systèmes non linéaires. Nous sommes intéressés à la comparaison entre la commande classique de type PID et les commandes intelligentes de types LOGIQUE FLOUE et de type RESEAUX NEURONE ARTIFICIEL pour conclure la meilleure performance de la commande.

Alors ce travail présenté qui se divise en six chapitres Représente la structure et la synthèse de la loi de commande et les avantages et les inconvénients et les performances de chaque type de commande A étudié.

Le premier chapitre nous présentons le principe du vol d'un hélicoptère réel. Ensuite nous avons présenté une brève description du simulateur (TRMS). Ainsi que son principe de fonctionnement. Enfin, nous avons expliqué l'environnement de développement utilise.

Dans le deuxième chapitre, nous avons procédé par la modélisation analytique pour avoir un modèle mathématique du TRMS, un modèle qui est non linéaire et couplé. Bien que nous avons essayé d'adapter le modèle au système réel, une erreur de modélisation due essentiellement aérodynamique négligées persiste.

Dans le troisième chapitre, nous avons vu la théorie de la commande par le RÉGULATEUR PID et nous avons faites les Résultat de validation de commande PID en système réel avec des différentes consignes, en fin voire le test de robustesse de la commande avec une consigne échelon.

Dans le quatrième chapitre, nous avons vu la théorie de la commande floue et le principe de fonctionnement. Ainsi, nous avons cité les avantages et les inconvénients de cette technique de commande, nous avons faites les Résultat de validation de commande par logique floue en système réel avec des différentes consignes, en fin voire le test de robustesse de la commande avec une consigne échelon.

Dans le cinquième chapitre, nous avons introduit différentes notions théoriques concernant les réseaux de neurones tels que l'architecture, l'activation et l'apprentissage. Chaque réseau est caractérisé par un certain nombre de paramètres.

Dans le sixième chapitre, nous avons fait la comparaison entre la commande classique type PID et les commande intelligente type FLOUE et type RESEAUX NEURONE. À travers les résultats de la comparaison, nous avons trouvé les performances des différentes lois de commande développées pour la commande du TRMS (classique et intelligente).

Enfin, nous pouvons dire que la commandes classique type PID et la commande intelligente type FLOUE contrairement aux caractéristiques (si J1 faible en PID implique J2 faible en FLOUE et Le contraire est vrai). Et la commandes classique type PID toujours faible en critère J1 par rapport la commande RESEAUX NEURONE, mais la commande RESEAUX NEURONE est toujours faible en critère J2 par rapport la commande PID.

BIBLIOGRAPHIE :

BIBLIOGRAPHIE :

- [1] **T. Hamel, R. Mahony, R. Lozano and J.P. Ostrowski** "Dynamic modeling and configuration stabilization for an X4-flyer", Proceedings of the Triennial World Congress on Automatic Control, Barcelona, Spain, 21-26 Jul 2002, pp. 846-851.
- [2] **Alouani . F** « Commande par Logique Floue Appliquée aux pendules inversés et au simulateur de Vol d'Hélicoptère : Simulation et Expérimentation », mémoire de magister, Faculté des Sciences de l'Ingénieur, Université de Mohamed Boudiaf de M'sila (2006).
- [3] **Bouarroudj. N** « Hybridation des Techniques de Commande Floue Mode de Glissement Optimisées par Algorithme d'Essaim de Particules. Application au système TRMS », Département d'automatique ; mémoire de magister ; option : automatique des systèmes Industriels, école nationale polytechnique laboratoire de commande des processus; Année (2010/2011).
- [4] **A. Rahideh, M.H. Shaheed, H.J.C. Huijberts** « Dynamic modelling of a TRMS using analytical and empirical approaches », Control Engineering Practice 16 (2008) 241–259.
- [5] **T. Peng Wen** « Time Optimal and Robust Control of Twin Rotor System », IEEE International Conference on Control and Automation Guangzhou, CHINA - (May 30 to June 1, 2007).
- [6] **A. Rahideh, M. H. Shaheed, A. H.Bajodah** « Adaptive Nonlinear Model Inversion Control of a Twin Rotor System Using Artificial Intelligence ».
- [7] **A.Hacen et B.Leila** « Synthèse et implémentation pratique de lois de commande hybrides multi-contrôleurs. Application au simulateur d'hélicoptère TRMS », Mémoire d'ingénieur ENP (2008).
- [8] **Peng Wen and Te-Wei Lu** « Decoupling Control of a Twin Rotor MIMO System using Robust Deadbeat Control Technique », Faculty of engineering and Surveying, University of Southern Queensland Toowoomba 4350, QLD, Australia.
- [9] **R. KETATA** « Méthodologies de régulation numérique incluant la logique floue » Thèse de Docteur L.A.A.S Toulouse – 1992

BIBLIOGRAPHIE :

- [10] **M. BAUER** « Application de contrôleurs à logique floue pour la commande vectorielle des machines à induction: commande en vitesse et en position », Rapport de stage ingénieur- ESIM-22 Février 1995
- [11] **A. IBALIDEN** « Implantation d'un régulateur de type flou sur des commandes d'onduleurs pilotant des machines alternatives : application à la détermination des correcteurs » - Rapport d'activités 1994-1995
- [12] **B. BEAUFRERE** « Application de la logique floue à la planification de trajectoires de robots mobiles dans des environnements inconnus » - Thèse de Docteur de l'université de Poitiers, Décembre 1994.
- [13] **P. Y. GLORONNEC** « Algorithme d'apprentissage pour systèmes d'inférences floues » - INSA de Rennes (IRISIA)- 1999.
- [14] **M. BAUER** « Application de contrôleurs à logique floue pour la commande vectorielle des machines à induction: commande en vitesse et en position » Rapport de stage ingénieur- ESIM-22 Février 1995
- [15] **B. DEMAYA,** « Commande floue des systèmes à dynamiques complexes- Application à la commande d'un moteur thermique »- Thèse de Docteur LAAS Toulouse - 17 Octobre 1994
- [16] www.helicoptere.net
- [17] **KAFI Mohamed Radouane,** «Commande robuste et non linéaire: Application à un simulateur d'hélicoptère », 1999
- [18] Feedback installation manual, 33-007-0
- [19] REAL TIME WORKSHOP, user's manual
- [20] REAL TIME WINDOWS TARGET, user's manual
- [21] **Hachid.F , Farah.F** « Synthèse et implementation pratique de Lois de commandes non linéaires : Application à un simulateur d'hélicoptère » ,mémoire de fin d'études d'Ingénieur d'état en automatique, Alger , (Juin 2005).
- [22] **Boudia.A** « Commande non Linéaire d'un Hélicoptère à Deux Degré de Liberté (TRMS) Basé sur le PID à Gains Adaptatifs » ; mémoire de master ; Département du Génie Electrique ; Spécialité Automatique ; université Mohamed Boudiaf m'sila ,(2015) .

BIBLIOGRAPHIE :

- [23] All the world's Rotorcraft, www.aviastar.org.
- [24] **Ben Chaabane, A.** « Commandes Hybrides Neuro -glissantes et Neuro Floues-glissantes Appliquées au Simulateur D'hélicoptère TRMS », Mémoire de Magister, Ecole Nationale Polytechnique, Algérie, 2011.
- [25] Twin Rotor MIMO System Control Experiments Manual 33-949S, Ed01 122006, Feedback Part No. 1160-33949S, Feedback Instruments Ltd, Park Road, Crow Borough, East Sussex, TN62QR, UK, 2006.
- [26] **M. Fliess et C. Join** : Intelligent PID controllers. *In 16th Mediterrean Conference on Control and Automation*, Ajaccio, France, 2008.
- [27] **Popov V. M.** (1973). *Hyperstability of control systems*. Berlin: Springer-Verlag.
- [28] **J. Kang, W. Meng, A. Abraham et H. Liu**: An adaptive PID neural network for complex nonlinear system control. *Neurocomputing*, 135(0):79{85, 2014.
- [29] **H. Bühler**, « *Réglage par logique floue* », Presses polytechnique et université romandes, 1994.
- [30] **M. Jimshidi**, « *Fuzzy logic and control* », Ed Printice Hall, 1993
- [31] **H. Bühler**, « *Réglage par logique floue* », Presses polytechnique et université romandes, 1994.
- [32] **J.F. Jodouin**, *Les Réseaux de Neurones, Principes et Définitions*, Ed Hermes, (1994).
- [33] **J. Héroult**, *Réseaux Neuronaux et Traitement de Signal, Traité des Nouvelles Technologies*, Ed Technip Paris, (1994).
- [34] **J.A. Freeman and D.M Skapura**, *Neural Networks Algorithms, Applications and programming Techniques*, Addison-Wesley Publishing Company, (1992).
- [35] **D. Hammerstrom**, Working With Neural Networks, IEEE Spectrum, Vol.30, N°.7, pp.46-53, (1993).
- [36] **T. Si Ali , D.Tebani** : «Réalisation de la commande d'un onduleur de tension par réseau de neurons sous environnement MATLAB » ,mémoire de fin du étude U.S.T.Oran 2001 .
- [37] **Freeman**, *Neural Networks*, Ed Addison-Wesley, (1992).

BIBLIOGRAPHIE :

- [38] **Shun-Ichi**, *A Mathematical Foundations of Neurocomputing*, Proc of the IEEE, Vol. 78, no 9, p 1443-1463, (1990).
- [39] **Y. HARKOUSS**, «Application de réseaux de neurones à la modélisation de composants et de dispositifs micro-ondes non linéaires », Thèse de doctorat de l'Université de Limoges, décembre 1998.
- [40] **Zeghlache,S.**, « Commande Non Linéaire d'un Appareil à Vol Vertical », Thèse de Doctorat , Département D'électronique, Université de M'sila, 2014.

Résumé :

Ce travail présente une comparaison entre la commande classique et intelligente utilisés pour la commande d'attitude du système non linéaire tel que TRMS (Twin Rotor MIMO System), en mode couplé. Pour une trajectoire donnée, A cet effet nous développerons en premier lieu la présentation du simulateur d'hélicoptère, ensuite en deuxième phase nous présentons la modélisation du simulateur de vol d'hélicoptère, suivie de quelques simulations en boucle ouverte, troisième phase nous présentons un rappel théorique sur la commande par régulateur PID, suivie les résultat expérimentaux en temps réel puis en quatrième phase nous présentons un rappel théorique sur la commande par la logique floue , suivie les résultat expérimentaux en temps réel puis Le cinquième phase une bref rappelle théorique sur la commande par RESEAUX DE NEURONES suivie de quelques simulations, en fin, Le sixième phase nous avons fait une comparaison entre les commandes classiques et les commandes intelligentes.

Mots clés :

TRMS – commande PID – logique floue – réseaux neurone

*ملخص :

يقدم هذا العمل مقارنة بين التحكم الكلاسيكي والذكي المستخدم وذلك ل السيطرة على النظام اللاخطي في الوضع المزدوج بالنسبة لمسار معين ، لهذا الغرض طورنا هذا العمل ' سنقوم أولاً بتطوير عرض محاكي المروحية ' ثم في المرحلة الثانية نقدم نمذجة محاكي مروحية الطيران ، تليها بعض المحاكاة في حلقة مفتوحة ، المرحلة الثالثة نقدم تذكيراً نظرياً للتحكم PID

تليها النتائج في الوقت الفعلي وفي المرحلة الرابعة نقدم تذكيراً نظرياً على التحكم بواسطة المنطق الغامض ، يليه النتائج في الوقت الفعلي ثم المرحلة الخامسة عبارة عن خلاصة نظرية قصيرة على الأمر من قبل الشبكات العصبية تليها بعض المحاكاة ، في النهاية ، قمنا بالمرحلة السادسة مقارنة بين التحكم الكلاسيكي والذكي

كلمات مفتاحية :

منطق غامض - خلايا عصبية - متحكم PID – ثنائي الدوار