

الجمهورية الجزائرية الديمقراطية الشعبية  
République Algérienne Démocratique et Populaire  
وزارة التعليم العالي و البحث العلمي  
Ministère de L'enseignement Supérieur et de la Recherche Scientifique



Université Mohamed Khider Biskra  
Faculté des Sciences et de la Technologie  
Département de Génie Electrique  
Filière : Automatique  
Option : Automatique Avancée

Réf:.....

Mémoire de Fin d'Etudes  
En vue de l'obtention du diplôme:

**MASTER**

*Thème*

**Commande d'un simulateur d'hélicoptère (TRMS)  
par une approche hybride (Neuro-Floue)**

Présenté par :

**BENFATMA Yassine**

**MANAA Mohamed Salah**

Soutenu le : 04 Juin 2012

Devant le jury composé de :

**Mr KOURD Yahia**

**M.A A**

**Président**

**Mr CHELIHI Abdelghani**

**M.A A**

**Encadreur**

**Mr BOUMEHRAZ Mohamed**

**M.CA**

**Examineur**

**Année universitaire : 2011 / 2012**

الجمهورية الجزائرية الديمقراطية الشعبية  
République Algérienne Démocratique et Populaire  
وزارة التعليم العالي و البحث العلمي  
Ministère de l'enseignement Supérieur et de la Recherche scientifique



Université Mohamed Khider Biskra  
Faculté des Sciences et de la Technologie  
Département de Génie Electrique  
Filière : automatique  
Option : Automatique Avancée

Mémoire de Fin d'Etudes

En vue de l'obtention du diplôme:

**MASTER**

***Thème***

**Commande d'un simulateur d'hélicoptère (TRMS)**

**par une approche hybride Neuro-Floue**

Présenté par :

**BENFATMA Yassine**

**MANAA Mohamed Salah**

Avis favorable de l'encadreur :

**CHELIHI Abdelghani**

*Signature*

**Avis favorable du Président du Jury**

**COURD Yahia**

*Signature*

**Cachet et signature**

الجمهورية الجزائرية الديمقراطية الشعبية  
République Algérienne Démocratique et Populaire  
وزارة التعليم العالي و البحث العلمي  
Ministère de l'enseignement Supérieur et de la Recherche scientifique



Université Mohamed Khider Biskra  
Faculté des Sciences et de la Technologie  
Département de Génie Electrique  
Filière : automatique  
Option : Automatique Avancée

## *Thème :*

**Commande d'un simulateur d'hélicoptère (TRMS)  
par une approche hybride Neuro-Floue**

**Proposé par : CHELIHI Abdelghani**

**Dirigé par : CHELIHI Abdelghani**



# Remerciements

Nous tenons à remercier tout premièrement dieu le tout puissant pour la volonté , la sante et la patience , qu' il nous a donne durant toues ces longues années .

Ainsi, nous, tenons également à exprimer nos vifs remerciement à notre encadreur M<sup>f</sup> CHELIHI A.ELGANI pour avoir d'abord proposée ce thème, pour suivi continuel tout le long de la réalisation de ce mémoire et qui n'a pas cessé de nous donnée ses conseils.

Nous tenons à remercier vivement toutes personnes qui nous ont aidés à élaborer et réaliser ce mémoire, ainsi à tous ceux qui nous ont aidés de prés ou de loin à accomplir ce travail.

Nos remerciements vont aussi à tous les enseignants de département d'automatique qui ont contribué à notre formation par ailleurs, nos remerciements à tous les membres du jury qui ont accepté de juger notre travail.

En fin, nous tenons à exprimer notre reconnaissance à tous nos amis et collègues pour le soutien moral et matériel.....

# Dédicace

*Avant tout, je tien à remercies le bon dieu, et l'unique qui  
m'offre le courage*

*et la volonté nécessaire pour affronter les différentes de la vie,*

*Je dédie ce modeste travail*

*A mes parons feu (ma mère et mon père) .*

*A mes frères : Ammar. Abdelaziz et le petit Akram*

*A mes sœurs : fatiha et samiya*

*A touts ma famille.*

*A mes amis et mes collègues de la promotion 2012*

*« Automatique ».*

*A mes féaux amis (chaque un à son nom).*

*Surtout :*

*Manaa salah.*

*Massa alaa.*

*Tabet alaa.*

*B.yassine*

# Dédicace

*Avant tout, je tien à remercies le bon dieu, et l'unique qui  
m'offre le courage*

*et la volonté nécessaire pour affronter les différentes de la vie,*

*Je dédie ce modeste travail*

*A ma mère.*

*A mon père.*

*A mes frères FOUZI, RACHID, CHOUKRI  
NADIR, SEIF, ACHREF, HIFEM*

*A mes sœur Ahlem farida.*

*A touts ma famille.*

*A mes amis et mes collègues de la promotion  
« Automatique ».*

*A mes féaux amis (chaque un à son nom).*

*Surtout :*

*Benfatma yassine*

*Massa alaa.*

*Tabet alaa.*

*MANAA MOHAMED SALAH*

## Sommaire

### Notation

### Introduction générale

## CHAPITRE I : Présentation du simulateur d'hélicoptère

<b>I.1 Introduction</b> .....	3
<b>I.2 Principe de vol d'un hélicoptère</b> .....	3
<b>I.3 Description du simulateur</b> .....	5
I.3.1 Propulseurs.....	7
I.3.2 Commande des moteurs .....	7
I.3.3 Les encodeurs optiques .....	7
I.3.4 La base du TRMS.....	8
I.3.5 Boîtier marche/arrêt.....	9
I.3.6 La SCSI BOX.....	9
I.3.7 Carte d'acquisition :ADVANTECH PCI 1711 .....	10
<b>I.4 Environnement de développement</b> .....	11
<b>I.5 Conclusion</b> .....	15

## CHAPITRE II : Modélisation du TRMS

<b>II.1 Introduction</b> .....	16
<b>II.2 Modèle non linéaire</b> .....	17
II.2.1 Modélisation du sous système d'élévation.....	17
II.2.1.1 Moment de la gravitation .....	18
II.2.1.2 Moment de la force aérodynamique.....	19
II.2.1.3 Moment de la force centrifuge.....	20
II.2.1.4 Moment de la force de friction.....	21
II.2.1.5 Moment d'inertie .....	21
II.2.2 Modélisation du sous système d'azimut.....	22
II.2.2.1 Influence de l'angle d'élévation sur le moment d'inertie.....	22
II.2.2.2 Influence de l'angle d'élévation sur le bras de levier des forces.....	22
II.2.2.3 Moment de la force aérodynamique.....	23
II.2.2.4 Moment de la force de friction.....	23
II.2.2.5 Moment d'inertie .....	24
II.2.3 Modélisation des propulseurs.....	24
II.2.4 Effet gyroscopique.....	26
<b>II.3 Modèle d'état non linéaire</b> .....	27
<b>II.4 Le modèle découplé</b> .....	29
II.4.1 Modèle 1 DDL vertical.....	29
II.4.2 Modèle 1 DDL horizontal.....	30

<b>II.5 Simulations</b> .....	30
II.5. 1 simulation de modèle du TRMS.....	32
II.5. 2 Résultant de simulation.....	33
<b>II.6 Conclusion</b> .....	35
<b>CHAPITRE III : Théories de la logique floue, les réseaux neurone et Neuro-Floue</b>	
<b>III.1 Introduction</b> .....	36
<b>III.2 la Logique floue</b> .....	36
III.2.1 Introduction .....	36
III.2.2 Généralité sur la logique floue .....	37
III.2.2.2 Opérateur de la logique flo.....	37
III.2.2.2.1 Opérateur OU (l'intersection) .....	37
III.2.2.2.2 Opérateur ET (l'intersection) .....	37
III.2.2.2.3 Opérateurs NON (complément) .....	37
III.2.2.3 Représentation des fonction d'appartenance.....	38
III.2.2.3.1 Fonction triangulaire.....	38
III.2.2.3.2 Fonctions trapézoïdale.....	38
III.2.2.3.3 Fonction gaussienne.....	38
III.2.3 Structure de base d'un contrôleur flou .....	39
III.2.3.1 Base de connaissances .....	39
III.2.3.2 Fuzzification .....	39
III.2.3.3 Règles d'inférence floue .....	40
III.2.3.4 Défuzzification .....	40
III.2.3.4.1 Méthode du maximum .....	40
III.2.3.4.2 Méthode de la moyenne du maximum.....	40
III.2.3.4.3 Méthode du centre de gravité.....	40
<b>III.3. Les réseaux de neurone</b> .....	41
III.3.1 Introduction .....	41
III.3.2.Neurone biologique.....	41
III.3.3.Neurone formel.....	42
III.3.4.Les architectes neuronaux.....	43
III.3.4.1.Les réseaux de neurones non bouclés .....	44
III.3.4.1.1Réseau non boucles Mono-Couche.....	44
III.3.4.1.2 Les Réseaux non boucles Multi-Couches.....	45
III. 3.4.2. Les réseaux de neurones bouclés.....	45

III.3.4.3. Les réseaux de neurones à connexions complexes.....	46
III.3.5. Processus d'apprentissage.....	46
III.3.5.1. Apprentissage supervisé.....	46
III.3.5.2.Apprentissage non supervisé.....	47
<b>III.4.NEURO-FLOU.....</b>	<b>47</b>
III.4.1. Introduction .....	47
III.4.2. Avantages et inconvénients de la logique floue et des réseaux de neurones....	48
III.4.3. Méthodes neuro-floues.....	49
III.4.3.1. Première méthode neuro-floue.....	49
III.4.3.2. Deuxième méthode neuro-floue.....	50
III.4.3.3. Troisième méthode neuro-floue.....	50
III.4.4 systèmes d'inférence neuro-flou adaptatifs ANFIS.....	51
III.4.4.1 Définition.....	51
III.4.4.2. Les avantages de l'ANFIS.....	51
III.4.4.3. architecture de l'ANFIS.....	52
III.4.4.4. Algorithme d'apprentissage de l'ANFIS .....	55
III.4.4.4.1. Algorithme de Rétro Propagation .....	57
III.4.4.4.2. Algorithme de moindre carré.....	57
<b>III.5.Conclusion.....</b>	<b>59</b>

## **CHAPITRE IV : Commande Neuro-Floue du simulateur d'hélicoptère (TRMS)**

<b>IV.1 Introduction.....</b>	<b>60</b>
<b>IV.2 Architecture du système de commande Neuro-flou (ANFIS) .....</b>	<b>60</b>
IV.2.1 Schéma de commande .....	60
IV.2.2 Conception du contrôleur ANFIS.....	61
IV.2.2.1 Structure .....	61
IV.2.2.2 Apprentissage .....	62
IV.2.2.3. Implémentation .....	63
<b>IV.3 Commande par Régulateur Neuro-Flou du Modèle Vertical (RNFVER).....</b>	<b>64</b>
IV.3.1 Schéma de commande .....	64
IV.3.2 Conception du régulateur RNFVER.....	64
IV.3.3 Simulation et interprétation des résultats.....	67
<b>IV.4 Commande par Régulateur Neuro-Flous du Modèle Horizontal (RNFHOR) ..</b>	<b>70</b>

IV.4.1 Schéma de commande.....	70
IV.4.2 Conception du régulateur RNFHOR.....	70
IV.4.3 Simulation et interprétation des résultats .....	73
<b>IV.4 Commande Neuro-Flous décentralisée du TRMS (modèle couplé).....</b>	<b>76</b>
IV.4 .1.Schéma de commande .....	76
IV.4.3 Simulation et interprétation des résultats .....	77
<b>IV.5 Commande Neuro-Floue 4-RNF du TRMS (modèle couplé).....</b>	<b>82</b>
IV.5 .1 Schéma de commande.....	82
IV.5 .2 Conception des régulateurs RNFVH et RNFHV.....	82
IV.5.3 Simulation et interprétation des résultats .....	85
<b>IV. 6 Conclusion .....</b>	<b>90</b>
<b>Conclusion général</b>	
<b>Annexes A</b>	
<b>Annexes B</b>	
<b>Annexes C</b>	
<b>Bibliographie</b>	

## Notation

$M_{v1}$	: Le moment résultant correspondant aux forces de gravités.
$m_{mr}$	: La masse du rotor principal.
$m_m$	: La masse de la partie principale de la poutre.
$m_{tr}$	: La masse du rotor secondaire.
$m_t$	: La masse de la partie secondaire de la poutre.
$m_{cb}$	: La masse du contrepoids.
$m_b$	: La masse de la poutre du contrepoids.
$m_{ms}$	: La masse de l'hélice principale.
$m_{ts}$	: La masse de l'hélice secondaire.
$l_m$	: La longueur de la partie principale de la poutre.
$l_t$	: La longueur de la partie secondaire de la poutre.
$l_b$	: La longueur de la poutre du contrepoids.
$l_{cb}$	: La distance entre le contrepoids et l'articulation.
$g$	: L'accélération gravitationnelle.
$M_{v2}$	: Le moment de la force aérodynamique développé par le rotor principal.
$\omega_m$	: La vitesse angulaire du rotor principal.
$F_v(\omega_m)$	: Exprime la dépendance de la force aérodynamique de la vitesse angulaire du rotor principal.
$M_{v3}$	: Moment de la force centrifuge.
$\Omega_h$	: Vitesse angulaire de la poutre autour de l'axe vertical
$\alpha_h$	: l'angle d'azimute de la poutre.
$M_{v4}$	: Moment de friction.
$\Omega_v$	: La vitesse angulaire autour de l'axe horizontal
$\alpha_v$	: l'angle d'élévation de la poutre.
$k_v$	: Constante de friction à déterminer expérimentalement (par identification).
$J_v$	: Le moment d'inertie par rapport à l'axe horizontal
$r_{ms}$	: Le rayon de l'hélice principale.
$r_{ts}$	: Le rayon de l'hélice secondaire.
$M_h$	: la somme des moments des forces agissantes dans le plan horizontal
$J_h$	: La somme des moments d'inertie par rapport à l'axe vertical
$\omega_t$	: La vitesse angulaire du rotor secondaire.
$F_h(\omega_t)$	: Exprime la dépendance de la force aérodynamique de la vitesse angulaire du rotor secondaire
$k_h$	: Constante de friction à déterminer expérimentalement par l'identification.
$J_{tr}$	: Le moment d'inertie dans le moteur secondaire.
$J_{mr}$	: Le moment d'inertie dans le moteur principal.
$\omega$	: Vitesse angulaire du moteur
$u$	: Tension de commande
$I$	: Moment d'inertie.
$R$	: Résistance de l'armature.
$K_b$	: Constante de la FEM
$K_i$	: Constante du couple
$T_L$	: Couple résistant généré par la charge.

---

$T_{mr}$	: La constante du temps du moteur principal.
$T_{tr}$	: La constante du temps du moteur secondaire.
$K_{mr}$	: Le gain statique du moteur principal.
$K_{tr}$	: Le gain statique du moteur secondaire.
SIF	: système d'inférences floues
ANFIS	: Adaptive Network Based Fuzzy Inference System
RNF	: Régulateur Neuro-Flou
RNFVER	: Régulateur Neuro-Flou du Modèle Vertical
RNFHOR	: Régulateur Neuro-Flou du Modèle Horizontal
MIMO	: Multiple input Multiple output
SISO	: signal input signal output
TRMS	: Twin Rotor MIMO System



<b>Figure III.12</b> : apprentissage non supervisé.....	47
<b>Figure III.13</b> : Principe du Neuro-flou .....	48
<b>Figure III.14</b> : Premières architecture des réseaux neuro-floues .....	49
<b>Figure III.15</b> : Troisième architecture des réseaux Neuro-Flou Réalisation en série.....	50
<b>Figure III.16</b> : Troisième architecture des réseaux Neuro-Flou Réalisation en parallèle.....	51
<b>Figure III.17</b> : L'Architecture générale de l'ANFIS.....	52
<b>Figure III.18</b> : architecture d'un ANFIS à 2 entrées avec 9 règles.....	54
<b>Figure III.19</b> : illustration de méthode d'apprentissage Hybride .....	56
<b>Figure III.22</b> : Archetière de contrôleur ANFIS exploité sur la boîte à outil de logiciel MATLAB.....	59
<b>Figure IV.1</b> : Schéma de commande par un régulateur neuro-flous (ANFIS) .....	60
<b>Figure IV.2</b> : Architecture du modèle ANFIS .....	61
<b>Figure IV.3</b> : Schéma d'apprentissage du régulateur neuro-flous (ANFIS).....	63
<b>Figure IV.4</b> : Organigramme de conception d'un système neuro-flou sous MATLAB .....	63
<b>Figure IV.5</b> : Schéma de commande de la partie verticale du TRMS par un régulateur neuro-flou RNFVER .....	64
<b>Figure IV.6</b> : Les fonctions d'appartenances de l'erreur $e_v$ et la dérivée de l'erreur $de_v$ .....	65
<b>Figure IV.7</b> : signal de commande avant et après apprentissage du RNFVER .....	66
<b>Figure IV.8</b> : la courbe en 3D de la sortie $u_v$ du régulateur en fonction des deux entrées $e_v$ et $de_v$ du contrôleur RNFVER .....	66
<b>Figure IV.9</b> : Résultats de simulation de la commande RNFVER pour une référence à signal carré d'amplitude $\pi/6$ .....	68
<b>Figure IV.10</b> : Résultats de simulation de la commande RNFVER pour une référence sinusoïdale d'amplitude $\pi/6$ .....	68
<b>Figure IV.11</b> : Résultats de simulation de la commande RNFVER pour une référence à signal carré d'amplitude $\pi/6$ avec perturbation de (0.1rad) à l'instant ( $t=160$ sec).....	69
<b>Figure IV.12</b> : Résultats de simulation de la commande RNFVER pour Une Référence sinusoïdale d'amplitude $\pi/6$ avec Perturbation de (0.1rad) à l'instant ( $t=160$ sec) .....	69
<b>Figure IV.13</b> : Schéma de commande de la partie horizontale du TRMS par un régulateur neuro-flou RNFHOR .....	70
<b>Figure IV.14</b> : Les fonctions d'appartenances de l'erreur $e_h$ et la dérivée de l'erreur $de_h$ ...	71
<b>Figure IV.15</b> : signal de commande avant et après apprentissage du RNFHOR.....	72
<b>Figure IV.16</b> : la courbe en 3D de la sortie $u_h$ du régulateur en fonction des deux entrées $e_h$ et $de_h$ du contrôleur RNFHOR .....	72
<b>Figure IV.17</b> : Résultats de simulation de la commande RNFHOR pour une référence à signal carré d'amplitude $\pi/6$ .....	74
<b>Figure IV.18</b> : Résultats de simulation de la commande RNFHOR pour une référence sinusoïdale d'amplitude $\pi/6$ .....	74
<b>Figure IV.19</b> : Résultats de simulation de la commande RNFHOR pour une référence à signal carré d'amplitude $\pi/6$ avec perturbation de (0.1rad) à l'instant ( $t=160$ sec).....	75

<b>Figure IV.20</b> : Résultats de simulation de la commande RNFHOR pour une référence sinusoïdale d'amplitude $\pi/6$ avec perturbation de (0.1rad) à l'instant ( $t=160\text{sec}$ ).....	75
<b>Figure IV.21</b> : Schéma de commande décentralisée du TRMS MIMO .....	76
<b>Figure IV.22</b> : Résultats de simulation de la commande décentralisée du TRMS MIMO pour une référence à signal carré d'amplitude $\pi/6$ .....	78
<b>Figure IV.23</b> : Résultats de simulation de la commande décentralisée du TRMS MIMO pour une référence sinusoïdale d'amplitude $\pi/6$ .....	79
<b>Figure IV.24</b> : Résultats de simulation de la commande décentralisée du TRMS MIMO pour une référence à signal carré d'amplitude $\pi/6$ avec perturbation de (0.2rad) à l'instant ( $t=160\text{sec}$ ) .....	80
<b>Figure IV.25</b> : Résultats de simulation de la commande décentralisée du TRMS MIMO pour une référence sinusoïdale d'amplitude $\pi/6$ avec perturbation de (0.2rad) à l' instant ( $t=160\text{sec}$ ).....	81
<b>Figure IV.26</b> : Schéma de commande 4-RNF du TRMS MIMO .....	82
<b>Figure IV.27</b> : Les fonctions d'appartenances de l'erreur ( $e_{hv}$ et $e_{vh}$ ) et la dérivée de l'erreur ( $de_v$ et $de_h$ ) ..	83
<b>Figure IV.28</b> : signal de commande avant et apprêt apprentissage du RNFVH .....	84
<b>Figure IV.29</b> : signal de commande avant et apprêt apprentissage du RNFHV .....	84
<b>Figure IV.30</b> : Résultats de simulation de la commande centralisée du TRMS MIMO pour une référence à signal carré d'amplitude $\pi/6$ .....	86
<b>Figure IV.31</b> : Résultats de simulation de la commande centralisée du TRMS MIMO pour une référence sinusoïdale d'amplitude $\pi/6$ .....	87
<b>Figure IV.32</b> : Résultats de simulation de la commande centralisée du TRMS MIMO pour une référence à signal carré d'amplitude $\pi/6$ avec perturbation de (0.2rad) à l'instant ( $t=250\text{sec}$ ) .....	88
<b>Figure IV.33</b> : Résultats de simulation de la commande centralisée du TRMS MIMO pour une référence sinusoïdale d'amplitude $\pi/6$ avec perturbation de (0.2rad) à l' instant ( $t=250\text{ sec}$ ) .....	89

## RESUME

L'intelligence artificielle par sa définition, elle exprime le savoir faire humain, elle est devenue actuellement un outil précieux de commande dans divers domaines.

Ce mémoire présente la commande d'un simulateur de vol d'un hélicoptère (TRMS) par des Techniques de l'Intelligence Artificielles. Après avoir décrire et présenter le simulateur, nous avons développé la partie modélisation du TRMS qui à permet d'obtenir un modèle multivariable non linéaire sous forme espace d'état. Le découplage de ce dernier nous a conduis à des modèles monovariabiles non linéaires décrivant la partie horizontale et la partie verticale. Après une étude théorique des techniques de l'intelligence artificielle, nous avons élaboré des régulateurs neuro-flous pour les modèles SISO afin de montrer sa robuste vis-à-vis la non linéarité. Nous nous sommes intéressés aussi à la commande décentralisée du modèle couplé du TRMS par les régulateurs neuro-flous SISO proposée comme solution pour compenser les interactions entre les deux parties du TRMS. Ainsi, l'impact des techniques de l'intelligence artificielle les quelles surpassent les limites des techniques classiques et possèdent des caractéristiques essentielles pour l'amélioration des performances de la commande proposée. Les résultats de simulations par Matlab/simulink et des tests de robustesse seront présentés.

**Mots-clés:** commande neuro-floue (ANFIS), Simulateur de vol d'hélicoptère, TRMS, commande décentralisée, système non linéaire.

### ملخص :

الذكاء الاصطناعي من خلال تعريفه، أنها تعرب عن المعرفة البشرية، أصبح الآن أداة قيمة في ميدان التحكم في مختلف المجالات.

هذه الرسالة تعرض التحكم في جهاز محاكاة الطيران لطائرة هليكوبتر (TRMS) بواسطة تقنيات الذكاء الاصطناعي. بعد وصف وتقديم الجهاز، قمنا بدراسة تفصيلية للجهاز التي سمحت لنا الحصول على نموذج متعدد المتغيرات و غير خطي. أدى فصل هذا الأخير إلى الحصول على نماذج وحيدة المتغيرات و غير الخطية تصف الجزء الأفقي والجزء العمودي.

بعد دراسة نظرية للتقنيات الذكاء الاصطناعي، قمنا بوضع أجهزة تحكم العصبية-الغامضة للنماذج وحيدة المتغيرات وذلك بهدف أظهر فاعليته ضد المنظمات غير الخطي. نحن مهتمون أيضا بالتحكم اللامركزي للنموذج الكلي لجهاز (TRMS) بواسطة أجهزة التحكم العصبية-الغامضة ذات المتغير الواحد كحل لمشكل التفاعلات بين شطري الجهاز. وبالتالي، فإن تأثير تقنيات الذكاء الاصطناعي الذي تجاوز حدود التقنيات التقليدية وتمتلك خصائص أساسية لتحسين أداء الرقابة المقترحة. يتم عرض نتائج المحاكاة بواسطة Simulink /Matlab واختبارات متانة.

**كلمات مفتاحية:** التحكم العصبي-الغامض, جهاز محاكاة الطيران لطائرة هليكوبتر, TRMS, التحكم اللامركزي, النظام غير الخطي.

## **INTRODUCTION GENERALE**

Suites aux différents progrès technologiques que connaît le monde dans tous les domaines, et particulièrement dans le domaine de l'aéronautique, l'automatique a connu un essor considérable. L'élaboration de lois de commande de systèmes a fait l'objet de nombreux travaux de recherche. L'application de certaines de ces méthodes à des avions et aux hélicoptères a montré leur efficacité. Cependant, les systèmes à commander devenant de plus en plus complexes, les lois de commande deviennent elles-mêmes complexes et parfois difficiles à mettre en œuvre. Pour cela, des prototypes (ou simulateurs) ont été conçus dans le but de tester de nouvelles lois de commandes, qui ne peuvent être implémentées directement sur les systèmes réels. Le simulateur de vol d'hélicoptère *TRMS* (Twin Rotor MIMO System) est l'un de ces prototypes. 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.

D'un autre côté, l'obtention d'un modèle précis du procédé devient nécessaire. Les méthodes de modélisation traditionnelles s'avèrent souvent incapables de représenter le comportement global d'un système. L'automaticien est alors confronté à un problème de choix pour la construction d'une loi de commande.

La logique floue a été introduite pour approcher le raisonnement humain à l'aide d'une représentation adéquate des connaissances. Son intérêt 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, [7-15]. Cependant, un système flou est difficile à appréhender. Sa commande et son réglage peuvent être relativement long. Il s'agit parfois beaucoup plus de tâtonnement que d'une réelle réflexion. Il manquait donc à la logique floue un moyen d'apprentissage performant pour régler un système flou, c'est *les réseaux de neurones*.

Les réseaux de neurones peuvent fournir une solution intéressante pour des problématiques de contrôle des systèmes non linéaires. En effet, leur utilisation ne nécessite

pas l'existence d'une modélisation formelle de ces systèmes. Par ailleurs, leurs capacités de mémorisation, d'apprentissage, d'adaptation et le parallélisme du calcul représentent des fonctions très utiles à tout système complexe, [20-21].

Les réseaux neuro-flous sont nés de l'association des réseaux de neurones avec la logique floue, de manière à tirer profits des avantages de chacune de ces deux techniques. La principale propriété des réseaux neuro-flous est leur capacité à traiter dans un même outil des connaissances numériques et symboliques d'un système. Ils permettent donc d'exploiter les capacités d'apprentissage des réseaux de neurones d'une part et les capacités de raisonnement de la logique floue d'autre part, [26]. Différentes combinaisons de ces deux techniques d'intelligence artificielle existent et mettent en avant des propriétés différentes.

Dans le cadre du présent travail, il s'agit de s'intéresser à la commande du simulateur *TRMS* par une approche neuro-floue dans le but de commander les deux angles (élévation et orientation) via des modèles découplés non linéaire représentant la partie verticale et la partie horizontale.

Le contenu de ce mémoire est structuré en quatre chapitres :

Dans le **premier chapitre**, nous présentons le système en question, ses constituants, son principe de fonctionnement, et l'environnement de développement.

Le **second chapitre** sera consacré à une modélisation analytique du TRMS, suivie de quelques simulations en boucle ouverte.

Dans le **troisième chapitre**, nous présentons ce chapitre sur trois sous titre principale les réseaux de neurone et la logique floue et en fin la théorie hybride entre les deux Neuro-Flou précisément ANFIS.

Le **quatrième chapitre**, concerne la synthèse et l'application des lois de commande sur Matlab/Simulink. Les résultats de simulation seront présentés puis discutés selon leurs performances et robustesses.

Ce travail sera clôturé par une conclusion générale.

# chapitre

## I

### Présentation du simulateur d'hélicoptère

Chapitre  
I***Présentation du simulateur d'hélicoptère*****I.1 Introduction**

Les systèmes aéronautiques relèvent de processus suffisamment complexes et onéreux au point où il n'est plus possible d'implémenter des commandes directement sur ceux-ci. D'un autre côté, le développement de l'électronique de puissance permet l'élaboration de systèmes miniatures. En effet, plusieurs prototypes sont mis en œuvre pour tester les commandes avant de les transférer sur le système réel. Le simulateur d'hélicoptère (TRMS) est l'un de ces prototypes qui supporte l'implémentation des diverses commandes. Il sera donc l'objet de notre travail.

Ce chapitre abordera un petit aperçu sur le principe du vol des hélicoptères, ensuite seront présentés les détails du simulateur.

**I.2 Principe de vol d'un hélicoptère**

Essayons de simplifier le principe de vol d'un hélicoptère en nous référant à l'avion. Dans un avion, la traction est assurée par les hélices qui font prendre de la vitesse à l'avion. Quant à la portance, ce sont les ailes qui, par leurs formes aérodynamiques, créent une force proportionnelle (au carré généralement) de la vitesse de traction provoquant, ainsi, l'envol de l'avion, dès que sa vitesse est suffisamment importante.

Pour un hélicoptère, c'est le rotor qui joue à la fois le rôle de l'aile et de l'hélice. Il assure, ainsi, traction et portance.

Ainsi, Pour changer de direction l'hélicoptère doit changer l'angle d'attaque de son rotor pour créer une portance dans la direction voulue.

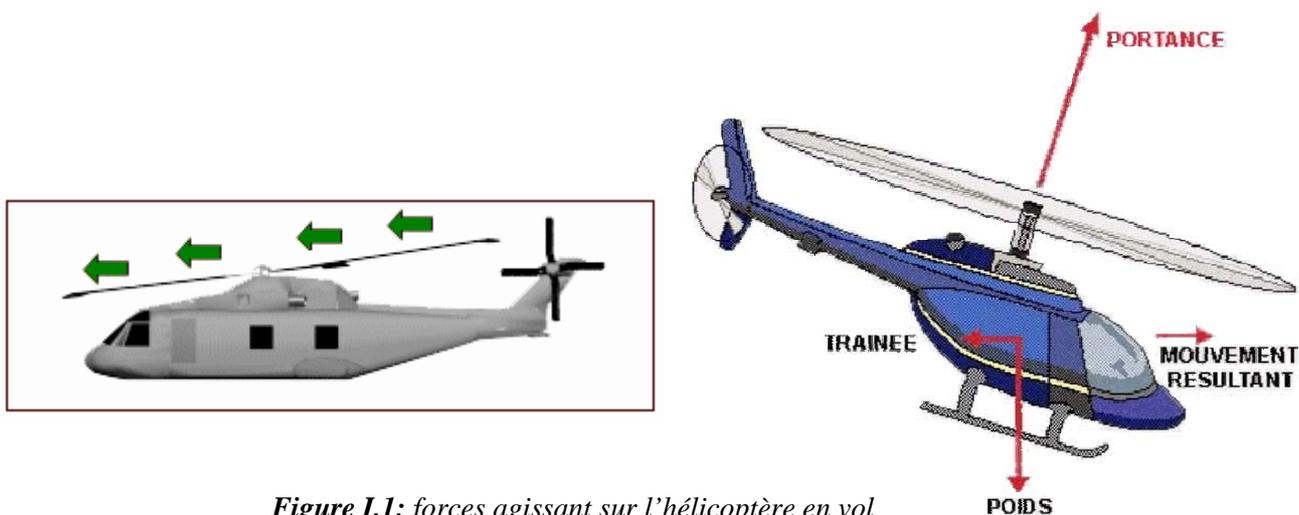


Figure I.1: forces agissant sur l'hélicoptère en vol

Le rotor est entraîné en rotation par un groupe motopropulseur, constitué d'un moteur (à pistons ou turbine à gaz), d'un embrayage et d'un réducteur aussi appelé BTP (boîtier de transmission principal).

Cependant, comme pour l'hélice, le rotor exerce sur l'air un couple dû à son mouvement. L'air en retour (principe d'action-réaction), exerce un autre couple d'intensité égale mais de sens opposé. Imaginez que vous essayez d'introduire une vis encastrée dans un mur extrêmement dur avec une visseuse puissante, vous tourneriez avec la visseuse ! C'est le même phénomène qui est observé avec le rotor principal.

Pour contrer cet effet de couple et permettre un vol stable, on dispose à l'arrière des hélicoptères un "rotor de queue". Son rôle consiste à contrer le couple du rotor principal.

C'est le système le plus couramment utilisé actuellement mais il existe d'autres systèmes qui jouent le même rôle tel que : jet d'air, fenestron, deux rotors principaux (ce qui annule l'anti-couple), mais aussi deux rotors d'anti-couple.

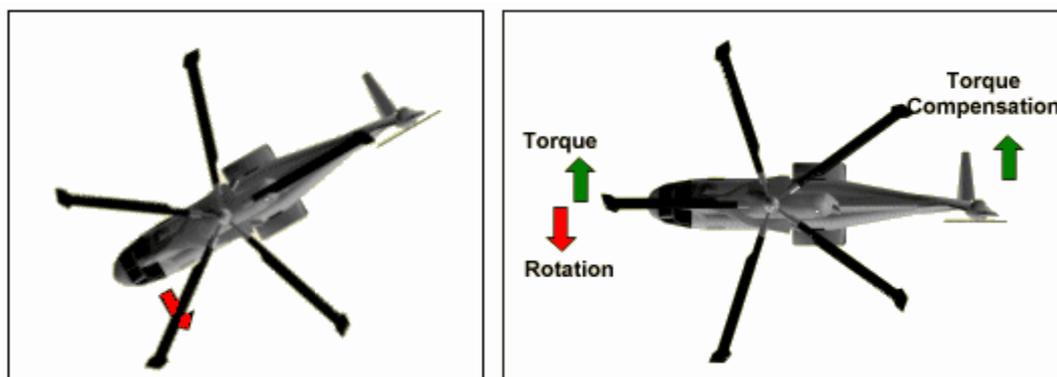


Figure I.2 : Effet de couple et rôle du rotor de queue

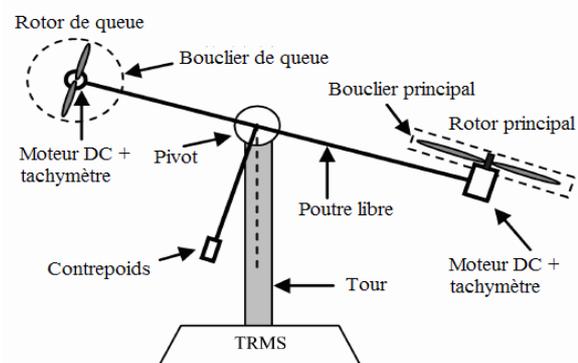
Le rotor anti-couple consomme environ 10 % de la puissance, son interaction sonore avec le rotor principal le rend bruyant, dangereux (extrêmement coupant), sensible au vent de travers, il est également coûteux et fragile, toutes ces raisons motivent les constructeurs à l'éviter ou à l'optimiser.

### I.3 Description du simulateur [1], [2]

Le TRMS (Twin Rotor Mimo System) est un dispositif de laboratoire conçu dans le but d'expérimenter de nouvelles lois de commandes. Son comportement est similaire à celui d'un hélicoptère. Du point de vue commande ; c'est un exemple de système présentant une non linéarité d'ordre supérieur fortement couplé.

Le TRMS est formé de (figure I.3):

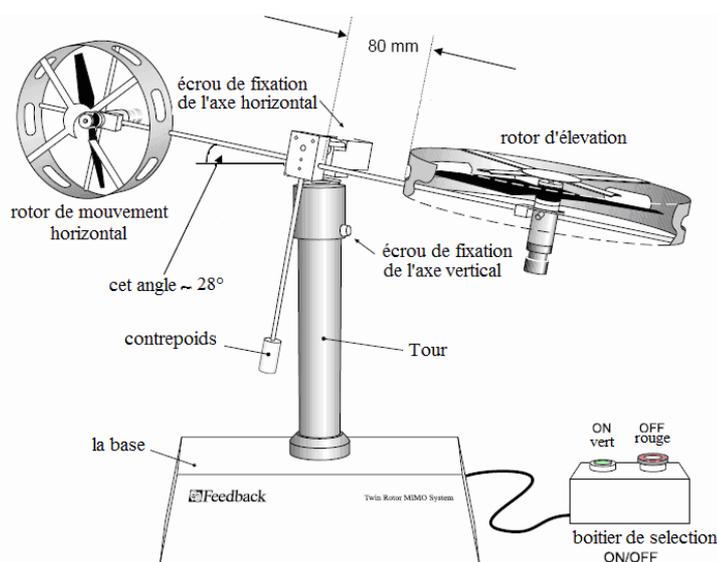
- une poutre pouvant pivoter sur sa base de manière à ce qu'elle puisse tourner librement dans les plans vertical et horizontal ;
- Deux propulseurs (principal et secondaire) fixés aux deux extrémités de la poutre. Ils sont formés d'une hélice, d'un moteur à courant continu et d'un bouclier de sécurité ;
- Un contrepois fixé sur une tige collée au niveau du pivot de la poutre, son rôle est de diminuer les vibrations (oscillations) de cette dernière ;
- Une tour pour maintenir la poutre ;
- Une base comprenant des circuits électronique pour l'adaptation, la synchronisation et le 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)

L'état de la poutre est décrit par 4 variables : les angles horizontal (azimut) et vertical (élévation) mesurés à l'aide d'un capteur se trouvant au niveau du pivot (et qui représente également les sorties du système).

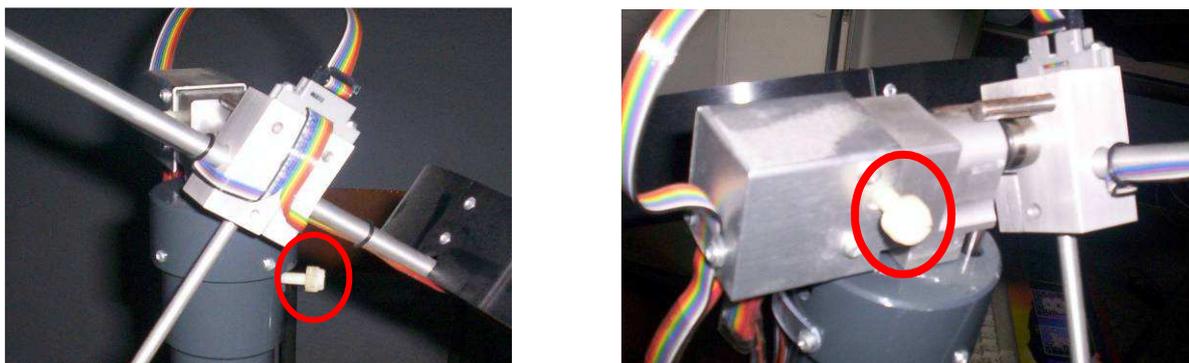
Les deux autres variables sont les vitesses angulaires des rotors, mesurées par des générateurs tachymétrique couplés aux moteurs DC. Elles ne sont, néanmoins, pas utilisées dans les commandes.



**Figure I.4 :** Description du TRMS

Il existe un fort couplage entre les actions des rotors, chaque rotor influence les deux angles.

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 à l'aide de 2 vis. (Figure I.5)



**Figure I.5 :** Vis de fixation (à gauche axe vertical, à droite axe horizontal)

Dans un hélicoptère réel, la force aérodynamique est contrôlée par changement de l'angle d'attaque. Cependant, pour le TRMS, l'angle d'attaque étant fixe, la force

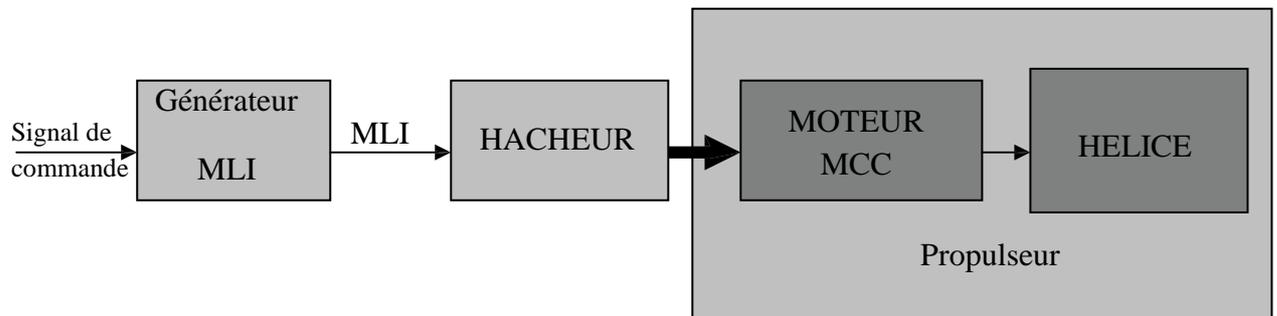
aérodynamique est commandée par changement de la vitesse des rotors. On utilise donc les tensions appliquées aux moteurs comme variable de commande.

### I.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.

### I.3.2 Commande des moteurs

Les moteurs électriques à courant continu offrent l'avantage de présenter des vitesses de rotation proportionnelles 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 produisant aux bornes du moteur une tension de valeur moyenne réglable. Ce 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.6 : schéma synoptique du fonctionnement du propulseur*

### I.3.3 Les encodeurs optiques [3]

Ces capteurs servant à la mesure des angles d'azimut et d'élévation se situent à l'intérieur du pivot sphérique de la poutre. Il s'agit de deux encodeurs optiques incrémentaux dont le principe de fonctionnement est le suivant [figure I.6]:

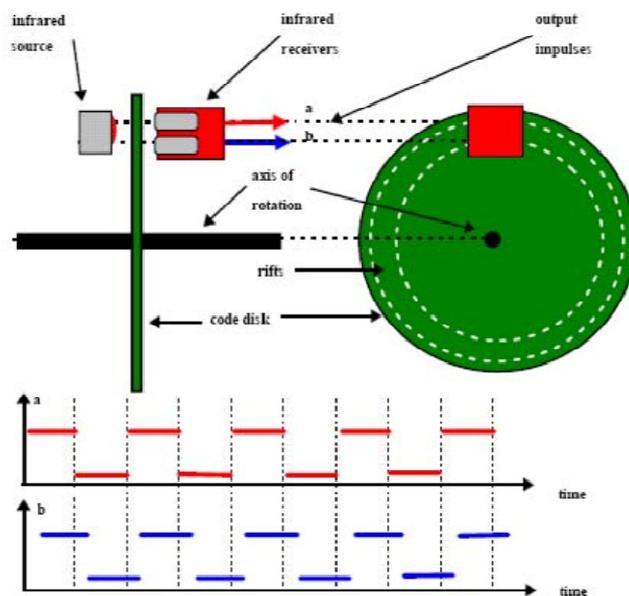


Figure I.7 : encodeurs optiques

Les faisceaux lumineux émis par les deux photodiodes (A et B) passent à travers deux anneaux de fentes sur le disque. Les fentes sont décalées, de sorte que les tensions des récepteurs (A et B) soient des ondes rectangulaires en quadrature de phase. Le dispositif ainsi réalisé permet de déterminer le sens de rotation.

### I.3.4 La base du TRMS [2]

À l'arrière de la base se trouvent les différentes connectiques (figure I.7):

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

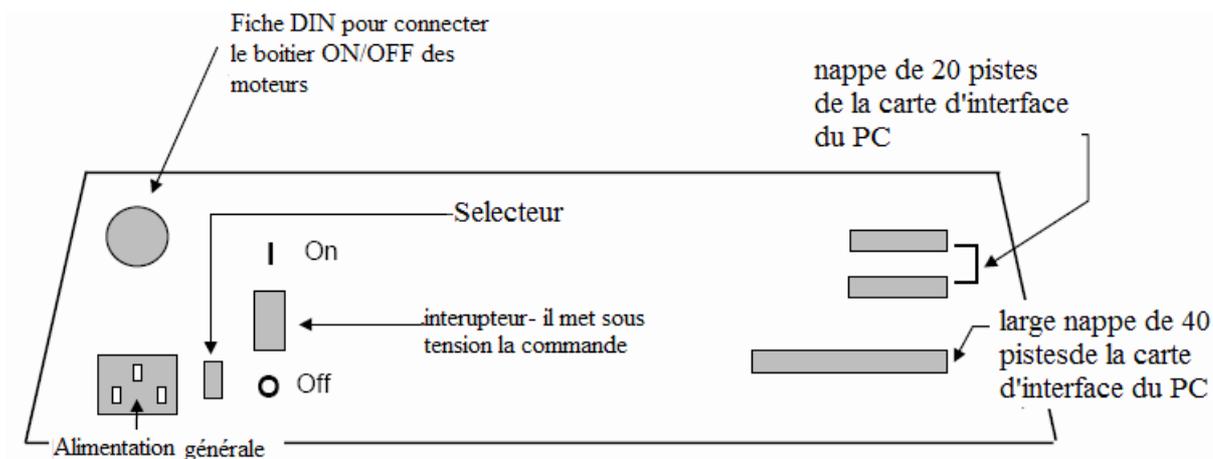
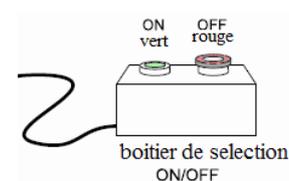


Figure I.8 : La base du TRMS

### I.3.5 Boîtier marche/arrêt

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



### I.3.6 La SCSI BOX

La SCSI BOX (figure I.9.b) a comme rôle d'adapter les signaux entre la carte d'acquisition et les différents connecteurs de la base du TRMS et de 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 commande des moteurs) ;
- un autre petit connecteur PL3 pour les entrées analogiques (retour tachymétrique des deux moteurs).

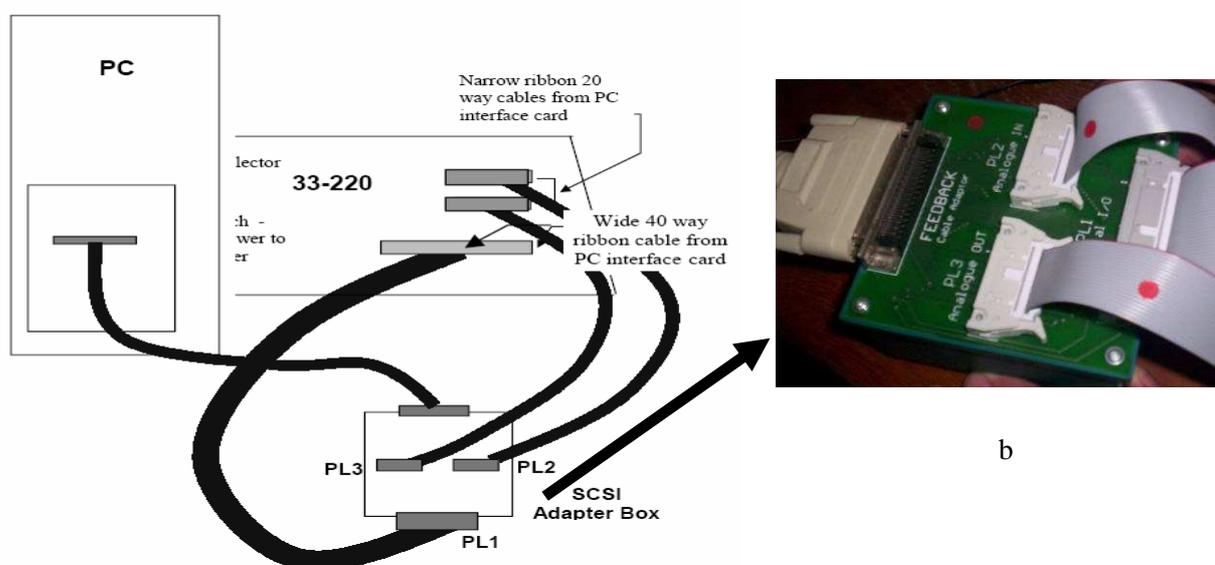


Figure I.9 : Branchement des différents connecteurs

### I.3.7 Carte d'acquisition ADVANTECH PCI 1711 [pci1711]

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 digitales. 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 entrées analogiques programmables est contrôlable par software ;
- Chaque canal dispose de sa gamme individuelle stockée dans la RAM de la carte ;
- 2 sorties analogiques (convertisseurs D/A) ;
- 16 canaux d'entrées digitales ;
- 16 canaux de sorties digitales ;
- Un compteur / timer programmable ;
- Scan automatique des gains /canaux.

Son principal rôle dans notre application est de permettre la commande digitale d'un système continu à travers un PC, en convertissant les signaux analogiques en numériques et

vice-versa. Dans notre cas, on utilise :

- Deux sorties analogiques (convertisseur D/A) pour délivrer les tensions de commande 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.

#### I.4 Environnement de développement [4], [6]

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-In-the-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 provoque des pertes de temps considérables, en testant et en calibrant à nouveau le régulateur implémenté. Voici, ci-dessous, un comparatif succinct entre cette approche et l'approche classique :

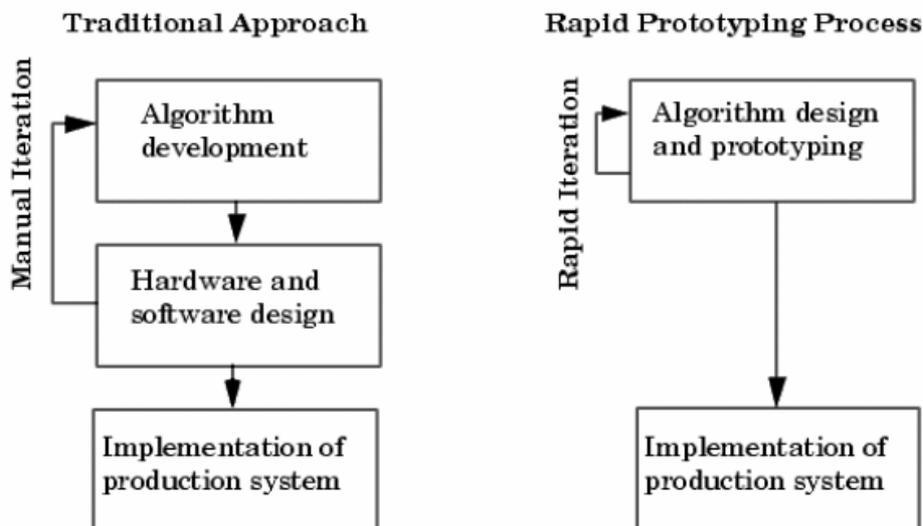


Figure I.10 : Rapide prototypings VS processus de développement traditionnel

Avec le « hardware-inthe-loop », on teste directement le contrôleur synthétisé en simulation sur le système physique réel, à l'aide d'un simple PC de développement. En effet,

sont nécessaire :

- un PC standard (pour le développement et pour la commande) ;
- un carte d'acquisition pour la conversion A/D et D/A ;
- le système physique à commander ;
- les logiciels appropriés.

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, à l'aide duquel on programme avec des objets ou blocks. Ces objets peuvent être soit des block standards ou bien des blocks spécialisées conçus par l'utilisateur (ex: les S-functions) ;
- **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 génère par real time workshop et fait l'édition des liens pour générer un exécutable qui communique avec la cible d'implémentation (target), c'est le compilateur 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. de ce fait, il ne permet pas l'exécution en 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 joue le rôle d'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, également, des bibliothèques, ou drivers de la carte d'acquisition, pour les intégrer dans notre programme

Voici, ci-dessous, un schéma (figure I.11) qui illustre la relation qui existe entre les différents modules :

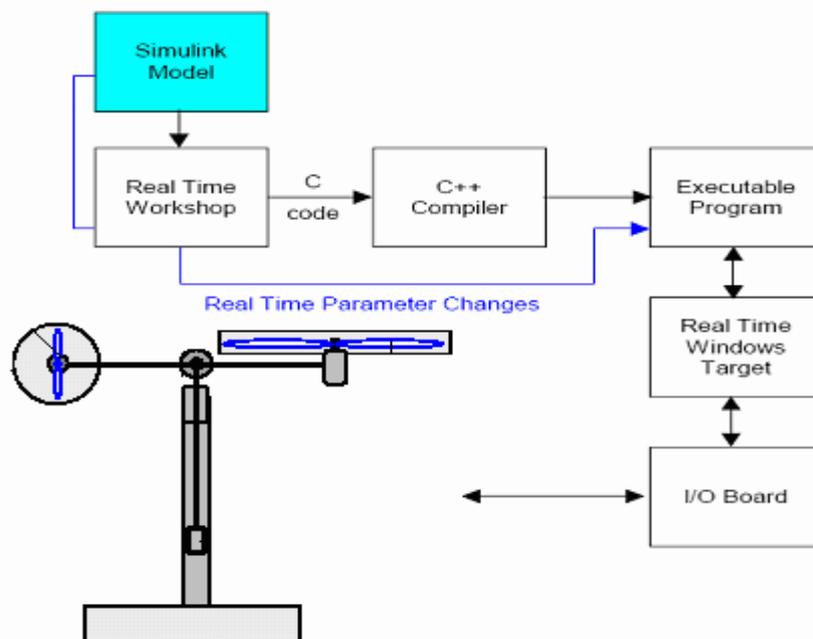


Figure I.11 : processus de génération du code exécutable

Et voici un autre schéma (figure I.12) qui illustre les différentes cibles (targets) et les différentes manières dont on peut implémenter notre régulateur :

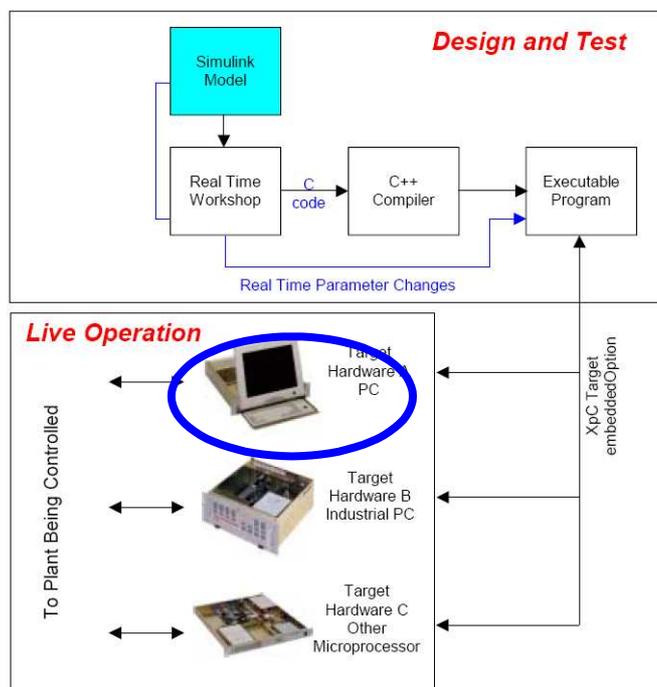


Figure I.12 : différentes cibles d'implémentation

Pour passer d'un régulateur simulé (non temps réel) vers un programme temps réel, il suffit uniquement 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, la librairie utilisée est la suivante :

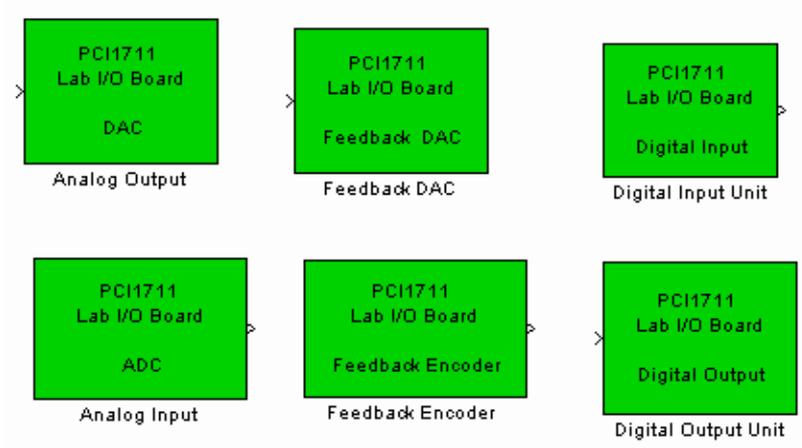


Figure I.13 : Librairie de la carte d'acquisition

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

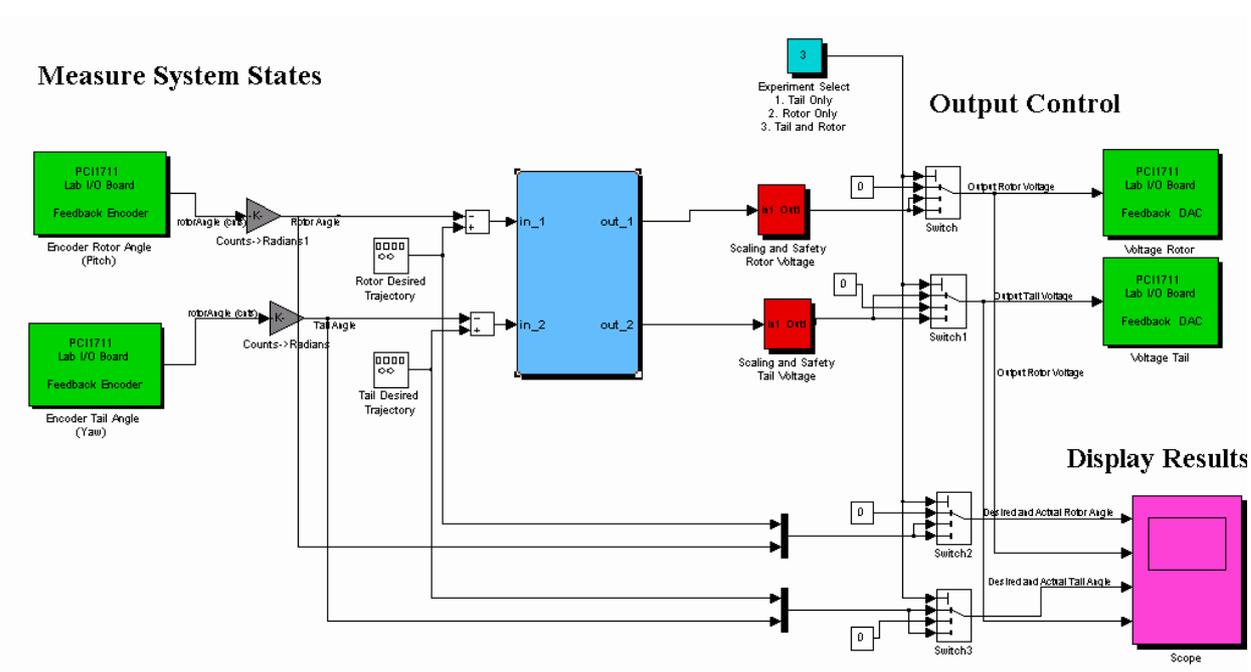


Figure I.14 : exemple d'implémentation d'un contrôleur

**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  $\pm 2.5V$ .

**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 dans 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 uniquement 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 fait une brève description du simulateur (TRMS), ainsi que son principe de fonctionnement. Enfin, on a expliqué l'environnement de développement utilisé.

chapitre

III

**Modélisation du TRMS**

# Chapitre II

## *Modélisation du TRMS*

### **II.1 Introduction**

L'analyse du comportement dynamique d'un système passe nécessairement par la recherche de l'ensemble des relations entre les grandeurs d'entrée et de sortie de celui-ci.

Cet ensemble constitue un modèle mathématique exprimant la dynamique du système sous forme d'équations différentielles ordinaires, à dérivées partielles, ou d'équations aux différences.

La construction d'un modèle mathématique peut s'effectuer par application des lois usuelles de la physique, ce modèle est alors dit « modèle de connaissance ». Cependant, en automatique, il arrive que, pour un manque d'informations disponibles sur le système, on se contente de reproduire le comportement entrée/sortie du système à partir de données expérimentales en considérant ce dernier comme une boîte noire. On parle alors d'identification.

On désire, toutefois, que le modèle puisse prévoir au mieux le comportement du système aux diverses excitations (commandes, perturbations ...). Ainsi, plus il est détaillé plus il est fidèle au système. Néanmoins, cela engendre une complication de l'étude et de la synthèse d'éventuelles lois de commande. Un compromis doit alors être fait en passant par des hypothèses simplificatrices afin de pouvoir répondre aux contraintes pratiques.

Dans ce chapitre, nous commencerons par une modélisation détaillée du simulateur présenté dans le chapitre précédent, on présentera ensuite les modèles découplés et on terminera par quelques simulations en boucle ouverte.

## II.2 Modèle non linéaire [3]

Les méthodes de modélisation classiques des systèmes couplés, non linéaires et d'ordre élevé, basées sur les équations de Lagrange, conduisent dans la majorité des cas à des modèles complexes. Pour cela, on utilisera les équations fondamentales de la dynamique (Lois de Newton).

Le modèle physique est développé sous certaines hypothèses simplificatrices :

- la dynamique des sous-systèmes de propulsion peut être décrite par des équations différentielles du premier ordre ;
- les frottements dans le système sont du type visqueux ;
- les sous-système hélices-air peuvent être décrits par les postulats de la théorie de l'écoulement aérodynamique.

### II.2.1 Modélisation du sous système d'élévation

Considérons le mouvement de la poutre uniquement dans le plan vertical (rotation autour de l'axe horizontal), la seconde loi de Newton donne :

$$M_v = J_v \frac{d^2 \alpha_v}{dt^2} \quad (\text{II.1})$$

Avec :

$$M_v = \sum_{i=1}^4 M_{vi} \quad (\text{II.2})$$

$$J_v = \sum_{i=1}^8 J_{vi} \quad (\text{II.3})$$

Ainsi l'équation (II.1) peut être mise sous la forme :

$$J_v \ddot{\alpha}_v = M_{v1} + M_{v2} + M_{v3} + M_{v4} \quad (\text{II.4})$$

Où :

$M_v$  : La somme des moments dans le plan vertical.

$J_v$  : La somme des moments d'inertie par rapport à l'axe horizontal.

$\alpha_v$  : L'angle d'élévation de la poutre.

$M_{v1}$  : Moment de la gravitation.

$M_{v2}$  : Moment de la force aérodynamique.

$M_{v3}$  : Moment de la force centrifuge.

$M_{v4}$  : Moment de friction.

### II.2.1.1 Moment de la gravitation $M_{v1}$

La figure suivante montre l'ensemble des forces de gravité qui agissent sur le sous-système d'élévation. Le moment gravitationnel résultant est la somme des moments de chaque force.

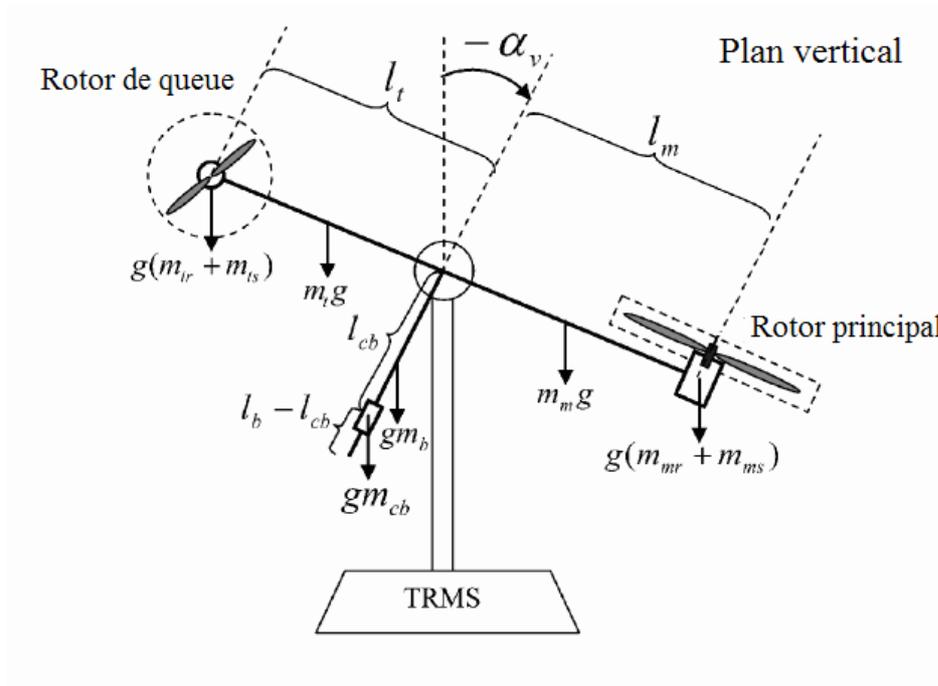


Figure II.1 : Forces de gravité agissantes sur le TRMS

$$M_{v1} = g \left\{ \left[ \left( \frac{m_t}{2} + m_{tr} + m_{ts} \right) \cdot l_t - \left( \frac{m_m}{2} + m_{mr} + m_{ms} \right) \cdot l_m \right] \cos \alpha_v - \left( \frac{m_b}{2} l_b + m_{cb} l_{cb} \right) \sin \alpha_v \right\} \quad (\text{II.5})$$

Ecriture sous forme condensée :

$$M_1 = g\{[A - B] \cos \alpha_v - C \sin \alpha_v\} \quad (\text{II.6})$$

$$\text{Avec } \begin{cases} A = \left(\frac{m_t}{2} + m_{tr} + m_{ts}\right) \cdot l_t \\ B = \left(\frac{m_m}{2} + m_{mr} + m_{ms}\right) \cdot l_m \\ C = \left(\frac{m_b}{2} l_b + m_{cb} l_{cb}\right) \end{cases}$$

Où :

$M_{v1}$  : Le moment résultant correspondant aux forces de gravités.

$m_{mr}$  : La masse du rotor principal.

$m_m$  : La masse de la partie principale de la poutre.

$m_{tr}$  : La masse du rotor secondaire.

$m_t$  : La masse de la partie secondaire de la poutre.

$m_{cb}$  : La masse du contrepoids.

$m_b$  : La masse de la poutre du contrepoids.

$m_{ms}$  : La masse de l'hélice principale.

$m_{ts}$  : La masse de l'hélice secondaire.

$l_m$  : La longueur de la partie principale de la poutre.

$l_t$  : La longueur de la partie secondaire de la poutre.

$l_b$  : La longueur de la poutre du contrepoids.

$l_{cb}$  : La distance entre le contrepoids et l'articulation.

$g$  : L'accélération gravitationnelle.

### II.2.1.2 Moment de la force aérodynamique $M_{v2}$

La figure II.2 ci-dessous montre la direction de la force aérodynamique de portance dans le plan vertical. Cette force dépend de la vitesse angulaire du rotor principal dont la relation est déterminée expérimentalement.

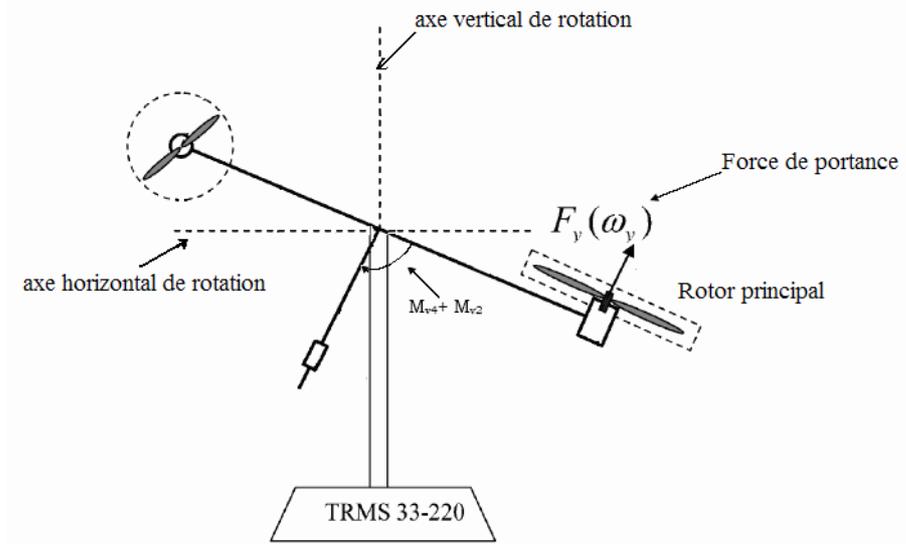


Figure II.2 : Moments de la force aérodynamique et de friction dans le TRMS

$$M_{v2} = l_m F_v(\omega_m) \tag{II.7}$$

Où :

$M_{v2}$  : Le moment de la force aérodynamique développé par le rotor principal.

$\omega_m$  : La vitesse angulaire du rotor principal.

$F_v(\omega_m)$  : Exprime la dépendance de la force aérodynamique de la vitesse angulaire du rotor principal.

### II.2.1.3 Moment de la force centrifuge $M_{v3}$

La masse de la poutre lorsqu'elle effectue un mouvement de rotation dans le plan horizontal génère une force orthogonale à la direction de rotation. Cette force est proportionnelle au carré de la vitesse angulaire de la poutre dans le plan horizontal, elle est nulle lorsque la poutre est élevée à  $\frac{\pi}{2}$  rad, et négative lorsque l'angle d'élévation est supérieur à  $\frac{\pi}{2}$  rad, et positive lorsqu'il est inférieur à  $\frac{\pi}{2}$  rad, son expression mathématique est :

$$M_{v3} = -\Omega_h^2 \left\{ \left( \frac{m_t}{2} + m_{tr} + m_{ts} \right) l_t + \left( \frac{m_m}{2} + m_{mr} + m_{ms} \right) l_m + \left( \frac{m_b}{2} l_b + m_{cb} l_{cb} \right) \right\} \sin \alpha_v \cos \alpha_v \tag{II.8}$$

Avec :

$$\Omega_h = \frac{d\alpha_h}{dt}$$

$\Omega_h$  : Vitesse angulaire de la poutre autour de l'axe vertical et  $\alpha_h$  est l'angle d'azimute de la poutre.

Ecriture sous forme compacte :

$$M_{v3} = -\Omega_h^2(A + B + C)\sin\alpha_v\cos\alpha_v \quad (\text{II.9})$$

#### II.2.1.4 Moment de la force de friction $M_{v4}$

$$M_{v4} = -\Omega_v k_v \quad (\text{II.10})$$

Avec :

$$\Omega_v = \frac{d\alpha_v}{dt}$$

$\Omega_v$  : La vitesse angulaire autour de l'axe horizontal

$k_v$  : Constante de friction à déterminer expérimentalement (par identification).

#### II.2.1.5 Moment d'inertie $J_v$

Le moment d'inertie par rapport à l'axe horizontal ne dépend pas de l'angle d'élévation car dans le plan vertical, la distribution de la masse est indépendante de l'angle  $\alpha_v$ .

$$\left\{ \begin{array}{l} J_{v1} = m_{mr}l_m^2 \\ J_{v2} = m_m \frac{l_m^2}{3} \\ J_{v3} = m_{cb}l_{cb}^2 \\ J_{v4} = m_b \frac{l_b^2}{3} \\ J_{v5} = m_{tr}l_t^2 \\ J_{v6} = m_t \frac{l_t^2}{3} \\ J_{v7} = \frac{m_{ms}}{2}r_{ms}^2 + m_{ms}l_m^2 \\ J_{v8} = m_{ts}r_{ts}^2 + m_{ts}l_t^2 \end{array} \right. \quad (\text{II.11})$$

Où :

$r_{ms}$  : Le rayon de l'hélice principale.

$r_{ts}$  : Le rayon de l'hélice secondaire.

## II.2.2 Modélisation du sous système d'azimut

Le mouvement dans le plan horizontal est repéré par l'angle d'azimut  $\alpha_h$ . On remarque, cependant, que la rotation de la poutre dans le plan horizontal est influencée par l'angle d'élévation  $\alpha_v$ .

### II.2.2.1 Influence de l'angle d'élévation sur le moment d'inertie

Le moment d'inertie d'un corps est fonction de la masse et de sa distribution géométrique dans ce corps. Pour cette raison, dans plusieurs systèmes mécaniques multidimensionnels comme c'est le cas du TRMS, on remarque que la distribution géométrique de la masse dépend de la position géométrique du corps, comme le montre la figure II.3.

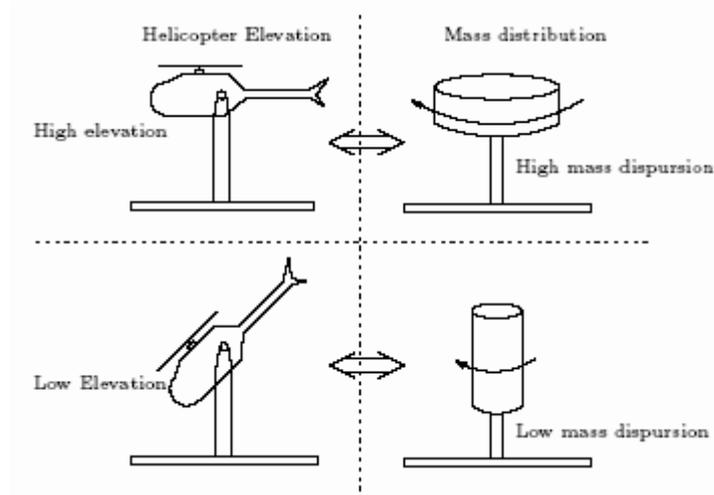


Figure II.3 : influence de l'angle d'élévation sur le moment d'inertie

### II.2.2.2 Influence de l'angle d'élévation sur le bras de levier des forces

Le moment d'une force dépend de la distance perpendiculaire entre cette force et l'axe de rotation. Dans notre cas, le mouvement dans le plan horizontal est généré par deux forces :

- La force de friction au niveau de l'articulation dont le bras de levier est supposé tellement petit qu'il n'est pas influencé par l'angle d'élévation ;
- Et la force aérodynamique qui possède un bras de levier variable en fonction de l'angle d'élévation  $\alpha_v$

De la même manière que pour le sous-système d'élévation, le mouvement de la poutre dans le plan horizontal peut être décrit par la deuxième loi de Newton :

$$M_h = J_h \frac{d^2 \alpha_h}{dt^2} \quad (\text{II.12})$$

Où :  $M_h$  est la somme des moments des forces agissantes dans le plan horizontal, et  $J_h$  est la somme des moments d'inertie par rapport à l'axe vertical.

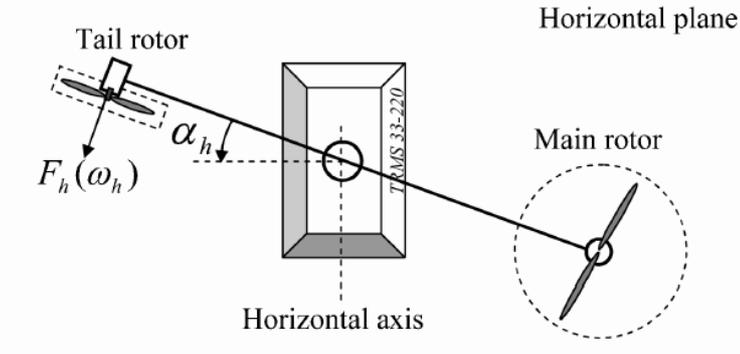
Ainsi :

$$M_h = \sum_{i=1}^2 M_{hi} \quad (\text{II.13})$$

$$J_h = \sum_{i=1}^8 J_{hi} \quad (\text{II.14})$$

### II.2.2.3 Moment de la force aérodynamique

Pour déterminer le moment de la force aérodynamique, considérons le cas présenté dans la figure II.4.



**Figure II.4 :** Moment de la force aérodynamique dans le plan horizontal

$$M_{h1} = l_t F_h(\omega_t) \cos \alpha_v \quad (\text{II.15})$$

Où :

$\omega_t$  : La vitesse angulaire du rotor secondaire.

$F_h(\omega_t)$  : Exprime la dépendance de la force aérodynamique de la vitesse angulaire du rotor secondaire.

### II.2.2.4 Moment de la force de friction $M_{h2}$

$$M_{h2} = -\Omega_h k_h \quad (\text{II.16})$$

Avec :  $k_h$  Constante de friction à déterminer expérimentalement par l'identification.

### II.2.2.5 Moment d'inertie $J_h$

Les moments d'inertie par rapport à l'axe vertical sont :

$$\left\{ \begin{array}{l} J_{h1} = \frac{m_m}{3} (l_m \cos \alpha_v)^2 \\ J_{h2} = \frac{m_t}{3} (l_t \cos \alpha_v)^2 \\ J_{h3} = \frac{m_b}{3} (l_b \sin \alpha_v)^2 \\ J_{h4} = m_{tr} (l_t \cos \alpha_v)^2 \\ J_{h5} = m_{mr} (l_m \cos \alpha_v)^2 \\ J_{h6} = m_{cb} (l_{cb} \sin \alpha_v)^2 \\ J_{h7} = \frac{m_{ts}}{2} r_{ts}^2 + m_{ts} (l_{lt} \cos \alpha_v)^2 \\ J_{h8}^2 = m_{ms} r_{ms}^2 + m_{ms} (l_m \cos \alpha_v)^2 \end{array} \right. \quad (\text{II.17})$$

Ou sous forme compact :

$$J_h(\alpha_v) = D \cos^2 \alpha_v + E \sin^2 \alpha_v + F \quad (\text{II.18})$$

Avec :

$$\left\{ \begin{array}{l} D = \left( \frac{m_m}{3} + m_{mr} + m_{ms} \right) l_m^2 + \left( \frac{m_t}{3} + m_{tr} + m_{ts} \right) l_t^2 \\ E = \frac{m_b}{3} l_b^2 + m_{cb} l_{cb}^2 \\ F = m_{ms} r_{ms}^2 + \frac{m_{ts}}{2} r_{ts}^2 \end{array} \right.$$

Où

$J_{tr}$  : Le moment d'inertie dans le moteur secondaire.

$J_{mr}$  : Le moment d'inertie dans le moteur principal.

### II.2.3 Modélisation des propulseurs

Le TRMS possède deux propulseurs constitués chacun d'un moteur à courant continu plus une hélice, un pour le rotor principal et l'autre pour le rotor de queue (rotor secondaire). Les deux moteurs DC sont identiques, mais avec des charges mécaniques différentes.

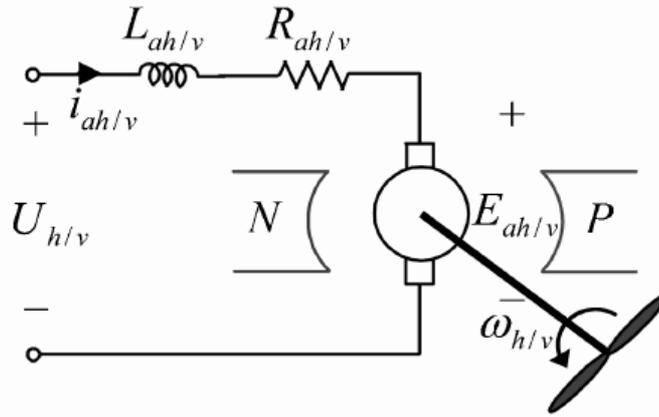


Figure II.5 : Circuit de l'un des propulseurs du TRMS

L'équation différentielle du moteur DC est supposée du premier ordre, en négligeant la constante de temps électrique devant la constante de temps mécanique.

Cette équation différentielle est donnée par :

$$I\dot{\omega} = \frac{K_i}{R}(u - K_b\omega) - T_L \quad (\text{II.19})$$

Avec :

$\omega$  : Vitesse angulaire du moteur (rad/s) .

$u$  : Tension de commande (V).

$I$  : Moment d'inertie.

$R$  : Résistance de l'armature.

$K_b$  : Constante de la FEM.

$K_i$  : Constante du couple.

$T_L$  : Couple résistant généré par la charge.

La charge  $T_L$ , représente le couple résistant généré par les frottements mécaniques et les frottements aérodynamiques de l'hélice avec l'air. Ces frottements sont difficilement modélisables. Pour cela, leurs caractéristiques statiques sont déterminées expérimentalement. Elles dépendent de la forme et des dimensions de chaque hélice. Dans le TRMS, l'hélice du rotor principal est différente de celle du rotor secondaire.

Pour cela, on a introduit deux nouvelles variables :  $U_{vv}$  pour le vertical et  $U_{hh}$  pour l'horizontal, comme l'illustre la figure (II.6).

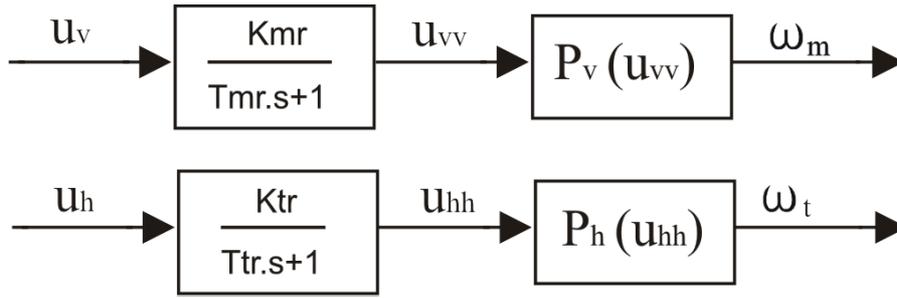


Figure II.6 : schéma bloc des propulseurs

Où :

$T_{mr}$  : La constante du temps du moteur principal.

$T_{tr}$  : La constante du temps du moteur secondaire.

$K_{mr}$  : Le gain statique du moteur principal.

$K_{tr}$  : Le gain statique du moteur secondaire.

$P_v(U_{vv})$  et  $P_h(U_{hh})$  sont les caractéristiques statiques des frottements, déterminées expérimentalement.

Ainsi, les sous-systèmes propulseurs sont décrits par des équations différentielles du premier ordre avec des fonctions non linéaires à la sortie :

$$\begin{cases} \frac{du_{vv}}{dt} = \frac{1}{T_{mr}} (-u_{vv} + K_{mr}u_v) \\ \omega_m = P_v(u_{vv}) \end{cases} \quad (\text{II.20})$$

$$\begin{cases} \frac{du_{hh}}{dt} = \frac{1}{T_{tr}} (-u_{hh} + K_{tr}u_h) \\ \omega_t = P_h(u_{hh}) \end{cases} \quad (\text{II.21})$$

#### II.2.4 Effet gyroscopique

Lorsqu'on met une hélice en rotation autour d'un axe, plusieurs phénomènes surviennent, notamment, l'effet gyroscopique ou le mouvement de précession.

De plus, un objet en rotation possède un moment cinétique (ou moment angulaire) proportionnel à son inertie et sa vitesse angulaire  $\vec{S} = I \cdot \vec{\omega}$ . Le théorème de Koenig met en relation le moment cinétique et la somme des moments :

$$\frac{d\vec{S}}{dt} = \sum_i \vec{M}(\vec{F}_i)$$

En effet, dans un hélicoptère on a une hélice (masse) qui tourne autour d'un axe

principal tout comme dans un gyroscope. Ce que l'on retient en premier lieu, c'est que l'axe et l'hélice vont toujours tendre à être sur le même axe de rotation, et lorsqu'une force viendra perturber le système (propulsion, frottement de l'air sur les pales,...), celui-ci voudra la compenser en entamant une rotation autour d'un axe du plan horizontal parallèle à la force appliquée.

Donc par analogie, toute force appliquée ailleurs que sur le centre de masse du système provoquera un mouvement résultant perpendiculaire, en plus de la résistance de l'air lors du mouvement du simulateur.

D'où les relations suivantes :

$$S_v = J_v \Omega_v - J_{tr} \omega_t \quad (\text{II.22})$$

$J_{tr} \omega_t$  : est le moment cinétique généré par le rotor de queue pour contrer l'effet gyroscopique dû à la rotation de l'hélice principale.

$$S_h = J_h(\alpha_v) \Omega_h - J_{mr} \omega_m \cos \alpha_v \quad (\text{II.23})$$

$J_{mr} \omega_m \cos \alpha_v$  : est le moment cinétique généré par le rotor principal pour contrer l'effet gyroscopique dû à la rotation de l'hélice du rotor secondaire.

### II.3 Modèle d'état non linéaire

En utilisant les relations mathématiques développées précédemment, on peut réécrire les équations régissant le mouvement du système comme suit :

$$\frac{dS_v}{dt} = l_m F_v(\omega_m) - \Omega_v k_v + g((A - B) \cos \alpha_v) - \frac{1}{2} \Omega_h^2 (A + B + C) \sin 2\alpha_v \quad (\text{II.24})$$

$$\frac{d\alpha_v}{dt} = \Omega_v \quad (\text{II.25})$$

$$\Omega_v = \frac{S_v + J_{tr} \omega_t}{J_v} \quad (\text{II.26})$$

$$\frac{du_{vv}}{dt} = \frac{1}{T_{mr}} (-u_{vv} + K_{mr} u_v) \quad (\text{II.27})$$

$$\omega_m = P_v(u_{vv})$$

$$\frac{dS_h}{dt} = l_t F_h(\omega_t) \cos \alpha_v - \Omega_h k_h \quad (\text{II.28})$$

$$\frac{d\alpha_h}{dt} = \Omega_h \quad (\text{II.29})$$

$$\Omega_h = \frac{S_h + J_{mr}\omega_m \cos\alpha_v}{J_h(\alpha_v)} \quad (\text{II.30})$$

$$\frac{du_{hh}}{dt} = \frac{1}{T_{tr}}(-u_{hh} + K_{tr}u_h) \quad (\text{II.31})$$

Ainsi,  
en

$$\omega_t = P_h(u_{hh})$$

classifiant les différentes variables comme suit :

- Vecteur d'entrée :  $U = [U_v U_h]^T$
- Vecteur d'état :  $X = [\alpha_v S_v u_{vv} \alpha_h S_h u_{hh}]^T$
- Vecteur de sortie :  $Y = [\alpha_v \alpha_h]^T$

On obtient une représentation d'état non linéaire sous la forme : 
$$\begin{cases} \dot{X} = f(X) + g(X)U \\ Y = h(X) \end{cases}$$

où il y a 6 variables d'état.

Représentation d'état :

$$\left\{ \begin{array}{l} \dot{x}_1 = \frac{x_2 + J_{tr}P_h(x_6)}{J_v} \\ \dot{x}_2 = l_m F_v(P_v(x_3)) - k_v \left[ \frac{x_2 + J_{tr}P_h(x_6)}{J_v} \right] + g((A - B)\cos(x_1) - C\sin(x_1)) \\ \quad - \frac{1}{2} \left[ \frac{x_5 + J_{mr}P_v(x_3)\cos(x_1)}{J_h(x_1)} \right]^2 (A + B + C)\sin(2x_1) \\ \dot{x}_3 = \frac{1}{T_{mr}}(-x_3 + K_{mr}u_1) \\ \dot{x}_4 = \frac{x_5 + J_{mr}P_v(x_3)\cos(x_1)}{J_h(x_1)} \\ \dot{x}_5 = l_t F_h(P_h(x_6))\cos(x_1) - k_h \left[ \frac{x_5 + J_{mr}P_v(x_3)\cos(x_1)}{J_h(x_1)} \right] \\ \dot{x}_6 = \frac{1}{T_{tr}}(-x_6 + K_{tr}u_2) \end{array} \right. \quad (\text{II.32})$$

La figure (II.7), montre le diagramme fonctionnel du TRMS :

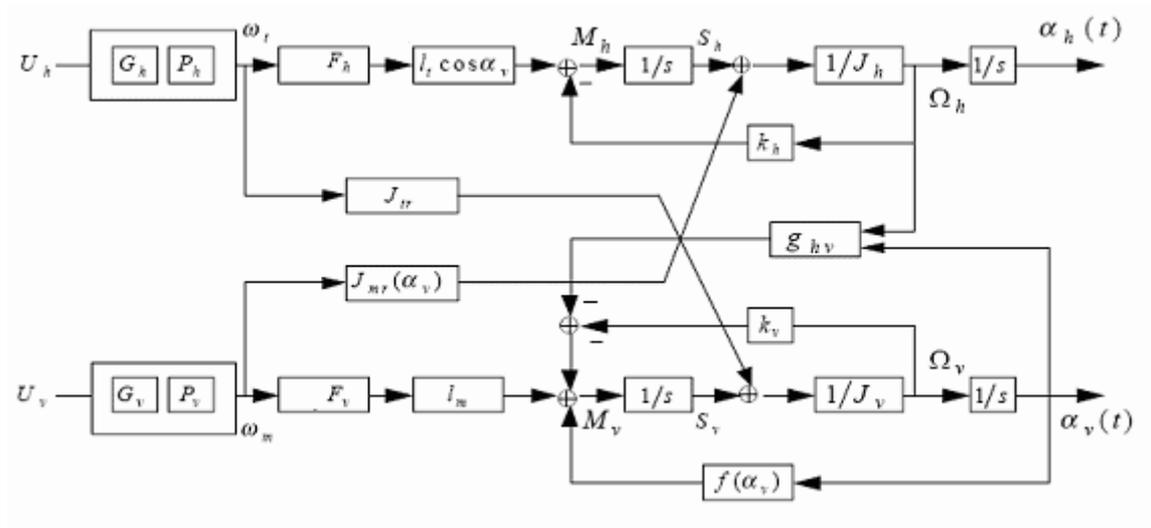


Figure II.7: Schéma bloc du TRMS

## II.4 Le modèle découplé

En contraignant le mouvement de la tige dans un seul plan soit l'horizontal ou le vertical, on obtient deux sous modèles ayant chacun un degré de liberté.

### II.4.1 Modèle 1 DDL vertical

Ce modèle est dérivé du modèle couplé, en fixant l'angle d'azimut  $\alpha_h$ , et en posant  $u_h = 0$ . On choisit le vecteur d'état suivant :

$$X = [\alpha_v \ S_v \ u_{vv}]^T$$

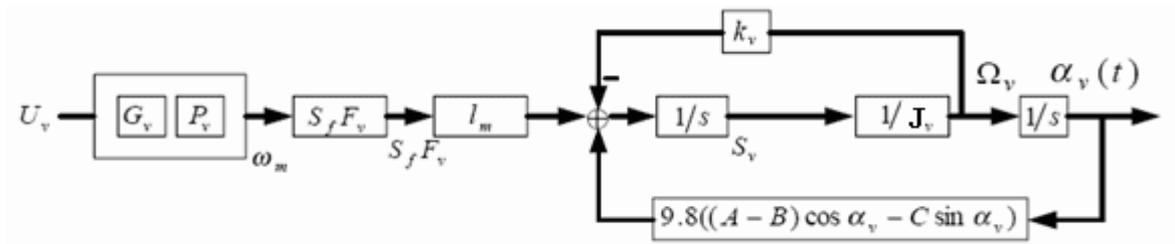


Figure II.8 : Schéma bloc du modèle vertical

La représentation d'état est alors :

$$\begin{cases} \dot{x}_1 = \frac{x_2}{J_v} \\ \dot{x}_2 = l_m F_v(P_v(x_3)) - k_v \cdot \frac{x_2}{J_v} + g((A-B)\cos(x_1) - C\sin(x_1)) \\ \dot{x}_3 = \frac{1}{T_{mr}} (-x_3 + K_{mr} u_1) \end{cases} \quad (II.33)$$

**Remarque :**

Le modèle vertical obtenu ne dépend pas de l'angle d'azimut  $\alpha_h$ .

**II.4.2 Modèle 1 DDL horizontal**

De la même façon que pour le modèle vertical, dans le modèle couplé on pose  $\alpha_v = \alpha_{v0}$  Et  $u_v = 0$ . Et on choisit  $X = [\alpha_h S_h u_{hh}]^T$  comme vecteur d'état.

Le modèle horizontal est ainsi :

$$\begin{cases} \dot{x}_1 = \frac{1}{J_h(\alpha_{v0})} x_2 \\ \dot{x}_2 = l_t F_h(P_h(x_3)) \cos(\alpha_{v0}) - \frac{k_h}{J_h(\alpha_{v0})} x_2 \\ \dot{x}_3 = \frac{1}{T_{tr}} (-x_3 + K_{tr} u_h) \end{cases} \quad (\text{II.34})$$

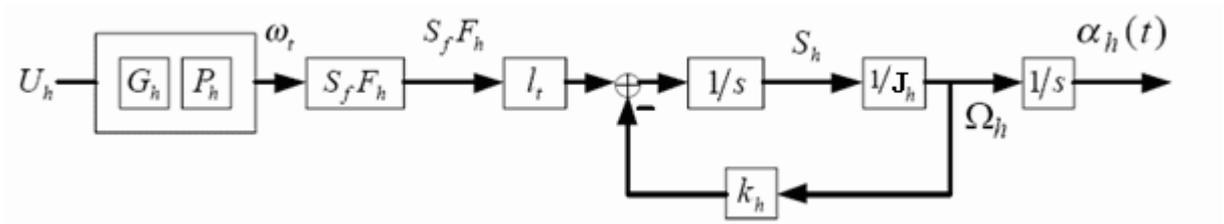


Figure II.9 : Schéma bloc du modèle horizontal

**II.5 Simulations**

Afin de pouvoir utiliser le modèle du TRMS pour des simulations ou pour développer des lois de commandes, nous devons déterminer les différents paramètres et constantes de temps du système, ainsi que les caractéristiques statiques des fonctions non linéaires en utilisant l'identification.

Les courbes obtenues expérimentalement (annexe A), sont approximées par des polynômes en utilisant la fonction *polyfit* de Matlab.

Ainsi, pour le rotor principal on obtient les expressions suivantes:

**Fonction non linéaire du rotor principal :**

$$P_v(U_{vv}) = 90,99. U_{vv}^6 + 599,73. U_{vv}^5 - 129,26. U_{vv}^4 - 1238,64. U_{vv}^3 + 63,45. U_{vv}^2 + 1283,41. U_{vv} \quad (\text{II.36})$$

**Fonction aérodynamique principal :**

$$F_v(\omega_m) = -3,48.10^{-12}.\omega_m^5 + 1,09.10^{-9}.\omega_m^4 + 4,123.10^{-6}.\omega_m^3 - 1,632.10^{-4}.\omega_m^2 + 9,544.10^{-2}.\omega_m \quad (\text{II.35})$$

Et pour le rotor secondaire on a :

**Fonction non linéaire du rotor secondaire :**

$$P_h(U_{hh}) = 2020.U_{hh}^5 - 194,69.U_{hh}^4 - 4283,15.U_{hh}^3 + 262,27.U_{hh}^2 + 3796,83.U_{hh} \quad (\text{II.37})$$

**Fonction aérodynamique secondaire**

$$F_t(\omega_t) = -3.10^{-14}.\omega_t^5 - 1,595.10^{-11}.\omega_t^4 + 2,511.10^{-7}.\omega_t^3 - 1,808.10^{-4}.\omega_t^2 + 0,0801.\omega_t \quad (\text{II.38})$$

Pour les rotors principal et secondaire, les polynômes décrivant les fonctions non linéaires des moteurs peuvent être remplacés par :

Rotor principal :

$$P_v(U_{vv}) = 600.a \tan(2U_{vv}) \quad (\text{II.39})$$

Rotor secondaire :

$$P_h(U_{hh}) = 1200.a \tan(3,8U_{vv}) \quad (\text{II.40})$$

Les paramètres du modèle sont représentés sous forme de tableau, ils seront utilisés lors de la simulation du modèle [5]

**Tableau II.1 : Les paramètres du modèle**

Paramètre	Valeur
A	0.0947
B	0.1105
C	0.011702
D	0.048814
E	0.0016087
F	0.006225
$J_v(kg.m^2)$	0.055448
$J_{tr}(kg.m^2)$	$1.6543.10^{-5}$
$J_{mr}(kg.m^2)$	$2.65.10^{-5}$
$l_m(m)$	0.24
$l_t(m)$	0.25

$T_{mr}$	1.432
$T_{tr}$	0.3842
$K_{mr}$	1
$K_{tr}$	1
$K_v$	0.00545371
$K_h$	0.0095

II.5. 1 simulation de modèle du TRMS :

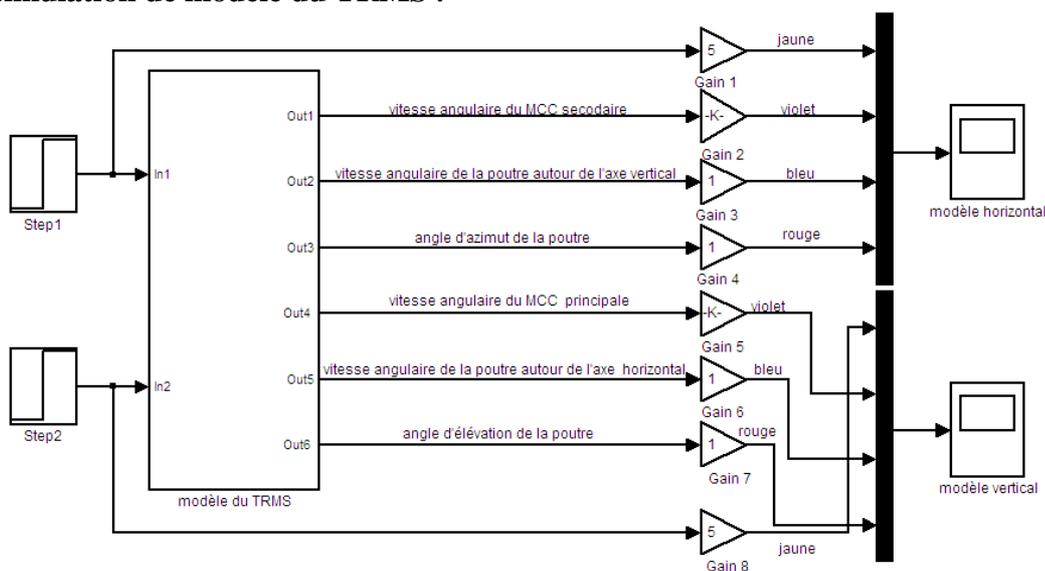


Figure II.10 : schéma de simulation sous matlab/simulink du TRMS

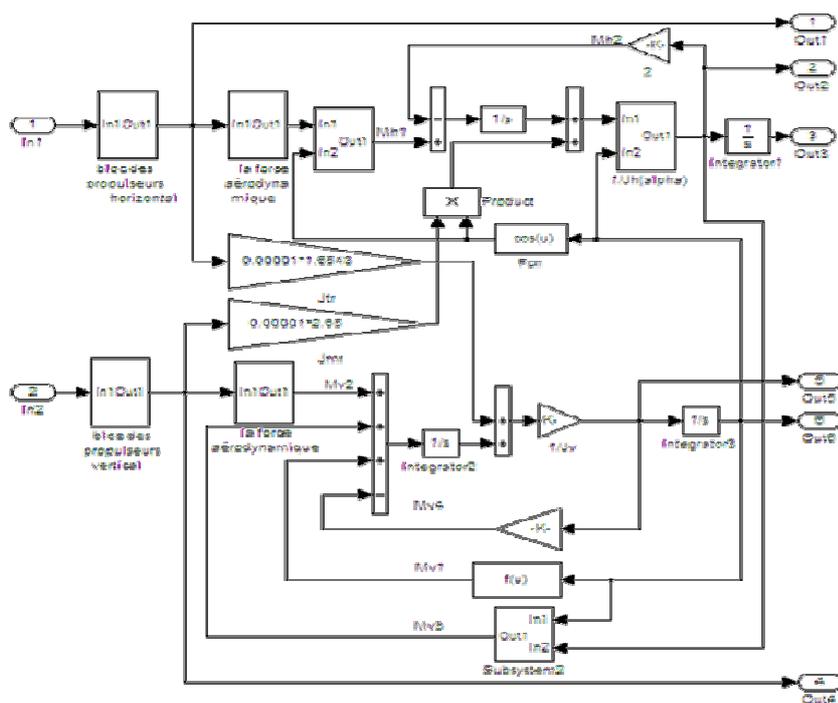
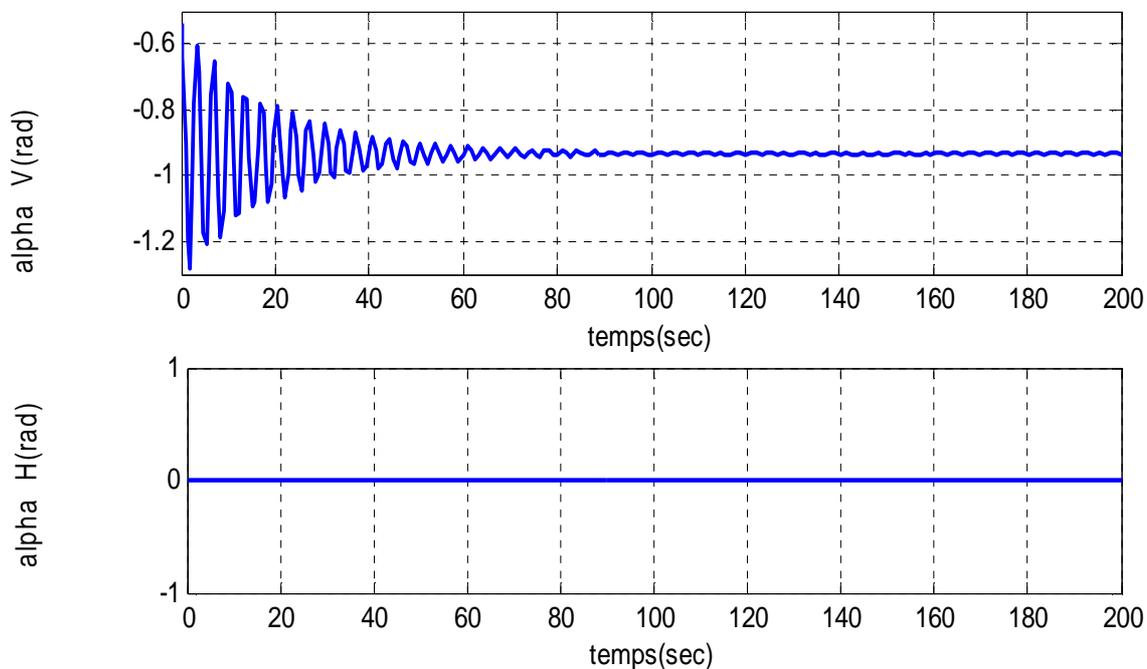


Figure II.11: schéma de simulation du TRMS en détail

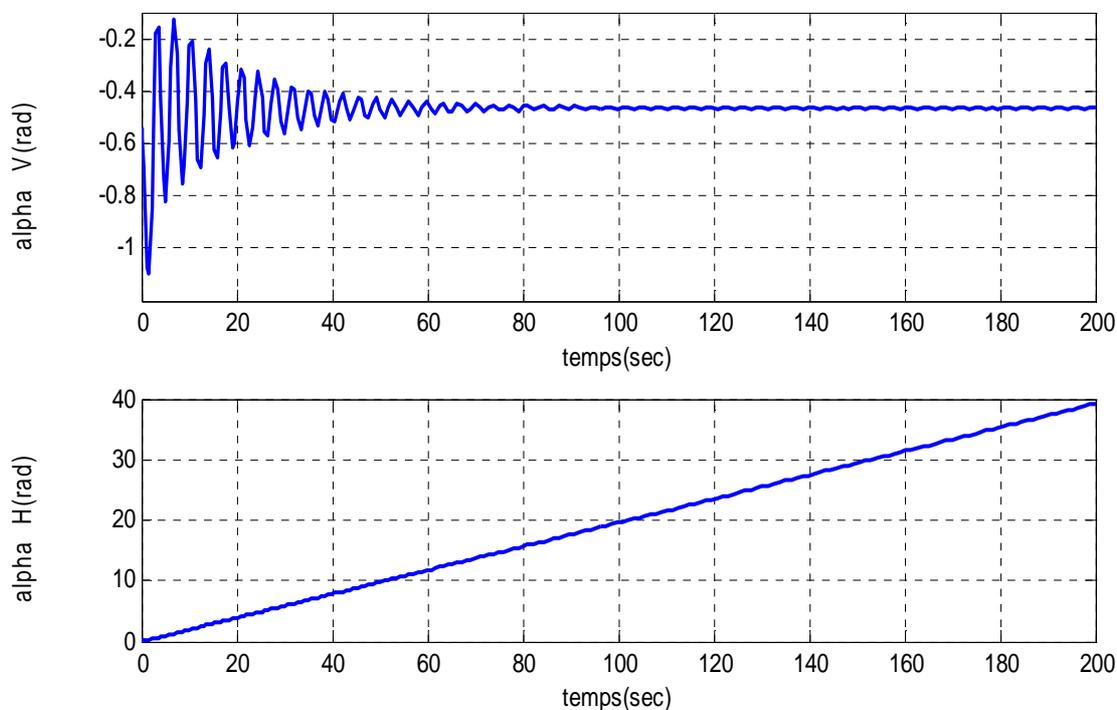
**II.5. 2 Résultant de simulation :**

- **Modèle couplée**

Les figures II.12 et II.13, montrent les réponses libres et forcées du simulateur d'hélicoptère, avec les conditions initiales :  $\alpha_v = \alpha_h = 0^\circ$  :



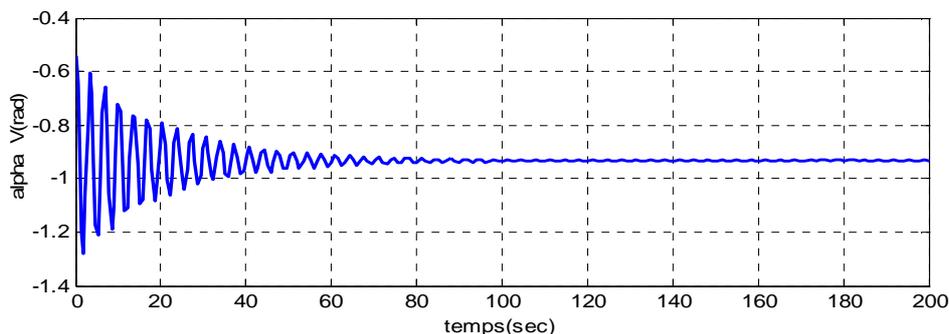
**Fig.II.12 :** réponses libres du TRMS avec  $\alpha_{v0} = -0.54(\text{rad})$   $\alpha_{h0} = 0(\text{rad})$ ,  $u_v = 0.5(\text{volt})$ ,  $U_v = U_h = 0(\text{volt})$



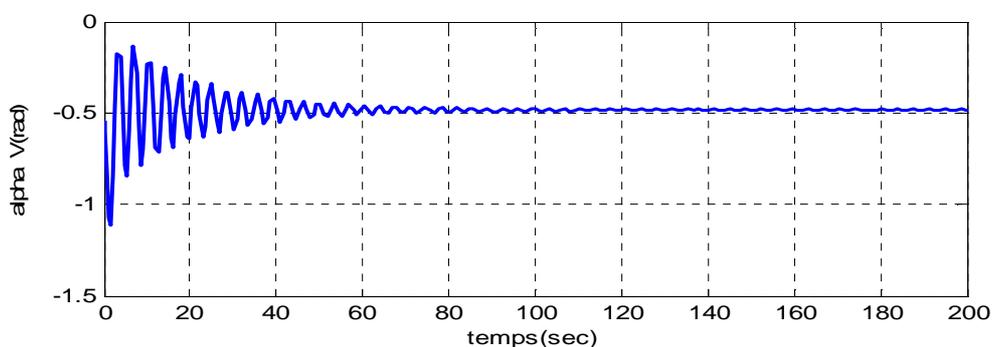
**Figure II.13 :** réponses du TRMS pour  $U_v = U_h = 0.5(\text{volt})$  et  $\alpha_{v0} = -0.54(\text{rad})$  et  $\alpha_{h0} = 0(\text{rad})$

- modèle découplé

**Modèle vertical**

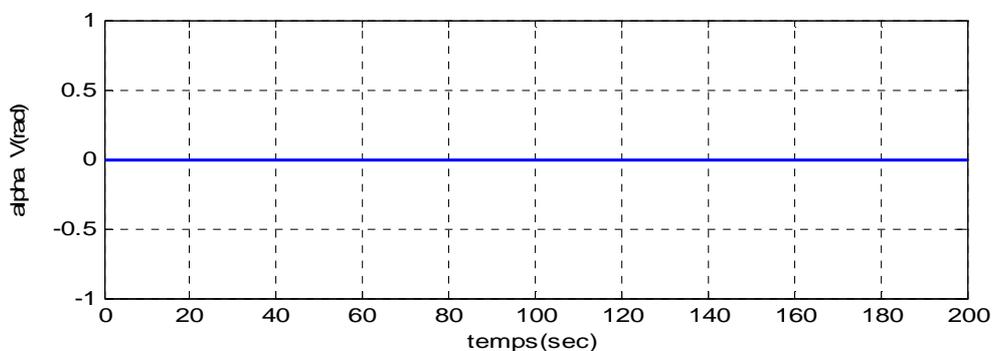


*Fig. II.14: réponses libres du modèle vertical avec  $\alpha_{v0} = -0.54(\text{rad}), u_v = 0(\text{volt})$*

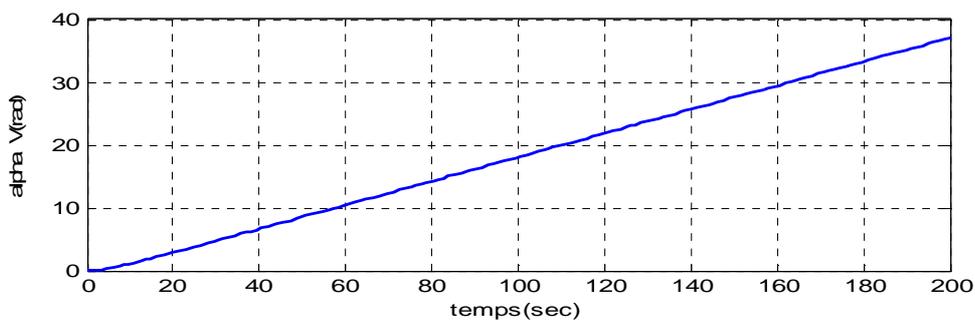


*Fig. II.15 : réponses libres du modèle vertical avec  $\alpha_{v0} = -0.54(\text{rad}), u_v = 0.5(\text{volt})$*

**Modèle horizontal**



*Fig. II.16 : réponses libres du modèle horizontal avec  $\alpha_{h0} = 0(\text{rad}), u_h = 0(\text{volt})$*



*Fig. II.17 : réponses libres du modèle horizontal avec  $\alpha_{h0} = 0(\text{rad}), u_h = 0.5(\text{volt})$*

La réponse libre du sous-système d'élévation est oscillatoire amortie, cela est due aux forces gravitationnelles qui agissent uniquement sur le plan vertical, et poussent le simulateur à se stabiliser en un point d'équilibre  $\alpha_v = -0.93$  rad. Par contre, le sous-système d'élévation reste dans sa position d'origine tant qu'il n'y pas d'excitation du Rotor de queue.

Lorsqu'on excite le système, le sous-système vertical tend vers un nouveau point d'équilibre car la commande 0.5 n'est pas suffisante pour l'élever vers un angle supérieur. Cependant, le sous système horizontal a le comportement d'un intégrateur, il diverge même pour de petites excitations. Ceci est dû essentiellement à la faible inertie du mouvement horizontal où il n'y a pas de force de gravité qui s'oppose au mouvement.

## II.6 Conclusion

Dans ce chapitre, on a établi un modèle mathématique du simulateur d'hélicoptère TRMS, celle-ci a abouti à une représentation d'état fortement non linéaire, couplée et découplée. Cette modélisation nous permettra par la suite de concevoir des contrôleurs.

# chapitre

# III

**Théories de la logique floue, les réseaux de neurone  
et  
Neuro-Floue**

Chapitre  
III***Théories de la logique floue, les réseaux de neurone  
et Neuro-Floue*****III.1 Introduction**

Dans ce chapitre, nous allons présenter et analyser quelques structures intelligentes basées sur les concepts tels que la logique floue, les réseaux de neurones et les réseaux neuro-flous. Cette démarche est menée afin de lever une contrainte en contrôle relative à la nécessité de disposer d'un modèle le plus fidèle possible, sachant que l'erreurs de modélisation et les imperfections des modèles, contribuent à dégrader notamment les performances des lois de commande conventionnelles[7].

Dans la première partie nous étudieront en détail les principes de la logique floue et ses applications dans le domaine du réglage puis nous décrirons les différentes solutions pour un organe de commande basé sur les principes de la logique floue.

Ensuite, dans la deuxième partie de ce chapitre, nous discuterons des algorithmes dits "réseaux de neurones". Nous présenterons les propriétés des réseaux de neurones, les différentes structures et les méthodes d'apprentissages connues dans la littérature.

Enfin, dans la troisième partie de ce chapitre nous nous concentrerons en particulier sur la projection des systèmes flous dans un réseau de neurone afin de former un système neuro-floue en tirant les profits de la logique floue et des réseaux de neurone.

**III.2 la Logique floue****III.2.1 Introduction**

La logique floue est une théorie connue depuis que **Zadeh** a introduit le concept de sous-ensemble flou en 1965. Par la suite, en 1974, **Mamdani** introduisait la commande floue pour la régulation de processus industriels, dans les années 80, la commande floue connaît un essor considérable au Japon, notamment grâce aux travaux de Sugeno. La logique floue permet de faire le lien entre modélisation numérique et Modélisation symbolique, ce qui

a permis des développements industriels spectaculaires à partir d'algorithmes très simples de traduction de connaissances symboliques en entité numérique et inversement. elle présente en effet l'avantage d'utiliser des règles linguistiques simples permettant de traduire facilement le savoir faire d'un expert pour répondre à une problématique spécifique [8] .

### III.2.2 Généralité sur la logique floue

La description d'une certaine situation imprécise ou incertaine peut contenir des expressions floues comme par exemple: **très grand, grand, moyen, petit**. Ces expressions forment les valeurs d'une variable  $x$ , appelée "**linguistique**", soumise à des fonctions appelées fonctions d'appartenance, [9-10].

#### III.2.2.2 Opérateur de la logique floue

Les variables linguistiques sont liées entre elles au niveau des règles d'inférence par des opérateurs ET ou OU. Il s'agit d'opérateurs de la logique floue qui interviennent sur les fonctions d'appartenance représentant les variables linguistiques. De plus, il existe l'opérateur NON qui s'appelle (complément, négation, inverse)

##### III.2.2.2.1 Opérateur OU (l'intersection)

La fonction d'appartenance  $\mu_{A \cup B}$  de deux ensembles A et B est définie pour tout  $\mu \in u$ , par :

$$\text{➤ Mamdani : } \mu_{A \cup B}(u) = \max \{ \mu_A(u), \mu_B(u) \} \quad (\text{III.1})$$

$$\text{➤ Sugeno: } \mu_{A \cup B}(u) = \mu_A(u) + \mu_B(u) - \mu_A(u) * \mu_B(u) \quad (\text{III.2})$$

##### III.2.2.2.2 Opérateur ET (l'intersection)

La fonction d'appartenance  $\mu_{A \cap B}$  de deux ensembles A et B est définie pour tout  $\mu \in u$ , par :

$$\text{➤ Mamdani : } \mu_{A \cap B}(u) = \min \{ \mu_A(u), \mu_B(u) \} \quad (\text{III.3})$$

$$\text{➤ Sugeno: } \mu_{A \cap B}(u) = \mu_A(u) * \mu_B(u) \quad (\text{III.4})$$

##### III.2.2.2.3 Opérateurs NON (complément)

La fonction d'appartenance  $\mu_{A'}(u)$  d'un ensemble A est définie, pour tout  $\mu \in u$ , par

$$\mu_{A'}(u) = 1 - \mu_A(u) \quad (\text{III.5})$$

### III.2.2.3 Représentation des fonction d'appartenance [11]

on représente les variables linguistiques par leurs fonctions d'appartenances. Donc à chaque sous-ensemble flou  $A_i$  est associé une fonction d'appartenance  $\mu_{A_i}(x)$  ou  $x$  est la variable linguistique. Tel que, à chaque point  $x$  est associé une valeur précise de  $\mu_{A_i}(x)$ , qui désigne, le degré d'appartenance de  $x$  à  $A_i$ .

La fonction d'appartenance peut être présentée par plusieurs forme : triangulaire, trapézoïdale et gussienne. On peut définir d'autres formes de fonctins d'appartenance, mais dans le réglage par logique flous, les formes déjà citées et illustrées sur la figure (III.2) sont largement suffisantes pour délimiter les ensembles flous.

#### III.2.2.3.1 Fonction triangulaire[11]

Elle est définie partrois paramaitres  $\{a,b,c\}$ , qui déterminent les cordonnées des trois somets.

$$\mu(x) = \max \left( \min \left( \frac{x-a}{b-a}, \frac{c-x}{c-b}, 0 \right) \right) \quad (\text{III.6})$$

#### III.2.2.3.2 Fonctions trapézoïdale

Elle est définie par quatre paramètre  $\{a,b,c,d\}$  ,qui déterminent les coordonnées des quatre sommes.

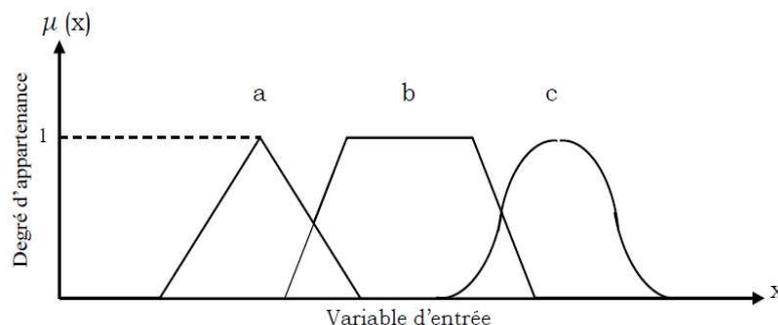
$$\mu(x) = \max \left( \min \left( \frac{x-a}{b-a}, 1, \frac{d-x}{d-c}, 0 \right) \right) \quad (\text{III.7})$$

#### III.2.2.3.3 Fonction gussienne

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

$$\mu(x) = \exp \left( -\frac{(x-m)^2}{2\sigma^2} \right) \quad (\text{III.8})$$

La figure (III.1) donne quelques fonctions d'appartenance.

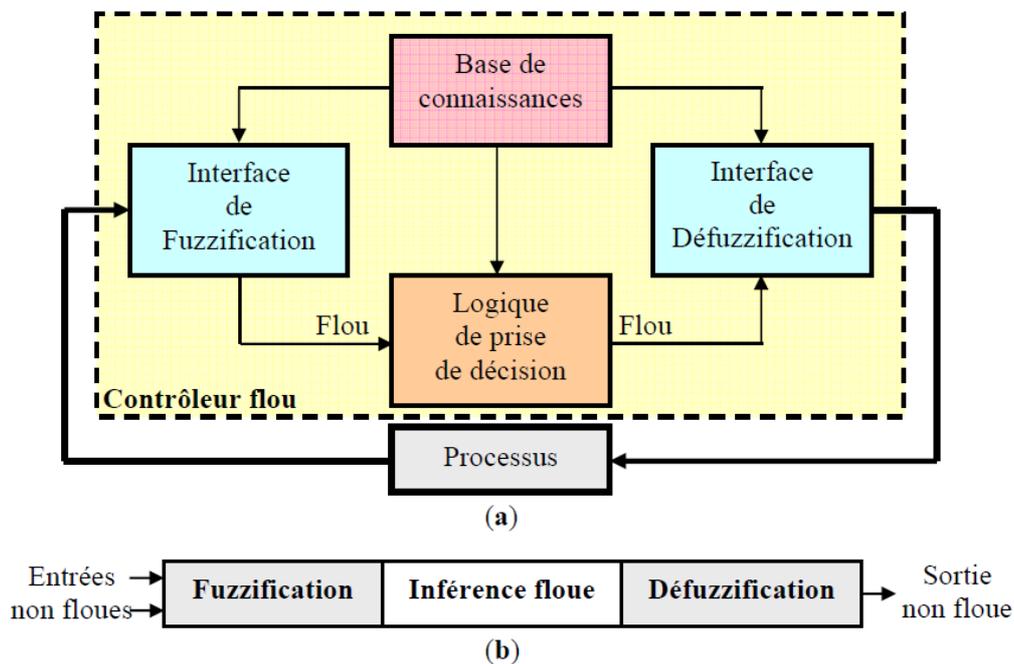


**Fig. III.1** : Exemple de fonctions d'appartenance.

a) Fonction triangulaire. b) Fonction trapézoïdale. c) Fonction gaussienne.

### III.2.3 Structure de base d'un contrôleur flou

Le schéma synoptique général d'un contrôleur flou est représenté dans la figure (III.2), [12-13], [14-15] :



**Fig. III.2 a) :** Schéma synoptique d'un contrôleur flou  
**b) :** configuration d'un contrôleur flou.

#### III.2.3.1 Base de connaissances

La base de connaissance comprend une connaissance du domaine d'application et les but du contrôle prévu. Elle est composée :

1. D'une base de données fournissant les informations nécessaires pour les fonctions de normalisation [16].
2. La base de règle constitue un ensemble d'expressions linguistiques structurées autour d'une connaissance d'expert, et représentée sous forme de règle :

#### III.2.3.2 Fuzzification

la fuzzification est l'opération qui consiste à affecter pour chaque entrée physique, un degré d'appartenance à chaque sous-ensemble flou. En d'autres termes c'est l'opération qui permet le passage du numérique (grandeurs physiques) au symbolique (variables floues).

Pour illustrer le mécanisme de la fuzzification, nous allons donné un exemple en fixant comme valeur d'entrée  $e_K = 0.45$ . le resultat de la fuzzification sera présenté sur la figure

(III.3). on remarque que pour cette erreur correspond les ensemble flous PP et PM avec les degrés d'appartenances  $\mu_{PP}(e_K) = 0.75$  et  $\mu_{PM}(e_K) = 0.25$ .

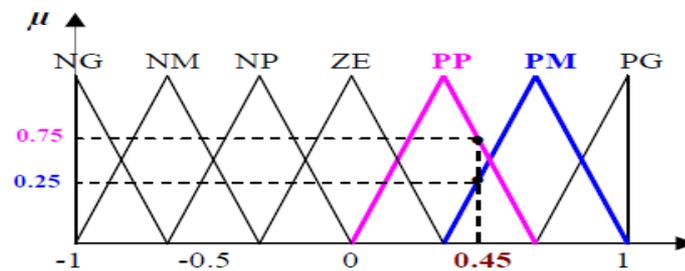


Fig.III.3 : exemple de fuzzification

### III.2.3.3 Règles d'inférence floue

Les règles d'inférence peuvent être décrites de plusieurs façons, linguistiquement, symboliquement ou bien par matrice d'inférence.

Les trois méthodes d'inférence les plus usuelles sont : **Max-Produit**, **somme-produit** et **Max-Min** (Implication de Mamdani) [17], cette dernière méthode la plus utilisée à cause de sa simplicité, elle réalise l'opérateur "ET" par la fonction "Min", la conclusion "ALORS" de chaque règle par la fonction "Min" et la liaison entre toutes les règles (opérateur "OU") par la fonction Max, [18-19].

### III.2.3.4 Défuzzification

Plusieurs stratégies de défuzzification existent. Les plus utilisées sont [20]:

#### III.2.3.4.1 Méthode du maximum

Comme son nom l'indique, la commande en sortie est égale à la commande ayant la fonction d'appartenance maximale.

La méthode du maximum simple, rapide et facile mais elle introduit des ambiguïtés et une discontinuité de la sortie (par fois on trouve deux valeurs maximales).

#### III.2.3.4.2 Méthode de la moyenne du maximum

Pour lesquelles la fonction d'appartenance issue de l'inférence est elle considérée, comme valeur de sortie, la moyenne de toutes les valeurs maximale [9].

#### III.2.3.4.3 Méthode du centre de gravité

La méthode du centre de gravité est l'une des méthodes les plus mentionnées dans la littérature. L'abscisse du centre de gravité peut être déterminée en utilisant la forme générale :

$$y = \frac{\sum_{l=1}^M v' \mu_{B'}(v')}{\sum_{l=1}^M \frac{\mu_{B'}(v')}{\sigma'^2}} \quad (\text{III.9})$$

Ou  $v'$  désigne le centre de gravité de la fonction d'appartenance de l'ensemble flou  $B'$  et de défuzzificateur évalue premièrement  $\mu_{B'}(v')$  et  $\sigma'$  est une mesure de support de la fonction d'appartenance pour la  $l^{\text{ième}}$  règle pour les fonctions d'appartenance gaussienne  $\sigma'$  est l'écart type.

### III.3. Les réseaux de neurones

#### III.3.1. Introduction

Les réseaux de neurones artificiels (RNA) sont des réseaux fortement connectés de processeurs une sortie unique sur la base des informations qu'il reçoit. Ils infèrent des propriétés émergentes permettant de solutionner les problèmes jadis qualifiés de complexe.

Dans un réseau, chaque sous-groupe fait un traitement indépendant des autres et transmet le résultat de son analyse au sous-groupe suivant. L'information donnée au réseau va donc se propager de la couche d'entrée à la couche de sortie, en passant soit par aucune, une ou plusieurs couches intermédiaires. Il est à noter qu'en fonction de l'algorithme d'apprentissage, il est aussi possible d'avoir une propagation de l'information à reculons.

#### III.3.2. Neurone biologique

Le neurone est une cellule composée d'un corps cellulaire et d'un noyau. Le corps cellulaire se ramifie pour former ce que l'on nomme les dendrites. Celles-ci sont parfois si nombreuses que l'on parle alors de chevelure dendritique ou d'arborisation dendritique. C'est par les dendrites que l'information est acheminée de l'extérieur vers le soma, corps du neurone [20]. L'information traitée par le neurone chemine ensuite le long de l'axone (unique) pour être transmise aux autres neurones. La transmission entre deux neurones n'est pas directe. En fait, il existe un espace intercellulaire de quelques dizaines d'Angstroms (10-9 m) entre l'axone du neurone afférent et les dendrites (on dit une dendrite) du neurone efférent. La jonction entre deux neurones est appelée la synapse (Figure III.4).

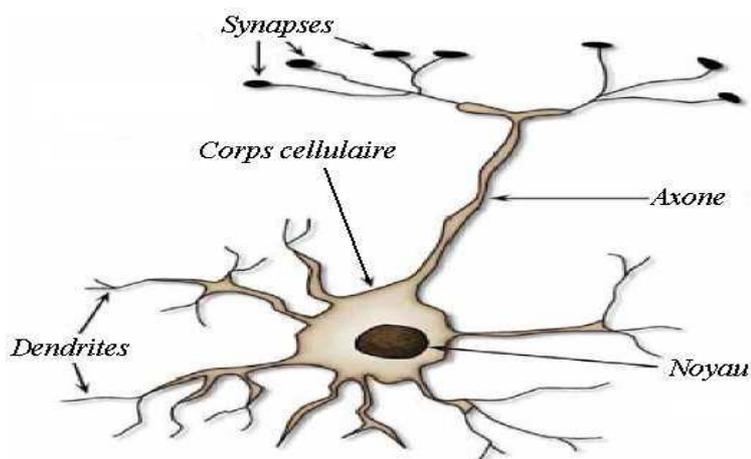


Figure III.4 : Schéma simplifié d'un neurone biologique.

### III.3.3.Neurone formel

Le neurone formel est un modèle mathématique simplifié du neurone biologique, il présente un certain nombre d'entrées, les dendrites, un corps traitant les entrées suivant la méthode du tout ou rien, et un axone véhiculant la réponse du neurone. La première modélisation d'un neurone découle des résultats des travaux significatifs de Mac Culloch et Pitts (1943) [21].

La (Figure III.5) représente un modèle de base d'un neurone formel.

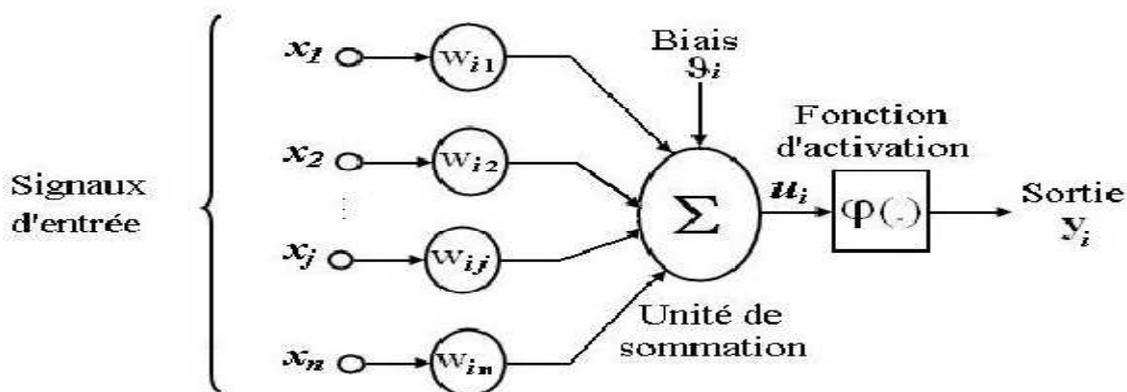


Fig. III.5 : Modèle de base d'un neurone formel

Le modèle de la figure (III.5) est composé de :

- ✓ Des entrées du neurone formel  $x_i, i = 1, 2, \dots, n$  ;

- ✓ Des paramètres de pondération  $w_{ij}$ ,
- ✓ De la fonction d'activation ou de seuillage (non-linéaire, forme en sigmoïde, ect...),
- ✓ Une sortie  $y$  du neurone formel.

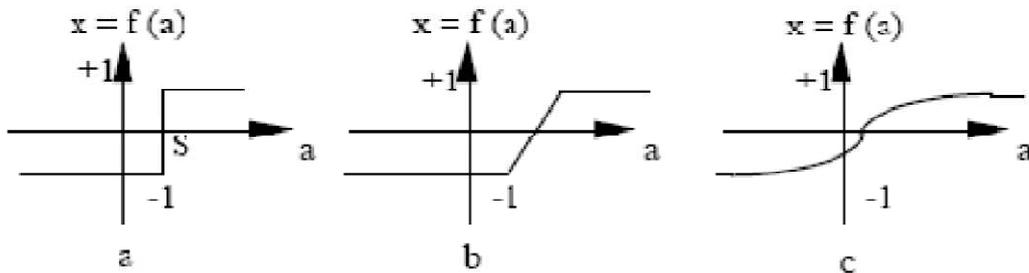
La sortie  $u_i$  du neurone formel est donnée par la relation suivant0

$$u_i = \sum_{j=0}^n w_{ji} x_j \quad (\text{III.10})$$

A partir de cette valeur, une fonction d'activation calcule la valeur de sortie  $j$  du neurone. C'est cette valeur qui sera transmise aux neurones avals :

$$y_i = \varphi(u_j) \quad (\text{III.11})$$

Il existe de nombreuses formes possibles pour la fonction d'activation. Les plus courantes sont présentées sur la (Figure III.6). On remarquera qu'à la différence des neurones biologiques dont l'état est binaire, la plupart des fonctions d'activations sont continues, offrant une infinité de valeurs possibles comprises dans l'intervalle  $[0, +1]$  (ou  $[-1, +1]$ ).



**Fig. III.6 :** Différents types de fonctions d'activation pour le neurone formel,

*a : fonction à seuil (S : valeur du seuil), b : linéaire par morceaux, c : sigmoïde.*

Nous constatons que les équations qui décrivent le comportement des neurones formels n'introduisent pas la notion de temps. En effet, et c'est le cas pour la plupart des modèles actuels de réseaux de neurones, nous avons à faire à des modèles à temps discret, dont le comportement des composants ne varie pas dans le temps.

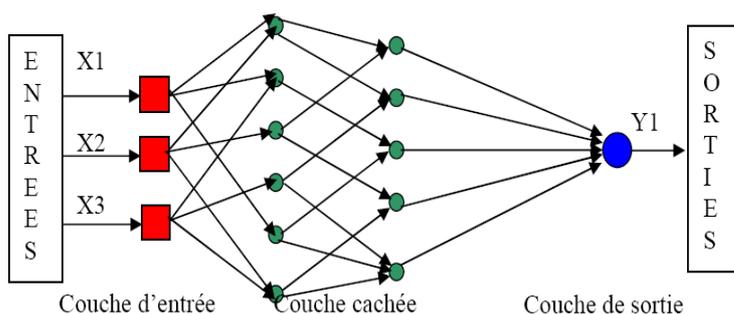
### III.3.4. Les architectes neurales

Les connexions entre les neurones qui composent le réseau, décrivent la topologie du modèle. Elle peut être quelconque, mais le plus souvent il est possible de distinguer une certaine régularité (réseau à connexion complète) [22].

### III.3.4.1. Les réseaux de neurones non bouclés

Un réseau de neurone non bouclé ou réseau unidirectionnel (feedforward networks) est représenté par un ensemble de neurones connectés entre eux telle que l'information circulant des couches entrées vers les couches sorties au travers d'éventuelles couches intermédiaires sans retour en arrière.

Ce type de réseau comprend deux groupes d'architecte : les réseaux Mono-Couche et les réseaux Multi-couche. La (Figure III.7) illustre un RNA Multi-Couche non bouclé

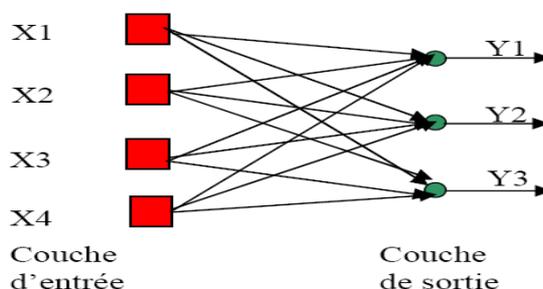


*Fig. III.7 : Réseau de neurone non bouclé*

#### III.3.4.1.1. Réseau non bouclé Mono-Couche :

Ce type de Réseau possède une couche d'entrée recevant les stimuli à traiter par l'intermédiaire des nœuds sources. Cette couche se projette en une couche de sortie composée de neurones (nœuds de calcul) transmettant les résultats du traitement au milieu extérieur .

La (Figure III.8) présente un réseau proactif Mono-Couche a 4 nœuds et 3 neouds de sortie. désignation Mono-Couche est attribuée à la couche de sortie .



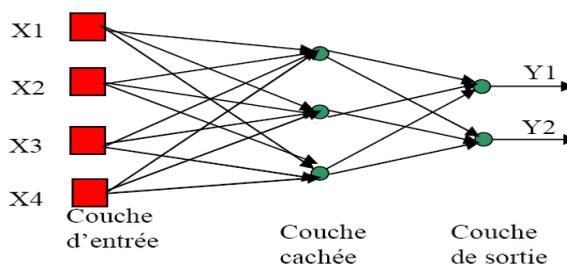
*Fig. III.8: Réseau non bouclé mono-couche*

### III.3.4.1.2 les Réseaux non bouclés Multi-Couches

Ce type de réseaux proctifs se carctérise par présence d'une ou de plusieurs couche cachées. dont les nœuds de calcul correspondants s'appellent neurones cachés . les couches cachées s'interposent entre l'entrée du réseau et sa sortie. leurs role est d'effectuer un prétraitement des signaus d'entrée, reçus par la couche d'entré en provenance du milieu extérieur, et de tranmettre les résultats correspondants à la couche de sortie ou seront déterminées les réponses finales du réseau.

Ce role de prétraitement fait que ,en ajoutant une ou plusieurs couches cachées, le réseau est capable d'extraire plus de propriétés statistiques que celles extraites d'un réseau similaire ayant moins de couche cachées. Ceci est utile pour réaliser des fonctions plus complexes que de simple séparations linéaires.

La figure ( III.9) illustre un réseau à une seule couche cachée comportant 4 unités d'entrées, 3 unités cachées et 2 unités de sortie (réseau 4-3-2).

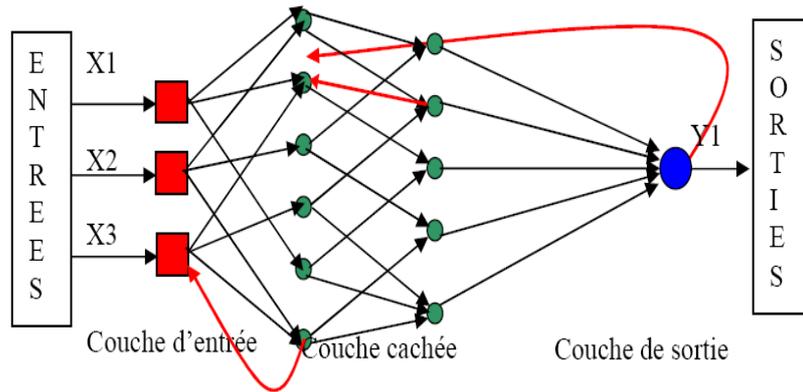


*Fig. III.9 : Réseau non bouclé Multi-Couche*

Ce réseau est dit complètement connecté dans le sens que chaque nœud d'une couche est connecté à tous les nœud de la couche adjacente suivante. Si éventuellement, des connexions manquaient entre des neurones de deus couche voisines, le réseau serait dit partiellement connecté.

### III. 3.4.2. Les réseaux de neurones bouclés

Les résaux récurrents (recurrent networks) possèdent une structure similaire à celle des réseaux unidirectionnels mais complétée par des connexions entre éléments de la même couche ou vers des couches amont. Ces réseaux sont assez puissants, car leur fonctionnement est séquentiel et adopte un comportement dynamique.



**Fig. III.10 :** Réseaux de neurones bouclés

### III.3.4.3. Les réseaux de neurones à connexions complexes [23].

Dans la catégorie des réseaux à connexions complexes se trouve tous les réseaux qui ne trouvent pas leur place dans les deux catégories précédemment citées. Nous pouvons mentionner les SOM (*Self Organizing Map*), LVQ (*Learning Vector Quantization*) ART (*Adaptive Resonance Theorie*), RCE (*Restrict-ed Coulomb Energy*), les réseaux probabilistes, etc.

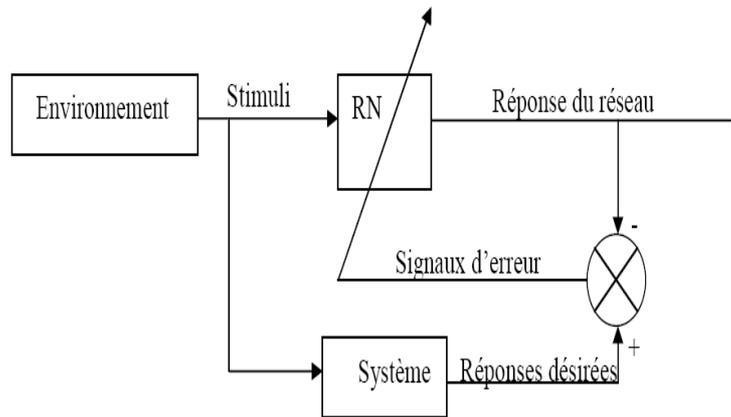
### III.3.5. Processus d'apprentissage

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

L'apprentissage est la caractéristique principale des réseaux de neurones et il peut se faire de différentes règles et manières [24].

#### III.3.5.1. Apprentissage supervisé [25].

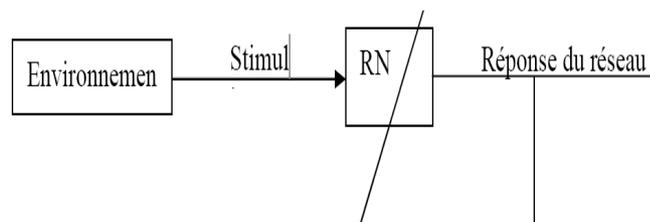
L'apprentissage supervisé consiste à présenter au réseau une série d'exemples sur ses entrées, ainsi le réseau calcule sur ses sorties un vecteur d'état qui est comparé au vecteur des états désirés. Une erreur est calculée entre le vecteur des sorties désirées et le vecteur des sorties calculées par le réseau. L'algorithme d'apprentissage adapte les poids du réseau de façon à corriger son comportement comme le montre la figure ( III.11 )



**Fig. III.11:** Apprentissage supervisé

### III.3.5.2. Apprentissage non supervisé

L'apprentissage est qualifié de non supervisé lorsque, seules les valeurs d'entrée sont disponibles. Il repose sur un critère interne de conformité du comportement du réseau par rapport à des spécifications générale et non sur des observations externes. Dans ce cas, l'apprentissage est basé sur des probabilités. Le réseau va se modifier en fonction des régularités statiques de l'entrée. L'algorithme d'apprentissage exploite pour se faire une mesure prédéterminée de la qualité de représentation de la connaissance afin d'ajuster les paramètres du réseau[25]. Ce type d'apprentissage est représenté schématiquement en figure (III.12).



**Fig. III.12:** apprentissage non supervisé

## III.4. NEURO-FLOU

### III.4.1. Introduction :

Les systèmes neuro-flous sont des systèmes flous formés par un algorithme d'apprentissage inspiré de la théorie des réseaux de neurones. La technique d'apprentissage opère en fonction de l'information locale et produit uniquement des changements locaux dans le système flou d'origine. Les règles floues codées dans le système neuro-flou représentent les

échantillons imprécis et peuvent être vues en tant que prototypes imprécis des données d'apprentissage, [26].

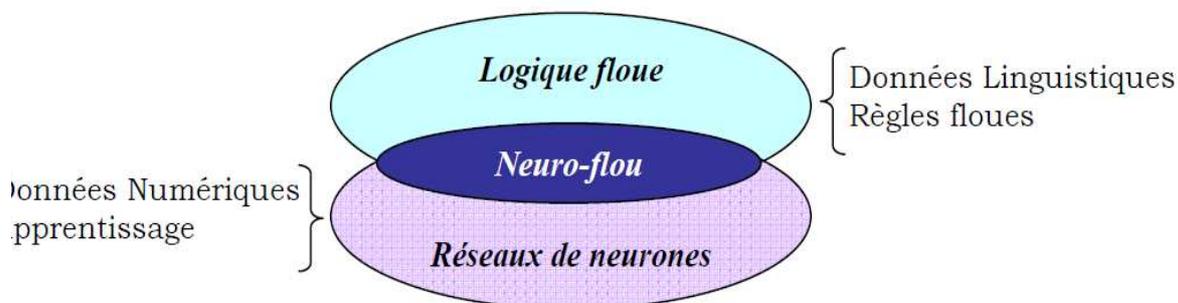


Fig. III.13. : Principe du Neuro-flou

#### III.4.2. Avantages et inconvénients de la logique floue et des réseaux de neurones :

L'utilisation simultanée des réseaux de neurones et de la logique floue, permet de tirer les avantages des deux méthodes : les capacités d'apprentissage de la première et la lisibilité et la souplesse de la seconde.

Afin de résumer l'apport du neuro-flou, le Tableau (III.1) regroupe les avantages et les inconvénients de la logique floue et des réseaux de neurones, [26].

Les systèmes neuro-flous sont créés afin de synthétiser les avantages et de surmonter les inconvénients des réseaux neuronaux et des systèmes flous

Les algorithmes d'apprentissage peuvent être employés pour déterminer les paramètres des systèmes flous. Ceci revient à créer ou améliorer un système flou de manière automatique, au moyen des méthodes spécifiques aux réseaux neuronaux.

Un aspect important est que le système reste toujours interprétable en termes de règles floues, vu qu'il est basé sur un système flou.

Réseaux de neurones	Logique floue
<b>Avantages</b>	
<ul style="list-style-type: none"> <li>• Aucune connaissance antérieure sur les règles.</li> <li>• Le modèle mathématique non requis.</li> <li>• Plusieurs algorithmes d'apprentissage sont disponibles.</li> </ul>	<ul style="list-style-type: none"> <li>• La connaissance antérieure sur les règles peut être utilisée .</li> <li>• Le modèle mathématique non requis.</li> <li>• Une interprétation et implémentation simple.</li> </ul>

Inconvénients	
<ul style="list-style-type: none"> <li>• Boite noire (manque de traçabilité )</li> <li>• L'adaptation aux environnements différents est difficile et le réapprentissage est souvent obligatoire.</li> <li>• La connaissance antérieure ne peut pas être employée (apprentissage a partir de zéro).</li> <li>• Aucune garantie sur la convergence de l'apprentissage.</li> </ul>	<ul style="list-style-type: none"> <li>• Ne peut pas apprendre.</li> <li>• Les règles doivent être disponibles .</li> <li>• Adaptation difficile au changement de l'environnement.</li> <li>• Aucune méthodes formelle pour l'ajustement .</li> </ul>

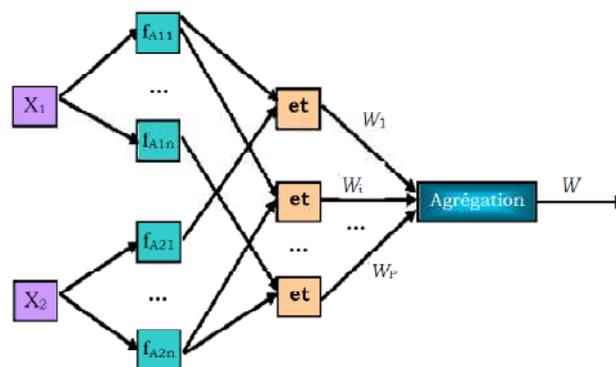
**Tableau. III.1 :** Comparaison entre la logique floue et les réseaux de neurone

### III.4.3. Méthodes neuro-floues :

Plusieurs méthodes ont été développées depuis 1988 et sont le plus souvent orientées vers la commande de systèmes complexes et les problèmes de classification. Il existe ainsi trois méthodes neuro-floues.

#### III.4.3.1. Première méthode neuro-floue :

Cette méthode neuro-floue est basée sur le codage du système d'inférence floue sous la forme d'un réseau de neurones multicouches dans lequel les poids correspondent aux paramètres du système. L'architecture du réseau dépend du type de règle et des méthodes d'inférence, d'agrégation et de défuzzification choisies, figure (III.14).



**Fig. III.14.:** Premières architecture des réseaux neuro-floues

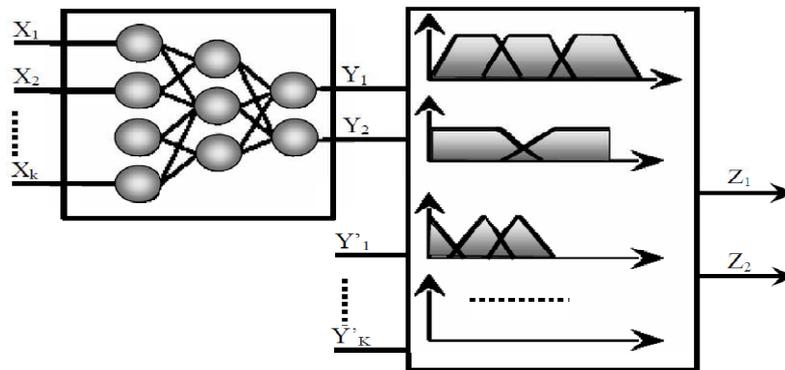
### III.4.3.2. Deuxième méthode neuro-floue :

Elle consiste à utiliser les réseaux de neurones pour remplacer chacune des composantes d'un système de commande floue. Ces réseaux sont destinés à l'apprentissage des fonctions d'appartenance, au calcul de l'inférence et à la réalisation de la phase d'agrégation et de défuzzification. Ils peuvent réaliser l'extraction des règles floues en analysant la corrélation qui existe entre les entrées et les sorties du réseau de neurones.

Ces approches permettent de résoudre deux problèmes importants de la logique floue : le déterminisme des fonctions d'appartenance et l'adaptation à l'environnement du système.

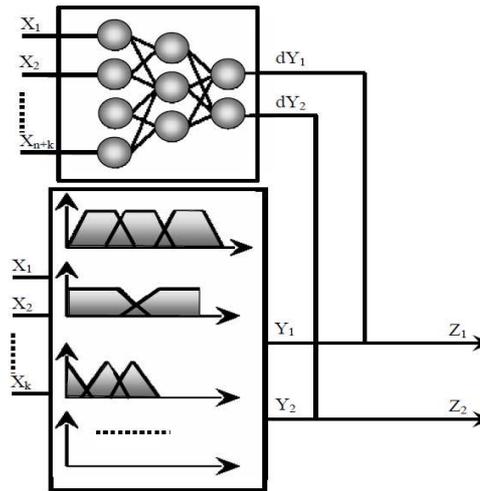
### III.4.3.3. Troisième méthode neuro-floue :

Cette troisième méthode utilise des réseaux de neurones et des systèmes flous associés en série ou en parallèle. Plusieurs variantes d'utilisation sont ainsi possibles :



*Fig. III.15. : Troisième architecture des réseaux Neuro-Flou Réalisation en série*

- Le réseau de neurones fonctionne en amont du système flou. Les variantes d'entrées du système flou sont déterminées à partir des sorties du réseau de neurones (dans le cas où elles ne sont pas mesurables directement) ou encore un réseau de neurones effectue une tâche de classification ou de reconnaissance de formes, suivie d'un système flou d'aide à la décision.
- Un réseau de neurones qui fonctionne en aval du système flou, dans le but d'ajuster les sorties d'un système de commande floue à de nouvelles connaissances obtenues, les variables de sorties étant les erreurs sur les variables de sortie du système flou, [27].



**Fig. III.16. :** Troisième architecture des réseaux Neuro-Flou Réalisation en parallèle

Notre étude s'intéressera particulièrement au premier type d'architecture des réseaux neuro-floues, celui-ci présente une certaine facilité d'implémentation neuronale moyennement satisfaisante, aussi va-t-il nous permettre de déterminer les paramètres des fonctions d'appartenances. Nous allons présenter l'architecture du contrôleur ANFIS dans ce qui suit.

### III.4.4 systèmes d'inférence neuro-flou adaptatifs ANFIS

#### III.4.4.1 Définition

L'ANFIS (Adaptive Network Based Fuzzy Inference System) se reporte à la manière d'appliquer de divers techniques d'apprentissage développées à la littérature du réseau de neurone aux systèmes d'inférence flou.

#### III.4.4.2. Les avantages de l'ANFIS [28]

Les avantages de cette technique sont :

- ✓ Réduction de la taille de la base de règles : il suffit d'avoir des règles générales, les détails seront fournis par le RN.
- ✓ Réduction de la complexité de l'apprentissage : le RN doit simplement apprendre les cas particuliers ou les exceptions, pas le problème complet.
- ✓ Efficacité immédiate dès le début de l'apprentissage et possibilité d'éviter des comportements initiaux erratiques.
- ✓ Exploitation de la connaissance disponible, grâce à la base de règles.

### III.4.4.3. architecture de l'ANFIS

L'ANFIS c'est un système d'inférence adaptatif neuro-flou qui consiste à utiliser un réseau neurone de type MLP à 5 couches pour lequel chaque couche correspond à la réalisation d'une étape d'un système d'inférence floue de type Takagi Sugeno. Pour la simplicité, nous supposons que le système d'inférence flou à deux entrées  $x$  et  $y$ , et à comme une sortie  $f$ . Supposer que la base de règle contient deux règles floues de type Takagi-Sugeno.

$$\text{Règle1 : SI } x \text{ est } A_1 \text{ et } y \text{ est } B_1 \text{ ALORS } f_1 = p_1 x + q_1 y + C_1 \quad (\text{III.12})$$

$$\text{Règle2 : SI } x \text{ est } A_2 \text{ et } y \text{ est } B_2 \text{ ALORS } f_2 = p_2 x + q_2 y + C_2 \quad (\text{III.13})$$

L'ANFIS à une architecture posée par cinq couches comme représenté sur la figure (III.17).

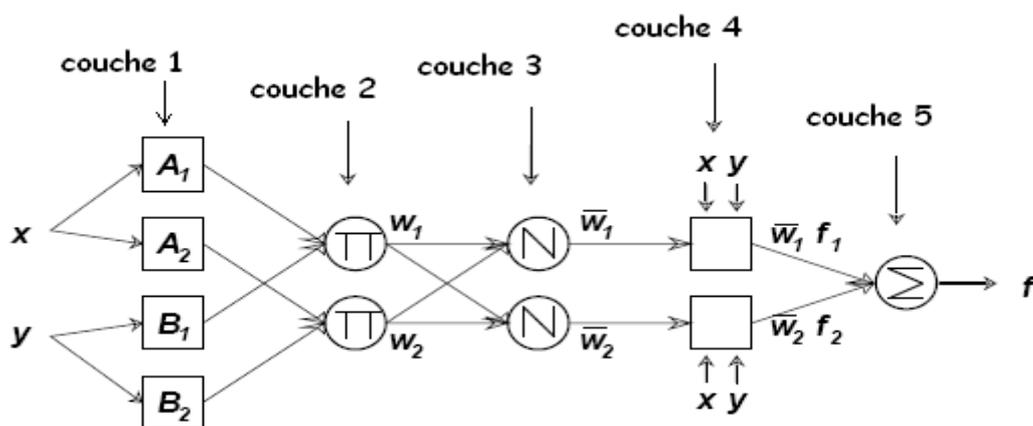


Fig. III.17. : L'Architecture générale de l'ANFIS.

Une architecture classique peut être décrite de la manière suivante :

**La première couche :** comporte autant de neurones qu'il y'a de sous ensembles flous dans le système d'inférence représenté. Chaque neurone d'un sous ensemble flou particulier calcule le degré de vérité par sa fonction de transfert. La seule restriction sur le choix de cette fonction concerne sa dérivabilité. En retrouve dans la littérature, l'utilisation, des fonctions gaussiennes ou les paramètres modifiables sont le centre et la pente de la gaussienne (variance).

La fonction d'activation des neurones  $i$  de la première couche :

$$O_i^1 = \mu_{Ai}(x) \quad (\text{III.14})$$

avec  $x$  est l'entrée au neurone  $i$ , et  $Ai$  est un sous ensemble floue correspondant au variable  $x$ . En d'autres façon,  $O_i^1$  est la fonction d'appartenance du  $Ai$  qu'indique le degré auquel donné  $x$  satisfait le quantifier  $Ai$ . Nous choisissons  $\mu_{Ai}(x)$  pour être en forme de (Gaussien, triangle, trapézoïdal) avec le maximum égal à 1 et le minimum égal à 0. tel que les fonctions généralisées de ces formes sont :

Triangle:

$$\mu(x) = \max \left( \min \left( \frac{x-a}{b-a}, \frac{c-x}{c-b}, 0 \right) \right) \quad (\text{III.15})$$

Trapézoïdale :

$$\mu(x) = \max \left( \min \left( \frac{x-a}{b-a}, 1, \frac{d-x}{d-c}, 0 \right) \right) \quad (\text{III.16})$$

Gaussienne :

$$\mu(x) = \exp \left( -\frac{(x-m)^2}{2\sigma^2} \right) \quad (\text{III.17})$$

Où  $\{a, b, c, \sigma\}$  est l'ensemble des paramètres. Pendant que les valeurs de ces paramètres changent, les fonctions en forme précédente changent en conséquence, de ce fait présenter de diverses formes de fonction d'appartenance sur la variable linguistique  $Ai$ . Les paramètres dans cette couche désigné sous le nom des paramètres de fonction d'appartenance [29].

**La deuxième couche:** qu'on dit première couche cachée, sert à calculer le degré d'activation des prémisses. Chacun des neurones de cette couche représentent la prémisse d'une règle [29]. Ils reçoivent le degré de vérité des différents sous-ensembles flous composant cette prémisse et ont en charge le calcul de son propre degré de vérité. Les fonctions d'activation utilisées pour ces neurones dépendant des opérateurs présents dans les règles (*ET* ou *OU*).

La fonction d'activation des neurones  $i$  de la première couche :

$$O_k^2 = w_k = \mu_{Ai}(x) \times \mu_{Bj}(y) \quad (\text{III.18})$$

Ou  $k$  : représente le nombre de règle,  $i$  : représente le nombre de partition de  $x$ , et  $j$  : le nombre de partition de  $y$ .

**La troisième couche:** deuxième couche cachée, normalise les degrés d'activations des règles.

Chaque neurone dans cette couche est alors, un neurone de normalisation. L' $i$ ème neurone calcule le rapport entre le  $i$ ème poids de règles et la somme de tous les poids des règles.

$$O_k^3 = \bar{w}_k = \frac{w_k}{\sum w_i} \tag{III.19}$$

**La quatrième couche:** troisième couche cachée, sert à déterminer la partie conséquence des règles par l'intermédiaire des paramètres  $(p, q, c)$ . La fonction de chaque neurone dans cette couche peut se formuler comme suite:

$$O_k^4 = \bar{w}_k f_k = \bar{w}_k \times (p_k x + q_k y + c_k) \tag{III.20}$$

Où  $\bar{w}_k$  est la sortie de couche précédent, et  $(p_i, q_i, c_i)$ . est l'ensemble des paramètres. Ces paramètres sont désignés sous le nom "les paramètres conséquents". [29]

**La couche de sortie :** cette couche contient un seul neurone, dont leur fonction est de calculer la sortie globale (résultant), comme addition de tous les signaux entrants, c'est-à-dire :

$$O_k^5 = \sum_{k=1}^n \bar{w}_k f_k \tag{III.21}$$

La figure (III.18) représente un système ANFIS, à 2 entrées chaque entrée repartie en trois sous ensemble floue et 9 règles.

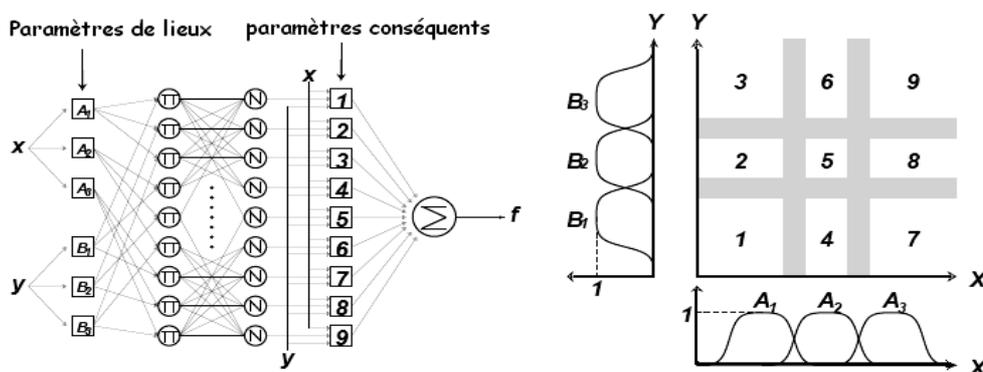


Fig. III.18 : Architecture d'un ANFIS à 2 entrées avec 9 règles.

<i>Couches</i>	Type des couches	Le nombre de neurone dans la couche
<i>Couche 0</i>	couche d'entrées	$n$
<i>Couche 1</i>	Fuzzification	$(p.n)$
<i>Couche 2</i>	inférence	$p^n$
<i>Couche 3</i>	La normalisation	$p^n$
<i>Couche 4</i>	Linéarisation des fonctions	$p^n$
<i>Couche 5</i>	sortie	1

**Tableau. III.2 :** les différentes couches d'un système ANFIS

Tel que :

$p$  : le nombre des sous ensembles flous d'entrée (partition flou).

$n$  : le nombre des entrées.

Noter que les neurones dans ANFIS rencontrent généralement:

- fonction d'appartenance définie par différentes formes.
- Règles (habituellement produit).
- Normalisation (division de somme et d'arithmétique).
- Fonctions (régressions linéaires et multiplication avec  $\bar{w}$ , tel que  $\bar{w}$  est la normalisation du poids  $w$ ).
- la sortie (Somme Algébrique).

#### III.4.4.4. Algorithme d'apprentissage de l'ANFIS

Le système ANFIS exploite le mécanisme d'apprentissage des réseaux de neurone sur des techniques d'inférence floues. D'une autre façon, L'ANFIS est un système d'inférence floue (SIF), dont l'ajustement des paramètres des fonctions d'appartenances est confié à l'algorithme d'apprentissage proprement dite "rétro propagation", ou en combinaison avec d autre type d'algorithmes tel que le moindre carré.

Dans l'architecture ANFIS proposée sur la figure (III.23), la sortie globale peuvent être exprimé en tant que des combinaisons linéaires des paramètres conséquents [25]. Avec plus précision, la conclusion (la sortie) sur la figure (III.23) peut être récrit comme :

$$f = \frac{w_1}{w_1+w_2} f_1 + \frac{w_2}{w_1+w_2} f_2 \tag{III.22}$$

$$= (\overline{w_1}x)p_1 + (\overline{w_1}y)q_1 + (\overline{w_1})c_1 + (\overline{w_2}x)p_2 + (\overline{w_2}y)q_2 + (\overline{w_2})c_2 \tag{III.23}$$

La sortie est une fonction linéaire des paramètres conséquences  $(p, q, C)$ . ANFIS est représentation paramétrique deux ensembles de paramètres:  $S1$  et  $S2$  tel que :

$S1$  représente les paramètres des ensembles flous utilisés pour la fuzzification dans la première couche de système ANFIS

$$S1 = \{ \{a_{11}, b_{11}, c_{11}\}, \{a_{12}, b_{12}, c_{12}\}, \dots, \{a_{1p}, b_{1p}, c_{1p}\}, \dots, \{a_{np}, b_{np}, c_{np}\} \}$$

Ou  $p$  est le nombre de partition floue de chacun des variables d'entrées et  $n$  est le nombre de variables d'entrées.

$S2$  représente les coefficients des fonctions linéaires (les paramètres conséquences)

$$S2 = \{ p_1, p_2, p_3 \dots, q_1, q_2, q_3 \dots, C_1, C_2, C_3 \dots \} \tag{III.24}$$

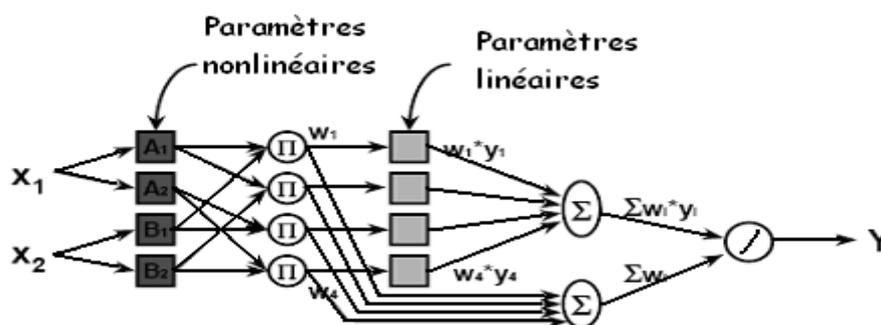


Fig. III.19. : Illustration de méthode d'apprentissage Hybride

Type de passage	Passage Vers l'avant	Passage En arrière
Paramètres affectés		
Paramètre de fonction d'appartenance (non linéaire $a_i, b_i, c_i$ )	fixe	Rétro propagation
Paramètre de coefficient (Linéaire $p, q, C$ )	moindres carrés	fixe

Tableau III.3 : L'une des techniques d'ajustement des paramètres de l'ANFIS

Le passage en avant:  $S1$  est fixe et  $S2$  est calculé en utilisant l'algorithme de moindre carré sur l'erreur (LSE). (Le LSE est appliqué seulement une fois lorsque commencer à obtenir les valeurs initiales des paramètres conséquents).[30]

Passage en arrière:  $S2$  est fixe et  $S1$  est calculé en utilisant l'algorithme de Rétro Propagation qui est basé sur l'algorithme de gradient. [30]

#### III.4.4.4.1. Algorithme de Rétro Propagation

Mesure d'erreur  $E_k$  pour l'entrée  $k^{ieme}$  donnée d'apprentissage:

$$E_k = \sum_{i=1}^{N(L)} (d_i - x_{L,i})^2 \quad (\text{III.25})$$

$N(L)$  : Nombre de neurones dans la couche  $L$

$d_i$ :  $i$ eme composant de vecteur de sortie désiré

$x_{L,i}$ :  $i$ eme composant de vecteur de sortie de réel (de l'ANFIS)

$$E = \sum_{k=1}^k E_k \quad (\text{III.26})$$

La mesure d'erreur globale  $E$ :

Pour chaque paramètre  $w_i \in s_1$  la modification soumit à la relation suivant :

$$\Delta w_i = -\mu \frac{dE}{dw_i} \quad (\text{III.27})$$

Où  $\mu$  est une constante positive appelée pas d'apprentissage

#### III.4.4.4.2. Algorithme de moindre carré [31]

*Formulation du problème :*

On a le modèle linéaire suivant:

$$\begin{aligned} d(i) &= \sum_{l=0}^{L-1} h_{s,l} x(i-l) + u(i) \\ &= h_s^T X(i) + u(i) \end{aligned} \quad (\text{III.28})$$

$d(i)$  est le signal observé (désiré) a l'instant  $i$  obtenu a partir du signal d'entrée  $x(i)$ .  $h_{s,L}$  sont les paramètres inconnus du modèle et  $u(i)$  représente le bruit de mesure qui est

une variable aléatoire (non observable). Il est d'usage de supposer que  $u(i)$  est blanc de moyenne nulle, et dont la variance est  $\sigma_u^2$ .

Notre objectif est d'estimer les paramètres  $h_{s,l}$  en utilisant le critère des moindres carrés suivant:

$$J(n) = \sum_{i=0}^n \gamma^{n-1} [d(i) - h^T(n)X(i)]^2 \quad (\text{III.30})$$

Où

$$X(i) = [x(i) \ x(i-1) \ \dots \ x(i-L+1)]^T \quad (\text{III.31})$$

Est le signal d'entrée à l'instant  $i$ , et

$$h(n) = [h_0(n) \ h_1(n) \ \dots \ h_{L-1}(n)]^T$$

est le filtre RIF à l'instant  $n$ .  $\lambda$  est un facteur de pondération qui prend toujours une valeur positive:  $0 < \lambda \leq 1$ . Ce facteur est aussi appelé facteur d'oubli car il sert à oublier les données qui correspondent à un passe distant. Le cas particulier  $\lambda = 1$  correspond à une mémoire infinie.

### Les équations normales

Le problème pose est de déterminer le vecteur des coefficients  $h(n)$  qui minimise  $J(n)$ . La solution est obtenue en calculant les dérivées de la fonction cout  $J(n)$  par rapport aux éléments  $h_l(n)$  et en les égalant à zéro:

$$\sum_{i=0}^n \gamma^{n-1} [d(i) - h^T(n)X(i)]X(i) = 0_{L \times 1} \quad (\text{III.32})$$

Soit :

$$\sum_{i=0}^n \gamma^{n-i} d(i)X(i) = \sum_{i=0}^n \gamma^{n-i} [X(i)X^T(i)]h(n) \quad (\text{III.33})$$

Il vient alors

$$P(n) = R(n)h(n) \quad (\text{III.34})$$

$$h(n) = R^{-1}(n)P(n) \quad (\text{III.35})$$

avec :

$$R(n) = \sum_{i=0}^n \gamma^{n-i} [X(i)X^T(i)] \quad (\text{III.36})$$

$$P(n) = \sum_{i=0}^n \gamma^{n-i} d(i)X(i) \quad (\text{III.37})$$

Les équations précédentes peuvent se calculés de la manière récursive:

$$\begin{aligned} R(n) &= \gamma \sum_{i=0}^{n-1} \gamma^{n-1-i} X(i)X^T(i) + X(n)X^T(n) \\ &= \gamma R(n-1) + X(n)X^T(n) \end{aligned} \quad (\text{III.38})$$

On a aussi :

$$P(n) = \gamma P(n-1) + d(n)X(n) \quad (\text{III.39})$$

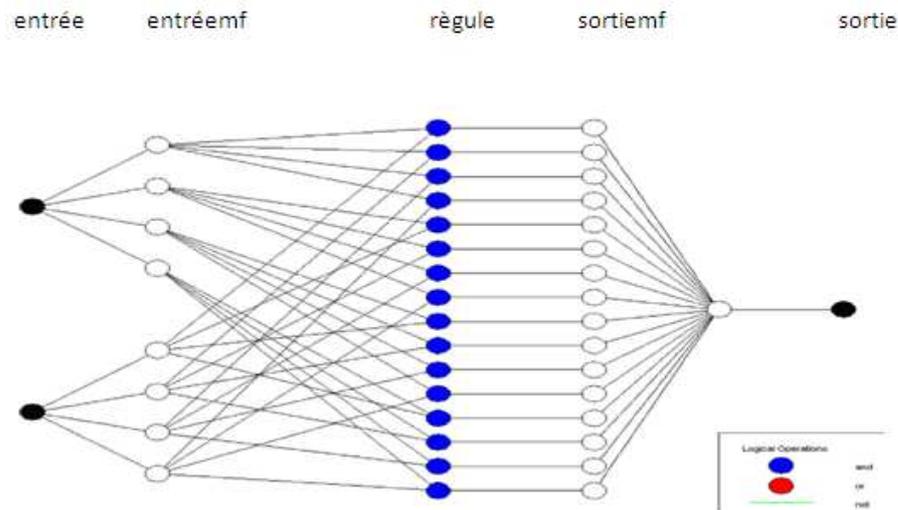


Fig. III.22. : Archetière de contrôleur ANFIS exploité sur la boîte à outil de logiciel MATLAB

### III.5.Conclusion

Ce chapitre est consacré à la présentation des concepts fondamentaux de la logique floue et des réseaux de neurone en montrant les avantages et les inconvénients de chaque théorie dans le domaine de la commande. Par ailleurs, Une structure hybride Neuro-Floue est présentée tout en profitant de la contribution de chaque théorie dans la commande des systèmes non linéaires.

# chapitre

# IV

**Commande Neuro-Floue du simulateur d'hélicoptère**  
**(TRMS)**

# Chapitre IV

## Commande Neuro-Floue du simulateur d'hélicoptère (TRMS)

### IV.1 Introduction

Dans ce chapitre on s'intéresse à la commande du TRMS par des régulateurs Neuro-flous conçus à partir des modèles découplés du simulateur de la partie verticale et horizontale développés dans le chapitre II. Tel que, chaque régulateur sera synthétisé et paramétré séparément selon l'algorithme présenté dans le chapitre précédent. Des travaux de simulation sur les deux schémas de commande seront effectués pour différents types de consigne afin de montrer l'apport d'une approche Neuro-floue à la commande des systèmes non linéaires. Par la suite, ces deux régulateurs seront appliqués au modèle couplé multivariable du TRMS avec la même configuration, le schéma de commande sera également simulé pour des différentes consignes de l'angle d'élévation et d'orientation dans le but de tester la robustesse de ces régulateur vis-à-vis le couplage qui existe entre la partie verticale et la partie horizontale.

### IV.2 Architecture du système de commande Neuro-flou (ANFIS)

#### IV.2.1 Schéma de commande

Le système de commande par régulateur Neuro-flou monovariante de chaque partie du TRMS peut être schématisé par la figure suivante :

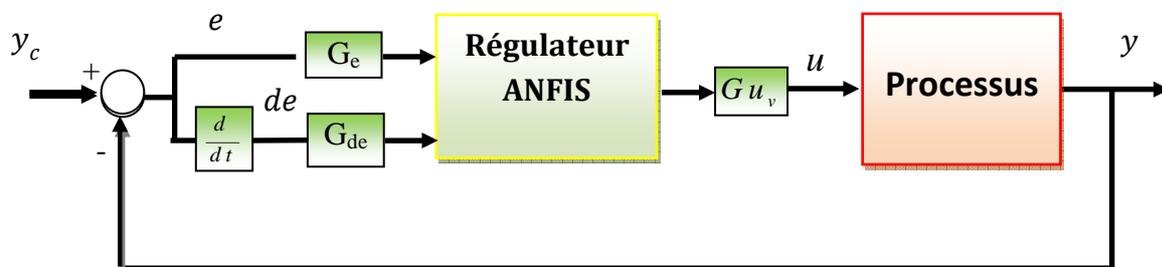


Fig. IV.1 : Schéma de commande par un régulateur neuro-flou (ANFIS)

Le régulateur neuro-floue utilisé est de type Sugeno, qui présente deux avantages majeurs par rapport à celui de type de Mamdani :

- Temps de calcul de la commande réduit (le temps consommé par la procédure de défuzzification est évité), moins de règles nécessaires pour réaliser une tâche donnée,
- Facilité de mettre sous forme de réseau adaptatif non linéaire

## IV.2.2 Conception du contrôleur ANFIS

### IV.2.2.1 Structure

Notre contrôleur neuro-floue est un contrôleur **ANFIS** qui utilise un système d'inférences floues (SIF) du type Takagi Sugeno. Il est composé de cinq couches comme il est représenté sur les figures (III.22). Ce contrôleur permet la génération des règles floues :

$$\text{R\`egle1 : SI } e \text{ est } \mathbf{A}_1 \text{ et } de \text{ est } \mathbf{B}_1 \quad . \quad \text{ALORS } U_1 = F(e, de)$$

•  
•  
•

$$\text{R\`egle49 : SI } e \text{ est } \mathbf{A}_7 \text{ et } de \text{ est } \mathbf{B}_7 \quad . \quad \text{ALORS } U_{49} = F(e, de)$$

Où  $e$  et  $de$  sont des variables d'entrée et  $A_1, A_2, \dots, A_7$  et  $B_1, B_2, \dots, B_7$  sont des ensembles flous. Lorsque  $U = \text{constante}$ , on obtient un modèle de Sugeno d'ordre zéro ; et lorsque la commande  $U$  est une combinaison linéaire des entrées :  $U = aE + b dE + c$ , on obtient un modèle de Sugeno de premier ordre.

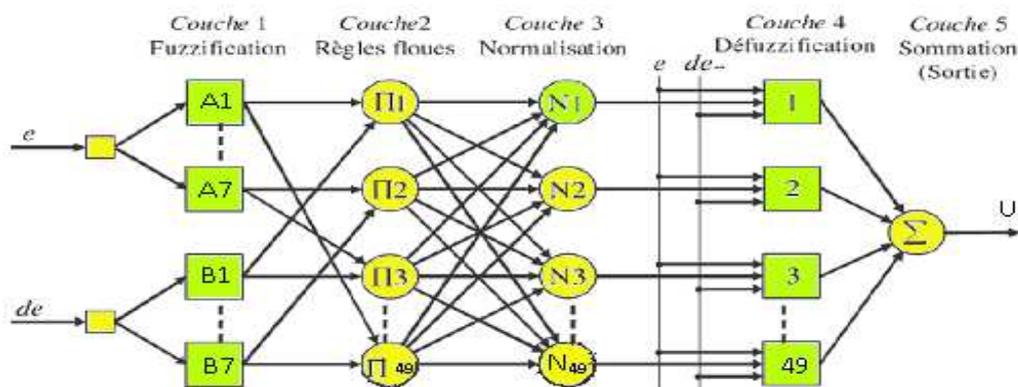


Fig. IV.2 : Architecture du modèle ANFIS.

Comme a été déjà présenté en théorie les cinq couches du réseau ANFIS ont pour rôle :

### Couche 1 :

La sortie de chaque nœud dans cette couche est exprimée par :

$$O_{1,i} = \mu_{A_i}(e) \quad , \text{ pour } i = 1, \dots, 7 \quad (IV.1)$$

$$O_{1,i} = \mu_{B_{i-7}}(de) \quad , \text{ pour } i = 8, \dots, 14 \quad (IV.2)$$

Tel que  $\mu_{A_i}$  et  $\mu_{B_i}$  sont les fonctions d'appartenances des ensembles flous  $A_i$  et  $B_i$  respectivement.

### Couche 2 :

Les sorties des nœuds de cette couche sont exprimées par (49 nœuds):

$O_{2,1} = O_{1,1} \cdot O_{1,8} = W_1$	jusqu'à	$O_{2,7} = O_{1,1} \cdot O_{1,14} = W_7$
$O_{2,8} = O_{1,2} \cdot O_{1,8} = W_8$	jusqu'à	$O_{2,14} = O_{1,2} \cdot O_{1,14} = W_{14}$
		.
		.
$O_{2,42} = O_{1,7} \cdot O_{1,8} = W_{43}$	jusqu'à	$O_{2,7} = O_{1,7} \cdot O_{1,14} = W_{49}$

### Couche 3 :

La sortie du nœud est exprimée par :

$$O_{3,i} = \bar{W}_i = \frac{W_i}{\sum_{k=1}^{49} W_k} \quad i = 1, \dots, 49 \quad (IV.3)$$

### Couche 4 :

Les nœuds de cette couche sont à paramètres ajustables, leurs fonctions sont de la forme :

$$O_{4,i} = \bar{W}_i \cdot (p_i \cdot e + q_i \cdot de + c_i) \quad i = 1, \dots, 49 \quad (IV.4)$$

Tel que  $p$ ,  $q$  et  $c$  sont les paramètres à ajuster.

### Couche 5 :

Cette couche contient un seul nœud qui réalise la somme des sorties des nœuds de la couche précédente :

$$O_5 = \sum_{i=1}^{49} O_{4,i} \quad (IV.5)$$

Les paramètres des prémisses sont supposés fixes, donc  $O_5$  peut être écrite sous la forme :

$$O_5 = \sum_{i=1}^{49} (\bar{W}_i \cdot e) \cdot p_i + (\bar{W}_i \cdot de) \cdot q_i + (\bar{W}_i) \cdot c_i \quad i = 1, \dots, 49 \quad (IV.6)$$

#### IV.2.2.2 Apprentissage

Pour estimer les paramètres du réseau de neurone, nous avons utilisé un algorithme hybride (figure IV.3) qui a été présenté dans le chapitre précédent, il détermine les paramètres liés aux fonctions d'appartenance, et les paramètres conséquents.

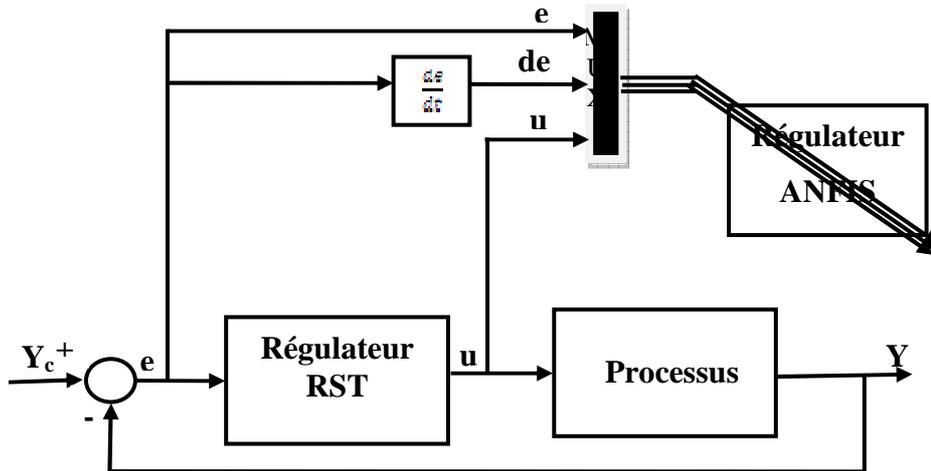


Fig. IV.3 : Schéma d'apprentissage du régulateur neuro-flou (ANFIS)

#### IV.2.2.3 Implémentation

Une fois l'étape d'apprentissage est achevée avec une erreur tolérée, le régulateur est appliqué sur le processus à commander pour tester son efficacité (obtention des performances désirées) et sa robustesse vis-à-vis les perturbations et l'effet de la non linéarité.

Ainsi, on peut résumer les étapes de conception d'un contrôleur ANFIS dans l'éditeur de neuro-flou sous Matlab par l'organigramme suivant :

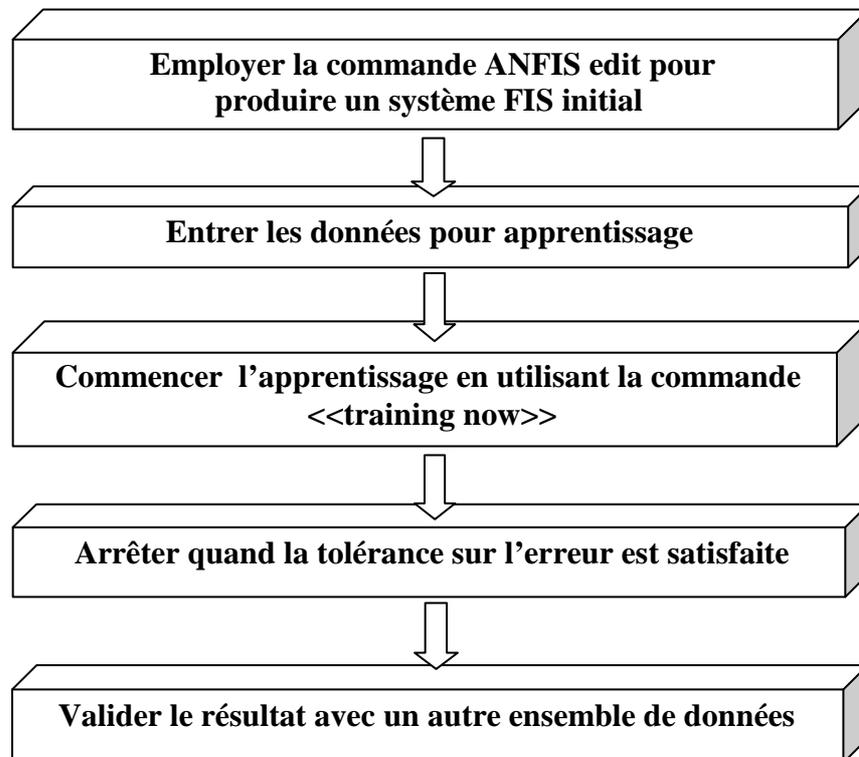


Fig. IV.4 : Organigramme de conception d'un système neuro-flou sous MATLAB.

### IV.3 Commande par Régulateur Neuro-Flou du Modèle Vertical (RNFVER)

#### IV.3.1 Schéma de commande

Le schéma de commande du modèle non linéaire (équation II.33) de la partie verticale du TRMS est donné par la figure (IV.5) tel que la grandeur à commander est l'angle d'élévation  $\alpha_v$  du TRMS contrôlée par la tension d'alimentation du rotor principal  $u_v$ ,

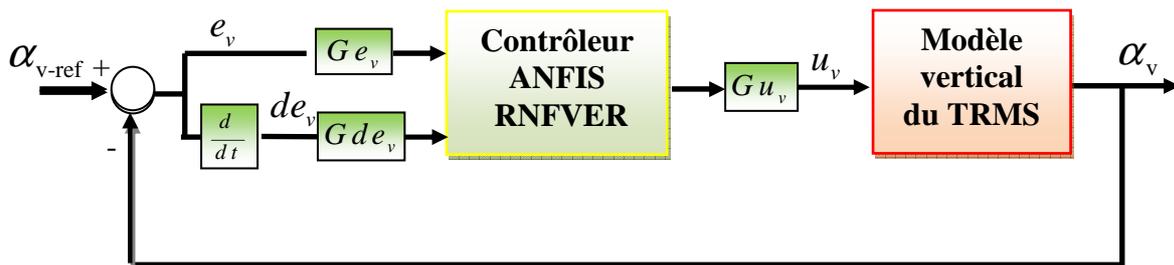


Fig. IV.5: Schéma de commande de la partie verticale du TRMS par un régulateur neuro-flou RNFVER

#### IV.3.2 Conception du régulateur RNFVER

La conception du régulateur RNFVER de la partie verticale est faite selon l'algorithme présenté précédemment pour neuf types de fuzzification des variables d'entrées ( $e_v$  et  $de_v$ ) afin de montrer l'influence du nombre d'ensemble flou et le type de la fonction d'appartenance sur la qualité d'apprentissage du réseau ANFIS. Les résultats de simulation ont permis d'obtenir le tableau suivant :

Type de fonction / Nombre De fonction	Fonction triangulaire	Fonction trapézoïdale	Fonction gaussienne
Trois fonctions	0.1205	0.1130	0.0952
Cinq fonctions	0.1048	0.0890	0.0806
Sept fonctions	0.0818	0.08014	0.0775
Neuf fonctions	0.0978	0.07596	0.0759

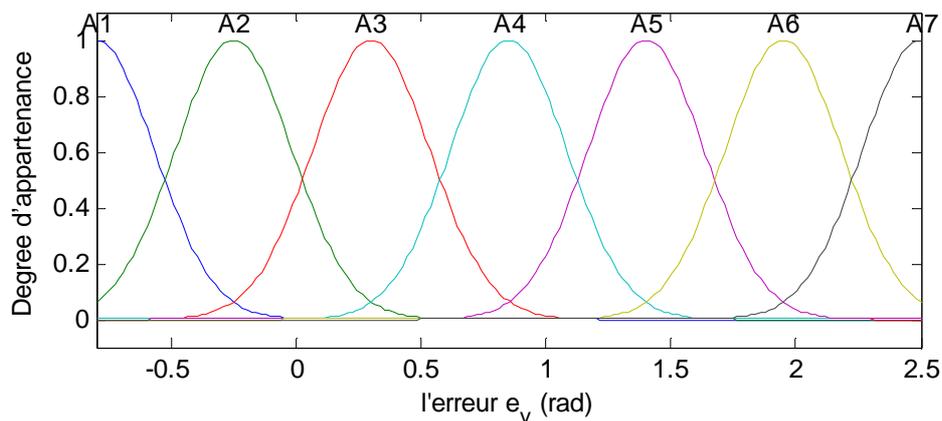
Tableau IV.1 : Erreur d'apprentissage pour neuf types de fuzzification du régulateur RNFVER

Les résultats d'apprentissage pour les neuf types de fuzzification permettent de tirer les remarques suivantes :

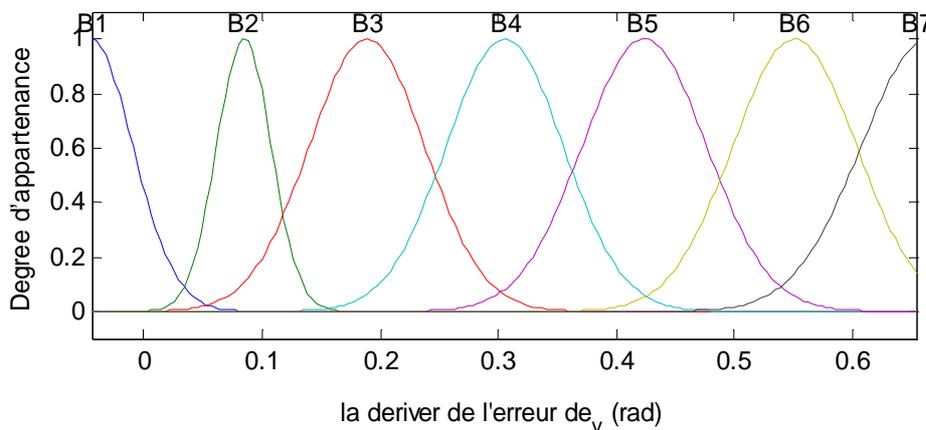
- La fonction d'appartenance gaussienne donne la plus faible erreur d'apprentissage.
- L'augmentation du nombre d'ensemble floue fait diminuer l'erreur.

Ainsi, nous avons opté pour la configuration du régulateur neuro-floue de la partie verticale (RNFVER) la structure suivante :

- Fuzzification : sept fonctions d'appartenances de type gaussien pour l'erreur ( $e_v$ ) et la dérivée de l'erreur ( $de_v$ ), voir figure (IV.6)
- 49 neufs règles floues
- Erreur de tolérance de l'ordre de 0.07
- Algorithme d'apprentissage 'Hybride'



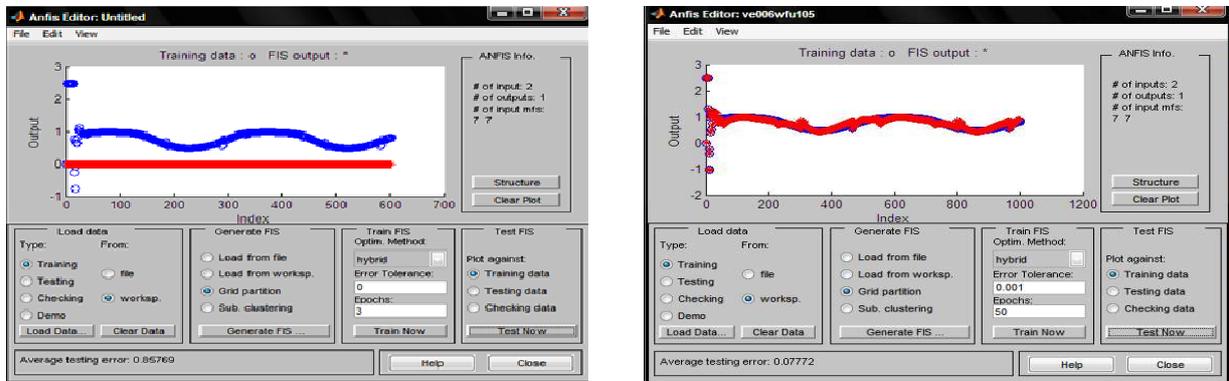
a) Fuzzification de  $e_v$



b) Fuzzification de  $de_v$

**Fig. IV.6 :** Les fonctions d'appartenances de l'erreur  $e_v$  et la dérivée de l'erreur  $de_v$

L'exécution de l'opération d'apprentissage par l'éditeur ANFIS (Fuzzy Logic Toolbox/Matlab) avec les données d'apprentissages ( $e_v, de_v$  et  $u_v$ ) obtenue d'un comportement souhaitée d'un régulateur RST ; on obtient les résultats suivantes :



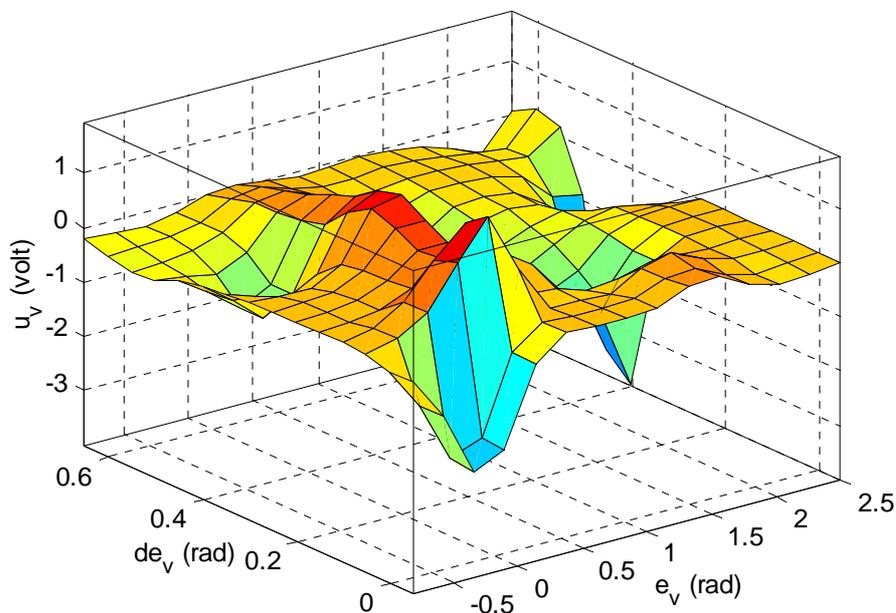
a) Avant apprentissage

b) après apprentissage

**Fig. IV.7 :** Signal de commande avant et après apprentissage du RNFVER

■ Sortie du correcteur RST    ■ Sortie du correcteur ANFIS

La figure suivante représente une surface de trois dimensions pour la variation du sortie du régulateur RNFVER ( $u_v$ ) par apport à l'erreur de l'angle d'élévation  $e_v$  et la variation d'erreur  $de_v$



**Fig. IV.8 :** La courbe en 3D de la sortie  $u_v$  du régulateur en fonction des deux entrées  $e_v$  et  $de_v$  du contrôleur RNFVER

Les gains de normalisation et de commande de la partie verticale sont donnés par le tableau suivant :

Les gains de normalisation et de commande	$Ge_v$	$Gde_v$	$Gu_v$
	0.23	0.49	0.1

*Tableau IV.2 : Tableau des gains de normalisation et de commande du modèle verticale du TRMS*

### IV.3.3 Simulation et interprétation des résultats

Le schéma de commande donné figure (IV.5) est simulé avec et sans perturbation pour deux type de consigne; un signal carré d'amplitude  $\pi/6$  sur un profil du temps de 300 seconds et un signal de référence sinusoïdale d'amplitude  $\pi/6$  et de fréquence 0.005 Hz sur un profil de temps de 300 seconds. Les résultats de simulation sont présentés dans les figures (IV.9), (IV.10), (IV.11) et (IV.12), on constate bien qu'on a abouti à des résultats suivant :

- Un temps de réponse de 20 seconds.
- Une poursuite du signal de référence avec une erreur statique nulle.
- Une commande non énergétique admissible  $-2.5V \leq u_v \leq 2.5V$
- Un rejet de perturbations

Les oscillations apparues au démarrage sont dues à l'effet de l'inertie de la poutre horizontale du TRMS sur l'angle d'élévation.

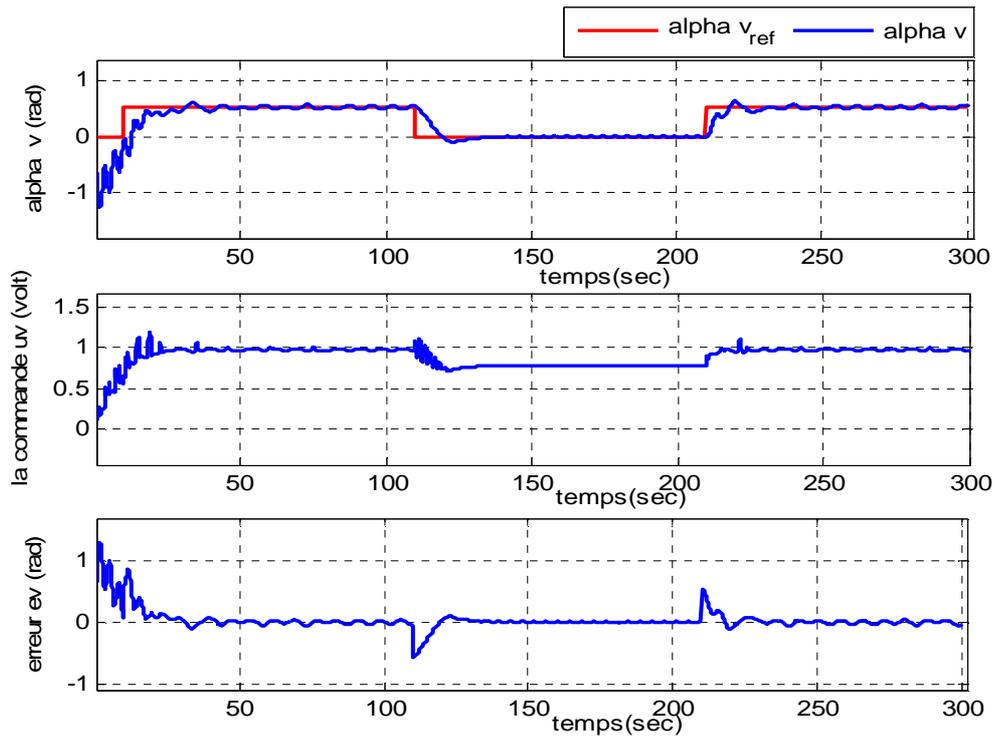


Fig. IV.9 : Résultats de simulation de la commande RNFVER pour une référence à signal carré d'amplitude  $\pi/6$

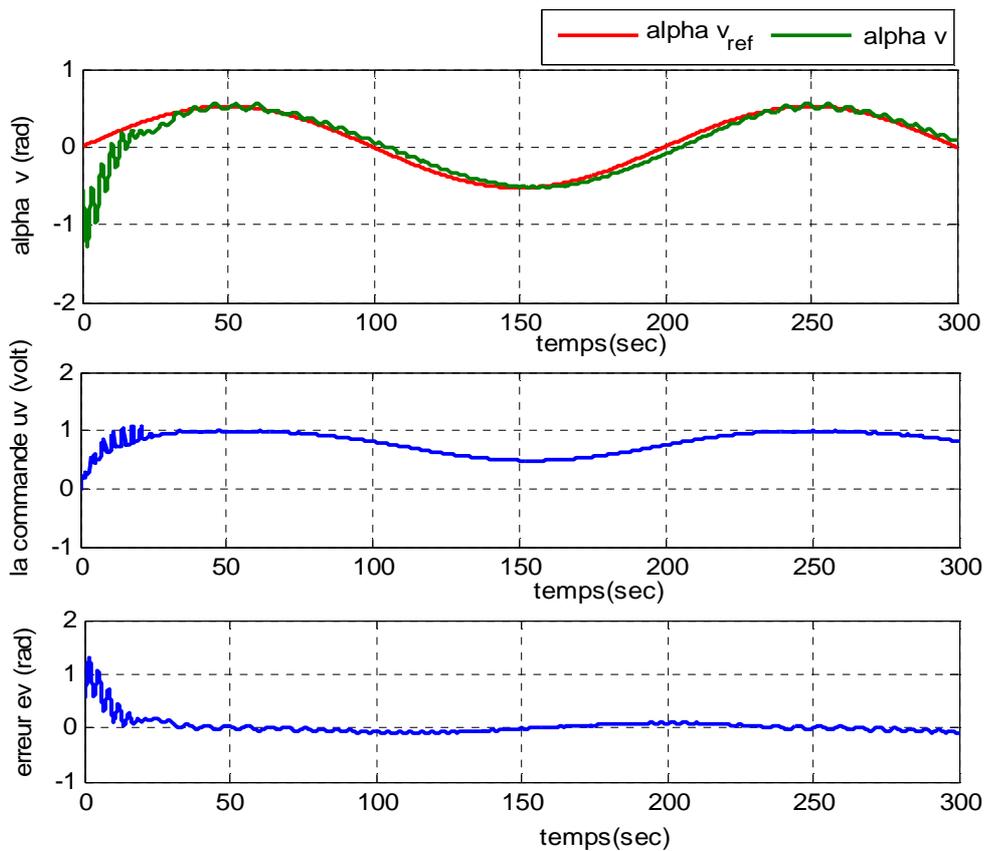


Fig. IV.10: Résultats de simulation de la commande RNFVER pour une référence sinusoïdale d'amplitude  $\pi/6$

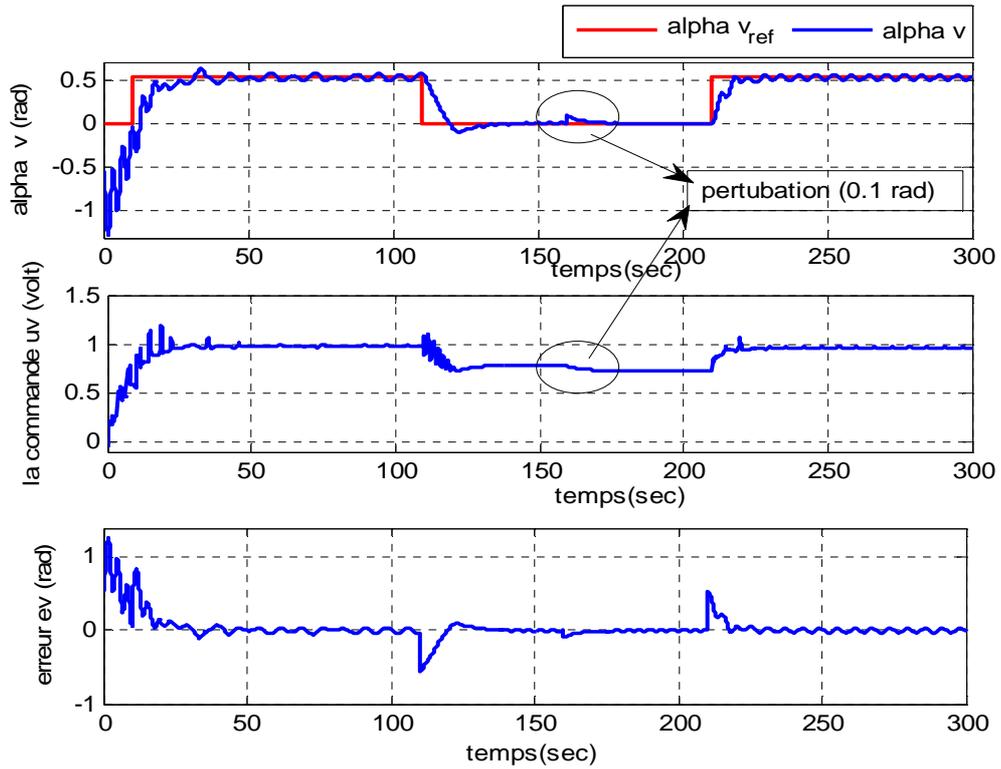


Fig. IV.11 : Résultats de simulation de la commande RNFVER pour une référence à signal carré d'amplitude  $\pi/6$  avec perturbation de (0.1rad) a l'instant ( $t=160sec$ )

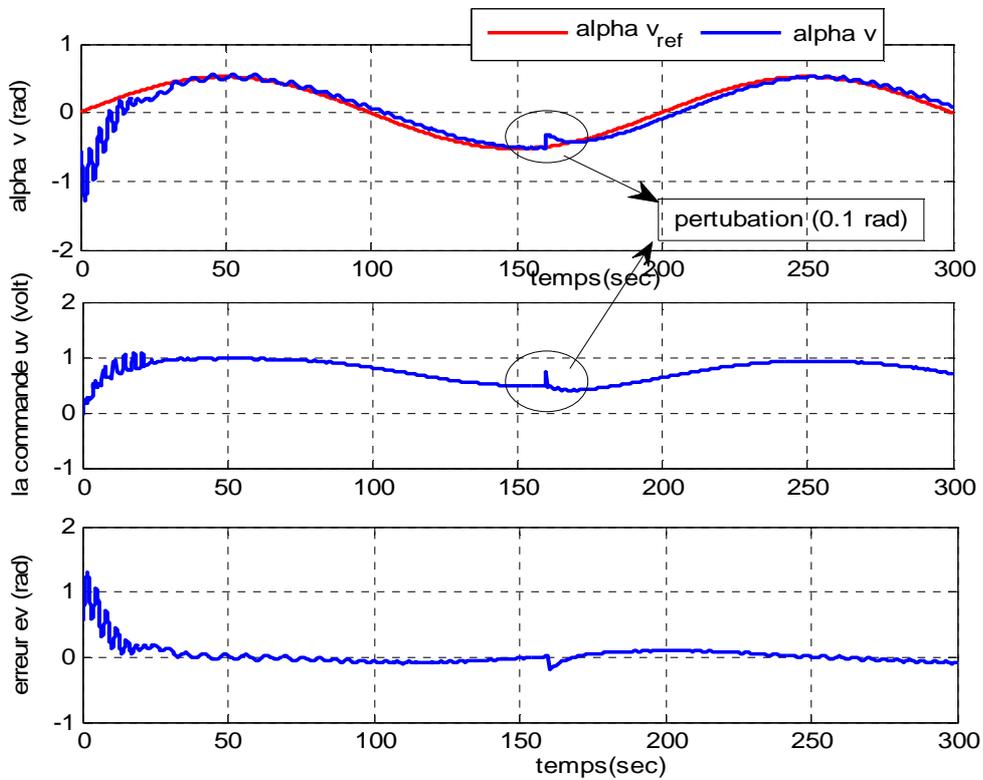


Fig. IV.12 : Résultats de simulation de la commande RNFVER pour une référence sinusoïdale d'amplitude  $\pi/6$  avec perturbation de (0.1rad) a l'instant ( $t=160sec$ )

## IV.4 Commande par Régulateur Neuro-Flous du Modèle Horizontal (RNFHOR)

### IV.4.1 Schéma de commande

De la même manière, nous présentons le schéma de commande du modèle non linéaire (équation II.34) de la partie horizontale du TRMS par la figure (IV.13), où la grandeur à commander est l'angle d'orientation  $\alpha_h$  du TRMS commandée par la tension d'alimentation du rotor secondaire  $u_h$ ,

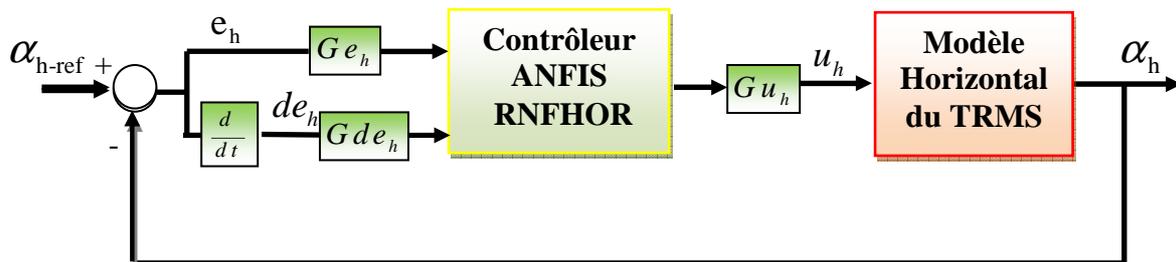


Fig. IV.13: Schéma de commande de la partie horizontale du TRMS par un régulateur neuro-floue RNFHOR

### IV.4.2 Conception du régulateur RNFHOR

La conception du régulateur RNFHOR de la partie horizontale est faite de la même façon que la partie verticale, des testes sur différents types de fuzzification ont été faites aussi avec les mêmes nombres d'ensemble floue et les mêmes fonctions d'appartenances. Ainsi, nous obtenons le tableau suivant :

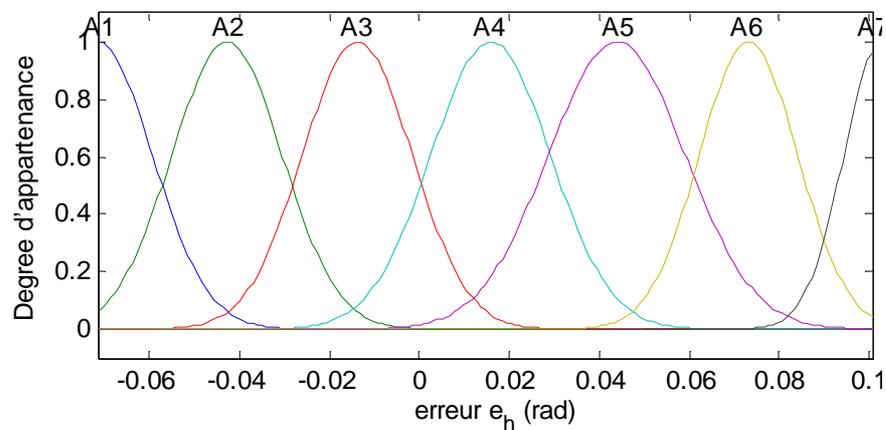
Type de fonction / Nombre De fonction	Fonction triangulaire	Fonction trapézoïdale	Fonction gaussienne
Trois fonctions	0.006367	0.007920	0.002335
Cinq fonctions	0.001888	0.001792	0.001703
Sept fonctions	0.001381	0.001471	0.001371
Neuf fonctions	0.001145	0.001185	0.001073

Tableau IV.3 : Erreur d'apprentissage pour neuf types de fuzzification du régulateur RNFHOR

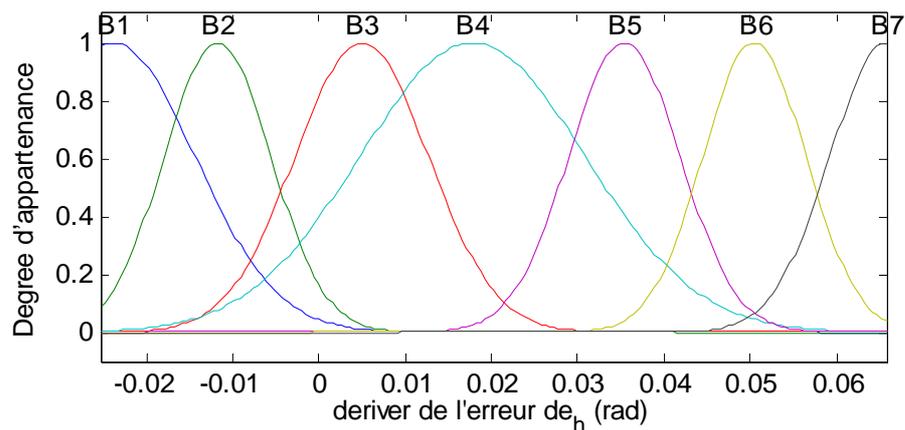
Les résultats d'apprentissage pour les neuf types de fuzzification permettent de tirer les mêmes remarques obtenues pour le régulateur RNFVER.

Ainsi, nous avons opté pour la configuration du régulateur neuro-flou de la partie horizontale (RNFHOR) la structure suivante :

- Fuzzification : sept fonctions d'appartenances de type gaussien pour l'erreur ( $e_h$ ) et la dérivée de l'erreur ( $de_h$ ), voir figure (IV.14)
- 49 neufs règles floues
- Erreur de tolérance de l'ordre de 0.001
- Algorithme d'apprentissage 'Hybride'



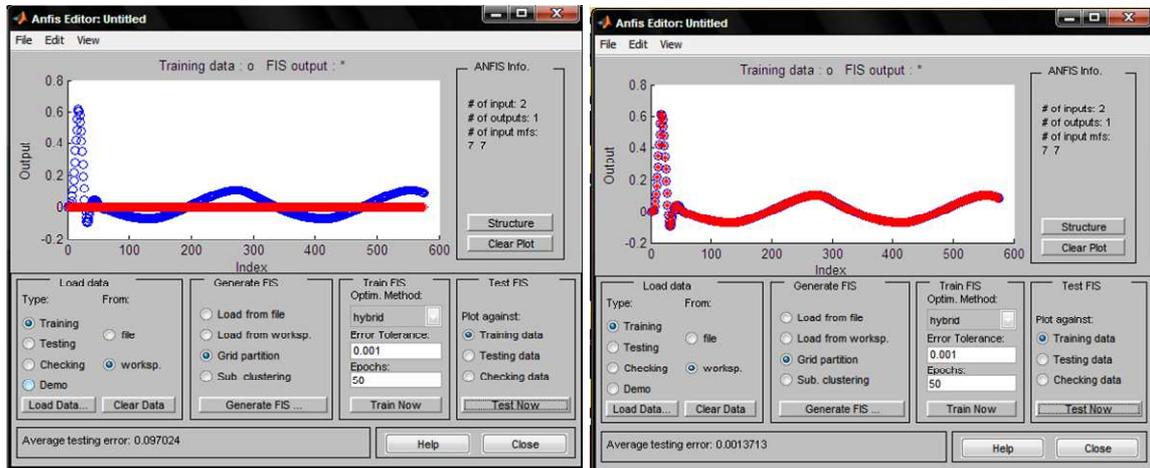
a) Fuzzification de  $e_h$



b) Fuzzification de  $de_h$

**Fig. IV.14 :** Les fonctions d'appartenances de l'erreur  $e_h$  et la dérivée de l'erreur  $de_h$

L'apprentissage par l'éditeur ANFIS (Fuzzy Logic Toolbox/Matlab) avec les données  $(e_h, de_h \text{ et } u_h)$  obtenues d'un comportement souhaité d'un régulateur RST permet d'obtenir les résultats suivants :

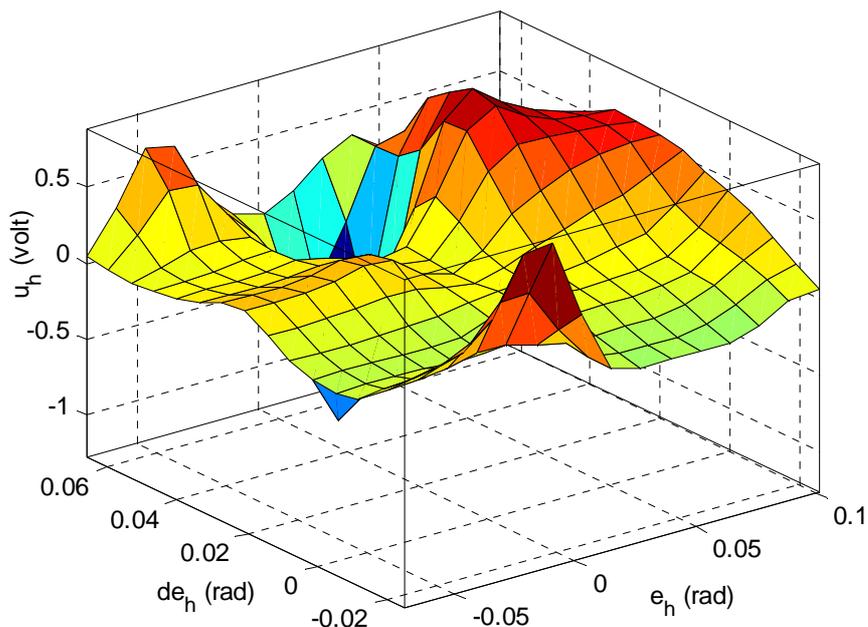


a) Avant apprentissage

b) après apprentissage

**Fig. IV.15 :** signal de commande avant et après apprentissage du RNFHOR  
■ Sortie du correcteur RST    ■ Sortie du correcteur ANFIS

La figure suivante représente une surface de trois dimensions pour la variation de la sortie du régulateur RNFHOR (tension  $u_h$ ) par apport à l'erreur de l'angle d'azimut  $e_h$  et sa vitesse de variation  $de_h$



**Fig. IV.16 :** La courbe en 3D de la sortie  $u_h$  du régulateur en fonction des deux entrées  $e_h$  et  $de_h$  du contrôleur RNFHOR

Les gains de normalisation et de commande de la partie horizontale pour le régulateur RNFHOR sont donnés par le tableau suivant :

Les gains de normalisation et de commande	$Ge_h$	$Gde_h$	$Gu_h$
	0.29	0.2	30

*Tableau IV.4 : Tableau des gains de normalisation et de commande du modèle horizontal du TRMS*

#### IV.4.3 Simulation et interprétation des résultats

Le schéma de commande du modèle horizontal donné figure (IV.13) est simulé pour les deux signaux de référence utilisés dans la commande du modèle vertical : un signal carré d'amplitude  $\pi/6$  et un signal de référence sinusoïdale d'amplitude  $\pi/6$  et de fréquence 0.005 Hz sur un profil de temps de 300 seconds. Les résultats de simulation sont présentés dans les figures (IV.17) et (IV.18), (IV.19) et (IV.20), les remarques sont :

- Un temps de réponse de 20 seconds.
- Une poursuite du signal de référence avec une erreur statique nulle.
- Une commande non énergétique admissible  $-2.5V \leq u_v \leq 2.5V$
- Un régime dynamique sans oscillations
- Un rejet de perturbations

Il est important de noter que la précision obtenue dans la partie horizontale est meilleur.

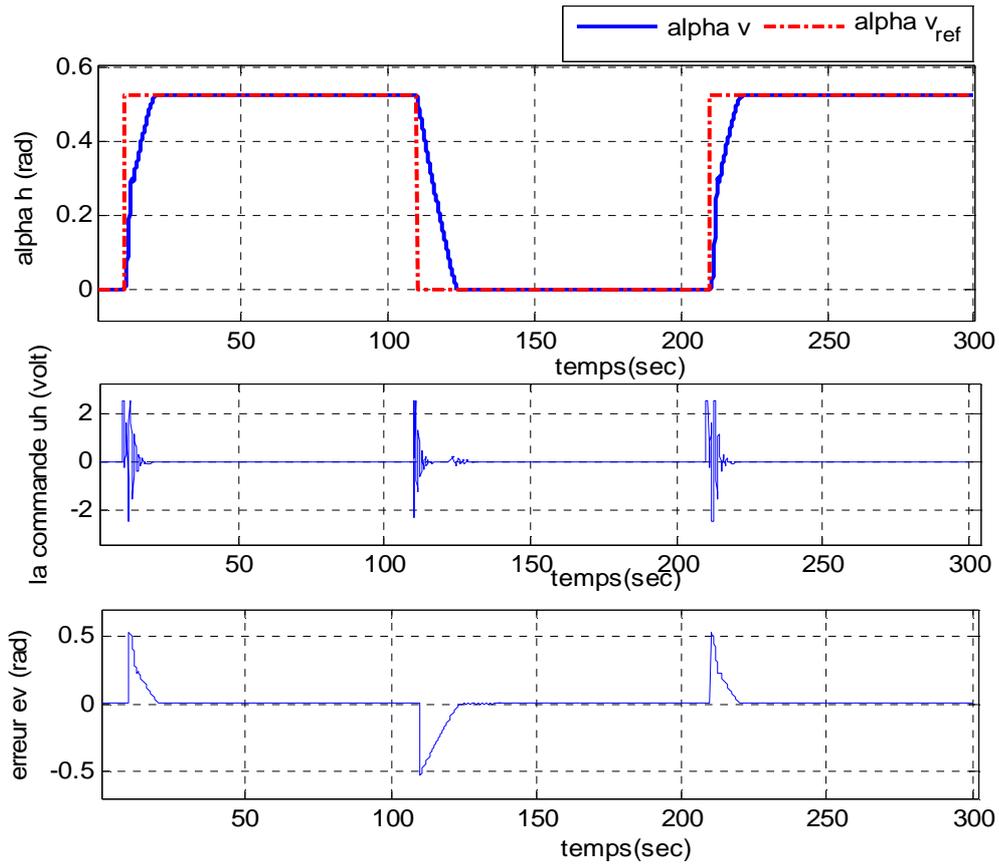


Fig. IV.17 : Résultats de simulation de la commande RNFHOR pour une référence à signal carré d'amplitude  $\pi/6$

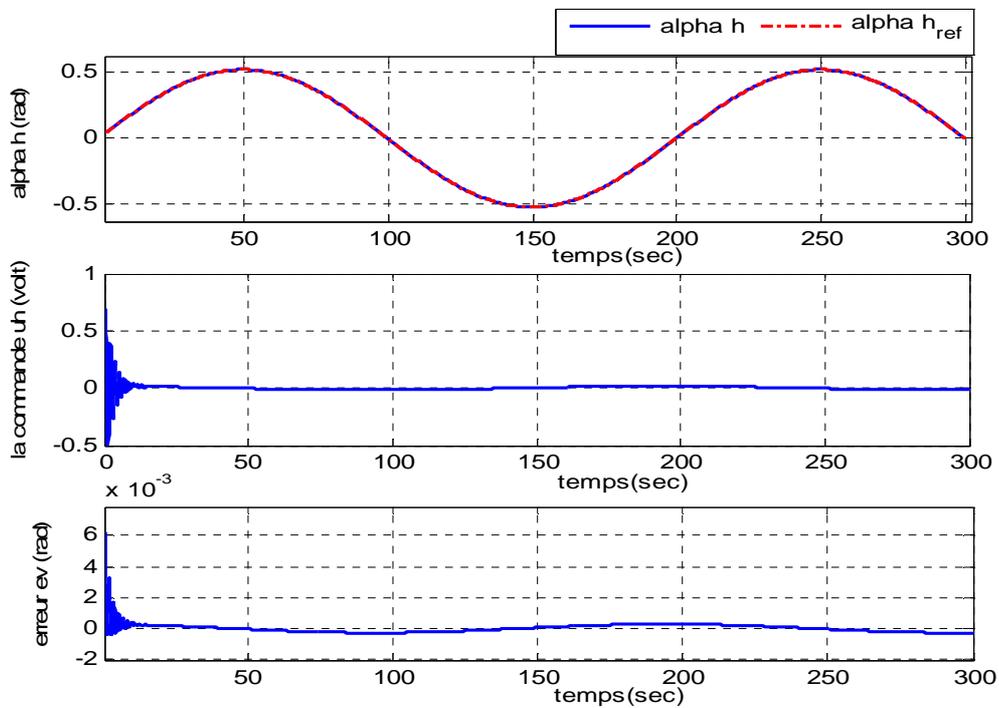


Fig. IV.18 : Résultats de simulation de la commande RNFHOR pour une référence sinusoïdale d'amplitude  $\pi/6$

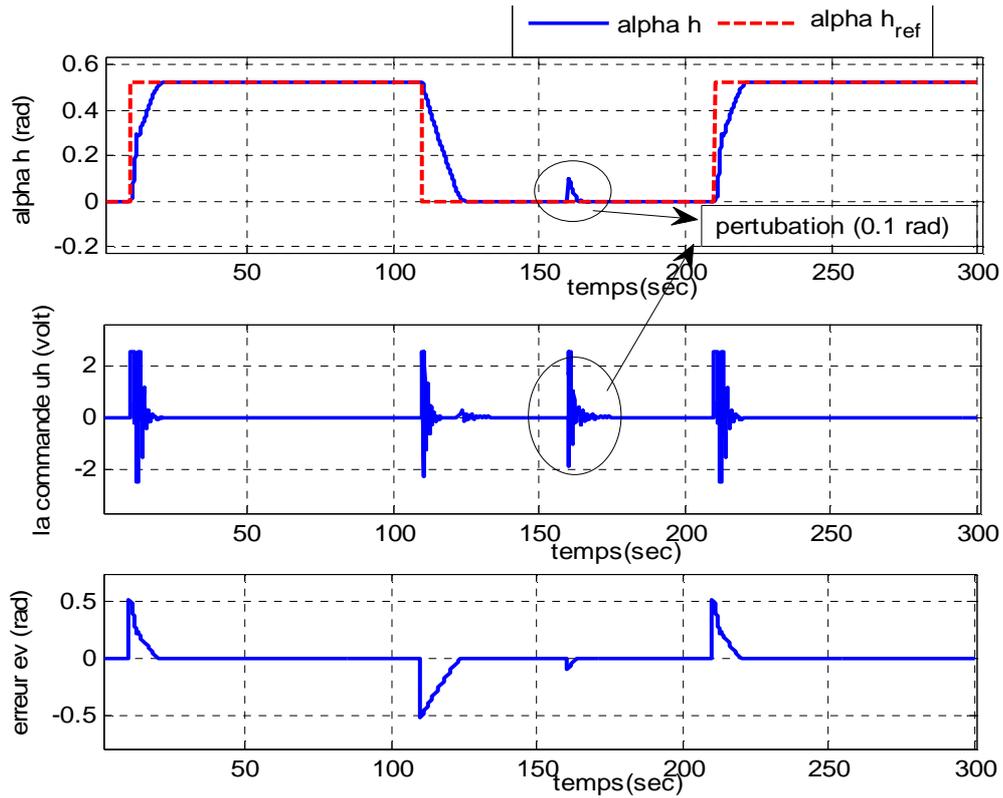


Fig. IV.19 : Résultats de simulation de la commande RNFHOR pour une référence à signal carré d'amplitude  $\pi/6$  avec perturbation de (0.1rad) a l'instant ( $t=160sec$ )

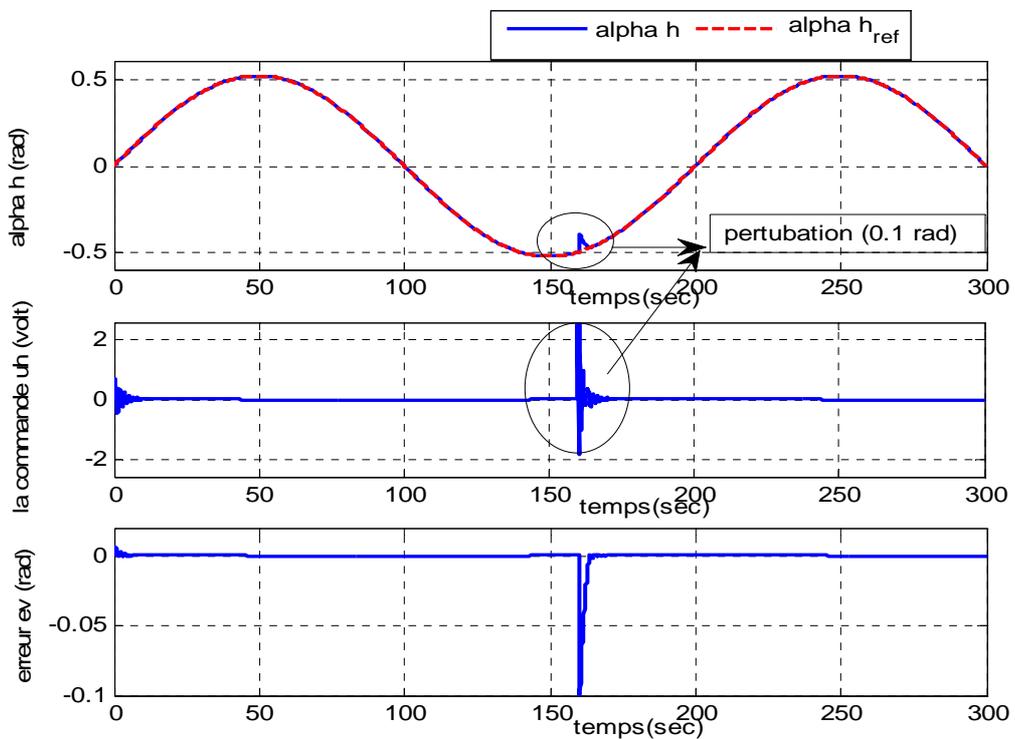


Fig. IV.20 : Résultats de simulation de la commande RNFHOR pour une référence sinusoïdale d'amplitude  $\pi/6$  avec perturbation de (0.1rad) a l'instant ( $t=160sec$ )

## IV.4 Commande Neuro-Floue décentralisée du TRMS (modèle couplé)

### IV.4 .1.Schéma de commande

L'idée d'une commande décentralisée du TRMS selon le schéma fonctionnel donné par la figure (IV.21) est inspirée de d'autres travaux, exceptionnellement celui de [32] qui développe une commande optimale pour le TRMS MIMO à partir des contrôleurs SISO optimaux indépendants, où l'effet de couplage et les variations paramétriques entre les deux sous systèmes sont considérés comme perturbations. De la même manière, les deux régulateurs Neuro-Flous RNFVER et RNFHOR sont jointés ensemble pour le contrôle du TRMS. Ainsi, l'angle d'orientation  $\alpha_h$  et d'élévation  $\alpha_v$  sont commandés par les deux tensions d'alimentation  $u_h$  et  $u_v$  des rotors secondaire et principale respectivement.

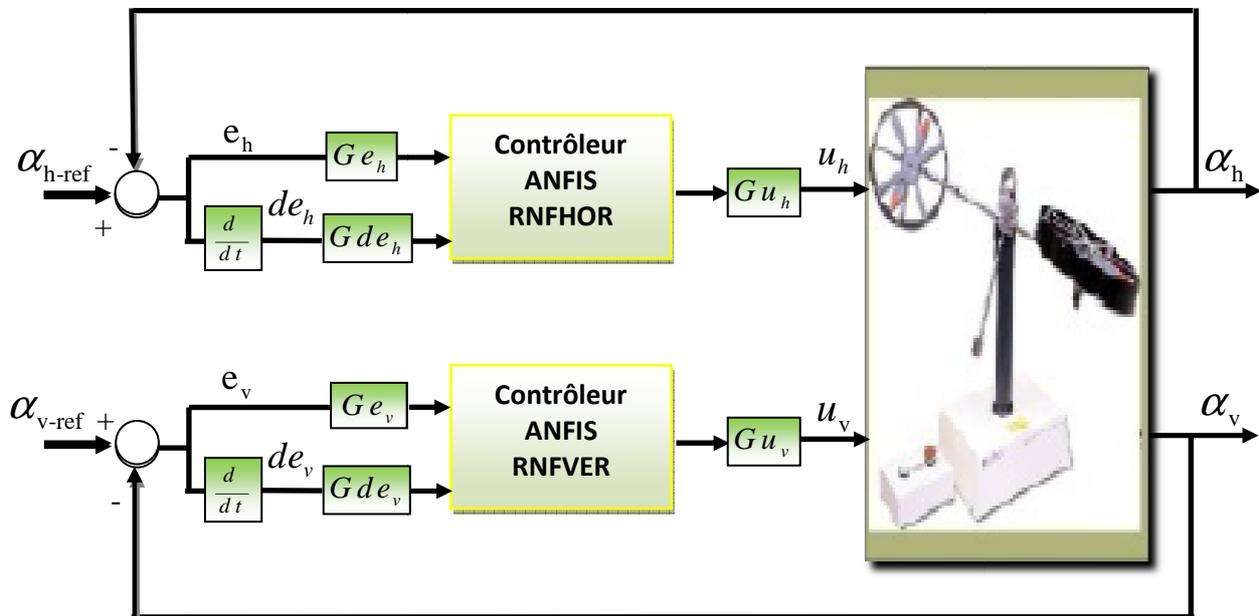


Fig. IV.21: Schéma de commande décentralisée du TRMS MIMO

Les gains de normalisation et de commande des régulateurs sont réajustés en cours de simulations, les valeurs finales sont données par le tableau suivant :

Les gains de normalisation et de commande	$Ge_h$	$Gde_h$	$Gu_h$	$Ge_v$	$Gde_v$	$Gu_v$
	0.01	1/300	40	0.08	0.3	0.3

**Tableau IV.6 :** Tableau des gains de normalisation et de commande décentralisée du TRMS MIMO

**Remarque :**

Les deux régulateurs RNFVER et RNFHOR sont utilisés avec les mêmes configurations présentées dans la commande des modèles découplés du TRMS.

#### IV.4.3 Simulation et interprétation des résultats

Le schéma de commande du TRMS MIMO donné par la figure (IV.21) est simulé de même pour les deux signaux de référence utilisés dans la commande des modèles découplés. Les figures (IV.22) et (IV.23) présentent les résultats de simulation en absence de perturbation. Tandis que, les figures (IV.24) et (IV.25) présentent les résultats de simulation avec présence de perturbation de l'ordre de 0.2 rad sur les deux angles du TRMS appliquée à l'instant 160 (sec). Ces résultats permet de tirer les constatations suivantes :

- Une augmentation pour le temps de réponse (environs 35 seconds).
- Une poursuite du signal de référence avec des oscillations.
- Les commandes sont toujours dans la région des valeurs admissibles  $-2.5V \leq u_v \leq 2.5V$
- Un régime dynamique présente des oscillations plus importantes pour l'angle d'élévation
- Un rejet de perturbations

Il important de noter que les résultats pour un signal de référence sinusoïdale sont meilleurs par rapport au signal carré. La dégradation des performances pour les deux angles est expliquée par l'effet du couplage. Toutefois, on peut dire que les deux régulateurs ont permis de commander le TRMS avec des performances plus au moins acceptable vis-à-vis la dynamique instable de ce procédé.

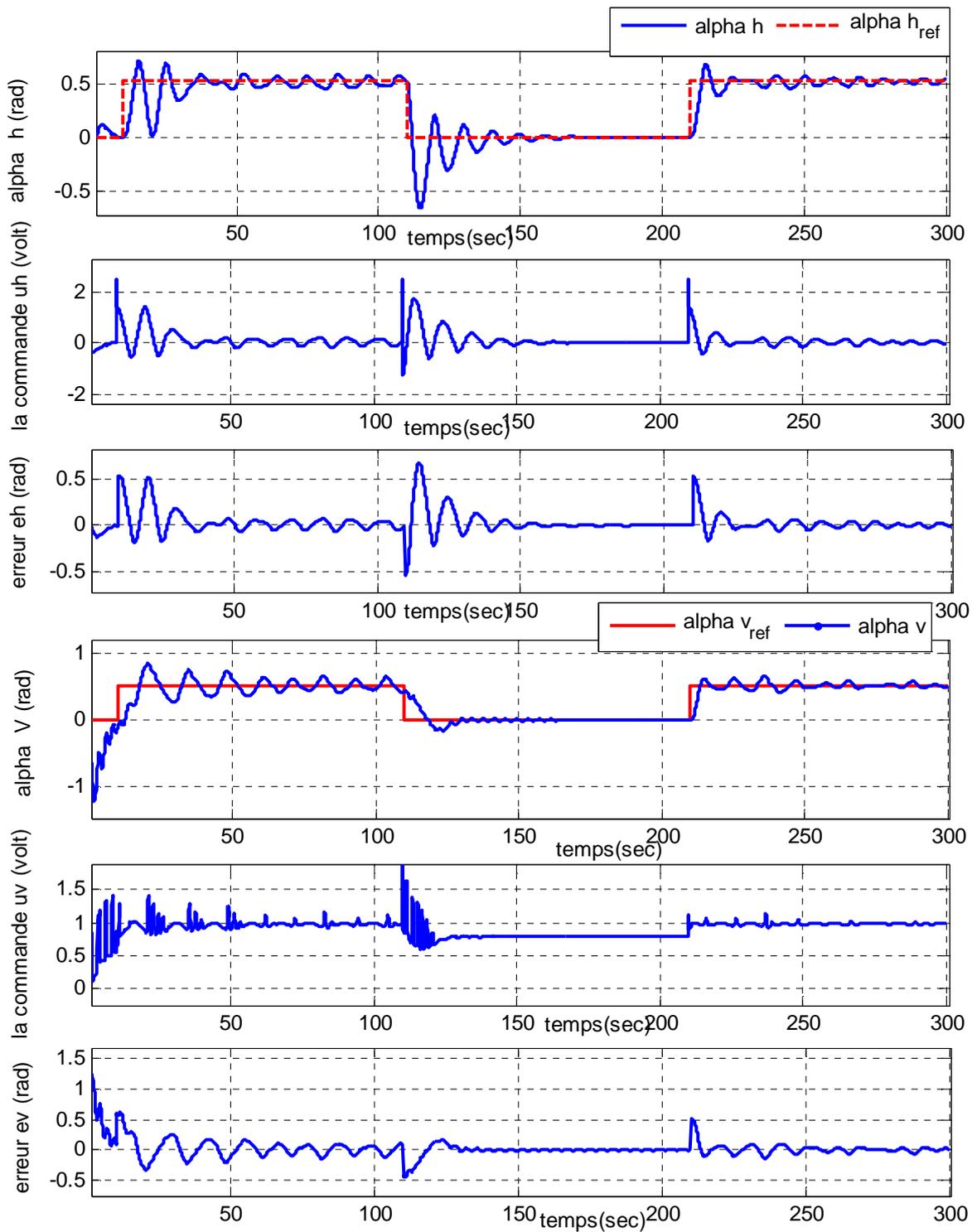


Fig. IV.22 : Résultats de simulation de la commande décentralisée du TRMS MIMO pour une référence à signal carré d'amplitude  $\pi/6$

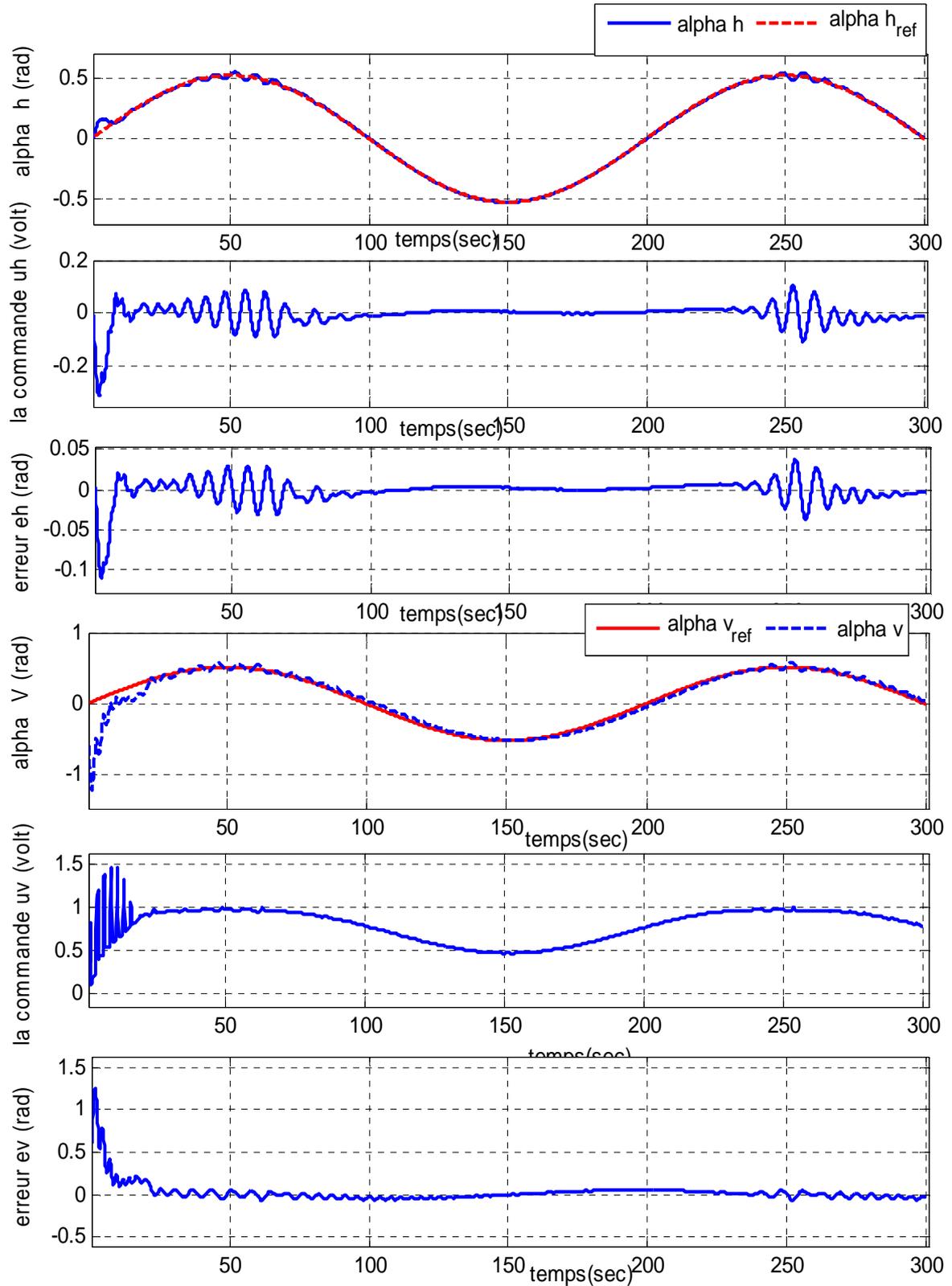
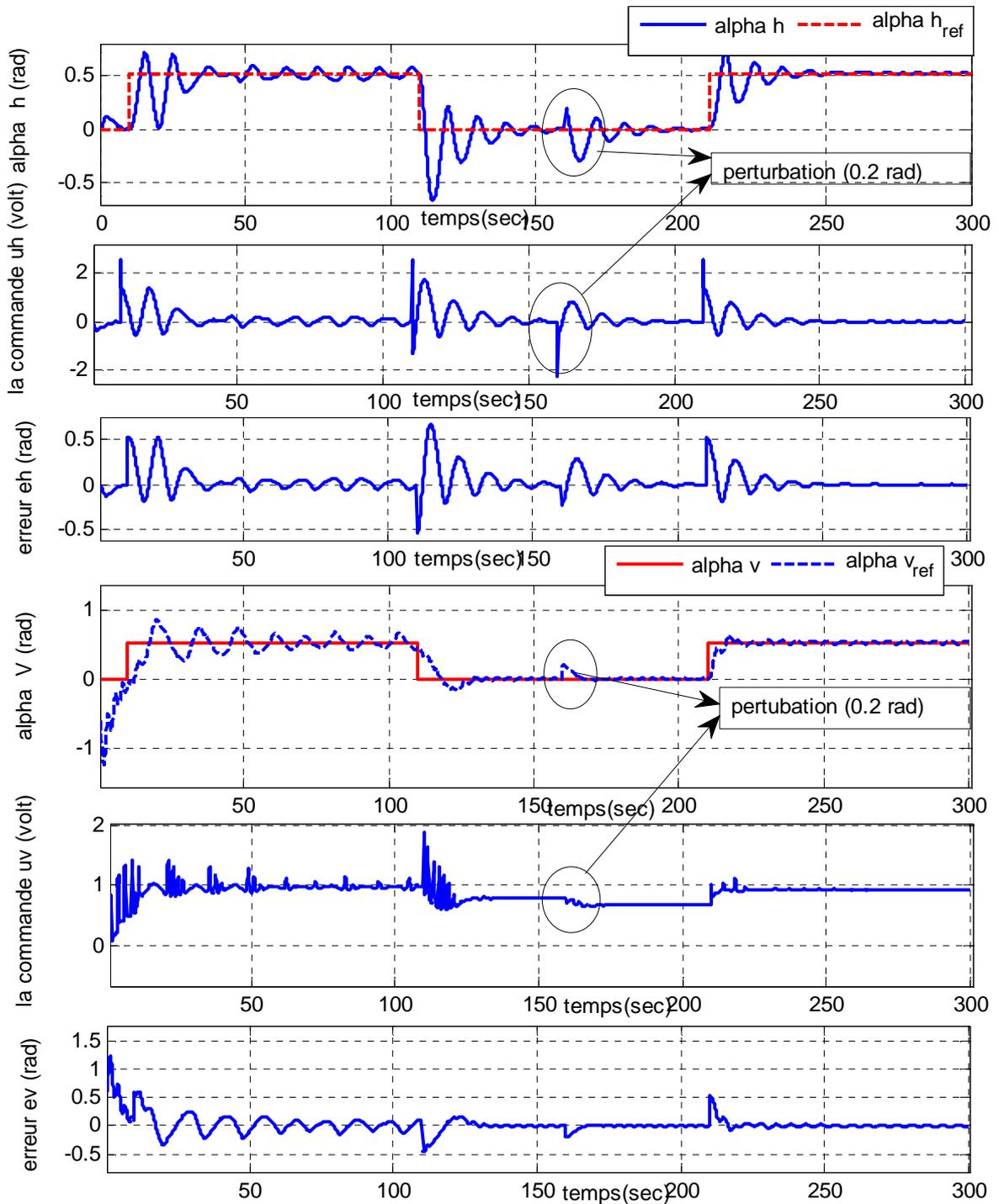


Fig. IV.23 : Résultats de simulation de la commande décentralisée du TRMS MIMO pour une référence sinusoïdale d'amplitude  $\pi/6$



**Fig. IV.24 :** Résultats de simulation de la commande décentralisée du TRMS MIMO pour une référence à signal carré d'amplitude  $\pi/6$  avec perturbation de (0.2rad) à l'instant ( $t=160sec$ )

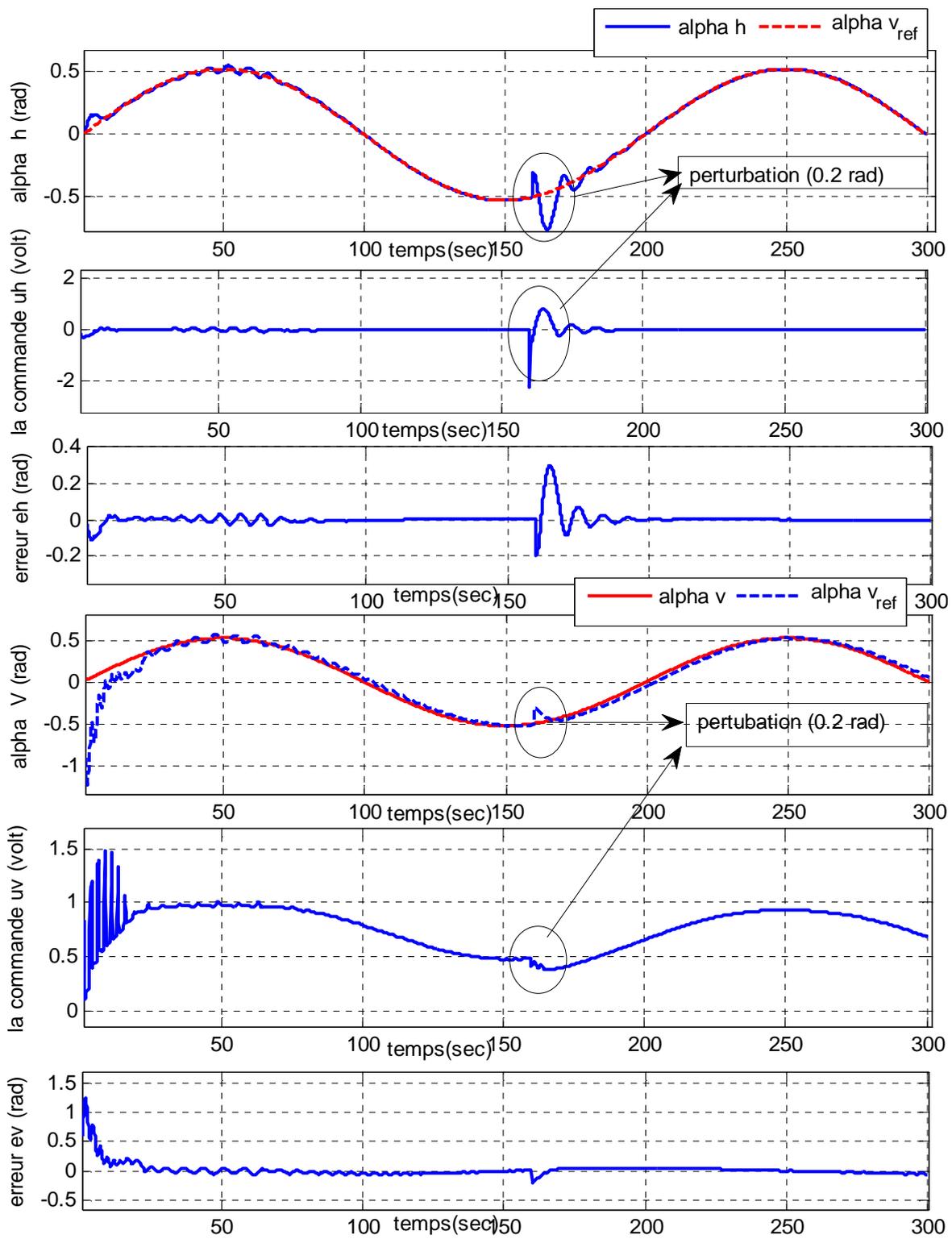


Fig. IV.25 : Résultats de simulation de la commande décentralisée du TRMS MIMO pour une référence sinusoïdale d'amplitude  $\pi/6$  avec perturbation de (0.2rad) à l' instant ( $t=160sec$ )

## IV.5 Commande Neuro-Floue 4-RNF du TRMS (modèle couplé)

### IV.5 .1 Schéma de commande

Dans cette partie du travail, nous avons proposée une structure de commande qui prend en considération le couplage existant entre les deux angles du TRMS (figure (IV.26)). Tel que, deux régulateurs neuro-flous sont ajoutés, un régulateur RNFVH conçu pour générer une commande additionnelle  $u_{vh}$  dans le but de compenser l'effet de la commande  $u_h$  sur la l'angle d'élévation  $\alpha_v$  et un régulateur RNFHV pour additionner une commande  $u_{hv}$  afin de compenser à sont tour l'effet de la commande  $u_v$  sur l'angle d'orientation  $\alpha_h$ . Ainsi, les lois de commande du TRMS MIMO couplé sont données par les équations suivantes :

$$\begin{cases} u_h = u_{hh} + u_{vh} \\ u_v = u_{vv} + u_{hv} \end{cases} \quad (IV.1)$$

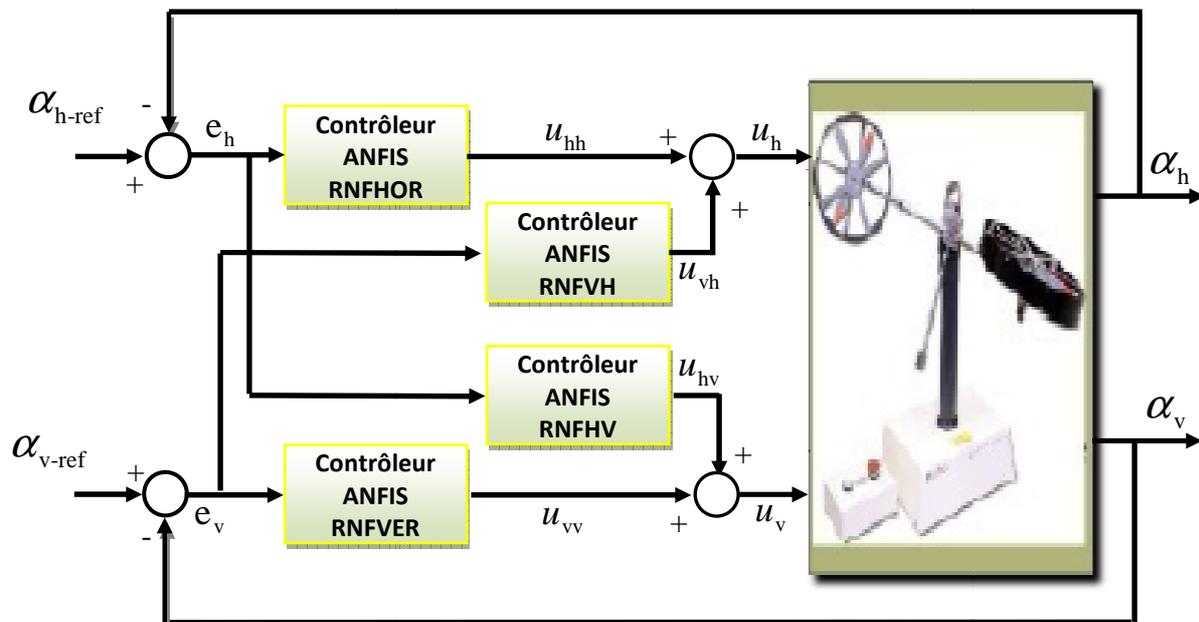


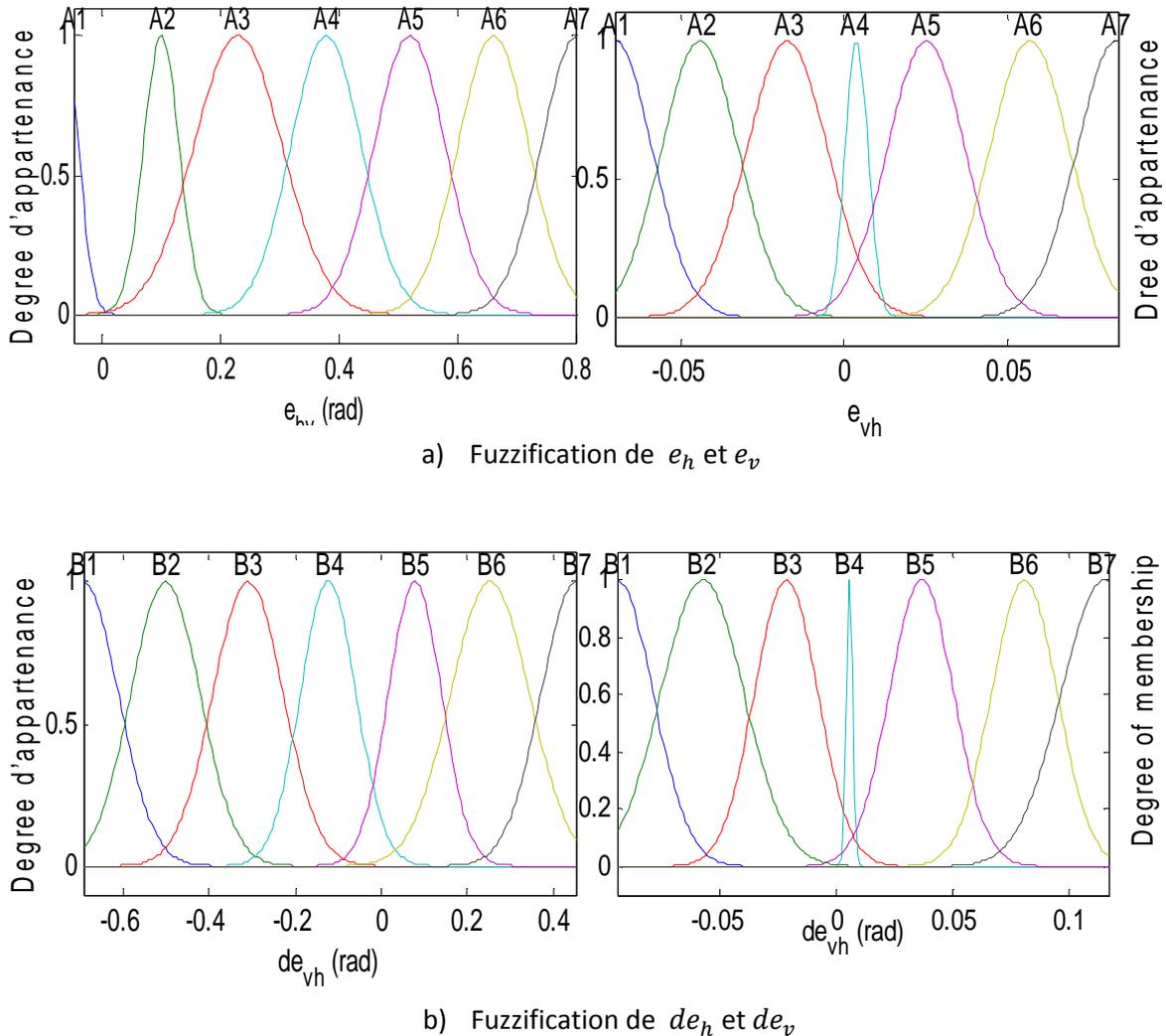
Fig. IV.26: Schéma de commande 4-RNF du TRMS MIMO

### IV.5 .2 Conception des régulateurs RNFVH et RNFHV

Les régulateurs RNFVH et RNFHV sont élaborés de la même façon, la fuzzification de l'erreur et de la dérivée de l'erreur pour les deux angles du TRMS sont données par la figure (IV.27).

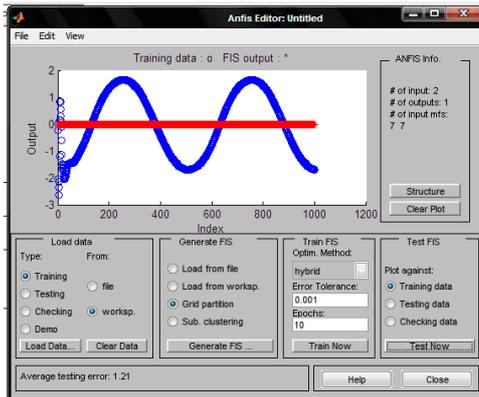
Ainsi, la configuration des régulateurs RNFVH et RNFHV ont la structure suivante :

- Fuzzification : sept fonctions d'appartenances de type gaussien pour l'erreur ( $e_h$ ) et la dérivée de l'erreur ( $de_h$ ), voir figure (IV.14)
- 49 neufs règles floues
- Erreur de tolérance de l'ordre de 0.001
- Algorithme d'apprentissage 'Hybride'

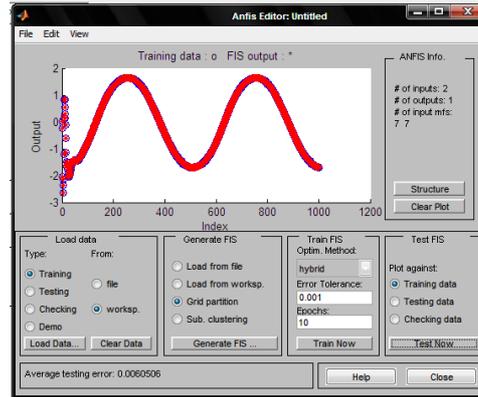


**Fig. IV.27** Les fonctions d'appartenances de l'erreur ( $e_{hv}$  et  $e_{vh}$ ) et la dérivée de l'erreur ( $de_v$  et  $de_h$ ).

L'exécution de l'opération d'apprentissage par l'éditeur ANFIS (Fuzzy Logic Toolbox/Matlab) avec les données d'apprentissages ( $e_v, de_v$  et  $u_{vh}$ ) pour le régulateur RNFVH et ( $e_h, de_h$  et  $u_{hv}$ ) pour le régulateur RNFHV, qui exprime la compensation par des régulateurs classiques de type RST, on obtient les résultats suivantes :

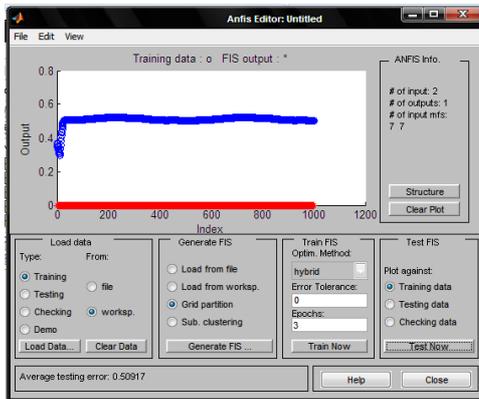


b) Avant apprentissage

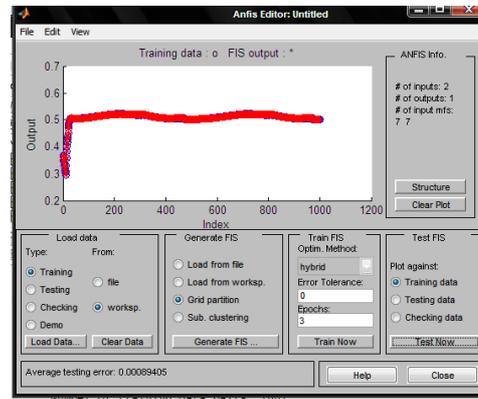


c) après apprentissage

**Fig. IV.28 :** Signal de commande avant et après apprentissage du RNFVH  
■ Sortie du correcteur RST ■ Sortie du correcteur ANFIS



c) Avant apprentissage



d) après apprentissage

**Fig. IV.29 :** Signal de commande avant et après apprentissage du RNFHV  
■ Sortie du correcteur RST ■ Sortie du correcteur ANFIS

Les gains de normalisation et de commande des régulateurs sont réajustés en cour de simulation, les valeurs finales sont données par le tableau suivant :

Les gains de normalisation et de commande	Contrôleur RNFVH			Contrôleur RNFHV		
	$Ge_{hv}$	$Gde_{hv}$	$Gu_{hv}$	$Ge_{vh}$	$Gde_{vh}$	$Gu_{vh}$
	0.05	0.3	0.49	0.019	0.059	0.5

**Tableau IV.7 :** Tableau des gains de normalisation et de commande des régulateurs RNFVH et RNFHV du TRMS MIMO

**Remarque :**

Les deux régulateurs RNFVER et RNFHOR sont utilisés avec les mêmes configurations présentées dans la commande des modèles découplés du TRMS ; Les gains de normalisation pour deux autres régulateurs sont toujours les mêmes, donnés par le tableau IV.6.

**IV.5.3 Simulation et interprétation des résultats**

Le schéma de commande du TRMS MIMO donné par la figure (IV.26) est simulé de même pour les deux signaux de référence. Les résultats de simulation sont donnés par les figure (IV.30) et (IV.31), (IV.32) et (IV.33) présentent les résultats de simulation sans et avec perturbations. Ces résultats permet de tirer les constatations suivantes :

- Un temps de réponse (environs 35 seconds) sur les deux angles.
- Une poursuite du signal de référence avec des oscillations.
- Les commandes sont toujours dans la région des valeurs admissibles  $-2.5V \leq u_v \leq 2.5V$
- Un régime dynamique avec des oscillations.
- Un rejet de perturbations.

A partir de ces remarques on constate que les deux régulateurs de compensation RNFVH et RNFHV n'ont pas apportés les améliorations souhaitées par rapport à la structure de commande décentralisée.

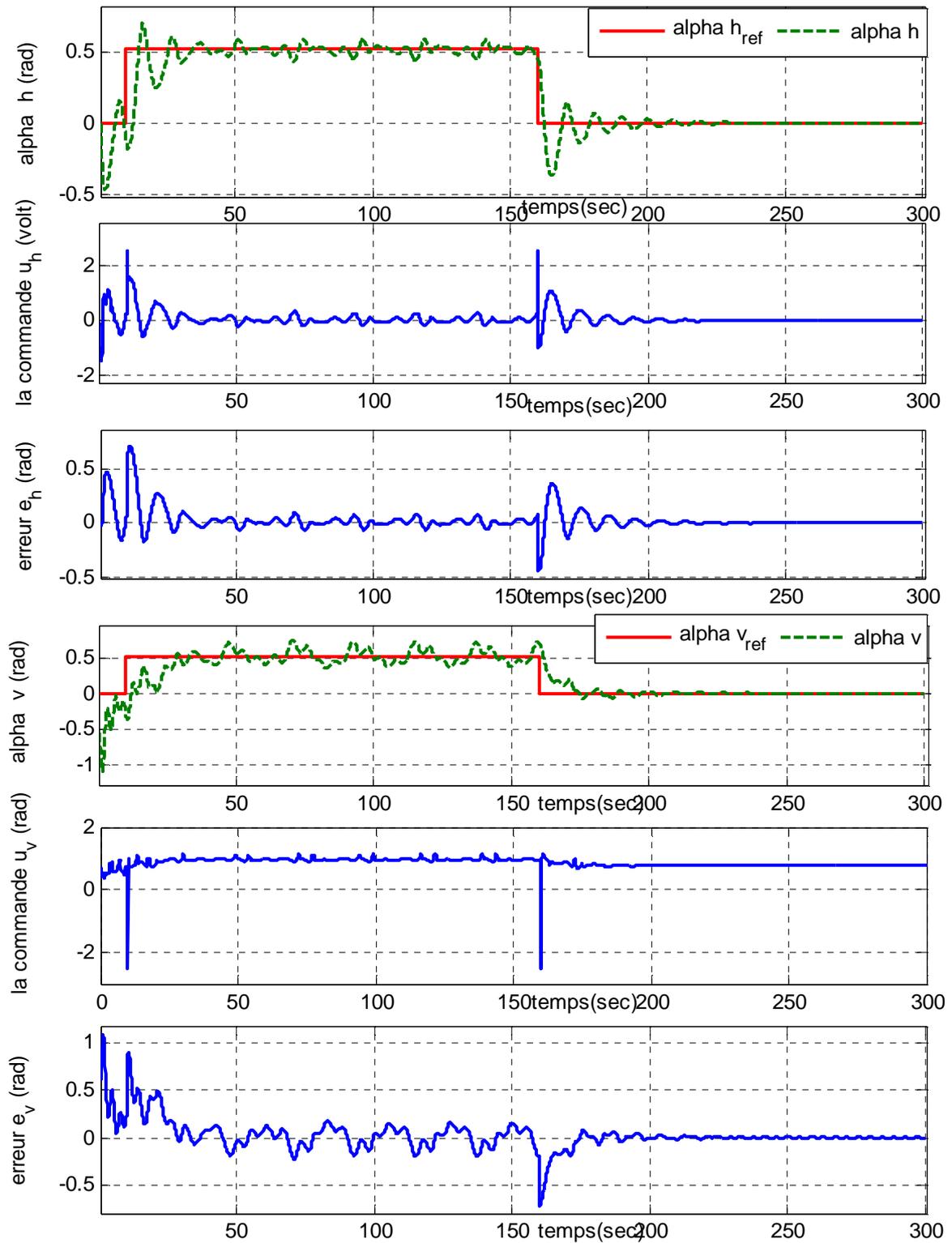


Fig. IV.30 : Résultats de simulation de la commande centralisée du TRMS MIMO pour une référence à signal carré d'amplitude  $\pi/6$

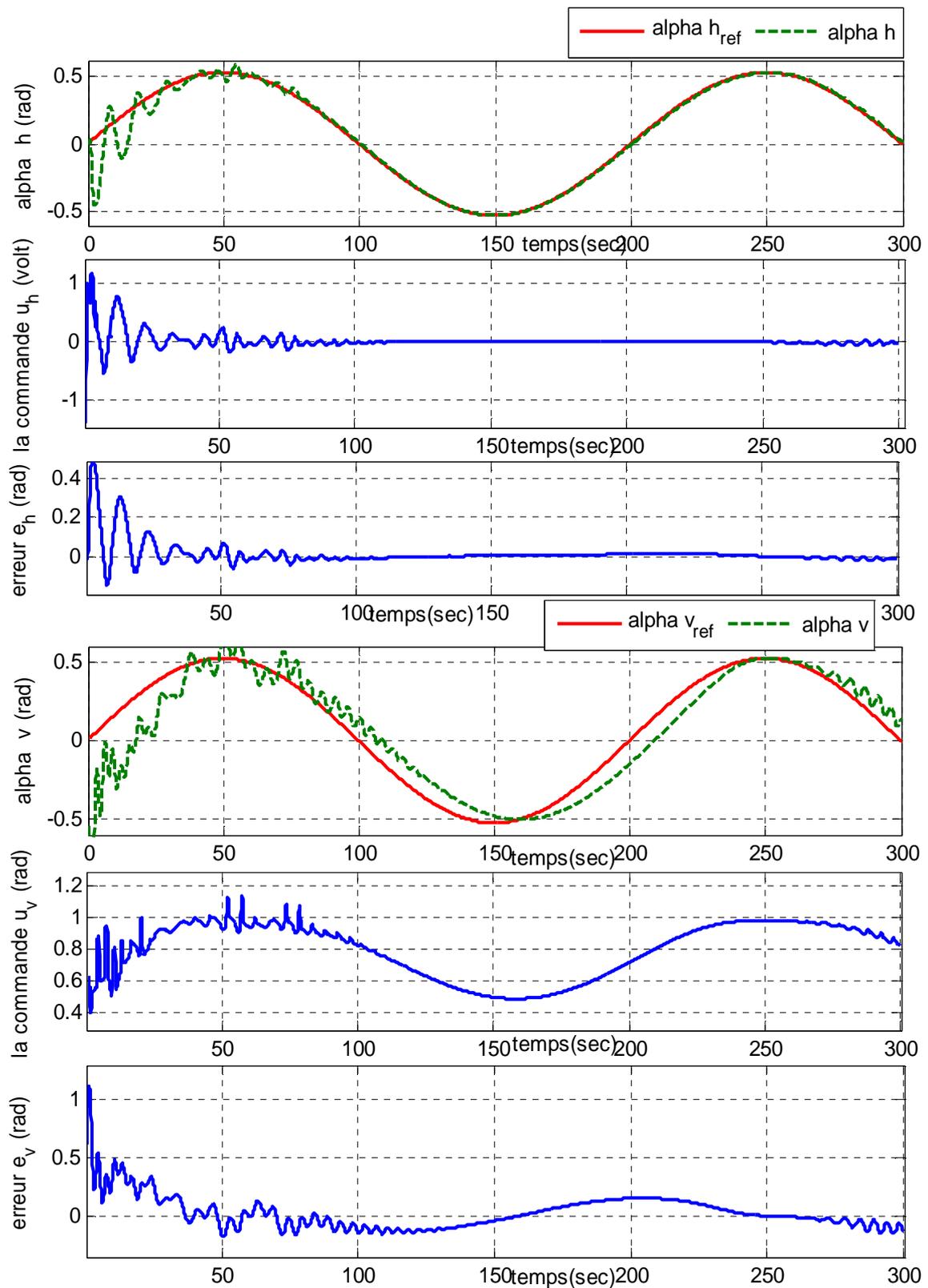
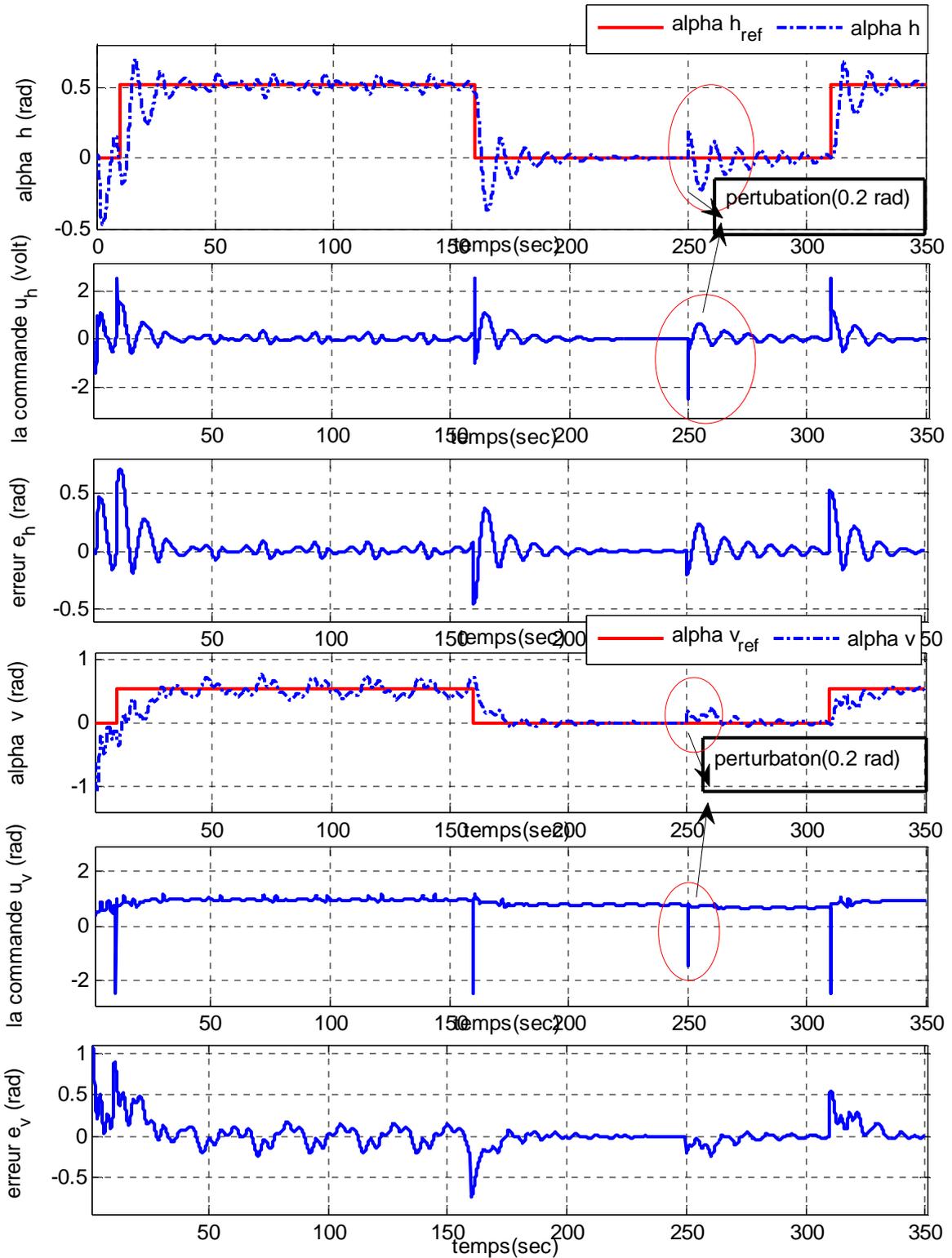
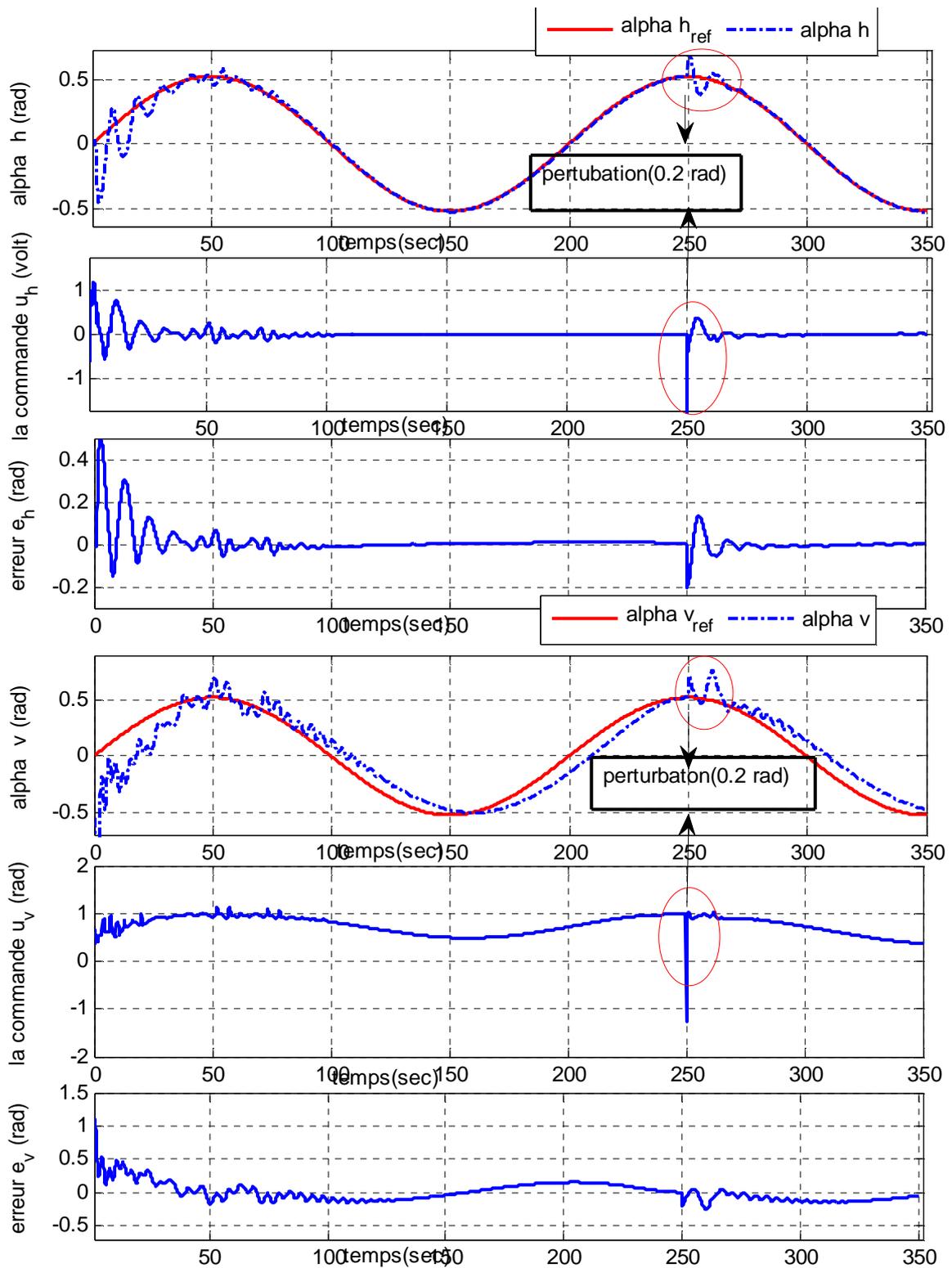


Fig. IV.31 : Résultats de simulation de la commande centralisée du TRMS MIMO pour une référence sinusoïdale d'amplitude  $\pi/6$



**Fig. IV.32 :** Résultats de simulation de la commande centralisée du TRMS MIMO pour une référence à signal carré d'amplitude  $\pi/6$  avec perturbation de (0.2rad) à l'instant ( $t=250\text{sec}$ )



**Fig. IV.33 :** Résultats de simulation de la commande centralisée du TRMS MIMO pour une référence sinusoïdale d'amplitude  $\pi/6$  avec perturbation de (0.2rad) à l'instant ( $t=250$  sec)

## IV.6 Conclusion

Cette partie du mémoire est consacrée à l'application des techniques de commande neuro-floues présentées dans le chapitre précédent au modèle couplé et découplé du simulateur d'hélicoptère (TRMS) présenté au chapitre II.

Pour le modèle découplé, les résultats obtenus montrent que la stabilisation est assurée pour l'élévation comme pour l'azimut mais avec des dépassements qui diffèrent selon le mode de fonctionnement. On observe que la qualité des réponses du système dans le mode horizontal découplé est meilleure de point de vue temps de réponse et oscillations par rapport au système en mode vertical découplé.

En mode couplé; on constate aussi que la structure de commande décentralisée proposée est robuste vis-à-vis les interactions entre les deux rotors principale et secondaire par ce que ils ont compensés le couplage. Donc, un contrôle d'un système multivariable non linéaire sans courir à des outils et des techniques très complexes.

## **CONCLUSION GENERALE**

L'objectif général de ce mémoire était l'application des techniques de l'intelligence artificielle pour la commande d'un simulateur de vol d'hélicoptère (TRMS).

Nous avons commencé par aborder en détail la description et la modélisation du simulateur afin de comprendre la complexité dynamique du système, Nous avons vu qu'il constitué de deux rotors principal et secondaire chargés a l'élévation et l'orientation du TRMS. Donc, on abouti à un modèle non linéaire multivariable fortement couplé, découplable en sous système non linéaire SISO.

Après avoir présenté la théorie de la logique floue et la théorie des réseaux de neurones, la théorie de commande Neuro-floue à fait l'objet d'une étude détaillée en particulier l'approche ANFIS afin de pouvoir l'appliquée sur les modèles du TRMS.

L'application de la commande neuro-floue aux modèles découplés à prouver sa robustesse vis-à-vis la non linéarité. L'efficacité de ce type de commande et conditionnée d'un part par une conception optimale (nombre d'ensemble flou et type de fonction d'appartenance) et d'autre part par un bon apprentissage qui dépend essentiellement de l'ensemble des données d'apprentissage du régulateur, elle doit couvrir toute la plage de son fonctionnement.

L'idée d'une commande décentralisée du TRMS MIMO à donner des résultats intéressants, chose qui permet de réclame la robustesse des régulateurs neuro-flou vis-à-vis le couplage qui existe entre les sous systèmes sans courir à savoir la nature et la complexité de ces interactions. Donc, une commande d'un système multivariable non linéaire par des régulateur neuro-flou SISO.

Les résultats obtenus lors de notre travail encouragent la poursuite des recherches dans ce sens nous pouvons également envisager les point suivants :

- Modélisation du TRMS par l'approche hybride (neuro-floue).
- la commande décentralisée multidimensionnelle (plus que 2 entrées et deux 2 sorties) par des régulateurs SISO dont la supervision est faite par un algorithme génétique

## BIBLIOGRAPHIE

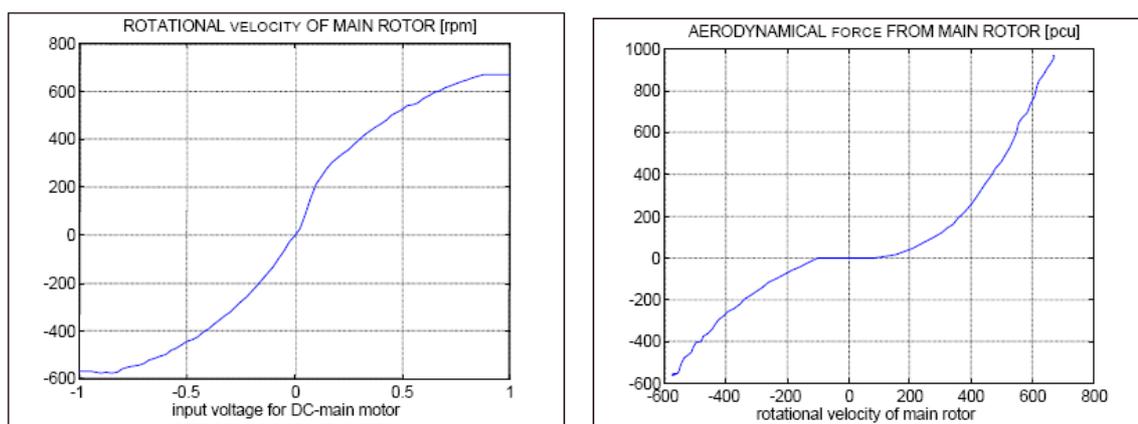
- [01] **F. Farah, F. Hachid**  
“Synthèse et implémentation pratique de lois de commandes non linéaires : Application d’un simulateur d’hélicoptère” *projet de fin d’études, Ecole Nationale polytechnique d’Alger* juin 2005.
- [02] **Manual of the Twin Rotor MIMO System**  
“Installation and Commissioning”  
*manual 33-948\_949-0V61 Ed 01 06 2002.*
- [03] **Manual of the Twin Rotor MIMO System**  
“Control Experiment”,  
*manual 33-948\_949-1V61 Ed 01 06 2002.*
- [04] **Manual of the Twin Rotor MIMO System**  
“Environment introduction and software installation” ,  
*manual 33-000V61Ed 04 02 2002.*
- [05] **Manual of the Twin Rotor MIMO System**  
“advanced teaching manual”,  
*manual 330074M5 ed 2001.*
- [06] **Manual**, Getting Started with Real-Time Workshop for Use with SIMULINK,  
American: *Math Work Inc., 2002-2005.*
- [07] **Mellah R.**  
“contribution la commande neuro-floue. Application à la robotique ”  
*thèse de Doctora, université des sciences et de la technologie Houari Boumedienne. Alger Mai 2006*
- [08] **FAUCHER. J.**  
“les plans d’expériences pour les réglage et commande à base logique floue ”  
*thèse de Doctora, institut national Polytechnique toulouse. Spécialité : génie Electrique et automatique*
- [09] **H. Bühler**  
“ Réglage par logique floue ”  
*Presses Polytechniques et Universitaires Romandes, 1994.*
- [10] **P. Borne, J. Rozinoer, J. Y. Dieulot**  
“Introduction à la logique floue ” ; *Edition technip, 1998.*
- [11] **Gacogne, L.**  
“ Eléments de logique flous ” *paris : Hermés 1997.*

- [12] **J. Yan, M. Ryan & J. Power,**  
“Using fuzzy logic”, *Prentice Hall International (UK), 1994.*
- [13] **E. Mamdani,**  
“An experiment in linguistic synthesis with a fuzzy logic controllers”,  
*Inter. Jour. on Man-Machine Studies, Vol. 7, pp. 1-13, 1975.*
- [14] **Kickert. W. J.M and Mamdani, E.H,**  
“Analysis of a fuzzy logic controller”, *Fuzzy Sets and Systems*, 1, 29–44, 1978.
- [15] **Backley, J. J.**  
“Theory of the fuzzy controller: an introduction”, *Fuzzy Sets and Systems*, 51, 249–258, 1992.
- [16] **Nezar. M.**  
“Diagnostic des associations convertisseurs statiques machines asynchrones en utilisant les techniques de l'intelligence artificielle”, *thèse de doctorat d'état de l'université de Batna, 2006.*
- [17] **L. Baghli,**  
“Contribution à la Commande de la Machine Asynchrone, Utilisation de la Logique Floue, des Réseaux de Neurones et des Algorithmes Génétiques”,  
*Thèse de Doctorat, Université Henri Poincaré, France, 1999.*
- [18] **Mizumoto, M,**  
“Fuzzy controls under various fuzzy reasoning methods”, *Information sciences*, 45, 129–151, 1988.
- [19] **Hellendoorn. H,**  
“Closure properties of the compositional rule of inference”, *Fuzzy Sets and Systems*, 35, 163–183, 1990.
- [20] **E. Davalo et P. Naim,**  
“Des Réseaux de Neurones”, Editions Eyrolles, deuxième édition, 1993.
- [21] **F. Moutarde,**  
“Introduction aux Réseaux de Neurones”, Centre de Robotique CAOR), Ecole des Mines de Paris, Mai 2008.
- [22] **Bone P.pozinor J.Dieulot J.Dubois L.**  
“introduction a la commande floue” ditions Techniq,1998 paris
- [23] **O. Abdeslam.,**  
“Techniques neuromimétiques pour la commande dans les systèmes électriques: application au filtrage actif parallèle dans les réseaux électriques basse tension”,  
*thèse de doctorat,*
- [24] **Breton, S.**  
“Une approche neuronale du contrôle robotique utilisant la vision Binoculaire par reconstruction tridimensionnelle“ *These de Doctort, Université de haute alsace U.F.R. des Scienc et Techniques Juin 1999.*

- [25] **Fabien, M.**  
« Introduction aux réseaux de neurones et leurs application en Automatique », *Ecole Mines de Paris, Décembre 2003*
- [26] **Nauck, D, et R. Kruse,**  
“What are Neuro-Fuzzy Classifiers? ”, *Seventh International Fuzzy Systems Association World Congress IFSA '97*, Vol. IV, pp. 228-233, Academie de Prague, 1997.
- [27] **Racoceanu. D,**  
“Contribution à la surveillance des Systèmes de Production en utilisant les Techniques de l’Intelligence Artificielle”, *Habilitation à dirigé des recherches, université de Franche-Comté de Besançon, 2006.*
- [28] **M. Al-Harthi**  
"design of a fuzzy- based pid contrpller for a dc servomotor position control "  
*First international conference on electrical systems PCSE'05 May 9-11 2005*  
*O.E.bouaghi Univ.Algeria.*
- [29] **D.Taibi, A.Goléa,N.Golea**  
"commande vectorielle directe sans capteur d'un moteur induction associé a un observateur d'état par filtre de kalman etendue".*First international conference on electrical systems PCSE'05 May 9-11 2005,O.E.bouaghiUniv.Algeri.*
- [30] **Yi He, Desmond M. A. Brdossy, Gerhard D., T. Ringger, S. Aicher** "debruitage par “logique floue” des signaux d’ultrasons originantd’ *une approche d'essai non destructive*".*Otto-Graf-Journal Vol. 15, 2004*
- [31] **A. Maddi, A. Guessoum, D. Berkani, O. Belkina**  
«Etude de la méthode des moindres carrée récursive et application au signal de parole» *3rd International Conference: Sciences ofElectronic, Technologies of Information and Telecommunications March 17-21, 2005 – TUNISIA*
- [32] **TE-W EI LU, 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*

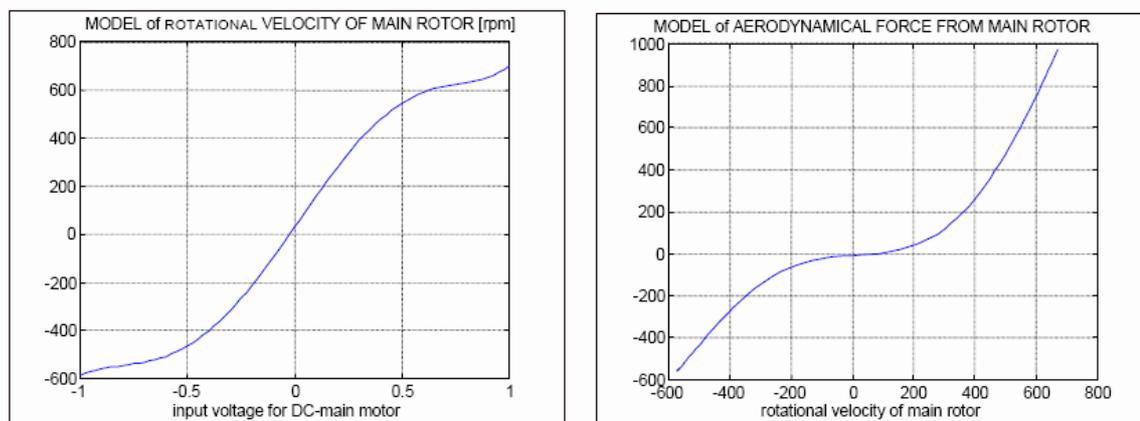
ANNEXE  
A**Courbe expérimentales des caractéristiques non linéaires [5]**

On obtient expérimentalement les deux courbes de la vitesse de rotation des rotors en fonction de la tension d'alimentation des moteurs (principal et secondaire).



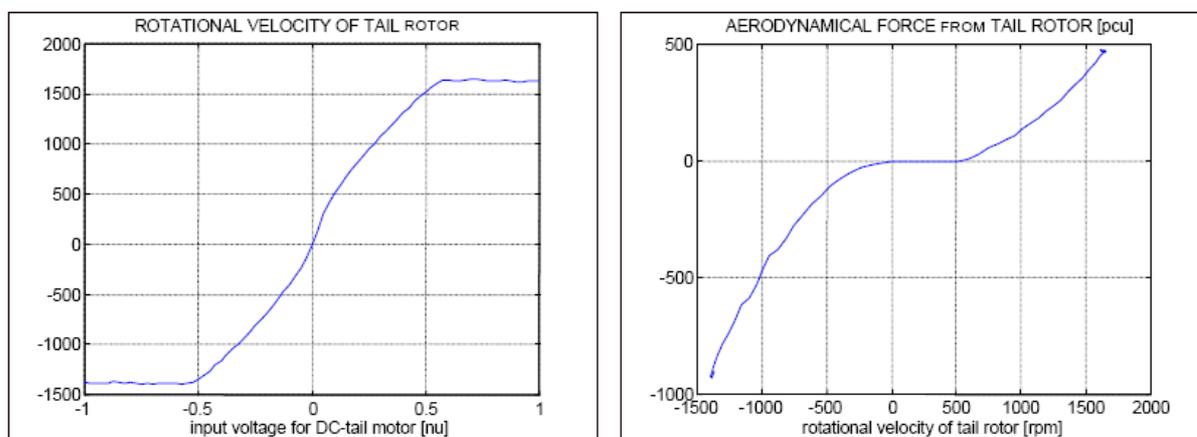
*Figure A.1 : graphes expérimentaux représentant la vitesse de rotation du rotor principal en fonction de la tension à l'entrée du rotor (à gauche) et la force aérodynamique engendrée par la rotation du rotor principal (à droite)*

L'utilisation de la fonction *polyfit* de Matlab, nous permet d'approximer ces courbes par des polynômes déjà donnés au chapitre II, le tracé de ces polynômes est illustré par les figures ci-dessous :

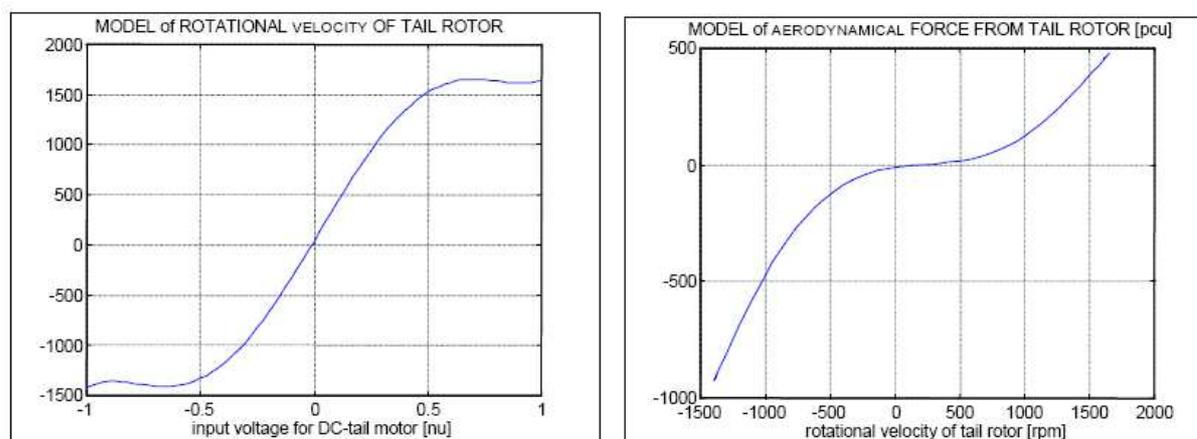


*Figure A.2 : approximation polynomiale des caractéristiques du rotor principal*

De même, les courbes expérimentales des forces aérodynamiques en fonction des vitesses de rotation des rotors, sont données par les figures suivantes :



**Figure A.3 :** graphes expérimentaux représentant la vitesse de rotation du rotor secondaire en fonction de la tension à l'entrée du rotor (à gauche) et la force aérodynamique engendrée par la rotation du rotor secondaire (à droite)



**Figure A.4 :** approximation polynomiale des caractéristiques du rotor de queue

Les simulations effectués dans ce travail et faites par l'éditeur ANFIS (toolbox fuzzy) donné figure (B.1). Les différentes opérations sont décrites par les étapes A, B, C sont p :

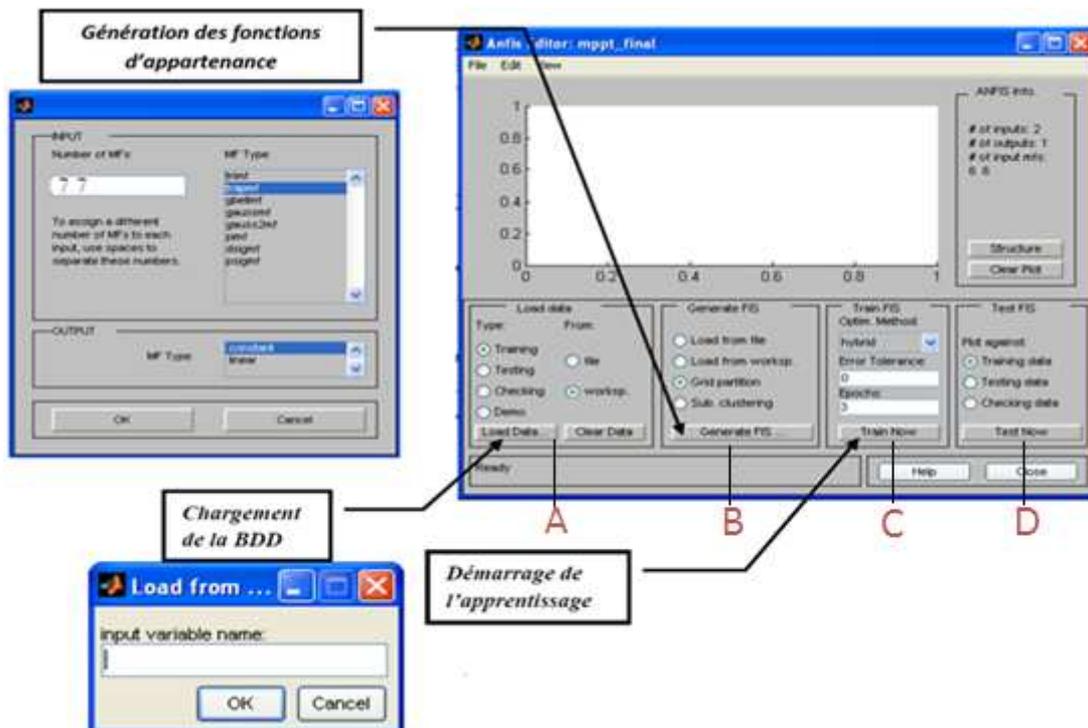


Fig. B.1 : éditeur ANFIS dans toolbox fuzzy sous Matlab

#### A. Chargement et effacement des données (Loading, and Clearing the Data) :

Pour charger un ensemble de données en utilisant la zone de chargement de données sur l'interface graphique:

1. Spécifiez le type de données.
2. Sélectionnez les données à partir d'un **fichier** ou l'**espace de travail** de MATLAB.
3. Cliquez sur **Load Data**.

Pour effacer un ensemble de données spécifiques définies à partir de l'interface graphique:

1. Dans la zone de données de charge, sélectionnez le type de données.

2. Cliquez sur effacer les données.

## **B. Générer ou charger un FIS (Generating or Loading the Initial FIS Structure) :**

Avant de commencer la formation de la FIS, vous devez spécifier une structure FIS (modèle initial). Pour spécifier la structure du modèle, effectuez l'une des tâches suivantes:

1. Charger une liste déjà sauvegardée Sugeno-type de structure de la FIS à partir d'un fichier ou l'espace de travail MATLAB.
2. Générer le modèle initial de la FIS en choisissant l'une des techniques de partitionnement suivantes

**Grid partition** (Grille partition) génère une sortie unique de type Sugeno FIS en utilisant la grille de partitionnement des données.

**Sub. Clustering** (Sous. Clustering) - Génère un modèle initial de la formation ANFIS

En appliquant d'abord le regroupement soustractive sur les données.

Pour afficher une représentation graphique de la structure du modèle initial de la FIS, cliquez sur Structure.

## **C. Apprentissage du FIS (training the FIS) :**

L'apprentissage du FIS qui permet l'adaptation du réseau est fait selon les étapes suivantes :

1. En Optim. Méthode, choisissez **hybride** ou **backpropaga** que la méthode d'optimisation (méthode d'adaptation des paramètres des fonctions d'appartenances).
2. Entrez le nombre **Epoques** et de la **tolérance d'erreur** pour définir les critères d'arrêt de l'apprentissage. Le processus de formation du FIS s'arrête à chaque fois le nombre maximum est atteint époque ou l'erreur de tolérance est atteinte.
3. Cliquez Former maintenant pour former le FIS.

Cette action permet de régler les paramètres de fonction d'appartenance et affiche les parcelles d'erreur.

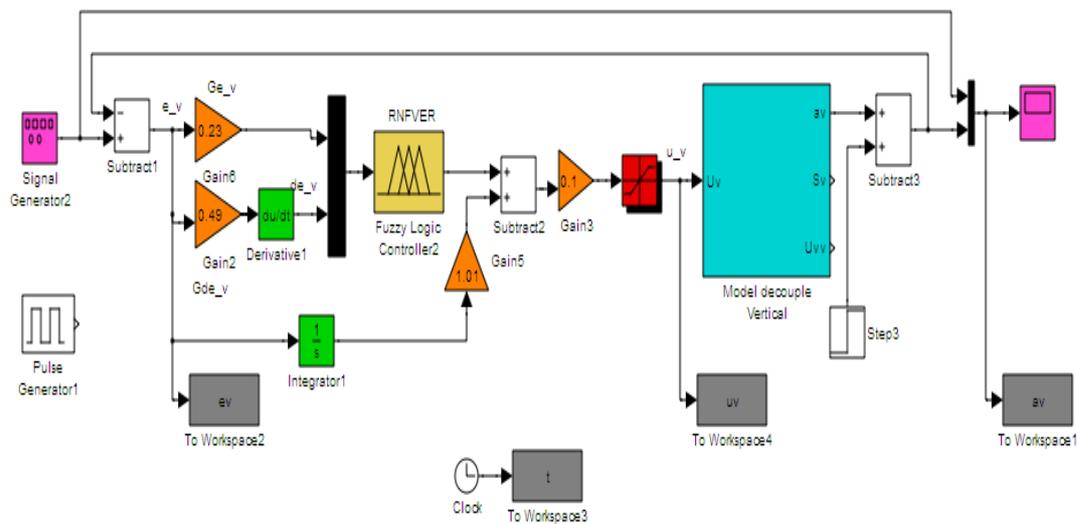
## **D. Validation du FIS (Validating the Trained FIS) :**

Une fois que le FIS est formé, l'étape de validation du modèle FIS est faite par un teste (**Testing**) ou le contrôle (**Checking**) des données qui diffère de celle utilisée pour former les FIS.

Pour valider les FIS formés:

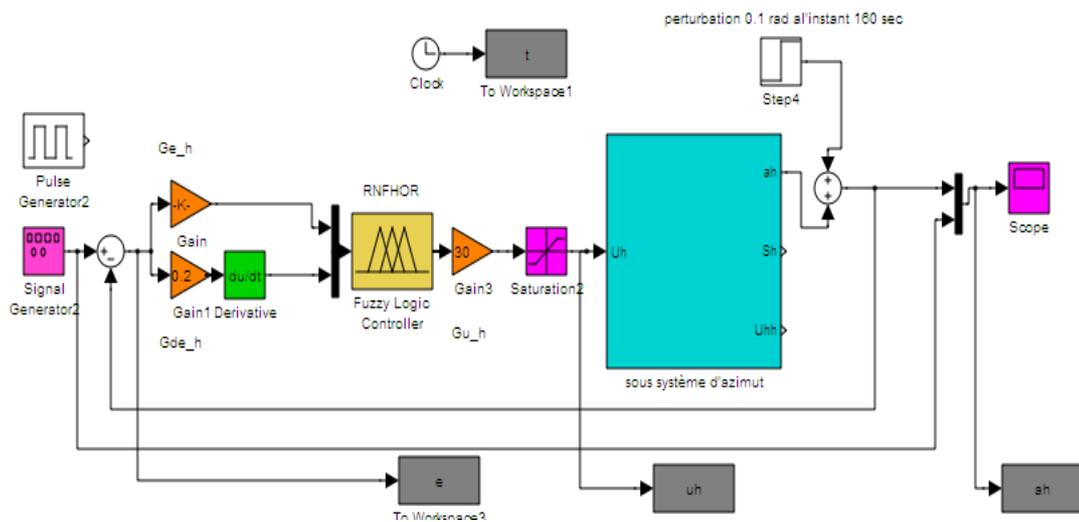
1. Sélectionnez le jeu de la validation des données et cliquez sur Load Data.
2. Cliquez sur Tester maintenant.

Les schémas des commandes pour les modèles découpler (vertical et horizontale) et les modèles couplés (décentralise et centralise) sur les figures suivant :



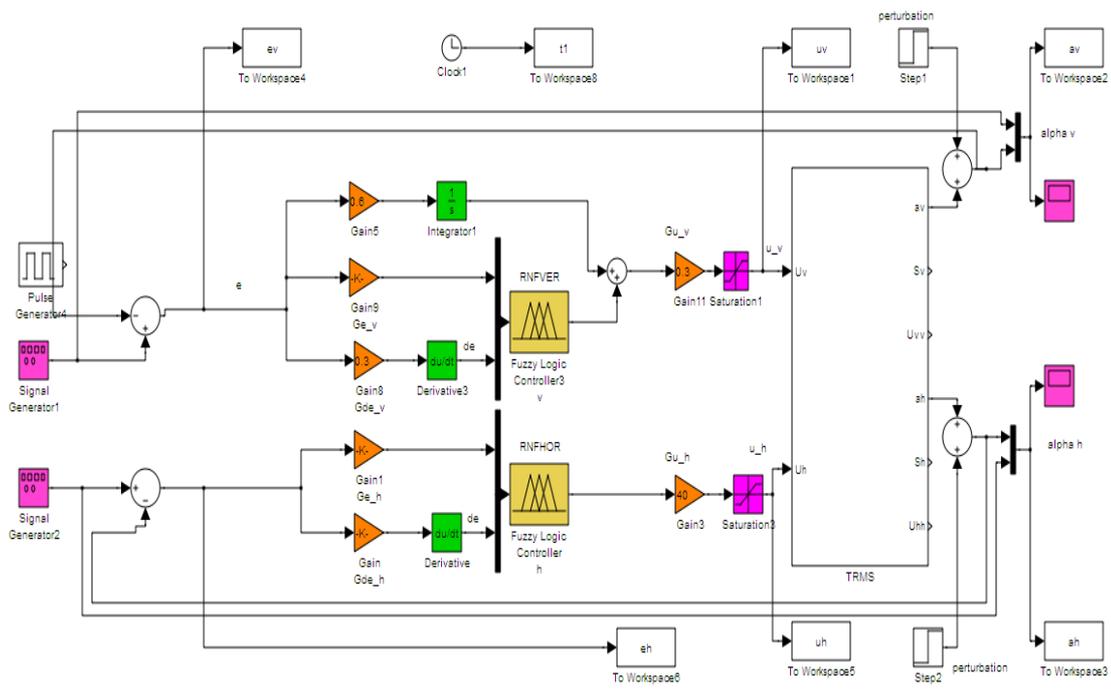
**Fig. B.1.** schéma de commande sous Matlab/Simulink de l'axe vertical par la correcteur

*Par la correcteur RNFVER*

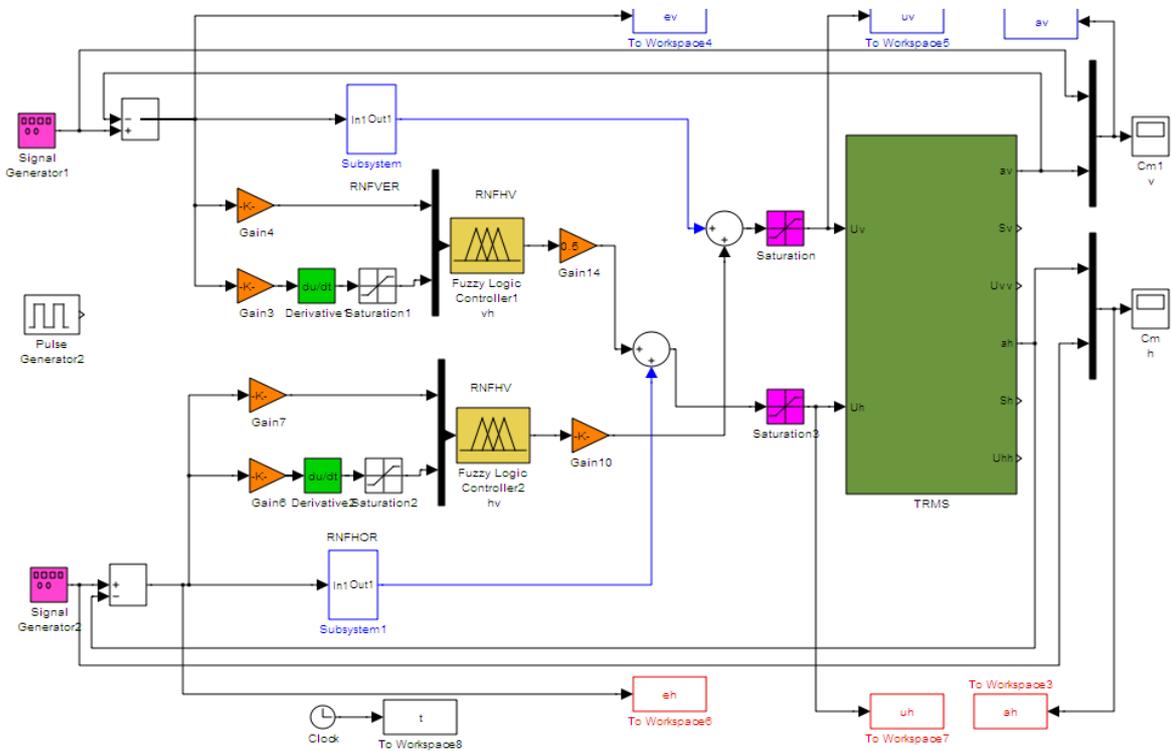


**Fig.2. :** schéma de commande sous Matlab/Simulink de l'axe horizontale par la correcteur

*Par la correcteur RNFHOR*



**Fig. B.3 :** Schéma de simulation du modèle TRMS non linéaire couplé  
Par la méthode décentralisée (RNFVER et RNFHOR)



**Fig. B.4 :** Schéma de simulation du modèle TRMS non linéaire couplé  
Par la méthode centralisée (RNFVER et RNFHOR et RNFHV et RNFVH )