

Chapitre Cinc

APPLICATION ET COMMANDE D'UN ROBOT MANIPULATEUR

5.1. Introduction	93
5.2. Modélisation Géométrique	93
5.3. Modélisation Cinématique	98
5.4. Modélisation Dynamique (MDD) et (MDI)	99
5.5. Procédure de Navigation	99
5.6. Méthode d'évitement	99
5.7. Le Contrôleur Flou pour l'évitement d'obstacle	100
5.8. Le Contrôleur Neuro-Flou pour l'évitement d'obstacle	115
5.9 Conclusion	14

APPLICATION ET COMMANDE D'UN ROBOT MANIPULATEUR

5.1. Introduction

La conception et la commande des robots nécessitent le calcul de certains modèles mathématiques, tels que :

- ✓ les modèles de transformation entre l'espace opérationnel (dans lequel est définie la situation de l'organe terminal), et l'espace articulaire (dans lequel est définie la configuration du robot), on distingue parmi ces modèles :
- ✓ Les modèle géométrique direct et inverse, qui expriment la situation de l'organe terminal en fonction de variable articulaire du mécanisme et inversement.
- ✓ Les modèle cinématique directe et inverse exprime la vitesse de l'organe terminal en fonction des vitesses articulaire et inversement.
- ✓ les modèle dynamique définissent l'équation du mouvement du robot, qui permettent d'établir le relation entre le couple du force exercée par les actionneurs et les positions, vitesses et accélérations des articulations

Le formalisme mathématique fait appel aux matrices de transformation homogènes de dimension (4x4). La matrice homogène ${}^i T_j$ représente la transformation permettant de passer du repère R_i au repère R_j

$${}^i T_j = \begin{bmatrix} iA_j & iA_j & iA_j & iA_j \\ 0 & 0 & 0 & 1 \end{bmatrix} = \begin{bmatrix} is_j & inj & iaj & ip_j \\ 0 & 0 & 0 & 1 \end{bmatrix} \quad (5.1)$$

Ou is_j, inj , et iaj désignent respectivement les vecteur unitaires suivent les axes x_j, y_j et z_j du repère R_j exprimés dans le repère R_i , et ou ip_j est le vecteur exprimant l'origine du repère R_j , les vecteur is_j, inj, iaj de la matrice d'orientation iA_j sont les cosinus directeur.

5.2. Modélisation Géométrique

5.2.1. Description géométrique du bras à étudier

le bras manipulateur à étudier est un robot articulé à 6 d. d. l "robot staubil RX-90 (figure 5.1)".

La cinématique du porteur est de type RRR et le poignet comporte trois rotations. On détermine ensuite les paramètres géométriques du robot. le placement des repères est indique sur la figure 5.1 et les paramètres géométriques sont donnés dans le tableau 5.1.

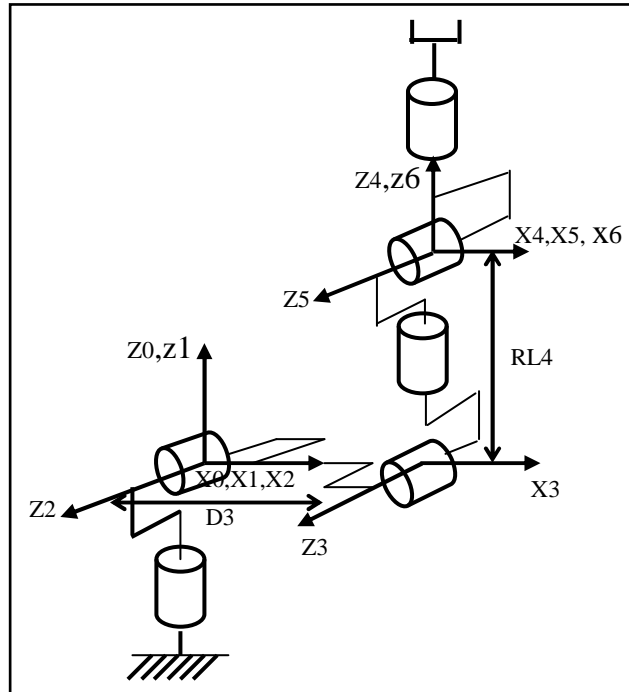


Figure 5.1 - placement des repères et notation pour le robot staubil RX-90

j	σ_j	α_j	d_j	θ_j	r_j
1	0	0	0	θ_1	0
2	0	$\pi/2$	0	θ_2	0
3	0	0	D3	θ_3	0
4	0	$-\pi/2$	0	θ_4	RL4
5	0	$\pi/2$	0	θ_5	0
6	0	$-\pi/2$	0	θ_6	0

Tableau 5.1 - paramètres géométriques du robot staubil RX-90

5.2.2. Modèle géométrique direct du bras à étudier (robot staubil RX-90)

A partir du tableau (5.1), la relation (1.7) permet d'écrire les matrices de transformation élémentaires $J^{-1}T_J$ le produit 0T_6 de ces matrices a pour composantes :

$$s_x = c_1(c_{23}(c_4c_5c_6 - s_4s_6) - s_{23}s_5c_6) - s_1(s_4c_5c_6 + c_4s_6)$$

$$s_y = s_1(c_{23}(c_4c_5c_6 - s_4s_6) - s_{23}s_5c_6) + c_1(s_4c_5c_6 + c_4s_6)$$

$$\begin{aligned}
sz &= s23(c4c5c6 - s4s6) + c23s5c6 \\
nx &= c1(-c23(c4c5s6 + s4c6) + s23s5s6) + s1(s4c5s6 - c4c6) \\
ny &= s1(-c23(c4c5s6 + s4c6) + s23s5s6) - c1(s4c5s6 - c4c6) \\
nz &= -s23(c4c5s6 + s4c6) - c23s5s6 \\
ax &= -c1(c23c4s5 + s23c5) + s1s4s5 \\
ay &= -s1(c23c4s5 + s23c5) - c1s4s5 \\
az &= -s23c4s5 + c23c5 \\
px &= -c1(s23RL4 - c2D3) \\
py &= -s1(s23RL4 - c2D3) \\
pz &= c23RL4 + s2D3
\end{aligned}$$

avec : $c23 = \cos(\theta_2 + \theta_3)$ et $s23 = \sin(\theta_2 + \theta_3)$

5.2.3. Modèle géométrique inverse du bras à étudier (robot Staubli RX-90)

Principe de la méthode de Paul :

considérons un robot manipulateur dont la matrice de transformation homogène a pour expression :

$${}^0T_n = {}^0T_1(q_1) {}^1T_2(q_2) \dots {}^{n-1}T_n(q_n) \quad (5.2)$$

soit U_0 la situation désirée telle que :

$$U_0 = \begin{bmatrix} sx & nx & ax & px \\ sy & ny & ay & py \\ sz & nz & az & pz \\ 0 & 0 & 0 & 1 \end{bmatrix} \quad (5.3)$$

on cherche à résoudre le système d'équation suivant :

$$U_0 = {}^0T_1(q_1) {}^1T_2(q_2) \dots {}^{n-1}T_n(q_n) \quad (5.4)$$

Pour trouver les solutions de l'équation (5.4), Paul [PAU 81] a proposé une méthode qui consiste à pré multiplier successivement les deux membres de l'équation (5.4) par les matrices ${}^jT_{j-1}$ pour j variant de 1 à $n-1$, opérations qui permettent d'isoler et d'identifier l'une après l'autre les variables articulaires que l'on recherche.

pour un robot à six degrés de liberté par exemple, on procède comme suit :

- multiplicateur à gauche de l'expression (5.4) par 1T_0 :

$${}^1T_0 U_0 = {}^1T_2 {}^2T_3 {}^3T_4 {}^4T_5 {}^5T_6 \quad (5.5)$$

Le terme de droite est fonction des variables q_2, \dots, q_6 le terme de gauche n'est fonction que des éléments de U_0 et de la variable q_1 ;

- Identification terme à des deux membres de l'équation (5.5). On est ramené à un système à une ou deux équations fonction de q_1 uniquement, dont la structure appartient à un type particulier parmi une dizaine de types possible.
- Multiplication à gauche de l'expression (5.5) par 2T_1 et calcul de q_2 .

la succession des équations permettant le calcul de tout les q_j est la suivante :

$$\begin{aligned}
 U_0 &= {}^0T_1 {}^1T_2 {}^2T_3 {}^3T_4 {}^4T_5 {}^5T_6 \\
 {}^1T_0 U_0 &= {}^1T_2 {}^2T_3 {}^3T_4 {}^4T_5 {}^5T_6 \\
 {}^2T_1 U_1 &= {}^2T_3 {}^3T_4 {}^4T_5 {}^5T_6 \\
 {}^3T_2 U_2 &= {}^3T_4 {}^4T_5 {}^5T_6 \\
 {}^4T_3 U_3 &= {}^4T_5 {}^5T_6 \\
 {}^5T_4 U_4 &= {}^5T_6
 \end{aligned} \tag{5.6}$$

avec : $U_{j+1} = {}^{j+1}T_j U_j$ pour $j=0, \dots, 4$.

Le modèle géométrique inverse du robot *stabil RX-90* tous calculs faits, on obtient les solutions suivantes :

$$\theta_1 = \text{atan2}(P_y, P_x)$$

$$\theta_1' = \theta_1 + \Pi$$

$$\theta_2 = \text{atan2}(s_2, c_2)$$

avec :

$$c_2 = \frac{YZ - \varepsilon X \sqrt{X^2 + Y^2 - Z^2}}{X^2 + Y^2}$$

$$\varepsilon = \pm 1$$

$$s_2 = \frac{XZ + \varepsilon Y \sqrt{X^2 + Y^2 - Z^2}}{X^2 + Y^2}$$

$$B_1 = P_{XC1} + P_{YS1}$$

$$X = -2P_Z \cdot D_3$$

$$Y = -2B_1 \cdot D_3$$

$$Z = (RL_4)^2 - (D_3)^2 - (P_Z)^2 - (B_1)^2$$

$$\theta_3 = a \tan 2\left(\frac{-P_{ZS2} - B_1 c_2 + D_3}{RL_4}, \frac{-B_1 s_2 + P_{ZC2}}{RL_4}\right)$$

$$\theta_3 = a \tan 2\left(\frac{-P_{ZS2} - B_1 c_2 + D_3}{RL_4}, \frac{-B_1 s_2 + P_Z}{RL_4}\right)$$

$$\theta_4 = a \tan 2(s_1 a_x - c_1 a_y, -c_2 (c_1 a_x + s_1 a_y) - s_2 a_z)$$

$$\theta_4' = \theta_4 + \pi$$

$$\theta_5 = a \tan 2(s_5, c_5)$$

$$s5 = -c4[c23(c1ax + s1ay) + s23az] + s4(s1ax - c1ay)$$

Avec: $c5 = -s23(c1ax + s1ay) + c23az$

$$\theta_6 = a \tan 2(s6, c6)$$

avec: $s6 = -c4(c1sx - c1sy) - s4[c23(c1sx + s1sy) + s23sz]$

$$c6 = -c4(s1nx - c1ny) - s4[c23(c1nx + s1ny) + s23nz]$$

$$c1 = \cos(\theta_1)$$

$$s1 = \sin(\theta_1)$$

avec : $c23 = \cos(\theta_2, \theta_3)$

$$c4 = \cos(\theta_4)$$

$$s4 = \sin(\theta_4)$$

$$s23 = \sin(\theta_2, \theta_3)$$

La figure 5.2 représente les différentes positions pour atteindre *robot staubil RX-90* un point dans l'espace.

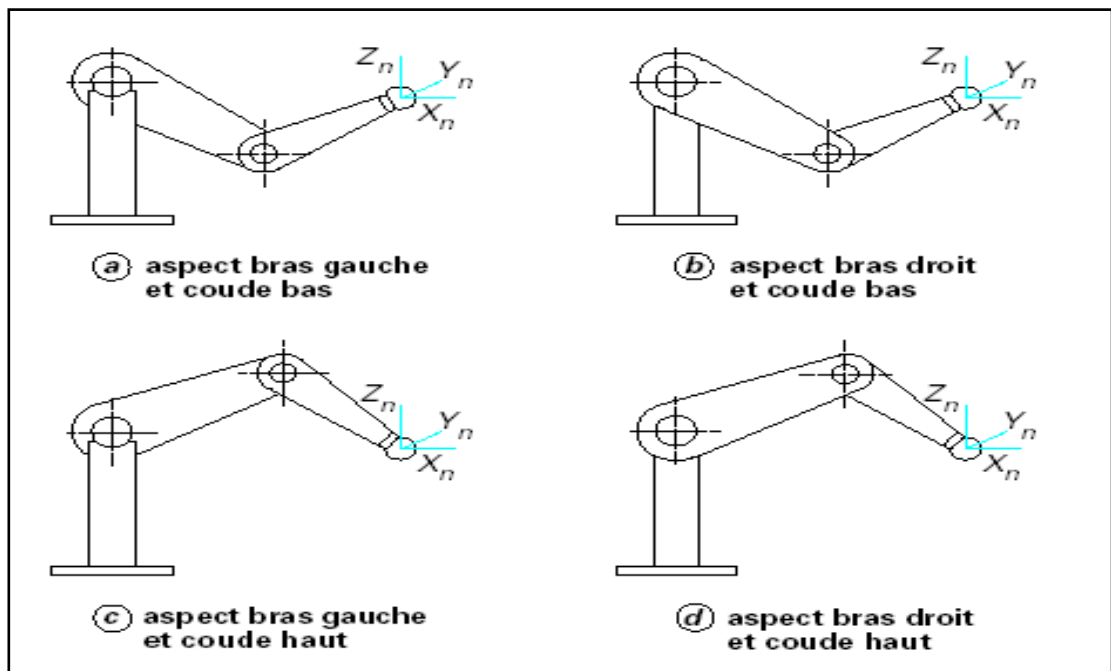


Figure 5.2 - différentes positions pour atteindre point dans l'espace

Remarques

positions singulières

- Lorsque $p_x=p_y=0$ ce qui correspond à $S23RL4-C2D3=0$, le point $O4$ se trouve sur l'axe $Z0$ (voir figure 5.3.a).Les deux arguments utilisés pour le calcul de θ_1 sont nuls et, de ce fait, θ_1 est indéterminé. on peut fixer θ_1 à une valeur quelconque
- Lorsque: $C23(C1ax+s1ay)+s23az=Hx=0$ et $s1ax-c1ay=Hz=0$, la fonction atan2 utilisée pour calculer θ_4 a ses deux arguments nuls et est donc indéterminée. (Voir figure5.3.b). on peut fixer θ_4 à une valeur quelconque

- c) Une troisième position singulière, lorsque $C3=0$ sera mise en évidence avec le modèle cinématique. cette singularité ne pose pas de problème pour le modèle géométrique inverse (voir figure 5.3.c).

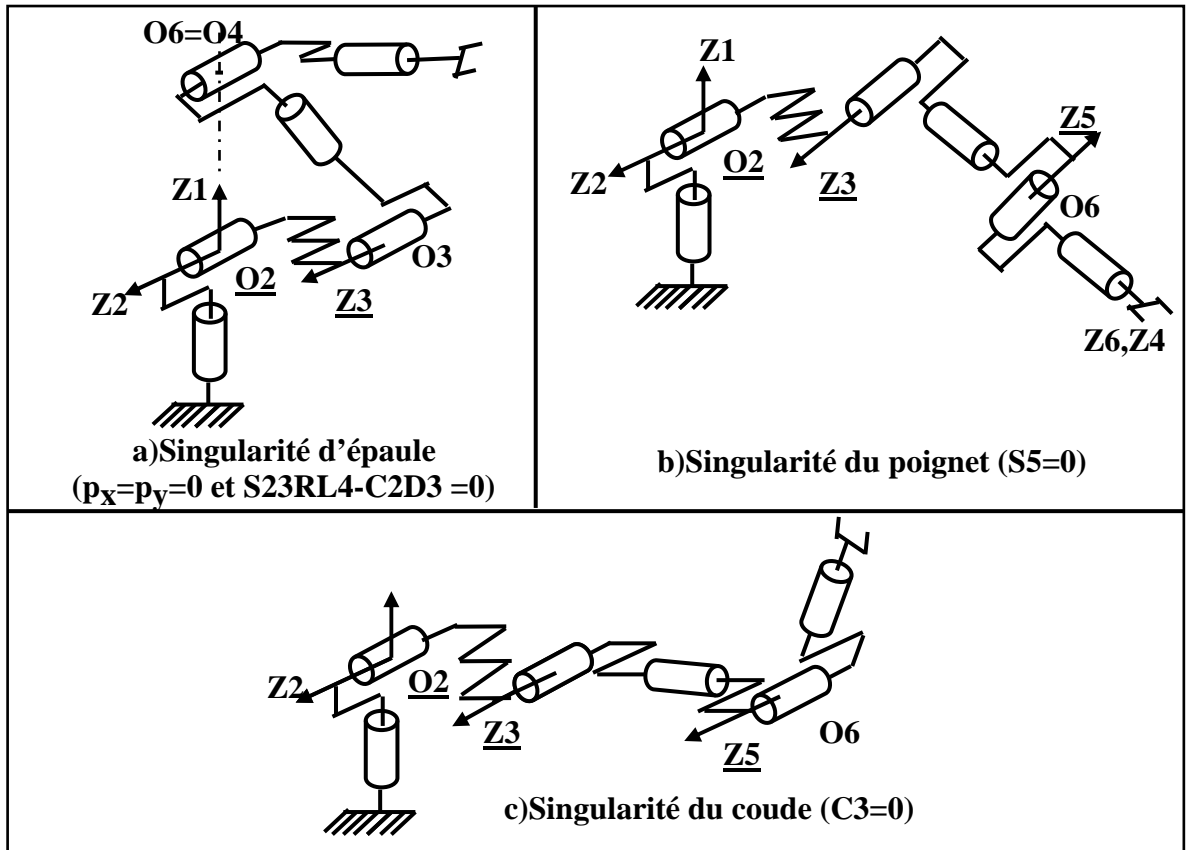


Figure 5.3 - positions singulières du robot Staubli RX-90

5.3. Modélisation Cinématique

La cinématique complète la modélisation géométrique en établissant les relations entre les vitesses des paramètres articulation (q) et les vitesses des variables articulation. ces relations sont désignées par la notation $\dot{X} = J.\dot{q}$ ou J est une matrice qui dépend de la configuration du SMA.

La propriété évidente du modèle cinématique est sa linéarité par rapport aux vitesses. il est donc a priori plus aisée à manipuler que le modèle géométrique,

5.3.1. Modélisation cinématique directe (MCD)

Le modèle cinématique directe d'un robot manipulateur décrit les vitesses des coordonnées opérationnelles (\dot{X}), en fonction des vitesses articulaires (\dot{q}). Il est noté :

$$\dot{X} = J.\dot{q} \quad (5.7)$$

ou: J désigne la matrice jacobienne de dimension ($m \times n$)

5.3.2. Modélisation cinématique inverse (MCI)

L'objectif du modèle cinématique inverse d'un robot manipulateur, sont de calculer les vitesses des coordonnées articulaires (\dot{q}), en fonction des vitesses opérationnelles (\dot{X}).

Il est noté :

$$\dot{q} = J^{-1} \dot{X} \quad (5.8)$$

5.4. Modélisation Dynamique (MDD) et (MDI)

Le modèle dynamique est la relation entre les couples (et/ou forces) appliqués aux actionneurs et les positions, vitesses et accélérations articulaires. on représente le modèle dynamique par une relation de la forme :

$$\Gamma = f(q, \dot{q}, \ddot{q}, fe) \quad (5.9)$$

avec :

- Γ : vecteur des couples/forces des actionneurs, selon que l'articulation est rotoïde ou prismatique. Dans la suite, on écrira tout simplement couples
- q : vecteur des positions articulaires
- \dot{q} : vecteur des vitesses articulaires
- \ddot{q} : vecteur des accélérations articulaires
- fe : vecteur représentant l'effort extérieur (forces et moments) qu'exerce le robot sur l'environnement .

On convient d'appeler *modèle dynamique inverse*, ou tout simplement modèle dynamique, la relation de la forme (5.9).

Le modèle dynamique direct est celui qui exprime les accélérations en fonction des positions, vitesses et couples des articulations. il est alors représenté par la relation :

$$\ddot{q} = g(q, \dot{q}, \Gamma, fe) \quad (5.10)$$

5.5. Procédure de Navigation

Le problème de commande d'un bras manipulateur en présence d'obstacle, s'agit de trouver une méthode à amener le bras d'un point de départ (source) à point final (cible) en évitant l'obstacle rencontré.

5.6. Méthode d'évitement

La figure 5.4 décrit l'état d'un bras manipulateur se dirigeant vers un point cible en évitant un obstacle fixe.

A partir de deux vecteur(cible et évitement)en calculant le vecteur du direction qui est égale à la somme vectorielle des vecteurs V_{cib} et V_{evi} ou :

$$\vec{V}_{dir} = \vec{V}_{evi} + \vec{V}_{cib} \quad (5.11)$$

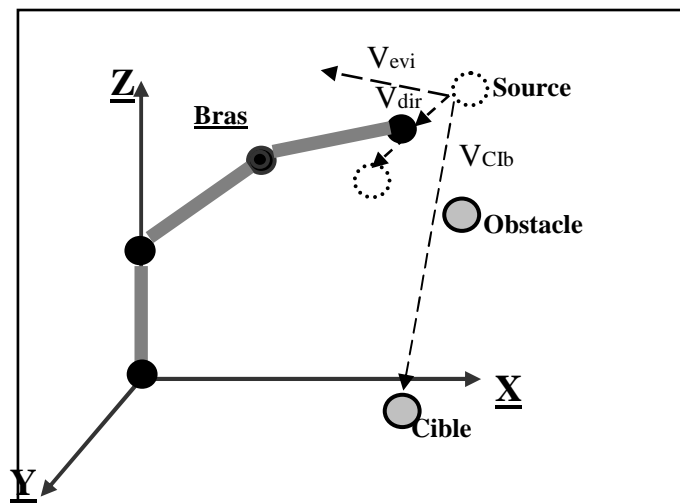


Figure 5.4- dirige vers un point (cible)

- V_{cib} : vecteur de la cible.
- V_{evi} : vecteur d'évitement.
- V_{dir} : vecteur du direction.

On peut calculer directement Le vecteur de la cible V_{cib} à partir de la position de cible et pour le vecteur d'évitement V_{evi} on utilise un raisonnement par la logique floue.

5.7. Le Contrôleur Flou pour l'évitement d'obstacle

L'objectif est la création d'un contrôleur flou simple capable d'évaluer le vecteur d'évitement V_{evi} correspond à la position relative actuelle d'obstacle (figure 5.5).

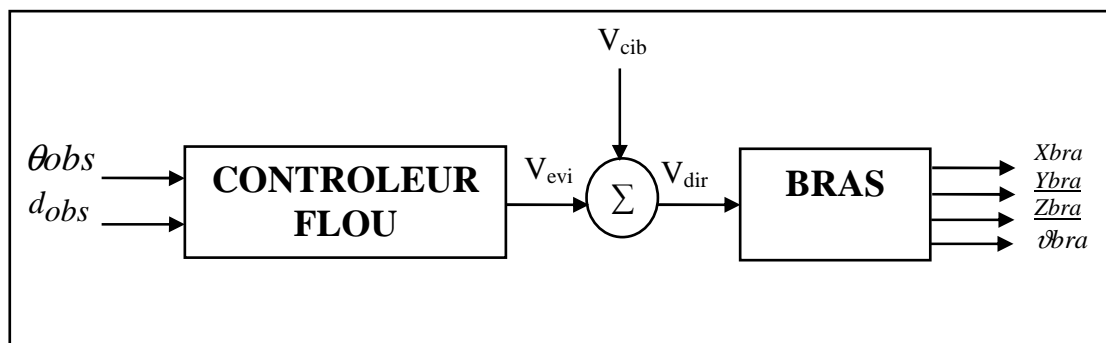


Figure 5.5- Entrées/sorties du contrôleur flou

Le contrôleur possède deux entrées et une sortie, les entrées sont l'angle et la distance par rapport à l'obstacle, la sortie est le vecteur d'évitement. Pour le bras ; l'entrée est le vecteur direction et les sorties sont les deux coordonnées absolues (x_{bra} , y_{bra} , Z_{bra}) et l'angle de bras ϑ_{bra} .

En choisissant le meilleur type de contrôleur flou, c'est le type qui a été proposé par Takagi-Ségéno. pour traiter ce problème, avec les fonctions d'appartenance gaussiennes..

5.7.1 La Fuzzification :

✓ La Fuzzification de l'angle d'obstacle (θ_{obs}) :

On suppose que le robot (bras manipulateur) peut percevoir un obstacle dans une direction qui appartient à l'intervalle $[-90^\circ 90^\circ]$. La fonction d'appartenance est représentée en sept sous-ensembles flous (figure 5.6) :

- **GN** : Large négatif
- **MN** : Moyenne négatif .
- **PN** : Petite négatif .
- **EZ** : Environnement de Zéro.
- **PP** : Petite positif.
- **MP** : Moyenne positif.
- **GP** : Large positif.

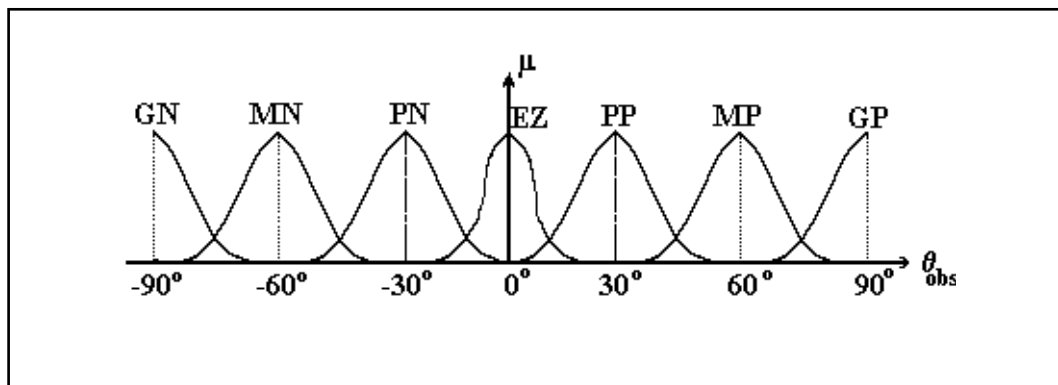


Figure 5.6. Fuzzification de l'angle d'obstacle θ_{obs}

✓ La Fuzzification de distance d'obstacle (d_{obs}) :

pour la fonction d'appartenance de la distance (d_{obs}) « distance entre l'élément terminal du bras et l'obstacle », On choisit trois sous-ensembles flous

- **P** : Petite.
- **M** : Moyenne.
- **G** : Grande.

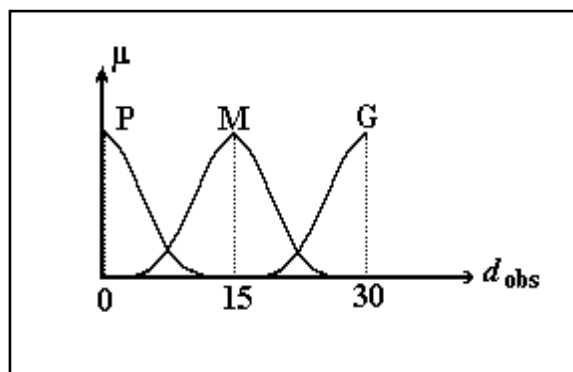


Figure 5.7- Fuzzification de la distance

✓ **La fuzzification de l'angle d'évitement ϑ_{evi} :**

on définit les fonctions d'appartenance de l'angle d'évitement ϑ_{evi} par neuf(09) sous ensembles flous appartenant à l'intervalle $[-135, 135]$.

- **GGN** : Grande Grande Négative.
- **GN** : Grande Négative.
- **MN** : Moyenne Négative.
- **PN** : Petite Négative.
- **EZ** : Environnement de Zéro.
- **PP** : Petite Positive
- **MP** : Moyenne Positive.
- **GP** : Grande Positive
- **GGP** : Grande Grande Positive.

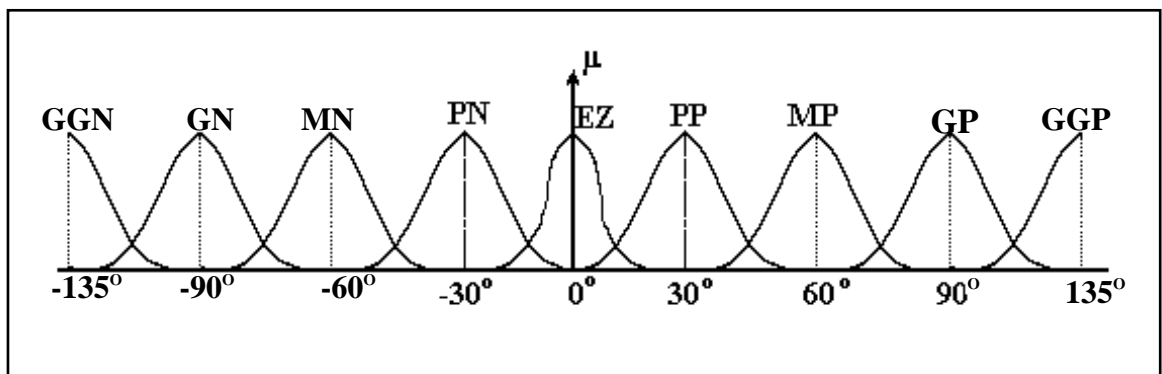


Figure 5.8- Fuzzification de l'angle d'évitement

5.7.2. La base des règles floues du contrôleur (l'inférence)

Le jeu de règles peut être décrit sous forme de tableau, il s'interprète comme une visualisation du plan de phase, dans le cas où les variables d'entrée sont : l'angle d'évitement et la distance, chaque case du tableau indique la déduction d'une règle.

pour cette situation on a trois cas (trois tableaux d'inférence) :

1^{ère} cas : $\vartheta_{\text{obstacle}}$ **supérieure** à $\vartheta_{\text{source}}$ (lorgne terminal) et **inférieure** à ϑ_{cible} ($\vartheta_{\text{obstacle}} \in [\vartheta_{\text{source}} \vartheta_{\text{cible}}]$) c'est à dire l'évitement se fait à gauche comme l'indique la figure ci dessous.

2^{ème} cas : $\vartheta_{\text{obstacle}}$ **inférieure** à $\vartheta_{\text{source}}$ (lorgne terminal) et **supérieure** à ϑ_{cible} ($\vartheta_{\text{obstacle}} \in [\vartheta_{\text{source}} \vartheta_{\text{cible}}]$) c'est à dire l'évitement se fait à droite comme l'indique la figure ci dessous.

3^{ème} cas : $\vartheta_{\text{obstacle}}$ **supérieure ou inférieure** à $\vartheta_{\text{source}}$ et ϑ_{cible} ($\vartheta_{\text{obstacle}} \notin [\vartheta_{\text{source}} \vartheta_{\text{cible}}]$) c'est à dire l'évitement se fait à gauche ou droite comme l'indique la figure ci – dessous.

Etude du 1^{ère} cas :

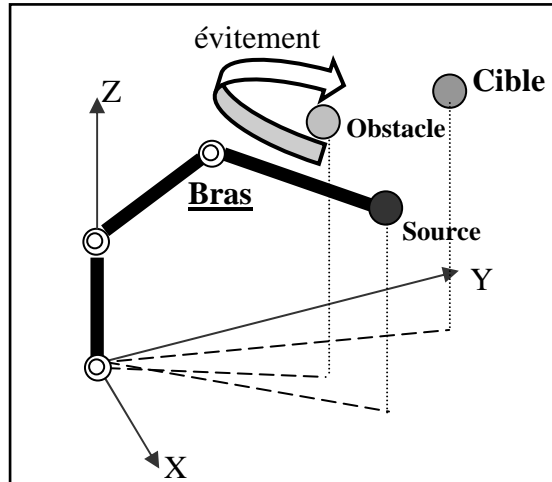


Figure 5.9- évitement gauche

On compte 21 règles pour l'inférence qui sont :

- SI (dobs P est et \varnothing obs est EZ) ALORS (\varnothing rep est GGN), OU
- SI (dobs M est et \varnothing obs est EZ) ALORS (\varnothing rep est GGN), OU
- SI (dobs G est et \varnothing obs est EZ) ALORS (\varnothing rep est GN), OU
- SI (dobs P est et \varnothing obs est PP) ALORS (\varnothing rep est GGN), OU
- SI (dobs M est et \varnothing obs est PP) ALORS (\varnothing rep est GN), OU
- SI (dobs G est et \varnothing obs est PP) ALORS (\varnothing rep est MN), OU
- SI (dobs P est et \varnothing obs est MP) ALORS (\varnothing rep est GN), OU
- SI (dobs M est et \varnothing obs est MP) ALORS (\varnothing rep est MN), OU
- SI (dobs G est et \varnothing obs est MP) ALORS (\varnothing rep est PN), OU
- SI (dobs P est et \varnothing obs est GP) ALORS (\varnothing rep est MN), OU
- SI (dobs M est et \varnothing obs est GP) ALORS (\varnothing rep est PN), OU
- SI (dobs G est et \varnothing obs est GP) ALORS (\varnothing rep est EZ), OU
- SI (dobs P est et \varnothing obs est PN) ALORS (\varnothing rep est GGN), OU
- SI (dobs M est et \varnothing obs est PN) ALORS (\varnothing rep est GGN), OU
- SI (dobs G est et \varnothing obs est PN) ALORS (\varnothing rep est GGN), OU
- SI (dobs P est et \varnothing obs est MN) ALORS (\varnothing rep est GGN), OU
- SI (dobs M est et \varnothing obs est MN) ALORS (\varnothing rep est GGN), OU
- SI (dobs G est et \varnothing obs est MN) ALORS (\varnothing rep est GGN), OU

- SI (dobs P est et ϑ_{obs} est GN) ALORS (ϑ_{rep} est GGN), OU
- SI (dobs M est et ϑ_{obs} est GN) ALORS (ϑ_{rep} est GGN), OU
- SI (dobs G est et ϑ_{obs} est GN) ALORS (ϑ_{rep} est GGN).

Les règles sont résumées dans le tableau suivant :

ϑ_{rep}		ϑ_{obs}						
		GN	MN	PN	EZ	PP	MP	GP
d_{obs}	P	GGN	GGN	GGN	GGN	GGN	GN	MN
	M	GGN	GGN	GGN	GGN	GN	MN	PN
	G	GGN	GGN	GGN	GN	MN	PN	EZ

Tableau 5.2 La base des règles de l'angle d'évitement

Etude du 2^{ème} cas :

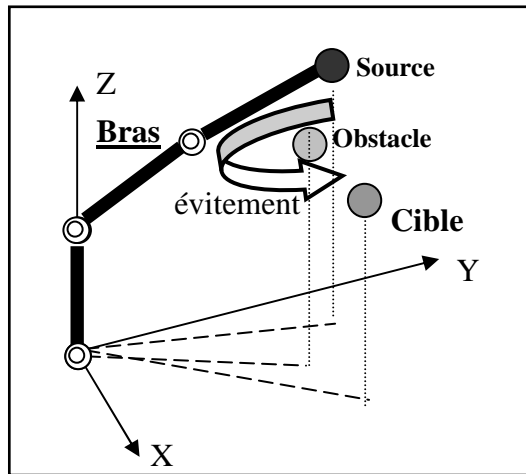


Figure 5.10- évitement droite

On compte 21 règles pour l'inférence qui sont :

- SI (dobs P est et ϑ_{obs} est EZ) ALORS (ϑ_{rep} est GGP), OU
- SI (dobs M est et ϑ_{obs} est EZ) ALORS (ϑ_{rep} est GP), OU
- SI (dobs G est et ϑ_{obs} est EZ) ALORS (ϑ_{rep} est MP), OU
- SI (dobs P est et ϑ_{obs} est PN) ALORS (ϑ_{rep} est GP), OU
- SI (dobs M est et ϑ_{obs} est PN) ALORS (ϑ_{rep} est MP), OU
- SI (dobs G est et ϑ_{obs} est PN) ALORS (ϑ_{rep} est PP), OU
- SI (dobs P est et ϑ_{obs} est MN) ALORS (ϑ_{rep} est MP), OU
- SI (dobs M est et ϑ_{obs} est MN) ALORS (ϑ_{rep} est PP), OU
- SI (dobs G est et ϑ_{obs} est MN) ALORS (ϑ_{rep} est EZ), OU
- SI (dobs P est et ϑ_{obs} est GN) ALORS (ϑ_{rep} est PP), OU

SI	(dobs M est et ϑ_{obs} est GN)	ALORS	(ϑ_{rep} est PP),	OU
SI	(dobs G est et ϑ_{obs} est GN)	ALORS	(ϑ_{rep} est EZ),	OU
SI	(dobs P est et ϑ_{obs} est PP)	ALORS	(ϑ_{rep} est GGP),	OU
SI	(dobs M est et ϑ_{obs} est PP)	ALORS	(ϑ_{rep} est GGP),	OU
SI	(dobs G est et ϑ_{obs} est PP)	ALORS	(ϑ_{rep} est GGP),	OU
SI	(dobs G est et ϑ_{obs} est MP)	ALORS	(ϑ_{rep} est GGP),	OU
SI	(dobs M est et ϑ_{obs} est MP)	ALORS	(ϑ_{rep} est GGP),	OU
SI	(dobs P est et ϑ_{obs} est MP)	ALORS	(ϑ_{rep} est GGP),	OU
SI	(dobs P est et ϑ_{obs} est GP)	ALORS	(ϑ_{rep} est GGP),	OU
SI	(dobs M est et ϑ_{obs} est GP)	ALORS	(ϑ_{rep} est GGP),	OU
SI	(dobs G est et ϑ_{obs} est GP)	ALORS	(ϑ_{rep} est GGP).	

Les règles sont résumées dans le tableau suivant :

ϑ_{rep}		ϑ_{obs}						
		GN	MN	PN	EZ	PP	MP	GP
d_{obs}	P	PP	MP	GP	GGP	GGP	GGP	GGP
	M	PP	PP	MP	GP	GGP	GGP	GGP
	G	EZ	EZ	PP	MP	GGP	GGP	GGP

Tableau 5.3 La base des règles de l'angle d'évitement

Etude du 3ème cas :

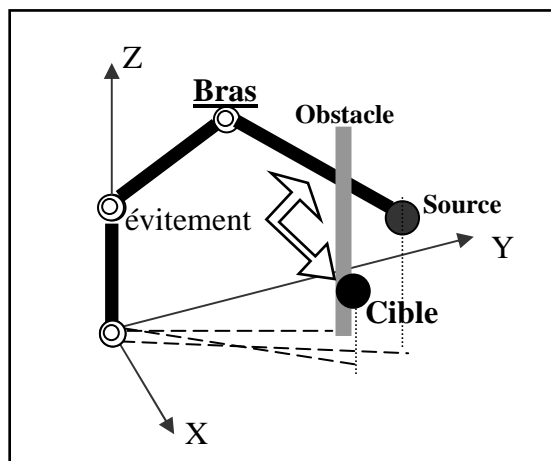


Figure 5.11- évitement gauche droite

On compte 21 règles pour l'inférence qui sont :

SI	(dobs P est et ϑ_{obs} est GN)	ALORS	(ϑ_{rep} est PP),	OU
----	---	-------	-------------------------------	----

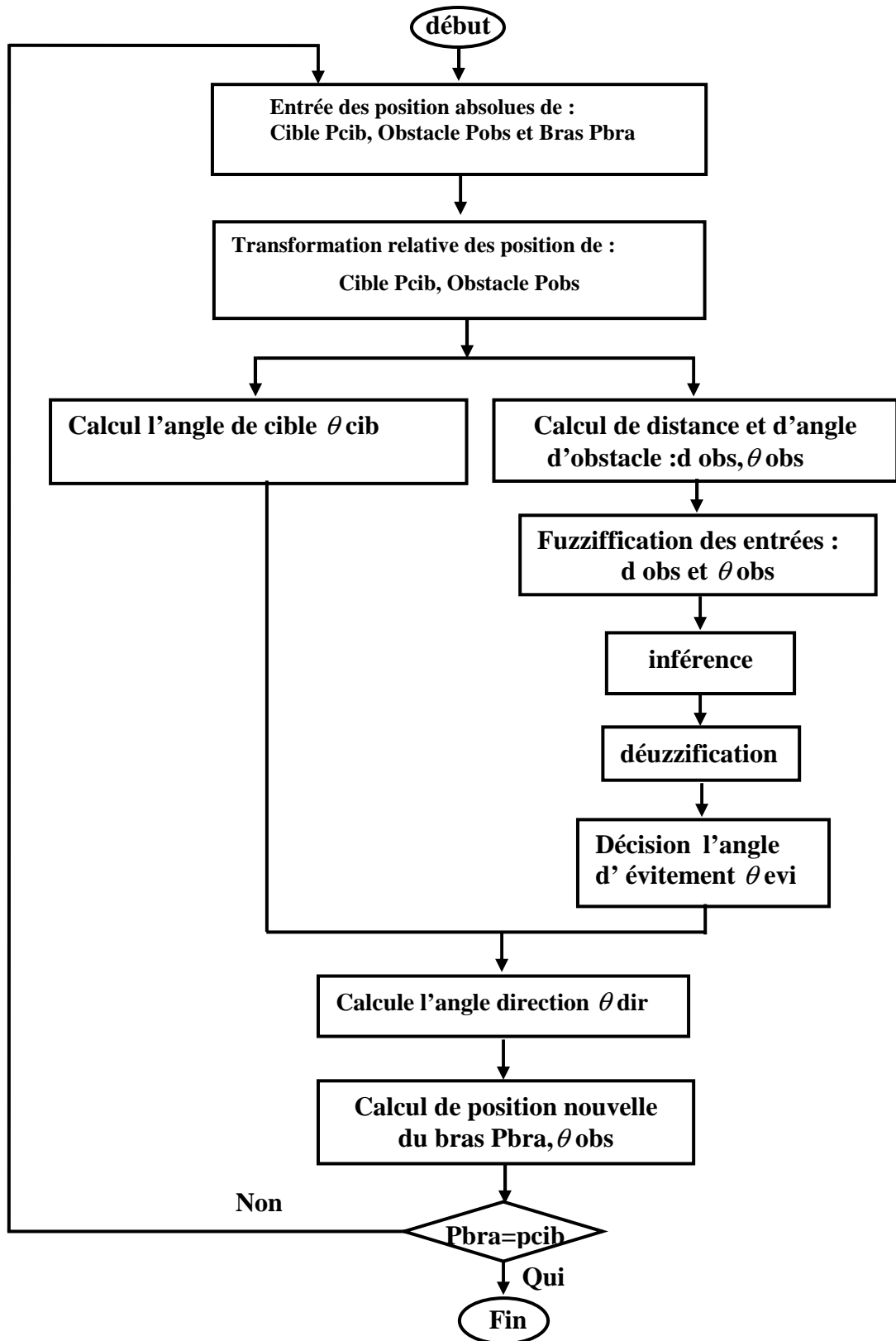
SI	(dobs P est et ϑ_{obs} est MN)	ALORS	(ϑ_{rep} est MP),	OU
SI	(dobs P est et ϑ_{obs} est PN)	ALORS	(ϑ_{rep} est GP),	OU
SI	(dobs P est et ϑ_{obs} est EZ)	ALORS	(ϑ_{rep} est GGN),	OU
SI	(dobs P est et ϑ_{obs} est PP)	ALORS	(ϑ_{rep} est GN),	OU
SI	(dobs P est et ϑ_{obs} est MP)	ALORS	(ϑ_{rep} est MN),	OU
SI	(dobs P est et ϑ_{obs} est GP)	ALORS	(ϑ_{rep} est PN),	OU
SI	(dobs M est et ϑ_{obs} est GN)	ALORS	(ϑ_{rep} est PP),	OU
SI	(dobs M est et ϑ_{obs} est MN)	ALORS	(ϑ_{rep} est PP),	OU
SI	(dobs M est et ϑ_{obs} est PN)	ALORS	(ϑ_{rep} est MP),	OU
SI	(dobs M est et ϑ_{obs} est EZ)	ALORS	(ϑ_{rep} est GN),	OU
SI	(dobs M est et ϑ_{obs} est PP)	ALORS	(ϑ_{rep} est MN),	OU
SI	(dobs M est et ϑ_{obs} est MP)	ALORS	(ϑ_{rep} est PN),	OU
SI	(dobs M est et ϑ_{obs} est GP)	ALORS	(ϑ_{rep} est PN),	OU
SI	(dobs G est et ϑ_{obs} est GN)	ALORS	(ϑ_{rep} est EZ),	OU
SI	(dobs G est et ϑ_{obs} est MN)	ALORS	(ϑ_{rep} est PP),	OU
SI	(dobs G est et ϑ_{obs} est PN)	ALORS	(ϑ_{rep} est PP),	OU
SI	(dobs G est et ϑ_{obs} est EZ)	ALORS	(ϑ_{rep} est PN),	OU
SI	(dobs G est et ϑ_{obs} est PP)	ALORS	(ϑ_{rep} est PN),	OU
SI	(dobs G est et ϑ_{obs} est MP)	ALORS	(ϑ_{rep} est PN),	OU
SI	(dobs G est et ϑ_{obs} est GP)	ALORS	(ϑ_{rep} est EZ).	

Les règles sont résumées dans le tableau suivant :

		ϑ_{obs}						
		GN	MN	PN	EZ	PP	MP	GP
d_{obs}	P	PP	MP	GP	GGN	GN	MN	PN
	M	PP	PP	MP	GN	MN	PN	PN
	G	EZ	PP	PP	PN	PN	PN	EZ

Tableau 5.4 la base des règles de l'angle d'évitement

5.7.3. Orgnigramme



Organigramme 5.1 la méthode proposée

5.7.4. Simulation

les résultats de simulation sont obtenues à l'aide d'un programme sur un micro-ordinateur édité sous MATLAB 5.3 ou 6.

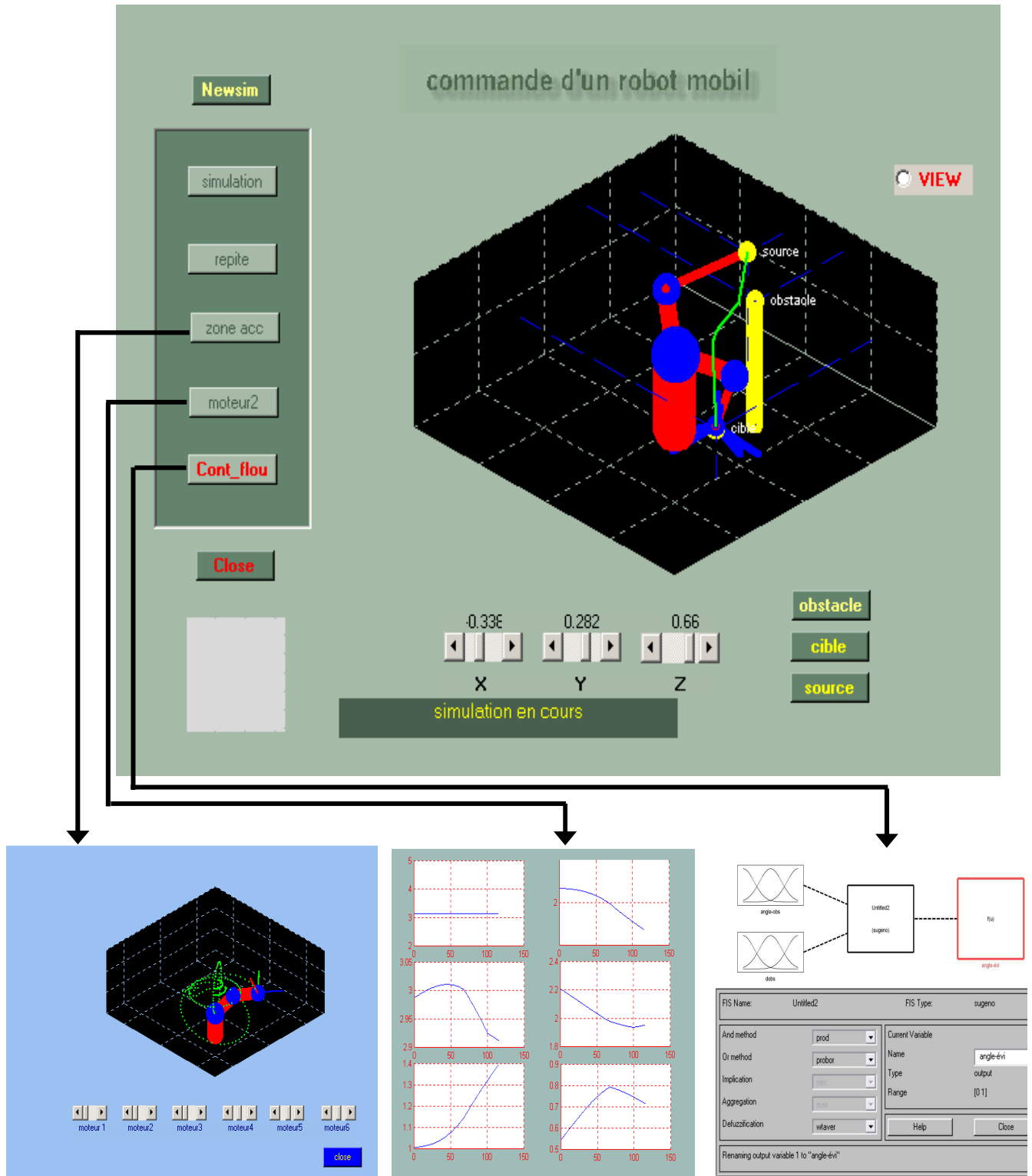


Figure 5.12- Menu générale du programme de Simulation

Les Essais (1.2.3.4.5) : représente le déplacement d'un bras manipulateur dans l'espace à partir de la position initiale (source) en arrivent à sa position finale (cible) tout en évitement l'obstacle rencontre.

Les essais (1.3.4) : représentes le déplacement d'un bras avec : $\vartheta_{\text{obstacle}} \in [\vartheta_{\text{source}} \vartheta_{\text{cible}}]$

Nous avons l'expérience pour illustrer l'importance de calcul de la distance entre la machine et l'obstacle dans la détermination de la trajectoire. Qui on peut dire que la machine ne vient pas près de l'obstacle que si elle entrave le cours de son cible.

Les essais (2.5) : représentes le déplacement d'un bras avec : $\vartheta_{\text{obstacle}} \notin [\vartheta_{\text{source}} \vartheta_{\text{cible}}]$

Ici, nous pouvons voir l'inverses du premières essais où nous mettons un obstacle sur le chemin de la machine afin de voir comment éviter et déplacer à cette obstacle. Et nous pouvons voir plusieurs types d'évitement et déplacement de l'obstacle :

Les 2^{ème} et 4^{ème} essais : déplacement avec la position de l'obstacle entre la position de cible et la source

Le 3^{ème} essai : évitement avec l'obstacle très proche à la source

Le 5^{ème} essai : évitement avec l'obstacle très proche à la cible

Le 1^{ère} essais : déplacement avec l'obstacle très loin à la source et la cible

Les figures (1.2...5) représentent les variations des 6 angles des moteurs ($\vartheta_1, \vartheta_2, \vartheta_3, \vartheta_4, \vartheta_5, \vartheta_6$).

1ère ESSAI

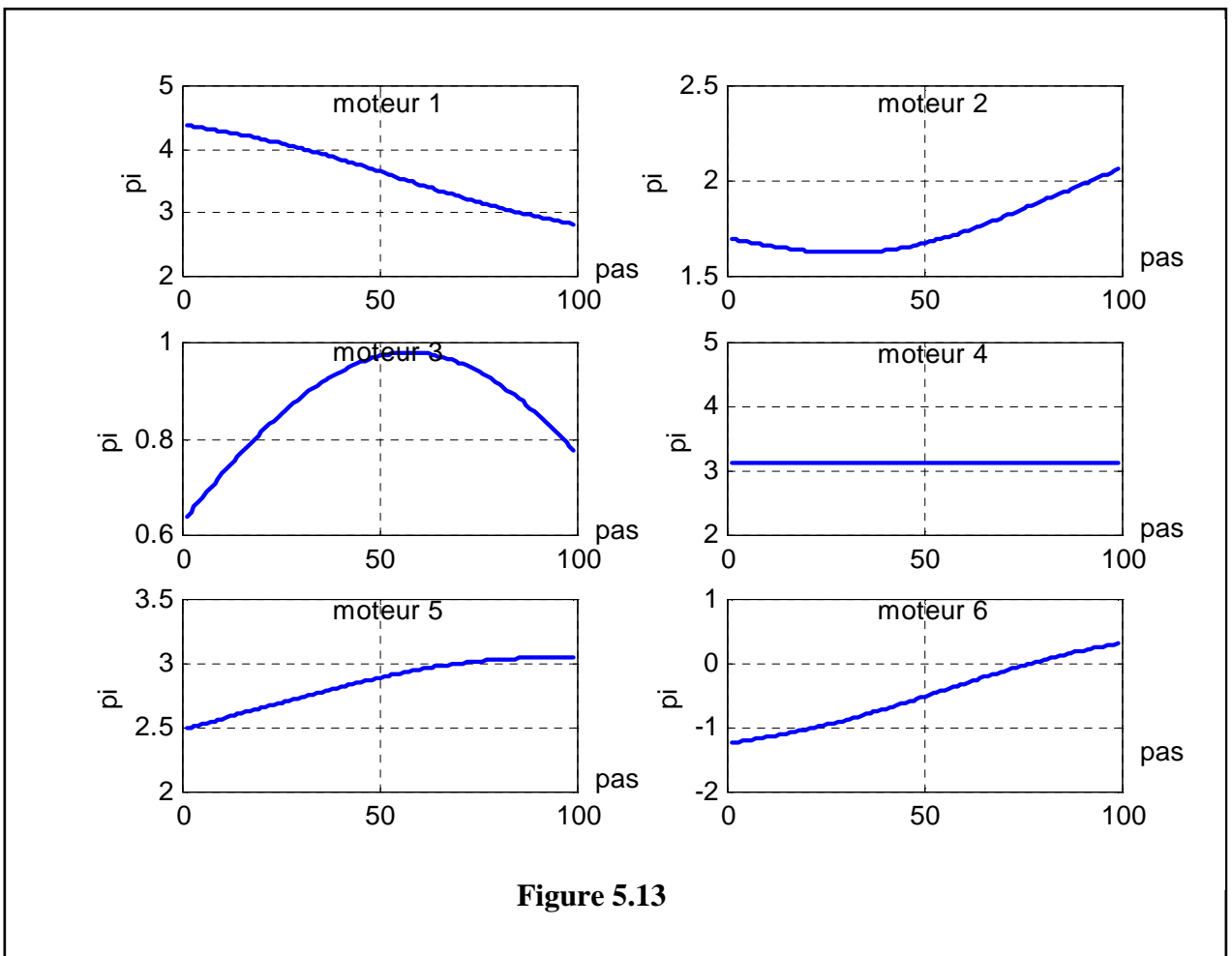
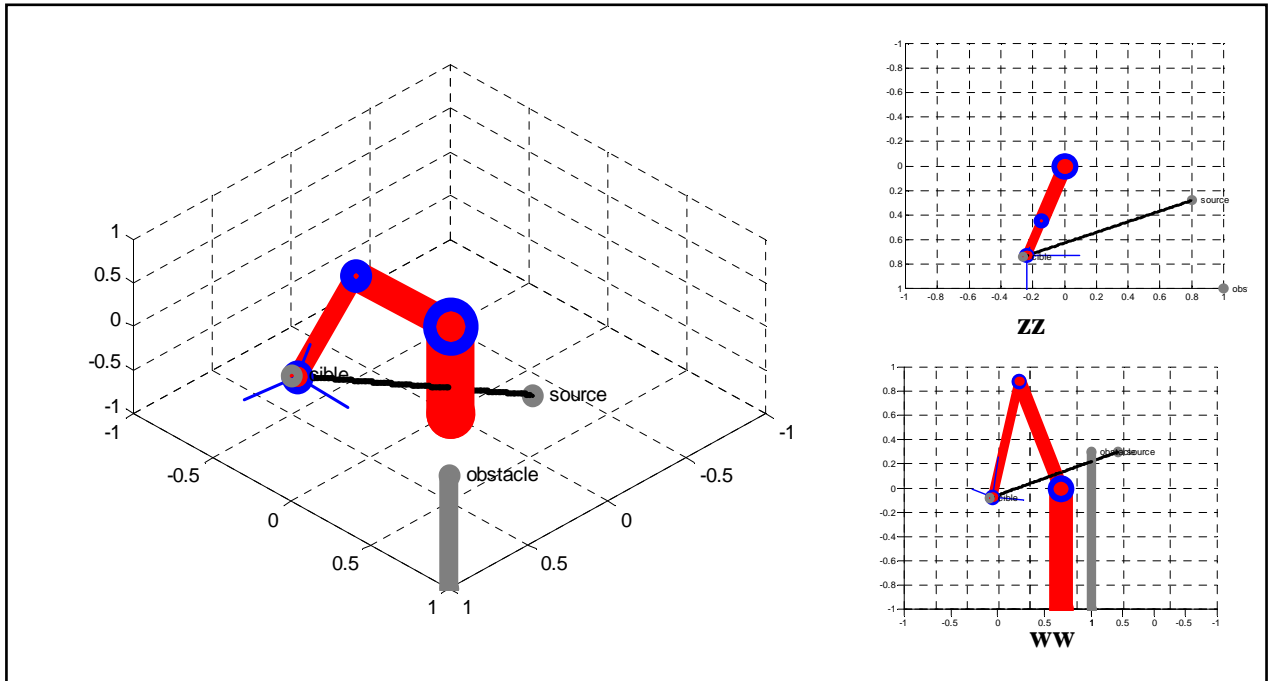


Figure 5.13

2^{ème} ESSAI

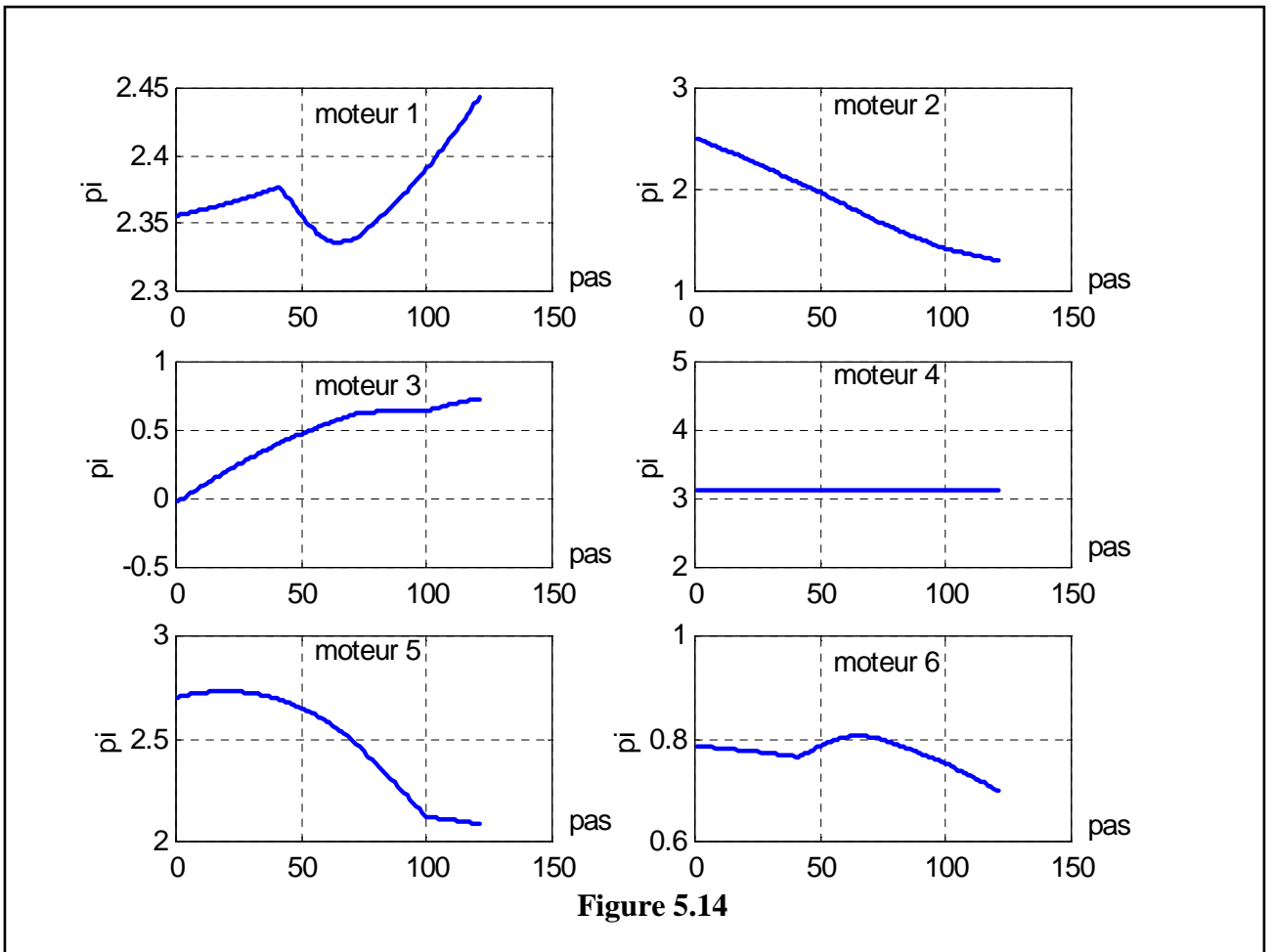
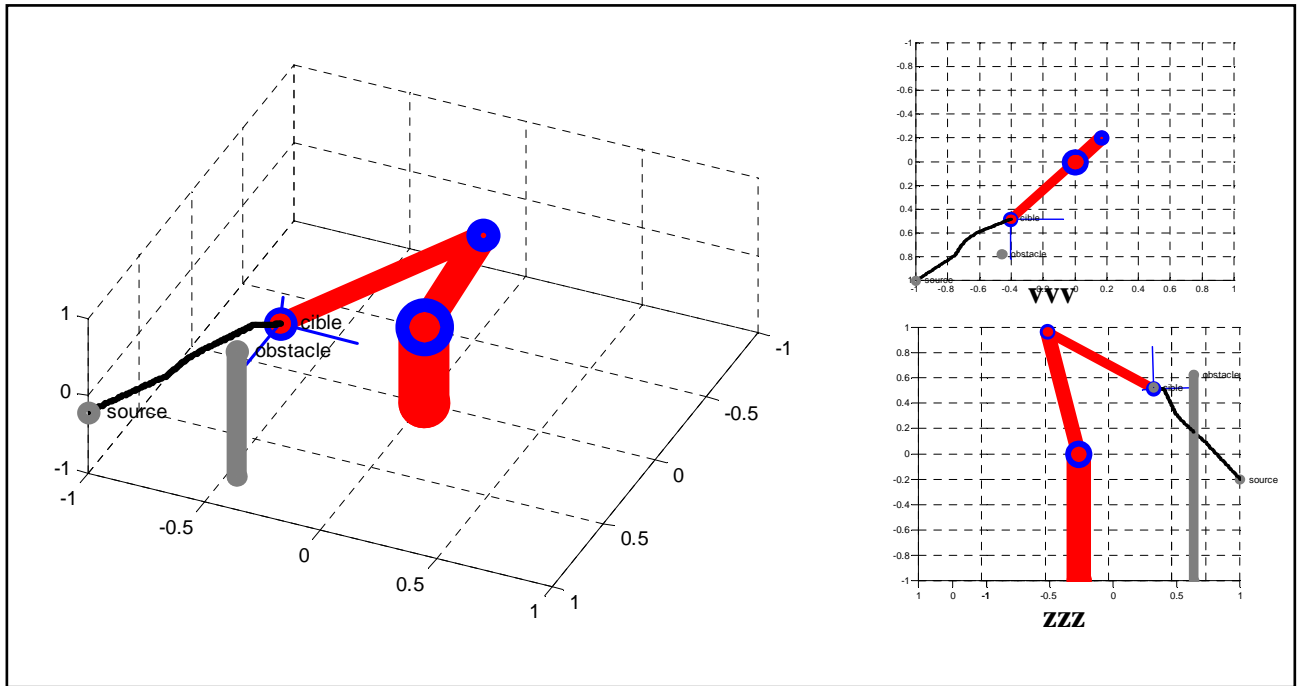


Figure 5.14

3^{ème} ESSAI

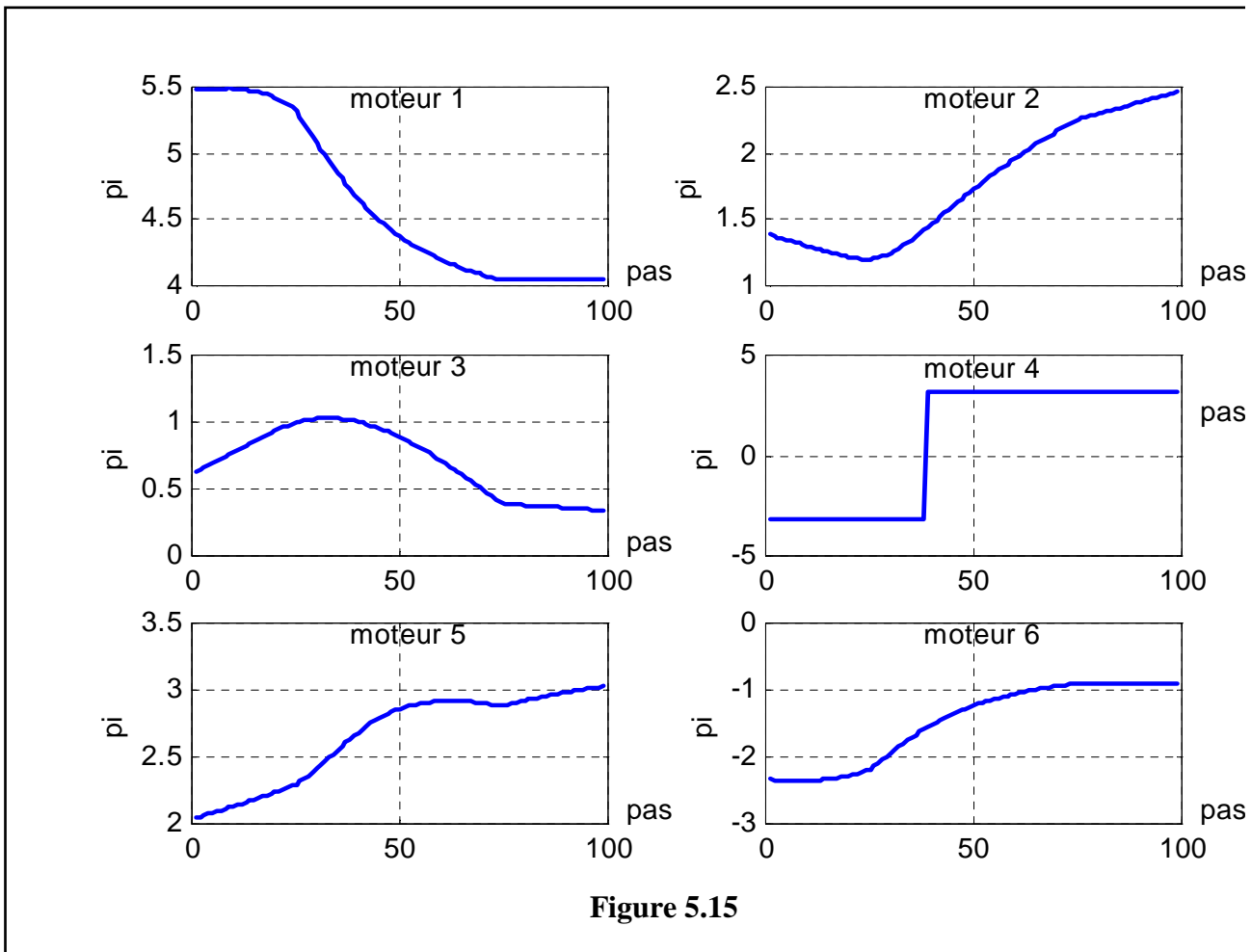
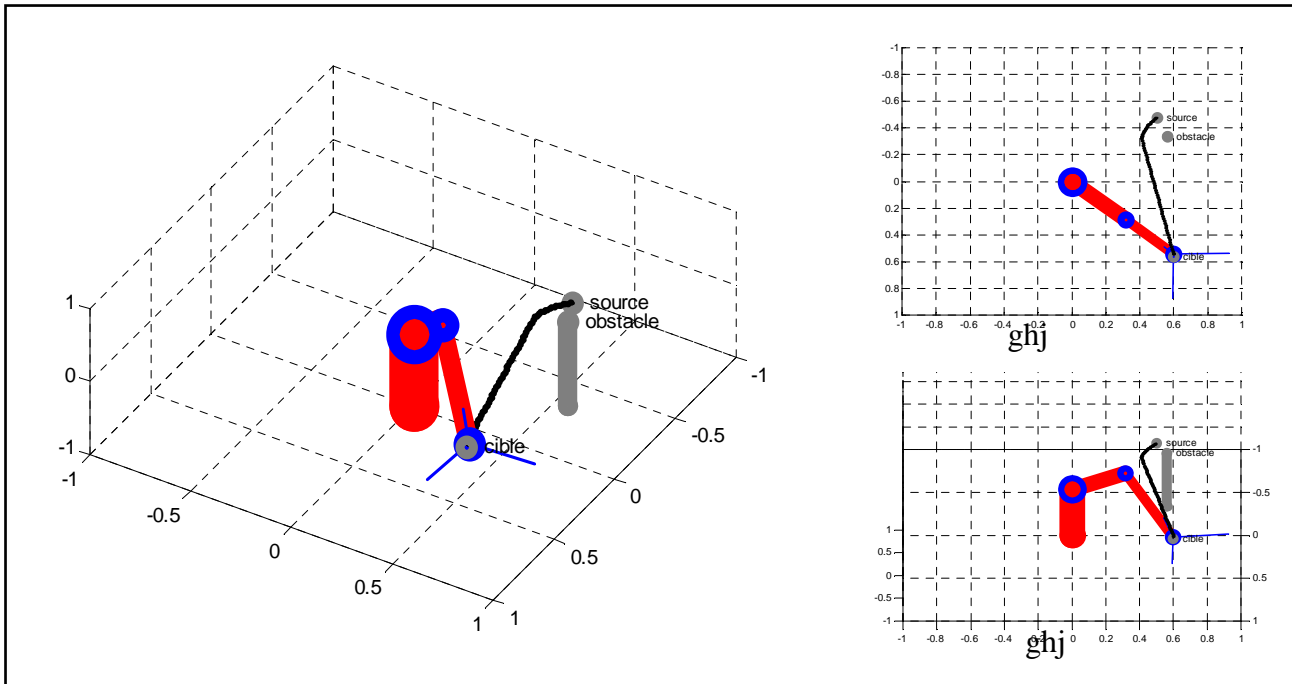


Figure 5.15

4^{ème} ESSAI

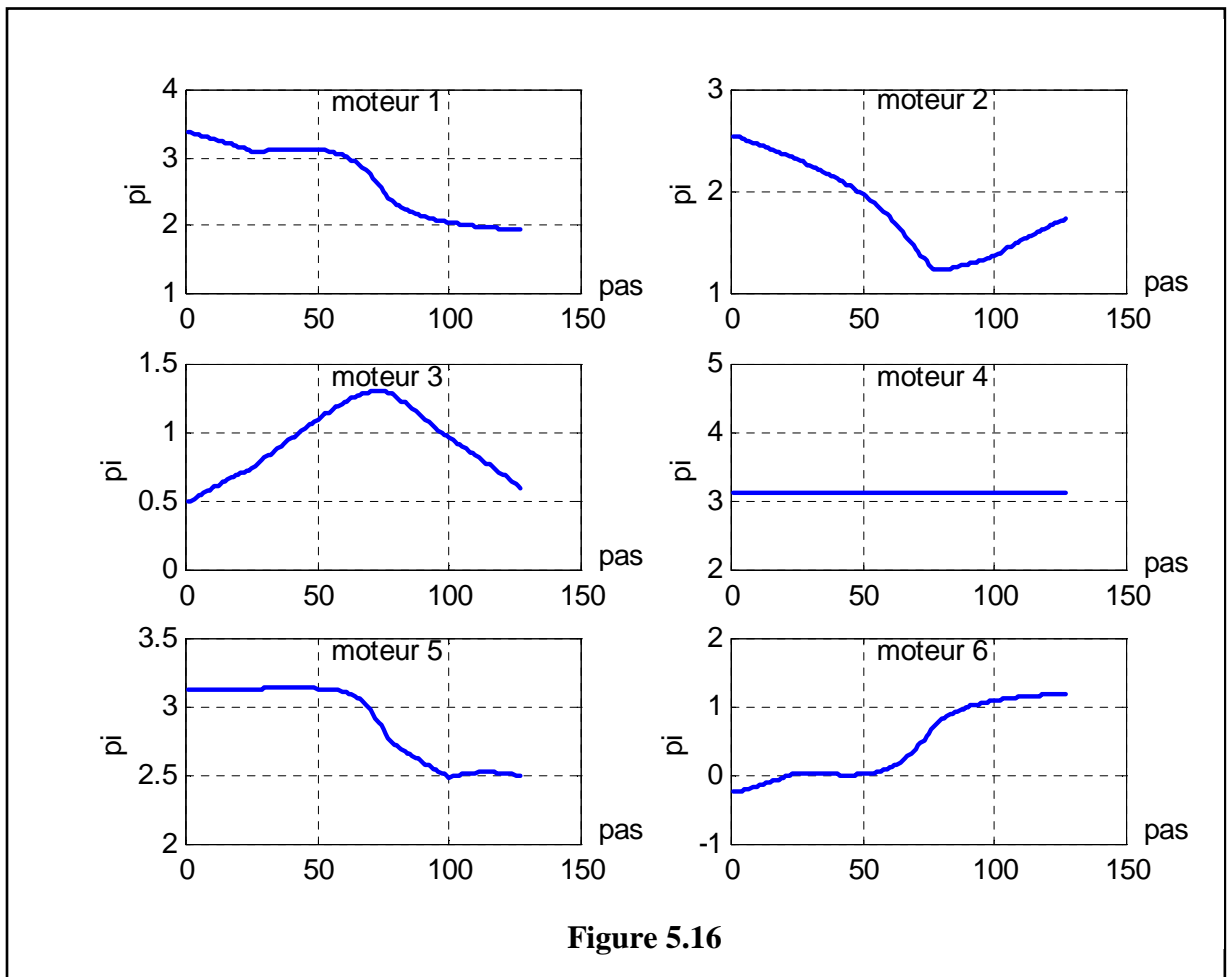
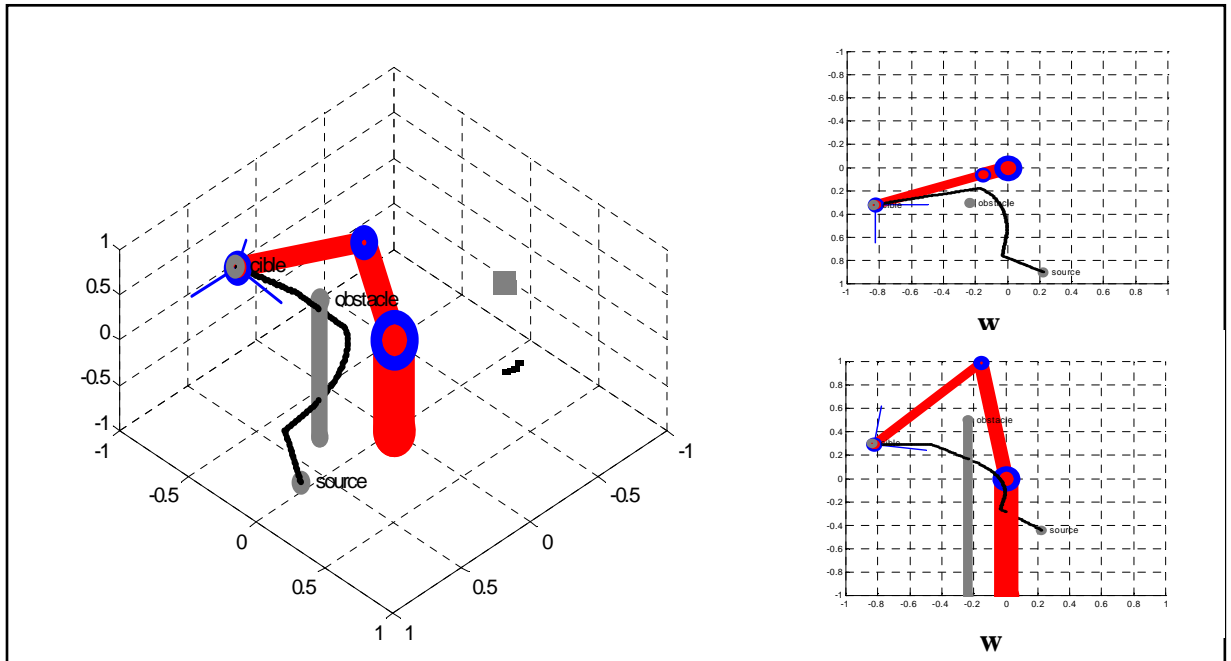


Figure 5.16

5^{ème} ESSAI

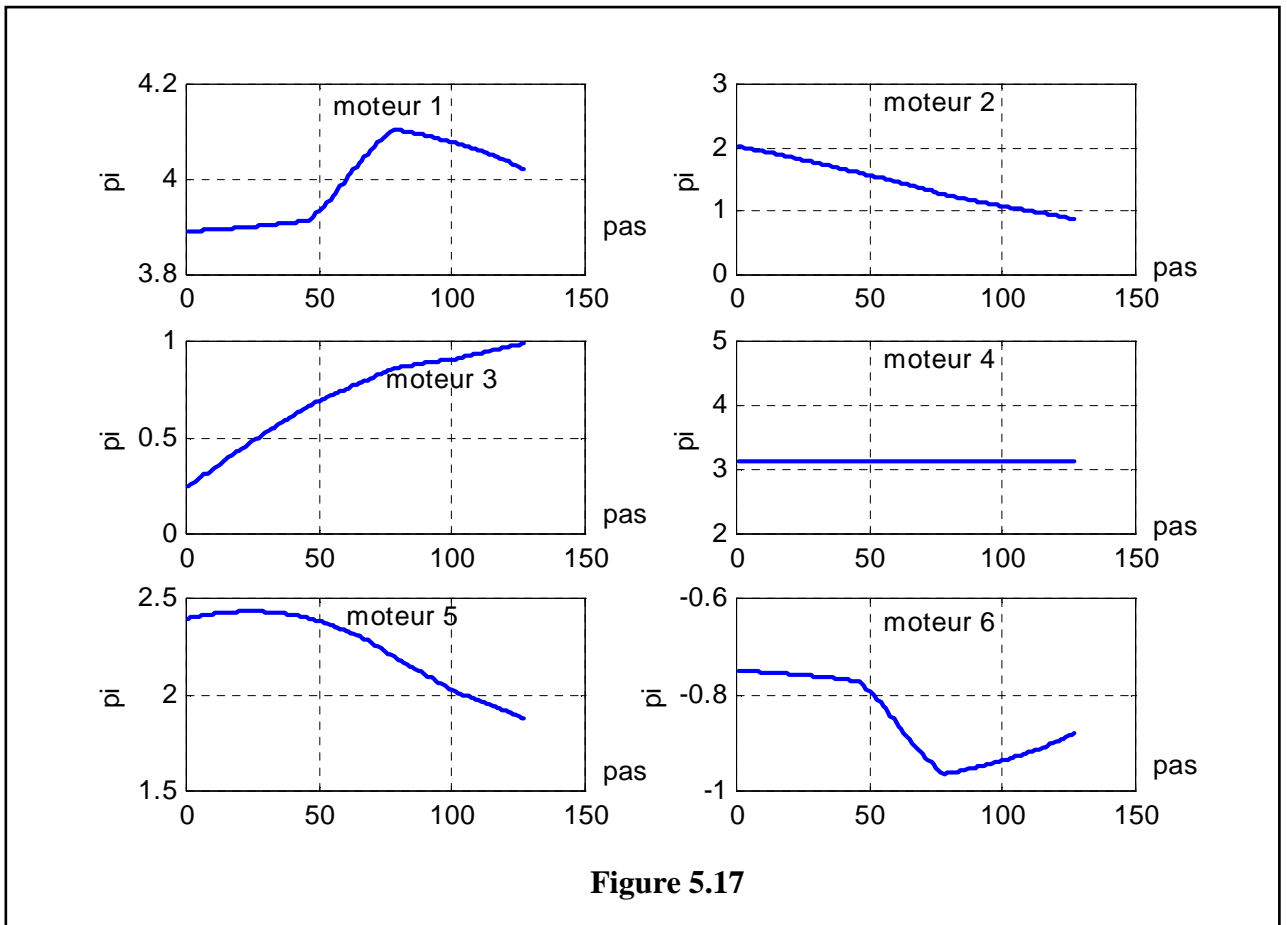
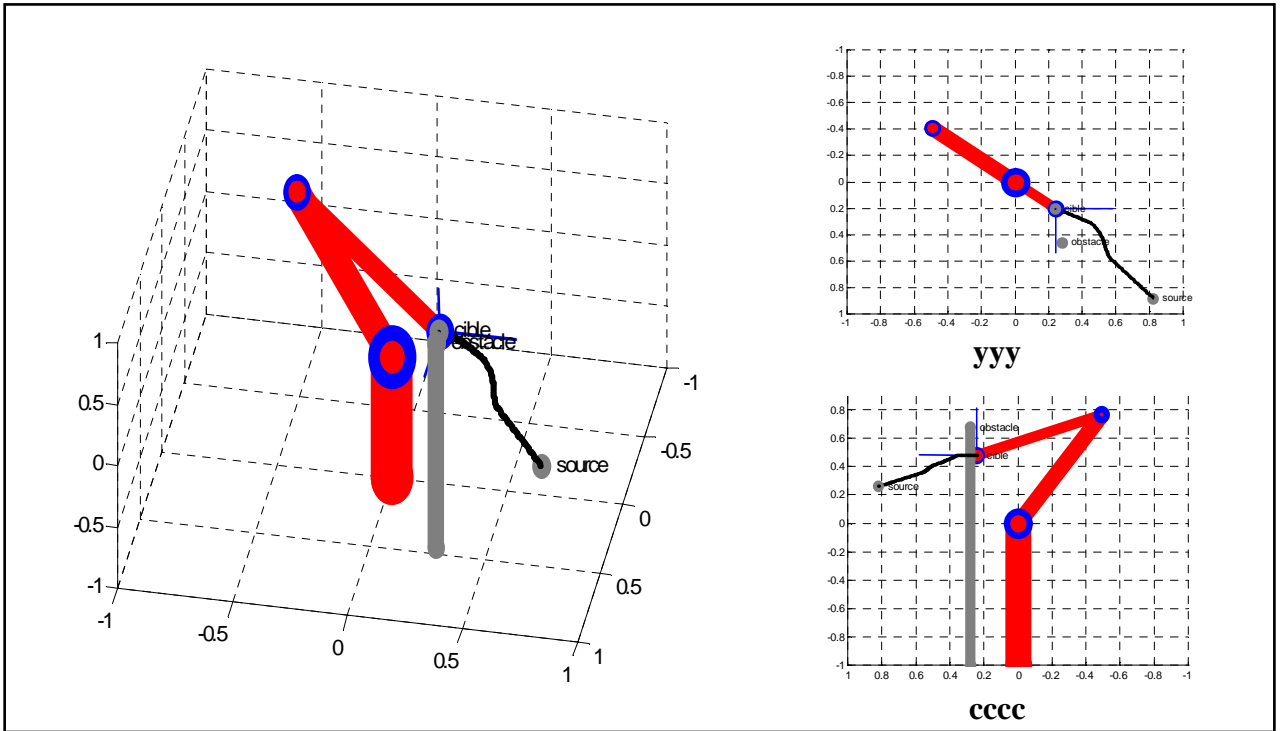


Figure 5.17

5.8. Le Contrôleur Neuro-Flou pour l'évitement d'obstacle

L'objectif est la création d'un contrôleur hybride (ANFIS) simple capable d'évaluer le vecteur d'évitement V_{evi} correspond à la position relative actuelle d'obstacle.

Le contrôleur possède deux entrées et une sortie, les entrées sont l'angle et la distance par rapport à l'obstacle, la sortie est le vecteur d'évitement.

Pour le bras ; l'entrée est le vecteur direction et les sorties sont les deux coordonnées absolues (x_{bra} , y_{bra} , Z_{bra}) et l'angle de bras ϑ_{bra} . Et Pour le Modèle du bras : l'entrée est le vecteur direction et la sortie sont l'entrée du système (l'angle et la distance), et En choisi La même type de contrôleur flou qui appliqué dans la première partie.

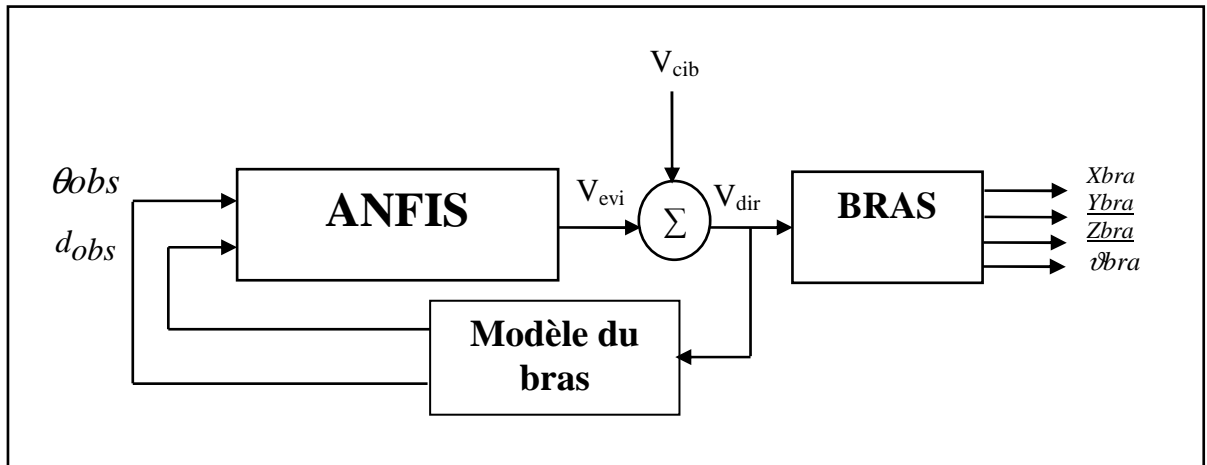


Figure 5.18- L'architecture générale du régulateur

Il prend deux entrées *l'angle* et la *distance*; on suppose que le bras est à l'état initial d'équilibre : l'angle et la vitesse sont nuls ; une impulsion fait traverser le bras vers la droite (angle positif) ou vers la gauche (angle négatif), avec une vérification des déplacements. Il prend en sortie *la vitesse d'évitement* ; la vitesse du basculement va provoquer une réaction plus ou moins énergique du logiciel qui représente le contrôleur. La vitesse sera proportionnelle à la mesure de l'angle et de la distance. Ainsi, un mouvement vers la gauche (négatif) devra provoquer un déplacement du bras vers la gauche (négatif).

- Univers de discours de l'entrée 1 : l'angle $[-0.4, 0.7]$ rad ;
- Univers de discours de l'entrée 2 : la distance $[-0.5, 0.5]$ m;

Chaque entrée est répartie en 7 ou 3 (respectivement) ensembles flous la forme des fonctions d'appartenance. Pour ce régulateur ANFIS il emploie les paramètres suivants :

- Fonctions d'appartenances triangulaires.

- 21 règles.
- 1000 a 1400 Itération d'apprentissage.
- 05 couches.
- 10 neurones dans la première couche cachée.
- 21 neurones dans la deuxième couche cachée.
- 21 neurones dans la troisième couche cachée.
- 21 neurones dans la quatrième couche cachée.
- Un seul neurone dans la cinquième couche de sortie.
- les coefficients d'apprentissage : $\eta_a = 0.2$, $\eta_b = 0.1$, $\eta_c = 0.09$

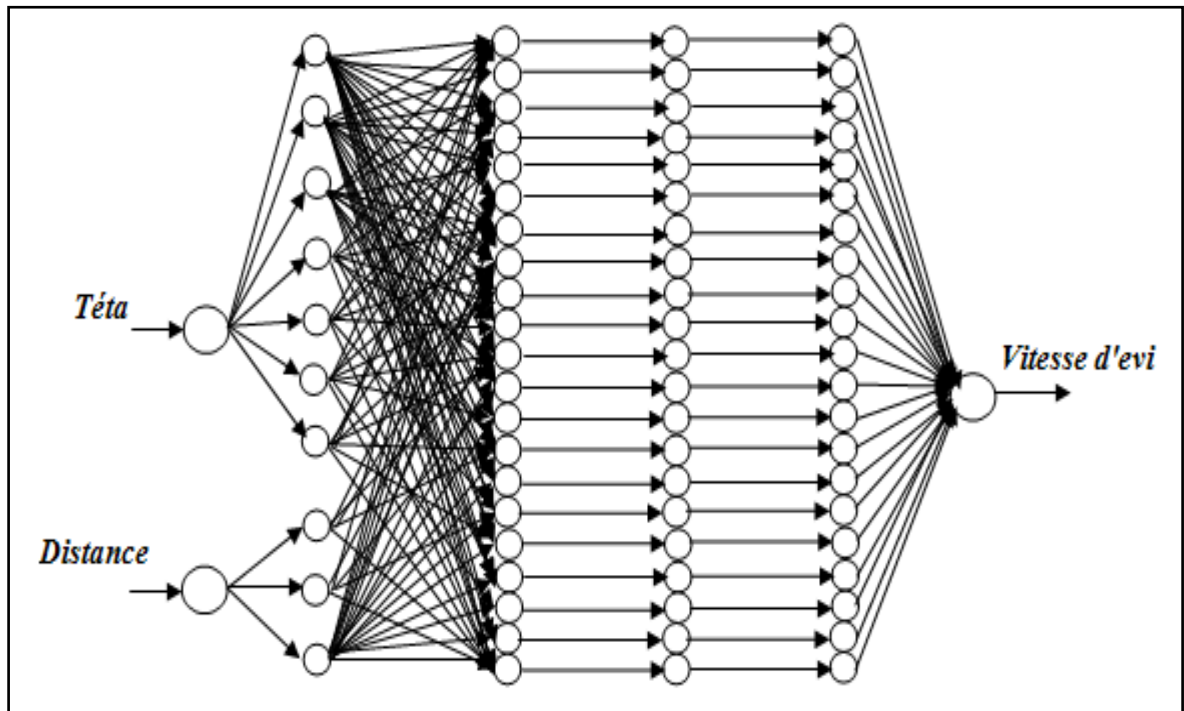


Figure 5.18- L'architecture de l'ANFIS proposé.

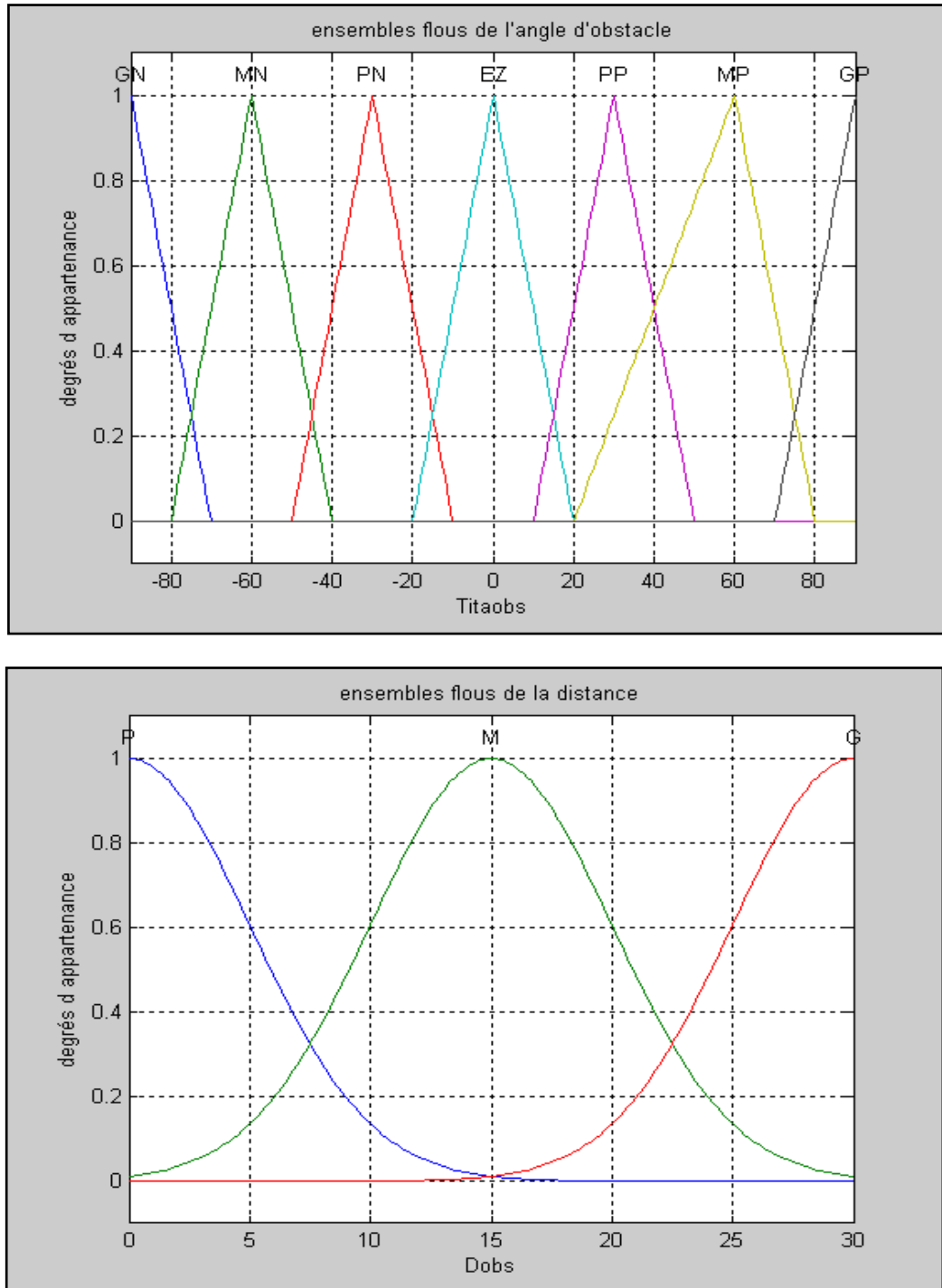


Figure 5.19- les partitions floues initiale des deux entrées « Titaobs et Dobs ».

5.8.1. Phase d'apprentissage

L'apprentissage consiste à un entraînement du réseau :

Etape 1 : activation du réseau

On présente au système (ANFIS) deux entrées l'angle (tétaObs) et la vitesse (disObd), après la fuzzification de la première couche caché l'ANFIS calcule le degré de chaque règle dont la base des règles contient 21 règles plantée dans 21 neurones de la deuxième couche caché

$$O_{2,I} = w_I = \mu_{A_i} (teta) \times \mu_{B_j} (dis) \quad (5.12)$$

Tel que : $i=1$ jusqu' a 7 et $j=1$ jusqu' a 3 donc on a 21 opération de produit qui implique 21 règle .

La troisième couche caché elle à une opération de normalisation ou les sorties de la coche

précédente sont effectue a :

$$\overline{w_I} = \frac{w_I}{\sum_{I=1}^{25} w_I} \quad (5.13)$$

Tant que notre ANFIS est de type Takagi-Ségéno, la quatrième couche caché nécessite le calcul des fonctions :

$$f_I = p_I \times teta + q_I \times dis + r_I \quad (5.14)$$

L'ensemble des paramètres $S2 = \{ p_I, q_I, r_I \}$ est calculée par l'utilisation de moindre carré (la matrice de pseudo inverse).

Pour des valeurs données de S1 [$S1$: est l'ensemble des paramètres $\{ a_i, b_i, c_i \}$], en utilisant le vecteur du deux entrées pour formé l'équation suivant :

$$B = pinv(A \times X) \quad (5.15)$$

Où B contient les éléments de S2; avec:

$$A = [\overline{w_1} \times teta \dots \overline{w_1} \times dis \dots \overline{w_1} \dots \overline{w_{21}}]$$

et X contient le vecteur de sortie Vit-evi.

Où la sortie de cette couche est :

$$O_{4,I} = \overline{w_I} \times f_I \quad (5.16)$$

La sortie de réseau se développe tel que : $y = \sum \overline{w_I} \times f_I$

Avec: $y = O_5 \quad (5.17)$

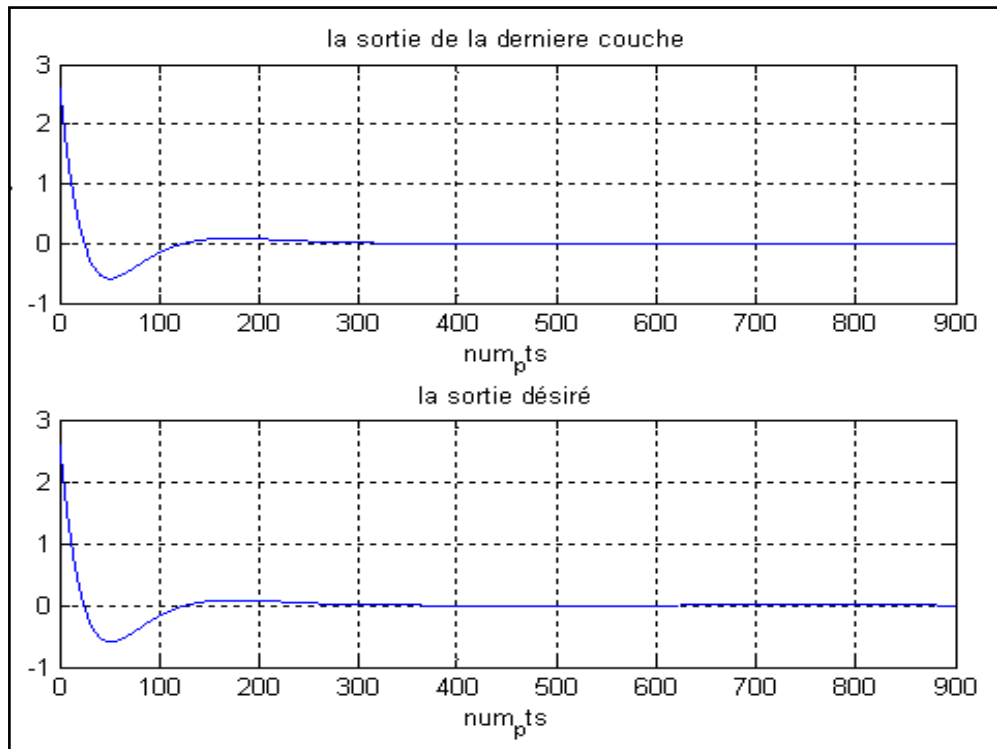


Figure 5.20- sortie du système et la sortie désiré

- Le nombre de paramètre optimiser par l’algorithme de rétro propagation est 30.
- Le nombre de paramètre optimiser par le moindre carrée est 75.

Etape 2 : calcule d’erreur

Comparer la sortie de système à la sortie réel connue (la sortie désiré « 900 valeur ») alors :

$$e = y - Vevi \tag{5.18}$$

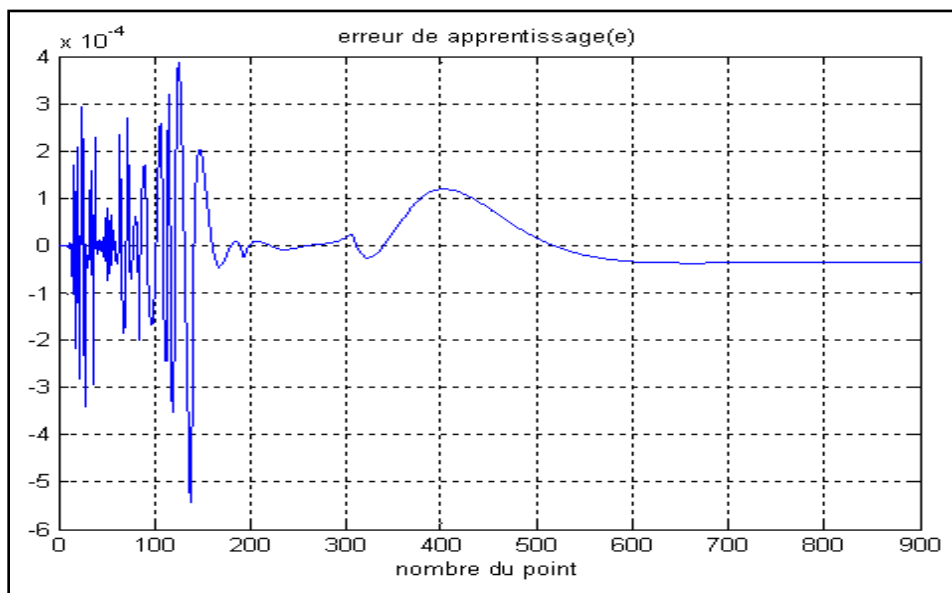


Figure 5.21- l’erreur d’apprentissage (e) après 1000

Les calculs de l'erreur d'apprentissage conduites a calculons l'erreur quadratique moyenne tel que :

$$EQM = \frac{\sum e^2}{num_pts} \quad (5.19)$$

Num_pts est le nombre des points.

L'opération de changement des paramètres initiaux de S1 est basée sur le calcul de l'erreur e L'adaptation (ajustement, mise à jour) de S1, se fait par la méthode du gradient basée essentiellement sur la formule itérative suivante :

$$a_i(n+1) = a_i(n) + del_a_i(n) \quad (5.20)$$

$$b_i(n+1) = b_i(n) + del_b_i(n) \quad (5.21)$$

$$c_i(n+1) = c_i(n) + del_c_i(n) \quad (5.22)$$

Où n est le numéro de l'itération

Avec:

$$del_a_i = -\eta \times delta_a_i \quad (5.23)$$

η : Le pas d'apprentissage.

Dans se cas d'entraînement il est voulu de maître trois valeurs de η attendu que chaque w_i englobe un élément de S1.

Et comme ça l'entraînement du réseau sa marche jusqu'à la condition sur l'erreur (EQM) ou sur le nombre d'itération est atteinte, ce qui donne l'ajustement des paramètres de S1 (les nouveaux paramètres des ensembles flous).

Le chois de facteur η joue un rôle très sérieux pendant l'opération d'apprentissage, leur chois peut converge ou diverge les résultats vers des résultats distant ou touchants a des résultats désirés.

L'opération de changement des paramètres de S1 à 1000 itérations est faite pour la minimisation de la fonction coût (erreur).

Par exemple:

Pour les valeurs $\eta_a=0.02$, $\eta_b=0.3$ et $\eta_c=0.05$, l'évaluation de l'erreur quadratique moyenne est présenté sur la figure suivante « après 1000 itération »:

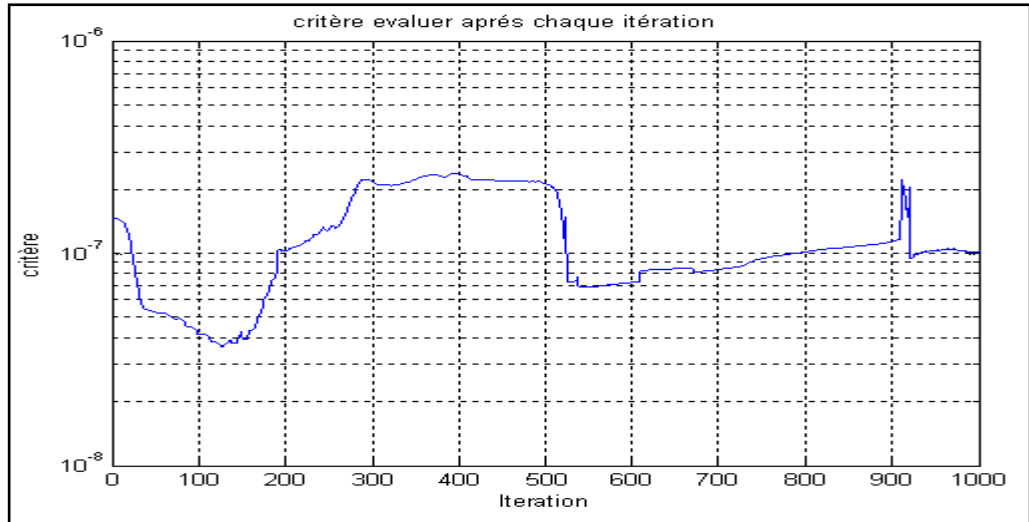


Figure 5.22- critère d'évaluation (EQM) après 1000 itérations.

Pour les valeurs $\eta_a=0.2$, $\eta_b=0.1$ et $\eta_c=0.09$:

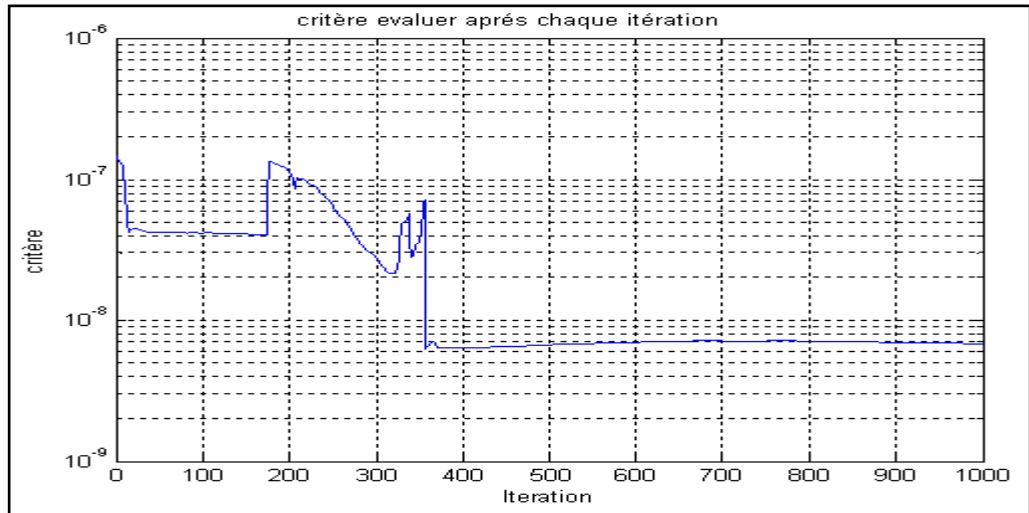


Figure 5.23- critère d'évaluation (EQM) après 1000 itérations

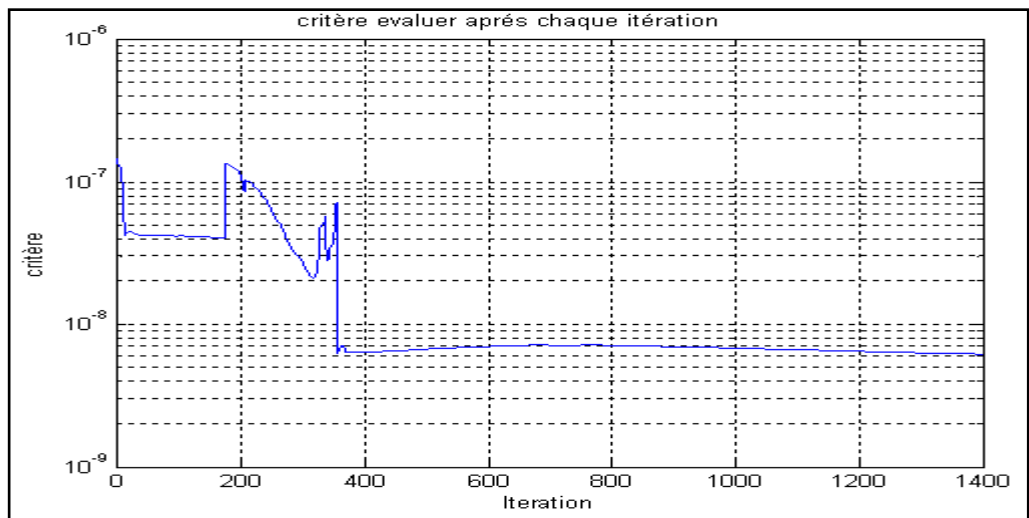


Figure 5.24- critère d'évaluation (EQM) après 1400 itérations.

5.8.2. phase du teste

La phase de teste comme indique le nom consiste à un teste du réseau, c'est-à-dire la validation des résultat obtenue a partir du phase d'apprentissage (vérification du système).

On présente au système (ANFIS) 100 valeur de l'angle (téta) et la distance (dis) déferente a les 900 valeur qui utilisant dans la phase d'apprentissage et on garde les paramètres ajusté de S1 (les paramètres a_i , b_i et c_i du dernière itération) pour leur utilisation dans l'activation du réseau actuelle.

En utilisant la même méthode précédente d'activation du réseau qu'elle consiste les étapes suivantes :

- La fuzzification des entrées au niveau de la première couche caché.
- Calcule les poids des règles (21 règles).
- La normalisation des poids au niveau de la troisième couche.
- calcule des fonctions linéaires (5.14)

Tel que l'ensemble des paramètres $S2 = \{ p_i, q_i, r_i \}$ (déferent a celle présenté dans la phase précédente) qui calculée aussi par l'utilisation de la matrice de pseudo inverse.

- Calcule la sortie dont l'utilisation de l'équation (5.17)

Après l'activation du système on va calculer l'erreur E et l'erreur EQM concernant la phase d'utilisation (teste) pour vérifier l'efficacité du système d'ANFIS.

Les figures suivantes présenté l'erreur E pour les déferents paramètres de S1 qui concernant les itérations (1000, 1400). (L'erreur entre les 100 valeurs de l'angle d'évitement désirée et l'angle d'évitement)

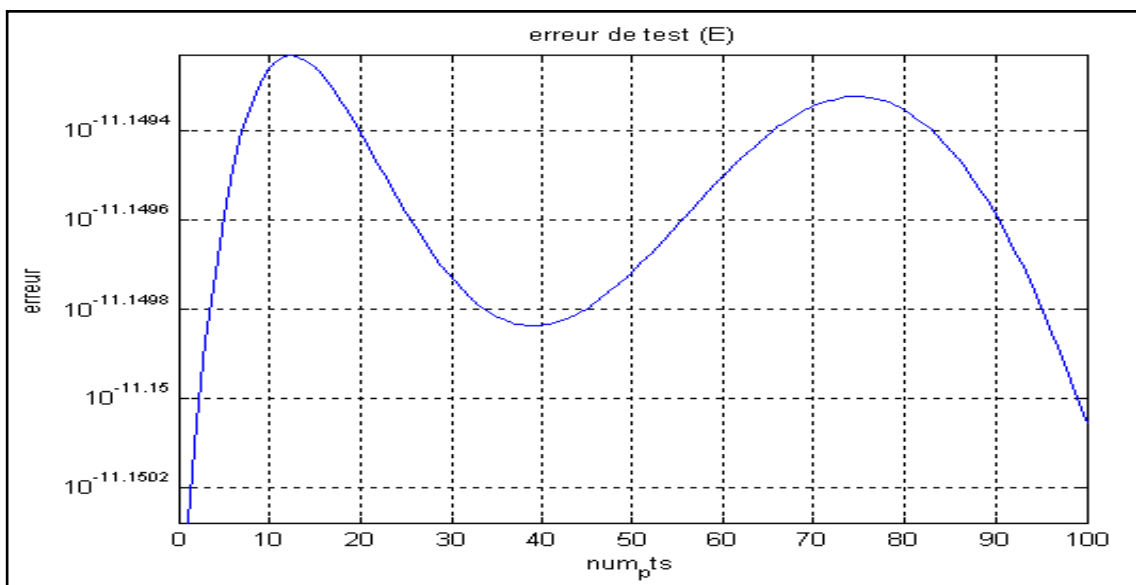


Figure V.14 : l'erreur de test qu'utilise les paramètres ajusté après 1200 itération.

5.8.3. Interprétation des Résultats

Dans ce qui passe nous avons étudié un système non linéaire qu'il est contrôlé par un contrôleur de type ANFIS. Les figures précédente montre les résultats de simulation de ce système pour les deux phases (phase d'apprentissage et phase de teste).

A partir des résultats de simulation de la phase d'apprentissage on trouve (ce qui concerne les Figures 5-(23-24)) que l'erreur (EQM) d'apprentissage diminue avec l'augmentation du nombre d'itération, tel que :

Dans la figure (5-23) on utilise 1000 itérations et on trouve après ces itérations que le SSE (de la dernière itération) est $EQM = 6.796 \times 10^{-09}$, dans la figure (5-24) on utilise 1400 itérations et on obtient $EQM = 6.1449 \times 10^{-09}$, ce qui montre que l'augmentation du nombre d'itération à une effet sur la diminution de l'erreur d'apprentissage EQM jusqu' a l'obtention de l'erreur désiré.

D'autre part nous remarquons aussi que le choix du coefficient (pas) d'apprentissage η a une influence sur l'évaluation de l'erreur quadratique moyenne (sur l'entraînement du réseau) comme il indique dans les figures 5.(22-23), tel que dans la figure 5.22 on a choisie $\eta_a=0.02$, $\eta_b=0.3$ et $\eta_c=0.05$ et on trouve que l'erreur ($EQM = 1.0268 \times 10^{-07}$) diverge ver a un valeur distant a celle désiré tandis que dans la figure 5-23 ou $\eta_a=0.2$, $\eta_b=0.1$ et $\eta_c=0.09$ l'erreur ($EQM = 6.796 \times 10^{-09}$) Converge a EQM désiré (10^{-09}), ce qui nous conduit a dire que le coefficient d'apprentissage μ joue un rôle important puisqu'il règle la vitesse avec laquelle se fait la descente du gradient ; trop petit, il ne permet d'atteindre une valeur suffisamment faible de l'erreur qu'au bout d'un très grand nombre d'itérations, et même fait encourir le risque d'un minimum local (insatisfaisant) duquel il est impossible de sortir, trop grand il peut conduire à s'éloigner du minimum d'erreur recherché et donc peut empêcher une solution acceptable d'être atteinte.

Ce qui concernant les résultats du phase de teste (mode d'utilisation) :

Après l'utilisation des paramètres ajusté résulte a partir de la phase d'apprentissage (les nouveaux paramètres des ensembles flous) nous constatons que l'erreur obtenue (entre la sortie résulte et la sortie désiré) dans cette phase est très petit a celle de la phase d'apprentissage comme illustre les figures 5-(21-24) tel que l'erreur quadratique moyenne de réseau sera $EQM = 5.0217 \times 10^{-23}$, ce qui permet de dire que le réseau est déjà capable d'effectuer sa tâche correctement.

5.9. Conclusion

Dans ces chapitre on donne la modélisation des robots manipulateurs et ces différents types (modélisation géométrique, cinématique et dynamique), ainsi la description géométrique du bras à étudier robot stäubli RX-90 et leur modélisation géométrique (directe et inverse)

Et nous verrons dans ce chapitre la navigation d'un bras manipulateur avec l'application de la logique flou et l'adaptation neuronal (Système hybride ANFIS) pour l'optimisation la trajectoire et l'évitement d'obstacle.

Et à la fin dans notre étude, on conclut que:

- 1- l'erreur est diminuée avec l'augmentation des nombres d'itération et des nombres de base de donnée.
- 2- Le choix du pas d'apprentissage est très important, sa valeur est généralement choisie expérimentalement.
- 3- l'ajustement des paramètres des ensembles flous permet de produire une bonne raisonner de réseau.
- 4- les performances du système ANFIS sont plus précises, cause de l'adaptation entre la capacité d'approximation des réseaux de neurones (surtout lorsque le nombre des entrées est élevé) et la rapidité de la vitesse d'exécution des algorithmes flous.