

République Algérienne Démocratique et Populaire
Ministère de l'Enseignement Supérieur et de la Recherche Scientifique
Université A. MIRA-BEJAIA



Faculté de Technologie

Département Génie Electrique

MEMOIRE DE FIN D'ETUDES

Pour l'obtention du diplôme de Master

Filière : Electrotechnique

Option : Automatismes Industriels

Thème

*Commande neuro-floue adaptative
d'un robot Delta*

Présenté par :

M^r BENISSAD Rassim

Encadre par :

M^r MELAHI Ahmed

Membre de jury:

M^{me} MEDJAHED Sakina

M^r AZZI Abdelmalek

Année Universitaire : 2022/2023

Remerciements

*Je tiens remercier ALLAH qui m'a aidé et qui m'a donné la
patience et le courage durant ces années d'études.*

*Je commence par exprimer ma profonde reconnaissance et mes vifs
remerciements à mon promoteur Mr. MELAHI AHMED pour ses
orientations, encouragements, ses conseils et sa disponibilité. Je
remercie également tous les membres du jury d'avoir accepté
d'assister à la présentation de ce travail.*

*Enfin, j'adresse mes remerciements à toute ma famille sans oublier
mes amis et mes collègues.*

Dédicaces

Je dédie ce Modeste travail à

*A L'homme, mon précieux offre du dieu, qui doit ma vie, ma
réussite et tout mon respect : mon père ;*

*A la femme qui n'a jamais dit non à mes exigences et qui n'a
épargné aucun effort pour me rendre heureux : ma mère ;*

A mon frère Rami ;

Un merci pour ma famille ;

*A mes amis de la promotion, tant de souvenir gardés tout au long
de mon parcours universitaire ;*

A tous les enseignants qui m'ont aidé de près ou de loin ;

Merci à tous

Liste des figures

Figure I.1: Classification des températures d'une pièce en deux sous-ensembles	2
Figure I.2: Différentes formes des fonctions d'appartenance [6]	4
Figure I.3: Le principe d'un système neuro-flou [10]	5
Figure I.4: Première architecture Neuro-floue	6
Figure I.5: Forme d'un neurone	7
Figure I.6: Forme d'opérateur flou ET	7
Figure I.7: Deuxième architecture des réseaux Neuro-Flou Réalisation en série	8
Figure I.8: Réseau de neurones fonctionnant en aval d'un système flou	9
Figure I.9: Système neuro-flou type Mamdani	10
Figure I.10: Système neuro-flou type Takagi-Sugeno	11
Figure I.11: système neuro-flou du type ANFIS	12
Figure I.12: Architecture ANFIS réalisant le système d'inférence floue de Tsukamoto	13
Figure I.13: Raisonnement flou de Tsukamoto	13
Figure II.14: Raymond Claver, inventeur du robot Delta [14]	15
Figure II.15: Schéma descriptif des constituants du robot Delta	16
Figure II.16: Modèle géométrique	17
Figure II.17: Représentation schématique du robot Delta [16]	18
Figure II.18: Vue du dessus de la base du Delta	18
Figure II.19: Schéma simplifié d'un bras du Delta	19
Figure II.20: Sphère de centre P et de rayon l_2 passant par C_1	20
Figure II.21: Modèle géométrique direct du robot Delta	20
Figure II.22: Modèle géométrique inverse du robot Delta	22
Figure II.23: Modèle cinématique du robot Delta	24
Figure III.24: Projection de la coordonnée curviligne $s(t)$ donnant $x(t)$, $y(t)$, $z(t)$	26
Figure III.25: Schéma Simulink de la commande neuro-floue et un moteur	27

Figure III.26:bloc d'un moteur.	27
Figure III.27:Bloc de génération de trajectoire dans le plan (X, Y, Z).	27
Figure III.28:Bloc de génération de la trajectoire de consigne.	28
Figure III.29:Bloc du modèle géométrique direct et inverse.	28
Figure III.30:Schéma de simulation.	29
Figure III.31:Evolution temporelle des coordonnées (X, Y, Z) et des coordonnées (alpha1, alpha2, alpha3) de poursuite de la trajectoire.	31
Figure III.32:Trajectoire dans le plan (X, Y, Z).	32
Figure III.33:Evolution temporelle des coordonnées (X, Y, Z) et des coordonnées (alpha1, alpha2, alpha3) de poursuite de la trajectoire.	33
Figure III.34:Trajectoire linéaire dans le plan (X, Y, Z).	34
Figure III.35:Evolution temporelle des coordonnées (X, Y, Z) et des coordonnées (alpha1, alpha2, alpha3) de poursuite de la trajectoire.	35
Figure III.36:Trajectoire circulaire dans le plan (X, Y, Z).	36
Figure III.37:Les blocs d'apprentissage	37
Figure III.38:Schéma de simulation adaptative.	38
Figure III.39:Evolution temporelle des coordonnées (X, Y, Z) et des coordonnées (alpha1, alpha2, alpha3) de poursuite de la trajectoire.	39
Figure III.40:Trajectoire circulaire dans le plan (X, Y, Z).	40

Liste des tableaux

Tableau I.1: Implication floue [8].....	4
---	---

Table des matières

Introduction générale.....	1
Chapitre I : La logique floue et Neuro-floue.....	2
Introduction	2
I.1 La logique floue	2
I.1.1 Définition de la logique Floue	2
I.1.2 Logique classique et logique floue.....	2
I.1.3 Historique de la logique floue	3
I.1.4 Eléments de base de la logique floue	3
I.1.4.1. Variables linguistiques	3
I.1.4.2. Sous-ensembles.....	3
I.1.4.3 Operateurs de la logique floue	4
I.1.4.4 Règles floues.....	4
I.2 Neuro-flou.....	5
I.2.1 La combinaison des réseaux neuronaux et de la logique floue	5
I.2.2 Définition du système neuro-flou	5
I.2.3 Structure d'un réseau neuro-flou	5
I.2.4 Architectures neuro-floues	6
I.2.4.1 Première architecture	6
I.2.4.2 Deuxième architecture	8
I.2.4.3 Troisième architecture.....	9
I.2.5 Systèmes Neuro-Flous hybrides intégrés	9
I.2.5.1 Systèmes Neuro-Flous intégrés type Mamdani.....	9
I.2.5.2 Systèmes neuro-flous intégrés type Takagi-Sugeno.....	11
I.2.5.3 Système ANFIS (Adaptative-Network-based Fuzzy Inference System)	12
Conclusion.....	14
Chapitre II : Le robot Delta	15
Introduction	15
II.1 Historique du robot delta	15
II.2 Caractéristiques du Robot Delta	16
II.3 Constituants du robot Delta	16

II.4 Le modèle géométrique	17
II.4.1 Modèle géométrique Direct	20
II.4.2 Modèle géométrique inverse.....	21
II.5 Modèle cinématique	23
II.5.1 Modèle cinématique direct	25
II.5.2 Modèle cinématique inverse	25
Conclusion.....	25
Chapitre III : Application de la commande neuro-floue au robot Delta.....	26
Introduction	26
III.1 Génération de trajectoires.....	26
III.2 Structure du régulateur sous Matlab.....	26
III.3 Simulation sous l'environnement Matlab/Simulink.....	28
III.4 Les résultats de simulation	30
III.4.1 Résultats de simulation pour la première trajectoire	30
III.4.2 Résultats de simulation pour la deuxième trajectoire.....	33
III.4.3 Résultats de simulation pour la troisième trajectoire	35
III.5 Application de la commande neuro-floue adaptative au robot Delta	36
III.5.1 Les blocs d'apprentissage.....	36
III.5.2 Schéma de simulation adaptative	38
III.5.3 Résultats de simulation adaptative pour la troisième trajectoire	39
III.6 Commentaire	40
Conclusion.....	40
Conclusion générale	41
BIBLIOGRAPHIE	43

Introduction générale

Introduction générale

La robotique est un domaine en pleine expansion, offrant des opportunités infinies pour l'automatisation des tâches dans de nombreux secteurs. Les robots Delta, également connus sous le nom de robots parallèles, sont largement utilisés dans l'industrie pour leurs capacités de manipulation précises et rapides. Cependant, le contrôle précis de ces robots Delta représente un défi majeur en raison de leur architecture complexe et de la présence de non-linéarités dans leur modèle dynamique.

Dans le cadre de ce mémoire de fin d'études, nous nous concentrons sur l'application de la commande neuro-floue adaptative pour améliorer le contrôle d'un robot Delta. La commande neuro-floue adaptative est une approche novatrice qui combine les concepts de la logique floue et des réseaux de neurones artificiels pour optimiser le contrôle des systèmes dynamiques.

Dans le chapitre 1, nous introduisons les concepts fondamentaux de la logique floue et de la neuro-floue. La logique floue permet de prendre en compte l'incertitude et les imprécisions en utilisant des ensembles flous et des règles linguistiques pour représenter et manipuler l'information. Les réseaux de neurones artificiels, quant à eux, sont des modèles mathématiques inspirés du fonctionnement du cerveau, capables d'apprentissage et d'adaptation.

Le chapitre 2 est consacré à la présentation du robot Delta et de son modèle géométrique et cinématique. Nous décrivons l'architecture du robot Delta, ses caractéristiques clés. Ensuite, nous exposons les équations mathématiques qui décrivent la géométrie et la cinématique du robot, permettant de relier les mouvements des actionneurs aux positions et orientations des effecteurs.

Dans le chapitre 3, nous utilisons le modèle géométrique du robot Delta pour effectuer des simulations à l'aide de MATLAB. L'objectif principal de cette partie est de tracer les trajectoires du robot Delta dans l'espace de travail. En utilisant les équations cinématiques, nous développons des algorithmes pour générer des trajectoires désirées et les suivre à l'aide du contrôleur neuro-flou adaptatif.

Ce mémoire de fin d'études vise à explorer les avantages et les limitations de la commande neuro-floue adaptative pour améliorer le contrôle d'un robot Delta. Nous mettrons en évidence les résultats obtenus grâce à la simulation et analyserons les performances du robot Delta dans différents scénarios.

*Chapitre I : La logique floue et Neuro-
floue*

Introduction

Dans ce chapitre, je veux présenter deux méthodes importantes dans le domaine d'intelligence artificielle. La logique floue et la commande neuro-floue sont deux concepts interconnectés, qui se complètent dans la conception de systèmes adaptatifs et de commande.

La logique floue est une méthode mathématique et informatique qui permet de prendre en compte l'imprécision et l'incertitude, utilisée dans les systèmes de contrôle industriels, les systèmes d'aide à la décision, la robotique.

La commande neuro-floue est une approche qui combine les principes de la logique floue et des réseaux neuronaux pour concevoir des systèmes de commande adaptatifs.

I.1 La logique floue

I.1.1 Définition de la logique Floue

La logique floue est une extension de la logique qui permet de modéliser les imperfections des données et de se rapprocher dans une certaine mesure de la flexibilité de l'humain. Elle intéresse actuellement les chercheurs, les ingénieurs, les industriels, mais plus généralement tous ceux qui voient le besoin de formaliser les méthodes empiriques, de généraliser les schémas naturels de raisonnement, d'automatiser la prise de décision dans leur domaine, et de construire des systèmes artificiels pour effectuer une tâche généralement effectuée par des humains [1].

I.1.2 Logique classique et logique floue

La logique classique et la logique floue sont deux approches distinctes de la modélisation et du raisonnement logique. Dans le cas de la température de la chambre, la logique floue permet de définir des sous-ensembles flous tels que "froid" et "chaud" avec des degrés d'appartenance variés. Par exemple, une température de 14,9 degrés peut avoir un degré élevé d'appartenance à l'ensemble "froid" et un degré plus faible d'appartenance à l'ensemble "chaud". De même, une température de 15,10 degrés peut avoir un degré élevé d'appartenance à l'ensemble "chaud" et un degré plus faible d'appartenance à l'ensemble "froid" [2].

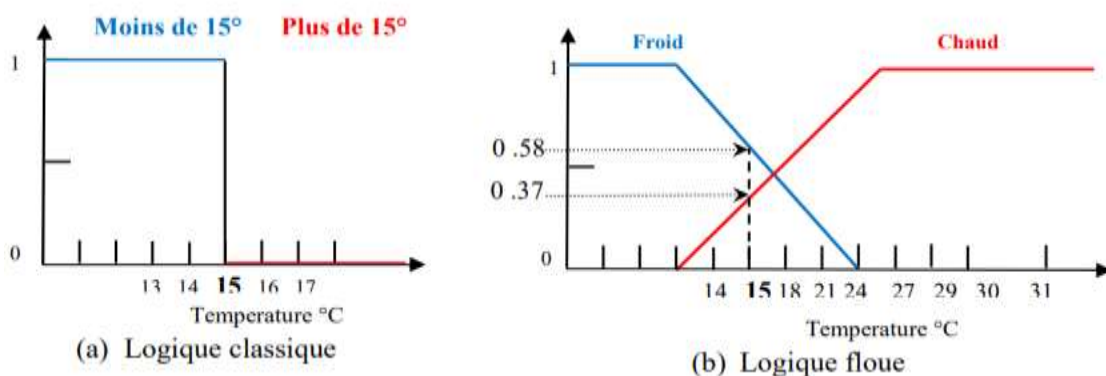


Figure I.1: Classification des températures d'une pièce en deux sous-ensembles

I.1.3 Historique de la logique floue

La logique floue a été développée par le mathématicien et ingénieur japonais Lotfi Zadeh dans les années 1960. Son travail fondateur a été publié en 1965, dans un article intitulé "Fuzzy Sets" ("Ensembles flous") dans lequel il a introduit le concept d'ensembles flous et a jeté les bases de la logique floue. Ainsi en 1975, le professeur Mamdani à Londres applique la théorie inventée par Zadeh sur la conduite d'un moteur à vapeur. La société danoise F.L.Smidth En 1978 réalise la régulation de fours de cimenterie [2]. Après en 1987 ont été utilisés des systèmes flous pour le contrôle d'accélération et le freinage quand la ligne a ouvert par Seiji Yasunobu et Soji Miyamoto de Hitachi. L'utilisation de cette méthode a été généralisée au Japon Grâce au chercheur japonais M. Sugeno, en 1990 [3].

I.1.4 Eléments de base de la logique floue

Dans cette partie, nous expliquerons de façon non exhaustive les éléments fondamentaux de la logique floue.

I.1.4.1. Variables linguistiques

La description imprécise d'une situation, d'un phénomène ou d'une quantité physique peut seulement être faite par des expressions relatives ou flous, à savoir : grand, petit, positif, négatif, nul, chaud, froid, etc. Ces différentes classes d'expressions floues appelées ensembles flous représentent des variables linguistiques. Pour pouvoir traiter numériquement ces variables linguistiques (généralement normalisées sur un intervalle bien défini nommé univers de discours), ils doivent suivre une définition mathématique basée sur des fonctions d'appartenance qui indiquent le degré de vérification de ces variables linguistiques par rapport à différents sous-ensembles flous de la même classe [4].

I.1.4.2. Sous-ensembles

Les sous-ensembles sont des ensembles flous qui représentent des catégories ou des concepts imprécis. Contrairement à la logique classique où un élément appartient soit à un ensemble, soit à un autre, en logique floue, un élément peut appartenir à plusieurs ensembles simultanément, mais avec des degrés d'appartenance différents. Les sous-ensembles en logique floue sont définis à l'aide de fonctions d'appartenance, qui attribuent à chaque élément du domaine un degré d'appartenance à un sous-ensemble spécifique. Ces fonctions d'appartenance décrivent comment les éléments du domaine sont liés au sous-ensemble.

Il existe différentes formes de fonctions d'appartenance utilisées pour représenter les sous-ensembles en logique floue, telles que les fonctions triangulaires, trapézoïdales, gaussiennes, etc. Chaque forme de fonction d'appartenance est adaptée à la représentation d'un type particulier de sous-ensemble ou de distribution [2][5].

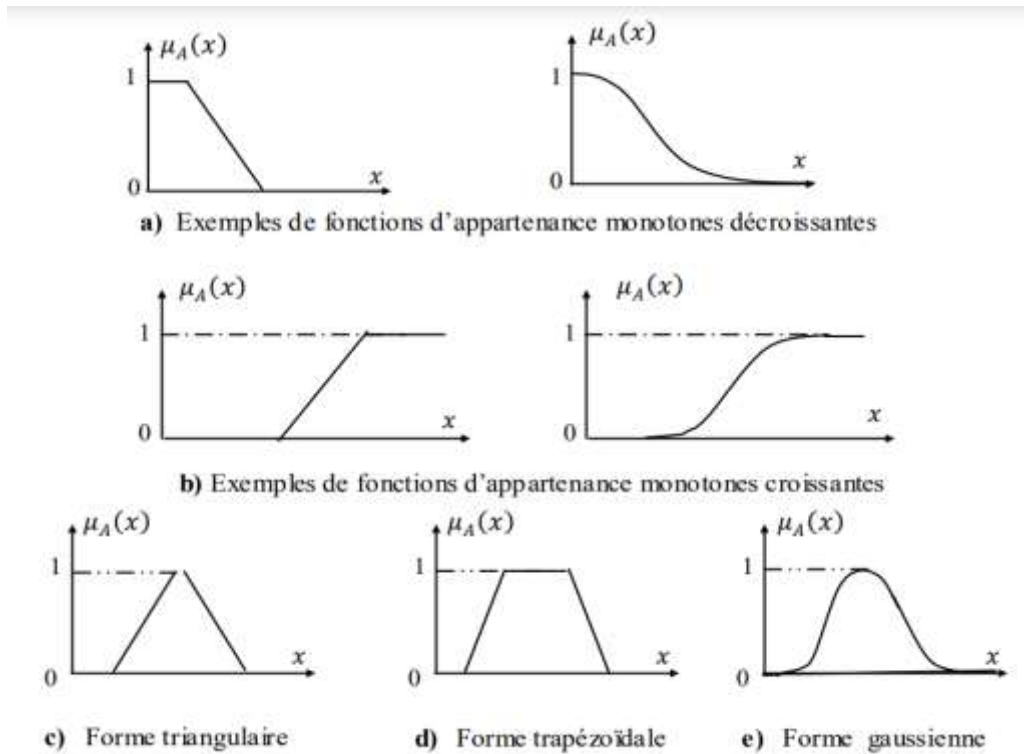


Figure 1.2: Différentes formes des fonctions d'appartenance [6]

I.1.4.3 Opérateurs de la logique floue

Une fois les ensembles flous définis, des opérations mathématiques sur ces types d'ensembles ont été développées. Les opérateurs mathématiques développés sont très similaires aux opérateurs traditionnels de la théorie des ensembles.[7]

Différents appellations	Implication flou
Zadeh	$Max\{min((\mu_A(x), \mu_B(y)), 1 - \mu_A(x))\}$
Mamdani	$Min(\mu_A(x), \mu_B(y))$
Reichenbach	$1 - (\mu_A(x) + \mu_B(y)) \cdot \mu_B(y)$
Willmott	$Max\{1 - \mu_A(x), min((\mu_A(x), \mu_B(y)), 1 - \mu_A(x))\}$
Diénes	$Max(1 - \mu_A(x), \mu_B(y))$
Brown goldi	$\mu_A(x) \leq \mu_B(y)$
Lukasiewicz	$Min(1, 1 - \mu_A(x) + \mu_B(y))$
Larsen	$(\mu_A(x), \mu_B(y))$

Tableau 1.1: Implication floue [8]

I.1.4.4 Règles floues

Une règle floue est une relation exprimée à l'aide d'une implication entre deux propositions floues. Par exemple, considérons deux variables linguistiques (V_1, U, T_{v1}) (V_2, U, T_{v2}) et les deux propositions suivantes "V₁ est A", "V₂ est B". Où A et B sont deux ensembles flous de V₁ et V₂.

Une règle floue peut être définie comme si V₁ est A alors V₂ est B. En général, certaines règles floues sont nécessaires pour prendre une décision dans une situation donnée. Les règles floues

peuvent être écrites de différentes manières [8] : *Linguistique * symbolique * par matrice d'inférence.

I.2 Neuro-flou

I.2.1 La combinaison des réseaux neuronaux et de la logique floue

Les techniques de réseaux de neurones peuvent être combinées dans des systèmes flous. Cela permet de l'adapter aux nouvelles situations et aux nouvelles règles, lors du fonctionnement en temps réel. De même, des techniques de logique floue peuvent être introduites dans les réseaux de neurones pour affiner et améliorer l'expressivité et la flexibilité des réseaux. Ces combinaisons sont appelées systèmes hybrides : Neuro-flou.

Effectivement, le choix de la manière d'intégrer les techniques de connaissance floue et d'apprentissage des réseaux de neurones dans un réseau hybride est crucial pour maximiser les avantages de chaque approche. Une intégration judicieuse permet de bénéficier de la simplicité de la connaissance floue et de la capacité d'apprentissage des réseaux de neurones, tout en réduisant le temps d'apprentissage grâce à l'introduction de connaissances a priori [9].

La figure I.3 résume les principes d'un système neuro-flou représentant l'interface entre la logique floue et les réseaux de neurones.

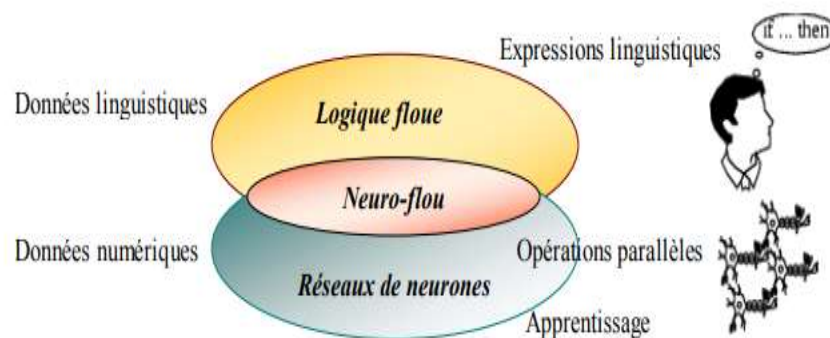


Figure I.3: Le principe d'un système neuro-flou [10]

I.2.2 Définition du système neuro-flou

Un système neuro-flou est un système flou entraîné par un algorithme d'apprentissage inspiré de la théorie des réseaux de neurones. Dans un système neuro-flou, les règles floues codées sont un prototype imprécis des données d'apprentissage. Les systèmes neuro-flous peuvent être utilisés comme approximateurs universels [11].

I.2.3 Structure d'un réseau neuro-flou

La structure d'un réseau neuro-flou s'inspire de la similarité organisationnelle entre un système d'inférence flou et un réseau neuronal à plusieurs couches.

Il existe quatre grandes catégories principales de combinaisons de réseaux de neuronaux avec la Logique floue : réseau flou neuronal, système neuronal/flou simultanément, modèles neuro-flous coopératifs et modèles neuro-flous hybrides[12].

I.2.4 Architectures neuro-floues

Depuis les années 1980, plusieurs architectures ont été développées qui visent généralement à maîtriser les systèmes complexes et les problèmes de classification, aboutissant à trois architectures hybrides neuro-floues [13].

I.2.4.1 Première architecture

La première méthode neuro-floue consiste à coder un système de raisonnement flou en forme de réseau de neurones multicouches dont les poids correspondent aux paramètres du système. L'architecture du réseau dépend du type de règles et des méthodes d'inférence, d'agrégation et de défuzzification choisies.

Par exemple, la forme "SI V_1 est A_{i1} et V_2 est A_{i2} ALORS $W = w_i$ " on utilise un réseau de neurones qui accepte en entrées les valeurs x_1 et x_2 tirées des variables V_1 et V_2 et dont les deux couches cachées correspondent respectivement au calcul de la valeur des fonctions d'appartenances A_{i1} pour x_1 et A_{i2} pour x_2 , respectivement, et est la valeur qui combine les conditions de chaque règle à l'aide de l'opérateur de combinaison approprié. Cette situation est illustrée schématiquement à la Figure I.4.

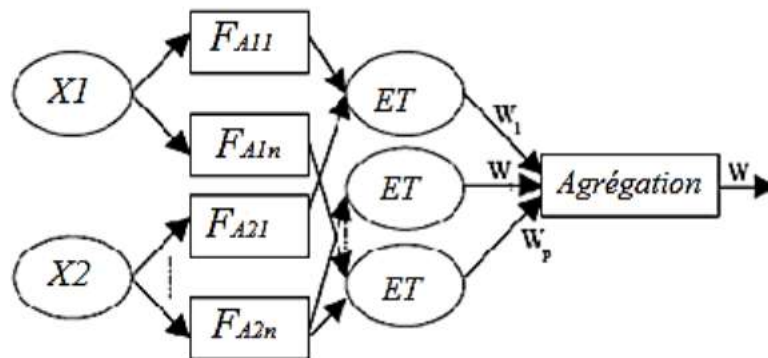


Figure I.4.: Première architecture Neuro-floue

► **Première couche** : Gère le calcul du degré d'appartenance pour chaque entrée. Les paramètres de cette couche caractérisent la fonction d'appartenance. Le calcul du degré d'appartenance est effectué par des neurones spécialisés, dont la fonction d'activation est la fonction d'appartenance. La fonction la plus couramment utilisée dans ce type d'architecture est la fonction gaussienne.

Il est donné comme suit :

$$y = e^{-(ax+b)^2} \quad (I.1)$$

Ainsi, les neurones dans la première couche présentent la forme suivante :

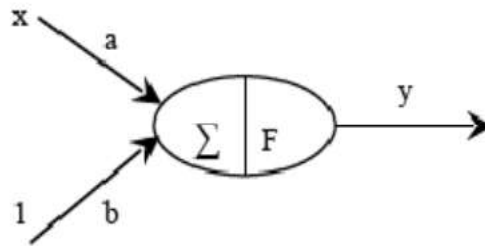


Figure I.5:Forme d'un neurone

➤ **La deuxième couche cachée** : évalue en parallèle les prémisses des règles. Dans ce cas, les paramètres de cette couche définissent l'opérateur de conjonction ET. Plusieurs méthodes ont été proposées pour l'évaluation floue de l'opérateur ET (cette évaluation est également appelée calcul de la valeur de vérité). En général, utilisez celui proposé par Lukasiewicz. Ce dernier offre une certaine facilité de mise en œuvre neuronale qui est raisonnablement satisfaisante. Défini par :

$$ET[fA_1(x_1), fA_2(x_2)] = \text{Max}(0, fA_1(x_1) + fA_2(x_2) - 1) \tag{I.2}$$

Si on pose :

$$u = fA_1(x_1) + fA_2(x_2) - 1$$

On aura :

$$ET[fA_1(x_1), fA_2(x_2)] = \text{Max}(0, u)$$

La fonction $\text{Max}(0, u)$ peut être évaluée approximativement par une fonction continue sigmoïde, elle est donnée par :

$$f = \text{Max}(0, u) = \frac{1}{1 + \exp\left(\frac{-u - 0,5}{0,227}\right)}$$

Donc notre opérateur de conjonction ET va prendre la forme suivante :

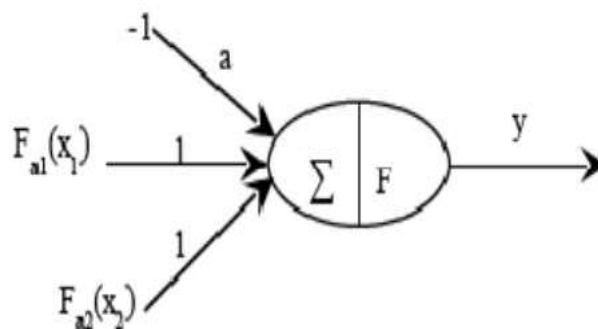


Figure I.6:Forme d'opérateur flou ET

➤ **La dernière couche** : Réalise l'opération d'inférence. Après avoir calculé les valeurs de vérités produites par les opérateurs conjonctifs, il faut maintenant établir une procédure permettant de déduire une sortie en relation avec ces valeurs de vérités. Cette procédure est réalisée soit par le

choix de ces valeurs, celui qui est le plus représentatif, c'est-à-dire celui qui a le plus haut degré de vérité, ou par le calcul du barycentre de toutes les valeurs. Une telle démarche peut être réalisée par un neurone unique, où les valeurs des vérités sont pondérées par les poids synaptiques de ce neurone.

Les poids synaptiques du neurone qui calcule le barycentre (qui fait la défuzzification) sont très importants, car ce sont eux qui pondèrent les résultats des règles. Une rétropropagation serait indispensable à appliquer sur cette couche (c'est-à-dire sur les poids synaptiques du neurone chargé de défuzzification).

➤ **Algorithme d'apprentissage** : Cet algorithme est basé sur l'algorithme de rétropropagation du gradient descendant. C'est le même que celui utilisé pour l'apprentissage des réseaux de neurones de type PMC. La seule différence est la normalisation des entrées du neurone chargé de défuzzification. Cette normalisation est nécessaire pour que les valeurs inférieures ou égales à 1 puissent être manipulées. Une telle approche est obtenue en divisant chaque entrée de ce neurone par la somme de toutes ses entrées.

I.2.4.2 Deuxième architecture

Cette méthode utilise les réseaux neuronaux ainsi que les systèmes flous associés en série ou en parallèle. Voici quelques variantes possibles :

- Fonctionnement du réseau neurones en amont du système flou (figure I.7). Les variantes d'entrée du système flou sont déterminées depuis les sorties du réseau neural (si elles ne sont pas directement mesurables) ou un réseau de neurones exécute une tâche de classification ou de reconnaissance des formes, suivie d'un système flou de soutien à la décision.

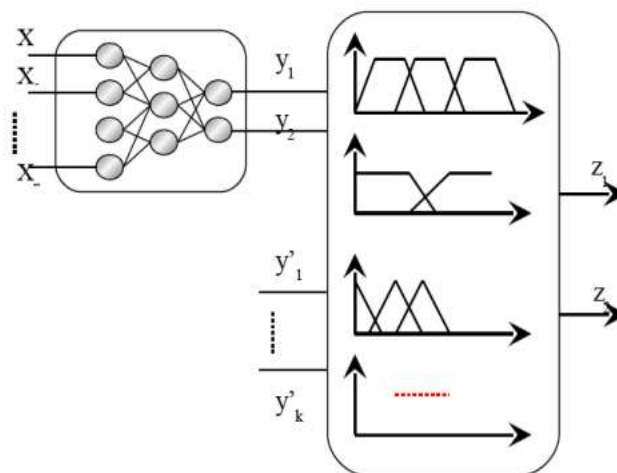


Figure I.7: Deuxième architecture des réseaux Neuro-Flou Réalisation en série

- un réseau de neurones fonctionnant en aval du système flou (figure I.8), dans le but d'adapter les sorties d'un système de contrôle flou aux nouvelles connaissances obtenues, les variables en sortie étant les erreurs dans les variables en sortie du système flouté.

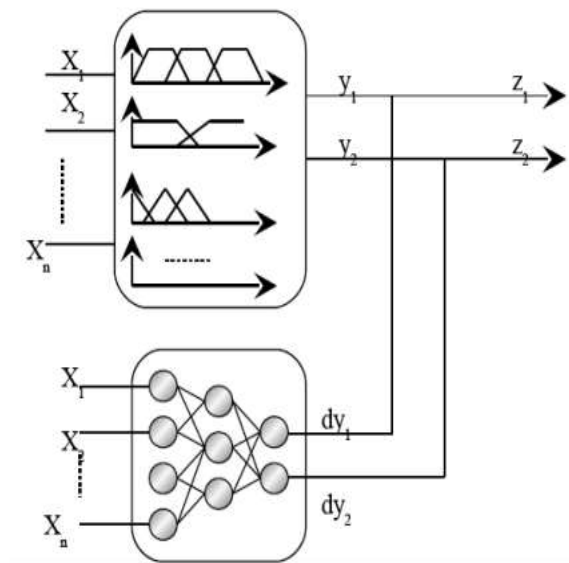


Figure I.8: Réseau de neurones fonctionnant en aval d'un système flou

I.2.4.3 Troisième architecture

Ce type d'association entre un réseau de neurones et un système flou revient à utiliser le réseau de neurones pour remplacer tout ou partie des composants du système flou. De tels réseaux sont utilisés pour apprendre des fonctions d'appartenance, calculer des inférences et effectuer des phases d'agrégation et de défuzzification. Des règles floues peuvent être extraites en analysant les corrélations qui existent entre les entrées et les sorties du réseau de neurones.

Ces approches sont très importantes car elles peuvent résoudre des problèmes importants :

- Déterminer et apprendre les fonctions d'appartenance.
- Détermination des règles.
- S'adapter à l'environnement du système.

I.2.5 Systèmes Neuro-Flous hybrides intégrés

I.2.5.1 Systèmes Neuro-Flous intégrés type Mamdani

Le système de neuro-flous Mamdani utilise la technique d'apprentissage de la propagation inverse pour apprendre les paramètres des fonctions d'appartenance. L'architecture du système neuro-flou de Mamdani est représentée sur la figure I.9.

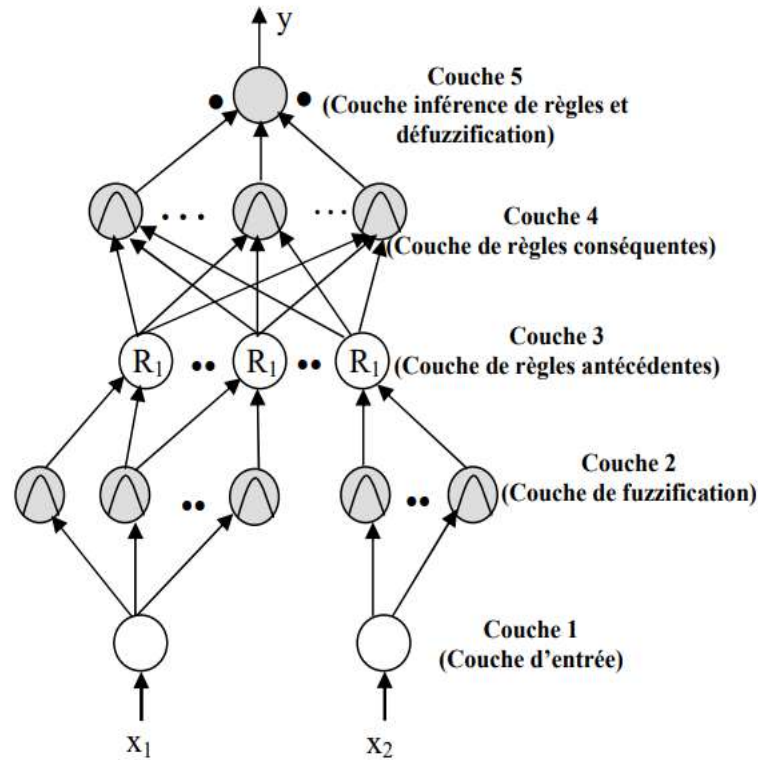


Figure I.9: Système neuro-flou type Mamdani

Couche 1 : C'est la couche d'entrée, chaque nœud dans cette couche correspond à une seule variable d'entrée, son rôle est de transmettre les valeurs d'entrée directement à la couche suivante.

Couche 2 : Il s'agit d'une couche de fuzzification et chaque nœud de cette couche correspond au seul label linguistique pour l'une des variables d'entrée de la couche 1. c.-à-d. Le lien de sortie représente la valeur d'appartenance qui indiquent dans quelle mesure une valeur d'entrée appartient à un ensemble flou et sont calculés à la couche 2. Un algorithme de classification détermine le nombre initial et les types de fonctions d'appartenance à affecter à chaque des variables d'entrée. Les formes finales des fonctions d'appartenance seront définies précisément pendant la formation en réseau.

Couche 3 : Il s'agit d'une couche de règles antécédentes. Le nœud à cette couche représente la partie précédente de la règle. En générale ce nœud utilise l'opérateur T-norm. La sortie d'un nœud dans la couche 3 représente la direction (firing strenght, en anglais) de la règle floue correspondante.

Couche 4 : il s'agit d'une Couche de règles conséquentes, ce nœud a en fait deux tâches. Combiner les antécédents de règles et déterminer le degré avec lequel ils appartiennent au label linguistique de sortie. Le nombre de nœuds dans cette couche est équivalent au nombre de règles.

Couche 5 : il s'agit d'une Couche de défuzzification. Ce nœud réalise la combinaison de toutes les parties conséquentes des règles à l'aide de l'opérateur de S-norme et calcule enfin la sortie nette (non floue) après défuzzification.

I.2.5.2 Systèmes neuro-flous intégrés type Takagi-Sugeno

Les systèmes neuro-flous de Takagi-Sugano utilisent une combinaison d'un algorithme de rétropropagation pour apprendre les fonctions d'appartenance et une méthode d'estimation des moindres carrés pour déterminer les coefficients des combinaisons linéaires dans les conclusions de règles. L'étape Une étape dans la procédure d'apprentissage se compose de deux parties : Dans la première partie, les entrées sont propagées et les paramètres de conclusion optimaux sont estimés par les moindres carrés. Tandis que les parties antécédentes (fonctions d'appartenance) sont supposées être fixes pour le cycle actuel dans l'ensemble d'apprentissage. Dans la deuxième partie, les entrées sont à nouveau propagées. Au cours de cette itération, nous utilisons la rétropropagation pour modifier les paramètres précédents et laisser les paramètres conclusion fixes. Ce processus est ensuite répété.

Le rôle de chaque couche dans le système neuro-flou de Takagi-Sugeno illustré à la figure I.10 est le suivant :

Couches 1, 2 & 3 : Leur rôle est similaire aux couches 1,2 et 3 du système type Mamdani.

Couche 4 (normalisation sens de la règle) : Chaque nœud de cette couche calcule le rapport du sens de la ième règle par rapport à la somme des sens de toutes les règles :

$$\bar{w}_i = \frac{w_i}{w_1 + w_2}, i = 1,2,\dots$$

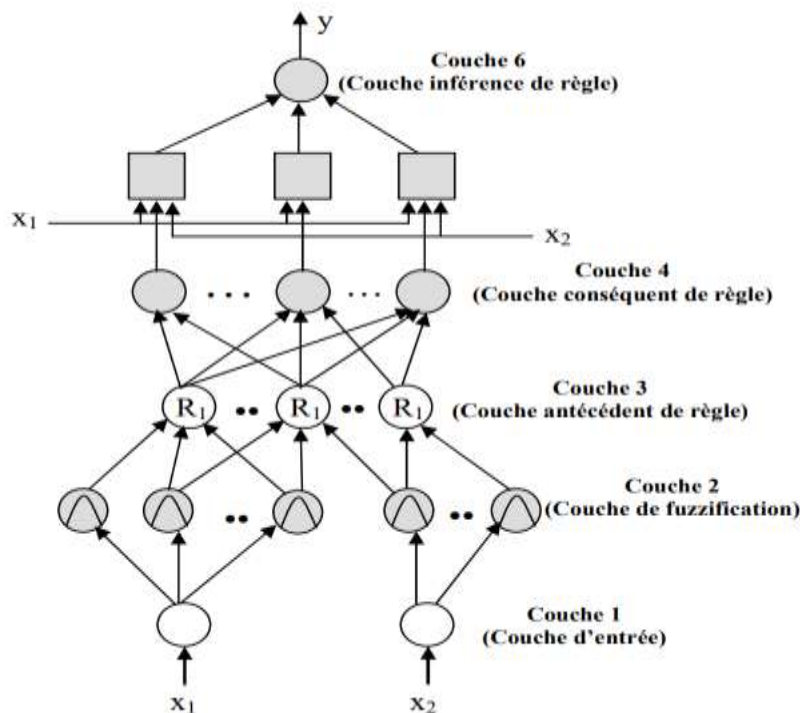


Figure I.10: Système neuro-flou type Takagi-Sugeno

Couche 5 (couche conséquente de règle) : Chaque nœud i dans cette couche est doté d'une fonction nœud :

$$\bar{w}_i f_i = \bar{w}_i (p_i x_1 + q_i x_2 + r_i)$$

Où \bar{w}_i est la sortie de la couche 4, et $\{p_i, q_i, r_i\}$ est l'ensemble des paramètres. Les paramètres conséquents sont déterminés par l'algorithme des moindres carrés moyens.

Couche 6 (couche inférence de règle) : L'unique nœud de cette couche calcule la sortie globale par addition de tous les signaux entrants :

$$Sortie\ globale = \sum \bar{w}_i f_i = \frac{\sum_i^n w_i f_i}{\sum_i^n w_i}$$

Dans ce qui suit, Nous décrivons brièvement le système intégré ANFIS qui utilise les complémentarités des réseaux de neurones et des systèmes avec des inférences floues implémentant un SIF de type Takagi-Sugeno.

I.2.5.3 Système ANFIS (Adaptative-Network-based Fuzzy Inference System)

ANFIS (acronyme pour Adaptive Network Based fuzzy Inference System) signifie système d'inférence floue implémenté dans le cadre de réseaux adaptatifs. Nous utilisons la méthode d'apprentissage hybride.

Cette architecture affine les règles floues dérivées d'experts humains pour décrire le comportement d'entrée-sortie de systèmes complexes. Une version modifiée du modèle ANFIS (Figure I.11) peut implémenter le système d'inférence floue de Tsukamoto. La version originale utilise un modèle flou de type Sugeno et codifie des règles floues de la forme :

$$R_r: Si\ x_i\ est\ \mu_{j_1}^{(1)} \wedge \dots \wedge x_n\ est\ \mu_{j_n}^{(n)}\ \text{ Alors}$$

$$y = \alpha_0^{(r)} + \alpha_1^{(r)} x_1 + \dots + \alpha_n^{(r)} x_n$$

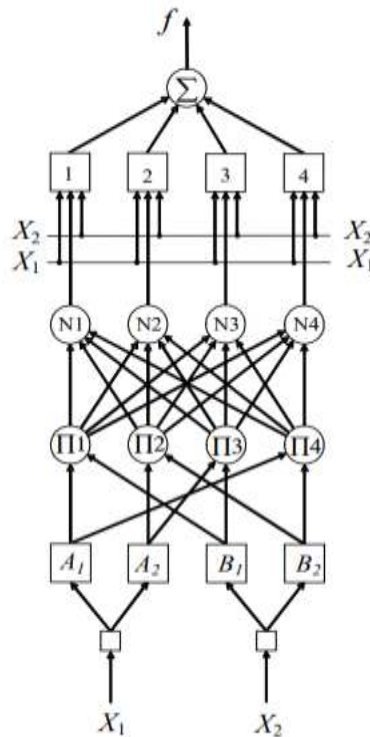


Figure I.11: système neuro-flou du type ANFIS

La sortie globale dans le SIF de Tsukamoto, est donnée par la moyenne pondérée de chaque sortie nette de règle induite par le sens de la règle (le produit ou minimum des degrés d'adéquation avec la partie prémisse) et les fonctions d'appartenance de sortie.

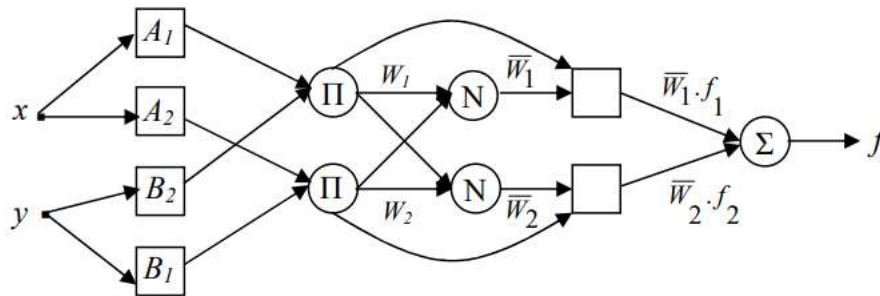


Figure I.12: Architecture ANFIS réalisant le système d'inférence floue de Tsukamoto

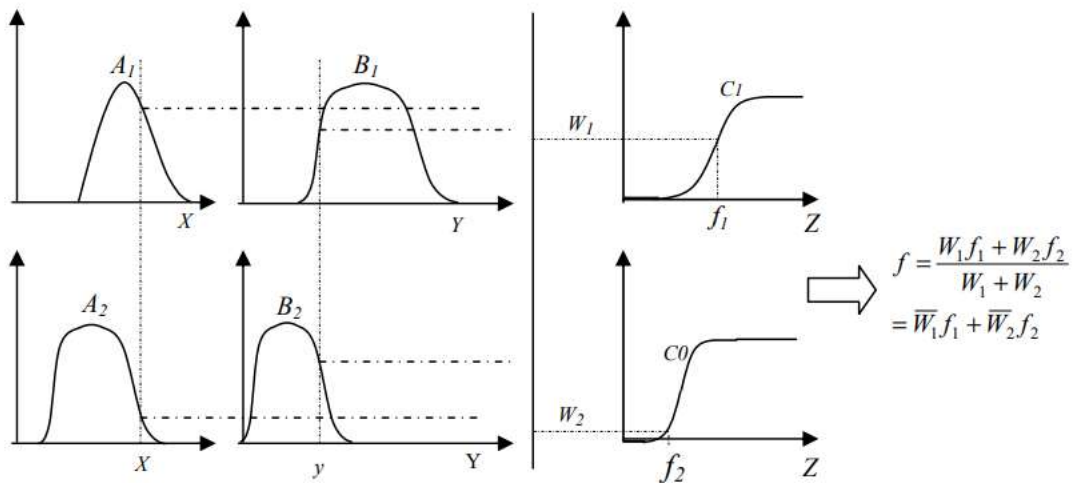


Figure I.13: Raisonnement flou de Tsukamoto

La première couche cachée est utilisée pour flouter les variables d'entrée. Des opérateurs de T-normes utilisés dans la deuxième couche cachée pour calculer la partie antécédente de règle. Une troisième couche normalise les sens des règles, suivie d'une quatrième couche, qui détermine les paramètres conséquents de la règle. La couche de sortie additionne tous les signaux d'entrée pour calculer la sortie totale.

Le procédé d'apprentissage du modèle ANFIS ne traite que du réglage des paramètres de la structure fixe. En effet, chaque terme linguistique est représenté par un seul ensemble flou. La méthode d'apprentissage utilise une combinaison de rétropropagation et d'estimation des moindres carrés moyens. La rétropropagation est utilisée pour mettre à jour les paramètres précédents (paramètres des règles floues) et MCM est utilisé pour mettre à jour les paramètres de conséquents (coefficients de combinaison linéaire). En cas d'utilisation de fonctions gaussiennes, on peut comparer le modèle ANFIS à un réseau de fonctions de base radiale (RBF).

Notez que le système ANFIS est implémenté dans la boîte à outils 'NeuroFuzzy' de MATLAB. Ce modèle donne d'excellents résultats en poursuite de trajectoire, approximation non linéaire, contrôle dynamique et traitement du signal [11].

Conclusion

Dans ce chapitre, j'ai présenté les notions de base de la théorie de la logique floue, la représentation de la déférence entre la logique floue et la logique classique et la représentation des éléments de base de la logique floue. Après j'ai présenté le système neuro-flou sa structure et son architecture et en termine avec la présentation du système neuro-flous hybrides intégrés.

Les systèmes flous exigent la disponibilité d'une expertise. Par conséquent, les performances de ces derniers sont étroitement liées aux techniques d'acquisition de connaissances et de justesse des informations acquises. Pour pallier à ces inconvénients, des méthodes combinant la logique floue et les réseaux de neurones ont été développées. De ce fait, les systèmes flous ainsi obtenus ont développé une capacité d'apprentissage.

Chapitre II : Le robot Delta

Introduction

Le robot delta est un type de robot parallèle qui se compose de trois bras articulés qui se déplacent indépendamment les uns des autres. Ces bras sont reliés à une plate-forme mobile qui leur permet de se déplacer dans l'espace de manière très rapide et précise.

Le robot delta est souvent utilisé dans l'industrie pour des tâches de précision telles que le tri, le positionnement, l'assemblage et l'emballage de produits. Sa structure lui permet de travailler dans des environnements où l'espace est limité ou où une grande précision est requise.

Les robots delta sont également utilisés dans d'autres domaines tels que la recherche scientifique, l'exploration spatiale, la médecine et la chirurgie, ainsi que dans des applications de divertissement telles que les parcs d'attractions et les spectacles.

II.1 Historique du robot delta

Le robot delta a été inventé par le professeur Reymond Clavel (Figure II.14) à l'École Polytechnique Fédérale de Lausanne (EPFL) en Suisse au début des années 1980. Il a conçu le robot delta en utilisant une configuration parallèle innovante de trois bras qui se déplacent indépendamment les uns des autres.

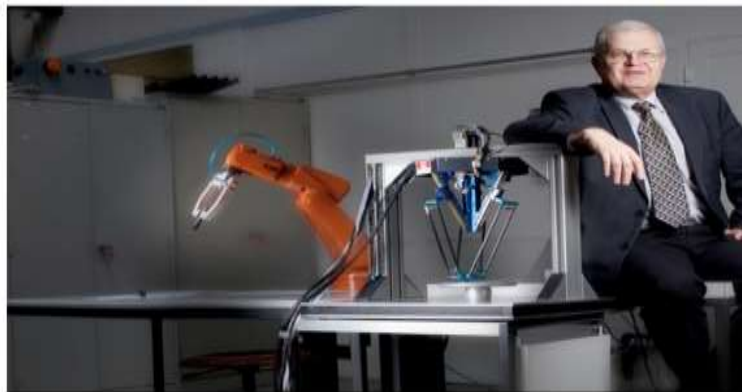


Figure II.14: Raymond Clavel, inventeur du robot Delta [14].

Le premier prototype de robot delta a été construit en 1985 et présenté lors d'une conférence internationale sur la robotique à Tokyo en 1986. Depuis lors, la conception du robot delta a été améliorée et développée pour une utilisation dans différents domaines industriels et commerciaux.

Le robot delta est devenu populaire dans l'industrie automobile pour sa capacité à assembler les petites pièces avec une grande précision. Il a également été utilisé dans l'industrie alimentaire pour trier et emballer les produits alimentaires à haute vitesse.

Avec l'avancement de la technologie, les robots delta ont été de plus en plus utilisés dans les domaines de la médecine, de l'exploration spatiale et de l'industrie de la mode pour la fabrication et la production de vêtements. Aujourd'hui, le robot delta est considéré comme l'un des robots les plus

précis et les plus rapides utilisés dans l'industrie et continue d'être développé pour de nouveaux domaines d'application [14][15]

II.2 Caractéristiques du Robot Delta

Le robot Delta a quatre caractéristiques principales qui le différencient des robots industriels conventionnels (robots sériels)

- Le Delta est un robot parallèle, c'est-à-dire que la connexion entre sa base d'amarrage et son organe terminal se fait à travers plusieurs chaînes cinématiques. (Trois chaînes cinématiques identiques pour le Delta conventionnel), Cette configuration permet d'augmenter considérablement la rigidité et la précision du robot et cela réduit l'espace de travail.
- La nacelle du robot (structure portant l'élément terminal) a la propriété de conserver en permanence une configuration horizontale et ce, de manière totalement passive, c'est-à-dire qu'il n'y a aucun contrôle sur l'orientation de la nacelle (propriété obtenue par la construction mécanique du robot).
- Les moteurs chargés d'entraîner le robot sont fixés au niveau de la base d'amarrage du robot. Cela signifie que la masse en mouvement du robot et son inertie sont fortement réduites, permettant ainsi des vitesses et des accélérations élevées lors de l'utilisation de ce robot.
- Les barres parallèles qui représentent un parallélogramme. Cette propriété permet à la nacelle du robot delta d'effectuer uniquement des mouvements de translation [16].

II.3 Constituants du robot Delta

La description des composants du robot Delta sur lesquels nous travaillerons est illustrée dans la figure II.15 [16].

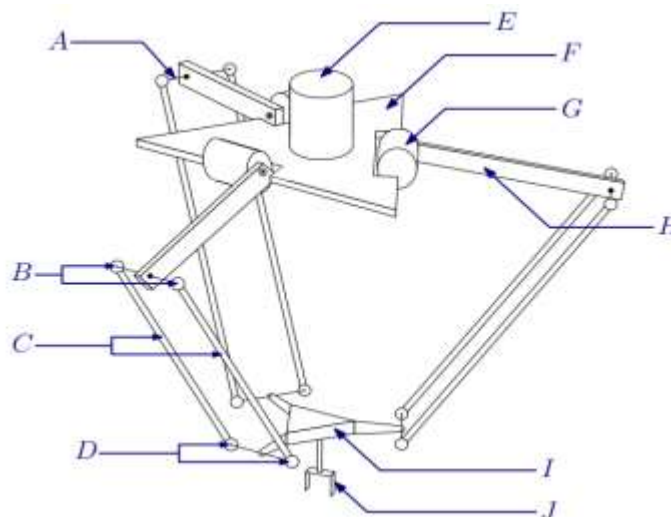


Figure II.15: Schéma descriptif des constituants du robot Delta.

- A : Barre de liaison entre le bras (H) et une paire de barres parallèles (C).
- B : Rotules supérieures.

- C : Barres parallèles. Ces deux barres forment, avec deux barres de liaison (A), un parallélogramme mobile, grâce aux rotules disposées aux quatre angles de ce parallélogramme. Les barres parallèles du Delta permettent de mouvoir la nacelle en translation seulement (selon les trois axes spatiaux) et les différentes rotations sont éliminées par l'effet de la configuration mécanique du robot (de manière passive).
- D : Rotules inférieures.
- E : Connexion de la base au battis fixe (à titre indicatif).
- F : Base du robot Delta sur laquelle sont fixées les trois chaînes cinématiques (bouclées) du robot.
- G : Actionneur d'une chaîne cinématique (articulation active). Cet actionneur constitue le seul élément du robot accessible directement par une commande ou pour des mesures (par le biais de son instrumentation). Celui-ci est fixé sur la base et actionne un seul bras (H).
- H : Bras d'une chaîne cinématique du Delta. Transmet le mouvement du moteur (G) vers la nacelle (I) à travers les barres parallèles (A).
- I : Nacelle du robot Delta. La nacelle est portée par trois chaînes cinématiques identiques. La nacelle constitue un support pour l'organe terminal (J).
- J : Organe terminal du robot Delta (Pince, . . .).

II.4 Le modèle géométrique

La modélisation géométrique d'un système robotique implique la création d'équations qui établissent une relation entre la position de l'extrémité de l'appareil, connue sous le nom d'effecteur, et les coordonnées articulaires du bras manipulateur (Figure II.16) [16].

Tous les ouvrages que nous avons consulté qui traitent le sujet du robot Delta utilisent le même modèle géométrique établi par Clavel.

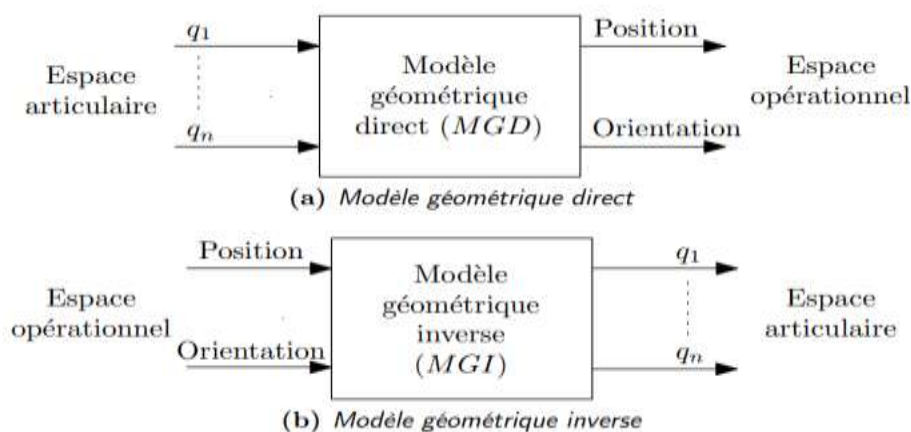


Figure II.16: *Modèle géométrique.*

La symétrie ternaire que présente le Delta (voir les figures II.17 et II.18). Nous permettons de travailler sur un seul bras.

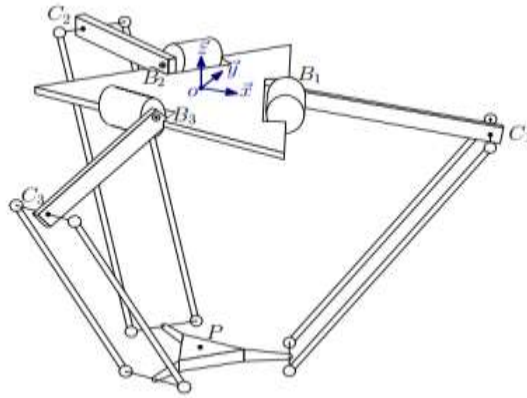


Figure II.17: Représentation schématique du robot Delta [16].

Le travail se fera d'une façon indépendante sur chaque repère. Les deux relations matricielles suivantes définissent le passage entre le repère $(x_0, y_0, z_0)_{R_0}$ et le repère $(x_i, y_i, z_i)_{R_i}$ (figure II.18).

$$\begin{pmatrix} x_0 \\ y_0 \\ z_0 \end{pmatrix} = \begin{pmatrix} \cos \varphi_i & -\sin \varphi_i & 0 \\ \sin \varphi_i & \cos \varphi_i & 0 \\ 0 & 0 & 1 \end{pmatrix} \begin{pmatrix} x_i \\ y_i \\ z_i \end{pmatrix} \tag{II.1}$$

$$\begin{pmatrix} x_i \\ y_i \\ z_i \end{pmatrix} = \begin{pmatrix} \cos \varphi_i & \sin \varphi_i & 0 \\ -\sin \varphi_i & \cos \varphi_i & 0 \\ 0 & 0 & 1 \end{pmatrix} \begin{pmatrix} x_0 \\ y_0 \\ z_0 \end{pmatrix}$$

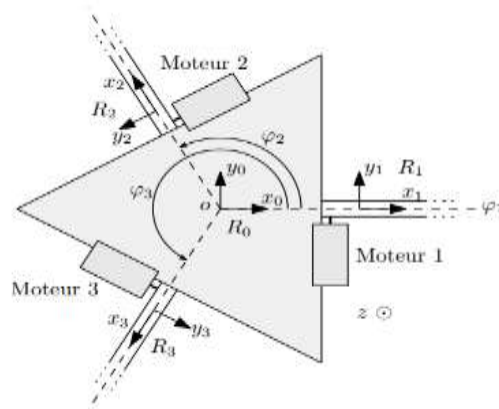


Figure II.18: Vue du dessus de la base du Delta.

Les valeurs prises par les φ_i sont $\varphi_1 = 0$, $\varphi_2 = 2\pi/3$ et $\varphi_3 = 4\pi/3$ selon que $i = 1, 2$ ou 3 . La figure II.19 présente le schéma annoté d'un bras du Delta en considérant les deux barres parallèles comme une seule.

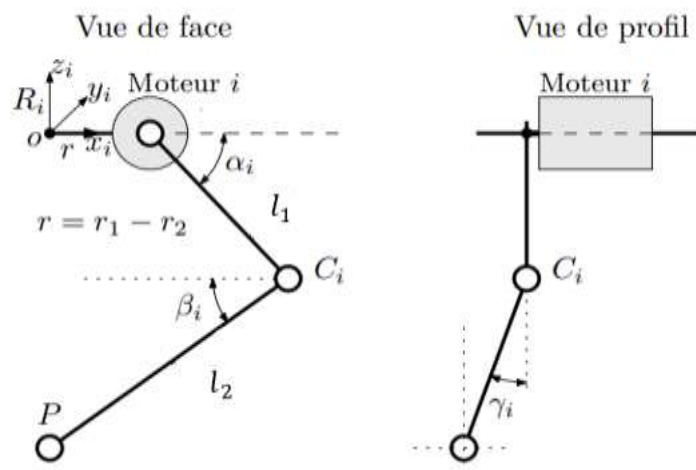


Figure II.19: Schéma simplifié d'un bras du Delta.

Les points C_i sont utilisés comme repères intermédiaires lors de l'élaboration des modèles géométriques directs et inverses. Leurs coordonnées correspondent aux valeurs suivantes :

$$\begin{pmatrix} x \\ y \\ z \end{pmatrix}_{C_i/R_i} = \begin{pmatrix} r + l_1 \cos \alpha_i \\ 0 \\ -l_1 \sin \alpha_i \end{pmatrix} \quad (\text{II.2})$$

Pour trouver la position de C_i dans la sphère de rayon l_1 et d'origine R_0 , Avec $i = 1, 2$ ou 3 . En appliquant la rotation de repère :

$$\begin{pmatrix} x \\ y \\ z \end{pmatrix}_{C_i/R_0} = \begin{pmatrix} \cos \varphi_i & -\sin \varphi_i & 0 \\ \sin \varphi_i & \cos \varphi_i & 0 \\ 0 & 0 & 1 \end{pmatrix} \begin{pmatrix} r + l_1 \cos \alpha_i \\ 0 \\ -l_1 \sin \alpha_i \end{pmatrix}$$

$$\begin{pmatrix} x \\ y \\ z \end{pmatrix}_{C_i/R_0} = \begin{pmatrix} (r + l_1 \cos \alpha_i) \cos \varphi_i \\ (r + l_1 \cos \alpha_i) \sin \varphi_i \\ -l_1 \sin \alpha_i \end{pmatrix} \quad (\text{II.3})$$

On peut voir aussi que le centre de l'effecteur $P(x, y, z)_{R_0}$ est l'intersection de trois sphères de rayon l_2 centrées en C_i , avec $i = 1, 2$ et 3 . L'équation qui définit la sphère est la suivante :

$$(X - x)^2 + (Y - y)^2 + (Z - z)^2 = l_2^2 \quad (\text{II.4})$$

(X, Y, Z) : les coordonnées d'un point appartenant à la sphère.

(x, y, z) : les coordonnées du centre de cette sphère.

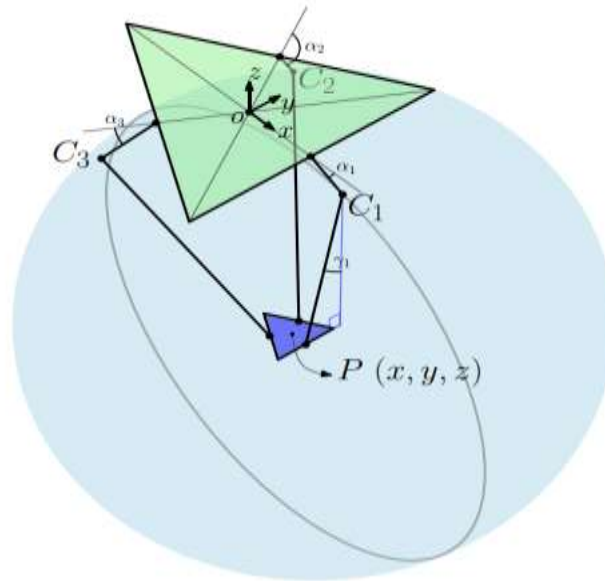


Figure II.20: Sphère de centre P et de rayon l_2 passant par C_1 .

II.4.1 Modèle géométrique Direct

Le modèle géométrique a pour objectif de calculer la position de l'organe terminale en fonction des angles articulaires ($\alpha_1, \alpha_2, \alpha_3$).

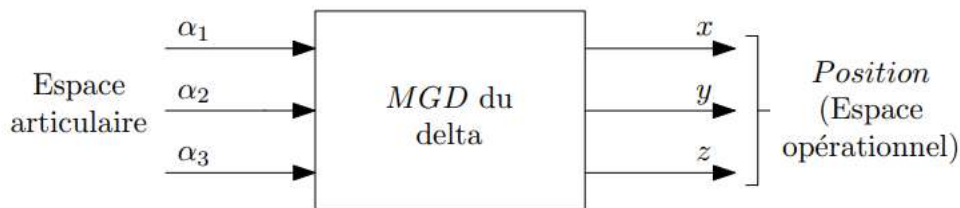


Figure II.21: Modèle géométrique direct du robot Delta.

Le calcul du *MGD* pour une structure parallèle telle que le robot Delta n'est pas aussi systématique que pour d'autres types de structures. La méthode présentée est celle développée par Clavel dans [17].

En remplaçant l'équation (II.3) dans l'équation de la sphère (figure II.20) on trouve :

$$\begin{aligned}
 &x^2 - 2x(r + l_1 \cos \alpha_i) \cos \varphi_i + (r + l_1 \cos \alpha_i)^2 \cos^2 \varphi_i + \\
 &y^2 - 2y(r + l_1 \cos \alpha_i) \sin \varphi_i + (r + l_1 \cos \alpha_i)^2 \sin^2 \varphi_i + \\
 &z^2 + 2zl_1 \sin \alpha_i + l_1^2 \sin^2 \alpha_i = l_2^2
 \end{aligned}
 \tag{II.5}$$

Cela se traduit par :

$$\begin{aligned}
 &x^2 - 2x(r + l_1 \cos \alpha_i) \cos \varphi_i + y^2 - 2y(r + l_1 \cos \alpha_i) \sin \varphi_i + \\
 &z^2 + 2zl_1 \sin \alpha_i = l_2^2 - l_1^2 - r^2 - 2rl_1 \cos \alpha_i
 \end{aligned}
 \tag{II.6}$$

avec : $i = 1, 2, 3$

En supposant :

$$\begin{cases} D_i = -l_2^2 + l_1^2 + r^2 + 2rl_1 \cos \alpha_i \\ E_i = 2(r + l_1 \cos \alpha_i) \cos \varphi_i \\ F_i = 2(r + l_1 \cos \alpha_i) \sin \varphi_i \\ G_i = -2l_1 \sin \alpha_i \end{cases}$$

Ainsi, nous sommes confrontés à la résolution de trois équations à trois inconnues :

$$\begin{cases} x^2 - E_1x + y^2 + F_1y + z^2 - G_{1z} = -D_1 & \text{(II.7.a)} \end{cases}$$

$$\begin{cases} x^2 - E_2x + y^2 + F_2y + z^2 - G_{2z} = -D_2 & \text{(II.7.b)} \end{cases}$$

$$\begin{cases} x^2 - E_3x + y^2 + F_3y + z^2 - G_{3z} = -D_3 & \text{(II.7.c)} \end{cases}$$

Après la simplification on trouve :

$$\begin{aligned} y &= \frac{H_1}{H_2}z + \frac{H_3}{H_2} \\ x &= \frac{H_5}{H_2}z + \frac{H_4}{H_2} \end{aligned} \quad \text{avec} \quad \begin{cases} H_1 = -(E_3 - E_1)(G_2 - G_1) + (G_3 - G_1)(E_2 - E_1) \\ H_2 = -(F_3 - F_1)(E_2 - E_1) + (E_3 - E_1)(F_2 - F_1) \\ H_3 = -(D_3 - D_1)(E_2 - E_1) + (E_3 - E_1)(D_2 - D_1) \\ H_4 = (D_3 - D_1)(F_2 - F_1) - (F_3 - F_1)(D_1 - D_2) \\ H_5 = -(G_3 - G_1)(F_2 - F_1) - (F_3 - F_1)(G_1 - G_2) \end{cases} \quad \text{(II.8)}$$

Finalement :

$$z = \frac{M - \sqrt{M^2 - 4LN}}{2L} \quad \text{(II.9)}$$

Sachant que :

$$\begin{cases} L = 1 + \frac{H_5^2 + H_1^2}{H_2^2} \\ M = -\frac{2(H_5H_4 + H_1H_3)}{H_2^2} + \frac{E_1H_5 + F_1H_1}{H_2} + G_1 \\ N = D_1 + \frac{H_4^2 + H_3^2}{H_2^2} - \frac{E_1H_4 + F_1H_3}{H_2} \end{cases}$$

II.4.2 Modèle géométrique inverse

Le modèle géométrique inverse du robot Delta permet de déterminer les trois angles articulaires $(\alpha_1, \alpha_2, \alpha_3)$ nécessaires pour atteindre une position souhaitée (x, y, z) de la nacelle.

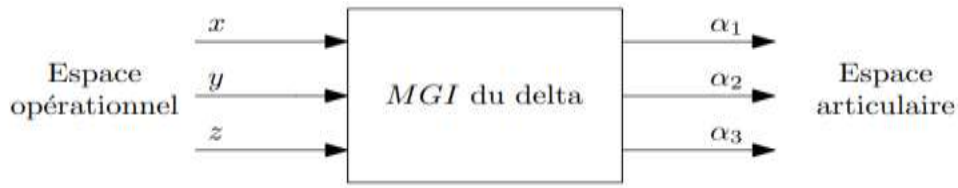


Figure II.22: Modèle géométrique inverse du robot Delta.

On peut résoudre les équations (II.7) en $(\alpha_1, \alpha_2, \alpha_3)$ et trouver le MGI. En reprenant l'équation (II.6), on sépare les termes en $(\cos \alpha_i)$ et en $(\sin \alpha_i)$:

$$(2l_1 z) \sin \alpha_i + (2rl_1 - 2l_1 x \cos \varphi_i - 2l_1 y \sin \varphi_i) \cos \alpha_i = (l_2^2 - l_1^2 - r^2 + 2ry \sin \varphi_i - z^2 - y^2 - x^2 + 2rx \cos \varphi_i) \quad (\text{II.10})$$

Avec : $i = 1, 2, 3$

En supposant :

$$\begin{cases} A_i = -2l_1 z \\ B_i = 2rl_1 - 2l_1 x \cos \varphi_i - 2l_1 y \sin \varphi_i \\ C_i = l_2^2 - l_1^2 - r^2 + 2ry \sin \varphi_i - z^2 - y^2 - x^2 + 2rx \cos \varphi_i \end{cases}$$

Donc l'équation (II.10) devient:

$$-A_i \sin \alpha_i + B_i \cos \alpha_i = C_i$$

En posant :

$$t = \tan \left(\frac{\alpha_i}{2} \right)$$

Après la simplification on trouve :

$$\alpha_i = 2 \arctan \left(\frac{-A_i + \sqrt{A_i^2 + B_i^2 - C_i^2}}{B_i + C_i} \right) \quad (\text{II.11})$$

Avec $i=1, 2$ et 3 .

Remarque : pour cette solution on voit qu'il y a un problème de singularité mathématique de type $(0/0)$ et on voit bien que le numérateur s'annule aussi $(B_i + C_i = 0)$. Une autre formulation du MGI qui règle ce problème.

Dans le repère R_i on a :

$$\begin{cases} (x_{c_i} - x)^2 + (y_{c_i} - y)^2 + (z_{c_i} - z)^2 = l_2^2 \\ \begin{pmatrix} x \\ y \\ z \end{pmatrix}_{C_i/R_i} = \begin{pmatrix} r + l_1 \cos \alpha_i \\ 0 \\ -l_1 \sin \alpha_i \end{pmatrix} \end{cases} \quad (\text{II.12})$$

(x, y, z) les coordonnées de la nacelle, sachant que $y_{c_i} = 0$. La première équation devient :

$$x_{c_i}^2 - 2xx_{c_i} + x^2 + y^2 + z_{c_i}^2 - 2zz_{c_i} + z^2 = l_2^2 \quad (\text{II.13})$$

Après la simplification on trouve :

$$2x_{c_i}(x - r) + 2zz_{c_i} = l_1^2 - l_2^2 - r^2 + x^2 + y^2 + z^2 \quad (\text{II.14})$$

Donc :

$$x_{c_i} = \frac{B - 2zz_{c_i}}{A} \quad (\text{II.15})$$

Sachant que :

$$\begin{cases} A = 2(x - r) \\ B = l_1^2 - l_2^2 - r^2 + x^2 + y^2 + z^2 \end{cases}$$

Finalement :

$$z_{c_i} = \frac{b - \sqrt{b^2 - ac}}{a}$$

$$\text{Avec : } \begin{cases} a = A^2 + 4z^2 \\ b = -2z(rA - B) \\ c = B^2 - 2rAB - A^2l_1^2 + A^2r_1^2 \end{cases}$$

Donc la solution :

$$\alpha_i = \arcsin\left(\frac{-z_{c_i}}{l_1}\right) \quad (\text{II.16})$$

II.5 Modèle cinématique

La modélisation cinématique d'un robot manipulateur consiste à établir les relations mathématiques entre les mouvements des articulations (vitesses articulaires) et les mouvements de l'organe terminal (vitesses opérationnelles). Cela permet de comprendre comment les mouvements des articulations affectent les mouvements de l'extrémité du robot.

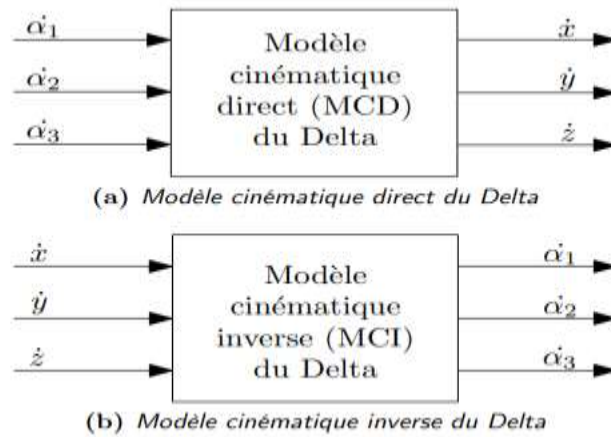


Figure II.23: Modèles cinématiques du robot Delta.

Le modèle cinématique directe du robot Delta (voir figure II.23.a), les vitesses opérationnelles de l'organe terminal sont reliées aux vitesses articulaires (généralisées) à l'aide de la matrice Jacobienne.

Dans le modèle cinématique inverse du robot Delta (voir figure II.23.b), le lien entre les vitesses articulaires (vitesses généralisées) et les vitesses opérationnelles (vitesses de la nacelle) peut être établi à l'aide de la matrice Jacobienne inverse.

Effectivement, dans le cadre du modèle géométrique, les relations qui relient les coordonnées articulaires (α_1 , α_2 et α_3) aux coordonnées opérationnelles (x , y et z) peuvent être utilisées pour déterminer la matrice Jacobienne du robot Delta de manière simple et directe.

$$\begin{cases} x = f_1(\alpha_i) \\ y = f_2(\alpha_i) \\ z = f_3(\alpha_i) \end{cases} \quad (\text{II.17})$$

En dérivant les relations entre les coordonnées articulaires (α_1 , α_2 , α_3) et les coordonnées opérationnelles (x , y , z) par rapport à la variable temporelle t :

$$\begin{pmatrix} \dot{x} \\ \dot{y} \\ \dot{z} \end{pmatrix} = \begin{pmatrix} \frac{\partial f_1}{\partial \alpha_1} & \frac{\partial f_1}{\partial \alpha_2} & \frac{\partial f_1}{\partial \alpha_3} \\ \frac{\partial f_2}{\partial \alpha_1} & \frac{\partial f_2}{\partial \alpha_2} & \frac{\partial f_2}{\partial \alpha_3} \\ \frac{\partial f_3}{\partial \alpha_1} & \frac{\partial f_3}{\partial \alpha_2} & \frac{\partial f_3}{\partial \alpha_3} \end{pmatrix} \begin{pmatrix} \dot{\alpha}_1 \\ \dot{\alpha}_2 \\ \dot{\alpha}_3 \end{pmatrix} \quad (\text{II.18})$$

Si l'on exprime les angles α_i en fonction des variables opérationnelles (x , y et z), les équations peuvent être formulées comme suit :

$$\begin{cases} \alpha_1 = g_1(x, y, z) \\ \alpha_2 = g_2(x, y, z) \\ \alpha_3 = g_3(x, y, z) \end{cases} \quad (\text{II.19})$$

II.5.1 Modèle cinématique direct

Dans le cadre de l'établissement du modèle cinématique direct (MCD) du robot Delta, les formules (II.8) et (II.9) de la méthode établie dans la référence sont utilisées. Les éléments de la Jacobienne sont obtenus en effectuant une dérivation numérique de ces formules, comme expliqué précédemment.

$$\left\{ \begin{array}{l} \frac{\partial f_i}{\partial \alpha_1} = \frac{f_i(\alpha_1+\Delta, \alpha_2, \alpha_3) - f_i(\alpha_1, \alpha_2, \alpha_3)}{\Delta} \\ \frac{\partial f_i}{\partial \alpha_2} = \frac{f_i(\alpha_1, \alpha_2+\Delta, \alpha_3) - f_i(\alpha_1, \alpha_2, \alpha_3)}{\Delta} \\ \frac{\partial f_i}{\partial \alpha_3} = \frac{f_i(\alpha_1, \alpha_2, \alpha_3+\Delta) - f_i(\alpha_1, \alpha_2, \alpha_3)}{\Delta} \\ [f_1 \ f_2 \ f_3] \equiv MGD \end{array} \right. \quad (II.20)$$

Δ : le pas d'intégration et i allant de 1 à 3.

II.5.2 Modèle cinématique inverse

Pour établir le modèle cinématique inverse (MCI) du robot Delta, on peut utiliser la dérivation numérique des relations du modèle géométrique inverse. En adoptant le même principe que précédemment, nous pouvons obtenir les éléments de la Jacobienne inverse en utilisant la méthode des différences finies partielles.

$$\left\{ \begin{array}{l} \frac{\partial g_i}{\partial \alpha_1} = \frac{g_i(x+\Delta, y, z) - g_i(x, y, z)}{\Delta} \\ \frac{\partial g_i}{\partial \alpha_2} = \frac{g_i(x, y+\Delta, z) - g_i(x, y, z)}{\Delta} \\ \frac{\partial g_i}{\partial \alpha_3} = \frac{g_i(x, y, z+\Delta) - g_i(x, y, z)}{\Delta} \\ [g_1 \ g_2 \ g_3] \equiv MGI \end{array} \right. \quad (II.21)$$

Δ : le pas d'intégration et i allant de 1 à 3.

Conclusion

Les robots parallèles, tels que le robot Delta, présentent plusieurs avantages en termes de rigidité mécanique et de dynamique rapide. Le robot Delta est considéré comme la dernière génération de robots parallèles et se distingue par son poids léger et ses accélérations élevées.

Le robot Delta est un robot manipulateur performant et polyvalent, capable d'effectuer des mouvements rapides, précis et fluides. Sa structure en forme de delta, combinée à des repères individuels pour chaque bras, facilite la modélisation géométrique et le contrôle du robot, offrant ainsi des avantages significatifs dans diverses applications industrielles et de recherche.

***Chapitre III : Application de la
commande neuro-floue au robot Delta***

Introduction

La combinaison du modèle géométrique et de la commande neuro-floue pour la planification et le contrôle de la trajectoire du robot delta peut être implémentée et simulée dans MATLAB. MATLAB offre un environnement de développement puissant pour la modélisation, la simulation et l'implémentation de systèmes de contrôle robotique.

Dans ce chapitre nous allons présenter l'application par la simulation de la commande neuro-floue sur un robot Delta

III.1 Génération de trajectoires

La génération de trajectoires dans l'espace des articulations n'est pas possible pour le robot Delta. Par conséquent, il est nécessaire de générer des trajectoires dans l'espace opérationnel. Dans ce cas, la génération de trajectoires dans l'espace opérationnel repose sur la connaissance de l'évolution de la coordonnée curviligne $s(t)$. À partir de cette coordonnée curviligne, l'évolution des coordonnées articulaires $x(t)$, $y(t)$, $z(t)$ est obtenue par projection (voir figure III.24).

Ensuite, le modèle géométrique inverse est appliqué pour obtenir les références articulaires qui seront injectées dans les actionneurs du robot [1].

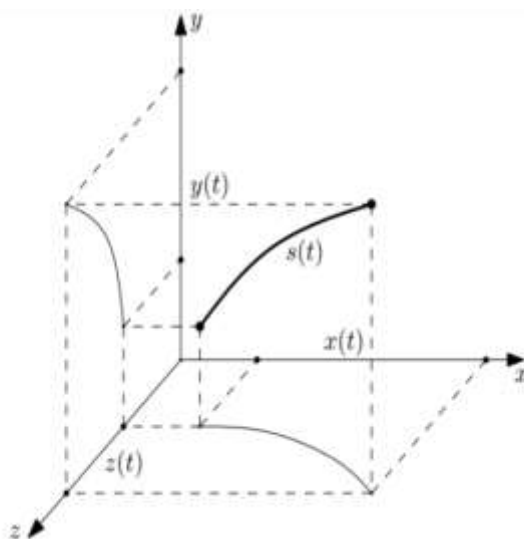


Figure III.24: Projection de la coordonnée curviligne $s(t)$ donnant $x(t)$, $y(t)$, $z(t)$.

III.2 Structure du régulateur sous Matlab

Le schéma suivant représente la structure de la commande neuro-floue et un moteur d'un bras du robot Delta.

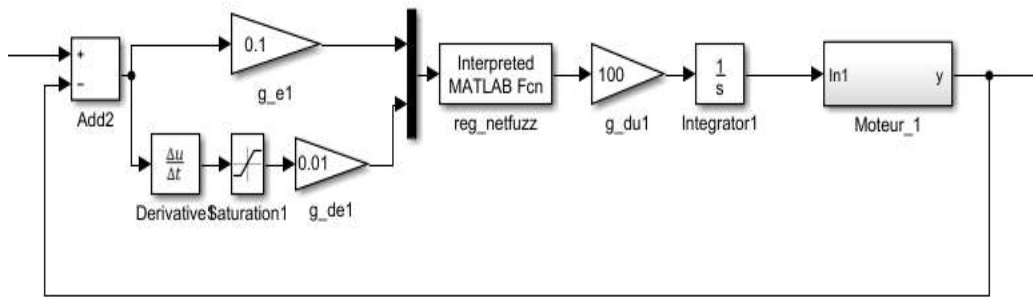


Figure III.25: Schéma Simulink de la commande neuro-floue et un moteur.

Le schéma ci-dessous montre le bloc d'un moteur :

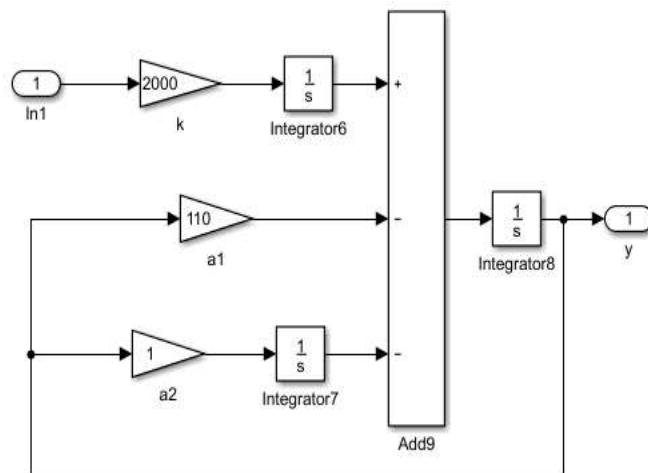


Figure III.26: bloc d'un moteur.

La figure suivante représente la génération de la trajectoire dans le plan (X, Y, Z) :

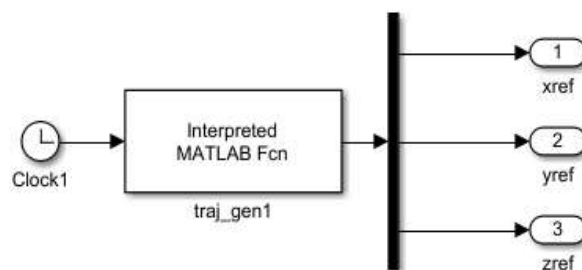


Figure III.27: Bloc de génération de trajectoire dans le plan (X, Y, Z).

Le schéma suivant représente la structure de la commande. Il regroupe deux blocs sont utilisés pour générées la trajectoire de consigne :

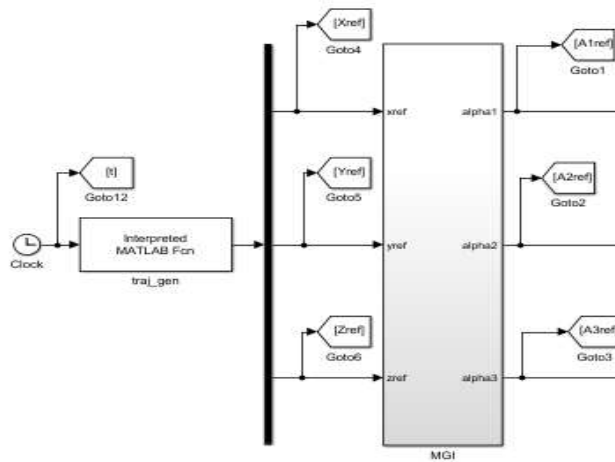


Figure III.28: Bloc de génération de la trajectoire de consigne.

La figure suivante représente le modèle géométrique direct et inverse :

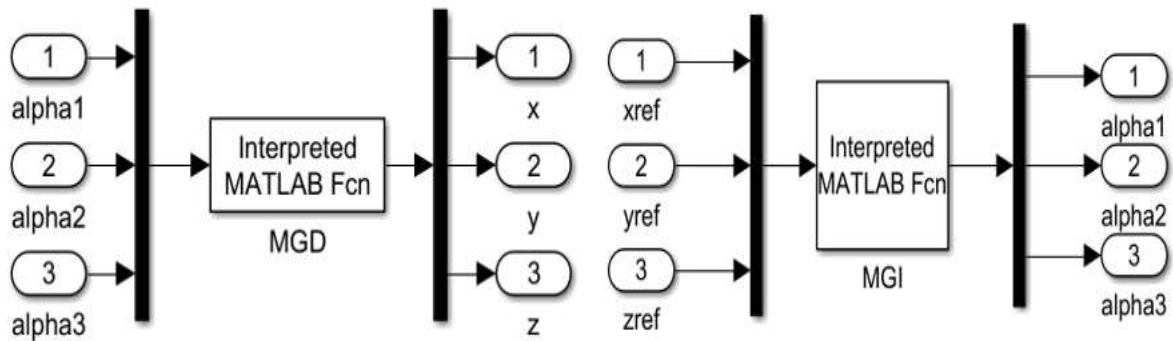


Figure III.29: Bloc du modèle géométrique direct et inverse.

III.3 Simulation sous l'environnement Matlab/Simulink

La combinaison du modèle géométrique et de la commande neuro-floue pour la planification et le contrôle de la trajectoire du robot delta peut être implémentée et simulée dans MATLAB. MATLAB offre un environnement de développement puissant pour la modélisation, la simulation et l'implémentation de systèmes de contrôle robotique.

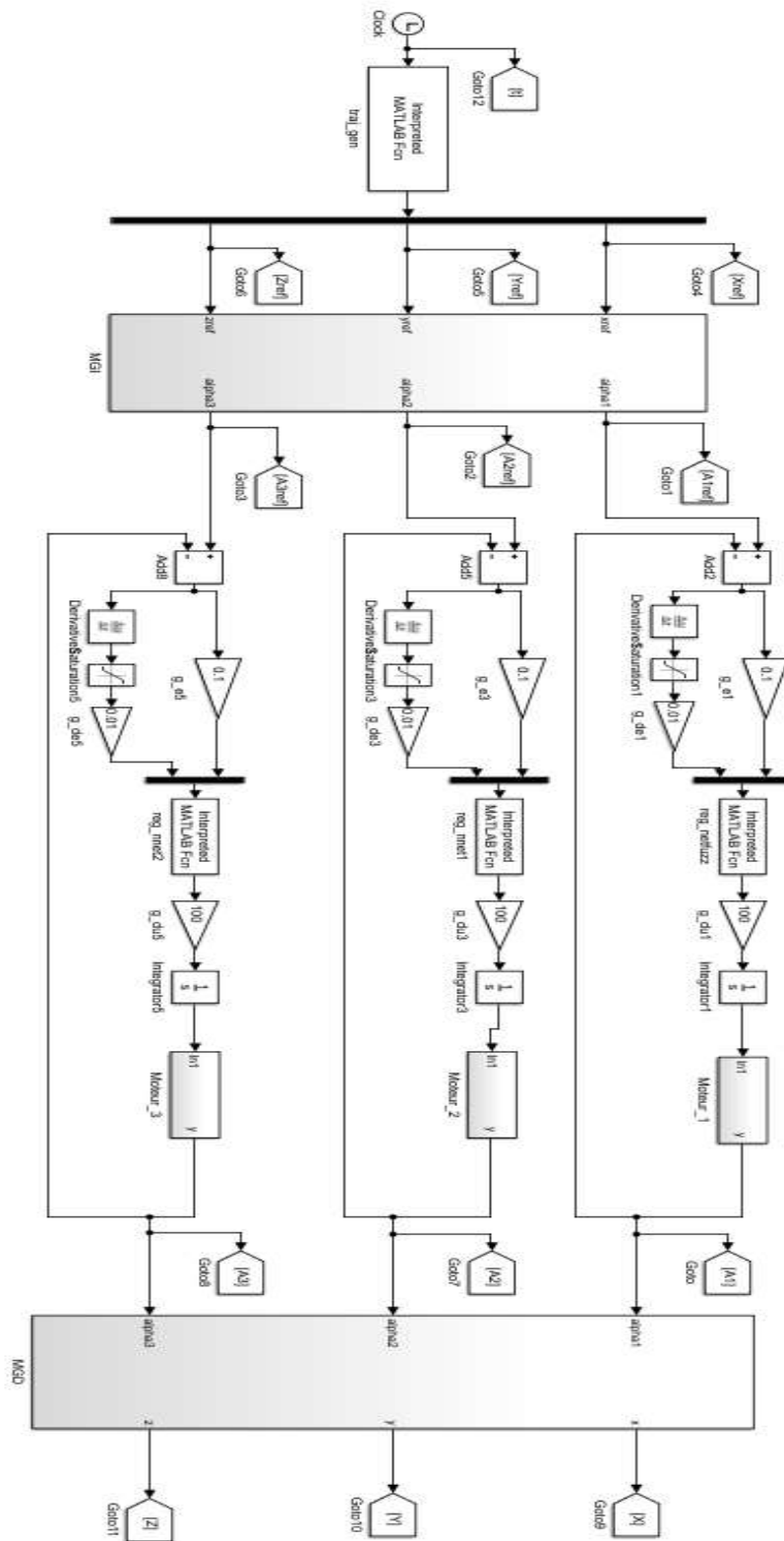


Figure III.30:Schéma de simulation.

Comme montre la figure III.30: la simulation de la trajectoire du robot Delta à l'aide du modèle géométrique pour calculer les positions et les orientations de l'effecteur final en fonction des angles des moteurs.

En combinant le modèle géométrique avec la commande de neuro-floue pour optimiser la trajectoire du robot Delta en temps réel et régler les paramètres du modèle géométrique, tels que la vitesse des moteurs, les gains de contrôle et les trajectoires intermédiaires, en fonction des entrées sensorielles et des objectifs de la tâche.

Il existe trois régulateurs indépendants, chacun prend en charge la capture des positions de l'articulation à laquelle il est rattaché, et calcule de façon complètement indépendante la commande à appliquer à son actionneur.

III.4 Les résultats de simulation

Pour simuler la trajectoire du robot delta à l'aide du modèle géométrique, nous pouvons utiliser les équations géométriques directes et inverses du robot delta pour calculer les positions et les orientations de l'effecteur final en fonction des angles des moteurs.

En utilisant MATLAB, nous pouvons simuler le comportement du robot delta dans différentes conditions et tester différentes stratégies de commande neuro-floue pour optimiser la trajectoire du robot.

III.4.1 Résultats de simulation pour la première trajectoire

La figure III.31: présente les consignes et les réponses de la trajectoire en fonction de temps ainsi que les erreurs de la poursuite, on remarque que la poursuite est assurée avec des faibles erreurs dans le plan X, Y, Z.

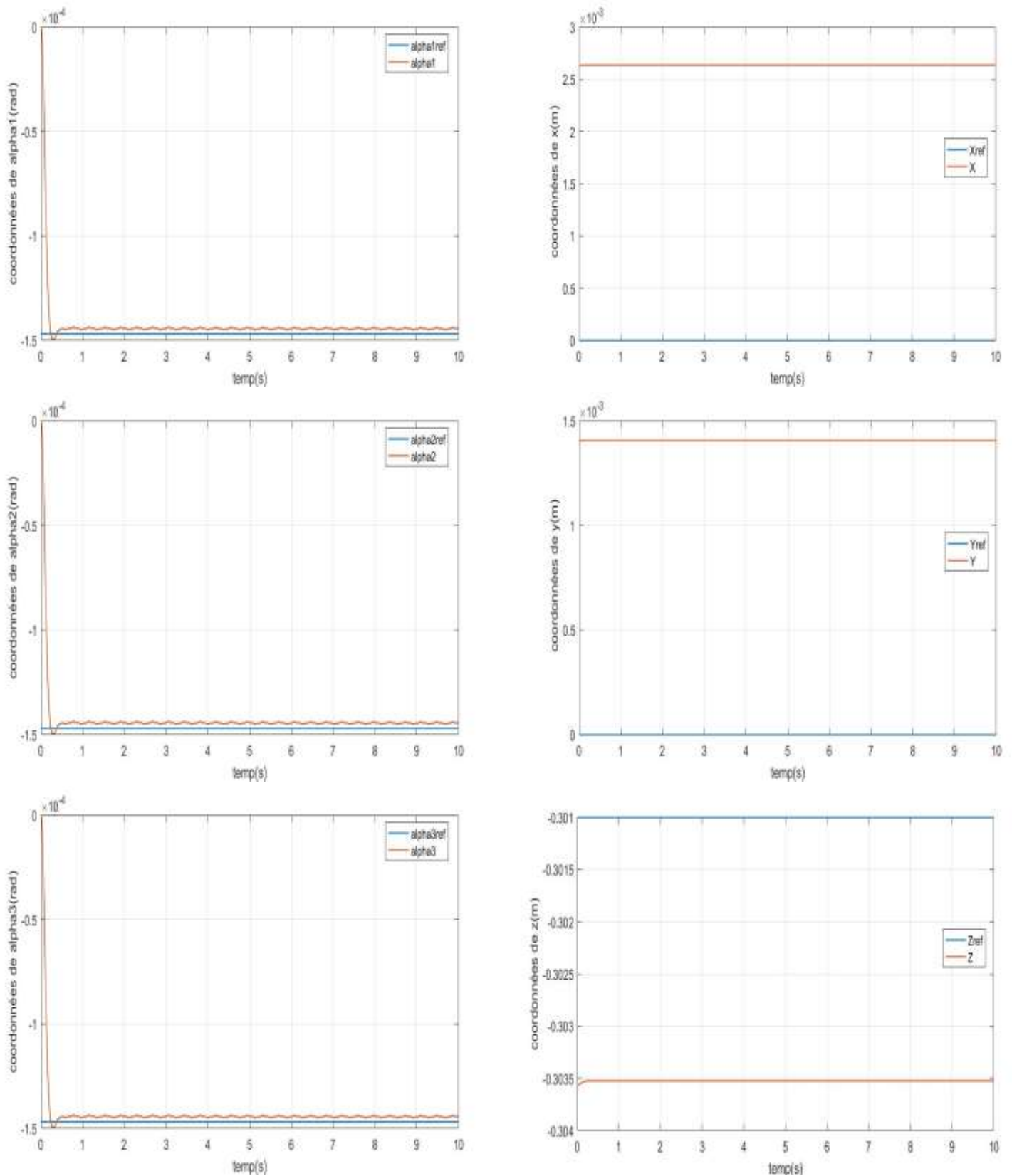


Figure III.31: Evolution temporelle des coordonnées (X, Y, Z) et des coordonnées (α_1 , α_2 , α_3) de poursuite de la trajectoire.

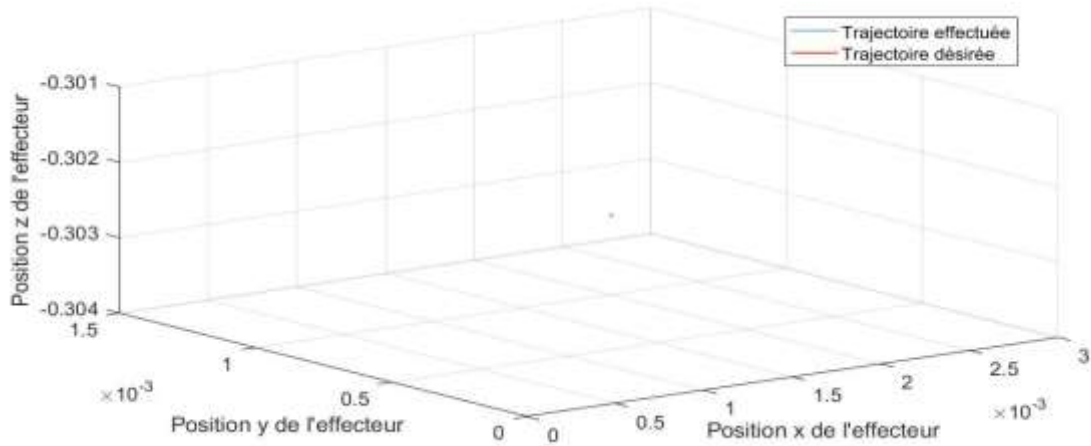


Figure III.32: Trajectoire dans le plan (X, Y, Z) .

La figure III.32 : présente une trajectoire effectuée et une trajectoire désirée (les consignes et les réponses).

D'après la trajectoire obtenue on constate que la poursuite est assurée avec des faibles erreurs dans le plan (X, Y, Z) .

III.4.2 Résultats de simulation pour la deuxième trajectoire

La figure (III.33) : présente les consignes et les réponses de la trajectoire de chaque articulation en fonction de temps et assurée avec des faibles erreurs.

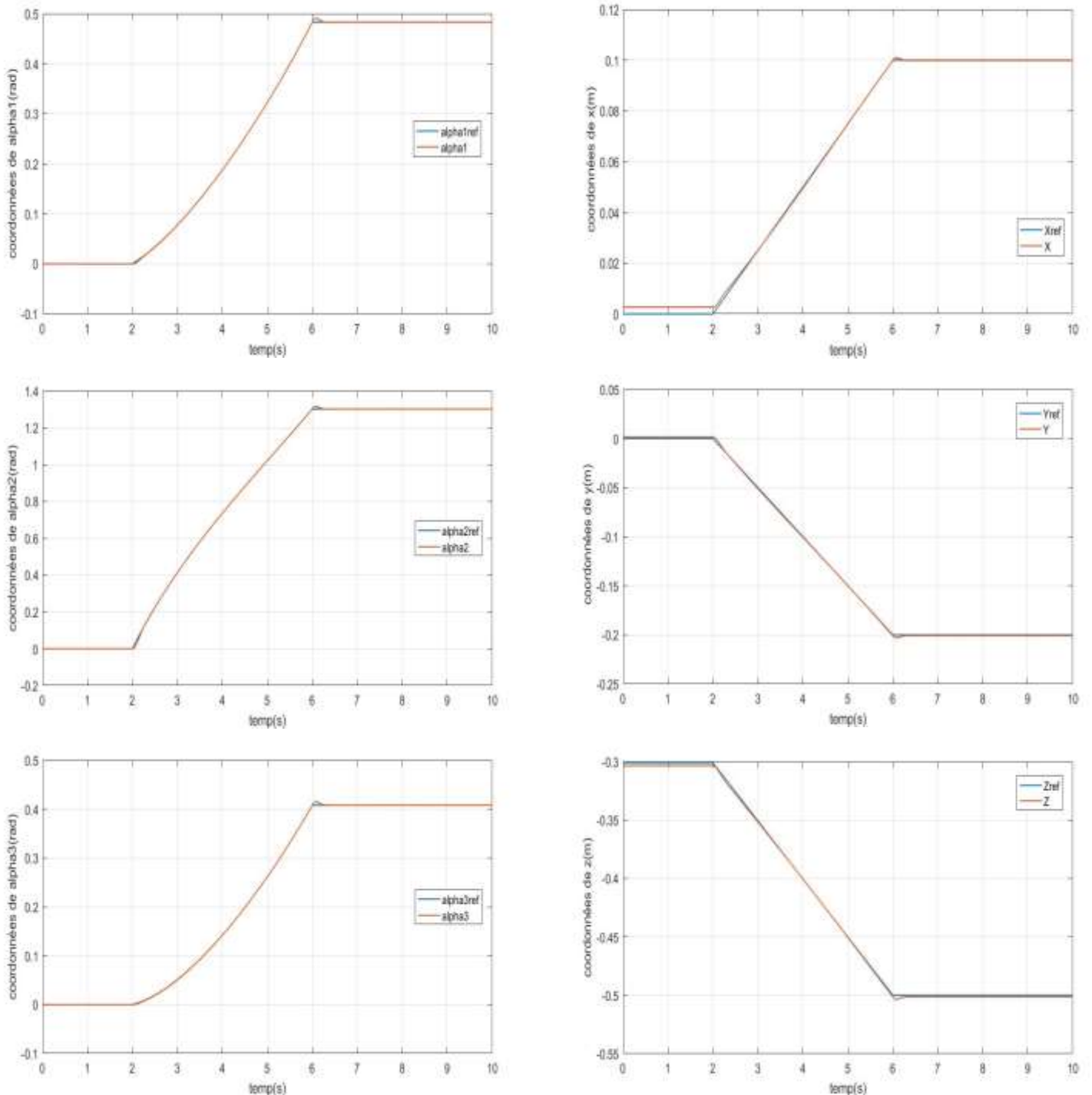


Figure III.33: Evolution temporelle des coordonnées (X, Y, Z) et des coordonnées (α_1 , α_2 , α_3) de poursuite de la trajectoire.

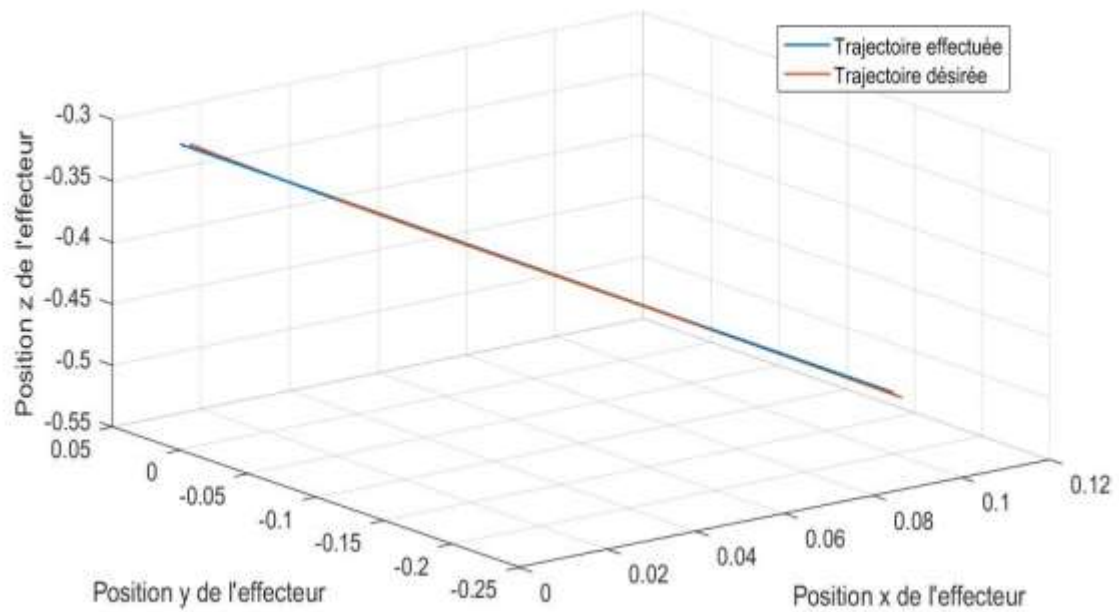


Figure III.34: Trajectoire linéaire dans le plan (X, Y, Z) .

La figure III.34 : présente une trajectoire effectuée et une trajectoire désirée (les consignes et les réponses).

D'après la trajectoire obtenue on constate que la poursuite est assurée avec des faibles erreurs dans le plan (X, Y, Z) .

III.4.3 Résultats de simulation pour la troisième trajectoire

La figure III.35 : présente les consignes et les réponses de la trajectoire de chaque articulation en fonction de temps et assurée avec des faibles erreurs.

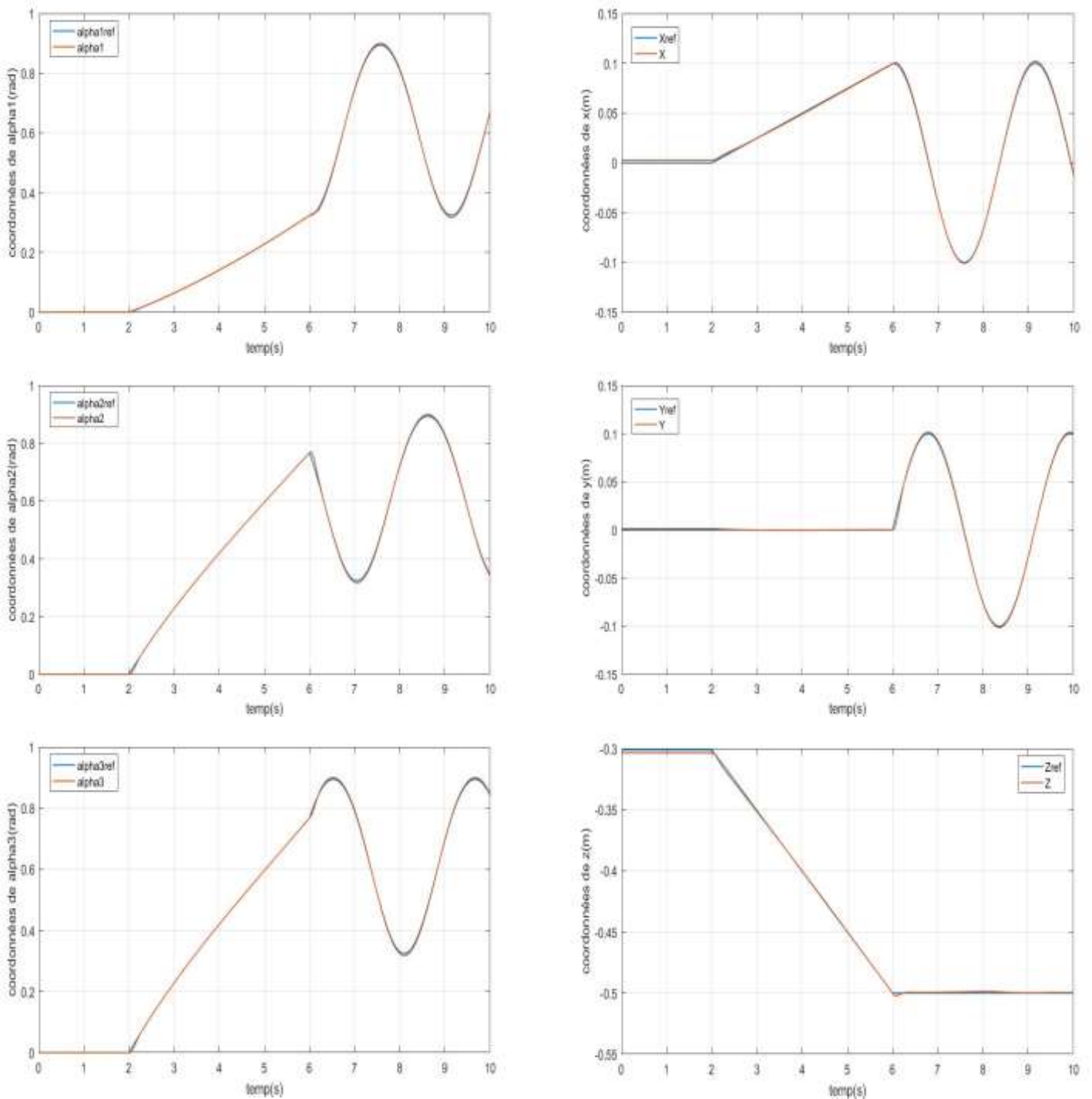


Figure III.35: Evolution temporelle des coordonnées (X, Y, Z) et des coordonnées (α_1 , α_2 , α_3) de poursuite de la trajectoire.

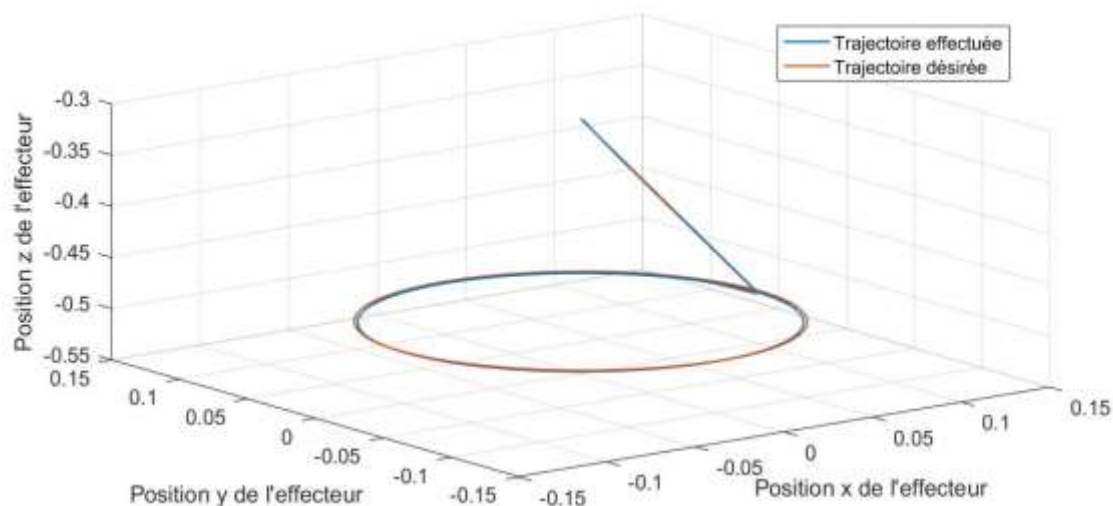


Figure III.36: Trajectoire circulaire dans le plan (X, Y, Z) .

Figure III.36 : présente une trajectoire circulaire effectuée et une trajectoire désirée (les consignes et les réponses).

D'après la réponse obtenue on constate que la poursuite est assurée avec des faibles erreurs dans le plan (X, Y, Z) .

III.5 Application de la commande neuro-floue adaptative au robot Delta

III.5.1 Les blocs d'apprentissage

Dans ce qui suit, nous allons appliquer la commande neuro-floue adaptative au robot Delta. La figure suivante présente les blocs d'apprentissage qui servent pour l'ajustement des paramètres des régulateurs neuro-flous.

Ces blocs sont identiques, chacun concerne une articulation du robot Delta. L'apprentissage se base sur l'algorithme de rétropropagation de l'erreur. Dans cet algorithme, nous avons besoin de calculer les taux de variations des sorties par rapport aux variations des entrées, ce qui peut poser quelques problèmes si les signaux d'entrées ne varient pas. Donc, il faut faire attention et éviter que ces taux prennent des valeurs inadéquates (très grandes), ce qui peut conduire à de mauvaises valeurs pour les paramètres des régulateurs.

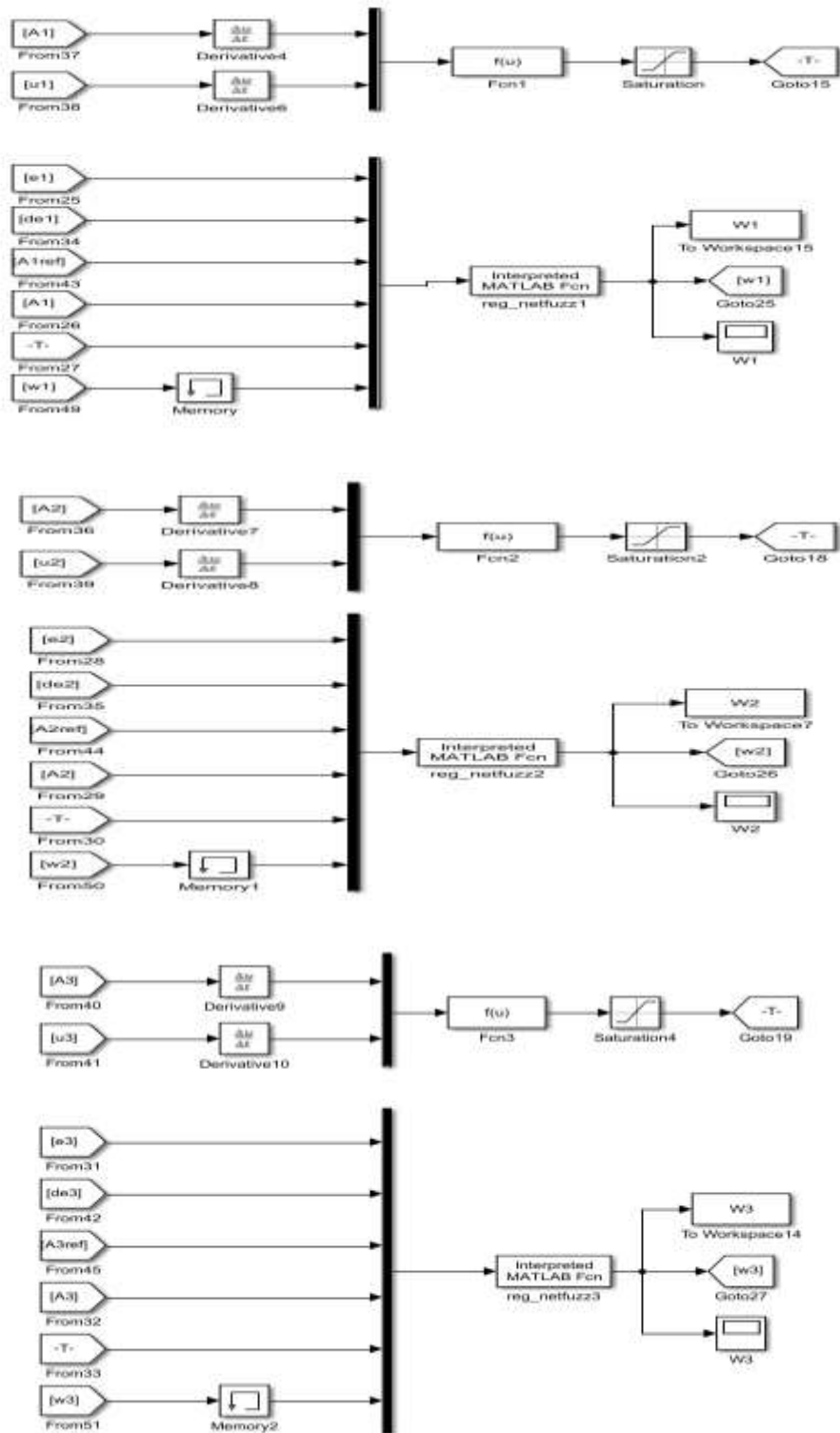


Figure III.37: Les blocs d'apprentissage

III.5.2 Schéma de simulation adaptative

La figure suivante présente la structure de la commande neuro-floue adaptative. Chaque régulateur a comme entrée, en plus des autres, celle concernant les paramètres w obtenus par les blocs d'apprentissage.

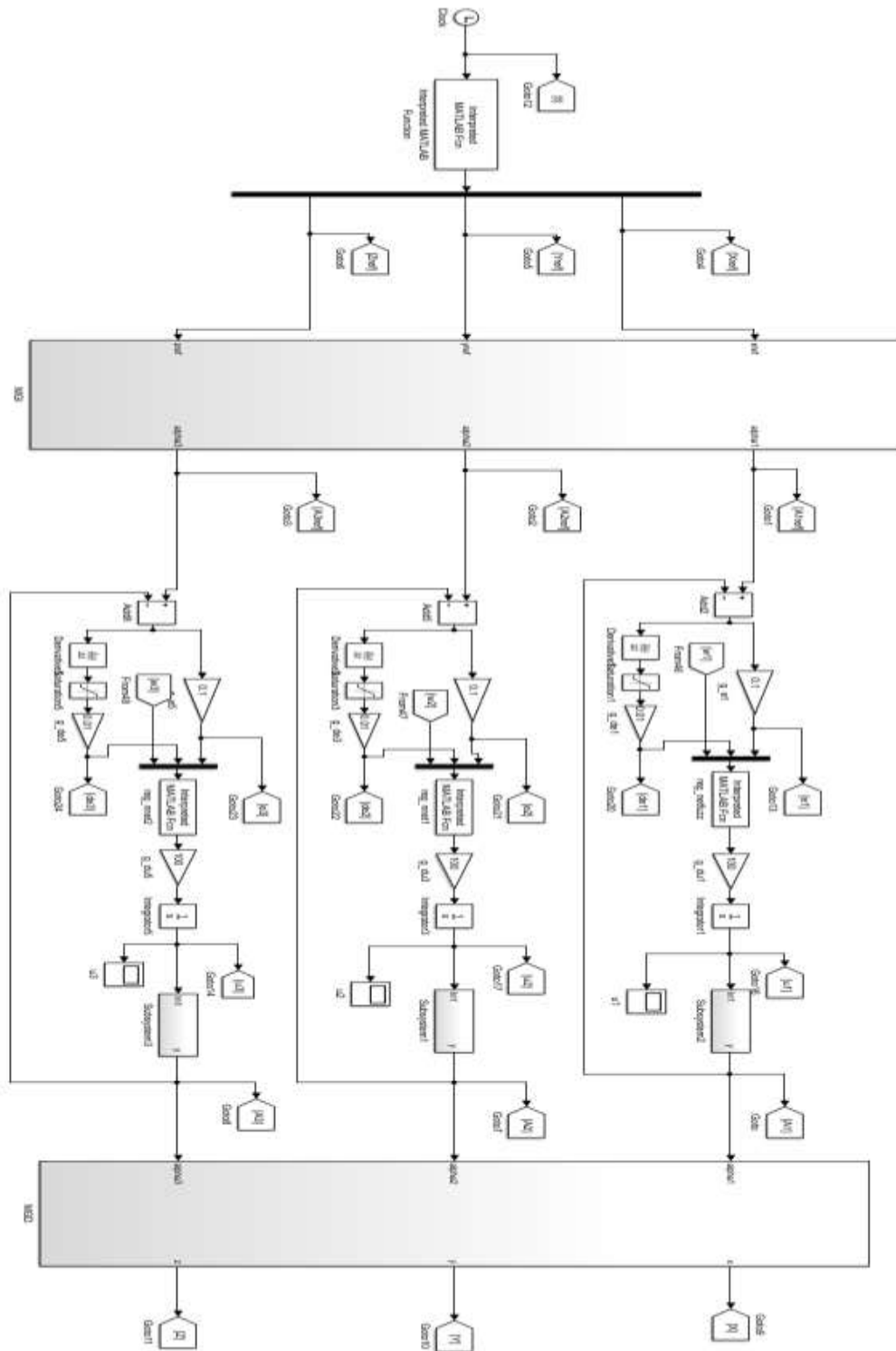


Figure III.38: Commande neuro-floue adaptative.

III.5.3 Résultats de simulation adaptative pour la troisième trajectoire

La figure suivante présente les résultats de simulation en utilisant la commande neuro-floue adaptative.

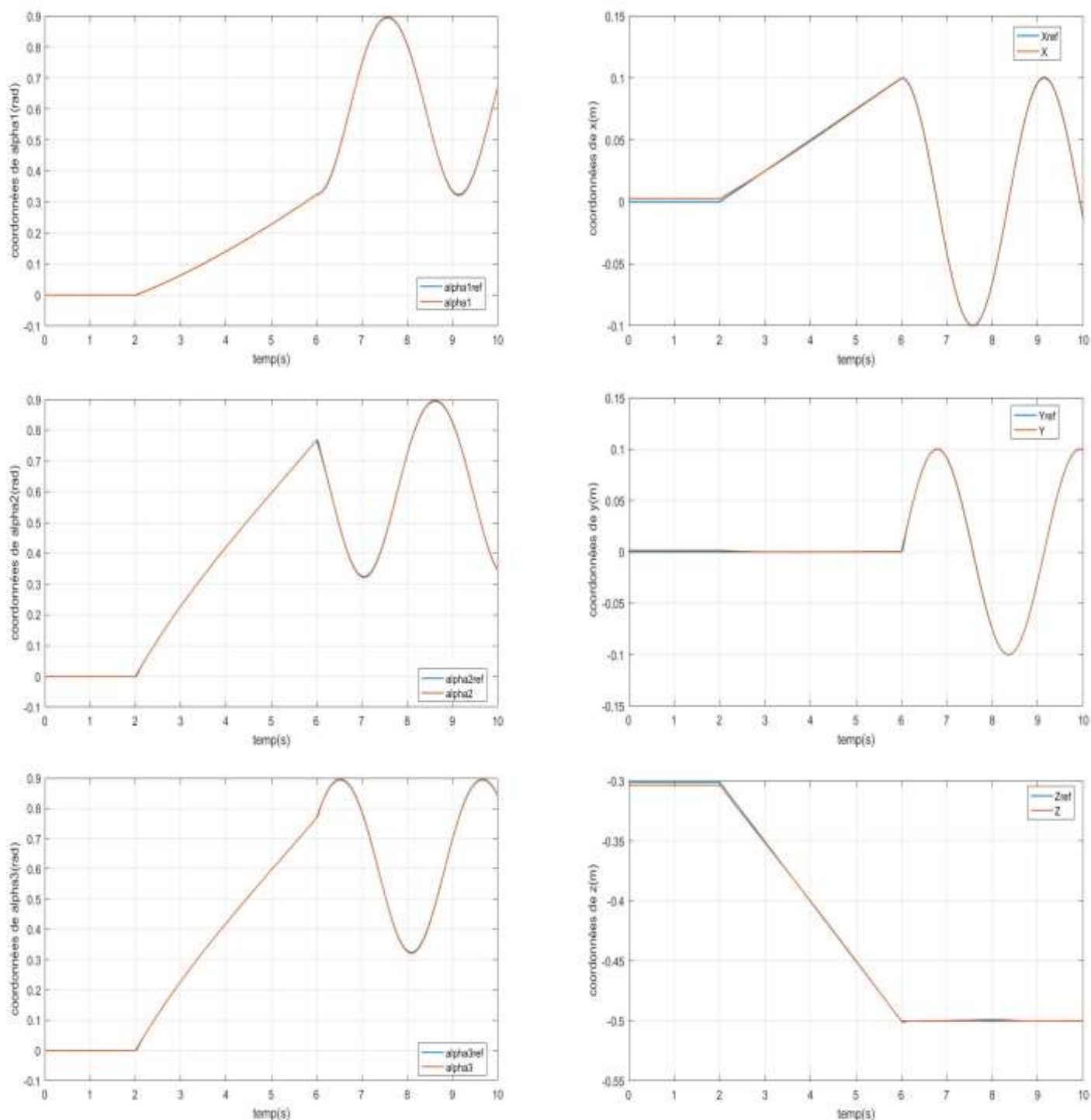


Figure III.39: Evolution temporelle des coordonnées (X, Y, Z) et des coordonnées (α_1 , α_2 , α_3) de poursuite de la trajectoire.

La figure III.39 : présente les consignes et les réponses de la trajectoire après l'adaptation de chaque articulation en fonction de temps et assurée avec des petites erreurs.

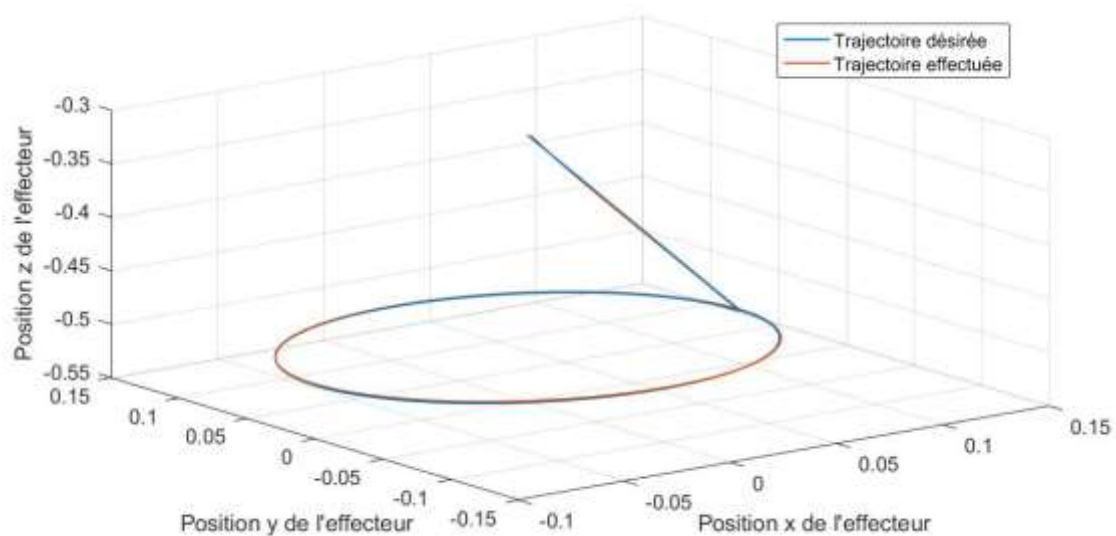


Figure III.40: Trajectoire circulaire dans le plan (X, Y, Z).

Figure III.40 : présente une trajectoire circulaire effectuée et une trajectoire désirée après l'adaptation (les consignes et les réponses).

III.6 Commentaire

La trajectoire de la commande neuro-floue adaptative est fluide et précise, car elle est capable de s'adapter aux changements dans l'environnement ou les objectifs à atteindre. De plus, elle est capable d'optimiser sa performance en temps réel, ce qui permettra d'obtenir des résultats plus rapides et plus efficaces.

Conclusion

La commande neuro-floue appliquée au robot Delta nous a montré son efficacité. Nous avons testé la simulation avec trois trajectoires, ce qui nous a montré que cette commande arrive à suivre les consignes avec des erreurs négligeables et qui donne des résultats très satisfaisants.

Conclusion générale

Conclusion générale

Ce mémoire de fin d'études s'est concentré sur l'application de la commande neuro-floue adaptative pour améliorer le contrôle d'un robot Delta. Nous avons exploré les concepts de la logique floue et de la neuro-floue, puis avons présenté le robot Delta et son modèle géométrique et cinématique. Enfin, nous avons utilisé le modèle géométrique pour effectuer des simulations et tracer les trajectoires du robot Delta à l'aide de MATLAB.

La commande neuro-floue adaptative s'est avérée être une méthode prometteuse pour le contrôle des robots Delta. Elle permet d'exploiter les avantages de la logique floue et des réseaux de neurones artificiels pour ajuster en temps réel les paramètres de commande du robot en fonction des variations et des incertitudes de l'environnement. Cette approche offre ainsi la possibilité d'améliorer la précision, la robustesse et la flexibilité du contrôle du robot Delta.

Les simulations réalisées ont démontré l'efficacité de la commande neuro-floue adaptative pour générer et suivre des trajectoires précises pour le robot Delta. En utilisant les équations géométriques du robot, nous avons pu définir des trajectoires désirées dans l'espace de travail du robot. Grâce à la commande neuro-floue adaptative, le robot Delta a été en mesure de suivre ces trajectoires avec précision et stabilité, en tenant compte des contraintes physiques et des variations environnementales.

L'amélioration de la trajectoire du robot Delta revêt une importance cruciale dans de nombreuses applications, notamment l'assemblage, la manipulation d'objets, la soudure et bien d'autres tâches industrielles. Une trajectoire précise permet d'optimiser l'efficacité et la sécurité des opérations du robot, réduisant ainsi les erreurs et les temps d'exécution.

Cependant, il est essentiel de noter que l'obtention de trajectoires optimales pour un robot Delta nécessite une modélisation précise du robot et une calibration appropriée des paramètres de contrôle. De plus, la commande neuro-floue adaptative doit être adaptée aux spécificités du robot Delta pour garantir une performance optimale.

Il convient cependant de noter que ce mémoire présente certaines limites et ouvre des possibilités d'amélioration. Les simulations ont fourni des résultats encourageants, mais des expérimentations réelles sur un robot Delta physique seraient nécessaires pour valider et confirmer ces résultats. De plus, des recherches supplémentaires pourraient être menées pour optimiser les paramètres du contrôleur neuro-flou adaptatif, explorer d'autres méthodes de commande et évaluer les performances du robot Delta dans des environnements complexes et dynamiques.

BIBLIOGRAPHIE

BIBLIOGRAPHIE

- [1] : B. Bouchon-Meunier, « La logique floue », Que sais-je ? PUF.
- [2] Aid Lahcene, “ Modélisation et simulation du confort dans un bâtiment intelligent par le formalisme DEVS”, Thèse de doctorat en sciences, Université des Sciences et de la Technologie Mohamed Boudiaf d’Oran, 2016.
- [3] MEDJROUD Sarah « Un système de navigation à base de la logique floue : Application sur un système multi robots » Mémoire de Master en Informatique, Université Ibn Khaldoun de Tiaret, 2018.
- [4] BOUAFIA Abdelouahab, « Techniques de commande prédictive et floue pour les systèmes d’électronique de puissance : application aux redresseurs a MLI », thèse de doctorat, université Sétif, 2010.
- [5] KHAMMAR Fatma, « Contribution à l’Etude de la Commande d’une Machine Asynchrone : Approches Intelligentes », Thèse de doctorat en sciences, Université Badji Mokhtar d’Annaba, 2017.
- [6] ELBIA Youcef, « Commande Floue Optimisée d’une Machine Asynchrone à Double Alimentation et à Flux Orienté », Mémoire Ingénieur d’Etat en Electrotechnique Option Commande Electrique, Université El-hadj Lakhdar de Batana, 2009.
- [7] M. Fodil, « Commande Adaptative par Logique Floue de la Machine Asynchrone », mémoire de magister, Université de M’sila, 2007.
- [8] M. MEKKI Idriss, « Commande vectorielle de la machine asynchrone à double alimentation MADA à base d’un régulateur flou », mémoire de master, université de M’sila, 2014.
- [9] Najet Aouchar Et Chafika Bekhaled : ‘Application des systèmes hybrides neuro-flous au diagnostic des huiles de transformateurs’ mémoire de fin d’étude d’ingénieur d’état en électronique, école nationale polytechnique, juin 2005.
- [10] Daniel. R (2006), Contribution à la surveillance des systèmes de production en utilisant les techniques de l’intelligence artificielle. Rapport d’habilitation à diriger des recherches, université de Franche-Comté de Besançon.
- [11] Mordjaoui Mourad, MODELISATION DES EFFETS ELECTROMAGNETIQUES « Apport de la logique floue et neuro-floue », Thèse de doctorat en sciences, Université De Batna ,2008 ,178 p

- [12] Daikh. Fatma Zohra, Hamadouche. Mohamed Amine” Commande neuro-flou inverse des systèmes non linéaires” Article scientifiques et publications, UNIVERSITE ZIANE Achour-Djelfa.
- [13] R Mellah : « Contribution à la commande adaptative neuro-floue, application à la robotique », Thèse de Doctorat, Université des Sciences et de la Technologie Houari Boumediene, Alger, Mai 2006.
- [14] KHELIFI Oualid et AZZOUG Larbi, « Modélisation et Commande d’un Robot Delta », projet de fin d’études : automatique, Alger : ENP, 2010, p105
- [15] BOUDEHANE Fadal Allah et LABCHRI Amine, « Modélisation et Commande d’Ordre Fractionnaire d’un Robot Parallèle de Type Delta ISIR88 », projet de fin d’études : Automatique, Alger : ENP, 2020
- [16] Abdeldjalil BENSALID Cherif MOKRAOUI, « Modélisation et Commande d'un Robot de Type Delta de la Simulation à l'Expérimentation », projet de fin d’études : automatique, Alger : ENP, 2019
- [17] Clavel, Raymond, « Conception d’un robot parallèle rapide à 4 degrés de liberté ». Thèse de doctorat, Ecole Polytechnique Fédérale de Lausanne - EPFL, 1991.