

DOMAINE : SCIENCES ET TECHNOLOGIES
FILIERE : automatique
SPECIALITE : Automatique et systèmes



FACULTE : TECHNOLOGIE
DEPARTEMENT : GENIE ELECTRIQUE
N° D'ORDRE:

MEMOIRE DE FIN D'ETUDE EN VUE DE L'OBTENTION DU
DIPLOME DE MASTER

Présenté par :
MEZID Mohamed

THEME

**Approximation du modèle géométrique inverse d'un
robot manipulateur par les réseaux de neurones
artificiels**

Soutenu le 26 /06/ 2019 devant le jury composé de :

KHETTAB Khoutir	UNIVERSITE MOHAMED BOUDIAF - M'SILA	Président
AIB Abdelghani	UNIVERSITE MOHAMED BOUDIAF - M'SILA	Encadreur
BOUKHALFA Abdelouahab	UNIVERSITE MOHAMED BOUDIAF - M'SILA	Examineur

Résumé :

Dans ce travail, nous présentons un procédé pour l'approximation du modèle géométrique inverse d'un robot manipulateur série par un réseau de neurones artificiel, tout en évitant les problèmes classiques liés au calcul du modèle géométrique inverse par des méthodes donnant une solution forme fermée (une solution explicite). Notre procédé utilise le modèle géométrique direct calculé par la convention de Denavit-Hartenberg, pour générer la base de données d'apprentissage en identification inverse par un réseau de neurones.

Mots clés : Model géométrique direct, Model géométrique inverse, réseau de neurones artificiels

Abstract:

In this paper, we present a procedure for approximating the inverse geometric model of a serial manipulator robot by an artificial neural network, while avoiding the classical problems associated with calculating the inverse geometric model by methods giving a closed shape solution (an explicit solution). Our procedure uses the direct geometric model calculated by the Denavit-Hartenberg Convention, to generate the learning database in reverse identification with a neural network.

Keywords: Direct geometric model, geometric model, Neural network

ملخص:

في هذه الورقة، نقدم إجراء لتقريب النموذج الهندسي العكسي لروبوت مناوّر تسلسلي بواسطة شبكة عصبونية اصطناعية، مع تجنب المشكلات الكلاسيكية المرتبطة بحساب النموذج الهندسي العكسي بطرق تعطي حلاً لشكل مغلق (حل واضح). يستخدم الإجراء الخاص بنا النموذج الهندسي المباشر المحسوب بواسطة اتفاقية دينايفيت-هارتمبرك، لإنشاء قاعدة بيانات التعلم في تحديد عكسي بواسطة شبكة عصبونية.

كلمات مفتاحية: النموذج الهندسي العكسي، النموذج الهندسي المباشر، شبكة عصبونية

Remerciements

Avant de présenter ce travail, je tiens à remercier **DIEU** tout puissant, de m'avoir permis d'arriver à ce niveau, et aussi pour m'avoir donné beaucoup de patience et de courage sans oublier mes parents qui n'ont lésiné sur aucun problème pour apporter toute l'aide nécessaire pour atteindre ce niveau qui me permettra d'assurer mon avenir.

A travers cette mémoire je tiens à présenter mes sincères remerciements et mes profondes reconnaissances à mon encadreur **AIB Abdelghani** pour sa patience, sa disponibilité et surtout ses judicieux Conseils, qui ont contribué à alimenter ma réflexion.

Je tiens à remercier tous les enseignants au département de génie électrique de l'université **Mohamed Boudiaf** à m'sila pour leurs aides, soutiens et leurs conseils ainsi que tout le staff administratif du département.

Merci à tous ceux et celles qui ont contribués de près ou de loin dans l'accomplissement de ce travail.

Dédicace

A ma bien-aimée mère Mahjouba, qui m'a entouré d'amour, d'affection et qui fait tout pour ma réussite, que dieu la garde ;

Mon père, Mohamed Lemin qui m'a aidé à devenir ce que je suis aujourd'hui, que dieu le garde et le protège ;

Mon grand frère Deddah que je pris comme mon idole.

Ma grande sœurs mghayli .

Ma grande sœur beybih.

Ma petite sœur oumama, que j'aime profondément.

Mon petit frère que j'adore houdifa.

Ma petite mariem la prunelle de mes yeux.

Mon petit frère Abdala, Aucune dédicace ne saurait exprimer l'amour l'estime et le respect que j'ai toujours eu pour toi.

Mon petit frère cheikh, mon conseiller, et ami fidèle, qui m'a assisté dans les moments difficiles et m'a pris doucement par la main pour traverser ensemble des épreuves pénibles....

A ma grande sœur, Khadija le bonheur d'avoir une grande sœur sur laquelle on peut compter.

Ma sœur, Bebouta la généreuse, merci pour tous les sacrifices consentis et ses précieux conseils.

A mon deuxième père Med abdallahi Memin qui m'a offert l'occasion d'étudier en Algérie que dieu le garde.

A mes chers adorable frères Memin, Brahim Salem, Baba.

A Tout le monde.

Puisse Dieu vous garder, éclairer votre route et vous aider à réaliser à votre tour vos vœux les plus chers.

Liste des matières

Table des matières

Introduction générale.....	1
----------------------------	---

Chapitre I Les réseaux de neurones Artificiels

I.1 Introduction	2
I.2. Les neurones biologiques	2
I.3 Structure des neurones :	2
I.3.1 Le corps cellulaire :	3
I.3.2 Les dendrites :	3
I.4 Neurone formel :	4
I.5 L'utilisation des réseaux de neurones :	5
I.6 Apprentissage :	6
I.6.1 L'apprentissage supervisé :	6
I.6.2 L'apprentissage non supervisé :	6
I.6.3 L'apprentissage hybride :	7
I.8 Algorithme de retro Propagation du Gradient :	7
I.8.1 Le principe :	7
I.8.2 L'algorithme :	7
I.8.2.1 Définition du réseau :	7
I.8.2.2 Les étapes de l'algorithme :	8
I.9 Architecture des réseaux de neurone :	9
I.10 Réseaux Propagation vers l'avant « Feed-Forward » :	9
I.10.1 Perceptron monocouche SLP :	10
I.10.2 Perceptron multicouches PMC :	10
I.11 Structure d'interconnexion des RNA :	11
I.11.1 Réseau multicouche :	11

I.11.2 Réseau à connexions locales :	12
II.11.3 Réseau à connexions récurrentes :	12
I.11.4 Réseau à connexions complexes :	13
I.12 Architecture des RNA :	13
I.12.1 Réseaux statiques :	13
I.12.2 Réseaux dynamiques :	14
I.12.3 Réseaux auto-organisés :	14
I.12 Application des réseaux de neurones :	14
I.14 Avantages et Inconvénients des réseaux de neurones	14
I.14.1. Avantages des réseaux de neurones	15
I.14.2. Inconvénients des réseaux de neurones	15
I.15 Conclusion :	15

Chapitre II Modélisation des robots manipulateurs

II.1 Introduction	16
II.2 Classification et fonctions des robots	16
II.2.1 Classification des robots	16
II.2.2 Fonctions des robots	16
II.3 Historique	16
II.4 La modélisation	18
II.5 Modélisation géométrique d'un robot manipulateur [2]	19
II.5.1 Modèle géométrique direct (MGD)	19
II.5.1.1 Positionnement et transformation homogènes	19
II.5.1.2 Rotation et représentation de la rotation(angles euler)	21
II.5.1.3 La convention de denavit-hartenberg pour le calcul de MGD	23
II.6 Modèle géométrique inverse (MGI)	25

II.6.1 Méthodes de calcul de la MGI	25
II.6.2 Les problèmes liés au calcul du modèle géométrique inverse	25
II.7 Les modèles géométriques direct et inverse dans la boucle de commande d'un robot	28
II.8 Conclusion	29

Chapitre III Approximation du modèle géométrique inverse par les réseaux de neurones artificiels

III.1 Introduction :	30
III.2 Identification des processus par réseaux de neurones	30
III.1.1 Identification directe	31
III.1.2 Identification inverse	32
III.2 Approximation du MGI du robot Mentor par Identification inverse	32
III.2.1 la méthode proposée pour l'approximation de MGI du robot Mentor	33
III.2.2 Calcul du MGD du robot Mentor [8]	33
III.2.3 Choix de la topologie et de l'architecture du réseau :	37
III.2.4 Remplissage de la base des données pour l'apprentissage :	37
III.3 Résultats d'approximation du model géométrique inverse par les réseaux de neurones artificiels	38
III.3.1 Résultats de l'architecture a :	38
III.3.2 Résultats de l'architecture B :	41
III.3.3 Architecture c :	45
III.4 Conclusion	49
Conclusion générale	50

Liste des figures

Liste des figures

Figure I.1 : Neurone biologique	3
Figure I.2 : Le modèle de McCulloch-Pitts	4
Figure I.3 : Quelques types de fonctions d'activations	5
Figure I.4: Exemple de réseau MLP à une couche cachée avec 5 entrées, 3 neurones dans la couche cachée, et quatre 4 sorties.....	8
Figure I.5 : Topologies des réseaux de neurones artificiels.....	9
Figure I.6: Perceptron entrée poids sortie	10
Figure I.7 : Exemple d'un réseau de type perceptron multicouches.....	11
Figure I.8 : Réseau multicouche classique	12
Figure I.9 : Réseau à connexions locales	13
Figure I.10 : Réseau à connexions récurrentes.....	14
Figure I.11 : Réseau à connexions complexes.....	15
Figure II.1 : Exemple du modèle cinématique.....	19
Figure II.2 trois rotations successives.....	21
Figure II.3 : Robot a structure ouverte simple.....	23
Figure II.4: exemple d'expressions des quatre paramètres de denavit-Hartenberg.....	24
Figure II.5 MGD et MGI dans une boucle de commande d'un robot.....	28
Figure III.1 Schéma d'identification directe d'un processus par réseau de neurones.	31
Figure III.2 Schéma d'identification inverse d'un processus avec un réseau de neurones.	32
Figure III.3 Identification inverse pour l'approximation du MGI du robot Mentor	33
Figure III.4 Image du robot Mentor.	33
Fig III.5 Les translations homogènes	34

Figure III.6 performance du réseau de neurones de l'architecture a	38
Figure III.7 Courbe de régression général du réseau de neurone pour l'architecture a.....	38
Figure III.8 les orientations réelles et approximées(ϑ_1)	39
Figure III.9 les orientations réelles et approximées(ϑ_2).....	39
Figure III.10 les orientations réelles et approximées(ϑ_3)	40
Figure III.11 performance du réseau de neurones de l'architecture b.....	41
Figure III.12 Courbe de régression général du réseau de neurone pour l'architecture b	41
Figure III.13 les orientations réelles et approximées(ϑ_1)	42
Figure III.14 les orientations réelles et approximées(ϑ_2).....	42
Figure III.15 les orientations réelles et approximées(ϑ_3)	43
Figure III.16 performance du réseau de neurones de l'architecture 3	44
Figure III.17 Courbe de régression général du réseau de neurone pour l'architecture c.....	44
Figure III.18 les orientations réelles et approximées(θ_1).....	45
Figure III.19 les orientations réelles et approximées(θ_2).....	45
Figure III.20 les orientations réelles et approximées(θ_3).....	46

Liste des tableaux

Liste des tableaux

Tableau I.1 : L'analogie entre les neurones biologiques et les neurones artificiels. . 5

Tableau III.1 paramètre de Denavit-Hartenberg.....35

INTRODUCTION GENERALE

Introduction générale

L'utilisation de manipulateurs dans l'industrie n'a cessé d'augmenter ces dernières années. Ils ont permis d'augmenter la production et la précision des mouvements effectués. La plus grande partie de ces manipulateurs est constituée des manipulateurs sériels dont la morphologie peut facilement être comparée à celle de l'être humain. Plus récemment, les manipulateurs parallèles sont apparus dans les domaines où l'on a besoin d'effectuer des mouvements rapides ou des placements très précis (simulateur de vol, usinage, chirurgie, etc.) ou transporter de lourdes charges. Un problème important non résolu lié à l'utilisation de ce type de mécanisme est la gestion des différentes solutions aux modèles géométriques inverse.

La capacité des réseaux de neurones à approximer des fonctions uniformément continues a été prouvée en plusieurs articles. Les réseaux de neurones, avec la capacité d'approximer une grande classe de fonctions non linéaires, fournissent une structure canonique et faisable pour la représentation des systèmes dynamiques non linéaires. Ainsi, les réseaux de neurones sont un outil puissant que ce soit pour améliorer des applications quand le modèle mathématique du système non linéaire est disponible mais imprécis, ou pour établir un modèle du système à partir des données expérimentales lorsqu'un modèle théorique est indisponible ou impraticable.

Le but de cette mémoire est de trouver une solution pour le problème du modèle géométrique inverse ; et pour la résolution de ce problème nous avons choisi les réseaux de neurones artificiels par leur capacité d'approximer une grande classe de fonctions non linéaires comme solution.

Cette mémoire commence par une introduction sur les réseaux de neurone artificiel.

Le deuxième chapitre est dédié à une présentation des différents modèles utilisés pour décrire le mouvement des articulations d'un bras manipulateur, le modèle couvre des aspects des modèles géométriques, modèles cinématique et les modèles dynamiques.

Le troisième chapitre constitue notre contribution à l'approximation du modèle géométrique inverse du robot mentor.

CHAPITRE I
LES RESEAUX DE NEURONES
ARTIFICIELLES

CHAPITRE I : LES RESEAUX DE NEURONES ARTIFICIELLES

I.1 Introduction

Historiquement les premiers modèles neuromimétiques ont été développés dans la mouvance cybernéticienne. Ils devaient permettre de valider et d'utiliser en robotique les premiers modèles neurobiologiques de la conscience. Les cybernéticiens tendaient en effet vers une conception matérialiste réductrice de la conscience (conception béhavioriste selon laquelle la conscience est issue d'une combinaison de boucles réflexes). L'adaptation informatique des principes organisationnels du cortex devait alors leur permettre de vérifier cette hypothèse.

Les réseaux de neurones sont des modèles mathématiques et informatiques, des assemblages d'unités de calculs appelés neurones formels, et dont l'inspiration originelle était un modèle de la cellule nerveuse humaine. Cet héritage de la neurobiologie forme une composante importante de la matière, et le souci de maintenir une certaine correspondance avec le système nerveux humain a animé et continue à animer une part importante des recherches dans le domaine.

Malgré cet héritage, l'essentiel des travaux d'aujourd'hui ont pour objet le réseau de neurones formels et non son corrélat neurobiologique. Vu comme des systèmes de calcul, les réseaux de neurones possèdent plusieurs propriétés qui les rendent intéressants d'un point de vue théorique, et fort utiles en pratiques[1].

I.2. Les neurones biologiques

Le cerveau humain contient près de 86 milliards de neurones, et il existe environ 200 types de neurones. Dans un neurone nous pouvons distinguer trois régions principales (figure I.1) : le corps cellulaire qui contient le noyau du neurone ainsi que la machine biochimique nécessaire à la synthèse d'enzymes ; les dendrites, qui se divisent comme les branches d'un arbre, recueillent l'information d'autres neurones et l'acheminent vers le corps de la cellule ; l'axone, généralement très long et unique[2] , il conduit l'information du corps cellulaire vers d'autres neurones avec qui il fait des connexions appelées synapses:

- Recevoir des signaux en provenance de neurones voisins.
- Intégrer ces signaux.
- Engendrer un influx nerveux.

- Le conduire.
- Le transmettre à un autre neurone capable de le recevoir.

I.3 Structure des neurones :

Un neurone est constitué de trois parties :

- Le corps cellulaire.
- Les dendrites.
- L'axone.

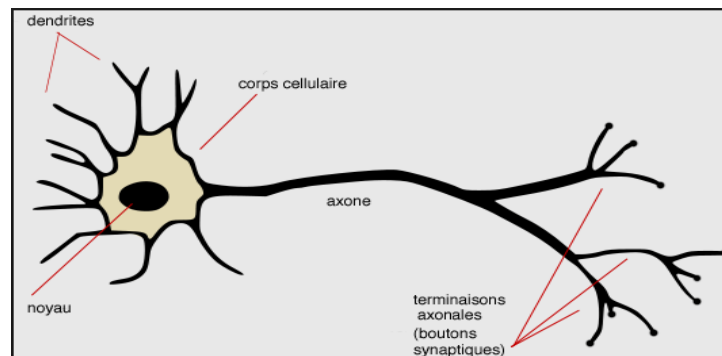


Figure I.1 : Neurone biologique

I.3.1 Le corps cellulaire :

Il contient le noyau du neurone et effectue les transformations biochimiques nécessaires à la synthèse des enzymes et des autres molécules qui assurent la vie du neurone, il a une forme pyramidale ou sphérique dans la plupart des cas.

I.3.2 Les dendrites :

Chaque neurone possède une « chevelure » de dendrites. Celles-ci sont de fines extensions tubulaires, de quelques dixièmes de microns. De diamètre et d'une longueur de quelques dizaines de microns. Elles sont les récepteurs principaux du neurone pour capter les signaux qui lui parviennent.

I.3.3 L'axone :

L'axone, qui est à proprement parler la fibre nerveuse, sert de moyen de transport connectés les uns aux autres suivant des répartitions spatiales complexes, les connexions entre deux neurones se font en des endroits appelés synapses où ils sont séparés par un petit espace synaptique de l'ordre d'un centième de micron.

D'une façon simple, on peut dire que le soma du neurone traite les courants électriques qui lui proviennent de ses dendrites, et qu'il transmet le courant électrique (sous forme

d'impulsions chaque une de durée d'environ 1ms et une amplitude d'environ 100mv) résultant de ce traitement aux neurones auxquels il est connecté par l'intermédiaire de son axone[3].

Le schéma classique présenté par les biologistes est celui d'un soma effectuant une sommation des influx nerveux transmis par ses dendrites Si la sommation dépasse un certain seuil, le neurone répond par un influx nerveux au potentiel d'action qui se propage le long de son axone, Si la sommation est inférieure à ce seuil, le neurone reste inactif[3]

I.4 Neurone formel :

Par définition un neurone formel est une fonction algébrique non linéaire, paramétrée, à valeurs bornées.

McCulloch et Pitts ont proposés une unité à seuil binaire comme un modèle neuronal diagramme de McCulloch-Pitts est montré dans la Figure suivante :

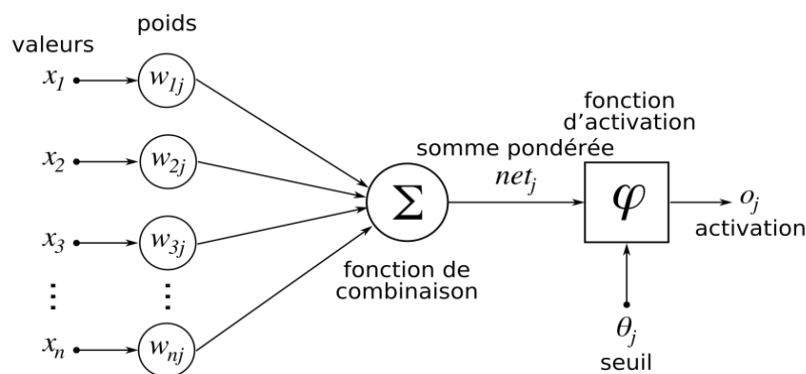


Figure I.2 : Le modèle de McCulloch-Pitts

Ce modèle mathématique calcule la somme pondérée de ses n entrées, $j=1,2,\dots,n$, Le nœud est caractérisé par un seuil μ , il génère une sortie de « 1 » si cette somme est supérieure au seuil μ , et autrement une sortie de « 0 » Mathématiquement.

$$v = \begin{cases} 1 & \text{Si } \sum_{i=1}^n w_i x_i \geq \mu \\ 0 & \text{Si } \sum_{i=1}^n w_i x_i < \mu \end{cases} \quad (I.1)$$

Où $F(.)$ est la fonction d'activation, et w_j est le poids synaptiques associé à la sortie, parfois, on considère le seuil m comme un autre poids $w_0 = -\mu$ et qui est connecté à un neurone avec une entrée constante, $x_0 = 1$.

Les poids positifs correspondent à des synapses excitatrices, tandis que les poids négatifs correspondent aux synapses inhibitrices. Il y a une analogie évidente (Tableau I-1) avec les neurones biologiques :

Neurones biologiques	Neurones artificiels
Synapses	Connections pondérées
Axons	Sorties
Dendrites	Entrées

Tableau I.1 : L’analogie entre les neurones biologiques et les neurones artificiels.

Le modèle de McCulloch et Pitts a été généralisé par de nombreuses façon, une généralisation évidente et celle d'utiliser une autre fonction d'activation que la fonction à seuil, *e.g.*, la fonction logistique, tangente hyperbolique, ou bien Gaussienne, qui son montrées dans la (Figure I.3).La fonction sigmoïde est de loin la plus utilisée dans les RNA, c'est une fonction qui est caractérisée par sa propriété asymptotique [4].

La fonction sigmoïde standard est la fonction logistique et elle est définie par :

$$G(x) = 1/(1 + \exp(-bx))$$

Où b est le paramètre de pente.

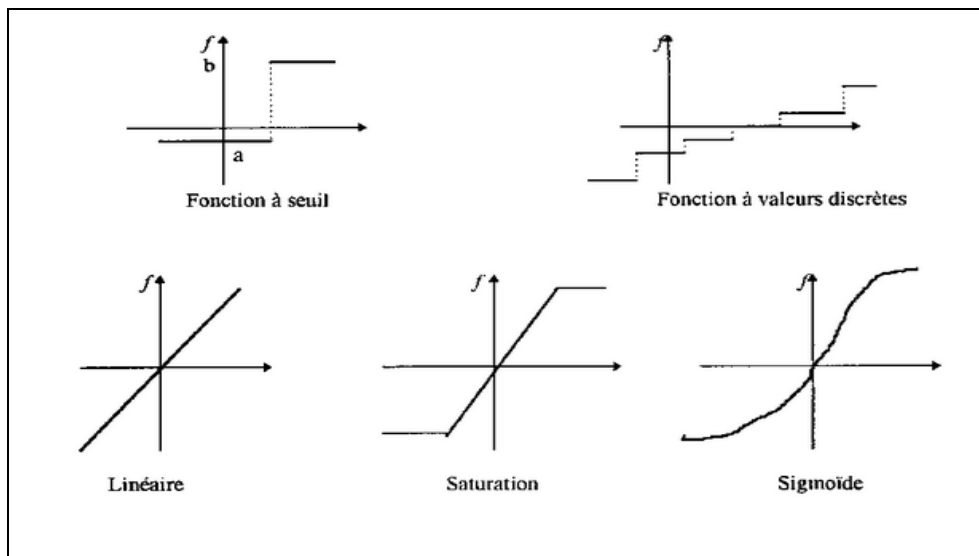


Figure I.3 : Quelques types de fonctions d'activations[4]

I.5 L'utilisation des réseaux de neurones :

Les réseaux de neurones ont une remarquable faculté à donner un sens, extraire des règles et des tendances à partir de données compliquées, bruitées et imprécises. Ils peuvent s'utiliser pour extraire des modèles et détecter des tendances reposant sur des

fonctions mathématiques compliquées qui sont trop difficiles, voire impossible, à modéliser à l'aide de techniques analytiques ou paramétriques traditionnelles.

L'une des propriétés intéressantes des réseaux de neurones est qu'ils savent prévoir avec précision des données qui ne faisaient pas partie des données d'apprentissage, un processus connu sous le nom de généralisation, Compte tenu de ces caractéristiques et de leur domaine d'application étendu, les réseaux de neurones sont particulièrement bien adaptés à l'application de problématiques concrètes dans les domaines de la recherche scientifique, commerciale et industrielle. Ci-dessous, un certain nombre de domaines dans lesquels les réseaux de neurones ont été appliqués avec succès [5] :

- Traitement du signal
- Prédiction
- Maîtrise des processus
- Robotique
- Classification
- Prétraitement des données
- Reconnaissance de formes
- Analyse de l'image et synthèse vocale
- Diagnostiques et suivi médical
- Marché boursier et prévisions
- Demande de crédits ou de prêts immobiliers

I.6 Apprentissage :

La capacité à apprendre un trait fondamental de l'intelligence, d'un autre côté ce que veut dire l'apprentissage est parfois difficile à décrire, le processus d'apprentissage, dans le contexte des réseaux de neurones peut être vu comme un problème ajustement et d'adaptation de l'architecture et des poids du réseau de neurone de façon que le RNA exécute efficacement une tâche spécifique. Au lieu de spécifier un ensemble de règles, les réseaux de neurones apprennent automatiquement à partir de la collection d'exemples, et c'est ce qui les rend très attractives[6].

Il y a trois types de paradigmes d'apprentissage :

- Supervisé
- Non supervisé
- Hybride

I.6.1 L'apprentissage supervisé :

C'est actuellement le mode d'apprentissage le plus couramment utilisé. Son principe est élémentaire : on soumet au réseau à un grand nombre d'exemples pour lesquels l'entrée commise par le réseau.

Le plus répandu des algorithmes d'apprentissage supervisé est l'algorithme de rétro propagation du gradient d'erreur qui, appliqué aux réseaux Multi Couches[7].

I.6.2 L'apprentissage non supervisé :

Contrairement aux modes supervisés, seule une base d'entrées est fournie au réseau. Celui-ci doit déterminer lui-même ses sorties en fonction des similarités détectées entre les différentes entrées, c'est-à-dire en fonction d'une règle d'auto-organisation, Il explore la structure des données, ou encore la corrélation entre les modèles d'entrée et les organise à partir de cette corrélation en des catégories.

I.6.3 L'apprentissage hybride :

Il combine l'apprentissage supervisé et non supervisé. Dans de nombreux problèmes on ne dispose pas de l'information nécessaire à la construction d'une base d'apprentissage complète. On ne dispose souvent que d'une information qualitative permettant l'évaluation de la réponse calculée, sans pour autant connaître la réponse la plus adaptée.

I.7 Algorithme d'apprentissage :

Pour chaque type de réseaux de neurones artificiels, il existe des méthodes ou des algorithmes d'apprentissage permettant une modification des poids et de biais.

On retrouve plusieurs algorithmes d'apprentissage tels que l'algorithme du perceptron, la règle d'apprentissage de Hebb, la règle d'apprentissage de Widrow-Hoff (règle Delta) ainsi que des algorithmes généralisés de Widrow-Hoff pour les réseaux multicouches (rétropropagation du gradient de l'erreur...).

I.8 Algorithme de retro Propagation du Gradient :**I.8.1 Le principe :**

La rétro-propagation du gradient consiste à propager « à l'envers » (de la couche de sortie vers la couche d'entrée) l'erreur obtenue sur les exemples de la base d'apprentissage. On utilise pour cela l'erreur quadratique (le carré de la différence entre ce qu'on obtient et ce qu'on désire).

Si on calcule la dérivée partielle de l'erreur quadratique par rapport aux poids des connexions (gradient), il est possible de déterminer la contribution des poids à l'erreur

générale, et de corriger ces poids de manière à se rapprocher du résultat souhaité. La correction par itération corrige plus ou moins les poids par l'intermédiaire d'un coefficient η [8].

I.8.2 L'algorithmme :

I.8.2.1 Définition du réseau :

Soit un réseau multicouche défini par :

- Une couche d'entrée à m cellules d'entrées $x_i = e_i$ (Il ne s'agit pas de neurones, ces cellules présentent simplement les entrées e_i au réseau).
- Une couche de sortie p neurones d'activation Z_k
- Une couche cachée a n neurones d'activation y_j
- $n * m$ connexions entre la couche d'entrée et la couche cachée, chacune pondérée par v_{ij}
- $n * p$ connexions entre la couche cachée et la couche de sortie, chacune pondérée par w_{jk} .

I.8.2.2 Les étapes de l'algorithmme :

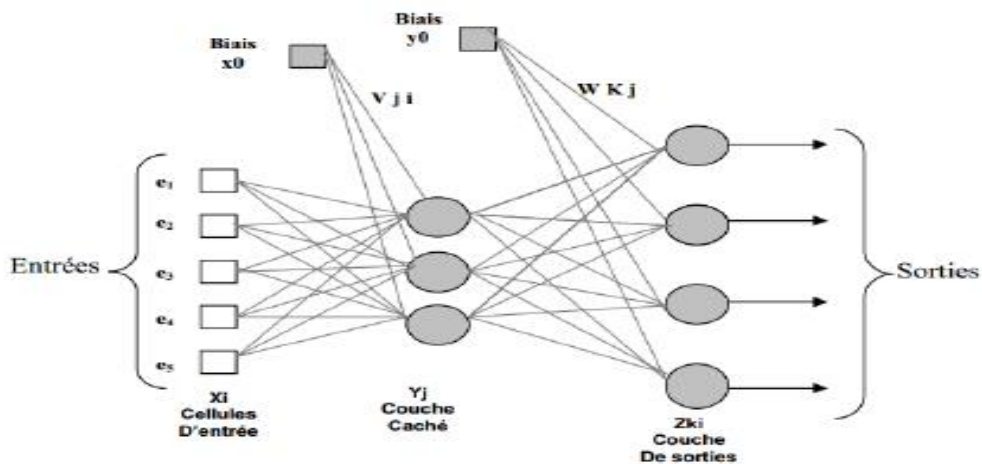


Figure I.4: Exemple de réseau MLP à une couche cachée avec 5 entrées, 3 neurones dans la couche cachée, et quatre 4 sorties

- ETAPE 1: Initialisation au hasard ou aléatoire des poids des connexions et v_{ij} et w_{jk}
- ETAPE 2: Propagation des entrées $x_i = e_i$.

On propage vers la couche cachée :

$$y_j = f \left(\sum_{i=1}^m x_i v_{ij} + x_0 \right) \tag{I.2}$$

Puis de la couche cachée vers la couche de sortie :

$$z_k = f \left(\sum_{i=1}^n y_i w_{ki} + y_0 \right) \quad (\text{I.3})$$

Les valeurs x_0 et y_0 sont des biais, f est la fonction d'activation qu'on a choisie où on a défini notre réseau MLP.

• ETAPE 3: Rétro-propagation de l'erreur pour chaque neurone de la couche de sortie, on calcule l'erreur, c'est-à-dire la différence entre la sortie désirée S_k et la sortie réelle (obtenue) Z_k .

$$E_k = Z_k (1 - Z_k) (S_k - Z_k) \quad (\text{I.4})$$

On propage cette erreur sur la couche cachée ; l'erreur de chaque neurone de la couche cachée est donnée par :

$$F_j = y_j (1 - y_j) \sum_{k=1}^P w_{kj} E_k \quad (\text{I.5})$$

• ETAPE 4: Correction des poids des connexions : Il reste maintenant la modification des poids des connexions et aussi les biais.

- Entre la couche d'entrée et la couche cachée :

$$\begin{cases} \Delta v_{ji} = \eta \cdot y_j \cdot F_j \\ \Delta x_0 = \eta \cdot F_j \end{cases} \quad (\text{I.6})$$

-Entre la couche cachée et la couche de sortie :

- η Etant un paramètre qu'il reste à déterminer.

-BOUCLER à l'étape 2 jusqu'à un critère d'arrêt à définir.

I.9 Architecture des réseaux de neurone :

On classe généralement les réseaux de neurones en deux catégories : les réseaux faiblement connectés à couches que l'on appelle des réseaux "feedforward" ou réseaux directs et les réseaux fortement connectés que l'on appelle des réseaux "récurrents". Dans ces deux configurations, on retrouve des connexions totales ou partielles entre les couches. La figure Suivante présente les différentes topologies d'un Réseau de neurones artificiels[9].

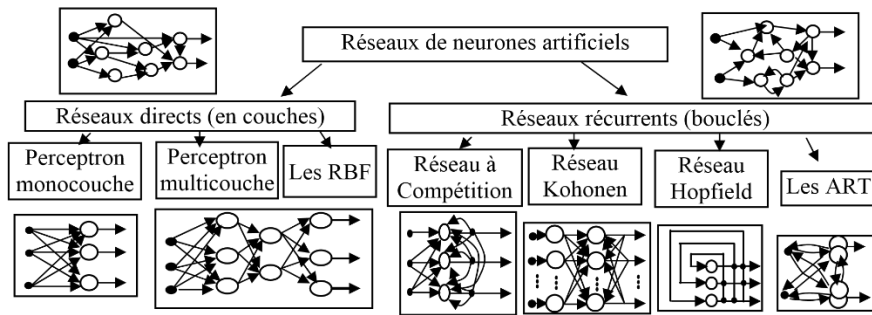


Figure I.5 : Topologies des réseaux de neurones artificiels

I.10 Réseaux Propagation vers l'avant « Feed-Forward » :

Un réseau de neurone "Feed-forward" appelé aussi réseau non bouclé ou unidirectionnels ou statique est présenté par un ensemble de neurones connectés entre eux telle que l'information circulant des entrées vers les sorties sans retour en arrière (dans un sens unique). Le calcul de la sortie se fait en propageant les calculs de la gauche vers la droite, Ces architectures sont les plus couramment utilisées.

I.10.1 Perceptron monocouche SLP :

Le perceptron est un réseau très simple grâce à sa composition d'une couche d'entrée et d'une couche de sortie sans avoir de couche cachée. Sa structure est montrée dans la (figure I.6), Différents neurones sont arrangés dans une couche d'entrées et ils sont connectés avec chaque neurone de la couche de sortie par l'intermédiaire des poids.

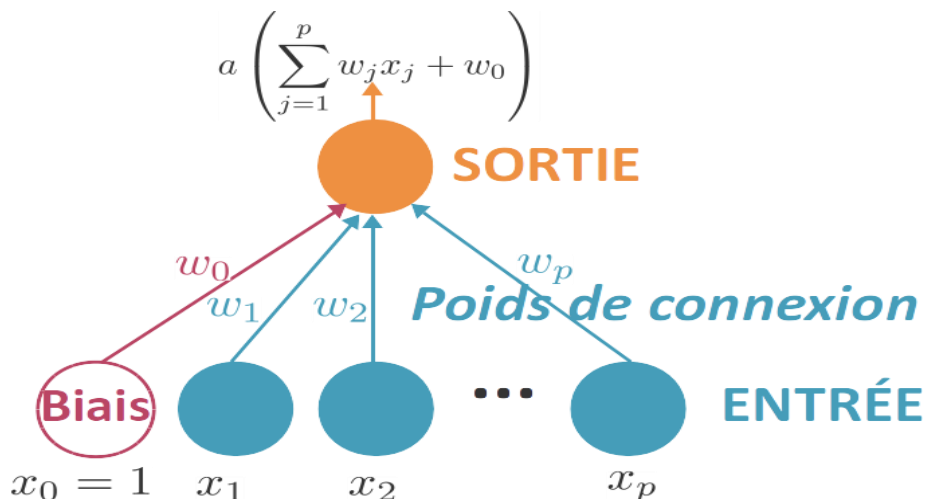


Figure I.6: Perceptron entrée poids sortie

I.10.2 Perceptron multicouches PMC :

Le PMC est une extension du perceptron monocouche qui dispose d'un nombre de couches quelconque et un nombre de neurones (ou d'entrées) par couche également quelconque. Les neurones y sont arrangés en couches successives : la première couche qui forme le vecteur des données d'entrée est appelée couche d'entrée tandis que la

dernière couche qui produit les résultats est appelée couche de sortie. Toutes les autres couches qui se trouvent au milieu sont appelées couches cachées. La (figure I.7) donne un exemple d'un réseau contenant une couche d'entrée, deux couches cachées et une couche de sortie. Les neurones sont reliés entre eux par des connexions pondérées.

Les neurones de la couche d'entrée sont connectés uniquement à la couche suivante tandis que les neurones des couches cachées ont la particularité d'être connecté à tous les neurones de la couche précédente et de la couche suivante, par contre il n'y a pas de connexions entre les neurones elle-même dans n'importe quelle couche.

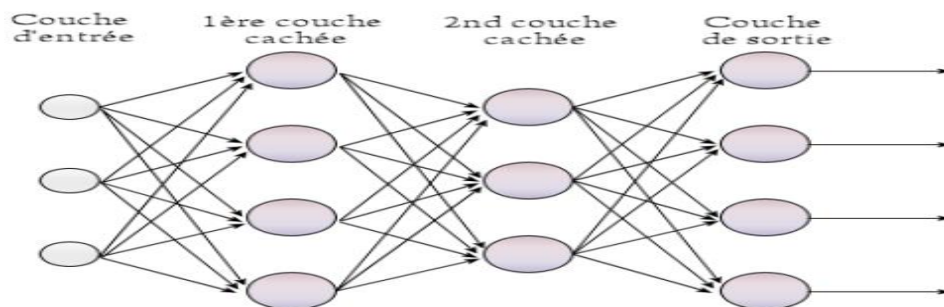


Figure I.7 : Exemple d'un réseau de type perceptron multicouches

Le choix du nombre de couches cachées dépend généralement de la complexité du problème à résoudre, en théorie une seule couche cachée peut être suffisante pour résoudre un problème donné mais il se peut que le fait de disposer de plusieurs couches cachées permette de résoudre plus facilement un problème complexe[10].

I.11 Structure d'interconnexion des RNA :

Les connexions entre les neurones qui composent le réseau décrivent la topologie du modèle. Elle peut être quelconque, mais le plus souvent il est possible de distinguer une certaine régularité.

I.11.1 Réseau multicouche :

Les neurones sont arrangés par couche. Il n'y a pas de connexion entre neurones d'une même couche et les connexions ne se font qu'avec les neurones des couches avales (figure I.8). Habituellement, chaque neurone d'une couche est connecté à tous les neurones de l'accouche suivante et celle-ci seulement. Ceci nous permet d'introduire la notion de sens de parcours de l'information (de l'activation) au sein d'un réseau et donc définir les concepts de neurone d'entrée, neurone de sortie. Par extension, on appelle couche d'entrée l'ensemble des neurones d'entrée, couche de sortie l'ensemble des neurones de sortie.

Les couches intermédiaires n'ayant aucun contact avec l'extérieur sont appelées couches cachées[11].

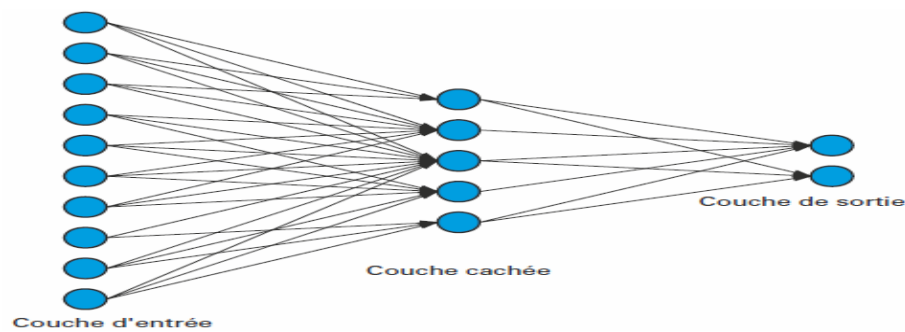


Figure I.8 : Réseau multicouche classique

I.11.2 Réseau à connexions locales :

Il s'agit d'une structure multicouche, mais qui à l'image de la rétine, conserve une certaine topologie. Chaque neurone entretient des relations avec un nombre réduit et localisée neurones de la couche avale (figure I.9). Les connexions sont donc moins nombreuses de neurones de la couche avale. Les connexions sont donc moins nombreuses que dans le cas d'un réseau multicouche classique[12].

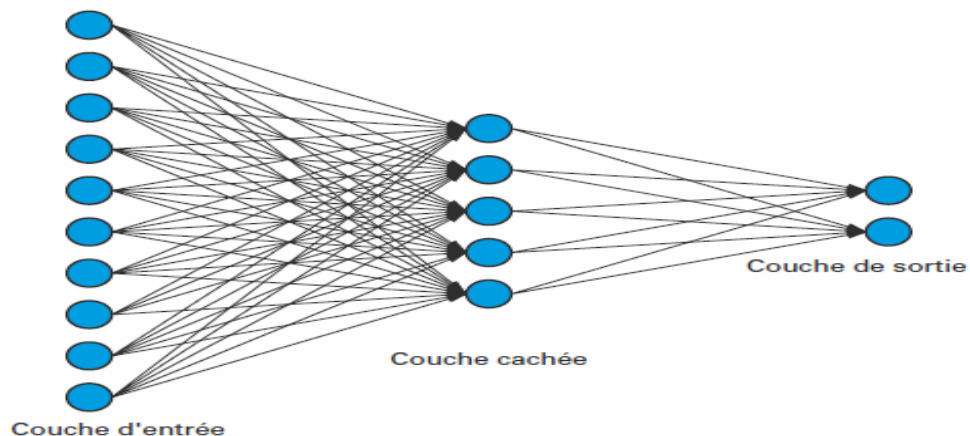


Figure I.9 : Réseau à connexions locales

II.11.3 Réseau à connexions récurrentes :

Un réseau de ce type signifie qu'une ou plusieurs sorties de neurones d'une couche aval sont connectées aux entrées des neurones de la couche amont ou de la même couche. Ces connexions récurrentes ramènent l'information en arrière par rapport au sens de propagation défini dans un réseau multicouche.

Les réseaux à connexions récurrentes sont des réseaux plus puissants, car ils sont séquentiels plutôt que combinatoires comme l'étaient ceux décrits précédemment (figure I.10). La rétroaction de la sortie vers l'entrée permet à un réseau de ce type de présenter un comportement temporel[13].

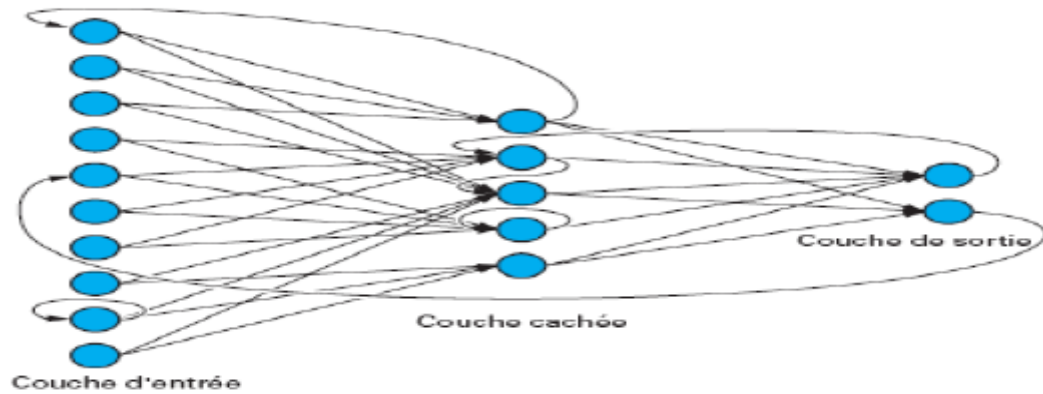


Figure I.10 : Réseau à connexions récurrentes

I.11.4 Réseau à connexions complexes :

Chaque neurone est connecté à tous les neurones du réseau y compris lui-même, c'est la structure d'interconnexion la plus générale.

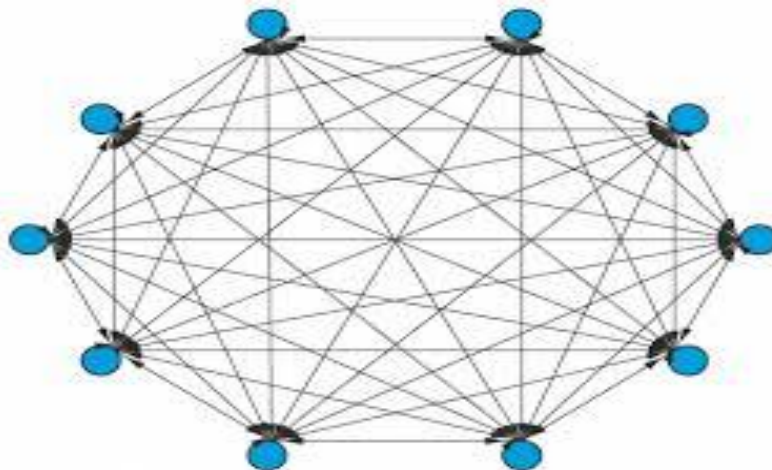


Figure I.11 : Réseau à connexions complexes

I.12 Architecture des RNA :

L'architecture d'un réseau de neurones définit son fonctionnement et joue un rôle important dans son comportement. Elle est en fonction de ses couches et de la structure de connexions de ses neurones ; ces paramètres permettent de distinguer les différentes classes et/ou types d'architecture neuronales.

I.12.1 Réseaux statiques :

Ce type de réseaux est organisé généralement en couches de neurones. Chaque neurone d'une couche reçoit ses entrées à partir des neurones de la couche précédente ou tout simplement de l'entrée du réseau. Dans tels réseaux il n'existe pas de feed-back (boucles de retour d'informations). Le traitement des données se fait en sens unique et le flux d'informations circule directement de la couche d'entrée à la couche de sortie ;

le traitement est donc réalisé en boucle ouverte. Ces réseaux peuvent être utilisés pour les problèmes de classification ou d'approximation des fonctions.

I.12.2 Réseaux dynamiques :

Les réseaux dynamiques ou bien récurrents, sont les réseaux pouvant comporter des boucles (feed-back) entre les neurones. En général, la sortie de chaque neurone peut être envoyée vers l'entrée de tous les autres neurones du réseau. Ainsi, ces boucles ramènent l'information en provenance de la couche de sortie sur la couche d'entrée simultanément avec le signal d'entrée présent au même instant.

Un réseau dynamique peut donner une sortie différente en lui présentant la même entrée des instants différents, contrairement aux réseaux statiques qui donnent la même sortie pour une même entrée.

Au début, ces réseaux étaient souvent utilisés pour les problèmes de classification et de mémorisation. Actuellement, dans plusieurs travaux la structure interne dynamique de ces réseaux est exploitée pour l'identification ou la commande des systèmes dynamiques.

I.12.3 Réseaux auto-organisés :

Les réseaux de neurones auto-organisés sont des réseaux qui changent leurs structures internes pendant l'apprentissage. Ainsi les neurones se regroupent typologiquement suivant la représentation des exemples. Ces réseaux sont des dérivées des modèles de Kohonen.

I.12 Application des réseaux de neurones :

Les réseaux de neurones servent dans aujourd'hui à toutes sortes d'applications dans divers domaines. On peut citer par exemples :

- Autopilotage des avions.
- Système de guidage des automobiles.
- Lecture automatique des chèques bancaires et d'adresses postales.
- Les réseaux de neurones sont utilisés aussi pour les systèmes de vision par ordinateur.
- Ils sont utilisés en robotique et en télécommunication.
- Ils sont aussi utilisés dans les domaines de finance.
- Ils sont utilisés pour le diagnostic médical.
- Production des systèmes de traitement signal et pour la synthèse de la parole.

I.14 Avantages et Inconvénients des réseaux de neurones

I.14.1. Avantages des réseaux de neurones

- Capacité de représenter n'importe quelle fonction, linéaire ou pas, simple ou complexe
- Faculté d'apprentissage à partir d'exemples représentatifs, par rétro-propagation des erreurs". L'apprentissage (ou construction du modèle) est automatique ;
- Résistance au bruit ou au manque de fiabilité des données ;
- Simple à manier, beaucoup moins de travail personnel à fournir que dans l'analyse statistique classique. Aucune compétence en mathématiques, informatique statistique requise ;
- Comportement moins mauvais en cas de faible quantité de données ;
- Pour l'utilisateur novice, l'idée d'apprentissage est plus simple à comprendre que les complexités des statistiques multi-variables.

I.14.2. Inconvénients des réseaux de neurones

- L'absence de méthode systématique permettant de définir la meilleure topologie du réseau et le nombre de neurones à placer dans la (ou les) couche(s) cachée(s) ;
- Le choix des valeurs initiales des poids du réseau et le réglage du pas d'apprentissage, qui jouent un rôle important dans la vitesse de convergence ;
- Le problème du sur-apprentissage (apprentissage au détriment de la généralisation)
- La connaissance acquise par un réseau de neurone est codée par les valeurs des poids synaptiques, les réseaux de neurones sont donc des boîtes noires où les connaissances sont inintelligibles pour l'utilisateur [15].

I.15 Conclusion :

Dans ce chapitre on a essayé de donner en bref une description sur les réseaux de neurones artificiels, les différents types d'architectures et de modèle qui existent. On a aussi présenté une définition de l'apprentissage des réseaux de neurones, puis une description générale de l'algorithme de rétro-propagation, les domaines d'utilisation des réseaux de neurones.

Contrairement aux méthodes classiques qui ont montré leurs limites, les réseaux de neurones ont montré leurs tendances à s'adapter à des problèmes complexes grâce à leur grande capacité de calcul et d'apprentissage. Ils sont l'objet d'utilisation dans les différents domaines tels que : La reconnaissance des formes, l'approximation des fonctions non linéaires, le traitement des images, le traitement de données...etc.

CHAPITRE II
Modélisation des robots
manipulateurs

CHAPITRE II : Modélisation des robots manipulateurs

II.1 Introduction

Si l'on définit la robotique comme le domaine d'une relation intelligente entre perception et action, la robotique et l'intelligence artificielle sont de toute évidence étroitement liées. La robotique constitue un challenge pour l'intelligence artificielle puisqu'elle lui permet de tester sur des objets réels et dans notre univers complexe des méthodes souvent mises au point sur des exemples de complexité restreinte (jeux ...).

Un robot est un système mécanique capable de se mouvoir (robot mobile) ou de déplacer son effecteur (robot manipulateur) afin de réaliser une tâche qui lui a été assignée. Les mouvements d'un robot sont engendrés par l'application de forces ou de couples qui sont transmis aux différents axes du robot par l'intermédiaire d'actionneurs (moteurs électriques...).

II.2 Classification et fonctions des robots[16]

II.2.1 Classification des robots

La classification la plus couramment utilisée en robotique prend en compte quatre générations de robots :

a. Robots de première génération : Ils sont programmables et ne réalisent que des tâches qui demandent peu de précision. Ils fonctionnent en boucle ouverte (sans feedback) dans des univers figés.

b. Robots de génération 1.5: Ils disposent de fonctions sensorielles relativement simples (capteurs de couple, de pression, de proximité, ...) et fonctionnent en boucle fermée (avec feedback). Ils sont capables d'une plus grande précision et d'actes réflexes. Ce sont essentiellement des robots-manipulateurs employés pour des travaux d'emballage, de tri, des robots-peintres, soudeurs et assembleurs.

c. Robots de deuxième génération : Ils disposent de fonctions de perception élaborées (vision par caméra, micros ultrasensibles, balayage laser ou sonar, stéréognosie, ...) qui nécessitent une interprétation et une coordination avec des effecteurs (roues, pinces, bras manipulateurs, lasers, ...).

d. Robots de troisième génération : Ce sont les plus évolués, ils n'existent pour l'instant qu'à l'état embryonnaire, l'aspect décisionnel prend chez eux une grande

importance, sans toutefois se confondre avec le contrôle direct des effecteurs ou des capteurs. Ils sont ainsi capables de déterminer de manière autonome leur comportement suivant l'état de l'environnement de façon à remplir les tâches qui leur sont assignées.

II.2.2 Fonctions des robots

Les robots intègrent quatre fonctions principales :

a. La fonction perception : Son rôle consiste à recueillir des informations sur l'environnement grâce à des capteurs (capteurs inertiels, caméras, détecteurs à ultra-sons, détecteurs de proximité, micros, ...) et à les traiter (prétraitement, intégration, codage, ...) pour en fournir une représentation exploitable par la fonction modélisation.

b. La fonction modélisation : Elle consiste à traiter les informations qui lui sont fournies par la fonction perception pour constituer et maintenir à jour un modèle interne du monde extérieur. Ce modèle est celui qui sera utilisé par la fonction décision.

c. La fonction action : Elle permet au robot d'agir sur le réel au moyen d'effecteurs (moteurs de déplacement, manipulateurs variés, laser, ...).

d. La fonction décision : Elle articule le fonctionnement des trois autres fonctions, détermine leurs arguments, lance leur action et analyse leurs résultats, en vue d'effectuer une tâche globale.

A ces quatre fonctions (on parle de boucle perception/modélisation/décision/action), vient se rajouter la fonctionnalité en général moins importante "la communication" : Son rôle consiste le plus souvent à permettre des échanges d'information entre l'utilisateur et le robot dans un langage facile à utiliser, par l'intermédiaire de divers terminaux (clavier, écran, interface vocale, ...).

II.3 Historique

1947 : Premier manipulateur électrique téléopéré.

1954 : Premier robot programmable.

1961 : Apparition d'un robot sur une chaîne de montage de General Motors.

1961 : Premier robot avec contrôle en effort.

1963 : Utilisation de la vision pour commander un robot.

La modélisation représente le lien entre l'espace opérationnel et l'espace de la tâche. Il existe 3 types de la modélisation :

- a. Modélisation géométrique.
- b. Modélisation cinématique.
- c. Modélisation dynamique.

II.4 La modélisation[17]

La modélisation du robot est nécessaire, elle nous permet d'avoir une vision globale de l'état et du comportement du robot manipulateur. Pour commander le robot, on utilise le modèle géométrique :

a. Le modèle géométrique

- ✓ Il permet la commande en position.
- ✓ Il considère le robot comme étant un système mécanique de $(n+1)$ corps rigides, reliés entre eux en série ou en parallèle par l'intermédiaire d'articulation.

On distingue deux types d'articulation :

- Articulation rotoïde : Caractérisée par un mouvement de révolution.
- Articulation prismatique : Caractérisée par un mouvement de translation.
- ✓ Chaque configuration géométrique du robot est définie par un ensemble des variables articulaire (**rotation, translation**) d'un corps par rapport au corps précédent.
- ✓ Les variables articulaires (rotation, translation) sont appelées coordonnées articulaires notées q_i .
- ✓ Le modèle géométrique du robot est l'expression en fonction des coordonnées articulaires, de l'attitude de l'organe terminal du robot par rapport à sa base.

N.B : Pour nos calculs, on a choisi la convention de Denavit-Hartenberg.

b. Modèle cinématique

Le modèle cinématique sert à analyser les robots en mouvement (vitesse et accélération).



Voici un exemple d'un modèle cinématique (figure II.1) :

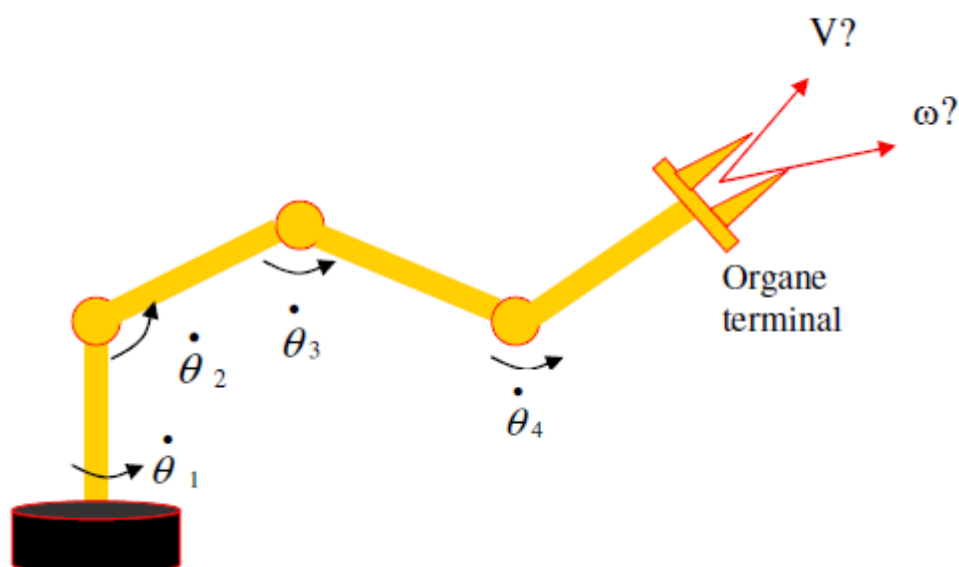


Figure II.1 : Exemple du modèle cinématique.

c. Modèle dynamique

Le modèle dynamique exprime les équations du mouvement du robot, qui permettent d'établir les relations entre les couples (forces exercés par les actionneurs) et les situations des articulations.

II.5 Modélisation géométrique d'un robot manipulateur [9]

II.5.1 Modèle géométrique direct (MGD)

II.5.1.1 Positionnement et transformation homogènes

Soit $\begin{bmatrix} P_x & P_y & P_z \end{bmatrix}$ les coordonnées cartésiennes d'un point P arbitraire, mesuré dans le repère $R_i [O_i, x_i, y_i, z_i]$:

$$P = \begin{bmatrix} P_x \\ P_y \\ P_z \\ 1 \end{bmatrix} \quad (\text{II.1})$$

-Représentation d'une direction (vecteurs libre) :

$$U = \begin{bmatrix} U_x \\ U_y \\ U_z \\ 0 \end{bmatrix} \quad \text{(II.2)}$$

-Représentation d'un plan :

le plan $\alpha x + \beta y + \gamma z = \delta$ est représenté par un vecteur $Q = [\alpha, \beta, \gamma, \delta]$.

Pour tout point appartenant à Q, $Q.P = 0$

$$Q.P = [\alpha, \beta, \gamma, \delta] \begin{bmatrix} P_x \\ P_y \\ P_z \\ 1 \end{bmatrix} \quad \text{(II.3)}$$

On définit la matrice de transformation homogène par :

$${}^i T_j = [{}^i s_j, {}^i n_j, {}^i a_j, {}^i P_j] = \begin{bmatrix} s_x & n_x & a_x & P_x \\ s_y & n_y & a_y & P_y \\ s_z & n_z & a_z & P_z \\ 0 & 0 & 0 & 1 \end{bmatrix} \quad \text{(II.4)}$$

- ${}^i s_j, {}^i n_j, {}^i a_j$ sont les vecteurs unitaires des axes X_j, Y_j et Z_j du repère R_j exprimé dans R_i

- P_j vecteur exprimant l'origine du repère R_j dans le repère R_i

On écrit aussi :

$${}^i T_j = \begin{bmatrix} {}^i A_j & {}^i P_j \\ 0 & 0 & 0 & 1 \end{bmatrix} = \begin{bmatrix} {}^i s_j & {}^i n_j & {}^i a_j & {}^i P_j \\ 0 & 0 & 0 & 1 \end{bmatrix} \quad \text{(II.5)}$$

-La matrice A représente la matrice de rotation ou d'orientation du repère R_i par rapport à R_j

- La colonne P représente la translation du repère R_i par rapport au repère R_j

Finalement, la matrice de transformation ${}^i T_j$:

- Est considéré comme la représentation du repère R_j dans le repère R_i .
- Permet le passage du repère R_i au repère R_j

Soit $\text{Tr}(a, b, c)$ la transformation d'une translation pure où a , b , et c sont les translations le long des axes x, y et z respectivement. Quand l'orientation est conservée, la matrice de transformation de cette translation a la forme suivante :

$${}^i T_j = \text{Tr}(a, b, c) = \begin{bmatrix} 1 & 0 & 0 & a \\ 0 & 1 & 0 & b \\ 0 & 0 & 1 & c \\ 0 & 0 & 0 & 1 \end{bmatrix} \quad (\text{II.6})$$

On utilise par la suite la notation $\text{Tr}(u, d)$ pour désigner une translation d'une valeur d le long de l'axe u .

Propriétés : $\text{Tr}(a, b, c) = \text{Tr}(x, a) \cdot \text{Tr}(y, b) \cdot \text{Tr}(z, c)$

L'ordre des multiplications étant quelconque [18].

II.5.1.2 Rotation et représentation de la rotation (angles euler) [19]

a. Principe

Faire trois rotations successive d'angle α, β, γ au tour des axes z, y et x du repère courant (Figure II.2).

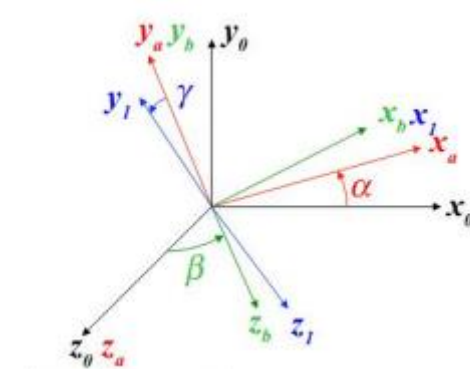


Figure II.2 trois rotations successives

b. Matrice de rotations

On définit $\text{Rot}(x, \theta)$ la transformation homogène qui s'exprime par :

$${}^i T_j = \text{Rot}(x, \theta) = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & c\theta & -s\theta & 0 \\ 0 & s\theta & c\theta & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} = \begin{bmatrix} 0 \\ \text{rot}(x, \theta) 0 \\ 0 \\ 0 \ 0 \ 0 \ 1 \end{bmatrix} \quad (\text{II.7})$$

$\text{Rot}(x, \theta)$ désigne la rotation ou l'orientation de repère R_i d'un angle θ autour de l'axe x du repère R_j .

De la même façon on définit la rotation autour de y par :

$${}^i T_j = \text{Rot}(y, \theta) = \begin{bmatrix} c\theta & 0 & -s\theta & 0 \\ 0 & 1 & 0 & 0 \\ s\theta & 0 & c\theta & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} = \begin{bmatrix} 0 \\ \text{rot}(y, \theta) 0 \\ 0 \\ 0 \ 0 \ 0 \ 1 \end{bmatrix} \quad (\text{II.8})$$

Et la rotation autour de z par :

$${}^i T_j = \text{Rot}(z, \theta) = \begin{bmatrix} c\theta & -s\theta & 0 & 0 \\ s\theta & c\theta & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} = \begin{bmatrix} 0 \\ \text{rot}(z, \theta) 0 \\ 0 \\ 0 \ 0 \ 0 \ 1 \end{bmatrix} \quad (\text{II.9})$$

b. Transformation inverse

Soit R , une matrice de rotation :

$$R_{01} = \begin{bmatrix} r_{11} & r_{12} & r_{13} \\ r_{21} & r_{22} & r_{23} \\ r_{31} & r_{32} & r_{33} \end{bmatrix} \quad (\text{II.10})$$

$$\left. \begin{aligned} \beta &= -\arcsin(-r_{31}) \\ \alpha &= \arctan 2 \left(\frac{r_{21}}{\cos \beta}; \frac{r_{11}}{\cos \beta} \right) \\ \gamma &= \arctan 2 \left(\frac{r_{32}}{\cos \beta}; \frac{r_{33}}{\cos \beta} \right) \end{aligned} \right\} \text{pour } \beta \neq \mp \frac{\pi}{2} \quad (\text{II.11})$$

$$\beta = \frac{\pi}{2}; \alpha = 0 \rightarrow \lambda = \arctan 2(r_{12}, r_{22})$$

$$\beta = -\frac{\pi}{2}; \alpha = 0 \rightarrow \lambda = -\arctan 2(r_{12}, r_{22})$$

II.5.1.3 La convention de denavit-hartenberg pour le calcul de MGD

Méthodologie à suivre pour décrire les robots à structure ouverte simples (Figure II.3). Une structure ouverte simple est composée de $n+1$ corps notés $C_0 \dots C_n$ et de n articulations.

Le corps C_0 désigne la base du robot et le corps C_n le corps qui porte l'organe terminal. L'articulation j connecte le corps C_j au corps C_{j-1} .

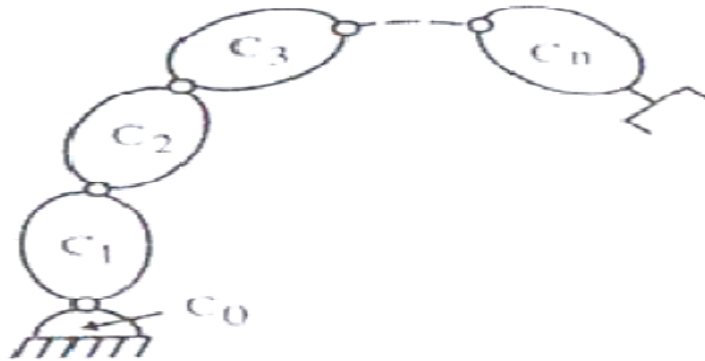


Figure II.3 : Robot a structure ouverte simple

a. Principe

- Fixer des repères à chaque corps du robot.
- Calculer les matrices homogènes entre chaque corps.
- Calculer la matrice homogène entre base et organe terminal.

b. Hypothèses

On suppose que le robot est constitué d'un chaînage de $n+1$ corps liés entre eux par n articulations rotoïdes ou prismatiques, à chaque corps on associe un repère R_i , les repères sont numérotés de 0 à n .

La i ème articulation, dont la position est notée q_i est le point qui relie les corps $i-1$ et i .

Le repère R_j fixé au corps C_j est défini de sorte que :

- L'axe z_j est porté par l'axe de l'articulation j .

- L'axe x_j est porté par la perpendiculaire commune aux axes z_j et z_{j+1} . Si les axes z_j et z_{j+1}

c. Les paramètres de Denavit-Hartenberg

Le passage du repère R_{j-1} au repère R_j s'exprime en fonction des quatre paramètres géométriques suivants : (Figure II.4)

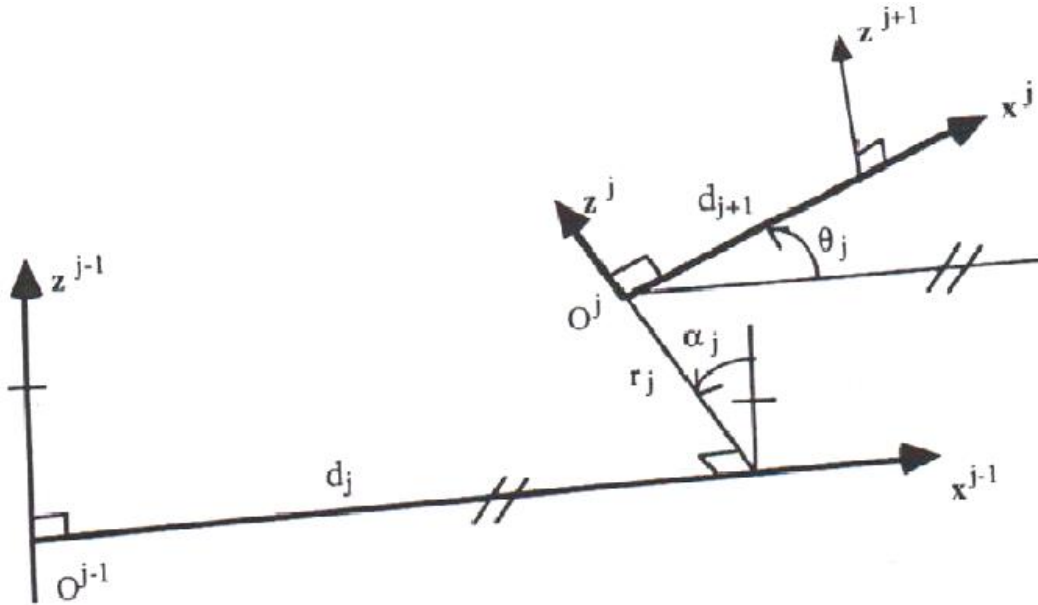


Figure II.4: exemple d'expressions des quatre paramètres de Denavit-Hartenberg

- 1- α_j : angle entre les axes z_{j-1} et z_j correspondant à une rotation autour de x_{j-1}
- 2- d_j : distance entre z_{j-1} et z_j le long de x_{j-1}
- 3- θ_j : angle entre les axes x_{j-1} et x_j correspondant à une rotation autour de z_j
- 4- r_j : distance entre x_{j-1} et x_j le long de z_j
- Si l'articulation i est de type prismatique, alors d_i est variable.
- Si l'articulation i est de type rotoïdes, alors θ_i est variable.

A partir de cette description on peut définir la matrice de transformation homogène définissant le repère R_j dans le repère R_{j-1} .

On a :

$${}^{j-1}T_j = Rot(x, \alpha_j) Trans(x, d_j) Rot(z, \theta_j) Trans(z, r_j) \tag{II.12}$$

$${}^{j-1}T_j = \begin{bmatrix} {}^{j-1}A_j & {}^{j-1}P_j \\ 0_{1,3} & 1 \end{bmatrix} = \begin{bmatrix} C\theta_j & -S\theta_j & 0 & d_j \\ C\alpha_j S\theta_j & C\alpha_j C\theta_j & -S\alpha_j & -r_j S\alpha_j \\ S\alpha_j S\theta_j & S\alpha_j C\theta_j & C\alpha_j & r_j C\alpha_j \\ 0 & 0 & 0 & 1 \end{bmatrix} \quad \text{(II.13)}$$

Et

$${}^jT_{j-1} = \begin{bmatrix} {}^{j-1}A_j^T & -{}^{j-1}A_j^T {}^{j-1}P_j \\ 0_{1,3} & 1 \end{bmatrix} = \begin{bmatrix} -d_j C\theta_j \\ {}^{j-1}A_j^T & d_j S\theta_j \\ 0_{1,3} & -r_j \\ & 1 \end{bmatrix} \quad \text{(II.14)}$$

Le modèle géométrique direct (MGD) est l'ensemble des relations qui permettent d'exprimer la situation de l'organe terminal, c.à.d. les coordonnées opérationnelles du robot, en fonction des coordonnées articulaires.

Dans le cas d'une chaîne ouverte simple, il peut être représenté par la matrice de passage 0T_n :

$${}^0T_n = {}^0T_1(q_1) {}^1T_2(q_2) \dots \dots \dots {}^{n-1}T_n(q_n) \quad \text{(II.15)}$$

Le modèle géométrique direct du robot peut être représenté par la relation :

$$X = f(q) \quad \text{(II.16)}$$

q étant le vecteur des variables articulaires tel que :

$$q = [q_1 q_2 \dots \dots q_n]^T \quad \text{(II.17)}$$

les coordonnées opérationnelles sont définies par :

$$X = [x_1 x_2 \dots \dots x_m] = [P_x P_y P_z n_x n_y n_z a_x a_y a_z]^T \quad \text{(II.18)}$$

II.6 Modèle géométrique inverse (MGI)

II.6.1 Méthodes de calcul de la MGI

Le modèle géométrique direct d'un robot permet de calculer les coordonnées opérationnelles donnant la situation de l'organe terminal en fonction des coordonnées articulaires.

Le problème inverse consiste à calculer les coordonnées articulaires correspondant à une Situation données de l'organe terminal. Lorsqu'elle existe la forme explicite qui donne toutes les solutions possibles (il y a rarement unicité de solution) constitue le modèle géométrique inverse.

Position du problème :

Soit ${}^fT_E^d$ la matrice de transformation homogène définissant la situation du robot (repère R0) dans le repère atelier. Dans le cas général, on peut exprimer ${}^fT_E^d$ sous la forme :

$${}^0T_E^d = Z^0 T_n(q) E$$

Avec :

- Z est la matrice de transformation homogène définissant la situation du repère (repère R0) dans le repère atelier.
- 0T_n est la matrice de transformation homogène du repère terminal Rn dans le repère R0, fonction du vecteur des variables articulaires q :
- E est la matrice de transformation homogène définissant le repère outil RE dans le repère Rn.

Lorsque $n \geq 6$, on peut écrire la relation suivante en regroupant dans le membre de droite tous les termes connus :

Principe :

$${}^0T_n(q) = Z^{-1} T_E^d E^{-1} \tag{II.19}$$

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

$${}^0T_n = {}^0T_1(q_1) {}^1T_2(q_2) \dots {}^{n-1}T_n(q_n) \tag{II.20}$$

Soit U_0 la situation désirée telle que :

$$U_0 = \begin{bmatrix} 1 & 0 & 0 & P_x \\ 0 & 1 & 1 & P_y \\ 0 & 0 & 0 & P_z \\ 0 & 0 & 0 & 1 \end{bmatrix} = \begin{bmatrix} s_x & n_x & a_x & 0 \\ s_y & n_y & a_y & 0 \\ s_z & n_z & a_z & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} = \begin{bmatrix} s_x & n_x & a_x & P_x \\ s_y & n_y & a_y & P_y \\ s_z & n_z & a_z & P_z \\ 0 & 0 & 0 & 1 \end{bmatrix} \tag{II.21}$$

On cherche à résoudre le système d'équation suivant :

$$U_0 = {}^0T_1(q_1) {}^1T_2(q_2) \dots {}^{n-1}T_n(q_n) \tag{II.22}$$

Méthode de solution :

Pour trouver les solutions de l'équation Paul a proposé une méthode qui consiste à pré multiplier successivement les deux membres de l'équation par la matrice ${}^jT_{j-1}$

Étape 1

$$X = C T_n G_k \quad (\text{II.23})$$

En développant :

$$C^{-1}X = C^{-1}C T_n G_k \quad (\text{II.24})$$

$$C^{-1}X = T_n G_k \quad (\text{II.25})$$

$$C^{-1}X G_{k-1} = T_n G_k G_k \quad (\text{II.26})$$

$$C^{-1}X G_{k-1} = T_n \quad (\text{II.27})$$

tel que $C^{-1}X G_{k-1}$: connu et T_n : inconnu

Où, $T_n = [A_1 A_2 \dots A_n] \Rightarrow$ 12 relations avec 'n' inconnus.

Étape 2

Simplifier les équations séparer certains d.d.l. (si possible)

Étape 3

résoudre les équations simplifiées.

Généralement \Rightarrow relations trigonométriques ;

\Rightarrow relations non-linéaires ;

\Rightarrow pas toujours facile à résoudre.

4 types d'équation souvent rencontrés :

L'inconnu de ces types d'équation est θ et il est calculé avec $\text{Atan2}(y,x)$ qui donne l'angle dans son cadran .

II.6.2 Les problèmes liés au calcul du modèle géométrique inverse

Le calcul du modèle géométrique inverse (MGI) consiste à déterminer quelles sont les valeurs des variables articulaires lorsque sont données la position et l'orientation d'un

repère lié à l'organe terminal. Dans le cas d'un robot parallèle à six degrés de liberté cette résolution est analogue, pour chaque chaîne, à celle du modèle géométrique inverse d'un robot sériel. Si, d'une manière générale, pour un robot parallèle ce modèle est beaucoup plus simple à résoudre que sa réciproque, il convient de noter qu'il peut être malgré tout difficile. A l'origine de ce phénomène est la structure de la chaîne élémentaire reliant l'effecteur à la base. Si cette chaîne est composée de plusieurs branches, il peut y avoir plusieurs millions de solutions.

Le problème de la modèle géométrique inverse s'agit de résoudre un système de m équations à n inconnues, ce système étant non linéaire.

Le nombre de solutions dépend de l'architecture du robot manipulateur et de l'amplitude des articulations.

Trois cas se présentent pour calculer le MGI :

1. Solutions en nombre fini.
2. Absence de solution, lorsque la position de l'organe terminal désirée est en dehors de la zone accessible du robot.
3. Infinité de solutions lorsque :
 - le robot est redondant vis-à-vis la tâche.
 - le robot se trouve dans certaines configurations singulières.

Lorsqu'il est possible de calculer toutes les configurations permettant d'atteindre une situation donnée d'un robot manipulateur ce dernier est dit résoluble.

II.7 Les modèles géométriques direct et inverse dans la boucle de commande d'un robot

L'étude du modèle géométrique permet d'établir le lien entre les coordonnées articulaires et les coordonnées cartésiennes et angulaires de l'extrémité de l'effecteur. Les coordonnées articulaires sont les valeurs des variables de commande du robot sur lesquelles on peut influencer par l'intermédiaire des actionneurs (moteurs) a fin de déplacer l'organe terminal.

Ces modèles sont utilisés pour l'analyse de l'espace de travail, la simulation, la commande, la calibration, la planification de trajectoire etc..[20]

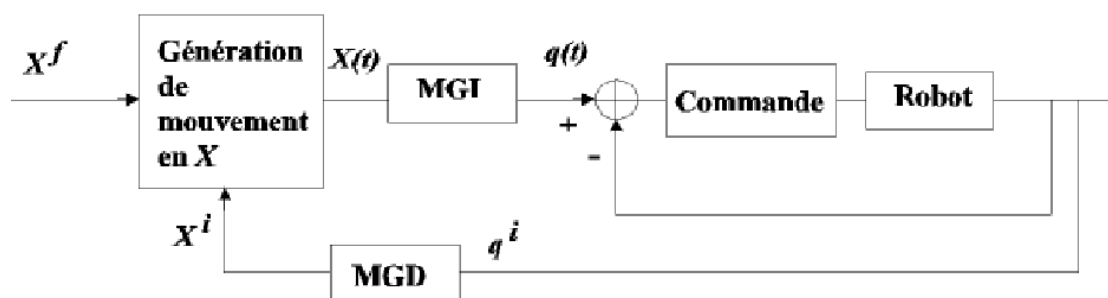


Figure II.5 MGD et MGI dans une boucle de commande d'un robot

II.8 Conclusion

Dans ce chapitre on a vu les méthodes importantes pour la modélisation d'un robot manipulateur à n articulations.

D'une manière générale, on recherche toujours le modèle le plus simple qui permet d'expliquer de manière satisfaisante le comportement du processus dans son domaine d'application.

On distingue :

- Les modèles géométriques qui expriment la situation de l'organe terminal en fonction de la configuration du mécanisme
- Les modèles cinématiques permettent de contrôler la vitesse de déplacement du robot afin de connaître la durée d'exécution d'une tâche.
- Les modèles dynamiques qui permet de donner la relation entre les couples (et/ou forces) appliqués aux actionneurs et les positions, vitesses et accélérations articulaires

CHAPITRE III
**Approximation du modèle
géométrique inverse par les
réseaux de neurones artificiels**

CHAPITRE III : Approximation du modèle géométrique inverse par les réseaux de neurones artificiels

III.1 Introduction :

Les systèmes réels sont difficiles à étudié, donc on est amené à les représenter mathématiquement pour pouvoir les commander.

Il y a un intérêt considérable pour ces dernières années en explorant l'application des réseaux de neurones artificiels pour l'identification. L'identification des systèmes exige la sélection d'une classe de la fonction (ou modèle) pour approximer le comportement entrée-sortie des systèmes de la meilleure façon possible. Dans beaucoup de situations, telles que l'identification des systèmes dynamiques, le rendement du système physique à modeler est une fonction des entrées et des sorties antérieures.

III.2 Identification des processus par réseaux de neurones

Par leur capacité d'approximation universelle, les réseaux de neurones sont bien adaptés pour l'identification des systèmes non linéaires. En effet dans le cas ou le model de système est une fonction non linéaire, l'objectif est alors d'approximer cette fonction par les RNA (réseaux de neurones artificiels). Cette approximation est réalisée par apprentissage des poids du réseau, l'apprentissage peut se faire hors ligne ou en ligne :

- Dans le cas de hors ligne, l'apprentissage est basé sur un ensemble de données définissant la fonction commande,
- Dans le cas d'en ligne, la mise à jour des poids est essentiellement adaptative.

Le principe de l'identification par réseau neuronaux consiste à substituer aux modèles paramétriques classiques des modèles neuronaux, c'est-à-dire proposer un modèle établissant une relation entre son entrée et sa sortie et à déterminer, à partir du couple

des signaux d'entrée-sortie, le comportement du modèle. Deux raisons importantes nous motivent :

- Prédire le comportement d'un système pour différentes conditions de fonctionnement ;
- Elaborer une loi de commande à appliquer au processus, pour qu'il réalise l'objectif assigné.

Nous citerons deux techniques d'identification à base de réseaux de neurones multicouches : la méthode d'identification directe et la méthode d'identification inverse.

III.1.1 Identification directe

La figure III.1 montre le schéma général d'identification directe d'un processus. Sur cette figure, le réseau de neurones identificateur RNI est utilisé en parallèle avec un processus de type boîte noire. La sortie du processus, y , est comparée avec la sortie du réseau de neurones, \hat{y} , puis l'erreur $y - \hat{y}$ est utilisée afin d'affiner les paramètres du système neuronal.

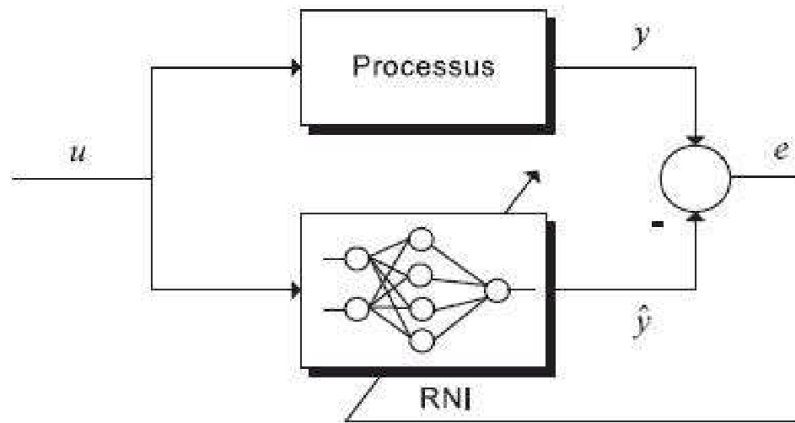


Figure III.1 Schéma d'identification directe d'un processus par réseau de neurones.

Considérons un système non linéaire avec une entrée $u(k)$ et une sortie $y(k)$

$y(k)$ peut dépendre de $u(k)$ seulement, ou de $u(k)$ et les états précédents de y et/ou u c'est à dire :

$$y(k)=F(u(k))$$

Ou

$$y(k)=F(u(k-1), \dots, u(k-m), y(k-1), y(k-2), \dots, y(k-n))$$

Le principe de l'identification directe par réseau de neurones consiste à construire le réseau (en générale MLP) qui approxime la fonction F, cette approximation se fait par apprentissage hors ligne.

Etant donné la fonction F, il faut construire un RNA pour réaliser l'approximation, on génère une base de données de N échantillon $y(k-1), y(k-2), \dots, y(k-n), u(k-1), \dots, u(k-m)$ (les entrés) et on obtient $y(k)$ (la sortie) à partir de F. L'objectif de l'apprentissage est de déterminer la fonction F à partir de ces données. A l'instant t , on donne au RNA :

$y(k-1), y(k-2), \dots, y(k-n), u(k-1), \dots, u(k-m)$ et on obtient à la sortie $\hat{y}(k)$. La mise à jour des poids du réseau est basée sur la minimisation de l'erreur entre $y(k)$ et $\hat{y}(k)$.

$$J = \frac{1}{2} \sum_{i=1}^m y_i - \hat{y}_i$$

III.1.2 Identification inverse

Dans cette méthode, l'entrée du processus est comparée avec la sortie de l'identificateur neuronal RNI est la sortie du processus est injectée comme entrée du réseau de neurones (figure III.2). Après un apprentissage hors-ligne du modèle inverse, le RNI peut être configuré afin d'assurer un contrôle directe du processus.

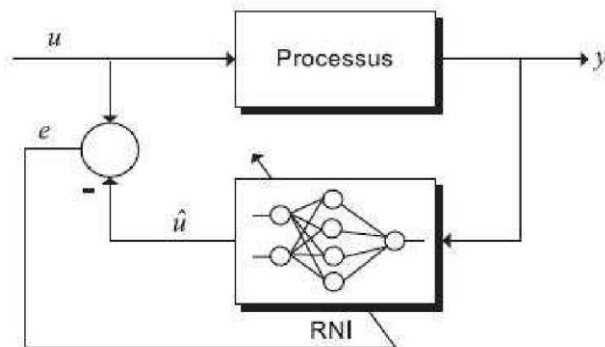


Figure III.2 Schéma d'identification inverse d'un processus avec un réseau de neurones[15]

III.2 Approximation du MGI du robot Mentor par Identification inverse

Le problème inverse consiste à calculer les coordonnées articulaires correspondant à une situation donnée de l'organe terminal. Lorsqu'elle existe, la forme explicite qui donne toutes les solutions possibles (il y a rarement unicité de solution) constitue ce que l'on appelle le modèle géométrique inverse.

Le MGI ce n'est pas toujours facile à calculer dans ce fait on va proposer dans la suite du chapitre une méthode pour l'approximation de ce modèle.

III.2.1 la méthode proposée pour l'approximation de MGI du robot Mentor

Pour l'approximation du MGI on va utiliser les réseaux de neurones (fig III.3) qui seront notre solution du fait qu'ils peuvent modéliser n'importe quelle fonction non linéaire, et possèdent aussi la capacité d'apprentissage à partir d'exemples (entrées /sorties) d'un système, et elle possède les caractéristiques de la mémorisation par la généralisation.

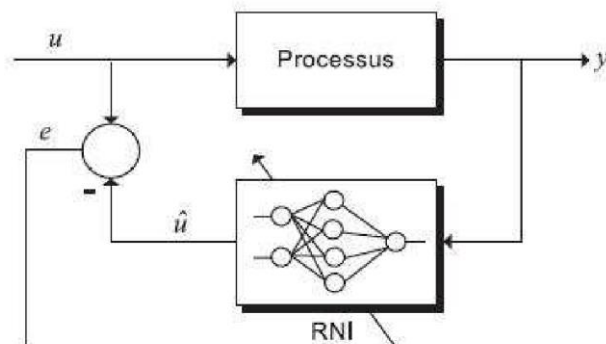


Figure III.3 Identification inverse pour l'approximation du MGI du robot Mentor [7]

III.2.2 Calcul du MGD du robot Mentor [8]

a. le robot Mentor :

Le robot manipulateur Mentor est un robot de 5 degrés de liberté de type rotoïde (5R), 3 degrés de liberté (3ddl) pour le porteur et 2ddl pour l'organe terminal plus une pince, au bout de ce dernier.



Figure III.4 Image du robot Mentor.

b. Fixage des repères DH (de Denavit-Hartenberg)

La modélisation géométrique directe du robot **Mentor** est réalisée en utilisant les translations homogènes, selon la méthode de Denavit-Hartenberg[2]

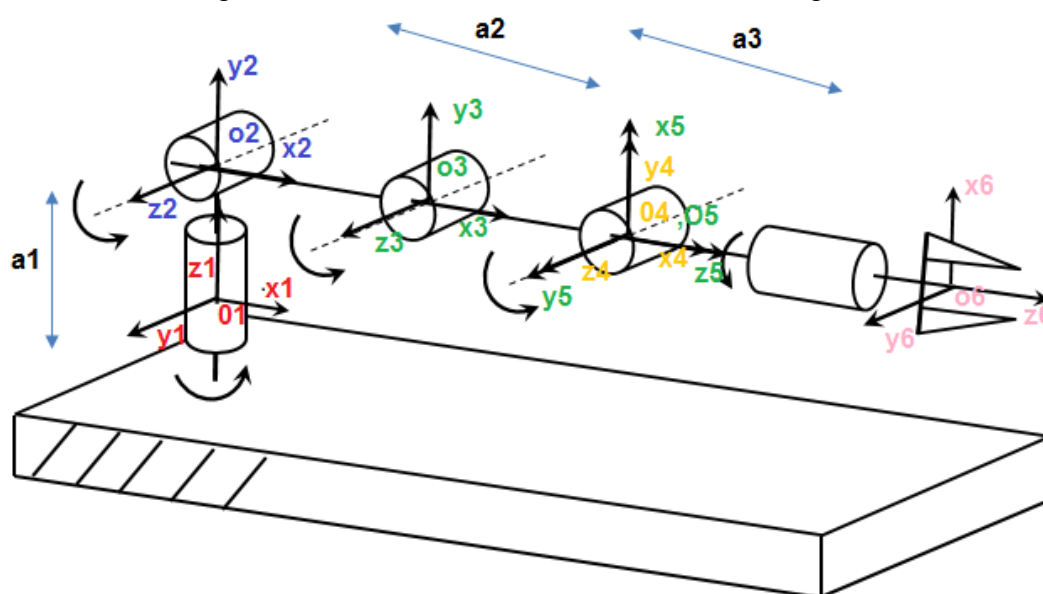


Fig III.5 Les translations homogènes

c. Tableau de Denavit-Hartenberg :

	1	2	3	4	5
α	90^0	0	0	90^0	0
ai	0	a2	a3	0	0
θ	$\theta 1$	$\theta 2$	$\theta 3+90^0$	$\theta +90^0$	$\theta 5$
di	a1	0	0	0	0

Tableau III.1 paramètre de Denavit-Hartenberg

α_i : angle entre les axes z_{j-1} et z_j correspondant à une rotation autour de x_{j-1}

d_i : distance entre z_{j-1} et z_j le long de x_{j-1}

θ_i : angle entre les axes x_{j-1} et x_j correspondant à une rotation autour de z_j

ai : distance entre x_{j-1} et x_j le long de z_j

d. Matrices de transformations homogènes

A partir du tableau et compte tenu de la matrice de passage élémentaire de Denavit-Hartenberg, on écrit les matrices des transformations homogènes :

-La matrice de transformation T_1^0 (exprime le repère **R1** dans le repère **R0**) :

$$T_1^0 = \begin{bmatrix} c_1 & 0 & 0 & 0 \\ s_1 & 0 & -c_1 & 0 \\ 0 & 1 & 0 & r_1 \\ 0 & 0 & 0 & 1 \end{bmatrix} \quad \text{(III.1)}$$

-La matrice de transformation T_2^1 (exprime le repère **R2** dans le repère **R1**) :

$$T_2^1 = \begin{bmatrix} c_2 & s_2 & s_2 & r_2 c_2 \\ s_2 & c_2 & 0 & r_2 s_2 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \quad \text{(III.2)}$$

-La matrice de transformation T_3^2 (exprime le repère **R3** dans le repère **R2**) :

$$T_3^2 = \begin{bmatrix} c_3 & -s_3 & s_3 & r_3 c_3 \\ s_3 & c_3 & 0 & r_3 s_3 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \quad \text{(III.3)}$$

-La matrice de transformation T_4^3 (exprime les repères \mathbf{R}_4 dans le repère \mathbf{R}_3) :

$$T_4^3 = \begin{bmatrix} -s_4 & 0 & 0 & 0 \\ c_4 & 0 & s_4 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \quad (\text{III.4})$$

-La matrice de transformation T_5^4 (exprime les repères \mathbf{R}_5 dans le repère \mathbf{R}_4) :

$$T_5^4 = \begin{bmatrix} c_5 & -s_5 & s_5 & 0 \\ s_5 & c_5 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \quad (\text{III.5})$$

Avec : c_i : Cosinus de θ_i et s_i : Sinus de θ_i

La matrice homogène total T_5^0 correspond au passage du repère \mathbf{R}_0 au repère \mathbf{R}_5 est calculer par :

$$T_5^0 = T_1^0 T_2^1 T_3^2 T_4^3 T_5^4 \quad (\text{III.6})$$

Elle est représentée sous la forme suivante :

$$T_5^0 = \begin{bmatrix} s_x & n_x & a_x & P_x \\ s_y & n_y & a_y & P_y \\ s_z & n_z & a_z & P_z \\ 0 & 0 & 0 & 1 \end{bmatrix} \quad (\text{III.7})$$

Avec :

$$\begin{cases} s_x = c_1 s_5 (s_2 + c_2 s_3) - s_{234} c_1 c_5 \\ s_y = s_5 (s_1 s_2 - c_1 + c_2 s_1 s_3) s_{234} c_5 s_1 \\ s_z = c_{234} c_5 + s_2 s_3 s_5 \end{cases} \quad (\text{III.8})$$

$$\begin{cases} n_x = s_{234} c_1 s_5 + c_1 c_5 (s_2 + c_2 s_3) \\ n_y = c_5 (s_1 s_2 - c_1 + c_2 s_1 s_3) + s_{234} s_1 s_5 \\ n_z = c_5 s_2 s_3 - c_{234} s_5 \end{cases} \quad (\text{III.9})$$

$$\begin{cases} a_x = -c_1 (s_{23} s_4 + s_{234} s_5) \\ a_y = -s_1 (s_{23} s_4 + s_{234} s_5) \\ a_z = c_{23} s_4 - s_5 (s_{23} s_4 - c_{23} c_4) \end{cases} \quad (\text{III.10})$$

$$\begin{cases} P_x = c_1 (r_3 c_{23} + r_2 c_2) \\ P_y = s_1 (r_3 c_{23} + r_2 c_2) \\ P_z = r_1 + r_3 s_{23} + r_2 s_2 \end{cases} \quad (\text{III.11})$$

Avec :

$c_i : \cos(\theta_i) ;$

$c_{ijk} : \cos(\theta_i + \theta_j + \theta_k) ;$

$s_i : \sin(\theta_i) ;$

$s_{ijk} : \sin(\theta_i + \theta_j + \theta_k) ;$

III.2.3 Choix de la topologie et de l'architecture du réseau :

A priori, la relation entre les entrées et les sorties est complexe (en particulier non linéaire) ce qui implique l'utilisation d'un réseau multicouche (ou de plusieurs réseaux multicouches).

Il n'existe pas de méthodes générales pour le choix du nombre de neurones sur chaque couche cachée ainsi que le nombre de ces dernières, donc pour le choix on a comparé trois architectures des réseaux de neurones pour le but de choisir la plus efficace.

On a examiné les architectures par leurs courbes de performances, de régression, du teste de l'approximation du réseau de neurones, et le temps d'apprentissage.

-Architecture a :

Dans cette architecture on a utilisé un réseau qui se compose d'une seule couche cachée de 15 neurones, de 6 entrées et 5 cellules de sortie.

-Architecture b :

Dans cette architecture on a utilisé un réseau qui se compose de 2 couches cachées de 25 neurones, de 6 entrées et 5 cellules de sortie.

-Architecture c :

Dans cette architecture on a utilisé un réseau qui se compose de 3 couches cachées de 90 neurones, de 6 entrées et 5 cellules de sortie.

III.2.4 Remplissage de la base des données pour l'apprentissage :

Pour le remplissage de notre base des données on va utiliser le model géomatique direct de notre robot mentor ; on va varier à chaque fois les entrées qui sont dans une boucle for de 0 à $2*\pi$ en fin de trouver des exemplaire des sorties (x ,y,z,tx,ty,tz) et des entrées $(\vartheta_1, \vartheta_2, \vartheta_3, \vartheta_4, \vartheta_5)$, à l'aide de ces exemplaire le base de donnée est remplis ,

Algorithme :

- Définir les longueurs des bras du robot mentor
- Initialiser les entrées du model géométrique direct du robot mentor à zéro

-faire varier chaque fois les entrées de $0 : \pi/12 : \pi/2$ et les mettre dans une boucle for

-trouver les sorties (x y z tx ty tz) à chaque variation des entrées ($\theta_1 \theta_2 \theta_3 \theta_4 \theta_5$)

-Enregistrer les entrées et les sorties trouvées

3.5 Apprentissage du réseau de neurones :

L'apprentissage est de type supervisé. La base d'apprentissage est composée de couple de vecteurs d'entrée et des vecteurs de sortie associés. L'algorithme d'apprentissage initial fait appel à la règle de Hebb. Une entrée x est appliquée sur les neurones d'entrée du réseau et l'on force dans le même temps les valeurs des neurones de sortie à y. Les poids de chaque connexion sont alors modifiés selon la coactivité du neurone afférent (entrée) et du neurone efférent (sortie). Cet algorithme est itéré sur tous les exemples de la base d'apprentissage.

III.3 Résultats d'approximation du model géométrique inverse par les réseaux de neurones artificiels

La courbe de performance représente la valeur quadratique après la fin de l'apprentissage, plus que l'erreur est petit le réseau est plus généralisable.

La courbe de régression représente le pourcentage des données valider, testées et apprenties.

Les courbes des orientations réelles et approximées montrent la capacité du réseau de neurones d'approximer le model géométrique inverse ; dans ses courbes on a pris des valeurs réel (les entrées du model géométrique direct qui sont les sorties du réseau de neurones) et des valeurs approximées par le réseau de neurones et on les représentés sur le même graphe.

III.3.1 Résultats de l'architecture a :

Dans cette architecture on a utilisé un réseau de neurone qui se compose d'une seule couche cachée de 15 neurones, de 6 entrées et 5 cellules de sortie.

Dans la figure III.6 on va voir la performance de cette architecture, dans la figure III.7 la régression générale, dans les figures III.8, III.9, III.10 les orientations réelles et approximées

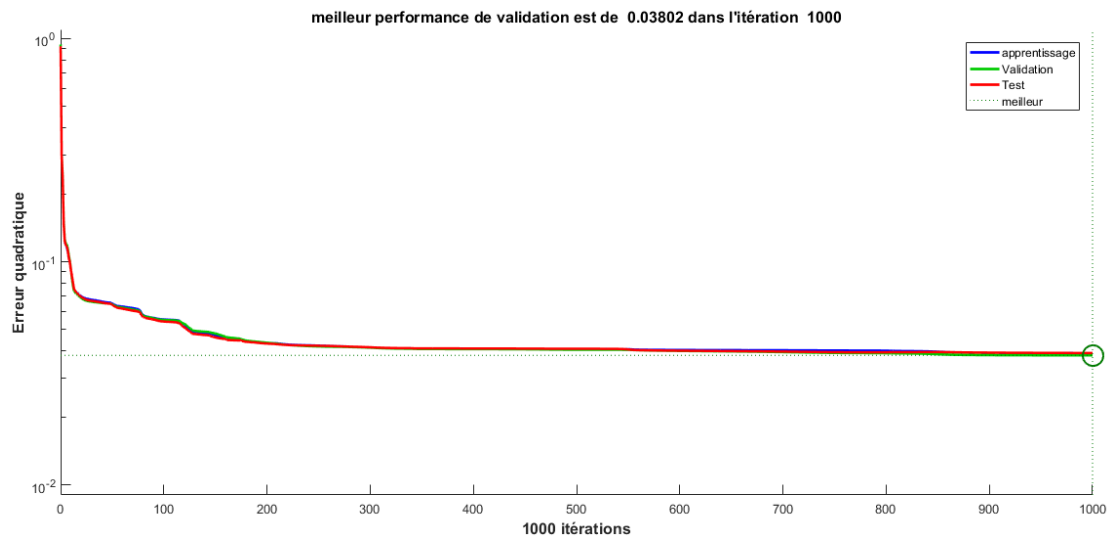


Figure III.6 performance du réseau de neurones de l'architecture a

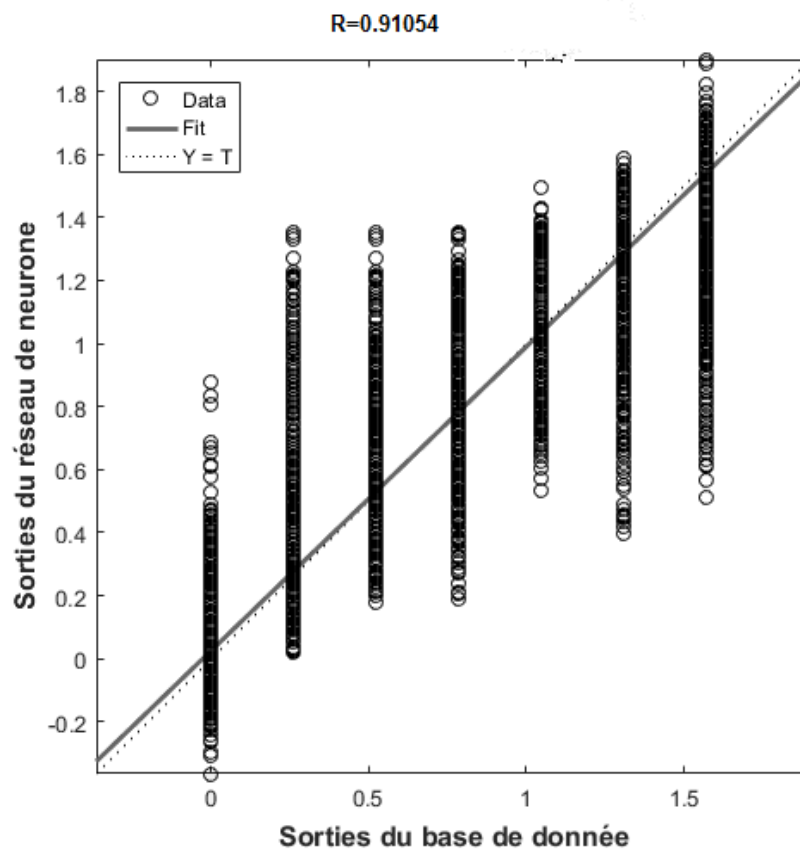


Figure III.7 Courbe de régression général du réseau de neurone pour l'architecture a

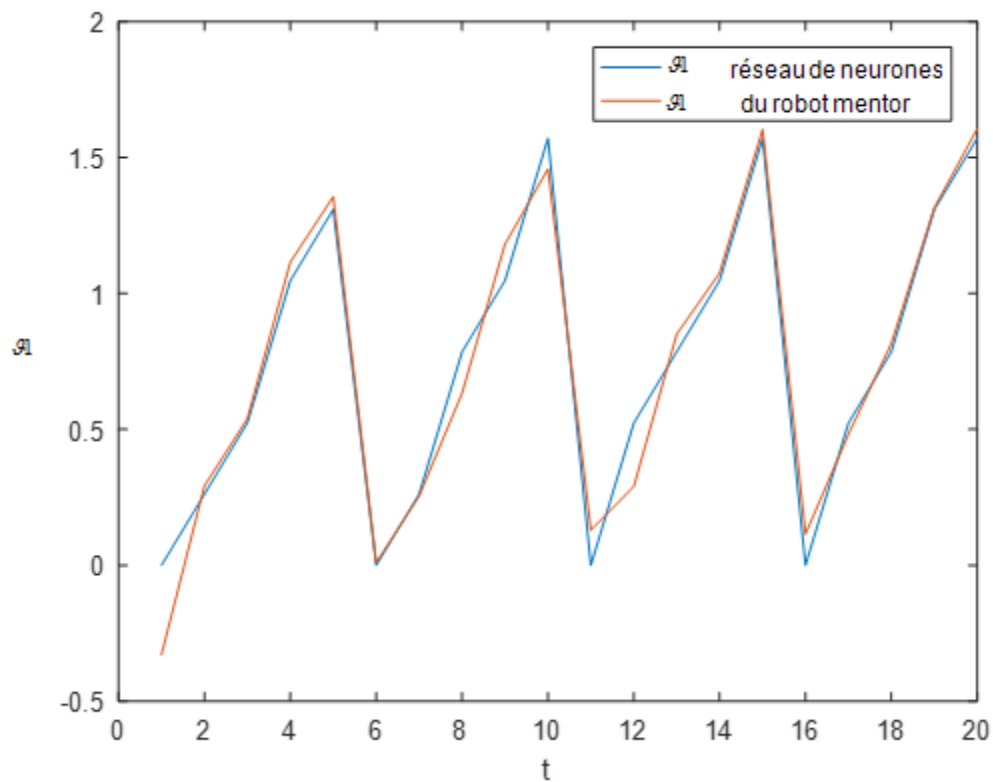


Figure III.8 les orientations réelles et approximées(θ_1)

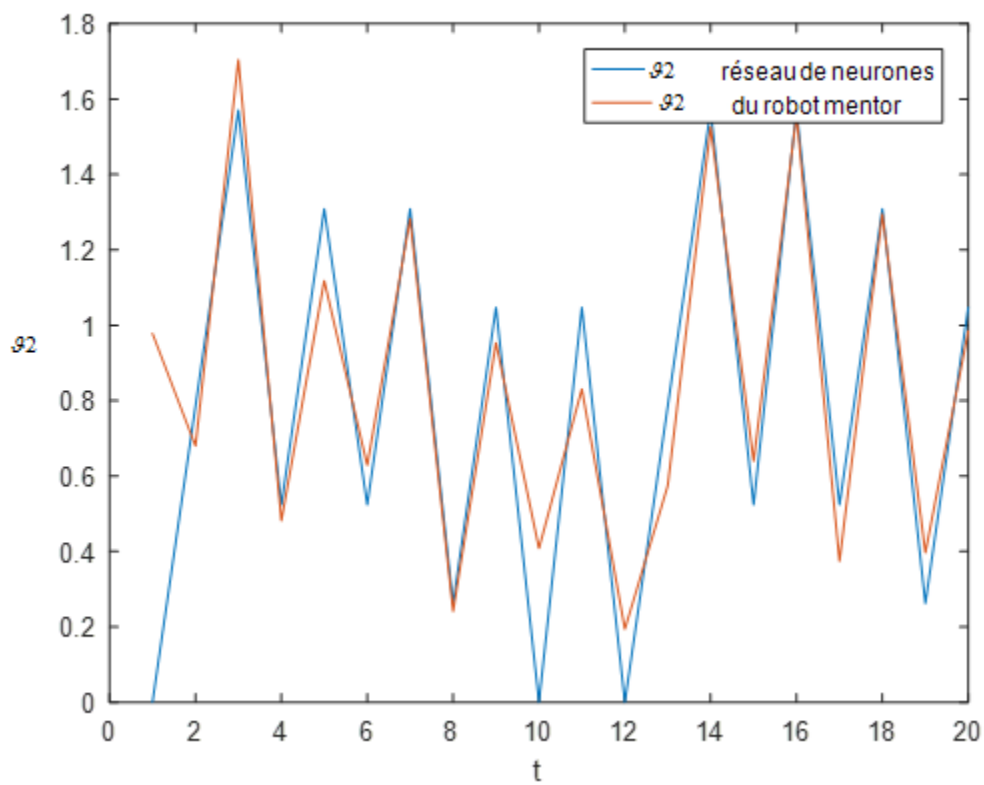


Figure III.9 les orientations réelles et approximées(θ_2)

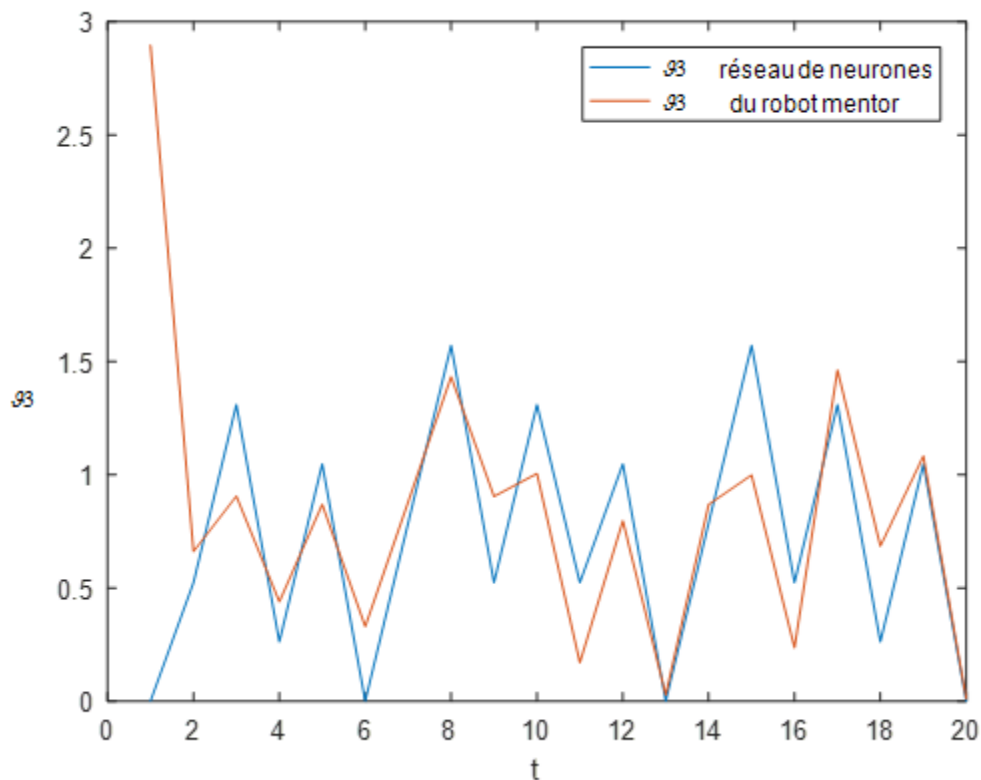


Figure III.10 les orientations réelles et approximées(θ)

Interprétations des courbes de l'architecture a :

Dans cette architecture on a une erreur quadratique de 0.03, 92% des données sont validés, 92% sont testés, l'apprentissage est fait avec 92% des données, la régression général est de 92%, le temps d'apprentissage été de 15 minutes, et les courbes des orientations réelles et approximées ne sont pas compatibles, donc **l'architecture a** sera un mauvais choix.

III.3.2 Résultats de l'architecture B :

Dans cette architecture on a utilisé un réseau de neurone qui se compose de 2 couches cachées de 30 neurones, de 6 entrées et 5 cellules de sortie.

Dans la figure III.11 on va voir la performance de cette architecture, dans la figure III.12 la régression générale, dans les figures III.13, III.14, III.15 les orientations réelles et approximées

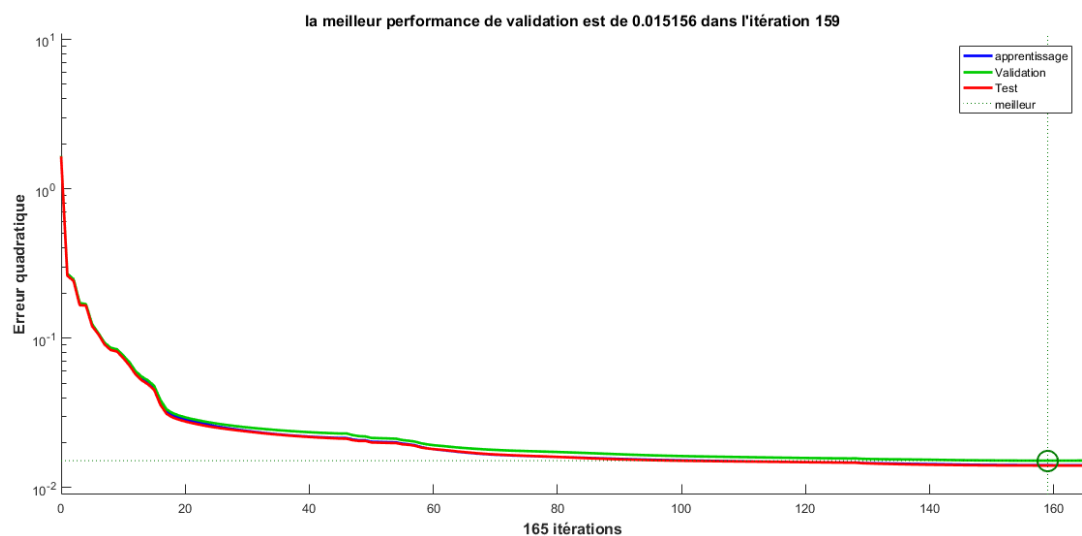


Figure III.11 performance du réseau de neurones de l'architecture b

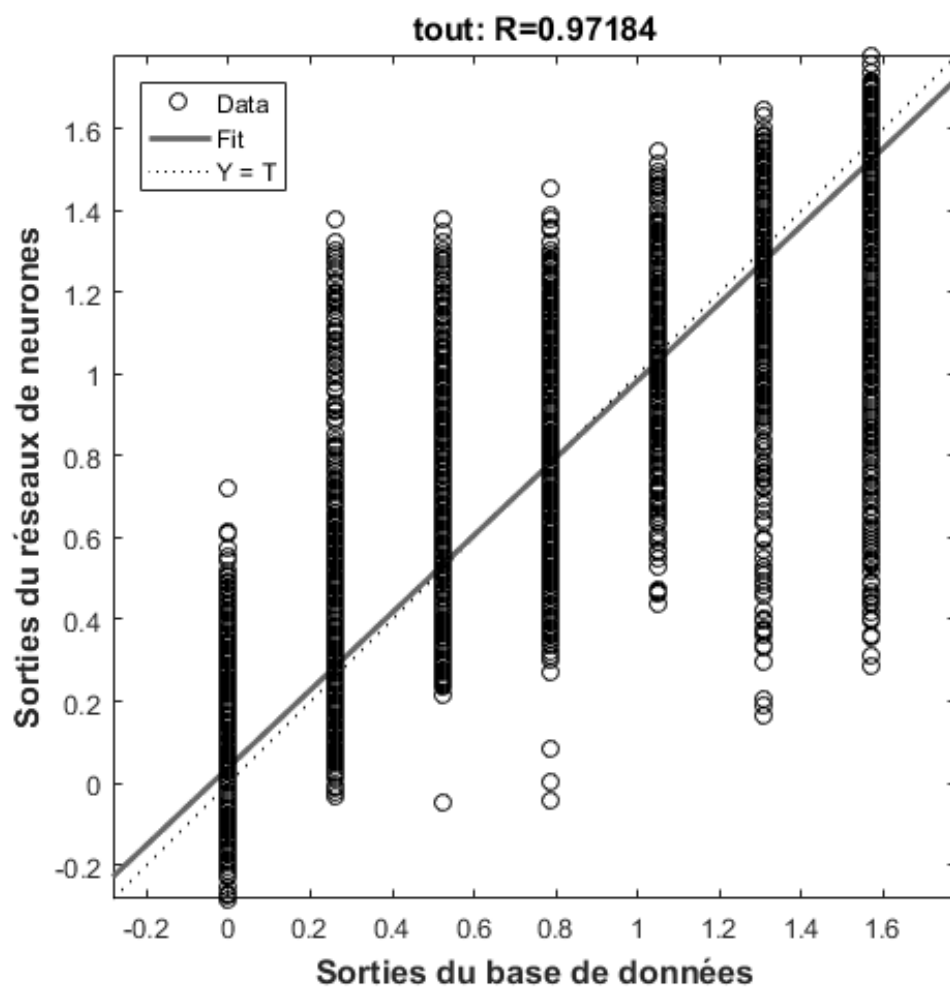


Figure III.12 Courbe de régression général du réseau de neurone pour l'architecture b

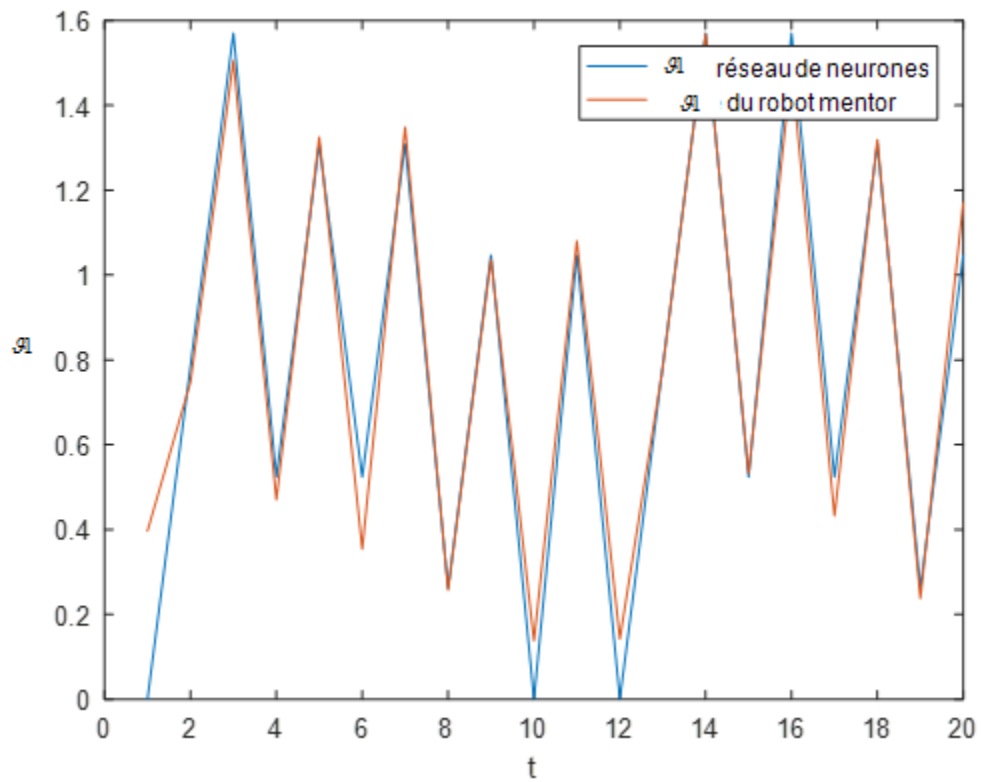


Figure III.13 les orientations réelles et approximées(θ_1)

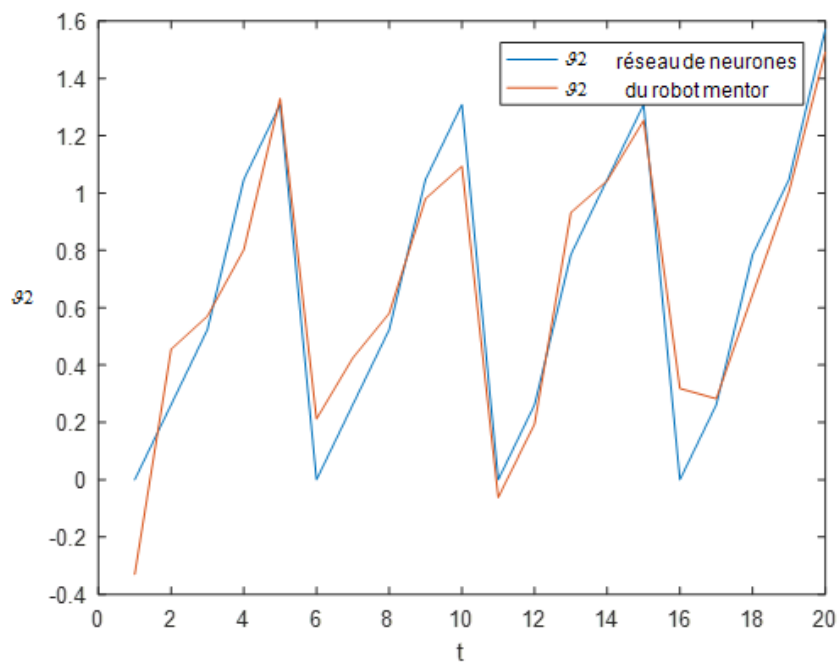


Figure III.14 les orientations réelles et approximées(θ_2)

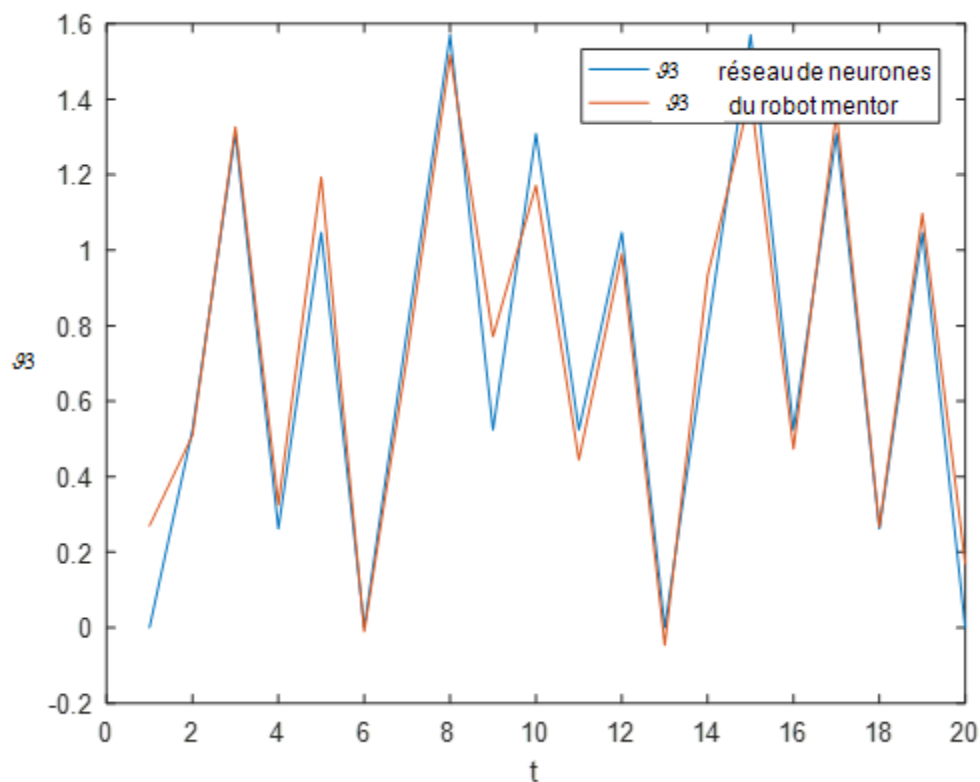


Figure III.15 les orientations réelles et approximées(θ)

Interprétations des courbes de l'architecture b :

Dans cette architecture on a une erreur quadratique de 0.01, 97% des données sont validés, 97% sont testés, l'apprentissage est fait avec 97% des données, la régression générale est de 97%, le temps d'apprentissage été de 40 minutes, et les courbes des orientations réelles et approximées ne sont pas compatibles, donc **l'architecture b** sera un mauvais choix.

III.3.3 Architecture c :

Dans cette architecture on a utilisé un réseau de neurone qui se compose de 3 couches cachée de 90 neurones, de 6 entrées et 5 cellules de sortie.

Dans la figure III.16 on va voir la performance de cette architecture, dans la figure III.17 la régression générale, dans les figures III.18, III.19, III.20 les orientations réelles et approximées

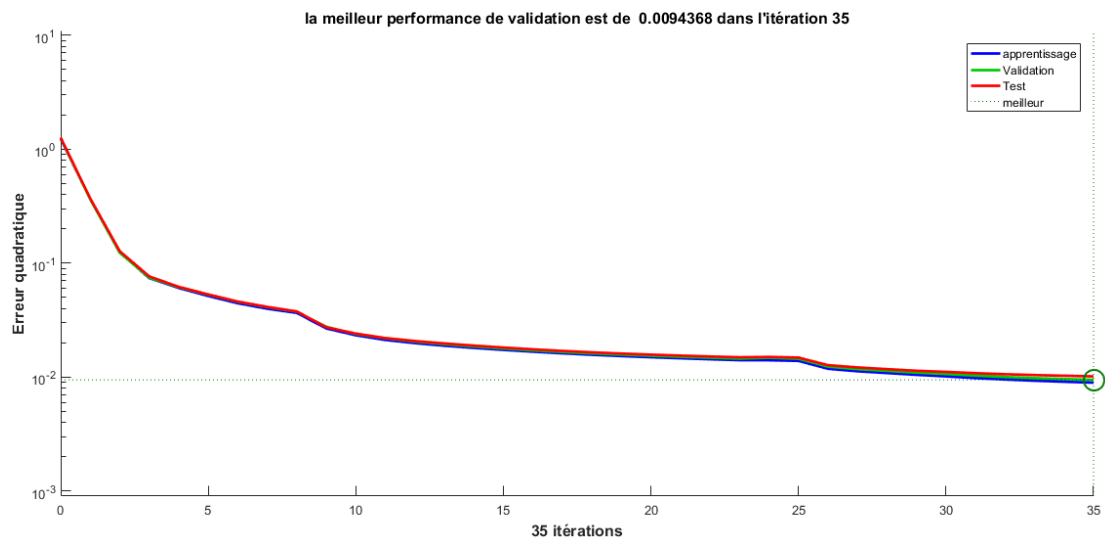


Figure III.16 performance du réseau de neurones de l'architecture 3

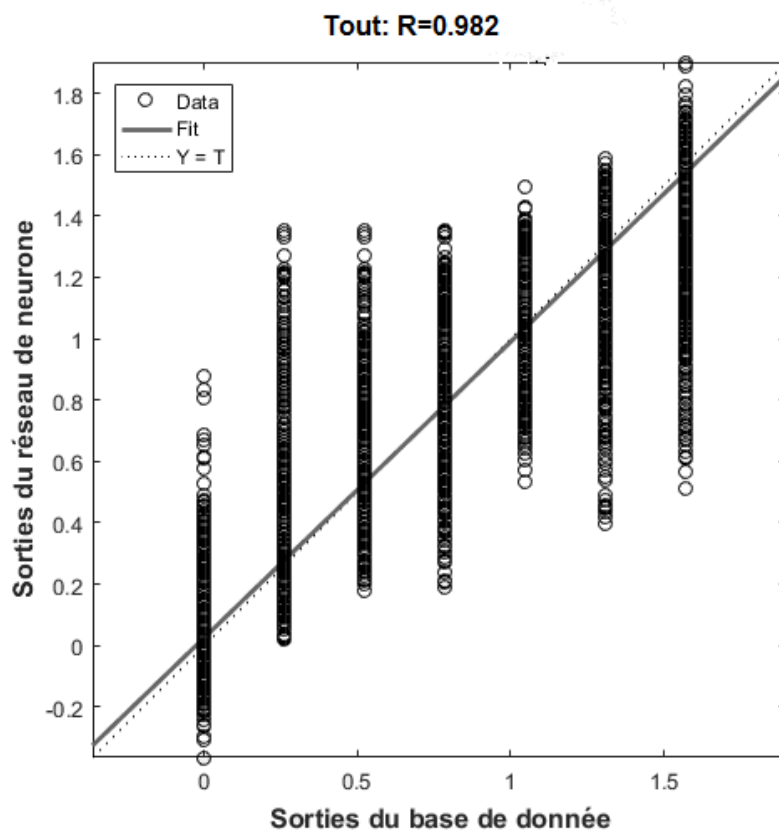


Figure III.17 Courbe de régression général du réseau de neurone pour l'architecture c

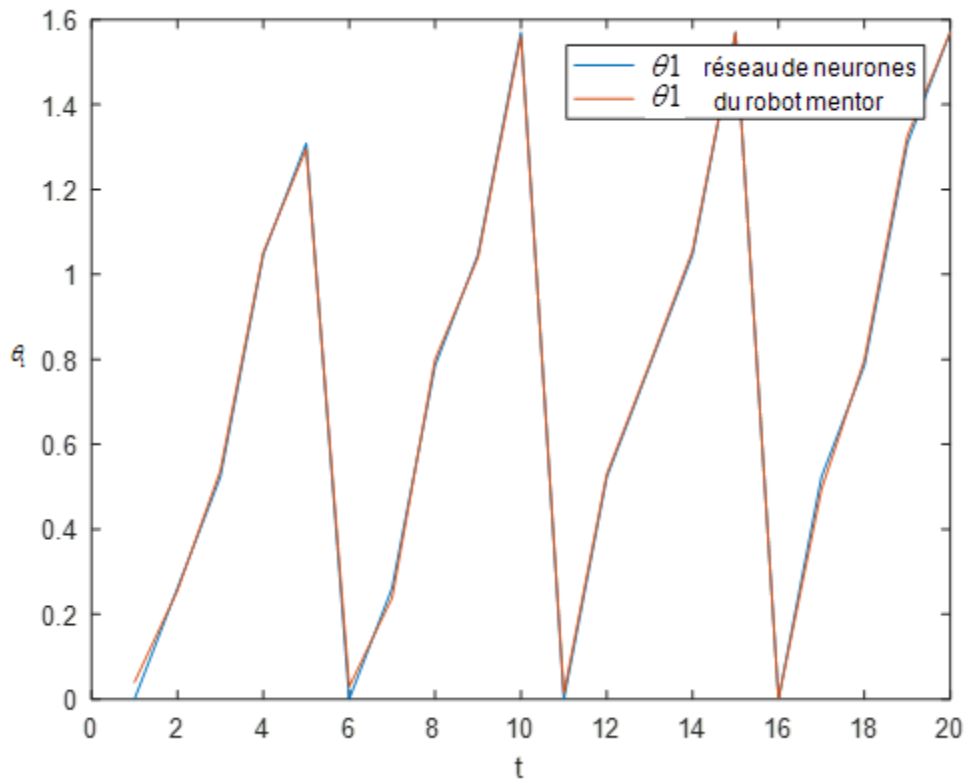


Figure III.18 les orientations réelles et approximées(θ_1)

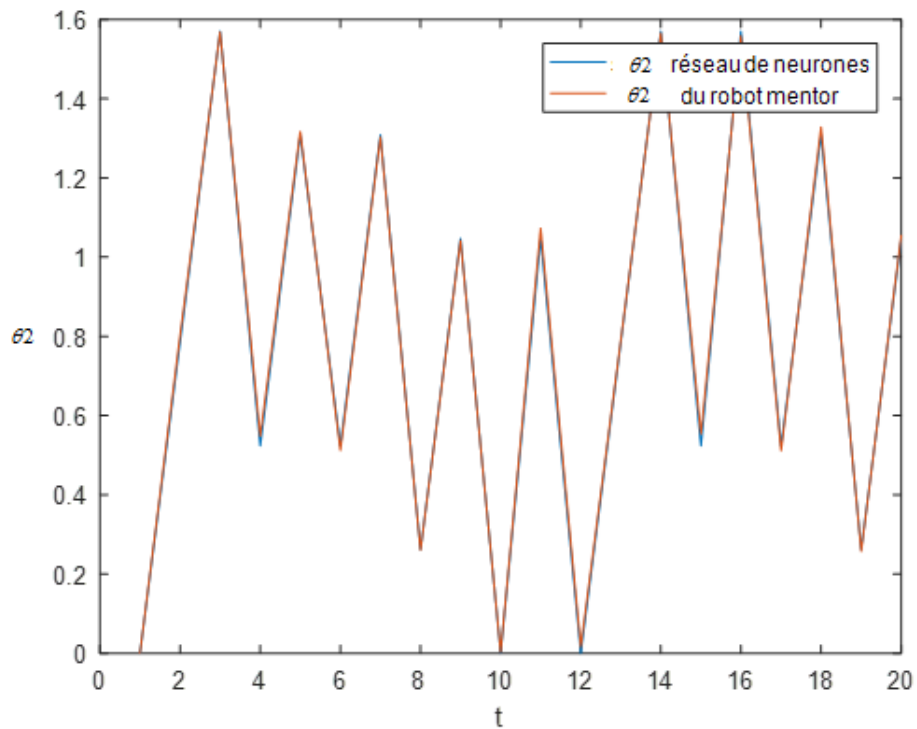


Figure III.19 les orientations réelles et approximées(θ_2)

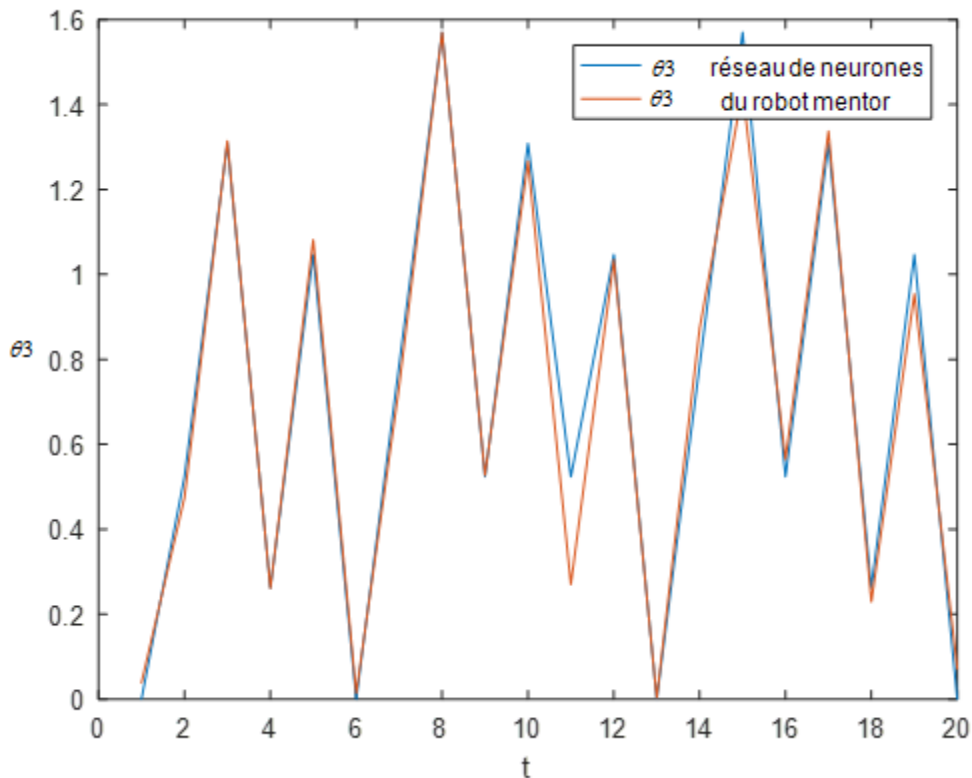


Figure III.20 les orientations réelles et approximées(θ_3)

Interprétations des courbes de l'architecture 3 :

Dans cette architecture on a une erreur quadratique de 0.009, 98% des données sont validés, 98% sont testés, l'apprentissage est fait avec 98% des données et la régression général est de 98% et le temps d'apprentissage été de 1h25minutes

Dans cette architecture on a une erreur quadratique de 0.009, 98% des données sont validés, 98% sont testés, l'apprentissage est fait avec 98% des données, la régression général est de 98%, le temps d'apprentissage été de 1h25 minutes, et les courbes des orientations réelles et approximées sont compatibles, donc **l'architecture c** sera un meilleur choix.

En résumé **l'architecture c** est la meilleure architecture car il a une erreur quadratique plus petit, un pourcentage de régression total plus grand que les autres architectures ce qui signifie la généralisation de son réseau de neurones, un temps d'apprentissage très importants, les courbes de comparaison entre les orientations

réelles et les orientations approximées sont compatibles ce qui signifie l'efficacité de l'approximation de son réseau de neurones.

On constate que plus le nombre de neurones est grand plus que le réseau de neurone est bon, la même chose pour le nombre de couche cachée et en fin le temps d'apprentissage plus qu'il est long plus que le réseau de neurone est bon.

III.4 Conclusion

Dans ce chapitre on a vu la modélisation d'un robot manipulateur réel qui est le robot mentor, on a calculé son modèle géométrique direct afin de calculer une base de données pour l'apprentissage du réseau de neurone et pour faire l'identification inverse.

On a vu 3 tests sur des architectures de réseau de neurones pour le choix d'une meilleure, afin de faire une bonne approximation du modèle géométrique inverse.

En fin on a constaté que plus le nombre de neurones est grand, plus qu'on augmente le nombre de couches cachées, plus que le temps d'apprentissage plus long l'approximation du réseau de neurones est très bonne.

CONCLUSION GENERAL

Conclusion générale

Dans cette mémoire, on a montré la pertinence d'utiliser les réseaux de neurones artificiels pour l'approximation du modèle géométrique inverse d'un robot manipulateur série. En effet, on a calculé le modèle géométrique direct du robot pour construire la base des données d'apprentissage par la méthode de Denavit-Hartenberg, ensuite on a appris, validé et testé des architectures neuronales pour une identification inverse par un réseau de neurones. Une étude comparative entre les différentes architectures proposées a été effectuée pour montrer l'effet de nombre des couches cachées et le nombre de neurones dans chaque couche sur la précision du modèle calculé.

BIBLIOGRAPHIE

BIBLIOGRAPHIE

- [1] Jean-François Jodouin. Les réseaux neuromimétiques, modèles et applications. Hermès, Paris, 1994.
- [2] Antonelo. E, Schrauwen. B, Dutoit. X, Stroobandt. D, Nuttin. M: Event detection and localization in mobile robot navigation using reservoir computing. Artificial Neural Networks ICANN, pp. 660-69, 2007.
- [3] Eric DAVALO, Patrick NAIM, Des réseaux de neurones. Deuxième édition, deuxième tirage 1993..
- [4] Y. Hen Hu, J.-Neng Hwang, "Handbook of neural network signal processing", CRC Press LLC, United States of America, 2002.
- [5] <http://www.statsoft.fr/concepts-statistiques/reseaux-de-neurones-automatisees/reseaux-de-neurones-automatisees.htm#prevision.2001>.
- [6] B. Krose, P. V. Der Smagt, "Introduction to neural networks", Eighth edition, Amsterdam, 1996
- [7] H. Bruyninckx et J. Shutter. « Introduction to intelligent robotics.», Rapport technique, Katholieke Universiteit Leuven 2001.
- [8] H. A. Talebi, F. Abdollahi, R. V. Patel, K. Khorasani, « Neural Network-Based State Estimation of Nonlinear Systems », Springer, New York, 2010.
- [9] K.S. Narendra and K. Parthasarathy, « Identification and control of dynamical systems using neural networks », 1990.
- [10] KS. Narendra, « Neural networks for identification and control », Center for Systems Science, Yale University, Dec. 1998.
- [11] S. Chen and S. A. Billings, « Neural Networks for nonlinear dynamic system modeling and Identification », 1992.
- [12] R. Haber, L. Keviczky, « Nonlinear system identification. 2. Nonlinear system structure identification », Kluwer Academic Publishers, 1999.
- [13] K. Kara, « Application des réseaux de neurones à l'identification des systèmes non linéaires », Thèse de Magister, Université de Constantine 1995.
- [15] S. ZEGHLACHE, Commande Intelligente, cours, université Mohamed Boudiaf-m'sila. 2008.

[16] BOUHALASSA LOUBNA, Planification de trajectoire d'un robot basée sur les réseaux de neurones et les algorithmes génétiques ,Magister de l'université de sciences et de la technologie d'oran Mohamed Boudiaf ,2010.

[17] W. Khalil « Commande des robots manipulateurs», Hermes Science Publications, 2002.

[18] A.AIB, Modélisation des robots manipulateurs, cours, université Mohamed boudiaf-m'sila.2017.

[19] S.BORSALI, modélisation des robots, cours, université Abou-Bekr Belkaid - Tlemcen.2012.

[20] MELAKHESSOU Lakhdar, « Contrôle et identification des systèmes non linéaires par les techniques neuronales »memoire de Magister, Université de Batna.