

**REPUBLIQUE ALGERIENNE DEMOCRATIQUE ET POPULAIRE
MINISTERE DE L'ENSEIGNEMENT SUPERIEUR ET DE LA
RECHERCHE SCIENTIFIQUE**

UNIVERSITE DE M'SILA

FACULTE DE TECHNOLOGIE

DEPARTEMENT DE GENIE ELECTRIQUE



DOMAINE : SCIENCES ET TECHNOLOGIE

FILIERE : GENIE ELECTRIQUE

OPTION : AUTOMATIQUE ET SYSTEMES

**MEMOIRE DE FIN D'ETUDES EN VUE DE L'OBTENTION DU
DIPLOME DE MASTER EN GENIE ELECTRIQUE
SPECIALITE : AUTOMATIQUE ET SYSTEMES**

Par : AICHE Ilyes

THEME

COMMANDE DES SYSTEMES NON LINEAIRES PAR LA TECHNIQUE DEEP LEARNING

Soutenu devant le jury composé de :

.....	Université Mohamed Boudiaf - M'sila	Président
.....	Université Mohamed Boudiaf - M'sila	Examineur
Pr. BENSLIMANE TARAK	Université Mohamed Boudiaf - M'sila	Encadreur

Année universitaire : 2019 / 2020

Remerciements

Ma gratitude doit d'abord être
exprimée envers le bon Dieu « الله »
le tout puissant, qui m'a donné la
sagesse, la patience, le courage et la
volonté pour qu'on puisse terminer
ce travail

Je tiens à remercier tous ceux
qui ont contribué de près ou de loin,
à la réalisation de ce travail, en
Particulier Mr BENSLIMANE

Mes sincères remerciements
S'adressent aussi aux membres de
jury d'avoir accepté de juger mon travail.

je remercie aussi tous les
enseignants, sans exception, car c'est
grâce à eux que je suis arrivé à faire
ce travail.

Sans oublier toute la
famille mon père, ma mère,
mes frères et sœurs et mes
amis.

Dédicace

Je dédie ce modeste travail à mes chers parents, merci pour tous les sacrifices consentis afin de me permettre d'être ce que je suis ; Je leur exprime humblement ma profonde gratitude et mon attachement éternel.

À mes chers frères : Abdelhak et taher et tous mes sœurs

À toute la grande famille AICHE, surtout : Khair-Eddine et Mohiédine

À mes chères amies surtout : Saad, Salahelddine et tous groupe K24

À toute la promotion 2020 de la spécialité automatique et systèmes.

AICHE ILYÈS

SOMMAIRE

Sommaire

CHAPITE I : État de l'art sur les pendules inversé

I.1	Introduction	3
I.2	Pendule inversé.....	3
I.2.1	proprietes du pendule inverse.....	4
I.2.2	principe de fonctionnement	4
I.2.3	exemples d'applications.....	5
I.3	la modélisation	7
I.3.1	modélisation de système chariot-pendule a deux degrés de liberté	7
I.3.2	modélisation de machine à courant continue	10
I.3.3	modélisation du système global	12
I.3.4	simulation de machine à courant continue	13
I.3.5	simulation du système global	16
I.4	conclusion.....	18

CHAPITE II : Reseaux de neurones artificielle et apprentissage profond

II.1	introduction.....	19
II.2	réseaux de neurone artificiel et biologique.....	19
II.2.1	neurone biologique	20
II.2.2	neurone artificielle	20
II.2.3	structure de base	21
II.2.4	historique	22
II.2.5	composition d'un réseau de neurone (RNA)	22
II.2.6	fonctions d'activations	23
II.2.7	topologie de RNA	23

II.2.7.1	réseaux non bouclé (FEED-FORWORD)	24
II.2.7.2	réseaux bouclé (RECURENT)	24
II.2.8	apprentissage des réseaux de neurones	25
II.2.8.1	apprentissage supervisé	25
II.2.8.1	apprentissage non supervisé	25
II.2.9	algorithme d'apprentissage de RNA	26
II.2.9.1	algorithme de rétro-propagation du gradient.....	26
II.2.9.2	exemple d'application de rétro propagation	28
II.2.10	domaines d'applications des RNA	29
II.2.10.1	régression non linéaire ou modélisation des données statiques	29
II.2.10.2	modélisation de processus dynamiques non linéaires	29
II.2.10.3	commande de processus	30
II.2.10.4	classification	30
II.3	commande et identification des processus par RNA	30
II.3.1	identification par RNA	30
II.3.1.1	identification directe	31
II.3.1.2	identification inverse	31
II.3.2	commande par RNA	32
II.3.2.1	commande directe basée sur l'inverse du système	32
II.3.2.2	commande inverse-directe	33
II.3.2.3	apprentissage d'un contrôleur conventionnel ;.....	33
II.4	avantages et inconvénients des réseaux de neurones	34
II.4.1	avantages des réseaux de neurones	34
II.4.2	inconvénients des réseaux de neurones	34
II.5	apprentissage profond ou « DEEP LEARNING »	34
II.6	conclusion.....	36

CHAPITE III : Reseaux de neuron artificiel pour stabilisation de pendule inverse

III.1	introduction.....	37
III.2	stabilisation de pendule inversé par feedback linéarisant	37
III.2.1	mise en équation du contrôle par feedback linéarisant	37
III.2.2	élaboration du contrôleur avec Simulink.....	38
III.3	stabilisation de pendule inverse par une contrôleur à base de réseau de neurone artificielle	39
III.3.1	apprentissage supervisé hors ligne.....	40
III.3.2	résultats de simulation.....	40
III.3.3	test de robustesse de l'apprentissage hors ligne avec l'architecture 9.....	54
III.4	conclusion.....	55

LISTE DES FIGURES

Liste des figures

Figure I.1 : Photo d'un pendule inversé classique avec ses composantes.....	1
Figure I.2 : schéma d'un pendule inversé	3
Figure I.3 : Corps d'être humain vu comme pendule inversé	3
Figure I.4 : Robot BIPED en 3D	4
Figure I.5 : Robot JOE en 3D	4
Figure I.6 : Pendule gyroscope	4
Figure I.7 : Synoptique de la maquette du pendule inversé	4
Figure I.8 : Schéma de l'ensemble chariot et pendule	5
Figure I.9 : Schéma électrique et mécanique d'induit de la MCC	8
Figure I.10 : Schéma bloc du MCC.	10
Figure I.11 : Schéma global de l'ensemble pendule-chariot-moteur	10
Figure I.12 : Schéma Matlab/Simulink du modèle moteur à courant continu	12
Figure I.13 : Vitesse rotorique du MCC avec $V_c=2V$	12
Figure I.14 : Couple moteur.....	12
Figure I.15 : Courant d'induit.....	12
Figure I.16 : Vitesse rotorique du MCC avec couple résistant à $t=0.5\text{seconds}$	13
Figure I.17 : Couple moteur.....	13
Figure I.18 : Courant d'induit	13
Figure I.19 : modèle de l'ensemble chariot-moteur-pendule dans Matlab/Simulink	14
Figure I.20 : Position du chariot ($\theta = 0$).....	15
Figure I.21 : Angle du pendule ($\theta = 0$).....	15
Figure I.22 : Position du chariot ($\theta = \pi$).....	15
Figure I.23 : Angle du pendule ($\theta = \pi$).....	15
Figure II.1 : Réseaux de neurones biologiques et Réseaux de neurones artificiels.....	19
Figure II.2 : Neurone biologique et neurone artificiel.....	20
Figure II.3 : Modèle d'un neurone artificiel.....	21

LISTE DES FIGURES

Figure II.4 : architecture d'un réseau de neurones.....	23
Figure II.5 : différentes fonctions d'activations.....	23
Figure II.6 : réseau de neurone non bouclé (FEED-FORWARD).....	24
Figure II.7 : réseau de neurone bouclé (recurrent).....	24
Figure II.8 : apprentissage supervisé.....	25
Figure II.9 : apprentissage non supervisé.....	26
Figure II.10 : système d'apprentissage en ligne et hors ligne.....	26
Figure II.11 : algorithme de retro propagation.....	28
Figure II.12 : schéma d'identification directe d'un processus par réseau de neurone.....	31
Figure II.13 : schéma d'identification inverse d'un processus par réseau de neurone.....	31
Figure II.14 : commande basée sur l'inverse de système.....	32
Figure II.15 : commande inverse-direct par réseau de neurone.....	33
Figure II.16 : schéma d'apprentissage d'un correcteur conventionnel.....	33
Figure II.17: Illustration d'un réseau de neurones artificiels multicouches utilisé comme technique d'apprentissage profond	35
Figure III.1 : Modèle Simulink du pendule inversé commandé par un contrôleur feedback linéarisant.....	38
Figure III.2 : Résultats de simulation du pendule inversé commandé par un contrôleur feedback linéarisant	38
Figure III.3 : Modèle Simulink du pendule inversé commandé par un contrôleur feedback linéarisant et celui commandé par réseau de neurone.....	40
Figure III.4 : Architecture du RNA d'une seule couche cachée de 10 neurones utilisé dans l'outil d'apprentissage du RNA « nntraintool ».....	41
Figure III.5 : Courbe de performance (évolution de la valeur de l'erreur quadratique moyenne) du réseau de neurone d'une seule couche cachée de 10 neurones.....	41
Figure III.6 : Position et l'angle du pendule inversé commandé par FEEDBACK et par réseau de neurone d'une seule couche cachée de 10 neurones.....	41
Figure III.7 : Architecture de réseau de 2 couches cachées, 7 neurones dans la 1ère couche cachée et 3 neurones dans la 2ème couche cachée.1.....	42
Figure III.8 : Courbe de performance (évolution de la valeur de l'erreur quadratique moyenne) du réseau de neurone de 2 couches cachées, 7 neurones dans la 1ère couche cachée et 3 neurones dans la 2ème couche cachée.....	42

LISTE DES FIGURES

Figure III.9 : Position et l'angle du pendule inversé commandé par FEEDBACK et par réseau de neurone de 2 couches cachées, 7 neurones dans la 1ère couche cachée et 3 neurones dans la 2ème couche cachée.....	43
Figure III.10 : Architecture de réseau de 2 couches cachées, 6 neurones dans la 1ère couche cachée et 4 neurones dans la 2ème couche cachée.....	43
Figure III.11 : Courbe de performance (évolution de la valeur de l'erreur quadratique moyenne) du réseau de neurone de 2 couches cachées, 6 neurones dans la 1ère couche cachée et 4 neurones dans la 2ème couche cachée.....	44
Figure III.12 : Position et l'angle du pendule inversé commandé par FEEDBACK et par réseau de neurone de 2 couches cachées, 6 neurones dans la 1ère couche cachée et 4 neurones dans la 2ème couche cachée.....	44
Figure III.13 : Architecture de réseau de 2 couches cachées, 5 neurones dans la 1ère couche cachée et 5 neurones dans la 2ème couche cachée.....	45
Figure III.14 : Courbe de performance (évolution de la valeur de l'erreur quadratique moyenne) du réseau de neurone de 2 couches cachées, 5 neurones dans la 1ère couche cachée et 5 neurones dans la 2ème couche cachée.....	45
Figure III.15 : Position et l'angle du pendule inversé commandé par FEEDBACK et par réseau de neurone de 2 couches cachées, 5 neurones dans la 1ère couche cachée et 5 neurones dans la 2ème couche cachée.....	45
Figure III.16 : Architecture de réseau de 2 couches cachées, 4 neurones dans la 1ère couche cachée et 6 neurones dans la 2ème couche cachée.....	46
Figure III.17 : Courbe de performance (évolution de la valeur de l'erreur quadratique moyenne) du réseau de neurone de 2 couches cachées, 4 neurones dans la 1ère couche cachée et 6 neurones dans la 2ème couche cachée.....	46
Figure III.18 : Position et l'angle du pendule inversé commandé par FEEDBACK et par réseau de neurones de 2 couches cachées, 4 neurones dans la 1ère couche cachée et 6 neurones dans la 2ème couche cachée.....	47
Figure III.19 : Architecture de réseau de 2 couches cachées, 3 neurones dans la 1ère couche cachée et 7 neurones dans la 2ème couche cachée.....	47
Figure III.20 : Courbe de performance (évolution de la valeur de l'erreur quadratique moyenne) du réseau de neurones de 2 couches cachées, 3 neurones dans la 1ère couche cachée et 7 neurones dans la 2ème couche cachée.....	48
Figure III.21 : Position et l'angle du pendule inversé commandé par FEEDBACK et par réseau de neurones de 2 couches cachées, 3 neurones dans la 1ère couche cachée et 7 neurones dans la 2ème couche cachée.....	48
Figure III.22 : Architecture de réseau de 3 couches cachées, 7 neurones dans la 1ère couche cachée et 2 neurones dans la 2ème couche cachée et 1 neurone dans la 3ème couche cachée.....	49

LISTE DES FIGURES

Figure III.23 : Courbe de performance (évolution de la valeur de l'erreur quadratique moyenne) du réseau de neurones de 3 couches cachées, 7 neurones dans la 1ère couche cachée et 2 neurones dans la 2ème couche cachée et 1 neurone dans la 3ème couche cachée.....	49
Figure III.24 : Position et l'angle du pendule inversé commandé par FEEDBACK et par réseau de neurones de 3 couches cachées, 7 neurones dans la 1ère couche cachée et 2 neurones dans la 2ème couche cachée et 1 neurone dans la 3ème couche cachée.....	50
Figure III.25 Architecture de réseau de 3 couches cachées, 1 neurone dans la 1ère couche cachée et 2 neurones dans la 2ème couche cachée et 7 neurones dans la 3ème couche cachée.....	50
Figure III.26 : Courbe de performance (évolution de la valeur de l'erreur quadratique moyenne) du réseau de neurones de 3 couches cachées, 1 neurone dans la 1ère couche cachée et 2 neurones dans la 2ème couche cachée et 7 neurones dans la 3ème couche cachée.....	51
Figure III.27 : Position et l'angle du pendule inversé commandé par FEEDBACK et par réseau de neurones de 3 couches cachées, 1 neurone dans la 1ère couche cachée et 2 neurones dans la 2ème couche cachée et 7 neurones dans la 3ème couche cachée.....	51
Figure III.28 : Architecture de réseau de 3 couches cachées, 1 neurone dans la 1ère couche cachée et 7 neurones dans la 2ème couche cachée et 2 neurones dans la 3ème couche cachée.....	52
Figure III.29 : Courbe de performance (évolution de la valeur de l'erreur quadratique moyenne) du réseau de neurones de 3 couches cachées, 1 neurone dans la 1ère couche cachée et 7 neurones dans la 2ème couche cachée et 2 neurones dans la 3ème couche cachée.....	52
Figure III.30 : Position et l'angle du pendule inversé commandé par FEEDBACK et par réseau de réseau de neurones de 3 couches cachées, 1 neurone dans la 1ère couche cachée et 7 neurones dans la 2ème couche cachée et 2 neurones dans la 3ème couche cachée.....	53
Figure III.31 : Courbe de l'angle de pendule commandé par réseau de neurone de type apprentissage hors-ligne avec perturbation à l'instant 25 seconde.....	54

LISTE DES TABLEAUX

Liste des tableaux

Tableau (I.1) : Valeurs numériques du système pendule-chariot.....	7
Tableau (I.2) : Les paramètres du MCC.....	12
Tableau (II.1) : Analogie entre composants neuronaux biologiques et formels.....	21
Tableau (III.1) : Erreurs quadratiques moyennes après 1000 itérations lors de l'apprentissage des différentes architectures utilisées pour la commande du pendule inversé avec le même nombre totale de neurones.....	53

ABREVIATIONS ET SYMBOLES

Abréviations et symboles

MCC : machine à courant continu

SIMO : Single Input Multiple Output

CMG : Control Moment Gyro

m : masse du pendule

M : Masse de chariot

l : Demi-longueur de pendule

d : Coefficient de frottement de rotation du pendule

g : Intensité de la pesanteur (gravité)

b : Coefficient de frottement des roues du chariot

$x(t)$: Position du chariot

$\dot{x}$: La vitesse du chariot

$\ddot{x}$: L'accélération du chariot

$\theta(t)$: Angle de rotation du pendule

$\dot{\theta}$: La vitesse angulaire de pendule

$\ddot{\theta}$: L'accélération du pendule

$F(t)$: Force exercée sur le chariot

E_{cM} : L'énergie cinétique du chariot

E_{cm} : L'énergie cinétique du pendule

E_c : L'énergie cinétique totale de l'ensemble chariot et pendule

E_p : L'énergie potentielle totale de l'ensemble chariot et pendule

R_a : Résistance de l'induit

L_a : Inductance de l'induit

J_m : Moment d'inertie

f_m : Coefficient de frottement visqueux

K_b : Constante électrique du moteur

ABREVIATION ET SYMBOLES

k_m : Constante mécanique

c_r : Couple résistant

c_m : Couple moteur

v_c : La tension d'alimentation de l'induit du moteur

i_a : Intensité du courant

e : F.c.é.m

Ω_m : vitesse angulaire de moteur

r : Le rayon de la poulie

$\vec{r}_c$: La position du centre de gravité du pendule

$\vec{v}_c$: La vitesse du centre de gravité du pendule

$\vec{i}$: Vecteur unitaire du repère x

$\vec{j}$: Vecteur unitaire du repère y

L : L'équation générale d'Euler-Lagrange

RNA : réseaux de neurone artificiel

RNI : réseaux de neurone indicateur

RNC : réseaux de neurone contrôleur

W : les poids de réseau

b : les biais de réseau

f : fonction d'activation

E_m : Erreur quadratique commise au niveau de la couche de sortie du réseau

N : Nombre d'exemple dans la base d'apprentissage.

η : Taux d'apprentissage.

INTRODUCTION GENERALE

Introduction générale

Depuis toujours, l'être humain a cherché à rendre sa vie plus facile, c'est-à-dire avoir tous ses besoins en minimisant ses tâches et ses efforts. L'automatique est un domaine qui a largement contribué pour réaliser cet objectif, et cela quel que soit la discipline (électronique, mécanique, physique, chimie, biologie, écologie, économie, agronomie ...etc.). Basée sur la notion de retour d'information, la conception des systèmes automatisés se fait de manière générale sur trois phases dont le premier est la modélisation, ensuite vient l'analyse, celle-ci se consacre essentiellement à la synthèse d'un organe maître qui est le correcteur, ce dernier est la troisième et dernière phase.

Les exigences accrues de l'industrie sont à l'origine de l'utilisation de techniques de réglage et de commande de plus en plus performantes. Ceci est dû d'une part, à l'évolution de l'automatique par l'introduction de méthodes intelligentes de commande des systèmes de plus en plus complexes. Ce qui a permis la conception des contrôleurs qui permettent la mise en pratique de ces techniques. Un de ces commandes intelligent est le contrôleur neuronal.

L'utilisation des réseaux de neurones s'est avérée extrêmement réussie et appropriée, particulièrement dans le domaine de l'Automatique pour la résolution des problèmes de modélisation, d'identification et de commande des systèmes notamment non linéaires et complexes, ainsi que le de traitement de données complexes (apprentissage profond (deep learning)) [1].

Par définition, un système non linéaire est un système régi par un ensemble d'équations différentielles ordinaires à coefficients non constants. Il décrit l'évolution temporelle des variables d'état d'un système physique. Large définition explique la complexité, la diversité des systèmes non linéaires ainsi que la variété des méthodes qui s'y appliquent [2]. Comme un exemple de système non linéaire complexe, on adopte, dans ce mémoire, le pendule inversé.

Le pendule inversé a fait l'objet d'une grande discussion tout au long de ces dernières années. Cet intérêt est dû au fait que le problème de la commande du pendule inversé est fondamentalement le même que ceux impliqués dans plusieurs autres systèmes tels que le lancement des fusées, la propulsion des missiles et la stabilisation des satellites [3]. Le pendule inversé a toujours été utilisé pour tester les nouvelles méthodes de commande, car il possède plusieurs caractéristiques attrayantes : c'est un système non linéaire, couplé, intrinsèquement instable et il possède plusieurs implications pratiques [4].

L'objectif principal de cette étude est l'application de contrôleur neuronal avec différentes architectures (mono couche cachée et multicouches cachées (une des techniques de l'apprentissage profond (deep learning)) pour la stabilisation d'un pendule inversé. Dans un premier temps il s'agit de la synthèse d'une loi de commande par FEEDBACK linéarisation.

Dans une seconde étape, il s'agit de faire l'apprentissage du contrôleur neuronal à partir des résultats obtenus de la commande par feedback linéarisation par création d'une base de données

qui contient les entrées et les sorties. Enfin, une étude comparative des différents contrôleurs neuronaux est détaillée.

L'illustration de ce travail a été organisée en trois chapitres :

Le premier chapitre est consacré à la modélisation du pendule inversé et le moteur à courant continu qui l'entraîne, pour obtenir des modèles qu'on a utilisés pour la simulation du système en boucle ouverte.

Dans le deuxième chapitre, on introduit le réseau de neurones artificiel, ainsi que les étapes à suivre pour l'identification et la commande.

Le troisième chapitre est consacré à la conception du contrôleur neuronal avec différentes architectures à base des données de simulation du pendule inverse commandé par un FEEDBACK linéarisation dans un état stationnaire du chariot. Enfin, une étude comparative des caractéristiques des différents contrôleurs neuronaux est développée.

CHAPITRE I
ETAT DE L'ART
SUR LES
PENDULES
INVERSE

I.1 INTRODUCTION

Dans le domaine scientifique, l'automatique a souvent recours à des cas d'études particuliers, qui sont représentatifs de grandes classes d'applications, et dont le caractère spectaculaire est confirmé. De plus, avec l'expérience, la connaissance de ces cas s'est affinée et ils fournissent aujourd'hui une base idéale pour comparer de façon valable les avantages et les inconvénients d'approches différentes. Le pendule inversé est un de ces cas-types. [5]

Dans ce chapitre on s'intéresse à l'étude du pendule inversé. Nous expliquons ses propriétés et le principe de fonctionnement avec des exemples d'application, par la suite nous élaborons son modèle mathématique en se basant sur le formalisme d'Euler-Lagrange. Enfin nous élaborons le modèle de la machine à courant continu (MCC) et ses résultats de simulation. Finalement nous présentons le système global pendule-chariot-MCC dans l'espace d'état ainsi que les résultats de la simulation en boucle ouverte (BO).

I.2 PENDULE INVERSE

Le pendule inversé est un système instable, SIMO (Single Input Multiple Output) et possède des non-linéarités non négligeables dues à sa structure dynamique et aux forces de friction. C'est un outil didactique et un problème classique utilisé en automatique, où il est considéré comme un très bon exemple pour les systèmes instables et non linéaires. [4]

C'est pourquoi il est souvent utilisé pour tester les performances et la robustesse de nouvelles lois de commande [6]. Par ailleurs, on peut trouver dans la littérature plusieurs travaux appropriés à la commande d'un simple pendule inversé par des lois basées sur un problème de contrôle de l'énergie associées à une séquence de balancement qui permet de relever le pendule de sa position d'équilibre stable. [7]

Il est composé de deux éléments principaux :

- un chariot libre en translation le long d'un rail de guidage.
- un pendule pesant solidaire du chariot et libre en rotation.

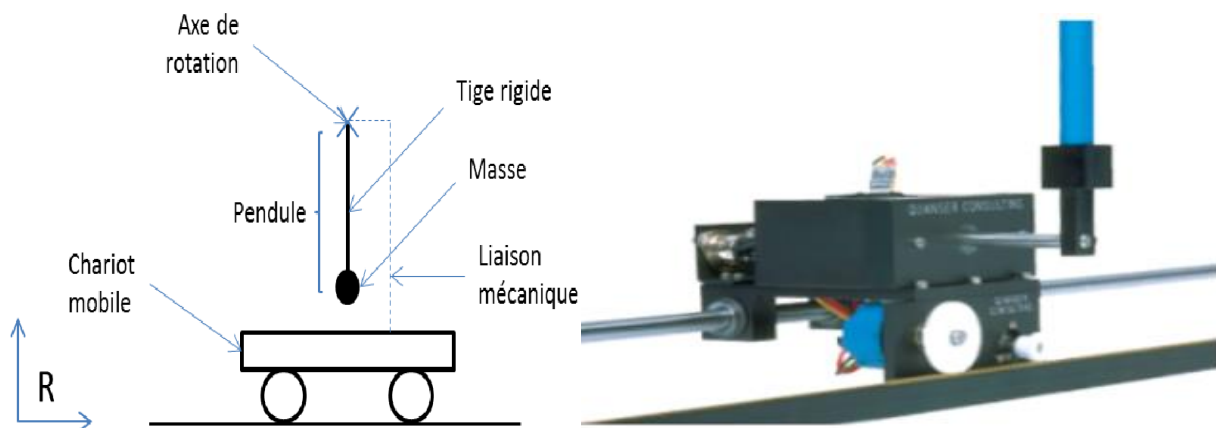


Figure (I.1) : Photo d'un pendule inversé classique avec ses composantes.

On peut trouver divers types de pendules inversés tels que : le pendule inversé simple, le pendule inversé double, le pendule inversé de Furuta, etc. [8]

I.2.1 PROPRIETES DU PENDULE INVERSE [9]

❖ *Système instable :*

Il est très clair expérimentalement que le système est instable dans la position haute du pendule.

❖ *Système sous actionné :*

Ce qui veut dire que le nombre d'actionneurs (un seul moteur) est inférieur au nombre de variables de configuration (x et θ , respectivement mouvement de translation et mouvement angulaire).

Dans ce cas, le système est dit : **sous actionné**.

❖ *Système holonome :*

Un système mécanique S est holonome, si la position de ses différentes parties peut être caractérisée par n variables indépendantes $q_1 \dots q_n$ appelées coordonnées généralisées du système. On dit alors que S est un système holonome à n degrés de liberté.

Le pendule inversé est également un système holonome avec les coordonnées généralisées :

$$q_1=x \text{ et } q_2=\theta$$

❖ *Système non linéaire :*

Notre système (pendule inversé) contient des termes de couplage entre les composantes de l'état tel que $\sin x_3 x_4^2$ ce sont les non linéarités du système.

I.2.2 PRINCIPE DE FONCTIONEMENT

Ce système mécanique consiste en un chariot de masse (M) qui peut se déplacer horizontalement et librement sur un rail de guidage en supportant une tige de masse (m) libre en rotation autour d'un pivot (Figure (I.2)).

L'application d'une force $f(t)$ va provoquer le déplacement du chariot $x(t)$ et la déviation du pendule d'un angle $\theta(t)$ par rapport à la verticale. Ceci constitue un système mécanique sous actionné à deux degrés de liberté et un seul actionneur.

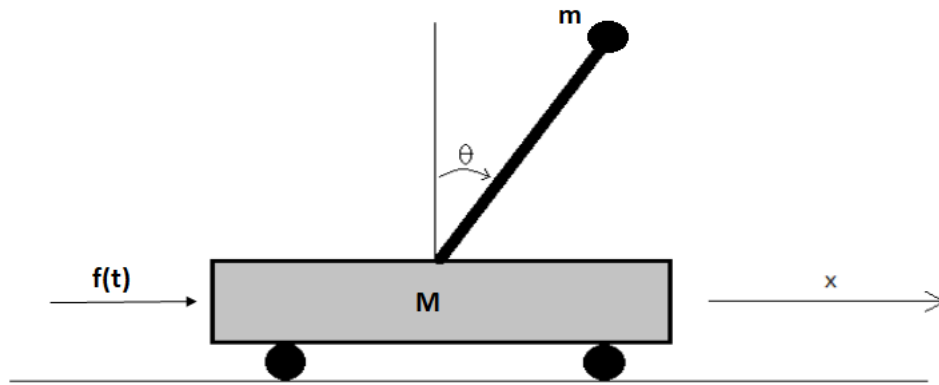


Figure (I.2) : schéma d'un pendule inversé

Le but de la commande serait de ramener le pendule à partir d'une position initiale à sa position d'équilibre instable et le maintenir autour de cette position [10]. Pour cela, quand le pendule penche vers la droite, le chariot doit le rattraper en effectuant un mouvement vers la droite, et inversement, La difficulté c'est de régler l'intensité et la forme de la réaction du chariot en fonction de l'angle que le pendule fait avec la verticale [11]. Pour la simplicité, souvenez-vous quand étiez petit et que vous essayez de tenir en équilibre une tige sur votre doigt. Vous deviez en permanence ajuster la position de votre main pour stabiliser la tige .le pendule inversé fait la même chose, la seule différence étant que son support (chariot) se translate dans une seule direction, alors que le doigt ce déplace dans toutes les directions.

I.2.3 EXEMPLES D'APPLICATIONS

➤ En médecine

L'homme est vu comme un double pendule inversé en série, dont les deux axes de rotations principales sont les chevilles et les hanches. Quand nous sommes en position debout, nos articulations travaillent sans arrêt pour nous y maintenir. Les spécialistes qui travaillent à la réalisation de prothèses (remplacement chirurgical d'un organe ; la pièce ou l'appareil de remplacement : prothèse dentaire) pour les hanches sont amenés à utiliser le modèle de double pendule inversé pour calculer l'ensemble des contraintes qui sont soumises à la prothèse [12] comme le montre la figure suivante.

Le premier pendule est articulé à la cheville et représente les membres inférieurs considérés groupés. Le second pendule est articulé à la hanche et représente la partie supérieure du corps. On accélère en se penchant avant et on ralentit en se penchant en arrière. Le système est le même que le pendule inversé.

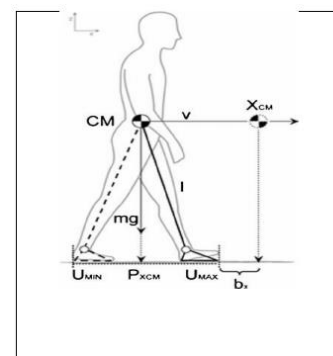


Figure (I.3) : Corps d'être humain

Vu comme pendule inversé.

➤ En robotique

Dans le même ordre d'idée, un robot est vu comme un pendule inversé. Pour maintenir le robot en équilibre il suffit de stabiliser le pendule inversé correspondant à sa position verticale au cours de son déplacement et en présence d'autres perturbations. [13]

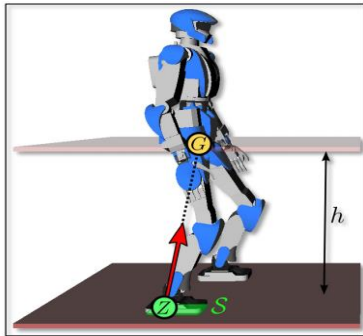


Figure (I.4) : Robot BIPED en 3D



Figure (I.5) : Robot JOE en 3D

➤ Dans l'aérospatiale

Dans ce domaine aussi, l'étude des systèmes à pendules a une grande importance, par exemple pour commander et stabiliser l'attitude du satellite, le lancement des fusées ...etc. Pour stabiliser l'attitude d'un satellite par les actionneurs gyroscopiques, les actionneurs gyroscopiques également appelés girodynes (CMG : Control Moment Gyro) constituent un nouveau système d'actionneurs qui permettent de générer des couples dynamiques de commande de basculement d'attitude d'un satellite. L'expérience consiste à le rappeler vers la position d'équilibre (instable), donc il est vu comme un pendule inversé. [14]



Figure (I.6) : Pendule gyroscope

➤ Dans la recherche

Vu leur complexité, les systèmes à pendules sont très utilisés pour tester des lois de commande modernes. Le schéma de la figure (I.7) ci-dessous représente la maquette de la pendule inversé.

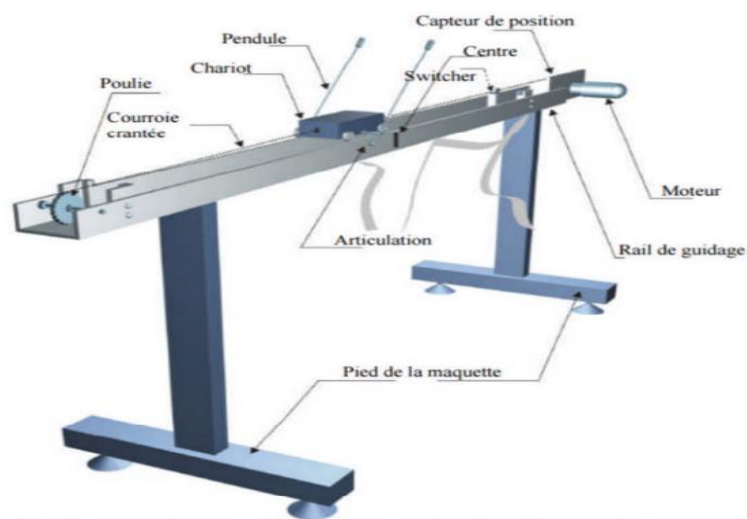


Figure (I.7) : Synoptique de la maquette du pendule inversé

I.3 MODELISATION

L'ensemble du chariot et pendule à deux degrés de liberté sont représentées par deux coordonnées généralisées, x pour le déplacement horizontal du chariot, θ pour la rotation du pendule. La direction positive de x est le sens à droite en mètre et celui de l'angle est le sens des aiguilles d'une montre en radian. [15]

Dans cette partie, nous allons établir les équations de mouvement du pendule inversé en se basant sur la figure suivante :

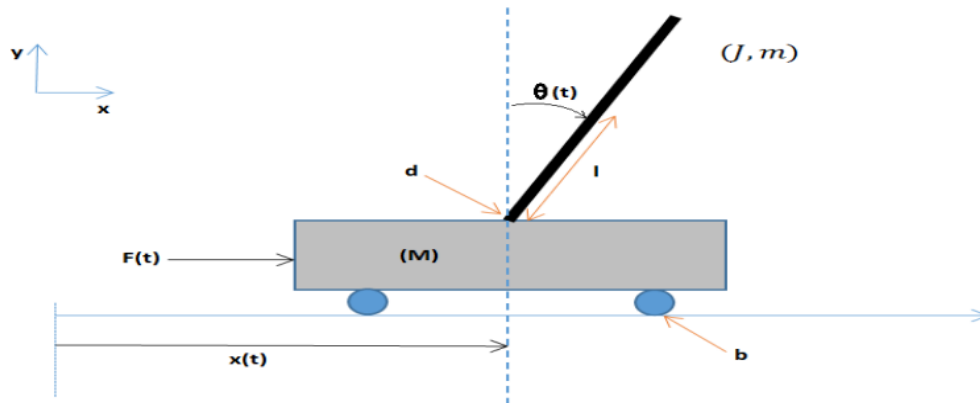


Figure (I.8) : Schéma de l'ensemble chariot et pendule

Les variables et les paramètres du système chariot-pendule sont : [16]

symbole	Quantité	Valeur	Unité
m	masse du pendule	0.2	<i>Kg</i>
M	Masse de chariot	2.3	<i>Kg</i>
l	Demi-longueur de pendule	0.3	<i>m</i>
d	Coefficient de frottement de rotation du pendule	0.005	<i>N ms/rad</i>
g	Intensité de la pesanteur	9.81	<i>m/ s²</i>
b	Coefficient de frottement des roues du chariot	0.00005	<i>Ns/m</i>
J	l'inertie de pendule	0.001	<i>Kg. m²</i>
$\theta(t)$	Angle de rotation du pendule	-----	<i>Radian</i>
$X(t)$	Position du chariot	-----	<i>m</i>
$F(t)$	Force exercée sur le chariot	-----	<i>N</i>

Tableau (I.1) : Valeurs numériques du système pendule-chariot

I.3.1 MODELSATION DE SYSTEME CHARIOT-PENDULE A DEUX DEGRES DE LIBERTE [10][17][11]

Energie cinétique du système en mouvement

L'énergie cinétique totale de notre système est traduite par la somme de l'énergie cinétique totale du chariot et celle du pendule. La cinétique du chariot est son déplacement linéaire sur les rails et celle du pendule est le balancement de ce dernier sur son axe de rotation.

L'énergie cinétique du chariot en mouvement est donnée par l'équation :

$$E_{cM} = \frac{1}{2} M \dot{x}^2 \quad \text{I. 1}$$

L'énergie cinétique du pendule est exprimée par l'équation :

$$E_{cm} = \frac{1}{2} m v_c^2 + \frac{1}{2} J \dot{\theta}^2 \quad \text{I.2}$$

Avec :

v_c : Vitesse du centre de gravité du pendule

$\dot{\theta}$: Vitesse angulaire du pendule

J : Moment d'inertie du pendule

La position du centre de gravité du pendule, notée $\vec{r}_c$ à partir de ces coordonnées, est donnée par :

$$\vec{r}_c = (x + l \sin \theta) \vec{i} + l \cos \theta \vec{j} \quad \text{I.3}$$

$\vec{i}$: Vecteur unitaire du repère x

$\vec{j}$: Vecteur unitaire du repère y

Donc, la vitesse du centre de gravité du pendule est :

$$\vec{v}_c = \frac{d\vec{r}_c}{dt} = (\dot{x} + l \cos \theta \dot{\theta}) \vec{i} - l \dot{\theta} \sin \theta \vec{j} \quad \text{I.4}$$

En remplaçant $\vec{v}_c$ par son équation (I.4) dans l'équation de l'énergie cinétique du pendule (I.2), on obtient :

$$E_{cm} = \frac{1}{2} m (\dot{x}^2 + 2 \dot{x} l \dot{\theta} \cos \theta + l^2 \dot{\theta}^2 \cos^2 \theta + l^2 \dot{\theta}^2 \sin^2 \theta) + \frac{1}{2} J \dot{\theta}^2 \quad \text{I.5}$$

Après simplification (vu que $(\cos^2 \theta + \sin^2 \theta) = 1$), on obtient :

$$E_{cm} = \frac{1}{2} m (\dot{x}^2 + 2 \dot{x} l \dot{\theta} \cos \theta + l^2 \dot{\theta}^2) + \frac{1}{2} J \dot{\theta}^2 \quad \text{I.6}$$

Alors, l'énergie cinétique totale de l'ensemble chariot et pendule est exprimée par :

$$E_c = E_{cm} + E_{cM} = \frac{1}{2} [m (\dot{x}^2 + 2 \dot{x} l \dot{\theta} \cos \theta + l^2 \dot{\theta}^2) + J \dot{\theta}^2 + M \dot{x}^2] \quad \text{I.7}$$

Energie potentielle du système

L'énergie potentielle totale de notre système est traduite seulement par le pendule qui est en mouvement, vu que le chariot est en mouvement sur un rail horizontal.

L'énergie potentielle du centre de gravité du pendule est donné par :

$$E_p = m g l \cos \theta \quad \text{I.8}$$

Modèle de connaissance de l'ensemble chariot et pendule

Maintenant que l'expression de l'énergie cinétique totale (I.7) et l'expression de l'énergie potentielle totale (I.8) sont établies, on utilise l'équation générale d'Euler-Lagrange pour déterminer les équations du mouvement de l'ensemble chariot-pendule.

$$L = \frac{1}{2} [m(\dot{x}^2 + 2\dot{x}l\dot{\theta}\cos\theta + l^2\dot{\theta}^2) + J\dot{\theta}^2 + M\dot{x}^2] - mgl\cos\theta \quad \text{I.8}$$

Donc, à partir de l'équation de Lagrange suivante (I.10), on définit les équations de Lagrange par rapport à chaque degré de liberté pour le pendule inversé :

$$F_i = \frac{d}{dt} \frac{\partial L}{\partial \dot{q}_i} - \frac{\partial L}{\partial q_i} \quad \text{I.10}$$

▪ **Premier degrés de liberté : $q(t)=x(t)$**

$$F - b\dot{x} = \frac{d}{dt} \frac{\partial L}{\partial \dot{x}} - \frac{\partial L}{\partial x} \quad \text{I.11}$$

F : Force exercée sur le chariot

La dérivée partielle du Lagrangien par rapport à $\dot{x}$ et t s'écrit :

$$(M + m)\ddot{x} + ml\ddot{\theta}\cos\theta - ml\dot{\theta}^2\sin\theta = F - b\dot{x} \quad \text{I.12}$$

▪ **Deuxième degrés de liberté : $q(t)=\theta(t)$**

$$-d\dot{\theta} = \frac{d}{dt} \frac{\partial L}{\partial \dot{\theta}} - \frac{\partial L}{\partial \theta} \quad \text{I.13}$$

La dérivée partielle du Lagrangien par rapport à $\dot{\theta}$ et t s'écrit :

$$(ml^2 + I)\ddot{\theta} + ml\ddot{x}\cos\theta + ml\dot{x}\sin\theta\dot{\theta} - ml\dot{x}\sin\theta\dot{\theta} - mgl\sin\theta = -d\dot{\theta} \quad \text{I.9}$$

Alors, le modèle de connaissance de l'ensemble chariot et pendule est décrit par le système d'équations suivant :

$$\begin{cases} (M + m)\ddot{x} + b\dot{x} + ml\ddot{\theta}\cos\theta - ml\dot{\theta}^2\sin\theta = F \\ (ml^2 + J)\ddot{\theta} + d\dot{\theta} + ml\ddot{x}\cos\theta - mgl\sin\theta = 0 \end{cases} \quad \text{I.10}$$

D'après le système d'équations (I.15), on peut remarquer la relation qui existe entre l'accélération du chariot $\ddot{x}$ et l'accélération angulaire du pendule $\ddot{\theta}$ qui est une relation de dépendance. Ainsi, lorsqu'il n'y a pas de force extérieure ($F=0$), si on déplace le pendule de sa position d'équilibre, il ne sera soumis qu'à sa propre inertie.

Le système d'équations (I.15) peut être écrit sous la forme de Gauchy (modèle d'état non linéaire).pour cela, on définit les variables d'état comme suit :

$$\begin{cases} x_1 = x \\ x_2 = \dot{x} \\ x_3 = \theta \\ x_4 = \dot{\theta} \end{cases} \Rightarrow \begin{cases} \dot{x}_1 = x_2 \\ \dot{x}_2 = \ddot{x} \\ \dot{x}_3 = x_4 \\ \dot{x}_4 = \ddot{\theta} \end{cases} \quad \text{I.16}$$

Après quelques manipulations mathématiques, on obtient le modèle non-linéaire (I.17), où la grandeur de commande est la force F .

$$\left\{ \begin{array}{l} \dot{x}_1 = x_2 \\ \dot{x}_2 = \frac{1}{h}F - \frac{b}{h}x_2 - \frac{m^2 g l^2 \sin x_3 \cos x_3}{h \left(N - \frac{m^2 l^2}{h} \cos^2 x_3 \right)} - \frac{m l d \cos x_3}{h \left(N - \frac{m^2 l^2}{h} \cos^2 x_3 \right)} x_4 - \frac{m^2 l^2 \cos^2 x_3}{h^2 \left(N - \frac{m^2 l^2}{h} \cos^2 x_3 \right)} F \\ \quad - \frac{m^3 l^3 \sin \theta \cos^2 x_3}{h^2 \left(N - \frac{m^2 l^2}{h} \cos^2 x_3 \right)} x_4^2 + \frac{m^2 l^2 b \cos^2 x_3}{h^2 \left(N - \frac{m^2 l^2}{h} \cos^2 x_3 \right)} x_2 + \frac{m l}{h} \sin x_3 x_4^2 \\ \dot{x}_3 = x_4 \\ \dot{x}_4 = \frac{m g l \sin x_3}{N - \frac{m^2 l^2}{h} \cos^2 x_3} - \frac{d}{N - \frac{m^2 l^2}{h} \cos^2 x_3} x_4 - \frac{m l \cos x_3}{h \left(N - \frac{m^2 l^2}{h} \cos^2 x_3 \right)} F \\ \quad - \frac{m^2 l^2 \sin x_3 \cos x_3}{h \left(N - \frac{m^2 l^2}{h} \cos^2 x_3 \right)} x_4^2 + \frac{b m l \cos x_3}{h \left(N - \frac{m^2 l^2}{h} \cos^2 x_3 \right)} x_2 \\ y_1 = x_1 \\ y_2 = x_3 \end{array} \right. \quad \text{I.17}$$

Où $h = M + m$, $N = m l^2 + J$

I.3.2 MODELISATION DE MCC

Comme le moteur utilisé pour entrainer le chariot est un moteur à courant continu à aimant permanent, le flux inducteur est constant. Le schéma électrique et mécanique équivalent de l'induit est donné par la figure (I.9)

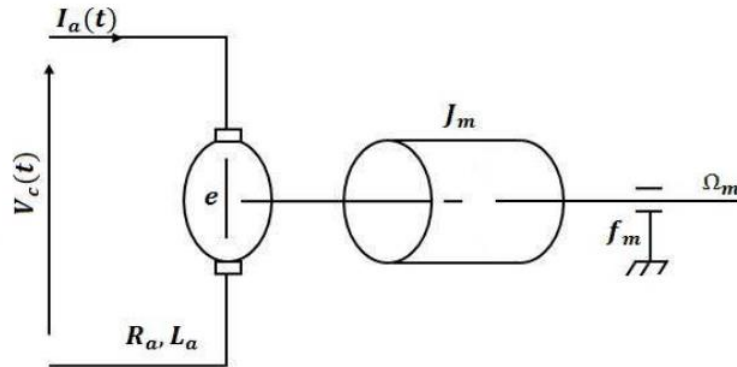


Figure (I.9) : Schéma électrique et mécanique d'induit de la MCC

✓ Equations électriques

Equation de l'induit :

$$v_c(t) = R_a i_a(t) + L_a \frac{di_a(t)}{dt} + e(t) \quad \text{I.18}$$

Equation de la force contre électromotrice :

$$e(t) = K_b \Omega_m(t) \quad \text{I.19}$$

✓ Equations Mécaniques

L'équation mécanique du moteur en rotation :

$$J_m \frac{d\Omega_m(t)}{dt} = c_m(t) - c_r(t) - f_m \Omega_m(t) \quad \text{I.20}$$

L'équation du couple : I.21

$$c_m(t) = k_m i_a(t)$$

Dans le domaine de LAPLACE, on détermine la fonction de transfert reliant la vitesse d'un moteur à courant continu $\Omega_m(t)$ par rapport à la tension d'entrée $v_c(t)$, comme suit :

$$(1) \text{ I.18 } \Rightarrow V_c(P) - E(P) = R_a I_a(P) + P L_a I_a \quad \text{I.22}$$

$$(2) \text{ I.19 } \Rightarrow E(P) = K_b \Omega_m(P) \quad \text{I.23}$$

$$(3) \text{ I.20 } \Rightarrow C_m(P) - C_r(P) = P J_m \Omega_m(P) + f_m \Omega_m(P) \quad \text{I.24}$$

$$(4) \text{ I.21 } \Rightarrow C_m(P) = k_m I_a(P) \quad \text{I.25}$$

D'où :

$$I_a(P) = \frac{\frac{1}{R_a}}{1 + \frac{L_a}{R_a} P} [V_c(P) - E(P)] \quad \text{I.26}$$

$$\Omega_m(P) = \frac{\frac{1}{f_m}}{1 + \frac{J_m}{f_m} P} [k_m I_a(P) - C_r(P)] \quad \text{I.27}$$

Les fonctions de transfert des différentes parties de la MCC sont données par :

Partie électrique :

$$\frac{I_a(P)}{V_c(P) - E(P)} = \frac{\frac{1}{R_a}}{1 + \frac{L_a}{R_a} P} \quad \text{I.28}$$

Partie mécanique :

$$\frac{\Omega_m(P)}{k_m I_a(P) - C_r(P)} = \frac{\frac{1}{f_m}}{1 + \frac{J_m}{f_m} P} \quad \text{I.29}$$

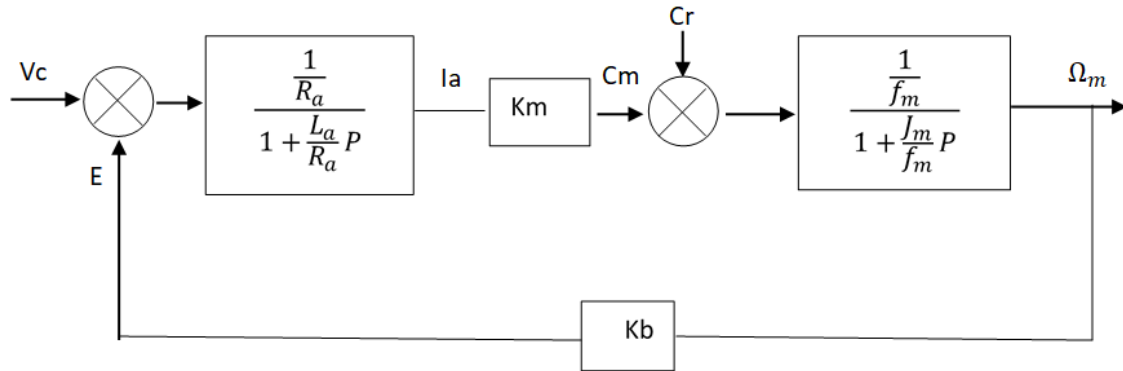


Figure (I.10) : Schéma bloc du MCC.

Les paramètres du moteur sont listés dans le tableau ci-dessous : [18]

Paramètres	Valeurs	Description
R_a	2.5Ω	Résistance de l'induit
L_a	$0.0025H$	Inductance de l'induit
J_m	$1.4 \times 10^{-5} Kg. m^2$	Moment d'inertie
f_m	$10^{-6} Kg. m^2/s$	Coefficient de frottement visqueux
K_b	$0.05N/A$	Constante électrique du moteur
k_m	$0.05Nm$	Constante mécanique
r	$0.0027 m$	le rayon de la poulie
c_r	-----	couple résistant
c_m	-----	couple moteur
$-2.5 \leq v_c \leq 2.5$	-----	La tension d'alimentation de l'induit du moteur
i_a	-----	Intensité du courant (A)
e	-----	F.c.é.m.
Ω_m	-----	vitesse angulaire de moteur

Tableau (I.2) : Les paramètres du MCC

I.3.3 MODELISATION DU SYSTEME GLOBAL

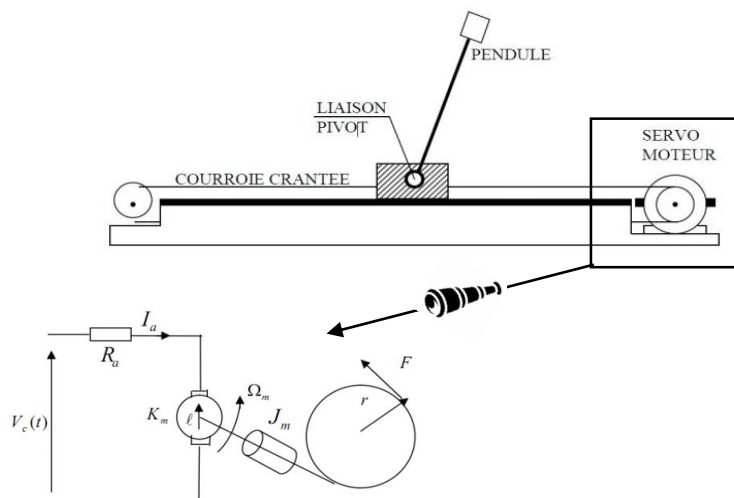


Figure (I.11) : Schéma global de l'ensemble pendule-chariot-moteur

Le schéma de la figure (1.11) illustre la relation existante entre le moteur commandé par la tension V_c et la force F permettant l'entraînement du chariot, donc le balancement du pendule. Pour déterminer l'expression de la force F en fonction de la tension d'alimentation du moteur V_c et afin de ne pas rendre plus complexe le modèle global de l'ensemble pendule-chariot-moteur, on néglige la dynamique du moteur, par conséquence, l'équation électrique (1.20) devient en tenant compte de l'équation (1.21) [19] :

$$V_c = R_a I_a + K_b \Omega_m \quad \text{I.30}$$

Et comme la position angulaire d'un point quelconque située sur le périmètre de la poulie (θ_r) est transformée en un point linéaire x par la relation :

$$x = r\theta_r \quad \text{I.31}$$

r : Le rayon de la poulie

On dérive l'équation (I.31) et obtient :

$$\dot{x} = r\dot{\theta}_r = r\Omega_m \quad \text{I.32}$$

La vitesse angulaire du moteur Ω_m s'exprime donc par rapport à la vitesse du chariot $\dot{x}$ par la relation :

$$\Omega_m = \frac{\dot{x}}{r} \quad \text{I.33}$$

L'expression du courant d'induit I_a s'écrit :

$$I_a(t) = \frac{V_c}{R_a} - \frac{K_b}{rR_a} \dot{x} \quad \text{I.34}$$

Le couple produit à l'arbre du moteur (équation (I.22)) crée une force qui est donnée par :

$$F = \frac{C_m}{r} = \frac{K_m}{r} I_a = \frac{K_m}{rR_a} V_c - \frac{K_m K_b}{r^2 R_a} \dot{x} \quad \text{I.35}$$

I.3.4 SIMULATION DU MCC :

SIMULINK est l'extension graphique de MATLAB permettant de représenter les fonctions mathématiques et les systèmes dynamiques sous forme de schémas en blocs.

Les équations (I.26) et (I.27) (ou même (I.28) et (I.29)) permettent de construire le schéma bloc exprimant l'équation mécanique (la vitesse) et l'équation électrique (courant d'induit) et l'équation électromagnétique (le couple moteur) du moteur à courant continu dans l'environnement Matlab/Simulink. La figure (I.12) présente le schéma bloc sous Simulink du modèle équivalent de moteur à courant continu à aimant permanent.

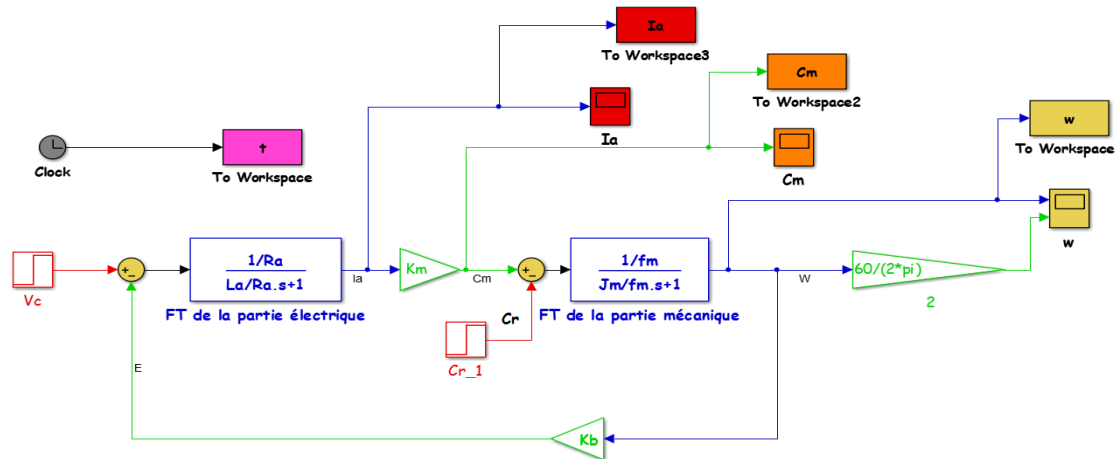


Figure (I.12) : Schéma Matlab/Simulink du modèle moteur à courant continu

Les figures suivantes représentent les résultats obtenus après la simulation de la machine à courant continu avec et sans couple résistant.

🚦 Couple résistant nulle :

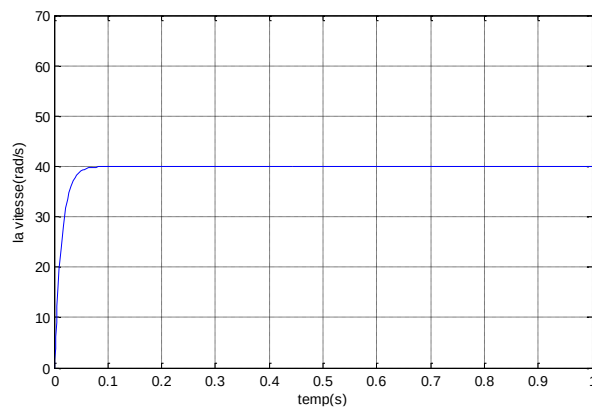
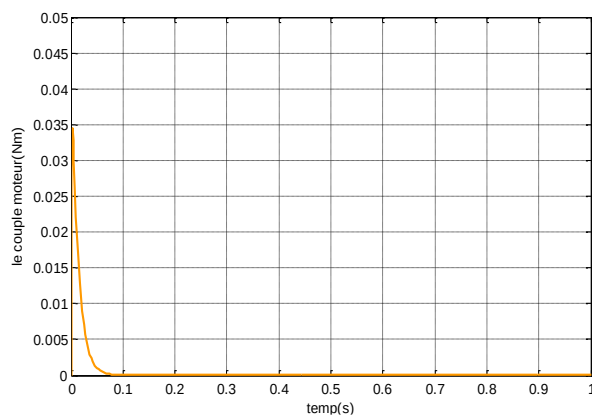
Figure (I.13) : Vitesse rotorique du MCC avec $V_c=2V$ 

Figure (I.14) : Couple moteur

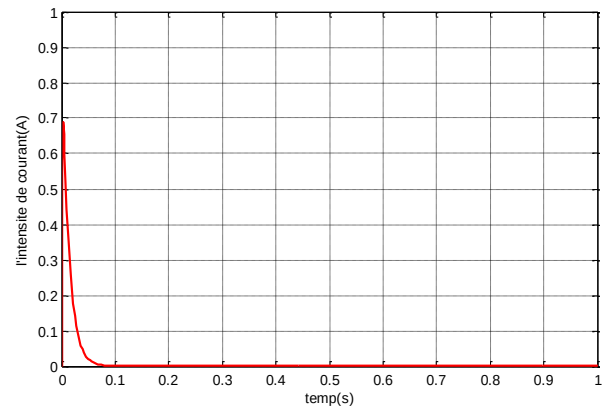


Figure (I.15) : Courant d'induit

✚ Avec couple résistant à $t=0.5$ seconds :

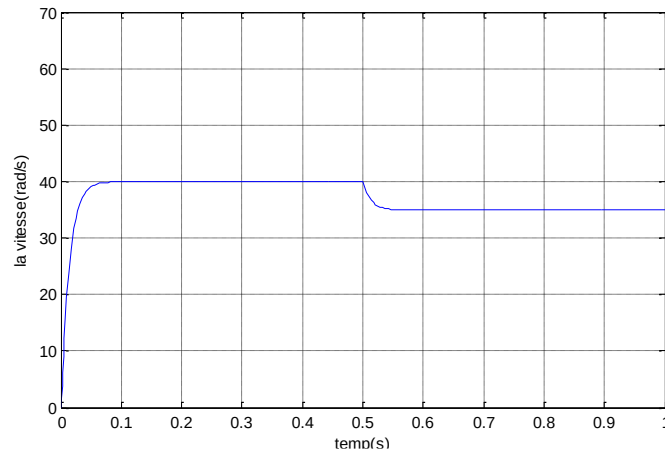


Figure (I.16) : Vitesse rotorique du MCC avec couple résistant à $t=0.5$ seconds

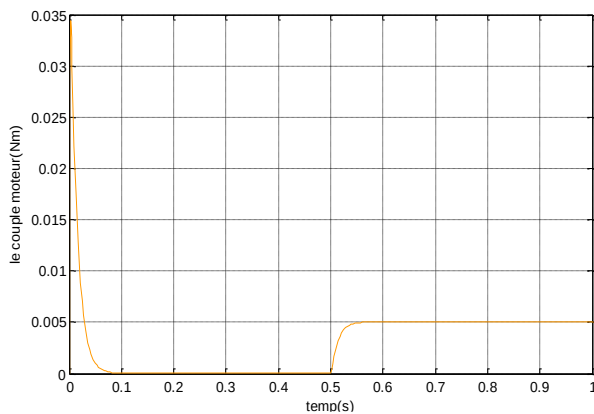


Figure (I.17) : Couple moteur

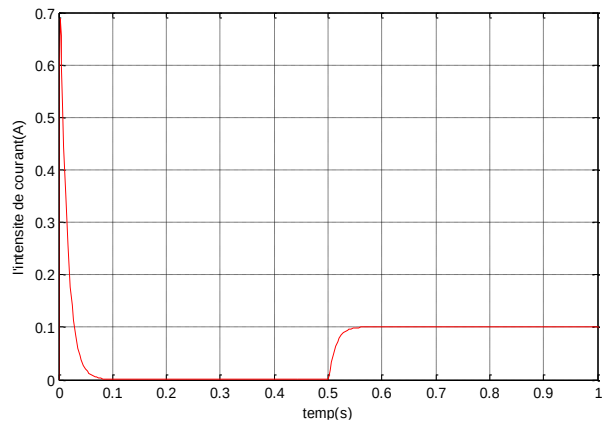


Figure (I.18) : Courant d'induit

INTERPRITATION DE RESULTATS :

Les figures (figure (I.13), figure (I.14) et figure (I.15)) montrent que : Au démarrage du moteur, le rotor (l'induit) absorbe une grande densité de courant pour vaincre sa propre inertie et se faire tourner, d'où l'apparition d'un pic de courant ; et puis lorsque la machine atteint une certaine vitesse stable le courant diminue jusqu'à une valeur proche de zéro comme le montre la figure (I.15). Le moteur n'atteindra pas une vitesse constante directement après son démarrage, mais la vitesse passera par un régime transitoire avant de se stabiliser à une valeur constante.

Après l'application du couple de charge à $t=0.5$ seconds, la vitesse rotorique diminue de 40rad/sec à 35rad/sec, après un régime transitoire, contrairement au courant d'induit qui augmente de 0.007A à 0.1A (figure (I.16), figure (I.17) et figure (I.18)).

I.3.5 SIMULATION DU SYSTEME GLOBAL :

A partir de l'équation de $\ddot{x}$ et $\ddot{\theta}$, on a construit le schéma de simulation sous « Matlab-Simulink » de l'ensemble (chariot-pendule-MCC). Le schéma est présenté par la figure (I.19).

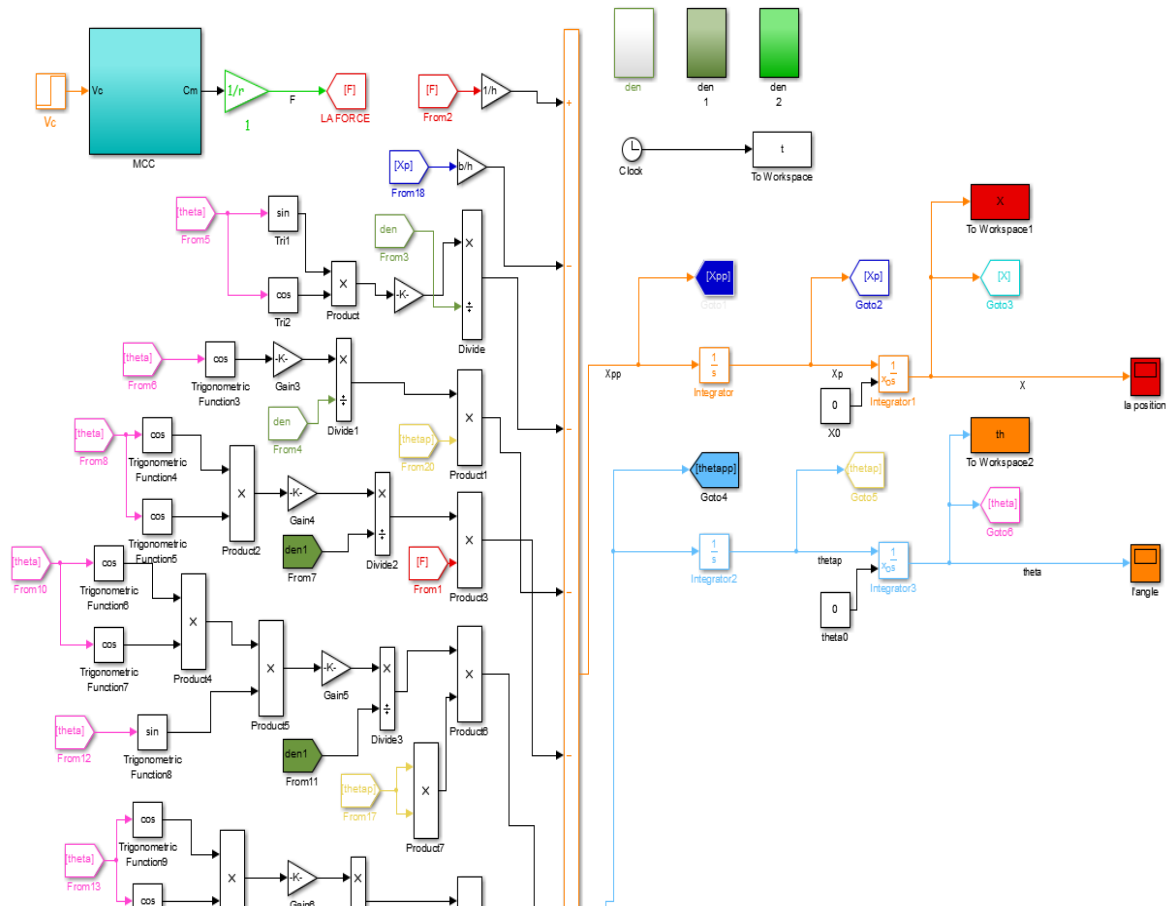


Figure (I.19) : Schéma bloc du modèle de l'ensemble chariot-moteur-pendule dans l'environnement Matlab/Simulink

Le pendule inverse libre (aucune force appliquée) possède deux points d'équilibre. Le premier point ($\theta = 0$) qui correspond à la position verticale de la tige est un point d'équilibre instable ; et le second point ($\theta = \pi$) point d'équilibre stable.

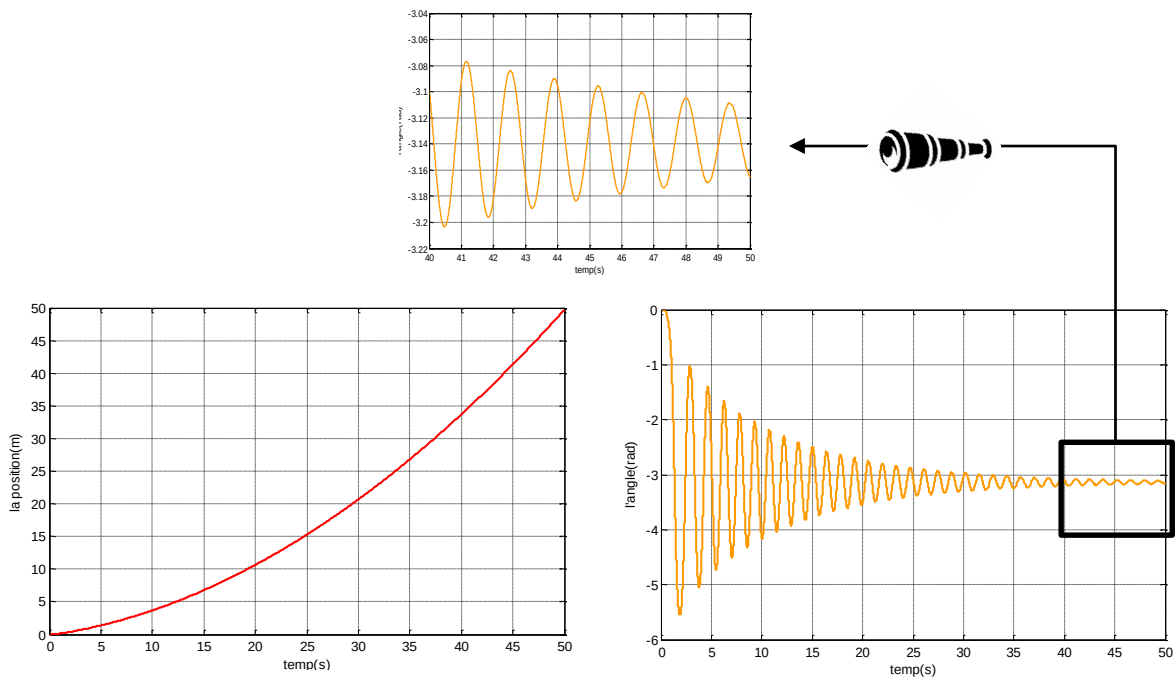
❖ $\theta = 0$ 

Figure (I.20) : Position du chariot

Figure (I.21) : Angle du pendule

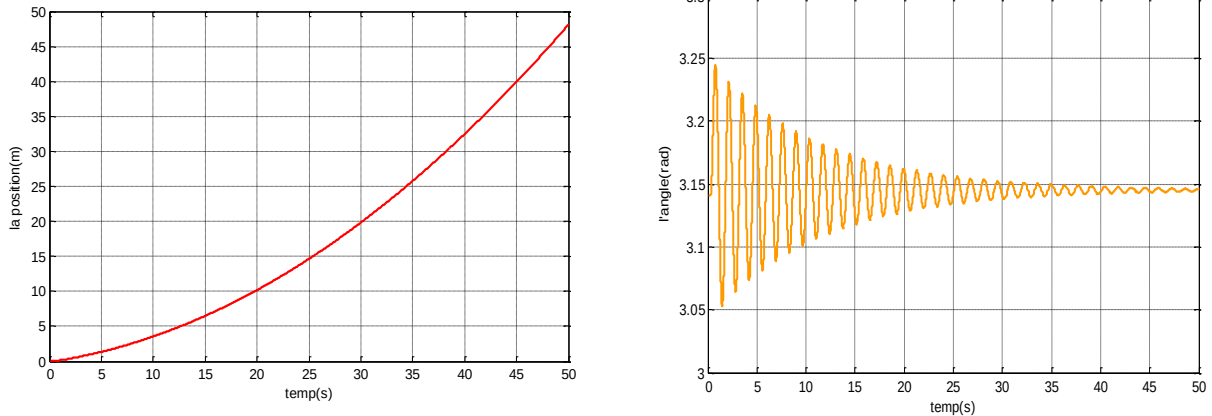
❖ ($\theta = \pi$)

Figure (I.22) : Position de chariot

Figure (I.23) : Angle de pendule

INTERPRITATION DE RESULTATS :

Les figures (1.20) et (1.21) montrent que lorsque le pendule est initialement en position instable haute ($\theta = 0$), il se stabilise après un régime transitoire dans sa position d'équilibre stable ($\theta = \pi = 3.14$) faisant une rotation d'un demi-tour.

Les figures (1.22) et (1.23) montrent que, lorsque le pendule est initialement dans sa position d'équilibre stable ($\theta = \pi$), le chariot continu à évoluer dans les deux cas, quel que soit la position initiale du pendule, à cause de l'effet de balancement du pendule.

I.4 CONCLUSION :

Dans ce chapitre, nous avons présenté le pendule inversé qui est le système non-linéaire choisi pour être modélisé, simulé et commandé. On a présenté ses composants, son principe de fonctionnement et ses propriétés de point de vue automatique. Puis, on a développé le modèle mathématique non linéaire du pendule inversé ; On a utilisé pour ce faire, le formalisme d'Euler-Lagrange en considérant le modèle dynamique du moteur (actionneur). La simulation en boucle ouverte du système a permis de constater sa non linéarité et son instabilité. Ces résultats illustrent la complexité du contrôle des systèmes non-linéaires sous-actionnés tel que le pendule inversé. Le chapitre suivant sera consacré à la présentation de l'une des techniques les plus répandues dans le domaine de la commande intelligente et qui est la commande par réseaux de neurones artificiels.

CHAPITRE II
RESEAUX DE
NEURONES
ARTIFICIEL ET
L'APPRENTISSAGE
PROFOND

II.1 INTRODUCTION

Les réseaux de neurones artificiels (RNA) connaissent depuis quelques années un succès dans divers domaines (physique, chimie, biologie, finance, etc.), notamment pour la modélisation de processus et la synthèse de lois de commandes. [20]

Dans ce chapitre nous allons présenter quelques aspects théoriques liés aux réseaux de neurones artificiels, à titre d'exemple nous citons le neurone biologique, le neurone formel, fonction d'activation, types de réseaux de neurones artificiels, modèles des réseaux de neurones artificiels. Ensuite, nous présentons l'un des processus indispensables pour les réseaux de neurones artificiels qui est l'apprentissage et l'une des techniques les plus répandues qui est l'apprentissage par rétro-propagation. Finalement, on présente les différentes applications du RNA comme technique de modélisation (identification) des systèmes et technique de commande des systèmes.

II.2 RESEAU DE NEURONE ARTIFICIEL ET BIOLOGIQUE

Dans le monde biologique, plusieurs structures neuronales existent et chaque structure est dotée d'une fonction particulière. Le neurone reçoit en continu des entrées, ces informations sont interprétées au niveau du corps cellulaire du neurone. La réponse à ces signaux est envoyée à travers l'axone qui fait synapse sur des milliers d'autres neurones. [21].

Pour les réseaux de neurones artificiels, Kohonen propose la définition suivante : "Les RNA sont des réseaux massivement connectés en parallèle d'éléments simples et leur organisation hiérarchique. Ils sont sensés interagir avec les objets du monde réel de la même manière que les systèmes nerveux biologiques". [21]

D'une autre manière, un réseau de neurones artificiels (RNA), est un ensemble de neurones formels associés en couches (ou sous-groupes) et fonctionnant en parallèle, chaque neurone dans une couche est connecté à tous les neurones de la couche suivante. Dans un réseau de neurones, l'information se propage de la couche d'entrée à la couche de sortie, en passant par une ou plusieurs couches intermédiaires dites couches cachées. La figure. (II.1) montre un RN biologique et un RN Artificiel d'une couche d'entrée, d'une couche cachée et d'une couche de sortie.

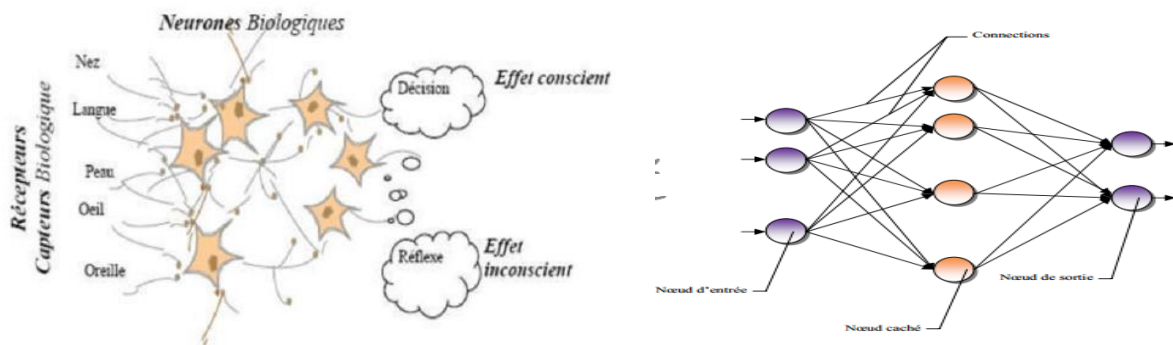


Figure (II.1) : Réseau de neurones biologiques (à gauche), Réseau de neurones artificiels (à droite).

II.2.1 NEURONE BIOLOGIQUE

L'élément de base du système nerveux central est le neurone. Le neurone est une cellule composée d'un corps cellulaire, dendrites, axone et synapse.

- Le corps cellulaire contient le noyau du neurone ainsi que la machine biochimique nécessaire à la synthèse des enzymes. Sa taille est de quelques microns de diamètre
- Les dendrites se sont de fines extensions tubulaires qui se ramifie autour du neurone et forme une sorte de vaste arborescence. Les signaux envoyés au neurone sont captés par les dendrites.
- L'axone : c'est le long de l'axone que les signaux partent du neurone. Contrairement aux dendrites qui se ramifient autour du neurone, l'axone est plus long et se ramifie à son extrémité ou il se connecte aux dendrites des autres neurones.
- Une synapse est une jonction entre deux neurones, et généralement entre l'axone d'un neurone et une dendrite d'un autre neurone. Le corps cellulaire se ramifie pour former les dendrites, c'est par ces dernières que l'information est acheminée de l'extérieur vers le corps cellulaire où elle est traitée. L'information traitée par le neurone chemine ensuite le long de l'axone (unique) pour être transmise aux autres neurones. [22] [23]

II.2.2 NEURONE ARTIFICIELE

Chaque neurone artificiel est un processeur élémentaire. Il reçoit un nombre variable d'entrées en provenance de neurones amont. A chacune de ces entrées est associé un poids (paramètre) w , représentatif de la force de la connexion. Chaque processeur élémentaire est doté d'une sortie unique, qui se ramifie ensuite pour alimenter un nombre variable de neurones avals.

Les réseaux de neurones formels sont issus d'une tentative de modélisation mathématique du cerveau humain, cette modélisation consiste à mettre en œuvre un système de réseau de neurones artificiels qu'il existe pour chaque élément composant le neurone biologique, un élément correspondant dans le neurone artificiel, comme le montre clairement la figure (II.2). [22]

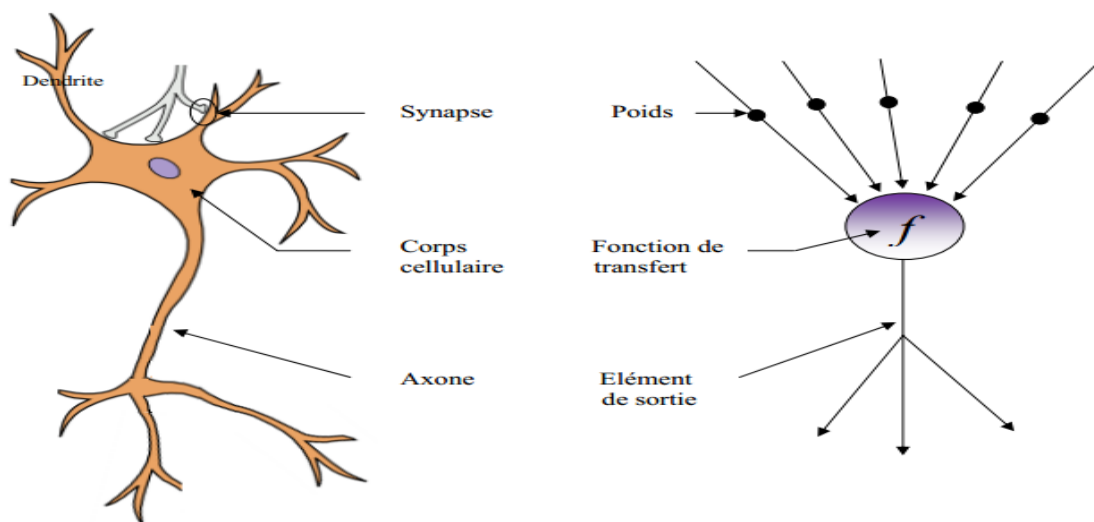


Figure (II.2) : Neurone biologique et neurone artificiel.

II.2.3 STRUCTURE DE BASE

Les réseaux de neurones formels sont à l'origine d'une tentative de modélisation mathématique du cerveau humain. Les premiers travaux datent de 1943 et sont l'œuvre de Mac Culloch et Pitts. Ils représentent un modèle assez simple pour les neurones et explorent les possibilités de ce modèle. La modélisation consiste à mettre en œuvre un système de réseau neuronal sous un aspect non biologique mais artificiel, donc pour chaque élément composant le neurone biologique, il y a son correspondant en neurone formel. Cette modélisation est résumée par le tableau suivant, qui permettra de voir clairement la transition du neurone biologique au neurone formel [24] :

Neurones biologique	Neurones formel
Synapses	poids de connexions
Axones	signal de sortie
Dendrites	signal d'entrée
Noyau	fonction d'activation

Tableau (II.1) : Analogie entre composants neuronaux biologiques et formels. [21]

Le premier modèle du neurone formel date des années quarante. Il été présenté par Mc Culloch et Pitts. S'inspirant de leurs travaux sur les neurones biologiques, ils ont proposés le modèle suivant : [25]

Le neurone formel est un modèle mathématique simplifié du neurone biologique comme illustré à la figure (II.3). Un neurone est essentiellement constitué d'un intégrateur qui effectue la somme pondérée de ses entrées. Le résultat n de cette somme est ensuite transformé par une fonction f qui produit la sortie a du neurone. Les R entrées du neurone correspondent au vecteur $\mathbf{p} = [p_1 p_2 \dots p_R]^T$ alors que $\mathbf{w} = [w_{1,1} w_{1,2} \dots w_{1,R}]^T$ représente le vecteur des poids du neurone [26]

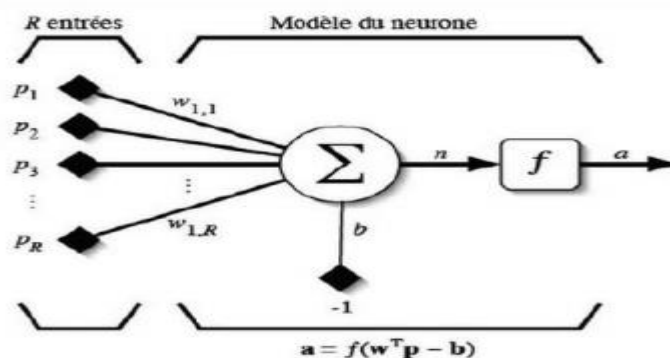


Figure (II.3) : Modèle d'un neurone artificiel.

La sortie n de l'intégrateur est donnée par l'équation suivante :

$$n = \sum_{j=1}^R w_{1,j} p_j - b \quad \text{II.1}$$

Que l'on peut aussi écrire sous forme matricielle :

$$n = w^T p - b \quad \text{II.2}$$

Le biais b s'appelle aussi le seuil d'activation. Lorsque le niveau d'activation atteint ou dépasse le seuil, alors l'argument de f devient positif (ou nul). Sinon, il est négatif. On ajoute la fonction d'activation f pour obtenir la sortie du neurone :

$$a = f(n) = f(w^T p - b) \quad \text{II.3}$$

II.2.4 HISTORIQUE [23]

- 1943 : Mc Culloch et Pitts présentent le premier neurone formel.
- 1949 : Hebb propose un mécanisme d'apprentissage (règle de Hebb).
- 1958 : Rosenblatt présente le premier réseau de neurones artificiels : le Perceptron. Il est inspiré du système visuel, et possède deux couches de neurones : perceptive et décisionnelle. Dans la même période, le modèle de l'ADALINE (ADaptive LINear Element) est présenté par Widrow. Ce sera le modèle de base des réseaux multicouches.
- 1969 : Minsky et Papert publient une critique des perceptrons en montrant leurs limites, ce qui va faire diminuer la recherche sur le sujet.
- 1972 : Kohonen présente ses travaux sur les mémoires associatives.
- 1982 : Hopfield démontre l'intérêt d'utiliser les réseaux récurrents pour la compréhension et la modélisation des fonctions de mémorisation.
- 1986 : Rumelhart popularise l'algorithme de rétropropagation du gradient, conçu par Werbos, qui permet d'entraîner les couches cachées des réseaux multicouches. Les réseaux neuronaux ont été depuis beaucoup étudiés, et ont trouvé énormément d'applications.

II.2.5 COMPOSITION D'UN RESEAU DE NEURONES (RNA) [27]

Un RNA est constitué généralement de trois couches :

- **Couche d'entrée** : Elle est constituée de d'ensemble neurones du réseau qui reçoivent les données du problème. Sa taille est donc déterminée directement par le nombre de variables d'entrée.
- **Couche de sortie** : Elle est constituée de l'ensemble des neurones de sortie du réseau. C'est cette couche-là qui fournit les résultats du problème.
- **Une ou plusieurs couches cachées** : Ce sont les couches qui se trouvent entre la couche d'entrée et la couche de sortie. Elles définissent l'activité interne du réseau.

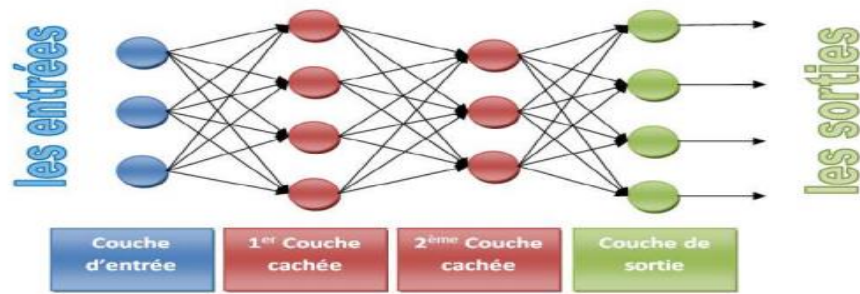


Figure (II.4) : architecture d'un réseau de neurones.

II.2.6 FONCTIONS D'ACTIVATION

Les fonctions d'activations représentent généralement certaines formes de non linéarité. L'une des formes de non linéarité la plus simple, et qui est appropriée aux réseaux discrets, est la fonction signe (Figure II.5.a). Une autre variante de ce type de non linéarité est la fonction de Heaviside (Figure II.5.b). Pour la majorité des algorithmes d'apprentissage il est nécessaire d'utiliser des fonctions sigmoïdes différentiables, telles que la fonction sigmoïde unipolaire, Figure II.5.c et la fonction sigmoïde bipolaire (Figure II.5.d). La classe, la plus utilisée des fonctions d'activation, dans le domaine de la modélisation et de la commande des systèmes non linéaires est la fonction sigmoïde bipolaire [28].

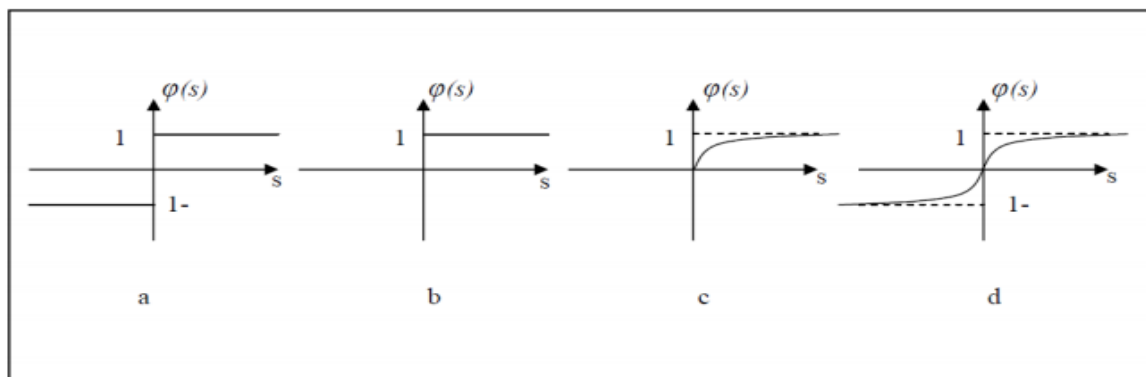


Figure (II.5) : différentes fonctions d'activations [28].

II.2.7 TOPOLOGIE DE RNA

On peut distinguer essentiellement deux types de réseaux de neurones formels : les réseaux non bouclés (feedforward neural networks) et les réseaux bouclés ou récurrents (feedback ou recurrent neural networks (RNN)).

II.2.7.1 RESEAU NON BOUCLE (FEED-FORWARD)

Un réseau de neurones non bouclé est constitué d'une ou plusieurs couches cachées, contenant chacune un certain nombre de neurones. Il réalise une (ou plusieurs) fonction algébrique de ses entrées par composition des fonctions réalisées par chacun de ses neurones. Dans un tel réseau, le flux de l'information circule des entrées vers les sorties sans "retour en arrière". Tout neurone dont la sortie est une sortie du réseau est appelé "neurone de sortie". Les autres, qui effectuent des calculs intermédiaires, sont appelés "neurones cachés" [29].

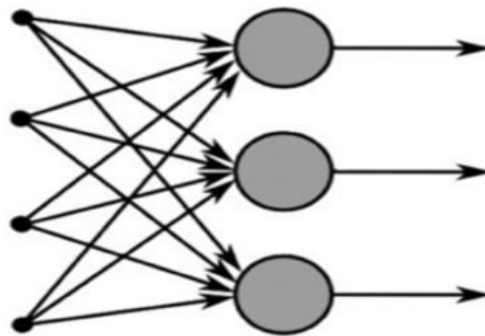


Figure (II.6) : réseau de neurone non bouclé (FEED-FORWARD).

II.2.7.2 RESEAU BOUCLE OU RECURRENT

Les RNA bouclés sont similaires aux réseaux non bouclés, sauf qu'ils n'ont pas de limite d'interconnexion entre les neurones formels, que ce soit les connexions directes ou les connexions de retour de boucle. Les informations peuvent être transmises dans le sens direct et dans le sens inverse. Cette caractéristique engendre des états internes au réseau qui lui donnent un comportement dynamique, ce qui constitue un outil extrêmement utile dans l'identification et la commande des systèmes dynamiques [30].

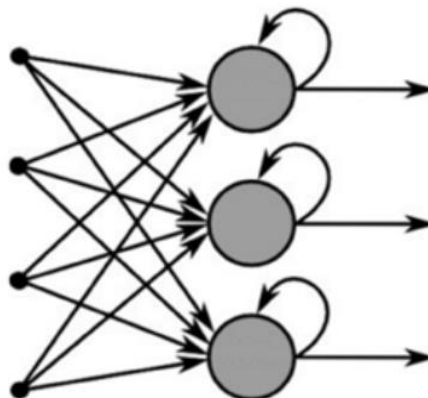


Figure (II.7) : réseau de neurone bouclé (récurrent).

II.2.8 APPRENTISSAGE DES RESEAUX DE NEURONES

Une fois l'architecture choisie, elle doit subir une phase d'apprentissage qui correspond à la phase du développement du réseau durant laquelle il réalise des modifications des poids de connexions du réseau, généralement par des algorithmes spécifiques, afin d'obtenir des valeurs optimales de ces poids. A la fin de cette opération, le réseau converge vers un fonctionnement adapté au problème qu'on désire résoudre, tout en fournissant, au préalable, des exemples d'apprentissage. Ces derniers doivent être suffisamment représentatifs ; autrement dit ; il faudra qu'ils couvrent aussi complètement que possible le domaine de fonctionnement désiré pour le réseau. On peut distinguer trois types d'apprentissage, un apprentissage supervisé, apprentissage non supervisé et apprentissage par renforcement. [31] [32].

II.2.8.1 APPRENTISSAGE SUPERVISE

Pour ce type d'apprentissage, un superviseur (professeur) fournit au réseau les entrées, et au même temps les sorties désirées. Le réseau doit ajuster ses poids de façon à réduire l'écart entre la sortie désirée et sa sortie réelle. Cette procédure est répétée jusqu'à ce qu'un critère de performance soit satisfait. [32].

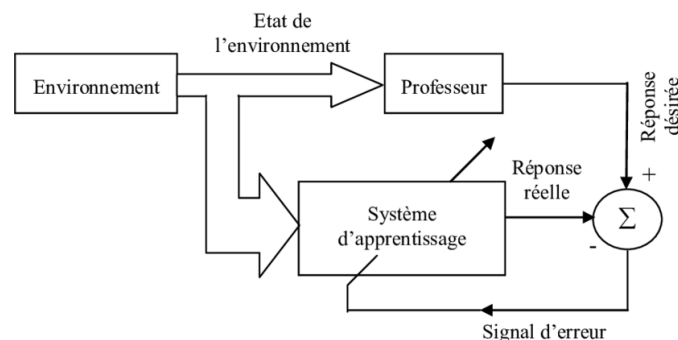


Figure (II.8) : apprentissage supervisé.

II.2.8.2 APPRENTISSAGE NON SUPERVISE

Dans l'apprentissage non supervisé, aucune information sur la sortie désirée n'est fournie au réseau. On présente une entrée au réseau et on le laisse évoluer librement jusqu'à ce qu'il se stabilise, ce comportement est connu sous le nom "auto organisation. Le réseau de neurones dans ce cas-là détecte automatiquement les régularités qui figurent dans les exemples présentés et il modifie les poids des connexions pour que les exemples ayant les mêmes caractéristiques de régularité provoquent la même sortie. Ce mode d'apprentissage est aussi appelé « apprentissage par compétition ». [33]

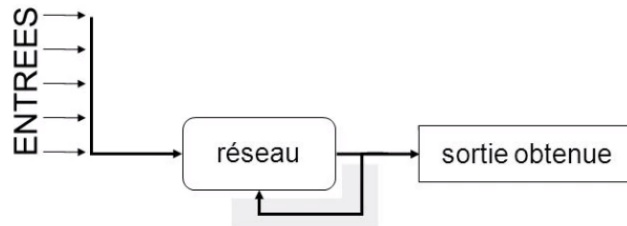


Figure (II.9) : apprentissage non supervisé.

Pour ces deux types d'apprentissage, il y a également un choix traditionnel entre :

- L'apprentissage Offline : l'apprentissage est basé sur un ensemble de données, toutes les données sont dans une base d'exemples d'apprentissage qui sont traités simultanément.
- L'apprentissage On-Line : l'apprentissage n'a pas besoin un ensemble de données, les exemples sont présentés les uns après les autres au fur et à mesure de leur disponibilité alors la mise à jour des poids est essentiellement adaptative. [25][33]

La Figure (II.10) montre les deux systèmes d'apprentissage en ligne ou hors ligne d'un RNA pour l'estimation d'un processus commandé.

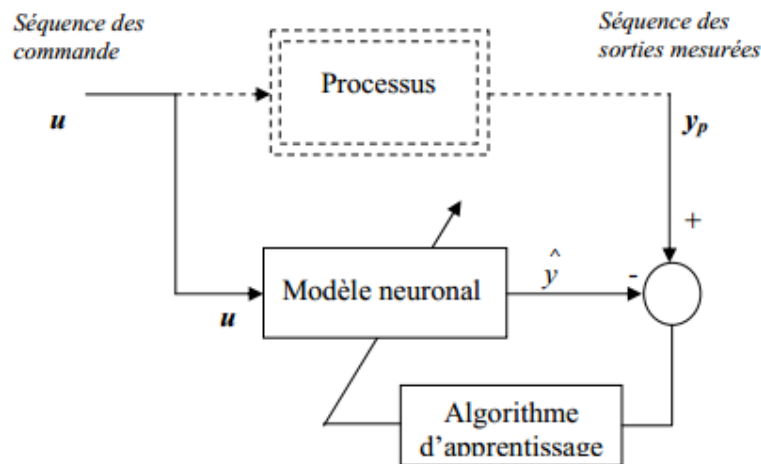


Figure (II.10) : système d'apprentissage en ligne et hors ligne. [34]

II.2.9 ALGORITHME D'APPRENTISSAGE DE RNA

II.2.9.1 ALGORITHME DE RETRO-PROPAGATION DU GRADIENT

La rétro propagation est basée sur des principes mathématiques. Cette méthode a donné de bons résultats dans beaucoup d'applications. Pour l'appliquer il suffit d'avoir des données d'entrées et des données de sortie. Cependant, toute information que possède un réseau de neurone est représentée par les poids d'interconnexions, cette information est acquise durant la phase

d'apprentissage. Cette propriété d'apprendre a permis l'application des tels réseaux dans plusieurs domaines, contrôle, identification, diagnostic,...etc. [35].

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 en corrige plus ou moins fortement les poids par l'intermédiaire d'un coefficient η . [35].

L'algorithme de rétro propagation du gradient de l'erreur se résume aux étapes suivantes : [33]

- 1- Initialisation des poids de connexions à des valeurs aléatoires de faible grandeur.
- 2- Présentation d'un couple (entrée, sortie désirée) de la base d'apprentissage.
- 3- Présentation de la forme d'entrée sur la couche d'entrée du réseau.
- 4- Calcul par propagation de la sortie.
- 5- Calcul des différents signaux d'erreur des différentes couches.
- 6- Mise à jour des matrices de connexions.
- 7- Tant que l'erreur est trop importante, retourner à l'étape 2.

$$E_m = \frac{1}{N} \sum_{k=1}^N e(k)^2 = \frac{1}{N} \sum_{k=1}^N (t(k) - a(k))^2 \quad \text{II.4}$$

E_m : Erreur quadratique commise au niveau de la couche de sortie du réseau

N : Nombre d'exemple dans la base d'apprentissage.

$e(k)$: Erreur quadratique commise à la sortie du réseau.

$t(k)$: sortie désirée.

$a(k)$: sortie de réseau.

Le mécanisme de recherche des poids optimaux du réseau est basé sur la minimisation de l'erreur dans l'espace des poids synaptiques du réseau de neurones. Ce qui conduit à l'équation de mise à jour du $j^{\text{ème}}$ poids et du $i^{\text{ème}}$ neurone suivante :

$$W_{ij}(t + 1) = W_{ij}(t) + \eta \left(\frac{\partial E_m}{\partial W_{ij}(t)} \right) \quad \text{II.5}$$

Avec :

η : Taux d'apprentissage.

$W_{ij}(t + 1)$: Nouveau poids.

$W_{ij}(t)$: Ancien poids.

II.2.9.2 EXEMPLE D'APPLICATION DE RETROPROPAGATION [37]

Voici l'apprentissage d'un réseau de neurones à propagation avec l'algorithme de rétro propagation avec une couche cachée à trois neurones (Figure (II.10)).

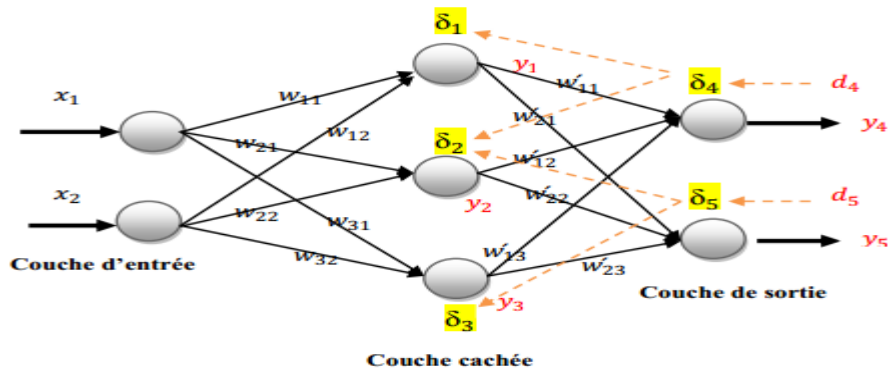


Figure (II.11) : algorithme de retro propagation.

- on calcule y_1, y_2, y_3, y_4 et y_5 :

$$\begin{cases} y_1 = \sigma(x_1 w_{11} + x_2 w_{12} - w_{01}) \\ y_2 = \sigma(x_1 w_{21} + x_2 w_{22} - w_{02}) \\ y_3 = \sigma(x_1 w_{31} + x_2 w_{32} - w_{03}) \\ y_4 = \sigma(y_1 w'_{11} + y_2 w'_{12} + y_3 w'_{13} - w'_{01}) \\ y_5 = \sigma(y_1 w'_{21} + y_2 w'_{22} + y_3 w'_{23} - w'_{02}) \end{cases} \quad \text{II.6}$$

- après avoir calculé les sorties de chaque neurone, on calcule les erreurs de la couche de sortie :

$$\begin{cases} \delta_4 = (d_4 - y_4) \cdot \sigma'(y_1 w'_{11} + y_2 w'_{12} + y_3 w'_{13} - w'_{01}) \\ \delta_5 = (d_5 - y_5) \cdot \sigma'(y_1 w'_{21} + y_2 w'_{22} + y_3 w'_{23} - w'_{02}) \end{cases} \quad \text{II.7}$$

- Et les erreurs de la couche cachée :

$$\begin{cases} \delta_1 = (w'_{11} \cdot \delta_4 + w'_{21} \cdot \delta_5) \cdot \sigma'(x_1 w_{11} + x_2 w_{12} - w_{01}) \\ \delta_2 = (w'_{12} \cdot \delta_4 + w'_{22} \cdot \delta_5) \cdot \sigma'(x_1 w_{21} + x_2 w_{22} - w_{02}) \\ \delta_3 = (w'_{13} \cdot \delta_4 + w'_{23} \cdot \delta_5) \cdot \sigma'(x_1 w_{31} + x_2 w_{32} - w_{03}) \end{cases} \quad \text{II.8}$$

- on calcule la nouvelle valeur de chaque poids entre la couche d'entrée et la couche cachée :

$$\begin{cases} w_{11}(t+1) = w_{11}(t) + \eta(t) \cdot \delta_1(t) \cdot x_1 \\ w_{21}(t+1) = w_{21}(t) + \eta(t) \cdot \delta_1(t) \cdot x_2 \\ w_{12}(t+1) = w_{12}(t) + \eta(t) \cdot \delta_2(t) \cdot x_1 \\ w_{22}(t+1) = w_{22}(t) + \eta(t) \cdot \delta_2(t) \cdot x_2 \\ w_{13}(t+1) = w_{13}(t) + \eta(t) \cdot \delta_3(t) \cdot x_1 \\ w_{23}(t+1) = w_{23}(t) + \eta(t) \cdot \delta_3(t) \cdot x_2 \end{cases} \quad \text{II.9}$$

Et entre la couche cachée et la couche de sortie :

$$\begin{cases} w'_{14}(t+1) = w'_{14}(t) + \eta(t) \cdot \delta_4(t) \cdot y_1 \\ w'_{24}(t+1) = w'_{24}(t) + \eta(t) \cdot \delta_4(t) \cdot y_2 \\ w'_{25}(t+1) = w'_{25}(t) + \eta(t) \cdot \delta_5(t) \cdot y_2 \\ w'_{35}(t+1) = w'_{35}(t) + \eta(t) \cdot \delta_5(t) \cdot y_3 \end{cases} \quad \text{II.10}$$

Il faut recommencer ces opérations pour tous les vecteurs d'apprentissage, puis tester la qualité de l'apprentissage avec les vecteurs de test qui n'ont pas servi à l'apprentissage : ce qui permet de tester les capacités de la généralisation du réseau. On commence tout d'abord l'entraînement par un choix aléatoire des valeurs initiaux du poids. Une fois que la sortie du réseau soit calculée, l'erreur correspondante et le gradient de l'erreur par rapport à tous les poids sont calculés. Les poids sont alors ajustés. On refait la même procédure pour tous les exemples d'apprentissage. Ce processus est répété jusqu'à ce que les sorties du réseau soient suffisamment proches des sorties désirées.

II.2.10 DOMAINES D'APPLICATION DES RNA

II.2.10.1 REGRESSION NON LINEAIRE OU MODELISATION DE DONNEES STATIQUES

Une immense variété de phénomènes statiques peut être caractérisée par une relation déterministe entre des causes et des effets ; les réseaux de neurones sont de bons candidats pour modéliser de telles relations à partir d'observations expérimentales, sous réserve que celles-ci soient suffisamment nombreuses et représentatives [38].

II.2.10.2 MODELISATION DE PRECESSUS DYNAMIQUES NON LINEAIRE

Modéliser un processus consiste à déterminer un ensemble d'équations mathématiques qui décrivent le comportement dynamique du processus, c'est-à-dire l'évolution de ses sorties en fonction de l'évolution de ses entrées [39].

II.2.10.3 COMMANDE DE PRECESSUS

La commande d'un processus consiste à concevoir un système comprenant un organe qui calcule la commande à appliquer au processus pour assurer un comportement dynamique spécifié par des cahiers de charges : régulation au voisinage d'un point de fonctionnement, poursuite d'une trajectoire de consigne, commande optimale.... L'ensemble commande / processus peut donc être considéré comme un système qui réalise une fonction (non linéaire) qu'un réseau de neurone peut approcher [39].

II.2.10.4 CLASSIFICATION

Une autre grande catégorie de problème industriel consiste à attribuer de façon automatique un objet à une classe, parmi d'autres classes possibles. Et en raison de leur propriété d'approximateurs universels, les réseaux de neurones sont capables d'estimer de manière précise la probabilité d'appartenance d'un objet inconnu à une classe parmi plusieurs possibles [40]

II.3 COMMANDE ET IDENTIFICATION DE PRECESSUS PAR RNA

La commande utilisant un contrôleur neuronal se distingue par deux étapes : une étape d'identification et une étape de commande

II.3.1 IDENTIFICATION PAR RNA

L'étape d'identification vise à reproduire le comportement d'un système qu'il soit linéaire ou non linéaire par un réseau de neurones identificateur (RNI).

L'identification est une procédure qui consiste dans l'ajustement des paramètres d'un modèle mathématique afin d'optimiser sa sortie pour qu'elle corresponde à la sortie du système à identifier. Dans RNA, l'identification se fait avec l'adaptation des poids du RNA dans le but d'avoir une sortie qui émule la sortie du système

La construction des lois de commande nécessite l'existence d'un modèle qui représente le mieux possible le comportement dynamique du processus à commander. Pour atteindre ce but, les réseaux de neurones représentent une bonne solution, vu leur capacités d'approximations universelle des fonctions non linéaire. L'identification des systèmes non linéaires par réseaux de neurones a fait l'objet de nombreuse recherche depuis plusieurs années .cette nouvelle approche fournit une description mathématiques très proche de la réalité des systèmes non linéaires. [41]

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

II.3.1.1 IDENTIFICATION DIRECTE

La figure (II.12) 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 neurone $\hat{y}$. puis l'erreur $y - \hat{y}$ est utilisée afin d'affiner les paramètres du système neuronal [25].

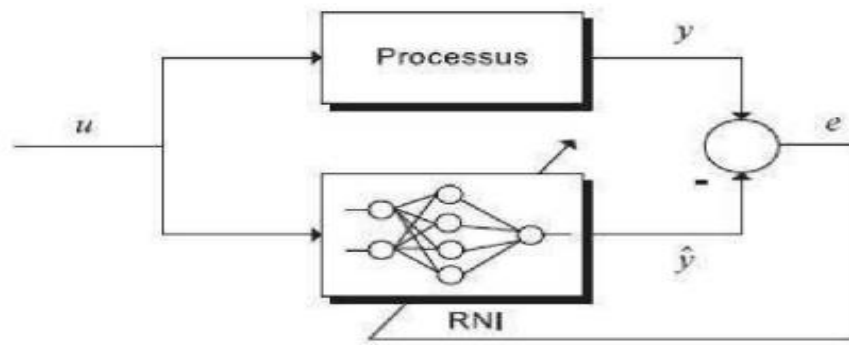


Figure (II.12) : schéma d'identification directe d'un processus par réseau de neurone.

II.3.1.2 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 II.13). Après un apprentissage hors-ligne du modèle inverse, le RNI peut être configuré afin d'assurer un contrôle direct du processus [25].

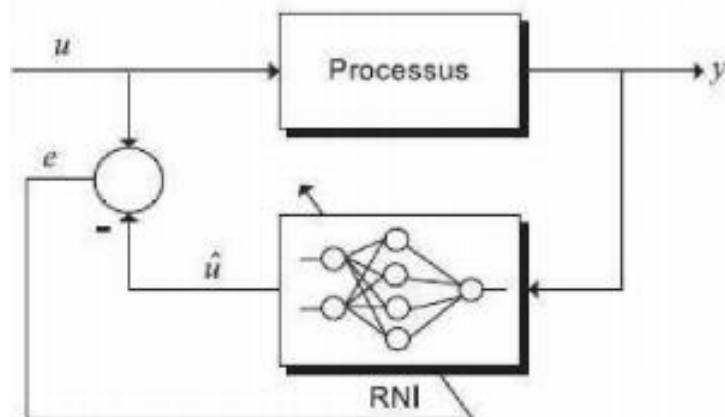


Figure (II.13) : schéma d'identification inverse d'un processus par réseau de neurone.

II.3.2 COMMANDE PAR RNA

L'étape de commande, celle-ci consiste à utiliser les connaissances acquises pendant l'identification pour fournir les signaux de commande dans le but de faire suivre au système commandé une consigne bien définie ou un état désiré de sortie en présence ou non de perturbations. Plusieurs architectures de commande neuronales existent dans la littérature. Dans cette section, nous exposons quelques-unes : la commande directe, la commande inverse-directe, la commande inverse et la commande indirecte. Nous présentons vers la fin d'autres schémas de commandes plus simples basées sur l'apprentissage d'un régulateur conventionnel [42] [43]

II.3.2.1 COMMANDE DIRECTE BASEE SUR L'INVERSE DU SYSTEME

Le schéma de principe de cette méthode est illustré sur la figure (II.14). La consigne de commande est appliquée en entrée au RNC et la sortie de ce RNC constitue la commande du processus. Dans ce cas, la commande peut se faire même sans la boucle de retour (commande en boucle ouverte). Cependant, cette méthode est très peu utilisée car ses performances sans très limitées [44].

Le principe de cette commande repose sur une d'identification par modèle inverse. La figure (II.14) représente le schéma de commande inverse avec un (RNC). Cette architecture reprend le même principe que celui de l'identification inverse montrée dans la figure (II.13). En effet, L'entrée de référence r est comparée à la sortie y du processus pour former l'erreur de poursuite, $e = r - y$ qui sert à modifier les paramètres du réseau en ligne. Après avoir appris le modèle inverse, le neuro-contrôleur délivre la sortie u du RNC qui est la commande injectée en entrée du processus, l'erreur est alors nulle et la sortie y est égale à la référence r . Ce principe est identique au RNI de la figure (II.13) ou lorsque l'apprentissage du modèle inverse est accompli, la sortie du RNI est égale à l'entrée du processus. L'avantage de la commande inverse avec un RNC est le suivi en temps réel de l'évolution du processus, car l'apprentissage est réalisé en ligne. Comme les performances de cet organe de commande dépendent étroitement de la fidélité du modèle inverse, la stabilité et le niveau de performance ne sera pas garanti dans le cas où le modèle inverse n'existe pas ou s'il est difficile à trouver [25].

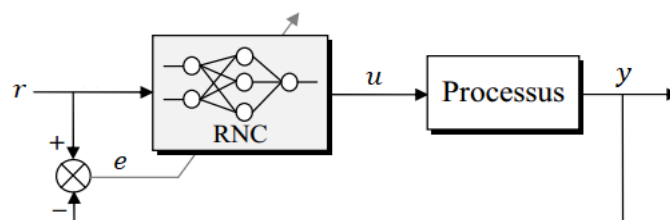


Figure (II.14) : commande basée sur l'inverse de système.

II.3.2.2 COMMANDE INVERSE-DIRECTE

La commande inverse-directe consiste à estimer d'abord l'inverse du processus en utilisant un RNI, et ensuite, ce même réseau sera utilisé en tant que RNC du processus en boucle fermée. Le schéma de principe de cette méthode est montré à la figure (II.15). [44]

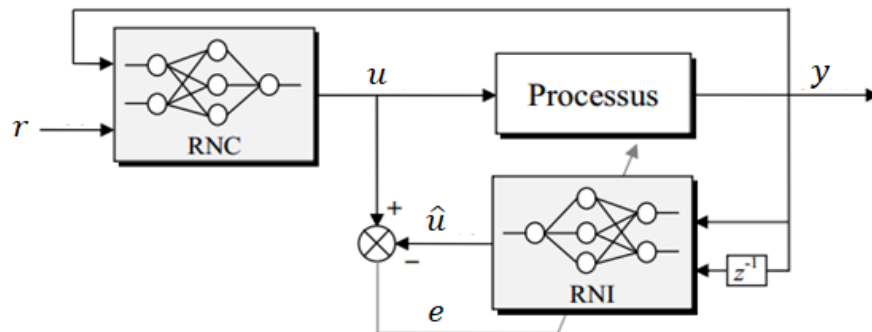


Figure (II.15) : commande inverse-direct par réseau de neurone

II.3.2.3 APPRENTISSAGE D'UN CONTROLLEUR CONVENTIONNEL

Ci-après, d'autres types d'architectures de commande neuronale plus simples sont exposés. Elles sont basées sur l'apprentissage d'un contrôleur conventionnel déjà inséré dans la boucle de commande. La figure (II.16) regroupe quatre architectures de commande [44].

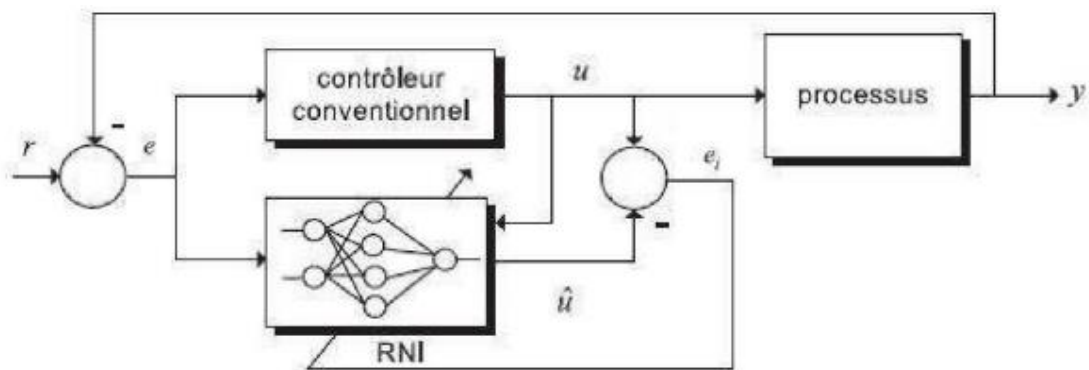


Figure (II.16) : schéma d'apprentissage d'un correcteur conventionnel.

II.4 AVANTAGES ET INCONVENIENTS DES RESEAUX DE NEURONES

II.4.1 AVANTAGES DES RESEAUX DE NEURONES

- L'identification ou la commande neuronale d'un système ne demande pas la connaissance de son modèle mathématique interne [44].
- La commande neuronale s'adapte facilement aux changements du système (commande adaptative) [44].
- moins coûteuses en termes d'implémentation matérielle [44].
- Comportement moins mauvais en cas de faible quantité de données [25].
- Capacité de représenter n'importe quelle fonction, linéaire ou pas, simple ou complexe [25].

II.4.2 INCONVENIENTS DES RESEAUX 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) [25]
- 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 [25]

II.5 APPRENTISSAGE PROFOND OU DEEP LEARNING

L'apprentissage profond fait partie d'une famille de méthodes d'apprentissage automatique fondées sur l'apprentissage de modèles de données. Une observation, une image par exemple, peut être représentée de différentes façons par un vecteur de données, notamment en fonction de :

- L'intensité des pixels dont elle est constituée ;
- Ses différentes arêtes ;
- Ses différentes régions, aux formes particulières.

Certaines représentations et une bonne capacité d'analyse automatique des différenciations rendent la tâche d'apprentissage plus efficace [45].

Les techniques d'apprentissage profond constituent une classe d'algorithmes d'apprentissage automatique qui :

- Utilisent différentes couches d'unité de traitement non linéaire pour l'extraction et la transformation des caractéristiques ; chaque couche prend en entrée la sortie de la

précédente ; les algorithmes peuvent être supervisés ou non supervisés, et leurs applications comprennent la reconnaissance de modèles et la classification statistique ;

- Fonctionnent avec un apprentissage à plusieurs niveaux de détail ou de représentation des données ; à travers les différentes couches, on passe de paramètres de bas niveau à des paramètres de plus haut niveau, où les différents niveaux correspondent à différents niveaux d'abstraction des données.

L'apprentissage profond utilise des couches cachées de réseaux de neurones artificiels, des « machines de Boltzmann restreintes », et des séries de calculs propositionnels complexes. Les algorithmes d'apprentissage profond s'opposent aux algorithmes d'apprentissage peu profonds du fait du nombre de transformations réalisées sur les données entre la couche d'entrée et la couche de sortie, où une transformation correspond à une unité de traitement définie par des poids et des seuils [45].

Le concept d'apprentissage profond prend forme dans les années 2010, avec la convergence de quatre facteurs :

- **Des réseaux de neurones artificiels multicouches** (eux-mêmes issus entre autres du concept de perceptron, datant de la fin des années 1950) ;
- Des algorithmes d'analyse discriminante et apprenants (dont l'émergence remonte aux années 1980) ;
- Des machines dont la puissance de traitement permet de traiter des données massives ;
- Des bases de données suffisamment grandes, capables d'entraîner des systèmes de grandes tailles [45].

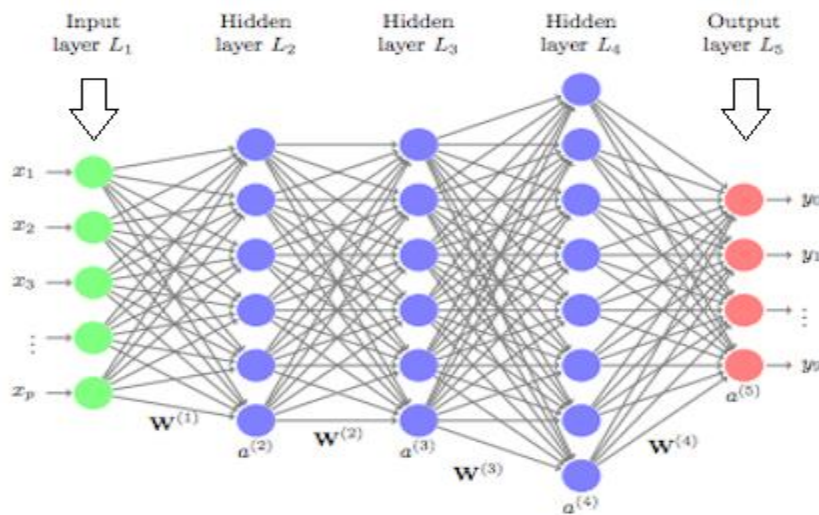


Figure (II.17) : Illustration d'un réseau de neurones artificiels multicouches utilisé comme technique d'apprentissage profond (Université de Cincinnati, USA, [46])

Dans ce contexte, on considère d'étudier l'application des réseaux de neurones artificiels multicouches pour contrôler, en position et angle, efficacement et avec robustes le pendule inversé entraîné par un machine à courant continu.

II.6 CONCLUSION

Dans ce chapitre nous avons présenté les différents concepts et aspects liés aux réseaux de neurones artificiels, et aussi les différents types d'architectures et de modèle qui existent.

Nous avons aussi présenté le concept de l'apprentissage profond (deep learning), tout en définissant sa relation avec les réseaux de neurones artificiels, où ceux à plusieurs couches cachées (multicouches) sont souvent adoptés par rapport à d'autres techniques utilisées dans le domaine du deep learning. Le chapitre suivant dédié à l'application des réseaux de neurones artificiels à une couche et à multicouches pour contrôler, en position et en angle, le pendule inversé entraîné par un machine à courant continue.

CHAPITRE III
RESEAX DE
NEURONE POUR
LA
STABILISATION
DE PENDULE
INVERSE

III.1 INTRODUCTION

La stabilité est une exigence critique dans la conception d'un asservissement. Une perte de stabilité entraîne au mieux un comportement oscillatoire et donc une incapacité à régler ou poursuivre, au pire la génération de signaux de grande énergie qui vont endommager le système [47].

Ce chapitre décrit les modèles de commande utilisant les RNA pour la stabilisation d'un pendule inversé dans sa position verticale instable, en utilisant les résultats obtenus par la commande par FEEDBACK linéarisation. Plusieurs architectures sont étudiées et comparées. La commande par RNA est choisie comme étant une stratégie de commande appropriée et l'implémentation de l'algorithme est décrite. La performance de la commande par RNA est évaluée et elle est comparée à la commande par FEEDBACK linéarisation.

III.2 STABILISATION DE PENDULE INVERSE PAR FEEDBACK LINEARISATION

Un contrôleur de type FEEDBACK linéarisation a été développé pour contrôler le système non linéaire de pendule inversé. Les résultats (entrée (signal de commande) et sorties (position et vitesse du chariot, position et vitesse angulaire du pendule)) sont stockés sous forme de base de donnée. [48]

III.2.1 MISE EN EQUATION DU CONTROLE PAR FEEDBACK LINEARISATION

Les équations suivantes montrent le développement mathématique du calcul de la loi de contrôle développée pour le contrôleur de pendule inversé. Les quatre premières équations (équation III.1, III.2, III.3 et III.4) sont introduites dans l'équation principale (équation III.5). L'équation principale calcule la force (U) requise pour garder le pendule stable [48].

$$h_1 = \frac{3}{4l} g \sin \theta \quad \text{III.1}$$

$$h_2 = \frac{3}{4l} g \cos \theta \quad \text{III.2}$$

$$f_1 = m \left(l \sin \theta \dot{\theta}^2 - \frac{3}{8} g \sin 2\theta \right) - f_x \quad \text{III.3}$$

$$f_2 = M + m \left(1 - \frac{3}{4} \cos^2 \theta \right) \quad \text{III.4}$$

$$u = \frac{f_2}{h_2} [h_1 + k_1(\theta - \theta_d) + k_2\theta + c_1(x - x_d) + c_2\dot{x}] - f_1 \quad \text{III.5}$$

$$u = \frac{M + m \left(1 - \frac{3}{4} \cos^2 \theta \right)}{\frac{3}{4l} g \cos \theta} \left[\frac{3}{4l} g \sin \theta + k_1(\theta - \theta_d) + k_2\theta + c_1(x - x_d) + c_2\dot{x} \right] - m \left(l \sin \theta \dot{\theta}^2 - \frac{3}{8} g \sin 2\theta \right) - f_x \quad \text{III.6}$$

III.2.2 ÉLABORATION DU CONTROLEUR AVEC SIMULINK

Le schéma Simulink ci-dessous montre le modèle Simulink du pendule inverse commandé par un contrôleur feedback linéarisant.

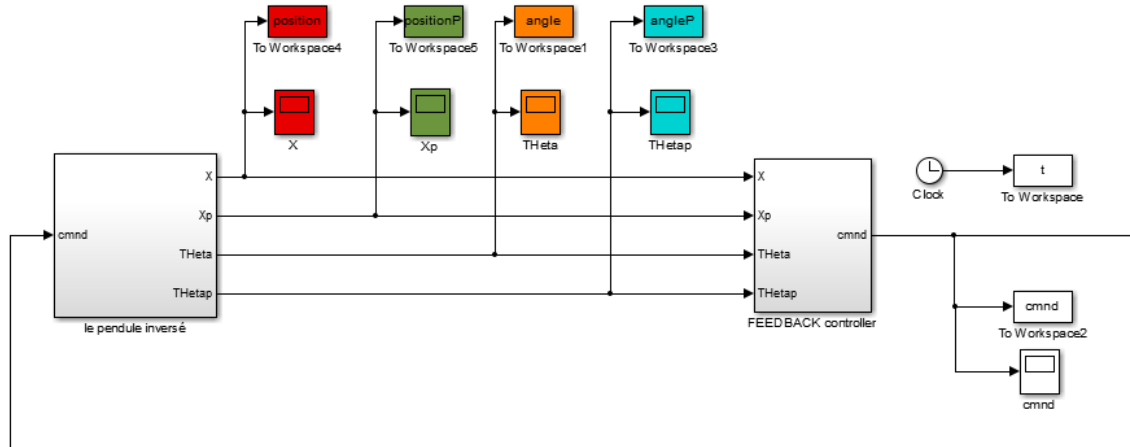


Figure (III.1) : Modèle Simulink du pendule inversé commandé par un contrôleur feedback linéarisant.

La figure III.2 montre les résultats de simulation du pendule inversé commandé par un contrôleur feedback linéarisant avec position initiale linéaire du chariot de 0 mètre et position initiale angulaire du pendule de 0.4 radian, pour une position angulaire de référence du pendule de 0 radian. Ces résultats montrent que le pendule inversé se stabilise au bout de 7 seconds, avec une erreur de -0.5×10^{-5} radian de l'angle du pendule par rapport à sa référence.

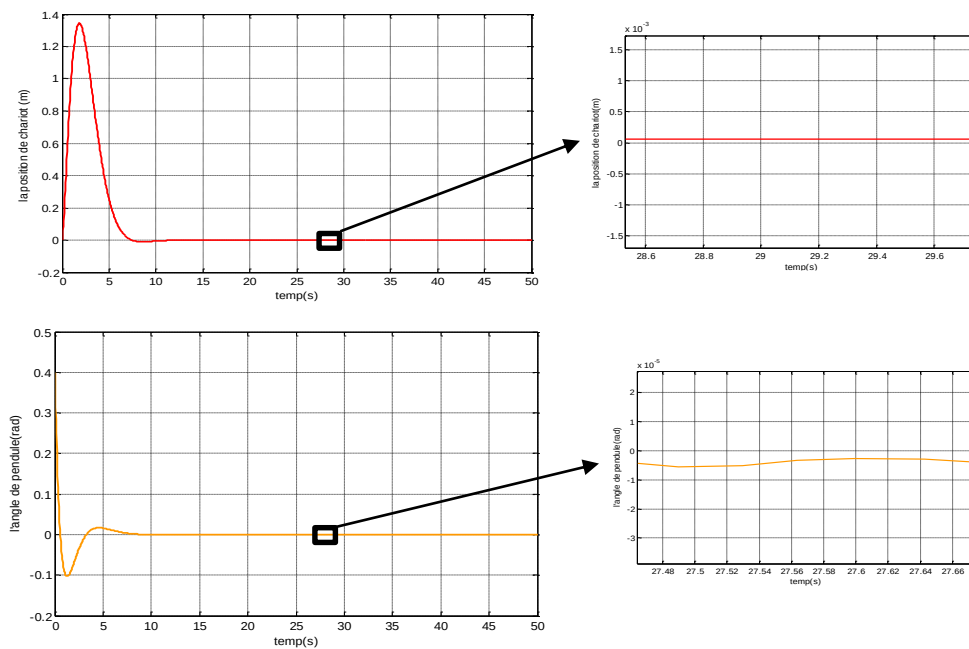


Figure (III.2) : Résultats de simulation du pendule inversé commandé par un contrôleur feedback linéarisant avec position initiale linéaire du chariot de 0 mètre et position initiale angulaire du pendule de 0.4 radian.

III.3 STABILISATION DE PENDULE INVERSE PAR UNE CONTROLLEUR A BASE DE RESEAU DE NEURONS ARTIFICIELS

Dans notre étude, nous avons remplacé le régulateur FEEDBACK par un régulateur neuronal. Pour former le réseau neuronal à imiter le contrôleur existant, les vecteurs d'entrées et des cibles de contrôle du contrôleur doivent être collectées du schéma de la commande du pendule inversé (figure III.1).

La détermination et le choix du réseau optimal pour un processus donné sont des problèmes ouverts, malgré l'existence de quelques travaux, qui permettent pour une vaste classe de réseaux, de déterminer l'architecture optimale.

Avant de concevoir un réseau de neurones artificiels, il faut tout d'abord [27] :

- **Fixer le nombre de couches cachées** : Mis à part les couches d'entrée et de sortie, l'analyste doit décider du nombre de couches intermédiaires ou cachées. Sans couche cachée, le réseau n'offre que de faibles possibilités d'adaptation ; avec une couche cachée, il est capable, avec un nombre suffisant de neurones, d'approximer toute fonction continue.
- **déterminer le nombre de neurones par couches cachées** Actuellement, il n'existe pas une loi qui nous dicte exactement le nombre de neurones nécessaires au niveau des couches cachées. Donc on ne sait pas comment construire le réseau, ni combien de neurones sont dans la couche cachée, ni combien de liens synaptiques. En effet, si le réseau possède un très grand nombre de poids et de neurones, le réseau est trop souple et si ce nombre est trop petit, le réseau est trop rigide et présente des mauvaises performances.
- **Test d'arrêt** : La détermination du critère d'arrêt est cruciale dans la mesure où la convergence peut passer par des minima locaux. En effet, le test d'arrêt est la mesure des performances du réseau pour savoir si la convergence du réseau est atteinte. D'une façon générale, on cherche à arrêter l'algorithme si l'erreur E est minimale c'est-à-dire si le gradient de l'erreur est proche de zéro.
- **Taux d'apprentissage** : Ce paramètre détermine la vitesse de convergence. Si la valeur de démarrage de η est grande, alors on aura un apprentissage très rapide mais au prix de la création des oscillations dans l'erreur totale qui empêcheront l'algorithme de converger vers un minimum désiré. Le réseau devient instable. Dans la plupart des cas si la fonction d'erreur possède plusieurs minimums locaux, le réseau subira un blocage dans l'un d'eux. Toutes ces conditions nous obligent à commencer l'apprentissage avec une petite valeur de η , si on veut attendre un minimum global même si l'apprentissage est long.
- **Seuil de tolérance** : Ce paramètre critique détermine la précision de la réponse du réseau. Théoriquement, l'algorithme doit se terminer dès que le minimum de l'erreur commise par le réseau sera atteint, correspondant à un gradient nul, ce qui n'est jamais rencontré en pratique. C'est pourquoi on fixe à priori ce seuil afin d'arrêter l'apprentissage.

III.3.1 APPRENTISSAGE SUPERVISE HORS LIGNE

Pour former le réseau neuronal à imiter le contrôleur existant, un vecteur d'entrées et les cibles de contrôle du contrôleur doivent être collectées du schéma de la commande du pendule inversé (figure III.1).

On a créé un tableau de cellules pour combiner les 4 entrées différentes en 1 vecteur d'entrée. On a utilisé la fonction "newff" qui permet de spécifier notre réseau (le nombre de couches, le nombre de neurones dans les couches cachées et les fonctions d'activation utilisées).

Ce réseau est développé à l'aide de nntool toolboxes, ces toolboxes permet pour les poids à ajuster dans l'environnement 'matlab' mais lorsqu'un réseau de neurone est converti vers Simulink, les poids sont définis et ne peuvent pas être ajustés alors l'apprentissage en ligne n'est pas possible

Concernant l'apprentissage de ce réseau, on applique algorithme d'apprentissage de Levenberg-Marquardt

Pour le choix on a comparé neuf architectures des réseaux de neurones pour le but de choisir la plus efficace.

Lorsque l'apprentissage est terminé et les poids sont définis, un bloc de réseau de neurones sur Simulink est généré et placé à la place du contrôleur existant

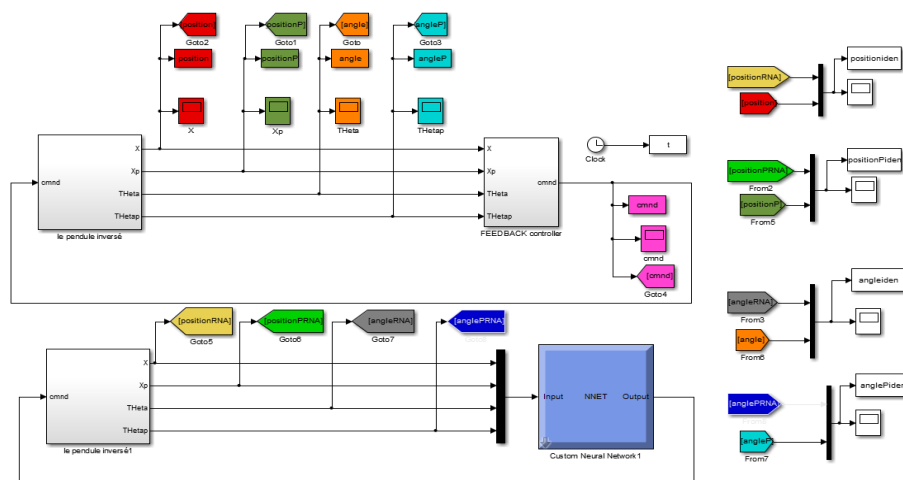


Figure (III.3) : Modèle Simulink du pendule inversé commandé par un contrôleur feedback linéarisant et celui commandé par réseau de neurone.

III.3.2 RESULTATS DE SIMULATION

- RNA de 10 neurones dans la couche cachée

Dans cette architecture, on a utilisé un réseau de neurone qui se compose d'une seule couche cachée de 10 neurones, de 4 entrées et une seul sortie. Les entrées sont la position et l'angle du pendule inversé et leurs dérivées, et la sortie est le signal de commande du pendule (la force (U) requise pour garder le pendule stable).

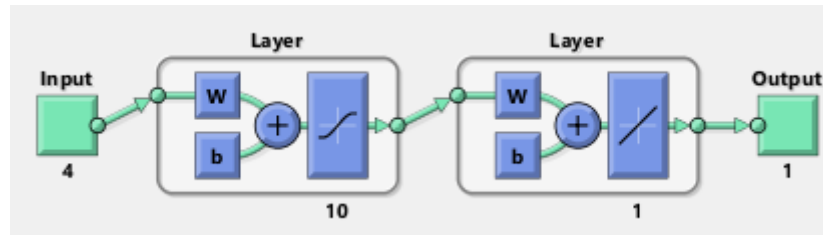


Figure (III.4) : Architecture du RNA d'une seule couche cachée de 10 neurones utilisé dans l'outil d'apprentissage du RNA « nntaintool ».

Dans la figure (III.5), on montre les performances d'apprentissage de cette architecture, décrites par l'évolution de la valeur de l'erreur quadratique moyenne de la sortie par rapport à sa référence. On remarque qu'après 1000 itérations, l'erreur quadratique moyenne atteint une valeur de 2.1588×10^{-9} .

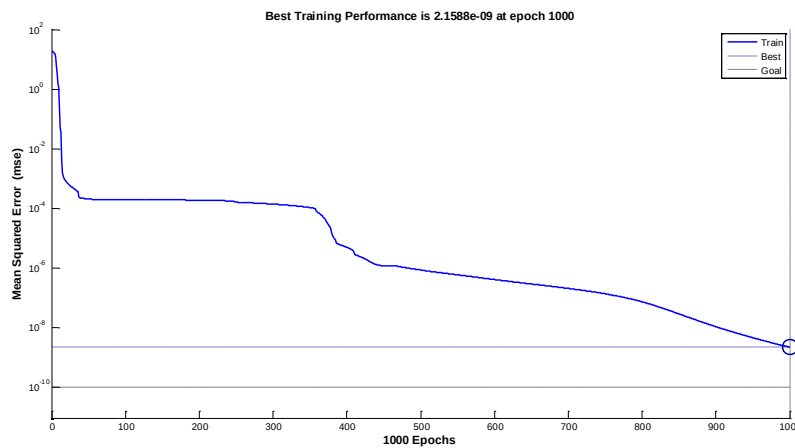


Figure (III.5) : Courbe de performance (évolution de la valeur de l'erreur quadratique moyenne) du réseau de neurone d'une seule couche cachée de 10 neurones

Dans la figure (III.6), les sorties du système contrôlé par FEEDBACK linéarisation et les sorties du même système contrôlé par RNA sont présentées.

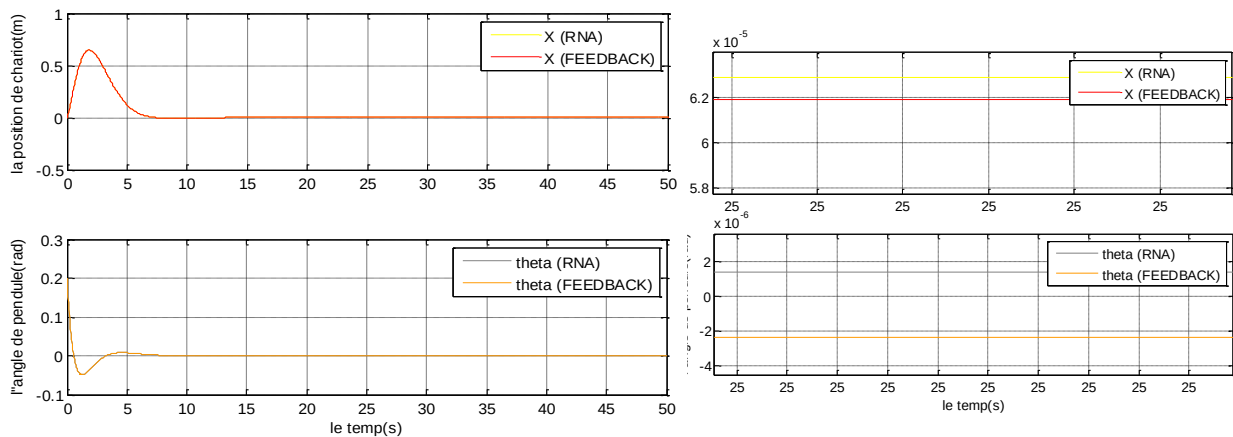


Figure (III.6) : Position et l'angle du pendule inversé commandé par FEEDBACK et par réseau de neurone d'une seule couche cachée de 10 neurones.

- RNA de 7 neurones dans la 1^{ère} couche cachée et 3 neurones dans la 2^{ème} couche cachée

Dans cette architecture, on a utilisé un réseau de neurone qui se compose de deux couches cachées, 7 neurones dans la 1^{ère} couche cachée et 3 neurones dans la 2^{ème} couche cachée avec 4 entrées et une seule sortie.

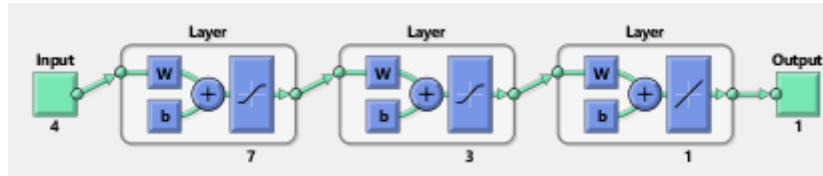


Figure (III.7) : Architecture de réseau de 2 couches cachées, 7 neurones dans la 1^{ère} couche cachée et 3 neurones dans la 2^{ème} couche cachée.

La figure (III.8) présente les performances d'apprentissage de cette architecture, sous forme de l'évolution de la valeur de l'erreur quadratique moyenne de la sortie par rapport à sa référence. On remarque qu'après 1000 itérations, l'erreur quadratique moyenne atteint une valeur de 2.5732×10^{-7} .

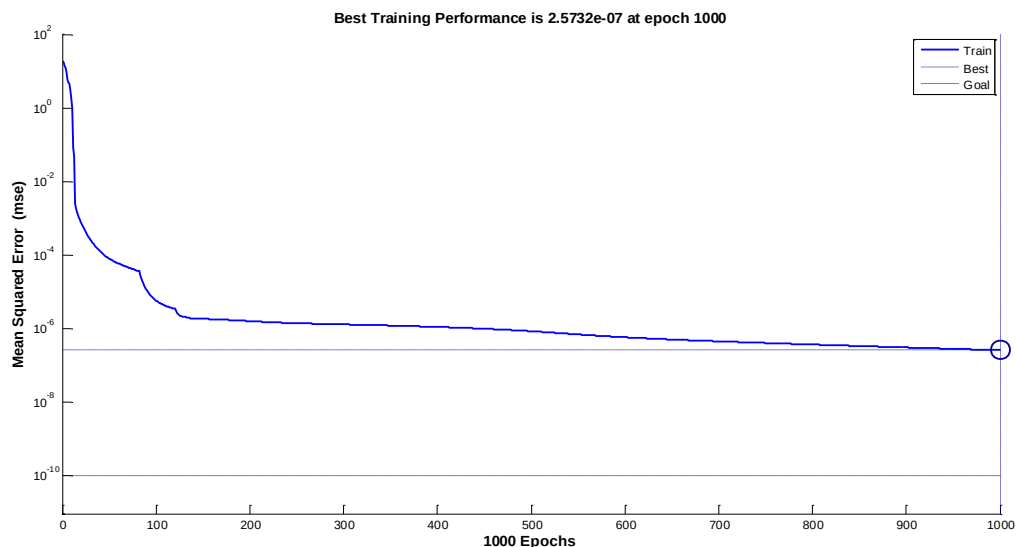


Figure (III.8) : Courbe de performance (évolution de la valeur de l'erreur quadratique moyenne) du réseau de neurone de 2 couches cachées, 7 neurones dans la 1^{ère} couche cachée et 3 neurones dans la 2^{ème} couche cachée.

Dans la figure (III.9), les sorties du système contrôlé par FEEDBACK linéarisation et les sorties du même système contrôlé par RNA sont présentées.

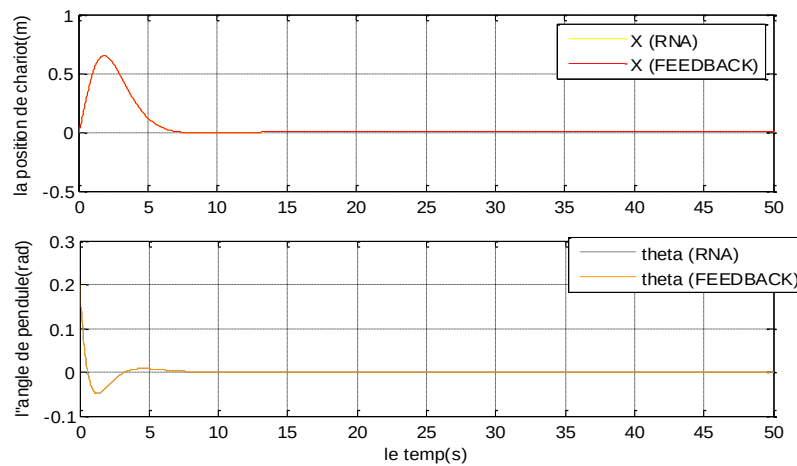


Figure (III.9) : Position et l'angle du pendule inversé commandé par FEEDBACK et par réseau de neurone de 2 couches cachées, 7 neurones dans la 1^{ère} couche cachée et 3 neurones dans la 2^{ème} couche cachée.

- **RNA de 6 neurones dans la 1^{ère} couche cachée et 4 neurones dans la 2^{ème} couche cachée**

Dans cette architecture, on a considéré un réseau de neurone qui se compose de deux couches cachées, 6 neurones dans la 1^{ère} couche cachée et 4 neurones dans la 2^{ème} couche cachée avec 4 entrées et une seule sortie.

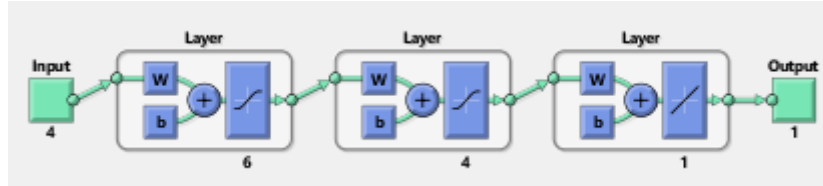


Figure (III.10) : Architecture de réseau de 2 couches cachées, 6 neurones dans la 1^{ère} couche cachée et 4 neurones dans la 2^{ème} couche cachée.

La figure (III.11) présente les performances d'apprentissage de cette architecture, sous forme de l'évolution de la valeur de l'erreur quadratique moyenne de la sortie par rapport à sa référence. On remarque qu'après 1000 itérations, l'erreur quadratique moyenne atteint une valeur de 8.3032×10^{-8} .

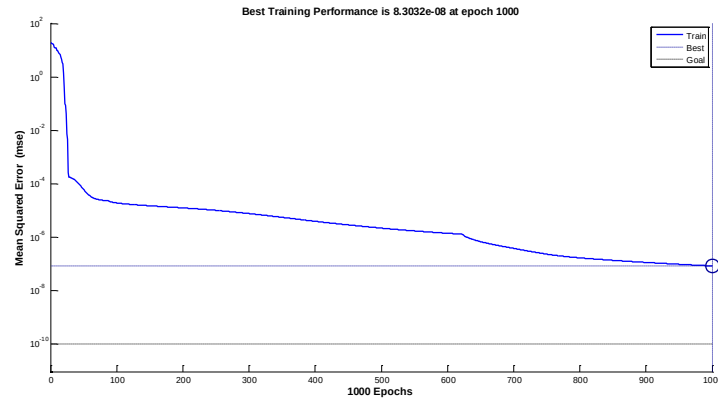


Figure (III.11) : Courbe de performance (évolution de la valeur de l'erreur quadratique moyenne) du réseau de neurone de 2 couches cachées, 6 neurones dans la 1^{ère} couche cachée et 4 neurones dans la 2^{ème} couche cachée.

Dans la figure (III.12), les sorties du système contrôlé par FEEDBACK linéarisation et les sorties du même système contrôlé par RNA sont présentées.

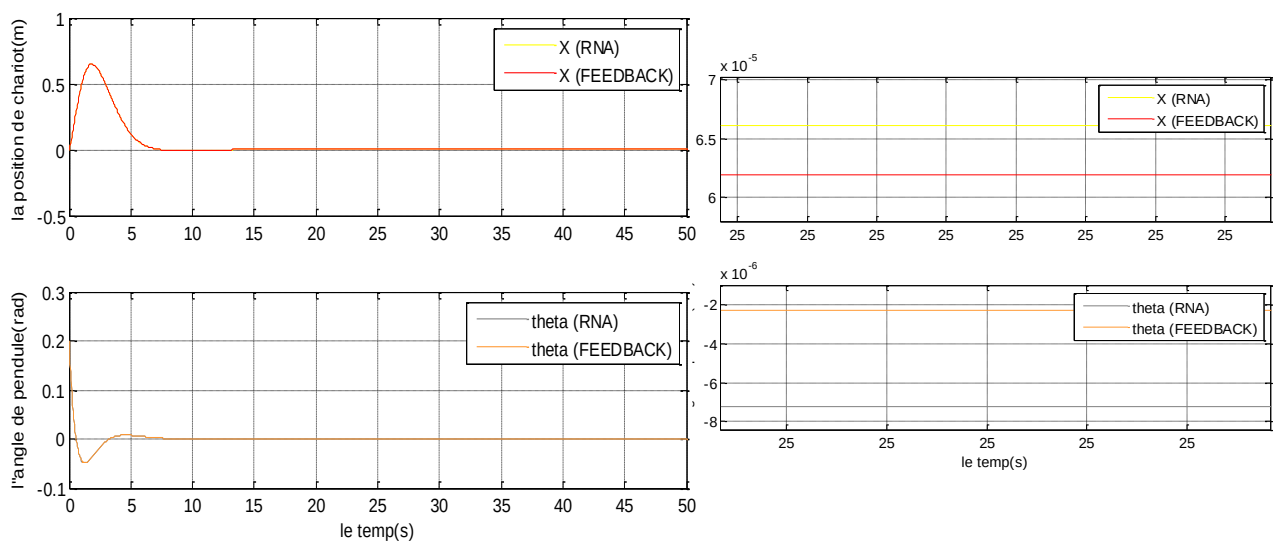


Figure (III.12) : Position et l'angle du pendule inversé commandé par FEEDBACK et par réseau de neurone de 2 couches cachées, 6 neurones dans la 1^{ère} couche cachée et 4 neurones dans la 2^{ème} couche cachée.

- RNA de 5 neurones dans la 1^{ère} couche cachée et 5 neurones dans la 2^{ème} couche cachée

Dans cette architecture, on a utilisé un réseau de neurone qui se compose de deux couches cachées, 5 neurones dans la 1^{ère} couche cachée et 5 neurones dans la 2^{ème} couche cachée avec 4 entrées et une seule sortie.

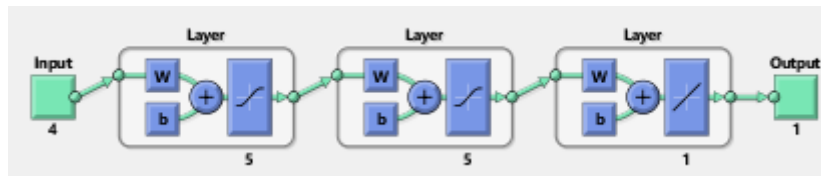


Figure (III.13) : Architecture de réseau de 2 couches cachées, 5 neurones dans la 1^{ère} couche cachée et 5 neurones dans la 2^{ème} couche cachée.

La figure (III.14) présente les performances d'apprentissage de cette architecture, sous forme de l'évolution de la valeur de l'erreur quadratique moyenne de la sortie par rapport à sa référence. On remarque qu'après 1000 itérations, l'erreur quadratique moyenne atteint une valeur de 1.9506×10^{-8} .

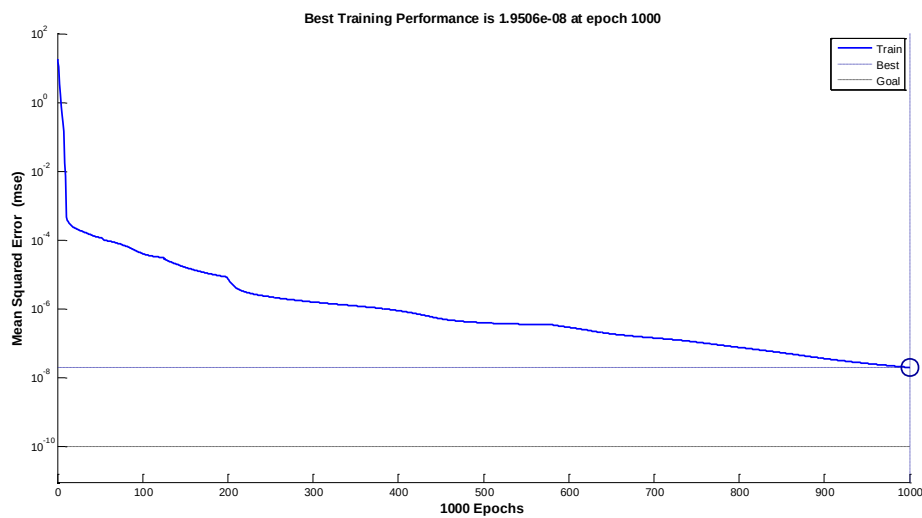


Figure (III.14) : Courbe de performance (évolution de la valeur de l'erreur quadratique moyenne) du réseau de neurone de 2 couches cachées, 5 neurones dans la 1^{ère} couche cachée et 5 neurones dans la 2^{ème} couche cachée.

Dans la figure (III.15), les sorties du système contrôlé par FEEDBACK linéarisation et les sorties du même système contrôlé par RNA sont présentées.

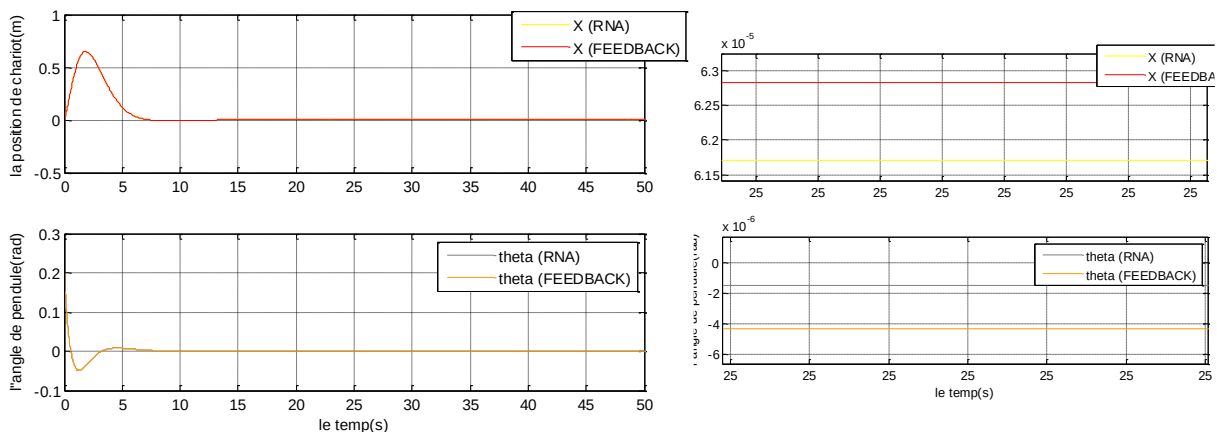


Figure (III.15) : Position et l'angle du pendule inversé commandé par FEEDBACK et par réseau de neurone de 2 couches cachées, 5 neurones dans la 1^{ère} couche cachée et 5 neurones dans la 2^{ème} couche cachée.

- RNA de 4 neurones dans la 1^{ère} couche cachée et 6 neurones dans la 2^{ème} couche cachée

Dans cette architecture, on a considéré un réseau de neurone composé de deux couches cachées, 4 neurones dans la 1^{ère} couche cachée et 6 neurones dans la 2^{ème} couche cachée avec 4 entrées et une seule sortie.

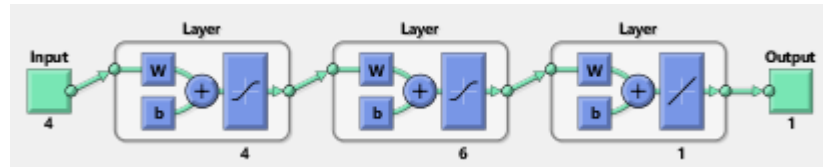


Figure (III.16) : Architecture de réseau de 2 couches cachées, 4 neurones dans la 1^{ère} couche cachée et 6 neurones dans la 2^{ème} couche cachée.

La figure (III.17) présente les performances d'apprentissage de cette architecture, sous forme de l'évolution de la valeur de l'erreur quadratique moyenne de la sortie par rapport à sa référence. On remarque qu'après 1000 itérations, l'erreur quadratique moyenne atteint une valeur de 1.0524×10^{-8} .

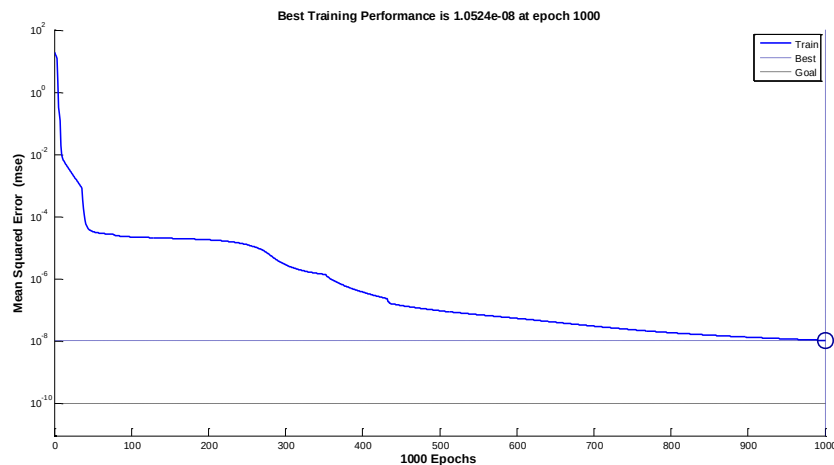


Figure (III.17) : Courbe de performance (évolution de la valeur de l'erreur quadratique moyenne) du réseau de neurone de 2 couches cachées, 4 neurones dans la 1^{ère} couche cachée et 6 neurones dans la 2^{ème} couche cachée.

Dans la figure (III.18), les sorties du système contrôlé par FEEDBACK linéarisation et les sorties du même système contrôlé par RNA sont présentées.

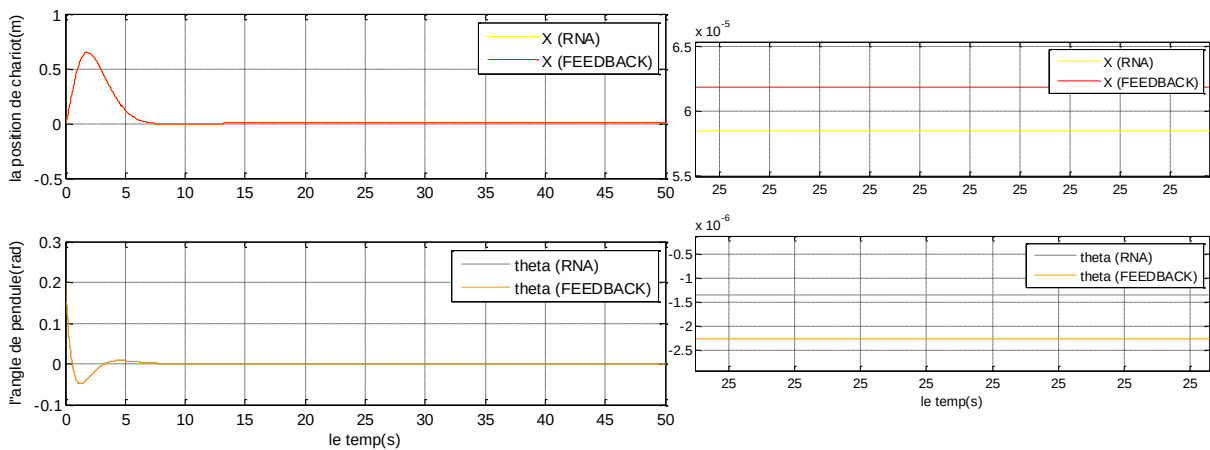


Figure (III.18) : Position et l'angle du pendule inversé commandé par FEEDBACK et par réseau de neurones de 2 couches cachées, 4 neurones dans la 1^{ère} couche cachée et 6 neurones dans la 2^{ème} couche cachée.

- RNA de 3 neurones dans la 1^{ère} couche cachée et 7 neurones dans la 2^{ème} couche cachée

Dans cette architecture, on a considéré un réseau de neurone qui se compose de deux couches cachées, 3 neurones dans la 1^{ère} couche cachée et 7 neurones dans la 2^{ème} couche cachée avec 4 entrées et une seule sortie.

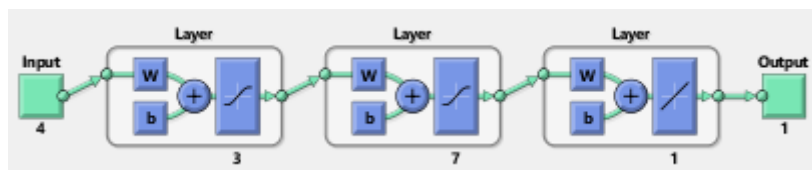


Figure (III.19) : Architecture de réseau de 2 couches cachées, 3 neurones dans la 1^{ère} couche cachée et 7 neurones dans la 2^{ème} couche cachée.

La figure (III.20) présente les performances d'apprentissage de cette architecture, sous forme de l'évolution de la valeur de l'erreur quadratique moyenne de la sortie par rapport à sa référence. On remarque qu'après 1000 itérations, l'erreur quadratique moyenne atteint une valeur de 2.5954×10^{-8} .

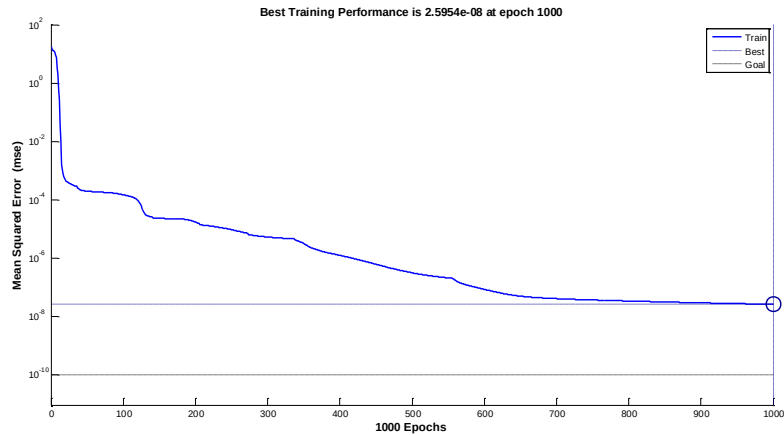


Figure (III.20) : Courbe de performance (évolution de la valeur de l'erreur quadratique moyenne) du réseau de neurones de 2 couches cachées, 3 neurones dans la 1^{ère} couche cachée et 7 neurones dans la 2^{ème} couche cachée.

Dans la figure (III.21), les sorties du système contrôlé par FEEDBACK linéarisation et les sorties du même système contrôlé par RNA sont présentées.

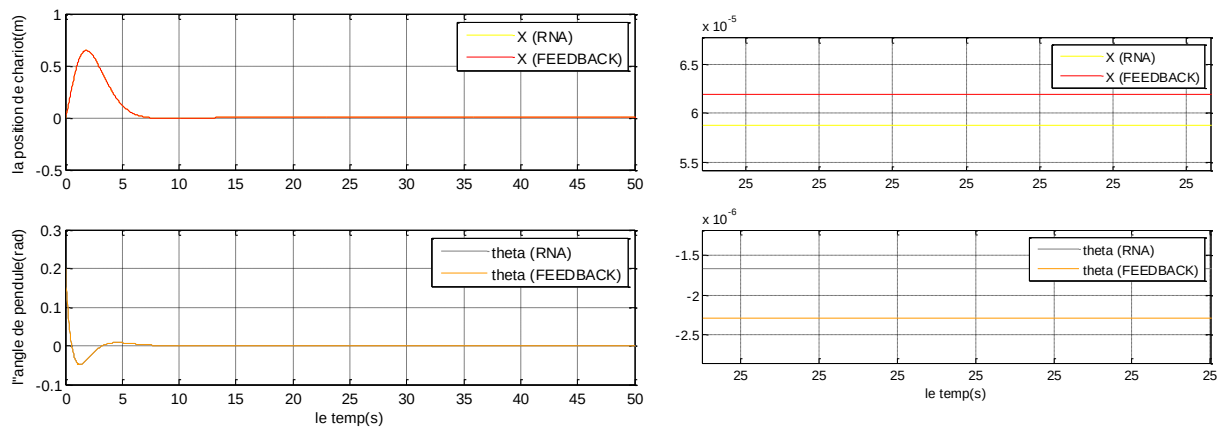


Figure (III.21) : Position et l'angle du pendule inversé commandé par FEEDBACK et par réseau de neurones de 2 couches cachées, 3 neurones dans la 1^{ère} couche cachée et 7 neurones dans la 2^{ème} couche cachée.

- RNA de 7 neurones dans la 1^{ère} couche cachée et 2 neurones dans la 2^{ème} couche cachée et 1 neurone dans la 3^{ème} couche cachée

Dans cette architecture, on a adopté un réseau de neurone qui se compose de 3 couches cachées, 7 neurones dans la 1^{ère} couche cachée et 2 neurones dans la 2^{ème} couche cachée et 1 neurone dans la 3^{ème} couche cachée, avec 4 entrées et une seule sortie.



Figure (III.22) : Architecture de réseau de 3 couches cachées, 7 neurones dans la 1^{ère} couche cachée et 2 neurones dans la 2^{ème} couche cachée et 1 neurone dans la 3^{ème} couche cachée.

La figure (III.23) présente les performances d'apprentissage de cette architecture, sous forme de l'évolution de la valeur de l'erreur quadratique moyenne de la sortie par rapport à sa référence. On remarque qu'après 1000 itérations, l'erreur quadratique moyenne atteint une valeur de 6.0703×10^{-7} .

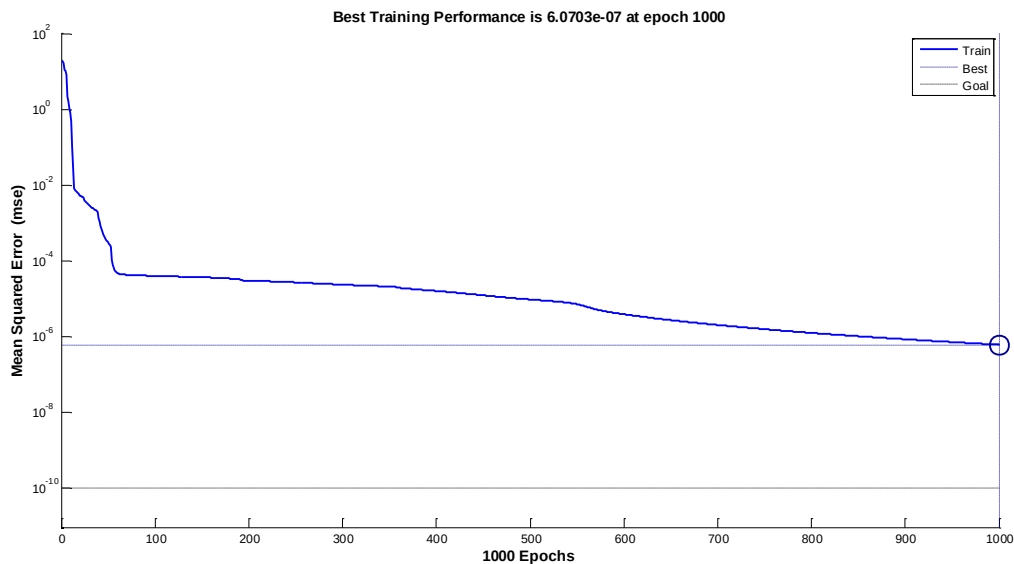


Figure (III.23) : Courbe de performance (évolution de la valeur de l'erreur quadratique moyenne) du réseau de neurones de 3 couches cachées, 7 neurones dans la 1^{ère} couche cachée et 2 neurones dans la 2^{ème} couche cachée et 1 neurone dans la 3^{ème} couche cachée.

Dans la figure (III.24), les sorties du système contrôlé par FEEDBACK linéarisation et les sorties du même système contrôlé par RNA sont présentées.

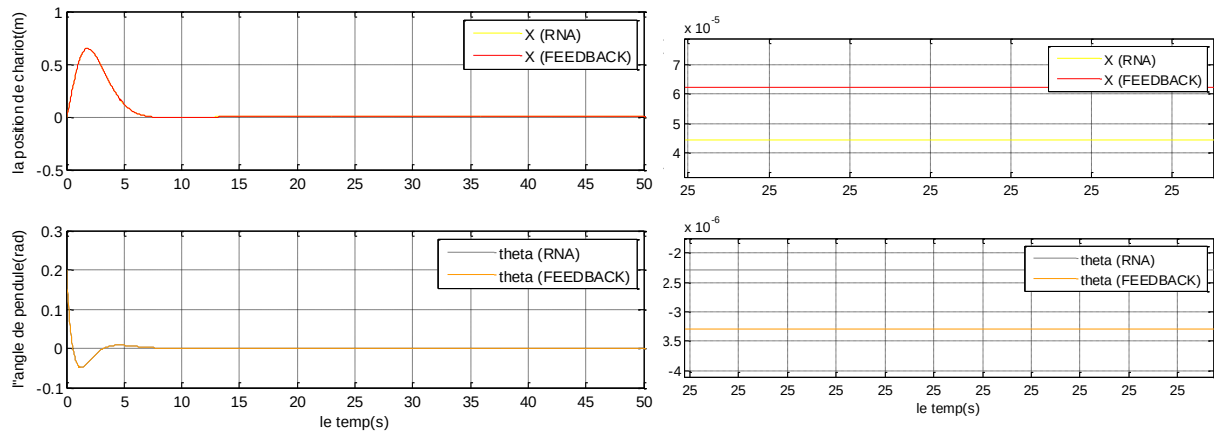


Figure (III.24) : Position et l'angle du pendule inversé commandé par FEEDBACK et par réseau de neurones de 3 couches cachées, 7 neurones dans la 1^{ère} couche cachée et 2 neurones dans la 2^{ème} couche cachée et 1 neurone dans la 3^{ème} couche cachée.

- **RNA de 1 neurone dans la 1^{ère} couche cachée et 2 neurones dans la 2^{ème} couche cachée et 7 neurones dans la 3^{ème} couche cachée**

Dans cette architecture, on a choisi un réseau de neurone qui se compose de 3 couches cachées, 1 neurone dans la 1^{ère} couche cachée et 2 neurones dans la 2^{ème} couche cachée et 7 neurones dans la 3^{ème} couche cachée, avec 4 entrées et une seule sortie.



Figure (III.25) : Architecture de réseau de 3 couches cachées, 1 neurone dans la 1^{ère} couche cachée et 2 neurones dans la 2^{ème} couche cachée et 7 neurones dans la 3^{ème} couche cachée.

La figure (III.26) présente les performances d'apprentissage de cette architecture, sous forme de l'évolution de la valeur de l'erreur quadratique moyenne de la sortie par rapport à sa référence. On remarque qu'après 1000 itérations, l'erreur quadratique moyenne atteint une valeur de 8.3042×10^{-7} .

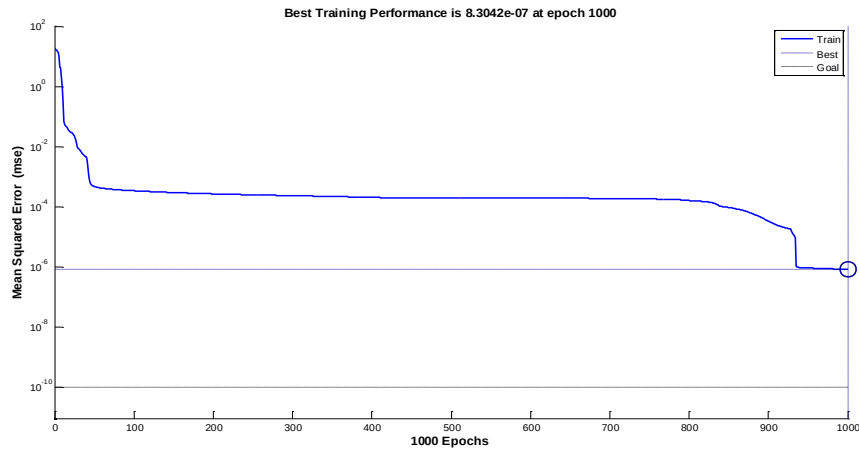


Figure (III.26) : Courbe de performance (évolution de la valeur de l'erreur quadratique moyenne) du réseau de neurones de 3 couches cachées, 1 neurone dans la 1^{ère} couche cachée et 2 neurones dans la 2^{ème} couche cachée et 7 neurones dans la 3^{ème} couche cachée.

Dans la figure (III.27), les sorties du système contrôlé par FEEDBACK linéarisation et les sorties du même système contrôlé par RNA sont présentées.

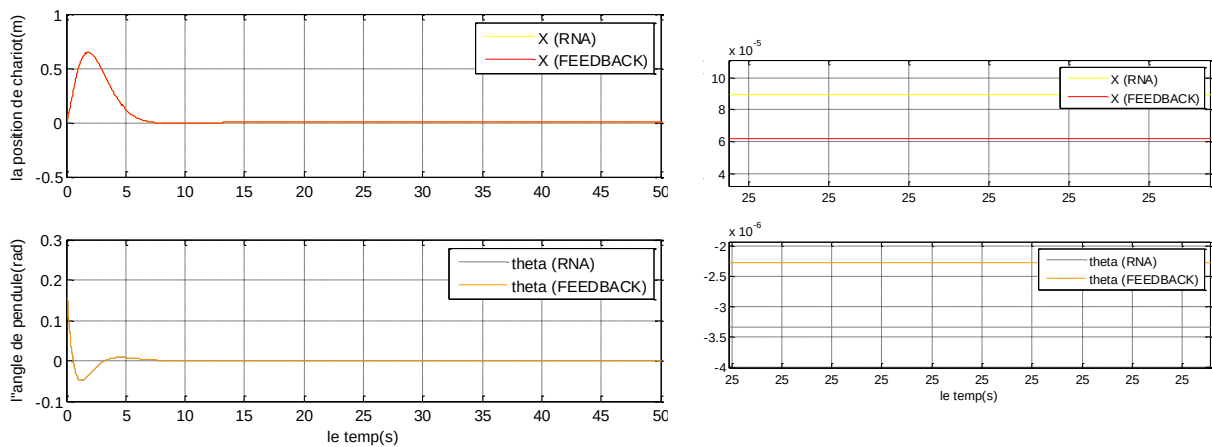


Figure (III.27) : Position et l'angle du pendule inversé commandé par FEEDBACK et par réseau de neurones de 3 couches cachées, 1 neurone dans la 1^{ère} couche cachée et 2 neurones dans la 2^{ème} couche cachée et 7 neurones dans la 3^{ème} couche cachée.

- RNA de 1 neurone dans la 1^{ère} couche cachée et 7 neurones dans la 2^{ème} couche cachée et 2 neurones dans la 3^{ème} couche cachée

Dans cette architecture, on a utilisé un réseau de neurone qui se compose de 3 couches cachées, 1 neurone dans la 1^{ère} couche cachée et 7 neurones dans la 2^{ème} couche cachée et 2 neurones dans la 3^{ème} couche cachée, avec 4 entrées et une seule sortie.

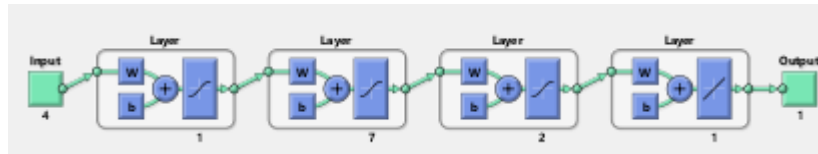


Figure (III.28) : Architecture de réseau de 3 couches cachées, 1 neurone dans la 1^{ère} couche cachée et 7 neurones dans la 2^{ème} couche cachée et 2 neurones dans la 3^{ème} couche cachée.

La figure (III.29) présente les performances d'apprentissage de cette architecture, sous forme de l'évolution de la valeur de l'erreur quadratique moyenne de la sortie par rapport à sa référence. On remarque qu'après 1000 itérations, l'erreur quadratique moyenne atteint une valeur de 1.0701×10^{-9} .

est Training Performance is $1.0701e-09$ at epoch :

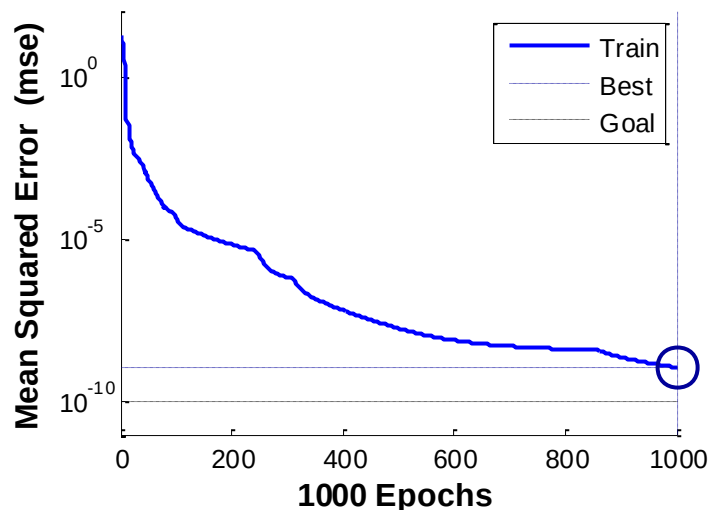


Figure (III.29) : Courbe de performance (évolution de la valeur de l'erreur quadratique moyenne) du réseau de neurones de 3 couches cachées, 1 neurone dans la 1^{ère} couche cachée et 7 neurones dans la 2^{ème} couche cachée et 2 neurones dans la 3^{ème} couche cachée.

Dans la figure (III.30), les sorties du système contrôlé par FEEDBACK linéarisation et les sorties du même système contrôlé par RNA sont présentées.

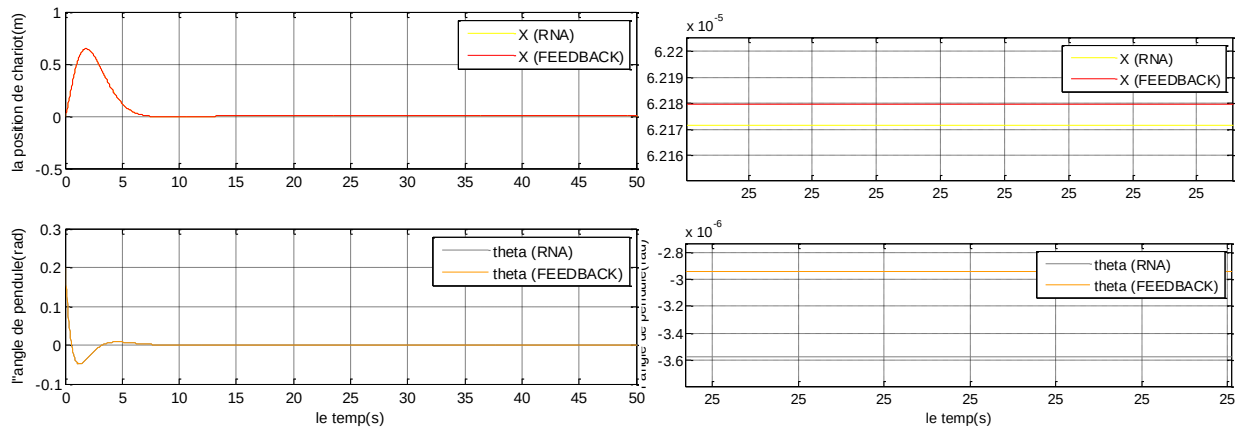


Figure (III.30) : Position et l'angle du pendule inversé commandé par FEEDBACK et par réseau de neurones de 3 couches cachées, 1 neurone dans la 1^{ère} couche cachée et 7 neurones dans la 2^{ème} couche cachée et 2 neurones dans la 3^{ème} couche cachée.

Dans le tableau suivant sont résumées, les erreurs quadratiques moyennes après 1000 itérations lors de l'apprentissage des différentes architectures utilisées pour la commande du pendule inversé avec le même nombre totale de neurones (10 neurones) :

Número d'Architecture du RNA	Type d'Architecture du RNA	Erreur quadratique moyenne après 1000 itérations
1	10	$2.158 \cdot 10^{-9}$
2	7 / 3	$2.573 \cdot 10^{-7}$
3	6 / 4	$8.303 \cdot 10^{-8}$
4	5 / 5	$1.950 \cdot 10^{-8}$
5	4 / 6	$1.052 \cdot 10^{-8}$
6	3 / 7	$2.595 \cdot 10^{-8}$
7	7 / 2 / 1	$6.0703 \cdot 10^{-7}$
8	1 / 2 / 7	$8.307 \cdot 10^{-7}$
9	1 / 7 / 2	$1.070 \cdot 10^{-9}$

Tableau (III.1) : Erreurs quadratiques moyennes après 1000 itérations lors de l'apprentissage des différentes architectures utilisées pour la commande du pendule inversé avec le même nombre totale de neurones

Les résultats de simulation du pendule inversé commandé par un contrôleur à base de différentes architecture de RNA avec position initiale linéaire du chariot de 0 mètre et position initiale angulaire du pendule de 0.2 radian, pour une position angulaire de référence du pendule de 0 radian montre que :

- Similairement à la commande par un contrôleur FEEDBACK linéarisant, le pendule inversé se stabilise au bout de 7 seconds, avec des erreurs statiques très faibles de l'angle du pendule par rapport à sa référence, avec un faible dépassement de -0.05 radian.
- La première architecture et la dernière architecture offrent la moindre erreur quadratique moyenne après 1000 itérations. Par conséquent, elles sont les plus rapides à converger exigeant le moindre nombre d'itérations pour cette convergence avec le minimum temps de calcul par rapport aux autres architectures.

III.3.3 TEST DE ROBUSTESSE DE L'APPRENTISSAGE HORS LIGNE AVEC L'ARCHITECTURE 9

On va essayer de perturber le système en appliquant une impulsion dans l'instant 25s

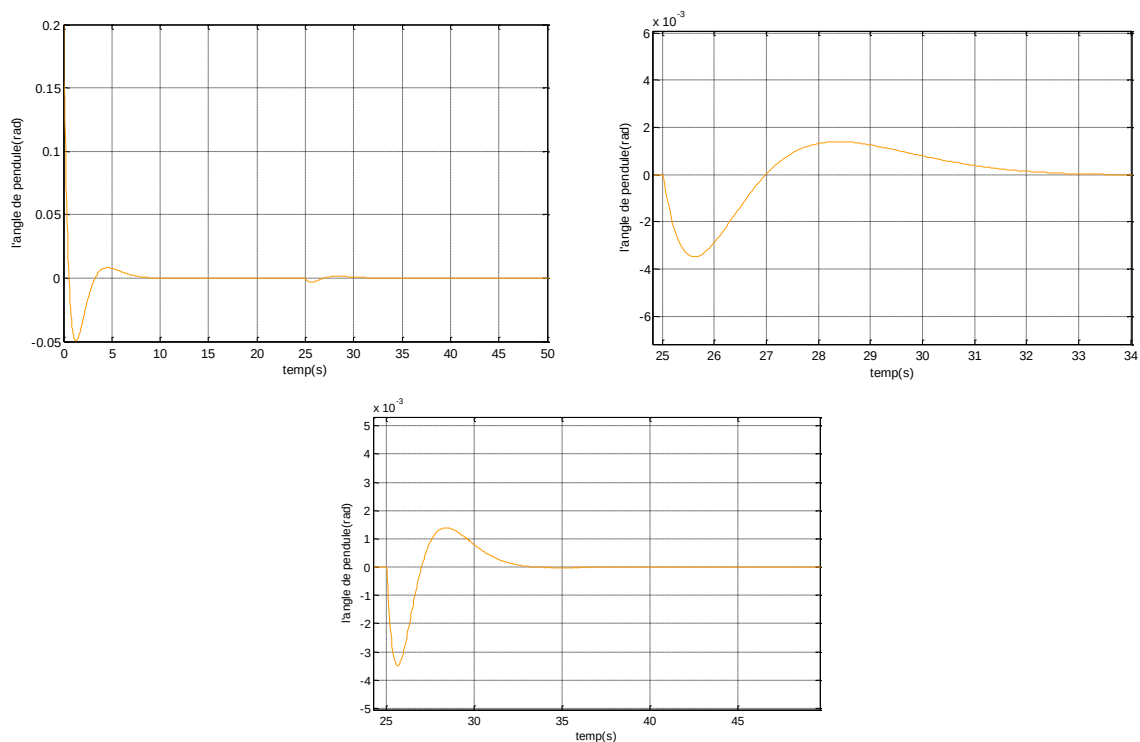


Figure (III.31) : Courbe de l'angle de pendule commandé par réseau de neurone de type apprentissage hors-ligne avec perturbation à l'instant 25 seconde.

Si une perturbation produit dans le processus, le signal d'erreur est grand, ceci ajuste les poids afin que la réponse de système reste suivie.

Les résultats des figures (III.31) montrent qu'après une perturbation produit dans le système, l'angle du pendule inversé a réussi à rejoindre sa valeur de référence (0 radian) après une légère oscillation de 6 secondes et de dépassement maximal de -0.0024 radian. Cela montre l'efficacité du contrôleur neuronal en termes de robustesse.

III.4 CONCLUSION

Dans ce chapitre, nous avons présenté la méthode de commande supervisée par réseau de neurone d'un pendule inversé avec apprentissage hors-ligne basé sur le pendule inversé commandé par contrôleur type feedback linéarisant. En effet, on a utilisé les résultats obtenus en simulant le pendule inversé commandé par un contrôleur type feedback linéarisant pour créer une base de données à but de l'apprentissage hors ligne du réseau de neurone pour la conception du RNA. Différentes architectures de RNA avec le même nombre total de neurones sont proposées et comparées en terme de performance de convergence (nombre d'itérations et erreur quadratique moyenne). Finalement les différentes architectures de RNA sont classifiées.

CONCLUSION GENERALE

Conclusion générale

Au cours de notre travail nous sommes intéressés à l'application du contrôleur neuronal avec différentes architectures (mono couche cachée et multicouches cachées (une des techniques de l'apprentissage profond (deep learning)) pour la stabilisation d'un pendule inversé, dans le but de comparer de leurs caractéristiques.

Au début, nous avons donné un état de l'Art sur les pendules inversées. Nous avons ensuite fait la modélisation de l'ensemble chariot-pendule-MCC. Cette étape nous a permis de déterminer le modèle mathématique de cet ensemble. La modélisation du pendule inversé était nécessaire afin de synthétiser la loi de commande FEEDBACK linéarisation adopté en premier lieu. L'application de ce régulateur pour contrôler et stabilisé le pendule inversé a permis de concevoir une base de donnée (entrées-sorties) qui est utilisée par la suite pour l'apprentissage du régulateur neuronal considéré. Après introduction des différentes notions théoriques concernant les réseaux de neurones tels que l'architecture (réseau de neurones monocouche et réseau de neurones multicouches (DEEP neural network)), la fonction d'activation et l'apprentissage, nous avons procédé à la simulation du système global (ensemble du chariot-pendule-MCC) en boucle ouverte, et ensuite en boucle fermée (pour stabilisation du pendule inversé) avec régulateur FEEDBACK linéarisation. Par la suite, on a procédé à l'apprentissage de différentes architectures du régulateur neuronal utilisé à la place du régulateur FEEDBACK tout en gardant le même nombre global de neurones par régulateur neuronal. Les résultats ont montré que tous les régulateurs convergent, mais avec un temps de convergence relativement différent. Cela implique que certaines architectures convergent plus rapidement que les autres, ce qui implique qu'ils exigent moins de temps et d'espace numérique de calcul pour l'apprentissage. Suite au travail présenté dans ce mémoire, on suggère les perspectives suivantes :

- Faire la même étude sans limitation de nombre d'itérations, mais en fixant un seuil d'erreur toléré pour l'apprentissage comme condition d'arrêt.
- L'application des régulateurs neuronaux avec différentes architectures à d'autres systèmes tels que les systèmes électriques (commande des machines électriques, commande les filtres actifs des réseaux électriques, ...etc.)
- L'application des réseaux de neurones avec différentes architectures comme classificateur d'états des systèmes tels que le diagnostic de défauts.

REFERENCES

BIBLIOGRAPHIQUE

Références Bibliographique :

- [1] **Widrow, B., Lehr, M.A.**, 1990. "30 years of adaptive neural networks : Perceptron, madaline and Backpropagation," Proceeding of the IEEE, vol. 15, pp. 365–428.
- [2] **Emira NECHADI**, « Cours SYSTEMES NON LINEAIRES », université Ferhat Abbas de Sétif 1
- [3] **Matignon D.** "Représentations en variables d'état de modèles de guides d'ondes avec dérivation fractionnaire". PhD Thesis, Université de Paris-Sud, Orsay. 1998.
- [4] **Mullhaupt Ph.** "Introduction à l'Analyse et à la commande des systèmes non linéaires". Lausanne, Juin 2007.
- [5]: **Mudry Freddy.** « Modélisation et régulation d'un pendule inversé. », Fascicule de travaux pratiques, Institut d'Automatisation industrielle de l'Ecole d'Ingénieurs du Canton de Vaud, Suisse 2003.
- [6] **Laurant F. Sylvie G. Andre T.** " Commande floue 1, de la stabilisation à la supervision " Paris, Hermès Sciences Publication, 2003.
- [7] **K.J. Aström, K. Furuta**,« Swinging up a pendulum by energy control», *Automatica*, Vol. 36, N°. 2, 2000, pp. 287 -295, 2000.
- [8] : <https://www.techno-science.net/definition/9026.html> consulté le : 12/06/2020 à 9:50 AM
- [9] : **F.Lahouazi**, « Mise en œuvre d'une stratégie de commande neurofloue : application a un pendule inversé », Mémoire de magister, Université Mouloud Mammeri Tizi-Ouzou, Algérie(2011).
- [10] : **H. Ludvigsen A. Shiriaev, A. Pogromsky and O. Egeland**, « On global properties of passivity based control of an inverted pendulum », *International Journal Of Robust And Nonlinear Control*, (10) :283–300, 2000
- [11] **D.Bensoussan** : « commande moderne ; approche par modèles continus et discrets » presses internationales polytechniques, 2008
- [12] **A. Djoudi , T. Benabdellah**, ‘synthèse et Application des lois de commande nonlinéaires sur un pendule inverse’, Mémoire d'Ingénieur , Ecole Nationale Polytechnique, 2007.
- [13] **Jong H. Park and Kyoung**, ‘Biped robot walking using gravity-compensated inverted pendulum mode and computed troque control ‘. *International Conference on Robotics and Automation*, Belgium, 1998.
- [14] <http://www.onera.fr/dcsd/gyrodynes/>
- [15] **C.Aguilar** ,« approximate feedback linearization and sliding mode contrôl for the single inverted pendulum » queen's university mathematics and engineering, august 2002

- [16] **Menouar MEZIANI, Mourad AKKOUL**, « Modélisation et simulation d'un pendule inversé », Mémoire de fin d'étude de MASTER ACADEMIQUE, Université Mouloud Mammeri de Tizi-Ouzou, 2016
- [17] **S.Riachy** : « contribution a l'estimation et a commande de système mecanique sous-actionnes » docteur de l'école centrale de Lille discipline automatique et informatique industrielle , 2008
- [18] Manuel d'installation du pendule inverse feedback Digital pendulum controller 33-201.
- [19] **Lam J.** "Control of an inverted pendulum". Dept of Electrical &computer Engineering, University of California. U.S.A. 2004
- [20] **Mohamed Yessin Ammar** "Mise en œuvre Mise en œuvre de réseaux de neurones pour la modélisation de cinétique réactionnelle en Batch/continu" Thèse de Doctorat de l'Université de Toulouse 2007.
- [21] **Touzet, C.**, 1992. "Les réseaux de neurones artificiels : introduction au connexionnisme, 150 pages, Préface de Jeanny Hérault," EC2 éd., Paris.
- [22] **Herault, J., Jutten, C.**, 1994. "Les réseaux de neurones et le traitement du signal," Editions Hermès, Paris.
- [23] **Touzet, C.**, 1990. "Contribution à l'étude et au développement de modèles connectionnistes séquentiels de l'apprentissage,"Thèse de Doctorat, Université de Montpellier II.
- [24] **ANANE Hamid , AHMED ZAID Tassadit** « Commande par Modèle Inverse neuronal, application à la commande en vitesse d'un moteur à courant continu » , MEMOIRE DE master ,Université Mouloud MAMMERI Tizi-Ouzou ,2013
- [25] **S.ZEGHLACHE**, « Commande Intelligente, cours », université Mohamed Boudiaf m'sila.2008.
- [26] **S. NEDJAUME** « Entraînement D'un Réseau De Neurones MLP Par La Méthode BFGS », mémoire master, Université Larbi Ben M'hidi Oum El Bouaghi, 2015/2016
- [27] **N. OUKACINE** « Utilisation des réseaux de neurones pour la reconstitution de défauts en évaluation non destructive », mémoire magister, Université Mouloud Mammeri de Tizi-ouzou, 03/07/2012
- [28] **GUIANI Issam** « Commande Adaptative Neuronale par Retour de Sortie des Systèmes Non Linéaires » thèse de magister à l'université de Constantine, 2007
- [29] **Wilamowsky, B.M.**, 2009. "Neural Network Architectures and Learning Algorithmes"
- [30] **Suzuki. K.** artificiel neural networks-methodological Advances and biomedical application.in tech 2011
- [31] **Sabrina YESSAADI** ; "Un modèle basé Template Matching/ Réseau de Neurones pour la Reconnaissance des visages". Memoire de magister. Université Badji Mokhtar, Annaba, 2006.

- [32] **MELAKHESSOU Lakhdar**, « Contrôle et identification des systèmes non linéaires par les techniques neuronales », Thèse de Magister, Université de Batna.
- [33] **ARIF Salim BOUCHAALA Abdelbari** , « Commande de la machine synchrone à aimant permanent (MSAP) par les réseaux de neurones artificiels », mémoire de master, UNIVERSITE KASDI MERBAH OUARGLA, 2019
- [34] **AMOURA Karima**, « Réseaux de neurones et espace d'état pour l'identification et la prédiction » Memoire de magister , Université Mouloud Mammeri de Tizi-Ouzou,
- [35] **H. AZZOUZ, B. ZOUHIR** « diagnostic des défaillances par la technique de réseau de neurone artificiel appliqué sur la machine synchrone a aimants permanents », Mémoire d'ingéniorat de l'Université Mohamed Boudiaf de M'sila, Algérie, Juin 2005.
- [36] **G. Campion, G. Bastin et B. D'Andréa-Novel**. « Structural Properties and Classification of Kinematic and Dynamic Models of Wheeled Mobile Robots » IEEE Transactions on Robotics and Automation, vol. 12, no. 1, pages 47–62, 1996
- [37] **LEMITA ABDALLAH**, « Commande adaptative par structure neuronale récurrente pour la conduite d'un procédé de traitement d'eau usée », Mémoire de magister, UNIVERSITE FERHAT ABBAS – SETIF 1-UFAS (ALGERIE).
- [38] **G. DREYFUS** « les réseaux de neurones, une technique opérationnelle pour le traitement des données industrielles, économiques et financières » École Supérieure de Physique et de Chimie Industrielles de la Ville de Paris (ESPCI), Novembre 1997.
- [39] **I.RIVALS, L PERSONNAZ** « Les réseaux de neurones formels pour la modélisation, la commande et la classification » CNRS Editions, Paris 2003
- [40] **B.GOSSELIN** « Application de réseaux de neurones artificiels a la connaissance automatique de caractères manuscrits » Thèse de Doctorat, université Mons 1996.
- [41] **Kara, K** « application des RNA a l'identification des systèmes non linéaires », thèse de magister, université de Constantine 1995.
- [42] **D. Ould Abdeslam**, *Techniques neuromimétiques pour la commande dans les systèmes électriques : Application au filtrage actif parallèle dans les réseaux électriques basse tension*, Thèse de Doctorat, Université de Haute Alsace, Décembre 2005.
- [43] **P. Wira**, *Approches neuromimétiques pour l'identification et la commande*, Habilitation à Diriger des Recherches, Université de Haute Alsace, Novembre 2009.
- [44] **M. BECHOUCHE Ali**, « utilisation des techniques avancées pour l'observation et la commande d'une machine asynchrone : application à une éolienne », thèse de doctorat, université mouloud Mammeri de tizi-ouzou, 2013
- [45] https://fr.wikipedia.org/wiki/Apprentissage_profond
- [46] <https://orbograph.com/deep-learning-how-will-it-change-healthcare/>
- [47]. **PIERRE APKARIAN**, « ÉLÉMENTS DE LA THÉORIE DE LA COMMANDE ROBUSTE »

[48] Nitin S Bhattathiri, P Anitha Saraswathi : "ANN Control of Non-Linear and Unstable System and its Implementation on Inverted Pendulum". SRM University, Chennai, India, 2014.

RESUME

ملخص:

في هذه الأطروحة، نحن مهتمون بمجال شبكة الخلايا العصبية الاصطناعية ذات البنى المختلفة المطبقة للتحكم في الأنظمة غير الخطية مثل تثبيت النواس المقلوب. تم تصميم قاعدة بيانات المداخل والمخارج (إشارة التحكم-الموضع وزاوية النواس المقلوب) بناءً على محاكاة النواس المقلوب الذي تم التحكم فيه باستخدام وحدة تحكم من نوع FEEDBACK خطية. بعد ذلك، يتم إجراء تعلم البنى المختلفة للمنظم العصبي المستخدم بدلاً من وحدة تحكم FEEDBACK مع الاحتفاظ بنفس العدد الإجمالي للخلايا العصبية لكل وحدة تحكم عصبية. أظهرت النتائج أن جميع بنى وحدات التحكم تتقارب، ولكن مع وقت تقارب مختلف نسبياً. هذا يعني أن بعض البنى تتقارب بشكل أسرع من غيرها، مما يعني أنها تتطلب وقتاً ومساحة حوسبة رقمية أقل للتعلم.

كلمات مفتاحية: النواس المقلوب، شبكة الخلايا العصبية الاصطناعية، وحدة تحكم من نوع FEEDBACK خطية، وحدة تحكم عصبية

Résumé :

Dans ce mémoire, nous sommes intéressés aux réseaux de neurones artificiels avec différentes architectures appliqués pour la commande des systèmes non-linéaires tels que la stabilisation du pendule inversé.

Une base de données entrée-sorties (signal de commande-position angulaire du pendule inversé) est conçu à base de la simulation du pendule inversé contrôlé est stabilisé en utilisant un contrôleur de type FEEDBACK linéarisant. Par la suite, l'apprentissage de différentes architectures du régulateur neuronal utilisé à la place du régulateur FEEDBACK tout en gardant le même nombre global de neurones par régulateur neuronal est effectué. Les résultats ont montré que tous les régulateurs convergent, mais avec un temps de convergence relativement différent. Cela implique que certaines architectures convergent plus rapidement que les autres ce qui implique qu'ils exigent moins de temps et d'espace numérique de calcul pour l'apprentissage.

Mots clés : Pendule inversé, réseau de neurone artificiel, feedback linéarisant, contrôleur neuronal

Abstract:

In this thesis, we are interested in artificial neural networks with different architectures applied for the control of non-linear systems such as the stabilization of the inverted pendulum.

An input-output database (control signal-angular position of the inverted pendulum) is designed based on the simulation of the controlled inverted pendulum is stabilized using a linearizing FEEDBACK type controller. Subsequently, learning of different architectures of the neural regulator used in place of the FEEDBACK regulator while keeping the same overall number of neurons per neural regulator is performed. The results showed that all the regulators converge, but with a relatively different convergence time. This implies that some architectures converge faster than others do, which means that they require less time and digital computing space for learning.

Keywords: inverted pendulum ,artificial neural network, feedback linearization, neural controller