

République Algérienne Démocratique et Populaire
Ministère de l'Enseignement Supérieur et de la Recherche Scientifique
Université Akli Mohand Oulhadj – Bouira



Faculté des sciences et des sciences appliquées
Département de Génie Electrique

Mémoire de fin d'étude

En vue de l'obtention du diplôme de Master en Electronique

Spécialité : Electronique des systèmes embarqués

Thème :

Contrôle d'un pendule inversé par un réseau de neurones artificiels

Réalisé Par :

LAOUBI Mohamed Chems eddine

Encadré par :

Mr.LADJOUZI Samir

Année Universitaire 2017/2018

Remerciements

*Je tiens tout d'abord à remercier Dieu le tout puissant,
qui m'a donné la force et la patience d'accomplir ce modeste travail*

*Arrivé au terme de notre travail je tiens à remercier
mon encadreur Mr S. Ladjouzi de ses précieux conseils et son aide
durant toute la période du travail.*

*Je remercie également mes parents et mes amis pour leurs
encouragements et leurs soutiens*

*En fin j'exprime mes remerciements à tous ceux qui ont contribué de
prés et de loin à l'élaboration de ce travail*

Table des matières

Introduction générale..... 1

Chapitre I: Généralités sur les réseaux de neurones artificiels

I.1.	Introduction	3
I.2.	Historique	3
I.3.	Neurone biologique	4
I.4.	Neurone artificiel.....	5
I.5.	Composition d'un réseau de neurones.....	8
I.6.	Architecture des réseaux de neurones	8
I.6.1.	Réseau de neurones non bouclé (feed-forward).....	8
I.6.2.	Réseau de neurones bouclé (feed-back).....	9
I.7.	Apprentissage d'un réseau de neurones	10
I.7.1.	Apprentissage supervisé.....	10
I.7.2.	Apprentissage semi-supervisé.....	10
I.7.3.	Apprentissage non supervisé.....	11
I.8.	Algorithmes d'apprentissage	11
I.8.1.	L'algorithme de la rétro-propagation du gradient.....	11
I.8.2.	L'algorithme de Levenberg-Marquardt	13
I.8.3.	Régularisation Bayésienne.....	13
I.8.4.	Méthode de gradient conjugué.....	14
I.9.	Quelques types de réseaux de neurones	14
I.9.1.	Réseaux à une seule couche (monocouche).....	14
I.9.2.	Perceptron Multi Couches.....	15
I.9.3.	Réseau à fonction radiale	16
I.10.	Applications	16
I.11.	Conclusion	17

Chapitre II : Description du système pendule inversé

II.1	Introduction	18
II.2	Qu'est-ce qu'un pendule ?	18
II.3	Principe de fonctionnement	19

II.4. Intérêt de l'étude d'un pendule inversé	19
II.4.1. Le domaine de la médecine	20
II.4.2. Le domaine de Robotique.....	20
II.4.3. Le domaine de l'aérospatiale	21
II.4.4. Dans La Recherche	22
II.5. Modélisation du pendule inversé	22
II.5.1. Energie cinétique du système	24
II.5.2. L'énergie potentielle du système	25
II.5.3. Equations d'Euler-Lagrange.....	25
II.5.4. Modèle d'état de l'ensemble chariot-pendule :	26
II.6. Elaboration du modèle avec SIMULINK.....	27
II.7. Résultat de simulation	28
II.8 Conclusion	30

Chapitre III : Contrôle du pendule inversé par un réseau de neurones artificiels

III.1 Introduction	31
III.2 Conception d'un contrôleur pour le pendule inversé	31
III.2.1 Equations du contrôleur	31
III.2.2 Élaboration du contrôleur avec SIMULINK.....	32
III.3 Stabilisation du pendule inversé par le contrôleur	32
III.4 Stabilisation du pendule inversé par un réseau de neurones	34
III.4.1. Apprentissage avec l'algorithme Levenberg-Marquardt.....	34
III.4.2. Apprentissage avec la méthode de régularisation Bayésienne (RB).....	39
III.5. Interprétation des résultats	42
III.6 Conclusion.....	43
Conclusion générale	44
Bibliographie	45

Liste des figures

<i>Figure I. 1 : Représentation schématique d'un neurone biologique.....</i>	<i>5</i>
<i>Figure I. 2 : Correspondance entre neurone artificiel et neurone biologique.....</i>	<i>5</i>
<i>Figure I. 3 : La géométrie du neurone artificiel.....</i>	<i>6</i>
<i>Figure I. 4: Fonction linéaire.....</i>	<i>7</i>
<i>Figure I. 5 : Fonction Sigmoidale</i>	<i>7</i>
<i>Figure I. 6 : Fonction gaussienne</i>	<i>7</i>
<i>Figure I. 7 : Composition d'un réseau de neurones.....</i>	<i>8</i>
<i>Figure I. 8 : Réseau non bouclé à connexions partielles</i>	<i>9</i>
<i>Figure I. 9 : Réseau non bouclé à connexions totales.....</i>	<i>9</i>
<i>Figure I. 10 : Structure d'un réseau de neurones bouclées.....</i>	<i>9</i>
<i>Figure I. 11 : Apprentissage supervisé.....</i>	<i>10</i>
<i>Figure I. 12 : Apprentissage non supervisé.....</i>	<i>11</i>
<i>Figure I. 13: Réseau à une seule couche.....</i>	<i>14</i>
<i>Figure I. 14 : Architecture du perceptron multicouche.....</i>	<i>15</i>
<i>Figure I. 15: Structure d'un réseau de neurones de type RBF</i>	<i>16</i>
<i>Figure II. 1 : Pendule inversé en coordonnées généralisées x et θ</i>	<i>18</i>
<i>Figure II. 2 : Exemple sur l'équilibre à base du pendule inversé.</i>	<i>19</i>
<i>Figure II. 3 :Le corps de l'être humain vu comme un double pendule inversé.....</i>	<i>20</i>
<i>Figure II. 4: Le Segway.....</i>	<i>21</i>
<i>Figure II. 5: Le robot BIPED.....</i>	<i>21</i>
<i>Figure II. 6 : Pendule gyroscopique inversé.....</i>	<i>22</i>
<i>Figure II. 7 : Modèle d'un pendule inversé</i>	<i>23</i>
<i>Figure II. 8 : Schéma Simulink du pendule inversé.....</i>	<i>28</i>
<i>Figure III. 1 : Le model Simulink de la loi de control non lineaire.....</i>	<i>32</i>
<i>Figure III. 4 : Schéma de la commande du pendule inversé.....</i>	<i>33</i>
<i>Figure III. 5 : Résultat de simulation de stabilisation du pendule inversé par le contrôleur.</i>	<i>33</i>
<i>Figure III. 6 : architecture du réseau</i>	<i>34</i>
<i>Figure III. 7 : Commande du pendule inversé par le RNA avec l'algorithme LM.....</i>	<i>35</i>
<i>Figure III. 8 : Stabilisation du pendule inversé par le RNA avec l'algorithme LM.....</i>	<i>35</i>
<i>Figure III. 9 : l'outil d'apprentissage du RNA par l'algorithme LM.....</i>	<i>36</i>
<i>Figure III. 10 : MSE d'apprentissage du RNA par l'algorithme LM.....</i>	<i>37</i>
<i>Figure III. 12 : Schéma pour calculer le MSE entre le neuro-contrôleur et le contrôleur.....</i>	<i>37</i>
<i>Figure III. 13 : l'erreur MSE de la commande par RNA et le contrôleur</i>	<i>38</i>
<i>Figure III. 14 :Commande du pendule inversé par le RNA avec l'algorithme RB.....</i>	<i>39</i>
<i>Figure III. 15 : Stabilisation du pendule inversé par le RNA avec l'algorithme RB.....</i>	<i>40</i>
<i>Figure III. 16: l'apprentissage du RNA par la méthode RB.</i>	<i>40</i>
<i>Figure III. 17 : L'erreur quadratique MSE d'apprentissage par l'algorithme RB</i>	<i>41</i>
<i>Figure III. 18 :Schéma pour calculer le MSE entre le neuro-contrôleur et le contrôleur</i>	<i>41</i>
<i>Figure III. 19 : l'erreur entre la commande du RNA et le contrôleur avec l'algorithme RB.</i>	<i>42</i>

Liste des tableaux

<i>Tableau II. 1 : Paramètres de l'ensemble chariot et pendule.....</i>	<i>23</i>
<i>Tableau III. 1: MSE entre le contrôleur et le RNA avec trainlm</i>	<i>38</i>
<i>Tableau III. 2: MSE entre le contrôleur et le RNA avec l'algorithme RB</i>	<i>42</i>
<i>Tableau III. 3 : Comparaison entre MSE de trainlm et trainbr.....</i>	<i>43</i>

Liste des abréviations

<i>b</i>	<i>biais</i>
<i>CMG</i>	<i>Control Moment Gyro</i>
<i>Ec</i>	<i>Energie Cinétique</i>
<i>Ecc</i>	<i>Energie cinétique du chariot</i>
<i>Ecp</i>	<i>Energie cinétique du pendule</i>
<i>Ep</i>	<i>Energie potentielle</i>
<i>F</i>	<i>Force</i>
<i>FF</i>	<i>Feed Forward</i>
<i>FLN</i>	<i>Functional Link Network</i>
<i>FPGA</i>	<i>Field Programmable Gate Array</i>
<i>IA</i>	<i>Intelligence Artificielle</i>
<i>L</i>	<i>Lagrangien</i>
<i>LFD</i>	<i>Loi Fondamentale de la Dynamique</i>
<i>LM</i>	<i>Levenberg Marquardt</i>
<i>MLP</i>	<i>Multi Layer Perceptron</i>
<i>MSE</i>	<i>Mean Square Error</i>
<i>MSE reg</i>	<i>Mean square error reguled</i>
<i>MSW</i>	<i>Mean Square Weight</i>
<i>PID</i>	<i>Proportional Integral Derivative</i>
<i>PMC</i>	<i>Perceptron Multi Couches</i>
<i>RB</i>	<i>Régularisation Bayésienne</i>
<i>RBF</i>	<i>Radial Basis Function</i>
<i>RNA</i>	<i>Réseaux de Neurones Artificiels</i>
<i>SIMO</i>	<i>Single Input Multiple Output</i>
<i>W</i>	<i>Wight</i>

Introduction générale

L'industrie moderne a besoin d'obtenir les meilleurs résultats pour la commande des systèmes travaillant toujours aux limites des contraintes. Les scientifiques proposent plusieurs méthodes de commande comme éventuelle solution, l'une de ces méthodes est l'utilisation de la technique de réseaux de neurones artificiels (RNA) dans le processus de contrôle, soit dans la conception du modèle soit dans la conception du contrôleur. Les réseaux de neurones artificiels constituent l'une des approches d'intelligence artificielle (IA) dont le développement se fait à travers les méthodes par lesquelles l'homme essaye toujours d'imiter la nature et de reproduire des modes de raisonnement et de comportement qui lui sont propres. Ces derniers présentent plusieurs avantages notamment comme approximateurs globaux qui peuvent fournir des solutions efficaces aux problèmes de contrôle des systèmes non linéaires [1]

Le pendule inversé est l'un de ces systèmes non linéaires qui occupe une place importante dans l'industrie comme un outil de transport, qui 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. 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, intrinsèquement instable et il possède plusieurs implications pratiques [2].

Dans le cas des systèmes non-linéaires, les techniques de commande classiques telles que le PID ont montré souvent leur insuffisance surtout quand les systèmes à étudier présentent de fortes non linéarités. Le manque de connaissances a priori nécessaire pour l'élaboration du modèle mathématique était en quelque sorte dans cet échec.

Face à ce problème, le recours aux méthodes de commandes par apprentissage est devenu une nécessité car les systèmes de commande obtenus ainsi procèdent par collecte de données empiriques, stocker et extraire les connaissances contenues dans celle-ci et utiliser ces connaissances pour réagir à de nouvelles situations : on est passé à la commande intelligente. La possibilité d'apprentissage peut réduire l'effort humain lors de la conception des contrôleurs et permet de découvrir des structures de contrôle plus efficaces que celles déjà connues [3].

L'objet de ce travail est l'étude et l'application d'une commande avec un réseau de neurones pour la stabilisation d'un pendule inversé qui est un système non linéaire, instable, SIMO (Single Input Multiple Output). Afin d'illustrer et comparer les différentes règles d'apprentissage pour le réseau de neurones, on va appliquer deux algorithmes et leurs tests de performances mis en œuvre.

Pour cela, Le présent mémoire est organisé en trois chapitres qui sont résumés comme suit :

Dans Le premier chapitre on va présenter quelques généralités et techniques de bases des réseaux de neurones qui nous serviront lors de la phase de stabilisation du pendule inversé.

Le deuxième chapitre est consacré à la description et modélisation du pendule inversé, et les modèles obtenus sont utilisés pour la simulation du système.

Le troisième chapitre est consacré à la conception d'un réseau de neurones pour contrôler le pendule inversé, et la discussion des résultats obtenue de la simulation.

Enfin une conclusion générale clôture le mémoire.

Chapitre I : Généralités sur les réseaux de neurones artificiels

I.1. Introduction

Les réseaux de neurones artificiels (RNA) constituent l'une des approches d'intelligence artificielle (IA) dont le développement se fait à travers les méthodes par lesquelles l'homme essaye toujours d'imiter la nature et de reproduire des modes de raisonnement et de comportement qui lui sont propre [1]

Nous présentons dans ce premier chapitre un état de l'art de ces réseaux de neurones. D'abord, nous rappellerons l'historique et la définition des réseaux de neurones, avant de décrire les architectures neuronales les plus utilisées, à savoir les modèles statiques et dynamiques. Nous poursuivrons en exposant les différents types d'apprentissage et quelques exemples des réseaux de neurones, ainsi que leurs domaines d'applications.

I.2. Historique

L'histoire des réseaux de neurones artificiels revient en 1943, où Mc Culloch et Pitts ont proposé des neurones formels mimant les neurones biologiques et capables de mémoriser des fonctions booléennes simple [4].

En 1949, Hebb a mis en évidence l'importance du couplage synaptique dans l'apprentissage par renforcement ou dégénérescence des liaisons inter-neuronales lors de l'interaction du cerveau avec le milieu extérieur.

Le premier modèle opérationnel est le perceptron simple inspiré du modèle visuel et capable d'apprentissage. Il a été proposé en 1958 par Rosenblatt.

Les limites du Perceptron monocouche du point de vue performance ont été montrées en 1969 par les mathématiciens Minsky et Papert.

Les travaux de Hopfield en 1982 ont montrés que des réseaux de neurones artificiels étaient capables de résoudre des problèmes d'optimisation et ceux de Kohonen (1982) ont montré qu'ils étaient capables des résoudre des tâches de classification et de reconnaissance [4].

En 1988, Broomhead et Lowe s'inspirant des fonctions radiales, développèrent les réseaux RBF (Radial Basis Function). Il ouvrirent la voie a Chen et Billings (1992) et leurs

Functional Link Network (FLN) ou une transformation non linéaire des entrées permet une réduction de la puissance de calcul ainsi qu'une convergence plus rapide.

En 2013, Les réseaux de neurones profonds sont similaires aux réseaux PMC mais avec plus de couches cachées. Plus le nombre de neurones est élevé, plus le réseau est « profond ». Les réseaux profonds obtiennent d'excellentes performances pour la classification d'images et de sons.

En 2015, les géants du logiciel (Google, Facebook, IBM, etc.) ont recruté pratiquement tous les experts (universitaires) en réseaux de neurones profonds pour développer de nouvelles applications.

Nous pouvons donc estimer que les RNA ont répondu à nos attentes et sont en passe de résoudre certains des challenges de l'IA [5].

I.3. Neurone biologique [6]

Les cellules nerveuses, appelées neurones, sont les éléments de base du système nerveux central. Celui-ci en posséderait environ cent milliards. Les neurones possèdent de nombreux points communs dans leur organisation générale et leur système biochimique avec les autres cellules.

Le neurone se compose essentiellement de:

- ◆ **Corps cellulaire (soma)** : il est centré par un noyau, toutes les informations recueillies par les synapses sont acheminées vers le corps cellulaire
- ◆ **Synapse** : qui est une jonction entre deux neurones ; et généralement entre l'axone d'un neurone et une dendrite d'un autre neurone.
- ◆ **Dendrites** : ce sont de fines extensions tubulaires qui se ramifient autour du neurone et forment une sorte de vaste arborescence. Elles captent les signaux envoyés au neurone.
- ◆ **L'axone** : qui est la partie qui s'occupe de la transmission de l'information issue du corps cellulaire ; conduisant des signaux électriques de la sortie d'un neurone vers l'entrée d'un autre neurone.

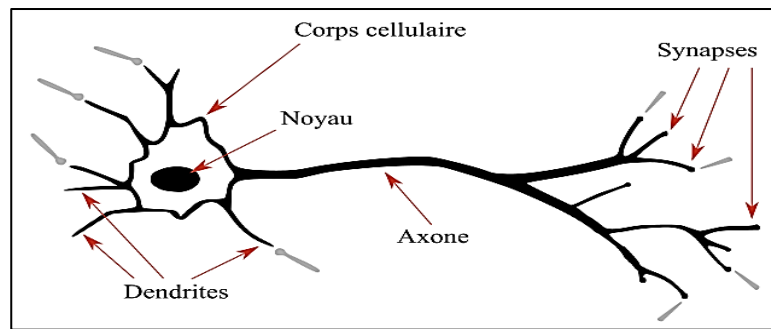


Figure I. 1 : Représentation schématique d'un neurone biologique

I.4. Neurone artificiel

Un neurone artificiel est un modèle simplifié de neurone biologique (figure I.2), ce modèle imite quelques fonctions du neurone biologique, comme la mémorisation associative, l'apprentissage et le travail en parallèle [7].

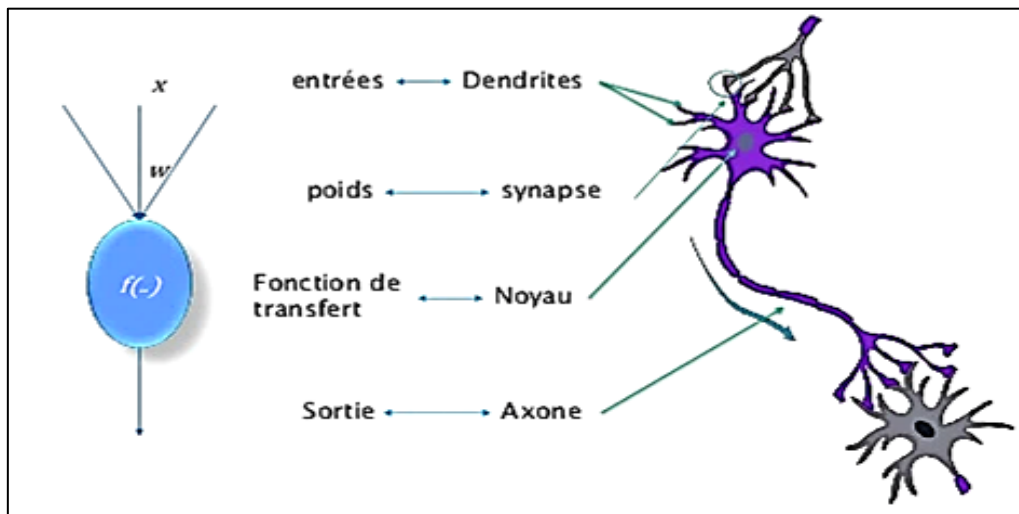


Figure I. 2 : Correspondance entre neurone artificiel et neurone biologique

Par analogie avec la biologie, un neurone artificiel (figure I.3), reçoit l'information provenant des entrées x_i ($i = 1, 2, 3 \dots n$) par l'intermédiaire des connexions dont on affecte à chacune d'elles un poids w_i abréviation de Wight en anglais pondérant l'information, et aussi représentatif de la force de la connexion. Le neurone artificiel fonctionne en deux étapes [7] :

La première phase représente les prétraitements des données reçus en calculant le potentiel v_j des neurones j par la fonction suivante :

$$v_j = b_j + \sum_{i=1}^n w_{i,j} x_i \quad (\text{I.1})$$

Où

$w_{i,j}$: désigne le poids de la connexion liant le neurone j à l'entrée i ;

b_j : terme constant appelé biais, il est considéré comme le poids d'une entrée x_0 égal à 1.

Ainsi la relation s'écrit plus simplement :

$$v_j = \sum_{i=0}^n w_{i,j} x_i \tag{I.2}$$

Dans la deuxième phase, une fonction de transfert g appelée également **fonction d'activation**, calcule la valeur de l'état interne S_j du neurone j à partir de la valeur du potentiel v_j . Cette valeur désignera la sortie du neurone :

$$S_j = g(v_j) = g\left(\sum_{i=0}^n w_{i,j} x_i\right) \quad (i = 1, 2, \dots, n) \tag{I.3}$$

Le choix de la fonction d'activation se révèle dans certains cas être un élément constitutif important des réseaux de neurones. Ainsi, le neurone peut être défini mathématiquement comme étant une fonction algébrique, non linéaire (suivant g) et bornée, des entrées x_i et paramétrée par les poids $w_{i,j}$:

$$S_j = g(x_i, w_{j,i}) = g_{w_{j,i}}(x_i) \quad (i = 1, 2, \dots, n) \tag{I.4}$$

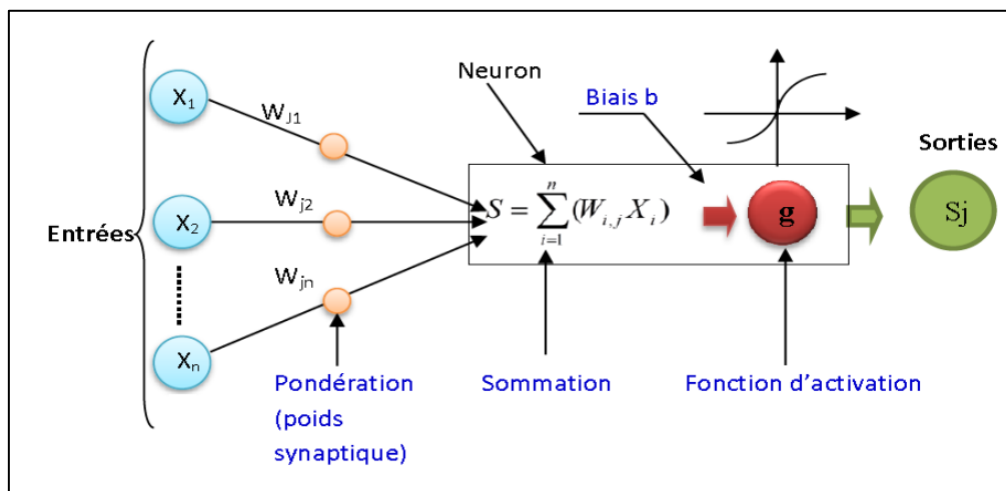


Figure I. 3 : La géométrie du neurone artificiel

a. Fonction d'activation

La fonction d'activation définit le potentiel de sortie d'un neurone en termes de niveaux d'activité de ses entrées. Il existe plusieurs fonctions d'activations, les plus utilisées sont [7]:

➤ Fonction linéaire

$$f(x) = x$$

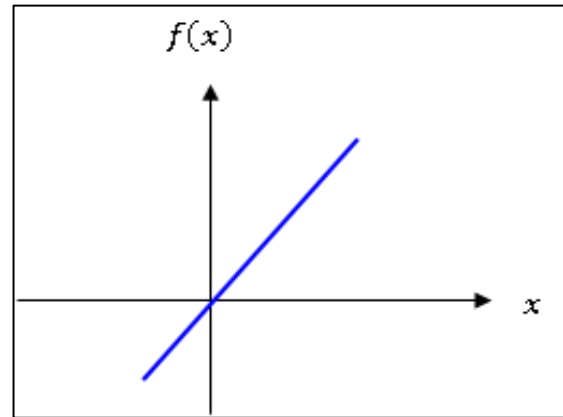


Figure I. 4: Fonction linéaire

➤ Fonction Sigmoide

$$f(x) = \frac{1}{1 + e^{-bx}}$$

b : Gain de la fonction d'activation.

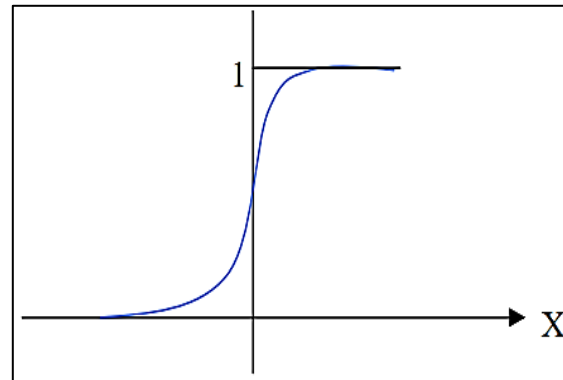


Figure I. 5 : Fonction Sigmoide

➤ Fonction gaussienne

$$f(x) = \exp\left(-\frac{x^2}{2\sigma^2}\right)$$

σ : l'écart type de la gaussienne

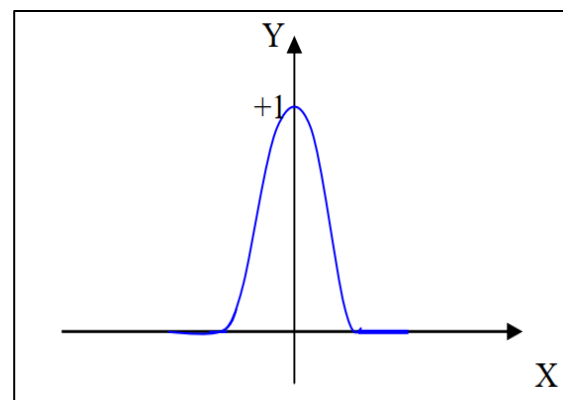


Figure I. 6 : Fonction gaussienne

I.5. Composition d'un réseau de neurones

Un RNA est constitué généralement de trois couches [7]:

- ◆ **Une couche d'entrée** : Elle est constituée de l'ensemble des variables d'entrée.
- ◆ **Une couche de sortie** : Elle est constituée de l'ensemble des neurones de sortie du réseau. C'est cette couche-là qui fournit les sorties principales.
- ◆ **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. En général, les fonctions d'activations sont non linéaires au niveau de ces couches.

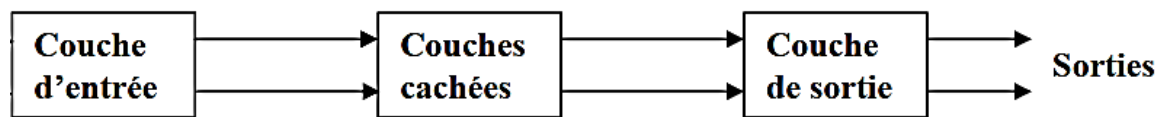


Figure I. 7 : Composition d'un réseau de neurones

I.6. Architecture des réseaux de neurones

Suivant la logique d'interconnexion choisie, les réseaux de neurones se distinguent en deux grandes familles : les réseaux non bouclés (statiques) et les réseaux bouclés (dynamiques).

I.6.1. Réseau de neurones non bouclé (feed-forward)

Un réseau de neurones non bouclé est représenté graphiquement par un ensemble de neurones connectés entre eux. L'information circulant des entrées vers les sorties sans retour en arrière. C'est à dire si l'on se déplace dans le réseau à partir d'un neurone quelconque ensuivant les connexions, on ne peut pas revenir au neurone de départ. Les réseaux de neurones non bouclés sont des outils statiques, utilisés principalement pour effectuer des tâches d'approximation de fonctions non linéaires et de modélisation de processus non linéaires [8].

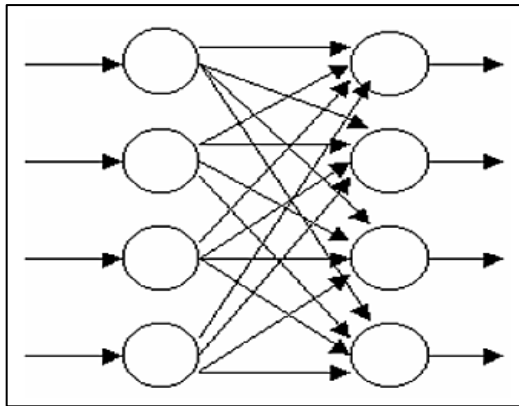


Figure I. 9 : Réseau non bouclé à connexions totales

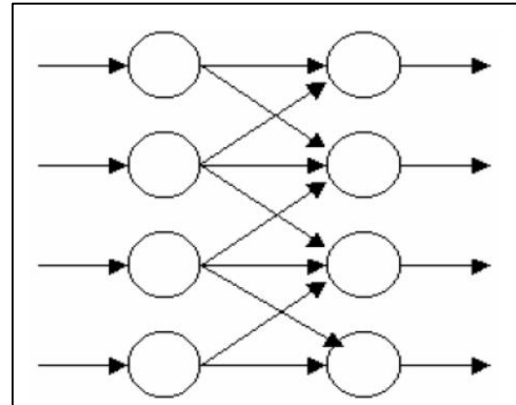


Figure I. 8 : Réseau non bouclé à connexions partielles

I.6.2. Réseau de neurones bouclé (feed-back)

Ce sont des réseaux qui ont un ou plusieurs rebouclages internes, leurs sorties à un instant donné dépendront des entrées aux mêmes instants, et aux instants antérieurs. Ces connexions récurrentes ramènent l'information en arrière par rapport au sens de propagation. Les rebouclages rajoutent donc un effet de mémorisation du passé. Ces réseaux de neurones bouclés constituent un système dynamique "à temps discret", régi par une (ou plusieurs) équation aux différences non linéaires, résultant de la composition des fonctions réalisées par chacun des neurones et des retards associés à chacune des connexions. Ils sont utilisés pour effectuer des tâches de modélisation et d'adaptation de systèmes dynamiques, de commande de processus, ou de filtrage [8].

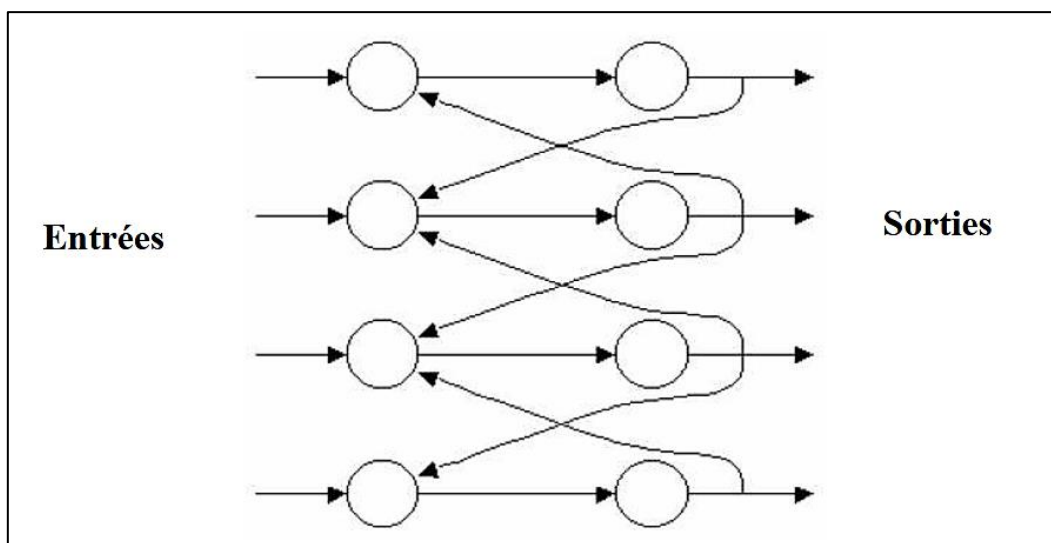


Figure I. 10 : Structure d'un réseau de neurones bouclées

I.7. Apprentissage d'un réseau 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.

Au niveau des types d'apprentissage, il a été défini trois grandes classes selon que l'apprentissage est dit supervisé, semi- supervisé et non supervisé [9]

I.7.1. Apprentissage supervisé

Ce mode est le plus courant, l'utilisateur dispose d'un comportement de références précis qu'il désire faire apprendre au réseau. Schématiquement, il consiste à évaluer les poids synaptiques minimisant l'erreur sur une base d'apprentissage, cette base comprend un ensemble d'observations pour lesquelles on connaît à la fois les entrées et les sorties souhaitées.

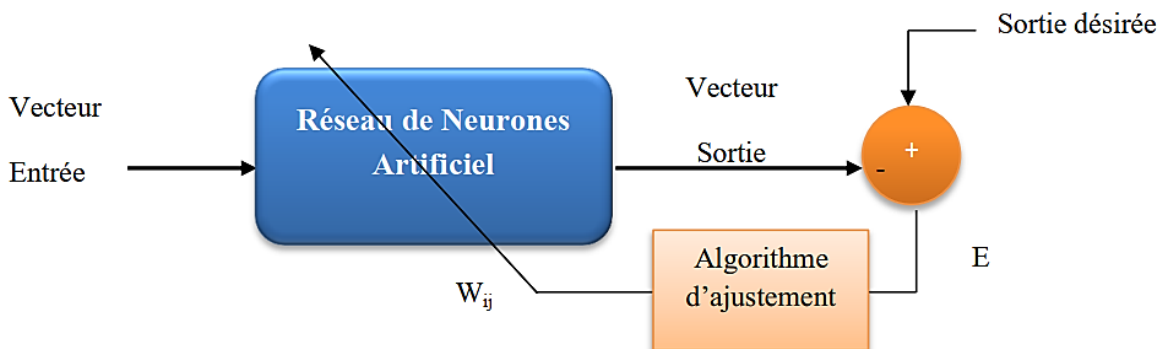


Figure I. 11 : Apprentissage supervisé

I.7.2. Apprentissage semi-supervisé

Cet apprentissage est identique au précédent dans la mesure où il se base sur la présence d'un concepteur, cependant la valeur exacte de la sortie n'est pas disponible, en général la seule information disponible est un signal d'échec ou de succès.

I.7.3. Apprentissage non supervisé

Ce mode d'apprentissage est moins intuitif, il correspond au cas où l'on ne dispose pas de bases d'apprentissage, par exemple lorsqu'on ne sait pas à priori déterminer ponctuellement si une sortie est ou non valable. L'apprentissage repose alors sur un « critère interne » de conformité du comportement du réseau par rapport à des spécifications générales et non sur des observations.

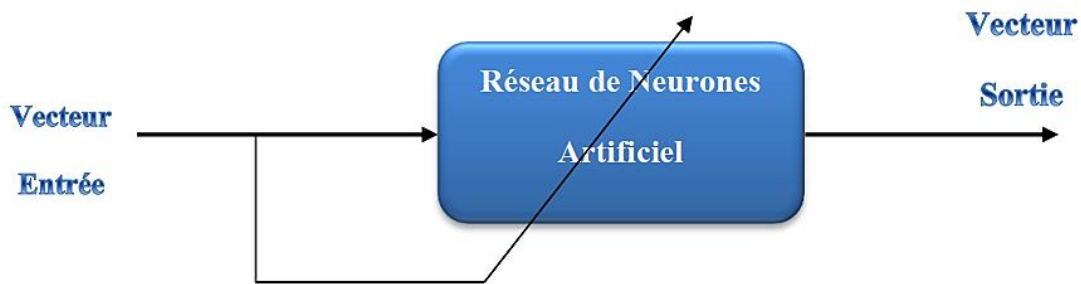


Figure I. 12 : Apprentissage non supervisé

I.8. Algorithmes d'apprentissage

L'apprentissage est une phase de développement du réseau de neurones durant laquelle le comportement de réseau est modifié jusqu'à l'obtention du comportement désiré. Cette phase nécessite des exemples d'apprentissage et un algorithme d'apprentissage. Le problème d'optimisation consiste à trouver les coefficients du réseau minimisant une fonction d'erreur globale, appelée fonction de coût. Cette optimisation se fait de manière itérative, en modifiant les poids en fonction du gradient de la fonction de coût. L'algorithme d'apprentissage le plus communément utilisé est celui de rétro-propagation de l'erreur

I.8.1. L'algorithme de la rétro-propagation du gradient

Plusieurs variantes de cet algorithme ont été développées par des chercheurs travaillant dans des domaines différents. La première formulation de la version actuelle a été faite par Werbos en 1974. Il a été appliqué pour les réseaux multicouches par Rumlethart en 1986.

L'algorithme de la rétro-propagation altère les coefficients synaptiques du réseau dans le sens inverse du gradient du critère d'erreur, en utilisant seulement les données d'entrée/sortie; A chaque itération, on retire un exemple d'apprentissage et on calcule une nouvelle estimation du poids synaptique. Cette itération consiste en deux phases [10] :

- **Propagation** : à chaque itération, un élément de l'ensemble d'apprentissage est introduit à travers la couche d'entrée. L'évaluation des sorties du réseau se fait couche par couche, de l'entrée vers la sortie .
- **Rétro-propagation** : cette étape est similaire à la précédente. Cependant, les calculs s'effectuent dans le sens inverse (de la sortie vers l'entrée). À la sortie du réseau, on forme le critère de performance E en fonction de la sortie réelle de système et sa valeur désirée. Puis, on évalue le gradient de E par rapport aux différents poids en commençant par la couche de sortie et en remontant vers la couche d'entrée

L'algorithme de la rétro-propagation du gradient est un algorithme itératif conçu pour minimiser un critère quadratique d'erreur entre la sortie obtenue d'un réseau multicouche et la sortie désirée. Cette minimisation est réalisée par une configuration des poids adéquate.

L'erreur est la différence entre la valeur désirée pour le neurone de sortie et sa valeur calculée par propagation. En effet, l'algorithme nécessite une fonction continue, non-linéaire et différentiable comme fonction de transfert du neurone [7]

➤ **Expression de la fonction de coût**

Afin de quantifier et mesurer la pertinence et du coup le taux d'erreur du réseau de neurones, nous avons désormais besoin de ce que l'on appelle une **fonction de coût** ou de perte ("loss or **cost function**" en anglais).

La fonction la plus couramment utilisée, et dont nous sommes servis lors de cette étude est la fonction dite fonction d'erreur quadratique moyenne (MSE pour Mean Square Error) dont la définition est donnée comme suit [11] :

Soit une base d'apprentissage constituée de N exemples. Pour chaque exemple n ($n \leq N$), l'erreur $e(n)$ est calculée comme étant la différence entre la cible $d_i(n)$ et la valeur de sortie de réseau $y_i(n)$ (Equation I-5).

$$e(n) = d_i(n) - y_i(n) \quad (\text{I-5})$$

Pour tout l'ensemble N de l'apprentissage la fonction erreur MSE est donnée par l'Equation (I-6) :

$$MSE = \frac{1}{N} \sum_{n=1}^N e(n)^2 \quad (\text{I-6})$$

Plus la valeur de cette fonction de coût est petite, plus le modèle reproduit fidèlement les observations utilisées pour l'apprentissage. Le but de l'algorithme d'apprentissage consiste donc à ajuster les poids de réseau afin de réduire cette erreur quadratique à son minimum.

I.8.2. L'algorithme de Levenberg-Marquardt

L'algorithme de Levenberg-Marquardt (LM) appartient à la classe des méthodes quasi-newtoniennes. Il s'agit d'une méthode itérative de minimisation de fonctions qui consiste à modifier les paramètres θ_k selon la formule suivante (Equation I-7) [11]

$$\theta_{k+1} = \theta_k - [H(\theta_k) + \mu_{k+1}I]^{-1} \nabla J(\theta_k) \quad (\text{I-7})$$

où $H(\theta_k)$ est la matrice Hessienne de la fonction de coût J , I la matrice identité, μ_{k+1} un scalaire appelé pas et $\nabla J(\theta_k)$ le gradient de la fonction de coût.

Le calcul de l'inverse de la matrice $[H(\theta_k) + \mu_{k+1}I]^{-1}$ peut s'effectuer par des méthodes d'inversion directe.

L'algorithme de Levenberg-Marquardt est le plus rapide et assure la meilleure convergence vers un minimum de l'erreur quadratique, pour les problèmes d'approximation des fonctions où le nombre des poids du réseau est inférieur à cents. Quand le nombre de poids augmente l'efficacité de l'algorithme LM diminue [12]

I.8.3. Régularisation Bayésienne

La méthode de régularisation Bayésienne (RB) consiste à modifier la fonction de coût (MSE) en lui ajoutant un terme de régularisation (MSW) égal à la somme des carrés des poids (W_{ij}) et des biais (b_k) afin de pénaliser les valeurs absolues élevées des poids. On obtient donc une fonction du coût MSE « régulée » MSE_{reg} qui est donnée comme suit (Équation I-8) [11]:

$$MSE_{reg} = \gamma MSE + (1 - \gamma) MSW \quad (\text{I-8})$$

Avec

$$MSW = \frac{1}{N_p} \sum_{j=1}^{N_p} w_j^2 \quad (\text{I-9})$$

où γ est un scalaire positif qui pondère l'importance des valeurs des poids sur l'erreur quadratique moyenne, N_p le nombre de paramètres du réseau et w_j un des poids W_{ij} ou des biais b_k du réseau de neurones.

Ainsi, la régularisation Bayésienne minimise la combinaison linéaire de MSE_{reg} pour obtenir une valeur optimale de γ . Cette modification provoque une diminution des poids et biais et force le réseau à avoir une bonne réponse, en évitant le sur-apprentissage. La régularisation est résolue avec l'algorithme de Levenberg-Marquardt. Ce dernier assure une mesure du nombre de paramètres (poids et biais) effectivement utilisés par le réseau.

I.8.4. Méthode de gradient conjugué

La méthode du gradient conjugué est plus souvent utilisée comme méthode itérative pour résoudre de manière approchée les grands systèmes non linéaires. Sa force vient du fait que le nombre d'itérations nécessaires pour obtenir une bonne approximation de x est petit devant la taille N du système. Elle repose sur la recherche de directions successives permettant d'atteindre la solution exacte du système étudié [10].

I.9. Quelques types de réseaux de neurones

Il existe plusieurs types de réseaux de neurones, nous citons principalement

I.9.1. Réseaux à une seule couche (monocouche)

La structure d'un réseau monocouche est telle que des neurones organisés en entrée soient entièrement connectés à d'autres neurones organisés en sortie par des poids modifiables.

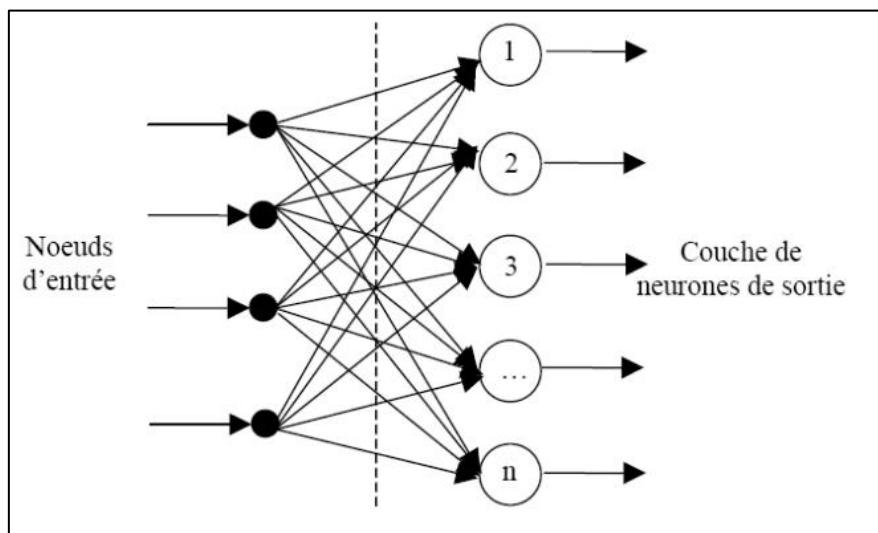


Figure I. 13: Réseau à une seule couche

I.9.2. Perceptron Multi Couches

Les problèmes rencontrés avec les réseaux à couche unique, ont poussé les chercheurs à développer d'autres réseaux, à savoir le réseau de Kohonen en 1981, le réseau d'Hopfield 1982 et le perceptron multicouche. Ce dernier avait un problème majeur qui été l'impossibilité de faire l'apprentissage des couches cachées jusqu'à 1985 avec l'introduction de l'algorithme de rétro-propagation.

Le perceptron Multi Couches PMC ou MLP (Multi Layer Perceptron) en anglais est une extension du perceptron monocouche, avec une ou plusieurs couches cachées entre l'entrée et la sortie. L'idée principale est de grouper des neurones dans une couche. En place ensuite bout à bout plusieurs couches et on connecte complètement les neurones de deux couches adjacentes (Figure I.14). Les entrées des neurones de la deuxième couche sont donc en fait les sorties des neurones de la première couche. Les neurones de la première couche sont reliés au monde extérieur et reçoivent tous le même vecteur d'entrée. Ils calculent alors leurs sorties qui sont transmises aux neurones de la deuxième couche, etc. Les sorties des neurones de la dernière couche forment la sortie du réseau.

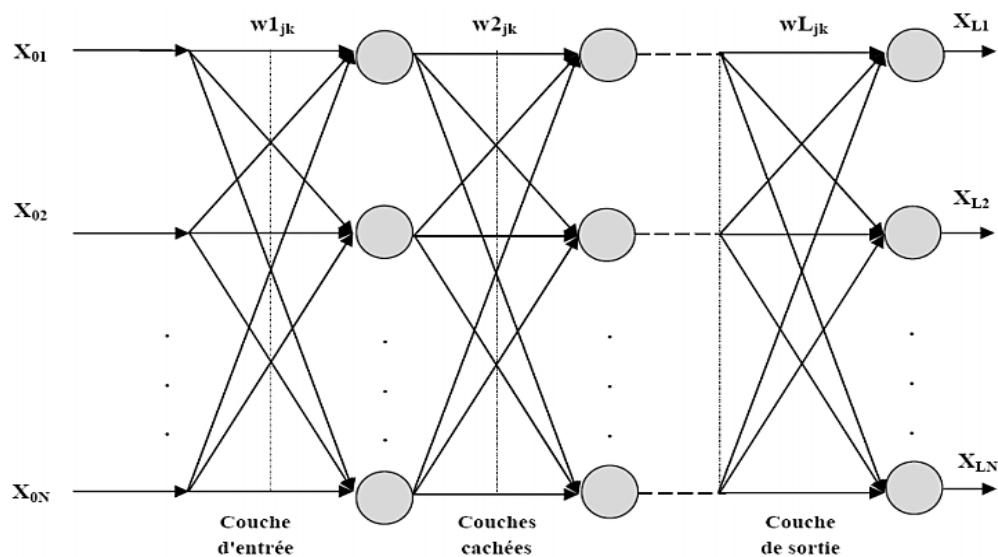


Figure I. 14 : Architecture du perceptron multicouche

Les réseaux multicouches sont beaucoup plus puissants que les réseaux simples à une seule couche. En utilisant deux couches (une couche cachée et une couche de sortie), à condition d'employer une fonction d'activation sigmoïde sur la couche cachée, on peut entraîner un réseau à produire une approximation de la plupart des fonctions, avec une précision arbitraire (cela peut cependant requérir un grand nombre de neurones sur la couche cachée). Sauf dans de rares cas, les réseaux de neurones artificiels exploitent deux ou trois couches [13].

I.9.3. Réseau à fonction radiale

Le réseau à fonction radiale (RBF) a la même structure que le Perceptron multicouches. Ils sont utilisés dans les mêmes genres de problèmes que les PMC à savoir, en classification et en prédiction. À l'exception de sa fonction d'activation qui est une fonction de type Gaussienne. Ce réseau, à cause de son architecture, utilise le plus souvent la règle d'apprentissage de correction d'erreur et la règle par apprentissage compétitif. Il peut avoir un apprentissage qui combine en même temps l'apprentissage supervisé et l'apprentissage non supervisé. Il est caractérisé par un apprentissage plus rapide et plus simple [14].

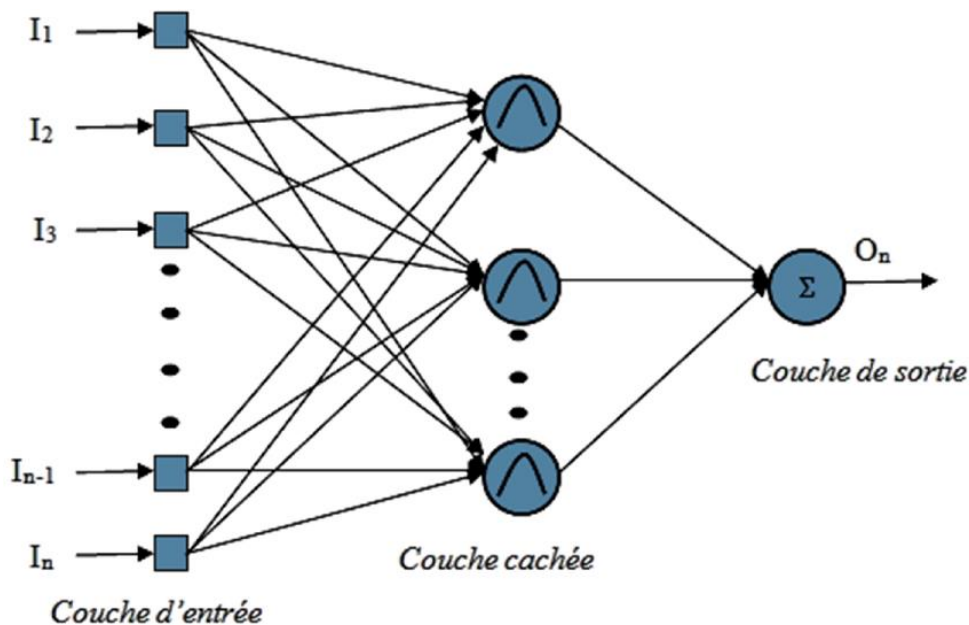


Figure I. 15: Structure d'un réseau de neurones de type RBF

I.10. Applications

Aujourd'hui, les réseaux de neurones artificiels ont de nombreuses applications dans des secteurs très variés [12]:

- **Traitement d'images** : reconnaissance de caractères et de signatures, compression d'images, reconnaissance de forme, cryptage, classification, etc.
- **Traitement du signal** : filtrage, classification, identification de source, traitement de la parole...etc.
- **Contrôle** : commande de processus, diagnostic, contrôle qualité, asservissement des robots, systèmes de guidage automatique des automobiles et des avions...etc.

- **Défense** : guidage des missiles, suivi de cible, reconnaissance du visage, radar, sonar, lidar, compression de données, suppression du bruit...etc.
- **Optimisation** : planification, allocation de ressource, gestion et finances, etc.
- **Simulation** : simulation du vol, simulation de boîte noire, prévision météorologique ...etc.

I.11. Conclusion

Les Réseaux de Neurones Artificiels sont considérés comme des approches très intéressantes dans le domaine de l'Intelligence Artificielle. Ils sont connus par leur puissance d'apprentissage et généralisation. Les Réseaux de Neurones Artificiels sont appliqués dans les reconnaissances des formes, comme dans l'approximation de fonctions. Dans ce chapitre, un aperçu général est donné sur la notion de Réseaux de Neurones Artificiels, en passant par la définition, la structure et le fonctionnement. Les différents types de réseaux sont présentés aussi avec le domaine d'applications.

Chapitre II : Description du système pendule inversé

II.1 Introduction

Dans le domaine scientifique, l'électronique 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 [15].

Dans ce chapitre, nous décrivons les caractéristiques générales du système pendule inversé et nous expliquons ses applications variées, par la suite nous élaborons son modèle mathématique en se basant sur le formalisme d'Euler-Lagrange. Enfin nous présentons ce système dans l'espace d'état ainsi que les résultats de simulation.

II.2 Qu'est-ce qu'un pendule ?

Souvenez-vous quand vous étiez petit et que vous essayiez 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 (le chariot) se translate dans une seule direction, alors que le doigt se déplace dans toutes les directions.

Un pendule inversé est un système dynamique multi variable non linéaire instable, il est considéré comme un très bon exemple pour les systèmes instables et non linéaires. Depuis 1950, les pendules inversés sont des plates-formes d'expérimentations classiques dans les laboratoires d'automatique Ils sont utilisés pour tester les performances des techniques de commande pour des constantes de temps très rapides. Ils ont été utilisés pour illustrer des idées de commande linéaire comme la stabilisation des systèmes instables. Vue leur nature non linéaire, les pendules sont aussi utilisés pour illustrer des idées émergentes du contrôle non linéaire [16].

Le pendule inversé que nous considérons est composé de deux éléments (figure II.1):

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

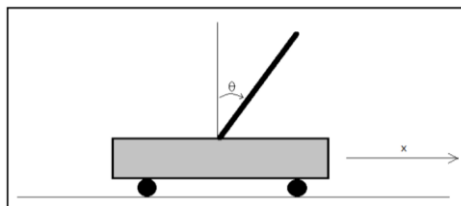


Figure II. 1 : Pendule inversé en coordonnées généralisées x et θ

II.3 Principe de fonctionnement

Son principe de fonctionnement est très simple en théorie : Après avoir ramené le pendule de sa position d'équilibre basse à la position verticale haute, il faut le maintenir dans cette position. Pour cela, quand le pendule penche vers la droite, le chariot doit le rattraper en effectuant un mouvement vers la droite, et inversement. Le problème se pose dans le réglage de l'intensité et la forme de réaction que doit subir le chariot en fonction de l'angle que le pendule fait avec la verticale. Ce système présente des caractéristiques intéressantes permettant d'illustrer quelques problèmes types, il s'agit d'un système SIMO (Single Input Multiple Output), instable décrit par un modèle non linéaire.

L'ensemble du chariot-pendule a deux degrés de liberté qui sont représentés 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 [16].

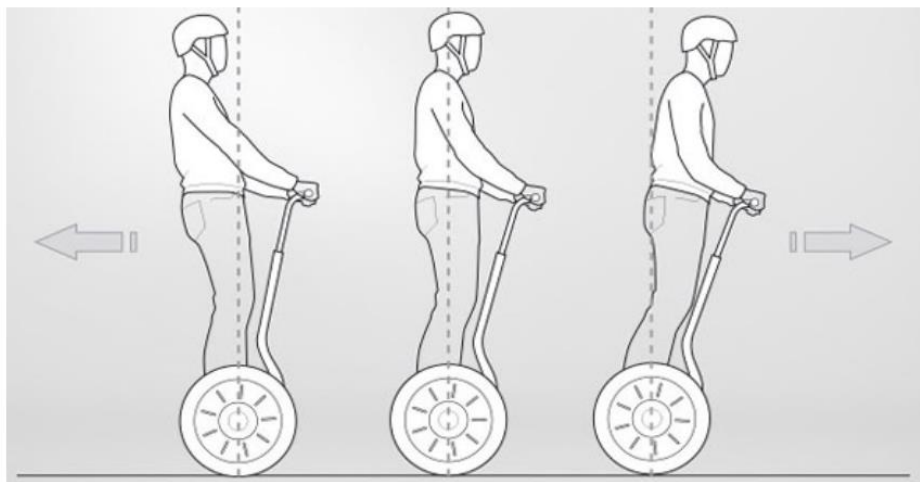


Figure II. 2 : Exemple sur l'équilibre à base du pendule inversé.

II.4. Intérêt de l'étude d'un pendule inversé

L'étude de pendule inversé est idéale pour les étudiants comme nous en électronique et en robotique qui font les laboratoires de recherches parce qu'ils font intervenir beaucoup de notions intéressantes pour eux: la programmation, l'électronique, l'automatisation et la mécanique [17].

On cite dans cette section quelques applications de ce concept dans le monde réel dans des domaines variées :

II.4.1. Le domaine de la 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 du double pendule inversé pour calculer l'ensemble des contraintes qui sont soumises à la prothèse, comme le montre la figure (figure.II.3) suivante

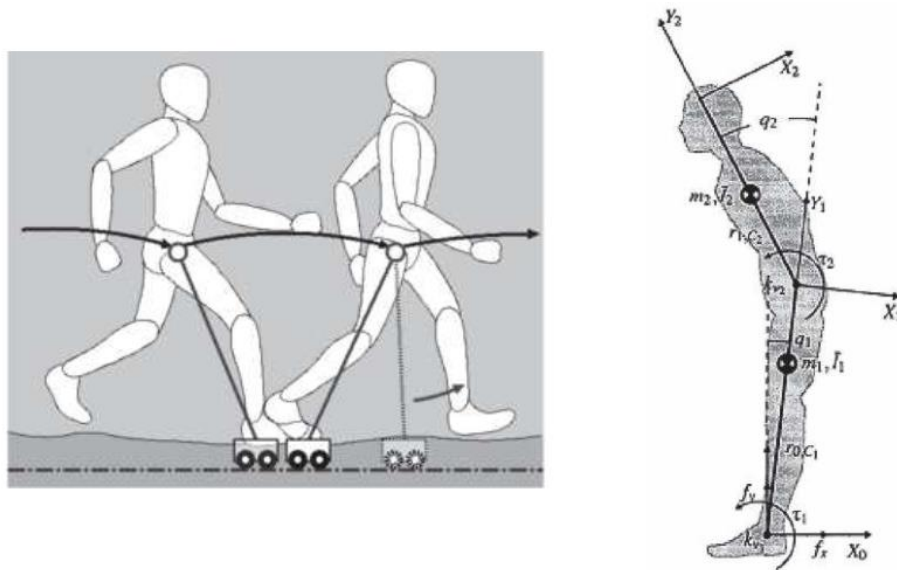


Figure II. 3 :Le corps de l'être humain vu comme un double pendule inversé

Le premier pendule est articulé à la cheville et représente les membres inférieurs considérés comme groupés. Le second pendule est articulé à la hanche et représente la partie supérieure du corps. On accélère en se penchant en avant et on ralentit en se penchant en arrière. Le système est le même que le pendule inversé [17].

II.4.2. Le domaine de Robotique

Dans le même ordre d'idée, un robot est vu comme un pendule inversé, et donc pour maintenir le robot en équilibre il suffit de stabiliser le pendule inversé correspondant à sa position verticale haute au cours de son déplacement et en présence d'autres perturbations. A l'occasion nous citons quelques exemples des robots qui sont déjà étudiés et conçus base de ce principe [17] :

- **Le SEGWAY** : Il est vu comme un pendule inversé simple (une seule tige) se déplaçant à l'aide de deux roues.



Figure II. 4: Le Segway

- **Le robot BIPED** : peut se présenter par deux doubles pendules inversés en série, dont les deux pendules sont couplés par une tige avec deux articulations.

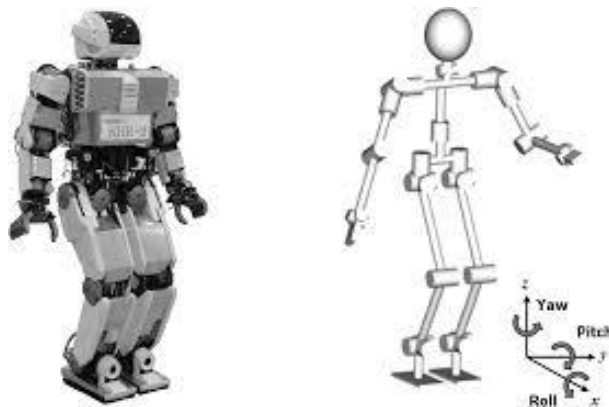


Figure II. 5: Le robot BIPED

II.4.3. Le domaine de l'aérospatiale

Dans ce domaine aussi, l'étude des pendules inversés a une grande importance, par exemple pour commander et stabiliser l'attitude du satellite, le lancement des fusées, etc [17].

- **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.

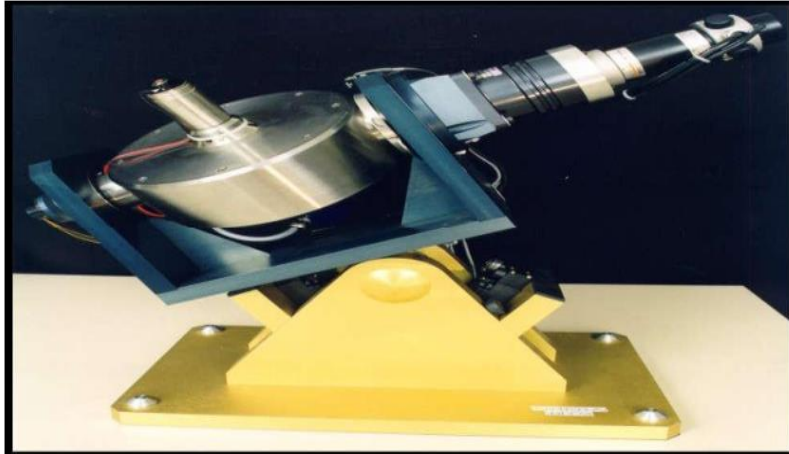


Figure II. 6 : Pendule gyroscopique inversé

- **Pour le lancement des fusées**

Lors de son lancement, une fusée est vue comme un pendule inversé simple, donc il faut maintenir sa position verticale par rapport à la terre, et pour cela, des moteurs à propulsion sont installés dans les deux côtés de la fusée (gauche et droite).

II.4.4. Dans La Recherche

Vu sa complexité, le pendule inversé est considéré comme l'un des meilleurs systèmes pour tester des lois de commande moderne [17].

II.5. Modélisation du pendule inversé

Dans cette partie, nous allons établir les équations du mouvement du pendule inversé, en se basant sur le schéma de la figure II.7

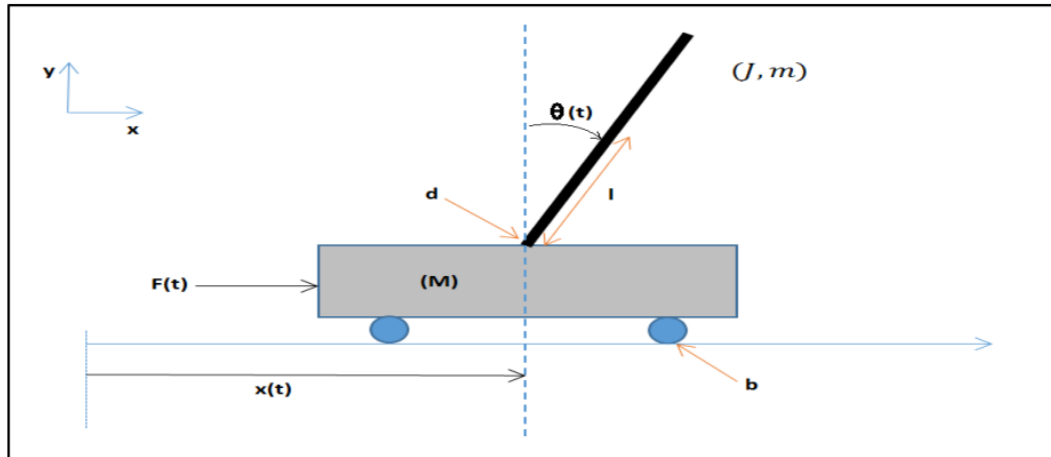


Figure II. 7 : Modèle d'un pendule inversé

Pour toute le reste du chapitre, on définit les paramètres suivants

Tableau II. 1 : Paramètres de l'ensemble chariot et pendule

Parametre	Description
M	Masse du chariot (Kg)
m	Masse du pendule (Kg)
l	Demi-longueur du pendule (m)
F(t)	Forces exercées sur le chariot (N)
x(t)	Position du chariot (m)
$\theta(t)$	L'angle du pendule (rad)
d	Coefficient de frottements du pendule (Nms/rad)
b	Coefficient de frottements du déplacement du chariot (Ns.m)
g	Intensité de la pesanteur (m.s ²)
J	Inertie du pendule en son centre de masse

Il existe deux méthodes de détermination des équations du mouvement d'un pendule inversé: celle de la loi fondamentale de la dynamique (LFD) de Newton qui est basée sur le concept de force, et celle de formalisme d'Euler-Lagrange qui est basée sur le principe de la conservation d'énergie mécanique [16]

Dans ce travail on s'intéresse au formalisme d'Euler-Lagrange dans lequel le lagrangien (L) est défini comme une différence entre l'énergie cinétique (E_c) et l'énergie potentielle (E_p) du système :

$$L = E_c - E_p \quad (\text{II.1})$$

II.5.1. Energie cinétique du système

Le système en mouvement compose du chariot qui se déplace linéairement sur les rails et le pendule qui se balance sur son axe de rotation (les roues du chariot sont négligeables)

L'énergie cinétique du chariot est donnée par :

$$E_{cc} = \frac{1}{2} M \dot{x}^2 \quad (\text{II.2})$$

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

$$E_{cp} = \frac{1}{2} m v_c^2 + \frac{1}{2} j \dot{\theta}^2 \quad (\text{II.3})$$

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 :

$$\vec{r}_c = (x + l \sin \theta) \vec{i} + l \cos \theta \vec{j} \quad (\text{II.4})$$

\vec{i} et \vec{j} sont respectivement les vecteurs unitaires du repère x et y

la vitesse du centre de gravité est exprimée par :

$$v_c = \frac{d\vec{r}_c}{dt} = (\dot{x} + l \dot{\theta} \cos \theta) \vec{i} - (l \dot{\theta} \sin \theta) \vec{j} \quad (\text{II.5})$$

En substituant l'équation (II.5) dans l'équation (II.3), l'expression de l'énergie cinétique du pendule devient :

$$E_{cp} = \frac{1}{2} m (\dot{x}^2 + 2 \dot{x} l \dot{\theta} \cos \theta + l^2 \dot{\theta}^2 \cos^2 \theta + l^2 \dot{\theta}^2 \sin^2 \theta) + \frac{1}{2} j \dot{\theta}^2 \quad (\text{II.6})$$

Après la simplification du terme $(\cos^2 \theta + \sin^2 \theta) = 1$, l'expression de l'énergie cinétique s'écrit:

$$E_{cp} = \frac{1}{2} m (\dot{x}^2 + 2 \dot{x} l \dot{\theta} \cos \theta + l^2 \dot{\theta}^2) + \frac{1}{2} j \dot{\theta}^2 \quad (\text{II.7})$$

Finalement, l'énergie cinétique totale chariot-pendule est donc :

$$E_c = E_{cc} + E_{cp} = \frac{1}{2} M\dot{x}^2 + \frac{1}{2} m(\dot{x}^2 + 2\dot{x}l\dot{\theta} \cos \theta + l^2\dot{\theta}^2) + \frac{1}{2} j\dot{\theta}^2 \quad (\text{II.8})$$

II.5.2. L'énergie potentielle du système

Le chariot étant en mouvement sur un rail horizontal, seul le pendule en mouvement possède une énergie potentielle, celle s'est donnée par :

$$E_p = mgl.\cos \theta \quad (\text{II.9})$$

II.5.3. Equations d'Euler-Lagrange

Quand on établit les expressions de l'énergie cinétique E_c de l'ensemble chariot-pendule (équation (II.8)) et l'énergie potentielle du système E_p (équations (II.9)), on utilise l'équation générale d'Euler-Lagrange pour déterminer les équations du mouvement de l'ensemble chariot-pendule :

$$L = \frac{1}{2} M\dot{x}^2 + \frac{1}{2} m(\dot{x}^2 + 2\dot{x}l\dot{\theta} \cos \theta + l^2\dot{\theta}^2) + \frac{1}{2} j\dot{\theta}^2 - mgl.\cos \theta \quad (\text{II.10})$$

L'équation générale d'Euler-Lagrange est donnée par :

$$\frac{d}{dt} \left(\frac{\partial L}{\partial \dot{\varepsilon}_j} \right) - \frac{\partial L}{\partial \varepsilon_j} + \frac{\partial D_f}{\partial \dot{\varepsilon}_j} = F_j \quad (\text{II.11})$$

$\varepsilon(t)$: Désigne les degrés de liberté du pendule

D_f : Désigne l'énergie dissipée par frottement

F_j : La force généralisée

L : Représente le lagrangien (équation(II.1))

En considérant que le système ait deux degrés de liberté, x pour le déplacement horizontal du chariot et θ pour la rotation du pendule, la modélisation du système selon la loi générale l'Euler-Lagrange est donnée par :

$$\frac{d}{dt} \left(\frac{\partial L}{\partial \dot{\varepsilon}_j} \right) - \frac{\partial L}{\partial \varepsilon_j} = F_j \quad (\text{II.12})$$

Pour le degré de liberté $\varepsilon(t)=x(t)$:

$$\frac{d}{dt} \left(\frac{\partial L}{\partial \dot{x}} \right) - \frac{\partial L}{\partial x} = F - b\dot{x} \quad (\text{II.13})$$

où : F représente la force exercée sur le chariot

La dérivée partielle de lagrangien suivant \dot{x} s'écrit :

$$\frac{d}{dt}(M\dot{x} + m\dot{x} + ml \cos \theta \dot{\theta}) - 0 = F - b\dot{x} \quad (\text{II.14})$$

La première équation de Lagrange s'écrit alors :

$$(M + m)\ddot{x} + ml\ddot{\theta} \cos \theta - ml\dot{\theta}^2 \sin \theta = F - b\dot{x} \quad (\text{II.15})$$

Pour le degré de liberté $\varepsilon(t) = \theta(t)$:

$$\frac{d}{dt}\left(\frac{\partial L}{\partial \dot{\theta}}\right) - \frac{\partial L}{\partial \theta} = -d\dot{\theta} \quad (\text{II.16})$$

La dérivée de lagrangien suivant $\dot{\theta}$ et \mathbf{t} s'écrit :

$$\frac{d}{dt}(ml\dot{x} \cos \theta + ml^2\dot{\theta} + j\dot{\theta}) - (-ml\dot{x} \sin \theta \dot{\theta} + mgl \sin \theta) = -d\dot{\theta} \quad (\text{II.17})$$

La deuxième équation de Lagrange est :

$$(ml^2 + j)\ddot{\theta} + ml\ddot{x} \cos \theta - ml\dot{x} \dot{\theta} \sin \theta + ml\dot{x} \dot{\theta} \sin \theta - mgl \sin \theta = d\dot{\theta} \quad (\text{II.18})$$

Le modèle de connaissance de l'ensemble chariot-pendule est donné par le système d'équations :

$$\begin{cases} h\ddot{x} + b\dot{x} + ml\ddot{\theta} \cos \theta - ml\dot{\theta}^2 \sin \theta = F \\ ml\ddot{x} \cos \theta + N\ddot{\theta} + d\dot{\theta} - mgl \sin \theta = 0 \end{cases} \quad (\text{II.19})$$

où :

$$\begin{cases} h = M + m \\ N = ml^2 + j \end{cases}$$

Le système d'équations (II.19) montre la dépendance qui existe entre l'accélération du chariot \ddot{x} et l'accélération angulaire du pendule $\ddot{\theta}$. Ainsi, lorsque $F=0$ (pas de force extérieure), si on déplace le pendule de sa position d'équilibre, il ne sera soumis qu'à sa propre inertie, il se mettrait alors à osciller et puisque le pendule est fixé sur le chariot, ce dernier commencerait également à se mouvoir.

II.5.4. Modèle d'état de l'ensemble chariot-pendule :

Afin d'obtenir le modèle d'état de l'ensemble chariot-pendule, on utilise le modèle (II.19), le vecteur d'état est :

$$X = [x_1 \ x_2 \ x_3 \ x_4]^T = [x \ \dot{x} \ \theta \ \dot{\theta}]^T \quad (\text{II.20})$$

Après quelques manipulations mathématiques [16], on obtient le modèle non linéaire (II.21) où la grandeur de commande est la force F :

$$\left\{ \begin{array}{l} \dot{x}_1 = x_2 \\ \dot{x}_2 = \frac{-bN}{hN - m^2l^2 \cos^2 x_3} x_2 - \frac{m^2l^2 g}{hN - m^2l^2 \cos^2 x_3} \cos x_3 \sin x_3 + \frac{mld \cos x_3}{hN - m^2l^2 \cos^2 x_3} x_4 \\ \quad + \frac{mlN \sin x_3}{hN - m^2l^2 \cos^2 x_3} x_4^2 + \frac{FN}{hN - m^2l^2 \cos^2 x_3} \\ \dot{x}_3 = x_4 \\ \dot{x}_4 = \frac{mgl}{N} \sin x_3 - \frac{d}{N} x_4 + \frac{mlb \cos x_3}{hN - m^2l^2 \cos^2 x_3} x_2 + \frac{m^3l^3 g \cos^2 x_3 \sin x_3}{N(hN - m^2l^2 \cos^2 x_3)} - \frac{m^2l^2 d \cos^2 x_3}{N(hN - m^2l^2 \cos^2 x_3)} x_4 \\ \quad - \frac{m^2l^2 \cos x_3 \sin x_3}{hN - m^2l^2 \cos^2 x_3} x_4^2 - \frac{mlF}{hN - m^2l^2 \cos^2 x_3} \cos x_3 \\ y_1 = x_1 \\ y_2 = x_3 \end{array} \right. \quad (II.21)$$

II.6. Elaboration du modèle avec SIMULINK

Pour élaborer le modèle Simulink du système pendule inversé, on commence par construire un fichier mfile dans lequel on introduit les valeurs des paramètres du système [12] :

```
M=2.3;
m=0.2;
l=0.3;
d=0.005;
b=0.00005;
g=9.81;
J=0.0099;
h=M+m;
N=m*l*l+J;
```

D'après le modèle obtenu (Equation II-21) on va pouvoir le traduire sous forme de schémas block avec SIMULINK comme suit :

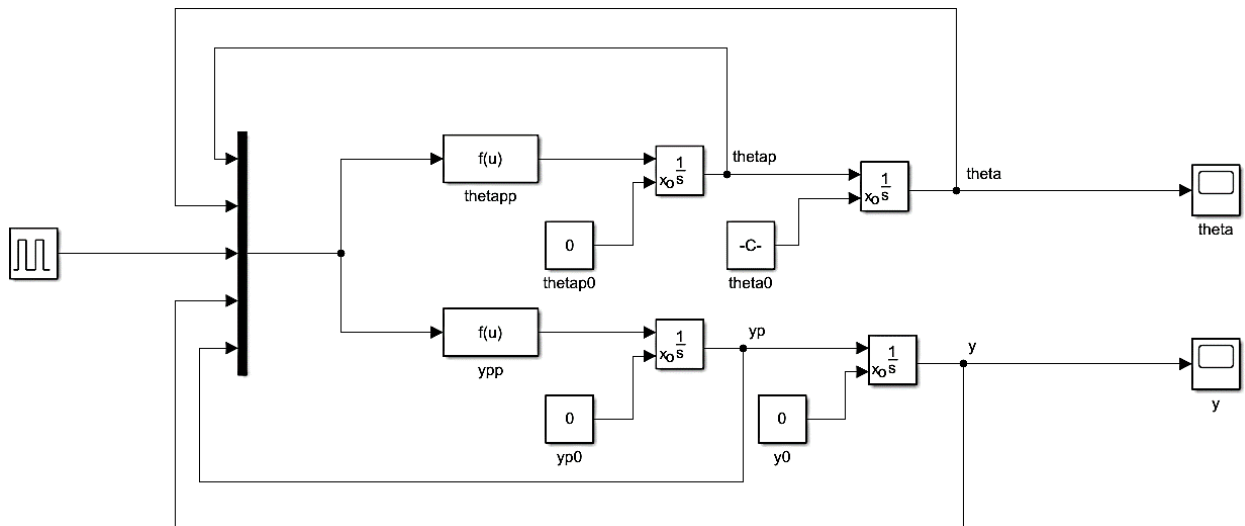


Figure II. 8 : Schéma Simulink du pendule inversé

II.7. Résultat de simulation

On représente dans les figures ci-dessous les résultats de simulation du modèle du pendule inversé pour différents cas de la position du pendule :

Une force sous forme d’impulsion de durée $T=0.1$ sec et d’amplitude de 1 N est appliquée.

- **Premier cas (pendule simple) :**

Position du chariot et du pendule : $[x \dot{x} \theta \dot{\theta}] = [0,0,0,0]$

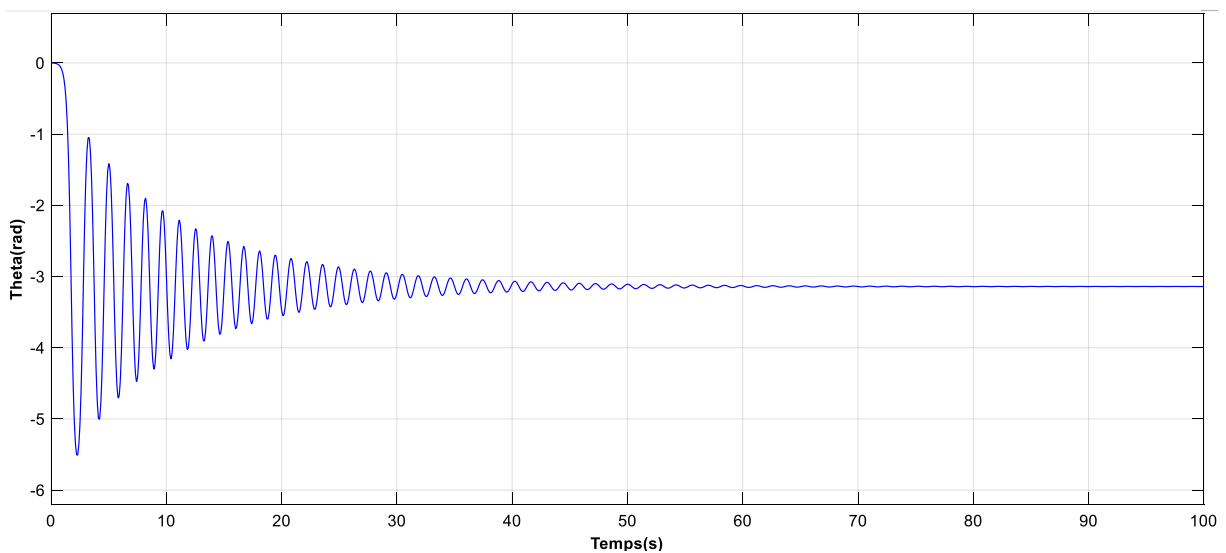


Figure II. 9 : Angle du pendule pour $[x \dot{x} \theta \dot{\theta}] = [0,0,0,0]$

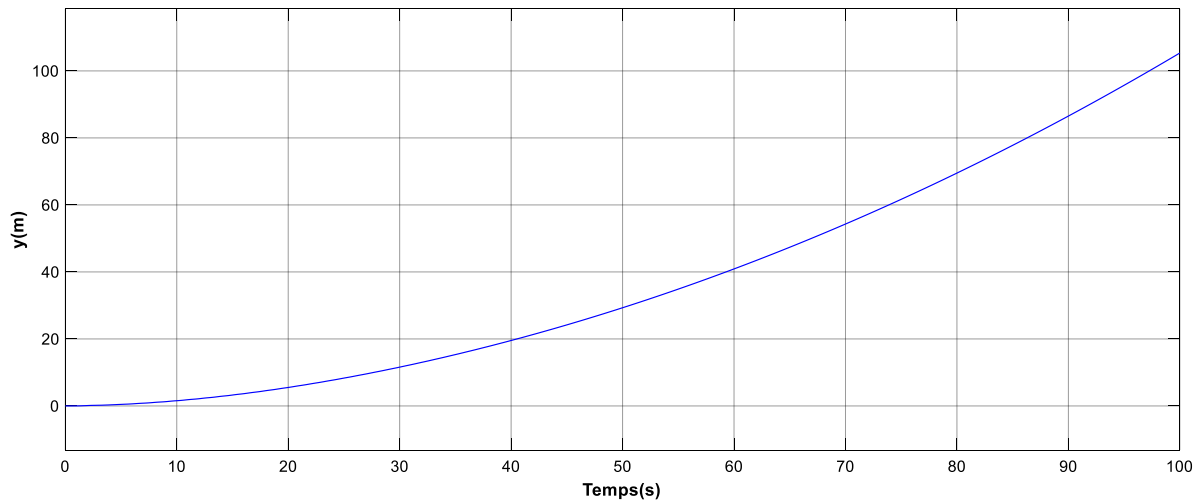


Figure II. 10 : Position du chariot pour $[x \dot{x} \theta \dot{\theta}] = [0,0,0,0]$

• Deuxieme cas (pendule inversé) :

Position du chariot et du pendule : $[x \dot{x} \theta \dot{\theta}] = [0,0,\pi,0]$

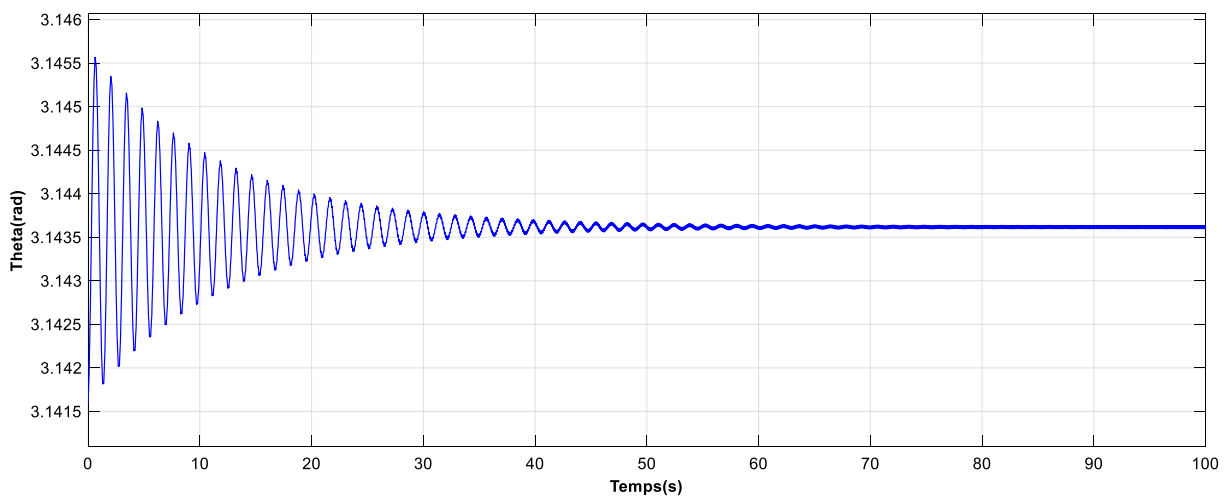


Figure II. 11 : Angle du pendule pour $[x \dot{x} \theta \dot{\theta}] = [0,0,\pi,0]$

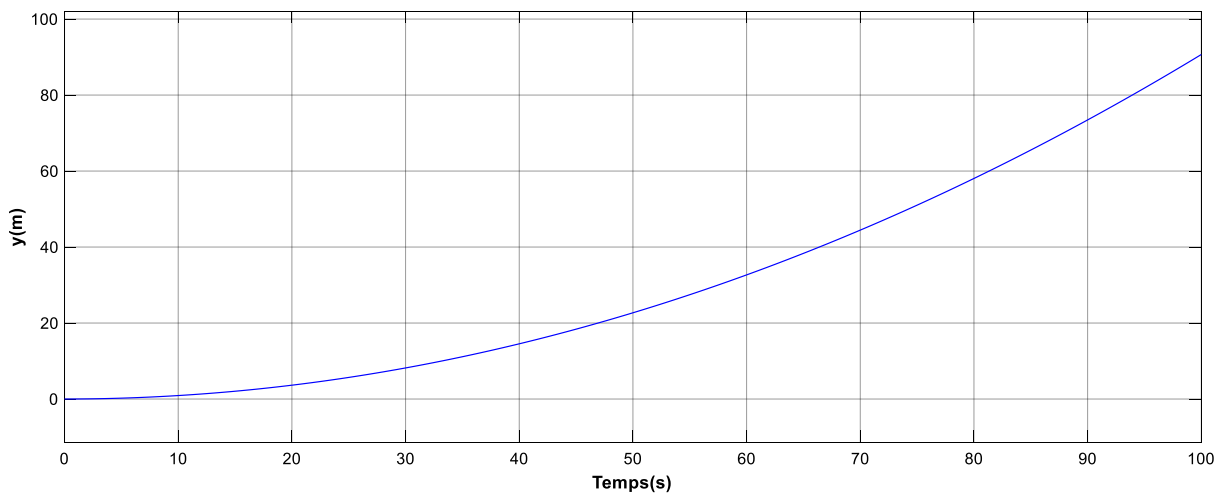


Figure II. 12 : Position du chariot pour $[x \dot{x} \theta \dot{\theta}] = [0,0,\pi,0]$

Ces résultats illustrent la complexité du modèle. Les figures II.9 et II.10 montrent que lorsque le pendule est initialement en position instable ($\theta=0$) il se stabilise après un régime transitoire dans sa position d'équilibre stable ($\theta=\pi$). Les figures II.11 et II.12 montrent que lorsque le pendule est initialement dans sa position d'équilibre stable ($\theta=\pi$), lorsqu'il est écarté grâce à la force F , il y revient après le régime transitoire.

II.8 Conclusion

Ce chapitre a été consacré à la description et la modélisation du pendule inversé puis au développement du modèle pendule-chariot sur Simulink. Vu l'instabilité du système, une commande est impérative, pour tenter de stabiliser le système au tour du point $\theta = 0$ qui traduit l'inversement du pendule, et le développement de cette commande fera l'objet du troisième chapitre.

Chapitre III : Contrôle du pendule inversé par un réseau de neurones artificiels

III.1 Introduction

Développer un contrôleur pour le pendule inversé est pratiquement difficile. Cependant, il n'existe pas, jusqu'à présent, d'outils mathématiques très généraux pour la modélisation ou la commande des systèmes non linéaires. Les propriétés des approximateurs universels et d'apprentissage encouragent l'étude des capacités des réseaux de neurones à reproduire le comportement dynamique des systèmes non linéaires [14].

Dans ce chapitre on va concevoir un réseau de neurones qui imite un contrôleur dans le but de contrôler le système pendule inversé qu'on a développé dans le chapitre précédent, puis on va présenter dans les différents résultats de simulation obtenus.

III.2 Conception d'un contrôleur pour le pendule inversé

Un contrôleur de type " feedback linearization controller " a été développé pour contrôler le système non linéaire de pendule inversé. Le principe de ce contrôleur est d'annuler les non-linéarités dans le système de sorte que le système en boucle fermée est plus linéaire [18].

III.2.1 Equations du contrôleur

Les équations suivantes sont une 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 entrées dans l'équation principale. L'équation principale (équation III.5) calcule la force requise, U pour garder le pendule stable [19].

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

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

$$f_1 = m(l \sin \theta \cdot \dot{\theta}^2 - \frac{3}{8} g \sin 2\theta) - f\ddot{x} \quad (\text{III.3})$$

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

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

III.2.2 Élaboration du contrôleur avec SIMULINK

Pour les simulations, M, m, l et g sont mises aux valeurs du modèle de pendule. Les valeurs numériques suivantes sont utilisées : $k_1=25$, $k_2=10$, $C_1=1$, $C_2=2,6$. Aussi $x_d=0$ mètres et $\theta_d=0$ rad, qui sont la position désirée du chariot et l'angle du pendule respectivement. Un modèle SIMULINK de la loi de contrôle ci-dessus a été développé et montré dans la figure III.1.

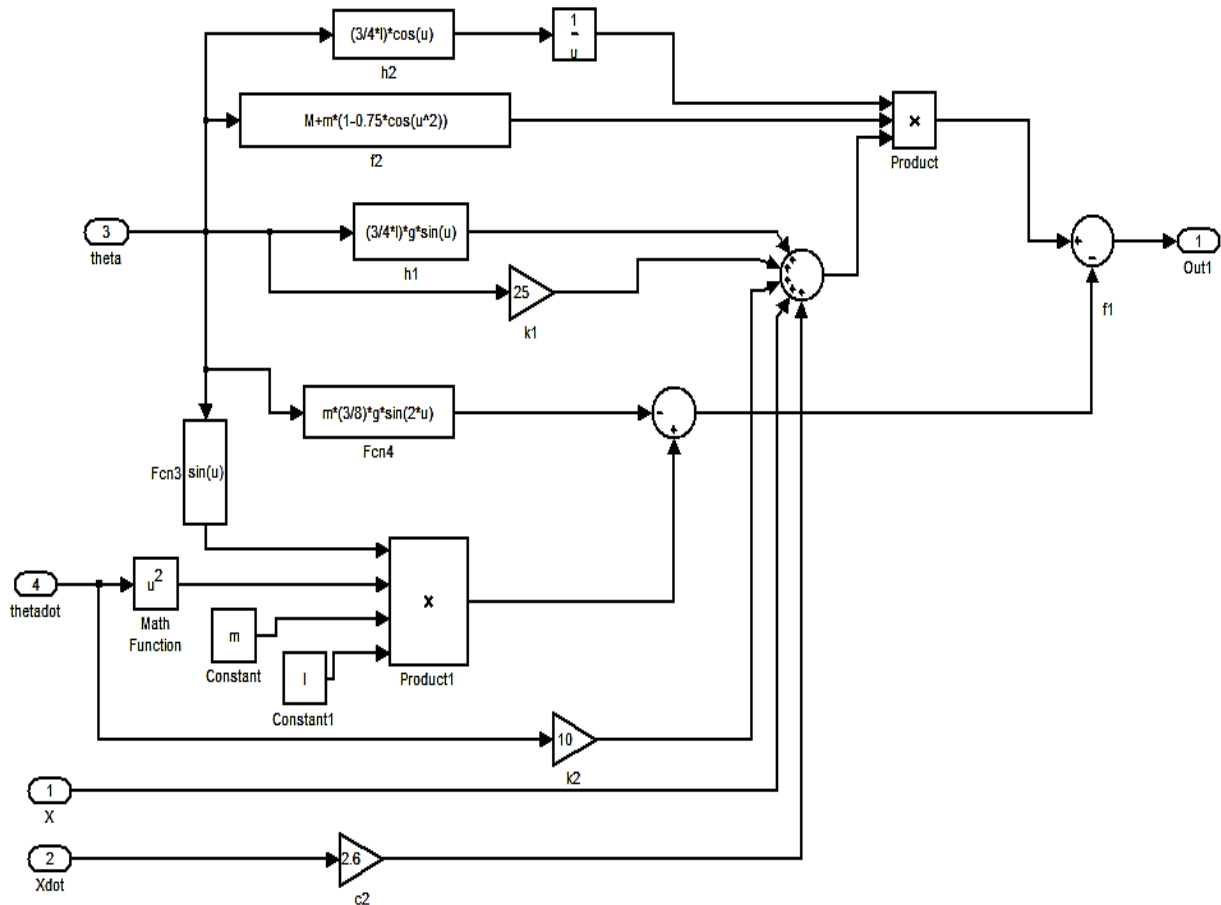


Figure III. 1 : Le model Simulink de la loi de control non lineaire

Les entrées de ce contrôleur sont les 4 états de sortie du modèle de pendule non linéaire. La loi de contrôle calcule la grandeur et la force correctes pour maintenir la stabilité du pendule.

III.3 Stabilisation du pendule inversé par le contrôleur

Pour stabiliser le pendule inversé autour du point d'équilibre, on doit le contrôler d'abord avec la loi de commande.

Le schéma Simulink ci-dessous montre le modèle Simulink avec le contrôleur. Le bruit d'entrée « Band-Limited White Noise » est un signal d'excitation qui représente une bande de fréquences riche. Il est utilisé pour obtenir une réponse entrée-sortie unique. Les quatre signaux d'entrée (y, ydot, theta, thetadot) et la sortie cible (out) sont exportés vers « Workspace » l'espace de travail Matlab.

Les conditions initiales du pendule inversé sont données par : $[x \dot{x} \theta \dot{\theta}] = [0, 0, 0.1, 0]$

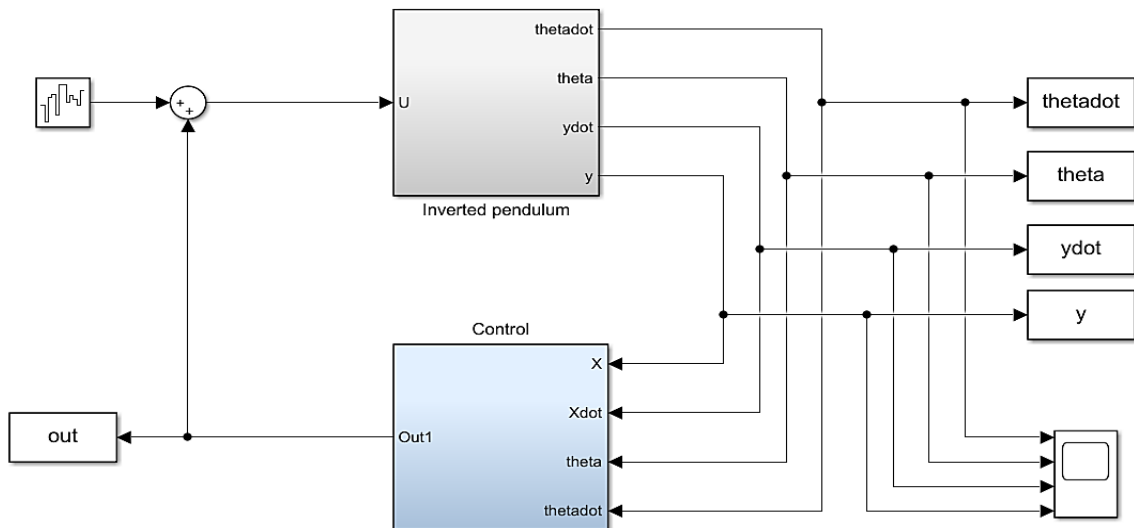


Figure III. 2 : Schéma de la commande du pendule inversé

● **Résultat de simulation :**

Les résultats de simulations obtenues, en utilisant le contrôleur pour contrôler les quatre grandeurs du système pendule inversé sont donnés par la figure III.5

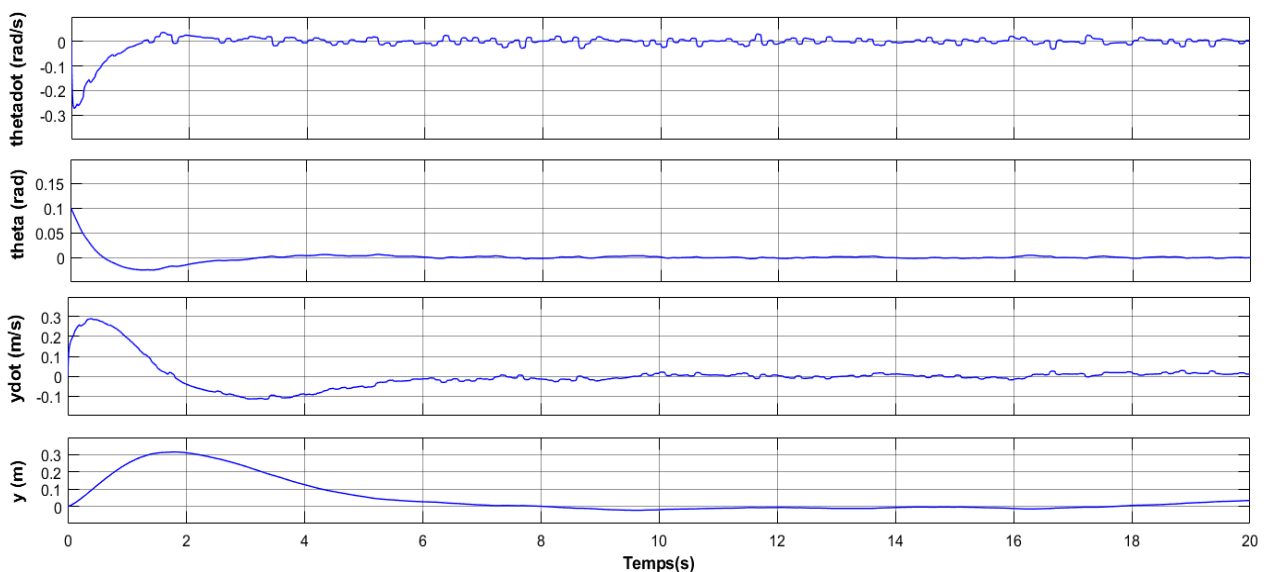


Figure III. 3 : Résultat de simulation de stabilisation du pendule inversé par le contrôleur

La figure III.5 montre que le système pendule inversé se stabilise au bout de 6 s tout en respectant les contraintes imposées. En effet le déplacement du chariot ne dépasse pas 0.3 m.

III.4 Stabilisation du pendule inversé par un réseau de neurones

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.4).

Concernant l'apprentissage de ce réseau, on applique deux algorithmes d'apprentissage qui sont l'algorithme de Levenberg-Marquardt et la méthode de régularisation Bayésienne.

III.4.1. Apprentissage avec l'algorithme Levenberg-Marquardt

Un réseau neuronal multicouche à propagation avant « feed-forward » (FF) construit pour imiter le contrôleur, un code Matlab a été développé pour entraîner le réseau neuronal.

Dans ce code 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 le nombre de couches, le nombre de neurones dans les couches cachées et les fonctions d'activation utilisées. Ce réseau de neurones comporte 10 neurones dans la couche cachée et 1 neurone dans la couche de sortie, et la fonction d'activation dans la couche cachée est la sigmoïde bipolaire et linéaire pour la couche de sortie (Figure III.6)

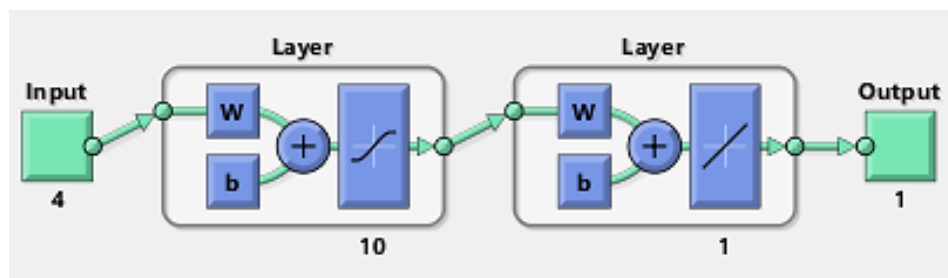


Figure III. 4 : architecture du réseau

Pour la phase d'apprentissage on a choisi la fonction d'apprentissage Levenberg-Marquardt « trainlm » parmi plusieurs fonctions, car elle est la meilleure, du point de vue convergence, rapidité et précision. Ceci est la configuration standard des fonctions d'activation dans PMC.

Le nombre d'itérations pour cet exemple est défini par défaut sur 1000. Pendant l'apprentissage, le vecteur d'entrée sera transmis au réseau neuronal et les poids seront ajustés 1000 fois.

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.7)

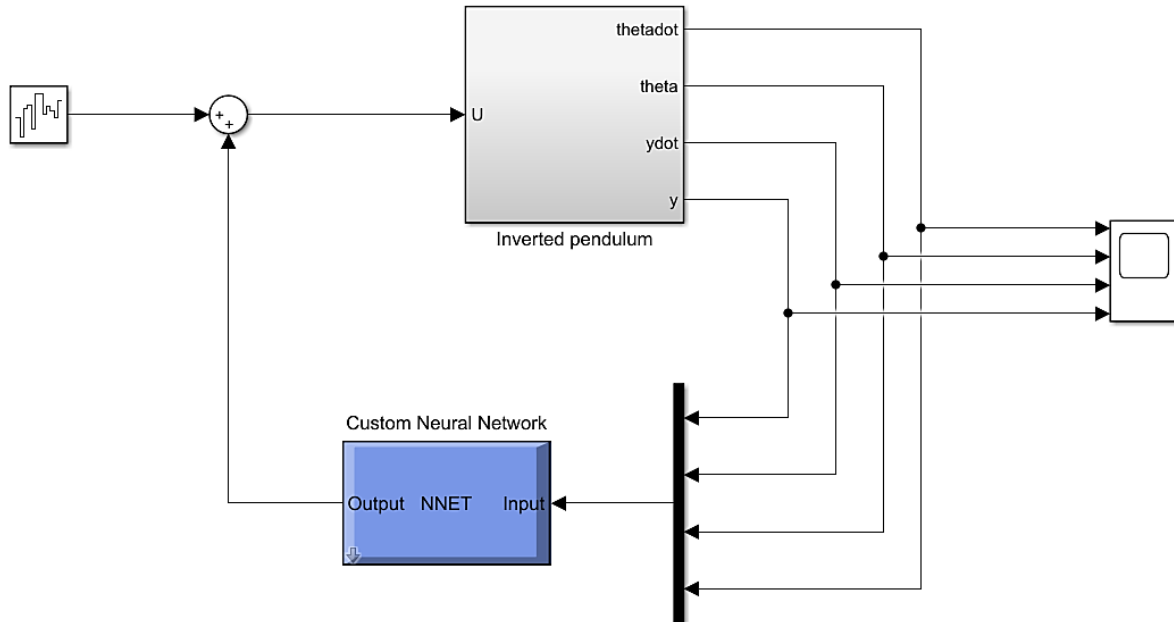


Figure III. 7 : Commande du pendule inversé par le RNA avec l'algorithme LM

• **Résultat de simulation :**

Les résultats de simulations obtenues, en utilisant le réseau de neurone avec l'algorithme d'apprentissage Levenberg-Marquardt, sont donnés par la figure suivante :

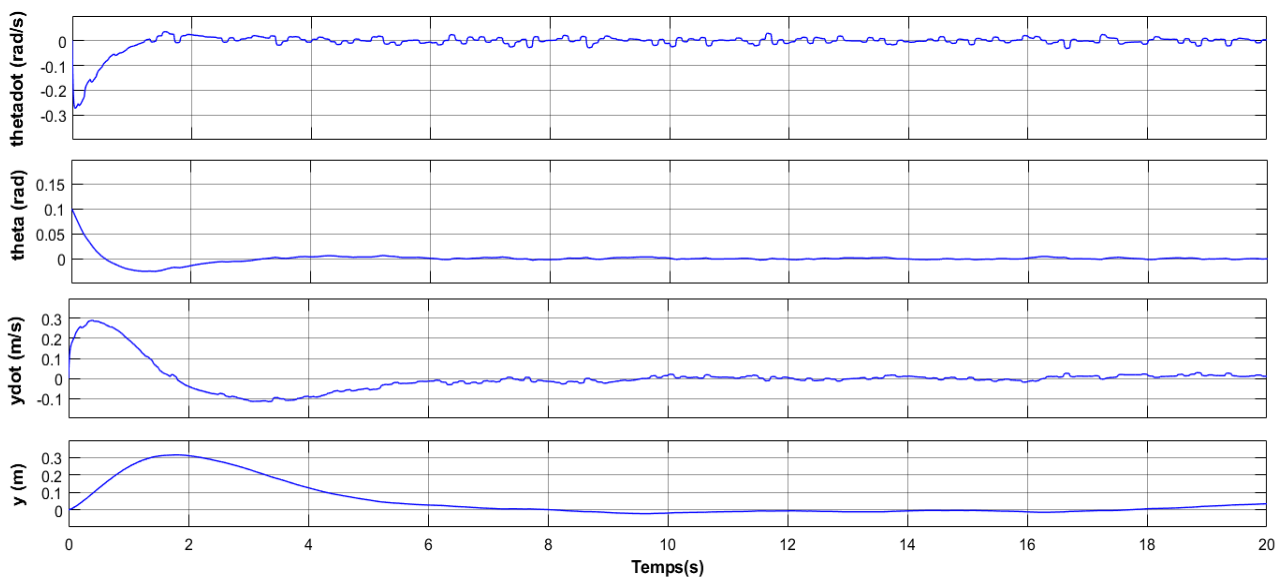


Figure III. 8 : Stabilisation du pendule inversé par le RNA avec l'algorithme LM

La figure III.8 représente l'état des quatre grandeurs du système pendule inversé en utilisant le réseau de neurones avec l'algorithme LM, et montre que le système se stabilise au bout de 6 s tout en respectant les contraintes imposées. En effet le déplacement du chariot ne dépasse pas 0.3 m.

La figure III.9 représente l'outil d'apprentissage du réseau de neurones où le temps d'apprentissage écoulé et le MSE sont indiqués.

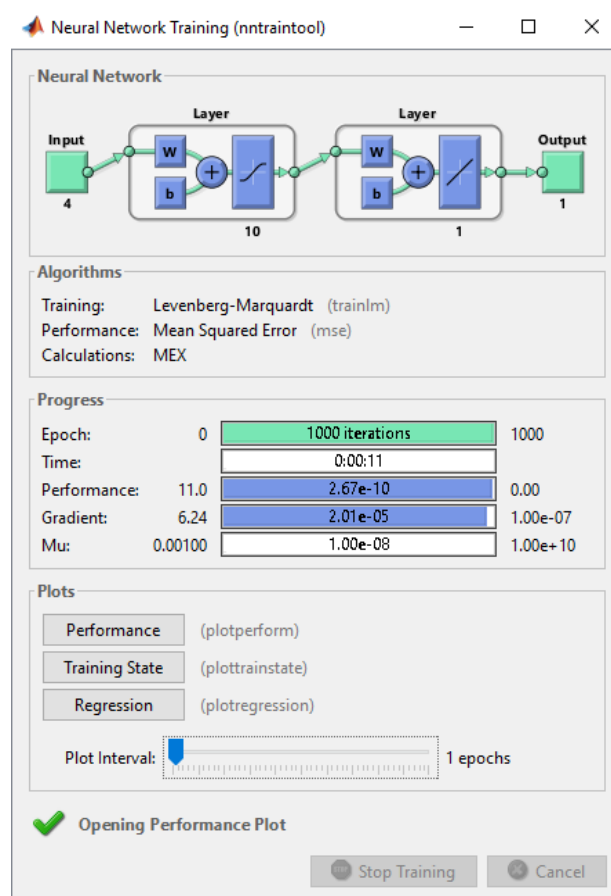


Figure III. 9 : l'outil d'apprentissage du RNA par l'algorithme LM

La figure III.10 représente le graph de l'erreur quadratique MSE d'apprentissage du réseau de neurones artificiel

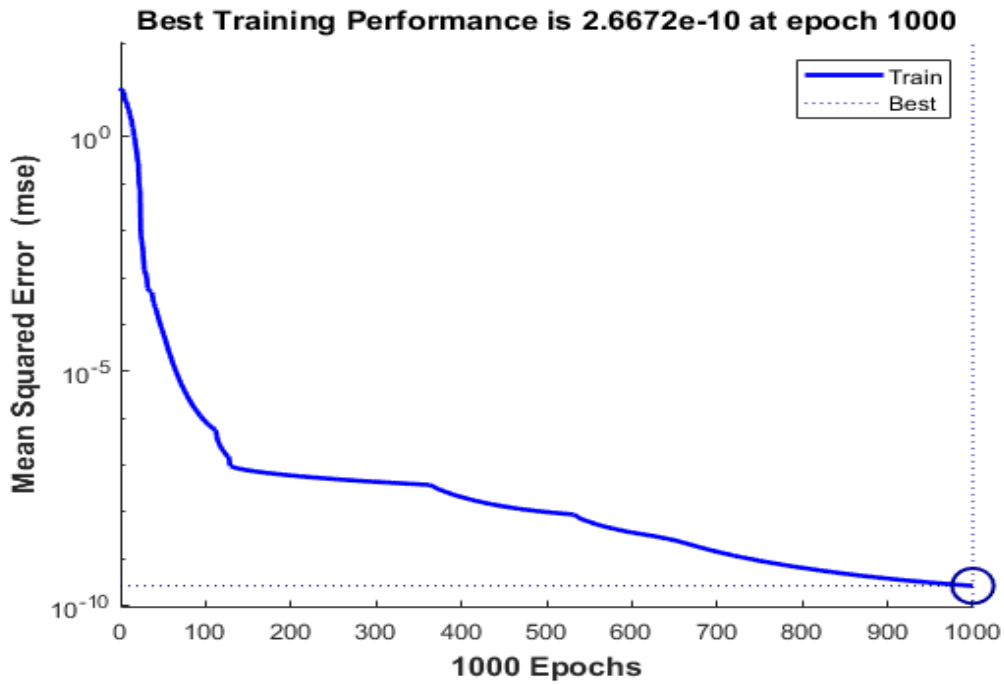


Figure III. 10 : MSE d'apprentissage du RNA par l'algorithme LM.

La figure III.12 représente le schéma qui permet de calculer l'erreur MSE de la commande entre le réseau de neurones et le contrôleur

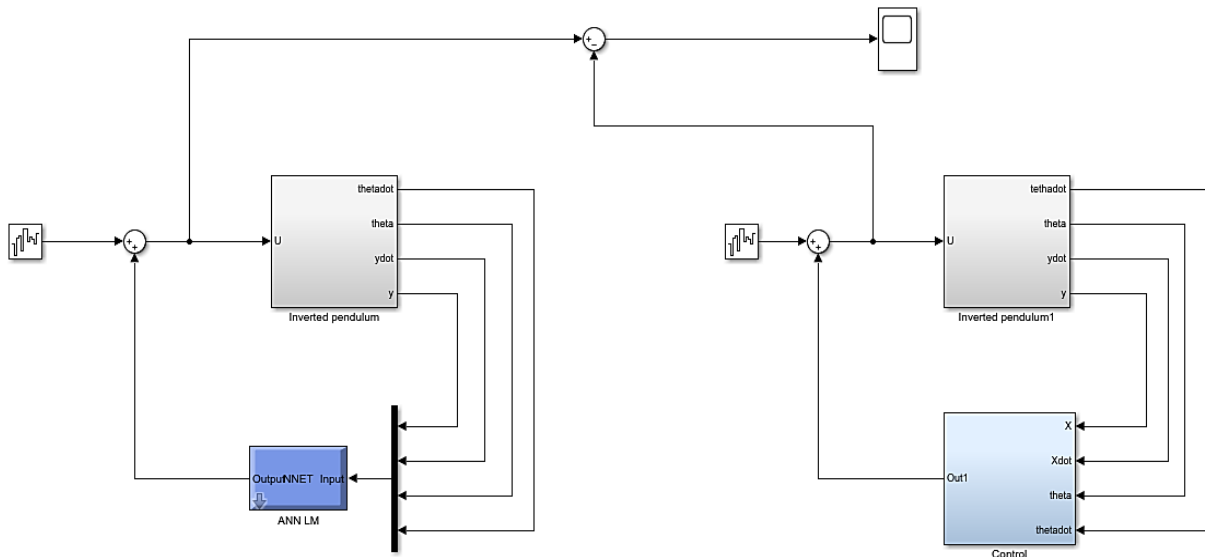


Figure III. 5 : Schéma pour calculer le MSE entre le neuro-contrôleur et le contrôleur

La sortie de la simulation est donnée par la figure suivante :

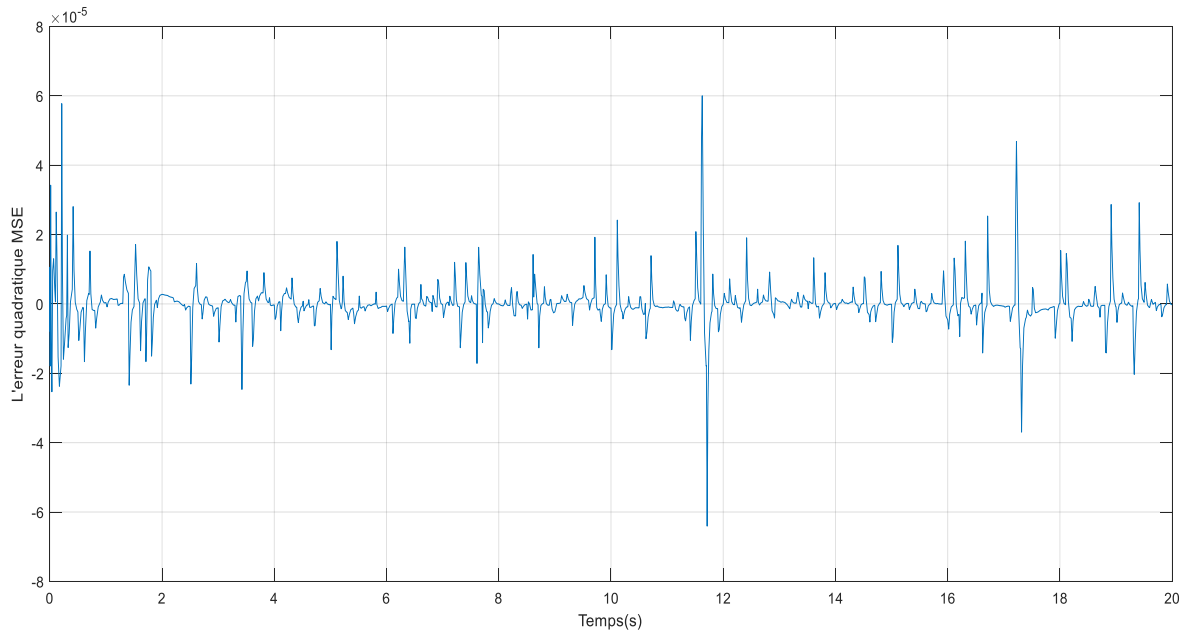


Figure III. 6 : l'erreur MSE de la commande par RNA et le contrôleur

Pour calculer le MSE entre les sorties du schéma avec le neuro-contrôleur et celles avec le contrôleur, on a exporté l'erreur entre eux (nommé `lm`) à partir de paramètre du scope vers le « Workspace » de Matlab puis on a utilisé la fonction Matlab `mse(lm(:,2))` pour calculer le MSE

Tableau III. 1: MSE entre le contrôleur et le RNA avec `trainlm`

Grandeur	MSE avec l'algorithme LM
Commande	$5.3490 * 10^{-11}$
thetadot	$4.3265 * 10^{-14}$
theta	$6.7255 * 10^{-16}$
Ydot	$5.3201 * 10^{-14}$
y	$1.2348 * 10^{-13}$

III.4.2. Apprentissage avec la méthode de régularisation Bayésienne (RB)

Dans cette partie, on va utiliser la même structure du réseau mais cette fois l'apprentissage sera avec la méthode de régulation Bayésienne.

Le réseau de neurones utilise la règle d'apprentissage de régularisation Bayésienne « trainbr » pour mettre à jour les poids. Le nombre d'itérations pour cet exemple est défini par défaut sur 1000. Pendant l'apprentissage, le vecteur d'entrée sera transmis au réseau neuronal et les poids seront ajustés 1000 fois.

Après la phase d'apprentissage un bloc de réseau de neurones sera généré et mise en place pour contrôler le pendule inversé (Figure III.14)

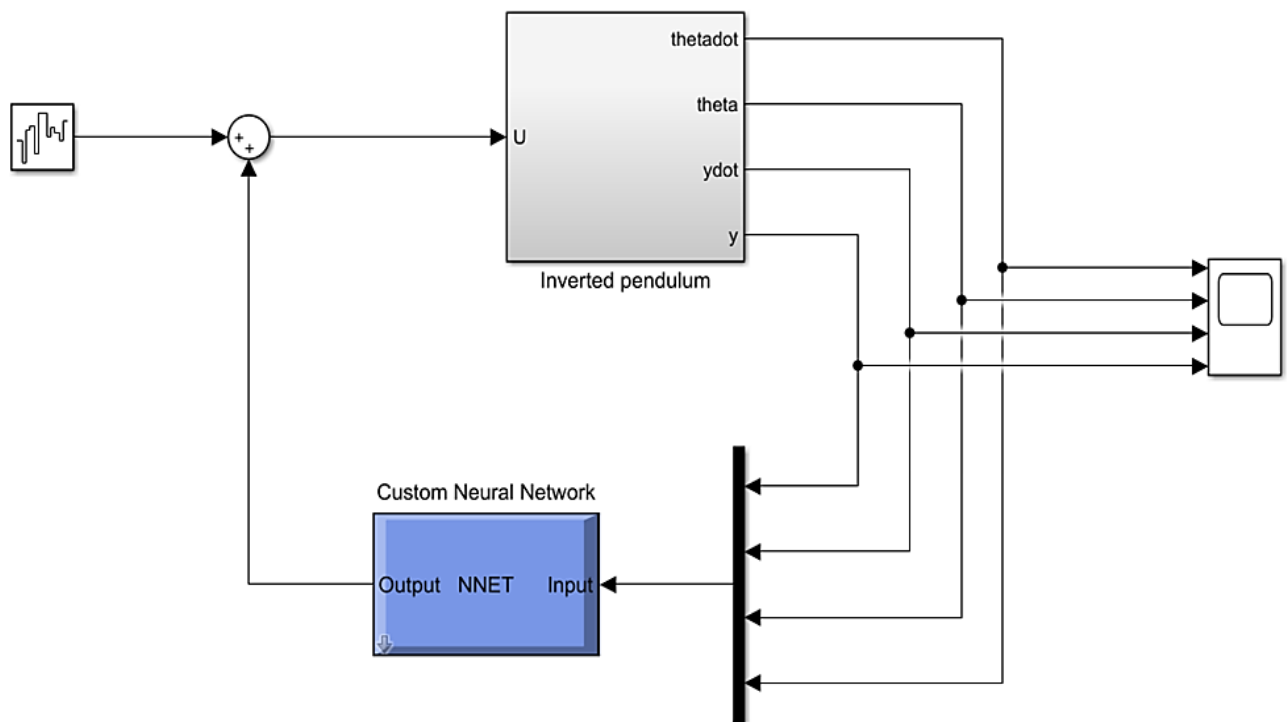


Figure III. 7 :Commande du pendule inversé par le RNA avec l'algorithme RB

- **Résultat de simulation :**

Les résultats de simulations obtenues, en utilisant le réseau de neurone avec la méthode de régulation Bayésienne sont donnés par la figure III.15.

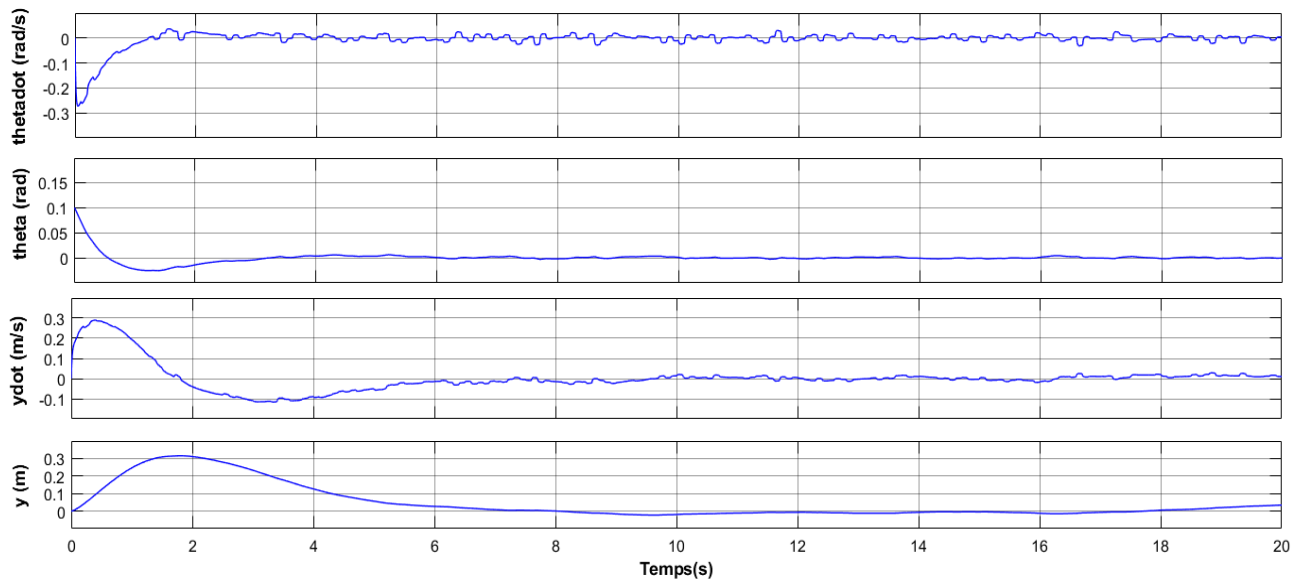


Figure III. 8 : Stabilisation du pendule inversé par le RNA avec la méthode RB

La figure III.15 représente l'état des quatre grandeurs du système pendule inversé en utilisant le réseau de neurones avec l'algorithme RB, et montre que le système se stabilise au bout de 6 s tout en respectant les contraintes imposées. En effet le déplacement du chariot ne dépasse pas 0.3 m.

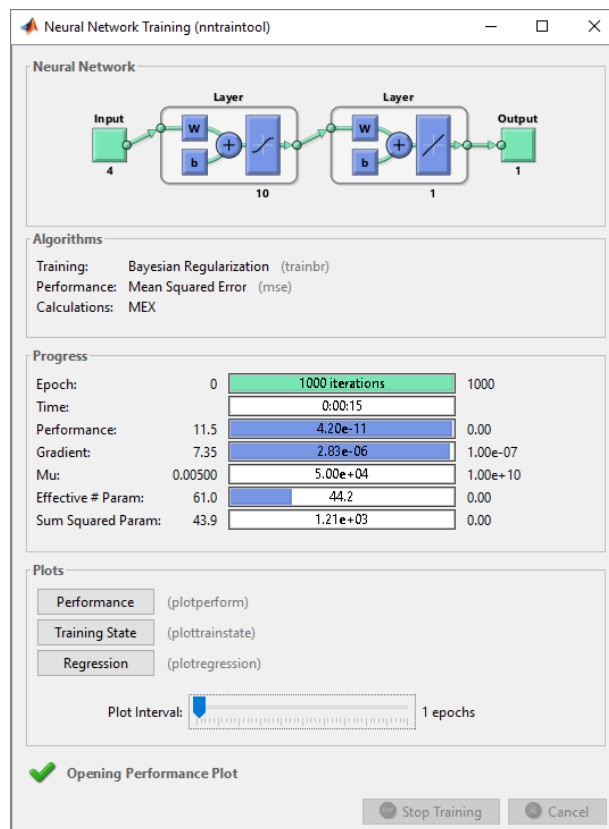


Figure III. 9: l'apprentissage du RNA par la méthode RB.

La figure III.16 représente l’outil d’apprentissage du réseau de neurones où le temps d’apprentissage écoulé et le MSE sont indiqués.

Le graph de l’erreur quadratique MSE d’apprentissage du réseau de neurones artificiel avec l’algorithme RB est représenté par la figure suivante :

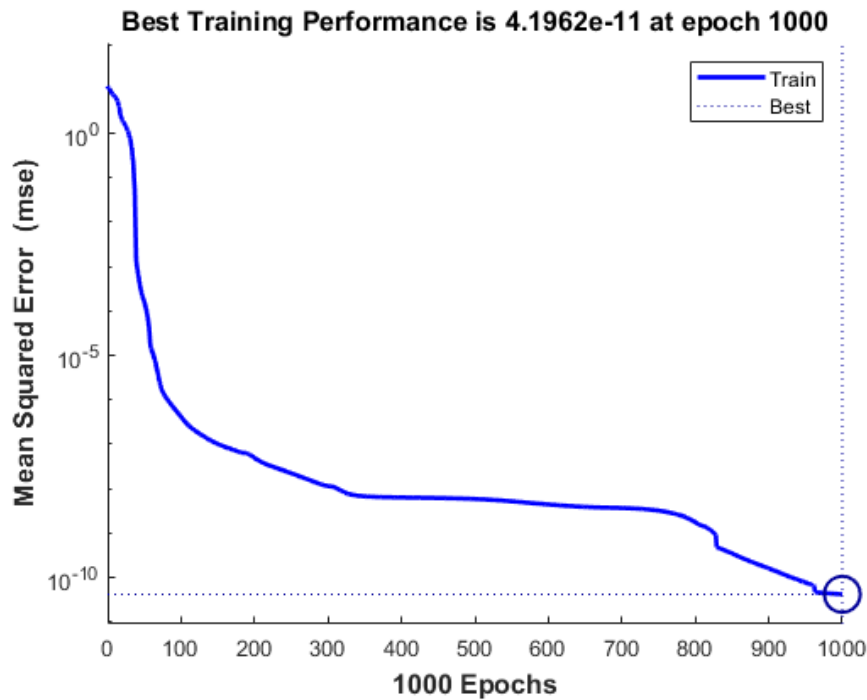


Figure III. 10 : L'erreur quadratique MSE d'apprentissage par l'algorithme RB

La figure III.18 représente le schéma qui permet de calculer l’erreur MSE de la commande entre le réseau de neurones avec l’algorithme RB et le contrôleur

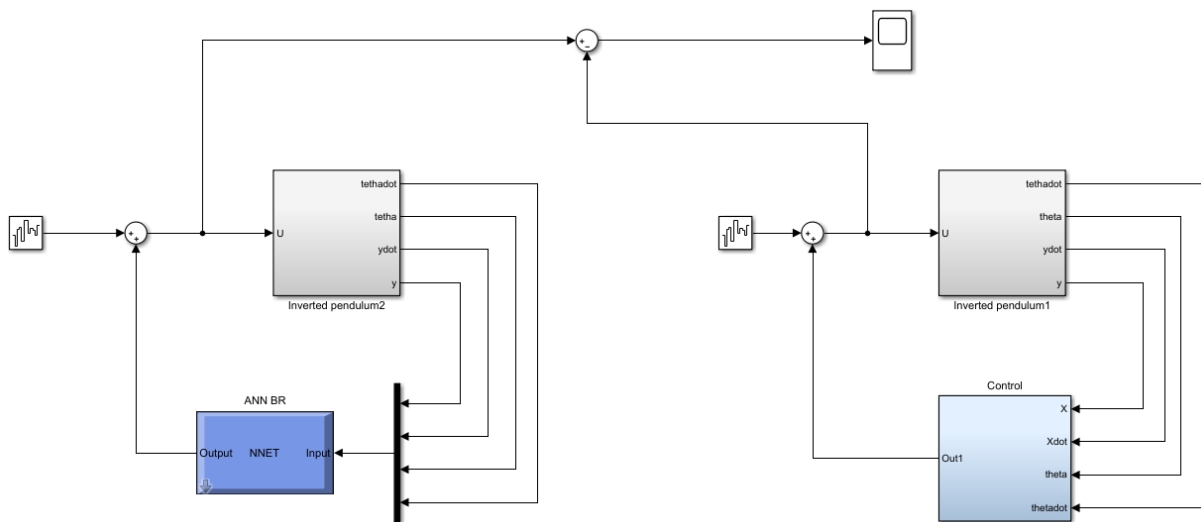


Figure III. 11 :Schéma pour calculer le MSE entre le neuro-contrôleur et le contrôleur

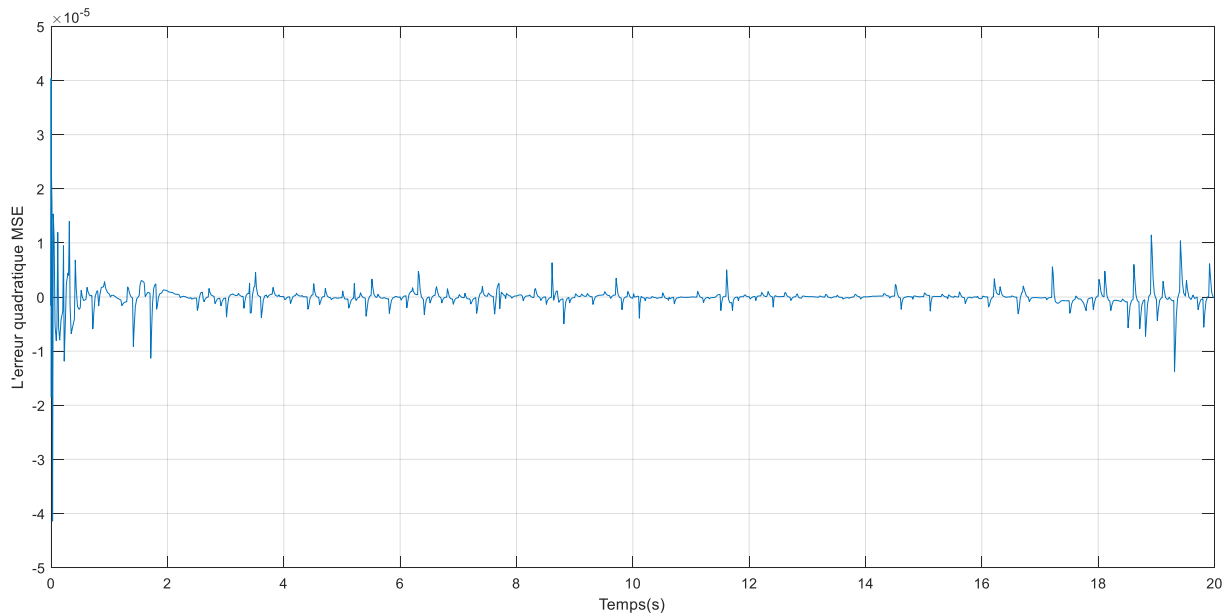


Figure III. 12 : l'erreur entre la commande du RNA et le contrôleur avec l'algorithme RB

Tableau III. 2: MSE entre le contrôleur et le RNA avec l'algorithme RB

Grandeur	MSE avec l'algorithme RB
Commande	$8.7718 * 10^{-12}$
thetadot	$3.2296 * 10^{-15}$
theta	$5.4262 * 10^{-17}$
ydot	$6.0088 * 10^{-15}$
y	$3.4599 * 10^{-15}$

III.5. Interprétation des résultats

- A partir de la figure III.5 on peut constater que le contrôleur élaboré arrive à stabiliser le pendule inversé autour du point d'équilibre. En effet il a pu le ramener de sa position initiale ($\theta=0,1$ rad) à sa position désirée
- A partir des graphes de la figure III.9 et la figure III.15 on constate que le neuro-contrôle a réussi d'imiter le contrôleur est de stabiliser le pendule inversé autour du point d'équilibre soit avec l'algorithme de Levenberg-Marquardt ou bien la méthode de régularisation Bayésienne. On constate aussi que le système se stabilise au bout de 6 à 7 sec.

- A partir des figures III.10 et III.16, on remarque que le temps d'apprentissage du réseau de neurones avec l'algorithme de LM (11 sec) est rapide par rapport à la méthode de régularisation bayésienne (15 sec)
- D'après les figures III.11 et III.17, on remarque que le MSE d'apprentissage (performance d'apprentissage) à 1000 itérations en appliquant la méthode de BR qui tend vers $4.1962 \cdot 10^{-11}$, est mieux qu'à ce avec l'algorithme de LM qui tend vers $2.6672 \cdot 10^{-10}$
- Le tableau III.3 représente une comparaison de performance d'algorithme entre l'erreur quadratique MSE du réseau de neurones avec l'algorithme Levenberg-Marquardt et le réseau de neurones avec la méthode de régularisation Bayésienne.

Tableau III. 3 : Comparaison entre MSE de trainlm et trainbr

Grandeur	MSE avec l'algorithme RB	MSE avec l'algorithme LM
Commande	$8.7718 \cdot 10^{-12}$	$5.3490 \cdot 10^{-11}$
thetadot	$3.2296 \cdot 10^{-15}$	$4.3265 \cdot 10^{-14}$
theta	$5.4262 \cdot 10^{-17}$	$6.7255 \cdot 10^{-16}$
ydot	$6.0088 \cdot 10^{-15}$	$5.3201 \cdot 10^{-14}$
y	$3.4599 \cdot 10^{-15}$	$1.2348 \cdot 10^{-13}$

- A partir du tableau III.3, on constate que la méthode de régularisation bayésienne donne des résultats meilleurs que ceux de l'algorithme de LM. On constate aussi que le taux d'erreur des deux algorithmes pour toutes les grandeurs est très petit, et cela veut dire que le réseau de neurones est une approximation très précise du contrôleur d'origine

III.6 Conclusion

Dans ce chapitre nous avons présenté la méthodes de contrôle du pendule inversé avec un réseau de neurones multicouches en appliquant l'algorithme de Levenberg-Marquardt et la méthode de régularisation Bayésienne. Ensuite on a présenté les résultats de simulation et le taux d'erreur entre le contrôleur d'origine et les réseaux de neurones, et on a conclut que le réseau de neurones a réussi à contrôler le pendule inversé autour du point d'équilibre désiré et qu'il est une approximation très précise du contrôleur d'origine.

Conclusion générale

Le travail de recherche présenté dans ce mémoire porte sur l'étude des réseaux de neurones artificiels pour le contrôle d'un pendule inversé. Dans l'objectif d'avoir un réseau de neurone performant avec un taux d'erreur minimum, on a étudié et simulé le système avec deux méthodes d'apprentissage.

L'apprentissage du réseau dépend de plusieurs paramètres qui sont : l'erreur à atteindre par la fonction de coût, le nombre d'itération, le taux d'apprentissage et la fonction d'activation. Une bonne répartition des différentes bases à apprendre au réseau donne de très bons résultats, ce qui est montré dans la phase de test. Donc un bon apprentissage dépend aussi du bon choix des bases d'apprentissage pour obtenir une bonne généralisation. Des généralités et des techniques de base sur les réseaux de neurones ont été détaillés dans le premiers chapitre.

La modélisation du pendule inversé à l'aide du formalisme de Lagrange-Euler a été détaillée dans le deuxième chapitre, pour ensuite réaliser une linéarisation qui nous définit les fonctions de transfert régissant du comportement du système. Les fonctions obtenues met en évidence les fortes non linéarités, et l'instabilité du système

Finalement, le troisième chapitre est complètement consacré à l'élaboration d'un réseau de neurones pour la commande du pendule inversé, dans un premier temps, on a réalisé une commande classique qui stabilise le pendule invresé autour de son point d'équilibre instable, après on a élaboré un réseau de neurones qui imite le contrôleur en appliquant deux algorithmes d'apprentissages qui sont l'algorithme de LevenbergMarquardt et la méthode de régularisation Bayésienne.

Les résultats obtenus dans le troisième chapitre montrent clairement l'efficacité du réseau de neurones élaboré à contrôler le pendule inversé. Le réseau de neurones a montré qu'il a la capacité de pouvoir construire un modèle mathématique à partir de données empiriques et la qualité théorique d'être un bon approximateur avec un taux d'erreur très réduit et un temps d'apprentissage court.

Notre étude s'est limitée dans ce mémoire sur le plan théorique, nous envisageons d'implémenter la technique du contrôle du pendule inversé par un réseau neuronal sur une partie hardware tel que FPGA et cela pour tester la fiabilité pratique et valider les résultats obtenues par cette commande.

Bibliographie

- [1] Merzouka NOURESSADAT : **"Etude des performances des réseaux de neurones dynamiques à représenter des systèmes réels : une approche dans l'espace d'état"**. Mémoire de Mgister. Université de Setif 1, 2009.
- [2] Sofiane IAMARENE, Ferhat AISSANI : **"Commande d'un pendule inversé"**. **Mémoire de Master**. Université Abderrahmane Mira, Bejaia. 2015.
- [3] Mohammed SALEM, Djamel Eddine CHAOUCH, Fayçal KHELFI : **« Commande neuronale inverse des systèmes non linéaires »**. 4th International Conference on Computer Integrated Manufacturing CIP, pp- 7. Centre Universitaire Mustapha Stambouli, Mascara, Algérie, 2007.
- [4] Wira PATRICE : **" Réseaux de Neurones artificiels : architectures et applications "**. Université de haute Alsace, laboratoire MIPS, 2009.
- [5] Touzet CLAUDE : **"Les réseaux de neurones artificiels-Introduction au connexionnisme"**. Editions la Machotte, 2016.
- [6] M. BENTOUMI ,**« Outils pour la détection et la classification. Application au diagnostic de défauts de surface de rail. »** Thèse de Doctorat, Université Henri Poincaré Nancy 1, 200
- [7] Nacima OUKACINE : **"Utilisation des réseaux de neurones pour la reconstitution dedéfauts enévaluation non destructive"**. Mémoire de Magister. Université Mouloud Mammeri, Tizi-ouzou, 2012.
- [8] G, Dreyfus ; J. M. Martinez ; M. Samuelides M.B .Gordon ; F. Badran ; S .Thiria ; L. Hérault, **« Réseaux de neurones : Méthodologie et applications »**, Edition EYROLLES, 2004.
- [9] Sabrina YESSAADI ; **"Un modèle basé Template Matching/ Réseau de Neurones pour la Reconnaissance des visages"**. Memoire de magister. Université Badji Mokhtar, Annaba, 2006.

- [10] Zahir ASRADJ : "**Identification des systèmes non linéaires par les réseaux de neurones**". Mémoire de magister. Université Abderahmane Mira, Béjaia, 2009.
- [11] JELIL, Radhia ABD : "**Modélisation de la relation entre les paramètres du procédé plasma et les caractéristiques de la qualité du matériau textile par apprentissage de données physiques**". Thèse de doctorat. Université de Lille 1, France, 2010.
- [12] Youcef DJERIRI : "**Les Réseaux de Neurones Artificiels**". Mémoire de magister. Université de Sidi-Bel-Abbes, 2017.
- [13] Parizeau Marc : "**RESEAUX DE NEURONES**". Université de Laval, Canada, 2004.
- [14] DjamelEddine CHAOUCH : "**Contrôle robuste des systèmes dynamiques non linéaires incertains par des approches de l'intelligence artificielle**". Thèse de Doctorat. Université Mohamed Boudiaf , Oran. 2016.
- [15] Mudry FREDDY : "**Modélisation et régulation d'un pendule inversé**". Institut d'Automatisation industrielle de l'Ecole d'Ingénieurs du Canton de Vaud. Suisse, 2003.
- [16] Billal TOUATI , Mahdi MOUSSAOUI : "**Commande d'un pendule inversé par mode glissant**". Mémoire de master. Université Abderrahmane Mira. Bejaia. 2017.
- [17] Nedjwa KERMAD : "**Etude et implémentation sur microcontrôleur de contrôleur flou**". Mémoire de magister. Université Mohamed Boudiaf, Oran. 2015.
- [18] Callinan TIM : "**Artificial Neural Network identification and control of the inverted pendulum**". 2003.
- [19] 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.

Résumé

Le travail présenté dans ce mémoire a pour objectif d'appliquer la commande par un réseau de neurones artificiel pour stabiliser un pendule inversé. Ce travail se constitue de trois chapitres, D'abord, Le premier chapitre porte sur des généralités sur les réseaux de neurones. Ensuite dans le second chapitre on a présenté le modèle mathématique du pendule inversé après la présentation et l'étude du système. L'application de la commande par le réseau de neurone sur le pendule inversé est l'objet du troisième chapitre , les résultats de simulation ont bien montré la robustesse et la fiabilité de la commande et la stabilisation du pendule inversé. Les résultats sont obtenus via Simulink/Matlab.

Mots clés : Réseaux de neurones artificiels, PMC, Commande, Pendule inversé, Algorithme de rétro-propagation.

Abstract

The objective of the work presented in this dissertation is to apply the command by an artificial neural network to stabilize an inverted pendulum. This work consists of three chapters. First of all, the first chapter deals with generality of neural networks. In the second chapter, we presented the mathematical model of the inverted pendulum after the presentation and the study of the system. The application of the control by the neural network on the inverted pendulum is the object of the third chapter, the simulation results have shown the robustness and reliability of the control and the stabilization of the inverted pendulum. The results are obtained via Simulink / Matlab.