



Mémoire de fin d'étude

Pour l'obtention du diplôme de Master

Filière : Automatique  
Spécialité : Automatique

Présenté par : **AYAD Mahieddine**

Thème

***Stabilisation d'un pendule  
inversé rotatif par approche neuronale***

Soutenu publiquement, le 03/07/2023, devant le jury composé de :

M M'HAMEDI Mohamed	Maître Assistant (A)	ESSA. Tlemcen	Président
M MERAD Lotfi	Professeur	ESSA. Tlemcen	Directeur de mémoire
M ARICHI Fayssal	Maître de conférences classe A	ESSA. Tlemcen	Co-Directeur de mémoire
M TAHOUR Ahmed	Professeur	ESSA. Tlemcen	Examineur 1
Mme KHEDIM Zeyneb	Maître de conférences classe B	ESSA. Tlemcen	Examineur 2

Année universitaire : 2022 /2023

# Dédicaces

“

*À mes parents, à mes amis et à tous ceux qui ont cru en moi, sans qui je ne serais pas là où je suis aujourd'hui,*

*Merci d'avoir été les pages blanches qui ont rempli les chapitres de mon histoire.*

”

*- Mahieddine*

# Remerciements

Tout d'abord, je remercie Allah le tout puissant, pour m'avoir donné le courage, la patience, la volonté et la force nécessaires pour surmonter tous les obstacles qui se sont hissés au travers de mon chemin.

Je tiens à remercier tout particulièrement mon encadreur **M. MERAD Lotfi**, pour son aide, la qualité de son suivi ainsi que pour tous les conseils et les informations qu'il m'a prodigué.

Je souhaite également exprimer ma gratitude à mon co-encadreur **M. ARICHI Faysal**, pour l'aide compétente qu'il m'a apporté, pour sa patience et son encouragement.

Je désire remercier également **M. TAHOUR Ahmed** pour les renseignements précieux qu'il m'a fourni.

Mes remerciements vont également à **M. M'HAMEDI Mohamed**, Maître assistant (A) à l'ESSAT, pour avoir accepté de présider le jury de ma soutenance, ainsi que : **M. TAHOUR Ahmed**, professeur à l'ESSAT et **Mme. KHEDIM Zeyneb**, Maître de conférences classe B à l'ESSAT, d'avoir accepté d'être membres du jury et pour l'intérêt qu'ils ont manifesté pour mon travail.

Pour finir, je souhaite remercier toute personne ayant contribué de près ou de loin à la réalisation de ce travail.

# Résumé

Dans ce mémoire, nous proposons une approche basée sur les réseaux de neurones artificiels pour contrôler le mouvement du pendule inversé rotatif et le stabiliser en position verticale.

Nous allons présenter une analyse du système et une étude des différentes techniques de commande existantes pour ce type de pendule. Nous expliquerons également les principes de base des réseaux de neurones et les différentes architectures utilisées pour la commande de systèmes dynamiques.

Ensuite, Nous détaillerons les différentes étapes de conception des réseaux de neurones. Nous évaluerons la performance des réseaux de neurones en comparant les résultats obtenus. Enfin, Nous analyserons également la robustesse du système face à des perturbations externes.

En somme, ce travail de recherche contribue à l'amélioration de la commande des systèmes dynamiques instables tels que le pendule inversé rotatif, en utilisant les réseaux de neurones comme outil de commande.

---

**Mots clés :** Pendule inversé rotatif, Réseau de neurones artificiels, Apprentissage supervisé, Apprentissage par renforcement, Deep Learning.

---

# Abstract

In this thesis, we propose an approach based on artificial neural networks to control the motion of the rotary inverted pendulum and stabilize it in a vertical position.

We will present an analysis of the system and a study of the various existing control techniques for this type of pendulum. We will also explain the basic principles of neural networks and the different architectures used for the control of dynamic systems.

Next, we will detail the different stages of neural network design. We will evaluate the performance of the neural networks by comparing the obtained results. Finally, we will also analyze the system's robustness in the face of external disturbances.

In summary, this research work contributes to the improvement of the control of unstable dynamic systems such as the rotary inverted pendulum, using neural networks as a command tool.

---

**Keywords :** Rotary inverted pendulum, Artificial neural networks, Supervised learning, Reinforcement learning, Deep Learning.

---

## ملخص

في هذا الأطروحة، نقترح نهجاً يعتمد على الشبكات العصبونية الاصطناعية للتحكم في حركة المنبثق المقلوب الدائري وثبتيته في وضعية رأسية.

سنقدم تحليلاً للنظام ودراسة للتقنيات المختلفة للتحكم الموجودة لهذا النوع من المنابثق. سنشرح أيضاً المبادئ الأساسية للشبكات العصبونية والهيكل المختلفة المستخدمة للتحكم في الأنظمة الديناميكية.

بعد ذلك، سنوضح مراحل تصميم الشبكات العصبونية المختلفة. سنقيّم أداء الشبكات العصبونية من خلال مقارنة النتائج المحصل عليها. وفي النهاية، سنحلل أيضاً قوة النظام في مواجهة التشويشات الخارجية.

باختصار، يسهم هذا العمل البحثي في تحسين التحكم في الأنظمة الديناميكية غير المستقرة مثل المنبثق المقلوب الدائري، باستخدام الشبكات العصبونية كأداة للتحكم.

---

كلمات مفتاحية : المدفوع العكسي الدائري، الشبكات العصبونية الاصطناعية، التعلم الإشرافي، التعلم بالتعزيز، التعلم العميق.

---

# Table des matières

Dédicaces . . . . .	I
Remerciements . . . . .	II
Résumé . . . . .	III
Abstract . . . . .	IV
V . . . . .	ملخص
Introduction générale . . . . .	1
Chapitre 1: État de l'art sur le pendule inversé rotatif . . . . .	3
1.1 Introduction . . . . .	4
1.2 Présentation du système . . . . .	4
1.2.1 Historique . . . . .	4
1.2.2 Types de pendules inversés rotatifs . . . . .	5
a. Pendule inversé rotatif simple . . . . .	5
b. Pendule inversé rotatif double . . . . .	6
c. Pendule inversé rotatif à deux axes . . . . .	7
d. Pendule inversé rotatif triple . . . . .	7
1.2.3 Intérêt de l'étude du pendule inversé . . . . .	8
1.3 État de l'art sur le PIR . . . . .	9
1.3.1 Commande à base d'énergie (EBC) . . . . .	9
1.3.2 Commande PID . . . . .	10
1.3.3 Commande par placement de pôles (PP) . . . . .	11
1.3.4 Commande LQR . . . . .	12
1.3.5 Commande LQG . . . . .	13
1.3.6 Commande H-infini . . . . .	13
1.3.7 Commande à base de logique floue . . . . .	14
1.3.8 Commande par mode glissant . . . . .	15
1.3.9 Commande par réseaux de neurones artificiels (RNA) . . . . .	16
1.4 Limitation des méthodes de stabilisation existantes dans la littérature . . . . .	16

1.5	Conclusion . . . . .	17
<b>Chapitre 2: Théorie des réseaux de neurones artificiels . . . . .</b>		<b>18</b>
2.1	Introduction . . . . .	19
2.2	Principe des réseaux de neurones artificiels . . . . .	19
2.2.1	Historique . . . . .	20
2.2.2	Neurone biologique . . . . .	21
2.2.3	Neurone artificiel . . . . .	22
2.3	Les architectures des réseaux de neurones . . . . .	22
2.3.1	Réseaux de neurones MLP (Perceptron multicouche) . . . . .	22
2.3.2	Réseaux de neurones RBF (Fonction à base radiale) . . . . .	24
2.3.3	Les réseaux de neurones à convolution CNN . . . . .	25
2.3.4	Réseaux de neurones récurrents RNN . . . . .	26
2.3.5	Réseaux Neuro-Flous adaptatifs (ANFIS) . . . . .	27
	a. Première couche : Fuzzification . . . . .	28
	b. Deuxième couche : Règles floues . . . . .	29
	c. Troisième couche : Normalisation . . . . .	29
	d. Quatrième couche : Défuzzification . . . . .	29
	e. Cinquième couche : Sommation . . . . .	29
2.4	Apprentissage supervisé . . . . .	30
2.4.1	Les fonctions de coût . . . . .	30
2.4.2	La rétropropagation du gradient . . . . .	30
2.4.3	La régularisation . . . . .	30
2.5	Apprentissage par renforcement . . . . .	31
2.5.1	Agent et environnement . . . . .	31
2.5.2	Politique . . . . .	31
2.5.3	Récompense . . . . .	32
2.5.4	Action et critique . . . . .	32
2.5.5	Algorithme DDPG . . . . .	33
	a. L'architecture des réseaux de neurones . . . . .	33
	b. Mémoire de relecture . . . . .	34
	c. Mise à jour du réseau critique (critic update) . . . . .	34
	d. Mise à jour du réseau acteur (actor update) . . . . .	34
	e. Exploration et exploitation (exploration and exploitation) . . . . .	35
	f. Politique déterministe . . . . .	35
2.6	Conclusion . . . . .	35
<b>Chapitre 3: Conception des contrôleurs à base de RNA (Application au pendule inversé rotatif) . . . . .</b>		<b>36</b>
3.1	Introduction . . . . .	37



3.2	Conception des commandes neuronales linéaires . . . . .	37
3.2.1	Commande linéaire à base de réseaux MLP . . . . .	38
3.2.2	Commande linéaire à base de réseaux RBF . . . . .	40
3.2.3	Commande linéaire à base de réseaux ANFIS . . . . .	42
3.2.4	Commande linéaire à base de réseaux DDPG . . . . .	44
3.2.5	Test de robustesse . . . . .	48
3.3	Conception des commandes neuronales non linéaires . . . . .	50
3.3.1	Commande non linéaire à base de réseaux MLP . . . . .	50
3.3.2	Commande non linéaire à base de réseaux RBF . . . . .	52
3.3.3	Commande non linéaire à base de réseaux ANFIS . . . . .	53
3.3.4	Commande non linéaire à base de réseaux DDPG . . . . .	55
3.4	Étude comparative . . . . .	59
3.5	Conclusion . . . . .	61
	<b>Conclusion et perspectives . . . . .</b>	<b>62</b>
	<b>Bibliographie . . . . .</b>	<b>65</b>

# Table des figures

1.1	Pendule inversé rotatif simple . . . . .	6
1.2	Double pendule inversé rotatif . . . . .	6
1.3	Pendule inversé rotatif à deux axes . . . . .	7
1.4	Triple pendule inversé rotatif . . . . .	7
1.5	Comparaison entre un humain et un pendule . . . . .	8
1.6	Robot de transport "Segway" . . . . .	9
1.7	Schéma du fonctionnement Swing-up . . . . .	10
1.8	Schéma du contrôleur PID cascade . . . . .	11
1.9	Schéma bloc LQG . . . . .	13
1.10	Plan de phase du mode glissant . . . . .	15
2.1	Vue d'ensemble d'un RNA . . . . .	19
2.2	Modèle d'un neurone biologique . . . . .	21
2.3	Schéma d'un neurone artificiel . . . . .	22
2.4	Structure du réseau MLP . . . . .	24
2.5	Structure du réseau RBF . . . . .	25
2.6	Structure du réseau CNN . . . . .	26
2.7	Structure du réseau RNN . . . . .	27
2.8	Principe de la neuro-floue . . . . .	27
2.9	Structure des réseaux ANFIS . . . . .	28
2.10	Schéma d'un apprentissage supervisé . . . . .	30
2.11	Schéma d'un apprentissage par renforcement . . . . .	32
2.12	Algorithme de l'apprentissage DDPG . . . . .	33
3.1	Structure du réseau neuronal MLP . . . . .	38
3.2	Apprentissage du réseau neuronal MLP . . . . .	39
3.3	Évolution de l'angle $\alpha$ pour le réseau MLP . . . . .	39
3.4	Évolution de l'angle $\theta$ pour le réseau MLP . . . . .	40
3.5	Évolution de la commande $u$ pour le réseau MLP . . . . .	40
3.6	Apprentissage du réseau neuronal RBF . . . . .	41
3.7	Évolution de l'angle $\alpha$ pour un réseau RBF . . . . .	41
3.8	Évolution de l'angle $\theta$ pour un réseau RBF . . . . .	42
3.9	Évolution de la commande $u$ pour un réseau RBF . . . . .	42

3.10	Fonctions d'appartenance ANFIS . . . . .	43
3.11	Évolution de l'angle $\alpha$ pour un réseau ANFIS . . . . .	43
3.12	Évolution de l'angle $\theta$ pour un réseau ANFIS . . . . .	44
3.13	Évolution de la commande $u$ pour un réseau ANFIS . . . . .	44
3.14	Structure des réseaux neuronaux "Critique" et "Acteur" . . . . .	45
3.15	Apprentissage par renforcement DDPG . . . . .	46
3.16	Schéma sous SIMULINK de l'algorithme DDPG . . . . .	47
3.17	Évolution de l'angle $\alpha$ pour un réseau DDPG . . . . .	47
3.18	Évolution de l'angle $\theta$ pour un réseau DDPG . . . . .	48
3.19	Évolution de la commande $u$ pour un réseau DDPG . . . . .	48
3.20	Test de robustesse MLP, ANFIS, DDPG . . . . .	49
3.21	Test de robustesse pour un réseau RBF . . . . .	49
3.22	Apprentissage du réseau neuronal MLP . . . . .	50
3.23	Évolution de l'angle $\alpha$ pour un réseau MLP . . . . .	51
3.24	Évolution de l'angle $\theta$ pour un réseau MLP . . . . .	51
3.25	Évolution de la commande $u$ pour un réseau MLP . . . . .	51
3.26	Évolution de l'angle $\alpha$ pour un réseau RBF . . . . .	52
3.27	Évolution de l'angle $\theta$ pour un réseau RBF . . . . .	52
3.28	Évolution de la commande $u$ pour un réseau RBF . . . . .	53
3.29	Fonctions d'appartenance $\alpha$ pour un réseau ANFIS . . . . .	53
3.30	Fonctions d'appartenance $\dot{\alpha}$ pour un réseau ANFIS . . . . .	54
3.31	Schéma sous SIMULINK de la commande ANFIS hybride . . . . .	54
3.32	Évolution de l'angle $\alpha$ pour un réseau ANFIS . . . . .	54
3.33	Évolution de l'angle $\theta$ pour un réseau ANFIS . . . . .	55
3.34	Évolution de la commande $u$ pour un réseau ANFIS . . . . .	55
3.35	Structure des réseaux neuronaux "Critique" et "Acteur" . . . . .	56
3.36	Apprentissage par renforcement pour un réseau DDPG . . . . .	57
3.37	Évolution de l'angle $\alpha$ pour un réseau DDPG . . . . .	58
3.38	Évolution de l'angle $\theta$ pour un réseau DDPG . . . . .	58
3.39	Évolution de la commande $u$ pour un réseau DDPG . . . . .	58
3.40	Évolution de l'angle $\alpha$ pour l'étude comparative . . . . .	59
3.41	Évolution de l'angle $\theta$ pour l'étude comparative . . . . .	60
3.42	Évolution de la commande $u$ pour l'étude comparative . . . . .	60

# Liste des tableaux

2.1	Fonctions d'activation des réseaux de neurones artificiels . . . . .	23
3.1	Paramètres de l'apprentissage DDPG . . . . .	46

# Liste des sigles et acronymes

<b>PIR</b>	<i>Pendule Inversé Rotatif</i>
<b><math>\theta</math></b>	<i>Angle du bras</i>
<b><math>\alpha</math></b>	<i>Angle du pendule</i>
<b><math>\tau</math></b>	<i>Couple</i>
<b>LQR</b>	<i>Linéaire Quadratique Régulateur</i>
<b>LQG</b>	<i>Linéaire Quadratique Gaussien</i>
<b>PID</b>	<i>Proportionnel Intégral Dérivé</i>
<b>PD</b>	<i>Proportionnel Dérivé</i>
<b>ACO</b>	<i>Ant Colony Optimization (Optimisation par colonies de fourmis)</i>
<b>GA</b>	<i>Algorithme génétique</i>
<b>DE</b>	<i>Differential Evolution (Évolution différentielle)</i>
<b>PSO</b>	<i>Particle Swarm Optimization (Optimisation par essaim de particules)</i>
<b>RNA</b>	<i>Réseau de Neurones Artificiels</i>
<b>ANFIS</b>	<i>Adaptative Neuro Fuzzy Inference System</i>
<b>MLP</b>	<i>MultiLayer Perceptron (Perceptron multicouches)</i>
<b>RBF</b>	<i>Radial Basis Function (Fonction à base radiale)</i>
<b>DDPG</b>	<i>Deep Deterministic Policy Gradient</i>

## Liste des tableaux

---

<b>FLC</b>	<i>Fuzzy Logic Controller (Contrôleur Flou)</i>
<b>SMC</b>	<i>Sliding Mode Controller (Contrôleur par mode glissant)</i>

# Introduction générale

## Contexte

L'automatisation industrielle est une discipline qui s'intéresse à la conception, au développement et à la mise en place de systèmes automatisés pour diverses applications dans l'industrie. Son objectif principal est d'améliorer l'efficacité, la fiabilité et la rentabilité des processus industriels en utilisant des technologies de pointe telles que l'Internet des objets (IoT), l'intelligence artificielle (IA) et la robotique. En somme, l'automatisation industrielle est en constante évolution grâce aux avancées technologiques et aux innovations dans le domaine de l'ingénierie des systèmes automatisés.

L'état de l'art du pendule inversé rotatif comprend un large éventail de méthodes de contrôle. Outre les approches classiques telles que la commande PID, la commande LQR et la commande PP, de nouvelles méthodes ont émergé, telles que la commande adaptative, la commande floue et la commande par réseaux de neurones. Ces approches permettent de mieux gérer les non-linéarités et les perturbations du système,

## Objectifs

Notre étude durant ce mémoire se concentre sur la commande du pendule inversé rotatif en utilisant une méthode novatrice basée sur les réseaux de neurones artificiels. L'objectif principal de cette recherche est de développer une méthode de commande précise et efficace pour réguler la position du pendule. Pour ce faire, nous avons opté pour l'utilisation de divers types de réseaux de neurones, qui nous permettent d'apprendre le comportement dynamique du système et de générer des signaux de commande adaptés. Cette méthode offre de nombreux avantages par rapport aux méthodes de commande conventionnelles, notamment une plus grande précision, une meilleure résilience face aux perturbations et une plus grande capacité d'adaptation aux changements du système.

# Organisation du mémoire

Dans le premier chapitre, nous allons commencer par présenter l'état de l'art en ce qui concerne le pendule inversé rotatif. Nous aborderons une vue d'ensemble sur les commandes modernes utilisées.

Dans le deuxième chapitre, nous allons consacrer une section détaillée pour présenter les réseaux de neurones artificiels. Nous commencerons par une définition claire de ce qu'est un réseau de neurones artificiels, ainsi que des notions et des termes nécessaires pour comprendre cette technique.

Dans le troisième chapitre de notre étude, nous proposerons un modèle de contrôleur basé sur les réseaux de neurones artificiels en essayant plusieurs types. Nous allons diviser l'étude en deux étapes : D'abord en se concentrant sur la partie linéaire du pendule. Ensuite, sur la partie non linéaire.

En conclusion de ce mémoire, nous présenterons les résultats obtenus pour cette étude. Nous soulignerons les avantages et les limites de cette méthode ainsi que son efficacité et sa robustesse comparées à d'autres méthodes de commande. Nous discuterons également des perspectives futures de recherche pour améliorer la précision et la fiabilité de la commande.



# Chapitre 1

## État de l'art sur le pendule inversé rotatif

### 1.1 Introduction

Dans ce chapitre, nous nous concentrerons sur l'état de l'art du pendule inversé rotatif. Nous allons présenter les différentes approches ainsi que les méthodologies de conception des contrôleurs les plus importants. Ensuite, nous présenterons la recherche bibliographique sur les travaux effectués : Dans un premier temps, nous allons présenter une brève description de ce qu'est un pendule inversé rotatif. Puis, nous détaillerons les travaux les plus représentatifs de la littérature traitant le problème de stabilisation du pendule dans sa position d'équilibre instable.

Finalement dans ce chapitre, on essaiera de faire émerger une problématique de recherche qui fera l'objet de notre contribution.

### 1.2 Présentation du système

Le pendule inversé rotatif (PIR) est un système mécanique sous-actionné qui est intrinsèquement non linéaire et instable. Depuis des décennies, il a été largement utilisé comme une configuration expérimentale pour expliquer et tester différents types d'algorithmes de contrôle. Il s'agit d'un système mécanique constitué d'un bras rigide suspendu à un axe de rotation horizontal et à l'extrémité duquel est fixé un pendule. Contrairement au pendule simple qui oscille dans un plan vertical, le pendule inversé rotatif peut osciller dans plusieurs plans et est capable d'effectuer des rotations autour de l'axe horizontal. Le pendule est généralement maintenu en équilibre instable en position verticale, ce qui signifie que toute perturbation entraîne des mouvements oscillatoires et/ou rotatifs.

#### 1.2.1 Historique [1]

Le pendule inversé tire son nom du fait que la masse est suspendue en haut et que la force est appliquée en bas, ce qui est l'inverse du pendule simple. Il s'agit d'un système dynamique qui a été largement étudié depuis le début du XXe siècle.

Les premières découvertes autour de son système dynamique ont été initiées par Stephenson en 1908, qui a démontré que le pendule inversé pourrait être stabilisé. Cela a été confirmé l'année suivante en 1909.

En 1932, Lowenstern a développé les équations générales de mouvement pour les pendules inversés. Ce n'est que dans les années 1960 que la dynamique des pendules inversés était bien comprise. Les auteurs présentent plusieurs traitements de la littérature entre 1960 et 1970 et déclarent que Roberge a démontré une première solution au système à pendule inversé unique du M.I.T.

Donald, Higdon et Cannon ont décrit, en 1963, des systèmes avec de multiples pendules à l'Université de Stanford et que Schaefer et Cannon ont discuté, en 1966, des systèmes à pendule inversé articulé et flexible.

À la fin des années 1960, le pendule inversé est devenu le système le plus populaire pour l'enseignement et la recherche. Des solutions de grand intérêt ont alors été proposées pour résoudre divers problèmes de contrôle. Au premier stade, le problème de stabilisation autour de la position verticale et le problème de basculement vers le haut sont résolus indépendamment. Au dernier stade, une solution complète est alors proposée.

Il faudra attendre 1992 pour l'invention du pendule inversé rotatif par le professeur Katsuhisa Furuta, d'où l'appellation "Pendule de Furuta".

Certaines solutions sont alors suivies pour résoudre le problème de suivi et de contrôle basé sur l'observateur avec le principe de séparation. Il est important de noter que la structure simple du modèle à pendule inversé permet la réalisation de validations expérimentales et est devenue l'une des principales motivations pour l'utiliser comme référence en éducation et en recherche.

De plus, de nombreux systèmes d'ingénierie sont basés sur le principe de stabilisation du pendule inversé. Pour le décollage de roquettes et le vol de missiles, par exemple, la première réalisation a été réalisée par Norris au California Institute of Technology en 1956, dans son mémoire d'ingénieur. La marche des bipèdes en utilisant le principe de stabilisation du pendule inversé a été initiée par les pionniers Hemami, Vukobratovic, Raibert et Kajita dans leurs travaux.

### 1.2.2 Types de pendules inversés rotatifs

Dans le monde de l'ingénierie, il existe une multitudes de pendules : Certains sont montés sur deux roues, d'autres possédant un volant d'inertie... Pour le cas du pendule inversé rotatif, il existe également plusieurs sous types, les plus importants sont :

#### a. Pendule inversé rotatif simple

Il s'agit d'un pendule inversé constitué d'une masse attachée à une tige pivotant sur un axe vertical. Ce type de pendule inversé est l'un des plus courants et a été largement étudié dans la littérature. Il fera également l'objet de notre étude. (Fig. 1.1)

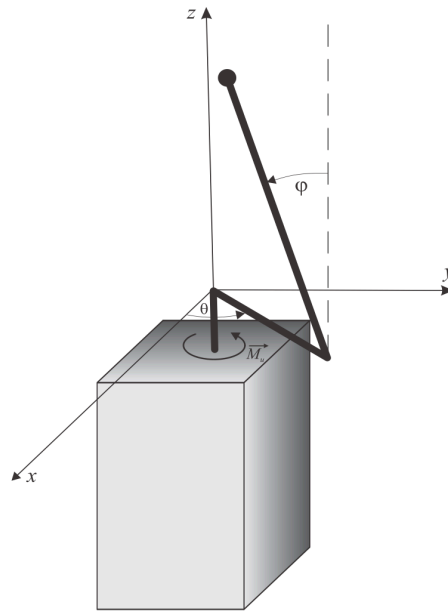


FIG. 1.1 : Pendule inversé rotatif simple [2]

### b. Pendule inversé rotatif double

Il s'agit d'un pendule inversé constitué de deux tiges pivotant sur un même axe vertical : L'une est fixée au bout de l'autre, le tout attaché à une base rotative. Ce type de pendule est plus complexe que le pendule inversé rotatif simple, car il nécessite une coordination plus fine des mouvements des deux tiges. (Fig. 1.2)

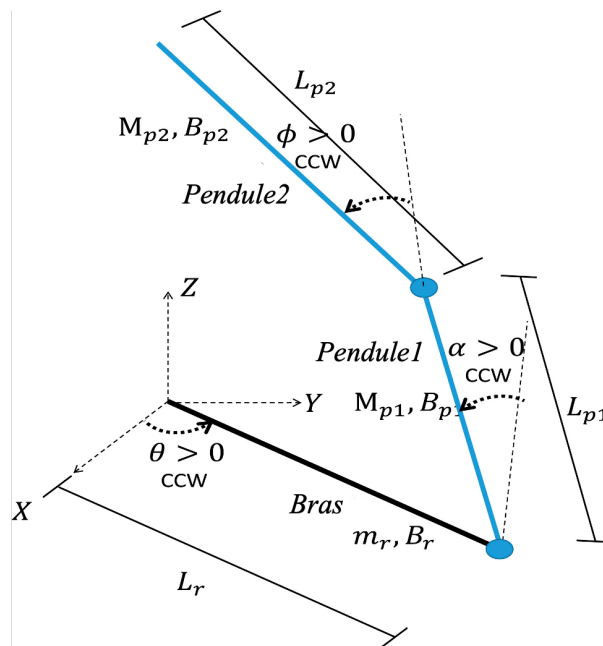


FIG. 1.2 : Double pendule inversé rotatif [3]

### c. Pendule inversé rotatif à deux axes

Dans ce cas, le pendule inversé est constitué de deux tiges pivotant sur deux axes perpendiculaires, attachées à une base rotative. Ce type de pendule est plus complexe que les pendules inversés rotatifs simples et doubles car il nécessite la coordination de deux mouvements de rotation différents, la base tourne pour dévier une tige mais risque de déstabiliser l'autre. (Fig. 1.3)

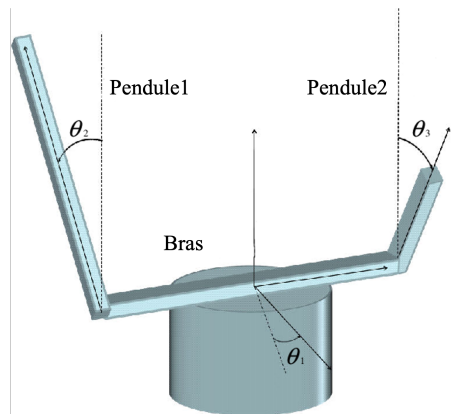


FIG. 1.3 : Pendule inversé rotatif à deux axes [4]

### d. Pendule inversé rotatif triple

C'est le même principe que le pendule inversé rotatif double, sauf qu'une nouvelle tige sera ajoutée au système, formant ainsi un système à quatre degrés de liberté. (Fig. 1.4)

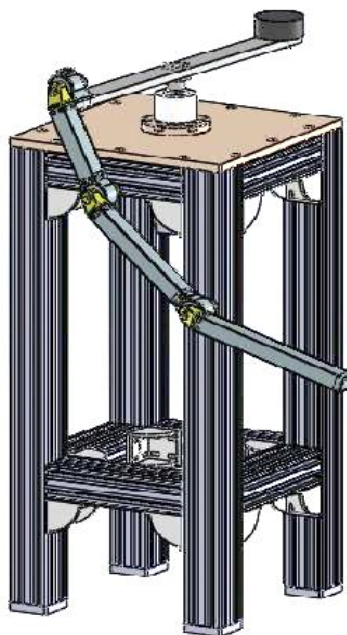


FIG. 1.4 : Triple pendule inversé rotatif [5]

Il existe d'autres types de pendules inversés rotatifs, mais ceux-ci sont les plus courants et ont été largement étudiés dans la littérature.

### 1.2.3 Intérêt de l'étude du pendule inversé

Le pendule inversé est un sujet fascinant qui intéresse de nombreux chercheurs dans plusieurs domaines. En effet, son étude implique plusieurs notions intéressantes telles que la programmation, l'automatisation, la mécanique et l'électronique.

Lorsqu'il se tient debout, l'homme peut être comparé à un pendule inversé double, dont les deux axes de rotations sont les chevilles et les hanches, comme le montre la figure (1.5). Pour maintenir son équilibre, les articulations travaillent en permanence, et l'étude de ce modèle est cruciale pour la conception de prothèses.

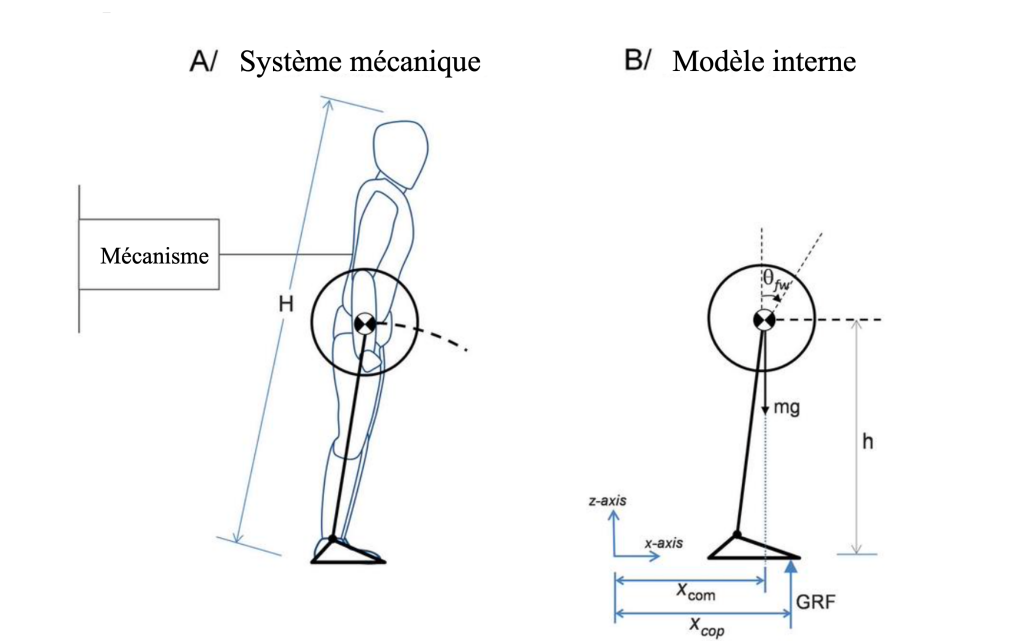


FIG. 1.5 : Comparaison entre un humain et un pendule [1]

Le pendule inversé est un sujet d'étude important surtout dans l'industrie de conception de prothèses et de robotique, où il est utilisé pour la conception de robots humanoïdes. De plus, de nombreuses applications réelles basées sur le concept du pendule inversé sont utilisées dans des domaines variés, notamment pour créer de nouveaux moyens de transport à deux roues tels que le Segway, qui permet d'avancer en se penchant en avant. (Fig. 1.6)

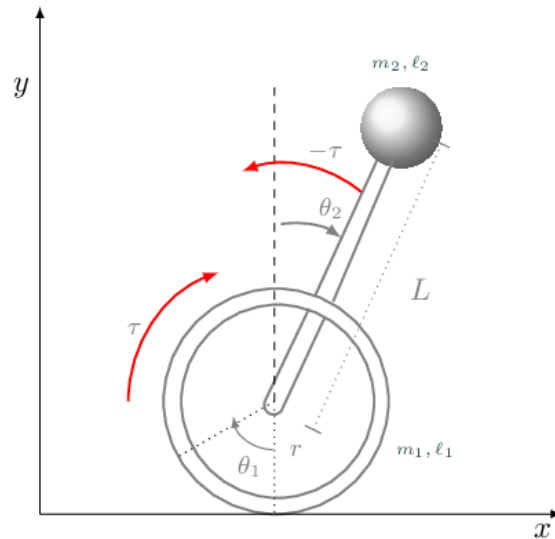


FIG. 1.6 : Robot de transport "Segway" [6]

En somme, l'étude du pendule inversé est un sujet fascinant qui peut offrir de nombreuses opportunités pour les chercheurs et les ingénieurs issues de plusieurs milieux.

### 1.3 État de l'art sur le PIR

Les principaux objectifs de contrôle du PIR sont : le contrôle de balancement, le contrôle de stabilisation et le contrôle de suivi de trajectoire. Les travaux de pointe proposés pour chaque objectif de contrôle ont été passés en revue : Il s'agit notamment de contrôleurs linéaires et d'autres non linéaires. La plupart des contrôleurs proposés appliqués au pendule inversé rotatif sont dépendants du modèle mathématique car ils sont principalement basés sur le mouvement intégral et/ou invariant, mais il existe aussi des contrôleurs qui ne nécessitent pas le besoin d'un modèle, on les appelle "model-free" [7].

#### 1.3.1 Commande à base d'énergie (EBC)

La commande EBC est une technique de contrôle qui permet de positionner un pendule en position verticale autour de sa zone d'équilibre instable, ce qui peut être très utile dans de nombreuses applications. Cette méthode est basée sur le principe de contrôle de l'énergie, où l'objectif est d'augmenter l'énergie du système progressivement jusqu'à ce que le pendule atteigne une zone linéaire [8].

L'objectif de commande EBC est mis en œuvre en hybridation avec une commande de stabilisation linéaire. En effet, lorsque le pendule est proche de la position d'équilibre instable, la commande de Swing-up échoue si aucune commande de stabilisation n'est

activée pour le maintenir dans cette position.

$$Commutation = \begin{cases} Stabilisation & |\alpha| \leq 20^\circ \\ Swing - up & |\alpha| > 20^\circ \end{cases} \quad (1.1)$$

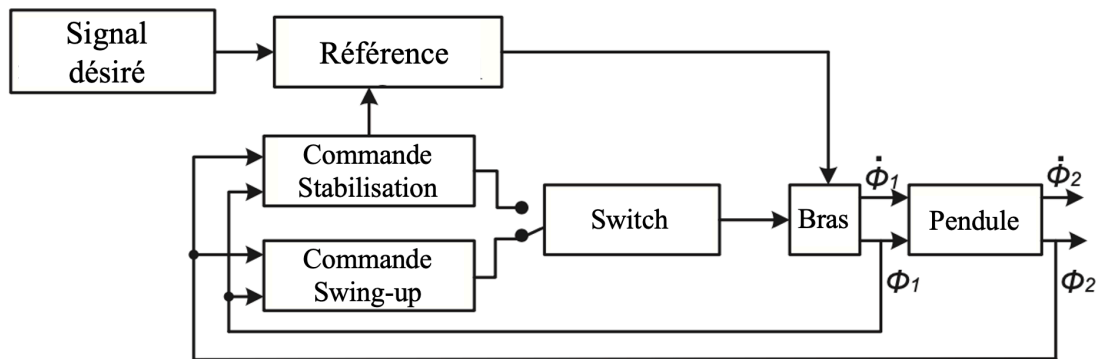


FIG. 1.7 : Schéma du fonctionnement Swing-up

La commande EBC a été étudiée [9] et les résultats obtenus indiquent que les stratégies de contrôle linéaire et non linéaire peuvent être utilisées pour stabiliser le pendule en position verticale. En général, la méthode EBC est conçue de sorte que le pendule passe par la zone d'équilibre instable. Dans son voisinage, le contrôleur de stabilisation est activé par une stratégie de commutation.

### 1.3.2 Commande PID

Dans le contexte du contrôle d'un pendule inversé rotatif, la commande PID consiste en trois termes :

Le terme proportionnel (P) : ajuste la vitesse de rotation du pendule pour le ramener à sa position de repos.

Le terme intégral (I) : assure que l'erreur continue à diminuer au fil du temps.

Le terme dérivé (D) : aide à anticiper les changements de direction du pendule.

La fonction de transfert d'un régulateur PID s'écrit de la manière suivante :

$$G_{PID}(s) = Kp + Kd * s + \frac{K_i}{s}$$

Le pendule inversé rotatif est un système à entrée unique et sorties multiples (SIMO). Dans les systèmes SIMO, un changement de l'une des sorties par des perturbations affecte la commande des autres sorties [10]. Ainsi, compte tenu du comportement non linéaire du



système, il est difficile d'atteindre un temps de stabilisation souhaité. Par conséquent, la meilleure structure de commande PID est celle dite "en cascade", car elle présente l'avantage d'atténuer l'effet des perturbations et d'améliorer la dynamique de l'ensemble de la boucle de contrôle. (Fig. 1.8)

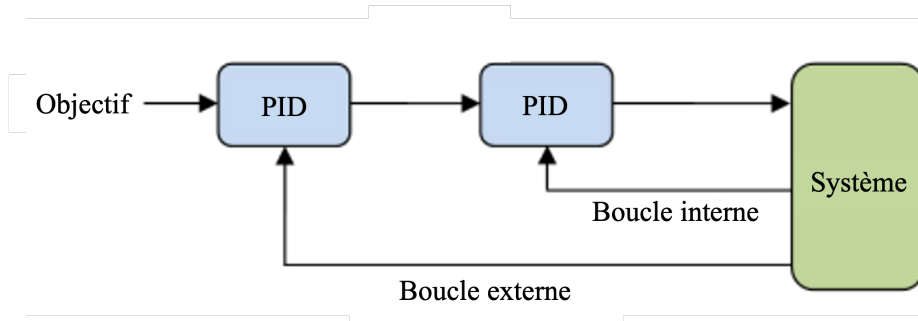


FIG. 1.8 : Schéma du contrôleur PID cascade

Des chercheurs ont proposé une commande PD à ordre fractionnaire pour la stabilisation du PIR [11]. Il a été démontré que le contrôleur PD d'ordre fractionnaire améliore les performances de commande du système car il est utilisé pour des systèmes de commande avec des plantes d'ordre entier et d'ordre fractionnaire.

Le principal problème dans les contrôleurs PID, PI et PD est de trouver les gains appropriés. Les gains acceptables peuvent être trouvés simplement par expérimentation via des essais et erreurs. Sinon, la méthode de Ziegler-Nichols peut être utilisée pour définir les gains initiaux. Cependant, des algorithmes méta-heuristiques peuvent être utilisés pour trouver les meilleurs gains par rapport à une fonction dite "objectif".

Iraj Hassanzadeh1 et Saleh Mobayen [12] ont réussi à concevoir un contrôleur PID basé sur plusieurs méta-heuristiques : L'algorithme génétique (GA), l'optimisation par essaim de particules (PSO) et l'optimisation par colonie de fourmis (ACO) pour le PIR. Les résultats de simulation et d'expérimentation montrent la robustesse et l'efficacité des contrôleurs proposés par rapport aux variations de paramètres, aux effets de bruit et aux perturbations de charge. Les méthodes proposées peuvent être considérées comme des moyens prometteurs pour le contrôle de divers systèmes non linéaires similaires.

### 1.3.3 Commande par placement de pôles (PP)

Lorsqu'on dispose du modèle linéarisé du système, un contrôleur à retour d'état peut facilement être connecté, car la conception de la matrice de gain  $K$  est la seule exigence de ce contrôleur.

La méthode de placement de pôles (PP) peut être utilisée dans ce cas pour obtenir une performance appropriée pour le contrôle du dépassement et du temps de montée.

Cette commande consiste à évaluer la matrice de gain  $K$  de manière à ce que les pôles souhaités dans le modèle linéaire du système RIP soient les nouvelles valeurs propres. Néanmoins, les solutions pour les pôles conjugués complexes sont très restrictives ou la détermination des pôles décalés doit être trouvée par des choix successifs pour obtenir un contrôle optimal [13].

Vishwa Nath et R. Mitra [13] ont proposé une commande par placement de pôles avec intégrateur. Ils ont utilisé une commande EBC pour le contrôle Swing-up. Les résultats montrent qu'une commande PP classique est suffisante car il s'agit d'un système de type 1. Les systèmes de type 1 possèdent déjà un intégrateur tandis que les systèmes de type 0 n'en possèdent pas.

### 1.3.4 Commande LQR

La régulation linéaire quadratique (LQR) est légèrement plus complexe que celle à placement de pôles (PP). La LQR fournit une loi de commande optimale pour un système linéaire avec un indice de performance quadratique  $J$  [14]. La problématique de conception consiste à trouver une entrée de commande  $u$  qui minimise l'indice de performance donné :

$$J = \int (x^T Q x + u^T R u) dt$$

où  $Q$  est une matrice de pondération d'état et  $R$  est une matrice de pondération de commande. Les matrices de pondération sont utilisées pour ajuster l'importance relative de chaque élément du vecteur d'état et du vecteur de commande dans la fonction de coût. Dans ce cas,  $K$  est donné par :

$$K = R^{-1} B^T P$$

La matrice symétrique définie positive  $P$  est la solution de l'équation de Riccati algébrique, qui est la suivante :

$$A^T P + P A - P B R^{-1} B^T P + Q = 0$$

De nombreux chercheurs ont utilisé l'approche LQR pour concevoir des contrôleurs pour le pendule inversé rotatif. Cela est probablement dû au fait que la commande LQR garantit la loi de commande optimale. Par exemple, certains chercheurs ont conçu une commande LQR basée sur l'ordre fractionnaire dans leur travail de recherche. Les résultats de stabilisation sont meilleures par rapport au LQR conventionnel [15].

D'autres chercheurs ont comparé les performances de l'approche LQR avec d'autres contrôleurs linéaires : Les résultats de cette comparaison sont contradictoires. Cela est dû au fait que la méthode LQR est montrée pour avoir une plus faible oscillation et un temps d'établissement inférieur à l'approche PP [16]. Cependant, les résultats rapportés

dans une autre étude [17] montrent que la commande PP a une robustesse supérieure à la commande LQR.

### 1.3.5 Commande LQG

Le contrôleur linéaire quadratique gaussien (LQG) est un autre type de LQR qui comprend un observateur de filtre de Kalman et un contrôleur LQR optimal comme indiqué dans la figure (1.9). Le filtre de Kalman est utilisé pour l'immunité au bruit et l'estimation d'état [18].

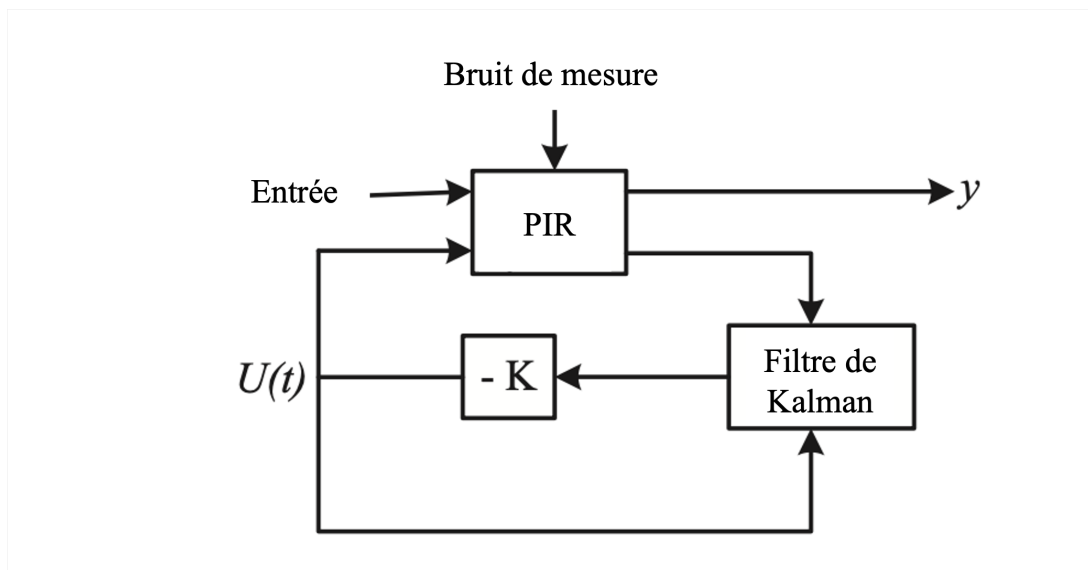


FIG. 1.9 : Schéma bloc LQG

La commande LQG a été proposée [19] pour assurer la stabilisation du pendule avec une déviation minimale des angles du bras et du pendule. Cela est basé sur l'application de grandes pénalités sur les angles du bras et du pendule dans la fonction de coût de la loi de commande optimale.

### 1.3.6 Commande H-infini

La commande  $H_\infty$  pour le pendule inversé rotatif représente une approche de contrôle robuste basée sur des techniques de filtrage. Elle est particulièrement adaptée aux systèmes sujets à des perturbations et à des incertitudes de modélisation. Le principe de base est de trouver un contrôleur qui minimise l'effet des perturbations externes sur le système tout en respectant certaines contraintes de performance. Cette approche se base sur la théorie de la commande optimale et utilise des techniques d'optimisation pour trouver le contrôleur optimal [20].

Les contrôleurs  $H_1$  et  $H_2$  sont des contrôleurs robustes et optimaux alternatifs. Un contrôleur  $H_1$  est conçu pour minimiser la sensibilité du système aux variations paramétriques et aux perturbations tout en maintenant une performance de contrôle souhaitée. Le contrôleur  $H_2$ , quant à lui, cherche à minimiser la norme de l'erreur de suivi de référence, qui mesure l'énergie de l'erreur entre la sortie du système et une référence désirée. Al-Jodah et al. [6] ont proposé un contrôleur EBC pour la commande Swing-up et un mélange de rétroaction d'état  $H_2$  et  $H_1$  pour la commande de stabilisation. Le contrôleur de stabilisation proposé a été comparé au contrôleur de rétroaction d'état complet conventionnel et au contrôleur LQR. Les résultats ont montré que la méthode de stabilisation proposée est meilleure que ses comparatifs.

### 1.3.7 Commande à base de logique floue

Les méthodes conventionnelles de réglage de commande s'appuient sur une modélisation précise du système et une analyse mathématique à l'aide de fonctions de transfert ou d'équations d'état. En revanche, la commande par logique floue offre une approche plus intuitive, permettant d'inclure les expériences acquises par les opérateurs (appelés "experts").

Cette approche a suscité un grand intérêt dans le domaine du contrôle en raison de sa robustesse [21]. En effet, ces commandes ne dépendent pas d'une modélisation analytique précise du système, ce qui les rend insensibles aux erreurs de modélisation. Cette approche peut approximer n'importe quelle loi de commande non linéaire en fonction du nombre d'ensembles flous. Cependant, la stabilité d'un contrôleur flou est difficile à confirmer. Cela est dû à la nature de la loi de commande, qui rend difficile la confirmation de la stabilité de Lyapunov. Différents types de FLC ont été utilisés pour différents objectifs de contrôle du PIR dans la littérature :

Dans [22], les auteurs ont élaboré une commande en utilisant un FLC en cascade . Les FLCs internes et externes sont basés sur un ensemble flou uniforme avec un nombre minimal de règles.

Les problèmes de commande Swing-up, de stabilisation et de suivi de trajectoire ont été résolus en utilisant un contrôleur Fuzzy-PD en cascade optimisé par intelligence artificielle basé sur un algorithme génétique (GA) et sur l'évolution différentielle (DE) [23]. Les résultats ont montré la supériorité de l'approche par évolution différentielle par rapport aux autres méthodes utilisées.

Une commande FLC dite "testable" (ou "composite" en anglais) pour le Swing-up et la stabilisation du système a été proposée par Jen-Hsing Li [24]. Il s'agit essentiellement d'une commande de rétroaction d'état par sommation floue des commandes FLC et PD. Une comparaison a été faite et les performances de la commande FLC a été montrée

comme étant meilleure en comparaison avec la commande LQR pour la stabilisation du pendule inversé rotatif [25].

### 1.3.8 Commande par mode glissant

La commande par mode glissant est une méthode de commande non linéaire qui repose sur la commutation de fonctions des variables d'état. Elle est utilisée pour créer différentes surfaces ou hypersurfaces de glissement, dont le but est de forcer la dynamique du système à correspondre à la dynamique définie par l'équation de l'hypersurface. Lorsque l'état se trouve sur l'hypersurface, le système est dans un état de glissement. De cette manière, les caractéristiques dynamiques du système ne seront pas affectées par les interférences externes, et les paramètres peuvent être déterminés tant que les conditions de l'état de glissement sont assurées. L'approche des modes glissants consiste à amener la trajectoire d'état d'un système vers la surface de glissement et de la faire commuter à l'aide d'une logique de commutation appropriée autour de celle-ci jusqu'au point d'équilibre, d'où le phénomène de glissement [26].

L'objectif de la commande par mode glissant se résume en deux points essentiels :

- Synthétiser une surface  $S(x, t)$ , telle que toutes les trajectoires du système obéissent à un comportement désiré de poursuite, de régulation et de stabilité.
- Déterminer une loi de commande (commutation)  $U(x, t)$  qui est capable d'attirer toutes les trajectoires d'état vers la surface de glissement et les maintenir sur cette surface. Cette commande guide le vecteur d'état  $x(t)$  d'un système vers l'hypersurface  $s(x) = 0$ .

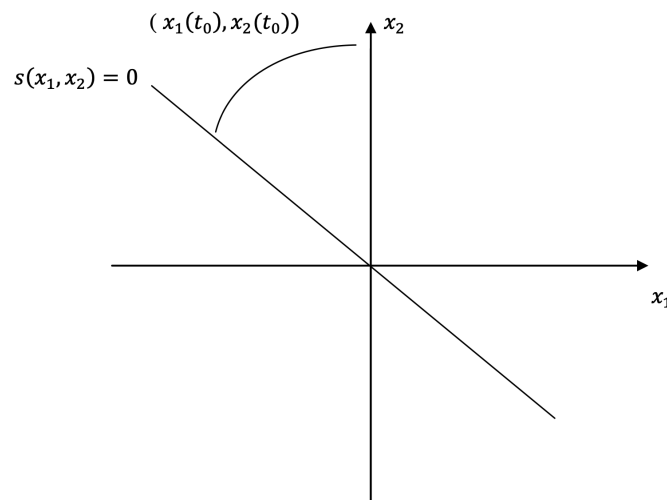


FIG. 1.10 : Plan de phase du mode glissant

Kurode et al. [6] ont proposé une commande par mode glissant pour la stabilisation du pendule inversé rotatif. La méthode proposée a été comparée aux contrôleurs PD et LQR.

Les résultats montrent que la commande par mode glissant peut effectuer le swing-up en peu de temps et qu'elle est plus robuste que les contrôleurs PD et LQR, respectivement.

L'approche SMC a été hybridés avec d'autres contrôleurs, en particulier avec un contrôleur flou [27]. Ce schéma de commande présente les avantages des deux constituants et s'écarte des limites de chaque constituant. En particulier pour supprimer les vibrations dans le régime permanent causées par le contrôleur SMC.

### 1.3.9 Commande par réseaux de neurones artificiels (RNA)

Les réseaux de neurones artificiels ont de nombreuses utilisations pratiques dans de nombreux domaines. Ils sont souvent utilisés pour résoudre des problèmes de classification, de prédiction, de reconnaissance de motifs, de traduction automatique, d'analyse de sentiments et de détection de défauts. Dans le domaine de l'automatisation, les réseaux de neurones ont été utilisés pour contrôler les mouvements des systèmes commandés, notamment le pendule inversé rotatif, car elle s'inspire des neurones biologiques.

En 2016, Javier Moreno-Valenzuela et al. [28] ont proposé une commande à base de réseaux de neurones artificiels pour le contrôle de trajectoire du pendule inversé rotatif. Les résultats ont montré de bonnes performances du système.

Benedetto Allotta et al. [29] ont réussi à concevoir une commande neuronale, non pas à apprentissage supervisé mais à apprentissage par renforcement. L'approche a donné une trajectoire finale satisfaisante malgré l'absence de base de données.

Jin Seok Noh et al. [30] ont développé une commande neuronale à base de fonction radial (RBF) pour la stabilisation d'un pendule inversé mobile. La conclusion montrait que l'ajout du contrôleur RBF au contrôleur PID a permis de créer de meilleures performances.

Rahul Agrawal et R.Mitra [31] se sont intéressés à l'étude d'une commande dite "ANFIS", qui utilise les réseaux de neurones artificiels pour concevoir un contrôleur flou. On l'appelle également commande "Neuro-floue". Les résultats de simulation démontrent que le contrôleur ANFIS est meilleur en comparaison avec les contrôleurs FLC et PID en termes de dépassement, temps de montée et de variation des paramètres.

## 1.4 Limitation des méthodes de stabilisation existantes dans la littérature

La plupart des contrôleurs proposés pour le PIR sont destinés à l'objectif de stabilisation du pendule. Bien que certaines recherches abordent également l'oscillation Swing-up, ce qui nécessite l'introduction d'une commutation.

Cependant, certaines de ces recherches n'ont appliqué les contrôleurs proposés qu'à des simulations. Bien que les études de simulation puissent être utilisées pour démontrer la robustesse face aux incertitudes et perturbations du modèle, il peut y avoir des difficultés dans la mise en œuvre pratique de tels contrôleurs. Celles-ci incluent la sélection du temps d'échantillonnage, le bruit des capteurs et le temps de latence total pour le système réel. Les résultats expérimentaux seraient préférables pour montrer l'efficacité et la robustesse des contrôleurs en raison des effets réels du système mentionnés.

Par conséquent, il est difficile, voire impossible, de mesurer l'efficacité et la robustesse des contrôleurs proposés par rapport aux paramètres physiques réels. Même pour ceux qui ont effectué la comparaison, ils n'ont pas pu en tirer une conclusion générale. En ce qui concerne le contrôle de mode de commutation, certains chercheurs ont utilisé le seuil d'angle de pendule tandis que d'autres ont utilisé le seuil d'énergie. Tous ces éléments ont l'inconvénient de causer des oscillations dans le système.

### 1.5 Conclusion

Au cours de ce chapitre, nous avons dressé un état de l'art sur le pendule inversé rotatif en passant en revue les commandes les plus intéressantes dans la littérature. Nous avons également étudié les travaux existants sur la comparaison de contrôleurs. Ces travaux montrent parfois des résultats contradictoires et inconstants. Plusieurs sources peuvent biaiser les résultats de comparaison, à savoir la procédure de préparation de la commande et les paramètres utilisés. Ainsi, il est important de proposer une approche d'évaluation et de comparaison fiable afin de sélectionner la commande la plus adéquate selon le cas étudié.

Malheureusement, il n'existe pas de "meilleure" commande pour la stabilisation du PIR. Par conséquent, tout ajout ou amélioration apportée aux commandes est bonne à prendre.

## Chapitre 2

# Théorie des réseaux de neurones artificiels



## 2.1 Introduction

La théorie des réseaux de neurones artificiels est un domaine de recherche en intelligence artificielle qui permet de modéliser des processus complexes à partir de données. Elle trouve son inspiration dans le fonctionnement du cerveau humain, qui est capable d'apprendre, de généraliser et de traiter des informations en parallèle.

Dans ce chapitre, nous allons nous focaliser sur l'étude de la théorie des réseaux de neurones artificiels. Pour comprendre cette approche, une définition claire de ce qu'est un RNA sera présentée, ainsi que les notions et termes essentiels tels que la base de données, les différents types d'apprentissage et la fusion entre les RNA et la logique floue.

## 2.2 Principe des réseaux de neurones artificiels

Les réseaux de neurones artificiels constituent une famille de fonctions permettant de construire une grande variété de modèles et de contrôleurs grâce à l'apprentissage. Inspiré du comportement d'un réseau de neurones biologiques, les RNA se composent de systèmes d'opérateurs non linéaires interconnectés et organisés en couches. Chaque neurone reçoit une ou plusieurs entrées, effectue une somme pondérée des entrées, ajoute un biais, puis applique une fonction d'activation pour produire une sortie [32][33].

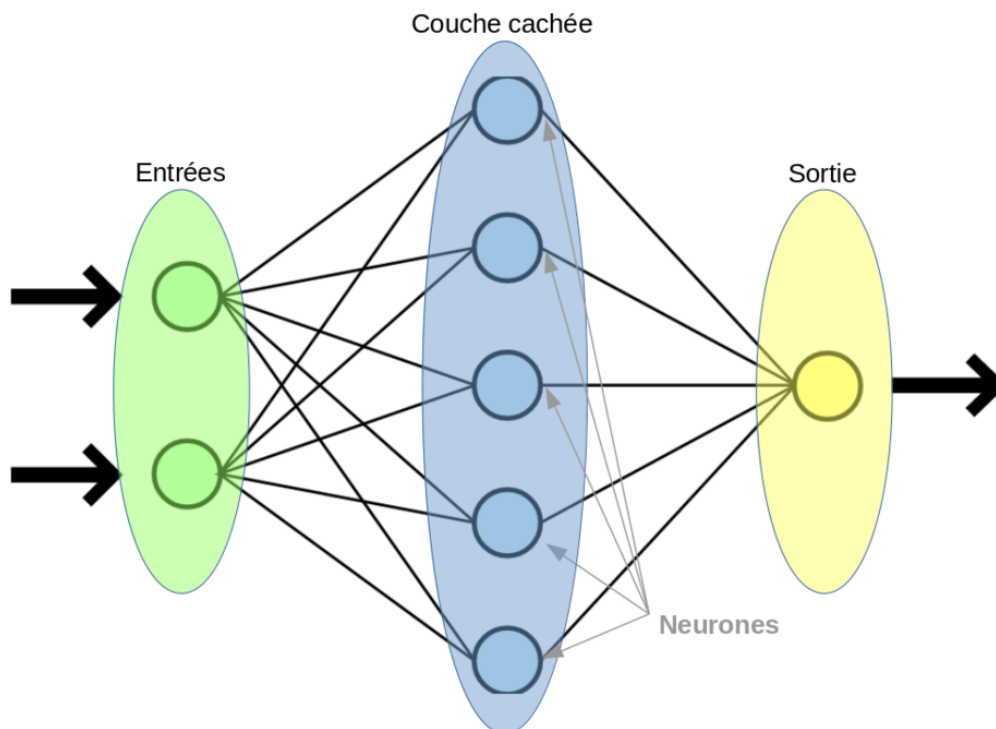


FIG. 2.1 : Vue d'ensemble d'un RNA

Cette méthode est très prisée dans les domaines où il est nécessaire de modéliser des fonctions complexes, robustes et où il faut un haut taux d'adaptabilité ainsi qu'une gestion de données non structurées. Plusieurs domaines d'applications sont des terrains envisageable pour les RNA, par exemple :

- La reconnaissance de formes.
- La vision par ordinateur.
- La traduction automatique.
- La prédiction de séries temporelles.
- La modélisation de systèmes dynamiques.
- La classification d'images.
- L'apprentissage automatique.

### 2.2.1 Historique [34]

Les travaux pionniers de Mac Culloch et Pitts en 1943 ont supposé que l'impulsion nerveuse résulte d'un calcul effectué par chaque neurone, et que la pensée émerge de l'effet collectif d'un réseau de neurones interconnectés. Bien que les premiers résultats prometteurs aient été obtenus à la fin des années 50, le manque de théorie solide a entravé leur avancement jusqu'aux années 80.

- **1890:** Le psychologue américain W. James introduit le concept de mémoire associative et propose la loi de Hebb, qui deviendra une loi de fonctionnement pour l'apprentissage sur les réseaux de neurones.

- **1943:** J. Mc Culloch et W. Pitts proposent une modélisation du neurone biologique, montrant que des réseaux de neurones formels simples peuvent réaliser des fonctions logiques, arithmétiques et symboliques complexes.

- **1949:** Le physiologiste américain D. Hebb explique le conditionnement chez l'animal par les propriétés des neurones eux-mêmes, proposant une loi de modification des propriétés des connexions entre neurones.

- **1957:** F. Rosenblatt développe le modèle du Perceptron, un neuro-ordinateur basé sur ce modèle et l'applique au domaine de la reconnaissance de formes.

- **1960:** B. Widrow développe le modèle Adaline, similaire au Perceptron mais avec une loi d'apprentissage différente, à l'origine de l'algorithme de rétropropagation de gradient.

- **1982:** J.J. Hopfield présente une théorie du fonctionnement et des possibilités des réseaux de neurones artificiels.

- **1983:** La Machine de Boltzmann est le premier modèle connu apte à traiter les

limitations du perceptron, mais son utilisation pratique est difficile en raison de la convergence de l'algorithme extrêmement longue.

• **1985:** La rétropropagation de gradient apparaît comme un algorithme d'apprentissage adapté aux Perceptrons multicouches, permettant de réaliser une fonction non linéaire d'entrée/sortie en décomposant cette fonction en une suite d'étapes linéairement séparables. Aujourd'hui, les réseaux multicouches et la rétropropagation de gradient sont les modèles les plus étudiés et les plus productifs en matière d'applications.

### 2.2.2 Neurone biologique

Le système nerveux est composé de milliards de cellules : c'est un réseau de neurones biologiques. En effet, les neurones ne sont pas indépendants les uns des autres, ils établissent entre eux des liaisons et forment des réseaux plus ou moins complexes. Le neurone biologique est composé de trois parties principales [34] :

- Le corps cellulaire composé du centre de contrôle traitant les informations reçues par les dendrites.
- Les dendrites sont les principaux fils conducteurs par lesquels transite l'information venue de l'extérieur.
- L'axone est fil conducteur qui conduit le signal de sortie du corps cellulaire vers d'autres neurones.
- Les synapses font effet de liaison et de pondération entre neurones et permettent donc aux neurones de communiquer entre eux. (Fig. 2.2)

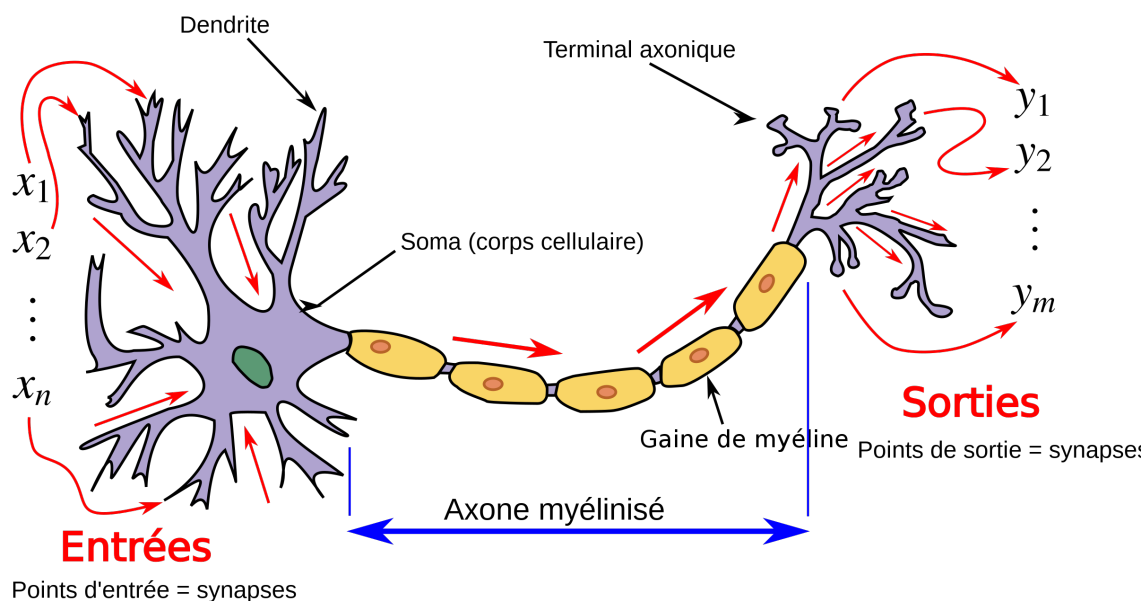


FIG. 2.2 : Modèle d'un neurone biologique [35]

### 2.2.3 Neurone artificiel

Le neurone artificiel est inspiré du fonctionnement du neurone biologique, où les éléments sont modélisés mathématiquement. Chaque neurone du réseau est un élément processeur, il est aussi défini comme étant une fonction non linéaire, paramétrée, à valeurs bornées [34].

Un neurone reçoit des valeurs en entrée  $x_i$  associées à des poids  $w_i$  représentant l'importance de ces entrées. Il renvoie une valeur unique comme sortie, qui peut être envoyée à plusieurs neurones en aval. Une fonction de combinaison calcule le potentiel du neurone qui est la somme pondérée des entrées et leurs poids à laquelle sera ajoutée le seuil  $b$ . La valeur en sortie  $y$  est calculée à partir d'une fonction  $f$  appelée fonction d'activation. Nous pouvons résumer le tout dans l'équation de sortie (2.1) :

$$y = f\left(\sum_{i=1}^N w_i x_i - b w_0\right) \quad (2.1)$$

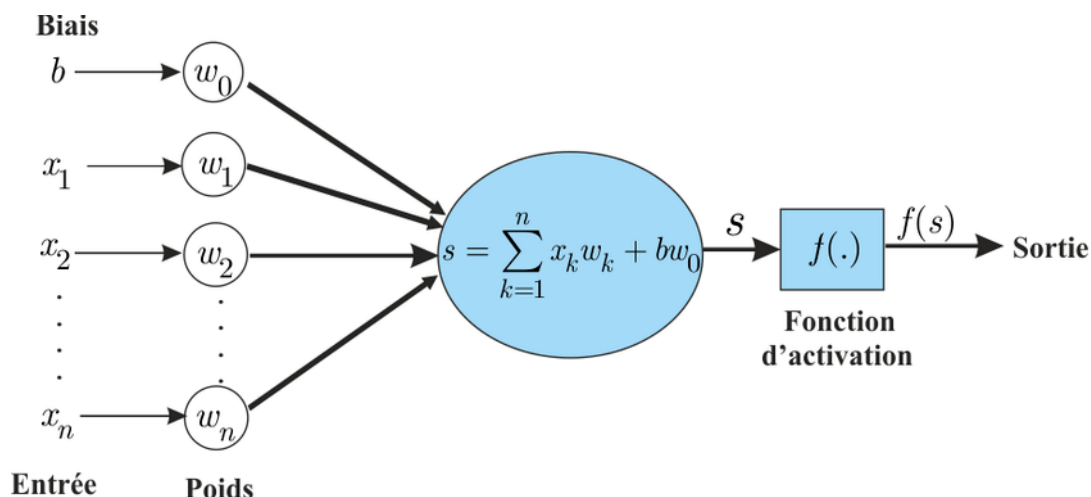


FIG. 2.3 : Schéma d'un neurone artificiel

## 2.3 Les architectures des réseaux de neurones

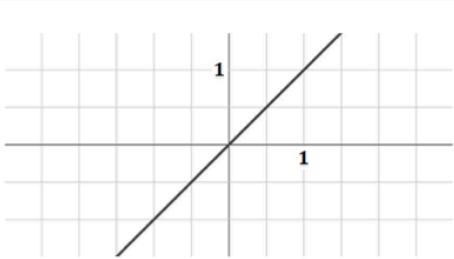
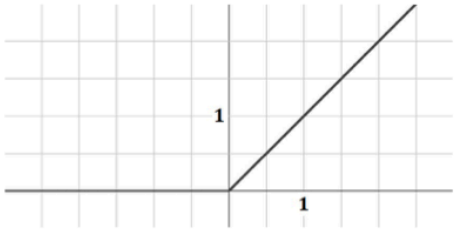
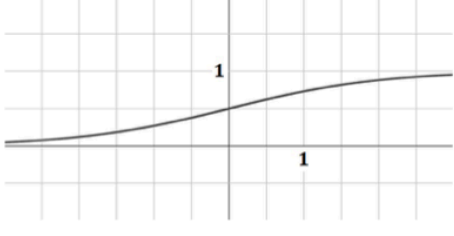
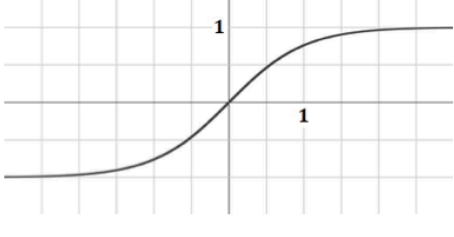
Les réseaux de neurones peuvent être classés en différentes catégories en fonction de leur architecture :

### 2.3.1 Réseaux de neurones MLP (Perceptron multicouche)

Les réseaux de neurones MLP sont une sous-catégorie de réseaux de neurones artificiels qui sont composés de plusieurs couches de neurones, comprenant une couche d'entrée, une

ou plusieurs couches cachées et une couche de sortie [36].

Chaque neurone dans les couches cachées et de sortie est connecté à tous les neurones de la couche précédente. Chaque connexion entre les neurones est pondérée par un poids qui est ajusté pendant l'apprentissage. Les neurones dans les couches cachées utilisent une fonction d'activation non linéaire pour transformer la somme pondérée des entrées, tandis que les neurones dans la couche de sortie utilisent une fonction d'activation appropriée pour la tâche à accomplir (par exemple, la fonction sigmoïde pour la classification binaire).

Nom	Graphes	Équation
Identité		$f(z) = z$
Rampe (Rectified Linear Units ReLU)		$f(z) = \begin{cases} 0 & \text{si } z < 0 \\ z & \text{si } z \geq 0 \end{cases}$
Sigmoïde		$f(x) = \frac{1}{1 + e^{-z}}$
Tangente hyperbolique		$f(z) = \tanh(z)$ $= \frac{2}{1 + e^{-2z}} - 1$

TAB. 2.1 : Fonctions d'activation des réseaux de neurones artificiels [37]

Ainsi, chaque signal alimentant un neurone dans une couche ultérieure est obtenu en multipliant l'entrée originale par un poids auquel on ajoute un seuil, puis en le faisant passer à travers une fonction d'activation [33]. La figure (2.4) montre la structure d'un réseaux de type MLP possédant trois entrées, quatre sorties et deux couches cachées, chacune possédant quatre et trois neurones respectivement.

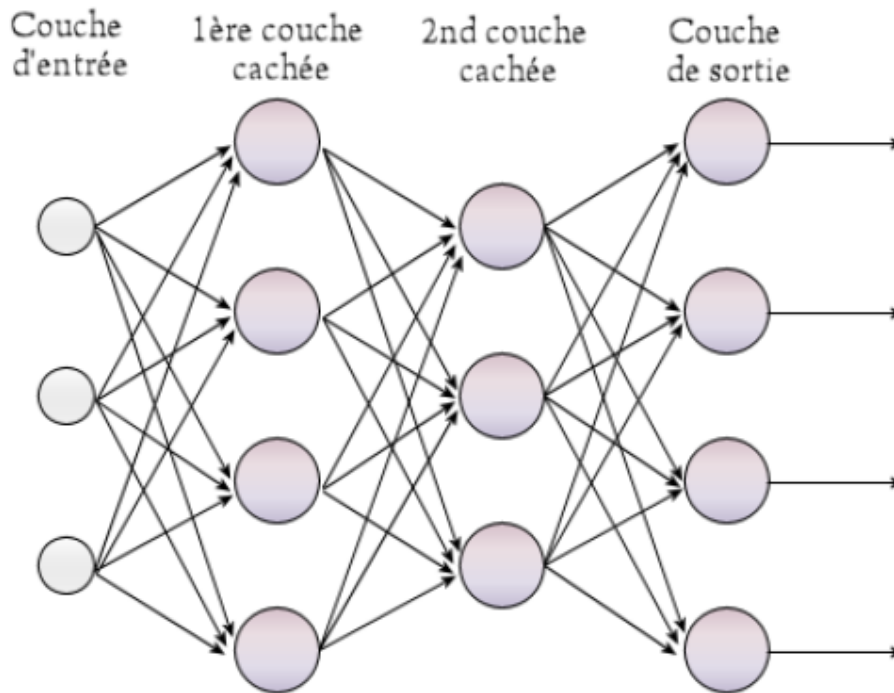


FIG. 2.4 : Structure du réseau MLP

Le réseau de neurones MLP est une méthode très efficace pour résoudre des problèmes complexes de classification et de régression, tels que la reconnaissance de caractères manuscrits, la prédiction de la consommation d'énergie, la prédiction de la qualité des produits et bien plus encore.

### 2.3.2 Réseaux de neurones RBF (Fonction à base radiale)

Contrairement aux réseaux de type MLP, les réseaux de neurones de type RBF sont composés de trois couches : une couche d'entrée, une et une seule couche cachée et une couche de sortie. La couche d'entrée reçoit les données d'entrée, la couche cachée calcule les fonctions de base radiale et la couche de sortie effectue la classification ou la prédiction [38].

La couche cachée est constituée d'un certain nombre de neurones RBF, chacun d'entre eux ayant une fonction de base radiale associée. Les fonctions de base radiale sont des fonctions qui dépendent uniquement de la distance entre l'entrée et le centre de chaque

neurone. Elles peuvent prendre différentes formes, telles que la fonction gaussienne, la fonction sigmoïde ou la fonction inverse de la distance.

Les neurones RBF calculent la distance entre l'entrée et leur centre, puis appliquent la fonction de base radiale correspondante à cette distance. La sortie de chaque neurone est alors pondérée par un poids spécifique et transmise à la couche de sortie [37].

Les réseaux RBF sont utilisés dans de nombreux domaines. En effet, ils sont très populaires dans les applications où la vitesse d'apprentissage est importante. Les réseaux RBF ont été utilisés avec succès pour la prédiction de séries temporelles, la reconnaissance de caractères manuscrits, la reconnaissance de la parole, la classification d'images et de signaux, la modélisation de processus, la prédiction de consommation d'énergie et bien plus encore. La figure (2.5) représente la structure d'un réseaux neuronale de type RBF.

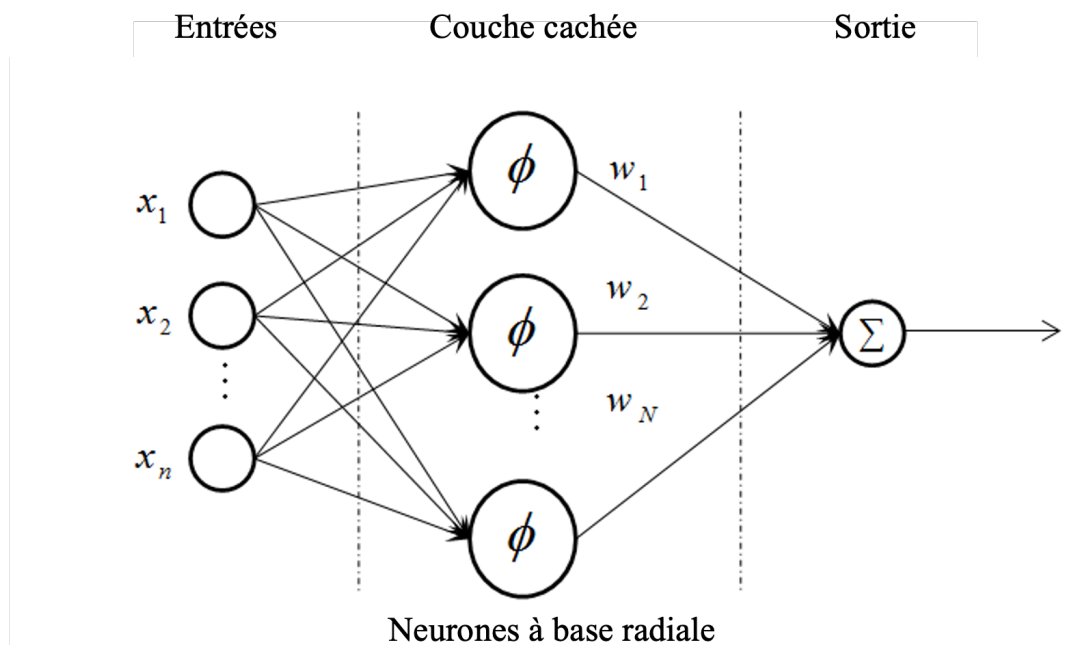


FIG. 2.5 : Structure du réseau RBF

### 2.3.3 Les réseaux de neurones à convolution CNN

Les réseaux de neurones à convolution sont particulièrement adaptés au traitement des données structurées de manière spatiale, comme les images. Leur principe de fonctionnement est inspiré du fonctionnement du cortex visuel humain. Ils utilisent des filtres de convolution pour extraire des caractéristiques des données.

La clé du succès des CNN réside dans l'utilisation des opérations de convolution. La convolution consiste à appliquer un noyau (ou filtre) à une image, en effectuant une opération de produit scalaire entre les valeurs du noyau et les pixels de l'image. Cette opération permet d'extraire des caractéristiques locales de l'image, telles que les contours,

les textures ou les motifs.

Les CNN sont constitués de plusieurs couches, notamment des couches de convolution, des couches de pooling et des couches entièrement connectées. Les couches de convolution sont responsables de l'extraction des caractéristiques en appliquant des filtres à l'image d'entrée. Les couches de pooling réduisent la dimensionnalité des caractéristiques extraites en agrégeant l'information spatiale. Les couches entièrement connectées sont utilisées pour la classification ou la régression en combinant les caractéristiques extraites [37]

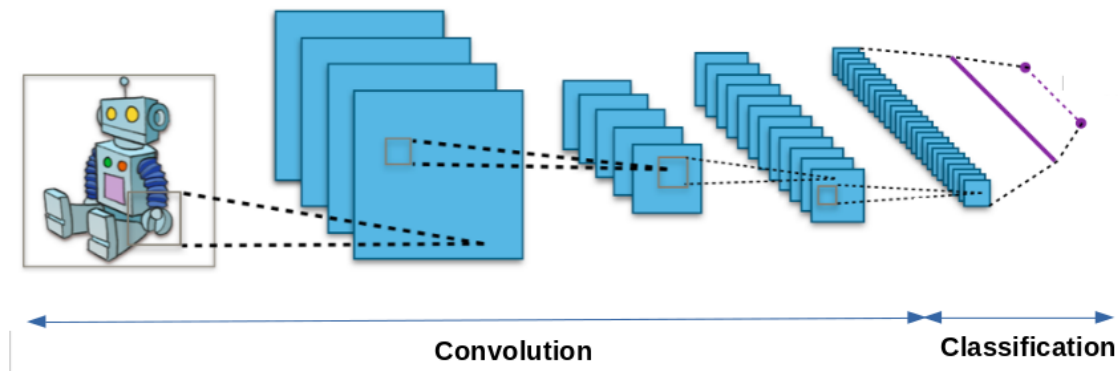


FIG. 2.6 : Structure du réseau CNN

Les réseaux de neurones à convolution sont largement utilisés dans le domaine de la vision par ordinateur, en raison de leur capacité à extraire des caractéristiques discriminantes à partir d'images. Ils ont été utilisés avec succès dans des tâches telles que la classification d'images, la détection d'objets, la segmentation sémantique, la reconnaissance faciale, la reconnaissance d'émotions et bien plus encore.

### 2.3.4 Réseaux de neurones récurrents RNN

Les réseaux de neurones récurrents sont des réseaux spécialisés dans le traitement des données séquentielles ou temporelles, telles que des séquences de mots, des séries temporelles ou des signaux audio. Ils utilisent des boucles de rétroaction pour conserver une mémoire de l'historique des données.

Le principe fondamental des RNN est de permettre la rétroaction d'informations à travers les différentes étapes de traitement. Contrairement aux réseaux de neurones classiques, les RNN ont des connexions récurrentes qui leur permettent de prendre en compte l'historique des données traitées.

Chaque unité d'un RNN est composée d'une entrée, d'une sortie et d'une rétroaction. Lorsque le réseau traite une séquence, chaque unité reçoit une entrée, produit une sortie et transmet l'information rétroactivement à l'unité suivante. Ainsi, les RNN sont capables de capturer les dépendances à long terme dans les données séquentielles [36].



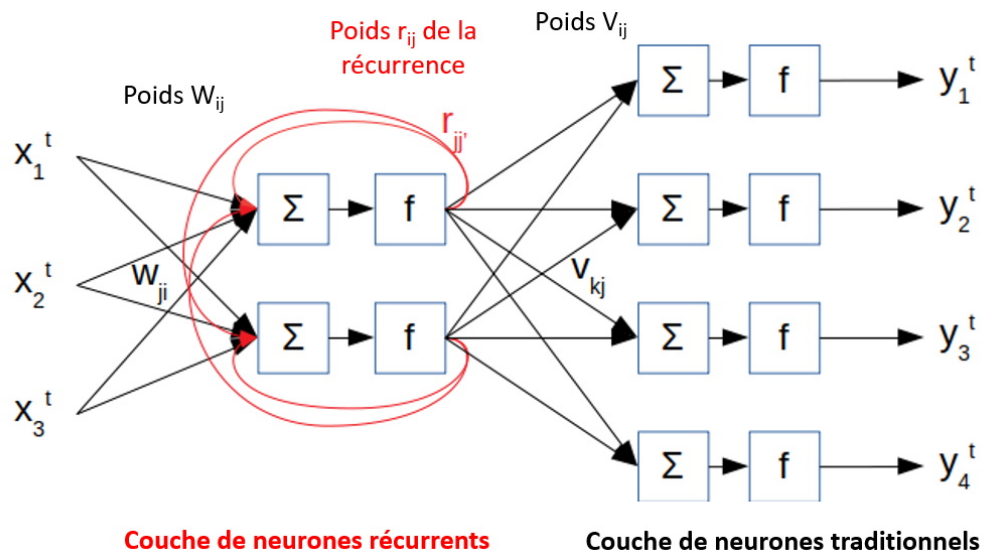


FIG. 2.7 : Structure du réseau RNN

Les RNN ont de nombreuses applications dans le domaine du traitement du langage naturel, de la génération de texte, de la traduction automatique, de la reconnaissance de la parole, de l'analyse de sentiments, de la prédiction de séries temporelles et bien plus encore. Les RNN sont capables de capturer les dépendances à long terme dans les phrases et de générer des séquences de mots cohérentes. Dans la prédiction de séries temporelles, les RNN sont utilisés pour modéliser les dépendances temporelles et prédire les valeurs futures d'une série. Ils sont utilisés dans des domaines tels que la finance, la météorologie, la santé et l'industrie pour prévoir des tendances et prendre des décisions basées sur ces prédictions.

### 2.3.5 Réseaux Neuro-Flous adaptatifs (ANFIS)

Les réseaux ANFIS (Adaptative Network Fuzzy Inference System) combinent les avantages des systèmes flous et des réseaux de neurones artificiels pour fournir une modélisation précise et souple pour les problèmes complexes. Ils sont capables de traiter des données incertaines et peuvent fournir des solutions plus robustes et plus précises que les méthodes classiques [34].



FIG. 2.8 : Principe de la neuro-floue

Le réseau adaptatif ANFIS est un réseau multicouche composé de cinq couches, où toutes les connexions ont un poids égal à 1. Les nœuds du réseau sont de deux types différents en fonction de leurs fonctionnalités. Les nœuds carrés (adaptatifs) contiennent des paramètres ajustables, tandis que les nœuds circulaires (fixes) n'ont pas de paramètres ajustables. Cependant, chaque nœud applique une fonction à ses signaux d'entrée. La figure (2.9) représente les différentes couches constituant un réseau de type ANFIS [39].

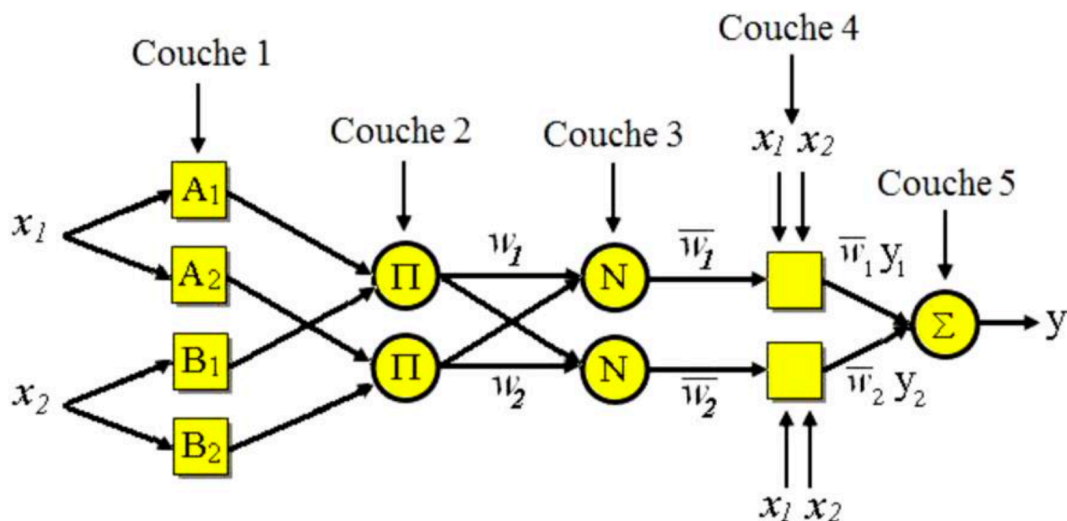


FIG. 2.9 : Structure des réseaux ANFIS

### a. Première couche : Fuzzification

La première couche du réseau joue un rôle essentiel dans la transformation des entrées. Chaque neurone de cette couche est associé à une variable linguistique spécifique. Les valeurs d'entrée sont soumises à une "fuzzification" en utilisant des fonctions d'appartenance, souvent représentées par des formes triangulaires, trapézoïdales ou Gaussiennes. Ces fonctions d'appartenance permettent de quantifier le degré d'appartenance d'une valeur d'entrée à une certaine catégorie linguistique. Ainsi, la couche de fuzzification capture les informations linguistiques des entrées et les prépare pour les étapes suivantes du processus d'inférence floue.

$$O_i^1 = \mu_{A_i}(x) \tag{2.2}$$

Où :

$x$  : entrée du nœud.

$A_i$  : terme linguistique associée à sa fonction.

$\mu$  : degré d'appartenance de  $x$  à  $A_i$ .

$O$  : sortie du nœud  $i$ .

### b. Deuxième couche : Règles floues

La deuxième couche du réseau ANFIS est composée d'un nœud pour chaque règle floue, responsable de la génération des poids synaptiques. Ces nœuds, appelés  $\Pi$ , ont un comportement fixe et calculent la valeur de sortie en effectuant le produit (opérateur "ET" de la logique floue) des signaux d'entrée correspondant au degré d'appartenance de la règle en question. Ainsi, la deuxième couche joue un rôle crucial dans la propagation de l'information et la détermination des poids synaptiques pour chaque règle floue du système.

$$O_i^2 = W_i = \mu_{A_i}(x_1) * \mu_{B_i}(x_2) \quad (2.3)$$

Où :

$i$  : représente le nombre de partition  $x_1$  et  $x_2$ .

### c. Troisième couche : Normalisation

Cette couche normalise les résultats fournis par la couche précédente. Les résultats obtenus représentent le degré d'implication de la valeur dans le résultat final.

$$O_i^3 = \frac{W_1}{W_1 + W_2} \quad (2.4)$$

Chaque nœud  $i$  de cette couche est un nœud circulaire appelé  $N$ . La sortie du nœud  $i$  est le degré d'activation normalisé de la règle  $i$ .

### d. Quatrième couche : Défuzzification

Chaque nœud de cette couche est relié aux entrées initiales. On calcule le résultat en fonction de son entrée et d'une combinaison linéaire du premier ordre des entrées initiales (Approche Takagi-Sugeno).

$$O_i^4 = V_i f_i = V_i(a_i x_1 + b_i x_2 + c_i) \quad (2.5)$$

Où  $V_i$  est la sortie de la couche 4 et  $a_i, b_i, c_i$  est l'ensemble des paramètres de sortie de la règle  $i$ .

### e. Cinquième couche : Somme

La cinquième et dernière couche s'occupe d'additionner les sorties de la couche précédente.

$$O_i^5 = \sum_{i=1} V_i f_i \quad (2.6)$$

## 2.4 Apprentissage supervisé

Les réseaux de neurones sont capables d'apprendre à partir de données. L'apprentissage supervisé se fait à partir d'une base de donnée prédéfinie, pour l'ajustement des poids et des biais du réseau de neurones afin de minimiser une fonction de coût.

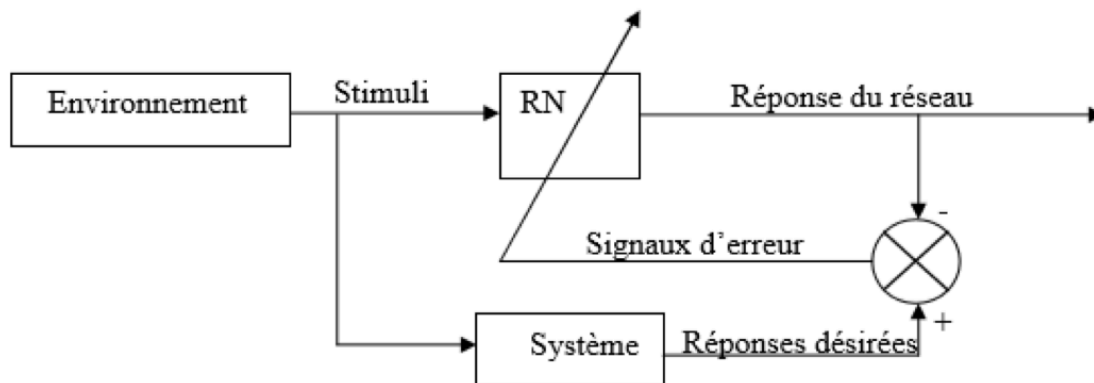


FIG. 2.10 : Schéma d'un apprentissage supervisé

### 2.4.1 Les fonctions de coût

La fonction de coût est une mesure de l'écart entre les prédictions du réseau et les sorties désirées. La fonction de coût la plus couramment utilisée est l'erreur quadratique moyenne (MSE) qui calcule la moyenne des carrés des écarts entre les prédictions et les sorties désirées. D'autres fonctions de coût sont également utilisées, comme la log-vraisemblance négative pour les problèmes de classification.

### 2.4.2 La rétropropagation du gradient

La rétropropagation du gradient est l'algorithme utilisé pour ajuster les poids et les biais du réseau de neurones. Il calcule le gradient de la fonction de coût par rapport aux poids et aux biais en utilisant la règle de la chaîne. Le gradient est ensuite utilisé pour mettre à jour les poids et les biais du réseau en utilisant la descente de gradient stochastique. La descente de gradient stochastique est une méthode d'optimisation qui ajuste les poids et les biais en suivant la direction opposée au gradient de la fonction de coût.

### 2.4.3 La régularisation

La régularisation est une technique utilisée pour prévenir le surapprentissage du réseau de neurones. Le surapprentissage se produit lorsque le réseau de neurones est trop

complexe par rapport aux données d'entraînement, ce qui conduit à une mauvaise généralisation des données. La régularisation vise à limiter la complexité du réseau de neurones en ajoutant une pénalité aux poids du réseau. La régularisation L1 ajoute une pénalité proportionnelle à la valeur absolue des poids, tandis que la régularisation L2 ajoute une pénalité proportionnelle au carré des poids.

## 2.5 Apprentissage par renforcement

Le principal avantage de l'apprentissage par renforcement, c'est qu'il ne nécessite aucune base de donnée. L'apprentissage se fait par un agent autonome à travers un processus essai-erreur pour maximiser une fonction de récompense [7]. Il existe plusieurs algorithmes d'apprentissage mais celui que nous allons utiliser dans notre étude sera l'algorithme DDPG (Deep Deterministic Policy Gradient).

### 2.5.1 Agent et environnement

Un agent est un système qui interagit avec son environnement. L'agent perçoit l'état actuel de l'environnement, prend des décisions (actions) en fonction de cet état, puis reçoit une rétroaction (récompense) de l'environnement en réponse à ses actions. L'objectif de l'agent est d'apprendre à prendre des décisions qui maximisent la récompense cumulative au fil du temps [7].

L'environnement représente le monde dans lequel l'agent évolue. Il peut être réel ou simulé, et peut varier en complexité. L'environnement fournit des informations à l'agent sous forme d'états, permet à l'agent d'effectuer des actions et génère des récompenses pour évaluer les performances de l'agent.

### 2.5.2 Politique

La politique dans l'apprentissage par renforcement définit la manière dont l'agent choisit ses actions en fonction de l'état actuel de l'environnement. Elle peut être soit déterministe (dans ce cas, l'action est choisie directement en fonction de l'état), soit stochastique (l'action est choisie selon une distribution de probabilité sur l'ensemble des actions possibles). L'objectif de l'agent est d'apprendre une politique optimale, c'est-à-dire une politique qui maximise la récompense cumulative au fil du temps [40].

### 2.5.3 Récompense

La récompense est une indication fournie par l'environnement pour évaluer la qualité des actions de l'agent. Elle peut être positive, négative ou neutre, et est généralement définie par un concepteur du système. L'agent cherche à maximiser la récompense cumulative au fil du temps en apprenant à prendre des actions qui génèrent des récompenses élevées.

La définition de récompenses appropriées est une tâche cruciale dans l'apprentissage par renforcement. Des récompenses mal définies peuvent entraîner un apprentissage inefficace ou des comportements indésirables de l'agent. Des techniques telles que les récompenses façonnées (shaped rewards) peuvent être utilisées pour guider l'agent vers des comportements souhaités.

### 2.5.4 Action et critique

L'apprentissage par renforcement utilise souvent une approche basée sur l'action et sur la critique pour apprendre la meilleure politique. L'acteur (actor) est responsable de générer des actions en fonction de l'état actuel de l'environnement, tandis que la critique (critic) est responsable d'estimer la valeur ou l'utilité d'une action dans un état donné [41].

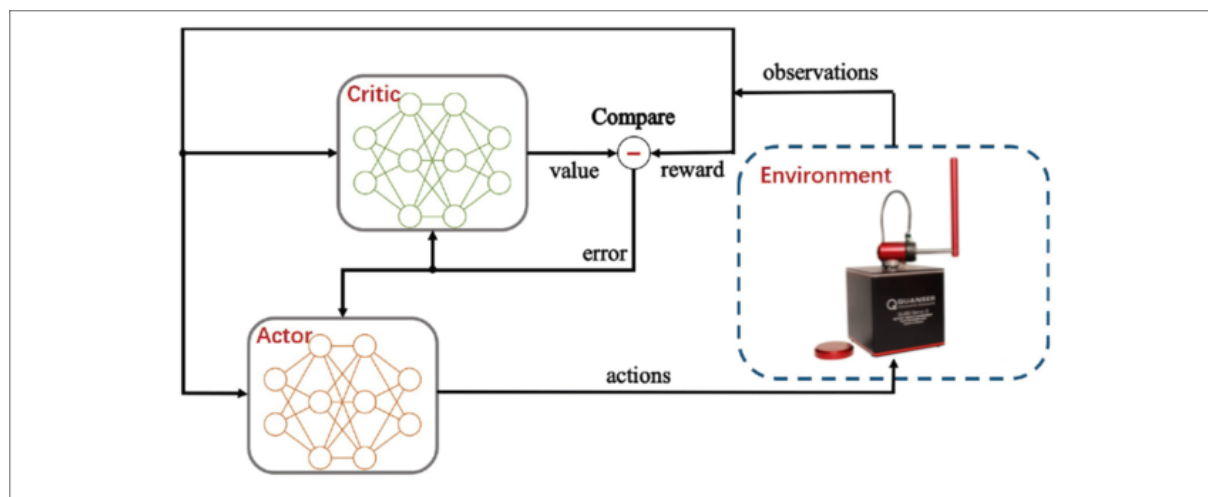


FIG. 2.11 : Schéma d'un apprentissage par renforcement

L'acteur est généralement représenté par un réseau de neurones qui prend l'état actuel en entrée et génère une action en sortie. L'objectif de l'acteur est d'apprendre une politique optimale qui maximise la récompense cumulative. Cela peut être réalisé en utilisant des techniques d'optimisation, telles que les méthodes des gradients de politique, où les paramètres du réseau de l'acteur sont ajustés pour maximiser la récompense cumulative.

La critique, quant à elle, est généralement représentée par un réseau de neurones qui prend l'état et l'action en entrée et estime la valeur de cette action dans cet état. La valeur représente l'utilité attendue de prendre cette action dans cet état donné. La critique est utilisée pour évaluer la qualité des actions prises par l'acteur et guider son apprentissage.

### 2.5.5 Algorithme DDPG

L'approche DDPG est un algorithme d'apprentissage par renforcement basé sur les réseaux de neurones profonds. Il est spécialement conçu pour résoudre des problèmes d'actions continues, comme celui du pendule inversé rotatif. L'algorithme combine un acteur et une critique [42].

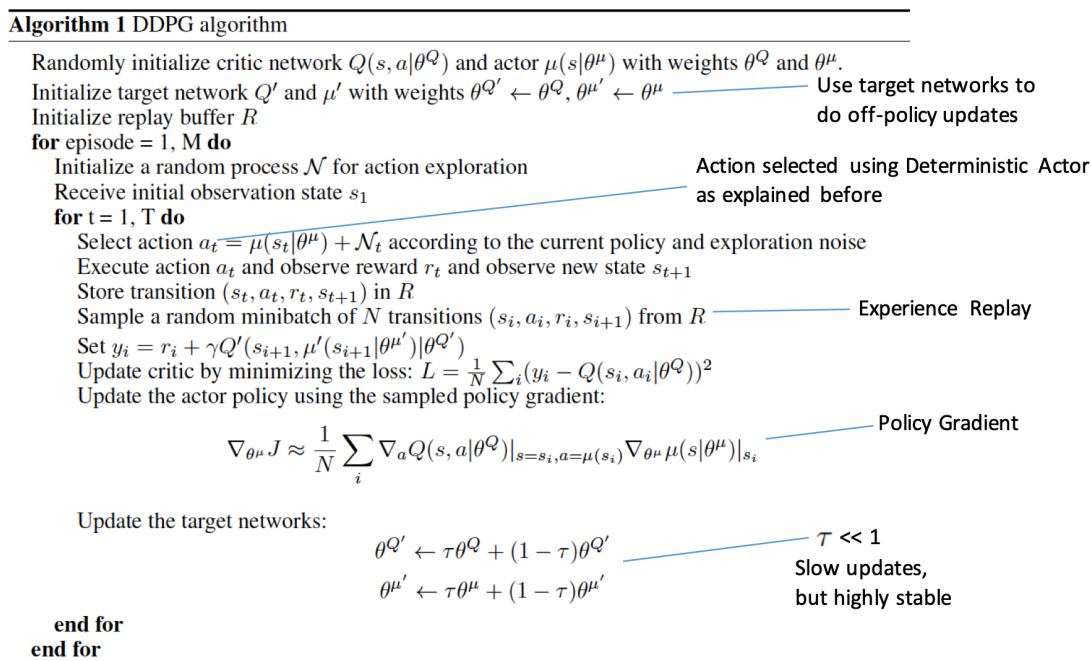


FIG. 2.12 : Algorithme de l'apprentissage DDPG

En utilisant l'acteur et la critique conjointement, l'algorithme DDPG permet d'appréhender une politique déterministe qui peut être utilisée pour prendre des décisions dans des environnements avec des actions continues. Cet algorithme a été largement utilisé dans des applications telles que la robotique, les jeux vidéo et la gestion de portefeuille, où les actions sont souvent des valeurs continues.

#### a. L'architecture des réseaux de neurones

Les réseaux de neurones utilisés dans DDPG peuvent varier en fonction du problème spécifique. Cependant, une architecture commune comprend généralement des couches

denses (fully connected layers) avec des couches d'activation non linéaires telles que ReLU (Rectified Linear Unit).

### b. Mémoire de relecture

La mémoire de relecture est une composante essentielle de DDPG. Elle est utilisée pour stocker les expériences passées de l'agent, qui sont ensuite échantillonnées aléatoirement pour l'apprentissage.

La mémoire de relecture est utilisée pour décorrélérer les transitions d'apprentissage et faciliter l'apprentissage à partir de données passées. Cela permet à l'agent d'apprendre à partir d'expériences diverses et d'éviter d'être biaisé par les transitions récentes.

### c. Mise à jour du réseau critique (critic update)

L'objectif de la mise à jour du réseau critique est de minimiser la différence entre la valeur prédite par le réseau critique et la valeur cible, qui est calculée en utilisant la récompense réelle et la valeur de l'état suivant. La fonction de perte quadratique (MSE) est souvent utilisée :

$$L = \frac{1}{N} \sum_i (y_i - Q(s_i, a_i | \theta^Q))^2 \quad (2.7)$$

Avec :

$$y_i = r_i + \gamma Q'(s_{i+1}, \mu'(s_{i+1} | \theta^{\mu'}) | \theta^{Q'})$$

$Q(s_i, a_i | \theta^Q)$  est la valeur prédite par le réseau critique pour l'état  $s$  et l'action  $a$ .

$r$  est la récompense immédiate.

$s_{i+1}$  est l'état suivant.

$\gamma$  est le facteur de remise (discount factor).

$\mu'(s_{i+1} | \theta^{\mu'})$  est l'action prédite par le réseau acteur cible pour l'état suivant.

### d. Mise à jour du réseau acteur (actor update)

La mise à jour du réseau acteur vise à maximiser la valeur prédite par le réseau critique pour les actions générées par le réseau acteur. Cela peut être réalisé en utilisant le gradient ascendant stochastique (SGD) :

$$\nabla_{\theta^\mu} J \approx \frac{1}{N} \sum_i \nabla_a Q(s, a | \theta^Q) |_{s=s_i, a=\mu(s_i)} \nabla_{\theta^\mu} \mu(s | \theta^\mu) |_{s_i} \quad (2.8)$$

Avec :

$J$  est la fonction objectif qui est la valeur prédite par le réseau critique pour les actions



générées par le réseau acteur.

$\nabla_a Q(s, a | \theta^Q)$  est le gradient de la valeur prédite par le réseau critique par rapport à l'action.

$\nabla_{\theta^\mu} \mu(s | \theta^\mu)$  est le gradient de l'action générée par le réseau acteur par rapport aux paramètres du réseau acteur.

### e. Exploration et exploitation (exploration and exploitation)

DDPG combine l'exploration et l'exploitation pour découvrir de nouvelles actions et exploiter les actions prometteuses. L'exploration est généralement réalisée en ajoutant un bruit à l'action générée par le réseau acteur. Un bruit gaussien ou un processus d'Ornstein-Uhlenbeck est souvent utilisé pour l'exploration dans DDPG [42].

En ajustant les paramètres du bruit d'exploration au fil du temps, on peut réduire progressivement l'exploration et se concentrer davantage sur l'exploitation des actions apprises.

### f. Politique déterministe

L'algorithme DDPG utilise une politique déterministe, ce qui signifie que pour chaque état donné, il génère directement une action déterministe plutôt que de tirer une action stochastique à partir d'une distribution. Cela simplifie l'apprentissage car il est possible de prendre des décisions précises en fonction des états, ce qui est particulièrement important pour les problèmes d'actions continues.

## 2.6 Conclusion

Dans ce chapitre, nous avons revu la théorie générale des réseaux de neurones artificiels. Il s'agit d'une discipline fascinante qui permet de résoudre des problèmes de classification, de prédiction et de contrôle dans de nombreux domaines. Les réseaux de neurones artificiels ont une architecture et un apprentissage différents des méthodes de contrôle traditionnelles et peuvent être utilisés pour améliorer la précision et la robustesse des systèmes de contrôle. Le pendule inversé rotatif est un exemple de système dynamique qui peut bénéficier de l'utilisation de réseaux de neurones pour le contrôle, et qui fera l'objet du chapitre suivant.

## Chapitre 3

# Conception des contrôleurs à base de RNA (Application au pendule inversé rotatif)

### 3.1 Introduction

Le pendule inversé rotatif est un système complexe et instable couramment utilisé en automatique pour évaluer les lois de commande. Dans ce chapitre, nous allons explorer l'application de plusieurs types de contrôleurs à base neuronale sur ce système : (Contrôleurs MLP/RBF/ANFIS/Réseaux Profond/DDPG). L'objectif étant de comparer leur efficacité et d'identifier celui qui produit les meilleurs résultats en terme de robustesse.

Dans ce chapitre, nous explorerons deux approches différentes pour la conception d'un contrôleur destiné au pendule inversé rotatif : La première approche se concentrera uniquement autour du point d'équilibre instable. Ensuite, nous aborderons une approche plus générale qui consiste à l'élaboration d'un contrôleur non linéaire.

Nous présenterons les résultats obtenus suite à l'application de cette commande au système étudié. Des analyses détaillées des performances seront effectuées, mettant en évidence les avantages de cette approche pour la commande du pendule inversé rotatif.

Enfin, nous procéderons à une étude comparative approfondie entre les différentes commandes proposées. Cette comparaison nous permettra de mettre en exergue les points forts et les limites de chaque méthode, ainsi que de démontrer l'efficacité et la robustesse de la commande neuronale pour le pendule inversé rotatif.

Ce chapitre vise donc à fournir une compréhension approfondie des approches de contrôle à base de RNA pour le pendule inversé rotatif, ainsi qu'à établir leur supériorité et leurs avantages par rapport aux méthodes classiques dans certaines conditions.

### 3.2 Conception des commandes neuronales linéaires

Dans une commande linéaire, le contrôleur s'occupe que de la partie linéarisé du pendule : C'est à dire qu'il s'active uniquement lorsque  $\alpha < 10^\circ$ . Nous avons utilisé les quatre erreurs d'état comme variables d'entrée, à savoir l'erreur de l'angle du pendule  $E\alpha$ , l'erreur de l'angle du bras rotatif  $E\theta$ , l'erreur de la vitesse angulaire du pendule  $E\dot{\alpha}$  et l'erreur de la vitesse angulaire du bras  $E\dot{\theta}$ , pour une seule variable de sortie, qui représente la commande du moteur  $u$ .

Avec :

$$E\theta = \theta_{reference} - \theta_t$$

$$E\alpha = \alpha_{reference} - \alpha_t$$

$$E\dot{\theta} = \dot{\theta}_{reference} - \dot{\theta}_t$$

$$E\dot{\alpha} = \dot{\alpha}_{reference} - \dot{\alpha}_t$$

Avant de concevoir les réseaux de neurones, il faut construire la base de données d'apprentissage. Pour générer l'ensemble de cette dernière, un contrôleur LQR optimal

### Chapitre 3. Conception des contrôleurs à base de RNA (Application au pendule inversé rotatif)

est conçu pour le pendule inversé rotatif [43]. L'objectif de ce contrôleur est de minimiser une fonction de coût. Après avoir obtenu le gain de retour d'état  $K$  qui est de :

$$K = \begin{bmatrix} -3.1623 & 41.1263 & -1.9329 & 3.6291 \end{bmatrix}$$

Nous avons collecté plusieurs échantillons d'observation des entrées et de la sortie sur une durée totale d'environ 20 secondes. Les 4 variables d'entrée ainsi que la variable de sortie ont été stockées dans des matrices respectivement. Cette étape permettra aux réseaux de neurones de les injecter dans le processus d'apprentissage supervisé (**rétropropagation du gradient**) afin d'imiter au maximum la loi de commande du superviseur.

#### 3.2.1 Commande linéaire à base de réseaux MLP

Les quatre signaux d'entrée et la sortie cible sont exportés dans l'espace de travail. Dans notre application, nous avons fait appel à la toolbox "Neural Networks" disponible sous MATLAB [44] pour la création et l'apprentissage du réseau neuronal MLP à partir de la base de données définie précédemment. La figure (3.1) montre la topologie du réseau de neurone MLP.

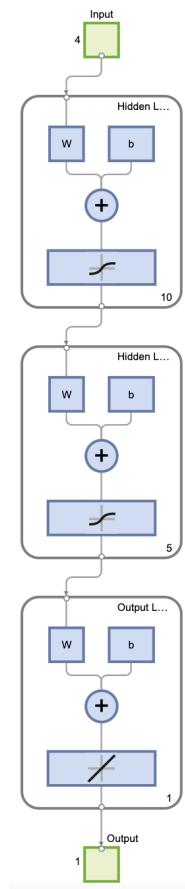


FIG. 3.1 : Structure du réseau neuronal MLP

### Chapitre 3. Conception des contrôleurs à base de RNA (Application au pendule inversé rotatif)

On peut constater que le réseau possède une couche d'entrée de 4 neurones, chacune correspondant à une entrée. Ensuite vient la première couches cachées de 10 neurones, suivie d'une deuxième couche cachée de 5 neurones. À la fin de celle ci, on trouve la couche de sortie composée d'un seule neurone qui représente la commande. Les fonctions d'activation dans les deux couches cachées sont des fonctions **tangente-sigmoïde**, et la couche de sortie utilise une fonction **linéaire**. L'algorithme d'apprentissage utilisé est celui de la **rétropropagation du gradient**.

Le graphique (Fig. 3.2) montre l'évolution de l'erreur au cours des itérations. Cette dernière est de l'ordre de  $10^{-3}$  après 425 itérations. Il s'agit d'une bonne performance d'apprentissage.

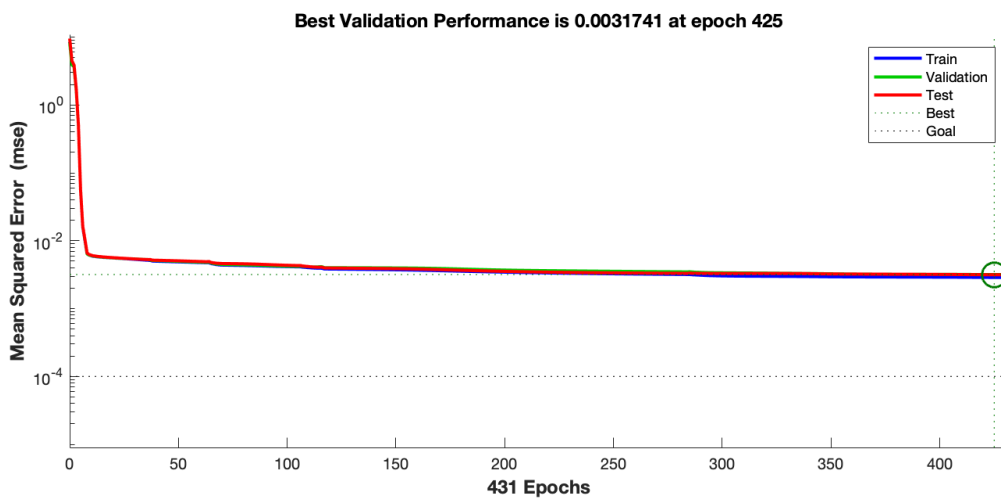


FIG. 3.2 : Apprentissage du réseau neuronal MLP

Les figures (3.3, 3.4, 3.5) représentent les résultats expérimentaux de l'angle du pendule, l'angle du bras rotatif ainsi que la commande délivrée :

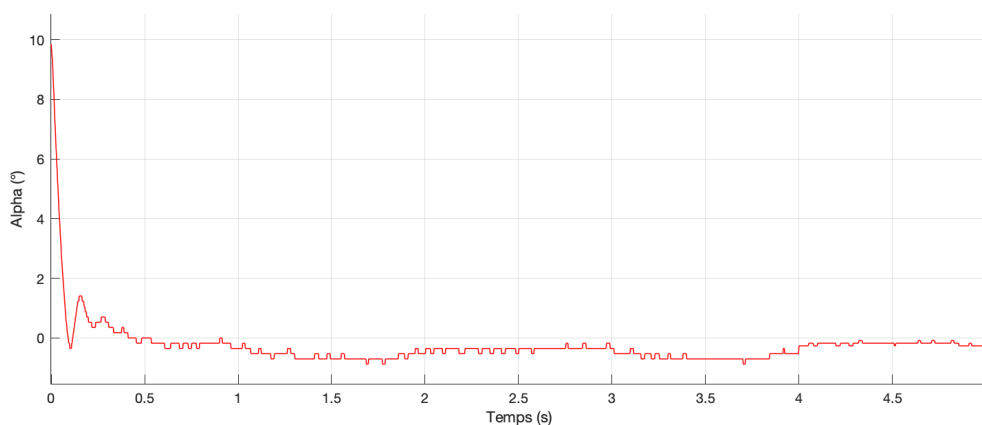


FIG. 3.3 : Évolution de l'angle  $\alpha$  pour le réseau MLP

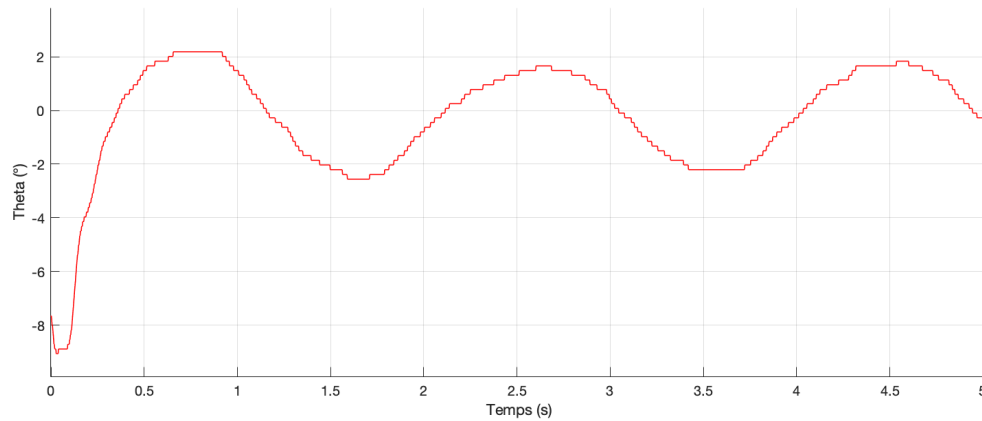


FIG. 3.4 : Évolution de l'angle  $\theta$  pour le réseau MLP

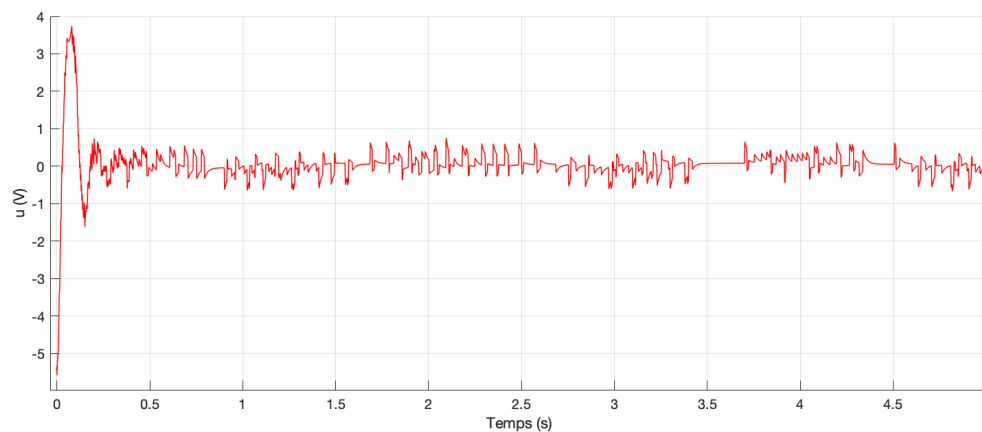


FIG. 3.5 : Évolution de la commande  $u$  pour le réseau MLP

On remarque la convergence très rapide de l'angle  $\alpha$  malgré la présence d'une erreur statique minimale (environ  $1^\circ$ ) qui est extrêmement négligeable. L'angle  $\theta$  quant à lui oscille entre  $\pm 2^\circ$ , qui s'avère lui être une bonne performance.

### 3.2.2 Commande linéaire à base de réseaux RBF

La commande neuronale RBF a été construite à partir de la même base de données [43]. Nous avons fait appel à la même toolbox MATLAB que pour le réseau MLP [44]. La topologie d'un réseau de neurone RBF ressemble à celle d'un réseau de neurones MLP, sauf qu'il possède une seule couche cachée : Notre réseau se constitue donc d'une couche d'entrée de 4 neurones, une couche cachée qui augmentera son nombre de neurones progressivement lors de l'apprentissage jusqu'à atteindre une erreur désirable, et une couche de sortie à un neurone qui représente la commande.

### Chapitre 3. Conception des contrôleurs à base de RNA (Application au pendule inversé rotatif)

La figure (3.6) montre la phase d'apprentissage de notre réseau.

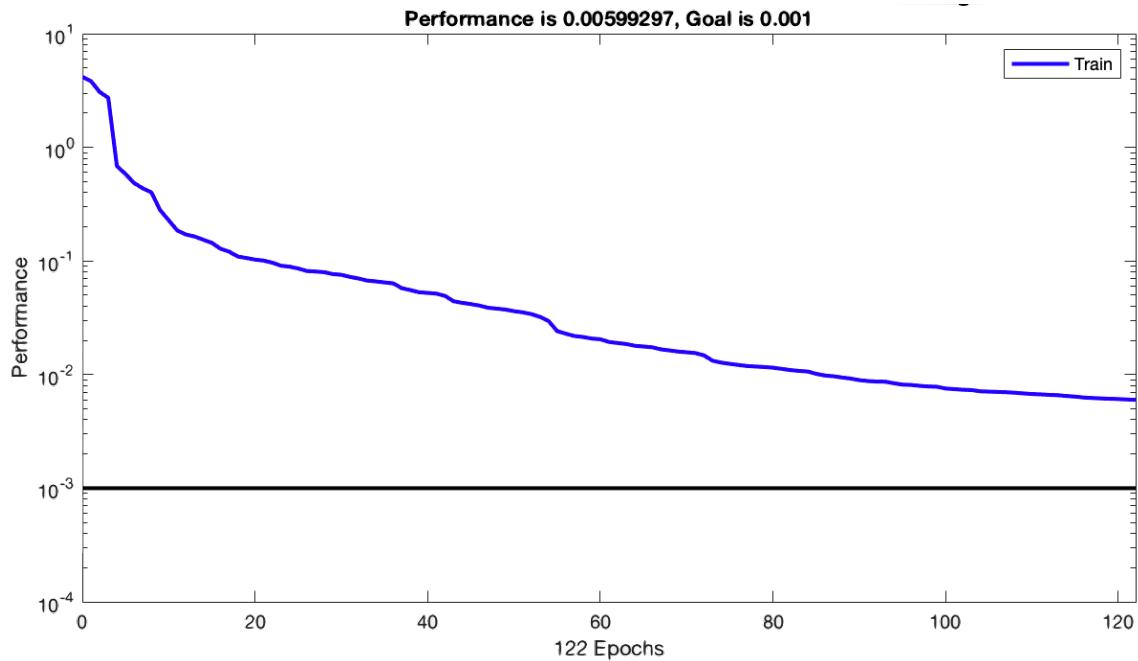


FIG. 3.6 : Apprentissage du réseau neuronal RBF

L'algorithme n'a pas pu atteindre l'erreur désirée de 0.01 en un temps d'apprentissage raisonnable, mais il a quand même réussi à atteindre une erreur de 0.05 avec une couche cachée de 122 neurones, donnant aussi un résultat très satisfaisant. Notre réseau de neurone RBF possède donc 122 neurones dans sa couche cachée.

Les figures (3.7, 3.8, 3.9) représentent les résultats expérimentaux de la commande :

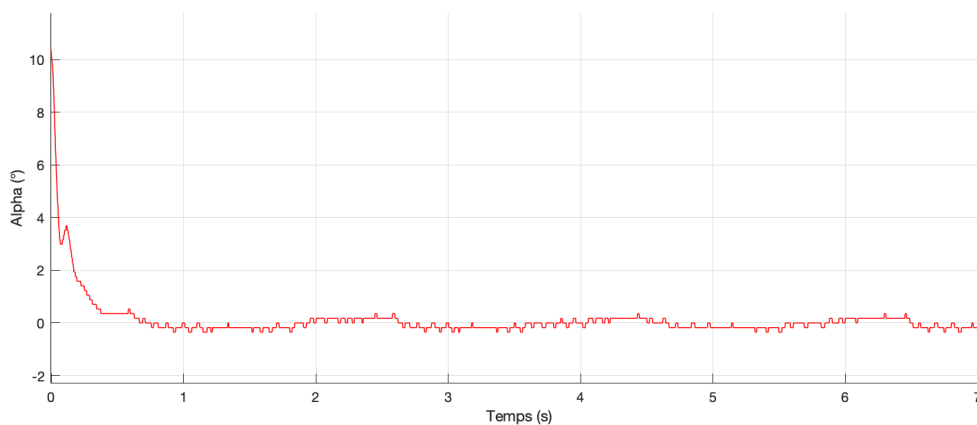


FIG. 3.7 : Évolution de l'angle  $\alpha$  pour un réseau RBF

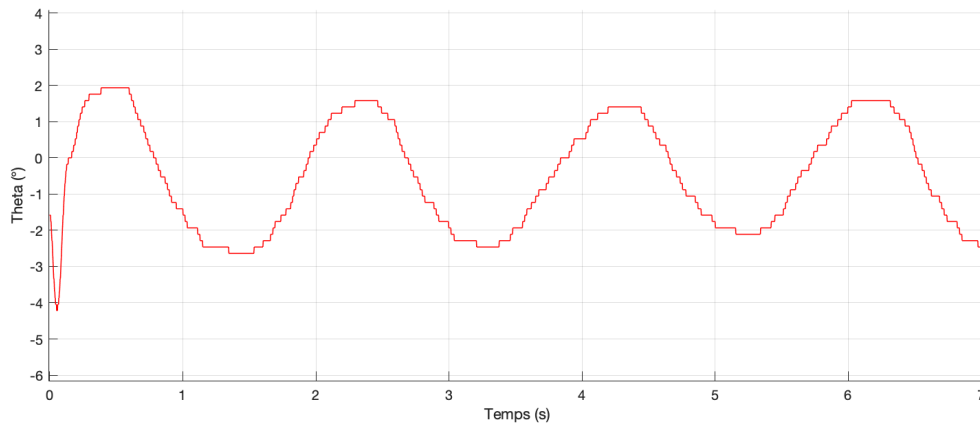


FIG. 3.8 : Évolution de l'angle  $\theta$  pour un réseau RBF

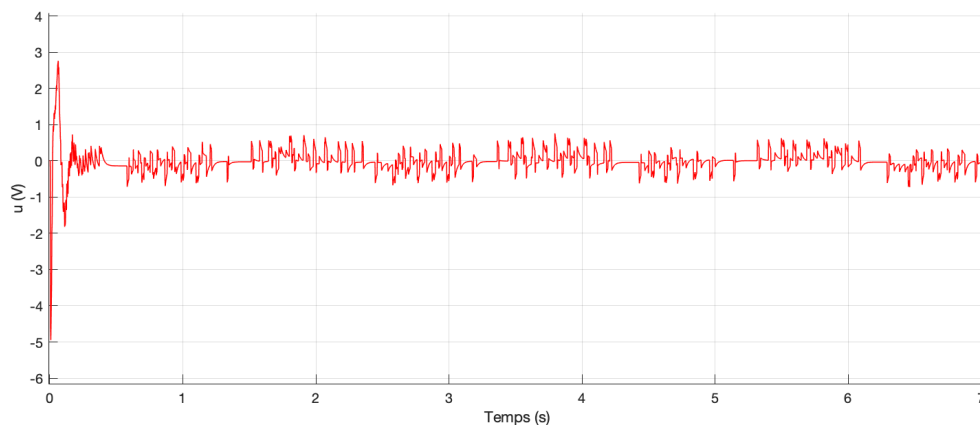


FIG. 3.9 : Évolution de la commande  $u$  pour un réseau RBF

Les résultats de la commande RBF sont très similaires à ceux de la commande MLP (Convergence très rapide et erreur statique négligeable). La seule différence réside dans l'apprentissage, où l'approche RBF nécessite plus de temps et fournit une erreur légèrement plus grande.

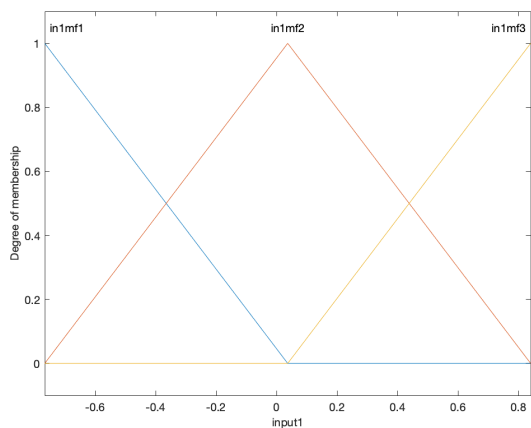
### 3.2.3 Commande linéaire à base de réseaux ANFIS

Pour la conception de la commande neuro-floue, la toolbox MATLAB **anfisedit** à été utilisée. Elle permet la création et l'apprentissage du réseau neuro-flou. Ce dernier nous fournira l'architecture de la commande floue.

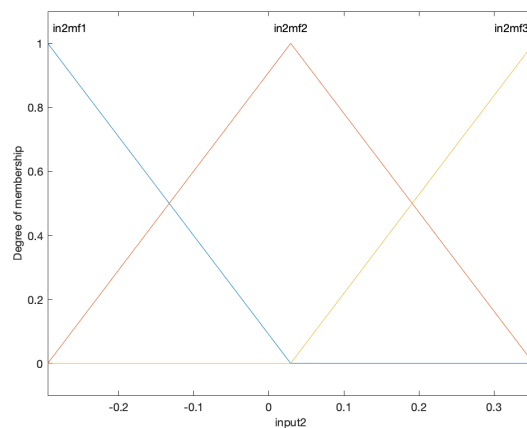
Nous avons choisit des fonctions d'appartenances de type triangulaire, avec trois fonctions d'appartenance par entrée (Fig. 3.10). Par conséquent, la commande floue se composera que  $3^4 = 81$  singletons. Les résultats nous ont données des singletons balayant l'intervale  $[-14.5 \quad 15.87]$ .



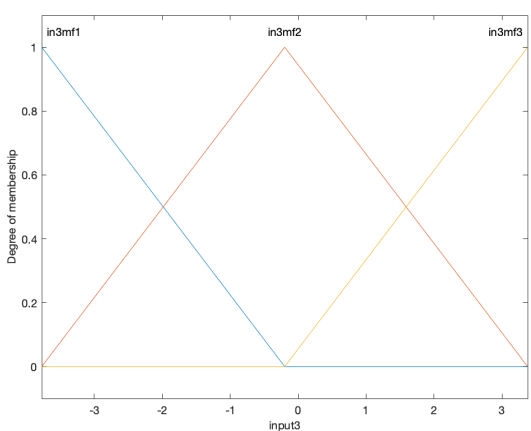
### Chapitre 3. Conception des contrôleurs à base de RNA (Application au pendule inversé rotatif)



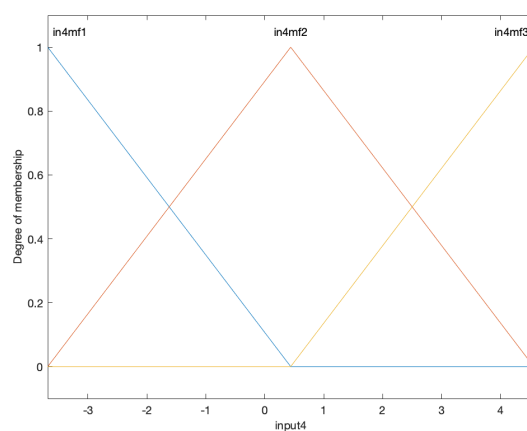
(a)  $\theta$



(b)  $\alpha$



(c)  $\dot{\theta}$



(d)  $\dot{\alpha}$

FIG. 3.10 : Fonctions d'appartenance ANFIS

L'erreur obtenue une fois la phase d'apprentissage achevée est de 0.0779898.

Les figures (3.11, 3.12, 3.13) représentent les résultats expérimentaux des angles et de la commande linéaire pour la commande neuro-flou (ANFIS).

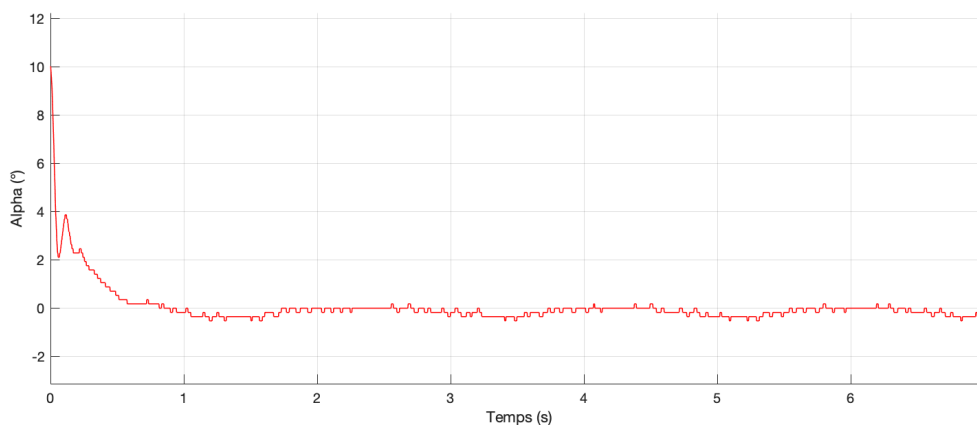


FIG. 3.11 : Évolution de l'angle  $\alpha$  pour un réseau ANFIS

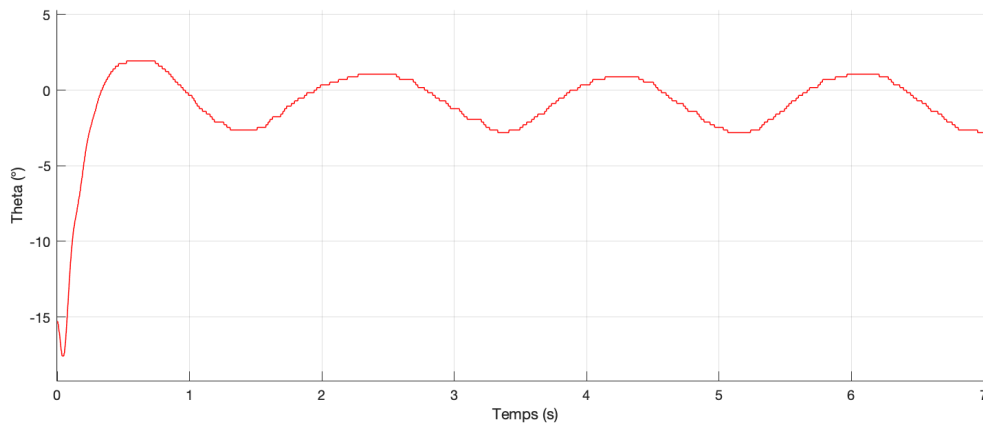


FIG. 3.12 : Évolution de l'angle  $\theta$  pour un réseau ANFIS

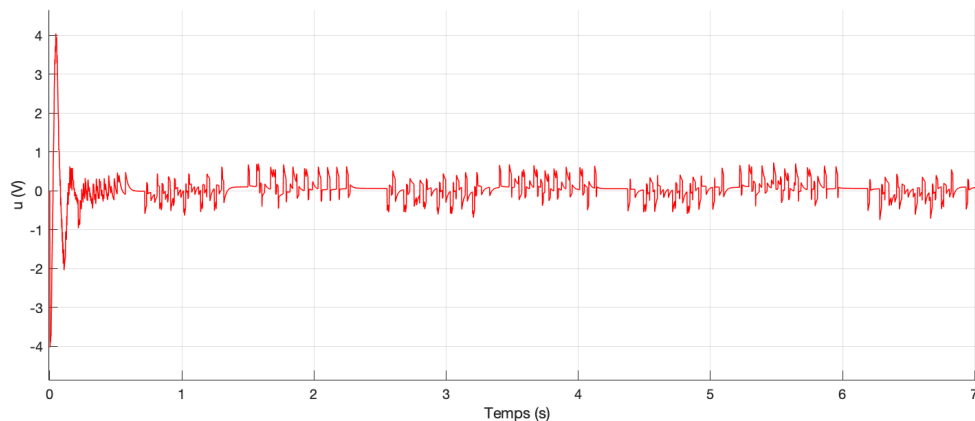


FIG. 3.13 : Évolution de la commande  $u$  pour un réseau ANFIS

Les résultats montrent que le contrôleur est un bon choix, ceci est interprétée par le temps de réponse réduit, le dépassement et l'erreur statique quasiment nuls. Les performances sont similaires aux contrôleurs précédents sauf qu'ici le temps d'apprentissage est beaucoup plus rapide : l'apprentissage atteint son erreur minimale en quelques secondes seulement.

#### 3.2.4 Commande linéaire à base de réseaux DDPG

La commande par approche DDPG n'a pas besoin de base de données car elle utilise un apprentissage par renforcement. Les deux réseaux de neurones "critique" et "acteur" ont été conçus après plusieurs essais. (Fig. 3.14)

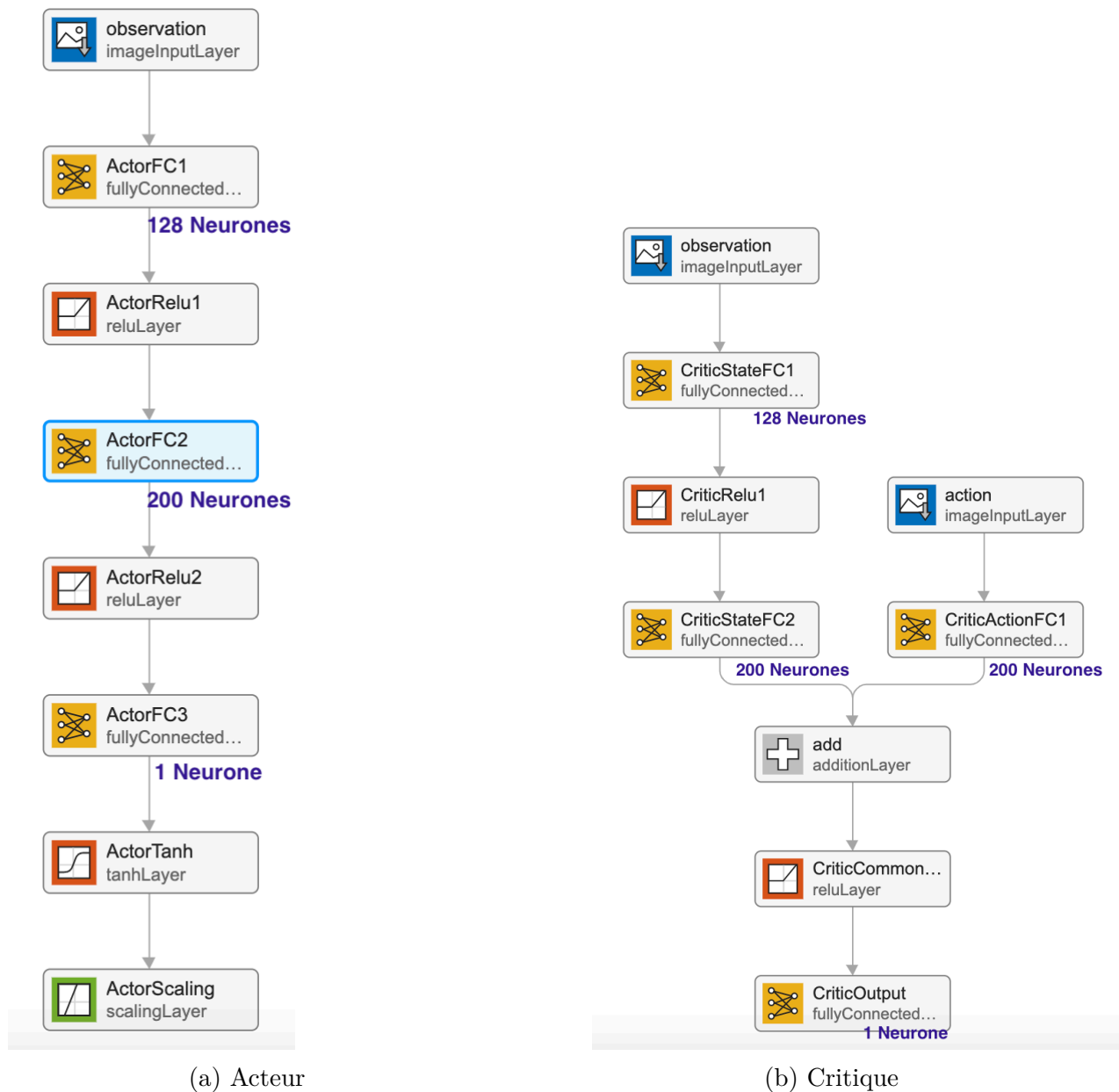


FIG. 3.14 : Structure des réseaux neuronaux "Critique" et "Acteur"

Les deux réseaux sont liés car le réseau "critique" a besoin du réseau "acteur" comme entrée. Le réseau acteur prend en compte les quatre états comme entrées, il possède deux couches cachées de 128 et 200 neurones respectivement, ainsi qu'une couche de sortie à un seul neurone. Le réseau critique quant à lui prend en compte les quatre états du système ainsi que la sortie du réseau acteur comme entrées. Il possède la particularité d'ajouter deux couches cachées en parallèle pour former la couche de sortie du réseau.

Le tableau (3.1) indique les paramètres utilisés lors de l'apprentissage.

Paramètre	Valeur
Temps d'échantillonnage	0.01
Bruit	1
Taux de décroissance du bruit	$2 * 10^{-4}$
Nombre d'épisodes	1000
Échantillons par épisode	200

TAB. 3.1 : Paramètres de l'apprentissage DDPG

La fonction de récompense utilisée est montrée dans l'équation suivante :

$$r = - (10\theta^2 + 20\alpha^2 + 1\dot{\alpha}^2 + 0.1u^2) - 100 (|\theta| > 60^\circ \parallel |\alpha| > 10^\circ \parallel |u| > 5V) \quad (3.1)$$

Pour faciliter l'apprentissage, nous avons également introduit des critères d'arrêt qui permettent d'arrêter l'épisode en cours lorsque certaines conditions sont dépassées :  $|\theta| > 60^\circ$  ou  $|\alpha| > 10^\circ$  ou  $|u| > 5V$ . La figure (3.15) représente l'évolution de la fonction de récompense lors de l'apprentissage, où on remarque qu'il passe par deux étapes bien distinctes.

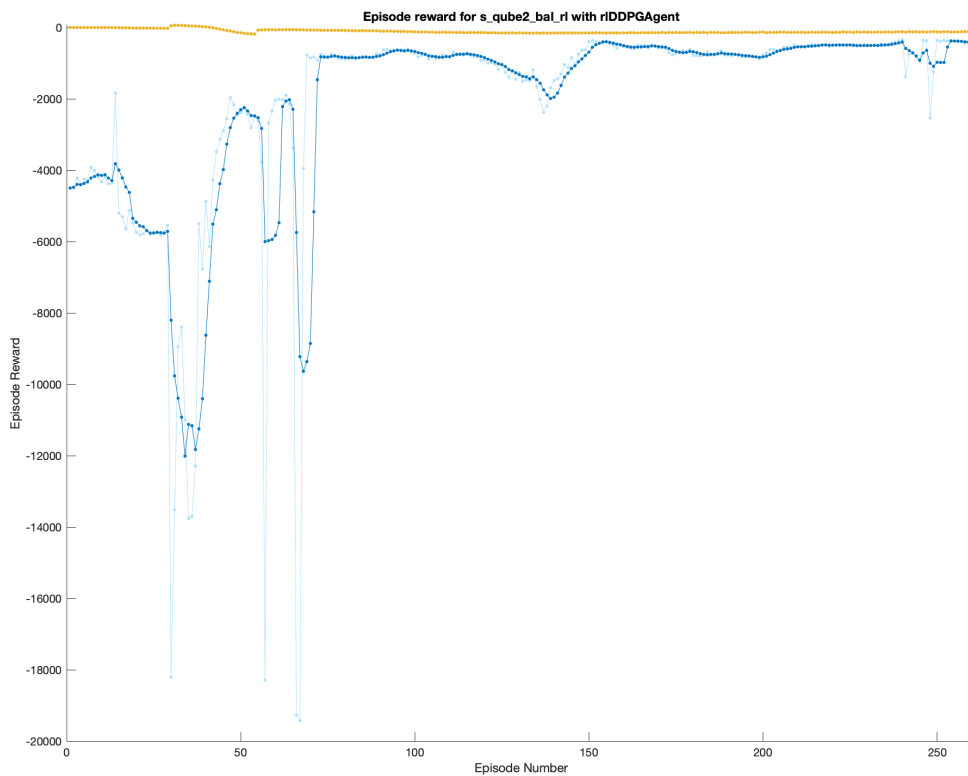


FIG. 3.15 : Apprentissage par renforcement DDPG

### Chapitre 3. Conception des contrôleurs à base de RNA (Application au pendule inversé rotatif)

La première étape est la phase d'exploration où l'algorithme essaie de diversifier ses zones de recherche pour ne pas tomber dans le piège de l'optimum local, avec une durée d'environ 100 épisodes. Vient ensuite la phase d'exploitation où il cherche à intensifier son apprentissage dans une même zone pour trouver un meilleur résultat.

L'agent a donc été entraîné jusqu'à ce qu'il atteigne une récompense d'environ -300. On remarque que seulement 200 épisodes ont suffi à atteindre cette récompense satisfaisante. Cela a servi à apprendre à l'agent à stabiliser le pendule.

La figure (3.16) montre la structure de l'algorithme sous SIMULINK. On peut constater la présence de l'agent qui se sert de l'observation des états ainsi que de la fonction de récompense pour générer une action. Cette dernière est directement injectée dans le modèle du PIR.

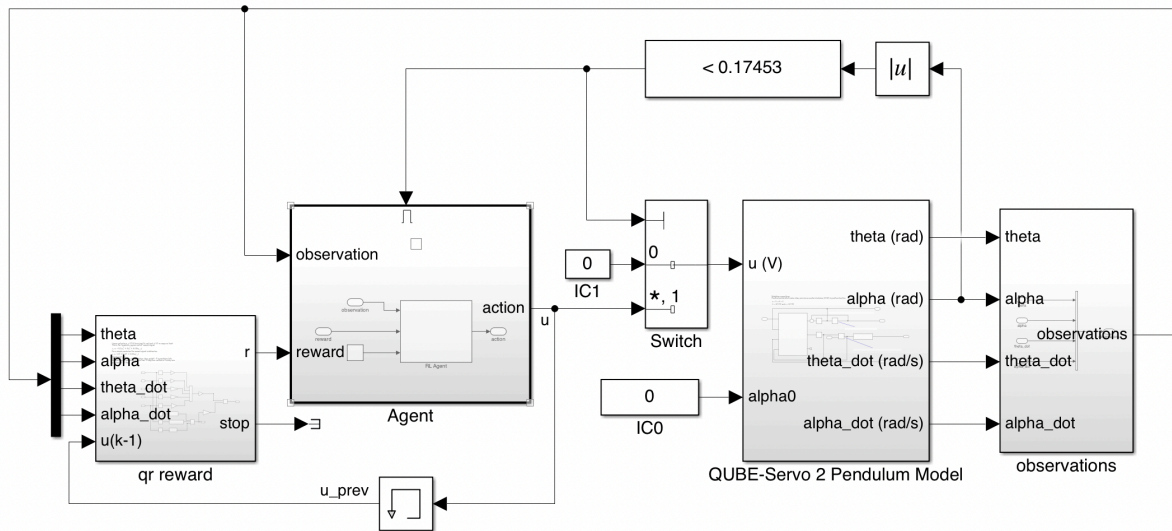


FIG. 3.16 : Schéma sous SIMULINK de l'algorithme DDPG

Les figures (3.17, 3.18, 3.19) montrent les résultats de simulation du contrôleur DDPG.

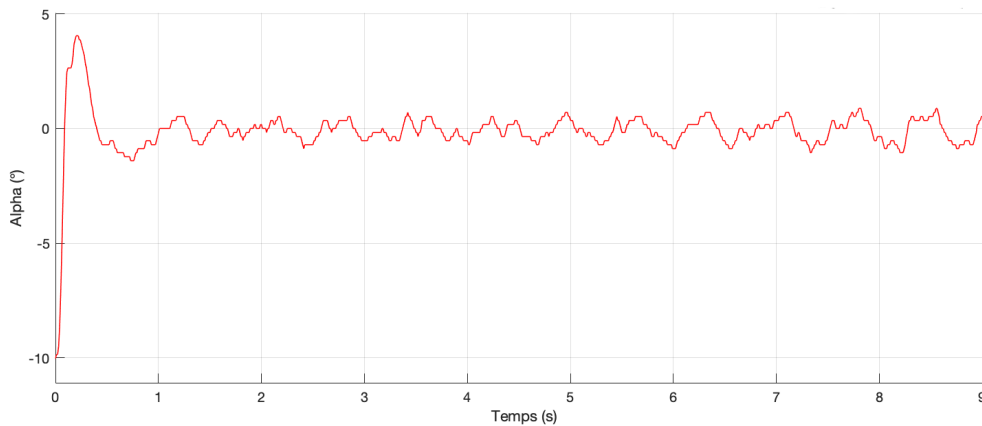


FIG. 3.17 : Évolution de l'angle  $\alpha$  pour un réseau DDPG

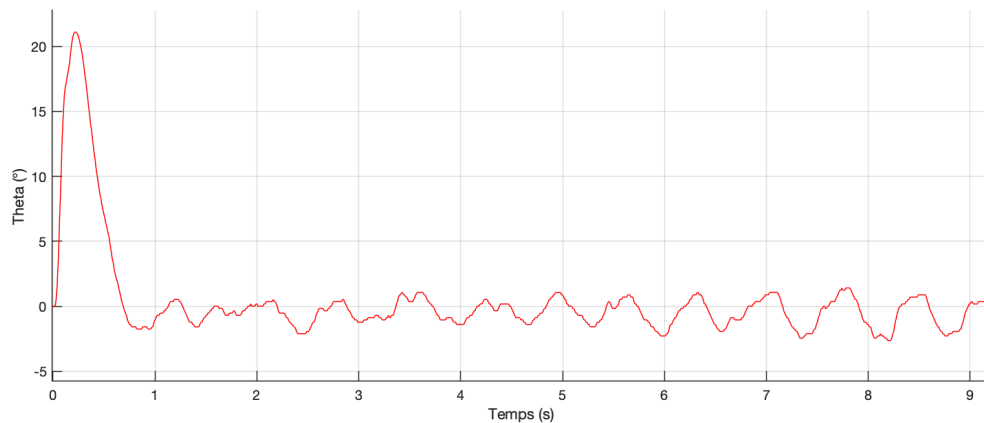


FIG. 3.18 : Évolution de l'angle  $\theta$  pour un réseau DDPG

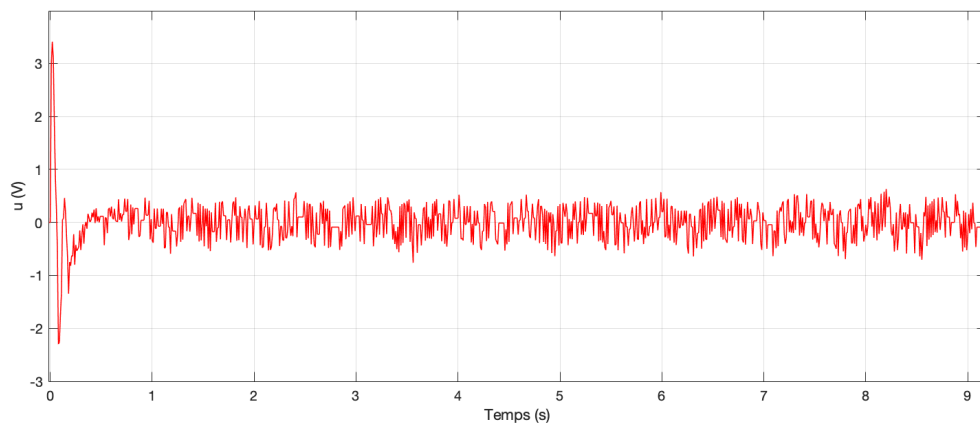


FIG. 3.19 : Évolution de la commande  $u$  pour un réseau DDPG

Les résultats de simulation montrent que le pendule inversé rotatif réussit à se stabiliser. On remarque un dépassement de  $4^\circ$  nettement plus grand que les autres commandes. Cependant, cette commande reste néanmoins performante.

### 3.2.5 Test de robustesse

Il est très important de tester la robustesse de nos commande, car en réalité notre milieu peut être exposé à des perturbations aléatoires (vent, coup de pousse d'un corps extérieur, ...) et cela peut dévier notre pendule de sa position d'équilibre instable. Nous avons donc essayé d'ajouter des impulsions à notre système au niveau de la commande qui agissent comme des perturbations, et ainsi voir le comportement de ce dernier. Les figures suivantes (3.20, 3.21) montrent les résultats concernant le test de robustesse.

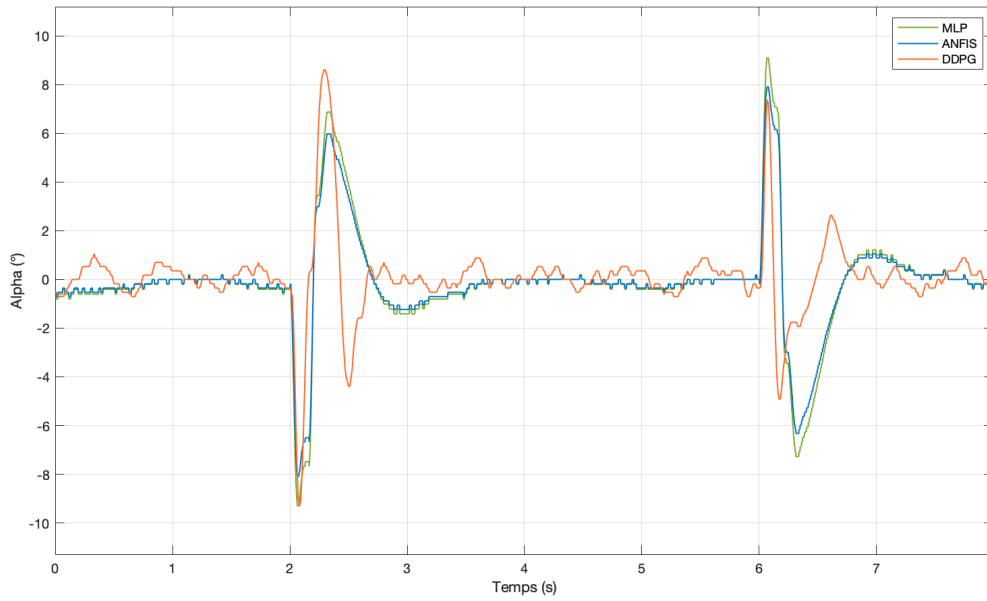


FIG. 3.20 : Test de robustesse MLP, ANFIS, DDPG

On peut constater que les commandes MLP, ANFIS et DDPG sont très robustes et donnent un comportement très satisfaisant malgré les perturbations. Le contrôleur DDPG montre un meilleur temps de réponse que les autres mais une erreur statique plus grande mais reste très performante.

Le contrôleur RBF (Fig. 3.21) quant à lui présente moins de robustesse. La commande réagit correctement uniquement face à des petites impulsions (1 Volt). Au-delà, la commande devient instable et le pendule tombe vers son point d'équilibre stable. Nous ne pouvons pas vraiment dire que la commande RBF a réussi au test de robustesse.

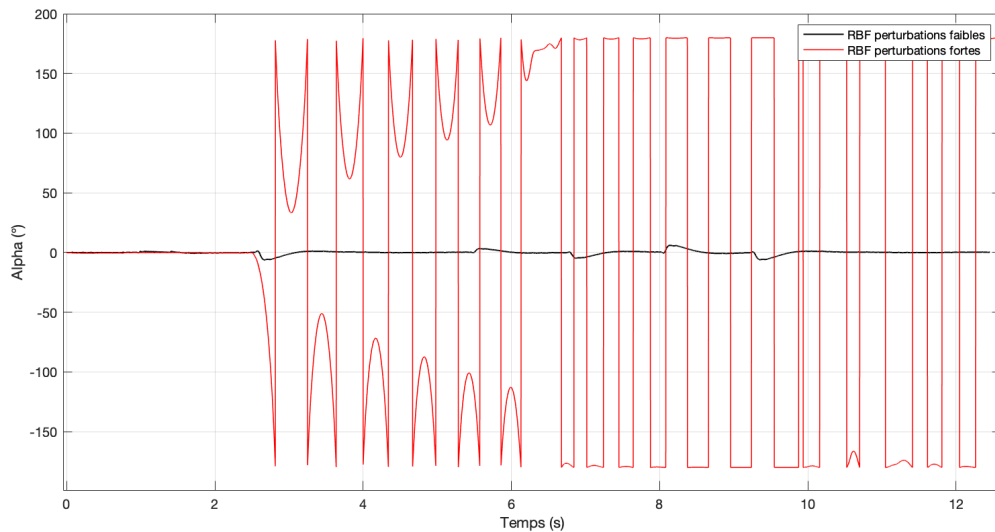


FIG. 3.21 : Test de robustesse pour un réseau RBF

### 3.3 Conception des commandes neuronales non linéaires

Maintenant que les tests linéaires ont prouvé leur efficacité, il est temps de se pencher vers des contrôleurs neuronales qui gèrent tout le contrôle neuronale du système non linéaire. La base de données utilisée est celle d'une commande hybride EBC+LQR, qui permet de faire monter le pendule de sa condition initiale ( $\alpha = \pm 180$ ) vers sa position d'équilibre instable ( $\alpha = 0$ ) et de le stabiliser [43].

#### 3.3.1 Commande non linéaire à base de réseaux MLP

Comme pour la commande linéaire, les mêmes étapes sont reproduites sauf la base de données différente. La topologie reste similaire : Le réseaux de neurones MLP contiendra une couche d'entrée de 4 neurones accueillant les quatre états, deux couches cachées de 10 et de 25 neurones respectivement, et une couche de sortie à neurone unique représentant la commande  $u$ . Les fonctions d'activations reste inchangés. Remarquons que nous avons simplement augmenter le nombre de neurones dans la deuxième couche cachée car le système doit effectuer une tache plus lourde.

L'erreur d'apprentissage est nettement plus conséquente que celle pour la commande linéaire, vu la taille de la base de données est plus importante. L'apprentissage atteindra une erreur de 0.038 à la 500ème itération. (Fig. 3.22)

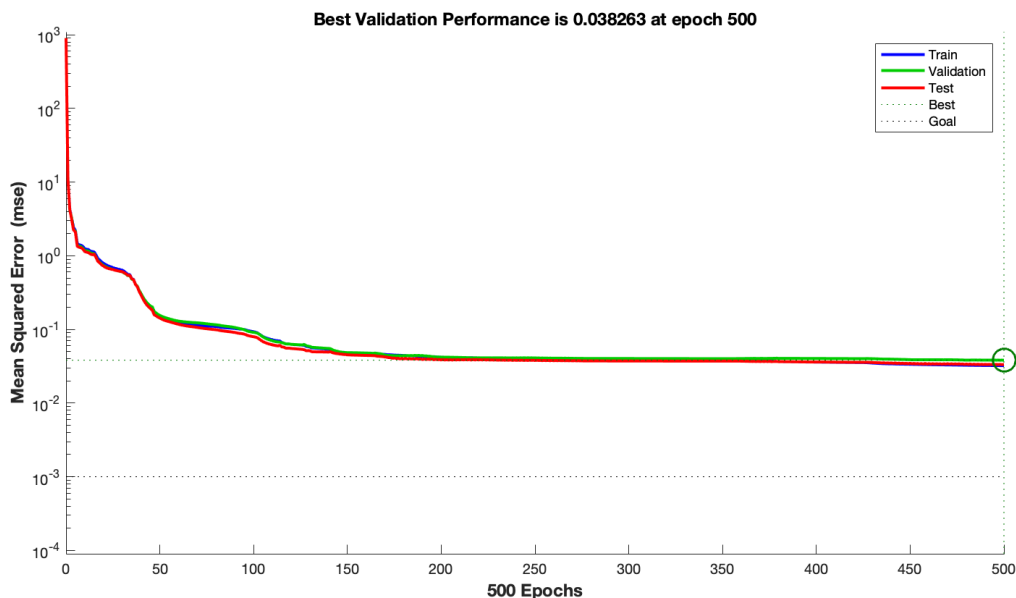


FIG. 3.22 : Apprentissage du réseau neuronal MLP



### Chapitre 3. Conception des contrôleurs à base de RNA (Application au pendule inversé rotatif)

---

Les figures (3.23), (3.24) et (3.25) montrent les résultats expérimentaux de la commande MLP non linéaire globale :

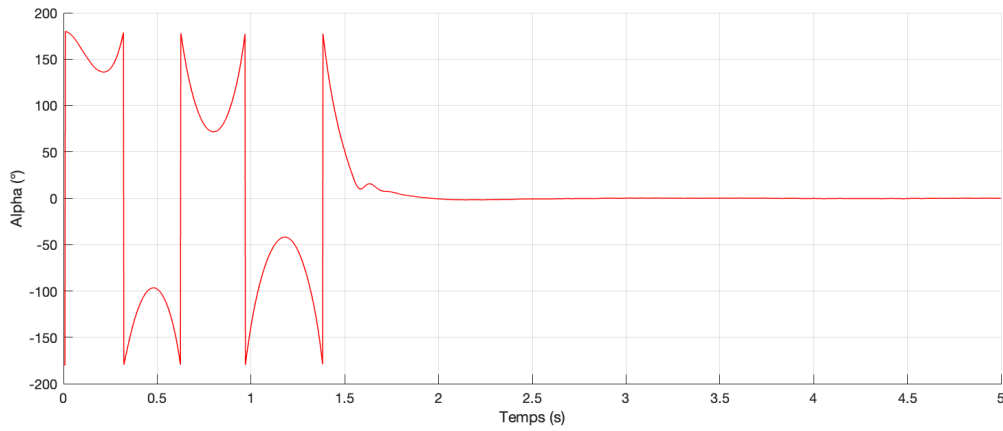


FIG. 3.23 : Évolution de l'angle  $\alpha$  pour un réseau MLP

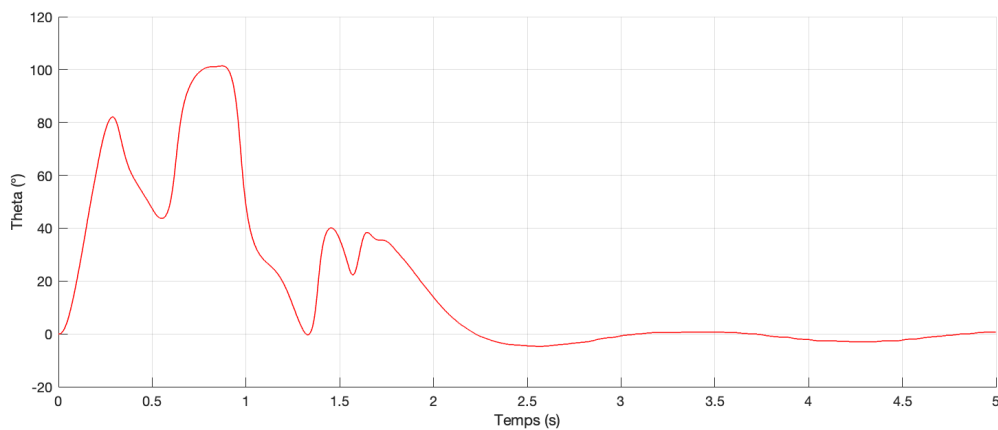


FIG. 3.24 : Évolution de l'angle  $\theta$  pour un réseau MLP

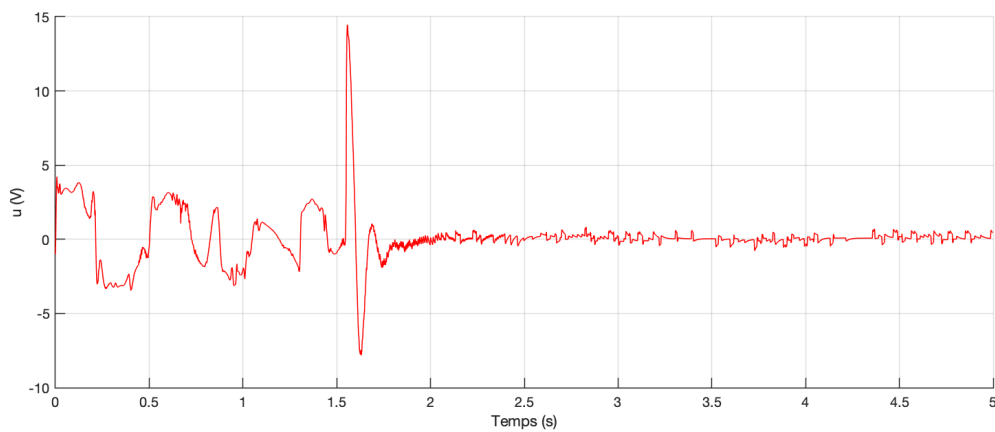


FIG. 3.25 : Évolution de la commande  $u$  pour un réseau MLP

## Chapitre 3. Conception des contrôleurs à base de RNA (Application au pendule inversé rotatif)

Les résultats montrent que la commande neuronale MLP est très performante : Le temps de réponse (2s) est très court en plus de l'erreur statique nulle. Les variables ne dépassent pas les limites physiques imposées du système.

### 3.3.2 Commande non linéaire à base de réseaux RBF

La commande RBF contient une seule couche cachée et ajoute à cette dernière des neurones progressivement lors de l'apprentissage supervisé. Pour atteindre une erreur acceptable et concevoir une commande performante, l'algorithme doit ajouter une quantité de neurones trop importante ( $>1000$  neurones) dépassant ainsi la limite du compilateur MATLAB, ce qui dépasse la capacité de la mémoire du PC.

Pour que la commande puisse être injectée dans le système, il faut concéder un nombre de neurones limité dans la couche cachée, ce qui donne comme résultat une erreur quadratique d'apprentissage trop importante et donc une commande instable. La commande RBF non linéaire n'est donc pas satisfaisante pour contrôler la stabilisation du pendule inversé rotatif (Fig. 3.26, 3.27, 3.28) :

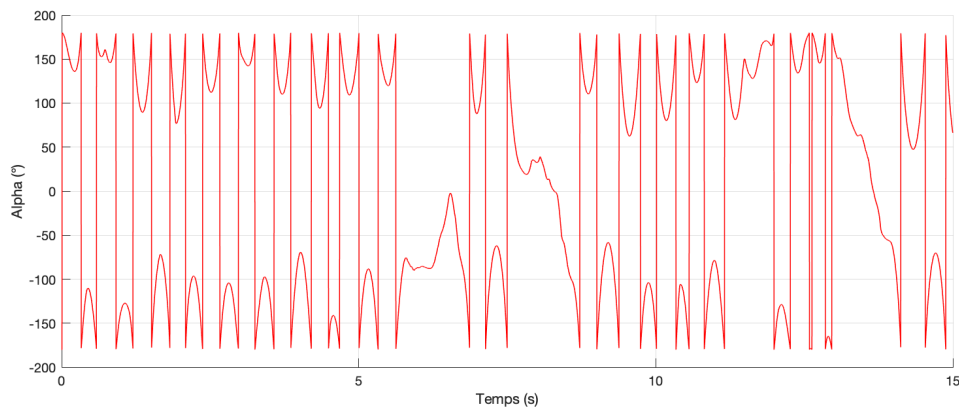


FIG. 3.26 : Évolution de l'angle  $\alpha$  pour un réseau RBF

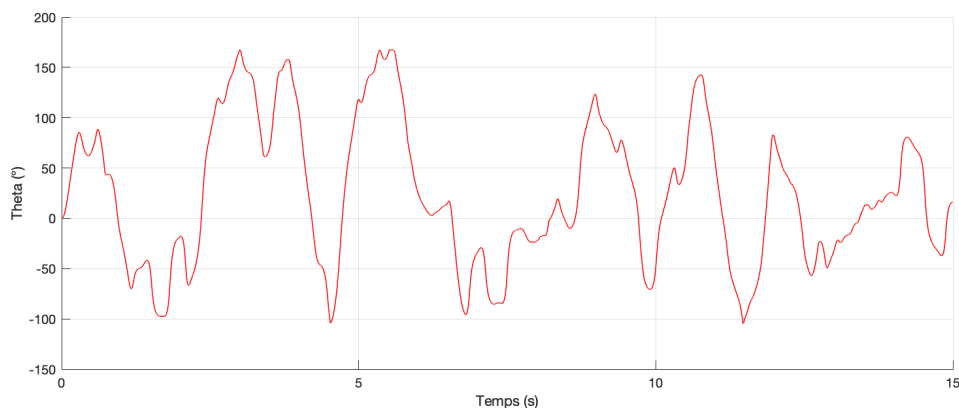


FIG. 3.27 : Évolution de l'angle  $\theta$  pour un réseau RBF

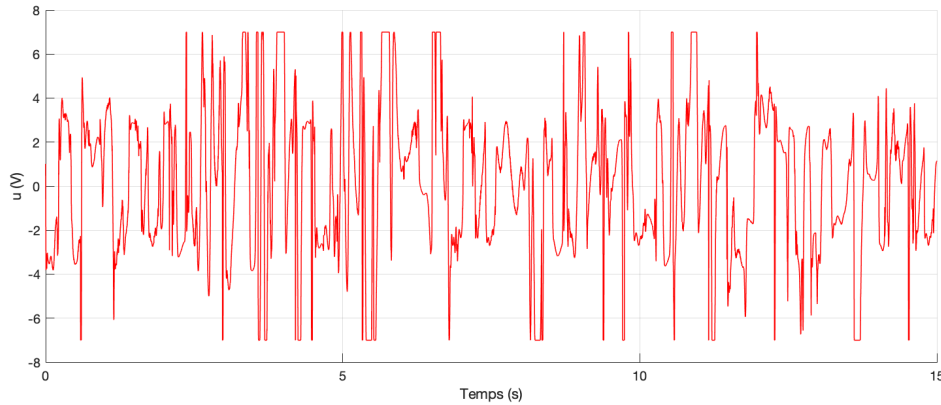


FIG. 3.28 : Évolution de la commande  $u$  pour un réseau RBF

### 3.3.3 Commande non linéaire à base de réseaux ANFIS

Le contrôleur neuro-flou ANFIS requiert une grande quantité de fonction d'appartenance pour assurer la commande non linéaire. De ce fait, nous avons décidé de concevoir une commande ANFIS hybride qui facilite l'apprentissage et réduit le temps de ce dernier.

En effet, nous avons conçu une commande ANFIS prenant en compte uniquement deux entrées (par souci de temps d'apprentissage) qui sont  $\alpha$  et  $\dot{\alpha}$ . Nous avons choisi 11 fonctions d'apprentissage de types gaussiennes par entrée pour avoir une erreur d'apprentissage raisonnable (Fig. 3.29, 3.30, 3.31). De plus, la toolbox **anfisedit** de MATLAB permet la conception d'une sortie de type linéaire et non pas constante. C'est à dire que :

$$u = p\alpha + q\dot{\alpha} + r$$

**-Exemple :**

Si  $\alpha$  est  $A_1$  et  $\dot{\alpha}$  est  $B_1$  alors  $u_1 = p_1\alpha + q_1\dot{\alpha} + r_1$

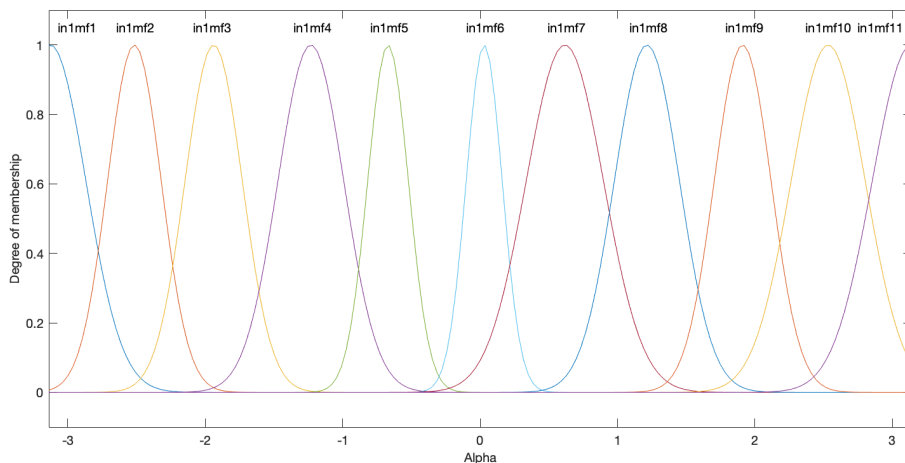


FIG. 3.29 : Fonctions d'appartenance  $\alpha$  pour un réseau ANFIS

### Chapitre 3. Conception des contrôleurs à base de RNA (Application au pendule inversé rotatif)

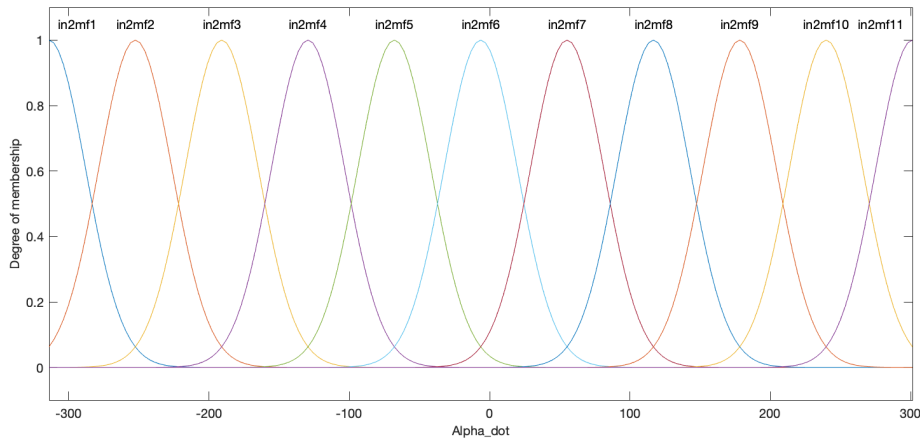


FIG. 3.30 : Fonctions d'appartenance  $\dot{\alpha}$  pour un réseau ANFIS

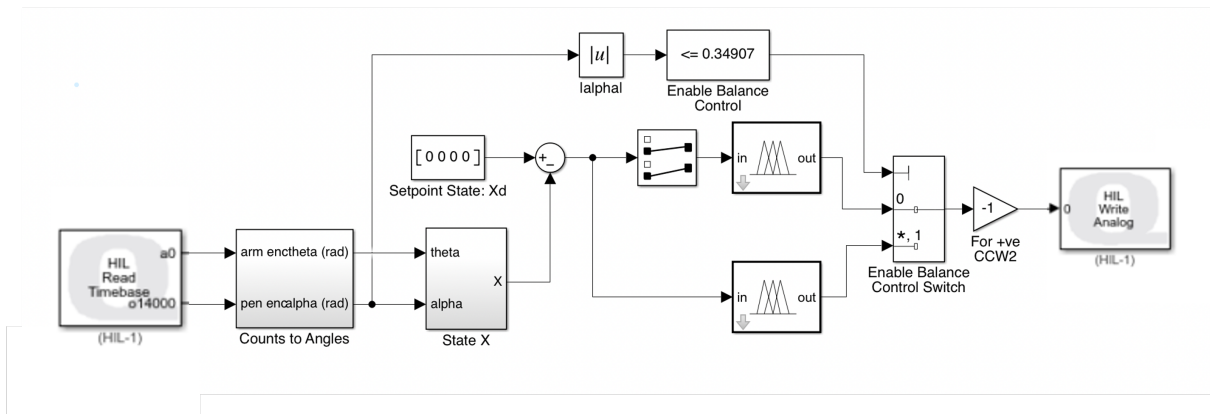


FIG. 3.31 : Schéma sous SIMULINK de la commande ANFIS hybride

Nous avons utilisé deux blocs "fuzzy", l'un contenant la commande ANFIS linéaire conçue précédemment et l'autre contenant la commande non linéaire que nous venons de voir, pour créer une commande hybride ANFIS. Les figures (3.32, 3.33 et 3.34) représentent les résultats expérimentaux :

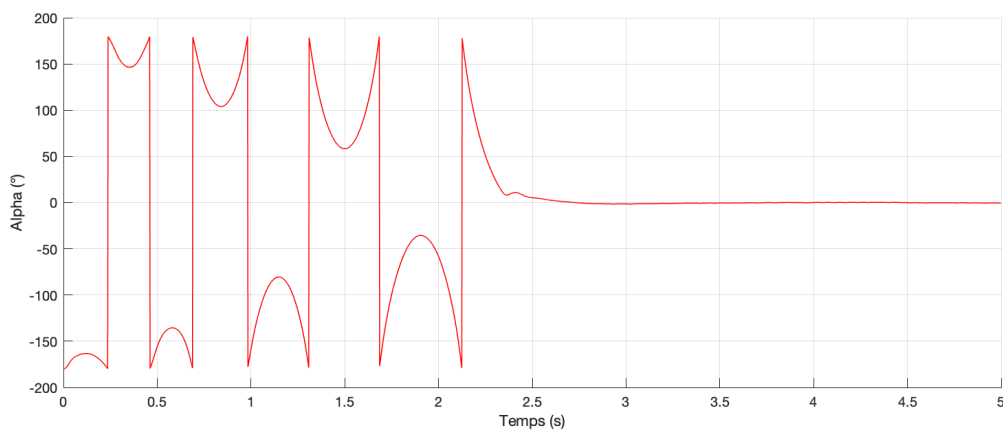


FIG. 3.32 : Évolution de l'angle  $\alpha$  pour un réseau ANFIS

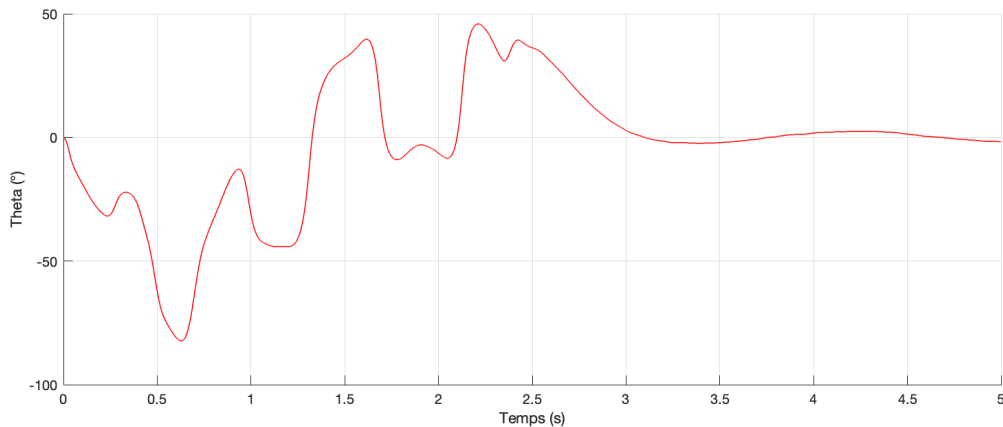


FIG. 3.33 : Évolution de l'angle  $\theta$  pour un réseau ANFIS

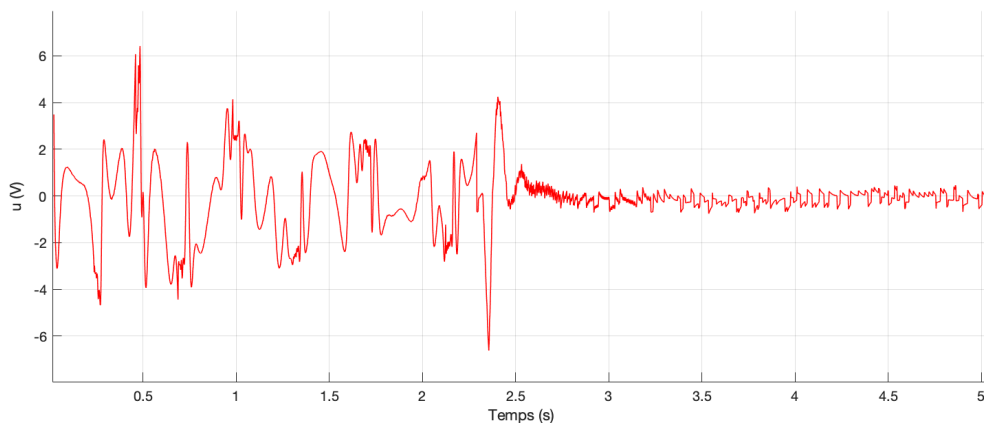


FIG. 3.34 : Évolution de la commande  $u$  pour un réseau ANFIS

Comparés à la commande MLP, ces résultats désignent une commande moins gourmande en alimentation (la commande ne dépasse pas les 6 Volts), mais un temps de réponse légèrement plus grand de l'ordre de 2.75s.

#### 3.3.4 Commande non linéaire à base de réseaux DDPG

La commande à base de réseaux DDPG non linéaire suit les mêmes étapes de conception. Les paramètres d'apprentissage seront les mêmes que ceux vu précédemment. Cependant, la fonction de récompense utilisée sera modifiée après de multiples essais et tentative, elle sera sous la forme suivante :

$$\begin{aligned}
 r = & -0.01 \left( 1\theta^2 + 5\alpha^2 + 0.05\dot{\alpha}^2 + 0.5u^2 \right) \\
 & -100 \left( |\theta| > 90^\circ \parallel |u| > 8V \right) \\
 & +10 \left( |\theta| < 20^\circ \parallel |\alpha| < 20^\circ \right) \\
 & +40 \left( |\theta| < 20^\circ \parallel |\alpha| < 20^\circ \parallel |u| < 2V \right)
 \end{aligned}
 \tag{3.2}$$

Nous avons également introduit des critères d'arrêt qui permettent d'arrêter l'épisode en cours lorsque les conditions suivantes sont dépassées :  $|\theta| > 90^\circ$  ou  $|u| > 8V$

L'algorithme possèdera également des réseaux de neurones différents pour assurer l'objectif de contrôle du pendule inversé rotatif. La structure du réseaux "critique" et celle du réseau "acteur" sont présentées dans la figure (3.35).

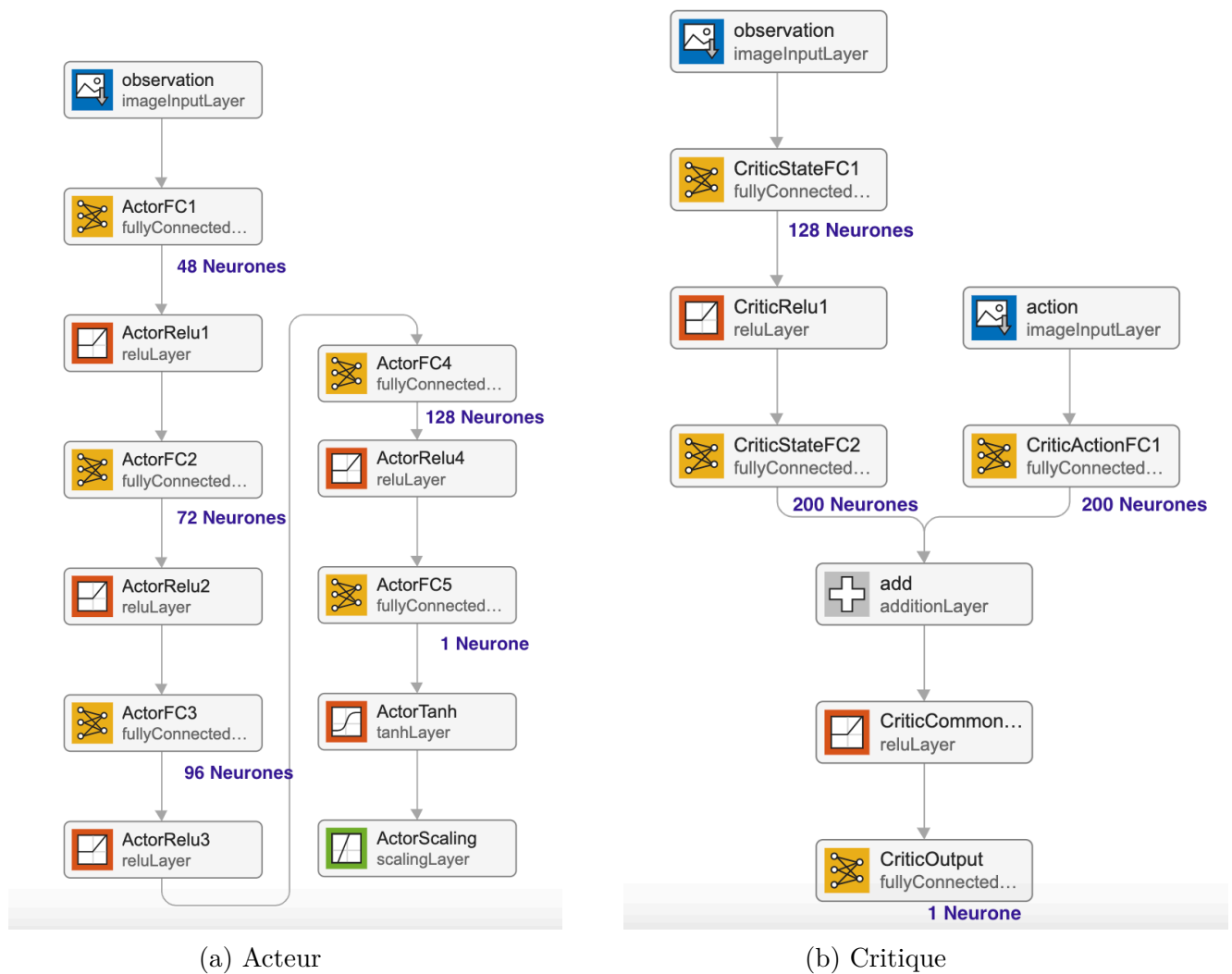


FIG. 3.35 : Structure des réseaux neuronaux "Critique" et "Acteur"

### Chapitre 3. Conception des contrôleurs à base de RNA (Application au pendule inversé rotatif)

Comme nous pouvons le constater, le réseaux de neurone "acteur" à été renforcé en ajoutant une couche cachée supplémentaire pour répartir les neurones. Nous pourrons voir par la suite qu'il produira une excellente performance d'apprentissage.

La figure (3.36) montre l'apprentissage effectué, mixant l'exploration et l'exploitation.

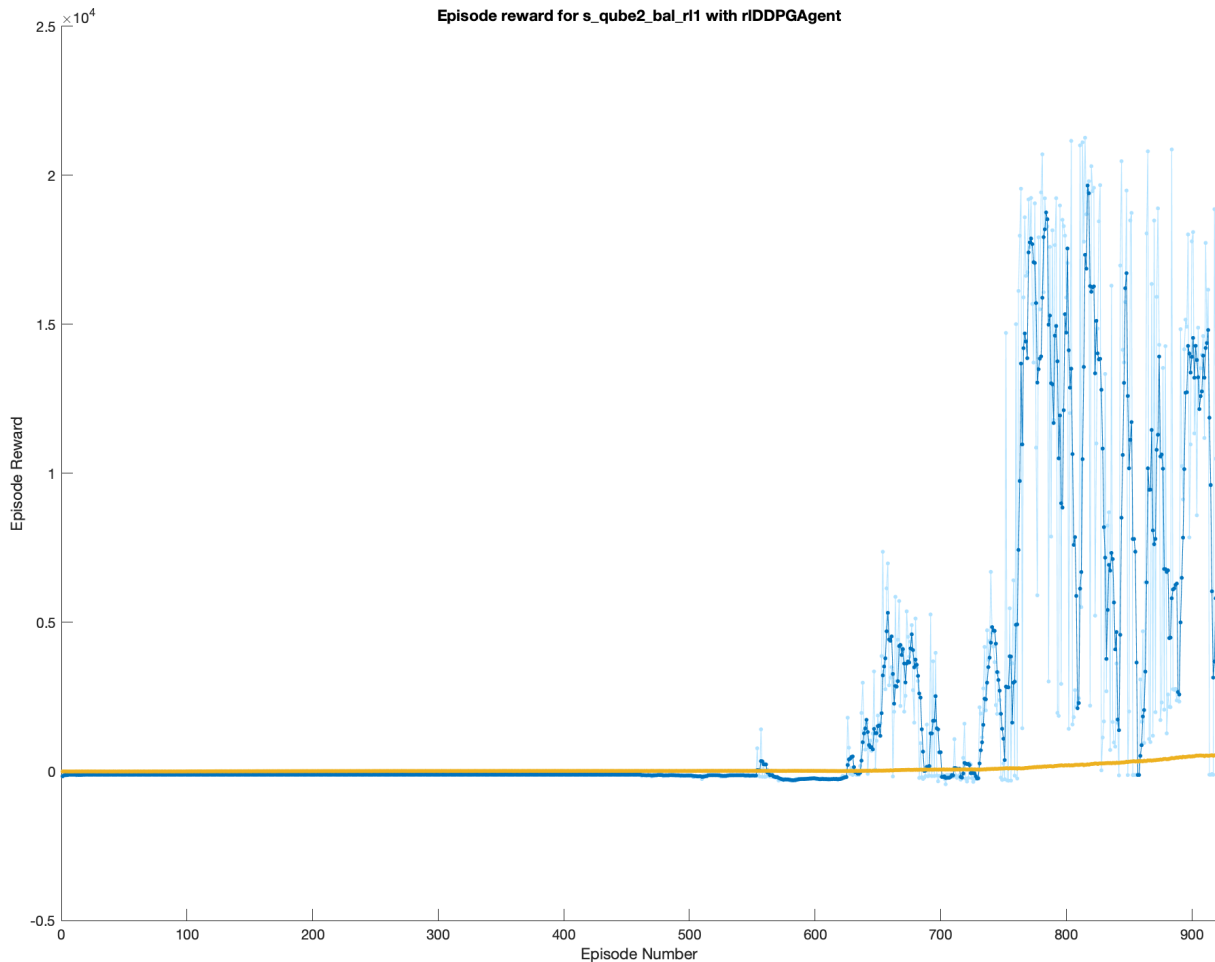


FIG. 3.36 : Apprentissage par renforcement pour un réseau DDPG

Nous constatons que l'algorithme DDPG a commencé à trouver des résultats de récompense satisfaisants seulement après 600 épisodes. Grâce à un mix d'exploration et d'exploitation, il a réussi à trouver un pic de récompense après 800 épisodes d'une valeur dépassant les  $2 * 10^4$ . On peut également remarquer que l'exploration de l'apprentissage continue pour essayer de trouver une meilleure récompense, sans réel succès : L'algorithme a déjà trouvé son optimum globale.

Les résultats de simulations suivants (Fig. 3.37, 3.38, 3.39) montrent le comportement des deux angles  $\alpha$ ,  $\theta$  ainsi que la commande  $u$

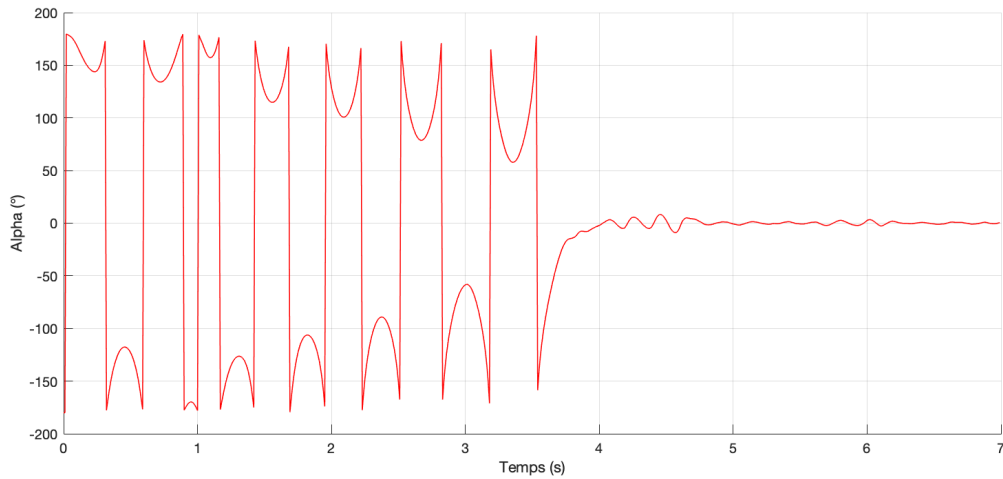


FIG. 3.37 : Évolution de l'angle  $\alpha$  pour un réseau DDPG

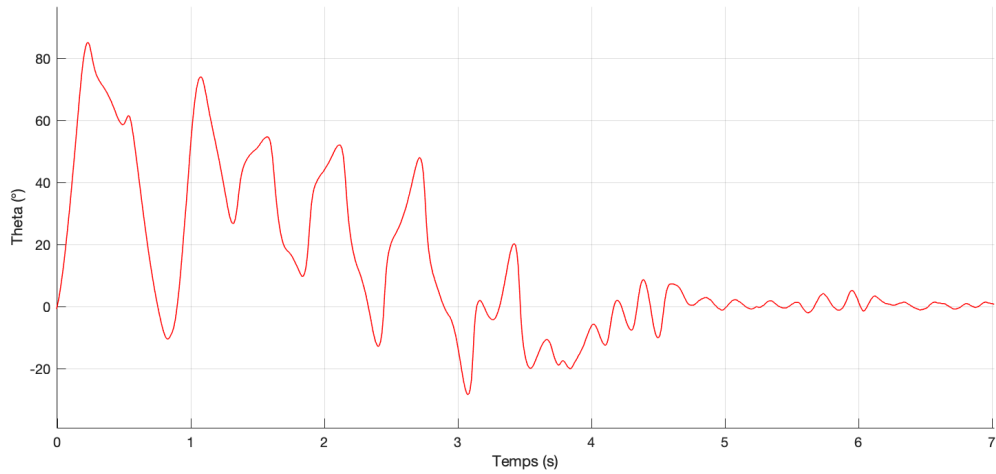


FIG. 3.38 : Évolution de l'angle  $\theta$  pour un réseau DDPG

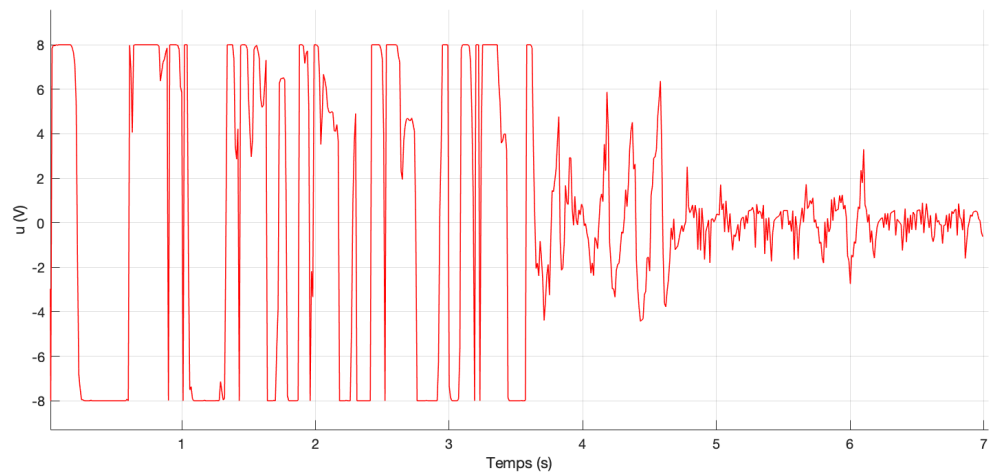


FIG. 3.39 : Évolution de la commande  $u$  pour un réseau DDPG



## Chapitre 3. Conception des contrôleurs à base de RNA (Application au pendule inversé rotatif)

---

Comparée à une commande neuronale supervisée, la commande neuronale par renforcement présente des oscillations dans le régime permanent. On remarque la présence de vibrations au niveau de l'angle  $\alpha$  d'environ  $\pm 5^\circ$  et de l'angle  $\theta$  d'environ  $\pm 8$  avec un temps de réponse 4 secondes.

Notons également que cette commande ne dépasse pas la barrière des  $\pm 8V$  comme souhaité. Il s'agit d'une excellente performance pour un algorithme à apprentissage par renforcement.

### 3.4 Étude comparative

Nous allons comparer les résultats des contrôleurs neuronaux non linéaires qui nous ont réussi à stabiliser le pendule inversé rotatif : C'est à dire les commandes à base de réseaux MLP, de réseaux ANFIS et de réseaux DDPG. Le but est d'essayer de voir les forces et les lacunes de chaque commande.

Les figures (3.40, 3.41, 3.42) montrent l'étude comparative des résultats expérimentaux.

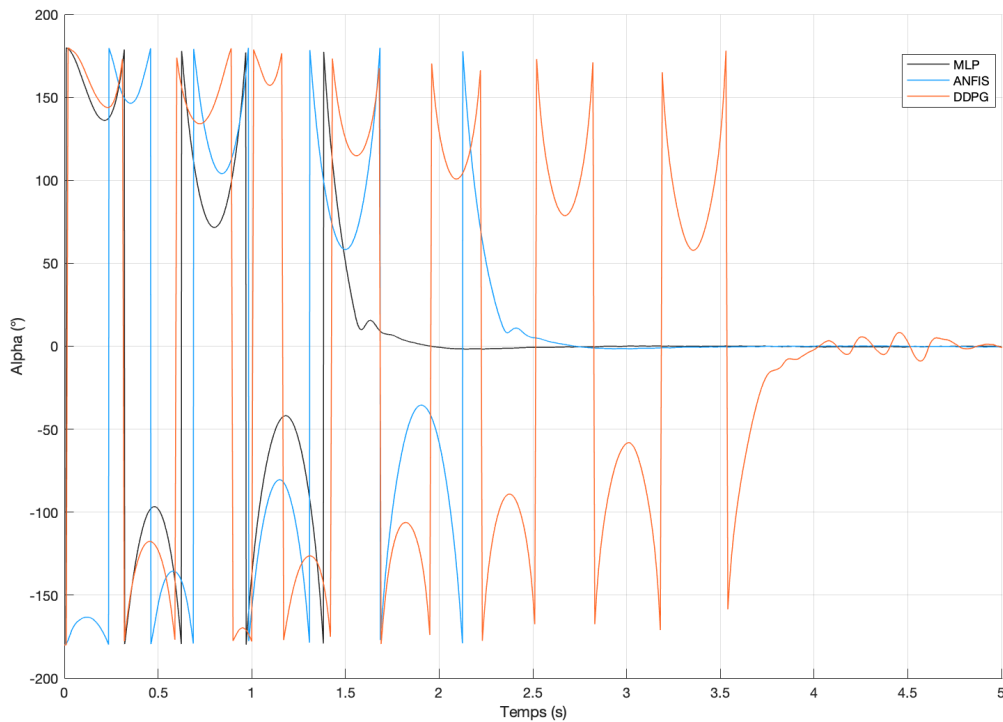


FIG. 3.40 : Évolution de l'angle  $\alpha$  pour l'étude comparative

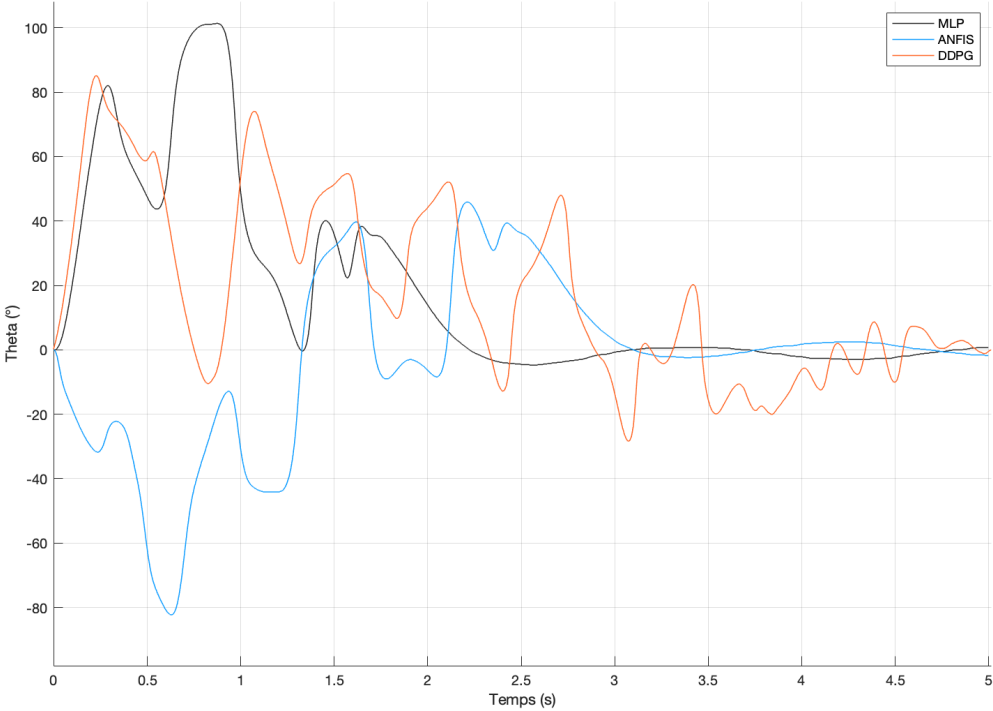


FIG. 3.41 : Évolution de l'angle  $\theta$  pour l'étude comparative

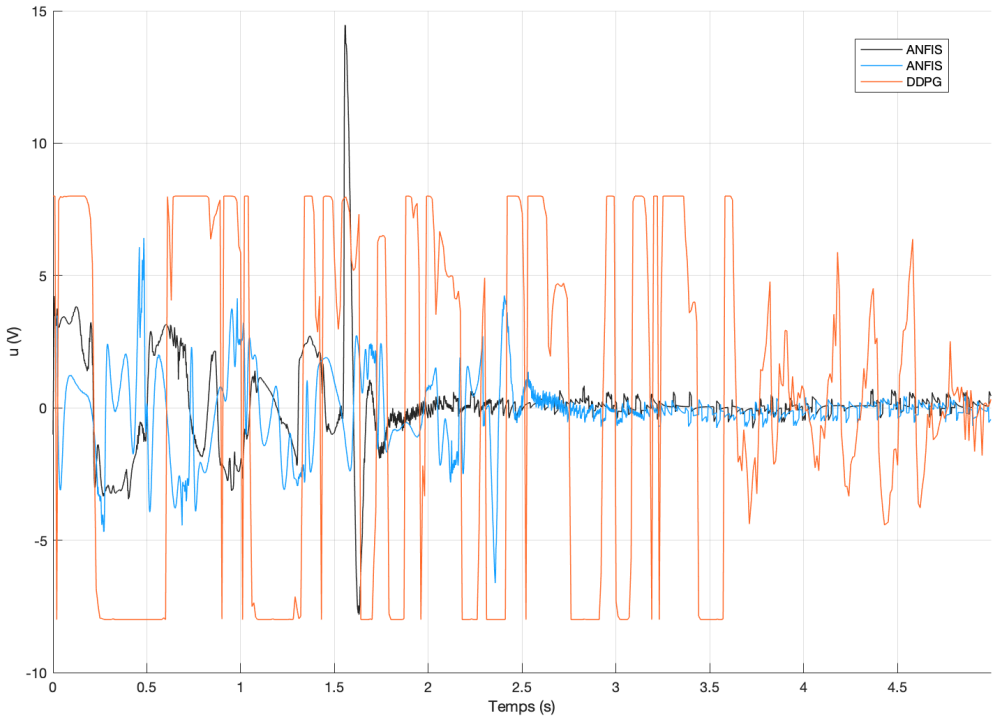


FIG. 3.42 : Évolution de la commande  $u$  pour l'étude comparative

Les trois commandes neuronales produisent d'excellents résultats. Remarquons que la commande MLP est la plus rapide mais est également celle qui possède le plus grand pic de commande ( $u \sim 14V$ ). La commande DDPG est celle qui produit le plus d'oscillations en régime permanent. On note que le temps de réponse est de l'ordre de 4s.

## 3.5 Conclusion

Les contrôleurs linéaires que nous avons vu confirment dans la globalité de bons résultats. Dans chaque cas, le mouvement du pendule est commandé vers la position d'équilibre instable voulue ( $|\alpha| = 0^\circ$ ), et le déplacement du bras reste dans les limites physiques ( $|\theta| < 120^\circ$ ). De plus, la commande n'est pas gourmande en temps de calcul, et la tension ne dépasse pas les limites matérielles notés dans la notice ( $|u| < 15V$ ).

Concernant les résultats des contrôleurs neuronaux non linéaire, il est clair que les réseaux de neurones artificiels proposent d'excellents résultats dans la plage non linéaire du pendule inversé rotatif. Il est clair que la présence d'une base de données au préalable facilite énormément les performances du système. Cependant, même si nous manquons de données, il est quand même possible de concevoir une commande à apprentissage par renforcement, qui présente des résultats très satisfaisants.

# Conclusion et perspectives

# Conclusion générale

L'étude présentée dans ce mémoire a eu pour objectif d'étudier et de comparer plusieurs types de commande neuronales, et de montrer qu'elles peuvent être utilisés pour commander des systèmes non-linéaires tel que le pendule inversé rotatif.

Les lois de commande proposées ont été conçues et appliquées à la stabilisation du PIR, qui est un outil didactique et un problème classique souvent utilisé pour tester les caractéristiques des nouvelles lois de commande.

Les résultats de notre étude comparative entre ces commandes confirment que la commande neuronale par apprentissage supervisé reste plus performante que celle par apprentissage par renforcement. Cependant, il n'est pas toujours évident d'avoir en sa possession une base de donnée d'entraînement efficace : C'est là que l'apprentissage par renforcement se démarque des autres.

En perspective il serait intéressant de poursuivre ce travail en approfondissant l'application de cette technique (méthode de transformation) sur un autre système à paramètres incertains (génie mécanique, énergétique).

# Perspectives

Les perspectives de cette étude offrent un aperçu des développements futurs et des opportunités de recherche dans le domaine de la commande par réseaux de neurones du pendule inversé rotatif, ainsi que dans le domaine plus large de l'automatisme des systèmes. Voici quelques perspectives qui peuvent être explorées :

- Amélioration de l'architecture du réseau de neurones : Bien que nous ayons exploré différentes architectures de réseaux de neurones, il reste encore de la place pour des améliorations. Des recherches supplémentaires peuvent être entreprises pour développer des architectures plus complexes et plus sophistiquées, telles que les réseaux neuronaux profonds, les réseaux de neurones récurrents ou les réseaux de neurones convolutifs, afin de mieux modéliser les comportements dynamiques du pendule inversé rotatif.
- Exploration de nouvelles techniques d'apprentissage par renforcement : Bien que nous ayons utilisé l'algorithme DDPG dans cette étude, il existe de nombreuses autres techniques d'apprentissage par renforcement qui peuvent être explorées. Des approches telles que l'apprentissage par renforcement profond (Deep Reinforcement Learning), l'apprentissage par renforcement hiérarchique ou l'apprentissage par renforcement inverse peuvent être étudiées pour améliorer l'efficacité et la stabilité de l'apprentissage du contrôleur.
- Application à d'autres systèmes : Les techniques de commande par réseaux de neurones peuvent être étendues à d'autres systèmes complexes et non linéaires. La méthodo-

logie utilisée dans cette étude peut être adaptée et appliquée à des domaines tels que la robotique, les systèmes de contrôle de processus industriels ou les systèmes de gestion de l'énergie, ouvrant ainsi de nouvelles perspectives de recherche et d'application.

## Appréciation personnelle

Au cours de cette étude, j'ai été captivé par les possibilités offertes par les réseaux de neurones dans le contrôle des systèmes dynamiques. J'ai pu constater leur puissance et leur capacité à modéliser des comportements complexes, tout en s'adaptant et en apprenant de manière autonome. Cela ouvre de nouvelles perspectives pour améliorer les performances des systèmes de contrôle et repousser les limites de ce qui était auparavant considéré comme impossible.

J'ai également été impressionné par la flexibilité et la polyvalence des différentes architectures de réseaux de neurones, telles que les réseaux MLP, RBF, neuro-flous et autres. Chacune de ces architectures offre des avantages spécifiques et peut être adaptée en fonction des caractéristiques et des exigences du système à contrôler. Cela démontre la richesse et la diversité des outils disponibles pour les ingénieurs en automatisme.

De plus, l'apprentissage par renforcement a été une révélation pour moi. La possibilité de former un agent à prendre des décisions optimales en interagissant avec son environnement, en maximisant une récompense, a ouvert de nouvelles perspectives dans le domaine du contrôle intelligent. Les concepts d'acteur-critique et les algorithmes tels que DDPG m'ont permis de comprendre et d'apprécier les mécanismes sous-jacents de l'apprentissage par renforcement.

En résumé, cette étude m'a permis d'approfondir mes connaissances en commande par réseaux de neurones, d'explorer de nouvelles techniques de contrôle et d'apprécier les perspectives futures de ce domaine en constante évolution. Je suis reconnaissant d'avoir eu l'occasion de contribuer à la recherche et au développement de solutions innovantes dans le domaine de l'automatisme des systèmes. Cette expérience a renforcé ma passion pour l'ingénierie et a stimulé mon désir de continuer à explorer de nouveaux horizons dans le domaine du contrôle intelligent.

# Bibliographie

- [1] MORCELI Youcef and LASKRI Mohamed Amine. Réalisation et commande d'un pendule inversé. *Mémoire de Master, Université de Blida*, 2022.
- [2] Petar Mandić, Mihailo P Lazarević, and Tomislav B Šekara. Fractional order pd control of furuta pendulum : D-decomposition approach. In *ICFDA'14 International Conference on Fractional Differentiation and Its Applications 2014*, pages 1–7. IEEE, 2014.
- [3] Ibrahim M Mehedi, Uzair Ansari, and Ubaid M AL-Saggaf. Three degrees of freedom rotary double inverted pendulum stabilization by using robust generalized dynamic inversion control : Design and experiments. *Journal of Vibration and Control*, pages pages 2174–2184, 2020.
- [4] Keigo Noguchi, Masaki Izutsu, Norihiro Kamamichi, Tetsuo Shiotsuki, Jun Ishikawa, and Katsuhisa Furuta. Swinging up and stabilization control of double furuta pendulums by safe manual control. In *2009 IEEE/RSJ International Conference on Intelligent Robots and Systems*, pages 4232–4237, 2009.
- [5] Zied Ben Hazem, Mohammad Javad Fotuhi, and Zafer Bingül. A comparative study of the friction models with adaptive coefficients for a rotary triple inverted pendulum. In *2018 6th International Conference on Control Engineering Information Technology (CEIT)*, pages 1–6, 2018.
- [6] Ammar Aljodah, Hassan Zargarzadeh, and Ts. Dr. Maythem Abbas Al-Adilee. Experimental verification and comparison of different stabilizing controllers for a rotary inverted pendulum. *Proceedings - IEEE International Conference on Control System, Computing and Engineering, ICCSCE 2013*, pages 417–423.
- [7] Yuxi Li. Deep reinforcement learning : An overview. *arXiv preprint*, 2017.
- [8] Tomohide Maeba, Mingcong Deng, Akira Yanou, and Tomohiro Henmi. Swing-up controller design for inverted pendulum by using energy control method based on lyapunov function. In *Proceedings of the 2010 International Conference on Modelling, Identification and Control*, pages 768–773. IEEE, 2010.

- [9] K.J. Åström and K. Furuta. Swinging up a pendulum by energy control. *Automatica*, 36:pages 287–295, 2000.
- [10] Fuat Peker, Ibrahim Kaya, Erdal Cokmez, and Serdal Atic. Cascade control approach for a cart inverted pendulum system using controller synthesis method. In *2018 26th Mediterranean Conference on Control and Automation (MED)*, pages 1–6, 2018.
- [11] Petar Mandić, Mihailo P. Lazarević, and Tomislav B. Šekara. Fractional order pd control of furuta pendulum : D-decomposition approach. In *ICFDA'14 International Conference on Fractional Differentiation and Its Applications 2014*, pages 1–7, 2014.
- [12] Iraj Hassanzadeh and Saleh Mobayen. Controller design for rotary inverted pendulum system using evolutionary algorithms. *Mathematical Problems in Engineering*, 2011.
- [13] Vishwa Nath and R Mitra. Swing-up and control of rotary inverted pendulum using pole placement with integrator. In *2014 Recent Advances in Engineering and Computational Sciences (RAECS)*, pages 1–5. IEEE, 2014.
- [14] Brian DO Anderson and John B Moore. *Optimal control : linear quadratic methods*. Courier Corporation, 2007.
- [15] Arturo Rojas-Moreno, Juan Hernandez-Garagatti, Oscar Pacheco-De La Vega, and Luis Lopez-Lozano. Fo based-lqr stabilization of the rotary inverted pendulum. In *2016 Chinese Control and Decision Conference (CCDC)*, pages 4292–4297, 2016.
- [16] Wei-chen Lee, Yue-Ying Lyu, Tzung-Yu Hsu, and Ching-chih Wei. Using the taguchi methods to study the balance control of a rotary inverted pendulum. In *2014 International Conference on Advanced Robotics and Intelligent Systems (ARIS)*, pages 171–175, 2014.
- [17] Slavka Jadlovska and Jan Sarnovsky. A complex overview of modeling and control of the rotary single inverted pendulum system. *Advances in Electrical and Electronic Engineering*, Volume 11(Number 2) :pages 73–85, 2013.
- [18] Etienne Videcoq, Manuel Girault, Kamélia Bouderbala, Hichem Nouira, José Salgado, and Daniel Petit. Parametric investigation of linear quadratic gaussian and model predictive control approaches for thermal regulation of a high precision geometric measurement machine. *Applied Thermal Engineering*, 78:pages 720–730, 2015.
- [19] N. Teja Chiluvuri, Omkar A. Harshe, Cameron D. Patterson, and William T. Baumann. Using heterogeneous computing to implement a trust isolated architecture for cyber-physical control systems. In *Proceedings of the 1st ACM Workshop on Cyber-Physical System Security, CPSS '15*, page 25–35. Association for Computing Machinery, 2015.



- [20] G Rigatos, P Siano, M Abbaszadeh, S Ademi, and A Melkikh. Nonlinear h-infinity control for underactuated systems : the furuta pendulum example. *International Journal of Dynamics and Control*, Volume 6:pages 835–847, 2018.
- [21] Mukhtar Fatihu Hamza, Hwa Jen Yap, and Imtiaz Ahmed Choudhury. Genetic algorithm and particle swarm optimization based cascade interval type 2 fuzzy pd controller for rotary inverted pendulum system. *Mathematical Problems in Engineering*, 2015.
- [22] Benedikt Alt, Christoph Hartung, and Ferdinand Svaricek. Robust fuzzy cascade control revised : Application to the rotary inverted pendulum. In *2011 19th Mediterranean Conference on Control Automation (MED)*, pages 1472–1477, 2011.
- [23] Sung-Kwun Oh, Wook-Dong Kim, and Witold Pedrycz. Design of optimized cascade fuzzy controller based on differential evolution : Simulation studies and practical insights. *Engineering Applications of Artificial Intelligence*, 25:pages 520–532, 2012.
- [24] Jen-Hsing Li. Composite fuzzy control of a rotary inverted pendulum. In *2013 IEEE International Symposium on Industrial Electronics*, pages 1–5, 2013.
- [25] Moez Ul Hassan, Muhammad B. Kadri, and Imran Amin. Proficiency of fuzzy logic controller for stabilization of rotary inverted pendulum based on lqr mapping. In Francesco Masulli, Gabriella Pasi, and Ronald Yager, editors, *Fuzzy Logic and Applications*, pages 201–211, Cham, 2013. Springer International Publishing.
- [26] Ahmad Taher Azar and Fernando E. Serrano. *Adaptive Sliding Mode Control of the Furuta Pendulum*, pages 1–42. Springer International Publishing, Cham, 2015.
- [27] H. Ahangar-Asr, M. Mansouri, M. Teshnehlab, and A. R. Pazoki. A hybrid strategy for the control of rotary inverted pendulum. In *2011 International Conference on Electrical and Control Engineering*, pages 5656–5659, 2011.
- [28] Javier Moreno-Valenzuela, Carlos Aguilar-Avelar, Sergio A. Puga-Guzmán, and Víctor Santibáñez. Adaptive neural network control for the trajectory tracking of the furuta pendulum. *IEEE Transactions on Cybernetics*, 46:pages 3439–3452, 2016.
- [29] Benedetto Allotta, Luca Pugi, and Fabio Bartolini. Reinforcement neural network for the stabilization of a furuta pendulum. In Marco Ceccarelli, editor, *Proceedings of EUCOMES 08*, pages 287–294, Dordrecht, 2009. Springer Netherlands.
- [30] Jin Seok Noh, Geun Hyung Lee, and Seul Jung. Position control of a mobile inverted pendulum system using radial basis function network. *International Journal of Control, Automation and Systems*, 8:pages 157–162, February 2010.

- [31] Rahul Agrawal and R. Mitra. Adaptive neuro fuzzy inference structure controller for rotary inverted pendulum. In Aswatha Kumar M., Selvarani R., and T V Suresh Kumar, editors, *Proceedings of International Conference on Advances in Computing*, pages 1163–1170, New Delhi, 2012. Springer India.
- [32] Pierre Borne, Mohamed Benrejeb, and Joseph Haggège. *Les réseaux de neurones : présentation et applications*, volume 15. Editions OPHRYS, 2007.
- [33] Messaoud Mokhtari. *Commande adaptative des systèmes non linéaires par l’approche backstepping neuronale*. PhD thesis, Université de Batna 2, 2015.
- [34] MELIK Billel and Naima BOUHABZA. Commande par logique floue de type 2 et commande neuro-floue appliquées à un pendule inversé. *Mémoire de Master, Université de Blida*, 1, 2015.
- [35] Kistler WM Gerstner W. Spiking neuron models : Single neurons populations plasticity book. *Cambridge U.K : Cambridge University Press*, 2002.
- [36] Mohammed Said ACHBI. Commande tolérante aux défauts en utilisant les réseaux de neurones artificiels et les systèmes d’inférence floue. *Mémoire de Magister, Spécialité : Génie Electrique, Université Mohamed Khider-Biskra*, 2012.
- [37] Belhadjer Hakim and Sarouer Brahim. *Classification des images avec les réseaux de neurones convolutionnels*. PhD thesis, Université Mouloud Mammeri, 2018.
- [38] ACHBI Mohammed Said. *Commande tolérante aux défauts en utilisant les Réseaux de Neurones Artificiels et les Systèmes d’Inférence Floue*. PhD thesis, 2012.
- [39] Amrouche Sylia and Azouani Djillali. *Commande neuro floue du canal longitudinal d’un drone à voilure fixe*. PhD thesis, Université Mouloud Mammeri, 2018.
- [40] Jongchan Baek, Hayoung Jun, Jonghyuk Park, Hakjun Lee, and Soohye Han. Sparse variational deterministic policy gradient for continuous real-time control. *IEEE Transactions on Industrial Electronics*, 68:pages 9800–9810, 2020.
- [41] Chayoung Kim. Temporal consistency-based loss function for both deep q-networks and deep deterministic policy gradients for continuous actions. *Symmetry*, 13:page 2411, 2021.
- [42] Tianbing Xu, Qiang Liu, Liang Zhao, and Jian Peng. Learning to explore via meta-policy gradient. *Book : International Conference on Machine Learning*, pages 5463–5472, 2018.

- [43] Mahieddine AYAD. Commande en temps réel d'un pendule inversé rotatif par la logique floue. *Mémoire d'Ingénieur, Spécialité : Automatique, École supérieure en sciences appliquées de Tlemcen*, 2023.
- [44] Matlab neural network toolbox documentation. MathWorks Inc.