

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



Faculté des Sciences Technologies
Département de génie Electrique
Filière : Electrotechnique
Spécialité : Electrotechnique industrielle

Projet de fin d'études
En vue de l'obtention du diplôme

MASTER

Thème

**Synthèse de commande intelligente basée sur
l'algorithme ABC appliquée à la MAS identifiée
par un RNA**

Présenté par :

BENZOUAD Aymane

KRIMI Fahim

Devant le jury composé de :

President M^r. N. TAIB

Promoteur M^r. A. MELAHI

Examineur M^r. R. SAOU

Année universitaire : 2022 / 2023

Remerciements

*Louange à Dieu le Tout Puissant de nous avoir donné le courage et la patience
de réaliser ce travail.*

*Nous sincères considérations et remerciements aux membres du jury
pour nous avoir fait l'honneur d'avoir accepté de faire partie du notre jury et de nous
donner de leur temps pour examiner ce travail*

*Nous tenons à exprimer notre profonde gratitude et nos sincères remerciements à notre
encadreur Mr MELAHI Votre sérieux, votre compétence et votre sens du devoir nous a
énormément marqué. Veuillez trouver ici l'expression de notre respectueuse considération
et notre profonde admiration pour toutes vos qualités scientifiques et humaines.*

*A mon cher ami HAMZA plus particulièrement pour sa compréhension, son soutien
moral et son aide, tout au cours de cette période, merci infiniment.*

*Nous plus vive considération va également à tous nous enseignants qui nous ont formé
tout au long de notre parcours universitaire.*

Merci à tous ce qui ont contribué de loin ou de prêt à la réalisation de ce mémoire.

Fahim et Aymane

Dédicace

J'ai le plaisir de dédier ce modeste travail en signe de respect et de reconnaissance :

A mes très chers parents, pour votre amour, vos sacrifices, votre soutien et vos prières qui m'ont accompagné tous au long de mon parcours universitaire. Que Dieu vous protège.

A mes adorables sœurs, frères et beaux-frères pour votre amour, encouragement permanent.

A mes chers amis tout à son nom, merci pour votre présence et votre amitié.

Aymanne

Dédicace

J'ai le plaisir de dédier ce modeste travail en signe de respect et de reconnaissance :

A Dieu, tout puissant, de m'avoir donné la force pour survivre, ainsi que l'audace pour dépasser toutes les difficultés et réaliser ce travail.

A ma chère mère pour son amour, son soutien, son encouragement et pour tous ce qu'elle m'a apporté de belles choses au cours de ma vie. Que Dieu vous protège.

*A la mémoire de mon père avec lequel je n'aurais pas le plaisir de partager cet événement, mais qui est et qui demeurera dans mon cœur à tous jamais.
Que Dieu l'accueille en son vaste paradis.*

A la mémoire de mes grands-parents. Que Dieu les accueille en son vaste paradis.

A mes chers frères : Abd-Elwahab et Fares. A mes adorables sœurs : Zineb, Wahiba, Betitra, leïla et Radia. Votre présence m'est indispensable, je vous souhaite beaucoup de bonheurs et de succès.

A mon grand frère Abd-Elwahab qui a été plus qu'un frère. Votre amour et votre soutien intarissables m'aident chaque jour. Que Dieu vous protège.

A mon frère FARES plus particulièrement pour sa compréhension, son soutien moral et son aide, tout au cours de cette période, merci infiniment.

A ma belle-sœur Lamia celle qui a rendu notre famille plus proche et plus aimé sans oublier mes nièces et mes neveux,

A mes grands-parents, que Dieu vous accorde une longue vie.

A mes tantes, oncles, cousins, cousines ainsi toute la grande famille.

A mon super et adorable binôme Aymen ainsi que sa famille.

A mes chers amis tout à son nom, merci pour votre présence et votre amitié

Fahim

Sommaire

Liste des abréviations

Liste des tableaux

Liste des figures

Liste des annexes

Introduction générale.....1

Chapitre I : Modélisation de la machine asynchrone.....2

I.1.Introduction.....2

I.2. Présentation et constitution de la machine asynchrone.....2

I.3. Principe de fonctionnement d'une machine asynchrone.....3

I.4. Modélisation de la machine asynchrone.....4

I.5. Différents référentiels.....12

I.6. Représentation d'état de la machine asynchrone.....13

I.7. Simulation de la machine asynchrone sous Matlab /Simulink.....14

I.8. Conclusion.....15

Chapitre II : Intelligence artificielle.....16

II.1. Introduction.....16

II.2. Les réseaux de neurones (RNA).....16

II.3. L'algorithme de colonies d'abeilles artificielles.....23

II.4. Conclusion.....28

Chapitre III : Application de la commande sur la machine asynchrone29

III.1. Introduction.....29

III.2. Simulation numérique.....29

III.3. Modélisation de la machine asynchrone par les réseaux de neurones.....31

III.4. Application de la commande prédictive basée sur les réseaux de neurones et l'algorithme ABC.....41

III.5. Conclusion.....46

Conclusion générale.....47

Références bibliographiques.....48

Table des matières

Annexes

Notations symboliques et abréviations

ω	Vitesse électrique du rotor [rad/s]
ψ	Fonction d'activation de type sigmoïde
σ	Coefficient de dispersion
θ	Position angulaire de rotor par rapport au stator [rad]
θ	Angle électrique entre l'axe A et l'axe d [rad]
γ	Coefficient de Blondel
s, r	Indice relatif au stator et rotor respectivement
q	Indice de l'axe quadratique
p	Nombre de pair de pole
i	Courant [A]
g	Glissement
f	Coefficient de frottement $kg.m^2/s$
d	Indice de l'axe direct
V	Tension [V]
SN	Sources de nourriture
M	Inductance mutuelle propre [H]
J	Moment d'inertie [$kg.m^2$]
H	Nombre d'itérations totale
Ω	Vitesse mécanique du rotor [rad/s]
SLP	Le perceptron monocouche
RNA	Réseaux de neurones artificiel
RBF	Réseaux de neurones à base radiale
MLP	Perceptron multicouche
MAS	Machine asynchrone
I_{sq}, I_{rq}	Courant statorique et rotorique quadratique [A]
h	Taille de l'intervalle de temps.
ART	Théorie de Résonance Adaptative

ABC	Artificial bee colony
ω_s	Pulsation du champ statorique [rad/s]
ω_r	Pulsation du champ rotorique [rad/s]
θ_r	Angle électrique entre l'axe a et l'axe d [rad]
z_i	Poids reliant la couche cachée à la couche de sortie
x_i	Entrées des neurones
w_i	Poids reliant la couche cachée à la couche d'entrée.
v_j	Terme constant appelé biais
v_j	Potentiel des neurones
p_j	Valeur de probabilité
l_s	Inductance propre d'enroulements statorique [H]
l_r	Inductance propre d'enroulements rotorique [H]
$i_{sa,b,c}$ et $i_{ra,b,c}$	Courants statoriques et rotoriques de phases [A]
b_i	Biais.
\hat{X}	Sortie de la machine asynchrone par réseau de neurones
$W_{j,i}$	Le poids de la connexion liant le neurone j à l'entrée i
V_{sq}, V_{rq}	Tension statorique et rotorique quadratique [V]
V_{sd}, V_{rd}	Tension statorique et rotorique directe [V]
$V_{sa,b,c}$ et $V_{ra,b,c}$	Tension de phase (stator rotor) [V]
T_r	Constante de temps rotorique
R_s	Résistance statorique [Ω]
R_r	Résistance rotorique [Ω]
$[P(\theta)]$	Matrice de Park
M_{sr}	Inductance mutuelle entre le stator et le rotor [H]
M_s	Inductance mutuelle statorique [H]
M_{rs}	Inductance mutuelle entre le rotor et le stator [H]
M_r	Inductance mutuelle rotorique [H]
L_s	Inductance statorique [H]
L_r	Inductance rotorique [H]

I_{sd}, I_{rq}	Courant statorique et rotorique directe [A]
C_r	Couple résistant [N.m]
C_{em}	Couple électromagnétique [N.m]
A_s, B_s, C_s	Axe de repère triphasé statorique
A_r, B_r, C_r	Axe de repère triphasé rotorique
Ω_s	Vitesse de synchronisme [rad/s]
Φ_{sq}, Φ_{rq}	Flux statorique et rotorique quadratique [Wb]
Φ_{sd}, Φ_{rd}	Flux statorique et rotorique directe [Wb]
Φ_s	Flux statorique [Wb]
Φ_r	Flux rotorique [Wb]
$[P(\theta)]^{-1}$	Matrice de Park inverse

Liste des tableaux

Tableau	Titre	Page
II.1	Différentes fonctions d'activations utilisées dans les RNA.	19
II.2	Pseudo code de l'algorithme ABC.	27

Liste des figures

Figure	Titre	Page
I.1	Représentation schématique d'une machine asynchrone triphasée.	4
I.2	Modèle d'une phase avec force électromotrice.	5
I.3	Représentation schématique du modèle biphasé équivalent.	8
I.4	Résultat de simulation en boucle ouverte de la MAS.	14
II.1	Neurone artificiel	18
II.2	Fonctions d'activations : (a) du neurone « seuil » ; (b) du neurone « linéaire », et (c) du neurone « Sigmoidé ».	18
II.3	Optimisation par colonies d'abeilles artificielles	24
III.1	Représentation graphique des résultats de la simulation du comportement de la machine asynchrone.	30
III.2	Modélisation de la machine asynchrone par réseaux de neurones.	31
III.3	Modèle neuronal de la machine asynchrone.	31
III.4	Courant statorique suivant l'axe d.	33
III.5	Erreur d'identification par MLP de Courant statoriques suivant l'axe d.	34
III.6	Courant statoriques suivant l'axe q.	34
III.7	Erreur d'identification par MLP de Courant statoriques suivant l'axe q.	35
III.8	La vitesse rotorique.	35
III.9	Erreur d'identification par MLP de la vitesse rotorique.	36
III.10.A	Vitesse rotorique en fonction de nombre d'itération (taille=10, H=10)	37
III.10. B	Vitesse rotorique en fonction de nombre d'itération (taille=10, H=10)	37
III.11. A	Vitesse rotorique en fonction de nombre d'itération (taille=10, H=30)	38

III.11. B	Vitesse rotorique en fonction de nombre d'itération (taille=10, H=30)	38
III.12. A	Vitesse rotorique en fonction de nombre d'itération (taille=100, H=10)	39
III.12. B	Vitesse rotorique en fonction de nombre d'itération (taille=100, H=10)	39
III.13. A	Vitesse rotorique en fonction de nombre d'itération (taille=100, H=100)	40
III.13. B	Vitesse rotorique en fonction de nombre d'itération (taille=100, H=100)	40
III.14	Structure de la commande prédictive via les algorithmes ABC	42
III.15	Schéma développé de l'algorithme ABC	43
III.16	Tension statorique direct	44
III.17	Tension statorique quadratique	44
III.18	Courant statorique direct	45
III.19	Courant statorique quadratique	45
III.20	Vitesse rotorique	46

INTRODUCTION
GENERALE

INTRODUCTION GENERALE

Les moteurs asynchrones sont des équipements électromécaniques largement utilisés dans de nombreuses applications industrielles en raison de leur efficacité, de leur simplicité de construction et de leur faible coût. Cependant, afin de tirer pleinement parti du potentiel des machines asynchrones et d'améliorer leurs performances, il est essentiel de développer des techniques de commande avancées [1], [2].

L'intelligence artificielle (IA) s'est ainsi révélée être un domaine prometteur pour optimiser le fonctionnement des systèmes électromécaniques.

Dans cette étude, nous explorons l'utilisation de l'intelligence artificielle, en particulier des réseaux de neurones et de l'algorithme ABC, pour développer une commande prédictive pour les machines asynchrones.

L'objectif principal de la commande prédictive est de prédire le comportement futur de la machine asynchrone en utilisant des modèles de prédiction basés sur les réseaux de neurones [3].

Les réseaux de neurones sont capables d'apprendre à partir de données historiques et de détecter des schémas et des tendances complexes. Ils peuvent donc être utilisés pour modéliser et prédire le comportement de la machine asynchrone en fonction de variables d'entrée telles que la tension d'alimentation et le courant et la vitesse.

L'algorithme ABC s'inspire du comportement des colonies d'abeilles pour résoudre des problèmes d'optimisation. Il explore de manière itérative l'espace des solutions possibles pour trouver les meilleurs paramètres de commande qui maximisent les performances de la machine.

Ce mémoire est organisé de la manière suivante :

Le premier chapitre est consacré à la machine asynchrone, en mettant le point sur son fonctionnement, sa structure, en plus, sa modélisation, qui repose sur des équations électromagnétiques et mécaniques complexes ainsi des hypothèses simplificatrices.

Le deuxième chapitre est consacré à l'intelligence artificielle où on se base sur les réseaux de neurones et l'algorithme ABC (Artificial Bee Colony).

Dans **le troisième chapitre**, nous commençons par la présentation de la commande prédictive par RNA et l'algorithme ABC pour les machines asynchrones avec des études de cas d'où les résultats expérimentaux présentés pour illustrer l'efficacité de cette approche.

On terminera par une conclusion générale.

CHAPITRE

I

I.1. Introduction

Les machines électriques ou machines tournantes sont des dispositifs électriques qui permettent de générer un mouvement ou une énergie mécanique à partir d'une énergie électrique et inversement. Ces dernières sont classées en trois catégories qui sont les machines à courant continu, les machines synchrones et les machines asynchrones. Les machines électriques sont caractérisées par leurs simplicités de conception, de fabrication, d'entretien, de robustesse et peu coûteuse avec un excellent rendement.

Dans ce chapitre, nous donnerons un aperçu sur la MAS suivie de son modèle biphasé lié au système d'axes liés aux champs tournant, la modélisation de cette machine en utilisant la transformation de Park. En se basant sur les hypothèses simplificatrices conventionnelles.

I.2. Présentation et constitution de la machine asynchrone

C'est une machine à courant alternatif pour laquelle la vitesse de rotation de l'arbre est différente de la vitesse de rotation du champ [4]. Elle est la plus utilisée dans le domaine des puissances supérieures à quelques kilowatts, car elle offre de nombreux avantages tels que sa simplicité de construction, sa grande fiabilité, son faible coût et sa facilité d'entretien...etc. [5].

Les moteurs asynchrones, souvent appelés moteurs à induction, se composent d'un stator et d'un rotor, constitués de tôles d'acier au silicium, et contiennent des fentes pour les enroulements.

✚ **Le stator**, est la partie fixe de la machine où est connectée l'alimentation électrique [6]. Il est constitué d'un empilage de tôles d'acier qui a la forme d'un cylindre. Le stator est fixe et encoché à l'intérieur. Il porte à l'intérieur des encoches un bobinage formant pair de pôles dont l'une des extrémités aboutit à la source et l'autre à une plaque à bornes avec laquelle on choisit le type de couplage : étoile ou triangle selon la tension de réseau du moteur.

✚ **Le rotor et les roulements**, partie tournante qui permet de mettre en rotation la charge mécanique. Le rotor est constitué d'une pile de tôles formant un cylindre plein ; ces tôles comportent à leurs périphéries extérieures des encoches destinées à recevoir les conducteurs. Un mince entrefer sépare le rotor du stator [7].

Il existe deux types de rotors, les bobinés d'où les conducteurs logés dans les encoches du rotor forment un enroulement triphasé, dont le nombre de pôles est le même que celui du stator. Le rotor à cage d'écureuil qui est constitué de barres métalliques identiques parallèles

ou non à l'axe du rotor. De chaque côté, les extrémités de ces barres sont réunies entre elles par des couronnes métalliques peu résistives. L'ensemble forme donc une cage, appelée cage d'écureuil [8].

I.3. Principe de fonctionnement d'une machine asynchrone

Le principe de fonctionnement d'un moteur asynchrone est basé sur l'interaction électromagnétique d'un champ magnétique tournant généré par le courant triphasé fourni par le réseau à l'enroulement du stator, et le courant induit dans l'enroulement du rotor est coupé par le champ tournant lorsque les conducteurs du rotor sont déconnectés [9].

Dans le stator, le courant génère un champ magnétique tournant. La fréquence statorique du courant détermine la fréquence de rotation de ce champ magnétique. La vitesse synchrone fait référence à la vitesse de rotation du champ tournant. Par conséquent, les changements de flux magnétique affectent les enroulements du rotor (**champ magnétique**).

Le courant rotorique est généré par la force électromotrice induite. L'apparition du couple, qui tend à déplacer le rotor pour contrecarrer les fluctuations de flux, est due à ces courants (loi de Lenz).

Le courant induit traverse le rotor court-circuité balayé par ce champ magnétique tournant. Le rotor est ainsi soumis à des forces de Laplace électromagnétiques qui se combinent pour générer le couple moteur entraînant le rotor en rotation. Bien que le rotor tourne dans le même sens que le champ magnétique tournant, sa vitesse doit être inférieure à celle du champ magnétique tournant.

La différence de vitesse du rotor par rapport à la vitesse du champ tournant permet de définir un paramètre caractérisant une machine asynchrone appelé glissement « g » exprimé en pourcentage et défini par :

$$g = \left(\frac{\Omega_s - \Omega_r}{\Omega_s} \right) \times 100 \% \quad (\text{I. 1})$$

Avec :

g : Glissement de la vitesse de rotation.

Ω_s : Vitesse de synchronisme en radians par seconde (rad/s).

Ω_r : Vitesse de rotation du rotor en radians par seconde (rad/s).

I.4. Modélisation de la machine asynchrone

La modélisation des systèmes réels pour obtenir des représentations des modèles mathématiques capables de prédire le comportement futur de ces systèmes. Plus un modèle se rapproche de la réalité, plus il devient complexe. En utilisant des hypothèses simplificatrices, les phénomènes mineurs sont ignorés tout en retenant les plus fondamentaux.

I.4.1. Hypothèses simplificatrices

Pour la modélisation des moteurs asynchrones, on utilise les hypothèses simplificatrices suivantes :

- Circuit magnétique non saturé et à perméabilité constant.
- Pertes ferromagnétiques négligeables.
- On néglige l'effet de peau
- Répartition spatiale sinusoïdale de la force magnétomotrice de l'entrefer.

Les conséquences importantes de ces hypothèses comprennent :

- Additivité des flux.
- Constance de l'auto-induction.
- La variation sinusoïdale de la mutuelle inductance entre les bobinages stator et rotor en fonction de l'angle électrique de leurs axes magnétiques.

I.4.2. Mise en équation de la machine asynchrone

Les moteurs asynchrones (**Figure I.1**) sont constitués d'un circuit statorique fixe comprenant trois enroulements identiques (A_s, B_s, C_s) décalés de 120° les uns des autres et d'un circuit rotorique mobile comprenant trois enroulements identiques (A_r, B_r, C_r) décalés. Le circuit du rotor est composé d'une différence de 120° entre eux.

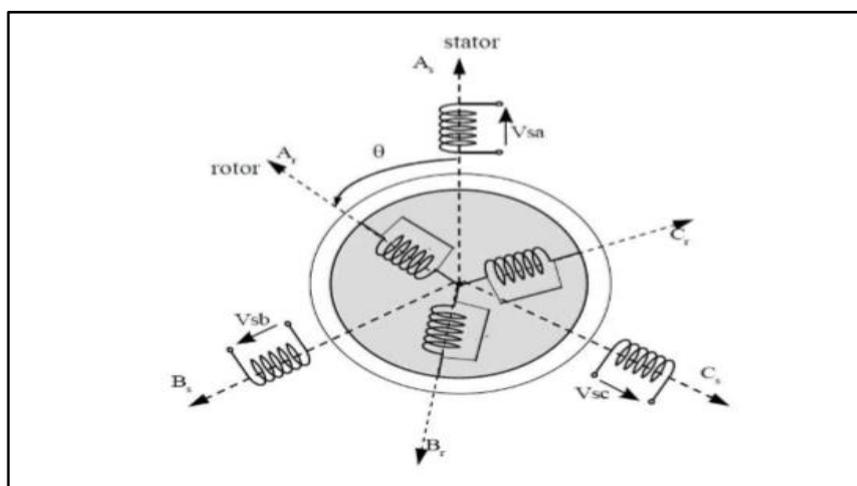


Figure I.1 : Représentation schématique d'une machine asynchrone triphasée.

L'angle θ caractérise la position angulaire du rotor par rapport au stator.

I.4.2.1. Équations des tensions

Chaque enroulement peut être représenté par la (Figure I.2) :

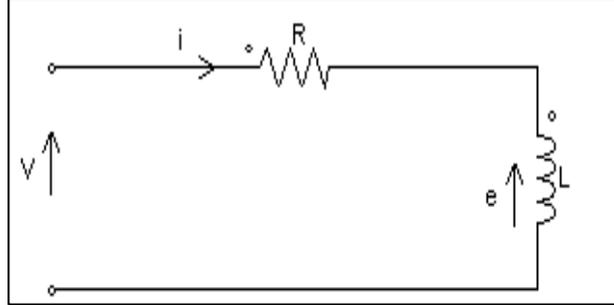


Figure I.2 : Modèle d'une phase avec force électromotrice.

Appliquons la loi des mailles à ce circuit, nous obtenons :

$$V = Ri + e \tag{I.2}$$

Avec la loi de Faraday

$$e = + \frac{d\phi}{dt} \tag{I.3}$$

Au final, nous pouvons écrire l'équation complète de tension :

$$V = Ri + \frac{d\phi}{dt} \tag{I.4}$$

Nous déduisons pour l'ensemble des phases statoriques en notation matricielle [13].

$$\begin{bmatrix} V_{sa} \\ V_{sb} \\ V_{sc} \end{bmatrix} = \begin{bmatrix} R_s & 0 & 0 \\ 0 & R_s & 0 \\ 0 & 0 & R_s \end{bmatrix} \begin{bmatrix} i_{sa} \\ i_{sb} \\ i_{sc} \end{bmatrix} + \frac{d}{dt} \begin{bmatrix} \phi_{sa} \\ \phi_{sb} \\ \phi_{sc} \end{bmatrix} \tag{I.5}$$

L'écriture matricielle cadencée :

$$[V_{3s}] = [R_s] \cdot [i_{3s}] + \frac{d[\phi_{3s}]}{dt} \tag{I.6}$$

L'écriture matricielle pour les trois phases rotoriques :

$$\begin{bmatrix} V_{ra} \\ V_{rb} \\ V_{rc} \end{bmatrix} = \begin{bmatrix} R_r & 0 & 0 \\ 0 & R_r & 0 \\ 0 & 0 & R_r \end{bmatrix} \begin{bmatrix} i_{ra} \\ i_{rb} \\ i_{rc} \end{bmatrix} + \frac{d}{dt} \begin{bmatrix} \phi_{ra} \\ \phi_{rb} \\ \phi_{rc} \end{bmatrix} \tag{I.7}$$

L'écriture matricielle cadencée :

$$[V_{3r}] = [R_r] \cdot [i_{3r}] + \frac{d[\phi_{3r}]}{dt} \quad (\text{I.8})$$

Les trois enroulements A_r , B_r , C_r sont en court-circuit (**Figure I.1**), d'où

$$[V_{3r}] = [0 \quad 0 \quad 0]^T \quad (\text{I.9})$$

I.4.2.2. Équations magnétiques

Chaque flux contient des interactions avec tous les courants de phase y compris le sien (notion de flux/inductance). Pour la phase statorique, on écrit :

$$\phi_{sa} = l_s i_{sa} + M_s i_{sb} + M_s i_{sc} + M_{aa} i_{ra} + M_{ab} i_{rb} + M_{ac} i_{rc} \quad (\text{I.10})$$

En matriciel :

$$\begin{bmatrix} \phi_{sa} \\ \phi_{sb} \\ \phi_{sc} \end{bmatrix} = \begin{bmatrix} l_s & M_s & M_s \\ M_s & l_s & M_s \\ M_s & M_s & l_s \end{bmatrix} \begin{bmatrix} i_{sa} \\ i_{sb} \\ i_{sc} \end{bmatrix} + \hat{M}_{sr} \begin{bmatrix} \cos(\theta) & \cos(\theta - \frac{4\pi}{3}) & \cos(\theta - \frac{2\pi}{3}) \\ \cos(\theta - \frac{2\pi}{3}) & \cos(\theta) & \cos(\theta - \frac{4\pi}{3}) \\ \cos(\theta - \frac{4\pi}{3}) & \cos(\theta - \frac{2\pi}{3}) & \cos(\theta) \end{bmatrix} \begin{bmatrix} i_{ra} \\ i_{rb} \\ i_{rc} \end{bmatrix} \quad (\text{I.11})$$

En posant :

$$[L_s] = \begin{bmatrix} l_s & M_s & M_s \\ M_s & l_s & M_s \\ M_s & M_s & l_s \end{bmatrix} \quad (\text{I.12})$$

Et

$$[M_{sr}] = \hat{M}_{sr} \begin{bmatrix} \cos(\theta) & \cos(\theta - \frac{4\pi}{3}) & \cos(\theta - \frac{2\pi}{3}) \\ \cos(\theta - \frac{2\pi}{3}) & \cos(\theta) & \cos(\theta - \frac{4\pi}{3}) \\ \cos(\theta - \frac{4\pi}{3}) & \cos(\theta - \frac{2\pi}{3}) & \cos(\theta) \end{bmatrix} \quad (\text{I.13})$$

Les équations du flux statoriques s'écrivent sous une forme cadencée :

$$[\phi_{3s}] = [L_s][i_{3s}] + [M_{sr}][i_{3r}] \quad (\text{I.14})$$

D'une façon similaire, les équations du flux rotoriques s'expriment sous la forme matricielle :

$$[\phi_{3r}] = [L_r][i_{3r}] + [M_{rs}][i_{3s}] \quad (\text{I.15})$$

Avec

$$[L_r] = \begin{bmatrix} l_r & M_r & M_r \\ M_r & l_r & M_r \\ M_r & M_r & l_r \end{bmatrix} \quad (\text{I.16})$$

Et

$$[M_{rs}] = [M_{sr}]^T \quad (\text{I.17})$$

Les deux équations (I.14) et (I.15) sont appelées équations magnétiques des moteurs asynchrones. En substituant les équations de débit (I.14) et (I.15) dans équations de tension (I.6) et (I.8), on obtient :

$$[V_{3s}] = [R_s] \cdot [i_{3s}] + \frac{d}{dt} \{ [L_s][i_{3s}] + [M_{sr}][i_{3r}] \} \quad (\text{I.18})$$

$$[V_{3r}] = [R_r] \cdot [i_{3r}] + \frac{d}{dt} \{ [L_r][i_{3r}] + [M_{rs}][i_{3s}] \} \quad (\text{I.19})$$

I.4.2.3. Équations du couple électromagnétique

L'expression du couple électromagnétique est donnée par la formule générale suivante [10] :

$$C_{em} = \frac{1}{2} [i]^T \cdot \left\{ \frac{d[L]}{d\theta} \right\} \cdot [i] \quad (\text{I.20})$$

Avec

$$[i] = \begin{bmatrix} [i_{3s}] \\ [i_{3r}] \end{bmatrix} \quad (\text{I.21})$$

$$[L] = \begin{bmatrix} [L_s] & [M_{sr}] \\ [M_{rs}] & [L_r] \end{bmatrix} \quad (\text{I.22})$$

Tous les termes des sous-matrices $[L_s]$ et $[L_r]$ sont constants, donc leurs dérivées par rapport à θ sont nulles, il y a donc :

$$C_{em} = \frac{1}{2} [[i_{3s}]^T \quad [i_{3r}]^T] \cdot \begin{bmatrix} \mathbf{0} & \frac{d[M_{sr}]}{d\theta} \\ \frac{d[M_{rs}]}{d\theta} & \mathbf{0} \end{bmatrix} \cdot \begin{bmatrix} [i_{3s}] \\ [i_{3r}] \end{bmatrix} \quad (\text{I.23})$$

Sachant que :

$$[i_{3r}]^T \cdot \left[\frac{d[M_{rs}]}{d\theta} \right] \cdot [i_{3s}] = [i_{3s}]^T \cdot \left[\frac{d[M_{sr}]}{d\theta} \right] \cdot [i_{3r}] \quad (\text{I.24})$$

Nous aurons :

$$C_{em} = [i_{3s}]^T \cdot \left[\frac{d[M_{sr}]}{d\theta} \right] \cdot [i_{3r}] \quad (I.25)$$

I.4.2.4. Équation mécanique

Elle exprime la relation entre le couple électromécanique et le couple résistant.

$$C_{em} - C_r = J \frac{d\Omega}{dt} + f\Omega \quad (I.26)$$

Avec :

C_{em} : Le couple électromécanique.

C_r : Le couple résistant.

Ω : La vitesse mécanique du rotor.

J : Le moment d'inertie de l'ensemble des parties tournantes.

f : Le coefficient de frottement.

On aura ainsi un système d'équations différentielles dans lequel certains des coefficients ne sont pas constants. Pour surmonter cette difficulté, on utilise alors une transformation mathématique, qui permet de décrire le comportement de la machine à l'aide d'équations différentielles à coefficients constants, à partir de laquelle on recourt à la transformation de Park.

I.4.3. Transformation de Park

La transformation de Park consiste à convertir le système d'arbres des enroulements du stator et du rotor (un système triphasé équilibré) en un système biphasé avec des axes orthogonaux (o , d et q). Où d est l'axe direct, q est l'axe orthogonal et o l'axe unipolaire (axe supplémentaire) (Figure I.3).

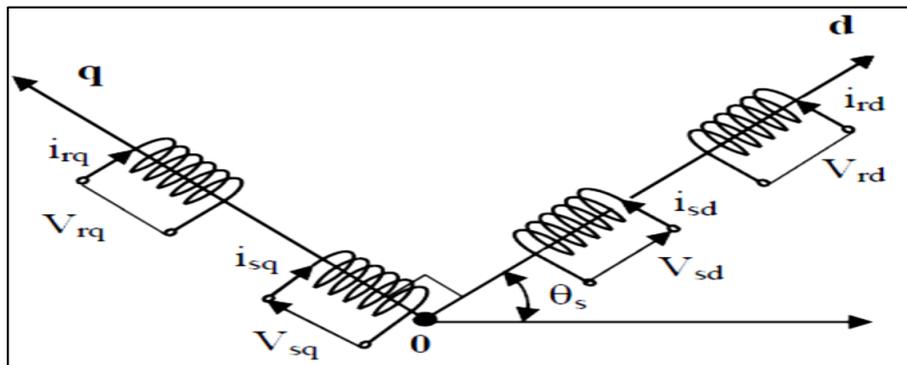


Figure I.3 : Représentation schématique du modèle biphasé équivalent.

La matrice de Park $[P(\theta)]$ de transformation de grandeurs physiques rend cette transformation possible. Cette transformation nous donne un système d'équations différentielles à coefficients constants. Les matrices Park et Park inverse sont définies comme suit [10] :

$$[P(\theta)] = \sqrt{\frac{2}{3}} \begin{bmatrix} \frac{1}{\sqrt{2}} & \cos \theta & -\sin \theta \\ \frac{1}{\sqrt{2}} & \cos(\theta - \frac{2\pi}{3}) & -\sin(\theta - \frac{2\pi}{3}) \\ \frac{1}{\sqrt{2}} & \cos(\theta - \frac{4\pi}{3}) & -\sin(\theta - \frac{4\pi}{3}) \end{bmatrix} \quad (\text{I.27})$$

$$[[P(\theta)]]^{-1} = \sqrt{\frac{2}{3}} \begin{bmatrix} \frac{1}{\sqrt{2}} & \frac{1}{\sqrt{2}} & \frac{1}{\sqrt{2}} \\ \cos \theta & \cos(\theta - \frac{2\pi}{3}) & \cos(\theta - \frac{4\pi}{3}) \\ -\sin \theta & -\sin(\theta - \frac{2\pi}{3}) & -\sin(\theta - \frac{4\pi}{3}) \end{bmatrix} \quad (\text{I.28})$$

Les composantes relatives et réelles sont reliées par :

$$\begin{bmatrix} X_o \\ X_d \\ X_q \end{bmatrix} = [P(\theta)]^{-1} \cdot \begin{bmatrix} X_a \\ X_b \\ X_c \end{bmatrix} \quad (\text{I.29})$$

$$\begin{bmatrix} X_a \\ X_b \\ X_c \end{bmatrix} = [P(\theta)] \cdot \begin{bmatrix} X_o \\ X_d \\ X_q \end{bmatrix} \quad (\text{I.30})$$

La composante homopolaire (X_o) dans un système équilibré est nulle.

I.4.3.1. Application de la transformation de Park à la MAS

Pour chaque jeu de grandeurs (stator et rotor). Si on note θ_s (resp. Par θ_r) angle de la transformation de Park de la grandeur stator (resp. rotor). Les références des transformations de Park des grandeurs stator et rotor doivent coïncider pour simplifier les équations de la machine asynchrone. Cela se fait en reliant les angles θ_s et θ_r par la relation suivante :

$$\theta_s = \theta + \theta_r \quad (\text{I.31})$$

En transformant les grandeurs statoriques du courant, de la tension et du flux à l'aide de la transformée de Park, on obtient :

✚ Pour le stator :

$$\begin{aligned} [V_{odqs}] &= [P(\theta_s)]^{-1}[V_{3s}] \text{ avec } [V_{odqs}] = [V_{os} \ V_{ds} \ V_{qs}]^T \\ [i_{odqs}] &= [P(\theta_s)]^{-1}[i_{3s}] \text{ avec } [i_{odqs}] = [i_{os} \ i_{ds} \ i_{qs}]^T \\ [\phi_{odqs}] &= [P(\theta_s)]^{-1}[\phi_{3s}] \text{ avec } [\phi_{odqs}] = [\phi_{os} \ \phi_{ds} \ \phi_{qs}]^T \end{aligned} \quad (\text{I.32})$$

✚ Pour le rotor :

$$\begin{aligned} [V_{odqr}] &= [P(\theta_r)]^{-1}[V_{3r}] \text{ avec } [V_{odqr}] = [V_{or} \ V_{dr} \ V_{qr}]^T \\ [i_{odqr}] &= [P(\theta_r)]^{-1}[i_{3r}] \text{ avec } [i_{odqr}] = [i_{or} \ i_{dr} \ i_{qr}]^T \\ [\phi_{odqr}] &= [P(\theta_r)]^{-1}[\phi_{3r}] \text{ avec } [\phi_{odqr}] = [\phi_{or} \ \phi_{dr} \ \phi_{qr}]^T \end{aligned} \quad (\text{I.33})$$

I.4.3.2. Application aux équations des tensions

En appliquant la transformée de Park à l'équation (I.6), on obtient :

$$[P(\theta_s)][V_{odqs}] = [R_s][P(\theta_s)][i_{odqs}] + \frac{d}{dt}\{[P(\theta_s)][\phi_{odqs}]\} \quad (\text{I.34})$$

En multipliant par $[P(\theta_s)]^{-1}$, on obtient :

$$\begin{aligned} [V_{odqs}] &= [R_s][i_{odqs}] + [P(\theta_s)]^{-1} \frac{d}{dt}\{[P(\theta_s)][\phi_{odqs}]\} \\ &= [R_s][i_{odqs}] + [P(\theta_s)]^{-1}[P(\theta_s)] \frac{d[\phi_{odqs}]}{dt} + [P(\theta_s)]^{-1}[\phi_{odqs}] \frac{d[P(\theta_s)]}{dt} \end{aligned} \quad (\text{I.35})$$

$$[V_{odqs}] = [R_s][i_{odqs}] + \frac{d[\phi_{odqs}]}{dt} + [P(\theta_s)]^{-1}[\phi_{odqs}] \frac{d[P(\theta_s)]}{dt} \quad (\text{I.36})$$

On a :

$$[P(\theta_s)]^{-1} \frac{d[P(\theta_s)]}{dt} = \frac{d\theta_s}{dt} \begin{bmatrix} 0 & 0 & 0 \\ 0 & 0 & -1 \\ 0 & 1 & 0 \end{bmatrix} \quad (\text{I.37})$$

Considérant un moteur en court-circuit avec un rotor dans une cage d'écurueil asynchrone, $V_{rd} = V_{rq} = 0$, alors les équations électriques et magnétiques du moteur s'écrivent sous le modèle vectoriel suivant :

$$\begin{cases} V_{ds} = R_s i_{ds} + \frac{d\phi_{ds}}{dt} - \omega_s \phi_{qs} \\ V_{qs} = R_s i_{qs} + \frac{d\phi_{qs}}{dt} + \omega_s \phi_{ds} \\ V_{dr} = R_r i_{dr} + \frac{d\phi_{dr}}{dt} - \omega_r \phi_{qr} = 0 \\ V_{qr} = R_r i_{qr} + \frac{d\phi_{qr}}{dt} + \omega_r \phi_{dr} = 0 \end{cases} \quad (\text{I.38})$$

I.4.3.3. Application aux équations des flux

On retrouve, après un développement les flux statoriques et rotoriques dans le repère de park comme suite :

$$\begin{cases} \phi_{ds} = L_s i_{ds} + M i_{dr} \\ \phi_{qs} = L_s i_{qs} + M i_{qr} \\ \phi_{dr} = L_r i_{dr} + M i_{ds} \\ \phi_{qr} = L_r i_{qr} + M i_{qs} \end{cases} \quad \text{Avec :} \quad \begin{cases} L_s = l_s - M_s \\ L_r = l_r - M_r \\ M = \frac{3}{2} M_{sr} \end{cases} \quad (\text{I.39})$$

On introduit l'équation de flux (I.39) dans l'équation (I.38), on obtient :

$$\begin{cases} V_{ds} = R_s i_{ds} + L_s \frac{di_{ds}}{dt} + M \frac{di_{dr}}{dt} - \omega_s L_s i_{qs} - \omega_s M i_{qr} \\ V_{qs} = R_s i_{qs} + L_s \frac{di_{qs}}{dt} + M \frac{di_{qr}}{dt} + \omega_s L_s i_{ds} + \omega_s M i_{dr} \\ 0 = R_r i_{dr} + L_r \frac{di_{dr}}{dt} + M \frac{di_{ds}}{dt} - \omega_r L_r i_{qr} - \omega_r M i_{qs} \\ 0 = R_r i_{qr} + L_r \frac{di_{qr}}{dt} + M \frac{di_{qs}}{dt} + \omega_r L_r i_{dr} + \omega_r M i_{ds} \end{cases} \quad (\text{I.40})$$

Les systèmes d'équation (I.39) et (I.40) constituent le modèle magnétique et électrique de la machine asynchrone.

I.4.3.4. Application aux équations mécanique et électromagnétique

En changeant les variables dans l'équation (I.25), on obtient :

$$C_{em} = [P(\theta_s)]^T [i_{odqs}]^T \left[\frac{d[M_{sr}]}{d\theta} \right] [P(\theta_r)] [i_{odqr}] \quad (\text{I.41})$$

Avec :

$$\begin{cases} [i_{3s}] = [P(\theta_s)] [i_{odqs}] \\ [i_{3r}] = [P(\theta_r)] [i_{odqr}] \end{cases}$$

On a le produit :

$$[P(\theta_s)]^T \left[\frac{d[M_{sr}]}{d\theta} \right] [P(\theta_s)] = M \begin{bmatrix} 0 & 0 & 0 \\ 0 & 0 & -1 \\ 0 & 1 & 0 \end{bmatrix} \quad (\text{I.42})$$

D'où :

$$C_{em} = p \cdot M \cdot (i_{dr} i_{qs} - i_{qr} i_{ds}) \quad (\text{I.43})$$

Où sous d'autres formes :

$$\begin{cases} C_{em} = p \cdot (\phi_{qr} i_{dr} - \phi_{dr} i_{qr}) \\ C_{em} = p \cdot (\phi_{ds} i_{qs} - \phi_{qs} i_{ds}) \\ C_{em} = p \cdot \frac{M}{L_r} (\phi_{dr} i_{qs} - \phi_{qr} i_{ds}) \end{cases} \quad (\text{I.44})$$

L'expression à utiliser dépend du vecteur d'état choisi. Dans notre travail, nous utiliserons la dernière expression. Afin de concevoir un modèle complet, il est nécessaire d'ajouter les équations mécaniques du mouvement (I.26)

I.5. Différents référentiels

I.5.1. Définition des différents référentiels

On exprime les équations et grandeurs de la machine dans le repère (d, q) , qui forme un angle électrique θ_s avec le stator et un angle électrique θ_r avec le rotor.

Il faut choisir un objet de référence en fonction du phénomène à étudier et de la simplification induite par ce choix. Il existe trois options importantes. Les marquages dq peuvent être apposés sur le stator, le rotor ou le champ tournant [11], [12].

I.5.1.1. Référentiel fixe par rapport au stator

Cette référence permet d'étudier les conditions de démarrage et de freinage du moteur électrique (avec des variations de vitesse importantes). Il se caractérise par :

$$\omega_s = \frac{d\theta_s}{dt} = 0 \text{ et } \omega_r = -\omega \quad (\text{I.45})$$

I.5.1.2. Référentiel fixe par rapport au rotor

Dans ce cas, le système d'arbres (d, q) est fixe par rapport au rotor et tourne avec la vitesse ($\omega_s = \omega$). Cet arbre permet d'étudier le régime transitoire des moteurs synchrones et asynchrones à vitesse constante.

I.5.1.3. Référentiel fixe par rapport au champ tournant

Dans ce cas, ce référentiel est souvent utilisé lorsque la fréquence de l'alimentation est constante. L'avantage d'utiliser ce référentiel est que les constantes peuvent être obtenues

en régime permanent. Cela facilite sa régulation. Il ne simplifie pas les équations de la machine. Il se caractérise par :

$$\frac{d\theta_s}{dt} = \omega_s \text{ et } \frac{d\theta_r}{dt} = \omega_s - \omega = \omega_r \quad (\text{I.46})$$

I.6. Représentation d'état de la machine asynchrone

Afin d'avoir un modèle complet du MAS, nous avons substitué les expressions de flux dans les équations de tension. Quatre équations électriques sont obtenues en fonction des composantes du courant et/ou du flux du stator et du rotor.

Le choix de la sortie est très important car il est lié à l'objectif de commande, dans notre cas les tensions stator (V_{ds}, V_{qs}) et la vitesse ω_s du champ tournant sont considérées comme variables de commande, le couple résistant (C_r) est considéré comme perturbation, et Courant stator et flux rotorique et vitesse mécanique ($i_{ds}, i_{qs}, \phi_{dr}, \phi_{qr}, \Omega$) comme variables d'état, la référence choisie est celle du champ tournant, on obtient donc [13] :

$$x = [I_{ds} \ I_{qs} \ \phi_{dr} \ \phi_{qr} \ \Omega]^T \quad (\text{I.47})$$

La connexion des équations mécaniques aux quatre équations électriques donne le modèle d'une machine asynchrone dans le cadre de Park, qui peut prendre la forme suivante :

$$\begin{cases} \frac{dx}{dt} = f(x) + hu + s_r C_r \\ y = \begin{bmatrix} I_{ds} \\ I_{qs} \\ \Omega \end{bmatrix} \end{cases} \quad (\text{I.48})$$

Avec :

$$u = [V_{ds} \ V_{qs}]^t$$

$$\frac{dx}{dt} = [i_{ds} \ i_{qs} \ \dot{\theta}_{dr} \ \dot{\theta}_{qr} \ \dot{\Omega}]^t$$

$$\phi_r = \sqrt{\phi_{dr}^2 + \phi_{qr}^2}$$

$$f(x) = \begin{bmatrix} -\gamma I_{ds} + \omega_s I_{qs} + \frac{k}{T_r} \phi_{dr} + p\Omega K \phi_{qr} \\ -\omega_s I_{ds} - \gamma I_{qs} - p\Omega K \phi_{dr} + \frac{k}{T_r} \phi_{qr} \\ \frac{M}{T_r} I_{ds} - \frac{1}{T_r} \phi_{dr} + (\omega_s - p\Omega) \phi_{qr} \\ \frac{M}{T_r} I_{qs} - (\omega_s - p\Omega) \phi_{dr} - \frac{1}{T_r} \phi_{qr} \\ p \frac{M}{JT_r} (\phi_{dr} I_{qs} - \phi_{qr} I_{ds}) - \frac{f\Omega}{J} \end{bmatrix}; h = \begin{bmatrix} \frac{1}{\sigma L_s} & 0 \\ 0 & \frac{1}{\sigma L_s} \\ 0 & 0 \\ 0 & 0 \end{bmatrix}; S_r = \begin{bmatrix} 0 \\ 0 \\ 0 \\ 0 \\ -1 \\ J \end{bmatrix} \quad (I.49)$$

Et :

$$T_r = \frac{L_r}{R_r}$$

$$\sigma = 1 - \frac{M^2}{L_s L_r}$$

$$K = \frac{M}{\sigma L_s L_r}$$

$$\gamma = \frac{R_s}{\sigma L_s} + \frac{R_r M^2}{\sigma L_s L_r^2}$$

P=2 : nombre de pair de pôles de la MAS.

I.7. Simulation de la machine asynchrone sous Matlab/Simulink

La simulation est devenue une tâche primordiale pour les chercheurs avant d'entamer toute réalisation, Le but de cette simulation est de valider le modèle adopté de la machine asynchrone en boucle ouvert. On a choisi le logiciel Matlab/Simulink très connu pour sa puissance de calcul.

La **Figure I.4** représente l'évolution de la vitesse de rotation en fonction de temps, la montée en vitesse est quasi linéaire au début de démarrage jusqu'à une vitesse de (149.85rad/s).

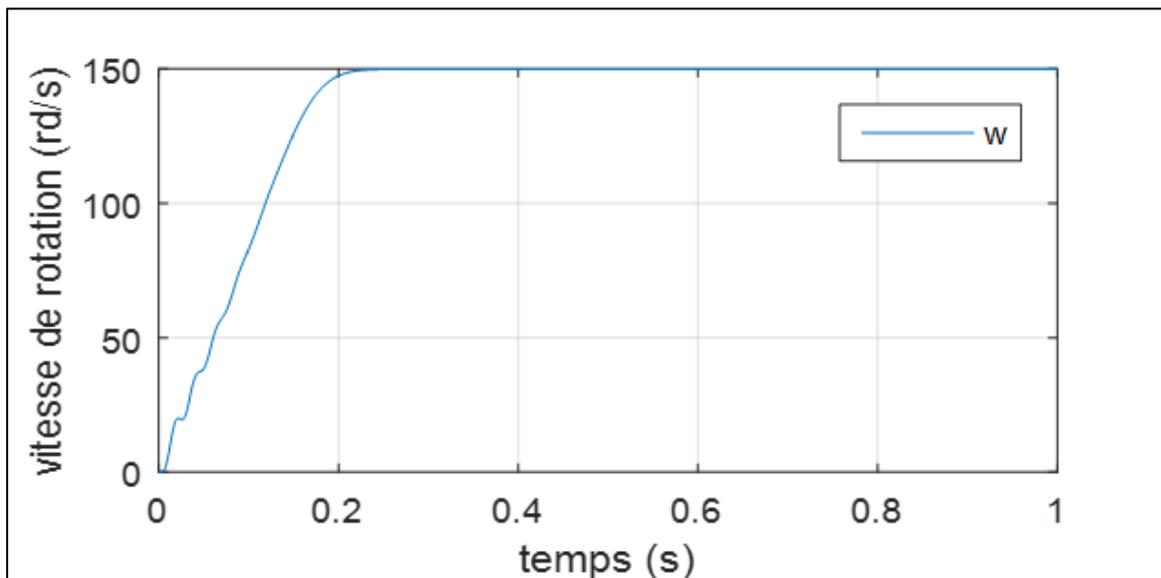


Figure I.4 : Résultat de simulation en boucle ouverte de la MAS.

I.8. Conclusion

Dans ce chapitre, nous avons mis en place la modélisation et la simulation du moteur à induction. Ce type de machine est largement utilisé dans l'industrie en raison de sa structure simple et de sa robustesse.

Son modèle pratique applique la transformation de Park pour passer d'un système triphasé à un système diphasique, en tenant compte des hypothèses simplificatrices et du choix du référentiel.

CHAPITRE

II

II.1. Introduction

L'intelligence artificielle (IA) est une discipline en constante évolution qui utilise des algorithmes sophistiqués et des modèles mathématiques pour permettre aux machines de réaliser des tâches intelligentes. Parmi les techniques les plus importantes de l'IA, on trouve les réseaux de neurones artificiels (RNA) et l'algorithme ABC. Les RNA sont des modèles informatiques qui imitent le fonctionnement des neurones biologiques, tandis que l'algorithme ABC est une méthode d'optimisation inspirée du comportement des abeilles.

Ces techniques sont utilisées pour résoudre une grande variété de problèmes, tels que la reconnaissance de la parole, la vision par ordinateur, la prédiction du marché boursier et la recommandation de contenu en ligne.

Avec les avancées récentes dans les domaines du traitement de données massives et de l'apprentissage profond, les RNA et l'algorithme ABC ont ouvert de nouvelles perspectives pour l'IA et ont le potentiel de transformer radicalement notre façon de vivre et de travailler.

II.2. Les réseaux de neurones (RNA)

II.2.1. Définition

Un réseau de neurones artificiels (RNA) est un système de traitement de l'information dont la structure est inspirée du système nerveux biologique. Il est composé de neurones formels qui sont fortement connectés pour former un réseau de processeurs élémentaires travaillant en parallèle. Chaque processeur élémentaire calcule une sortie unique en fonction des informations qu'il reçoit. Il est important de souligner que tout réseau de neurones hiérarchique peut être considéré comme un réseau plus simple. [14].

II.2.2. Neurone biologique

Les neurones sont des cellules spécialisées du système nerveux qui permettent la transmission de l'information nerveuse. Ils ont une structure complexe, avec un corps cellulaire, des dendrites, un axone et des terminaisons synaptiques. Les neurones sont capables de générer et de transmettre des signaux électrochimiques appelés influx nerveux. Il existe plusieurs types de neurones, tels que les neurones sensitifs, moteurs et interneurones. Les synapses, qui sont les connexions entre les neurones, peuvent être renforcées ou affaiblies en fonction de l'expérience et de l'apprentissage. Les neurones communiquent entre eux par le biais de neurotransmetteurs, des substances chimiques qui se lient à des récepteurs sur la membrane des

cellules cibles. Les maladies neurologiques peuvent être liées à des anomalies dans le fonctionnement des neurones ou dans leurs connexions. [15]

II.2.3. Neurones Artificiels

Les neurones artificiels, sont des modèles simplifiés de neurones biologiques, qui imitent certaines des fonctions des neurones biologiques, telles que la mémoire associative, l'apprentissage et le travail parallèle. [16]

II.2.4. Modélisation générale

Les neurones artificiels (**Figure II.1**), reçoivent des informations des entrées $x_i = (i = 1,2,3 \dots n)$ via des connexions qui attribuent à chaque neurone un poids w_i , représente également le force de la connexion. Les neurones artificiels fonctionnent en deux étapes :

✚ La première étape consiste à calculer le potentiel v_j du neurone j via la fonction suivante, et à prétraiter les données reçues :

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

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

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

La relation s'écrit donc plus simplement :

$$v_j = \sum_{i=0}^n w_{j,i} x_i \quad (\text{II. 2})$$

✚ Dans la deuxième étape, la fonction de transfert g , également appelée fonction d'activation, calcule la valeur de l'état interne S_j du neurone j en fonction de la valeur de la latente v . Cette valeur précisera la sortie du neurone :

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

Dans certains cas, le choix de la fonction d'activation s'est avéré être une partie importante des réseaux de neurones (**Tableau II.1**). Ainsi, un neurone peut être défini mathématiquement comme une fonction algébrique, non linéaire et bornée de l'entrée x_i , paramétrée par des poids w_j, i [17] :

$$S_j = g(x_i, w_{j,i}) = g w_{j,i}(x_i) \quad i = (1,2,3 \dots n) \quad (\text{II. 4})$$

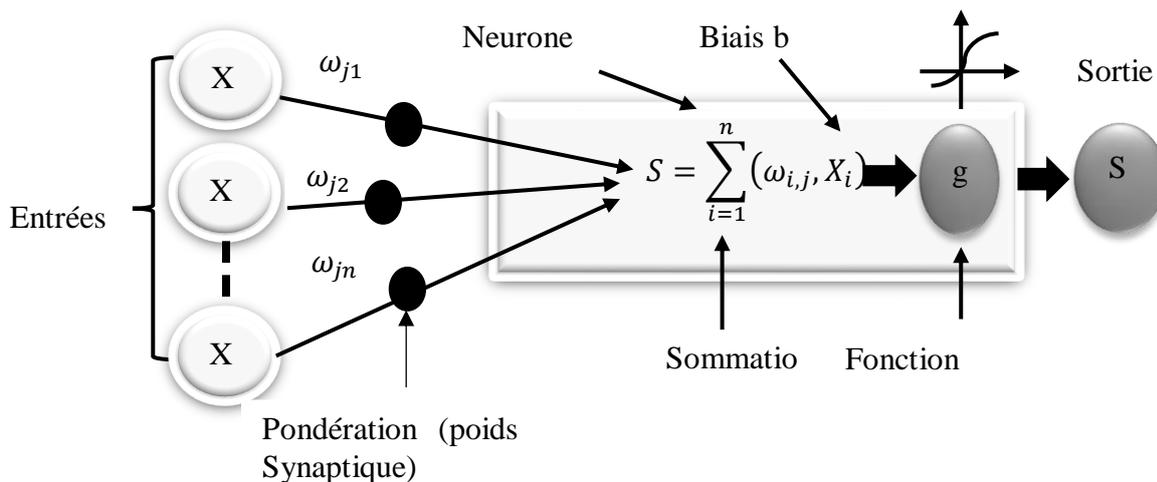


Figure II.1 : Neurone artificiel

Les fonctions d'activation les plus couramment utilisées sont les fonctions "seuil" ("limite dure"), "linéaire" et "sigmoïde". Le contexte d'utilisation des neurones, la fonction est illustrée à la (Figure II.2) [18].

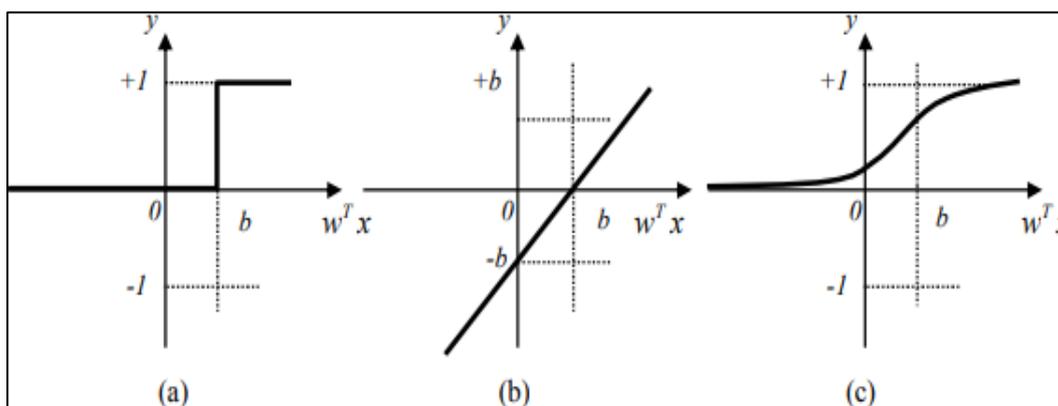


Figure II.2 : Fonctions d'activations : (a) du neurone « seuil » ; (b) du neurone « linéaire », et (c) du neurone « Sigmoïde ».

Tableau II.1 : Différentes fonctions d'activations utilisées dans les RNA.

Nom de la fonction	Relation entrée/sortie	Icône	Nom MATALB
Seuil	$y = 0$ si $s < 0$ $y = 1$ si $s \geq 0$		Hardlim
Seuil symétrique	$y = -1$ si $s < 0$ $y = 1$ si $s \geq 0$		Hardlims
Linéaire	$y = s$		Purelin
Linéaire saturée	$y = 0$ si $s \leq 0$ $y = s$ si $0 \leq s \leq 1$ $y = 1$ si $s \geq 1$		Satlin
Linéaire saturée symétrique	$y = -1$ si $s < -1$ $y = s$ si $-1 \leq s \leq 1$ $y = 1$ si $s > 1$		Satlins
Linéaire positive	$y = 0$ si $s \leq 0$ $y = s$ si $s \geq 0$		Poslin
Sigmoïde	$y = \frac{1}{1 + e^{-s}}$		Logsig
Tangente hyperbolique	$y = \frac{e^s - e^{-s}}{e^s + e^{-s}}$		Tansig
Compétitive	$y = 1$ si s maximum $y = 0$ autrement		Compet

La fonction linéaire est très simple, elle affecte directement l'entrée à la sortie :

$$y = s \quad (\text{II. 5})$$

Appliquée aux neurones, cette fonction est illustrée à (**Figure II.2**). Dans ce cas, la sortie du neurone est son niveau d'activation, dont le passage par zéro se produit lorsque $w^T x = b$.

Les fonctions sigmoïdes apparaissent comme des approximations continuellement différentiables des fonctions d'activation linéaires par morceaux ou des fonctions de seuil. Il a l'avantage d'être régulier, monotone, continûment différentiable et borné entre 0 et 1 [19] :

$$f(x) = \frac{1}{1 + e^{-x}} \quad (\text{II. 6})$$

II.2.5. Propriétés des réseaux de neurones

En général, les réseaux de neurones ont les propriétés suivantes [20] :

- **Le parallélisme** : c'est la base des architectures de réseaux neuronaux, qui un ensemble d'entités fondamentales travaillant simultanément.
- **La capacité d'adaptation** : considère comme une capacité d'apprentissage, qui permet au réseau de prendre en compte de nouvelles contraintes ou de nouvelles données du monde extérieur. Elle se caractérise par sa capacité d'auto-organisation, qui assure son stabilité en tant que systèmes dynamiques.
- **La généralisation** : est sa capacité à répondre de manière satisfaisante aux entrées qui ne font pas partie de son histoire d'exemples de formation.
- **La mémoire distribuée** : correspond à une carte des activations neuronales. Cette carte est en quelque sorte un encodage du fait de la mémoire.

II.2.6. Architecture des réseaux de neurones

On distingue deux principaux types d'architectures de réseaux de neurones :

II.2.6.1. Réseau de neurones non bouclé (feed-forward)

Connus sous le nom de « réseaux non récurrents », il exécute une fonction (ou des fonctions) algébrique(s) sur son entrée en combinant les fonctions exécutées par chaque neurone [21]. Représenté graphiquement par un ensemble de neurones interconnectés. Les informations circulent de l'entrée à la sortie sans retour en arrière.

Utilisés pour effectuer des tâches d'approximation de fonctions non linéaires et de modélisation de processus non linéaires [17]. On trouve :

➤ **Le perceptron monocouche « SLP »** : est un réseau très simple, il s'agit généralement de suivre l'apprentissage supervisé selon les règles de correction d'erreurs ou selon la règle de Hebb [22].

➤ **Le perceptron multicouches « PMC »** : Les neurones sont disposés en couches. Il n'y a pas de connexions entre les neurones d'une même couche, seulement avec les neurones des couches en aval. En règle générale, chaque neurone d'une couche est connecté à tous les neurones de la couche suivante, et uniquement à cette couche.

➤ **Réseaux à fonction radiale « RBF »** : Sont très similaires aux PMC, mais leur particularité réside dans leur utilisation caractéristique des fonctions gaussiennes comme fonctions de base [22]. En raison de son architecture, les règles d'apprentissage par correction d'erreurs et les règles d'apprentissage par compétition sont les plus couramment utilisées.

II.2.6.2. Réseau de neurones bouclé (feed-back)

Aussi connus sous le nom de « réseaux récurrents », ces réseaux renvoient des informations dans le sens de propagation opposé tel que défini dans les réseaux multicouches. Ces connexions sont généralement locales [14].

II.2.7. Types d'apprentissage des réseaux de neurones

Est une étape de développement d'un réseau de neurones au cours de laquelle le comportement du réseau est modifié jusqu'à l'obtention du comportement souhaité. [23, 24]. En termes d'algorithmes d'apprentissage, selon que le superviseur intervient ou non, il peut être divisé en trois types [24] :

II.2.7.1. Apprentissage supervisé

Appelé par ce nom car la sortie attendue pour chaque ensemble d'entrées doit être connue, et le réseau de neurones ajustera ses paramètres pour minimiser l'erreur entre la sortie attendue et sa sortie réelle. Ce modèle est le plus courant, où l'utilisateur a un comportement de référence spécifique qu'il souhaite que le réseau apprenne. En principe, elle consiste à évaluer des poids synaptiques pour minimiser l'erreur sur une base apprise constituée d'un ensemble d'observations dont on connaît les entrées et les sorties souhaitées.

II.2.7.2. Apprentissage semi- supervisé

Cet apprentissage est le même que l'apprentissage précédent en ce sens qu'il est basé sur la présence du concepteur, mais la valeur exacte de la sortie n'est pas disponible, souvent la seule information disponible est le signal d'échec ou de réussite [17]

II.2.7.3. Apprentissage non supervisé

Il correspond à la situation où il n'y a pas de base d'apprentissage, comme par exemple ne pas savoir a priori comment juger si une sortie est valide ou non dans le temps. Il consiste à détecter automatiquement les régularités qui se produisent dans les exemples et à modifier les poids des connexions afin que les exemples avec la même caractéristique de régularité produisent la même sortie [17, 24].

II.2.8. Règles d'apprentissage

L'apprentissage des RNA est basé sur des règles d'apprentissage. Nous citons ci-dessous les règles les plus générales [24, 25].

II.2.8.1. La règle de Hebb

C'est la méthode d'apprentissage la plus ancienne et elle s'inspire de la biologie. Il reflète le renforcement de la connexion reliant deux neurones activés. Si au moins un des deux neurones n'est pas activé, les poids de la connexion ne sont pas modifiés.

II.2.8.2. Règle de Delta (Windrow -Hoff)

Elle calcule la différence entre la valeur de sortie et la valeur attendue pour ajuster les poids synaptiques. Il emploie une fonction d'erreur appelée "moindre carré moyen" basée sur la variance utilisée pour ajuster les poids.

II.2.8.3. Règle d'apprentissage compétitive

Il n'implique qu'un seul neurone. Les données sont regroupées par catégorie. Ainsi, des neurones similaires seront placés dans la même classe en fonction des dépendances de données et représentés par un seul neurone.

II.2.8.4. Règle de correction d'erreurs

Cette règle s'applique généralement aux paradigmes d'apprentissage supervisé, c'est-à-dire où le réseau reçoit des entrées et des sorties correspondantes. Si nous pensons que y est la sortie calculée par le réseau et d est la sortie attendue, le principe de cette règle est d'utiliser l'erreur $(d - y)$ pour modifier la connexion afin de réduire l'erreur globale du système, donc le réseau s'adaptera jusqu'à ce que y soit égal à d .

II.2.9. Domaines d'application

Le domaine d'application des réseaux de neurones est très large, voici quelques exemples [26].

- **Contrôle** : En raison de leurs propriétés (parallélisme de traitement, capacité d'adaptation et de généralisation, etc.). Le réseau neuronal est appliqué au contrôle intelligent.
- **Traitement d'images** : Reconnaissance de caractères et de signatures, compression d'images, reconnaissance de formes, cryptage, classification, etc.
- **Traitement du signal** : Filtrage, classification, identification de source, traitement de la parole, où le réseau agit comme un filtre pour éliminer le bruit.

- **Défense** : Guidage de missiles, suivi de cible, reconnaissance faciale, radar, sonar, lidar, compression de données, suppression de bruit, etc.
- **Optimisation** : Nécessite généralement une optimisation de la planification, de l'allocation des ressources, de la gestion et du financement des opérations ou des tâches.
- **Simulation** : Simulation de vol, simulation de boîte noire, prévisions météo...etc.
- **Diagnostic** : Les réseaux de neurones sont bien adaptés pour résoudre les problèmes de diagnostic, en utilisant la classification automatique des signaux et des formes [27].

II.3. L'algorithme de colonies d'abeilles artificielles

Un certain nombre d'algorithmes ont émergé au cours de la dernière décennie, tentant d'exploiter les principes du comportement collectif des abeilles pour résoudre plusieurs problèmes difficiles dans différents domaines.

L'optimisation des colonies d'abeilles artificielles (ABC) appartient à la catégorie des algorithmes principalement inspirés du comportement des abeilles dans la nature [28], et plus largement au domaine de l'intelligence computationnelle et des métaheuristiques. Une procédure relativement développée par **Karaboga et Basturk** depuis 2005 pour les problèmes d'optimisation continue

II.3.1. Principe de l'algorithme ABC

Dans cet algorithme, les solutions candidates à un problème d'optimisation sont représentées par des sources alimentaires. Chaque source de nourriture possède une certaine quantité de nectar qui caractérise sa qualité (fitness). La population de la ruche est divisée en trois groupes d'abeilles qui parcourent l'espace de recherche afin de trouver des sources de nourriture.

Chaque groupe correspond au vol d'un des trois types d'abeilles : ouvrières, spectatrices et exploratrices. Les ouvrières errent et parcourent le voisinage autour des sources de nourriture à la recherche de meilleures sources que celles visitées. Ensuite, ils partagent la qualité de la source avec les spectatrices. Ceux-ci se concentrent sur les sources alimentaires de la plus haute qualité. Lorsqu'une source de nourriture est complètement explorée, elle est abandonnée et les exploratrices partent au hasard pour trouver de nouvelles sources (**Figure II.3**).

Le nombre d'abeilles ouvrières et de spectatrices correspond au nombre de sources de nourriture. Il y a généralement une abeille exploratrice. La source de nourriture est un vecteur de dimension D , où D est la dimension du problème, et le nombre de sources de nourriture (SN) est un paramètre de l'algorithme [26].

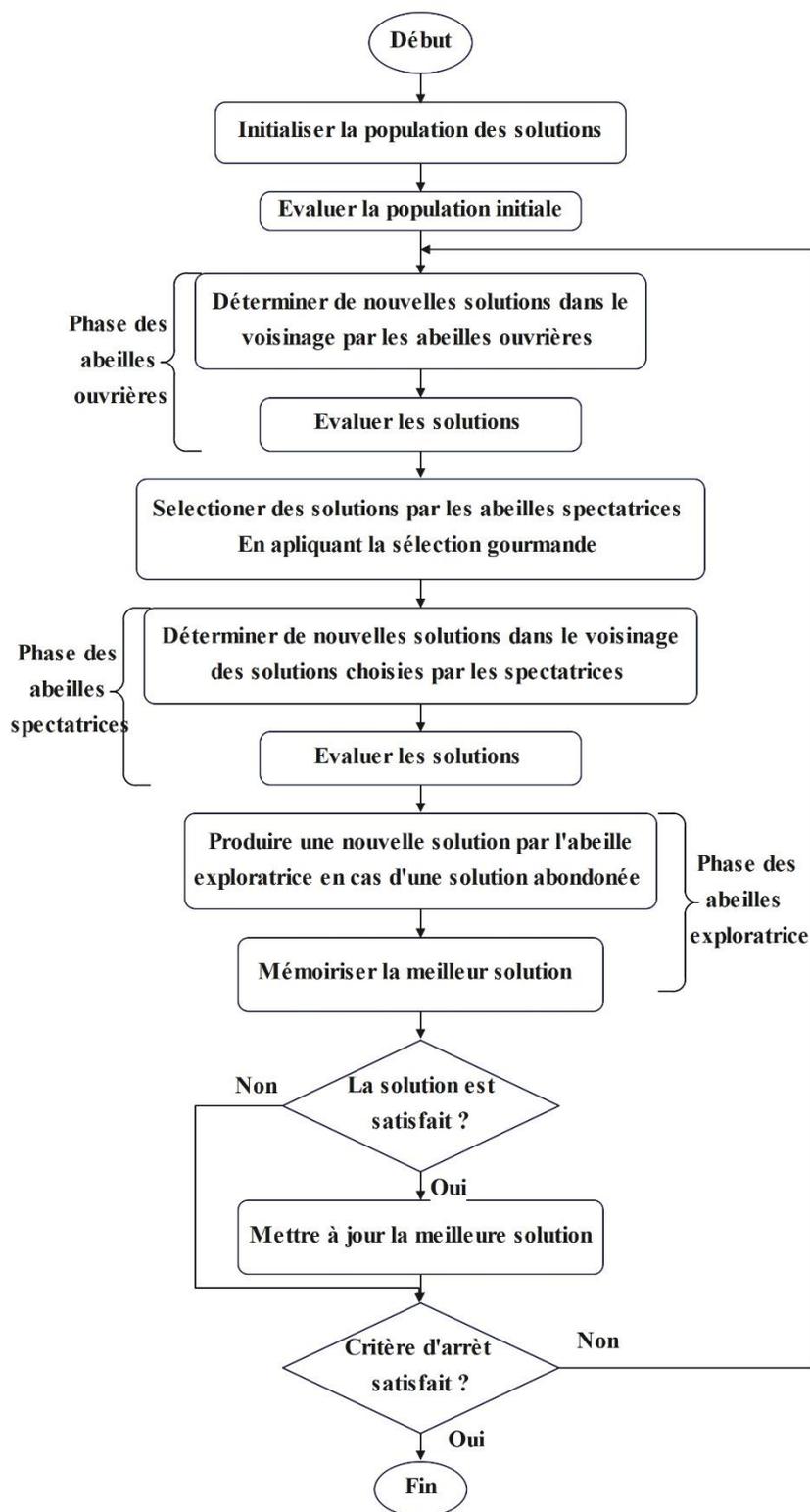


Figure II.3 : Optimisation par colonies d’abeilles artificielles [32].

II.3.2. Le processus de l'algorithme ABC

II.3.2.1. Initialisation de l'algorithme

Cet algorithme génère des individus dont la population initiale SN est distribuée aléatoirement, où SN représente la taille de la population et, $X_i = \{x_{i,1}, x_{i,2} \dots x_{i,D}\}$, 'i' dans la population est la source alimentaire, D est la dimensionnalité du problème d'optimisation.

$$X_{ij} = x_{min,j} + rand[0,1](x_{max,j} - x_{min,j}), i \in \{1, \dots, SN\}, j \in \{1, \dots, D\} \quad (II.7)$$

✚ **Phase ouvrière :** Pour générer une nouvelle solution (source de nourriture), l'algorithme ABC utilise l'expression suivante :

$$v_{ij} = x_{ij} + \phi_{ij}(x_{ij} - x_{kj}) \quad (II.8)$$

Où ϕ_{ij} appartenant à l'intervalle $[-1, 1]$, v_{ij} est une nouvelle position d'une source de nourriture potentielle près de x_{ij} qu'est l'ancienne source de nourriture, $K \in \{1, 2, \dots, SN\}$, ($k \neq i$) et $j \in \{1, \dots, D\}$ sont choisis au hasard.

✚ **Phase spectatrice :** Au cours de cette phase, les abeilles employées utilisent des danses frétilantes pour partager des informations sur les nouvelles sources de nourriture x_i avec les abeilles spectatrices. (Waggle dance). La valeur de probabilité P_i de la solution x_i en utilisant son fitness est calculée comme suit :

$$P_i = \frac{f_i t_j}{\sum_{j=1}^{SN} f_i t_j} \quad (II.9)$$

Il existe d'autres façons de calculer la probabilité P_i dont l'une que l'on retrouve souvent dans la littérature est la suivante :

$$P_i = \frac{0.9 f_i t_j}{f_i t_{max}} + 0.1 \quad (II.10)$$

Où $f_i t_j$ est le fitness de la solution x_i , et La quantité de nectar associée à la source x_i est calculée à l'aide de l'expression suivante :

$$fit(x_i) = \begin{cases} \frac{1}{f(x_i) + 1}, & f(x_i) \geq 0 \\ 1 + |f(x_i)|, & f(x_i) < 0 \end{cases} \quad (II.11)$$

Où $f(x_i)$ est la valeur de la fonction objective de la solution associée x_i .

Si la nouvelle solution v_{ij} est meilleure que l'ancienne solution x_i alors v_{ij} remplace x_i dans la population, sinon x_i est préservé dans la mémoire et son compteur d'abandon est incrémenté par 1. Notez que si la nouvelle solution v_{ij} en dehors de l'espace de recherche nous devons la réinitialiser pour qu'elle soit à l'intérieur de l'espace de recherche.

✚ **Phase exploratrice** : Si le nombre de cycles de recherche atteint une certaine limite, mais qu'aucune meilleure solution n'est trouvée autour d'une certaine source de nourriture, la source de nourriture est abandonnée, puis les abeilles exploratrices génèrent aléatoirement de nouvelles sources de nourriture selon l'équation (II.9).

La dernière étape consiste à voir si l'algorithme remplit les cratères d'arrêt, si oui, gardez la meilleure solution, sinon l'algorithme passe à l'itération suivante.

II.3.2.2. Pseudo codes de l'algorithme ABC

La structure générale de la méthode d'optimisation de l'algorithme ABC est en (Tab II.2).

Tableau II.2 :Pseudo code de l'algorithme ABC. [30]

<ol style="list-style-type: none"> 1. Générer la population initiale $X_i, i=1\dots, SN$ par l'équation (II.8) 2. Évaluer la population initiale par l'équation (II. 11) 3. Cycle = 1 4. Répéter 5. Pour chaque abeille Employed { <ul style="list-style-type: none"> • Appliquer la sélection gourmande • Produire une nouvelle solution v_{ij} par l'équation (II.9) • Calculer la valeur de la fonction d'adéquation $f_i t_i$ par l'équation (II.12) 6. Calculer la probabilité P_i associée à la solution x_i par l'équation (II.10) 7. Pour chaque abeille Onlooker { <ul style="list-style-type: none"> • Appliquer la sélection gourmande • Sélectionner une solution x_i en fonction de P_i • Produire une nouvelle solution v_{ij} • Calculer la valeur de la fonction d'adéquation $f_i t_i$ 8. S'il y'a une solution abandonnée <ul style="list-style-type: none"> • Alors remplacer la avec une nouvelle solution qui sera produite aléatoirement par L'équation (II.8) 9. Mémoriser la meilleure solution trouvée 10. $cycle = cycle+1$ 11. Jusqu'à $cycle = cyclemax$

II.3.3. Applications de l'algorithme ABC

L'algorithme des colonies d'abeilles artificielles a fait l'objet d'un grand nombre d'applications dans divers domaines tels que l'électronique, la bio-informatique, le traitement d'images, et l'exploration de données. Plusieurs études ont montré l'efficacité et la fiabilité de l'application du l'algorithme ABC et ces remarquables résultats d'où on trouve parmi eux :

L'algorithme ABC a été utilisé par **Karaboga et al** comme outil d'apprentissage appliqué aux réseaux de neurones artificiels en ajustant les poids synaptiques [31].

En 2012, **Akay et al** ont proposé d'utiliser l'algorithme ABC pour résoudre des problèmes d'optimisation de grande dimension [32].

Une étude a montré l'application du modèle ABC dans la classification floue en considérant une approche hybride de détection des tumeurs à partir d'images IRM [33].

Dans le cadre de l'optimisation multi-objectif, **Prajati et Srivastava** ont proposé l'application de l'algorithme ABC aux problèmes d'optimisation de la puissance réactive du réseau électrique [34].

On constate que l'application de la stratégie ABC touche en fait plusieurs domaines d'intérêt, c'est pourquoi la compréhension de cette métaheuristique reste une voie de recherche prometteuse.

II.4. Conclusion

Les Réseaux de Neurones Artificielle sont l'un des outils les plus couramment utilisés pour résoudre des problèmes complexes dans différents domaines, tels que la détection de défauts statoriques dans les machines asynchrones, grâce à ses propriétés associées à des capacités d'apprentissage. Cet apprentissage dépend de l'architecture du réseau et des règles d'adaptation des poids des neurones.

La méthode d'optimisation des colonies d'abeilles artificielles (ABC) est l'une des méthodes d'optimisation récentes. Elle est représentée par un algorithme qui peut être appliqué à de nombreux problèmes d'optimisation en ingénierie, gestion, contrôle... Cette méthode a la capacité d'enrichir l'étude des régions prometteuses de l'espace des solutions grâce au processus d'échange et d'amélioration de l'information.

CHAPITRE

III

III.1. Introduction

Ce chapitre porte sur l'utilisation des réseaux de neurones pour la modélisation de la machine asynchrone, en utilisant l'algorithme ABC pour optimiser les paramètres du modèle. Cette approche nécessite une expertise en modélisation et en traitement de données, ainsi qu'une validation expérimentale rigoureuse pour garantir la fiabilité et la sécurité de la commande de la machine.

L'algorithme ABC est une méthode d'optimisation inspirée du comportement des abeilles dans la recherche de nourriture. Le chapitre explore les étapes nécessaires pour utiliser les réseaux de neurones et l'algorithme ABC dans la modélisation de la machine asynchrone, ainsi que les avantages et les défis de cette approche.

III.2. Simulation numérique

La méthode d'Euler est une méthode de résolution numérique des équations différentielles ordinaires qui approxime les solutions en utilisant des formules de différences finies. Pour appliquer cette méthode, il est essentiel de choisir un pas de temps approprié, qui subdivise l'intervalle de temps sur lequel l'équation différentielle est résolue. Plus le pas de temps h est petit, plus l'approximation sera précise, mais cela augmente également la quantité de calcul nécessaire.

- **Formule de base de la méthode d'Euler.**

Pour une équation différentielle de la forme :

$$\dot{\mathbf{y}}(t) = \mathbf{f}(t, \mathbf{y}(t)) \quad (\text{III. 1})$$

Avec :

$\mathbf{y}(t)$: La fonction inconnue.

$\mathbf{f}(t, \mathbf{y}(t))$: La fonction décrivant la dérivée de \mathbf{y} par rapport à t .

La méthode d'Euler utilise la formule suivante :

$$\mathbf{y}(t + h) = \mathbf{y}(t) + h * \mathbf{f}(t, \mathbf{y}(t)) \quad (\text{III. 2})$$

Avec :

h : La taille de l'intervalle de temps.

$\mathbf{y}(t + h)$: Est la valeur approximée de \mathbf{y} à l'instant $t + h$.

- **Simulation numérique de modélé de la machine asynchrone par la méthode d'Euler.**

La simulation numérique de modèles de machines asynchrones est une approche courante pour étudier leur comportement. Dans notre simulation, nous avons défini les paramètres spécifiques de la machine asynchrone, tels que les résistances, les inductances et les moments d'inertie. En itérant sur un certain nombre d'itérations, nous pouvons observer l'évolution des variables d'état au fil du temps, ce qui nous permettra d'analyser le comportement de la machine asynchrone. Les résultats de la simulation sont représentés dans la **Figure III.1**.

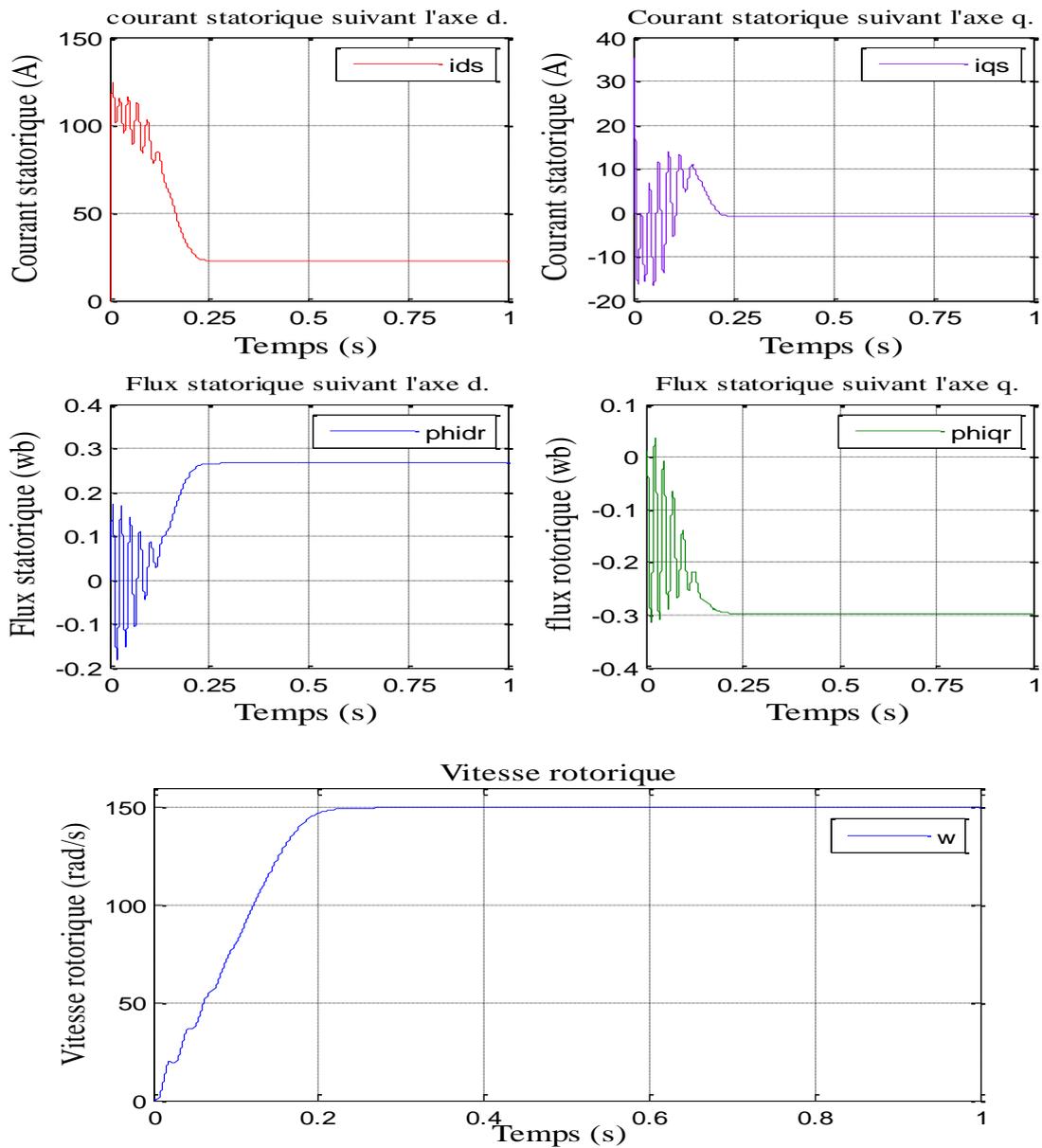


Figure III.1 : Représentation graphique des résultats de la simulation du comportement de la machine asynchrone.

III.3. Modélisation de la machine asynchrone par les réseaux de neurones

Le schéma de principe d'identification est illustré par la **Figure III.2**.

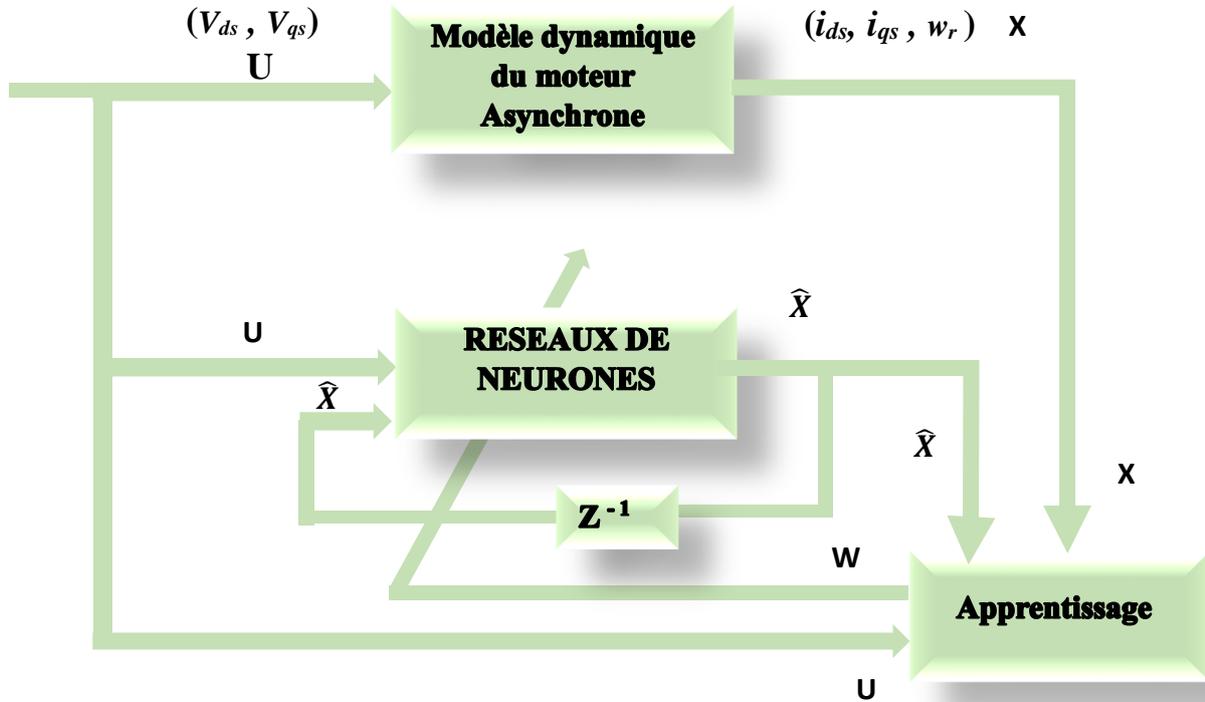


Figure III.2 : Modélisation de la machine asynchrone par réseaux de neurones.

Nous allons développer le modèle neuronal pour identifier le comportement dynamique de la machine asynchrone. Le choix du modèle neuronal à apprentissage dépend de la complexité du modèle d'état qui décrit le fonctionnement de la machine. Le modèle neuronal de la machine asynchrone dispose de cinq entrées ($V_{ds}, V_{qs}, \widehat{i}_{ds}, \widehat{i}_{qs}, w_r$) et trois sorties (vitesse rotoriques \widehat{w}_r et les courants ($\widehat{I}_{ds}, \widehat{I}_{qs}$)) qui sont illustrées dans la **Figure III.3**.

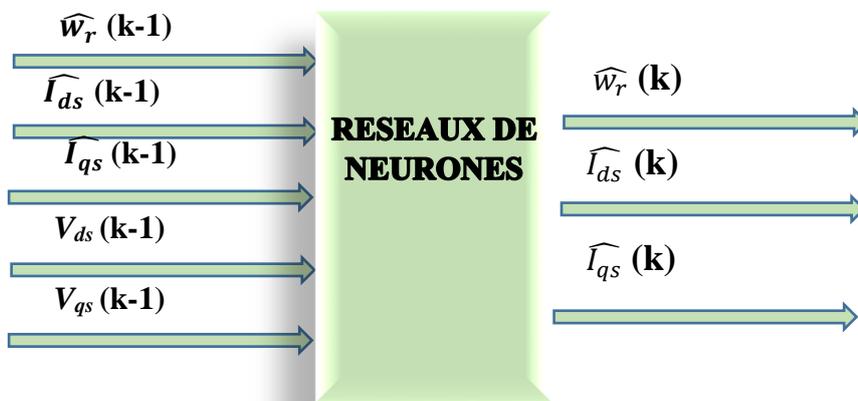


Figure III.3. : Modèle neuronal de la machine asynchrone.

III.3.1. Modélisation par perceptron multicouche (Multilayer Perceptron MLP)

L'objectif de l'utilisation d'un réseau de neurones pour modéliser une machine asynchrone est de créer un modèle non linéaire qui permet de déterminer les sorties, à savoir la vitesse de rotation du rotor w_r et les courants statoriques (i_{ds}, i_{qs}), à partir de données antérieures sur la vitesse et les courants de statoriques.

Les sorties $\widehat{w}_r, \widehat{I}_{ds}, \widehat{I}_{qs}$ du réseau sont calculées à partir des équations suivantes :

$$\widehat{w}_r = \psi_1(z_1 \cdot h + b_1) \quad (\text{III. 3})$$

$$\widehat{I}_{sd} = \psi_2(z_2 \cdot h + b_2) \quad (\text{III. 4})$$

$$\widehat{I}_{sq} = \psi_3(z_3 \cdot h + b_3) \quad (\text{III. 5})$$

Avec :

$$h = \varphi(w_1 \cdot \widehat{w}_r(k-1) + w_2 \cdot \widehat{I}_{sd}(k-1) + w_3 \cdot \widehat{I}_{sq}(k-1) + w_4 \cdot v_{sd}(k-1) + w_5 \cdot v_{sq}(k-1)) \quad (\text{III. 6})$$

Où :

$w_i = 1...3$: Sont les poids reliant la couche cachée à la couche d'entrée.

$z_i = 1...3$: Sont les poids reliant la couche cachée à la couche de sortie.

ψ : Est une fonction d'activation de type sigmoïde $f(x) = \frac{1}{1+e^{-ax}}$ est une fonction d'activation de type linéaire, ($f(x) = x$). b_i $i = 1...3$ est le biais.

Pour créer un modèle non linéaire avec un réseau MLP, les coefficients de pondération du réseau (poids et biais des neurones) doivent être ajustés pour que les sorties du réseau soient proches des sorties souhaitées. Ce processus est appelé apprentissage supervisé et utilise un ensemble d'apprentissage composé d'exemples pour minimiser une fonction de coût quadratique basée les différences entre les sorties du réseau et les sorties du système.

III.3.1.1. Choix de l'architecture du réseau MLP

Choisir le nombre de neurones par couche cachée dans un réseau de neurones est une tâche difficile sans règle systématique. Ce nombre est crucial car il détermine la capacité de calcul du réseau. Un nombre insuffisant de neurones compromet la capacité du réseau à résoudre le problème, tandis qu'un nombre excessif de neurones peut nuire à la généralisation.

Nous avons utilisé la méthode d'essai et d'erreur pour trouver le nombre optimal de neurones. Nous avons commencé avec des structures à une couche cachée et avons augmenté graduellement le nombre de neurones jusqu'à atteindre les performances souhaitées. Ce processus nous a permis d'évaluer plusieurs structures à une ou deux couches cachées, utilisant des fonctions d'activation sigmoïdales pour les neurones des couches cachées et des fonctions d'activation linéaires pour la sortie du réseau. Plusieurs essais ont été nécessaires pour trouver des architectures capables de généralisation.

III.3.1.2. Choix de l'algorithme d'adaptation

Parmi les nombreux algorithmes d'adaptation qui se trouvent dans le logiciel de Matlab, on a choisi l'algorithme de Levenberg-Marquardt qui est basé sur l'approximation de Newton et il est plus puissant par rapport aux autres algorithmes qui sont parfois très rapides, cependant, ils ne donnent pas des résultats acceptables. L'algorithme de Levenberg-Marquardt converge mieux mais demande plus de mémoire surtout lorsque la taille du réseau devient assez grande.

III.3.1. 3.Simulation

Dans cette étape les réseaux de neurones ont été construits en utilisant des fonctions d'activation sigmoïdales pour les neurones des couches cachées et des fonctions d'activation linéaires pour la sortie du réseau. Les poids et les biais des neurones ont été choisis aléatoirement à l'aide de l'algorithme de Levenberg Marquardt. Après avoir exploré plusieurs architectures de réseaux MLP, il a été constaté qu'un réseau à une couche cachée avec 5 neurones dans cette couche était capable de modéliser la machine.

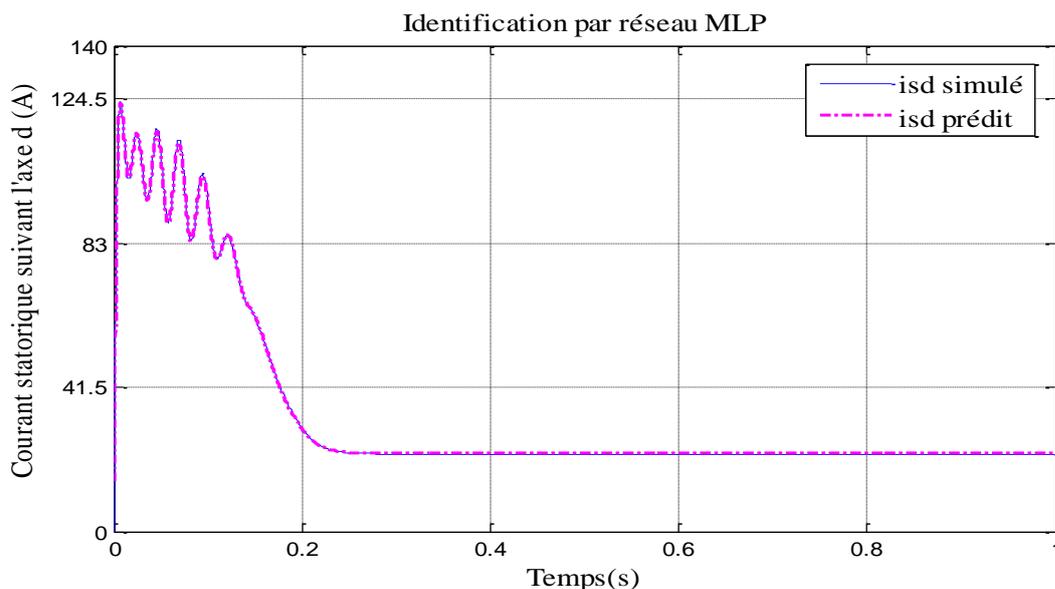


Figure. III.4. Courant statorique suivant l'axe d.

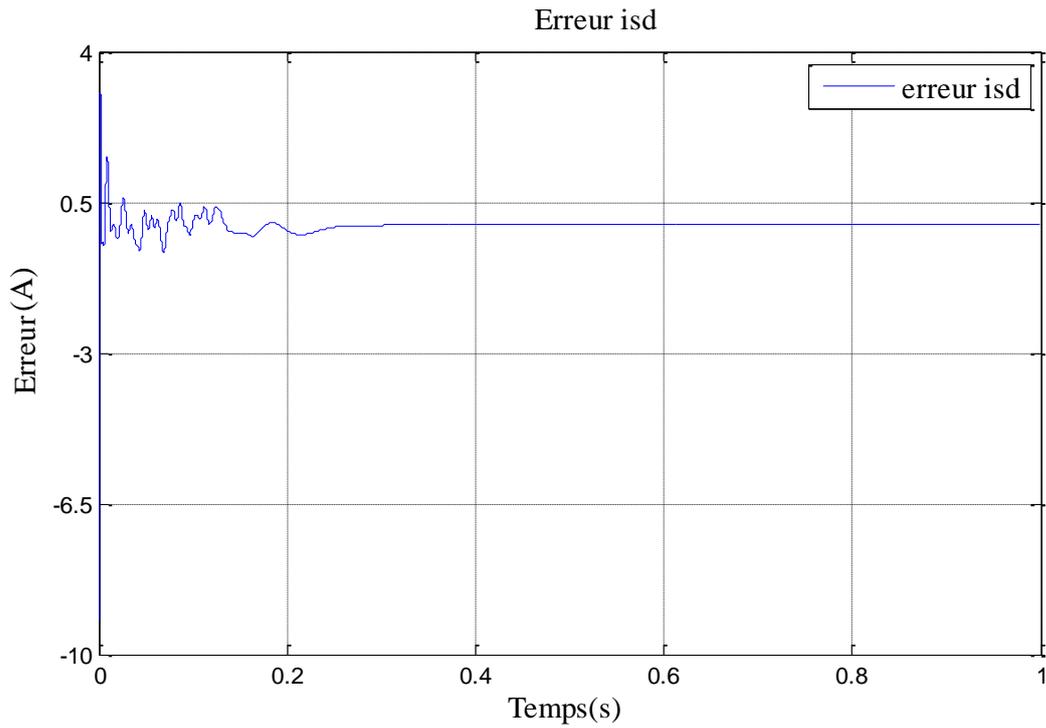


Figure.III.5 : Erreur d'identification par MLP de Courant statoriques suivant l'axe d.

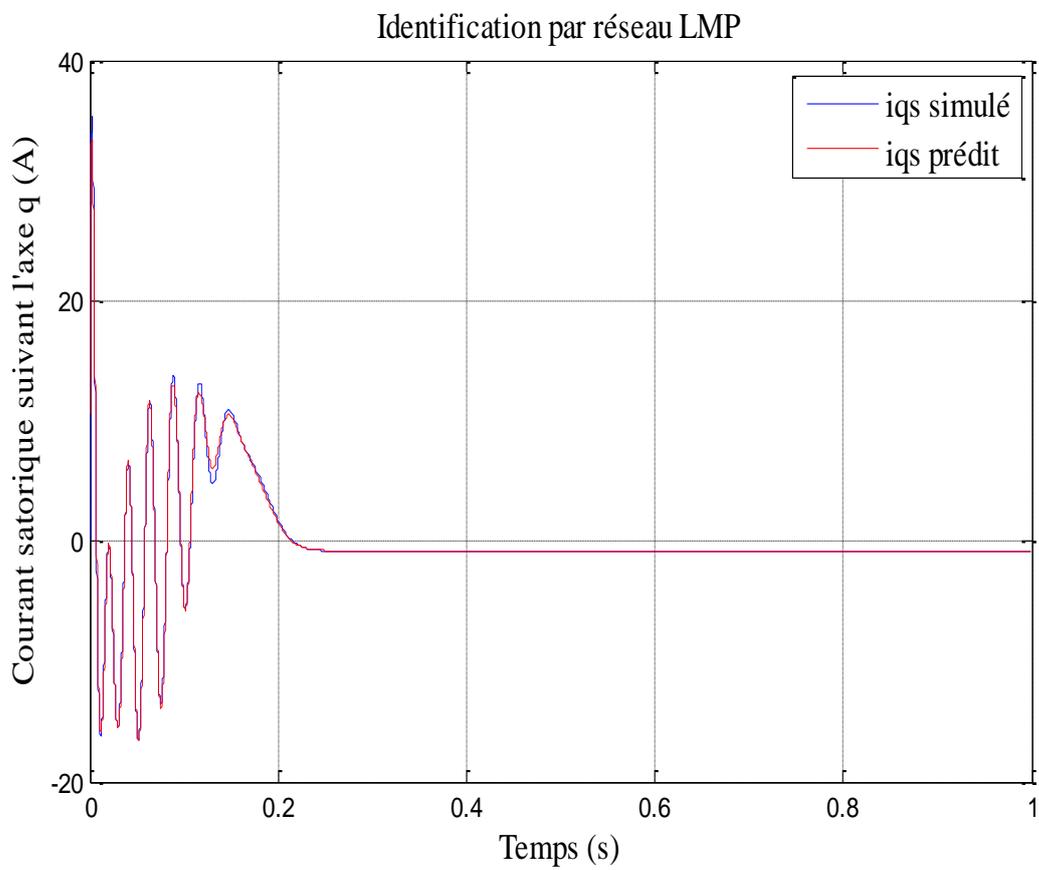


Figure. III.6 : Courant statoriques suivant l'axe q.

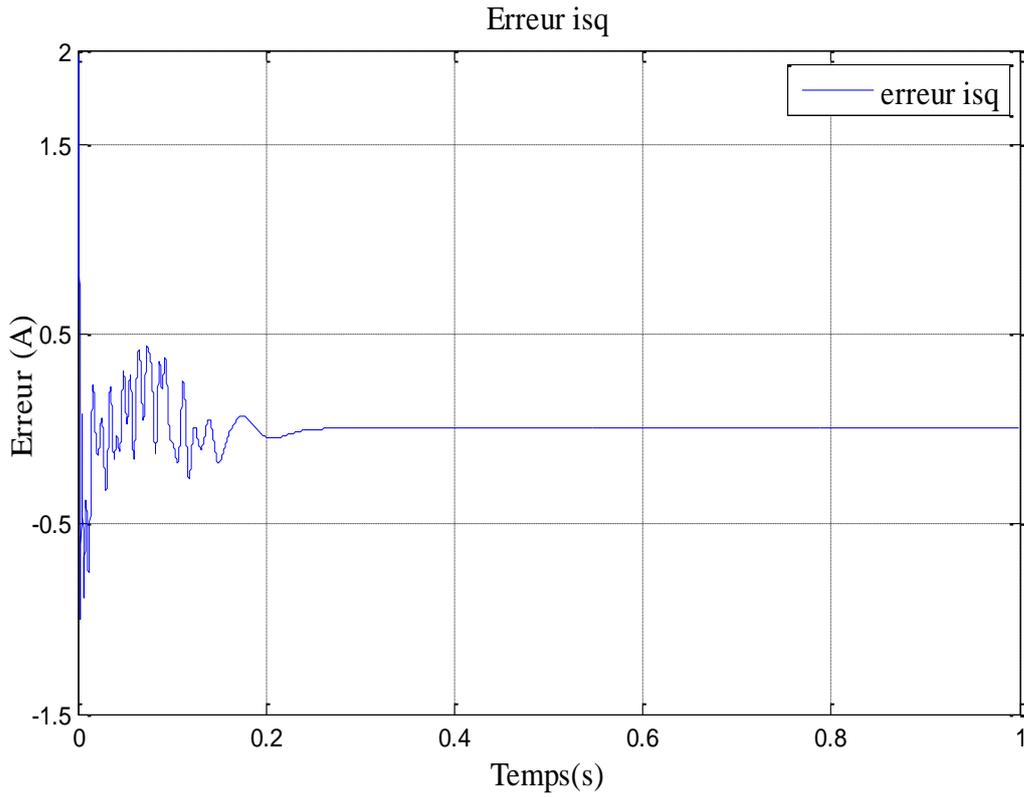


Figure.III.7 : Erreur d'identification par MLP de Courant statoriques suivant l'axe q.

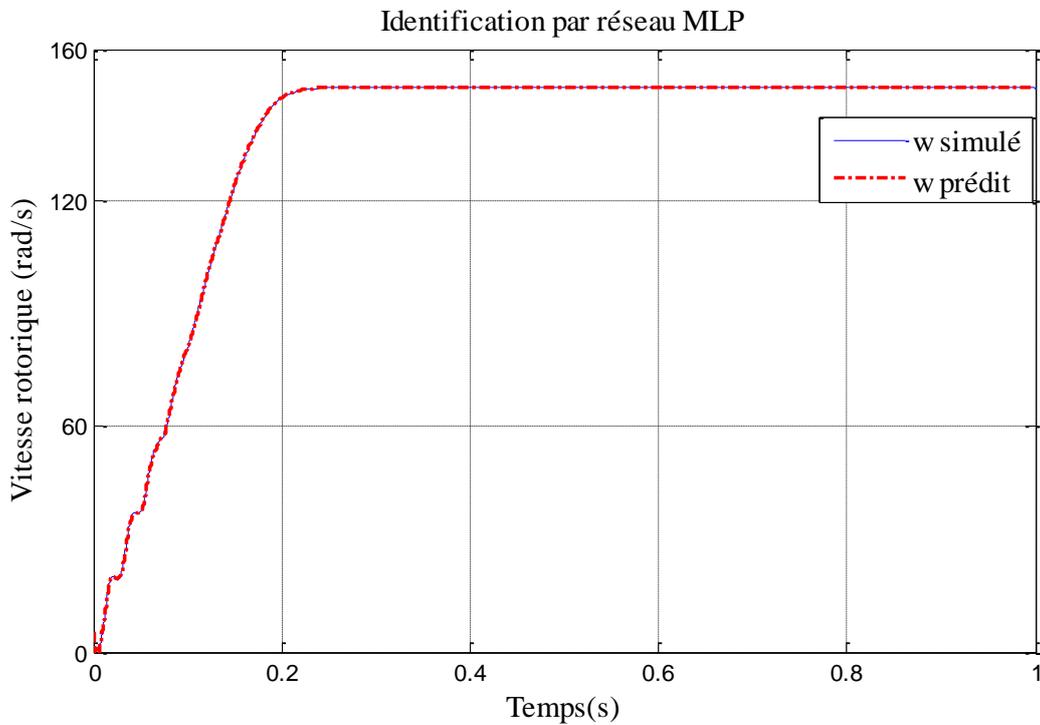


Figure.III.8 : La vitesse rotorique.

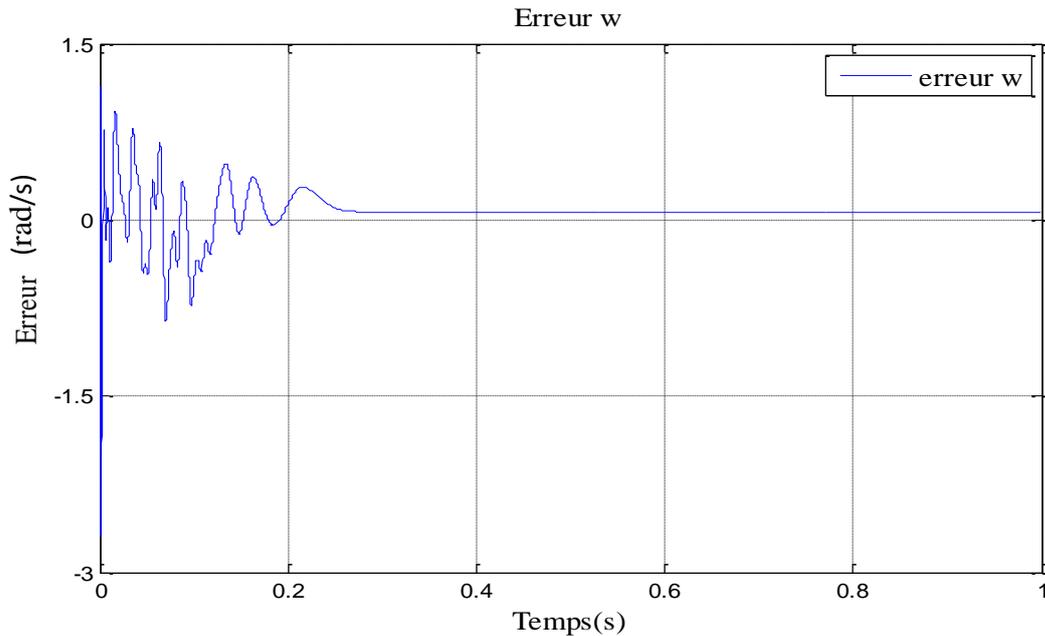


Figure.III.9 : Erreur d'identification par MLP de la vitesse rotorique.

Les figures **III.5**, **III.7** et **III.9** montrent une diminution de l'erreur au fil de la séquence d'apprentissage, ce qui indique que le réseau de neurones a bien assimilé les sorties de la machine.

Les figures **III.4**, **III.6** et **III.8** présentent des sorties de la machine et du réseau quasi identiques, démontrant ainsi l'efficacité du réseau à apprendre de son environnement.

III.3.2. Prédicteur de la machine asynchrone à base de RNA

Le réseau de neurones est entraîné à partir de données d'entrée et de sortie connues pour produire des prédictions précises de la vitesse de la rotation de la machine asynchrone en fonction du temps.

Dans cette simulation plusieurs essais ont été effectués en modifiant à chaque fois la taille des d'apprentissage et l'horizon de prédiction H, afin de déduire la plus proche au réel. Un réseau de neurones artificiels avec une seule couche cachée avec deux neurones est utilisé pour modéliser la relation entre les données d'entrée et la vitesse de rotation de la machine asynchrone. La couche cachée permet au réseau de neurones de capturer des relations non linéaires complexes entre les données d'entrée et la vitesse de rotation de la machine. Dans notre travail, on a fait quatre essais :

Essai 01

Taille = 10, H=10

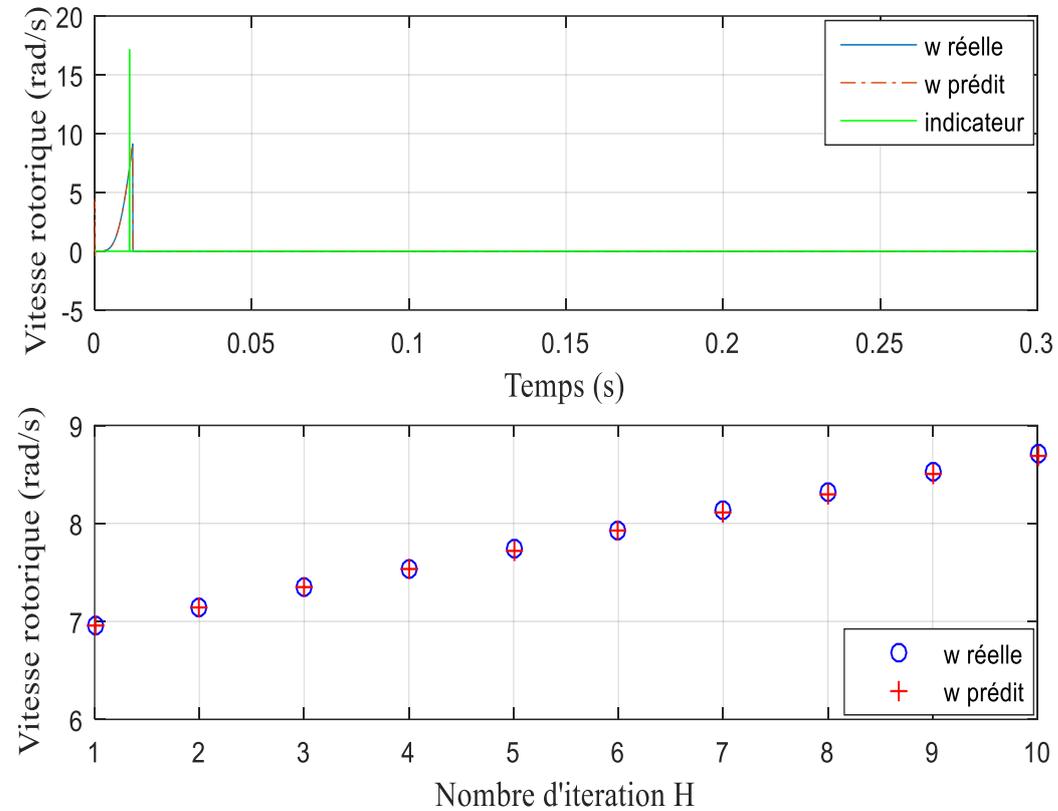


Figure.III.10.A : vitesse rotorique en fonction de nombre d'itération (taille=10, H=10)

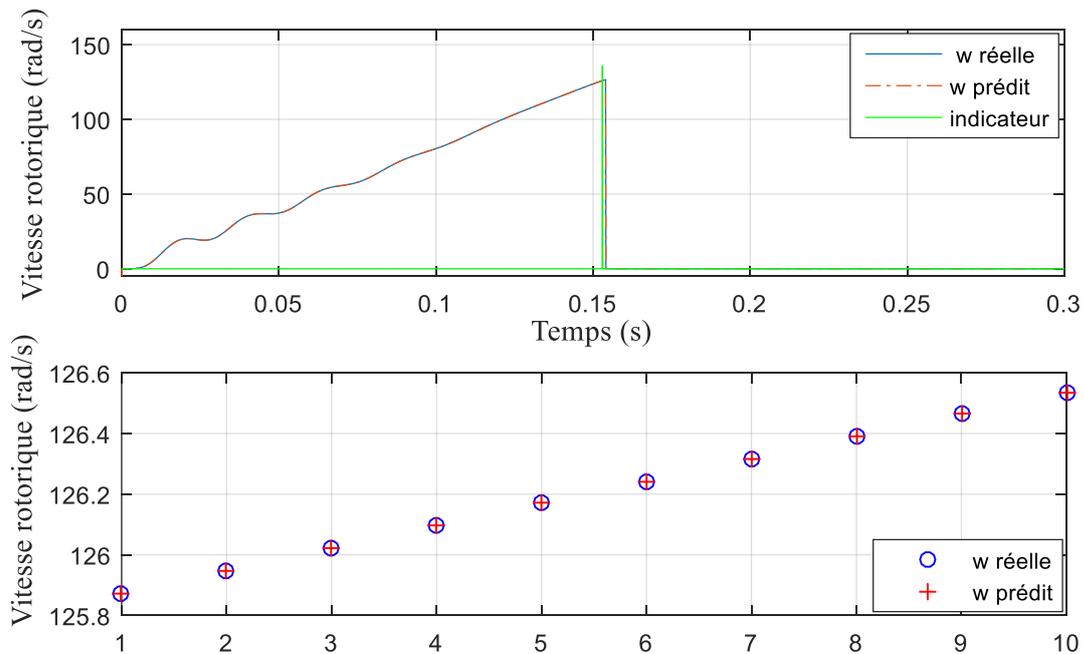


Figure.III.10.B : vitesse rotorique en fonction de nombre d'itération (taille=10, H=10)

Essai 02

Taille = 10, H=30

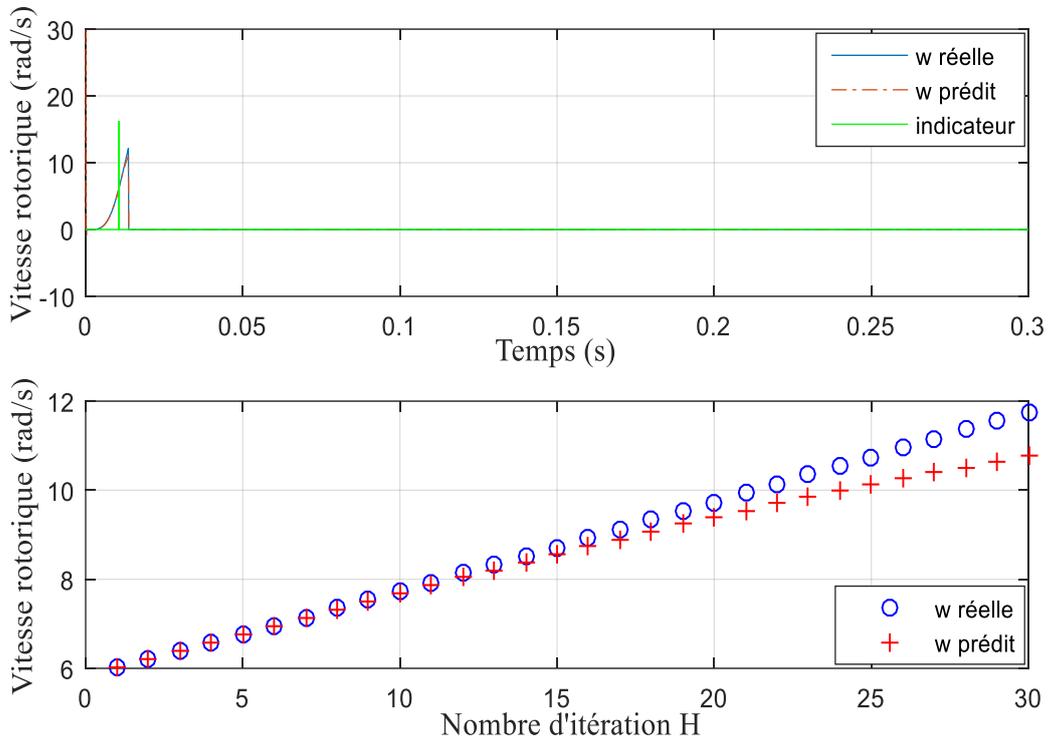


Figure.III.11.A : vitesse rotorique en fonction de nombre d'itération (taille=10, H=30)

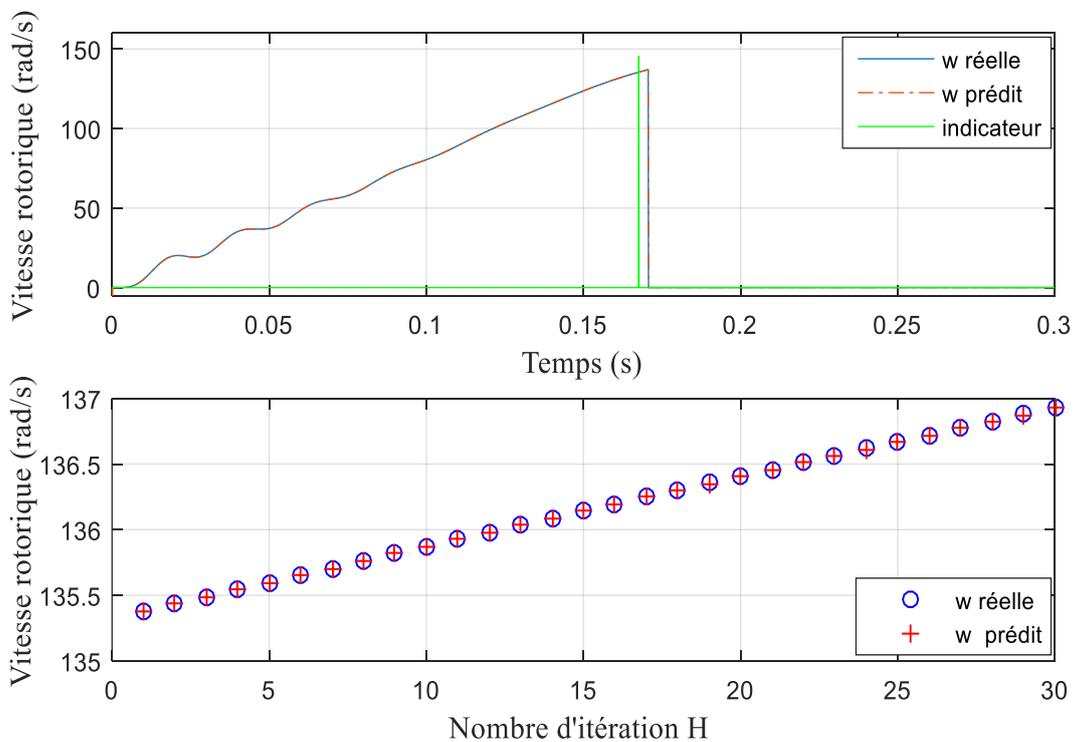


Figure.III.11. B : vitesse rotorique en fonction de nombre d'itération (taille=10, H=30)

Essai 03

Taille = 100, H=10

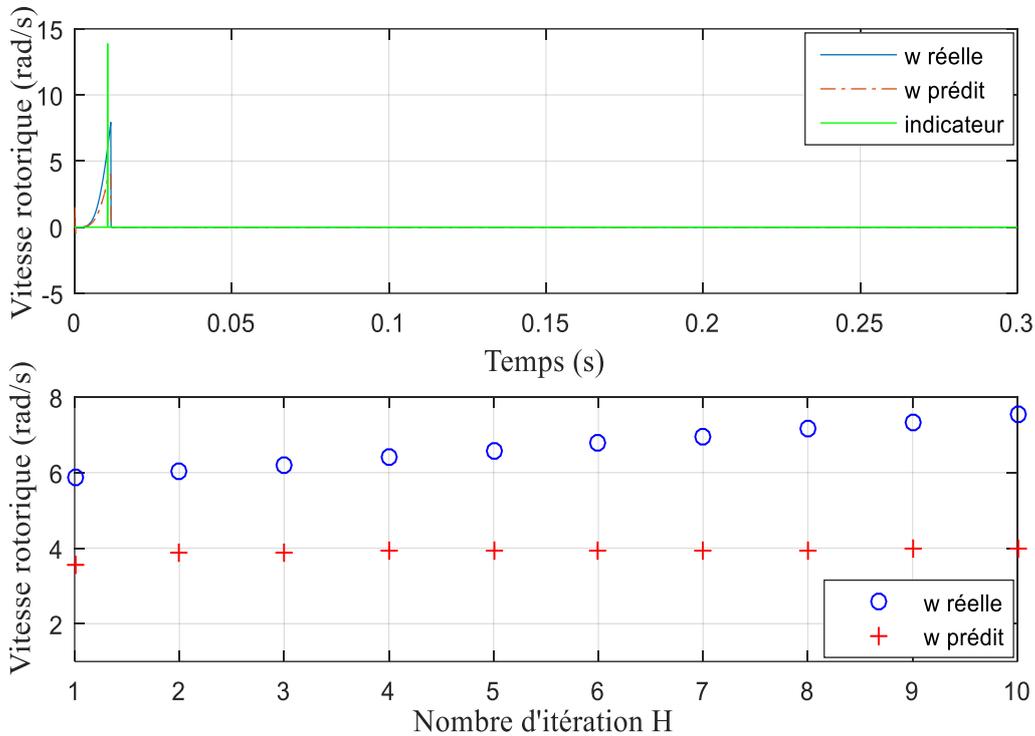


Figure.III.12. A : vitesse rotorique en fonction de nombre d'itération (taille=100, H=10)

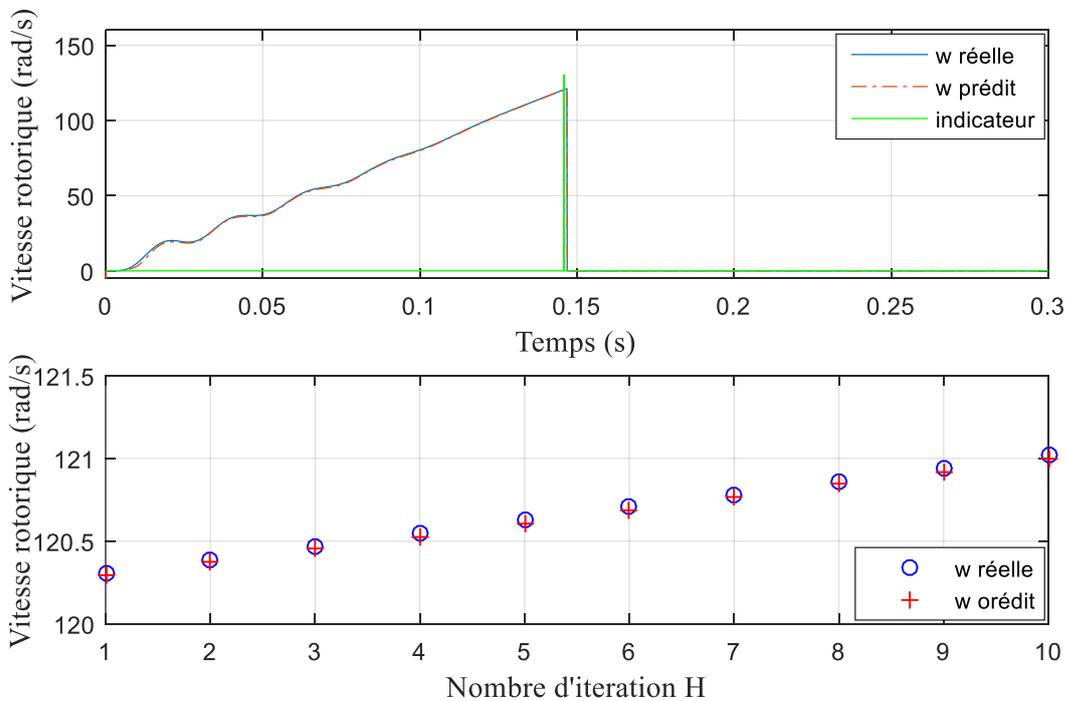


Figure.III.12. B : vitesse rotorique en fonction de nombre d'itération (taille=100, H=10)

Essai 04

Taille = 100, H=100

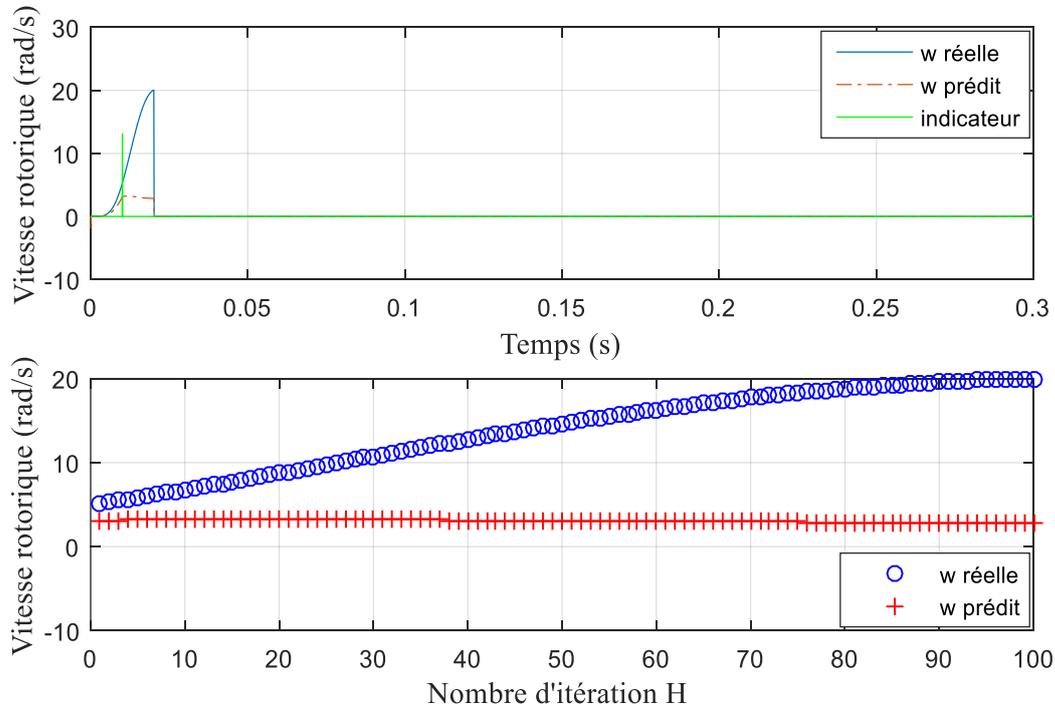


Figure.III.13 A : vitesse rotorique en fonction de nombre d'itération (taille=100, H=100)

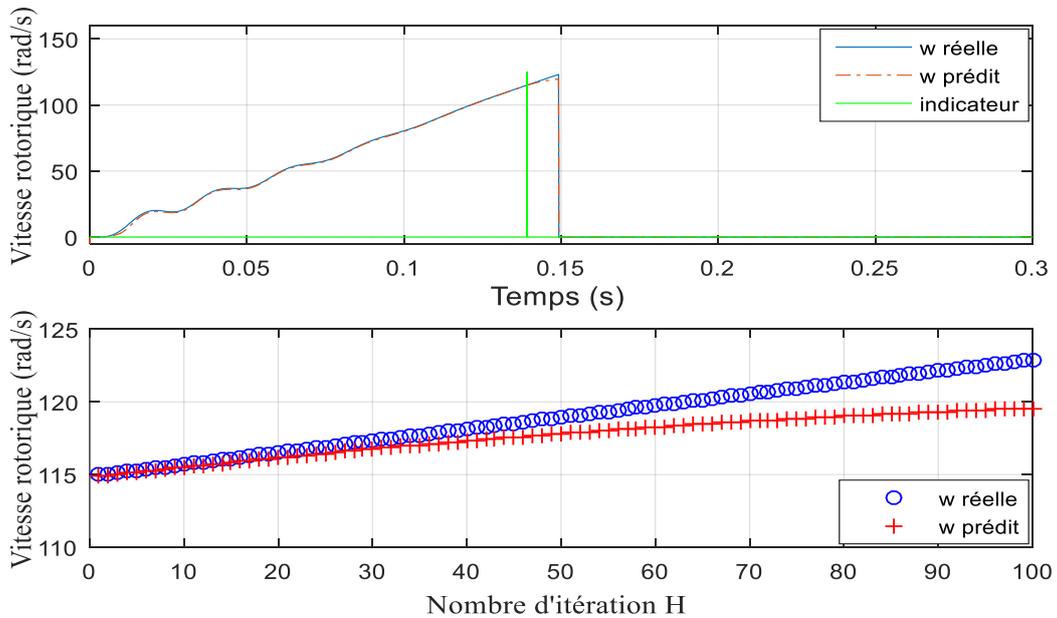


Figure.III.13. B : vitesse rotorique en fonction de nombre d'itération (taille=100, H=100)

III.3.3. Interprétation

Après avoir effectué notre simulation, nous avons constaté que notre programme fonctionnait de manière optimale, en produisant des prédictions de la vitesse rotoriques en accord avec les valeurs réelles. Bien que nous ayons observé une divergence au début de la simulation. Cela met en évidence l'importance de bien choisir la taille des données d'apprentissage et l'horizon de prédiction. Notre programme est capable de saisir les schémas et les tendances des données d'entrée, ce qui se traduit par des prévisions précises de la vitesse rotoriques.

III.4. Application de la commande prédictive basée sur les réseaux de neurones et l'algorithme ABC

Ce régulateur prédictif utilise un réseau de neurones artificiels pour prédire le comportement de la machine asynchrone dans un horizon H . Ce réseau de neurones est entraîné en ligne en utilisant l'historique des signaux mesurés issus de la machine. Pour pouvoir effectuer cet apprentissage, le réseau doit être de taille adéquate pour ne pas alourdir les calculs. En fait, c'est un calcul parallèle qui doit être utilisé.

Une fois l'apprentissage fait, L'algorithme ABC utilise des copies du réseau obtenu pour chaque abeille lors de la recherche de uds et uqs ou leurs incréments selon le cas. Chaque abeille propose ce qu'elle a trouvé et un choix de la meilleure proposition sera effectué selon le critère « Cout » qui évalue l'écart des sorties par rapport aux consignes. La **Figure III.14** présente un schéma de principe de ce régulateur.

Chaque abeille, dans l'algorithme ABC, choisira une paire de valeurs pour uds et uqs (c'est-à-dire un point dans l'espace de recherche). Le coût associé à ce point est calculé en utilisant la fonction de coût qui représente la performance du modèle. L'algorithme ABC utilise ensuite ces coûts pour mettre à jour les positions des abeilles dans l'espace de recherche en utilisant des mécanismes inspirés du comportement des abeilles (**Figure III.15**).

En résumé, l'apprentissage des réseaux de neurones artificiels et l'optimisation par algorithme ABC peuvent être utilisés pour la commande de la machine asynchrone.

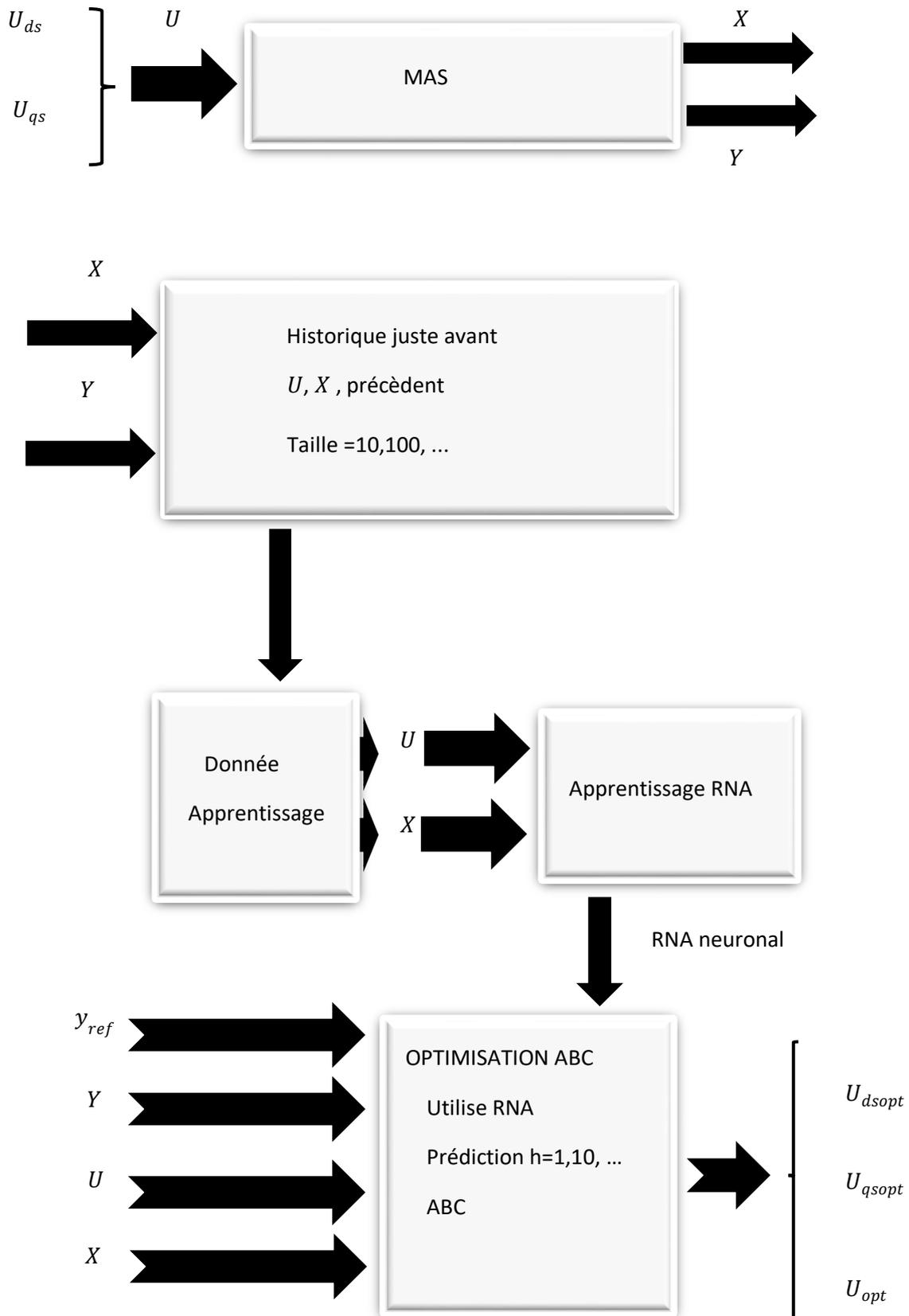


Figure III.14 : Structure de la commande prédictive via les algorithmes ABC

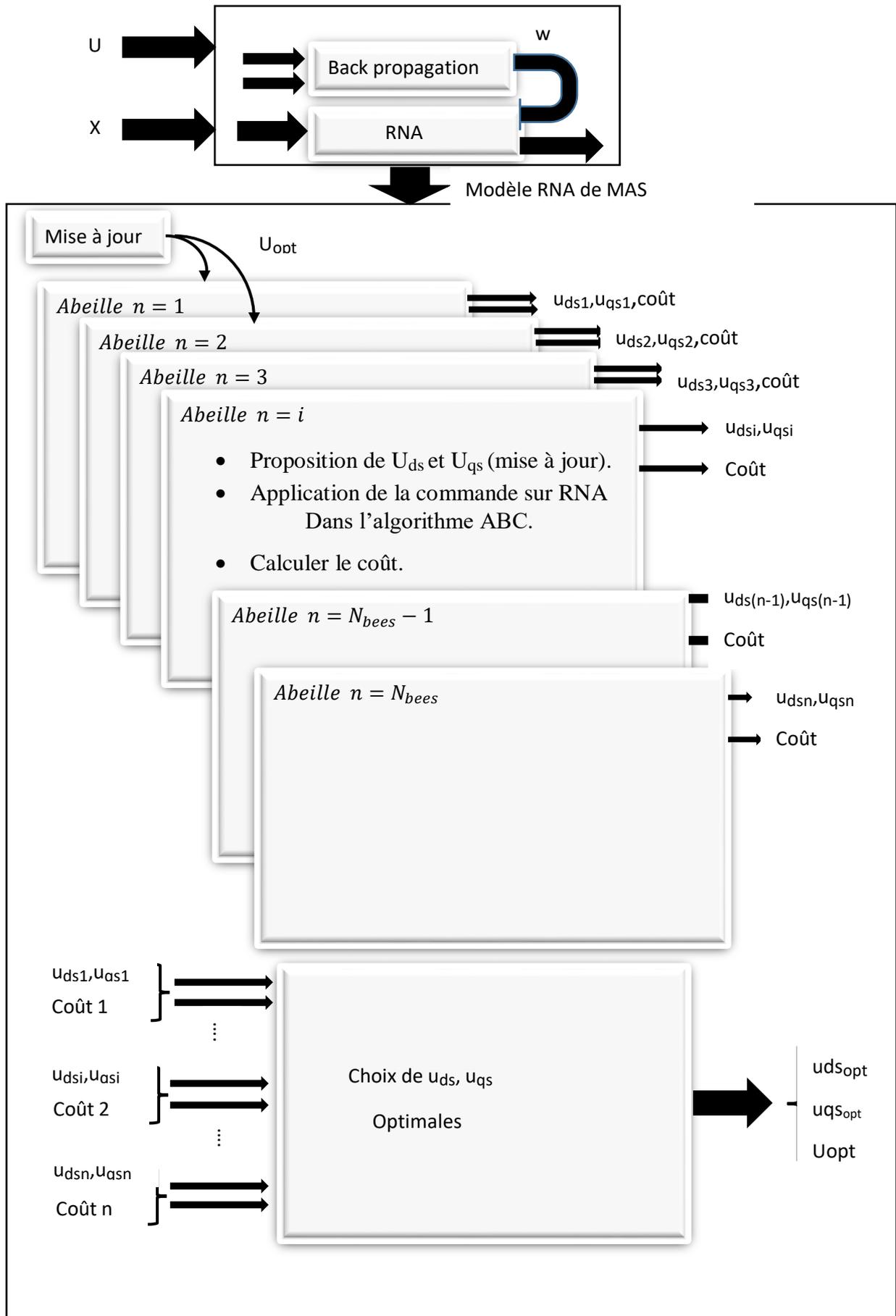


Figure III.15 : Schéma développé de l'algorithme ABC

III.4.1. Simulation

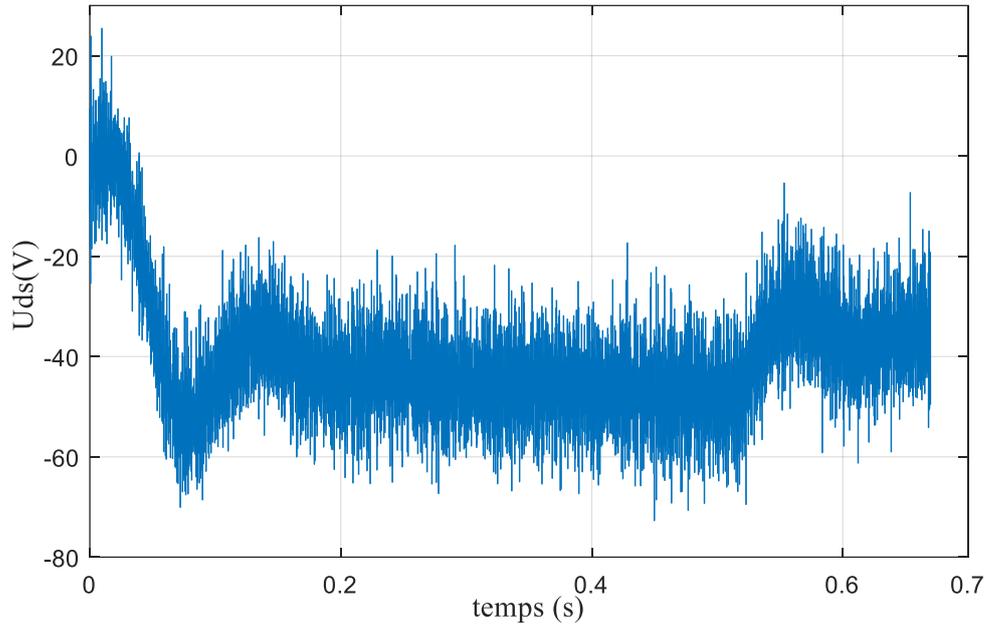


Figure III.16 : Tension statorique direct

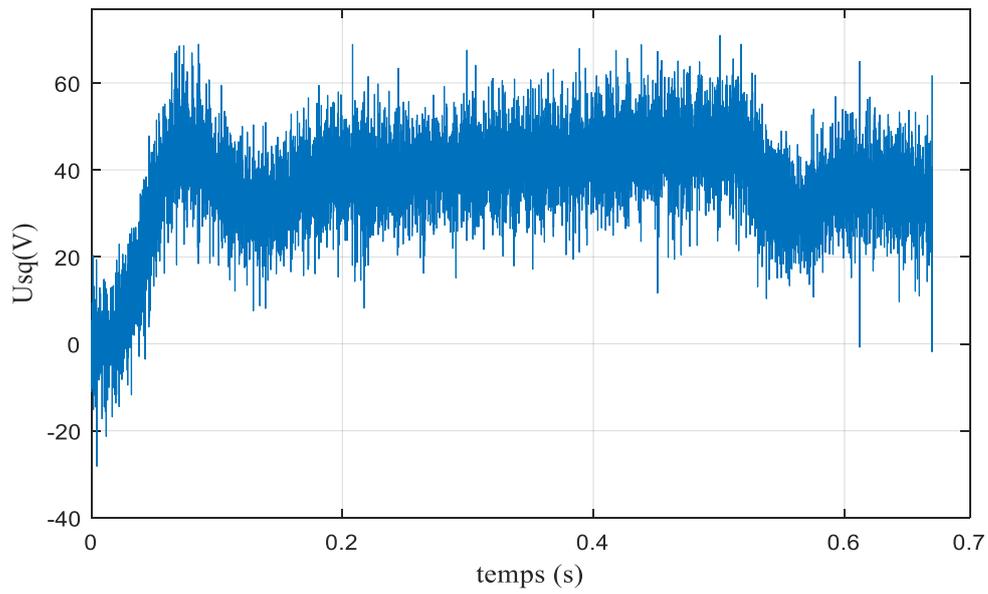


Figure III.17 : Tension statorique quadratique

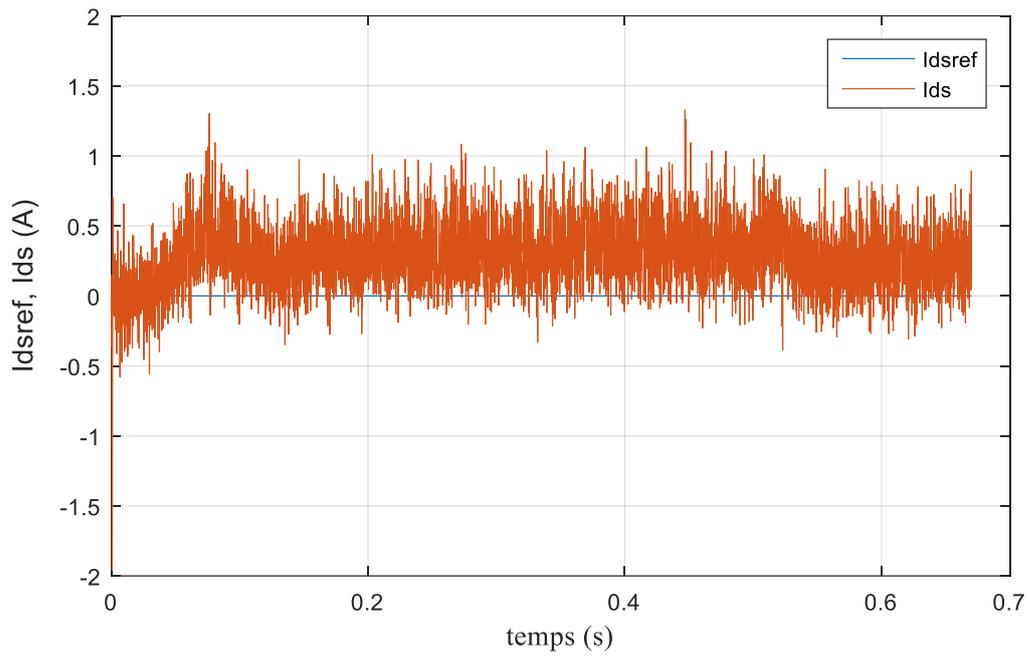


Figure III.18 : Courant statorique direct

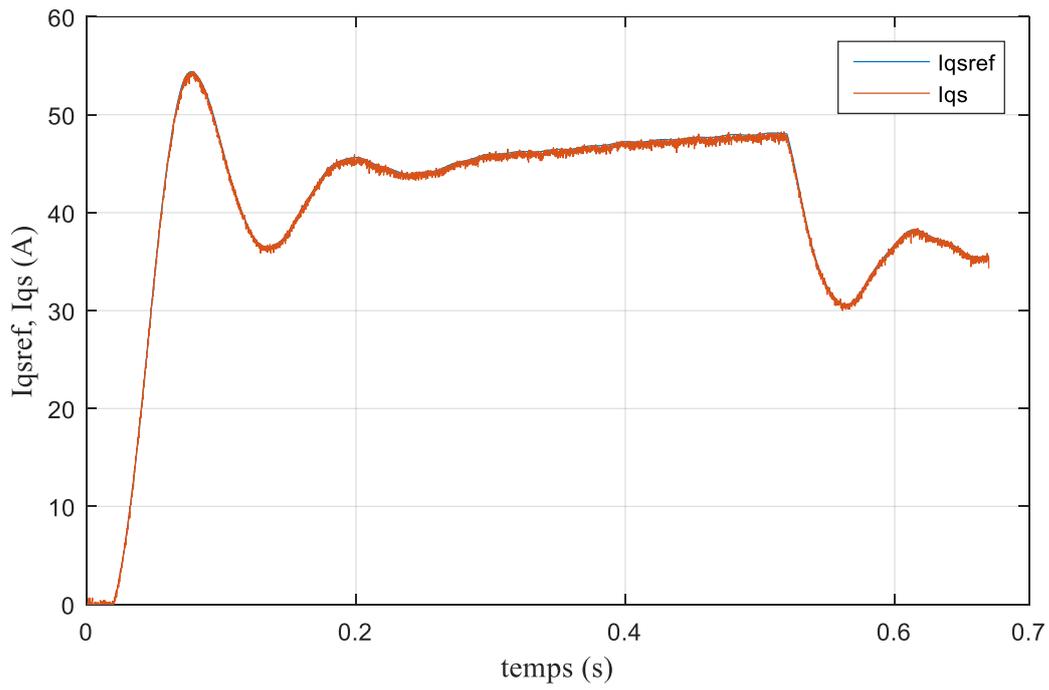


Figure III.19 : Courant statorique quadratique

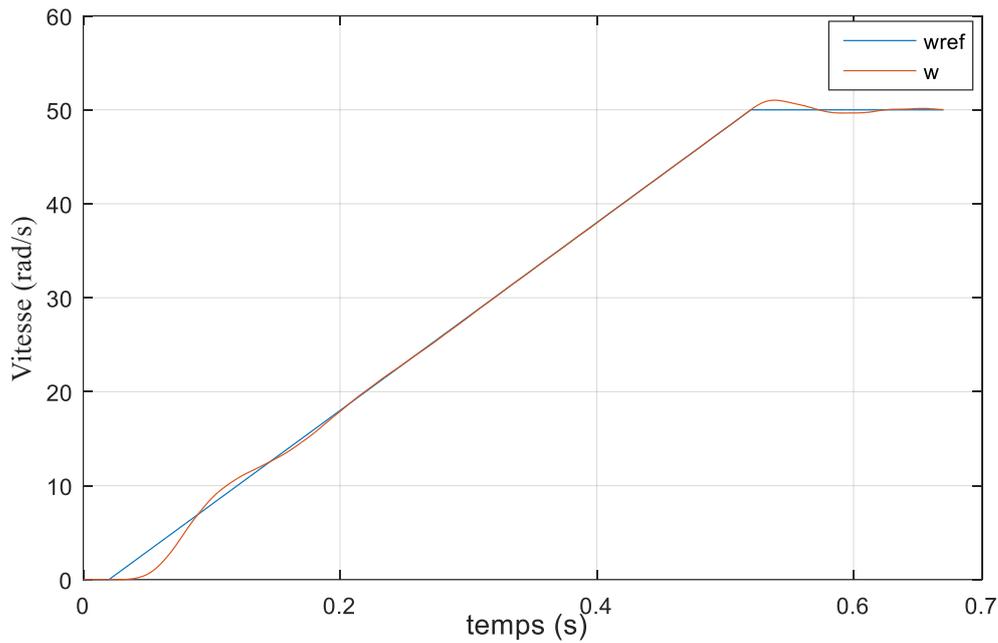


Figure III.20 : Vitesse rotorique

III.4.2. Interprétation

Les résultats de simulation obtenus sont prometteurs ce qui explique que l'utilisation de l'algorithme a permis de trouver les valeurs adéquates des signaux de commande de la machine asynchrone.

L'erreur de poursuite au début est due au fait que le réseau de neurones prend du temps pour capturer le comportement de la machine surtout qu'il n'a aucune connaissance préalable sur la machine sous commande.

Vu que l'algorithme ABC donne directement les valeurs des tensions statoriques (non pas les incréments), ces signaux présentent des variations rapides qui peuvent être réduites en utilisant des filtres adéquats ou bien choisir les paramètres du réglage associés au régulateur.

III.5. Conclusion

L'application de la commande sur la machine asynchrone à base d'un apprentissage par les réseaux de neurones et d'une optimisation par l'algorithme ABC est une approche innovante et puissante. Cette approche permet de commander les machines sans aucune connaissance préalable. Le régulateur obtenu peut se configurer pour bien conduire le comportement de la machine conformément à un certain critère de performances.

*CONCLUSION
GENERALE*

CONCLUSION GÉNÉRALE

Dans ce travail, nous avons mis en évidence, par simulation sous MATLAB, l'application de l'intelligence artificielle, en particulier des réseaux de neurones artificiels (RNA) et de l'algorithme ABC (Artificial Bee Colony), pour développer une commande prédictive pour les machines asynchrones.

Cette approche présente un potentiel significatif pour améliorer les performances de la machine asynchrone. En utilisant des modèles de prédiction basés sur les réseaux de neurones, nous sommes en mesure de prédire le comportement des machines asynchrones.

Pour atteindre ce but, nous avons présenté de manière générale les principes fondamentaux de la machine asynchrone. Nous avons utilisé la transformation de Park pour obtenir un modèle simple qui représente facilement le fonctionnement de la machine asynchrone à induction. Ensuite, nous avons exploré en détail les réseaux de neurones et l'algorithme ABC (Colony de l'abeille artificielle).

Enfin, pour résoudre notre problème d'optimisation des performances des machines asynchrones, nous avons utilisé des réseaux de neurones pour prédire leur comportement et l'algorithme ABC pour optimiser les paramètres de commande. Nous avons adopté une approche graduelle dans l'application de notre commande à la machine asynchrone en effectuant plusieurs tests afin de montrer l'efficacité de notre commande.

REFERENCES

BIBLIOGRAPHIQUES

REFERENCES BIBLIOGRAPHIQUES

- [1] **A. Tidjani** « Identification paramétrique d'un moteur asynchrone à cage d'écureuil par la technique optimisation d'essaim de particules » Mémoire de master académique en électrotechnique université Kasdi Merbah de OUARGULA **2016**.
- [2] **A. CHAHMI** « Identification paramétrique de la machine asynchrone dédiée au diagnostic». Thèse de doctorat en électrotechnique. Université USTHB **2017**.
- [3] **Abdallah Bezzini** « Commande Prédictive Non Linéaire en Utilisant Les Systèmes Neuro-Flous et les Algorithmes Génétiques » Magister en automatique. Université de Biskra **2015**.
- [4] : **K. BENSMAIL, A. GUENAOUI** « Étude d'une machine asynchrone à multi-étoile » Mémoire de master, Université de Bejaia Algérie, **2021**.
- [5] : **N. AMOURA, C. ARFI** « Commande prédictive d'un moteur asynchrone triphasé » Mémoire de master, Université de Bejaia Algérie, **2021**.
- [6] **Tidjani Abd el jaouad, Al Amoudi Badreddine** « Identification paramétrique d'un moteur asynchrone à cage d'écureuil par la technique « optimisation d'essaim de particules » » Mémoire Master Académique, Université Kasdi Merbah Ouargla **2016**.
- [7] **RABIA Lakhdar**, Modélisation, identification expérimentale et commande vectorielle d'un moteur asynchrone triphasé à cage'', mémoire de master, université d'Ouargla, **2015**.
- [8] **A. ACHOUR, T. ISSAD** « Application de la technique des modes glissants dans la commande de la machine asynchrone » Mémoire de master, Université de Bejaia Algérie, **2020**.
- [9] **M. K. TIBERMACHINE** « Commande vectorielle d'une machine asynchrone » Mémoire de master, Université de Biskra Algérie, **2019**.
- [10] **J. Lesenne F. Notelet G. Ségier**, « Introduction à l'électrotechnique approfondie » ; édition Technique et Documentations ; **1981**.
- [11] **L. BAGHLI**, « Contribution à la commande de la machine asynchrone, utilisation de la logique floue, des réseaux de neurones et algorithmes génétiques » Thèse de doctorat, Université Henri Poincaré, Nancy-I, France ; **1999**.
- [12] **Y. ELBIA**, « Commande floue optimisée d'une machine asynchrone à double alimentation et à flux orienté », mémoire de Magister, Université Batna, **2009**.

REFERENCES BIBLIOGRAPHIQUES

- [13] **T. Laamayad**, « Commande Optimale d'une Machine Asynchrone Apport de La Logique Floue », mémoire de Magister, Université Batna, **2008**.
- [14]. **D. KARABOGA, B. BASTURK**, « On the pereformance of artificial bee colony (abc) algorithm », *Appl. Soft Comput.*j.8(1),687-697,**2008**.
- [15]. **National Institute of Neurological Disorders and Stroke**. (2021). Neurological disorders. Retrieved from <https://www.ninds.nih.gov/Disorders/All-Disorders..>
- [16] **Karaboga D., Akay B.**, « A modified Artificial Bee Colony (ABC) algorithm for constrained optimization problems », Erciyes University, The Department of Computer Engineering, 38039 Melikgazi, Kayseri, Turkiye, **2010**.
- [17] **D. Karaboga, B. Akay, and C. Ozturk**, “Artificial bee colony (ABC) optimization algorithm for training feed-forward neural networks,” in International conference on modeling decisions for artificial intelligence, **2007**, pp. 318–329.
- [18] **K. Balasubramani and K. Marcus**, “Artificial bee colony algorithm to improve brain MR image segmentation,” *Int. J. Comput. Sci. Eng.*, vol. 5, no. 1, p. 31, **2013**.
- [19] **D. Karaboga, B. Gorkemli, C. Ozturk, and N. Karaboga**, “A comprehensive survey: artificial bee colony (ABC) algorithm and applications,” *Artif. Intell. Rev.*, vol. 42, no. 1, pp. 21–57, **2014**.
- [20] **C. TOUZET**, *les reseaux de neurones artificiels, introduction au connexionnisme : cours, exercices et travaux pratiques*, hal, **1992**.
- [21] **M. Chahreddine**, *Mise au point d'un programme d'optimisation pour le stockage d'energie thermique*, Oran : magister, Université d'Oran, Mohamed Boudiaf, **2013**.
- [22] **A. OUALI**, « Évaluation de la stabilité transitoire des réseaux électriques par les réseaux de neurones artificiels et les plans d'expériences » Mémoire de Magister, Université des sciences et de la technologie Houari Boumediene, **2008**.
- [23] **I. KADA and M. B. Chaïmaa**, *Diagnostic des défauts de la MAS à l'aide des réseaux de neurones artificiels*, BLIDA: Mémoire de Master, Université SAAD DAHLAB, **2020**.
- [24] **M. C. LAOUBI**, *Contrôle d'un pendule inversé par un réseau de neurones artificiels*, Bouira: Mémoire de master, Université Akli Mohand Oulhadj, **2018**.

REFERENCES BIBLIOGRAPHIQUES

- [25] **M. SIHALI** , Diagnostic de défauts à base d'observateurs à entrées inconnues : Application au système hydraulique, TIZI-OUZOU: Mémoire de Master, UNIVERSITE MOULOUD MAMMERI, **2019**.
- [26] **M. Y. AMMAR**, mise en œuvre de reseaux de neurones pour la modelisation de cinetiques reactionnelles en vue de la transposition batch/continu, Tunisie: PHD THESE, Ecole Nationale d'Ingénieurs de Sfax, **2007**.
- [27] **O. SID**, Diagnostic de défauts de la machine asynchrone à cage par les réseaux de neurones, Alge: Mémoire de magister, Université des Sciences et de la Technologie Houari Boumediene, **2011**.
- [28] **Y. Djeriri**, Les Réseaux de Neurones Artificiels, university of sidi-bel-abbes, **2017**.
- [29] **K. MENDAZ**, Adaptation des commandes floues et Neuro-floue aux actionnaires asynchrones en mode dégradé : application au transport ferroviaire, SIDI-BEL-ABBES: Thèse de Doctorat, UNIVERSITE DJILLALI LIABES, **2015**.
- [30] **G. Dreyfus**, Réseaux de neurones méthodologie et application, Edition EYROLLES, **2004**.
- [31] **F. MEDJILI**, Modélisation par Réseaux de Neurones Artificiels (RNA) et commande Prédictive non linéaire d'une station de production d'eau froide, ANNABA: Mémoire de magister, UNIVERSITE BADJI MOKHTAR, **2010**.
- [32] **D. E. CHAOUCH**, Contrôle robuste des systèmes dynamiques non linéaires incertains par des approches de l'intelligence artificielle, Oran : Thèse de Doctorat en Sciences, Université d'Oran Mohamed Boudiaf, **2016**.
- [33] **M. Rima**, Apprentissage des réseaux de neurones MLP par une méthode hybride à base d'une métaheuristique, Université Larbi Ben M'hidi Oum-El-Bouaghi, **2019**.
- [34] **M. BENRAHMOUNE**, Diagnostic des défaillances d'une turbine à gaz à base des réseaux de neurones artificiels pour l'amélioration de leur système de détection des vibrations, Djelfa: PhD Thèse, Université Ziane Achour , **2017**.

ANNEXES

ANNEXES

Les paramètres de la machine asynchrone que nous avons utilisée dans nos simulations sont :

Résistance d'une phase statorique	$R_s = 0.485 \quad \Omega$
Résistance d'une phase rotorique	$R_r = 0.485 \quad \Omega$
Inductance cyclique statorique	$L_s = 0.0274$
Inductance cyclique rotorique	$L_r = 0.0274$
Inductance mutuelle cyclique propre	$M = 0.0258$
Moment d'inertie	$J = 0.031 \text{ kg.m}^2$
Coefficient de frottement	$f = 0.081 \text{ N.s/rad}$
Nombre de pair de pole	$p = 2$
Couple résistante	$C_r = 0$

```
function dx=mas(u)

ids=u(1);
iqs=u(2);
phidr=u(3);
phiqr=u(4);
w=u(5);
uds=u(6);
uqs=u(7);
ws=u(8);
Cr = 0;
Rs=0.485;
Rr=0.3805;
Ls=0.0274;
Lr=0.0274;
M=0.0258;
J=0.031;
f=0.081;
p=2;
sigma=1-(M^2/Lr/Ls);
gamma=(Rs+Rr*M^2/Lr^2)/(sigma*Ls);
k=M/sigma/Ls/Lr;
Tr=Lr/Rr;
Ce=(p*M/Lr)*(phidr*iqs-phiqr*ids);
```

ANNEXES

```
isddot=-gamma*ids+ws*iqs+(k/Tr)*phidr+p*k*w*phiqr+(1/sigma/Ls)*uds;
isqdot=-gamma*iqs-ws*ids+(k/Tr)*phiqr-p*k*w*phidr+(1/sigma/Ls)*uqs;
phirddot=(M/Tr)*ids-(1/Tr)*phidr+(ws-p*w)*phiqr;
phirqdot=(M/Tr)*iqs-(1/Tr)*phiqr-(ws-p*w)*phidr;
wdot=1/J*(Ce-Cr)-(f/J)*w;
dx=[isddot isqdot phirddot phirqdot wdot]';
return
```

*TABLE DES
MATIÈRES*

Table de matière

Liste des abréviations

Liste des tableaux

Liste des figures

Liste des annexes

Introduction générale	1
Chapitre I : Modélisation de la machine asynchrone	2
I.1. Introduction.....	2
I.2. Présentation et constitution de la machine asynchrone.....	2
I.3. Principe de fonctionnement d'une machine asynchrone.....	3
I.4. Modélisation de la machine asynchrone.....	4
I.4.1. Hypothèses simplificatrices.....	4
I.4.2. Mise en équation de la machine asynchrone.....	4
I.4.2.1. Équations des tensions.....	5
I.4.2.2. Équations magnétiques.....	6
I.4.2.3. Équations du couple électromagnétique.....	7
I.4.2.4. Équation mécanique.....	8
I.4.3. Transformation d. Park.....	8
I.4.3.1. Application de la transformation de Park à la MAS.....	9
I.4.3.2. Application aux équations des tensions.....	10
I.4.3.3. Application aux équations des flux.....	11
I.4.3.4. Application aux équations mécanique et électromagnétique.....	11
I.5. Différents référentiels.....	12
I.5.1. Définition des différents référentiels.....	12
I.5.1.1. Référentiel fixe par rapport au stator.....	12
I.5.1.2. Référentiel fixe par rapport au rotor.....	12
I.5.1.3. Référentiel fixe par rapport au champ tournant.....	12
I.6. Représentation d'état de la machine asynchrone.....	13
I.7. Simulation de la machine asynchrone sous Matlab /Simulink.....	14
I.8. Conclusion.....	15

Chapitre II : Intelligence artificielle.....	16
II.1. Introduction.....	16
II.2. Les réseaux de neurones (RNA).....	16
II.2.1. Définition.....	16
II.2.2. Neurone biologique.....	16
II.2.3. Neurones Artificiels.....	17
II.2.4. Modélisation générale.....	17
II.2.5. Propriétés des réseaux de neurones.....	20
II.2.6. Architecture des réseaux de neurones.....	20
II.2.6.1. Réseau de neurones non bouclé (feed-forward).....	20
II.2.6.2. Réseau de neurones bouclé (feed-back).....	21
II.2.7. Types d'apprentissage des réseaux de neurones.....	21
II.2.7.1. Apprentissage supervisé.....	21
II.2.7.2. Apprentissage semi- supervisé.....	21
II.2.7.3. Apprentissage non supervisé.....	21
II.2.8. Règles d'apprentissage.....	22
II.2.8.1. La règle de Hebb.....	22
II.2.8.2. Règle de Delta (Windrow -Hoff).....	22
II.2.8.3. Règle d'apprentissage compétitive.....	22
II.2.8.4. Règle de correction d'erreurs.....	22
II.2.9. Domaines d'application.....	22
II.3. L'algorithme de colonies d'abeilles artificielles.....	23
II.3.1. Principe de l'algorithme ABC.....	23
II.3.2. Le processus de l'algorithme ABC.....	25
II.3.2.1. Initialisation de l'algorithme.....	25
II.3.2.2. Pseudo codes de l'algorithme ABC.....	26
II.3.3. Applications de l'algorithme ABC.....	27
II.4. Conclusion.....	28
Chapitre III : Application de la commande sur la machine asynchrone	29
III.1. Introduction.....	29
III.2. Simulation numérique.....	29
III.3. Modélisation de la machine asynchrone par les réseaux de neurones.....	31
III.3.1. Modélisation par perceptron multicouche (Multilayer Perceptron MLP).....	32

III.3.1.1. Choix de l'architecture du réseau MLP	32
III.3.1.2. Choix de l'algorithme d'adaptation.....	33
III.3.1. 3.Simulation.....	33
III.3.2. Prédicteur de la machine asynchrone à base de RNA.....	36
III.3.3. Interprétation.....	41
III.4. Application de la commande prédictive basée sur les réseaux de neurones et l'algorithme ABC.....	41
III.4.1. Simulation.....	44
III.4.2. Interprétation.....	46
III.5. Conclusion.....	46
Conclusion générale.....	47
Références bibliographiques.....	48
Table des matières	
Annexes	

Résumé

Ce mémoire explore l'utilisation de l'intelligence artificielle, en particulier les réseaux de neurones (RNA) et l'algorithme ABC (Artificial Bee Colony), pour développer une commande prédictive pour les machines asynchrones.

Les réseaux de neurones sont utilisés pour modéliser et prédire le comportement de la machine en fonction de variables d'entrée. L'algorithme ABC est utilisé pour optimiser les paramètres de commande et améliorer les performances de la machine asynchrone. Des résultats de simulation sous Matlab prouvent l'efficacité de cette approche.

Ce mémoire propose une méthode créative pour améliorer la commande des machines asynchrones en combinant les techniques d'intelligence artificielle avec les principes de l'électromécanique.

Mots-clés : Machine asynchrone, réseaux de neurones artificiels (RNA), algorithme ABC (Artificial Bee Colony)

ABSTRACT

This thesis explores the use of artificial intelligence, particularly neural networks (RNA) and the Artificial Bee Colony (ABC) algorithm, to develop predictive control for asynchronous machines.

Neural networks are used to model and predict the behavior of the machine based on input variables. The ABC algorithm is used to optimize control parameters and improve the performance of the asynchronous machine. Simulation results using Matlab prove the effectiveness of this approach.

This thesis proposes a creative method for improving the control of asynchronous machines by combining artificial intelligence techniques with electromechanical principles.

Keywords: Asynchronous machine, artificial neural networks (RNA), Artificial Bee Colony (ABC) algorithm.

ملخص

تستكشف هذه الأطروحة استخدام الذكاء الاصطناعي، ولا سيما الشبكات العصبية (RNA) وخوارزمية ABC (مستعمرة النحل الاصطناعية)، لتطوير التحكم التنبؤي للألات غير المتزامنة. تُستخدم الشبكات العصبية لنمذجة سلوك الآلة والتنبؤ به بناءً على متغيرات الإدخال. تُستخدم خوارزمية ABC لتحسين معلمات التحكم وتحسين أداء الجهاز غير المتزامن. نتائج المحاكاة تحت MATLAB تثبت فعالية هذا النهج. تقترح هذه الأطروحة طريقة إبداعية لتحسين التحكم في الآلات غير المتزامنة من خلال الجمع بين تقنيات الذكاء الاصطناعي ومبادئ الميكانيكا الكهروميكانيكية.

الكلمات المفتاحية: آلة غير متزامنة، شبكات عصبية اصطناعية RNA، خوارزمية ABC اصطناعية مستعمرة النحل

