Normandie Université

THÈSE

Pour obtenir le diplôme de doctorat

Spécialité AUTOMATIQUE, SIGNAL, PRODUCTIQUE, ROBOTIQUE

Préparée au sein de l'Université de Rouen Normandie

Synthèse d'un contrôleur prédictif auto-adaptatif réglé par réseau de neurones artificiels

Présentée et soutenue par
Houssam MOUMOUH

Thèse soutenue publiquement le (31/03/2021) devant le jury composé de		
Mme. Sihem TEBBANI	Professeur / CentraleSupelec / Gif-sur-Yvette	Rapporteuse
M. Didier GEORGES	Professeur / Grenoble INP-ENSE3 / Grenoble	Rapporteur
M. Pedro CASTILLO GARCIA	Chargé de recherche / CNRS / Compiègne	Examinateur
M. Stéphane GRIEU	Professeur / Univ. Perpignan / Perpignan	Examinateur
M. Ahmed CHEMORI	Chargé de recherche / CNRS / Montpelier	Examinateur
M. Nicolas LANGLOIS	DR.HDR, ESIGELEC, Saint-Etienne-du-Rouvray	Directeur de thèse

Thèse dirigée par : Nicolas LANGLOIS, IRSEEM

UNIVERSITÉ DE ROUEN NORMANDIE

MIIS

ESIGELEC irseem

Avant-propos

C'est avec un grand plaisir que je réserve ces quelques lignes en signe de gratitude et de profonde reconnaissance à toutes celles et ceux qui, de près ou de loin, ont contribué à la réussite de mes travaux de thèse CIFRE effectués en collaboration entre SEGULA Technologies et l'Institut de Recherche en Systèmes Électroniques EMbarqués (IRSEEM) de l'ESIGELEC Rouen, France.

Je tiens à exprimer ma très vive reconnaissance envers Monsieur Nicolas LANGLOIS, Enseignant-Chercheur HDR, directeur de la formation par la recherche et responsable du pôle A& S, mon directeur de thèse, pour ses qualités pédagogiques et scientifiques, pour ses encouragements incessants ainsi que pour pour m'avoir accordé sa confiance durant ces trois années de thèse. Ses conseils et son aide ont été les plus utiles pour l'aboutissement de ce travail. J'exprime aussi ma gratitude à Monsieur Madjid HADDAD, mon responsable scientifique et responsable recherche et innovation en Ile de France de SEGULA Technologie, pour sa qualité de suivi, pour ses encouragements durant les périodes les plus difficiles, pour sa flexibilité, et surtout pour sa confiance durant les trois années de thèse.
J'adresse toute ma reconnaissance à Madame Tebbani Sihem Professeure à Centrale SUPELEC et Monsieur Georges Didier Professeur à Grenoble INP d'avoir accepté d'être rapporteurs de mes travaux. Je les remercie vivement de leur lecture approfondie de mon mémoire et leurs remarques constructives et enrichissantes.

Je remercie profondément Monsieur Chemori Ahmed Chargé de recherche au CNRS de Montpelier, Monsieur Castillo Garcia Perdro Chargé de recherche au CNRS compiègne, et Monsieur Grieu Stephane Professeur à l'université de Perpignan d'avoir accepté avec grand plaisir de participer à ce jury et d'examiner ce travail.

Mes remerciements s'adressent à tous les membres de l'IRSEEM, plus particulièrement à Monsieur Ghaleb HOBLOS, Chef du département Génie Electrique et Energie au sein de l'ESIGELEC de m'avoir permis d'enseigner au sein de sa dominante Automatique et Robotique Industrielle.

Je suis reconnaissante envers mes amis et collègues de l'IRSEEM pour qui j'ai beaucoup d'amitié. Je les remercie pour leur amicale présence et pour la sympathie qu'ils m'ont constamment témoignée. Je leur exprime ici toute ma gratitude.

Enfin, J'exprime ma profonde gratitude à mes très chers parents et les personnes qui comptent le plus dans ma vie, à mes chèrs sœfrères, à mon grand amour, ma femme et la couleur de ma vie. Je leur dédie le plus beau de mes plus belles idées et tous mes succès. À tous ceux que j'aime, merci !

Table des matières

Liste des figures **vii**

Liste des tableaux **x**

Publications de l'auteur **xi**

Glossaire **xiii**

Introduction générale **1**

Chapitre 1
État de l'art sur la commande prédictive

1.1 Introduction 5
1.2 Notions de base sur la commande prédictive 5
1.2.1 Historique 5
1.2.2 Domaines d'application 7
1.2.3 Principe de fonctionnement de la MPC 8
1.2.4 Choix des paramètres de réglage de la MPC 9
1.3 État de l'art sur les approches de réglage de la commande prédictive 11
1.3.1 Les approches analytiques 12
1.3.2 Les approches expérimentales 13
1.3.3 Les approches heuristiques 14
1.3.4 Les approches méta-heuristiques 18
1.4 Conclusion 18

Chapitre 2
Apprentissage et dimensionnement des réseaux de neurones artificiels

2.1 Introduction 20
2.2 Rappel sur les réseaux de neurones artificiels 20
2.2.1 Définition et principe d'utilisation 20

2.2.2 Historique des RNA 21
2.3 Apprentissage des réseaux de neurones artificiels 23
2.3.1 Règles d'apprentissage 23
2.3.2 État de l'art sur les méthodes d'apprentissage 24
2.3.2.1 Apprentissage d'un perceptron 24
2.3.2.2 Apprentissage des réseaux de neurones multi-couches . . . 26
2.4 Dimensionnement des réseaux de neurones artificiels 29
2.4.1 État de l'art sur les méthodes de dimensionnement 29
2.4.1.1 Méthodes de dimensionnement statique 29
2.4.1.2 Méthodes de dimensionnement dynamique 31
2.5 Apprentissage et dimensionnement du RNA par ELM : application sur le réglage de la MPC 32
2.5.1 Apprentissage du RNA par l'ELM 32
2.5.2 Dimensionnement du RNA par l'ELM 34
2.6 Application de l'ELM au réglage de la MPC 37
2.6.1 Application sur 20 données dans la base d'apprentissage et 500 tests 37
2.6.2 40 données dans la base d'apprentissage et 500 tests 39
2.6.3 300 données dans la base d'apprentissage et 500 tests 41
2.7 Conclusion 46

Chapitre 3
Proposition d'une approche de réglage de la commande prédictive par RNA

3.1 Introduction 49
3.2 Formulation mathématique de la commande prédictive 49
3.2.1 Représentation d'état augmentée à temps discret 49
3.2.2 Formulation de la commande prédictive à base d'état en absence de contraintes 51
3.2.3 Formulation mathématique de la commande prédictive à base d'état en présence de contraintes 54
3.2.3.1 Contraintes sur les entrées de commande 54
3.2.3.2 Contraintes sur les incréments de commande 55
3.2.3.3 Contraintes sur les sorties 56
3.2.3.4 Formulation des contraintes 57
3.3 Stratégie de réglage des paramètres de la MPC basée sur des RNA 57
3.3.1 Structure du RNA 57
3.3.1.1 Les entrées du RNA 58

3.3.1.2 Les sorties du RNA 58
3.3.2 Construction de la base de données d'apprentissage du RNA par PSO 58
3.3.2.1 Principe de formulation mathématique du PSO 59
3.3.2.2 Formulation du problème de réglage de la MPC par PSO . 61
3.4 Application en simulation de l'approche proposée 62
3.4.0.1 Application numérique à un système SISO 64
3.4.0.2 Application numérique à un système SISO retardé 67
3.4.0.3 Application numérique sur un système MIMO retardé . . 70
3.5 Conclusion 74

Chapitre 4
Robustesse en performances de la commande prédictive

4.1 Introduction 77
4.2 État de l'art sur les approches de robustification de la commande prédictive 77
4.2.1 Approches basées sur la représentation polynomiale 78
4.2.2 Approches basées sur la représentation d'état 85
4.2.3 Synthèse comparative des approches de robustification de la commande prédictive 89
4.3 Influence des paramètres de réglage sur la robustesse de la commande prédictive 95
4.3.1 Comparaison entre la MPC réglée par RNA et la GPC non robustifiées 95
4.3.2 Comparaison entre la MPC réglée par RNA et la GPC avec robustification 97
4.3.3 Comparaison entre la MPC réglée par RNA avec et sans robustification 99
4.4 Conclusion 101

Chapitre 5
Application de la MPC réglée par RNA sur un drone QTW

5.1 Introduction 103
5.2 Drone convertible "QTW" 103
5.2.1 Structure et modes de vol 103
5.2.2 Modélisation dynamique des drones QTW 104
5.3 Modélisation T-S des systèmes non linéaires 106
5.3.1 Représentation d'état des modèles T-S 107
5.3.2 Obtention du modèle T-S 108
5.3.3 Stabilité des systèmes T-S 109
5.4 Modélisation T-S du drone QTW en vol vertical 111

5.4.1 Construction du modèle T-S du drone convertible en mode de vol vertical 111
5.4.2 Validation du modèle 113
5.5 Application de la MPC réglée par RNA au drone QTW 116
5.6 Conclusion 122

Conclusion générale et perspectives **123**

Bibliographie

Bibliographie **126**

Liste des figures

1.1 Bilan des applications de la commande prédictive linéaire jusqu'en 1998 [36] 8
1.2 Bilan des applications de la commande prédictive non linéaire entre 1998 et 2003 [36] 8
1.3 Principe de la commande prédictive [36] 9
1.4 Influence de l'horizon de commande [36] 10
1.5 Influence de l'horizon de prédiction [36] 11

2.1 Exemple de structure d'un RNA [36] 21
2.2 Connexion entre les neurones [36] 25
2.3 Carte unidimensionnelle 26
2.4 Carte bi-dimensionnelle "voisinage hexagonal" 27
2.5 Architecture RNA-ELM 33
2.6 Horizon de commande pour 20 données et 500 tests 38
2.7 Horizon de prédiction pour 20 données et 500 tests 38
2.8 Erreur quadratique moyenne pour 20 données 39
2.9 Nombre de neurones de la couche cachée en fonction du nombre d'itérations pour 20 données 40
2.10 Horizon de commande pour 40 données et 500 tests 40
2.11 Horizon de prédiction pour 40 données et 500 tests 41
2.12 Erreur quadratique moyenne pour 40 données 42
2.13 Nombre de neurones de la couche cachée en fonction du nombre d'itérations pour 40 données 42
2.14 Horizon de commande pour 300 données et 500 tests 43
2.15 Horizon de prédiction pour 300 données et 500 tests 44
2.16 Erreur quadratique moyenne pour 300 données 44
2.17 Nombre de neurones de la couche cachée en fonction du nombre d'itérations pour 300 données 45

3.1 Structure du RNA 58
3.2 Évolution de la position d'une particule dans l'espace 60

3.3 Stratégie de réglage de la commande prédictive par RNA 63
3.4 Sortie du moteur en fonction du temps 66
3.5 Signaux de commande en fonction du temps 66
3.6 Sortie du système retardé en fonction du temps 69
3.7 Signaux de commande en fonction du temps 69
3.8 Sortie y_1 en fonction du temps 72
3.9 Sortie y_2 en fonction du temps 73
3.10 Signal de commande u_1 en fonction du temps 73
3.11 Signal de commande u_2 en fonction du temps 74

4.1 Modèle CARIMA 78
4.2 Structure du régulateur polynomial équivalent 79
4.3 Régulateur polynomial équivalent 80
4.4 Correcteur initial avec la structure du bouclage standard 83
4.5 Correcteur GPC à deux degrés de liberté avec paramétrisation de Youla . 84
4.6 Organigramme proposé pour la robustification de la GPC 85
4.7 Structure de la MPC MIMO robustifiée face à des incertitudes non-structurées additives via le paramètre de Youla [152] 87
4.8 Organigramme proposé pour la robustification de la MPC 89
4.9 Réponses indicielles du système avec GPC primaire 91
4.10 Marges de robustesse B_r pour les différentes méthodes. 92
4.11 Réponse indicielle du système en absence de perturbations 92
4.12 Signaux de commande du système en absence de perturbation 93
4.13 Réponses indicielles avec correcteur robustifié en présence des perturbations. 94
4.14 Évolution des signaux de commande en présence de perturbations 94
4.15 Sortie du système en fonction du temps 96
4.16 Signaux de commande en fonction du temps 96
4.17 Sorties du système avec robustification 98
4.18 Signaux de commandes avec robustification 98
4.19 Sorties avec et sans robustification de la MPC en fonction du temps 100
4.20 Signaux de commande avec et sans robustification de la MPC en fonction du temps 100

5.1 Les modes de vols du drone QTW [37] 104
5.2 Configuration du drone QTW 105
5.3 Angle de roulis du modèle NL et du modèle TS 114
5.4 Erreur sur l'angle de roulis 114
5.5 Angle de tangage du modèle NL et du modèle TS 115

5.6 Erreur sur l'angle de tangage 115
5.7 Angle de lacet du modèle NL et du modèle TS 115
5.8 Erreur sur l'angle de lacet 116
5.9 Stratégie de commande proposée 117
5.10 Évolution des sorties d'attitudes et d'altitude en fonction du temps 118
5.11 Évolution des sorties angulaires et linéaire en fonction du temps 119
5.12 Signaux de commande en fonction du temps 120

Liste des tableaux

2.1 Comparaison des performances 45

3.1 Paramètres de la MPC utilisés en simulation pour le moteur électrique SISO 65
3.2 Comparaison des performances de la MPC 65
3.3 Paramètres de réglage de la MPC utilisés en simulation 68
3.4 Paramètres de la MPC utilisés en simulation pour un système SISO retardé 68
3.5 Comparaison des performances de la MPC 70
3.6 Paramètres de réglage de la MPC pour le système (3.53) 71
3.7 Performances de la sortie y_1 72
3.8 Performances de la sortie y_2 72

4.1 Paramètre Q de robustification. 91
4.2 Comparaison des performances 99
4.3 Comparaison des performances 101

5.1 Valeurs des paramètres des contrôleurs prédictifs pour chaque sous modèle selon l'approche proposée 116
5.2 Valeurs des paramètres des contrôleurs prédictifs pour chaque sous-modèle selon l'approche analytique (Turki 2018) [165] 117
5.3 Comparaison des performances relatives à l'angle de roulis 119
5.4 Comparaison des performances relatives à l'angle de tangage 120
5.5 Comparaison des performances relatives à l'angle de lacet 121

Publications de l'auteur

Conférences internationales avec actes et comité de lecture

— MOUMOUH Houssam, LANGLOIS Nicolas et HADDAD Madjid. "A Novel Tuning approach for MPC parameters based on Artificial Neural Network". *International Conference on Control and Automation*, Edinburgh, Scotland, Juillet 16-19, 2019.

— MOUMOUH Houssam, LANGLOIS Nicolas et HADDAD Madjid. "A Novel Tuning approach for parameters based on Artificial Neural Network : An application to FOPDT systems". *15th European Workshop on Advanced Control and Diagnosis, ACD* , Bologne, Italy, Novembre 21-23, 2019.

— MOUMOUH Houssam, LANGLOIS Nicolas et HADDAD Madjid. "Off-line Robustness Improvement of a Predictive Controller Using a Novel Tuning Approach Based on Artificial Neural Network". *16 th IEEE International Conference on Control Automation, ICCA* , Hokkaido, Japan, Octobre 9-11, 2020.

— MOUMOUH Houssam, LANGLOIS Nicolas et HADDAD Madjid. "Robustness of Model Predictive Control Using a Novel Tuning Approach Based on Artificial Neural Network". *8th Mediterranean Conference on Control and Automation* , Saint-Raphael, France, Septembre 16-18, 2020.

Communications Orales

— MOUMOUH Houssam, LANGLOIS Nicolas et HADDAD Madjid. "A Novel Tuning approach for MPC parameters based on Artificial Neural Network". *GT Commande Prédictive Non linéaire du GDR MACS*, ONERA, Châtillon, France, 04 Juin 2018.

— MOUMOUH Houssam, LANGLOIS Nicolas et HADDAD Madjid. "Comparative study between two Tuning MPC approaches : Application to a simulated QTW UAV". *GT Commande Prédictive Non linéaire du GDR MACS*, CentralSupelec, France, 08 Février 2019.

Poster

— MOUMOUH Houssam, LANGLOIS Nicolas et HADDAD Madjid. "Élaboration d'un contrôleur prédictif auto-adaptatif à base de réseaux de neurones artificiels". *Journée des Doctorants de l'IRSEEM (JDD)*, Rouen, France, 08 Mars 2019.

Glossaire

Acronymes et abréviations

A	ANN	Réseaux de neurones artificiels (Artificial Neural Network)
	ARMA	Auto Regressive Moving Average
C	CSE	Énergie du signal de commande (Control Signal Energy)
D	DMC	Dynamic Matrix Control
E	ELMR	Extrem Learning Machine
	EHAC	Extended Horizon Adaptive Control
F	FOPDT	Système du premier ordre avec retard (First Order Plus Dead Time)
G	GPC	Generalized Predictive Control
I	IAE	Integral of Absolute Error
L	LMI	Inégalité(s) matricielle(s) linéaire(s) (Linear Matrix(ces) Inequality(ies))
	LTI	Linéaire invariant dans le temps (Linear Time Invariant)
	LTV	Linéaire variant dans le temps (Linear Time Varying)
	LQR	Régulateur quadratique linéaire (Linear Quadratique Regulator)
	LQG	Linear Quadratic Gaussian
M	MIMO	Multi-entrées multi-sorties (Multi-Input Multi-Output)
	MPC	Commande prédictive à base de modèle (Model-based Predictive Control)
	MTE	Moyenne de l'erreur de poursuite (Mean Tracking Error)
	MPHD	Model Predictive Heuristic Control
	MAC	Model Algorithmic Control
	MUSMUR	MUlti Step Multivaraible Adaptive Control
	MOEA	Multi-Objective Evolutionary Algorithm
O	OV	Dépassement (OVershoot)
	OS-ELM	Online Sequential-Extrem Learning Machine
P	PCA	Analyse des composantes principales (Principal Component Analysis)
	PFC	Predictive Functional Control

	PSO	Particle Swarm Optimisation
Q	QTW	Quad Tilt Wing
	Quasi-LPV	Quasi linéaire à paramètres variables (quasi-Linear Parameter Varying)
R	RLS	Méthode des moindres carrés récursifs (Recursive least squares)
	RT	Temps de montée (Rise Time)
	RPN	Robust Performance Number
S	SISO	Mono-entrée mono-sortie (Single-Input Single-Output)
	SDI	Index de degrée de stabilité (Stability Degree Index)
	SVD	Décomposition en valeurs singulières (Decomposition of Singular Values)
	ST	Temps de réponse (Settling Time)
	SOPDT	Système du second ordre avec retard (Second Order Plus Dead Time)
T	T-S	Takagi-Sugeno
	TE	Tracking Error
U	UAV	Unmanned Aerial Vehicle
	UPC	Unified Predictive Control
V	VARU	VARiance du signal de commande (VARiance of control signal)

Symboles et notations

$x(.) \in \mathbb{R}^{n_1}$	Vecteur d'état
$u(.) \in \mathbb{R}^{n_{in}}$	Vecteur de commande
$y(.) \in \mathbb{R}^{n_{out}}$	Vecteur de sortie
$A \in \mathbb{R}^{n_{etat} \times n_{etat}}$	Matrice d'état
$B \in \mathbb{R}^{n_{etat} \times n_{in}}$	Matrice de commande
$C \in \mathbb{R}^{n_{out} \times n_{etat}}$	Matrice d'observation
$A_m \in \mathbb{R}^{n_1 \times n_1}$	Matrice d'état initiale
$B_m \in \mathbb{R}^{n_1 \times n_{in}}$	Matrice d'entrée initiale
$C_m \in \mathbb{R}^{n_{out} \times n_1}$	Matrice de sortie initiale
ΔU	Vecteur de commande incrémentale
Y_{des}	Vecteur des trajectoires de sortie désirées
$\bar{R}$	Matrice de pondération du vecteur ΔU
$\hat{Y}$	Vecteur des sorties prédictes
F	Matrice de pondération du vecteur x dans l'équation de prédiction
Φ	Matrice de pondération du vecteur ΔU dans l'équation de prédiction

J	Fonction coût de la MPC
H	Matrice Hessienne
M_{cons}	Matrice pondérant Δu dans l'inéquation des contraintes
η_{cons}	Vecteur multiplicateur de Lagrange
γ_{cons}	Vecteur supérieur ou égal au produit $M_{cons}\Delta u$
K_{mpc}	Vecteur de gain de commande par retour d'état
K_y	Vecteur de gain de rétroaction liée à la sortie
K_{cons}	Vecteur de gain de commande liée aux contraintes
A_{cl}	Matrice d'état en boucle fermée
N_c	Horizon de commande
N_p	Horizon de prédiction
λ	Facteur de pondération
υ^{max}	Valeur propre maximale de A_{cl}
υ^{min}	Valeur propre minimale de A_{cl}
$cond(H)$	Conditionnement de la matrice Hessienne
$\theta(.)$	Variable de prémisse

Introduction générale

Depuis plusieurs décennies, la commande automatique s'est imposée dans l'industrie. L'objectif derrière est de respecter le cahier des charges, d'augmenter la productivité, de réduire au mieux le temps de conception et d'améliorer les performances.
Toutefois, le choix d'une commande s'avère de plus en plus compliqué au regard du nombre important de contrôleurs développés et des différents outils qui facilitent leurs utilisations. Ce choix dépend forcement de la complexité du système à contrôler, des informations disponibles sur le procédé et des différentes perturbations et variations paramétrique affectant le système au cours du temps.
La synthèse d'une commande intelligente ou dite auto-adaptative permet de prendre en compte ces différentes contraintes et également d'élargir le champs d'application à des systèmes plus complexes. En effet, une commande est dite adaptative si ses paramètres sont mis à jour de façon automatique afin de prendre en compte les différentes variations du système en cours d'utilisation.
Dans le cadre de nos travaux de recherche, nous nous intéresserons particulièrement à la synthèse d'une loi de commande prédictive à base d'état MPC (Model Predictive Control). La MPC, comme son nom l'indique s'appuie sur une prédiction du comportement futur du système.
L'une des formes classiques de la MPC considère trois paramètres à régler et qui influencent significativement le comportement des systèmes en boucle fermée. Ces paramètres sont : l'horizon de commande, l'horizon de prédiction et le facteur de pondération. Il est important de rappeler que les valeurs attribuées à ces trois paramètres doivent être méticuleusement choisies pour obtenir les meilleures performances possibles en termes de stabilité, rapidité et précision.
Selon la littérature, trois grandes catégories de méthodes de réglage de la MPC peuvent être identifiées :

- la catégorie des approches dites "expérimentales",
- la catégorie des approches dites " heuristiques",
- la catégorie des approches dites " analytiques"

Chacune de ces approches présentant des avantages et des inconvénients qui seront

évoqués ultérieurement.
Notre contribution dans ce travail de recherche est de palier ces problèmes en proposant une approche de réglage de la MPC basée sur un Réseau de Neurones Artificiels RNA, afin qu'elle soit applicable à n'importe quelle classe de système SISO (Single Input-Single Output) où MIMO (Multi Input-Multi Output), avec ou sans contraintes en présence où non de perturbations.
L'approche de réglage proposée est complètement originale et aucune publication scientifique ne traite la problématique de réglage de la MPC en utilisant les RNA. Certes, il existe dans la littérature des approches qui lient la MPC et les RNA mais pas pour le réglage paramétrique. La majorité de ces approches utilisent les RNA pour la modélisation des systèmes qui sont par la suite contrôlés par la MPC.

C'est la raison pour laquelle, au travers le présent mémoire, nous présenterons une approche de réglage originale basée sur la théorie des RNA. Nous montrerons que cette approche est applicable à n'importe quelle classe des systèmes SISO (Single Input-Single Output) et MIMO (Multi Input-Multi Output), avec ou sans contraintes et en présence ou non de perturbations.
Classiquement, le recours aux RNA nécessite de se pencher sur une double problématique : la construction d'une base de données et le choix d'un algorithme d'apprentissage, c'est pourquoi seront justifiés le choix de l'optimisation par essaim de particules PSO (Paticle Swarm Optimisation) pour le premier et de l'algorithme OS-ELM (Online Sequential-Extrem Learning Machine) pour le second.

Les performances obtenues par l'approche neuronale seront comparées avec plusieurs méthodes de réglage afin de prouver sa pertinence.
Les résultats obtenus à l'issue de ces trois années de thèse seront présentés dans ce mémoire qui est composé de cinq chapitres :

— **Chapitre 1 : État de l'art sur la commande prédictive**
  Dans ce premier chapitre, un aperçu général sur la commande prédictive, son historique et ses différents domaines d'application seront présentés. Nous dresserons ensuite un état de l'art détaillé sur les différentes approches de réglage de la commande prédictive en montrant à chaque fois les avantages et les limites de chaque stratégie de réglage et finalement montrer l'intérêt de l'approche proposée.
— **Chapitre 2 : Apprentissage et dimensionnement des réseaux de neurones artificiels**
  Le deuxième chapitre sera consacré à l'outil principal dans l'approche proposée : les RNA.
  Dans un premier temps, un rappel synthétique sur les RNA sera effectué en

présentant leurs principes de fonctionnement et un historique de leurs utilisations. Dans un deuxième temps, un état de l'art sur les méthodes d'apprentissage et de dimensionnement sera effectué. Finalement, dans un troisième temps nous proposerons une approche qui permet à la fois d'établir la structure optimale et de faire l'apprentissage en lignes du RNA. La pertinence de cette approche sera ensuite prouvée au travers plusieurs applications.

— **Chapitre 3 : Proposition d'une approche de réglage de la MPC basée sur des RNA**

Le troisième chapitre constitue le coeur de notre travail car c'est dans celui-ci que sera présentée l'approche de réglage proposée. Dans un premier temps, la loi de commande prédictive à base d'état sera formulée. Dans un deuxième temps, nous présenterons la stratégie de réglage proposée basée sur un RNA en définissant les entrées et les sorties de ce dernier. Nous présenterons également les différents critères de performances qui seront considérés pour effectuer une étude comparative.

Ce chapitre sera ensuite clôturé par l'application de l'approche sur différents systèmes (SISO, MIMO avec et sans retard).

— **Chapitre 4 : Robustesse en performances de la commande prédictive**

Ce quatrième chapitre sera dédié à l'étude de la robustesse en performances de la commande prédictive. Dans un premier temps un état de l'art sur les différentes approches de robustification de la MPC sera présenté. Ensuite, dans un deuxième temps l'influence du réglage paramétrique de la MPC sur la robustesse en performances sera mis en avant. Une application de la MPC réglée par RNA sur des systèmes SISO sera présentée afin de montrer l'influence de cet outil sur la robustification.

— **Chapitre 5 : Application de la MPC réglée par RNA sur un drone QTW**

Dans ce dernier chapitre, l'objectif est d'étendre l'approche proposée afin qu'elle soit applicable sur les systèmes MIMO non linéaires.

L'approche par RNA sera donc appliquée sur un système de drone convertible QTW. Dans un premier temps, nous présenterons ce type de drone, leur modes de vol et leur modélisation dynamique. Dans un deuxième temps, nous établirons le modèle "Takagi-Suggeno" de ce modèle dynamique. Dans un troisième et dernier temps, une application de la MPC réglée par l'utilisation d'un RNA sur un drone QTW sera présentée et comparée avec une autre approche de la littérature.

Chapitre 1

État de l'art sur la commande prédictive

Sommaire

1.1 Introduction 5

1.2 Notions de base sur la commande prédictive 5

1.2.1 Historique 5

1.2.2 Domaines d'application 7

1.2.3 Principe de fonctionnement de la MPC 8

1.2.4 Choix des paramètres de réglage de la MPC 9

1.3 État de l'art sur les approches de réglage de la commande prédictive 11

1.3.1 Les approches analytiques 12

1.3.2 Les approches expérimentales 13

1.3.3 Les approches heuristiques 14

1.3.4 Les approches méta-heuristiques 18

1.4 Conclusion 18

1.1 Introduction

L'objectif principal derrière le développement des lois de commande est d'améliorer significativement le comportement des systèmes industriels en garantissant la fiabilité et en satisfaisant les objectifs définis par un cahier de charge donné.

La commande prédictive est l'une des techniques les plus utilisées dans l'industrie car elle s'appuie sur la prédiction du comportement futur en ayant suffisamment de données sur le modèle du système à contrôler. D'où son appellation en anglais Model Predictive Control (MPC).

La MPC appartient à un domaine varié qui englobe plusieurs types de commande, telle que la commande optimale, la commande multivariable, et les commandes avec ou sans contraintes. Le choix de la commande prédictive repose essentiellement sur des avantages majeurs qui la différencient des autres commandes :

- elle peut être employée pour contrôler une grande variété de processus, du système relativement simple aux systèmes multivariables non linéaires.
- elle permet l'optimisation des signaux de commande de façon à prendre en compte les contraintes.
- elle assure le rejet de perturbations et la prise en compte des variations paramétriques affectant le système.
- elle possède la capacité de la compensation du retard

Ce premier chapitre porte sur la présentation de la commande prédictive. Nous commencerons par donner un aperçu général de l'historique de la MPC et de ses domaines d'application.

Ensuite, nous détaillerons les approches de réglage de la commande prédictive en présentant les avantages et les inconvénients de chaque type d'approche afin de montrer l'intérêt de la méthode de réglage que nous proposerons ensuite.

1.2 Notions de base sur la commande prédictive

Depuis les années 70, les industriels manifestent de l'intérêt pour la MPC [41], [133], [6], [134] et [137].

1.2.1 Historique

Depuis 1978, les premières applications de la commande prédictive ont été proposées et synthétisées pour le contrôle des systèmes multivariables.

(Richalet et al.) ont conçu une approche heuristique de la commande prédictive basée sur le modèle en anglais "Model Predictive Heuristic Control" (MPHC) connu par la suite

sous le nom "Model Algorithm Control" (MAC). Cette approche consistait à imposer des trajectoires de référence à suivre pour les sorties tout en minimisant la variance de l'erreur [136].

Ensuite dans les années 80, (Culter et al.) ont réalisé une commande prédictive dite "Dynamic" car basée sur la modélisation dynamique du processus "Dynamic Matrix Control" (DMC). Le signal de commande de ce contrôleur est calculé par la méthode des moindres carrés, dans l'objectif de minimiser l'erreur de poursuite sans prise en compte des contraintes [46]. L'optimisation est répétée à chaque période d'échantillonage en utilisant les données mesurées et fournies par le système. Le but était d'appliquer cet algorithme sur des systèmes linéaires multivariables, en évaluant les réponses indicielle et impulsionnelle du procédé afin d'estimer l'action de la commande sur la sortie.

Ces succès ont poussé des chercheurs à s'intéresser à la commande prédictive adaptative : (Keyser et al,) ont proposé une approche dite adaptative sous le nom "Extended Prediction Self Adaptative Control" (EPSAC). Dans cet algorithme, ils ont introduit un signal de commande constant qui est appliqué au système non linéaire tout au long de l'horizon de prédiction et ainsi qu'un prédicteur sous-optimal pour la minimisation d'une fonction coût [48] .

Toujours sur la commande adaptative, (Ydstie et al,) ont proposé une stratégie pour les systèmes monovariables sous le nom de commande prédictive à horizon étendu "Extended Horizon Adaptive Control" (EHAC) [175]. Cette approche repose sur l'utilisation des modèles Entrées/Sorties pour calculer, à chaque itération la séquence de commande optimale pour maintenir une erreur nulle au-delà du retard pur du système.

En 1987, (Clarke et al,) ont présenté une commande prédictive généralisée "Generalized Predictive Control" (GPC) basée sur un modèle CARIMA en introduisant l'horizon de prédiction sur le signal de commande [42]. Ce signal est ensuite appliqué aux systèmes à nom minimum de phase avec retard et instables en boucle ouverte, en agissant conformément au principe de l'horizon fuyant pour une utilisation simple sur des systèmes "Single-Input Single-Output" (SISO) industriels.

La facilité d'implémentation de la commande prédictive a permis son application à plusieurs secteurs industriels tels que :la chimie, l'aérospatiale et l'industrie du papier [65] et [134].

Pour une application simple sur les systèmes SISO industriels, (Stoica et al,) ont proposé un algorithme "Predictive Functional Control" (PFC) qui permet de régler les paramètres de la commande prédictive en agissant sur les performances considérées dans l'algorithme [153].

Ensuite, la commande prédictive à base de modèle (Model Predictive Control) utilisant la représentation d'état a été proposée par (Morari et al,) afin de faciliter l'analyse de

la stabilité, la rapidité et la robustesse [122]. Cette dernière représentation a favorisé le développement de la théorie de la MPC en présence de perturbations et de bruits de mesures.
La plupart des approche citées ci-dessus font appel à la commande prédictive pour le contrôle des systèmes linéaires. C'est pour cela que par la suite plusieurs chercheurs ont orienté leurs travaux sur les systèmes non linéaires. Nous citons quelques travaux sur les systèmes non linéaires.

(Chen et al,) ont traité des problèmes d'instabilité liés aux systèmes non linéaires au moyen de la commande prédictive avec une formulation sous contraintes [40].
La mise en oeuvre de la MPC non linéaire a été ensuite présentée dans plusieurs travaux [113], [71], [7] et [35].
Le rappel que nous avons effectué a permis d'avoir une idée sur l'évolution de la commande prédictive dans un ordre chronologique, de la commande classique au développement des commandes non linéaire sous contraintes et qui permettent le rejet des perturbations.

1.2.2 Domaines d'application

Dès les années 70, l'utilisation de la commande prédictive s'est répondue dans différents domaines d'activités industrielles, aussi bien pour les performances qu'elle permet d'atteindre que pour sa facilité de mise en oeuvre. Son champ d'application s'étend désormais à tous les domaines industriels. Dans un premier temps, ce sont les industriels de la pétrochimie et la métallurgie [136] qui ont manifesté de l'intérêt pour la commande prédictive. Ensuite, elle a été utilisée avec succès dans différentes applications et pour plusieurs champs d'activités. En 2003, la commande prédictive faisait déjà l'objet de 4000 applications [134]. Les Tableaux 1.1 et 1.2 représentent les bilans des applications industrielles de la commande prédictive entre 1978 et 2003.

La prise en compte des contraintes et le rejet de perturbations ont ainsi mis en évidence son aptitude à être utilisée dans des domaines variés sur des systèmes multivariables, notamment dans le domaine de l'automobile, dans le contrôle de la transmission, dans la régulation des émissions d'un moteur Diesel [2], [23], ou encore dans le contrôle de la direction d'un véhicule autonome [60]. Le domaine médical a aussi éprouvé de l'intérêt pour la commande prédictive. Dans [170] par exemple, cette loi de commande a été employée afin de contrôler les doses d'insuline à injecter à un patient diabétique.

Son aptitude à être appliquée aux systèmes avec un grand nombre de variables la rend intéressante dans le domaine de la robotique surtout face à des systèmes complexes. Nous citons à titre d'exemple l'utilisation de la commande prédictive dans le contrôle d'un robot humanoïde [50]et du robot HRP-2 [100]. Dans le domaine aérospatial, la MPC a été aussi

Area	Aspen Technology	Honeywell Hi-Spec	Adersa	Invensys	Total	%
Raffinage	1200	480	280	25	1985	48,5
Pétrochimie	450	80	-	20	550	13,4
Chimie	100	20	3	21	144	3,5
Papier	18	50	-	-	68	1,7
Air et Gaz	-	10	-	-	10	0,2
Métallurgie	8	6	7	16	37	0,9
Alimentaire	-	-	41	10	51	1,3
Polymère	17	-	-	-	17	0,4
Cuisson	-	-	42	3	45	1,1
Aérospatial	-	-	13	-	13	0,3
Automobile	-	-	7	-	7	0,2
Autre	40	50	1045	30	1165	28,5
Total	1833	696	1438	125	4092	100

Figure 1.1 – Bilan des applications de la commande prédictive linéaire jusqu'en 1998 [36]

Area	Adersa	Aspen Technology	Continental Controls	DOT Products	Pavillon Technologies	Total	%
Chimie	2		15		5	22	23,7
Polymère		1		5	15	21	22,6
Air et gaz			18			18	19,3
Raffinage					13	13	14
Alimentaire					9	9	9,7
Autres	1	5	3			9	9,7
Papier					1	1	1
Total	3	6	36	5	43	93	100

Figure 1.2 – Bilan des applications de la commande prédictive non linéaire entre 1998 et 2003 [36]

appliquée aux vannes électromécaniques [20], aux engins spatiaux [92], [131], [95] et [72] et aux satellites [131], sans oublier son intégration dans le contrôle des systèmes énergétiques comme des centrales hydro-électriques [51], des éoliennes [120], des échangeurs thermiques [177], des réseaux de distribution d'eau [150] et des centrales de traitement des eaux [76].

Par la suite, nous présenterons en détails le principe de fonctionnement de la commande prédictive et nous aborderons ainsi la problématique du réglage paramétrique de la MPC.

1.2.3 Principe de fonctionnement de la MPC

Le principe de la commande prédictive est de s'appuyer sur la prédiction des états futurs d'un système pour optimiser le signal de commande à appliquer à l'instant présent sur la base de la minimisation d'une fonction coût généralement notée J

A chaque instant, la minimisation de la fonction coût permet d'obtenir sur un horizon de prédiction une séquence de commande optimale. La MPC fait aussi appel au principe de

l'horizon fuyant qui permet, à son tour, d'extraire le premier élément de cette séquence de commande et de l'appliquer précisément au système à contrôler. Les autres éléments du signal de commande peuvent être négligés car à la période d'échantillonage qui suit, les séquences est décalée.
L'objectif est de pouvoir maintenir un écart minimal entre la sortie du système et la référence, pour mettre en évidence le caractère anticipatif de la MPC. La Figure 1.3 illustre le principe de fonctionnement de la commande prédictive :

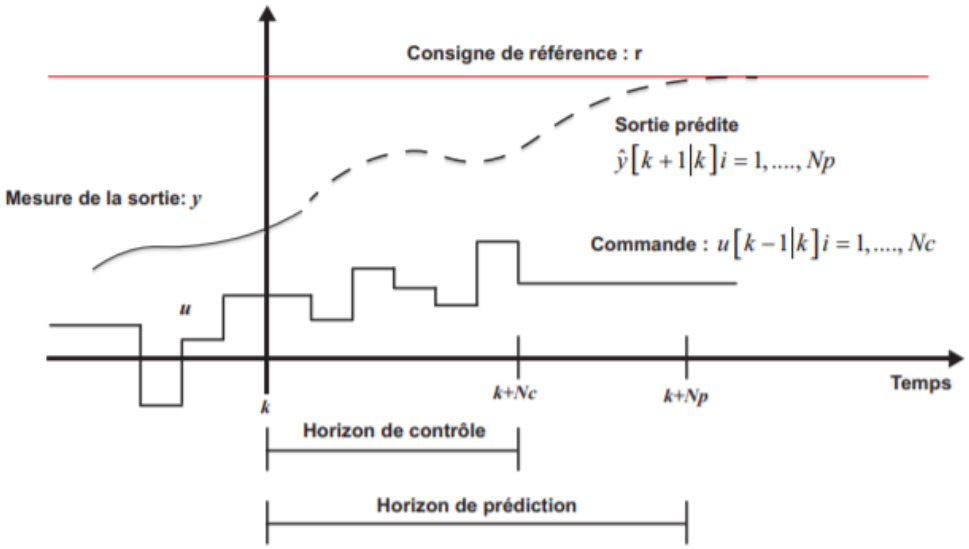


Figure 1.3 – Principe de la commande prédictive [36]

Le calcul de la séquence de commande optimale u à appliquer au système se fait à chaque instant t_k jusqu'à l'instant t_{k+N_c} où N_c est l'horizon de commande de la MPC. Pour cela, il se base sur un modèle du système à contrôler permettant de prédire la sortie y jusqu'à l'instant t_{k+N_p} et sur une trajectoire de référence à poursuivre (r présentée en rouge dans la Figure 1.3).

1.2.4 Choix des paramètres de réglage de la MPC

Selon la littérature, la synthèse d'une loi de commande prédictive requiert communément la détermination de trois paramètres qui sont :

1. L'horizon de commande N_c.
2. L'horizon de prédiction N_p.
3. Le facteur de pondération λ.

Ces paramètres influencent d'une manière significative le comportement du système à commander en termes de stabilité, rapidité et précision. Leur choix doit être fait avec

grande attention. Le réglage de ces paramètres dépend de plusieurs élements tels que : la période d'échantillonnage, la dynamique du système, le retard, etc. selon la littérature, plusieurs travaux ont été menés sur la problématique du réglage de ces paramètres.

1. **Choix de l'horizon de commande** Le choix de N_c égal à l'unité est reconnu comme étant suffisant dans la plupart des cas [24]. Cependant, une valeur plus élevée permet d'atteindre des objectifs plus complexes lorsque les systèmes à contrôler présentent un nombre élevé de degrés de liberté [42]. Or, plus la valeur de l'horizon de commande est élevée, plus le temps du calcul du signal de commande est important. Cela s'explique par le fait que N_c fixe la dimension des matrices à inverser dans le calcul de la séquence de commande optimale. Un compromis doit être alors trouvé [125].
   La Figure 1.4 montre l'influence de l'horizon de commande pour un horizon de prédiction constant et un facteur de pondération également constant :

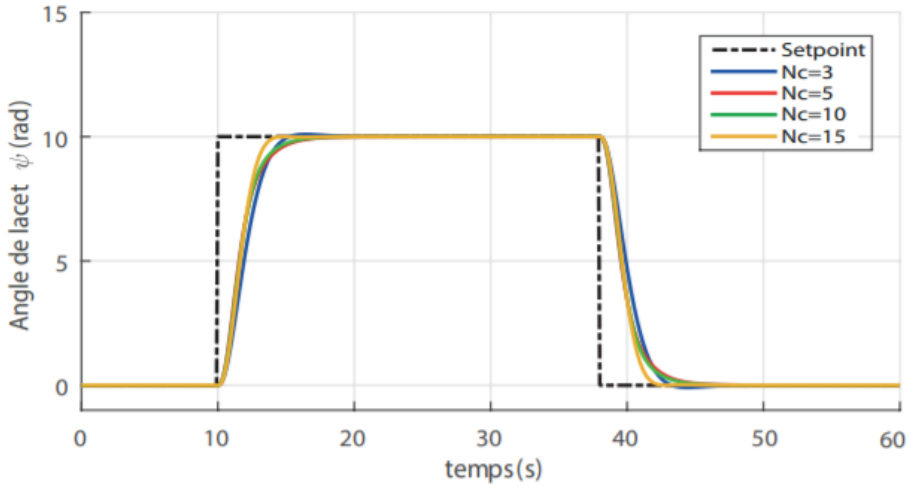


Figure 1.4 – Influence de l'horizon de commande [36]

2. **Choix de l'horizon de prédiction** Comme mentionné précédemment, la commande prédictive fait appel au principe de l'horizon fuyant. À chaque itération, l'horizon de prédiction N_p permet d'avoir la visibilité sur les futurs états du système jusqu'à un temps fini, ce qui donne à l'algorithme de résolution une meilleure qualité des informations relatives à la dynamique d'évolution du système. Cependant, plus N_p est grand, plus la qualité de la prédiction est faible. En effet, des erreurs peuvent apparaître au cours du temps entre le modèle de prédiction et le système réel. Dans la littérature, il n'existe pas de méthodes génériques permettant de déterminer la valeur de l'horizon de prédiction de manière optimale. Mais, un compromis peut toutefois être trouvé entre une valeur minimale, qui offre peu d'informations sur l'évolution du système mais assurant une meilleure qualité de prédiction, et

une valeur maximale, qui fournit une plus large quantité d'informations mais des incertitudes plus importantes par rapport à l'état futur réel du système [125] et [36]. La Figure 2.2 montre l'influence de l'horizon de prédiction pour un horizon de commande constant et un facteur de pondération également constant :

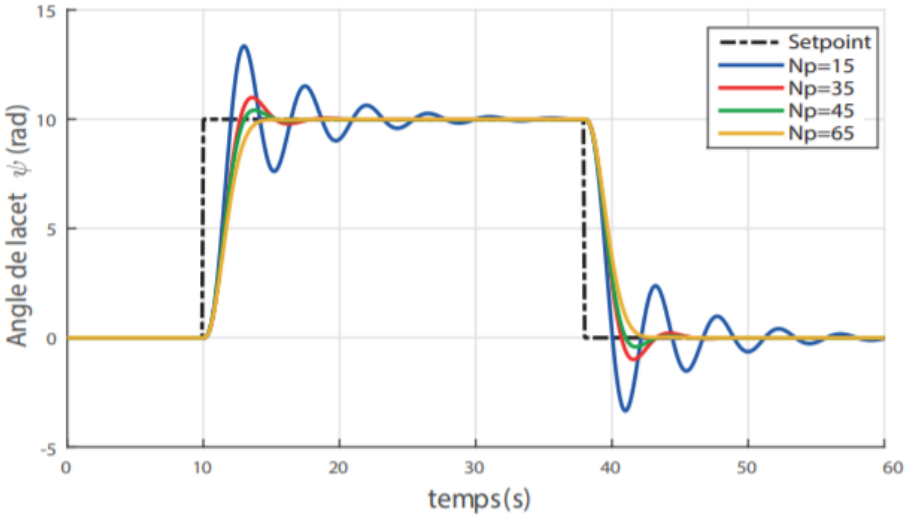


Figure 1.5 – Influence de l'horizon de prédiction [36]

3. **Choix du facteur de pondération** Tout comme son nom l'indique, le facteur de pondération λ joue un rôle primordial de dosage. Il permet, dans la fonction coût d'équilibrer ou pas l'erreur de poursuite et l'impact de la commande. Une augmentation excessive de λ implique une demande exigeante de la commande, et par conséquent, la dynamique du système est ralentie. Par contre, une diminution de λ entraîne une commande énergique qui rend la réponse du système de plus en plus rapide. Il faut noter que le réglage de ce paramètre est assez délicat. En effet, tout comme l'horizon de commande, il influence sur la stabilité du système en boucle fermée [125] et [24].

Après avoir expliqué le rôle principal que joue le choix des paramètres de réglages dans la synthèse la MPC, nous allons mener un état de l'art sur les approches les plus significatives abordant la problématique du réglage des paramètres la commande prédictive.

1.3 État de l'art sur les approches de réglage de la commande prédictive

Le réglage des paramètres d'un contrôleur prédictif constitue l'élément central pour répondre aux exigences croissantes en terme de performances (stabilité, rapidité et

précision) des systèmes industriels complexes. Ce besoin croissant de garantir la robustesse en performances des procédés industriels a conduit des chercheurs à développer plusieurs approches d'ajustement des paramètres de la MPC [135], [46], [42], [103], [112], [148], [133], [151], [174], [67] et [73].
Dans la littérature, on a pu distinguer trois grandes catégories d'approches de réglage de la MPC :

1. Les approches analytiques,
2. Les approches expérimentales,
3. Les approches heuristiques.

Le paragraphe qui suit présente un état de l'art détaillé sur les différents types d'approches.

1.3.1 Les approches analytiques

La première catégorie d'approches regroupe les méthodes d'ajustement des paramètres de la MPC à caractère analytique (plus ou moins numériques). Ces méthodes présentent l'avantage de fournir les valeurs optimales des paramètres de réglage de la MPC pour des systèmes multivariables. Par contre, elles demandent un temps de calcul élevé, et ne permettent pas de traiter la problématique de l'adaptabilité.

Les travaux publiés par (Shridhar et al,) visent à optimiser les paramètres de la DMC selon une approche analytique exacte dédiée aux systèmes SISO et MIMO s'écrivant sous la forme d'un FOPDT. Leur méthode garantit la stabilité du procédé en boucle ouverte en absence de contraintes [148], [151] et [147].
(Gholaminejad et al.) et (Bagueri et al.) ont proposé une méthode qui permet de présenter tout type de processus physique par un modèle FOPDT (ce qui est parfois inadéquat). Ils mettent à jour les paramètres de la MPC en s'appuyant sur la méthode des moindres carrés récursifs (Recursive Least Square ou RLS). Dans le cas où le gain obtenu par le RLS n'est pas réalisable, la résolution des Inégalités Linéaires Matricielles (Linear Matrix(ces) Inequality(ies) ou LMI) est faite pour générer le gain désiré pour des systèmes monovariables [70] et [18].

Les travaux publiés par (Turki et al,) permettent de calculer les paramètres optimaux de la MPC applicables aux systèmes SISO, MIMO avec et sans contraintes [165]. Cette approche se base sur plusieurs hypothèses d'optimisation [167] ce qui constitue une problématique pour son application en temps réel.

En synthèse sur les méthodes de réglage analytique, ces dernières ont l'avantage de fournir les valeurs optimales des paramètres. Seuls quelques articles ont été publiés en raison de la complexité du problème. La plupart des modèles fournis se base sur des

hypothèses. Cette approximation dans la modélisation des systèmes physiques peut être restrictive dans le cas de certains systèmes industriels.

1.3.2 Les approches expérimentales

La deuxième catégorie d'approches regroupe les approches expérimentales, appelées aussi "empiriques", où les paramètres de la MPC sont déterminés en fonction du retour d'expérience du concepteur du système [33].

Ces approches font parties des approches les plus simples et intuitives pour l'ajustement des paramètres de réglage considérés (N_c, N_p, λ) mais elles nécessitent comme leur nom l'indique l'accès à un banc expérimental.

(Bursali et al.) ont proposé une approche qui permet la détermination de la valeur du facteur de pondération optimal [34]. Cette approche nécessite un nombre d'expériences important avec la présence d'un superviseur qui doit acquérir une bonne connaissance du système étudié. La méthode proposée est basée sur la conception expérimentale statistique et l'optimisation Box-Wilson. Le facteur de pondération et le temps d'échantillonnage sont réglés de façon à obtenir les meilleures performances de contrôle. Cette approche a été ensuite exploitée par (Bursali et al.) qui l'a appliquée pour le contrôle d'un bioréacteur par lots [34] .

(Chang et al.) ont proposé une approche de réglage expérimentale de la MPC appliquée à des convertisseurs numériques de puissances [38]. La variation d'inductance et la dérivé de la tension d'entrée constituent la problématique principale qui a poussé l'auteur à proposer ce type d'approche basée sur les courants échantillonnés entre deux sorties de modulation de largeur d'impulsion. Cette problématique peut être résolue par la technique d'adaptation paramétriques en ligne qu'ils ont proposée.

(Valencia et al,) ont proposé une nouvelle approche de réglage des paramètres de la MPC [169]. L'originalité de cette approche réside dans la modélisation du système à contrôler. Cette dernière repose sur des informations du système relativement brutes. Des tests expérimentaux en laboratoire ont été réalisés pour deux applications afin de prouver l'efficacité de la solution qui combine les algorithmes et le matériel utilisé.

Par ailleurs, (Shadmand et al) ont proposé dans la même catégorie d'approches une technique de compensation de puissance réactive [143]. Ici, le réglage porte uniquement sur le facteur de pondération qui est obtenu grâce à l'erreur absolue de l'objectif d'optimisation et aux contraintes correspondantes.

En conclusion, les approches à base de données expérimentales pour le réglage de la MPC sont celles les plus utilisées dans l'industrie pour la détermination de la combinaison des trois paramètres qui assure la fiabilité et l'obtention des performances souhaitées. Cependant, il s'agit d'approches de réglage approximatif.

1.3.3 Les approches heuristiques

La troisième catégorie d'approches regroupe les méthodes "heuristiques". Ces approches visent à trouver une approximation des valeurs des paramètres de la MPC pour atteindre les objectifs fixés dans le cahier des charges.
Selon la littérature, il existe plusieurs type d'approches heuristiques :

Les approches basées sur les algorithmes génétiques : par exemple (Ben Aicha et al,) ont proposé une méthode basée sur l'optimisation multi-objectif et faisant appel aux algorithmes génétiques [21]. L'optimisation des paramètres de réglage est fondée sur un compromis entre les performances en boucle fermée et le modèle obtenu. Ensuite, (Haber et al.) et (Naeem et al,) ont proposé une approche basée sur la reformulation de la fonction coût en intégrant les différents critères de performance (le temps d'établissement du régime stationnaire, le dépassement maximal et la longueur des horizons) [75],[124]. Cette approche a permis l'approximation de tout type de système physique par une fonction de transfert apériodique du second ordre.

Les approches basées sur la logique floue : il existe un nombre important d'articles dans la littérature visant à calculer les paramètres de réglage de la MPC en utilisant un superviseur flou [3], [4], [5], [115], [116] et [114]. L'approche proposée par (Ali et al,) permet d'obtenir le couple de paramètres (N_p, λ) optimal minimisant l'erreur entre la sortie réelle et la référence et d'atteindre les objectifs souhaiter en terme de performances [4]. L'algorithme proposé est constitué de deux phases, à savoir : la phase d'observation et la phase de déclenchement. La méthode vise à formuler le contrôleur prédictif par un mécanisme automatique, qui intègre la mise à jour des paramètres tout en maintenant la réponse en boucle fermée dans un domaine temporel prédéfini. Cette méthode a été inspirée par les travaux de (AlGhazzawi et al,) mais, sans utiliser la sensibilité de sortie pour régler les paramètres de la MPC [3].

Les approches basées sur les équations de sensibilité : (AlGhazzawi et al,) ont proposé une approche de réglage en ligne basée sur les équations de sensibilité pour des systèmes multivariables avec contraintes [3]. Cette approche est basée sur une approximation linéaire de la sortie prédite en boucle fermée. Le facteur de pondération est réglé de façon à obtenir les meilleures performances. La mise à jour de λ se fait au moyen de la sensibilité de la réponse en boucle fermée. Les horizons de commande et de prédiction N_c et N_p sont alors définis à l'aide des directives d'optimisation générales dans la littérature. Cette stratégie vise à orienter la réponse de rétroaction dans un domaine temporel spécifique pour

satisfaire les spécifications souhaitées en utilisant un algorithme d'optimisation de moindres carrés.

(Sarhadi et al,) ont proposé une approche dans le domaine fréquentiel à partir d'une analyse de robustesse pour optimiser les paramètres de la GPC [141]. La fonction de sensibilité est utilisée pour présenter cette technique à la fois pour l'analyse de robustesse et le réglage paramétrique du contrôleur. Cette approche permet d'obtenir les performances fixées par le cahier des charges mais elle est lourde en terme de calculs.

Les approches de réglage basées sur l'analyse de la variance : Des chercheurs ont fait le choix d'utiliser la méthode d'analyse de la variance en anglais ANalysis Of VAriance (ANOVA), en se basant sur la regression non linéaire pour le calcul des valeurs optimales des paramètres de la MPC [91], [126], [14], [15], [57], [17] et [19].

(Bagueri et al,) ont proposé une fonction analytique dédiée aux systèmes retardés FOPDT [15] . Cette approche a été appliquée dans un essai au laboratoire sur un procédé réel "pH" non linéaire. Une autre mise en oeuvre réussie du calcul des paramètres de la MPC a été optimisée par le même auteur dans qui a proposé une méthode basée sur la performance nominale en boucle fermée et qu'il a appliquée sur un procédé FOPDT [19]. Cette approche est basée sur le suivi de la trajectoire et l'analyse de la régression non linéaire. (Ebrahimi et al,) ont prouvé l'efficacité de l'ANOVA sur plusieurs cas de systèmes SISO et MIMO [57]. D'autres travaux continuent à utiliser cette approche [18], toujours avec des systèmes FOPDT où les contraintes sont inactives afin de stabiliser ces systèmes en boucle fermée et en obtenant les paramètres de réglage optimaux [16]. La limite de ces approches réside dans le fait qu'elles sont basées sur l'approximation physique de procédé réel par un modèle FOPDT, ce qui n'est pas toujours réalisable.

Les approches par correspondance du contrôleur : (Cairano et al,) ont proposé une méthode de réglage paramétrique de la commande prédictive par la méthode de correspondance des contrôleurs en définissant les matrices de poids, et les matrices d'états de telle manière que le contrôleur prédictif se comporte comme un contrôleur linéaire quadratique [49]. Les procédures suivies pour les synthèses des contrôleurs sont basées sur l'ajustement de la fonction de coût de la MPC en résolvant un problème d'optimisation convexe.

(Tran et al,) ont proposé une méthode qui vise à faire correspondre la fonction de transfert d'un contrôleur arbitraire à temps discret et de la MPC pour obtenir le même comportement [161]. Les paramètres MPC se calculent également en résolvant un problème d'optimisation par LMI convexe sous contraintes.

Les approches heuristiques par optimisation : (Tran et al,) ont proposé une approche pour la recherche d'extremum en anglais Extremum Seeking (SE) [162]. Cette approche a pour objectif de calculer par auto-optimisation les matrices de pondération de la fonction coût à partir de la bande passante imposée en boucle fermée. Ils ont suggéré une méthode pour obtenir la variance de sortie minimale. Leur idée principale réside dans la réalisation d'un équilibre optimal entre la robustesse en performances et la performance nominale du système bouclé. La bande passante optimale désirée a été calculée sur la base d'un algorithme ES qui se présente sous une forme d'optimisation adaptative.

Un algorithme évolutionnaire multi-objectif en anglais Multi-Objective Evolutionary Algorithm (MOEA), a été élaboré par (Gutizrrez et al,) [74]. Cet algorithme consiste à proposer une sélection automatique des paramètres de la MPC afin d'obtenir un bon compromis entre la fiabilité, la performance et le coût. Ce type d'algorithme s'est avéré être un outil efficace pour ce type de système de commande [62], [63] et [102]. (Han et al,) ont eu recours à une optimisation à base des bornes min-max pour trouver les valeurs appropriées des paramètres de la MPC [77].

Le réglage de la MPC par la résolution des différents problèmes d'optimisation a été largement étudié dans la littérature [27]. Par exemple, certains travaux publiés ont évoqué la détermination des paramètres du contrôleur prédictif en résolvant une optimisation par essaim (en anglais Swarm) basée sur les distributions de Cauchy et sur le modèle ARMA (en anglais Auto Regressive Moving Average) [53] et [154]. D'autres ont eu recours à une optimisation multi-objectif récursive [108] ou à l'identification récursive du système [121]. L'avantage de cette structure est qu'elle est moins exigeante en termes de calcul que la GPC classique. Les valeurs appropriées de la MPC ont été sélectionnées en se basant sur la stabilité du système en boucle fermée. Une optimisation convexe a été étudiée par [144] et [145] où la stratégie de réglage est basée sur la spécification de la performance en boucle fermée dans le domaine fréquentiel. Deux problèmes d'optimisation ont été résolus par leur méthode présentant une programmation semi-définie (en anglais SemiDefinite Programming ou SDP). Le premier comportait une optimisation de la réponse en fréquence pour déterminer les degrés de liberté, et le deuxième a porté sur une optimisation des matrices de pondération. L'efficacité de cette méthode a été testée avec des systèmes SISO et MIMO lorsque les contraintes sont inactives.

Les approches basées sur les méthodes de filtrage : (Schawrz et al,) ont proposé une approche de réglage des paramètres de la MPC qui est basée essentiellement sur la méthode d'identification des paramètres du modèle en boucle fermée [142]. Le

principal atout de cette approche est sa capacité à pouvoir exploiter les paramètres du modèle physique pour sélectionner les paramètres optimaux du contrôleur. Dans le même sens et pour réduire significativement l'impact du bruit, ils ont intégré également un filtre de "Rhinehart" dans la conception du système.

Les approches heuristiques visant l'optimisation des performances et l'analyse de stabilité : La stabilité est l'une des principales exigences pour tout système de commande. Dans notre contexte, les spécifications pour la stabilité du système dépendent fortement des paramètres de la MPC.

Dans [69], (Gerksic et al,) ont proposé une approche de réglage par un modèle multi-paramétrique. Cette approche est utile lorsque les perturbations sont très importantes. Leur méthode de réglage est basée sur l'analyse linéaire locale du système en boucle fermée qui est un outil extrêmement précieux pour l'analyse de la performance de l'action de rétroaction. Cet algorithme est particulièrement important dans les applications de industriel pour le suivi des trajectoires. Ensuite, le cas problème du réglage de la MPC à deux degrés de liberté avec des contraintes actives a été étudié dans [78]. Les paramètres de la MPC sont alors déterminés à partir des performances souhaitées, à savoir la stabilité, le dépassement du cas le plus défavorable, les oscillations en sortie de processus et le temps de stabilisation du cas le plus défavorable.

Quant aux effets des paramètres de la MPC sur le comportement en boucle fermée, ils ont été publiés dans [66] . En particulier, (Garriga et al,) se sont concentrés sur la localisation des valeurs propres sur une large gamme. Pour cela, ils ont utilisé le "Jacobien" en boucle fermée connu pour son exigence en termes calculatoires [97]. Ils ont suggéré une sélection automatique des paramètres du contrôleur prédictif basée sur l'analyse des pôles stables en boucle fermée. La procédure de réglage est basée sur une optimisation multi-objectif et des courbes de compromis entre la performance moyenne et le nombre de degrés de liberté.

Les approches basées sur le RPN (Robuste Performance Number) : Le RPN représente un critère pour la mesure de la commandabilité d'un système. Le RPN a été présenté lors d'une étude effectuée par (Trierweiler et al,) pour des systèmes multivariables [163] et [164]. L'idée principale de la méthode de réglage proposée basée sur le RPN est d'appliquer correctement le principe de la récursivité sur le système et les matrices de pondération. Les avantages de l'utilisation du RPN résident dans sa capacité à prendre en compte le comportement du procédé en phase nominale et les performances désirées en boucle fermée, et aussi sa capacité à quantifier correctement la fréquence d'échantillonage en fonction de la directionalité du système. Par conséquent, la robustesse en performances du système est assurée.

Toutes ces méthodes ont un caractère heuristique (à base d'essais-erreurs) qui ne permet pas toujours d'identifier explicitement la zone de robustesse du système. La plupart d'entre elles sont basées sur des techniques statistiques exigeantes en termes de calculs.

1.3.4 Les approches méta-heuristiques

Le réglage de la MPC via la résolution des différents problèmes d'optimisation a été également étudié dans la littérature. (Benkhoud et al,) ont proposé une approche sur la détermination des paramètres du contrôleur prédictif en résolvant un problème par PSO (Particle Swarm Optimisation) [27].
(Coelho et al,) [53] et (Suzuki et al,) [154] se sont intéressés à l'optimisation par essaim basée sur les distributions de Cauchy et Gaussian et sur le modèle ARMA (Auto Regressive Moving Average). Ces approches permettent de faire une optimisation des trois paramètres de réglage de la MPC, mais l'une des limites de ces approches est de ne pas prendre en compte les perturbations extérieures et la variation paramétrique affectant le système au cours du temps.

1.4 Conclusion

Ce premier chapitre a permis de donner un aperçu général sur la commande prédictive. Dans un premier temps, un rappel sur l'historique et les différents domaines d'application de la MPC ont été présentés en insistant sur ses particularités qui en font une approche efficace et particulièrement répandue dans le milieu industriel. Dans un deuxième temps, nous avons présenté les principes fondamentaux du fonctionnement de la MPC en introduisant la fonction coût qui fait appel aux paramètres de réglage. Le choix de ces derniers doit être fait avec une grande attention afin de garantir la poursuite de trajectoire imposée au début ce qui a motivé précisément nos travaux.

Dans un troisième temps, un état de l'art sur les stratégies de réglage de la MPC a été mené. L'analyse de cet état de l'art nous a permis d'avoir une idée globale sur les approches de réglage actuelles et leurs limites. La méthode proposée dans le chapitre qui suit porte sur une approche totalement nouvelle et innovante. Les limites exposées plus hauts seront dépassées notamment en ayant recours aux réseaux de neurones artificiels et leurs capacités d'apprentissage. La problématique principale dans l'utilisation du RNA réside dans la synthèse de la structure optimale et la construction de la base de données d'apprentissage.

Chapitre 2

Apprentissage et dimensionnement des réseaux de neurones artificiels

Sommaire

2.1 Introduction 20

2.2 Rappel sur les réseaux de neurones artificiels 20

2.2.1 Définition et principe d'utilisation 20

2.2.2 Historique des RNA 21

2.3 Apprentissage des réseaux de neurones artificiels 23

2.3.1 Règles d'apprentissage 23

2.3.2 État de l'art sur les méthodes d'apprentissage 24

2.3.2.1 Apprentissage d'un perceptron 24

2.3.2.2 Apprentissage des réseaux de neurones multi-couches 26

2.4 Dimensionnement des réseaux de neurones artificiels 29

2.4.1 État de l'art sur les méthodes de dimensionnement 29

2.4.1.1 Méthodes de dimensionnement statique 29

2.4.1.2 Méthodes de dimensionnement dynamique 31

2.5 Apprentissage et dimensionnement du RNA par ELM : application sur le réglage de la MPC 32

2.5.1 Apprentissage du RNA par l'ELM 32

2.5.2 Dimensionnement du RNA par l'ELM 34

2.6 Application de l'ELM au réglage de la MPC 37

2.6.1 Application sur 20 données dans la base d'apprentissage et 500 tests 37

2.6.2 40 données dans la base d'apprentissage et 500 tests 39

2.6.3 300 données dans la base d'apprentissage et 500 tests 41

2.7 Conclusion 46

2.1 Introduction

Dans le chapitre précédent, nous avons dressé un état de l'art sur la commande prédictive et ses différentes stratégies de réglage identifiées dans la littérature. Le présent chapitre sera consacré à la présentation des réseaux de neurones artificiels qui constituent l'outil principal dans l'originalité de l'approche que nous proposons. Dans un premier temps, un rappel synthétique sur les RNA sera effectué. Ces derniers sont appropriés à la reconnaissance de formes, et utilisés pour la résolution de plusieurs problèmes en automatique : l'identification, le diagnostic, le contrôle-commande... . Leur historique et principe d'utilisation seront également détaillés. Dans un deuxième temps, une étude des différents algorithmes d'apprentissage statique et dynamique sera menée. Le RNA exige en effet une constitution appropriée de la base de données d'apprentissage et une synthèse adéquate de l'algorithme d'apprentissage. Dans un troisième et dernier temps, la sélection d'une architecture et formation du réseau sera présenté, conduisant à la présentation de la première contribution proposée, à savoir une approche de dimensionnement du RNA dynamique. Cette approche consiste à déterminer le nombre de neurones adéquats dans la couche cachée en fonction de la complexité du problème étudié. La structure adéquate du RNA permet d'assurer les performances et un temps de calcul convenable tout en évitant le sur-dimensionnement et également le sous-dimensionnement en fonction de la taille de la base d'apprentissage.

2.2 Rappel sur les réseaux de neurones artificiels

L'objectif de cette partie est de présenter les RNA, leur historique, et principe d'utilisation inspiré des neurones biologiques.

2.2.1 Définition et principe d'utilisation

Un RNA est une structure d'unités élémentaires connectées entre elles dont le fonctionnement est inspiré des neurones biologiques. Grâce à sa capacité de généralisation, chaque neurone calcule sa sortie sur la base des informations qu'il reçoit. Les réseaux de neurones sont utilisés pour résoudre certains problèmes comme la classification de données, la reconnaissance d'images,... etc. Pour apporter des résultats appropriés, un RNA exige la constitution d'une base de données, la sélection d'une architecture et la formation du réseau.

Un réseau de neurones est généralement constitué d'une couche d'entrée, d'une ou plusieurs couches cachées et d'une couche de sortie. Chaque couche peut comporter un ou plusieurs neurones.

La Figure 2.1 présente un exemple de RNA comportant une couche d'entrée, deux couches de cachées et une couche de sortie.

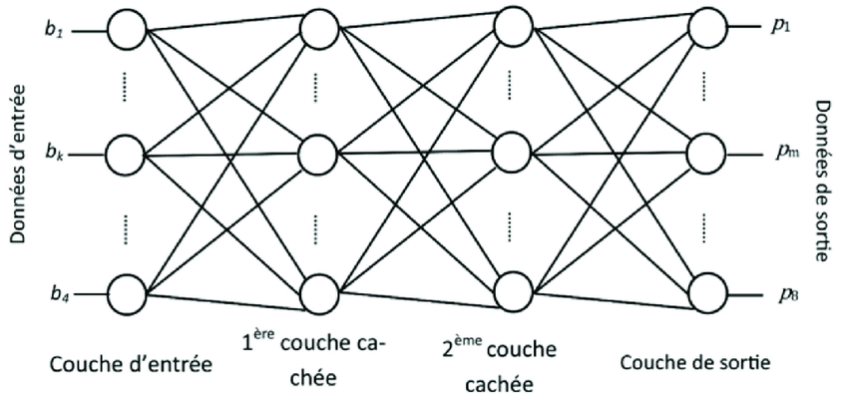


Figure 2.1 – Exemple de structure d'un RNA [36]

— **Couche d'entrée** : C'est la couche qui communique avec les données saisies. Les neurones de la couche d'entrée n'établissent aucun calcul, mais ils communiquent les données saisies aux neurones de la première couche cachée.
— **Couches cachées** : Les couche cachées du RNA représentent une partie importante dans la structure globale du RNA car ce sont elles qui récupèrent les signaux d'information de la couche d'entrée ou d'une couche cachée précédente. Ensuite, elles encodent ces signaux et envoient le signal de sortie de cette couche vers une autre couche cachée ou la couche de sortie.
— **Couche de sortie** : La couche de sortie présente un modèle de l'environnement externe. Le nombre de neurones de sortie est lié au nombre de signaux de sortie attendus.

2.2.2 Historique des RNA

L'histoire des réseaux de neurones peut être divisée en plusieurs périodes.

— **Première tentative**

La première étape vers les réseaux de neurones artificiels a eu lieu en 1943 lorsque (Pitts et al,), ont développé les premiers modèles de réseaux de neurones [117]. Les réseaux étaient basés sur des éléments simples considérés comme des dispositifs binaires à seuils fixes. Les résultats de leur modèle étaient sous la forme de fonctions logiques simples avec le caractère «tout ou rien» de l'activité nerveuse.

En 1949, (Hebb et al,), ont publié un travail qui a souligné le fait que les voies neuronales sont renforcées à chaque fois qu'elles sont utilisées, un concept fondamental et essentiel à la façon dont les humains apprennent [47].

— **Nouvelle technologie**

En 1958, (Rosenblatt et al,) ont mené un premier travail sur les "Perceptrons" [59]. Le "Perceptron" était un appareil électronique construit conformément aux principes biologiques et qui montrait une capacité d'apprentissage. Un autre système était l'ADALINE (ADAptive LInear Element) qui a été développé en 1960 par (Widrow et al,) [173]. La méthode utilisée pour l'apprentissage était différente de celle du Perceptron : elle utilisait la règle d'apprentissage des moindres carrés.

— **Décridibilisation des RNA**

En 1969, (Minsky et al,) ont écrit un livre sur les Perceptrons : An Introduction to Computational Geometry [119]. L'objectif visé était de discréditer la recherche sur les réseaux de neurones en montrant un certain nombre de problèmes fondamentaux et dans lesquels ils généralisaient les limites du perceptron mono-couche. Bien que les auteurs soient bien conscients que les perceptrons puissants ont plusieurs couches et que les perceptrons de base à anticipation de (Rosenblat et al,) ont trois couches, ils ont défini le Perceptron comme une machine à deux couches qui ne peut gérer que des problèmes linéairement séparables.

— **Développement**

En 1972, (Klopf et al,) ont développé une base d'apprentissage des neurones artificiels basée sur un principe biologique [99]. Ensuite en 1974, " Werbos" a développé une méthode d'apprentissage par rétro-propagation. Il a commencé a mettre en oeuvre cette approche en 1986.

En 1976, (Grossberg et al,) ont introduit la résonance adaptative comme théorie du traitement cognitif de l'information humaine par les réseaux de neurones artificiels [139].

— **Innovation et travaux récents**

L'algorithme de rétropropagation découvert à l'origine par (Werbos et al,) en 1974, a été redécouvert en 1986 avec le livre Learning Internal Representation by Error Propagation de (Rumelhart) [138]. La rétropropagation est une forme d'algorithme de descente de gradient utilisé par les réseaux de neurones artificiels pour minimiser l'erreur quadratique moyenne.

En 1985, l'Institut américain de la physique a commencé une conférence annuelle "Neural Networks for Computing". La première conférence ouverte sur les réseaux de neurones des temps modernes a eu lieu en 1987. Ensuite, la conférence internationale de l'IEEE sur les réseaux neuronaux s'est tenue à San Diego et la Société internationale des réseaux neuronaux (INNS) a été créée. En 1988, la revue INNS Neural Networks a été fondée, suivie de Neural Computation en 1989 et de l'IEEE Transactions on Neural Networks en 1990.

Entre 2009 et 2012, les réseaux de neurones récurrents et les réseaux de neurones à apprentissage profond ont été développés par notamment, (Schmidhuber et al,) [93].

Chez IBM en 2014, des chercheurs ont introduit le processeur (TrueNorth), avec une architecture similaire à celle existant dans le cerveau humain. Ensuite ils ont présenté le circuit intégré de la taille d'un timbre-poste capable de simuler le travail de millions de neurones et de 256 millions de synapses qui reçoivent les informations des autres neurones via l'axone et permettent donc aux neurones de communiquer entre eux en temps réel. Le système est capable d'exécuter de 46 à 400 milliards d'opérations synaptiques par seconde.

La prochaine partie sera consacrée à l'apprentissage des RNA qui constitue la phase la plus importante dans l'utilisation de cet outil.

2.3 Apprentissage des réseaux de neurones artificiels

L'apprentissage est la phase clé dans le développement d'un RNA. Durant cette étape, le comportement du réseau est modifié jusqu'à l'obtention du comportement désiré, et ce par l'ajustement des poids (connexions ou synapses) des neurones à une source d'information bien définie [47], [139], [138]. L'objectif essentiel de l'apprentissage est la résolution du problème par la prévision, la classification, ... etc. L'apprentissage consiste également à extraire des conformités parmi les données utilisées pour l'entraînement du réseau.

2.3.1 Règles d'apprentissage

Selon la littérature, les règles d'apprentissage peuvent être classifiées en trois catégories qui sont l'apprentissage supervisé, l'apprentissage non supervisé et l'apprentissage par renforcement.

— **Apprentissage supervisé**

Ce type d'apprentissage consiste à imposer au RNA un fonctionnement donné en forçant les sorties des réseaux à prendre des valeurs bien définies par modifications des les poids synaptiques [105].

L'apprentissage supervisé peut avoir lieu lorsqu'on dispose de toutes les données des variables d'entrées et de sorties. L'objectif est de prévoir les variables de sortie lorsque l'on dispose de nouvelles données en entrée en ayant une approximation de la fonction d'activation.

Le RNA se comporte donc comme un filtre dont les paramètres de transfert sont

ajustés à partir des couples entrée-sortie présentés.

L'adaptation des paramètres du réseau s'effectue en utilisant un algorithme d'optimisation. L'initialisation des poids synaptiques se fait le plus souvent de manière aléatoire.

— **Apprentissage non supervisé**

L'apprentissage non supervisé aussi appelé algorithme par compétition. Contrairement à l'apprentissage supervisé, seules les valeurs d'entrée sont disponibles [96].

Dans ce cas, les entrées provoquent une auto-adaptation du réseau afin d'engendrer des valeurs de sortie qui soient proches en réponse à des valeurs d'entrée similaires. L'objectif de ce type d'apprentissage est de faire une modélisation de la structure ou de la distribution des données.

— **Apprentissage par renforcement**

L'apprentissage par renforcement est une méthode d'apprentissage qui consiste à laisser l'algorithme apprendre des erreurs commises par le RNA [10].

L'objectif de cette règle d'apprentissage est d'exécuter la base de données d'entrées/ sorties par une interaction continue avec l'environnement afin de minimiser un critère de performance.

Pour pouvoir prendre les bonnes décisions, les RNA se retrouvent directement confrontés à des choix. Si le RNA se trompe, il est " pénalisé ". Au contraire, si il prend la bonne décision, il est " récompensé ".

2.3.2 État de l'art sur les méthodes d'apprentissage

Le processus d'apprentissage est le même pour n'importe quelle structure du réseau de neurones. La première tache consiste à faire une classification. La deuxième tache porte sur l'entraînement ou processus itératif, et la troisième tache, c'est l'obtention des poids ω_i permettant d'avoir $f(x_i) = y_i$ de façon a éliminer les erreurs entre les y_i obtenus et attendus.

Dans la littérature, il existe plusieurs algorithmes d'apprentissage classés en deux catégories [47].

— Apprentissage d'un perceptron

— Apprentissage des réseaux de neurones multi-couches

2.3.2.1 Apprentissage d'un perceptron

L'apprentissage d'un réseau de neurones de type perceptron est largement présenté dans la littérature. Plusieurs approches sont identifiées :

— **Règle de Hebb d'apprentissage**

La règle de Hebb ou théorie de Hebb ou théorie des assemblées de neurones a été établie par (Hebb et al,) en 1949. Elle est à la fois utilisée comme hypothèse en neurosciences et comme concept dans les réseaux neuronaux en mathématiques.
Cette théorie est souvent résumée par la formule : « des neurones qui s'excitent ensemble, se lient entre eux » (cells that fire together, wire together). C'est une règle d'apprentissage des RNA dans le contexte de l'étude d'assemblées de neurones [47]
Elle se base sur une hypothèse scientifique en neurosciences qui permet de décrire les changements d'adaptation neuronale dans le cerveau ou dans un réseau de neurones pendant le processus d'apprentissage. Le principe de la loi de Hebb consiste a dire que si deux neurones sont activés en même temps alors la force de connexion augmente. L'algorithme consiste a initialiser aléatoirement les coefficients synaptiques ω_{ij} de liaison entre les neurones Ni et Nj, puis à choisir

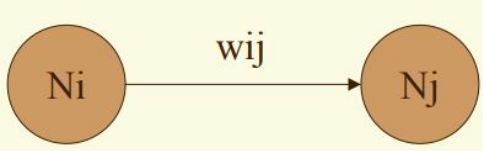


Figure 2.2 – Connexion entre les neurones [36]

un exemple qui regroupe une entrée définie (e) et une sortie définie (s) dans la base d'apprentissage. Pour cette entrée (e), il faut calculer à l'aide du réseau de neurones la sortie réelle (c). La dernière étape consiste à faire une comparaison entre la sortie définie et la sortie réelle : en cas d'égalité, il faut arrêter l'algorithme mais en cas de différence, la mise à jour des coefficient synaptiques est requise :

$$\omega_{ij} = \omega_{ij} + \mu \tag{2.1}$$

où μ une constante positive.

— **Apprentissage par réseau de Hopfield**

Le modèle de Hopfield a été présenté en 1982. Ce modèle très simple est basé d'une part sur le modèle des verres de spin de Ising et aussi sur le principe des mémoires associatives. C'est la raison pour laquelle ce type de réseau est dit associatif. Outre un intérêt pratique, ce réseau admet une analyse théorique précise et complète [79]. Il a par ailleurs, contribué à relancer les recherches sur les réseaux de neurones.
Le réseau de Hopfield est l'un des réseaux les plus simples, en particulier grâce à un apprentissage facile à mettre en oeuvre. Ce n'est qu'au cours de la phase de test que les problèmes les plus importants apparaissent.

La règle d'apprentissage proposée par Hopfield est basée sur la règle de Hebb. La règle de Hebb consiste à forcer les poids des liaisons entre les neurones actifs au même moment. Par contre, les poids seront diminués si les neurones sont dans des états contraires. Dans le cas de Hopfield, cette règle est légèrement étendue si l'on considère que deux neurones dans l'état précédent sont actifs. La règle de modification des poids devient donc :

$$\omega_{ij} = s_i^p * s_j^p \tag{2.2}$$

ou s_i et s_j représente les sorties du réseau de neurones et p le nombre d'exemples. On remarque que la phase d'apprentissage est immédiate en calculant directement les poids à l'aide de cette fonction. C'est d'ailleurs l'un des seuls outils qui permet un calcul aussi simple et analytique des poids.

2.3.2.2 Apprentissage des réseaux de neurones multi-couches

La deuxième catégorie d'approches concerne les réseaux avec des structures multi-couches :

— **Approche auto-organisatrice ou carte de Kohonen**

La méthode de la carte auto-organisatrice est un algorithme d'apprentissage et de classification développé par (Kohenen et al,) en 1988. Cet algorithme repose sur la méthode de quantification vectorielle et fait partie des approches d'apprentissage non supervisé [101].

Cette approche consiste à regrouper les informations fournies par le réseau de neurones tout en respectant la topologie fournie par l'espace d'observations.

Il existe différentes structures topologiques de la carte de Kohonen Figure 2.3 et 2.5 : L'algorithme d'apprentissage consiste à adapter les poids de telle manière

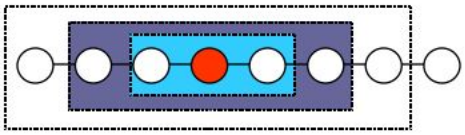


Figure 2.3 – Carte unidimensionnelle

que les exemples les plus proches dans l'espace soient associés aux mêmes neurones ou aux neurones les plus proches dans la carte.

Cet algorithme débute par la phase d'initialisation aléatoire des poids ω. Ensuite, à chaque itération t, il faut présenter un exemple de la base d'apprentissage $X(t)$. Ce dernier est choisi au hasard à l'entrée de la carte et comparé à tous les vecteurs poids. La dernière phase consiste à retrouver le neurone $j*$ gagnant qui est celui

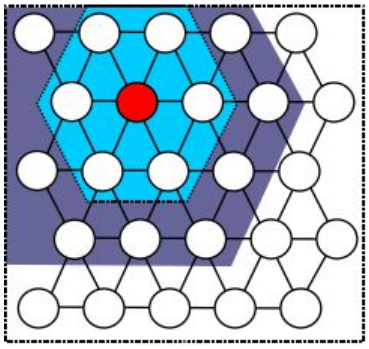


Figure 2.4 – Carte bi-dimensionnelle "voisinage hexagonal"

dont le vecteur poids $w_{j*}(t)$ est le plus proche de l'entrée $X(t)$ ou par définition :

$$d_N(X(t), \omega_{j*}(t)) = mind_N(X(t), \omega j(t)) \tag{2.3}$$

avec d_N la distance dans l'espace d'entrée, d la distance dans la carte X l'exemple choisi et j le neurone le plus proche de l'entrée choisie. La phase qui permet d'évaluer le voisinage du neurone gagnant dans la carte est définie par :

$$h_{j*}(j,t) = h(d(j,j*),t) \tag{2.4}$$

où h est la fonction voisinage. Finalement, la phase de coopération qui consiste à faire la mise des poids pour tous les neurones de la carte (l'adaptation) est d'autant plus forte que les neurones voisins de $j*$ sont donnés par les deux équations suivantes :

$$\omega_j(t+1) = \omega_j(t) + \Delta\omega_j(t) \tag{2.5}$$

$$\Delta\omega_j(t) = \epsilon(t) h_{j*}(j,t) * (X(t) + \omega_j(t)) \tag{2.6}$$

où ϵ est le pas d'apprentissage.

— **Algorithme de rétropropagation**

L'algorithme de rétropropagation consiste à faire circuler vers l'avant les données d'entrées afin d'obtenir une sortie calculée par le réseau, ensuite faire une comparaison entre la sortie calculée et la sortie réelle [173]. Cette comparaison est faite en minimisant une fonction d'erreur comme dans le cas du perceptron linéaire. L'étape la plus difficile par la suite consiste à calculer la contribution à cette erreur de chacun des poids synaptiques. En effet, les poids sont modifiés de telle sorte qu'à l'itération qui suit, l'erreur entre la sortie calculée et désirée soit minimale, en prenant en considération la présence des couches cachées. L'erreur est rétro-propagagée vers l'arrière jusqu'à la couche d'entrée en modifiant la pondération.
L'algorithme est initialisé par la définition de l'ensemble d'apprentissage composé

de K paires de vecteurs :

$$(x_1, o_1), (x_2, o_2), ..., (x_k, o_k) \tag{2.7}$$

où le vecteur d'entrée est :

$$x_p = (x_{p,0,1}, x_{p,0,2}, ..., x_{p,0,n})^t \tag{2.8}$$

le vecteur de la sortie désirée :

$$o_p = (o_{p,1}, o_{p,2}, ..., o_{p,m})^t \tag{2.9}$$

et le vecteur de la sortie réelle :

$$y_p = (y_{p,1,1}, y_{p,1,2}, ..., y_{p,l,m})^t \tag{2.10}$$

Par définition, l'entrée et la sortie du noeud sont définies par :

$$h_{(p,j,k)} = \sum_{i=0}^{n} W_{(j,k,i)} x_{(p,j-1,i)} \tag{2.11}$$

Et :

$$F(h_{(p,j,k)}) = \sum_{i=0}^{n} W_{(j,k,i)} x_{(p,j-1,i)} + \theta_{j,k} \tag{2.12}$$

Avec :

$W_{(j,k,i)}$: la connexion entre le neurone k et de la couche cachée $j-1$ et le neurone i de la couche j.

$h_{(p,j,k)}$: l'entrée complète du neurone k pour l'échantillon p de la couche j.

$\theta_{j,k}$: le poids fictif du neurone k de la couche j.

Le terme d'erreur totale commise sur les noeuds de sortie est défini par :

$$E_p = \frac{1}{2} \sum_{i=0}^{m} (O_{p,k} - x_{p,l,k})^2 = \frac{1}{2} \sum_{i=0}^{m} \delta_{p,k}^2 \tag{2.13}$$

avec $\delta_{p,k}$ l'erreur commise sur la sortie.

L'objectif est donc de minimiser cette erreur E_p. Le gradient est donc calculé pour chaque poids ω, puis, à leurs tour, ces poids sont modifiés dans le sens inverse du gradient. Le gradient de l'erreur est défini par :

$$\nabla E_p = \frac{\partial E_p}{\partial \omega_{l,j,k}} = -(O_{p,k} - x_{p,l,k}) \frac{\partial f(y_{p,l,k})}{\partial \omega_{l,k,j}} \tag{2.14}$$

On obtient alors :

$$\nabla E_p = \delta_{p,k} x_{p,l,k}(1 - x_{p,l,k}) x_{(p,l-1,k,j)} \tag{2.15}$$

La mise à jour des poids est exprimée par :

$$\nabla_p \omega l, k, j(t) = \lambda(O_{p,k} - xp, l, k) f'(y_{p,l,k}) x_{p,l-1,j} \tag{2.16}$$

Où λ est le pas compris entre 0 et 1 strictement.

2.4 Dimensionnement des réseaux de neurones artificiels

Dans cette partie, le focus est fait sur le dimensionnement d'un RNA. En effet, le dimensionnement consiste à établir la structure optimale du RNA en fonction de la complexité de la base d'apprentissage. La complexité d'un réseau avec plusieurs couches cachées est liée au nombre de neurones et au nombre de paramètres correspondants à ajuster. Dans le cas d'un réseau neuronal, est mesuré par le nombre de connexions synaptiques. Ce qui fait que l'addition d'un neurone dans la couche cachée augmente le nombre de paramètres à ajuster.

2.4.1 État de l'art sur les méthodes de dimensionnement

Dans la littérature, il existe plusieurs méthodes de dimensionnement, que l'on peut classer en deux catégories : les méthodes de dimensionnement statique (hors ligne) et les méthodes de dimensionnement dynamique (en ligne). Dans cette partie, un état de l'art sur les différentes méthodes de dimensionnement identifiées dans la littérature est présenté.

2.4.1.1 Méthodes de dimensionnement statique

— **Méthodes par décomposition SVD**
En 2006, (Teoh et al,) ont proposé une méthode qui permet d'augmenter le nombre de neurones dans une seule couche cachée du réseau de neurones, en utilisant des propriétés de la méthode de décomposition en valeurs singulières SVD (Singular Value Decomposition). Cette décomposition permet d'évaluer la généralisation du réseau avec une seule couche cachée [160]. La capacité de généralisation est mesurée par le degré d'indépendance linéaire des neurones dans la couche cachée, qui peut être indirectement quantifié par les valeurs singulières obtenues.
La décomposition SVD est une généralisation de la décomposition en éléments

propres, qui peut être utilisée pour analyser des matrices rectangulaires (la décomposition en éléments propres est définie seulement pour des matrices carrées). Par analogie, la SVD permet d'écrire une matrice en deux matrices simples, l'idée principale est de décomposer une matrice rectangulaire en trois matrices simples : deux matrices orthogonales et une matrice diagonale. Une matrice semi-définie positive peut être obtenue comme le produit d'une matrice par sa transposée. Cette matrice est évidemment carrée et symétrique, mais aussi ses valeurs propres sont toutes définies positives ou nulles et les vecteurs propres correspondant aux valeurs propres différentes sont orthogonaux.
Étant donnée une matrice réelle H, l'application de la SVD aboutit à la transformation orthogonale $H = UVT^T$, avec U une matrice orthonormée, ($U^TU = I$), et V une matrice diagonale qui contient les valeurs propres de H.
Pour estimer le nombre de neurones dans la couche cachée, il faut construire l'hyperplan de la matrice H dans le sens géométrique et définir son rang.

— **Méthode de K-moyen où algorithme de glouton**
C'est une méthode d'optimisation proposée par (Bang et al,) qui combine deux algorithmes, celui dit "K-moyen++ (K-Means++)" et celui dit de "glouton (Greedy algorithm)" [30]. Cette approche permet de retrouver le nombre optimal de neurones de la couche cachée d'un réseau multi-couches.
L'algorithme de glouton exécute une évaluation itérative de chaque solution. Il rejette ensuite les solutions les moins optimales par rapport à la solution la mieux connue, en acceptant et mettant ensuite à jour la solution la mieux connue avec les solutions qui révèlent être meilleures, qui dans ce cas est les valeurs du nombre de neurones dans la couche cachée.
Cependant, utiliser l'algorithme de glouton ne produit pas toujours la solution globalement optimale, car il ne considère pas toutes les solutions possibles du problème [44]. La méthode proposée par (Arthur et al,) a pour but de surmonter cet inconvénient en utilisant l'algorithme de K-moyen ++ [11], pour limiter l'espace de recherche pour l'algorithme de glouton en se concentrant sur les valeurs de fonction de coût qui sont susceptibles de mener au minimum global en renonçant aux valeurs qui sont moins susceptibles de faire ainsi. Cette méthode aboutira à plusieurs solutions, ce qui permet d'augmenter ainsi la probabilité de l'algorithme de glouton d'obtenir une solution globalement optimale.
— **Méthode PCA et apprentissage par renforcement**
Cette méthode a pour but de fixer le nombre de neurones de la couche cachée, en utilisant deux algorithmes : celui de l'analyse par composante principale PCA

(Principal componenent Analysis) et celui d'apprentissage par renforcement RL (Reinforcement Learning) [109].

Le PCA est un outil de compression de nombre de données ou de réduction de dimensions. Il peut être utilisé pour réduire un grand ensemble de variables à un petit ensemble qui contient toujours la plupart des informations dans le grand ensemble.

L'idée globale de la PCA est d'éliminer la corrélation entre les données et de projeter les données originales sur un sous-espace de dimension plus petite. L'algorithme se fait en quatre étapes : relever la moyenne, calculer la matrice de covariance Σ, calculer la valeur propre λ et les vecteurs propres Σ et U et finalement déterminer la valeur k pour le PCA.

L'apprentissage des neurones cachés est complété par l'approche RL qui est exécutée en 2 étapes : premièrement, initialiser trois valeurs de nombre de neurones selon la gamme (taille des composants principaux choisis) calculée par PCA, deuxièmement, l'approche par RL à son tour recherche le nombre de neurones le plus performant.

Les méthodes présentées précédemment sont des méthodes dites statiques, l'architecture du RNA étant ajustée dans la phase d'apprentissage. Durant la phase de test ou celle de validation il n'est plus possible d'agir sur le RNA. C'est la raison pour laquelle plus d'importance sera accordée aux méthodes de dimensionnement dynamique.

2.4.1.2 Méthodes de dimensionnement dynamique

— **Méthode Extreme Learning Machine et Factorisation QR**

Cette approche est basée sur un algorithme d'apprentissage dit "ELM" (Extreme Learning Machine) et la factorisation QR [176]. Cette méthode est capable de déterminer le nombre de neurones dans la couche cachée automatiquement par la factorisation QR de la matrice de sortie de la couche cachée. Les poids et biais peuvent être progressivement mis à jour pendant la croissance de nombre de neurones.

L'ELM est un algorithme d'apprentissage qui a été proposé par (Hiang et al,) pour le dimensionnement des réseaux ne comportant qu'une seule couche cachée [85]. La particularité de cette approche est que l'ELM produit aléatoirement ses poids et biais entre la couche d'entrée et la couche cachée, et calcule analytiquement les poids entre la couche cachée et la sortie par la résolution de problèmes de moindres carrés. En raison de sa vitesse d'apprentissage et sa meilleure capacité de généralisation, l'ELM a été largement appliqué dans plusieurs domaines tels que la reconnaissance d'images, la prédiction, la commande, le traitement des

signaux,...[85].

— **Méthode Pruning–ELM et Incemantal–ELM**
En 2017, (Yongjiao et al,) ont proposé des méthodes pour déterminer le nombre optimal de neurones ainsi qu'une stratégie de filtrage pendant le processus d'ajout de neurones [178]. Cette approche fusionne deux algorithmes, le premier, Pruning–ELM qui a pour but d'éliminer les neurones inutiles dans le réseau et le second, Incemantal–ELM, qui rajoute les neurones dans le réseau. Pour garder l'efficacité du réseau, l'ELM diminue les neurones dans le réseau et maintient uniquement les neurones qui peuvent satisfaire la demande de généralisation. Ainsi, une méthode est d'ajouter une stratégie automatique dans ELM pour pouvoir ajuster la structure nécessaire dans des problèmes différents et des applications, et améliorer la flexibilité des algorithmes.
Le but de cet algorithme est de garder les neurones efficaces pour résoudre des problèmes et de supprimer les neurones inutiles et inefficaces qui influencent négativement sur le RNA.

— **Méthode VC OS-ELM (Variable Complexity Online Sequential-Extreme Learning Machine)**
Cette méthode est très proche à celle présentée précédemment. Elle permet d'enlever ou d'ajouter les neurones dans la couche cachée. Cette méthode a été appliquée dans la prédiction de débit d'écoulement [107].
Pour évaluer les performances de VC OS-ELM par rapport à OS-ELM, des expériences ont été menées sur deux données hydrométéorologiques de deux sources d'eau « Englishman River station » et « Stave River Station ».

2.5 Apprentissage et dimensionnement du RNA par ELM : application sur le réglage de la MPC

Dans cette partie, nous appliquerons l'approche ELM pour l'apprentissage et le dimensionnement qui seront par la suite exploités pour le calcul des paramètres de la MPC.

2.5.1 Apprentissage du RNA par l'ELM

L'ELM est le nom donné à un réseau de neurones employant les poids synaptiques pour un algorithme d'apprentissage rapide. Typiquement, au lieu d'optimiser les paramètres du RNA par un algorithme itératif comme la rétro-propagation [138], l'algorithme d'ELM

a pour but de faire l'apprentissage d'une seule couche cachée d'un RNA feedforward. Comme défini précédemment, l'ELM a connu une forte croissance depuis sa découverte cette méthode permet de faire varier les paramètres du RNA sans itération. A partir d'une base de données entrée-sortie, l'ELM génère d'une manière aléatoire les données entre la couche d'entrée et la couche cachée. Ensuite, il calcule analytiquement les poids de sortie par la méthode des moindres carrés. Malgré un choix aléatoire des poids de la couche cachée, il a été prouvé que l'ELM est capable de réaliser l'approximation de n'importe quelle fonction. En raison de sa vitesse et son large spectre d'application, l'ELM a été utilisé largement au cours de la dernière décennie [85]. La Figure 2.5 présente l'architecture du RNA pour l'utilisation de l'ELM :

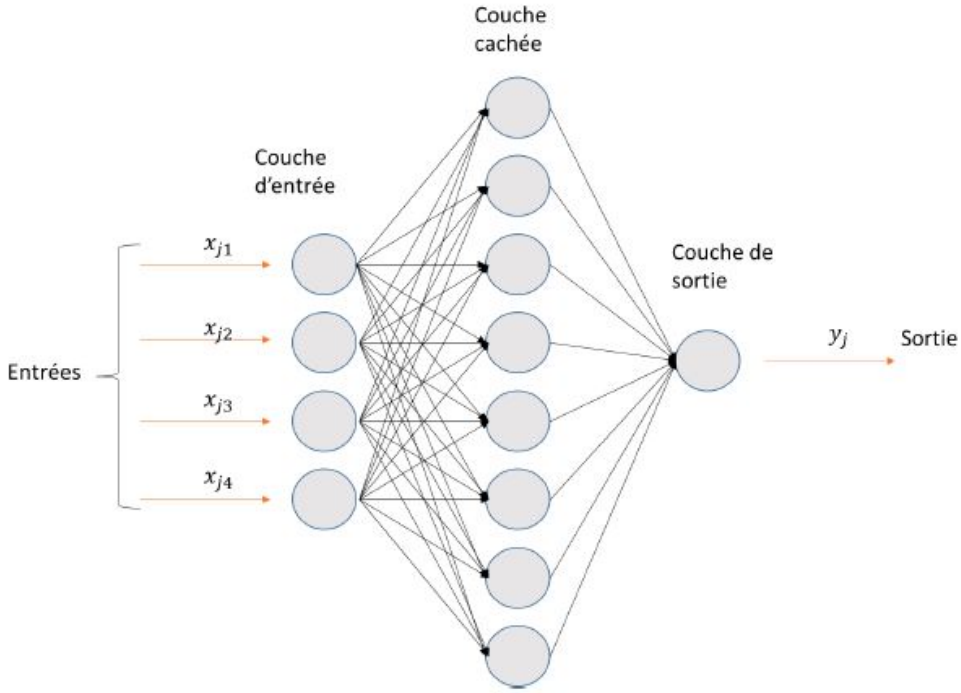


Figure 2.5 – Architecture RNA-ELM

L'algorithme pour former ce réseau est décrit comme suit :
Soit un ensemble de N échantillons (base d'apprentissage) (x_i, y_j) où x_i et y_j désignent respectivement les vecteurs d'entrée et de sortie du RNA. Alors, la sortie du RNA pour un nombre M de neurones cachés peut être décrit par :

$$y_j = \sum_{\substack{i=0 \\ i \neq i_0}}^{M} \beta_i f(w_i x_j + b_i), \quad j \in [1, N] \tag{2.17}$$

où f est la fonction d'activation, ω_i représente le poids entre la couche d'entrée et la couche de sortie, b_i le biais attribué à chaque neurone de la couche cachée et β_i le poids

de sortie qui lie chaque neurone de la couche cachée à la couche de sortie.

L'équation 2.17 peut être mise sous la forme vectorielle :

$$H\beta = Y \tag{2.18}$$

où H est la matrice de sortie de la couche cachée, définie par :

$$H = \begin{bmatrix} f(w_1 x_1 + b_1) & \dots & f(w_M x_1 + b_M) \\ f(w_1 x_2 + b_1) & \dots & f(w_M x_2 + b_M) \\ \vdots & & \\ \vdots & & \\ f(w_1 x_N + b_1) & \dots & f(w_M x_N + b_M) \end{bmatrix} \tag{2.19}$$

et :

$$\beta = \begin{bmatrix} \beta_1 & \beta_2 & \cdots \beta_M \end{bmatrix}^T \tag{2.20}$$

$$y = \begin{bmatrix} y_1 & y_2 & \cdots y_N \end{bmatrix}^T \tag{2.21}$$

La détermination du vecteur β est faite analytiquement par la résolution de l'erreur quadratique selon la méthode des moindres carrés.

$$S = || \ H\beta - Y \ ||^2 \tag{2.22}$$

$$(H\beta - Y)^T (H\beta - Y) = 0 \tag{2.23}$$

soit alors :

$$\beta = (H^T H)^{-1} H \times Y = H^+ \times Y \tag{2.24}$$

où H^+ est la pseudo inverse de la matrice H.
En théorie, cet algorithme reste valable pour toute fonction d'activation infiniment différentiable $f(\omega, x, b)$ sigmoïde, gaussienne, sinus, cosinus, exponentielle.... Ainsi, pour des raisons de calcul inverse des matrices et pour éviter une distorsion des informations, le nombre de neurones doit être inférieur au nombre des exemples dans la base d'apprentissage [85]. Le calcul des b_i et w_i est fait d'une manière aléatoire.

2.5.2 Dimensionnement du RNA par l'ELM

L'approche dite "Online Sequential Extreme Learning Machine" (OS-ELM), est inspirée du concept d'apprentissage hors ligne d'ELM. Les données à apprendre peuvent être présentées d'une façon séquentielle, c'est-à-dire exemple par exemple ou groupe par

groupe avec une taille variable ou fixe. Une fois l'apprentissage fini, ces exemples sont supprimés. Comme décrit dans [106], l'algorithme OS-ELM pour le dimensionnement se présente en deux phase essentielles :

— **Phase d'initialisation**

-L'apprentissage est initialisé à l'aide d'un nombre d'exemples bien définis. Soit N_0 le nombre initial des exemples, qui doit être plus grand que le nombre de neurones de la couche cachée.

-Initialisation des paramètres aléatoires du RNA (w_i et b_i).

-Calcul de la matrice initiale de sortie du RNA $H_0 = [f(w_i, x_j, b_i)]$ où i est le nombre de neurones et j représente le nombre des exemples.

-Calcul du vecteur poids entre les couches d'entrée et cachée.

$$\beta_0 = (H_0^T H_0)^{-1} H_0 * Y_0 = P_0 H_0^T Y_0 \tag{2.25}$$

avec :

$$P = P_0 H_0^T Y_0, K_0 = H_0^T H_0 = P^{-1} \tag{2.26}$$

— **Phase d'apprentissage séquentiel**

Pour un nombre d'exemples définis par k+1, N_{k+1} représente les nouveaux exemples qui arriveront par la suite.

-Calcul de la nouvelle matrice de sortie H_{k+1}.

-Mise à jour des poids de sortie selon l'équation :

$$\beta^{k+1} = \beta^k + K_{k+1}^{-1} * H_{k+1}^T (Y_{k+1} - H_{k+1}\beta^k) \tag{2.27}$$

avec :

$$K_{k+1} = K_k + H_{k+1}^T H_{k+1} \tag{2.28}$$

La complexité de la problématique réside dans la base de données d'apprentissage qui devient de plus en plus grande alors que le nombre de neurones de la couche cachée reste fixe. Donc, un premier algorithme est nécessaire pour trouver le nombre de neurones pour la base de données initiale (dimensionnement hors ligne). Ensuite, un autre algorithme est requis pour faire varier le nombre de neurones vis-à-vis les données qui arrivent par la suite (dimensionnement en ligne).

Une simple boucle « while » avec une condition sur l'erreur quadratique moyenne permet de rajouter des neurones, mais cela cause un problème de divergence de l'EQM. Pour éviter la problématique liée à la divergence, un algorithme d'EM-ELM (Erreur Minimized Extreme Learning Machine) sera employé [61]. Pour

chaque nouveau neurone rajouté, les poids de sortie sont mis à jour d'une manière incrémentale évitant ainsi le calcul inverse des matrices.
Soit un ensemble de bases de données d'apprentissage $(x_i, t_i)_{i=1}^{N}$, la condition sur l'erreur quadratique EQM est notée ϵ/. L'algorithme d'EM-ELM est présenté en deux phases :

— **Phase d'initialisation**

-Générer aléatoirement les paramètres d'entrées du RNA (w_1, b_1).

-Calculer la matrice de sortie du RNA (h_0) :

$$h_0 = [f(w_1, b_1, x_1), ..., ..., f(w_1, b_1, x_N)] \tag{2.29}$$

-Calculer l'erreur de prédiction $E(h_0)$:

$$E(h_0) = \left|\left|h_0 h_0^+ Y - Y\right|\right| \tag{2.30}$$

— **Phase de récurrence**

La phase de récurrence est formulée comme étant un algorithme d'incrémentation de nombre de neurones dans la couche cachée :

Tant que $E(H_j) > \epsilon$

-Ajouter un nouveau neurone dans le réseau. Le nombre de neurones dans le réseau est $L_{j+1} = L_j + 1$ et la matrice de sortie du RNA correspondante est $H_{j+1} = [H_j, \delta h_j]$, où :

$$\delta h_j = [f(w_{Lj+1}, b_{Lj+1}, x_1), ..., f(w_{Lj+1}, b_{Lj+1}, x_N)]^T \tag{2.31}$$

-Mise à jour du vecteur de poids de sortie β.

$$D_j = \frac{\delta h_j^T (I - H_j H_j^+)}{\delta h_j^T (I - H_j H_j^+) \delta h_j} \tag{2.32}$$

$$U_j = H_j^+ - H_j^+ \delta h_j D_j \tag{2.33}$$

$$B_{j+1} = H_{j+1}^+ Y \tag{2.34}$$

$$j = j + 1 \tag{2.35}$$

Fin tant que

Cet algorithme permet donc une mise à jour des poids sans causer de divergence au niveau de la sortie.

2.6 Application de l'ELM au réglage de la MPC

L'objectif de cette partie est de montrer l'influence de la taille de la base de données d'apprentissage et le nombre de neurones de la couche cachée sur les performances et sur la capacité de généralisation du réseau choisi.
Nous disposons d'une base de données de 12000 éléments constituée à partir d'une approche de réglage analytique de la MPC [166]. En effet, nous avons choisi aléatoirement 12000 systèmes SISO du premier ordre. Pour chacun de ces systèmes, les paramètres de réglages seront calculés et les performances obtenues seront relevées en appliquant la loi de commande. Cette base de données est composée des entrées et sorties qui sont eux même les entrées et sorties du RNA, nous retrouvons alors : en entrée du réseau les performances obtenus et en sortie les paramètres de réglage.
Dans cette partie, l'objectif est de comparer dans la phase de test, les résultats obtenus par l'approche de réglage analytique avec ceux obtenus au moyen du RNA. Cette comparaison permettra de conclure sur l'efficacité des RNA tout en montrant l'importance du choix de la structure et de la taille de la base de données d'apprentissage.
En entrée du réseau de neurones, nous avons choisi de mettre les performances qui sont le temps de montée, l'erreur de poursuite et l'indice de stabilité (sera défini par la suite). En sortie, nous retrouverons les paramètres de réglage. Dans cette partie, nous nous sommes intéressés à l'horizon de prédiction et l'horizon de commande, le but étant de montrer l'influence de la structure du réseau de neurones sur les performances. Dans le chapitre qui suivra, nous introduirons également le facteur de pondération.
Nous allons commencer par choisir à chaque fois le nombre d'exemples à prendre en compte dans la phase d'apprentissage et le nombre de tests à effectuer pour voir le niveau de généralisation. Ensuite, nous montrerons les résultats obtenus pour l'horizon de prédiction et l'horizon de commande ainsi que le nombre de neurones nécessaires dans la couche cachée. A chaque phase de test, nous allons comparer les résultats obtenus à l'aide du RNA aux résultats analytiques dont on dispose obtenus par [166].

2.6.1 Application sur 20 données dans la base d'apprentissage et 500 tests

Dans un premier temps, nous avons choisi aléatoirement 20 exemples pour la phase d'apprentissage et avons effectué un test sur 500 nouveaux exemples. La comparaison est faite avec les données calculées par l'approche de réglage analytique [166]. Les Figures 2.6 et 2.7, représentent les horizons de commande et de prédiction et le niveau de précision obtenu avec 20 données d'apprentissage.

Les Figures 2.6 et 2.7, montrent que le réseau de neurones n'a pas eu la capacité de

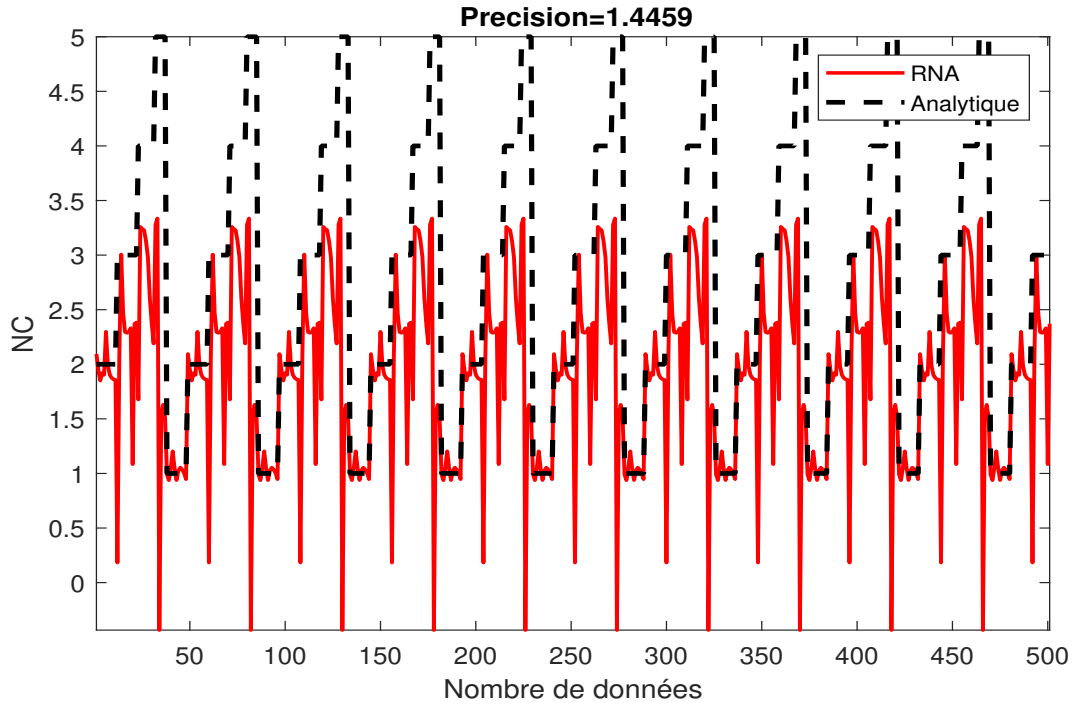


Figure 2.6 – Horizon de commande pour 20 données et 500 tests

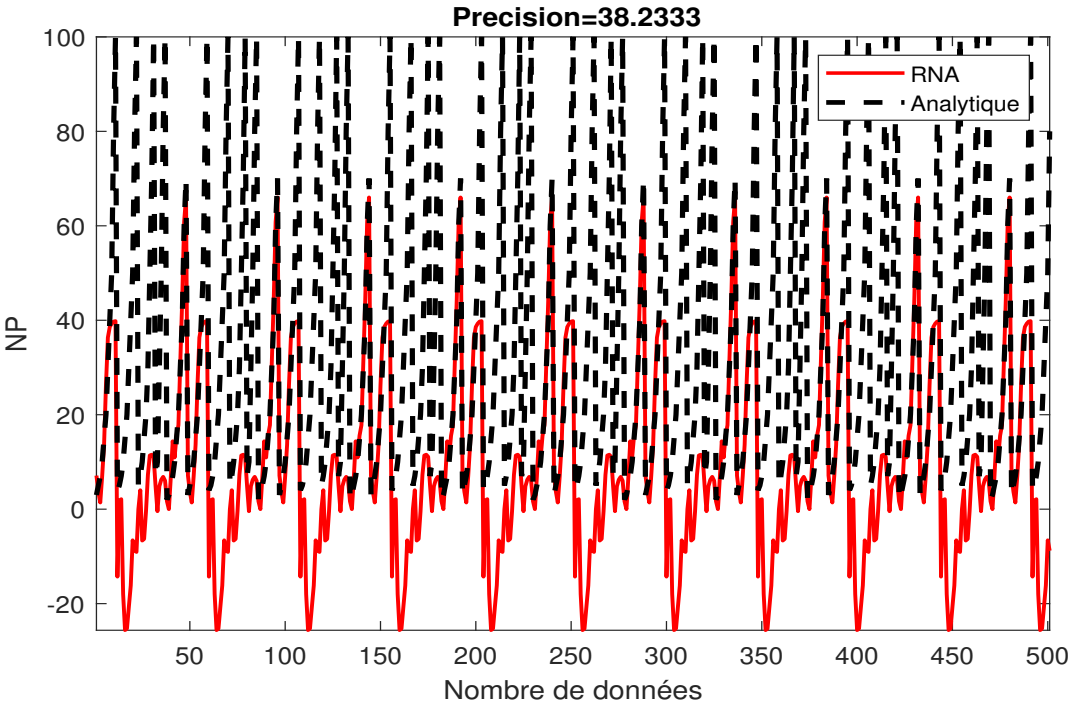


Figure 2.7 – Horizon de prédiction pour 20 données et 500 tests

s'approcher des valeurs obtenues par la méthode analytique. Ce phénomène est connu sous le nom de sous-apprentissage. Il se manifeste dans deux cas : soit la base d'apprentissage

n'est pas assez suffisante, soit le nombre de neurones n'est pas en adéquation avec le nombre de données dans la base.
Les Figures 2.8 et 2.9, représentent respectivement l'erreur quadratique moyenne et le nombre de neurones obtenus pour cette base d'apprentissage par la méthode d'OS-ELM, en fonction du nombre d'itérations :

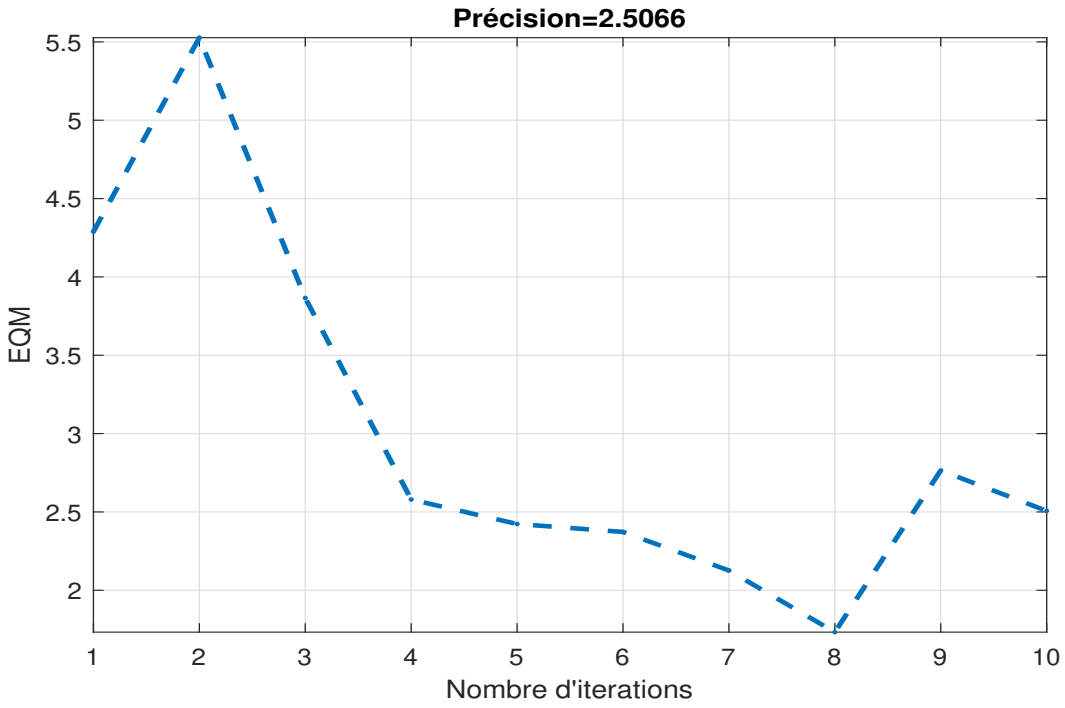


Figure 2.8 – Erreur quadratique moyenne pour 20 données

On constate qu'avec l'augmentation du nombre de neurones, l'EQM converge vers une valeur finale qui n'est pas la même dans la phase de test. On a pu identifier également que le nombre de neurones continue à augmenter sans cesse. Par contre, l'EQM reste toujours élevée en dépassant un seuil de 2. Cela s'explique du fait que le phénomène de sous-apprentissage est causé par le nombre de neurones insuffisant et également le nombre d'exemples insuffisant pris en compte dans la phase d'apprentissage.

2.6.2 40 données dans la base d'apprentissage et 500 tests

Dans cette partie, nous allons élargir le nombre de données dans la phase d'apprentissage en passant de 20 exemples à 40 exemples pour voir l'influence de ce changement sur l'EQM entre l'approche par RNA et la méthode analytique.
Les Figures 2.10 et 2.11, représentent les horizons de commande et de prédiction et le niveau de précision obtenu avec 40 données d'apprentissage.

Avec plus de données dans la base de données d'apprentissage, l'erreur entre les

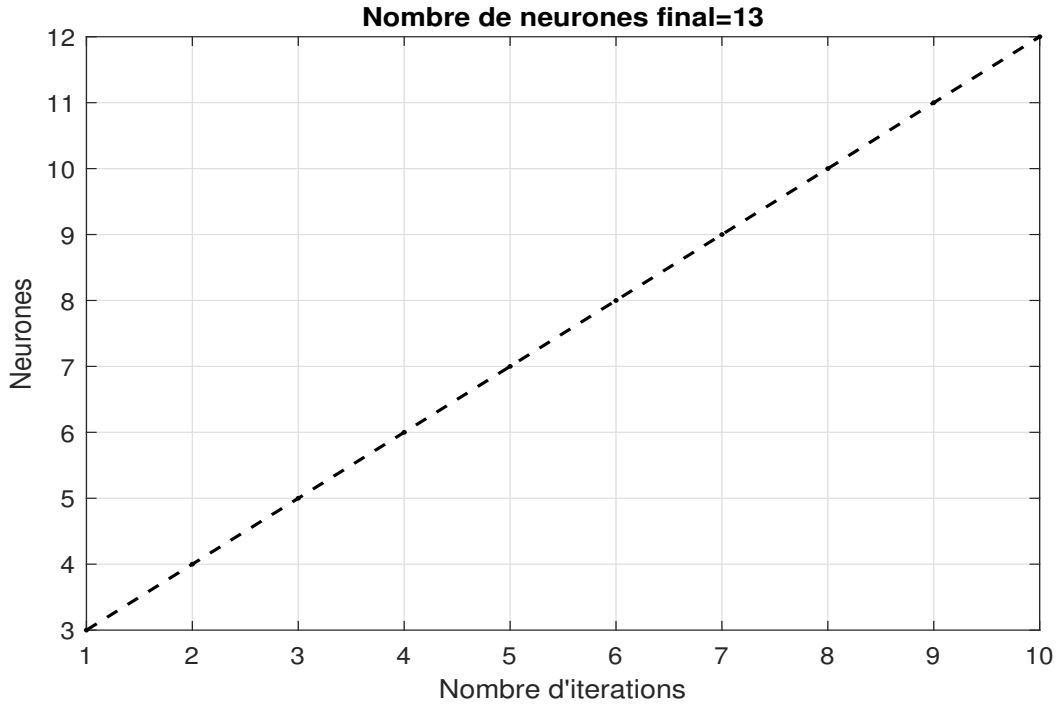


Figure 2.9 – Nombre de neurones de la couche cachée en fonction du nombre d'itérations pour 20 données

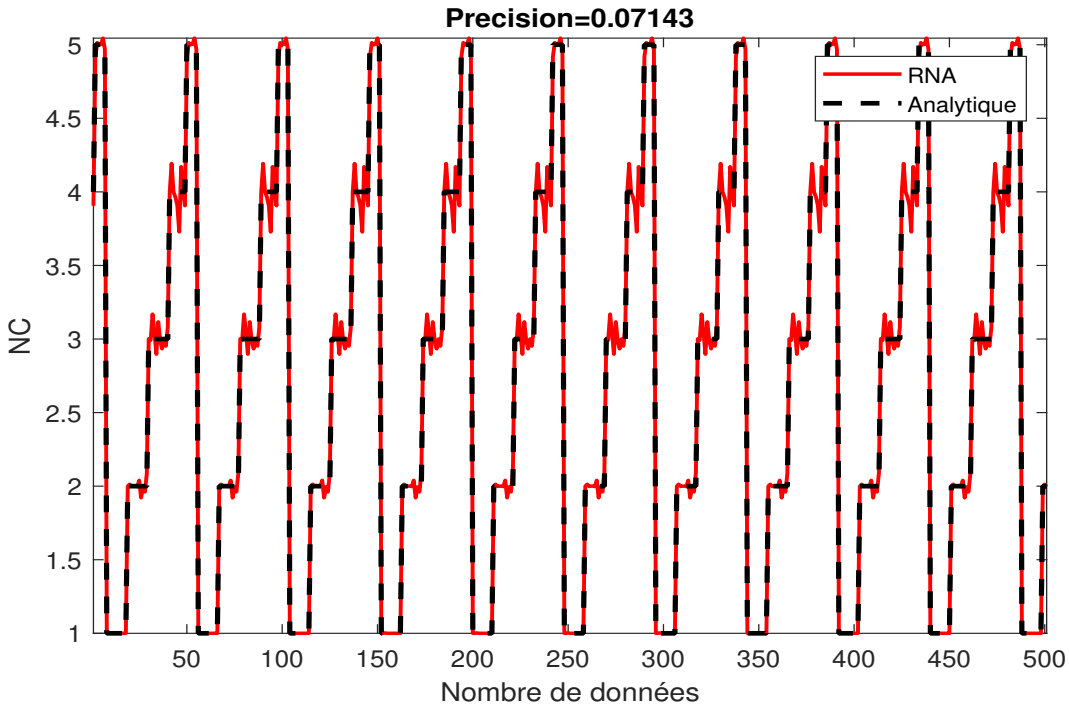


Figure 2.10 – Horizon de commande pour 40 données et 500 tests

résultats analytiques et les résultats obtenus au moyen des RNA est moins élevée, comme

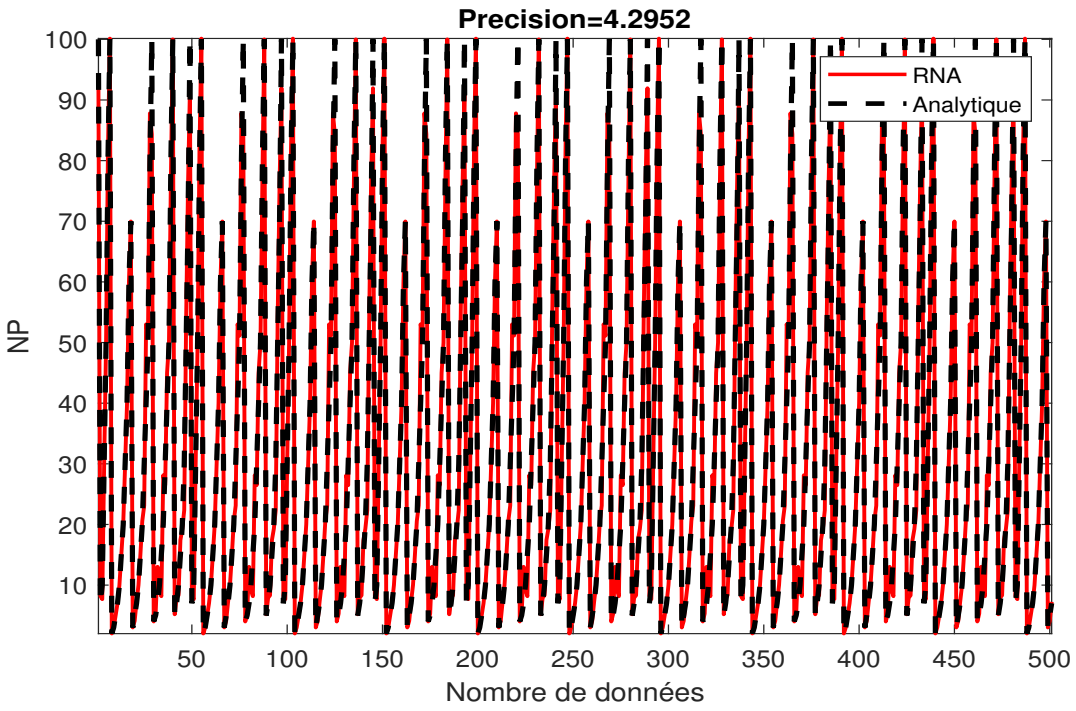


Figure 2.11 – Horizon de prédiction pour 40 données et 500 tests

on peut le constater clairement sur les Figures 2.10 et 2.11, où la courbe rouge du RNA et la courbe noire des valeurs obtenues analytiquement sont quasiment superposées.
Les Figures 2.12 et 2.13, représentent respectivement l'erreur quadratique moyenne et le nombre de neurones obtenus pour une base d'apprentissage de 40 exemples et un test sur 500 éléments.

Ces résultats montrent qu'à chaque augmentation de l'EQM, le RNA s'adapte en rajoutant un nouveau neurone dans la couche cachée jusqu'à la fin de la phase d'apprentissage pour minimiser cette erreur. Au niveau de la précision, on a pu remarquer clairement que le RNA obtient une généralisation adéquate en la comparant à celle obtenue par la Figure 2.8 avec une erreur de précision de "2.58". Nous pouvons également conclure que le phénomène de sous-apprentissage était dû au nombre de données insuffisant.

2.6.3 300 données dans la base d'apprentissage et 500 tests

Afin d'éviter le phénomène de sous-apprentissage et d'obtenir une meilleure généralisation par le RNA, et une meilleure précision sur les valeurs prédites tout en évitant le phénomène de sur-apprentissage, nous avons élargi la base de donnée d'apprentissage en injectant 300 données dans cette base et effectué un test sur 500 éléments.
L'algorithme de dimensionnement automatique a été lancer en parallèle pour trouver le

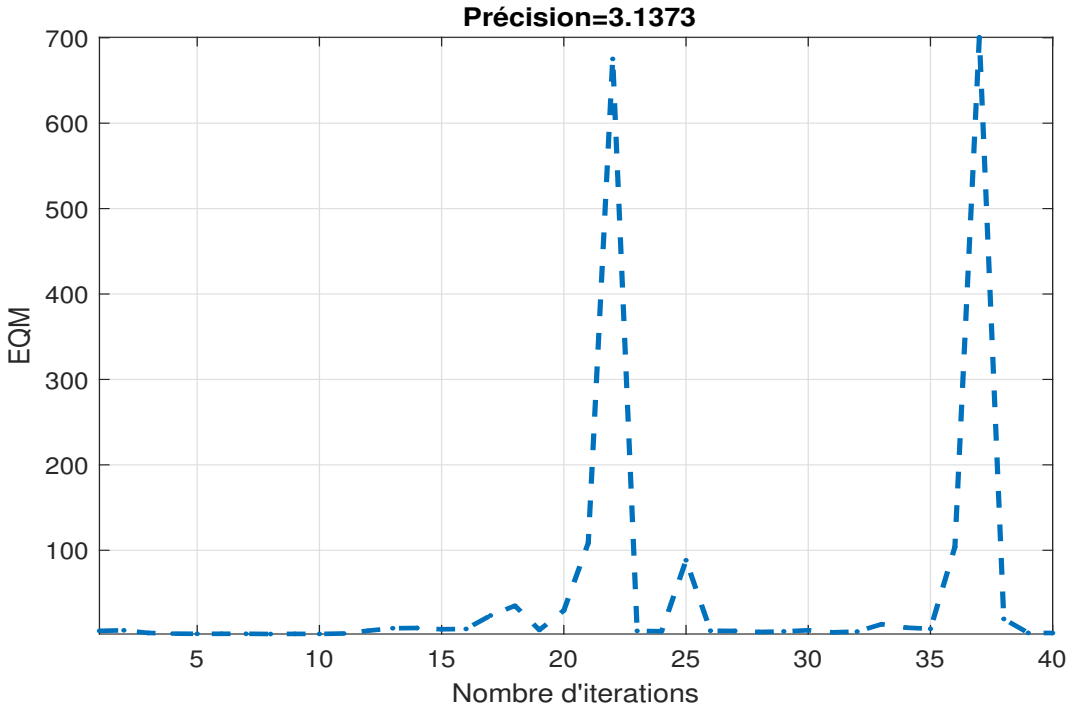


Figure 2.12 – Erreur quadratique moyenne pour 40 données

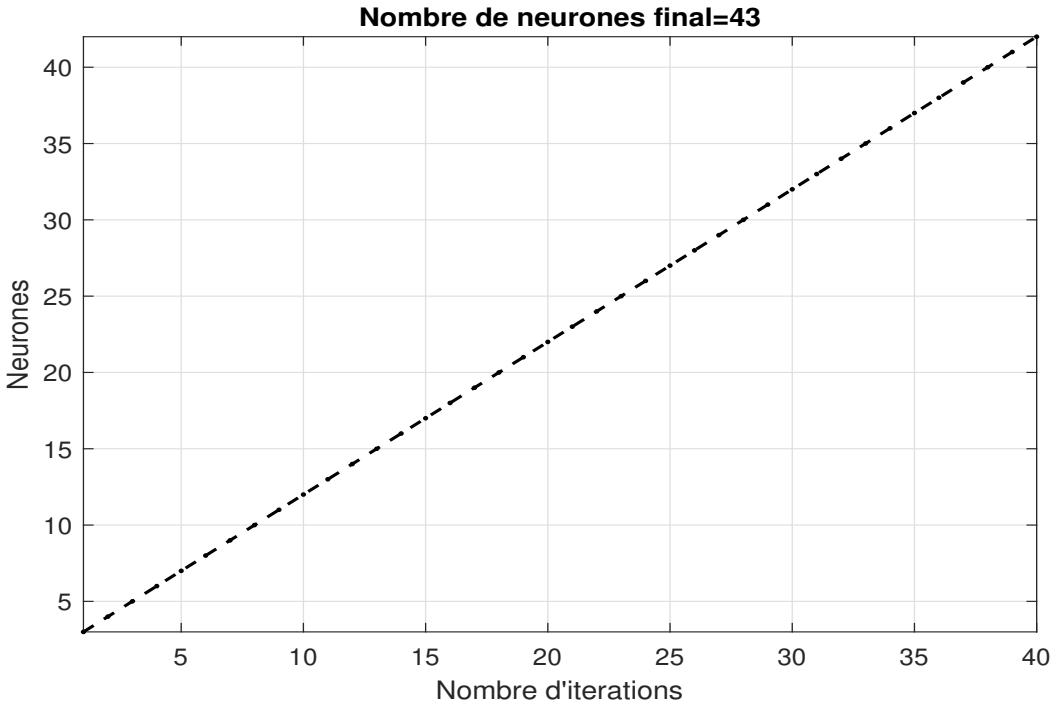


Figure 2.13 – Nombre de neurones de la couche cachée en fonction du nombre d'itérations pour 40 données

nombre exact de neurones sans distorsion d'informations et sans une quantité de neurones

excessive qui sera coûteuse en terme calculatoire. Les Figures, 2.14 et 2.15, représentent les deux horizons de commande et de prédiction et le niveau de précision obtenu avec 300 exemples d'apprentissage.

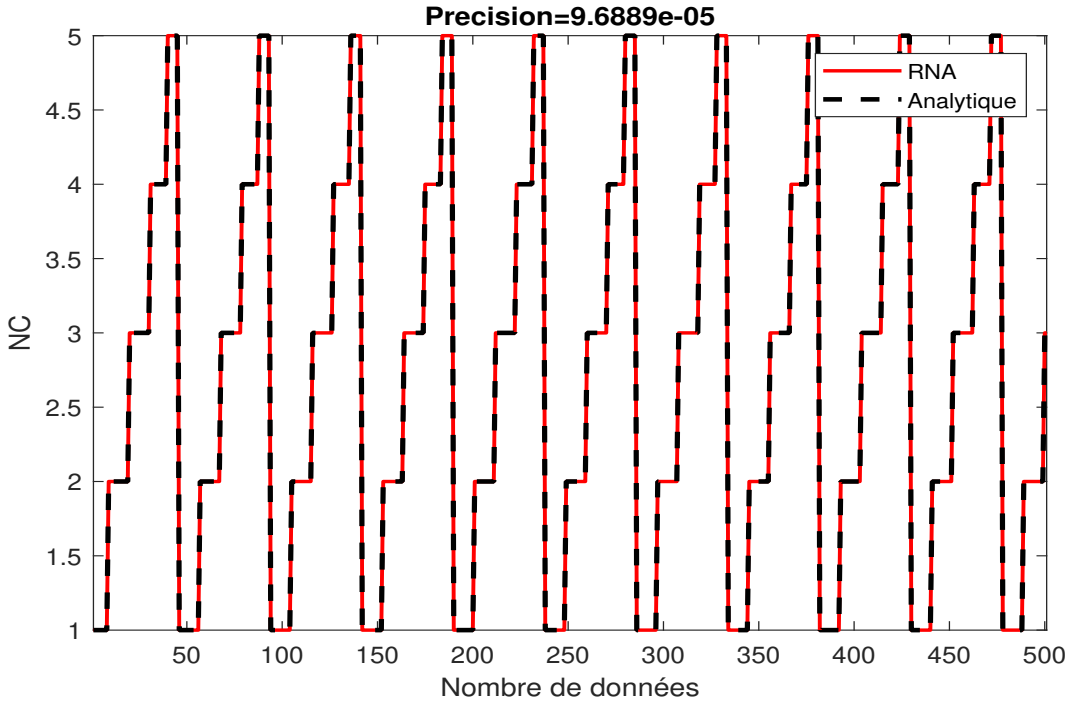


Figure 2.14 – Horizon de commande pour 300 données et 500 tests

A partir de ces figures, on constate que le réseau de neurones a pu finalement obtenir des résultats pertinents. Le RNA a réussi la prédiction de 500 données dans la phase de test qui ne sont pas présentes dans la base d'apprentissage et donc avec 300 éléments, nous avons un RNA synthétisé avec une meilleure généralisation. Les deux erreurs de prédiction sur les deux horizons sont de l'ordre de $3 * 10^{-3}$ et $1 * 10^{-3}$. Les courbes qui représentent les résultats obtenus par la méthode analytique et le RNA sont superposées avec une erreur quasiment nulle.

Les deux Figures 2.16 et 2.17 représentent respectivement l'erreur quadratique moyenne et le nombre de neurones obtenus pour une base d'apprentissage de 300.

Les deux Figures 2.16 et 2.17 montrent que la méthode proposée pour le dimensionnement automatique est efficace. Nous avons pu observer qu'à partir de 60 données le réseau de neurones avait déjà réussi à obtenir une meilleure généralisation sur les données avec une EQM inférieure à 10^{-3} (critère à ne pas dépasser pour rajouter des neurones).

A environ 170 données, l'algorithme a rajouté quatre nouveaux neurones. Cela est dû au fait que l'erreur quadratique a dépassé le seuil fixé à 10^{-3}.

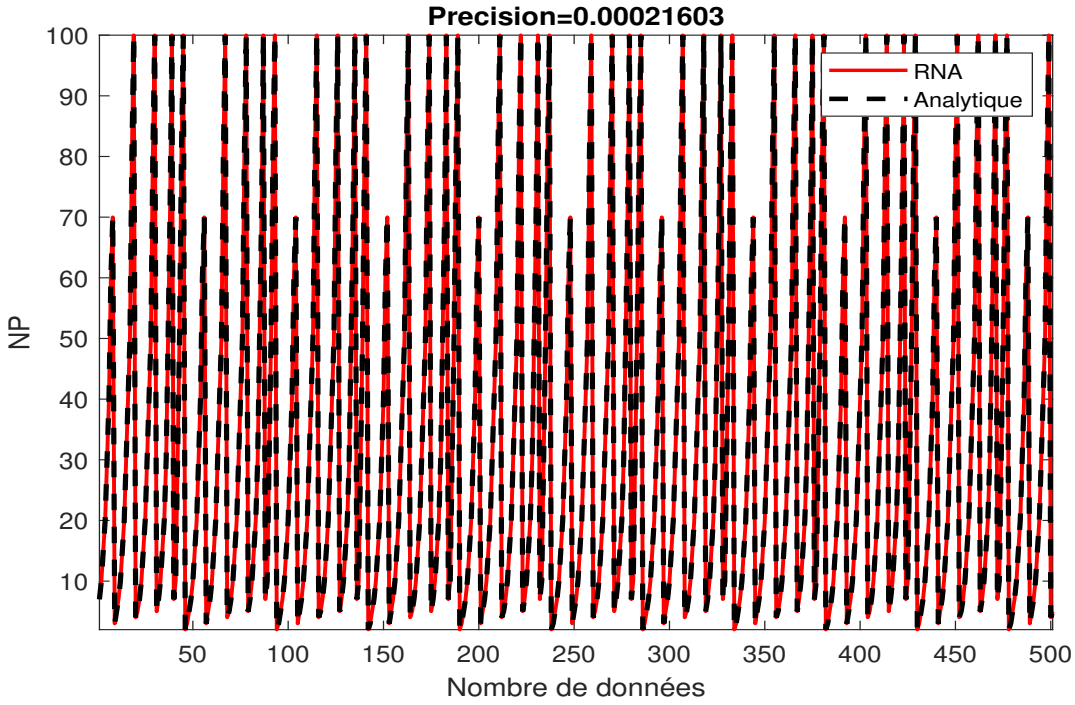


Figure 2.15 – Horizon de prédiction pour 300 données et 500 tests

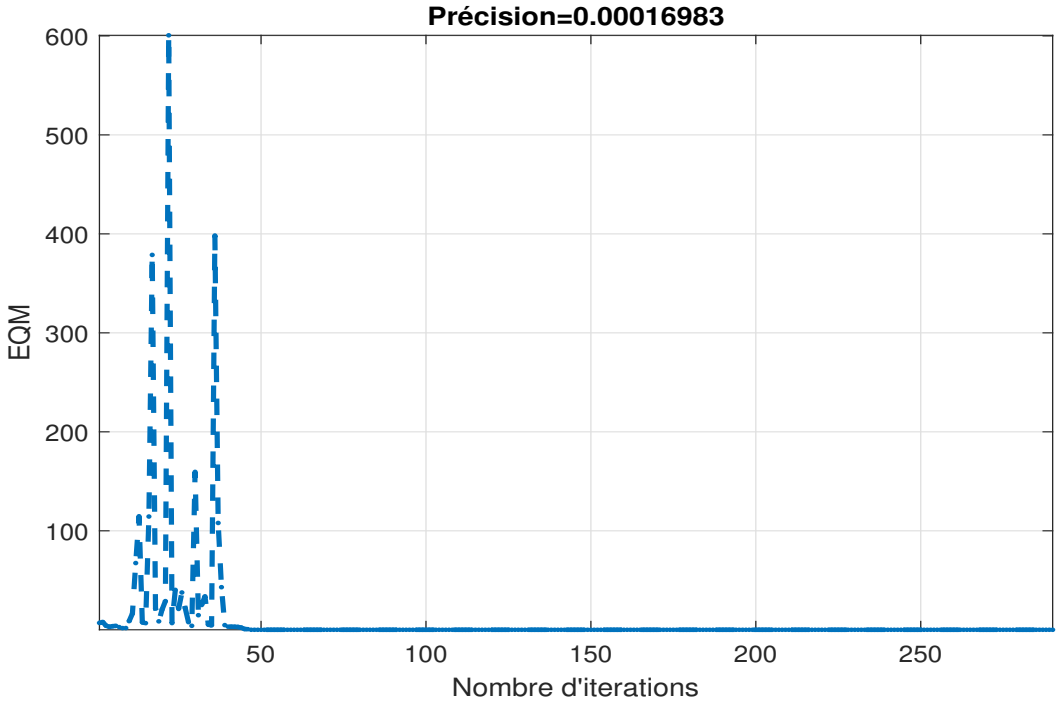


Figure 2.16 – Erreur quadratique moyenne pour 300 données

Afin de synthétiser les résultats obtenus, le tableau 2.1 indique les performances

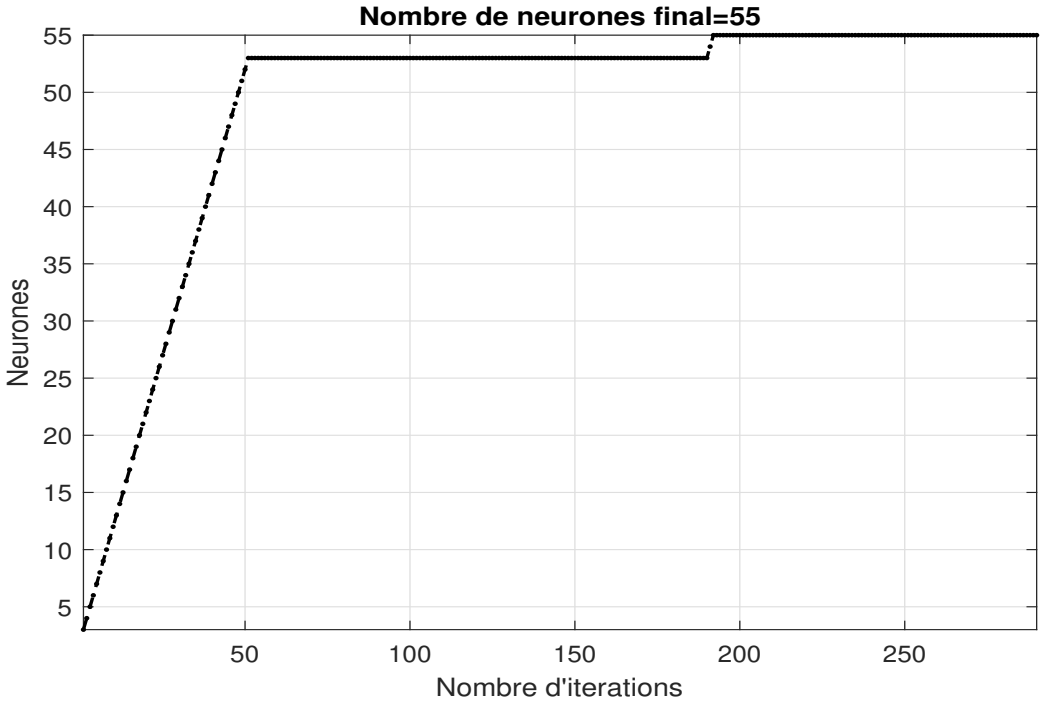


Figure 2.17 – Nombre de neurones de la couche cachée en fonction du nombre d'itérations pour 300 données

obtenues par l'approche analytique et celle basée sur les RNA avec en entrée les performances désirées (Indice De stabilité : IDS, le Temps de montée : TM, et l'Erreur de poursuite cumulée : ERR cum) et en sortie les deux paramètres que nous souhaitons calculer qui sont l'horizon de commande et l'horizon de prédiction.

Tableau 2.1 – Comparaison des performances

Performances			**Horizon de prédiction**		**Horizon de commande**	
IDS	TM	Err cml	RNA	Analytique	RNA	Analytique
0.1215	16.9652	6.6526	29.9998	30	1	1
0.1130	18.3346	6.9991	29.9998	40	1	1
0.1056	19.7006	7.3487	59.9996	60	1	1
0.1037	20.0601	7.4442	69.9998	70	1	1
0.1327	5.1732	5.3918	2.9986	3	2	2
0.2304	5.2804	4.7899	4.9987	5	2	2

Les résultats obtenus sont en adéquation avec les courbes présentées précédemment où les deux courbes (analytique et par RNA) sont superposées, ce qui prouve que la base de donnée d'apprentissage du réseau de neurones et le nombre de neurones obtenus par l'OS-ELM sont bien établis.

Les valeurs obtenues sont quasi similaires entre l'approche analytique et l'approche

neuronal avec une erreur inférieure à 10^{-3}.

2.7 Conclusion

Dans ce deuxième chapitre, la définition, le principe de fonctionnement et l'historique des ont été présentés. Ensuite, un état de l'art sur les approches d'apprentissage qui constituent une étape clé dans la formation d'un RNA a été détaillé.
Nous avons également dressé un état de l'art sur les approches de dimensionnement des RNA qui permettent de définir la structure optimale du réseau. L'objectif était d'identifier les différents problèmes causés par le RNA comme le sur-apprentissage et le sous-apprentissage.
Par ailleurs et dans ce contexte, une solution efficace qui permet de dimensionner automatiquement un RNA en utilisant une approche en ligne OS-ELM a été élaborée. L'OS-ELM est connu pour son temps de calcul assez faible par rapport aux autres algorithmes comme la rétro propagation. L'initialisation des paramètres du RNA est aléatoire. Ensuite, les poids synaptiques entre la couche cachée et la couche de sortie sont calculés automatiquement et d'une manière incrémentale.
Pour montrer l'efficacité de l'algorithme proposé, une application au calcul de deux paramètres de réglage de la MPC (horizons de commande et prédiction) a été effectuée. La comparaison est faite entre les résultats obtenus par l'approche analytique (pré-données) et les résultats obtenus par l'approche neuronale. On a pu constater que les résultats sont quasi-similaires avec une EQM inférieure à 10^{-3}, ce qui a permis de valider l'approche en montrant également l'influence de la taille de la base de données d'apprentissage et du nombre de neurones dans la couche cachée.

Dans le chapitre suivant nous présenterons en détails l'approche proposée pour le calcul des paramètres de réglage de la MPC.

Chapitre 3

Proposition d'une approche de réglage de la commande prédictive par RNA

Sommaire

3.1 Introduction 49

3.2 Formulation mathématique de la commande prédictive . . . 49

3.2.1 Représentation d'état augmentée à temps discret 49

3.2.2 Formulation de la commande prédictive à base d'état en absence de contraintes 51

3.2.3 Formulation mathématique de la commande prédictive à base d'état en présence de contraintes 54

3.2.3.1 Contraintes sur les entrées de commande 54

3.2.3.2 Contraintes sur les incréments de commande 55

3.2.3.3 Contraintes sur les sorties 56

3.2.3.4 Formulation des contraintes 57

3.3 Stratégie de réglage des paramètres de la MPC basée sur des RNA 57

3.3.1 Structure du RNA 57

3.3.1.1 Les entrées du RNA 58

3.3.1.2 Les sorties du RNA 58

3.3.2 Construction de la base de données d'apprentissage du RNA par PSO 58

3.3.2.1 Principe de formulation mathématique du PSO 59

3.3.2.2 Formulation du problème de réglage de la MPC par PSO 61

3.4 Application en simulation de l'approche proposée 62

3.4.0.1 Application numérique à un système SISO 64
3.4.0.2 Application numérique à un système SISO retardé . . 67
3.4.0.3 Application numérique sur un système MIMO retardé 70
3.5 Conclusion 74

3.1 Introduction

Dans le premier chapitre, la problématique liée au réglage de la commande prédictive et les motivations qui ont conduit à la synthèse de cette loi de commande optimale ont été exposées. Le deuxième chapitre a porté sur la présentation des réseaux de neurones artificiels qui constituent la piste que nous explorons comme outils de réglage de la MPC. Nous avons également proposé un algorithme qui permet, en plus de faire l'apprentissage, de déterminer la structure optimale du RNA en fonction de nombre de données stockées dans la base d'apprentissage du réseau.
Dans ce chapitre, une approche originale pour le réglage de la commande prédictive basée sur des RNA sera présentée. L'objectif est de montrer que cette approche permet d'assurer la stabilité optimale du système et qu'elle est applicable sur différents systèmes (SISO, MIMO) quelque soit leurs complexité.
Dans un premier temps, la formulation mathématique de la loi de commande à base d'état en absence de contraintes, puis avec contraintes sera exposée. Dans un deuxième temps, nous présenterons la stratégie de réglage basée sur les RNA. Plusieurs critères de performances seront ensuite considérés pour faire une étude comparative avec d'autre approches de réglage identifiées dans la littérature. Cette comparaison permettra de montrer l'intérêt de la méthode proposée.
Finalement, pour clôturer ce troisième chapitre, une application numérique de l'approche proposée sur différents systèmes SISO et MIMO avec et sans retard, sera présentée.

3.2 Formulation mathématique de la commande prédictive

Dans cette partie, la formulation mathématique à temps discret de la commande prédictive à base d'état avec et sans contraintes est présentée.

3.2.1 Représentation d'état augmentée à temps discret

Soit un système linéaire multivariable représenté par le modèle d'état à temps discret suivant :

$$\mathrm{x}_m(k+1) = A_m x_m(k) + B_m u(k)$$

(3.1)

$$y(k) = C_m x_m(k) \tag{3.2}$$

où $y \in \mathbb{R}^{n_{out}}$ est la sortie du système, $u \in \mathbb{R}^{n_{in}}$ le vecteur de commande et x est le vecteur d'état de dimensions $(n_1 \times 1)$. k est un entier naturel positif. Dans (3.1), A_m, B_m et C_m sont respectivement les matrices d'état, de commande et d'observation de dimensions respectives $(n_1 \times n_1)$, $(n_1 \times n_{in})$ et $(n_{out} \times n_1)$ avec :

— n_{in} : nombre d'entrées
— n_{out} : nombre de sorties
— n_1 : nombre d'états

Pour la suite, le modèle augmenté du système sera considéré. Ainsi, en considérant les différences :

$$\Delta x_m(k) = x_m(k) - x_m(k-1) \tag{3.3}$$

$$\Delta u(k) = u(k) - u(k-1) \tag{3.4}$$

l'équation 3.1 devient :

$$\begin{aligned} \Delta x_m(k+1) &= A_m(x_m(k) - x_m(k-1)) + B_m(u(k) - u(k-1)) \\ &= A_m \Delta x_m(k) + B_m \Delta u(k). \end{aligned} \tag{3.5}$$

Soit ainsi le nouveau vecteur d'état $x(k)$ défini par :

$$x(k) = \begin{bmatrix} \Delta x_m(k) \\ y(k) \end{bmatrix}. \tag{3.6}$$

Par ailleurs, l'équation 3.2 devient :

$$y(k+1) - y(k) = C_m(x_m(k+1) - x_m(k)) \tag{3.7}$$

$$= C_m \Delta x_m(k+1) \tag{3.8}$$

$$= C_m A_m \Delta x_m(k) + C_m B_m \Delta u(k) \tag{3.9}$$

Les équations (3.5) et (3.6) constituent la représentation d'état augmentée d'un modèle multivariable à temps discret :

$$x(k+1) = Ax(k) + B\Delta u(k) \tag{3.10}$$

$$y(k) = Cx(k) \tag{3.11}$$

avec :

$$\overbrace{\begin{bmatrix}\Delta x_m(k+1)\\ y(k+1)\end{bmatrix}}^{x(k+1)} = \overbrace{\begin{bmatrix}A_m & 0_m^T\\ C_mA_m & I_{(n_{out}\times n_{out})}\end{bmatrix}}^{A}\overbrace{\begin{bmatrix}\Delta x_m(k)\\ y(k)\end{bmatrix}}^{x(k)} + \overbrace{\begin{bmatrix}B_m\\ C_mB_m\end{bmatrix}}^{B}\Delta u(k) \tag{3.12}$$

$$y(k) = \overbrace{\begin{bmatrix}0_m & I_{(n_{out}\times n_{out})}\end{bmatrix}}^{C}\begin{bmatrix}\Delta x_m(k)\\ y(k)\end{bmatrix} \tag{3.13}$$

Par l'indice T nous désignons la transposée d'une matrice quelconque et I est la matrice identité. Dans (3.13), 0_m est une matrice nulle de dimension $n_{out} \times n_1$. Admettons que $n_{etat} = n_1 + n_{out}$, le triplet (A, B, C) de dimensions respectivement $n_{etat} \times n_{etat}$, $n_{etat} \times n_{in}$ et $n_{out} \times n_{etat}$ constitue le modèle augmenté discret invariant dans le temps du système MIMO considéré. L'intérêt du modèle d'état augmenté est d'intégrer l'action intégral dans la fonction coût.

3.2.2 Formulation de la commande prédictive à base d'état en absence de contraintes

Comme hypothèse, nous supposons que le système est observable et commandable. Le vecteur de l'incrément de commande ΔU de dimensions $(1 \times n_{in}N_c)$ est défini par :

$$\Delta U = \begin{bmatrix}\Delta u(k)^T & \Delta u(k+1)^T & \cdots & \Delta u(k+N_c-1)^T\end{bmatrix}^T. \tag{3.14}$$

Les vecteurs de la sortie désirée Y_{des} et de la sortie prédite $\hat{Y}$ sont définis comme suit :

$$Y_{des} = \begin{bmatrix}y_{des}(k+1)^T & y_{des}(k+2)^T & \cdots & y_{des}(k+N_p)^T\end{bmatrix}^T;$$

$$\hat{Y} = \begin{bmatrix}\hat{y}(k+1 \mid k)^T & \hat{y}(k+2 \mid k)^T & \cdots & \hat{y}(k+N_p \mid k)^T\end{bmatrix}^T.$$

La synthèse du contrôleur prédictif passe par une première étape de définition de la fonction coût J à minimiser :

$$J = (Y_{des} - \hat{Y})^T(Y_{des} - \hat{Y}) + \Delta U^T \bar{R} \Delta U \tag{3.15}$$

où $\bar{R}$ est une matrice carrée de dimensions $(n_{in}N_c \times n_{in}N_c)$ définie par :

$$\bar{R} = \begin{bmatrix} \lambda_1 & 0 & \cdots & 0 \\ 0 & \lambda_2 & \vdots & \vdots \\ \vdots & \cdots & \ddots & 0 \\ 0 & \cdots & 0 & \lambda_{n_{in}N_c} \end{bmatrix} \tag{3.16}$$

En se basant sur le modèle d'état augmenté (A, B, C), il est possible de calculer la séquence future du vecteur d'état selon la récurrence suivante [172] :

$$\begin{aligned} x(k+1 \mid k) &= Ax(k) + B\Delta u(k); \\ x(k+2 \mid k) &= Ax(k+1 \mid k) + B\Delta u(k+1); \\ &= A^2x(k) + AB\Delta u(k) + B\Delta u(k+1); \\ &\vdots \\ x(k+N_p \mid k) &= A^{N_p}x(k) + A^{N_p-1}B\Delta u(k) + A^{N_p-2}B\Delta u(k+1) \\ &\quad + \cdots + A^{N_p-N_c}B\Delta u(k+N_c-1). \end{aligned}$$

A partir du vecteur d'état prédit, en procédant par récurrence, la sortie estimée est alors :

$$\begin{aligned} y(k+1 \mid k) &= CAx(k) + CB\Delta u(k); \\ y(k+2 \mid k) &= CA^2x(k) + CAB\Delta u(k) + CB\Delta u(k+1); \\ y(k+3 \mid k) &= CA^3x(k) + CA^2B\Delta u(k) + CAB\Delta u(k+1) \\ &\quad + CB\Delta u(k+2); \\ &\vdots \\ y(k+N_p \mid k) &= CA^{N_p}x(k) + CA^{N_p-1}B\Delta u(k) + CA^{N_p-2}B\Delta u(k+1) \\ &\quad + \cdots + CA^{N_p-N_c}B\Delta u(k+N_c-1). \end{aligned}$$

La sortie prédite peut être exprimée alors sous la forme matricielle suivante :

$$\hat{Y} = Fx(k) + \Phi\Delta U \tag{3.17}$$

où F et Φ sont deux matrices de dimensions respectives $(n_{out}N_p \times n_{etat})$ et $(n_{out}N_p \times n_{in}N_c)$ définies par :

$$F = \begin{bmatrix} CA \\ CA^2 \\ CA^3 \\ \vdots \\ CA^{N_p} \end{bmatrix}; \tag{3.18}$$

$$\Phi = \begin{bmatrix} CB & 0 & 0 & ... & 0 \\ CAB & CB & 0 & ... & 0 \\ CA^2B & CAB & CB & ... & 0 \\ \vdots & & & & \\ CA^{N_p-1}B & CA^{N_p-2}B & CA^{N_p-3}B & ... & CA^{N_p-N_c}B \end{bmatrix}. \tag{3.19}$$

L'association des équations (3.15) et (3.17) permet d'obtenir la dérivée partielle de J par rapport à ΔU qui peut être écrite comme suit :

$$\frac{\partial J}{\partial \Delta U} = -2\Phi^T(Y_{des} - Fx(k)) + 2(\Phi^T\Phi + \bar{R})\Delta U \tag{3.20}$$

Afin d'obtenir la loi de commande optimale à appliquer au système, la condition d'optimalité suivante doit être respectée :

$$\frac{\partial J}{\partial \Delta U} = 0. \tag{3.21}$$

Dans le cas où la commande prédictive ne présente pas de contrainte, la séquence de commande optimale est alors exprimée comme suit :

$$\Delta U = (\Phi^T\Phi + \bar{R})^{-1}\Phi^T(Y_{des} - Fx(k)) \tag{3.22}$$

En appliquant le principe de l'horizon fuyant, seuls les premiers n_{in} éléments du signal de commande optimale sont appliqués au système :

$$u(k) = u(k-1) + \Delta u(k) \tag{3.23}$$

En remplaçant par 3.22 dans 3.23, on obtient :

$$u(k) = u(k-1) + \overbrace{\begin{bmatrix} I_m & 0_m & \cdots & 0_m \end{bmatrix}}^{n_{in}N_c}(\Phi^T\Phi + \bar{R})^{-1}\Phi^T(Y_{des} - Fx(k)). \tag{3.24}$$

Dans 3.24, le vecteur d'état $x(k)$ est requis par le contrôleur prédictif pour construire la sortie prédite. Lorsque ce vecteur n'est pas accessible, un observateur doit être utilisé pour l'estimer.

3.2.3 Formulation mathématique de la commande prédictive à base d'état en présence de contraintes

La présence des contraintes dans la plupart des systèmes dynamiques rend la synthèse des lois de commande plus compliquée surtout lors de l'application de ces commandes sur des systèmes industriels multi-variables non linéaires. Ces contraintes peuvent être de différentes natures. Dans le monde réel, deux types de contraintes sont distinguées : celles qui sont imposées par les limitations physiques et celles qui sont présentes afin de garantir des performances désirées [127].
La capacité à prendre en compte les contraintes est l'un des principaux atouts de la MPC. Nous allons intégrer ces différentes contraintes dans la formulation de la commande prédictive dans la partie qui suit.

3.2.3.1 Contraintes sur les entrées de commande

La première catégorie des contraintes que nous considérons est celle sur le signal de commande U définie par [172] :

$$U^{min} \leq U \leqslant U^{max} \tag{3.25}$$

où U^{min} et U^{max} représentent respectivement les valeurs minimale et maximale du signal de commande.

Chaque variable de commande est spécifiée par :

$$\begin{gathered} u_1^{min} \leq u_1 \leqslant u_1^{max} \\ u_2^{min} \leq u_2 \leqslant u_2^{max} \\ \vdots \\ u_{n_{in}}^{min} \leq u_{n_{in}} \leqslant u_{n_{in}}^{max} \end{gathered}$$

Ces contraintes sont généralement applicables aux instants qui suivent, et qui sont

exprimées par :

$$\overbrace{\begin{bmatrix} u(k) \\ u(k+1) \\ u(k+2) \\ \vdots \\ u(k+N_c-1) \end{bmatrix}}^{U} = \begin{bmatrix} I \\ I \\ I \\ \vdots \\ I \end{bmatrix} u(k-1) + \begin{bmatrix} I & 0 & 0 & \cdots & 0 \\ I & I & 0 & \cdots & 0 \\ I & I & I & \cdots & 0 \\ \vdots & & & & \\ I & I & \cdots & I & I \end{bmatrix} \overbrace{\begin{bmatrix} \Delta u(k) \\ \Delta u(k+1) \\ \Delta u(k+2) \\ \vdots \\ \Delta u(k+N_c-1) \end{bmatrix}}^{\Delta U} \tag{3.26}$$

3.2.3.2 Contraintes sur les incréments de commande

Les contraintes sur les incréments de commande sont généralement introduites pour vérifier la saturation du signal de commande face à des bornes de saturation. Elles sont dites aussi contraintes d'inégalité bien définies.
Soit la variable de commande incrémentale définie par ΔU soumise à des contraintes d'inégalité :

$$\Delta U^{min} \leq \Delta U \leqslant \Delta U^{max} \tag{3.27}$$

où ΔU^{min} et ΔU^{max} représentent respectivement les valeurs minimale et maximale de ΔU. Ces bornes sont définies dans le cas des systèmes MIMO par :

$$\Delta U^{min} = \begin{bmatrix} \Delta u_1^{min} & \Delta u_2^{min} & \cdots & \Delta u_{n_{in}}^{min} \end{bmatrix}$$
$$\Delta U^{max} = \begin{bmatrix} \Delta u_1^{max} & \Delta u_2^{max} & \cdots & \Delta u_{n_{in}}^{max} \end{bmatrix}$$

La définition des incréments des variables de commande est donnée par :

$$\Delta u_1^{min} \leq \Delta u_1 \leqslant \Delta u_1^{max}$$
$$\Delta u_2^{min} \leq \Delta u_2 \leqslant \Delta u_2^{max}$$
$$\vdots$$
$$\Delta u_{n_{in}}^{min} \leq \Delta u_{n_{in}} \leqslant \Delta u_{n_{in}}^{max}$$

Afin de pouvoir intégrer les contraintes de saturation dans la formulation mathématique de la MPC, il faut reformuler ces contraintes sous forme d'un problème d'optimisation. La résolution se fait par la suite en parallèle avec celle de la loi de commande.

La double inégalité (3.27) peut être mise sous la forme suivante :

$$-\Delta U \leqslant -\Delta U^{min} \tag{3.28}$$

$$\Delta U \leqslant \Delta U^{max} \tag{3.29}$$

Ce qui est équivalent également à l'écriture matricielle suivante :

$$\begin{bmatrix} -I \\ I \end{bmatrix} \Delta U \leq \begin{bmatrix} -\Delta U^{min} \\ \Delta U^{max} \end{bmatrix} \tag{3.30}$$

3.2.3.3 Contraintes sur les sorties

Dans cette partie, nous allons considérer les contraintes opérationnelles sur les sorties du système définies par :

$$Y^{min} \leq Y \leqslant Y^{max} \tag{3.31}$$

où Y^{min} et Y^{max} représentent respectivement les valeurs minimale et maximale acceptables pour Y. Ces deux contraintes d'inégalité sont définies dans le cadre des systèmes multivariables par :

$$Y^{min} = \begin{bmatrix} y_1^{min} & y_2^{min} & \cdots & y_{n_{out}}^{min} \end{bmatrix}$$

$$Y^{max} = \begin{bmatrix} y_1^{max} & y_2^{max} & \cdots & y_{n_{out}}^{max} \end{bmatrix}$$

Ces contraintes peuvent être formulées en fonction de ΔU :

$$Y^{min} \leq Fx(k) + \Phi\Delta U \leqslant Y^{max} \tag{3.32}$$

L'équation (3.32) peut alors être exprimée comme suit :

$$-\Phi\Delta U \leqslant -Y^{min} + Fx(k) \tag{3.33}$$

$$\Phi\Delta U \leqslant Y^{max} - Fx(k) \tag{3.34}$$

Les formules (3.33) et (3.34) peuvent être écrites sous la forme matricielle :

$$\begin{bmatrix} -\Phi \\ \Phi \end{bmatrix} \Delta U \leq \begin{bmatrix} -Y^{min} + Fx(k) \\ Y^{max} - Fx(k) \end{bmatrix}. \tag{3.35}$$

3.2.3.4 Formulation des contraintes

L'inégalité suivante est établie en combinant les équations (3.30), et (3.35) :

$$M_{cons}\Delta U \leq \gamma_{cons}, \tag{3.36}$$

M_{cons} et γ_{cons} sont définies par les matrices suivantes :

$$M_{cons} = \begin{bmatrix} -C_2 \\ C_2 \\ -I \\ I \\ -\Phi \\ \Phi \end{bmatrix}, \gamma_{cons} = \begin{bmatrix} -U^{min} + C_1 u(k-1) \\ U^{max} - C_1 u(k-1) \\ -\Delta U^{min} \\ \Delta U^{max} \\ -Y^{min} + Fx(k) \\ Y^{max} - Fx(k) \end{bmatrix}. \tag{3.37}$$

La prise en compte des contraintes peut être formulée comme étant un problème d'optimisation.

3.3 Stratégie de réglage des paramètres de la MPC basée sur des RNA

Dans cette partie, une explication détaillée de l'algorithme proposé pour le calcul des paramètres de réglage de la commande prédictive (λ, N_c, N_p) sera présentée, en se basant sur une approche complètement originale par RNA.
Dans un premier temps, la définition de la structure et de ses entrées et sorties du RNA sera détaillée.
Dans un deuxième temps, et afin de pouvoir utiliser correctement le RNA la problématique de construction d'une base d'apprentissage se pose automatiquement. Pour remédier à ce problème, la construction de cette base de données est faite avec l'utilisation d'une approche de réglage méta-heuristique basée sur un algorithme d'optimisation par essaim de particules dit "PSO" (Particle Swarm Optimisation).

3.3.1 Structure du RNA

Dans cette section, nous allons définir les entrées et les sorties du réseau de neurones qui sera considéré pour l'élaboration de cet algorithme.

3.3.1.1 Les entrées du RNA

En entrée du réseau de neurones, quatre paramètres qui constituent également les performances recherchées pour le système seront introduits :

- Le temps de montée (en anglais Rise Time ou RT) entre 10% et 90%.
- Le temps de réponse (en anglais Settling Time ou ST) qui est choisi inférieur à 2%
- Le dépassement (en anglais OVershoot ou OV). Ce critère présente le dépassement maximal sur les signaux de sortie.
- La moyenne de l'erreur de poursuite (en anglais The Average of The tracking Error ATE où MTE). Ce critère est utilisé pour évaluer la précision du système à chaque pas d'échantillonage.

3.3.1.2 Les sorties du RNA

L'objectif principal de nos travaux de recherche réside dans la détermination des paramètres de réglage de la MPC en se basant sur un RNA ; donc en sortie du réseau de neurones, nous retrouvons les trois paramètres de réglage :

- Horizon de commande ou de contrôle N_c,
- Horizon de prédiction N_p,
- facteur de pondération λ.

Au final, notre réseau de neurones se comportera comme une fonction mathématique avec des entrées définies et des sorties à calculer à partir de cette fonction. La structure du RNA est donc définie par le schéma de la Figure 3.1 :



Figure 3.1 – Structure du RNA

3.3.2 Construction de la base de données d'apprentissage du RNA par PSO

Pour pourvoir utiliser un RNA dans la synthèse d'un contrôleur prédictif robuste et à complexité réduite en vue d'une implémentation pratique, la construction d'une base de données d'apprentissage adéquate constitue notre principale préoccupation.
Cette partie est consacrée à la présentation d'une approche de réglage et d'optimisation,

qui permettra de construire cette base de données nécessaire au RNA. Nous nous intéressons à un algorithme méta-heuristique connu pour sa résolution des problèmes de commande reformulés comme des problèmes d'optimisation non linéaire sous contraintes. L'algorithme que nous proposons est basé sur une technique d'optimisation par essaim de particules PSO (Particle Swarm Optimisation). Cette technique facilitera le calcul des paramètres de réglage de la MPC qui seront alors considérés comme sorties de la base de données d'apprentissage.

3.3.2.1 Principe de formulation mathématique du PSO

L'algorithme PSO est un algorithme méta-heuristique à population de recherche globale proposé par [43] [55] [56] [146].
L'algorithme PSO est inspiré du comportement social des animaux évoluant en essaim comme les bancs de poissons et les groupes d'oiseaux. L'échange d'informations et la coopération entre les individus permettent la résolution de plusieurs problèmes d'optimisation complexes [68] [149]. Cette technique a suscité un intérêt remarquable dans divers domaines de l'ingénierie et de recherche récents, précisément celui du contrôle des processus industriels [32] [64].
Considérons un espace de recherche de dimension D, la i-ème particule de l'essaim de taille N_{pop}, est modélisée par un vecteur de position $x_n^i = (x_n^{i,1}, x_n^{i,2}, ..., x_n^{i,D})$ et un vecteur de vitesse $v_n^i = (v_n^{i,1}, v_n^{i,2}, ..., v_n^{i,D})$. En évaluant la fonction objectif à ce point, la qualité de la position pourra être connue.
Au début de l'algorithme, toutes les particules de l'essaim sont initialisées aléatoirement dans l'espace de recherche D. Puis, à chaque itération n, le vecteur de position et le vecteur de vitesse sont mis à jour respectivement à partir des deux équations de mouvement 3.38 et 3.39 [32] :

$$x_{n+1}^i = x_n^i + v_{n+1}^i \tag{3.38}$$

$$v_{n+1}^i = wv_n^i + c_1 r_{1,n}^i (p_n^i - x_n^i) + c_2 r_{2,n}^i (p_n^g - x_n^g) \tag{3.39}$$

où w représente le coefficient d'inertie, c_1 et c_2 respectivement les coefficients d'accélération cognitif et social, $r_{1,n}^i$ et $r_{2,n}^i$ deux nombres aléatoires de l'intervalle [0,1]. A chaque itération n, p_n^i représente la meilleure position obtenue de la i-ème particule et p_n^g la meilleure position trouvée dans l'ensemble de l'essaim.
A chaque déplacement, les nouvelles positions des particules sont évaluées par les deux équations 3.38 et 3.39. La Figure 3.2 représente la procédure graphique du mouvement d'une particule dans un espace de recherche à deux dimensions.

Afin d'éviter que des particules ne sortent de l'espace de recherche D défini, il faut

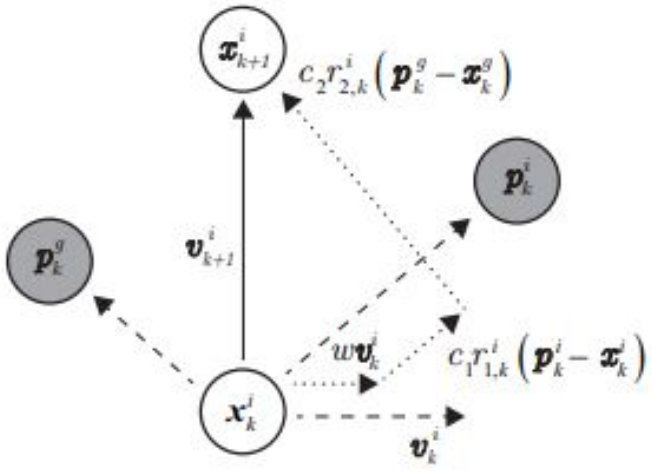


Figure 3.2 – Évolution de la position d'une particule dans l'espace

impérativement imposer des contraintes dites de confinement sur les variables de décision :

$$x^j_{min} \leq x^{i,j}_n \leq x^j_{max} \tag{3.40}$$

où $i \in [1, N_{pop}], j \in [1, D]$ et $n \in [1, N_{iter}]$ représentent respectivement les indices de la particule dans l'essaim, les composantes du vecteur des variables de décision et toutes les itérations de l'algorithme.

A ce propos, la technique la plus simple consiste à ne pas évaluer la nouvelle position de la particule. Néanmoins, il faut introduire dans l'algorithme PSO standard des mécanismes de confinement. Le principe consiste à dire que, si une particule sort de l'espace de recherche, il faut la ramener au point le plus proche dans cet espace et modifier ensuite sa vitesse.

D'un point de vue analytique, lorsqu'une particule de coordonnée $x^{i,j}_n$ du vecteur de position, sort de l'intervalle $[x^j_{min}, x^j_{max}]$, le mécanisme de confinement introduit consiste à lui attribuer la valeur du point de frontière le plus proche. Il convient de modifier les équations de mouvement précédentes en remplaçant, pour chaque particule, l'équation 3.38 par l'équation 3.41 :

$$x^{i,j}_n = min\{max(x_n i, j + v^{i,j}_n, x_{min^j}), x^j_{max}\} \tag{3.41}$$

La vitesse d'une telle particule $v^{i,j}_n$ est soit annulée soit remplacée par l'opposée pondérée défini par un poids μ, $\overline{v}^{i,j}_n$ comme présenter dans 3.42 :

$$\overline{v}^{i,j}_n = -\mu v^{i,j}_n, 0 \leq \mu \leq 1 \tag{3.42}$$

Au départ, les algorithmes PSO ont été formulés pour des problèmes d'optimisation sans contraintes. Les approches basées sur les techniques de pénalisation sont les plus adoptées [32]. Dans ce travail, nous utiliserons la technique de pénalité statique extérieure définie par :

$$\phi(x) = f(x) + \Sigma_{j=1}^{n_{con}} \wedge_j max[0, g_j(x)]^2 \tag{3.43}$$

où $\wedge_j$ et n_{con} représentent respectivement le coefficient de pénalité correspondant à la j-ème contrainte de type inégalité et le nombre total de ces contraintes, $f(.)$ la fonction coût du problème contraint et $g_j(.)$ les contraintes de type inégalité.
Afin de pouvoir implémenter cet algorithme par PSO en langage de programmation (MATLAB, C,...), nous adoptons l'algorithme 1 suivant :
où n représente l'itération courante et N_{iter} le nombre maximale d'itérations, w_{max} et

Algorithm 1 PSO

1: Définition des paramètres de contrôle de l'algorithme : N_{pop}, c_1, c_2 et N_{iter}
2: Initialisation aléatoire des N_{pop} positions x_0^i et vitesses v_0^i des particules dans l'essaim. Évaluation de cette population initiale et détermination de p_0^i et p_0^g
3: Incrémentation de l'indice des itérations n et application, pour chaque particule, des équations du mouvement 3.38 et 3.44.
4: Évaluation de la fonction coût à minimiser $\phi_n^i = \phi(x_n^i)$:
— si $\phi_n^i \leq lbest_n^i$ alors $lbest_n^i = \phi_n^i$ et $p_n^i = x_n^i$
— si $\phi_n^i \leq gbest_n^i$ alors $gbest_n^i = \phi_n^i$ et $p_n^g = x_n^i$
$lbest_n^i = \phi_n^i$ et $gbest_n^i = \phi_n^i$ représentant respectivement les meilleures performances de la i-ème particule et de l'ensemble de l'essaim.
5: Si le critère d'arrêt est satisfait (nombre max d'itérations ou autre). l'algorithme s'arrête avec la solution $x^* = argmin\{\phi(x_n^i), \forall i, n\}$. Sinon, retour à l'étape 3.

w_{min} sont respectivement les valeurs maximale et minimale du coefficient w.
Le facteur d'inertie w contrôle l'influence de la direction de la particule sur le déplacement futur. Les valeurs minimale et maximale du facteur d'inertie sont fixées dans [56] à $w_{min} = 0.4$ et $w_{max} = 0.9$. Ensuite, la nouvelle équation de mouvement liée à la vitesse devient :

$$v_{n+1}^i = wv_n^i + c_1 r_{1,n}^i (p_n^i - x_n^i) + c_2 r_{2,n}^i (p_n^g - x_n^g) \tag{3.44}$$

3.3.2.2 Formulation du problème de réglage de la MPC par PSO

L'objectif de cette partie est de formuler la problématique de réglage de la commande prédictive sous la forme d'un problème d'optimisation sous contraintes opérationnelles. La résolution de ce problème va se faire en se basant sur la PSO.
Le réglage des trois paramètres de la MPC (N_c, N_p, λ) est formulé comme étant un

problème d'optimisation défini comme suit [27] :

$$\begin{cases} Minimiser f(x,t); \\ x = N_p, N_c, \lambda; \\ sous - contraintes : \\ g_1(x) = N_c - N_p \leq 0 \\ g_2(x) = \lambda \geq 0 \end{cases}$$

où $f(.)$ est la fonction objectif du problème, $g_j(.)$ représente les contraintes d'inégalité, et $x = (N_p, N_c, \lambda) \in \mathbb{R}^m$ est le vecteur de décision.
La fonction coût du problème défini précédemment est donnée par l'équation 3.45 :

$$f_i(x,t) = \int_0^{+\infty} t(e_i(x,t) + u_i^2(x,t)dt \tag{3.45}$$

où $u_i(t)$ représente le signal de commande du contrôleur prédictif et $e_i(x,t)$ est l'écart entre la référence et la sortie du système à contrôler.
Pour pouvoir construire la base de données d'apprentissage nécessaire au RNA, nous utiliserons le PSO pour le calcul des paramètres de réglage. Pour chaque système, nous relèverons les performances obtenues en comparant la sortie réelle et la sortie désirée. Ces performances seront introduites en entrée du RNA et les paramètres de réglage en sortie. Cette procédure sera répétée pour toute classe de système (SISO, MIMO, avec et sans retard...).

3.4 Application en simulation de l'approche proposée

L'objectif de cette partie est de montrer l'efficacité de l'approche proposée basée sur un RNA en l'appliquant sur différentes classes de systèmes.
Dans cette section, trois applications numériques seront présentées : la première concerne le contrôle d'un système SISO du 2nd ordre, la deuxième un système SISO retardé, et finalement la troisième sur système multi-variable retardé.
Afin de pouvoir appliquer un RNA pour le calcul des paramètres de réglage et juste après la construction de la base de donnée d'apprentissage par PSO et l'utilisation de l'algorithme d'apprentissage en ligne par OS-ELM, la stratégie de commande proposée pour la simulation est donnée par la Figure 3.3 :
L'objectif est de montrer l'efficacité de l'approche basée sur un RNA. Dans ce sens, nous allons établir une étude comparative entre des approches de réglage de la MPC existantes dans la littérature et celle que nous proposons. Les critères de performances que nous avons choisi pour mener cette comparaison sont les suivants :

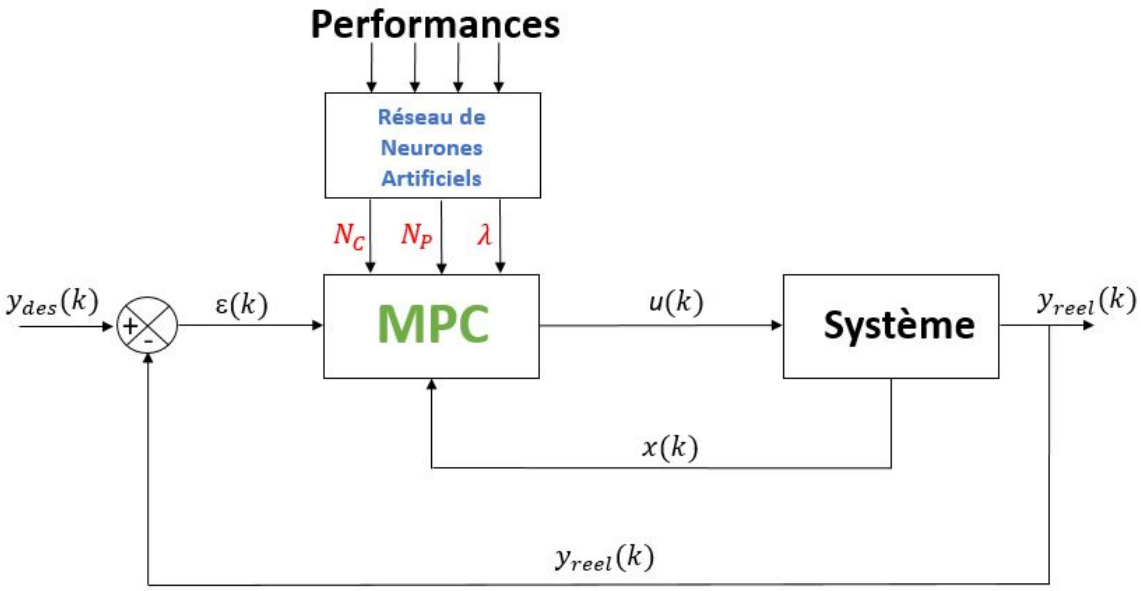


Figure 3.3 – Stratégie de réglage de la commande prédictive par RNA

1. **L'indice de stabilité (en anglais Stability Degree Index ou SDI) :** Ce critère permet d'évaluer la stabilité du procédé en boucle fermée (BF). Il est défini à partir de la différence entre le rayon du cercle unité et le module du pôle de la fonction de transfert le plus éloigné du centre du cercle [115] :

$$SDI(k) = 1 - max\Big(|p_1|, |p_2|, \cdots, |p_{A_N}| \Big), \tag{3.46}$$

où $\{p_1, p_2, \cdots, p_{A_N}\}$ représente les valeurs propres de la matrice d'état du système en boucle fermée A_{cl} définie par :

$$A_{cl} = A - BK_{mpc} \tag{3.47}$$

Le système sera stable en boucle fermée si est seulement si son SDI appartient à l'intervalle $]0, 1[$. Le système sera considéré en limite de stabilité si le SDI prend une valeur nulle. De même, il sera considéré instable si le SDI devient strictement négatif. .

2. **Le temps de montée (en anglais Rise Time ou RT) :** il est choisi entre 10% et 90%.
3. **Le temps de réponse (en anglais Settling Time ou ST) :** il est choisi inférieur à 2%.
4. **Le dépassement (en anglais OVershoot ou OV) :** Ce critère présente le dépassement maximal sur les signaux de sortie.

5. **La moyenne de l'erreur de poursuite ou MTE :** Ce critère est utilisé pour évaluer la précision du système à chaque pas d'échantillonnage.

3.4.0.1 Application numérique à un système SISO

— **Description du système**

Le système considéré est un moteur électrique avec une transmission flexible soumis à une inertie de charge [54].

Ce moteur électrique est modélisé par une fonction de transfert du second ordre donnée par l'équation 3.48 :

$$Z(s) = \frac{\Omega_M(s)}{V_M(s)} = \frac{K_n}{J_n s^2 + f_n s + C_n} \tag{3.48}$$

avec :

$$\begin{cases} K_n = & 2 & \text{Nm.rad/s/V;} \\ f_n = & 0.006 & \text{Nm.rad/s;} \\ J_n = & 0.002 & \text{kg.m}^2\text{;} \\ C_n = & 2 & \text{Nm.} \end{cases}$$

où V_M est l'entrée du système qui représente la tension (en V) appliquée au moteur. Ω_M la sortie du système est la vitesse angulaire du moteur (en rad/s).

La pulsation naturelle w_n et le facteur d'amortissement ξ de ce moteur sont donnés par :

$$w_n = \sqrt{\frac{C_n}{J_n}} = 31.623 \text{ rad/s}, \quad \xi = \frac{f_n}{2 w_n J_n} = 0.4741.$$

Avec une période d'échantillonnage $T_s = 5$ ms, la fonction de transfert du système (3.48) en temps discret est donnée par l'équation (3.49) :

$$G(z) = \frac{\Omega(z)}{V(z)} = \frac{0.01241 z + 0.11235}{z^2 - 1.96 z + 0.9851} \tag{3.49}$$

le modèle d'état discret équivalent du système est :

$$A_m = \begin{bmatrix} 1.9604 & -0.9851 \\ 1.000 & 0 \end{bmatrix}; B_m = \begin{bmatrix} 1 \\ 0 \end{bmatrix}; C_m = \begin{bmatrix} 0.0124 & 0.0123 \end{bmatrix}.$$

Le signal de commande du système est soumis à des contraintes d'inégalité linéaire :

$$U_{min} \leq U \leq U_{max}; \tag{3.50}$$

Avec :

$$\begin{cases} U_{min} = -5\text{V}; \\ U_{max} = +5\text{V}. \end{cases}$$

— **Résultats de simulation**

Afin de montrer la pertinence de l'approche par RNA, nous proposons de la comparer avec deux autres approches :

1. Approche de réglage heuristique des paramètres de la MPC [54] (Dumur 2000).
2. Approche de réglage analytique des paramètres de la MPC [166] (Turki 2017).

Les valeurs de paramètres de réglage déterminées selon les approche de [54], de [166] et de notre approche, ainsi que la période d'échantillonage utilisée sont regroupées dans le Tableau 3.1 :

Tableau 3.1 – Paramètres de la MPC utilisés en simulation pour le moteur électrique SISO

	$T_s(s)$	N_c	N_p	λ
Dumur2000	0.05	1	8	0.42
Turki 2017	0.05	2	10	0.02
Approche Proposée	0.05	4	39	5.00

Les résultats de simulation obtenus à partir de ces trois approches sont présentés dans les Figures 3.4 et 3.5. La Figure 3.4 représente la comparaison entre les sorties réelles des trois approches et la sortie désirée. La Figure 3.5 représente les trois signaux de commande obtenus pour chaque approche afin de commander le moteur. Le tableau 3.2 regroupant les performances obtenues permet de comparer ces trois approches :

L'examen des résultats observés sur les Figures 3.4 et 3.5 et le Tableau 3.2

Tableau 3.2 – Comparaison des performances de la MPC

	Dumur2000	**Turki 2017**	**Approche Proposée**
OV(%)	23	16	0.1
RT(s)	0.29	0.26	0.41
ST(s)	0.69	0.61	0.47
MTE	0.670	0.480	0.009
SDI	0.183	0.219	0.536

met en évidence la pertinence des performances obtenues par notre approche. En particulier, l'approche proposée permet d'obtenir les meilleures performances en termes de dépassement qui est quasiment nul. Ce critère est important dans les applications industriels car il peut causer un dysfonctionnement des actionneurs et

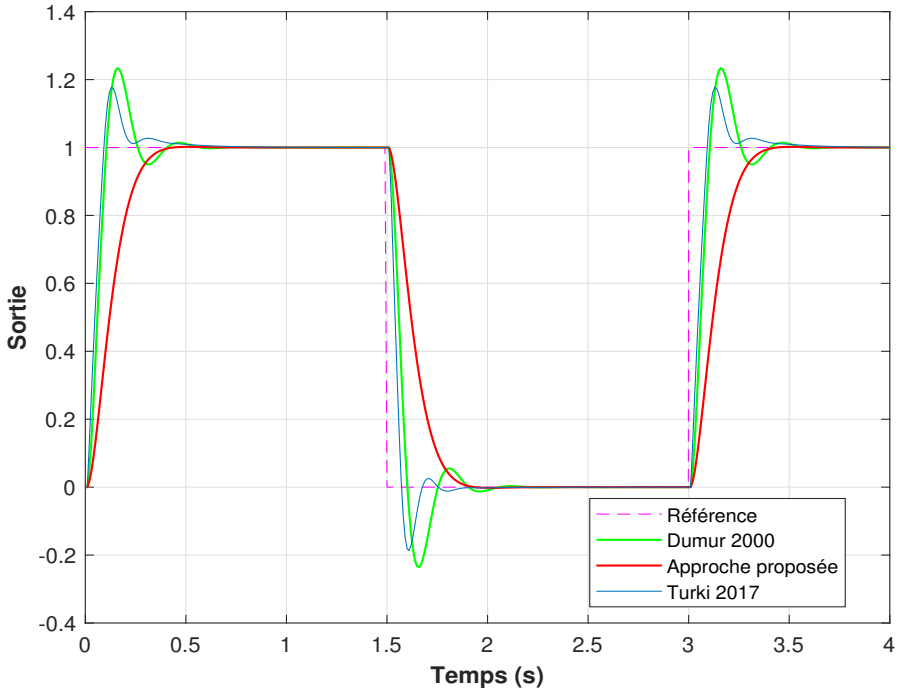


Figure 3.4 – Sortie du moteur en fonction du temps

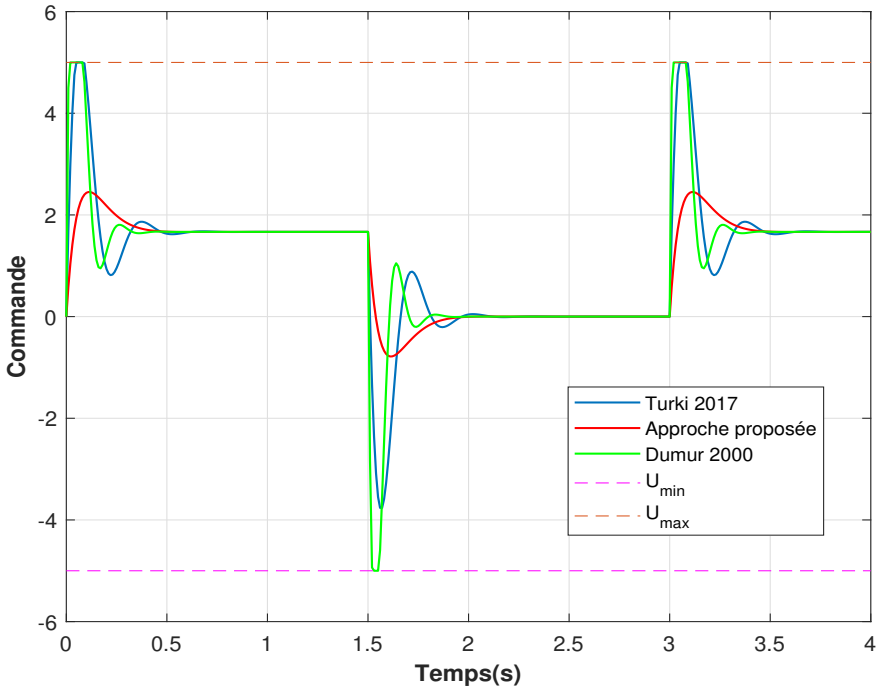


Figure 3.5 – Signaux de commande en fonction du temps

réduire leurs durée de vie.

En outre, la réponse la plus rapide est attribuée à notre approche, comme indiqué dans le tableau 3.2 où le temps de réponse le moins important est donné par

l'approche par RNA [80].

Il est important de noter que l'erreur de poursuite la plus faible est également obtenue par notre méthode. Ce qui permet de conclure que l'approche la plus précise est celle que nous proposons.

Nous avons effectué également une comparaison du SDI, qui permet d'évaluer le degré de stabilité des sorties en BF. En effet plus le SDI est proche de l'unité, plus le système est stable en BF. Le SDI que nous avons obtenu par notre approche est le plus proche de l'unité en le comparant avec les deux autres méthodes.

Finalement, l'approche proposée permet d'obtenir le meilleur compromis entre précision, stabilité et rapidité, ce qui permet de conclure que les meilleurs paramètres de réglage sont donné par l'approche basée sur les réseaux de neurones.

3.4.0.2 Application numérique à un système SISO retardé

Dans cette partie, l'approche proposée à base de RNA est appliquée en simulation pour contrôler un système du premier ordre avec retard FOPDT (First Order Plus Dead Time). L'objectif dans cette partie est de montrer qu'un bon réglage des paramètres du contrôleur prédictif permet un rejet du retard sans avoir à reconstruire une autre base d'apprentissage mais seulement en se réfèrant à la base de données déjà construite pour les systèmes SISO.

Les performances obtenues seront comparées avec celles de deux autres approches pour montrer l'efficacité de la MPC réglé par le RNA.

— **Description du système**

Plusieurs systèmes dans la littérature peuvent être approximés par des systèmes avec retard [31]. Le système que nous avons considéré est modélisé par la fonction de transfert définie par [142] :

$$G(s) = \frac{1}{1 + 0.5s}\, e^{-s} \tag{3.51}$$

Nous avons considéré également pour ce système que le signal de commande est soumis aux contraintes d'inégalité suivantes :

$$\begin{cases} U_{min} = 0\text{V}; \\ U_{max} = 2.5\text{V}. \end{cases}$$

— **Résultats de simulation**

Une comparaison entre l'approche neuronale et deux autres approches recensées qui lient le réglage de la commande prédictive et les systèmes retardés sera présenté

dans cette partie [148] et [91].

Chacune de ces approches de réglage propose une méthode différente de calcul des paramètres de la MPC :

1. Approche de réglage analytique des paramètres de la MPC (Schridar et Cooper 1996) [148] .
2. Approche de réglage heuristique des paramètres de la MPC à base d'ANOVA (Iglesias et Sunjuan 2016) [91] .

Les méthodes de calcul des paramètres de réglage proposées par (Schridar et Cooper 1996) et (Iglesias et Sunjuan 2016) sont données dans le tableau 3.3 :

$$G_{fopdt}(s) = \frac{K_{fopdt}e^{-\theta_{fopdt}s}}{\tau_{fopdt}s + 1} \tag{3.52}$$

Tableau 3.3 – Paramètres de réglage de la MPC utilisés en simulation

	(Schridar et Cooper 1996) [148]	**(Iglesias et Sunjuan 2016) [91]**
N_c	$\in \mathbb{N}$, de 1 à 6	$\in \mathbb{N}$, de 1 à 6
N_p	$\frac{5\tau_{fopdt}}{T_s} + \frac{\theta_{fopdt}}{T_s} + 1$	$\frac{5\tau_{fopdt}}{T_s} + \frac{\theta_{fopdt}}{T_s} + 1$
λ	fK^2_{fopdt}	$1.631K_{fopdt}(\frac{\theta_{fopdt}}{\tau_{fopdt}})^{0.4094}$

Les valeurs des paramètres de réglage déterminées selon les approche de [148], de [91] et l'approche que nous proposons ainsi que le temps d'échantillonage pour chaque approche sont regroupées dans le Tableau 3.6 :

Tableau 3.4 – Paramètres de la MPC utilisés en simulation pour un système SISO retardé

	$T_s(s)$	N_c	N_p	λ
Schridar et Cooper 1996	0.05	3	61	1.134
Iglesias et Sunjuan 2016	0.05	3	61	2.166
Approche Proposée	0.05	4	39	5.001

Les résultats de simulation obtenus sont présentés dans les Figures 3.6 et 3.7 [81]. La Figure 3.6 représente les signaux de sorties réelles des trois approches et la référence.

La Figure 3.7 montre les trois signaux de commandes et les contraintes d'inégalité définies précédemment.

A des fins de comparaison, le Tableau 3.5 regroupe les performances obtenues selon les trois approches. Les critères considérés restent identiques au cas précédent :

Au regard de l'allure des réponses représentées par les Figure 3.6, 3.7 et des performances indiquées dans le Tableau 3.5, on arrive à observer clairement

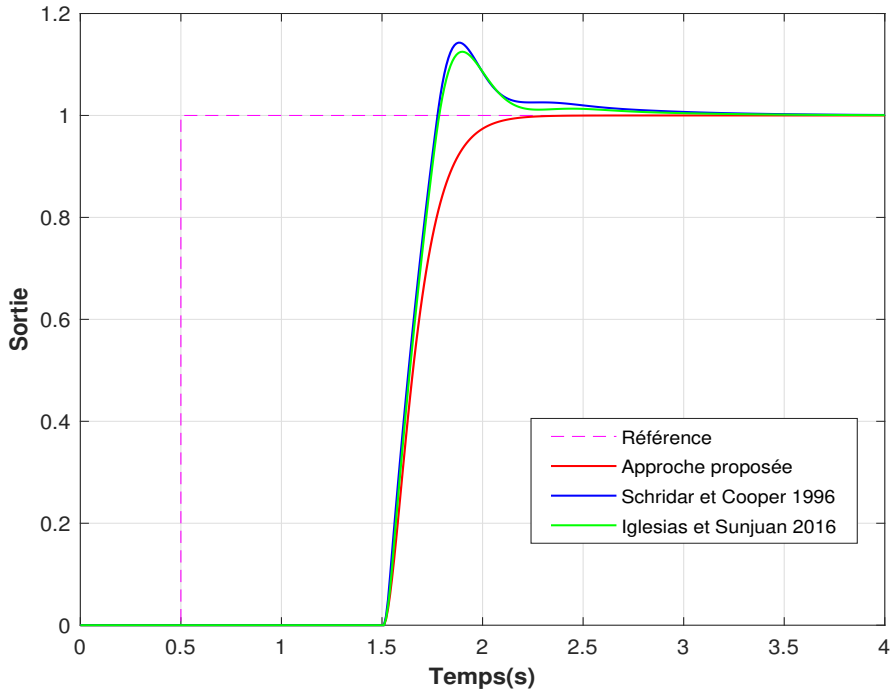


Figure 3.6 – Sortie du système retardé en fonction du temps

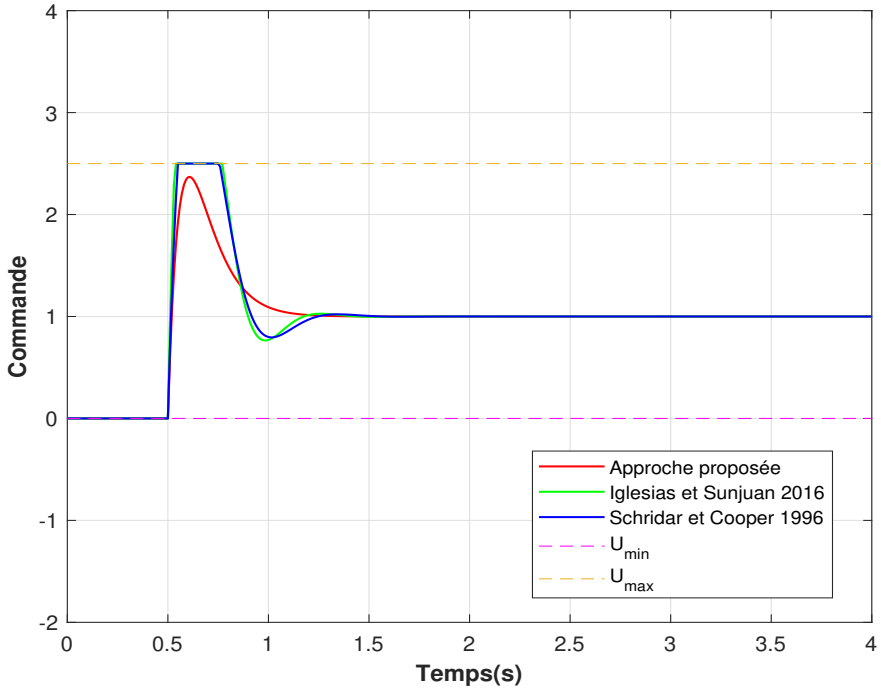


Figure 3.7 – Signaux de commande en fonction du temps

l'efficacité de l'approche proposée par rapport aux deux autres approches fournies par la littérature. En effet, on remarque que le temps de réponse le moins élevé est obtenu avec notre approche, ce qui permet de conclure sur la rapidité à atteindre

Tableau 3.5 – Comparaison des performances de la MPC

	Schridar et Cooper 1996	**Iglesias et Sunjuan 2016**	**Approche proposée**
OV(%)	11.79	10.55	0.5
RT(s)	1.613	1.654	1.711
ST(s)	2.986	2.762	2.252
MTE	0.192	0.210	0.018
SDI	0.149	0.138	0.498

la valeur finale en régime permanent.

En termes de précision des signaux de sorties, les trois approches mènent à une erreur de poursuite faible mais celle que nous proposons permet d'aller plus loin avec une erreur de poursuite quasi nulle. Cette constatation est validée également par un dépassement dix fois plus faible avec l'approche neuronale par rapport aux deux approches, heuristique et analytique.

La comparaison des SDI montre que le degré de stabilité le plus élevé est obtenu avec notre approche et aussi plus proche de l'unité en le comparant aux deux indices obtenus selon les deux autres méthodes.

Il est important de noter également qu'en allant plus loin en termes de contraintes sur les signaux de commande l'approche que nous proposons évite la saturation alors que les deux autres approches saturent avec une contrainte supérieure $U_{max} = 2.5$, ce qui nous permet de dire que la réponse la plus souple est obtenue par la méthode que nous proposons.

On peut donc conclure que le réglage paramétrique de la commande prédictive par RNA permet d'obtenir le meilleur compromis entre précision, stabilité et rapidité.

3.4.0.3 Application numérique sur un système MIMO retardé

Dans la partie précédente, nous avons démontré l'efficacité de l'approche par RNA sur les systèmes mono-variables (SISO non retardé et SISO retardé). L'objectif dans cette partie est de montrer que l'approche que nous proposons est applicable sur n'importe quel système y compris les systèmes multivariables qu'ils soient à retard ou non.

Afin de pouvoir appliquer la méthode sur les systèmes multivariables, nous avons construit une nouvelle base de données d'apprentissage des systèmes MIMO avec deux entrées et deux sorties. La taille de cette base d'apprentissage est la même que celle utilisée pour les systèmes SISO.

Ensuite, pour conclure sur la pertinence de l'approche neuronale, nous allons faire une comparaison avec une autre approche issue de la littérature et qui traite la problématique du réglage de la commande prédictive des systèmes MIMO.

— **Description du système**

Le système considéré est un système multivariable avec deux entrées et deux sorties modélisé par une fonction de transfert sous la forme matricielle définie par l'équation 3.53 :

$$G(s) = \begin{bmatrix} \frac{1.5e^{-2s}}{3s+1} & \frac{0.6e^{-3s}}{2s+1} \\ \frac{0.6e^{-3s}}{2s+1} & \frac{1.5e^{-2s}}{3s+1} \end{bmatrix} \tag{3.53}$$

— **Résultats de simulation**

Afin de montrer la pertinence de la méthode proposée, nous avons décidé d'effectuer une comparaison avec une approche de la littérature qui lie le réglage paramétrique et les systèmes MIMO retardés.

Cette approche se base sur un calcul heuristique des paramètres du contrôleur prédictif proposée par (Ebrahimi et al,) est sera notée par la suite "(Ebrahimi 2015)" [57].

Les valeurs des paramètres de réglage de la MPC utilisées en simulation ainsi que le temps d'échantillonage considéré pour chaque approche dans la comparaison sont représentés dans le Tableau 3.6 :

Tableau 3.6 – Paramètres de réglage de la MPC pour le système (3.53)

	$T_s(s)$	N_c	N_p	λ_1	λ_2
Ebrahimi 2015	0.5	13	15	1.690	0.830
Approche proposée	0.5	3	19	4.236	3.951

Les résultats de simulation obtenus sont présentés dans les Figures 3.8, 3.9, 3.10 et 3.11.

La Figure 3.8 représente la première sortie y_1 du système selon les deux approches en fonction du temps. La Figure 3.9 affiche la seconde sortie y_2 selon les deux approches toujours en fonction du temps. Dans les deux Figures 3.10 et 3.11 sont représentées respectivement l'évolution des signaux de commande de la première entrée et la seconde entrée selon les deux approches. Les deux tableaux 3.7 et 3.8 représentent respectivement les performances obtenues pour les sorties y_1 et y_2 :

En observant les résultats de simulation et les performances obtenues, on peut conclure que :

1. En termes de précision sur les sorties réelles y_1 et y_2, par rapport à la référence on remarque que l'erreur de poursuite la moins élevée est obtenue selon notre approche. On obtient également un dépassement quatre fois plus faible que celui obtenu par l'approche de (Ebrahimi et al,) [57]. Le dépassement est quasi nul

Tableau 3.7 – Performances de la sortie y_1

	Ebrahimi 2015	**Approche proposée**
OV(%)	2.122	0.443
RT(s)	226.36	227.02
ST(s)	233.20	229.15
MTE	0.079	0.076
SDI	0.3298	0.4620

Tableau 3.8 – Performances de la sortie y_2

	Ebrahimi 2015	**Approche proposée**
OV(%)	2.5896	0.9780
RT(s)	53.226	55.051
ST(s)	70.308	65.741
MTE	0.079	0.076
SDI	0.3298	0.4620

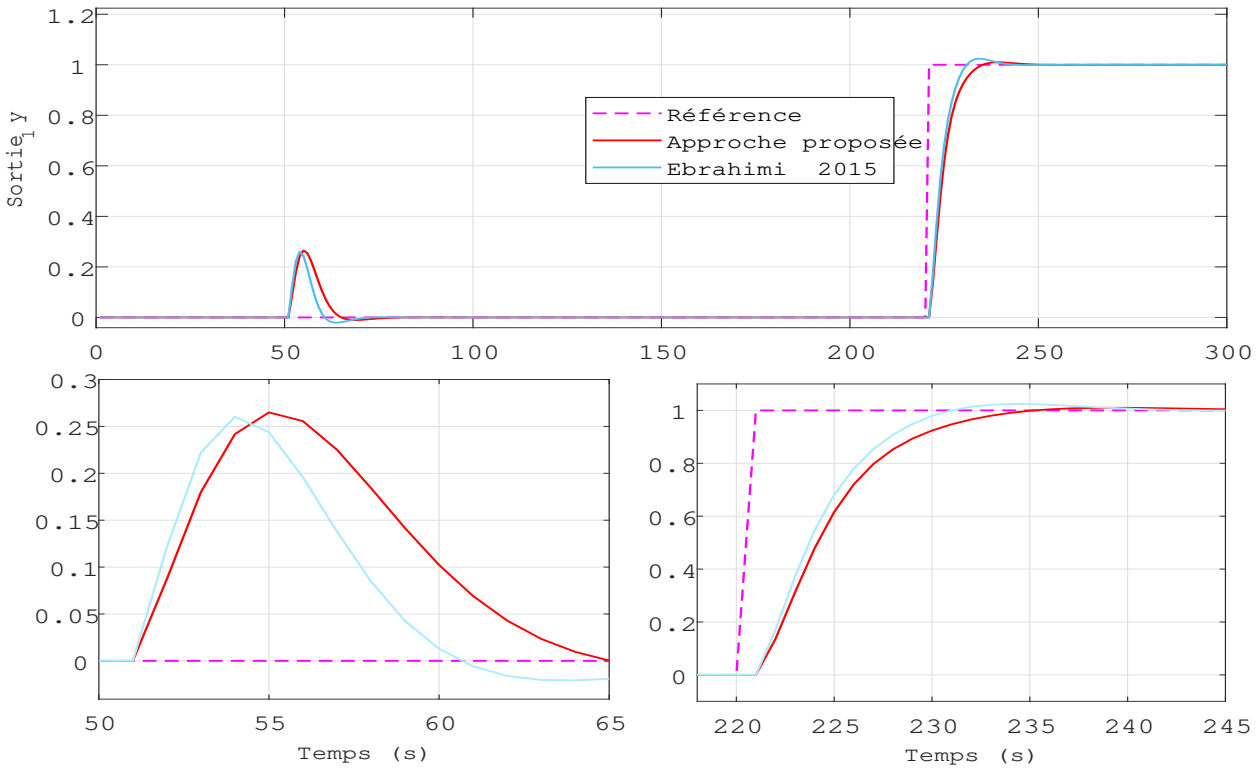


Figure 3.8 – Sortie y_1 en fonction du temps

pour la méthode que nous proposons.

2. Concernant l'évaluation de la stabilité, qui est caractérisée par le SDI, on a pu remarquer en effectuant une comparaison des deux sorties que le SDI le plus élevé et le plus proche de l'unité est obtenu par notre approche, ce qui assure la stabilité en boucle fermée.

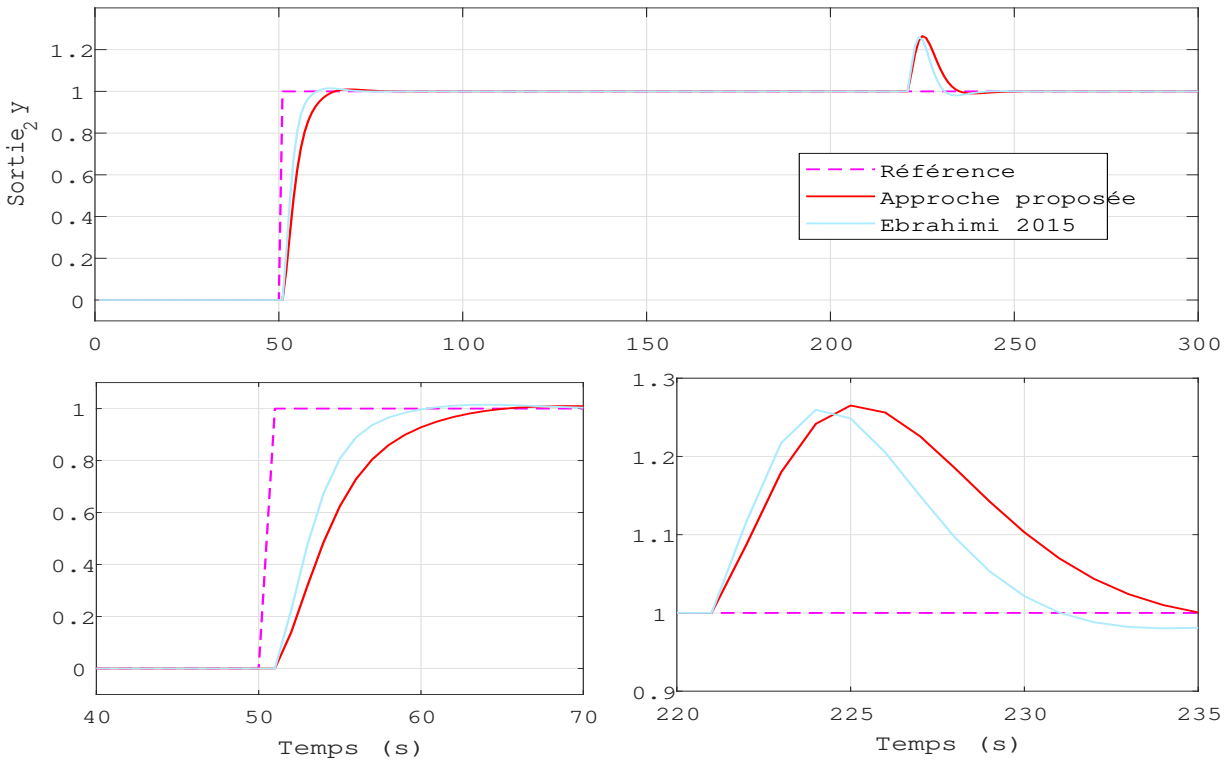


Figure 3.9 – Sortie y_2 en fonction du temps

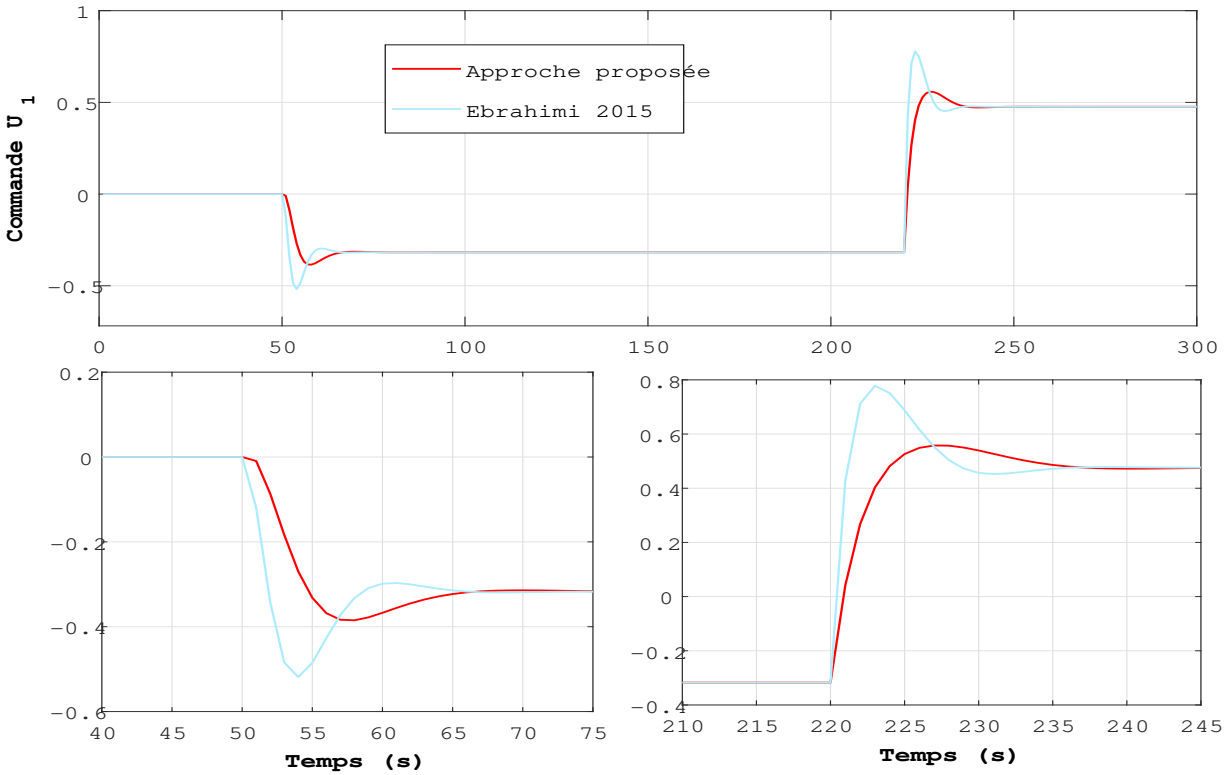


Figure 3.10 – Signal de commande u_1 en fonction du temps

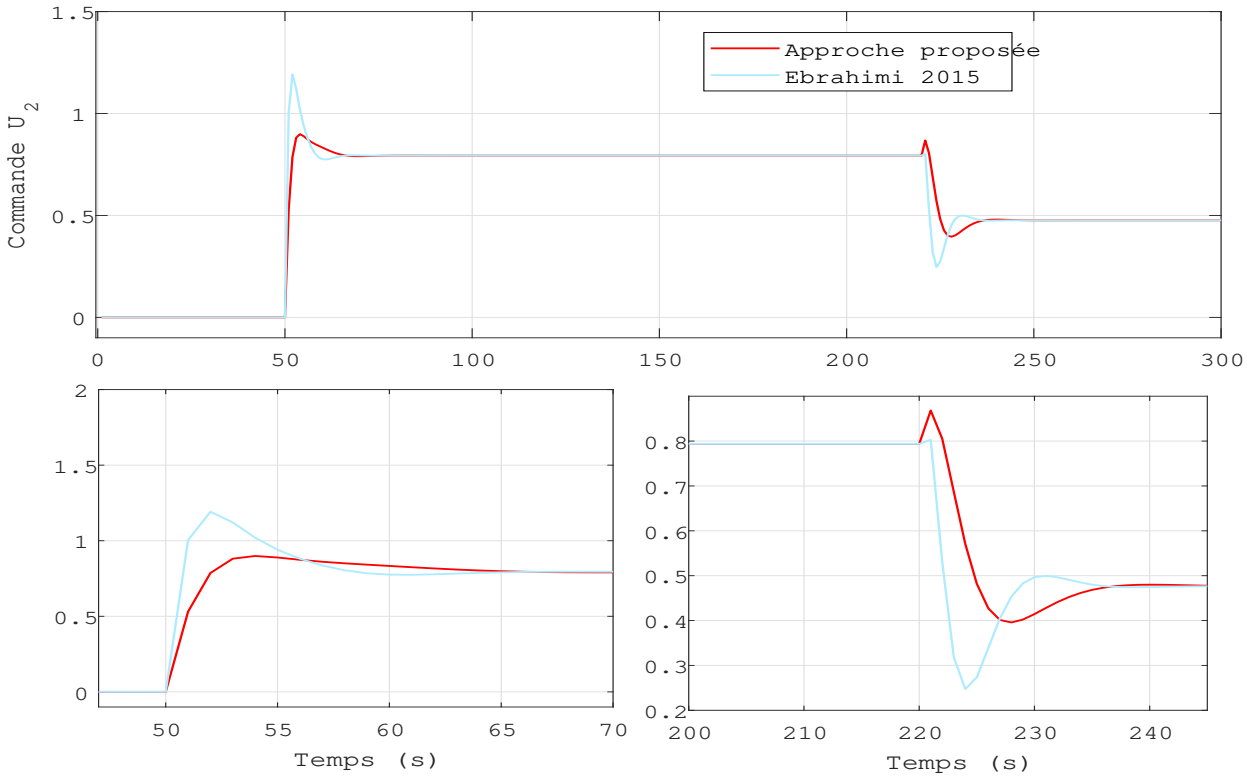


Figure 3.11 – Signal de commande u_2 en fonction du temps

3. L'observation des résultats obtenus en termes de temps de montée et temps de réponse, nous a permis de constater que la réponse la plus rapide et plus souple est obtenue selon notre approche. En effet, le temps de réponse pour atteindre la valeur finale en régime permanent est nettement moins élevé pour l'approche neuronale par comparaison avec celui obtenu par l'approche heuristique [57].
4. Finalement l'évolution des deux signaux de commande 3.10 et 3.11, permet de conclure que les signaux de commande les plus fluides sont attribués à notre approche. Cette dernière permet d'avoir un pic moins élevé par rapport à celui obtenu par (Ebrahimi et al,).

Par conséquent, nous pourrons conclure qu'avec l'approche que nous proposons on a réussi à assurer les meilleures performances en termes de stabilité, précision et rapidité.

3.5 Conclusion

Le travail effectué dans ce chapitre a porté sur la proposition d'une approche de réglage originale par réseaux de neurones artificiels pour le calcul des paramètres de la commande prédictive.

Nous avons commencé par rappeler la formulation mathématique à base d'état de la

commande prédictive en se basant sur le modèle d'état augmenté du système dans les deux cas, avec et sans contraintes.
Ensuite, nous avons défini la structure du RNA utilisé pour le calcul des paramètres de réglage en introduisant les entrées et sorties du réseau.
Après avoir établi la structure du RNA, nous avons présenté la stratégie de réglage proposée. A suivi une étude comparative entre notre méthode différents approches identifiées dans la littérature.
Cette étude a permis de montrer, d'une part que l'approche proposée est applicable à différentes catégories de systèmes (SISO avec ou sans retard, MIMO avec ou sans retard), et que d'autre part, avec la combinaison (N_c, N_p, λ) nous arrivons à obtenir le meilleur compromis entre stabilité, rapidité et précision.

Dans le prochain chapitre, la problématique de la robustesse en performances de l'approche proposée en présence de perturbations sera abordée, et ensuite comparée avec des approches de réglage de la commande prédictive robustifiées. L'objectif sera donc de montrer qu'un bon réglage paramétrique de la MPC permet de rejeter les perturbations sans avoir recours à des outils de robustificiation.

Chapitre 4

Robustesse en performances de la commande prédictive

Sommaire

4.1 Introduction 77

4.2 État de l'art sur les approches de robustification de la commande prédictive 77

4.2.1 Approches basées sur la représentation polynomiale 78

4.2.2 Approches basées sur la représentation d'état 85

4.2.3 Synthèse comparative des approches de robustification de la commande prédictive 89

4.3 Influence des paramètres de réglage sur la robustesse de la commande prédictive 95

4.3.1 Comparaison entre la MPC réglée par RNA et la GPC non robustifiées 95

4.3.2 Comparaison entre la MPC réglée par RNA et la GPC avec robustification 97

4.3.3 Comparaison entre la MPC réglée par RNA avec et sans robustification 99

4.4 Conclusion 101

4.1 Introduction

La synthèse d'une commande prédictive se base principalement sur le modèle du système à contrôler. Toutefois, la difficulté vient du fait que souvent il n'existe pas de modèle précis du processus réel, dû à de multiples raisons : informations incomplètes sur le processus, caractéristiques variant avec le temps ....
C'est pour cette raison que la synthèse des contrôleurs se fait avec des modèles incomplets, et ensuite ces contrôleurs doivent être insensibles aux incertitudes du système réel et aux perturbations qui s'appliquent sur ce dernier. En d'autre termes, il est préférable que le régulateur soit robuste. Un contrôleur sera dit robuste s'il maintient ses propriétés (stabilité et performances) en présence d'incertitudes ou de perturbations internes ou externes.
Il existe principalement deux manières pour satisfaire ces demandes : d'une part, la conception d'une commande adaptative ayant la capacité de modifier ses paramètres de réglages au cours du temps (en ligne), et d'autre part, par la conception d'un contrôleur robuste qui permette le rejet de ces perturbations.

L'objectif de ce chapitre est de tester la robustesse en performances de la commande prédictive en présence de perturbations extérieures et de bruit de mesure. Dans un premier temps, un état de l'art sur les approches de robustification de la MPC sera exposé, ces dernières ayant pour objectif de robustifier le contrôleur sans changement du comportement entré/sortie.
Ensuite, dans un deuxième temps, nous appliquerons ces stratégies de robustification sur des systèmes LTI (Linear Time Invariant) SISO.
Dans un troisième et dernier temps, nous appliquerons la commande prédictive réglée par RNA sur des systèmes SISO perturbés afin de montrer sa pertinence et qu'une combinaison optimale des paramètres de réglage permet le rejet de perturbations sans avoir recours à des outils de robustification lourd en terme calculatoire.

4.2 État de l'art sur les approches de robustification de la commande prédictive

Dans cette section, nous présentons un état de l'art sur les approches de robustification de la commande prédictive. Dans la littérature, les approches de robustification sont regroupées selon deux grands for esmalismes : le formalisme basé sur la représentation polynomiale (fonctions de transfert) et le formalisme basé sur la représentation d'état.

4.2.1 Approches basées sur la représentation polynomiale

La généralisation du formalisme polynomial a été proposée par (D.W Clarke et al,) en 1987. L'objectif était de développer une nouvelle version de la commande appelée « Commande Prédictive Généralisée » dite GPC (Generalized Predictive Control). Cette appellation est née d'une collaboration avec de grands groupes industriels aux États-Unis et en Europe [45].
La spécificité de la commande prédictive généralisée est de se baser sur le développement d'un modèle mathématique linéaire.
Le modèle le plus souvent utilisé, dans la pratique, est celui dit CARIMA (Controlled Auto Regressive Integrated Moving Average). Ce dernier assure la prédiction du comportement futur de la dynamique réelle du système. C'est une classe de contrôleurs prédictifs à deux degrés de liberté [1].
Le modèle CARIMA est donné par l'équation 4.1 :

$$A(q^{-1})y(t) = B(q^{-1})u(t-1) + \frac{C(q^{-1})}{\Delta(q^{-1})}\xi(t) \tag{4.1}$$

Où $u(t)$, $y(t)$ et $\xi(t)$ représentent respectivement l'entrée, la sortie et la perturbation. $\xi(t)$ est un signal de moyenne variable. Le polynôme $C(q^{-1})$ modélise l'influence du bruit sur le système. L'introduction de $\Delta(q^{-1}) = 1 - q^{-1}$ dans le modèle de bruit permet d'ajouter une action intégrale dans le correcteur afin d'annuler l'erreur statique.
Ce modèle est représenté par la Figure 4.1 :

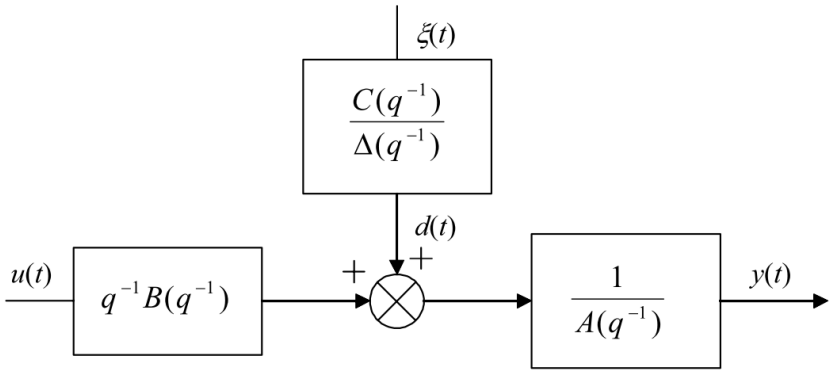


Figure 4.1 – Modèle CARIMA

Où les fonctions $A(q^{-1})$, $B(q^{-1})$, $C(q^{-1})$ représentent des polynômes définis par l'équation 4.2

$$Avec : \begin{cases} A(q^{-1}) &= 1 + a_1 q^{-1} + ... + a_{n_a} q^{-n_a} \\ B(q^{-1}) &= b_0 + b_1 q^{-1} + ... + b_{n_b} q^{-n_b} \\ C(q^{-1}) &= 1 + c_1 q^{-1} + ... + c_{n_c} q^{-n_c} \end{cases} \tag{4.2}$$

L'écriture RST de la GPC est traduite graphiquement par la Figure 4.2. Le contrôleur est à deux degrés de liberté permettant de séparer la dynamique de régulation de la dynamique de poursuite [58] : L'équation mathématique du modèle RST est donnée 4.3 :

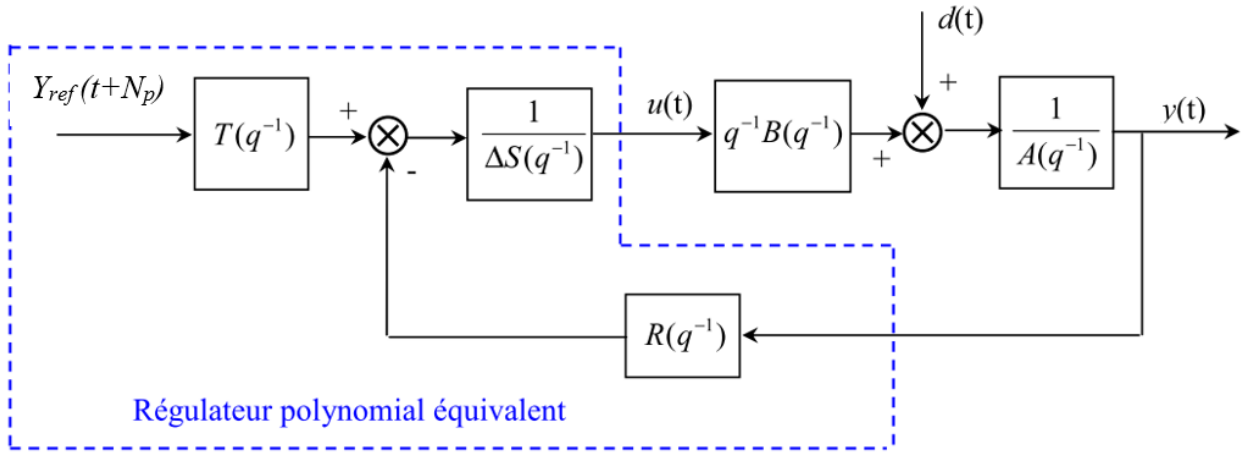


Figure 4.2 – Structure du régulateur polynomial équivalent

$$S(q^{-1})\Delta u(t) = -R(q^{-1})y(t) + T(q^{-1})y_{ref}(t+N_P) \tag{4.3}$$

Avec :

$$\begin{cases} S(q^{-1}) &= C(q^{-1}) + n_1^T q^{-1} \\ R(q^{-1}) &= n_1^T \\ T(q^{-1}) &= C(q^{-1}) n_1^T [q^{-N_P+1}\ q^{-N_P}\ q^{-N_P-1} \dots 1]^T \end{cases}$$

Pour robustifier la GPC, l'idée a été d'introduire un paramètre Q dans la boucle de commande[13].

L'introduction de se paramètre est illustrée dans la Figure 4.3 : B_0 et A_0 représentent respectivement les valeurs initiales de A et B dans le modèle RST. Le paramètre Q est utilisé pour rajouter une couche de robustification. Il permet d'introduire un degré de liberté supplémentaire par rapport au polynôme C (Figure 4.1), ce qui se traduit par l'accès à un plus grand nombre de correcteurs que ceux atteints avec le polynôme C. C'est dans ce contexte que plusieurs travaux ont été menés pour la robustification de la GPC :

— **Approche SGPC**

En 1992 (Kouvaritakis et al,) ont proposé une approche SGPC (Stable generalized predictive control) de robustificiation destinée à une classe des contrôleurs mono-variables pour lesquels la poursuite de la consigne et la robustesse **(SISO)** sont assurées avec des marges acceptables [29].

Cette approche permet de retrouver le paramètre Q afin de maximiser la marge de robustesse en minimisant une norme pondérée et de garantir ainsi la stabilité en

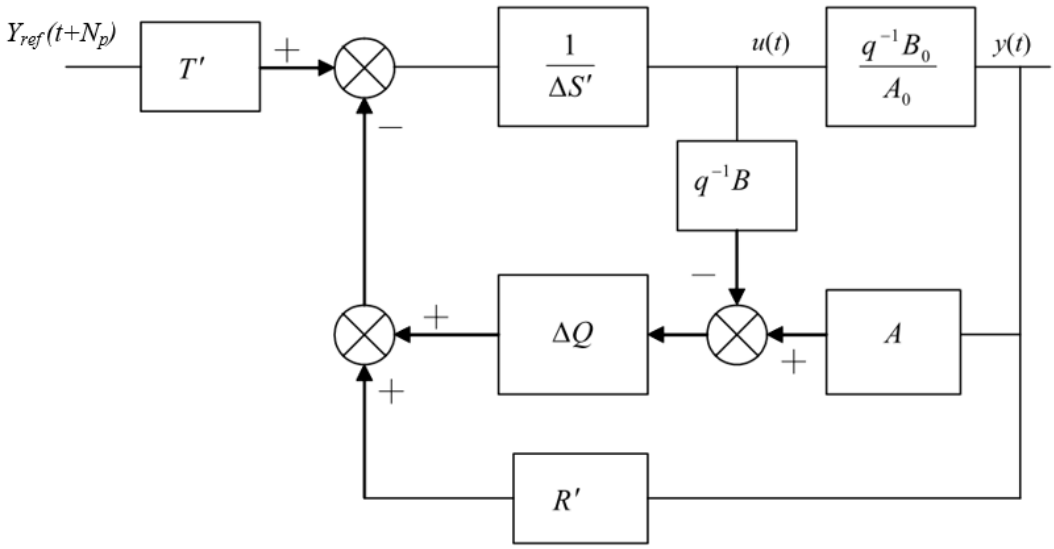


Figure 4.3 – Régulateur polynomial équivalent

boucle fermée.

La marge de robustesse du système est donc définie par l'équation 4.11 :

$$B_r = \left| \frac{A_c(q^{-1})C(q^{-1})}{A(q^{-1})R(q^{-1})} \right| = \left| \frac{1}{U} \right| \tag{4.4}$$

L'objectif est d'augmenter la robustesse. Donc il faut retrouver le facteur Q tel que $\frac{1}{U}$ soit maximisé. En introduisant une pondération W qui permet de pondérer de façon plus importante les fréquences où les incertitudes du modèle tel que :

$$min_Q \left\| W(q^{-1})U(q^{-1}) \right\| = min_Q sup_\omega \left| W(e^{j\omega})U(e^{j\omega}) \right| \tag{4.5}$$

Il faut alors retrouver Q tel que la norme H_∞ soit minimisée.

Pour $A(q^{-1})$ stable, une expression de Q_{opt} est donnée par l'équation 4.16 [29] et[1] :

$$Q_{opt} = \frac{Q_{n_{opt}}}{Q_{d_{opt}}} = \frac{-W(q^{-1})A(q^{-1})R_0(q^{-1}) + \frac{W(1)A(1)R_0(1)}{A_c(1)}}{W(q^{-1})A(q^{-1})^2\Delta(q^{-1})} \tag{4.6}$$

où $R_0(q^{-1})$ est le correcteur initial (pour $C = 1$).

En revanche d'après (Pedro et al,), le rejet de perturbation est plus lent avec cette approche [13]. Il est possible de modifier la pondération des fréquences dans la minimisation du critère, mais le compromis entre le rejet de perturbations et la robustesse en performance est difficilement géré.

— **Approche C-design**

Cette approche repose sur un choix simple de C pour les cas où le système est

stable. Ce choix procure une robustesse équivalente à celle obtenue par l'approche précédente. Elle a été utilisée par plusieurs auteurs, notamment en 1995 puis en 2003, par (Yoon et al,), (Rodriguez et al,) qui ont utilisé la méthode C-design pour robustifier la commande prédictive [168] et [13] . En 1995 (Yoon et al,) ont proposé une nouvelle stratégie de robustification basée sur l'introduction d'un polynôme noté *(C-polynomial)* dans le modèle CARIMA [168].
Le polynôme C proposé est alors :

$$C(q^{-1}) = A(q^{-1})(1 - \beta q^{-1})^{1-\delta(P)} \tag{4.7}$$

Avec le choix suivant des paramètres de la GPC :
($N_u \geq n$, $N_P = n$ et $\lambda = \rho(i) = \alpha^{-2i}$) où n est l'ordre du système décrit par :

$$n = \frac{q^{-1}B(q^{-1})}{A(q^{-1})\Delta(q^{-1})} \tag{4.8}$$

— β doit être approximativement de l'ordre du pôle dominant de A .
— $\delta(P)$ représente le degré du polynôme P. Ce polynôme correspond à la définition de la trajectoire de référence.

Pour une consigne de type échelon, $\delta(P) = 0$ et $\lambda = constante$ alors :

$$C(q^{-1}) = A(q^{-1})(1 - \beta q^{-1})$$

Cette méthode est simple et permet de trouver une marge de robustesse plus grande que celle de la **SGPC**. Néanmoins, si le système est instable, le calcul de C n'est plus valable et il est alors nécessaire de passer au préalable par une autre méthode comme, par exemple, celle de la SGPC.
Avec cette approche, le compromis entre la robustesse en performances et le rejet de perturbation est aussi compliqué à obtenir [168].

— **Approche Q-design**

L'objectif de cette approche est de retrouver le paramètre Q sans dégrader les performances du système [130]
Considérons l'expression de la marge de robustesse obtenue à partir de l'équation 4.11.

$$B_r = \left| \frac{A_c(q^{-1})}{A(q^{-1})R_0(q^{-1})} \frac{R_0(q^{-1})}{R_0(q^{-1}) + \Delta A(q^{-1})Q(q^{-1})} \right| \tag{4.9}$$

Dans l'expression 4.9, A_c et A sont fixés en fonction du réglage paramétrique de la GPC. Il serait possible de robustifier le système en modifiant $\frac{C}{R}$, mais le problème qui se pose est le choix, de la forme de $\frac{C}{R}$, car R dépend de C. Pour éviter ce

problème, l'idée de (P.Ansay et al,) a consisté à choisir un paramètre Q permettant de découpler les deux parties sous la forme [130] :

$$Q = \frac{R_0}{A}\frac{M^*}{C^*}$$

On a alors :

$$B_r = \left| \frac{A_c(q^{-1})}{A(q^{-1})R_0(q^{-1})} \left(\frac{1}{1 + \frac{\Delta(q^{-1})M^*}{C^*}} \right) \right| \tag{4.10}$$

Dans l'expression (4.10), la première partie est donnée par les paramètres GPC initiale, et la deuxième partie est fixée librement.
Le choix proposé est de considérer $\frac{1}{1+\frac{\Delta(q^{-1})M^*}{C^*}}$ comme un filtre passe-haut, afin que le système devienne ainsi moins sensible aux incertitudes en hautes fréquences.
La structure classiquement définie est donnée par :

$$\frac{1}{1 + \frac{\Delta(q^{-1})M^*}{C^*}} = \frac{1 - \mu_1 q^{-1}}{(1 - \mu_2)(\mu_2 - \mu_1)q^{-1}}$$

Avec : $\mu_1 \prec 1$ et $0 \prec \mu_2 \leq \mu_1$
Par conséquent, 4.10 devient :

$$B_r = \left| \left(\frac{A_c(q^{-1})}{A(q^{-1})R_0(q^{-1})} \right) \frac{1 - \mu_1 q^{-1}}{(1 - \mu_1)(\mu_2 - \mu_1)q^{-1}} \right| \tag{4.11}$$

Dans le cas où le système est instable, le choix de $Q = \frac{R_0}{A}\frac{M^*}{C^*}$ n'est pas possible car A est instable. La solution est donc de diviser A en deux parties une stable A^s et une instable A^u sous la forme $A = A^s A^u$, et l'on choisit Q sous la forme :

$$Q = \frac{R_0}{A^s}\frac{M^*}{C^*}$$

Le filtre passe-haut devient un filtre d'ordre plus élevé.
La méthode **Q-design**, permet de gérer le compromis entre le rejet de perturbation et la robustesse, et elle peut aussi être appliquée à un système instable.

— **Approche par paramétrisation de Youla**

La paramétrisation de Youla ou appelée aussi $Q - paramtrisation$ est un outil qui permet d'établir un lien entre l'ensemble des correcteurs stabilisant un système. Cette paramétrisation permet de formuler le problème de robustification comme étant un problème d'optimisation. Cette approche a fait l'objet de plusieurs travaux de recherches et la détermination du paramètre de Youla varie en fonction des auteurs [13].
La synthèse de transfert de Youla est formée par un polynôme discret à temps

variant. Les coefficients de ce polynôme sont obtenus à partir de la résolution d'un problème d'optimisation quadratique formulé par les spécifications fréquentielles définies par le cahier des charges.

Pour obtenir le paramètre caractérisant cette approche, il faut définir une structure de bouclage pour le correcteur initial défini dans la Figure 4.2. La structure considérée est donnée dans la Figure 4.4 [13] :

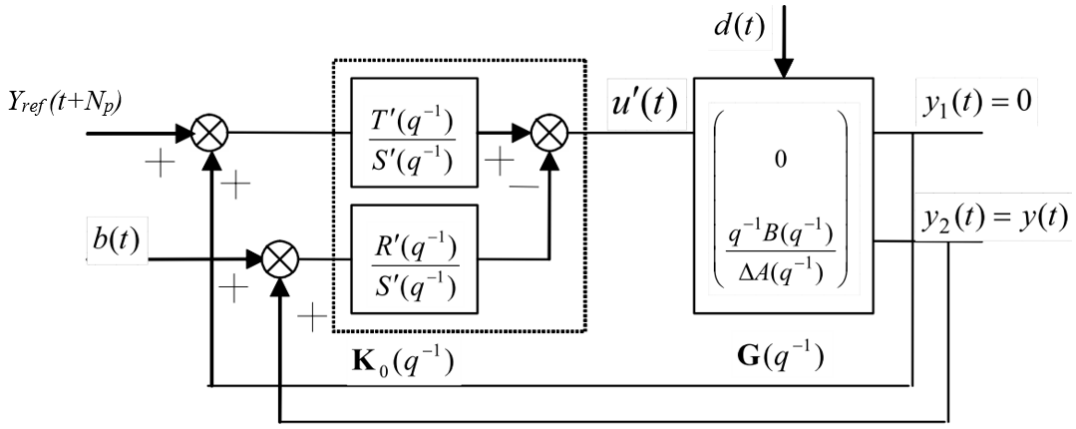


Figure 4.4 – Correcteur initial avec la structure du bouclage standard

En utilisant la structure définie dans la Figure (4.4), un paramètre de Youla a deux degrés de liberté peut être déterminer. L'action intégrale est incluse dans le modèle du système par : $G(q^{-1}) = \begin{pmatrix} 0 \\ \frac{q^{-1}B}{\Delta A} \end{pmatrix}$ *et* $K_0(q^{-1}) = \begin{pmatrix} \frac{T'}{S'} & \frac{R'}{S'} \end{pmatrix}$

La modification de la représentation donnée par la Figure 4.4, permet de faire apparaître les deux paramètres Q_1 et Q_2 à l'intérieur du même bloc comme montré sur la Figure 4.5, ce qui permettra par la suite de déterminer le paramètre de Youla Q :

La matrice H qui représente la structure globale qui prend en compte les signaux de perturbation et de bruit est donnée par :

$$\begin{pmatrix} \xi \\ u \\ y \end{pmatrix} \tag{4.12}$$

$$= H \begin{pmatrix} b \\ d \\ y_{ref} \end{pmatrix} \tag{4.13}$$

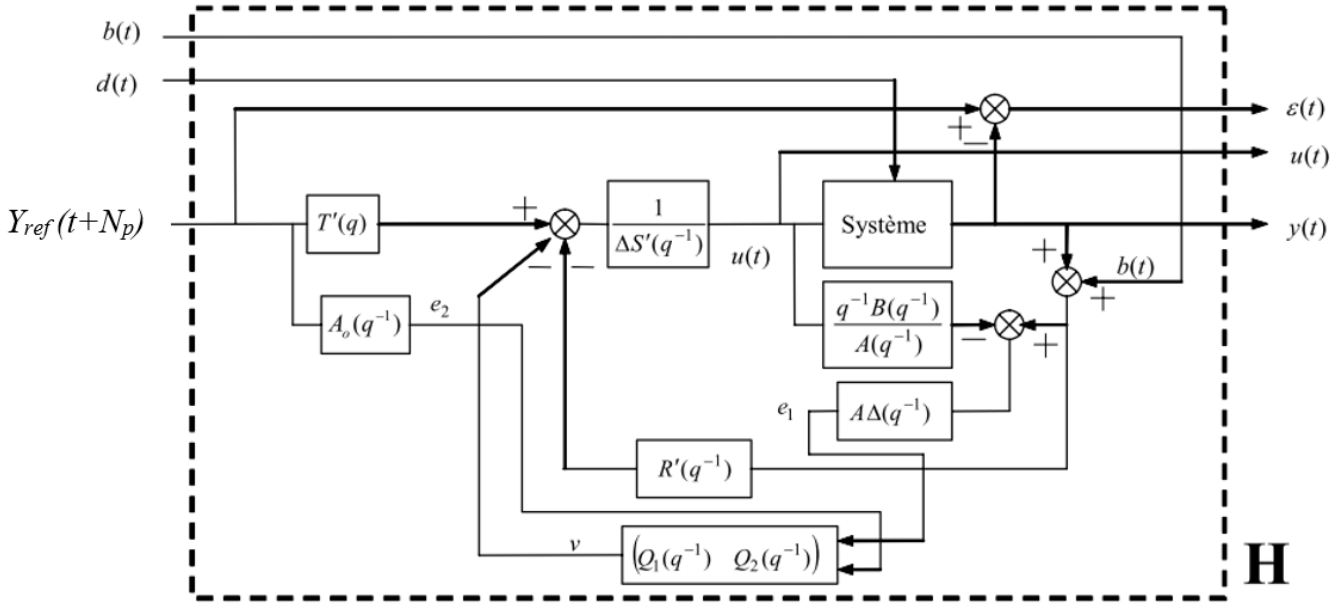


Figure 4.5 – Correcteur GPC à deux degrés de liberté avec paramétrisation de Youla

$$H = \begin{pmatrix} \frac{q^{-1}BR'}{A_0A_c} + \frac{q^{-1}B\Delta A}{A_0A_c}Q_1 & -\frac{\Delta S'}{A_0A_c} + \frac{q^{-1}B\Delta}{A_0A_c}Q_1 & \frac{A_0A_cq^{-N_P} - q^{-1}BT'}{A_0A_c} + \frac{q^{-1}BA_0}{A_0A_c}Q_2 \\ -\frac{R'A}{A_0A_c} - \frac{\Delta A^2}{A_0A_c}Q_1 & -\frac{R'}{A_0A_c} - \frac{\Delta A}{A_0A_c}Q_1 & \frac{T'A}{A_0A_c} - \frac{AA_0}{A_0A_c}Q_2 \\ -\frac{q^{-1}BR'}{A_0A_c} - \frac{q^{-1}B\Delta A}{A_0A_c}Q_1 & \frac{\Delta S'}{A_0A_c} - \frac{q^{-1}B\Delta}{A_0A_c}Q_1 & \frac{T'q^{-1}B}{A_0A_c} - \frac{q^{-1}BA_0}{A_0A_c}Q_2 \end{pmatrix}$$

A partir de la Figure 4.5 et de l'expression de H donnée par 4.13, on constate que le paramètre Q_1 modifie les fonctions de transfert associant $b(t)$ et $d(t)$ à $\xi(t)$, $u(t)$ et $y(t)$.
Par ailleurs, le paramètre Q_2 modifie les fonctions de transfert qui relient y_{ref} à $\xi(t)$, $u(t)$ et $y(t)$. Par conséquence, Q_1 modifie la dynamique de la boucle fermée, et Q_2 modifie la dynamique de poursuite [152].

Dans cette section, nous avons présenté les différentes approches pour robustifier un contrôleur GPC. L'organigramme suivant 4.6 résume les étapes de robustifications basées sur la représentation polynomiale :

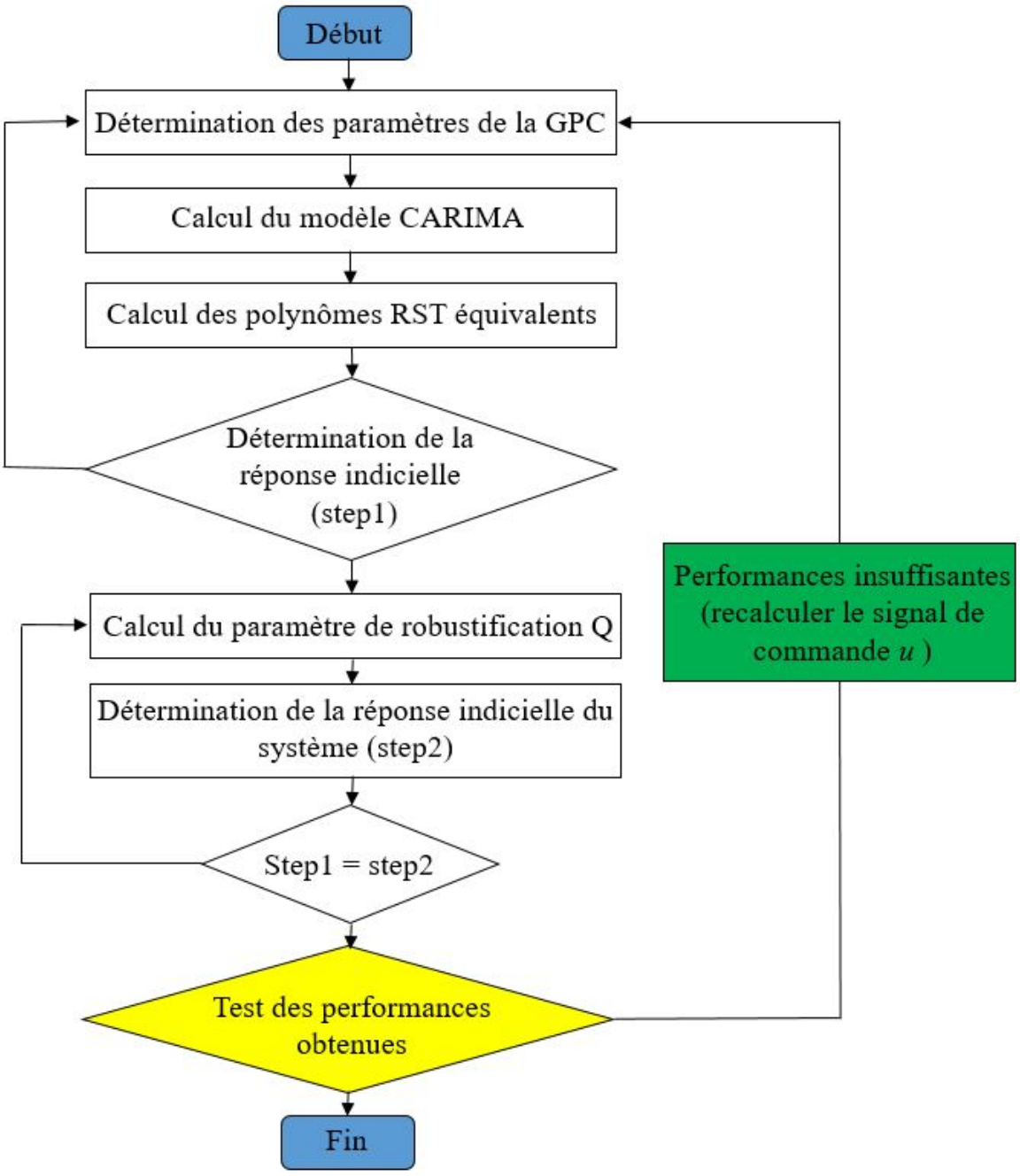


Figure 4.6 – Organigramme proposé pour la robustification de la GPC

4.2.2 Approches basées sur la représentation d'état

Dans cette partie, nous allons effectué un état de l'art sur les approches de robustification basées sur la représentation d'état. Dans la littérature, on a pu identifier plusieurs approches de robustification qui sont basées sur la commande prédictive sous formalisme d'état qui permet une manipulation des systèmes multivariables.

— **Approche Min-Max**

La robustification de la MPC par approche min-max repose sur la minimisation de la fonction coût nominale J. Cette fonction est formulée comme étant un problème d'optimisation convexe sous contraintes LMI. La commande prédictive est alors déterminée à base de programmation semi-définie [118] [28] .
Soient $J(k)$ la fonction coût nominale à minimiser qui dépend des paramètres de réglage de la MPC (λ , N_u et N_P) et $V(x(k)) = x(k)^T P x(k)$, avec $P \succ 0$ une fonction quadratique de Lyapunov, $V(x(k))$ qui majore $J(k)$, et γ sa borne supérieure telles que :

$$V(x(k/k)) \leq \gamma \tag{4.14}$$

$$maxJ(k) \leq V(x(k/k)) \tag{4.15}$$

Le problème d'optimisation consiste à retrouver la loi de commande par retour d'état $u(k+i/k) = Fx(k+i/k)$ avec la résolution du problème LMI suivant :
Soient $x(k) = x(k/k)$ le vecteur d'état du système S mesuré à chaque pas d'échantillonnage k et les vecteurs u_{max} , y_{max} les contraintes maximales respectivement sur la commande et la sortie. Le système est asymptotiquement stabilisé par un correcteur par retour d'état, avec une minimisation de la borne supérieure de la fonction coût $J(k)$ sous contraintes si, à chaque pas d'échantillonnage (k=0,1,... ,∞), il existe une matrice symétrique définie positive Q et une matrice Y représentant les solutions du problème suivant :

$$min\gamma \tag{4.16}$$

Sous

$$\begin{bmatrix} 1 & x(k/k)^T \\ x(k/k) & Q \end{bmatrix} \geq 0 \tag{4.17}$$

$$\begin{bmatrix} Q & QA_j^T + Y^T B_j^T & QQ_1^{1/2} & Y^T R^{1/2} \\ A_j Q^T + B_j Y & Q & 0 & 0 \\ Q_1^{1/2} Q & 0 & \gamma I & 0 \\ R^{1/2} Y & 0 & 0 & \gamma I \end{bmatrix} \geq 0 \tag{4.18}$$

Avec j= 1, 2,...,L

$$\begin{bmatrix} u_{max}^2 I & Y \\ Y^T & Q \end{bmatrix} \geq 0 \tag{4.19}$$

$$\begin{bmatrix} y_{max}^2 I & C(A_j Q^T + B_j Y) \\ (A_j Q^T + B_j Y)^T C^T & Q \end{bmatrix} \geq 0 \tag{4.20}$$

Une fois le problème LMI précédent résolu (où Q , Y et Q_1 sont les inconnus), le régulateur par retour d'état est donné par :

$$u(k) = Fx_m(k) = YQ^{-1}x_m(k) \tag{4.21}$$

L'approche de résolution la plus souvent utilisée dans la commande robuste est celle de H_∞ basée soit sur la résolution des deux équations algébriques de Ricatti ou bien celle basée sur l'approche des LMIs [28].

— **Approche par paramétrisation de Youla**

L'approche de Youla pour la robustification de la commande prédictive à base d'état (MPC), a été utilisée comme celle pour la GPC dans plusieurs travaux de recherches. Le paramètre de robustificiation de Youla permet de modifier la fonction de sensibilité qui, à son tour, améliore la robustesse, sans avoir à changer le comportement E/S du processus contrôlé par le correcteur initial.

Cette approche a été proposée par (Stoica et al,) afin de robustifier une loi de commande prédictive à base d'état appliquée aux systèmes représentés par des modèles linéaires à temps discret [152]. La Figure 4.7 représente la solution du problème d'optimisation convexe du paramètre de Youla : Le paramètre de Youla

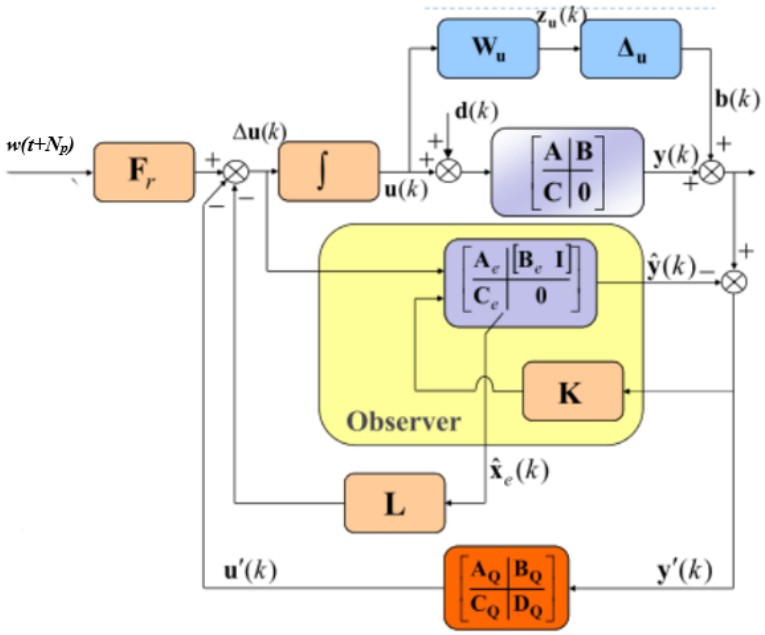


Figure 4.7 – Structure de la MPC MIMO robustifiée face à des incertitudes non-structurées additives via le paramètre de Youla [152]

est représenté dans la Figure 4.7 sous la forme d'une représentation d'état. Le correcteur MPC initial permet de déterminer F_r et K par la représentation d'état

du modèle augmenté. W_u représente la pondération sur le signal de commande et Δ_u un vecteur d'incertitude non structurées additives.

Afin de faciliter l'implémentation industrielle du régulateur robustifié, (Stoica et al,) ont également conçu une *boîte à outils pour le logiciel MATLAB-TM*, regroupant la technique globale de la robustification de la commande prédictive en présence de perturbations et d'incertitudes non-structurés qui garantissent les performances définies initialement par le système nominal [152].

— **Approche basée sur l'utilisation du filtrage de Kalman**

L'utilisation du Filtre de Kalman a été utilisée pour l'estimation de l'état dans le processus perturbé par le calcul d'un *polynôme* T dans la conception. Cette approche a pour objectif d'introduire une description complètement nouvelle des contrôleurs prédictifs à base de modèle (MPC) dans la représentation d'état [111] et et [9].

.

A partir des différents approches et comme déjà présentées pour la robustification basée sur la représentation polynomiale, l'organigramme 4.8 présente les étapes de robustification d'un contrôleur prédictif basé sur la représentation d'état :

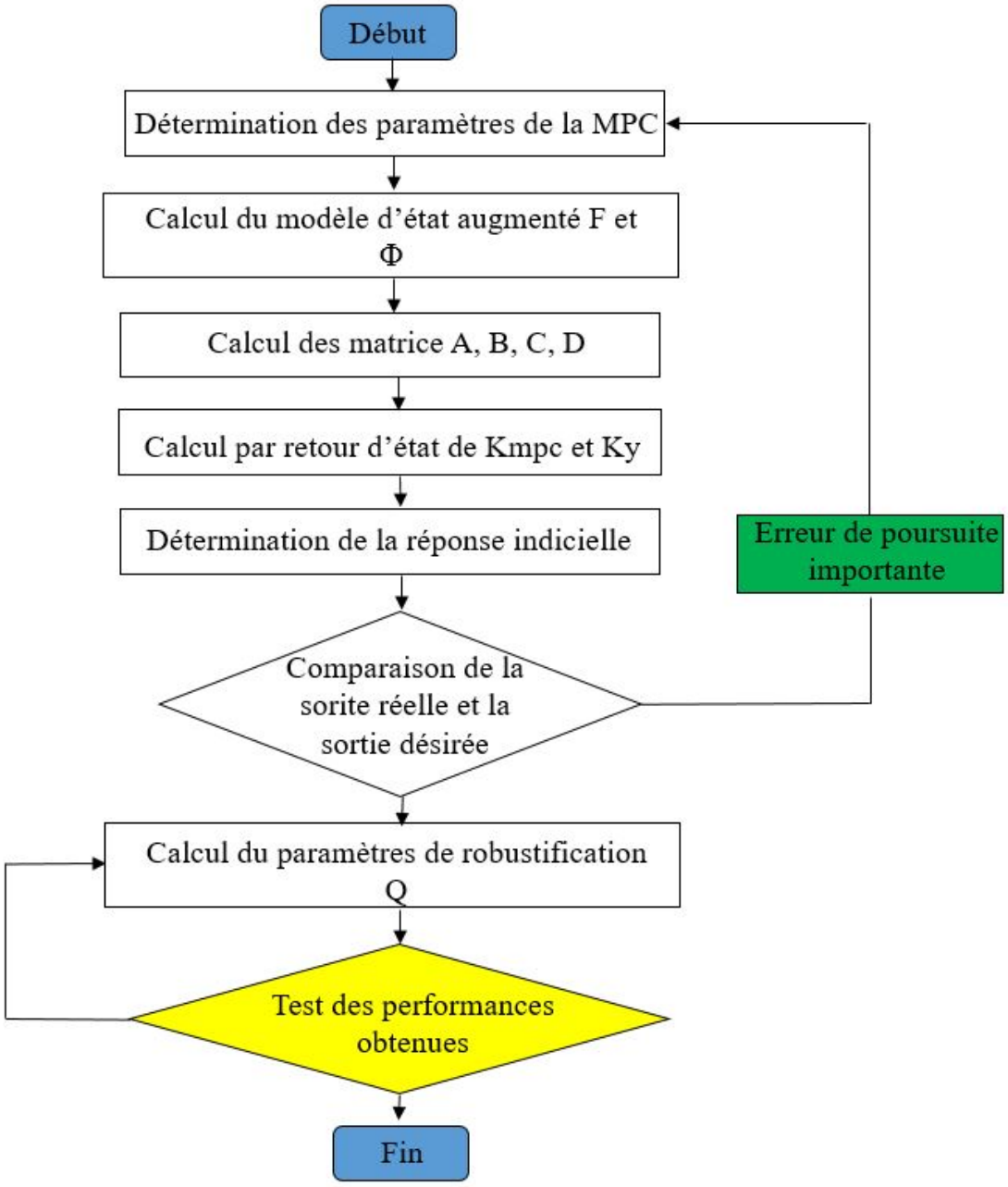


Figure 4.8 – Organigramme proposé pour la robustification de la MPC

4.2.3 Synthèse comparative des approches de robustification de la commande prédictive

Dans cette partie, une étude comparative des différentes approches identifiées dans la littérature sera présentée. L'objectif est de choisir l'approche qui sera utilisée en combinaison avec l'approche proposée tout en identifiant les avantages et limites de chaque approche.

— **Système considéré**

Le systèmes considéré est LTI (Linear Time Invariant) SISO [1][168], utilisé à plusieurs reprises dans la littérature portant sur la commande prédictive. Ce système développé par (A.Debbeh et al,) est représenté par une fonction de transfert du troisième ordre donné par l'équation 4.22 :

$$\frac{Y(s)}{U(s)} = \frac{225}{s^3 + 3.4s^2 + 227.4s + 225} \tag{4.22}$$

Pour une période d'échantillonnage de $T_e = 0.1s$, on obtient :

$$\left\{ \begin{array}{c} A_0 = 1 - 1.039z^{-1} + 0.9079z^{-2} - 0.7118z^{-3} \\ B_0 = 0.03084 + 0.1005z^{-1} + 0.02595z^{-2} \end{array} \right\} \tag{4.23}$$

Les paramètres considérés pour le réglage de la GPC sont : $N_P = 5$, $N_u = 1$ et $\lambda = 1$ [1].

Le correcteur sous la forme polynomiale RST équivalent de la GPC initiale que l'on note R_0, S_0, T_0 est donné par 4.24 [28] :

$$\left\{ \begin{array}{c} R_0 = 2.85 - 2.073z^{-1} \\ S_0 = 1 \\ T_0 = 0.2133 + 0.1917z^{-1} + 0.1653z^{-2} + 0.1293z^{-3} + 0.07697z^{-4} \end{array} \right\} \tag{4.24}$$

— **Test des différentes approches de robustification en absence de perturbations**

La réponse indicielle de ce système avec un correcteur initial $C(q^{-1}) = 1$ sans recours à des outils de robustification est donnée par la Figure 4.9 :

On remarque clairement que la sortie du système représenté par la Figure 4.9 est oscillante en régime permanent.

Dans ce qui suit, nous allons robustifier le correcteur initialement définit du GPC par les approches de robustifications afin de dresser une étude comparative entre ces approches. L'objectif étant de déterminer le paramètre Q de la Figure 4.3.

Le Tableau 4.1 donne le paramètre Q de chaque approche et qui sera utilisé par la suite en simulation pour comparer les quatre approches SGPC, C-design, Q-design et Youla :

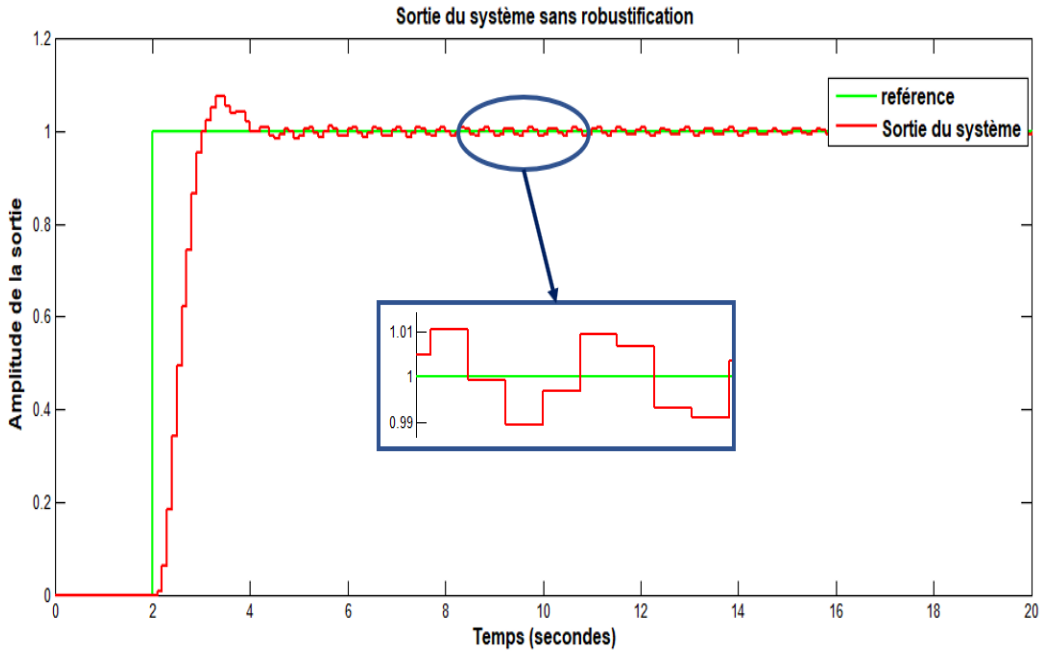


Figure 4.9 – Réponses indicielles du système avec GPC primaire

Approches polynomiales	Paramètres	Paramètre de robustification Q
SGPC	$W = \frac{1-0.8q^{-1}}{0.2}$	$Q_{opt} = \frac{-3.33+4.96q^{-1}-1.87q^{-2}}{(1-0.8q^{-1})(1-0.9q^{-1})^2}$
C-design	$\beta = 0.8$	$Q_{opt} = \frac{M}{C} = \frac{-3.345+2.077q^{-1}}{(1-0.8q^{-1})(1-0.9q^{-1})}$
Q-design	$\mu_1 = 0.8\ \mu_2 = 0.6$	$Q = \frac{-2.12+1.56}{(1-0.8q^{-1})(1-0.9q^{-1})}$
Youla	$\mu_1 = 0.9\ \mu_2 = 0.7\ A_s = 1$	$Q = \frac{-2.204+1.626q^{-1}}{1-1.6q^{-1}+0.63q^{-2}}$

Tableau 4.1 – Paramètre Q de robustification.

Pour conclure sur l'efficacité de chacune de ces approches, nous allons commencer par tracer le diagramme de gain (en dB) de la marge de robustesse (voir Figure 4.10) :

Les meilleures marges de robustesse sont obtenues avec les approches C-design et SGPC puisqu'en hautes fréquences, elles prennent des valeurs de gain de 20dB et 33dB respectivement.

Après avoir testé les différentes approches de robustification, nous allons, par la suite, les appliquer sur le système initialement défini par l'équation 4.22.

La Figure 4.11 représente la réponse indicielle du système en absence de

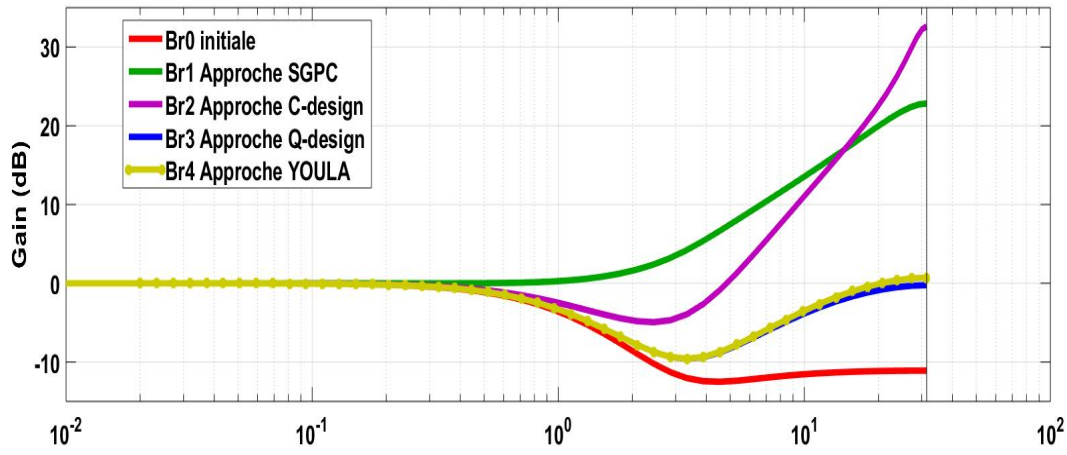


Figure 4.10 – Marges de robustesse B_r pour les différentes méthodes.

perturbation :

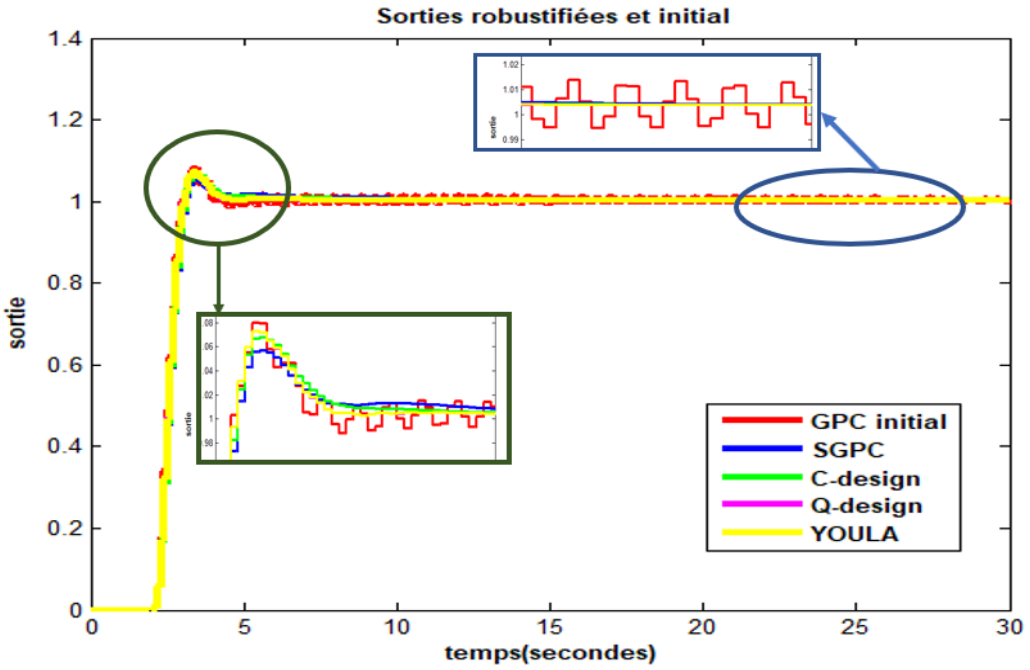


Figure 4.11 – Réponse indicielle du système en absence de perturbations

A partir de la Figure 4.9, nous avons montré que la réponse indicielle du système sans outil de robustification était oscillante suite à la dynamique négligée du système.

En analysant la Figure 4.11, nous observons que la réponse indicielle du système avec le correcteur robustifié par les différentes approches n'est plus oscillante. Concernant le signal de commande représenté dans la Figure 4.12, nous remarquons également une réponse des signaux de commande plus lisse avec les commandes robustifiées :

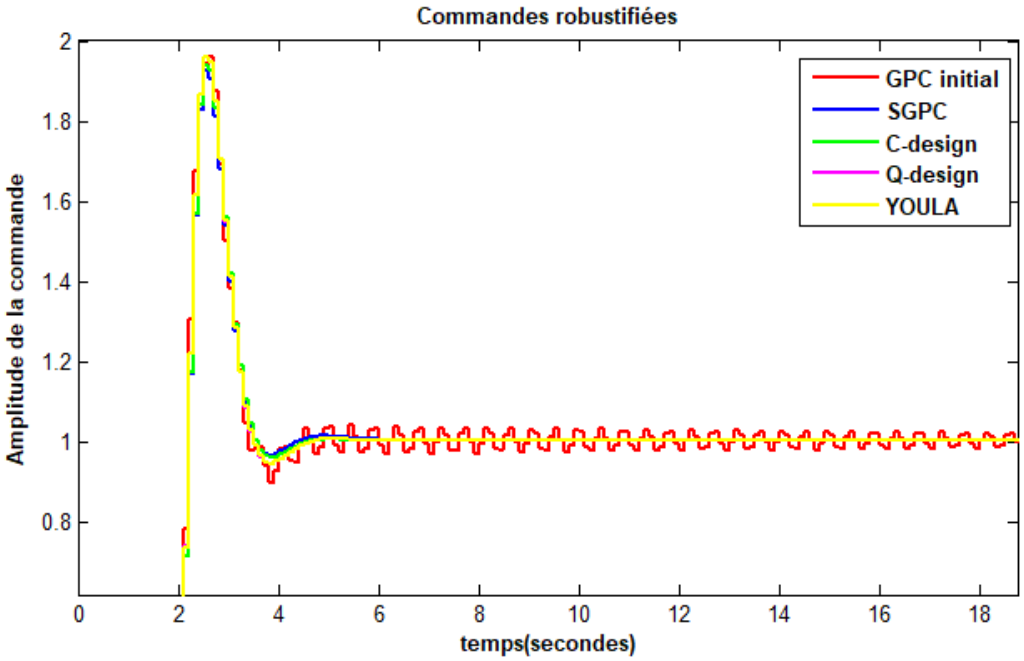


Figure 4.12 – Signaux de commande du système en absence de perturbation

Nous remarquons aussi que les oscillations sont atténuées par la robustification.

— **Test des différentes approches de robustification en présence de perturbations**

Dans cette partie, nous allons examiner l'impact de la robustification sur la dynamique de la boucle fermée en présence de perturbations. C'est dans ce sens que nous effectuons les simulations sur le système avec un échelon en entrée appliqué à l'instant $t = 2s$ et une perturbation $d(t)$ de type échelon d'amplitude 0.1 appliquée à partir de $t = 7s$ comme présenté dans la Figure 4.13.

La Figure 4.13 représente la sortie du système en présence de perturbations avec le correcteur initial et les différents correcteurs robustifiés.

D'après cette figure, nous observons que les différentes approches de robustifications permettent le rejet de la perturbation injectée par comparaison avec le correcteur initial non robustifié qui, lui conduit à une réponse oscillante en régime permanent.

Nous constatons également une légère différence entre les approches de robustification, notamment au niveau de la rapidité à rejeter la perturbation. L'approche SGPC est la plus lente avec un temps de rejet de $\Delta t_1 = 6s$, suivie par l'approche C-design avec un temps de $\Delta t_2 = 3.5s$. A l'aide des approches Youla et Q-design le rejet de perturbations est beaucoup plus rapide avec des valeurs presque similaires $\Delta t_3 = \Delta t_4 = 2.5s$.

La Figure 4.14 représente les signaux de commande appliqués au système en présence de perturbations avec le correcteur initial et les différents régulateurs

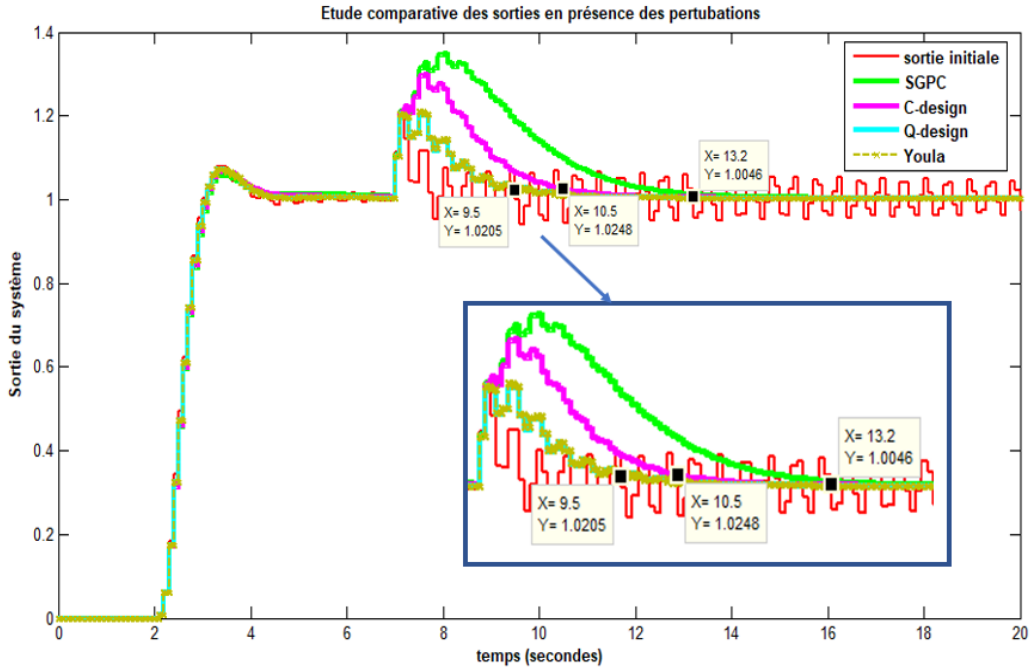


Figure 4.13 – Réponses indicielles avec correcteur robustifié en présence des perturbations.

robustifiés :

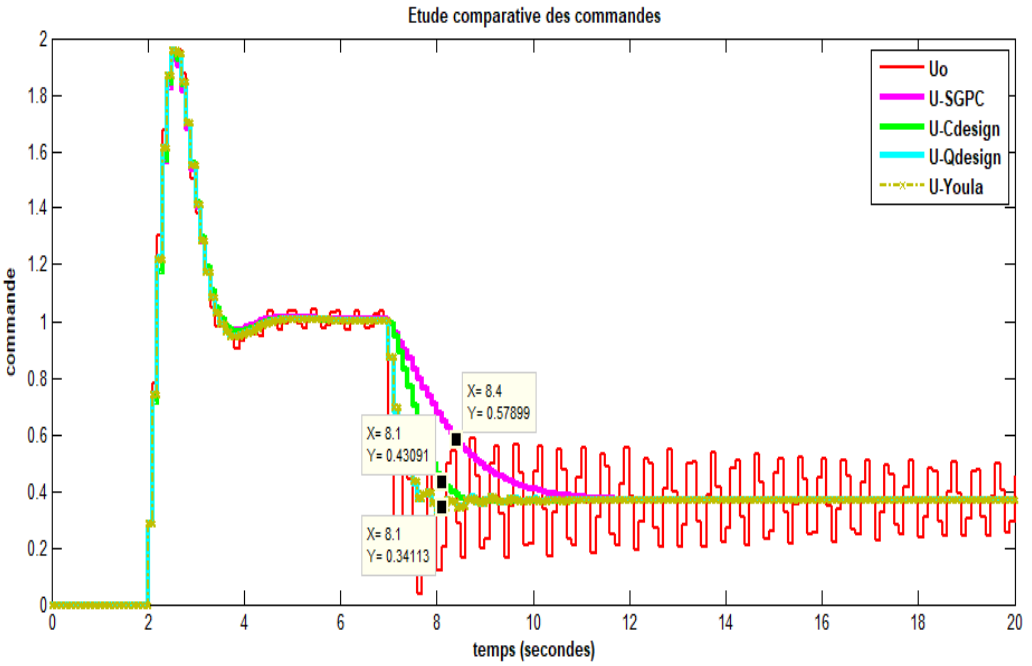


Figure 4.14 – Évolution des signaux de commande en présence de perturbations

Concernant les signaux de commande présentés dans la Figure 4.14, nous observons qu'avec toutes les approches de robustification le signal de commande ne présente plus d'oscillation. La seul différence se situe au niveau de l'amplitude des ocsillations, qui est plus faible avec les approches de Youla et Q design avec une valeur de $0.34u.a$ ensuite C-design $0.43u.a$ et suivi finalement par SGPC avec une

amplitude de $0.58u.a$.

En comparant les performances obtenues par les quatre approches de robustification, nous avons pu remarquer que l'approche de Youla et celle par Q-design permettent un rejet de perturbations plus rapide et une amplitude du signal de commande plus faible, et donc des performances quasi similaires et nettement supérieures à celles obtenues par les deux autres approches (C-design et SGPC). Donc notre choix s'est porté sur l'approche de Youla qui est plus simple en termes de calculs et plus performante.

Dans la partie qui suit l'approche de Youla sera considéré comme outil de référence pour la robustification de la commande prédictive.

4.3 Influence des paramètres de réglage sur la robustesse de la commande prédictive

L'objectif de cette partie est de montrer l'importance et le rôle que joue le réglage paramétrique de la commande prédictive sur le rejet de perturbations. Nous allons considérer dans cette partie l'approche par paramétrisation de Youla pour robustifier la MPC par RNA.

4.3.1 Comparaison entre la MPC réglée par RNA et la GPC non robustifiées

Pour montrer l'influence des paramètres de réglage sur le rejet de perturbations, nous allons commencer par faire une comparaison entre deux approches non robustifiées : une MPC réglée par RNA et la GPC initiale.

Le système considéré en simulation est le même que celui défini précédemment par l'équation 4.23.

Les paramètres de réglage considérés en simulation sont donnés par :

- pour la GPC initiale [1] : $N_P = 5$, $N_u = 1$ et $\lambda = 1$.
- pour la MPC réglée par RNA : $N_P = 23$, $N_c = 12$ et $\lambda = 4,7918$

Dans cette simulation, nous avons injectée également un bruit de mesure à l'instant $t = 14s$ comme présenté dans la Figure 4.13, en plus de la perturbation $d(t)$ injecté à l'instant $t = 7s$. Ce bruit de mesure $b(t)$ est un bruit blanc avec une variance de $0,025$.

Dans la Figure 4.15, on observe la comparaison de la sortie du système en présence de la perturbation et du bruit de mesure sans robustification de la loi de commande :

La Figure 4.16 montre l'évolution des signaux de commande appliqués au système en

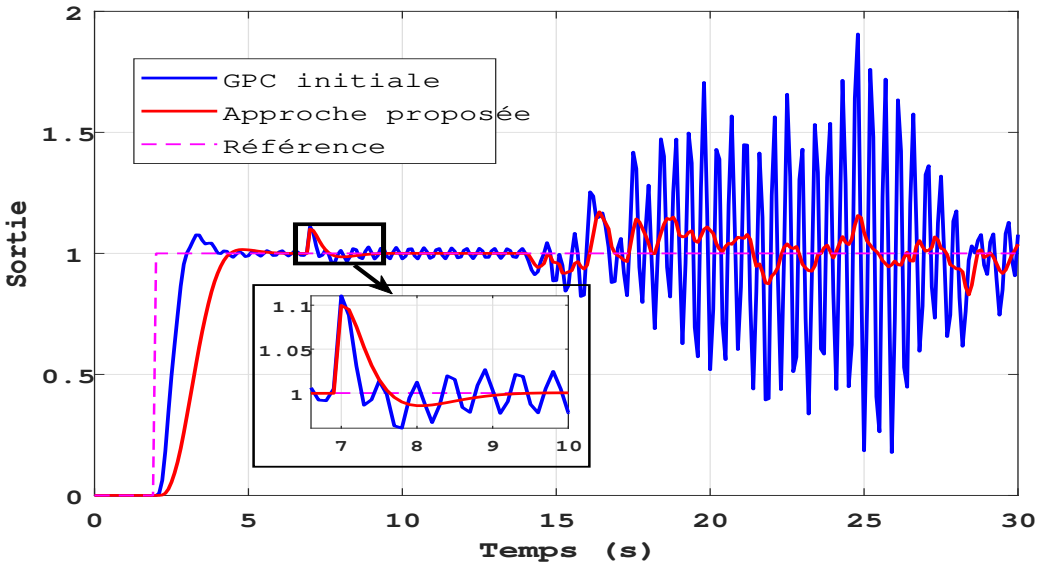


Figure 4.15 – Sortie du système en fonction du temps

présence de la perturbation et du bruit de mesure sans robustification : la comparaison

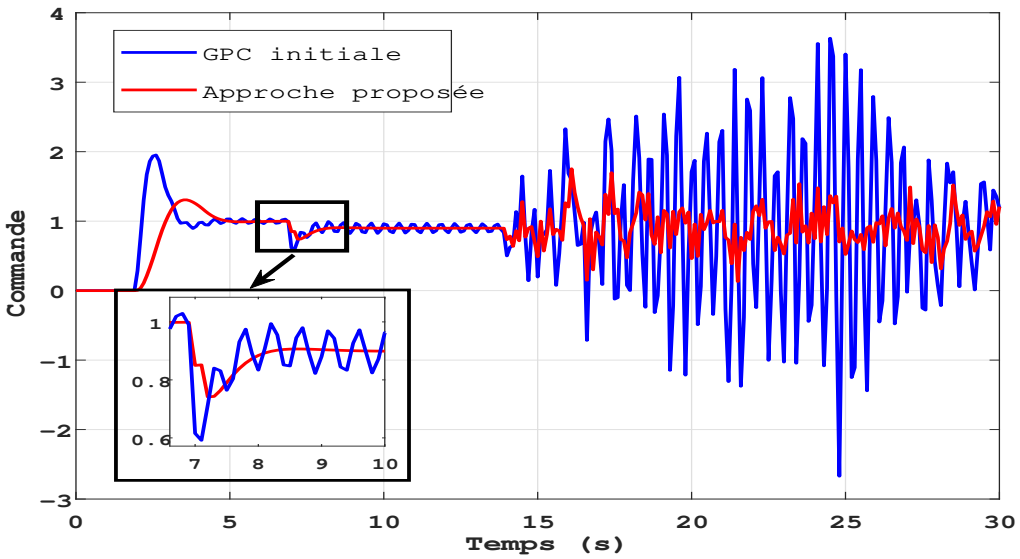


Figure 4.16 – Signaux de commande en fonction du temps

des deux courbes dans les deux Figures 4.15 et 4.16 montre clairement qu'avec l'approche proposée, on arrive à avoir des meilleures performances. Précisément, en termes de rejet de perturbation à l'instant $t = 7s$, et une atténuation du bruit à l'instant $t = 14s$, en comparaison avec l'approche basée sur la GPC à son tour ne permet pas le rejet de perturbation et ainsi qu'une atténuation du bruit quasi inexistante [83].

A partir de ces deux figures, le rôle du réglage paramétrique de la commande prédictive

est clairement mis en avant. Ce réglage optimale permet le rejet de perturbation sans avoir recours à des outils de robustificiation, qui peuvent certes améliorer les performances mais ces outils sont lourds en terme calculatoire.
Dans la partie qui suit, nous allons faire une comparaison entre l'approche MPC réglée par RNA robustifiée par la paramétrisation de Youla et celle de la GPC robustifiée par la même approche et on ferra également une comparaison en performances afin de montrer l'intérêt de la robustification.

4.3.2 Comparaison entre la MPC réglée par RNA et la GPC avec robustification

L'objectif de cette partie est de montrer l'impact du réglage de la commande prédictive en effectuant une comparaison entre : l'approche proposée pour le réglage de la MPC basée sur un RNA robustifiée et l'approche GPC réglée par une méthode heuristique [152]. L'objectif est d'injecter le paramètre Q de la robustificiation dans la boucle de régulation pour voir son influence sur le rejet de perturbation et bruit de mesure.
Cette comparaison regroupe trois approches :

- L'approche de (Pedro et al,), en 2003 sera appelée (Pedro 2003) [152] : une GPC réglée par une méthode heuristique et robustifiée par l'approche de Youla.
- L'approche proposée [80] : une MPC réglée par les RNA et robustifiée par l'approche de Youla.
- L'approche GPC initiale : une GPC réglée par une méthode heuristique sans robustification.

Le système considéré en simulation est le même que celui représenté par l'équation 4.22 défini précédemment.
Les paramètres de réglage utilisés en simulation sont les mêmes que ceux utilisés dans la partie précédente, (pour la GPC avec robustification par [80] (Pedro 2003)) : $N_P = 5$, $N_u = 1$ et $\lambda = 1$, (pour la MPC réglée RNA (Approche proposée)) : $N_P = 23$, $N_c = 12$ et $\lambda = 4,7918$.
Les Figures 4.17 et 4.18 représentent respectivement les sorties et les signaux de commande du système défini par l'équation 4.22 en présence d'une perturbation à l'instant $t = 7s$ et d'un bruit de mesure à l'instant $t = 14s$.
Les trois sorties présentées sur les Figures représentent respectivement une GPC sans robustification, une GPC avec robustification (Pedro 2003) et l'approche proposée avec robustification. L'objectif derrière cette simulation est de montrer l'influence de la robustification sur le rejet de perturbations en boucle fermée.
La comparaison entre la MPC réglée par RNA avec et sans robustification sera présentée

dans la partie suivante. Le Tableau 4.2 dresse une comparaison en performances des trois

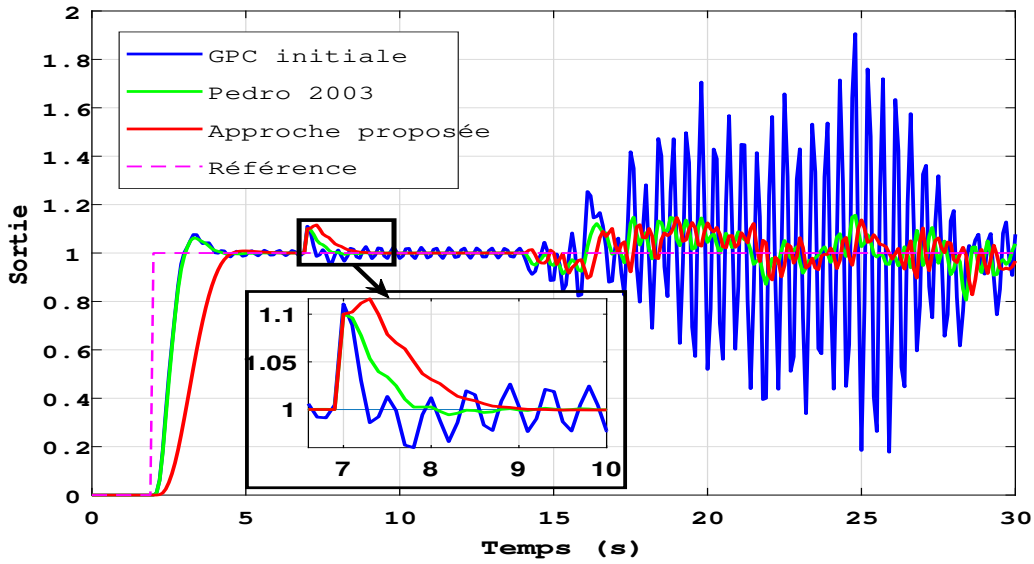


Figure 4.17 – Sorties du système avec robustification

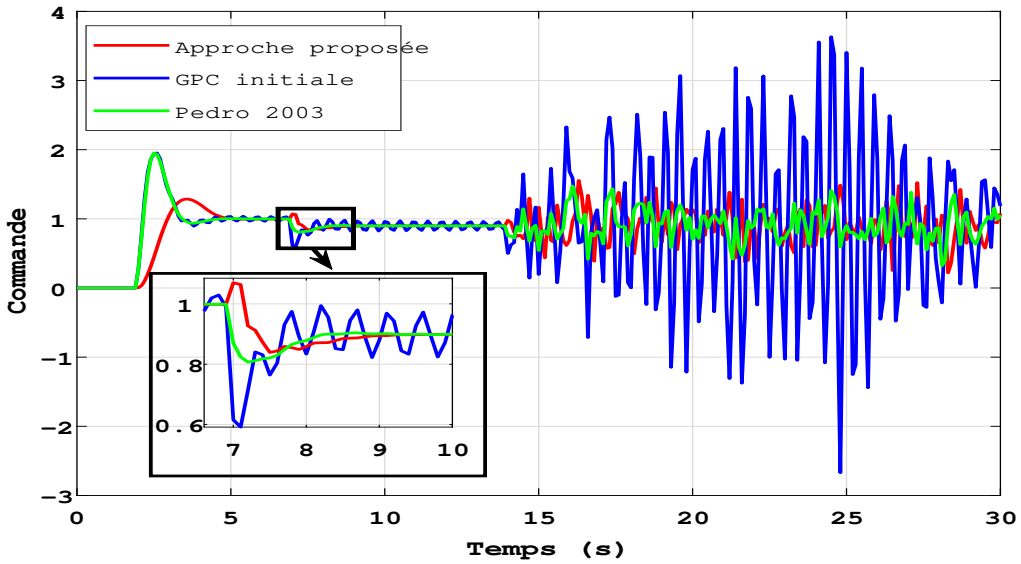


Figure 4.18 – Signaux de commandes avec robustification

approches.

Les indicateur de performances considérés sont : le temps de montée, le temps de réponse et le dépassement, comme déjà présentés dans le chapitre précédent auxquels sont ajoutés deux indicateurs supplémentaires liés au rejet de perturbations sont considérées :

— Le temps d'annulation ou d'atténuation de perturbation (T_{ap}),

— L'énergie de commande maximale ($|U_{max}|$)

Tableau 4.2 – Comparaison des performances

	GPC initiale	Pedro 2003	Approche proposée
RT (s)	**0.6**	**0.6**	1.3
ST (s)	3.8	**3.6**	**3.6**
OV (%)	6.3	6.3	**0.9**
T_{ap}(s)	Perturbation non atténuée	2.6	**2.1**
$\|U_{max}\|$	3.633	1.9608	**1.284**

Dans le Tableau 4.2, on observe qu'avec l'approche par RNA avec robustification de meilleures performances sont obtenues par comparaison avec celles obtenues par les deux autres approches : Le dépassement est quasi nul et un temps nécessaire pour annuler la perturbation moins élevé. En termes de rapidité, le temps de réponse est quasi similaire pour les deux approches (Pedro 2003) et l'approche proposée [82].
Dans la partie suivante, une comparaison est effectuée entre la MPC réglée RNA sans sans robustification et avec robustification par la paramétrisation de Youla.

4.3.3 Comparaison entre la MPC réglée par RNA avec et sans robustification

Afin d'observer l'influence du réglage paramétrique de la MPC sur le rejet des perturbations ainsi que l'apport de la robustification, une comparaison dans cette partie est effectuée entre la MPC réglée par RNA sans robustification et celle robustifiée selon l'approche de paramétrisation de Youla.
En effet, le but est de montrer qu'un couple des paramètres optimaux (N_c, N_p, λ) permet de rejeter les perturbations sans avoir recours à des outils de robustification.
Le système considéré dans cette comparaison est celui présenté dans l'équation 4.22.
Les paramètres de réglage utilisés en simulation sont les mêmes pour les deux approches : $N_P = 23$, $N_c = 12$ et $\lambda = 4,7918$.
Les sorties et les signaux de commande avec et sans robustification en présence d'une perturbation à l'instant $t = 7s$ et d'un bruit de mesure à l'instant $t = 14s$ sont représentés respectivement dans les Figures 4.19 et 4.20 :

Le tableau 4.3 regroupe les résultats obtenus :
Des performances similaires sont observées pour la MPC réglée par RNA qu'elle soit robustifiée ou non.
A partir des résultats obtenus, on peut tirer la conclusion suivante : l'apport de la robustesse par paramétrisation de Youla sur l'approche proposée n'est pas significatif.
En plus d'avoir les mêmes résultats, l'approche proposée sans robustification ne demande

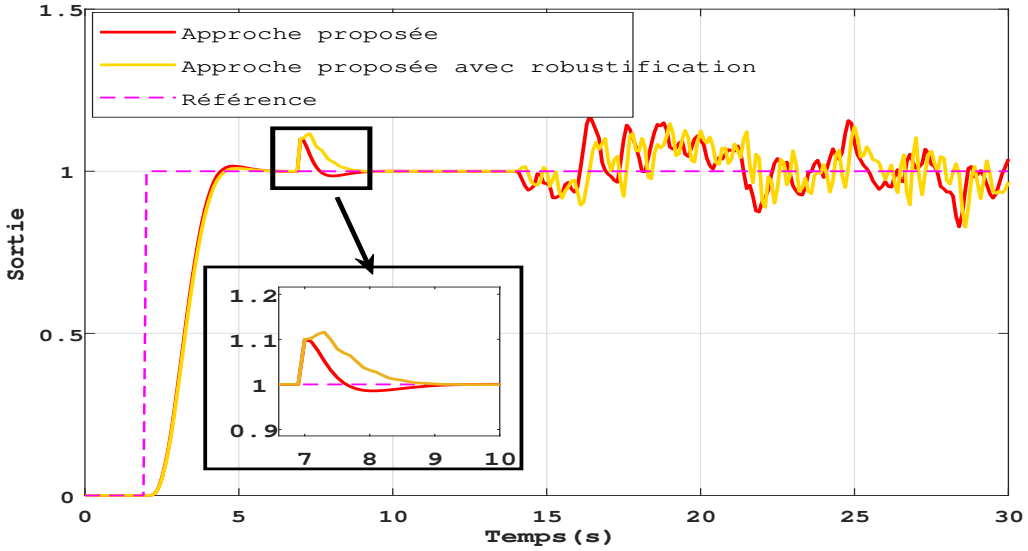


Figure 4.19 – Sorties avec et sans robustification de la MPC en fonction du temps

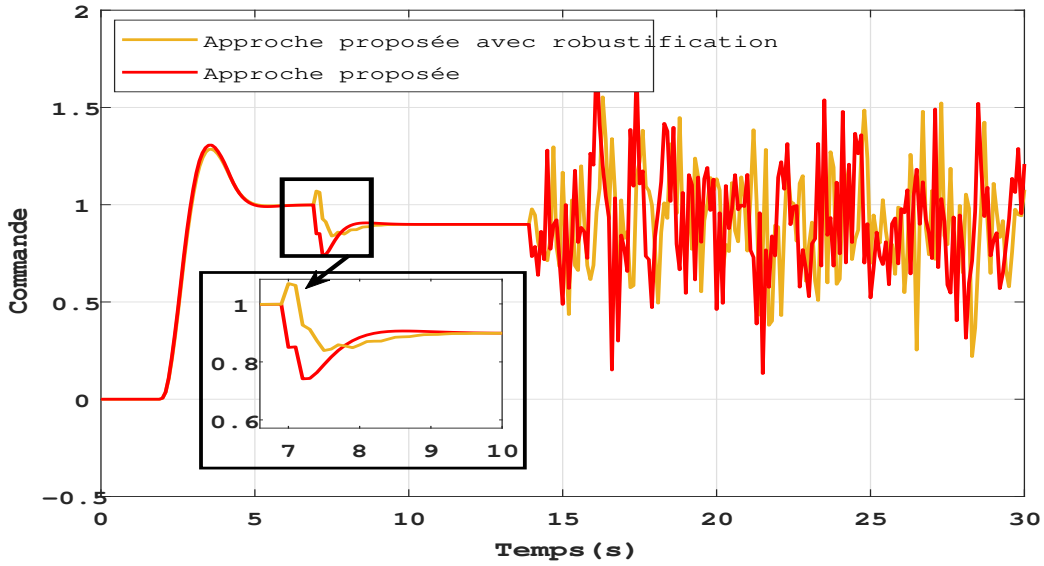


Figure 4.20 – Signaux de commande avec et sans robustification de la MPC en fonction du temps

pas un temps de calcul élevé alors que l'apport de la robustification (calcul du paramètre Q comme montré dans la première partie du chapitre) est généralement lourd en termes calculatoires spécialement sur les systèmes multivariables .

Donc un bon réglage du couple des paramètres (N_c, N_p, λ) de la MPC dans le cas du système étudié permet le rejet de perturbation sans rajout de robustification dans la boucle de régulation.

Tableau 4.3 – Comparaison des performances

	Approche proposée avec robustification	Approche proposée sans robustification
RT (s)	2.8	2.8
ST (s)	4.2	4.2
OV (%)	0.9	0.9
T_{ap}(s)	2.2	2.1
$\|U_{max}\|$	1.660	1.584

4.4 Conclusion

Dans ce chapitre, la robustesse de la commande prédictive en présence d'une perturbation et d'un bruit de mesure a été étudiée.
Dans un premier temps, un état de l'art a été effectué sur les différentes approches de robustification. Ces approches sont regroupées en deux grandes catégories :

— le formalisme basé sur la représentation polynomiale,

— le formalisme basé sur la représentation d'état.

Dans un deuxième temps, une étude comparative des différentes approches en simulation a permis d'extraire la méthode la plus efficace sur la classe de systèmes considérée.
Le choix s'est donc porté sur l'approche par paramétrisation de Youla, cette dernière reposant sur le rajout d'un paramètre Q dans la boucle de régulation afin de robustifier le système face aux différents types de perturbations.
Finalement et dans un troisième temps, une comparative a été effectuée sur trois axes :

— Comparaison entre la MPC réglée par RNA et une GPC sans robustification

— Comparaison entre la MPC réglée par RNA et la GPC avec robustification

— Comparaison entre la MPC réglée par RNA avec et sans robustification

Ces différentes comparaisons ont permis de montrer l'influence des paramètres de réglages sur la robustesse en performances de la commande prédictive en présence de perturbations.
Dans le prochain et dernier chapitre de ce mémoire, l'approche de réglage de la MPC basée sur les RNA sera appliquée sur un système de drone MIMO non linéaire. Pour ce faire, une approche de linéarisation des systèmes multivariables non linéaires sera appliquée. L'objectif sera de montrer que l'approche de réglage par RNA proposée peut être appliquée aux classes de systèmes SISO retardés ou non, MIMO retardés ou non, et finalement les systèmes MIMO non linéaires à plusieurs degrés de libertés.

Chapitre 5

Application de la MPC réglée par RNA sur un drone QTW

Sommaire

5.1 Introduction 103

5.2 Drone convertible "QTW" 103

5.2.1 Structure et modes de vol 103

5.2.2 Modélisation dynamique des drones QTW 104

5.3 Modélisation T-S des systèmes non linéaires 106

5.3.1 Représentation d'état des modèles T-S 107

5.3.2 Obtention du modèle T-S 108

5.3.3 Stabilité des systèmes T-S 109

5.4 Modélisation T-S du drone QTW en vol vertical 111

5.4.1 Construction du modèle T-S du drone convertible en mode de vol vertical 111

5.4.2 Validation du modèle 113

5.5 Application de la MPC réglée par RNA au drone QTW . . . 116

5.6 Conclusion 122

5.1 Introduction

Dans le chapitre précédent, une application de l'approche de réglage de la MPC basée sur les RNA sur des systèmes SISO en présence de perturbations a été présentée. Le but était de montrer que l'approche proposée permet le rejet de ces perturbations sans avoir recours à des outils de robustification lourds en termes calculatoires.
L'objectif dans ce chapitre est de montrer que l'approche de réglage de la commande prédictive par RNA peut être étendue et appliquée aux systèmes MIMO non linéaires.
Ainsi, le système considéré dans ce chapitre sera un drone convertible QTW (Quad-Tilt Wing) non linéaire. En effet, la problématique du contrôle des drones est toujours d'actualité et revient donc à synthétiser une loi de commande qui assure un suivi de trajectoire des signaux de références.
Dans un premier temps, une présentation des drones QTW sera établie, en commençant par leur structure mécanique leurs modes du vol, et leur modélisation dynamique. Dans un deuxième temps, nous établirons le modèle le "Takagi- Sugeno" (T-S) de ce type de drone. Cette technique permet d'écrire un système non linéaire sous forme de plusieurs systèmes linéaires ou quasi linéaires représentant le même comportement du système initial.
Dans un troisième et dernier temps avant de conclure ce chapitre, une application sur un drone QTW sera présentée afin de valider la stratégie développée pour le réglage de la MPC.

5.2 Drone convertible "QTW"

Un drone quadrirotor est défini par une voilure tournante avec quatre rotors pour en assurer la portance [132] . Le système QTW considéré est équipé de quatre ailes, deux à l'arrière et deux à l'avant du châssis. Ces ailes peuvent être tournées de la position horizontale à la position verticale et vice versa.

5.2.1 Structure et modes de vol

Dans la littérature, on peut distinguer trois mouvements du drone QTW en vol : le roulis, le tangage et le lacet [37].
De ce fait, les modes de fonctionnement de vol des drones QTW sont définis par [25] :

- Vol vertical : ce type de vol repose sur les rotors du drone, il correspond au décollage et atterrissage vertical.
- Vol stationnaire. Il correspond à un mode de vol intermédiaire qui relie les modes vertical et horizontal.

— Vol horizontal : dans ce mode de vol, le drone a le même comportement qu'un avion où les angles d'inclinaison des rotors et des ailes sont quasi-nuls.

La Figure 5.1 représente l'architecture du drone QTW dans ses différents modes de vol [25] :

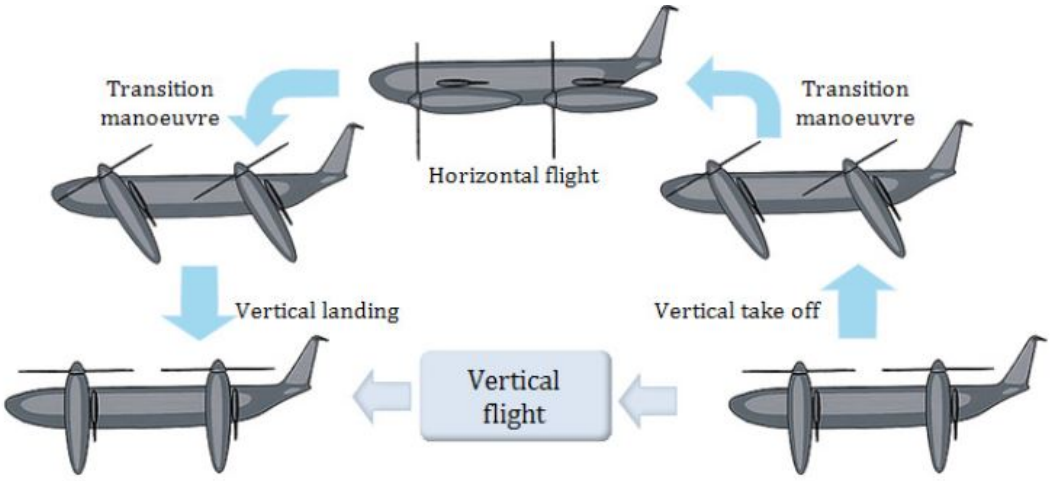


Figure 5.1 – Les modes de vols du drone QTW [37]

5.2.2 Modélisation dynamique des drones QTW

Afin d'établir le modèle dynamique non linéaire du drone, on considère le formalisme de Newton-Euler établi par (E. Cetinso et al,) [37]. Cette étude est simplifiée en se basant sur plusieurs hypothèses : la masse de l'aéronef est constante, la force de traînée pour le fuselage est négligée et la considération que le foyer aérodynamique et le centre de gravité sont juxtaposés.

La configuration du drone considéré est donnée par la Figure 5.2 :

Dans cette figure on distingue deux repères :

— Le repère inertiel fixe, noté $\boldsymbol{R}_i\,(O_i, x_i, y_i, z_i)$, supposé galiléen avec z_i pointé vers le centre terrestre.

— Le repère lié au corps du véhicule, noté $\boldsymbol{R}_b\,(O_b, x_b, y_b, z_b)$, dont l'origine coïncide avec le centre de gravité et où x_b est l'axe longitudinal, y_b l'axe transversal et z_b l'axe vertical.

Les trois angles d'Euler considérés et relatifs aux mouvements sont respectivement le roulis, le tangage et le lacet définis par $-\pi/2 \leq \phi \leq \pi/2$, $-\pi/2 \leq \theta \leq \pi/2$ et $-\pi \leq \psi \leq \pi$.

Les dynamiques de position et d'attitude sont exprimées dans le repère $\boldsymbol{R}_b$. Avec l'utilisation de la formulation de Newton-Euler, la dynamique du drone est décrite par

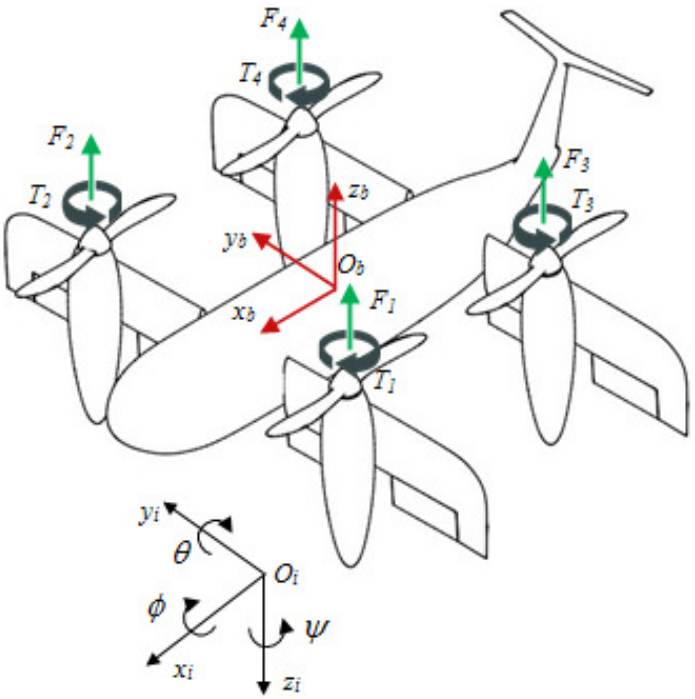


Figure 5.2 – Configuration du drone QTW

l'équation 5.1 :

$$\begin{pmatrix} m\boldsymbol{I} & 0 \\ 0 & \boldsymbol{J} \end{pmatrix} \begin{pmatrix} \dot{\boldsymbol{\nu}} \\ \dot{\boldsymbol{\vartheta}} \end{pmatrix} + \begin{pmatrix} 0 \\ \boldsymbol{\vartheta} \wedge \boldsymbol{J}\boldsymbol{\vartheta} \end{pmatrix} = \begin{pmatrix} \boldsymbol{F}_t \\ \boldsymbol{M}_t \end{pmatrix} \tag{5.1}$$

où $\boldsymbol{I}$ est la matrice identité, m la masse du drone QTW, $\boldsymbol{J}$ la matrice d'inertie et où $\boldsymbol{F}_t$ et $\boldsymbol{M}_t$ représentent respectivement les forces extérieures et les moments agissant sur le drone dans le repère $\boldsymbol{R}_b$.

En utilisant les équations de mouvement par le formalisme de Newton-Euler, le modèle dynamique complet du drone convertible QTW est défini par [12] :

$$m\ddot{\boldsymbol{\xi}} = \boldsymbol{F}_t \tag{5.2}$$

$$\boldsymbol{J}\dot{\boldsymbol{\vartheta}} = -\boldsymbol{M}_t + \boldsymbol{\vartheta} \wedge \boldsymbol{J}\boldsymbol{\vartheta} \tag{5.3}$$

où $\boldsymbol{J} = diag\,(J_{xx}, J_{yy}, J_{zz})$ est la matrice d'inertie du robot et J_{xx}, J_{yy} et J_{zz} représentent les inerties respectivement selon les axes x, y et z.

Le modèle dynamique global suivant donné par l'équation 5.4 se compose des deux mouvements de rotation et translation [12] et [110]. Ce modèle est établi à partir des équations 5.2 et 5.3 :

$$\begin{cases} \ddot{x} = \dfrac{1}{m}\left(\left(cos\psi cos\theta cos\gamma - \left(cos\phi sin\theta cos\psi + sin\phi sin\psi\right) sin\gamma\right) u_1 + F_{a,x}\right) \\ \ddot{y} = \dfrac{1}{m}\left(\left(sin\psi cos\theta cos\gamma - \left(cos\phi sin\theta sin\psi - sin\phi cos\psi\right) sin\gamma\right) u_1 + F_{a,y}\right) \\ \ddot{z} = \dfrac{1}{m}\left(\left(-sin\theta cos\gamma - cos\phi cos\theta sin\gamma\right) u_1 + F_{a,z}\right) + g \\ \dot{p} = \dfrac{u_2}{J_{xx}} + \dfrac{J_{yy} - J_{zz}}{J_{xx}} qr - \dfrac{J_r}{J_{xx}} q\omega_r s\gamma + M_\phi \\ \dot{q} = \dfrac{u_3}{J_{yy}} + \dfrac{J_{zz} - J_{xx}}{J_{yy}} pr + \dfrac{J_r}{J_{yy}} \left(ps\gamma + rc\gamma\right) \omega_r + M_\theta \\ \dot{r} = \dfrac{u_4}{J_{zz}} + \dfrac{J_{xx} - J_{yy}}{J_{zz}} pq - \dfrac{J_r}{J_{zz}} qc\gamma\omega_r + M_\psi \end{cases} \tag{5.4}$$

où $\omega_r = \omega_1 - \omega_2 - \omega_3 + \omega_4$ est la vitesse angulaire résiduelle des rotors et u_1, u_2, u_3 et u_4 les entrées de commande du drone.

Par la suite, le mode considéré sera celui du vol vertical pour simplifier le calcul. Dans ce mode de vol, le drone convertible QTW se comporte comme un quadri-rotor classique où les moments et forces aérodynamiques sont négligés. L'équation 5.4 est donc écrite sous la forme suivante 5.5 :

$$\begin{cases} \ddot{x} = \dfrac{1}{m}\left(-cos\phi sin\theta cos\psi - sin\phi sin\psi\right) u_1 \\ \ddot{y} = \dfrac{1}{m}\left(-cos\phi sin\theta sin\psi + sin\phi cos\psi\right) u_1 \\ \ddot{z} = -\dfrac{1}{m} cos\phi cos\theta u_1 + g \\ \dot{p} = \dfrac{u_2}{J_{xx}} + \dfrac{J_{yy} - J_{zz}}{J_{xx}} qr - \dfrac{J_r}{J_{xx}} q\omega_r \\ \dot{q} = \dfrac{u_3}{J_{yy}} + \dfrac{J_{zz} - J_{xx}}{J_{yy}} pr + \dfrac{J_r}{J_{yy}} p\omega_r \\ \dot{r} = \dfrac{u_4}{J_{zz}} + \dfrac{J_{xx} - J_{yy}}{J_{zz}} pq \end{cases} \tag{5.5}$$

Dans la partie qui suit, ce modèle QTW sera linéarisé en utilisant l'approche par modélisation T-S.

5.3 Modélisation T-S des systèmes non linéaires

La modélisation Takagi-Sugeno dite "T-S" permet de représenter un système non linéaire sous la forme de plusieurs sous-modèles linéaires [158], [140]. Ce passage est fait au moyen d'une linéarisation autour des points de fonctionnement [94] ou en appliquant l'approche par secteurs non linéaires [128]

L'approche T-S permet la prise en compte du comportement non linéaire des modèles

dynamiques par des modèles locaux dans les différentes zones de fonctionnement. Le modèle non linéaire est représenté sous la forme d'une interpolation par des modèles (LTI) pondérés avec des « fonctions d'appartenance » [156] et [179]

5.3.1 Représentation d'état des modèles T-S

Un système non linéaire peut être représenté sous forme d'un modèle d'état à paramètres variants donné par l'équation 5.6 :

$$\begin{aligned} \dot{x}(t) &= A(\theta)\, x(t) + B(\theta)\, u(t) \\ \dot{y}(t) &= C(\theta)\, x(t) \end{aligned} \tag{5.6}$$

où A, B et C représentent les matrices du système et θ le vecteur des variables dîtes de prémisse. Ces variables dépendent à la fois des entrées de commande u et du vecteur d'état x du système [84], [104].
Le vecteur θ (appelé aussi vecteur des variables de décision) doit être borné :

$$\theta^{min} \leq \theta \leq \theta^{max} \tag{5.7}$$

où θ^{min} et θ^{max} représentent respectivement les valeurs minimale et maximale du vecteur de décision. Le modèle est donc obtenu à travers un nombre fini de sous-modèles LTI définis par :

$$\begin{cases} \dot{x}(t) = & \sum_{i=1}^{N_{lti}} \mu_i(\theta)\Big(A_i\left(x\left(t\right)\right) x(t) + B_i\left(x\left(t\right)\right) u(t)\Big) \\ y(t) = & \sum_{i=1}^{N_{lti}} \mu_i(\theta) C_i\left(x\left(t\right)\right) x(t) \end{cases}$$

où N_{lti} est le nombre de sous-modèles linéaires, $A_i \in \mathbb{R}^{n_x \times n_x}$, $B_i \in \mathbb{R}^{n_u \times n_x}$ et $C_i \in \mathbb{R}^{n_y \times n_x}$ des matrices invariantes, et $\mu_i(\theta)$ représentent les fonction d'activation du $i^{\text{éme}}$ sous-modèle associé.
Les fonctions d'activation $\mu_i(\theta)$ sont non linéaires, et peuvent être de forme triangulaire, sinusoïdale ou gaussienne. Ces fonctions doivent respecter la propriété de convexité suivante :

$$\sum_{i=1}^{N_{lti}} \mu_i\left(\theta\right) = 1 \tag{5.8}$$

avec

$$0 \leq \mu_i\left(\theta\right) \forall\ i = 1,\ 2, ...,\ N_{lti}$$

Les fonctions μ_i dépendent du vecteur des variables de prémisse $\theta\left(t\right) = [\theta_1\left(t\right), \theta_2\left(t\right), ..., \theta_p\left(t\right)]$ et d'un opérateur de logique. Dans la modélisation T-S, l'opérateur

considéré est le "*Et*" comme étant le produit défini par l'équation 5.9 :

$$\mu_i\left(\theta\left(t\right)\right) = \prod_{j=1}^{p} F_j^i\left(\theta_j\left(t\right)\right) \tag{5.9}$$

où F_j^i désigne, pour $j \in \{1, 2, ...p\}$ les fonctions d'appartenance des ensembles flous et $\mu_i\left(\theta\left(t\right)\right) > 0, \forall\ t \geqslant 0$.
La normalisation des sommes pondérées permet d'écrire le modèle T-S général du système représenté précédemment par 5.6 sous la forme définie par l'équation 5.10 [86] :

$$\begin{cases} \dot{x}\left(t\right) = \sum_{i=1}^{N_{lti}} \mu_i\left(\theta\left(t\right)\right)\left(A_i\left(x\left(t\right)\right)x\left(t\right) + B_i\left(x\left(t\right)\right)u\left(t\right)\right) \\ \dot{y}\left(t\right) = \sum_{i=1}^{N_{lti}} \mu_i\left(\theta\left(t\right)\right)\left(C_i\left(x\left(t\right)\right)x\left(t\right)\right), \quad i \in 1, ..., N_{lti} \end{cases} \tag{5.10}$$

5.3.2 Obtention du modèle T-S

Il existe plusieurs approches dans la littérature qui permettent d'obtenir le modèle T-S d'un système non linéaire [129], [22].
Nous avons choisi d'utiliser l'approche par Multi-modèles de secteurs non linéaires [171]. La particularité de cette approche est d'aboutir à une structure T-S qui décrit parfaitement la dynamique du système étudié. Son avantage réside dans la minimisation de l'erreur d'approximation en réduisant le nombre de modèles locaux [179].
Le principe de construction est basé sur la Transformation Polytopique Convexe (TPC) des termes non linéaires [157] dont la définition est la suivante :

Soit $\theta_j(x, u)$ pour $j \in \{1, 2, ..., p\}$, l'ensemble des fonctions non linéaires sur un domaine compact $D \subset \mathbb{R}^{n_1} \times \mathbb{R}^{m_1}$ à valeurs dans $\mathbb{R}$ avec $x \in \mathbb{R}^{n_1}$ et $u \in \mathbb{R}^{m_1}$. Il existe deux fonctions (v=[1,2]) telles que :

$$\mu_v : D \mapsto [0, 1] \tag{5.11}$$

$$(x, u) \mapsto \mu_v(x, u) \tag{5.12}$$

Les fonctions $\theta_j(x, u)$ peuvent être formulées sous la forme convexe suivante :

$$\theta(x, u) = \mu_1(x, u)\theta^{max} + \mu_2(x, u)\theta^{min} \tag{5.13}$$

où :

$$\begin{cases} \theta^{max} = \max\limits_{(x,u)\in D}\{\theta(x,u)\} \\ \theta^{min} = \min\limits_{(x,u)\in D}\{\theta(x,u)\} \end{cases} \tag{5.14}$$

avec :

$$\mu_1(x,u) + \mu_2(x,u) = 1 \tag{5.15}$$

Les fonctions de pondération μ_1 et μ_2 sont déterminées par :

$$\mu_1 = \frac{\theta(x,u) - \theta^{min}}{\theta^{max} - \theta^{min}};\ \ \mu_2 = 1 - \mu_1 = \frac{\theta^{max} - \theta(x,u)}{\theta^{max} - \theta^{min}} \tag{5.16}$$

Selon cette approche de construction des modèles T-S, le nombre de modèles locaux linéaires obtenus est $N_{lti} = 2^r$, où r est un entier naturel représentant le nombre de variables de prémisse [8], [128]. Donc, on peut conclure également que le nombre de règles floues et de termes non linéaires sont liés de manière exponentielle.
Dans ce qui suit nous aborderons la stabilité des systèmes modélisés par l'approche Takagi-Sugeno.

5.3.3 Stabilité des systèmes T-S

Dans la littérature, il existe plusieurs études sur la stabilité des modèles établis selon l'approche T-S [88], [52], [24] . L'une des techniques la plus utilisée est celle basée sur la théorie de Lyapunov. Afin d'assurer la stabilité des systèmes T-S et de garantir des conditions suffisantes, la formulation des Inégalités Matricielles Linéaires (LMI) sera employée. Cette formulation est définie par le théorème suivant : [156] :
Soit un modèle T-S autonome ($u(t) = 0$) donné par :

$$\dot{x}(t) = \sum_{i=1}^{N_{lti}} \mu_i(\theta)\Big(A_i x(t)\Big) \tag{5.17}$$

Le système (5.17) est asymptotiquement stable si et seulement s'il existe une matrice P telle que $P = P^T > 0$ vérifiant les LMIs suivantes pour $i \in \{1, 2, \cdots, N_{lti}\}$:

$$A_i^T P + P A_i < 0 \tag{5.18}$$

□

Soit la fonction de Lyapunov suivante :

$$V(x(t)) = x(t)^T P x(t) \tag{5.19}$$

avec $P = P^T > 0$.

Le système (5.10) est stable si :

$$\dot{V}(x(t)) = \dot{x}^T(t)Px(t) + x^T(t)P\dot{x}(t) < 0 \tag{5.20}$$

Cette condition est équivalente à :

$$\dot{V}(x(t)) = \dot{x}^T(t)\Big(\sum_{i=1}^{N_{lti}} \mu_i(\theta(t))(A_i^T P + PA_i)\Big)x(t) < 0 \tag{5.21}$$

L'interconnexion des sous-modèles stables n'est pas toujours stable. En effet avec un nombre de sous-modèles élevé, la détermination de la matrice P qui satisfait toutes les N_{lti} LMIs reste une problématique scientifique d'actualité. D'après (Takagi et al,), l'utilisation des fonction de Lyapunov polyquadratiques données par (5.22) permet de résoudre ce problème avec un minimum de restriction [158], en calculant plusieurs matrices P_i à la place d'une seule :

$$\dot{V}(x(t)) = x(t)^T \sum_{i=1}^{N_{lti}} \mu_i(\theta(t))\Big(P_i x(t)\Big) \tag{5.22}$$

Par conséquent, une fonction candidate de Lyapunov non quadratique a été introduite et définie par l'équation 5.23 :

$$V(x(t)) = \max\{V_1(x(t)), V_2(x(t)), \cdots, V_{N_{lti}}(x(t))\} \tag{5.23}$$

avec :

$$V_i(x(t)) = x(t)^T P_i x(t), \quad P_i = P_i^T > 0, \text{pour } i = \{1, 2, \cdots, N_{lti}\} \tag{5.24}$$

Le Théorème suivant précise les conditions de stabilité des fonctions de Lyapunov particulières :

Le modèle T-S autonome (5.17) est stable s'il existe des matrices symétriques et définies positives P_j et des scalaires strictement positifs τ_{ijk} tels que :

$$A_i^T P_j + P_j A_i + \sum_{k=1}^{N_{lti}} \tau_{ijk}(P_j - P_k) < 0 \tag{5.25}$$

avec $\begin{cases} P_j > 0 \\ \tau_{ijk} > 0 \end{cases}$ et $i, j = \{1, 2, \cdots, N_{lti}\}$.

□

Ces équations qui permettent de conclure sur les conditions de stabilité sont moins

restrictives mais la résolution du problème est plus compliqué car l'introduction de ces fonctions quadratiques conduit à l'apparition d'inégalités matricielles unilinéaires [158].

5.4 Modélisation T-S du drone QTW en vol vertical

Afin de simplifier la problématique du contrôle d'un drone convertible QTW, nous allons considérer un modèle simplifié qui traite le mouvement de rotation du drone en mode vertical.

5.4.1 Construction du modèle T-S du drone convertible en mode de vol vertical

Le modèle dynamique des mouvements de rotation en mode vertical est défini par le système d'équations 5.26 [26] :

$$\begin{cases} \ddot{\phi} = \dfrac{\tau_\phi}{I_{xx}} + \dfrac{I_{yy} - I_{zz}}{I_{xx}}\dot{\theta}\dot{\psi} \\ \ddot{\theta} = \dfrac{\tau_\theta}{I_{yy}} + \dfrac{I_{zz} - I_{xx}}{I_{yy}}\dot{\phi}\dot{\psi} \\ \ddot{\psi} = \dfrac{\tau_\psi}{I_{zz}} + \dfrac{I_{xx} - I_{yy}}{I_{zz}}\dot{\phi}\dot{\theta} \end{cases} \tag{5.26}$$

où ϕ, θ et ψ sont respectivement les angles d'attitude du drone (roulis, tangage et lacet), I_{xx}, I_{yy} et I_{zz} désignent les différents moments d'inertie autour des axes x, y et z, et l la distance entre les rotors et le centre de gravité. Les signaux de commande associés au système $u_{2;3;4}$ sont explicitement donnés par l'équation 5.27 :

$$\begin{bmatrix} \tau_\phi \\ \tau_\theta \\ \tau_\psi \end{bmatrix} = \begin{bmatrix} kl & -kl & kl & -kl \\ kl & kl & -kl & -kl \\ k\lambda & -k\lambda & -k\lambda & k\lambda \end{bmatrix} \begin{bmatrix} \omega_1^2 \\ \omega_2^2 \\ \omega_3^2 \\ \omega_4^2 \end{bmatrix} \tag{5.27}$$

où k représente la portance, λ le rapport entre couple et force et ω_i la vitesse angulaire du i-ème moteur. La représentation d'état du système est donnée par :

$$\begin{cases} \dot{x} = A(x)x + Bu \\ y = Cx \end{cases} \tag{5.28}$$

avec $x = \begin{bmatrix} \phi & \dot{\phi} & \theta & \dot{\theta} & \psi & \dot{\psi} \end{bmatrix}^T$ le vecteur d'état et $u = \tau_{\phi;\theta;\psi}$, les entrées de

commandes.
Les matrices d'état sont définies par :

$$A(x) = \begin{bmatrix} 0 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & a_1\dot{\theta} \\ 0 & 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & a_2\dot{\phi} \\ 0 & 0 & 0 & 0 & 0 & 1 \\ 0 & 0 & 0 & a_3\dot{\theta} & 0 & 0 \end{bmatrix}; B = \begin{bmatrix} 0 & 0 & 0 \\ b_1 & 0 & 0 \\ 0 & 0 & 0 \\ 0 & b_2 & 0 \\ 0 & 0 & 0 \\ 0 & 0 & b_3 \end{bmatrix}$$

avec :

$$a_1 = \frac{I_{yy} - I_{zz}}{I_{xx}}; a_2 = \frac{I_{zz} - I_{xx}}{I_{yy}}; a_3 = \frac{I_{xx} - I_{yy}}{I_{zz}}$$

et :

$$b_1 = \frac{1}{I_{xx}}; b_2 = \frac{1}{I_{yy}}; b_3 = \frac{1}{I_{zz}}$$

En observant la matrice d'état $A(x)$, on distingue deux non-linéarités dues à $\dot{\phi}$ et $\dot{\psi}$. Le vecteur de prémisse est donc donné par :

$$r\left(x\left(t\right)\right) = \left[\dot{\phi}, \dot{\psi}\right] \tag{5.29}$$

donc la variable de dimension est $p = 2$

Nous effectuons ensuite l'application de la transformation polytopique convexe, présentée précédemment, pour chaque variable de prémisse $z_i\left(x\left(t\right)\right)$. La réalisation est ensuite effectuée pour chacune des variables de prémisse [27]. La répartition en deux zones permet d'obtenir :

$$r_1\left(x\left(t\right)\right) = F_0^1\left(z_1\left(x\right)\right)\underline{r}_1 + F_1^1\left(r_1\left(x\right)\right)\bar{r}_1 \tag{5.30}$$

$$r_2\left(x\left(t\right)\right) = F_0^2\left(r_2\left(x\right)\right)\underline{r}_2 + F_1^2\left(r_2\left(x\right)\right)\bar{r}_2 \tag{5.31}$$

où :

$$\begin{cases} F_0^1 = \dfrac{\bar{r}_1 - r_1\left(x\left(t\right)\right)}{\bar{r}_1 - \underline{r}_1} \\ F_1^1 = \dfrac{r_1\left(x\left(t\right)\right) - \underline{z}_1}{\bar{r}_1 - \underline{r}_1} \\ F_0^2 = \dfrac{\bar{r}_2 - r_2\left(x\left(t\right)\right)}{\bar{r}_2 - \underline{r}_2} \\ F_1^2 = \dfrac{r_2\left(x\left(t\right)\right) - \underline{r}_2}{\bar{r}_2 - \underline{r}_2} \end{cases} \tag{5.32}$$

avec :

$$\bar{r}_p = \max\left\{r_p\left(x\left(t\right)\right)\right\} \quad \text{et} \quad \underline{r}_p = \min\left\{r_p\left(x\left(t\right)\right)\right\},\ \forall p = 1,2 \tag{5.33}$$

Les quatre combinaisons $\mu_1 = F_0^1 F_0^2$, $\mu_2 = F_0^1 F_1^2$, $\mu_3 = F_1^1 F_0^2$ et $\mu_4 = F_1^1 F_1^2$, permettent d'écrire le modèle non linéaire sous la forme de quatre sous-modèles linaires ($z = 2^p = 2^2$) en utilisant une inter-connexion par les différentes fonctions d'appartenance $\mu_i,\ i \in \{1,2,3,4\}$. Les matrices d'état des quatres sous-modèles sont données par :

$$A_1 = \begin{bmatrix} 0 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & a_1\underline{r}_2 \\ 0 & 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & a_2\underline{r}_1 \\ 0 & 0 & 0 & 0 & 0 & 1 \\ 0 & 0 & 0 & a_3\underline{r}_2 & 0 & 0 \end{bmatrix} \quad A_2 = \begin{bmatrix} 0 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & a_1\bar{r}_2 \\ 0 & 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & a_2\underline{r}_1 \\ 0 & 0 & 0 & 0 & 0 & 1 \\ 0 & 0 & 0 & a_3\bar{r}_2 & 0 & 0 \end{bmatrix}$$

$$A_3 = \begin{bmatrix} 0 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & a_1\underline{r}_2 \\ 0 & 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & a_2\bar{r}_1 \\ 0 & 0 & 0 & 0 & 0 & 1 \\ 0 & 0 & 0 & a_3\underline{r}_2 & 0 & 0 \end{bmatrix} \quad A_4 = \begin{bmatrix} 0 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & a_1\bar{r}_2 \\ 0 & 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & a_2\bar{r}_1 \\ 0 & 0 & 0 & 0 & 0 & 1 \\ 0 & 0 & 0 & a_3\bar{r}_2 & 0 & 0 \end{bmatrix}$$

tandis que la matrice B reste inchangée car elle ne présente pas de non-linéarité (constante).

Dans la partie suivante, la validation du modèle T-S obtenu sera abordée.

5.4.2 Validation du modèle

Dans la modélisation des systèmes dynamiques non linéaires, la validation du modèle T-S obtenu reste une étape clé avant l'application de la commande. Afin de pouvoir valider le modèle T-S développé dans le cadre de ce travail, nous avons considéré des consignes différentes. L'objectif est donc d'analyser les erreurs entre les sorties fournies par ces deux modèles.

Les Figures 5.3 , 5.5 et 5.7 affichent les trois angles de sortie des vols en attitude pour le modèle non linéaire et le modèle T-S. Ensuite, les trois Figures 5.4, 5.6, 5.8 présentent les erreurs de poursuite entre le modèle non linéaire et le modèle T-S.

On remarque que les deux modèles (T-S et NL) convergent vers les mêmes valeurs en régime permanent pour les trois sorties définies par les angles de roulis, de tangage et de lacet. On observe également que l'erreur est nulle en régime permanent avec une amplitude

quasi nulle. Sur la base des résultats obtenus pour les sorties et les erreurs de poursuites des différents angles, on peut conclure que le modèle T-S obtenu est représentatif du modèle non linéaire.

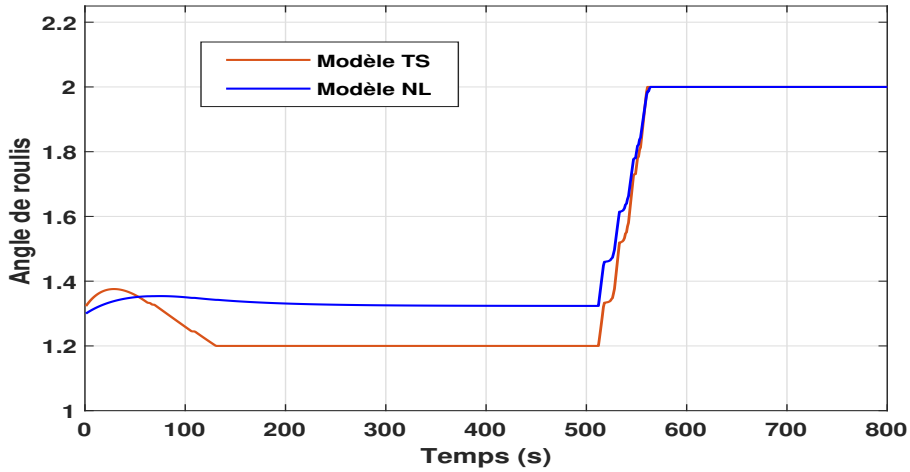


Figure 5.3 – Angle de roulis du modèle NL et du modèle TS

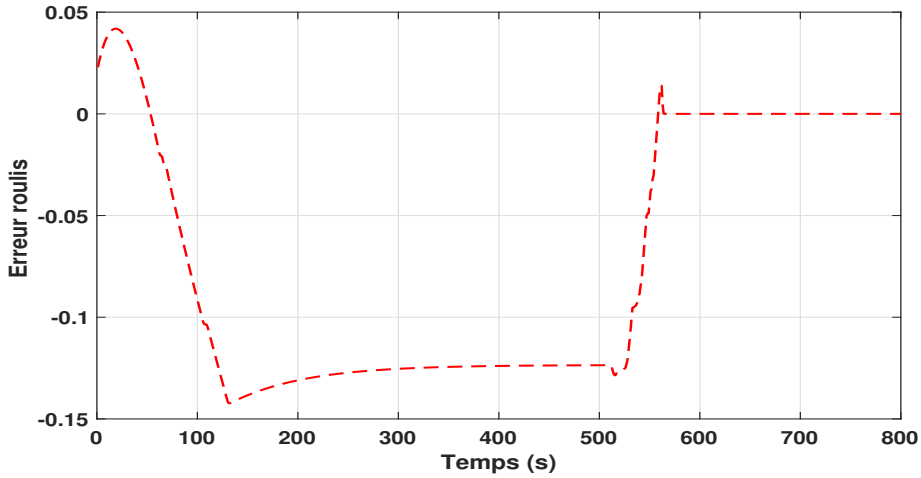


Figure 5.4 – Erreur sur l'angle de roulis

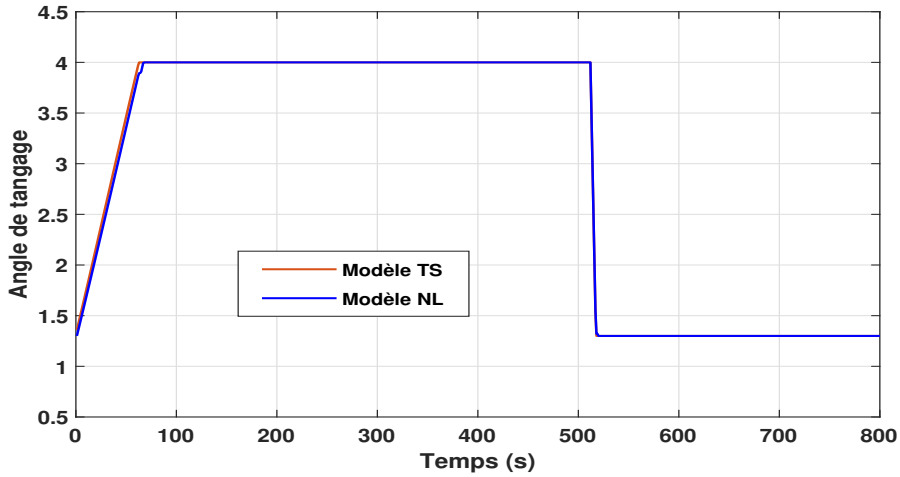


Figure 5.5 – Angle de tangage du modèle NL et du modèle TS

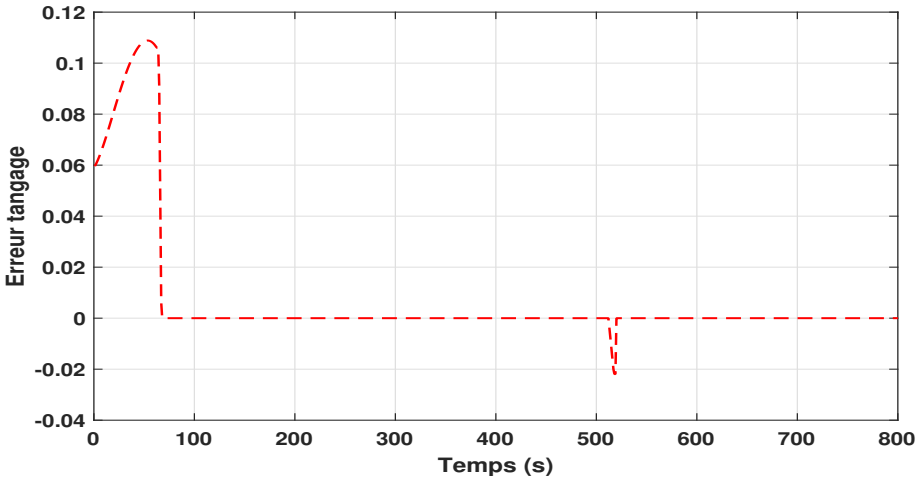


Figure 5.6 – Erreur sur l'angle de tangage

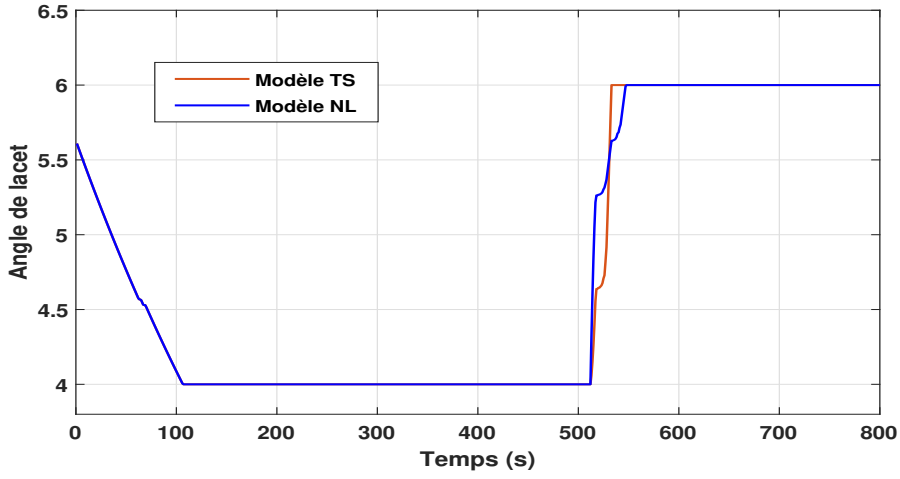


Figure 5.7 – Angle de lacet du modèle NL et du modèle TS

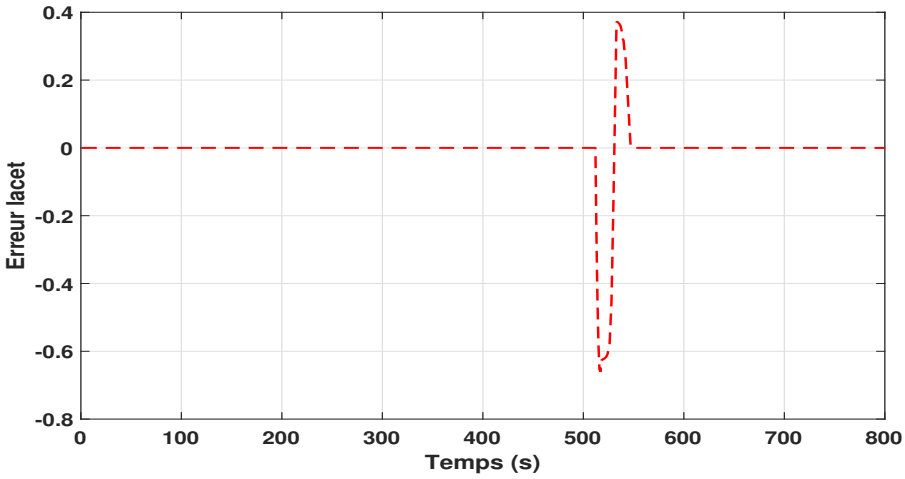


Figure 5.8 – Erreur sur l'angle de lacet

5.5 Application de la MPC réglée par RNA au drone QTW

Dans cette partie, un contrôleur MPC réglé par RNA sera synthétisé afin de contrôler le mouvement de rotation des trois angles du drone en mode de vol vertical.
La Figure 5.9 représente la stratégie de commande proposée qui s'appuie sur les quatre sous-modèles T-S pour le contrôle du drone :

Le calcul des paramètres de réglage des quatre contrôleurs prédictifs (MPC1, MPC2, MPC3, MPC4) est effectué au moyen de l'approche par RNA déjà présentée dans les chapitre précèdent.
Afin de montrer l'efficacité de notre approche, nous effectuerons une comparaison des résultats avec ceux obtenus selon la méthode de réglage analytique pour les systèmes MIMO non linéaires proposée par [165].
Les paramètres de réglage calculés selon les deux méthodes sont donnés dans les Tableaux 5.1 et 5.2, pour chacun des quatre sous-modèles obtenus du modèle T-S :

Tableau 5.1 – Valeurs des paramètres des contrôleurs prédictifs pour chaque sous modèle selon l'approche proposée

Paramètres	**Sous-modèle 1**	**Sous-modèle 2**	**Sous-modèle 3**	**Sous-modèle 4**
$\mathbf{N_c}$	11	17	7	10
$\mathbf{N_p}$	34	29	23	21
λ	5	11	10	8

Il est important de noter qu'au niveau des paramètres de réglage obtenus par RNA, l'horizon de prédiction est moins important avec l'approche neuronale et que donc le temps

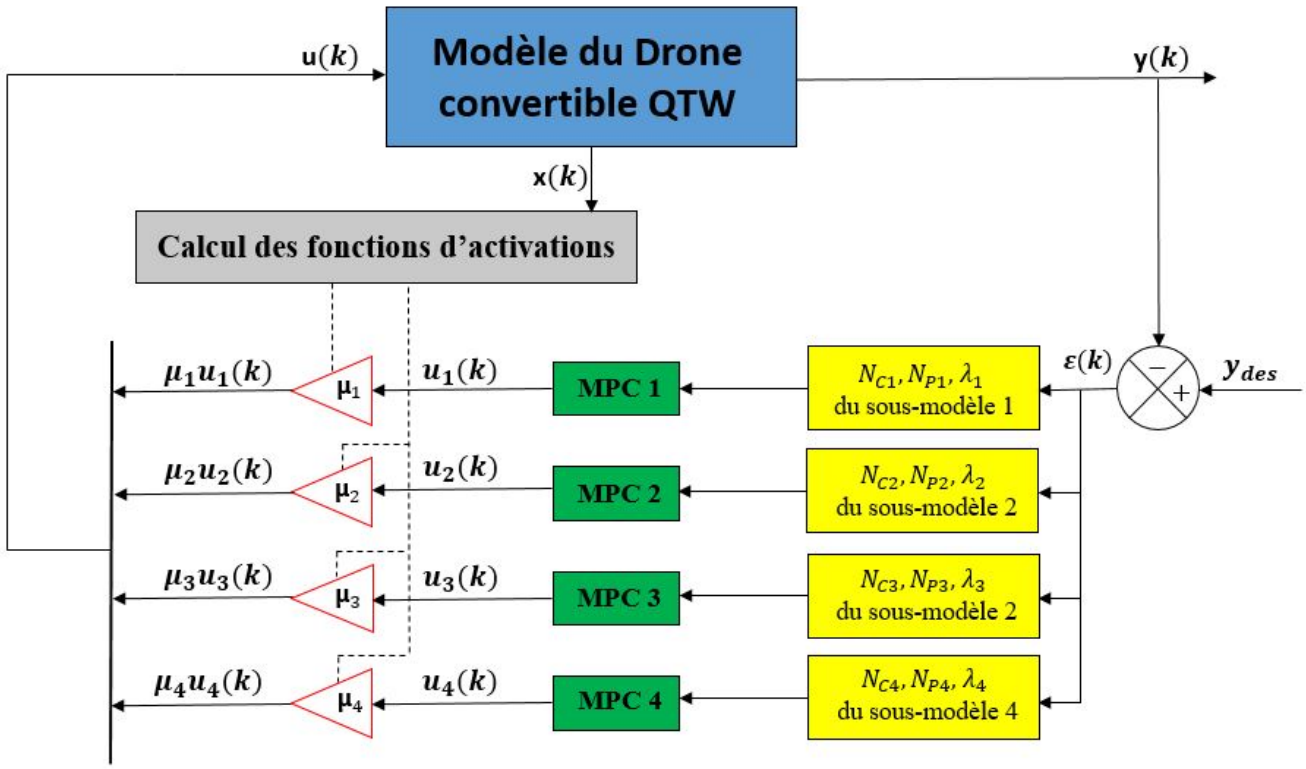


Figure 5.9 – Stratégie de commande proposée

Tableau 5.2 – Valeurs des paramètres des contrôleurs prédictifs pour chaque sous-modèle selon l'approche analytique (Turki 2018) [165]

Paramètres	Sous-modèle 1	Sous-modèle 2	Sous-modèle 3	Sous-modèle 4
$\mathbf{N_c}$	32	28	4	10
$\mathbf{N_p}$	43	39	42	19
λ	8	16	18	9

de calcul en simulation sera moins élevé.

Les résultats de simulation obtenus sur le modèle T-S en comparant les deux approches sont présentés dans les Figures 5.10, 5.11 et 5.12.

La Figure 5.10 montre l'évolution de l'altitude des angles de roulis, de tangage, et de lacet en fonction du temps.

La Figure 5.11 montre l'évolution de la sortie désirée (où la référence), avec l'évolution des signaux de sortie des vitesses angulaires et de la vitesse linéaire.

Les signaux de commande en fonction du temps sont représenté dans la Figure 5.12.

D'après les résultats obtenus sur les Figures 5.10, 5.11 et 5.12, on constate clairement l'efficacité de l'approche de réglage de la commande prédictive par RNA sur le suivi du trajectoire des systèmes multi-variables.

Pour les trois sorties, l'approche proposée permet d'obtenir de meilleures performances

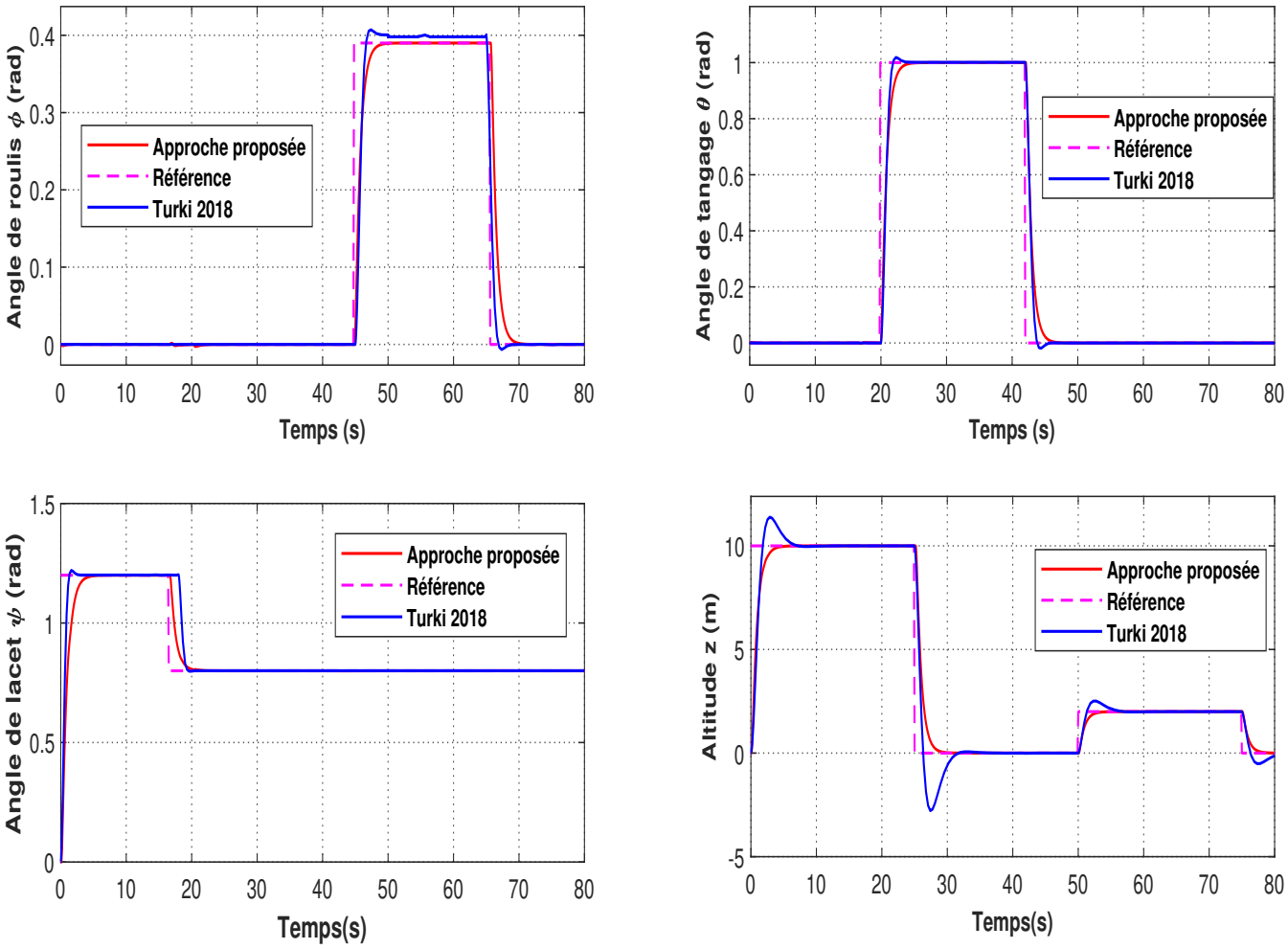


Figure 5.10 – Évolution des sorties d'attitudes et d'altitude en fonction du temps

sur quatre des cinq critères considérés.

En effet, en termes de précision, on remarque que l'erreur de poursuite entre les différents signaux de sorties et la consigne est moindre avec l'approche proposée. L'observation des résultats de simulation permet également de conclure sur le dépassement qui est quasi-nul. Le constat est le même pour le temps mis à atteindre la valeur finale en régime permanent : l'approche neuronale affiche un temps de réponse moindre et donc une sortie plus rapide. En termes de stabilité, les signaux de sorties obtenus par les deux approches sont stables, mais celle que nous proposons permet d'aller plus loin avec un indice de stabilité plus proche de 1, et donc par conséquent une meilleure stabilité en boucle fermée.

Au niveau des signaux de commande obtenus au moyen de la MPC, la méthode de réglage par RNA permet d'obtenir l'amplitude la moins élevée qui se traduit par des commandes plus souples.

Afin de confirmer cette constatation, une comparaison en performances entre les deux

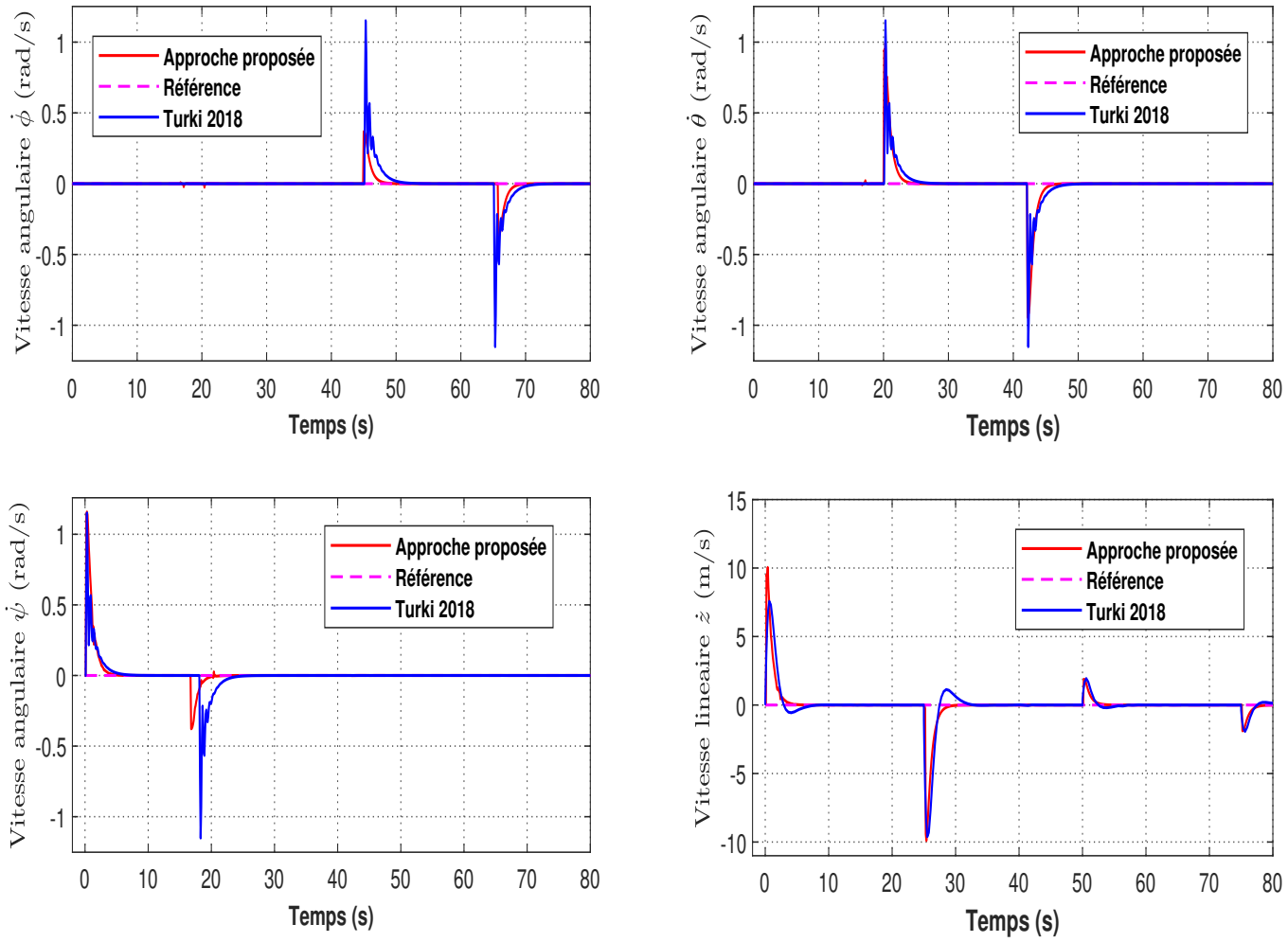


Figure 5.11 – Évolution des sorties angulaires et linéaire en fonction du temps

approches est détaillée dans les Tableaux 5.3, 5.4 et 5.5. Les indicateurs de performances considérés sont ceux déjà considérés pour les précédentes applications.
Les tableaux ci-dessous compare les performances d'une sortie donnée (Angle de roulis, de tangage et de lacet) :

Tableau 5.3 – Comparaison des performances relatives à l'angle de roulis

	Turki 2018	Approche proposée
RT (s)	**45.03**	45.09
ST (s)	**48.25**	52.78
OV (%)	3.8	**0.1**
MTE (s)	2.9	**1.3**
IDS	0.340	**0.762**

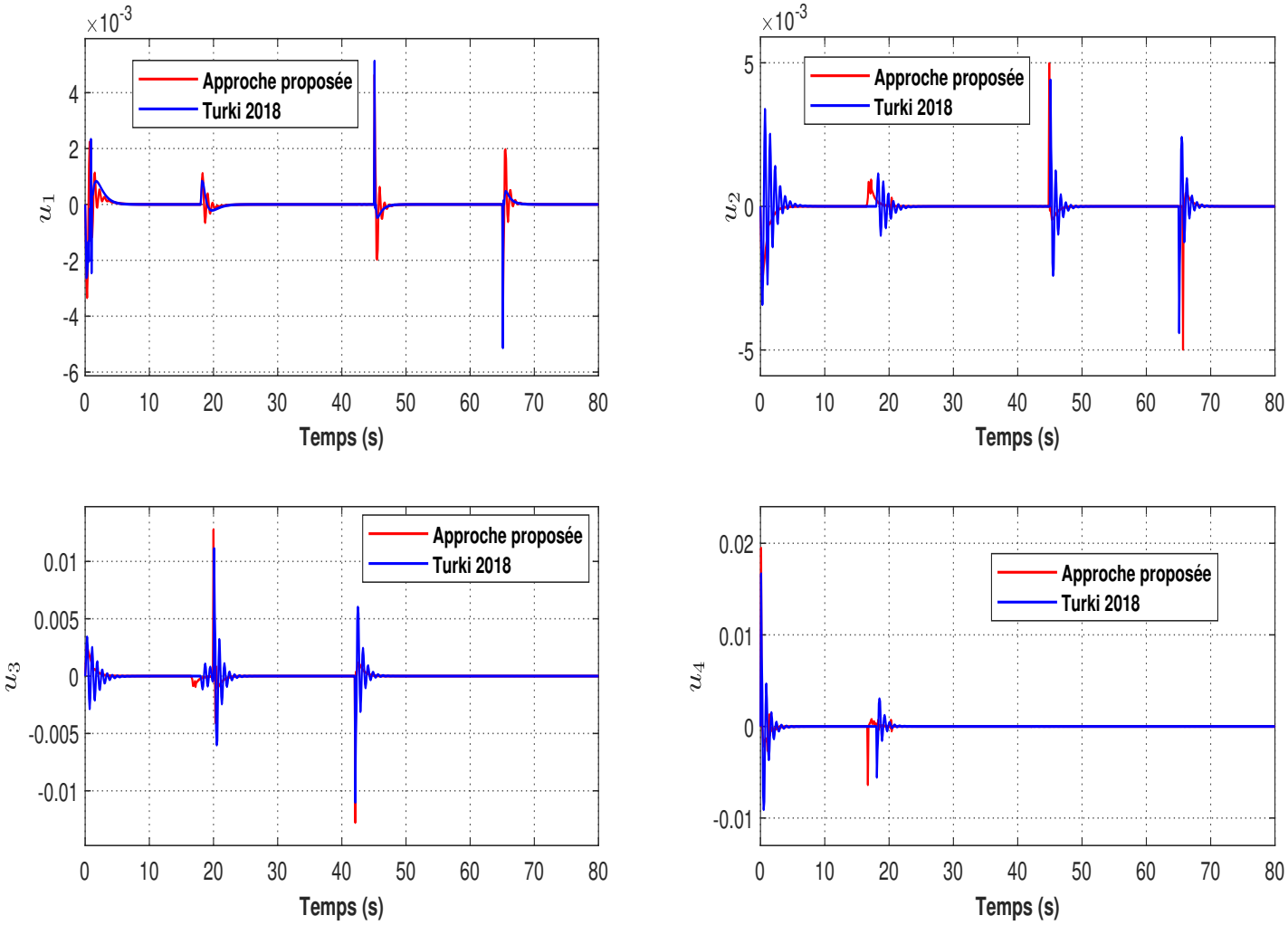


Figure 5.12 – Signaux de commande en fonction du temps

Tableau 5.4 – Comparaison des performances relatives à l'angle de tangage

	Turki 2018	Approche proposée
RT (s)	**21**	21.96
ST (s)	24.26	**23.89**
OV (%)	2.7	**0.1**
MTE (s)	1.8	**1.2**
IDS	0.640	**0.739**

Les résultats présentés dans ces tableaux confirment la constatation tirée à partir de l'observation des simulations numériques des sorties du drone convertible QTW.
Par conséquent, nous pouvons conclure que l'approche de réglage par RNA de la commande prédictive proposée permet d'atteindre le meilleur compromis en terme de précision, de stabilité optimale du système en boucle fermée, de stabilité numérique et de

Tableau 5.5 – Comparaison des performances relatives à l'angle de lacet

	Turki 2018	Approche proposée
RT (s)	**1.40**	2.13
ST (s)	4.26	**3.68**
OV (%)	1.4	**0**
MTE (s)	3.2	**2.7**
IDS	0.580	**0.657**

rapidité.

5.6 Conclusion

Dans ce chapitre, nous avons commencé par présenter les drones convertibles de type QTW, leurs modes de vol ainsi qu'un modèle dynamique non linéaire global issu de la littérature.
Un modèle T-S représentatif du drone durant un vol vertical a ensuite été établi.
Ce modèle T-S a permis de développer quatre contrôleurs prédictifs s'appuyant chacun sur l'un de ses sous-modèles.

Après la détermination des sous-modèles linéaires, l'approche de réglage par RNA a été appliquée pour calculer les paramètres de réglages des quatre contrôleurs.
Afin d'évaluer la pertinence de l'approche de réglage proposée, une comparaison avec les résultats obtenus par une méthode analytique a été effectué. Cette comparaison en faveur de l'approche proposée montre que cette dernière est applicable sur n'importe quelle classe des systèmes (SISO, SISO retardé, MIMO, MIMO retardé, et MIMO non linéaire pour lesquels un modèle T-S peut être établi).

Conclusion générale et perspectives

Les travaux de recherche menés durant cette thèse ont porté sur la problématique du réglage paramétrique de la commande prédictive par une approche complètement originale basée sur les réseaux de neurones artificiels.
L'approche proposée permet de synthétiser un contrôleur prédictif auto-adaptatif applicable sur des systèmes SISO et MIMO linéaires en présence et en absence de perturbations quelque soit leur ordre, ainsi que sur des systèmes non linéaires décrits par un modèle T-S, tout en respectant l'hypothèse que les différents systèmes considérés sont commandables et observables.
L'originalité de l'approche réside dans l'utilisation des réseaux de neurones artificiels considérés comme l'un des outils le plus utilisé en intelligence artificielle. Le RNA a permis de faire une généralisation de l'approche sur plusieurs classes de systèmes.
A travers les différentes applications, nous avons prouvé que l'approche proposée possède plusieurs avantages :

- La prise en compte des contraintes.
- La garantie de la stabilité en boucle fermée.
- Le rejet de perturbations.
- La prise en compte des variations paramétriques.
- La garantie du suivi de trajectoire avec une erreur de poursuite minimale et un dépassement quasi nul.

Dans le premier chapitre de ce mémoire, nous avons introduit la commande prédictive, son principe de fonctionnement et son historique d'utilisation. Ensuite, un état de l'art sur les différentes approches de réglage de la MPC existantes dans la littérature a été effectué afin de déterminer le besoin actuel et les différentes limites à surmonter.
Le deuxième chapitre a porté sur les RNA, en commençant par présenter leur principe de fonctionnement, et l'historique de leurs utilisations. Ensuite, un état de l'art synthétique a été mené sur les stratégies d'apprentissage et de dimensionnement des RNA. Dans ce même chapitre, la proposition d'une approche d'apprentissage et de dimensionnement en ligne (OS-ELM) a été présentée. Nous avons, par la suite, appliqué l'OS-ELM sur des systèmes SISO afin de montrer sa pertinence.

Dans le troisième chapitre, nous avons présenté l'approche proposée de réglage de la MPC par RNA, en commençant par la formulation mathématique de la commande prédictive à base d'état. Puis, nous nous somme appuyés sur l'approche PSO pour la construction de la base de données d'apprentissage et l'OS-ELM pour l'apprentissage et la définition de la structure optimale du RNA. Afin de montrer la pertinence de l'approche proposée, nous avons sélectionné des critères de performances et effectuer une comparaison entre notre méthode et des approches existantes pour différents systèmes (SISO, MIMO avec et sans retard).

L'objectif du chapitre quatre est de tester la robustesse en performances de la MPC en présence de perturbations et de bruits de mesure. Le chapitre a été introduit avec un état de l'art des approches de robustifications de la commande prédictive afin de robustifier le contrôleur sans changer le comportement d'entrée/sortie. Nous avons ensuite effectué une étude comparative de ces approches afin de choisir la méthode la plus robuste en terme de performances. Le choix a porté sur l'approche de robustification de YOULA.
Finalement, l'approche proposée a été appliquée sur des systèmes perturbés pour prouver qu'une combinaison des paramètres de réglage permettant le rejet des perturbations sans avoir recours à des outils de robustification.

Dans le cinquième et dernier chapitre, une extension de l'approche proposée a été présentée pour le contrôle des systèmes multivariables non linéaires.
Le système que nous avons considéré est un drone convertible QTW non linéaire. Dans un premier temps, les drones QTW ont été présentés et un modèle dynamique a été défini. Ensuite, dans un deuxième temps, nous nous sommes appuyés sur la technique de "Takagi-Sugeno" (T-S) pour écrire ce système sous la forme de plusieurs sous-systèmes linéaires ou quasi linéaires représentant le comportement du système initial. Nous avons validé cette propriété en simulation en comparant le comportement du système non linéaire avec celui des sous systèmes linéaires.
Une application de l'approche proposée sur le système de drone a été présentée. Les résultats obtenus ont été comparés avec ceux d'une approche de réglage analytique. L'approche proposée a montré une efficacité remarquable en termes de performances et de suivi de trajectoire.
Les différentes applications en simulation de l'approche proposée ont permis d'observer des résultats encourageants et prometteurs pour la synthèse d'une commande prédictive auto-adaptative.

Néanmoins, il existe des perspectives qui pourrait apporter des améliorations et que

nous estimons abordables, parmi ces perspectives on peut citer : Dans la continuité des travaux réalisés, des perspectives de recherche sont identifiées. Sur des aspects théoriques il est proposé de :

- Étudier théoriquement la stabilité optimale en boucle fermée du système non linéaire complet sans passer par le modèle réduit. En effet le modèle réduit d'un système donné n'est jamais complètement représentatif du comportement non linéaire du système, ce qui ne permet pas de conclure sur la performance et la stabilité du système en boucle fermée.
- Construire une base de données d'apprentissage du réseau de neurones en se basant sur des résultats de tests réel et non sur une approche d'optimisation avec plusieurs hypothèses. Une base d'apprentissage basée sur des données réelles permettra d'obtenir les paramètres de réglage spécifiques au système à contrôler.
- Effectuer une étude théorique sur les incertitudes paramétriques liées à la représentativité du système réel, pour pouvoir conclure sur l'approche de réglage proposée.
- Synthétiser une loi de commande tolérante aux défaut actionneurs pour les systèmes non linéaires afin d'améliorer les performances en boucle fermée. En effet, les différents défauts injectées dans notre étude de robustesse sont des défauts instantanés qui ne permettent pas d'évaluer l'intérêt de la mise à jour des paramètres de réglage de la MPC.
- Injecter directement en entrée du réseau de neurones les matrices d'état du système à contrôler sans devoir passer par le modèle d'état augmenté. Ce dernier permet certes d'introduire l'action intégral dans la boucle de régulation mais demande un temps de calcul important.

Sur les aspects applicatifs, il est proposé de :

- Appliquer en temps réel l'approche proposée sur des système industriels complexes en présence de contraintes et des perturbations, afin de confirmer l'impact du réglage paramétrique prouvé en simulation dans le cadre de ces travaux.
- Faire une comparaison entre les résultats obtenus en simulation et ceux obtenues sur des procédés réels, pour voir si le modèle utilisé est représentatif.

Bibliographie

[1] Debbeh Abdesslem. Commande prédictive généralisée robuste etude et application. 2006.

[2] Ines Abidi, Jerome Bosche, Ahmed El Hajjaji, and Abdel Aitouche. Control of a turbocharged diesel engine with egr system using takagi-sugeno's approach. In *Control & Automation (MED), 2012 20th Mediterranean Conference on*, pages 960–965. IEEE, 2012.

[3] Ashraf Al-Ghazzawi, Emad Ali, Adnan Nouh, and Evanghelos Zafiriou. On-line tuning strategy for model predictive controllers. *Journal of Process Control*, 11(3) :265–284, 2001.

[4] Emad Ali. Automatic tuning of model predictive controllers based on fuzzy logic. *optimization*, 2(2) :1, 2001.

[5] Emad Ali. Heuristic on-line tuning for nonlinear model predictive controllers using fuzzy logic. *Journal of Process Control*, 13(5) :383–396, 2003.

[6] F Allgöwer, TA Badgwell, JS Qin, JB Rawlings, and SJ Wright. Advances in control/highlights of ecc'99//chapt. 12. nonlinear predictive controls and moving horizon estimation, 1999.

[7] Frank Allgöwer and Alex Zheng. *Nonlinear model predictive control*, volume 26. Birkhäuser, 2012.

[8] Benyamine Allouche. *Modélisation et commande des robots nouvelles approches basées sur les modèles Takagi-Sugeno*. PhD thesis, Université de Valenciennes et du Hainaut-Cambresis, 2016.

[9] Didier Dumur Anamaria Luca, Pedro Rodriguez-Ayerbe. Control of disturbed lpv systems in a lmi setting. *Proceedings of the 18th world congress the international federation of automatic control*, 2011.

[10] Christopher MacLeod Ann Reddipogu, Grant Maxwell and Malcolm Simpson. A novel artificial neural network trained using evolutionary algorithms for reinforcement learning. 2002.

[11] Vassilvitskii S Arthur D. k-means++ : The advantages of careful seeding. *Proceedings of the eighteenth annual ACM-SIAM symposium on Discrete algorithms*, 2007.

[12] R. Austin. *Unmanned Aircraft Systems : UAVs Design, Development and Deployment*. John Wiley & Sons Ltd.

[13] Pedro Rodriguez Ayerbe. Robustification de lois de commande prédictive par la paramétrisation de youla. 2003.

[14] Peyman Bagheri and Ali Khaki-Sedigh. An anova based analytical dynamic matrix controller tuning procedure for fopdt models. *Amirkabir International Journal of Modeling, Identification, Simulation & Control*, 42(2) :55–64, 2010.

[15] Peyman Bagheri and Ali Khaki-Sedigh. Tuning of dynamic matrix controller for fopdt models using analysis of variance. *IFAC Proceedings Volumes*, 44(1) :12319–12324, 2011.

[16] Peyman Bagheri and Ali Khaki-Sedigh. An analytical tuning approach to multivariable model predictive controllers. *Journal of Process Control*, 24(12) :41–54, 2014.

[17] Peyman Bagheri and Ali Khaki-Sedigh. Closed form tuning equations for model predictive control of first-order plus fractional dead time models. *International Journal of Control, Automation and Systems*, 13(1) :73–80, 2015.

[18] Peyman Bagheri and Ali Khaki Sedigh. Analytical approach to tuning of model predictive control for first-order plus dead time models. *IET Control Theory & Applications*, 7(14) :1806–1817, 2013.

[19] Peyman Bagheri and Ali Khaki Sedigh. Robust tuning of dynamic matrix controllers for first order plus dead time models. *Applied Mathematical Modelling*, 39(22) :7017–7031, 2015.

[20] Tribeni Prasad Banerjee, Joydeb Roychoudhury, Swagatam Das, and Ajith Abraham. Hybrid intelligent predictive control system for high speed bldc motor in aerospace application. In *Emerging Trends in Engineering and Technology (ICETET), 2010 3rd International Conference on*, pages 258–262. IEEE, 2010.

[21] Faten Ben Aicha, Faouzi Bouani, and Mekki Ksouri. A multivariable multiobjective predictive controller. *International Journal of Applied Mathematics and Computer Science*, 23(1) :35–45, 2013.

[22] L. Ben Hamouda, M. Ayadi, and N. Langlois. Fuzzy fault tolerant predictive control for a diesel engine air path. *International Journal of Control, Automation and Systems*, 14(2) :443–451, 2016.

[23] L. Ben Hamouda, O. Bennouna, M. Ayadi, and N. Langlois. Fuzzy model predictive reconfigurable control for nonlinear systems subject to actuators faults. In *International Conference on Automation and Computing*, pages 140–145, Bedfordshire, 2014.

[24] Lamia Ben Hamouda. *Sur la synthèse de commandes prédictives tolérantes aux défauts à base de modèles TS flous*. PhD thesis, École Nationale d'ingénieurs de Tunis et Université de Rouen, 2015.

[25] K. BenKhoud, S. Bouallègue, and M. Ayadi. *Rapid control prototyping of a quad-tilt-wing unmanned aerial vehicle*. 2017.

[26] Khaled Benkhoud and Soufiene Bouallègue. Model predictive control design for a convertible quad tilt-wing uav. In *Control Engineering & Information Technology (CEIT), 2016 4th International Conference on*, pages 1–6. IEEE, 2016.

[27] Khaled Benkhoud and Soufiene Bouallègue. Dynamics modeling and advanced metaheuristics based lqg controller design for a quad tilt wing uav. *International Journal of Dynamics and Control*, pages 1–22, 2017.

[28] Mohamed Abdelmoula Benlahrache. Commande prédictive, et commande tolérante aux défauts appliquées au système éolien. 2017.

[29] A.O.T.Chang B.Kouvaritakis, J.A.Rossiter. table generalized predictive control : an algorithm with guaranteed stability. 1992.

[30] Rigit A R Bong D, Tan J. Optimization of the backpropagation hidden layer by hybrid k-means-greedy algorithm for time series prediction. *In Industrial Electronics Applications (ISIEA)*, 2010.

[31] Carlos Bordons and Eduardo F Camacho. A generalized predictive controller for a wide class of industrial processes. *IEEE Transactions on control systems technology*, 6(3) :372–387, 1998.

[32] Ayadi M Benrejeb M Bouallègue S, Haggège J. Pid-type fuzzy logic controller tuning based on particle swarm optimization. *Engineering Applications of Artificial Intelligence*, 7(6) :484–493, 2012.

[33] Patrick Boucher and Didier Dumur. *La commande prédictive*, volume 8. Editions Technip, 1996.

[34] N Bursali, B Akay, S Ertunc, H Hapoglu, and M Alpbaz. New tuning method for generalized predictive control of the production of s. cerevisiae. *Food and bioproducts processing*, 79(1) :27–34, 2001.

[35] Eduardo F Camacho and Carlos Bordons Alba. *Model predictive control*. Springer Science & Business Media, 2013.

[36] Nicolas Cazy. *Commande prédictive pour la réalisation de tâches d'asservissement visuel successives*. PhD thesis, Rennes 1, 2016.

[37] E. Cetinsoy, S. Dikyar, C. Hancer, K.T. Oner, E. Sirimoglu, M. Unel, and M.F. Aksit. Design and construction of a novel quad tilt-wing uav. *Mechatronics*, 2012.

[38] Ye-Then Chang and Yen-Shin Lai. Online parameter tuning technique for predictive current-mode control operating in boundary conduction mode. *IEEE Transactions on Industrial Electronics*, 56(8) :3214–3221, 2009.

[39] Ye-Then Chang and Yen-Shin Lai. Online parameter tuning technique for predictive current-mode control operating in boundary conduction mode. *IEEE Transactions on Industrial Electronics*, 56(8) :3214–3221, 2009.

[40] Hong Chen and Frank Allgöwer. A quasi-infinite horizon nonlinear model predictive control scheme with guaranteed stability ⋆. *Automatica*, 34(10) :1205–1217, 1998.

[41] David W Clarke. Application of generalized predictive control to industrial processes. *IEEE Control systems magazine*, 8(2) :49–55, 1988.

[42] David W Clarke, C Mohtadi, and PS Tuffs. Generalized predictive control part 1. the basic algorithm. *Automatica*, 23(2) :137–148, 1987.

[43] Kennedy J Clerc M. The particle swarm-explosion, stability, and convergence in a multidimensional complex spaces. *IEEE Transactions on Evolutionary Computation*, 6(1) :58–73, 2002.

[44] Rivest R Stein C Cormen T H, Leiserson C. Introduction to algorithms second edition. 2001.

[45] B.C. Ramaker C.R. Cutler. A computer control algorithm. *Automatic Control Conference*, 1980.

[46] Charles R Cutler and Brian L Ramaker. Dynamic matrix control ? ? a computer control algorithm. In *joint automatic control conference*, number 17, page 72, 1980.

[47] Hebb D. The organization of behavior. 1949.

[48] Robin MC De Keyser and Achiel R Van Cauwenberghe. A self-tuning multistep predictor application. *Automatica*, 17(1) :167–174, 1981.

[49] Stefano Di Cairano and Alberto Bemporad. Model predictive control tuning by controller matching. *IEEE Transactions on Automatic Control*, 55(1) :185–190, 2010.

[50] Dimitar Dimitrov, Alexander Sherikov, and Pierre-Brice Wieber. A sparse model predictive control formulation for walking motion generation. In *Intelligent Robots and Systems (IROS), 2011 IEEE/RSJ International Conference on*, pages 2292–2299. IEEE, 2011.

[51] Quoc Tran Dinh, Carlo Savorgnan, and Moritz Diehl. Real-time sequential convex programming for nonlinear model predictive control and application to a hydro-power plant. In *Decision and Control and European Control Conference (CDC-ECC), 2011 50th IEEE Conference on*, pages 5905–5910. IEEE, 2011.

[52] M Djeghaba. *Modélisation multimodèle et commande prédictive d'une station d'épuration*. PhD thesis, Université Badji Mokhtar-Annaba, 2012.

[53] Leandro Dos Santos Coelho and Renato A Krohling. Predictive controller tuning using modified particle swarm optimization based on cauchy and gaussian distributions. In *Soft computing : methodologies and applications*, pages 287–298. Springer, 2005.

[54] Didier Dumur, Patrick Boucher, and Gérard Ramond. Direct adaptive generalized predictive control application to motor drives with flexible modes. *CIRP Annals-Manufacturing Technology*, 49(1) :271–274, 2000.

[55] Kennedy J Eberhart R. A new optimizer using particle swarm theory. pages 39–43, 1995.

[56] Shi Y Eberhart R. A new optimizer using particle swarm theory. pages 81–86, 2001.

[57] Zahed Ebrahimi, Peyman Bagheri, and Ali Khaki-Sedigh. Tuning of generalized predictive controllers for first order plus dead time models based on anova. In *2015 23rd Iranian Conference on Electrical Engineering*, pages 928–933. IEEE, 2015.

[58] C.Bordons E.F.Camacho. Model predictive control. *Ed. Springer-Verlag, 2nd ed*, 2004.

[59] Rosenblatt F. The perceptron : a probabilistic model for information storage and organization in the brain. *Psychol. Rev*, 65(6) :386–408, 1958.

[60] Paolo Falcone, Francesco Borrelli, Jahan Asgari, Hongtei Eric Tseng, and Davor Hrovat. Predictive active steering control for autonomous vehicle systems. *IEEE Transactions on control systems technology*, 15(3) :566–580, 2007.

[61] Lin Q Gay R Feng G, Huang G. Error minimized extreme learning machine with growth of hidden nodes and in-cremental learning. *IEEE Transactions on neural networks*, 20, 1357.

[62] Peter J Fleming and Robin C Purshouse. Evolutionary algorithms in control systems engineering : a survey. *Control engineering practice*, 10(11) :1223–1241, 2002.

[63] David B Fogel. *Evolutionary computation : toward a new philosophy of machine intelligence*, volume 1. John Wiley & Sons, 2006.

[64] Sandou G. Metaheuristic strategy for the hierarchical predictive control of large scale energy networks. *Journal of Control Engineering and Applied Informatics*, 11(1) :32–40, 2009.

[65] Carlos E Garcia, David M Prett, and Manfred Morari. Model predictive control : theory and practice-a survey. *Automatica*, 25(3) :335–348, 1989.

[66] Jorge L Garriga and Masoud Soroush. Model predictive controller tuning via eigenvalue placement. In *2008 American Control Conference*, pages 429–434. IEEE, 2008.

[67] Jorge L Garriga and Masoud Soroush. Model predictive control tuning methods : A review. *Industrial & Engineering Chemistry Research*, 49(8) :3505–3515, 2010.

[68] Potvin J Gendreau M. Handbook of metaheuristics. 2010.

[69] S Gerksic and B Pregelj. Tuning of a tracking multi-parametric predictive controller using local linear analysis. *IET control theory & applications*, 6(5) :669–679, 2012.

[70] Tahereh Gholaminejad, Ali Khaki-Sedigh, and Peyman Bagheri. Adaptive tuning of model predictive control based on analytical results. In *2016 4th International Conference on Control, Instrumentation, and Automation (ICCIA)*, pages 226–232. IEEE, 2016.

[71] Ernesto Granado. *Commande prédictive à base de programmation semi définie*. PhD thesis, INSA de Toulouse, 2004.

[72] Rohit Gupta, Uroš V Kalabić, Stefano Di Cairano, Anthony M Bloch, and Ilya V Kolmanovsky. Constrained spacecraft attitude control on so (3) using fast nonlinear model predictive control. In *American Control Conference (ACC), 2015*, pages 2980–2986. IEEE, 2015.

[73] RC Gutiérrez-Urquídez, Guillermo Valencia-Palomo, Oscar M Rodríguez-Elias, and L Trujillo. Systematic selection of tuning parameters for efficient predictive controllers using a multiobjective evolutionary algorithm. *Applied Soft Computing*, 31 :326–338, 2015.

[74] RC Gutiérrez-Urquídez, Guillermo Valencia-Palomo, Oscar M Rodríguez-Elias, and L Trujillo. Systematic selection of tuning parameters for efficient predictive controllers using a multiobjective evolutionary algorithm. *Applied Soft Computing*, 31 :326–338, 2015.

[75] Robert Haber, Ulrich Schmitz, and Ruth Bars. Optimal choice of horizons for predictive control by using genetic algorithms. *Annual reviews in control*, 28(1) :53–58, 2004.

[76] Honggui Han and Junfei Qiao. Nonlinear model-predictive control for industrial processes : An application to wastewater treatment process. *IEEE Transactions on Industrial Electronics*, 61(4) :1970–1982, 2014.

[77] Kai Han, Jun Zhao, and Jixin Qian. A novel robust tuning strategy for model predictive control. In *2006 6th World Congress on Intelligent Control and Automation*, volume 2, pages 6406–6410. IEEE, 2006.

[78] Ning He, Dawei Shi, Jiadong Wang, Michael Forbes, Johan Backstrom, and Tongwen Chen. Automated two-degree-of-freedom model predictive control tuning. *Industrial & Engineering Chemistry Research*, 54(43) :10811–10824, 2015.

[79] Wisel T N Hopfield J. Neural computation of decisions in optimization problems. *Biological Cybernetics*, 55(1), 1985.

[80] Madjid Haddad Houssam Mouhmouh, Nicolas Langlois. A novel tuning approach for mpc parameters based on artificial neural network. 2019.

[81] Madjid Haddad Houssam Mouhmouh, Nicolas Langlois. A novel tuning approach for parameters based on artificial neural network : An application to fopdt systems. 2019.

[82] Madjid Haddad Houssam Mouhmouh, Nicolas Langlois. Off-line robustness improvement of a predictive controller using a novel tuning approach based on artificial neural network. 2020.

[83] Madjid Haddad Houssam Mouhmouh, Nicolas Langlois. Robustness of model predictive control using a novel tuning approach based on artificial neural network. 2020.

[84] Yun Huang and Ali Jadbabaie. Nonlinear h inifinity control an enhanced quasi-lpv approach. *IFAC Proceedings Volumes*, 32(2) :2754–2759, 1999.

[85] Siew C Huang G, Zhu W. Extreme learning machine. *heory and applications. Neurocomputing*, 2006.

[86] Dalil Ichalal. *Estimation et diagnostic de systèmes non linéaires décrits par un modèle de Takagi-Sugeno*. PhD thesis, Institut National Polytechnique de Lorraine-INPL, 2009.

[87] Dalil Ichalal, Benoît Marx, Didier Maquin, and José Ragot. Nonlinear observer based sensor fault tolerant control for nonlinear systems. *IFAC Proceedings Volumes*, 45(20) :1053–1058, 2012.

[88] Dalil Ichalal, Benoit Marx, José Ragot, and Didier Maquin. Simultaneous state and unknown inputs estimation with pi and pmi observers for takagi sugeno model with unmeasurable premise variables. In *Control and Automation, 2009. MED'09. 17th Mediterranean Conference on*, pages 353–358. IEEE, 2009.

[89] Dalil Ichalal, Benoît Marx, José Ragot, and Didier Maquin. State estimation of nonlinear systems using multiple model approach. In *American Control Conference, 2009. ACC'09.*, pages 4636–4641. IEEE, 2009.

[90] Dalil Ichalal, Benoît Marx, José Ragot, and Didier Maquin. State estimation of takagi–sugeno systems with unmeasurable premise variables. *IET Control Theory & Applications*, 4(5) :897–908, 2010.

[91] Edinzo J Iglesias, Marco E Sanjuán, and Carlos A Smith. Tuning equation ford dynamic matrix control in siso loops. *Ingeniería y desarrollo*, (19), 2006.

[92] Yuichi Ikeda, Takashi Nakajima, and Yuichi Chida. Attitude control of spacecraft by nmpc with consideration of singularity avoidance of cmg. In *Decision and Control (CDC), 2012 IEEE 51st Annual Conference on*, pages 1733–1739. IEEE, 2012.

[93] Schmidhuber J. Deep learning in neural networks. *Neural Netw*, 61(3), 2014.

[94] Tor Arne Johansen, Robert Shorten, and Roderick Murray-Smith. On the interpretation and identification of dynamic takagi-sugeno fuzzy models. *IEEE Transactions on Fuzzy systems*, 8(3) :297–313, 2000.

[95] Uros Kalabic, Rohit Gupta, Stefano Di Cairano, Anthony Bloch, and Ilya Kolmanovsky. Constrained spacecraft attitude control on so (3) using reference governors and nonlinear model predictive control. In *American Control Conference (ACC), 2014*, pages 5586–5593. IEEE, 2014.

[96] Wen-Tsuen Chen Keun-Rong. A neural network model which combines unsupervised and supervised learning. *IEEE TRANSACTIONS ON NEURAL NETWORKS*, 1993.

[97] B Khan and JA Rossiter. Alternative parameterisation within predictive control : a systematic selection. *International Journal of Control*, 86(8) :1397–1409, 2013.

[98] B. Wittenmark K.J. Åström. Computer controlled systems. 1997.

[99] $\text{Klopf}_1 972.\ Brainfunctionandadaptivesystems - aheterostatictheory.AirForceResearchLaboratoriesTechnicalReport, 1553(1), 1972.$

[100] Jonas Koenemann, Andrea Del Prete, Yuval Tassa, Emanuel Todorov, Olivier Stasse, Maren Bennewitz, and Nicolas Mansard. Whole-body model-predictive control applied to the hrp-2 humanoid. In *Intelligent Robots and Systems (IROS), 2015 IEEE/RSJ International Conference on*, pages 3346–3351. IEEE, 2015.

[101] Chrisley R Kohonen T, Barna G. Statistical pattern recognition with neural networks. *Proceedings of the IEEE Conference on Neural Network*, 1988.

[102] Abdullah Konak, David W Coit, and Alice E Smith. Multi-objective optimization using genetic algorithms : A tutorial. *Reliability Engineering & System Safety*, 91(9) :992–1007, 2006.

[103] JH Lee and ZH Yu. Tuning of model predictive controllers for robust performance. *Computers & chemical engineering*, 18(1) :15–37, 1994.

[104] DJ Leith and WE Leithead. Global reconstruction of nonlinear systems from families of linear systems. *IFAC Proceedings Volumes*, 35(1) :435–440, 2002.

[105] Zhou Juaqing Liang Jiuzhen. Procedure neural networks with supervised learning. 2012.

[106] Saratchandran P Sundararajan N Liang N, Huang B. A fast and accurate online sequential learning algorithm for feedforward networks. *IEEE Transactions on neural networks*, 17, 2006.

[107] Canon A Lima A, Hsieh W. Variable complexity online sequential extreme learning machine, with applications to streamflow prediction. *Journal of Hydrology*, 2017.

[108] Wei Liu and George Wang. Auto-tuning procedure for model-based predictive controller. In *Systems, Man, and Cybernetics, 2000 IEEE International Conference on*, volume 5, pages 3421–3426. IEEE, 2000.

[109] Herbert K Liu J, Wang J. Xml clustering by principal component analysis. in tools with artifi-cial intelligence. *16th IEEE International Conference on Artifi-cial Intelligence*, 2004.

[110] R. Lozano. *Unmanned Aerial Vehicles : Embedded Control*. John Wiley & Sons, ISTE, 2010.

[111] Anamaria Luca. Outils ensemblistes danalyse et de synthèse des lois de commande robustes pour des systèmes incertains. 2011.

[112] J Ward MacArthur. Rmpct : A new robust approach to multivariable predictive control for the process industries. In *The 1996 Control Systems Conference*, pages 53–60, 1996.

[113] Jan Marian Maciejowski. *Predictive control : with constraints*. Pearson education, 2002.

[114] Jerry Mamboundou. *Commande prédictive à base d'état adaptive indirecte supervisée par la logique floue*. PhD thesis, Rouen, 2013.

[115] Jerry Mamboundou and Nicolas Langlois. Indirect adaptive model predictive control supervised by fuzzy logic. In *Fuzzy Systems (FUZZ), 2011 IEEE International Conference on*, pages 2979–2986. IEEE, 2011.

[116] Jerry Mamboundou and Nicolas Langlois. Robustness analysis of indirect adaptive model predictive control supervised by fuzzy logic. In *Industrial Technology (ICIT), 2012 IEEE International Conference on*, pages 284–291. IEEE, 2012.

[117] Pitts W McCulloch W. A logical calculus of the ideas immanent in nervous activity. *Bull. Math. Bioph*, pages 115–133, 1943.

[118] Ernesto Granado Migloire. Commande prdedictive à base de programmmation sémi-définie. 2005.

[119] Ppert S Minsky M. Perceptrons : An introduction to computational geometry. *MIT Press, Cambridge*, 1969.

[120] Mahmood Mirzaei, Niels Kjølstad Poulsen, and Hans Henrik Niemann. Robust model predictive control of a wind turbine. In *American Control Conference (ACC), 2012*, pages 4393–4398. IEEE, 2012.

[121] S-M Moon, RL Clark, and DG Cole. The recursive generalized predictive feedback control : theory and experiments. *Journal of Sound and Vibration*, 279(1) :171–199, 2005.

[122] Manfred Morari. Model predictive control : multivariable control technique of choice in the 1990s ? 1993.

[123] Manfred Morari and Jay H Lee. Model predictive control : past, present and future. *Computers & Chemical Engineering*, 23(4) :667–682, 1999.

[124] W Naeem, R Sutton*, J Chudley, FR Dalgleish, and S Tetlow. An online genetic algorithm based model predictive control autopilot design with experimental verification. *International Journal of Control*, 78(14) :1076–1090, 2005.

[125] Khaoula Nagoudi-Layerle. *Application de la commande prédictive reconfigurable sur le circuit d'air du moteur Diesel*. PhD thesis, Rouen, 2011.

[126] Amir Reza Neshasteriz, Ali Khaki-Sedigh, and Houman Sadjadian. An analysis of variance approach to tuning of generalized predictive controllers for second order plus dead time models. In *Control and Automation (ICCA), 2010 8th IEEE International Conference on*, pages 1059–1064. IEEE, 2010.

[127] Hoai Nam Nguyen. *Commande sous contraintes pour des systèmes dynamiques incertains : une approache basée sur l'interpolation*. PhD thesis, Supélec, 2012.

[128] Hiroshi Ohtake, Kazuo Tanaka, and Hua O Wang. Fuzzy modeling via sector nonlinearity concept. *Integrated Computer-Aided Engineering*, 10(4) :333–341, 2003.

[129] Mohammed Oudghiri. *Commande multi-modèles tolérante aux défauts : Application au contrôle de la dynamique d'un véhicule automobile.* PhD thesis, Université de Picardie Jules Verne, 2008.

[130] V.Wertz P.Ansay, M.Gevers. Enhancing the robustness of gpc via a simple choice of the youla parameter. *European Journal of Control*, 1998.

[131] Fateme Pirouzmand and Nemat Ollah Ghahramani. Robust model predictive control based on mras for satellite attitude control system. In *Control, Instrumentation, and Automation (ICCIA), 2013 3rd International Conference on*, pages 53–58. IEEE, 2013.

[132] D. Poinsot. Commande d'un drone en vue de la conversion vol rapide-vol stationnaire. 2008.

[133] S Joe Qin and Thomas A Badgwell. An overview of industrial model predictive control technology. In *AIChE Symposium Series*, volume 93, pages 232–256. New York, NY : American Institute of Chemical Engineers, 1971-c2002., 1997.

[134] S Joe Qin and Thomas A Badgwell. A survey of industrial model predictive control technology. *Control engineering practice*, 11(7) :733–764, 2003.

[135] K Yamuna Rani and Heinz Unbehauen. Study of predictive controller tuning methods. *Automatica*, 33(12) :2243–2248, 1997.

[136] Jacques Richalet, A Rault, JL Testud, and J Papon. Model predictive heuristic control. *Automatica (Journal of IFAC)*, 14(5) :413–428, 1978.

[137] J Anthony Rossiter. *Model-based predictive control : a practical approach*. CRC press, 2003.

[138] Williams R Rumelhart D, Hinton G. Learning internal representations by error propagation. *Parallel Distributed Processing*, 1(1), 1986.

[139] Grossberg S. Adaptive pattern classification and universal recoding. *MIT Press, Cambridge*, 23(3), 1976.

[140] Antonio Sala. On the conservativeness of fuzzy and fuzzy-polynomial control of nonlinear systems. *Annual Reviews in Control*, 33(1) :48–58, 2009.

[141] P Sarhadi, K Salahshoor, and A Khaki-Sedigh. Robustness analysis and tuning of generalized predictive control using frequency domain approaches. *Applied Mathematical Modelling*, 36(12) :6167–6185, 2012.

[142] MH Schwarz, CS Cox, and J Börcsök. A filtered tuning method for a gpc controller. In *Signals and Systems Conference (ISSC 2010), IET Irish*, pages 180–185. IET, 2010.

[143] Mohammad B Shadmand, Robert S Balog, and Haitham Abu Rub. Auto-tuning the cost function weight factors in a model predictive controller for a matrix converter var compensator. In *2015 IEEE Energy Conversion Congress and Exposition (ECCE)*, pages 3807–3814. IEEE, 2015.

[144] Gaurang Shah and Sebastian Engell. Tuning mpc for desired closed-loop performance for siso systems. In *18th Mediterranean Conference on Control & Automation*, pages 628–633. IEEE, 2010.

[145] Gaurang Shah and Sebastian Engell. Tuning mpc for desired closed-loop performance for mimo systems. In *Proceedings of the 2011 American Control Conference*, pages 4404–4409. IEEE, 2011.

[146] Eberhart R Shi Y. Empirical study of particle swarm optimization. pages 1945–1950, 1999.

[147] Rahul Shridhar and Douglas J Cooper. A novel tuning strategy for multivariable model predictive control. *ISA transactions*, 36(4) :273–280, 1997.

[148] Rahul Shridhar and Douglas J Cooper. A tuning strategy for unconstrained siso model predictive control. *Industrial & Engineering Chemistry Research*, 36(3) :729–746, 1997.

[149] Michalewicz Siarry P. Advances in metaheuristics for hard optimization. 2008.

[150] Piotr Skworcow, Bogumil Ulanicki, Hossam AbdelMeguid, and Daniel Paluszczyszyn. Model predictive control for energy and leakage management in water distribution systems. 2010.

[151] Rahul Sridhar and Douqlas J Cooper. A tuning strategy for unconstrained multivariable model predictive control. *Ind. End. Chern. Res*, 37 :4003–40, 1998.

[152] C. Stoica. Robustification de lois de commande prédictive multivariables. 2008.

[153] Dumur D Stoica. "predictive functional contro. 2008.

[154] Ryohei Suzuki, Fukiko Kawai, Hideyuki Ito, Chikashi Nakazawa, Yoshikazu Fukuyama, and Eitaro Aiyoshi. Automatic tuning of model predictive control using particle swarm optimization. In *2007 IEEE Swarm Intelligence Symposium*, pages 221–226. IEEE, 2007.

[155] T. Takagi and M. Sugeno. Fuzzy identification of systems and its application to modeling and control. *IEEE Transactions. Systems, Man and Cybernetics*, 15 :116–132, 1985.

[156] K. Tanaka and M. Sugeno. Stability analysis and design of fuzzy control systems. *Fuzzy Sets and Systems*, 45 :135–156, 1992.

[157] K. Tanaka and H. Wang. *Fuzzy control system design and analysis. a Linear Matrix Inequality approach*. John Wiley and Sons Inc, 2001.

[158] Kazuo Tanaka, Takayuki Ikeda, and Hua O Wang. Fuzzy regulators and fuzzy observers : relaxed stability conditions and lmi-based designs. *IEEE Transactions on fuzzy systems*, 6(2) :250–265, 1998.

[159] Kazuo Tanaka, Masataka Nishimura, and Hua O Wang. Multi-objective fuzzy control of high rise/high speed elevators using lmis. In *American Control Conference, 1998. Proceedings of the 1998*, volume 6, pages 3450–3454. IEEE, 1998.

[160] Xiang C Teoh E, Tan K. Estimating the number of hidden neurons in a feedforward network using the singular value decomposition. *IEEE Transactions on Neural Networks*, 17(6), 2006.

[161] Quang N Tran, Leyla Özkan, and ACPM Backx. Generalized predictive control tuning by controller matching. *Journal of Process Control*, 25 :1–18, 2015.

[162] Quang N Tran, Joni Scholten, Leyla Ozkan, and ACPM Backx. A model-free approach for auto-tuning of model predictive control. *IFAC Proceedings Volumes*, 47(3) :2189–2194, 2014.

[163] Jorge Otávio Trierweiler and Sebastian Engell. The robust performance number : A new tool for control structure design. *Computers & chemical engineering*, 21 :S409–S414, 1997.

[164] Jorge Otávio Trierweiler and Luciano André Farina. Rpn tuning strategy for model predictive control. *Journal of Process Control*, 13(7) :591–598, 2003.

[165] Marwa Turki, Khaled Benkhoud, Nicolas Langlois, and Yassine Adnan. Analytical tuning of a tsmpc dedicated to nonlinear mimo systems. *European Control Conference (ECC)*, pages 1367–1372, 2018.

[166] Marwa Turki, Nicolas Langlois, and Yassine Adnan. An analytical tuning approach for adaptive mpc parameters applied to ltv siso systems. *American Control Conference (ACC)*, pages 1534–1539, 2018.

[167] Marwa Turki, Nicolas Langlois, and Adnan Yassine. A tuning approach for constrained mpc : Nominal stability ensured and energy consumption optimized. In *Systems, Process and Control (ICSPC), 2017 IEEE Conference on*, pages 1–6. IEEE, 2017.

[168] D. W. Clarke T.W. Yoon. Observer design in receding-horizon predictive control. 1995.

[169] G Valencia-Palomo and JA Rossiter. Programmable logic controller implementation of an auto-tuned predictive control based on minimal plant information. *ISA transactions*, 50(1) :92–100, 2011.

[170] Tom Van Herpe, Niels Haverbeke, Bert Pluymers, Greet Van den Berghe, and Bart De Moor. The application of model predictive control to normalize glycemia of critically ill patients. In *Control Conference (ECC), 2007 European*, pages 3116–3123. IEEE, 2007.

[171] Hua O Wang, Kazuo Tanaka, and Michael F Griffin. An approach to fuzzy control of nonlinear systems : Stability and design issues. *IEEE transactions on fuzzy systems*, 4(1) :14–23, 1996.

[172] Liuping Wang. *Model predictive control system design and implementation using MATLAB®*. Springer Science & Business Media, 2009.

[173] Hoff M Widrow B. Adaptive switching circuits. *Technical report 1553-1, Stanford Electron. Labs*, 1960.

[174] Willy Wojsznis, John Gudaz, Terry Blevins, and Ashish Mehta. Practical approach to tuning mpc. *ISA transactions*, 42(1) :149–162, 2003.

[175] BE Ydstie. Extended horizon adaptive control. *IFAC Proceedings Volumes*, 17(2) :911–915, 1984.

[176] Qin Y Ye Y. Qr factorization based incremental extreme learning machine with growth of hidden nodes. *Pattern Reco-gnition Letters*, 2015.

[177] Xiaohong Yin, Shaoyuan Li, Ning Li, Yi Zheng, and Wenjian Cai. Energy-saving-oriented control strategy for vapor compression refrigeration cycle systems. In *Industrial Electronics and Applications (ICIEA), 2014 IEEE 9th Conference on*, pages 1561–1566. IEEE, 2014.

[178] Ye Y Wang G Yongjiao S, Yuangen C. Method pruning extreme learning machine. 2017.

[179] T. Youssef, M. Chadli, H.R. Karimi, and R. Wangd. Actuator and sensor faults estimation based on proportional integral observer for ts fuzzy model. *Journal of the Franklin Institute*, 354(6) :2524–2542, 2017.