



DOCTORAL THESIS

3rd Cycle Doctoral (D-LMD)

Presented by

HEDROUG Mohamed Elamine

With a view to obtaining the doctoral diploma in 3rd Cycle Doctoral (D-LMD)

Branch: Automatic

Specialty: Automatic

Topic

Fuzzy Predictive Control with Constraints for Dynamic Systems

Supported, on 18 /06/ 2026, before the jury composed of:

Last and first name	Grade	Institution of affiliation	Designation
Mr Bessissa Lakhdar	Professeur	Université de Djelfa	President
Ms Bdirina El Khansa	Professeur	Université de Djelfa	Supervisor
Mr Guesmi Kamel	Professeur	Université de Djelfa	Co-Supervisor
Ms Azzouzi Messaouda	Professeur	Université de Djelfa	Examiner
Mr Henini Noureddine	Professeur	Université de Médéa	Examiner
Mr Aissat Sid Ali	MCA	Université de Djelfa	Examiner

Djelfa University, FST - 2026



THESE DE DOCTORAT

Doctorat 3^{ème} Cycle (D-LMD)

Présentée par

HEDROUG Mohamed Elamine

En vue de l'obtention du diplôme de Docteur en 3^{ème} Cycle D-LMD

Filière : Automatique

Spécialité : Automatique

Thème

Commande prédictive floue avec contraintes des systèmes dynamiques

Soutenue publiquement, le 18 /06/2026, devant le jury composé de :

Nom et Prénom	Grade	Etablissement de rattachement	Désignation
Mr Bessissa Lakhdar	Professeur	Université de Djelfa	Président
Mme Bdirina El Khansa	Professeur	Université de Djelfa	Directrice de thèse
Mr Guesmi Kamel	Professeur	Université de Djelfa	Co-directeur de thèse
Mme Azzouzi Messaouda	Professeur	Université de Djelfa	Examinatrice
Mr Henini Nouredine	Professeur	Université de Médéa	Examineur
Mr Aissat Sid Ali	MCA	Université de Djelfa	Examineur

Je dédie ce travail modeste à toutes les personnes qui ont eu, de près ou de loin, une influence sur mon parcours académique et humain.

Je le dédie en premier lieu à ma chère mère, que Dieu lui accorde sa miséricorde qui fut la prière sincère, la tendresse infinie et la lumière ayant éclairé mon chemin, même après son départ. Je lui offre cet effort en espérant que Dieu l'inscrive dans la balance de ses bonnes actions et qu'Il lui accorde la plus haute récompense.

À ma grande famille pour le soutien moral, les conseils sincères et les encouragements constants qui ont grandement contribué à renforcer ma détermination et à poursuivre mon parcours.

À mes enseignants et professeurs qui ne m'ont jamais ménagé ni leur savoir ni leurs conseils, et qui ont largement contribué à forger ma réflexion académique et à construire ma personnalité scientifique. Je leur exprime mon profond respect et ma sincère gratitude.

À mes camarades d'études avec lesquels j'ai partagé les efforts et les défis du parcours universitaire, et dont l'esprit de coopération a permis de surmonter de nombreuses difficultés. Merci pour la solidarité et le soutien mutuel.

À mes amis qui ont été un soutien moral précieux et des compagnons de route, et qui ne m'ont jamais refusé une parole bienveillante ou un encouragement sincère dans les moments difficiles.

Enfin, à toutes les personnes qui ont cru en moi, m'ont soutenu par une prière sincère, un mot encourageant ou un geste humain noble je dédie ce travail, qui représente le fruit d'un effort collectif et d'une influence partagée.

Ce mémoire n'est pas uniquement le résultat d'un effort individuel, mais le reflet du soutien, de l'affection et de la confiance de nombreuses personnes, à qui j'adresse mes plus vifs remerciements et ma profonde reconnaissance.

Mohamed Elamine HEDROUG

Djelfa 2026

Remerciements

Louange à Allah, Seigneur des mondes, par la grâce duquel s'accomplissent les bonnes œuvres. Je Le remercie pour la force, la patience et la persévérance qu'Il m'a accordées tout au long de mon parcours universitaire et qui m'ont permis d'achever ce travail.

J'exprime ensuite mes sincères remerciements à mon encadrante, **Pr. Bdirina El Khansa**, pour son encadrement scientifique, ses conseils précieux et son suivi constant, qui ont grandement contribué à la réalisation de ce mémoire dans les meilleures conditions.

Je tiens également à adresser mes vifs remerciements à l'assistant de l'encadrement, **Pr. Kamel Guesmi**, pour son soutien scientifique, ses orientations pertinentes et ses remarques constructives qui ont permis d'améliorer la qualité de ce travail.

Je remercie vivement les membres du jury pour l'honneur qu'ils me font en acceptant d'évaluer ce mémoire, ainsi que pour leurs observations judicieuses et leurs recommandations enrichissantes.

Enfin, j'exprime ma profonde gratitude à l'ensemble des enseignants qui ont contribué à ma formation académique au cours de mon parcours universitaire et dont les enseignements ont été d'une grande valeur.

Mohamed Elamine HEDROUG

Djelfa 2026

تندرج هذه الدراسة في إطار تطوير استراتيجيات تحكم متقدمة للأنظمة الديناميكية غير الخطية والمعقدة، التي تتميز بوجود لاختطيات، عدم يقين، وقيود تشغيلية تجعل التحكم التقليدي محدود الفعالية. في هذا السياق، تتمحور إشكالية البحث حول كيفية تحسين أداء التحكم التنبؤي القائم على النموذج (MPC) وتعزيز مرونته وقدرته على التكيف مع تغير ظروف التشغيل، خاصة عند التعامل مع أنظمة غير خطية أو ذات معاملات متغيرة زمنياً. تعتمد المنهجية المقترحة على الدمج المنهجي بين التحكم التنبؤي والمنطق الضبابي من خلال نماذج تاكاغي-سوجينو (Takagi-Sugeno)، حيث تم إدماج نظام استدلال ضبابي لضبط أوزان دالة الكلفة في الـ MPC بشكل ديناميكي، بما يحقق توازناً فعالاً بين دقة التتبع وجهد التحكم. كما تم تطوير نماذج تمثيلية تحليلية، متعددة الأوجه (Polytopic). أظهرت النتائج، من خلال تطبيقات محاكاة على روبوت متنقل تفاضلي، تفوق التحكم التنبؤي الضبابي المقترح من حيث دقة تتبع المسار، مقاومة الاضطرابات، تقليل زمن الاستقرار، وتحسين الكفاءة الطاقوية مقارنة بالتحكم التنبؤي التقليدي وغير الخطي. وتخلص الدراسة إلى أن الإطار المقترح يمثل حلاً فعالاً ومرناً للتحكم في الأنظمة غير الخطية المعقدة.

الكلمات المفتاحية: التحكم التنبؤي الضبابي (FMPC)، أنظمة تاكاغي-سوجينو الضبابية، الأنظمة غير الخطية، النمذجة متعددة الأوجه، الروبوت المتنقل التفاضلي، تتبع المسار.

Résumé

Cette étude porte sur le développement d'une stratégie de commande avancée destinée aux systèmes dynamiques non linéaires et complexes, caractérisés par des non-linéarités, des incertitudes et des contraintes opérationnelles. Elle vise à améliorer les performances de la commande prédictive basée sur un modèle (MPC) et à renforcer sa capacité d'adaptation face aux variations des conditions de fonctionnement. La méthodologie proposée repose sur l'intégration du contrôle prédictif et de la logique floue à travers des modèles de Takagi-Sugeno (T-S), avec un ajustement dynamique des pondérations de la fonction de coût afin d'assurer un compromis efficace entre précision de suivi et effort de commande. Des modèles analytiques et polytopiques sont également développés. Les résultats de simulation obtenus sur un robot mobile différentiel démontrent la supériorité de l'approche proposée en termes de précision du suivi, de robustesse et d'efficacité énergétique, confirmant ainsi l'efficacité et la flexibilité du cadre proposé pour la commande des systèmes non linéaires complexes.

Mots-clés : Commande Prédictive Floue (FMPC), Systèmes Flous de Takagi-Sugeno, Systèmes Non Linéaires, Modélisation Polytopique, Robot Mobile Différentiel, Suivi de Trajectoire.

Abstract

This study focuses on the development of an advanced control strategy for nonlinear and complex dynamic systems characterized by nonlinearities, uncertainties, and operational constraints. It aims to improve the performance of Model Predictive Control (MPC) and to enhance its adaptability to variations in operating conditions. The proposed methodology is based on the integration of predictive control and fuzzy logic through Takagi-Sugeno (T-S) models, with a dynamic adjustment of the cost function weights to ensure an effective trade-off between tracking accuracy and control effort. Analytical and polytopic models are also developed. Simulation results obtained on a differential mobile robot demonstrate the superiority of the proposed approach in terms of tracking accuracy, robustness, and energy efficiency, there by confirming the effectiveness and flexibility of the proposed framework for controlling complex nonlinear systems.

Keywords: Fuzzy Model Predictive Control (FMPC), Takagi-Sugeno Fuzzy Systems, Nonlinear Systems, Polytopic Modeling, Differential Mobile Robot, Trajectory Tracking.

TABLE DES MATIÈRES

Liste des abréviations	J
Nomenclature	K
Liste des figures	L
Liste des tableaux	Q
Introduction générale	1
CHAPITRE I: Commande prédictive floue: Fondements, classification et stratégies d'intégration	3
I.1. Introduction.....	4
I.2. Concepts fondamentaux	6
I.2.1. Principes fondamentaux de la commande prédictive basée sur un modèle	6
I.2.2. Fondements de la logique floue et des systèmes d'inférence	15
I.3. Taxonomie des travaux de recherche en contrôle prédictif flou	31
I.3.1. Classification selon le type du système flou utilisé	31
I.3.2. Classification selon la nature du système contrôlé	34
I.4. Architectures et stratégies d'intégration du contrôle prédictif flou	37
I.4.1. Intégration de modèles flous comme prédicteurs internes.....	37
I.4.2. Gestion des contraintes floues dans le contrôle prédictif	43
I.4.3. Contrôle prédictif flou adaptatif	45
I.4.4. MPC à gain flou programmé.....	53
I.5. Conclusion	55
CHAPITRE II: Commande prédictive basée sur la logique floue pour les systèmes linéaires	56
II.1. Introduction.....	57
II.2. MPC pour les systèmes à paramètres variant linéairement (basé sur les modèles flous T-S)....	58
II.2.1. Modèle flou Takagi–Sugeno pour les systèmes linéaires variant dans le temps	58
II.2.2. Modèle de prédiction	59
II.2.3. Fonction de coût et contraintes	61
II.2.4. Solution du problème d'optimisation.....	62
II.3. Réglage dynamique des poids de la commande prédictive (MPC)	63
II.3.1. Principe de l'ajustement dynamique.....	63
II.3.2. Procédure d'ajustement des poids basée sur le flou	64
II.3.3. Intégration dans la formulation de la MPC	64

II.3.4. Impact sur les performances du contrôle	65
II.4. Application: Commande prédictive floue d'un robot mobile	65
II.4.1. Description du système	65
II.4.2. Modélisation cinématique d'un robot mobile non-holonomique à deux roues.....	65
II.4.3. Modèle d'erreur cinématique	66
II.4.4. Contrôle prédictif Flou (CPF)	68
II.4.5. Preuve de stabilité.....	71
II.4.6. Résultats et discussion	72
II.5. Conclusion	93
CHAPITRE III: Construction unifiée des modelés Takagi–Sugeno.....	95
III.1. Introduction.....	96
III.2. Énoncé du problème et système non linéaire.....	96
III.3. Approche de modélisation floue de Takagi–Sugeno (TS) pour les systèmes non linéaires	97
III.3.1. Évaluation de l'erreur et fonction de coût quadratique moyenne (MSE).....	98
III.3.2. Identification des paramètres par l'algorithme de Levenberg–Marquardt (LM).....	99
III.3.3. Analyse de stabilité par inégalités matricielles linéaires (LMI):	100
III.3.4. Résultats numériques et validation par simulation	101
III.3.5. Phase d'entraînement	101
III.3.6. Validation du modèle avec un signal d'entrée inédit.....	104
III.3.7. Comparaison des réponses système réel / modèle TS	105
III.4. Transformation de secteur non linéaire et représentation polytopique	109
III.4.1. Principes de la transformation sectorielle non linéaire (NST)	109
III.4.2. Formulation générale des systèmes non linéaires.....	109
III.4.3. Hypothèses de bornage sectoriel des variables de planification	109
III.4.4. Représentation paramétrique dépendante des variables de planification	109
III.4.5. Décomposition en modèles polytopiques convexes.....	110
III.4.6. Construction des fonctions d'appartenance linéaires par secteur	110
III.4.7. Modèle polytopique équivalent pour l'analyse et la commande	110
III.4.8. Résultats numériques et validation par simulation	111
III.4.9. Avantages, limitations et perspectives d'application de l'approche NST	118
III.5. Identification de modèle flou Takagi–Sugeno basée sur les données en utilisant l'algorithme de Levenberg–Marquardt	118

III.5.1. Acquisition et préparation des données d'entraînement.....	118
III.5.2. Architecture du modèle flou T-S et fonctions d'appartenance gaussiennes	119
III.5.3. Initialisation des paramètres du modèle	119
III.5.4. Procédure d'optimisation de Levenberg-Marquardt	119
III.5.5. Critères d'évaluation et validation des performances.....	120
III.5.6. Résultats numériques et validation par simulation	120
III.6. Évaluation des techniques de modélisation de systèmes non linéaires: TS analytique, NST polytopique et modèles TS basés sur LM pilotés par les données.....	129
III.6.1. Modélisation floue de Takagi–Sugeno	131
III.6.2. Transformation de secteur non linéaire et représentation polytopique.....	131
III.6.3. Identification de modèle flou Takagi–Sugeno basée sur les données en utilisant l'algorithme de Levenberg–Marquardt	132
III.7. Conclusion	133
CHAPITRE IV: Représentation polytopique de Takagi–Sugeno et conception de commande prédictive pour les robots mobiles	134
IV.1. Introduction	135
IV.2. Modèle non linéaire du robot mobile différentiel.....	136
IV.2.1. Modèle cinématique non linéaire.....	136
IV.2.2. Variables d'état et d'entrée	136
IV.2.3. Modèle de dynamique des erreurs	136
IV.3. Construction du modèle polytopique de Takagi–Sugeno.....	137
IV.3.1. Modèle d'erreur non linéaire équivalent	138
IV.3.2. Définition des bornes de la variable (Z_l et Z_u)	138
IV.3.3. Génération de sommets (structure polytopique à 16 sommets)	139
IV.3.4. Calcul des fonctions d'appartenance	139
IV.3.5. Matrices du système pour chaque sommet (A_i, B_i).....	139
IV.3.6. Validation du modèle polytopique	140
IV.4. Comparaison entre les modèles polytopiques et non linéaires du robot mobile différentiel	140
IV.4.1. Conditions initiales et paramètres.....	141
IV.4.2. Comparaison des variables d'état e_x, e_y, e_θ	141
IV.4.3. Indices de performance (RMSE et ISE)	144
IV.5. Conception intégrée du contrôle prédictif basé sur le modèle de Takagi–Sugeno	145

IV.5.1. Dérivation de la matrice de Lyapunov pour la stabilité du modèle polytopique T–S.....	145
IV.5.2. Conception de MPC pour le système polytopique T–S.....	147
IV.6. Simulation et analyse comparative entre TS–MPC et NMPC.....	151
IV.6.1. Configuration du scénario de simulation.....	151
IV.6.2. Comparaison de la trajectoire réelle avec la trajectoire de référence.....	152
IV.6.3. Comparaison des erreurs de suivi e_x, e_y, e_θ entre les deux contrôleurs.....	153
IV.6.4. Comparaison des entrées de contrôle de référence et appliquées.....	154
IV.6.5. Analyse des indices de performance (ISE, RMSE).....	157
IV.6.6. Analyse du temps de calcul.....	158
IV.7. Conclusion.....	159
Conclusion générale.....	161
Annexes.....	163
Bibliographie.....	165
Liste des travaux.....	180

Liste des abréviations

Abréviation	Signification
MPC	Commande Prédictive par Modèle
FPC	Commande Prédictive Floue
LF	Logique Floue
FIS	Système d'Inférence Floue
MIMO	Système Multi-Entrées Multi-Sorties
MAC	Commande Algorithmique par Modèle
DMC	Commande par Matrice Dynamique
EHAC	Commande Adaptative à Horizon Étendu
EPSAC	Commande Auto-Adaptative à Prédiction Étendue
GPC	Commande Prédictive Généralisée
PFC	Commande Fonctionnelle Prédictive
CRHPC	Commande Prédictive à Horizon Glissant sous Contraintes
CPN	Commande Prédictive Non Linéaire
PID	Correcteur Proportionnel–Intégral–Dérivé
F1MPC	Commande Prédictive par Modèle Flou de Type 1
F2MPC	Commande Prédictive par Modèle Flou de Type 2
LPV	Systèmes à Paramètres Variant Linéairement
DDMR	Robot Mobile à Conduite Différentielle
NST	Transformation Sectorielle Non Linéaire
LM	Algorithme de Levenberg–Marquardt
LMI	Inégalités Matricielles Linéaires
PDC	Compensation Distribuée Parallèle
TS–MPC	Contrôle Prédictif par Modèle basé sur Takagi–Sugeno
ISE	Intégrale de l'Erreur Quadratique
IAE	Intégrale de l'Erreur Absolue
ITSE	Intégrale de l'Erreur Quadratique Temps
ITAE	Intégrale de l'Erreur Absolue Temps
RMSE	Racine de l'Erreur Quadratique Moyenne
ISA	Intégrale de l'Action de Commande au Carré
WRLS	Moindres Carrés Récursifs Pondérés
ANFIS	Système d'Inférence Neuro-Flou Adaptatif
PSO	Optimisation par Essaim de Particules
RST	Régulateur R-S-T

Nomenclature

Symbole	Désignation
x	Variable d'état
y	Sortie du système
u	Entrée / commande
Δu	Incrément de commande
y_{ref}	Trajectoire de référence
\hat{y}	Sortie prédate
N_p	Horizon de prédiction
N_c	Horizon de commande
T_e	Période d'échantillonnage
Q	Matrice de pondération de l'erreur
R	Matrice de pondération de l'effort de commande
$\mu(x)$	Fonction d'appartenance
$\mu_A(x)$	Degré d'appartenance de x à l'ensemble flou A
α_i	Degré d'activation de la règle i
J	Fonction de coût (critère de performance)
$A(k), b(k)$	Matrices du système variant dans le temps
A_i, b_i	Matrices locales associées à la règle floue i
$\mu_i(k)$	Degré d'appartenance de la règle floue i
$\xi_i(k)$	Fonction de pondération floue normalisée
$Q(k), R(k)$	Matrices de pondération de l'état et de la commande (variables dans le temps)
v	Vitesse linéaire du robot
w	Vitesse angulaire du robot
v_{ref}	Vitesse linéaire de référence
ω_{ref}	Vitesse angulaire de référence
$p = [x, y, \theta]^T$	Vecteur de posture (position et orientation)
p_r	Posture de référence
e_x, e_y, e_θ	Composantes de l'erreur de posture
D_{ed}	Distance entre le robot réel et le robot virtuel
$f(x, u)$	Fonction dynamique non linéaire du système
R	Nombre de règles floues (modèles locaux)
z	Variable de prémisse (variable de planification)
c_i	Centre de la fonction d'appartenance gaussienne
σ_i	Écart-type (largeur) de la fonction gaussienne
$J(\theta)$	Fonction de coût (erreur)
$\rho_i(t)$	Variables de planification dans la NST
$\rho_i^{\min}, \rho_i^{\max}$	Bornes inférieure et supérieure des variables de planification
\mathcal{D}	Ensemble de données d'apprentissage
Δt	Période d'échantillonnage

Liste des figures

Figure I.1. Construction de l'action de commande à l'aide de l'approche de commande prédictive basée sur modèle (MPC).	7
Figure I.2. Philosophie de la commande prédictive.	8
Figure I.3. Le mécanisme de fonctionnement de la commande prédictive.....	8
Figure I.4. Schéma de principe de la commande prédictive.	9
Figure I.5. Généalogie des algorithmes de la MPC linéaire.....	11
Figure I.6. Généalogie des approches de la commande prédictive non linéaire (CPN).	12
Figure I.7. Illustration des concepts d'appartenance dans les ensembles classiques et flous.....	16
Figure I.8. Structure d'un sous-ensemble flou normalisé.	17
Figure I.9. Comparaison des fonctions d'appartenance floues.....	18
Figure I.10. Représentation des opérations union, intersection et complément flous.	20
Figure I.11. Architecture d'un système expert flou.....	21
Figure I.12. Variable linguistique.....	23
Figure I.13. Structure d'un système d'inférence flou.....	24
Figure I.14. Le processus de fuzzification dans un SIF de type Mamdani.	25
Figure I.15. L'Activation des règles dans un SIF.....	26
Figure I.16. L'Agrégation des sorties floues dans un SIF.	27
Figure I.17. Conversion de la sortie floue en valeur nette (COG vs MM).	28
Figure I.18. Empreinte de l'Incertitude (EI) entre les fonctions d'appartenance supérieure et inférieure.....	29
Figure I.19. Fonctionnement et architecture d'un système de logique floue de type 2.....	30
Figure I.20. MPC basé sur un modèle flou avec compensation distribuée parallèle.	43
Figure I.21. Gestion des contraintes souples dans le contrôle prédictif flou.....	45
Figure I.22. Mécanisme général pour la mise à jour des paramètres du modèle local dans le contrôle prédictif adaptatif.....	48
Figure I.23. Mécanisme adaptatif pour les mises à jour des fonctions d'appartenance dans le contrôle prédictif flou.....	50
Figure I.24. Stratégies automatiques de génération et d'élagage de règles floues dans la commande prédictive floue adaptative	51

Figure I.25. Structure de la commande prédictive à gain flou	54
Figure II.1. Robot mobile à conduite différentielle.	66
Figure II.2. Évolution de la transformation de l'erreur de suivi du robot.	66
Figure II.3. Schéma bloc de l'approche proposees.....	71
Figure II.4. Caractérisation des fonctions d'appartenance pour le modèle flou T-S.....	73
Figure II.5. Fonctions d'appartenance des entrées du système flou T-S de type-1.	74
Figure II.6. Fonctions d'appartenance des entrées du système flou T-S de type-2.	74
Figure II.7. Réponse du robot sous F1MPC et MPC sur une trajectoire en forme de 8.	76
Figure II.8. Erreurs de suivi utilisant l'approche MPC.	76
Figure II.9. Erreurs de suivi utilisant l'approche F1MPC.....	77
Figure II.10. Vitesses linéaire et angulaire utilisant l'approche MPC.	77
Figure II.11. Vitesses linéaire et angulaire utilisant l'approche F1MPC.	78
Figure II.12. Résultats du suivi de trajectoire circulaire en utilisant les méthodes de contrôle prédictif et les méthodes proposées de contrôle prédictif par modèle flou (F1MPC et F2MPC).	79
Figure II.13. Erreurs de suivi utilisant l'approche MPC.	80
Figure II.14. Erreurs de suivi utilisant l'approche F1MPC.	80
Figure II.15. Erreurs de suivi utilisant l'approche F2MPC.	81
Figure II.16. Vitesses linéaire et angulaire utilisant l'approche MPC.	81
Figure II.17. Vitesses linéaire et angulaire utilisant l'approche F1MPC.	82
Figure II.18. Vitesses linéaire et angulaire utilisant l'approche F2MPC.	82
Figure II.19. Suivi des trajectoires circulaires et sinusoïdales à l'aide de contrôleurs MPC flous de type 1 et de type 2, ainsi que du contrôleur MPC.....	83
Figure II.20. Erreurs de suivi utilisant l'approche MPC.	84
Figure II.21. Erreurs de suivi utilisant l'approche F1MPC.	84
Figure II.22. Erreurs de suivi utilisant l'approche F2MPC.	85
Figure II.23. Vitesses linéaire et angulaire utilisant l'approche F1MPC.	85
Figure II.24. Vitesses linéaire et angulaire utilisant l'approche F2MPC.	86
Figure II.25. Vitesses linéaire et angulaire utilisant l'approche MPC.	86
Figure II.26. Robustesse du suivi des trajectoires circulaire et sinusoïdale.	89
Figure II.27. Erreurs de suivi utilisant l'approche MPC.	90
Figure II.28. Erreurs de suivi utilisant l'approche F1MPC.	90

Figure II.29. Erreurs de suivi utilisant l'approche F2MPC.	90
Figure II.30. Illustration du test de robustesse des vitesses angulaire et linéaire utilisant l'approche F1MPC.....	91
Figure II.31. Illustration du test de robustesse des vitesses angulaire et linéaire utilisant l'approche F2MPC.....	91
Figure II.32. Illustration du test de robustesse des vitesses angulaire et linéaire utilisant l'approche MPC.	92
Figure III.1. Signal d'excitation utilisé pour l'entraînement.....	102
Figure III.2. Comparaison des trajectoires réelles et du modèle T-S pour l'état x_1 (Phase d'entraînement)	103
Figure III.3. Comparaison des trajectoires réelles et du modèle T-S pour l'état x_2 (Phase d'entraînement)	103
Figure III.4. Fonctions d'appartenance gaussiennes des règles floues	104
Figure III.5. Signal d'entrée d'excitation de validation	105
Figure III.6. Comparaison des trajectoires réelles et du modèle T-S pour l'état x_1 (Phase de validation).....	106
Figure III.7. Comparaison des trajectoires réelles et du modèle T-S pour l'état x_2 (Phase de validation).....	106
Figure III.8. Trajectoire de l'erreur de validation pour l'état x_1	107
Figure III.9. Trajectoire de l'erreur de validation pour l'état x_2	107
Figure III.10. Portrait de phase du système réel et du modèle T-S (Phase de validation)	108
Figure III.11. Signal d'entrée (Cas 1).....	113
Figure III.12. Comparaison des trajectoires de $x_1(t)$ entre le système non linéaire et le modèle polytopique (Cas 1).....	113
Figure III.13. Comparaison des trajectoires de $x_2(t)$ entre le système non linéaire et le modèle polytopique (Cas 1).....	114
Figure III.14. Portrait de phase dans le plan (x_1, x_2) (Cas 1)	114
Figure III.15. Évolution de la fonction d'appartenance $\mu_i(x_2, u)$ pendant la simulation (Cas 1)	115
Figure III.16. Signal d'entrée (Cas 2).....	115
Figure III.17. Comparaison des trajectoires de $x_1(t)$ entre le système non linéaire et le modèle polytopique (Cas 2).....	116

Figure III.18. Comparaison des trajectoires de $x_2(t)$ entre le système non linéaire et le modèle polytopique (Cas 2).....	116
Figure III.19. Portrait de phase dans le plan (x_1, x_2) pour le cas 2	117
Figure III.20. Évolution des fonctions d'appartenance (Cas 2).....	117
Figure III.21. Évolution temporelle de l'état x_1 pour le système non linéaire réel et le modèle Takagi–Sugeno identifié (phase d'entraînement)	123
Figure III.22. Évolution temporelle de l'état x_2 pour le système non linéaire réel et le modèle Takagi–Sugeno identifié (phase d'entraînement).	124
Figure III.23. Comparaison des portraits de phase du système non linéaire réel et du modèle flou Takagi–Sugeno identifié pendant l'entraînement.....	124
Figure III.24. Fonctions d'appartenance gaussiennes des règles floues (phase d'entraînement).	125
Figure III.25. Signal d'entrée de validation $u_{\text{test}}(t)$ utilisé pour évaluer le modèle identifié	126
Figure III.26. Évolution temporelle de l'état x_1 pour le système non linéaire réel et le modèle Takagi–Sugeno identifié (phase de validation).	126
Figure III.27. Évolution temporelle de l'état x_2 pour le système non linéaire réel et le modèle Takagi–Sugeno identifié (phase de validation).	127
Figure III.28. Trajectoire d'erreur de validation pour l'état x_1	127
Figure III.29. Trajectoire d'erreur de validation pour l'état x_2	128
Figure III.30. Comparaison des portraits de phase en phase de validation	128
Figure IV.1. Comparaison de l'erreur de suivi $e_x(t)$ entre le modèle non linéaire et le modèle polytopique.	142
Figure IV.2. Comparaison de l'erreur de suivi $e_y(t)$ entre le modèle non linéaire et le modèle polytopique.	142
Figure IV.3. Comparaison de l'erreur de suivi d'orientation $e_\phi(t)$ entre le modèle non linéaire et le modèle polytopique T–S.....	143
Figure IV.4. Comparaison des erreurs instantanées entre le modèle non linéaire et le modèle polytopique.	143
Figure IV.5. Comparaison des portraits de phase de $e_y(t)$ par rapport à $e_x(t)$ pour les modèles T–S non linéaires et polytopiques.....	144
Figure IV.6. Diagramme bloc du cadre de contrôle TS–MPC pour le robot à entraînement différentiel.....	150

Figure IV.7. Comparaison entre la trajectoire de référence et la trajectoire réelle du robot obtenue en utilisant TS-MPC et NMPC.	152
Figure IV.8. Évolution de l'erreur de suivi longitudinal $e_x(t)$ en utilisant TS-MPC et NMPC.....	153
Figure IV.9. Évolution de l'erreur de suivi latéral $e_y(t)$ en utilisant TS-MPC et NMPC.....	153
Figure IV.10. Comparaison de l'erreur de cap $e_\theta(t)$ entre TS-MPC et NMPC.....	154
Figure IV.11. Comparaison de la vitesse linéaire de référence $v(t)$ avec l'entrée de contrôle générée par le contrôleur TS-MPC.....	154
Figure IV.12. Comparaison de la vitesse angulaire de référence $\omega(t)$ et de l'entrée de contrôle TS-MPC	155
Figure IV.13. Comparaison de la vitesse linéaire de référence $v(t)$ avec l'entrée de contrôle générée par le contrôleur NMPC.....	156
Figure IV.14. Comparaison de la vitesse angulaire de référence $\omega(t)$ et de l'entrée de contrôle NMPC.....	156
Figure IV.15. Comparaison des performances basée sur l'erreur quadratique moyenne (RMSE) des erreurs de suivi pour TS-MPC et NMPC.....	157
Figure IV.16. Temps de calcul par étape pour TS-MPC et NMPC sur l'horizon de simulation complet.	158

Liste des tableaux

Tableau I.1. Commande prédictive linéaire: évolution et variantes.	10
Tableau I.2. Commande prédictive non linéaire: évolution et variantes.	11
Tableau I.3. Avantages et Inconvénients de la commande prédictive (MPC).....	14
Tableau I.4. Récapitulatif des fonctions d'appartenance floues: formules et applications typiques.	18
Tableau I.5. Les différents modèles flous et leurs propriétés essentielles.	28
Tableau I.6. Méthodes d'adaptation dans les applications de commande prédictive basée sur un modèle flou.	52
Tableau II.1. Règles floues pour Q et R	75
Tableau II.2. Valeurs des sorties linguistiques.	75
Tableau II.3. Analyse des erreurs de suivi.	78
Tableau II.4. Comparaison des indices de performance des erreurs des contrôleurs.	87
Tableau II.5. Évaluation de l'intensité d'action de commande à l'aide de l'ISA et de l'effort de commande.....	88
Tableau II.6. Comparaison des indices de performance des erreurs des contrôleurs.	93
Tableau II.7. Évaluation de l'intensité d'action de commande à l'aide de l'ISA et de l'effort de commande.....	93
Tableau III.1. Paramètres optimisés pour le modèle flou (5 règles)	102
Tableau III.2. Paramètres optimisés pour le modèle flou (5 règles)	122
Tableau III.3. Comparaison des modèles T–S analytiques, NST polytopiques et T–S basés sur LM en termes de structure, de complexité et de précision de modélisation.	130
Tableau IV.1. Indices de performance pour les variables d'erreur (RMSE et ISE).....	145

Introduction générale

Les systèmes dynamiques modernes dans les procédés industriels modernes caractérisent par leur complexité structurelle, leur comportement non linéaire et leur sensibilité aux variations temporelles des variables opérationnelles [1]. De ce fait, la conception et l'application de méthodes de contrôle avancées sont essentielles pour garantir la performance et la sécurité, et améliorer l'efficacité [2]. Simultanément, l'exigence de mécanismes de commande sophistiqués, aptes à assurer la stabilité, le suivi rigoureux et la conformité aux impératifs opérationnels, s'intensifie, notamment dans les domaines de la robotique, de l'industrie et de la conduite autonome [3]. Parmi ces méthodologies, la commande prédictive basée sur un modèle (Model Predictive Control, MPC) se distingue par une efficacité notable, attribuable à sa capacité d'anticiper le comportement futur du système au moyen d'un modèle dynamique, de résoudre un problème d'optimisation à chaque itération, et d'intégrer de manière explicite les contraintes au sein de l'algorithme de commande [4].

Néanmoins, la performance optimale du contrôleur MPC est étroitement corrélée à la justesse du modèle employé, notamment dans le cas de systèmes non linéaires ou à paramètres variant (LPV), pour lesquels une représentation linéaire simple des dynamiques réelles s'avère complexe [5]. Afin de pallier ces difficultés, le recours aux modèles flous de type Takagi–Sugeno (T–S) s'est avéré être une approche pertinente, permettant de transformer les systèmes non linéaires en une combinaison convexe de plusieurs modèles linéaires locaux, ce qui offre la possibilité d'exploiter les outils d'analyse linéaire avancés, tels que les inégalités matricielles linéaires (Linear Matrix Inequalities, LMI), et de faciliter l'utilisation du MPC (Model Predictive Control) dans un contexte stable et efficient [6].

En outre, la logique floue constitue un moyen intrinsèque de gérer l'incertitude, le bruit et l'approximation, et permet l'intégration de l'expertise humaine dans les mécanismes de commande. L'intégration du contrôle prédictif à base de modèle (Model Predictive Control, MPC) et de la logique floue a mené au développement d'une catégorie avancée de contrôleurs, désignée sous le terme de contrôle prédictif flou (Fuzzy Model Predictive Control, FMPC), permettant l'obtention de performances supérieures, d'une réponse plus atténuée et d'une capacité d'adaptation améliorée comparativement au contrôle prédictif traditionnel.

Dans ce contexte, la présente thèse a pour objectif d'élaborer un cadre exhaustif pour la conception d'un contrôleur prédictif flou destiné aux systèmes dynamiques soumis à des contraintes, avec une étude appliquée spécifique au robot mobile différentiel, lequel représente un modèle emblématique dans les applications de navigation autonome. Les travaux de cette thèse sont scindés en quatre chapitres et

articulés comme suit: Afin de réaliser cet objectif, la présente note débute par une présentation exhaustive de l'état de l'art dans son **premier chapitre**, lequel est consacré à l'examen de l'évolution historique et pratique du contrôle prédictif, de la logique floue, ainsi que des méthodes modernes de contrôle prédictif flou, de leurs domaines d'application et de leurs avantages respectifs.

Subséquentement, **le deuxième chapitre** est dédié à l'étude de la commande prédictive floue appliquée aux systèmes linéaires ou à paramètres variables, en mettant l'accent sur l'intégration des systèmes flous dans la structure de la commande prédictive afin d'ajuster les pondérations de coût et d'améliorer la poursuite de trajectoire. Ce chapitre expose également une application impliquant un robot mobile, laquelle illustre l'incidence de la logique floue sur l'amélioration de la qualité du contrôle en comparaison avec le contrôle prédictif traditionnel, tant en ce qui concerne la précision du suivi que la régularité des signaux de contrôle, ou encore la capacité de résistance aux perturbations.

Le troisième chapitre est consacré à la modélisation des systèmes non linéaires, en utilisant le formalisme de Takagi–Sugeno, et propose une méthodologie structurée pour reformuler le système non linéaire sous une forme adéquate pour l'analyse et la commande. Le chapitre inclut également une analyse comparative destinée à évaluer la fidélité de cette représentation et son aptitude à préserver le comportement effectif du système.

Le quatrième chapitre aborde la modélisation du robot mobile différentiel, l'élaboration d'un modèle d'erreurs de suivi, et sa représentation sous forme floue multi-modèle afin de permettre l'exploitation d'outils de commande linéaire. Le chapitre expose également la conception d'un contrôleur prédictif fondé sur cette représentation, et se conclut par une comparaison méthodique entre ce contrôleur et le contrôle prédictif non linéaire, dans le but de souligner les distinctions au niveau de la structure et des mécanismes de fonctionnement entre les deux approches dans le contexte du contrôle avancé.

Par le biais de cette séquence, combinant l'étude théorique, la modélisation, l'analyse et le développement d'algorithmes de contrôle, cette note établit un cadre méthodologique pour la conception de contrôleurs prédictifs flous destinés aux systèmes dynamiques, et illustre leur application à un système robotique mobile. Le présent travail met en évidence le potentiel d'application de tels modèles au sein des systèmes de contrôle, notamment dans les situations exigeant une prise en compte effective des non-linéarités et des variations dynamiques du système.

Des conclusions sont finalement tirées à l'issue de ces quatre chapitres, en proposant des améliorations à apporter à ces travaux de thèse ainsi que des pistes de recherche qui pourraient utilement compléter ce travail.

CHAPITRE I

COMMANDE PREDICTIVE FLOUE : FONDEMENTS,
CLASSIFICATION ET STRATEGIES D'INTEGRATION

I.1. Introduction

Les systèmes environnementaux, industriels et technologiques ont des dynamiques complexes qui sont difficiles à prévoir ou à décrire, rendant leur comportement difficile à comprendre en évaluant leurs composants séparés. Ils sont difficiles à contrôler en raison de leurs non-linéarités et de leur sensibilité aux conditions initiales. En raison des incertitudes intrinsèques, des comportements émergents tels que l'auto-organisation, la sensibilité aux conditions initiales, et une modélisation inexacte, les systèmes complexes, non linéaires, et mal spécifiés sont difficiles à contrôler. Étant donné que les processus non linéaires ont peu de traits communs, leur modélisation et leur identification sont difficiles. Un des objectifs de l'étude dans ce domaine est d'améliorer nos capacités de modélisation pour caractériser une large classe de systèmes non linéaires. "Quelle est la stabilité des dynamiques ?" est une question courante dans l'étude des dynamiques des systèmes complexes. Pouvons-nous réguler le système pour maintenir ses dynamiques dans un bon bassin d'attraction ou le sortir de mauvais ? De nombreux systèmes dépendent de la stabilité et de la contrôlabilité [1]. Les dynamiques non linéaires provenant des boucles de rétroaction rendent les sorties imprévisibles et les systèmes adaptatifs résistants à la modification. Bien que la théorie du contrôle classique se soit concentrée sur la régulation des systèmes dynamiques individuels, les défis modernes impliquent de coordonner et de diriger de grands groupes de systèmes interconnectés [7]. Les systèmes mal spécifiés ont de nombreux composants interagissant et des exigences floues, ce qui rend l'analyse, la conception et l'intervention difficiles. La non-linéarité, où de petits changements d'entrées peuvent provoquer de grands changements imprévus ou chaotiques dans les sorties à cause des boucles de rétroaction et des interactions de haut ordre ; la sensibilité aux conditions initiales, où l'état futur d'un système peut être très sensible à de petites variations de son point de départ, introduisant de l'aléa ; et la difficulté de modélisation, car capturer précisément les dynamiques complexes des systèmes non linéaires est une tâche difficile.

Dans ce cas, le Contrôle par Modèle Prédictif (MPC) a été adopté comme une technique de contrôle largement utilisée pour gérer des systèmes multivariables complexes. Un avantage fondamental du MPC est qu'il intègre directement les contraintes opérationnelles — telles que les limites des actionneurs et la disponibilité des ressources sur les entrées, sorties et états — directement dans le problème d'optimisation en ligne, contrairement aux contrôleurs conventionnels, qui nécessitent généralement des solutions ad-hoc pour les contraintes [4]. En utilisant un modèle pour prédire le comportement futur du système, le MPC agit de manière anticipatoire, lui permettant de surpasser les contrôleurs réactifs, en particulier pour les processus avec des retards significatifs. Cela garantit que les actions de contrôle générées respectent toujours ces limites, ce qui est crucial pour une exploitation sûre et efficace du

système. Il effectue une optimisation dynamique en temps réel d'une fonction de coût définie par l'utilisateur, telle que la minimisation de l'erreur de suivi, déterminant la séquence de commande optimale plutôt que de simplement fournir un retour réactif [4], [8]. Une telle fonctionnalité permet de calculer les entrées de contrôle optimales qui non seulement atteignent les points de consigne souhaités, mais optimisent également des mesures de performance telles que la consommation d'énergie, la qualité du produit ou l'efficacité économique [9]. De plus, le MPC est naturellement adapté aux systèmes Multi-Entrée Multi-Sortie (MIMO), gérant simultanément les interactions complexes et les couplages entre toutes les variables du processus, ce qui constitue une raison principale de son succès dans l'industrie des procédés [10].

La fusion du contrôle prédictif basé sur un modèle (MPC) et de la logique floue (FL) crée un cadre puissant connu sous le nom de contrôle prédictif flou (FPC), qui exploite efficacement les avantages des deux paradigmes. Le MPC fournit une approche structurée pour gérer les contraintes et optimiser le comportement futur du système sur un horizon de prédiction, tandis que la logique floue offre une méthodologie flexible pour incorporer les connaissances linguistiques, gérer l'incertitude, et exprimer les objectifs de performance de manière transparente et intuitive. Cette synergie permet au FPC de traiter les systèmes complexes, non linéaires et incertains de manière plus efficace que le MPC ou la logique floue seuls. En traduisant les objectifs de contrôle et les contraintes en objectifs flous et contraintes floues, le FPC permet une conception de contrôle plus interprétable par l'humain et robuste, conduisant souvent à une amélioration des performances en termes de vitesse de réponse, de réduction du dépassement, et de satisfaction des contraintes [5,11]. De plus, l'intégration de la prise de décision floue au sein de la structure du MPC facilite l'utilisation des opérateurs d'agrégation paramétrés, qui peuvent être ajustés pour équilibrer plusieurs objectifs conflictuels — tels que la précision de suivi, l'effort de contrôle, et la robustesse — à travers une formulation unifiée et calculablement gérable. Cette combinaison est particulièrement bénéfique dans les applications impliquant des retards temporels, des incertitudes de modèle, et des dynamiques non linéaires, comme le démontrent diverses études industrielles et académiques [12], [13] et [14].

Ce chapitre offre une exploration complète du Contrôle Prédictif Flou basé sur un modèle, débutant dans la première section avec un examen des concepts fondamentaux, détaillant les principes clés du contrôle prédictif basé sur un modèle (MPC) et les bases de la logique floue (FL) et des systèmes d'inférence floue (FIS) pour établir leur puissant potentiel synergique. La deuxième section explore les architectures de systèmes prédominantes, couvrant la conception du MPC flou à gain programmé et expliquant en profondeur l'approche dominante du contrôle prédictif basé sur un modèle flou Takagi-Sugeno (T-S). La

troisième section examine ensuite des sujets avancés et des tendances récentes qui repoussent les limites du domaine, tandis que la quatrième section propose une analyse critique des performances et un étalonnage de ces contrôleurs par rapport aux méthodes traditionnelles. La cinquième section illustre l'implémentation pratique à travers une large gamme d'applications réelles et d'études de cas. Enfin, le chapitre se conclut en esquisant les défis persistants et les problèmes ouverts pour tracer les directions futures de la recherche, suivi d'une synthèse des principales conclusions dans la conclusion.

1.2. Concepts fondamentaux

1.2.1. Principes fondamentaux de la commande prédictive basée sur un modèle

Le contrôle prédictif, également connu sous le nom de contrôle à horizon glissant, est l'une des techniques de contrôle avancées qui sont apparues ces dernières années pour répondre aux exigences toujours plus grandes du monde industriel concernant les boucles de contrôle à hautes performances pour des systèmes complexes [1]. Cette méthode est fondée sur le concept de prédiction des dynamiques du système dans le but d'optimiser le comportement futur [4].

Le contrôle prédictif basé sur modèle (Model Predictive Control, MPC), une méthodologie centrale dans ce domaine, utilise un modèle mathématique explicite pour prédire la réponse future d'un système. À chaque intervalle de contrôle, l'algorithme MPC résout une série de problèmes d'optimisation non linéaire, en répondant à des questions essentielles comme: dans quelle direction va le processus (estimation de l'état), où est-ce qu'il devrait aller (optimisation du point cible à l'état stationnaire), et quelles actions de contrôle permettront de le conduire au mieux vers l'état désiré (optimisation dynamique). Cette procédure est répétée en continu, ce qui assure que le système s'ajuste en temps réel aux conditions changeantes [7]. D'abord développé pour répondre aux besoins spécifiques de contrôle dans des industries comme les centrales électriques et les raffineries de pétrole, le MPC a étendu son domaine d'application à un large éventail de secteurs, allant de la chimie et l'agroalimentaire à l'automobile et l'aérospatiale. Sa capacité à traiter des processus complexes et multivariables avec des contraintes en a fait un outil indispensable dans l'automatisation moderne [4].

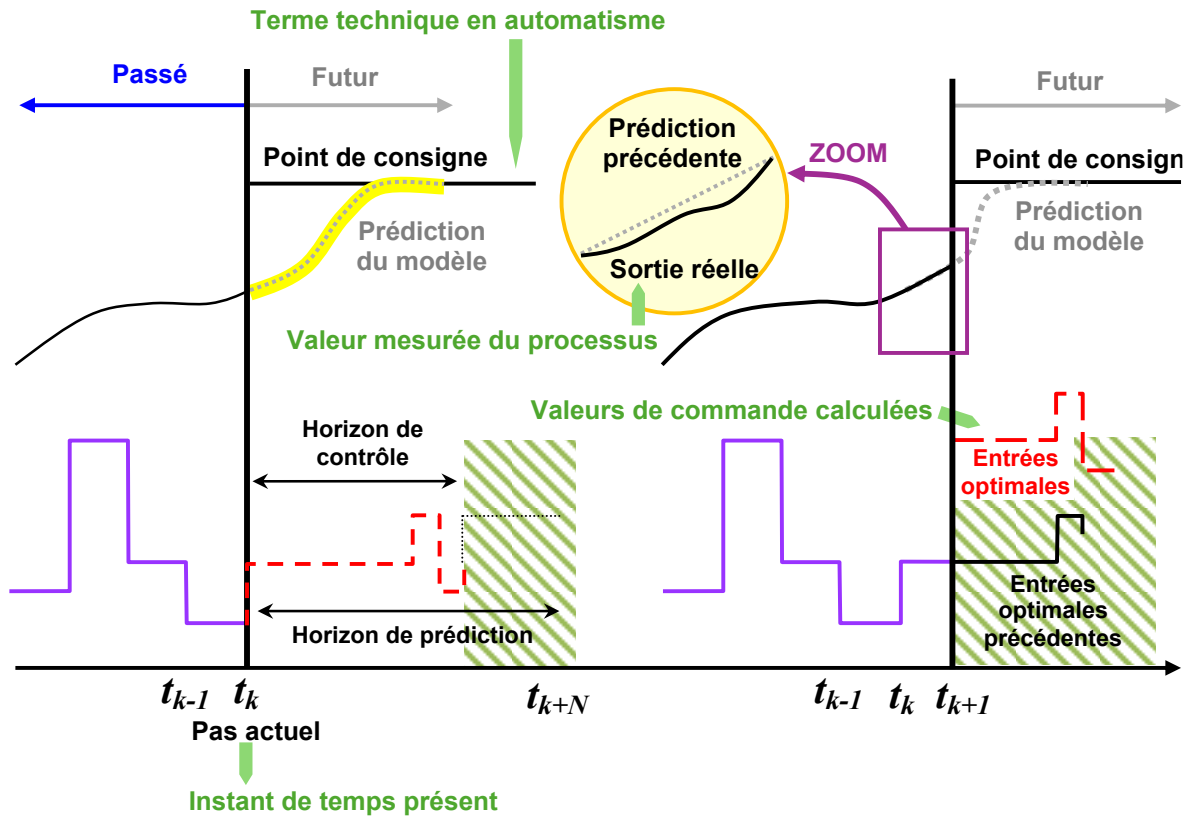


Figure I.1. Construction de l'action de commande à l'aide de l'approche de commande prédictive basée sur modèle (MPC).

Le principe « philosophique » de la commande prédictive est illustré à la Figure I.1. Dans un premier temps, un modèle du processus permet de prédire la sortie du système sur un horizon fini. Ensuite, en minimisant un critère de performance défini sur cet horizon, une séquence de commandes est calculée, mais seul le premier élément est appliqué au système. La même procédure est répétée à chaque période d'échantillonnage selon le principe de l'*horizon glissant*. L'objectif est de maintenir la sortie du système aussi proche que possible de la référence désirée, ce qui confère à la commande un caractère anticipatif [15-17].

Cette approche traduit en fait le comportement intuitif et anticipatif de l'être humain [18-19], par exemple lors de la conduite d'une voiture, du ski ou encore de la gestion d'un budget limité: une trajectoire est anticipée sur un horizon donné et des actions sont décidées afin de la suivre. Contrairement aux commandes classiques, qui se basent sur les erreurs passées entre la sortie et la consigne, la commande prédictive repose sur les *erreurs futures anticipées* [20-21], comme il est indiqué sur la figure I.2. [22].

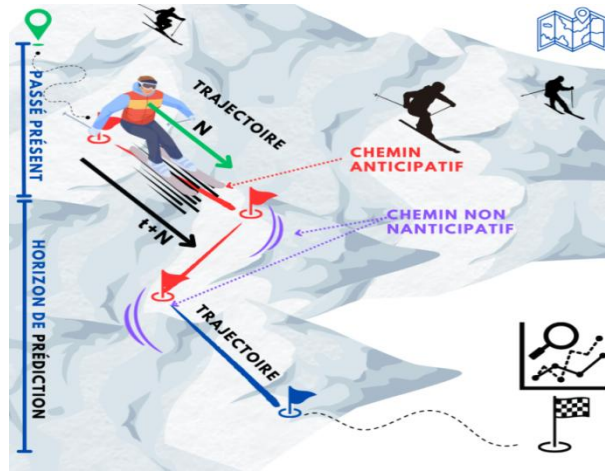


Figure I.2. Philosophie de la commande prédictive [22].

Les étapes générales de toute loi de commande prédictive peuvent être résumées comme suit:

1. À chaque instant d'échantillonnage k , les sorties futures $y_p(k + j/k)$ sont prédites sur un horizon de prédiction N_p à l'aide d'un modèle du système. Ces prédictions dépendent des entrées et sorties passées ainsi que des commandes futures $u(k + j/k)$ à déterminer.
2. Les signaux de commande futurs $u(k + j/k)$ sont calculés en résolvant un problème d'optimisation visant à minimiser un critère de performance, qui mesure l'écart entre les sorties prédites et la trajectoire de référence $y_{ref}(k + j)$, tout en réduisant l'effort de commande et en respectant les contraintes imposées.
3. Seul le premier élément de la commande optimale $u(k/k)$ est appliqué au système. À l'instant suivant, la procédure est répétée avec les mesures actualisées $y(k + 1)$, assurant ainsi un fonctionnement en **boucle fermée**, comme indiqué à la Figure I.3 [23-25].

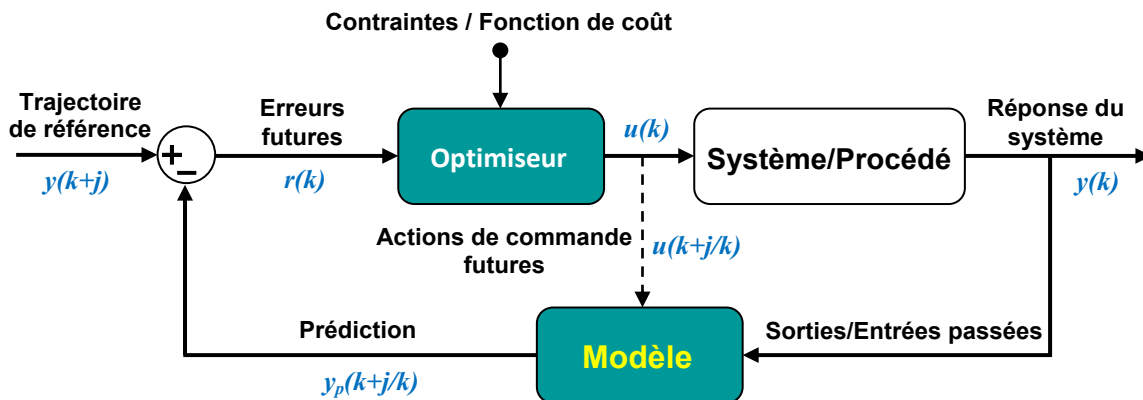


Figure I.3. Le mécanisme de fonctionnement de la commande prédictive [22].

De manière synthétique, la Figure I.4 illustre le principe général de la commande prédictive: créer un effet **anticipatif** en prédisant le comportement futur du système et en minimisant l'écart avec la trajectoire de référence, dans le cadre d'un problème de commande optimale sous contraintes et sur un horizon glissant.

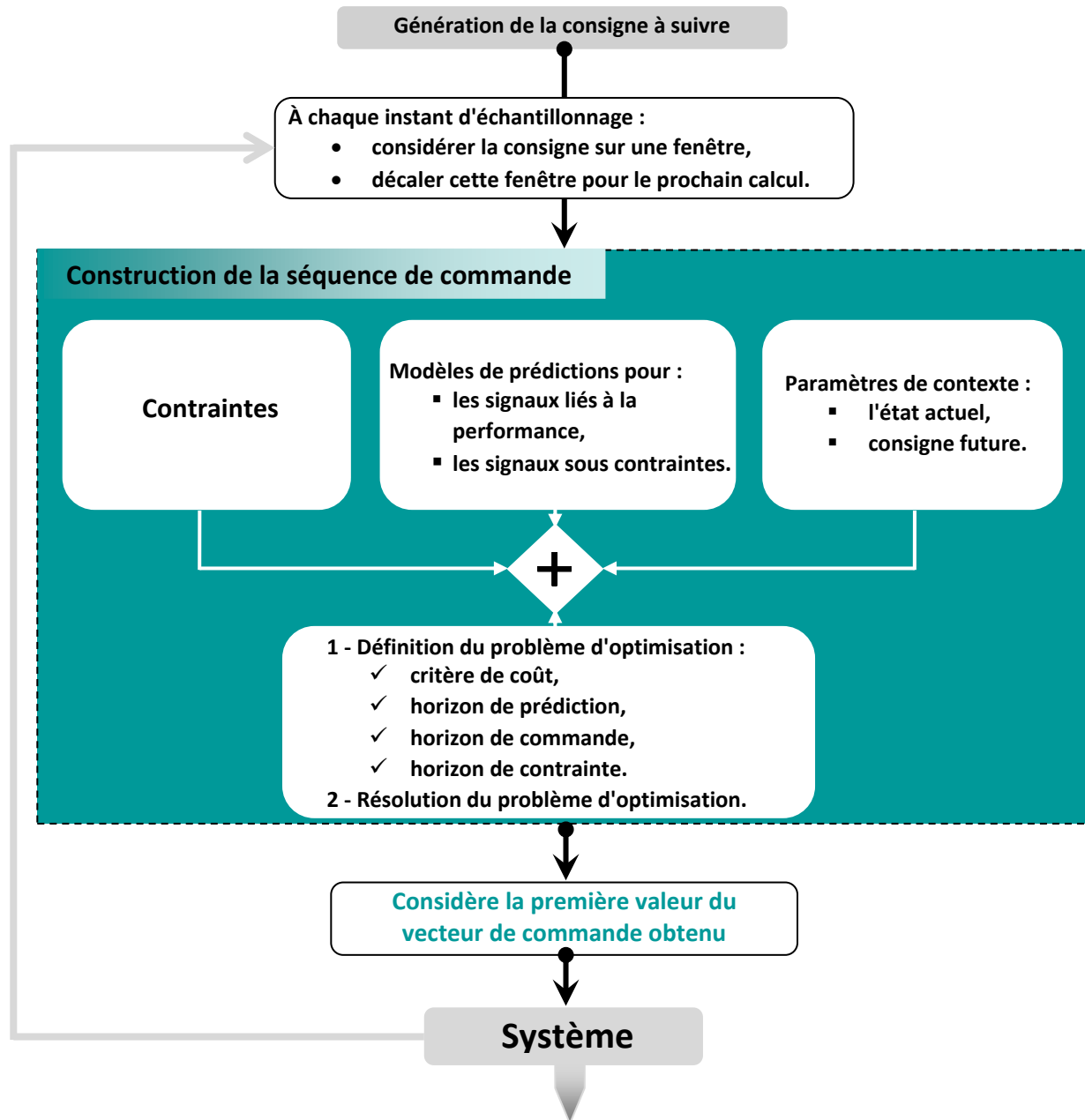


Figure I.4. Schéma de principe de la commande prédictive.

I.2.2.1. Historique, typologie et évolutions des algorithmes de commande prédictive

La commande prédictive (*Model Predictive Control – MPC*) est une technique avancée de régulation utilisée dans de nombreux domaines industriels et scientifiques. Elle repose sur la prédiction du comportement futur du système à l'aide d'un modèle, puis sur l'optimisation d'une séquence de commandes pour suivre une trajectoire de référence tout en respectant les contraintes du système. On distingue principalement deux grandes familles:

- **Commande prédictive linéaire:** basée sur des modèles linéaires et largement utilisée dans l'industrie.
- **Commande prédictive non linéaire (CPN):** adaptée à des systèmes plus complexes et non linéaires, avec des considérations de robustesse plus poussées.

Historiquement, la commande prédictive est apparue dans les années 1960 avec les travaux de Propoi ainsi que ceux de Lee et Markus.. Elle a connu un développement industriel dans les années 1970 avec le logiciel IDCOP de Richalet. De nombreuses variantes ont ensuite émergé, chacune répondant à des besoins spécifiques.

Tableau I.1. Commande prédictive linéaire: évolution et variantes.

Acronyme	Nom complet	Caractéristiques principales	Références
MAC	Model Algorithmique Control (anciennement MPHC)	Prédictions longues, référence imposée, multivariable	J. Richalet et al. [26]
DMC	Dynamic Matrix Control	Incrément de commande, horizon fini, moindres carrés	C.R.Cutler et al. [27]
EHAC	Extended Horizon Adaptive Control	Monovariante, prédiction via équation diophantienne	BE. Ydstie [28]
EPSAC	Extended Prediction Self-Adaptive Control	Non linéaire linéarisé, prédicteur sous-optimal	R.M.C. Keyser et al. [29]
GPC	Generalized Predictive Control	Basé sur le modèle CARIMA, applicable aux systèmes complexes	D.W.Clarke et al. [30]
PFC	Predictive Functional Control	SISO rapide, réglage simple mais sans rigueur formelle	J.Richalet et al. [31]
CRHPC	Constrained Receding Horizon Predictive Control	Contraintes terminales en horizon étendu	D. W. Clarke et al. [32]
MPC (état)	Model Predictive Control (formulation état)	Stabilité/robustesse via représentation d'état	N. L. Ricker [33]

Ces approches sont regroupées sous le terme générique de MPC linéaire, et ont été utilisées avec succès dans l'aérospatial, la pétrochimie, l'industrie du papier, etc. Voici une version réorganisée et modernisée de la Figure I.5 qui représente la généalogie des algorithmes de la MPC linéaire, représentée sous forme de graphe relationnel. Cette visualisation met en évidence les influences historiques entre les principales variantes d'algorithmes.

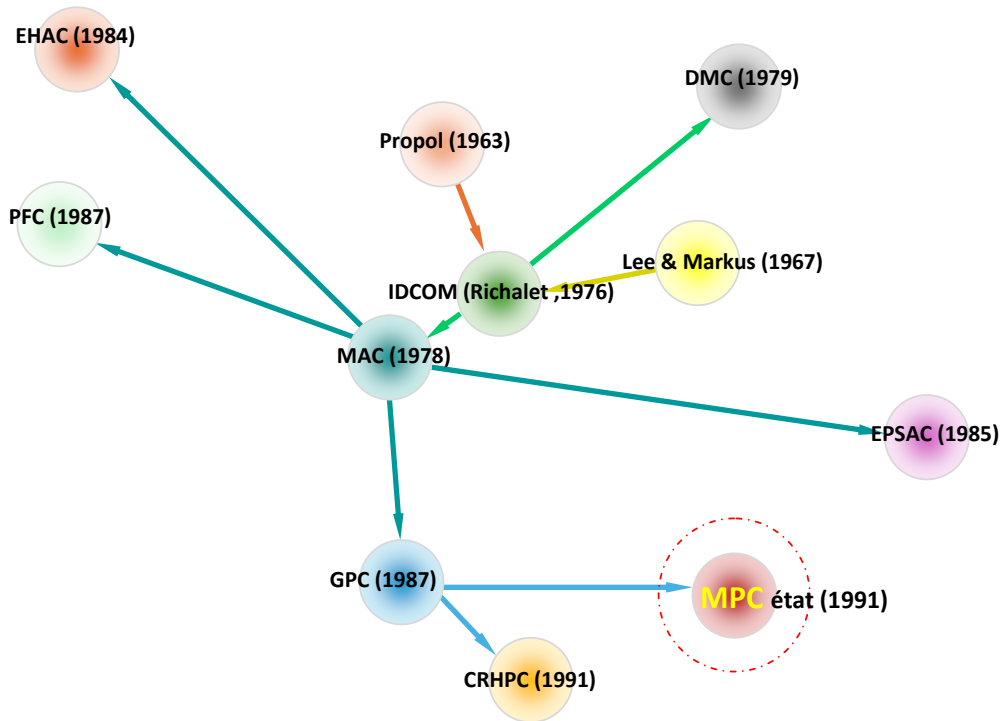


Figure I.5. Généalogie des algorithmes de la MPC linéaire.

- **Commande prédictive non linéaire: vers plus de robustesse**

La **CPN** étend le concept à des systèmes dynamiques non linéaires et potentiellement incertains. Elle conserve l'idée de prédiction à horizon glissant mais repose sur une **résolution temps réel** d'un problème de contrôle optimal non linéaire. Les approches suivantes ont été proposées pour renforcer la robustesse de la CPN:

Tableau I.2. Commande prédictive non linéaire: évolution et variantes.

Type	Approche	Caractéristiques	Références
Min-max	CPN robuste	Résolution min-max en boucle ouverte	Lall et al. [34], H. Chen et al. [35]
Multi-objectif	Optimisation robuste	Critères multiples pour incertitudes	J. Darlington et al. [36], B. Rustem [37]
Retour d'état	Interpolation de commande	Optimisation entre deux pas	M. Kothare et al., [38] L. Magni et al. [39]

Voici une représentation visuelle de la Figure I.6 qui représente la généalogie des approches de la commande prédictive non linéaire (CPN). Elle met en évidence les principales branches de développement de la CPN à partir de la méthode générale, avec les techniques **min-max**, **H^∞** , **multi-objective**, et **retour d'état intermittent**.

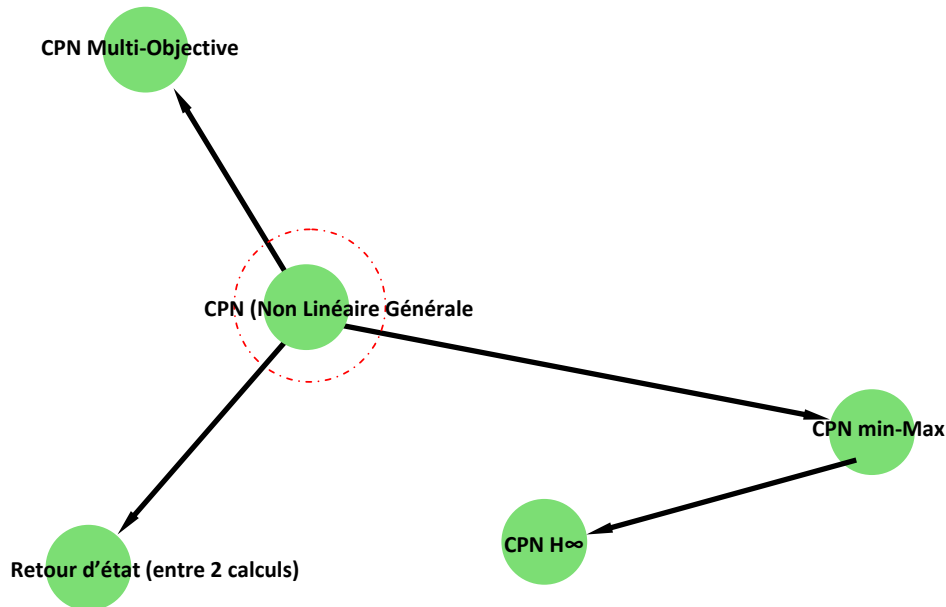


Figure I.6. Généalogie des approches de la commande prédictive non linéaire (CPN).

I.2.2.2. Éléments d'une commande prédictive

Tous les algorithmes de la commande prédictive reposent sur les mêmes composants fondamentaux. Selon les choix effectués pour chacun de ces éléments, on obtient une grande diversité de stratégies et d'algorithmes. Ces éléments sont: le modèle du système, le critère de performance, et l'algorithme d'optimisation.

A) Modèle du système

Le modèle constitue la pierre angulaire de toute approche prédictive. Il n'est pas uniquement destiné à la conception d'une loi de commande, mais surtout à la prédiction du comportement futur du procédé sous l'effet de sollicitations connues. Dans la pratique, il est embarqué dans le calculateur et qualifié de *modèle interne*. Selon les caractéristiques du processus et les spécifications de commande, il peut prendre différentes formes: équations aux différences, réponse impulsionnelle, réponse indicielle, ou encore modèles d'état discrets. D'autres représentations plus complexes existent, comme les modèles flous, neuronaux ou de Volterra [22].

B) Prédiction

L'étape de prédiction consiste à estimer les sorties futures sur un horizon de prédiction N_p , en fonction des mesures passées et des commandes futures envisagées. Ces prédictions sont ensuite utilisées pour construire la séquence optimale de commandes en résolvant un problème d'optimisation [25].

C) Critère de performance (fonction objectif)

Le calcul de la commande repose sur la minimisation d'un critère de performance, souvent exprimé sous forme quadratique, car il est différentiable et simple à manipuler. Ce critère mesure l'écart entre la sortie prédite et la trajectoire de référence, tout en pénalisant l'effort de commande:

$$J = \sum_{j=1}^{N_p} (\hat{y}(k+j) - y_{ref}(k+j))^T Q (\hat{y}(k+j) - y_{ref}(k+j)) + \sum_{j=1}^{N_c-1} \Delta u(k+j)^T R \Delta u(k+j) \quad (1.1)$$

Où Q et R sont des matrices de pondération positives [40]. Avec $\Delta u(k+j) = u(k+j) - u(k+j-1)$, représentant l'incrément de la commande et où $\hat{y}(k+j)$ représente la sortie prédite du système et $y_{ref}(k+j)$ la trajectoire de référence.

Le choix des paramètres horizon de prédiction N_p , horizon de commande N_c , période d'échantillonnage T_e influence directement la performance du correcteur. Une mauvaise sélection peut entraîner soit une commande trop agressive, soit une dynamique lente.

D) Contraintes

Les contraintes traduisent les limitations physiques du système (saturation des actionneurs, marges de sécurité, limitations économiques, etc.). Elles peuvent concerner:

- l'incrément de commande: $\Delta u_{min} \leq \Delta u \leq \Delta u_{max}$
- la commande elle-même: $u_{min} \leq u \leq u_{max}$
- la sortie du système: $y_{min} \leq y \leq y_{max}$
- l'état du système: $x_{min} \leq x \leq x_{max}$

Historiquement, la prise en compte explicite des contraintes a d'abord été contournée par des méthodes dites *anti-windup*, appliquées a posteriori. Depuis les années 1990, des approches plus rigoureuses intègrent directement les contraintes dans la synthèse de la commande, notamment grâce à la théorie des ensembles invariants et aux fonctions de Lyapunov non quadratiques [41-43].

E) Loi de commande

La séquence de commande optimale est obtenue en minimisant le critère sous contraintes. Dans le cas linéaire avec critère quadratique, la solution peut être calculée analytiquement (ex. algorithme Dynamic Matrix Control – DMC). Dans les autres cas, il est nécessaire d'utiliser des méthodes d'optimisation numérique, souvent itératives.

Cependant, pour garantir l'applicabilité en temps réel, ces méthodes doivent converger avant la fin de chaque période d'échantillonnage. C'est pourquoi le choix de l'algorithme d'optimisation revêt une importance capitale dans la stratégie de commande prédictive.

I.2.2.3. Avantages et inconvénients de la commande prédictive

La commande prédictive (MPC) constitue l'une des stratégies de commande les plus avancées et a démontré sa supériorité par rapport aux méthodes classiques, en particulier lorsqu'il s'agit de systèmes complexes ou présentant des comportements atypiques tels que: de grands retards, des oscillations marquées, une instabilité ou une dynamique à phase non minimale [24]. Voici un tableau récapitulatif des avantages et inconvénients de la commande prédictive (MPC):

Tableau I.3. Avantages et Inconvénients de la commande prédictive (MPC).

Catégorie	Aspect	Description
AVANTAGES	Facilité de mise en œuvre	Lois de commande simples à programmer, bonnes performances, réglage accessible même avec des connaissances limitées en automatique
	Prise en compte des contraintes	Intégration systématique des contraintes physiques (saturation des actionneurs, marges de sécurité, limites économiques)
	Caractère prédictif et anticipatif	Amélioration du suivi de trajectoire grâce à l'anticipation des consignes futures connues
	Adaptation aux perturbations	Compensation intrinsèque des retards et compensation anticipative (feedforward) des perturbations mesurables
	Méthodologie flexible	Approche ouverte, extensible aux cas non linéaires, utilisation explicite du modèle du procédé.
	Gain économique	Meilleure utilisation des actionneurs réduisant l'effort de commande et prolongeant leur durée de vie .
	Adaptation aux systèmes complexes	Particulièrement adaptée au contrôle multivariable et problèmes de suivi de trajectoire avec/sans contraintes.
	Stabilité numérique	Formulation mathématique assurant une stabilité fiable dans un large éventail d'applications.
INCONVÉNIENTS	Charge de calcul élevée	Résolution des algorithmes numériques en ligne nécessitant des ressources importantes et un temps de traitement long .
	Dépendance au modèle	Performance dépendant fortement de l'écart entre le modèle utilisé et le système réel .

Remarque sur les progrès technologiques récents

Il convient de souligner que les progrès technologiques récents ont fortement favorisé le développement du MPC [1], grâce notamment à:

- *la diversité et la puissance des techniques de modélisation et d'identification (même dans des environnements bruités),*
- *l'intégration des approches de commande robuste permettant de prendre en compte des incertitudes,*
- *la rapidité croissante des calculateurs numériques, rendant possible l'optimisation en ligne avec contraintes.*

I.2.2. Fondements de la logique floue et des systèmes d'inférence

La logique floue (LF) est une forme de logique à plusieurs valeurs qui traite du raisonnement approximatif plutôt que fixe et exact et qui permet de considérer toutes les connaissances qualitatives des opérateurs dans l'automatisation des systèmes. Elle a initialement émergé comme une technologie opérationnelle utilisée pour contrôler les processus industriels [44]. Contrairement à la logique classique, qui exige que les énoncés soient soit vrais ou faux, la logique floue permet des degrés de vérité, représentés par des valeurs entre 0 & 1. Cette caractéristique est très utile pour traiter des concepts vagues ou imprécis.

I.2.2.1. Théorie des sous ensembles flous

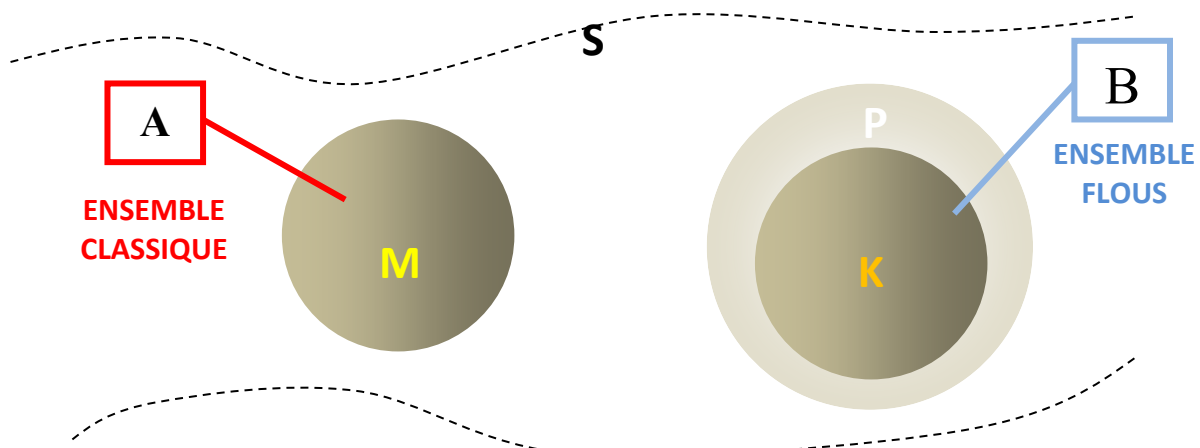
Le terme théorie des ensembles flous est apparu pour la première fois en 1965, lorsque le professeur Lotfi A. Zadeh, de l'université de Berkeley, aux États-Unis, a publié un article intitulé "Fuzzy Sets" [45]. Comme on le sait bien, dans la théorie des ensembles classique, un élément appartient soit à un ensemble, soit il n'y appartient pas. En revanche, les ensembles flous permettent une appartenance partielle, définie par une fonction d'appartenance qui attribue un degré d'appartenance à chaque élément de l'univers du discours (Zadeh, 1965 [45]; Hong & Lee, 1996 [46]).

L'un des principaux défis dans le développement de systèmes flous est l'acquisition de connaissances, telles que les fonctions d'appartenance (FA) et les règles floues. Traditionnellement, ceux-ci étaient définis par des experts humains, ce qui pouvait être chronophage et subjectif. Le processus a été automatisé grâce à l'apprentissage automatique. Quelques exemples d'algorithmes capables d'apprendre des règles et des fonctions d'appartenance à partir de données sont les réseaux neuronaux flous (Lin & Lee, 1991 [47]), les arbres de décision flous (Yuan & Shaw, 1995 [48]) et les algorithmes

génétiques (Yuan & Zhuang, 1996 [49]). Autre principe fondamental dans la logique floue est la théorie de la possibilité, que Zadeh en 1978 a introduite comme une contrepartie à la théorie des probabilités. La probabilité traite de l'ambiguïté quant à la survenue d'un événement, tandis que la possibilité utilise des distributions de probabilité d'ensembles flous pour décrire le doute sur les faits. La logique floue peut également diriger les haies linguistiques (par exemple, "très", "un peu") et les qualificateurs (par exemple, "probable", "possible"), ce qui la rend adaptée à la modélisation du langage humain et du raisonnement.

A) Le principe d'appartenance partielle

Au sein de la théorie des ensembles classique, l'appartenance d'un élément est une notion binaire: il appartient entièrement à un ensemble, ou n'y est pas. Si ce concept fondamental a servi de pierre angulaire à de nombreuses théories mathématiques, il se révèle inadéquat pour modéliser une multitude de situations courantes où l'appartenance est graduelle. Pour pallier cette limite, la théorie des ensembles flous a été développée. Celle-ci introduit le principe révolutionnaire d'appartenance partielle, selon lequel un élément peut appartenir à un degré variable, compris de 0 à 1, à un ensemble donné. Cette nuance se traduit visuellement par des contours indéfinis, comme l'illustre la Figure I.7, qui sont « flous » ou « graduels » et non plus nets [50].



- À gauche: contour net d'un ensemble classique. À droite: contour flou ou graduel d'un ensemble flou.
- M: Appartient totalement à « A » (appartenance = 1).
- K: Appartient totalement à « B » (appartenance = 1).
- P: Appartient partiellement à « B » (appartenance $\in]0, 1[$).
- S: N'appartient ni à « A » ni à « B » (appartenance = 0).

Figure I.7. Illustration des concepts d'appartenance dans les ensembles classiques et flous.

Utilisez la notation $\mu_A(x) = D$ (où μ est la fonction d'appartenance, A est l'ensemble flou, x est l'élément à l'ensemble A et D est le degré entre 0 et 1) pour gagner en précision et en professionnalisme. Cette représentation schématique met en contraste les logiques d'appartenance propres aux ensembles classiques et flous. Pour les ensembles classiques A et B , l'appartenance est une notion binaire, où l'élément M appartient totalement à A , avec une fonction d'appartenance $\mu_A(M) = 1$, tandis que l'élément K appartient totalement à B , noté $\mu_B(K) = 1$; à l'inverse, l'élément S n'appartient à aucun des deux ensembles, ce qui se traduit par $\mu_A(S) = \mu_B(S) = 0$. En revanche, la logique des ensembles flous introduit le principe d'appartenance graduelle, illustrée ici par l'élément P qui appartient partiellement à l'ensemble flou B , avec un degré d'appartenance compris strictement entre zéro et un, soit $0 < \mu_B(P) < 1$.

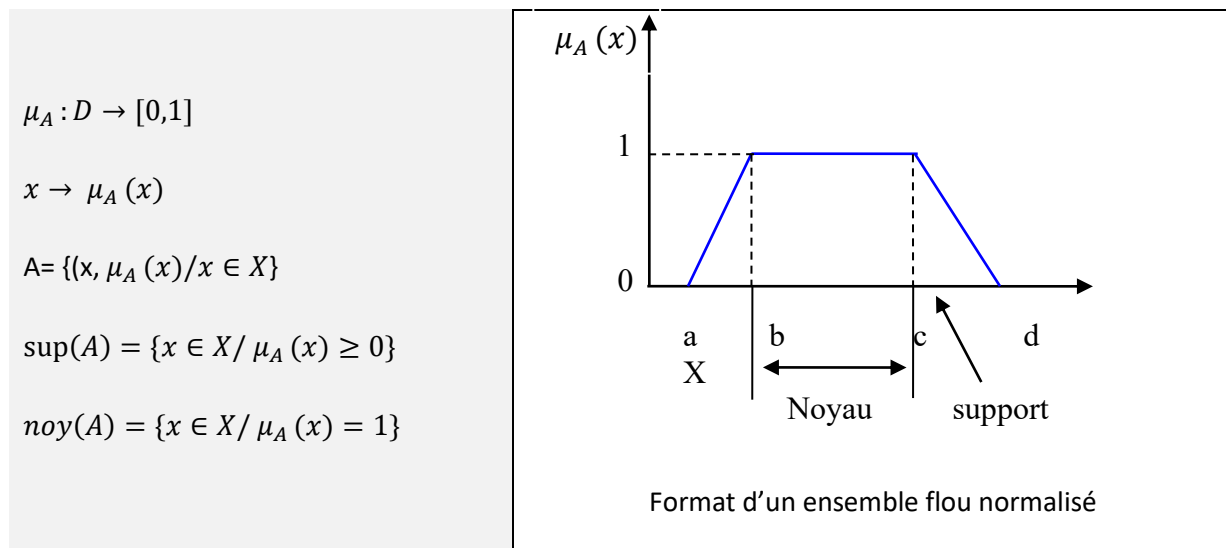


Figure I.8. Structure d'un sous-ensemble flou normalisé.

Les fonctions d'appartenance floues définissent la manière dont chaque entrée correspond à un degré d'appartenance compris de 0 à 1. Il existe plusieurs types de fonctions d'appartenance avec des formes telles que des courbes triangulaires, trapézoïdales, sigmoïdes et gaussiennes, chacune ayant une forme spécifique comme elle est présentée dans le tableau suivant.

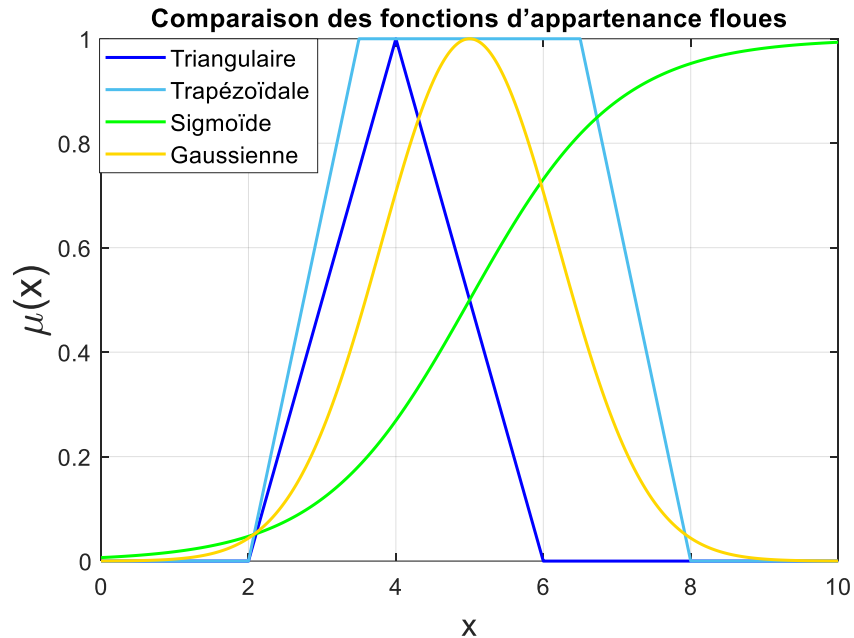


Figure I.9. Comparaison des fonctions d'appartenance floues.

Tableau I.4. Récapitulatif des fonctions d'appartenance floues: formules et applications typiques.

Fonction	Formule principale	Paramètres	Applications typiques
Triangulaire	$\mu_{\text{Triangulaire}}(x) = \max\left(\min\left(\frac{x-a}{b-a}, \frac{c-x}{c-b}\right), 0\right)$	a,b,c: bornes et sommet	Modèles simples, notions comme <i>faible, moyen, élevé</i>
Trapézoïdale	$\mu(x) = \max\left(\min\left(\min\left(\frac{x-a}{b-a}, 1\right), \frac{d-x}{d-c}\right), 0\right)$	a,b,c,d: coins du trapèze	Représente des intervalles « stables » (ex.: <i>zone acceptable</i>)
Sigmoïde	$\mu(x) = \frac{1}{1 + e^{-a(x-c)}}$	a: pente, c: centre	Transitions progressives (<i>petit → grand, froid → chaud</i>)
Gaussienne	$\mu(x) = \exp\left(-\frac{(x-c)^2}{2\sigma^2}\right)$	c: centre, σ : écart-type	Données naturelles, proximité autour d'une valeur centrale

Les intervalles flous [51] déterminent le nombre de variables floues associées à une grandeur d'entrée. Dans le cadre du réglage, trois à cinq intervalles sont généralement suffisants. Ces intervalles sont représentés à l'aide de symboles linguistiques, tels que Positif Petit (PP), Positif Moyen (PM) et Positif Grand (PG), Négatif Grand (NG), Négatif Moyen (NM), Négatif Petit (NP), Zéro (Z), entre autres. La logique floue repose sur des variables dites linguistiques, qui prennent ces valeurs symboliques comme

modalités dans l'univers du discours X (ou U). Ainsi, chaque valeur linguistique correspond à un ensemble flou au sein de cet univers.

B) Opérations fondamentales sur les ensembles flous [52]

Les principales opérations sur les ensembles flous permettent de modéliser des phénomènes où la frontière entre les éléments d'un ensemble n'est pas nette. Comme c'est le cas dans de nombreux domaines des sciences, notamment en intelligence artificielle et en traitement du signal.

Les principales opérations sur les ensembles flous sont semblables à celles sur les ensembles classiques, mais elles sont adaptées à la logique floue. Ces opérations sont:

- **Egalité:** Deux sous-ensembles flous A et B d'un ensemble U sont égaux si leurs fonctions d'appartenance sont identiques pour chaque élément x de U . En d'autres termes, A et B ont la même distribution de degrés d'appartenance pour tous les éléments de U . En termes mathématiques:

$$\mu_A(x) = \mu_B(x) \quad \forall x \in U \quad (1.2)$$

Cette propriété est importante en raisonnement flou car elle établit une égalité entre des ensembles à appartenance graduelle.

- **Inclusion:** Un sous-ensemble flou A est inclus dans B (noté $A \subseteq B$) si, pour chaque élément x de U appartenant à A , le degré d'appartenance de x à B est au moins aussi grand que son degré d'appartenance à A . Autrement dit, chaque élément de A appartient à B avec un degré d'appartenance qui ne diminue pas. En termes mathématiques:

$$\mu_A(x) \leq \mu_B(x) \quad \forall x \in U \quad (1.3)$$

Cette opération est analogue à l'inclusion dans les ensembles classiques, mais tient compte des degrés d'appartenance au lieu de simplement vérifier l'existence d'un élément.

- **Intersection:** C'est le minimum des degrés d'appartenance des sous-ensembles flous A et B . Cela signifie que, pour chaque x , le degré d'appartenance à $A \cap B$ est le plus faible des degrés d'appartenance à A et B . Cette opération est illustrée par (Figure I.10 a):

$$\forall x \in U, \mu_{A \cap B}(x) = \mu_A(x) \wedge \mu_B(x) = \min(\mu_A(x), \mu_B(x)) \quad (1.4)$$

Cette opération est logique dans un contexte flou car l'appartenance d'un élément à deux ensembles dépend de son degré d'appartenance à chacun.

- Union:** L'union floue de A et B est le sous-ensemble constitué des maximums des degrés d'appartenance des deux ensembles. Pour tout x, le degré d'appartenance à AUB est le plus élevé entre son degré d'appartenance à A et à B. Cela se formalise par (Figure I.10 b):

$$\forall x \in U, \mu_{A \cup B}(x) = \mu_A(x) \vee \mu_B(x) = \max(\mu_A(x), \mu_B(x)) \quad (1.5)$$

Si un élément est dans un ensemble, son degré d'appartenance à l'union sera au moins aussi élevé.

- Complément:** Le complément d'un sous-ensemble flou A est défini par le degré d'appartenance opposé de chaque élément à A. En d'autres termes, si x est dans A à un degré $\mu_A(x)$, alors il est à $1 - \mu_A(x)$ dans le complément de A. Cela signifie (Figure I.10 c):

$$\forall x \in U, \mu_{\bar{A}}(x) = 1 - \mu_A(x) \quad (1.6)$$

Cette opération inverse la fonction d'appartenance d'un ensemble flou, utile en logique floue pour exprimer des oppositions ou des négations.

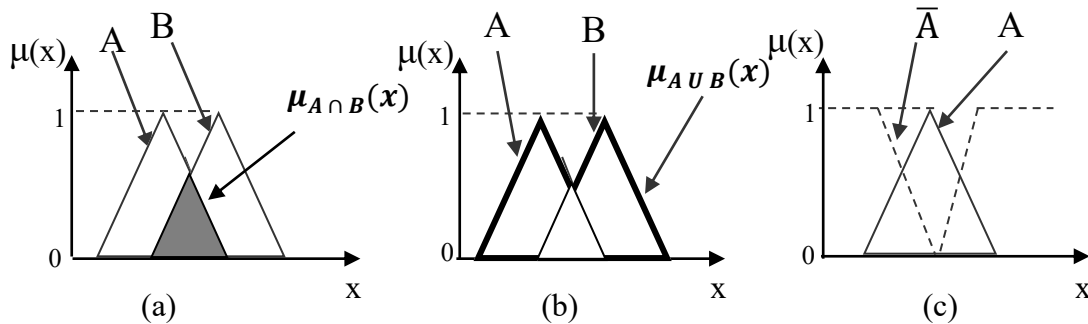


Figure I.10. Représentation des opérations union, intersection et complément flous.

Remarque: Ces opérations sont essentielles en logique floue pour traiter des ensembles à valeurs continues. Ces opérateurs sont souvent utilisés dans différents systèmes flous tels que les systèmes de contrôle, les bases de données et les systèmes de raisonnement automatisé. Cela est parfaitement illustré par l'architecture d'un système expert flou, qui est conçu pour exécuter ces opérations mêmes (Voir la Figure I.11) [53].

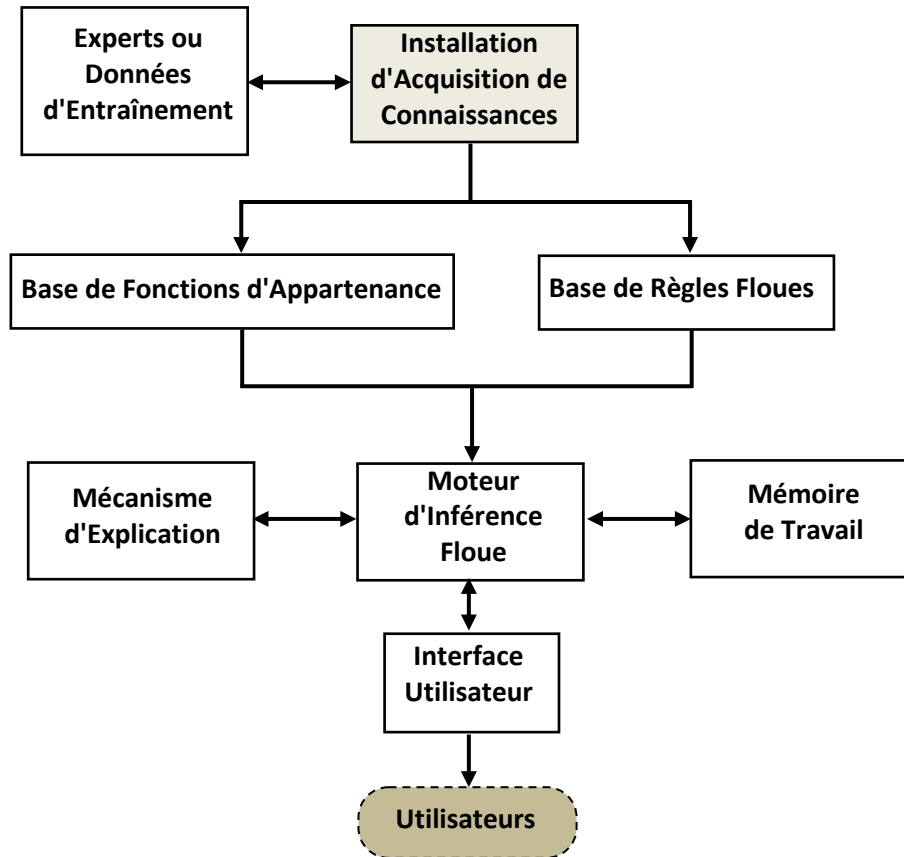


Figure I.11. Architecture d'un système expert flou.

Cette architecture commence par l'Interface Utilisateur, qui permet aux utilisateurs d'entrer des données nettes et d'obtenir les résultats du système. La Mémoire de Travail, un tableau noir de session, stocke ces données. Le moteur d'inférence flou, le cerveau du système, effectue la fuzzification, l'inférence et la défuzzification. Le moteur utilise deux bases de connaissances pour effectuer ces tâches: la Base de Fonctions d'Appartenance, qui spécifie les formes des ensembles flous comme "Chaud" ou "Lent," et la Base de Règles Floues, qui contient des règles SI-ALORS fournies par des experts et qui reflètent la logique de prise de décision. Les modules d'apprentissage comme la Knowledge Acquisition Facility automatisent la mise à jour des règles et des fonctions d'appartenance à partir d'experts humains ou de données d'entraînement. Enfin, le Mécanisme d'Explication explique quelles règles ont été déclenchées et comment elles ont été examinées pour garantir la transparence et la confiance des utilisateurs.

C) Règles floues: Système de règles floues

Les règles floues sont cruciales pour modéliser les systèmes complexes et incertains. Ces règles lient des prémisses floues à des conclusions malgré des données incertaines. Les règles floues sont utilisées dans

l'intelligence artificielle, le contrôle des systèmes et la prise de décision pour traiter des informations imprécises ou approximatives. L'implication dans les systèmes de raisonnement flou relie les prémisses aux conclusions en tenant compte de l'incertitude et de la gradation des valeurs. En logique classique, $A \Rightarrow B$ signifie que si A est vrai, alors B doit être vrai. Cependant, cette relation vrai/faux est trop rigide pour les systèmes réels avec incertitude.

En systèmes flous, l'implication est flexible, reliant des prémisses et des conclusions graduées. Les règles floues sont exprimées par des implications logiques, souvent de type "SI-ALORS". Une règle floue R se définit ainsi:

$$\forall x \in U, \text{ Si } x \text{ est } A \text{ Alors } y \text{ est } B \tag{1.7}$$

Où x représente la variable d'entrée et y la variable de sortie, A et B étant des ensembles flous qui représentent les catégories d'appartenance des variables. Ces règles peuvent aussi être plus complexes, intégrant plusieurs conditions. Dans ce cas, la règle floue peut être formulée comme suit:

$$\forall x \in U, \text{ R: Si } x_1 \text{ est } A_1 \text{ et } x_2 \text{ est } A_2, \dots \text{ et } x_n \text{ est } A_n \text{ Alors } y \text{ est } B \tag{1.8}$$

Considérons la règle floue suivante: "Si la distance au feu est courte et la vitesse de la voiture est maximale, alors il faut fortement freiner." [54] Ce type de règle est un exemple de **prédicat** où les relations entre les prémisses et la conclusion sont exprimées à travers des opérateurs **ET, OU, et NON**. La flexibilité des règles floues, avec la manipulation des degrés d'appartenance et l'utilisation des opérateurs logiques "**ET, OU, et NON**", permet de traiter des cas où la logique binaire classique (LBC) serait insuffisante. Ces règles sont un outil puissant pour l'intelligence artificielle et les systèmes d'aide à la décision.

1.2.2.2. Concept et formulation des variables linguistiques

En logique floue, les concepts des systèmes sont généralement représentés par des *variables linguistiques*. Une variable linguistique est une variable dont les valeurs sont des mots ou des expressions, appelées termes linguistiques, qui sont utilisées dans le langage naturel ou dans un langage artificiel pour décrire des phénomènes de manière approximative. Contrairement aux variables classiques qui prennent des valeurs numériques précises, les variables linguistiques prennent des valeurs qui sont souvent subjectives et dépendent de l'interprétation du langage (Zadeh, 1975). Selon C.T. Lin et Lee (1996), une VL est formellement définie comme suit [55]:

$$(X, U, T(X), \mu(x)) \quad \forall x \in U \quad (1.9)$$

Où $T(X)$ représente les valeurs linguistiques de la variable X , aussi appelées termes ou étiquettes linguistiques, avec X comme nom de la variable, U représentant l'univers du discours lié à X , et enfin $\mu(x)$ désignant la fonction d'appartenance des termes linguistiques.

Exemple: La variable linguistique "vitesse" peut avoir des termes linguistiques tels que "faible", "moyenne", "élevée", etc., chacun étant associé à une fonction d'appartenance représentant l'intensité de la correspondance entre la valeur mesurée et ces catégories.

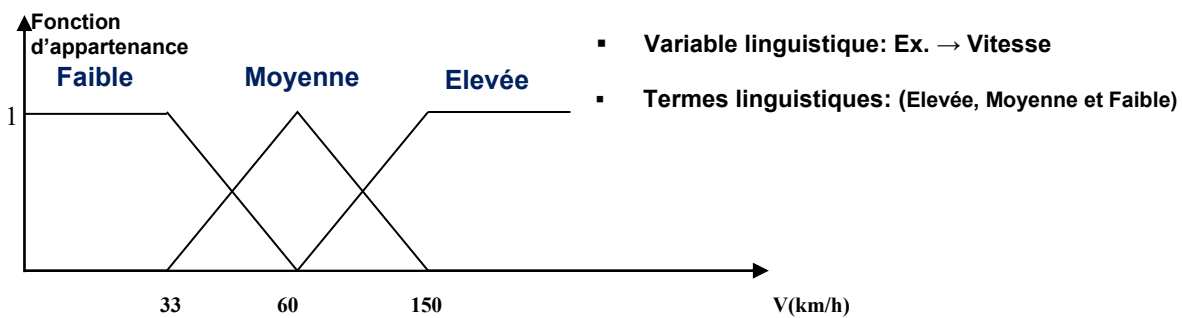


Figure I.12. Variable linguistique.

1.2.2.3. Structure et fonctionnement d'un système d'inférence flou (SIF)

Un système d'inférence flou (SIF) est un schéma de calcul populaire basé sur les concepts de la théorie des ensembles flous (x), des règles floues R_n et du raisonnement flou. Ce système vise à transformer des données d'entrée en données de sortie en fonction de l'évaluation d'un ensemble de règles. Selon leur contexte d'application, les SIF sont également appelés systèmes experts flous, modèles experts flous, contrôleurs flous ou encore systèmes flous [56]. Un SIF est constituée de trois étapes: la fuzzification, qui convertit les entrées précises en ensembles flous; le moteur d'inférence, qui évalue les règles floues à partir des entrées fuzzifiées; et la défuzzification, qui convertit l'ensemble flou de sortie en une valeur précise (voir Figure I.13).

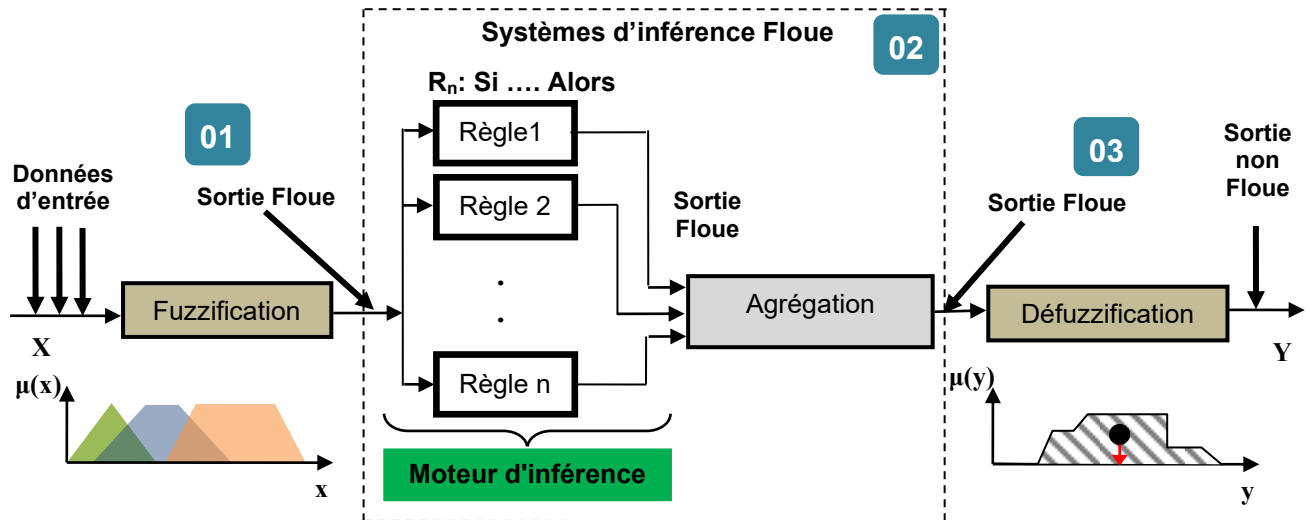


Figure I.13. Structure d'un système d'inférence flou.

- 1) **La fuzzification** est la première étape cruciale de tout SIF. Il souhaite relier les mesures précises et numériques à la logique floue. Pour cette transition, définissons un cadre conceptuel pour chaque variable d'entrée. Cet univers de discours représente toutes les valeurs possibles d'une variable (par exemple, une température de 0°C à 100°C). On divise ensuite l'univers en catégories sémantiques, appelées variables linguistiques (comme "froid", "tiède", "chaud" pour la température, ou "très négatif", "négatif", "moyen" pour un écart de mesure). Chaque catégorie est définie par une fonction d'appartenance floue. La fuzzification transforme une valeur d'entrée précise (comme 20°C) en information floue. Cette transformation s'effectue en évaluant le degré d'appartenance de la valeur d'entrée à chaque sous-ensemble flou défini. Ce degré, de 0 à 1, mesure l'appartenance d'une valeur numérique à une catégorie linguistique. À 20°C , degré d'appartenance: "chaud" 0,7, "tiède" 0,3. À la fin de cette étape, une seule entrée numérique est transformée en plusieurs paires (ensemble flou, degré d'appartenance), en vue de l'étape suivante: Le moteur l'inférence.
- 2) Le moteur d'inférence est le cœur du SIF où il représente un mécanisme de condensation des informations système grâce à un ensemble de règles spécifiques pour représenter un problème spécifique. Il évalue les règles floues, généralement sous la forme "SI-ALORS", grâce à les entrées fuzzifiées. Chaque règle est activée à un certain degré en fonction du degré d'appartenance des entrées aux ensembles flous correspondants. Les conséquences de ces règles sont ensuite combinées pour former un ensemble flou de sorties. Par exemple, si la règle "Temp. chaude \rightarrow

ventilateur rapide" est activée, la sortie "ventilateur rapide" est générée avec un degré spécifique. Les règles constituent la partie essentielle d'un système d'inférence flou.

- 3) Enfin, la défuzzification est l'étape finale du processus SIF. Ce processus est l'opposé de la fuzzification. Le flou de sortie évalué par le moteur d'inférence devient une valeur précise et utilisable. Plusieurs méthodes de défuzzification peuvent être employées, telles que la méthode du centroïde, qui calcule le centre de gravité de la fonction d'appartenance de sortie. Par exemple, "ventilateur rapide" devient une vitesse spécifique pour le ventilateur [57].

• **Mécanisme d'inférence floue:**

L'inférence floue de type Mamdani est basée sur une base de règles linguistiques avec des fonctions d'appartenance pour représenter des concepts. Le mécanisme d'inférence comprend les étapes suivantes:

- ✓ La première étape de ce mécanisme est **la fuzzification**, qui consiste à évaluer les fonctions d'appartenance des prédicats des règles. Il convertit une valeur d'entrée nette et numérique en une valeur floue en déterminant son degré d'appartenance à un ensemble linguistique. Il souhaite transformer des entrées précises et numériques (par ex., 19°C, 3,2 bars) en valeurs floues linguistiques (par ex., "température élevée" retourne un degré de 0,2 et "pression forte" renvoie un degré de 0,6, alors: ouverture vanne grande) en calculant leur degré d'appartenance à différents ensembles flous. C'est le pont entre le monde numérique précis des capteurs et le monde linguistique et approximatif du raisonnement humain, qui est capturé dans la base de règles floues.

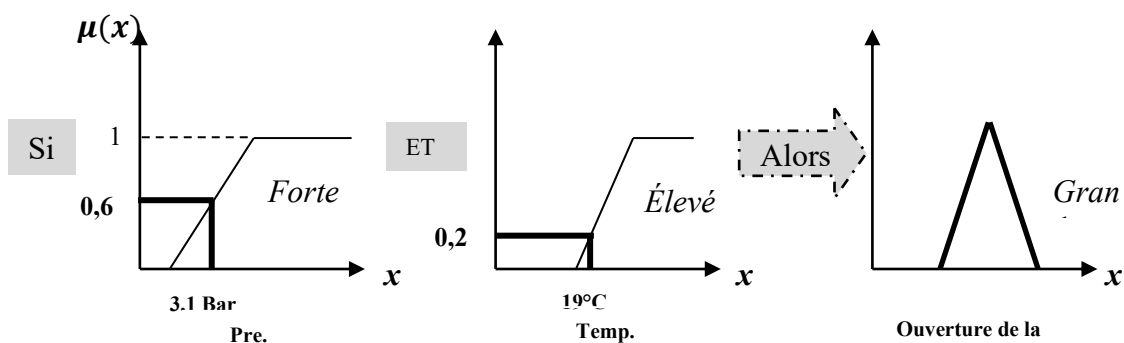


Figure I.14. Le Processus de fuzzification dans un SIF de type Mamdani.

- ✓ Une fois que toutes les entrées numériques ont été converties en degrés d'appartenance à leurs ensembles flous $u(x)$ respectifs, l'étape suivante consiste à évaluer les règles de la base de connaissances. C'est ce qu'on appelle l'activation. L'activation d'une règle combine les vérités

floues de ses conditions (souvent par un opérateur MIN) pour déterminer dans quelle mesure sa conclusion doit s'appliquer. La Figure I.15 montre le calcul pour une règle spécifique, probablement: "SI Pression Forte ET Température Élevée ALORS Ouverture Grande". L'opérateur ET (Combinaison logique): L'antécédent de la règle utilise l'opérateur ET. Dans la logique floue de Mamdani, la méthode la plus courante pour calculer le "ET" est de prendre le minimum des degrés d'appartenance. Ici, $\min(0,6, 0,2) = 0,2$. À droite: Application à la Conclusion (ALORS): Le degré d'activation calculé (0,2) est appliqué à la conclusion de la règle ("Ouverture Grande"), c'est l'implication. C'est-à-dire que la conclusion (Ouverture vanne grande) est vraie à 20%. Graphiquement, cela revient à couper la fonction d'appartenance de la sortie $u(y)$ (Grande) à la hauteur du degré d'activation (0,2).

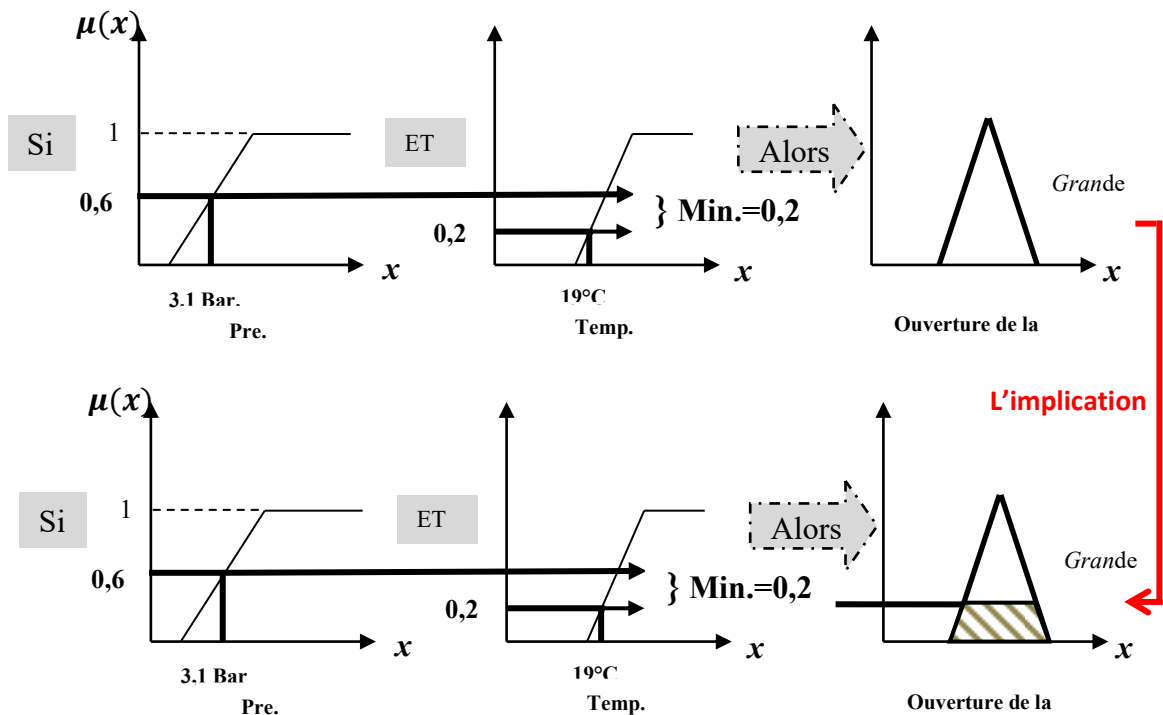


Figure I.15. L'Activation des Règles dans un SIF.

- ✓ Après l'étape d'activation, chaque règle a produit sa propre conclusion tronquée (un ensemble flou de sortie "coupé" à la hauteur de son degré d'activation). **L'agrégation** est le processus qui combine toutes ces conclusions partielles en un seul et unique ensemble flou composite qui représente la décision floue globale du système. L'agrégation fusionne les conclusions partielles de toutes les règles activées (généralement avec un opérateur MAX) pour former une seule sortie floue globale qui synthétise toutes les recommandations du système. la figure suivent

montre un cas simple où deux règles agissent sur la même variable de sortie: "Ouverture de vanne". Pour chaque valeur possible d'ouverture de vanne sur l'axe horizontal (x), le système prend la valeur d'appartenance la plus élevée (le maximum) entre les deux courbes. La ligne pointillée et la zone grise résultante forment le nouvel ensemble flou agrégé, qui est une fusion des recommandations des deux règles.

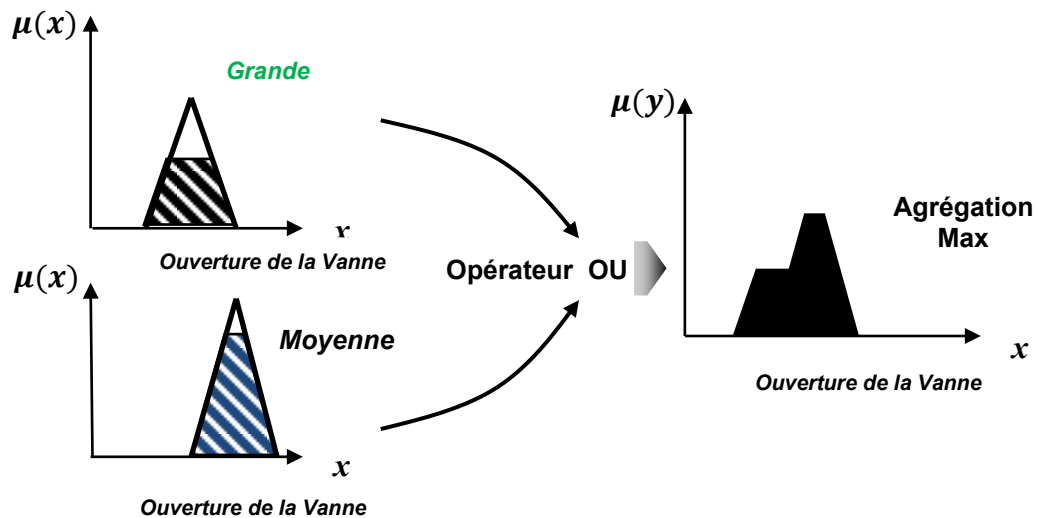


Figure I.16. L'Agrégation des sorties floues dans un SIF.

Cette sortie agrégée porte l'information de tout le système expert pour être convertie en une valeur de sortie nette et exploitable dans l'étape finale: **la défuzzification**. Il existe deux types, soit:

1. **Méthode du centre de gravité (COG - Center Of Gravity):** Cette méthode calcule l'abscisse du **centre de gravité** (le centre de la masse) de la surface sous la courbe de l'ensemble flou agrégé. La formule est donnée:

$$sortie = \frac{\int_U y \cdot \mu(y) \cdot dy}{\int_U \mu(y) \cdot dy} \quad (I.10)$$

Avec U c'est l'univers du discours.

2. **Méthode de la moyenne des maximums (MM - Mean of Maximum):** Cette méthode identifie d'abord la ou les valeurs de sortie (y) qui ont le degré d'appartenance le plus élevé (le "pic" de la courbe agrégée). Ensuite, elle calcule simplement la moyenne de ces valeurs. La formule est donnée:

$$sortie = \frac{\int_s y \cdot dy}{\int_s dy} \quad (I.11)$$

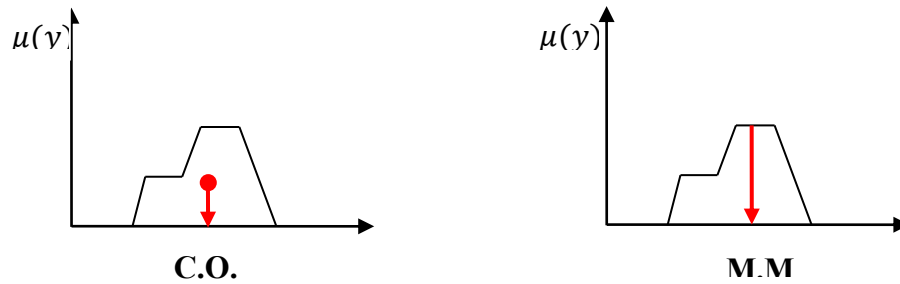


Figure I.17. Conversion de la Sortie Floue en Valeur Nette (COG vs MM).

1.2.2.4. Comparaison des modèles d'inférence floue de Mamdani et de Takagi-Sugeno-Kang (TSK)

Il existe divers types de systèmes d'inférence floue en fonction des opérateurs et de la méthode d'agrégation des règles. La méthode d'inférence floue de Mamdani est l'un des premiers systèmes de contrôle basés sur la logique floue [58]. Ses travaux reposent sur ceux de Lotfi A. Zadeh sur les algorithmes flous pour les systèmes complexes et les processus de décision. Le deuxième modèle, le TSK, est très efficace en calcul. Il est très attrayant dans les problèmes de contrôle, notamment dans les systèmes dynamiques non linéaires, grâce à ses techniques d'optimisation et d'adaptation [59]. Dans le tableau ci-dessous, nous détaillons les deux classes des modèles flous, et leurs propriétés principales :

Tableau I.5. Les différents modèles flous et leurs propriétés essentielles.

Aspect	Le modèle de Mamdani	Le modèle de Takagi-Sugeno-Kang (TSK)
Forme des règles	$R^{(1)}$ SI x_1 est A_1^1 et x_2 est A_2^1 et ... x_n est A_n^1 ALORS y est B^1	$R^{(1)}$ SI x_1 est A_1^1 et x_2 est A_2^1 et ... x_n est A_n^1 ALORS y est $f(x_1, \dots, x_n)$
Calcul du degré d'activation (α)	$\alpha_i(x) = \min(\mu_i(x_i))$	$\alpha_i(x) = \prod_{i=1}^n \mu_i(x_i)$
Calcul de l'implication	$\mu_i(y_i) = \min(\alpha_i(x), \mu_i(x_i))$	$y_i = \alpha_i(x) \cdot f_i(x_1, \dots, x_n)$
Calcul de l'agrégation	$\mu_c = \max(\alpha_i(x), \mu_i(x_i))$	$y_c = \sum_i^n \mu_i(x_i) \cdot y_i$
Sortie finale	$y = \frac{\sum_i^n \mu_i(x_i) \cdot y_i}{\sum_j^n \mu_j(x)}$	$y = \frac{\sum_i^n y_i}{\sum_i^n \alpha_i(x)}$
Avantage	Très intuitif et facile à interpréter, car il se rapproche du raisonnement humain. Idéal pour des systèmes experts où la compréhension des règles est primordiale.	<ul style="list-style-type: none"> • Beaucoup plus efficace computationalement (pas de défuzzification coûteuse). • Se comporte comme un universal approximator (peut modéliser des systèmes non-linéaires complexes).

▪ **Légende:** α : Degré d'activation (degré de vérité) de la règle. $\mu_i(x_i)$: Degré d'appartenance de la variable d'entrée. y : Valeur de sortie finale du système. y_i : Sortie de la règle i. x : Valeur d'entrée de la règle. f : Fonction numérique (souvent linéaire) dans le conséquent

Le modèle de Mamdani est plus intuitif et adapté au raisonnement expert, tandis que le modèle TSK est plus efficace computationnellement et mieux adapté à la modélisation de systèmes complexes.

La logique floue de type 2 constitue une extension du cadre flou classique (type 1) introduit par Zadeh, en intégrant l'incertitude non seulement dans les variables linguistiques, mais également dans les fonctions d'appartenance elles-mêmes (Voir chapitre 2). Un **ensemble flou de type 2 (T2FS)** associe à chaque élément $x \in X$ un *degré d'appartenance flou* dans l'intervalle $[0,1]$, plutôt qu'une valeur précise. L'incertitude sur les paramètres tels que la moyenne ou l'écart type est ainsi représentée par une fonction d'appartenance secondaire $\mu_2(x, \mu_1)$. Formellement, un ensemble gaussien de type 2 s'écrit:

$$\mu_2(x, \mu_1) = \exp\left(-\frac{(\mu_1 - m(x))^2}{2\sigma_m^2}\right) \tag{I.12}$$

Où $\mu_1 \in [0,1]$ désigne les *appartenances primaires* et μ_2 leurs *appartenances secondaires*. La région englobant toutes les valeurs possibles d'appartenance est appelée empreinte d'incertitude (Footprint of Uncertainty, FOU), délimitée par la fonction d'appartenance supérieure et inférieure (UMF, LMF).

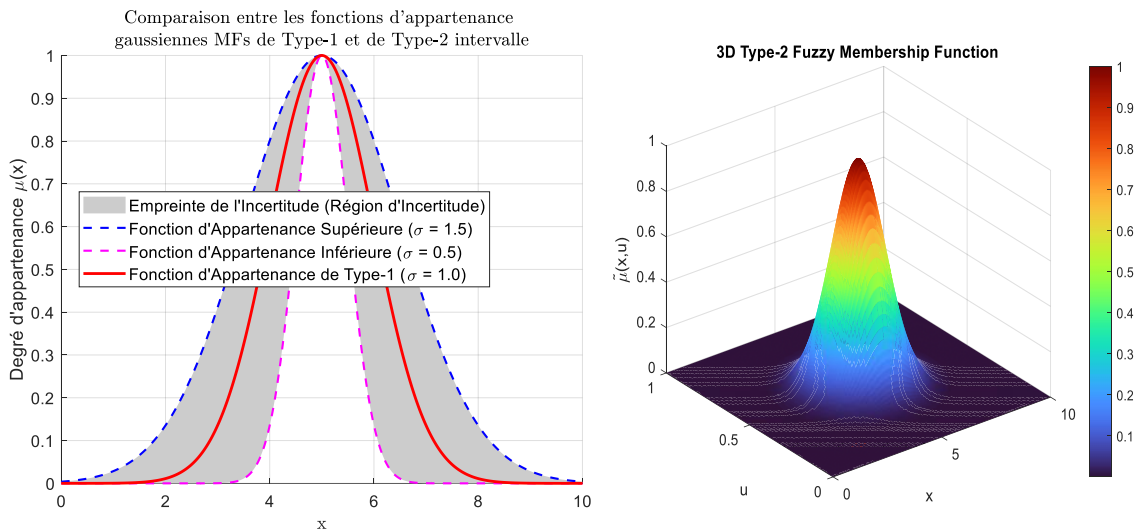


Figure I.18. Empreinte de l'incertitude (EI) entre les fonctions d'appartenance supérieure et inférieure.

Par exemple, une fonction gaussienne avec un écart type incertain $\sigma \in [\sigma_1, \sigma_2]$ définit un ensemble flou de type 2 par intervalles:

$$\begin{cases} \text{UMF}(x) = N(m, \sigma_2 : x) \\ \text{LMF}(x) = N(m, \sigma_1 : x) \end{cases} \tag{I.13}$$

La **FOU** traduit l'incertitude due au bruit, à l'imprécision des données ou au manque de connaissances expertes.

Les opérations sur les ensembles flous de type 2 (union, intersection, complément) reposent sur le **principe d'extension de Zadeh**. Pour deux ensembles \tilde{A}_1 et \tilde{A}_2 :

$$\tilde{A}_1 \cup \tilde{A}_2 = \int_x \mu_{\tilde{A}_1 \cup \tilde{A}_2}(x)/x \quad , \quad \tilde{A}_1 \cap \tilde{A}_2 = \int_x \mu_{\tilde{A}_1 \cap \tilde{A}_2}(x)/x \quad (I.14)$$

Où les opérations « join » et « meet » généralisent les compositions max–min aux fonctions d'appartenances secondaires.

Un système de logique floue de type 2 (T2FLS) conserve la même structure qu'un système de type 1 — fuzzification, inférence et traitement de sortie — mais remplace la défuzzification par une réduction de type, produisant un ensemble de type réduit (type 1).

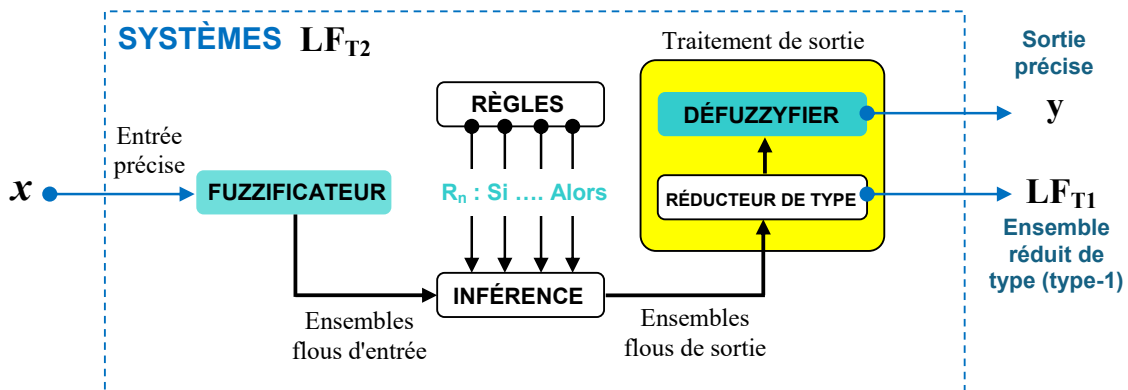


Figure I.19. Fonctionnement et architecture d'un système de logique floue de type 2 [24].

Cet ensemble représente le comportement collectif d'une multitude de systèmes flous de type 1 imbriqués. La sortie crisp $y(x)$ est obtenue par défuzzification.

$$y(x) = \frac{\sum_{k=1}^{\alpha} y_k \mu_Y(y_k)}{\sum_{k=1}^{\alpha} \mu_Y(y_k)} \quad (I.15)$$

Il existe des systèmes singleton et non-singleton selon que l'entrée est précise ou floue, ainsi que des systèmes Sugeno de type 2, dont les conséquents sont des fonctions plutôt que des ensembles. Lorsque les appartenances secondaires deviennent ponctuelles, le système se ramène à un système flou de type 1. Ainsi, la logique floue de type 2 offre une approche mathématiquement rigoureuse pour modéliser, propager et réduire l'incertitude, notamment dans les systèmes de contrôle et de décision affectés par le bruit ou la variabilité des données [60].

I.3. Taxonomie des travaux de recherche en contrôle prédictif flou

I.3.1. Classification selon le type du système flou utilisé

I.3.1.1. Commande prédictive floue de type 1 (CPF1)

Le contrôle prédictif flou de Type-1 (FMPC) utilise des ensembles flous avec des degrés d'appartenance précis pour représenter la dynamique du système, employant couramment les cadres Mamdani ou Takagi–Sugeno (T–S). Cette catégorie a été largement étudiée en raison de sa clarté conceptuelle, de sa complexité computationnelle réduite, et de sa capacité suffisante à encapsuler des non-linéarités modérées et des incertitudes. Les premières applications ont été observées dans le contrôle des procédés industriels, tels que la régulation de la température de fours et le traitement des eaux usées, où le Type-1 FMPC surpassait les régulateurs PID conventionnels en termes de rejet des perturbations et de précision de suivi [61-63]. Le modèle flou T–S s'est révélé particulièrement efficace pour les processus biologiques multivariables et non linéaires, permettant la dérivation de lois de contrôle prédictif sous forme d'espace d'état.

Le Type-1 FMPC ne s'est pas limité aux industries de procédés traditionnelles ; il a trouvé des applications précieuses dans plusieurs branches de l'ingénierie. Dans le domaine automobile, il a été utilisé pour ajuster le rapport air–carburant dans les moteurs à combustion [64] et améliorer la gestion énergétique des véhicules électriques hybrides rechargeables [65]. Il a également permis de coordonner les systèmes de stockage dans des réseaux multi-machines afin d'augmenter la stabilité de la tension, l'amortissement des oscillations et la régulation de la fréquence [66]. Le contrôle prédictif flou a amélioré la performance du freinage des trains en traitant le comportement non linéaire et complexe des systèmes de freinage [67].

Ce qui se démarque ces dernières années, c'est la progression du processus de conception lui-même. Des techniques d'optimisation itératives [68] ont été introduites, et les modèles flous ont été combinés avec des algorithmes heuristiques ou évolutionnaires pour augmenter la flexibilité et la robustesse. Dans l'ensemble, ces développements soulignent que le Type-1 FMPC reste très pertinent, continuant à servir d'outil important pour relever les défis des systèmes d'ingénierie modernes, à grande échelle et complexes. Le contrôle prédictif du modèle flou de Type-1 offre plusieurs avantages clés:

- Structure simple avec une charge computationnelle faible.
- Efficace pour les systèmes modérément non linéaires avec des incertitudes limitées.
- Faisabilité pratique pour une implémentation en temps réel.
- Intégration facile avec des algorithmes d'optimisation ou évolutionnaires.

Malgré sa capacité limitée à gérer des niveaux élevés d'incertitude, le Fuzzy MPC de Type-1 continue d'être largement adopté en raison de son équilibre entre simplicité et efficacité pratique. Pour pallier cette limitation, le MPC flou de Type-2 a été développé, offrant une flexibilité accrue dans la gestion de l'incertitude, comme discuté ci-après.

1.3.1.2. Commande prédictive floue de type 2 (CPF2)

Les systèmes de logique floue de Type-2 (T2FLS) constituent une extension naturelle des systèmes flous de Type-1, offrant un degré de liberté supplémentaire pour modéliser l'incertitude au sein des fonctions d'appartenance. Leur caractéristique principale est le Footprint of Uncertainty (FOU), dans lequel les degrés d'appartenance ne sont pas exprimés par une valeur précise unique, mais par un intervalle ou un ensemble. Cette représentation améliorée accroît la résilience face aux fluctuations de paramètres, au bruit et aux erreurs de modélisation.

Lorsqu'ils sont intégrés au Model Predictive Control (MPC), les T2FLS permettent de créer des contrôleurs plus flexibles et robustes que leurs homologues de Type-1. Par exemple, le T2 T-S FMPC a été appliqué à des systèmes de contrôle en réseau soumis à des pertes de paquets et des incertitudes de paramètres, où la découplage des fonctions d'appartenance de la plante et du contrôleur a réduit la conservativité et amélioré la stabilité [69,70]. Tang et al. [71] ont en outre introduit un contrôleur prédictif avec un observateur d'état pour gérer les pertes de paquets et la quantification des données, conçu hors ligne via LMIs avec des mises à jour des bornes d'erreur pour garantir la faisabilité récursive.

Les modèles flous de Type-2 ont également été combinés avec le *Generalized Predictive Control* (GPC) pour le contrôle adaptatif de systèmes tels que les réservoirs d'eau interconnectés et les réacteurs de fermentation de levure. Dans ces cas, le modèle flou a été entraîné en ligne en utilisant la descente de gradient pour les paramètres de prémisse et le RLSDF pour les conséquents, surpassant ainsi les modèles ARX et les systèmes flous de Type-1 [72,73].

Pour les **systèmes à paramètres distribués** (DPSs) décrits par des EDP, des techniques de réduction de modèle telles que la PCA ont été employées pour dériver des modèles en temps continu d'ordre réduit adaptés au contrôle prédictif. Cela a permis de stabiliser des procédés non linéaires et instables comme les réacteurs à barres catalytiques [74]. Dans les applications aérospatiales, le T2 FMPC a été appliqué au suivi de trajectoire de parafoils guidés, où les paramètres GPC auto-ajustables via T2FLS ont amélioré la précision et le rejet des perturbations [75].

D'autres contributions incluent l'utilisation de *réseaux neuronaux flous de Type-2* (T2FNNs) dans un cadre de Feedback Error Learning (FEL) pour contrôler des systèmes non linéaires avec retard d'entrée, où la stabilité était garantie par une approche Lyapunov [76]. Plusieurs travaux ont également abordé les défis des systèmes de contrôle en réseau, y compris le contrôle prédictif déclenché par événement avec perte de paquets modélisée par une chaîne de Markov [77], et le MPC en rétroaction sur sortie dynamique avec déclenchement par événement pour des états non mesurables [78].

Les applications se sont en outre étendues à des domaines diversifiés:

- **Systèmes médicaux:** T2 fuzzy MPC d'ordre fractionnaire pour la régulation de la glycémie chez les patients diabétiques [79].
- **Agriculture:** T2 T-S MPC avec une structure non-PDC et une fonction de Lyapunov non quadratique pour le contrôle climatique des serres [80].
- **Véhicules autonomes:** MPC adaptatif intégré à un système de gestion énergétique basé sur T2FLS pour améliorer l'efficacité énergétique et la sécurité [81].
- **Procédés industriels:** Réglage PID en temps réel dans des CSTRs bruités utilisant des systèmes flous de Type-2 [82].
- **Systèmes électriques :** commande prédictive floue adaptative de type 2 (T2 FMPC) pour l'amortissement des oscillations de fréquence dans des réseaux interconnectés, utilisant les dispositifs VFT et SMES [83].

Le contrôle prédictif du modèle flou de Type-2 offre plusieurs avantages clés:

- Meilleure gestion de l'incertitude par rapport aux systèmes de Type-1.
- Flexibilité de conception grâce à la séparation des fonctions d'appartenance de la plante et du contrôleur.
- Potentiel adaptatif via des mécanismes d'apprentissage en ligne.
- Charge computationnelle réduite lorsqu'il est combiné avec la réduction de modèle.
- Robustesse accrue face aux perturbations, à l'incertitude et aux pertes de données.

I.3.2. Classification selon la nature du système contrôlé

A) Systèmes linéaires

Les systèmes linéaires sont typiquement plus faciles à modéliser et à contrôler, car ils peuvent être exprimés à travers des équations différentielles linéaires ou sous forme d'espace d'état, démontrant une relation linéaire distincte entre les entrées et les sorties. Le contrôle prédictif flou (Fuzzy MPC) est utilisé dans ce contexte pour améliorer les performances grâce à l'ajustement des poids de la fonction de coût ou à une gestion plus adaptable des contraintes opérationnelles.

De nombreuses recherches ont démontré l'efficacité du MPC flou (Fuzzy-MPC) dans les systèmes linéaires. Li et Yang en 2003 [84] ont proposé une technique de contrôle prédictif contraint utilisant une optimisation floue multi-objectifs pour traiter les objectifs et contraintes flous. Cette méthode offrait une plus grande flexibilité pour consolider plusieurs objectifs par rapport à la technique conventionnelle consistant à ajouter des erreurs au carré pondérées, et son efficacité a été confirmée par sa mise en œuvre dans la planification de trajectoire visuelle de robots. De même, Sedghizadeh & Beheshti en 2017 [85] ont proposé un algorithme de Subspace Predictive Control à gains programmés flous pour des systèmes linéaires d'ordre faible. Dans cette conception, la logique floue a été utilisée pour planifier dynamiquement les gains du contrôleur de la Subspace Predictive Control (SPC), permettant au système de s'adapter aux dynamiques changeantes tout en conservant des performances maximales et la stabilité.

Des applications supplémentaires démontrent la polyvalence du MPC flou dans les domaines linéaires:

- Dans **le contrôle distribué**, Francisco et al. En 2019 [86] ont introduit un schéma DMPC (Distributed MPC) basé sur une négociation floue pour le classique système linéaire à quatre réservoirs. Ici, la logique floue a été utilisée pour évaluer les actions de contrôle proposées en fonction du coût opérationnel et des contraintes, aboutissant à une décision de contrôle finale fluide et efficace, sans la charge computationnelle associée à l'énumération de toutes les combinaisons possibles.
- Dans **le contrôle des véhicules**, Mao et al. En 2021 [87], ont présenté une stratégie Fuzzy-MPC pour le régulateur de vitesse adaptatif (ACC) utilisant un modèle linéaire de dynamique longitudinale entre deux véhicules. Un système flou a été conçu pour ajuster en ligne les poids de la fonction de coût (par exemple, suivi versus confort) en fonction de variables en temps réel telles que la distance relative et la vitesse relative. Cela a permis au contrôleur de s'adapter aux conditions de conduite variables (normales versus d'urgence), atteignant un meilleur compromis

entre sécurité, confort et efficacité énergétique par rapport à une MPC conventionnelle à poids fixes.

Ces applications soulignent l'efficacité et la mise en œuvre rapide du MPC flou dans les systèmes linéaires, grâce à la complexité mathématique et computationnelle réduite comparée aux systèmes non linéaires. Cependant, leur efficacité devient limitée lorsqu'il s'agit de systèmes très complexes présentant de fortes non-linéarités.

Les avantages du MPC flou pour les systèmes linéaires sont les suivants:

- La nature linéaire du système facilite l'intégration de la logique floue dans l'ajustement de la fonction de coût ou des contraintes [84]. Cette simplification du processus de conception et d'implémentation constitue l'un des bénéfices du système.
- Dans la MPC linéaire avec intégration floue, la dépendance aux problèmes d'optimisation convexe permet une implémentation en ligne efficace [85]. C'est l'une des raisons pour lesquelles la faisabilité en temps réel est possible.
- La logique floue est un outil utile pour gérer les inexactitudes de modélisation ou de mesure qui se situent dans la plage de fonctionnement linéaire [88]. La logique floue améliore l'efficacité dans les situations où l'incertitude est modérée.
- En permettant l'intégration flexible des critères opérationnels et économiques dans la prise de décision coopérative, la performance multi-objectifs dans les systèmes distribués peut être améliorée, comme indiqué par Francisco et al. en 2019 [86].
- L'ajustement dynamique des stratégies de contrôle (par exemple, les poids de la fonction de coût) assure de meilleures performances dans une variété de scénarios [87]. Cela signifie que le système est hautement adaptable à des conditions environnementales différentes.

B) Systèmes non linéaires

Concevoir le contrôle des systèmes non linéaires est difficile, due à leur nature complexe et à la difficulté de modélisation adéquate à l'aide de cadres linéaires conventionnels. À cet égard, les modèles flous, en particulier dans le cadre des modèles flous Takagi-Sugeno (TS), s'avèrent une logique viable. Un tel modèle approxime la dynamique d'un système non linéaire à l'aide d'une fusion linéaire par morceaux de modèle locaux. Ainsi, ils améliorent la prédictibilité et la conception du contrôle et réduisent en même temps les challenges de la modélisation non linéaire directe.

Beaucoup de mises en œuvre réussies du Fuzzy Model Predictive Control pour les systèmes non linéaires existent dans la littérature. Dans les piles à combustible à oxyde solide (SOFC), le modèle Fuzzy Hammerstein a été utilisé pour caractériser le système comme un composant non linéaire statique suivi d'une dynamique linéaire, facilitant ainsi son intégration dans un cadre de contrôle prédictif [89]. Un modèle flou T-S amélioré a été créé pour réguler la température des piles SOFC, celui-ci étant identifié et actualisé en temps réel à l'aide de données entrée-sortie, présentant des performances robustes en fonctionnement en boucle fermée [90].

Dans des systèmes dynamiques plus complexes, tels que les *véhicules hypersoniques*, un schéma de *contrôle prédictif flou adaptatif indirect* a été proposé pour approximer des fonctions non linéaires inconnues ou incertaines au sein du système. Des lois adaptatives sont incorporées pour garantir la stabilité du système et améliorer les performances en présence d'incertitudes et de perturbations [91]. Des stratégies comparables ont également été appliquées avec succès aux *unités chaudière-turbine ultra-supercritiques*, où des modèles flous T-S capturent les dynamiques non linéaires sur une large plage de fonctionnement, et des observateurs d'état étendus flous (FESO) sont utilisés pour améliorer l'estimation des perturbations et la précision du contrôle [92-93].

Des études récentes ont démontré l'efficacité du Fuzzy MPC dans des domaines non linéaires. Zhu et Nguang (2020) ont introduit un cadre qui combine un observateur de perturbations flou avec un contrôleur auxiliaire, permettant un suivi sans décalage et garantissant la faisabilité récursive en présence de perturbations groupées [94]. Zhu (2021) a introduit un contrôle prédictif flou à double mode (MPC) pour les unités chaudière-turbine, en utilisant un ensemble minimal robuste positivement invariant pour atténuer les perturbations et améliorer la réponse dynamique [95]. Dans le positionnement des plateformes semi-submersibles, le Fuzzy MPC utilisant une fonction de Lyapunov de contrôle garantissait la stabilité ; cependant, il nécessitait des ressources computationnelles substantielles, qui ont été atténuées par l'utilisation de variables de relâchement [96].

Avantages du MPC flou pour les systèmes non linéaires: 1. **Strong capability in modeling complexity:** L'approximation du système flou de T-S peut modéliser précisément les systèmes non linéaires sur un large domaine opérationnel. 2. **Flexible par rapport à l'incertitude et aux perturbations:** Les MPC flous prédictifs adaptatifs et basés sur des observateurs peuvent traiter efficacement les variations des paramètres, les perturbations externes et les changements de structuration [91], [94]. et 3. **Amélioration du suivi et des performances:** Les applications SOFC, chaudière-turbine et véhicule hypersonique ont

systématiquement montré une amélioration de la précision du suivi et de la résilience par rapport au MPC linéaire conventionnel [89-90], [95].

I.4. Architectures et stratégies d'intégration du contrôle prédictif flou

I.4.1. Intégration de modèles flous comme prédicteurs internes

L'intégration de modèles flous (*fuzzy models*) dans le cadre du Model Predictive Control (MPC) constitue une méthode courante pour la gestion des systèmes non linéaires et complexes [97-100]. Le concept fondamental consiste à remplacer le modèle linéaire traditionnel utilisé pour la prédiction par un modèle flou plus flexible. Ce dernier est capable d'encapsuler des comportements non linéaires difficiles à décrire uniquement à l'aide d'équations linéaires, permettant ainsi au système MPC de générer des prédictions plus précises et de s'adapter efficacement aux variations des conditions de fonctionnement [101-103].

L'un des cadres les plus couramment utilisés dans ce domaine est le modèle Takagi–Sugeno (T–S). Dans cette approche, l'espace de fonctionnement du système est divisé en plusieurs régions locales, et au sein de chaque région, un modèle linéaire est développé et lié à une règle floue "Si-Alors" [104-105]. Les sorties de ces modèles locaux sont combinées à travers des fonctions d'appartenance qui quantifient le degré d'activation de chaque règle sous des conditions de fonctionnement spécifiques. Grâce à cette agrégation, un modèle flou complet est obtenu et utilisé pour la prédiction dans le cadre du MPC. Cette structure permet de décrire le comportement non linéaire du système de manière fluide et progressive, évitant ainsi la nécessité de formulations mathématiques très compliquées [106-108].

Parallèlement au modèle T–S, des méthodologies hybrides telles que les modèles neuro-flous et les stratégies d'identification floue basées sur les données peuvent également être utilisées [62]. Dans ces approches, les règles et les fonctions d'appartenance sont dérivées directement des données réelles du système plutôt que des équations physiques précises. Cette méthodologie offre plus de flexibilité pour gérer des systèmes difficiles à représenter analytiquement, tout en facilitant le développement de prédicteurs flous capables d'apprendre et de s'adapter aux caractéristiques du système en temps réel [109-110].

I.4.1.1. Structure mathématique générale du modèle T–S

Le modèle flou de Takagi–Sugeno (T–S) est généralement représenté par un ensemble de règles de la forme:

$$\text{R\`egle } i : \text{ Si } z_1(k) \text{ est } F_{i1} \text{ et ... et } z_p(k) \text{ est } F_{ip}, \text{ alors } \begin{cases} x(k+1) = A_i x(k) + B_i u(k) \\ y(k) = C_i x(k) \end{cases} \quad (I.16)$$

Où:

- z : repr\`esentent les **variables de pr\`emisse** qui d\`eterminent les **r\`egions de fonctionnement**.
- F : sont les **ensembles flous** d\`efinis sur ces variables.
- A_i, B_i, C_i sont les **matrices du mod\`ele lin\`eaire local** correspondant \`a la r\`egle i .

La sortie globale du syst\`eme est obtenue sous la forme d'une combinaison lin\`eaire pond\`er\`ee des sorties des mod\`eles locaux:

$$x(k+1) = \sum_{i=1}^r w_i(z(k)) (A_i x(k) + B_i u(k)) \quad (I.17)$$

Soumis \`a:

$$y(k) = \sum_{i=1}^r w_i(z(k)) C_i x(k) \quad (I.18)$$

$$\text{Et } : \sum_{i=1}^r w_i(z(k)) = 1 \quad , \quad w_i(z(k)) \geq 0$$

Int\`egration du mod\`ele flou dans la formulation MPC

Dans le cadre du Model Predictive Control (MPC), le mod\`ele ci-dessus est utilis\`e pour g\`en\`erer des pr\`edictions des \`etats et des sorties du syst\`eme sur un horizon de pr\`ediction N_p . Le contr\`oleur pr\`edit le comportement futur du syst\`eme \`a partir des entr\`ees candidates en utilisant les \`equations du mod\`ele T-S, puis d\`etermine la s\`equence de commande optimale $u(k), u(k+1), \dots$ en r\`esolvant un probl\`eme d'optimisation. Un probl\`eme d'optimisation typique est formul\`e comme suit:

$$\min_U J = \sum_{j=1}^{N_p} \|y(k+j) - y_{ref}(k+j)\|_Q^2 + \sum_{j=0}^{N_c-1} \|\Delta u(k+j)\|_R^2 \quad (I.19)$$

Où: $U = [u(k), u(k+1), \dots, u(k+N_c-1)]$ est le vecteur de d\`ecision. Q, R sont les matrices de pond\`eration dans la fonction de co\`ut. Et, N_p, N_c sont respectivement les horizons de pr\`ediction et de commande. Avantages cl\`es de cette approche:

- **Capacit\`e \`a repr\`esenter la non-lin\`earit\`e:** Le partitionnement de l'espace de fonctionnement en r\`egions locales permet de conserver un mod\`ele simple, tout en \`etant capable de repr\`esenter des comportements complexes.
- **Impl\`ementabilit\`e pratique:** Le mod\`ele reste sous une forme lin\`eaire par morceaux, permettant de formuler l'optimisation comme un probl\`eme de Programmation Quadratique (QP).

- Flexibilité dans l'apprentissage: Les modèles flous peuvent être construits à partir de données expérimentales lorsque la modélisation analytique est difficile.

I.4.1.2. Stratégies de conception dans le MPC basé sur un modèle flou

Lors de l'intégration de modèles flous dans le cadre du Model Predictive Control (MPC), le défi dépasse la simple précision du modèle prédictif pour inclure la manière dont le contrôleur est conçu en harmonie avec la structure du modèle flou. Parmi les méthodologies proposées dans ce contexte, l'approche dite de Compensation Distribuée Parallèle (PDC – Parallel Distributed Compensation) se distingue comme étant la stratégie la plus largement utilisée et la plus couramment adoptée.

La stratégie de Compensation Distribuée Parallèle (PDC) constitue l'une des méthodologies de conception les plus courantes dans le cadre du MPC basé sur un modèle flou.

L'idée fondamentale consiste à concevoir un contrôleur MPC local pour chaque règle du modèle flou de Takagi–Sugeno (T–S), de telle sorte que chaque modèle local possède son propre contrôleur prédictif. L'entrée de commande globale est ensuite obtenue comme une combinaison convexe des sorties de tous les contrôleurs locaux, en utilisant les mêmes fonctions d'appartenance $w_i(z(k))$ employées lors de l'étape de modélisation.

Cette structure assure une cohérence entre le modèle flou et le contrôleur, ce qui constitue un avantage majeur lors de l'analyse de la stabilité et de la performance. De plus, le cadre PDC (Parallel Distributed Compensation) facilite la dérivation des conditions de stabilité basées sur les Inégalités Linéaires Matricielles (LMI – Linear Matrix Inequalities), permettant ainsi une vérification systématique de la stabilité en boucle fermée, même en présence de contraintes. Mathématiquement, pour chaque règle floue i , un problème d'optimisation MPC local est formulé comme suit:

$$\min_{U_i} J_i = \sum_{j=1}^{N_p} \|y_i(k+j) - y_{ref}(k+j)\|_Q^2 + \sum_{j=0}^{N_c-1} \|\Delta u_i(k+j)\|_R^2 \quad (I.20)$$

Soumis à la dynamique locale:

$$x(k+1) = A_i x(k) + B_i u_i(k) \quad , \quad y(k) = C_i x(k) \quad (I.21)$$

L'entrée de commande finale appliquée au procédé est ensuite calculée comme:

$$u(k) = \sum_{i=1}^r w_i(z(k)) u_i(k) \quad (I.22)$$

Où $w_i(z(k))$ représentent les fonctions d'appartenance floues, également utilisées dans le modèle de prédiction.

Avantages de l'approche PDC:

- Préserve la cohérence entre le modèle et le contrôleur.
- Facilite l'analyse de stabilité en utilisant les LMI (Linear Matrix Inequalities).
- Offre la capacité de traiter des systèmes non linéaires tout en maintenant des problèmes d'optimisation gérables.

Défis:

- La complexité computationnelle augmente avec le nombre de règles floues.
- Nécessite une sélection minutieuse des variables de prémisse afin d'éviter une sur-partition de l'espace de fonctionnement.

I.4.1.3. Identification et apprentissage du modèle dans le MPC basé sur un modèle flou

Le processus de construction des modèles flous constitue une étape fondamentale dans l'efficacité du MPC basé sur un modèle flou, car la précision du contrôleur prédictif dépend directement de la qualité du modèle utilisé pour la prédiction. Plusieurs stratégies ont été proposées dans la littérature pour la construction et l'apprentissage de tels modèles, allant du partitionnement de l'espace de fonctionnement basé sur le clustering à des approches neuro-floues hybrides.

Une méthode largement utilisée est le clustering flou, tel que le Fuzzy Satisfactory Clustering (FSC) et le Fuzzy C-Means (FCM) [111], qui sont employés pour diviser l'espace de fonctionnement en régions locales et extraire automatiquement les règles floues. Cette approche permet d'atteindre un équilibre entre la précision du modèle et les exigences computationnelles, et a été appliquée avec succès dans des procédés non linéaires de pH ainsi que dans des systèmes de chauffage, ventilation et climatisation (HVAC) [104], [112].

D'autres travaux de recherche se sont concentrés sur l'identification en ligne et l'estimation récursive, telles que l'utilisation des moindres carrés récurrents pondérés (WRLS – Weighted Recursive Least Squares) dans les générateurs de vapeur. Cette méthode a permis une mise à jour continue des paramètres du modèle ainsi qu'une adaptation aux conditions de fonctionnement changeantes [113]. De la même manière, un clustering flou récursif avec adaptation en ligne a été utilisé dans des réacteurs semi-batch afin de permettre aux contrôleurs prédictifs de gérer des systèmes à variation temporelle [114].

Les méthodologies hybrides, incluant les architectures neuro-floues (par exemple ANFIS et RFNN), ont été utilisées pour dériver directement à partir des données à la fois la base de règles et les fonctions d'appartenance. Ces modèles combinent les capacités d'apprentissage des réseaux de neurones avec

l'interprétabilité de la logique floue, et ont démontré leur efficacité dans des applications pratiques telles que la régulation de la température de vapeur dans les chaudières et les procédés industriels batch [115].

Dans les cas où la modélisation physique du procédé est difficile, certaines études se sont appuyées sur une identification floue basée sur les données (**data-driven fuzzy identification**).

Cette approche intègre le clustering, la régression, et les techniques d'estimation récursive afin de construire des modèles flous compacts et adaptatifs, convenant à une implémentation en temps réel dans le cadre du MPC [116]. Les principaux avantages de ces stratégies d'identification incluent:

- *Flexibilité*: la capacité à dériver des modèles directement à partir de données expérimentales ou opérationnelles.
- *Adaptativité*: la mise à jour en ligne assure une robustesse face aux perturbations et aux dynamiques variables dans le temps.
- *Efficacité*: les techniques de clustering réduisent le nombre de règles et préviennent une complexité excessive du modèle.

Malgré ces avantages, certains défis persistent, notamment le coût computationnel élevé lors de l'apprentissage en ligne et la nécessité de garantir la stabilité en boucle fermée lorsque les paramètres sont continuellement mis à jour.

1.4.1.4. Vérification de la stabilité dans le MPC basé sur un modèle flou

Garantir la stabilité constitue l'un des principaux défis du MPC basé sur un modèle flou, puisque le contrôleur prédictif repose sur une combinaison convexe de modèles locaux générés à travers la structure floue de Takagi–Sugeno (T–S). Parmi les méthodes les plus largement adoptées, les Inégalités Linéaires Matricielles (**LMI – Linear Matrix Inequalities**), utilisées dans le cadre de la Compensation Distribuée Parallèle (**PDC – Parallel Distributed Compensation**), se sont révélées être des approches efficaces et systématiques [117-118]. Le modèle flou T–S peut s'exprimer comme suit:

$$x(k + 1) = \sum_{i=1}^r w_i(z(k)) (A_i x(k) + B_i u(k)) \quad , \quad y(k) = \sum_{i=1}^r w_i(z(k)) C_i x(k) \quad (1.23)$$

avec des fonctions d'appartenance $w_i(z(k)) \geq 0$ et $\sum_{i=1}^r w_i(z(k)) = 1$.

Dans le cadre PDC , un contrôleur local est conçu pour chaque règle floue i :

$$u_i(k) = K_i x(k) \quad (1.24)$$

et l'entrée de commande globale est calculée comme une combinaison convexe:

$$u(k) = \sum_{i=1}^r w_i(z(k)) K_i x(k) \quad (1.25)$$

Les dynamiques en boucle fermée sont alors données par:

$$x(k+1) = \sum_{i=1}^r w_i(z(k)) (A_i + B_i K_i) x(k) \quad (1.26)$$

Afin d'assurer la stabilité, on cherche une fonction de Lyapunov quadratique commune $V(x) = x^T P x$ avec $P > 0$ telle que, pour chaque modèle local i :

$$(A_i + B_i K_i)^T P (A_i + B_i K_i) - P < 0 \quad (1.27)$$

Cette condition peut être reformulée sous forme d'une LMI:

$$\begin{bmatrix} P & (A_i + B_i K_i)^T P \\ P(A_i + B_i K_i) & P \end{bmatrix} > 0, \quad P > 0, \quad i = 1, \dots, r \quad (1.28)$$

La faisabilité de ces Inégalités Linéaires Matricielles (LMI – Linear Matrix Inequalities) garantit la stabilité du contrôleur prédictif flou pour toutes les régions de fonctionnement.

Cette méthodologie a été largement appliquée dans le contrôle des procédés et les systèmes énergétiques, car elle fournit un cadre systématique et computativement gérable pour la vérification de la stabilité en boucle fermée [118-121].

La Figure 1.20 illustre la structure du MPC basé sur un modèle flou avec Compensation Distribuée Parallèle (PDC – Parallel Distributed Compensation). Le système est divisé en plusieurs modèles linéaires locaux, chacun étant associé à une règle floue. Un contrôleur MPC local est conçu pour chaque règle, et le signal de commande final est obtenu comme une combinaison pondérée de tous les contrôleurs locaux, en utilisant les mêmes fonctions d'appartenance que celles du modèle.

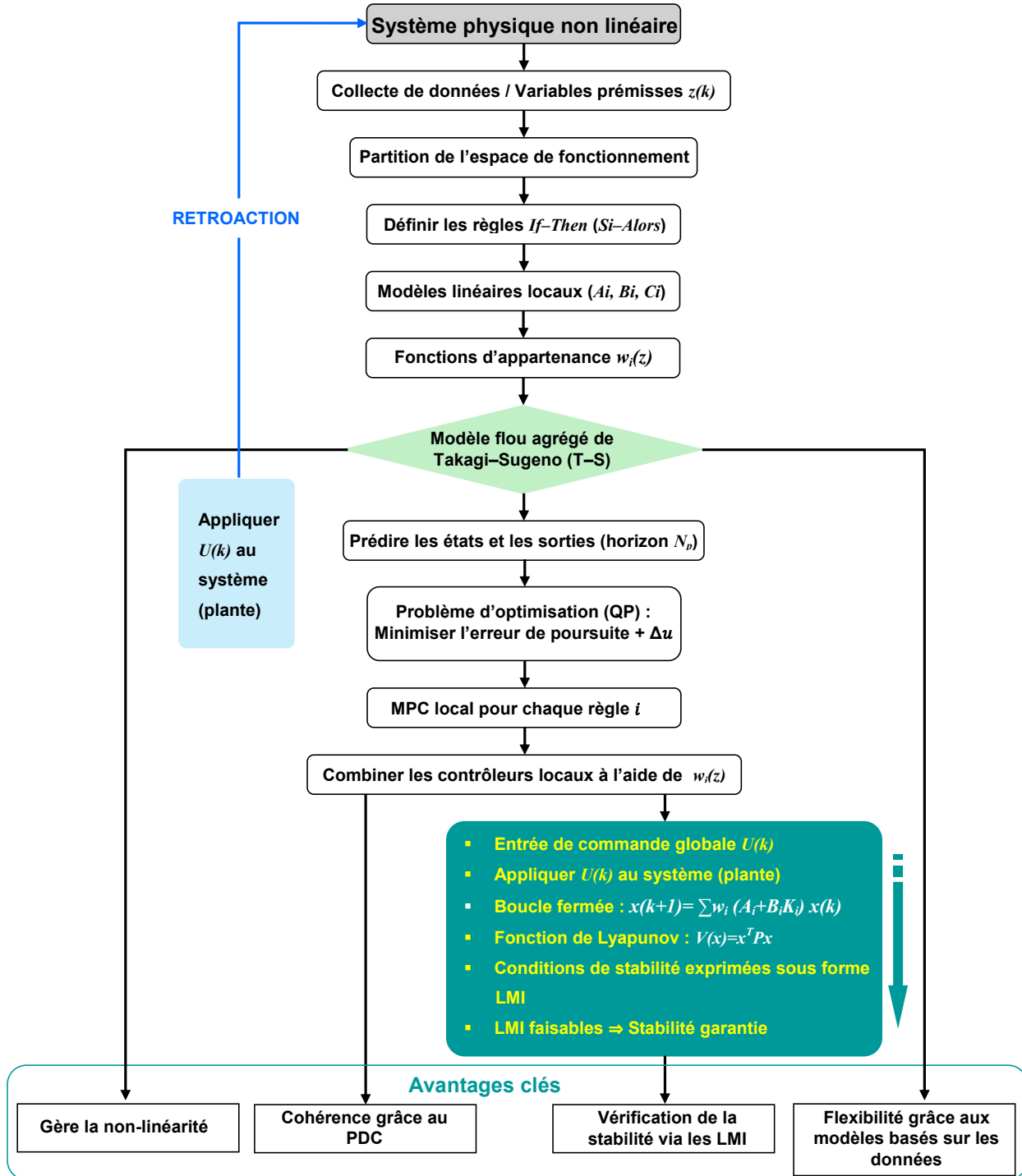


Figure I.20. MPC basé sur un modèle flou avec compensation distribuée parallèle.

I.4.2. Gestion des contraintes floues dans le contrôle prédictif

Dans le MPC classique, les contraintes sont formulées sous forme de bornes strictes, ce qui signifie que toute violation rend le problème d'optimisation infaisable. Dans le MPC flou, ces contraintes sont

remplacées par **des contraintes souples**, où de petites violations sont tolérées et attribuées à différents niveaux d'acceptation. Cela est réalisé grâce à des fonctions d'appartenance qui fournissent à chaque solution candidate un degré de satisfaction plutôt qu'une décision stricte oui/non. Ainsi, l'optimiseur peut équilibrer les violations légères des contraintes contre une amélioration des performances sans provoquer de blocage [84].

A. Représentation des contraintes avec des fonctions d'appartenance (et le concept de coupe α)

Au lieu d'écrire une borne stricte comme: $y \leq y_{max}$. la contrainte est exprimée à travers une fonction d'appartenance:

- Si y est dans la limite \rightarrow pleinement acceptable ($\mu = 1$).
- Si y dépasse légèrement la limite \rightarrow partiellement acceptable ($0 < \mu < 1$).
- Si y dépasse largement la limite \rightarrow inacceptable ($\mu = 0$).

Le concept de coupe α

Pour rendre cette représentation utilisable dans des formulations de programmation quadratique (QP) ou des Inégalités Linéaires Matricielles (LMI), le concept de coupe α est appliqué. Il extrait un ensemble faisable précis à un niveau de confiance choisi $\alpha \in [0,1]$:

$$A_\alpha = \{y: \mu(y) \geq \alpha\}.$$

- Avec un **haut** α , la borne résultante est très **stricte**.
- Avec un **bas** α , la contrainte est plus **relâchée**.

Cela permet au concepteur d'ajuster la rigidité ou la flexibilité des contraintes en fonction de l'application. La Figure I.21 montre le concept de gestion des contraintes dans le contrôle prédictif flou en utilisant des contraintes souples. Chaque contrainte est représentée par une fonction d'appartenance qui fournit différents degrés de satisfaction, permettant des violations limitées au prix de pénalités dans la fonction objectif.

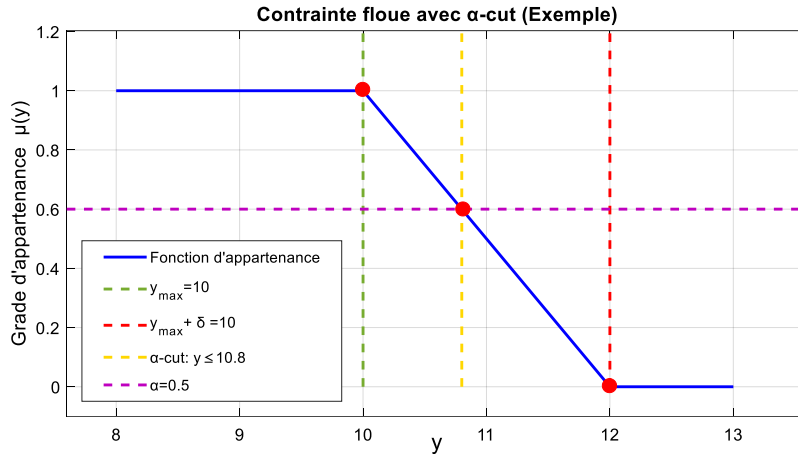


Figure I.21. Gestion des contraintes souples dans le contrôle prédictif flou.

B. Exemples d'applications:

La gestion des contraintes floues a été démontrée dans plusieurs applications réelles du MPC, y compris:

- Régulation des réacteurs à cuve agitée continue (CSTR): où les contraintes floues, combinées avec des conditions de stabilité basées sur les LMI, garantissent la stabilité exponentielle en boucle fermée sous des conditions de fonctionnement variables, comme montré dans Khairy et al. [122].
- Contrôle des centrales solaires: mise en œuvre du contrôle prédictif flou pour réguler la température de sortie de l'huile en utilisant des objectifs et des contraintes flous, offrant plus de flexibilité par rapport au contrôle prédictif basé sur un modèle traditionnel [123].
- Régulation des éoliennes: utilisation du contrôle prédictif multivarié basé sur un modèle flou avec des contraintes LMI pour gérer la vitesse du rotor et la production d'énergie dans des conditions de vent fluctuantes [121].
- Contrôle du convertisseur matriciel: où une stratégie de prise de décision floue remplace les facteurs de pondération dans le MPC à états finis pour gérer simultanément la traçabilité des courants d'entrée et de sortie [124].

Ces exemples montrent comment la gestion des contraintes floues peut être utilisée pour réaliser un contrôle robuste et multi-objectifs dans un large éventail de systèmes industriels.

I.4.3. Contrôle prédictif flou adaptatif

L'AFMPC est une méthode sophistiquée combinant les capacités de prévision explicites et multi-étapes du contrôle prédictif par modèle avec l'approximation universelle des systèmes de logique floue.

L'AFMPC intègre des mécanismes d'adaptation cruciaux pour maintenir l'efficacité du contrôleur face aux dynamiques réelles du système. Trois principales motivations justifient l'intégration de l'adaptation dans les systèmes de contrôle prédictif flou:

Tout d'abord, les processus industriels subissent fréquemment des fluctuations dans les conditions de fonctionnement, telles que des changements de charge, de consignes ou des impacts environnementaux. Les dynamiques de réaction dans un réacteur à cuve agitée continue (CSTR) sont significativement non linéaires et très sensibles aux changements de débit de refroidissement et de concentration d'alimentation [113]. Les dynamiques d'un générateur de vapeur nucléaire fluctuent considérablement avec le niveau de puissance, nécessitant un contrôleur capable de s'adapter à ces variations pour éviter les arrêts du réacteur ou les dommages à la turbine. Un contrôleur non adaptatif conçu pour un point de fonctionnement particulier peut présenter des performances sous-optimales ou une instabilité dans des conditions variables.

Ensuite, le décalage entre le modèle et la plante est un défi inhérent. Bien que les modèles flous de Takagi-Sugeno (T-S) soient très efficaces pour approximer des systèmes non linéaires complexes en combinant des modèles linéaires locaux, cependant, le vieillissement du système ou la dérive du processus peuvent en altérer leur précision. Pour y remédier, des techniques d'estimation de paramètres en ligne telles que l'algorithme des Moindres Carrés Récurifs Pondérés (WRLS) sont utilisées pour réviser les paramètres conséquents du modèle flou, reflétant ainsi avec précision la dynamique du processus [113], [125].

Troisièmement, l'existence de bruit, d'incertitude et de perturbations nécessite la mise en œuvre de stratégies adaptatives robustes. Les mesures souffrent fréquemment de corruption due au bruit, et les perturbations externes non modélisées peuvent interférer de manière significative avec l'intégrité opérationnelle des processus. Les mécanismes adaptatifs, tels que l'incorporation de facteurs d'oubli dans l'estimation réursive, facilitent la capacité du contrôleur à diminuer l'influence des informations obsolètes tout en priorisant les données contemporaines, augmentant ainsi la résilience face à de telles incertitudes [126]. De plus, le problème d'optimisation central au Contrôle Prédictif par Modèle (MPC) peut être complexe et exigeant en termes de calcul, en particulier lorsqu'il s'agit de systèmes non linéaires. Des méthodologies d'optimisation sophistiquées, y compris l'Optimisation par Colonie de Fourmis (ACO), ont été intégrées au Contrôle Prédictif Adaptatif Flou (AFMPC) pour relever efficacement ce défi et identifier des actions de contrôle optimales sans nécessiter des inversions de matrices coûteuses en termes de calcul [127].

Grâce à la mise en œuvre de la mise à jour en ligne du modèle et de techniques d'optimisation avancées, l'AFMPC maintient efficacement une performance soutenue, une robustesse et une fiabilité dans le contrôle de processus industriels complexes.

I.4.3.1. Mécanismes d'adaptation

a. Mise à jour des paramètres du modèle local

Dans le domaine du Contrôle Prédictif Flou Adaptatif, une méthodologie rigoureusement établie implique l'ajustement en temps réel des paramètres du modèle local au sein des modèles flous de T-S. Les principales techniques utilisées pour cette entreprise sont les Moindres Carrés Récursifs (RLS) et leur dérivé, les Moindres Carrés Récursifs Pondérés (WRLS), qui incorporent tous deux des facteurs d'oubli pour traiter efficacement les dynamiques variant dans le temps. La Figure I.22 illustre le mécanisme global de modification des paramètres du modèle local dans le contrôle prédictif flou adaptatif. Des algorithmes tels que RLS et WRLS sont mis en œuvre pour affiner itérativement les paramètres du modèle en fonction des erreurs de prédiction, garantissant ainsi la précision du modèle lorsque les conditions de fonctionnement fluctuent.

- Eliasi et al. en 2007 [125]: Ont appliqué WRLS avec un facteur d'oubli pour mettre à jour les paramètres conséquents d'un modèle flou T-S pour les générateurs de vapeur nucléaires, tout en gardant les prémisses des règles fixes.
- Dovžan & Škrjanc en 2010 [126]: Combiné le clustering récursif flou c-means avec un algorithme RLS flou, permettant l'adaptation simultanée des paramètres de prémisses (centres et largeurs des fonctions d'appartenance) et des paramètres de conséquence dans un réacteur semi-batch.
- Chengli et al. en 2013 [113]: Ont utilisé WRLS pour mettre à jour récursivement les paramètres conséquents dans un processus CSTR, avec des paramètres antécédents maintenus constants.
- Bououden et al. en 2015 [127]: Ont intégré l'identification floue adaptative en ligne en utilisant RLS avec l'optimisation par colonie de fourmis (ACO) pour optimiser les paramètres de contrôle pour le contrôle de processus non linéaires.
- Boulkaibet et al. en 2018 [128]: Ont proposé un schéma d'adaptation de modèle flou T-S basé sur Multi-Kernel LSSVR, où les paramètres antécédents sont ajustés par rétropropagation, et les paramètres conséquents sont adaptés en ligne en utilisant un algorithme de RLS à noyau à budget fixe.

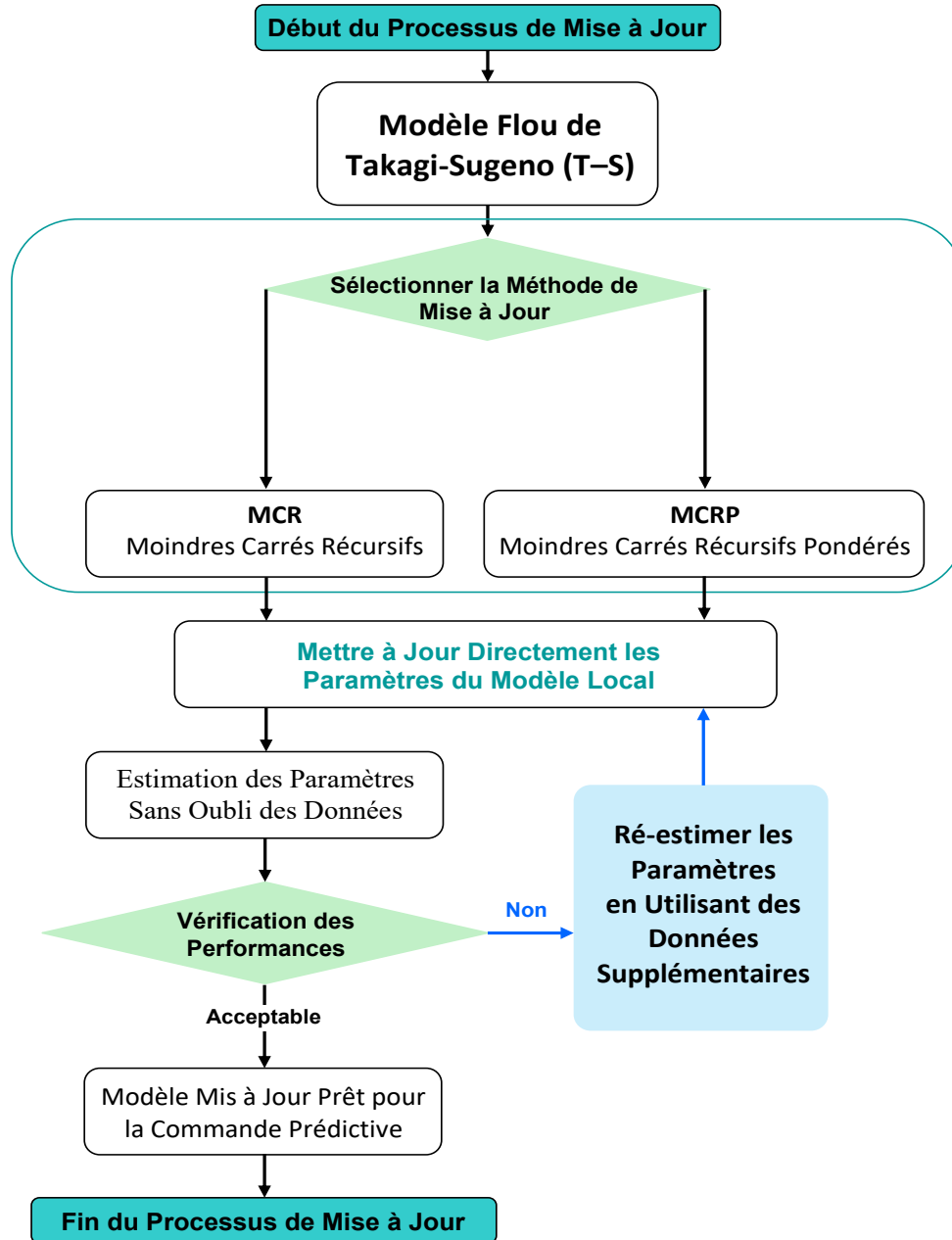


Figure I.22. Mécanisme général pour la mise à jour des paramètres du modèle local dans le contrôle prédictif adaptatif.

Mécanisme général pour la mise à jour des paramètres du modèle local dans le contrôle prédictif adaptatif En intégrant ces contributions, les approches utilisant RLS et WRLS ont prouvé leur capacité à maintenir la précision et la résilience des contrôleurs prédictifs flous, même face à des conditions de fonctionnement fluctuantes, des non-linéarités et des perturbations.

b. Mise à jour des fonctions d'appartenance ou de leurs centres

La modification des fonctions d'appartenance ou de leurs clés centrales constitue un mécanisme d'adaptation crucial dans le Contrôle Prédicatif Flou Adaptatif (AFMPC), améliorant ainsi la capacité du modèle flou à répondre aux conditions opérationnelles fluctuantes et aux dynamiques non linéaires.

Diverses méthodologies ont été articulées dans la littérature académique:

- ✓ Méndez et al. en 2016 [129] ont utilisé des Algorithmes Génétiques (AG) pour optimiser l'adaptation en ligne des ensembles flous en modifiant le paramètre de largeur des fonctions d'appartenance triangulaires corrélées à l'entrée du taux de perfusion, permettant ainsi au contrôleur de traiter efficacement à la fois la variabilité inter-patient et intra-patient.
- ✓ Wang et al. en 2016 [74] ont proposé une méthodologie basée sur des ensembles flous de type 2 à intervalle, qui incorpore des poids variant dans le temps mis à jour en ligne en utilisant un critère de variance minimale dérivé des erreurs prédictives. Les paramètres antécédents ont été discernés par une technique de Clustering Satisfaisant Flou (FSC), tandis que les paramètres conséquents ont été affinés en utilisant la méthode des moindres carrés.
- ✓ Hamdy et al. en 2017 [130] ont créé une méthode pour adapter les paramètres de la fonction d'appartenance en temps réel. Cette routine repose sur la stabilité de Lyapunov. Il ajuste les paramètres pour gérer les caractéristiques non linéaires et les retards de réseau.

La Figure I.25 illustre la modification des fonctions d'appartenance dans le contrôle prédictif flou. Les centres, les formes ou les poids des fonctions d'appartenance sont ajustés si nécessaire, afin que le modèle flou puisse mieux représenter la dynamique réelle du système.

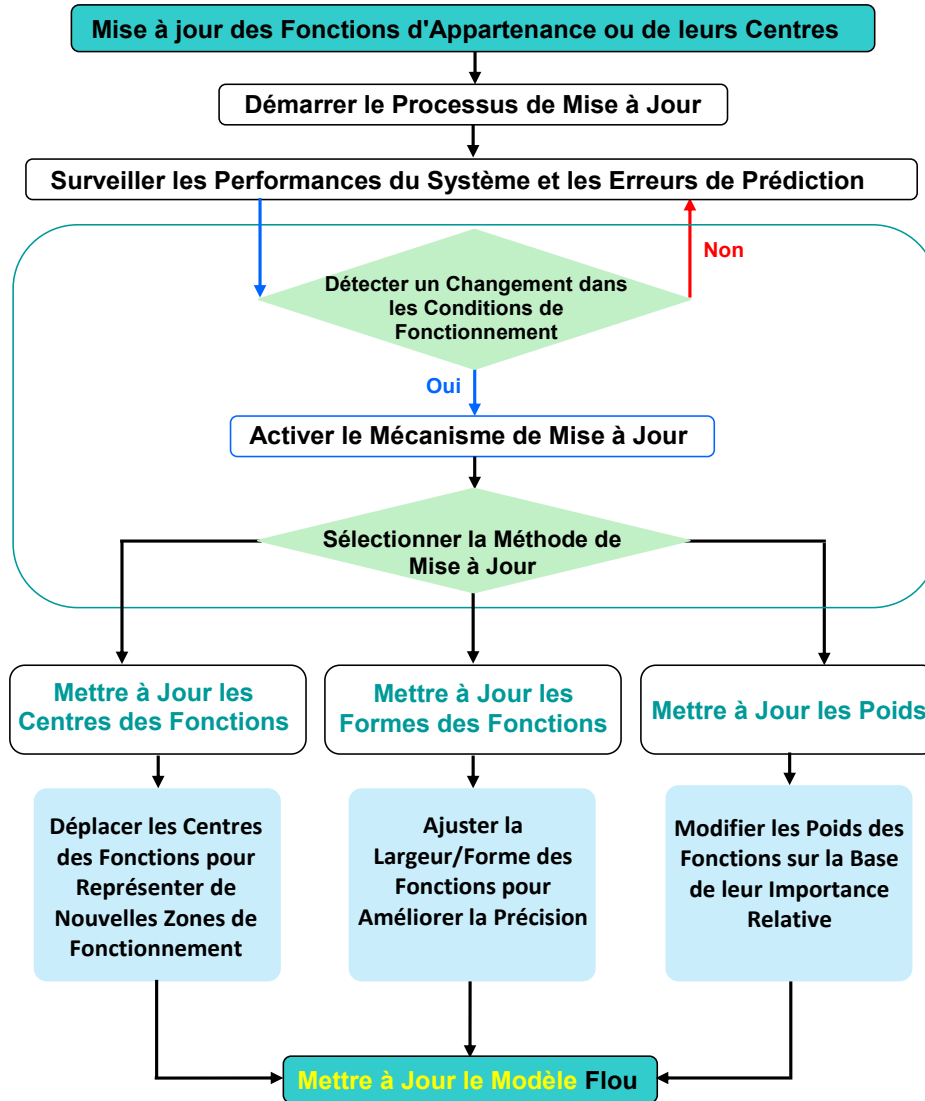


Figure I.23. Mécanisme adaptatif pour les mises à jour des fonctions d'appartenance dans le contrôle prédictif flou

c. Génération/Suppression en ligne des règles

Le changement adaptatif des bases de règles floues est un processus crucial dans le Contrôle Prédictif Flou Adaptatif, permettant au contrôleur de maintenir la compacité et la pertinence de sa structure dans des situations changeantes. Diverses stratégies ont été documentées:

- Mahfouf et al. en 2003 [131] ont impliqué un système qui combinait le Contrôle Prédictif Généralisé (GPC) avec un Contrôleur Logique Flou Auto-Organisant (SOFLC). Ce système est capable de créer, modifier et supprimer des règles floues pendant son fonctionnement. Il utilise une méthode de mémoire en deux parties pour se débarrasser des règles inutiles et maintenir l'ensemble de règles simple.

- Salahshoor et al. en 2013: Ont introduit un modèle flou évolutif de Takagi–Sugeno (eTS) qui génère de nouvelles règles en ligne par le biais d'un regroupement potentiel récursif et élimine les règles inactives en utilisant une méthode d'élagage basée sur un seuil. Cela maintient un ensemble de règles concis et flexible, adapté aux systèmes hybrides caractérisés par des dynamiques variables dans le temps.

- Xiaoying et al. en 2016 [132]: Ont développé un cadre de modèles multiples flous en utilisant le clustering flou c-means et l'identification de sous-espaces. Bien que la base de règles soit initialement construite hors ligne, elle est mise à jour en ligne avec des données en temps réel pour adapter le nombre et la structure des règles, trouvant un équilibre entre précision et complexité dans les applications des centrales électriques.

La Figure I.24 montre les stratégies de génération et de suppression automatiques des règles floues dans le contrôle prédictif flou adaptatif. De nouvelles règles sont créées lorsque des régions de fonctionnement inexplorées apparaissent, tandis que les règles inefficaces sont élaguées pour maintenir une base de règles compacte et efficace.

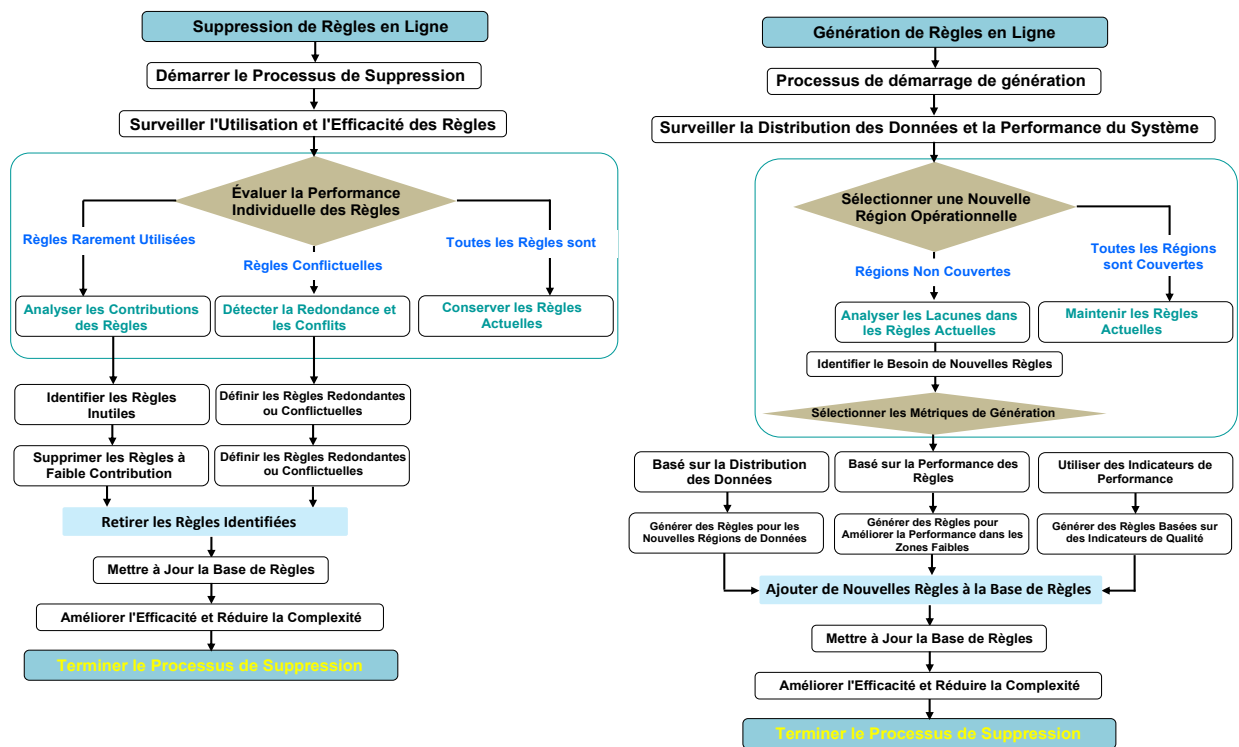


Figure I.24. Stratégies automatiques de génération et d'élagage de règles floues dans la commande prédictive floue adaptative

1.4.3.2. Évolution des méthodes de contrôle prédictif flou adaptatif

Ce tableau 1.6 met en évidence l'évolution des applications de la commande prédictive floue adaptative de 2003 à 2024, montrant comment différentes méthodes d'adaptation ont été employées pour traiter la non-linéarité, l'incertitude et les dynamiques variables dans le temps dans divers systèmes complexes.

Tableau I.6. Méthodes d'adaptation dans les applications de commande prédictive basée sur un modèle flou.

Année	Référence	Domaine d'application	Méthode d'adaptation
2003	Mahfouf et al.	Procédé chimique non linéaire (colonne de distillation)	GPSOFLC: hybride de GPC basé sur TSK et SOFLC de type Mamdani, utilisant RLS pour l'estimation en ligne des paramètres
2007	Eliasi et al.	Contrôle du niveau d'eau d'un générateur de vapeur nucléaire	Commande prédictive adaptative basée sur modèle flou TSK utilisant les moindres carrés récurrents pondérés (WRLS) pour l'adaptation des paramètres des conséquents
2010	Dovžan & Škrianc	Contrôle de température d'un réacteur semi-fermenté	Commande fonctionnelle prédictive floue adaptative (FPFC) utilisant le clustering flou c-means récurrent et RLS pour l'adaptation des paramètres des prémisses et des conséquents
2013	Chengli et al.	Réacteur à cuve agitée continue (CSTR)	Commande prédictive adaptative directe (DAPC) basée sur modèle flou T-S utilisant WRLS pour l'adaptation des paramètres des conséquents
2013	Salahshoor et al.	CSTR hybride avec entrées mixtes	Commande prédictive adaptative non linéaire (NMPC) utilisant modèle flou T-S évolutif (eTS) avec clustering potentiel et RLS pour l'adaptation de structure et des paramètres
2015	Bououden et al.	Commande de procédés non linéaires (CSTR, réservoir tampon)	Identification floue adaptative + Optimisation par colonie de fourmis (ACO) pour le réglage du contrôleur RST
2016	Méndez et al.	Administration d'anesthésie (contrôle de l'hypnose)	Réglage en ligne par algorithme génétique des fonctions d'appartenance floues dans le modèle prédictif
2016	Wang et al.	Systèmes à paramètres distribués non linéaires (réacteur à tige catalytique)	Réglage en ligne du modèle flou Intervalle Type-2 T-S avec poids variables dans le temps via la méthode de variance minimale
2017	Hamdy et al.	Systèmes non linéaires en réseau avec délais variables (oscillateur Van der Pol)	Commande logique floue adaptative (AFLC) combinée avec prédicteur d'état et prédicteur de Smith adaptatif pour la compensation des délais
2018	Boulkaibet et al.	Commande de procédés non linéaires (réservoir tampon, CSTR)	TS-LSSVR adaptatif: clustering FCM hors ligne pour la structure, rétropropagation en ligne pour les antécédents, et KRLS à budget fixe pour l'adaptation des paramètres des conséquents

2024	Manzoor et al.	Contrôle de fréquence de charge dans les systèmes électriques	Commande Prédictive Adaptative (APC) réglée par logique floue de Type-2 avec ajustement en ligne des poids utilisant la logique floue de type 2
------	----------------	---	---

I.4.4. MPC à gain flou programmé

Cette méthode utilise un système flou comme planificateur de gain supervisé pour modifier les paramètres d'un contrôleur de contrôle prédictif de modèle (MPC) linéaire fondamental en fonction du point de fonctionnement actuel du système. Les paramètres calibrés peuvent inclure les matrices de pondération de la fonction de coût (Q, R), les contraintes, ou les horizons de prédiction et de contrôle. L'objectif est de faciliter le fonctionnement efficace d'un contrôleur linéaire, adapté à un point de fonctionnement particulier, à travers diverses conditions de fonctionnement du système non linéaire. Cette stratégie a été mise en œuvre avec succès dans divers contextes, par exemple:

- La recherche a utilisé un cadre de contrôle prédictif à modèles multiples (MMPC). Une caractéristique principale était la mise en œuvre d'un modèle flou hiérarchique pour décomposer le comportement du système non linéaire en une collection de modèles linéaires locaux de Takagi-Sugeno. Le signal de contrôle final a été généré en intégrant les sorties de ces modèles locaux en utilisant des poids flous qui s'ajustent en fonction du point de fonctionnement actuel [112].
- La performance du Contrôle Prédictif par Sous-espace (SPC) a été améliorée en employant un système de Programmation de Gain Flou (FGS), optimisé par l'Optimisation par Essaim de Particules (PSO), pour ajuster les gains du contrôleur en fonction des variations de la dynamique du système. Cette approche met à jour les gains sans nécessiter de signaux d'excitation persistante (PE), réduisant ainsi la charge de calcul et améliorant l'efficacité temporelle [85].

La Figure I.25 illustre l'architecture du Contrôle Prédictif par Modèle à Gain Flou. Cette méthode utilise un contrôle prédictif de modèle linéaire (MPC) avec des paramètres qui sont dynamiquement modifiés par un système de supervision flou fonctionnant comme un planificateur de gain pour garantir une performance optimale dans des conditions de fonctionnement fluctuantes.

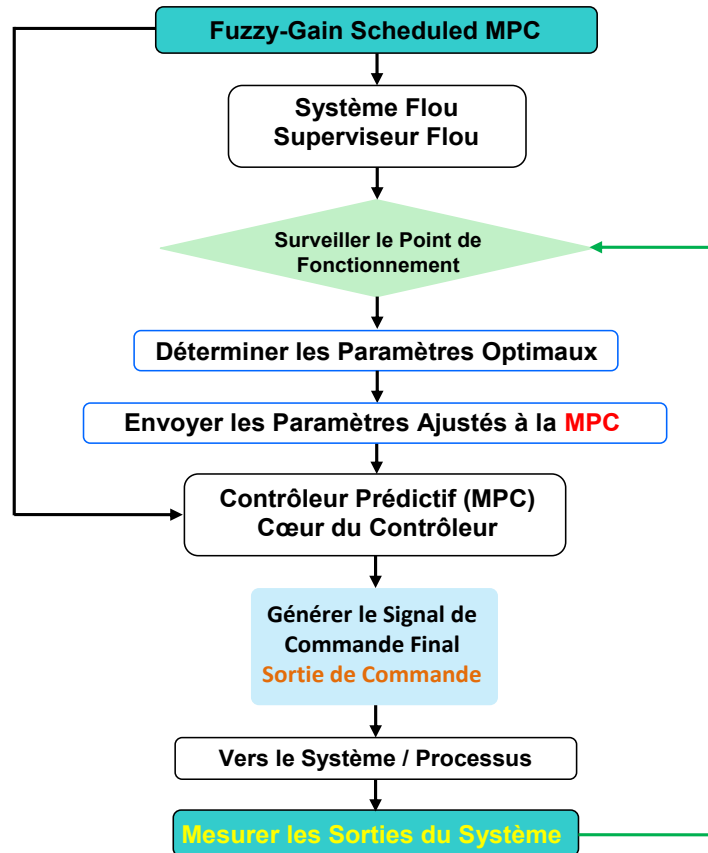


Figure I.25. Structure de la commande prédictive à gain flou

A. Avantages:

- **Simplicité:** Il évite la complexité computationnelle élevée du MPC non linéaire complet (NMPC) en continuant à utiliser un contrôleur linéaire en son cœur, en ajustant seulement ses paramètres.
- **Tire parti Conception bien établie:** Le système peut être construit sur une conception de MPC linéaire éprouvée et fiable, avec l'ajout de la couche d'adaptation floue pour améliorer les performances sur une plage de fonctionnement plus large.
- **Efficacité Computationnelle:** Il est généralement plus rapide à résoudre que le NMPC car il évite la nécessité de résoudre un problème d'optimisation non linéaire en ligne. Au lieu de cela, il résout un problème de Programmation Quadratique (QP) pour le contrôleur linéaire, tandis que le système flou agit comme une logique de commutation/réglage plus simple.

B. Inconvénients:

- Pas un véritable MPC non linéaire: Puisqu'il repose sur un contrôleur central linéaire, il peut ne pas capturer pleinement la dynamique non linéaire de l'usine aussi bien qu'un véritable NMPC utilisant un modèle de prédiction interne non linéaire. La performance pourrait être sous-optimale pour les systèmes hautement non linéaires.
- La performance dépend de la conception: La performance globale dépend fortement de la qualité de la conception de la base de règles floues et des plages de paramètres choisies (Controller Gain Ranges - CGRs). Déterminer ces éléments manuellement peut être difficile et repose sur l'expérience du concepteur.
- Adaptabilité limitée: Le système flou et ses règles sont généralement conçus hors ligne en fonction des connaissances du processus. Il peut ne pas bien s'adapter aux changements significatifs et imprévus dans la dynamique de l'usine qui n'ont pas été pris en compte lors de la phase de conception initiale.

I.5. Conclusion

Ce chapitre a résumé l'état actuel de la recherche sur le contrôle prédictif flou (FMPC), qui combine le contrôle prédictif à base de modèle (MPC) et la logique floue (FL). Nous avons d'abord rappelé les principes fondamentaux du MPC, approche prédictive, optimisation sous contraintes, horizon glissant, ainsi que les bases de la logique floue (LF), variables linguistiques, systèmes d'inférence pour modéliser les incertitudes et les connaissances expertes. Leur synergie crée le FMPC, idéal pour les systèmes complexes, non linéaires et mal spécifiés. Ensuite, nous avons classé les travaux antérieurs selon deux axes: le type de système flou et la nature du système contrôlé. Nous avons aussi expliqué l'intégration du FMPC en utilisant des modèles flous (notamment T-S) comme prédicteurs internes. Finalement, une analyse des avantages (gestion des contraintes, adaptabilité, robustesse) et des limites (charge calculatoire, dépendance au modèle, complexité de réglage) a positionné le FMPC par rapport aux méthodes de contrôle classiques. Le chapitre II portera sur une application fondamentale du FMPC: le Contrôle Prédictif Flou de type 1 pour les systèmes linéaires. En illustrant son utilisation dans des boucles de contrôle industrielles linéaires, mais flexibles pour améliorer les performances et gérer les incertitudes modérées.

CHAPITRE II

COMMANDE PRÉDICTIVE BASÉE SUR LA LOGIQUE FLOUE POUR LES SYSTÈMES LINÉAIRES

II.1. Introduction

La commande prédictive à base de modèle (*MPC*) s'est imposée comme une pierre angulaire de la théorie moderne de la commande, reconnue pour sa capacité à gérer des systèmes multivariables explicitement soumis à des contraintes. Son principe fondamental, qui consiste à utiliser un modèle dynamique pour prédire le comportement futur du système et optimiser une séquence d'actions de commande, en fait un outil particulièrement puissant pour une vaste gamme d'applications industrielles et d'ingénierie [129], [133]. Cependant, les performances d'un régulateur MPC conventionnel dépendent fortement de la sélection minutieuse, souvent statique, des matrices de pondération Q et R dans sa fonction de coût. Ces matrices déterminent le compromis crucial entre la performance de suivi et l'effort de commande. Dans des scénarios pratiques caractérisés par des conditions de fonctionnement dynamiques, des perturbations imprévues ou des paramètres du système variant dans le temps, des pondérations fixes peuvent entraîner des performances sous-optimales, une faible adaptabilité et une robustesse réduite [79].

La synergie entre MPC et la logique floue (Fuzzy Logic) est examinée dans ce chapitre pour surmonter cette contrainte. Le contrôleur prédictif flou « Fuzzy Predictive Control » est une approche de commande pour les systèmes dynamique. L'utilisation d'un système d'inférence floue pour ajuster en temps réel les matrices de pondération de la MPC constitue la principale nouveauté. Cette méthode permet au régulateur de modifier indépendamment son agressivité et sa sensibilité en fonction d'indicateurs de performance en temps réel, tels que les erreurs de poursuite et les conditions du système.

Ce chapitre est structuré de la manière suivante. Il commence par l'introduction du modèle flou de *Takagi-Sugeno* (T-S), un cadre puissant pour la modélisation des systèmes non linéaires sous forme de modèles locaux linéaires fusionnés de façon cohérente à l'aide de fonctions d'appartenance floues. Sur cette base, le contrôleur prédictif flou (FPC) est formulé en incorporant une pondération adaptative directement dans le problème d'optimisation de la commande prédictive (MPC). Le système d'inférence floue y joue le rôle de superviseur de haut niveau, ajustant dynamiquement les paramètres pour assurer des performances optimales dans tous les régimes de fonctionnement. À cet égard, une étude de cas complète du contrôle de suivi de trajectoire d'un robot mobile à conduite différentielle non-holonomique démontre l'efficacité pratique et la supériorité de la technique FPC proposée. On a réalisées des simulations sous différentes trajectoires complexes, y compris des trajectoires en 8, circulaires et composites, pour comparer les performances du contrôleur amélioré (F2MPC) avec flou (*F1MPC*) et par rapport à un contrôleur MPC conventionnelle. Les résultats montrent que l'approche proposée améliore

la précision de suivi, le rejet des perturbations, le temps de stabilisation et l'économie d'énergie. En résumé, ce chapitre fournit un paradigme clair et complet pour la conception et la mise en œuvre du contrôleur prédictif flou. Il montre que l'adaptabilité de la logique floue et la capacité prédictive de la MPC produisent un paradigme de contrôle robuste, intelligent et efficace pour les systèmes complexes actuels.

II.2. MPC pour Les systèmes à paramètres variant linéairement (basé sur les modèles flous T–S)

La MPC est une technique puissante largement utilisée dans les applications de contrôle qui nécessitent la prise en compte des contraintes du système et des dynamiques multivariées. Dans ce chapitre, nous nous concentrons sur une classe de MPC formulée pour les systèmes à Paramètres Variant Linéairement (LPV), où la nature variant dans le temps du modèle est capturée à l'aide d'une approche floue Takagi–Sugeno (T–S).

II.2.1. Modèle flou Takagi–Sugeno pour les systèmes linéaires variant dans le temps

Les systèmes flous Takagi–Sugeno (T–S) sont largement utilisés pour représenter des systèmes non linéaires et variant dans le temps en interpolant un ensemble de modèles linéaires locaux à travers des mécanismes d'inférence floue. Ces systèmes sont particulièrement efficaces dans les applications de contrôle où la dynamique du système change en fonction des conditions de fonctionnement. Un système flou T–S général est composé de R règles, chacune définie comme suit:

$$R_i: \text{Si } x_1(k) \text{ est } P_1^k \text{ et } x_2(k) \text{ est } P_2^L \dots \text{ et } x_n(k) \text{ est } P_n^L, \text{ alors } x^i(k+1) = A_i x(k) + b_i u(k) \quad (\text{II.1})$$

Où:

- $x(k) = [x_1(k), x_2(k), \dots, x_n(k)]^T \in \mathbb{R}^n$ est l'état du système à l'instant discret k
- $u(k) \in \mathbb{R}^m$ est l'entrée de commande
- $A_i \in \mathbb{R}^{n \times n}, b_i \in \mathbb{R}^{n \times m}$ sont les matrices constantes associées à la règle i
- P_j^k sont des ensembles flous définis sur les univers de discours des états.

Contrairement aux systèmes nets, un système flou T–S combine les résultats de toutes les R règles en utilisant une moyenne pondérée déterminée par les degrés d'appartenance $\mu_i(k)$. Le système dynamique global résultant est décrit par l'équation suivante:

$$x(k+1) = \frac{\sum_{i=1}^R \mu_i(k) (A_i x(k) + b_i u(k))}{\sum_{i=1}^R \mu_i(k)} \quad (\text{II.2})$$

Ou sous forme compacte:

$$x(k + 1) = A(k)x(k) + b(k)u(k) \quad (II.3)$$

Où les matrices variant dans le temps $A(k)$ et $b(k)$ sont définies comme suit:

$$A(k) = \sum_{i=1}^R \xi_i(k)A_i, b(k) = \sum_{i=1}^R \xi_i(k)b_i \quad (II.4)$$

Et les fonctions de pondération floues normalisées $\xi_i(k)$ sont données par:

$$\xi_i(k) = \frac{\mu_i(k)}{\sum_{i=1}^R \mu_i(k)} \quad (II.5)$$

Étant donné que les valeurs d'appartenance $\mu_i(k)$ dépendent de l'état actuel $x(k)$, les matrices $A(k)$ et $b(k)$ varient au fil du temps. En réévaluant dynamiquement les poids des modèles locaux en fonction des conditions de fonctionnement en temps réel, cette représentation basée sur le flou permet au contrôle prédictif de s'adapter aux comportements non linéaires.

II.2.2. Modèle de prédiction

Dans le Model Predictive Control, le modèle de prédiction sert de base pour optimiser les entrées de commande sur un horizon temporel futur. Dans le contrôle prédictif flou, un mécanisme d'inférence floue T-S est utilisé pour déterminer la dynamique du système à chaque étape temporelle. À l'instant k , le mécanisme de fusion floue produit les matrices variant dans le temps $A(k)$ et $b(k)$ en fonction de l'état actuel du système et des valeurs de référence. Ces matrices sont ensuite fixées pour l'ensemble de l'horizon de prédiction afin de permettre une optimisation tractable. Le modèle d'état-espace en temps discret est donné par:

$$x(k + 1) = A(k)x(k) + b(k)u(k) \quad (II.6)$$

Où:

- $x(k) \in \mathbb{R}^n$ est le vecteur d'état,
- $u(k) \in \mathbb{R}^m$ est l'entrée de commande,
- $A(k) \in \mathbb{R}^{n \times n}$, $b(k) \in \mathbb{R}^{n \times m}$ sont les matrices du système obtenues par fusion floue à partir de modèles linéaires locaux.

Bien que $A(k)$ et $b(k)$ soient à l'origine variant dans le temps en raison de la logique floue, elles sont considérées comme constantes tout au long de l'horizon de prédiction N_p pendant chaque intervalle de commande. Cette hypothèse simplifie la structure de prédiction et permet l'application des formulations classiques de MPC linéaire.

- ✓ **Prédiction d'état récurrente:** En utilisant les matrices fixes $A(k)$ et $b(k)$, les états prédits sur l'horizon de prédiction sont calculés comme suit:

$$x(k+1) = A(k)x(k) + b(k)u(k)$$

$$x(k+2) = A(k)x(k+1) + b(k)u(k+1) = A(k)^2x(k) + A(k)b(k)u(k) + b(k)u(k+1)$$

$$x(k+3) = A(k)^3x(k) + A(k)^2b(k)u(k) + A(k)b(k)u(k+1) + b(k)u(k+2)$$

⋮

$$x(k+N_p) = A(k)^{N_p}x(k) + \sum_{i=0}^{N_p-1} A(k)^{N_p-1-i}b(k)u(k+i) \quad (II.7)$$

- ✓ **Formulation matricielle:** Nous définissons le vecteur d'état prédit et le vecteur d'entrée de commande comme suit:

$$\begin{cases} x_p = \begin{bmatrix} x(k+1) \\ x(k+2) \\ \vdots \\ x(k+N_p) \end{bmatrix} \in \mathbb{R}^{n \times N_p} \\ U_p = \begin{bmatrix} u(k) \\ u(k+1) \\ \vdots \\ u(k+N_p-1) \end{bmatrix} \in \mathbb{R}^{m \times N_p} \end{cases} \quad (II.8)$$

Ensuite, les prédictions d'état empilées peuvent être exprimées de manière compacte comme suit:

$$x_p = F(k)x(k) + H(k)U_p \quad (II.9)$$

Avec:

- $F(k) \in \mathbb{R}^{n \cdot N_p \times n}$, $F(k) = \begin{bmatrix} A(k) \\ A(k)^2 \\ \vdots \\ A(k)^{N_p} \end{bmatrix}$
- $H(k) \in \mathbb{R}^{n \cdot N_p \times m \cdot N_p}$, $H(k) = \begin{bmatrix} b(k) & 0 & \dots & 0 \\ A(k)b(k) & b(k) & \dots & 0 \\ \vdots & \vdots & \ddots & \vdots \\ A^{N_p-1}(k)b(k) & A^{N_p-2}(k)b(k) & \dots & b(k) \end{bmatrix}$

Cette formulation permet la conception de la fonction de coût et la gestion des contraintes dans le cadre d'optimisation standard de la MPC, tout en intégrant une adaptation basée sur le flou à chaque étape.

II.2.3. Fonction de coût et contraintes

Dans le Model Prédictive Control, la performance du contrôleur est déterminée en minimisant une fonction de coût qui pénalise à la fois l'erreur de suivi et l'effort de contrôle sur un horizon de prédiction N_p . **La fonction de coût quadratique standard** est donnée par:

$$j = \sum_{i=0}^{N_p-1} \left(\|x(k+i) - x_{ref}(k+i)\|_Q^2 + \|u(k+i)\|_R^2 \right) \quad (II.10)$$

Où: $x(k+i)$: état prédit à l'instant $k+i$, $x_{ref}(k+i)$: trajectoire de référence à l'instant $k+i$, $u(k+i)$: entrée de commande à l'instant $k+i$, $Q \in \mathbb{R}^{n \times n}$: matrice de pondération de l'état positive définie, $R \in \mathbb{R}^{m \times m}$: matrice de pondération de la commande positive définie.

Cette fonction de coût capture deux objectifs contradictoires:

- Maintenir les états prédits $x(k+i)$ proches de la trajectoire souhaitée $x_{ref}(k+i)$
- Minimiser l'effort de commande $u(k+i)$, ce qui évite des entrées de commande agressives ou coûteuses en énergie.

II.2.3.1. Adaptation basée sur le flou de $Q(k)$ et $R(k)$

Contrairement à la MPC traditionnelle où Q et R sont fixes, le *contrôle prédictif flou* utilise un système d'inférence floue pour calculer dynamiquement $Q(k)$ et $R(k)$ en fonction des caractéristiques instantanées du système. Cela permet une adaptation des gains en ligne, améliorant ainsi la performance et la robustesse dans des conditions variables. La structure deviant:

$$j = \sum_{i=0}^{N_p-1} \left(\|x(k+i) - x_{ref}(k+i)\|_{Q(k)}^2 + \|u(k+i)\|_{R(k)}^2 \right) \quad (II.11)$$

Où $Q(k)$ et $R(k)$ sont calculés à chaque instant k .

II.2.3.2. Contraintes strictes dans le MPC

Dans le cadre du MPC, des contraintes strictes sont utilisées pour garantir que le système fonctionne strictement dans les limites physiques ou opérationnelles prédéfinies. Ces contraintes représentent des frontières qui ne doivent en aucun cas être violées, qu'elles soient appliquées aux variables d'état ou aux entrées de commande. Des inégalités linéaires ou non linéaires sont généralement incluses dans le problème d'optimisation effectué à chaque étape de commande. Ces contraintes prennent généralement la forme suivante:

$$\begin{cases} x_{min} \leq x(k+i) \leq x_{max} \\ \text{et} \\ u_{min} \leq u(k+i) \leq u_{max} \end{cases} \quad (\text{II.12})$$

Où:

- $x(k+i)$ est l'état prédit du système à l'instant futur $k+i$.
- $u(k+i)$ est l'entrée de commande prédite du système à l'instant futur $k+i$.
- $x_{min}, x_{max}, u_{min}, u_{max}$ représentent les limites minimales et maximales autorisées pour les états et les entrées, respectivement.

Les contraintes strictes sont essentielles pour maintenir la sécurité et la fiabilité du système en empêchant la violation des limites critiques, ce qui pourrait autrement entraîner une instabilité, une dégradation des performances ou des dommages physiques aux composants du système.

II.2.4. Solution du problème d'optimisation

La base de l'approche MPC réside dans la résolution d'un problème d'optimisation à chaque instant d'échantillonnage. L'objectif est de déterminer une série d'entrées de commande futures qui minimisent une fonction de coût spécifiée, représentant la performance prévue du système sur un horizon de prédiction limité.

L'inférence floue met à jour les matrices de pondération variant dans le temps dans cette fonction de coût dans le cadre du **FPC**. Cela permet au contrôleur d'adapter son comportement en temps réel, en fonction de l'état du système ou des conditions de fonctionnement. La fonction de coût quadratique sur l'horizon de prédiction N_p est définie comme suit:

$$J(U_p) = (X_{ref} - X_p)^T Q(k) (X_{ref} - X_p) + U_p^T R(k) U_p \quad (\text{II.13})$$

Où:

- $X_{ref} \in \mathbb{R}^{n \times N_p}$: Vecteur d'état de référence empilé sur l'horizon de prédiction,
- $X_p \in \mathbb{R}^{n \times N_p}$: Vecteur d'état prédit obtenu à partir du modèle du système,
- $U_p \in \mathbb{R}^{m \times N_p}$: Vecteur empilé des entrées de commande futures,
- $Q(k) \in \mathbb{R}^{n N_p \times n N_p}$: c'est la matrice de pondération adaptée par flou pour l'erreur d'état,
- $R(k) \in \mathbb{R}^{m N_p \times m N_p}$: C'est la matrice de pondération adaptée par flou pour l'effort de contrôle.

La trajectoire d'état prédite X_p est générée en utilisant le modèle de prédiction linéaire:

$$X_p = HU_p + Fx(k) \quad (II.14)$$

En substituant cela dans la fonction de coût, on obtient une expression purement quadratique en U_p . La séquence optimale des entrées de commande est ensuite obtenue en minimisant la fonction de coût par rapport à U_p :

$$U_p^*(k) = (H^T Q(k)H + R(k))^{-1} H^T Q(k) (X_{ref} - Fx(k)) \quad (II.15)$$

Cette phrase offre la réponse analytique au problème d'optimisation non contraint et montre comment la logique floue affecte le comportement du contrôleur. Sur la base d'indicateurs de performance du système, les matrices de pondération $Q(k)$ et $R(k)$ sont ajustées à chaque pas de temps pour adapter le contrôleur à des conditions changeantes.

II.3. Réglage dynamique des poids de la commande prédictive (MPC)

Dans les systèmes de Commande Prédictive (MPC), les matrices de pondération Q et R dans la fonction de coût sont des éléments fondamentaux qui déterminent le compromis entre le suivi précis de la trajectoire de référence et la minimisation de l'effort de contrôle. Ces poids sont habituellement choisis hors ligne et fixés tout au long de l'opération en fonction de l'expérience du concepteur ou d'un ajustement par essais et erreurs. Des poids fixes peuvent ne pas bien fonctionner dans des systèmes dynamiques ou soumis à des perturbations inattendues.

II.3.1. Principe de l'ajustement dynamique

Pour améliorer l'adaptabilité, les matrices de pondération $Q(k)$ et $R(k)$ peuvent être rendues variant dans le temps, mises à jour à chaque instant d'échantillonnage en fonction des conditions actuelles du système. Dans cette approche, un Système d'Inférence Floue (SIF) de type **Sugeno** est utilisé pour déterminer les valeurs de pondération instantanées en fonction d'un ensemble d'indicateurs en temps réel $z(k)$:

$$z(k) = [Z_1(k), Z_2(k), \dots, Z_s(k)]^T \quad (II.16)$$

Des exemples de $z(k)$ incluent:

- ✓ L'état actuel $x(k)$,
- ✓ l'erreur de suivi $e(k) = x(k) - x_{ref}(k)$,
- ✓ Vitesse, accélération, angle de cap, distance relative, etc.

Pour simplifier, les matrices de pondération sont supposées diagonales:

$$Q(k) = \text{diag}(q_1(k), q_2(k), \dots, q_n(k)), R(k) = \text{diag}(r_1(k), r_2(k), \dots, r_m(k)) \quad (\text{II.17})$$

Avec $q_i(k) > 0$ et $r_j(k) > 0$ pour garantir la **définie positive**.

II.3.2. Procédure d'ajustement des poids basée sur le flou

A. Fonctions d'appartenance

Pour chaque variable d'entrée $Z_j(k)$, L_j fonctions d'appartenance sont définies:

$$\mu_j^\ell(Z_j(k)), \ell = 1, 2, \dots, L_j \quad (\text{II.18})$$

B. Base de règles

Pour chaque règle r parmi les N règles au total:

$$\text{SI } Z_1 \in A_1^{(r)} \text{ et } \dots \text{ et } Z_s \in A_s^{(r)} \text{ Alors } q^{(r)} = [q_1^{(r)}, q_2^{(r)}, \dots, q_n^{(r)}], r^{(r)} = [r_1^{(r)}, r_2^{(r)}, \dots, r_m^{(r)}] \quad (\text{II.19})$$

Où $q_i^{(r)} > 0$ et $r_j^{(r)} > 0$ sont les valeurs de poids assignées par cette règle.

C. Activation de la règle et agrégation

Le degré d'activation de la règle r est calculé comme suit:

$$\mu_r(k) = \prod_{j=1}^s \mu_j^{(A_j^{(r)})}(Z_j(k)) \quad , \mu_r(k) \in [0,1] \quad (\text{II.20})$$

Normalisation:

$$\bar{\mu}_r(k) = \frac{\mu_r(k)}{\sum_{v=1}^N \mu_v(k)} \quad (\text{II.21})$$

Défuzzification (moyenne pondérée): $q_i(k) = \sum_{r=1}^N \bar{\mu}_r(k) \cdot q_i^{(r)}$, $r_j(k) = \sum_{r=1}^N \bar{\mu}_r(k) \cdot r_j^{(r)}$.

Ainsi: $Q(k) = \text{diag}(q_1(k), q_2(k), \dots, q_n(k)), R(k) = \text{diag}(r_1(k), r_2(k), \dots, r_m(k))$

II.3.3. Intégration dans la formulation de la MPC

Les poids mis à jour sont incorporés dans la fonction de coût de la MPC:

$$j(k) = \sum_{i=0}^{N_p-1} \left(\|x(k+i) - x_{ref}(k+i)\|_{Q(k)}^2 + \|u(k+i)\|_{R(k)}^2 \right) \quad (\text{II.22})$$

Définissant:

$$\tilde{Q}(k) = \text{blkdiag}(Q(k), \dots, Q(k)) \quad , \quad \tilde{R}(k) = \text{blkdiag}(R(k), \dots, R(k))$$

La formulation du programmation quadratique (QP) devient:

$$J = \frac{1}{2} U_p^T \mathcal{H}(k) U_p + f(x)^T U_p \quad (\text{II.23})$$

Avec:

$$\mathcal{H}(k) = H(k)^T \tilde{Q}(k)H(k) + \tilde{R}(k) \quad (II.24)$$

$$\mathcal{f}(x) = H(k)^T \tilde{Q}(k)(F(k)x(k) + X_{ref}) \quad (II.25)$$

II.3.4. Impact sur les performances du contrôle

L'ajustement dynamique des poids de la MPC permet au contrôleur de:

- Augmenter la sensibilité au suivi lorsque de grandes erreurs ou des changements rapides se produisent.
- Réduire l'effort de commande dans des conditions quasi-stables, diminuant ainsi la consommation d'énergie et prolongeant la durée de vie des actionneurs.
- S'adapter automatiquement aux changements environnementaux soudains ou aux variations des caractéristiques du système, améliorant ainsi à la fois la réponse dynamique et la stabilité.

Le résultat est un contrôleur MPC plus intelligent et adaptatif, capable d'atteindre des performances et une efficacité supérieures par rapport aux conceptions à poids fixes.

II.4. Application: Commande prédictive floue d'un robot mobile

II.4.1. Description du système

Le système étudié est un Robot Mobile à Conduite Différentielle Non-Holonomique (DDMR), largement utilisé dans les applications robotiques en raison de sa simplicité et de son adéquation aux tâches de navigation et de suivi de trajectoire. Le robot est constitué de deux roues entraînées indépendamment et montées sur le même axe. En contrôlant les vitesses différentielles des deux roues, la vitesse linéaire v et la vitesse angulaire w du robot sont générées.

II.4.2. Modélisation cinématique d'un robot mobile non-holonomique à deux roues

Un robot mobile à conduite différentielle (deux roues sur un axe commun) est illustré à la Figure II.1 La position du robot dans le référentiel global (X, Y) est déterminée en utilisant un repère local attaché à son corps, défini par $p = [x, y, \theta]$. Ce type de robot possède une contrainte non-holonomique qui l'empêche de glisser latéralement, décrite par:

$$\dot{y} \cos \theta - \dot{x} \sin \theta = 0 \quad (II.26)$$

Le modèle de mouvement du robot à conduite différentielle prend en compte les vitesses linéaire et angulaire, en s'alignant sur la direction de guidage et en considérant les mouvements de rotation éventuels, formant un modèle cinématique pouvant être exprimé comme suit:

$$\dot{p} = \begin{bmatrix} \cos \theta & 0 \\ \sin \theta & 0 \\ 0 & 1 \end{bmatrix} \begin{bmatrix} v \\ w \end{bmatrix} \quad (II.27)$$

L'état du robot est défini par x, y, θ , avec v et w comme entrées de commande.

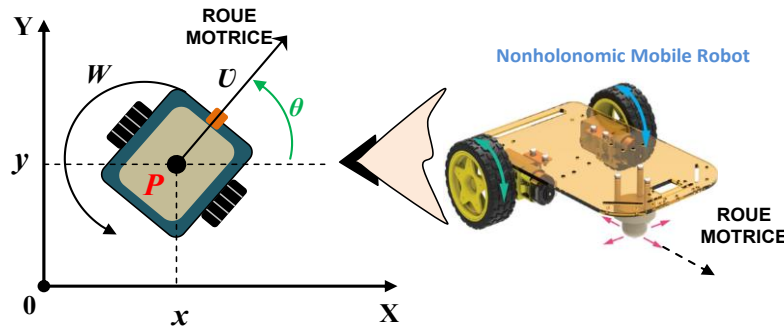


Figure II.1. Robot mobile à conduite différentielle.

II.4.3. Modèle d'erreur cinématique

Lors de l'élaboration d'une trajectoire pour qu'un robot mobile suive une trajectoire virtuelle avec une vitesse fixée, l'erreur de posture est la différence entre la position prévue du robot virtuel et la manière dont le robot physique est effectivement positionné et orienté, comme illustré à la Figure II.2.

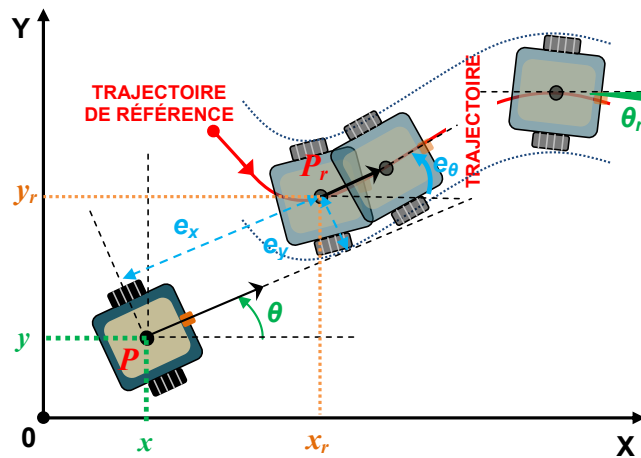


Figure II.2. Évolution de la transformation de l'erreur de suivi du robot.

La posture du robot de référence est représentée par le vecteur $p_r = [x_r \ y_r \ \theta_r]^T$, tandis que la posture réelle du robot est représentée par le vecteur $p = [x \ y \ \theta]^T$. La différence entre les postures est notée par le vecteur $e = [e_x \ e_y \ e_\theta]^T$, donnée par:

$$e = \begin{bmatrix} e_x \\ e_y \\ e_\theta \end{bmatrix} = \begin{bmatrix} \cos(\theta) & \sin(\theta) & 0 \\ -\sin(\theta) & \cos(\theta) & 0 \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} x_r - x \\ y_r - y \\ \theta_r - \theta \end{bmatrix} \quad (II.28)$$

La dynamique du robot mobile est définie par (II.27), et la dérivation de (II.28) conduit au modèle cinétique suivant:

$$\dot{e} = \begin{bmatrix} \cos(e_\theta) & 0 \\ \sin(e_\theta) & 0 \\ 0 & 1 \end{bmatrix} \begin{bmatrix} v_r \\ w_r \end{bmatrix} + \begin{bmatrix} -1 & e_y \\ 0 & -e_x \\ 0 & -1 \end{bmatrix} u \quad (II.29)$$

v_r et w_r désignent les vitesses du robot virtuel, servant respectivement d'entrées de commande feed-forward linéaires et angulaires. Elles peuvent s'exprimer comme suit:

$$\begin{cases} v_r = \pm \sqrt{\dot{x}_r^2 + \dot{y}_r^2} \\ w_r = (\dot{x}_r \dot{y}_r^2 - \dot{y}_r \dot{x}_r^2) / (\dot{x}_r^2 + \dot{y}_r^2) \end{cases} \quad (II.30)$$

u : représente l'entrée de commande dérivée de la combinaison des entrées de commande 'feedforward' et 'feedback' comme suit:

$$u = u_f + u_p = \begin{bmatrix} v_r \cos(e_\theta) \\ w_r \end{bmatrix} + \begin{bmatrix} v \\ w \end{bmatrix} \quad (II.31)$$

Où u_f représente l'entrée de commande feedforward et la sortie du contrôleur proposé. Nous dérivons le modèle d'erreur de suivi à partir des équations (II.29) et (II.31), c'est-à-dire:

$$\dot{e} = \begin{bmatrix} 0 & w & 0 \\ -w & 0 & 0 \\ 0 & 0 & 0 \end{bmatrix} e + \begin{bmatrix} 0 \\ \sin(e_\theta) \\ 0 \end{bmatrix} v_r + \begin{bmatrix} -1 & 0 \\ 0 & 0 \\ 0 & -1 \end{bmatrix} u_p \quad (II.32)$$

En linéarisant le modèle d'erreur (II.32) autour de la trajectoire de référence (où $e_x = e_y = e_\theta = 0$, $u_p = 0$), le modèle linéaire d'erreur cinématique suivant est obtenu:

$$\dot{e} = \begin{bmatrix} 0 & w_r & 0 \\ -w_r & 0 & v_r \\ 0 & 0 & 0 \end{bmatrix} e + \begin{bmatrix} -1 & 0 \\ 0 & 0 \\ 0 & -1 \end{bmatrix} u_p \quad (II.33)$$

L'équation (II.33) représente la représentation en espace d'état, garantissant la contrôlabilité tant que v_r ou w_r restent non nuls, nécessitant une procédure de contrôle robuste pour un guidage précis du robot.

II.4.4. Contrôle prédictif flou (CPF)

II.4.4.1. Modèle flou T-S

Les modèles flous T-S permettent de modéliser et de contrôler efficacement les systèmes dynamiques avec une grande précision et un nombre réduit de règles [134]. Ils permettent d'utiliser des contrôleurs linéaires localisés et d'obtenir une représentation précise du système. Plusieurs modèles linéaires d'erreur sont créés en fonction des vitesses de référence variables (v_r, w_r) dans un cadre en espace d'état, puis combinés dans le modèle flou T-S [135].

Dans un espace d'état discret, les systèmes flous T-S peuvent générer des systèmes dynamiques variables. Considérons le système de logique floue T-S avec R règles comme suit:

$$R_i: \text{SI } v_r(k) \text{ est } P_1^K \text{ et } w_r(k) \text{ est } P_2^L \text{ alors } e^i(k+1) = A_i e(k) + b_i u(k) \quad (\text{II.34})$$

Les vitesses du robot virtuel v_r et w_r varient avec le paramètre en temps discret ($k \in \{0, 1, 2, 3, \dots\}$). Les ensembles flous sont P_1^K, P_2^L , et $i = 1, 2, 3, \dots, R$, Le nombre de règles floues est R . Les matrices d'état et d'entrée sont A_i, b_i . L'équation aux différences en espace d'état est:

$$e(k+1) = (\sum_{i=1}^R \mu_i(k) (A_i e(k) + b_i u(k))) / (\sum_{i=1}^R \mu_i(k)) \quad (\text{II.35})$$

En considérant que $\zeta_i(k) = \mu_i(k) / \sum_{i=1}^R \mu_i(k)$ représente les fonctions de base floues pour la $i^{\text{ème}}$ règle floue, et en définissant $A(k) = \zeta_1 A_1 + \zeta_2 A_2 + \dots + \zeta_R A_R$ et $b(k) = \zeta_1 b_1 + \zeta_2 b_2 + \dots + \zeta_R b_R$, l'équation (II.35) peut être reformulée comme suit:

$$e(k+1) = A(k)e(k) + b(k)u(k) \quad (\text{II.36})$$

Ainsi, nous obtenons un modèle qui décrit le développement dynamique de $e(k)$'s et qui est identique au précédent. Les matrices $A(k)$ et $b(k)$ varient comme le font les fonctions de base floues.

II.4.4.2. Stratégie de contrôle

Le contrôle prédictif améliore le suivi de trajectoire en optimisant les entrées de commande sur un horizon temporel N_p . Il minimise les erreurs de suivi entre les trajectoires désirées et prédites en utilisant la fonction de coût quadratique suivante:

$$J(u_p, k) = \sum_{i=1}^{N_p} (e_r(k+i) - e(k+i))^T Q_f (e_r(k+i) - e(k+i)) + u_p^T(k, i) R_f u_p(k, i) \quad (\text{II.37})$$

Avec $Q_f \in \mathfrak{R}^n \times \mathfrak{R}^n, R_f \in \mathfrak{R}^m \times \mathfrak{R}^m, Q_f \geq 0$ et $R_f \geq 0$.

En considérant le cadre temporel modifié, la projection de sortie du modèle à l'instant N_p peut être décrite comme:

$$e(k + N_p) = \prod_{j=1}^{N_p-1} A(k + j)e(k) + \sum_{i=1}^{N_p} \left(\prod_{j=1}^{N_p-1} A(k + j) \right) \times B(k + i - 1)u_p(k + i - 1) + B(k + N_p - 1)u_p(k + N_p - 1) \quad (II.38)$$

Cette section décrit le vecteur d'erreur de prédiction et son efficacité pour montrer dans quelle mesure le robot suit la trajectoire:

$$E_p(k) = [e(k + 1)^T \quad e(k + 2)^T \quad \dots \quad e(k + N_p)^T]^T \quad (II.39)$$

De plus,

$$U_p(k) = [u_p^T(k)^T \quad u_p^T(k + 1)^T \quad \dots \quad u_p^T(k + N_p - 1)^T]^T \quad (II.40)$$

Après prise en compte de cela,

$$G(k) = [A(k) \quad A^2(k) \quad \dots \quad A^{N_p}(k)]^T \quad (II.41)$$

Et

$$H(k) = \begin{bmatrix} B(k) & 0 & \dots & 0 \\ A(k)B(k) & B(k) & \dots & 0 \\ \vdots & \vdots & \ddots & \vdots \\ A^{N_p}(k)B(k) & A^{N_p-1}(k)B(k) & \dots & B(k) \end{bmatrix} \quad (II.42)$$

Le vecteur des erreurs de prédiction pour le suivi du robot s'exprime comme suit:

$$E_p(k) = G(k)e(k) + H(k)U_p(k) \quad (II.43)$$

Avec $G \in \mathfrak{R}^{n.N_p} \times \mathfrak{R}^n$ et $H \in \mathfrak{R}^{n.N_p} \times \mathfrak{R}^{m.N_p}$.

La détermination du prochain point de référence est nécessaire pour le suivi de trajectoire. Un contrôle précis est difficile sans cette information. Les erreurs de commande futures doivent diminuer en fonction de la dynamique du modèle de référence (A_r). Une trajectoire de suivi de l'erreur de référence est choisie et représentée en espace d'état pour $i = 1, \dots, N_p$ comme suit:

$$e_r(k + i) = A_r^i e(k) \quad (II.44)$$

Supposons que la représentation vectorielle de la référence du robot, qui délimite l'erreur de suivi, soit exprimée comme:

$$E_p^r = [e_r(k+1)^T \quad e_r(k+2)^T \quad \dots \quad e_r(k+N_p)^T]^T \quad (II.45)$$

Le vecteur d'erreur de suivi de référence du robot est calculé en utilisant les Éqs. (II.44) et (II.45) comme suit:

$$E_p^r(k) = G_r e(k) \quad (II.46)$$

Dans ce cas:

$$G_r = [A_r \quad A_r^2 \quad \dots \quad A_r^{N_p}]^T \quad (II.47)$$

Où $E_p^r \in \mathfrak{R}^{n.N_p}$ représente l'ensemble de la période d'observation N_p , tandis que $G_r \in \mathfrak{R}^{n.N_p} \times \mathfrak{R}^n$.

Le cadre de Contrôle Prédictif du Modèle utilise la fonction de coût pour sélectionner les entrées de commande (II.37) et peut être réécrit comme suit:

$$J(U_p) = (E_p^r - E_p)^T \widetilde{Q}_f (E_p^r - E_p) + U_p^T \widetilde{R}_f U_p \quad (II.48)$$

L'obtention de la loi de commande appropriée implique la minimisation de la fonction de coût comme suit:

$$\frac{\partial J}{\partial U_p} = -2H^T \widetilde{Q}_f E_p^r + 2H^T \widetilde{Q}_f E_p + 2\widetilde{R}_f U_p \quad (II.49)$$

Par conséquent, le vecteur de commande peut être déterminé en utilisant les éqs. (II.43), (II.46) et (II.49) comme suit:

$$U_p(k) = (H^T \widetilde{Q}_f H + \widetilde{R}_f)^{-1} H^T \widetilde{Q}_f (G_r - G) e(k) \quad (II.50)$$

Q_f et R_f are sont les matrices de pondération utilisées pour formuler la fonction objective dans le problème d'optimisation de la MPC.

La technique proposée améliore le suivi de trajectoire et optimise la réponse du contrôleur MPC face aux variations du système. L'ajustement dynamique de la matrice de pondération permet au contrôleur de s'adapter à la dynamique du système et de suivre avec précision les trajectoires souhaitées. Le système de logique flou, applique d'abord la logique floue de type 1 (Figure 5) pour affiner le contrôle en fonction de la vitesse linéaire v_r du robot virtuel et de la distance actuelle D_{ed} entre le robot mobile et le robot virtuel. Ensuite, la logique floue de type 2 est utilisée (figures 6), et une comparaison entre les deux types est réalisée. Avec $Q_f = \text{diag}(q_1, q_2, q_3)$, $R_f = \text{diag}(r_1, r_2)$.

$$\bar{Q}_f = \begin{bmatrix} Q_f & 0 & \dots & 0 \\ 0 & Q_f & \ddots & 0 \\ \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & \dots & Q_f \end{bmatrix} \text{ and } \bar{R}_f = \begin{bmatrix} R_f & 0 & \dots & 0 \\ 0 & R_f & \ddots & 0 \\ \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & \dots & R_f \end{bmatrix} \quad (II.51)$$

Cela indique que $\bar{Q}_f \in \mathfrak{R}^{n.N_p} \times \mathfrak{R}^{n.N_p}$ et $\bar{R}_f \in \mathfrak{R}^{m.N_p} \times \mathfrak{R}^{m.N_p}$

En utilisant (II.50), nous pouvons maintenant exprimer la loi de commande en retour pour le MPC avec l'expression suivante:

$$u_p(k) = K_{mpc} \cdot e(k) \quad (II.52)$$

K_{mpc} est défini comme les m premières lignes de la matrice $\left[(H^T \bar{Q}_f H + \bar{R}_f)^{-1} H^T \bar{Q}_f (G_r - G) \right]$. $K_{mpc} \in \mathfrak{R}^m \times \mathfrak{R}^n$. La Figure II.3. illustre le mécanisme du contrôleur proposé. [149]

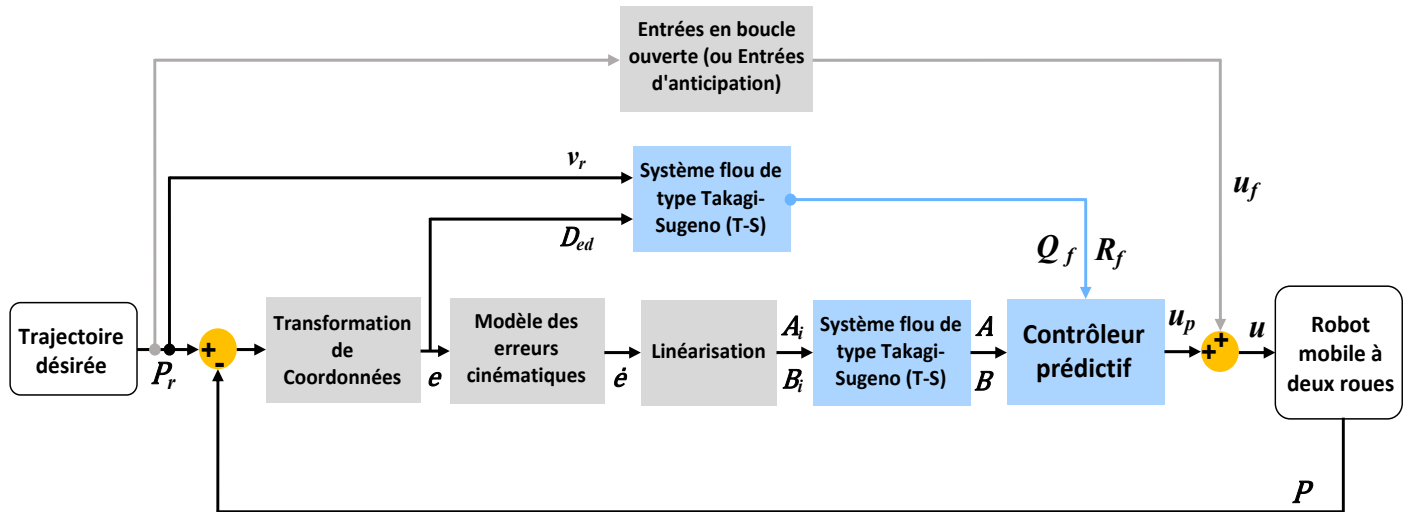


Figure II.3. Schéma bloc de l'approche proposée.

II.4.5. Preuve de stabilité

En partant de l'équation du modèle linéaire d'erreur (II.36) et en appliquant la loi de commande en rétroaction (II.52): $u_p(k) = K_{mpc} \cdot e(k)$. le système en boucle fermée devient:

$$e(k+1) = (A(k) + b(k)K_{mpc}(k))e(k) \quad (II.53)$$

L'objectif est de faire comprendre que:

$$\lim_{k \rightarrow \infty} e(k) = 0 \quad (II.54)$$

Ce qui signifie que le système en boucle fermée est stable au sens de Lyapunov.

Comme stratégie pour considérer la fonction de coût comme un **candidat de Lyapunov**, nous reformulons la fonction de coût introduite dans l'équation (II.48). Dans le cas de stabilité, où la référence est constante $E_p^r = 0$, cela se simplifie:

$$J(U_p) = E_p^T \widetilde{Q}_f E_p + U_p^T \widetilde{R}_f U_p \quad (II.55)$$

Puisque: $\widetilde{Q}_f > 0$ définie positive, et $\widetilde{R}_f > 0$ définie positive, il s'ensuit que:

$$J(U_p) > 0$$

De plus, puisque la loi de commande optimale MPC minimise cette fonction à chaque étape temporelle, nous avons que: **Jest définie positive**, et elle décroît de manière monotone dans le temps grâce à l'optimisation répétée.

Puisque Jest définie positive et strictement décroissante, elle sert de fonction candidate de Lyapunov valide pour le système en boucle fermée. Par conséquent, l'erreur de suivi $e(k)$ converge vers zéro lorsque $t \rightarrow \infty$, **démontrant la stabilité du système**.

II.4.6. Résultats et discussion

Pour valider l'efficacité de la stratégie de contrôle prédictif flou proposée, des simulations MATLAB ont été réalisées sur un robot mobile non-holonomique à conduite différentielle dans plusieurs scénarios de suivi de trajectoire. Les simulations ont été effectuées sous les contraintes suivantes:

- *Vitesse linéaire:* $u_1 \leq 1.5$ m/s
- *Vitesse angulaire:* $-10 \leq u_2 \leq 10$ rad/s
- *Vitesses de référence:* $v_r(t) \in [0, 1.5]$ m/s, $w_r(t) \in [-10, 10]$ rad/s
- *Horizon de prédiction:* $N_p = 4$

En se basant sur les valeurs extrêmes des vitesses de référence (v_r, w_r), quatre sous-systèmes linéarisés (A_1 – A_4, b_1 – b_4) ont été dérivés:

$$A_1 = \begin{bmatrix} 0 & -10 & 0 \\ 10 & 0 & 0 \\ 0 & 0 & 0 \end{bmatrix}, b_1 = \begin{bmatrix} -1 & 0 \\ 0 & 0 \\ 0 & -1 \end{bmatrix}, A_2 = \begin{bmatrix} 0 & 10 & 0 \\ -10 & 0 & 0 \\ 0 & 0 & 0 \end{bmatrix}, b_2 = \begin{bmatrix} -1 & 0 \\ 0 & 0 \\ 0 & -1 \end{bmatrix}, \quad (II.56)$$

$$A_3 = \begin{bmatrix} 0 & -10 & 0 \\ 10 & 0 & 1.5 \\ 0 & 0 & 0 \end{bmatrix}, b_3 = \begin{bmatrix} -1 & 0 \\ 0 & 0 \\ 0 & -1 \end{bmatrix}, A_4 = \begin{bmatrix} 0 & 10 & 0 \\ -10 & 0 & 1.5 \\ 0 & 0 & 0 \end{bmatrix}, b_4 = \begin{bmatrix} -1 & 0 \\ 0 & 0 \\ 0 & -1 \end{bmatrix}$$

Le modèle flou Takagi-Sugeno (T-S) utilise les fonctions d'appartenance montrées à la Figure II.4 pour ajuster dynamiquement le comportement du système en fonction des entrées de vitesse de référence. La base de règles se compose de quatre règles floues qui associent les conditions de vitesse aux sous-systèmes linéarisés correspondants:

- R_1 : si $v_r(k)$ est petite et $w_r(k)$ est négative alors $e^1(k+1) = A_1e(k) + b_1u(k)$
- R_2 : si $v_r(k)$ est petite et $w_r(k)$ est positive alors $e^2(k+1) = A_2e(k) + b_2u(k)$
- R_3 : si $v_r(k)$ est grande et $w_r(k)$ est négative alors $e^3(k+1) = A_3e(k) + b_3u(k)$
- R_4 : si $v_r(k)$ est grande et $w_r(k)$ est positive alors $e^4(k+1) = A_4e(k) + b_4u(k)$

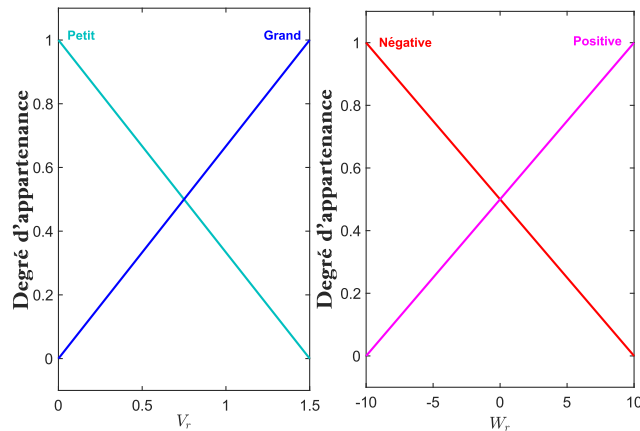


Figure II.4. Caractérisation des fonctions d'appartenance pour le modèle flou T-S.

Un système de logique floue de type 1 et type 2 est utilisé dans les deux scénarios de simulation pour déterminer précisément les matrices de pondération pour le contrôleur prédictif. Le système flou prend en entrée la vitesse linéaire v_r et la distance actuelle D_{ed} entre le mobile réel et le robot virtuelle. Ses entrées sont utilisées pour adapter les gains de contrôle selon la position de robot et son état de suivi, plus précisément:

- La distance D_{ed} est divisée en trois catégories floues: **Zéro**, **Moyen** et **Grand**, couvrant des valeurs de 0 à 0,3 mètres.
- La vitesse varie de 0,2 à 1,1 m/s et est divisée en deux catégories floues: Petite et Grande.

Le système d'inférence flou génère en sortie cinq paramètres:

Trois valeurs (q_1, q_2, q_3) pour ajuster la matrice de pondération d'état Q_f

Deux valeurs (r_1, r_2) pour ajuster la matrice d'effort de contrôle R_f .

La Fig. 5 (logique floue de type 1) et la Fig. 6 (logique floue de type 2) illustrent les fonctions d'appartenance des variables d'entrée, tandis que les Tableaux 1 et 2 présentent les règles floues qui associent ces entrées à des valeurs de sortie spécifiques pour les matrices de pondération.

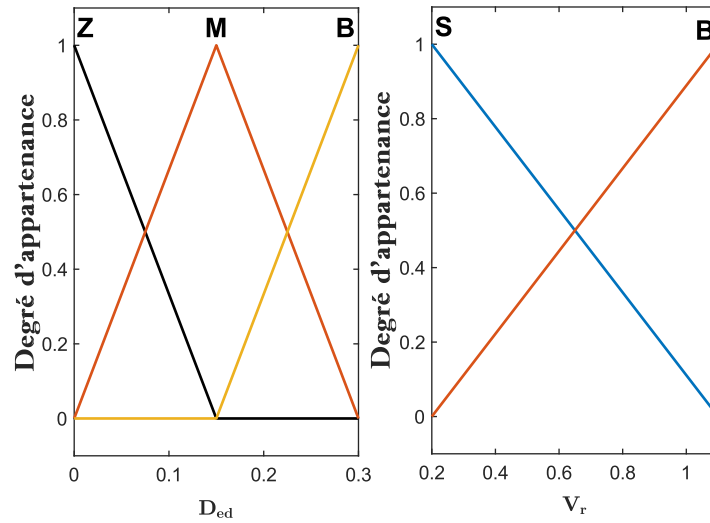


Figure II.5. Fonctions d'appartenance des entrées du système flou T-S de type-1.

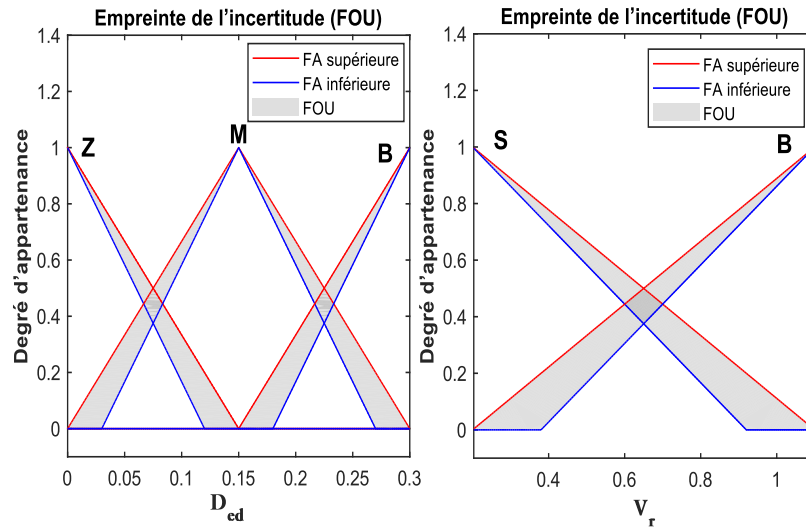


Figure II.6. Fonctions d'appartenance des entrées du système flou T-S de type-2.

Tableau II.1. Règles floues pour Q et R

Règles	D_{ed}	v_r	q_1	q_2	q_3	r_1	r_2
R_1	Zéro	Petite	faible	faible	Moyenne	faible	faible
R_2	Zéro	Grande	très faible	faible	élevée	faible	faible
R_3	Moyenne	Petite	Moyenne	Moyenne	faible	Moyenne	Moyenne
R_4	Moyenne	Grande	Moyenne	Moyenne	faible	Moyenne	Moyenne
R_5	Grande	Petite	élevée	élevée	très faible	élevée	élevée
R_6	Grande	Grande	élevée	élevée	très faible	élevée	élevée

Tableau II.2. Valeurs des sorties linguistiques.

Sorties linguistiques	q_1	q_2	q_3	r_1	r_2
Très faible	1	/	0.01	/	/
Faible	2	80	0.03	0.0001	0.0001
Moyenne	5	90	0.05	0.001	0.001
Élevée	10	100	2	0.01	0.01

II.4.6.1. Suivi d'une trajectoire en 8

Dans cette section, une comparaison est réalisée entre le contrôle prédictif flou de Type-1 proposé (F1PC) et la méthode de Model Prédictive Control (MPC) rapportée dans [111], [135]. La tâche consiste à guider le robot le long d'une trajectoire en 8, définie par l'Équation (II.57), sous des vitesses linéaires et angulaires variant dans le temps. La posture initiale du robot est fixée à $p(0) = [1.1 \quad 0.8 \quad \frac{\pi}{4}]$

Les matrices de pondération de la loi de commande proposée sont définies comme suit:

$$Q = \begin{bmatrix} 1 & 0 & 0 \\ 0 & 70 & 0 \\ 0 & 0 & 0.2 \end{bmatrix}, \quad R = \begin{bmatrix} 0.001 & 0 \\ 0 & 0.0001 \end{bmatrix}$$

$$\begin{cases} x_r(t) = 1.1 + 0.7 \sin\left(\frac{2\pi t}{30}\right) \\ y_r(t) = 0.9 + 0.7 \sin\left(\frac{4\pi t}{30}\right) \end{cases} \quad (\text{II.57})$$

La Figure II.6 présente une comparaison des performances de suivi de trajectoire du robot mobile sur une trajectoire en forme de 8 sous les stratégies MPC et Fuzzy-MPC de Type-1 (F1MPC). Les résultats

montrent une convergence stable et un suivi de trajectoire précis dans les deux cas, la stratégie F1MPC offrant un suivi plus précis que la MPC.

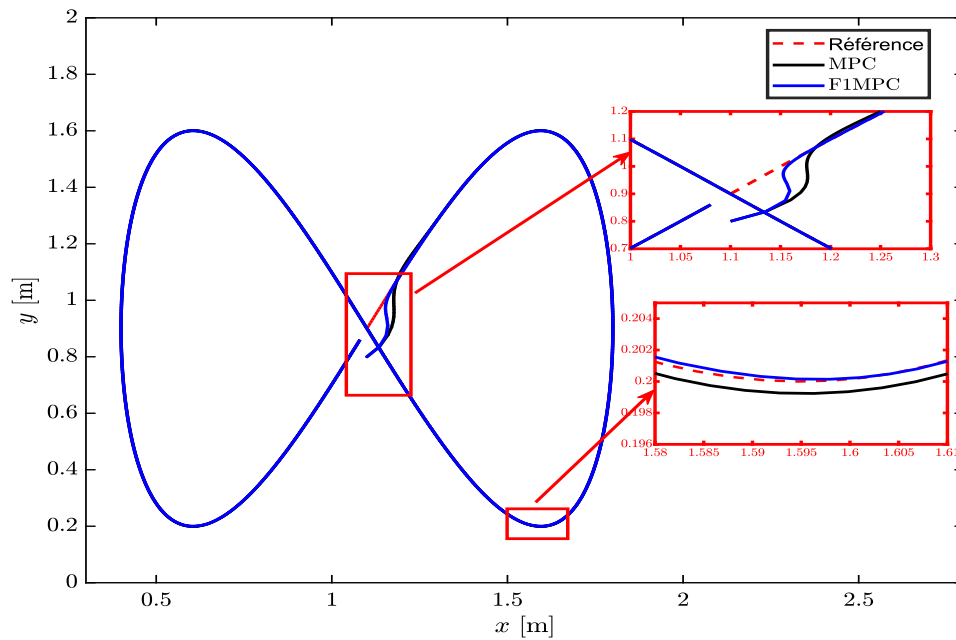


Figure II.7. Réponse du robot sous F1MPC et MPC sur une trajectoire en forme de 8.

La Figure II.7 illustre les courbes de l'erreur de suivi du contrôleur MPC pendant la trajectoire en forme de 8. Les résultats montrent que, bien que la MPC soit capable de stabiliser le système et de réduire les erreurs au fil du temps, elle présente des oscillations notables, en particulier sur l'erreur d'orientation θ . Ce comportement met en évidence *une limitation de sa capacité d'amortissement*, car le contrôleur nécessite plus de temps pour supprimer les oscillations et atteindre une convergence fluide.

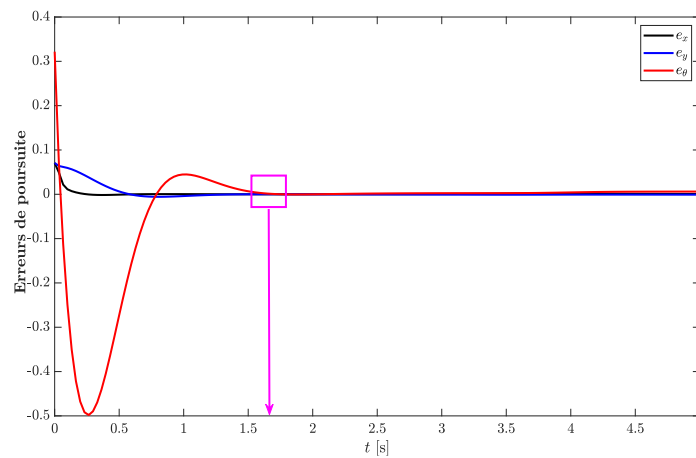


Figure II.8. Erreurs de suivi utilisant l'approche MPC.

La Figure II.8 présente les performances de l'erreur de suivi du contrôleur F1MPC sur la même trajectoire. Comparé à la MPC, le F1MPC montre une convergence des erreurs plus rapide avec des oscillations réduites à la fois sur la position e_x , e_y et sur l'orientation e_θ . Malgré une surtension transitoire brève, le contrôleur démontre une capacité d'amortissement plus forte et atteint un suivi stable plus rapidement, confirmant sa supériorité par rapport à la MPC conventionnelle dans les tâches de suivi de trajectoire en forme de 8.

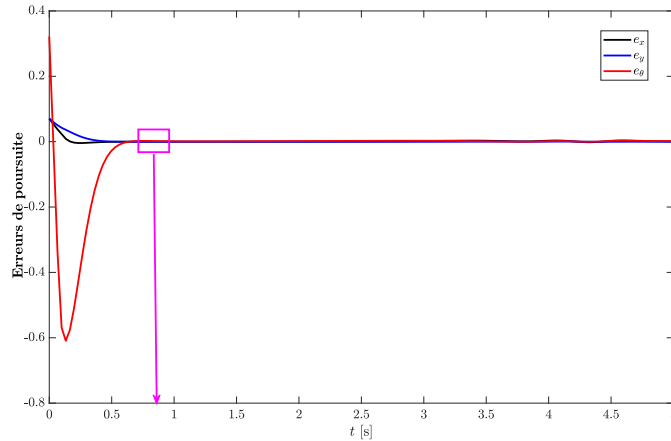


Figure II.9. Erreurs de suivi utilisant l'approche F1MPC.

La Figure II.9 illustre les variations des vitesses linéaire et angulaire du robot sous Model Predictive Control (MPC). Les courbes montrent que la vitesse linéaire varie approximativement entre 0 et 1,5 m/s, tandis que la vitesse angulaire oscille dans une plage d'environ -10 à $+10$ rad/s.

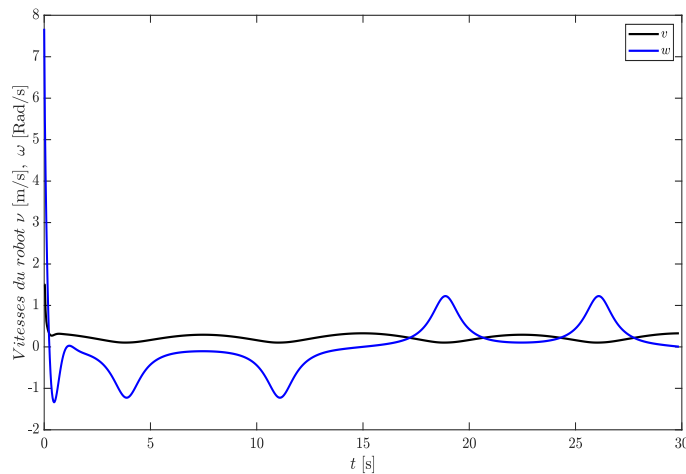


Figure II.10. Vitesses linéaire et angulaire utilisant l'approche MPC.

La Figure II.10 présente la réponse des vitesses linéaire et angulaire du robot en utilisant le contrôle prédictif flou de Type-1 (F1MPC). On peut observer que la vitesse linéaire reste approximativement dans la même plage (0 à 1,5 m/s), ainsi que la vitesse angulaire (-10 à +10 rad/s).

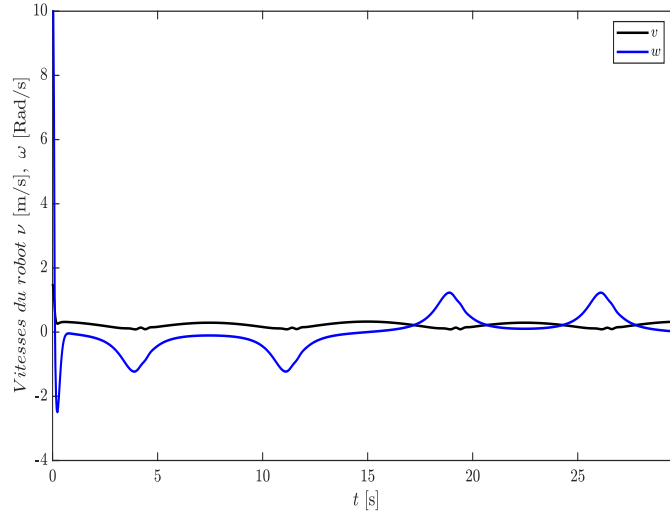


Figure II.11. Vitesses linéaire et angulaire utilisant l’approche F1MPC.

D’après l’analyse des données d’erreur de suivi présentées dans le Tableau II.3, il est évident que les deux contrôleurs, F1MPC et MPC, ont globalement obtenu de bonnes performances pour suivre la trajectoire désirée. La MPC a légèrement mieux performé pour minimiser l’erreur de la variable e_x (ISE = 0,0002 contre 0,0003 pour F1MPC), tandis que le F1MPC a clairement surpassé la MPC pour réduire l’erreur de la variable e_y (ISE = 0,0005 contre 0,0010 pour MPC). L’avantage le plus décisif a cependant été observé sur la variable e_θ , où le F1MPC a atteint une erreur près de 20 % plus faible (ISE = 0,0635 contre 0,0809 pour MPC). Cela fait du F1MPC le choix privilégié.

Tableau II.3. Analyse des erreurs de suivi.

	ISE(e_x)	ISE(e_y)	ISE(e_θ)
F1MPC	0.0003	0.0005	0.0635
MPC	0.0002	0.0010	0.0809

II.4.6.2. Suivi d’une trajectoire circulaire

Dans ce scénario, le robot mobile doit suivre une trajectoire circulaire définie mathématiquement par l’Équation (II.58):

$$\begin{cases} x_r(t) = 1.0 + \cos\left(\frac{2\pi t}{30}\right) \\ y_r(t) = 1.0 + \sin\left(\frac{2\pi t}{30}\right) \end{cases} \quad (II.58)$$

Le robot de référence suit cette trajectoire avec une vitesse linéaire constante $v_r = 0.2$ m/s et une vitesse angulaire $w_r = 0.2$ rad/s. Son état initial est défini comme $p(0) = [2.1 \ 1.0 \ 2.0]$.

Cette configuration fournit une trajectoire lisse et périodique, adaptée pour évaluer les performances de suivi de base de la stratégie de contrôle proposée dans des conditions idéales.

La Fig. II.12 illustre la performance de suivi de trajectoire du robot mobile sous les stratégies MPC, MPC flou de type 1 (F1MPC) et MPC flou de type 2 (F2MPC). Les résultats montrent une convergence robuste et une adhérence précise à la trajectoire circulaire souhaitée, les méthodes améliorées par la logique floue offrant une meilleure précision de suivi.

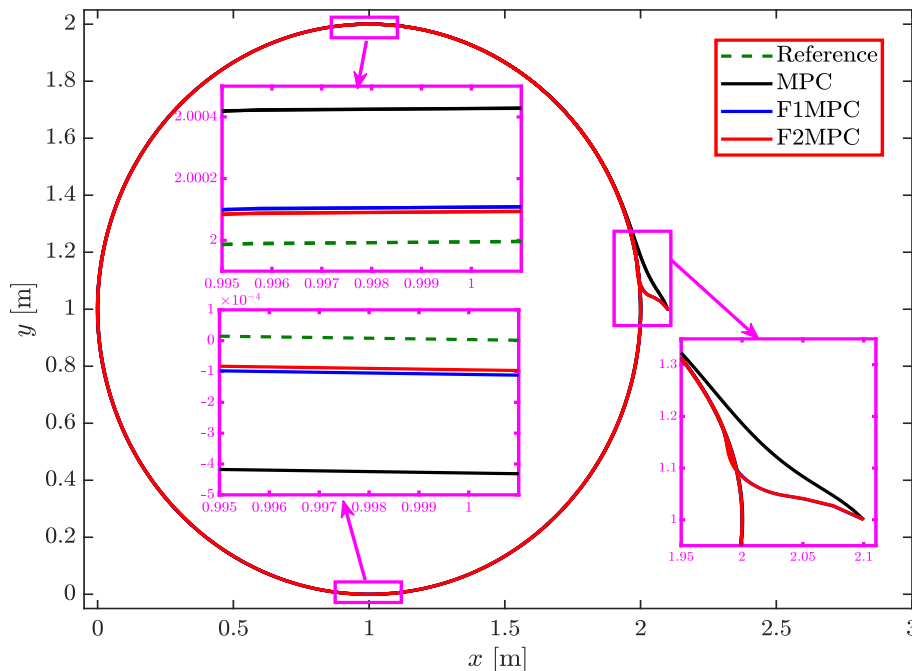


Figure II.12. Résultats du suivi de trajectoire circulaire en utilisant les méthodes de contrôle prédictif et les méthodes proposées de contrôle prédictif par modèle flou (F1MPC et F2MPC).

Les Figures II.13, 14 et 15 présentent les courbes d'erreur de suivi pour les contrôleurs MPC, F1MPC et F2MPC lors du suivi d'une trajectoire circulaire. La Figure II.13 montre que le contrôleur MPC conventionnel maintient une stabilité relative mais présente des oscillations notables, en particulier dans l'erreur d'orientation e_θ , ce qui indique une capacité d'amortissement limitée. En revanche, la Figure II.14 montre des performances améliorées du contrôleur F1MPC, qui, bien qu'il subisse des erreurs transitoires légèrement plus élevées (atteignant jusqu'à -1,2), parvient à revenir rapidement à la

stabilité. La Figure II.15 confirme encore la performance supérieure du contrôleur F2MPC, qui obtient des erreurs plus faibles et de plus courte durée, mettant en évidence l'efficacité de la structure de contrôle flou de type 2 pour fournir un suivi de trajectoire plus fluide et plus stable par rapport aux autres contrôleurs.

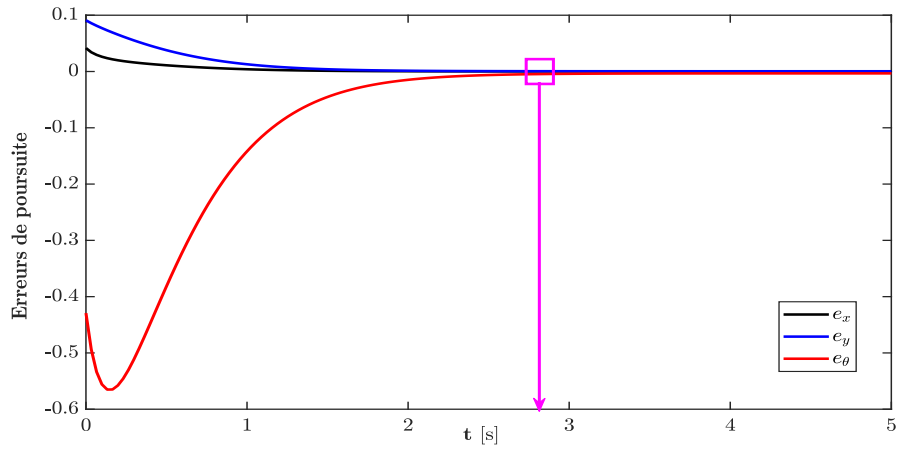


Figure II.13. Erreurs de suivi utilisant l'approche MPC.

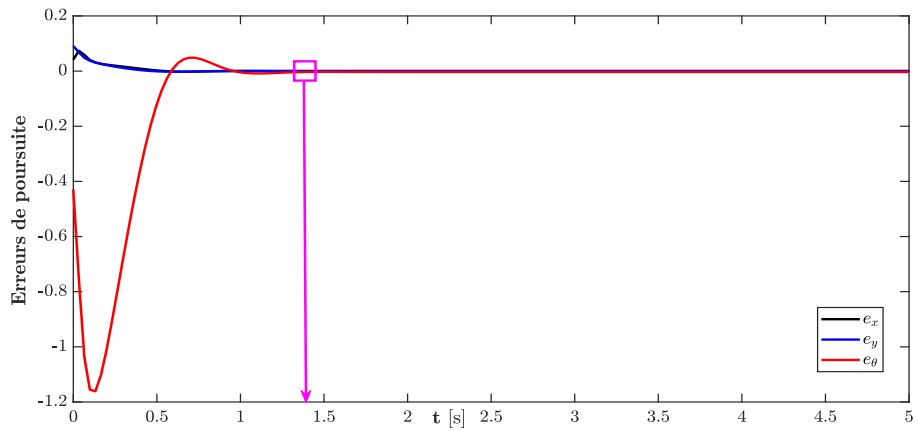


Figure II.14. Erreurs de suivi utilisant l'approche F1MPC.

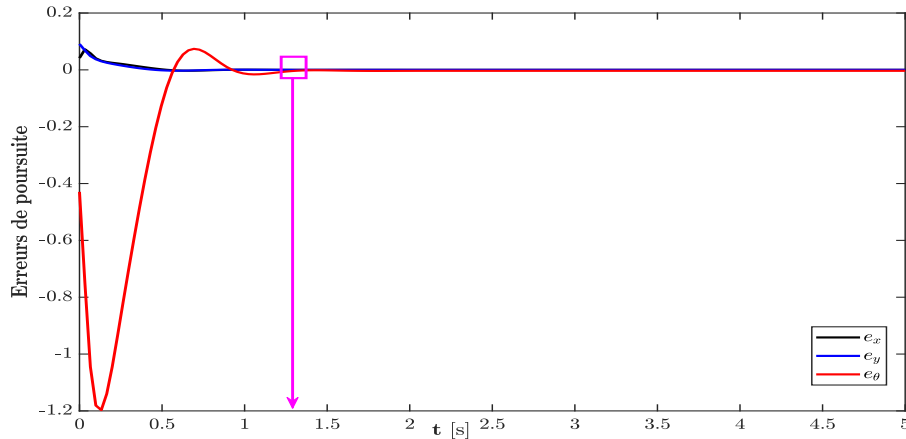


Figure II.15. Erreurs de suivi utilisant l'approche F2MPC.

Les Figures II.16 à II.18 comparent les entrées de contrôle de la vitesse linéaire v_r et de la vitesse angulaire w_r générées par le MPC, le MPC flou de type 1 (F1MPC) et le MPC flou de type 2 (F2MPC) lors du suivi de trajectoire circulaire. Les résultats révèlent des comportements de contrôle distincts :

MPC (Fig. II.16) : La vitesse linéaire v_r chute brusquement de 0,5 à 0,2 m/s en 0,1 seconde avant de se stabiliser. Sur la même période de 0,5 seconde, la vitesse angulaire w_r oscille de 2,3 rad/s à -0,5 rad/s avant d'atteindre un état stable proche de 0,2 rad/s après 2,3 secondes. F1MPC (Fig. II.17) : La vitesse linéaire v_r est plus variable, commençant à 0 m/s, atteignant un maximum de 1,1 m/s, puis revenant à environ 0,2 m/s en 0,15 seconde. La vitesse angulaire w_r présente une large plage dynamique, variant de 10 rad/s à -4 rad/s en seulement 0,3 seconde, et se stabilise en 1,2 seconde. Le contrôleur F2MPC (Fig. II.18) présente un schéma comportemental similaire à celui du F1MPC, mais se stabilise sur une période plus courte (<1,2 secondes). Cette conception présente un dépassement transitoire significativement plus élevé que celui du F1MPC avant d'atteindre l'équilibre.

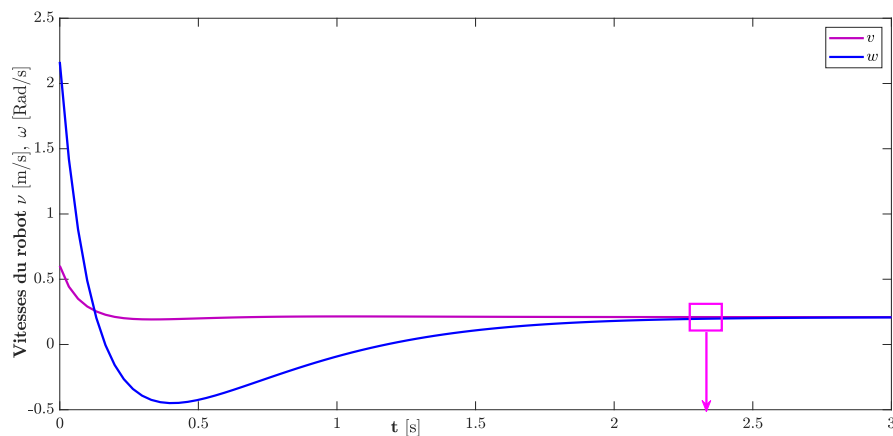


Figure II.16. Vitesses linéaire et angulaire utilisant l'approche MPC.

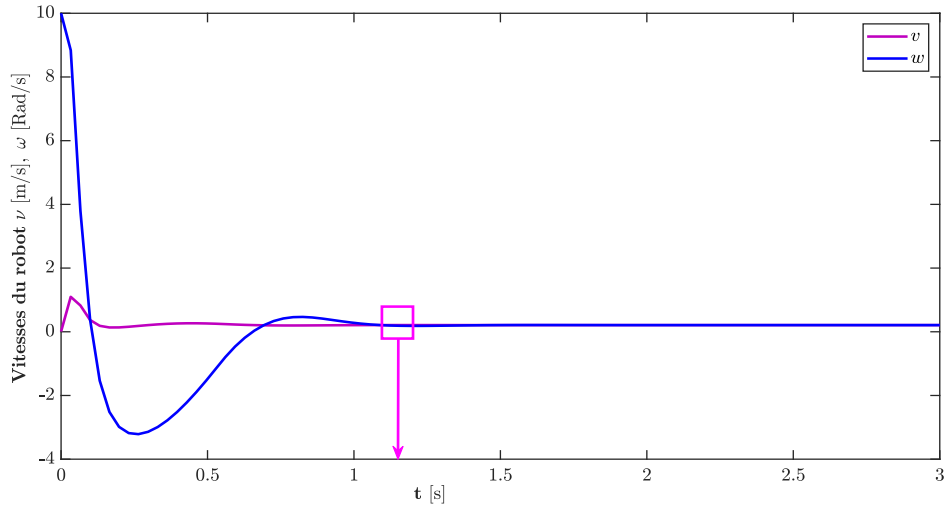


Figure II.17. Vitesses linéaire et angulaire utilisant l'approche F1MPC.

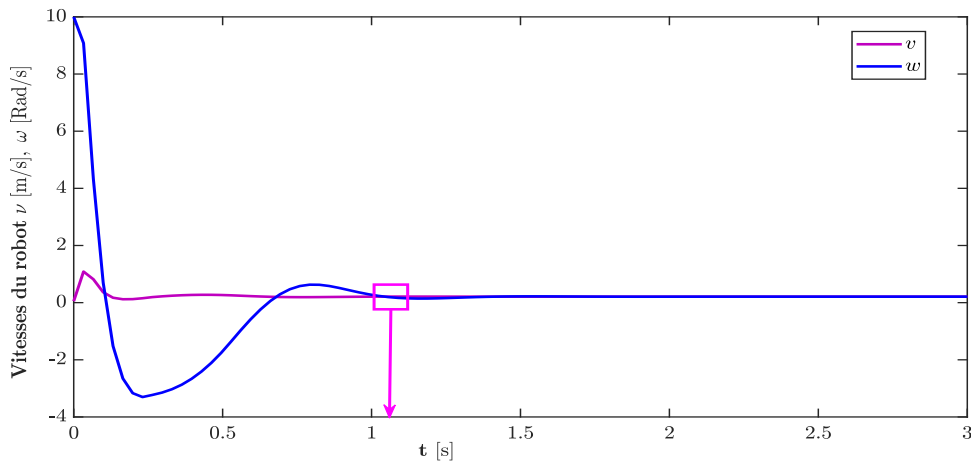


Figure II.18. Vitesses linéaire et angulaire utilisant l'approche F2MPC.

II.4.6.3. Suivi de trajectoires circulaire et sinusoïdale

Cette étude examine un scénario dans lequel un robot mobile a pour mission de suivre une trajectoire prédéfinie. L'objectif principal est de guider le robot mobile avec précision le long d'une trajectoire dynamique caractérisée par des vitesses linéaires et angulaires variables. La trajectoire se compose d'un segment circulaire initial, défini par l'équation (47), suivi d'un segment sinusoïdal représenté mathématiquement par l'équation (II.59). Le mouvement du robot commence à partir de $dep(0) = [1.9 \ 0.9 \ \pi/3]$.

$$\begin{cases} x_r(t) = 1.0 + \cos\left(\frac{2\pi t}{30}\right) \\ y_r(t) = 1.0 + \sin\left(\frac{2\pi t}{30}\right) \end{cases} \quad (II.59)$$

Cette trajectoire composite évalue la capacité du contrôleur à s'adapter à des trajectoires dynamiques et variant dans le temps: le segment circulaire mesure la cohérence, tandis que le segment sinusoïdal teste la réactivité face à des transitions plus exigeantes.

Dans cette section, les résultats des systèmes de contrôle prédictif flou proposés (F1MPC) et (F2MPC) sont comparés à ceux obtenus avec le contrôle prédictif par modèle (MPC) développé dans [135].

La Figure II.19 illustre la trajectoire suivie par le robot mobile sous trois scénarios de contrôle distincts: la première technique proposée (F1MPC), la deuxième technique proposée (F2MPC) et la technique de contrôle traditionnelle (MPC). Cette figure permet une comparaison directe de la réponse du robot sous différents commandements de contrôle, mettant en évidence les variations dans la précision de suivi et la fluidité du mouvement. Grâce à cette représentation visuelle, l'efficacité des deux techniques proposées (F1MPC et F2MPC) pour atteindre la trajectoire souhaitée peut être directement évaluée par rapport à l'approche traditionnelle MPC.

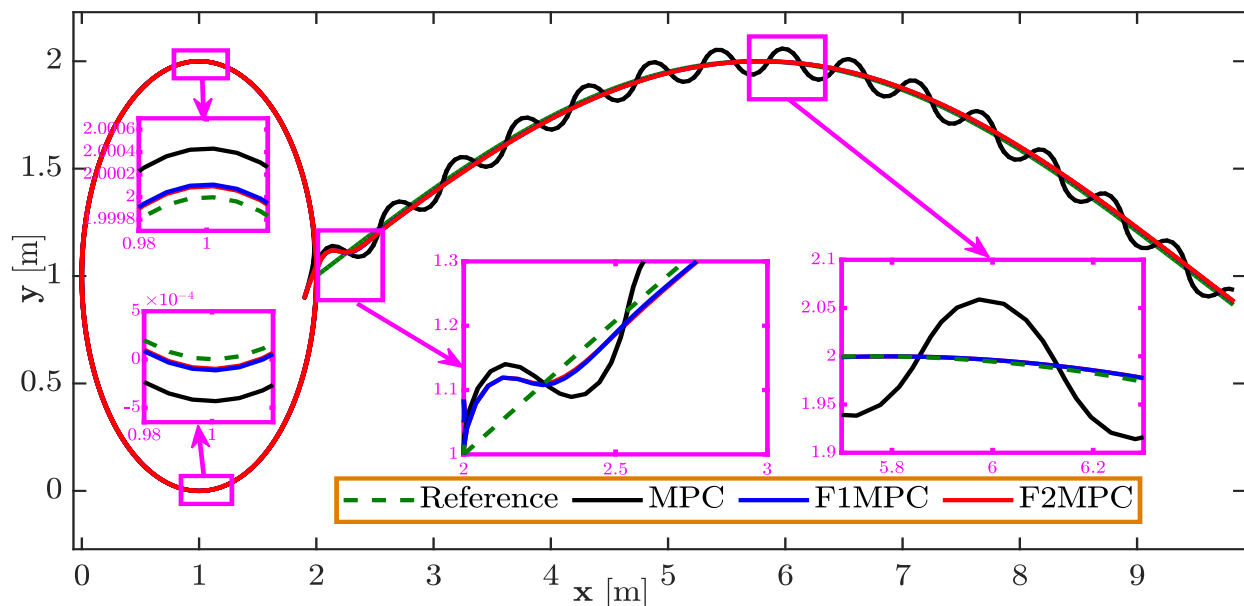


Figure II.19. Suivi des trajectoires circulaires et sinusoïdales à l'aide de contrôleurs MPC flous de type 1 et de type 2, ainsi que du contrôleur MPC

Les profils d'erreur de suivi correspondant à la trajectoire composite fournissent une évaluation comparative de trois stratégies de contrôle: MPC (Fig. II.20), MPC flou de type 1 (F1MPC) (Fig. II.21) et MPC flou de type 2 (F2MPC) (Fig. II.22). Pendant la phase initiale (0–30 s), où le robot suit une trajectoire circulaire régulière, le F2MPC montre une performance de suivi supérieure, caractérisée par des erreurs faibles et stables sur toutes les variables d'état, en particulier dans la composante angulaire e_θ . Le F1MPC donne des résultats modérément améliorés par rapport au MPC, qui présente des écarts directionnels

significatifs et une dynamique d'erreur moins stable. À mesure que le système passe à la trajectoire sinusoïdale plus complexe au-delà de 30 secondes, une augmentation des erreurs de suivi est observée en raison de la courbure accrue et de la variation directionnelle. Néanmoins, le F2MPC maintient une performance fluide et constante, s'adaptant efficacement à la complexité accrue de la trajectoire tout en maintenant les amplitudes d'erreur minimales. Le F1MPC s'adapte également de manière raisonnable, bien qu'avec un léger retard dans la réactivité. En revanche, le contrôleur MPC peine à maintenir la précision sous les exigences dynamiques plus élevées, comme en témoignent les oscillations prononcées et la stabilité angulaire dégradée.

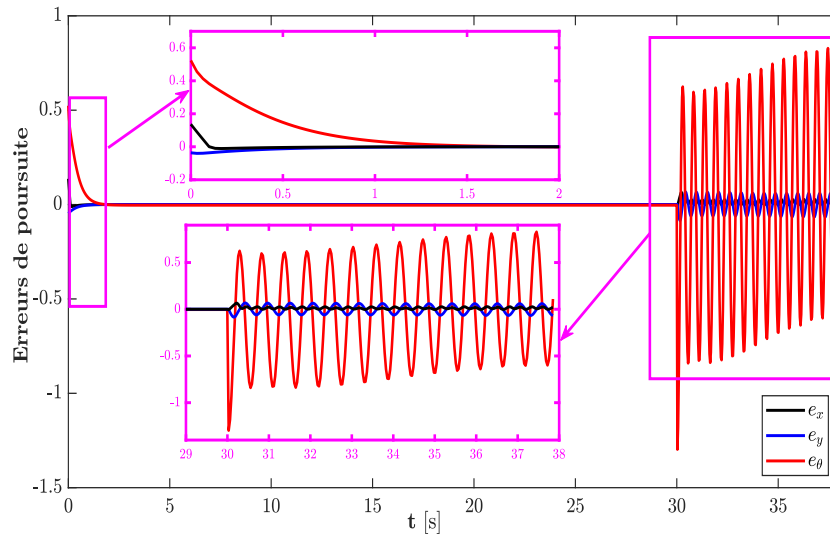


Figure II.20. Erreurs de suivi utilisant l'approche MPC.

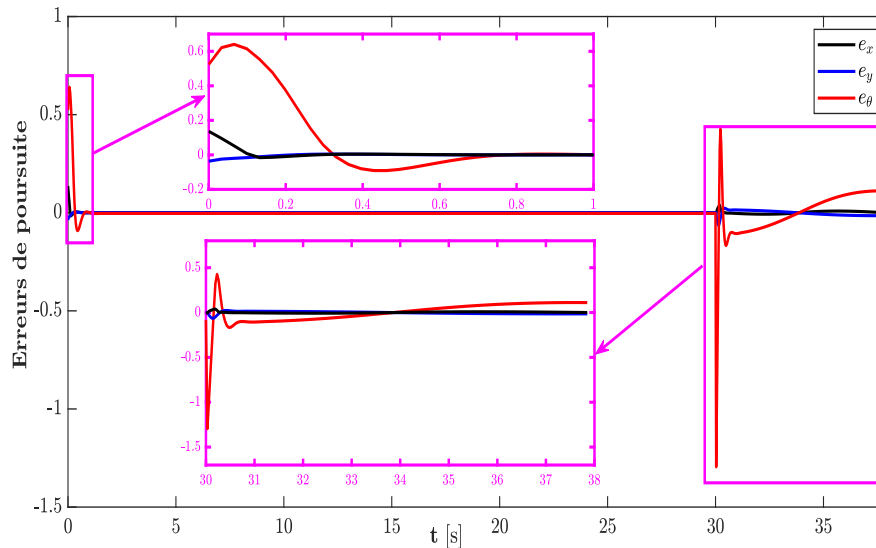


Figure II.21. Erreurs de suivi utilisant l'approche F1MPC.

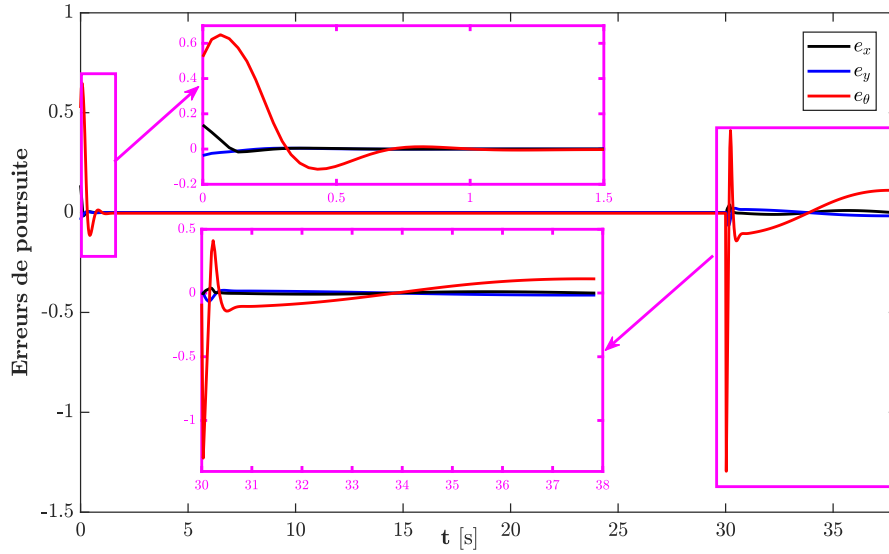


Figure II.22. Erreurs de suivi utilisant l'approche F2MPC.

Les Figures II.23 et II.24 offrent une vue détaillée des entrées de contrôle générées par les deux techniques de contrôle proposées, (F1MPC) dans la Figure 18 et (F2MPC) dans la Figure 19, respectivement. Ces figures affichent les valeurs réelles des signaux de contrôle (vitesses angulaires et linéaires) appliquées au robot au fil du temps. À partir de ces figures, on peut observer les caractéristiques des entrées de contrôle, telles que leur fluidité, la plage de valeurs dans laquelle elles varient, et leur réponse aux changements dans la trajectoire de référence.

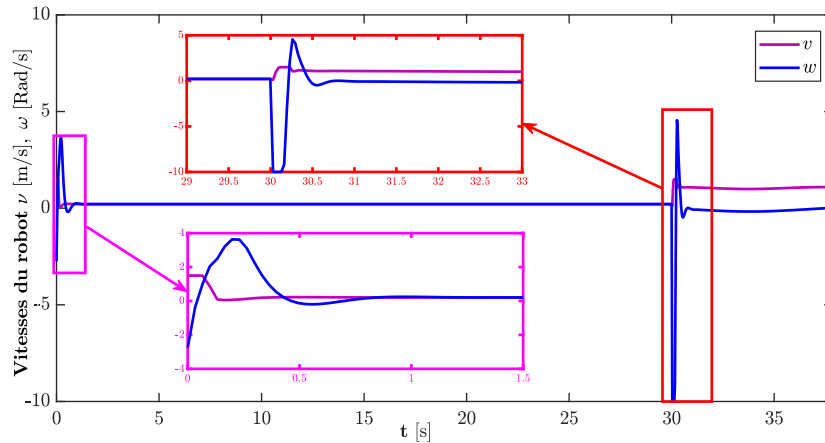


Figure II.23. Vitesses linéaire et angulaire utilisant l'approche F1MPC.

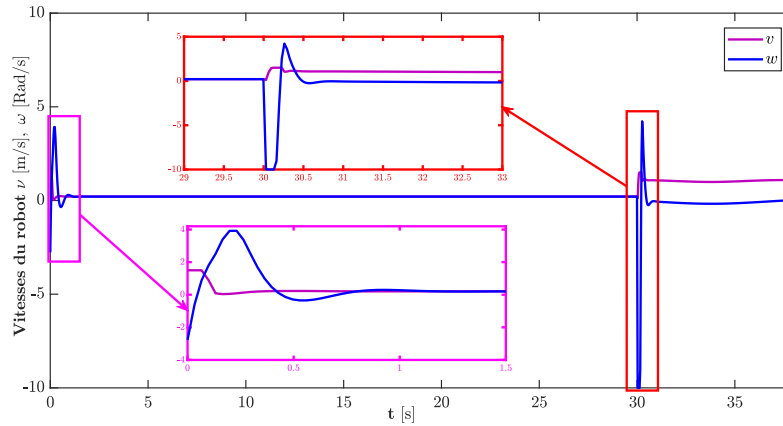


Figure II.24. Vitesses linéaire et angulaire utilisant l'approche F2MPC.

En revanche, la Fig. II.25 montre les entrées de contrôle générées par le contrôleur traditionnel (MPC). En comparant ces entrées avec celles présentées dans les Fig. II.23 et II.24, on peut observer les oscillations dans les valeurs des entrées, ce qui pourrait affecter les performances et la stabilité du robot.

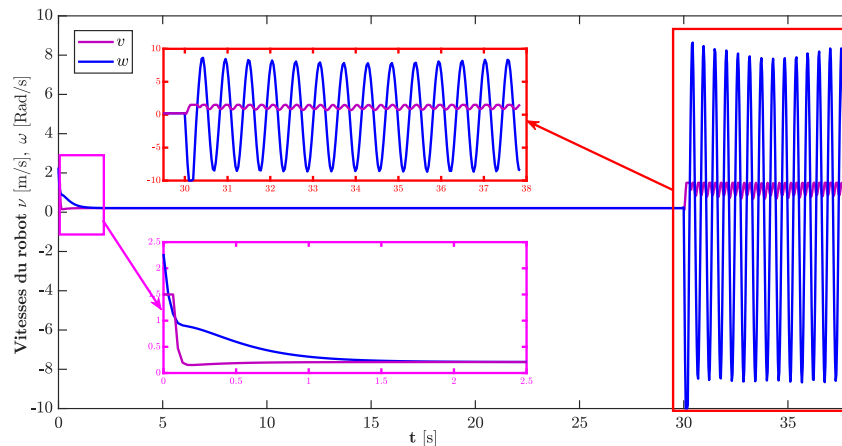


Figure II.25. Vitesses linéaire et angulaire utilisant l'approche MPC.

Pour comparer les performances de tous les contrôleurs en termes de précision de suivi, de stabilité et d'effort de contrôle, diverses métriques ont été adoptées, notamment: l'Erreur Absolue Intégrale (IAE), l'Erreur Quadratique Intégrale (ISE), l'Erreur Quadratique de Temps Intégral (ITSE), l'Erreur Absolue de Temps Intégral (ITAE), l'Erreur Quadratique Moyenne (RMSE), la Variance de Suivi, l'Actuation Intégrée au Carré (ISA) et l'Effort de Contrôle.

En termes de précision de suivi, le F2MPC présente les meilleures performances globales. Il montre systématiquement les valeurs d'erreur les plus faibles pour toutes les métriques (ISE, IAE, ITSE, ITAE,

RMSE), comme indiqué dans le Tableau II.4, et ce, pour les trois composants de suivi: position en x e_x , position en y e_y , et orientation e_θ . Par exemple, pour l'erreur d'orientation, le F2MPC réduit l'ISE de 2,2630 (dans MPC) à 0,2392 et l'ITAE de 126,8356 à 25,0638. Ces améliorations reflètent un contrôleur plus précis et réactif qui s'adapte efficacement aux trajectoires dynamiques.

La stabilité, mesurée par la variance de suivi, met encore en évidence les avantages des contrôleurs basés sur la logique floue. Tant le F1MPC que le F2MPC présentent une variance de suivi proche de zéro, ce qui indique un comportement du système hautement stable et fluide avec des fluctuations minimales à l'état stable. En revanche, le MPC présente une variance de 0,0599 dans le suivi de l'orientation, suggérant la présence d'oscillations et une fluidité réduite pendant la période d'état stable.

En ce qui concerne l'effort de contrôle et la consommation d'énergie, comme rapporté dans le Tableau II.5, le F2MPC s'avère encore être le plus efficace. Il montre les valeurs les plus faibles tant pour l'ISA (31,9548) que pour l'effort de contrôle total (973,9). Le F1MPC suit de près avec des valeurs légèrement plus élevées, tandis que le MPC nécessite beaucoup plus d'énergie (ISA de 291,5603 et effort de contrôle de 8875,6), indiquant une actuation moins efficace.

En résumé, le F2MPC offre la solution la plus équilibrée en combinant une précision de suivi élevée, une excellente stabilité et une consommation d'énergie minimale.

Tableau II.4. Comparaison des indices de performance des erreurs des contrôleurs.

Type de contrôleur	Type d'erreur	ISE	IAE	ITSE	ITAE	RMSE	Variance de suivi
MPC	e_x	0.0028	0.1151	0.0694	3.4472	0.0090	0.0001
	e_y	0.0159	0.3395	0.5236	10.7344	0.0205	0.0004
	e_θ	2.2630	3.9874	74.6035	126.8356	0.2447	0.0599
F1MPC	e_x	0.0013	0.0676	0.0178	1.8717	0.0064	0.0000
	e_y	0.0017	0.0999	0.0530	3.1752	0.0067	0.0000
	e_θ	0.2399	0.9599	5.2067	24.8545	0.0804	0.0065
F2MPC	e_x	0.0011	0.0595	0.0137	1.5939	0.0062	0.0000
	e_y	0.0015	0.0939	0.0474	2.9609	0.0064	0.0000
	e_θ	0.2392	0.9586	5.2848	25.0638	0.0802	0.0064

Tableau II.5. Évaluation de l'intensité d'action de commande à l'aide de l'ISA et de l'effort de commande.

Contrôleur	ISA	Effort de commande
MPC	291.5603	8875.6
F1MPC	32.1313	979.2
F2MPC	31.9548	973.9

II.4.6.4. Test de robustesse

Dans cette section, une perturbation limitée de type impulsion a été introduite dans le vecteur d'état du robot pendant des intervalles de temps prédéfinis afin d'évaluer la réponse des différents contrôleurs. La perturbation consiste en un vecteur à trois éléments ajouté directement à l'état $[x; y; \theta]$, produisant une déviation soudaine de la position et de l'orientation du robot. Elle a été appliquée sur une fenêtre temporelle très étroite (un seul pas par état), constituant une perturbation momentanée et de magnitude fixe, plutôt qu'une perturbation aléatoire continue (comme le bruit blanc).

L'objectif de cette configuration est d'évaluer la capacité du contrôleur à se rétablir rapidement après la déviation et à restituer la trajectoire de référence après la perturbation. Cela constitue une méthode efficace pour évaluer la robustesse et la résilience des systèmes de suivi.

Les résultats dérivés de la Fig. II.26 démontrent clairement la supériorité des techniques proposées (F1MPC et F2MPC) dans la gestion des effets des perturbations externes par rapport au contrôleur MPC traditionnel. Il est observé que les techniques proposées présentent des écarts plus faibles par rapport à la trajectoire de référence et y reviennent plus rapidement après avoir été soumises à une perturbation.

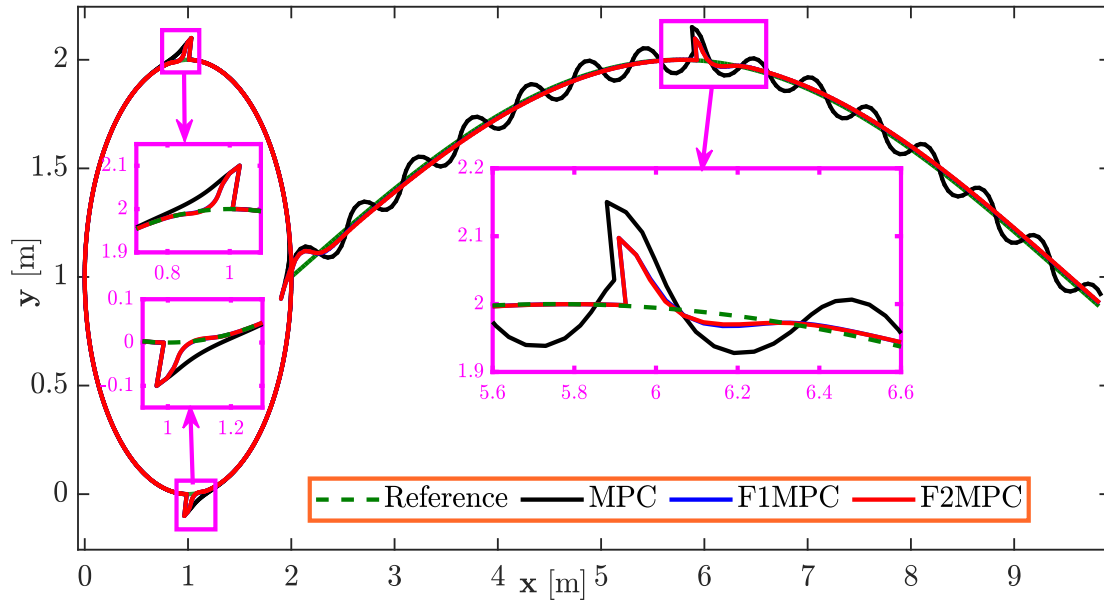


Figure II.26. Robustesse du suivi des trajectoires circulaire et sinusoïdale.

Les courbes d'erreur de suivi pour les contrôleurs MPC, F1MPC et F2MPC sous l'influence de perturbations de type impulsion ajoutées au vecteur d'état $[x;y;\theta]$ montrent clairement des différences distinctes dans la capacité de chaque contrôleur à rejeter les perturbations et à restaurer la trajectoire de référence. La Figure II.27 montre que le contrôleur MPC traditionnel souffre de déviations significatives, en particulier dans l'erreur d'orientation e_θ , accompagnées d'un retard notable dans la stabilisation et d'oscillations prolongées après l'apparition de la perturbation.

En revanche, les Figures II.28 et II.29 montrent que les contrôleurs F1MPC et F2MPC présentent des réponses plus rapides et plus stables. Les erreurs de suivi restent étroitement limitées et reviennent rapidement à leurs valeurs de référence après la perturbation. Les zooms dans les graphiques soulignent que le F2MPC offre des performances dynamiques supérieures par rapport au F1MPC, avec un amortissement amélioré et une convergence plus rapide, mettant en évidence l'efficacité de la structure de logique floue de type 2 dans l'amélioration de la robustesse et de la précision du suivi.

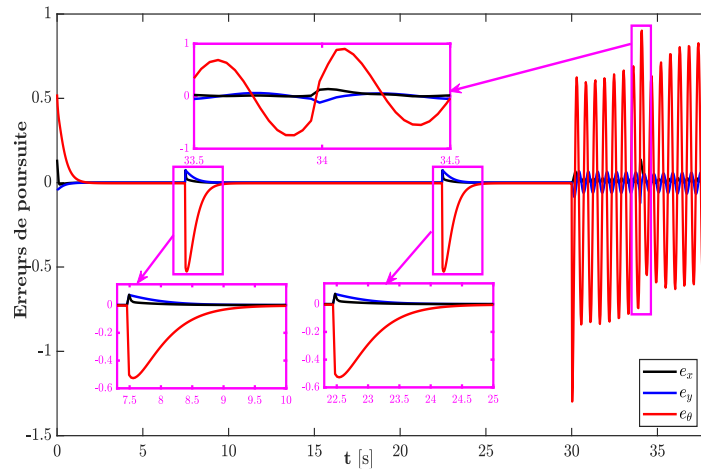


Figure II.27. Erreurs de suivi utilisant l'approche MPC.

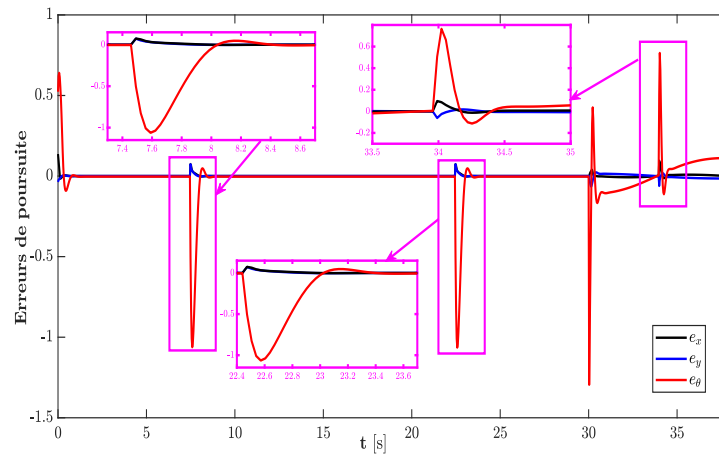


Figure II.28. Erreurs de suivi utilisant l'approche F1MPC.

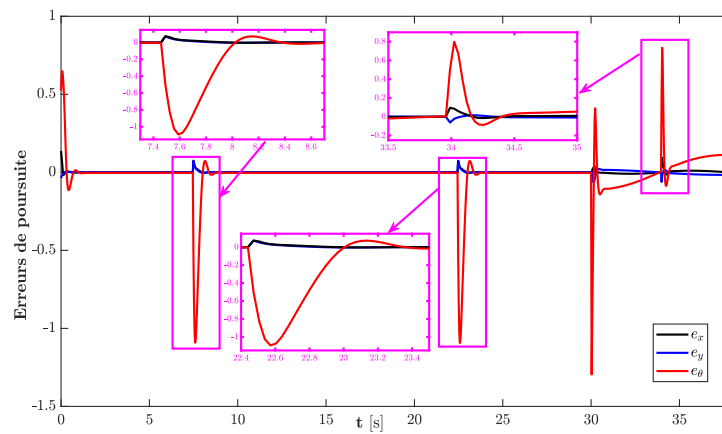


Figure II.29. Erreurs de suivi utilisant l'approche F2MPC.

Les Figures II.30 et II.31 fournissent des détails sur la réponse en vitesse du robot aux perturbations lors de l'utilisation des techniques proposées (F1MPC et F2MPC), respectivement. Ces figures montrent un

ajustement dynamique rapide des vitesses linéaire et angulaire du robot pour restaurer efficacement la trajectoire de référence après la perturbation.

En revanche, la Figure II.32 présente les résultats pour le contrôleur MPC traditionnel, où des réponses en vitesse plus lentes et des oscillations plus importantes sont observées, indiquant une réduction de l'efficacité dans le rejet des perturbations et la récupération de la trajectoire.

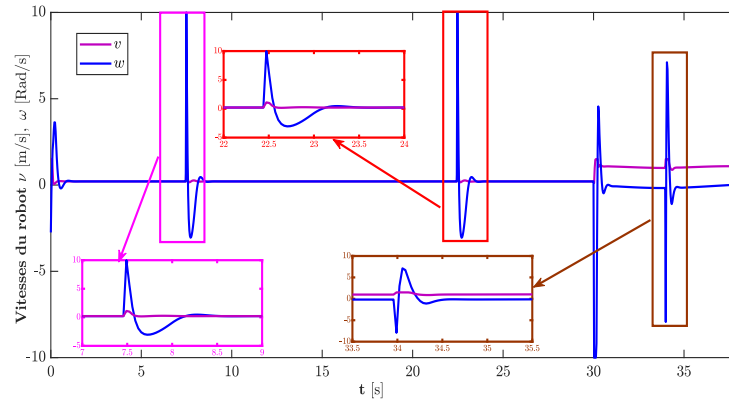


Figure II.30. Illustration du test de robustesse des vitesses angulaire et linéaire utilisant l'approche F1MPC.

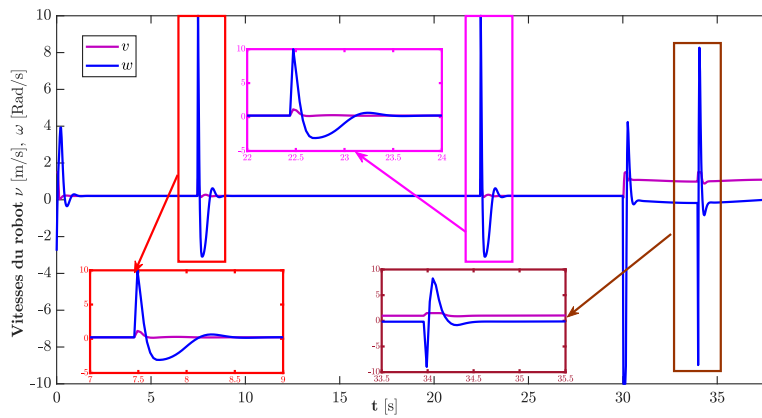


Figure II.31. Illustration du test de robustesse des vitesses angulaire et linéaire utilisant l'approche F2MPC

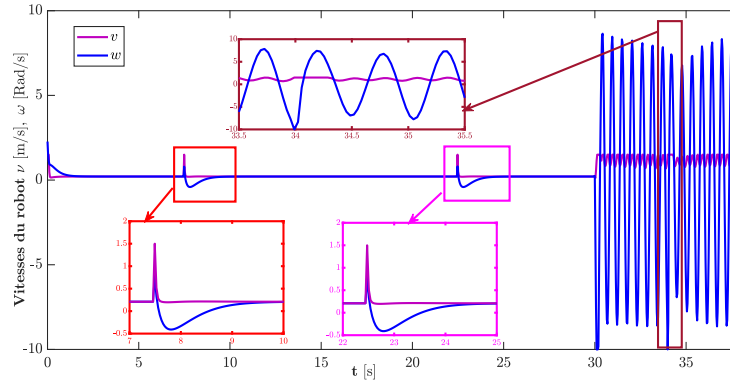


Figure II.32. Illustration du test de robustesse des vitesses angulaire et linéaire utilisant l'approche MPC.

Les résultats présentés dans les Tables II.6 et II.7 fournissent une comparaison détaillée et complète des performances des trois contrôleurs (MPC, F1MPC et F2MPC) lors des tests de robustesse. Selon la Table II.6, le contrôleur MPC conventionnel enregistre les valeurs d'erreur les plus élevées, en particulier dans l'erreur d'orientation e_θ , avec $ISE = 2,4647$, $ITAE = 136,2406$ et $RMSE = 0,2556$. Ces valeurs indiquent une capacité limitée à rejeter les perturbations et à restaurer rapidement la stabilité. De même, les erreurs de position dans le MPC sont clairement observées, par exemple, sur l'axe des y, avec $ISE = 0,0189$ et $ITAE = 11,7845$.

En revanche, tant le F1MPC que le F2MPC montrent des améliorations significatives, avec des valeurs considérablement plus faibles sur tous les indices de performance. Pour l'erreur d'orientation e_θ , le F1MPC enregistre $ISE = 0,8218$ et $ITAE = 38,3144$, tandis que le F2MPC présente des performances légèrement meilleures avec $ISE = 0,7957$ et $ITAE = 38,3338$. De plus, le RMSE est notablement réduit dans le F2MPC à $0,1454$, contre $0,1477$ dans le F1MPC et $0,2556$ dans le MPC. Les erreurs de suivi de position sont également considérablement plus faibles, par exemple, dans la direction y, le F2MPC atteint $ISE = 0,0025$ et $ITAE = 3,4571$, contre $0,0189$ et $11,7845$, respectivement, dans le MPC.

En ce qui concerne la variance de suivi, tant le F1MPC que le F2MPC maintiennent des valeurs proches de zéro (par exemple, $0,0001$) pour toutes les composantes d'erreur, ce qui indique une grande stabilité et des performances précises, tandis que le MPC présente une fluctuation notable dans l'orientation, avec une variance de $0,0650$.

L'analyse de l'effort de contrôle et de la consommation d'énergie dans le Tableau II.7 confirme encore la supériorité du F2MPC en termes d'efficacité énergétique, où il enregistre la valeur la plus basse d'ISA ($52,9609$) et l'effort de contrôle total ($1610,4$), contre F1MPC avec $ISA = 54,9244$ et Effort de Contrôle =

1669,9. En revanche, le MPC nécessite une entrée énergétique nettement plus élevée, avec ISA = 283,7293 et Effort de Contrôle = 8613,1, ce qui reflète un contrôle et une actuation moins efficaces.

Ces résultats démontrent clairement que le contrôleur F2MPC offre le meilleur équilibre entre précision de suivi, faible déviation sous perturbations et efficacité énergétique.

Tableau II.6. Comparaison des indices de performance des erreurs des contrôleurs.

Type de contrôleur	Type d'erreur	ISE	IAE	ITSE	ITAE	RMSE	Variance de suivi
MPC	e_x	0.0057	0.1602	0.1557	4.4730	0.0126	0.0001
	e_y	0.0189	0.4117	0.5678	11.7845	0.0224	0.0005
	e_θ	2.4647	4.6628	76.6240	136.2406	0.2556	0.0650
F1MPC	e_x	0.0030	0.1043	0.0586	2.6307	0.0094	0.0001
	e_y	0.0027	0.1255	0.0711	3.6467	0.0084	0.0001
	e_θ	0.8218	1.7343	14.9896	38.3144	0.1477	0.0216
F2MPC	e_x	0.0029	0.0955	0.0539	2.3423	0.0092	0.0001
	e_y	0.0025	0.1203	0.0662	3.4571	0.0082	0.0001
	e_θ	0.7957	1.7138	14.6780	38.3338	0.1454	0.0209

Tableau II.7. Évaluation de l'intensité d'action de commande à l'aide de l'ISA et de l'effort de commande.

Contrôleur	ISA	Effort de commande
MPC	283.7293	8613.1
F1MPC	54.9244	1669.9
F2MPC	52.9609	1610.4

II.5. Conclusion

En conclusion, ce chapitre a montré que l'intégration des techniques de logique floue dans le cadre du Model Predictive Control (MPC) ouvre de larges perspectives pour le développement de systèmes de contrôle capables de s'adapter à des environnements dynamiques et à des contraintes opérationnelles complexes. En formulant des modèles flous Takagi–Sugeno (T–S) et en les employant pour la prédiction et le contrôle, il est devenu possible de représenter des systèmes linéaires et variant dans le temps avec une précision accrue, améliorant ainsi l'efficacité de la réponse et la vitesse de suivi. De plus,

l'ajustement dynamique des matrices de pondération Q et R , basé sur des systèmes d'inférence floue, a démontré la capacité du système à équilibrer continuellement le suivi de trajectoire et la minimisation de l'effort de commande, garantissant à la fois une performance améliorée et une stabilité accrue.

Le Fuzzy Type-2 MPC (F2MPC) a montré de meilleurs résultats, atteignant un score de précision de position de 0.0011, supérieur à celui du Type-1 fuzzy MPC (0.0013) et du MPC conventionnel (0.0028). Le F2MPC a démontré une grande efficacité énergétique, utilisant 65% moins d'effort de contrôle que le MPC, et a bien géré les perturbations avec très peu de changement (0.0001) et une erreur d'orientation plus faible (RMSE de 0.1454 contre 0.2556 pour le MPC).

L'adaptabilité dynamique du modèle flou de T-S a permis des ajustements en temps réel, garantissant la stabilité lors des transitions de trajectoires complexes. Les travaux futurs se concentreront sur l'optimisation computationnelle pour la mise en œuvre en temps réel, la validation expérimentale dans des contextes non structurés, et les approches hybrides d'apprentissage automatique pour améliorer la flexibilité. Le cadre F2MPC améliore le contrôle des robots avec précision, efficacité et résilience pour des applications réelles. Outre sa précision, cette méthode fournit un cadre systématique pour la construction de modèles flous adaptés à la conception de contrôles avancés en temps réel. Cette approche est développée plus en détail dans le **chapitre suivant** pour les systèmes dynamiques, comme le contrôle prédictif flou (FPC) dans les systèmes non linéaires.

CHAPITRE III

CONSTRUCTION UNIFIEE DES MODELES TAKAGI- SUGENO

III.1. Introduction

Les systèmes dynamiques non linéaires constituent la pierre angulaire de la modélisation d'une large gamme de phénomènes complexes d'ingénierie et naturels, allant des processus mécaniques et électriques aux systèmes chimiques, biologiques et même socio-économiques [137] [138]. Cependant, en raison de leurs non-linéarités inhérentes, de tels systèmes présentent souvent des comportements tels que le couplage entre les variables, la sensibilité aux conditions initiales, les bifurcations et les non-linéarités structurelles, ce qui les rend difficiles à modéliser et à contrôler en utilisant des approches linéaires conventionnelles. Cette complexité pose un défi majeur en ingénierie de contrôle, où une modélisation précise est essentielle pour l'analyse de la stabilité, la conception de contrôleurs et l'optimisation du système [139].

En réponse à ces défis, ce chapitre développe un cadre complet pour remodeler un système dynamique non linéaire en utilisant un ensemble de techniques analytiques avancées et basées sur les données. L'objectif est d'obtenir des représentations précises, flexibles et analytiquement maniables qui capturent la dynamique non linéaire du système tout en restant adaptées à la conception de contrôle avancée. Pour y parvenir, trois approches de modélisation complémentaires sont introduites, chacune reflétant une perspective philosophique et méthodologique différente sur la gestion des non-linéarités.

Ce chapitre présente trois techniques de modélisation de systèmes non linéaires. Premièrement, la modélisation floue de Takagi-Sugeno (T-S) décrit les dynamiques complexes via une interpolation de sous-modèles linéaires locaux. Elle permet une analyse rigoureuse de la stabilité par des inégalités matricielles et des fonctions de Lyapunov. Deuxièmement, la transformation de secteur non linéaire (NST) recadre les non-linéarités dans une représentation polytopique convexe. Cette forme permet d'appliquer des outils d'analyse linéaire tout en conservant le comportement non linéaire essentiel. Troisièmement, l'identification de modèles T-S par l'algorithme de Levenberg-Marquardt ajuste les paramètres à partir de données expérimentales. Cette approche adaptative améliore la précision du modèle sous diverses conditions opératoires. Chaque méthodologie sera détaillée dans ses fondements, sa structure et sa mise en œuvre pratique. Leur performance sera évaluée en termes de précision, flexibilité et adéquation pour le contrôle avancé. Ce cadre unifié, mêlant analyse théorique et adaptation basée sur les données, servira de base à la conception de contrôleurs dans les chapitres suivants.

III.2. Énoncé du problème et système non linéaire

L'analyse et la modélisation des systèmes dynamiques non linéaires présentent des défis fondamentaux en ingénierie de contrôle, en raison du comportement dynamique complexe exhibé par ces systèmes et

de la difficulté à appliquer les méthodologies d'analyse linéaire traditionnelles. Dans ce cadre, cette section introduit le système non linéaire qui constitue le cœur de cette étude, tout en soulignant les défis associés à son analyse et à la conception de son contrôle.

Le système étudié est exprimé sous forme d'espace d'état comme suit:

$$\dot{x}(t) = f(x(t), u(t)) = \begin{bmatrix} -ux_1 + u \\ -x_1x_2 + u \end{bmatrix} \text{ où } x = \begin{bmatrix} x_1 \\ x_2 \end{bmatrix}, u \in \mathbb{R} \quad (\text{III.1})$$

Ce système est caractérisé par un couplage non linéaire entre les variables d'état x_1 et x_2 , en plus de la dépendance non linéaire de la dynamique du système par rapport à l'entrée de contrôle u . Cette complexité dynamique entraîne des difficultés dans l'analyse théorique directe et rend l'application efficace des techniques de conception de contrôle linéaire impossible.

Pour surmonter ces défis, ce travail adopte une méthodologie de modélisation avancée basée sur l'approximation du comportement non linéaire en utilisant plusieurs systèmes linéaires, tout en préservant les caractéristiques dynamiques fondamentales du système original. Les sections suivantes de ce chapitre présenteront trois techniques intégrées pour remodeler le système non linéaire:

- ✓ *Modélisation floue de Takagi-Sugeno (T-S)*: Cette technique représente le comportement non linéaire comme un mélange flou de modèles linéaires locaux activés par des fonctions d'appartenance gaussiennes.
- ✓ *Transformation sectorielle et représentation polytopique*: Cette approche convertit les relations non linéaires en une combinaison convexe de systèmes linéaires locaux au sein de secteurs de fonctionnement définis.
- ✓ *Identification de modèle flou utilisant l'algorithme de Levenberg-Marquardt*: Il s'agit d'une approche numérique visant à optimiser les paramètres du modèle flou en se basant sur des données d'entraînement acquises à partir du système réel.

III.3. Approche de modélisation floue de Takagi–Sugeno (TS) pour les systèmes non linéaires

Le système non linéaire introduit dans la section précédente est approximé en utilisant le cadre de modélisation floue de Takagi–Sugeno (TS). L'idée principale est de remplacer la dynamique non linéaire originale par un ensemble de modèles linéaires locaux, chacun obtenu en linéarisant le système autour d'un point de fonctionnement spécifique. Mathématiquement, le modèle flou TS est exprimé comme une combinaison convexe de ces modèles locaux:

$$\dot{x}(t) = \sum_{i=1}^R \mu_i(z(t))(A_i x(t) + B_i u(t) + D_i) \quad (\text{III.2})$$

où:

- R : nombre de règles floues (modèles locaux).
- A_i, B_i, D_i : sont les matrices du système obtenues par linéarisation au i -ème point de fonctionnement.
- $\mu_i(z(t))$: sont les fonctions d'appartenance, qui satisfont la propriété de convexité:

$$\mu_i(z(t)) \geq 0, \sum_{i=1}^R \mu_i(z(t)) = 1, \mu_i(z) = \frac{\exp\left(-\frac{1}{2}\left(\frac{z(t)-c_i}{\sigma_i}\right)^2\right)}{\sum_{i=1}^R \exp\left(-\frac{1}{2}\left(\frac{z(t)-c_i}{\sigma_i}\right)^2\right)} \quad (\text{III.3})$$

- $z(t)$: représente les variables de planification (prémisse) utilisées pour l'activation des modèles locaux.

le modèle linéarisé local autour de chaque point de fonctionnement (x_i, u_i) est défini par:

$$A_i = \left. \frac{\partial f}{\partial x} \right|_{\substack{x=x_i \\ u=u_i}} = \begin{bmatrix} -u_i & 0 \\ -x_{2i} & -x_{1i} \end{bmatrix},$$

$$B_i = \left. \frac{\partial f}{\partial u} \right|_{\substack{x=x_i \\ u=u_i}} = \begin{bmatrix} -x_{1i} + 1 \\ 1 \end{bmatrix}, \quad (\text{III.4})$$

$$D_i = f(x_i, u_i) - A_i x_i - B_i u_i = \begin{bmatrix} x_{1i} u_i \\ x_{1i} x_{2i} \end{bmatrix}$$

Dans ce travail, des fonctions d'appartenance gaussiennes sont adoptées pour pondérer la contribution de chaque modèle local. Ces fonctions assurent une interpolation fluide entre les différents modèles linéaires, préservant ainsi le comportement non linéaire du système original tout en maintenant le modèle mathématiquement tractable pour l'analyse et le contrôle [140].

III.3.1. Évaluation de l'erreur et fonction de coût quadratique moyenne (MSE)

Afin d'évaluer la précision du modèle flou de Takagi–Sugeno proposé, l'erreur est définie comme la différence entre la réponse réelle du système non linéaire et la sortie du modèle flou:

$$e(t) = x_{real}(t) - x_{t-s}(t) \quad (\text{III.5})$$

Où:

- $x_{real}(t)$: vecteur d'état du système non linéaire original.
- $x_{t-s}(t)$: vecteur d'état obtenu à partir du modèle flou TS.

Pour quantifier la qualité globale de l'approximation, une fonction de coût est formulée comme l'Erreur Quadratique Moyenne (MSE):

$$J = \frac{1}{N} \sum_{k=1}^N \|e(k)\|^2 \quad (\text{III.6})$$

Où:

- N : nombre d'échantillons dans l'horizon de simulation. $\|e(k)\|$: norme de l'erreur quadratique à l'étape de temps k .

L'objectif du processus d'identification et d'optimisation des paramètres est donc de minimiser la fonction de coût J selon le paramètre θ eq (III.8), ce qui garantit que le modèle flou approxime la dynamique du système non linéaire aussi précisément que possible.

III.3.2. Identification des paramètres par l'algorithme de Levenberg-Marquardt (LM)

Le but de l'algorithme de Levenberg-Marquardt (LM) est d'estimer les paramètres du modèle flou en minimisant l'écart entre le système non linéaire réel et le modèle flou TS approximé. Cela est réalisé en minimisant la fonction de coût de l'erreur quadratique moyenne (MSE) [140]:

$$J(\theta) = \frac{1}{N} \sum_{k=1}^N \|x_{real}(k) - x_{t-s}(k, \theta)\|^2 \quad (\text{III.7})$$

- **Vecteur de paramètres θ** : Dans ce travail, le vecteur de paramètres inclut à la fois les paramètres de la fonction d'appartenance et les points de fonctionnement:

$$\theta = \{x_{1i}, x_{2i}, u_i, c_i, \sigma_i \quad \dots \quad x_{1R}, x_{2R}, u_R, c_R, \sigma_R\}, i = \overline{1:R} \quad (\text{III.8})$$

Où:

- c_i : centres des fonctions d'appartenance gaussiennes.
- σ_i : variances (largeurs) des fonctions gaussiennes.
- x_{1i}, x_{2i}, u_i : points de fonctionnement autour desquels le système non linéaire est linéarisé.
- R : nombre de règles floues (modèles locaux).

Le vecteur de paramètres est affiné de manière itérative en utilisant la règle de mise à jour de Levenberg-Marquardt:

$$\theta(k+1) = \theta(k) - (J(\theta)^T J(\theta) + \lambda I)^{-1} J(\theta)^T e(\theta(k)) \quad (\text{III.9})$$

Où:

- $e(\theta(k))$: vecteur d'erreur entre le système réel et le modèle flou.
- $J(\theta)$: Matrice jacobienne de l'erreur par rapport à θ .
- λ : paramètre d'amortissement équilibrant la descente de gradient et les directions de Gauss-Newton.

Avantages:

- Convergence rapide grâce aux mises à jour basées sur le Jacobien,
- Stabilité numérique assurée par le facteur d'amortissement λ ,
- Ajustement intégré des fonctions d'appartenance et des points de fonctionnement pour obtenir une approximation précise du système non linéaire.

III.3.3. Analyse de stabilité par inégalités matricielles linéaires (LMI):

Pour garantir la stabilité globale du modèle flou optimisé de Takagi–Sugeno, une approche de stabilité de Lyapunov quadratique est employée. La condition de stabilité est formulée comme un ensemble d'Inégalités Matricielles Linéaires (LMI), qui sont résolues numériquement pour trouver une matrice de Lyapunov commune.

a) Formulation du théorème de stabilité de Lyapunov:

Le point d'équilibre du système flou est globalement asymptotiquement stable s'il existe une matrice P définie positive commune telle que pour tous les modèles locaux:

$$A_i^T P + P A_i < 0 \text{ Pour } i = 1, 2, \dots, R \quad (\text{III.10})$$

Où A_i est la matrice d'état du i -ème modèle linéaire local.

b) Formulation LMI et solution numérique:

En pratique, une condition plus stricte est utilisée pour garantir la faisabilité numérique et la robustesse:

$$A_i^T P + P A_i < -Q \text{ Pour } i = 1, 2, \dots, R \quad (\text{III.11})$$

Où Q est une petite matrice définie positive. Cette formulation est ensuite traitée et résolue à l'aide d'un solveur d'optimisation convexe approprié

L'existence d'une solution réalisable $P > 0$ au problème LMI ci-dessus garantit la stabilité asymptotique globale du modèle flou de Takagi–Sugeno. Cette approche fournit un certificat mathématique rigoureux de stabilité pour l'approximation du modèle non linéaire optimisé. [150]

III.3.4. Résultats numériques et validation par simulation

Cette section présente les résultats de simulation du modèle flou TS proposé, en mettant en évidence le processus d'entraînement, la vérification de la stabilité et les performances de validation pour évaluer à la fois la précision et la robustesse.

Les simulations des deux systèmes représentés par les équations (III.11) et (III.2) ont été réalisées avec un temps d'échantillonnage de $dt = 0.01$, une durée totale de $T = 50s$, et $N = 5000$ pas. Un modèle flou avec $R = 5$ a été entraîné en utilisant l'algorithme de Levenberg-Marquardt jusqu'à 200 itérations sous des contraintes de stabilité LMI.

III.3.5. Phase d'entraînement:

Le processus d'optimisation a été réalisé en utilisant l'algorithme de Levenberg-Marquardt (LM), incluant des contraintes LMI pour garantir la stabilité pendant l'entraînement. L'algorithme a commencé avec une valeur d'erreur initiale de 1772,96 et a progressivement réduit la valeur de la fonction objective (erreur quadratique moyenne) sur 200 itérations.

- ✓ **Comportement itératif:** Une diminution rapide de la valeur de l'erreur a été observée au cours des dix premières itérations (de 1772,96 à moins de 1), suivie d'une convergence stable lors des itérations suivantes.
- ✓ **Valeur d'erreur finale:** Après 200 itérations, la valeur finale de la fonction objective a atteint 0,328629, indiquant une grande précision dans l'approximation du système non linéaire à l'aide du modèle flou.
- ✓ **Arrêt de l'optimiseur:** L'algorithme s'est arrêté après avoir atteint le nombre maximal d'itérations autorisées (200 itérations), tout en maintenant la stabilité du modèle tout au long du processus d'optimisation.

A. Signal d'excitation pour l'apprentissage:

Un signal d'entrée composite avec plusieurs composantes de fréquence a été appliqué pour couvrir la plage de fonctionnement du système. L'entrée d'entraînement $u(t)$ est définie comme un signal par morceaux sur l'intervalle $0 \leq t \leq 50s$:

$$u(t) = \begin{cases} 0.5 \sin(0.2t) + 0.3 \sin(1.5t) + 0.2 \sin(3t) & 0 \leq t \leq 10 \\ \text{sawtooth}(2\pi \cdot 0.5t) & 10 \leq t \leq 20 \\ \text{squar}(2\pi \cdot 0.5t) & 20 \leq t \leq 30 \\ \text{chirp}(t, f_0 = 0.1, t_1 = 10, f_1 = 2) & 30 \leq t \leq 40 \\ H(t - 45) + 0.2\eta(t) & 40 \leq t \leq 50 \end{cases} \quad (\text{III.12})$$

Où: $\eta(t)$ est un bruit aléatoire gaussien de moyenne nulle.

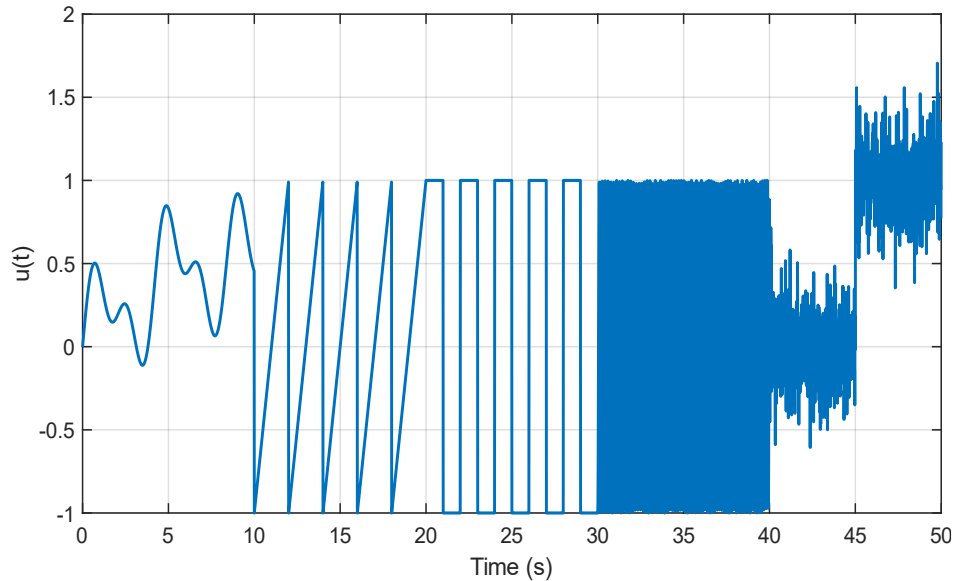


Figure III.1. Signal d'excitation utilisé pour l'entraînement

Cette figure III.1 représente le signal d'excitation d'entrée utilisé pendant la phase d'entraînement. Le signal est conçu pour être riche en composantes de fréquence et en variations temporelles afin d'exciter la dynamique complète du système. Le but est de s'assurer que le modèle de Takagi-Sugeno (T-S) puisse capturer le comportement non linéaire du système sur toute la plage de fonctionnement. L'amplitude reste bornée, ce qui fournit une excitation suffisante sans provoquer d'instabilité pendant la simulation.

Le tableau III.1 rapporte les paramètres optimisés du modèle flou. Chaque règle est définie par ses matrices locales, son vecteur d'entrée et ses paramètres d'appartenance gaussienne, qui ensemble garantissent une représentation précise du système non linéaire sur toute la plage de fonctionnement.

Tableau III.1. Paramètres optimisés pour le modèle flou (5 règles)

Règle	Matrice A_i	Matrice B_i	Matrice D_i	Centre c_i	Variance σ_i
1	$\begin{bmatrix} -0.2264 & 0 \\ -0.0551 & -0.0191 \end{bmatrix}$	$\begin{bmatrix} 0.9809 \\ 1 \end{bmatrix}$	$\begin{bmatrix} 0.0043 \\ 0.0011 \end{bmatrix}$	0.5218	0.7075
2	$\begin{bmatrix} -0.2033 & 0 \\ -0.2652 & -0.5894 \end{bmatrix}$	$\begin{bmatrix} 0.4109 \\ 1 \end{bmatrix}$	$\begin{bmatrix} 0.1198 \\ 0.1563 \end{bmatrix}$	0.4991	0.0620
3	$\begin{bmatrix} -0.3653 & 0 \\ -0.5071 & -0.5606 \end{bmatrix}$	$\begin{bmatrix} 0.4394 \\ 1 \end{bmatrix}$	$\begin{bmatrix} 0.2048 \\ 0.2843 \end{bmatrix}$	0.6292	10^{-6}
4	$\begin{bmatrix} -0.2259 & 0 \\ -0.0550 & -1.8670 \end{bmatrix}$	$\begin{bmatrix} -0.8670 \\ 1 \end{bmatrix}$	$\begin{bmatrix} 0.4218 \\ 0.1027 \end{bmatrix}$	1.2285	0.4899
5	$\begin{bmatrix} -0.2054 & 0 \\ -0.1198 & -0.0626 \end{bmatrix}$	$\begin{bmatrix} 0.9374 \\ 1 \end{bmatrix}$	$\begin{bmatrix} 0.0129 \\ 0.0075 \end{bmatrix}$	-0.2591	0.0048

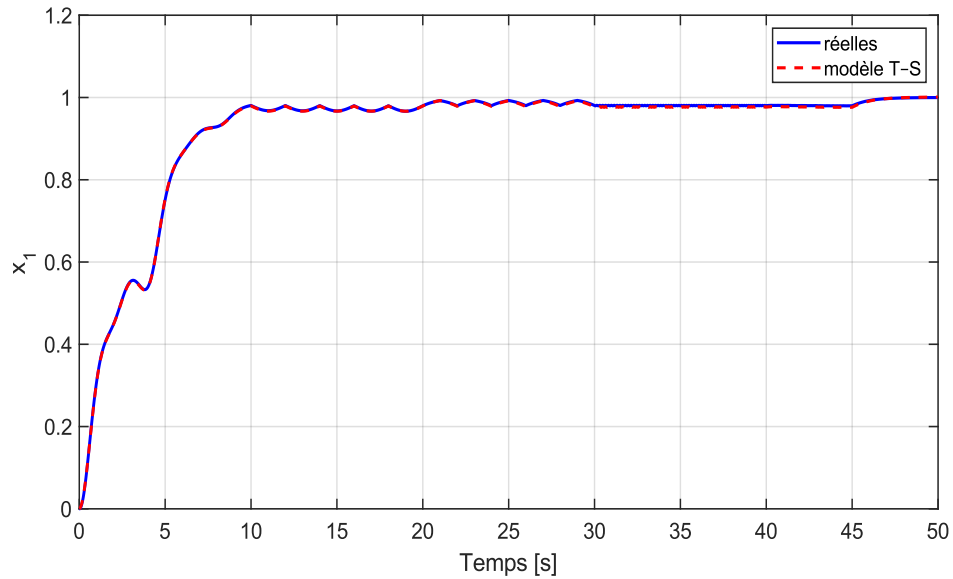


Figure III.2. Comparaison des trajectoires réelles et du modèle T-S pour l'état x_1 (Phase d'entraînement)

Cette figure III.2 montre une comparaison entre la trajectoire temporelle de la première variable d'état x_1 pour le système réel et celle produite par le modèle T-S pendant l'entraînement. La correspondance étroite entre les deux courbes reflète la capacité du modèle à reproduire avec précision la dynamique du système. De légères déviations peuvent apparaître dans les régions transitoires, mais elles restent dans des limites acceptables.

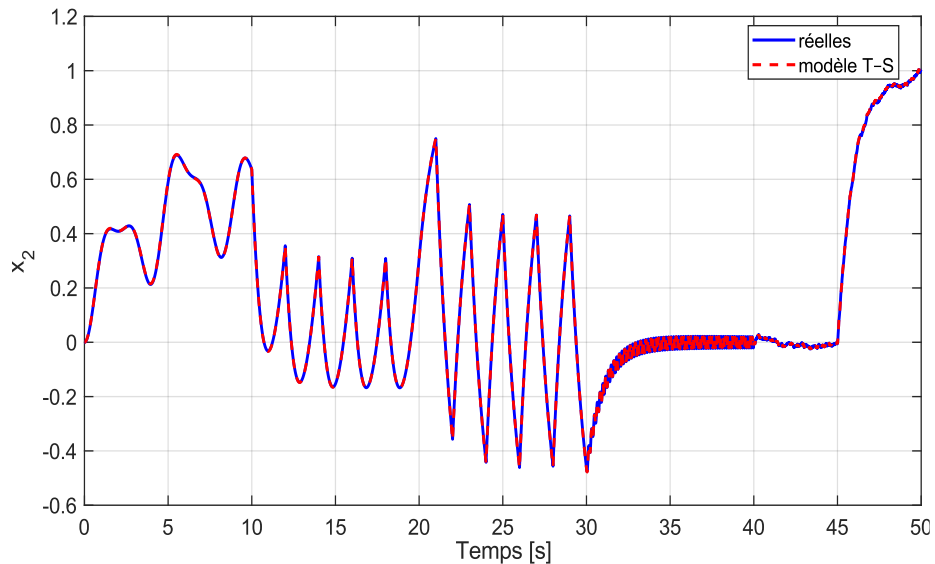


Figure III.3. Comparaison des trajectoires réelles et du modèle T-S pour l'état x_2 (Phase d'entraînement).

Figure III.3 Similaire à la Figure III.2, mais pour la deuxième variable d'état x_2 . Les résultats confirment que le modèle atteint une bonne précision dans la capture des dynamiques des deux états internes. Cela indique que l'interdépendance entre les états a été correctement représentée par les modèles linéaires locaux et les règles de fusion floues.

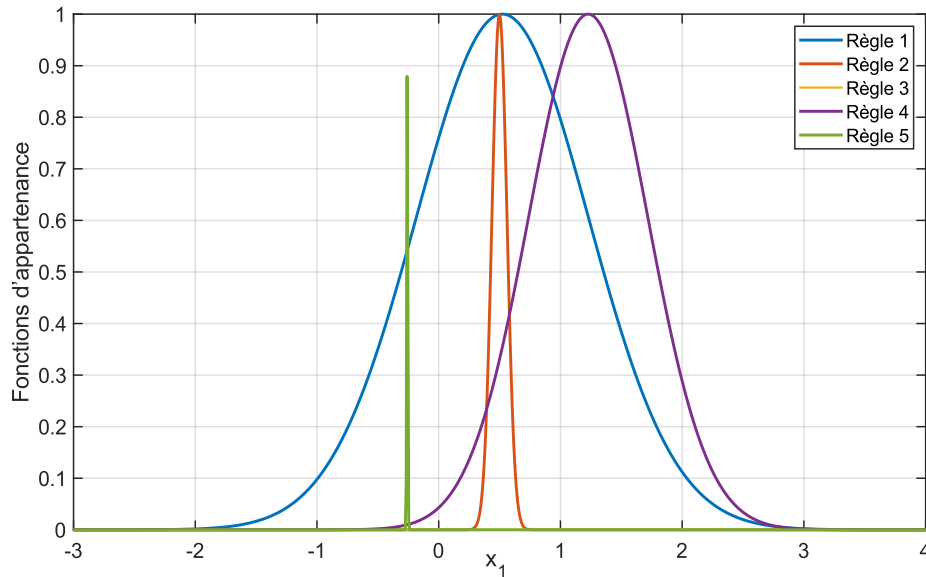


Figure III.4. Fonctions d'appartenance gaussiennes des règles floues

Cette figure III.4 illustre les fonctions d'appartenance gaussiennes associées aux règles floues. Ces fonctions déterminent le degré d'activation de chaque règle en fonction de la variable de prémisse. Le chevauchement entre les fonctions garantit des transitions fluides entre différents modèles linéaires locaux, empêchant des changements brusques dans la réponse du système. La distribution des centres gaussiens garantit une couverture complète du domaine de fonctionnement.

B. Vérification de la stabilité par résolution LMI:

La stabilité du système du modèle flou optimisé a été évaluée à l'aide d'Inégalités Matricielles Linéaires (LMI) fondées sur la théorie de la stabilité de Lyapunov, ce qui a permis d'obtenir une matrice de Lyapunov définie positive: $P = \begin{bmatrix} 20.8341 & -0.8747 \\ -0.8747 & 6.1582 \end{bmatrix}$. La positivité de cette matrice confirme que le modèle flou est globalement stable.

III.3.6. Validation du modèle avec un signal d'entrée inédit

La phase de validation teste le modèle flou TS optimisé avec un nouveau signal d'entrée pour évaluer sa capacité de généralisation et sa précision dans des conditions de fonctionnement non observées.

La généralisation du modèle a été validée en utilisant un signal d'entrée indépendant:

$$u_{test} = 0.6 \sin(0.3t) + 0.4 \sin(t) + 0.1 \times Noise \quad (III.13)$$

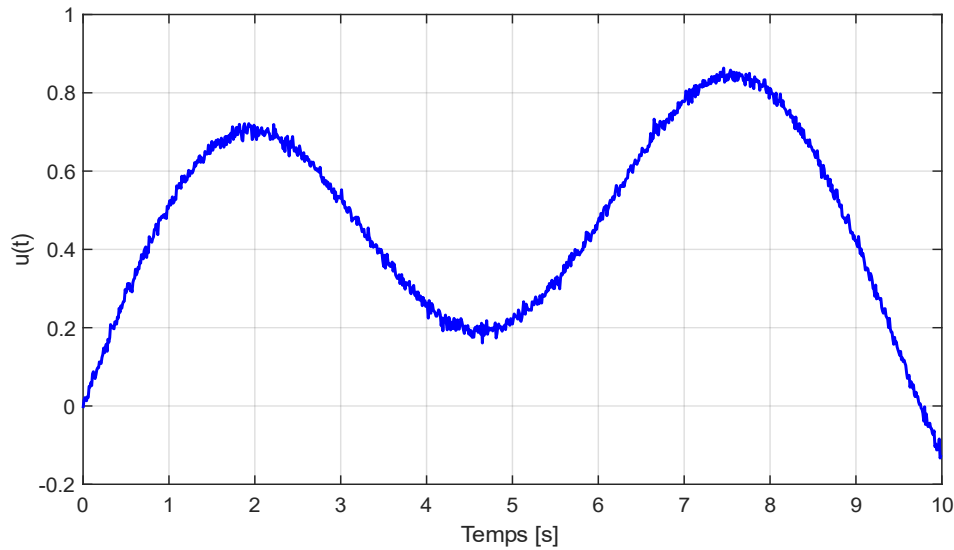


Figure III.5. Signal d'entrée d'excitation de validation

Cette figure III.5 représente le signal d'entrée utilisé pendant la phase de validation, qui diffère du signal d'entraînement. Le but est d'évaluer la capacité de généralisation du modèle au-delà des données d'entraînement. La nature composite de ce signal, impliquant à la fois des variations rapides et lentes, permet d'évaluer la performance du modèle dans de nouvelles conditions de fonctionnement. Les résultats de la validation:

- **MSE pour x_1 :** 4.3×10^{-6}
- **MSE pour x_2 :** 1.7×10^{-6}

Ces erreurs extrêmement petites confirment la capacité du modèle à généraliser avec précision et à reproduire les dynamiques réelles du système.

III.3.7. Comparaison des réponses système réel / modèle TS

La Figure III.6 compare la réponse de validation de la première variable d'état x_1 entre le système réel et le modèle T-S. L'accord étroit continu démontre que le modèle non seulement s'adapte aux données d'entraînement, mais se généralise également efficacement à des scénarios non vus.

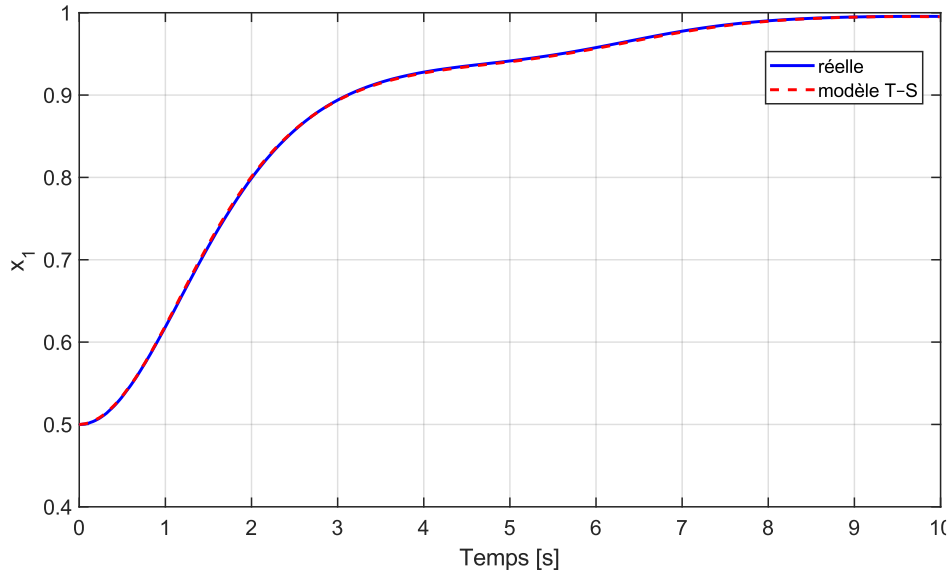


Figure III.6. Comparaison des trajectoires réelles et du modèle T-S pour l'état x_1 (Phase de validation)

Et en suite, la Figure III.7 présente la comparaison de validation pour la deuxième variable d'état x_2 . Les résultats reflètent ceux de x_1 , montrant une forte cohérence entre le système réel et le modèle T-S, confirmant ainsi sa fiabilité.

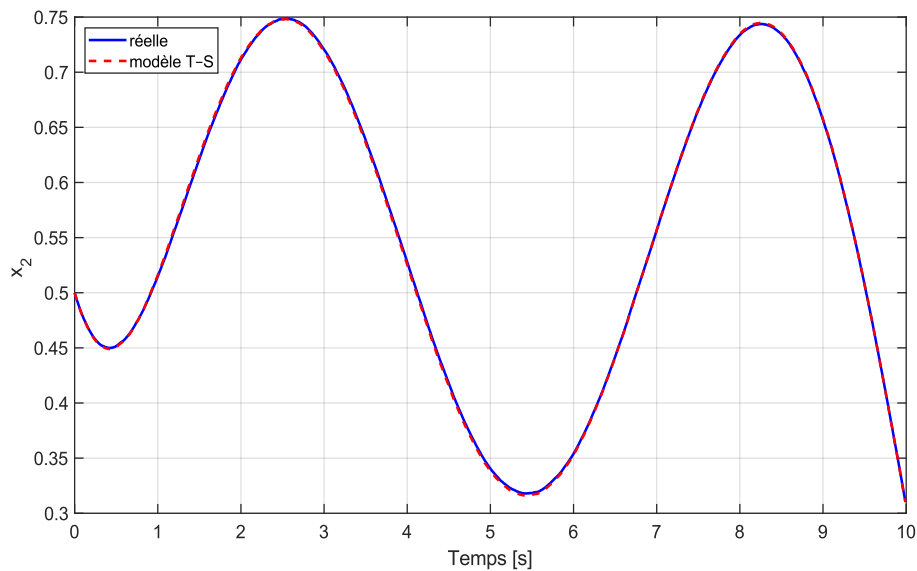


Figure III.7. Comparaison des trajectoires réelles et du modèle T-S pour l'état x_2 (Phase de validation)

Et après, la Figure III.8 montre la trajectoire d'erreur entre le x_1 réel et celui basé sur le modèle pendant la validation. L'erreur reste petite et bornée, ce qui correspond à la valeur de l'erreur quadratique

moyenne (MSE) rapportée dans l'article. Cela met en évidence la précision numérique et la robustesse du modèle identifié.

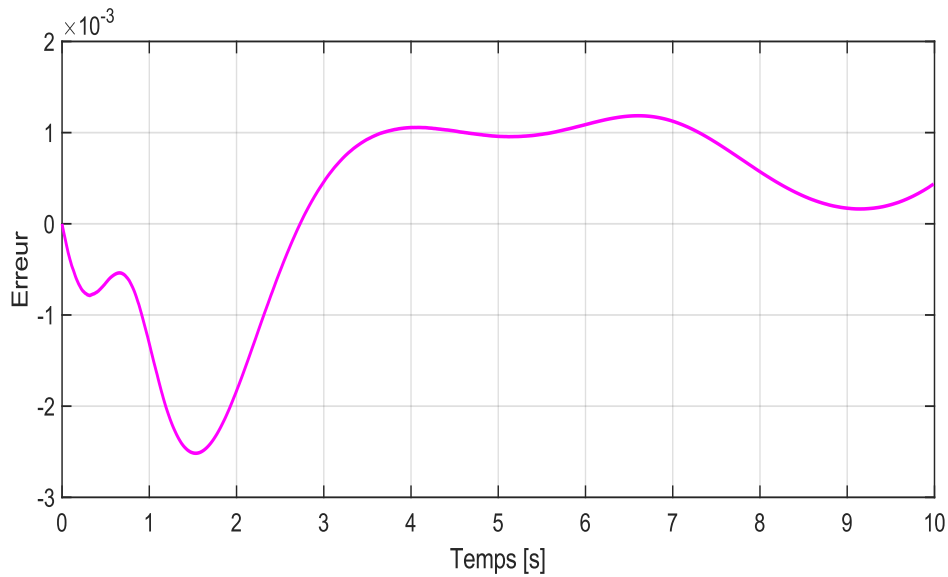


Figure III.8. Trajectoire de l'erreur de validation pour l'état x_1

La Figure III.9 représente la trajectoire d'erreur de la deuxième variable d'état x_2 pendant la validation. Le comportement de l'erreur est similaire à celui de x_1 , restant petit sans divergence, ce qui indique que le modèle décrit de manière fiable la dynamique des deux états.

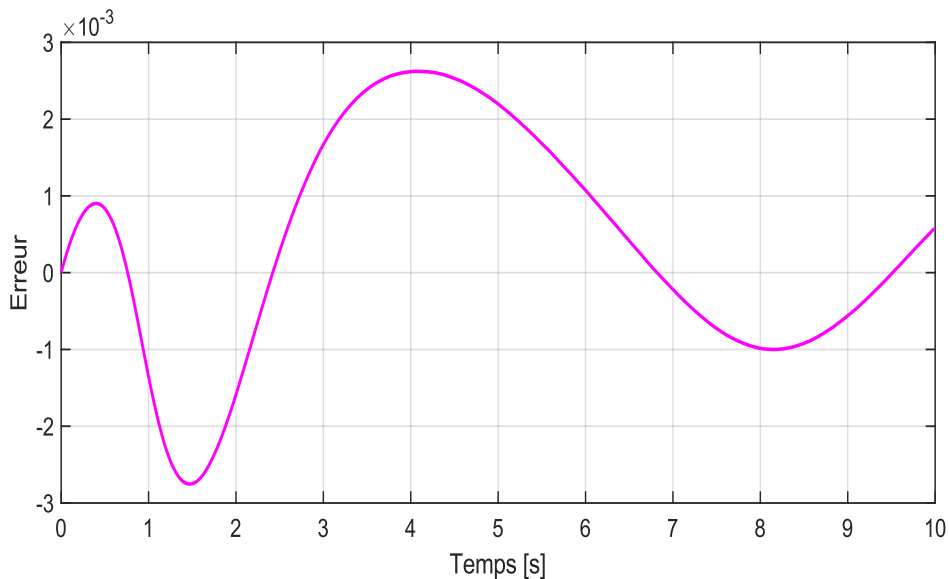


Figure III.9. Trajectoire de l'erreur de validation pour l'état x_2

Et enfin, La Figure III.10 illustre le portrait de phase dans le plan (x_1, x_2) à la fois pour le système réel et le modèle T-S pendant la validation. Les trajectoires convergent vers un point d'équilibre commun et

restent bornées, ce qui démontre visuellement la stabilité asymptotique globale. La grande similarité des trajectoires montre que le modèle non seulement capture les valeurs ponctuelles des états, mais préserve également le comportement dynamique qualitatif du système. Cette preuve graphique soutient la preuve de stabilité basée sur Lyapunov fournie dans l'article.

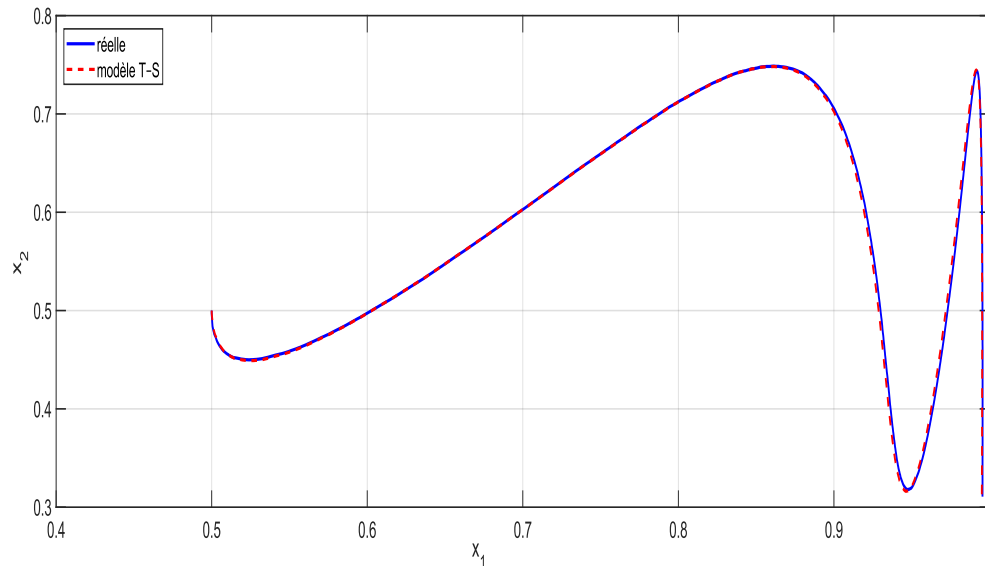


Figure III.10. Portrait de phase du système réel et du modèle T-S (Phase de validation)

Les résultats démontrent une convergence rapide des erreurs, une stabilité certifiée par l'analyse LMI, et une excellente généralisation avec des erreurs de validation minimales. Dans l'ensemble, le modèle reproduit efficacement la dynamique du système tout en préservant la stabilité, confirmant ainsi son adéquation pour des applications de contrôle avancées.

Bien que l'approche de modélisation floue de Takagi–Sugeno (T–S) offre un moyen efficace d'approximer les dynamiques non linéaires par le mélange fluide de modèles linéaires locaux via des fonctions d'appartenance gaussiennes, sa formulation dépend intrinsèquement de structures de règles et de fonctions d'appartenance prédéfinies. Pour élargir le cadre de modélisation et développer une représentation analytique plus générale, la section suivante introduit la méthode de Transformation Sectorielle Non-linéaire (NST). Cette approche reformule les relations non linéaires au sein de secteurs bornés pour générer un modèle polytopique convexe équivalent, permettant l'utilisation directe des techniques d'analyse et de contrôle linéaires pour les systèmes non linéaires et établissant une base adaptée aux applications de contrôle prédictif.

III.4. Transformation de secteur non linéaire et représentation polytopique

III.4.1. Principes de la transformation sectorielle non linéaire (NST)

La Transformation Sectorielle Non-linéaire (NST) est une approche systématique utilisée pour approximer un système dynamique non-linéaire par une structure polytopique équivalente (multi-modèle) adaptée à l'analyse et à la conception de contrôle dans le cadre de Takagi–Sugeno (T–S).

L'idée principale est de représenter la dynamique non linéaire à l'intérieur de secteurs bornés connus comme une combinaison convexe de modèles linéaires locaux [141].

III.4.2. Formulation générale des systèmes non linéaires

Considérons un système non linéaire général exprimé comme:

$$\dot{x}(t) = f(x(t), u(t)) \quad (\text{III.14})$$

Où $x(t) \in \mathbb{R}^n$ est le vecteur d'état et $u(t) \in \mathbb{R}^m$ est l'entrée de contrôle.

L'objectif est de trouver une représentation convexe de (III.14) qui puisse être traitée en utilisant la théorie du contrôle linéaire.

III.4.3. Hypothèses de bornage sectoriel des variables de planification

On suppose que les non-linéarités du système dépendent d'un ensemble de variables de planification $\rho_i(t)$, qui sont bornées dans des intervalles connus:

$$\rho_i(t) \in [\rho_i^{min}, \rho_i^{max}], \quad i = 1, \dots, p \quad (\text{III.15})$$

Chaque $\rho_i(t)$ peut être une fonction mesurable de l'état ou de l'entrée (par exemple, $\rho_i = x_j$ ou $\rho_i = u_k$). Ces variables définiront plus tard les sommets du modèle polytopique.

III.4.4. Représentation paramétrique dépendante des variables de planification

Le système non linéaire (III.14) peut souvent être exprimé sous une forme linéaire dépendante des paramètres:

$$\dot{x}(t) = A(\rho(t))x(t) + B(\rho(t))u(t) \quad (\text{III.16})$$

Où les matrices $A(\rho(t))$ et $B(\rho(t))$ varient de manière continue avec le vecteur de planification $\rho(t) = [\rho_1(t), \rho_2(t), \dots, \rho_p(t)]^T$.

III.4.5. Décomposition en modèles polytopiques convexes

En utilisant le principe de mélange convexe, (III.16) peut être réécrit comme une combinaison convexe de $N_v = 2^p$ systèmes de sommets linéaires:

$$A(\rho(t)) = \sum_{i=1}^{N_v} \mu_i(\rho(t))A_i \quad , \quad B(\rho(t)) = \sum_{i=1}^{N_v} \mu_i(\rho(t))B_i \quad (III.17)$$

Où:

- A_i, B_i : sont les matrices de sommets correspondant aux valeurs extrêmes des variables de planification
- $\mu_i(\rho(t))$: sont des fonctions de pondération (appartenance) satisfaisant les conditions de convexité.

$$\mu_i(\rho(t)) \geq 0 \quad , \quad \sum_{i=1}^{N_v} \mu_i(\rho(t)) = 1 \quad (III.18)$$

III.4.6. Construction des fonctions d'appartenance linéaires par secteur

Pour chaque variable de planification scalaire $\rho_j(t)$, deux fonctions de pondération linéaires sont définies sur son secteur (III.15):

$$\mu_{j1}(\rho_j) = \frac{\rho_j^{max} - \rho_j}{\rho_j^{max} - \rho_j^{min}} \quad , \quad \mu_{j2}(\rho_j) = 1 - \mu_{j1}(\rho_j) \quad (III.19)$$

Les fonctions de pondération globales $\mu_i(\rho(t))$ sont obtenues comme le produit tensoriel des fonctions individuelles:

$$\mu_i(\rho(t)) = \prod_{j=1}^p \mu_{j\sigma_j}(\rho_j(t)) \quad \sigma_j \in \{1,2\} \quad (III.20)$$

Chaque combinaison d'indices $(\sigma_1, \dots, \sigma_p)$ définit un sommet du modèle polytopique.

III.4.7. Modèle polytopique équivalent pour l'analyse et la commande

Substituer (III.17) dans (III.16) donne la représentation polytopique finale du système non linéaire:

$$\dot{x}(t) = \sum_{i=1}^{N_v} \mu_i(\rho(t))(A_i x(t) + B_i u(t)) \quad (III.21)$$

L'équation (III.21) décrit un modèle qui est linéaire à chaque sommet mais non linéaire dans le temps à travers les poids variables dans le temps $\mu_i(\rho(t))$. [140] , [143] .

III.4.8. Résultats numériques et validation par simulation

Pour illustrer la procédure de Transformation Sectorielle Non Linéaire (NST), considérons à nouveau le système dynamique non linéaire précédemment introduit dans l'équation (III.1):
$$\begin{cases} \dot{x}_1 = -ux_1 + u \\ \dot{x}_2 = -x_2x_1 + u \end{cases}$$

Dans ce modèle, les non-linéarités dépendent explicitement à la fois de l'entrée de contrôle u et de la variable d'état x_2 . Par conséquent, ces deux quantités sont sélectionnées comme variables de planification, qui sont supposées évoluer dans des intervalles bornés connus $u(t) \in [a_1, a_2], x_2(t) \in [b_1, b_2]$

Où les limites du secteur sont définies comme: $[a_1, a_2] = [0.2, 1.0], [b_1, b_2] = [0.5, 1.6]$

A. Matrices du système de sommets (local)

L'approche NST consiste à approximer le système non linéaire par une combinaison convexe de modèles linéaires locaux obtenus aux valeurs limites des variables de planification. Évaluer le système aux limites du secteur conduit à quatre modèles de sommet (locaux) définis par:

$$A_1 = \begin{bmatrix} -a_1 & 0 \\ -b_1 & 0 \end{bmatrix}, A_2 = \begin{bmatrix} -a_1 & 0 \\ -b_2 & 0 \end{bmatrix}, A_3 = \begin{bmatrix} -a_2 & 0 \\ -b_1 & 0 \end{bmatrix}, A_4 = \begin{bmatrix} -a_2 & 0 \\ -b_2 & 0 \end{bmatrix}, B = \begin{bmatrix} 1 \\ 1 \end{bmatrix}. \quad (III.22)$$

B. Fonctions d'appartenance (de mélange):

À chaque instant, l'influence relative de chaque modèle local est déterminée par des fonctions d'appartenance (ou de mélange) qui dépendent des valeurs actuelles de $u(t)$ et $x_2(t)$. Ces fonctions sont définies comme suit:

$$\begin{aligned} \mu_{11} &= \frac{a_2 - u}{a_2 - a_1}, \mu_{12} = 1 - \mu_{11} \\ \mu_{21} &= \frac{b_2 - x_2}{b_2 - b_1}, \mu_{22} = 1 - \mu_{21} \end{aligned} \quad (III.23)$$

Et les pondérations globales basées sur les règles, correspondant aux quatre sommets, sont calculées comme suit:

$$\mu_i = \{\mu_{11}\mu_{21}, \mu_{11}\mu_{22}, \mu_{12}\mu_{21}, \mu_{12}\mu_{22}\}, \sum_{i=1}^4 \mu_i = 1. \quad (III.24)$$

C. Représentation efficace du système:

La matrice de système effective à chaque instant est obtenue comme une combinaison convexe des matrices de sommet, pondérées par leurs fonctions d'appartenance correspondantes:

$$A_{\text{eff}}(t) = \sum_{i=1}^4 \mu_i(t) A_i \quad (III.25)$$

Par conséquent, le modèle polytopique global (de type Takagi-Sugeno) du système non linéaire peut être exprimé comme suit:

$$\dot{x}(t) = A_{\text{eff}}(t)x(t) + Bu(t) \quad (\text{III.26})$$

Où $A_{\text{eff}}(t)$ varie continuellement en fonction du point de fonctionnement actuel $(u(t), x_2(t))$.

Cette représentation offre une interpolation lisse et convexe entre les modèles linéaires locaux, permettant l'utilisation de techniques de conception de contrôle linéaire telles que la Compensation Distribuée Parallèle (PDC) ou le Contrôle Prédictif par Modèle (MPC) tout en conservant les caractéristiques non linéaires essentielles du système original.

D. Scénarios de simulation pour validation:

La simulation a été réalisée dans l'environnement MATLAB pour évaluer la précision du modèle polytopique basé sur NST proposé dans la reproduction des dynamiques du système non linéaire. Le temps total de simulation a été fixé à $T = 10\text{s}$ avec une période d'échantillonnage de $\Delta t = 0.01\text{ s}$.

Le système a été initialisé à $x(0) = [0.5 \quad 0.5]^T$, et les secteurs dépendants de l'entrée et de l'état ont été définis comme: $u(t) = [a_1, a_2] = [-1, 1]$, $x_2(t) = [b_1, b_2] = [-1, 1.6]$

Ces bornes ont été utilisées pour déterminer les limites de secteur pour les termes non linéaires et pour construire la représentation polytopique correspondante (de type T-S).

Pour évaluer la précision et la capacité de généralisation du modèle polytopique basé sur le NST, deux scénarios d'entrée différents ont été considérés. Ces entrées ont été conçues pour tester la performance du modèle sous des conditions d'excitation à la fois discontinues et lisses, couvrant ainsi une large gamme de comportements dynamiques.

Cas 1: Entrée par morceaux ou non lisse:

Dans le premier scénario, un signal discontinu a été appliqué pour exciter les dynamiques non linéaires à travers des variations brusques. L'entrée combine une forme d'onde carrée avec une petite perturbation aléatoire et est exprimée comme suit:

$$u_1(t) = 0.8 + 0.2\text{square}(0.5\pi) + 0.01\eta(t) \quad (\text{III.27})$$

Où $\eta(t)$ désigne un bruit aléatoire de faible amplitude.

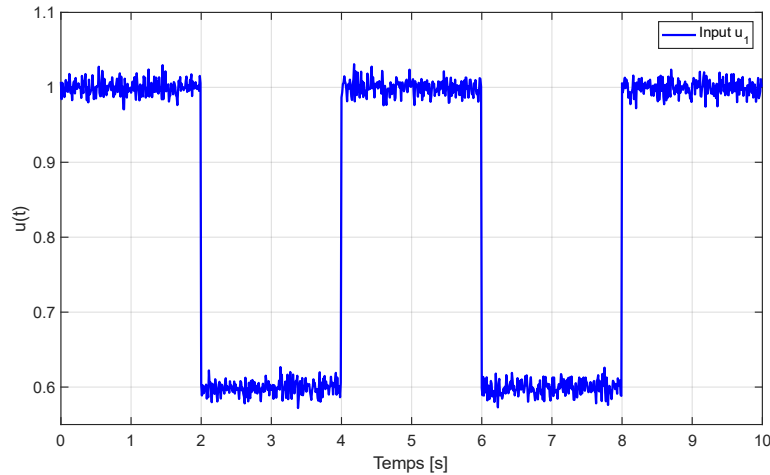


Figure III.11. Signal d'entrée (Cas 1).

Un tel signal est délibérément choisi pour tester la capacité du modèle polytopique basé sur NST à reproduire des commutations abruptes entre des modèles linéaires locaux. Il excite la dynamique non linéaire complète au sein des secteurs de fonctionnement définis, fournissant un benchmark rigoureux pour le mécanisme de mélange convexe.

La Figure III.12 compare l'évolution temporelle de la première variable d'état obtenue à partir du système non linéaire réel (courbe bleue) et du modèle polytopique basé sur le NST (courbe rouge en pointillés). Le chevauchement étroit entre les deux courbes confirme leur excellent accord, même sous une excitation d'entrée discontinue.

Des déviations transitoires mineures apparaissent lors de transitions d'entrée brusques, mais l'erreur quadratique moyenne (4.7905×10^{-6}) reste extrêmement petite, validant l'approximation de haute précision.

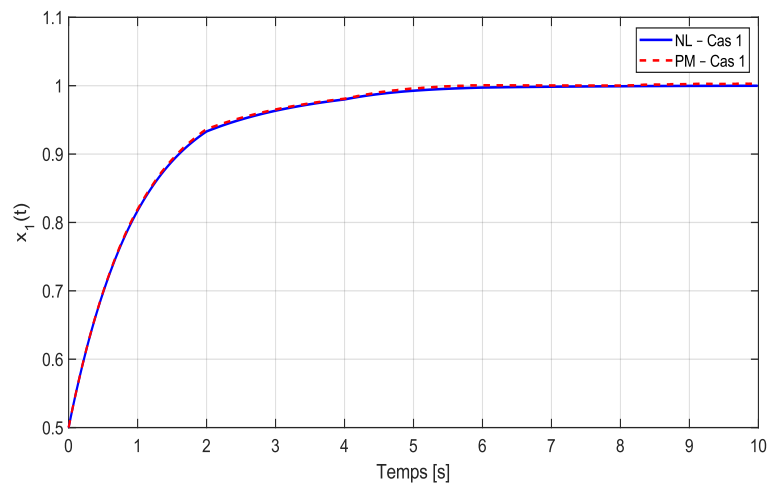


Figure III.12. Comparaison des trajectoires de $x_1(t)$ entre le système non linéaire et le modèle polytopique (Cas 1)

Et en suite, La Figure III.13 montre que la deuxième variable d'état présente la même correspondance de haute fidélité entre les modèles non linéaires et polytopiques. Cela confirme que le mélange convexe des modèles locaux capture avec succès le fort couplage entre $x_1(t)$ et $x_2(t)$, même lorsque l'entrée contient des discontinuités abruptes. L'erreur quadratique moyenne est de (2.9058×10^{-6}) .

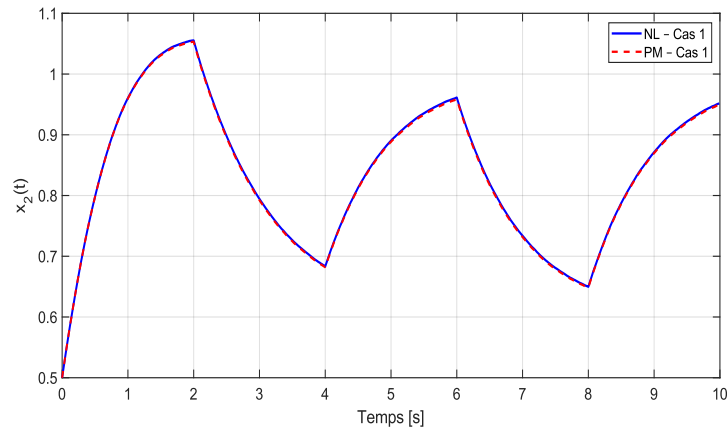


Figure III.13. Comparaison des trajectoires de $x_2(t)$ entre le système non linéaire et le modèle polytopique (Cas 1)

La Figure III.14 présente les trajectoires dans le plan de phase du système non linéaire (en bleu) et du modèle polytopique basé sur NST (en rouge pointillé) sous excitation discontinue. Les trajectoires presque coïncidentes indiquent que le modèle non seulement reproduit les valeurs d'état ponctuelles, mais préserve également le comportement dynamique qualitatif et les caractéristiques de stabilité globale du système réel.

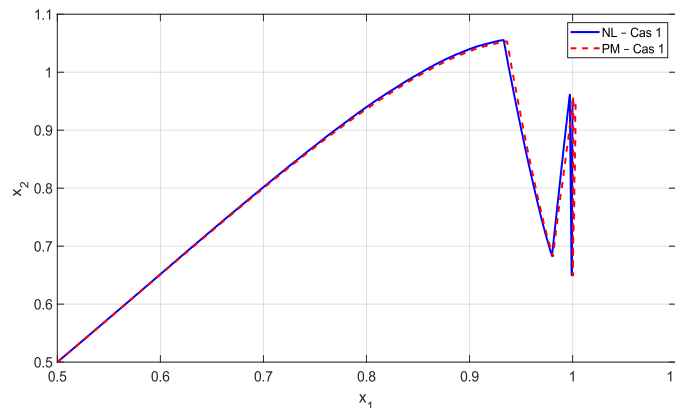


Figure III.14. Portrait de phase dans le plan (x_1, x_2) (Cas 1)

Et la Figure III.15 illustre l'évolution temporelle des quatre fonctions d'appartenance $\mu_i(t)$, calculées à partir des variables de planification $u(t)$ et $x_2(t)$. Chaque fonction représente le degré d'activation d'un modèle linéaire local, satisfaisant les conditions de convexité: $\sum_{i=1}^4 \mu_i = 1$ et $\mu_i \geq 0$. Les transitions

douces entre les courbes indiquent un passage progressif entre les modèles, assurant ainsi la stabilité numérique et empêchant les discontinuités dans la matrice effective du système $A_{\text{eff}}(t)$.

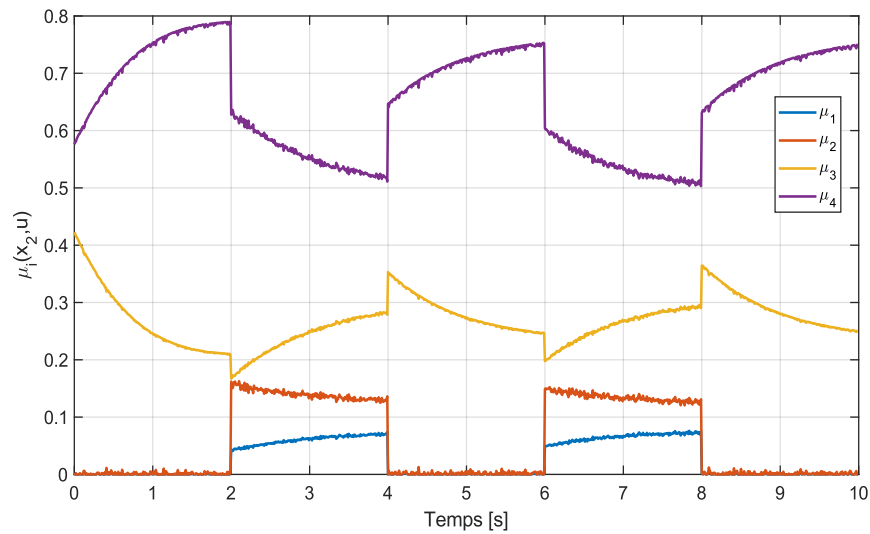


Figure III.15. Évolution de la fonction d'appartenance $\mu_i(x_2, u)$ pendant la simulation (Cas 1)

Cas 2: Entrée sinusoïdale lisse:

Dans le deuxième scénario, un signal d'excitation lisse composé de multiples composantes sinusoïdales avec des fréquences différentes a été utilisé. L'entrée est donnée par:

$$u_2(t) = 0.6 \sin(0.3t) + 0.4 \sin(t) + 0.01 \times \eta(t) \quad (\text{III.28})$$

La Figure III.16 illustre le deuxième signal d'entrée, un signal composite lisse contenant plusieurs composantes de fréquence. Il est utilisé pendant la phase de validation pour évaluer la capacité d'interpolation du modèle et sa précision dans le suivi du comportement non linéaire continu.

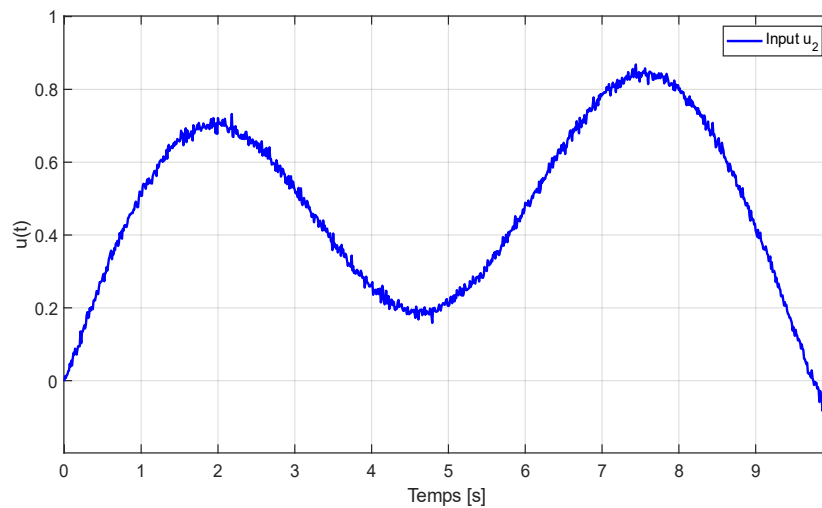


Figure III.16. Signal d'entrée (Cas 2)

La Figure III.17 montre que, sous une excitation sinusoïdale douce, le modèle polytopique reproduit presque parfaitement la trajectoire du premier état du système non linéaire. L'erreur quadratique moyenne extrêmement petite de 1.68×10^{-32} démontre un comportement numérique presque identique et confirme que la transformation NST préserve efficacement la dynamique non linéaire continue à travers des conditions de fonctionnement non observées.

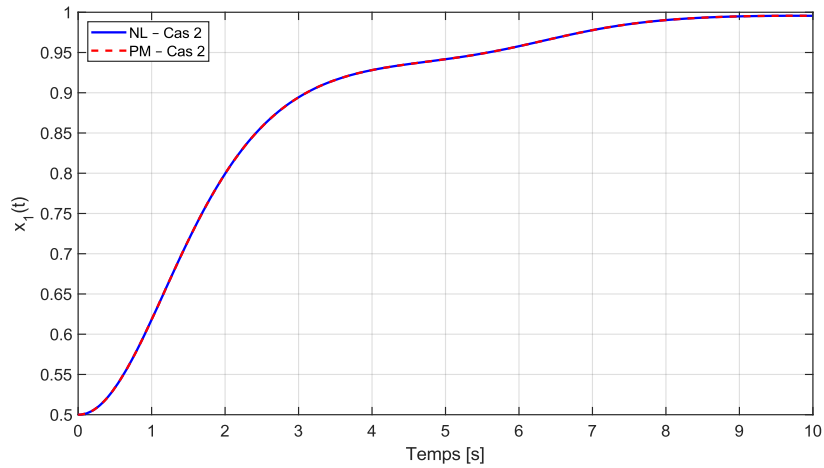


Figure III.17. Comparaison des trajectoires de $x_1(t)$ entre le système non linéaire et le modèle polytopique (Cas 2)

La Figure III.18 montre que la réponse du deuxième état présente la même concordance quasi parfaite, indiquant que la représentation polytopique se généralise avec précision au-delà du scénario d'entraînement. Cette cohérence confirme que l'interpolation convexe parmi les matrices de sommet A_i conserve la structure de couplage non linéaire sans aucun réajustement des paramètres. L'erreur quadratique moyenne est de (3.7717×10^{-33}) .

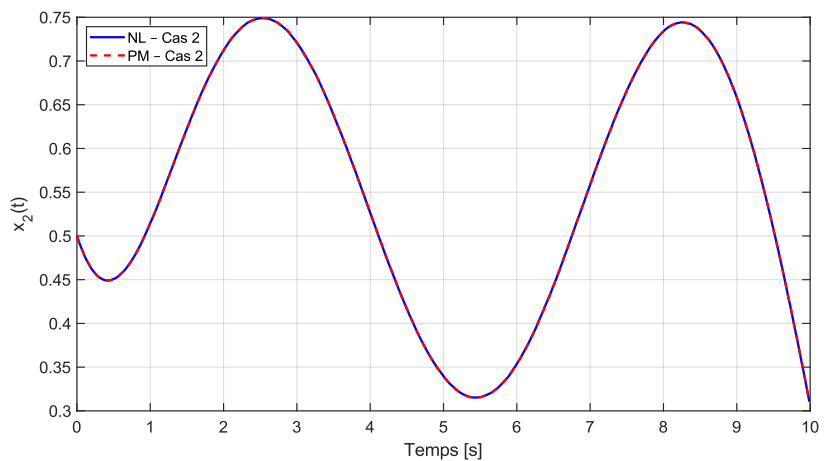


Figure III.18. Comparaison des trajectoires de $x_2(t)$ entre le système non linéaire et le modèle polytopique (Cas 2)

La Figure III.19 montre que, pour le cas d'entrée lisse, les trajectoires convergent vers un même point d'équilibre. Le chevauchement quasi parfait entre les trajectoires issues du modèle non linéaire et celles du modèle polytopique met en évidence la qualité de l'approximation fournie par la transformation NST et confirme la capacité du modèle polytopique à reproduire fidèlement la dynamique du système original.

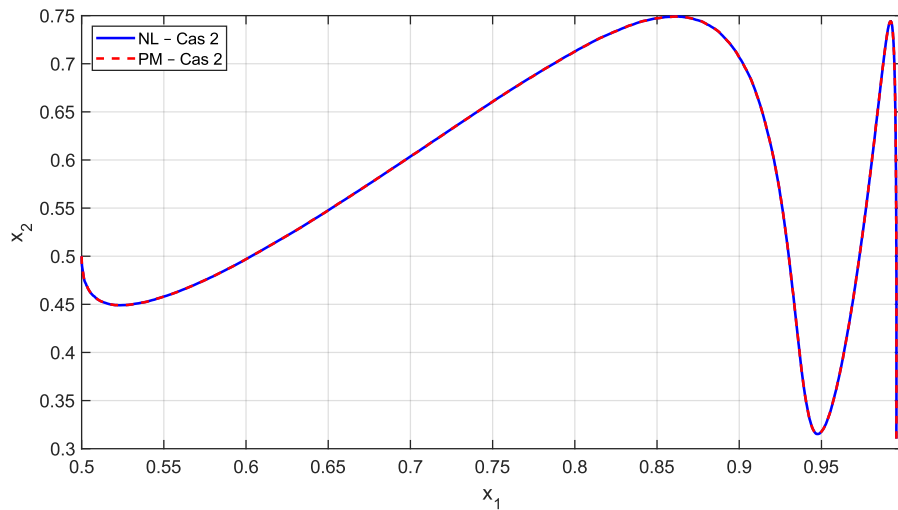


Figure III.19. Portrait de phase dans le plan (x_1, x_2) pour le Cas 2

La Figure III.20 montre les niveaux d'activation des quatre fonctions d'appartenance lors de la validation avec le signal d'entrée lisse. Les variations symétriques et continues de $\mu_i(t)$ confirment la stabilité du processus de mélange convexe et démontrent que le modèle basé sur le NST s'adapte de manière transparente aux différentes conditions de fonctionnement.

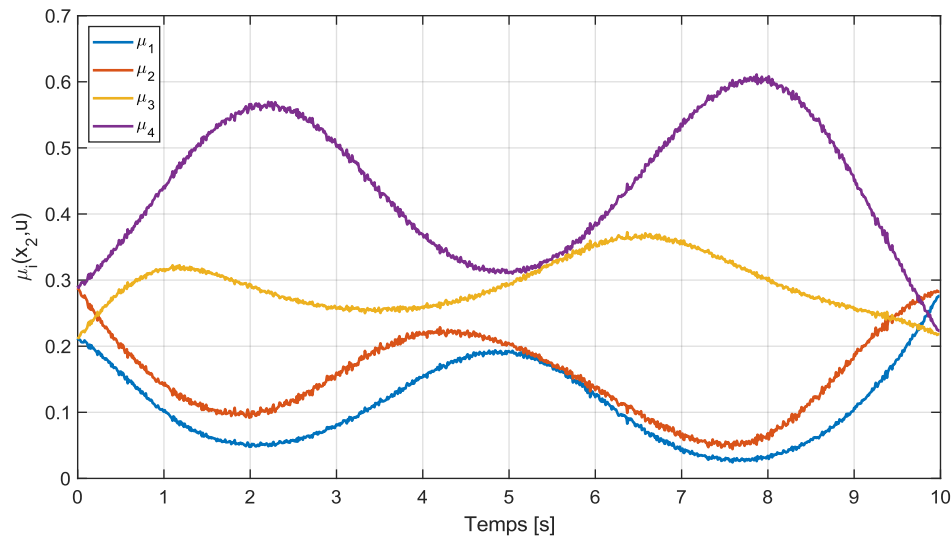


Figure III.20. Évolution des fonctions d'appartenance (Cas 2)

Ensemble, ces deux scénarios d'entrée fournissent un cadre de validation complet, garantissant que le modèle polytopique basé sur NST fonctionne de manière cohérente à la fois dans des conditions de commutation rapide et de dynamique fluide.

III.4.9. Avantages, limitations et perspectives d'application de l'approche NST

La transformation polytopique basée sur le NST offre un pont puissant entre la conception de contrôle non linéaire et linéaire en:

- Permettre l'utilisation de techniques basées sur les LMI et le MPC sur des systèmes non linéaires,
- Fournir un cadre convexe pour l'analyse de la stabilité et des performances,
- Permettre une approximation précise dans les limites du secteur défini.

Bien que la représentation NST polytopique fournisse un modèle très précis et analytiquement transparent adapté à la conception de contrôle, sa dérivation repose sur une connaissance préalable des fonctions non linéaires et de leurs bornes sectorielles. Pour améliorer la flexibilité et l'adaptabilité du cadre de modélisation en utilisant des données empiriques, la section suivante présente une approche d'identification de modèle Takagi–Sugeno (T–S) basée sur l'algorithme d'optimisation de Levenberg–Marquardt (LM). Cette méthodologie affine à la fois les composantes structurelles et paramétriques du modèle par une optimisation numérique itérative, améliorant ainsi la précision prédictive et permettant une généralisation au-delà des conditions analytiquement dérivées.

III.5. Identification de modèle flou Takagi–Sugeno basée sur les données en utilisant l'algorithme de Levenberg–Marquardt

Cette section présente la procédure générale pour identifier un modèle flou de Takagi–Sugeno (T–S) en utilisant l'algorithme d'optimisation de Levenberg–Marquardt (LM) basé sur des données obtenues à partir d'un système dynamique non linéaire. L'objectif est de construire un modèle flou qui approxime avec précision le comportement non linéaire en combinant des modèles linéaires locaux par inférence floue et optimisation numérique des paramètres [140].

III.5.1. Acquisition et préparation des données d'entraînement

La première étape consiste à générer un ensemble de données d'entraînement qui capture la relation entre les variables d'état du système, le signal d'entrée et les dérivées temporelles correspondantes des états. Ces données peuvent être obtenues soit par simulation numérique, soit par des mesures

expérimentales sous différentes conditions de fonctionnement pour garantir que le modèle représente toute la plage dynamique du système.

III.5.2. Architecture du modèle flou T–S et fonctions d'appartenance gaussiennes

Le modèle flou de Takagi–Sugeno représente la dynamique du système comme une combinaison convexe de plusieurs sous-modèles linéaires locaux. Chaque règle floue correspond à un modèle local et prend la forme générale suivante:

$$\dot{x}(t) = A_i x(t) + B_i u(t) \quad (III.29)$$

Où A_i et B_i sont les matrices de modèle local associées à la i -ème règle floue.

Le degré d'activation de chaque règle est défini par une fonction d'appartenance gaussienne:

$$\mu_i(z) = \exp\left[-\frac{1}{2}\left(\frac{z-c_i}{v_i}\right)^2\right] \quad (III.30)$$

Où z désigne la variable de planification, et c_i , v_i sont le centre et la largeur (variance) de la fonction gaussienne, respectivement.

La sortie globale du modèle est obtenue comme suit:

$$\dot{x}(t) = \sum_{i=1}^{n_r} \bar{\mu}_i(z) [A_i x(t) + B_i u(t)] \quad (III.31)$$

Où $\bar{\mu}_i = \frac{\mu_i}{\sum_j \mu_j}$ et n_r est le nombre total de règles floues [138].

III.5.3. Initialisation des paramètres du modèle

Tous les paramètres du modèle sont regroupés dans un seul vecteur de paramètres contenant les éléments associés à chaque règle:

$$\theta = [a_1, a_2, a_3, a_4, b_1, b_2, c, v]_i. \quad (III.32)$$

Les paramètres sont initialisés soit aléatoirement, soit en utilisant des connaissances préalables sur le système pour garantir une diversité suffisante entre les règles et améliorer la convergence pendant l'entraînement.

III.5.4. Procédure d'optimisation de Levenberg-Marquardt

Le processus d'identification consiste à minimiser la différence entre les dérivées d'état mesurées (ou simulées) et celles prédites par le modèle flou. Cette optimisation est réalisée à l'aide de l'algorithme de Levenberg-Marquardt, une méthode hybride combinant les avantages des approches de descente de

gradient et de Gauss-Newton. Il offre une convergence rapide et une stabilité numérique robuste. La fonction de coût est définie comme suit:

$$J(\theta) = \sum_{k=1}^N \| \dot{x}_{\text{data}}(k) - \dot{x}_{\text{model}}(k; \theta) \|^2 \quad (\text{III.33})$$

Et le vecteur de paramètres θ est mis à jour de manière itérative pour minimiser le coût jusqu'à ce que la convergence soit atteinte.

III.5.5. Critères d'évaluation et validation des performances

Après la phase d'entraînement, le modèle flou identifié est utilisé pour reproduire le comportement du système dans les mêmes conditions de fonctionnement utilisées pour la collecte des données. La réponse du modèle est comparée aux données de référence pour évaluer la précision du processus d'identification et évaluer la capacité du modèle à approximer les dynamiques non linéaires dans la plage considérée.

III.5.6. Résultats numériques et validation par simulation

Cet exemple illustre l'identification pratique d'un modèle flou de Takagi–Sugeno (T–S) à partir de données générées par un système dynamique non linéaire. Le flux de travail comprend (1) la génération de données d'entraînement, (2) la structure du modèle et la paramétrisation, (3) la formulation du coût d'identification et l'optimisation avec l'algorithme de Levenberg-Marquardt, et (4) la validation du modèle identifié.

A. Configuration de la simulation et génération des données

Les données d'entraînement sont générées en simulant le système non linéaire (inconnu de l'identificateur) sur un horizon fini. Soit l'intervalle d'échantillonnage $\Delta t = 0.01$ s et le temps total de simulation $T = 50$ s, ce qui donne $N = T/\Delta t$ échantillons. Une séquence d'entrée $u(t)$ est conçue pour exciter le système sur une large gamme de fréquences, incluant une somme de sinusoides par morceaux, des signaux en dents de scie, carrés et chirp (Éq. III.12, Figure III.1). La trajectoire d'état $x(t) \in \mathbb{R}^{n_x}$ est enregistrée et les dérivées temporelles $\dot{x}(t)$ sont calculées numériquement pour former l'ensemble d'entraînement:

$$\mathcal{D} = \{(x(k), u(k), \dot{x}(k))\}_{k=1}^N \quad (\text{III.34})$$

B. Paramétrisation du modèle flou T–S

Un modèle flou T–S avec n_r règles approxime la dynamique comme une combinaison convexe de sous-modèles linéaires locaux:

$$\dot{x}(t) = \sum_{i=1}^{n_r} \bar{\mu}_i(z(t))(A_i x(t) + B_i u(t)) \quad (III.35)$$

Où chaque modèle local est paramétré par $A_i \in \mathbb{R}^{n_x \times n_x}$ et $B_i \in \mathbb{R}^{n_x \times n_u}$. La force de tir normalisée est:

$$\bar{\mu}_i(z) = \frac{\mu_i(z)}{\sum_{j=1}^{n_r} \mu_j(z)}, \quad \mu_i(z) = \exp\left(-\frac{1}{2} \left(\frac{z-c_i}{v_i}\right)^2\right) \quad (III.36)$$

Avec z la variable de planification (ici choisie comme le premier état) et c_i, v_i le centre et la largeur gaussiens pour la règle i .

C. Définition du vecteur de paramètres et stratégie d'initialisation

Toutes les inconnues sont empilées dans un seul vecteur de paramètres θ . Pour un exemple à deux états ($n_x = 2$) et une entrée scalaire ($n_u = 1$), chaque règle contient huit paramètres:

$$\theta_i = [a_1^{(i)}, a_2^{(i)}, a_3^{(i)}, a_4^{(i)}, b_1^{(i)}, b_2^{(i)}, c_i, v_i]^T \quad (III.37)$$

De sorte que $A_i = \begin{bmatrix} a_1^{(i)} & a_2^{(i)} \\ a_3^{(i)} & a_4^{(i)} \end{bmatrix}$ et $B_i = [b_1^{(i)} b_2^{(i)}]^T$. Le vecteur complet des paramètres est $\theta = [\theta_1^T, \dots, \theta_{n_r}^T]^T$.

Les choix typiques utilisés dans l'exemple sont $n_r = 5$ et l'initialisation aléatoire (ou informée) de θ avec des centres c_i répartis sur la plage observée de la variable de planification.

D. Formulation du problème d'optimisation et paramètres du solveur

L'identification minimise l'erreur quadratique entre les dérivées mesurées et les dérivées prédites par le modèle. Pour N points de données, définissons le résidu à l'échantillon k :

$$r(k; \theta) = \dot{x}_{\text{data}}(k) - \dot{x}_{\text{model}}(k, \theta) \quad (III.38)$$

Et collecter tous les résidus dans un vecteur $r(\theta) \in \mathbb{R}^{n_x N}$. L'objectif des moindres carrés est:

$$J(\theta) = \|r(\theta)\|_2^2 \quad (III.39)$$

L'optimisation est effectuée avec l'algorithme de Levenberg-Marquardt (une variante de région de confiance combinant Gauss-Newton et descente de gradient), implémenté via un solveur standard des moindres carrés. Exemple de paramètres de solveur utilisés dans l'expérience:

- ✓ Nombre maximal d'itérations: 100
- ✓ Tolérance de fonction: 10^{-9}
- ✓ Tolérance de pas: 10^{-9}
- ✓ Maximum function evaluations: 20000.

Le solveur met à jour θ de manière itérative jusqu'à la convergence vers un minimum local de $J(\theta)$.

E. Modèles locaux identifiés:

Après l'optimisation de Levenberg--Marquardt, cinq sous-modèles linéaires locaux ont été identifiés, chacun correspondant à une règle floue dans le modèle de Takagi--Sugeno (TS). Chaque règle est décrite par un modèle local d'espace d'état et sa fonction d'appartenance gaussienne associée, comme précédemment défini dans les équations (III.29) et (III.30):

$$\dot{x}(t) = A_i x(t) + B_i u(t) \quad , \quad \mu_i(z) = \exp \left[-\frac{1}{2} \left(\frac{z - c_i}{v_i} \right)^2 \right]$$

Où $A_i \in \mathbb{R}^{2 \times 2}$ et $B_i \in \mathbb{R}^{2 \times 1}$ sont les matrices de système et d'entrée de la règle i , tandis que c_i et v_i désignent le centre et la largeur de la fonction d'appartenance gaussienne correspondante. Les paramètres identifiés sont listés ci-dessous:

Tableau III.2. Paramètres optimisés pour le modèle flou (5 règles)

Règle	Matrice A_i	Matrice B_i	Centre c_i	Variance σ_i
1	$\begin{bmatrix} -0.0064 & 0.3721 \\ -0.2727 & -2.4275 \end{bmatrix}$	$\begin{bmatrix} -1.3577 \\ 0.7189 \end{bmatrix}$	1.3613	0.4928
2	$\begin{bmatrix} -0.1628 & 1.1462 \\ -0.4113 & 0.0686 \end{bmatrix}$	$\begin{bmatrix} 3.0514 \\ 0.7802 \end{bmatrix}$	0.1404	0.8264
3	$\begin{bmatrix} 0.0590 & -0.7278 \\ 0.3915 & -0.6386 \end{bmatrix}$	$\begin{bmatrix} -0.4098 \\ 1.3105 \end{bmatrix}$	1.1722	0.3944
4	$\begin{bmatrix} 0.1030 & -0.9210 \\ 0.4103 & -0.1966 \end{bmatrix}$	$\begin{bmatrix} -0.6573 \\ 1.2352 \end{bmatrix}$	0.0568	0.3543
5	$\begin{bmatrix} 0.0903 & -0.5562 \\ 0.1694 & -0.7234 \end{bmatrix}$	$\begin{bmatrix} -0.6573 \\ 1.2352 \end{bmatrix}$	0.6015	0.3255

Chaque modèle local caractérise la dynamique du système dans une région de fonctionnement spécifique déterminée par la variable de planification $z = x_1$. La combinaison de ces modèles locaux à travers leurs fonctions d'appartenance normalisées forme le modèle flou Takagi--Sugeno identifié complet, qui représente avec précision le comportement non linéaire du système.

F. Validation sur les données d'entraînement:

Avant de passer à la phase de validation, il est essentiel d'évaluer dans quelle mesure le modèle Takagi-Sugeno identifié reproduit le comportement du système lorsqu'il est soumis au même signal d'entrée utilisé pendant la phase d'entraînement. Ce test fournit une évaluation initiale de la précision d'ajustement du modèle et de sa capacité à capturer les caractéristiques dynamiques intégrées dans les données d'entraînement. En comparant les trajectoires de référence avec celles prédites par le modèle, la proximité entre le modèle identifié et le système réel sous des conditions d'excitation identiques peut être évaluée.

La Figure III.21 montre l'évolution temporelle de la première variable d'état $x_1(t)$ pendant la phase d'entraînement. La ligne continue correspond au véritable système non linéaire, tandis que la ligne pointillée représente la sortie du modèle flou Takagi-Sugeno (T-S) identifié lorsqu'il est entraîné par la même entrée d'entraînement. Le chevauchement étroit entre les deux courbes démontre que le modèle flou capture avec précision le comportement dynamique du système original dans la région de fonctionnement couverte par les données d'entraînement.

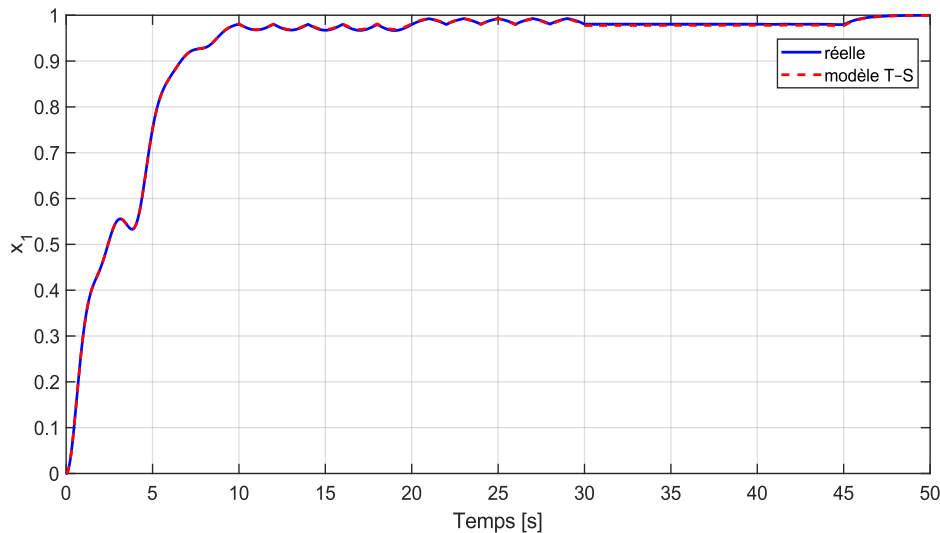


Figure III.21. Évolution temporelle de l'état x_1 pour le système non linéaire réel et le modèle Takagi-Sugeno identifié (phase d'entraînement)

La figure III.22 présente l'évolution temporelle du deuxième état $x_2(t)$ sous l'entrée d'entraînement. Similaire à la Figure III.21, la comparaison entre le système réel et le modèle T-S identifié confirme la capacité du modèle à reproduire les dynamiques non linéaires pour cet état également. La déviation minimale entre les deux trajectoires indique un ajustement réussi des relations d'espace d'état pendant l'identification.

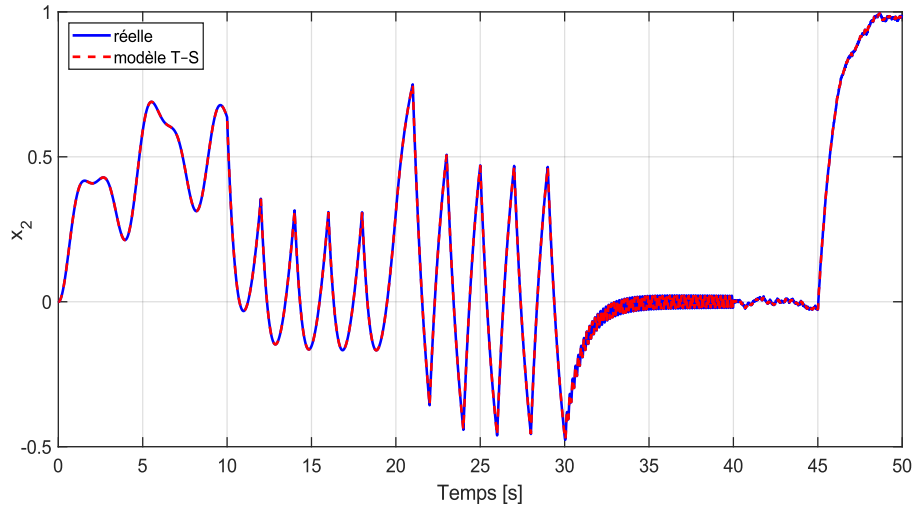


Figure III.22. Évolution temporelle de l'état x_2 pour le système non linéaire réel et le modèle Takagi–Sugeno identifié (phase d'entraînement).

La Figure III.23 compare les portraits de phase (x_1, x_2) du véritable système non linéaire et du modèle flou identifié. Les trajectoires presque coïncidentes dans le plan d'état illustrent que le modèle T–S préserve à la fois l'amplitude et la géométrie qualitative de la réponse dynamique du système. Cet accord visuel valide que le modèle capture avec précision le couplage non linéaire sous-jacent entre les deux états.

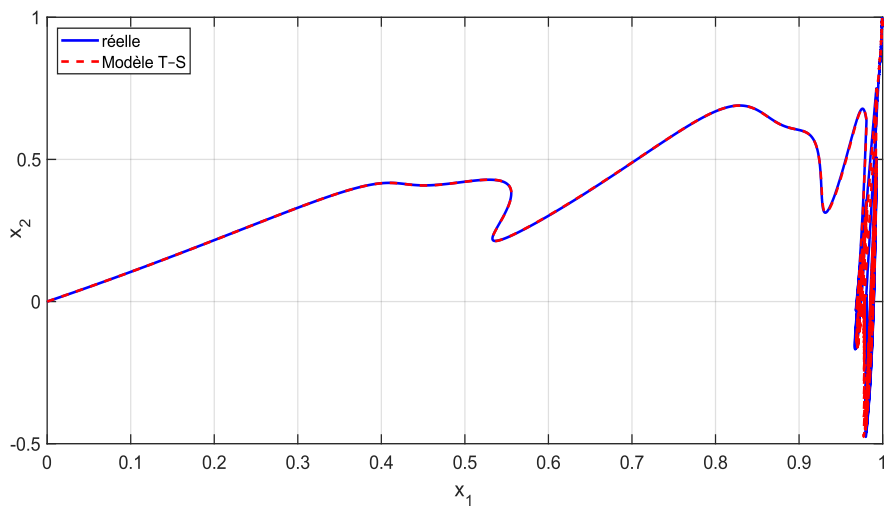


Figure III.23. Comparaison des portraits de phase du système non linéaire réel et du modèle flou Takagi–Sugeno identifié pendant l'entraînement.

La Figure III.24 affiche les fonctions d'appartenance gaussiennes associées aux règles floues identifiées pendant l'entraînement. Chaque courbe représente le degré d'activation d'une règle en fonction de la variable de planification $z = x_1$. La distribution et le chevauchement de ces fonctions garantissent une

interpolation fluide entre les sous-modèles linéaires locaux, permettant au système flou T–S d'approximer continuellement le comportement non linéaire sur toute la plage de fonctionnement.

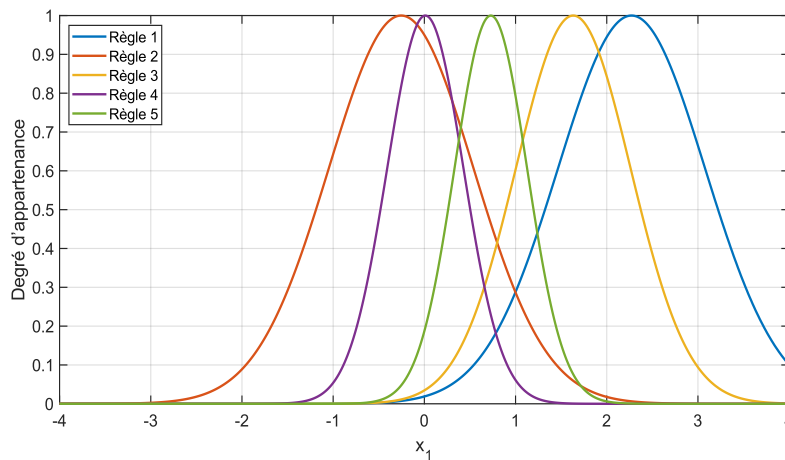


Figure III.24. Fonctions d'appartenance gaussiennes des règles floues (phase d'entraînement).

Les résultats obtenus en utilisant l'entrée d'entraînement indiquent que le modèle Takagi–Sugeno identifié peut reproduire avec précision la dynamique du système dans la plage de fonctionnement couverte pendant l'entraînement. Cependant, pour garantir la capacité de généralisation du modèle et sa performance fiable dans des conditions non observées, un test de validation indépendant est effectué en utilisant un nouveau signal d'entrée différent de celui utilisé lors de la phase d'entraînement.

G. Validation sur des données indépendantes et analyse de généralisation:

L'expérience de validation est menée pour évaluer la capacité de généralisation du modèle flou Takagi–Sugeno (T–S) identifié dans de nouvelles conditions de fonctionnement. Un signal d'entrée de validation, différent de celui utilisé pendant l'entraînement, est appliqué pour exciter le système sur une plage dynamique similaire. La simulation est effectuée sur un horizon temporel fixe avec le même pas d'échantillonnage que lors de la phase d'entraînement, garantissant ainsi la cohérence de la résolution des données. Pendant la validation, la variable x_1 est sélectionnée comme variable de planification utilisée pour calculer les degrés d'appartenance de chaque règle floue, selon la formulation gaussienne précédemment définie en (III.36). Les dynamiques globales du système ont ensuite été reconstruites en suivant la structure T–S de (III.35), en mélangeant les modèles locaux identifiés à travers leurs degrés d'appartenance normalisés.

Cette structure permet une interpolation fluide entre les modèles locaux, permettant au système flou T–S d'approximer avec précision les dynamiques non linéaires du processus original sous de nouvelles conditions d'excitation.

La Figure III.25 illustre le signal d'entrée de validation appliqué pour tester la capacité de généralisation du modèle T-S identifié. Ce signal diffère de l'entrée d'entraînement et est composé d'une combinaison de composantes sinusoïdales avec du bruit ajouté:

$$u_{\text{test}}(t) = 0.6\sin(0.3t) + 0.4\sin(4t) + 0.01\text{randn}(t) \quad (\text{III.40})$$

Il excite le système sur une large gamme de fréquences pour évaluer la performance du modèle dans des conditions de fonctionnement jamais vues auparavant.

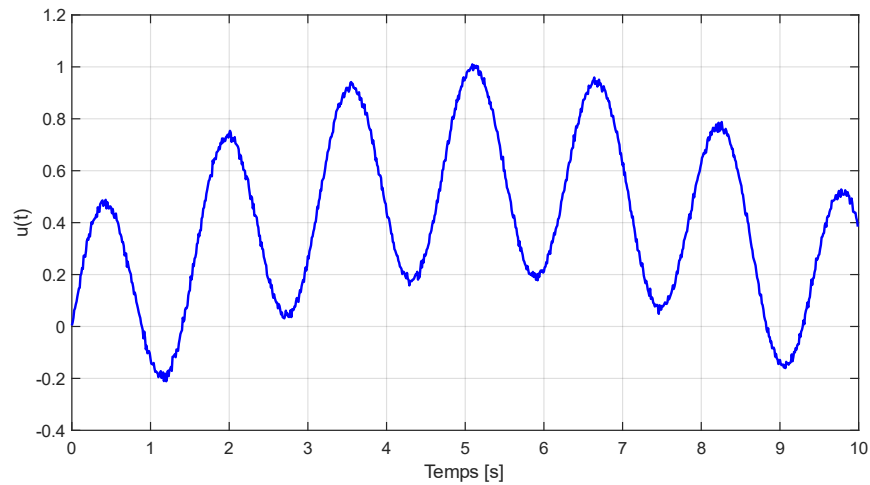


Figure III.25. Signal d'entrée de validation $u_{\text{test}}(t)$ utilisé pour évaluer le modèle identifié.

La Figure III.26 compare la première trajectoire d'état $x_1(t)$ obtenue à partir du système réel avec celle prédite par le modèle T-S sous une nouvelle entrée de validation. La correspondance étroite entre les deux signaux démontre la capacité du modèle à généraliser au-delà des données d'entraînement et à reproduire avec précision la dynamique du système dans des scénarios non vus.

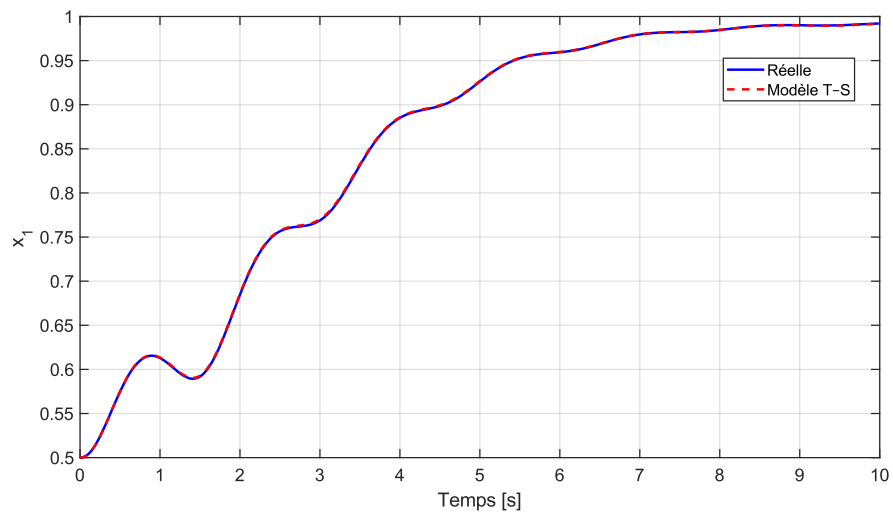


Figure III.26. Évolution temporelle de l'état x_1 pour le système non linéaire réel et le modèle Takagi-Sugeno identifié (phase de validation).

La Figure III.27 fournit le même type de comparaison pour le deuxième état $x_2(t)$. Encore une fois, le modèle flou T–S identifié suit de près la réponse réelle du système, confirmant que le modèle maintient la cohérence et la robustesse sous différentes excitations d'entrée.

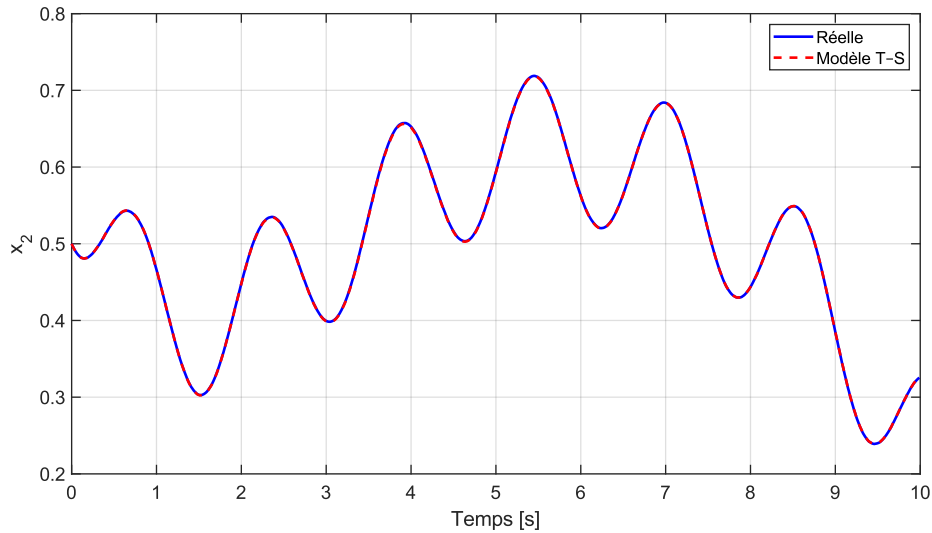


Figure III.27. Évolution temporelle de l'état x_2 pour le système non linéaire réel et le modèle Takagi–Sugeno identifié (phase de validation).

La Figure III.28 représente l'erreur d'estimation $e_{x_1}(t) = x_{1,true}(t) - x_{1,model}(t)$ au fil du temps pendant la validation. La petite magnitude et la nature bornée de cette erreur indiquent la haute fidélité de la prédiction du modèle flou pour le première état.

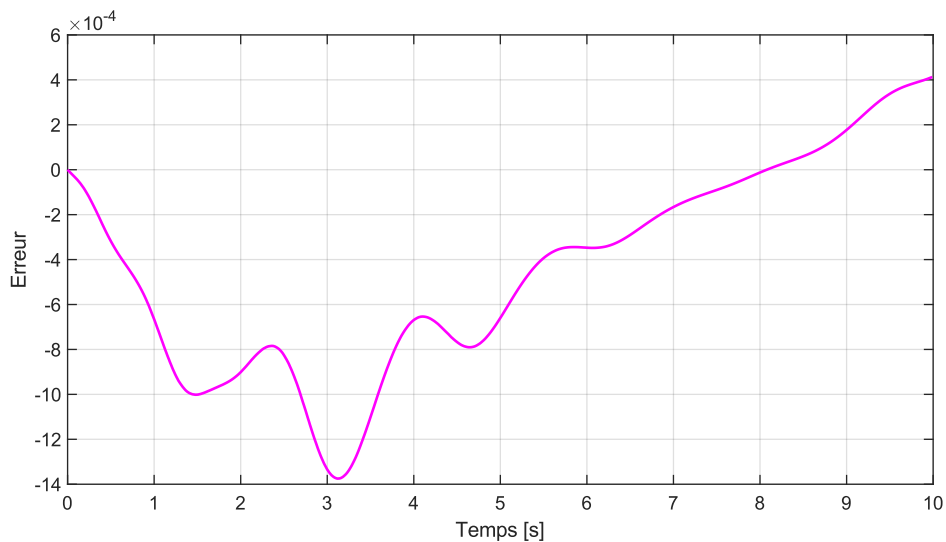


Figure III.28. Trajectoire d'erreur de validation pour l'état x_1 .

Analogue à la Figure III.28, la Figure III.29 montre l'erreur d'estimation pour le deuxième état $x_2(t)$. Les résultats révèlent une déviation minimale entre les trajectoires mesurées et simulées, confirmant que les

deux dynamiques d'état sont capturées avec précision par le modèle flou identifié. Quantitativement, les erreurs quadratiques moyennes sont très faibles:

$$MSE_{x_1} = 6.28 \times 10^{-8}, \quad MSE_{x_2} = 1.64 \times 10^{-7}.$$

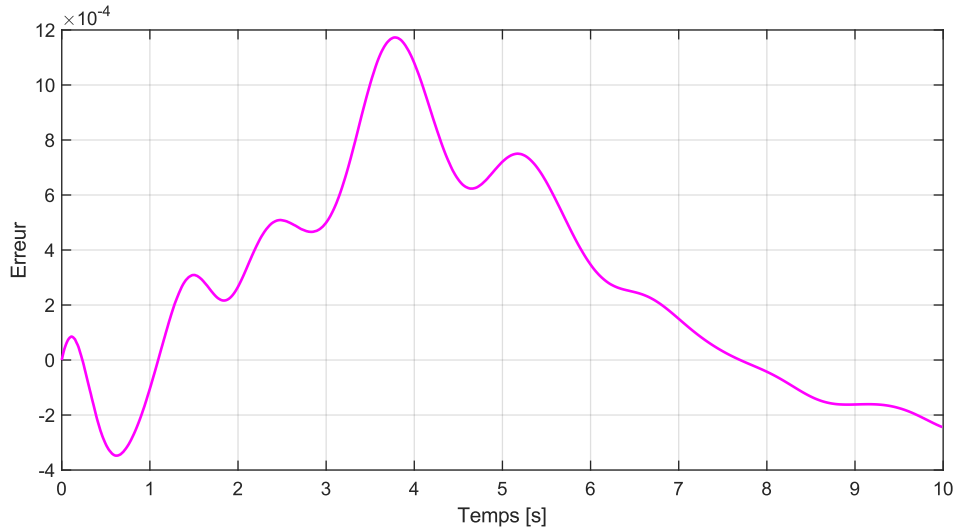


Figure III.29. Trajectoire d'erreur de validation pour l'état x_2 .

La Figure III.30 compare les trajectoires d'état (x_1, x_2) du système non linéaire réel et du modèle flou identifié sous l'entrée de validation. Les trajectoires quasiment superposées indiquent que le modèle reproduit fidèlement la structure dynamique globale et capture l'interdépendance non linéaire entre les états, même sous de nouvelles conditions d'excitation.

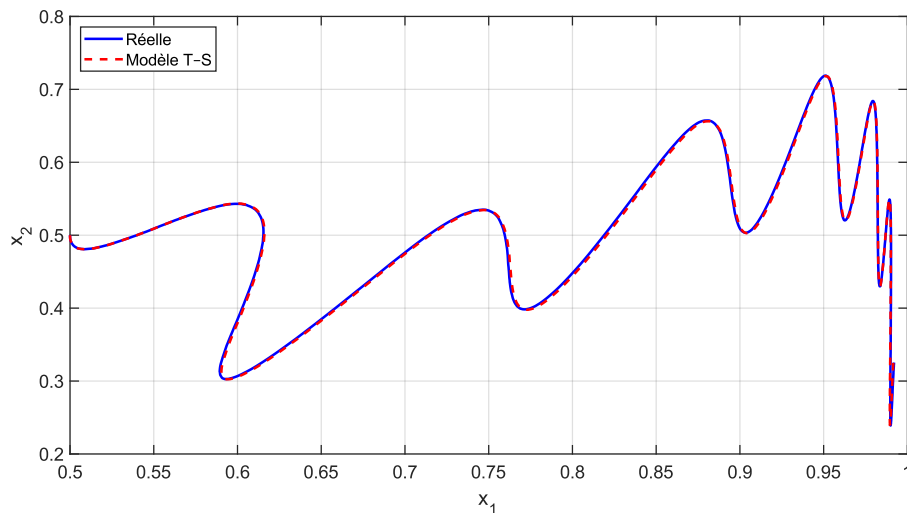


Figure III.30. Comparaison des portraits de phase en phase de validation

Après avoir présenté individuellement les trois méthodologies de modélisation des systèmes non linéaires — la modélisation analytique Takagi–Sugeno, la Transformation de Secteur Non Linéaire, et la modélisation T–S basée sur l’algorithme de Levenberg–Marquardt guidée par les données — ainsi que leurs fondements théoriques et leurs procédures de mise en œuvre, il devient essentiel de procéder à une évaluation comparative exhaustive. Cette section est ainsi dédiée à une comparaison systématique des performances de ces techniques selon des critères unifiés, incluant la précision numérique (telle que l’erreur quadratique moyenne), la capacité de généralisation, la complexité computationnelle et leur pertinence pour la conception de systèmes de commande avancés. L’objectif de cette analyse est d’offrir au lecteur une vision claire des conditions spécifiques dans lesquelles chaque approche se révèle la plus efficace, afin de faciliter le choix de la méthodologie de modélisation la mieux adaptée aux exigences pratiques de l’application et aux ressources disponibles.

III.6. Évaluation des techniques de modélisation de systèmes non linéaires: TS analytique, NST polytopique et modèles TS basés sur LM pilotés par les données

Cette section vise à évaluer et comparer trois différentes techniques de modélisation de systèmes non linéaires qui ont été développées et testées dans les sections précédentes, à savoir:

- 1) Modélisation floue de Takagi–Sugeno
- 2) Transformation de secteur non linéaire et représentation polytopique
- 3) Identification de modèle flou Takagi–Sugeno basée sur les données en utilisant l’algorithme de Levenberg–Marquardt.

La comparaison se concentre sur l’analyse des performances de ces modèles en termes de précision numérique (MSE), de méthodologie de modélisation, de capacité de généralisation au-delà des données utilisées pour leur construction, ainsi que de complexité computationnelle et d’applicabilité aux cadres de commande. Pour assurer une évaluation équitable et cohérente, les trois modèles ont été simulés en utilisant le même signal d’entrée de référence, tel que défini précédemment dans l’équation (III.13). Les réponses d’état $x_1(t)$ et $x_2(t)$ ont été analysées pour évaluer la capacité de chaque modèle à reproduire la dynamique originale du système non linéaire.

Le tableau (III.3) résume les principales caractéristiques méthodologiques de chaque approche, suivi d’une discussion détaillée des résultats numériques et des observations obtenues pour chaque technique de modélisation.

Tableau III.3. Comparaison des modèles T–S analytiques, NST polytopiques et T–S basés sur LM en termes de structure, de complexité et de précision de modélisation.

Élément	Transformation de Secteur Non Linéaire et Représentation Polytopique	Modélisation floue de Takagi–Sugeno	Identification d'un modèle flou de TS à partir de données à l'aide de l'algorithme de Levenberg–Marquardt
Objectif principal	Approximer le système non linéaire avec une représentation multi-modèle convexe en utilisant la transformation sectorielle.	Construire un modèle flou T–S basé sur la dérivation analytique des modèles linéaires locaux.	Identifier (apprendre) un modèle flou T–S à partir des données de simulation du système en utilisant un algorithme d'optimisation numérique.
Méthodologie	Analyse mathématique basée sur les bornes des variables (u, x_2) pour générer des modèles locaux aux frontières des secteurs.	Dérivation directe des matrices (A_i, B_i, D_i) par linéarisation locale et définition des fonctions d'appartenance gaussiennes.	Optimisation numérique des paramètres (A_i, B_i, c_i, v_i) pour minimiser l'erreur quadratique moyenne (MSE) en utilisant l'algorithme de Levenberg–Marquardt.
Nombre de sous-modèles (règles)	4 modèles (sommets) représentant les limites du secteur.	Nombre de règles choisies en fonction de la couverture du domaine (par exemple, 5 règles).	5 règles floues déterminées automatiquement pendant le processus d'apprentissage.
Type de fonctions de pondération (appartenance)	Fonctions d'appartenance linéaires.	Fonctions d'appartenance gaussiennes.	Fonctions d'appartenance gaussiennes.
Méthode de détermination des paramètres	Fixé et dérivé analytiquement des limites sectorielles.	Extrait par linéarisation locale autour des points de fonctionnement.	Déterminé numériquement par optimisation basée sur les données d'entraînement.
Source de données	Analyse symbolique du système non linéaire (sans données expérimentales).	Basé sur la modélisation analytique directe.	Basé sur des données obtenues à partir de la simulation du système réel.
Précision (MSE): entrée de test (III.13)	$x_1 = 7.7037 \times 10^{-33}$ $x_2 = 5.3217 \times 10^{-33}$	$x_1 = 1.1246 \times 10^{-6}$ $x_2 = 2.3621 \times 10^{-6}$	$x_1 = 3.4243 \times 10^{-9}$ $x_2 = 2.4015 \times 10^{-8}$
Capacité de généralisation	Limité	Bien	Excellent
Complexité computationnelle	Faible	Modéré.	Relativement élevé en raison de l'entraînement itératif.

Applicabilité au contrôle	Excellent en raison de sa nature convexe et analytique.	Très approprié et permet une conception équilibrée du contrôleur flou.	Idéal pour une application pratique car les paramètres sont appris à partir de données réelles.
----------------------------------	---	--	---

III.6.1. Modélisation floue de Takagi–Sugeno

L'approche de modélisation floue de Takagi–Sugeno représente le système non linéaire comme une combinaison de sous-modèles localement linéaires mélangés par des fonctions d'appartenance gaussiennes. Dans ce travail, le modèle a été construit en utilisant cinq règles floues, chacune correspondant à une région de fonctionnement distincte du système. La première variable d'état x_1 a été sélectionnée comme variable de planification pour déterminer le niveau d'activation de chaque règle, assurant une transition fluide entre les modèles locaux.

Lorsqu'il a été simulé sous le signal d'entrée de référence défini dans l'équation (III.13), le modèle a montré une grande précision numérique, avec des valeurs d'erreur quadratique moyenne de:

$$MSE_{x_1} = 1.1246 \times 10^{-6}, MSE_{x_2} = 2.3621 \times 10^{-6}$$

Ces valeurs d'erreur très faibles confirment la capacité du modèle à reproduire la dynamique du système non linéaire avec une excellente précision tout en maintenant une complexité computationnelle modérée.

III.6.2. Transformation de secteur non linéaire et représentation polytopique

La technique de Transformation Sectorielle Non-linéaire (NST) et de Représentation Polytopique fournit un cadre analytique pour approximer la dynamique des systèmes non-linéaires en utilisant une structure multi-modèle convexe. Dans cette méthode, les non-linéarités sont bornées dans des secteurs prédéfinis en fonction des plages de l'entrée u et de la variable d'état x_2 . En évaluant le système aux valeurs extrêmes de ces variables, quatre modèles de sommet (locaux) sont obtenus, qui forment collectivement un polytope convexe représentant le comportement global du système.

Sous le même signal d'entrée de référence défini dans l'équation (III.13), le modèle basé sur le NST a démontré une précision exceptionnellement élevée, avec des valeurs d'erreur quadratique moyenne de:

$$MSE_{x_1} = 7.7037 \times 10^{-33}, MSE_{x_2} = 5.3217 \times 10^{-33}.$$

Ces erreurs presque nulles indiquent que l'approche NST peut reproduire les dynamiques non linéaires du système presque parfaitement, grâce à sa nature analytique en forme fermée. De plus, cette méthode nécessite un effort de calcul très faible, car les paramètres du modèle sont dérivés directement des bornes de secteur analytiques plutôt que par optimisation numérique. Par conséquent, la

représentation polytopique basée sur le NST sert d'excellent point de référence analytique et de base robuste pour la modélisation orientée vers le contrôle ultérieur et la comparaison avec des approches floues basées sur les données.

III.6.3. Identification de modèle flou Takagi–Sugeno basée sur les données en utilisant l'algorithme de Levenberg–Marquardt:

L'approche d'identification de modèle flou Takagi–Sugeno (T–S) basée sur les données vise à construire un modèle de système non linéaire directement à partir des données de simulation en utilisant l'algorithme d'optimisation de Levenberg–Marquardt (LM).

Contrairement aux approches analytiques, cette méthode ne repose pas sur des formulations mathématiques prédéfinies des non-linéarités. Au lieu de cela, il identifie les paramètres du modèle, y compris les matrices locales (A_i, B_i) et les paramètres de la fonction d'appartenance gaussienne (c_i, v_i) , en minimisant l'erreur quadratique moyenne entre la sortie du modèle et la réponse du système pendant l'entraînement.

Le modèle identifié se compose de cinq règles floues, chacune décrivant un comportement linéaire local du système. Lorsqu'il est évalué sous le même signal d'entrée de test défini dans l'équation (III.13), le modèle a obtenu d'excellentes performances numériques, avec:

$$MSE_{x_1} = 3.4243 \times 10^{-9}, MSE_{x_2} = 2.4015 \times 10^{-8}.$$

Ces résultats démontrent que la procédure d'identification basée sur LM offre une représentation très précise et flexible des dynamiques non linéaires, correspondant étroitement à la réponse du système sur une large gamme de conditions de fonctionnement. Bien que le processus d'entraînement implique un coût de calcul plus élevé en raison de sa nature d'optimisation itérative, le modèle résultant présente une capacité de généralisation supérieure et une adaptabilité aux données non vues, ce qui en fait le choix le plus pratique et robuste pour les applications réelles.

III.7. Conclusion

Ce chapitre visait à construire un cadre de modélisation complet pour les systèmes dynamiques non linéaires qui équilibre la rigueur analytique avec la flexibilité pratique et fournit une base solide pour la conception de contrôle avancée. Trois méthodologies de modélisation complémentaires ont été développées et évaluées: la modélisation floue analytique de Takagi–Sugeno (T–S), la représentation par transformation de secteur non linéaire polytopique (NST), et l'identification floue T–S basée sur les données en utilisant l'algorithme d'optimisation de Levenberg–Marquardt (LM).

Le modèle analytique T–S a démontré l'efficacité de combiner plusieurs sous-modèles linéaires locaux à travers des fonctions d'appartenance gaussiennes pour obtenir une représentation lisse et précise des comportements non linéaires. L'approche NST polytopique a ensuite reformulé les relations non linéaires en une structure convexe à plusieurs sommets, permettant l'utilisation d'outils de contrôle et d'optimisation linéaires tout en conservant les caractéristiques non linéaires intrinsèques du système. Enfin, le modèle T–S basé sur les données et utilisant l'algorithme de Levenberg-Marquardt a tiré parti de l'optimisation numérique pour ajuster les paramètres du modèle directement à partir des données de simulation, améliorant ainsi la précision prédictive et la capacité de généralisation sur une large plage de fonctionnement.

Une comparaison quantitative entre les trois techniques a montré que le modèle NST polytopique atteint la précision la plus élevée, avec un MSE de l'ordre de 10^{-33} , suivi du modèle basé sur LM guidé par les données avec un MSE d'environ 10^{-9} , puis du modèle analytique T–S avec un MSE d'environ 10^{-6} . Cette analyse met en évidence l'avantage notable du modèle NST, qui allie structure convexe, transparence analytique et fidélité exceptionnelle, ce qui le rend particulièrement adapté aux applications de commande prédictive.

La contribution scientifique de ce chapitre réside dans l'intégration des principaux paradigmes de modélisation non linéaire au sein d'un cadre analytique et numérique unifié, ainsi que dans la présentation d'une évaluation détaillée de leurs performances permettant d'orienter le choix des stratégies de modélisation en fonction des exigences de précision et d'efficacité computationnelle. Sur cette base solide, le chapitre 4 se consacre à la mise en œuvre de la méthodologie de commande prédictive polytopique NST–MPC appliquée à un robot mobile à entraînement différentiel (DDMR), démontrant l'efficacité pratique du cadre de modélisation développé pour assurer un suivi de trajectoire de haute précision et des performances de commande non linéaire robustes.

CHAPITRE IV

REPRESENTATION POLYTOPIQUE DE TAKAGI-SUGENO ET CONCEPTION DE COMMANDE PREDICTIVE POUR LES ROBOTS MOBILES

IV.1. Introduction

Ce chapitre présente un cadre méthodologique complet pour modéliser, analyser et contrôler un robot mobile à entraînement différentiel en utilisant à la fois sa dynamique non linéaire et une représentation polytopique équivalente de Takagi–Sugeno (T–S). Les robots à entraînement différentiel sont largement utilisés dans les applications de navigation autonome cependant, leur structure cinématique intrinsèquement non linéaire pose des défis significatifs pour atteindre un suivi de trajectoire précis et pour concevoir des contrôleurs avec une stabilité garantie et une faisabilité en temps réel [144].

Pour relever ces défis, le chapitre commence par formuler le modèle cinématique non linéaire du robot et le reformuler en un modèle d'erreur de suivi de trajectoire, ce qui transforme la tâche de suivi de chemin en un problème de stabilisation dynamique. Les dynamiques d'erreur non linéaires sont ensuite systématiquement exprimées comme une combinaison convexe de multiples sous-systèmes linéaires en utilisant l'approche de modélisation polytopique T–S. Cette représentation permet l'exploitation d'outils de synthèse de contrôle avancés tels que les Inégalités Matricielles Linéaires (LMI), les fonctions de Lyapunov communes et le Contrôle Prédicatif par Modèle (MPC), tout en préservant le comportement essentiel du système non linéaire original. La précision du modèle T–S est rigoureusement évaluée par des simulations qui le comparent au modèle non linéaire d'erreur de suivi à travers divers indicateurs, y compris les réponses temporelles, le comportement des portraits de phase et des indices de performance quantitatifs tels que le RMSE et l'ISE. Suite à cette validation, le chapitre développe un schéma de contrôle prédictif basé sur le modèle T–S (TS–MPC) garanti en stabilité, où la matrice de Lyapunov stabilisante sert de coût terminal pour assurer la convergence en boucle fermée. Enfin, une analyse comparative approfondie entre le TS–MPC et le MPC non linéaire standard (NMPC) est fournie. La comparaison examine la précision de suivi, l'effort de contrôle et la charge computationnelle, démontrant les avantages et les limitations de chaque approche sous des trajectoires de référence complexes et variables dans le temps.

Dans l'ensemble, ce chapitre intègre une modélisation précise, une analyse de stabilité et des techniques de contrôle prédictif avancées dans un cadre unifié, soutenu par des résultats de simulation détaillés qui mettent en évidence l'efficacité et la pertinence pratique de la méthodologie proposée.

IV.2. Modèle non linéaire du robot mobile différentiel

IV.2.1. Modèle cinématique non linéaire

Le robot mobile à entraînement différentiel est l'une des configurations de robots à roues les plus largement utilisées en raison de sa structure mécanique simple et de sa facilité de contrôle. La posture du robot dans le plan est décrite par les coordonnées (x, y) et l'angle d'orientation θ , tandis que son comportement cinématique peut être exprimé par les équations non linéaires suivantes:

$$\begin{cases} \dot{x}(t) = v(t) \cos(\theta(t)) \\ \dot{y}(t) = v(t) \sin(\theta(t)) \\ \dot{\theta}(t) = \omega(t) \end{cases} \quad (\text{IV.1})$$

où:

- $v(t)$: est la vitesse linéaire
- $\omega(t)$: est la vitesse angulaire
- $\theta(t)$: est l'angle de cap du robot par rapport à l'axe (x) global

Ce modèle non linéaire constitue la base pour concevoir des algorithmes de suivi de trajectoire et de contrôle.

IV.2.2. Variables d'état et d'entrée:

À des fins de conception de contrôle, le vecteur d'état du robot est défini comme:

$$q(t) = \begin{bmatrix} x(t) \\ y(t) \\ \theta(t) \end{bmatrix} \quad (\text{IV.2})$$

Tandis que le vecteur d'entrée de contrôle est donné par:

$$u(t) = \begin{bmatrix} v(t) \\ \omega(t) \end{bmatrix} \quad (\text{IV.3})$$

De même, la posture de référence du robot virtuel suivant la trajectoire désirée est:

$$q_{ref}(t) = \begin{bmatrix} x_{ref}(t) \\ y_{ref}(t) \\ \theta_{ref}(t) \end{bmatrix} \quad (\text{IV.4})$$

IV.2.3. Modèle de dynamique des erreurs:

Pour résoudre le problème de suivi de trajectoire, les coordonnées de posture sont généralement transformées du repère global en repère local attaché au robot. Cette transformation permet d'exprimer

l'erreur de suivi en quantités significatives alignées avec la direction de mouvement du robot. Les erreurs de suivi (e_x, e_y, e_θ) sont définies comme suit:

$$\begin{bmatrix} e_x(t) \\ e_y(t) \\ e_\theta(t) \end{bmatrix} = \begin{bmatrix} \cos(\theta(t)) & \sin(\theta(t)) & 0 \\ -\sin(\theta(t)) & \cos(\theta(t)) & 0 \\ 0 & 0 & 1 \end{bmatrix} (q_{ref}(t) - q(t)) \quad (IV.5)$$

Où:

- e_x : représente l'erreur longitudinale (direction de conduite)
- e_y : est l'erreur latérale (perpendiculaire)
- e_θ : désigne l'erreur d'orientation

En supposant que les robots réel et de référence partagent le même modèle cinématique (Éq. IV.1) et en appliquant la transformation (Éq. IV.5), le modèle d'erreur cinématique peut être exprimé comme suit:

$$\begin{bmatrix} \dot{e}_x \\ \dot{e}_y \\ \dot{e}_\theta \end{bmatrix} = \begin{bmatrix} \cos(e_\theta) & 0 \\ \sin(e_\theta) & 0 \\ 0 & 1 \end{bmatrix} \begin{bmatrix} v_{ref} \\ \omega_{ref} \end{bmatrix} + \begin{bmatrix} -1 & e_y \\ 0 & -e_x \\ 0 & -1 \end{bmatrix} \begin{bmatrix} v \\ \omega \end{bmatrix} \quad (IV.6)$$

Où v_{ref} et ω_{ref} sont les vitesses linéaires et angulaires de référence.

L'entrée de contrôle est souvent décomposée en composantes anticipatives et rétroactives comme suit:

$$\begin{bmatrix} v \\ \omega \end{bmatrix} = \begin{bmatrix} v_{ref} \cos(e_\theta) \\ \omega_{ref} \end{bmatrix} + \begin{bmatrix} v_{fb} \\ \omega_{fb} \end{bmatrix} \quad (IV.7)$$

Substituer l'équation (IV.7) dans l'équation (IV.6) donne la dynamique finale de l'erreur de suivi [145]:

$$\begin{cases} \dot{e}_x = \omega_{ref} e_y - v_{fb} + e_y \omega_{fb} \\ \dot{e}_y = -\omega_{ref} e_x + v_{ref} \sin(e_\theta) - e_x \omega_{fb} \\ \dot{e}_\theta = -\omega_{fb} \end{cases} \quad (IV.8)$$

L'objectif de contrôle est de concevoir des signaux de rétroaction appropriés v_{fb} et ω_{fb} tels que:

$$\lim_{t \rightarrow \infty} e_x(t) = \lim_{t \rightarrow \infty} e_y(t) = \lim_{t \rightarrow \infty} e_\theta(t) = 0 \quad (IV.9)$$

Assurer un suivi précis de la trajectoire du mouvement de référence.

IV.3. Construction du modèle polytopique de Takagi-Sugeno

Le modèle d'erreur de suivi non linéaire du robot mobile à entraînement différentiel présenté dans l'équation (IV.8) contient plusieurs termes non linéaires bornés qui peuvent être systématiquement exprimés sous forme polytopique de Takagi-Sugeno (T-S). Cette représentation permet l'utilisation de

techniques avancées de synthèse de contrôle telles que la Compensation Distribuée Parallèle (PDC) et la conception basée sur les Inégalités Matricielles Linéaires (LMI) [146].

IV.3.1. Modèle d'erreur non linéaire équivalent:

Basé sur l'équation (IV.8), la dynamique de l'erreur de suivi peut être réécrite sous une forme matricielle compacte qui met en évidence les termes non linéaires comme suit:

$$\begin{bmatrix} \dot{e}_x \\ \dot{e}_y \\ \dot{e}_\theta \end{bmatrix} = \begin{bmatrix} 0 & \omega_{ref} & 0 \\ -\omega_{ref} & 0 & v_{ref} \frac{\sin(e_\theta)}{e_\theta} \\ 0 & 0 & 0 \end{bmatrix} \begin{bmatrix} e_x \\ e_y \\ e_\theta \end{bmatrix} + \begin{bmatrix} -1 & e_y \\ 0 & -e_x \\ 0 & -1 \end{bmatrix} \begin{bmatrix} v_{fb} \\ \omega_{fb} \end{bmatrix} \quad (IV.10)$$

Les non-linéarités sont clairement identifiées comme: $\omega_{ref}, v_{ref} \frac{\sin(e_\theta)}{e_\theta}, e_y, e_x$

Ces termes serviront de variables prémisses de la représentation de Takagi–Sugeno.

IV.3.2. Définition des bornes de la variable (Z_l et Z_u):

D'après l'équation (IV.10), le vecteur de prémisse est défini comme:

$$z(t) = \begin{bmatrix} \omega_{ref}(t) \\ v_{ref}(t) \text{sinc}(e_\theta(t)) \\ e_y(t) \\ e_x(t) \end{bmatrix} \quad (IV.11)$$

Chaque composante de $z(t)$ est supposée varier dans une région compacte connue:

$$z_j \in [z_j, \bar{z}_j], j = 1,2,3,4 \quad (IV.12)$$

Les bornes correspondantes sont sélectionnées en fonction des limites de fonctionnement du robot et des magnitudes attendues des erreurs de suivi:

$$\begin{aligned} \underline{\omega}_{ref} &\leq \omega_{ref}(t) \leq \bar{\omega}_{ref} \\ |e_\theta| &\leq \bar{e}_\theta < \pi, 0 < \underline{v}_{ref} \leq v_{ref} \leq \bar{v}_{ref} \\ v_{ref} \text{sinc}(\bar{e}_\theta) &\leq v_{ref} \text{sinc}(e_\theta) \leq \bar{v}_{ref} \\ |e_y| &\leq \bar{e}_y, |e_x| \leq \bar{e}_x \end{aligned} \quad (IV.13)$$

Ces limites définissent la région convexe où le modèle polytopique est valide et représente avec précision les dynamiques non linéaires.

IV.3.3. Génération de sommets (structure polytopique à 16 sommets):

Puisque quatre termes non linéaires indépendants apparaissent dans l'équation (IV.10), le nombre total de règles floues (ou de sommets) est:

$$r = 2^4 = 16$$

Chaque sommet correspond à une combinaison unique des bornes inférieures et supérieures des variables de prémisses. Les paramètres des sommets sont calculés comme suit:

$$\epsilon_i^j = z_j + \iota_j^i (\bar{z}_j - z_j), \quad \iota_j^i \in \{0,1\} \quad (\text{IV.14})$$

Où la variable binaire ι_j^i indique si la borne inférieure (0) ou supérieure (1) de z_j est sélectionnée. Les 16 sommets sont générés par énumération binaire des quatre indices $(\iota_1^i, \iota_2^i, \iota_3^i, \iota_4^i)$.

IV.3.4. Calcul des fonctions d'appartenance:

Chaque sommet i est associé à une fonction d'appartenance $h_i(z(t))$ qui détermine son poids dans la combinaison convexe. Les propriétés suivantes sont valables pour tout $z(t)$:

$$h_i(z(t)) \geq 0, \quad \sum_{i=1}^{16} h_i(z(t)) = 1 \quad (\text{IV.15})$$

Les fonctions d'appartenance sont construites comme des produits de termes de pondération unidimensionnels normalisés:

$$h_i(z(t)) = \prod_{j=1}^4 w_j^{\iota_j^i}(z_j(t)) \quad (\text{IV.16})$$

Avec:

$$\begin{aligned} w_j^1(z_j(t)) &= \frac{z_j(t) - z_j}{\bar{z}_j - z_j} \quad j = 1,2,3,4 \\ w_j^0(z_j(t)) &= 1 - w_j^1(z_j(t)) \end{aligned} \quad (\text{IV.17})$$

Cette structure garantit une interpolation fluide entre tous les systèmes de sommets.

IV.3.5. Matrices du système pour chaque sommet (A_i, B_i):

Le modèle polytopique global de Takagi–Sugeno de la dynamique de l'erreur de suivi est exprimé comme suit:

$$\dot{e}(t) = \sum_{i=1}^{16} h_i(z(t)) (A_i e(t) + B_i u_{fb}(t)) \quad (\text{IV.18})$$

$$\text{Où: } e(t) = \begin{bmatrix} e_x \\ e_y \\ e_\theta \end{bmatrix}, u_{fb}(t) = \begin{bmatrix} v_{fb} \\ \omega_{fb} \end{bmatrix}$$

Et chaque sommet i est défini par des matrices constantes:

$$A_i = \begin{bmatrix} 0 & \epsilon_i^1 & 0 \\ -\epsilon_i^1 & 0 & \epsilon_i^2 \\ 0 & 0 & 0 \end{bmatrix}, B_i = \begin{bmatrix} -1 & \epsilon_i^3 \\ 0 & -\epsilon_i^4 \\ 0 & -1 \end{bmatrix} \quad (\text{IV.19})$$

Les paramètres ϵ_i^j sont calculés à partir de l'Éq. (IV.14) en utilisant les bornes définies dans l'Éq. (IV.13). Ainsi, le système non linéaire est représenté exactement comme une combinaison convexe de 16 sous-systèmes linéaires.

IV.3.6. Validation du modèle polytopique:

Pour garantir la correction et la fiabilité de la représentation T–S, deux conditions de vérification doivent être satisfaites:

1. Condition de convexité:

La somme des fonctions d'appartenance doit être égale à l'unité pour tout $z(t)$:

$$\sum_{i=1}^{16} h_i(z(t)) = 1 \text{ garantir une combinaison convexe valide parmi tous les sommets.}$$

2. Condition de précision du modèle:

La réponse du modèle T–S $e_{TS}(t)$ est comparée avec le modèle non linéaire original $e_{NL}(t)$ sous des entrées de référence identiques. La déviation:

$$\Delta e(t) = e_{TS}(t) - e_{NL}(t)$$

doit rester négligeable (typiquement $|\Delta e(t)| < 10^{-3}$) dans les limites spécifiées, confirmant que la formulation T–S reproduit exactement le comportement non linéaire dans la région de fonctionnement définie.

IV.4. Comparaison entre les modèles polytopiques et non linéaires du robot mobile différentiel

Cette section vise à évaluer la précision du modèle polytopique de Takagi–Sugeno (T–S) dans la représentation des dynamiques non linéaires d'erreur de suivi du robot mobile à entraînement différentiel. Une série de simulations numériques a été réalisée pour comparer les deux modèles en termes des variables d'erreur e_x, e_y, e_θ , des trajectoires d'erreur instantanées, des portraits de phase et des indices de performance quantitatifs (RMSE et ISE). Toutes les simulations ont été effectuées dans les limites de fonctionnement prédéfinies pour garantir la validité de l'approximation polytopique.

IV.4.1. Conditions initiales et paramètres:

Les bornes des variables de prémisse définissant la région polytopique ont été sélectionnées comme suit:

$$z_{\text{lower}} = [-10, -2, -5, -5], z_{\text{upper}} = [10, 2, 5, 5]$$

En se basant sur ces limites, une structure polytopique à 16 sommets a été générée conformément à l'équation (IV.19) présentée dans la section précédente. Chaque sommet i est caractérisé par des matrices constantes A_i et B_i , représentant la dynamique linéaire locale à l'intérieur de la région définie par les variables prémisses.

$$z = [\omega_{ref}, v_{ref} \text{sinc}(e_\theta), e_y, e_x]^T$$

La simulation a été réalisée sur l'intervalle de temps $t \in [0, 10]$ secondes, avec une période d'échantillonnage $T_s = 0.1$ s, et l'état initial suivant:

$$e(0) = [e_x(0), e_y(0), e_\theta(0)]^T = [0.1, -0.05, 0.05]^T$$

Les vitesses linéaires et angulaires de référence ont été choisies comme suit:

$$v_{ref}(t) = 1.0 \text{ m/s}, \omega_{ref}(t) = 0.2 \text{ rad/s}$$

IV.4.2. Comparaison des variables d'état e_x, e_y, e_ϕ :

Cette sous-section présente une comparaison détaillée entre le modèle non linéaire et le modèle polytopique de Takagi–Sugeno (T–S) pour les trois états d'erreur de suivi: erreur longitudinale e_x , erreur latérale e_y et erreur d'orientation e_θ . Les résultats de simulation montrés dans les Figures IV.1–IV.5 démontrent la capacité du modèle polytopique à reproduire avec précision les dynamiques non linéaires sur l'ensemble de l'horizon temporel.

La Figure IV.1 montre l'évolution temporelle de l'erreur de suivi longitudinal $e_x(t)$ pour le modèle non linéaire et le modèle polytopique T–S. Les deux courbes présentent un chevauchement presque parfait, confirmant la capacité du modèle polytopique à reproduire avec précision la dynamique du système non linéaire.

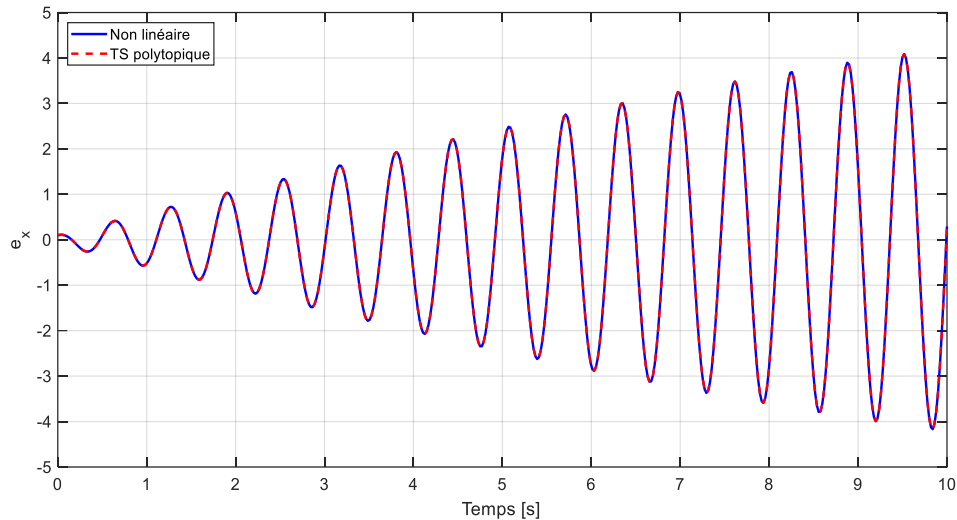


Figure IV.1. Comparaison de l'erreur de suivi $e_x(t)$ entre le modèle non linéaire et le modèle polytopique.

La Figure IV.2 montre la comparaison de l'erreur de suivi latéral $e_y(t)$ entre le modèle non linéaire et le modèle polytopique T-S. Les trajectoires correspondent exactement tout au long de la simulation, ce qui indique la haute précision de la représentation T-S dans les limites de fonctionnement spécifiées.

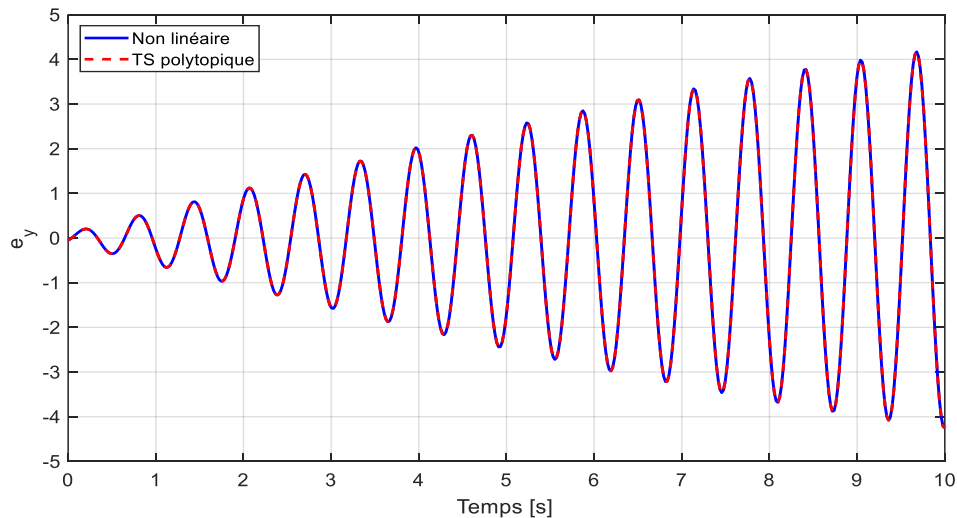


Figure IV.2. Comparaison de l'erreur de suivi $e_y(t)$ entre le modèle non linéaire et le modèle polytopique.

La Figure IV.3 montre l'évolution de l'erreur de suivi d'orientation $e_\theta(t)$ pour les modèles non linéaires et polytopiques T-S. Les deux courbes sont parfaitement alignées, démontrant la capacité du modèle polytopique à reproduire la dynamique de rotation du système sans déviation notable.

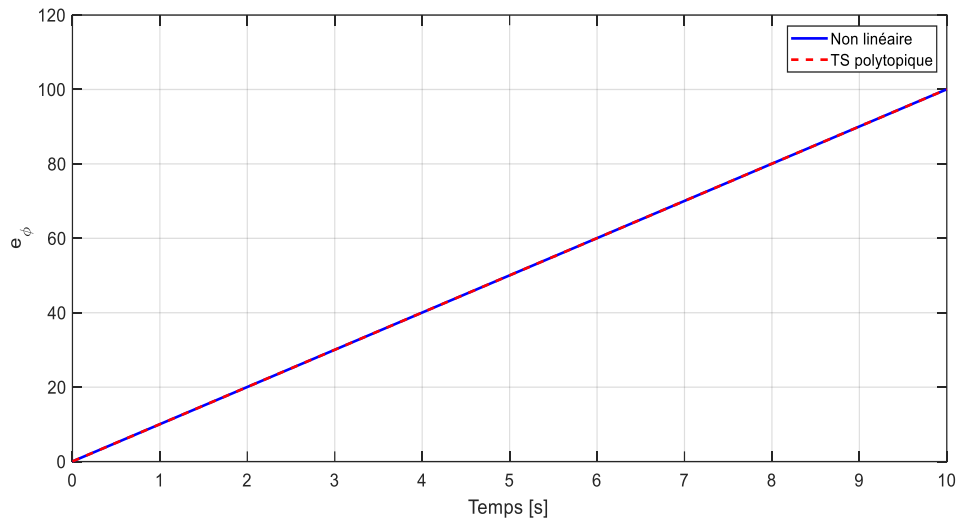


Figure IV.3. Comparaison de l'erreur de suivi d'orientation $e_{\theta}(t)$ entre le modèle non linéaire et le modèle polytopique T–S.

La Figure IV.4 illustre l'évolution des erreurs instantanées entre le modèle non linéaire et le modèle polytopique de Takagi–Sugeno (T–S) pour les variables $e_x(t)$, $e_y(t)$ et $e_{\theta}(t)$. Les erreurs restent très petites tout au long de la période de simulation, indiquant un chevauchement presque parfait entre les deux modèles et confirmant la précision de la représentation polytopique dans les limites de fonctionnement adoptées.

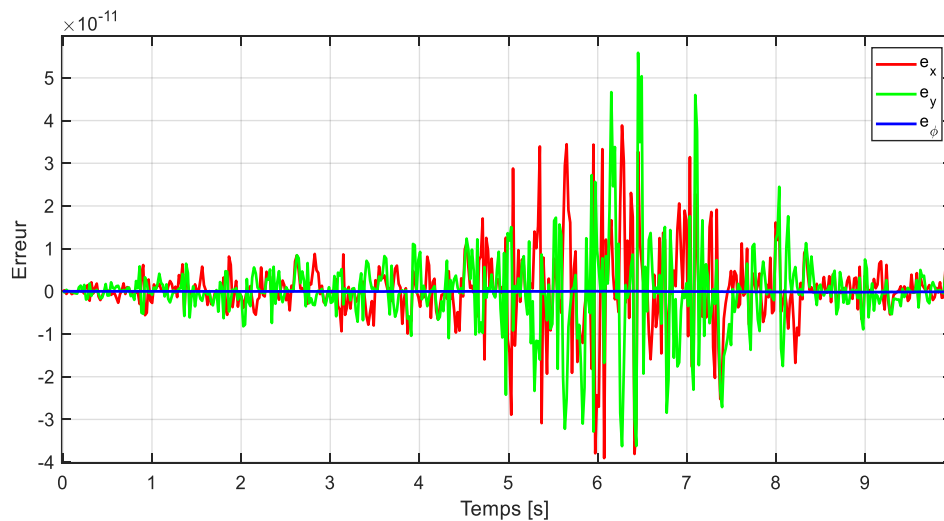


Figure IV.4. Comparaison des erreurs instantanées entre le modèle non linéaire et le modèle polytopique.

La Figure IV.5 présente le portrait de phase décrivant la relation entre $e_y(t)$ et $e_x(t)$ pour le modèle non linéaire et le modèle polytopique T–S. Les trajectoires présentent un chevauchement complet,

démontrant que le modèle polytopique préserve avec succès la structure de l'espace des phases et le comportement dynamique du système non linéaire original.

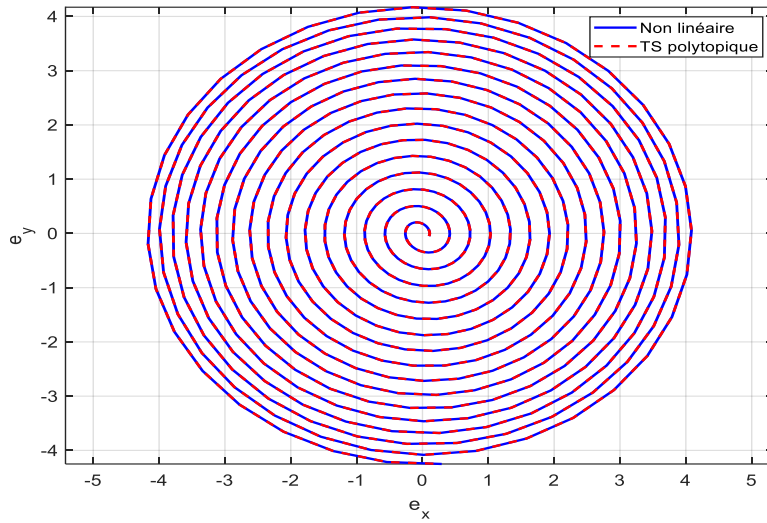


Figure IV.5. Comparaison des portraits de phase de $e_y(t)$ par rapport à $e_x(t)$ pour les modèles T-S non linéaires et polytopiques.

IV.4.3. Indices de performance (RMSE et ISE):

Pour évaluer quantitativement la divergence entre le modèle non linéaire (NL) et le modèle polytopique de Takagi-Sugeno (TS), deux indices de performance numériques ont été évalués pour les variables d'erreur e_x, e_y, e_θ :

- Racine de l'Erreur Quadratique Moyenne (RMSE)
- Intégrale de l'Erreur au Carré (ISE)

Ces indices fournissent une mesure directe de la manière dont le modèle TS reproduit fidèlement le comportement du système non linéaire. Le RMSE indique l'amplitude moyenne de la déviation entre les deux modèles et est calculé comme suit:

$$\text{RMSE}(e_j) = \sqrt{\frac{1}{N} \sum_{k=1}^N (e_{j,NL}(k) - e_{j,TS}(k))^2} \quad (\text{IV.20})$$

Où:

- $e_{j,NL}(k)$: erreur de la variable j dans le modèle non linéaire à l'échantillon k
- $e_{j,TS}(k)$: erreur correspondante dans le modèle TS

- N : nombre total d'échantillons

L'ISE mesure l'erreur quadratique accumulée sur toute la durée de la simulation et est définie par:

$$\text{ISE}(e_j) = \sum_{k=1}^N (e_{j,NL}(k) - e_{j,TS}(k))^2 \cdot \Delta t \quad (\text{IV.21})$$

Avec Δt désignant le pas de temps de simulation.

Les valeurs calculées de RMSE et ISE pour chaque variable d'erreur sont résumées dans le tableau ci-dessous:

Tableau.IV.1 Indices de performance pour les variables d'erreur (RMSE et ISE)

Variable	RMSE	ISE
e_x	8.8918×10^{-12}	7.9223×10^{-22}
e_y	9.6636×10^{-12}	9.3572×10^{-22}
e_θ	7.9354×10^{-14}	6.3096×10^{-26}

Les résultats indiquent clairement que la différence numérique entre le modèle non linéaire et le modèle polytopique TS est pratiquement négligeable. Les valeurs de RMSE se situent dans la plage de (10^{-12} to 10^{-14}), tandis que les valeurs d'ISE sont encore plus petites, allant de 10^{-22} to 10^{-26} . Ces déviations extrêmement petites confirment que le modèle TS représente une transformation polytopique des dynamiques non linéaires originales avec une correspondance numérique presque parfaite. Par conséquent, la formulation TS peut être utilisée de manière fiable pour la conception de contrôleurs et l'analyse subséquente, car elle préserve le comportement essentiel du système non linéaire.

IV.5. Conception intégrée du contrôle prédictif basé sur le modèle de Takagi–Sugeno

Cette section développe un schéma de contrôle prédictif par modèle (MPC) pour le modèle polytopique de Takagi–Sugeno (T–S) dérivé, intégrant une garantie de stabilité basée sur Lyapunov. Nous dérivons d'abord une matrice de Lyapunov commune P qui certifie la stabilité du modèle T–S, puis nous formulons le MPC en temps discret pour le système polytopique, et enfin nous montrons comment la matrice P commune entre dans le coût du MPC pour garantir la stabilité en boucle fermée.

IV.5.1. Dérivation de la matrice de Lyapunov pour la stabilité du modèle polytopique T–S:

A. Modélisation multi-vertex Takagi–Sugeno:

Comme le montre la section IV.3, la dynamique non linéaire de l'erreur de suivi peut être exactement représentée par un modèle flou polytopique de Takagi–Sugeno (T–S) composé de N_v systèmes linéaires de sommet. La représentation T–S en temps continu est donnée par:

$$\dot{e}(t) = \sum_{i=1}^{N_v} h_i(z(t))(A_i e(t) + B_i u_{fb}(t)) \quad (IV.22)$$

Où $e(t)$ est l'état d'erreur et $u_{fb}(t)$ est l'entrée de rétroaction (IV.18)–(IV.19). Chaque sommet i est caractérisé par des matrices constantes (A_i, B_i) , et sa fonction d'appartenance correspondante satisfait :

$$h_i(z) \geq 0, \quad \sum_{i=1}^{N_v} h_i(z) = 1$$

Le modèle T–S peut être vu de manière équivalente comme une Inclusion Différentielle Linéaire (LDI) évoluant à l'intérieur de l'enveloppe convexe des dynamiques de sommet $\{(A_i, B_i)\}$. L'objectif de cette sous-section est de construire une seule fonction de Lyapunov quadratique.

$$V(e) = e^T P e, \quad P = P^T > 0$$

Qui garantit la stabilité pour tous les systèmes de sommets du modèle polytopique.

B. Conditions de stabilité utilisant une fonction de Lyapunov commune:

Considérons une loi de commande par retour d'état appliquée à chaque sommet du modèle :

$$u_{fb}(t) = K_i e(t) \quad (IV.23)$$

La dynamique en boucle fermée associée au sommet i devient :

$$\dot{e}(t) = (A_i + B_i K_i) e(t) \quad (IV.24)$$

La dérivée temporelle de la fonction de Lyapunov est alors donnée par :

$$\dot{V}(e) = e^T [(A_i + B_i K_i)^T P + P(A_i + B_i K_i)] e \quad (IV.25)$$

Une condition suffisante de stabilité quadratique du système est :

$$(A_i + B_i K_i)^T P + P(A_i + B_i K_i) < 0, \quad i = 1, \dots, N_v \quad (IV.26)$$

Lorsque cette condition est satisfaite pour tous les sommets, la stabilité asymptotique du modèle flou T–S complet est garantie grâce à la propriété de convexité des fonctions d'appartenance.

C. Formulation LMI et calcul de la matrice de Lyapunov

L'inégalité de stabilité précédente est bilinéaire par rapport aux variables de décision P et K_i . Afin d'obtenir un problème convexe pouvant être résolu efficacement par les outils d'optimisation semi-définie, le changement de variable suivant est introduit :

$$Y_i = K_i X \quad (IV.27)$$

où $X = P^{-1}$ est une matrice symétrique définie positive.

La condition de stabilité peut alors être reformulée sous la forme de l'inégalité matricielle linéaire suivante :

$$(A_i X + B_i Y_i) + (A_i X + B_i Y_i)^T < 0 \quad (IV.28)$$

Afin d'accroître la robustesse et la marge de stabilité du système, les LMIs sont résolues sous des contraintes assurant la positivité de la matrice X ainsi que l'existence d'une solution strictement faisable.

Les contraintes principales s'écrivent :

$$(A_i X + B_i Y_i) + (A_i X + B_i Y_i)^T \leq -\eta I, i = 1, \dots, N_v \quad (IV.29)$$

Dans le cadre de l'approche TS/PDC, les conditions relatives aux termes croisés entre les différents sous-modèles doivent également être satisfaites. Ces contraintes s'écrivent :

$$(A_i X + B_i Y_j) + (A_i X + B_i Y_j)^T + (A_j X + B_j Y_i) + (A_j X + B_j Y_i)^T \leq -\eta I, i \neq j \quad (IV.30)$$

L'ensemble de ces contraintes garantit l'existence d'une fonction de Lyapunov quadratique commune pour tous les sommets du polytope flou, assurant ainsi la stabilité quadratique globale du système.

Le problème d'optimisation est résolu à l'aide du solveur CVX basé sur la programmation semi-définie. Une fois la solution obtenue, la matrice de Lyapunov commune est reconstruite à partir de la relation :

$$P = X^{-1} \quad (IV.31)$$

L'obtention d'une valeur optimale strictement positive de η atteste de la faisabilité stricte des contraintes LMIs et garantit la stabilité quadratique asymptotique du système flou Takagi–Sugeno en boucle fermée. La matrice P ainsi déterminée est ensuite utilisée comme matrice de coût terminale dans la formulation du contrôleur prédictif flou (FMPC), permettant d'intégrer explicitement les propriétés de stabilité de Lyapunov au processus de synthèse du contrôleur prédictif. [147]

IV.5.2. Conception de MPC pour le système polytopique T–S:

Cette sous-section développe le schéma de contrôle prédictif de modèle (MPC) en temps discret pour le modèle polytopique de Takagi–Sugeno (T–S). La procédure consiste à calculer les fonctions d'appartenance en ligne, à construire le modèle linéaire mélangé (variant dans le temps), à discrétiser les dynamiques T–S, à former le modèle de prédiction, à définir les horizons MPC, à introduire la fonction de coût (avec la matrice de Lyapunov P comme pénalité terminale), et enfin à formuler le problème d'optimisation comme un programme quadratique contraint (QP).

A. Calcul des fonctions d'appartenance et construction du modèle variant dans le temps:

À chaque instant d'échantillonnage (k), les variables de prémisse (planification) $z[k]$ sont mesurées ou estimées. À partir de $z[k]$, les fonctions d'appartenance normalisées $h_i(z[k])$ sont calculées exactement comme défini dans (IV.16). Par construction:

$$h_i(z[k]) \geq 0, \quad \sum_{i=1}^{N_v} h_i(z[k]) = 1.$$

En utilisant ces poids, le modèle T-S produit une approximation linéaire mélangée au temps k . Les matrices variant dans le temps sont définies comme suit:

$$\bar{A}(k) = \sum_{i=1}^{N_v} h_i(z[k])A_i \quad , \quad \bar{B}(k) = \sum_{i=1}^{N_v} h_i(z[k])B_i \quad (IV.32)$$

Ainsi, la dynamique en temps continu à l'instant k peut s'écrire comme suit:

$$\dot{e}(t) = \bar{A}(k)e(t) + \bar{B}(k)u_{fb}(t) \quad (IV.33)$$

Parce que la représentation polytopique est exacte, cette combinaison convexe correspond précisément au modèle non linéaire à ce moment-là.

B. Discrétisation du modèle T-S local:

Le MPC fonctionne en temps discret par conséquent, chaque modèle de sommet (A_i, B_i) est discrétisé en utilisant le temps d'échantillonnage T_s . Un maintien d'ordre zéro (ZOH) donne:

$$A_{i,d} = e^{A_i T_s} \quad , \quad B_{i,d} = \int_0^{T_s} e^{A_i \tau} B_i d\tau \quad (IV.34)$$

En utilisant les mêmes fonctions d'appartenance $h_i[k] = h_i(z[k])$, les dynamiques T-S en temps discret deviennent:

$$e[k+1] = \sum_{i=1}^{N_v} h_i[k](A_{i,d}e[k] + B_{i,d}u[k]) \quad (IV.35)$$

De manière équivalente, on définit les matrices discrètes-temps mélangées:

$$\bar{A}_d(k) = \sum_{i=1}^{N_v} h_i[k]A_{i,d} \quad , \quad \bar{B}_d(k) = \sum_{i=1}^{N_v} h_i[k]B_{i,d} \quad (IV.36)$$

Afin que:

$$e[k+1] = \bar{A}_d(k)e[k] + \bar{B}_d(k)u[k] \quad (IV.37)$$

C. Formation du modèle de prédiction multi-étapes:

Pour construire des prédictions MPC sur un horizon N_p , le modèle en temps discret:

$$e[k+1] = \bar{A}_d(k)e[k] + \bar{B}_d(k)u[k] \quad (IV.38)$$

Est propagé récursivement vers l'avant. La séquence de prédiction est:

$$\begin{aligned} e[k+2] &= \bar{A}_d(k+1)e[k+1] + \bar{B}_d(k+1)u[k+1] \\ &\vdots \\ e[k+N_p] &= \bar{A}_d(k+N_p-1)e[k+N_p-1] + \bar{B}_d(k+N_p-1)u[k+N_p-1] \end{aligned} \quad (IV.39)$$

En pratique, les matrices $\bar{A}_d(k+j)$, $\bar{B}_d(k+j)$ dépendent des variables de prémisse futures $z[k+j]$. Une approche courante et efficace sur le plan computationnel consiste à figer les fonctions

d'appartenance sur l'horizon, c'est-à-dire à utiliser $h_i[k]$ pour tout j , ce qui donne un modèle de prédiction localement linéaire adapté à la mise en œuvre en temps réel du MPC.

D. Définition des horizons de prédiction et de contrôle:

Le contrôleur MPC utilise un horizon de prédiction N_p et un horizon de contrôle $N_c \leq N_p$.

- Les états $e[k + j]$ sont prédits pour $j = 0, \dots, N_p - 1$.
- Les entrées de contrôle $u[k + j]$ sont des variables d'optimisation pour $j = 0, \dots, N_c - 1$.

Au-delà de l'horizon de contrôle, l'entrée est généralement maintenue constante:

$$u[k + j] = u[k + N_c - 1] \quad , \quad j \geq N_c \quad (\text{IV.40})$$

Le choix de N_p et N_c équilibre la précision de la prédiction, l'effort de calcul et la performance en boucle fermée.

E. Fonction de coût et utilisation de la matrice de Lyapunov (P) comme poids terminal:

Soient $Q \geq 0$ et $R > 0$ les matrices de pondération d'état et d'entrée, respectivement. Le coût MPC sur l'horizon N_p est:

$$J = \sum_{j=0}^{N_p-1} (e[k+j]^T Q e[k+j] + u[k+j]^T R u[k+j]) + e[k + N_p]^T P e[k + N_p] \quad (\text{IV.41})$$

Le coût terminal utilise la matrice de Lyapunov P dérivée dans la Section IV.5.1. Parce que P satisfait les inégalités de Lyapunov pour tous les sommets du modèle T–S, la pénalité terminale agit comme une fonction de Lyapunov de contrôle locale. Cette structure est fondamentale pour garantir la stabilité du schéma à horizon fuyant et est largement adoptée dans le MPC avec des modèles linéaires polytopiques ou variant dans le temps.

F. Formulation de programme quadratique:

Avec le modèle de prédiction et la fonction de coût définis, le problème MPC devient un programme quadratique convexe de dimension finie. La variable d'optimisation est le vecteur empilé des entrées de contrôle:

$$u = [u[k]^T \quad u[k + 1]^T \quad \dots \quad u[k + N_c - 1]^T]^T \quad (\text{IV.42})$$

La fonction objectif J est quadratique en u , et les états prédits dépendent linéairement de u . Le problème résultant est un QP standard qui peut être résolu efficacement à chaque instant d'échantillonnage. Seul le premier contrôle $u[k]$ est appliqué, après quoi l'horizon avance.

La figure IV.6 illustre l'architecture complète du cadre de contrôle TS-MPC proposé. Le processus commence avec la trajectoire de référence, qui est transformée en erreurs de suivi par une transformation de coordonnées entre le cadre du robot et le cadre global. Ces erreurs sont introduites dans le modèle cinématique du robot, qui est ensuite reformulé en une représentation polytopique de Takagi-Sugeno en définissant des bornes variables et en générant les matrices de sommet correspondantes A_i et B_i . Le solveur LMI calcule une matrice de Lyapunov commune P , garantissant la stabilité à travers tous les sommets polytopiques. Cette information est intégrée dans la couche d'optimisation MPC, où l'entrée de contrôle composée de termes d'anticipation et de rétroaction est calculée. Enfin, les signaux de contrôle calculés sont appliqués au robot pour obtenir un suivi de trajectoire précis et stable.

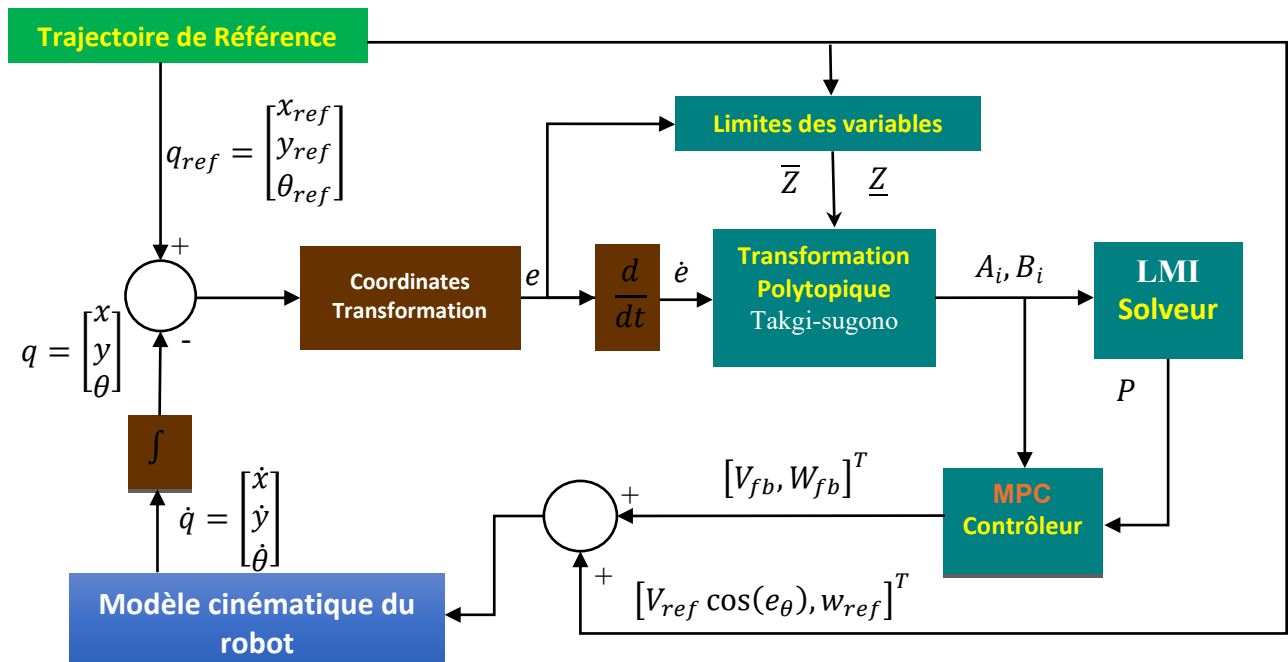


Figure IV.6. Diagramme bloc du cadre de contrôle TS-MPC pour le robot à entraînement différentiel.

Avec l'intégration complète des garanties de stabilité basées sur Lyapunov et la formulation MPC en temps discret pour le modèle polytopique T-S, la section suivante évalue le contrôleur proposé à travers une série de scénarios de simulation. Ces simulations comparent la stratégie TS-MPC avec le MPC non linéaire standard en termes de précision de suivi, d'effort de contrôle, d'efficacité computationnelle et de performance globale en boucle fermée.

IV.6. Simulation et analyse comparative entre TS–MPC et NMPC

IV.6.1. Configuration du scénario de simulation:

L'environnement de simulation pour à la fois TS–MPC et NMPC [148] a été configuré dans des conditions de fonctionnement identiques pour garantir une comparaison équitable et cohérente entre les deux stratégies de contrôle. Les principaux éléments de la configuration de la simulation sont résumés ci-dessous:

- **Durée de la simulation:** $T_{sim} = 30 \text{ s}$, $T_s = 0.033 \text{ s}$

fournissant une résolution temporelle suffisante pour un suivi de trajectoire précis et stable.

- **Trajectoire de référence:**

Un chemin composite non circulaire basé sur des fonctions sinusoïdales a été sélectionné pour imposer des variations continues de courbure, de vitesse et de cap. Cela garantit que les deux contrôleurs sont évalués dans des conditions de fonctionnement dynamiques et non triviales.

$$\begin{cases} x_r(t) = 1.1 + 0.7 \sin(\omega t) \\ y_r(t) = 0.9 + 0.7 \sin(2\omega t) \end{cases} \quad (\text{IV.43})$$

- **État initial:** $[x_0, y_0, \theta_0] = [1.1, 0.8, \pi/3]$

Utilisé pour les deux contrôleurs afin d'assurer des points de départ identiques.

- **Horizon de prédiction:** $N_p = 15$
- **Matrices de pondération MPC:**
 - Suivi d'état: $Q = \text{diag}(0.1, 40, 0.01)$
 - Contrôle effort: $R = \text{diag}(0.01, 0.01)$
 - Matrice terminale de Lyapunov : $P = [30.5364 \ 0 \ 0; 0 \ 128.3796 \ 196.5389; 0 \ 196.5389 \ 304.3703]$
- **Domaine du modèle T–S et limites de planification:**

Le modèle de Takagi–Sugeno a été construit en utilisant quatre variables de planification dont le domaine admissible est défini par les bornes inférieure et supérieure:

$$z_{\text{lower}} = [-26, -20, -20, -20] \quad , \quad z_{\text{upper}} = [26, 8, 20, 20]$$

Ces bornes définissent la région de fonctionnement (hyper-rectangle) dans laquelle le modèle T–S est valide. Chaque variable de planification z_i est mappée à ces limites, et les 16 sommets T–S sont générés en évaluant les matrices du système à toutes les combinaisons de bornes inférieures et supérieures.

Pendant la simulation, le vecteur de planification $z(t)$ est calculé en ligne, et les poids de sommet correspondants $h_i(t)$ sont mis à jour dynamiquement pour construire les matrices mélangées:

$$A_z(t) = \sum_{i=1}^{16} h_i(t)A_i \quad , \quad B_z(t) = \sum_{i=1}^{16} h_i(t)B_i$$

IV.6.2. Comparaison de la trajectoire réelle avec la trajectoire de référence:

La figure IV.7 illustre la trajectoire du robot obtenue en utilisant à la fois le contrôleur TS-MPC et le contrôleur NMPC, comparée à la trajectoire de référence souhaitée. Comme on peut l'observer, les deux contrôleurs sont capables de suivre avec précision la trajectoire complexe non circulaire cependant, le TS-MPC offre une réponse plus fluide et plus stable, en particulier dans les régions où la courbure varie.

Cela est évident à partir des oscillations réduites dans la trajectoire réelle et de son alignement plus étroit avec le chemin de référence tout au long de la période de simulation. L'utilisation du modèle polytopique (T-S) dans les calculs internes du contrôleur lui permet de représenter efficacement les dynamiques non linéaires avec une charge de calcul moindre, ce qui sera démontré plus en détail dans l'analyse du temps de calcul.

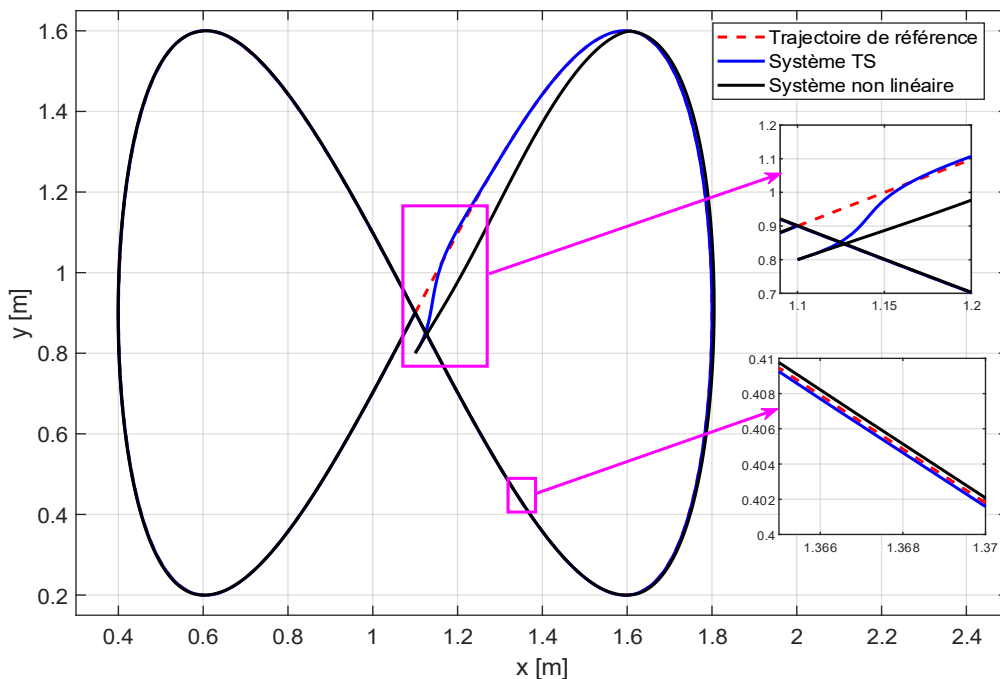


Figure IV.7. Comparaison entre la trajectoire de référence et la trajectoire réelle du robot obtenue en utilisant TS-MPC et NMPC.

IV.6.3. Comparaison des erreurs de suivi e_x , e_y , e_ϕ entre les deux contrôleurs:

Les figures IV.8 et IV.9 présentent l'évolution de l'erreur longitudinale e_x et de l'erreur latérale e_y , respectivement. Il est évident que le contrôleur Nonlinéaire-MPC atteint une réponse initiale très rapide, mais il souffre de grandes oscillations avant d'atteindre l'état stationnaire. Ce comportement est dû à la nature de la solution d'optimisation non linéaire, qui peut changer brusquement à mesure que l'état du robot évolue.

En revanche, le contrôleur TS-MPC présente une réponse plus douce avec significativement moins d'oscillations, convergeant vers une erreur nulle de manière progressive et constante sans fluctuations brusques.

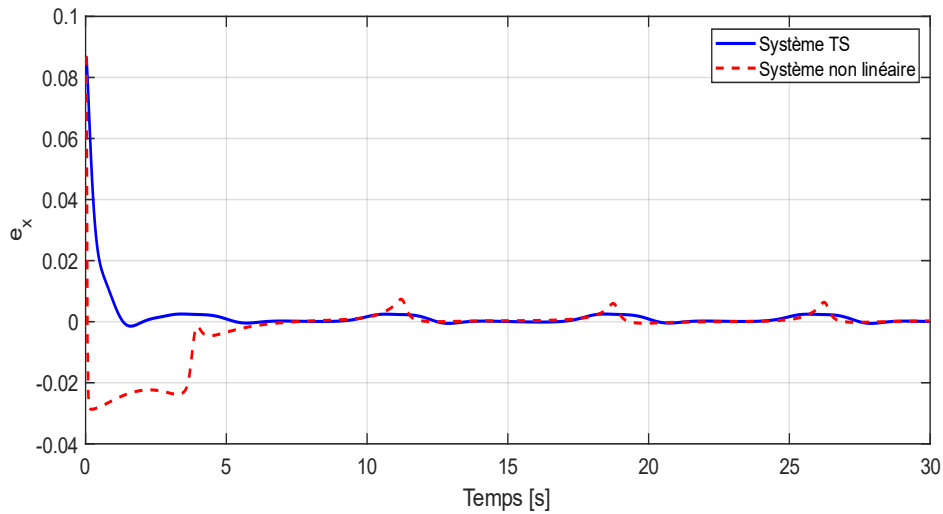


Figure IV.8. Évolution de l'erreur de suivi longitudinal $e_x(t)$ en utilisant TS-MPC et NMPC

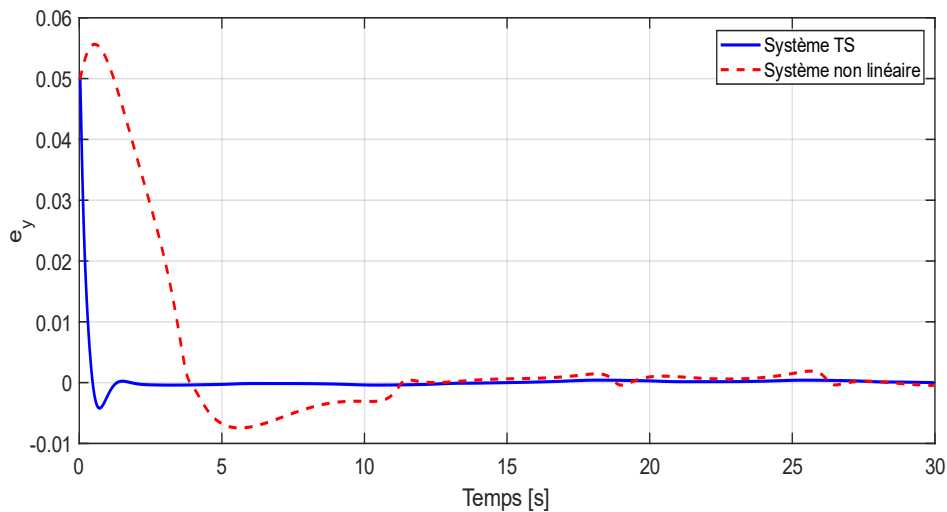


Figure IV.9. Évolution de l'erreur de suivi latéral $e_y(t)$ en utilisant TS-MPC et NMPC.

La figure IV.10 montre l'erreur d'orientation e_θ . On peut observer que le contrôleur TS-MPC réalise une stabilisation plus rapide et plus robuste par rapport au NMPC, en particulier dans les régions où la courbure du chemin de référence change.

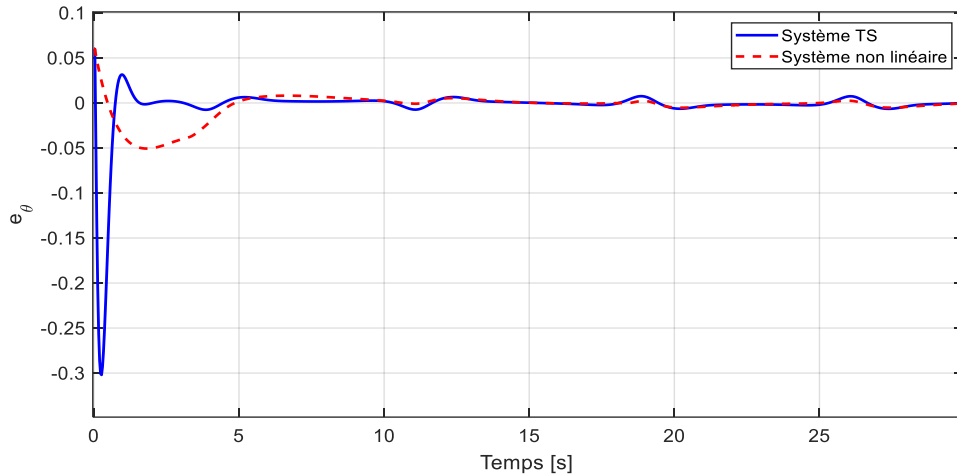


Figure IV.10. Comparaison de l'erreur de cap $e_\theta(t)$ entre TS-MPC et NMPC.

IV.6.4. Comparaison des entrées de contrôle de référence et appliquées:

La figure IV.11 montre la comparaison entre la vitesse linéaire de référence et l'entrée de contrôle générée par le contrôleur TS-MPC. Comme on peut l'observer, la sortie du contrôleur suit de près le profil de référence avec des variations douces et seulement des écarts mineurs, démontrant la capacité du TS-MPC à reproduire avec précision la trajectoire de vitesse souhaitée de manière stable et cohérente tout au long de la période de simulation.

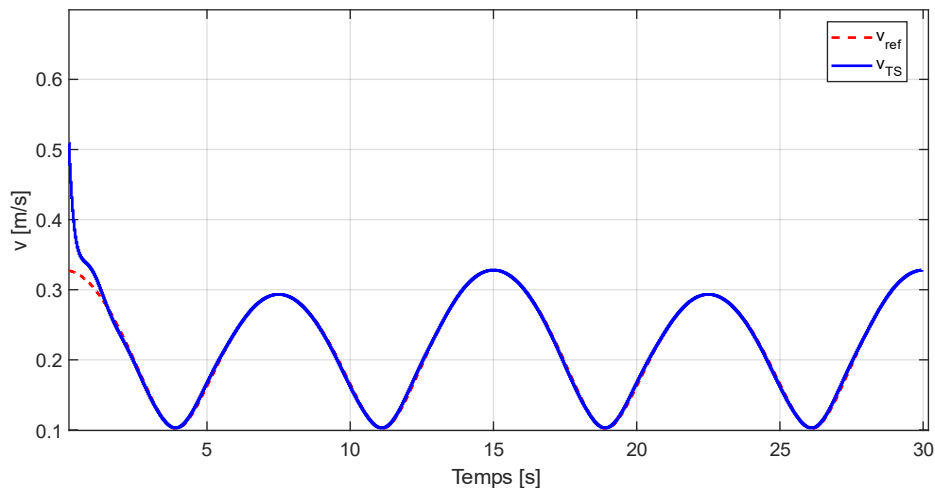


Figure IV.11. Comparaison de la vitesse linéaire de référence $v_r(t)$ avec l'entrée de contrôle générée par le contrôleur TS-MPC

La figure IV.12 montre la comparaison entre la vitesse angulaire de référence et l'entrée de contrôle angulaire produite par le contrôleur TS-MPC. Les résultats indiquent que le contrôleur suit efficacement le signal de référence, capturant à la fois sa forme et ses variations temporelles --sans introduire de changements brusques, ce qui reflète la douceur et la stabilité fournies par le modèle polytopique T-S utilisé dans la structure de contrôle.

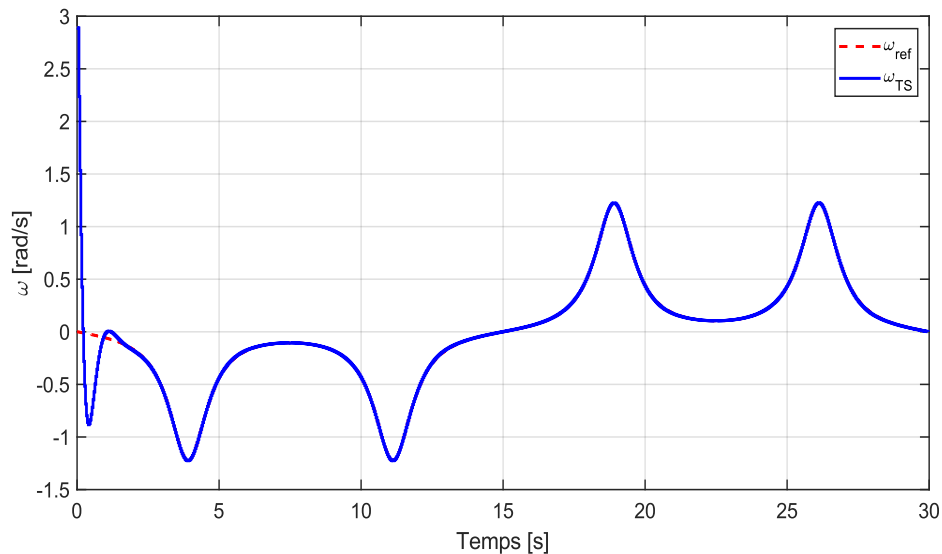


Figure IV.12. Comparaison de la vitesse angulaire de référence $\omega_r(t)$ et de l'entrée de contrôle TS-MPC

La figure IV.13 montre la comparaison entre la vitesse linéaire de référence et l'entrée de contrôle appliquée par le contrôleur NMPC. Bien que le contrôleur non linéaire suive avec succès la tendance générale du signal de référence, l'action de contrôle générée présente des oscillations notables et des variations plus brusques, en particulier pendant les intervalles où la vitesse change rapidement, reflétant la sensibilité de l'optimisation non linéaire à l'évolution de l'état.

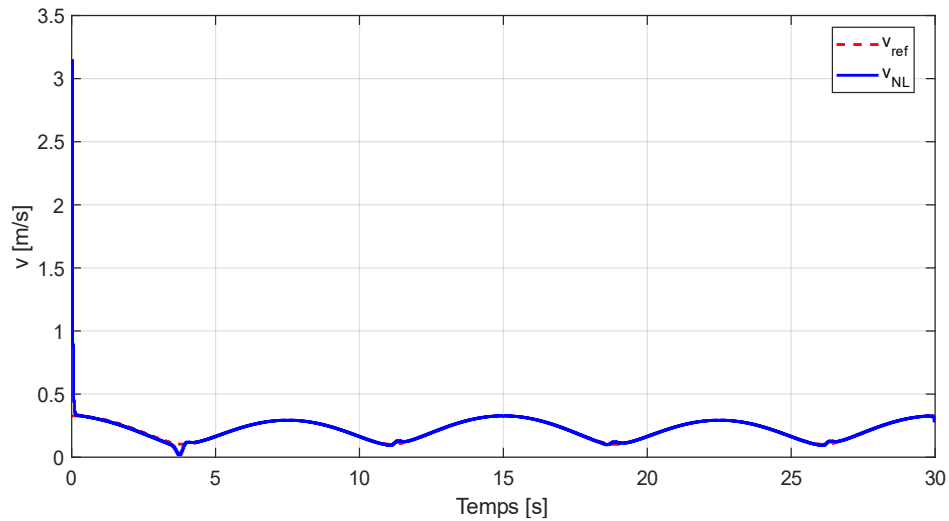


Figure IV.13. Comparaison de la vitesse linéaire de référence $v_r(t)$ avec l'entrée de contrôle générée par le contrôleur NMPC.

La figure IV.14 illustre la comparaison entre la vitesse angulaire de référence $\omega_{ref}(t)$ et l'entrée de contrôle angulaire générée par le MPC non linéaire. Les résultats montrent que le contrôleur réalise un suivi très précis, car le signal de contrôle se superpose presque complètement à la référence tout au long de la simulation. À l'exception de petites déviations transitoires initiales, le NMPC converge rapidement et maintient un suivi précis même lors de changements rapides de la vitesse angulaire.

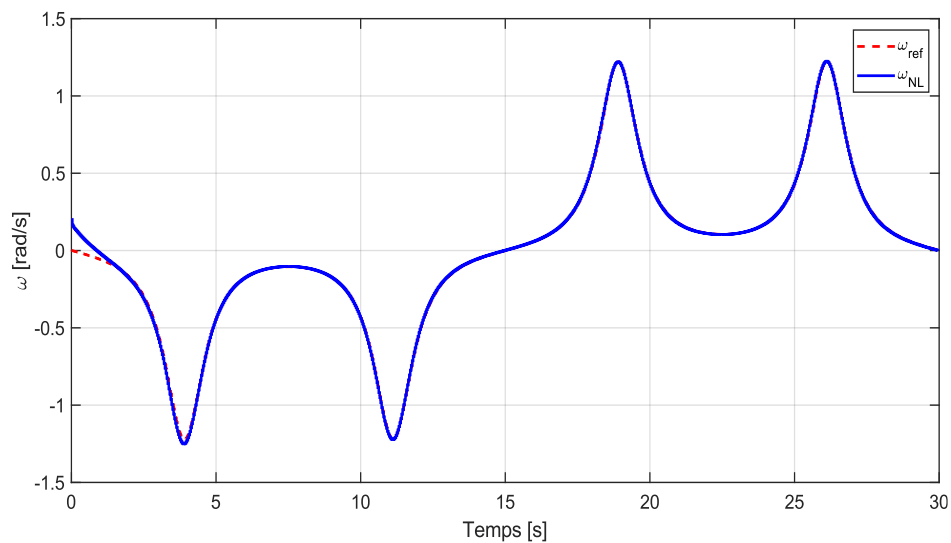


Figure IV.14. Comparaison de la vitesse angulaire de référence $\omega_r(t)$ et de l'entrée de contrôle NMPC

IV.6.5. Analyse des indices de performance (ISE, RMSE):

La figure IV.15 présente l'erreur quadratique moyenne (RMSE) pour les états de suivi e_x , e_y , et e_θ sous les deux contrôleurs TS-MPC et NMPC. Les résultats indiquent que le TS-MPC atteint un RMSE inférieur dans les états linéaires e_x et e_y , démontrant une meilleure précision dans le suivi de position par rapport au contrôleur non linéaire. En revanche, le NMPC montre une performance supérieure dans l'erreur d'orientation e_θ , où il obtient une valeur RMSE plus petite que le TS-MPC. La figure met en évidence les forces complémentaires des deux contrôleurs en fonction de la nature de la variable de suivi.

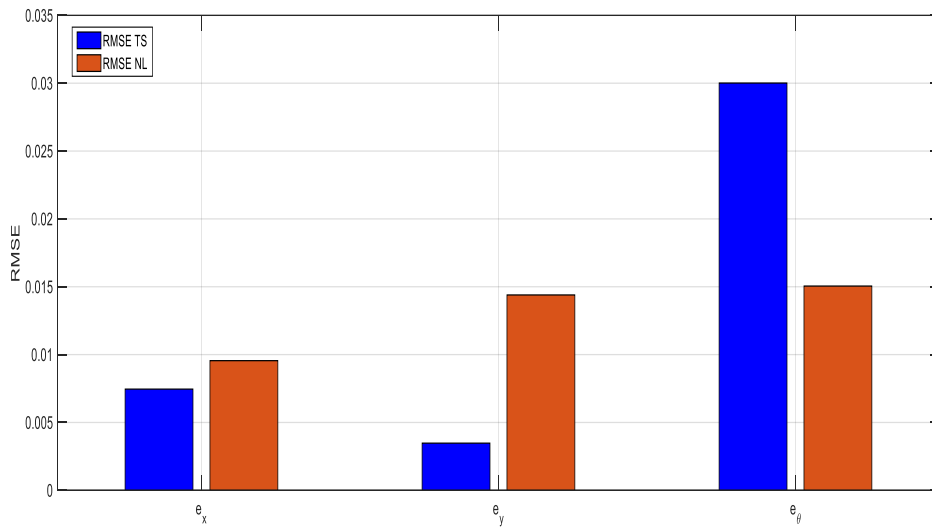


Figure IV.15. Comparaison des performances basée sur l'erreur quadratique moyenne (RMSE) des erreurs de suivi pour TS-MPC et NMPC.

Un examen détaillé de l'Intégrale de l'Erreur au Carré (ISE) pour chaque état de suivi révèle des caractéristiques de performance distinctes pour les deux contrôleurs. Le TS-MPC atteint des valeurs ISE nettement inférieures dans les composantes d'erreur linéaire, avec $ISE(e_x) = 0,00154$ et $ISE(e_y) = 0,00032$, comparées à $0,00261$ et $0,00617$ pour le MPC Non-linéaire, respectivement. Cela indique que le modèle polytopique T-S offre une représentation très précise de la dynamique translationnelle du robot, permettant au TS-MPC de réduire plus efficacement l'erreur accumulée dans les directions longitudinale et latérale.

À l'inverse, le MPC non linéaire démontre une performance supérieure dans l'erreur d'orientation, atteignant $ISE(e_\theta) = 0,00673$, ce qui est nettement inférieur à la valeur correspondante du TS-MPC de $0,02696$. Ce comportement est attendu, car le contrôleur non linéaire intègre explicitement la dynamique rotationnelle non linéaire complète dans son problème d'optimisation, ce qui lui permet de corriger les écarts angulaires plus rapidement tout au long de la trajectoire.

Dans l'ensemble, l'analyse ISE par État met en évidence un profil de performance complémentaire:

- TS-MPC excelle dans le suivi spatial (de position), offrant un contrôle plus fluide et plus précis du mouvement translationnel.
- Le MPC non linéaire fonctionne mieux dans la régulation de l'orientation, bénéficiant de sa capacité à exploiter pleinement les dynamiques de rotation non linéaires.

Ces perspectives offrent une compréhension plus approfondie des forces de chaque contrôleur et fournissent des conseils précieux lors de la sélection de l'architecture de contrôle la plus appropriée pour des scénarios de navigation robotique spécifiques.

IV.6.6. Analyse du temps de calcul:

Les simulations ont été effectuées sous MATLAB R2022b sur la même plateforme matérielle (AMD Ryzen 5 3400G, 16 Go RAM) pour les contrôleurs TS-MPC et NMPC, garantissant ainsi une comparaison équitable.

La figure IV.16 présente le temps de calcul par étape de contrôle pour les contrôleurs TS-MPC et NMPC. Comme le montre la figure, le TS-MPC atteint un temps de calcul nettement inférieur, restant constant tout au long des environ 900 étapes de contrôle.

En revanche, le NMPC présente des temps de calcul irréguliers avec des pics notables, causés par la nécessité de résoudre des problèmes d'optimisation non linéaire à chaque étape.

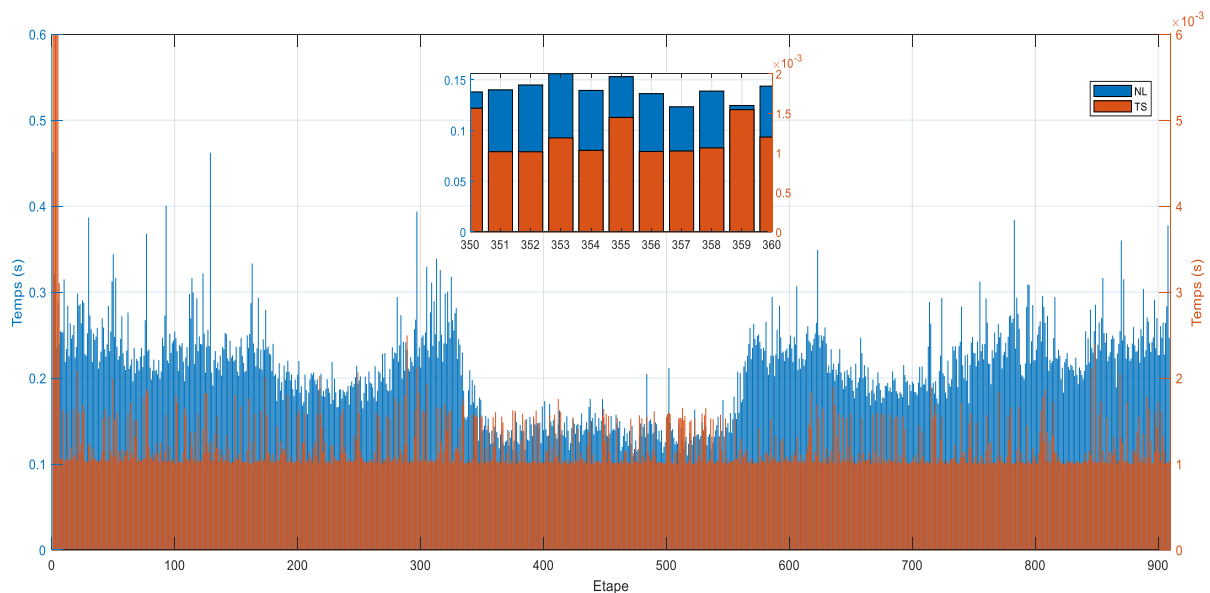


Figure IV.16. Temps de calcul par étape pour TS-MPC et NMPC sur l'horizon de simulation complet.

Suite aux résultats détaillés de la simulation et à l'analyse comparative entre les contrôleurs TS–MPC et NMPC, il devient possible d'évaluer clairement leur performance pratique lorsqu'ils traitent des trajectoires de référence complexes et la dynamique non linéaire du robot à entraînement différentiel. Les résultats mettent en évidence des différences significatives entre les deux méthodes en termes de précision de suivi, de douceur des actions de contrôle, de robustesse aux variations de courbure et d'efficacité computationnelle. Cette évaluation complète offre une compréhension plus approfondie des forces et des limitations de chaque stratégie de contrôle et clarifie leur adéquation aux conditions de fonctionnement en temps réel ou en évolution rapide.

IV.7. Conclusion:

Ce chapitre a présenté un cadre complet et rigoureux pour modéliser, analyser et contrôler un robot mobile à entraînement différentiel en utilisant à la fois sa formulation cinématique non linéaire et une représentation polytopique exacte de Takagi–Sugeno (T–S). Le modèle d'erreur de suivi de trajectoire non linéaire a été transformé en une combinaison convexe de seize systèmes linéaires de sommet, garantissant une reproduction exacte de la dynamique non linéaire dans la région de fonctionnement définie. Les résultats de validation en particulier les erreurs instantanées proches de zéro et les valeurs extrêmement petites de RMSE et ISE rapportées dans le Tableau IV.1—ont démontré que la structure polytopique T–S atteint une correspondance presque parfaite avec le véritable comportement non linéaire, confirmant son adéquation pour la conception de contrôle avancée.

En s'appuyant sur cette représentation précise, une stratégie de contrôle prédictif par modèle (TS–MPC) garantissant la stabilité a été développée en intégrant une matrice de Lyapunov commune comme pénalité terminale. Cette construction garantissait la stabilité en boucle fermée à travers toutes les combinaisons convexes possibles des dynamiques des sommets. Le contrôleur résultant fonctionnait sur un modèle linéaire variable dans le temps et mélangé, tout en maintenant une traçabilité computationnelle, ce qui le rendait bien adapté aux applications robotiques en temps réel.

L'étude de simulation comparative entre le TS–MPC et le MPC non linéaire standard a fourni un aperçu plus approfondi des caractéristiques de performance des deux approches. Les résultats de suivi de trajectoire (Figures IV.6– IV.9) ont montré que, bien que les deux contrôleurs aient réussi à suivre le chemin de référence, le TS–MPC a obtenu des réponses nettement plus fluides avec moins d'oscillations dans les erreurs longitudinales et latérales. Le NMPC, bien que très précis—en particulier dans le suivi de la vitesse angulaire montré dans les Figures IV.12– IV.13—a montré des variations de contrôle plus nettes et une sensibilité plus élevée aux changements rapides de courbure. De plus, les comparaisons des

entrées de contrôle (Figures IV.10– IV.12) ont mis en évidence la supériorité de la douceur et de la robustesse de l'action du TS–MPC.

Un avantage clé de l'approche TS–MPC a été confirmé dans l'analyse du temps de calcul (Figure IV.15), où le contrôleur basé sur TS a maintenu un temps d'exécution constamment bas par étape de contrôle, contrairement au NMPC, qui a montré des pics irréguliers et significativement plus importants en raison de l'optimisation non linéaire répétée. Cet aspect est crucial pour les applications embarquées de robots mobiles qui nécessitent des performances de contrôle prévisibles et en temps réel.

Dans l'ensemble, le chapitre a démontré que le cadre de modélisation T–S, combiné à une conception MPC basée sur Lyapunov, offre une alternative puissante au MPC non linéaire classique. Il offre des performances de suivi comparables (et dans certains cas supérieures), des entrées de contrôle plus fluides et une efficacité computationnelle considérablement améliorée. Ces résultats établissent la formulation TS–MPC comme un candidat solide pour une mise en œuvre réelle dans des systèmes de navigation autonomes nécessitant à la fois précision et faisabilité en temps réel.

Conclusion générale

Les travaux présentés dans cette thèse se concentrent sur le problème du contrôle des systèmes dynamiques non linéaires et soumis à des contraintes, à travers le développement d'un cadre intégré basé sur la modélisation floue de type Takagi–Sugeno (T–S) combinée au contrôle prédictif (MPC), afin d'assurer des performances élevées, une stabilité fiable et une mise en œuvre en temps réel. La valeur essentielle de cette recherche se justifie par les défis croissants que présentent les systèmes contemporains, lesquels se distinguent par leur complexité intrinsèque, leur non-linéarité, leur incertitude inhérente, ainsi que par l'impératif de concevoir des algorithmes de contrôle aptes à intégrer de manière efficiente diverses catégories de contraintes.

Dans les chapitres successifs, ce mémoire a présenté un cadre méthodologique complète combinant analyse théorique, techniques de modélisation avancées et formulation de lois de commande, avant d'être appliqué et évalué sur un cas pratique constitué d'un robot mobile différentiel. Cette thèse a suivi une méthodologie systématique et progressive, articulée autour de quatre étapes principales. La première a consisté à jeter les bases théoriques par une revue exhaustive du contrôle prédictif, de la logique floue, de la modélisation T-S et des mécanismes de leur intégration dans un cadre de Contrôle Prédictif Flou. Ensuite, la conception d'un contrôleur prédictif flou pour les systèmes linéaires et à paramètres variables, en réalisant un ajustement dynamique des matrices de coût Q et R à l'aide d'un système de règles floues fondé sur l'état du système et les erreurs instantanées. Puis, dans le troisième chapitre, l'adoption de méthodes de modélisation basées sur la logique floue pour les systèmes non linéaires (la modélisation analytique de type T–S, la transformation sectorielle (NST) permettant d'obtenir une représentation polytopique et la modélisation floue fondée sur les données à l'aide de l'algorithme LM). Enfin, la construction d'un modèle polytopique de haute précision pour le robot mobile et son utilisation dans la conception d'un contrôleur TS–MPC stable fondé sur une matrice de Lyapunov commune, suivie d'une comparaison avec le contrôleur NMPC classique.

Les résultats ont montré que l'intégration d'un système flou finement réglé au sein du cadre MPC améliore le suivi de trajectoire, renforce la stabilité et réduit les oscillations, augmenter la robustesse face aux perturbations, et réduire l'effort de commande jusqu'à 80 % par rapport au MPC. De plus, les approches F1MPC, et en particulier F2MPC, ont surpassé le MPC dans presque tous les indicateurs de performance. En ce qui concerne la modélisation unifiée des systèmes non linéaires, les comparaisons ont donné des résultats importants, à savoir:

- NST: offre la meilleure représentation mathématique et la plus grande précision ($MSE \approx 10^{-33}$), ce qui le rend particulièrement adapté au MPC et aux méthodes basées sur les LMI.
- T-S analytique: offre une précision remarquable avec une complexité intermédiaire, établissant un équilibre efficient entre la simplicité et la fidélité du modèle.
- T-S basé sur l'apprentissage (LM-Based) Indique la plus grande aptitude à la généralisation et les résultats les plus élevés sur les données ($MSE \approx 10^{-8}$).

Ces modèles ont démontré leur aptitude à représenter le système non linéaire avec une grande précision tout en conservant fidèlement son comportement dynamique réel. Et d'autre part, concernant la construction d'un modèle composé de seize sommets polytopiques, issu de la combinaison des variables non linéaires, les éléments suivants ont été démontrés:

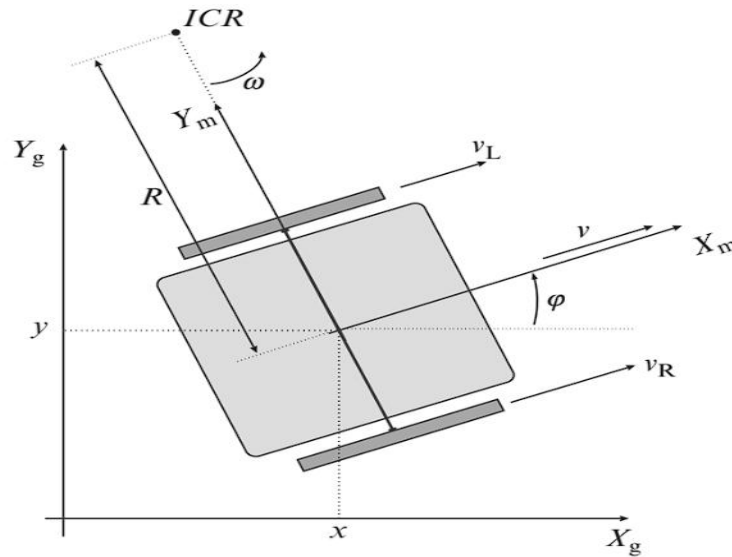
- des valeurs de RMSE et d'ISE quasi nulles
- une correspondance pratiquement parfaite entre le modèle polytopique et le système non linéaire
- une excellente capacité à représenter la dynamique des erreurs de suivi

Cette thèse met en évidence que l'intégration entre la logique floue et les techniques de contrôle prédictif offre un cadre cohérent et efficace pour traiter des systèmes complexes et non linéaires. Le contrôleur TS-MPC a également démontré des performances remarquables en termes de précision, de fluidité des signaux et d'efficacité computationnelle, ce qui en fait une approche prometteuse pour un large éventail d'applications industrielles, notamment en robotique mobile ainsi que dans d'autres systèmes avancés. Par ailleurs, ce travail est loin d'être achevé, plusieurs perspectives sont envisagées:

1. Les problèmes de robustesse des systèmes basés sur l'intelligence artificielle seront abordés pour un type plus général de processus non linéaires.
2. Coupler TS-MPC avec des réseaux de neurones ou l'apprentissage par renforcement pour une modélisation data-driven et une adaptation en ligne des paramètres du contrôleur.
3. Développer des algorithmes optimisés et des techniques de réduction de modèle (polytope, règles) pour applications haute fréquence (drones, robotique agile).

Annexes

Annexe A : Modélisation cinématique du robot mobile différentiel



A.1 Définitions et notations

On considère un robot mobile à deux roues motrices indépendantes.

- V : vitesse linéaire du robot
- ω : vitesse angulaire
- V_L, V_R : vitesses linéaires des roues gauche et droite
- θ_L, θ_R : vitesses angulaires des roues
- r : rayon des roues
- L : distance entre les deux roues
- (x, y, φ) : position et orientation dans le repère global
- (x_m, y_m) : coordonnées dans le repère local
- ICR : centre instantané de rotation

A.2 Relations cinématiques des roues

Les vitesses des roues sont données par:

$$V_L = r\theta_L, V_R = r\theta_R$$

La relation entre les vitesses et le rayon de rotation R est:

$$V_L = (R - \frac{L}{2})\omega, V_R = (R + \frac{L}{2})\omega$$

A.3 Vitesse linéaire et angulaire du robot

On obtient les expressions fondamentales :

$$\omega = \frac{V_R - V_L}{L}$$

$$V = \frac{V_L + V_R}{2}$$

A.4 Modèle cinématique dans le repère global

Les équations du mouvement sont :

$$\begin{aligned}\dot{x} &= V \cos \varphi \\ \dot{y} &= V \sin \varphi \\ \dot{\varphi} &= \omega\end{aligned}$$

A.5 Modèle discret (intégration d'Euler)

En utilisant une période d'échantillonnage T_s :

$$\begin{aligned}x(k+1) &= x(k) + T_s V(k) \cos(\varphi(k)) \\ y(k+1) &= y(k) + T_s V(k) \sin(\varphi(k)) \\ \varphi(k+1) &= \varphi(k) + T_s \omega(k)\end{aligned}$$

Annexe B : Problème de suivi de trajectoire

B.1 Définition de l'erreur:

Soit :

- $R = [x_r, y_r, \varphi_r]^T$: trajectoire de référence
- $R_m = [x, y, \varphi]^T$: état du robot

Le vecteur d'erreur est :

$$e = \begin{bmatrix} x_e \\ y_e \\ \varphi_e \end{bmatrix}$$

avec :

$$\begin{aligned}x_e &= (x_r - x) \cos \varphi + (y_r - y) \sin \varphi \\ y_e &= -(x_r - x) \sin \varphi + (y_r - y) \cos \varphi \\ \varphi_e &= \varphi_r - \varphi\end{aligned}$$

B.2 Dynamique de l'erreur

Les équations différentielles de l'erreur sont :

$$\begin{aligned}\dot{x}_e &= \omega y_e - V + V_r \cos \varphi_e \\ \dot{y}_e &= -\omega x_e + V_r \sin \varphi_e \\ \dot{\varphi}_e &= \omega_r - \omega\end{aligned}$$

B.3 Représentation matricielle

$$\dot{e} = \begin{bmatrix} 0 & \omega & 0 \\ -\omega_r & 0 & V_r \\ 0 & 0 & 0 \end{bmatrix} e + \begin{bmatrix} -1 & 0 \\ 0 & 0 \\ 0 & -1 \end{bmatrix} u_B$$

avec :

$$u_B = \begin{bmatrix} V - V_r \cos \varphi_e \\ \omega - \omega_r \end{bmatrix}$$

Bibliographie

- [1] A. Khaki-Sedigh, "Control Systems: a historical and philosophical perspective: History and philosophy of control systems". Springer Nature, 2025.
- [2] X. Li, M. Yan, X. Zhang, et al., "Efficient data-driven predictive control of nonlinear systems: A review and perspectives," *Digital Chemical Engineering*, Art. no. 100219, 2025.
- [3] P. D. Christofides, R. Scattolini, D. M. De la Peña, et al., "Distributed model predictive control: A tutorial review and future research directions," *Computers & Chemical Engineering*, vol. 51, pp. 21–41, 2013.
- [4] M. Schwenzer, M. Ay, T. Bergs, and D. Abel, "Review on model predictive control: An engineering perspective," *International Journal of Advanced Manufacturing Technology*, vol. 117, no. 5–6, pp. 1327–1349, Nov. 2021, doi: 10.1007/s00170-021-07682-3.
- [5] J. M. Da Costa Sousa and U. Kaymak, "Model predictive control using fuzzy decision functions," *IEEE Transactions on Systems Man and Cybernetics Part B (Cybernetics)*, vol. 31, no. 1, pp. 54–65, Jan. 2001, doi: 10.1109/3477.907564.
- [6] D. Lee and J. Hu, "Local model predictive control for T–S fuzzy systems," *IEEE Transactions on Cybernetics*, vol. 47, no. 9, pp. 2556–2567, Sep. 2017, doi: 10.1109/TCYB.2016.2616100.
- [7] Y.-Y. Liu and A.-L. Barabási, "Control principles of complex systems," *Reviews of Modern Physics*, vol. 88, no. 3, Sep. 2016, doi: 10.1103/revmodphys.88.035006.
- [8] Y.-G. Xi, D.-W. Li, and S. Lin, "Model Predictive Control — Status and challenges," *Acta Automatica Sinica*, vol. 39, no. 3, pp. 222–236, Mar. 2013, doi: 10.1016/s1874-1029(13)60024-5.
- [9] M. G. Forbes, R. S. Patwardhan, H. Hamadah, and R. B. Gopaluni, "Model Predictive Control in Industry: Challenges and opportunities," *IFAC-PapersOnLine*, vol. 48, no. 8, pp. 531–538, Jan. 2015, doi: 10.1016/j.ifacol.2015.09.022.
- [10] I. Škrjanc and D. Matko, "Fuzzy predictive functional control in the state space domain," *Journal of Intelligent and Robotic Systems*, vol. 31, no. 3, pp. 283–297, 2001.

- [11] C.-L. Su and S.-Q. Wang, "Robust model predictive control for discrete uncertain nonlinear systems with time-delay via fuzzy model," *Journal of Zhejiang University. Science A*, vol. 7, no. 10, pp. 1723–1732, Sep. 2006, doi: 10.1631/jzus.2006.a1723.
- [12] X. Chang and G. Yang, "Fuzzy robust constrained model predictive control for nonlinear systems," *Asian Journal of Control*, vol. 13, no. 6, pp. 947–955, Jun. 2010, doi: 10.1002/asjc.221.
- [13] U. Kaymak and J. M. Sousa, "Model based fuzzy predictive control applied to a simulated gantry crane," in *Proc. 2nd Asian Control Conf.*, Seoul, Korea, vol. 3, July 1997, pp. 455–458.
- [14] S. Mollov, R. Babuska, J. Abonyi, and H. B. Verbruggen, "Effective optimization for fuzzy model predictive control," *IEEE Transactions on Fuzzy Systems*, vol. 12, no. 5, pp. 661–675, Oct. 2004, doi: 10.1109/tfuzz.2004.834812.
- [15] A. I. Propoi, "Use of LP methods for synthesizing sampled-data automatic systems", *Automn Remote Control*, Vol. 24, No. 7, pp. 837–844, 1963.
- [16] Qin, S. J. & Badgwell, T. A. "A survey of industrial model predictive control technology". *Control Eng. Pract.* Vol. 11 No. 7, pp. 733–764, 2003.
- [17] J.-M. Coron, L. Grüne, and K. Worthmann, "Model predictive control, cost controllability, and homogeneity," *SIAM Journal on Control and Optimization*, vol. 58, no. 5, pp. 2979–2996, 2020.
- [18] D. M. Wolpert and Z. Ghahramani, "Computational principles of movement neuroscience," *Nature Neuroscience*, vol. 3, no. 11, pp. 1212–1217, 2000.
- [19] R. Grush, "The emulation theory of representation: Motor control, imagery, and perception," *Behavioral and Brain Sciences*, vol. 27, no. 3, pp. 377–396, 2004.
- [20] C. E. Garcia, D. M. Prett, and M. Morari, "Model predictive control: Theory and practice—A survey," *Automatica*, vol. 25, no. 3, pp. 335–348, 1989.
- [21] S. J. Qin and T. A. Badgwell, "A survey of industrial model predictive control technology," *Control Engineering Practice*, vol. 11, no. 7, pp. 733–764, 2003.
- [22] B. El Khansa, "Commande prédictive avec contraintes: Application aux systèmes non linéaires," Ph.D. dissertation, Dept. Génie Électrique, École Nationale Polytechnique - Alger, Algeria, 2016.
- [23] I. Bombard, B. Da Silva, P. Dufour, P. Laurent, and J. Lieto, "Contrôle par commande prédictive d'un procédé de cuisson sous infrarouge de peintures en poudre," in *Proc. Société Française de Thermique (SFT)*, 2008.

- [24] E. F. Camacho and C. Bordons, *Model Predictive Control*, 2nd ed. Berlin, Germany: Springer, 2007.
- [25] J. B. Rawlings, D. Q. Mayne, and M. M. Diehl, *Model Predictive Control: Theory, Computation, and Design*, 2nd ed. Madison, WI, USA: Nob Hill Publishing, 2017.
- [26] J. Richalet, A. Rault, J. L. Testud, and J. Papon, "Algorithmic control of industrial processes," in *Proc. 4th IFAC Symp. Identification and System Parameter Estimation*, 1976, pp. 1119–1167.
- [27] C. R. Cutler and B. L. Ramaker, "Dynamic matrix control—A computer control algorithm," in *Proc. AIChE Annual Meeting*, Houston, TX, USA, 1979.
- [28] B. E. Ydstie, "Extended horizon adaptive control," in *Proc. 9th IFAC World Congress*, vol. 7, Budapest, Hungary, 1984, pp. 133–137.
- [29] R. M. C. De Keyser and A. R. Van Cauwenberghe, "Extended prediction self-adaptive control," in *Proc. IFAC Symp. Identification and System Parameter Estimation*, York, U.K., 1985.
- [30] D. W. Clarke, C. Mohtadi, and P. S. Tuffs, "Generalized predictive control—I: The basic algorithm," *Automatica*, vol. 23, no. 2, pp. 137–148, 1987.
- [31] J. Richalet, A. Rault, J. L. Testud, and J. Papon, "Model predictive heuristic control: Application to industrial processes," *Automatica*, vol. 14, pp. 413–428, 1978.
- [32] D. W. Clarke and R. Scattolini, "Constrained receding horizon predictive control," *Proc. IEE, Part D: Control Theory and Applications*, vol. 138, no. 4, pp. 347–354, 1991.
- [33] N. L. Ricker, "Model-predictive control: State of the art," in *Proc. 4th Int. Conf. Chemical Process Control*, Padre Island, TX, USA, 1991, pp. 271–296.
- [34] S. Lall and K. Glover, "A game theoretic approach to moving horizon control," in *Advances in Model-Based Predictive Control*. Oxford, U.K.: Oxford University Press, 1994, pp. 294–308.
- [35] H. Chen, C. W. Scherer, and F. Allgöwer, "A game theoretic approach to nonlinear robust receding horizon control of constrained systems," in *Proc. American Control Conf.*, 1997, pp. 3073–3077.
- [36] J. Darlington, C. C. Pantelides, B. Rustem, and B. A. Tanyi, "Decreasing the sensitivity of open-loop optimal solutions in decision making under uncertainty," *European Journal of Operational Research*, vol. 121, no. 2, pp. 343–362, 2000.

- [37] B. Rustem, "Stochastic and robust control of nonlinear economic systems," *European Journal of Operational Research*, vol. 73, pp. 304–318, 1994.
- [38] M. Kothare, V. Balakrishnan, and M. Morari, "Robust constrained model predictive control using linear matrix inequalities," *Automatica*, vol. 32, no. 10, pp. 1361–1379, 1996.
- [39] L. Magni, H. Nijmeijer, and A. J. van der Schaft, "A receding horizon approach to the nonlinear H^∞ control problem," *Automatica*, vol. 37, no. 5, pp. 429–435, 2001.
- [40] P. Tatjewski, *Advanced Control of Industrial Processes: Structures and Algorithms*. London, U.K.: Springer-Verlag, 2007.
- [41] J. C. Hennet, "Discrete-time constrained linear systems," *Control and Dynamical Systems*, vol. 71, pp. 157–213, 1995.
- [42] K. Bdirina, D. Djoudi, and M. Lagoun, "Linear predictive control with state variable constraints," in *Proc. 6th Global Conf. Optimization, AIP Conf. Proc.*, vol. 1499, 2012, pp. 182–186.
- [43] J. M. Maciejowski, *Predictive Control with Constraints*. London, U.K.: Pearson Education Limited, 2002.
- [44] E. H. Mamdani and S. Assilian, "An experiment in linguistic synthesis with a fuzzy logic controller," *International Journal of Man-Machine Studies*, vol. 7, no. 1, pp. 1–13, 1975.
- [45] L. A. Zadeh, "Fuzzy sets," *Information and Control*, vol. 8, no. 3, pp. 338–353, 1965.
- [46] T.-P. Hong and C.-Y. Lee, "Induction of fuzzy rules and membership functions from training examples," *Fuzzy Sets and Systems*, vol. 84, no. 1, pp. 33–47, 1996.
- [47] C.-T. Lin and C. S. G. Lee, "Neural-network-based fuzzy logic control and decision system," *IEEE Transactions on Computers*, vol. 40, no. 12, pp. 1320–1336, 1991.
- [48] Y. Yuan and M. J. Shaw, "Induction of fuzzy decision trees," *Fuzzy Sets and Systems*, vol. 69, no. 2, pp. 125–139, 1995.
- [49] Y. Yuan and H. Zhuang, "A genetic algorithm for generating fuzzy classification rules," *Fuzzy Sets and Systems*, vol. 84, no. 1, pp. 1–19, 1996.
- [50] V. Dimitrov and V. Korotkich, *Fuzzy Logic: A Framework for the New Millennium*. New York, NY, USA: Springer, 2002.

- [51] H.-F. Wang and C.-C. Fu, "A generalization of fuzzy goal programming with preemptive structure," *Computers & Operations Research*, vol. 24, no. 9, pp. 819–828, 1997.
- [52] H. Asmuni, "Fuzzy methodologies for automated university timetabling solution construction and evaluation," Ph.D. dissertation, University of Nottingham, Nottingham, U.K., 2008.
- [53] T.-P. Hong and C.-Y. Lee, "Induction of fuzzy rules and membership functions from training examples," *Fuzzy Sets and Systems*, vol. 84, no. 1, pp. 33–47, 1996.
- [54] N. K. Kasabov, *Foundations of Neural Networks, Fuzzy Systems, and Knowledge Engineering*. Cambridge, MA, USA: MIT Press, 1996.
- [55] L. A. Zadeh, "The concept of a linguistic variable and its application to approximate reasoning—III," *Information Sciences*, vol. 9, no. 1, pp. 43–80, 1975.
- [56] R. Czabanski, M. Jezewski, and J. Leski, "Introduction to fuzzy systems," in *Theory and Applications of Ordered Fuzzy Numbers: A Tribute to Professor Witold Kosiński*. Cham, Switzerland: Springer, 2017, pp. 23–43.
- [57] J. M. Alonso, C. Castiello, L. Magdalena, et al., "An overview of fuzzy systems," in *Explainable Fuzzy Systems: Paving the Way from Interpretable Fuzzy Systems to Explainable AI Systems*. Cham, Switzerland: Springer, 2021, pp. 25–47.
- [58] I. Iancu, "A Mamdani type fuzzy logic controller," *Fuzzy Logic—Controls, Concepts, Theories and Applications*, vol. 15, no. 2, pp. 325–350, 2012.
- [59] M. Sugeno and T. Yasukawa, "A fuzzy-logic-based approach to qualitative modeling," *IEEE Transactions on Fuzzy Systems*, vol. 1, no. 1, pp. 7–31, 1993.
- [60] M. Tabakov, A. B. Chlopowiec, and A. R. Chlopowiec, "A novel classification method using the Takagi–Sugeno model and a type-2 fuzzy rule induction approach," *Applied Sciences*, vol. 13, no. 9, p. 5279, 2023.
- [61] L. Dai, X. Chen, L. Chen, and X. Liu, "Fuzzy predictive control strategy in the application of the industrial furnace temperature control," in *Mechanical Engineering and Control Systems*. Singapore: World Scientific, 2016, pp. 194–197, doi: 10.1142/9789814740616_0042.
- [62] P. M. Vallejo Llamas and P. Vega, "Analytical fuzzy predictive control applied to wastewater treatment biological processes," *Complexity*, 2019, Art. no. 5720185, doi: 10.1155/2019/5720185.

- [63] S. Bouzid, M. Ramdani, and S. Chenikher, "Quality fuzzy predictive control of water in drinking water systems," *Automatic Control and Computer Sciences*, vol. 53, no. 6, pp. 492–501, Nov. 2019, doi: 10.3103/S0146411619060026.
- [64] T. Sardarmehni, A. A. Ashtiani, and M. B. Menhaj, "Fuzzy model predictive control of normalized air-to-fuel ratio in internal combustion engines," *Soft Computing*, vol. 23, no. 15, pp. 6169–6182, 2019, doi: 10.1007/s00500-018-3270-2.
- [65] S. Najjaran, Z. Rahmani, and M. Hassanzadeh, "Fuzzy predictive control strategy for plug-in hybrid electric vehicles over multiple driving cycles," *International Journal of Dynamics and Control*, vol. 10, no. 3, pp. 930–941, 2022, doi: 10.1007/s40435-021-00862-9.
- [66] H. Ahsan and M. D. Mufti, "Sweeping power system stabilisation with a parametric fuzzy predictive control of a generalised energy storage device," *IET Generation, Transmission & Distribution*, vol. 14, no. 25, pp. 6087–6096, Dec. 2020, doi: 10.1049/iet-gtd.2020.0940.
- [67] X. Wang, Q. Wang, and S. Liang, "Predictive control algorithm for urban rail train brake control system based on T–S fuzzy model," *Computers, Materials & Continua*, vol. 64, no. 3, pp. 1859–1867, 2020, doi: 10.32604/cmc.2020.011032.
- [68] Y. Yang and B. Ding, "An iterative optimization approach for fuzzy predictive control," *International Journal of Control, Automation and Systems*, vol. 18, no. 8, pp. 2157–2164, 2020, doi: 10.1007/s12555-019-0488-4.
- [69] Q. Lu, P. Shi, H.-K. Lam, and Y. Zhao, "Interval type-2 fuzzy model predictive control of nonlinear networked control systems," *IEEE Transactions on Fuzzy Systems*, vol. 23, no. 6, pp. 2317–2328, 2015, doi: 10.1109/TFUZZ.2015.2417975.
- [70] H. Xie, J. Wang, and X. Tang, "Constrained model predictive control of interval type-2 T–S fuzzy systems with Markovian packet loss," *Journal of Control, Automation and Electrical Systems*, vol. 28, no. 3, pp. 323–336, 2017, doi: 10.1007/s40313-017-0310-6.
- [71] X. Tang, L. Deng, J. Yu, and H. Qu, "Output feedback predictive control of interval type-2 T–S fuzzy systems with Markovian packet loss," *IEEE Transactions on Fuzzy Systems*, vol. 26, no. 4, pp. 2450–2459, 2018, doi: 10.1109/TFUZZ.2017.2771502.

- [72] R. Antão, A. Mota, and R. E. Martins, "Generalized predictive control using interval type-2 fuzzy models," in Proc. IEEE Int. Conf. Fuzzy Systems (FUZZ-IEEE), Istanbul, Turkey, 2015, pp. 1–8, doi: 10.1109/FUZZ-IEEE.2015.7337967.
- [73] R. Antão, A. Mota, and R. E. Martins, "Model-based control using interval type-2 fuzzy logic systems," *Soft Computing*, vol. 22, pp. 607–620, 2018, doi: 10.1007/s00500-016-2358-9.
- [74] M. Wang, J. A. Paulson, H. Yan, and H. Shi, "An adaptive model predictive control strategy for nonlinear distributed parameter systems using the type-2 Takagi–Sugeno model," *International Journal of Fuzzy Systems*, vol. 18, no. 5, pp. 792–805, 2016, doi: 10.1007/s40815-015-0115-3.
- [75] K. Zou, Q. Zhang, and Z. Zhang, "Parafoil path tracking control based on GPC parameters self-correction with interval type-2 fuzzy logic," in Proc. Chinese Control and Decision Conf. (CCDC), Shenyang, China, 2018, pp. 1144–1149, doi: 10.1109/CCDC.2018.8407301.
- [76] K. Sabahi, S. Ghaemi, and M. A. Badamchizadeh, "Feedback error learning-based type-2 fuzzy neural network predictive controller for a class of nonlinear input delay systems," *Transactions of the Institute of Measurement and Control*, vol. 41, no. 13, pp. 3651–3665, 2019, doi: 10.1177/0142331219834998.
- [77] S. Wang, X. Tang, L. Deng, H. Qu, L. Tian, and C. Tan, "Predictive control for interval type-2 fuzzy system with event-triggered scheme," *Advances in Fuzzy Systems*, vol. 2019, Art. no. 9365767, pp. 1–13, 2019, doi: 10.1155/2019/9365767.
- [78] X. Tang, L. Deng, and H. Qu, "Predictive control for networked interval type-2 T–S fuzzy system via an event-triggered dynamic output feedback scheme," *IEEE Transactions on Fuzzy Systems*, vol. 27, no. 8, pp. 1573–1586, 2019, doi: 10.1109/TFUZZ.2018.2883370.
- [79] A. Mohammadzadeh and T. Kumbasar, "A new fractional-order general type-2 fuzzy predictive control system and its application for glucose level regulation," *Applied Soft Computing*, vol. 91, Art. no. 106241, 2020, doi: 10.1016/j.asoc.2020.106241.
- [80] A. Hamza and M. Ramdani, "Non-PDC interval type-2 fuzzy model predictive microclimate control of a greenhouse," *Journal of Control, Automation and Electrical Systems*, vol. 31, no. 1, pp. 62–72, 2020, doi: 10.1007/s40313-019-00532-4.

- [81] D. Phan et al., "Cascade adaptive MPC with type-2 fuzzy system for safety and energy management in autonomous vehicles: A sustainable approach for future of transportation," *Sustainability*, vol. 13, no. 18, Art. no. 10113, 2021, doi: 10.3390/su131810113.
- [82] J. Tavoosi, M. Shirkhani, A. Abdali, A. Mohammadzadeh, M. Nazari, S. Mobayen, J. H. Asad, and A. Bartoszewicz, "A new general type-2 fuzzy predictive scheme for PID tuning," *Applied Sciences*, vol. 11, no. 21, Art. no. 10392, 2021, doi: 10.3390/app112110392.
- [83] S. Manzoor, M. Mufti, F. I. Bakhsh, and A. Ahmad, "Type-2 fuzzy-based adaptively predictive controlled variable frequency transformer coordinated to SMES for improved load frequency control," *IET Generation, Transmission & Distribution*, vol. 18, no. 3, pp. 609–624, 2024, doi: 10.1049/gtd2.13100.
- [84] S.-Y. Li and Y.-P. Yang, "On-line constrained predictive control algorithm using multi-objective fuzzy optimization and a case study," *Fuzzy Optimization and Decision Making*, vol. 2, no. 2, pp. 123–142, 2003. <https://doi.org/10.1023/A:1023495100920>.
- [85] S. Sedghizadeh and S. Beheshti, "Particle swarm optimization based fuzzy gain scheduled subspace predictive control," *Engineering Applications of Artificial Intelligence*, vol. 67, pp. 331–344, 2018, doi: 10.1016/j.engappai.2017.10.009.
- [86] M. Francisco, Y. Mezquita, S. Revollar, P. Vega, and J. F. De Paz, "Multi-agent distributed model predictive control with fuzzy negotiation," *Expert Systems with Applications*, vol. 129, pp. 68–83, 2019, doi: 10.1016/j.eswa.2019.03.056.
- [87] J. Mao, L. Yang, Y. Hu, K. Liu, and J. Du, "Research on vehicle adaptive cruise control method based on fuzzy model predictive control," *Machines*, vol. 9, no. 8, Art. no. 160, 2021, doi: 10.3390/machines9080160.
- [88] L. Hang and D.-H. Kim, "Enhanced model-based predictive control system based on fuzzy logic for maintaining thermal comfort in IoT smart space," *Applied Sciences*, vol. 8, no. 7, Art. no. 1031, 2018, doi: 10.3390/app8071031.
- [89] F. Jurado, "Predictive control of solid oxide fuel cells using fuzzy Hammerstein models," *Journal of Power Sources*, vol. 158, no. 1, pp. 245–253, 2006, doi: 10.1016/j.jpowsour.2005.08.041.

- [90] J. Yang, X. Li, H.-G. Mou, and L. Jian, "Predictive control of solid oxide fuel cell based on an improved Takagi–Sugeno fuzzy model," *Journal of Power Sources*, vol. 193, no. 2, pp. 699–705, 2009, doi: 10.1016/j.jpowsour.2009.04.022.
- [91] P. Guan, S. Wang, and X.-H. Liu, "Indirect adaptive fuzzy predictive control of hypersonic vehicle," in *Proc. Chinese Control and Decision Conf. (CCDC)*, Guiyang, China, 2013, pp. 215–220, doi: 10.1109/CCDC.2013.6561165.
- [92] C. Chen, L. Pan, S. Liu, L. Sun, and K. Y. Lee, "A sustainable power plant control strategy based on fuzzy extended state observer and predictive control," *Sustainability*, vol. 10, no. 12, Art. no. 4824, 2018, doi: 10.3390/su10124824.
- [93] L. Kong and J. Yuan, "Disturbance-observer-based fuzzy model predictive control for nonlinear processes with disturbances and input constraints," *ISA Transactions*, vol. 90, pp. 74–88, 2019, doi: 10.1016/j.isatra.2018.12.041.
- [94] J. Zhu and S. K. Nguang, "Fuzzy model predictive control with enhanced robustness for nonlinear system via a discrete disturbance observer," *IEEE Access*, vol. 8, pp. 220631–220645, 2020, doi: 10.1109/ACCESS.2020.3043359.
- [95] J. Zhu, "Nonlinear dynamic investigation and anti-bifurcation control of a boiler-turbine unit via dual-mode fuzzy model predictive control strategy," *Journal of the Franklin Institute*, vol. 358, no. 15, pp. 7365–7393, 2021, doi: 10.1016/j.jfranklin.2021.07.029.
- [96] M. Wan, J. Du, and H. Yi, "Dynamic positioning for semi-submersible platform using stable fuzzy model predictive control," *Proc. Inst. Mech. Eng., Part I: Journal of Systems and Control Engineering*, vol. 238, no. 1, pp. 73–86, 2024, doi: 10.1177/09596518231182280.
- [97] M. Lepetič, I. Škrjanc, H. G. Chiacchiarini, and D. Matko, "Predictive functional control based on fuzzy model: Magnetic suspension system case study," *Engineering Applications of Artificial Intelligence*, vol. 16, no. 5–6, pp. 425–430, 2003, doi: 10.1016/S0952-1976(03)00071-X.
- [98] S. Oblak and I. Škrjanc, "A comparison of fuzzy and CPWL approximations in the continuous-time nonlinear model-predictive control of time-delayed Wiener-type systems," *Journal of Intelligent & Robotic Systems*, vol. 47, no. 2, pp. 125–137, 2006, doi: 10.1007/s10846-006-9075-z.

- [99] S. Blažič and I. Škrjanc, "Design and stability analysis of fuzzy model-based predictive control—A case study," *Journal of Intelligent & Robotic Systems*, vol. 49, no. 3, pp. 279–292, 2007, doi: 10.1007/s10846-007-9147-8.
- [100] Y. Ma and Y. Cai, "A fuzzy model predictive control based upon adaptive neural network disturbance observer for a constrained hypersonic vehicle," *IEEE Access*, vol. 6, pp. 5927–5938, 2018, doi: 10.1109/ACCESS.2017.2780118.
- [101] F. Zhao, J. Ou, and W. Du, "Pattern-based fuzzy predictive control for a chemical process with dead time," *Engineering Applications of Artificial Intelligence*, vol. 13, no. 1, pp. 37–45, 2000, doi: 10.1016/S0952-1976(99)00035-4.
- [102] J. Abonyi, L. Nagy, and F. Szeifert, "Fuzzy model-based predictive control by instantaneous linearization," *Fuzzy Sets and Systems*, vol. 120, no. 1, pp. 109–122, 2001, doi: 10.1016/S0165-0114(99)00118-9.
- [103] M. Lepetič, I. Škrjanc, H. G. Chiacchiarini, and D. Matko, "Predictive functional control based on fuzzy model: Comparison with linear predictive functional control and PID control," *Journal of Intelligent & Robotic Systems*, vol. 36, no. 4, pp. 467–480, 2003, doi: 10.1023/A:1024476802010.
- [104] N. Li, S.-Y. Li, and Y.-G. Xi, "Multi-model predictive control based on the Takagi–Sugeno fuzzy models: A case study," *Information Sciences*, vol. 165, nos. 3–4, pp. 247–263, 2004, doi: 10.1016/j.ins.2003.10.011.
- [105] I. Škrjanc and S. Blažič, "Predictive functional control based on fuzzy model: Design and stability study," *Journal of Intelligent & Robotic Systems*, vol. 43, no. 4, pp. 283–299, 2005, doi: 10.1007/s10846-005-5138-9.
- [106] L. Dalhoumi, M. Chtourou, and M. Djemel, "Impulse fuzzy model based predictive control for nonlinear systems," in *Proc. 12th Int. Multi-Conf. Systems, Signals & Devices (SSD)*, Chemnitz, Germany, 2015, pp. 1–6, doi: 10.1109/SSD.2015.7348084.
- [107] P. Muthukumar, P. Balasubramaniam, and K. Ratnavelu, "T–S fuzzy predictive control for fractional order dynamical systems and its applications," *Nonlinear Dynamics*, vol. 86, no. 2, pp. 751–763, 2016, doi: 10.1007/s11071-016-2919-6.

- [108] L. Teng, Y. Wang, W. Cai, and H. Li, "Fuzzy model predictive control of discrete-time systems with time-varying delay and disturbances," *IEEE Transactions on Fuzzy Systems*, vol. 26, no. 3, pp. 1192–1206, 2018, doi: 10.1109/TFUZZ.2017.2717798.
- [109] G. Karer, G. Mušić, I. Škrjanc, and B. Zupančič, "Hybrid fuzzy modelling for model predictive control," *Journal of Intelligent & Robotic Systems*, vol. 50, no. 3, pp. 297–319, 2007, doi: 10.1007/s10846-007-9166-5.
- [110] X. Liu, J. Liu, and P. Guan, "Neuro-fuzzy generalized predictive control of boiler steam temperature," *Journal of Control Theory and Applications*, vol. 5, no. 1, pp. 83–88, 2007, doi: 10.1007/s11768-005-5258-6.
- [111] D. Sáez, C. E. Cortés, and A. Núñez, "Hybrid adaptive predictive control for the multi-vehicle dynamic pick-up and delivery problem based on genetic algorithms and fuzzy clustering," *Computers & Operations Research*, vol. 35, no. 11, pp. 3412–3438, 2008, doi: 10.1016/j.cor.2007.01.025.
- [112] M. He, W. Cai, and S. Li, "Multiple fuzzy model-based temperature predictive control for HVAC systems," *Information Sciences*, vol. 169, nos. 1–2, pp. 155–174, 2005, doi: 10.1016/j.ins.2004.02.016.
- [113] C. L. Su, M. Lin, and P. Li, "Direct adaptive fuzzy predictive control and its application to CSTR process," *Applied Mechanics and Materials*, vols. 241–244, pp. 1191–1194, 2012, doi: 10.4028/www.scientific.net/AMM.241-244.1191
- [114] K. Salahshoor, E. Safari, and I. Ahangari, "A novel adaptive fuzzy predictive control for hybrid systems with mixed inputs," *Engineering Applications of Artificial Intelligence*, vol. 26, nos. 5–6, pp. 1512–1531, 2013, doi: 10.1016/j.engappai.2013.02.003.
- [115] J. Paulusová, L. Körösi, M. Dúbravská, and M. Paulus, "Neuro-fuzzy predictive control of thermo-optical plant," in *Proc. Int. Conf. Process Control (PC)*, Štrbské Pleso, Slovakia, 2015, pp. 238–243, doi: 10.1109/PC.2015.7169921.
- [116] X. Wu, J. Shen, Y. Li, and K. Y. Lee, "Data-driven modeling and predictive control for boiler-turbine unit using fuzzy clustering and subspace methods," *ISA Transactions*, vol. 53, no. 3, pp. 699–708, 2014, doi: 10.1016/j.isatra.2013.12.033.

- [117] H. Xie, J. Wang, and X. Tang, "Robust model predictive control of uncertain discrete-time T-S fuzzy systems," in Proc. Chinese Control and Decision Conf. (CCDC), Yinchuan, China, 2016, pp. 4066–4071, doi: 10.1109/CCDC.2016.7531695.
- [118] X. Chang and G. Yang, "Fuzzy robust constrained model predictive control for nonlinear systems," Asian Journal of Control, vol. 13, no. 6, pp. 947–955, 2011, doi: 10.1002/asjc.221.
- [119] Y. Zhao, H. Gao, and T. Chen, "Fuzzy constrained predictive control of nonlinear systems with packet dropouts," IET Control Theory & Applications, vol. 4, no. 9, pp. 1665–1677, 2010, doi: 10.1049/iet-cta.2009.0274.
- [120] B. Ding, "Comments on 'Constrained infinite-horizon model predictive control for fuzzy discrete-time systems'," IEEE Transactions on Fuzzy Systems, vol. 19, no. 3, pp. 598–600, 2011, doi: 10.1109/TFUZZ.2011.2126580.
- [121] S. Bououden, M. Chadli, S. Filali, and A. E. Hajjaji, "Fuzzy model-based multivariable predictive control of a variable speed wind turbine: LMI approach," Renewable Energy, vol. 37, no. 1, pp. 434–439, 2012, doi: 10.1016/j.renene.2011.06.025.
- [122] M. Khairy, A. L. Elshafei, and H. M. Emara, "LMI-based design of constrained fuzzy predictive control," Fuzzy Sets and Systems, vol. 161, no. 7, pp. 893–918, 2010, doi: 10.1016/j.fss.2009.10.020.
- [123] A. Flores, D. Sáez, J. Araya, M. Berenguel, and A. Cipriano, "Fuzzy predictive control of a solar power plant," IEEE Transactions on Fuzzy Systems, vol. 13, no. 1, pp. 58–68, 2005, doi: 10.1109/TFUZZ.2004.839658.
- [124] F. Villarroel, J. R. Espinoza, C. A. Rojas, J. Rodríguez, M. Rivera, and D. Sbarbaro, "Multiobjective switching state selector for finite-states model predictive control based on fuzzy decision making in a matrix converter," IEEE Transactions on Industrial Electronics, vol. 60, no. 2, pp. 589–599, 2013, doi: 10.1109/TIE.2012.2206343.
- [125] H. Eliasi, H. Davilu, and M. B. Menhaj, "Adaptive fuzzy model-based predictive control of nuclear steam generators," Nuclear Engineering and Design, vol. 237, no. 6, pp. 668–676, 2007, doi: 10.1016/j.nucengdes.2006.08.007.

- [126] D. Dovžan and I. Škrjanc, "Predictive functional control based on an adaptive fuzzy model of a hybrid semi-batch reactor," *Control Engineering Practice*, vol. 18, no. 8, pp. 979–989, 2010, doi: 10.1016/j.conengprac.2010.04.004.
- [127] S. Bououden, M. Chadli, and H. R. Karimi, "An ant colony optimization-based fuzzy predictive control approach for nonlinear processes," *Information Sciences*, vol. 299, pp. 143–158, 2015, doi: 10.1016/j.ins.2014.11.050.
- [128] I. Boulkaibet, K. Belarbi, S. Bououden, M. Chadli, and T. Marwala, "An adaptive fuzzy predictive control of nonlinear processes based on multi-kernel least squares support vector regression," *Applied Soft Computing*, vol. 73, pp. 572–590, 2018, doi: 10.1016/j.asoc.2018.08.044.
- [129] J. A. Méndez, A. Marrero, J. A. Rebozo, and A. León, "Adaptive fuzzy predictive controller for anesthesia delivery," *Control Engineering Practice*, vol. 46, pp. 1–9, 2016, doi: 10.1016/j.conengprac.2015.09.009.
- [130] M. Hamdy, S. Abd-Elhaleem, and M. A. Fkirin, "Adaptive fuzzy predictive control for a class of networked nonlinear systems with time-varying delay," *IEEE Transactions on Fuzzy Systems*, vol. 26, no. 4, pp. 2135–2144, 2018, doi: 10.1109/TFUZZ.2017.2764851.
- [131] M. Mahfouf, M. F. Abbod, and D. A. Linkens, "Online elicitation of Mamdani-type fuzzy rules via TSK-based generalized predictive control," *IEEE Transactions on Systems, Man, and Cybernetics, Part B (Cybernetics)*, vol. 33, no. 3, pp. 465–475, 2003, doi: 10.1109/TSMCB.2003.810901.
- [132] X. Hu, J. Wang, L. Zhang, and B. Wang, "Data-driven modelling and fuzzy multiple-model predictive control of oxygen content in coal-fired power plant," *Transactions of the Institute of Measurement and Control*, vol. 39, no. 11, pp. 1631–1642, 2017, doi: 10.1177/0142331216644498.
- [133] M. G. Forbes, R. S. Patwardhan, H. Hamadah, and R. B. Gopaluni, "Model predictive control in industry: Challenges and opportunities," *IFAC-PapersOnLine*, vol. 48, no. 8, pp. 531–538, 2015, doi: 10.1016/j.ifacol.2015.09.022.
- [134] M. L. Hadjili and K. Kara, "Modelling and control using Takagi–Sugeno fuzzy models," in *Proc. Saudi Int. Electronics, Communications and Photonics Conf. (SIEPCP)*, 2011, pp. 1–6, doi: 10.1109/SIEPCP.2011.5876946.

- [135] M. E. Hedroug, E. K. Bdirina, and K. Guesmi, "Fuzzy predictive controller for trajectory tracking of differential-drive mobile robot," in Proc. 2nd Int. Conf. Electronics, Energy and Measurement (IC2EM), 2023, pp. 1–6, doi: 10.1109/IC2EM59347.2023.10419477.
- [136] L. Yang, M. Guo, A. Mohammadzadeh, and A. Mosavi, "Taylor series-based fuzzy model predictive control for wheeled robots," *Mathematics*, vol. 10, no. 14, Art. no. 2498, 2022, doi: 10.3390/math10142498.
- [137] H. Omrane, M. S. Masmoudi, and M. Masmoudi, "Fuzzy logic-based control for autonomous mobile robot navigation," *Computational Intelligence and Neuroscience*, vol. 2016, Art. no. 9548482, pp. 1–10, 2016, doi: 10.1155/2016/9548482.
- [138] J.-J. E. Slotine and W. Li, *Applied Nonlinear Control*. Englewood Cliffs, NJ, USA: Prentice Hall, 1991.
- [139] H. K. Khalil, *Nonlinear Systems*, 3rd ed. Upper Saddle River, NJ, USA: Prentice Hall, 2002.
- [140] M. Chadli and P. Borne, *Multiple Models Approach in Automation: Takagi–Sugeno Fuzzy Systems*. London, U.K.: ISTE Ltd; Hoboken, NJ, USA: John Wiley & Sons, 2013.
- [141] H. Ohtake, K. Tanaka, and H. O. Wang, "Fuzzy modeling via sector nonlinearity concept," in Proc. IEEE Int. Conf. Fuzzy Systems, 2001, pp. 127–132.
- [142] K. Gasso, G. Mourot, and J. Ragot, "Structure identification in multiple model representation: Elimination and merging of local models," in Proc. 40th IEEE Conf. Decision and Control (CDC), Orlando, FL, USA, 2001, pp. 2992–2997.
- [143] A. Akhenak, "Conception d'observateurs non linéaires par approche multimodèle: Application au diagnostic," Ph.D. dissertation, Institut National Polytechnique de Lorraine, Nancy, France, 2004. [Online]. Available: <https://www.theses.fr/2004INPL114N>.
- [144] R. Siegwart, I. R. Nourbakhsh, and D. Scaramuzza, *Introduction to Autonomous Mobile Robots*, 2nd ed. Cambridge, MA, USA: MIT Press, 2011.
- [145] Y. Kanayama, Y. Kimura, F. Miyazaki, and T. Noguchi, "A stable tracking control method for an autonomous mobile robot," in Proc. IEEE Int. Conf. Robotics and Automation (ICRA), 1990, pp. 384–389.
- [146] G. Klančar, A. Zdešar, S. Blažič, and I. Škrjanc, *Wheeled Mobile Robotics: From Fundamentals Towards Autonomous Systems*. Oxford, U.K.: Butterworth-Heinemann, 2017.

- [147] K. Tanaka and H. O. Wang, *Fuzzy Control Systems Design and Analysis: A Linear Matrix Inequality Approach*. New York, NY, USA: John Wiley & Sons, 2001.
- [148] F. Kühne, J. M. G. da Silva Jr., and W. F. Lages, "Mobile robot trajectory tracking using model predictive control," in *Proc. VII SBAI / II IEEE Latin American Robotics Symp. (LARS)*, São Luís, Brazil, 2005.
- [149] M. E. Hedroug, E. K. Bdirina, K. Guesmi, B. Nail, I. E. Tibermacine, and A. Ma'arif, "Improved Trajectory Tracking for Nonholonomic Mobile Robots Via Dynamic Weight Adjustment in Type-2 Fuzzy Model Predictive Control," *International Journal of Robotics and Control Systems*, vol. 5, no. 3, pp. 1919–1948, 2025, doi: 10.31763/ijrcs.v5i3.1926.
- [150] M. E. Hedroug, E. K. Bdirina, and K. Guesmi, "Identification of a Class of Affine Nonlinear Systems by T-S Fuzzy Structure with Lyapunov based LMI Stability Analysis," in *Proceedings of the 2025 5th International Conference on Applied Automation and Industrial Diagnostics (ICAAID)*, 2025, pp. 1–6, doi: 10.1109/ICAAID68975.2025.11358194.

Liste des travaux

Publications

- [1] **Mohamed Elamine Hedroug**, El Khansa Bdirina, Kamel Guesmi, Bachir Nail, Imad Eddine Tibermacine and Alfian Ma'arif, Improved trajectory tracking for nonholonomic mobile robots via dynamic weight adjustment in type-2 fuzzy model predictive control. International Journal of Robotics and Control Systems, 2025, vol. 5, no. 3, pp. 1919–1948. URL: <https://pubs2.ascee.org/index.php/IJRCs/article/view/1926>

Conférences Internationales

- [1] **Mohamed Elamine Hedroug**, El Khansa Bdirina and Kamel Guesmi, Fuzzy predictive controller for trajectory tracking of differential-drive mobile robot. The 2nd International Conference on Electronics, Energy and Measurement (IC2EM 2023), 2023, pp. 1–6. URL: <https://ieeexplore.ieee.org/document/10419477>
- [2] **Mohamed Elamine Hedroug**, El Khansa Bdirina and Kamel Guesmi, Identification of a class of affine nonlinear systems by T-S fuzzy structure with Lyapunov-based LMI stability analysis. The 5th International Conference on Applied Automation and Industrial Diagnostics (ICAAID 2025), 2025. URL: <https://conference.univ-djelfa.dz/icaaid25/>