

RÉPUBLIQUE DU CAMEROUN

Paix-Travail-Patrie

UNIVERSITÉ DE YAOUNDE I

Faculté des Sciences

Centre de Recherche et de formation

Doctotale en Sciences, Technologies
et Geosciences

Unité de Recherche et de Formation

Doctorale en Physique et Applications

B.P 812 Yaoundé



REPUBLIC OF CAMEROON

Peace-Work-Fatherland

THE UNIVERSITY OF YAOUNDE I

Faculty of Science

Postgraduate School of Science,

Technology and
Geosciences

Research and Postgraduate Training

Unit for Physics and Applications

P.O. Box 812 Yaoundé

Laboratoire d'Énergie et des Systèmes Électriques et Électroniques

Laboratory of Energy, and Electric and Electronic Systems

Modélisation et Commande des Manipulateurs Mobiles Continuum

T H È S E

Présentée à

L'Université de Yaoundé I

En vue de l'obtention du grade de

Docteur/PhD de physique,

Option : Systèmes électriques et électroniques

Par

MAHAMAT LOUTFI Imrane

Matricule : 13P2981

Master de Physique

Sous la co-direction de :

FRÉDÉRIC BIYA MOTTO M.C, *Fac Sc.(UYI)*

ACHILLE MELINGUI M.C, *ENSPY (UYI)*



Année académique 2022-2023

Dédicace

Ce travail est particulièrement dédié à mes chers parents.

Remerciements

Je tiens tout d'abord à adresser mes plus vifs remerciements et exprimer ma profonde gratitude au Professeur(M.C) Frédéric BIYA MOTTO et au Professeur(M.C) Achille MELINGUI, mes directeurs de thèse qui, par leur parfait encadrement, ont contribué à ce que ce travail de recherche se déroule dans de meilleures conditions. Je vous remercie pour votre soutien scientifique, pour votre confiance, pour vos multiples conseils qui ont été pour moi une source d'enrichissement et d'encouragement, Ce travail n'aurait pu être mené à bout sans votre aide et votre attention permanente sur l'évolution de mes recherches tout au long de cette thèse.

Mes remerciements vont ensuite aux membres du jury, qui auront la lourde tâche d'évaluer mon travail de thèse.

Mes remerciements vont aussi aux membres des jurys de la pré-soutenance et de l'audition, qui grâce à leurs remarques, le document a été amélioré .

Je tiens à remercier le chef de département, le Professeur NDJAKA Jean Marie ainsi que son personnel pour m'avoir offert un cadre et une atmosphère de travail agréables.

Mes remerciements vont également aux Professeurs du laboratoire d'électronique, pour leur accueil et leur soutien quotidiens. Mes remerciements vont particulièrement au Professeur ESSIMBI ZOBO Bernard et au Professeur EYEBE FOU DA Jean sire qui ont été d'un positif sans faille ; au Professeur BODO Bertrand et Professeur MBINACK Clément pour leurs conseils et leur aide en programmation.

Je tiens également à remercier Dr. MVOGO AHANDA Joseph Jean-Baptiste pour son soutien inconditionnel et pour son esprit critique qui m'a aidé à approfondir et perfectionner ce projet. Je ne saurais oublier mes camarades de laboratoire BOUYOM, GINO, MEDZO, NTOUBA, SEGNI et MBIYA pour la sincère collaboration, la solidarité, l'esprit de groupe.

Une profonde reconnaissance et des remerciements particuliers à mes parents, mes grands frères et à mes soeurs pour leur soutien permanent le long de mes études.

Aucun mot ne sera jamais assez fort pour exprimer tout ce que je dois à ma femme DJAMILA Idirissou qui m'a accompagné tout au long de ce travail par son soutien inconditionnel de tous côtés.

Mes remerciements vont également à mon très cher ami MAHAMAT HASSAN Babikir en particulier, et à tous ceux qui m'ont soutenu d'une manière ou d'une autre, ont contribué à l'élaboration de ce travail.

Les essais expérimentaux ont été réalisés au Centre de Recherche en Informatique, Signal et Automatique de Lille (CRISTal) de l'Université des sciences et technologies de Lille. C'est pourquoi, je remercie vivement l'ensemble du personnel et principalement le Professeur Rochdi MERZOUKI pour avoir autorisé la réalisation des dits essais.

Résumé

Cette thèse s'intéresse à la modélisation et la commande des manipulateurs mobiles continus pour l'assistance aux personnes à mobilité réduite. Nous utilisons la plate-forme robotique Robotino XT comme cas d'étude. Cette dernière est composée d'un manipulateur continu monté sur une plate-forme mobile omnidirectionnelle. Dans cette thèse, nous proposons un contrôleur champ de potentiel artificiel (APF)- neuro-flou à intervalle de type-2 (IT2FNN). Cette hybridation vise à exploiter les avantages offerts par chacune de ces approches pour concevoir un contrôleur robuste, intelligent et efficace en temps de calcul. Nous faisons également une étude comparative de quatre modèles d'apprentissage, à savoir le perceptron multicouche (MLP), les fonctions à base radiale (RBF), la régression par vecteur support (SVR) et le système d'inférence neuro-flou Co-adaptatif (CANFIS) pour la modélisation géométrique de la partie manipulatrice. Enfin, les deux contrôleurs précédents sont coordonnés par un système neuronal pour la commande de l'ensemble du Robotino XT. Les performances des différents contrôleurs ont été évaluées par une série de simulations et expériences en temps réel.

Mots clés :

Manipulateurs mobiles continus, Robots mobiles, Manipulateurs continus, Navigation autonome, Champ de potentiel artificiel, Réseaux de neurones, logique floue et Systèmes neuro-flous.

Abstract

This thesis focuses on the modelling and control of mobile continuum manipulators for the assistance of people with reduced mobility, using the Robotino XT robotic platform as a case study. The latter is composed of a continuum manipulator mounted on an omnidirectional mobile platform. In this thesis, we propose an artificial potential field (APF)-interval neurofuzzy controller type-2 (IT2FNN). This hybridisation aims to exploit the advantages offered by each of these approaches to design a robust, intelligent, and computationally efficient controller. We also make a comparative study of four learning models, namely the multilayer perceptron (MLP), radial basis functions (RBF), support vector regression (SVR), and co-adaptive neuro-fuzzy inference system (CANFIS) for the geometric modelling of the manipulator part. Finally, the two previous controllers are coordinated by a neural system for the control of the whole Robotino XT. The performance of the different controllers was evaluated through a series of real-time simulations and experiments.

Keywords : Continuum mobile manipulators, Mobile robots, Continuum manipulators, Autonomous navigation, Artificial potential field, Neural networks, Fuzzy logic and Neuro-fuzzy systems.

Table des matières

Dédicace	i
Remerciements	iii
Résumé	v
Abstract	vii
Acronymes et abréviations	xv
Introduction Générale	1
1 État de l'art	7
1.1 Introduction	7
1.2 Navigation autonome des robots mobiles	8
1.3 Modélisation géométrique directe et inverse des manipulateurs continus 12	
1.3.1 Modélisation géométrique directe	12
1.3.2 Modélisation géométrique inverse	16
1.4 Modélisation et Commande des mobiles manipulateurs continus	16
1.5 Conclusion	18
2 Méthodologies et Matériels utilisés	19
2.1 Introduction	19
2.2 Navigation autonome des robots mobiles	19
2.2.1 Description de la plate-forme expérimentale	19

2.2.2	Logique floue à intervalle de type-2	20
2.2.3	Contrôleur neuroflou à intervalle de type-2	24
2.2.4	Architecture IT2FNN proposée	28
2.3	Modélisation géométrique des manipulateurs continuus	36
2.3.1	Approches basées sur l'apprentissage	38
2.3.2	Modélisation géométrique du CBHA	48
2.4	Commande du mobile manipulateur continuum : Cas du RobotinoXT . . .	53
2.4.1	Modélisation du mobile manipulateur continuum	53
2.4.2	Système de coordination neuronal	55
2.4.3	Commande du manipulateur CBHA	56
2.5	Conclusion	57
3	Résultats des simulations, expériences et discussions	59
3.1	Introduction	59
3.2	Navigation autonome des robots mobiles : cas du Robotino	59
3.3	Modélisation géométrique du manipulateur continuum : Cas du CBHA . .	69
3.3.1	Processus de validation	69
3.3.2	Resultats des simulations et discussions	70
3.4	Navigation du manipulateur mobile continuum : Cas du Robotino XT . . .	78
3.5	Conclusion	86
	Conclusion Générale	87
	Bibliographie	89

Liste des figures

Figure 1	Bras manipulateurs industriels	1
Figure 2	Robot manipulateur à joints flexibles	2
Figure 3	Manipulateurs continus	3
Figure 4	Mobiles manipulateurs continus	3
Figure 5	Robots d'assistance	4
Figure 1.1	Robots d'assistance	8
Figure 1.2	Modélisation géométrique d'un manipulateur continu	13
Figure 1.3	Repères attachés à un mobile manipulateur continu	17
Figure 2.1	La plateforme mobile Robotino avec capteurs infrarouges.	20
Figure 2.2	Direction de la force potentielle artificielle lorsque le robot fait face à un obstacle frontal en cas d'orientation négative (\widetilde{NE}) ou positive (\widetilde{PO}) du point cible par rapport à la position du robot.	22
Figure 2.3	Quelques situations de minima locaux.	22
Figure 2.4	Stratégies classiques de navigation.	23
Figure 2.5	Structure de contrôle IT2FNN	26
Figure 2.6	Les distances entre le robot et les obstacles les plus proches sur ses côtés gauche, avant et droit, et étiquetées comme "Near" (\widetilde{N}) et "Far" (\widetilde{F}).	26

Figure 2.7	L'orientation de la position de la cible par rapport à la position actuelle du robot, modélisée par trois fonctions d'appartenance gaussiennes de type-2 à intervalle et étiquetée comme suit " <i>Négative</i> " (\widetilde{NE}), " <i>Zero</i> " (\widetilde{Z}) " <i>Positive</i> " (\widetilde{PO}).	27
Figure 2.8	La force artificielle longitudinale (F_x) et la force artificielle latérale (F_y) modélisées par trois fonctions d'appartenance gaussiennes de type-2 à intervalle et étiquetées comme " <i>Négative</i> " (\widetilde{NE}), " <i>Zéro</i> " (\widetilde{Z}) " <i>Positive</i> " (\widetilde{PO}).	27
Figure 2.9	Structure du contrôleur flou de type-2 à intervalle [Wang 2004] . . .	28
Figure 2.10	Ensemble flou à intervalle de type-2 avec une moyenne incertaine et un écart-type fixe [Wang 2004].	29
Figure 2.11	Topologie de la MLP	39
Figure 2.12	Topologie de la RBF	41
Figure 2.13	Architecture du CANFIS	42
Figure 2.14	Problème de convergence	46
Figure 2.15	Système d'apprentissage composé	47
Figure 2.16	Manipulateur continuum CBHA	49
Figure 2.17	Schémas d'apprentissage supervisé distale	49
Figure 2.18	Capture de mouvement optitrack	51
Figure 2.19	Espace de travail du CBHA	52
Figure 2.20	Système mobile manipule	53
Figure 2.21	Repère du monde sur la plate-forme mobile	54
Figure 2.22	Contrôle du mobile manipulateur RobotinoXT	55
Figure 2.23	Modèle du système de coordination à l'aide de réseau de neurones	56
Figure 3.1	Premier scénario : Les stratégies de suivi de couloir, d'évitement d'obstacle et d'atteinte de la cible en présence d'obstacles à géométrie régulière.	61

Figure 3.2	Premier scénario : Vitesses longitudinales fournies par les contrôleurs IT2FL et IT2FNN.	62
Figure 3.3	Premier scénario : Vitesses latérales fournies par les contrôleurs IT2FL et IT2FNN.	62
Figure 3.4	Deuxième scénario : les stratégies de suivi de couloir, d'évitement d'obstacles et d'atteinte de la cible dans des situations de présence de minima locaux.	64
Figure 3.5	Deuxième scénario : Vitesses longitudinales fournies par les contrôleurs IT2FL et IT2FNN.	64
Figure 3.6	Deuxième scénario : Vitesses latérales fournies par les contrôleurs IT2FL et IT2FNN.	65
Figure 3.7	Troisième scénario : les stratégies de suivi de couloir, d'évitement d'obstacles et d'atteinte de la cible en présence d'obstacles de forme irrégulière.	65
Figure 3.8	Troisième scénario : Vitesses longitudinales fournies par les contrôleurs IT2FL et IT2FNN.	66
Figure 3.9	Troisième scénario : Vitesses latérales fournies par les contrôleurs IT2FL et IT2FNN.	66
Figure 3.10	Le quatrième scénario : la navigation dans des environnements dynamiques.	67
Figure 3.11	Le quatrième scénario : Vitesses longitudinales fournies par les contrôleurs IT2FL et IT2FNN.	68
Figure 3.12	Le quatrième scénario : Vitesses latérales fournies par les contrôleurs IT2FL et IT2FNN	68
Figure 3.13	Erreur euclidien suivant les axes X et Y	75
Figure 3.14	Erreur euclidien suivant la position z et l'orientation θ	76
Figure 3.15	Erreur euclidien suivant les orientations ϕ et φ	77
Figure 3.16	Environnement experimental	79

Figure 3.17	Mise en oeuvre en temps réel : Prédiction des tensions du fil-potentiomètre par le réseau MLP	80
Figure 3.18	Mise en oeuvre en temps réel : Erreurs euclidiennes, MLP	81
Figure 3.19	Environnement expérimental : Premier scénario	82
Figure 3.20	Environnement expérimental : Second scénario	82
Figure 3.21	Premier scénario : Trajectoire décrite par la position de la pointe du robot.	83
Figure 3.22	Premier scénario : Vitesses générées par le contrôleur de la plateforme mobile (T2FNN).	84
Figure 3.23	Second scénario : Trajectoire décrite par la position de l'extrémité du Robot	85
Figure 3.24	Second scénario : Vitesses générées par le contrôleur de la plateforme mobile (T2FNN).	85

Liste des tables

Table 2.1	Base de règles floues dans T2PFL	24
Table 3.1	Comparaisons des résultats expérimentaux	63
Table 3.2	Resultats des modèles MLP, RBF, SVR et CANFIS.	71
Table 3.3	Valeur absolue des erreurs euclidiennes suivant la position	71
Table 3.4	Valeur absolue des erreurs euclidiennes suivant l'orientation	71
Table 3.5	Coefficient de corrélation linéaire selon la Position R	72
Table 3.6	Coefficient de corrélation linéaire selon l'Orientation R	72
Table 3.7	Résultats obtenus par chaque modèle de réseau neuronal sur les données de test	74
Table 3.8	Valeurs des signaux de commande générés par le contrôleur du manipulateur continu dans le premier scénario.	83
Table 3.9	Valeurs des signaux de commande générés par le contrôleur du manipulateur continuum dans le second scénario.	86

Acronymes et abréviations

APF	Artificial Potential Field (champ de potentiel artificiel)
CANFIS	Co-Active Neuro Fuzzy Interface System
CBHA	Compact Bionique Handling Assistant
CC	Courbure Constante
D-H	Denavit-Hartenberg
DLS	Distal Learning Supervisor (Apprentissage supervisé distal)
DSC	Dynamic Surface Control
EQM	Erreur Quadratique Moyenne
FL	Fuzzy Logic (logique floue)
FNN	Fuzzy Neural Network (réseau neuro-flou)
IT2FL	Interval Type 2 Fuzzy Logic (logique floue à intervalle de type-2)
IT2FNN	Interval Type-2 Fuzzy Neural Network(neuro-flou à intervalle de type-2)
IT-2-GMF	Fonctions d'appartenance gaussiennes de type-2 à intervalle
KM	KarniK-Mendel
MF	Membership Function (fonction d'appartenance)
MGD	Modèle Géométrique Direct
MGI	Modèle Géométrique Inverse
MLP	Multi Layer Perceptron (perceptron multi couche)
MSE	Mean square error(erreur quadratique moyenne)
NN	Neural Network (réseau de neurones)
PID	Proportional Integral Derivative
PQ	Programmation Quadratique
RBF	Radial Basis Function (fonction à base radiale)
RIT-2-MF	Fonction d'appartenance de type-2 à intervalle rectangulaire
SVM	Support Vector Machine (machine à vecteur de support)
SVR	Support Vector Regression (regression par les vecteurs supports)

Introduction Générale

Contexte et motivation

De nos jours, l'importance de la robotique ne saurait être déniée. De l'industrie [Bahrin 2016] à l'exploration planétaire [Yim 2003], en passant par la médecine [Hamet 2017], le transport [Michael 2011] et l'agriculture [Marinoudi 2019], les applications sont innombrables. A l'origine, les concepteurs de robots visaient à reproduire le bras humain pour remplacer l'homme dans certaines tâches laborieuses ou dangereuses. Ils ont donc pensé à un système poly-articulé avec des articulations contrôlables, appelé *bras manipulateur*, Figure 1. Afin de reproduire plus fidèlement le bras humain, les concepteurs ont pensé à remplacer les articulations rigides par des articulations flexibles ; et le système a pris le nom de *robot manipulateur à joints flexibles*, Figure 2. Ces robots sont très efficaces pour les tâches traditionnelles où les environnements de travail sont très structurés (industrie). Cependant, si ces robots devaient être utilisés dans des environnements moins structurés et restreints, leurs corps rigides limiteraient considérablement leur champ d'application.



FIGURE 1 – Bras manipulateurs industriels

En s'inspirant de la nature, les concepteurs ont pensé à reproduire les mouvements



FIGURE 2 – Robot manipulateur à joints flexibles

de la trompe de l'éléphant, des tentacules des calamars, des langues des lézards ; ce qui a donné naissance aux *robots continuums*, Figure 3. Ces robots ont une flexibilité inhérente et peuvent se plier le long de leur structure. En imaginant toutes les manoeuvres qu'un éléphant peut effectuer avec sa trompe, on comprend vite l'intérêt que les chercheurs portent à ce style de robot. Ils trouvent leurs applications dans la médecine, le sauvetage et les interventions chirurgicales. Cependant, si les recherches abondent sur les manipulateurs continus montés sur des structures statiques, peu se sont intéressées aux *manipulateurs mobiles continuums* où non seulement l'espace de travail est étendu, mais aussi les degrés de liberté (ddl) supplémentaires peuvent être utilisés pour la réalisation des tâches plus complexes, Figure 4.

Un manipulateur continuum mobile est généralement considéré comme un manipulateur continuum monté sur une plate-forme mobile. Contrairement au manipulateur continuum fixe, ce dernier peut effectuer des manipulations dans un espace de travail beaucoup plus grand en raison de sa mobilité à grande échelle et de sa dextérité. Les applications des mobiles manipulateurs continus sont nombreuses. On peut citer : l'exploration spatiale [Michael 2011], les opérations de sauvetage [Marinoudi 2019], l'assistance aux personnes âgées ou handicapées, etc. Cette thèse vise à proposer une aide (robot) pour l'assistance

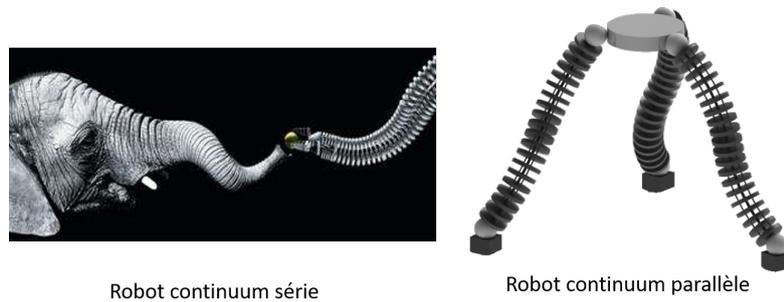


FIGURE 3 – Manipulateurs continus

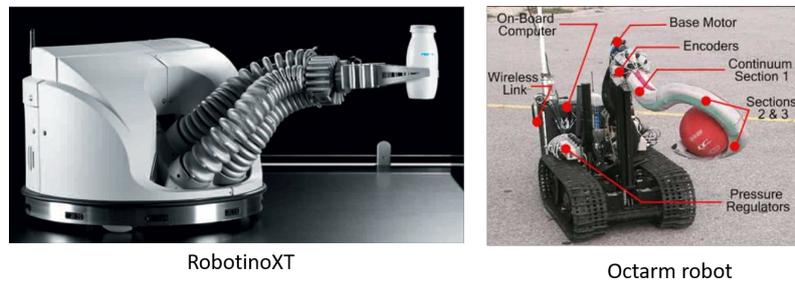


FIGURE 4 – Mobiles manipulateurs continus

aux personnes handicapées moteur, plus précisément aux développements des algorithmes pour la navigation autonome des mobiles manipulateurs continus en environnements non structurés.

Au cours de ces dernières décennies, la robotique de service a suscité un intérêt aussi bien social que financier. Elle a pour objectif l'assistance aux personnes à mobilité réduite (blessés médullaires, personnes souffrant de maladies neuromusculaires, d'agénésie musculaire, etc) dans les tâches nécessitant la mobilité telles que les tâches ménagères, la saisie et la manutention des objets, la restauration, etc, Figure 5. Cependant, ces applications ne sont possibles que si les robots sont contrôlés. Le contrôle d'un robot passe par une modélisation complète de sa structure mécatronique et une élaboration des lois de commande appropriées. Dans cette thèse, nous proposons des lois de commande aussi bien pour la plateforme mobile que le manipulateur continuum.

Pour mener à bien leurs missions, les robots mobiles sont parfois équipés de sys-



FIGURE 5 – Robots d'assistance

tèmes de navigation autonomes très performants. Par conséquent, la navigation autonome des robots mobiles est devenue un sujet de recherche majeur [Pandey 2017, Faisal 2013, Hagrais 2004b, Beom 1995]. En fonction des informations (complètes ou incomplètes) disponibles dans l'environnement de navigation, les systèmes de navigation peuvent être regroupés en deux grandes classes, à savoir la navigation globale et la navigation locale ou navigation réactive. En navigation globale, le robot mobile dispose d'une vue d'ensemble de l'environnement de navigation, avec une connaissance a priori des positions des obstacles et de la cible à atteindre, tandis qu'en navigation locale, le robot mobile n'a qu'une connaissance locale de son environnement. Pourtant, étant donné la nature dynamique des environnements de navigation des robots mobiles, les approches de navigation globale peuvent facilement devenir chronophages. Par conséquent, de nombreux travaux de recherche sur la navigation des robots mobiles se sont concentrés sur la méthode de navigation locale, où le temps de calcul est rationnel [Patle 2019, Subbash 2019, Karray 2016, Sanchez 2015, Kim 2014, Mbede 2012]. Certaines méthodes de navigation autonome, lorsqu'elles sont mises en oeuvre seules, peuvent conduire à des performances médiocres, alors que leurs combinaisons, lorsqu'elles sont bien pensées, peuvent donner des performances exceptionnelles. Dans cette thèse, nous proposons une combinaison de trois méthodes largement utilisées dans la navigation autonome

des robots mobiles, et dont l'implémentation individuelle ne donne pas toujours les performances escomptées. Il s'agit du Champ de Potentiel Artificiel (APF) qui est rapide et facile à mettre en oeuvre mais qui est confronté aux problèmes de minima locaux et de robustesse. La logique floue (FL) est robuste mais exigeante en temps de calcul. Enfin, les réseaux de neurones (NN) ont une capacité de généralisation exceptionnelle, mais sont confrontés à des problèmes de collecte de données pour la base d'apprentissage et de robustesse. Cette thèse vise à exploiter les avantages offerts par chacune de ces approches pour concevoir un contrôleur qui est à la fois robuste, intelligent et efficace en temps de calcul.

Le modèle géométrique direct (MGD) est un module essentiel pour la conception des lois de commande des manipulateurs continus. Contrairement aux manipulateurs rigides pour lesquels il peut être facilement établi, il reste un véritable défi pour leurs homologues continus. Les approches basées sur le modèle et sur l'apprentissage sont couramment utilisées dans la modélisation géométrique directe des manipulateurs continus. Les approches basées sur le modèle conduisent généralement à des modèles géométriques imprécis en raison de plusieurs hypothèses de modélisation, tandis que les approches basées sur l'apprentissage donnent généralement des performances acceptables. Cependant, le choix d'un modèle d'apprentissage approprié reste une tâche difficile. Dans cette thèse, nous proposons une étude comparative expérimentale et structurelle des modèles d'apprentissage couramment utilisés, à savoir le perceptron multicouche (MLP), les fonctions à base radiale (RBF), la régression par vecteur de support (SVR) et le système d'inférence neuro-flou co-adaptatif (CANFIS). Le modèle géométrique inverse est dérivé du MGD via l'apprentissage distal [Jordan 1992].

Enfin, nous proposons un système de contrôle intelligent pour la navigation autonome du mobile manipulateur continu évoluant dans des environnements intérieurs non structurés. Ce système intègre des lois de commande développées respectivement pour la plate-forme mobile et le manipulateur continu.

Contributions

Les différents résultats obtenus ont fait l'objet de deux publications scientifiques (dont un article indexé et une conférence) :

1. **I. Mahamat Loutfi**, A. Melingui, J. J. B. Mvogo Ahanda, F. Biya Motto and R. Merzouki, « Artificial potential field neuro-fuzzy controller for autonomous navigation of mobile robots ». *Proceedings of the Institution of Mechanical Engineers, Part I : Journal of Systems and Control Engineering*, 235(7), 1179-1192, 2021.
2. **I. Mahamat Loutfi**, A. B. Boutchouang, A. Melingui, O. Lakhal, F.B. Motto and R. Merzouki, «Learning-Based Approaches for Forward Kinematic Modeling of Continuum Manipulators». *IFAC PapersOnLine*, 53(2), 9899-9904, 2020.

Organisation de la thèse

Le reste de cette thèse sera structuré de la manière suivante :

- Le premier chapitre portera sur l'état de l'art, principalement sur la modélisation et la commande des manipulateurs mobiles continus. Ce chapitre permettra de positionner notre travail par rapport à la littérature.
- Le deuxième chapitre se concentrera au développement des algorithmes des lois de commande proposées, à savoir le contrôleur APF-Neuro flou pour la plate-forme mobile, le contrôleur neuronal pour le manipulateur continuum et le système de coordination intelligent pour la commande du manipulateur continuum mobile.
- Le troisième chapitre présentera les résultats des différentes simulations et expériences réalisées, ainsi qu'une discussion sur ces derniers.
- Enfin, le document se terminera par une conclusion, résumant les principales contributions de cette thèse, et discutant des futurs travaux dérivés.

État de l'art

1.1 Introduction

Inspirés par la nature (bras humain, animaux, reptiles,...), les robots ont longtemps été utilisés pour remplacer ou assister l'homme, afin d'augmenter la productivité, la précision, l'endurance, ou pour réaliser des tâches dangereuses ou ennuyeuses. Les robots à articulation rigide sont bien adaptés à ces tâches où le désir de mouvements répétitifs de haute vitesse et de haute précision est essentiel. Cependant, les applications incluant la navigation et le fonctionnement dans des environnements complexes et encombrés tels que les opérations de sauvetage, les interventions chirurgicales [Webster 2006], [Webster III 2009] nécessitent une interaction intelligente avec l'environnement de navigation. Il s'avère que les manipulateurs à articulations rigides ne siéent pas mieux à ces applications. Ils présentent des risques pour la sécurité, principalement en raison de leur taille, de leur poids et de la structure inflexible de leur corps. Les robots de type continuum, qui peuvent se plier en tout point de leur structure, deviennent une alternative pour des telles applications.

Ce chapitre se concentre essentiellement sur le positionnement de nos travaux dans la littérature. Les notions élémentaires sur les définitions et les concepts de base pourront être trouvées dans plusieurs ouvrages dans la littérature [Mendel 2017, Bishop 2006, Bishop 1995, Khatib 1986]. Nous présentons les différentes contributions existantes sur la modélisation et la commande des manipulateurs mobiles continus. Nous commençons par la présentation de l'état sur la navigation autonome des robots mobiles, suivi par la modélisation et la commande des manipulateurs continus. La modélisation et la

commande des manipulateurs mobiles continus sont présentées par la suite, et une conclusion clôture le chapitre.

1.2 Navigation autonome des robots mobiles

La navigation autonome des robots mobiles est un sujet de recherche d'actualité, Figure 1.1. En fonction des informations (complètes ou incomplètes) disponibles dans l'environnement de navigation, les systèmes de navigation peuvent être regroupés en deux grandes classes, à savoir la navigation globale et la navigation locale ou navigation réactive. En navigation globale, le robot mobile dispose d'une vue d'ensemble de l'environnement de navigation, avec une connaissance à priori des positions des obstacles et de la cible à atteindre, tandis qu'en navigation locale, le robot mobile n'a qu'une connaissance locale de son environnement.



FIGURE 1.1 – Robots d'assistance

Parmi les approches réactives de navigation autonome, la logique floue apparaît comme une technique efficace pour faire face aux incertitudes et aux imprécisions présentes dans les environnements de navigation des robots mobiles [Sanchez 2015, Hagrais 2004b, Pandey 2017, Melingui 2014d]. En outre, étant donné le grand nombre de situations aux-

quelles les robots mobiles sont confrontés pendant la navigation, il sera difficile et long pour les experts humains d'examiner toutes ces situations pour trouver la bonne décision de commande. Ainsi, les algorithmes du "soft computing" tels que la logique floue deviennent des candidats appropriés dans de telles situations. Par exemple, la méthode de logique floue de type-1 a été largement utilisée pour la navigation et le contrôle des robots mobiles [Beom 1995, Faisal 2013, Surmann 1995, Parhi 2005, Masmoudi 2016, Bakdi 2017]. Sur la base des connaissances des experts, des systèmes flous de type-1 ont été développés en utilisant des règles floues et des fonctions d'appartenance à valeurs non floues. Afin d'améliorer les performances de ces systèmes, des techniques d'apprentissage, telles que les algorithmes génétiques, ont également été mises en oeuvre pour régler les paramètres des fonctions d'appartenance [Pratihari 1999, Hagrass 2004a]. Cependant, il a été démontré que les systèmes flous de type-1 ne peuvent pas entièrement gérer ou faire face aux incertitudes linguistiques et numériques associées aux environnements de navigation non structurés, changeants et dynamiques des robots mobiles, car ils utilisent les valeurs crêtes des fonctions d'appartenance. En effet, une fois que les fonctions d'appartenance des systèmes flous de type-1 sont choisies ou réglées, la performance du système global devient fixe, et les effets des incertitudes ne peuvent donc pas être efficacement réduits ou éliminés. D'autre part, les systèmes de logique floue d'ordre supérieur, tels que les systèmes de logique floue à intervalle de type-2 utilisant des ensembles flous à intervalle de type-2, se sont révélés capables de modéliser et de traiter de telles incertitudes et impressions [Mendel 2019, Sanchez 2015, Hagrass 2004b]. En effet, cette capacité est justifiée par le fait qu'une fonction d'appartenance à intervalle de type-2 peut être vue comme possédant un nombre indénombrable d'ensembles de type-1. La troisième dimension des ensembles de logique floue de type-2 et sa plage d'incertitude leur confèrent plus de degrés de liberté suffisants pour une meilleure modélisation de l'incertitude que les ensembles flous de type-1. Dans la littérature, il existe un grand nombre de travaux qui comparent les performances des systèmes flous de type-1 et de type-2 comme dans [Sanchez 2015, Hagrass 2004b, Melin 2013, Datouo 2017]. En outre, de nombreux algo-

rithmes d'optimisation tels que le paradigme d'optimisation chimique [Astudillo 2013], l'optimisation par essaim de particules [Maldonado 2013], l'algorithme d'apprentissage hybride [MéNdez 2013], ont également été mis en oeuvre pour trouver les paramètres optimaux des fonctions d'appartenance floues.

En ce qui concerne la navigation des robots mobiles, de nombreux travaux de recherche ont été menés sur les systèmes flous de type-2. [Junratanasiri 2011] ont utilisé la logique floue de type-2 à intervalle pour l'évitement d'obstacles et la stabilisation de la position de robots mobiles naviguant dans des environnements dynamiques.[Huang 2017] ont proposé une approche intégrée de la logique floue avec la logique floue à intervalle de type-2 pour la modélisation et le contrôle d'un pendule inversé mobile à deux roues. L'objectif de la commande était d'atteindre la position et la direction souhaitées tout en maintenant l'équilibre du système. Un contrôleur de logique floue de type-2 pour le suivi des murs par un robot mobile à roues dans des environnements d'apprentissage par renforcement a également été proposé dans [Juang 2009]. [Karray 2016] se sont attachés à résoudre le problème du suivi de la trajectoire de manipulateurs mobiles non holonome en présence d'obstacles. Un contrôleur adaptatif basé sur l'algorithme de logique floue de type-2 optimisé par un algorithme de renforcement a été conçu pour minimiser l'effet des perturbations et des incertitudes du modèle pendant le suivi de la trajectoire. Deux autres contrôleurs flous ont été utilisés respectivement pour maintenir le robot loin des obstacles et atteindre un point cible virtuel. [Hagras 2004b] a utilisé un contrôleur flou à intervalle de type-2 pour implémenter le comportement de navigation de base et la coordination entre ces comportements pour produire un contrôleur hiérarchisé de type-2 pour les robots mobiles à deux roues naviguant dans des environnements intérieurs et extérieurs changeants et non structurés. Cependant, la plupart des méthodes susmentionnées sont basées sur une architecture hiérarchique et l'ensemble du système peut devenir compliqué et prendre beaucoup de temps pour accomplir les différents comportements de navigation. Par conséquent, il devient important d'avoir des structures uniques telles que les réseaux neuronaux flous qui peuvent combiner en même temps différents compor-

tements de navigation. C'est ainsi que [Kim 2014] ont proposé la méthode d'évitement des obstacles dans la stabilisation de la position des robots mobiles à roues à l'aide d'un réseau neuronal flou à intervalle de type-2 (IT2FNN). Un système de navigation utilisant un réseau de neurones flous à intervalle de type-2 (IT2FNN) avec l'apprentissage par renforcement a été proposé dans [Yi 2019] pour la navigation de robots mobiles dans des environnements complexes. Les capacités d'approximation de l'IT2FNN ont été spécifiquement utilisées pour résoudre la relation de mappage entre les espaces d'état et d'action dans l'algorithme de renforcement. En général, l'opération traditionnelle de réduction de type-2 utilisée dans les systèmes FNN de type-2 nécessite une procédure itérative [Mendel 2017, Liu 2011], telle que la procédure Karnik-Mendel (K-M) pour déterminer les sorties des systèmes neuro-flous. Cependant, lorsque le nombre de règles floues devient important et la base d'apprentissage énorme, l'apprentissage peut prendre des dizaines ou des centaines d'heures. À cet égard, l'opération traditionnelle de réduction de type a été remplacée par des limites pondérées dans [Juang 2012] pour réduire le temps d'apprentissage des systèmes IT2FNN. Dans le même ordre d'idées, un réseau neuronal flou à intervalle de type-2 simplifié, dans lequel la procédure itérative K-M chronophage est remplacée par deux q-facteurs de conception q_l et q_r , a été proposé dans [Lin 2013]. Ceci est d'une importance capitale dans le contexte de la navigation autonome, où le temps de réponse des systèmes de navigation est critique. En outre, le temps de réponse peut être réduit de manière significative si des stratégies visant à réduire la taille de la base de règles floues sont considérées [Jin 2000, Yam 1999, Melingui 2014d].

Précédemment, dans le cadre de la navigation autonome de robots mobiles, il a été proposé un système IT2FL avec les forces d'attraction et de répulsion du champ de potentiel artificiel (APF) comme variables linguistiques supplémentaires [Melingui 2014d, Melingui 2013a]. Ce contrôleur était capable d'effectuer des comportements de navigation de base avec une base de règles floues réduites. L'environnement de navigation a été divisé en zones de navigation, comprenant des zones avec des minima locaux et sans minima locaux. Le contrôleur APF évoluait virtuellement dans les zones sans minima locaux, tandis

que le contrôleur IT2FL fonctionnait dans les zones avec minima locaux. En effet, lors de la navigation dans des zones sans minima locaux, les sorties du contrôleur APF sont directement transférées à la sortie du contrôleur IT2FL en utilisant la fonction d'appartenance (MF) "none". En effet, pour une règle floue donnée, la MF "none" permet de ne pas considérer certaines variables linguistiques dans le calcul de la sortie. Dans cette thèse, les performances du précédent contrôleur IT2FL sont améliorées en remplaçant l'opération de réduction de type de K-M par deux q-facteurs de conception q_l et q_r qui sont appris via un algorithme de rétropropagation. Le contrôleur proposé est plus efficace que les contrôleurs IT2FL, car non seulement le processus de réduction de type est amélioré, la base de règles floues est réduite, mais aussi, le contrôleur a une capacité de généralisation qui est fournie par l'architecture neuro-floue. Le contrôleur proposé est plus efficace que les contrôleurs IT2FNN classiques car il fonctionne avec une base de règles floues réduite, une réduction qui résulte de l'utilisation comme entrées auxiliaires des forces longitudinales et latérales générées par l'approche APF [Melinguì 2014d]. Le résultat est un contrôleur robuste et efficace en temps de calcul, adapté aux applications en temps réel telles que la navigation autonome de robots mobiles. En effet, non seulement le contrôleur est robuste (logique floue de type-2), mais la taille de la base de règles floues est réduite et l'opération de réduction de type est efficace sur le plan de calcul.

1.3 Modélisation géométrique directe et inverse des manipulateurs continus

1.3.1 Modélisation géométrique directe

L'apprentissage automatique est un outil de plus en plus utilisé dans l'étude des nouvelles générations des robots qui combinent flexibilité mécanique, élasticité des matériaux et légèreté. De tels robots imitent le comportement d'êtres vivants comme les bras de pieuvres [Laschi 2009], les muscles [Pritts 2004], les tentacules [Walker 2013],

les troncs d'éléphants [Cieślak 1999],[Hannan 2003], et des membres de céphalopodes [McMahan 2004]. Ces propriétés les rendent particulièrement adaptés à un grand nombre d'applications, notamment la chirurgie [Bailey 2005], les opérations sous-marines [Lane 1997], et l'exploration [Walker 2012], [Liu 2016].

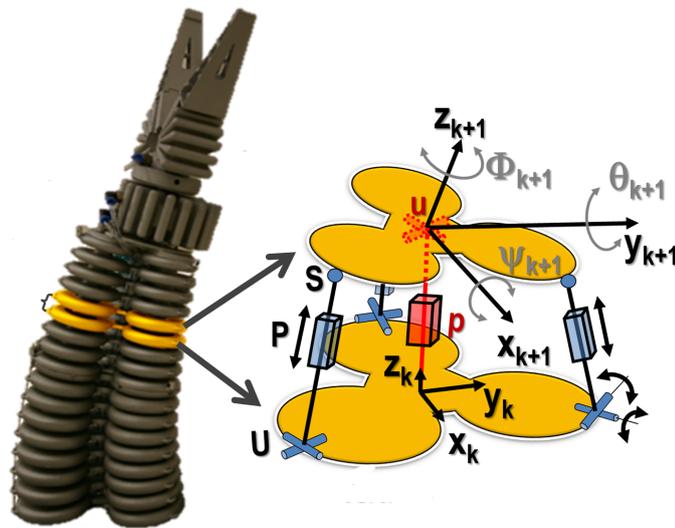


FIGURE 1.2 – Modélisation géométrique d'un manipulateur continu

Les manipulateurs continus sont conçus avec des matériaux flexibles et héritent de comportements non stationnaires dus aux effets d'hystérésis de certains actionneurs, à la viscoélasticité et à la perte de certaines propriétés physiques des matériaux qui les composent. Contrairement à leurs homologues rigides, constitués de corps rigides, et dont le MGD peut être facilement dérivé, ces caractéristiques rendent leur modélisation géométrique difficile à établir, Figure 1.2. Les contributions sur le MGD des manipulateurs continus peuvent être résumées en deux approches principales, à savoir les approches basées sur le modèle et les approches basées sur l'apprentissage. Les approches basées sur le modèle impliquent l'établissement d'un modèle géométrique basé sur des hypothèses approximatives concernant la structure physique du robot manipulateur. En ce qui concerne les approches basées sur l'apprentissage, l'espace paramétrique est divisé en plusieurs groupes en fonction des modes de fonctionnement du robot. Un modèle ma-

thématique dérivé des algorithmes d'apprentissage permet d'établir une relation entre les effets (observation d'experts, mesures de capteurs et données statistiques) et les causes (références d'entrée).

En ce qui concerne les approches basées sur des modèles, diverses méthodes ont été proposées pour apporter une solution au MGD des manipulateurs continus [Singh 2018, Godage 2015, Falkenhahn 2014, Falkenhahn 2015, Lakhali 2015]. Pour établir la modélisation géométrique directe d'un manipulateur continu composé de 4 sections, [Hannan 2003] ont considéré que le mouvement de flexion d'une section du manipulateur pouvait être mieux décrit en utilisant une courbure constante [Webster III 2010]. [Godage 2011] ont utilisé des fonctions de formes qui incorporent la structure du manipulateur pour modéliser un bras continu à sections multiples et à longueur variable. [Escande 2012] ont utilisé l'approche de la courbure constante pour modéliser un manipulateur bionique à deux sections imitant la trompe de l'éléphant. La même approche a été utilisée par [Rolf 2012] pour la géométrie directe de l'assistant de manipulation bionique (BHA). Cependant, l'hypothèse d'une courbure constante n'est pas toujours vérifiée, en particulier pour les manipulateurs continus coniques, et les contributions susmentionnées conduisent généralement à des modèles MGD imprécis. Pour améliorer la précision des modèles géométriques, la courbure variable a été mise en oeuvre par certains chercheurs. Ainsi, [Chirikjian 1994] ont considéré la colonne dorsale du robot continu comme une série d'un très grand nombre de corps et ont proposé une forme de courbe mathématique particulière pour résoudre la géométrie directe. Cependant, la méthode proposée est limitée aux robots de type continu avec un ensemble limité de formes et de mouvements. [Jones 2006] ont utilisé la même approche en formulant le problème de la géométrie du tronc comme une série de substitutions appliquées à une matrice de transformation homogène modifiée et calculée à l'aide de l'approche de Denavit-Hartenberg (D-H). [Mahl 2014] ont proposé une courbure variable pour modéliser un manipulateur AMB comme une série de segments à courbure constante (CC). Cependant, bien que ces contributions améliorent de manière significative les performances des modèles géomé-

triques obtenus, elles demeurent médiocres.

Des techniques basées sur l'apprentissage ont également été explorées pour fournir une solution à la géométrie directe des manipulateurs continus [Boutchouang 2021, Frazelle 2020, Thuruthel 2017, Thuruthel 2018, Goharimanesh 2020]. Ces techniques sont plus attrayantes car, elles peuvent tolérer une large gamme d'incertitudes et gérer des non-linéarités sévères. Ainsi, [Giorelli 2013b] ont utilisé un réseau de neurones non récurrents capable de déterminer la position de l'extrémité d'un manipulateur souple se déplaçant dans un espace tridimensionnel en fonction des forces appliquées aux câbles qui entraînent ce manipulateur à courbure non constante. Pour faire face aux incertitudes du système, [He 2015] ont proposé des réseaux de neurones adaptatifs qui approximent le modèle inconnu du robot et adaptent les interactions robot-patient. [Melingui 2014c] ont proposé deux architectures de réseaux neuronaux, à savoir les réseaux à base radiale et les perceptrons, pour fournir une solution à la géométrie directe du CBHA. Cependant, bien que tous ces modèles d'apprentissage conduisent à des modèles plus ou moins précis que les approches basées sur les modèles, choisir le bon modèle d'apprentissage reste une tâche difficile.

Cette thèse présente une étude comparative sur les performances des modèles RBF, MLP, CANFIS et SVR dans le cadre de la modélisation géométrique directe, de manipulateurs continus à plusieurs sections. Un manipulateur continuum à deux sections, appelé CBHA, a été utilisé comme étude de cas. Les échantillons de la base de données d'apprentissage sont collectés à l'aide d'un système de capture de mouvement de haute précision [Nagyauté 2018]. La comparaison est effectuée sur la base d'une analyse coefficient de corrélation [Taylor 1990]. Un coefficient de corrélation représente le degré d'association linéaire entre deux variables. Plus le coefficient R est proche de 1, quelle que soit la direction, plus l'association existante est forte, indiquant une relation linéaire entre les deux variables. En outre, les performances sont également évaluées en fonction du temps d'apprentissage, de l'erreur quadratique moyenne sur l'ensemble de test et des erreurs cartésiennes restantes.

1.3.2 Modélisation géométrique inverse

Dans le cadre de la modélisation géométrique inverse, les méthodes les plus largement proposées pour résoudre le problème géométrie inverse des manipulateurs redondants sont basées sur la pseudo-inverse jacobienne [Tzafestas 1996], [Mayorga 1992]. Pour les manipulateurs continus, plusieurs méthodes numériques basées sur la matrice jacobienne ont été proposées [Jones 2006], [Shi 2004]. Cependant, pour les manipulateurs continus avec de nombreux degrés de liberté, la charge de calcul du Jacobien pseudo-inverse devient prohibitive. En outre, la structure non linéaire de leur modèle mathématique ne permet pas souvent de trouver les solutions analytiques requises pour le calcul des éléments de la matrice jacobienne [Melinguì 2017, Giorelli 2015, Thuruthel 2018, Goharimanesh 2020]. [Giorelli 2013a] ont utilisé un réseau de neurones à action directe pour approximer le modèle géométrique inverse (MGI) d'un manipulateur souple à courbure non constante piloté par trois câbles. Dans cette thèse, une approche d'apprentissage supervisé distal (DSL) basée sur un réseau neuronal est proposée pour la modélisation géométrique inverse.

1.4 Modélisation et Commande des mobiles manipulateurs continus

En ce qui concerne la modélisation et la commande des manipulateurs mobiles continus [Mbakop 2020, McMahan 2006, Li 2015, Hawks 2019], Figure 1.3, la plupart des efforts ont été consacrés à la modélisation et la commande de la partie manipulatrice. [McMahan 2006] ont réalisé une série de tests dans l'eau et en extérieur d'un manipulateur mobile continuum à chenilles appelé Octarm afin de démontrer son adaptabilité dans des environnements contraints. [Li 2015] ont résolu le problème de la modélisation géométrique inverse de ce robot, en tenant compte de la pose de l'effecteur final, des contraintes environnementales et des obstacles environnants. L'approche consiste à trouver une configuration réalisable du manipulateur proche de sa configuration précédente.

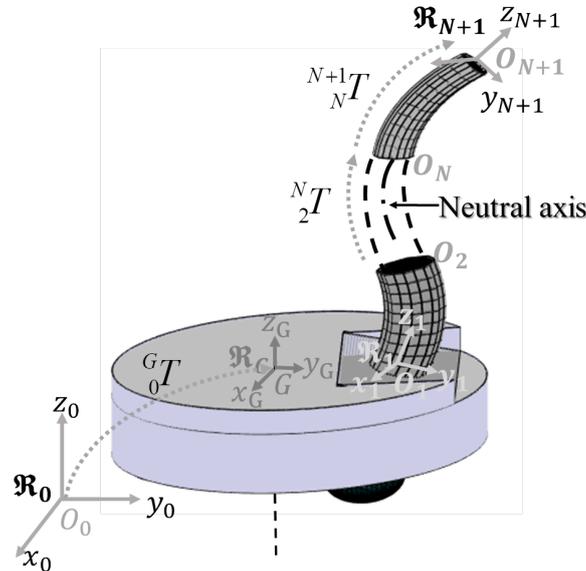


FIGURE 1.3 – Repères attachés à un mobile manipulateur continu

Ainsi, en partant de la section liée à l'effecteur vers la section liée à la plateforme mobile, la configuration réalisable de chaque section est calculée. Ensuite, une méthode analytique tenant compte des contraintes spécifiques sur la pose de chaque section est utilisée pour calculer la géométrie inverse de l'Octarm section par section. Cependant, l'algorithme de résolution de la géométrie inverse prend beaucoup de temps, car il nécessite un calcul itératif pour trouver une configuration possible de chaque section. Pour planifier le mouvement d'un manipulateur continuum mobile à section unique, [Hawks 2019] ont proposé l'algorithme RRT/A*, qui intègre l'espace de configuration de la section continuum du robot ainsi que ses contraintes spécifiques.

Dans cette thèse, nous développons un système de coordination qui intègre les deux contrôleurs précédents (parties mobile et manipulatrice) pour contrôler l'ensemble du système, notamment le mobile manipulateur continu.

1.5 Conclusion

Ce chapitre a présenté les points principaux de la revue de la littérature sur la modélisation et la commande des robots continus. Il a été divisé en trois parties, la première a présenté la revue de la littérature sur la navigation autonome des robots mobiles ou la plupart des méthodes sont mises en oeuvre seules, et ont conduit à de mauvaises performances. **Le chapitre suivant propose une hybridation de trois méthodes (APF, Logique floue et Réseau de neurones) pour améliorer l'intelligence et obtenir des performances acceptables (robustesse et rapidité).**

La deuxième s'est intéressée à la modélisation des manipulateurs continus, ou le choix de modèles d'apprentissage appropriés reste difficile car les paires de données d'échantillons collectées pour l'apprentissage ne fournissent pas d'informations sur le modèle d'apprentissage à utiliser. **Nous présentons également dans le chapitre suivant une étude comparative expérimentale et structurelle des quatre modèles d'apprentissage (MLP, RBF, SVR et CANFIS) couramment utilisés.**

Et enfin, la dernière partie sur la modélisation et la commande des manipulateurs mobiles continus nous a permis de positionner notre travail par rapport à la littérature sur la modélisation et la commande des manipulateurs mobiles continus encore embryonnaire. **Le chapitre suivant se terminera par la présentation d'un système de coordination neuronal pour le contrôle de l'ensemble du système (mobile manipulateur continuum).**

Méthodologies et Matériels utilisés

2.1 Introduction

Ce chapitre présente nos différentes contributions sur la commande des manipulateurs mobiles continus. Nous avons contribué aussi bien sur la partie mobile que sur la partie manipulatrice. Nous présentons essentiellement des lois de commande proposées. Nous commencerons par présenter notre modeste contribution sur la navigation autonome des robots mobiles. La modélisation géométrique des manipulateurs continus suivra, et enfin, la modélisation de l'ensemble clôturera le chapitre.

2.2 Navigation autonome des robots mobiles

Notre contribution dans cette partie porte sur l'amélioration de la contribution proposée dans [Melinguì 2014d]. En effet, bien que cette approche soit efficace, elle présente les problèmes de temps de calcul causés par l'algorithme de réduction de type. Pendant nos travaux, nous avons amélioré ce temps de calcul en remplaçant cet algorithme de réduction de type par deux couches de réseaux de neurones [Imrane 2021].

2.2.1 Description de la plate-forme expérimentale

Le robot mobile Robotino représenté dans la Figure 2.1 est un robot mobile holonome développé par la société Festo Didactic. Il comporte trois unités d'entraînement omnidirectionnelles montées à un angle de 120° les unes aux autres. Trois moteurs à courant

continu avec des codeurs optiques, ayant une résolution de 2048 incrément par tour, et les réducteurs avec un rapport de transmission de 1 : 16 et une vitesse allant jusqu'à 10 km/h. Les trois modules d'entraînement du Robotino sont intégrés dans un châssis en acier inoxydable soudé au laser solide, qui contient neuf capteurs de distance infrarouge, un capteur inductif et des capteurs analogiques optiques réfléchissants. L'alimentation est fournie par deux batteries de 12V de gel de plomb qui permettent un temps de fonctionnement jusqu'à deux heures. Les dimensions du robot sont de 370 mm de diamètre et 210 mm de hauteur avec un poids total d'environ 11kg.

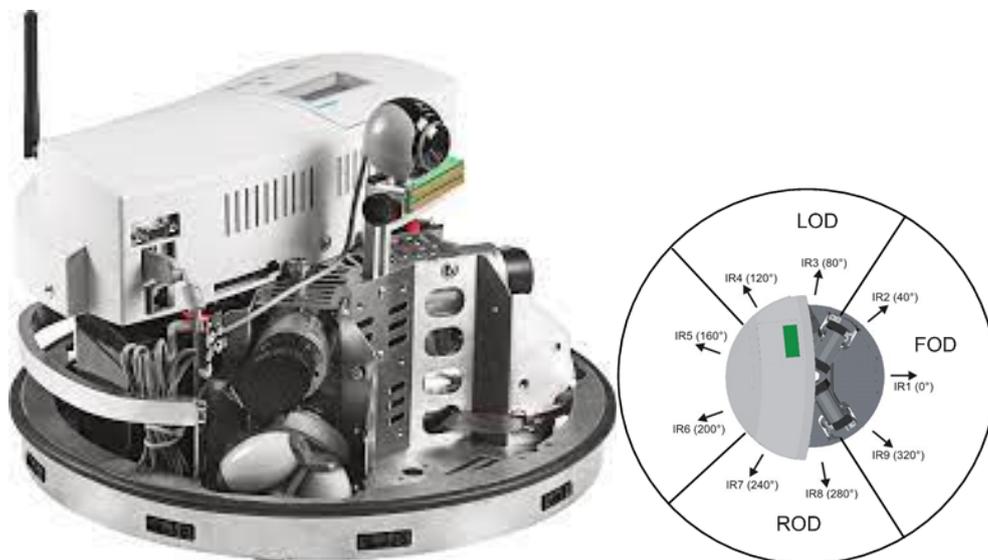


FIGURE 2.1 – La plateforme mobile Robotino avec capteurs infrarouges.

2.2.2 Logique floue à intervalle de type-2

Cette partie se concentre sur la description du contrôleur IT2FL développé dans [Melinguì 2014d]. Nous avons trouvé cette description nécessaire car le contrôleur IT2FNN proposé est une extension de ce dernier. Le contrôleur IT2FL développé dans [Melinguì 2014d] est une hybridation des concepts de logique floue et de champ de potentiel artificiel dans une même architecture de commande. La méthode APF pour la navigation des robots mobiles est efficace en termes de temps de calcul et prend en compte

à la fois les problèmes d'évitement d'obstacles et de planification de trajectoire, mais se heurte au problème des minima locaux. D'autre part, la navigation par la méthode de la logique floue peut être réalisée sans problème de minima locaux, mais elle est coûteuse en temps de calcul lorsque la taille de la base des règles floues augmente. Ces auteurs ont donc jugé nécessaire de combiner les deux méthodes pour en exploiter le meilleur. Cette hybridation avait pour objectif d'exploiter l'efficacité de calcul de la méthode APF et la capacité des méthodes de logique floue à obtenir des comportements de navigation sans problèmes de minima locaux. Après avoir remarqué que la plupart des situations de collision se produisaient lorsque la force attractive et la force répulsive de l'APF étaient colinéaires, l'environnement de navigation a été divisé en trois zones, comme le montre la Figure 2.2. Deux zones où il est difficile de rencontrer des situations de minima locaux et une autre où le robot peut rencontrer des situations de minima locaux (Figure 2.3). Le défi était d'associer ces méthodes sans utiliser un système de commutation, qui implique généralement quelques oscillations au moment de la transition d'un contrôleur à un autre. Pour y parvenir, un contrôleur APF a d'abord été conçu en utilisant les informations de différents capteurs, et ses sorties ont été utilisées comme entrées supplémentaires pour le contrôleur IT2FL. Ainsi, pendant la navigation dans les zones sans minima locaux, les sorties du contrôleur APF ont été directement transférées à la sortie du contrôleur IT2FL en utilisant la fonction d'appartenance "none". D'autre part, IT2FL n'a été utilisé que dans la zone intermédiaire. Cette association a permis de réaliser la navigation autonome du robot mobile appelé Robotino avec seulement 15 règles floues, comme le montre le tableau 2.1. Ce nombre de règles floues est nettement inférieur aux 40 règles et 128 règles développées pour la même plateforme respectivement dans [Melinguì 2013a] et [Oltean 2010].

Les variables d'état suivantes du robot $(d_l, d_f, d_r, \Delta X, \alpha, F_x, F_y)$ ont été utilisées comme entrées du contrôleur IT2FL et la vitesse longitudinale V_x et la vitesse latérale V_y comme sorties. d_l , d_f et d_r représentent respectivement les distances entre le robot et l'obstacle(Figure 2.4) le plus proche sur la gauche, l'avant et la droite du robot. ΔX est la

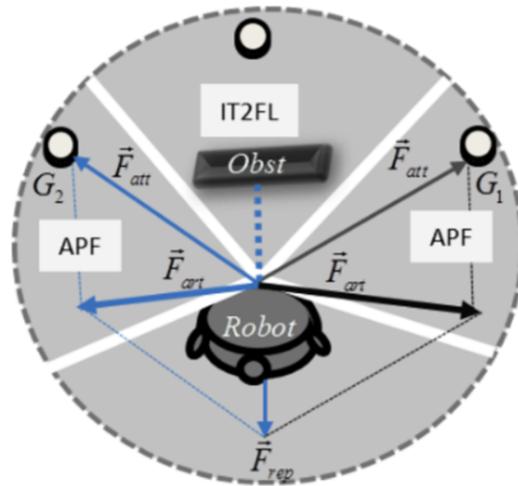


FIGURE 2.2 – Direction de la force potentielle artificielle lorsque le robot fait face à un obstacle frontal en cas d'orientation négative (\widetilde{NE}) ou positive (\widetilde{PO}) du point cible par rapport à la position du robot.

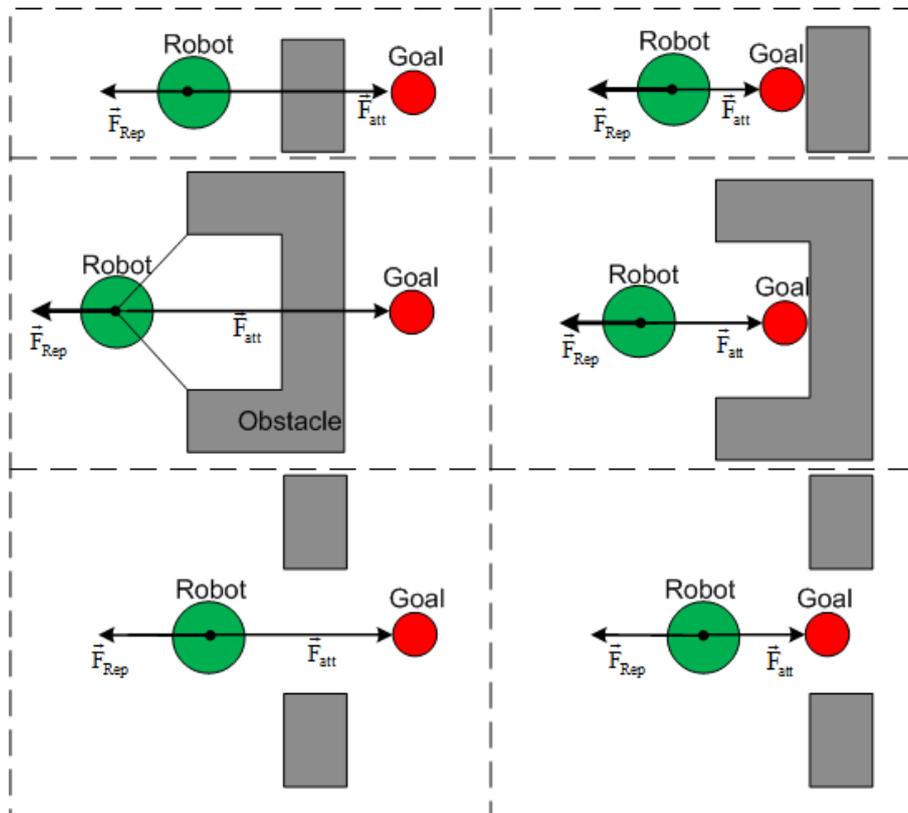


FIGURE 2.3 – Quelques situations de minima locaux.

distance entre la position actuelle du robot et la position de la cible et α est l'orientation de la cible par rapport à la position actuelle du robot. F_x et F_y représentent respectivement les forces artificielles longitudinales et latérales générées par le contrôleur APF. Les variables linguistiques d_l , d_f et d_r ont été modélisées par une fonction d'appartenance de type-2 à intervalle rectangulaire (RIT-2-MF) et deux fonctions d'appartenance (MF) gaussiennes de type-2 à intervalle (IT-2-GMF) représentées respectivement par les étiquettes "None" ($\bar{N}one$), "Near" " \bar{N} " et "Far" (\bar{F}). La distance entre le robot et la position de la cible a été modélisée par un RIT-2-MF et deux IT-2-GMF représentés respectivement par les étiquettes "None" ($\bar{N}one$), "Near" " \bar{N} " et "Far" (\bar{F}). L'orientation de la cible par rapport à la position actuelle du robot α , la force artificielle longitudinale (F_x) et la force artificielle latérale (F_y) ont été modélisées par une RIT-2-MF et trois IT-2-GMF représentées respectivement par les étiquettes "None" ($\bar{N}one$), "Negative" ($\bar{N}E$), "Zro" (\bar{Z}) et "Positive" ($\bar{P}O$). Enfin, les sorties, c'est-à-dire, la vitesse longitudinale V_x et la vitesse latérale V_y ont été modélisées par cinq IT-2-GMFs représentés respectivement par les étiquettes "FastBackward" ($\bar{F}B\bar{W}$), "Backward" ($\bar{B}\bar{W}$), "Stop" (\bar{S}), "Faward" ($\bar{F}\bar{W}$), et "FastFaward" ($\bar{F}F\bar{W}$). Les degrés d'appartenance inférieur et supérieur du RIT-2-MF étiqueté "none" sont égaux à un. L'utilisation de la force artificielle longitudinale (F_x), de la force artificielle latérale (F_y) en tant que variables d'entrée supplémentaires, combinée à la fonction d'appartenance "None", permet une réduction significative du nombre de règles floues. Nous renvoyons le lecteur à [Melinguì 2014d, Melinguì 2013a] pour plus de détails sur le contrôleur IT2FL.

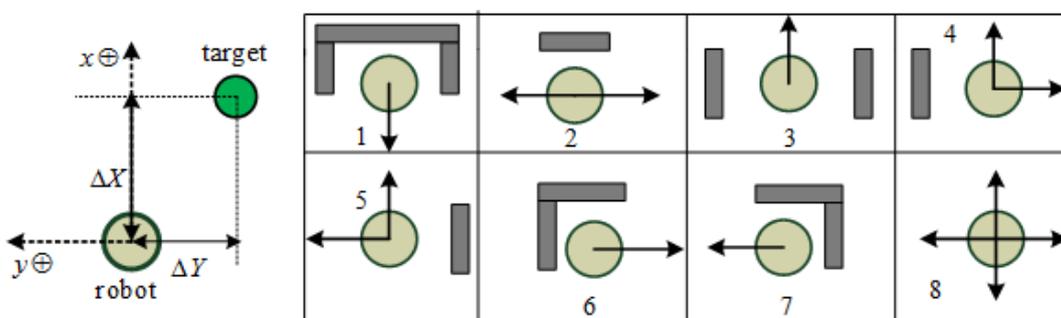


FIGURE 2.4 – Stratégies classiques de navigation.

TABLE 2.1 – Base de règles floues dans T2PFL

N^0	d_l	d_f	d_r	Δ_X	α	F_x	F_y	V_x	V_y
1	None	None	None	None	\widetilde{NE}	\widetilde{PO}	\widetilde{NE}	\widetilde{FW}	\widetilde{BW}
2	None	None	None	None	\widetilde{NE}	\widetilde{PO}	\tilde{Z}	\widetilde{FW}	\tilde{S}
3	None	None	None	None	\widetilde{NE}	\widetilde{PO}	\widetilde{PO}	\widetilde{FW}	\widetilde{FW}
4	None	None	None	None	\widetilde{PO}	\widetilde{PO}	\widetilde{NE}	\widetilde{FW}	\widetilde{BW}
5	None	None	None	None	\widetilde{PO}	\widetilde{PO}	\tilde{Z}	\widetilde{FW}	\tilde{S}
6	None	None	None	None	\widetilde{PO}	\widetilde{PO}	\widetilde{PO}	\widetilde{FW}	\widetilde{FW}
7	\tilde{F}	\tilde{F}	\tilde{F}	None	None	\tilde{Z}	\widetilde{NE}	\tilde{S}	\widetilde{FBW}
8	\tilde{F}	\tilde{F}	\tilde{F}	None	None	\tilde{Z}	\tilde{Z}	\tilde{S}	\tilde{S}
9	\tilde{F}	\tilde{F}	\tilde{F}	None	None	\tilde{Z}	\widetilde{PO}	\tilde{S}	\widetilde{FFW}
10	\tilde{F}	\tilde{F}	\tilde{N}	None	\tilde{Z}	None	None	\widetilde{FW}	\tilde{S}
11	\tilde{N}	\tilde{F}	\tilde{F}	None	\tilde{Z}	None	None	\widetilde{FW}	\tilde{S}
12	\tilde{F}	\tilde{N}	\tilde{F}	\tilde{N}	\tilde{Z}	None	None	\tilde{S}	\tilde{S}
13	\tilde{F}	\tilde{N}	\tilde{N}	\tilde{N}	\tilde{Z}	None	None	\tilde{S}	\tilde{S}
14	\tilde{N}	\tilde{N}	\tilde{F}	\tilde{N}	\tilde{Z}	None	None	\tilde{S}	\tilde{S}
15	\tilde{N}	\tilde{N}	\tilde{N}	\tilde{N}	\tilde{Z}	None	None	\tilde{S}	\tilde{S}

2.2.3 Contrôleur neuroflou à intervalle de type-2

Dans cette section, le contrôleur neuro-flou à intervalle de type-2 (IT2FNN) proposé est implémenté dans le cas de la navigation autonome des robots mobiles. L'idée centrale étant d'émuler le contrôleur proposé dans [Melinguì 2014d], tout en le rendant efficace en temps de calcul, les mêmes variables d'entrée (d_l , d_f , d_r , ΔX , α , F_x , F_y) et de sortie (V_x , V_y) que celles décrites précédemment ont été utilisées.

L'architecture de contrôle est donnée dans la Figure 2.5. Les fonctions d'appartenance des variables linguistiques relatives à la distance (d_l , d_f , d_r , ΔX) avec les paramètres

de réglage initiaux sont représentées dans la Figure 2.6. Les fonctions d'appartenance de l'orientation de la cible par rapport à la position actuelle du robot α et les forces artificielles longitudinales F_x et latérales F_y sont représentées respectivement sur les Figure 2.7 et Figure 2.8. Compte tenu du nombre de variables d'entrée et des fonctions d'appartenance associées, le nombre de règles floues devrait être de 432 règles floues, ce qui est important. Cependant, en tenant compte des avantages des approches APF et logique floue, ce nombre peut être grandement réduit (15 règles floues, [Melinguì 2014d]). Les règles floues développées dans le cas du contrôleur IT2FL dans [Melinguì 2014d] sont ajustées dans cette thèse en utilisant un processus de rétropropagation via une architecture neuro-floue. L'écart type σ , la moyenne incertaine qui prend des valeurs dans $[m_{k1}^i, m_{k2}^i]$, l'intervalle des centroïdes $[w_l^i, w_r^i]$ de la partie conséquente, et les q-facteurs de conception $[q_l, q_r]$ sont réglés en utilisant les équations de rétropropagation développées dans la sous-section suivante.

Les stratégies simples de navigation des robots mobiles telles que l'atteinte d'une cible, le suivi de murs, le suivi de couloirs, l'évitement d'obstacles de forme U (Figure 2.4) peuvent impliquer des milliers de données d'entraînement. Cependant, il est difficile pour les concepteurs d'avoir une connaissance spécifique a priori de tous les échantillons d'entraînement. Par conséquent, l'IT2FNN proposé est formé directement avec des règles floues, et avec quelques données d'échantillon numérique obtenues en utilisant le contrôleur IT2FL développé dans [Melinguì 2014d]. Les variables d'entrée ($d_l, d_f, d_r, \Delta X, \alpha, F_x, F_y$) et les variables de sortie (V_x, V_y) sont enregistrées pour chaque combinaison de variables d'entrée. Les vecteurs suivants $5 : 10 : 308 ; (cm)$, $0 : 50 : 300 ; (cm)$, $-60 : 15 : 60 ; (degr)$, et $-100 : 50 : 100 ; (N)$ sont respectivement considérés dans le processus de discrétisation pour les distances aux obstacles (d_l, d_f, d_r), la distance à la cible (ΔX), l'orientation du robot par rapport à la position de la cible (α), et les forces artificielles longitudinales et latérales (F_x, F_y). Un nombre total de 137781 échantillons est enregistré. Les moyennes m_{k1}^i et m_{k2}^i , l'écart-type σ , les facteurs de pondération conséquents w_l^i et w_r^i , et les q-facteurs q_l et q_r sont respectivement réglés pour que les sorties du contrôleur IT2FNN

soient proches des sorties IT2FL. L'objectif du processus d'apprentissage est d'abord de se comporter comme le contrôleur IT2FL en ce qui concerne l'évitement des obstacles et l'atteinte de la cible et d'améliorer les performances de généralisation pour la navigation autonome des robots mobiles.

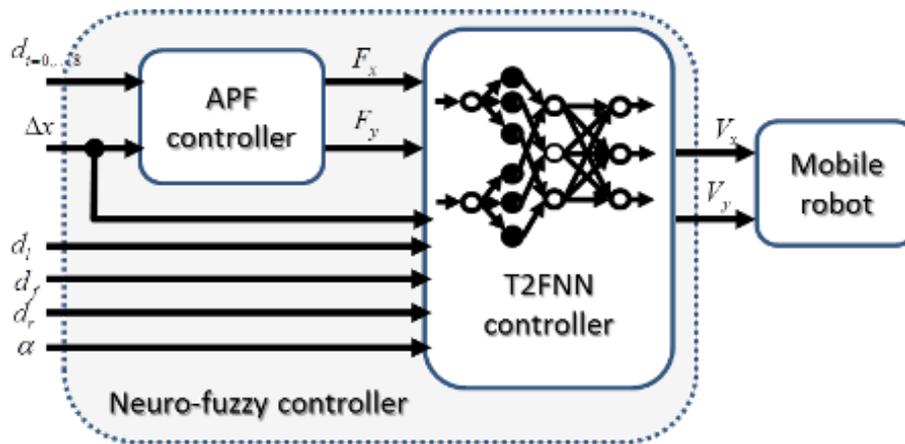


FIGURE 2.5 – Structure de contrôle IT2FNN

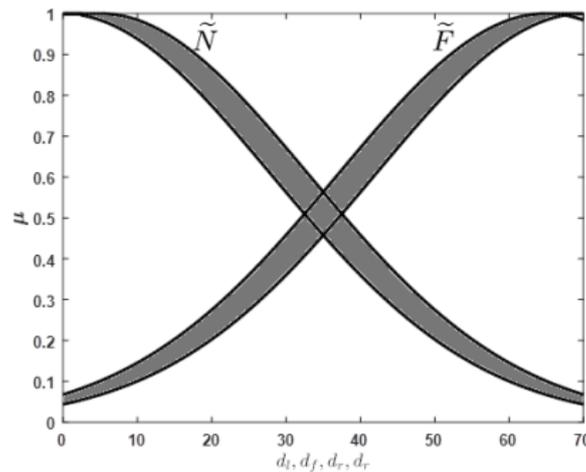


FIGURE 2.6 – Les distances entre le robot et les obstacles les plus proches sur ses côtés gauche, avant et droit, et étiquetées comme "Near" (\tilde{N}) et "Far" (\tilde{F}).

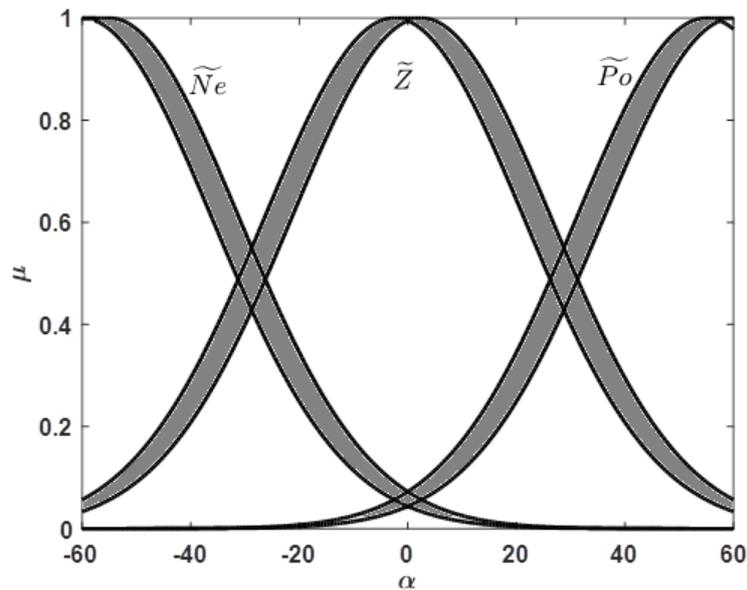


FIGURE 2.7 – L'orientation de la position de la cible par rapport à la position actuelle du robot, modélisée par trois fonctions d'appartenance gaussiennes de type-2 à intervalle et étiquetée comme suit "Négative" (\widetilde{NE}), "Zero" (\widetilde{Z}) "Positive" (\widetilde{PO}).

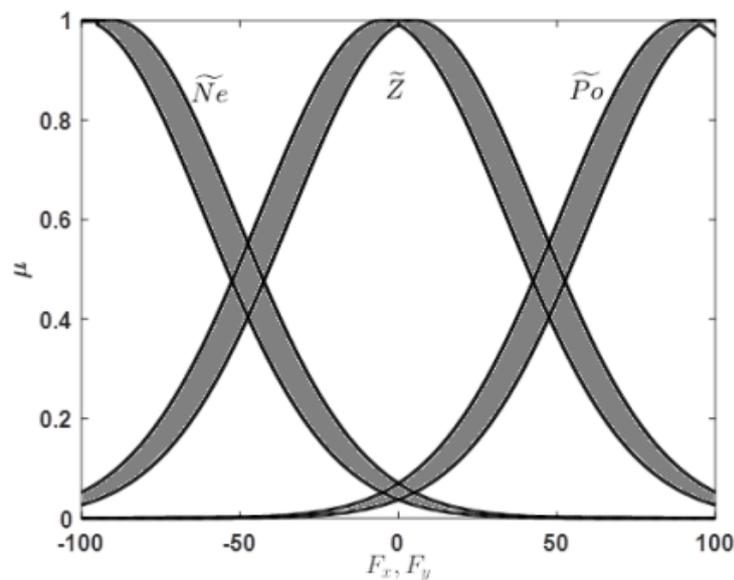


FIGURE 2.8 – La force artificielle longitudinale (F_x) et la force artificielle latérale (F_y) modélisées par trois fonctions d'appartenance gaussiennes de type-2 à intervalle et étiquetées comme "Négative" (\widetilde{NE}), "Zéro" (\widetilde{Z}) "Positive" (\widetilde{PO}).

2.2.4 Architecture IT2FNN proposée

La topologie du contrôleur IT2FNN se compose de quatre couches, comme illustré sur la figure 2.9. Les couches I et II forment la partie antérieure de l'IT2FNN, tandis que les couches III et IV forment la partie conséquente. La fonction de chaque couche est détaillée ci-dessous.

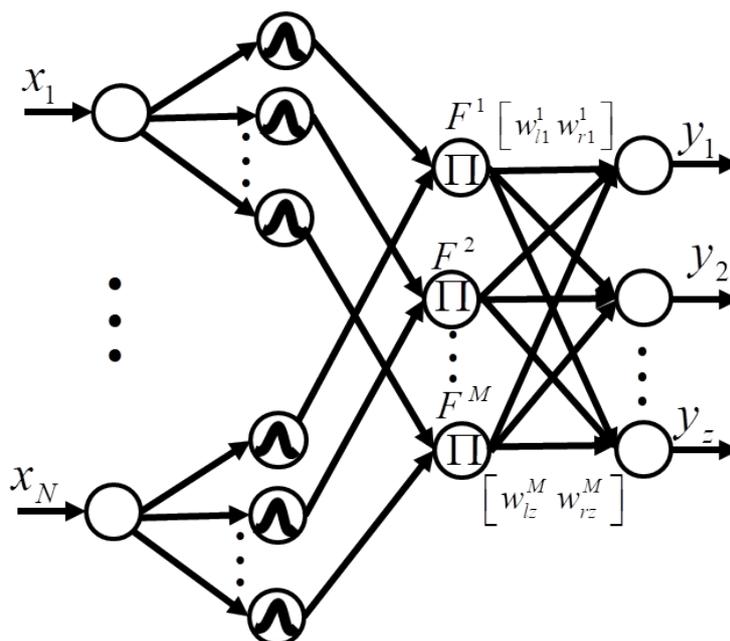


FIGURE 2.9 – Structure du contrôleur flou de type-2 à intervalle [Wang 2004]

(i) **Couche I** (couche d'entrée) :

Dans cette couche, chaque noeud représente une variable d'entrée précise. Cette couche reçoit la variable d'état actuel (x_1, x_2, \dots, x_N) du robot en entrée.

(ii) **Couche II** (couche de fuzzification) :

Cette couche est en charge de l'opération de fuzzification. Dans cette couche, chaque noeud est constitué d'une fonction d'appartenance principale gaussienne ayant un écart-type fixe σ et une moyenne incertaine qui prend des valeurs en $[m_{k_1}^i, m_{k_2}^i]$. Les fonctions d'appartenance supérieure et inférieure pour une entrée x_k et une règle i peuvent être écrites comme suit :

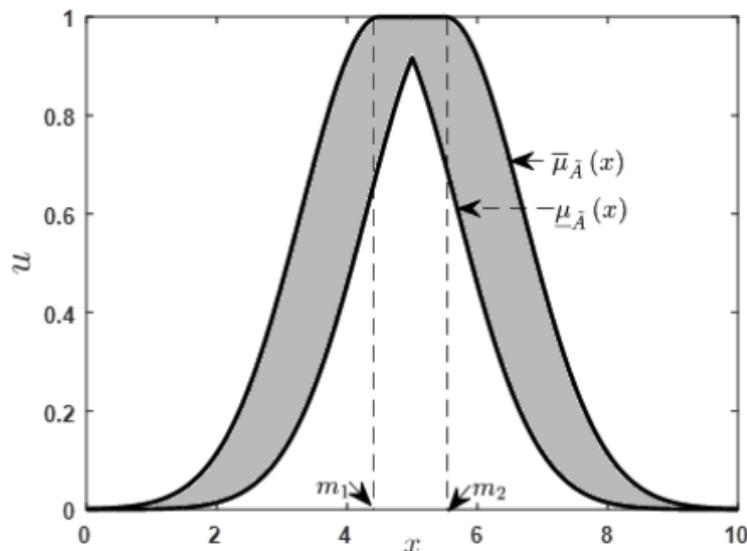


FIGURE 2.10 – Ensemble flou à intervalle de type-2 avec une moyenne incertaine et un écart-type fixe [Wang 2004].

$$\bar{u}_{\tilde{F}_j^i}(x_k) = \begin{cases} N(m_{k_1}^i, \sigma_k^i, x_k), & x_k < m_{k_1}^i \\ 1, & m_{k_1}^i \leq x_k \leq m_{k_2}^i \\ N(m_{k_2}^i, \sigma_k^i, x_k), & x_k > m_{k_2}^i \end{cases} \quad (2.1)$$

$$\underline{u}_{\tilde{F}_j^i}(x_k) = \begin{cases} N(m_{k_2}^i, \sigma_k^i, x_k), & x_k \leq \frac{m_{k_1}^i + m_{k_2}^i}{2} \\ N(m_{k_1}^i, \sigma_k^i, x_k), & x_k > \frac{m_{k_1}^i + m_{k_2}^i}{2} \end{cases} \quad (2.2)$$

Où $N(m_k^i, \sigma_k^i, x_k) = \exp\left[-\frac{1}{2}\left(\frac{x_k - m_k^i}{\sigma_k^i}\right)^2\right]$. Ainsi, la sortie de cette couche peut être exprimée comme un intervalle de type-1 $[\bar{u}_{\tilde{F}_j^i}(x_k), \underline{u}_{\tilde{F}_j^i}(x_k)]$. Un ensemble flou de type-2 à moyenne incertaine est illustré à la Figure 2.10.

(iii) **Couche III** (couche d'inférence) :

Dans cette couche, chaque noeud calcule la puissance d'inférence. La puissance d'inférence F^i de l'IT2FNN pour la i ème règle est un intervalle de type-1, exprimé comme suit :

$$F^i = \left[\underline{f}^i, \overline{f}^i \right] \quad (2.3)$$

Avec

$$\underline{f}^i = \underline{u}_{\tilde{F}_1^i}(x_1) * \dots * \underline{u}_{\tilde{F}_n^i}(x_n) = \prod_{j=1}^n \underline{u}_{\tilde{F}_1^i}(x_j) \quad (2.4)$$

$$\overline{f}^i = \overline{u}_{\tilde{F}_1^i}(x_1) * \dots * \overline{u}_{\tilde{F}_n^i}(x_n) = \prod_{j=1}^n \overline{u}_{\tilde{F}_1^i}(x_j) \quad (2.5)$$

Où $\underline{u}_{\tilde{F}_1^i}(x_j^p)$ et $\overline{u}_{\tilde{F}_1^i}(x_j^p)$ représentent respectivement les valeurs des fonctions d'appartenance inférieur et supérieur. * et n font références à la norme t minimale ou au produit et au nombre d'entrées de la i ème règle de l'IT2FNN. La règle SI-ALORS de l'IT2FNN s'exprime comme suit :

$$\begin{aligned} \text{R}^i: \text{SI } x_1 \text{ est } \tilde{F}_1^i, \text{ et } x_2 \text{ est } \tilde{F}_2^i, \text{ et } \dots, \text{ et } x_n \text{ est } \tilde{F}_n^i \\ \text{ALORS } y_1 \text{ est } [w_{l1}^i, w_{r1}^i], \text{ et } \dots, \text{ et } y_Z \text{ est } [w_{lZ}^i, w_{rZ}^i] \end{aligned} \quad (2.6)$$

iv **Couche IV** (couche de traitement de sortie) :

Cette couche combine les sorties de la couche III ($\underline{f}^i, \overline{f}^i$), l'ensemble d'intervalles centroïdes $[w_l^i, w_r^i]$ de la partie conséquente [Mendel 2017], et les q-facteurs de conception $[q_l, q_r]$ permettant un ajustement adaptatif des valeurs supérieures et inférieures sans utiliser la procédure itérative K-M pour trouver les points d'extrémité L et R [Lin 2013]. L'utilisation des q-facteurs réduit considérablement la complexité des systèmes de logique floue de type-2. Elle rend ces derniers adaptés aux applications en temps réel telles que la navigation autonome de robots mobiles, vu la taille importante de leur base de règles. En utilisant les q-facteurs, les points d'extrémité gauche y_l et droite y_r peuvent être calculés comme suit :

$$y_l = \frac{(1 - q_l) \sum_{i=1}^M \underline{f}^i w_l^i + q_l \sum_{i=1}^M \overline{f}^i w_l^i}{\sum_{i=1}^M \underline{f}^i + \overline{f}^i} \quad (2.7)$$

$$y_r = \frac{(1 - q_r) \sum_{i=1}^M \bar{f}^i w_r^i + q_r \sum_{i=1}^M \underline{f}^i w_r^i}{\sum_{i=1}^M \underline{f}^i + \bar{f}^i}. \quad (2.8)$$

Enfin, chaque sortie defuzzifiée $y(\vec{x})$ de l'IT2FNN est la moyenne de y_l et y_r définie comme suit :

$$y(\vec{x}) = \frac{y_l + y_r}{2}. \quad (2.9)$$

Réduction de type

La réduction de type [Mendel 2006] est le processus qui permet de passer de la logique floue de type-2 à la logique floue de type-1. Ce travail vise à rendre le contrôleur IT2FL proposé par [Melinguì 2014d] efficace en termes de temps de calcul en remplaçant l'opération de réduction de type (K-M proposé par [Melinguì 2014d]) par deux q-facteurs de conception q_l et q_r qui seront appris à l'aide d'un algorithme de rétropropagation. L'idée est d'émuler ce contrôleur tout en le rendant efficace en termes de temps de calcul.

Algorithme de rétropropagation

Ici nous cherchons à obtenir du réseau une réponse préétablie comme étant correcte. Le processus d'apprentissage de l'IT2FNN via l'algorithme de rétro-propagation minimise la fonction d'erreur suivante :

$$e^p = \frac{1}{2} [y(\vec{x}^p) - d^p]^2, \quad p = 1, \dots, P, \quad (2.10)$$

Avec P le nombre de données d'apprentissage (\vec{x}^p, d^p) . Soit η le taux d'apprentissage, les valeurs moyennes m_{k1}^i et m_{k2}^i , l'écart type σ , les facteurs de pondération conséquents w_l^i et w_r^i , les q-facteurs de la i ème règle q_l et q_r sont mis à jour comme suit :

$$m_{k1}^i(p+1) = m_{k1}^i(p) - \eta \left. \frac{\partial e^p}{\partial m_{k1}^i} \right|_p, \quad (2.11)$$

$$m_{k2}^i(p+1) = m_{k2}^i(p) - \eta \left. \frac{\partial e^p}{\partial m_{k2}^i} \right|_p, \quad (2.12)$$

$$\sigma_k^i(p+1) = \sigma_k^i(p) - \eta \left. \frac{\partial e^p}{\partial \sigma_k^i} \right|_p, \quad (2.13)$$

$$w_l^i(p+1) = w_l^i(p) - \eta \left. \frac{\partial e^p}{\partial w_l^i} \right|_p, \quad (2.14)$$

$$w_r^i(p+1) = w_r^i(p) - \eta \left. \frac{\partial e^p}{\partial w_r^i} \right|_p, \quad (2.15)$$

$$q_l^i(p+1) = q_l^i(p) - \eta \left. \frac{\partial e^p}{\partial q_l^i} \right|_p, \quad (2.16)$$

$$q_r^i(p+1) = q_r^i(p) - \eta \left. \frac{\partial e^p}{\partial q_r^i} \right|_p. \quad (2.17)$$

L'équation (2.11) peut être réécrite comme suit :

$$m_{k1}^i(p+1) = m_{k1}^i(p) - \eta \left[\left. \frac{\partial e^p}{\partial y(\vec{x})} \frac{\partial y(\vec{x})}{\partial y_l} \frac{\partial y_l}{\partial m_{k1}^i} \right|_p + \left. \frac{\partial e^p}{\partial y(\vec{x})} \frac{\partial y(\vec{x})}{\partial y_r} \frac{\partial y_r}{\partial m_{k1}^i} \right] \Big|_p. \quad (2.18)$$

De (2.18), (2.9) et (2.10), il s'ensuit que :

$$\left. \frac{\partial e^p}{\partial y(\vec{x})} \right|_p = y(\vec{x}) - d^p, \quad (2.19)$$

$$\left. \frac{\partial y(\vec{x})}{\partial y_l} \right|_p = 1/2, \quad (2.20)$$

$$\left. \frac{\partial y(\vec{x})}{\partial y_r} \right|_p = 1/2, \quad (2.21)$$

$$\left. \frac{\partial y_l}{\partial m_{k1}^i} \right|_p = \left[\frac{\partial y_l}{\partial \bar{u}_{\tilde{F}_k^i}(x_k)} \frac{\partial \bar{u}_{\tilde{F}_k^i}(x_k)}{\partial m_{k1}^i} + \frac{\partial y_l}{\partial \underline{u}_{\tilde{F}_k^i}(x_k)} \frac{\partial \underline{u}_{\tilde{F}_k^i}(x_k)}{\partial m_{k1}^i} \right], \quad (2.22)$$

$$\left. \frac{\partial y_r}{\partial m_{k1}^i} \right|_p = \left[\frac{\partial y_r}{\partial \bar{u}_{\tilde{F}_k^i}(x_k)} \frac{\partial \bar{u}_{\tilde{F}_k^i}(x_k)}{\partial m_{k1}^i} + \frac{\partial y_r}{\partial \underline{u}_{\tilde{F}_k^i}(x_k)} \frac{\partial \underline{u}_{\tilde{F}_k^i}(x_k)}{\partial m_{k1}^i} \right]. \quad (2.23)$$

De (2.22),(2.23), (2.1) et (2.2), les relations suivantes sont dérivées :

$$\frac{\partial \bar{u}_{\tilde{F}_k^i}(x_k)}{\partial m_{k1}^i} = \begin{cases} \frac{(x_k - m_{k1}^i) N(m_{k1}^i, \sigma_k^i, x_k)}{\sigma_k^{i2}}, & x_k < m_{k1}^i \\ 0, & m_{k1}^i \leq x_k \leq m_{k2}^i \\ 0, & x_k > m_{k2}^i \end{cases} \quad (2.24)$$

$$\frac{\partial \underline{u}_{\tilde{F}_k^i}(x_k)}{\partial m_{k1}^i} = \begin{cases} 0, & x_k \leq \frac{m_{k1}^i + m_{k2}^i}{2} \\ \frac{(x_k - m_{k1}^i) N(m_{k1}^i, \sigma_k^i, x_k)}{\sigma_k^{i2}}, & x_k > \frac{m_{k1}^i + m_{k2}^i}{2} \end{cases} \quad (2.25)$$

$$\frac{\partial \bar{u}_{\tilde{F}_k^i}(x_k)}{\partial \sigma_k^i} = \begin{cases} \frac{(x_k - m_{k1}^i)^2 N(m_{k1}^i, \sigma_k^i, x_k)}{\sigma_k^{i3}}, & x_k < m_{k1}^i \\ 0, & m_{k1}^i \leq x_k \leq m_{k2}^i \\ \frac{(x_k - m_{k2}^i)^2 N(m_{k2}^i, \sigma_k^i, x_k)}{\sigma_k^{i3}}, & x_k > m_{k2}^i \end{cases} \quad (2.26)$$

$$\frac{\partial \underline{u}_{\tilde{F}_k^i}(x_k)}{\partial \sigma_k^i} = \begin{cases} \frac{(x_k - m_{k2}^i)^2 N(m_{k2}^i, \sigma_k^i, x_k)}{\sigma_k^{i3}}, & x_k \leq \frac{m_{k1}^i + m_{k2}^i}{2} \\ \frac{(x_k - m_{k1}^i)^2 N(m_{k1}^i, \sigma_k^i, x_k)}{\sigma_k^{i3}}, & x_k > \frac{m_{k1}^i + m_{k2}^i}{2} \end{cases} \quad (2.27)$$

$$\frac{\partial y_l}{\partial \bar{u}_{\tilde{F}_k^i}(x_k)} = \frac{\left(\prod_{\substack{j=1 \\ j \neq k}}^n \bar{u}_{\tilde{F}_j^i} \right) (q_l w_l^i - y_l)}{\sum_{i=1}^M \underline{f}^i + \bar{f}^i}. \quad (2.28)$$

De la même manière,

$$\frac{\partial y_l}{\partial \underline{u}_{\tilde{F}_k^i}(x_k)} = \frac{\left(\prod_{\substack{j=1 \\ j \neq k}}^n \underline{u}_{\tilde{F}_j^i} \right) ((1 - q_l) w_l^i - y_l)}{\sum_{i=1}^M \underline{f}^i + \bar{f}^i}, \quad (2.29)$$

$$\frac{\partial y_r}{\partial \bar{u}_{\tilde{F}_k^i}(x_k)} = \frac{\left(\prod_{\substack{j=1 \\ j \neq k}}^n \bar{u}_{\tilde{F}_j^i} \right) (q_r w_r^i - y_r)}{\sum_{i=1}^M \underline{f}^i + \bar{f}^i}, \quad (2.30)$$

$$\frac{\partial y_r}{\partial \underline{u}_{\tilde{F}_k^i}(x_k)} = \frac{\left(\prod_{\substack{j=1 \\ j \neq k}}^n \underline{u}_{\tilde{F}_j^i} \right) ((1 - q_r) w_r^i - y_r)}{\sum_{i=1}^M \underline{f}^i + \bar{f}^i}. \quad (2.31)$$

Si $x_k^p < m_{k_1}^i$, donc en substituant (2.28), (2.24), (2.29) et (2.25) dans (2.22) [écrivons ceci (2.22')], puis en remplaçant (2.19), (2.20), (2.21), (2.22'), et (2.23') dans (2.18), nous aurons :

$$\begin{aligned}
m_{k_1}^i(p+1) &= m_{k_1}^i(p) \\
&- \left(\frac{1}{2} \eta \frac{(y(\bar{x}^p) - d^p)(x_k - m_{k_1}^i) N(m_{k_1}^i, \sigma_k^i, x_k)}{\sigma_k^{i2}} \right. \\
&\times \left. \frac{\left(\prod_{\substack{j=1 \\ j \neq k}}^n \bar{u}_{\bar{F}_j^i}(x_k) \right) [(q_l w_l^i + q_r w_r^i) - (y_l + y_r)]}{\sum_{i=1}^M \underline{f}^i + \bar{f}^i} \right). \tag{2.32}
\end{aligned}$$

Si $x_k^p \geq m_{k_1}^i$ et $x_k^p > \frac{1}{2} (m_{k_1}^i + m_{k_2}^i)$, et en suivant la même procédure, on obtient :

$$\begin{aligned}
m_{k_1}^i(p+1) &= m_{k_1}^i(p) \\
&- \left(\frac{1}{2} \eta \frac{(y(\bar{x}^p) - d^p)(x_k - m_{k_1}^i) N(m_{k_1}^i, \sigma_k^i, x_k)}{\sigma_k^{i2}} \right. \\
&\times \left. \frac{\left(\prod_{\substack{j=1 \\ j \neq k}}^n \bar{u}_{\bar{F}_j^i}(x_k) \right) [((1-q_l)w_l^i + (1-q_r)w_r^i) - (y_l + y_r)]}{\sum_{i=1}^M \underline{f}^i + \bar{f}^i} \right). \tag{2.33}
\end{aligned}$$

L'équation de rétropropagation pour $m_{k_2}^i$ peut être dérivée en suivant une procédure similaire à celle décrite ci-dessus.

Si $x_k^p < m_{k_1}^i$ (d'où également $x_k^p \leq (m_{k_1}^i + m_{k_2}^i)/2$)

$$\begin{aligned}
\sigma_k^i(p+1) &= \sigma_k^i(p) - \frac{1}{2} \eta \frac{(y(\bar{x}^p) - d^p)}{(\sigma_k^i)^3} \\
&\times \left[(x_k^p - m_{k_1}^i)^2 N(m_{k_1}^i, \sigma_k^i, x_k^p) \right. \\
&\frac{\left(\prod_{\substack{j=1 \\ j \neq k}}^n \bar{u}_{\bar{F}_j^i}(x_k) \right) (q_l w_l^i + q_r w_r^i - y_l - y_r)}{\sum_{i=1}^M \bar{f}^i + \underline{f}^i} \\
&+ (x_k^p - m_{k_2}^i)^2 N(m_{k_2}^i, \sigma_k^i, x_k^p) \\
&\left. \frac{\left(\prod_{\substack{j=1 \\ j \neq k}}^n \underline{u}_{\bar{F}_j^i}(x_k) \right) ((1-q_l)w_l^i + (1-q_r)w_r^i - y_l - y_r)}{\sum_{i=1}^M \bar{f}^i + \underline{f}^i} \right]. \tag{2.34}
\end{aligned}$$

Si $x_k^p > m_{k_2}^i$

$$\begin{aligned}
\sigma_k^i(p+1) &= \sigma_k^i(p) - \frac{1}{2}\eta \frac{(y(\bar{x}^p) - d^p)}{(\sigma_k^i)^3} \\
&\times \left[(x_k^p - m_{k_2}^i)^2 N(m_{k_2}^i, \sigma_k^i, x_k^p) \right. \\
&\quad \left. \frac{\left(\prod_{\substack{j=1 \\ j \neq k}}^n \bar{u}_{\bar{F}_j^i} \right) (q_l w_l^i + q_r w_r^i - y_l - y_r)}{\sum_{i=1}^M \bar{f}^i + \underline{f}^i} \right. \\
&\quad \left. + (x_k^p - m_{k_1}^i)^2 N(m_{k_1}^i, \sigma_k^i, x_k^p) \right. \\
&\quad \left. \frac{\left(\prod_{\substack{j=1 \\ j \neq k}}^n \underline{u}_{\bar{F}_j^i} \right) ((1-q_l)w_l^i + (1-q_r)w_r^i - y_l - y_r)}{\sum_{i=1}^M \bar{f}^i + \underline{f}^i} \right]. \tag{2.35}
\end{aligned}$$

Les facteurs de pondération conséquents peuvent être obtenus de manière similaire. Considérons w_l^i , l'équation de rétropropagation peut-être exprimée comme suit :

$$\begin{aligned}
w_l^i(p+1) &= w_l^i(p) \\
&\quad - \eta \left[\frac{\partial e^p}{\partial y(\bar{x})} \frac{\partial y(\bar{x})}{\partial y_l} \frac{\partial y_l}{\partial w_l^i} + \frac{\partial e^p}{\partial y(\bar{x})} \frac{\partial y(\bar{x})}{\partial y_r} \frac{\partial y_r}{\partial w_l^i} \right] \Big|_p, \tag{2.36}
\end{aligned}$$

Avec

$$\frac{\partial y_l}{\partial w_l^i(x_k^p)} = \frac{(1-q_l) \prod_{j=1}^n \underline{u}_{\bar{F}_j^i} + q_l \prod_{j=1}^n \bar{u}_{\bar{F}_j^i}}{\sum_{i=1}^M \underline{f}^i + \bar{f}^i}, \tag{2.37}$$

et

$$\frac{\partial y_r}{\partial w_l^i(x_k^p)} = 0. \tag{2.38}$$

En substituant (2.19), (2.20), (2.21) (2.37), et (2.38) dans (2.36)

$$\begin{aligned}
w_l^i(p+1) &= w_l^i(p) - \frac{1}{2}\eta (y(\bar{x}^p) - d^p) \\
&\quad \times \frac{(1-q_l) \prod_{j=1}^n \underline{u}_{\bar{F}_j^i} + q_l \prod_{j=1}^n \bar{u}_{\bar{F}_j^i}}{\sum_{i=1}^M \underline{f}^i + \bar{f}^i}. \tag{2.39}
\end{aligned}$$

En suivant la même procédure, w_r^i s'obtient comme suit :

$$\begin{aligned}
w_r^i(p+1) &= w_r^i(p) - \frac{1}{2}\eta (y(\bar{x}^p) - d^p) \\
&\quad \times \frac{(1-q_r) \prod_{j=1}^n \underline{u}_{\bar{F}_j^i} + q_r \prod_{j=1}^n \bar{u}_{\bar{F}_j^i}}{\sum_{i=1}^M \underline{f}^i + \bar{f}^i}. \tag{2.40}
\end{aligned}$$

Comme ci-dessus, on peut déduire l'évolution des q-facteurs q_l et q_r . Considérons q_l , l'équation rétropropagation peut-être écrite comme suit :

$$q_l(p+1) = q_l(p) - \eta \left[\frac{\partial e^p}{\partial y(\bar{x})} \frac{\partial y(\bar{x})}{\partial y_l} \frac{\partial y_l}{\partial q_l} + \frac{\partial e^p}{\partial y(\bar{x})} \frac{\partial y(\bar{x})}{\partial y_r} \frac{\partial y_r}{\partial q_l} \right] \Big|_p, \quad (2.41)$$

Avec

$$\frac{\partial y_l}{\partial q_l(x_k^p)} = \frac{-\sum_{i=1}^M \underline{f}^i w_i^i + \sum_{i=1}^M \bar{f}^i w_i^i}{\sum_{i=1}^M \underline{f}^i + \bar{f}^i}, \quad (2.42)$$

et

$$\frac{\partial y_r}{\partial q_l(x_k^p)} = 0. \quad (2.43)$$

En substituant (2.19), (2.20), (2.21) (2.42), et (2.43) dans (2.41), nous avons :

$$q_l(p+1) = q_l(p) - \frac{1}{2}\eta (y(\bar{x}^p) - d^p) \times \frac{-\sum_{i=1}^M \underline{f}^i w_i^i + \sum_{i=1}^M \bar{f}^i w_i^i}{\sum_{i=1}^M \underline{f}^i + \bar{f}^i}. \quad (2.44)$$

En suivant la même procédure, q_r s'obtient comme suit :

$$q_r(p+1) = q_r(p) - \frac{1}{2}\eta (y(\bar{x}^p) - d^p) \times \frac{-\sum_{i=1}^M \underline{f}^i w_r^i + \sum_{i=1}^M \bar{f}^i w_r^i}{\sum_{i=1}^M \underline{f}^i + \bar{f}^i}. \quad (2.45)$$

2.3 Modélisation géométrique des manipulateurs continus

La commande de tout système physique passe par une modélisation. Cependant, le choix d'un modèle performant pour les manipulateurs continus reste un véritable défi en raison de leur conception mécanique complexe. La modélisation de tels systèmes peut être effectuée en utilisant les lois de la physique (approches basées sur le modèle), et

les équations géométriques et dynamiques obtenues peuvent être résolues en utilisant des méthodes de résolution numériques. Cependant, les méthodes numériques prennent généralement beaucoup de temps pour une mise en oeuvre en temps réel. En outre, le comportement de courbure non constante des robots continus exige une identification précise des paramètres du robot. Cependant, il n'est pas toujours facile d'identifier certaines classes de robots continus, tel que le CBHA, car il est fabriqué par prototypage rapide. Par conséquent, les approches basées sur l'apprentissage deviennent une alternative et sont de plus en plus utilisées pour la modélisation et le contrôle des manipulateurs continus. Cet intérêt pour les outils d'apprentissage automatique peut être justifié par les caractéristiques et les performances qu'ils offrent. Par exemple, les réseaux neuronaux possèdent de nombreuses propriétés et capacités utiles pour la modélisation de fonctions non linéaires, la mise en correspondance entrée-sortie, l'adaptabilité, la tolérance aux pannes, l'intégration à très grande échelle, etc. ([Haykin 1994]). Grâce à ces caractéristiques, les réseaux de neurones ont été utilisés pour résoudre de nombreux problèmes de régression (voir, par exemple, [Li 2011] et [Ko 2013]) et de classification ([Bischof 1992] et [Specht 1990]), et connaissent un grand succès dans le domaine de l'intelligence artificielle en général, et celui de la robotique en particulier. D'autre part, les machines à vecteurs de support (SVM) sont utilisées depuis longtemps pour résoudre les problèmes de classification de données. La solution au problème de classification est donnée sous la forme d'une combinaison linéaire de certains échantillons de la base de données d'apprentissage, appelés vecteurs de support. Les régresseurs à vecteurs de support (SVR) sont une extension des SVM ; ils ont prouvé leur efficacité dans l'approximation de fonctions non linéaires et la prédiction de séries temporelles ([Chakrabarty 2016], [Vapnik 2013], [Smola 2004]). Les SVR transforment le problème de régression en un problème de programmation quadratique (PQ), de sorte que des solutions globales peuvent être dérivées et que le problème de régression peut être résolu sans les problèmes de minimums locaux rencontrés dans de nombreuses architectures d'apprentissage. Dans ce travail, quatre approches de régression couramment utilisées sont examinées ([Loutfi 2020]) pour mieux aborder la modélisation

géométrique directe et une approche neuronale basée sur apprentissage supervisé distal (DSL) pour le modèle inverse.

2.3.1 Approches basées sur l'apprentissage

Au cours de ces dernières décennies, les approches basées sur l'apprentissage ont été fortement explorées pour fournir une solution à la géométrie directe et inverse des manipulateurs continus. Ces approches sont de plus en plus attrayantes car elles peuvent tolérer une large gamme d'incertitudes et gérer de fortes non-linéarités.

2.3.1.1 Perceptron multicouche (MLP)

Une architecture de réseau de neurones à perceptron multicouche (MLP) composée d'une couche d'entrée, d'une ou plusieurs couches cachées et d'une couche de sortie est illustrée à la Figure 2.11. La couche d'entrée ne reçoit aucun signal des autres couches tandis que la couche de sortie n'envoie aucun signal à une autre couche. Les couches cachées transfèrent des informations de la couche d'entrée à la couche de sortie via un ensemble d'unités interconnectées appelées neurones. Dans cette structure, chaque couche est équipée d'éléments de traitement et chaque élément est entièrement interconnecté par des connexions pondérées aux éléments de la couche suivante. Il est à noter que les neurones de chaque couche travaillent en parallèle pour générer les différentes sorties. La couche d'entrée reçoit les données ; ces données sont traitées dans les couches cachées et affichées dans la couche de sortie. Toutes les connexions entre les différentes couches se voient attribuer un poids qui est ajusté à l'aide d'un algorithme d'apprentissage. Les non-linéarités sont introduites dans l'architecture à l'aide de fonctions d'activation telles que sigmoïde, tangente hyperbolique ou ReLU. À tout neurone i , le niveau d'excitation $v_i(x)$ est calculé comme suit :

$$v_i(x) = \sum_{i=1}^n \beta_i x_i \quad (2.46)$$

Avec β_i le poids de valeur d'entrée correspondant et x_i défini comme l'entrée du neurone i . $y_i = f(v_i)$ est la fonction d'activation appliquée au niveau d'excitation pour déterminer sa sortie. Généralement, la fonction sigmoïde est utilisée comme fonction d'activation pour les couches cachées et est définie comme suit :

$$y_i = \frac{1}{1 + e^{-v_i}} \quad (2.47)$$

Pour trouver l'erreur des variables à la sortie du réseau, le processus d'apprentissage est utilisé comme moyen d'actualiser toutes les connexions pondérées. Il est possible de déterminer l'erreur δy_i qui existe entre la valeur souhaitée et la sortie réelle de chaque neurone i en :

$$\delta \beta_{ij} = -\alpha \frac{\partial \delta y_i}{\partial \beta_{ij}} \quad (2.48)$$

Avec β_{ij} le poids entre l'entrée x_i et le neurone i .

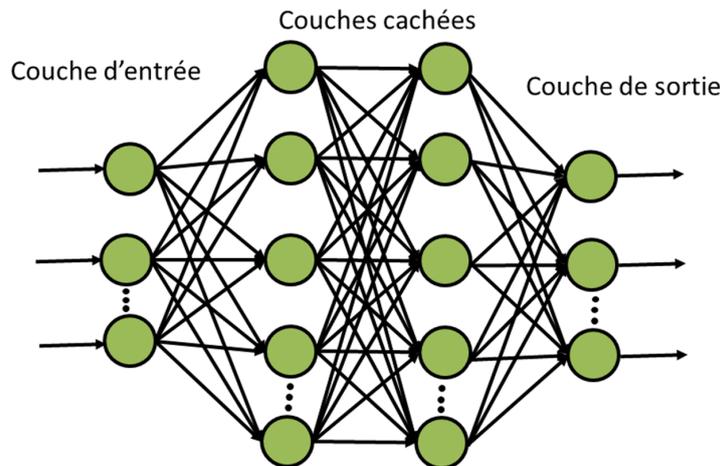


FIGURE 2.11 – Topologie de la MLP

2.3.1.2 Fonction à base radiale (RBF)

Le réseau neuronal à fonction de base radiale (RBF) est basé sur l'apprentissage supervisé. Dans la littérature, les réseaux RBF ont été proposés indépendamment par de

nombreux chercheurs (par exemple dans [Ghorbani 2016] et [Elish 2014]) et constituent une alternative populaire aux MLP. L'architecture du réseau neuronal RBF est similaire à celle du MLP. Elle se compose d'une couche d'entrée, d'une couche intermédiaire et d'une couche de sortie, comme le montre la Figure 2.12. Les neurones d'entrée représentés par les variables d'entrée z_j se propagent vers la couche suivante. Chaque neurone de la couche cachée est associé à une fonction du noyau ϕ_j (généralement une fonction gaussienne) caractérisée par un centre c_j et une largeur σ_j . La principale différence est que la RBF possède une couche cachée qui contient des noeuds appelés unités RBF. Chaque RBF possède deux paramètres clés qui décrivent l'emplacement du centre de la fonction et sa déviation ou largeur. L'unité cachée mesure la distance entre un vecteur de données d'entrée et le centre de ses unités RBF. La fonction RBF atteint son maximum lorsque la distance entre son centre et celui du vecteur de données d'entrée est nulle et diminue progressivement lorsque la distance augmente. Il n'y a qu'une seule couche cachée dans un réseau RBF, et par conséquent seulement deux ensembles de poids pour connecter la couche d'entrée à la couche de sortie. Les poids reliant la couche d'entrée à la couche cachée contiennent les paramètres des fonctions de base. Les poids reliant la couche cachée à la couche de sortie sont utilisés pour former des combinaisons linéaires des activations des fonctions de base (unités cachées) afin de générer les sorties du réseau. Comme les unités cachées sont non linéaires, les sorties de la couche cachée peuvent être combinées linéairement, ce qui permet un traitement rapide des données. C'est pourquoi, dans ce type de réseau, l'information n'est pas totalement distribuée sur l'ensemble des connexions : la distribution de l'information est directement dépendante du nombre de noyaux et de leur position, chaque connexion supportant un élément d'information parfaitement identifiable. On peut donc considérer que les "réseaux RBF" constituent une forme particulière de réseau à information semi distribuée. Les expressions du noyau et de la fonction de sortie sont données respectivement par :

$$\phi_j(\|z - c_j\|) = \exp\left(-\frac{1}{2}\left(\frac{\|z - c_j\|}{\sigma_j}\right)^2\right) \quad (2.49)$$

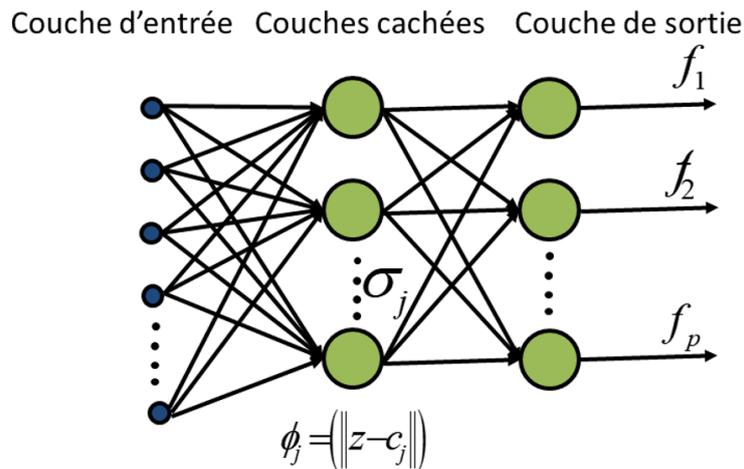


FIGURE 2.12 – Topologie de la RBF

$$f(z) = \sum_{j=1}^p \lambda_j \phi_j(\|z - c_j\|) \quad (2.50)$$

Avec p le nombre de fonctions de base radiales et λ_j le poids associé aux fonctions radiales.

2.3.1.3 Systèmes d'Inférence Neuro-Floue Adaptatifs Co-Actifs (CANFIS)

Les systèmes d'inférence neuro-floue adaptatifs co-actifs (CANFIS) sont une extension des systèmes d'inférence neuro-floue adaptatifs traditionnels (ANFIS) destinés à traiter les systèmes MIMO (multi-sorties et multi-intrants). La topologie de ce système, composée de cinq (5) couches, est illustrée à la Figure 2.13. Chaque couche de cette architecture est destinée à exécuter une étape de système d'inférence floue de type Takagi-Sugeno. L'idée est d'utiliser un concept mathématique dans un cadre de logique floue pour cartographier la relation entre les sorties et les entrées d'un système donné. Dans le cas d'un système à sorties multiples, l'utilisation de la topologie CANFIS devient essentielle, dans la mesure où elle permet de mieux traiter les dépendances entre les différentes sorties du système et rend l'estimation de chaque sortie moins sensible au bruit.

La tâche de chaque couche peut être définie comme suit :

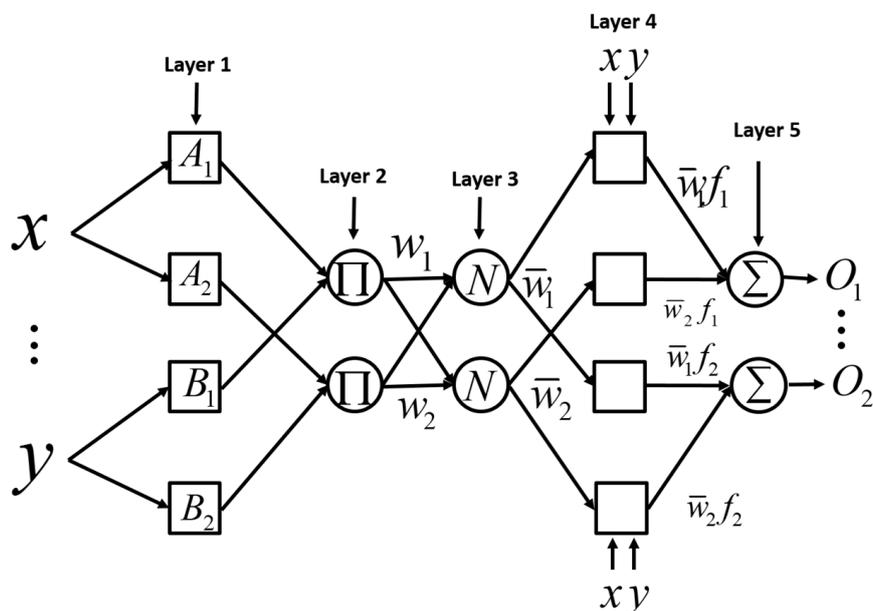


FIGURE 2.13 – Architecture du CANFIS

(i) **Couche 1** (Paramètres de prémisses) :

La première couche, appelée couche de fuzzification, transforme les variables physiques en variables linguistiques appelées degré d'appartenance ou valeur d'appartenance. Chaque noeud de cette couche correspond à un degré d'appartenance à un ensemble flou et représente le degré d'appartenance de l'entrée en question à l'un des ensembles flous. Soit x un élément du vecteur d'entrée, la sortie de la première couche correspondant à cette entrée s'exprime comme suit :

$$O_{1,k} = \mu_{A_i}(x), \text{ for } i = 1, 2 \quad (2.51)$$

où $\mu_{A_i}(x)$ désigne le degré d'appartenance défini dans le cas de la fonction de cloche comme suit :

$$\mu_{A_i}(x) = \frac{1}{1 + \left| \frac{x - c_{ij}}{a_{ij}} \right|^{b_{ij}}} \quad (2.52)$$

où (a_{ij}, b_{ij}, c_{ij}) désignent les paramètres qui contrôlent la forme de la fonction d'appartenance.

(ii) **Couche 2** (Puissance de tir) :

La couche 2 constitue la couche de la puissance de tir. La sortie de chaque noeud de cette couche est le produit de tous les signaux entrants de la première couche :

$$O_{2,k} = w_1 = \mu_{A_i}(x) \times \mu_{B_i}(x), \quad i = 1, 2 \quad (2.53)$$

(iii) **Couche 3** (Normalisation des puissances de tir) :

La troisième couche normalise les puissances de tir. La sortie de chaque noeud est exprimée comme suit :

$$O_{3,k} = \bar{w}_k = \frac{O_{2,k}}{\sum_{k=1}^m O_{2,k}} \quad (2.54)$$

(iv) **Couche 4** (Paramètres conséquents) :

Chaque noeud de cette couche représente un noeud adaptatif et il est associé à une fonction définie comme suit :

$$O_{4,k} = \bar{w}_k f_k = \bar{w}_k (p_k x + q_k y + r_k) \quad (2.55)$$

où, \bar{w}_k est la sortie de la couche 3 et $\{p_k, q_k, r_k\}$ est l'ensemble des paramètres conséquents.

(v) **Couche 5** (Couche de la sortie) :

La cinquième couche, qui est la couche de sortie, permet de calculer la sortie entière en additionnant toutes les données entrantes, s'écrit comme suit :

$$O_{5,k} = \sum_{k=1}^m O_{4,k} \quad (2.56)$$

2.3.1.4 Régression à Vecteur de Support (SVR)

Les SVM proposés par [Vapnik 2013] sont une classe d'algorithmes qui se distinguent par l'utilisation de fonctions à noyau, l'absence de problèmes de minima locaux inhérents à de nombreuses architectures d'apprentissage, la faible densité de la solution et le contrôle de la capacité obtenue en agissant sur la marge, ou le nombre de vecteurs de support. Ils sont initialement développés pour résoudre des problèmes de classification

(voir par exemple [Vapnik 2013] et [Smola 2004]). Cependant, avec l'introduction de la fonction de perte insensible à ε , les SVR ont été étendus pour résoudre des problèmes de régression. Le but de cette approche est de déterminer l'hyperplan optimal représentant le jeu de données. Cet hyperplan doit interpoler les observations avec une certaine marge ε , marge définie par une fonction perte d'insensibilité ε . Cette méthode a plusieurs avantages parmi lesquels, la robustesse contre le bruit et les erreurs, ainsi que la possibilité de son utilisation dans les cas non-linéaires grâce aux fonctions noyaux. Le principal avantage des SVR est qu'ils peuvent être utilisés pour éviter les difficultés liées à l'utilisation de fonctions linéaires dans l'espace caractéristique de haute dimension et que le problème d'optimisation est transformé en programmes quadratiques convexes doubles.

Le problème de régression à une sortie par vecteur de support avec une fonction de perte insensible à ε consiste à trouver une fonction linéaire $f : R^d \rightarrow R$ telle que :

$$f(x) = w\phi(x) + b \quad (2.57)$$

avec $x \in R^d$ le produit scalaire dans R^d et ϕ une transformation non linéaire de R^d vers l'espace de haute dimension R^h .

Étant donné un ensemble d'échantillons indépendants et identiquement distribués, c'est-à-dire $\{(x_i, y_i)\}_{i=1}^n$ où $x \in R^d$ est un vecteur d'entrée et $y \in R$ une sortie observable. Le SVR simple est destiné à résoudre le problème de la recherche des paramètres $w \in R^h$ et $b \in R$ qui minimisent la fonction objectif définie par :

$$\min_{w \in R^h, b \in R} \frac{1}{2} \|w\|^2 + C \sum_{i=0}^n L_\varepsilon(y_i - f(x_i)) \quad (2.58)$$

où $L_\varepsilon = \max\{0, |f(x_i) - y_i| - \varepsilon\}$ est la fonction de perte d'erreur insensible à ε , le paramètre C mesure le compromis entre la capacité de généralisation et la précision dans les données d'apprentissage, et le paramètre ε définit le degré de tolérance aux erreurs. La combinaison linéaire des échantillons d'apprentissage dans l'espace transformé avec des erreurs absolues égales ou supérieures à ε est la solution finale (w et b).

Le problème de la régression multidimensionnelle a été généralisé dans le cas d'un système à sorties multiples. Ce cas a été traité dans [Melinguì 2017] où les sorties de dimension M peuvent être vues comme plusieurs cas à sortie unique où (2.57) est étendu comme suit :

$$f^j(x) = w^j \cdot \phi(x) + b^j, \quad j = 1, 2, \dots, M \quad (2.59)$$

avec w^j et b^j définissant les paramètres du régresseur. Ces problèmes de régression peuvent être résolus en minimisant la fonction objectif suivante :

$$\min_{W, B} \frac{1}{2} \sum_{j=1}^M \|w^j\|^2 + C \sum_{i=1}^n L(\mu_i) \quad (2.60)$$

avec $L(\cdot)$ la fonction de perte étendue de Vapnik ε -insensible basée sur la norme L_2 [Sánchez-Fernández 2004]. Le lecteur intéressé est renvoyé à ([Sánchez-Fernández 2004], [Melinguì 2017]) pour une description détaillée du SVR.

Apprentissage direct supervisé

La modélisation inverse directe traite le problème de l'apprentissage d'un modèle inverse comme un problème classique d'apprentissage supervisé. L'idée est d'observer le comportement entrée/sortie de l'environnement et d'entraîner le modèle inverse directement en inversant les rôles des entrées et des sorties. Les données sont fournies à l'algorithme par échantillonnage dans l'espace d'action et observation des résultats dans l'espace de sensation [Jordan 1992]. Bien que la modélisation inverse directe se soit avérée être une technique viable [Chiddarwar 2010], elle présente deux inconvénients qui limitent son efficacité : Premièrement, lorsque l'environnement est caractérisé par un mappage d'une à plusieurs actions aux sensations, le mappage inverse fera correspondre plus d'une image à un point donné. La manière particulière dont l'incohérence Figure 2.14 est résolue dépend de la forme de la fonction coût ; l'utilisation de la somme des erreurs quadratiques donne une moyenne arithmétique des points qui correspondent au même point final (centroïde).

Si le centroïde n'appartient pas au collecteur des images, les mappages non linéaires entre plusieurs points peuvent produire des images inverses non convexes. Le deuxième inconvénient de la modélisation inverse directe est qu'elle n'est pas "dirigée but". L'algorithme échantillonne dans l'espace d'action sans tenir compte des cibles particulières ou des erreurs dans l'espace de sensation. En d'autres termes, il n'y a pas de moyen direct de trouver une action qui correspond à une sensation particulière souhaitée. Pour surmonter ces deux problèmes, [Jordan 1992] ont proposé une nouvelle architecture de contrôle qu'ils ont appelée apprentissage supervisé distal.

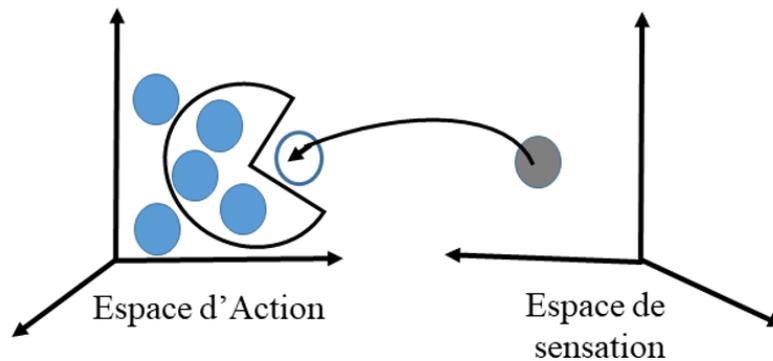


FIGURE 2.14 – Problème de convergence

[Jordan 1992]

Approche d'apprentissage supervisé distal

L'apprentissage supervisé distal consiste à composer un système d'apprentissage tel que représenté sur la Figure 2.15. L'état actuel de l'environnement est $X[n-1]$. Ici, l'état n'aborde pas les questions relatives à la représentation de l'état et à l'estimation de l'état. Les représentations d'état peuvent impliquer des valeurs retardées d'actions précédentes ou des variables d'état internes qui sont induites dans le cadre de la procédure d'apprentissage [[Jordan 1992]]. L'intention est $p[n-1]$, l'action est $u[n-1]$, et le résultat prédit par le modèle direct est $\tilde{y}[n]$. Nous désignerons également le résultat réel par $y[n]$ et le

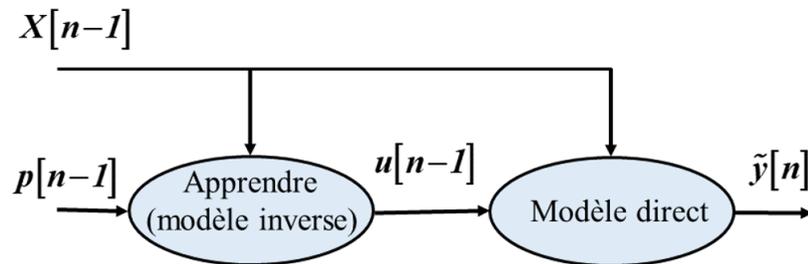


FIGURE 2.15 – Système d'apprentissage composé

[Jordan 1992]

résultat souhaité par $y^*[n]$. Le modèle prévisionnel est un modèle qui prédit le résultat de l'environnement en fonction de l'état actuel et de l'action. Le modèle prévisionnel peut être appris en appliquant des actions et en comparant les résultats réels $y[n]$ aux résultats prédits $\tilde{y}[n]$. L'idée de [Jordan 1992] pour résoudre le problème inverse était d'éviter complètement la modélisation inverse directe. Ils ont utilisé le fait que la composition des modèles inverse et direct doit donner la fonction d'identité. Ils ont proposé de former d'abord un réseau neuronal pour modéliser la géométrie directe, puis d'utiliser ce réseau pour former indirectement le modèle inverse. Le système d'apprentissage composite peut être entraîné par n'importe quel algorithme d'apprentissage supervisé (algorithme de rétro-propagation, règle d'apprentissage delta généralisée,...); cependant, l'algorithme d'apprentissage ne doit pas modifier le modèle direct (maintien des poids directs) pendant l'entraînement du système composite. Le modèle inverse sera finalement appris si les paires d'entrées-sorties d'apprentissage représentent la fonction d'identité. De cette façon, l'effet est que seule une des nombreuses solutions possibles est choisie pour un point cible donnée. Mais, sans informations supplémentaires sur la structure particulière du mappage entrée-action, il n'y a aucun moyen de prédire lequel de l'ensemble éventuellement infini de modèles inverses la procédure trouvera. De plus, une autre vertu de l'approche d'apprentissage distale est la possibilité d'incorporer des contraintes supplémentaires dans la procédure d'apprentissage. Dans ce travail, un terme de pénalité au carré est ajouté à la

fonction objective du réseau neuronal inverse dans un schéma de LIS pour sélectionner un modèle inverse particulier à partir du collecteur de redondance.

2.3.2 Modélisation géométrique du CBHA

Cette sous-section se concentre sur l'utilisation des modèles MPL, RBF, CANFIS et SVR pour l'estimation de la géométrie directe et une approche d'apprentissage supervisé distal des réseaux MLP et RBF pour le modèle inverse. La section commence par une description de la plateforme expérimentale, suivie par la construction de la base de données d'apprentissage.

Description du manipulateur continuums CBHA et formulation du problème

Le robot CBHA est un manipulateur continuum à deux sections inspiré de la trompe de l'éléphant et fabriqué en matériaux polyamide. Il se compose de deux sections de flexion, chacune équipée de trois actionneurs pneumatiques (tubes), d'un axe de poignet et d'une pince souple, comme le montre la Figure 2.16. L'alimentation en pression de chaque tube est contrôlée par un régulateur PID. Les allongements des différents tubes sont fournis par des potentiomètres à six fils placés le long de chaque tube. La déformation élastique du CBHA se traduit par des mouvements avec un nombre virtuellement infini de degrés de liberté. Le manipulateur CBHA est composé de neuf capteurs, six potentiomètres à fil, installés à la surface de chaque tube dorsal flexible pour mesurer leurs allongements réels. Deux capteurs sont utilisés pour la partie rotative, et le dernier pour détecter l'état de la pince. Les propriétés des polyamides et des actionneurs pneumatiques qui le composent en font une plateforme difficile aussi bien pour la modélisation que pour le contrôle. Elle hérite principalement d'un effet de compliance et de mémoire dû aux propriétés des matériaux polyamides et d'un effet d'hystérésis dû à ses actionneurs pneumatiques. La CBHA est la partie manipulatrice montée sur la plate-forme mobile à roues omnidirectionnelles,

l'ensemble du système est appelé RobotinoXT, et elle est illustrée à la Figure 2.16.

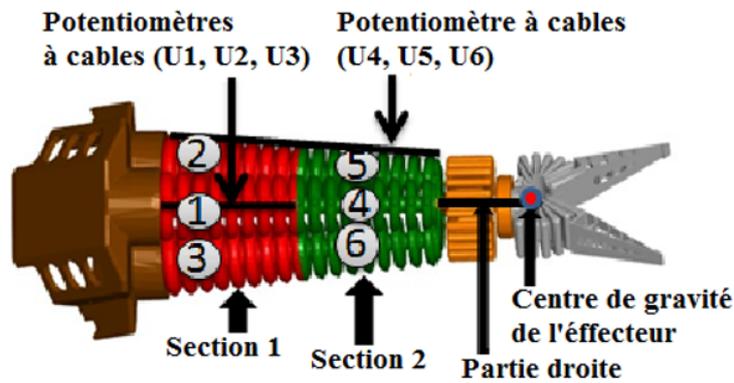


FIGURE 2.16 – Manipulateur continu CBHA

Approche d'apprentissage supervisé distale pour l'IKM de la CBHA

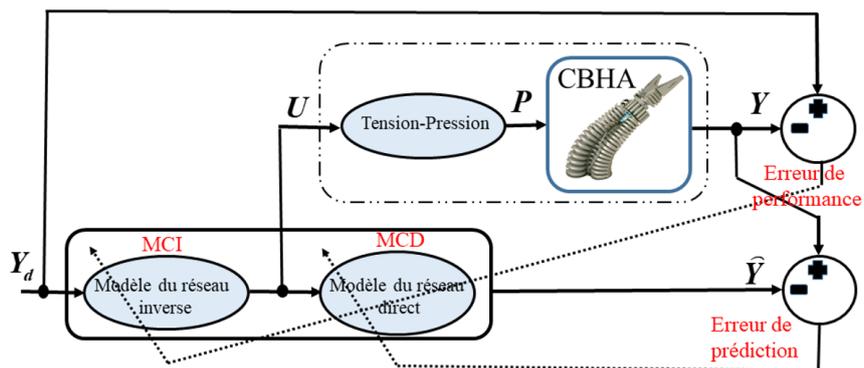


FIGURE 2.17 – Schémas d'apprentissage supervisé distale

En se concentrant sur le CBHA, dans la première phase le réseau neuronal direct (FNN) est d'abord appris pour approximer le modèle géométrique direct du CBHA (pour plus

de détails sur la modélisation géométrique directe du CBHA, nous renvoyons le lecteur à ([Loutfi 2020],[Melinguì 2013b]).

Dans la deuxième phase, une solution inverse particulière est obtenue en plaçant le réseau de neurones inverses (INN) et le FNN en série, et en remplaçant le (système Tension-Pression + CBHA) par le modèle géométrique direct qui avait été entraîné précédemment Figure 2.17. À ce stade, le système d'apprentissage composite peut être entraîné par n'importe quel algorithme d'apprentissage supervisé (algorithme de rétro-propagation dans ce travail); toutefois, l'algorithme d'apprentissage doit être contraint de manière à ne pas modifier le modèle direct (maintien de matrices de poids directs fixes) pendant l'entraînement du système composite. Pour plus de détails sur le DSL, nous renvoyons le lecteur à ([Jordan 1992], [Melinguì 2014b]). $Y_d = [X_d, Y_d, Z_d]^T$ désigne la position souhaitée de l'extrémité de la CBHA, et $Y = [X, Y, Z]^T$ est la position réelle de l'extrémité de la CBHA. $U = [U_1, U_2, \dots, U_6]^T$ est la tension prédite du potentiomètre à fil, et $\hat{Y} = [\hat{X}, \hat{Y}, \hat{Z}]^T$ est la position prédite de l'extrémité du CBHA. Le FNN se compose de 6 entrées (U), et de 3 sorties (\hat{Y}), tandis que le INN se compose de 3 entrées (Y_d), et de 6 sorties (U). L'erreur de prédiction ($Y - \hat{Y}$) et l'erreur de performance ($Y_d - Y$) sont respectivement utilisées pour l'apprentissage des modèles directs et inverses. Pour sélectionner une fonction géométrique inverse particulière, un terme de pénalité au carré est ajouté à la fonction objective du réseau neuronal inverse. La fonction de coût donne les résultats suivants :

$$J = \frac{1}{2} (Y_d - Y)^T (Y_d - Y) + \lambda \frac{1}{2} \|U\|^2 \quad (2.61)$$

Avec $\|\cdot\|$ la norme euclidienne. Il a été démontré que [Wu 2006], plus le coefficient λ est grand, plus U devient petit. Le terme de pénalité λ offre une possibilité de contrôler efficacement la magnitude de U . Par conséquent, il permet de sélectionner une solution inverse particulière. Dans ce travail, une fonction inverse qui minimise la norme euclidienne des tensions fil-potentiomètre est sélectionnée ($\lambda = 0,001$ dans les résultats présentés).

Procédure d'acquisition des données

La performance des différents modèles est réalisée en comparant les positions fournies par chaque modèle avec celles obtenues à partir d'un système de capture de mouvement optitrack ([Nagymáté 2018]). La configuration de l'expérience est représentée sur la Figure 2.18. Le système de capture de mouvement, composé de quatre caméras, peut suivre un mouvement de corps rigide avec une précision de $\pm 0,1$ mm. Les marqueurs réfléchissants sont fixés à l'effecteur terminal du robot, et ce dernier est déplacé dans tout son espace de travail. L'exploration de l'espace de travail du CBHA est effectuée en fournissant une pression dans les six tubes du CBHA. Les vecteurs de pression d'entrée sont obtenus en discrétisant la variable de pression avec un pas de 0,5 bar. Un vecteur d'entrée $[p_1, p_2, \dots, p_6]$, avec $p_i \in [0.0, 0, 0, 0.0, 0.5, 1.0, 1.0, 1.5, 2.0]$ et $i = 1, 2, \dots, 6$ est fourni aux tubes CBHA. Il en résulte une base de données d'échantillons de 15625 et un temps de fonctionnement du robot d'environ 13 heures. Il est à noter que les échantillons sont enregistrés après un équilibre mécanique complet du robot qui prend environ 3 secondes. Ce retard est dû au système de régulation de pression dans les différents tubes car la régulation s'effectue tube par tube. Pour chaque vecteur de pression d'entrée, la position de l'effecteur terminal du robot et les tensions fil-potentiomètre correspondantes sont enregistrées. L'espace de travail généré est montré dans la Figure 2.19.

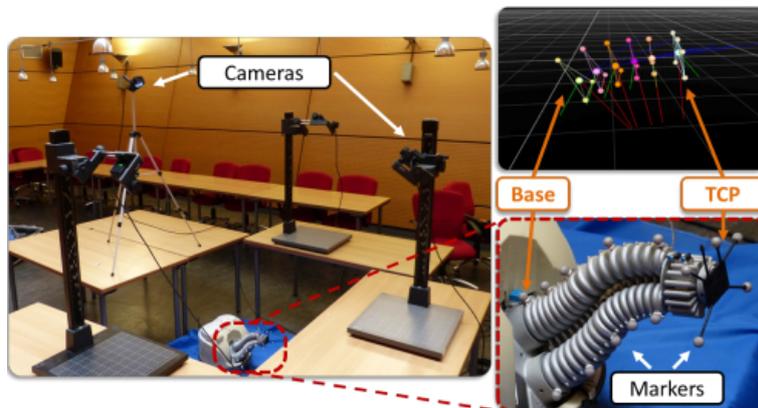


FIGURE 2.18 – Capture de mouvement optitrack

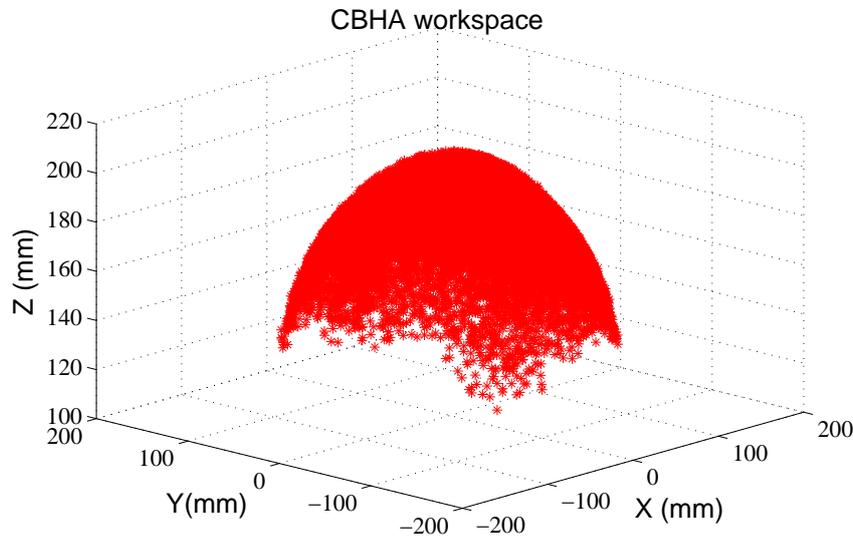


FIGURE 2.19 – Espace de travail du CBHA

Afin de bien généraliser les différents modèles et aussi par souci d'éviter le surajustement, la méthode d'arrêt précoce pour l'apprentissage a été prévue ([Prechelt 1994]). Cette méthode nécessite qu'après quelques époques utilisant l'ensemble d'apprentissage, les matrices de poids du modèle soient fixées, et cette dernière soit utilisée en mode direct à l'aide de l'ensemble de validation. Le processus est répété jusqu'à ce que l'erreur quadratique moyenne (MSE) de l'ensemble de validation atteigne sa valeur minimale. L'entraînement a été arrêté lorsque le critère d'arrêt est atteint ou lorsque le nombre maximum d'époques est atteint. Parmi les modèles qui répondent aux critères d'arrêt, le modèle avec le MSE minimum est sélectionné comme le meilleur. La base de données est normalisée dans la plage $[0.1, 0.9]$ et est divisée au hasard dans le rapport 70 : 15 : 15 : 15 pour l'ensemble d'apprentissage, de validation et de test, respectivement. Les tensions fil-potentiomètre sont considérées comme des entrées et les variables de position et d'orientation sont considérées comme des sorties.

2.4. Commande du mobile manipulateur continuum : Cas du RobotinoXT53

2.4 Commande du mobile manipulateur continuum : Cas du RobotinoXT

Dans cette dernière section, nous nous concentrons sur la modélisation et la commande du mobile manipulateur continuum. Le système est considéré comme deux sous-systèmes, notamment, la plateforme mobile et le manipulateur continuum. Les contrôleurs développés dans les sections précédentes sont coordonnés par un système de coordination neuronal pour assurer la navigation du manipulateur mobile continuum. Ce chapitre traite également le cas d'un manipulateur mobile continuum particulier, en l'occurrence, le RobotinoXT.

2.4.1 Modélisation du mobile manipulateur continuum

Le mobile manipulateur peut être considéré comme une combinaison de deux sous-systèmes, c'est-à-dire la base mobile holonome et le manipulateur continuum.

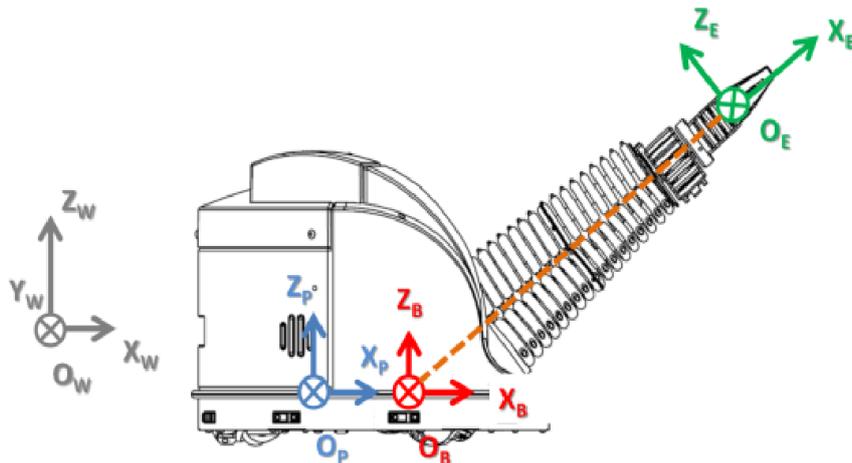


FIGURE 2.20 – Système mobile manipulateur

En observant la Figure 2.20, on remarque quatre repères de coordonnées principales fixés sur le système :

- Le repère du monde $\{O_W, X_W, Y_W, Z_W\}$;
- Le repère de la plate-forme mobile $\{O_P, X_P, Y_P, Z_P\}$, attaché à son centre de gravité (COG) ;
- Le repère de base CBHA $\{O_B, X_B, Y_B, Z_B\}$;
- Et le repère de l'effecteur final $\{O_E, X_E, Y_E, Z_E\}$, fixé au centre de la pince.

La position/orientation de l'effecteur final du manipulateur par rapport au repère du monde OW est donnée par :

$${}^W T_E = {}^B T_E * {}_B^P T * {}_P^W T \quad (2.62)$$

Où $*$ est l'opérateur de multiplication.

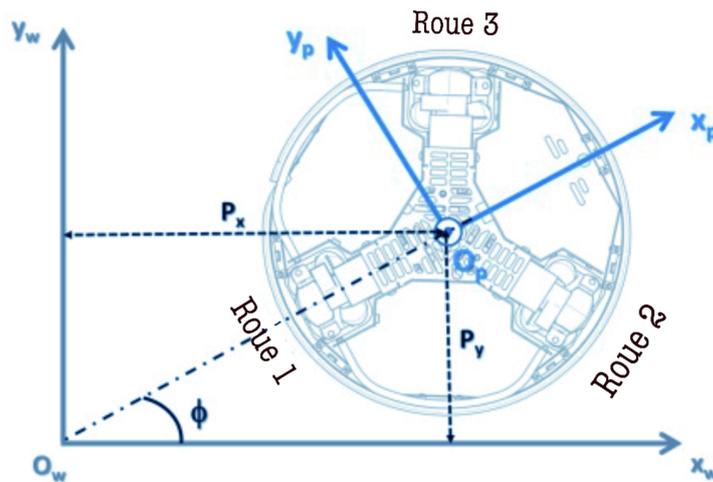


FIGURE 2.21 – Repère du monde sur la plate-forme mobile

Soit l'angle entre les axes X_W et X_P , l'angle de rotation ϕ , comme le montre la Figure 2.21. La matrice de transformation ${}^P W T$ est exprimée par :

$${}^P W T = \begin{bmatrix} \cos(\phi) & -\sin(\phi) & 0 & P_x \\ \sin(\phi) & \cos(\phi) & 0 & P_y \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \quad (2.63)$$

2.4. Commande du mobile manipulateur continuum : Cas du RobotinoXT55

Avec $[P_x, P_y]$ les coordonnées du centre de gravité de la plate-forme mobile. Soit D la distance entre le centre de gravité de la plate-forme mobile et la base du CBHA. La matrice de transformation constante ${}^B_p T$ est donnée par :

$${}^B_p T = \begin{pmatrix} 1 & 0 & 0 & D \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{pmatrix} \quad (2.64)$$

La matrice de transformation ${}^E_B T$ est exprimée par :

$${}^E_B T = NN(W, L) \quad (2.65)$$

2.4.2 Système de coordination neuronal

L'architecture de commande proposée dans ce travail est décrite à la Figure 2.22. Elle comprend un système de coordination neuronal pour la distribution des différentes tâches, un contrôleur APF neuro-floue de type-2 qui assure la navigation de la plate-forme mobile, et un contrôleur neuronal qui prend en entrée les positions et fournit les pressions de commande en sortie.

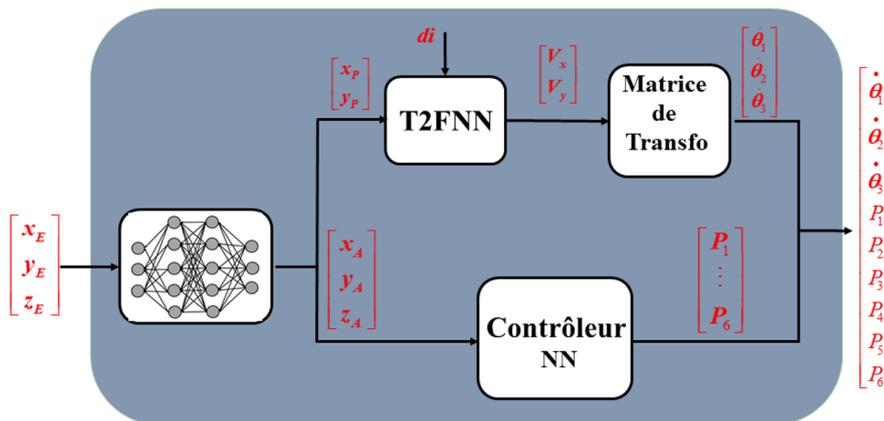


FIGURE 2.22 – Contrôle du mobile manipulateur RobotinoXT

Pour assurer la coordination, un réseau de neurone de type perceptron multicouche (MLP) est utilisé pour approximer les coordonnées le long des axes X_B et Y_B . Il est à noter que les coordonnées approximées n'ont pas besoin d'être précises, car la plate-forme mobile réalise la différence $[x_T - x_A, y_T - y_A, 0]^T$ et le point approximé $[x_A, y_A, z_A]^T$ doit seulement appartenir à l'espace de travail du manipulateur. Les coordonnées Z_T et le signe de la coordonnée Y_T sont considérés comme des entrées pour le réseau neuronal. Comme le montre la Figure 2.23, le réseau de neurone fournit les coordonnées x_A et y_A comme sorties aux deux modules.

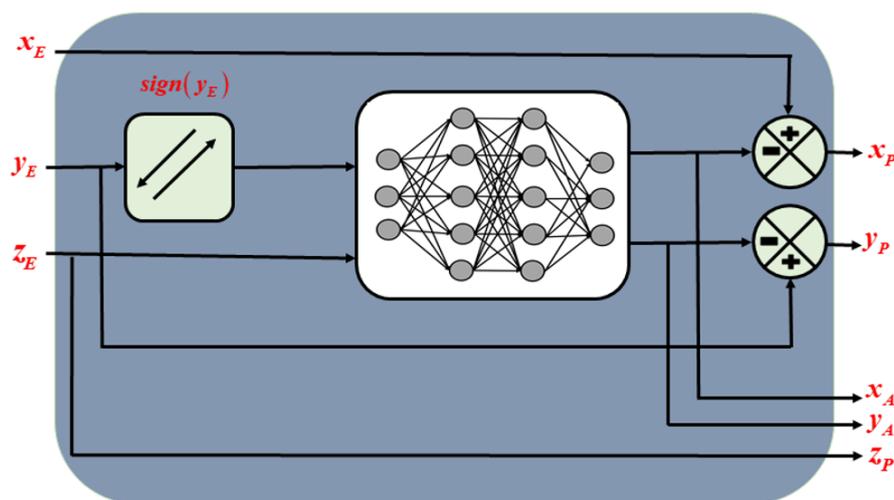


FIGURE 2.23 – Modèle du système de coordination à l'aide de réseau de neurones

2.4.3 Commande du manipulateur CBHA

La commande du CBHA est basée sur un contrôleur neuronal adaptatif proposé dans [Melingu 2014a]. Ce dernier intègre les deux modèles (MGD et MGI) développés dans la section précédente. Les commandes adaptatives basées sur les réseaux neuronaux ont été largement utilisées dans la littérature pour la commande de systèmes non linéaires [Nguyen 1990], [Plett 2003], [Daachi 2012], [Daachi 2006], [Le 2014]. L'algorithme s'exécute en deux étapes. Dans la première étape, un réseau neuronal (un émulateur) apprend

à reproduire les comportements du système. Dans la deuxième étape, un contrôleur, un autre réseau neuronal, apprend à contrôler l'émulateur. La complexité du CBHA réside dans la difficulté d'obtenir un émulateur précis, parce que ce dernier imite le comportement que le système avait au moment de la construction de la base de données d'apprentissage. Cependant, du fait que les comportements ne sont pas stationnaires, ce modèle change au cours du temps.

2.5 Conclusion

En résumé, dans ce chapitre, nous avons proposé pour la plateforme mobile, une approche intelligente basée sur la combinaison des approches de l'APF, de la FL et de réseau de neurones dans une même architecture de commande. Cette hybridation APF-neuro-floue nous a permis de simplifier la conception du contrôleur du robot mobile, de réduire la taille de la base de règles floues et le temps de calcul du contrôleur. Le manipulateur CBHA quand à lui a été modélisé en utilisant les approches basées sur l'apprentissage. Enfin, quant à la commande de l'ensemble du manipulateur mobile continuum, notre système de contrôle est basé sur deux contrôleurs, coordonnés par un système de réseaux de neurones. Ils sont implémentés dans une même architecture de commande.

Résultats des simulations, expériences et discussions

3.1 Introduction

Dans ce chapitre, les résultats des simulations et des tests expérimentaux sont présentés. La première partie présente les résultats obtenus pour la navigation autonome de la plateforme mobile appelée Robotino, la deuxième partie présente ceux du manipulateur continuum appelé CBHA ; et enfin, la dernière partie s'intéresse à ceux du manipulateur mobile continuum. Dans chaque partie, une analyse et une discussion des résultats obtenus sont également présentées.

3.2 Navigation autonome des robots mobiles : cas du Robotino

Dans cette section, quatre scénarios sont considérés pour valider le contrôleur IT2FNN proposé, trois dans des environnements statiques, et un dans un environnement dynamique. Dans chaque scénario, il y a quatre ou cinq obstacles que le robot doit éviter tout en se déplaçant vers la position cible. Les obstacles sont placés de manière à ce que la trajectoire du robot mobile les traverse et rencontre ensuite d'autres obstacles. Les scénarios sont choisis pour évaluer si le contrôleur IT2FNN proposé peut émuler le contrôleur

IT2FL proposé dans des travaux antérieurs [Melinguì 2014d] tout en étant efficace en termes de temps de calcul. Le premier scénario tente d'évaluer les stratégies de suivi de couloir, d'évitement d'obstacle et d'atteinte de cible en présence d'obstacles à géométrie régulière. Le second scénario permet d'évaluer si les problèmes de minima locaux sont effectivement éliminés. Un obstacle est placé juste après le point cible. Les stratégies de suivi du couloir, d'évitement de l'obstacle et d'atteinte de la cible sont également évaluées. Le troisième scénario tente d'évaluer les stratégies de suivi de couloir, de suivi de mur, d'évitement d'obstacle et d'atteinte de cible, cette fois en présence d'obstacles à géométrie irrégulière. Enfin, le quatrième scénario valide le contrôleur proposé dans un environnement dynamique, où trois obstacles mobiles sont impliqués. Pour chaque scénario, le contrôleur proposé est comparé au contrôleur IT2FL, proposé par [Melinguì 2014d], afin d'évaluer si le contrôleur proposé atteint ou non les objectifs attendus (c'est-à-dire à la fois l'émulation IT2FL et l'efficacité computationnelle).

Les résultats expérimentaux dans le cas du premier scénario qui évalue les stratégies de suivi de couloir, d'évitement d'obstacle et d'atteinte de la cible en présence d'obstacles à géométrie régulière sont présentés sur les figures 3.1, 3.2, et 3.3. La figure 3.1 montre que l'IT2FL et l'IT2FNN peuvent atteindre la cible et éviter les obstacles. Le robot suit un chemin plus court et plus lisse en utilisant IT2FNN plutôt que IT2FL. Cela implique que l'IT2FNN peut atteindre la position cible plus rapidement que l'IT2FL. Cependant, l'IT2FNN passe à côté de certains obstacles. La figure 3.2 montre que la vitesse longitudinale de l'IT2FNN est supérieure à celle de l'IT2FL. De plus, l'IT2FL a un comportement plus oscillant, principalement lors de l'approche de la position cible. Dans la figure 3.3, nous observons presque les mêmes plages de vitesse pour les deux contrôleurs, avec IT2FL ayant une vitesse plus stable que IT2FNN. D'après ce qui précède, nous remarquons que l'IT2FL tente d'appliquer fidèlement les différentes règles floues, ce qui entraîne parfois des variations abruptes dans sa trajectoire. En revanche, l'IT2FNN, en plus de les appliquer, effectue une certaine interpolation entre ces règles (une sorte de généralisation). Le tableau I montre les distances parcourues et le temps passé par chaque

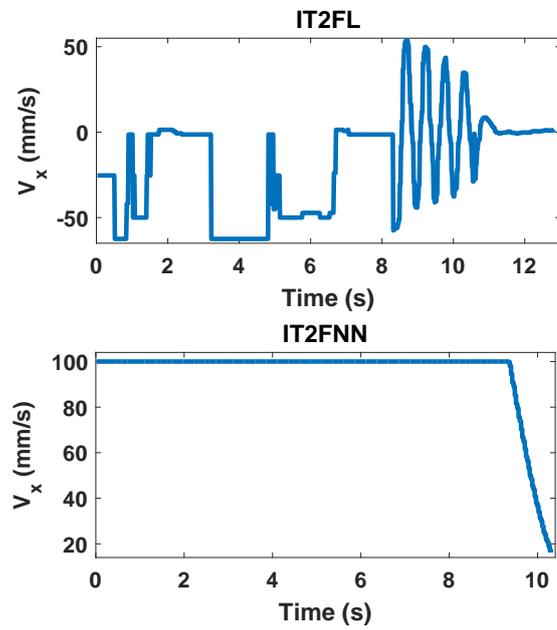


FIGURE 3.2 – Premier scénario : Vitesses longitudinales fournies par les contrôleurs IT2FL et IT2FNN.

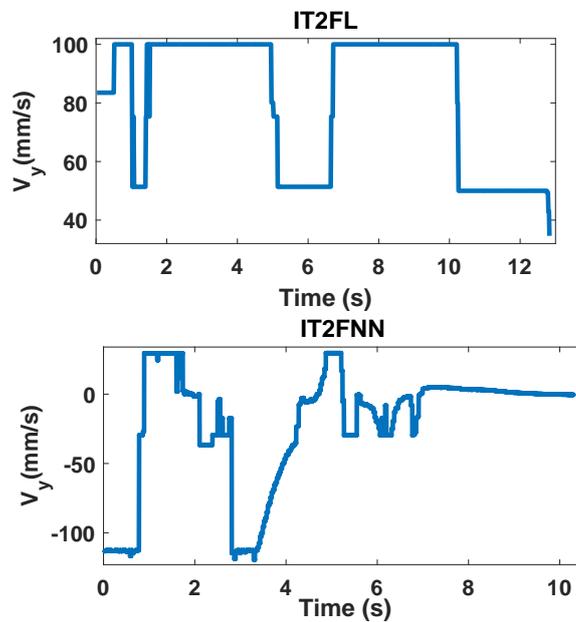


FIGURE 3.3 – Premier scénario : Vitesses latérales fournies par les contrôleurs IT2FL et IT2FNN.

TABLE 3.1 – Comparaisons des résultats expérimentaux

Exp	Déplacement		Trajet	Temps (s)	
	IT2FL	distances(m)	IT2FL	IT2FNN	IT2FNN
(0,0)–(300,0)	3.35	3.13	14.04		10.30
(0,0)–(250,0)	2.35	2.53	10.85		9.962
(0,0)–(300,-50)	3.38	3.16	14.23		10.89
(0,0)–(300,0)	4.07	4.01	29.73		24.51

vitesses latérales est la même que dans le scénario précédent, et les deux contrôleurs ont presque les mêmes comportements. Nous observons un comportement similaire à celui du premier scénario.

Le troisième scénario évalue les stratégies de suivi de couloir, de suivi de mur, d'évitement d'obstacle et d'atteinte de la cible, cette fois avec des obstacles de forme irrégulière. Les figures 3.7, 3.8, et 3.9 montrent les résultats expérimentaux obtenus. Une fois de plus, la figure 3.7 montre que les systèmes IT2FL et IT2FNN peuvent atteindre la cible et éviter les obstacles, quelle que soit la géométrie des obstacles rencontrés. Le robot suit le chemin le plus court et le plus lisse en utilisant IT2FNN plutôt que IT2FL. L'IT2FL présente des comportements oscillants principalement à proximité de la position cible. La plage de vitesses est la même pour les deux contrôleurs, et les vitesses de IT2FNN sont plus stables que celles de IT2FL, comme le montrent les figures 3.8 et 3.9.

Dans le quatrième scénario, le contrôleur proposé est validé dans un environnement dynamique. Ce dernier comprend trois obstacles mobiles de forme circulaire, une configuration de couloir et une configuration d'obstacles en forme de U formée par le couloir et le premier obstacle mobile. La réactivité du contrôleur proposé face aux obstacles mobiles et sa capacité d'adaptation aux événements imprévus sont évaluées. Les résultats expérimentaux obtenus sont donnés dans les figures 3.10, 3.11, et 3.12. La figure 3.10 montre que IT2FL et IT2FNN peuvent atteindre une cible donnée dans un environnement dynamique

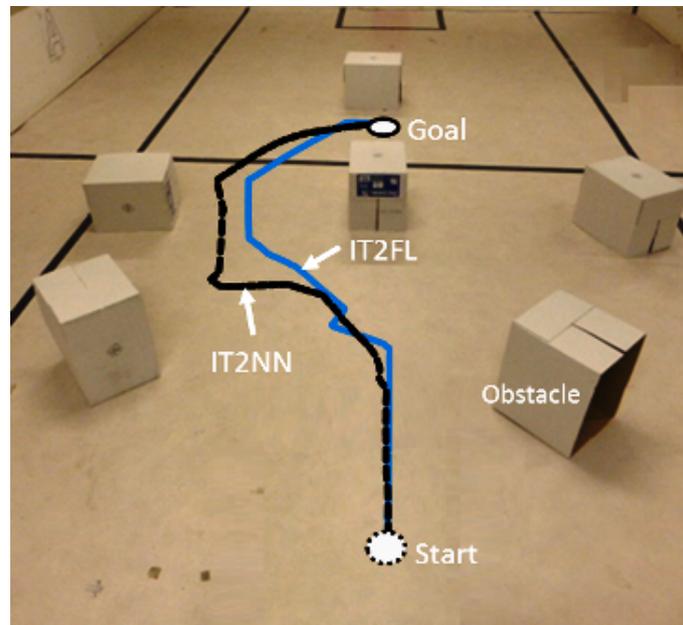


FIGURE 3.4 – Deuxième scénario : les stratégies de suivi de couloir, d'évitement d'obstacles et d'atteinte de la cible dans des situations de présence de minima locaux.

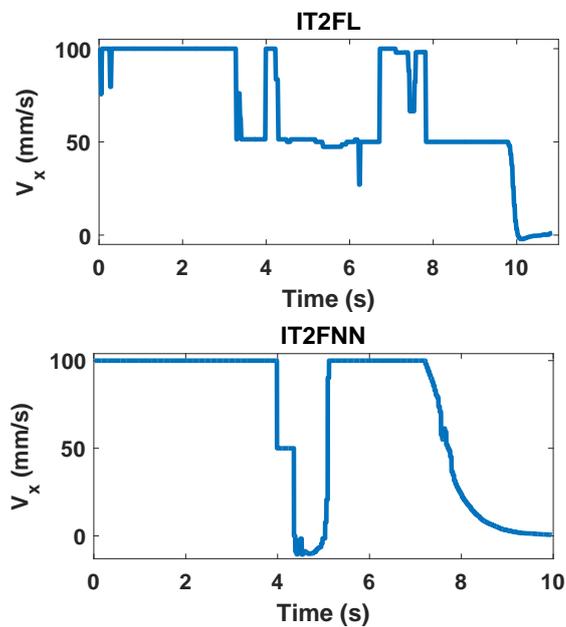


FIGURE 3.5 – Deuxième scénario : Vitesses longitudinales fournies par les contrôleurs IT2FL et IT2FNN.

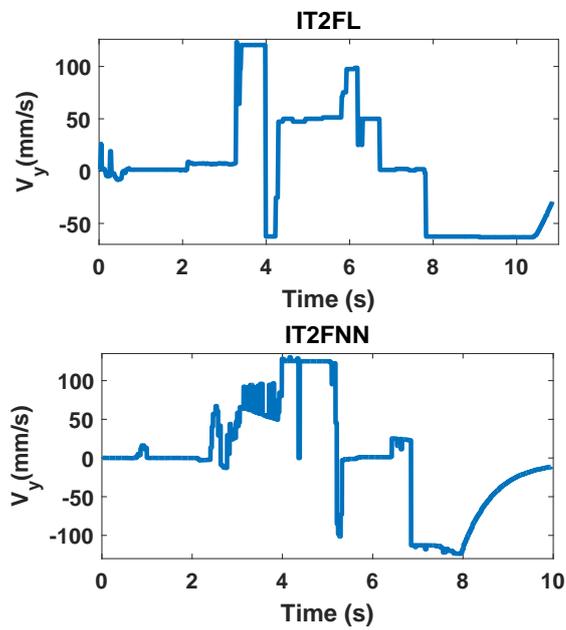


FIGURE 3.6 – Deuxième scénario : Vitesses latérales fournies par les contrôleurs IT2FL et IT2FNN.

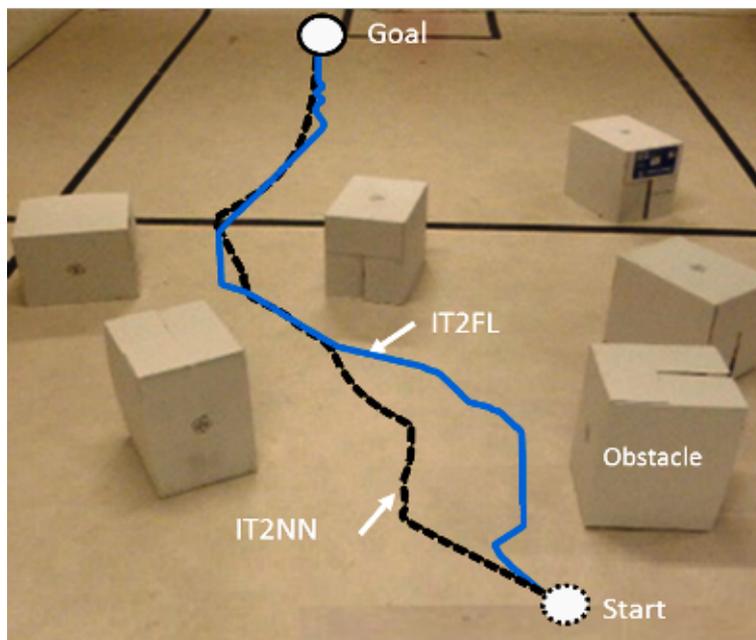


FIGURE 3.7 – Troisième scénario : les stratégies de suivi de couloir, d'évitement d'obstacles et d'atteinte de la cible en présence d'obstacles de forme irrégulière.

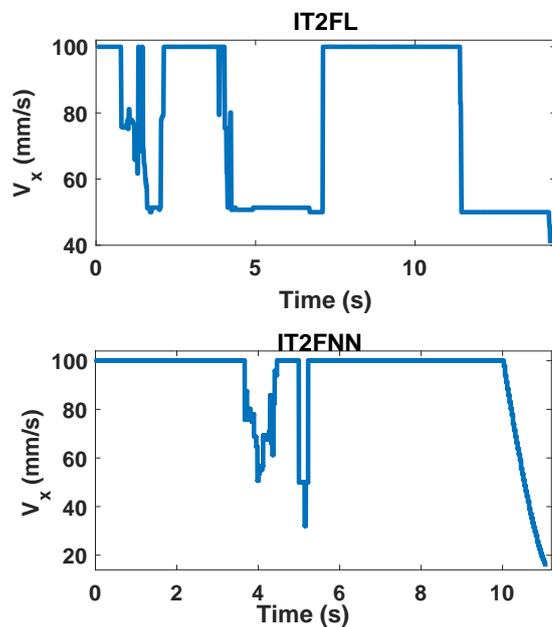


FIGURE 3.8 – Troisième scénario : Vitesses longitudinales fournies par les contrôleurs IT2FL et IT2FNN.

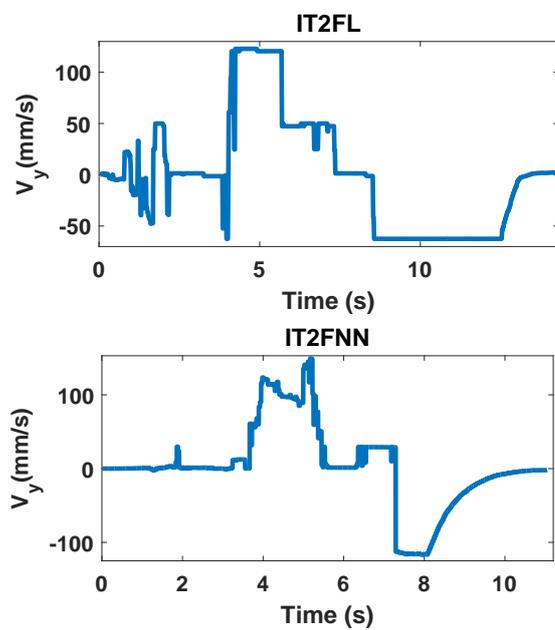


FIGURE 3.9 – Troisième scénario : Vitesses latérales fournies par les contrôleurs IT2FL et IT2FNN.

tout en traitant les obstacles mobiles et les événements imprévus. Les trajectoires réalisées par le robot sont presque identiques avec quelques oscillations au début de la trajectoire de l'IT2FL. Le contrôleur IT2FNN reste plus rapide que IT2FL, comme dans les trois scénarios précédents. Les vitesses générées par le contrôleur IT2FL sont environ deux fois plus élevées que celles du contrôleur IT2FNN. La vitesse longitudinale de l'IT2FNN est également plus stable que celle de l'IT2FL, comme le montrent les figures 3.11 et 3.12.

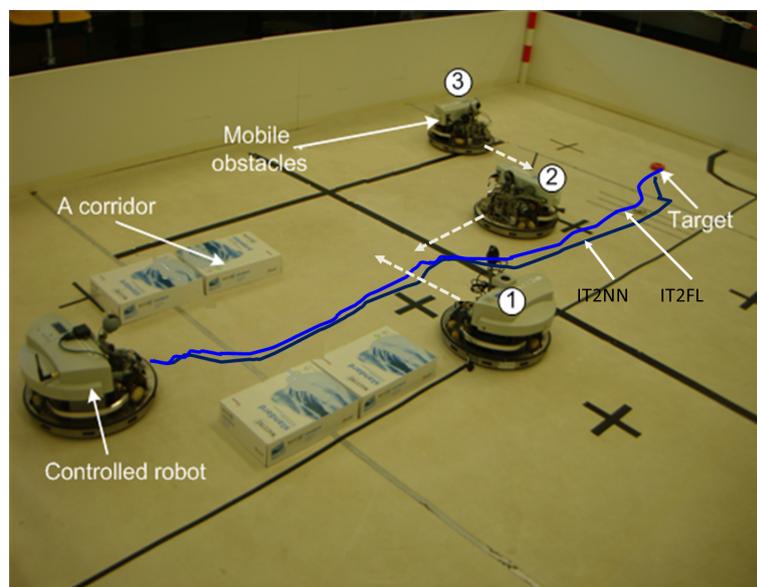


FIGURE 3.10 – Le quatrième scénario : la navigation dans des environnements dynamiques.

En résumé, l'analyse des résultats obtenus dans les trois scénarios révèle que les deux contrôleurs peuvent atteindre une position cible donnée tout en évitant les obstacles rencontrés. Nous pouvons également dire que les objectifs attendus ont été atteints, c'est-à-dire la conception d'un contrôleur robuste et efficace en termes de temps de calcul. De plus, au delà de l'atteinte des objectifs initialement définis, nous avons remarqué que la capacité de généralisation de l'IT2FNN lui permet de gérer les phases de transition entre les différentes règles floues. Une sorte d'interpolation est ainsi réalisée entre les différentes règles floues. Nous avons également remarqué que IT2FL tente d'implémenter fidèlement

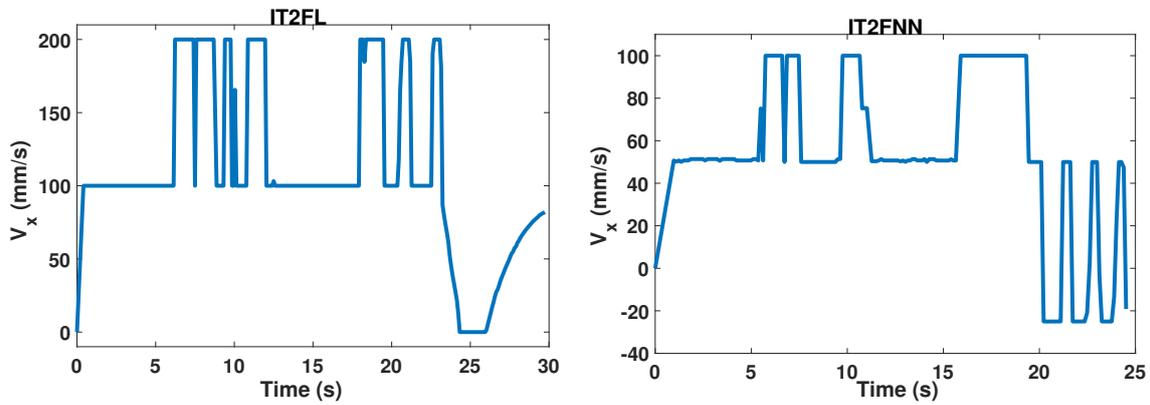


FIGURE 3.11 – Le quatrième scénario : Vitesses longitudinales fournies par les contrôleurs IT2FL et IT2FNN.

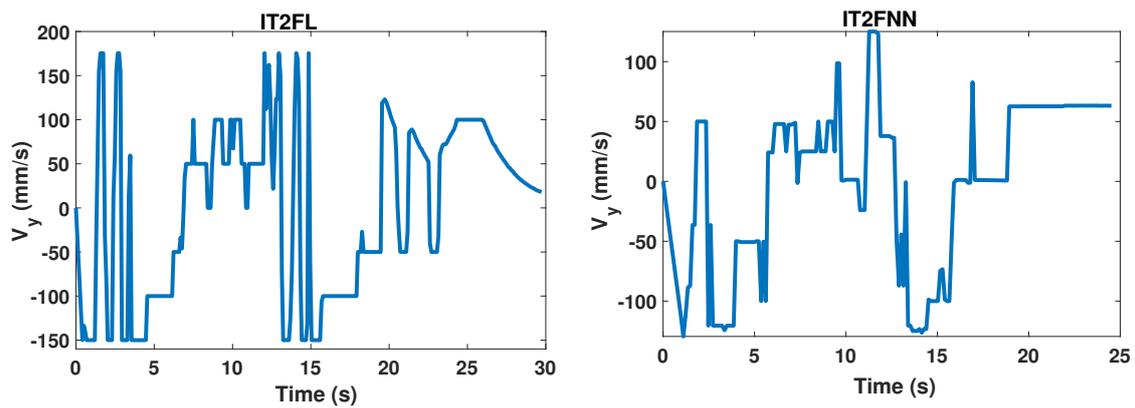


FIGURE 3.12 – Le quatrième scénario : Vitesses latérales fournies par les contrôleurs IT2FL et IT2FNN

3.3. Modélisation géométrique du manipulateur continuum : Cas du CBHA9

les différentes règles floues au détriment de la forme de la trajectoire générée (présence d'oscillations et de variations brusques). Enfin, l'interpolation entre les différentes règles floues effectuée par l'IT2FNN entraîne parfois le robot à se rapprocher trop près de certains obstacles.

3.3 Modélisation géométrique du manipulateur continuum : Cas du CBHA

En suivant la même stratégie, dans cette section nous présentons ([Loutfi 2020]) notre étude comparative expérimentale et structurelle des modèles d'apprentissage couramment utilisés, à savoir le perceptron multicouche (MLP), les fonctions à base radiale (RBF), la régression par vecteur de support (SVR) et le système d'inférence neuro-flou adaptatif coactif (CANFIS) sur le modèle géométrique direct de la CBHA et un modèle géométrique inverse basé sur une approche d'apprentissage supervisé distale dont des réseaux neuronaux à perceptron multicouche (MLP) sont intégrés.

3.3.1 Processus de validation

Le processus d'apprentissage de chaque modèle est effectué dans le logiciel MATLAB à l'aide d'un processeur Intel(R) Core(TM) *i7 – 7500U* à $2,70GHz$ et $2,90 GHz$.

Pour le cas du réseau MLP, la configuration adoptée est constituée de deux couches cachées de 28 neurones chacune, avec une fonction d'activation sigmoïde au niveau des couches cachées et une fonction d'activation linéaire dans la couche de sortie, produisant des performances satisfaisantes.

En ce qui concerne le réseau RBF, nous avons utilisé 60 neurones dans la couche cachée et des fonctions d'activations gaussiennes d'une largeur de 0,5, ce réseau a donné des performances satisfaisantes.

Concernant le modèle CANFIS, il est composé de 15 fonctions d'appartenance de type gaussiennes pour chaque variable d'entrée et de 15 règles de type Takagi et Sugeno. Il a

atteint des performances satisfaisantes.

Enfin dans le cas du modèle SVR, le regroupement flou des C-moyens a été utilisé pour l'initialisation des centres de fonctions gaussiennes et une règle d'apprentissage hybride ([Jang 1993]) qui combine la méthode du gradient et l'estimation des moindres carrés (LSE) pour identifier les paramètres du réseau a été utilisée. Les paramètres suivants : $C = 4000$, $\varepsilon = 0,4e^{-6}$ et $\sigma = 1,2$ ont permis d'obtenir des performances très satisfaisantes pour le cas SVR. Où C est une constante fixe qui contrôle le compromis entre l'erreur d'apprentissage et le terme de régularisation, ε est l'erreur insensible, et σ est l'écart type des fonctions noyau gaussiennes. Une sélection de variables à rebours par suppression de blocs est mise en oeuvre pour la sélection des variables.

3.3.2 Resultats des simulations et discussions

Le tableau 3.2 présente les résultats obtenus pour chaque modèle, l'erreur quadratique moyenne (EQM), le temps d'apprentissage, ainsi que les paramètres de chaque modèle sont fournis. On note des MSE de 5.3481×10^{-5} , 7.0691×10^{-5} , 2.3383×10^{-5} , et 4.5232×10^{-6} pour MLP, RBF, CANFIS, et SVR, respectivement. Les temps d'apprentissage sont approximativement de $01h45mn$, $01h15mn$, $02h30mn$, et $03h41mn$ pour MLP, RBF, CANFIS, et SVR, respectivement. Les erreurs euclidiennes résultant de la comparaison avec les poses fournies par le système de capture de mouvement sont répertoriées dans le tableau 3.3 et le tableau 3.4. Les erreurs cartésiennes absolues maximales de $4,164mm$, $6,039$, $2,79$ et $3,638$ sont constatées pour les modèles MLP, RBF, CANFIS et SVR, respectivement, tandis que les erreurs d'orientation maximales sont de $11,604$, $15,148$, $5,667$ et $6,976$, respectivement. Les coefficients de corrélation représentant le degré d'association linéaire entre les valeurs estimées et les variables des valeurs mesurées sont reportés dans le tableau 3.5 et le tableau 3.6. Un coefficient d'environ $0,99$ est obtenu pour chaque modèle.

Globalement, les performances obtenues par chaque modèle d'apprentissage sont satisfaisantes en termes de suivi de la position de l'effecteur final du robot CBHA. Chaque

3.3. Modélisation géométrique du manipulateur continuum : Cas du CBH~~A~~1

TABLE 3.2 – Resultats des modèles MLP, RBF, SVR et CANFIS.

Modèles	Paramètres	MSE	temps d'apprentissage
MLP	28 neurons	$5.3481e^{-5}$	01h45
RBF	60 neurons, $\sigma = 0.5$	$7.0691e^{-5}$	01h15
CANFIS	15 If-then rules	$2.3383e^{-5}$	02h30
SVR	$C = 4000$, $\sigma = 1.2$	$4.5232e^{-6}$	03h41

TABLE 3.3 – Valeur absolue des erreurs euclidiennes suivant la position

Modèles	$X(mm)$	$Y(mm)$	$Z(mm)$
MLP	2.679	3.219	4.164
RBF	5.960	5.302	6.039
CANFIS	2.033	3.116	3.638
SVR	1.406	1.191	2.796

TABLE 3.4 – Valeur absolue des erreurs euclidiennes suivant l'orientation

Modèles	ψ°	θ°	ϕ°
MLP	11.604	2.224	10.876
RBF	15.148	3.183	12.096
CANFIS	6.852	1.746	6.976
SVR	4.870	0.856	5.667

TABLE 3.5 – Coefficient de corrélation linéaire selon la Position R .

Modèles	$X(mm)$	$Y(mm)$	$Z(mm)$
MLP	0.99994	0.99991	0.99937
RBF	0.99971	0.99945	0.998
CANFIS	0.99995	0.99995	0.99941
SVR	0.99998	0.99997	0.99974

TABLE 3.6 – Coefficient de corrélation linéaire selon l'Orientation R

Modèles	ψ°	θ°	ϕ°
MLP	0.99967	0.99982	0.99899
RBF	0.9941	0.9998	0.99812
CANFIS	0.99883	0.99988	0.99908
SVR	0.9994	0.99995	0.99969

3.3. Modélisation géométrique du manipulateur continuum : Cas du CBHA

modèle peut suivre l'effecteur final du robot CBHA avec une erreur de position maximale de 6mm et une erreur d'orientation maximale de 15°. Le coefficient R est également supérieur à 0,99 démontrant la forte relation linéaire entre les valeurs estimées et les valeurs mesurées. Nous pouvons conclure que les approches basées sur l'apprentissage peuvent modéliser efficacement la relation hautement non linéaire entre l'espace de configuration et l'espace de travail du robot CBHA, sans les problèmes de singularités géométriques dans les positions étirées, généralement rencontrés dans les approches basées sur des modèles.

En ce qui concerne la performance de chaque modèle, le modèle SVR obtient la meilleure performance, suivi par CANFIS, MLP et RBF, respectivement. Cependant, RBF a le meilleur temps de convergence, suivi par MLP, CANFIS, et SVR, respectivement.

Outre les avantages communs aux différents modèles, en particulier les capacités de meilleure approximation et de généralisation, chacun de ces modèles présente des avantages et des inconvénients. Les RBFs ont l'avantage, par exemple, d'avoir une architecture de réseau simple. Ils peuvent modéliser toute fonction non linéaire en utilisant une seule couche cachée. Cela élimine certaines décisions de conception telles que le nombre de couches cachées. Le temps de convergence est très rapide ; mais aussi la transformation linéaire de la couche de sortie peut être optimisée en utilisant des techniques de modélisation linéaire traditionnelles, qui sont rapides et ne souffrent pas de problèmes tels que les minima locaux rencontrés dans les techniques d'apprentissage MLP. Cependant, les capacités de généralisation des RBF sont inférieures à celles des MLP car l'approximation est locale et l'extrapolation au-delà des données connues est faible. Les SVR offrent une solution globale car le problème de régression est transformé en un problème d'optimisation convexe. Il s'agit d'un réel avantage par rapport aux réseaux neuronaux, qui sont confrontés à des problèmes de minimums locaux et qui, pour cette raison, peuvent ne pas être robustes sur l'ensemble de la base d'apprentissage. Cependant, un inconvénient courant des méthodes non paramétriques telles que les SVR est l'absence de transparence des résultats. L'inconvénient de ce modèle est qu'il n'est pas adapté aux grands ensembles

de données. CANFIS présente l'avantage que les paramètres du système d'inférence sont codés comme des poids dans le réseau neuronal et peuvent donc être optimisés en utilisant des méthodes d'apprentissage puissantes. Cependant, le temps de calcul est très élevé, et le processus d'apprentissage nécessite la maîtrise des concepts de base de la logique floue et des réseaux neuronaux.

En résumé, selon les résultats des différents histogrammes nous pouvons dire que les approches basées sur l'apprentissage sont une bonne alternative pour la modélisation géométrique directe des robots continus. Quel que soit le modèle d'apprentissage utilisé, les performances obtenues sont satisfaisantes. Cependant, les modèles RBF et SVR sont recommandés pour les petites bases de données et les modèles MLP et CANFIS pour les grandes bases de données.

Concernant le modèle géométrique inverse, il a été validé sur l'ensemble de données de test. L'évaluation des régresseurs formés en termes d'EQM (erreur quadratique moyenne) sur les échantillons de test a donné les valeurs indiquées dans le tableau 3.7. La première colonne présente les topologies de réseau neuronal, tandis que la deuxième colonne indique le nombre de neurones cachés pour chaque topologie. La troisième colonne décrit le MSE obtenu dans l'ensemble de validation, et la dernière colonne présente le modèle approximé. L'EQM est de l'ordre de 10^{-4} pour le modèle géométrique inverse.

TABLE 3.7 – Résultats obtenus par chaque modèle de réseau neuronal sur les données de test

Topologies de Réseau de neurones	Neurones	MSE	
MLP (2 couches cachées)	16	$1.160.10^{-4}$	MGI
RBF ($\sigma = 0.2210$)	74	$4.172.10^{-4}$	

En outre, un réseau MLP est implémenté en temps réel (Figure 3.16). L'installation consiste en un système de prototype rapide intitulé « DSpace » qui est une plateforme appropriée pour l'implémentation en temps réel basée sur le logiciel MATLAB/Simulink.

3.3. Modélisation géométrique du manipulateur continu : Cas du CBH75

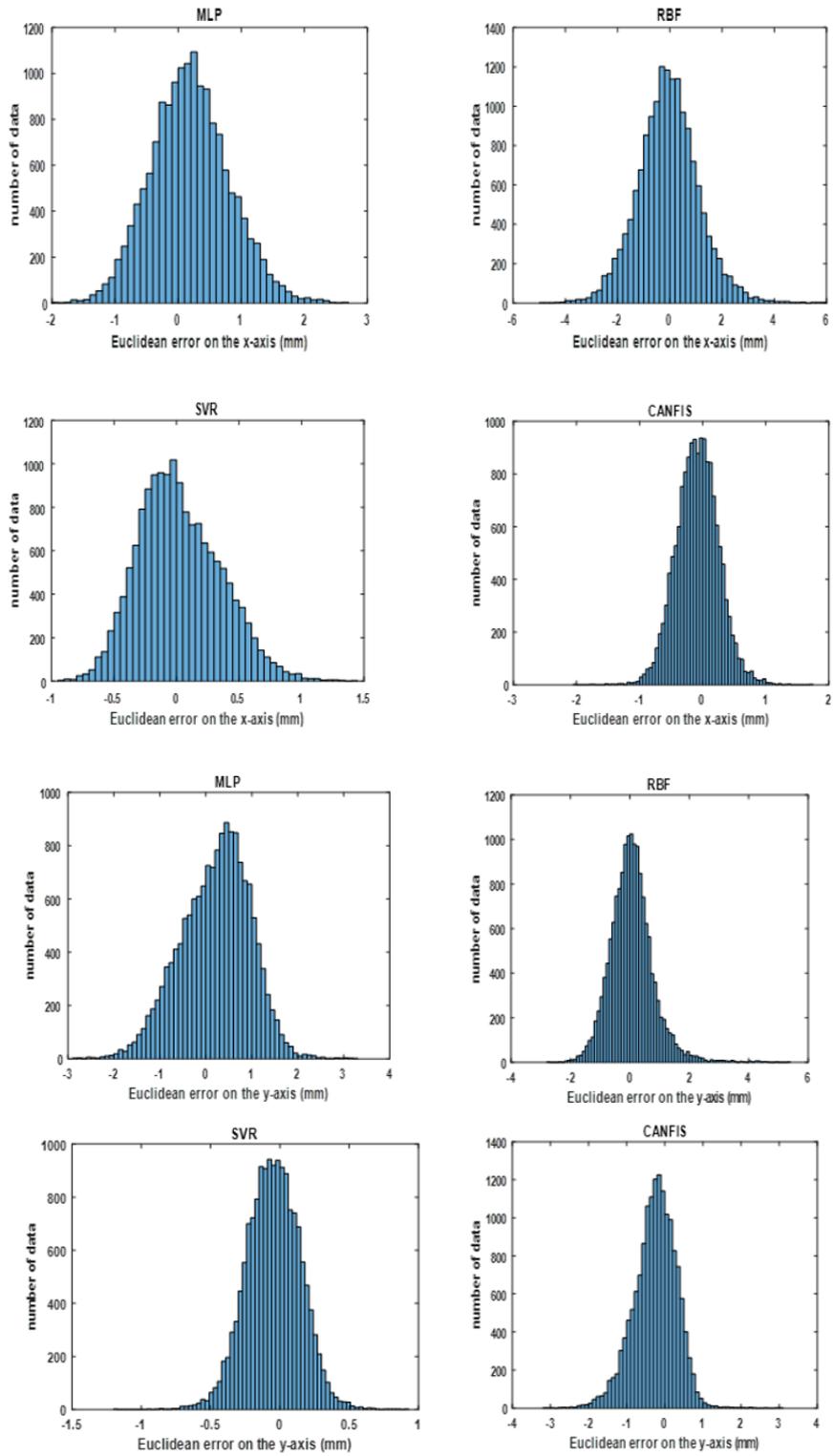
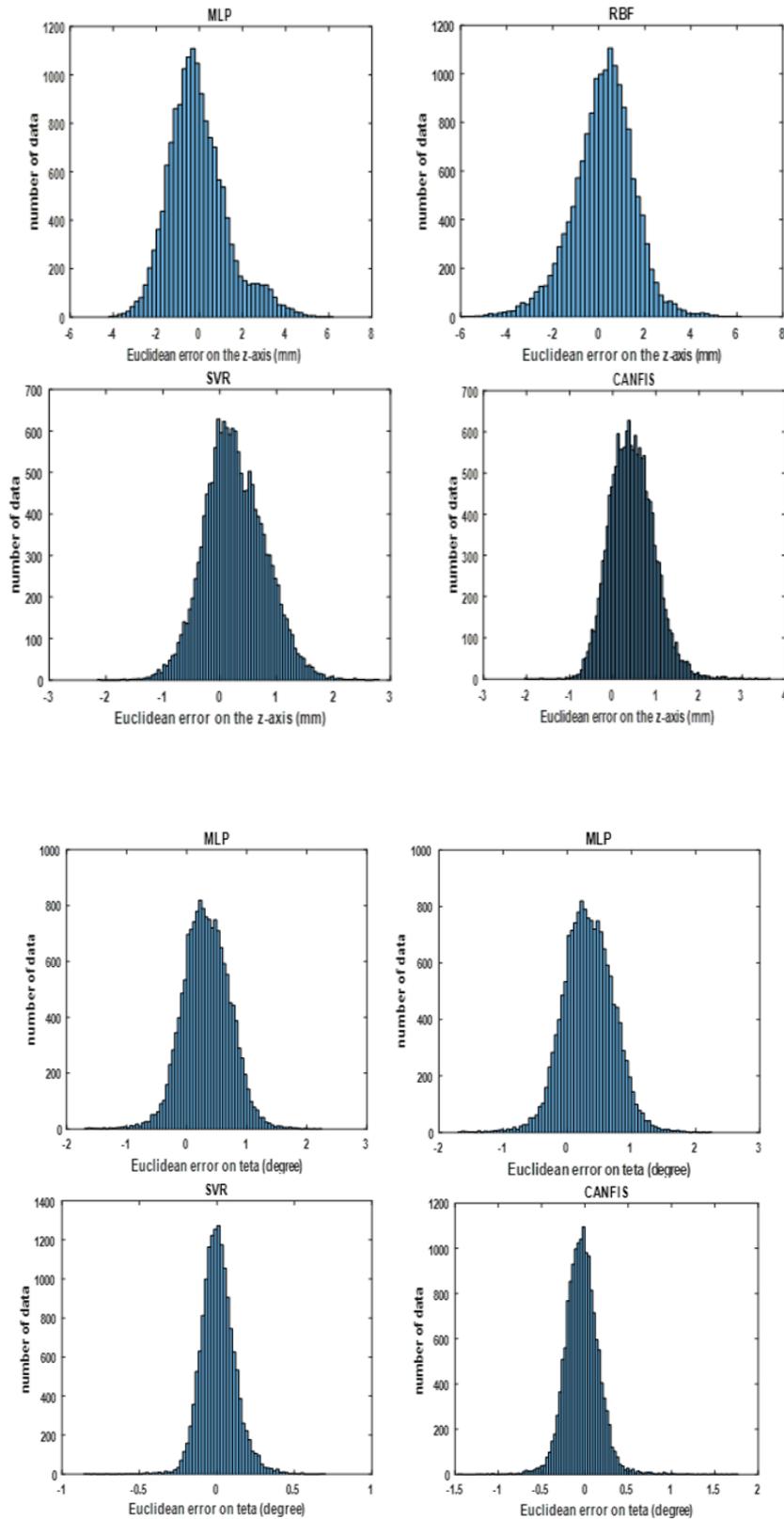


FIGURE 3.13 – Erreur euclidien suivant les axes X et Y

FIGURE 3.14 – Erreur euclidien suivant la position z et l'orientation θ

3.3. Modélisation géométrique du manipulateur continu : Cas du CBH7

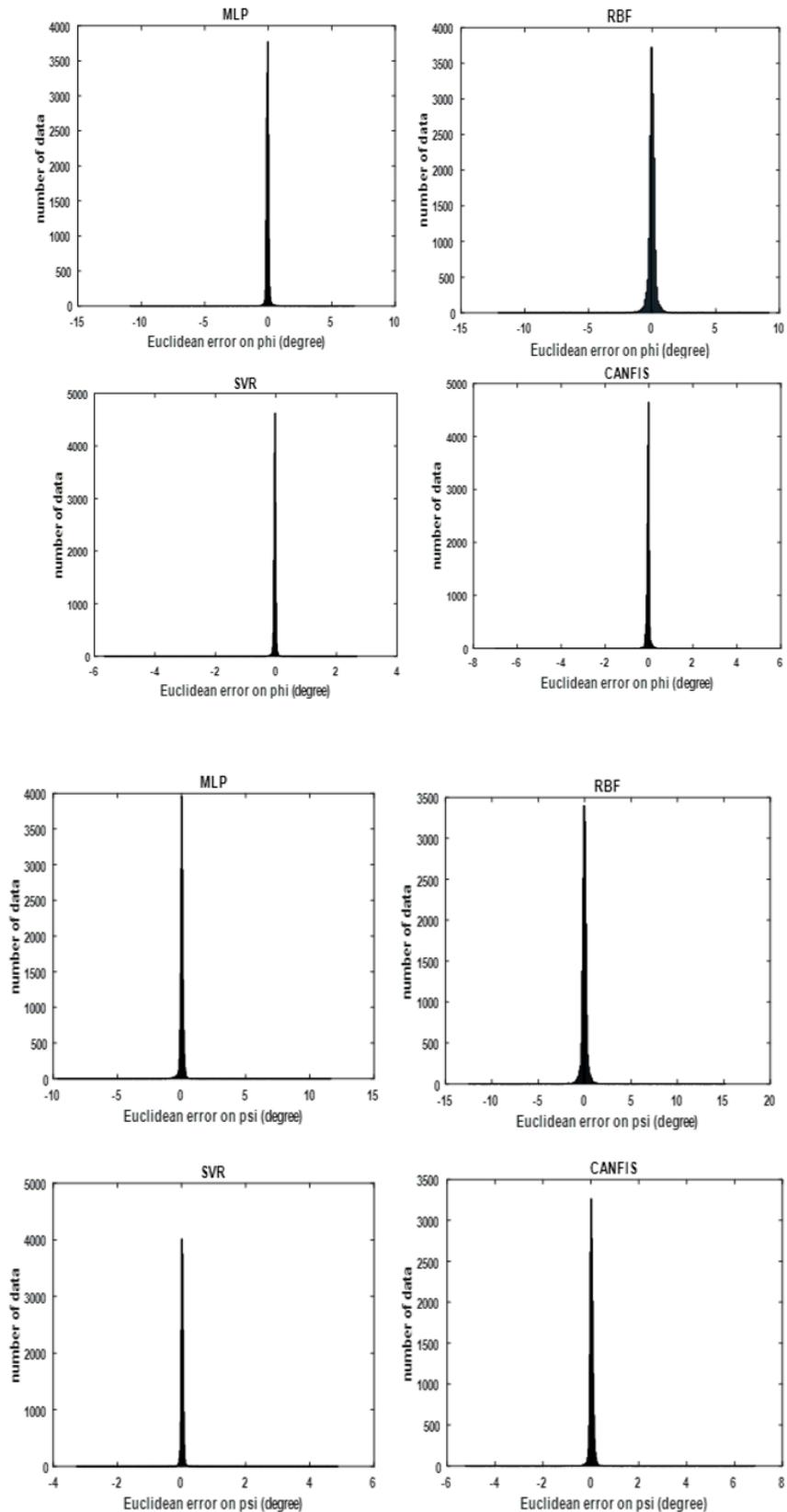


FIGURE 3.15 – Erreur euclidien suivant les orientations ϕ et ψ

La configuration se compose également du logiciel « ControlDesk » qui offre une interface utilisateur polyvalente pour la surveillance et le contrôle du manipulateur CBHA. L'ordinateur de simulation est relié à DSpace par la carte de liaison DS814. La DS1005 est la carte du processeur principal de DSpace, qui possède un PowerPC 750GX - 1 GHz. Notre objectif est de montrer que les modèles neuronaux obtenus au cours du processus d'apprentissage peuvent prédire en temps réel les tensions des potentiomètres à fil du CBHA. Cependant, en raison des limitations physiques des actionneurs pneumatiques, le robot a besoin d'un certain temps pour obtenir un équilibre mécanique (environ 10 secondes). Ce travail n'aborde pas le problème du suivi de trajectoire, mais celui de la géométrie inverse du CBHA. Le suivi de trajectoire dans l'espace de travail du robot nécessite non seulement une transformation des tensions en pressions, mais aussi la conception d'un contrôleur adaptatif robuste. Dans cette sous-section, nous présentons uniquement les performances obtenues par le MLP. Le RBF fournit de faibles performances pour une pression constante (la pression est presque nulle dans la première expérience), nous avons donc jugé inutile d'utiliser le modèle RBF pour une implémentation en temps réel (avec une pression variable). L'expérience est réalisée en utilisant des actionneurs pneumatiques à vitesse variable. Cette expérience est un véritable banc d'essai pour le modèle neuronal, car il est préférable de déplacer le CBHA à une vitesse constante, et d'attendre environ 10 secondes pour passer au point suivant. Mais, comme nous pouvons le voir sur la Figure 3.17, le modèle neuronal prédit en temps réel, les tensions des fils-potentiomètres du CBHA. Nous observons une erreur euclidienne d'environ $\pm 0.03V$, représentée sur la Figure 3.18, correspondant à un allongement d'environ $\pm 0.005m$.

3.4 Navigation du manipulateur mobile continuum : Cas du Robotino XT

Afin de valider le contrôleur développé, des expériences en temps réel ont été menées. Dans cette dernière section, nous présentons deux scénarios. La figure 3.19 et la figure

3.4. Navigation du manipulateur mobile continuum : Cas du Robotino XT79

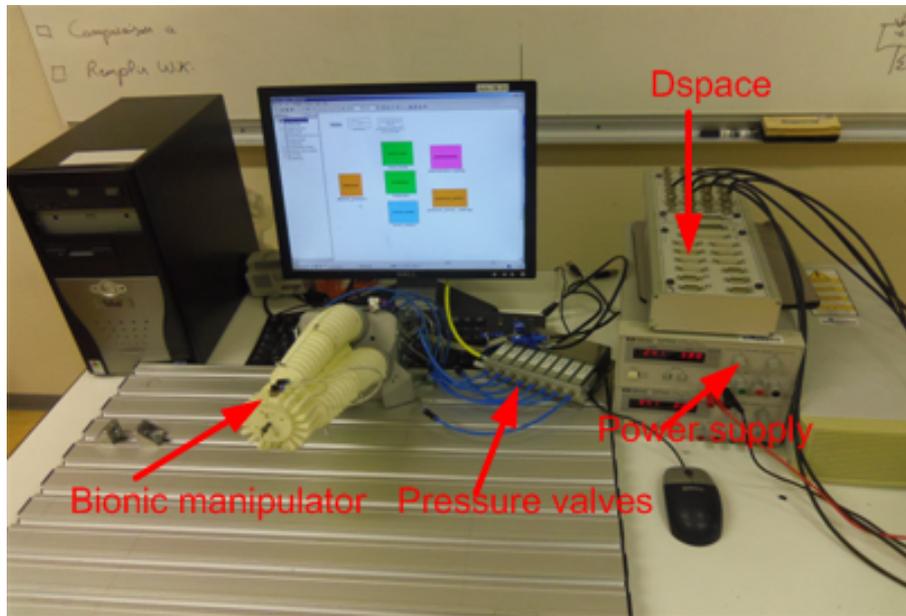


FIGURE 3.16 – Environnement experimental

3.20 représentent le premier et le second environnement expérimental respectivement. La localisation du robot est basée sur un système d'odométrie lors de la navigation de la plateforme mobile. En revanche, le processus de suivi de trajectoire pour l'atteinte de la cible n'est pas basé sur un système de traitement d'image. Ici, le robot embarque un capteur infrarouge sur l'effecteur du robot qui lui permet de saisir l'objet une fois qu'il est à proximité de ce dernier. Le capteur infrarouge est préféré à la caméra, car notre objectif est de valoriser notre contrôleur développé. L'utilisation d'une caméra s'apparentera à un suivi de trajectoire de la cible basé sur des outils de traitement d'image. Chaque environnement contient des obstacles mobiles et statiques. La tâche est de saisir l'objet de couleur rouge. Les différents contrôleurs ne suivent pas une trajectoire ou un chemin prédéfini. La sortie du contrôleur de la plateforme mobile dépend de la configuration de l'environnement, tandis que le contrôleur du manipulateur continuum effectue le suivi de points. Le mouvement du CBHA, de la position initiale à la cible, est donc libre.

Dans la première phase de l'expérience qui équivaut au premier scénario, l'objet est

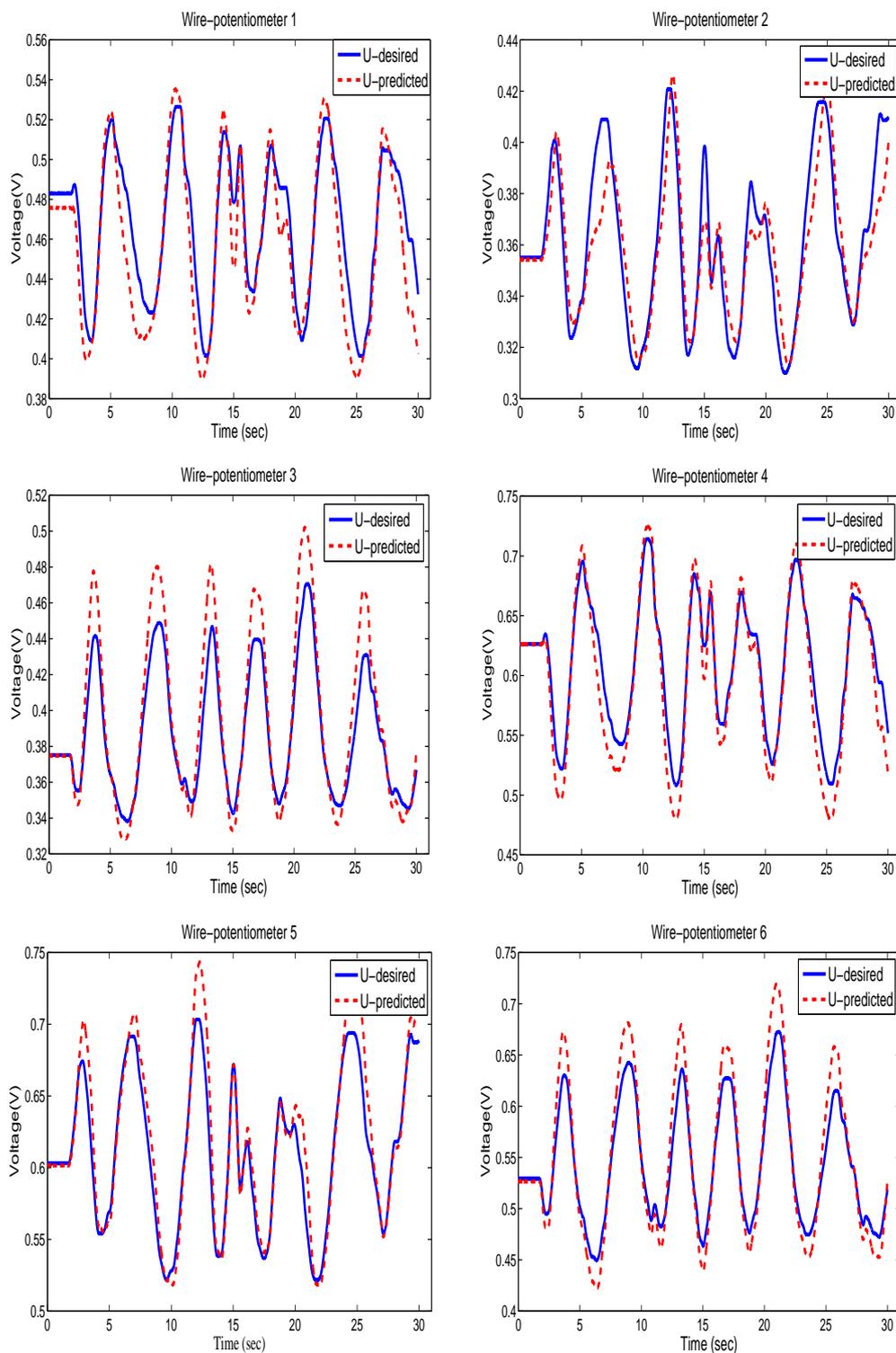


FIGURE 3.17 – Mise en oeuvre en temps réel : Prédiction des tensions du fil-potentiomètre par le réseau MLP

3.4. Navigation du manipulateur mobile continuum : Cas du Robotino XT81

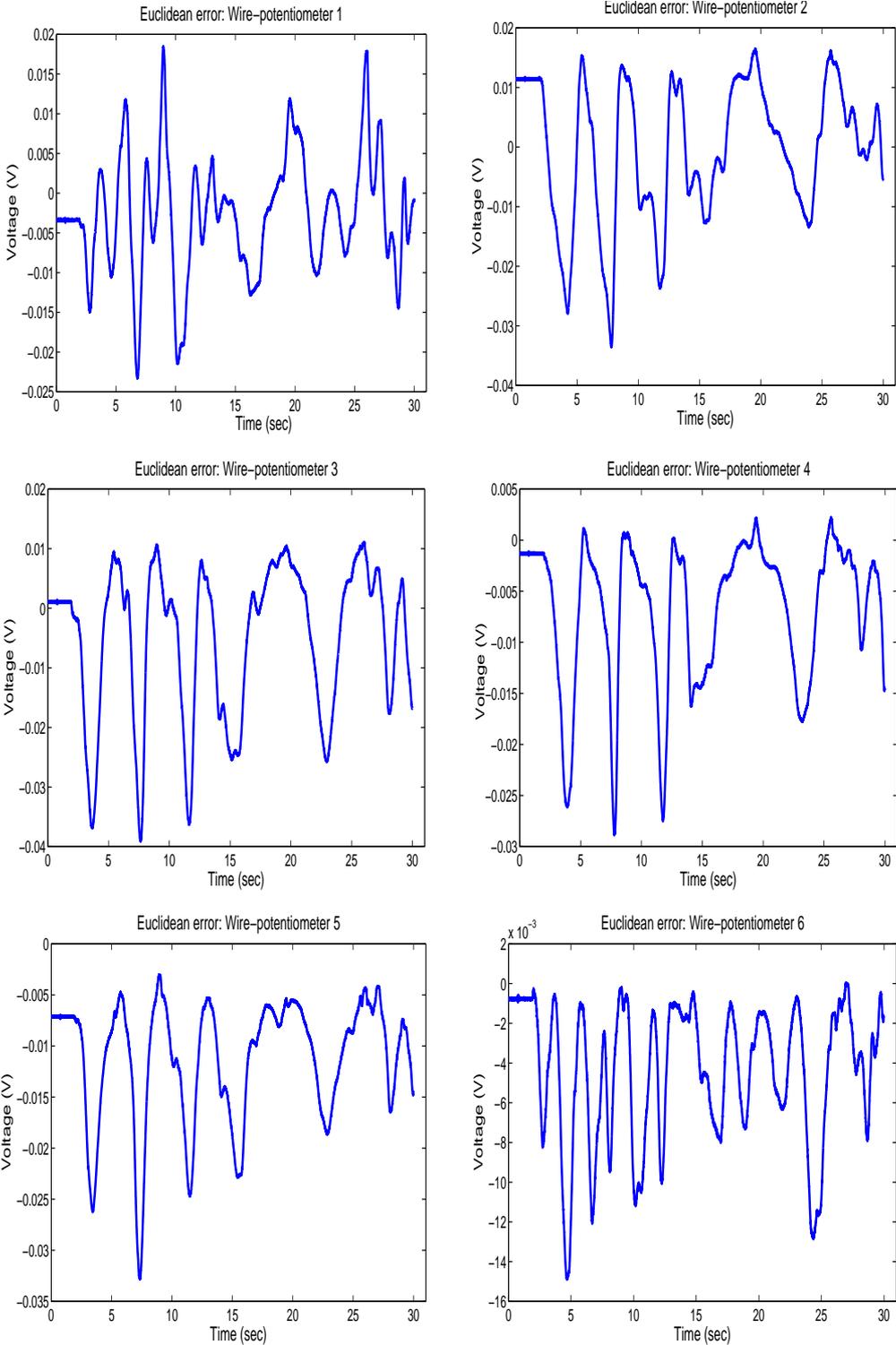


FIGURE 3.18 – Mise en oeuvre en temps réel : Erreurs euclidiennes, MLP

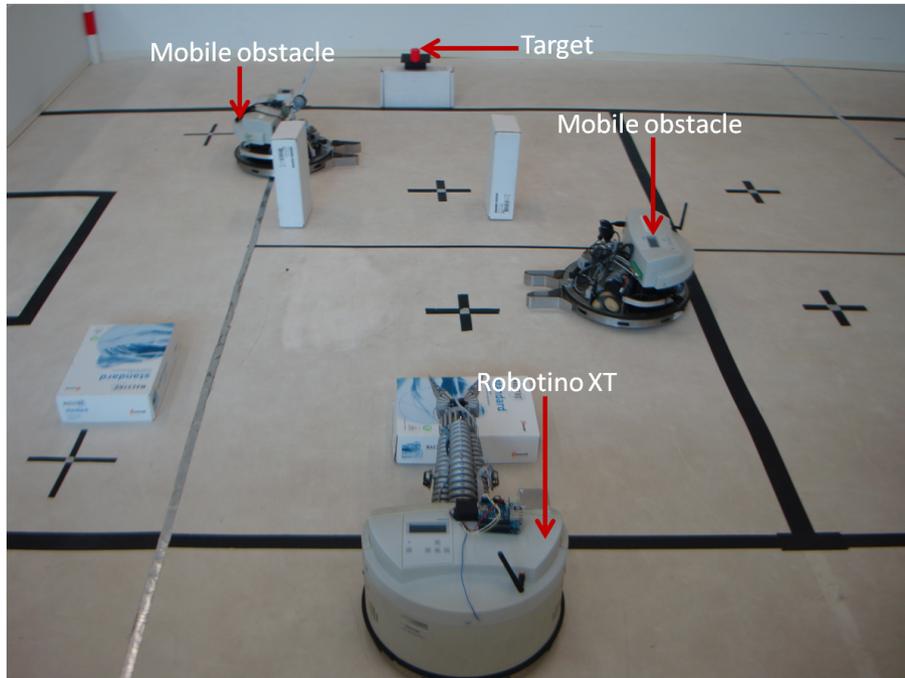


FIGURE 3.19 – Environnement expérimental : Premier scénario

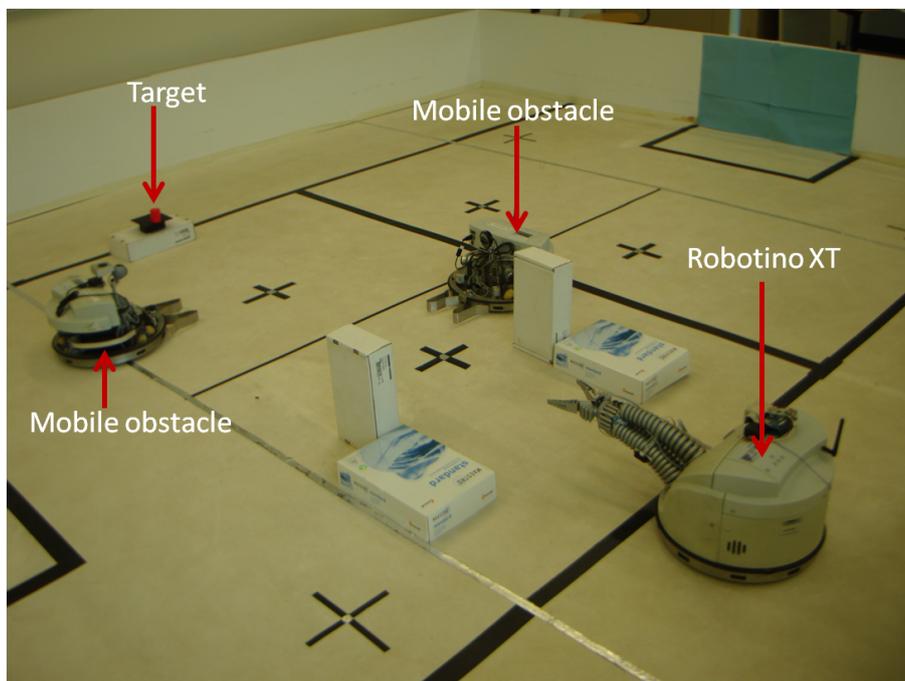


FIGURE 3.20 – Environnement expérimental : Second scénario

3.4. Navigation du manipulateur mobile continuum : Cas du Robotino XT83

situé à une distance de $(2000, 0, 112)mm$ par rapport au référentiel de la plate-forme. La figure 3.21 représente la trajectoire décrite par la position de la pointe du robot. Les vitesses générées par le contrôleur de la plate-forme mobile (contrôleur potentiel neuro-flou de type 2) sont représentées sur la figure 3.22. Le tableau 3.8 répertorie les signaux de commande générés par le contrôleur à réseau neuronal adaptatif. Étant donné que la coordonnée z de l'objet se trouve au voisinage de la position initiale du manipulateur, l'algorithme ne fonctionne qu'en deux étapes. Le nombre des étapes est considéré comme le nombre de fois qu'un signal de commande est appliqué au CBHA.

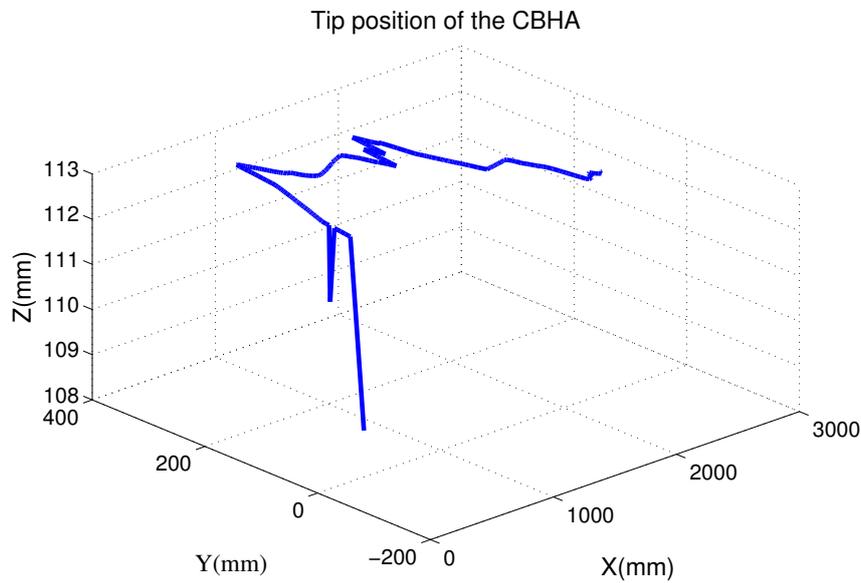


FIGURE 3.21 – Premier scénario : Trajectoire décrite par la position de la pointe du robot.

TABLE 3.8 – Valeurs des signaux de commande générés par le contrôleur du manipulateur continu dans le premier scénario.

Pression	p_1	p_2	p_3	p_4	p_5	p_6	Time (s)
1ère Etape	0.6204	0.5906	0.6306	0.4408	0.6554	0.1240	3.00
2nd Etape	0.6166	0.5919	0.6314	0.4420	0.6512	0.1251	1.5357

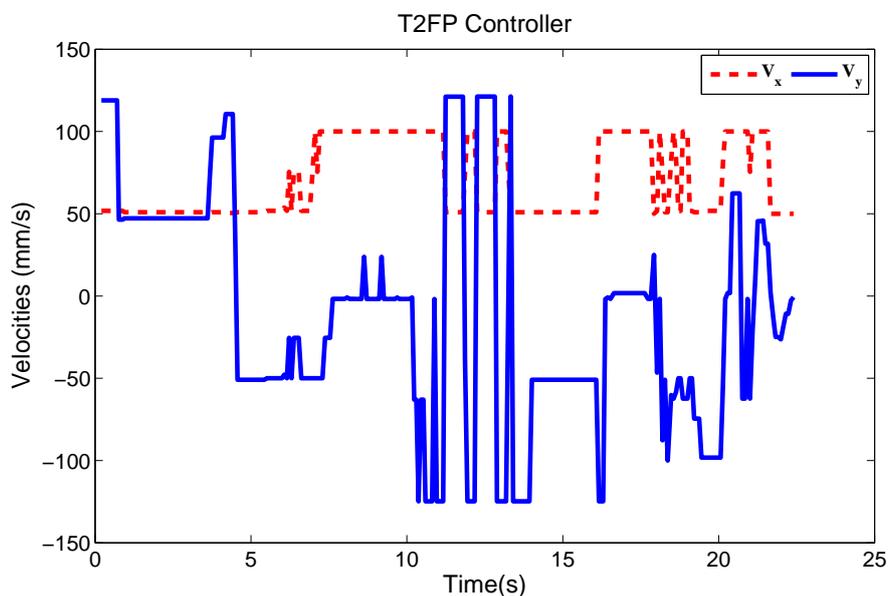


FIGURE 3.22 – Premier scénario : Vitesses générées par le contrôleur de la plate-forme mobile (T2FNN).

Dans la seconde phase de l'expérience qui équivaut au second scénario, l'objet est situé à $(2000, 0, 245)mm$ par rapport au cadre de la plate-forme mobile. La trajectoire décrite par la position de l'extrémité du robot est représentée sur la figure 3.23. Les vitesses générées par le contrôleur de la plate-forme mobile (contrôleur hybride potentiel neuro-flou de type 2) sont représentées sur la figure 3.24. Le tableau 3.9 répertorie les signaux de commande générés par le contrôleur à réseau neuronal adaptatif. Il s'exécute en trois étapes, du fait qu'ici la coordonnée de z se retrouve un peu plus éloignée de la position initiale.

Au vu des résultats obtenus, notre contrôleur réalise de bonnes performances. Il effectue ses tâches sans collision, tout en prenant un temps raisonnable. La trajectoire décrite par la position de la pointe du robot est imprévisible, du fait de l'environnement dynamique et inconnu. Les vitesses générées par le contrôleur de la plate-forme mobile dépendent de la configuration locale des obstacles.

3.4. Navigation du manipulateur mobile continuum : Cas du Robotino XT85

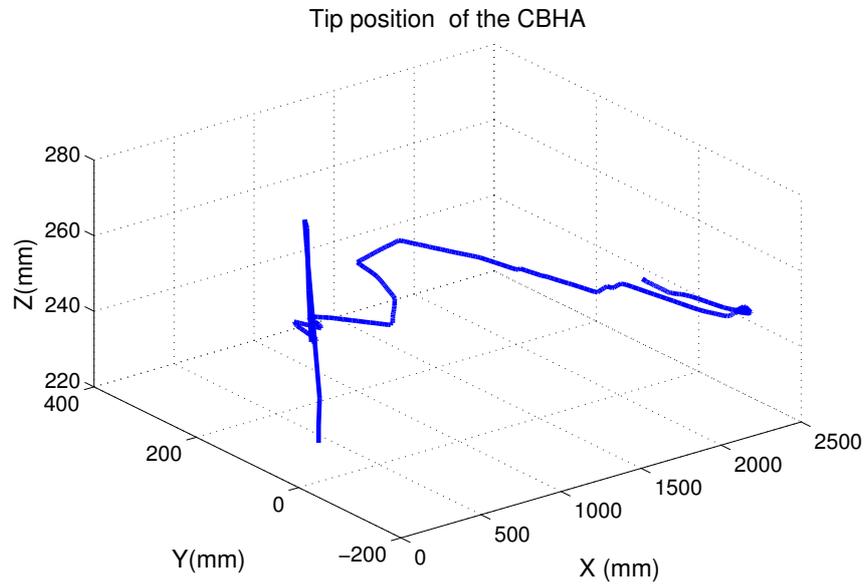


FIGURE 3.23 – Second scénario : Trajectoire décrite par la position de l'extrémité du Robot

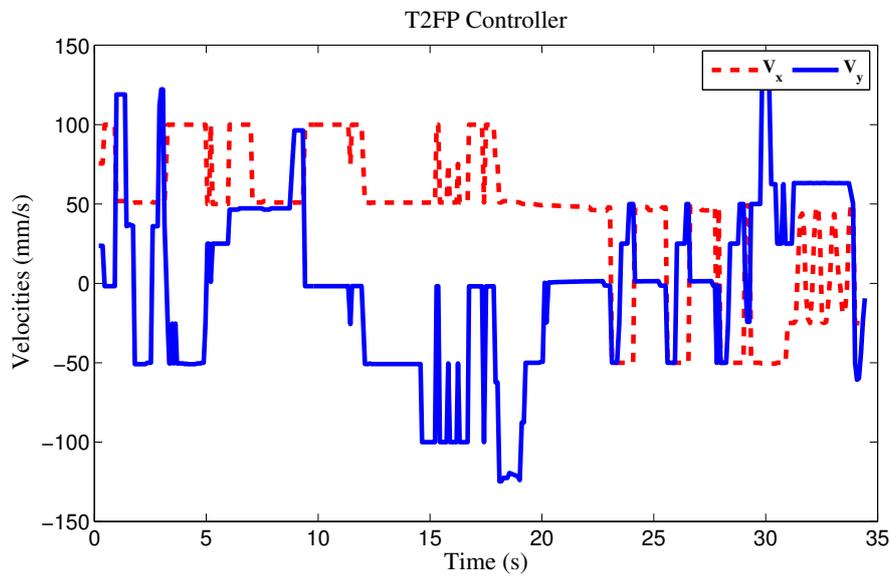


FIGURE 3.24 – Second scénario : Vitesses générées par le contrôleur de la plate-forme mobile (T2FNN).

TABLE 3.9 – Valeurs des signaux de commande générés par le contrôleur du manipulateur continu dans le second scénario.

Pression	p_1	p_2	p_3	p_4	p_5	p_6	Temps (s)
1ère Etape	0.0450	0.4550	0.2749	0.8014	0.6451	0.2521	3.00
2ème Etape	0.0274	0.4740	0.2908	0.7064	0.7257	0.2966	1.4132
3ème Etape	0.0117	0.4878	0.3031	0.6168	0.7970	0.3307	1.1275

3.5 Conclusion

Dans ce chapitre, nous avons présenté, analysé et discuté les résultats des différentes simulations et expériences réalisées. Nous avons observé que le robotinoXT pouvait atteindre les cibles définies tout en évitant les obstacles rencontrés. Cependant, nous avons noté quelques cas d'échec dus notamment au système de localisation. Cette limite peut être remédiée en plaçant par exemple une camera miniaturisée au niveau de l'effecteur du manipulateur continu.

Conclusion générale et Perspectives

Cette thèse s'articule autour de trois principaux thèmes :

- **Navigation autonome de robots mobiles** : Nous avons proposé un système de navigation robuste et efficace en termes de temps de calcul pour les robots mobiles en utilisant un IT2FNN. L'opération de réduction de type K-M est remplacée par deux q-facteurs de conception qui sont appris par l'algorithme de rétropropagation du gradient dans la structure IT2FNN [Imrane 2021] afin de rendre le contrôleur efficace en termes de temps de calcul. Comme ont montré les résultats expérimentaux, le robot mobile utilisant le contrôleur IT2FNN peut atteindre les cibles et éviter les obstacles de manière satisfaisante, quelle que soit la forme de l'obstacle rencontré. Les comparaisons des distances parcourues et du temps de déplacement ont montré que les performances du robot avec l'IT2FNN sont meilleures que celles du contrôleur IT2FL proposé par [Melinguì 2014d], c'est-à-dire que le nouvel IT2FNN garantira un temps de déplacement plus court et réalisera des distances de déplacement plus courtes que l'IT2FL précédent. Par conséquent, nous pouvons conclure que le nouveau système de navigation est une approche plus pratique et fonctionne mieux que le contrôleur IT2FL proposé dans [Melinguì 2014d]. La structure du réseau neuronal flou, qui peut effectuer plusieurs tâches simultanément en un temps raisonnable, serait responsable ces avantages.

On peut s'attendre à ce que d'autres études impliquant l'IT2FNN pour la navigation de robots mobiles autonomes aient lieu dans le futur.

- **Modélisation géométrique directe et inverse** : Une étude comparative de quatre techniques d'apprentissage dans le cadre de la modélisation géométrique directe du manipulateur mobile continuum à sections multiples a été réalisée. Des modèles MLP, RBF, SVR et CANFIS ont été implémentés pour la modélisation géométrique d'un manipulateur continuum à deux sections [Loutfi 2020]. Les pré-

dictions des différentes topologies ont été comparées respectivement à un système de capture de mouvement de haute précision. De nombreux critères de performance ont été utilisés pour évaluer les performances des différentes techniques basées sur l'apprentissage, à savoir le coefficient de corrélation, le temps d'apprentissage, l'erreur quadratique moyenne et les erreurs cartésiennes restantes. Au vu des résultats obtenus, une meilleure précision du modèle SVR a été observée, suivie respectivement par les CANFIS, RBF et MLP. Le modèle SVR est le plus gourmand en temps d'apprentissage, suivi respectivement par les modèles CANFIS, MLP et RBF. Cependant, le temps de calcul des différents régresseurs obtenus après la phase d'apprentissage est presque le même ; le RBF étant le moins long et CANFIS le plus long. Nous avons également proposé le modèle géométrique inverse via l'apprentissage supervisé distal en utilisant les réseaux de neurones MLP.

Dans des travaux futurs, il serait intéressant d'évaluer les performances des autres modèles d'apprentissage automatique dans le cadre de la modélisation géométrique inverse de manipulateurs de type continuum.

- **Navigation autonome de manipulateurs à continuum mobile** : Une architecture de contrôle qui combine le contrôle de la plate-forme mobile et du manipulateur continuum du RobotinoXT a été également développée. Deux sous-contrôleurs sont coordonnés par un système de réseaux neuronaux. Ils sont implémentés dans la même architecture de commande, et fonctionnent simultanément. Par le biais d'expériences, nous avons montré que le RobotinoXT peut saisir un objet de forme prédéfinie et connue en temps réel.

Au regard des erreurs observées dans les expériences, il serait intéressant à l'avenir de prendre en compte la dynamique du manipulateur mobile continuum dans le développement des différentes lois de contrôle.

Bibliographie

- [Astudillo 2013] Leslie Astudillo, Patricia Melin et Oscar Castillo. *Chemical optimization paradigm applied to a fuzzy tracking controller for an autonomous mobile robot*. International Journal of Innovative Computing, Information and Control, vol. 9, no. 5, pages 2007–2018, 2013.
- [Bahrin 2016] Mohd Aiman Kamarul Bahrin, Mohd Fauzi Othman, Nor Hayati Nor Azli et Muhamad Farihin Talib. *Industry 4.0 : A review on industrial automation and robotic*. Jurnal Teknologi, vol. 78, no. 6-13, 2016.
- [Bailly 2005] Yan Bailly et Yacine Amirat. *Modeling and control of a hybrid continuum active catheter for aortic aneurysm treatment*. In Robotics and Automation, 2005. ICRA 2005. Proceedings of the 2005 IEEE International Conference on, pages 924–929. IEEE, 2005.
- [Bakdi 2017] Azzeddine Bakdi, Abdelfetah Hentout, Hakim Boutami, Abderraouf Maoudj, Ouarda Hachour et Brahim Bouzouia. *Optimal path planning and execution for mobile robots using genetic algorithm and adaptive fuzzy-logic control*. Robotics and Autonomous Systems, vol. 89, pages 95–109, 2017.
- [Beom 1995] Hee Rak Beom et Hyung Suck Cho. *A sensor-based navigation for a mobile robot using fuzzy logic and reinforcement learning*. IEEE transactions on Systems, Man, and Cybernetics, vol. 25, no. 3, pages 464–477, 1995.
- [Bischof 1992] Horst Bischof, Werner Schneider et Axel J Pinz. *Multispectral classification of Landsat-images using neural networks*. IEEE transactions on Geoscience and Remote Sensing, vol. 30, no. 3, pages 482–490, 1992.
- [Bishop 1995] Christopher M Bishop. *Neural networks for pattern recognition*. Oxford university press, 1995.
- [Bishop 2006] Christopher M Bishop. *Pattern recognition*. Machine learning, vol. 128, no. 9, 2006.

- [Boutchouang 2021] AH Bouyom Boutchouang, Achille Melingui, JJB Mvogo Ahanda, Othman Lakhal, Frederic Biya Motto et Rochdi Merzouki. *Forward Kinematic Modeling of Conical-Shaped Continuum Manipulators*. *Robotica*, vol. 39, no. 10, pages 1760–1778, 2021.
- [Chakrabarty 2016] Ankush Chakrabarty, Vu Dinh, Martin J Corless, Ann E Rundell, Stanisław H Żak et Gregery T Buzzard. *Support vector machine informed explicit nonlinear model predictive control using low-discrepancy sequences*. *IEEE Transactions on Automatic Control*, vol. 62, no. 1, pages 135–148, 2016.
- [Chiddarwar 2010] Shital S Chiddarwar et N Ramesh Babu. *Comparison of RBF and MLP neural networks to solve inverse kinematic problem for 6R serial robot by a fusion approach*. *Engineering Applications of Artificial Intelligence*, vol. 23, no. 7, pages 1083–1092, 2010.
- [Chirikjian 1994] Gregory S Chirikjian et Joel W Burdick. *A modal approach to hyper-redundant manipulator kinematics*. *IEEE Transactions on Robotics and Automation*, vol. 10, no. 3, pages 343–354, 1994.
- [Cieślak 1999] Radosław Cieślak et Adam Morecki. *Elephant trunk type elastic manipulator-a tool for bulk and liquid materials transportation*. *Robotica*, vol. 17, no. 1, pages 11–16, 1999.
- [Daachi 2006] Boubaker Daachi et Abdelaziz Benallegue. *A neural network adaptive controller for end-effector tracking of redundant robot manipulators*. *Journal of Intelligent and Robotic Systems*, vol. 46, no. 3, pages 245–262, 2006.
- [Daachi 2012] Boubaker Daachi, Tarek Madani et Abdelaziz Benallegue. *Adaptive neural controller for redundant robot manipulators and collision avoidance with mobile obstacles*. *Neurocomputing*, vol. 79, pages 50–60, 2012.
- [Datouo 2017] R Datouo, F Biya Motto, B Essimbi Zobo, Achille Melingui, Ismail Benskrane et Rochdi Merzouki. *Optimal motion planning for minimizing energy consumption of wheeled mobile robots*. In *2017 IEEE International Conference on Robotics and Biomimetics (ROBIO)*, pages 2179–2184. IEEE, 2017.

- [Elish 2014] Mahmoud O Elish. *A comparative study of fault density prediction in aspect-oriented systems using MLP, RBF, KNN, RT, DENFIS and SVR models*. Artificial Intelligence Review, vol. 42, no. 4, pages 695–703, 2014.
- [Escande 2012] Coralie Escande, Rochdi Merzouki, Pushparaj Mani Pathak et Vincent Coelen. *Geometric modelling of multisection bionic manipulator : Experimental validation on robotinoxt*. In Robotics and Biomimetics (ROBIO), 2012 IEEE International Conference on, pages 2006–2011. IEEE, 2012.
- [Faisal 2013] Mohammed Faisal, Ramdane Hedjar, Mansour Al Sulaiman et Khalid Al-Mutib. *Fuzzy logic navigation and obstacle avoidance by a mobile robot in an unknown dynamic environment*. International Journal of Advanced Robotic Systems, vol. 10, no. 1, page 37, 2013.
- [Falkenhahn 2014] Valentin Falkenhahn, Tobias Mahl, Alexander Hildebrandt, Rüdiger Neumann et Oliver Sawodny. *Dynamic modeling of constant curvature continuum robots using the Euler-Lagrange formalism*. In 2014 IEEE/RSJ International Conference on Intelligent Robots and Systems, pages 2428–2433. IEEE, 2014.
- [Falkenhahn 2015] Valentin Falkenhahn, Tobias Mahl, Alexander Hildebrandt, Rüdiger Neumann et Oliver Sawodny. *Dynamic modeling of bellows-actuated continuum robots using the Euler-Lagrange formalism*. IEEE Transactions on Robotics, vol. 31, no. 6, pages 1483–1496, 2015.
- [Frazelle 2020] Chase Frazelle, Jonathan Rogers, Ioannis Karamouzas et Ian Walker. *Optimizing a Continuum Manipulator’s Search Policy Through Model-Free Reinforcement Learning*. In 2020 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS), pages 5564–5571. IEEE, 2020.
- [Ghorbani 2016] Mohammad Ali Ghorbani, Hojat Ahmad Zadeh, Mohammad Isazadeh et Ozlem Terzi. *A comparative study of artificial neural network (MLP, RBF) and support vector machine models for river flow prediction*. Environmental Earth Sciences, vol. 75, no. 6, page 476, 2016.

- [Giorelli 2013a] M Giorelli, F Renda, G Ferri et C Laschi. *A Feed-Forward Neural Network Learning the Inverse Kinetics of a Soft Cable-Driven Manipulator Moving in Three-Dimensional Space*. in proc. IEEE Int. Conf. on Intelligent Robots and Systems, pages 5033–5039, 2013.
- [Giorelli 2013b] Michele Giorelli, Federico Renda, Gabriele Ferri et Cecilia Laschi. *A feed-forward neural network learning the inverse kinetics of a soft cable-driven manipulator moving in three-dimensional space*. In Intelligent Robots and Systems (IROS), 2013 IEEE/RSJ International Conference on, pages 5033–5039. IEEE, 2013.
- [Giorelli 2015] Michele Giorelli, Federico Renda, Marcello Calisti, Andrea Arienti, Gabriele Ferri et Cecilia Laschi. *Neural network and jacobian method for solving the inverse statics of a cable-driven soft arm with nonconstant curvature*. IEEE Transactions on Robotics, vol. 31, no. 4, pages 823–834, 2015.
- [Godage 2011] Isuru S Godage, Emanuele Guglielmino, David T Branson, Gustavo A Medrano-Cerda et Darwin G Caldwell. *Novel modal approach for kinematics of multisection continuum arms*. In Intelligent Robots and Systems (IROS), 2011 IEEE/RSJ International Conference on, pages 1093–1098. IEEE, 2011.
- [Godage 2015] Isuru S Godage, Gustavo A Medrano-Cerda, David T Branson, Emanuele Guglielmino et Darwin G Caldwell. *Modal kinematics for multisection continuum arms*. Bioinspiration & biomimetics, vol. 10, no. 3, page 035002, 2015.
- [Goharimanesh 2020] M Goharimanesh, Ali Mehrkish et Farrokh Janabi-Sharifi. *A fuzzy reinforcement learning approach for continuum robot control*. Journal of Intelligent & Robotic Systems, vol. 100, no. 3, pages 809–826, 2020.
- [Hagras 2004a] Hani Hagras, Victor Callaghan et Martin Colley. *Learning and adaptation of an intelligent mobile robot navigator operating in unstructured environment based on a novel online Fuzzy-Genetic system*. Fuzzy Sets and Systems, vol. 141, no. 1, pages 107–160, 2004.

- [Hagras 2004b] Hani A Hagras. *A hierarchical type-2 fuzzy logic control architecture for autonomous mobile robots*. IEEE Transactions on Fuzzy systems, vol. 12, no. 4, pages 524–539, 2004.
- [Hamet 2017] Pavel Hamet et Johanne Tremblay. *Artificial intelligence in medicine*. Metabolism, vol. 69, pages S36–S40, 2017.
- [Hannan 2003] Michael W Hannan et Ian D Walker. *Kinematics and the implementation of an elephant’s trunk manipulator and other continuum style robots*. Journal of robotic systems, vol. 20, no. 2, pages 45–63, 2003.
- [Hawks 2019] Zachary Hawks, Chase Frazelle, Keith E Green et Ian D Walker. *Motion planning for a continuum robotic mobile lamp : Defining and navigating the configuration space*. In 2019 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS), pages 2559–2566. IEEE, 2019.
- [Haykin 1994] Simon Haykin. *Neural networks : a comprehensive foundation*. Prentice Hall PTR, 1994.
- [He 2015] Wei He, Shuzhi Sam Ge, Yanan Li, Effie Chew et Yee Sien Ng. *Neural network control of a rehabilitation robot by state and output feedback*. Journal of Intelligent & Robotic Systems, vol. 80, no. 1, pages 15–31, 2015.
- [Huang 2017] Jian Huang, MyongHyok Ri, Dongrui Wu et Songhyok Ri. *Interval type-2 fuzzy logic modeling and control of a mobile two-wheeled inverted pendulum*. IEEE Transactions on Fuzzy Systems, vol. 26, no. 4, pages 2030–2038, 2017.
- [Imrane 2021] Mahamat Loutfi Imrane, Achille Melingui, Joseph Jean Baptiste Mvogo Ahanda, Frédéric Biya Motto et Rochdi Merzouki. *Artificial potential field neuro-fuzzy controller for autonomous navigation of mobile robots*. Proceedings of the Institution of Mechanical Engineers, Part I : Journal of Systems and Control Engineering, vol. 235, no. 7, pages 1179–1192, 2021.
- [Jang 1993] J-SR Jang. *ANFIS : adaptive-network-based fuzzy inference system*. IEEE transactions on systems, man, and cybernetics, vol. 23, no. 3, pages 665–685, 1993.

- [Jin 2000] Yaochu Jin. *Fuzzy modeling of high-dimensional systems : complexity reduction and interpretability improvement*. IEEE Transactions on Fuzzy Systems, vol. 8, no. 2, pages 212–221, 2000.
- [Jones 2006] Bryan A Jones et Ian D Walker. *Kinematics for multisection continuum robots*. IEEE Transactions on Robotics, vol. 22, no. 1, pages 43–55, 2006.
- [Jordan 1992] Michael I Jordan et David E Rumelhart. *Forward models : Supervised learning with a distal teacher*. Cognitive science, vol. 16, no. 3, pages 307–354, 1992.
- [Juang 2009] Chia-Feng Juang et Chia-Hung Hsu. *Reinforcement ant optimized fuzzy controller for mobile-robot wall-following control*. IEEE Transactions on Industrial Electronics, vol. 56, no. 10, pages 3931–3940, 2009.
- [Juang 2012] Chia-Feng Juang et Kai-Jie Juang. *Reduced interval type-2 neural fuzzy system using weighted bound-set boundary operation for computation speedup and chip implementation*. IEEE Transactions on Fuzzy Systems, vol. 21, no. 3, pages 477–491, 2012.
- [Junratanasiri 2011] Sittichok Junratanasiri, Sansanee Auephanwiriyaikul et Nipon Theera-Umporn. *Navigation system of mobile robot in an uncertain environment using type-2 fuzzy modelling*. In 2011 IEEE International Conference on Fuzzy Systems (FUZZ-IEEE 2011), pages 1171–1178. IEEE, 2011.
- [Karray 2016] Amal Karray, Malek Njah, Moez Feki et Mohamed Jallouli. *Intelligent mobile manipulator navigation using hybrid adaptive-fuzzy controller*. Computers & Electrical Engineering, vol. 56, pages 773–783, 2016.
- [Khatib 1986] Oussama Khatib. *Real-time obstacle avoidance for manipulators and mobile robots*. In Autonomous robot vehicles, pages 396–404. Springer, 1986.
- [Kim 2014] Cheol-Joong Kim et Dongkyoung Chwa. *Obstacle avoidance method for wheeled mobile robots using interval type-2 fuzzy neural network*. IEEE Transactions on Fuzzy Systems, vol. 23, no. 3, pages 677–687, 2014.

- [Ko 2013] Chia-Nan Ko et Cheng-Ming Lee. *Short-term load forecasting using SVR (support vector regression)-based radial basis function neural network with dual extended Kalman filter*. Energy, vol. 49, pages 413–422, 2013.
- [Lakhal 2015] O. Lakhal, A. Melingui et R. Merzouki. *Hybrid approach for modeling and solving of kinematics of compact bionic handling assistant manipulator*. Mechanics, IEEE/ASME Transactions on, vol. PP, no. 99, pages 1–1, 2015.
- [Lane 1997] David M Lane, J Bruce C Davies, Giuseppe Casalino, Giorgio Bartolini, Giorgio Cannata, Gianmarco Veruggio, Miquel Canals, C Smith, Desmond J O’Brien, M Pickett et al. *AMADEUS : advanced manipulation for deep underwater sampling*. IEEE Robotics & Automation Magazine, vol. 4, no. 4, pages 34–45, 1997.
- [Laschi 2009] Cecilia Laschi, Barbara Mazzolai, V Mattoli, M Cianchetti et P Dario. *Design of a biomimetic robotic octopus arm*. Bioinspiration & biomimetics, vol. 4, no. 1, page 015006, 2009.
- [Le 2014] Tien Dung Le et Hee-Jun Kang. *An adaptive tracking controller for parallel robotic manipulators based on fully tuned radial basic function networks*. Neurocomputing, vol. 137, pages 12–23, 2014.
- [Li 2011] Chaofeng Li, Alan Conrad Bovik et Xiaojun Wu. *Blind image quality assessment using a general regression neural network*. IEEE Transactions on neural networks, vol. 22, no. 5, pages 793–799, 2011.
- [Li 2015] Jinglin Li et Jing Xiao. *A general formulation and approach to constrained, continuum manipulation*. Advanced Robotics, vol. 29, no. 13, pages 889–899, 2015.
- [Lin 2013] Yang-Yin Lin, Shih-Hui Liao, Jyh-Yeong Chang et Chin-Teng Lin. *Simplified interval type-2 fuzzy neural networks*. IEEE transactions on neural networks and learning systems, vol. 25, no. 5, pages 959–969, 2013.
- [Liu 2011] Xinwang Liu et Jerry M Mendel. *Connect Karnik-Mendel algorithms to root-finding for computing the centroid of an interval type-2 fuzzy set*. IEEE Transactions on Fuzzy Systems, vol. 19, no. 4, pages 652–665, 2011.

- [Liu 2016] Shuntao Liu, Zhixiong Yang, Zhijun Zhu, Liangliang Han, Xiangyang Zhu et Kai Xu. *Development of a dexterous continuum manipulator for exploration and inspection in confined spaces*. *Industrial Robot : An International Journal*, 2016.
- [Loutfi 2020] I Mahamat Loutfi, AH Bouyom Boutchouang, A Melingui, O Lakhal, F Biya Motto et R Merzouki. *Learning-Based Approaches for Forward Kinematic Modeling of Continuum Manipulators*. *IFAC-PapersOnLine*, vol. 53, no. 2, pages 9899–9904, 2020.
- [Mahl 2014] Tobias Mahl, Alexander Hildebrandt et Oliver Sawodny. *A variable curvature continuum kinematics for kinematic control of the bionic handling assistant*. *IEEE transactions on robotics*, vol. 30, no. 4, pages 935–949, 2014.
- [Maldonado 2013] Yazmin Maldonado, Oscar Castillo et Patricia Melin. *Particle swarm optimization of interval type-2 fuzzy systems for FPGA applications*. *Applied Soft Computing*, vol. 13, no. 1, pages 496–508, 2013.
- [Marinoudi 2019] Vasso Marinoudi, Claus G Sørensen, Simon Pearson et Dionysis Bochtis. *Robotics and labour in agriculture. A context consideration*. *Biosystems Engineering*, vol. 184, pages 111–121, 2019.
- [Masmoudi 2016] Mohamed Slim Masmoudi, Najla Krichen, Mohamed Masmoudi et Nabil Derbel. *Fuzzy logic controllers design for omnidirectional mobile robot navigation*. *Applied soft computing*, vol. 49, pages 901–919, 2016.
- [Mayorga 1992] René V Mayorga, Andrew KC Wong et N Milano. *A fast procedure for manipulator inverse kinematics evaluation and pseudoinverse robustness*. *Systems, Man and Cybernetics, IEEE Transactions on*, vol. 22, no. 4, pages 790–798, 1992.
- [Mbakop 2020] Steeve Mbakop, Gilles Tagne, Othman Lakhal, Rochdi Merzouki et Sergey V Drakunov. *Path planning and control of mobile soft manipulators with obstacle avoidance*. In *2020 3rd IEEE International Conference on Soft Robotics (RoboSoft)*, pages 64–69. IEEE, 2020.

- [Mbede 2012] Jean Bosco Mbede, Achille Melingui, Bernard Essimbi Zobo, Rochdi Merzouki et Belkacem Ould Bouamama. *zSlices based type-2 fuzzy motion control for autonomous robotino mobile robot*. In Proceedings of 2012 IEEE/ASME 8th IEEE/ASME International Conference on Mechatronic and Embedded Systems and Applications, pages 63–68. IEEE, 2012.
- [McMahan 2004] William McMahan, Bryan Jones, Ian Walker, Vilas Chitrakaran, Arjun Seshadri et Darren Dawson. *Robotic manipulators inspired by cephalopod limbs*. Proceedings of the Canadian Engineering Education Association (CEEA), 2004.
- [McMahan 2006] William McMahan, V Chitrakaran, M Csencsits, D Dawson, Ian D Walker, Bryan A Jones, M Pritts, D Dienno, M Grissom et Christopher D Rahn. *Field trials and testing of the OctArm continuum manipulator*. In Robotics and Automation, 2006. ICRA 2006. Proceedings 2006 IEEE International Conference on, pages 2336–2341. IEEE, Orlando, FL USA , 15-19 May 2006.
- [Melin 2013] Patricia Melin, Leslie Astudillo, Oscar Castillo, Fevrier Valdez et Mario Garcia. *Optimal design of type-2 and type-1 fuzzy tracking controllers for autonomous mobile robots under perturbed torques using a new chemical optimization paradigm*. Expert Systems with Applications, vol. 40, no. 8, pages 3185–3195, 2013.
- [Melingui 2013a] Achille Melingui, Taha Chettibi, Rochdi Merzouki et Jean Bosco Mbede. *Adaptive navigation of an omni-drive autonomous mobile robot in unstructured dynamic environments*. In 2013 IEEE International Conference on Robotics and Biomimetics (ROBIO), pages 1924–1929. IEEE, 2013.
- [Melingui 2013b] Achille Melingui, Coralie Escande, Benoudjit Nabil, Rochdi Merzouki et JB Mbede. *Qualitative approach for forward kinematic modeling of a Compact Bionic Handling Assistant trunk*. accepted to 19th World Congress of the Int. Federation of Aut. Control, Cape Town, South Africa, 24-29 August 2014, 2013.
- [Melingui 2014a] A Melingui, R Merzouki et JB Mbede. *Compact bionic handling arm control using neural networks*. Electronics Letters, vol. 50, no. 14, pages 979–981, 2014.

- [Melingui 2014b] A Melingui, R Merzouki, JB Mbede, C Escande, B Daachi et N Benoudjit. *Qualitative approach for inverse kinematics modeling of a Compact Bionic Handling Assistant trunk*. accepted at 2014 IEEE WCCI : International Joint Conference on Neural Networks, Beijing, China, 6-11 July 2014.
- [Melingui 2014c] Achille Melingui, Coralie Escande, Nabil Benoudjit, Rochdi Merzouki et Jean Bosco Mbede. *Qualitative approach for forward kinematic modeling of a compact bionic handling assistant trunk*. IFAC Proceedings Volumes, vol. 47, no. 3, pages 9353–9358, 2014.
- [Melingui 2014d] Achille Melingui, Rochdi Merzouki, Jean Bosco Mbede et Taha Chetibi. *A novel approach to integrate artificial potential field and fuzzy logic into a common framework for robots autonomous navigation*. Proceedings of the Institution of Mechanical Engineers, Part I : Journal of Systems and Control Engineering, vol. 228, no. 10, pages 787–801, 2014.
- [Melingui 2017] Achille Melingui, Joseph Jean-Baptiste Mvogo Ahanda, Othman Lakhall, Jean Bosco Mbede et Rochdi Merzouki. *Adaptive algorithms for performance improvement of a class of continuum manipulators*. IEEE Transactions on Systems, Man, and Cybernetics : Systems, vol. 48, no. 9, pages 1531–1541, 2017.
- [Mendel 2006] Jerry M Mendel, Hani Hagrass et Robert I John. *Standard background material about interval type-2 fuzzy logic systems that can be used by all authors*, 2006.
- [Mendel 2017] Jerry M Mendel. *Uncertain rule-based fuzzy systems*. In Introduction and new directions, page 684. Springer, 2017.
- [Mendel 2019] Jerry M Mendel. *Type-2 fuzzy sets as well as computing with words*. IEEE Computational Intelligence Magazine, vol. 14, no. 1, pages 82–95, 2019.
- [MéNdez 2013] Gerardo M MéNdez et Maria De Los Angeles HernáNdez. *Hybrid learning mechanism for interval A2-C1 type-2 non-singleton type-2 Takagi–Sugeno–Kang fuzzy logic systems*. Information Sciences, vol. 220, pages 149–169, 2013.

- [Michael 2011] Nathan Michael, Jonathan Fink et Vijay Kumar. *Cooperative manipulation and transportation with aerial robots*. *Autonomous Robots*, vol. 30, no. 1, pages 73–86, 2011.
- [Nagymáté 2018] Gergely Nagymáté et Rita M Kiss. *Application of OptiTrack motion capture systems in human movement analysis : A systematic literature review*. *Recent Innovations in Mechatronics*, vol. 5, no. 1., pages 1–9, 2018.
- [Nguyen 1990] Derrick H Nguyen et Bernard Widrow. *Neural networks for self-learning control systems*. *Control Systems Magazine, IEEE*, vol. 10, no. 3, pages 18–23, 1990.
- [Oltean 2010] SE Oltean, M Dulău et R Puskas. *Position control of Robotino mobile robot using fuzzy logic*. In *2010 IEEE International Conference on Automation, Quality and Testing, Robotics (AQTR)*, volume 1, pages 1–6. IEEE, 2010.
- [Pandey 2017] Anish Pandey, Shalini Pandey et Preeti Gupta. *Intelligent navigation and control of a mobile robot in static and dynamic environments using hybrid fuzzy architecture*. *International Journal of Autonomic Computing*, vol. 2, no. 3, pages 255–281, 2017.
- [Parhi 2005] Dayal R Parhi. *Navigation of mobile robots using a fuzzy logic controller*. *Journal of intelligent and robotic systems*, vol. 42, no. 3, pages 253–273, 2005.
- [Patle 2019] BK Patle, Anish Pandey, DRK Parhi, A Jagadeeshet al. *A review : On path planning strategies for navigation of mobile robot*. *Defence Technology*, 2019.
- [Plett 2003] Gregory L Plett. *Adaptive inverse control of linear and nonlinear systems using dynamic neural networks*. *Neural Networks, IEEE Transactions on*, vol. 14, no. 2, pages 360–376, 2003.
- [Pratihari 1999] Dilip Kumar Pratihari, Kalyanmoy Deb et Amitabha Ghosh. *A genetic-fuzzy approach for mobile robot navigation among moving obstacles*. *International Journal of Approximate Reasoning*, vol. 20, no. 2, pages 145–172, 1999.

- [Prechelt 1994] Lutz Prechelt *et al.* *Proben1 : A set of neural network benchmark problems and benchmarking rules.* 1994.
- [Pritts 2004] Michael B Pritts et Christopher D Rahn. *Design of an artificial muscle continuum robot.* In IEEE International Conference on Robotics and Automation, 2004. Proceedings. ICRA'04. 2004, volume 5, pages 4742–4746. IEEE, 2004.
- [Rolf 2012] Matthias Rolf et Jochen J Steil. *Constant curvature continuum kinematics as fast approximate model for the Bionic Handling Assistant.* In 2012 IEEE/RSJ International Conference on Intelligent Robots and Systems, pages 3440–3446. IEEE, 2012.
- [Sánchez-Fernández 2004] Matilde Sánchez-Fernández, Mario de Prado-Cumplido, Jerónimo Arenas-García et Fernando Pérez-Cruz. *SVM multiregression for nonlinear channel estimation in multiple-input multiple-output systems.* IEEE transactions on signal processing, vol. 52, no. 8, pages 2298–2307, 2004.
- [Sanchez 2015] Mauricio A Sanchez, Oscar Castillo et Juan R Castro. *Generalized type-2 fuzzy systems for controlling a mobile robot and a performance comparison with interval type-2 and type-1 fuzzy systems.* Expert Systems with Applications, vol. 42, no. 14, pages 5904–5914, 2015.
- [Shi 2004] XH Shi, YC Liang, HP Lee, WZ Lin, X Xu et SP Lim. *Improved Elman networks and applications for controlling ultrasonic motors.* Applied Artificial Intelligence, vol. 18, no. 7, pages 603–629, 2004.
- [Singh 2018] Inderjeet Singh, Yacine Amara, Achille Melingui, Pushparaj Mani Pathak et Rochdi Merzouki. *Modeling of continuum manipulators using pythagorean hodograph curves.* Soft robotics, vol. 5, no. 4, pages 425–442, 2018.
- [Smola 2004] Alex J Smola et Bernhard Schölkopf. *A tutorial on support vector regression.* Statistics and computing, vol. 14, no. 3, pages 199–222, Statistics and computing, Springer2004.

- [Specht 1990] Donald F Specht. *Probabilistic neural networks*. Neural networks, vol. 3, no. 1, pages 109–118, 1990.
- [Subbash 2019] Panati Subbash et Kil To Chong. *Adaptive network fuzzy inference system based navigation controller for mobile robot*. Frontiers of Information Technology & Electronic Engineering, vol. 20, no. 2, pages 141–151, 2019.
- [Surmann 1995] Hartmut Surmann, Joerg Huser et Liliane Peters. *A fuzzy system for indoor mobile robot navigation*. In Proceedings of 1995 IEEE International Conference on Fuzzy Systems., volume 5, pages 71–76. IEEE, 1995.
- [Taylor 1990] Richard Taylor. *Interpretation of the correlation coefficient : a basic review*. Journal of diagnostic medical sonography, vol. 6, no. 1, pages 35–39, 1990.
- [Thuruthel 2017] Thomas George Thuruthel, Egidio Falotico, Federico Renda et Cecilia Laschi. *Learning dynamic models for open loop predictive control of soft robotic manipulators*. Bioinspiration & biomimetics, vol. 12, no. 6, page 066003, 2017.
- [Thuruthel 2018] Thomas George Thuruthel, Egidio Falotico, Federico Renda et Cecilia Laschi. *Model-based reinforcement learning for closed-loop dynamic control of soft robotic manipulators*. IEEE Transactions on Robotics, vol. 35, no. 1, pages 124–134, 2018.
- [Tzafestas 1996] S Tzafestas, A Zagorianos et T Pimenides. *A solution to the velocity control of redundant robots*. Mathematics and computers in simulation, vol. 41, no. 3, pages 201–208, 1996.
- [Vapnik 2013] Vladimir Vapnik. The nature of statistical learning theory. Springer science & business media, 2013.
- [Walker 2012] Ian D Walker, Ryan Mattfeld, Alper Mutlu, Alan Bartow et Nivedhitha Giri. *A novel approach to robotic climbing using continuum appendages in in-situ exploration*. In 2012 IEEE Aerospace Conference, pages 1–9. IEEE, 2012.
- [Walker 2013] Ian D Walker. *Continuous backbone continuum robot manipulators*. International Scholarly Research Notices, vol. 2013, 2013.

- [Wang 2004] Chi-Hsu Wang, Chun-Sheng Cheng et Tsu-Tian Lee. *Dynamical optimal training for interval type-2 fuzzy neural network (T2FNN)*. IEEE Transactions on Systems, Man, and Cybernetics, Part B (Cybernetics), vol. 34, no. 3, pages 1462–1477, 2004.
- [Webster III 2009] Robert J Webster III, Joseph M Romano et Noah J Cowan. *Mechanics of precurved-tube continuum robots*. IEEE Transactions on Robotics, vol. 25, no. 1, pages 67–78, 2009.
- [Webster III 2010] Robert J Webster III et Bryan A Jones. *Design and kinematic modeling of constant curvature continuum robots : A review*. The International Journal of Robotics Research, vol. 29, no. 13, pages 1661–1683, 2010.
- [Webster 2006] Robert J Webster, Allison M Okamura et Nah J Cowan. *Toward active cannulas : Miniature snake-like surgical robots*. In 2006 IEEE/RSJ international conference on intelligent robots and systems, pages 2857–2863. IEEE, 2006.
- [Wu 2006] Wei Wu, Hongmei Shao et Zhengxue Li. *Convergence of batch BP algorithm with penalty for FNN training*. In Neural Information Processing, pages 562–569. Springer, 2006.
- [Yam 1999] Yeung Yam, Péter Baranyi et Chi-Tin Yang. *Reduction of fuzzy rule base via singular value decomposition*. IEEE Transactions on fuzzy Systems, vol. 7, no. 2, pages 120–132, 1999.
- [Yi 2019] Zeren Yi, Guojin Li, Shuang Chen, Wei Xie et Bugong Xu. *A navigation method for mobile robots using interval type-2 fuzzy neural network fitting Q-learning in unknown environments*. Journal of Intelligent & Fuzzy Systems, vol. 37, no. 1, pages 1113–1121, 2019.
- [Yim 2003] Mark Yim, Kimon Roufas, David Duff, Ying Zhang, Craig Eldershaw et Sam Homans. *Modular reconfigurable robots in space applications*. Autonomous Robots, vol. 14, no. 2, pages 225–237, 2003.