

UNIVERSITÉ DU QUÉBEC À TROIS-RIVIÈRES

OPTIMISATION ET COMMANDE AVANCÉES POUR LE SUIVI DES
TRAJECTOIRES D'UN ROBOT MOBILE DE TYPE VOITURE

MÉMOIRE PRÉSENTÉ COMME EXIGENCE PARTIELLE DE LA MAÎTRISE EN
INGÉNIERIE CONCENTRATION GÉNIE MÉCANIQUE

PAR
Joslin Numbi

Mai 2025

UNIVERSITÉ DU QUÉBEC À TROIS-RIVIÈRES

MAÎTRISE EN INGÉNIERIE CONCENTRATION GÉNIE MÉCANIQUE (M. Sc. A.)

Direction de recherche :

Nadjet Zioui

Prénom Nom

Directrice de recherche

Jury d'évaluation

Martin Bolduc, Université du Québec à Trois-Rivières

Prénom Nom, Nom établissement

Évaluateur interne

Dermouche Reda, École nationale supérieure de technologie

Prénom Nom, Nom établissement

Évaluateur externe

Nadjet Zioui, Université du Québec à Trois-Rivières

Prénom Nom, Nom établissement

Directrice de recherche

Université du Québec à Trois-Rivières

Service de la bibliothèque

Avertissement

L'auteur de ce mémoire ou de cette thèse a autorisé l'université du Québec à Trois-Rivières à diffuser à des fins non lucratives, une copie de son mémoire ou de sa thèse.

Cette diffusion n'entraîne pas une renonciation de la part de l'auteur à ses droits de propriété intellectuelle, incluant le droit d'auteur, sur ce mémoire ou cette thèse. Notamment, la reproduction ou la publication de la totalité ou d'une partie importante de ce mémoire ou de cette thèse requiert son autorisation.

Résumé

Les robots mobiles sont de plus en plus utilisés dans l'industrie pour diverses tâches telles que la fabrication, l'entreposage et la logistique. Toutefois, certains problèmes se posent notamment les difficultés des robots à suivre les trajectoires ; ces aspects jouent un rôle crucial dans l'efficacité et la précision des robots lors de leurs déplacements. Dans cette étude, des algorithmes avancés pour l'amélioration du suivi des trajectoires sont développés. Tout d'abord, un réseau de neurones artificiels est construit dans le but d'identifier les paramètres du robot afin d'améliorer la précision de son modèle dynamique. Ensuite, le modèle dynamique du robot est construit, puis simulé en mettant l'accent sur l'amélioration de ses capacités de suivi des trajectoires. Pour ce faire, on utilise l'algorithme PSO classique et sa version quantique (QPSO) pour corriger les erreurs de suivi. L'objectif est de comparer les performances de ces deux algorithmes permettant d'atteindre les valeurs optimales des constantes du régulateur PD. Les taux de convergence et la stabilité de ces deux algorithmes de commande sont analysés au fil des itérations afin d'examiner la contribution de la composante quantique à la performance de son algorithme. Pour finir, un protocole de téléportation quantique pour l'échange de données entre le robot mobile et sa station de commande a été proposé pour simuler la téléportation de certaines données. Les circuits quantiques étant coûteux à construire et à entretenir, le logiciel MATLAB académique a été utilisé pour simuler ces algorithmes.

Dédicace

Je dédie ce travail de recherche à :

Ma mère et mon père

Mes sœurs et mes frères

Tous ceux qui, de près ou de loin, ont contribué à la réalisation de ce travail.

Joslin

Remerciements

Je tiens à remercier tout spécialement ma directrice de recherche, la professeure Nadjet Zioui, pour l'opportunité qui m'a été donnée de travailler au sein de son équipe au Laboratoire de L'Intelligence Artificielle Appliquée (LI2A) de l'UQTR et pour son encadrement et ses précieux conseils qui ont été indispensables pour la réalisation de ce mémoire.

Je remercie également le Conseil de Recherches en Sciences Naturelles et en Génie du Canada (CRSNG) qui a financé cette recherche.

Une pensée particulière à tous les membres de ma famille ainsi qu'à toutes les personnes qui, de près ou de loin, m'ont soutenu tout au long de mon cheminement.

Pour finir, mes remerciements vont à mes cher.e.s ami.e.s Dina Yav, Angy Ndas, Mignon Omombo et Karel Kishiko pour le soutien et les encouragements qu'ils ne cessent de m'apporter.

Table des matières

Résumé.....	iii
Dédicace.....	iv
Remerciements.....	v
Table des matières	vi
Liste des tableaux.....	x
Liste des figures	xi
Liste d'abréviations et de symboles	xiii
Liste des notations et des acronymes	xiii
Liste des symboles	xiii
1 Introduction.....	1
1.1 Contexte général de la robotique mobile à roues	1
1.2 Problématique	2
1.3 Objectifs	Erreur ! Signet non défini.
1.4 Méthodologie	4
1.5 Organisation du mémoire	6
Chapitre 1 - État de l'art.....	7
Chapitre 2 - Conception d'un ANN classique et de sa version quantique pour le suivi des trajectoires d'un robot mobile de type voiture	12
2.1 Architecture d'un ANN	12
2.1.1 Les fonctions d'activation classiques	13
2.1.1.1 Fonction d'activation sigmoïde	13
2.1.1.2 Fonction d'activation linéaire	14
2.1.1.3 Fonction d'activation de rectification linéaire.....	14
2.1.2 Fonctions d'activation inspirées du calcul quantique	15

2.1.2.1 Fonction d'activation sigmoïde inspirée du calcul quantique	16
2.1.2.2 Fonction d'activation de rectification linéaire inspirée du calcul quantique	17
2.1.2.3 Fonction d'activation linéaire inspirée du calcul quantique.....	18
2.2 Conception et entraînement d'un ANN inspiré du calcul quantique	19
Chapitre 3 - Optimisation des paramètres du régulateur PD basée sur les versions classique et quantique du PSO pour le suivi des trajectoires d'un robot mobile de type voiture	23
3.1 Modèle mathématique du robot mobile de type voiture	23
3.2 Optimisation par essaim de particules (PSO).....	31
3.3 Optimisation de l'essaim particulaire inspirée du calcul quantique (QPSO)	34
Chapitre 4 - Téléportation quantique pour la commande à distance d'un robot mobile de type voiture	36
4.1 Travaux connexes.....	36
4.2 Notions de téléportation quantique	38
4.3 Architecture du système	43
4.3.1 Station de commande.....	43
4.3.2 Robot mobile de type voiture.....	45
4.3.3 Circuit de téléportation quantique	46
Chapitre 5 – Résultats et discussion.....	50
5.1 ANN et QANN pour l' optimisation des paramètres en vue d'un suivi des trajectoires d'un robot mobile de type voiture	50
5.1.1 Paramètres de simulation.....	50
5.1.2 Résultats et discussion.....	51
5.1.2.1 Base de données pour l'apprentissage	51
5.1.2.2 Base de données pour la validation.....	54

5.2 Optimisation des paramètres du régulateur PD basée sur la version quantique du PSO pour le suivi des trajectoires d'un robot mobile de type voiture	55
5.2.1 Paramètres de simulation.....	55
5.2.2 Résultats sans algorithme de commande	57
5.2.3 Résultats et discussion sur l'utilisation du régulateur PD basé sur PSO et QPSO pour le suivi d'une trajectoire circulaire.....	59
5.2.3.1 Temps de convergence du PSO et du QPSO au fil des itérations	60
5.2.3.2 Tests de robustesse du PSO et du QPSO face aux changements environnementaux.....	61
5.2.3.3 Les performances du PSO et du QPSO à réduire le RMSE.....	62
5.2.4 Résultats et discussion sur l'utilisation du régulateur PD basé sur PSO et QPSO pour le suivi d'une trajectoire sinusoïdale	64
5.2.4.1 Temps de convergence du PSO et du QPSO au fil des itérations	64
5.2.4.2 Tests de robustesse du PSO et du QPSO face aux changements environnementaux.....	65
5.2.4.3 Les performances du PSO et du QPSO à réduire le RMSE.....	67
5.2.5 Résultats et discussion sur l'utilisation du régulateur PD basé sur PSO et QPSO pour le suivi d'une trajectoire linéaire	68
5.2.5.1 Temps de convergence de PSO et de QPSO au fil des itérations.....	68
5.2.5.2 Tests de robustesse du PSO et du QPSO face aux changements environnementaux.....	70
5.2.5.3 Les performances du PSO et du QPSO à réduire le RMSE.....	72
5.3 Téléportation quantique pour la commande à distance d'un robot mobile de type voiture.....	74
5.3.1 Téléportation des données quantiques pour la trajectoire circulaire	74
5.3.1.1 Données sur les vitesses longitudinale, latérale et angulaire	74
5.3.1.2 Données sur les positions par rapport aux axes des x, des y et des z.....	76

5.3.2 Téléportation des données quantiques pour la trajectoire sinusoïdale.....	77
5.3.2.1 Données sur les vitesses longitudinale, latérale et angulaire	77
5.3.2.2 Données sur la position par rapport aux axes des x, des y et des z.....	79
Chapitre 6 – Conclusion.....	81
References.....	83
Publications.....	93

Liste des tableaux

Tableau 5.1. Les paramètres de l'architecture de l'ANN.....	50
Tableau 5.2 : Les paramètres du robot CLMR.....	55
Tableau 5.3 : Les paramètres des PSO/QPSO dans des conditions idéales.....	56
Tableau 5.4 : Les paramètres des PSO/QPSO dans des conditions non idéales.....	56

Liste des figures

Figure 1.1 Déviation d'un robot mobile de la trajectoire prédéfinie.....	3
Figure 2.1 Réseau de neurones artificiels.....	12
Figure 2.2 Comparaison de la fonction sigmoïde classique et celle quantique.....	16
Figure 2.3 Comparaison de la fonction Relu classique et celle quantique.....	17
Figure 2.4 Comparaison de la fonction linéaire classique et celle quantique.....	19
Figure 3.1 Cinématique du robot mobile sur une surface plane en 2D.....	24
Figure 3.2 Dynamique du robot mobile sur une surface plane en 2D.....	28
Figure 4.1 Circuit de téléportation quantique (Image réalisée avec Quirk).....	39
Figure 4.2 Schéma de l'architecture du système.	43
Figure 5.1 l'évolution au fil des itérations des fonctions de coût pour : (a) les masses et (b) les moments d'inertie.....	51
Figure 5.2 Comparaison entre les données des masses réelles et celles prédites par : (a) QANN et (b) CANN.....	52
Figure 5.3 Comparaison entre les données sur les moments d'inertie réelles et celles prédites par : (a) QANN et (b) CANN.....	53
Figure 5.4 Comparaison entre les données réelles et celles prédites pour : (a) les masses et (b) les moments d'inertie.....	54
Figure 5.5 Trajectoire du robot avant optimisation.....	57
Figure 5.6 Évolution du RMSE avant optimisation.....	57
Figure 5.7 Forces et couple appliqués au robot au cours du temps pour la trajectoire circulaire.....	58
Figure 5.8 Évolution de K_p et K_d au fil des itérations pour la trajectoire circulaire.....	60

Figure 5.9 Forces et moment d'inertie appliqués au robot pour la trajectoire circulaire....	61
Figure 5.10 Angle de braquage du robot au cours du temps pour la trajectoire circulaire...	62
Figure 5.11 Évolution du RMSE du robot au fil du temps pour la trajectoire circulaire...	63
Figure 5.12 Le suivi de la trajectoire circulaire du robot au cours du temps.....	63
Figure 5.13 Évolution de K_p et K_d au fil des itérations pour la trajectoire sinusoïdale...	64
Figure 5.14 Application des forces et du moment d'inertie au robot au cours du temps pour la trajectoire sinusoïdale.....	65
Figure 5.15 Angle de braquage du robot au fil du temps pour la trajectoire sinusoïdale...	65
Figure 5.16 Évolution du RMSE du robot au fil du temps pour la trajectoire sinusoïdale..	67
Figure 5.17 Le suivi de la trajectoire sinusoïdale du robot au cours du temps.....	68
Figure 5.18 Évolution de K_p et K_d au fil des itérations pour la trajectoire en ligne droite...	69
Figure 5.19 Corrélation entre différents nombres de particules, itérations et RMSE pour : (a) QPSO et (b) PSO.....	70
Figure 5.20 Forces et moment d'inertie appliqués au robot au fil du temps pour la trajectoire linéaire.....	71
Figure 5.21 Angle de braquage du robot au fil du temps pour la trajectoire linéaire.....	72
Figure 5.22 Évolution du RMSE du robot au fil du temps pour la trajectoire linéaire.....	73
Figure 5.23 Trajectoire linéaire du robot au fil du temps.....	73
Figure 5.24 Comparaison entre les données réelles et celles téléportées des vitesses : (a) longitudinale, (b) latérale et (c) angulaire.....	75
Figure 5.25 Positions x, y et z pour la trajectoire circulaire : (a) position x, (b) position y, (b) position z.....	76
Figure 5.26 Comparaison entre les données réelles et celles téléportées des vitesses : (a) longitudinale, (b) latérale et (c) angulaire.....	77
Figure 5.27 Positions x, y et z pour la trajectoire sinusoïdale.....	79

Liste d'abréviations et de symboles

Liste des notations et des acronymes

QANN	Réseau neuronal artificiel quantique (Quantum artificial neural network) ;
CANN	Réseau neuronal artificiel classique (Classical artificial neural network) ;
ML	Apprentissage machine (Machine learning) ;
QPSO	Optimisation des essaims de particules inspirée du calcul quantique (Quantum-inspired particle swarm optimization) ;
PSO	Optimisation des essaims de particules (Classical particle swarm optimization) ;
RMSE	Erreur quadratique moyenne (Root mean squared error) ;
PD	Proportionnel Dérivé (Proportional-Derivative) ;
LiDAR	Détection et télémétrie par la lumière (Light detection and ranging) ;
UAV	Véhicules aériens sans pilote (Unmanned aerial vehicles) ;
CLMR	Robot mobile de type voiture (Car-like mobile robot).

Liste des symboles

K_p	Composante proportionnelle (Proportional component) ;
K_d	Composante dérivée (Derivative component) ;
X	Vecteur d'entrée ;
W^1	Vecteur de poids de la couche d'entrée ;

- W^2 Vecteur de poids de la couche intermédiaire ;
- b^1 Biais de la couche d'entrée ;
- b^2 Biais de la couche intermédiaire ;
- $Z_{1,2}$ Vecteur d'entrée de la fonction d'activation de la couche intermédiaire ;
- f_1 Sorties de la fonction d'activation à chaque nœud de la couche intermédiaire ;
- f_2 Sorties de la fonction d'activation à chaque nœud de la couche de sortie ;
- $|\Psi\rangle$ Bit quantique (Quantum bit) ;
- θ Amplitudes de probabilité des composantes d'état $|0\rangle$ et $|1\rangle$;
- ϕ Phase relative entre les composantes d'état $|0\rangle$ et $|1\rangle$;
- $x(t)$ Déplacements du robot dans l'axe des abscisses ;
- $y(t)$ Déplacements du robot dans l'axe des ordonnées ;
- $z(t)$ Angle entre la direction du robot et l'axe des abscisses ;
- u Vitesse longitudinale ;
- v Vitesse latérale ;
- r Vitesse angulaire ;
- ζ Vecteur vitesse du CLMR ;
- J^{-1} Inverse de la matrice jacobienne $J(\theta)$;
- l Longueur du CLMR ;
- r Vitesse angulaire ;
- $R_{u,v}$ Résultante des vitesses longitudinale et latérale ;

- $\dot{\zeta}_n$ Vecteur accélération nonholonomique du CLMR;
- \dot{u}_n Composante longitudinale de l'accélération nonholonomique;
- \dot{v}_n Composante latérale de l'accélération nonholonomique ;
- \dot{r}_n Composante angulaire de l'accélération nonholonomique ;
- W Matrice nonholonomique de l'état ;
- $R_{\dot{u}, \dot{v}, \dot{r}}$ Résultante des \dot{u}_n , \dot{v}_n et \dot{r}_n ;
- KE Énergie cinétique;
- PE Énergie potentielle;
- m Masse du robot ;
- \dot{X}_C Vitesse du centre d'inertie par rapport à l'axe des abscisses ;
- \dot{Y}_C Vitesse du centre d'inertie par rapport à l'axe des ordonnées ;
- I_{CZ} Moment d'inertie par rapport à l'axe z ;
- τ Vecteur couple du CLMR;
- D Matrice dynamique du CLMR;
- $n(\zeta)$ Autres effets de l'environnement sur le CLMR ;
- Vf Coefficient de frottement visqueux ;
- V Vitesses des particules ;
- P Positions des particules ;
- C_1 Poids cognitif ou personnel ;
- C_2 Poids social ;
- I Matrice identitaire.
- X Porte de Pauli-X ;
- Z Porte de Pauli-Z.

1 Introduction

1.1 Contexte général de la robotique mobile à roues

Les robots à roues sont des dispositifs automatiques ou semi-automatiques qui se déplacent à l'aide des roues pour remplir des tâches bien déterminées [1]. Pour un fonctionnement effectif des robots, plusieurs disciplines entrent en jeu. La mécanique s'occupe de la conception du châssis et des parties qui interagissent avec l'environnement extérieur du robot [2] ; l'électronique, quant à elle, assure l'alimentation en énergie du robot et la conception des circuits qui permettent la bonne connexion entre le régulateur, le moteur et les capteurs [3] ; l'informatique entre en jeu pour le développement des algorithmes, le traitement des données et le développement de l'interface permettant la communication entre l'utilisateur et la machine [4] ; la dernière discipline, l'automatique se charge de la commande, la modélisation et la simulation du robot dans son environnement afin d'avoir une planification et une prise de décision pour réduire l'effet des contraintes environnementales sur les mouvements du robot [5].

Les robots mobiles sont de plus en plus utilisés dans divers secteurs de l'industrie. Leurs capacités à naviguer dans des environnements dynamiques avec facilité et précision les rendent indispensables pour des tâches telles que la logistique d'entrepôt, la construction ou tout autre travail jugé dangereux pour l'homme. En fonction des différents critères tels que le type d'environnement, la stabilité et la maniabilité du robot, ceux-ci peuvent être classés sous différents groupes, à savoir : robots à roues, robots à chenilles, robots volants, robots sous-marins, etc. Tenant compte de certains critères qui seront développés par la suite, le robot mobile de type voiture a été sélectionné pour cette étude.

1.2 Problématique

Les robots mobiles de type voiture ont plusieurs avantages, tels que la stabilité grâce aux quatre roues équitablement réparties autour du châssis ; cette disposition des roues permet au robot de se déplacer de manière rectiligne, ce qui est essentiel pour un suivi des trajectoires prédéfinies [6]. Les robots de type voiture sont dotés de mécanismes de direction qui leur permettent de manœuvrer et de changer facilement de direction.

Malgré tous les avantages offerts par les robots de type voiture, l'optimisation et la commande de leurs paramètres présentent des défis considérables, dus à diverses raisons liées aux techniques qui interviennent dans la robotique. En outre, les contraintes mécaniques de l'environnement sur robot, le coefficient de friction qui dépend de plusieurs facteurs, à savoir la surface de contact des roues, la nature du terrain, la vitesse du robot et les conditions environnementales [2] ; cela peut entraîner des échanges de données biaisées dues à une mauvaise calibration ou une défaillance des moteurs [7], les perturbations environnementales qui créent des interférences dans les capteurs ; un algorithme de fusion erroné, des problèmes de feedback qui ne permettent pas la correction de la trajectoire du robot, ou le temps de réaction inadéquat du robot lors de la réception et l'envoi des données entre le robot et sa station de commande.

L'une où la combinaison de certaines de ces anomalies cause la déviation du robot mobile de la trajectoire prédéfinie ; cette déviation est généralement proportionnelle à la pertinence des anomalies. La Figure 1.1 illustre la déviation d'un robot mobile de la trajectoire prédéfinie.

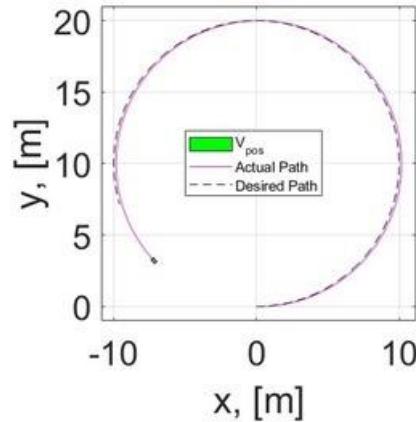


Figure 1.1 : Déviation d'un robot mobile de la trajectoire prédéfinie.

Le robot mobile dans la figure 1.1 dévie de sa trajectoire prédéfinie tout en extrapolant la vitesse appliquée à ses roues. Cela est visible, car au moment t de la simulation, la position réelle du robot est beaucoup plus avancée par rapport à la position désirée.

Afin de pallier ce genre d'anomalies, plusieurs techniques ont été développées par les chercheurs à travers le monde. Cependant, ces techniques réduisent l'erreur de suivi tout en le stabilisant un tant soit peu, ce qui souligne la pertinence de continuer la recherche en vue d'une optimisation plus avancée.

1.3 Objectifs

L'objectif de ce travail est de proposer des techniques d'optimisation avancées pour un fonctionnement optimal et une amélioration du suivi des trajectoires d'un robot mobile de type voiture. Pour ce faire, ce travail a été subdivisé en quatre grandes parties qui sont :

1. La construction d'un réseau de neurones artificiels dans le but d'identifier les paramètres du robot afin d'améliorer la précision du modèle dynamique du robot.

2. Le développement de l'algorithme d'optimisation de l'essaim particulaire qui permet d'adapter les paramètres du régulateur proportionnel dérivé (PD) pour minimiser l'erreur de poursuite liée au modèle dynamique du robot.
3. Le développement d'une version quantique de l'algorithme d'optimisation de l'essaim particulaire et d'une autre d'un réseau de neurones artificiels afin de diminuer le temps de traitement important dû au volume considérable des données.
4. La proposition d'une méthode de communication sans fil basée sur du calcul quantique ; cela permet d'assurer la communication entre le robot mobile et sa station de commande.

1.4 Méthodologie

Afin de répondre à la problématique soulevée, une méthodologie subdivisée en quatre étapes a été utilisée :

La première étape était d'approfondir les connaissances tout en comparant les différentes méthodes utilisées à travers le monde dans le but d'optimiser et de commander efficacement les mouvements des robots mobiles.

Les connaissances acquises lors de la revue de littérature permettent de proposer la construction d'ANN pour identifier les paramètres du modèle [8], à savoir sa masse et son moment d'inertie. Ces paramètres à valeur fixe peuvent fluctuer dans certaines conditions de vitesse élevée ou de vibrations, telles que celles causées par les moteurs ou la conduite sur des surfaces irrégulières. La dilatation thermique peut entraîner des changements dans les propriétés des matériaux du robot, ce qui peut légèrement modifier sa masse et son

moment d'inertie. Une estimation précise de ces paramètres est donc essentielle pour une optimisation avancée afin de garantir un suivi parfait de trajectoires données.

L'usage des ANNs permet d'acquérir avec précision les paramètres du modèle dynamique du robot. Cependant, d'autres types d'erreurs sont aussi à considérer, par exemple des données potentiellement biaisées provenant des différents capteurs, ou encore un temps de réaction inadéquat dans la prise de décisions du robot. Pour pallier ce genre d'erreurs, la troisième étape consiste à développer un algorithme de commande adaptatif et robuste qui permet d'optimiser les paramètres du régulateur PD afin de minimiser et de stabiliser l'erreur de poursuite [9].

Les deux algorithmes proposés nécessitent une quantité considérable de données pour une optimisation avancée des paramètres. Cela peut impliquer un prolongement significatif du temps de traitement des données ; des calculs plus avancés tels que le calcul quantique connu pour l'accélération exponentielle du temps de traitement pourraient être une solution à ce problème. À cet effet, les versions quantiques des algorithmes d'optimisation des essaims de particules (PSO) et CANN ont été proposées et développées. Les robots mobiles nécessitent une communication sans fil et en temps réel [10]. C'est pourquoi, un autre algorithme basé sur le calcul quantique a été proposé afin d'assurer la communication bidirectionnelle entre le robot mobile de type voiture (CLMR) et sa station de commande.

Pour finir, les analyses des résultats portant sur les performances des versions classiques et quantiques de l'ANN, du PSO et de la téléportation de données entre le robot et sa station de commande ont été menées.

1.5 Organisation du mémoire

La structure de ce travail est organisée en six chapitres :

L'État de l'art constitue le premier chapitre de ce mémoire. Il donne une vision générale de ce qui se fait en robotique mobile à roues dans le but d'optimiser et de commander les mouvements des robots mobiles.

Dans le deuxième chapitre, des versions classique et quantique d'un ANN ont été construites, puis entraînées à prédire la masse et le moment d'inertie des CLMRs, en tenant compte des erreurs quadratiques moyennes (RMSE) entre les trajectoires souhaitées et celles suivies par les robots.

Le chapitre trois porte sur l'utilisation des versions classique et quantique du PSO dans le but de trouver les paramètres optimaux d'un régulateur PD. Cela a permis de réduire et de stabiliser les erreurs de suivi dues aux effets de l'environnement dynamique du robot.

Le quatrième chapitre décrit la construction d'un protocole de téléportation quantique pour l'échange de données telles que la vitesse et la position, sous forme d'informations quantiques, entre le robot et sa station de commande.

Le cinquième chapitre est dédié à la validation expérimentale qui se présente sous forme d'analyses poussées des résultats obtenus après l'application des méthodes choisies.

Pour finir, le dernier chapitre est consacré à une conclusion portant sur les différentes méthodes et algorithmes utilisés dans les cinq premiers chapitres.

Chapitre 1 - État de l'art

De nombreuses études ont été menées à travers le monde afin d'améliorer les performances des robots mobiles à roues. Ces études sont plus particulièrement basées sur la capacité des CLMRs à suivre des trajectoires données. Pour ce faire, une revue de littérature sur les principales méthodes développées par les chercheurs a été réalisée.

Wang et al. [11] ont incorporé des notions d'apprentissage par renforcement profond basées sur des méthodes de planification des trajectoires et de commande des mouvements de bout en bout. Des résultats satisfaisants ont été obtenus après la modélisation physique du robot et l'optimisation des données obtenues par détection et télémétrie par la lumière (LiDAR) ; pour cela, une méthode qui traite avec précision les distances relatives entre les obstacles et le robot a été utilisée, améliorant ainsi la précision et l'adaptabilité, tout en évitant les obstacles. Mutti et al. [12] ont réduit le temps, la puissance de calcul et la quantité de données réelles nécessaires en vue d'obtenir un modèle précis du robot mobile en développant un algorithme basé sur un réseau neuronal récurrent. Ils l'ont fait en entraînant le modèle sur des données LiDAR synthétiques et complémentaires pour intégrer des transformations rigides dans le modèle d'apprentissage profond, puis en affinant le modèle sur des positions réelles à l'aide des données réelles et des mesures de caméras externes pour produire des étiquettes d'entraînement. Hohenfeld et al. [13] ont quant à eux démontré que les circuits quantiques, dans les configurations d'apprentissage par renforcement quantique-classique, peuvent apprendre de manière optimale, facilitant la navigation d'un robot à roues dans des environnements simulés avec beaucoup moins de paramètres par rapport à l'apprentissage par renforcement profond classique. Cependant, les résultats ont montré que les réseaux neuronaux classiques obtenaient de meilleurs résultats concernant

les temps de formation et la stabilité. Certains scientifiques [14,15] ont également développé des algorithmes similaires pour les mêmes tâches de navigation des robots mobiles. Contrairement à l'article précédent [13], l'algorithme utilisé dans cet article [14] s'inspire du phénomène d'effondrement dans la mesure quantique et de l'amplification de l'amplitude dans le calcul quantique. Yun et al. [16] ont mis en œuvre l'apprentissage par renforcement multi-agent quantique pour améliorer la commande entre plusieurs robots connectés, tout en garantissant une bonne coordination dans leur environnement. Hu et al. [17] ont proposé un apprentissage Q hybride flou A* quantique multi-étape, qu'ils ont mélangé à un algorithme de champ potentiel artificiel pour améliorer la planification des trajectoires des robots mobiles dans des environnements remplis de pièges et de passages étroits. Des travaux intéressants, tels que celui de Spanogianopoulos [18], ont permis d'atteindre un niveau élevé de robustesse en utilisant un de ces algorithmes pour la localisation et le suivi optimal des trajectoires d'un robot mobile. D'autres scientifiques [19-21] se sont concentrés sur ces types d'algorithmes afin d'identifier les chemins les plus courts, les plus fluides et les plus sûrs pour les robots dans des environnements statiques et dynamiques. Les articles [22-23] assurent une planification optimale des trajectoires, tout en évitant les obstacles. Harmania et al. [24] ont développé trois robots E-Puck pour qu'ils naviguent sur le même terrain sans entrer en collision. C'était fait en utilisant le traitement d'images d'une caméra aérienne. Les résultats de l'expérience ont démontré que le système multirobot basé sur les simulations reflétait le comportement d'un système multirobot réel. Das et al. [25] ont conçu un algorithme d'optimisation d'essaims de particules amélioré (IPSO) pour la même raison ; leurs robots ont pris des décisions individuelles tout en se coordonnant et en travaillant les uns avec les autres pour accomplir des tâches partagées.

Zhang et al. [26] ont quant à eux, réussi à réduire la distance parcourue par le robot en vue de minimiser la perte en énergie. Le PSO classique offre des avantages indéniables. Cependant, certains défis restent à relever, tels que les limites de l'exploration globale, l'incapacité à gérer efficacement un grand nombre d'obstacles et la difficulté à trouver l'équilibre entre l'exploration et l'exploitation. Ces types de problèmes pourraient être résolus en intégrant certaines notions de calcul quantique dans les algorithmes PSO ; cela pourrait expliquer l'essor de l'application de l'informatique quantique à la robotique mobile. Des chercheurs [27] ont identifié les différents domaines d'application de l'informatique quantique pour améliorer l'efficacité et la précision des systèmes robotisés. L'informatique quantique peut être utilisée pour améliorer le suivi des trajectoires et la navigation des robots en augmentant la capacité d'exploration globale, la robustesse aux optima locaux et la vitesse de convergence des algorithmes PSO. Fernandes et al. [28] ont su éviter les minima locaux dans la planification des trajectoires de leur CLMR, tant dans des situations statiques que dans des situations dynamiques. Xue et al. [29] ont atteint le même objectif avec un algorithme d'optimisation des essaims de particules inspirée du calcul quantique (QPSO), sans pour autant prendre en compte les obstacles statiques. Ces algorithmes permettent de générer des trajectoires sûres et efficaces tout en réduisant le gaspillage d'énergie. Lan et al. [30] ont créé un IQPSO pour réduire la complexité des calculs et le temps de recherche de la meilleure trajectoire. Fernandes et al. [31] ont quant à eux mis au point un essaim de particules avec une population plus diversifiée, qui permet d'élargir l'espace de recherche. Certaines méthodes ont été développées pour minimiser de manière significative le nombre de particules nécessaires pour l'optimisation des trajectoires. Tokgo et al. [32] ont mis au point un système basé sur le calcul quantique qui

intègre la planification des trajectoires locales et globales afin d'améliorer la navigation du robot dans un environnement d'obstacles aléatoires. Dian et al. [33] ont conçu un algorithme PSO qui utilise une courbe continue de Bézier d'ordre élevé afin de prendre en compte la longueur des trajectoires ainsi que les conditions qui minimisent les collisions. Dong et al. [34] ont résolu la tendance du PSO à glisser vers les optimums locaux, en associant celui-ci à l'informatique quantique. Bien que ces articles fournissent des indications précieuses sur l'usage des algorithmes avancés de l'ANN et du PSO en robotique mobile, ils présentent certaines lacunes, comme la difficulté à éviter le surapprentissage des ANNs, l'insuffisance des comparaisons entre le PSO et le QPSO concernant leurs capacités à trouver les solutions optimales aux problèmes, le temps nécessaire pour converger vers les solutions et la stabilité des paramètres au fil des itérations. Pour résoudre ces problèmes, il faudrait poursuivre les recherches sur les techniques hybrides qui combinent les forces des deux approches.

Dans ce mémoire, une quantité considérable de données sur les masses, les moments d'inertie et les RMSE entre les trajectoires souhaitées et les trajectoires réelles des CLMRs a été prise en compte. Le but est d'examiner les performances des approches quantique et classique des réseaux de neurones en termes de valeurs optimales prédites pour la masse et le moment d'inertie du CLMR. Dans ce mémoire, ont également été utilisés les algorithmes PSO et QPSO pour corriger les erreurs de suivi des trajectoires d'un CLMR. L'objectif était d'examiner les performances de ces deux approches en termes de valeurs optimales pour les constantes du régulateur PD. Par la suite, leurs taux de convergence et leurs stabilités au fil des itérations ont été évalués. Pour finir, ce travail présente l'utilisation de la téléportation dans le contexte de la commande sans fil en temps réel du CLMR. Ainsi, il

nous est proposé, une connexion sans fil qui fournit des données de vitesse de la station de commande au robot de type voiture et qui renvoie ensuite la position actuelle du robot à la station. La stratégie consiste à simuler l'échange de données entre les deux plateformes à l'aide d'une méthode inspirée du protocole de téléportation quantique dans le but d'accroître la vitesse de traitement et la précision des données fournies.

Pour tester les performances des algorithmes PSO/QPSO et du protocole de téléportation proposé, il est nécessaire de définir au préalable un modèle mathématique du CLMR et de sa station de commande. À cet effet, le deuxième chapitre présente la construction d'un réseau neuronal artificiel classique (CANN), et d'un réseau neuronal artificiel quantique (QANN), dans le but d'identifier la masse et le moment d'inertie du robot.

Chapitre 2 - Conception d'un ANN classique et de sa version quantique pour le suivi des trajectoires d'un robot mobile de type voiture

2.1 Architecture d'un ANN

Considérons l'ANN présenté dans la figure 2.1:

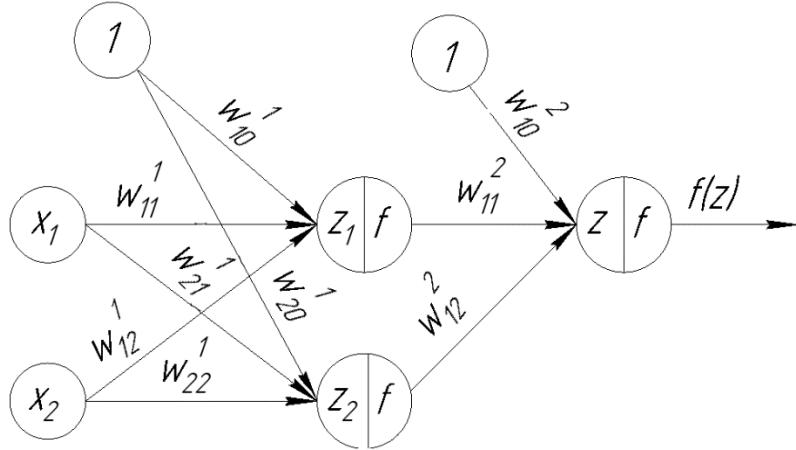


Figure 2.1: Réseau de neurones artificiels (ANN).

Soit un ANN développé pour les tâches de régression ou de classification, le vecteur des entrées X et le vecteur des poids W^1 sont utilisés pour trouver le vecteur des entrées des fonctions d'activation aux deux noeuds de la couche intermédiaire [35] :

$$Z_{1,2} = W^1 \cdot X \rightarrow \begin{bmatrix} Z_1 \\ Z_2 \end{bmatrix} = \begin{bmatrix} w_{10}^1 & w_{11}^1 & w_{12}^1 \\ w_{20}^1 & w_{21}^1 & w_{22}^1 \end{bmatrix} \begin{bmatrix} 1 \\ x_1 \\ x_2 \end{bmatrix} \quad (2.1)$$

Après la réception des signaux provenant du biais et des deux entrées, chaque nœud de la couche intermédiaire fait la somme de toutes les valeurs de ses entrées. Ensuite, chaque somme obtenue est traitée par la fonction d'activation du nœud. Le vecteur des sorties de chaque fonction d'activation à la couche intermédiaire peut être trouvé comme suit [35] :

$$f_1(Z_{1,2}) = \begin{bmatrix} f_1(Z_1) \\ f_1(Z_2) \end{bmatrix} \quad (2.2)$$

Le vecteur des sorties de chaque fonction d'activation à la couche intermédiaire peut alors être considéré comme un vecteur des entrées de la couche de sortie. Par conséquent, le vecteur des entrées et celui des sorties de la fonction d'activation de la couche de sortie sont calculés par analogie aux équations 2.1 et 2.2. Cela signifie que le vecteur des sorties de la couche intermédiaire $f(Z_{1,2})$ et le vecteur des poids W^2 sont utilisés pour trouver le vecteur des entrées de la fonction d'activation de la couche de sortie [35] :

$$Z = W^2 \cdot f_1(Z_{1,2}) \rightarrow Z = [w_{10}^2 \quad w_{11}^2 \quad w_{12}^2] \begin{bmatrix} 1 \\ f_1(z_1) \\ f_1(z_2) \end{bmatrix} \quad (2.3)$$

Le vecteur de sortie de la fonction d'activation de la couche de sortie est trouvé comme suit [35]:

$$f_2(Z) = f_2 \left([w_{10}^2 \quad w_{11}^2 \quad w_{12}^2] \begin{bmatrix} 1 \\ f_1(z_1) \\ f_1(z_2) \end{bmatrix} \right) \quad (2.4)$$

2.1.1 Les fonctions d'activation classiques

Une fonction d'activation est une équation mathématique qui détermine si un neurone doit être activé ou non. Ce type de fonction contribue à introduire la non-linéarité, permettant ainsi aux modèles d'apprendre des relations complexes entre le flux de données d'entrée et celui de sortie. Les fonctions d'activation suivantes sont les plus répandues :

2.1.1.1 Fonction d'activation sigmoïde

La sortie de la sigmoïde se présente sous forme probabiliste d'un signal booléen en fonction des valeurs d'entrée. Si la valeur à l'entrée est négative, la probabilité d'obtenir une valeur

de sortie égale à zéro est élevée. Inversement, dans le cas d'une valeur positive à l'entrée, la probabilité que la valeur de sortie soit égale à un est élevée. La fonction d'activation sigmoïde est principalement utilisée pour effectuer des tâches de classification. La relation mathématique entre les valeurs d'entrée et celles de sortie peut être définie comme suit [35]:

$$\text{Sigmoid}(Z) = \frac{1}{1-e^{-Z}}; \quad Z \in \mathbb{R} \quad (2.5)$$

2.1.1.2 Fonction d'activation linéaire

La fonction d'activation linéaire ne transforme aucunement la valeur de son entrée. Cette valeur est donc textuellement transmise à la sortie, qu'il s'agisse d'une valeur négative ou d'une valeur positive. La relation mathématique entre les valeurs d'entrée et celles de sortie peut être définie comme suit [35] :

$$\text{Linear}(Z) = Z; \quad Z \in \mathbb{R} \quad (2.6)$$

2.1.1.3 Fonction d'activation de rectification linéaire

Relu est une déclinaison de la fonction d'activation linéaire, qui transmet textuellement les valeurs positives de l'entrée à la sortie. Pour les valeurs négatives, la fonction Relu ne transmet que zéro comme valeur de sortie. La relation mathématique entre les valeurs d'entrée et celles de sortie peut être définie comme suit [35] :

$$\text{Relu}(Z) = \max(0, Z); \quad Z \in \mathbb{R} \quad (2.7)$$

2.1.2 Fonctions d'activation inspirées du calcul quantique

Le concept consiste à exprimer l'état du qubit en fonction des valeurs d'entrées classiques Z , puis à utiliser les principes de l'informatique quantique tels que la superposition et la mesure quantique. La probabilité obtenue à la mesure de l'état du qubit servira de valeur de sortie à notre fonction d'activation quantique.

Pour exprimer l'état d'un qubit en fonction de n'importe quelle entrée classique Z , on considère que l'état de ce qubit représente la valeur d'entrée de la fonction d'activation quantique. Ainsi, son état quantique peut être exprimé de la manière suivante :

$$|\Psi\rangle = \cos \theta |0\rangle + \sin \theta e^{i\phi} |1\rangle, \quad (2.8)$$

θ représente l'amplitude de la probabilité des composantes d'état $|0\rangle$ et $|1\rangle$. Les valeurs de l'amplitude varient entre 0 et π ;

ϕ représente la phase relative entre les composantes d'état $|0\rangle$ et $|1\rangle$. Les valeurs de la phase relative varient entre 0 et 2π .

Pour exprimer le qubit en fonction d'une valeur d'entrée classique Z , nous avons pris ϕ égal à 0 rad et l'angle θ a été pris comme le produit de $\frac{\pi}{2}$ et de la valeur de Z à la sortie de la fonction sigmoïde classique. Ces opérations donnent des valeurs de θ comprises entre 0 et $\frac{\pi}{2}$ rad.

$$\theta = \left(\frac{1}{1-e^{-Z}} \right) \frac{\pi}{2}, [rad] \quad (2.9)$$

Après avoir effectué une mesure quantique de $|\Psi\rangle$, si le résultat obtenu donne $|0\rangle$, cela signifie que la probabilité que $|\Psi\rangle$ soit dans l'état classique $|0\rangle$ est plus élevée que celle d'être dans l'état classique $|1\rangle$:

$$(\cos \theta)^2 > (\sin \theta)^2 \quad (2.10)$$

Après avoir effectué la mesure quantique de $|\Psi\rangle$, si le résultat obtenu donne $|1\rangle$, cela signifie que la probabilité que $|\Psi\rangle$ soit dans l'état classique $|0\rangle$ est inférieure ou égale à celle d'être dans l'état classique $|1\rangle$:

$$(\cos \theta)^2 \leq (\sin \theta)^2 \quad (2.11)$$

2.1.2.1 Fonction d'activation sigmoïde inspirée du calcul quantique

La fonction d'activation sigmoïde inspirée du calcul quantique prend le qubit $|\Psi\rangle$ comme entrée, puis effectue sur celui-ci une mesure quantique. La sortie de cette fonction d'activation présente la valeur de la probabilité de $|\Psi\rangle$ d'être dans l'état classique $|1\rangle$:

$$Q\text{Sigmoid}(|\Psi\rangle) = (\sin \theta)^2 \quad (2.12)$$

La figure 2.2 présente le graphique de la fonction sigmoïde classique ainsi que celui de la fonction sigmoïde quantique, afin de les comparer.

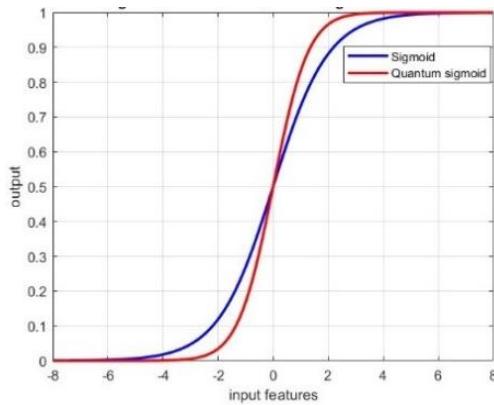


Figure 2.2 : Comparaison de la fonction sigmoïde classique et de la fonction sigmoïde quantique.

Sur la figure 2.2, les deux fonctions sigmoïdes ont un comportement similaire ; elles partent de moins l'infini avec les sorties fixées à zéro avant d'augmenter progressivement pour des

valeurs d'entrée supérieure ou égale à -6 pour la sigmoïde classique et supérieure ou égale à -4 pour la sigmoïde quantique. Les deux fonctions convergent lorsque la valeur d'entrée est de 0 ; pour cette valeur, les deux fonctions d'activation sigmoïde produisent des sorties égales à 0.5. Au-delà de ce point, la sortie de la fonction sigmoïde quantique dépasse la valeur de sortie de la sigmoïde classique pour les mêmes valeurs de l'entrée. La sigmoïde inspirée du calcul quantique atteint sa sortie maximale pour une entrée supérieure ou égale à 4, tandis que la sigmoïde classique l'atteint pour une entrée supérieure ou égale à 6.

2.1.2.2 Fonction d'activation de rectification linéaire inspirée du calcul quantique

La fonction d'activation de rectification linéaire inspirée du calcul quantique est égale à la fonction sigmoïde d'inspiration quantique pondérée par la valeur classique Z :

$$QRelu(|\psi|) = (\sin \theta)^2 \cdot Z \quad (2.13)$$

Pour comparer le graphique de la fonction d'activation de rectification linéaire classique et celui de la fonction inspirée du calcul quantique, la figure 2.3 est présentée ci-dessous :

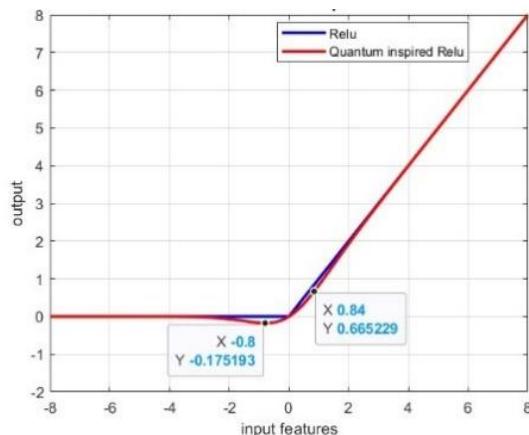


Figure 2.3. Comparaison de la fonction Relu classique et de la fonction Relu quantique.

La fonction d'activation de rectification linéaire classique et celle inspirée du calcul quantique ont beaucoup de similitudes. Pour les valeurs d'entrée négatives, les deux fonctions d'activation donnent en sortie 0; et pour les valeurs d'entrée positives, elles génèrent toutes les deux des sorties égales aux valeurs d'entrée. Cependant, pour les valeurs d'entrée comprises entre -2 et 2, la fonction inspirée du calcul quantique subit une légère transformation où les valeurs étaient un peu plus petites que les valeurs attendues.

2.1.2.3 Fonction d'activation linéaire inspirée du calcul quantique

La fonction d'activation linéaire inspirée du calcul quantique prend le qubit $|\psi\rangle$ comme entrée, puis effectue ensuite une mesure quantique sur celui-ci. Si l'entrée classique Z est négative, la sortie de la fonction d'activation quantique sera donc la probabilité d'avoir $|\psi\rangle$ dans l'état classique $|0\rangle$, pondérée par la valeur classique Z . Inversement, si l'entrée classique Z est positive, la sortie de la fonction sera égale à la probabilité d'avoir $|\psi\rangle$ dans l'état classique $|1\rangle$, pondérée par la valeur classique Z :

$$Qlinear(|\psi\rangle) = \begin{cases} (\cos \theta)^2 \cdot Z, & Z < 0 \\ (\sin \theta)^2 \cdot Z, & Z \geq 0 \end{cases} \quad (2.14)$$

Pour comparer le graphique de la fonction d'activation linéaire classique et celui de la fonction d'activation linéaire inspirée du calcul quantique, la figure 2.4 est présentée ci-dessous :

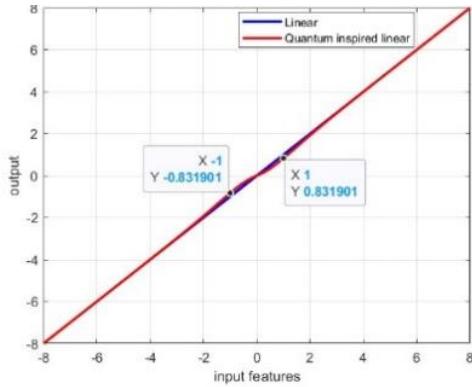


Figure 2.4. Comparaison de la fonction linéaire classique et de la fonction linéaire inspirée du calcul quantique.

Les deux fonctions d'activation linéaires présentent de nombreuses similitudes. Pour les valeurs négatives comme pour celles positives, les deux fonctions d'activation donnent en sortie des valeurs identiques aux valeurs d'entrée. Cependant, pour les valeurs d'entrée comprises entre -2 et 0, les valeurs de sortie de la fonction d'activation inspirée du calcul quantique subissent une légère transformation où les valeurs de sortie étaient un peu plus grandes que les valeurs attendues. En revanche, pour les valeurs d'entrée comprises entre 0 et 2, les valeurs de sortie de la fonction d'activation inspirée du calcul quantique subissent une petite transformation où les valeurs sont devenues un peu plus petites que les valeurs attendues.

2.2 Conception et entraînement d'un ANN inspiré du calcul quantique

Pour la conception du ANN, différentes configurations de robots ont été expérimentées en faisant varier la masse et le moment d'inertie, tout en suivant une trajectoire identique pour tous. En raison des contraintes dynamiques et de l'absence d'optimisation, chaque robot, avec des propriétés de masse et d'inertie distinctes, a généré une trajectoire unique. Vingt positions des robots sur leurs trajectoires respectives, à des moments fixes, ont été

enregistrées comme RMSE pour chaque robot. Ainsi, en disposant au préalable des masses et des moments d'inertie des robots, un modèle ANN a été entraîné pour prédire la masse et l'inertie d'un robot en fonction de son RMSE. L'entraînement a été réalisé grâce à une quantité considérable de données où les masses étaient comprises entre 5 et 20 kg, et les moments d'inertie entre 0,1 et $0,25 \text{kg} \cdot \text{m}^2$.

Afin de définir les paramètres de l'architecture du QANN, des robots mobiles tentant de suivre la même trajectoire ont été considérés. Les contraintes dynamiques et l'absence de tout processus d'optimisation ont rendu impossible un suivi parfait de la trajectoire souhaitée. En considérant les RMSE des robots à 20 moments identiques de la simulation, ce qui représente le nombre d'entrées du QANN; et en fixant à deux le nombre des sorties représentant les masses et les moments d'inertie des robots. Nous pouvons trouver la relation mathématique entre les entrées et les sorties.

Pour l'optimisation des poids et des biais, le vecteur d'entrées X et le vecteur des poids W^1 sont considérés. Cela permet de trouver le vecteur d'entrées des fonctions d'activation des 20 nœuds de la couche intermédiaire.

$$Z_{1\dots 20}^1 = W^1 \cdot X \rightarrow \begin{bmatrix} Z_1^1 \\ \vdots \\ Z_{20}^1 \end{bmatrix} = \begin{bmatrix} w_{1-0}^1 & \dots & w_{1-20}^1 \\ \vdots & \ddots & \vdots \\ w_{20-0}^1 & \dots & w_{20-20}^1 \end{bmatrix} \begin{bmatrix} x_1 \\ \vdots \\ x_{20} \end{bmatrix} \quad (2.15)$$

Le vecteur d'entrées des fonctions d'activation inspirées du calcul quantique, au niveau des 20 nœuds de la couche intermédiaire, peut être défini selon les équations 2.8 et 2.9 :

$$\begin{bmatrix} |\Psi\rangle_1^1 \\ \vdots \\ |\Psi\rangle_{20}^1 \end{bmatrix} \quad (2.16)$$

Le vecteur des sorties des fonctions d'activation inspirées du calcul quantique, à chacun des 20 nœuds de la couche intermédiaire, peut être trouvé comme suit :

$$QRelu(|\psi\rangle^1_{1\dots 20}) = \begin{bmatrix} QRelu(|\psi\rangle^1_1) \\ \vdots \\ QRelu(|\psi\rangle^1_{20}) \end{bmatrix} = \begin{bmatrix} (\sin \theta^1_1)^2 \cdot Z^1_1 \\ \vdots \\ (\sin \theta^1_{20})^2 \cdot Z^1_{20} \end{bmatrix} \quad (2.17)$$

Le vecteur des sorties des fonctions d'activation inspirées du calcul quantique, à chacun des 20 nœuds de la couche intermédiaire $QRelu(|\psi\rangle^1_{1\dots 20})$ et le vecteur des poids W^2 ont été utilisés pour trouver le vecteur d'entrées des fonctions d'activation de la couche de sortie :

$$Z^2_{1,2} = W^2 \cdot QRelu(|\psi\rangle^1_{0\dots 20}) \rightarrow \begin{bmatrix} Z^2_1 \\ Z^2_2 \end{bmatrix} = \begin{bmatrix} w^2_{1-0} & \dots & w^2_{1-20} \\ w^2_{2-0} & \dots & w^2_{2-20} \end{bmatrix} \begin{bmatrix} (\sin \theta^1_1)^2 \cdot Z^1_1 \\ \vdots \\ (\sin \theta^1_{20})^2 \cdot Z^1_{20} \end{bmatrix} \quad (2.18)$$

Afin de trouver le vecteur d'entrées des fonctions d'activation des 2 nœuds de la couche de sortie, nous considérons les équations 2.8 et 2.9 :

$$\begin{bmatrix} |\psi\rangle^2_1 \\ |\psi\rangle^2_2 \end{bmatrix} \quad (2.19)$$

Le vecteur des sorties des fonctions d'activation inspirées du calcul quantique, à la couche de sortie, peut être trouvé de manière analogue à l'équation 2.14 :

$$Qlinear(|\psi\rangle^2_1) = \begin{cases} (\cos \theta^2_1)^2 \cdot Z^2_1, & Z^2_1 < 0 \\ (\sin \theta^2_1)^2 \cdot Z^2_1, & Z^2_1 \geq 0 \end{cases} \quad (2.20)$$

$$Qlinear(|\psi\rangle^2_2) = \begin{cases} (\cos \theta^2_2)^2 \cdot Z^2_2, & Z^2_2 < 0 \\ (\sin \theta^2_2)^2 \cdot Z^2_2, & Z^2_2 \geq 0 \end{cases} \quad (2.21)$$

Après avoir trouvé le signal final aux deux nœuds de sortie, les RMSE entre les valeurs de sortie du QANN et les valeurs réelles ont été corrigés en minimisant la fonction de coût à l'aide d'un algorithme d'optimisation. Plus la fonction de coût tend vers zéro, plus le QANN devient précis pour prédire les sorties à partir des entrées.

Pour ce faire, la première dérivée des poids entre la couche intermédiaire et la couche de sortie a été calculée :

$$der(W^2) = (Qlinear(|\psi\rangle_{1,2}^2) - Y_{1,2}) \cdot \left(QRelu(|\psi\rangle_{1\dots 20}^1)\right)^T \quad (2.22)$$

Les poids entre la couche intermédiaire et la couche de sortie sont mis à jour à chaque itération :

$$W_{New}^2 = W_{old}^2 - LR \cdot der(W_{old}^2) \quad (2.23)$$

La dérivée première des biais entre la couche intermédiaire et la couche de sortie est calculée comme suit :

$$der(b^2) = (Qlinear(|\psi\rangle_{1,2}^2) - Y_{1,2}) \quad (2.24)$$

Les biais entre la couche intermédiaire et la couche de sortie sont mis à jour à chaque itération :

$$b_{New}^2 = b_{old}^2 - LR \cdot der(b_{old}^2) \quad (2.25)$$

La première dérivée des fonctions d'activation entre la couche d'entrée et la couche intermédiaire est calculée comme suit :

$$der(QRelu) = (W^2)^T \cdot (Qlinear(|\psi\rangle_{1,2}^2) - Y_{1,2}) \quad (2.26)$$

La première dérivée du vecteur d'entrées des fonctions d'activation des 20 nœuds de la couche intermédiaire est calculée comme suit :

$$der(Z_{1\dots 20}^1) = der(QRelu)_{i1} \cdot der\left(QRelu(|\psi\rangle_{1\dots 20}^1)\right)_{i1}, \text{with } i = 1, \dots, 20 \quad (2.27)$$

La première dérivée des poids entre la couche d'entrée et la couche intermédiaire :

$$der(W^1) = der(Z_{1\dots 20}^1) \cdot (X)^T \quad (2.28)$$

Mise à jour des poids entre la couche d'entrée et la couche intermédiaire :

$$W_{New}^1 = W_{old}^1 - LR \cdot der(W_{old}^1) \quad (2.29)$$

La dérivée première des biais entre la couche d'entrée et la couche intermédiaire :

$$der(b^1) = der(Z_{1 \dots 20}^1) \quad (2.30)$$

Mise à jour des biais entre la couche d'entrée et la couche intermédiaire :

$$b_{New}^1 = b_{old}^1 - LR \cdot der(b_{old}^1) \quad (2.31)$$

Chapitre 3 - Optimisation des paramètres du régulateur PD basée sur les versions classique et quantique du PSO pour le suivi des trajectoires d'un robot mobile de type voiture

3.1 Modèle mathématique du robot mobile de type voiture

Les robots mobiles de type voiture sont généralement caractérisés par la mobilité des roues et le mécanisme autonome de direction. Pour ce faire, un algorithme simulant les mouvements dynamiques d'un CLMR a été implémenté. L'objectif est de comparer les capacités des algorithmes PSO et QPSO, à minimiser et à stabiliser les RMSE entre les trajectoires voulues et celles suivies par le CLMR. Les vitesses et les accélérations du robot sont donc communiquées par le biais d'équations différentielles variables dans le temps.

Les équations 3.1 jusqu'à 3.10 sont tirées ou inspirées de la littérature [36-39].

Les positions souhaitées du robot dans l'espace tridimensionnel sont décrites comme suit :

$$\eta_d = \begin{cases} x(t) = 10 \cdot \sin(0.1 \cdot t_i) \text{ [m]} \\ y(t) = 10 - 10 \cdot \cos(0.1 \cdot t_i) \text{ [m]} \\ z(t) = 0.1 \cdot t_i \text{ [rad]} \end{cases} \quad (3.1)$$

x(t) représente les déplacements du robot dans l'axe des abscisses.

y(t) représente les déplacements du robot dans l'axe des ordonnées.

z(t) représente les angles formés par la direction du robot et l'axe des abscisses.

Pour trouver les vitesses et les accélérations souhaitées, les dérivées première et seconde des positions souhaitées sont respectivement effectuées :

$$\dot{\eta}_d = \begin{cases} \dot{x}(t) = \cos(0.1 \cdot t_i) \text{ [m/s]} \\ \dot{y}(t) = \sin(0.1 \cdot t_i) \text{ [m/s]} \\ \dot{z}(t) = 0.1 \text{ [rad/s]} \end{cases} \quad (3.2)$$

$$\ddot{\eta}_d = \begin{cases} \ddot{x}(t) = -0.1 \cdot \sin(0.1 \cdot t_i) \text{ [m/s}^2] \\ \ddot{y}(t) = 0.1 \cdot \cos(0.1 \cdot t_i) \text{ [m/s}^2] \\ \ddot{z}(t) = 0 \text{ [rad/s}^2] \end{cases} \quad (3.3)$$

Les équations cinématiques et le système de coordonnées cartésiennes ont été utilisés pour mieux comprendre les mouvements du CLMR. La figure 3.1 montre un CLMR se déplaçant dans un système de coordonnées cartésiennes X_IY_I . Le robot possède un système de coordonnées cartésiennes X_RY_R , situé au niveau de son centre de gravité. Les informations relatives au cadre global X_IY_I , sont trouvées grâce aux vitesses instantanées u , v et r .

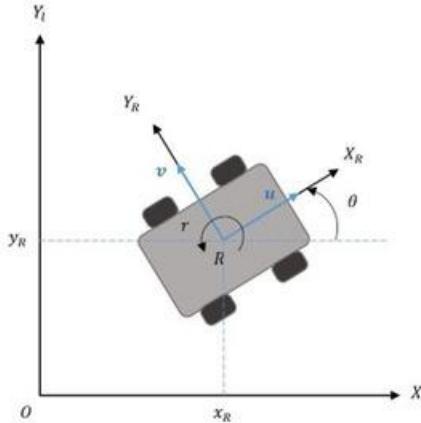


Figure 3.1: Cinématique du robot mobile sur une surface plane en 2D (adapté de [40]).

Les équations cinématiques des mouvements sont représentées comme suit :

$$\begin{cases} \dot{X} = u \cos \theta - v \sin \theta \\ \dot{Y} = u \sin \theta + v \cos \theta \\ \dot{\theta} = r \end{cases} \quad (3.4)$$

Où

u : la vitesse longitudinale, exprimée en m/s ;

v : la vitesse latérale, exprimée en m/s ;

r : la vitesse angulaire, exprimée en rad/s ;

θ : angle entre le vecteur de la vitesse longitudinale du robot et l'axe x du système global.

La transformation des équations du mouvement en une forme matricielle d'équations

cinématiques différentielles inverses donne les résultats suivants :

$$\begin{bmatrix} u \\ v \\ r \end{bmatrix} = \begin{bmatrix} \cos \theta & \sin \theta & 0 \\ -\sin \theta & \cos \theta & 0 \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} \dot{X} \\ \dot{Y} \\ \dot{\theta} \end{bmatrix} \rightarrow \zeta = J^{-1}(\theta) \cdot \dot{\eta}_d \quad (3.5)$$

Où

ζ : le vecteur des vitesses du CLMR;

$J^{-1}(\theta)$: l'inverse de la matrice jacobienne $J(\theta)$.

La matrice jacobienne transforme le vecteur des vitesses, lié au référentiel du robot, en un référentiel global. En d'autres termes, $J(\theta)$ tient compte de l'orientation du robot dans le référentiel global. Le CLMR est un système nonholonomique défini par ses contraintes de vitesse plutôt que par ses contraintes de position. Cela souligne l'incapacité du robot à se déplacer latéralement. Le CLMR ne peut qu'avancer ou reculer. L'angle de braquage δ est considéré comme l'angle formé par les pneus avant et l'axe de direction du robot. Cet angle est donc calculé comme suit :

$$\delta = \tan^{-1}(l \cdot r / R_{u,v}), \text{with } R_{u,v} = \sqrt{u^2 + v^2} \quad (3.6)$$

Où

l : la longueur du CLMR, exprimée en mètres;

r : la vitesse angulaire, exprimée en rad/s ;

$R_{u,v}$: la résultante des vitesses longitudinale et latérale, exprimée en m/s .

L'angle de braquage est généralement limité par des contraintes mécaniques ; Les valeurs de cet angle varient généralement entre -45° et $+45^\circ$. Les angles avec des valeurs négatives

indiquent les virages à gauche, tandis que les angles avec des valeurs positives indiquent les virages à droite. Le CLMR étant un système nonholonomique, la vitesse latérale doit, en tout temps, être égale à zéro. Il en résulte ce qui suit :

$$\begin{bmatrix} u_n \\ v_n \\ r_n \end{bmatrix} = \begin{bmatrix} 1 \\ 0 \\ (\tan \delta)/l \end{bmatrix} [\sqrt{u^2 + v^2 + r^2}] \rightarrow \zeta_n = W \cdot R_{u,v,r} \quad (3.7)$$

Où

ζ_n : le vecteur des vitesses nonholonomiques du CLMR;

u_n : la composante longitudinale du vecteur ζ_n , exprimée en m/s ;

v_n : la composante latérale du vecteur ζ_n , exprimée en m/s ;

r_n : la composante angulaire du vecteur ζ_n , exprimée en rad/s ;

W : la matrice d'état nonholonomique;

$R_{u,v,r}$: la résultante des vitesses longitudinale, latérale et angulaire, exprimée en m/s .

Pour trouver le vecteur des accélérations du CLMR, les composants du vecteur des vitesses de l'équation 3.5 sont dérivés en fonction du temps comme suit :

$$\dot{\eta}_d = J(\theta) \cdot \zeta, \quad \dot{\eta}_d \equiv \begin{bmatrix} \dot{\eta}_{dx} [m/s] \\ \dot{\eta}_{dy} [m/s] \\ \dot{\eta}_{d\theta} [rad/s] \end{bmatrix} \quad (3.8)$$

$$\ddot{\eta}_d = J(\dot{\theta}) \cdot \zeta + J(\theta) \cdot \dot{\zeta}, \quad \ddot{\eta}_d \equiv \begin{bmatrix} \ddot{\eta}_{dx} [m/s^2] \\ \ddot{\eta}_{dy} [m/s^2] \\ \ddot{\eta}_{d\theta} [rad/s^2] \end{bmatrix} \quad (3.9)$$

Dans l'équation 3.9, le vecteur des vitesses est remplacé par la formule trouvée dans l'équation 3.5 pour trouver une équation explicite de l'accélération du CLMR. Le résultat est le suivant :

$$\dot{\zeta} = J^{-1}(\theta) \cdot (\ddot{\eta}_d - J(\dot{\theta}) \cdot J^{-1}(\theta) \cdot \dot{\eta}_d), \dot{\zeta} \equiv \begin{bmatrix} \dot{u} [m/s^2] \\ \dot{v} [m/s^2] \\ \dot{r} [rad/s^2] \end{bmatrix} \quad (3.10)$$

Le CLMR étant un système nonholonomique, l'équation du vecteur des accélérations du CLMR devient la suivante :

$$\begin{bmatrix} \dot{u}_n \\ \dot{v}_n \\ \dot{r}_n \end{bmatrix} = \begin{bmatrix} 1 \\ 0 \\ (\tan \delta)/l \end{bmatrix} [\sqrt{\dot{u}^2 + \dot{v}^2 + \dot{r}^2}] \rightarrow \dot{\zeta}_n = W \cdot R_{\dot{u}, \dot{v}, \dot{r}} \quad (3.11)$$

Où

$\dot{\zeta}_n$: le vecteur des accélérations nonholonomiques du CLMR;

\dot{u}_n : la composante longitudinale du vecteur $\dot{\zeta}_n$, exprimée en m/s^2 ;

\dot{v}_n : la composante latérale du vecteur $\dot{\zeta}_n$, exprimée en m/s^2 ;

\dot{r}_n : la composante angulaire du vecteur $\dot{\zeta}_n$, exprimée en m/s^2 ;

W : la matrice d'état nonholonomique;

$R_{\dot{u}, \dot{v}, \dot{r}}$: la résultante de \dot{u}_n , \dot{v}_n et \dot{r}_n , exprimée en m/s^2 ;

Pour obtenir les équations du mouvement, essentielles pour la conception et la commande du CLMR, ainsi que pour la conception du régulateur et du système de navigation appropriés afin d'optimiser ses performances en matière de suivi des trajectoires, l'équation d'Euler-Lagrange suivante est utilisée, avec les paramètres dynamiques définis dans la figure 3.2 [36-39] :

$$\frac{d}{dt} \left(\frac{\partial L}{\partial v} \right) - \frac{\partial L}{\partial x} = F, \text{with } L = KE - PE \quad (3.12)$$

Où

KE : l'énergie cinétique, exprimée en joule;

PE : l'énergie potentielle, exprimée en joule.

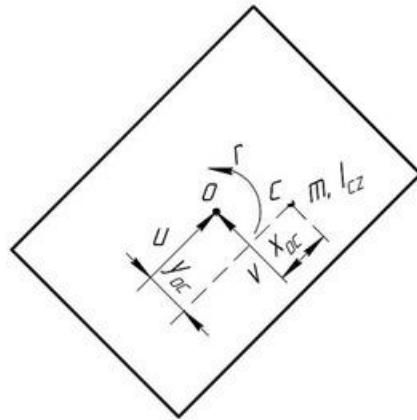


Figure 3.2: Dynamique du robot mobile sur une surface plane en 2D.

Le robot étant terrestre, il ne possède pas d'énergie potentielle. L'équation de Lagrange devient donc la suivante :

$$L = KE = \frac{1}{2}m(\dot{X}_C^2 + \dot{Y}_C^2) + \frac{1}{2}I_{CZ} \cdot r^2 \quad (3.13)$$

Où

m : la masse du robot, exprimée en Kg;

\dot{X}_C : la vitesse du centre d'inertie par rapport à l'axe des abscisses, exprimée en m/s ;

\dot{Y}_C : la vitesse du centre d'inertie par rapport à l'axe des ordonnées, exprimée en m/s ;

I_{CZ} : le moment d'inertie par rapport à l'axe z, exprimée en Nm ;

r : la vitesse angulaire par rapport à l'axe z, exprimée en rad/s .

Les deux vitesses au centre d'inertie par rapport à l'axe des abscisses et à l'axe des ordonnées sont définies dans le système d'équations suivant :

$$\begin{cases} \dot{X}_C = \dot{x}_{bc} = u - y_{bc} \cdot r \\ \dot{Y}_C = \dot{y}_{bc} = v + x_{bc} \cdot r \end{cases} \quad (3.14)$$

Où

x_{bc} est la distance entre l'origine du référentiel fixé au centre de gravité du robot et le centre d'inertie par rapport à l'axe des abscisses, exprimée en mètres;

y_{bc} est la distance entre l'origine du référentiel fixé au centre de gravité du robot et le centre d'inertie par rapport à l'axe des ordonnées, exprimée en mètres.

En remplaçant l'équation 3.14 par le lagrangien de l'équation 3.13 :

$$L = \frac{1}{2}m(u^2 - 2u \cdot r \cdot y_{bc} + v^2 + 2v \cdot r \cdot x_{bc} + r^2(x_{bc}^2 + y_{bc}^2)) + \frac{1}{2}I_{CZ} \cdot r^2 \quad (3.15)$$

Le système d'équations d'Euler-Lagrange peut être calculé comme suit :

$$\left\{ \begin{array}{l} \frac{\partial L}{\partial u} = m \cdot u - m \cdot r \cdot y_{bc} \\ \frac{\partial L}{\partial v} = m \cdot v + m \cdot r \cdot x_{bc} \\ \frac{\partial L}{\partial r} = -m \cdot u \cdot y_{bc} + m \cdot v \cdot x_{bc} + m \cdot r(x_{bc}^2 + y_{bc}^2) + I_{CZ} \cdot r \end{array} \right. \quad (3.16)$$

Le système d'équations du mouvement par rapport au référentiel fixé au centre de gravité peut être calculé comme suit:

$$\left\{ \begin{array}{l} F_x = \frac{d}{dt} \left(\frac{\partial L}{\partial u} \right) = m \cdot \dot{u} - m \cdot \dot{r} \cdot y_{bc} - m \cdot r(v + x_{bc} \cdot r) \quad [N] \\ F_y = \frac{d}{dt} \left(\frac{\partial L}{\partial v} \right) = m \cdot \dot{v} + m \cdot \dot{r} \cdot x_{bc} + m \cdot r(u - y_{bc} \cdot r) \quad [N] \\ M_z = \frac{d}{dt} \left(\frac{\partial L}{\partial r} \right) = \\ -m \cdot \dot{u} \cdot y_{bc} - m \cdot u(v + x_{bc} \cdot r) + m \cdot \dot{v} \cdot x_{bc} + m \cdot v(u - y_{bc} \cdot r) + \\ m \cdot \dot{r}(x_{bc}^2 + y_{bc}^2) + m \cdot r[2x_{bc}(u - y_{bc} \cdot r) + 2y_{bc}(v + x_{bc} \cdot r)] + I_{CZ} \cdot \ddot{r} [Nm] \end{array} \right. \quad (3.17)$$

Dans le cas du CLMR, l'origine du référentiel fixé au centre de gravité coïncide avec le centre d'inertie. Cela implique que x_{bc} et y_{bc} sont tous deux égaux à zéro. Par conséquent, le couple appliqué au CLMR est influencé par la dynamique de translation et de rotation du robot ; les équations dynamiques du mouvement du robot peuvent alors être exprimées sous la forme matricielle suivante :

$$\begin{bmatrix} F_x \\ F_y \\ M_z \end{bmatrix} = \begin{bmatrix} m & 0 & 0 \\ 0 & m & 0 \\ 0 & 0 & I_{CZ} \end{bmatrix} \begin{bmatrix} \dot{u} \\ \dot{v} \\ \dot{r} \end{bmatrix} + \begin{bmatrix} m \cdot r \cdot v \\ m \cdot r \cdot u \\ 0 \end{bmatrix} \Rightarrow \tau = D \cdot \dot{\zeta} + n(\zeta) \quad (3.18)$$

où

$\dot{\zeta}$: le vecteur des accélérations nonholonomiques du CLMR;

τ : le vecteur des efforts dynamiques ou efforts généralisés agissant sur le robot mobile;

D : la matrice dynamique du CLMR;

$n(\zeta)$: les autres effets de l'environnement sur le CLMR.

Pour un recalibrage adéquat de la commande d'accélération et de vitesse, les paramètres dynamiques, tels que le couple et la commande C, sont considérés. La commande C ajuste à la fois la vitesse et l'accélération, tout en tenant compte de la position :

$$C = K_p \cdot \varepsilon + K_d \cdot \dot{\varepsilon} \quad (3.19)$$

Où

$$\varepsilon = \eta_d - \eta, \text{ où } \varepsilon \text{ exprimée en } \begin{bmatrix} m \\ m \\ rad \end{bmatrix};$$

$$\dot{\varepsilon} = \dot{\eta}_d - \dot{\eta} \text{ où exprimée en } \begin{bmatrix} m/s \\ m/s \\ rad/s \end{bmatrix}.$$

Avec K_p et K_d respectivement la composante proportionnelle et la composante dérivée du régulateur PD.

La nouvelle équation de la commande d'accélération devient la suivante :

$$\dot{\zeta} = D^{-1}(\tau - n(\zeta) - Vf \cdot \zeta) + J^{-1}(\theta) \cdot C, \text{ où } \dot{\zeta} \text{ exprimée en } \begin{bmatrix} m/s^2 \\ m/s^2 \\ rad/s^2 \end{bmatrix} \quad (3.20)$$

Où

Vf : le coefficient de frottement visqueux.

La commande de vitesse et celle de la position du robot peuvent être calculées par analogie avec la méthode d'Euler, comme suit :

$$\zeta_{t+1} = \zeta_t + \dot{\zeta}_t \cdot dt, \text{ où } \zeta_{t+1} \text{ exprimée en } \begin{bmatrix} m/s \\ m/s \\ rad/s \end{bmatrix} \quad (3.21)$$

$$\eta_{t+1} = \eta_t + \dot{\eta}_t \cdot dt, \text{with } \eta_0 = \begin{cases} x(0) = 0 [m] \\ y(0) = 0 [m] \\ z(0) = 0 [rad] \end{cases} \quad (3.22)$$

Dans les cas où la commande d'accélération $\dot{\zeta}_t$ est constante, la dérivée du vecteur d'état peut être trouvée comme suit : $\dot{\eta}_t = J(\theta)_t \cdot \zeta_t$. Toutefois, si des changements considérables de la commande d'accélération sont observés avant la correction par rétroaction, la dérivée du vecteur d'état sera trouvée comme suit : $\dot{\eta}_t = J(\theta)_t \cdot \zeta_t + \dot{\zeta}_t \cdot dt$. Le terme $\dot{\zeta}_t \cdot dt$ représente le changement de l'accélération au fil du temps. Le vecteur d'état, représentant la position améliorée du robot, est donné comme suit :

$$\eta_{t+1} = \eta_t + (J(\theta)_t \cdot \zeta_t + \dot{\zeta}_t \cdot dt)dt \quad (3.23)$$

Après avoir proposé un algorithme qui simule avec succès les mouvements dynamiques du CLMR, les étapes suivantes consistent à mettre en œuvre deux algorithmes, PSO et QPSO pour améliorer la capacité de suivi des trajectoires du robot.

3.2 Optimisation par essaim de particules (PSO)

PSO est un algorithme métaheuristique, s'inspirant du comportement social des volées d'oiseaux ou des bancs de poissons. Ces types d'algorithmes sont généralement utilisés pour trouver des solutions optimales à des problèmes donnés en effectuant des recherches itératives dans l'espace de recherche. Dans le cas du robot étudié, l'espace de recherche est compris entre 0 et 500. Les particules PSO représentent des solutions potentielles et se déplacent dans l'espace de recherche pour trouver la meilleure solution.

Première étape, 20 particules ont été initialisées de manière aléatoire à des emplacements uniformément répartis dans l'espace de recherche ; les vitesses des particules ont également été initialisées de manière aléatoire avec une distribution uniforme, mais réduites d'un

facteur de 2 pour pallier le risque de dépassement. Ensuite, les particules se déplacent itérativement 100 fois dans l'espace de recherche dans le but de trouver les positions optimales. Une fonction d'évaluation utilise itérativement différentes valeurs de K_p et K_d , qui sont respectivement les coordonnées x et y de la meilleure position globale de la particule la mieux placée dans la zone de recherche. Ainsi, la meilleure position globale est mise à jour lorsque l'évaluation de l'aptitude de la fonction d'aide est inférieure à la meilleure aptitude globale qui, au début de la simulation, est initialisée à l'infini avant de commencer à prendre les valeurs de l'aptitude de la meilleure position globale mise à jour. Par analogie, les mises à jour des meilleures positions personnelles ont lieu lorsque l'évaluation de l'aptitude de la fonction d'aide est inférieure à la meilleure aptitude personnelle qui, au début de la simulation, est initialisée à la meilleure position aléatoire de chaque particule.

À partir de leurs positions initiales, les particules se déplacent vers leurs nouvelles positions avec des vitesses qui peuvent être calculées comme suit :

$$V_i^{t+1} = W \cdot V_i^t + C_1 \cdot r_1(P_{b1}^t - P_i^t) + C_2 \cdot r_2(G_b^t - P_i^t) \quad (3.24)$$

$$P_i^{t+1} = P_i^t + V_i^{t+1} \quad (3.25)$$

Où

W représente le poids inertiel. La valeur du poids d'inertie est comprise entre 0 et 1. Son rôle est de déterminer les directions des particules. Les valeurs élevées de W indiquent une plus grande influence du mouvement précédent des particules, tandis que les valeurs plus faibles proches de 0 indiquent le contraire;

V_i^{t+1} : la vitesse de la ième particule à l'instant t+1;

P_i^{t+1} : la position de la ième particule à l'instant t+1;

C_1 : le poids cognitif ou personnel;

r_1 et r_2 sont des nombres aléatoires compris entre 0 et 1;

C_2 : le poids social;

P_{b1}^t : la meilleure position personnelle à l'instant t;

G_b^t : la meilleure position globale à l'instant t.

Les mouvements des particules sont influencés par chaque partie de la formule de vitesse.

Le terme $W \cdot V_i^t$ mène les particules dans la partie de l'essaim ayant un potentiel plus élevé de trouver la meilleure solution. Le terme $C_1 \cdot r_1(P_{b1}^t - P_i^t) + C_2 \cdot r_2(G_b^t - P_i^t)$ compare les solutions précédentes pour trouver la meilleure solution globale de l'espace de recherche. Plus précisément, $C_1 \cdot r_1(P_{b1}^t - P_i^t)$ est l'influence cognitive qui améliore les positions individuelles, et $C_2 \cdot r_2(G_b^t - P_i^t)$ est l'influence sociale qui fait que les particules suivent la meilleure solution globale.

En résumé, le poids d'inertie est fixé à une valeur qui permet de réduire l'influence des mouvements antérieurs des particules. Ensuite, les poids cognitif et social sont fixés à des valeurs permettant d'assurer l'équilibre entre la recherche de nouveaux sous-domaines (exploration) et l'amélioration d'une bonne solution (exploitation). Ces paramètres permettent une convergence rapide de l'essaim tout en permettant l'exploration de la zone, évitant aux particules de s'enfermer dans des optimums locaux. En ajustant constamment les positions et les vitesses en fonction des résultats optimaux personnels et globaux, les particules convergent progressivement vers le résultat optimal. La position optimale correspond aux valeurs des coordonnées x et y de la meilleure position globale, qui sont attribuées respectivement à K_p et K_d qui permettent de corriger les erreurs de suivi de la

trajectoire du CLMR en réduisant le RMSE. À la fin des itérations, la valeur de la meilleure position sera considérée comme K_p et K_d optimisés.

$$G_b \equiv \begin{bmatrix} K_p \\ K_d \end{bmatrix} \quad (3.26)$$

G_b meilleur point global à la fin des itérations.

3.3 Optimisation de l'essaim particulaire inspirée du calcul quantique (QPSO)

L'algorithme QPSO est une variante plus avancée de l'algorithme PSO classique, utilisant les principes de la mécanique quantique, tels que l'état de superposition, pour renforcer les capacités de recherche des particules dans l'essaim. La différence majeure entre les deux types d'algorithmes réside dans le fait que l'algorithme PSO déplace ses particules dans l'espace de recherche, en déterminant les nouvelles positions en fonction des vitesses, influencées par l'expérience des particules, tandis que l'algorithme QPSO utilise des qubits pour coder les particules, ce qui leur confère une plus grande diversité au sein de la population. La représentation de l'état quantique correspondant à la sphère de Bloch est la suivante :

$$|\Psi\rangle = \cos \theta |0\rangle + e^{-i\phi} \cdot \sin \theta |1\rangle \quad (3.27)$$

Où $|\Psi\rangle$ est l'état aléatoire d'un qubit.

$0 \leq \theta \leq \pi$, représente les amplitudes de probabilité des composantes d'état $|0\rangle$ et $|1\rangle$.

$0 \leq \phi \leq 2\pi$, représente la phase relative entre les composantes d'état $|0\rangle$ et $|1\rangle$.

Pour l'algorithme QPSO proposé, la rotation de l'angle quantique est utilisée pour décrire les mouvements des particules dans l'espace de recherche.

La formule de la vitesse de chaque particule est donnée comme suit :

$$V_i^{t+1} = W \cdot V_i^t + C_1 \cdot r_1(\theta_{P_b}^t - \theta_i^t) + C_2 \cdot r_2(\theta_{G_b}^t - \theta_i^t) \quad (3.28)$$

L'angle quantique du qubit $|\psi\rangle$, qui décrit les mouvements des particules dans l'espace de recherche, est calculé comme dans l'équation suivante :

$$\theta_i^{t+1} = \theta_i^t + V_i^{t+1} \quad (3.29)$$

Où

θ_i^{t+1} : la rotation de l'angle quantique représentant la ième particule au temps t+1;

$\theta_{P_b}^t$: la rotation de l'angle quantique représentant la meilleure particule personnelle au

temps t;

$\theta_{G_b}^t$: la rotation de l'angle quantique représentant la meilleure particule globale à l'instant

t.

Au début de leurs recherches, les particules θ_i sont initialisées aléatoirement et uniformément entre 0 et $\pi/2$, avant de se déplacer dans leur espace de recherche selon l'équation 3.29. Ensuite, au fil des itérations, toutes les particules convergent vers la position optimale. Pour l'algorithme QPSO, les valeurs des constantes proportionnelle et dérivée sont trouvées comme suit:

$$\begin{cases} \text{Max}_{of K_p} \cdot \sin(\theta_{G_b}) = K_p \\ \text{Max}_{of K_d} \cdot \sin(\theta_{G_b}) = K_d \end{cases} \quad (3.30)$$

Où

θ_{G_b} : la rotation de l'angle quantique représentant la meilleure position globale;

$\text{Max}_{of K_p}$: la valeur maximale de la constante proportionnelle, qui est égale à 500;

$\text{Max}_{of K_d}$: la valeur maximale de la constante dérivée, qui est égale à 500.

Chapitre 4 - Téléportation quantique pour la commande à distance d'un robot mobile de type voiture

4.1 Travaux connexes

De nombreux scientifiques ont inclus la téléportation quantique dans leurs projets de recherche, car les perspectives en matière de traitement de l'information, de cryptographie et de communication sont indéniables. Une équipe de chercheurs [41] a utilisé un réseau quantique pour transporter des données de manière fiable ; des points quantiques, qui sont des nanocristaux semi-conducteurs émettant de la lumière dans la bande des télécommunications, ont été utilisés à cet effet. Un autre travail remarquable portant sur l'application de la téléportation quantique à un spectre plus large de systèmes physiques a été effectué [42] en enchevêtrant le qubit d'un système à variables continues et en téléportant ensuite l'état du qubit dans un système discret. Dans l'étude de Salimian et al. [43], un système de téléportation quantique permettant de transférer l'état de deux qubits supraconducteurs du premier au second laboratoire a été développé en connectant des qubits supraconducteurs entre eux et à des résonateurs LC. Pendant la téléportation quantique, il arrive parfois des défauts sous la forme de bruits imprévisibles ; Zhang et ses collaborateurs [44] ont résolu le problème en développant un système qui commande en amont du canal de décohérence, et qui permet de transférer le qubit intriqué. Ils ont ensuite mesuré l'environnement sonore associé au qubit intriqué dans le canal de décohérence. Dans leur étude [45], Villaseñor et Malaney ont utilisé des états non gaussiens variables continus comme états ressources de téléportation quantique pour téléporter des états cohérents et comprimés via des canaux bruyants.

Il est possible de démontrer la robustesse d'une technique de téléportation quantique bidirectionnelle, Bolokian et al. [46] l'ont fait afin de partager des états quantiques.

La téléportation quantique permet donc de commander et de sécuriser les systèmes dynamiques, en augmentant leurs vitesses de traitement, la fiabilité des données traitées et la défense contre les cyberattaques. Khoshnoud et al. [47] ont proposé une technique quantique pour sécuriser et améliorer la transmission des données dans la commande automatisée. Panda et al. [48] ont commandé à distance la direction d'un robot mobile à commande quantique à l'aide de l'algorithme de calcul quantique de Zidan. Quant à Khoshnoud et al. [49]; ils ont incorporé l'intrication quantique et la cryptographie quantique dans des systèmes mécaniques macroscopiques avant de téléporter des états quantiques via des canaux classiques afin d'améliorer la commande des systèmes dynamiques et l'autonomie des plates-formes mobiles. Hongtao et al. [50] ont développé un protocole sur la distribution bidirectionnelle de clés quantiques, les méthodes de téléportation quantique utilisées sur l'internet des véhicules (IoV) pour la sécurité des informations. Chiti et al. [51] ont utilisé la téléportation quantique et l'échange d'intrication entre drones pour créer des états intriqués ayant ensuite été utilisés pour la distribution de l'informatique quantique et du protocole E91. Mastriani et al. [52] ont créé et distribué deux photons intriqués à partir d'un satellite quantique et de deux drones au-dessus des nuages. Ensuite, chaque drone a créé sa paire de photons intriqués et a distribué par téléportation l'un d'entre eux à la station terrestre mobile la plus proche. Ciobanu et al. [53] ont également travaillé sur la communication quantique entre les satellites et la station au sol, grâce à un cadre qui permet la distribution des photons intriqués à deux nœuds distants via une constellation de satellites équipés de la capacité de retransmettre un faisceau de photons.

Dans l'article [54], les chercheurs ont utilisé des satellites comme serveurs centraux en développant des algorithmes quantiques variationnels pour l'apprentissage collaboratif local. L'apprentissage s'est fait en émergeant de vastes ensembles de données et de modèles ML. Ainsi, la téléportation quantique a garanti la transmission de données entre les stations à haute altitude et des drones acheminant les informations quantiques vers la station au sol. Gonzalez-Raya et al. [55] ont également étudié la téléportation quantique entre une station et un satellite pendant des turbulences aériennes. Leur étude portait sur la dégradation de l'intrication due à des erreurs dans la distribution. Ces erreurs pourraient être causées par des changements atmosphériques, des turbulences ou l'inefficacité des détecteurs. Les résultats ont révélé l'inefficacité de la téléportation quantique dans ces conditions. Pour compenser, une station intermédiaire a été incluse pour la génération d'état ou la focalisation du faisceau afin de réduire les impacts des turbulences atmosphériques. Mastriani et al. [56] ont démontré que la téléportation quantique peut être utilisée pour établir une communication entre des sous-marins et des satellites. Cela a été rendu possible grâce au développement d'un protocole de téléportation quantique utilisant deux sous-marins nucléaires immergés situés de chaque côté de l'océan, ainsi qu'un satellite quantique qui transmet des paires intriquées et des bits de désambiguïsation optique à des bouées situées à la surface de la mer, lesquelles sont associées aux sous-marins.

4.2 Notions de téléportation quantique

La téléportation quantique est l'un des protocoles fondamentaux de l'information quantique ; elle consiste en un déplacement d'un état quantique d'une position A vers une position B, quelle que soit la distance qui les sépare. Ce processus n'implique pas de déplacement

physique de l'état quantique. En fait, l'état quantique d'origine est détruit lors de sa mesure à la position de l'émetteur (position A), avant d'être reproduit à la position du récepteur (position B).

Pour téléporter un quelconque qubit d'une position A à une position B, deux autres qubits sont nécessaires :

$$|\Psi\rangle_i = \alpha_i|0\rangle + \beta_i|1\rangle, \quad i = 1, 2 \text{ et } 3, \quad (4.1)$$

$$\alpha_i = \cos \theta_i \text{ et } \beta_i = \sin \theta_i e^{i\phi_i}.$$

$|\Psi\rangle_i$ est le ième qubit.

$0 \leq \theta \leq \pi$, représente les amplitudes de probabilité des composantes d'état $|0\rangle$ et $|1\rangle$.

$0 \leq \phi \leq 2\pi$, représente la phase relative entre les composantes d'état $|0\rangle$ et $|1\rangle$.

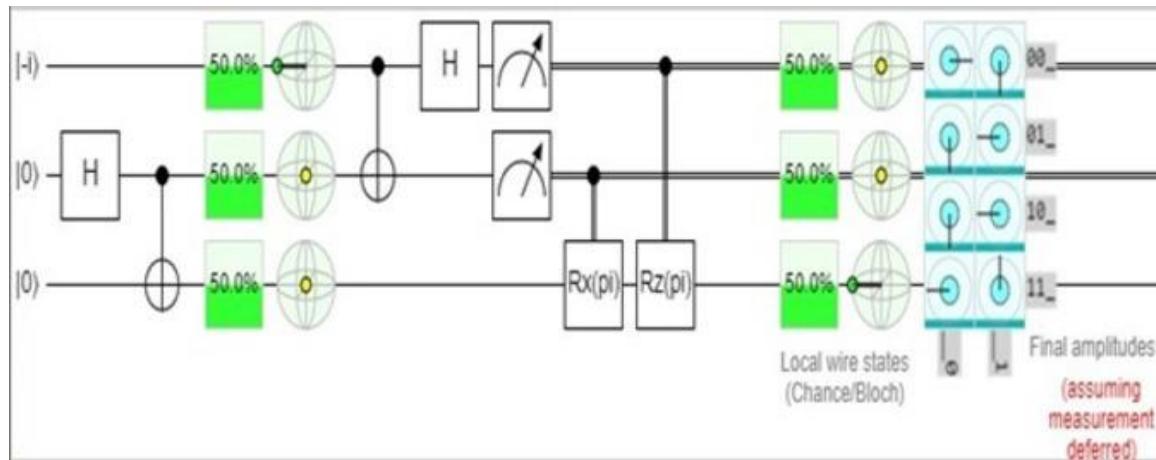


Figure 4.1: Circuit de téléportation quantique (Image réalisée avec Quirk).

La figure 4.1 illustre la téléportation réussie de $|-\rangle$ du début du circuit, jusqu'à sa fin.

Dans les deux positions, l'état du qubit $|-\rangle$ est observé avec une probabilité de 50% d'être dans l'état $|1\rangle$ après sa mesure. Les angles θ et ϕ ont été trouvés respectivement égaux à $+90^\circ$ et -90° . Ce circuit quantique a été construit et simulé sur Quirk. Ensuite, l'implémentation du circuit de téléportation quantique a été réalisée à l'aide de MATLAB

academic R2023b, utilisant des portes quantiques sous leurs formes matricielles. La téléportation d'un quelconque état quantique a été parfaitement réalisée. En d'autres termes, les amplitudes de probabilité des composantes d'état $|0\rangle$ et $|1\rangle$ et la phase relative entre les composantes d'état $|0\rangle$ et $|1\rangle$, sont textuellement reproduites lors de la mesure.

La conception d'un circuit de téléportation quantique, se fait grâce aux étapes suivantes :

Étape 1 : Création d'un état de Bell.

Tout d'abord, un état de superposition a été créé en initialisant les états du deuxième et du troisième qubit à $|0\rangle$. Ensuite, une porte d'Hadamard a été appliquée au deuxième qubit, avant d'appliquer une porte NON commandée au troisième qubit, avec le deuxième qubit commandé. La combinaison de ces opérations a créé l'état $|\phi^+\rangle$, qui est l'un des quatre états de Bell :

$$H|\psi\rangle_2 \rightarrow H|0\rangle = \frac{1}{\sqrt{2}}|0\rangle + \frac{1}{\sqrt{2}}|1\rangle \quad (4.2)$$

$$H|\psi\rangle_2 \otimes |\psi\rangle_3 \rightarrow (H \otimes \mathbb{I})(|\psi\rangle_2 \otimes |\psi\rangle_3) = \frac{1}{\sqrt{2}}|00\rangle + \frac{1}{\sqrt{2}}|10\rangle \quad (4.3)$$

$$(CNOT)(H \otimes \mathbb{I})(|\psi\rangle_2 \otimes |\psi\rangle_3) \rightarrow |\phi^+\rangle = \frac{1}{\sqrt{2}}|00\rangle + \frac{1}{\sqrt{2}}|11\rangle \quad (4.4)$$

Étape 2 : Réalisation d'une intrication.

Deux ou plusieurs systèmes quantiques sont intriqués lorsque leurs états ne peuvent pas être décrits indépendamment. En d'autres termes, l'état d'un système quantique est directement lié aux autres états, quelles que soient les distances qui les séparent.

Dans le cas présent, le premier qubit est intriqué à l'état de Bell trouvé dans l'équation 4.4, en appliquant une porte NON commandée au deuxième qubit, le premier étant le qubit commandé. Pour finir, une porte d'Hadamard est appliquée au premier qubit commandé.

Tout d'abord, le produit tensoriel entre le premier qubit de l'équation 4.1 et l'état de Bell de l'équation 4.1 est réalisé comme suit :

$$|\psi\rangle_1 \otimes |\phi^+\rangle = \frac{1}{\sqrt{2}}(\alpha|000\rangle + \alpha|011\rangle + \beta|100\rangle + \beta|111\rangle) \quad (4.5)$$

$$(CNOT \otimes \mathbb{I})(|\psi\rangle_1 \otimes |\phi^+\rangle) = \frac{1}{\sqrt{2}}(\alpha|000\rangle + \alpha|011\rangle + \beta|110\rangle + \beta|101\rangle) \quad (4.6)$$

$$\begin{aligned} (\mathbb{H} \otimes \mathbb{I} \otimes \mathbb{I})(CNOT \otimes \mathbb{I})(|\psi\rangle \otimes |\phi^+\rangle) \rightarrow & (\mathbb{H} \otimes \mathbb{I} \otimes \mathbb{I})\left(\frac{1}{\sqrt{2}}(\alpha|000\rangle + \alpha|011\rangle + \beta|110\rangle + \right. \\ & \left. \beta|101\rangle)\right) \end{aligned} \quad (4.7)$$

Les équations allant de 4.1 à 4.7 sont tirées ou inspirées de la littérature [57-59].

Les différentes composantes de l'équation 4.7 peuvent être développées comme suit :

$$\begin{aligned} \frac{\alpha}{\sqrt{2}}(\mathbb{H} \otimes \mathbb{I} \otimes \mathbb{I})|000\rangle \rightarrow & \frac{\alpha}{\sqrt{2}}H|0\rangle \otimes \mathbb{I}|0\rangle \otimes \mathbb{I}|0\rangle = \frac{\alpha}{\sqrt{2}}\left(\frac{1}{\sqrt{2}}|0\rangle + \frac{1}{\sqrt{2}}|1\rangle\right) \otimes |0\rangle \otimes |0\rangle = \\ \frac{\alpha}{2}(|000\rangle + |100\rangle) \end{aligned} \quad (4.8)$$

$$\begin{aligned} \frac{\alpha}{\sqrt{2}}(\mathbb{H} \otimes \mathbb{I} \otimes \mathbb{I})|011\rangle \rightarrow & \frac{\alpha}{\sqrt{2}}H|0\rangle \otimes \mathbb{I}|1\rangle \otimes \mathbb{I}|1\rangle = \frac{\alpha}{\sqrt{2}}\left(\frac{1}{\sqrt{2}}|0\rangle + \frac{1}{\sqrt{2}}|1\rangle\right) \otimes |1\rangle \otimes |1\rangle = \\ \frac{\alpha}{2}(|011\rangle + |111\rangle) \end{aligned} \quad (4.9)$$

$$\begin{aligned} \frac{\beta}{\sqrt{2}}(\mathbb{H} \otimes \mathbb{I} \otimes \mathbb{I})|110\rangle \rightarrow & \frac{\beta}{\sqrt{2}}H|1\rangle \otimes \mathbb{I}|1\rangle \otimes \mathbb{I}|0\rangle = \frac{\beta}{\sqrt{2}}\left(\frac{1}{\sqrt{2}}|0\rangle - \frac{1}{\sqrt{2}}|1\rangle\right) \otimes |1\rangle \otimes |0\rangle = \\ \frac{\beta}{2}(|010\rangle - |110\rangle) \end{aligned} \quad (4.10)$$

$$\begin{aligned} \frac{\beta}{\sqrt{2}}(\mathbb{H} \otimes \mathbb{I} \otimes \mathbb{I})|101\rangle \rightarrow & \frac{\beta}{\sqrt{2}}H|1\rangle \otimes \mathbb{I}|0\rangle \otimes \mathbb{I}|1\rangle = \frac{\beta}{\sqrt{2}}\left(\frac{1}{\sqrt{2}}|0\rangle - \frac{1}{\sqrt{2}}|1\rangle\right) \otimes |0\rangle \otimes |1\rangle = \\ \frac{\beta}{2}(|001\rangle - |101\rangle) \end{aligned} \quad (4.11)$$

L'équation 4.7 peut être exprimée comme suit :

$$\begin{aligned} (\mathbb{H} \otimes \mathbb{I} \otimes \mathbb{I})(CNOT \otimes \mathbb{I})(|\psi\rangle \otimes |\phi^+\rangle) = & \frac{\alpha}{2}(|000\rangle + |100\rangle + |011\rangle + |111\rangle) + \\ & \frac{\beta}{2}(|010\rangle - |110\rangle + |001\rangle - |101\rangle) \end{aligned} \quad (4.12)$$

Pour une meilleure compréhension, l'équation 4.12 sera écrite sous la forme suivante :

$$(H \otimes I \otimes I)(CNOT \otimes I)(|\psi\rangle \otimes |\phi^+\rangle) = \frac{1}{2} [|00\rangle(\alpha|0\rangle + \beta|1\rangle) + |01\rangle(\alpha|1\rangle + \beta|0\rangle) + |10\rangle(\alpha|0\rangle - \beta|1\rangle) + |11\rangle(\alpha|1\rangle - \beta|0\rangle)] \quad (4.13)$$

Étape 3 : Mesure des états quantiques.

La mesure d'un état quantique consiste en la détermination de son état classique. Lorsque ce processus est effectué, l'état quantique pouvant être en état de superposition, change en état classique ayant le taux de probabilité le plus élevé. Le cas échéant, lors de la mesure du premier et du deuxième qubit, le troisième qubit recevra les résultats sous la forme d'états de base d'un système quantique à trois qubits, qui sont $|00\rangle$, $|01\rangle$, $|10\rangle$ et $|11\rangle$.

La mesure quantique est irréversible ; elle détruit l'information quantique encodée dans le système, pour ne conserver que l'information classique. Pour recréer l'état du qubit détruit, les portes de Pauli X et Z sont appliquées au troisième état du qubit :

Si le troisième qubit reçoit le résultat de la mesure $|00\rangle$, il y applique la porte d'identité, ce qui équivaut à ne pas effectuer d'opération :

$$I(\alpha|0\rangle + \beta|1\rangle) \rightarrow (\alpha|0\rangle + \beta|1\rangle) \quad (4.14)$$

Si le troisième qubit reçoit le résultat de la mesure $|01\rangle$, il y applique la porte de Pauli X, qui permute les états de α et β :

$$X(\alpha|1\rangle + \beta|0\rangle) \rightarrow (\alpha|0\rangle + \beta|1\rangle) \quad (4.15)$$

Si le troisième qubit reçoit le résultat de la mesure $|10\rangle$, il y applique la porte de Pauli Z, qui change le signe de β :

$$Z(\alpha|0\rangle - \beta|1\rangle) \rightarrow (\alpha|0\rangle + \beta|1\rangle) \quad (4.16)$$

Si le troisième qubit reçoit le résultat de la mesure $|11\rangle$, il y applique la combinaison des portes ZX, qui permute les états α et β , puis change le signe de β :

$$ZX(\alpha|1\rangle - \beta|0\rangle) \rightarrow Z(\alpha|0\rangle - \beta|1\rangle) \rightarrow (\alpha|0\rangle + \beta|1\rangle) \quad (4.17)$$

Après avoir mesuré le premier et le deuxième qubit, puis effectué les portes de Pauli X, Z ou leurs combinaisons sur le troisième qubit, l'état $(\alpha|0\rangle + \beta|1\rangle)$, qui correspond à l'état du qubit initial $|\psi\rangle$ de l'équation 4.1, a été retrouvé dans tous les quatre cas. Ce qui prouve que l'état du qubit a été téléporté avec succès.

4.3 Architecture du système

L'objectif est d'échanger les paramètres de vitesse et les coordonnées de position, sous forme d'informations quantiques, entre la station de commande et le robot mobile. Le système proposé se compose de trois parties indépendantes, à savoir la station de commande, le circuit de téléportation quantique et le robot mobile, tel que présenté dans la figure 4.2.

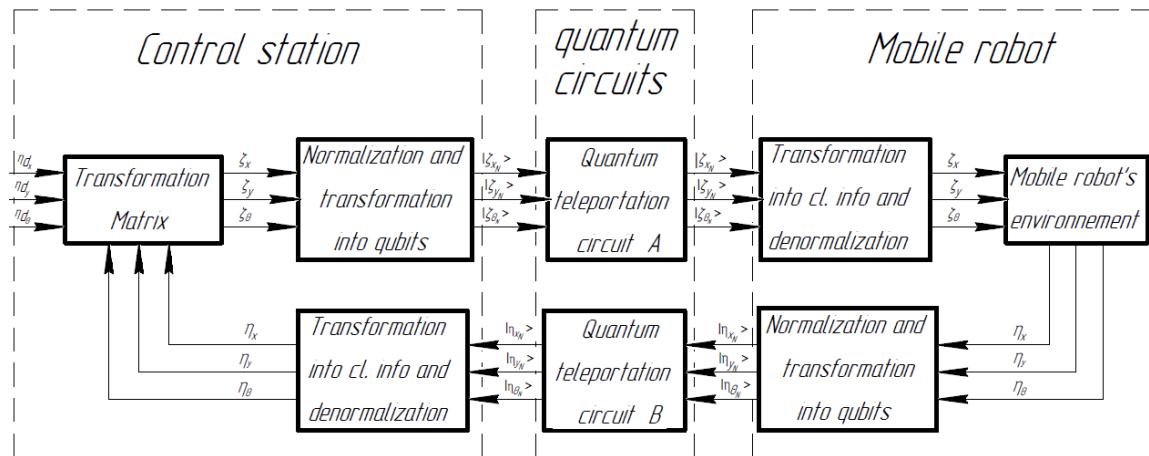


Figure 4.2 : Schéma de l'architecture du système.

4.3.1 Station de commande

La station de commande est la partie fixe de ce système. Elle interagit avec l'utilisateur, lui permettant d'entrer des informations sur la trajectoire à suivre. Cette partie permet à l'utilisateur de changer les paramètres de simulation.

Avant de téléporter la commande des vitesses de l'équation 3.21, il est nécessaire de les normaliser afin de pouvoir les représenter facilement sous forme de qubits. La normalisation de la vitesse longitudinale nonholonomique est effectuée selon la relation suivante :

$$u_{nN} = \frac{u_n}{\max of u_n} \quad (4.18)$$

La normalisation de la vitesse latérale nonholonomique est effectuée de la manière suivante:

$$v_{nN} = \frac{v_n}{\max of v_n} \quad (4.19)$$

La normalisation de la vitesse angulaire nonholonomique est effectuée selon la relation suivante :

$$r_{nN} = \frac{r_n}{\max of r_n} \quad (4.20)$$

$\max of u_n$, $\max of v_n$ et $\max of r_n$ sont respectivement les valeurs maximales de la vitesse longitudinale nonholonomique, la vitesse latérale nonholonomique et la vitesse angulaire nonholonomique.

La transformation des vitesses normalisées en qubit se fait en utilisant la formule du qubit aléatoire:

$$|\Psi\rangle = \begin{pmatrix} \cos \theta \\ \sin \theta e^{i\phi} \end{pmatrix} \quad (4.21)$$

Pour simplifier les calculs, l'angle ϕ est supposé être toujours nul. Pour transformer les vitesses normalisées en qubits, l'identité trigonométrique à double angle du cosinus est utilisée :

$$\cos(2\alpha) = \cos^2 \alpha - \sin^2 \alpha \quad (4.22)$$

$$\cos(2\alpha_1) = u_{nN},$$

En considérant $\cos(2\alpha_1)$ égale à u_{n_N} , on obtient la vitesse longitudinale normalisée sous forme de qubit :

$$|u_{n_N}\rangle = \begin{pmatrix} \cos(1/2 \cdot \cos^{-1}(u_{n_N})) \\ \sin(1/2 \cdot \cos^{-1}(u_{n_N})) \end{pmatrix} \quad (4.23)$$

La vitesse latérale normalisée sous forme de qubit sera la suivante :

$$|v_{n_N}\rangle = \begin{pmatrix} \cos(1/2 \cdot \cos^{-1}(v_{n_N})) \\ \sin(1/2 \cdot \cos^{-1}(v_{n_N})) \end{pmatrix} \quad (4.24)$$

La vitesse angulaire normalisée sous forme de qubit sera :

$$|r_{n_N}\rangle = \begin{pmatrix} \cos(1/2 \cdot \cos^{-1}(r_{n_N})) \\ \sin(1/2 \cdot \cos^{-1}(r_{n_N})) \end{pmatrix} \quad (4.25)$$

4.3.2 Robot mobile de type voiture

Le robot mobile reçoit les données sous forme quantique. Il est important de les remettre sous forme classique, afin que le robot puisse les traiter. Cette opération est réalisée pour dénormaliser les vitesses téléportées.

Pour les vitesses longitudinale, latérale et angulaire téléportées, les valeurs dénormalisées sont respectivement calculées comme dans les équations 4.26, 4.27 et 4.28 :

$$u_{n_D} = \max of u_n \cdot (\cos^2(1/2 \cdot \cos^{-1}(u_{n_N})) - \sin^2(1/2 \cdot \cos^{-1}(u_{n_N}))) \quad (4.26)$$

$$v_{n_D} = \max of v_n \cdot (\cos^2(1/2 \cdot \cos^{-1}(v_{n_N})) - \sin^2(1/2 \cdot \cos^{-1}(v_{n_N}))) \quad (4.27)$$

$$r_{n_D} = \max of r_n \cdot (\cos^2(1/2 \cdot \cos^{-1}(r_{n_N})) - \sin^2(1/2 \cdot \cos^{-1}(r_{n_N}))) \quad (4.28)$$

Après avoir traité les informations téléportées, le robot commence à se déplacer selon l'équation 3.23. Les données de sa position sont enregistrées et téléportées à la station de

commande, afin de corriger les erreurs dans le suivi de la trajectoire. Les normalisations des coordonnées réelles du robot par rapport à l'axe x, de y et de θ sont trouvées dans les équations :

$$X_N = \frac{X}{\max of X}, \text{ with } \max of X = 10.0783m \quad (4.29)$$

$$Y_N = \frac{Y}{\max of Y}, \text{ with } \max of Y = 20.1561m \quad (4.30)$$

$$\theta_N = \frac{\theta}{\max of \theta}, \text{ with } \max of \theta = 6.0110rad \quad (4.31)$$

Les coordonnées normalisées de la position du robot seront ensuite transformées en qubits qui seront téléportés du robot mobile à la station de commande. Pour ce faire, la même analogie de l'identité trigonométrique à double angle du cosinus est utilisée comme suit :

$$|X_N\rangle = \begin{pmatrix} \cos(1/2 \cdot \cos^{-1}(X_N)) \\ \sin(1/2 \cdot \cos^{-1}(X_N)) \end{pmatrix} \quad (4.32)$$

$$|Y_N\rangle = \begin{pmatrix} \cos(1/2 \cdot \cos^{-1}(Y_N)) \\ \sin(1/2 \cdot \cos^{-1}(Y_N)) \end{pmatrix} \quad (4.33)$$

$$|\theta_N\rangle = \begin{pmatrix} \cos(1/2 \cdot \cos^{-1}(\theta_N)) \\ \sin(1/2 \cdot \cos^{-1}(\theta_N)) \end{pmatrix} \quad (4.34)$$

4.3.3 Circuit de téléportation quantique

Le circuit de téléportation quantique peut établir une connexion sécurisée et de haute qualité, permettant le transfert instantané de données entre le robot et sa station de commande. Un autre avantage de la téléportation quantique par rapport à la communication sans fil classique, est la fidélité des données téléportées. Pour l'algorithme de téléportation proposé, l'équation 4.7 est utilisée sous forme matricielle :

$$(H \otimes I \otimes I)(CNOT \otimes I)(|\psi\rangle \otimes |\phi^+\rangle) = \begin{bmatrix} \alpha_{11}/2 \\ \beta_{11}/2 \\ \beta_{12}/2 \\ \alpha_{12}/2 \\ \alpha_{13}/2 \\ -\beta_{13}/2 \\ -\beta_{14}/2 \\ \alpha_{14}/2 \end{bmatrix}, \text{ with } |\psi\rangle_{11} = \begin{bmatrix} \alpha_{11}/2 \\ \beta_{11}/2 \end{bmatrix} \quad (4.35)$$

Les mesures du qubit d'état aléatoire $|\psi\rangle$ et du second qubit du circuit $|0\rangle$ sont effectuées en recherchant les plus grandes probabilités. Supposons que le résultat de la mesure de $|\psi\rangle_i$ soit égal à 0. Cela signifie que la probabilité que $|\psi\rangle_i$ soit dans l'état classique $|0\rangle$ est supérieure ou égale à celle d'être dans l'état $|1\rangle$. Dans ce cas, l'expression mathématique suivante $(\alpha_i)^2 \geq (\beta_i)^2$ est vraie. Par analogie, si le résultat de la mesure de $|\psi\rangle_i$ est égal à 1, la probabilité que $|\psi\rangle_i$ soit dans l'état classique $|0\rangle$ est inférieure à sa probabilité d'être $|1\rangle$. Dans ce cas, l'expression mathématique suivante $(\alpha_i)^2 < (\beta_i)^2$ est vraie.

Sachant que :

La probabilité que $|\psi\rangle_i$ soit dans l'état classique $|0\rangle$ est égale à $(\alpha_i)^2$.

La probabilité que $|\psi\rangle_i$ soit dans l'état classique $|1\rangle$ est égale à $(\beta_i)^2$.

Après les mesures, si les résultats de $|\psi\rangle_1$ et du deuxième qubit du circuit $|0\rangle$ sont égaux à 0, alors ce qui suit est vrai :

$$|\psi\rangle_{1\,tel} = 2 \cdot I \cdot |\psi\rangle_{11} = 2 \begin{pmatrix} 1 & 0 \\ 0 & 1 \end{pmatrix} \begin{pmatrix} \alpha_{11}/2 \\ \beta_{11}/2 \end{pmatrix} = \begin{pmatrix} \alpha_{11} \\ \beta_{11} \end{pmatrix} = \begin{pmatrix} \alpha \\ \beta \end{pmatrix} \quad (4.36)$$

I représente la matrice identitaire.

Dans le cas où le résultat de la mesure de $|\psi\rangle_1$ est égal à 0 et que le résultat de la mesure du second qubit du circuit $|0\rangle$ est égal à 1, l'état téléporté du qubit aléatoire est détaillé dans l'équation qui suit.

$$|\psi\rangle_{1\text{ tel}} = 2 \cdot X \cdot |\psi\rangle_{12} = 2 \begin{pmatrix} 0 & 1 \\ 1 & 0 \end{pmatrix} \begin{pmatrix} \beta_{12}/2 \\ \alpha_{12}/2 \end{pmatrix} = \begin{pmatrix} \alpha_{12} \\ \beta_{12} \end{pmatrix} = \begin{pmatrix} \alpha \\ \beta \end{pmatrix} \quad (4.37)$$

X est la porte de Pauli-X dans sa forme matricielle.

Dans le cas où le résultat de la mesure de $|\psi\rangle_1$ est égal à 1 et que le résultat de la mesure du second qubit du circuit $|0\rangle$ est égal à 0, l'état téléporté du qubit aléatoire est présenté dans l'équation 4.38.

$$|\psi\rangle_{1\text{ tel}} = 2 \cdot Z \cdot |\psi\rangle_{13} = 2 \begin{pmatrix} 1 & 0 \\ 0 & -1 \end{pmatrix} \begin{pmatrix} \alpha_{13}/2 \\ -\beta_{13}/2 \end{pmatrix} = \begin{pmatrix} \alpha_{13} \\ \beta_{13} \end{pmatrix} = \begin{pmatrix} \alpha \\ \beta \end{pmatrix} \quad (4.38)$$

Z est la porte de Pauli-Z dans sa forme matricielle.

Dans le cas où les résultats de la mesure de $|\psi\rangle_1$ et de la mesure du second qubit du circuit $|0\rangle$ sont égaux à 1, l'état téléporté du qubit aléatoire est présenté dans l'équation qui suit.

$$|\psi\rangle_{1\text{ tel}} = 2 \cdot Z \cdot X \cdot |\psi\rangle_{14} = 2 \begin{pmatrix} 1 & 0 \\ 0 & -1 \end{pmatrix} \begin{pmatrix} 0 & 1 \\ 1 & 0 \end{pmatrix} \begin{pmatrix} -\beta_{14}/2 \\ \alpha_{14}/2 \end{pmatrix} = \begin{pmatrix} \alpha_{14} \\ \beta_{14} \end{pmatrix} = \begin{pmatrix} \alpha \\ \beta \end{pmatrix} \quad (4.39)$$

Dans les quatre cas présentés, la téléportation du qubit d'état aléatoire est réussie, démontrant les performances du circuit proposé.

Au cours du processus de retour de l'information vers la station de commande, celle-ci reçoit des données sur la position sous leur forme quantique. Pour les transformer en information classique, la dénormalisation des données téléportées est nécessaire. Pour ce faire, les téléportations des positions par rapport à l'axe x, à l'axe y et à l'angle θ sont respectivement effectuées conformément aux équations 4.40 à 4.42.

$$X_D = \max of X \cdot (\cos^2(1/2 \cdot \cos^{-1}(X_N)) - \sin^2(1/2 \cdot \cos^{-1}(X_N))) \quad (4.40)$$

$$Y_D = \max of Y \cdot (\cos^2(1/2 \cdot \cos^{-1}(Y_N)) - \sin^2(1/2 \cdot \cos^{-1}(Y_N))) \quad (4.41)$$

$$\theta_D = \max of \theta \cdot (\cos^2(1/2 \cdot \cos^{-1}(\theta_N)) - \sin^2(1/2 \cdot \cos^{-1}(\theta_N))) \quad (4.42)$$

La station de commande dispose au préalable de la position souhaitée du robot. Ainsi, en fonction de cette information, des données de vitesse sont envoyées au robot mobile. Les paramètres dynamiques, tels que la dynamique des moteurs, le moment d'inertie, la répartition de la masse, le frottement et la traction, ainsi que les facteurs environnementaux, influencent le mouvement du robot. Cela se traduit par des positions réelles qui ne coïncident pas avec les positions souhaitées. Par conséquent, les positions réelles sont utilisées pour corriger les erreurs dans le suivi de la trajectoire :

$$\varepsilon_T = \eta_d - \eta_T \quad \text{et} \quad \dot{\varepsilon}_T = \dot{\eta}_d - \dot{\eta}_T, \quad \text{Où } \dot{\eta}_T = J(\theta) \cdot \zeta_T \quad (4.43)$$

Où :

$\varepsilon_T, \dot{\varepsilon}_T$: L'erreur de la position téléportée et l'erreur de la vitesse téléportée;

η_d, η_T : La position souhaitée et la position téléportée réelle;

$\dot{\eta}_d, \dot{\eta}_T$: La vitesse souhaitée et la vitesse téléportée réelle.

Chapitre 5 – Résultats et discussion

5.1 ANN et QANN pour l'optimisation des paramètres en vue d'un suivi des trajectoires d'un robot mobile de type voiture

5.1.1 Paramètres de simulation

Pour comparer les performances à prédire la masse et le moment d'inertie d'un robot, certains paramètres des deux ANNs ont été fixés dans le tableau 5.1 :

Tableau 5.1. Les paramètres de l'architecture des ANNs :

Paramètres	Nombre
Nœuds d'entrée	20
Nœuds de la couche intermédiaire	20
Nœuds de la sortie	2
Taux d'apprentissage	0.0050233 pour CANN 0.0091494 pour QANN
Epochs	5000

Plus de 100 robots ayant des masses et des moments d'inertie différents ont essayé de suivre la même trajectoire sans aucun processus d'optimisation. Ainsi, le choix de 20 nœuds pour les couches d'entrée correspond aux différentes positions des robots dans leurs trajectoires respectives à des moments précis. Le nombre de nœuds dans la couche d'entrée doit être suffisamment élevé pour que l'apprentissage soit optimal. Les 20 nœuds de la couche intermédiaire et le taux d'apprentissage ont été choisis à l'aide des algorithmes PSO/QPSO, qui trouvent les paramètres avec lesquels le modèle fait les meilleures prédictions pendant l'apprentissage.

Une base de données de 50 ensembles a été traitée en 5000 itérations avec des taux d'apprentissage déterminés à l'aide des algorithmes d'optimisation des essaims de particules. Ainsi, après avoir entraîné notre modèle, une autre base de données de 21 ensembles a été utilisée pour valider la performance du QANN et celle du CANN. Il est important de souligner que la base de données de 21 ensembles n'a pas été traitée au préalable par les deux ANNs.

5.1.2 Résultats et discussion

5.1.2.1 Base de données pour l'apprentissage

La fonction de coût est un excellent indicateur de performance des ANNs. Ainsi, la figure 5.1 montre l'évolution de la fonction de coût de CANN et celle de QANN, au fil des itérations. Dans notre cas, ces fonctions de coût ont été calculées comme les erreurs quadratiques moyennes entre les valeurs prédites et les valeurs réelles.

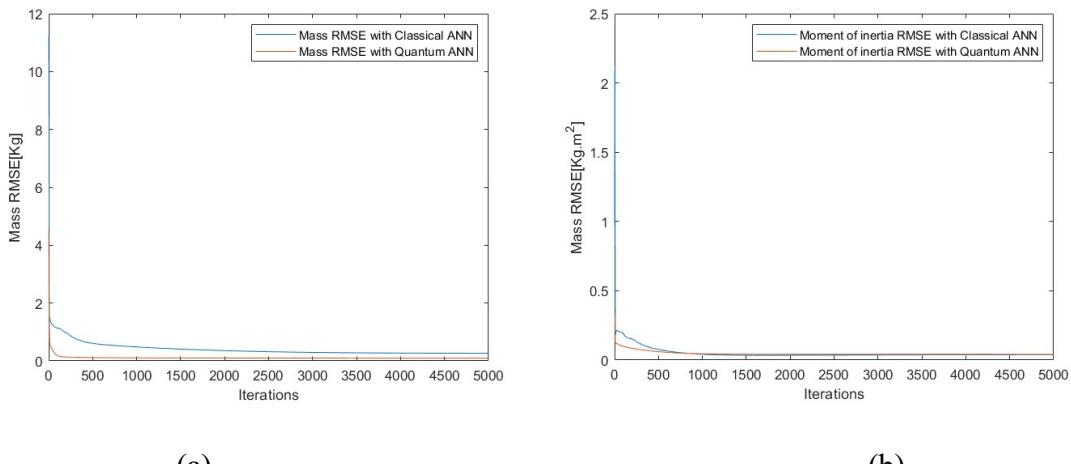


Figure 5.1 : l'évolution au fil des itérations des fonctions de coût pour : (a) les masses et (b) les moments d'inertie.

Figure 5.1 (a) présente la performance de CANN et celle de QANN à réduire leurs fonctions de coût dans l'estimation des masses. Les deux ANNs ont montré un comportement similaire, réduisant leurs fonctions de coût qui partaient d'une erreur

moyenne de 4.7 Kg pour QANN et de 10 Kg pour CANN. Ensuite, les deux fonctions descendent drastiquement à 0.025 Kg pour QANN et à 0.8 Kg pour CANN à la 75^e itération. Au fil des itérations, les erreurs moyennes commencent à décroître plus lentement, atteignant ainsi une valeur de 0.012 Kg pour QANN et de 0.21 Kg pour CANN à la 5000^e itération. Figure 5.1 (b) montre la performance de CANN et celle QANN à réduire leurs fonctions de coût dans l'estimation des moments d'inertie. Les deux ANNs ont également montré un comportement semblable, réduisant les fonctions de coût qui partaient d'une erreur moyenne de 0.5 Kg · m² pour QANN et de 2.1 Kg · m² pour CANN à la première itération. Ensuite, les deux fonctions descendent à 0.026 Kg · m² pour QANN et à 0.054 Kg · m² pour CANN à la 500^e itération. Comme pour figure 5.1(a), les erreurs moyennes commencent à décroître progressivement au fil des itérations, atteignant ainsi 0.012 Kg · m² pour QANN et 0.029 Kg · m² pour CANN à la 5000^e itération.

Les deux ANNs ont été exécutés plus de 30 fois. Dans la totalité des cas, le QANN a mieux performé en réduisant plus efficacement les fonctions de coût dans les estimations des masses et dans les estimations des moments d'inertie. Le CANN réduisait les fonctions de coût à des valeurs acceptables. Cependant CANN a une tendance à tomber dans des minima locaux.

Autre manière de montrer la supériorité du QANN par rapport au CANN, serait de comparer leurs prédictions aux données réelles telles que présentées dans la figure 5.2 et 5.3:

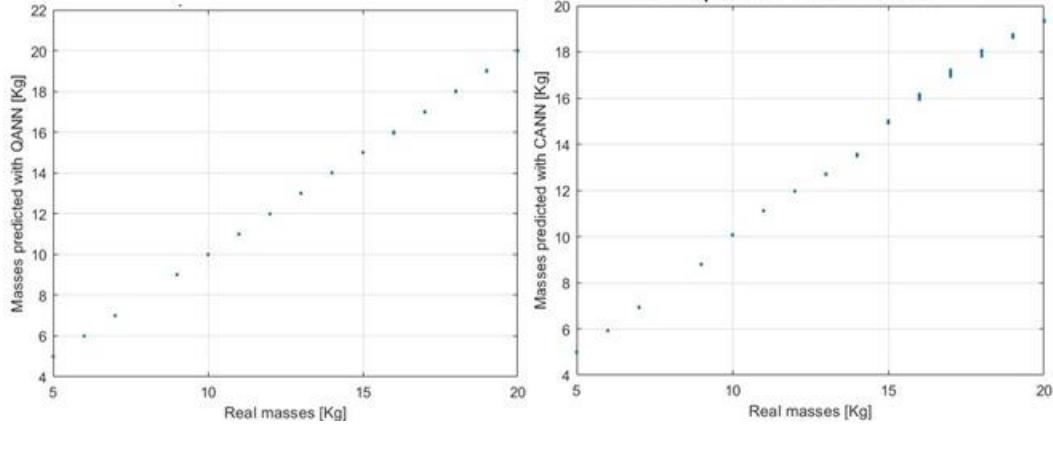


Figure 5.2 : Comparaison entre les données des masses réelles et celles prédites par : (a) QANN et (b) CANN.

La figure 5.2 (a) montre que pour chacune des masses réelles présentées sur l'axe des abscisses, QANN prédit les valeurs avec une précision de 0.012 Kg près, dans l'axe des ordonnées. En ce qui concerne CANN, Figure 5.2 (b) montre que pour chacune des masses réelles présentées dans l'axe des abscisses, CANN prédit les valeurs équivalentes avec une erreur de 0.14 Kg dans l'axe des ordonnées. CANN prédit avec précision les masses comprises entre 5 à 15 KG, pendant que QANN prédit avec une meilleure précision toutes les masses.

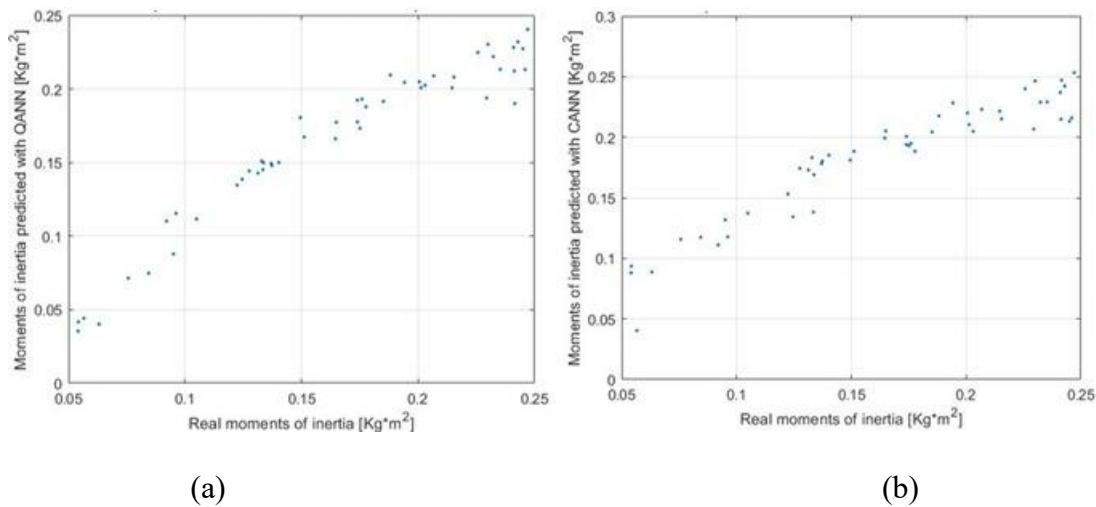


Figure 5.3 : Comparaison entre les données sur les moments d'inertie réelles et celles prédites par : (a) QANN et (b) CANN.

Les figures 5.3 montrent que les deux ANNs parviennent à prédire différents moments d'inertie avec une précision acceptable de $0.0176 \text{ Kg} \cdot m^2$ pour QANN et de $0.029 \text{ Kg} \cdot m^2$ pour CANN. Comme pour la masse, le QANN parvient à prédire les moments d'inertie plus efficacement dans la totalité des cas.

Pendant l'apprentissage, les performances du QANN surpassent celles du CANN, fournissant des fonctions de coût plus basses. Ainsi, les valeurs prédites par QANN se rapprochent le plus des valeurs réelles de la base de données d'apprentissage.

5.1.2.2 Base de données pour la validation

Après avoir entraîné les deux modèles des ANNs, une base de données qui n'avait jamais été traitée au préalable a été utilisée pour valider les capacités des deux ANNs à généraliser leurs prédictions respectives.

Figure 5.4 présente et compare les prédictions de données conservées pour la validation.

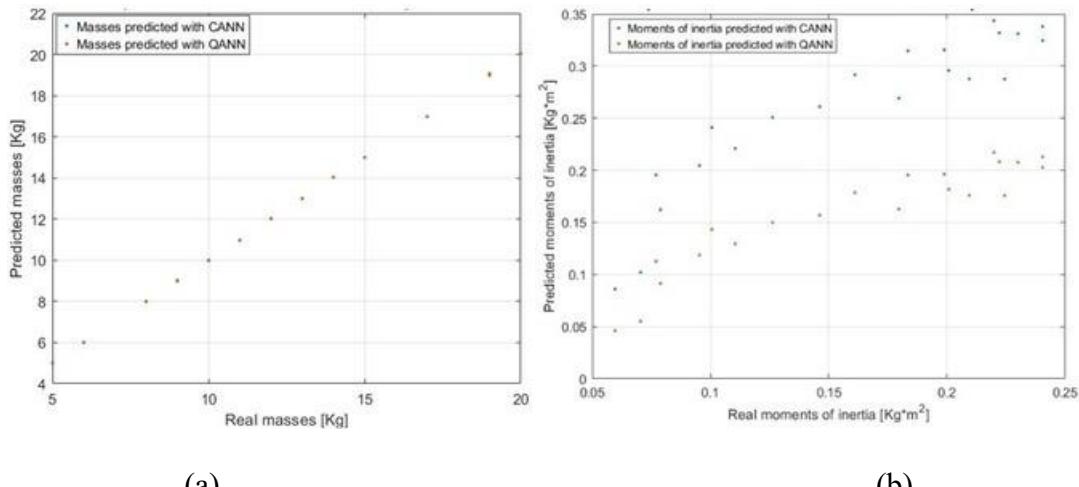


Figure 5.4 : Comparaison entre les données réelles et celles prédites pour : (a) les masses et (b) les moments d'inertie.

La figure 5.4 (a) montre une correspondance parfaite entre les masses prédictes par les deux ANNs. En fait, pour la base de données de validation, QANN et CANN ont tous deux prédit les masses avec une fonction de cout de 0,0237 kg. Ces résultats montrent que QANN a bien équilibré les valeurs prédictes avec la base de données d'apprentissage et celle de validation. Le CANN a, quant à lui, mieux prédit avec la base de données de validation. La figure 5.4 (b) montre que QANN a prédit les moments d'inertie avec une fonction de cout de $0,0246 \text{ Kg} \cdot m^2$, ce qui est raisonnable si l'on considère la fonction de coût de $0,0176 \text{ Kg} \cdot m^2$ obtenue avec la base de données d'apprentissage. Pour CANN, les moments d'inertie ont été prédits avec une fonction de coût de $0,103 \text{ Kg} \cdot m^2$. Une différence significative entre les prédictions faites avec la base de données d'apprentissage et celles faites avec la base de données de validation est constatée. Ainsi, les résultats présentés montrent que QANN s'équilibre bien en évitant le sous-ajustement et le surajustement des paramètres. Tandis que, CANN fait un surajustement des paramètres, ce qui conduit à des prédictions peu fiables. Les résultats du QANN sont prometteurs pour diverses tâches complexes de régression et de classification. QANN permettra donc d'éviter le surapprentissage, qui consiste à considérer les bruits comme des données pertinentes.

5.2 Optimisation des paramètres du régulateur PD basée sur la version quantique du PSO pour le suivi des trajectoires d'un robot mobile de type voiture

5.2.1 Paramètres de simulation

Pour comparer le PSO au QPSO implémentés pour la correction d'erreur dans le suivi des trajectoires de notre robot mobile de type voiture, le logiciel MATLAB academic R2023b a été choisi comme plateforme de simulation.

le tableau 5.2 contient les valeurs des paramètres du robot mobile.

Tableau 5.2 : Les paramètres du robot CLMR.

Paramètre (Unité)	Valeur
Masse (kg)	10
Inertie (kg.m ²)	0.1
Longueur (m)	0.3
Largeur (m)	0.2
Coefficient de frottement visqueux (N.s/m)	0.475

Les paramètres des algorithmes PSO/QPSO sont respectivement présentés dans le tableau 5.3 et le tableau 5.4.

Tableau 5.3 : Les paramètres des algorithmes PSO/QPSO dans des conditions idéales.

Paramètre (Définition)	Valeur
W (poids d'inertie)	0.3
C_1 (Poids cognitif ou personnel)	1.5
C_2 (poids social)	1.5
Plage d'initialisation des particules	Entre [0,500] pour le PSO et [0,π/2] pour le QPSO
Plage d'initialisation des vitesses des particules	Entre [0,250] pour le PSO et [0,π/4] pour le QPSO

Tableau 5.4 : Les paramètres des algorithmes PSO/QPSO dans des conditions non- idéales.

Parameter (Definition)	Value
W (poids d'inertie)	0.3
C_1 (Poids cognitif ou personnel)	1.5
C_2 (poids social)	1.5
Plage d'initialisation des particules	0
Plage d'initialisation des vitesses des particules	Entre [0,5] pour le PSO et [0,(π/2) · 0.01] pour le QPSO

5.2.2 Résultats sans algorithme de commande

Pour la première simulation, la constante de la commande proportionnelle et celle de la commande dérivée ont été fixées à zéro afin d'observer les effets des contraintes dynamiques sur le robot pendant son suivi de trajectoire. La figure 5.5 représente la trajectoire, un cercle parfait de 20 mètres de diamètre.

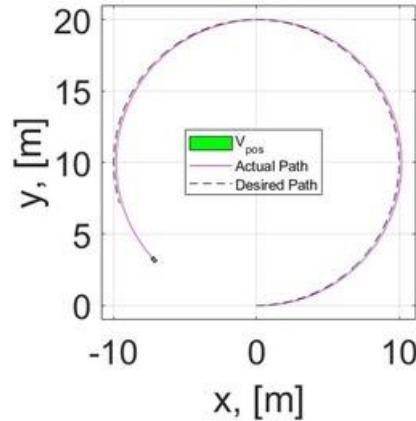


Figure 5.5 : Trajectoire du robot avant optimisation.

Lorsque les paramètres de commande du régulateur PD sont fixés à zéro, le RMSE connaît une croissance linéaire lente et régulière dans le temps. Le RMSE va de 0 à 0.2307 m^2 à la fin de la simulation, tel que l'illustre dans la figure 5.6.

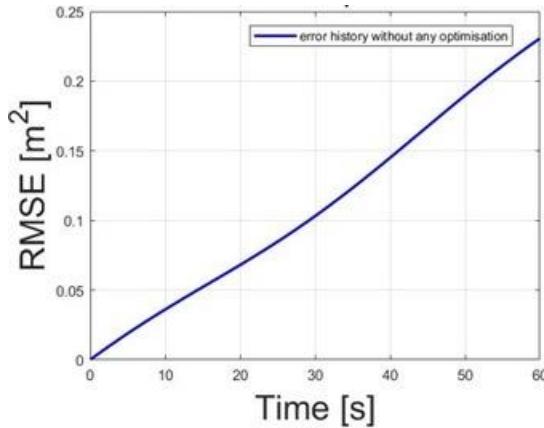


Figure 5.6 : Évolution du RMSE avant l'optimisation.

Sans algorithme d'optimisation, le robot s'écarte de la trajectoire souhaitée, comme le montre la figure 5.5. La force longitudinale et le moment d'inertie du robot ne sont pas correctement réglés, ce qui entraîne des couples excessifs sur le robot. Cela cause un écart évident puisque la position du robot à l'instant t dépasse largement la position souhaitée. Le RMSE a été calculé en connaissant la différence entre la position souhaitée et la position réelle du robot. Les paramètres de commande optimisés sont censés améliorer le suivi de la trajectoire en influençant avec précision les forces latérales, longitudinales ainsi que le moment d'inertie.

La figure 5.7 fournit des informations sur les forces et le moment d'inertie appliqués au robot. Un test de robustesse a été introduit en diminuant l'inertie du robot de 15% à la 20e seconde et en augmentant sa masse de 15% à la 35e seconde. Ces changements ont eu un impact considérable sur les forces et le moment d'inertie du robot. Au début de la simulation, la force longitudinale du robot a été mesurée comme étant égale à 9,47737 N. À la 35e seconde, cette force constante est brusquement passée à 10,8274 N, corrigeant ainsi l'erreur introduite dans l'estimation de la masse du robot. La force est ensuite restée constante jusqu'à la fin de la simulation.

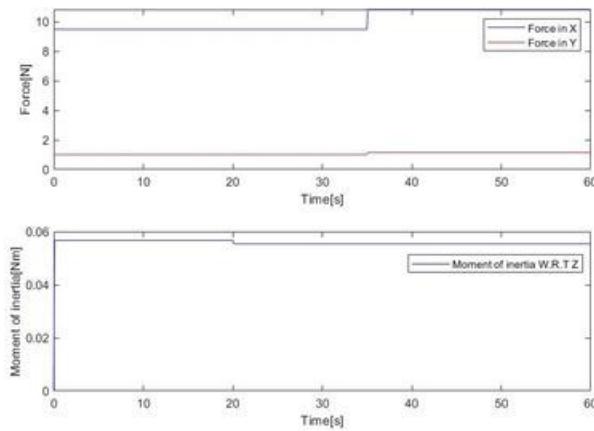


Figure 5.7 : Forces et couple appliqués au robot au fil du temps pour le suivi de la trajectoire circulaire.

Le CLMR est un véhicule nonholonomique, ce qui signifie qu'aucune force latérale n'est appliquée intentionnellement au robot. Cependant, une petite force de 1,00499 N est apparue au début de la simulation, avant de passer à 1,15574 N à la 35e seconde. Cette force résulte des effets combinés de la vitesse longitudinale et angulaire du robot. Quant au moment d'inertie, il était initialement mesuré à 0 Nm avant de monter à 0,0567 Nm à la 0,1e seconde. Ensuite, le moment d'inertie est resté constant avant de tomber à 0,005538 Nm entre la 20e et la 20.1e seconde. Par la suite, le moment d'inertie est resté le même jusqu'à la fin de la simulation. Cette expérience montre la capacité du système à répondre aux changements brusques de son environnement, même sans optimisation des paramètres du régulateur PD.

5.2.3 Résultats et discussion sur l'utilisation du régulateur PD basé sur PSO et QPSO pour le suivi d'une trajectoire circulaire

La comparaison entre le PSO et le QPSO consiste à trouver lequel de ces deux algorithmes donnera les meilleures valeurs de K_p et K_d pour réduire le RMSE du robot dans le suivi de la trajectoire. Au cours de la simulation, les deux algorithmes ont été exécutés avec 20 particules chacun pour rechercher la meilleure position, qui représente les valeurs de K_p et K_d . Les espaces de recherche vont de 0 à 500 pour le PSO et de 0 à $\pi/2$ pour le QPSO. Après une initialisation aléatoire uniforme, les particules se déplacent à la recherche de la position optimale de chacun des essaims dans un délai de 100 itérations.

5.2.3.1 Temps de convergence du PSO et du QPSO au fil des itérations

Les deux algorithmes ont été exécutés plus de 100 fois. Des légères variations de K_p et K_{pQ} , contre des variations considérables de K_d and K_{dQ} ont été remarquées. Ces variations sont dues à la nature stochastique de deux algorithmes. En d'autres termes, le caractère aléatoire permet d'explorer un espace de recherche plus large.

Dans 80% des cas, l'algorithme QPSO a convergé plus rapidement vers les valeurs optimales, comme le montre la figure 5.8. À part la convergence rapide, les valeurs K_{pQ} et K_{dQ} étaient un peu plus stables que K_p et K_d au fil des itérations. Les paramètres obtenus à l'aide du PSO et du QPSO ont permis de réduire et de stabiliser le RMSE à $2.844 \cdot 10^{-5} m^2$.

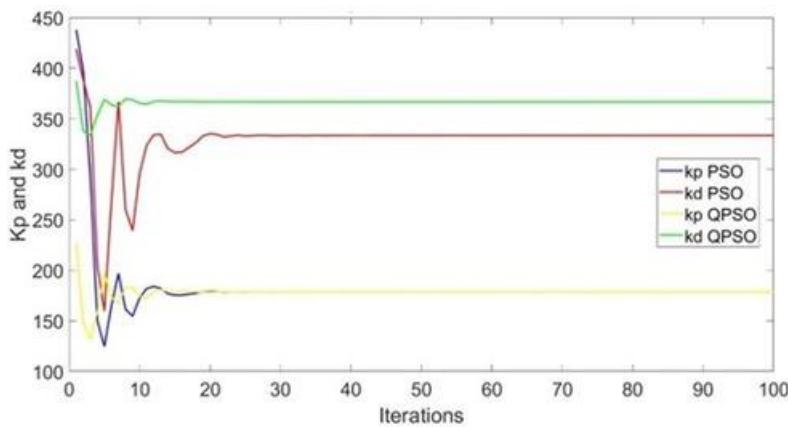


Figure 5.8 : Évolution de K_p et K_d au fil des itérations pour le suivi de la trajectoire circulaire.

La figure 5.8 présente les nombres d'itérations nécessaires aux algorithmes PSO et QPSO pour trouver les valeurs optimales de K_p et K_d . Le PSO a nécessité environ 25 itérations pour se stabiliser et converger vers les valeurs optimales, tandis que QPSO a nécessité environ 20 itérations. Malgré un temps de convergence plus long, l'algorithme PSO a démontré le même niveau de stabilité après avoir atteint les valeurs optimales.

5.2.3.2 Tests de robustesse du PSO et du QPSO face aux changements environnementaux

La figure 5.9 montre l'impact du test sur la robustesse du robot, qui a été introduit précédemment.

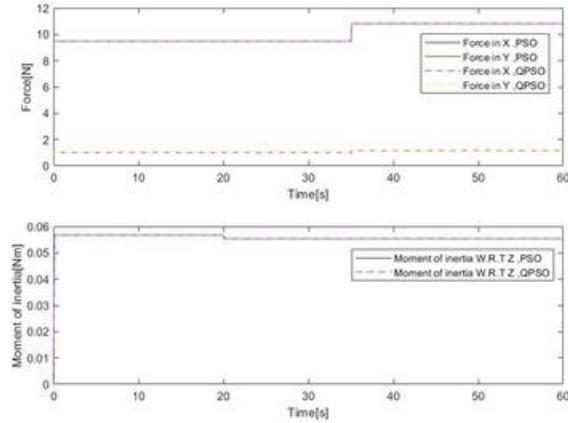


Figure 5.9: Forces et moment d'inertie appliqués au robot pour le suivi de la trajectoire circulaire.

La figure 5.9 montre que dans les deux cas du PSO et du QPSO, il y a eu une légère oscillation avant de stabiliser la force longitudinale à 9,477 N. À la 35e seconde, une compensation abrupte pour l'augmentation de 15% dans l'estimation de la masse a eu lieu. Cette compensation a conduit à une croissance soudaine de la force longitudinale à la 30.1e seconde, atteignant ainsi 10,827 N. La force longitudinale est ensuite restée constante jusqu'à la fin de la simulation. Pour la force latérale, une valeur de 1 Newton a d'abord été observée avant de passer à 1,156 N en compensation de l'augmentation de la masse. La force latérale est ensuite restée constante jusqu'à la fin de la simulation. Le moment d'inertie était à 0,056 Nm avant de diminuer à 0,055 Nm à la 20e seconde. Par la suite, Le moment d'inertie est resté constant jusqu'à la fin de la simulation.

Pour suivre la trajectoire souhaitée, le CLMR a utilisé le mouvement vers l'avant pendant que l'angle de braquage passait de 0 à 1,432 degrés au cours de la 0,1e secondes. Pour le reste, l'angle de braquage est resté constant jusqu'à la fin de la simulation.

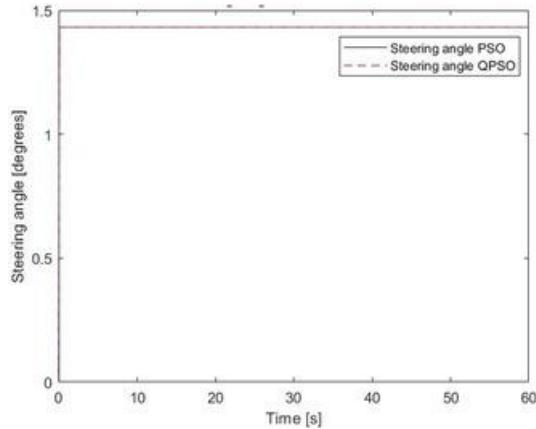


Figure 5.10 : Angle de braquage du robot au fil du temps pour le suivi de la trajectoire circulaire.

5.2.3.3 Les performances du PSO et du QPSO à réduire le RMSE

Les résultats des simulations montrent les similitudes entre PSO et QPSO, qui ont une correspondance d'ordre de $10^{-11} m^2$. Dans 55% des cas, l'algorithme QPSO s'est avéré plus performant, trouvant ainsi des paramètres plus précis. Ces paramètres ont permis de réduire et de stabiliser efficacement le RMSE de suivi de la trajectoire du robot. La figure 5.11 montre qu'au début de la simulation, les deux RMSEs sont passés de 0 à $5.63 \cdot 10^{-4} m^2$ avant de diminuer tout en oscillant jusqu'à se stabiliser à $2.175 \cdot 10^{-4} m^2$. Ensuite, les RMSEs sont restés constants jusqu'à la fin de la simulation. La différence entre les résultats du PSO et du QPSO étant minime, les deux algorithmes ont été considérés comme stabilisant le RMSE de la même manière.

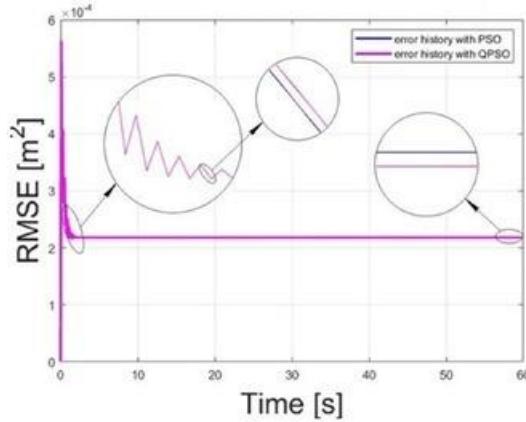


Figure 5.11 : Évolution du RMSE du robot au fil du temps pour le suivi de la trajectoire circulaire.

La figure 5.12 compare les trajectoires optimisées avec PSO et QPSO par rapport à la trajectoire souhaitée du CLMR. Les oscillations précédant la stabilisation du RMSE peuvent être remarquées dans la figure 5.11 comme l'équivalent aux zigzags du robot au début de son mouvement. Au fil du temps, lorsque les deux RMSE deviennent stables, les trajectoires optimisées avec PSO et QPSO convergent doucement vers la trajectoire souhaitée du CLMR.

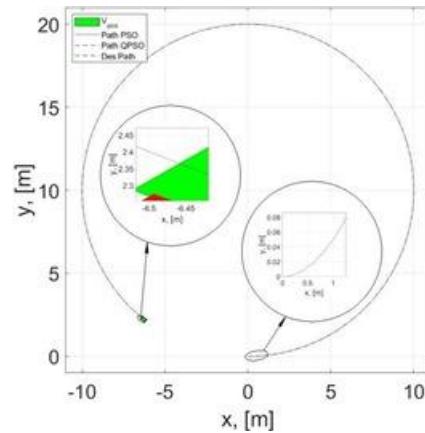


Figure 5.12 : Le suivi de la trajectoire circulaire du robot au cours du temps.

5.2.4 Résultats et discussion sur l'utilisation du régulateur PD basé sur PSO et QPSO pour le suivi d'une trajectoire sinusoïdale

Pour la trajectoire sinusoïdale, la simulation a été réalisée avec 20 particules pour chacun des deux algorithmes dans des espaces de recherche allant de 0 à 500 pour la PSO et de 0 à $\pi/2$ pour la QPSO. Après une initialisation aléatoire uniforme, les particules se déplacent à la recherche de positions optimales dans un délai de 100 itérations.

5.2.4.1 Temps de convergence du PSO et du QPSO au fil des itérations

Les deux algorithmes ont été exécutés 100 fois. Des petites variations de K_p et K_{pQ} étaient enregistrées tandis que les variations de K_d et K_{dQ} étaient plus conséquentes. Les solutions obtenues diffèrent légèrement les unes des autres en raison de la nature stochastique des algorithmes. Tel que présenté dans la figure 5.13, l'algorithme QPSO convergeait plus rapidement vers les valeurs optimales de K_{pQ} et K_{dQ} après environ 10 à 15 itérations. Le PSO, à son tour, a besoin d'approximativement 5 itérations supplémentaires pour converger vers les valeurs optimales de K_p et K_d . Les deux algorithmes ont cependant conduit à la réduction du RMSE jusqu'à $3.893 \cdot 10^{-4} m^2$.

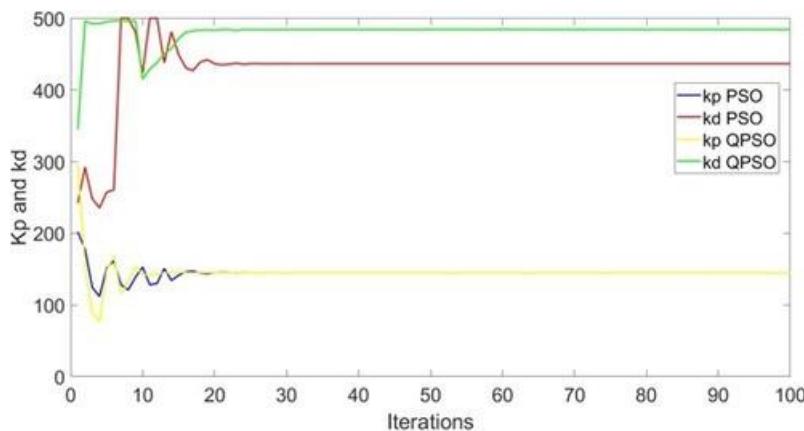


Figure 5.13 : Évolution de K_p et K_d au fil des itérations pour le suivi de la trajectoire sinusoïdale.

5.2.4.2 Tests de robustesse du PSO et du QPSO face aux changements environnementaux
Le test de robustesse du robot a eu un impact considérable sur les forces et le moment d'inertie appliqués au robot.

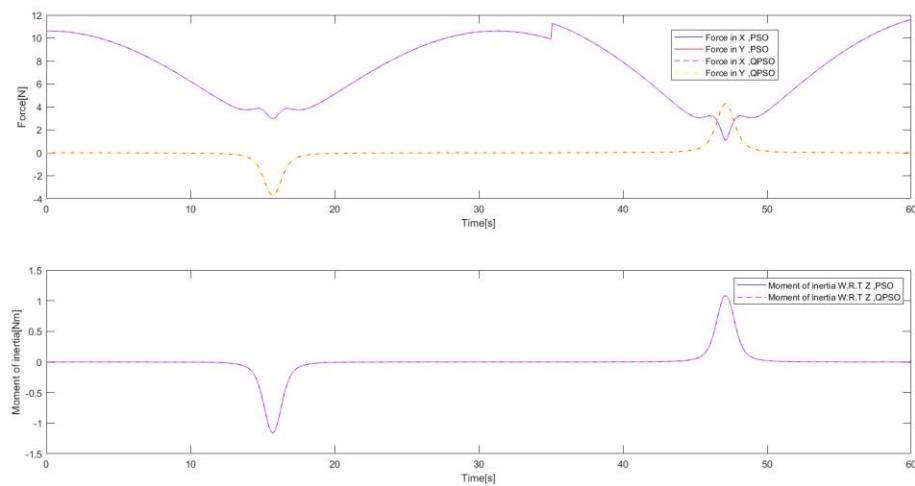


Figure 5.14 : Application des forces et du moment d'inertie au robot au fil du temps pour le suivi de la trajectoire sinusoïdale.

La figure 5.14 présente les forces longitudinale et latérale appliquées au robot. La composante principale est la force longitudinale, qui se comporte comme une transformation de la fonction sinusoïdale en commençant par son amplitude de 10,619 N, avant de diminuer jusqu'à 2,963 N, constituant un minimum local. Ensuite, à la 30.1e seconde, la deuxième amplitude de même valeur s'est produite. Le maximum global de 11,283 N se produit à la 35^e seconde pour compenser le test de robustesse. Ensuite, le minimum global de 1,072 N s'est produit à la 47.1e seconde. À la fin de la simulation, le graphique a augmenté pour atteindre le maximum global de 11,584 N.

La force latérale est présentée comme égale à zéro avec deux oscillations de -3,736 N et 4,293 N respectivement à la 15.6e et à la 47.1e seconde. La force latérale est un biais en raison de la résistance à la rotation du robot. Le graphique du moment d'inertie du robot est en grande partie égal à zéro au cours de la simulation. Comme pour le graphique de la force latérale, deux oscillations se sont produites à la 15.6e et à la 47,1e seconde. La première oscillation est allée jusqu'à -1,142 Nm, la deuxième oscillation est allée de 0 à 1,137 Nm, avant de redescendre à zéro pour le reste de la simulation. Pour les graphiques de la vitesse latérale et du moment d'inertie, le signe négatif indique la résistance au braquage de la voiture dirigée vers la gauche, et le signe positif indique la résistance au braquage de la voiture lorsqu'elle est dirigée vers la droite. La différence entre les valeurs des deux oscillations montre comment le régulateur traite le test de robustesse.

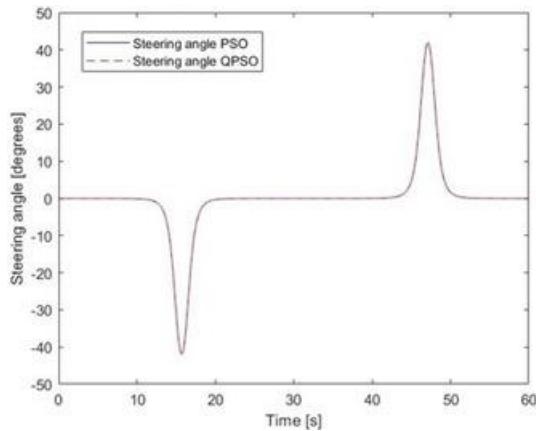


Figure 5.15 : Angle de braquage du robot au fil du temps pour le suivi de la trajectoire sinusoïdale.

Le CLMR suit la trajectoire sinusoïdale avec le mouvement vers l'avant, au même moment l'angle de braquage passe de 0 à -0.55° au cours des 12 premières secondes. Ensuite, pour correspondre à la courbe, l'angle augmente brusquement jusqu'à -41.98° au cours des 4.3 secondes suivantes, avant de revenir à -0.55° . Entre la 19e et la 43e seconde, le graphique

de l'angle de braquage passe lentement de $-0,55$ à $0,55^\circ$, avant d'épouser la courbe de manière analogue à l'opération précédente avec le signe positif.

5.2.4.3 Les performances du PSO et du QPSO à réduire le RMSE

Les algorithmes PSO et QPSO donnent des résultats identiques à $10^{-11}m^2$ près. Dans 60% des cas, l'algorithme QPSO parvient à réduire et à stabiliser de manière effective le RMSE. Dans la figure 5.16, les RMSE des deux algorithmes sont passés de $0,057$ à $4.138 \cdot 10^{-5}m^2$. Ensuite, les RMSE sont restés faibles jusqu'à la fin de la simulation, avec des valeurs finales de $3.893 \cdot 10^{-5}m^2$. Au cours de la simulation, plus précisément à la 15e et à la 47e seconde, les RMSE ont connu des oscillations, conduisant à deux maxima locaux de $0.0072 m^2$.

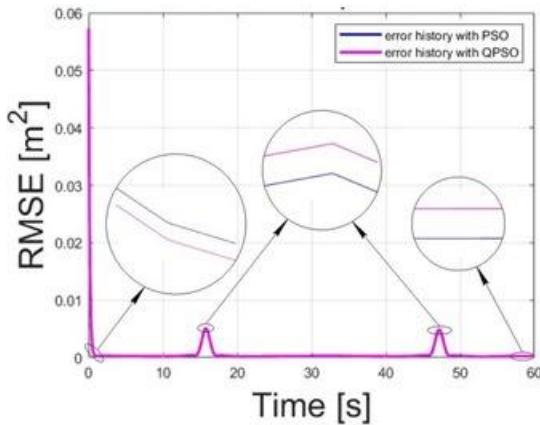


Figure 5.16 : Évolutions des RMSE de PSO et de QPSO au fil du temps pour le suivi de la trajectoire sinusoïdale.

La figure 5.17 montre le suivi des trajectoires après optimisation avec PSO et QPSO. Les deux trajectoires sont comparées entre elles et avec la trajectoire souhaitée. Au début de la simulation, le robot était placé perpendiculairement à la trajectoire souhaitée. Néanmoins, le robot a pu corriger ses trajectoires pour les faire correspondre à la trajectoire souhaitée.

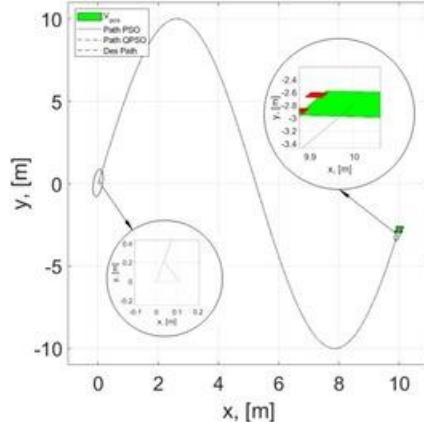


Figure 5.17: Le suivi de la trajectoire sinusoïdale à l'aide de PSO et de QPSO au fil du temps.

5.2.5 Résultats et discussion sur l'utilisation du régulateur PD basé sur PSO et QPSO pour le suivi d'une trajectoire linéaire

Ces simulations ont été réalisées avec des paramètres critiques pour les algorithmes QPSO et PSO. Cela consistait à initialiser 40 particules pour 100 itérations à la position 0 de la zone de recherche. Les vitesses des particules ont aussi été initialisées de manière aléatoire avec des valeurs comprises entre 0 et 1% de la vitesse maximale. Ces paramètres ont été choisis pour analyser facilement les comportements et la dynamique des essaims de PSO et de QPSO tout en améliorant l'exploration de l'espace de recherche. Cette approche a également été choisie pour défier ces algorithmes en réduisant la diversité des essaims, ce qui ralenti le processus de recherche.

5.2.5.1 Temps de convergence de PSO et de QPSO au fil des itérations

Après avoir exécuté ces algorithmes 100 fois, ont été observées des variations considérables de K_{pQ} et K_{dQ} , ainsi que d'énormes variations de K_p et K_d . La figure 5.18

montre que l'algorithme QPSO converge plus rapidement vers ses valeurs optimales après environ 10 à 25 itérations. L'algorithme PSO a besoin d'approximativement 30 à 50 itérations pour converger vers ses valeurs optimales. Dans 100% des cas, QPSO a trouvé les paramètres optimaux, tandis que PSO a 50% de chances de trouver les paramètres optimaux. Ces résultats mettent en évidence la différence entre ces algorithmes en ce qui concerne le temps nécessaire pour converger vers les paramètres optimaux lorsque les conditions sont critiques.

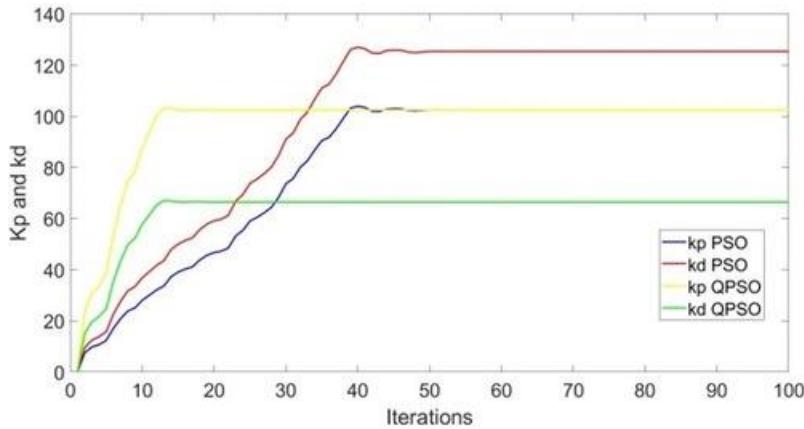


Figure 5.18 : Évolutions de K_p et K_d au fil des itérations pour le suivi de la trajectoire en ligne droite.

Pour analyser facilement les comportements et la dynamique des particules de PSO et de QPSO dans leurs essaims respectifs, une comparaison qualitative et une analyse statistique ont été effectuées pour illustrer la cohérence et la variabilité des performances entre PSO et QPSO. Les figures 5.19 a et b présentent la corrélation entre différents nombres de particules, d'itérations et de RMSE.

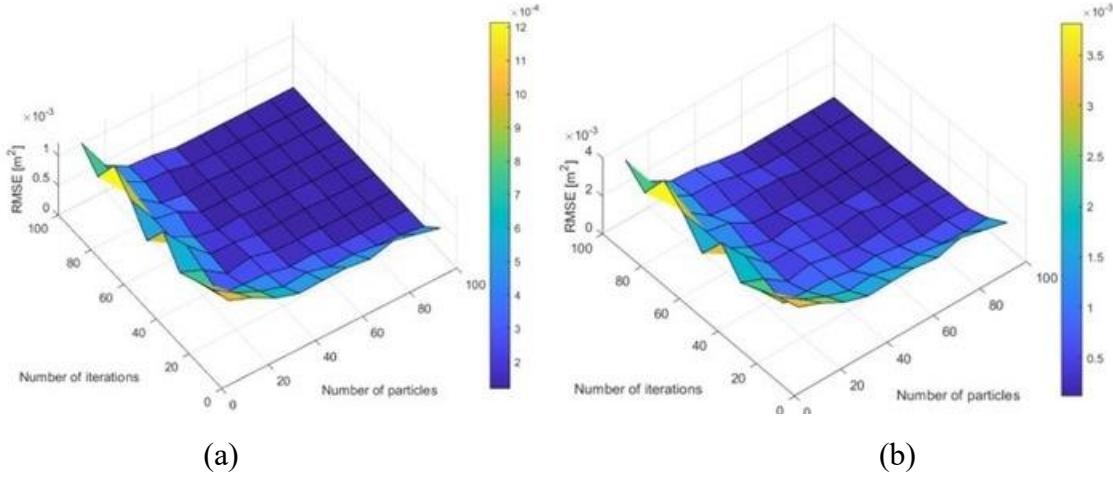


Figure 5.19 : Corrélation entre différents nombres de particules, d’itérations et de RMSE pour : (a) QPSO et (b) PSO.

Les figures 5.19 a et b montrent que les deux algorithmes se comportent de la même manière, réduisant efficacement les RMSE lorsque les nombres de particules et d'itérations sont plus élevés. Plus précisément, le nombre de particules influence davantage les essaims que le nombre d'itérations. Ces résultats montrent que QPSO donne de meilleurs résultats que PSO dans tous les cas.

5.2.5.2 Tests de robustesse du PSO et du QPSO face aux changements environnementaux

Comme pour les scénarios précédents, la figure 5.20 montre l'impact du test sur la robustesse du robot en présentant les couples appliqués au robot.

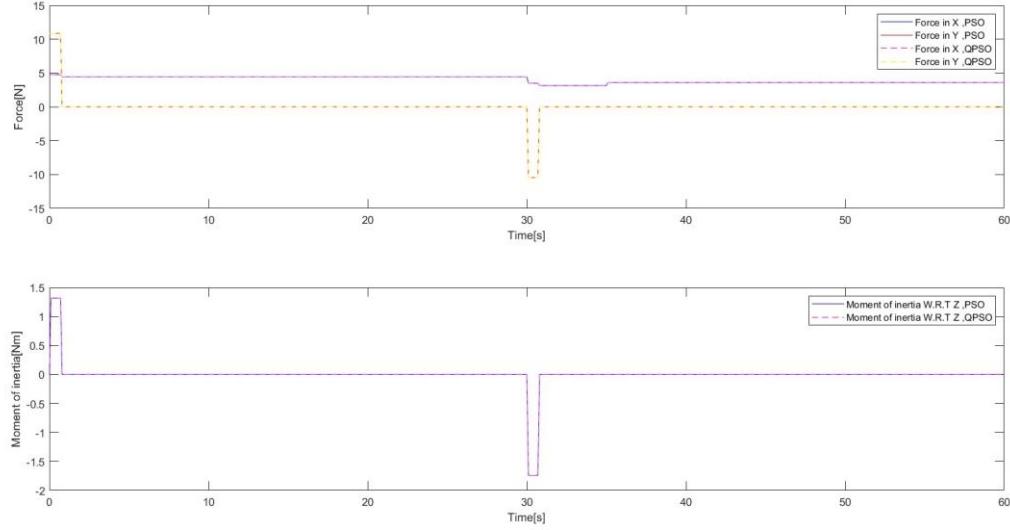


Figure 5.20 : Forces et moment d'inertie appliqués au robot au fil du temps pour le suivi de la trajectoire linéaire.

La force longitudinale est la principale composante, sa valeur commence à 4.76 N pendant 0.7 seconde en raison du mouvement vers l'avant combiné au mouvement de braquage fort et brusque du robot. Entre la 0.8ème et la 30ème secondes, la force longitudinale était égale à 4.4 N. Ensuite, pour compenser le second virage, la force a diminué à 3.5 N à la seconde suivante. Pour le test de robustesse du robot, le graphique passe de 3.14 à 3.59 N à la 35ème seconde. La force latérale et moment d'inertie présentent beaucoup de similitudes. Leurs valeurs ont été égales à zéro tout au long de la simulation à deux exceptions près, qui se sont produites les 0.7 premières secondes et entre la 30ème et la 30.7ème secondes. A ces deux moments, la force latérale était respectivement égale à 10.8 et -10.4 N, le moment d'inertie était respectivement égal à 1.32 et -1.74 Nm. Les valeurs négatives impliquent que le robot se dirige vers la droite.

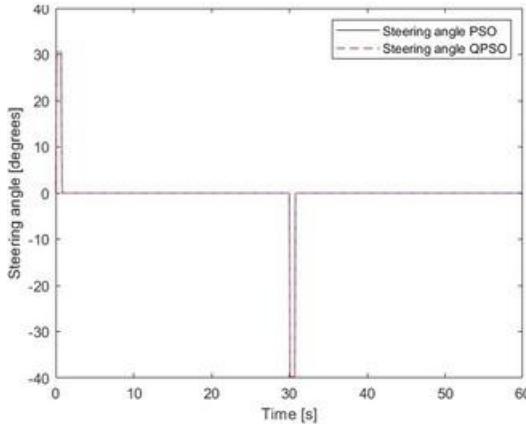


Figure 5.21 : Angle de braquage du robot au fil du temps pour la trajectoire linéaire.

Le moment d'inertie est une conséquence du mouvement vers l'avant combiné à l'angle de braquage, qui était égal à zéro pour l'ensemble de la simulation, sauf à deux moments où des croissances abruptes de 30.5 et de -39.8 ont été observées.

5.2.5.3 Les performances du PSO et du QPSO à réduire le RMSE

La figure 5.22 illustre les similitudes entre les algorithmes PSO et QPSO, ayant des résultats identiques de l'ordre de $10^{-11} m^2$. Dans 60% des cas, QPSO a plus efficacement réduit et stabilisé le RMSE comparé à l'algorithme PSO. Les graphiques des deux algorithmes ont commencé à 0 avant d'augmenter jusqu'à $0.0096 m^2$. Ensuite, Ils sont descendus tout en oscillant à $0.000138 m^2$ à la deuxième seconde. Pour le reste de la simulation, les graphiques sont restés assez bas, à l'exception de la deuxième oscillation qui a atteint $0.014 m^2$. Cette oscillation s'est produite entre la 30ème et la 32ème secondes pour compenser l'angle de braquage, permettant au robot de suivre la trajectoire souhaitée. Après cela et jusqu'à la fin de la simulation, les graphiques ont été trouvés assez bas avec des valeurs égales à $7.268 \cdot 10^{-5} m^2$.

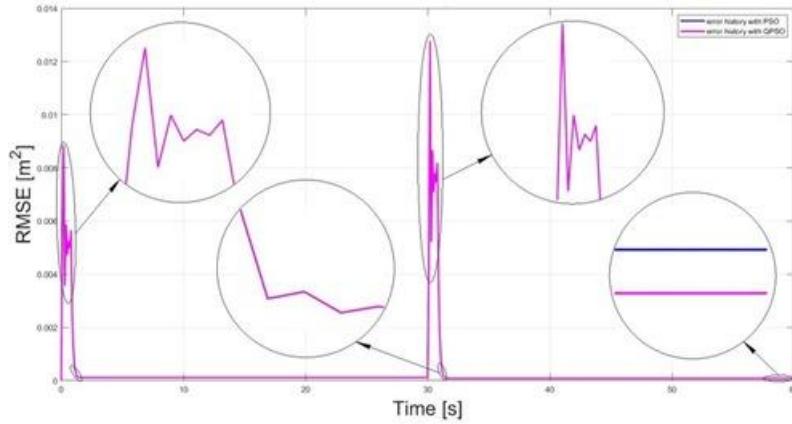


Figure 5.22 : Évolutions des RMSE du robot au fil du temps pour la trajectoire linéaire.

La figure 5.23 montre la comparaison entre la trajectoire souhaitée et les trajectoires optimisées avec les algorithmes PSO et QPSO. Au début de la simulation, le robot a été placé de manière oblique à la trajectoire souhaitée ; à la trentième seconde, la trajectoire a brusquement changé, ce qui a rendu difficile le suivi de celle-ci. Néanmoins, le robot a été en mesure de corriger ses trajectoires pour qu'elles correspondent à la trajectoire souhaitée.

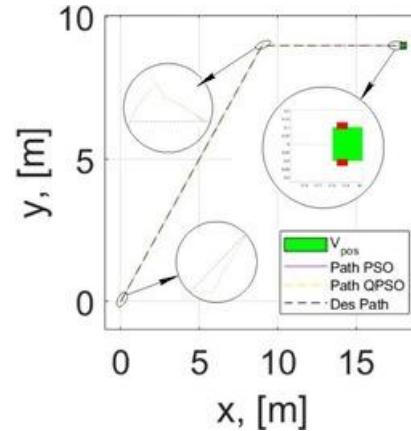


Figure 5.23: Trajectoire linéaire du robot au fil du temps.

Les résultats obtenus sont prometteurs, car ils soulignent les avantages de l'algorithme QPSO, qui pourrait avoir diverses applications en robotique mobile, telles que la commande à distance des voitures autonomes, la navigation des robots et la planification de leurs trajectoires.

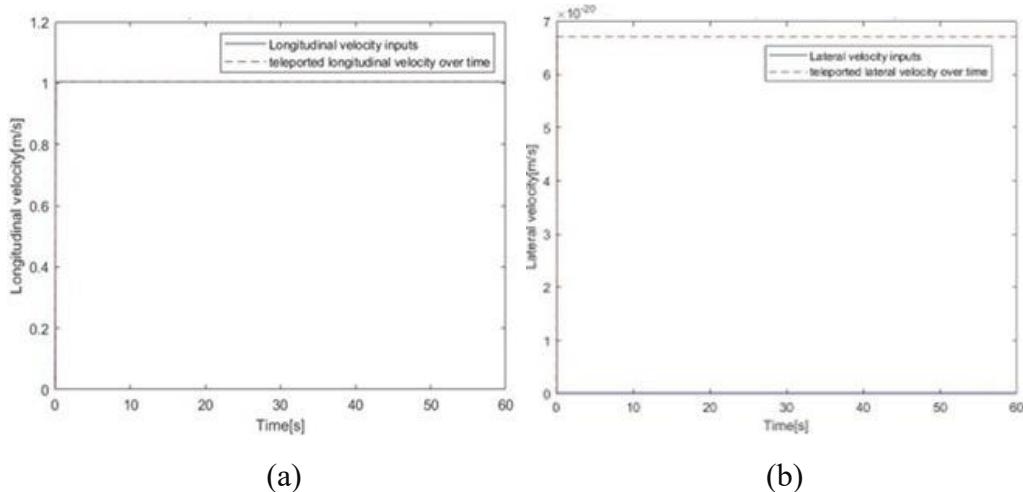
5.3 Téléportation quantique pour la commande à distance d'un robot mobile de type voiture

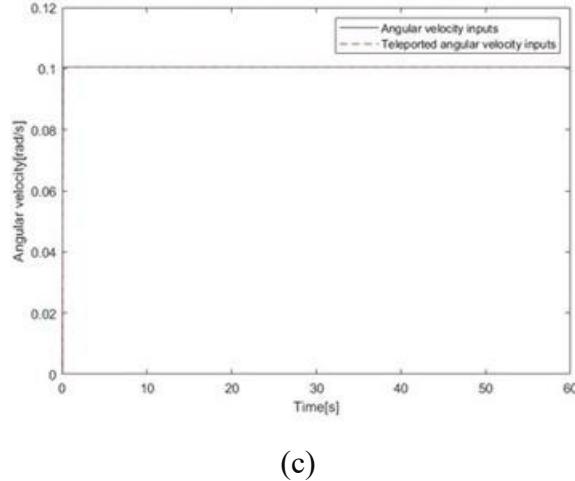
Pour téléporter textuellement l'état de chacun des qubits, le circuit quantique doit les détruire à l'entrée du circuit, puis restaurer des états quantiques identiques en reproduisant les amplitudes de probabilité des composantes d'état $|0\rangle$ et $|1\rangle$, mais aussi en reproduisant les phases relatives entre les composantes d'état $|0\rangle$ et $|1\rangle$. Pour prouver cette affirmation, les performances du circuit proposé ont été évaluées en comparant les graphiques des données avant d'être détruites et les graphiques des mêmes données après la téléportation.

5.3.1 Téléportation des données quantiques pour la trajectoire circulaire

5.3.1.1 Données sur les vitesses longitudinale, latérale et angulaire

La figure 5.24 illustre la correspondance entre les graphiques des vitesses avant et après leur téléportation.





(c)

Figure 5.24 : Comparaison entre les données réelles et les données téléportées des vitesses : (a) longitudinale, (b) latérale et (c) angulaire.

La figure 5.24 (a) compare la vitesse longitudinale avant téléportation à celle d'après téléportation. Les deux vitesses longitudinales sont passées de 0 m/s à 1.005 m/s en l'espace de 0.1 seconde pour ensuite rester constantes jusqu'à la fin de la simulation. La vitesse latérale téléportée est partie de 0 pour se stabiliser à $6.713 \cdot 10^{-20} \text{ m/s}$, telle qu'illustrée dans la figure 5.24 (b). La vitesse latérale avant téléportation est restée égale à 0 m/s tout au long de la simulation. La différence étant minime, on peut supposer une téléportation parfaite de la vitesse latérale. La figure 5.24 (c) montre la correspondance parfaite entre la vitesse angulaire avant téléportation et celle d'après téléportation. Au début de la simulation, les deux vitesses angulaires ont commencé à 0 avant d'augmenter soudainement jusqu'à 0.1005 rad/s en l'espace de 0.1 seconde. Les vitesses angulaires sont ensuite restées constantes jusqu'à la dernière seconde de la simulation.

5.3.1.2 Données sur les positions par rapport aux axes des x, des y et des z

Comme pour les vitesses, la comparaison des positions du robot avant et après la téléportation a été effectuée pour garantir la téléportation exacte des données. La figure 5.25 montre la précision et la correspondance entre les positions réelles et les positions téléportées, certifiant la haute qualité du circuit proposé.

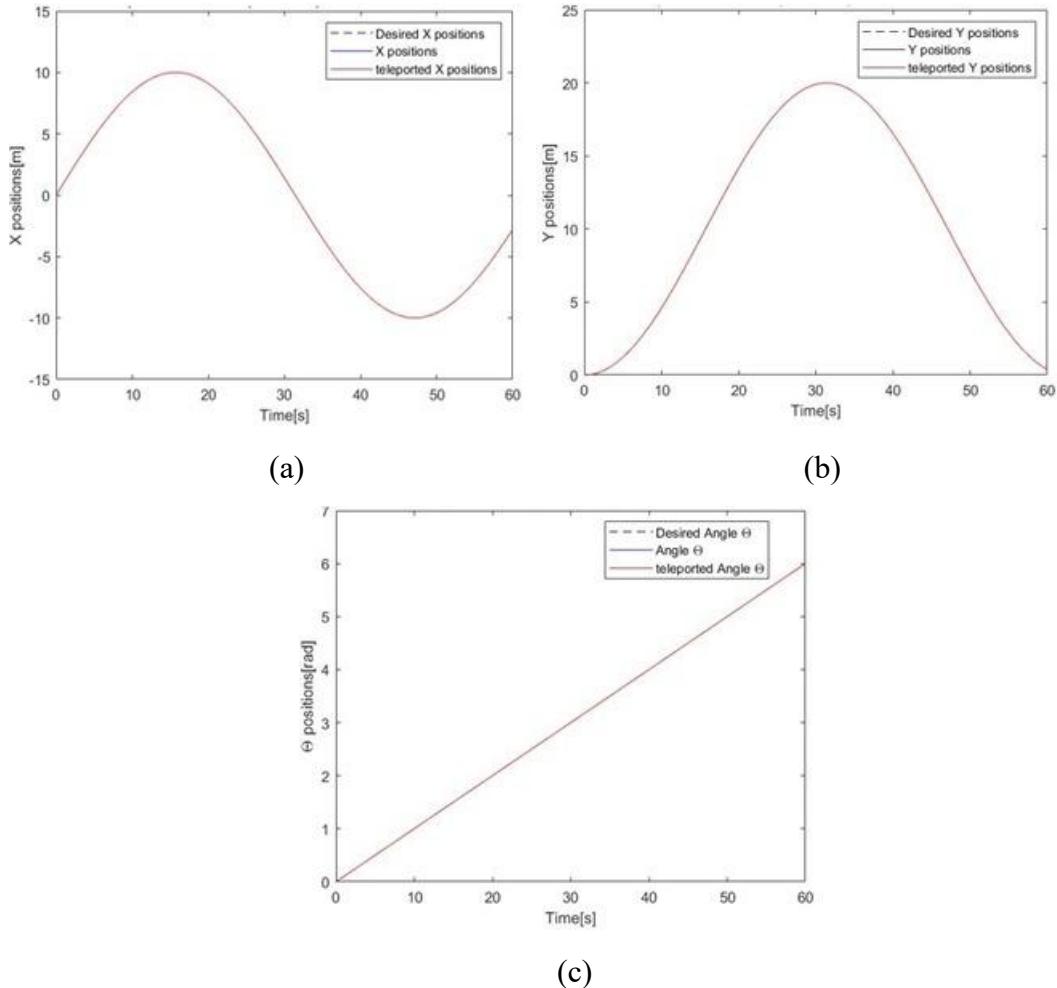


Figure 5.25 : Positions x, y et z pour la trajectoire circulaire : (a) position x, (b) position y, (c) position z.

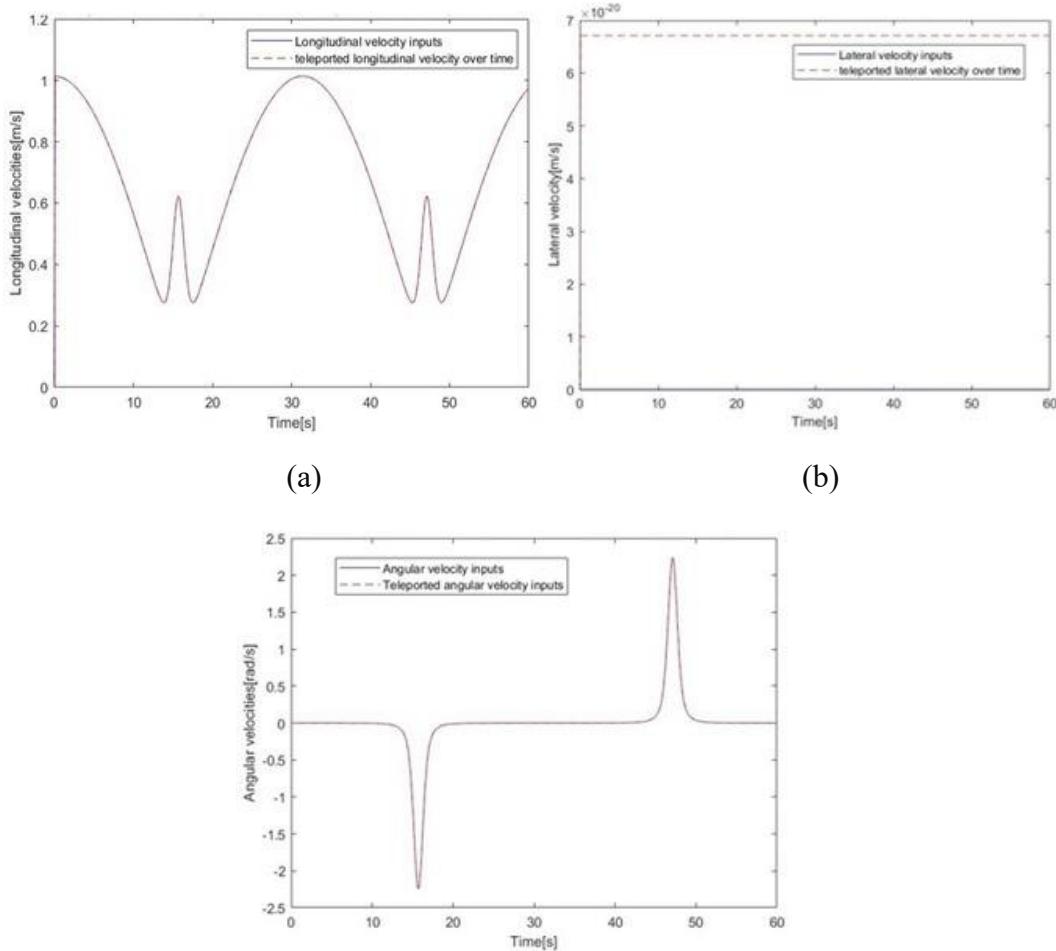
La figure 5.25 (a) représente différentes positions du robot par rapport à l'axe des abscisses. Les deux graphiques du mouvement du robot sont identiques; ils sont similaires à une onde sinusoïdale dont l'amplitude est fixée à 10 m et la période à 65 secondes. La figure 5.25 (b) illustre la correspondance parfaite des positions par rapport à l'axe des ordonnées. Les

graphiques représentent une courbe de Gauss avec un sommet à 20 m centré à la 30e seconde. La figure 5.25 (c) illustre les positions relatives à θ , qui est l'angle entre l'axe des x du robot et l'axe des x du système global. θ d'avant téléportation coïncide avec θ d'après téléportation. Les graphiques croissent de manière linéaire avec une pente de 0.1, allant ainsi de 0 à 6 radians à la 60e seconde.

5.3.2 Téléportation des données quantiques pour la trajectoire sinusoïdale

5.3.2.1 Données sur les vitesses longitudinale, latérale et angulaire

La figure 5.26 montre la correspondance entre le graphique des données des vitesses et celui des données téléportées correspondantes.



(c)

Figure 5.26: Comparaison entre les données réelles et les données téléportées des vitesses : (a) longitudinale, (b) latérale et (c) angulaire.

La figure 5.26 (a) illustre la coïncidence entre la vitesse linéaire d'entrée et la vitesse linéaire téléportée. Les deux vitesses longitudinales sont passées de 0 à 1.013 m/s , ce qui correspond au maximum global du graphique à la première et à la 30e seconde de la simulation. Les deux graphiques des vitesses longitudinales sont semblables à une ligne ondulée qui atteint ses minima locaux de 0.276 m/s à la 14e, 18e, 45e et 49e secondes et atteint ses maxima locaux de 0.622 m/s à la 15.5e et 45e secondes. La vitesse latérale téléportée est partie de 0 pour se stabiliser à $6.713 \cdot 10^{-20} \text{ m/s}$, tandis que, la vitesse latérale avant téléportation est restée égale à 0 m/s tout au long de la simulation. La différence étant insignifiante, on peut supposer une téléportation parfaite de la vitesse latérale. La figure 5.26 (c) montre la correspondance parfaite entre la vitesse angulaire avant et après téléportation. Au début de la simulation, les deux vitesses angulaires commençaient à 0 avant de diminuer soudainement jusqu'à -2.242 rad/s entre la 11.5e et la 15.5e secondes. Le signe négatif indique la résistance à la rotation de la voiture lors du braquage à gauche. Ensuite, les vitesses augmentent lentement jusqu'à 0 rad/s entre la 15.5e et la 19.5e secondes. Les deux vitesses angulaires sont restées égales à zéro jusqu'à la fin de la simulation. Mais entre la 43.5e et la 47.2e secondes, ces vitesses augmentent subitement jusqu'à 2.242 rad/s . Le signe positif indique la résistance à la rotation de la voiture lors du braquage à droite. Pour finir, les vitesses baissent progressivement jusqu'à 0 rad/s entre la 47.2e et la 50e secondes.

5.3.2.2 Données sur la position par rapport aux axes des x, des y et des z

La figure 5.27 montre la correspondance entre les positions avant et après leurs téléportations pour certifier la qualité du circuit utilisé.

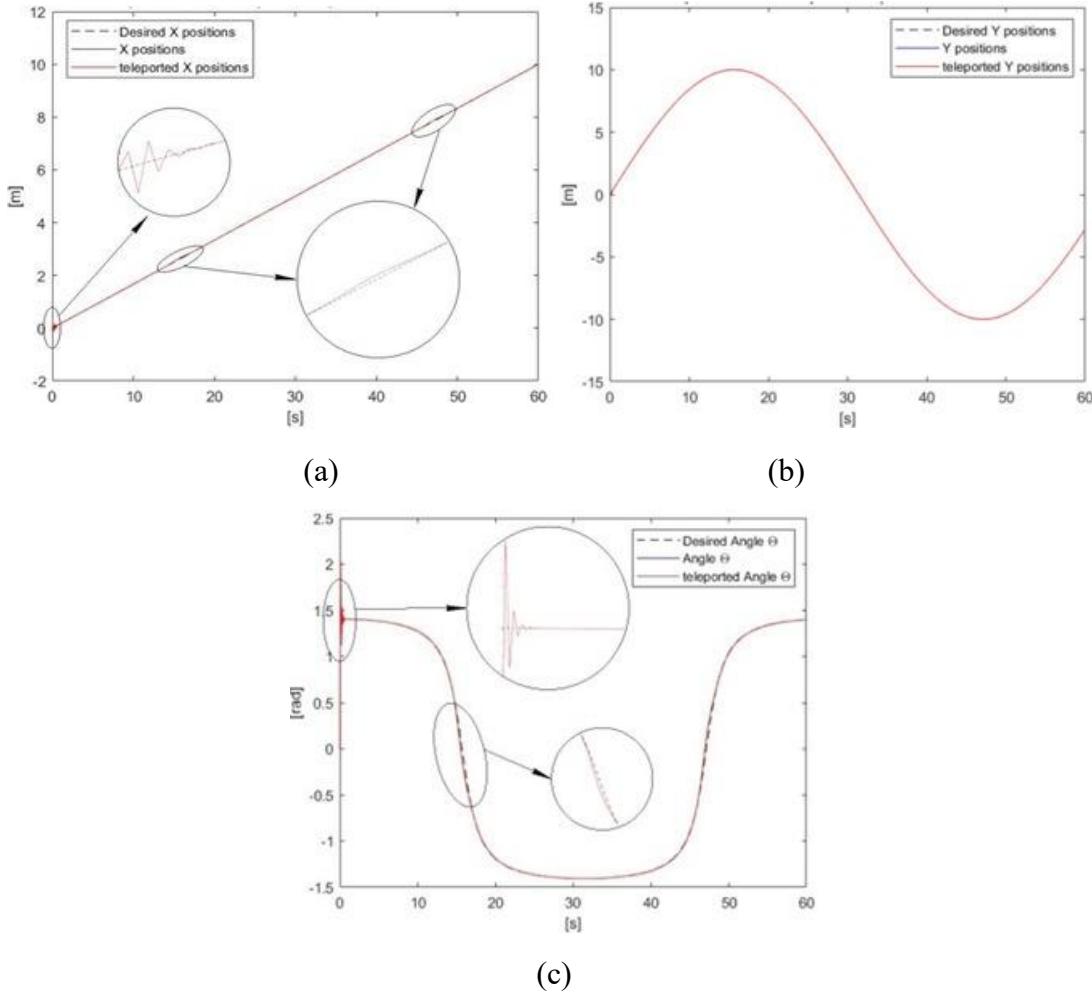


Figure 5.27 : Positions x, y et z pour le suivi de la trajectoire sinusoïdale : (a) position x, (b) position y, (c) position z.

La figure 5.27 (a) illustre les positions relatives à l'axe des x avant et après téléportation. Ainsi, les deux graphiques sont similaires, ils croissent de manière linéaire avec une pente de 0.167 allant de 0 à 10 m à la 60e seconde. La figure 5.27 (b) représente les positions du robot par rapport à l'axe des y avant et après téléportation. Les graphes sont identiques et semblables à une onde sinusoïdale dont l'amplitude est fixée à 10 m et la période à 65

secondes. La figure 5.27 (c) présente les graphiques des positions en fonction de θ . La ligne de base des deux graphiques est établie à 1.398 radians à la 0.1e et la 60e secondes, tandis que le point minimum de -1.40431 radians s'est produit à la 30e seconde. Les graphiques ressemblent donc à une fonction gaussienne inversée, capturant efficacement la coïncidence des graphes des positions de θ avant et après téléportation.

Ces résultats démontrent avec succès l'utilisation de la téléportation quantique pour l'échange des données entre deux systèmes robotisés ; cela pourrait être pertinent pour des transferts sécuritaires tout en surmontant les limites de la communication classique sans fil, comme les ondes électromagnétiques et les signaux codés transférés par WI-FI. Les signaux de la communication classique sont généralement affectés par le bruit et les interférences dans des environnements difficiles, ils peuvent aussi être atténués sur des longues distances séparant le robot mobile de sa station de commande. La téléportation quantique est donc une meilleure alternative à la communication classique pour un transfert imperturbable de données dans des domaines tels que : Exploration spatiale Permettant des communications instantanées et sécurisées entre la Terre et les sondes, exploration sous-marine, régions polaires où les réseaux classiques ne sont pas fiables, la téléportation quantique pourrait offrir un moyen de transmettre des données sans être affecté par les conditions environnantes.

Chapitre 6 – Conclusion

L'optimisation et la régulation des paramètres du régulateur PD ont constitué des défis importants dans le suivi des trajectoires du CLMR. Afin de pallier ces challenges, une optimisation avancée pour l'amélioration du suivi des trajectoires est proposée en trois étapes, à savoir, la construction du CANN et du QANN pour identifier les paramètres du CLMR, le développement des algorithmes PSO et QPSO pour adapter les trajectoires tenant compte des paramètres du régulateur PD, et pour finir, le développement d'un protocole de téléportation quantique pour assurer la communication entre le CLMR et sa station de commande.

Pour la première étape, les algorithmes CANN et QANN ont été programmés pour prédire la masse et le moment d'inertie d'un CLMR. Les résultats observés montrent que dans 100% des cas, QANN performe mieux, prédisant des valeurs qui se rapprochent le plus des valeurs réelles. CANN performe moins bien comparé à QANN, car il fait du surajustement. Les résultats du QANN sont prometteurs pour diverses tâches de régression et de classification pour lesquelles les modèles classiques font du surajustement des paramètres. Les résultats de la deuxième étape montrent que, dans des conditions normales, les algorithmes PSO et QPSO donnaient des résultats similaires en ce qui concerne leurs RMSE respectifs. Cependant, QPSO a obtenu une meilleure optimisation de K_p et K_d , et cela avec un meilleur temps de convergence et une meilleure stabilité lorsque les particules ont été initialisées à zéro avec des vitesses initiales inférieures à 1 % de la vitesse maximale. Ces résultats proposent l'algorithme QPSO pour diverses applications de commande à distance, telles que les voitures autonomes, la navigation robotique et la planification de la trajectoire des drones. Pour la dernière étape,

les données originales et les données téléportées des vitesses et des positions du robot ont été comparées pour garantir la précision du protocole proposé. Les résultats comparés étaient similaires, ce qui prouve que le protocole de téléportation proposé fonctionne parfaitement bien, même lorsqu'il est soumis à des tests de robustesse.

Les versions quantiques des trois algorithmes développés dans ce travail ont généralement montré de meilleures performances, tant en termes de temps de calcul que de qualité des solutions obtenues. Toutefois, ces résultats sont basés sur des simulations, et un ordinateur quantique réel offrirait des avantages supplémentaires. Néanmoins, la construction et l'entretien de ces machines restent coûteux en raison de la nécessité de maintenir des conditions extrêmes, comme des températures proches du zéro absolu, ainsi que des défis liés à la stabilité et à la fiabilité des qubits. C'est pourquoi, pour nos travaux futurs, nous prévoyons mettre en œuvre ces algorithmes dans un environnement réel afin d'évaluer leurs performances sur du matériel quantique physique. Cela pourrait conduire à une accélération exponentielle de la vitesse de traitement et à une amélioration de la robustesse, augmentant ainsi l'efficacité du robot pour le suivi des trajectoires.

References

- [1] B. Hu, Z. Cao, and M. Zhou, "An Efficient RRT-Based Framework for Planning Short and Smooth Wheeled Robot Motion Under Kinodynamic Constraints," *IEEE Transactions on Industrial Electronics*, vol. 68, no. 4, pp. 3292-3302, 2021, doi: 10.1109/TIE.2020.2978701.
- [2] V. S. Medeiros, E. Jelavic, M. Bjelonic, R. Siegwart, M. A. Meggiolaro, and M. Hutter, "Trajectory Optimization for Wheeled-Legged Quadrupedal Robots Driving in Challenging Terrain," *IEEE Robotics and Automation Letters*, vol. 5, no. 3, pp. 4172-4179, 2020, doi: 10.1109/LRA.2020.2990720.
- [3] F. M. Campos, "Ardosia: Simulating Circuits and Robotic Systems in a Single Learning Platform," *IEEE Transactions on Learning Technologies*, vol. 16, no. 2, pp. 166-177, 2023, doi: 10.1109/TLT.2023.3243688.
- [4] A. Carfi and F. Mastrogiovanni, "Gesture-Based Human–Machine Interaction: Taxonomy, Problem Definition, and Analysis," *IEEE Transactions on Cybernetics*, vol. 53, no. 1, pp. 497-513, 2023, doi: 10.1109/TCYB.2021.3129119.
- [5] L. Yan, B. Ma, Y. Jia, and Y. Jia, "Observer-Based Trajectory Tracking Control of Nonholonomic Wheeled Mobile Robots," *IEEE Transactions on Control Systems Technology*, vol. 32, no. 3, pp. 1114-1121, 2024, doi: 10.1109/TCST.2024.3351073.
- [6] P. Corke, W. Jachimczyk, and R. Pillat, "Mobile Robot Vehicles," in *Robotics, Vision and Control: Fundamental Algorithms in MATLAB®*, P. Corke, W.

- Jachimczyk, and R. Pillat Eds. Cham: Springer International Publishing, 2023, pp. 127-160.
- [7] L. Wang, X. Niu, T. Zhang, H. Tang, and Q. Chen, "Accuracy and robustness of ODO/NHC measurement models for wheeled robot positioning," *Measurement*, vol. 201, p. 111720, 2022/09/30/ 2022, doi: <https://doi.org/10.1016/j.measurement.2022.111720>.
- [8] H. X. Nguyen, H. Q. Cao, T. T. Nguyen, T. N.-C. Tran, H. N. Tran, and J. W. Jeon, "Improving robot precision positioning using a neural network based on Levenberg Marquardt–APSO algorithm," *IEEE Access*, vol. 9, pp. 75415-75425, 2021.
- [9] Y. Zhang, F. Gao, and F. Zhao, "Research on path planning and tracking control of autonomous vehicles based on improved RRT* and PSO-LQR," *Processes*, vol. 11, no. 6, p. 1841, 2023.
- [10] I. Calvo, E. Villar, C. Napole, A. Fernández, O. Barambones, and J. M. Gil-García, "Reliable control applications with wireless communication technologies: Application to robotic systems," *Sensors*, vol. 21, no. 21, p. 7107, 2021.
- [11] Y. Wang, B. Shen, Z. Nan, and W. Tao, "An End-to-End Path Planner Combining Potential Field Method With Deep Reinforcement Learning," *IEEE Sensors Journal*, vol. 24, no. 16, pp. 26584-26591, 2024, doi: 10.1109/JSEN.2024.3425530.
- [12] S. Mutti, N. Pedrocchi, A. Valente, and G. Dimauro, "Sim-to-Real RNN-Based Framework for the Precise Positioning of Autonomous Mobile Robots," *IEEE Access*, vol. 12, pp. 163948-163957, 2024, doi: 10.1109/ACCESS.2024.3488175.

- [13] H. Hohenfeld, D. Heimann, F. Wiebe, and F. Kirchner, "Quantum Deep Reinforcement Learning for Robot Navigation Tasks," *IEEE Access*, vol. 12, pp. 87217-87236, 2024, doi: 10.1109/ACCESS.2024.3417808.
- [14] N. F. Bar, H. Yetis, and M. Karakose, "An Approach Based on Quantum Reinforcement Learning for Navigation Problems," in *2022 International Conference on Data Analytics for Business and Industry (ICDABI)*, 25-26 Oct. 2022 2022, pp. 593-597, doi: 10.1109/ICDABI56818.2022.10041570.
- [15] A. Alomari and S. A. P. Kumar, "ReLAQA: Reinforcement Learning-Based Autonomous Quantum Agent for Quantum Applications," *IEEE Transactions on Artificial Intelligence*, pp. 1-11, 2024, doi: 10.1109/TAI.2024.3437335.
- [16] W. J. Yun, J. P. Kim, S. Jung, J. H. Kim, and J. Kim, "Quantum Multiagent Actor-Critic Neural Networks for Internet-Connected Multirobot Coordination in Smart Factory Management," *IEEE Internet of Things Journal*, vol. 10, no. 11, pp. 9942-9952, 2023, doi: 10.1109/JIOT.2023.3234911.
- [17] L. Hu, C. Wei, and L. Yin, "Fuzzy A* quantum multi-stage Q-learning artificial potential field for path planning of mobile robots," Elsevier, <https://doi.org/10.1016/j.engappai.2024.109866>
- [18] S. Spanogianopoulos (2018) Particle Swarm Optimization and Applications in Robotics: A Survey, 9th International Conference on Information, Intelligence, Systems and Applications (IISA), pp. 1-7, DOI: 10.1109/IISA.2018.8633635.

- [19] I. Thammachantuek, M. Ketcham, S. Mirjalili (2022) "Path planning for autonomous mobile robots using multi-objective evolutionary particle swarm optimization, PLOS ONE, Vol. 17, No. 8, DOI: 10.1371/journal.pone.0271924.
- [20] M. N. A. Wahab, C. M. Lee, M. F. Akbar, F. H. Hassan (2020) Path Planning for Mobile Robot Navigation in Unknown Indoor Environments Using Hybrid PSOFS Algorithm, IEEE Access, Vol. 8, DOI: 10.1109/ACCESS.2020.3021605.
- [21] Z. Li, Y. Wenjie, L. Guangxu, Q. Guangxu, L. Yunchuan (2023) Particle Swarm Algorithm Path-Planning Method for Mobile Robots Based on Artificial Potential Fields, Sensors, Vol. 23, No. 13, pp. 6082, DOI: 10.3390/s23136082.
- [22] N. Baygin, M. Baygin, M. Karakose (2018) PSO Based Path Planning Approach for Multi Service Robots in Dynamic Environments, International Conference on Artificial Intelligence and Data Processing (IDAP), pp. 1-5, DOI: 10.1109/IDAP.2018.8620801.
- [23] S. Alaliyat, R. Oucheikh, I. Hameed, (2019) Path Planning in Dynamic Environment Using Particle Swarm Optimization Algorithm, 8th International Conference on Modeling Simulation and Applied Optimization (ICMSAO), pp. 1-5, DOI: 10.1109/ICMSAO.2019.8880434.
- [24] T. T. Harmania, M. K. D. Hardhienata, K. Priandana (2021) Development of Multi-Robot Systems Using Particle Swarm Optimization Algorithm for Task Allocation, IEEE Region 10 Symposium (TENSYMP), pp. 1-8, DOI: 10.1109/TENSYMP52854.2021.9551012.
- [25] P. K. Das, B. M. Sahoo, H. S. Behera, S. Vashisht (2016) An improved particle swarm optimization for multi-robot path planning, International Conference on

- Innovation and Challenges in Cyber Security (ICICCS-INBUSH), pp. 97-106, DOI: 10.1109/ICICCS.2016.7542324.
- [26] J. Zhang, Y. Lu, L. Che, M. Zhou (2022) Moving-Distance-Minimized PSO for Mobile Robot Swarm, IEEE Transactions on Cybernetics, Vol. 52, No. 9, DOI: 10.1109/TCYB.2021.3079346.
- [27] C. Petschnigg, M. Brandstötter, H. Pichler, M. Hofbaur, B. Dieber (2019) Quantum Computation in Robotic Science and Applications, International Conference on Robotics and Automation (ICRA), pp. 803-810, DOI: 10.1109/ICRA.2019.8793768.
- [28] P. B. Fernandes, R. C. L. Oliveira, J. V. Fonseca Neto (2022) Trajectory planning of autonomous mobile robots applying a particle swarm optimization algorithm with peaks of diversity, Applied Soft Computing Journal, Vol. 116, DOI: 10.1016/j.asoc.2021.108108.
- [29] T. Xue, R. Li, M. Tokgo, J. Ri, G. Han (2017) Trajectory planning for autonomous mobile robot using a hybrid improved QPSO algorithm, Soft Computing: A Fusion of Foundations, Methodologies and Applications, Vol. 21, No. 9, pp. 2421-2437, DOI: 10.1007/s00500-015-1956-2.
- [30] L. Lan, G. Tongbin, C. Kangkang, Z. Qi (2023) Trajectory Planning in Robot Joint Space Based on Improved Quantum Particle Swarm Optimization Algorithm, Applied Sciences, Vol. 13, No. 12, pp. 7031, DOI: 10.3390/app13127031.
- [31] P. B. Fernandes, R. C. L. D. Oliveira, J. V. F. Neto (2018) A Modified QPSO for Robotic Vehicle Path Planning, IEEE Congress on Evolutionary Computation (CEC), pp. 1-7, DOI: 10.1109/CEC.2018.8477681.

- [32] M. Tokgo, R. Li, C. Kim, L. Hu, M. Kim (2014) A Method for Trajectory Planning of Mobile Robot in Random Obstacles Environment, Sixth International Conference on Intelligent Human-Machine Systems and Cybernetics, Vol. 2, pp. 51-54, DOI: 10.1109/IHMSC.2014.115.
- [33] S. Dian, J. Zhong, B. Guo, J. Liu, R. Guo (2022) A smooth path planning method for mobile robot using a BES-incorporated modified QPSO algorithm, Expert Systems With Applications, vol. 208, pp. 118256, DOI: 10.1016/j.eswa.2022.118256.
- [34] L. Dong, X. Yuan, C. Zhang, Y. Song, Q. Xu, F. Zhou (2020) Ameliorated Particle Swarm Optimization Algorithm and Its Application in Robot Path Planning, Chinese Automation Congress (CAC), pp. 5544-5549, DOI: 10.1109/CAC51589.2020.9327685.
- [35] O. A. Montesinos López, A. Montesinos López, and J. Crossa, "Fundamentals of Artificial Neural Networks and Deep Learning," in *Multivariate Statistical Machine Learning Methods for Genomic Prediction*, O. A. Montesinos López, A. Montesinos López, and J. Crossa Eds. Cham: Springer International Publishing, 2022, pp. 379-425.
- [36] Kelly, A. *Mobile Robotics: Mathematics, Models, and Methods*; Cambridge University Press: New York, NY, USA, 2013.
- [37] Khare, D.; Prashant, K.V.; Mohan, S. Design and control of mobile robots with two and four independent rotatable power wheels. In *Advances in Industrial Machines and Mechanisms, Proceedings of the Conference on Industrial Problems on*

- Machines and Mechanisms (IPRoMM 2020)*, 2020; Springer: Singapore, 2021.
https://doi.org/10.1007/978-981-16-1769-0_14.
- [38] Kumar, S.; Mohan, S.; Skitova, V. Designing and implementing a versatile agricultural robot: A vehicle manipulator system for efficient multitasking in farming operations. *Machines* **2023**, *11*, 776.
<https://doi.org/10.3390/machines11080776>.
- [39] Dudek, G.; Jenkin, M. *Computational Principles of Mobile Robotics*, 2nd ed.; Cambridge University Press: Cambridge, UK, 2010.
- [40] Harik, E.H.C.; Korsaeth, A. The heading weight function: A novel LiDAR-based local planner for nonholonomic mobile robots, *Sensors* **2019**, *19*, 3606.
<https://doi.org/10.3390/s19163606>.
- [41] M. Anderson *et al.*, "Quantum teleportation using highly coherent emission from telecom C-band quantum dots," *npj Quantum Information*, vol. 6, no. 1, p. 14, 2020/01/31 2020, doi: 10.1038/s41534-020-0249-5.
- [42] M. El Kirdi, A. Slaoui, N. Ikken, M. Daoud, and R. A. Laamara, "led quantum teleportation between discrete and continuous physical systems," *Physica Scripta*, vol. 98, no. 2, p. 025101, 2023.
- [43] S. Salimian, M. K. Tavassoly, and N. Sehati, "Quantum Teleportation of the Entangled Superconducting Qubits via LC Resonators," *International Journal of Theoretical Physics*, vol. 62, no. 4, p. 85, 2023/04/17 2023, doi: 10.1007/s10773-023-05304-0.
- [44] P. Zhang, X. Lu, S. Kuang, and D. Dong, "Optimal tripartite quantum teleportation protocols via noisy channels by feed-forward control and environment-assisted

- measurement," *Results in Physics*, vol. 60, p. 107632, 2024/05/01/ 2024, doi: <https://doi.org/10.1016/j.rinp.2024.107632>.
- [45] E. Villaseñor and R. Malaney, "Enhancing Continuous Variable Quantum Teleportation using Non-Gaussian Resources," in *2021 IEEE Global Communications Conference (GLOBECOM)*, 7-11 Dec. 2021 2021, pp. 1-6, doi: 10.1109/GLOBECOM46510.2021.9685767.
- [46] M. Bolokian, A. A. Orouji, and M. Houshmand, "A bidirectional quantum remote state preparation scheme and its performance analysis in noisy environments," *Optical and Quantum Electronics*, vol. 55, no. 9, p. 835, 2023/07/07 2023, doi: 10.1007/s11082-023-05110-2.
- [47] F. Khoshnoud, I. I. Esat, S. Javaherian, and B. Bahr, "Quantum entanglement and cryptography for automation and control of dynamic systems," *arXiv preprint arXiv:2007.08567*, 2020.
- [48] B. Panda, N. Tripathy, S. Sahu, B. Behera, and W. Elhady, "Controlling Remote Robots Based on Zidan's Quantum Computing Model," *Computers, Materials & Continua*, vol. 73, pp. 6225-6236, 01/01 2022, doi: 10.32604/cmc.2022.028394.
- [49] F. Khoshnoud, L. Lamata, C. de Silva, and M. Quadrelli, *Quantum Teleportation for Control of Dynamic Systems and Autonomy*. 2020.
- [50] "Discussion on the Application of Quantum Teleportation Technology in the Internet of Vehicles," *Academic Journal of Computing & Information Science*, vol. 6, no. 7, 2023, doi: 10.25236/AJCIS.2023.060720.

- [51] F. Chiti, R. Picchi, and L. Pierucci, "Metropolitan Quantum-Drone Networking and Computing: A Software-Defined Perspective," *IEEE Access*, vol. 10, pp. 126062-126073, 2022, doi: 10.1109/ACCESS.2022.3226127.
- [52] M. Mastriani, S. S. Iyengar, and L. Kumar, "Satellite quantum communication protocol regardless of the weather," *Optical and Quantum Electronics*, vol. 53, no. 4, 2021, doi: 10.1007/s11082-021-02829-8.
- [53] B.-C. Ciobanu, V. Iancu, and P. G. Popescu, "EntangleNetSat: A Satellite-Based Entanglement Resupply Network," *IEEE Access*, vol. 10, 2022, doi: 10.1109/ACCESS.2022.3185403.
- [54] T. Wang, P. Li, Y. Wu, L. Qian, Z. Su, and R. Lu, "Quantum-Empowered Federated Learning in Space-Air-Ground Integrated Networks," *IEEE Network*, vol. 38, no. 1, 2024, doi: 10.1109/MNET.2023.3318083.
- [55] T. Gonzalez-Raya, S. Pirandola, and M. Sanz, "Satellite-based entanglement distribution and quantum teleportation with continuous variables," *Communications Physics*, vol. 7, no. 1, 2024, doi: 10.1038/s42005-024-01612-x.
- [56] M. Mastriani, S. S. Iyengar, and K. J. Latesh Kumar, "Bidirectional teleportation for underwater quantum communications," *Quantum Information Processing*, vol. 20, no. 1, 2021, doi: 10.1007/s11128-020-02970-5.
- [57] S. Filatov and M. Auzinsh, "Towards Two Bloch Sphere Representation of Pure Two-Qubit States and Unitaries," *Entropy*, vol. 26, no. 4, p. 280, 2024. [Online]. Available: <https://www.mdpi.com/1099-4300/26/4/280>.

- [58] K. Khalfaoui, T. Boudjedaa, and E. H. Kerkouche, "Automatic design of quantum circuits," *Quantum Information Processing*, vol. 20, no. 9, p. 283, 2021/09/01 2021, doi: 10.1007/s11128-021-03208-8.
- [59] M. Sisodia, "A Theoretical Study of Controlled Quantum Teleportation Scheme for n-qubit Quantum State," *International Journal of Theoretical Physics*, vol. 61, no. 12, p. 270, 2022/12/08 2022, doi: 10.1007/s10773-022-05260-1.

Publications

Les travaux de recherche menés dans ce mémoire ont été valorisés par la publication de deux articles dans des revues scientifiques réputées, bénéficiant d'un processus rigoureux de révision par les pairs.

Liste des articles publiés

- [1] J. Numbi, N. Zioui, and M. Tadjine, "Quantum Particle Swarm Optimisation Proportional-Derivative Control for Trajectory Tracking of a Car-like Mobile Robot," *Electronics*, vol. 14, no. 5, p. 832, 2025. [Online]. Available: <https://www.mdpi.com/2079-9292/14/5/832>.
- [2] J. Numbi, N. Zioui, and M. Tadjine, "The Concept of Quantum Teleportation for Remote Control of a Car-like Mobile Robot," *Robotics*, vol. 14, no. 3, p. 25, 2025. [Online]. Available: <https://www.mdpi.com/2218-6581/14/3/25>.