

Titre: Design et développement d'un quadrirotor joueur de tennis de table
Title: avec des hélices inclinables

Auteur: Radu Ionut Popescu
Author:

Date: 2020

Type: Mémoire ou thèse / Dissertation or Thesis

Référence: Popescu, R. I. (2020). Design et développement d'un quadrirotor joueur de tennis de table avec des hélices inclinables [Mémoire de maîtrise, Polytechnique Montréal]. PolyPublie. <https://publications.polymtl.ca/5475/>
Citation:

 **Document en libre accès dans PolyPublie**
Open Access document in PolyPublie

URL de PolyPublie: <https://publications.polymtl.ca/5475/>
PolyPublie URL:

Directeurs de recherche: Maxime Raison, & Sofiane Achiche
Advisors:

Programme: Génie mécanique
Program:

POLYTECHNIQUE MONTRÉAL

affiliée à l'Université de Montréal

**Design et développement d'un quadrirotor joueur de tennis de table avec des
hélices inclinables**

RADU IONUT POPESCU

Département de génie mécanique

Mémoire présenté en vue de l'obtention du diplôme de *Maîtrise ès sciences appliquées*

Génie mécanique

Septembre 2020

POLYTECHNIQUE MONTRÉAL

affiliée à l'Université de Montréal

Ce mémoire intitulé :

Design et développement d'un quadrirotor joueur de tennis de table avec des hélices inclinables

présenté par **Radu Ionut POPESCU**

en vue de l'obtention du diplôme de *Maîtrise ès sciences appliquées*

a été dûment accepté par le jury d'examen constitué de :

Marek BALAZINSKI, président

Maxime RAISON, membre et directeur de recherche

Sofiane ACHICHE, membre et codirecteur de recherche

Ali SAFAEI, membre

RÉSUMÉ

Les bras manipulateurs joueurs de tennis de table les plus avancés sont chers et leur configuration requière beaucoup d'espace. On pourrait considérer l'utilisation de robots aériens pour exécuter cette tâche, mais la plupart des avions à décollage et atterrissage verticales (ADAV) ne sont pas suffisamment rapides pour reproduire des mouvements de frappes. L'objectif de la recherche présentée dans ce mémoire est de développer un nouveau type de robot aérien joueur de tennis de table. Un prototype de quadrirotor ayant des hélices inclinables est d'abord considéré pour permettre de suivre des trajectoires agressives. Le robot a besoin d'atteindre des vitesses de 3.5 m/s au point d'impact et de rester suffisamment léger pour être agile. Ensuite, pour obtenir de hautes performances sur ce requis, un contrôleur Itérative Linéaire Quadratique Régulateur (iLQR) qui suit des trajectoires ayant un snap minimum est implémenté. Le contrôle de la boucle interne est délégué à un microcontrôleur PX4 pour le tangage, le lacet et le roulis pour assurer une bonne robustesse et une haute fréquence. Cette approche est testée dans une simulation réaliste et ensuite le framework complet pour cette application est développé sur un ordinateur embarqué. Des résultats expérimentaux ont été obtenus avec des caméras de capture de mouvements donnant la position et le temps d'impact. Cette information est envoyée au quadrirotor par communication sans-fil et la trajectoire est exécutée immédiatement. Au meilleur de nos connaissances, il s'agit du premier robot aérien étant capable de retourner des balles de tennis de table lancées par un humain. Un taux de succès de 40% est obtenu sur les frappes avec le modèle réel du quadrirotor, significativement supérieur à ce qui était possible d'atteindre auparavant avec un quadrirotor.

ABSTRACT

State-of-art table tennis robot manipulators are expensive and their setup require a lot of space. One could consider using aerial robots for this task, but most vertical takeoff and landing (VTOL) vehicles are not fast enough to reproduce hitting motions. The objective of the research presented in this thesis is to develop a novel type of aerial robot tennis table player. A prototype of a quadrotor that uses tilting propellers is first considered to enable the possibility of aggressive trajectory tracking. The system needs to reach speeds up to 3.5 m/s at the position of impact and to remain light enough to be agile. Next, to obtain high performances for this requirement, an Iterative Quadratic Linear Controller (iLQR) method that follows minimum snap planned trajectories is implemented. Inner-loop control is delegated to a PX4 microcontroller for roll, pitch and yaw to ensure good robustness and high frequency. This approach is tested in a realistic simulation and then the complete software for this task is developed on an onboard computer. Experimental results have been conducted with a motion capture system to have the full state estimate of the system. The trajectory of the ball is also estimated by the motion capture system, giving the position and time of impact. This information is then sent to quadrotor wirelessly and the trajectory is executed immediately. To the best of our knowledge, this is the first aerial robot capable to return tennis table balls thrown by a human. Hitting rates of 40% are achieved with the real quadrotor, significantly better than what was possible before for a quadrotor.

TABLE DES MATIÈRES

RÉSUMÉ.....	III
ABSTRACT	IV
TABLE DES MATIÈRES	V
LISTE DES TABLEAUX.....	VIII
LISTE DES FIGURES.....	IX
LISTE DES SIGLES ET ABRÉVIATIONS	XII
CHAPITRE 1 INTRODUCTION.....	1
1.1 La robotique en réadaptation.....	1
1.2 Le tennis de table en robotique	1
1.3 Objectifs de recherche et plan du mémoire.....	3
CHAPITRE 2 REVUE DE LITTÉRATURE	4
2.1 Trajectoires agressives	4
2.2 Robotique de trajectoires agressives	6
2.2.1 Bras robotiques et autres systèmes pour trajectoires agressives	6
2.2.2 Quadrirotors pour trajectoires agressives.....	8
2.3 Contrôle des quadrirotors	11
2.3.1 Contrôle standard des quadrirotors traditionnels	11
2.3.2 Contrôle des quadrirotors pour trajectoire agressives	11
2.3.3 Contrôle des quadrirotors modifiés	14
2.4 Conclusion sur la revue de littérature.....	16
CHAPITRE 3 RATIONELLE DU PROJET	17
3.1 Résumé de la problématique	17
3.2 Objectif général	17

3.3	Objectifs spécifiques	17
CHAPITRE 4 ARTICLE 1: DESIGN AND DEVELOPMENT OF A NOVEL TYPE OF TABLE TENNIS AERIAL ROBOT PLAYER WITH TILTING PROPELLERS		
4.1	Introduction	19
4.2	Methods	21
4.2.1	Quadrotor dynamic model.....	21
4.2.2	Iterative LQR Controller	25
4.2.3	Hardware Setup	32
4.3	Results	35
4.4	Discussion	40
4.5	Conclusion.....	41
CHAPITRE 5 DÉTAILS SUPPLÉMENTAIRES SUR LA MÉTHODOLOGIE ET LES RÉSULTATS 49		
5.1	Résultats en simulation.....	49
5.2	Estimation de la trajectoire de la balle	53
5.3	Conception mécanique	56
5.3.1	Choix du mécanisme	56
5.3.2	Mécanisme d'inclinaison des hélices	56
5.3.3	Mécanisme de la gimbal.....	58
5.4	Choix des gains	59
CHAPITRE 6 DISCUSSION GÉNÉRALE		
6.1	Résumé de la discussion de l'article	61
6.2	Discussion additionnelle	62
6.2.1	Comparaison des résultats entre la simulation et le réel	62
6.2.2	Agressivité du quadrirotor.....	62

6.2.3	Conception mécanique	63
6.2.4	Effets des perturbations sur le iLQR	63
6.2.5	Bénéfices de l'utilisation du iLQR par rapport au LQR	64
6.3	Limites et perspectives du projet.....	64
6.3.1	Conception mécanique	65
6.3.2	Système de capture de mouvement	65
6.3.3	Estimation de la trajectoire de la balle	66
CHAPITRE 7	CONCLUSION ET RECOMMANDATIONS	67
RÉFÉRENCES	68

LISTE DES TABLEAUX

Tableau 2.1 Comparaison des robots joueurs de ping-pong	8
Tableau 2.2 Comparaison des quadrirotors joueurs de ping-pong.....	10
Tableau 4.1 Table tennis hits performed for the quadrotor evaluation.	37
Tableau 4.2 Root-mean-squared errors and aggressivity for an experimental ping pong hit.	40
Tableau 5.1 Erreur quadratique moyenne (EQM) et agressivité pour une frappe typique de ping-pong en simulation.	53

LISTE DES FIGURES

Figure 2.1 Bras robotique performant des mouvements de lancers à haute vitesse. © 2008 IEEE [14]	5
Figure 2.2 Passage à travers un cerceau réalisé par un quadrirotor. © 2011 IEEE [16]	6
Figure 2.3 Un des premier système robotique joueur de tennis de table capable de retourner des balles. © 2003 IEEE [7]	7
Figure 2.4 Quadrirotor jongleur. © 2011 IEEE [19]	9
Figure 2.5 Divers prototypes de quadrirotors modifiés. a) Quadrirotor avec hélices inclinables selon l'axe du bras. © 2014 IEEE [35] b) Quadrirotor avec hélices inclinables dans l'axe perpendiculaire au bras. © 2016 IEEE [36] c) Quadrirotor en forme de H avec bras jumelés inclinables. 2019 Creative Commons Attribution-NonCommercial-ShareAlike 4.0 International License [38]	16
Figure 4.1. Overview of the quadrotor configuration. a) Principal forces applied to the system; b) Tilt angle, α	22
Figure 4.2. Top view schematics of the quadrotor.	23
Figure 4.3 Condition scheme to evaluate the new control signal computed.	30
Figure 4.4. Overview of the components of the developed quadrotor platform. Reflective markers are used to identify the position and velocity of the quadrotor. A FlySky radio receiver is only used to power on and off the platform.	33
Figure 4.5. The developed quadrotor in action during a hitting motion.	33
Figure 4.6. System architecture used for the experiment with estimated orientation \mathbf{q} and angular velocity $\boldsymbol{\omega}$, and desired inputs \mathbf{u}^* . \mathbf{p} , \mathbf{q} , \mathbf{p} and $\boldsymbol{\omega}$ are position, orientation, linear velocity and angular velocity. \mathbf{p}^{des} and \mathbf{p}^{des} are the desired position and linear velocity provided by the trajectory generation algorithm. Φ , Φ , \mathbf{a} represent orientation, body rates and linear acceleration measurements from the IMU unit.	35

Figure 4.7. Desired (dashed red) and iLQR with tilt (solid blue) trajectories with impact point (red square) for an experimental ping pong hit.	38
Figure 4.8. Desired (dashed red) and iLQR with tilt (solid blue) cartesian positions with impact point (red square) for an experimental ping pong hit.	38
Figure 4.9. Desired (dashed red) and iLQR with tilt (solid blue) Cartesian velocities with impact point (red square) for an experimental ping pong hit.	39
Figure 4.10. iLQR with tilt (solid blue) angles with impact point (red square) for an experimental ping pong hit.	39
Figure 4.11 Control inputs with impact point (red square) for an experimental ping pong hit, where $Tcmd$, ϕcmd , θcmd , ψcmd , αcmd are the control inputs associated to $U1$, $U2$, $U3$, $U4$ and $U5$ respectively.	40
Figure 5.1 Environnement de simulation pour le quadrirotor joueur de ping-pong sur V-REP.	49
Figure 5.2 Trajectoire désirée (rouge) et actuelle (bleu) avec le point d'impact estimé (carrée rouge) pour un mouvement de frappe typique de ping-pong en simulation.	50
Figure 5.3 Positions cartésiennes désirées (rouge) et actuelles (bleu) ainsi que le point d'impact estimé (carrée rouge) pour un mouvement de frappe typique de ping-pong en simulation.	51
Figure 5.4 Vitesses cartésiennes désirées (rouge) et actuelles (bleu) ainsi que le point d'impact estimé (carrée rouge) pour un mouvement de frappe typique de ping-pong en simulation.	52
Figure 5.5 Orientations actuelles performées (bleu) ainsi que le point d'impact estimé (carrée rouge) pour un mouvement de frappe typique de ping-pong en simulation.	52
Figure 5.6 Commandes envoyées au quadrirotor ainsi que le point d'impact estimé (carrée rouge) pour un mouvement de frappe typique de ping-pong en simulation.	53
Figure 5.7 Trajectoire parabolique de la balle de ping-pong.	54
Figure 5.8 Balle de ping-pong utilisée pour les tests expérimentaux. Des morceaux carrés de ruban réfléchissant sont placés sur la balle pour la capter par le système de capture de mouvements.	56
Figure 5.9 Mécanisme d'inclinaison au niveau du servo-moteur.	57

Figure 5.10 Mécanisme d'inclinaison au niveau du bras.	58
Figure 5.11 Vue d'ensemble du mécanisme d'inclinaison du quadrirotor.	58
Figure 5.12 Mécanisme de la gimbal.	59

LISTE DES SIGLES ET ABRÉVIATIONS

	English	Français
GA	Genetic Algorithm	Algorithme Génétique
iLQR	Iterative Linear Quadratic Regulator	Commande Itérative Linéaire Quadratique
LQR	Linear Quadratic Regulator	Commande Linéaire Quadratique
MPC	Model Predictive Control	Commande Prédictive
MRAC	Model Reference Adaptive Control	Contrôle Adaptatif avec Model de Référence
PD	Proportionnal Derivative	Proportionnelle Dérivateur
PID	Proportionnal Derivative Integrator	Proportionnelle Dérivateur Intégrateur
ACI	Adenylyl cyclase 1	
ADN	Acide désoxyribonucléique	
CaM	Calmodulin	
HOT	Highly optimized tolerance	

CHAPITRE 1 INTRODUCTION

La médecine de réadaptation vise à améliorer les habiletés physiques des individus avec des déficiences ou des déshabilités physiques provenant d'une amputation, une malformation, une blessure à la coiffe des rotateurs ou un accident nerveux par exemple. Au Canada seulement, on reporte 165 000 blessures du cerveau chaque année [1]. La réadaptation diffère des autres médecines par plusieurs points dont le fait qu'il s'agisse d'un processus de d'éducation, qu'elle requière la participation de la personne handicapé en ce qui concerne la planification du programme et qu'elle exige que des objectifs soient fixés et mesurés [2]. De plus, la motivation du patient a été identifiée comme étant un facteur important influençant sur les résultats de la thérapie [3].

1.1 La robotique en réadaptation

Ainsi, les robots peuvent s'avérer plus utiles en médecine de réadaptation puisqu'ils offrent plusieurs avantages par rapport aux méthodes traditionnelles. Tout d'abord, ils permettent d'avoir des mouvements répétitifs qui sont précis et mesurables pour chaque patient. De plus, le côté ludique d'un jeu, que ce soit avec ou sans robotique, permet de rendre l'aspect répétitif plus intéressant et engageant en incorporant un semblant de progression dans celui. Plusieurs produits existent dont la plupart sont très coûteux et requièrent qu'une salle complète leur soit dédiée. Par exemple, la compagnie Redox a conçu le produit Hirob combinant un jeu de réalité virtuelle avec un robot de réadaptation qui consiste en une selle que le patient contrôle [4]. Il permet d'améliorer le control du tronc et la stabilité du patient. La compagnie Man&tel propose quant à elle un jeu dont les inputs de contrôle proviennent d'une balance tri-dimensionnelle, faisant ainsi travailler les jambes du patient [5]. De nombreuses autres produits sont également offerts.

Les bénéfices du tennis de table plus particulièrement pour la réadaptation ainsi que pour l'amélioration des capacités physiques des personnes handicapées ou âgées sont nombreux. Le tennis de table permet notamment d'améliorer la coordination visuelle et moteur, la précision et l'agilité. Il a d'ailleurs été inclus dans l'un des premier jeux paralympiques en 1960 [6].

1.2 Le tennis de table en robotique

Le tennis de table est considéré comme une tâche difficile à performer pour un bras robotique parce que cela requière d'être particulièrement rapide et précis. Plusieurs recherches ont néanmoins

adressé ce problème [7], [8]. Les performances atteintes aujourd'hui frôlent la perfection. Par exemple, le bras robotique Barrett WAM qui a 7 degrés de libertés et qui est fixé au plafond réussit à renvoyer 88% des balles servies par un humain à l'aide d'intelligence artificielle [9]. En revanche, les robots sériels comme ce dernier ne sont pas adaptés pour des trajectoires agressives comme celles requises par des mouvements de frappe au tennis de table, et ce, en raison de leur inertie trop importante. De meilleures performances peuvent en effet être atteintes par des robots parallèles comme l'Omron Forpheus [10], mais ce dernier est très imposant et utilise tout l'espace d'une pièce tout en étant aussi très coûteux.

Une avenue encore peu explorée est celle des robots aériens joueurs de tennis de table. Dans [11], un quadrirotor apprend à généraliser des mouvements de frappes basés sur la démonstration humaine. La position de la balle est traquée par une caméra embarquée. Cependant, cette recherche n'aborde que le problème d'interception de la balle. Le taux de succès atteint est de 20% depuis des balles tirées vers le drone aléatoirement. Dans [12], des résultats plus convaincants sont démontrés avec un drone. Un environnement de test pour un drone commercial est développé et démontre l'habileté du drone à renvoyer des balles tirées par un robot lanceur. Les résultats sont présentés en vidéo pour la première fois dans le cas d'un drone [13]. En revanche, les capacités agressives du drone sont poussées au maximum, ce qui requiert même l'utilisation d'une corde pour ralentir le drone dans les trajectoires les plus extrêmes. De plus, les frappes présentées manquent de variétés puisque le drone semble toujours n'aller que vers l'avant et il garde la même hauteur. Cela en fait des frappes triviales, ne demandant pas de trajectoires compliquées à suivre.

Les performances agressives requises pour un drone joueur de ping-pong sont élevées. Pour les identifier, nous faisons les hypothèses suivantes selon une trajectoire typique :

- 1 seconde est le temps alloué pour frapper une balle afin de ne pas autoriser son deuxième rebond,
- Le drone doit parcourir la moitié de la longueur d'une table de ping-pong, soit 1,38m, pour frapper une balle,
- Il est immobile en position de départ,
- Il doit finir en position de frappe avec une accélération nulle, de sorte qu'il puisse faire demi-tour à temps pour enchaîner une seconde frappe.

Cela implique en somme qu'une accélération maximale de 11.04 m/s^2 est requise pour qu'un drone soit véritablement capable de joueur au ping-pong.

1.3 Objectifs de recherche et plan du mémoire

Dans ce contexte, l'objectif de ce mémoire est de concevoir puis de contrôler un quadrirotor qui est capable de jouer au tennis de table. Plus spécifiquement, ceci implique de développer un nouveau type de robot aérien étant capable d'atteindre des accélérations élevées de 11.04 m/s^2 pour être rapide sur de courtes distances. De plus, il faudra implémenter un contrôleur avancé sur le système pour avoir une précision élevée en suivi de trajectoire puisque l'erreur permmissible pour frapper une balle n'est que de quelques centimètres.

Le mémoire est organisé de la manière suivante. Le Chapitre 2 présente le contexte théorique du projet, comportant une analyse des trajectoires agressives déjà réalisées avec des robots aériens ainsi que les stratégies de contrôles déjà explorées sur ceux-ci dans le but d'atteindre des performances toujours plus élevées dépendamment de l'application visée qu'elle soit spécifique ou générale. Ensuite, au Chapitre 3, la problématique du projet est expliquée plus en profondeur et les objectifs spécifiques de la recherche sont exposés. Les résultats principaux sont présentés au Chapitre 4 dans l'article scientifique *Design and development of a novel type of table tennis aerial robot player with tilting propellers* soumis pour publication au journal *Elsevier Mechatronics*. Au Chapitre 5 sont présenté des résultats complémentaires notamment en ce qui concerne la simulation. Finalement, les Chapitres 6-7 servent de discussion générale et de conclusion sur la recherche.

CHAPITRE 2 REVUE DE LITTÉRATURE

Le présent chapitre vise à donner un aperçu général de l'état de l'art du domaine afin de mieux faire ressortir les capacités robotiques actuelles pour le tennis de table. On pourra également identifier les défis potentiels concernant le tennis de table avec un quadrirotor. Dans cette optique, il y sera défini en premier lieu les trajectoires agressives de manière qualitative et quantitative. Des exemples plus concrets des capacités actuelles en robotique de trajectoires agressives seront présentés par la suite. Cela permettra d'identifier l'avancement dans ce domaine pour les bras robotiques et les quadrirotors. Finalement, les différentes stratégies de contrôle pour les quadrirotors seront mentionnées. Elles comprennent le contrôle standard des quadrirotors traditionnels, le contrôle des quadrirotors pour trajectoire agressives plus spécifiquement ainsi que le contrôle des quadrirotors modifiés.

2.1 Trajectoires agressives

Une trajectoire agressive dans le domaine de la robotique peut être définie comme étant une trajectoire ayant des vitesses et des accélérations relativement élevées. Le robot est près de sa vitesse maximale atteignable qui dépend de sa dynamique propre. Pour un bras robotique, la vitesse de mouvement dépend directement de la vitesse de rotation des moteurs qu'il emploie pour faire tourner ses joints.

Par exemple, Senno et coll. [14] ont atteint des manipulations dynamiques à hautes vitesses pour un bras robotique qui parvient à lancer une balle vers une cible. Celui-ci atteint des pics de vitesse aux joints de 7 rad/s ainsi que des couples aux joints de plus de 100 Nm. De plus, les couples aux joints augmentent brusquement durant la manœuvre. Ceci mène donc à un mouvement qui est agile et très rapide. La Figure 2.1 présente ce résultat.

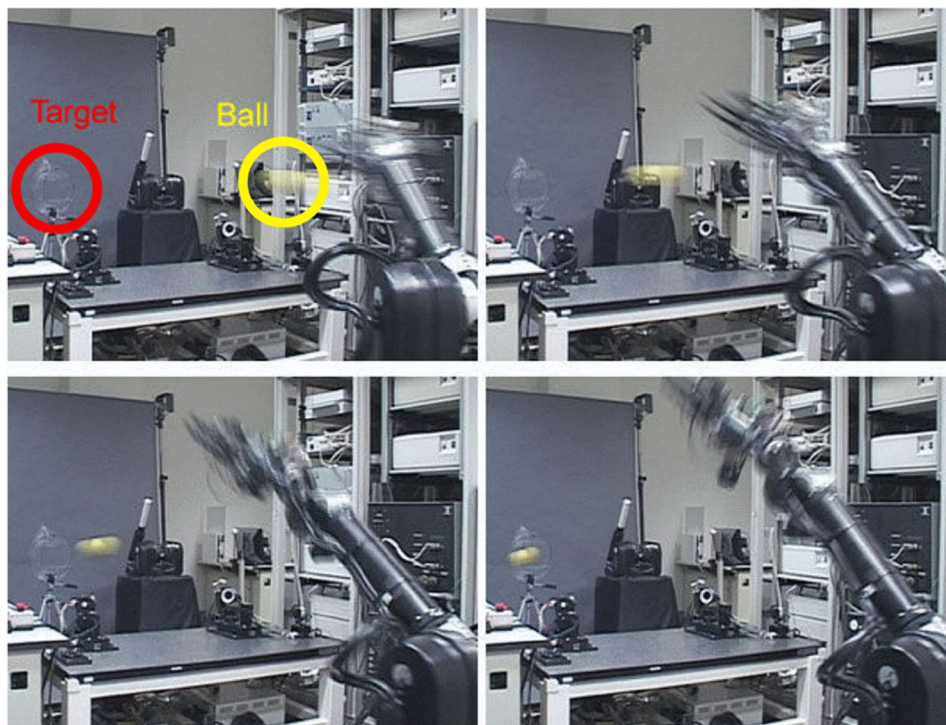


Figure 2.1 Bras robotique performant des mouvements de lancers à haute vitesse. © 2008 IEEE [14]

En ce qui concerne les quadrirotors, on pourrait définir une trajectoire agressive comme étant une trajectoire avec des changements brusques de direction, des accélérations élevées ainsi que des vitesses hautes. Pour être agressif, le quadrirotor a besoin d'avoir une inertie faible, une poussée totale qui est la plus élevée possible ainsi qu'un temps de réponse faible sur ces inputs de contrôles.

Par exemple, Loianno et coll. [15] ont pu atteindre des vitesses de 4.5 m/s ainsi que des accélérations supérieures à 1.5g avec un quadrirotor ayant seulement 250 grammes. Dans ce cas-ci, c'est surtout la masse et les dimensions faibles (15 cm de diamètre) qui ont permis d'atteindre cette agressivité. Ensuite, Mellinger et coll. [16] ont démontré l'agressivité de leur quadrirotor avec des trajectoires passant par des ouvertures verticales et horizontales étroites avec des vitesses atteignant 2.5 m/s. Ils présentent ces manœuvres agressives en vidéo [17], [18]. La Figure 2.2 montre le passage rapide d'un quadrirotor à travers un cerceau lancé par un humain qui est l'une des manœuvres réalisées pour démontrer l'agressivité de leur système.

En résumé, une trajectoire agressive est difficilement quantifiable, on peut plus facilement l'identifier de manière visuelle en la comparant à d'autres trajectoires performées avec un système

dynamique semblable. Il est également possible d'identifier une trajectoire agressive en fonction de l'application robotique en question.

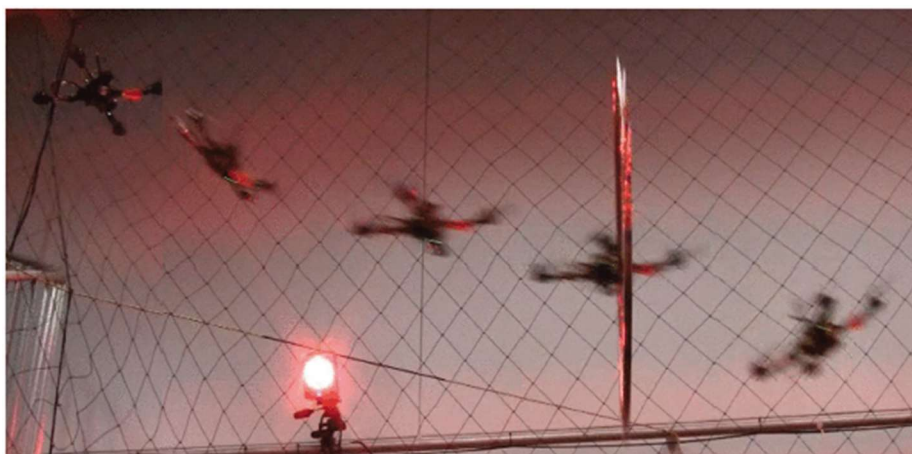


Figure 2.2 Passage à travers un cerceau réalisé par un quadrirotor. © 2011 IEEE [16]

2.2 Robotique de trajectoires agressives

Faire jouer un robot au tennis de table est également un autre exercice qui peut démontrer la performance du robot. Pour performer cette tâche, il faut évidemment planifier des trajectoires agressives et les effectuer avec précision. Donc, en demandant à la fois précision et vitesse, cela permet de prouver la qualité du contrôleur implémenté sur le système robotique.

2.2.1 Bras robotiques et autres systèmes pour trajectoires agressives

Acosta et coll. [7] ont franchis un cap en fabriquant un système robotique qui retourne environ 90% des balles contre un humain comparativement à des taux de 50% qui était atteignable à l'époque. Toutefois, le champ d'action du robot est restreint, ce qui fait gagner du temps. De plus, le robot n'est pas conçu pour pouvoir renvoyer tous les types de balles reçues, car il n'agit que dans un espace plan. La Figure 2.3 montre ce prototype.

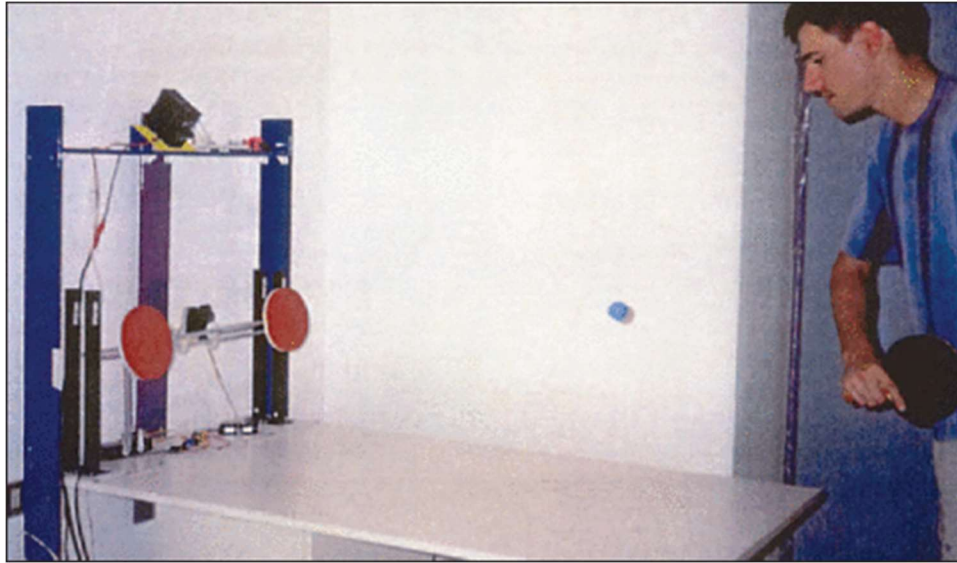


Figure 2.3 Un des premier système robotique joueur de tennis de table capable de retourner des balles. © 2003 IEEE [7]

Une différente approche est employée par Mulling et coll. [8] qui ont conçu un bras robotique biomimique pour jouer au tennis de table. Celui-ci est capable tout d'abord de retourner 99% des balles envoyées par un lanceur de balles. Ensuite, Mulling et coll. [9] ont utilisé le même bras robotique, mais en apprenant des mouvements de frappes humains, le robot est capable passer de 74.4% à 88% de balles retournées après une heure contre un humain. Cependant, les robots sériels ne sont pas adaptés à des trajectoires agressives en raison de leur grande inertie. On fait alors plutôt appel à la robotique parallèle pour atteindre de plus hautes performances pour le tennis de table. Par exemple, Omron Forpheus [10] joueur de ping-pong prendrait toute une salle, mais permet de renvoyer des balles réellement agressives qui lui sont envoyées. De plus, il permet d'effectuer plusieurs échanges de suite. Le

Tableau 2.1 présente un comparatif des différents robots joueurs de ping-pong.

Tableau 2.1 Comparaison des robots joueurs de ping-pong

Nom	Nombre de degrés de liberté	Vitesse de contrôle	Pourcentage de retour contre un humain	Vitesse maximale de la raquette	Prix	Masse (kg)
Omron Forpheus [10]	6	1 milliseconde	99% ¹	-	-	-
Ping-Pong Prototype [7]	5	-	90%	-	-	-
Barrett WAM joueur de ping-pong [8]	7	-	88%	6.5 m/s	135 000\$	25

2.2.2 Quadrirotors pour trajectoires agressives

Notons d'abord les résultats obtenus sur un quadrirotor étant capables de jongler seul ou en coopération avec un autre quadrirotor [19]. Le quadrirotor a une raquette de badminton attachée sur le dessus et il effectue des mouvements de va-et-vient verticaux. La Figure 2.4 montre ce système. En revanche, l'agressivité est surtout performée dans la direction verticale, ce qui est le plus triviale pour ce type de système puisque tous ses moteurs fournissent la poussée dans cette direction.

¹ Selon les résultats présentés en vidéo, le prototype est capable de retourner la quasi-totalité des frappes du même type que celles utilisées pour évaluer les autres robots joueurs de ping-pong.



Figure 2.4 Quadrirotor jongleur. © 2011 IEEE [19]

Ensuite, pour ce qui est des quadrirotors et des robots aériens, les tentatives de jouer au tennis de table sont beaucoup plus rare que pour les bras robotiques. En effet, il est difficile d'obtenir la même agressivité sur un quadrirotor en raison de la complexité de sa dynamique hautement instable. Donc, on n'arrive pas encore à avoir un contrôle à la fois rapide et précis. De plus, sa dynamique sous-actionnée (seulement 4 rotors pour 6 degrés de liberté) l'empêche de suivre des trajectoires parfaitement linéaires. Une tentative notable provient de Silva et coll. [11] qui ont appris à un quadrirotor à imiter des mouvements de frappes fait par un humain qui pilote le quadrirotor en premier lieu. Le quadrirotor est capable de choisir parmi une banque de frappe celle qui est la plus adaptée à la situation et de l'exécuter. En revanche, les frappes ne doivent pas être renvoyées. Le quadrirotor n'a qu'à les toucher avec la palette. Ils obtiennent seulement 30% de frappes réussies en simulation et 20% sur le modèle réel. En effet, la plupart des balles manquées le sont à cause du manque de vitesse du quadrirotor. De plus, Zhou et coll. [12] ont également réussi à démontrer quelques frappes contre un lanceur de balles avec un quadrirotor commercial. Il y a malheureusement très peu de détails sur la performance de celui-ci hormis une vidéo démontrant quelques frappes [13]. Le

Tableau 2.2 compare la performance des quadrirotors joueurs de ping-pong.

Tableau 2.2 Comparaison des quadrirotors joueurs de ping-pong

Nom	Type de quadrirotor utilisé	Masse (kg)	Dimension moteur-à-moteur (mm)	Pourcentage de retour	Prix
Quadrirotor autonome intelligent [11]	Parrot AR.Drone 2.0 [20]	0.380	520	20%	389\$
Plateforme de test pour calculs embarqués [12]	3DR IRIS+ de 3DRobotics	1.282	550	-	750\$

Ce résultat ouvre des perspectives d'améliorations potentielles. L'utilisation d'un quadrirotor pour jouer au ping-pong peut avoir quelques avantages par rapport à l'utilisation d'un bras robotique. D'emblée, le lieu exact de frappe de balle est dans un espace de travail qui est au-dessus de la table, ce qui est difficilement concevable avec un bras robotique sur socle (soit sur la table, soit sur le sol). La trajectoire du bras peut être souvent gênée par la position de la table. Le quadrirotor dépense de l'énergie à voler, mais est peut-être moins coûteux qu'un bras robotique sur socle comme le Omron Forpheus [10] ou le Barrett WAM [8] qui s'attache au plafond, et ce, dans un contexte de trajectoire agressive.

Le quadrirotor joueur de ping-pong peut également être intéressant dans une optique de réadaptation. En effet, ce peut être une manière ludique pour les personnes âgées ou handicapées de regagner leurs mobilités en jouant au ping-pong contre un robot. De plus, le système prend en théorie moins de place qu'un robot sur socle, ce qui implique qu'on n'a pas besoin de lui réserver une salle et qu'il est transportable plus facilement.

2.3 Contrôle des quadrirotors

Le défi principal de cette recherche consiste en l'obtention d'un contrôle précis en suivi de trajectoire agressive pour un quadrirotor. Nous présentons donc dans cette section les stratégies de contrôles déjà testées pour les quadrirotors simples, modifiées ou pour ceux qui ont des comportements agressifs. Cela permettra d'identifier si les contrôleurs déjà implémentés auparavant sont suffisants pour atteindre les performances requises pour jouer au tennis de table ou s'il faudra implémenter de nouveaux algorithmes plus sophistiqués.

2.3.1 Contrôle standard des quadrirotors traditionnels

Le quadrirotor peut se faire asservir par un contrôleur proportionnel-dérivatif (PD) qui est largement utilisé en industrie et qui est très simple à concevoir. C'est d'ailleurs la première stratégie de contrôle qui a été employée [21]. En effet, on peut directement lier les 4 mouvements principaux du quadrirotor qui sont le mouvement en roulis, en tangage, en lacet et le mouvement verticale à des forces virtuelles U_2 , U_3 , U_4 et U_1 respectivement. On y associe ensuite des gains sur l'erreur. Typiquement, pour contrôler le mouvement vertical, on contrôle directement la puissance totale des 4 rotors. Pour les 3 mouvements restants, nous avons la relation suivante:

$$U_{2,3,4} = Kp_{\varphi,\theta,\psi} (e_{\varphi,\theta,\psi}) + Kd_{\varphi,\theta,\psi} (\dot{e}_{\varphi,\theta,\psi}), \quad (2.1)$$

où φ, θ, ψ sont les angles de roulis, de tangage et de lacet respectivement. e représente l'erreur et \dot{e} est le taux de changement de l'erreur. On retrouve souvent l'utilisation du contrôleur proportionnel-intégrateur-dérivatif (PID) pour contrôler à la fois les angles de roulis, de tangage et de lacet ainsi que la position du quadrirotor dans l'espace [22], [23], [24]. Par la suite, le quadrirotor a servi comme une plateforme de test idéal pour tous les algorithmes de contrôles existants ainsi que pour le développement de nouveaux algorithmes, et ce, dû au défi de contrôle qu'imposait ce système.

2.3.2 Contrôle des quadrirotors pour trajectoire agressives

Le contrôle des quadrirotors pour le suivi de trajectoires agressives est un domaine de recherche qui a reçu beaucoup d'attention. Mellinger et coll. [16] ont effectué une contribution importante dans le domaine en traitant le problème de conception de trajectoire et de contrôleurs faisables dynamiquement qui amène le quadrirotor dans un état désiré dans l'espace. Un contrôleur à base

de PID non-linéaire est utilisé. Chaque trajectoire est composée de plusieurs segments et chaque phase de la trajectoire est contrôlée par un contrôleur différent. Ceux-ci sont au nombre de 3 :

- contrôleur d'attitude;
- contrôleur de maintien;
- contrôleur de suivi de trajectoire 3D.

Pour effectuer une trajectoire agressive réussie, il faut ajuster les gains de chacune des phases composants la trajectoire, ce qui se fait par un processus itératif d'essai-erreurs expérimental. Cela est donc un désavantage notable. L'agressivité des trajectoires est testée par différentes expérimentations dont des vols à travers des ouvertures étroites verticales et horizontales ainsi que le perchage sur une surface inclinée et inversée. Durant ces expérimentations, des vitesses de 2 m/s sont atteintes et des erreurs de quelques centimètres sur la trajectoire sont obtenues. Étant donné qu'il faut ajuster les gains intensivement pour chaque trajectoire agressive, le quadrirotor ne peut pas effectuer de nouvelles trajectoires agressives spontanées, ce qui limite considérablement l'application pratique.

Dans la même optique de travail, Mellinger et coll. [25] ont également publié un article important sur la génération de trajectoire pour quadrirotor. Ils développent un algorithme qui permet la génération de trajectoires optimales en temps réel à partir d'une séquence de point 3D choisi et de choix de position en lacet. L'algorithme minimise le snap du quadrirotor, soit la quatrième dérivée de la position en fonction du temps. Ensuite, un cap important a été franchi par Richter et coll. [26] qui améliorent la rapidité du calcul de la génération de trajectoire polynomial et qui permettent l'utilisation de polynomial à plus haut degré. Cela permet de choisir le temps alloué à chaque segment de la trajectoire en modifiant la vitesse le long du segment, ce qui est une grande valeur ajoutée. Cela permet de planifier et de générer des trajectoires en position 3D, mais aussi en temps.

En termes d'agressivité, Loianno et coll. [15] ont accompli un contrôle agressif avec un quadrirotor de seulement 250 grammes qui n'est équipé que d'une caméra et d'une centrale inertielle (IMU). Les expérimentations sont très semblables à son prédécesseur qui fait des trajectoires agressives à travers des fenêtres étroites. La principale contribution de l'article est l'obtention d'un contrôle agressif en n'utilisant qu'une caméra directement sur le quadrirotor pour estimer l'état de celui-ci au lieu d'avoir recours à un système externe de capture de mouvements. Des vitesses de 4.5 m/s

sont atteintes ainsi que des accélérations supérieures à 1.5g. De plus, un contrôleur non-linéaire est utilisé et des erreurs de l'ordre de quelques centimètres sur la position sont obtenues.

En cherchant à améliorer la précision du suivi de trajectoire agressive, Faessler et coll. [27] ont résolu le problème de besoin d'apprentissage itératif pour le suivi précis de trajectoires rapides. Ils y ont introduit un algorithme qui fait face à la saturation des moteurs en priorisant les entrées de contrôle. Puis, ils utilisent un algorithme de contrôle plus puissant pour le contrôle des vitesses angulaires du quadrirotor, soit un contrôleur linéaire-quadratique-régulateur (LQR) qui permet de réduire les erreurs de suivi de trajectoire. Ainsi, l'erreur en position du suivi de trajectoire est réduite de 25% par rapport à l'état de l'art dans le domaine. Par ailleurs, Faessler et coll. [28] ont également prouvé que le modèle dynamique d'un quadrirotor soumis à des effets de traînée de rotors linéaires est linéarisable en position et en lacet. Cette propriété est utilisée pour calculer des termes de contrôle feed-forward directement depuis une trajectoire à suivre. Un contrôleur à base de feedback non-linéaire est utilisé et la méthode prouve une amélioration du suivi de trajectoire à partir de vitesses supérieures à 0.5 m/s. Or, la performance n'est évaluée que sur des trajectoires en cercle ou en boucle (lemniscate). La méthode réduit de 50% l'erreur moyenne quadratique de suivi de trajectoire. Toutefois, l'accélération progressive vers des vitesses plus élevées facilite l'atteinte de telles performances.

Des tentatives d'utilisation d'algorithmes de contrôles plus sophistiqués ont été faites pour le suivi de trajectoires agressives. Ainsi, Foehn et coll. [29] ont voulu démontrer la possibilité d'utiliser un contrôleur plus avancé pour éviter d'avoir un contrôleur en cascade. Ils implémentent un contrôleur LQR qui est linéarisé en fonction de l'état du quadrirotor. Puis, les expérimentations réelles sont faites en n'utilisant qu'un système visuel à bord du quadrirotor. Toutefois, les expérimentations agressives ne sont que sur des trajectoires circulaires et le travail ne montre pas de changement de trajectoire brusque.

Kamel et coll. [30] ont implémenté de la commande prédictive (MPC), l'une des techniques de contrôle les plus puissantes en terme de précision. Ils l'utilisent pour le suivi en trajectoire de drone de diverses formes. Une trajectoire 3D est suivie pour évaluer la performance du contrôleur, mais le drone est trop lourd pour être vraiment agressif et seulement des trajectoires en boucles sont suivies. Finalement, Kamel et coll. [31] ont comparé aussi le MPC linéaire à celui non-linéaire pour le suivi de trajectoire agressif. Sous des perturbations de vents, l'erreur quadratique moyenne en suivi de

trajectoire est de 10.8 cm pour le MPC linéaire et 7.1 cm pour le MPC non-linéaire. Mais, la masse du drone ainsi que sa taille sont trop grandes pour qu'il soit adapté pour des trajectoires agressives. Puis, les trajectoires à suivre ne sont pas très rapides puisque le drone ne semble pas atteindre des vitesses plus élevées que 1 m/s.

2.3.3 Contrôle des quadrirotors modifiés

Notons que nous n'aborderons ici que les quadrirotors et non les hexarotors, les octorotors et autres types de drones avec plus de quatre hélices. Ces drones surchargés sont généralement plus lourds dû à l'ajout de moteurs et autres composantes supplémentaires et donc ils sont moins adaptés pour des trajectoires agressives.

Plusieurs tentatives ont été faites pour rendre les quadrirotors plus agiles dans les airs en les équipant d'hélices à incidences modifiables ou d'hélices inclinables. En effet, dû à sa nature sous-actionné, le quadrirotor est un système qui doit d'abord s'incliner pour avancer, car il doit orienter son vecteur force vers l'avant. Les mouvements en rotation et en translation sont par ce fait couplés. Donc, ceci peut potentiellement nuire à son agilité, et d'autant plus que son inertie augmente.

Cutler et coll. [32] ont conçu et testé en expérimentation réelle un quadrirotor avec hélice à incidence modifiable. Ils utilisent pour cela quatre servomoteurs supplémentaires en plus des quatre moteurs qui font tourner les hélices. Ils observent les avantages suivant par rapport à un quadrirotor avec hélice à incidence fixe : augmentation du taux de changement de la poussée, diminution de la saturation des contrôles et l'habilité de générer de la poussée dans les deux sens. Tout cela permet théoriquement d'être plus agressif en vol. Cependant, ils utilisent un contrôle PD basique. Les travaux démontrent notamment une amélioration par rapport à une hélice à incidence fixe pour un suivi de position verticale agressif (3.5 m/s). Or, la comparaison pour une trajectoire 3D agressive n'est pas effectuée.

En vue de faire face aux inconvénients du système standard du quadrirotor qui est sous-actionné, Ryll et coll. [33] ont conçu un nouveau type de quadrirotor aussi qui est cette fois-ci avec des hélices inclinables selon l'axe qui les relie au centre de celui-ci (voir Figure 2.5 a). Ils montrent que ce nouveau concept peut suivre des trajectoires qui sont infaisables pour un quadrirotor standard. De plus, des expérimentations réelles sont faites en suivi de trajectoires circulaires avec une attitude constante et en rotation en tangage et en roulis sur place. Mais, il n'y a pas d'essai

agressif. D'autre part, Moutinho et coll. [34] ont réalisé un concept où 2 hélices sur 4 sont fixes. Les deux hélices mobiles sont opposées l'une à l'autre et peuvent s'incliner selon l'axe qui les relie à la base du quadrirotor et également selon un autre axe perpendiculaire au dernier. Toutefois, aucun vol n'est réalisé. Finalement, Ryll et coll. [35] ont fait des tests intéressants sur leur concept. Ils font une trajectoire en forme de huit avec un quadrirotor à hélices inclinables. Ils obtiennent 3.9 cm au maximum sur la norme de l'erreur sur la position pour le suivi de trajectoire avec une moyenne de 2.2 cm. Mais, la trajectoire est lente et aucune trajectoire agressive n'est testée. Ils concluent que la dégradation des performances en vols provient du faible taux de contrôle des servomoteurs qui inclinent les hélices ainsi que de leur délai de 18 millisecondes. Odelga et coll. [36] proposent un système d'inclinaison pour les hélices originale puisque les hélices s'incline par rapport à l'axe perpendiculaire au bras (voir Figure 2.5 c). Le travail est préliminaire, néanmoins des résultats sont présentés en simulation avec un contrôleur PID.

D'autres modèles avec des systèmes dynamiques plus simples, donc potentiellement plus simple à contrôler, qui visent à améliorer l'agressivité spécifiquement sont conçus. Par exemple, Alkamachi et coll. [37] ont développé les équations dynamiques d'un quadrirotor en forme de H dont les hélices sont inclinables selon l'axe qui les relie à la base de celui-ci. Les inclinaisons sont couplées, c'est-à-dire que les quatre angles d'inclinaisons sont les mêmes en tout temps. Néanmoins, le concept, permettant de voler d'avant en arrière en maintenant la hauteur, permet d'accroître la vitesse en diminuant la traînée ainsi que d'obtenir des accélérations rapides d'avant en arrière. Le contrôleur utilisé est un PID et les gains sont ajustés par un algorithme génétique (GA). En revanche, les résultats ne sont obtenus qu'en simulation et sont assez préliminaires. De plus, aucun suivi de trajectoire n'est testé. Par ailleurs, Burke et coll. [38] ont développé un algorithme un peu plus sophistiqué, soit le MRAC (Model Reference Adaptive Control) pour contrôler un quadrirotor en forme de H avec les hélices inclinables (voir Figure 2.5 c). Ceux-ci montrent également un bras manipulateur contrôlé par un PID sur le quadrirotor. Toutefois, le bras n'est modélisé que comme une perturbation. De plus, les résultats sont présentés en simulation et pour le suivi de trajectoires non agressives.

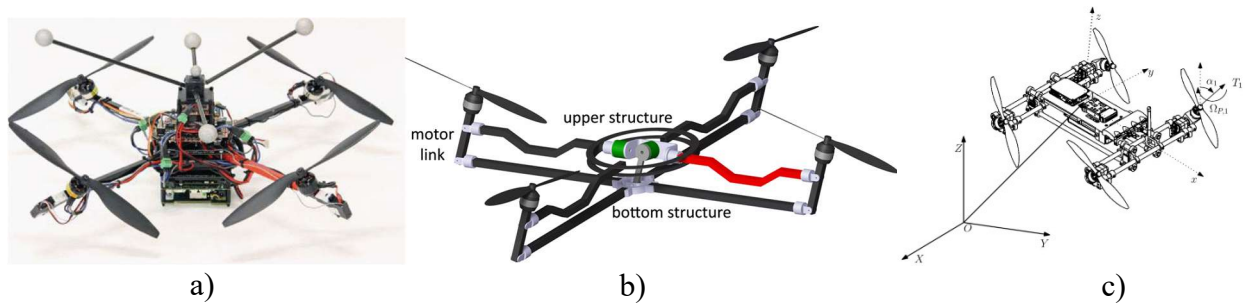


Figure 2.5 Divers prototypes de quadrirotors modifiés. a) Quadrirotor avec hélices inclinables selon l'axe du bras. © 2014 IEEE [35] b) Quadrirotor avec hélices inclinables dans l'axe perpendiculaire au bras. © 2016 IEEE [36] c) Quadrirotor en forme de H avec bras jumelés inclinables. 2019 Creative Commons Attribution-NonCommercial-ShareAlike 4.0 International License [38]

2.4 Conclusion sur la revue de littérature

Les bras robotiques joueurs de tennis de table sont nombreux et ont des taux de succès de près de 99% pour le renvoi de balles contre un humain. Ils sont aujourd'hui parfaitement capables de faire une partie de ping-pong et de mieux performer qu'un joueur humain de niveau débutant. En revanche, ils sont dispendieux et leur installation est spacieuse. En ce qui concerne les quadrirotors joueurs de ping-pong, ils sont très rares et leurs performances sont rudimentaires. Leur agressivité a été testée dans plusieurs recherches, mais n'a pas été spécifiquement testée pour cette tâche qui en requière plus que ce qui a été fait jusqu'à présent. Les contrôleurs appliqués sur les quadrirotors modifiés ou non sont souvent basiques (PD, PID et LQR), et ce, même pour ceux nécessitant de l'agressivité. Certains ont appliqués de la commande prédictive (MPC et MRAC), mais n'ont pas montré le potentiel en termes d'agressivité. On peut aussi conclure que les recherches sur la plupart des quadrirotors modifiés n'ont pas encore testé leur potentiel agressif également.

CHAPITRE 3 RATIONELLE DU PROJET

3.1 Résumé de la problématique

Les bras robotiques joueurs de tennis de table performants sont dispendieux et leur installation requière beaucoup d'espace. La plupart des robots aériens ne sont pas suffisamment rapides pour être capable d'atteindre le point d'impact désiré à temps tout en demeurant suffisamment stable. Ils ont d'abord des limitations mécaniques, car ils n'ont pas été conçu dans le but de réaliser ces types de mouvements agressifs spécifiquement. Ils sont également pourvus de contrôleurs qui ne sont pas suffisamment performants pour leur permettre d'atteindre l'agressivité et la précision requises.

3.2 Objectif général

L'objectif générale de ce mémoire est de concevoir et contrôler un quadrirotor étant capable de jouer au tennis de table.

3.3 Objectifs spécifiques

Pour concevoir un système robotique de la sorte remplissant une fonction précise, il faut passer par une série d'étapes qui sont énumérées dans les objectifs spécifiques qui suivent :

1. OS1. Développer le système mécanique du quadrirotor.
 - a. Choisir les composantes électroniques nécessaires pour le quadrirotor.
 - b. Choisir les composantes mécaniques standards qui seront achetées, celles qui seront imprimées en 3D ainsi que celles qui seront usinées.
 - c. Modéliser fidèlement le quadrirotor avec ses composantes dans le logiciel CATIAV5.
 - d. Itérer le modèle notamment en modifiant les pièces non-standards pour augmenter la robustesse du mécanisme d'inclinaison des bras, diminuer le poids, réduire le bras de levier de la poussé et diminuer les vibrations dans la structure.

- e. Assembler le quadrirotor.
2. OS2. Développer le contrôleur du quadrirotor.
- a. Modéliser la dynamique du système du quadrirotor avec des bras inclinables.
 - b. Implémenter un algorithme de commande prédictive pour le système.
 - c. Tester le contrôle en simulation pour évaluer les performances.
3. OS3. Tester les performances réelles du quadrirotor joueur de tennis de table.
- f. Développer les codes requis pour le contrôle sur un ordinateur embarqué.
 - g. Développer les codes de communication sans-fil.
 - h. Développer les codes fournissant l'estimation de la position, la vitesse et l'orientation du quadrirotor en temps réelle à partir du système de capture de mouvements.
 - i. Construire un environnement de test sécuritaire pour les résultats expérimentaux.
 - j. Générer et évaluer les résultats expérimentaux avec le quadrirotor joueur de tennis de table.

CHAPITRE 4 ARTICLE 1: DESIGN AND DEVELOPMENT OF A NOVEL TYPE OF TABLE TENNIS AERIAL ROBOT PLAYER WITH TILTING PROPELLERS

The following article was submitted to Elsevier Mechatronics journal on August 8th, 2020.

Radu Ionut Popescu, Maxime Raison, George Marian Popescu, David Saussié, Sofiane Achiche.

Abstract

Table tennis has long fascinated roboticists as a particularly difficult task which requires fast movements, accurate control and adaptation to task parameters. In addition, the development of such devices is useful for educating the undergraduate students. Up to now, several groups have developed robotic manipulators playing table tennis and other commercial interactive robots. Moreover, drones are booming right now, and they can cover a larger workspace than manipulators, for a lower device volume and at a lower price. In this context, this paper aims to propose a drone playing table tennis to analyze the benefits and the disadvantages for this task. Most vertical takeoff and landing (VTOL) vehicles are not fast enough to reproduce hitting motions. The proposed prototype is an innovative quadrotor combining a tilting propeller and a state-dependent iterative linear quadratic control (iLQR). The performance of the proposed solution is evaluated with real experiments that show the success of the approach, reaching hitting rates of 40%.

Keywords

Autonomous quadrotor; Aggressive trajectory tracking; Table tennis; Iterative Linear Quadratic Controller; Tilting propellers

4.1 Introduction

Table tennis can be a challenging task for a robot [1]. Some of the main challenges that arise when playing autonomous table tennis with a robot are precise trajectory tracking in space and time, and fast trajectory tracking. Several researches have been conducted to address this issue [2], [3]. In [4], a manipulator capable to imitate human's stroke patterns is developed. A robotic arm can return 88% of the balls served by a human player in [5] and 70% in [6]. However, serial robots are not suitable for aggressive trajectories like those required by table tennis hitting motions, since they have high inertia like the Barrett WAM table tennis player which has a mass of 25kg [3]. Better

performances can be achieved by a parallel manipulator such as the Omron Forpheus [7], but the setup takes the space of a whole room and is expensive.

To the best of our knowledge, little research has investigated the ability of a drone to play table tennis. In [8], a quadrotor is learning hitting motions based on human demonstrations. The position of the ball is tracked with an on-board camera. However, only the problem of intercepting the table tennis ball is addressed in the work, achieving success rates of 20% from balls randomly fired towards the real quadrotor. In [9], a test bed is developed to demonstrate the ability of a quadrotor to return balls from a throwing robot. A video is presenting the obtained results [10]. The aggressive capabilities of the quadrotor are tuned to maximum and this requires using a rope to decelerate the quadrotor in the most challenging hitting movements. Moreover, those hitting movements lack of variety and the quadrotor seems mostly to only move in the forward direction. Up to this date, no specific controller and design for a drone has been made to address this exact issue.

Many advances in drone control have been made in the field of aggressive trajectory tracking for drones in the past decade. In this study, the adjective "aggressive" concerning drones is defined as a state in which the drone reaches speeds of at least 2 m/s, according to the usage of the term in the references [11] and [12]. For example, a nonlinear Proportional Integral Derivative (PID) controller for quadrotors was introduced and experimentally validated in [11]. A framework for minimum snap trajectory generation was also developed, working in synergy with the controller and allowing the quadrotor to follow complex trajectories [13] other than straight lines. Each trajectory is composed of multiple segments and in order to obtain high performances on aggressive trajectories, gains must be tuned iteratively on the real system. Thus, it cannot adapt online to new trajectories. Linear Quadratic Regulator (LQR) controller has also been developed on quadrotor for aggressive trajectories [14], [15], [16]. However, only lemniscate and circular trajectories have been tested experimentally. Moreover, linear and nonlinear Model Predictive Control (MPC) have been implemented and validated experimentally on a hexarotor [17]. The size, mass and inertia of the drone though are not suitable for aggressive movements and the trajectories demonstrated do not exceed speeds of 1m/s. In [9], the time taken for a table tennis ball to cross a normal sized table of length $L = 2.76$ m was measured to be $t = 1.1$ s. If we assume that both the drone must travel a distance of 1 meter to reach the ball and that it starts in a stationary state, then it will inevitably need to reach speeds higher than 1m/s.

Other advanced optimal control methods as iLQR have the potential to overcome aggressive trajectory tracking challenges. iLQR was first introduced in [18] for nonlinear dynamic systems and compared to the most efficient algorithms using three dynamic systems: a 2 degree of freedom (DoF) arm, a model of muscle actuators and an inverted pendulum. Moreover, iLQR was also applied to a quadrotor in [19] based on a smoothing method from [20] for a cable-suspended load transportation task in simulation. Results showed that the iLQR controller produces smaller load position tracking error than the LQR method. This was validated on real world experiments for a load transportation task in [21]. In [22], H_∞ and iLQR nonlinear optimal control techniques were applied on a quadrotor for a leader-follower team formation task and path tracking. Although the iLQR method was found to give better results than H_∞ , only the last method was applied on real experiments because it showed higher robustness to model uncertainties and external disturbances. A proposed improvement was to combine the robustness of the H_∞ with the performance of the iLQR method.

Modified quadrotors have already been addressed in [23] with variable pitch actuators and tilting propellers [24], [25], [26], [27], [28]. Their potential for performing aggressive trajectories remains to be tested in real life setting.

The main objective of this work is to develop and control a quadrotor capable of playing table tennis.

4.2 Methods

4.2.1 Quadrotor dynamic model

In this Section the equations of motion for the quadrotor with tilting arms are presented. First, let the world frame, \mathcal{W} , be defined by axes x^W, y^W, z^W , and the body frame, \mathcal{B} , by x^B, y^B, z^B . Figure 4.1 illustrates the principal forces applied to the system. To go from body to world coordinates, we introduce a rotation matrix based on X-Y-Z Euler angles, associated to ϕ (roll), θ (pitch), and ψ (yaw) respectively:

$${}^W R_B = \begin{bmatrix} c(\theta)c(\psi) & s(\phi)s(\theta)c(\psi) - c(\phi)s(\psi) & c(\phi)s(\theta)c(\psi) + s(\phi)s(\psi) \\ c(\theta)s(\psi) & cs(\phi)s(\theta)s(\psi) + (\phi)c(\psi) & c(\phi)s(\theta)c(\psi) - s(\phi)c(\psi) \\ -s(\theta) & s(\phi)c(\theta) & c(\phi)c(\theta) \end{bmatrix} \quad (4.1)$$

where $c(\cdot)$ is the cosine function and $s(\cdot)$ is the sine function.

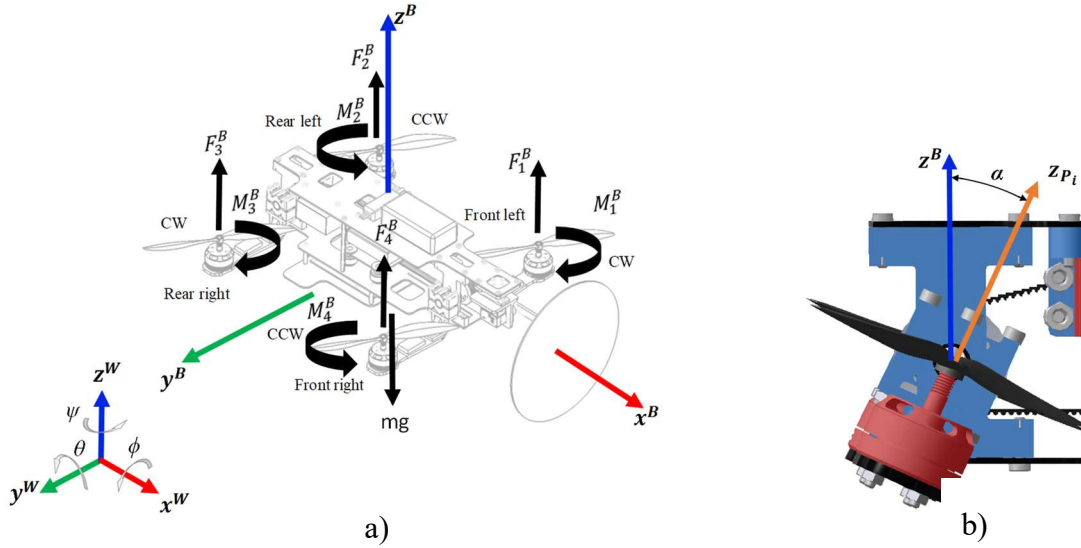


Figure 4.1. Overview of the quadrotor configuration. a) Principal forces applied to the system; b) Tilt angle, α .

Each motor generates a force in the z_{P_i} -direction according to the spinning speed Ω_i of the i^{th} propeller and its k_n rotor thrust coefficient. Torque is also generated in relation to the rotor drag coefficient k_m in the same rotation direction of the spinning propeller. Thus, negative torque is generated by clockwise (CW) rotors and positive torque for counter clockwise (CCW) rotors. A basic convention is to set the front right and rear left propeller to rotate counter clockwise, and the remaining clockwise. The relation between a propeller's thrust, torque and rotational speed has been found to be quadratic from blade element and momentum theory [29], and also deduced from experimental system identification [11]. Then, F_i^B and M_i^B can be formulated as follows:

$$F_i = k_n \Omega_i^2 \quad \text{for } i = 1, 2, 3, 4 \quad (4.2)$$

$$M_i = (-1)^i k_m F_i \quad \text{for } i = 1, 2, 3, 4 \quad (4.3)$$

Next, let us define control inputs to simplify our dynamic representation. Let U_1 be the total generated force in the z-axis by the four rotors. U_2 , U_3 and U_4 represent roll, pitch and yaw motions respectively:

$$\begin{aligned}
U_1 &= k_n \cos \alpha \sum_{i=1}^4 \Omega_i^2, \\
U_2 &= Lk_n(\Omega_1^2 + \Omega_2^2 - \Omega_3^2 - \Omega_4^2) \cos \alpha + k_mk_n(-\Omega_1^2 + \Omega_2^2 - \Omega_3^2 + \Omega_4^2) \sin \alpha, \\
U_3 &= Lk_n(-\Omega_1^2 + \Omega_2^2 + \Omega_3^2 - \Omega_4^2) \cos \alpha, \\
U_4 &= Lk_n(-\Omega_1^2 - \Omega_2^2 + \Omega_3^2 + \Omega_4^2) \sin \alpha + k_mk_f(-\Omega_1^2 + \Omega_2^2 - \Omega_3^2 + \Omega_4^2) \cos \alpha.
\end{aligned} \tag{4.4}$$

For consistency, lets also represent U_5 as the tilting angle since we are controlling it directly the same way as the other control inputs. Lever arm, L , for the rotors is the same in the y-axis and the z-axis, and the ping pong paddle is placed at a distance l from the center of gravity (CoG). This is shown in Figure 4.2.

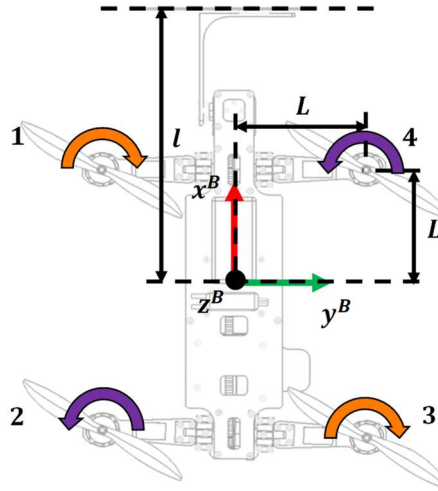


Figure 4.2. Top view schematics of the quadrotor.

For this specific configuration, the allocation matrix \mathcal{A}_N and \mathcal{A}_M for translation and torque which links force and moment distribution to our control inputs has been found in [27] as

$$\mathcal{A}_N = k_n \begin{bmatrix} s(U_5) & s(U_5) & s(U_5) & s(U_5) \\ 0 & 0 & 0 & 0 \\ c(U_5) & c(U_5) & c(U_5) & c(U_5) \end{bmatrix}. \tag{4.5}$$

$$\mathcal{A}_M = \begin{bmatrix} Lc(U_5) - k_ms(U_5) & Lc(U_5) + k_ms(U_5) & -Lc(U_5) - k_ms(U_5) & -Lc(U_5) + k_ms(U_5) \\ -Lc(U_5) & Lc(U_5) & Lc(U_5) & -Lc(U_5) \\ -Ls(U_5) - k_mc(U_5) & -Ls(U_5) + k_mc(U_5) & Ls(U_5) - k_mc(U_5) & Ls(U_5) + k_mc(U_5) \end{bmatrix}. \tag{4.6}$$

This has been used to accurately represent the model in simulation.

The translation dynamics representing the acceleration of the center of mass neglecting drag are as follows [27]:

$$m\ddot{\mathbf{p}} + \begin{bmatrix} 0 \\ 0 \\ mg \end{bmatrix} = {}^wR_B U_1 \begin{bmatrix} \tan(U_5) \\ 0 \\ 1 \end{bmatrix}. \quad (4.7)$$

Next, torque equations can be found using Euler equations of motion:

$$I\dot{\boldsymbol{\omega}}_B + \boldsymbol{\omega}_B \times I\boldsymbol{\omega}_B = \sum_{i=1}^4 ({}^B\mathbf{r}_{P_i} \times {}^B\mathbf{R}_{P_i} \mathbf{F}_i - {}^B\mathbf{R}_{P_i} \mathbf{M}_i), \quad (4.8)$$

where $\boldsymbol{\omega}_B$ and $\dot{\boldsymbol{\omega}}_B$ are defined as the angular velocity and the angular acceleration of the quadrotor in the body frame respectively and with the distance from the quadrotor center of mass and each propeller defined as follows:

$${}^B\mathbf{r}_{P_i} = \mathbf{R}_z \left((i-1) \frac{\pi}{2} \right) \begin{bmatrix} L \\ L \\ 0 \end{bmatrix}. \quad (4.9)$$

Here, \mathbf{R}_z is the rotation matrix around the z axis defined as

$$\mathbf{R}_z = \begin{bmatrix} c(\theta) & -s(\theta) & 0 \\ s(\theta) & c(\theta) & 0 \\ 0 & 0 & 1 \end{bmatrix}, \quad (4.10)$$

and ${}^B\mathbf{R}_{P_i}$ is the matrix defining the rotation of the i th propeller frame to the quadrotor body frame

$${}^B\mathbf{R}_{P_i} = \begin{bmatrix} c(\alpha) & 0 & s(\alpha) \\ 0 & 1 & 0 \\ -s(\alpha) & 0 & c(\alpha) \end{bmatrix}, \quad (4.11)$$

By developing the right side of equation 4.8, the torque dynamics are obtained:

$$\mathbf{M}_B = \mathbf{J}_B \dot{\boldsymbol{\omega}}_B + \boldsymbol{\omega}_B \times \mathbf{J}_B \boldsymbol{\omega}_B = \mathcal{A}_M \begin{bmatrix} \Omega_1^2 \\ \Omega_2^2 \\ \Omega_3^2 \\ \Omega_4^2 \end{bmatrix}. \quad (4.12)$$

To achieve high performance, it is crucial that the inner loop controller, responsible for attitude and body rates control runs faster than the outer loop. In this work we used a commercial flight controller for the inner loop dynamics as its performance has been tested and validated multiple times in similar research [17]. As in [17], we consider the inner loop system dynamics in a cascaded approach for the high level controller running at lower rates. We also used the system identification

approach to identify the first order angular dynamics. Then the first order dynamics for the inner loop can be expressed as follows:

$$\dot{\phi} = \frac{1}{\tau_{\phi}} (k_{\phi} \phi_{des} - \phi), \quad (4.13)$$

$$\dot{\theta} = \frac{1}{\tau_{\theta}} (k_{\theta} \theta_{des} - \theta), \quad (4.14)$$

$$\dot{\psi} = \dot{\psi}_{des} \quad (4.15)$$

where τ_{ϕ} , τ_{θ} and k_{ϕ} , k_{θ} are time constants and gains for roll and pitch movements respectively. ϕ_{des} , θ_{des} and $\dot{\psi}_{des}$ are the desired roll, pitch and yaw angular velocity. As a simplification, the vehicle heading is assumed to follow the desired yaw instantaneously. This is possible since it has no effect on position tracking. Moreover, yaw errors for this application are resolved with the use of a gimbal, since even if the yaw is off by a high degree it is possible to correct this error by giving a yaw command to the gimbal and hitting the ball straight. Constants defined in these expressions are identified experimentally by flying the system manually without the high-level controller acting on it.

Now, the simplified nonlinear model of the system can be derived from Equations (4.7), (4.13), (4.14) and (4.15) as:

$$\left\{ \begin{array}{l} \dot{\phi} = \frac{1}{\tau_{\phi}} (k_{\phi} U_1 - \phi) \\ \dot{\theta} = \frac{1}{\tau_{\theta}} (k_{\theta} U_2 - \theta) \\ \psi = U_3 \\ \ddot{x} = \frac{U_1}{m} (c(\phi)s(\theta)c(\psi) + s(\phi)s(\psi) + c(\theta)c(\psi)\tan(U_5)) \\ \ddot{y} = \frac{U_1}{m} (c(\phi)s(\theta)s(\psi) - s(\phi)c(\psi) + c(\theta)s(\psi)\tan(U_5)) \\ \ddot{z} = -g + \frac{U_1}{m} (c(\phi)c(\theta) - s(\theta)\tan(U_5)) \end{array} \right. \quad (4.16)$$

4.2.2 Iterative LQR Controller

In combination with the tilt, the main contribution of this work is the implementation of an iLQR optimal controller for a quadrotor table tennis player. In this section, this model-based optimal

control method that optimizes a whole control sequence for discrete time nonlinear dynamics is presented. To solve the optimal control problem, we use an approach based on a differential dynamic programming (DDP) algorithm [30]. In Subsection 4.2.2.1 we describe the controller and in Subsection 4.2.2.2 we discuss its implementation.

4.2.2.1 Controller description

Definitions

To formulate the iLQR, we first define the state vector $\mathbf{x} \in \mathbb{R}^9$ as follows:

$$\mathbf{x} = (\phi \quad \theta \quad \psi \quad x \quad \dot{x} \quad y \quad \dot{y} \quad z \quad \dot{z})^T, \quad (4.17)$$

and control inputs vector $\mathbf{u} \in \mathbb{R}^5$:

$$\mathbf{u} = (U_1 \quad U_2 \quad U_3 \quad U_4 \quad U_5)^T. \quad (4.18)$$

The discrete-time dynamics are modeled by the function \mathbf{f} found in Equations (4.16), such that

$$\mathbf{x}_{t+1} = \mathbf{f}(\mathbf{x}_t, \mathbf{u}_t), \quad (4.19a)$$

$$\mathbf{f} = \begin{bmatrix} \frac{1}{\tau_\phi} (k_\phi u_1 - x_1) \\ \frac{1}{\tau_\theta} (k_\theta u_2 - x_2) \\ u_3 \\ x_5 \\ \frac{u_1}{m} (c(x_1)s(x_2)c(x_3) + s(x_1)s(x_3) + c(x_2)c(x_3)\tan(u_5)) \\ x_7 \\ \frac{u_1}{m} (c(x_1)s(x_2)s(x_3) - s(x_1)c(x_3) + c(x_2)s(x_3)\tan(u_5)) \\ x_9 \\ -g + \frac{u_1}{m} (c(x_1)c(x_2) - s(x_2)\tan(u_5)) \end{bmatrix}, \quad (4.19b)$$

describes the system evolution from time t to $t+1$. Now, let's define a trajectory $\{\mathbf{X}, \mathbf{U}\}$ on a finite horizon N as a sequence of states $\mathbf{X} = \{\mathbf{x}_0, \mathbf{x}_1, \dots, \mathbf{x}_N\}$ resulting from the corresponding control sequence $\mathbf{U} = \{\mathbf{u}_0, \mathbf{u}_1, \dots, \mathbf{u}_{N-1}\}$. The iLQR algorithm optimizes the solution by reducing the total cost related to the control inputs. We define the *total cost* function J as the sum of the immediate cost ℓ and the final cost ℓ_f incurred during the trajectory when starting from \mathbf{x}_0 :

$$J(\mathbf{x}_0, \mathbf{U}) = \sum_{t=0}^{N-1} \ell(\mathbf{x}_t, \mathbf{u}_t) + \ell_f(\mathbf{x}_N) . \quad (4.20)$$

The *cost-to-go* is defined as the cost from our state t to N :

$$J_t(\mathbf{x}, \mathbf{U}_t) = \sum_{i=t}^{N-1} \ell(\mathbf{x}_i, \mathbf{u}_i) + \ell_f(\mathbf{x}_N) , \quad (4.21)$$

where $\mathbf{U}_t = \{\mathbf{u}_t, \mathbf{u}_{t+1}, \dots, \mathbf{u}_{N-1}\}$ represents the control sequence that remains to be applied to the system. We define the value function V as:

$$V_t(\mathbf{x}) = \min_{\mathbf{U}_t} J_t(\mathbf{x}, \mathbf{U}_t) . \quad (4.22)$$

At the final time step, it simplifies to

$$V(\mathbf{x}_N) = \min_{\mathbf{U}_t} \ell_f(\mathbf{x}_N) . \quad (4.23)$$

Then the value function results in being the optimal sum of the immediate cost with the value function, V' , meaning that it is being evaluated at the next time step:

$$V(\mathbf{x}) = \min_{\mathbf{u}} [\ell(\mathbf{x}, \mathbf{u}) + V'(\mathbf{f}(\mathbf{x}, \mathbf{u}))] . \quad (4.24)$$

Forward rollout

The first part of the algorithm involves simulating the dynamics using function \mathbf{f} , a given \mathbf{U} , and an initial state \mathbf{x}_0 in order to generate the trajectory $\{\mathbf{X}, \mathbf{U}\}$. This is called the forward rollout. The total cost of the trajectory can then be calculated as the sum of the immediate cost at each time step.

Backward pass

The goal of the backward pass is to compute a local solution to (4.24) using a quadratic Taylor expansion. To achieve convergence to a local minimum, we add small perturbations $(\delta\mathbf{x}, \delta\mathbf{u})$ in the value function:

$$Q(\delta\mathbf{x}, \delta\mathbf{u}) = \ell(\mathbf{x} + \delta\mathbf{x}, \mathbf{u} + \delta\mathbf{u}) + V'(\mathbf{f}(\mathbf{x} + \delta\mathbf{x}, \mathbf{u} + \delta\mathbf{u})) . \quad (4.25)$$

The second-order expansion of (4.25) is as follows:

$$Q_{\mathbf{x}} = \ell_{\mathbf{x}} + \mathbf{f}_{\mathbf{x}}^T V'_{\mathbf{x}} , \quad (4.26a)$$

$$Q_{\mathbf{u}} = \ell_{\mathbf{u}} + \mathbf{f}_{\mathbf{u}}^T V'_{\mathbf{x}}, \quad (4.26b)$$

$$Q_{\mathbf{xx}} = \ell_{\mathbf{xx}} + \mathbf{f}_{\mathbf{x}}^T V'_{\mathbf{xx}} \mathbf{f}_{\mathbf{x}} + V'_{\mathbf{x}} \cdot \mathbf{f}_{\mathbf{xx}}, \quad (4.26c)$$

$$Q_{\mathbf{ux}} = \ell_{\mathbf{ux}} + \mathbf{f}_{\mathbf{u}}^T V'_{\mathbf{xx}} \mathbf{f}_{\mathbf{x}} + V'_{\mathbf{x}} \cdot \mathbf{f}_{\mathbf{ux}}, \quad (4.26d)$$

$$Q_{\mathbf{uu}} = \ell_{\mathbf{uu}} + \mathbf{f}_{\mathbf{u}}^T V'_{\mathbf{xx}} \mathbf{f}_{\mathbf{u}} + V'_{\mathbf{x}} \cdot \mathbf{f}_{\mathbf{uu}}, \quad (4.26e)$$

where $\mathbf{f}_{\mathbf{x}}$, $\mathbf{f}_{\mathbf{u}}$, $\mathbf{f}_{\mathbf{xx}}$, $\mathbf{f}_{\mathbf{uu}}$, $\mathbf{f}_{\mathbf{ux}}$, $\ell_{\mathbf{x}}$, $\ell_{\mathbf{u}}$, $\ell_{\mathbf{xx}}$, $\ell_{\mathbf{ux}}$ and $\ell_{\mathbf{uu}}$ are first and second order partial derivatives. The last terms in Equations (4.26c, 4.26d, 4.26e) denote the product of a vector. The optimal perturbation on the control $\delta \mathbf{u}^*$ to the initial guess \mathbf{U} is found by minimizing the cost of Q :

$$\delta \mathbf{u}^*(\delta \mathbf{x}) = \underset{\delta \mathbf{u}}{\operatorname{argmin}} Q(\delta \mathbf{x}, \delta \mathbf{u}) = \mathbf{k} + \mathbf{K} \delta \mathbf{x}, \quad (4.27)$$

where

$$\mathbf{k} = -Q_{\mathbf{uu}}^{-1} Q_{\mathbf{u}}, \quad (4.28)$$

and

$$\mathbf{K} = -Q_{\mathbf{uu}}^{-1} Q_{\mathbf{ux}}, \quad (4.29)$$

are the feedforward and the feedback gain matrix respectively coming from derivation developed in [18]. We then plug this policy back into the second-order expansion of Q and obtain after simplification the following results:

$$V_{\mathbf{x}} = Q_{\mathbf{x}} - \mathbf{K}^T Q_{\mathbf{uu}} \mathbf{k}, \quad (4.30a)$$

$$V_{\mathbf{xx}} = Q_{\mathbf{xx}} - \mathbf{K}^T Q_{\mathbf{uu}} \mathbf{K}. \quad (4.30b)$$

Control update

After calculating the previous values, we evaluate the new trajectory with a line search of the locally-linear policy as follows:

$$\hat{\mathbf{x}}_0 = \mathbf{x}_0, \quad (4.31a)$$

$$\hat{\mathbf{u}}_i = \mathbf{u}_i + \mathbf{k}_i + \mathbf{K}_i(\hat{\mathbf{x}}_i - \mathbf{x}_i), \quad (4.31b)$$

$$\hat{\mathbf{x}}_{i+1} = \mathbf{f}(\hat{\mathbf{x}}_i, \hat{\mathbf{u}}_i). \quad (4.31c)$$

If the cost of the new trajectory is less, we simulate another trajectory starting from the last one and repeat the process until a convergence threshold is reached.

4.2.2.2 Controller implementation

Design choices

We implemented the optimal control method in a way that the nonlinear dynamics is subject to linearization at each time step as in [18]:

$$\mathbf{x}_{t+1} = \mathbf{A}_t \mathbf{x}_t + \mathbf{B}_t \mathbf{u}_t . \quad (4.32)$$

$\mathbf{A}_t \in \mathbb{R}^{9 \times 9}$ and $\mathbf{B}_t \in \mathbb{R}^{9 \times 5}$ are the Jacobian of \mathbf{f} with respect to \mathbf{x} and \mathbf{u} respectively at time step t and are presented in Appendix A. We choose the following quadratic form cost functions as it was found to give the best results:

$$\ell(\mathbf{x}_t, \mathbf{u}_t) = \frac{1}{2} (\mathbf{x}_t - \mathbf{x}_{t,des})^T \mathbf{Q} (\mathbf{x} - \mathbf{x}_{t,des}) + \frac{1}{2} \mathbf{u}_t^T \mathbf{R} \mathbf{u}_t , \quad (4.33a)$$

$$\ell_f(\mathbf{x}_N) = \frac{1}{2} (\mathbf{x}_N - \mathbf{x}_{N,des})^T \mathbf{Q} (\mathbf{x} - \mathbf{x}_{N,des}) , \quad (4.33b)$$

where $\mathbf{x}_{t,des}$ denotes the desired state at time step t and $\mathbf{x}_{N,des}$ is the desired state at the end of the horizon. \mathbf{Q} is defined as the state-cost weighted matrix and \mathbf{R} as the control weighted matrix and need to be tuned. A prediction horizon N of 50 steps corresponding to 0.5 seconds is chosen as a default value. Choosing a value of 20 or less can make the quadrotor unstable on the returning phase of the trajectory since it does not predict that fast change of direction early enough in time to adjust its aggressivity. Choosing other values up to 100 does not make a significant difference on the solution.

Note that the costs are expressed as least-square residuals in the form of $c(\mathbf{x}, \mathbf{u}) = \frac{1}{2} \|(\mathbf{x}_t - \mathbf{x}_{t,des})\|^2$. When this is the case, the Hessians of \mathbf{f} can be neglected [18]. Usually, DDP requires second-order derivatives of the dynamics. Since we keep only first-order terms, we have a Gauss-Newton approximation known as iLQR [18]. DDP and iLQR have been compared in [18] for three different control problems. iLQR finds a better solution and, in all cases, iLQR arrives at the solution in less iterations and with less computation time.

The iLQR has the convergence of quasi-Newton method [18] and thus converge to at least a local minimum.

The sequence of the algorithm is implemented as follows:

1. Initialize with the current state \mathbf{x}_0 of the quadrotor and set the initial control sequence $\mathbf{U}_0 = \{\mathbf{u}_{0_0}, \mathbf{u}_{0_1}, \dots, \mathbf{u}_{0_{N-1}}\}$ where $\mathbf{u}_0 = (mg \ 0 \ 0 \ 0 \ 0)^T$ for all the horizon.
 2. Do a forward rollout, i.e. simulate a new trajectory by integrating (19b) with finite differences for a given \mathbf{U} to get the state trajectory \mathbf{X} .
 3. Compute the total cost of the trajectory as the sum of the immediate cost ℓ for each time step.
 4. Compute derivatives $\ell_{\mathbf{x}}, \ell_{\mathbf{u}}, \ell_{\mathbf{xx}}, \ell_{\mathbf{ux}}, \ell_{\mathbf{uu}}$ of cost and $\mathbf{f}_{\mathbf{x}}, \mathbf{f}_{\mathbf{u}}$ with \mathbf{A}_t and \mathbf{B}_t for each time step.
 5. Do a backward pass, i.e., estimate the value function recursively to find \mathbf{K} and \mathbf{k} .
 6. Compute the new control signal \mathbf{U}_{new} with (4.31) iteratively for each time step using the dynamics \mathbf{f} of the system to find the new trajectory \mathbf{X}_{new} .
 7. Evaluate the new trajectory as in step 3 and set the initial control sequence $\mathbf{U}_0 = \mathbf{U}_{\text{new}}$.
- Finally, we need to follow the conditions in Figure 4.3.

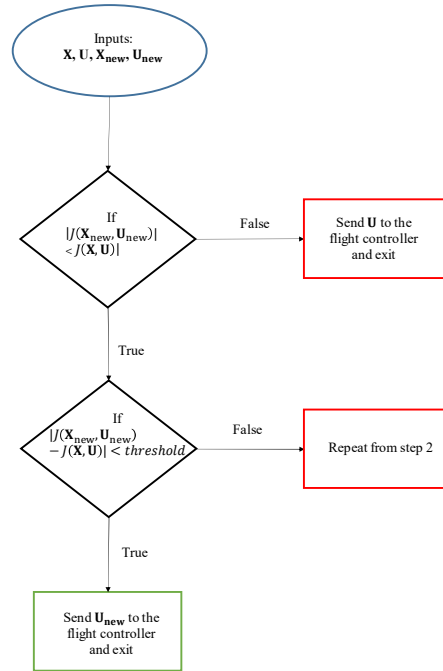


Figure 4.3 Condition scheme to evaluate the new control signal computed.

A typical convergence threshold value of 0.001 is chosen.

Minimum snap trajectory integration

To have a good estimate of what an aggressive trajectory should look like, we use a framework to create a minimum snap trajectory for a whole sequence and at a chosen discretization time

matching 0.01 seconds based on [13] and [12]. To have an optimal trajectory, we smoothly transition through the waypoints. We write a trajectory as piecewise functions of order n over m time intervals as follows:

$$\sigma_T(t) = \begin{cases} \sum_{i=0}^n \sigma_{Ti1} t^i & t_0 \leq t < t_1 \\ \sum_{i=0}^n \sigma_{Ti2} t^i & t_1 \leq t < t_2 \\ \vdots & \\ \sum_{i=0}^n \sigma_{Tim} t^i & t_{m-1} \leq t < t_m \end{cases}, \quad (4.34)$$

where $\sigma = [x, y, z, \psi]^T$ and σ_T is the trajectory of one single output. To solve the optimization problem, we minimize the following:

$$\begin{aligned} \min \int_{t_0}^{t_m} \mu \left\| \frac{d^k \sigma_T}{dt^k} \right\|^2 dt, \\ \text{s.t.} \quad \sigma_T(t_i) = \sigma_i, \quad i = 0, \dots, m \\ \frac{d^p \sigma_T}{dt^p} \Big|_{t=t_j} = 0 \text{ or free, } j = 0, m; p = 1, \dots, k \end{aligned} \quad (4.35)$$

where μ is a constant that makes the integrand nondimensional and $k = 4$ since we want to minimize the snap which is the fourth derivative.

All trajectories are generated by imposing 3 waypoints: departure position, impact position and return position. The departure position is the actual position of the quadrotor and we set the return position to be the same. The impact position is the estimated position where the quadrotor needs to hit the ball at an estimated time of impact t_{impact} . Each waypoint is defined as follows:

$$\mathbf{s} = \{t, x, x^{(1)}, x^{(2)}, x^{(3)}, x^{(4)}, y, y^{(1)}, y^{(2)}, y^{(3)}, y^{(4)}, z, z^{(1)}, z^{(2)}, z^{(3)}, z^{(4)}, \psi, \psi^{(1)}, \psi^{(2)}, \psi^{(3)}, \psi^{(4)}\}, \quad (4.36)$$

where t is a time constraint, and $x, x^{(1)}, x^{(2)}, x^{(3)}, x^{(4)}$ denotes position, velocity, acceleration, jerk and snap constraints on coordinate x . We set the following constraints on the departure \mathbf{s}_d , impact \mathbf{s}_i and return \mathbf{s}_r waypoints:

$$\mathbf{s}_d = \{t_d, x_d, 0, 0, 0, 0, y_d, 0, 0, 0, 0, z_d, 0, 0, 0, 0, \psi_d, 0, 0, 0, 0\}, \quad (4.37a)$$

$$\mathbf{s}_i = \{t_{impact}, x_i, 2, 0, 0, 0, y_i, 0, 0, 0, 0, z_i, 0, 0, 0, 0, \psi_i, 0, 0, 0, 0\}, \quad (4.37b)$$

$$\mathbf{s}_r = \{t_r, x_r, 0, 0, 0, 0, y_r, 0, 0, 0, 0, z_r, 0, 0, 0, 0, \psi_r, 0, 0, 0, 0\}, \quad (4.37c)$$

where $t_d, x_d, y_d, z_d, \psi_d$ are constraints on departure time, x-y-z coordinates and yaw respectively. \emptyset means that there is no constraint. Moreover, we always set t_d to zero and find suitable for our application to always set t_r to 2 seconds. In fact, having a lower value can limit our capacity to remain stable on certain trajectories since we will have to travel a high distance in a shorter period of time. On the other hand, having a value higher than 2 seconds does not allow us to come at the initial position fast enough to hit multiple balls in a row. t_{impact}, x_i, y_i and z_i are calculated by the framework estimating the parabolic trajectory of the ball. Note that yaw is chosen to be always zero to have better performances on trajectory generation and we use a gimbal to orient the paddle towards the center of the table.

This computes a whole sequence of 200 points for 2 seconds which is then fed to the iLQR controller per batches of $N = 50$ as the desired states $x, \dot{x}, y, \dot{y}, z, \dot{z}$, in \mathbf{x}_{des} and we set all desired angles to zero during the whole trajectory.

4.2.3 Hardware Setup

Our prototype is lightweight and agile enough to achieve aggressive behavior. The quadrotor is custom-built from a mix of 3D printed parts and off-the-shelf parts to accommodate special requirements such as space for a moving paddle, integration of tilting arms and compact installation of electronic parts. The overall cost including all electronic and mechanical parts is under 1500CAN\$. Moreover, the frame sheets are made of 3mm custom made machined carbon fiber. The developed quadrotor platform weight 1.255 kg and has a maximal thrust-to-weight ratio of ~ 2.3 at full throttle. From our detailed CAD model, we also obtained the following inertia matrix, neglecting non-diagonal terms

$$I = \begin{bmatrix} 0.004 & 0 & 0 \\ 0 & 0.011 & 0 \\ 0 & 0 & 0.013 \end{bmatrix} [\text{kg m}^2].$$

The mechanical frame is designed so that it forms a H-shape with the tilting arms. It features high-end quadrotor components whose performances have been validated in similar research [14], including 6 inch propellers (Gemfan 6045), brushless motors (Cobra CM-2208/20 2000Kv) and consumer-grade electronic speed controllers (ESC) to fit with the motors (DYS XSD20A). The tilting mechanism is made of a timing pulley and belt mechanism connected to the front and back

arms allowing their rotation through an ultra-high-speed servo motor for the tilting actuation (Corally CS-5016 HV), visible in Figure 4.4 and Figure 4.5. This has a maximum torque of 1.62 Nm and a maximum rotation speed of 19.1 rad/s. A Battery Eliminator Circuit (BEC) provides voltage conversion for the servo motor.

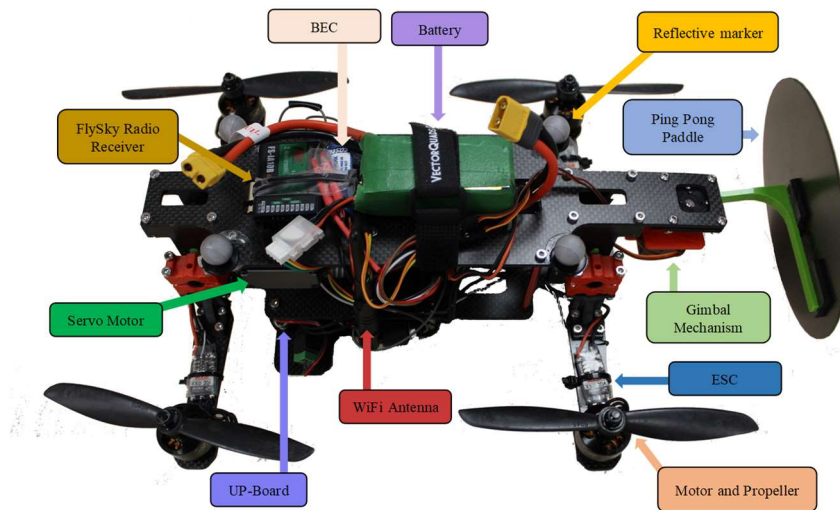


Figure 4.4. Overview of the components of the developed quadrotor platform. Reflective markers are used to identify the position and velocity of the quadrotor. A FlySky radio receiver is only used to power on and off the platform.



Figure 4.5. The developed quadrotor in action during a hitting motion.

The controller architecture is split into a fast-low-level controller for attitude and body rates control. On top of that a high-level iLQR controller was implemented. Low-level control is provided by a Pixhawk flight stack which has both accelerometer and gyroscope sampling data at 1kHz [31]. The data is then integrated, filtered, and can be published at 250Hz on the companion computer. An UP-board with Intel® ATOM™ x5-Z8350 64 bits processors, 1.92GHz, 4GB DDR3L RAM and 32GB eMMC stockage is then used as an on-board computer [32]. This runs the high-level controller at 100Hz using an Ubuntu operating system [33] with ROS [34]. The UP-board is powered by a 12V 3S Lipo battery, with voltage conversion and distribution performed by a power-supply board. The quadrotor platform is also equipped with a Wifi USB-dongle for time synchronization and terminal access, and a Laird RM024 radio module for receiving state estimates and trajectory commands. State estimates are obtained by a BTS SMART-DX 700 motion capture system [35] able to run constantly at only 45Hz on an ethernet User Datagram Protocol (UDP). Thus, position and linear velocity estimates are acquired at lower rates. Attitude estimates are fused together with the Inertial Measurement Unit (IMU) measurements via an Extended Kalman Filter (EKF), provided to the high-level controller at 100Hz and then updated at 45Hz. This means that the estimated attitude is updated at 100Hz, while the estimated position/velocity is only updated at 45Hz. Figure 4.6 illustrates this architecture.

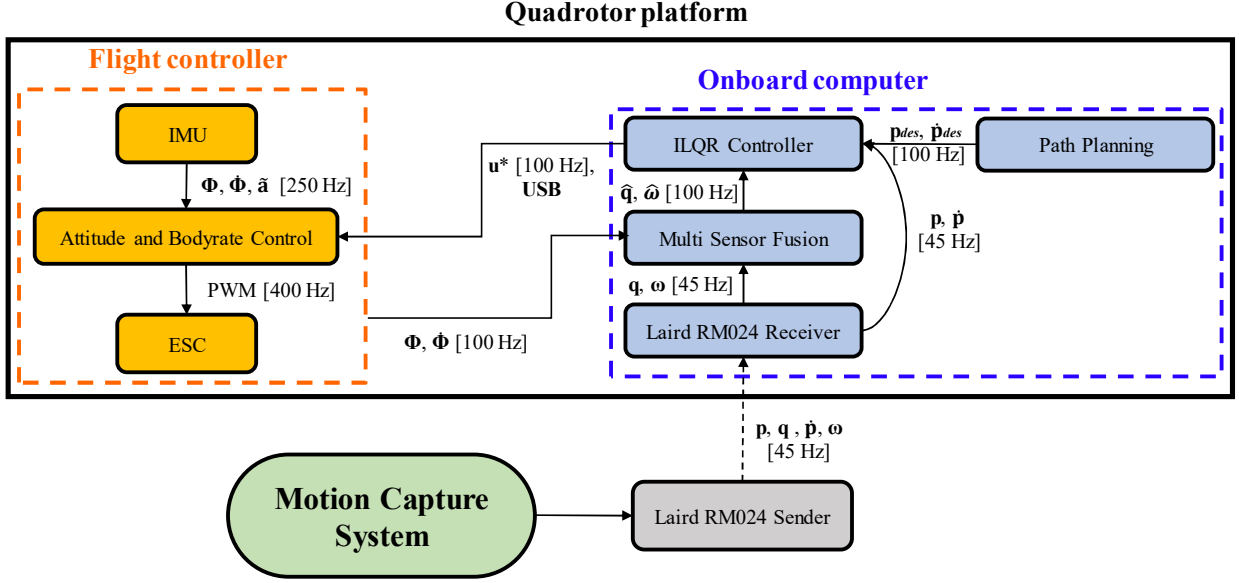


Figure 4.6. System architecture used for the experiment with estimated orientation $\hat{\mathbf{q}}$ and angular velocity $\hat{\boldsymbol{\omega}}$, and desired inputs \mathbf{u}^* . \mathbf{p} , \mathbf{q} , $\dot{\mathbf{p}}$ and $\boldsymbol{\omega}$ are position, orientation, linear velocity and angular velocity. \mathbf{p}_{des} and $\dot{\mathbf{p}}_{des}$ are the desired position and linear velocity provided by the trajectory generation algorithm. Φ , $\dot{\Phi}$, $\ddot{\mathbf{a}}$ represent orientation, body rates and linear acceleration measurements from the IMU unit.

4.3 Results

Real-time experiments have been conducted with the quadrotor platform presented in Section 4.2. In order to evaluate the performance of the approach, a human threw a table tennis ball towards the quadrotor during hover to simulate a normal exchange during a game. Then, the parabolic trajectory of the ball was estimated in real-time based on the initial position and velocity captured by the motion capture system cameras. The impact point is chosen to be the highest point in the z-axis after the bounce. This allows to compensate for position and estimate errors since the ball technically remains at the same x - z position during a longer period. Readers can be referred to the following video showing some experimental results with the quadrotor: https://www.youtube.com/watch?v=7d_N0LZcuzk&t=21s.

First, gains were tuned by a trial-and-error procedure for a offset of three trajectories which accurately represent the type of trajectories performed during a hitting motion with the quadrotor. Those trajectories are based on the position the quadrotor will have to reach and in what time

interval to be able to hit a real ball that will be thrown in the future. We find that accurate tuning was not necessary and good performance was already obtained with default gains. However, after some tuning the quadratic cost terms were set to $\mathbf{Q} = \text{diag}([1, 20, 1, 2.7, 0.5, 2.2, 0.5, 3.5, 0.5])$ and $\mathbf{R} = \text{diag}([0.1, 5, 10, 10, 5])$ while inputs for the impact were limited to $0.6 \text{ s} \leq t_{\text{impact}} \leq 1.2 \text{ s}$, $-0.5 \text{ m} \leq x_i \leq 0.34 \text{ m}$, $0.32 \text{ m} \leq y_i \leq 0.72 \text{ m}$ and $1.0 \text{ m} \leq z_i \leq 1.5 \text{ m}$ due to the lack of space and to avoid dangerous and unpredictable maneuvers.

We tested 20 hits and obtained a success rate of 40%. Note that we consider a return as a success only if the ball falls back on the opposite part of the table if not it's a miss. The details of these trials are shown in Tableau 4.1. We recorded the initial position in space captured by the motion capture cameras x_e, y_e, z_e and the estimated initial velocity of the ball vx_e, vy_e, vz_e and information on the estimated impact point. The trials are well diversified considering the available area since the time of impact has a range of 0.49 seconds between the shortest time to reach the ball and the case where the quadrotor has the longest time available. The position of impact has ranges of 0.75 m , 0.37 m and 0.28 m in x, y, z -directions respectively. This led to the quadrotor performing diversified aggressive trajectories.

Figure 4.7 depicts a typical trajectory performed by the quadrotor for a hitting motion. The actual trajectory tracking is shown in detail in Figure 4.8. In Figure 4.9, the velocities tracking is presented. Moreover, in Figure 4.10 the orientation of the quadrotor during the trajectory is shown. Finally, the controls inputs associated to the trajectory are presented in Figure 4.11. Details on aggressivity and root-mean-squared errors for x, y and z -directions are shown in Tableau 4.2.

Tableau 4.1 Table tennis hits performed for the quadrotor evaluation.

Trial	Result	t_{impact} (s)	x_i (m)	y_i (m)	z_i (m)	x_e (m)	y_e (m)	z_e (m)	vx_e (m/s)	vy_e (m/s)	vz_e (m/s)
1	Success	0.76	-0.31	0.38	1.39	1.77	0.55	1.68	-3.72	-0.21	0.54
2	Success	0.87	0.04	0.45	1.41	2.16	0.58	1.63	-3.33	-0.15	1.48
3	Success	0.70	-0.11	0.36	1.27	1.97	0.55	1.46	-4.01	-0.26	0.88
4	Miss	0.69	0.02	0.49	1.29	2.03	0.60	1.43	-3.87	-0.16	0.56
5	Miss	0.63	0.28	0.48	1.23	2.03	0.61	1.43	-3.53	-0.20	0.49
6	Miss	0.92	-0.43	0.47	1.37	2.27	0.62	1.40	-4.43	-0.15	2.28
7	Success	0.76	0.14	0.34	1.33	2.12	0.57	1.55	-3.45	-0.31	1.00
8	Success	0.79	-0.20	0.49	1.32	2.12	0.60	1.48	-4.12	-0.14	1.42
9	Success	0.77	-0.02	0.56	1.30	2.10	0.56	1.46	-3.76	-0.01	1.31
10	Miss	0.72	0.01	0.35	1.28	2.04	0.52	1.47	-3.75	-0.24	1.02
11	Miss	0.66	-0.02	0.53	1.22	2.12	0.61	1.39	-4.40	-0.12	0.89
12	Success	0.80	-0.13	0.34	1.33	2.12	0.55	1.49	-3.89	-0.26	1.45
13	Miss	0.75	-0.06	0.40	1.34	1.94	0.53	1.59	-3.53	-0.17	0.85
14	Miss	0.72	-0.07	0.45	1.35	1.87	0.58	1.61	-3.53	-0.17	0.52
15	Success	0.65	-0.11	0.34	1.31	1.68	0.50	1.57	-3.54	-0.24	0.04
16	Miss	0.63	-0.13	0.31	1.32	1.62	0.48	1.58	-3.52	-0.27	-0.17
17	Miss	1.00	-0.32	0.29	1.46	2.28	0.56	1.49	-3.86	-0.27	2.50
18	Miss	0.60	-0.47	0.50	1.18	1.26	0.57	1.34	-3.40	-0.12	0.69
19	Miss	0.52	-0.32	0.66	1.31	1.38	0.88	1.49	-3.84	-0.41	-1.17
20	Miss	0.93	-0.10	0.35	1.42	2.21	0.55	1.52	-3.49	-0.21	2.10

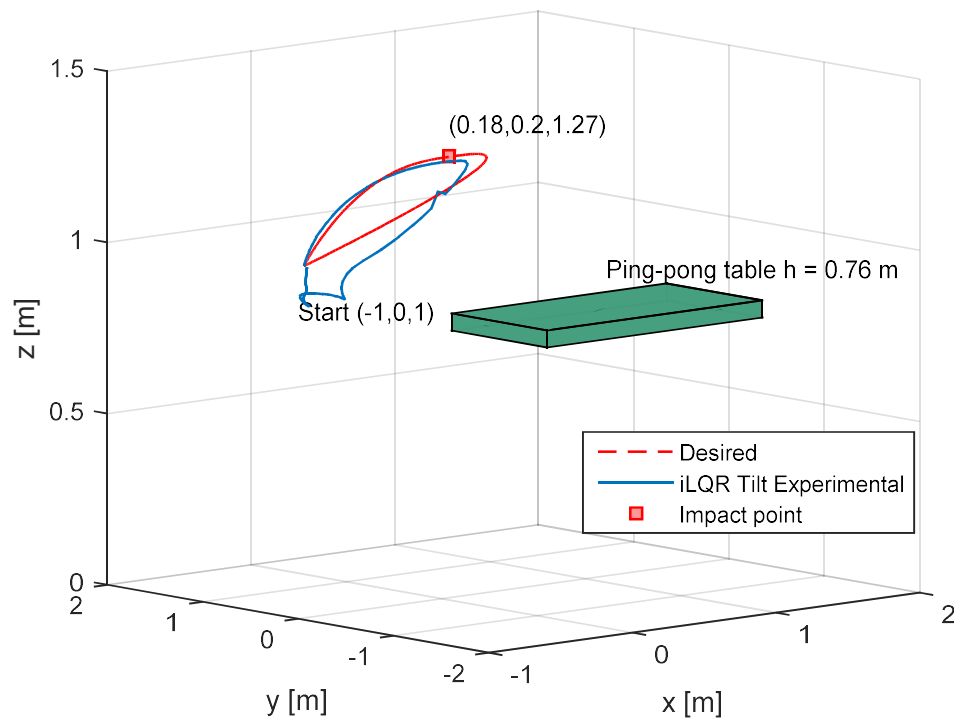


Figure 4.7. Desired (dashed red) and iLQR with tilt (solid blue) trajectories with impact point (red square) for an experimental ping pong hit.

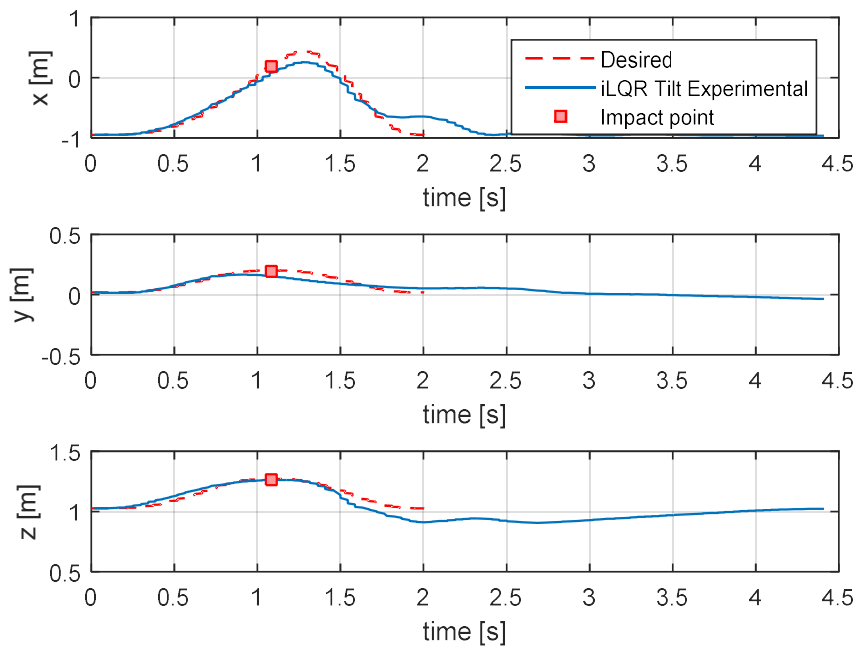


Figure 4.8. Desired (dashed red) and iLQR with tilt (solid blue) cartesian positions with impact point (red square) for an experimental ping pong hit.

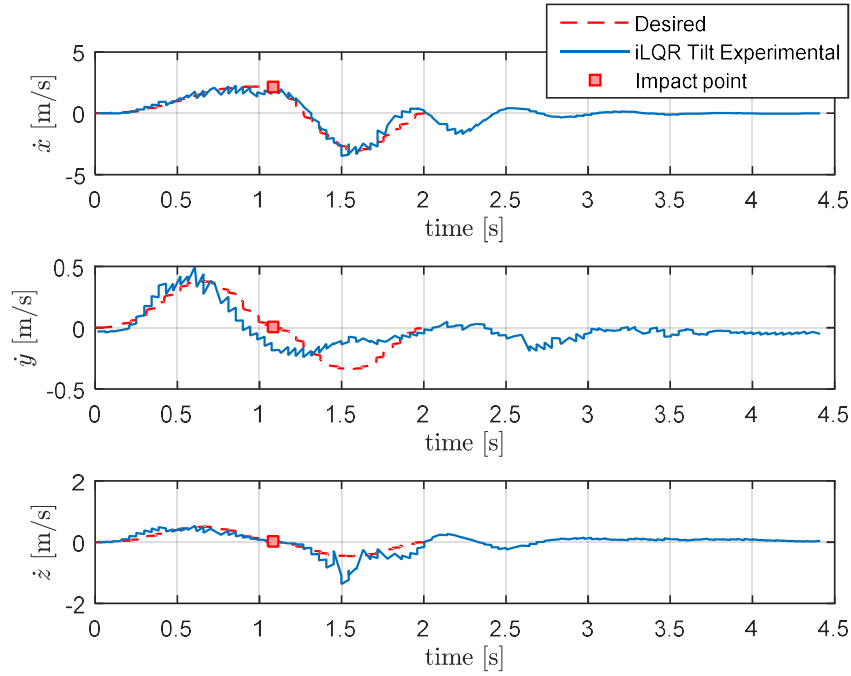


Figure 4.9. Desired (dashed red) and iLQR with tilt (solid blue) Cartesian velocities with impact point (red square) for an experimental ping pong hit.

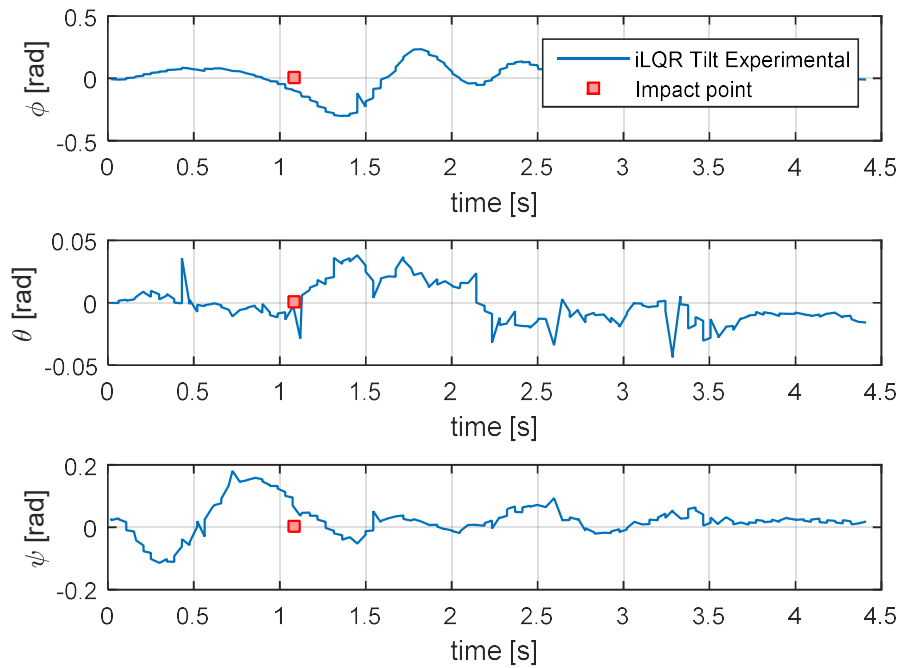


Figure 4.10. iLQR with tilt (solid blue) angles with impact point (red square) for an experimental ping pong hit.

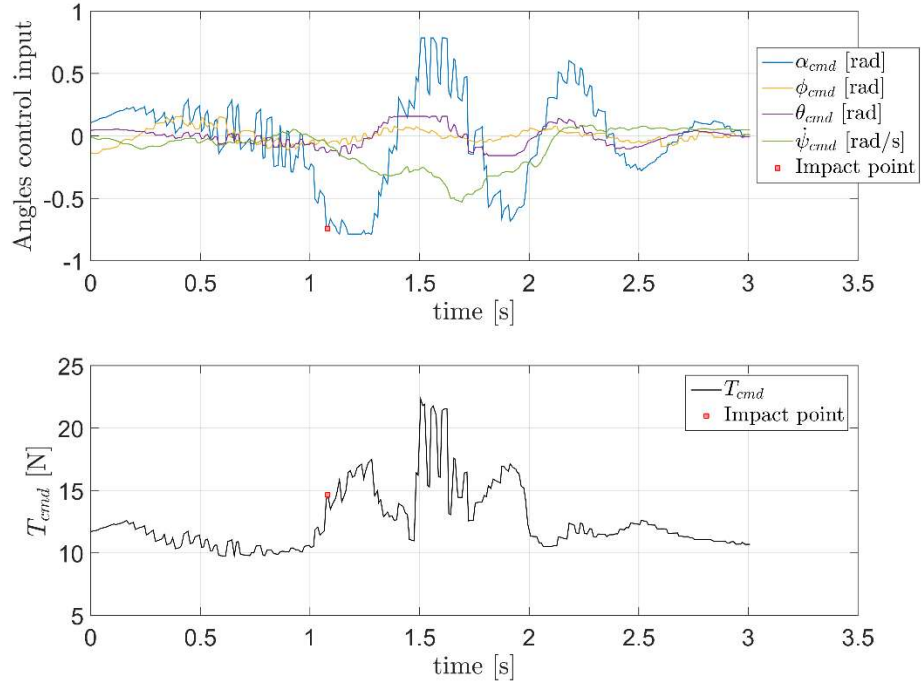


Figure 4.11 Control inputs with impact point (red square) for an experimental ping pong hit, where T_{cmd} , ϕ_{cmd} , θ_{cmd} , $\dot{\psi}_{cmd}$, α_{cmd} are the control inputs associated to U_1 , U_2 , U_3 , U_4 and U_5 respectively.

Tableau 4.2 Root-mean-squared errors and aggressivity for an experimental ping pong hit.

Max Acc. (m/s ²)	Max Vel. (m/s)	Max Pitch (deg)	Max Roll (deg)	Cartesian Component	RMSE position (m)	RMSE velocity (m/s)
7.67	3.49	2.51	17.26	x	0.120	0.414
				y	0.032	0.119
				z	0.044	0.232

4.4 Discussion

The main result of this paper is the success rate of the autonomous quadrotor playing table tennis which is 40%. This result is better compared to what was achieved previously with a quadrotor like in [8] where the hitting rate is only 20% and the quadrotor is only required to touch the ball with

the paddle instead of returning it. Contrarily, better results were achieved with serial manipulators as in [5] with a hitting rate of 88%.

The second main result is the high precision obtained on aggressive trajectory tracking with the proposed quadrotor having tilting propellers. Note that in Figure 4.7 the controller aggressivity was diminished for the returning phase of the trajectory to be more stable. This explains why the error on the desired trajectory is higher after the point of impact. In Figure 4.8, it can be observed that error is higher in the x-direction due to the response time of the servo motor employed for the tilting mechanism and because high aggressivity is required principally in that direction. In fact, Figure 4.11 shows that the tilting angle commanded hits the maximal threshold imposed (45 deg) two times in the trajectory. Figure 4.9 shows that due to the tilting mechanism on the quadrotor, high velocity is easily obtained in the x-direction. Moreover, we can observe in Figure 4.10 that the pitch angle during the trajectory, θ , remains below 0.05 radians (2.9 degrees), while achieving high accelerations and velocities in that direction. This is not possible for a conventional quadrotor. Root-mean-squared errors for x, y and z-directions in Tableau 4.2 show that although errors seem to be high, they are low enough to hit the ball in a fraction of second since the paddle used has a diameter bigger than 10 centimeters matching a conventional table tennis paddle. These errors are of 12.0 and 3.2cm in the x and y axis respectively. This is higher compared to aggressive trajectory tracking in [11], where errors of 1.3cm are obtained in both x and y axis.

The cost of our prototype (1500CAN\$ + motion capture system) is also significantly less than for a serial manipulator like the Barrett WAM used in [5] (135000USD\$ + motion capture system).

4.5 Conclusion

In this paper, we presented a novel type of table tennis quadrotor player with tilting propellers. Precise and aggressive trajectory tracking was achieved with an iLQR controller combined with minimum snap generated trajectories. Experiments have been conducted on a real quadrotor achieving hitting rate of 40%. Some limitations in this study includes the restricted rate of the motion capture system at 45Hz preventing from running the controller at 100Hz even if the hardware can run at that rate. Moreover, vibrations coming from the 3D printed plastic arms connections and the paddle not fixed rigidly to the quadrotor have a negative impact on the control achievable. In the future, it would be interesting to test the quadrotor on an official ping pong table

in an environment with more space and to program the platform to launch a new trajectory each time it sees a new ball coming and thus account for multiple exchanges against a human. Although the quadrotoir is capable of such behavior in terms of performances, this has not been implemented yet.

Appendix A: State space jacobians

The state space jacobians of the quadrotor systems are defined as:

$$\mathbf{A} = \begin{bmatrix} -\frac{1}{\tau_\varphi} & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & -\frac{1}{\tau_\varphi} & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 \\ A_{51} & A_{52} & A_{53} & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 \\ A_{71} & A_{72} & A_{73} & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 \\ A_{91} & 0 & A_{93} & 0 & 0 & 0 & 0 & 0 & 0 \end{bmatrix}, \quad (\text{A.1})$$

with

$$A_{51} = \frac{u_1}{m} (s(x_3)c(x_1) + s(x_1)s(x_2)c(x_3)), \quad (\text{A.2})$$

$$A_{52} = \frac{u_1}{m} (c(x_1)c(x_2)c(x_3) - s(x_2)c(x_3)\tan(u_5)), \quad (\text{A.3})$$

$$A_{53} = \frac{u_1}{m} (c(x_3)s(x_1) - c(x_1)s(x_2)s(x_3) - c(x_2)s(x_3)\tan(u_5)), \quad (\text{A.4})$$

$$A_{71} = \frac{u_1}{m} (-s(x_1)s(x_2)s(x_3) - c(x_1)c(x_3)), \quad (\text{A.5})$$

$$A_{72} = \frac{u_1}{m} (c(x_1)c(x_2)s(x_3) - s(x_2)s(x_3)\tan(u_5)), \quad (\text{A.6})$$

$$A_{73} = \frac{u_1}{m} (c(x_1)s(x_2)c(x_3) + s(x_1)s(x_3) + c(x_2)c(x_3)\tan(u_5)), \quad (\text{A.7})$$

$$A_{91} = \frac{u_1}{m} (-c(x_3)s(x_1)), \quad (\text{A.8})$$

$$A_{93} = \frac{u_1}{m} (-s(x_3)c(x_1) - c(x_3)\tan(u_5)), \quad (\text{A.9})$$

and

$$\mathbf{B} = \begin{bmatrix} 0 & \frac{k_\varphi}{\tau_\varphi} & 0 & 0 & 0 \\ 0 & 0 & \frac{k_\theta}{\tau_\theta} & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 0 \\ B_{51} & 0 & 0 & 0 & B_{55} \\ 0 & 0 & 0 & 0 & 0 \\ B_{71} & 0 & 0 & 0 & B_{75} \\ 0 & 0 & 0 & 0 & 0 \\ B_{91} & 0 & 0 & 0 & B_{95} \end{bmatrix}, \quad (\text{A.10})$$

with

$$B_{51} = \frac{1}{m} (s(x_3)s(x_1) + c(x_1)s(x_2)c(x_3) + c(x_2)c(x_3)\tan(u_5)), \quad (\text{A.11})$$

$$B_{55} = \frac{u_1}{m} (c(x_2)c(x_3)(1 + \tan(u_5)^2)), \quad (\text{A.12})$$

$$B_{71} = \frac{1}{m} (c(x_1)s(x_2)s(x_3) - s(x_1)c(x_3) + c(x_2)s(x_3)\tan(u_5)), \quad (\text{A.13})$$

$$B_{75} = \frac{u_1}{m} (c(x_2)s(x_3)(1 + \tan(u_5)^2)), \quad (\text{A.14})$$

$$B_{91} = \frac{1}{m} (c(x_3)c(x_1) - s(x_3)\tan(u_5)), \quad (\text{A.15})$$

$$B_{95} = \frac{u_1}{m} (-s(x_3)(1 + \tan(u_5)^2)). \quad (\text{A.16})$$

References

- [1] K. Mülling, J. Kober, J. Peters, Simulating human table tennis with a biomimetic robot setup, International Conference on Simulation of Adaptive Behavior, Springer, 2010, pp. 273-282.
- [2] L. Acosta, J. Rodrigo, J.A. Mendez, G.N. Marichal, M.J.I.r. Sigut, a. magazine, Ping-pong player prototype, IEEE robotics & automation magazine, 10 (2003) 44-52.
- [3] K. Mülling, J. Kober, J.J.A.B. Peters, A biomimetic approach to robot table tennis, Adaptive Behavior, 19 (2011) 359-376.
- [4] Z. Yu, Q. Huang, X. Chen, W. Zhang, J.J.A.i.M.E. Gao, Design of a redundant manipulator for playing table tennis towards human-like stroke patterns, Advances in Mechanical Engineering, 6 (2014) 807458.
- [5] K. Mülling, J. Kober, O. Kroemer, J.J.T.I.J.o.R.R. Peters, Learning to select and generalize striking movements in robot table tennis, The International Journal of Robotics Research, 32 (2013) 263-279.
- [6] S. Gomez-Gonzalez, Y. Nemmour, B. Schölkopf, J.J.R. Peters, Reliable Real-Time Ball Tracking for Robot Table Tennis, Robotics 8(2019) 90.
- [7] Omron, About forpheus. 2019. Accessed on the 12th of November 2019; <https://www.omron.com/innovation/forpheus.html>.
- [8] R. Silva, F.S. Melo, M. Veloso, Towards table tennis with a quadrotor autonomous learning robot and onboard vision, 2015 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS), IEEE, 2015, pp. 649-655.
- [9] W. Zhou, D. Nair, O. Gunawan, T. van Kessel, H.F. Hamann, A testing platform for on-drone computation, 2015 33rd IEEE International Conference on Computer Design (ICCD), IEEE, 2015, pp. 732-735.
- [10] W. Zhou, Drone-playing ping pong at IBM Research, 2016.
- [11] D. Mellinger, N. Michael, V. Kumar, Trajectory generation and control for precise aggressive maneuvers with quadrotors, The International Journal of Robotics Research,, 31 (2012) 664-674.
- [12] C. Richter, A. Bry, N. Roy, Polynomial trajectory planning for aggressive quadrotor flight in dense indoor environments, Robotics Research, Springer, 2016, pp. 649-666.

- [13] D. Mellinger, V. Kumar, Minimum snap trajectory generation and control for quadrotors, 2011 IEEE International Conference on Robotics and Automation, IEEE, 2011, pp. 2520-2525.
- [14] M. Faessler, D. Falanga, D.J.I.R. Scaramuzza, A. Letters, Thrust mixing, saturation, and body-rate control for accurate aggressive quadrotor flight, IEEE Robotics and Automation Letters, 2 (2016) 476-482.
- [15] M. Faessler, A. Franchi, D.J.I.R. Scaramuzza, A. Letters, Differential flatness of quadrotor dynamics subject to rotor drag for accurate tracking of high-speed trajectories, IEEE Robotics and Automation Letters, 3 (2017) 620-626.
- [16] P. Foehn, D. Scaramuzza, Onboard state dependent lqr for agile quadrotors, 2018 IEEE International Conference on Robotics and Automation (ICRA), IEEE, 2018, pp. 6566-6572.
- [17] M. Kamel, M. Burri, R.J.I.-P. Siegwart, Linear vs nonlinear mpc for trajectory tracking applied to rotary wing micro aerial vehicles, IFAC-PapersOnLine, 50 (2017) 3463-3469.
- [18] Li, W., & Todorov, E. (2004, August). Iterative linear quadratic regulator design for nonlinear biological movement systems. In ICINCO (1) (pp. 222-229).
- [19] W. Jasim, D.J.I.J.o.M. Gu, Identification, Control, Iterative linear quadratic regulator control for quadrotors leader-follower formation flight, International Journal of Modelling, Identification and Control, 31 (2019) 152-160.
- [20] J. van den Berg, Iterated LQR smoothing for locally-optimal feedback control of systems with non-linear dynamics and non-quadratic cost, 2014 American Control Conference, IEEE, 2014, pp. 1912-1918.
- [21] Y.N. Alothman, Optimal Control of Multiple Quadrotors for Transporting a Cable Suspended Payload, University of Essex, 2018.
- [22] W.M. Jasim, Improving Leader-Follower Formation Control Performance for Quadrotors, University of Essex, 2016.
- [23] Y. Tassa, T. Erez, E. Todorov, Synthesis and stabilization of complex behaviors through online trajectory optimization, 2012 IEEE/RSJ International Conference on Intelligent Robots and Systems, IEEE, 2012, pp. 4906-4913.

- [24] Pantoja, D.O.: Differential dynamic programming and newton's method. *International Journal of Control* 47(5), 1539–1553 (1988). DOI 10.1080/00207178808906114. URL <http://www.tandfonline.com/doi/abs/10.1080/00207178808906114>
- [25] Todorov, E., Li, W.: A generalized iterative LQG method for locally-optimal feedback control of constrained nonlinear stochastic systems. In: *Proceedings of the American Control Conference (ACC'05)*, pp. 300–306. Portland, OR, USA (2005). DOI 10.1109/ACC.2005.1469949
- [26] P. Geoffroy, N. Mansard, M. Raison, S. Achiche, E. Todorov, From inverse kinematics to optimal control, *Advances in Robot Kinematics*, Springer, 2014, pp. 409-418.
- [27] P. Geoffroy, O. Bordron, N. Mansard, M. Raison, O. Stasse, T. Bretl, A two-stage suboptimal approximation for variable compliance and torque control, *2014 European Control Conference (ECC)*, IEEE, 2014, pp. 1151-1157.
- [28] M. Cutler, N.-K. Ure, B. Michini, J. How, Comparison of fixed and variable pitch actuators for agile quadrotors, *AIAA Guidance, Navigation, and Control Conference*, 2011, pp. 6406.
- [29] M. Ryll, H.H. Bühlhoff, P.R. Giordano, First flight tests for a quadrotor UAV with tilting propellers, *2013 IEEE International Conference on Robotics and Automation*, IEEE, 2013, pp. 295-302.
- [30] A. Moutinho, E. Mateos, F. Cunha, The tilt-quadrotor: concept, modeling and identification, *2015 IEEE International Conference on Autonomous Robot Systems and Competitions*, IEEE, 2015, pp. 156-161.
- [31] M. Ryll, H.H. Bühlhoff, P.R.J.I.T.o.C.S.T. Giordano, A novel overactuated quadrotor unmanned aerial vehicle: Modeling, control, and experimental validation, *Transactions on Control Systems Technology*, 23 (2014) 540-556.
- [32] A. Alkamachi, E.J.F.U. Ercelebi, *Series: Mechanical Engineering, Modelling and control of H-shaped racing quadcopter with tilting propellers*, *Facta Universitatis, Series: Mechanical Engineering*, 15 (2017) 201-215.
- [33] J.-P. Burke, *Preliminary Design of Robust Adaptive Control Laws for an Aerial Manipulator*, 2019.

- [34] H. Fernando, A. De Silva, M. De Zoysa, K. Dilshan, S. Munasinghe, Modelling, simulation and implementation of a quadrotor UAV, 2013 IEEE 8th International Conference on Industrial and Information Systems, IEEE, 2013, pp. 207-212.
- [35] Y. Tassa, N. Mansard, E. Todorov, Control-limited differential dynamic programming, 2014 IEEE International Conference on Robotics and Automation (ICRA), IEEE, 2014, pp. 1168-1175.
- [36] L. Meier, D. Honegger, M. Pollefeys, PX4: A node-based multithreaded open source robotics framework for deeply embedded platforms, 2015 IEEE international conference on robotics and automation (ICRA), IEEE, 2015, pp. 6235-6240.
- [37] UP Board. 2020. Accessed on the 7th of February 2020; <https://up-board.org/>.
- [38] Ubuntu. 2020. Accessed on the 7th of February 2020; <https://ubuntu.com/>.
- [39] ROS. 2020. Accessed on the 7th of February 2020; <https://www.ros.org/>.
- [40] BTS Bioengineering. 2020. Accessed on the 7th of February 2020; <https://www.btsbioengineering.com/>.

CHAPITRE 5 DÉTAILS SUPPLÉMENTAIRES SUR LA MÉTHODOLOGIE ET LES RÉSULTATS

L'article présenté au Chapitre 3 abordait de manière concise la conception mécanique du quadrirotor développé dans cette recherche ainsi que le développement du contrôleur et les résultats expérimentaux avec le prototype réelle. Ce chapitre présente des résultats supplémentaires obtenues auparavant en simulation sur une table de ping-pong grandeur réglementaire afin de montrer que le quadrirotor a été désigné au préalable pour faire des vrais échanges contre un humain. De plus, d'autres détails manquants dans l'article par rapport à la méthodologie y sont présentés.

5.1 Résultats en simulation

L'article ne présentait pas les résultats obtenus au préalable en simulation. Nous utilisons le simulateur V-REP développée par Coppelia Robotics [39] étant l'outil parfait pour le prototypage rapide de nouveaux algorithmes. Pour évaluer la performance du contrôleur, nous avons donc tiré des balles depuis l'autre côté de la table de sorte que la trajectoire ressemble le plus possible à des frappes normalement effectuées par un adversaire humain. Les balles sont tirées aléatoirement afin d'atterrir un peu partout sur le terrain adverse pour avoir suffisamment de variété. Le taux de succès du retour des balles sur le terrain adverse est de 80%. La Figure 5.1 montre l'environnement utilisé.

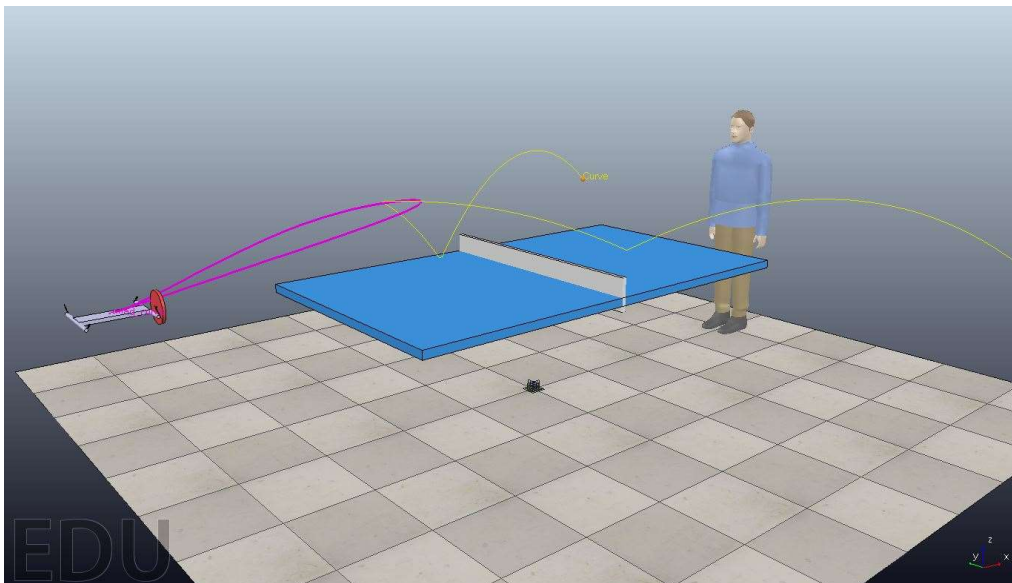


Figure 5.1 Environnement de simulation pour le quadrirotor joueur de ping-pong sur V-REP.

La Figure 5.2 illustre une trajectoire de frappe de balle typique effectuée par le quadrirotor en simulation.

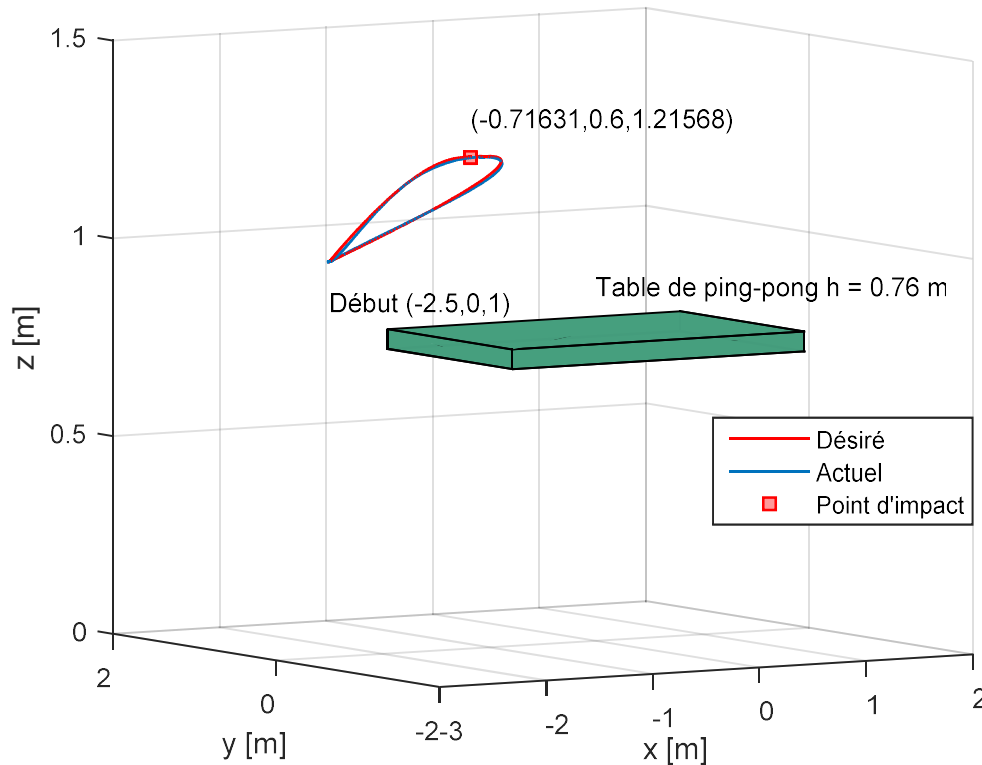


Figure 5.2 Trajectoire désirée (rouge) et actuelle (bleu) avec le point d'impact estimé (carrée rouge) pour un mouvement de frappe typique de ping-pong en simulation.

L'erreur par rapport à la trajectoire désirée est très faible puisque le modèle dynamique est parfait en simulation. On peut voir dans la Figure 5.3 plus en détails que la trajectoire désirée est effectivement très bien respectée dans les directions x, y et z. De plus, en ce qui concerne les vitesses désirées, on constate dans la Figure 5.4 qu'elles sont légèrement moins bien respectées, mais ce sont tout de même des performances exceptionnelles dû à la puissance du contrôleur utilisé. Les orientations du quadrirotor pendant l'exécution de cette trajectoire sont présentées dans la Figure 5.5. L'angle en tangage θ restent en dessous de 10 degrés (0,17 radians) comme prévu pour ce type de quadrirotor avec des bras inclinables. Ceci permet d'augmenter l'agressivité du système puisque le temps requis pour générer une poussée vers l'avant est dicté par le temps de réponse du servomoteur. Dans le cas, d'un quadrirotor conventionnel, cela dépend du temps requis pour l'incliner,

qui est normalement plus élevé. Ensuite, la commande effectuée durant la trajectoire est présentée à la Figure 5.6. Le Tableau 5.1 présente l'erreur quadratique moyenne pour cette trajectoire.

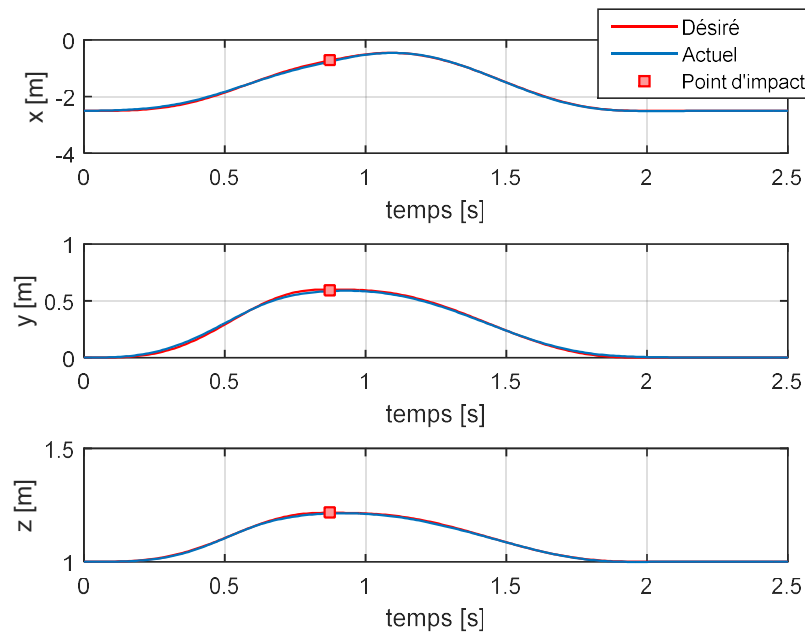


Figure 5.3 Positions cartésiennes désirées (rouge) et actuelles (bleu) ainsi que le point d'impact estimé (carrée rouge) pour un mouvement de frappe typique de ping-pong en simulation.

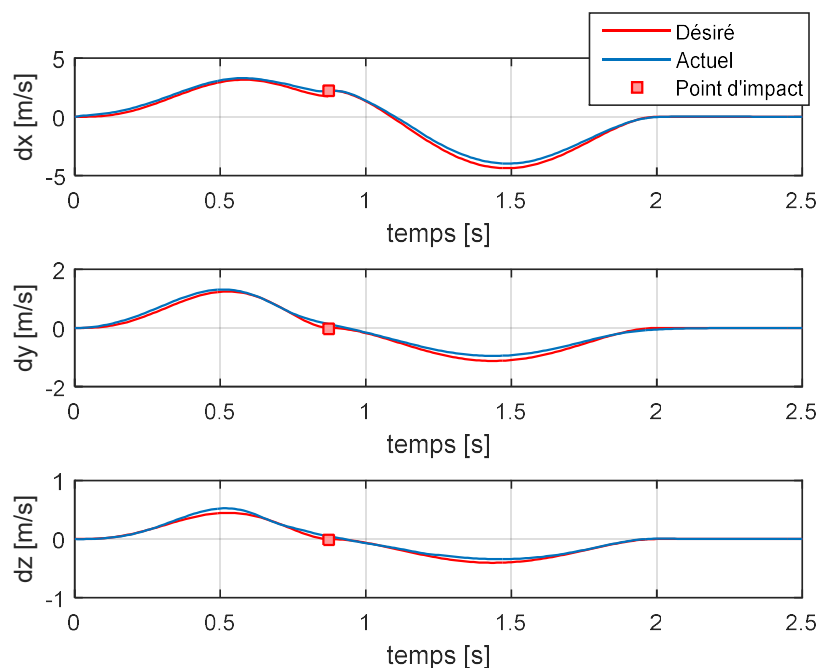


Figure 5.4 Vitesses cartésiennes désirées (rouge) et actuelles (bleu) ainsi que le point d'impact estimé (carrée rouge) pour un mouvement de frappe typique de ping-pong en simulation.

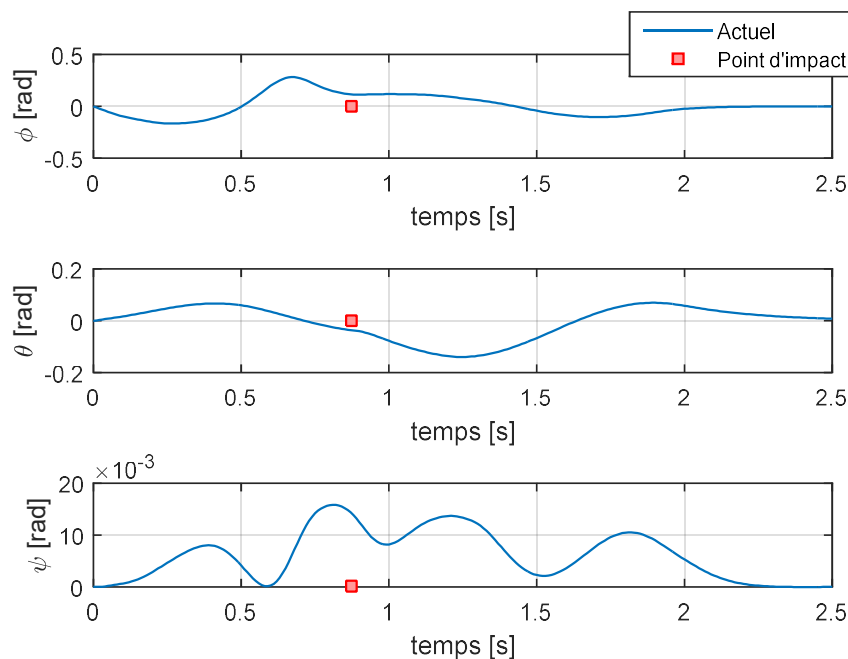


Figure 5.5 Orientations actuelles performées (bleu) ainsi que le point d'impact estimé (carrée rouge) pour un mouvement de frappe typique de ping-pong en simulation.

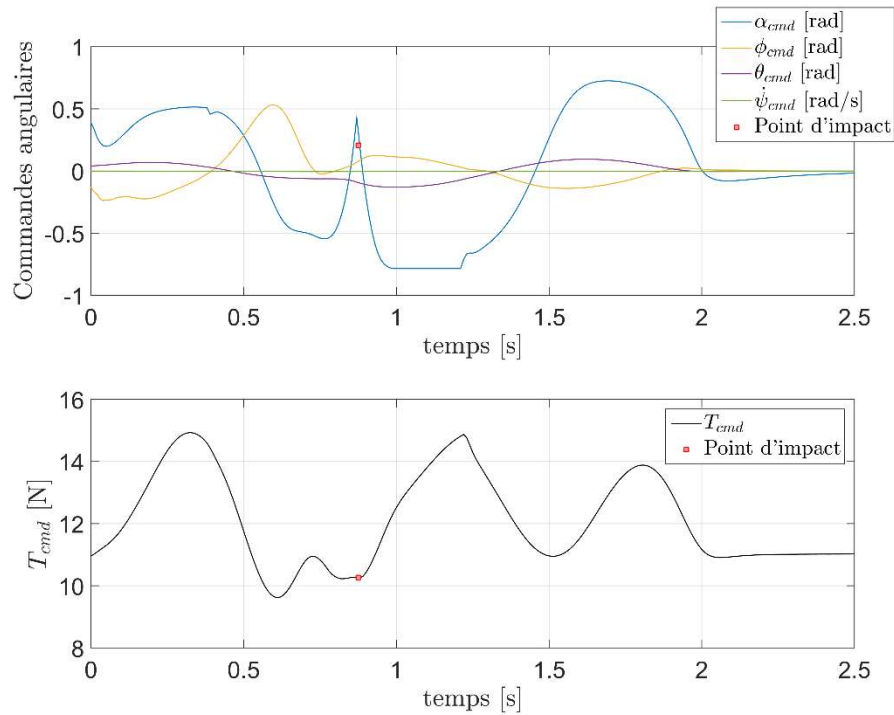


Figure 5.6 Commandes envoyées au quadrirotor ainsi que le point d'impact estimé (carrée rouge) pour un mouvement de frappe typique de ping-pong en simulation.

Tableau 5.1 Erreur quadratique moyenne (EQM) et agressivité pour une frappe typique de ping-pong en simulation.

Acc. max. (m/s ²)	Vitesse max. (m/s)	θ max. (deg)	ϕ max. (deg)	Composante cartésienne	EQM position (m)	EQM vitesse (m/s)	Err. absolue de position au pt. d'impact (m)	Err. absolue de vitesse au pt. d'impact (m/s)
17.09	4.23	8.01	16.12	x	0.021	0.225	0.038	0.050
				y	0.010	0.088	0.013	0.115
				z	0.003	0.033	0.003	0.041

5.2 Estimation de la trajectoire de la balle

Pour pouvoir frapper la balle avec le quadrirotor, il faut tout d'abord estimer précisément le point d'impact. Celui-ci est choisi comme étant la position de la hauteur maximale après le premier rebond. Les données qui nous intéressent sont le temps au point d'impact t_3 , ainsi que les positions

de la balle à ce temps donné x_3, y_3, z_3 selon les directions x, y et z respectivement. Pour ce faire, nous avons besoin des estimations initiales suivantes $x_1, y_1, z_1, v_{x1}, v_{y1}, v_{z1}$ et t_1 qui sont respectivement les positions et les vitesses initiales de la balle dans les directions x, y et z estimées par le système de capture de mouvement ainsi que le temps de capture initial. La Figure 5.7 montre la trajectoire parabolique prévue.

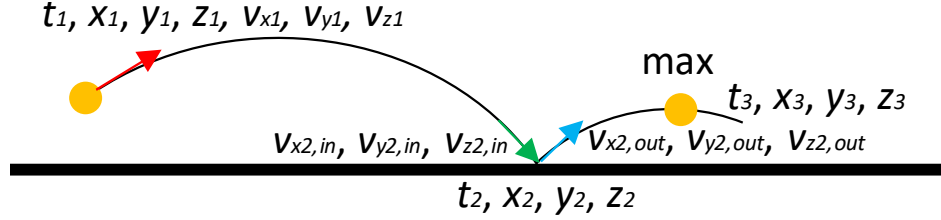


Figure 5.7 Trajectoire parabolique de la balle de ping-pong.

Le modèle dynamique simplifié de la balle de ping-pong est le suivant :

$$ma_x = -\frac{1}{2}C_D\rho_{air}Av_x^2, \quad (5.1)$$

$$ma_z = -mg, \quad (5.2)$$

où $g = -9,81\text{m/s}^2$ est l'accélération gravitationnelle, m la masse de la balle, C_D le coefficient de traînée de la balle, ρ_{air} la masse volumique de l'air et A l'aire de la balle. a_x et a_z sont les accélérations de la balle dans les directions x et z respectivement. Les mêmes équations que pour la direction x sont valables pour la direction y . Notons que nous considérons les autres forces qui agissent dans la direction x comme étant négligeables par rapport à la force de traînée. La force dominante dans la direction z est également considérée comme étant la gravité pour simplifier le problème.

Pour obtenir les informations du point d'impact, on utilise d'abord les équations cinématiques suivantes provenant de la définition de l'accélération dans la direction z :

$$v_{z2,in}^2 = v_{z1}^2 + 2g\Delta z, \quad (5.3)$$

$$v_{z2,in} = v_{z1} + gt_1, \quad (5.4)$$

où $\Delta z = z_2 - z_1$. En tenant compte des deux signes possibles de la racine carrée et en combinant les équations 5.3 et 5.4, la stratégie consiste à trouver d'abord le temps du premier rebond t_1 avec la relation suivante :

$$t_1 = \max \left(\frac{-v_{z1} + \sqrt{v_{z1}^2 + 2g\Delta z}}{g}, \frac{-v_{z1} - \sqrt{v_{z1}^2 + 2g\Delta z}}{g} \right). \quad (5.5)$$

Ensuite, la loi de la conservation de l'énergie est utilisée pour trouver la vitesse de la balle dans la direction z après le rebond [40] :

$$v_{z2,out} = K v_{z2,in}, \quad (5.6)$$

où K est le coefficient de restitution de la balle trouvé expérimentalement par un processus d'essai-erreur. En utilisant toujours les équations cinématiques on peut trouver les informations manquantes suivantes :

$$t_3 = \frac{-v_{z2,out}}{g}, \quad (5.7)$$

$$z_3 = z_2 + v_{z2,out} t_3 + \frac{1}{2} g t_3^2. \quad (5.8)$$

À partir de l'équation 4.1, le modèle cinématique continue peut être présenté comme suit [40] :

$$v_{x3} = v_{x1} \exp \left(-\frac{t}{2m} C_D \rho_{air} A v_{z2,out} t_3 v_{x1} \right). \quad (5.9)$$

Finalement, on obtient la position du point d'impact dans la direction x à partir de la relation suivante :

$$x_3 = x_1 + v_{x3} (t_3 - t_1). \quad (5.10)$$

La même démarche s'applique à la direction y . Notons que la friction est considérée comme étant négligeable dans les directions x et y , ce qui enlève la nécessité d'ajouter un coefficient de restitution dans ces directions également. Le coefficient de traînée est quant à lui aussi trouvé de manière expérimentale par un processus d'essai-erreur. La Figure 5.8 montre la balle utilisée.



Figure 5.8 Balle de ping-pong utilisée pour les tests expérimentaux. Des morceaux carrés de ruban réfléchissant sont placés sur la balle pour la capter par le système de capture de mouvements.

5.3 Conception mécanique

L'article ne présentait que des détails préliminaires concernant la conception mécanique du quadrirotor. Cette section vise à montrer au lecteur toute l'information nécessaire pour qu'il soit en mesure de comprendre le produit dans son intégralité.

5.3.1 Choix du mécanisme

Pour alléger le quadrirotor le plus possible, il a fallu concevoir un mécanisme qui soit le plus léger possible. Pour cette raison, nous avons décidé de permettre l'inclinaison des hélices dans un seul axe, soit celui où la plus grande agressivité est requise. Cela nous permet de n'utiliser qu'un seul servo-moteur par exemple. De plus, l'utilisation de la poulie pour la transmission du mouvement a paru être la solution la moins encombrante puisqu'elle permet de faire passer la transmission d'un côté à l'autre du quadrirotor en utilisant le moins d'espace et de pièce. Cela permet également de garder le mécanisme à l'intérieur du quadrirotor et ainsi de la protéger contre les chocs.

5.3.2 Mécanisme d'inclinaison des hélices

Tout le prototype du quadrirotor a été conçu tout d'abord en pensant à l'intégration d'un mécanisme d'inclinaison pour les hélices. La Figure 5.9 illustre ce mécanisme au niveau du servo-moteur.

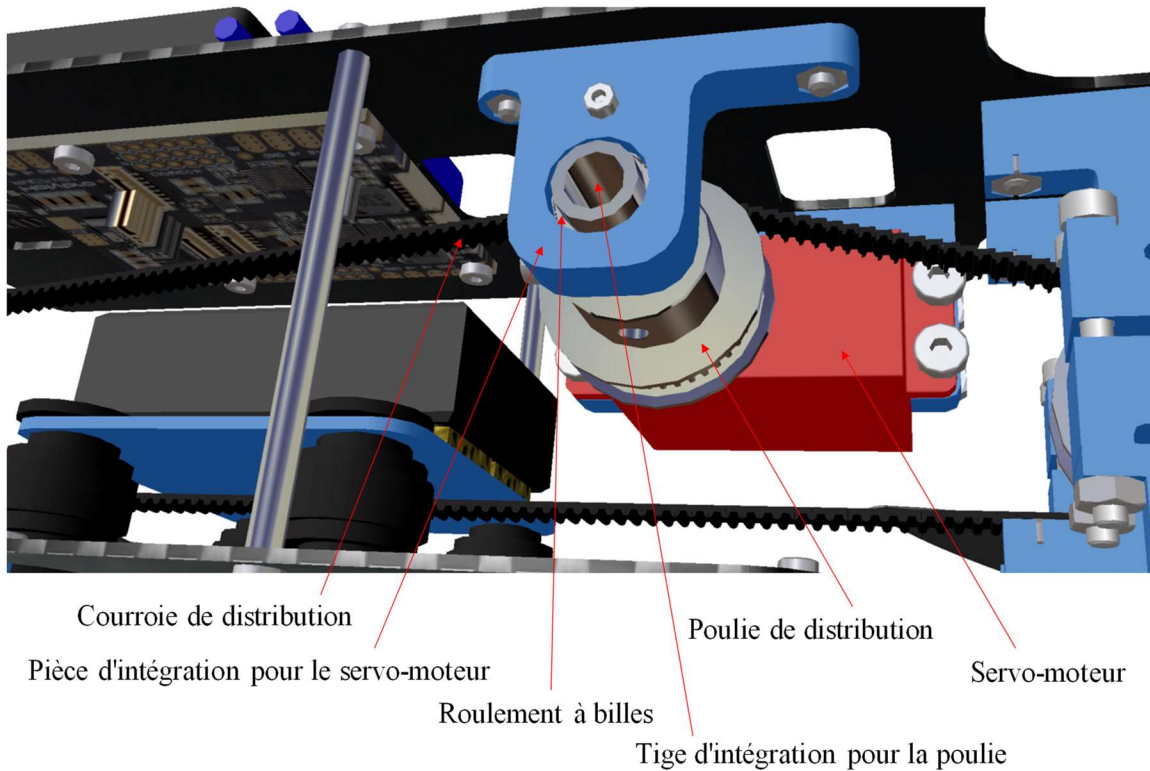


Figure 5.9 Mécanisme d'inclinaison au niveau du servo-moteur.

La position verticale de la poulie de distribution permet de tensionner suffisamment la courroie. Notons toutefois que pour augmenter l'adhérence de la courroie par rapport à la poulie, la courroie fait tout le tour de la poulie et revient au même endroit. Cette modification était nécessaire en pratique puisque l'adhérence est critique pour les performances du contrôleur. De plus, une bonne adhérence est nécessaire dû aux mouvements agressifs employés par le servo-moteur. Un roulement à billes est placé entre la pièce d'intégration pour le servo-moteur et la tige pour réduire la friction. Un principe de courroie et de poulie de distribution est également employé au niveau de la rotation des bras. La Figure 5.10 montre ce système au niveau des bras. La hauteur de la pièce de support pour le bras permet l'ajustement de la position de l'hélice pour la placer dans l'axe de rotation du bras. La Figure 5.11 montre une vue d'ensemble du mécanisme d'inclinaison du quadrirotor. Le parcours de la courroie est judicieusement choisi afin d'éviter le contact avec les autres pièces électroniques, ce qui est un véritable défi dû à l'espace restreint qui est disponible.

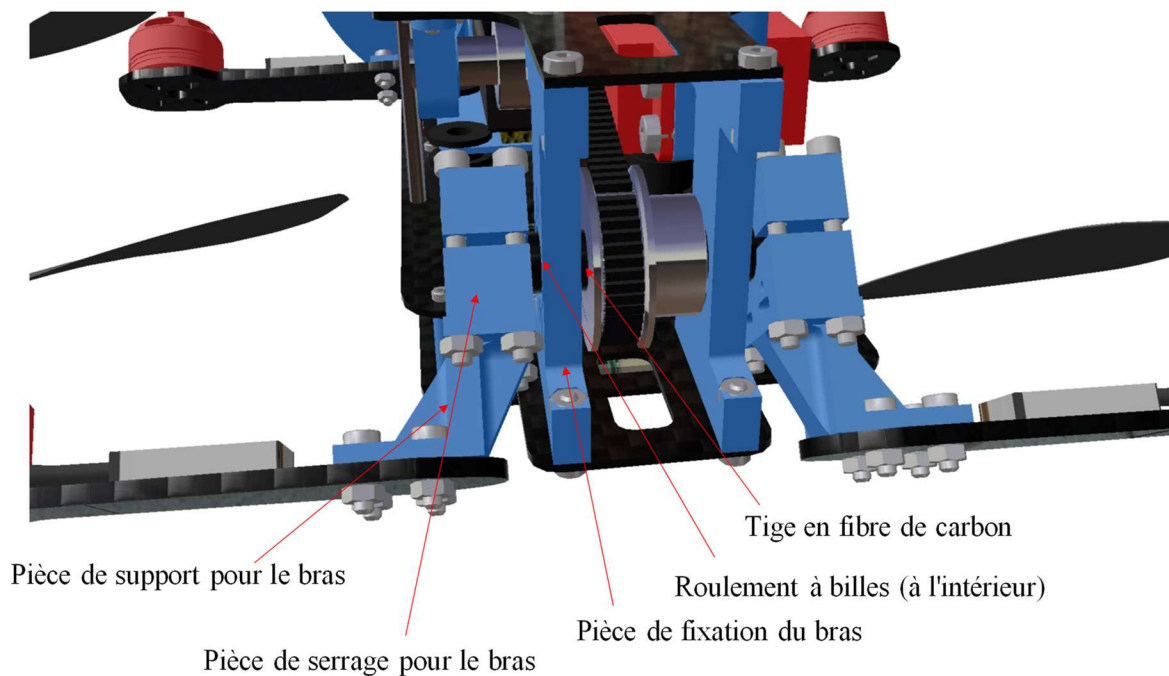


Figure 5.10 Mécanisme d'inclinaison au niveau du bras.

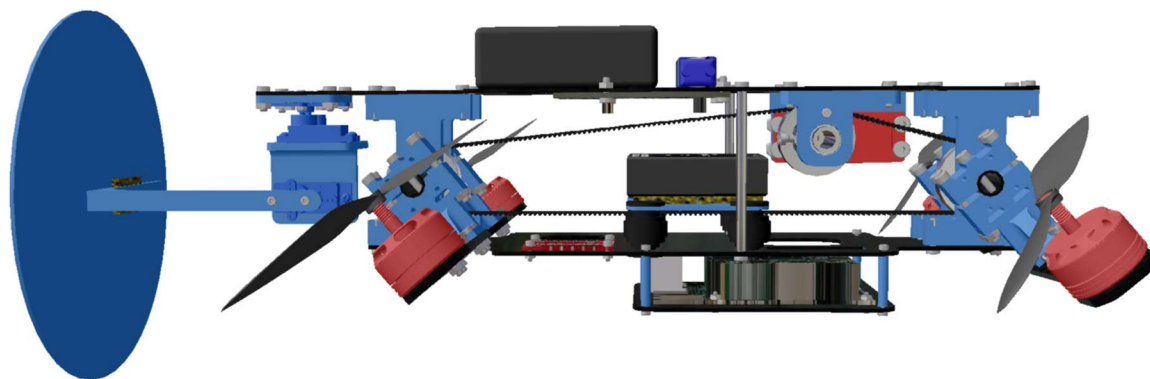


Figure 5.11 Vue d'ensemble du mécanisme d'inclinaison du quadrirotor.

5.3.3 Mécanisme de la gimbal

Les résultats obtenus en simulation nous permettaient d'en déduire que l'utilisation d'une gimbal facilitait grandement le renvoi de balle. Cela permet notamment de renvoyer la balle avec une meilleure précision sur le terrain adverse et également de simplifier les trajectoires agressives que

le quadrirotor doit accomplir. La même gimbal utilisée en simulation a donc été aussi conçu en réel permettant le mouvement en lacet et en tangage. Deux servo-moteurs y sont associés. La Figure 5.12 illustre la conception de la gimbal.

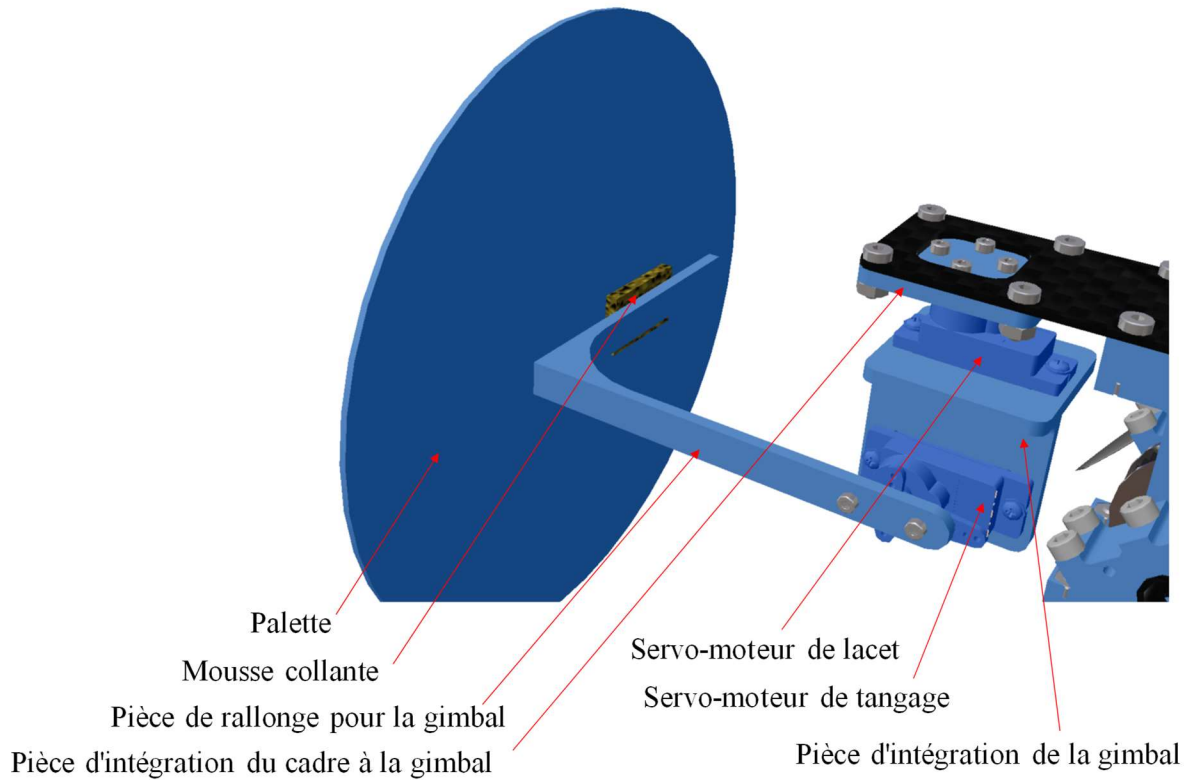


Figure 5.12 Mécanisme de la gimbal.

5.4 Choix des gains

Dans la formulation du contrôleur iLQR, il y a nécessité de définir 9 gains associés aux états du quadrirotor et 5 gains pour les commandes. Par exemple, nous avons utilisé les gains suivants : $\mathbf{Q} = \text{diag}([1, 20, 1, 2.7, 0.5, 2.2, 0.5, 3.5, 0.5])$ et $\mathbf{R} = \text{diag}([0.1, 5, 10, 10, 5])$. Nous allons donc essayer d'expliquer ici brièvement le processus du choix des gains. Toutefois, dû à la nature itérative du contrôleur, celui-ci n'a pas besoin d'un réglage intensif, et de bonnes performances peuvent déjà être obtenues avec des gains par défaut.

Tout d'abord, on met tous les gains à 1. Ensuite, les gains dans la matrice **R** permettent de donner plus ou moins de libertés à l'effort de contrôle. Une valeur élevée signifie qu'on désire minimiser l'effort de contrôle sur cette commande. Donc, si on veut être moins agressive pour être en mesure d'être stable, on va vouloir augmenter cette valeur. En ce qui concerne les gains dans la matrice **Q**, il faut augmenter la valeur si on veut prioriser le suivi pour cet état de la valeur désirée. Il n'y a pas d'ordre dans lequel il faut régler les gains, et on les ajuste en observant les résultats graphiquement sur une même trajectoire de référence.

CHAPITRE 6 DISCUSSION GÉNÉRALE

Contrôler un quadrirotor pour obtenir des performances comparables à des bras robotiques au jeu de tennis de table est un défi de taille. Néanmoins, dans ce mémoire, nous avons conçu et contrôlé un quadrirotor étant capable de retourner des balles lancées par un humain. Les performances démontrées sont également supérieures à ce qu'on peut retrouver dans la littérature en ce qui concerne les robots aériens. Dans ce chapitre, on revient tout d'abord sur les résultats présentés dans l'article au Chapitre 4 sur le prototype réel. Nous présentons ensuite une discussion additionnelle par rapport à ce qui n'était pas dans l'article. Puis, on aborde les limites et les perspectives de ce projet.

6.1 Résumé de la discussion de l'article

Il a été montré dans l'article au Chapitre 4 que notre prototype de quadrirotor joueur de tennis de table est capable de retourner 40% des balles lancées par un humain. Ce résultat est obtenu à partir d'un total de 20 lancés. Un succès est seulement considéré comme tel si la balle retombe sur le terrain adverse. Ces résultats sont plus élevés comparativement aux autres quadrirotors présents dans la littérature [11], [12].

Sur une trajectoire typique de frappe de balle, l'erreur est la plus élevée dans la direction x . Nous supposons que cela est dû au temps de réponse du servo-moteur responsable de l'inclinaison des bras qui doit fonctionner à une fréquence de 100 Hz. Si des différences d'angles trop brusques sont calculées par le contrôleur, celui-ci peut ne pas parfaitement respecter l'angle désiré. De plus, l'agressivité est la plus élevée dans cette direction, ce qui nuit certainement à la précision du suivi de trajectoire en x . Dû à la présence d'un mécanisme d'inclinaison des hélices, une vitesse élevée est facilement obtainable dans la direction x avec le prototype réel. Durant notre trajectoire typique, l'angle de tangage du quadrirotor reste en dessous de 0.05 radians (2.9 degrés) tout en ayant une vitesse et une accélération importante. Ceci permet d'avoir un comportement plus stable et sécuritaire tout en poussant l'agressivité du système à son maximum. L'accélération et la vitesse maximale durant la trajectoire sont de 7.67 m/s^2 et 3.49 m/s respectivement.

6.2 Discussion additionnelle

La discussion de l'article présentée au Chapitre 4 n'abordait que les résultats obtenus avec le prototype réel du quadrirotor. Cette section vise à donner une discussion supplémentaire en ce qui concerne notamment la comparaison des résultats entre la simulation et le réel, l'agressivité obtenue et la conception mécanique.

6.2.1 Comparaison des résultats entre la simulation et le réel

Tout d'abord, les tests expérimentaux ont établi le succès de l'approche, atteignant des taux de renvoi de balle de 80% et 40% en simulation et sur le quadrirotor réel respectivement. Les erreurs en position et en vitesse en simulation sont largement moindres que sur le modèle réel. Par exemple, l'erreur quadratique moyenne dans la direction x , n'est que de 2 centimètres en simulation comparativement à 12 centimètres avec le prototype réel. Nous attribuons cela à des différences de modélisation entre le système en simulation et le système réel. De plus, la traînée induite par la palette de ping-pong qui n'a pas été inclus sur le modèle en simulation peut réduire les performances de suivi de trajectoire dans la direction x notamment.

D'autres facteurs peuvent expliquer la baisse du taux de renvoi de balle en réel. Par exemple, dû à des limitations provenant du système de capture de mouvement, la boucle externe de contrôle ne peut pas dépasser 45 Hz. En simulation, les calculs du contrôleur iLQR peuvent rouler jusqu'à 100 Hz, ce qui est plus raisonnable comme vitesse de contrôle pour avoir de bonnes performances.

Ensuite, l'estimation de la trajectoire de la balle de ping-pong est quasi-parfaite en simulation et par le fait même l'est également le point d'impact. Il n'y a que quelques singularités provenant du simulateur lui-même qui nous empêche de simuler parfaitement le modèle de la balle. En revanche, la technique d'estimation de la trajectoire de la balle en réel est triviale et les erreurs sont beaucoup plus importantes.

6.2.2 Agressivité du quadrirotor

Il est difficile de comparer l'agressivité du quadrirotor à celles des bras robotiques. Visuellement, le quadrirotor conçu a le même niveau d'agressivité que les bras robotiques présentés dans la revue de littérature. De plus, ils sont tous les deux capables de remplir une fonction qui demande de l'agressivité en générale, soit de jouer au tennis de table. Les quadrirotors considérés comme

agressifs dans la littérature atteignent des vitesses de 2 m/s sur des trajectoires courtes et non répétitives [16] et nous atteignons 3.49 m/s avec le prototype réel.

6.2.3 Conception mécanique

La structure mécanique du quadrirotor est un aspect important du système global puisque sa conception a des effets importants sur les performances obtenues. La structure doit d'abord être la plus rigide possible pour éviter les vibrations qui nuisent à la performance du contrôleur. Un cadre ainsi que des bras rigides en fibre de carbone sont donc choisis à cet effet. Le quadrirotor se doit également d'être le plus compact et petit possible, ce qui veut dire que le bras de levier doit être réduit au maximum, et ce, afin de le rendre le plus agile possible. Les pièces d'intégration 3D sont également conçus de manière à être rigide afin de réduire les vibrations.

Le quadrirotor a été conçu en forme de H. Cela semblait être la solution la plus adaptée pour jouer au tennis de table, car on peut facilement positionner la palette et la gimbal à l'avant du drone en la reliant au cadre sans que celle-ci interfère avec les hélices.

Le mécanisme d'inclinaison est jumelé pour les deux bras, ce qui veut dire que l'angle d'inclinaison est le même à l'avant et à l'arrière. Ce choix de conception a été fait dans un optique de réduction maximale du poids puisque de cette manière, nous n'avons besoin que d'un seul servo-moteur. Notons que la plupart des quadrirotors modifiés dans la littérature sont plus lourds que les quadrirotors traditionnels. De plus, ils ont rarement été testés sur des trajectoires agressives.

L'hélice du prototype réel est placée au niveau de l'axe de rotation du bras pour représenter fidèlement le système dynamique modélisé pour lequel le contrôleur a été conçu. Cela est une modification importante par rapport aux autres quadrirotors, en forme de H plus particulièrement, présents dans la littérature. Habituellement, la différence entre l'axe de rotation et l'hélice est considérée comme une approximation par rapport au modèle dynamique utilisé [38]. Néanmoins, la hauteur du moteur est conséquente, ce qui fait que l'approximation n'est pas négligeable et peut nuire aux performances.

6.2.4 Effets des perturbations sur le iLQR

La comparaison entre les résultats obtenus en simulation et avec le prototype réel en laboratoire permet de bien observer les effets des perturbations externes et des incertitudes au niveau de la

modélisation sur le iLQR. L'erreur sur la trajectoire désirée est plus grande puisqu'on passe par exemple d'une erreur quadratique moyenne sur la position en x de 2.1 cm en simulation à 12 cm avec le prototype. Ensuite, puisqu'on a exécuté le contrôleur avec succès sur un prototype réel en laboratoire soumis à des perturbations, cela montre que le iLQR peut tenir compte des perturbations tout en gardant la stabilité du système.

6.2.5 Bénéfices de l'utilisation du iLQR par rapport au LQR

Le iLQR et le LQR ne résolvent pas le même problème de contrôle optimal. Le iLQR optimise les commandes sur une partie d'une trajectoire alors que le LQR les optimise par rapport à l'erreur actuelle sur la trajectoire. Ainsi, la performance en termes d'optimisation sur toute la trajectoire n'est pas comparable. Les contrôles optimaux trouvés par le iLQR sont très différents par rapport au LQR pour une même trajectoire et ils anticipent les changements de direction pour éviter des dépassements.

Une étude a montré que le iLQR avait une erreur quadratique moyenne sur une trajectoire agressive inférieure à un LQR. [43] De plus, l'ajout de rotors inclinables permet également de réduire cette erreur lors du suivi de trajectoire agressive avec un quadrirotor. Les efforts de contrôles sont similaires ce qui montre que l'utilisation de rotors inclinables et d'un iLQR permet d'obtenir de réduire l'erreur en suivi de trajectoire agressive sans augmenter les requis en termes de consommation d'énergie.

6.3 Limites et perspectives du projet

Les objectifs spécifiques (OS1, OS2 et OS3) présentés au Chapitre 3 ont tous été atteints dans ce mémoire. Néanmoins, les performances du quadrirotor joueur de tennis de table ne sont pas parfaites. Les résultats sont tout d'abord loin de ce qu'on peut atteindre avec un bras robotique et puis ils ne sont pas suffisants pour jouer une vraie partie contre un humain. Ceci est dû à plusieurs limitations qui seront discutées dans cette section. Ces limitations concernent plus spécifiquement la conception mécanique, le système de capture de mouvement ainsi que l'estimation de la balle. Des pistes principales d'amélioration sont également mentionnées pour chacun de ces points.

6.3.1 Conception mécanique

Même si les vibrations dans la structure ont été réduites au maximum, elles restent encore présentes comme on peut le voir dans la vidéo accompagnant l'article. Il y a tout d'abord des vibrations au niveau des bras dû au manque de rigidité des pièces imprimées en 3D, soit les pièces de serrage et de support pour le bras. Ces pièces sont imprimées en plastique *Polylactic Acid* (PLA). Une piste d'amélioration possible serait d'usiner ces pièces ou de les imprimer en métal. Cela permettrait de réduire significativement les vibrations au niveau du bras.

Il y a également beaucoup de vibrations présentes au niveau de la gimbal attachée à l'avant du quadrirotor. En effet, cette structure n'a pas été optimisée pour cela, mais plutôt dans une optique de réduction du poids. La connexion entre la gimbal et le cadre du quadrirotor est notamment très flexible. En revanche, puisque le système de la gimbal et de la palette n'est pas très lourd, cela n'a pas un effet catastrophique sur le contrôleur. À l'avenir, on pourrait donc faire un design plus rigide de ce système en incluant des tiges en aluminium par exemple.

Ensuite, on remarque que la courroie employée n'est pas parfaitement rigide. Ceci peut avoir des conséquences sur le contrôle du quadrirotor puisque quand on maintient le servo-moteur à un angle donné, le bras peut tourner légèrement dans les deux sens. Une piste d'amélioration possible consiste à utiliser un système de transmission avec des tiges en aluminium par exemple pour avoir un mécanisme le plus rigide possible. On peut également rajouter un servo-moteur pour le lier directement au bras avec un engrenage, réduisant ainsi tout jeu possible. Néanmoins, cette option rajoute du poids.

6.3.2 Système de capture de mouvement

Les estimations de la position, des vitesses et des orientations fournies en temps réel par le système de capture de mouvement ne sont disponibles qu'à une fréquence de 45 Hz au maximum. Ceci nous empêche d'avoir un contrôle à 100 Hz, qu'on pourrait avoir si ce n'était pas le cas. D'autres systèmes de capture de mouvements permettent d'atteindre des fréquences plus élevées en temps réel comme le système Vicon [41] par exemple.

6.3.3 Estimation de la trajectoire de la balle

Comme discuté précédemment l'estimation de la trajectoire de la balle n'est pas parfaite, ce qui diminue le taux de succès de renvoi de balle. Une piste d'amélioration possible consiste à trouver les coefficients de traînée et de restitution de manière expérimentale autrement que par un processus d'essai-erreur. On peut également se tourner vers des techniques d'estimation de trajectoire de balle de ping-pong plus sophistiquées appliquant de l'intelligence artificielle [42] permettant de réduire considérablement l'erreur d'estimation du point d'impact.

CHAPITRE 7 CONCLUSION ET RECOMMANDATIONS

L'objectif de ce mémoire était de concevoir ainsi que de contrôler un quadrirotor qui est capable de jouer au tennis de table. Un nouveau type de quadrirotor avec des bras inclinables a donc été conçu afin d'atteindre l'agressivité désirée. Pour obtenir un suivi précis sur des trajectoires agressives, un contrôleur iLQR a été implémenté. Ce contrôleur a été couplé à un planificateur de trajectoire réduisant le snap permettant ainsi que générer en temps réel des trajectoires qui peuvent être plus facilement suivi à partir d'un point d'impact imposé. Les résultats en simulation montrent d'abord l'efficacité de la méthode, ayant un taux de succès de renvoi de balle à partir de lancers aléatoires allant jusqu'à 80%. Cela laisse présager des résultats prometteurs également en réel. Le contrôleur de haut niveau a donc été implémenté sur un ordinateur embarqué pour effectuer des tests expérimentaux. Celui-ci s'occupe du contrôle sur les positions et les vitesses dans les directions x , y et z , ainsi que de l'angle d'inclinaison des bras. Un microcontrôleur PX4 mini sert de boucle interne, maintenant le quadrirotor à l'orientation calculée par le contrôleur de haut niveau. Le taux de succès avec le quadrirotor réel est de 40%, basé sur des lancers aléatoires effectués par un humain. De plus, la balle est renvoyée sur le terrain adverse par le quadrirotor.

Le quadrirotor développé est le premier robot aérien qui montrent des renvois de balles lancées par un humain. La précision obtenue en suivi de trajectoire agressive avec ce design mécanique ainsi qu'avec ce contrôleur ouvre la voie aussi vers d'autres applications potentielles ayant les mêmes requis.

Dans le futur, il serait intéressant de tester le quadrirotor sur une table de ping-pong officielle dans un environnement avec plus d'espace ainsi que de programmer la plateforme logicielle de sorte qu'elle tienne compte de la possibilité de faire plusieurs échanges de suite contre un humain. Même si les résultats en simulation montrent que le quadrirotor en serait capable, cela n'a pas encore été implémenté. Avant cela, il serait également pertinent de mettre en place les améliorations suggérées à la Section 6.3 pour améliorer les performances actuelles.

RÉFÉRENCES

- [1] NBIA. (2020) Brain Injury Statistics. [En ligne]. Disponible: <https://www.nbia.ca/brain-injury-statistics/>
- [2] M. P. Barnes et A. B. Ward, *Oxford handbook of rehabilitation medicine*. USA: Oxford University Press, 2005.
- [3] N. Maclean *et al.*, "A critical review of the concept of patient motivation in the literature on physical rehabilitation," *Soc Sci Med*, vol. 50, n°. 4, p. 495-506, 2000.
- [4] Redox. (2020) Hirob. [En ligne]. Disponible: <https://redox-labs.com/project/hirob/>
- [5] H.-J. Noh, S.-H. Lee et D.-H. Bang, "Three-Dimensional Balance Training Using Visual Feedback on Balance and Walking Ability in Subacute Stroke Patients: A Single-Blinded Randomized Controlled Pilot Trial," *Journal of Stroke and Cerebrovascular Diseases*, vol. 28, n°. 4, p. 994-1000, 2019.
- [6] Wikipedia. (2020) Jeux paralympiques d'été de 1960. [En ligne]. Disponible: https://fr.wikipedia.org/wiki/Jeux_paralympiques_d%27%C3%A9t%C3%A9_de_1960
- [7] L. Acosta *et al.*, "Ping-pong player prototype," *IEEE robotics & automation magazine*, vol. 10, n°. 4, p. 44-52, 2003.
- [8] K. Mülling, J. Kober et J. Peters, "A biomimetic approach to robot table tennis," *Adaptive Behavior*, vol. 19, n°. 5, p. 359-376, 2011.
- [9] K. Mülling *et al.*, "Learning to select and generalize striking movements in robot table tennis," *The International Journal of Robotics Research*, vol. 32, n°. 3, p. 263-279, 2013.
- [10] Omron. (2019) About forpheus. [En ligne]. Disponible: <https://www.omron.com/innovation/forpheus.html>
- [11] R. Silva, F. S. Melo et M. Veloso, "Towards table tennis with a quadrotor autonomous learning robot and onboard vision," communication présentée à 2015 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS), 2015, p. 649-655.
- [12] W. Zhou *et al.*, "A testing platform for on-drone computation," communication présentée à 2015 33rd IEEE International Conference on Computer Design (ICCD), 2015, p. 732-735.
- [13] W. Zhou. (2016) Drone-playing ping pong at IBM Research. [En ligne]. Disponible: <https://www.youtube.com/watch?v=YvbHXz3lccc>
- [14] T. Senoo, A. Namiki et M. Ishikawa, "High-speed throwing motion based on kinetic chain approach," communication présentée à 2008 IEEE/RSJ International Conference on Intelligent Robots and Systems, 2008, p. 3206-3211.
- [15] G. Loianno *et al.*, "Estimation, control, and planning for aggressive flight with a small quadrotor with a single camera and IMU," *IEEE Robotics and Automation Letters*, vol. 2, n°. 2, p. 404-411, 2016.
- [16] D. Mellinger, N. Michael et V. Kumar, "Trajectory generation and control for precise aggressive maneuvers with quadrotors," *The International Journal of Robotics Research*, vol. 31, n°. 5, p. 664-674, 2012.

- [17] D. Mellinger et V. Kumar. (2010) Aggressive Maneuvers for Autonomous Quadrotor Flight. [En ligne]. Disponible: <https://www.youtube.com/watch?v=MvRTALJp8DM>
- [18] D. Mellinger et V. Kumar. (2010) Aggressive Quadrotor Part II. [En ligne]. Disponible: https://www.youtube.com/watch?v=geqip_0Vjec
- [19] M. Müller, S. Lupashin et R. D'Andrea, "Quadrocopter ball juggling," communication présentée à 2011 IEEE/RSJ international conference on Intelligent Robots and Systems, 2011, p. 5113-5120.
- [20] Wikipedia. (2019,) Parrot AR. Drone. [En ligne]. Disponible: https://fr.wikipedia.org/wiki/Parrot_AR.Drone
- [21] S. Bouabdallah, "Design and control of quadrotors with application to autonomous flying," thèse de doctorat, dép. de microtechnique, Ecole Polytechnique federale de Lausanne, Lausanne, Switzerland, 2007. [En ligne]. Disponible: <https://infoscience.epfl.ch/record/95939>
- [22] A. Mohebbi, S. Achiche et L. Baron, "Integrated and concurrent detailed design of a mechatronic quadrotor system using a fuzzy-based particle swarm optimization," *Engineering Applications of Artificial Intelligence*, vol. 82, p. 192-206, 2019.
- [23] A. Mohebbi, S. Achiche et L. Baron, "Multi-criteria fuzzy decision support for conceptual evaluation in design of mechatronic systems: a quadrotor design case study," *Research in Engineering Design*, vol. 29, n°. 3, p. 329-349, 2018.
- [24] A. Mohebbi, S. Achiche et L. Baron, "Design of a vision guided mechatronic quadrotor system using design for control methodology," *Transactions of the Canadian Society for Mechanical Engineering*, vol. 40, n°. 2, p. 201-219, 2016.
- [25] D. Mellinger et V. Kumar, "Minimum snap trajectory generation and control for quadrotors," communication présentée à 2011 IEEE International Conference on Robotics and Automation, 2011, p. 2520-2525.
- [26] C. Richter, A. Bry et N. Roy, "Polynomial trajectory planning for aggressive quadrotor flight in dense indoor environments," dans *Robotics Research*: Springer, 2016, p. 649-666.
- [27] M. Faessler, D. Falanga et D. Scaramuzza, "Thrust mixing, saturation, and body-rate control for accurate aggressive quadrotor flight," *IEEE Robotics and Automation Letters*, vol. 2, n°. 2, p. 476-482, 2016.
- [28] M. Faessler, A. Franchi et D. Scaramuzza, "Differential flatness of quadrotor dynamics subject to rotor drag for accurate tracking of high-speed trajectories," *IEEE Robotics and Automation Letters*, vol. 3, n°. 2, p. 620-626, 2017.
- [29] P. Foehn et D. Scaramuzza, "Onboard state dependent lqr for agile quadrotors," communication présentée à 2018 IEEE International Conference on Robotics and Automation (ICRA), 2018, p. 6566-6572.
- [30] M. Kamel *et al.*, "Model predictive control for trajectory tracking of unmanned aerial vehicles using robot operating system," dans *Robot operating system (ROS)*: Springer, 2017, p. 3-39.

- [31] M. Kamel, M. Burri et R. Siegwart, "Linear vs nonlinear mpc for trajectory tracking applied to rotary wing micro aerial vehicles," *IFAC-PapersOnLine*, vol. 50, n°. 1, p. 3463-3469, 2017.
- [32] M. Cutler *et al.*, "Comparison of fixed and variable pitch actuators for agile quadrotors," communication présentée à AIAA Guidance, Navigation, and Control Conference, 2011, p. 6406.
- [33] M. Ryll, H. H. Bühlhoff et P. R. Giordano, "First flight tests for a quadrotor UAV with tilting propellers," communication présentée à 2013 IEEE International Conference on Robotics and Automation, 2013, p. 295-302.
- [34] A. Moutinho, E. Mateos et F. Cunha, "The tilt-quadrotor: concept, modeling and identification," communication présentée à 2015 IEEE International Conference on Autonomous Robot Systems and Competitions, 2015, p. 156-161.
- [35] M. Ryll, H. H. Bühlhoff et P. R. Giordano, "A novel overactuated quadrotor unmanned aerial vehicle: Modeling, control, and experimental validation," *Transactions on Control Systems Technology*, vol. 23, n°. 2, p. 540-556, 2014.
- [36] M. Odelga, P. Stegagno et H. H. Bühlhoff, "A fully actuated quadrotor UAV with a propeller tilting mechanism: Modeling and control," communication présentée à 2016 IEEE international conference on advanced intelligent mechatronics (AIM), 2016, p. 306-311.
- [37] A. Alkamachi et E. Ercelebi, "Modelling and control of H-shaped racing quadcopter with tilting propellers," *Facta Universitatis, Series: Mechanical Engineering*, vol. 15, n°. 2, p. 201-215, 2017.
- [38] J.-P. Burke, "Preliminary Design of Robust Adaptive Control Laws for an Aerial Manipulator," mémoire de maîtrise, School of aerospace and mechanical engineering, University of Oklahoma, Norman, Oklahoma, 2019. [En ligne]. Disponible: <https://hdl.handle.net/11244/319571>
- [39] Coppelia Robotics. (2020) CoppeliaSim. [En ligne]. Disponible: <https://www.coppeliarobotics.com/>
- [40] Y. Zhang *et al.*, "An adaptive trajectory prediction method for ping-pong robots," communication présentée à International conference on intelligent robotics and applications, 2012, p. 448-459.
- [41] Vicon. (2020) Vicon products. [En ligne]. Disponible: <https://www.vicon.com/>
- [42] H.-I. Lin, Z. Yu et Y.-C. Huang, "Ball Tracking and Trajectory Prediction for Table-Tennis Robots," *Sensors*, vol. 20, n°. 2, p. 333, 2020.