



École Centrale de Nantes
Option Robotique / Option Recherche & Développement

PROJET
Contrôle et optimisation d'un drone reconfigurable

Encadrant: Prof. Abdelhamid Chriette
Giacomo Battaglia
José Colombini

Nantes, 22/03/2023

Abstract

Ce document traite la conception et optimisation d'un drone reconfigurable de type repliable, dans le cadre d'un projet de LS2N (Laboratoire des Sciences Numériques de Nantes) et ECN (École Centrale de Nantes).

L'intérêt des drones reconfigurables est dû à leur morphologie adaptative, qui leur permet d'atteindre plusieurs missions dans les domaines de la défense, exploration, etc. Notre travail est inscrit dans la recherche sur ce sujet apportée par LS2N et ECN.

Dans ce document un état de l'art sur le travail précédent à ECN et les principales solutions en littérature est présenté. Un modèle optimal est choisi et son modèle mathématique développé. En plus, un premier étude du contrôle est présenté, avec contrôle optimale en utilisant un algorithme génétique.

Table des matières

Abstract	ii
1 Introduction	1
1.1 Contexte	1
1.2 Planification	2
2 État de l’art	5
2.1 Modèles examinés	6
2.1.1 Modèle ”Old ECN”	6
2.1.2 Modèle ”Origami”	6
2.1.3 Modèle ”Soft pneumatic”	7
2.1.4 Modèle ”Super Reconfigurable”	7
2.1.5 Modèle ”Crash 2 Squash”	7
2.1.6 Modèle ”Passively Folded”	8
2.2 Critères d’évaluation	9
2.3 Choix du modèle	9
3 Design	11
3.1 Modèle mathématique	11
3.1.1 Configuration 4-unfolded	13
3.1.2 Configuration 2-unfolded-2-folded	13
3.1.3 Configuration 4-folded	14
3.1.4 Transitions	14
3.2 Validation du modèle mathématique	14
3.2.1 Simulation 1	15
3.2.2 Simulation 2	15
4 Simulations	17
4.1 Environnements de simulation	17
4.1.1 Matlab/Simulink	18
4.1.2 ROS/Rviz	18
4.1.2.1 Modèle mathématique	21
4.1.3 ROS/Gazebo	23
4.2 Contrôle	23
4.2.1 Résultats du contrôle	24
4.2.1.1 PD/PID	24
4.2.1.2 SMC/Supertwisting	25

4.2.2	Réglage	26
4.2.2.1	Réglage PID avec GA	29
5	Conclusions	31
5.1	Résultats et points critiques	31
5.2	Futur du projet	31

Chapitre 1

Introduction

1.1 Contexte

Ce travail s'inscrit dans le domaine de la recherche sur les UAV (Unmanned Air Vehicle), en particulier sur les quadcopters. Les UAV sont utilisés dans une grande variété d'applications [4]: défense, livraison, observation de la terre, exploration, video making, etc. Cette variété est due à leurs agilité et masse réduite, et la capacité de monter une caméra.

Un autre avantage des quadcopters est le décollage vertical (VTOL) et la capacité de faire du hovering, c'est à dire d'atteindre une position fixe dans l'espace et y rester. Cette particularité n'est pas possible pour les véhicules à aile fixe car la force de portance qui les fait voler dépend de leur vitesse selon l'aérodynamique ($L \propto V^2$). Cette caractéristique fait que la classe de quadcopters est très versatile en application, comme cité auparavant.

Mais une des limitations des quadcopters sont les environnements fermés avec des obstacles, qui peuvent réduire la mobilité du véhicule. Ce type d'environnement est commun dans les missions de *rescue and save*, mapping, reconnaissance d'environnement, etc. Afin d'améliorer ultérieurement l'agilité des quadcopters, une solution qui a été pensée est de les rendre reconfigurables, c'est à dire de leur donner une morphologie adaptative, qui consiste à modifier la forme du véhicule pendant le vol selon la mission désignée ou l'environnement spécifique.

Il y a beaucoup de designs différents concernant les drones reconfigurables [7]. Étant donné l'ampleur de la matière (chapitre 2, figure 2.1), nous nous sommes concentrés sur la catégorie des drones repliables (*foldables*). Ce type de drone reconfigurable change sa

configuration en se pliant par exemple avec l'utilisation d'articulations rigides (comme les bras robotiques) ou d'autres solutions.

Notre travail est une continuation des efforts du groupe de ECN de l'année 2021-22 composé par N. Frappereau, V. Molina, M. Alfleite [18]. Dans leur travail, ils ont effectué un état de l'art, et ils ont conçu un design de drone à courroies, que nous analyserons dans le prochain chapitre (section 2.1.1, figure 1.1).

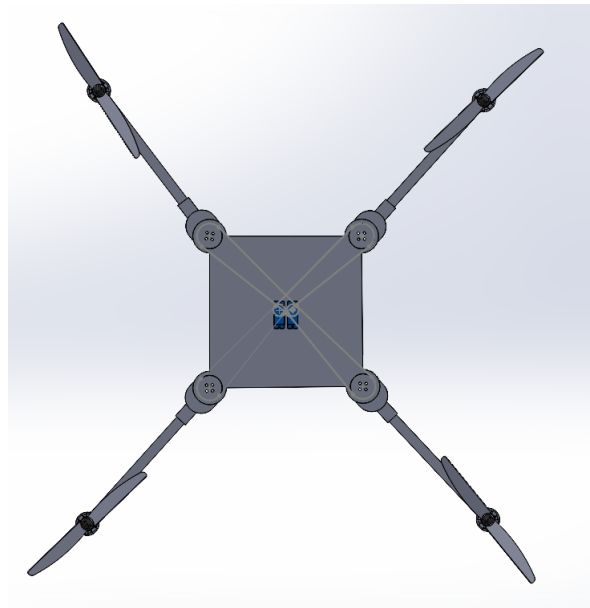


FIGURE 1.1: Modèle ECN de l'année 2021-22 [18].

1.2 Planification

Dans ce contexte de recherche au sein du L2SN, notre work package pour le projet est contenu dans une phase préliminaire de conception du nouveau drone.

Le work package contient:

- Étude de l'ancien design [18] conçu par N. Frappereau, V. Molina, M. Alfleite, état de l'art comparatif.
Milestone: choix du nouveau design pour le développement.
- Modélisation mathématique du design.
Milestone: validation du modèle avec schéma open loop.
- Développement en simulation Matlab/Simulink.
Milestone: démonstration de controle closed loop.

- Développement en simulation ROS simplifié avec Rviz.

Milestone: validation du modèle en comparaison avec Matlab/Simulink.

La planification du projet, pour l'atteinte de ces milestones, est montrée en figure 1.2.

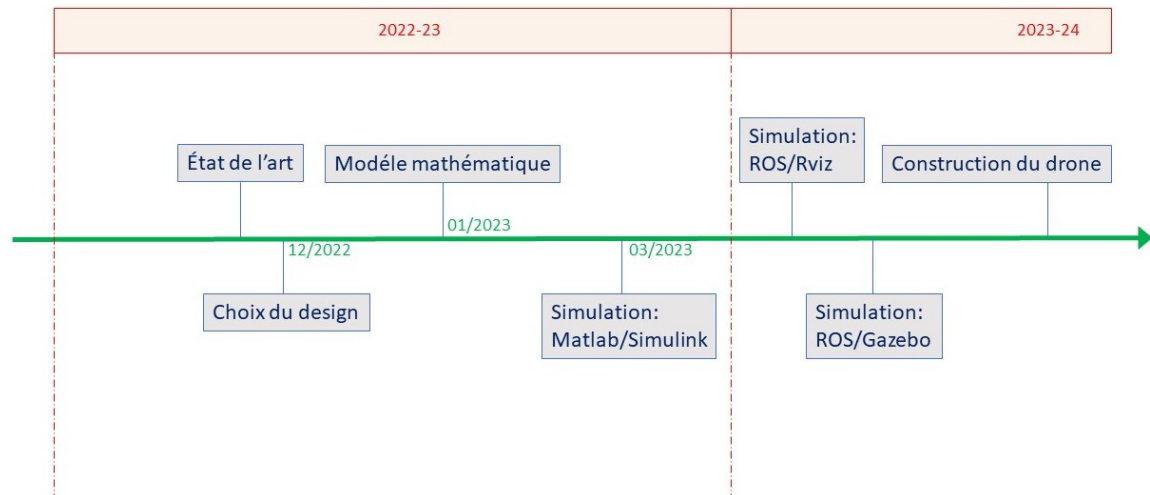


FIGURE 1.2: Notre planification prévue pour le projet.

Dans la prochaine année, nous avons planifié un design détaillé du drone, et développement détaillé en simulation.

Le résultat du design détaillé sera donc au final l'achat des pièces et la construction du premier prototype.

Le rapport est organisé de la manière suivante: dans le Chapitre 2 nous allons présenter l'état de l'art et le processus pour choisir le modèle de drone; le Chapitre 3 présente le modèle choisi et ses caractéristiques; le Chapitre 4 montre le développement, les simulations et les résultats; et le Chapitre 5 donne la conclusion du rapport et des attentes pour les prochaines étapes du projet.

Chapitre 2

État de l'art

Le domaine des drones reconfigurables présente un état de l'art très riche, car beaucoup de solutions ont été proposées [7] pour satisfaire le cahier de charges.

Les drones reconfigurables des diverses classes présentées dans la Figure 2.1 cherchent d'améliorer leurs capacités adaptatives: les drones repliables ont l'objectif de augmenter la mobilité du robot en ajoutant une nouvelle capacité d'éviter des obstacles avec la caractéristique de changement de géométrie en plein vol; les drones modulaires, comme [22], peuvent apporter des objets, les bio inspirés, comme [5], essaient de ajouter l'agilité et la capacité d'accessibilité du robot dans un environnement complexe avec la réduction de taille inspirée aux insectes.

Après avoir étudié les différents modèles proposés en littérature, nous avons sélectionné les plus adaptés à nos finalités, et nous les avons comparés avec le modèle déjà conçu par les étudiants de ECN [18] il y a quelques années.

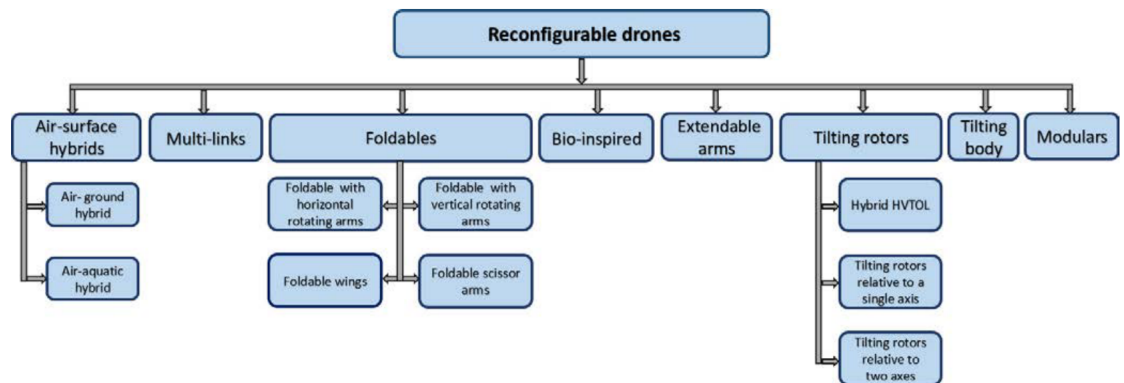


FIGURE 2.1: Schéma détaillant les types de drones reconfigurables [7].

2.1 Modèles examinés

Nous présentons ici les modèles les plus intéressants, en les évaluant avec les critères présentés dans la section 2.2.

2.1.1 Modèle "Old ECN"

Ce modèle [18] a été développé par les étudiants passés de ECN N. Frappereau, V. Molina, M. Alfleite. C'est un modèle assez intéressant pour la simplicité du concept. En effet, il utilise des courroies pour déplacer les 4 bras, 2 à la fois symétriquement. Cette stratégie lui permet de se mettre dans les configurations souhaitées (X, H, O, toutefois pas la T).

Toutefois, le mécanisme à courroies n'est pas assez robuste selon un point de vue structurel, et il requière deux servomoteurs pour déplacer les bras, ce qui n'est pas une solution efficace niveau energy/mass budget.

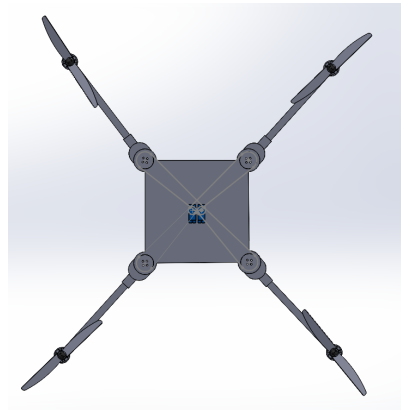


FIGURE 2.2: Modèle "Old ECN" [18].

2.1.2 Modèle "Origami"

Ce modèle [14] présente un mécanisme passif de folding/unfolding basé sur des bras qui peuvent se replier en soi-même, avec des parties magnétiques. La perplexité sur ce design sort du fait que sa masse pourrait être un peu trop élevée, dû aux matériaux ferromagnétiques.

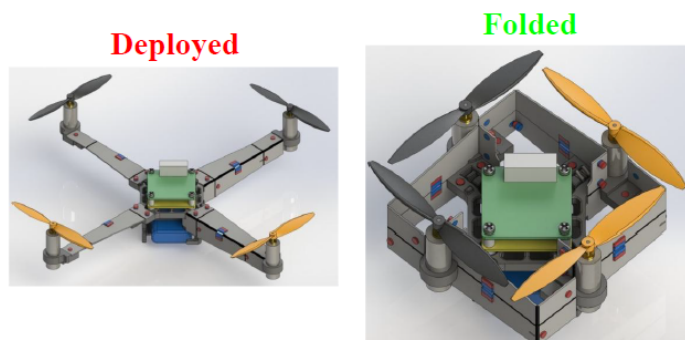


FIGURE 2.3: Modèle "Origami" [14].

En plus, la complexité de ce mécanisme est assez élevée, et il ne peut pas atteindre toutes les configurations souhaitées. Nous n'avons pas trouvé de littérature sur le contrôle de ce drone.

2.1.3 Modèle "Soft pneumatic"

Dans ce modèle [11], les bras sont des mécanismes pneumatiques, ce qui permet d'atteindre beaucoup de configurations différentes. Toutefois, ce type de mécanisme est assez complexe pour un drone, pour ce qui concerne la structure, la construction et un energy/-mass budget. En effet, les systèmes pneumatiques ont besoin de contrôleurs d'air, qui augmentent la masse totale du système.

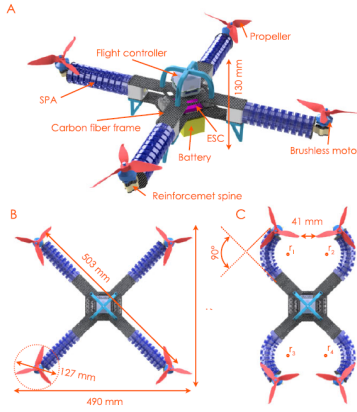


FIGURE 2.4: Modèle "Soft Pneumatic" [11].

Même en ayant beaucoup de possibilités et développements pour le contrôle, grâce à l'infinité de configurations atteignables avec les bras, sa complexité est justifiée seulement pour des applications spécifiques.

2.1.4 Modèle "Super Reconfigurable"

Ce modèle [3] est un peu plus "classique" dans le sens que les bras reconfigurables sont des bras robotiques actifs avec des articulations motorisées qui les déplacent. La particularité de ce modèle est le fait d'avoir deux rotors dans une direction et deux dans la direction opposée, avec des angles de tilt. Cette stratégie permet donc une majeure flexibilité dans le contrôle. D'après la littérature, le modèle a aussi des bonnes performances, avec un contrôle adaptatif, qui a été utilisé dans les essais documentés.

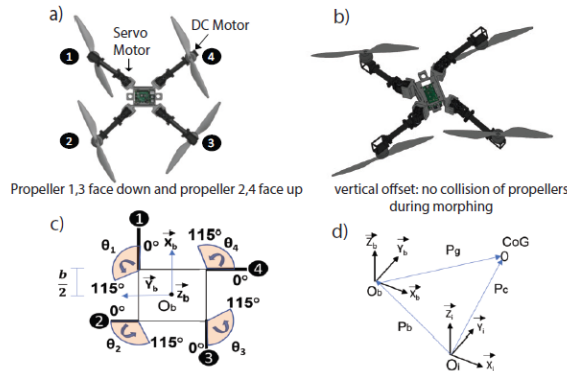


FIGURE 2.5: Modèle "Super Reconfigurable" [3].

Toutefois, ce modèle n'est pas le plus efficace selon le bilan de masse et énergie, car il utilise 4 servo-moteurs pour les bras en plus, ajoutés aux 4 moteurs pour les rotors. Ce choix apporte une complexité majeure au design, niveau structure et contrôle.

2.1.5 Modèle "Crash 2 Squash"

Ce modèle [1] utilise la collision pour se reconfigurer. Il a une structure asymétrique, où deux bras ne sont pas bloqués, mais ils peuvent se replier

quand le drone se cogne contre les murs latéraux d'un trou. Cette stratégie permet d'utiliser un mécanisme de reconfiguration passif, optimal pour le bilan de masse et énergie. Le problème de ce mécanisme toutefois est que le drone est assez grand, et il doit résister aux collisions. En plus, il ne peut pas atteindre beaucoup de configurations, mais seulement celle obtenue par collision.

La complexité de ce drone dérive de la nécessité d'être robuste à collision, et à la présence de beaucoup de parties mobiles à tester.

2.1.6 Modèle "Passively Folded"

Ce modèle [16, 17] a sa force dans sa simplicité. Il ressemble exactement un quadricopter classique, mais ses quatre bras ne sont pas fixes, mais ils sont joints au châssis avec des articulations rotoïdes pas actionnées. Dans ce cas, alors ce sont directement les rotors qui gèrent les mouvements des bras, en tournant en reverse pour les fermer. Il existe deux designs pour ce mécanisme: un avec des ressorts qui tirent les bras vers la position fermée, et un autre sans ressorts, qui utilise simplement le poids des bras et la friction des articulations.

C'est un modèle complètement passif, avec un poids et des dimensions similaires au drone classique, et il peut atteindre beaucoup de différentes configurations: deux bras repliés, pour passer dans des trou horizontaux; tout bras replié, pour passer dans des trous verticaux; deux bras repliés avec charge (charge utile limitée); vol parabolique dans un trou.

Le contrôle est similaire au drone classique, sauf pour les contraintes de poussée des rotors. La complexité du contrôle vient surtout de la gestion des transitoires.



FIGURE 2.6: Modèle "Crash 2 Squash" [1].



FIGURE 2.7: Modèle "Passively Folded" [16, 17].

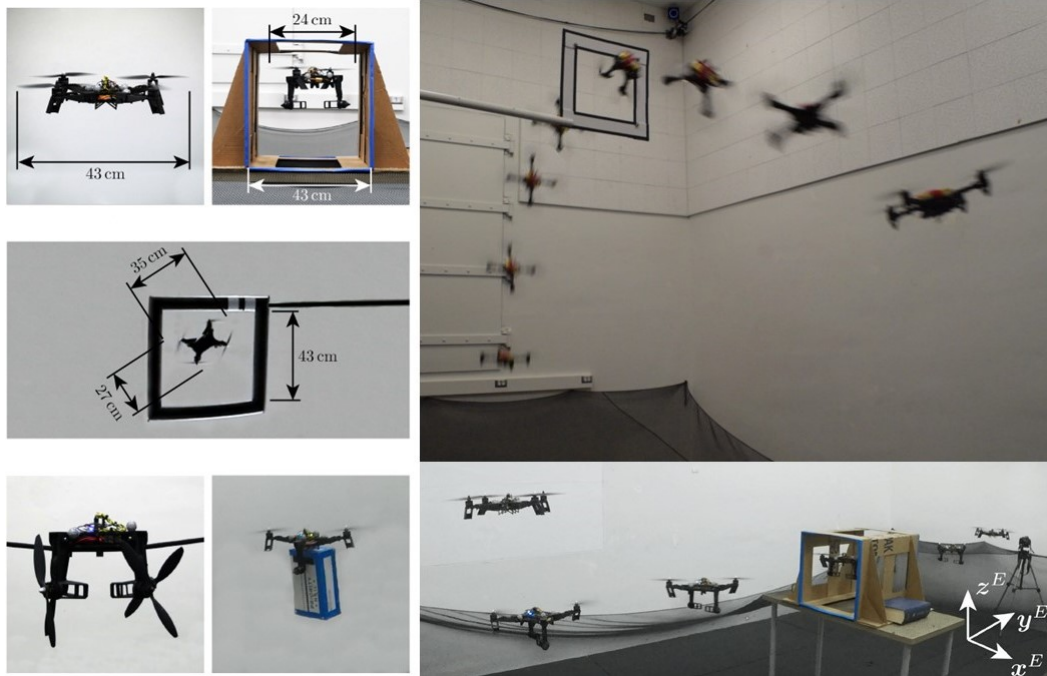


FIGURE 2.8: Le modèle "Passively Folded" en action [16, 17].

2.2 Critères d'évaluation

Les critères que nous avons suivis pour l'évaluation des modèles:

- **Structure**

propriétés structurelles, avantages/désavantages constructifs, différences avec le drone "classique";

- **Energy/mass budget**

économie de masse et puissance, due à la présence d'actionneurs, matériels particuliers, etc.;

- **Contrôle**

lois de contrôle déjà présentes en littérature, possibilités de développement et améliorations

- **Complexité**

complexité générale du design, niveau construction et niveau contrôle.

2.3 Choix du modèle

Nous avons enfin évalué les 4 critères avec des notes de 0 à 10, selon le tableau ci-dessous:

	Structure	E/M budget	Contrôle	Complexité
Old ECN	5	5	5	7
Origami	7	6	3	4
Soft Pneumatic	3	5	8	3
Super Reconfig	5	5	8	6
Crash 2 Squash	7	7	5	6
Passively Folded	7	9	7	7

On note que "Passively Folded" est le modèle avec les meilleures caractéristiques selon nos objectifs. En effet, c'est un design assez simple selon un point de vue structurel, en étant similaire aux drones classiques. En plus, c'est intéressant pour le contrôle, car c'est assez nouveau et donc améliorable. Vu qu'il est passif, il est aussi le drone le plus avantageux selon le mass/energy budget.

Nous avons donc décidé de nous concentrer sur l'étude et le développement de ce modèle, en particulier pour ce qui concerne son contrôle, que nous essayons d'améliorer par rapport à l'existant.

Chapitre 3

Design

3.1 Modèle mathématique

Ce type de quadricopter présente 3 configurations différentes, entre lesquelles il peut changer, contrôlé par l'utilisateur ou par l'autopilote, selon la mission assignée:

- 4-unfolded: les 4 bras sont étendus, et le drone ressemble tout à fait un drone classique;
- 2-unfolded-2-folded: 2 bras sont repliés, et leurs rotors tournent en reverse, pour équilibrer les couples autour de l'axe z;
- 4-folded: les 4 bras sont tous repliés, le drone est en chute libre, et les rotors contrôlent l'attitude du corps.

Dans les différentes configurations, ce qui change est la matrice d'allocation [16, 17] pour obtenir les forces et les couples générées par les rotors en fonction des vitesses de rotation, selon la dynamique classique du drone:

$$m\ddot{\vec{r}} = m\vec{g} + R_{EB}\vec{F}_B + \vec{F}_a \quad (3.1)$$

$$I\dot{\vec{\omega}} + \vec{\omega} \times I\vec{\omega} + \vec{M}_G + \vec{M}_a = \vec{M}_B \quad (3.2)$$

où: m masse du drone, \vec{r} position du drone dans le repère inertiel, R_{EB} matrice de rotation du repère body au repère inertiel, \vec{F}_B forces produites par les rotors (repère

body), \vec{F}_a forces aérodynamiques; I matrice d'inertie du drone, $\vec{\omega}$ vitesse angulaire du drone, \vec{M}_G moments gyroscopiques générés par les rotors, \vec{M}_a couples aérodynamiques, \vec{M}_B couples générées par les rotors.

La matrice d'allocation $A \in \mathbb{R}^{4 \times 4}$ est donc telle que:

$$\begin{pmatrix} F_{Bz} \\ M_{Bx} \\ M_{By} \\ M_{Bz} \end{pmatrix} = A \begin{pmatrix} \Omega_1^2 \\ \Omega_2^2 \\ \Omega_3^2 \\ \Omega_4^2 \end{pmatrix} \quad (3.3)$$

La matrice A dépend de la géométrie du drone et des caractéristiques des rotors, modélisées avec les constantes de poussé (κ_t) et de traînée en sens directe (κ_d^+) et inverse (κ_d^-). Les forces et les couples générées par chaque rotor suivent en effet approximativement une loi quadratique avec les vitesses de rotation: $F = \kappa_t \Omega^2$, $M = \kappa_d \Omega^2$.

Comme la loi qui lie les forces et les couples aux rotations des rotors est quadratique, le signe de la rotation des rotors (et donc leur sens de rotation) est ici perdu. Il sera quand même considéré dans le contrôle, où on le connaît effectivement, selon ce que l'on commande au drone de faire.

Pour ce qui concerne les transitions entre les trois configurations, nous considérons avec bonne approximation des transitions discrètes, et chaque configuration comme donc un système "indépendant" des autres, que le drone peut atteindre. Cette approximation est en effet acceptable, car les transitoires entre les configurations sont presque instantanées et les perturbations qu'elles génèrent sont rapidement récupérables grâce à la dynamique rapide du contrôle du drone. Des analyses plus approfondies pourront être effectuées dans le futur, quand il y aura un design plus détaillé du drone.

Pour ce qui concerne la notation des éléments géométriques du drone, nous suivons la figure 3.1.

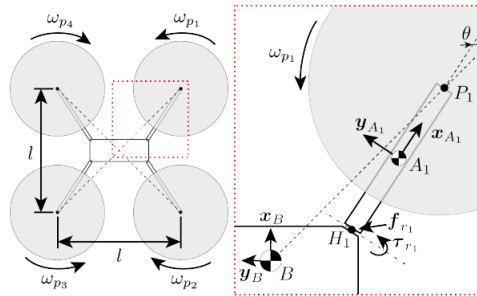


FIGURE 3.1: Paramètres géométriques principaux du drone[16, 17].

3.1.1 Configuration 4-unfolded

La configuration 4-unfolded est la plus classique, et elle ressemble tout à fait un drone standard, quadricopter. La matrice d'allocation dévient:

$$A_u = \begin{bmatrix} \kappa_t & \kappa_t & \kappa_t & \kappa_t \\ -\kappa_t l/2 & -\kappa_t l/2 & \kappa_t l/2 & \kappa_t l/2 \\ -\kappa_t l/2 & \kappa_t l/2 & \kappa_t l/2 & -\kappa_t l/2 \\ -\kappa_d^+ & \kappa_d^+ & -\kappa_d^+ & \kappa_d^+ \end{bmatrix} \quad (3.4)$$

3.1.2 Configuration 2-unfolded-2-folded

Dans la configuration 2-unfolded-2-folded les deux rotors sur les bras repliés tournent en reverse, afin de balancer les couples générés par les autres deux rotors. La forme rectangulaire du châssis centrale et la présence donc d'un angle θ de désalignement (figure 3.1) permet ce type d'équilibre. La force de poussée ici est la moitié, et cela empire forcément les prestations dynamiques du drone.

Des paramètres géométriques qui sont utiles pour décrire les couples générés (voir figure 3.1):

$$p_x = d_{P_i C, z}^B \cos\left(\frac{\pi}{4} + \theta\right) - \frac{\kappa_d^-}{\kappa_t} \sin\left(\frac{\pi}{4} + \theta\right) \quad (3.5)$$

$$p_y = d_{P_i C, z}^B \sin\left(\frac{\pi}{4} + \theta\right) + \frac{\kappa_d^-}{\kappa_t} \cos\left(\frac{\pi}{4} + \theta\right) \quad (3.6)$$

$$p_z = \frac{l\sqrt{2}}{2} \sin \theta \quad (3.7)$$

où $d_{P_i C, z}$ indique la distance en z du centre du rotor au centre de masse du drone.

La matrice d'allocation:

$$A_{2f} = \begin{bmatrix} \kappa_t & 0 & \kappa_t & 0 \\ -\kappa_t l/2 & \kappa_t p_x & \kappa_t l/2 & -\kappa_t p_x \\ -\kappa_t l/2 & -\kappa_t p_y & \kappa_t l/2 & \kappa_t p_y \\ -\kappa_d^+ & \kappa_t p_z & -\kappa_d^+ & \kappa_t p_z \end{bmatrix} \quad (3.8)$$

3.1.3 Configuration 4-folded

Dans la configuration 4-folded, le drone est en chute libre et les rotors font des actions de contrôle d'attitude, mais pas de force de poussé vers le haut.

La matrice d'allocation:

$$A_{4f} = \begin{bmatrix} 0 & 0 & 0 & 0 \\ \kappa_t p_x & \kappa_t p_x & -\kappa_t p_x & -\kappa_t p_x \\ \kappa_t p_y & -\kappa_t p_y & -\kappa_t p_y & \kappa_t p_y \\ \kappa_t p_z & -\kappa_t p_z & \kappa_t p_z & -\kappa_t p_z \end{bmatrix} \quad (3.9)$$

3.1.4 Transitions

Il reste de traiter les transitions entre les configurations. Une explication plus détaillé et des calculs sont présents en bibliographie [16, 2].

Toutefois, en synthèse, la condition pour le repliement d'un des bras peut se mettre comme une condition sur la force de poussée verticale et les couples:

$$c_f F_{Bz} - |c_{mx} M_{Bx}| - |c_{my} M_{By}| - |c_{mz} M_{Bz}| \geq 0 \quad (3.10)$$

où les coefficients dépendent de les propriétés de masse et géométrie du drone [16].

La conséquence de cela sont des limitations sur les couples que le drone peut générer dans la configuration 4-unfolded, par rapport à la force de poussée. En particulier, cela affecte surtout le couple de lacet, si l'on considère un vol stationnaire. Cette limitation réduit légèrement les performances de contrôle du drone.

3.2 Validation du modèle mathématique

Afin de valider le modèle nous avons effectué deux tests open loop (et closed loop simplifié) dans deux modèles différents:

- un modèle simplifié de vol stationnaire, où on adapte le modèle à la représentation d'état classique du drone;
- un modèle un peu plus complexe avec ajout de bruits des moteurs et de mesure.

Les deux simulations ont été faites sur Simulink.

3.2.1 Simulation 1

Dans ce modèle nous avons testé les configurations 4-unfolded et 2-unfolded-2-folded. C'est une simulation simplifiée de la dynamique d'un drone classique où le drone est contrôlé (closed loop) pour suivre une trajectoire à hélice, via un contrôle PD, SMC ou supertwisting. Comme le contrôle n'était pas notre objectif dans cette phase du développement, nous avons enlevé toutes les perturbations et utilisé un contrôle PD pour simplicité, en ne changeant rien du contrôle ou du reste du schéma.

C'est pourquoi nous avons adapté la représentation d'état pour qu'elle soit la représentation classique du drone, donc nous avons trouvé des matrices telles que:

$$u = P\Omega^2 \quad (3.11)$$

où u est le vecteur de commande, P la matrice d'allocation des commandes, Ω^2 le vecteur des vitesses des rotors carrées.

Les deux matrices calculées pour les configurations 4-unfolded et 2-unfolded-2-folded:

$$P_{2f} = \begin{bmatrix} \kappa_t & 0 & \kappa_t & 0 \\ -\kappa_t/2 & \kappa_t p_x/l & \kappa_t/2 & -\kappa_t p_x/l \\ -\kappa_t/2 & -\kappa_t p_y/l & \kappa_t/2 & \kappa_t p_y/l \\ -\kappa_d^+ & \kappa_t p_z & -\kappa_d^+ & \kappa_t p_z \end{bmatrix} \quad (3.12)$$

$$P_u = \begin{bmatrix} \kappa_t & \kappa_t & \kappa_t & \kappa_t \\ -\kappa_t/2 & -\kappa_t/2 & \kappa_t/2 & \kappa_t/2 \\ -\kappa_t/2 & \kappa_t/2 & \kappa_t/2 & -\kappa_t/2 \\ -\kappa_d^+ & \kappa_d^+ & -\kappa_d^+ & \kappa_d^+ \end{bmatrix} \quad (3.13)$$

Avec ces matrices, nous avons pu modifier simplement le bloc relatif au modèle dans le schéma. En cette manière nous vérifions la cohérence du modèle avec cette représentation d'état.

3.2.2 Simulation 2

Dans cette simulation, nous avons effectué des essais open loop et closed loop où nous avons validé toutes les trois configurations. Pour le closed loop le contrôle est un PD qui

génère des forces et couples désirées, et donc des vitesses de rotation des rotors désirées, qui après sont envoyé en input au modèle qui comprend aussi les moteurs.

Ces tests ont été effectués dans le simulateur de Erskine [8] (voir chapitre 4).

Chapitre 4

Simulations

4.1 Environnements de simulation

Pour un développement plus détaillé du drone, nous avons prévu plusieurs environnements pour les simulations, chacun avec des finalités différentes.

Nous avons idéé cette séquence d'environnements avec différents niveaux de détail, qui devront (nous espérons dans l'année 2023-24) ramener à la construction effective du drone.

Les modèles:

- *Matlab/Simulink*: nous avons re-adapté le simulateur de drone de Erskine [8] pour notre drone reconfigurable, afin d'avoir un environnement agile où tester des lois de contrôle rapidement;
- *ROS/Rviz*: nous avons mis les bases pour le développement d'un modèle du drone reconfigurable en ROS avec une application des mêmes équations et presque les mêmes simplifications du modèle Matlab/Simulink, pour pouvoir tester les interfaces et avoir un développement plus détaillé en futur quand il y aura un CAD plus précis du drone;
- *ROS/Gazebo*: nous avons planifié d'utiliser le simulateur RotorS de ETH [9] afin de valider le drone dans un simulateur déjà bien validé, ce qui permet d'avoir le niveau de détail le plus élevé.

La finalité de cette séquence de modèles est de nous permettre d'arriver à justifier un futur achat des pièces et une construction de ce drone.

4.1.1 Matlab/Simulink

La simulation de premier niveau est la partie Matlab/Simulink. Pour cela, nous avons repris et modifié le code du simulateur de Erskine [8], en l'adaptant au cas du drone reconfigurable.

Le simulateur se présente comme en figure 4.1.

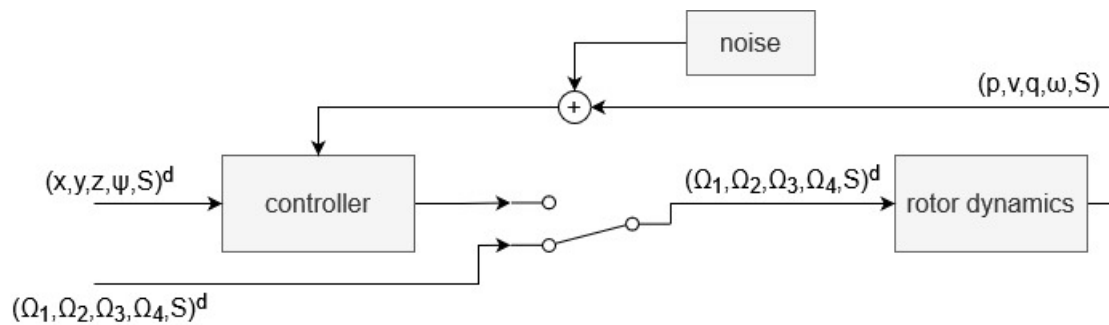


FIGURE 4.1: Le schéma du simulateur de Erskine [8] modifié pour le drone reconfigurable. S est la variable discrète qui définit la configuration du drone.

Il a deux modalités:

- open loop, où l'on injecte directement les vitesses angulaires aux rotors;
- closed loop, où on donne des valeurs de position (x, y, z) et lacet (ψ) désirés, et le contrôleur essaie d'asservir la tâche.

Comme nous utilisons l'hypothèse de états discrets entre les configurations, nous avons créé une variable d'état qui définit la configuration, et donc la matrice d'allocation à utiliser dans le calcul.

En un deuxième moment un modèle analytique a été décrit afin de pouvoir permettre la compréhension dans les divers angles du bras et aussi une simulation plus réelle, tant comme valider le modèle de états discrets.

Les résultats de le modèle analytique, démontré au chapitre 4.1.2.1, a au les mêmes résultats que le modèle discret dans quand la même configuration est appliqué, comment la figure 4.2

4.1.2 ROS/Rviz

Rviz est un outil de l'environnement ROS pour visualiser les robots. Il est possible de simuler le comportement du robot une fois que le modèle physique est connu et sa représentation en URDF a été faite.

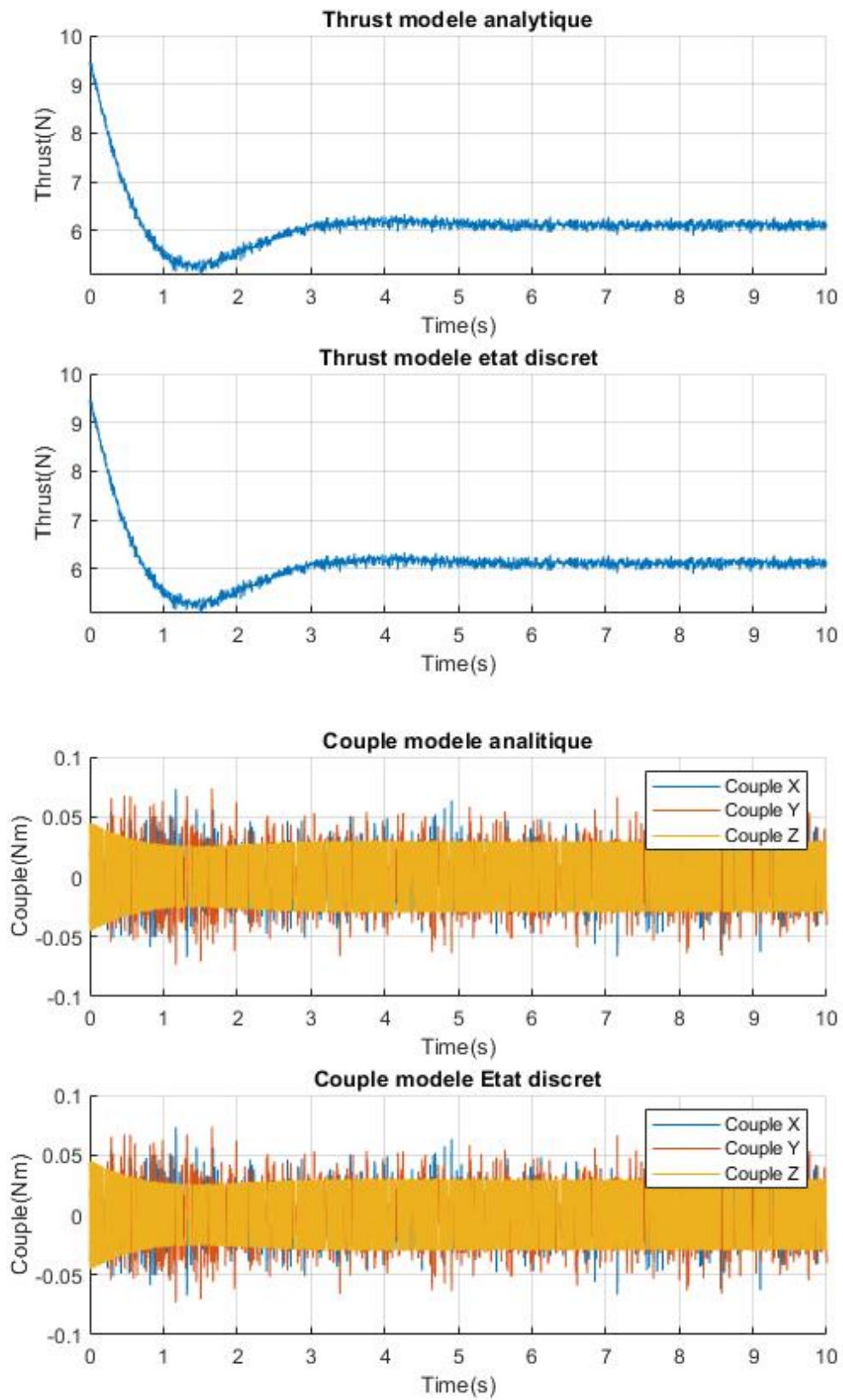


FIGURE 4.2: Les graphes sont les comparaisons des respectives données (force en z au drone (Thrust) et les couples en x , y et z) entre le modèle de état discret et le modèle analytique quand les deux ont les mêmes données de l'entrée.

Pour Rviz nous avons créé un modèle simplifié du robot en URDF avec un frame centrale et 4 bras qu'on a fait le design en SolidWorks et 4 hélices pour lesquelles on a utilisé un design trouve dans un base de données [20]. Il est conçu de manière à former les caractéristiques montré par la figure 3.1. Pour arriver à ce configuration la, on résout l'équation suivante:

$$\begin{cases} a + \frac{F_b}{2} = b + \frac{L_b}{2} \\ a = x \frac{L_a}{D_a} \\ b = \sqrt{L_a^2 - a^2} \end{cases} \quad (4.1)$$

Où x est l'angle a trouver, L_a la longueur du bras, L_b c'est le côté du frame plus grand après la coupure, F_b le côté le plus petit du frame après la coupure, D_a est le face du bras pour lier au frame, en fin a et b sont les longueurs à ajouter dans chaque côté du drone avec le bras au périmètre extérieur. L'équation 4.1 est non exacte, elle donne un résultat proche de celui désiré, qui a été après réglé manuellement au CAD. L'équation a donnée un résultat de $0.50rad$ et le désiré pour le carrée extérieur trouvé est 0.58 .

Cette équation non linéaire qui permet trouver l'angle θ de la figure 3.1. Ce angle entre le frame et les bras du drone qui va créer la stabilisation des couples avec des bras plies a cause de la forme rectangulaire du frame centrale et la forme carre externe dur drone.

Comme résultat le drone assemblé est compose de ces 9 parties et 8 joints (1 rotoide pour chaque bras et 1 continue pour chaque hélice), comment visible dans la Figure 4.4. Ce drone a de des caractéristiques suffisantes pour reproduire le comportement de manière simplifiée du drone. Il reste le modèle physique pour bien représenter le robot.

Notre simulation 3D a été testée en boucle ouverte pour valider les mouvements simples de rotation en tous les 3 axes (x , y , z) et le mouvement en z .

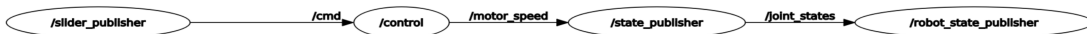


FIGURE 4.3: Graphe des nodes ROS.

Une structure similaire à celle de Matlab a aussi été organisée en ROS, mais pas implémentée, comme vu en Figure 4.3, de manière que ceux qui continueront dans le projet puissent l'implémenter directement et tester avec l'aide de les barres glissantes pour entrée de *setpoints*.

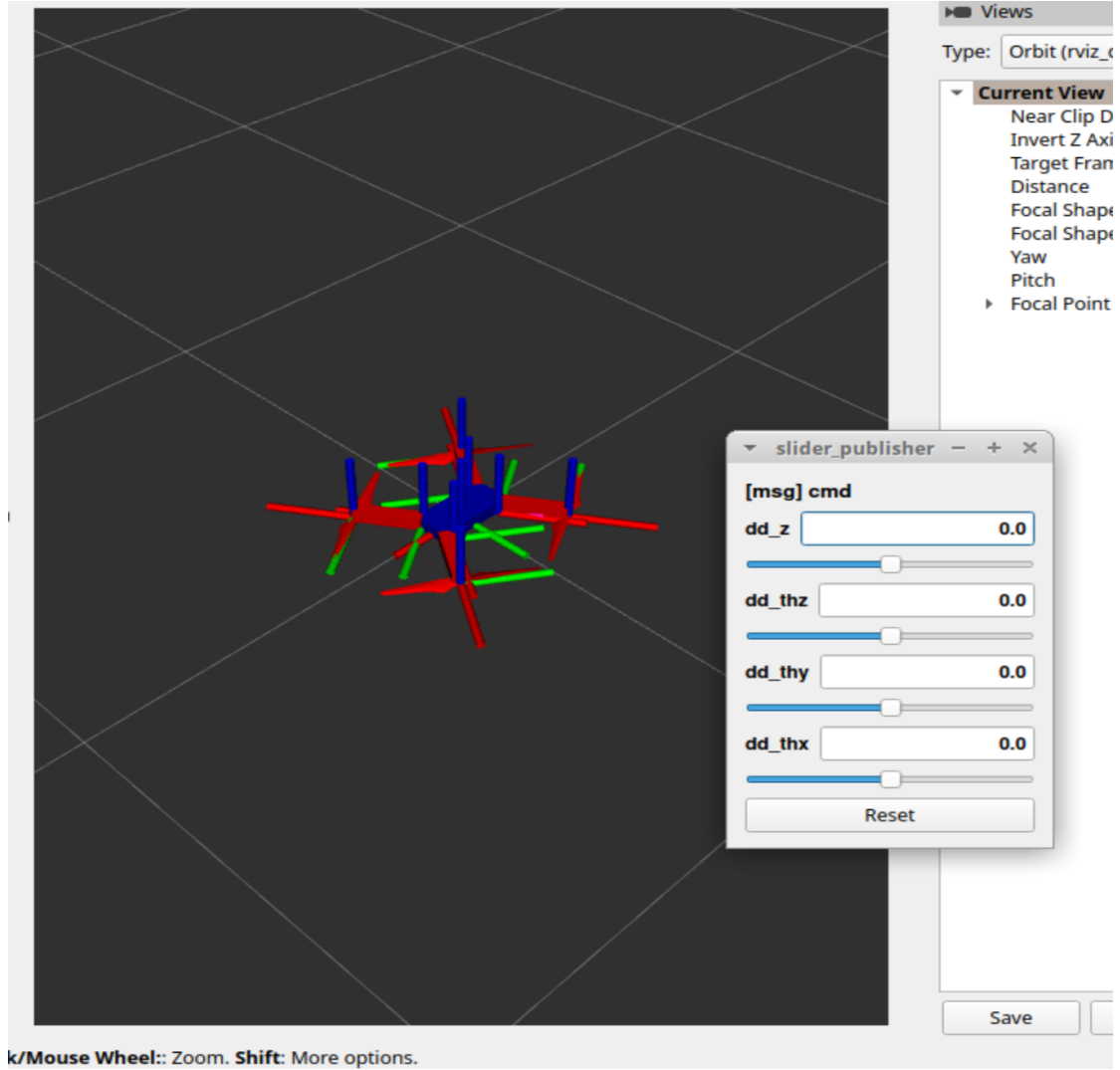


FIGURE 4.4: Modèle URDF importé dans Rviz.

4.1.2.1 Modèle mathématique

Dans ce section le modèle analytique créé pour simuler le système est montré. Ce travail est basé sur [4], [15] [6].

Notre modélisation comprendre tous comment le corps va se comporter sur une certaine configuration, il reste pour les prochaines travaux le calcul de la résistance de chaque joint du bras et sont déplacement sur les conditions corrects. Donc, les bras simules ont la position choisi manuellement et se comportent comment joints actives.

L'état du système est représenté dans la manière suivante:

$$X = [P_w^d \quad Tw_d^d \quad A_d^d \quad \Theta]^T \quad (4.2)$$

Où P_w^d est la pose du drone en relation au monde, Tw_d^d et A_d^d sont respectivement le twist et l'accélération du drone dans son propre repère. Θ est le vecteur de configuration des bras, composé par les angles θ_i de chaque bras en relation à sa position originale (configuration en X).

Pour étudier le robot on a modélisé chaque moteur de manière qu'ils vont générer un wrench proportionnel à sa rotation (Ω) et ses caractéristiques (K_i et K_l), dans son propre repère:

$$W_i = [0 \quad 0 \quad K_i \Omega^2 \quad 0 \quad 0 \quad K_l \Omega^2]^T \quad (4.3)$$

Pour simplifier le modèle on ne considère pas l'effet gyroscope, et comme force externe on considère juste la gravité.

Avec la connaissance de Θ nous pouvons transformer chaque W_i au repère du centre de masse du drone. Il est fait en deux pas: la transfert direct de les forces et des couples au repère du drone et aussi les couple créés pour la position relative du moteur et sa force. Donc on trouve W_d^i .

Pour fin nous pouvons calculer le wrench total sur le drone en son repère comment:

$$W_d = \sum_{i=1}^4 W_i + g_d \quad (4.4)$$

Le dernier étape est calculer l'état du robot une fois qu'on a les forces et couples sur lui. Pour arriver on considère que la simulation a un taux de actualisation r .

Alors, naturellement les équations dynamiques et cinématiques sont satisfaites:

$$\begin{cases} A_d^d(1:3) = \frac{W_d^d(1:3)}{m} & \text{Linaire} \\ A_d^d(4:6) = I^{-1} \cdot W_d^d(4:6) & \text{Angulaire} \\ Tw_{d/k+1}^d = Tw_{d/k}^d + \frac{A_d^d}{r} \\ Tw_w^d(1:3) = T_w^d \cdot Tw_{d/k+1}^d \\ P_{w/k+1}^d = P_{w/k}^d + \frac{Tw_w^d}{r} \end{cases}, \quad (4.5)$$

Le indice k et $k+1$ sont respectivement une représentation de l'échantillonnage, alors on considère le temps passe k pour calculer le prochaine interaction $k+1$. La Matrice I est la matrice de inertie du corps du drone.

4.1.3 ROS/Gazebo

La simulation dans environnement ROS/Gazebo a été planifié sur le simulateur RotorS [9] de ETH.

Le simulateur permet de simuler des scénarios, appliquer les architectures de contrôle développées sur les autres simulateurs et valider leurs performances.

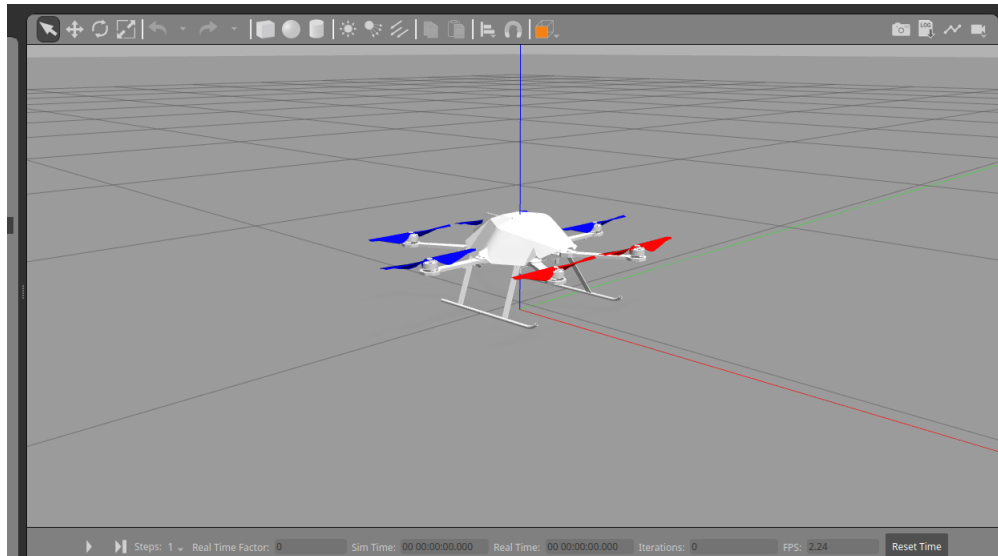


FIGURE 4.5: Aperçu de la simulation Gazebo du RotorS [9] de ETH.

Ce travail est planifié pour l'année 2023-24. Il faudra développer un modèle CAO du drone et l'exporter dans le simulateur, en maintenant la reconfigurabilité. Ce sera aussi l'occasion d'avoir un modèle qui adoucit l'hypothèse de états discrets, pour augmenter en détail et description des transitoires de changements de configuration.

4.2 Contrôle

Le contrôle a été développé en ce projet dans l'environnement Matlab/Simulink.

Nous avons utilisé:

- contrôleur PD, version de Erskine [8], et PID
- contrôleur SMC (Sliding Modes Control)
- contrôleur supertwisting

4.2.1 Résultats du contrôle

Nous avons donc essayé plusieurs approches de contrôle linéaire et non-linéaire. Nous les montrons brièvement ici.

Nous n'avons pas fait un réglage approfondi des contrôleurs parce que ce n'est pas la finalité du travail en cette phase.

4.2.1.1 PD/PID

Nous avons utilisé le PD de Erskine [8]. Le contrôleur a un problème de base d'erreur statique sur la position z (figure 4.6).

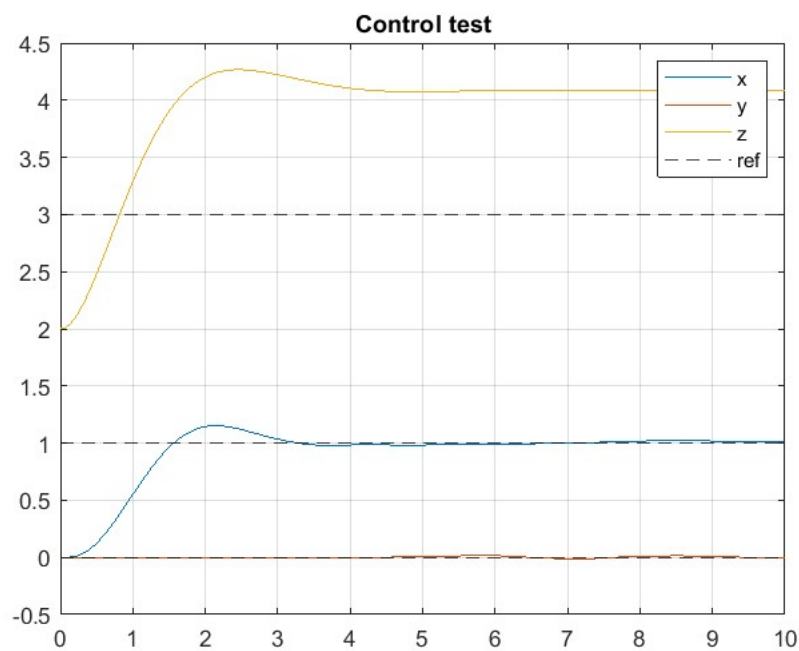


FIGURE 4.6: Test du contrôleur PD de Erskine [8].

Pour corriger l'erreur statique, nous avons modifié le contrôleur de Erskine [8] avec une partie intégrateur sur la partie position du contrôleur (figure 4.7).

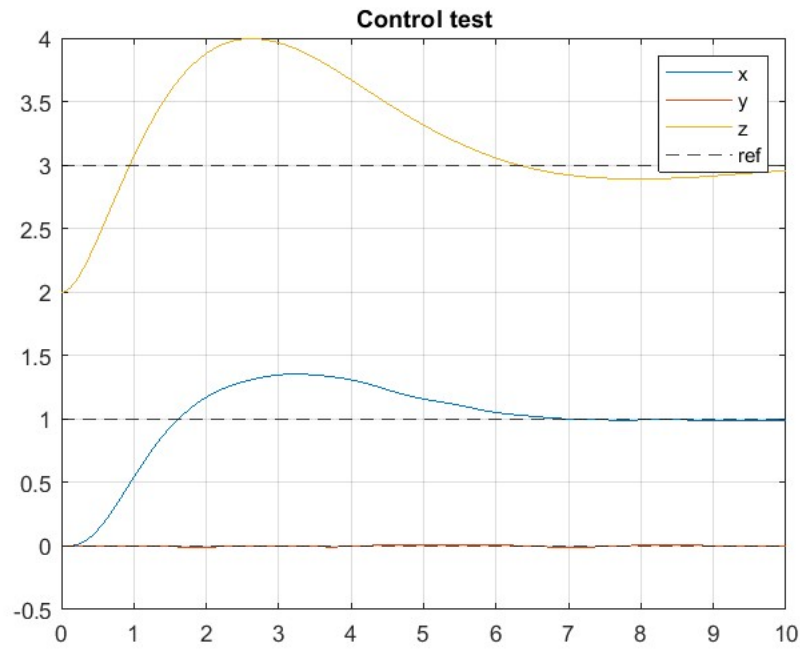


FIGURE 4.7: Test du contrôleur PID de Erskine [8] modifié.

4.2.1.2 SMC/Supertwisting

Nous mettons le SMC (Sliding Mode Control) et Supertwisting ensemble car ils sont deux méthodes de contrôle non-linéaire basés sur modes glissants. SMC est de ordre 1, Supertwisting est de ordre 2.

Tous les deux utilisent un PID dans la partie position, et le contrôle non linéaire dans la partie attitude.

Nous conseillons une étude de ces techniques de contrôle dans des ouvrages de contrôle non linéaire.

Tous les deux ont le même problème de instabilité, en particulier une instabilité qui sort après l'atteinte de la référence de position (figure 4.8). Il faudra corriger ce défaut pour utiliser cette approche de contrôle pour le drone.

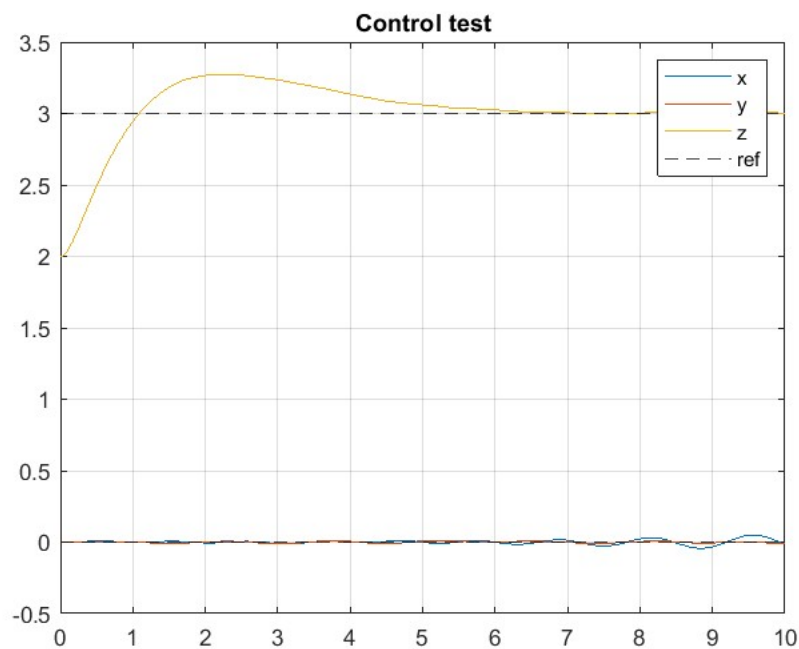


FIGURE 4.8: Test du contrôleur Supertwisting. On note l'oscillation instable en x .

4.2.2 Réglage

Pour utiliser de les contrôleurs de manière correcte nous avons besoin faire un réglage pour stabiliser le système. La manière plus simple de faire est l'utilisation de une méthode euristique, elle va enlever la difficulté de trouver le modèle mathématique pour régler les contrôleurs, en plus la méthode est plus flexible et nous permettre l'appliquer en plusieurs cas, une fois que la fonction a optimiser est bien fait.

Comme le système du drone est un système Multiple-Inputs Mutiple-Outputs (MIMO), sous actionné (avec des paramètres couples), nous avons choisi pour utiliser une méthode heuristique de réglage, l'algorithme génétique (GA). Le GA est déjà utilisé en plusieurs cas: en [10] ils montrent une solution pour le PID online a un résultat meilleur que Ziegler-Nichols, [12] montre la réglage d'un PID MIMO, [19] montre aussi une réglage d'un PID MIMO appliqué a une installation chimique et [21] utilise ce technique pour régler le contrôleur d'un drone quadrotor. Alors l'utilisation de le GA pour régler les contrôleurs PID est déjà une pratique exploré et qui nous enlève la difficulté de trouver le modèle mathématique que stabilise le système.

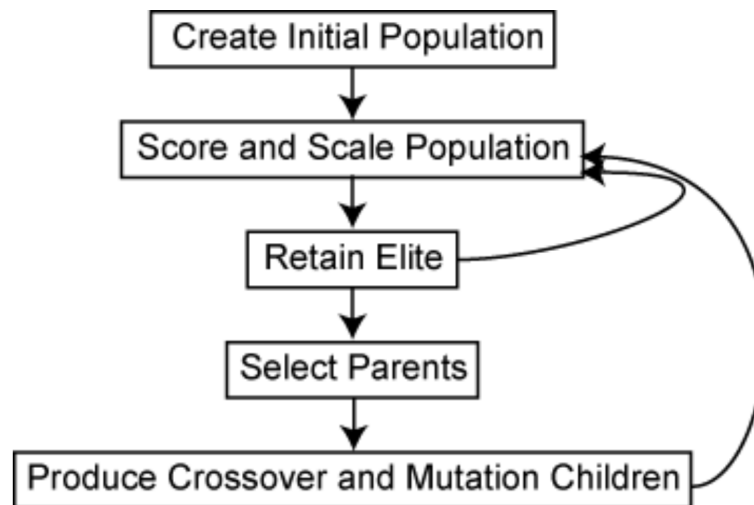


FIGURE 4.9: Le structure d'un GA selon [13].

Le GA est un algorithme basé sur l'idée de sélection naturelle, où l'objectif est de trouver le meilleur individu (un objet qui porte une configuration spécifique des paramètres à régler). Nous allons étudier une population (un grand groupe d'individus) et trouver les meilleurs et les croiser (crossover) pour la prochaine génération.

Selon [13] ce algorithme est composé de 5 étapes, comme le montre la Figure 4.9. Au premier moment, une population initiale est créée et après, elle est testée sur l'environnement de sélection pour voir ses résultats (étape 2). Les meilleurs sont pris pour la prochaine interaction et aussi pour créer la nouvelle génération. Elle est créée par le croisement avec mutation de deux parents (deux individus qui sont dans le groupe d'élite de la dernière génération). Au final, cette nouvelle population, composée des meilleurs de la dernière génération et de ces fils, est testée et reclassée pour générer une nouvelle génération. Le processus continue jusqu'à ce qu'un certain nombre d'interactions ait été fait.

Donc, nous avons besoin d'adapter la structure générique de l'algorithme à notre cas. Nous présenterons chaque adaptation réalisée en suivant l'ordre de la Figure 4.9.

Création initiale de la population: nous utilisons 2 méthodes pour faire cette partie-là. La première méthode est une méthode complètement aléatoire, qu'on génère un contrôleur de paramètres K entre un intervalle $[0, K_{max}]$. Pour la deuxième, nous avons essayé de trouver des contrôleurs manuellement au hasard pour utiliser comme point de départ, alors ce contrôleur est aussi dans la population et un certain nombre de fils est généré avec une mutation d'une distribution normale autour de leurs valeurs de K .

Simulation et classement de la population: la simulation est réalisée par Simulink, de manière qu'on puisse récupérer les données pour mesurer la performance du contrôleur. Les données classiques pour comprendre la performance d'un PID sont l'erreur et comment

il évolue dans le temps, le dépassement [21]. Avec les données d'intérêt on calcule le coût pour le GA avec l'équation 4.6. Notre équation se base sur les travaux [21], [10] et [12].

$$J = \int_0^{\inf} w_1 Err_p(t) + w_2 |Err_{yaw}(t)| + w_3 C(t)^2 dt + w_4 max(overshoot) \quad (4.6)$$

$$Err_p(t) = |Err_x(t)| + |Err_y(t)| + |Err_z(t)|$$

L'équation 4.6 considère les résultats suivants du système: Err_i ($i \in x, y, z, p$) est l'erreur de le drone pour les axes x, y et z respectivement et Err_p est l'erreur totale de position, la somme de toutes les autres erreurs en absolu. *Overshoot* est la valeur maximum en absolu de chaque axe après croiser la référence. Alors on prend le plus grand *overshoot* pour calculer J . Tous ces éléments ajoutent au coût de la fonction pour montrer que la convergence n'est pas bien faite. En fin il y a un dernier paramètre C , qui est la commande du drone et mesure l'énergie utilisée par le contrôle.

Les éléments w_i pondèrent les paramètres du système, les w_i sont choisis manuellement pour privilégier les paramètres selon l'intérêt du système. L'équation est au fin composée de l'intégral des éléments variable sur le temps et on ajoute les éléments non dépendants du temps (*overshoot*).

Choisir les meilleurs: après la simulation nous avons classé les individus en ordre croissant de coût (équation 4.6); comme nous voulons minimiser le coût, les meilleurs sont sélectionnés pour survivre et passer à l'étape suivante pour générer la nouvelle génération.

Sélectionner les parents et générer la nouvelle génération: les gagnants de la dernière étape ici sont choisis 2 à 2 au hasard comme parents pour un nouveau individu. Le processus de création d'un nouveau individu est composé de la mutation des parents: nous avons varié aléatoirement les valeurs de K entre -10% et +10%; et le *crossover*: nous avons choisi pour faire la moyenne des parents après la mutation. Le résultat est représenté par l'équation 4.7, où $rand(min, max)$ est un numéro aléatoire entre min et max.

$$child.K = \frac{(parent_1.K(1 + rand(-10\%, 10\%)) + parent_2.K(1 + rand(-10\%, 10\%)))}{2} \quad (4.7)$$

4.2.2.1 Réglage PID avec GA

Afin de valider l'algorithme, nous avons testé le réglage avec GA sur le contrôleur PID montré dans la section 4.2.1.1.

Le résultat que nous avons obtenu est de très bonne qualité pour ce qui concerne l'asservissement et la régulation (figures 4.10, 4.11).

Ce test a une intention démonstrative, et l'algorithme pourra être utilisé dans le futur pour le design détaillé du modèle.

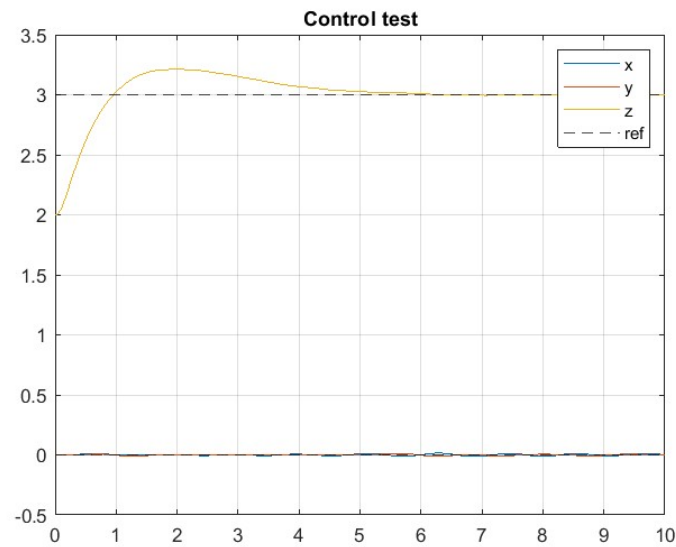


FIGURE 4.10: Le comportement de la position du contrôleur optimisé avec GA.

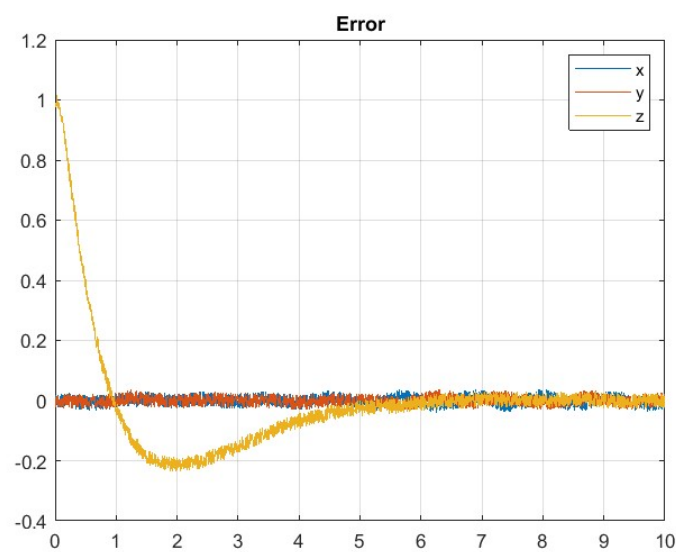


FIGURE 4.11: L'erreur en position du contrôleur optimisé avec GA.

Chapitre 5

Conclusions

5.1 Résultats et points critiques

D'après nos travaux, nous pouvons dire avoir reçu les objectifs prévus dans le planning initial du projet.

L'analyse de l'état de l'art nous a fait choisir un design optimal pour le prochain drone reconfigurable de ECN, qui puisse ouvrir une possibilité de s'insérer dans la recherche en ce domaine avec un modèle déjà performant.

Nous avons donc inséré en simulation le modèle mathématique de ce drone, et nous avons développé des lois de contrôle. Toutefois, le réglage des lois n'a pas été optimisé, donc des méthodes plus de détail (par exemple, algorithme génétique) pourraient être utilisés.

Nous avons aussi commencé le développement du drone en simulation ROS, avec un modèle simplifié dans Rviz, qui devra être complété pour devenir un bon miroir du Matlab/Simulink.

Le point critique principal du travail est la gestion des conditions de transition, et de la configuration 2-unfolded-2-folded.

5.2 Futur du projet

Pour le futur du projet, nous envisageons de suivre le planning montré en figure [1.2](#).

Ce qu'il faudra faire sera commencer une conception détaillée du drone, c'est à dire avancer dans le développement du design d'un point de vue structurel et du contrôle.

Il faudra selon nous se concentrer sur la conception de la structure du drone, avec son frame particulier et les joints passifs des bras. Des impressions 3D pourraient être effectuées pour en tester la fonctionnalité.

Tout ce qui est développement de système embarqué devra être pensé, incluant le choix du contrôleur et des moteurs et rotors.

Le design détaillé pourra donc être inséré dans nos environnements de simulation afin de valider des lois de contrôle pour le drone. Pour avoir un environnement de simulation encore plus représentatif, le simulateur RotorS pourra être implémenté avec notre drone reconfigurable.

Toute cette démarche sera donc nécessaire pour justifier l'achat des pièces et la construction du premier prototype du drone reconfigurable de ECN.

Bibliographie

- [1] Stefano Mintchev Amadeo Fabris Emanuele Alcone. “Crash 2 Squash: An Autonomous Drone for the Traversal of Narrow Passageways”. In: (2022).
- [2] Nathan Bucki. *Midair reconfigurable quadcopter*. <https://github.com/nlbucki/MidairReconfigurableQuadcopter>. 2021.
- [3] Jawad Mehmood Butt et al. “Adaptive Flight Stabilization Framework for a Planar 4R-Foldable Quadrotor: Utilizing Morphing to Navigate in Confined Environments”. In: (2022), pp. 1–7. DOI: [10.23919/ACC53348.2022.9867402](https://doi.org/10.23919/ACC53348.2022.9867402).
- [4] A. Chriette. “Lecture: RASOM 2022-2023, Ecole Centrale de Nantes”.
- [5] GCHE De Croon et al. “Design, aerodynamics, and vision-based control of the DelFly”. In: *International Journal of Micro Air Vehicles* 1.2 (2009), pp. 71–97.
- [6] L. Derafa, T. Madani, and A. Benallegue. “Dynamic Modelling and Experimental Identification of Four Rotors Helicopter Parameters”. In: *2006 IEEE International Conference on Industrial Technology*. 2006, pp. 1834–1839. DOI: [10.1109/ICIT.2006.372515](https://doi.org/10.1109/ICIT.2006.372515).
- [7] Saddam Hocine Derrouaoui et al. “A Comprehensive Review on Reconfigurable Drones: Classification, Characteristics, Design and Control Technologies”. In: *Unmanned Systems* 10.01 (2022), pp. 3–29. DOI: [10.1142/S2301385022300013](https://doi.org/10.1142/S2301385022300013). eprint: <https://doi.org/10.1142/S2301385022300013>. URL: <https://doi.org/10.1142/S2301385022300013>.
- [8] Julian Erskine. “Dynamic Control and Singularities of Rigid Bearing-Based Formations of Quadrotors”. In: (2021).
- [9] ETHZ. *RotorS Simulator*. https://github.com/ethz-asl/rotors_simulator. 2021.
- [10] Liu Fan and Er Meng Joo. “Design for auto-tuning PID controller based on genetic algorithms”. In: *2009 4th IEEE Conference on Industrial Electronics and Applications*. 2009, pp. 1924–1928. DOI: [10.1109/ICIEA.2009.5138538](https://doi.org/10.1109/ICIEA.2009.5138538).

- [11] Andreas Papadimitriou Jakub Haluska Jim Västnål. “Soft Pneumatic Actuated Morphing Quadrotor: Design and Development”. In: (July 2022).
- [12] Jih-Gau Juang, Ming-Te Huang, and Wen-Kai Liu. “PID Control Using Presearched Genetic Algorithms for a MIMO System”. In: *IEEE Transactions on Systems, Man, and Cybernetics, Part C (Applications and Reviews)* 38.5 (2008), pp. 716–727. DOI: [10.1109/TSMCC.2008.923890](https://doi.org/10.1109/TSMCC.2008.923890).
- [13] Mathworks. *What Is the Genetic Algorithm?* <https://it.mathworks.com/help/gads/what-is-the-genetic-algorithm.html>.
- [14] Stefano Mintchev. “Foldable technologies and compound eyes for pocket sized drones”. In: 2022.
- [15] V. Mistler, A. Benallegue, and N.K. M’Sirdi. “Exact linearization and noninteracting control of a 4 rotors helicopter via dynamic feedback”. In: *Proceedings 10th IEEE International Workshop on Robot and Human Interactive Communication. ROMAN 2001 (Cat. No.01TH8591)*. 2001, pp. 586–593. DOI: [10.1109/ROMAN.2001.981968](https://doi.org/10.1109/ROMAN.2001.981968).
- [16] Mark W. Mueller Nathan Bucki. “Design and Control of a Passively Morphing Quadcopter”. In: (2021).
- [17] Mark W. Mueller Nathan Bucki Jerry Tang. “Design and Control of a Midair-Reconfigurable Quadcopter using Unactuated Hinges”. In: (Dec. 2021).
- [18] Mirna Alfleite Nathan Frappereau Valentin Molina. “Conception et optimisation d’un drone reconfigurable”. In: (2021), p. 24.
- [19] Eugenio Turco Neto, Syed A. Imtiaz, and Salim Ahmed. “A sequential optimization procedure to tune MIMO PID controllers”. In: *2017 Indian Control Conference (ICC)*. 2017, pp. 415–420. DOI: [10.1109/INDIANCC.2017.7846510](https://doi.org/10.1109/INDIANCC.2017.7846510).
- [20] PX4. *x500*. Open Robotics. Aug. 2022. URL: <https://fuel.gazebosim.org/1.0/PX4/models/x500>.
- [21] Youcef Zennir et al. “Modelling and GA Fine-tuned PID Control of Quadrirotor UAV”. In: *2019 19th International Conference on Sciences and Techniques of Automatic Control and Computer Engineering (STA)*. 2019, pp. 53–58. DOI: [10.1109/STA.2019.8717238](https://doi.org/10.1109/STA.2019.8717238).
- [22] Moju Zhao et al. “Whole-body aerial manipulation by transformable multirotor with two-dimensional multilinks”. In: *2017 IEEE International Conference on Robotics and Automation (ICRA)*. 2017, pp. 5175–5182. DOI: [10.1109/ICRA.2017.7989606](https://doi.org/10.1109/ICRA.2017.7989606).