



Titre: Commande d'un quadricoptère par rejet actif de perturbations
Title:

Auteur: Vincent Dabin
Author:

Date: 2018

Type: Mémoire ou thèse / Dissertation or Thesis

Référence: Dabin, V. (2018). Commande d'un quadricoptère par rejet actif de perturbations
Citation: [Mémoire de maîtrise, École Polytechnique de Montréal]. PolyPublie.
<https://publications.polymtl.ca/3263/>

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

URL de PolyPublie: <https://publications.polymtl.ca/3263/>
PolyPublie URL:

Directeurs de recherche: David Saussié, & Ouassima Akhrif
Advisors:

Programme: génie électrique
Program:

UNIVERSITÉ DE MONTRÉAL

COMMANDE D'UN QUADRICOPTÈRE PAR REJET ACTIF DE PERTURBATIONS

VINCENT DABIN
DÉPARTEMENT DE GÉNIE ÉLECTRIQUE
ÉCOLE POLYTECHNIQUE DE MONTRÉAL

MÉMOIRE PRÉSENTÉ EN VUE DE L'OBTENTION
DU DIPLÔME DE MAÎTRISE ÈS SCIENCES APPLIQUÉES
(GÉNIE ÉLECTRIQUE)
AOÛT 2018

UNIVERSITÉ DE MONTRÉAL

ÉCOLE POLYTECHNIQUE DE MONTRÉAL

Ce mémoire intitulé :

COMMANDE D'UN QUADRICOPTÈRE PAR REJET ACTIF DE PERTURBATIONS

présenté par : DABIN Vincent

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

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

M. SAYDY Lahcen, Ph. D., président

M. SAUSSIÉ David, Ph. D., membre et directeur de recherche

Mme AKHRIF Ouassima, Ph. D., membre et codirectrice de recherche

M. CHARLAND-ARCAND Guillaume, M.Sc.A., membre

DÉDICACE

*À toutes les personnes et amis que j'ai rencontrés à Montréal et qui ont fait de mes deux
années de maîtrise une formidable expérience...
vous me manquerez!*

REMERCIEMENTS

Je tiens à remercier tout particulièrement mon directeur de recherche David Saussié et ma co-directrice de recherche Ouassima Akhrif pour m'avoir aidé et guidé tout au long de cette maîtrise recherche.

J'aimerais aussi remercier mes parents pour leur soutien émotionnel et financier sur ces deux dernières années. Je voudrais adresser une mention toute particulière à Gabriel Laparra et Pierre Daligault sans qui la vie d'étudiant en maîtrise recherche n'aurait pas été la même, ainsi que tous mes autres amis qui m'ont aidé et soutenu pendant toute la durée de ce projet, ce qui a été indispensable à son bon déroulement. Il serait injuste de ne pas remercier Anaëlle Nedelec pour son investissement dans les associations AECSP et Polyexplore qui ont permis d'égayer les longs hivers passés à Montréal. Et enfin un grand merci à Mathilde Do pour avoir rendu mon séjour à Montréal un des plus agréables qui soit !

Merci à tous !

RÉSUMÉ

Ce projet de recherche a pour but de synthétiser, optimiser et tester une loi de commande ADRC (*Active Disturbance Rejection Controller*) peu utilisée à ce jour dans le domaine de l'aérospatial. Ce nouveau type de commande consiste à estimer et rejeter les perturbations totales que subit le système (e.g., les forces du vent). Les performances en vol du quadricoptère sont améliorées en synthétisant un contrôleur à la fois robuste et adaptatif. Les performances de notre contrôleur seront comparées à celles de deux autres types de contrôleur plus classiques, i.e., un contrôleur PID (Proportionnel Intégral Dérivé) et un retour d'état LQR (*Linear Quadratic Regulator*).

Le travail effectué pour cette maîtrise a tout d'abord débuté par une revue de littérature approfondie du fonctionnement de l'ADRC, menant à une compréhension théorique complète des mathématiques régissant le contrôleur. Cette étape est indispensable pour être capable ensuite de synthétiser un contrôleur ADRC, mais aussi d'en analyser les résultats, puis d'optimiser les performances en réglant les paramètres adéquats du contrôleur.

Un des critères de performance le plus souvent utilisé pour comparer la robustesse des contrôleurs est la mesure de l'écart-type de l'erreur de la position du quadricoptère. Les simulations et les expériences consistent donc à placer notre drone dans un environnement venteux, en lui donnant une position de référence qu'il doit maintenir. Ainsi, moins le drone s'écarte de sa position, plus l'algorithme de contrôle est considéré comme efficace. Cette efficacité est mesurée et comparée entre les différents contrôleurs en jeu.

ABSTRACT

This master thesis focuses on the design, improvement and optimization of the controller ADRC (*Active Disturbance Rejection Controller*), the use of which is still uncommon in the field of aerospace. This new type of controller consists of estimating and rejecting the disturbances that our system undergoes (like wind forces for example). We will therefore seek to improve the flying performance of a quadrotor by synthesizing a controller that is both robust and adaptive. We will compare our controller's performance with two other, much more common controller types. The two controllers in question are a PID (*Proportional Integral Derivative*) and an LQR (*Linear Quadratic Regulator*).

This master thesis starts with a thorough literature review of the ADRC controller, leading to a theoretical understanding of the controller's operation. This step is essential to be able to synthesize an ADRC controller, analyze the results, and optimize the performance by tuning the appropriate parameters of the controller.

One of the most often used performance criteria to compare the robustness of controllers is the measurement of the standard deviation of the quadcopter position error. Simulations and experiments therefore consist in placing our drone in a windy environment, giving it a reference position, without moving from the desired position. Therefore, the less the drone moves, the more effective the control algorithm is. And it is this efficiency that we will measure and compare between the different controllers involved.

TABLE DES MATIÈRES

DÉDICACE	iii
REMERCIEMENTS	iv
RÉSUMÉ	v
ABSTRACT	vi
TABLE DES MATIÈRES	vii
LISTE DES TABLEAUX	ix
LISTE DES FIGURES	x
LISTE DES SIGLES ET ABRÉVIATIONS	xiii
CHAPITRE 1 INTRODUCTION	1
CHAPITRE 2 REVUE DE LITTÉRATURE	3
2.1 La naissance de l'ADRC	3
2.2 Fonctionnement de l'ADRC	5
2.2.1 Suiveur différentiel	5
2.2.2 Retour d'erreur d'état	7
2.2.3 Estimateur d'état étendu	9
2.2.4 Fonctionnement global	11
CHAPITRE 3 MODÉLISATION DU QUADRICOPTÈRE	13
3.1 Équations dynamiques	13
3.2 Les moteurs	19
3.3 Effet du vent	19
3.3.1 Modélisation du vent	20
3.3.2 Forces et moments induits	21
CHAPITRE 4 SYNTHÈSE DES CONTRÔLEURS	25
4.1 Contrôleur LQR	25
4.2 Contrôleur PID	27

4.3	ADRC	29
4.3.1	Estimateur linéaire	30
4.3.2	Estimateur non-linéaire	33
4.3.3	Optimisation des estimateurs	34
4.3.4	Décorrélation des pôles des états associés	38
4.3.5	Le multi-estimateur	42
CHAPITRE 5 COMPARAISON DES PERFORMANCES DE VOL DES DIFFÉRENTS CONTRÔLEURS EN SIMULATION		56
5.1	Résultats des simulations sur MATLAB pour le drone AscTec Pelican	56
5.2	Résultats des simulations sur Matlab pour le drone Crazyflie	60
5.3	Résultats des simulation dans ROS et Gazebo	64
CHAPITRE 6 COMPARAISON EXPÉRIMENTALE DES PERFORMANCES EN VOL DES DIFFÉRENTS CONTRÔLEURS		73
6.1	Mode opératoire des expériences	73
6.2	Résultats obtenus avec le Rolling-Spider	74
CHAPITRE 7 CONCLUSION ET RECOMMANDATIONS		81
BIBLIOGRAPHIE		83

LISTE DES TABLEAUX

Tableau 3.1	Tableau récapitulatif des notations	16
Tableau 3.2	Valeurs des constantes utilisées dans le modèle de Dryden	21
Tableau 4.1	Valeurs des gains	29
Tableau 5.1	Valeurs des paramètres du drone AscTec utilisées pour les simulations	56
Tableau 5.2	Écart-type selon les différents contrôleurs pour différentes grandeurs pour le drone AscTec Pelican	60
Tableau 5.3	Valeurs des paramètres du drone Crazyflie utilisées pour les simulations	61
Tableau 5.4	Valeurs des gains PID pour le Crazyflie	61
Tableau 5.5	Écart-type selon les différents contrôleurs pour x , z et θ pour le drone Crazyflie	64
Tableau 5.6	Écart-type en position pour le PID et son ADRC associé	69
Tableau 5.7	Écart-type en position du PID, de l'ADRC sans rejet des moments perturbateurs (ADRC1) et avec (ADRC2)	70
Tableau 5.8	Écart-type en position pour le PID et son ADRC associé	71
Tableau 5.9	Écart-type en position pour les différents ω	72
Tableau 5.10	Écart-type en θ pour les différents ω	72
Tableau 6.1	Écart-type selon les différents contrôleurs pour différentes grandeurs pour le drone Rolling-Spider	78
Tableau 6.2	Énergie consommée pour les différents contrôleurs pour 5 minutes de vol avec un vent de 3 m/s	80

LISTE DES FIGURES

Figure 2.1	Graphe fonctionnel de l'ADRC comprenant les 3 composantes TD, SEF, et ESO	5
Figure 2.2	Comparaison des gains linéaires et non-linéaires pour $\alpha = 0.5$ et $\delta = 0.1$	8
Figure 3.1	Représentation du système de coordonnées du quadricoptère (Tran et al., 2015)	13
Figure 3.2	Vitesse du vent selon les 3 axes en fonction du temps	21
Figure 3.3	Drone AscTec Pelican ($651 \times 651 \times 188$ mm)	22
Figure 3.4	Expérience permettant de mesurer les coefficients $C_{F[x,y,z]}$ et $C_{M[x,y,z]}$ pour plusieurs angles d'attaque du vent (Theys et al., 2014)	23
Figure 4.1	Représentation du contrôle LQR avec action intégrale (Tran et al., 2015)	26
Figure 4.2	Structure PID classique	28
Figure 4.3	Comparaison des gains linéaires et non-linéaires	33
Figure 4.4	Comparaison du moment perturbateur estimé et réel selon x	34
Figure 4.5	Écart-type de la position x en fonction de ω_0	35
Figure 4.6	Comparaison de la force perturbatrice estimée et réelle selon x pour $\omega_0 = 1$	36
Figure 4.7	Comparaison de la force perturbatrice estimée et réelle selon x pour $\omega_0 = 7$	36
Figure 4.8	Zoom sur la comparaison de la force perturbatrice estimée et réelle selon x pour $\omega_0 = 7$	37
Figure 4.9	Zoom sur la comparaison de la force perturbatrice estimée et réelle selon y pour $\omega_0 = 7$	37
Figure 4.10	Comparaison d'estimations de x_3 avec un pôle triple $\omega_0 = 2$ rad/s et de trois pôles décorrélés $\omega_1 = 2$ rad/s, $\omega_2 = 6$ rad/s et $\omega_3 = 20$ rad/s . . .	41
Figure 4.11	Comparaison d'estimations de x_1 avec un pôle triple $\omega_0 = 2$ rad/s et de trois pôles décorrélés $\omega_1 = 2$ rad/s, $\omega_2 = 6$ rad/s et $\omega_3 = 20$ rad/s . . .	42
Figure 4.12	Pic d'erreur d'un estimateur HGO en fonction de ϵ	43
Figure 4.13	Fonctionnement du multi-estimateur (Bernat and Stepien, 2015) . . .	43
Figure 4.14	Réduction du phénomène de pic	47
Figure 4.15	x_1 estimé en fonction du temps pour les différents estimateurs et le multi-estimateur	48
Figure 4.16	x_2 estimé en fonction du temps pour les différents estimateurs et le multi-estimateur	48

Figure 4.17	e_1 du meilleur estimateur simple et du multi-estimateur en fonction du temps	49
Figure 4.18	e_2 du meilleur estimateur simple et du multi-estimateur en fonction du temps	49
Figure 4.19	e_1 du meilleur estimateur simple et du multi-estimateur en fonction du temps	50
Figure 4.20	e_2 du meilleur estimateur simple et du multi-estimateur en fonction du temps	51
Figure 4.21	e_1 du meilleur estimateur simple et du multi-estimateur en fonction du temps avec $\frac{e_{simple}(0)}{e_{multi}(0)} = \frac{1}{10}$	52
Figure 4.22	e_2 du meilleur estimateur simple et du multi-estimateur en fonction du temps avec $\frac{e_{simple}(0)}{e_{multi}(0)} = \frac{1}{40}$	52
Figure 4.23	Comparaison de l'état réel, de l'état estimé par un estimateur simple et multiple de la force perturbatrice selon x	53
Figure 4.24	Comparaison de l'état réel, de l'état estimé par un estimateur simple et multiple de la force perturbatrice selon y	54
Figure 4.25	Comparaison des performances du contrôleur avec un estimateur simple et un multi-estimateur	54
Figure 5.1	Position x du drone en fonction du temps pour les différents contrôleurs	57
Figure 5.2	Position z du drone en fonction du temps pour les différents contrôleurs	57
Figure 5.3	Angles d'Euler ϕ du drone en fonction du temps pour les différents contrôleurs	58
Figure 5.4	Angle d'Euler θ du drone en fonction du temps pour les différents contrôleurs	59
Figure 5.5	Angle d'Euler ψ du drone en fonction du temps pour les différents contrôleurs	59
Figure 5.6	Drone Crazyflie	60
Figure 5.7	Position en x en fonction du temps pour un Crazyflie selon différentes vitesse d'estimation de l'ADRC	62
Figure 5.8	Position en z en fonction du temps pour un Crazyflie selon différentes vitesse d'estimation de l'ADRC	62
Figure 5.9	Angle θ en fonction du temps pour un Crazyflie selon différentes vitesse d'estimation de l'ADRC	63
Figure 5.10	Zoom #1 sur la figure 5.9 représentant l'angle θ en fonction du temps pour un Crazyflie selon différentes vitesse d'estimation de l'ADRC . .	63

Figure 5.11	Zoom #2 sur la figure 5.9 représentant l'angle θ en fonction du temps pour un Crazyflie selon différentes vitesses d'estimation de l'ADRC . . .	64
Figure 5.12	Comparaison de deux ADRC et du PID pour un échelon en x sans vent	65
Figure 5.13	Structure classique d'un ADRC	66
Figure 5.14	Structure intuitive d'un contrôle ADRC à deux niveaux	66
Figure 5.15	Structure corrigée d'un contrôle ADRC à deux niveaux	67
Figure 5.16	Comparaison du contrôleur PID et ADRC associé	68
Figure 5.17	Comparaison du PID, de l'ADRC sans rejet des moments perturbateurs (ADRC1) et avec (ADRC2)	69
Figure 5.18	Comparaison des PD-PID et PID-PD ainsi que leur contrôleur ADRC associé	70
Figure 5.19	Comparaison des contrôleurs ADRC pour différents ω	71
Figure 5.20	Comparaison des contrôleurs ADRC en θ pour différents ω	72
Figure 6.1	Vicon Tracker System	73
Figure 6.2	Caméra du Vicon Tracker System	74
Figure 6.3	Mini-drone Rolling-Spider de Parrot	75
Figure 6.4	Position estimée en x du drone pour les différents contrôleurs	75
Figure 6.5	Trajectoires estimées du drone en x et y pour les différents contrôleurs	76
Figure 6.6	Comparaison de la position x estimée par le drone et par le Vicon . .	76
Figure 6.7	Comparaison de la trajectoire x et y estimée par le drone et par le Vicon	77
Figure 6.8	Comparaison des écarts-types en fonction de la vitesse d'estimation selon la méthode d'estimation de la position	78
Figure 6.9	Comparaison des puissances (en unité arbitraire) consommées par les moteurs des différents contrôleurs au cours du temps	79

LISTE DES SIGLES ET ABRÉVIATIONS

ADRC	Active Disturbance Rejection Controller
PID	Proportional Integer Derivative
LQR	Linear Quadratic Regulator
ESO	Estimated State Observer
LESO	Linear Estimated State Observer
NLESO	Non-Linear Estimated State Observer
HGO	High-Gain Observer
SEF	State Error Feedback
LSEF	Linear State Error Feedback
NLSEF	Non-Linear State Error Feedback
TD	Tracking Differentiator

CHAPITRE 1 INTRODUCTION

L'automatique traite de la modélisation, de l'analyse et de la commande des systèmes dynamiques. Cette science, qui se fonde sur les mathématiques et la théorie du signal, est très utilisée par les ingénieurs, qui, depuis sa naissance à la fin du XIX^e siècle¹, tentent de commander au mieux des systèmes dynamiques de complexités et de formes extrêmement variées. Les domaines scientifique et industriel ont toujours été en quête de meilleures performances, ce qui a permis un développement rapide de ce domaine, qui est aujourd'hui source d'innovations et de recherche intenses. Dans cette perspective, nous allons revisiter cette théorie de la commande afin de nous intéresser plus particulièrement à un type de contrôleur relativement récent et encore peu utilisé à ce jour, le rejet actif de perturbations (*Active Disturbance Rejection Control* - ADRC).

Pour ce faire, il convient d'introduire correctement le sujet en question. Ainsi, le début de l'utilisation des premiers contrôleurs nommés régulateurs PID (Proportionnel-Intégral-Dérivé), remonte à la période de l'entre-deux-guerres, avec bien sûr un pic de développement pendant la seconde guerre mondiale. Les régulateurs PID ont répondu à la demande croissante de l'industrie et de l'automatisme. Ce contrôleur classique est encore très largement utilisé de nos jours par sa facilité de mise en œuvre. Cependant, au début des années 80, la demande pour les contrôleurs commencent à changer et ceci mène à des théories du contrôle qui permettent de commander des systèmes dits incertains, où les systèmes dynamiques ne sont pas clairement définis. Ceci implique les bruits, ou encore des variables peu connues ou inconnues. C'est ainsi que plusieurs méthodes se sont développées, comme les méthodes robustes, telles que la synthèse \mathcal{H}_∞ (Doyle et al., 1989), ou des commandes non linéaires, comme la commande par mode de glissement (Edwards and Spurgeon, 1998). C'est à la fin des années 80 et au début des années 90 que Jingquin Han publie plusieurs articles sur une nouvelle méthode de contrôle non conventionnelle (Han, 1988, 1995a,b, 2009, 1989). Cette technique, baptisée plus tard *Active Disturbance Rejection Control*, a pour avantage de rassembler les deux types d'incertitudes ; le premier type d'incertitude est celle concernant les défauts de modélisation du système, nommée donc incertitude interne, et la deuxième rassemble celles dû aux bruits issus de l'environnement, nommée donc incertitude externe. Le regroupement des deux a amené J. Han à inventer le concept d'«incertitude totale», notion fondamentale de l'ADRC. Durant ces vingt dernières années, l'ADRC a pu être implémenté dans plusieurs systèmes de différentes natures tels que le contrôle de mouvement (Dong et al.,

1. Bien que l'on puisse trouver des exemples de systèmes automatiques dès l'Antiquité.

2016; Xia and Liu, 2014), le convertisseur de puissance AC-DC (Sun and Gao, 2005), des contrôles de réactions chimiques, et bien d'autres encore.

Plus récemment, un autre type de système a ouvert une nouvelle branche de l'automatisme : les drones. Développés d'abord pour des fins militaires, les drones ont rapidement conquis le domaine public pour des missions de reconnaissance, de surveillance, de transports puis de loisirs pour le grand public. Ce nouveau genre d'appareil s'est développé de manière spectaculaire ces dernières années. Grâce à leurs capacités de vols inégalées en environnement urbain et rural, les drones ont rapidement trouvé leur place sur le marché et ont nourrit les fantasmes de beaucoup d'industriels comme Amazon pour le transport de colis (Pandit and Poojari, 2014), ou pour la recherche en armement (Noone and Noone, 2015). Dans un contexte de demande toujours croissant, les acteurs de ce secteur doivent faire face à des exigences en termes de contrôle et de maniabilité de plus en plus complexes. Les contrôleurs de base comme le simple PID ne suffisent alors plus pour répondre aux attentes des utilisateurs, ce qui a permis aux chercheurs de développer de nouveaux types de contrôles toujours plus performants. La stabilité et la maniabilité de l'appareil sont devenus les principaux arguments de vente, et donc de recherche. C'est dans cet élan que s'inscrit cette maîtrise, qui a pour but d'adapter une technique de contrôle innovante qui a déjà fait ses preuves sur des systèmes complexes aux quadricoptères. La robustesse et l'adaptivité de ce type de contrôle garantit une meilleure stabilité pour des systèmes incertains, ainsi que de meilleures performances de vol.

Le mémoire est organisé de la façon suivante. Le **chapitre 2** est consacré à la revue de littérature, et le **chapitre 3** à la modélisation du quadricoptère, ainsi que celle du vent et des forces induites. Le **chapitre 4** décrit la synthèse des différents contrôleurs utilisés. On présente ensuite dans le **chapitre 5** les résultats obtenus en simulation, et dans le **chapitre 6**, nous comparons les mesures de performance de vol en laboratoire. On conclut enfin sur les résultats de ce travail dans le **chapitre 7**.

CHAPITRE 2 REVUE DE LITTÉRATURE

Nous présentons dans ce chapitre la revue de littérature sur l'ADRC. Afin de comprendre au mieux son fonctionnement, il convient de décrire le raisonnement donnant lieu à la naissance de cette nouvelle méthode. Nous ferons ensuite un état de l'art complet sur toutes les composantes de ce contrôleur. C'est pourquoi nous présenterons d'abord les éléments de manière individuelle et indépendante (à savoir le suiveur différentiel (*Tracking Differentiator*, ou TD), le retour d'erreur d'état (*State Error Feedback* ou SEF) et l'estimateur d'état étendu (*Extended State Observer* ou ESO), puis son fonctionnement global en décrivant les interactions de ces différents éléments.

2.1 La naissance de l'ADRC

Par une série d'articles scientifiques dans les années 80-90 (Han, 1989; Han and Wang, 1994; Han, 1995b, 1988, 1995a) J. Han donne naissance à une méthode de contrôle nommée plus tard l'ADRC. Dans son article «From PID to Active Disturbance Rejection Control», (Han, 2009), J. Han va retracer le parcours de cette méthode, de ses débuts à ses applications. Nous nous appuyons surtout sur cet article dans cette partie.

Afin de comprendre l'idée derrière cette loi de commande, il convient de suivre le raisonnement de J. Han à sa genèse. Dans son article (Han, 2009), il énumère les défauts du contrôle PID, qui sont les suivants :

- la dégradation due au bruit dans la partie dérivative du contrôleur ;
- la sur-simplification et la perte de performance dans la combinaison linéaire des termes P, I et D ;
- les complications qu'apportent le terme intégrale.

Ces défauts, bien que minimes dans la plupart des cas, peuvent s'avérer extrêmement handicapants pour certains systèmes dynamiques, notamment pour des systèmes non-linéaires, bruités et/ou incertains. À cette époque, la technique du PID est quasiment omniprésente dans l'industrie. Cependant, l'émergence de systèmes dynamiques de plus en plus complexes a démontré les limites du PID et a poussé les ingénieurs et les scientifiques à mieux comprendre les mécanismes sous-jacents aux contrôleurs PID. C'est pour ces raisons que la plupart des méthodes de contrôle innovantes ont été conçues, telle que la commande adaptative et robuste, dont le but est d'avoir un meilleur contrôle même avec des incertitudes. C'est aussi dans cette optique que J. Han a commencé la démarche qui a finalement abouti à l'ADRC. Pour compenser les défauts présentés ci-dessus des contrôleurs PID, J. Han proposa quatre

solutions permettant la conception d'un nouveau contrôleur :

- une équation différentielle simple utilisée comme profil de référence ;
- un suiveur différentiel tolérant au bruit ;
- un retour de sortie non linéaire ;
- une estimation de l'incertitude totale et son rejet.

Si l'on compare ces nouveautés aux autres contrôleurs connus, il est intéressant de constater que le contrôleur proportionnel intégral est en général adapté pour un système avec des perturbations constantes, que le contrôle adaptatif convient aux systèmes avec des paramètres incertains (modélisation approximative), et que le contrôle robuste gère les dynamiques incertaines (présence de bruit). De plus, comme souligné par B-Z. Guo et Z-L. Zhao dans leur article (Guo B Z, 2013), les contrôles robustes et par mode de glissement considèrent les pires perturbations possibles, et se révèlent donc moins efficaces que ce que l'on peut faire avec le contrôleur ADRC. Une autre étude (Gao et al., 2001a) compare un contrôleur PID et ADRC sur un problème de contrôle de mouvement. Il en ressort que l'ADRC a de bien meilleures performances pour des réglages similaires. Bien que l'étude ne les compare pas pour des réglages fins, on retiendra que l'ADRC a l'avantage par rapport au PID d'avoir de meilleures performances sans avoir à réaliser une étape de réglages chronophage. Pour résumer, ce contrôleur est plus simple d'utilisation (argument pertinent pour des fins commerciales) et plus tolérant aux perturbations et incertitudes du système.

De manière pratique, le contrôleur ADRC possède plusieurs composantes permettant de mettre en œuvre les solutions proposées par J. Han ci-dessus. Nous rappelons à ce stade que l'idée principale de cette méthode est de pouvoir connaître l'incertitude totale (=incertitudes internes + perturbations externes) du système, qui est estimée avec la sortie du système. L'ADRC est donc constitué de trois composantes. La première est un suiveur différentiel, ou *Tracking Differentiator* (TD) en anglais. Son rôle est d'extraire les dérivées du signal de référence, et est utilisé comme profil de référence par le système. La deuxième composante un observateur d'état étendu, *Extended State Observer* (ESO). Il s'agit de la partie la plus importante de l'ADRC, car c'est celle-ci qui va estimer l'incertitude totale, clé de voûte de cette méthode de contrôle. Enfin, la troisième composante est le retour d'erreur d'état, LSEF (*Linear State Error Feedback*) ou encore NLSEF (*Non-Linear State Error Feedback*), dépendamment de la nature de ce dernier. Cette partie est un héritage du PID, qui a été modifié (dans la plupart des ADRC) pour bénéficier de la puissance d'une combinaison non linéaire de la variable de sortie, de son intégrale et de sa dérivée.

Maintenant que nous connaissons mieux l'architecture interne de l'ADRC, son rôle et ses objectifs de performance, nous allons pouvoir rentrer plus dans les détails de son fonctionnement théorique. Nous allons donc, dans notre prochaine partie, approfondir le développement

nécessaire à la compréhension totale du système global.

2.2 Fonctionnement de l'ADRC

Afin de présenter au mieux les trois différentes composantes de l'ADRC (Fig. 2.1), nous allons leur consacrer une sous-partie chacun. Leurs rôles étant très différents et relativement indépendants les uns des autres, nous pourrions d'abord comprendre en détails leur fonctionnement individuel avant de conclure sur les propriétés globales du système dans une dernière sous-partie. Nous allons donc présenter d'abord le suiveur différentiel (TD), puis le retour d'erreur d'état (SEF), et enfin l'estimateur d'état étendu (ESO).

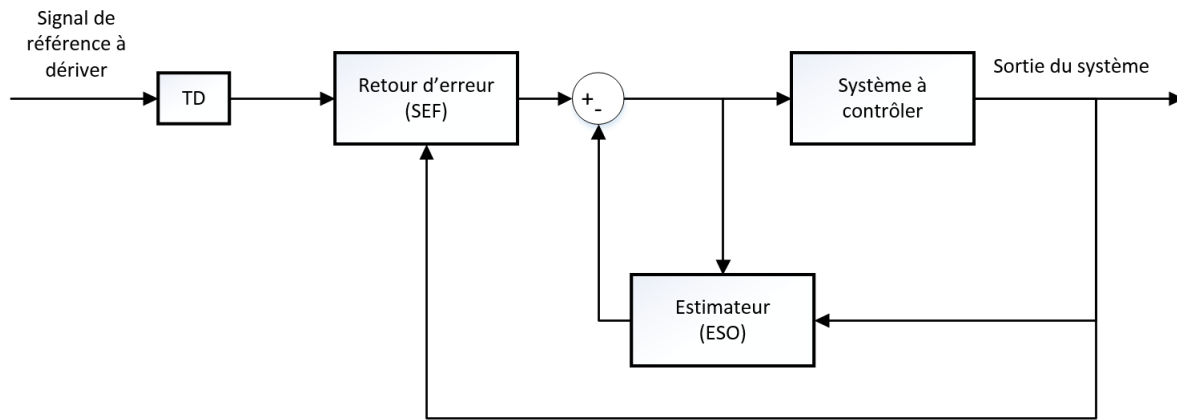


Figure 2.1 Graphe fonctionnel de l'ADRC comprenant les 3 composantes TD, SEF, et ESO

2.2.1 Suiveur différentiel

Cette composante a été proposée pour la première fois par J. Han en 1989 afin de répondre aux problèmes générés par le terme dérivatif dans le PID. Notamment lorsqu'il y a des bruits de mesure, la dérivation usuelle peut alors créer de grands écarts avec la valeur réelle. Le TD peut donc être considéré comme un dérivateur tolérant au bruit. C'est en s'inspirant de ses travaux (Han, 1988, 1999a) sur les dérivateurs non-linéaires que le TD va commencer à faire son apparition. Le principe du TD est contenu dans le théorème de J. Han publié dans les années 90 (Han and Wang, 1994) et a ensuite donné lieu à beaucoup de recherche en la matière (Han, 1999b; Guo and Zhao, 2011). Le théorème est énoncé ainsi, en prenant l'exemple d'un système d'ordre 2 :

Théorème 1 *Si la solution du système :*

$$\begin{cases} \dot{x}_1(t) = x_2(t) \\ \dot{x}_2(t) = f(x_1(t), x_2(t)) \end{cases} \quad (2.1)$$

est globalement asymptotiquement stable : $\lim_{t \rightarrow \infty} x_1(t) = 0$ et $\lim_{t \rightarrow \infty} x_2(t) = 0$ alors pour toute fonction mesurable bornée $v(t)$ et une constante $T > 0$ quelconque, la solution du système :

$$\begin{cases} \dot{z}_{1R}(t) = z_{2R}(t) \\ \dot{z}_{2R}(t) = R^2 f(z_{1R}(t) - v(t), \frac{z_{2R}(t)}{R}) \end{cases} \quad (2.2)$$

satisfait :

$$\lim_{R \rightarrow +\infty} \int_0^T |z_{1R}(t) - v(t)| dt = 0 \quad (2.3)$$

Concrètement, ce théorème nous permet de construire (sous condition de convergence de la solution du système 2.1) un estimateur des dérivées du signal c'est-à-dire que pour notre système dynamique (2.1), si la sortie de notre système est $v(t) = x_1(t)$, alors nous aurons $\lim_{R \rightarrow +\infty} z_{2R}(t) = \dot{v}(t)$. Nous pouvons donc obtenir la relation suivante en généralisant à un système d'ordre $n > 2$ (et avec une démonstration non précisée ici nécessitant la notion de dérivée généralisée de la théorie de distribution) :

$$\lim_{R \rightarrow +\infty} z_{iR}(t) = v^{(i-1)}(t) \quad (2.4)$$

On peut aussi utiliser le TD comme simple dérivateur d'un signal de référence (si besoin est). Par exemple, si le signal de référence est la position désirée du système, on peut tirer, grâce au TD, la vitesse instantanée désirée du système.

Afin de mieux saisir son fonctionnement, nous pouvons présenter un exemple de TD. Le cas le plus simple est celui d'un TD linéaire :

$$\begin{cases} \dot{z}_{1R}(t) = z_{2R}(t) \\ \dot{z}_{2R}(t) = -R^2(z_{1R}(t) - v(t)) - Rz_{2R}(t) \end{cases} \quad (2.5)$$

où $R > 0$ est un paramètre réglable. On reconnaît un système du type $\dot{\mathbf{Z}} = \mathbf{A}\mathbf{Z} + \mathbf{B}v$ où $\mathbf{A} = \begin{bmatrix} 0 & 1 \\ R^2 & R \end{bmatrix}$ et $\mathbf{B} = \begin{bmatrix} 0 \\ R^2 \end{bmatrix}$. En résolvant ce système linéaire, on obtient la convergence des variables d'état vers le signal de référence $v(t)$ (sous condition d'être continûment différentiable) et sa dérivée lorsque R est suffisamment grand. En pratique, R détermine la vitesse de convergence de cet estimateur de dérivée de signal de référence. Le TD a l'avantage d'être

peu exigeant envers ses conditions d'applications et d'être tolérant au bruit (ce qui n'a été démontré que dans le cas du TD linéaire (Guo and Zhao, 2011)). J. Han a aussi construit le TD pour les cas discrets (Han, 1999b) (qui ne sera pas développé ici). Les propriétés sont cependant similaires. Nous pouvons voir quelques exemples dans le livre de B-Z. Guo (Guo and Zhao, 2016) de TD linéaires et non-linéaires qui diffèrent légèrement dans leur application mais restent globalement similaires.

Nous retiendrons donc les propriétés énoncées par Z. Gao (Gao et al., 2001b), résumant ainsi le travail de J. Han. Ainsi, le TD agit comme un filtre anti-bruit, qui se révèle bien plus efficace que les filtres linéaires usuels. Il a en effet un décalage de phase bien plus faible que les filtres linéaires, et possède une réponse fréquentielle très appréciable. De plus, il possède un très bon rapport signal à bruit.

Le TD répond donc au problème soulevé par J. Han quant au terme dérivatif du PID cité en l'introduction. En effet, le PID est souvent utilisé comme un PI en pratique, car dériver un signal bruité produit de grandes erreurs par rapport à sa valeur réelle. Ce problème a donc été résolu grâce au TD présenté ci-dessus.

Nous allons à présent nous intéresser au retour d'erreur d'état, qui va traiter les signaux (signal de sortie et estimé des dérivées) sortant du TD. Nous privilégions ainsi une approche linéaire du système en suivant la boucle d'information dans l'ADRC. Il nous faut préciser que la composante TD n'est pas indispensable à l'ADRC, en effet nous pouvons trouver plusieurs exemples d'applications (Li et al., 2016) où l'ADRC ne contient pas de TD. Comme il nous sert à obtenir la dérivée d'un signal de référence, on peut quand même faire un contrôleur qui ne l'utilise pas (ce qui est la même chose que de faire un PI plutôt qu'un PID). Sa présence, bien que synonyme de meilleures performances, dépend donc du compromis entre qualité et complexité du contrôleur.

2.2.2 Retour d'erreur d'état

Si l'on poursuit le raisonnement tenu par J. Han et explicité en introduction de cette revue de littérature, nous arrivons à la deuxième composante du système : le retour d'erreur d'état (SEF). Le TD nous a permis d'obtenir la dérivée de notre signal de sortie, et nous allons l'exploiter avec ce composant. Nous pouvons conserver l'analogie faite avec le PID en introduction. Ainsi le TD joue le rôle du dérivateur et le SEF joue le rôle du pondérateur des termes PID. Habituellement, notre PID possède comme loi de commande :

$$u = K_p e + K_i \int e dt + K_d \dot{e} \quad (2.6)$$

où e désigne l'écart au signal de référence. Le PID est donc une combinaison linéaire des termes proportionnels, intégral et dérivé de e . J. Han a donc eu l'idée de faire une combinaison non-linéaire de ces termes (Han, 1994, 1995c) pour combler la perte de performance due à une combinaison linéaire. Un premier exemple est la loi de commande :

$$u = K_p |e_p|^{\alpha_p} \text{sign}(e_p) + K_i |e_i|^{\alpha_i} \text{sign}(e_i) + K_d |e_d|^{\alpha_d} \text{sign}(e_d) \quad (2.7)$$

où e_p désigne l'écart au signal de référence, e_i son intégrale et e_d sa dérivée. L'intérêt d'une telle méthode, lorsque $0 < \alpha < 1$, réside dans le fait que nous aurons de plus grands gains lorsque l'erreur est petite et de plus petits gains lorsque l'erreur est grande, ce qui intuitivement et en pratique fonctionne mieux, mais est plus compliqué à implémenter.

Pour améliorer les propriétés numériques de cet algorithme, J. Han, après avoir testé de nombreuses possibilités de mécanisme non-linéaire, suggéra la fonction fal pour remplacer les termes $|e|^\alpha \text{sign}(e)$:

$$\text{fal}(x, \alpha, \delta) = \begin{cases} |x|^\alpha \text{sign}(x), & |x| > \delta \\ \frac{x}{\delta^{1-\alpha}}, & |x| \leq \delta \end{cases} \quad (2.8)$$

En voici une représentation graphique (tirée de l'article de Z. Gao (Gao et al., 2001b)) permettant de voir que le gain est plus grand pour les petites erreurs, et que le gain est plus petit pour les erreurs plus grandes, comparativement à la combinaison linéaire :

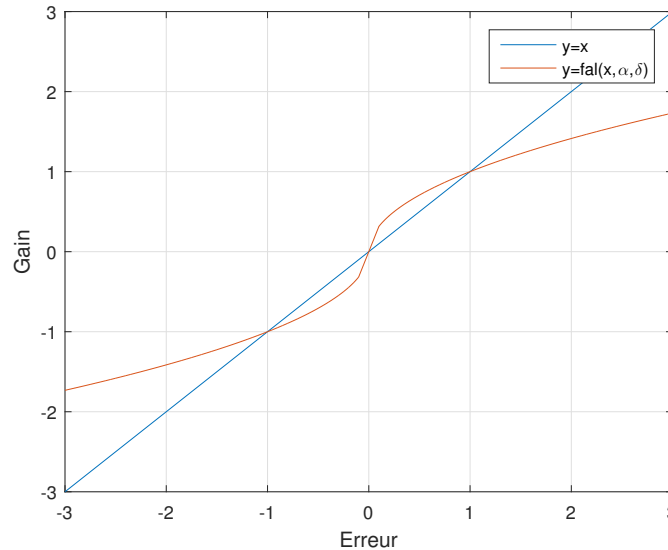


Figure 2.2 Comparaison des gains linéaires et non-linéaires pour $\alpha = 0.5$ et $\delta = 0.1$

Ce composant peut donc avoir des formes bien différentes selon les critères de performances

désirés. Il faut tout de même garder à l'esprit que plus la combinaison est complexe, plus l'implémentation risque d'être difficile. Il s'agit aussi ici, comme pour le TD, de trouver un compromis entre performance et complexité. La fonction suggérée par J. Han a tout de même largement été répandue pour sa bonne performance et sa simplicité (relative) d'implémentation. Nous pouvons d'ailleurs citer plusieurs exemples d'applications de l'ADRC où la combinaison des termes est linéaire (Li et al., 2016; Gao et al., 2001a; Xia and Liu, 2014), et non-linéaire (Gao et al., 2001b; Han, 1999a, 1988).

Après avoir vu les deux premières composantes, nous allons maintenant passer à la dernière, mais néanmoins la plus importante de ce contrôleur : l'ESO.

2.2.3 Estimateur d'état étendu

De manière générale, il est rare que l'on connaisse l'état du système en entier. Le plus souvent, on n'en connaît qu'une partie, grâce au signal de sortie. La technique usuelle pour connaître l'état du système complet est de construire un estimateur qui va estimer le reste de l'état du système depuis les données fournies par le signal de sortie. L'estimateur le plus répandu est celui de Luenberger, simple et efficace, dans lequel on peut choisir (via le gain de l'estimateur) la vitesse à laquelle l'estimateur converge vers l'état réel du système. Ceci marche très bien pour les systèmes linéaires mais de grands gains peuvent amplifier de manière significative l'erreur initiale de l'estimateur, créant de grands pics d'erreurs. Plusieurs solutions utilisant des gains non-linéaires ont donc été proposées pour contrer ce problème, comme l'estimateur par mode de glissement par exemple (Edwards and Spurgeon, 1998). En gardant à l'esprit l'idée de rassembler les perturbations internes et externes au système, J. Han a proposé un nouvel estimateur baptisé *Extended State Observer*. Beaucoup d'articles sont ensuite parus sur le sujet (Huang and Han, 2000; Xia et al., 2007). Pour un système SISO (une entrée et une sortie) d'ordre n (sous conditions que son degré relatif soit égale à n) tel que :

$$\begin{cases} \dot{x}_1(t) = x_2(t) \\ \dot{x}_2(t) = x_3(t) \\ \vdots \\ \dot{x}_n(t) = F(t, x_1, x_2, \dots, x_{n-1}(t)) + bu(t) + w(t) \\ y(t) = x_1(t) \end{cases} \quad (2.9)$$

où w représente les perturbations externes. Il est important de noter que la dynamique du système (qui se retrouve donc dans F) n'a pas besoin d'être explicitée. En effet, nous allons inclure cette dynamique dans la quantité à estimer. En agissant ainsi, nous compenserons

toutes les erreurs de modélisation qui auraient pu perturber la dynamique du système. La perturbation totale est donc définie par $F(t, \cdot) + w(t)$. L'idée de J. Han étant d'estimer cette quantité afin de l'annuler avec une loi de commande appropriée, nous allons en fait travailler sur un système équivalent :

$$\left\{ \begin{array}{l} \dot{x}_1(t) = x_2(t) \\ \dot{x}_2(t) = x_3(t) \\ \vdots \\ \dot{x}_n(t) = x_{n+1} + \bar{b}u(t) \\ \dot{x}_{n+1}(t) = h(t, x_1, x_2, \dots, x_n(t)) \\ y(t) = x_1(t) \end{array} \right. \quad (2.10)$$

où $x_{n+1} = w(t) + F(t, x_1, x_2, \dots, x_{n-1}(t)) + (b - \bar{b})u(t)$, et h est la dérivée de cette quantité. Cette variable d'état supplémentaire représente donc ce que nous appelons depuis le début l'incertitude totale. Nous verrons l'intérêt d'une telle extension dans le paragraphe suivant. L'estimateur de J. Han associé au système (2.10) est le suivant :

$$\left\{ \begin{array}{l} \dot{\hat{x}}_1(t) = \hat{x}_2(t) - g_1(\hat{x}_1(t) - y(t)) \\ \dot{\hat{x}}_2(t) = \hat{x}_3(t) - g_2(\hat{x}_1(t) - y(t)) \\ \vdots \\ \dot{\hat{x}}_n(t) = \hat{x}_{n+1}(t) - g_n(\hat{x}_1(t) - y(t)) + bu(t) \\ \dot{\hat{x}}_{n+1}(t) = -g_{n+1}(\hat{x}_1(t) - y(t)) \end{array} \right. \quad (2.11)$$

L'estimateur est donc de dimension $n+1$, car on souhaite estimer non seulement l'état complet du système (x_1, x_2, \dots, x_n) mais aussi estimer la dynamique incertaine plus les perturbations externes, d'où le terme d'estimateur étendue (*extended*). Une fois ces perturbations estimées, nous allons pouvoir les rejeter, d'où le terme *disturbance rejection* dans le nom ADRC. La loi de commande s'implémente donc de la manière suivante :

$$u(t) = -\frac{1}{\bar{b}}\hat{x}_{n+1} + u_0(t, \hat{x}_1, \dots, \hat{x}_n) \quad (2.12)$$

sous la condition que \bar{b} soit différent de 0.

Il paraît évident que la convergence de l'estimateur va dépendre très fortement du choix des fonctions g_i pour i allant de 1 à $n + 1$. Ce sujet a fait l'objet de nombreuses études, et il s'avère que pour des choix bien particuliers de fonctions, nous pouvons obtenir de très

bonnes performances de l'estimateur quant à sa robustesse et sa vitesse de convergence. On constate en général deux types de fonctions utilisées dans les articles traitant de l'ESO : soit des fonctions linéaires (facile à régler) soit des fonctions non-linéaires de la même forme que la fonction *fal* explicitée précédemment. Il nous faut tout de même préciser que ces fonctions particulières ont été trouvées de manière empirique, car les mathématiques sous-jacentes sont complexes. Par exemple, les démonstrations de convergence et de stabilité sur le sujet ont été très tardives par rapport à l'exploitation expérimentale. La convergence de l'ESO pour des systèmes non-linéaires incertains est prouvée par B-Z. Guo en 2011, la convergence de l'ADRC dans le cas linéaire MIMO est faite en 2011 (Xue and Huang, 2011) et enfin la démonstration de la convergence de l'ADRC a été faite en 2012 pour les systèmes non-linéaires SISO (Guo and Zhao, 2013) et en 2013 pour les systèmes non-linéaires MIMO (Guo B Z, 2013). Malgré cette complexité invisible pour l'utilisateur, il est tout de même relativement facile d'implémenter un tel observateur. Pour revenir aux fonctions g_i et leurs influences sur l'estimateur, il convient de donner quelques exemples de fonctions utilisées dans la littérature. Un de ceux qui reviennent le plus souvent sont les *Linear Extended State Observer* (LESO) où les fonctions g_i sont simplement des fonctions linéaires. On peut assimiler alors les g_i à de simples coefficients, des gains, et l'on retrouve souvent la même méthode pour établir des gains (Gao, 2006a,b; Li et al., 2016; Ma et al., 2016; Liu et al., 2013) qui permettent d'obtenir de bonnes performances. Elle consiste simplement à considérer un paramètre (la bande passante) qui nous donne directement tous les coefficients. Ainsi, en dimension 2 pour une bande passante de w_0 les trois fonctions g_i sont $g_1(e) = 3w_0e$, $g_2(e) = 3w_0^2e$ et $g_3(e) = w_0^3e$. On peut aussi citer l'article de Z. Gao qui utilise comme fonction g_i la fonction *fal* que l'on a vu précédemment (Gao et al., 2001a) ou encore celui de W. Dong (Dong et al., 2016) ou de Y. Huang (Y. Huang and Zhao, 2011). Nous noterons que le LESO est bien plus répandu, car très facile à implémenter. De plus la sélection des gains est très simple car elle ne dépend que de la bande passante désirée par l'utilisateur.

Maintenant que nous avons pu voir en détails les trois composantes de l'ADRC, nous allons consacrer une partie au fonctionnement global de l'ADRC.

2.2.4 Fonctionnement global

Afin de mieux comprendre l'agencement des trois composantes explicitées plus haut dans l'ADRC, nous pouvons nous en référer à la figure 2.1.

On peut résumer le fonctionnement du contrôleur ADRC en retraçant la conception de la commande. Tout d'abord, le TD va, soit depuis un signal de référence, soit depuis le signal de sortie (dépendamment de son rôle, sur le schéma il s'agit du signal de référence), générer

sa dérivée. Dans le premier cas, nous aurons accès au signal de référence et sa dérivée, comme par exemple la position désirée pour une trajectoire et sa vitesse, ce qui peut être utile dans le cas d'un contrôle de vitesse. Dans le second cas, nous aurons donc ainsi accès non seulement aux variables d'états donnés par la sortie, mais aussi à leurs dérivées, qui peuvent être indispensables à la commande (le terme D du PID par exemple). Les différences des estimées des variables d'état avec les signaux de référence (et ses dérivées) vont alors être injectées dans le SEF. Le SEF va alors nous permettre d'obtenir la commande en fonction de ces quantités, que ce soit obtenu par combinaison linéaire (comme dans le cas d'un PID), ou non-linéaire (NLSEF). Cette commande produite par la SEF va donc représenter la dynamique désirée du système. Cependant, les perturbations extérieures aux systèmes et les erreurs de modélisation (les incertitudes externes et internes) vont malheureusement perturber cette dynamique et nous n'aurons donc pas cette dynamique désirée. L'ESO va donc estimer en temps réel, la différence entre cette dynamique et la dynamique réelle. En corrigeant la commande sortante du SEF, nous allons obtenir la commande équivalente réalisant la dynamique souhaitée en compensant les incertitudes de notre système et les perturbations externes.

Il paraît important de remarquer que dans la plupart des cas de l'application de l'ADRC, ce schéma présenté est très souvent modifié. Le TD n'est pas tout le temps utilisé (dépendamment si l'on a besoin de la dérivée), il y a le choix d'utiliser un retour linéaire ou non-linéaire, tout comme utiliser un LESO ou régir la convergence de l'estimateur avec des fonctions non-linéaires. Il existe donc autant de contrôleurs ADRC que de systèmes dynamiques où l'on peut l'appliquer. Il s'agit donc de bien comprendre les spécificités de chaque composante afin que le contrôleur créé réponde au mieux aux performances désirés, que ce soit pour sa simplicité d'implémentation, rapidité, etc.

CHAPITRE 3 MODÉLISATION DU QUADRICOPTÈRE

La première étape afin de synthétiser une loi de commande consister à modéliser mathématiquement notre système. Nous allons donc créer un modèle mathématique du quadricoptère à l'aide des lois physiques bien connues régissant la dynamique du drone. Ce modèle permet donc de relier les grandeurs physiques du drone qui nous intéressent et que l'on observe (sa position et son orientation) aux vitesses de rotation des moteurs (qui représenteront notre commande du système). Ceci nous permettra d'établir des lois de contrôle à partir du modèle que nous avons établi.

3.1 Équations dynamiques

Il est nécessaire d'explicitier la dynamique du système afin de synthétiser le contrôleur. Nous pouvons voir les notations utilisées quant au repère sur la figure 3.1 utilisée par Tran (Tran et al., 2015) :

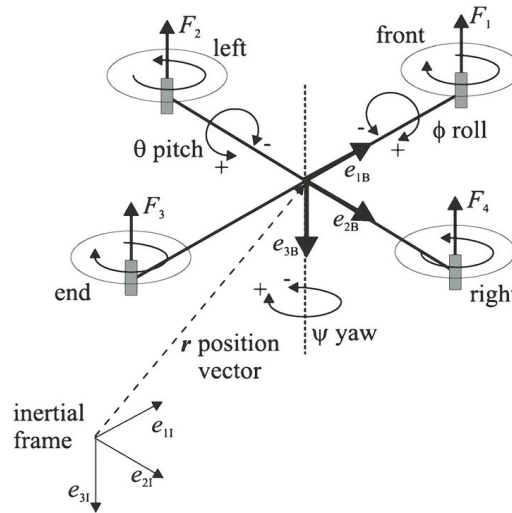


Figure 3.1 Représentation du système de coordonnées du quadricoptère (Tran et al., 2015)

Le passage du repère inertiel ($\vec{e}_{1I}, \vec{e}_{2I}, \vec{e}_{3I}$) au repère objet du quadricoptère ($\vec{e}_{1B}, \vec{e}_{2B}, \vec{e}_{3B}$) se fait au moyen de trois rotations successives décrites classiquement en aéronautique par les angles d'Euler ϕ, θ, ψ . On effectue une première rotation d'angle ψ autour de l'axe \vec{e}_{3I} , suivie d'une deuxième rotation d'angle θ autour de l'axe \vec{y} du repère intermédiaire, pour finir sur une troisième rotation d'angle ϕ autour de l'axe \vec{e}_{1B} . Ceci nous donne la matrice de rotation

$\mathbf{R}_{B/I}$ définie par :

$$\mathbf{R}_{B/I} = \begin{bmatrix} \cos \theta \cos \psi & \cos \theta \sin \psi & -\sin \theta \\ \sin \phi \sin \theta \cos \psi - \cos \phi \sin \psi & \sin \phi \sin \theta \sin \psi + \cos \phi \cos \psi & \sin \phi \cos \theta \\ \cos \phi \sin \theta \cos \psi + \sin \phi \sin \psi & \cos \phi \sin \theta \sin \psi - \sin \phi \cos \psi & \cos \phi \cos \theta \end{bmatrix} \quad (3.1)$$

On rappelle que pour une matrice de rotation $\mathbf{R}_{I/B} = \mathbf{R}_{B/I}^{-1} = \mathbf{R}_{B/I}^\top$. Dans le repère inertiel, l'application de la seconde loi du mouvement de Newton sur notre système donne :

$$\frac{d(m\vec{V})}{dt} = \vec{F} \quad (3.2)$$

où m désigne la masse du quadricoptère, \vec{V} son vecteur vitesse et \vec{F} les forces qui lui sont appliquées. Dans le cas d'un quadricoptère, chaque moteur i ($i = 1 \dots 4$), tournant à la vitesse ω_i , est équipé d'une hélice et exerce une force F_i sur le drone selon l'axe $-\vec{e}_{3B}$. On note u_1 leur résultante, de telle sorte que :

$$u_1 = \sum_{i=1}^4 F_i \quad (3.3)$$

et la force totale exercée par les quatre moteurs est $-u_1\vec{e}_{3B}$. Le quadricoptère est soumis à la force de gravité, donnée dans le repère inertiel par $mg_0\vec{e}_{3I}$. En projetant l'équation (3.2) sur les axes \vec{e}_{1I} , \vec{e}_{2I} et \vec{e}_{3I} , et en nommant x , y et z les coordonnées du centre de masse du quadricoptère, nous obtenons l'équation :

$$m \begin{bmatrix} \ddot{x} \\ \ddot{y} \\ \ddot{z} \end{bmatrix} = \begin{bmatrix} 0 \\ 0 \\ mg_0 \end{bmatrix} - \mathbf{R}_{I/B} \begin{bmatrix} 0 \\ 0 \\ u_1 \end{bmatrix} \quad (3.4)$$

soit sous forme scalaire :

$$\begin{cases} m\ddot{x} = -u_1(\cos \phi \sin \theta \cos \psi + \sin \phi \sin \psi) \\ m\ddot{y} = -u_1(\cos \phi \sin \theta \sin \psi - \sin \phi \cos \psi) \\ m\ddot{z} = mg_0 - u_1 \cos \phi \cos \theta \end{cases} \quad (3.5)$$

De façon similaire, la projection de l'équation du moment cinétique dans le repère objet donne :

$$\mathbf{J} \frac{d(\boldsymbol{\Omega})}{dt} + \mathbf{J}\boldsymbol{\Omega} \times \boldsymbol{\Omega} = \mathbf{M} \quad (3.6)$$

où $\mathbf{J} = \text{diag}(I_x, I_y, I_z)$ est la matrice d'inertie définie dans le repère objet¹, $\boldsymbol{\Omega}$ le vecteur vitesse de rotation du repère objet par rapport au repère inertiel, \times le produit vectoriel et \mathbf{M} la résultante des moments appliqués au drone exprimées dans le repère objet. Cette résultante comporte en général deux composantes principales : les moments induits par les rotors et les moments gyroscopiques, négligeables dans le cas présent. En observant la figure 3.1, et en notant L la distance des moteurs au centre de masse, on en déduit que les moteurs 2 et 4 exercent le moment $u_2 \vec{e}_{1B}$ où

$$u_2 = L(F_2 - F_4) \quad (3.7)$$

De façon similaire, les moteurs 1 et 3 exercent le moment $u_3 \vec{e}_{2B}$ où

$$u_3 = L(F_1 - F_3) \quad (3.8)$$

De plus, par application de la troisième loi de Newton, chaque moteur i exerce un moment de réaction M_i selon l'axe \vec{e}_{3B} ; le sens du moment est opposé au sens de rotation du moteur. Le moment total de réaction est alors $u_4 \vec{e}_{3B}$ où

$$u_4 = -M_1 + M_2 - M_3 + M_4 \quad (3.9)$$

En notant p , q et r les composantes dans le repère objet du vecteur vitesse de rotation $\boldsymbol{\Omega}$, l'équation (3.6) fournit le système d'équations scalaires (3.10) :

$$\begin{cases} I_x \dot{p} + (I_z - I_y)qr = u_2 \\ I_y \dot{q} + (I_x - I_z)pr = u_3 \\ I_z \dot{r} + (I_y - I_x)pq = u_4 \end{cases} \quad (3.10)$$

Pour compléter la modélisation dynamique, l'équation cinématique d'Euler reliant les angles d'Euler ϕ , θ , ψ aux vitesses angulaires p , q , r fournit le système (3.11) :

$$\begin{cases} \dot{\phi} = p + q \sin \phi \tan \theta + r \cos \phi \tan \theta \\ \dot{\theta} = q \cos \phi - r \sin \phi \\ \dot{\psi} = q \frac{\sin \phi}{\cos \theta} + r \frac{\cos \phi}{\cos \theta} \end{cases} \quad (3.11)$$

Avec les équations (3.5), (3.10) et (3.11), on obtient le système d'équations régissant la

1. Cette matrice est supposée diagonale du fait des symétries du quadricoptère.

dynamique du quadricoptère :

$$\left\{ \begin{array}{l} \ddot{x} = -\frac{u_1}{m}(\cos \phi \sin \theta \cos \psi + \sin \phi \sin \psi) \\ \ddot{y} = -\frac{u_1}{m}(\cos \phi \sin \theta \sin \psi - \sin \phi \cos \psi) \\ \ddot{z} = g_0 - \frac{u_1}{m} \cos \phi \cos \theta \\ \dot{\phi} = p + q \sin \phi \tan \theta + r \cos \phi \tan \theta \\ \dot{\theta} = q \cos \phi - r \sin \phi \\ \dot{\psi} = q \frac{\sin \phi}{\cos \theta} + r \frac{\cos \phi}{\cos \theta} \\ \dot{p} = qr \frac{I_y - I_z}{I_x} + \frac{u_2}{I_x} \\ \dot{q} = pr \frac{I_z - I_x}{I_y} + \frac{u_3}{I_y} \\ \dot{r} = pq \frac{I_x - I_y}{I_z} + \frac{u_4}{I_z} \end{array} \right. \quad (3.12)$$

Nous rappelons dans le tableau 3.1 toutes les notations utilisées jusqu'à présent :

Tableau 3.1 Tableau récapitulatif des notations

Notations	Signification
x, y, z	coordonnées du centre de masse du quadricoptère
p, q, r	vitesse angulaire
ϕ, θ, ψ	angles d'Euler
m	masse du quadricoptère
I_x, I_y, I_z	moment d'inertie selon \vec{e}_{1B} , \vec{e}_{2B} et \vec{e}_{3B}
F_i	force exercée par le moteur i
M_i	moment exercé par le moteur i
ω_i	vitesse de rotation du moteur i
g_0	constante de gravité
L	distance entre le centre de masse et les moteurs

Les entrées u_j vont nous servir de signaux de commande du système plutôt que les forces F_i et les moments M_i générés par chacun des moteurs, et ce, par souci de simplicité. En effet, l'utilisation des u_j permet un découplage naturel des axes. Comme nous le verrons dans la section 3.2 consacrée aux moteurs, les forces F_i et les moments induits M_i dépendent des vitesses de rotation des moteurs ω_i . Il est alors possible de trouver une transformation simple entre les u_j et les ω_i .

Dans l'optique de synthétiser nos trois contrôleurs (PID, LQR et ADRC), il nous est nécessaire de linéariser le système autour de la position d'équilibre qu'est le vol stationnaire

(*hovering state*). On considère un vol à une altitude z_0 . Dans ce cas, nous réécrivons les variables d'intérêt comme étant une valeur d'équilibre (indice 0) plus la déviation par rapport à cet équilibre (Δ) :

$$\left\{ \begin{array}{l} x = x_0 + \Delta x \\ y = y_0 + \Delta y \\ z = z_0 + \Delta z \\ p = p_0 + \Delta p \\ q = q_0 + \Delta q \\ r = r_0 + \Delta r \\ \phi = \phi_0 + \Delta \phi \\ \theta = \theta_0 + \Delta \theta \\ \psi = \psi_0 + \Delta \psi \\ u_1 = u_{10} + \Delta u_1 \\ u_2 = u_{20} + \Delta u_2 \\ u_3 = u_{30} + \Delta u_3 \\ u_4 = u_{40} + \Delta u_4 \end{array} \right. \quad (3.13)$$

La résolution du système d'équations (3.12) à l'équilibre (dérivées temporelles nulles) nous donne les conditions suivantes :

- Vitesses angulaires nulles : $p_0 = q_0 = r_0 = 0$;
- Angles d'Euler nuls : $\phi_0 = \theta_0 = 0 \Rightarrow$ le drone est à l'horizontal ;
- Position indifférente prise nulle : $x_0 = y_0 = 0$;
- Cap indifférent pris nul : $\psi_0 = 0$;
- Commandes en moment nulles : $u_2 = u_3 = u_4 = 0$;
- Commande en force : $u_1 = mg_0 \Rightarrow$ la force totale créée par les moteurs compense le poids.

Nous pouvons alors linéariser le système (3.12) autour de cet équilibre et obtenir ainsi :

$$\left\{ \begin{array}{l} \Delta \ddot{x} = -u_{10} \Delta \theta / m \\ \Delta \ddot{y} = u_{10} \Delta \phi / m \\ \Delta \ddot{z} = -\Delta u_1 / m \\ \Delta \dot{\phi} = \Delta p \\ \Delta \dot{\theta} = \Delta q \\ \Delta \dot{\psi} = \Delta r \\ \Delta \dot{p} = \Delta u_2 / I_x \\ \Delta \dot{q} = \Delta u_3 / I_y \\ \Delta \dot{r} = \Delta u_4 / I_z \end{array} \right. \quad (3.14)$$

ou sous la forme condensée classique :

$$\begin{cases} \Delta\ddot{x} = -g_0\Delta\theta \\ \Delta\ddot{y} = g_0\Delta\phi \\ \Delta\ddot{z} = -\Delta u_1/m \\ \Delta\ddot{\phi} = \Delta u_2/I_x \\ \Delta\ddot{\theta} = \Delta u_3/I_y \\ \Delta\ddot{\psi} = \Delta u_4/I_z \end{cases} \quad (3.15)$$

Un modèle d'état possible est alors :

$$\dot{\mathbf{x}} = \mathbf{Ax} + \mathbf{Bu} \quad (3.16)$$

où les vecteurs d'état et d'entrée sont respectivement :

$$\mathbf{x} = [\Delta x \ \Delta \dot{x} \ \Delta y \ \Delta \dot{y} \ \Delta z \ \Delta \dot{z} \ \Delta \phi \ \Delta \dot{\phi} \ \Delta \theta \ \Delta \dot{\theta} \ \Delta \psi \ \Delta \dot{\psi}]^\top \quad (3.17)$$

$$\mathbf{u} = [\Delta u_1 \ \Delta u_2 \ \Delta u_3 \ \Delta u_4]^\top \quad (3.18)$$

et les matrices de l'équation d'état sont :

$$\mathbf{A} = \begin{bmatrix} 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & -g_0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & g_0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \end{bmatrix} \quad \mathbf{B} = \begin{bmatrix} 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ -1/m & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 1/I_x & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 1/I_y & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1/I_z \end{bmatrix} \quad (3.19)$$

Ce système linéarisé va nous permettre de synthétiser les contrôleurs autour de cette position d'équilibre.

3.2 Les moteurs

Les expressions des forces F_i et des moments M_i en fonction des vitesses de rotation ω_i des moteurs sont données par :

$$F_i = k_F \omega_i^2 \quad (3.20)$$

et :

$$M_i = k_M \omega_i^2 \quad (3.21)$$

où k_F et k_M sont, respectivement, les constantes de forces et de moments. Il est alors possible à tout moment d'établir le lien entre les commandes u_j et les vitesses de rotation ω_i . On a ainsi :

$$\begin{bmatrix} u_1 \\ u_2 \\ u_3 \\ u_4 \end{bmatrix} = \begin{bmatrix} k_F & k_F & k_F & k_F \\ 0 & Lk_F & 0 & -Lk_F \\ Lk_F & 0 & -Lk_F & 0 \\ -k_M & k_M & -k_M & k_M \end{bmatrix} \begin{bmatrix} \omega_1^2 \\ \omega_2^2 \\ \omega_3^2 \\ \omega_4^2 \end{bmatrix} \quad (3.22)$$

ou de façon inverse :

$$\begin{bmatrix} \omega_1^2 \\ \omega_2^2 \\ \omega_3^2 \\ \omega_4^2 \end{bmatrix} = \begin{bmatrix} 1/(4k_F) & 0 & 1/(2Lk_F) & -1/(4k_M) \\ 1/(4k_F) & 1/(2Lk_F) & 0 & 1/(4k_M) \\ 1/(4k_F) & 0 & -1/(2Lk_F) & -1/(4k_M) \\ 1/(4k_F) & -1/(2Lk_F) & 0 & 1/(4k_M) \end{bmatrix} \begin{bmatrix} u_1 \\ u_2 \\ u_3 \\ u_4 \end{bmatrix} \quad (3.23)$$

La dynamique des moteurs est souvent approximée par un système d'ordre 1 (fonction de transfert du type : $\frac{1}{1+\tau s}$, τ étant le retard entre la vitesse de rotation désirée et réelle) mais nous considérerons qu'elle est suffisamment rapide pour être négligée.

3.3 Effet du vent

Dans cette section, nous expliquons tout d'abord comment modéliser le vent (et plus particulièrement sa vitesse en fonction de temps), les forces ainsi que les moments qu'il exercera sur le drone. Ce sont ces forces et ces moments qui joueront le rôle de « perturbation totale » qu'il nous faudra estimer dans le contrôleur ADRC. Il est donc important que cette modélisation soit la plus réaliste possible afin d'observer en simulation le comportement de notre contrôleur dans ces conditions. Il nous faut ajouter que ce sont les variations du vent qui vont nous intéresser (en effet estimer une force constante est bien plus facile que d'estimer un signal aléatoire). La modélisation du profil du vent se fera donc avec le modèle de Dryden (Dryden and Kuethe, 1931) et la modélisation des forces et moments exercées par le drone

se fera grâce au modèle de Khan (Khan and Nahon, 2015).

3.3.1 Modélisation du vent

La modélisation du vent est une étape cruciale pour les simulations, car nous avons besoin de représenter l'influence du vent le plus fidèlement possible. En effet, le vent n'a pas une vitesse constante mais est sujet à des variations que l'on peut modéliser grâce à des outils mathématiques particuliers que nous allons présenter dans cette partie. Après avoir expliqué le modèle permettant de simuler la vitesse du vent et ses variations au cours du temps, nous présenterons le modèle permettant d'obtenir les forces qui sont exercées sur le quadricoptère et ses rotors.

Pour cette modélisation nous allons nous baser sur l'article cité en introduction (Tran et al., 2015), qui reprend lui-même le modèle de Dryden (Dryden and Kuethe, 1931). Ce modèle nous donne les fonctions de densité spectrale de puissance suivante :

$$\begin{aligned} S_u(\Omega) &= \sigma_u^2 \frac{2L_u}{\pi} \frac{1}{1 + (L_u\Omega)^2} \\ S_v(\Omega) &= \sigma_v^2 \frac{2L_v}{\pi} \frac{1 + 12(L_v\Omega)^2}{(1 + 4(L_v\Omega)^2)^2} \\ S_w(\Omega) &= \sigma_w^2 \frac{2L_w}{\pi} \frac{1 + 12(L_w\Omega)^2}{(1 + 4(L_w\Omega)^2)^2} \end{aligned} \quad (3.24)$$

où u représente la direction selon laquelle souffle le vent, v et w correspondent aux deux autres axes de l'espace, $\sigma_{u,v,w}$ sont les intensités de turbulence, $L_{u,v,w}$ les longueurs caractéristiques et $\Omega = \frac{\omega}{V}$ les fréquences spatiales, avec V la vitesse moyenne du vent que l'on cherche à simuler. On peut donc obtenir les vitesses des vents grâce à des filtres linéaires associées aux équations (3.24) précédentes :

$$\begin{aligned} T_u(s) &= \sigma_u \sqrt{\frac{2L_u}{\pi V}} \frac{1}{1 + \frac{L_u}{V}s} \\ T_v(s) &= \sigma_v \sqrt{\frac{2L_v}{\pi V}} \frac{1 + \frac{2\sqrt{3}L_v}{V}s}{\left(1 + \frac{2L_v}{V}s\right)^2} \\ T_w(s) &= \sigma_w \sqrt{\frac{2L_w}{\pi V}} \frac{1 + \frac{2\sqrt{3}L_w}{V}s}{\left(1 + \frac{2L_w}{V}s\right)^2} \end{aligned} \quad (3.25)$$

avec V la vitesse moyenne du vent que l'on cherche à simuler. Ainsi en filtrant des bruits blancs, nous arrivons à obtenir des profils de vent réalistes (Fig. 3.2), pour une vitesse moyenne de 5 m/s selon l'axe x et les valeurs numériques (Tab. 3.2) données par l'ESDU (*Engineering Sciences Data Unit*).

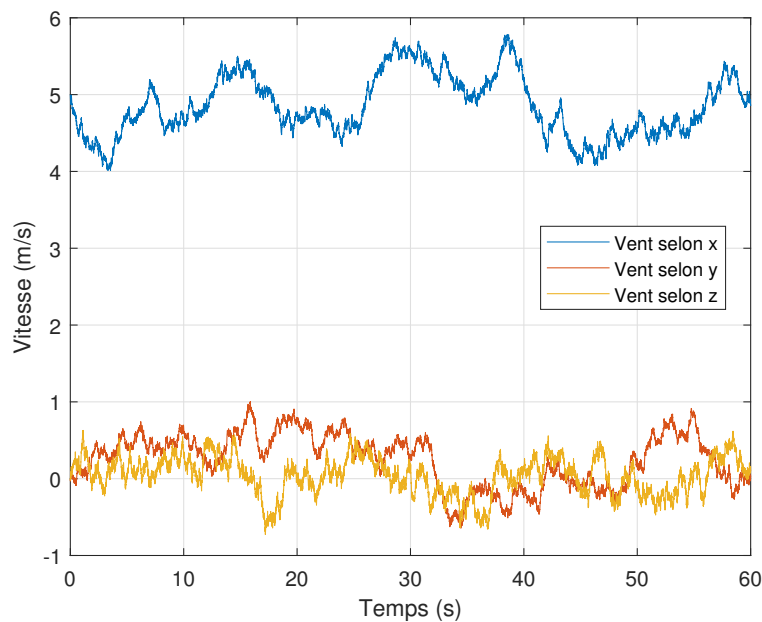


Figure 3.2 Vitesse du vent selon les 3 axes en fonction du temps

Tableau 3.2 Valeurs des constantes utilisées dans le modèle de Dryden

σ_u	1.1005
σ_v	0.6576
σ_w	0.4168
L_u	71.2063
L_v	21.4103
L_w	3.500

3.3.2 Forces et moments induits

Maintenant que nous avons réussi à obtenir la vitesse du vent en fonction du temps, nous devons étudier les forces et les moments induits par ce dernier sur le drone. L'influence principale du vent se manifeste tout d'abord sous forme de traînée. On considère ensuite l'effet du vent sur les rotors induisant des forces et des moments supplémentaires. Ces deux phénomènes seront donc traités indépendamment l'un de l'autre et superposés dans le modèle final.

Forces de traînée

La force de traînée est la force qu'exerce le vent sur l'ensemble de la structure du drone à l'exception des rotors. Cette force est de la forme :

$$F_{[x,y,z]} = \frac{1}{2} C_d \rho A_{[x,y,z]} V_{[x,y,z]} |V_{[x,y,z]}| \quad (3.26)$$

où C_d est le coefficient de traînée, ρ la masse volumique de l'air (1.225 kg/m^3), $A_{[x,y,z]}$ l'aire du drone face au vent selon les directions x , y et z et $V_{[x,y,z]}$ la vitesse du vent selon les mêmes directions.

La mesure du coefficient de traînée exacte du quadricoptère est complexe à réaliser. C'est pourquoi, pour les simulations, nous avons supposé que nous pouvions modéliser la force de traînée exercée sur le drone par une force équivalente à celle appliquée sur un cylindre de même dimension. Ainsi, pour les écoulements considérés, nous avons un nombre de Reynolds de l'ordre de 2×10^5 . Le coefficient de traînée pour un cylindre dans ces conditions est typiquement de l'ordre de l'unité. Nous avons utilisé pour nos simulations $C_d = 0.6$. Selon le drone considéré, il est possible de calculer l'aire de profil avec les données trouvées sur le site du constructeur. Par exemple, dans le cas du drone AscTec Pelican (Fig. 3.3), on trouve donc que la surface du drone exposée au vent est d'environ 0.12 m^2 selon les axes x et y . Nous prendrons le double de cette valeur pour l'axe z (on compte 2 bras exposés lorsque le drone est de profil, et 4 bras lorsqu'on le regarde du dessus).

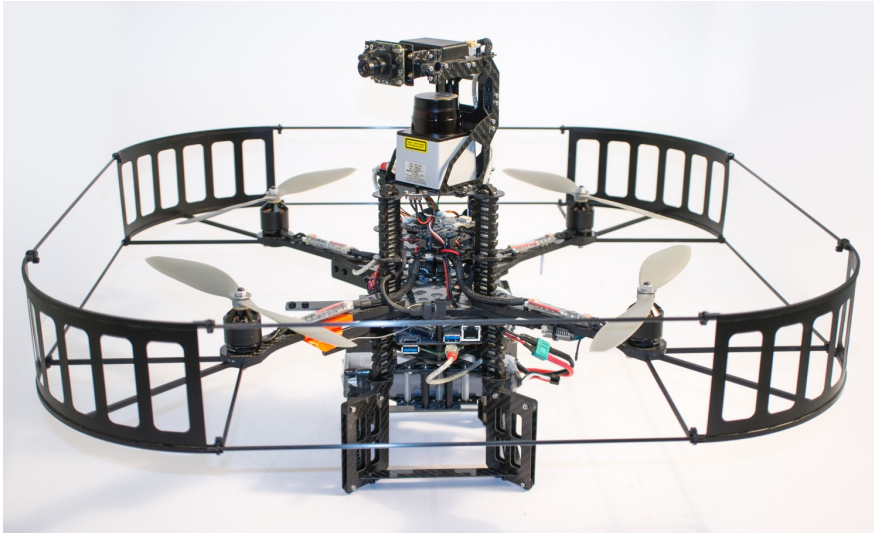


Figure 3.3 Drone AscTec Pelican ($651 \times 651 \times 188 \text{ mm}$)

Perturbations induites sur les rotors

La complexité du phénomène d'écoulement de l'air autour d'un rotor en mouvement a fait l'objet de beaucoup de recherche. Un des modèles les plus utilisés est celui de Khan (Khan and Nahon, 2015). Dans l'article (Tran et al., 2015), c'est aussi le modèle choisi afin de déterminer (par simulation du modèle) les forces et les moments exercées par le vent sur le rotor en fonction de la vitesse du vent et la vitesse de rotation du rotor. Les résultats sont ensuite stockés dans un tableau de référence (*look-up table*).

Cependant, pour cette étude, nous avons choisi de nous baser sur les résultats expérimentaux de l'article de Theys (Theys et al., 2014) qui nous donne les forces exercées par le vent sur un moteur en fonction de l'inclinaison du rotor, de la vitesse du vent et de la vitesse de rotation du rotor. Nous pouvons voir sur la figure 3.4 tirée de cet article (Theys et al., 2014) afin de mieux comprendre comment ont été effectués les mesures.

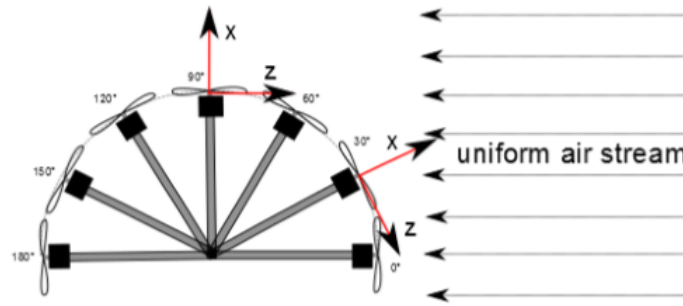


Figure 3.4 Expérience permettant de mesurer les coefficients $C_{F[x,y,z]}$ et $C_{M[x,y,z]}$ pour plusieurs angles d'attaque du vent (Theys et al., 2014)

Nous utilisons donc les coefficients définis dans (Theys et al., 2014) de la façon suivante dans l'article :

$$\begin{aligned} F_{[x,y,z]} &= C_{F[x,y,z]} \rho n^2 D^4 \\ M_{[x,y,z]} &= C_{M[x,y,z]} \rho n^2 D^5 \end{aligned} \quad (3.27)$$

où D est le diamètre du rotor, n le nombre de tours par secondes, ρ la masse volumique de l'air, $F_{[x,y,z]}$ et $M_{[x,y,z]}$ sont respectivement les forces et les moments exercés par le vent sur le rotor projetés sur les axes x , y et z tel que montré à la figure 3.4 (il est à noter que les axes sont inversés par rapport à notre modélisation du drone).

Nous obtenons ainsi les forces et les moments exercés par le vent sur les rotors en fonction de la vitesse de rotation des pales, la vitesse du vent et l'orientation des pales par rapport au vent (que l'on considérera constante comme l'on suppose que l'on reste proche de l'équilibre).

En regardant de près les coefficients mesurés, nous pouvons nous rendre compte que la compo-

sante dominante de la force totale est la poussée additionnelle. Cela veut dire que la principale influence du vent sur les rotors est l'apparition d'une force supplémentaire qui a tendance à faire augmenter l'attitude du drone.

Résultante des forces

Pour nos simulations, nous avons créé un bloc Simulink permettant de créer un profil de vent et d'en déduire les forces et moments perturbateurs sur le quadricoptère dépendamment de l'état (orientation et vitesse) de celui-ci.

Afin d'obtenir la résultante des forces totales dues au vent, nous avons additionné :

- la force selon x de traînée exercée par le vent selon x
- la force selon y de traînée exercée par le vent selon y
- la force selon z de traînée exercée par le vent selon z
- la force selon x exercée par le vent selon x sur le rotor
- la force selon y exercée par le vent selon x sur le rotor
- la force selon z exercée par le vent selon x sur le rotor

Ce qui implique que nous avons négligé :

- la force selon x exercée par le vent selon y sur le rotor
- la force selon y exercée par le vent selon y sur le rotor
- la force selon z exercée par le vent selon y sur le rotor
- la force selon x exercée par le vent selon z sur le rotor
- la force selon y exercée par le vent selon z sur le rotor
- la force selon z exercée par le vent selon z sur le rotor

Nous avons utilisé un raisonnement analogue pour la résultante des moments. Les forces et moments que nous avons négligés sont d'amplitudes bien inférieures à ceux conservés.

CHAPITRE 4 SYNTHÈSE DES CONTRÔLEURS

Le modèle dynamique étant maintenant établi, nous pouvons synthétiser les lois de commande permettant de maintenir une position de référence, malgré la présence du vent. Le critère comparatif entre nos trois différents contrôleurs sera l'écart-type en position, ce qui qualifie la capacité du drone à compenser les effets perturbateurs du vent. Ce chapitre est constitué de trois sections, chacune d'elles décrivant précisément la manière de synthétiser les contrôleurs LQR, PID et ADRC. La synthèse de l'ADRC est aussi accompagnée d'une optimisation de ce contrôleur.

4.1 Contrôleur LQR

La synthèse du contrôleur LQR nominal est décrite dans l'article (Tran et al., 2015), qui se base sur la résolution classique d'un problème de commande optimale. La synthèse LQR consiste à calculer un retour d'état $\mathbf{u} = -\mathbf{K}\mathbf{x}$ afin de minimiser un critère quadratique \mathcal{J} où, grâce aux matrices symétriques positives \mathbf{Q} et \mathbf{R} , on pondère à la fois le vecteur d'état \mathbf{x} et le vecteur de commande \mathbf{u} du système :

$$\mathcal{J}(\mathbf{u}) = \int_0^\infty \mathbf{x}^\top \mathbf{Q} \mathbf{x} + \mathbf{u}^\top \mathbf{R} \mathbf{u} dt \quad (4.1)$$

On rappelle que le système linéarisé (3.15) est sous la forme du modèle d'état (3.16) :

$$\begin{cases} \dot{\mathbf{x}} = \mathbf{A}\mathbf{x} + \mathbf{B}\mathbf{u} \\ \mathbf{y} = \mathbf{C}\mathbf{x} \end{cases} \quad (4.2)$$

avec $\mathbf{y} = [\Delta x \ \Delta y \ \Delta z \ \Delta \psi]^\top$ le vecteur de sortie d'intérêt et \mathbf{C} la matrice 4×12 correspondante. Afin d'assurer une meilleure précision sur le contrôle en position et en cap, on rajoute un intégrateur sur le vecteur d'erreur $\mathbf{e} = \dot{\boldsymbol{\xi}}$ défini par :

$$\mathbf{e} = \dot{\boldsymbol{\xi}} = \mathbf{r} - \mathbf{y} = \begin{bmatrix} \Delta x_{\text{ref}} - \Delta x \\ \Delta y_{\text{ref}} - \Delta y \\ \Delta z_{\text{ref}} - \Delta z \\ \Delta \psi_{\text{ref}} - \Delta \psi \end{bmatrix} \quad (4.3)$$

où $\mathbf{r} = [\Delta x_{\text{ref}} \ \Delta y_{\text{ref}} \ \Delta z_{\text{ref}} \ \Delta \psi_{\text{ref}}]^\top$ désigne le vecteur d'entrée de référence et $\boldsymbol{\xi}$ le vecteur d'état de l'intégrateur, tel que décrit dans (Tran et al., 2015), qui se base lui-même sur le

design d'un servomécanisme décrit dans (Young and Willems, 1972). Conformément à la figure 4.1 donnée dans l'article de référence, on cherche donc une loi de commande du type :

$$\mathbf{u} = -\mathbf{K}_{lqr}\mathbf{x} + \mathbf{K}_i\dot{\boldsymbol{\xi}} \quad (4.4)$$

ce qui donne en boucle fermée :

$$\begin{bmatrix} \dot{\mathbf{x}} \\ \dot{\boldsymbol{\xi}} \end{bmatrix} = \begin{bmatrix} \mathbf{A} - \mathbf{BK}_{lqr} & \mathbf{BK}_i \\ -\mathbf{C} & \mathbf{0}_{4 \times 4} \end{bmatrix} \begin{bmatrix} \mathbf{x} \\ \boldsymbol{\xi} \end{bmatrix} + \begin{bmatrix} \mathbf{0}_{12 \times 4} \\ \mathbf{I}_4 \end{bmatrix} \mathbf{r} \quad (4.5)$$

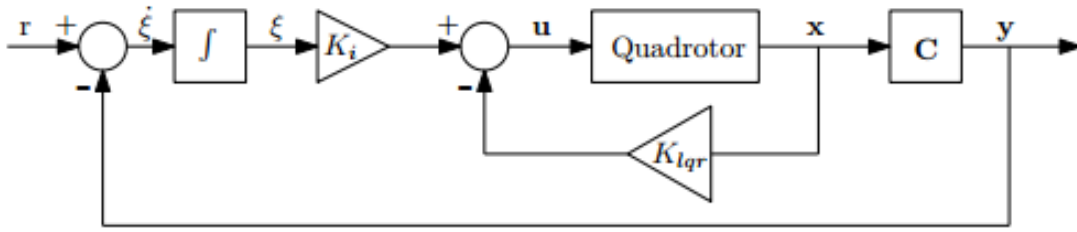


Figure 4.1 Représentation du contrôle LQR avec action intégrale (Tran et al., 2015)

Afin de synthétiser la loi de commande LQR, nous travaillons dans l'espace d'état augmenté de l'état de l'intégrateur, soit les nouvelles matrices \mathbf{A}_a et \mathbf{B}_a définies par :

$$\begin{cases} \mathbf{A}_a = \begin{bmatrix} \mathbf{A} & \mathbf{0}_{12 \times 4} \\ -\mathbf{C} & \mathbf{0}_{4 \times 4} \end{bmatrix} \\ \mathbf{B}_a = \begin{bmatrix} \mathbf{B} \\ \mathbf{0} \end{bmatrix} \end{cases} \quad (4.6)$$

et le vecteur d'état augmenté associé $\mathbf{x}_a = \begin{bmatrix} \mathbf{x}^\top & \boldsymbol{\xi}^\top \end{bmatrix}^\top$. Le critère quadratique à minimiser est alors :

$$\mathcal{J}(\mathbf{u}) = \int_0^\infty \mathbf{x}_a^\top \mathbf{Q}_a \mathbf{x}_a + \mathbf{u}^\top \mathbf{R} \mathbf{u} \, dt \quad (4.7)$$

Dans ce nouvel espace, le problème LQR se résout via l'équation de Riccati associée :

$$\mathbf{A}_a^\top \mathbf{P} + \mathbf{P} \mathbf{A}_a - \mathbf{P} \mathbf{B}_a \mathbf{R}^{-1} \mathbf{B}_a^\top \mathbf{P} + \mathbf{Q}_a = 0 \quad (4.8)$$

La solution \mathbf{P} de cette équation de Riccati nous permet d'obtenir directement la matrice de gain LQR recherchée :

$$\mathbf{K}_a = \mathbf{R}^{-1} \mathbf{B}_a^\top \mathbf{P} \quad (4.9)$$

Finalement, pour notre problème LQR avec action intégrale, la matrice de gains se décompose de la façon suivante :

$$\mathbf{K}_a = \begin{bmatrix} \mathbf{K}_{\text{lqr}} & -\mathbf{K}_i \end{bmatrix} \quad (4.10)$$

Pour les matrices de pondération, nous reprenons les valeurs utilisées dans (Tran et al., 2015) :

$$\mathbf{Q}_a = \text{diag} \left(10^4, 10^2, 10^4, 10^2, 10^4, 10^2, 1, 1, 1, 1, 1, 1, 10^3, 10^3, 10^3, 10^3 \right) \quad (4.11)$$

$$\mathbf{R} = \text{diag} \left(10^2, 10^2, 10^2, 10^2 \right) \quad (4.12)$$

Sous MATLAB, la résolution de l'équation de Riccati nous fournit la solution :

$$\mathbf{K}_a^\top = \begin{bmatrix} 0 & 0 & -11.2195 & 0 \\ 0 & 0 & -4.0915 & 0 \\ 0 & 11.2195 & 0 & 0 \\ 0 & 4.0915 & 0 & 0 \\ 11.8599 & 0 & 0 & 0 \\ 6.4283 & 0 & 0 & 0 \\ 0 & 7.0859 & 0 & 0 \\ 0 & 0.7252 & 0 & 0 \\ 0 & 0 & 7.0859 & 0 \\ 0 & 0 & 0.7252 & 0 \\ 0 & 0 & 0 & 1.6094 \\ 0 & 0 & 0 & 0.4080 \\ 0 & 0 & 3.1623 & 0 \\ 0 & -3.1623 & 0 & 0 \\ -3.1623 & 0 & 0 & 0 \\ 0 & 0 & 0 & -3.1623 \end{bmatrix} \quad (4.13)$$

4.2 Contrôleur PID

Le contrôleur PID donne une commande qui est une combinaison linéaire de l'erreur, de sa dérivée et de son intégrale. La figure 4.2 nous montre la structure d'un PID classique pour contrôler la position \mathbf{X} d'un système. On suppose que l'on mesure la position du système ainsi que sa vitesse.

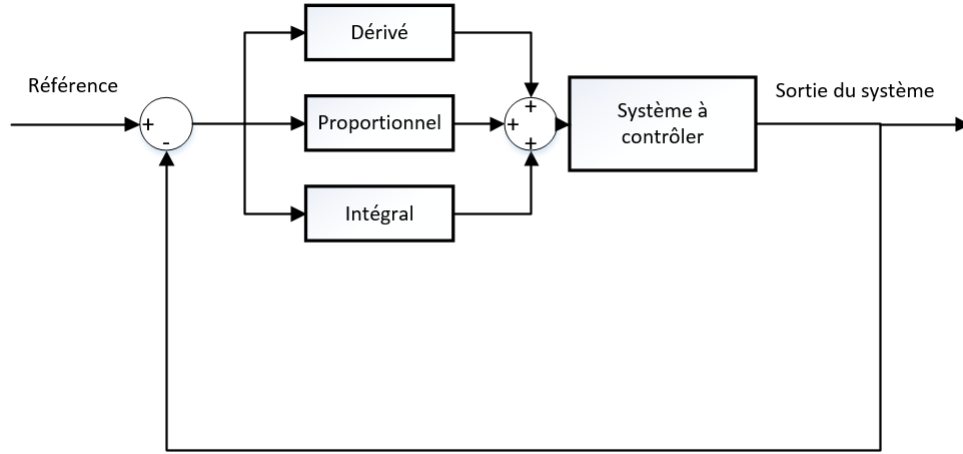


Figure 4.2 Structure PID classique

Afin de contrôler la position du drone, il convient de faire deux boucles de contrôle : une boucle bas niveau, qui va permettre de contrôler l'attitude du quadricoptère (ϕ, θ, ψ) , et une boucle haut niveau, qui va contrôler sa position (x, y, z) . Un quadricoptère étant un système sous-actionné, les boucles de commande des positions x et y pilotent les boucles de commande en ϕ et θ . Ainsi, le contrôle haut niveau va nous donner les angles d'Euler désirés qui vont permettre de corriger la position du drone, et le contrôle bas niveau va commander les vitesses de rotation des rotors afin que les angles d'Euler suivent précisément les angles d'Euler désirés. Les boucles de commande en z et en ψ agissent directement sur les vitesses de rotation des moteurs. Nous définissons donc ici les erreurs en position :

$$\begin{bmatrix} \Delta x \\ \Delta y \\ \Delta z \end{bmatrix} = \begin{bmatrix} x_{\text{ref}} - x \\ y_{\text{ref}} - y \\ z_{\text{ref}} - z \end{bmatrix} \quad (4.14)$$

et en attitude :

$$\begin{bmatrix} \Delta \phi \\ \Delta \theta \\ \Delta \psi \end{bmatrix} = \begin{bmatrix} \phi_{\text{ref}} - \phi \\ \theta_{\text{ref}} - \theta \\ \psi_{\text{ref}} - \psi \end{bmatrix} \quad (4.15)$$

où $[x_{\text{ref}}, y_{\text{ref}}, z_{\text{ref}}]^T$ est la position de référence désirée et $[\phi_{\text{ref}}, \theta_{\text{ref}}, \psi_{\text{ref}}]^T$ l'attitude désirée.

Les lois de commande correspondantes sont alors :

$$\begin{cases} \Delta u_1 = -k_{z_p}\Delta z - k_{z_d}\Delta \dot{z} - k_{z_i} \int \Delta z dt \\ \Delta u_2 = k_{\phi_p}\Delta \phi + k_{\phi_d}p \\ \Delta u_3 = k_{\theta_p}\Delta \theta + k_{\theta_d}q \\ \Delta u_4 = k_{\psi_p}\Delta \psi + k_{\psi_d}r \end{cases} \quad (4.16)$$

On choisit ensuite les angles d'Euler désirés en fonction des erreurs de position :

$$\begin{cases} \phi_{\text{ref}} = k_{y_p}\Delta y + k_{y_d}\Delta \dot{y} + k_{y_i} \int \Delta y dt \\ \theta_{\text{ref}} = -k_{x_p}\Delta x - k_{x_d}\Delta \dot{x} - k_{x_i} \int \Delta x dt \end{cases} \quad (4.17)$$

Il faut noter que les lois de commande en x et en y ci-dessus ne sont valides que pour $\psi_{\text{ref}} = 0$. En effet, dès que $\psi_{\text{ref}} \neq 0$, il faut opérer une transformation dépendant de ψ pour transformer les commandes des boucles en x et en y en commande en ϕ_{ref} et θ_{ref} adéquates. Comme nous réaliserons nos simulations avec le même quadricoptère, nous utiliserons les valeurs numériques du tableau 4.1 tirées de l'article de référence (Tran et al., 2015).

Tableau 4.1 Valeurs des gains

Coefficient	Valeur
k_{x,y_p}	1.5
k_{x,y_d}	0.4
k_{x,y_i}	0.1
k_{z_p}	10
k_{z_d}	6
k_{z_i}	1
k_{ϕ,θ_p}	5
k_{ϕ,θ_d}	0.3
k_{ψ_p}	0.04
k_{ψ_d}	0.04

4.3 ADRC

L'*Active Disturbance Rejection Controller* est une technique de contrôle relativement récente conçue par J. Han dans les années 90 (Han, 2009) et qui a continué à se développer jusqu'à aujourd'hui (Han, 1995a; Huang et al., 2001; Han, 2008; Chenlu et al., 2016).

4.3.1 Estimateur linéaire

Le contrôleur ressemble beaucoup à un PID classique, à ceci près que l'on rajoute un estimateur qui va estimer la «perturbation totale» du système, que ces perturbations viennent du modèle (perturbations internes) ou de l'environnement (perturbations externes). Bien que ce contrôleur ait été imaginé dans les années 90, les preuves de convergence et de stabilité n'ont été établi que très récemment (Guo and Zhao, 2011, 2013), ce qui témoigne de la complexité théorique sous-jacente et de l'intérêt toujours d'actualités que la communauté scientifique lui porte.

On reprend donc le design entier du contrôleur PID explicité ci-dessus, et, dans un premier temps, on rajoute un estimateur dans la boucle de contrôle bas niveau, c'est-à-dire pour le contrôle d'attitude. Les équations régissant la dynamique de l'attitude du quadricoptère peuvent s'écrire de la même manière qu'un système linéaire classique à deux dimensions (Han, 1995a) :

$$\begin{cases} \dot{\mathbf{x}}_1 = \mathbf{x}_2 \\ \dot{\mathbf{x}}_2 = \mathbf{x}_3 + \mathbf{B}\mathbf{u} \\ \dot{\mathbf{x}}_3 = \mathbf{h}(t) \end{cases} \quad (4.18)$$

où

$$\mathbf{x}_1 = [\Delta\phi \quad \Delta\theta \quad \Delta\psi]^\top \quad (4.19)$$

$$\mathbf{x}_2 = [\Delta\dot{\phi} \quad \Delta\dot{\theta} \quad \Delta\dot{\psi}]^\top \quad (4.20)$$

$$\mathbf{x}_3 = [x_{3,\phi} \quad x_{3,\theta} \quad x_{3,\psi}]^\top \quad (4.21)$$

$$\mathbf{h} = [h_\phi \quad h_\theta \quad h_\psi]^\top \quad (4.22)$$

$$\mathbf{u} = [\Delta u_2 \quad \Delta u_3 \quad \Delta u_4]^\top \quad (4.23)$$

$$\mathbf{B} = \begin{bmatrix} 1/I_x & 0 & 0 \\ 0 & 1/I_y & 0 \\ 0 & 0 & 1/I_z \end{bmatrix} \quad (4.24)$$

où \mathbf{x}_3 représente la « perturbation totale » à estimer selon ϕ , θ et ψ , en particulier les moments perturbateurs dus au vent, et \mathbf{h} sa dérivée. On construit donc un estimateur d'état

étendu linéaire (*Linear Extended State Observer* - LESO) :

$$\begin{cases} \mathbf{e} = \mathbf{x}_1 - \hat{\mathbf{x}}_1 \\ \dot{\hat{\mathbf{x}}}_1 = \hat{\mathbf{x}}_2 + \beta_1 \mathbf{e} \\ \dot{\hat{\mathbf{x}}}_2 = \hat{\mathbf{x}}_3 + \beta_2 \mathbf{e} + \mathbf{B}\mathbf{u} \\ \dot{\hat{\mathbf{x}}}_3 = \beta_3 \mathbf{e} \end{cases} \quad (4.25)$$

L'estimateur $\hat{\mathbf{x}}_3$ (s'il est bien construit) va suivre de manière précise la perturbation totale \mathbf{x}_3 , ce qui nous permettra de la rejeter dans la commande. Ce type d'estimateur est aussi appelé estimateur à grand gain (*High-Gain Observer* - HGO), qui fut l'objet d'études auparavant (Bornard and Hammouri, 1991; Atassi and Khalil, 1999).

On peut construire un LESO pour chaque angle d'Euler du fait que, sur le modèle linéaire, les trois axes de rotation sont découplés. Par exemple, pour la rotation en ϕ , nous avons le modèle linéaire :

$$\dot{\mathbf{x}}_\phi = \mathbf{A}_\phi \mathbf{x}_\phi + \mathbf{B}_\phi \Delta u_2 + \mathbf{E}_\phi h_\phi \quad (4.26)$$

où

$$\mathbf{x}_\phi = [\Delta\phi \quad \Delta\dot{\phi} \quad x_{3,\phi}]^\top \quad (4.27)$$

$$\mathbf{A}_\phi = \begin{bmatrix} 0 & 1 & 0 \\ 0 & 0 & 1 \\ 0 & 0 & 0 \end{bmatrix} \quad (4.28)$$

$$\mathbf{B}_\phi = \begin{bmatrix} 0 \\ 1/I_x \\ 0 \end{bmatrix} \quad (4.29)$$

$$\mathbf{E}_\phi = \begin{bmatrix} 0 \\ 0 \\ 1 \end{bmatrix} \quad (4.30)$$

Le LESO correspondant s'écrit alors :

$$\dot{\hat{\mathbf{x}}}_\phi = \mathbf{A}_\phi \hat{\mathbf{x}}_\phi + \mathbf{B}_\phi \Delta u_2 + \mathbf{L}\mathbf{C}(\mathbf{x}_\phi - \hat{\mathbf{x}}_\phi) \quad (4.31)$$

avec $\mathbf{L} = [\beta_1 \quad \beta_2 \quad \beta_3]^\top$ et $\mathbf{C} = [1 \quad 0 \quad 0]$. En combinant (4.26) et (4.31), la dynamique de l'erreur $\mathbf{e}_\phi = \mathbf{x}_\phi - \hat{\mathbf{x}}_\phi$ est alors :

$$\dot{\mathbf{e}}_\phi = \mathbf{A}_e \mathbf{e}_\phi + \mathbf{d} \quad (4.32)$$

où $\mathbf{d}_\phi = \mathbf{E}_\phi \mathbf{h}_\phi$ est un vecteur borné dépendant de la position (on suppose que l'on est stable)

et :

$$\mathbf{A}_e = \mathbf{A}_\phi - \mathbf{L}\mathbf{C} = \begin{bmatrix} -\beta_1 & 1 & 0 \\ -\beta_2 & 0 & 1 \\ -\beta_3 & 0 & 0 \end{bmatrix} \quad (4.33)$$

On choisit alors la fréquence de coupure ω_0 de la manière suivante (Gao, 2006a) :

$$\mathbf{L} = \begin{bmatrix} \beta_1 \\ \beta_2 \\ \beta_3 \end{bmatrix} = \begin{bmatrix} 3\omega_0 \\ 3\omega_0^2 \\ \omega_0^3 \end{bmatrix} \quad (4.34)$$

qui nous permet de placer le pôle triple ω_0 pour \mathbf{A}_e , et désigne ainsi la rapidité de la convergence de l'estimateur. Il ne nous reste plus qu'à rejeter l'estimé de la perturbation $\hat{\mathbf{x}}_3$ en corrigeant la commande de la même manière que J. Han (Han, 1995a) :

$$\mathbf{u}_{\text{final}} = \mathbf{B}^{-1}(\mathbf{u}_{\text{ref}} - \hat{\mathbf{x}}_3) \quad (4.35)$$

Cependant, la perturbation principale qui va jouer un rôle important sur la position est surtout la résultante des forces du vent plutôt que les moments. En effet, on n'observe pas d'amélioration significative sur le contrôle de position en ajoutant un LESO sur le contrôle bas niveau. On rajoute donc aussi un LESO sur la boucle de contrôle de position. Ce second LESO prend la même forme que le premier explicité dans (4.18), avec les changements suivants :

$$\mathbf{x}_1 = [\Delta x \quad \Delta y \quad \Delta z]^\top \quad (4.36)$$

$$\mathbf{x}_2 = [\Delta \dot{x} \quad \Delta \dot{y} \quad \Delta \dot{z}]^\top \quad (4.37)$$

$$\mathbf{x}_3 = [x_{3,x} \quad x_{3,y} \quad x_{3,z}]^\top \quad (4.38)$$

$$\mathbf{h} = [h_x \quad h_y \quad h_z]^\top \quad (4.39)$$

$$\mathbf{u} = [\Delta \phi_{\text{ref}} \quad \Delta \theta_{\text{ref}} \quad \Delta u_1]^\top \quad (4.40)$$

$$\mathbf{B} = \begin{bmatrix} 0 & -g_0 & 0 \\ g_0 & 0 & 0 \\ 0 & 0 & 1/m \end{bmatrix} \quad (4.41)$$

Il est à noter que si nous nous plaçons dans le cas où $\psi \neq 0$, il nous faut changer la matrice

B de sorte que :

$$\mathbf{B} = \begin{bmatrix} g_0 \sin \psi & -g_0 \cos \psi & 0 \\ g_0 \cos \psi & g_0 \sin \psi & 0 \\ 0 & 0 & 1/m \end{bmatrix} \quad (4.42)$$

Le seul paramètre à régler est donc ω_0 . Plus on le choisit grand, plus vite l'estimateur convergera, mais cela augmentera les risques d'instabilité. On fera donc plusieurs essais en fonction de ω_0 pour voir lequel donne les résultats les plus satisfaisants.

4.3.2 Estimateur non-linéaire

Afin d'essayer d'améliorer l'ESO (*Extended State Observer*), on peut utiliser des fonctions non linéaires (Han, 1994, 1995c) pour combler ce qu'il considère comme étant une sur simplification. On introduit ainsi la fonction fal (Fig. 4.3) définie par J. Han (Han, 1995c) :

$$\text{fal}(x, \alpha, \delta) = \begin{cases} |x|^\alpha \text{sign}(x), & |x| > \delta \\ \frac{x}{\delta^{1-\alpha}}, & |x| \leq \delta \end{cases} \quad (4.43)$$

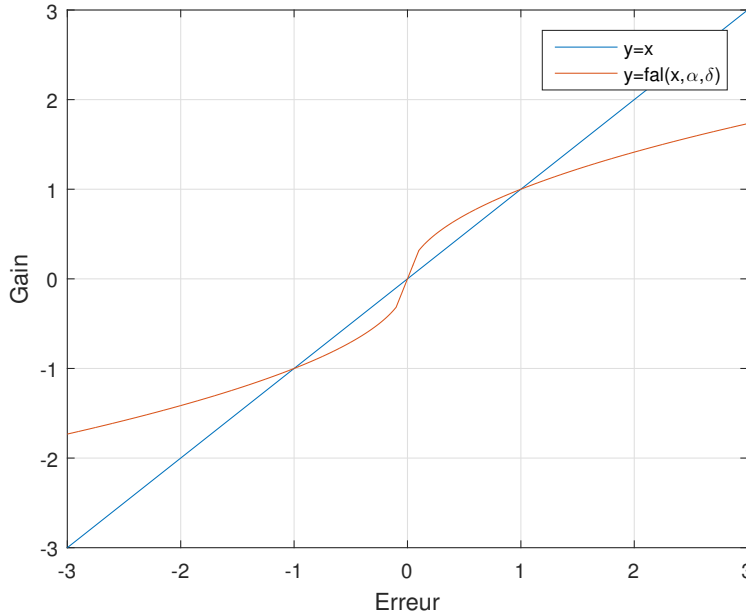


Figure 4.3 Comparaison des gains linéaires et non-linéaires

L'estimateur NLESO (Han, 1995c) (4.25) prend alors la forme suivante :

$$\left\{ \begin{array}{l} \mathbf{e} = \mathbf{x}_1 - \hat{\mathbf{x}}_1 \\ \mathbf{f}\mathbf{e}_1 = \text{fal}(\mathbf{e}, \alpha_1, \delta) \\ \mathbf{f}\mathbf{e}_2 = \text{fal}(\mathbf{e}, \alpha_2, \delta) \\ \dot{\hat{\mathbf{x}}}_1 = \hat{\mathbf{x}}_2 + \beta_1 \mathbf{e} \\ \dot{\hat{\mathbf{x}}}_2 = \hat{\mathbf{x}}_3 + \beta_2 \mathbf{f}\mathbf{e}_1 + \mathbf{B}\mathbf{u} \\ \dot{\hat{\mathbf{x}}}_3 = \beta_3 \mathbf{f}\mathbf{e}_2 \end{array} \right. \quad (4.44)$$

Cette fonction nous permet d'avoir de plus grands gains pour des petites erreurs (donc une dynamique plus rapide), et des plus petits gains pour des grandes erreurs (donc plus stable).

4.3.3 Optimisation des estimateurs

Dans un premier temps, nous allons optimiser l'estimateur linéaire en faisant varier ω_0 pour l'estimateur du contrôle position. L'estimateur pour le contrôle d'attitude suit très bien les moments perturbateurs pour $\omega_0 = 30$ (Fig. 4.4). Comme on peut le voir sur la figure 4.4,

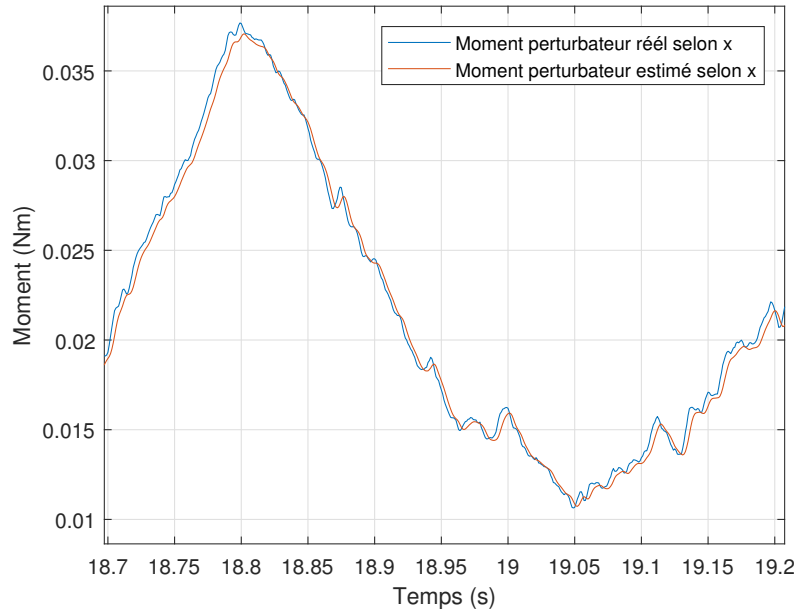


Figure 4.4 Comparaison du moment perturbateur estimé et réel selon x

l'estimé suit bien la valeur réelle. Comme on pouvait le prévoir, on distingue un léger retard de l'estimateur de l'ordre de ω_0^{-1} . Nous utilisons comme critère à optimiser l'écart-type sur la position selon x , étant donné que c'est celui-ci qui est présenté dans (Tran et al., 2015).

Lorsque l'on passe à $\omega_0 > 7$, notre contrôleur devient instable. Afin de mieux comprendre ce

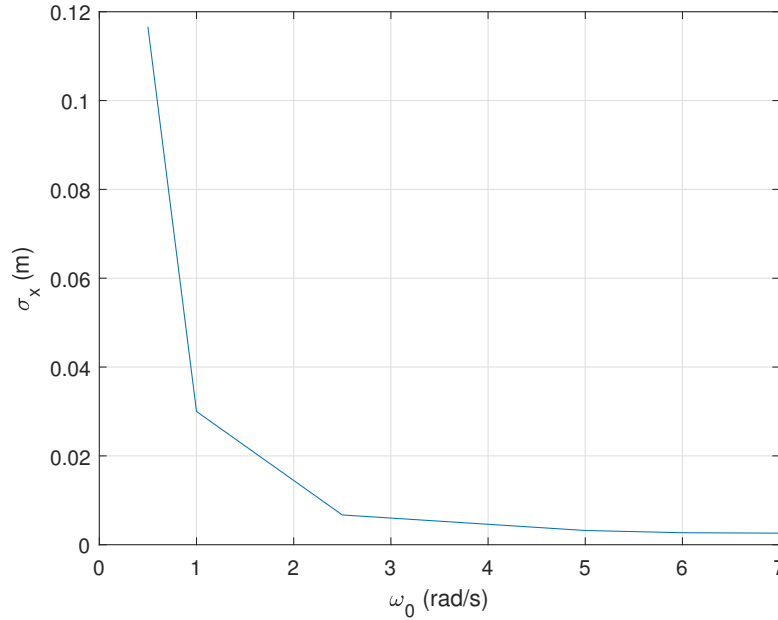


Figure 4.5 Écart-type de la position x en fonction de ω_0

phénomène, nous allons aussi présenter les perturbations et leurs estimées (Figs. 4.6, 4.7 et 4.8). Il est à noter ici que l'on ne représente pas des forces à proprement parler, mais des forces normalisées (divisées par mg_0), qui sont donc en radians et non en Newton, car l'on va s'en servir pour corriger les angles d'Euler désirés (qui eux sont en radians). On voit toujours que l'on a des retards de l'estimation sur les valeurs réelles de l'ordre de ω_0^{-1} . Cependant l'instabilité est particulièrement remarquable lorsque la force est proche de 0, comme on peut s'en rendre compte sur la figure 4.9. On va donc essayer d'améliorer la rapidité de l'estimateur tout en essayant de conserver la stabilité avec la méthode d'estimation non-linéaire explicitée plus haut. Elle est cependant plus complexe à optimiser du fait de nombreux paramètres à prendre en compte : β_1 , β_2 , β_3 , α_1 , α_2 et δ . Nous pouvons trouver un article qui propose des valeurs pour un contrôleur discret (Dong et al., 2016), mais les auteurs précisent que les coefficients β_i ont été trouvés empiriquement. Pour des raisons relatives à l'ordre du système, il faut que $\alpha_1 = 0.5$ et $\alpha_2 = 0.25$. Nous allons exprimer les coefficients β_i en fonction de ω_0 . Il ne reste ainsi que deux paramètres à régler, δ et ω_0 .

Il est naturel de penser que lorsque l'on augmente ω_0 , on va augmenter la rapidité de convergence de l'estimateur (et ainsi diminuer le retard par rapport aux valeurs réelles) et l'instabilité. En revanche en augmentant δ , on augmente la taille de la «couche limite» (zone avec des grands gains linéaires) tout en diminuant le gain associé à cette zone.

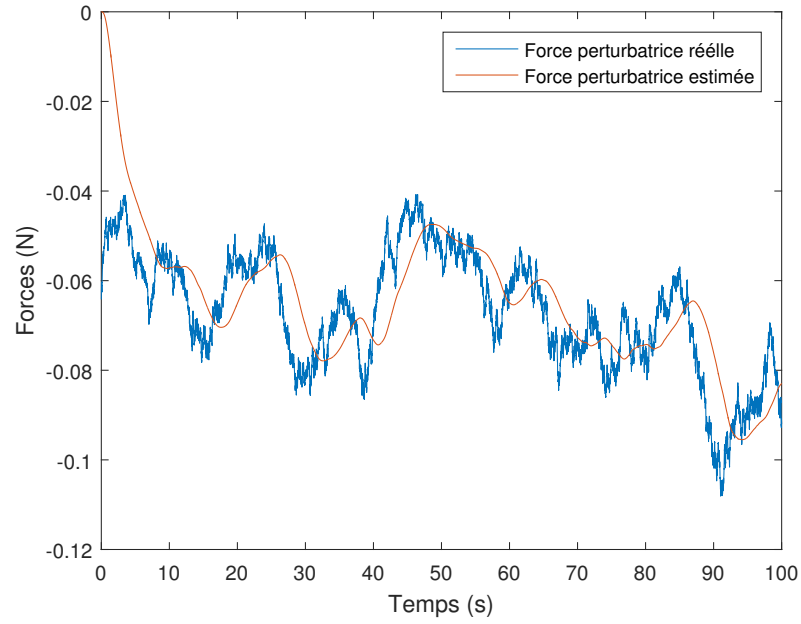


Figure 4.6 Comparaison de la force perturbatrice estimée et réelle selon x pour $\omega_0 = 1$

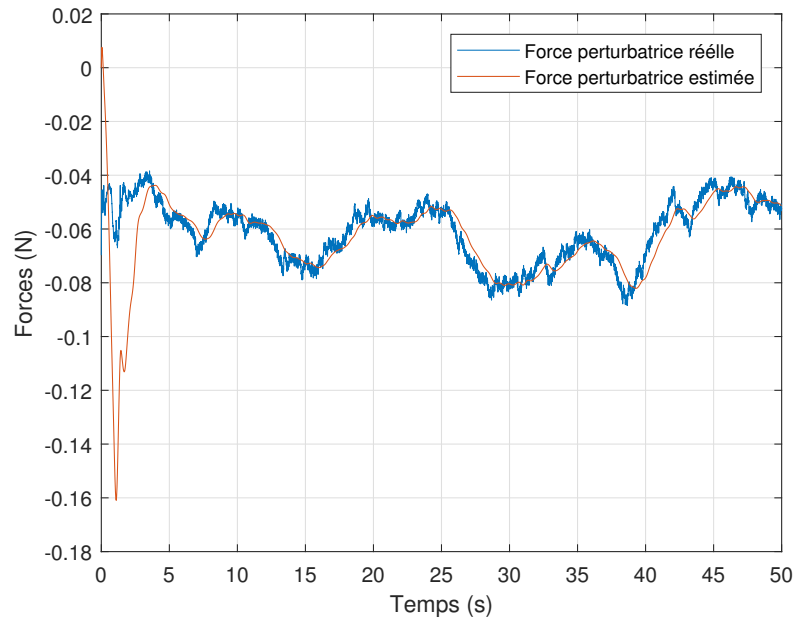


Figure 4.7 Comparaison de la force perturbatrice estimée et réelle selon x pour $\omega_0 = 7$

En faisant quelques essais en augmentant progressivement ω_0 et en adaptant δ en fonction, on peut faire deux remarques importantes :

- une augmentation de ω_0 requiert une augmentation de δ afin d'assurer la stabilité

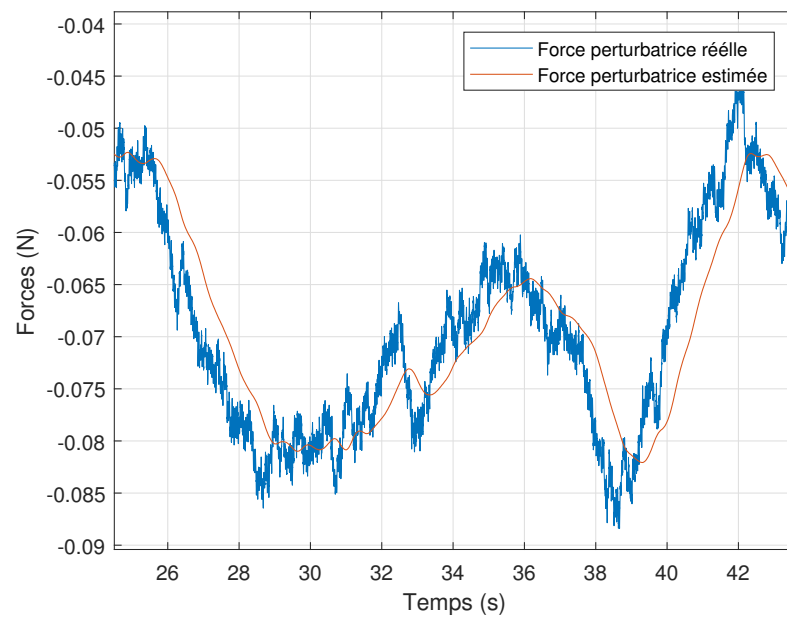


Figure 4.8 Zoom sur la comparaison de la force perturbatrice estimée et réelle selon x pour $\omega_0 = 7$

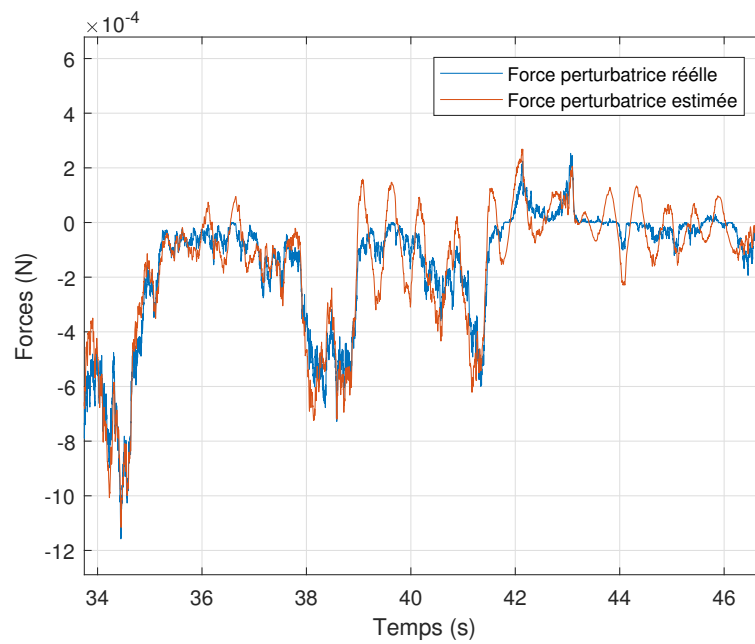


Figure 4.9 Zoom sur la comparaison de la force perturbatrice estimée et réelle selon y pour $\omega_0 = 7$

— l'erreur ne dépasse pas δ dans les simulations

Cela veut donc dire que le caractère non-linéaire de l'estimateur n'est pas exploité, mais nous arrivons pourtant à de meilleurs résultats. C'est donc la manière de choisir les coefficients β_i qui n'étaient pas idéales. On peut donc pour l'instant laisser de côté l'idée d'un estimateur non-linéaire, et chercher à optimiser les gains avec sa nouvelle forme :

$$\begin{cases} \mathbf{e} &= \mathbf{x}_1 - \hat{\mathbf{x}}_1 \\ \dot{\hat{\mathbf{x}}}_1 &= \hat{\mathbf{x}}_2 + \beta_1 \mathbf{e} \\ \dot{\hat{\mathbf{x}}}_2 &= \hat{\mathbf{x}}_3 + \frac{\beta_2}{\delta^{1-\alpha_1}} \mathbf{e} + \mathbf{B}\mathbf{u} \\ \dot{\hat{\mathbf{x}}}_3 &= \frac{\beta_3}{\delta^{1-\alpha_2}} \mathbf{e} \end{cases} \quad (4.45)$$

ce qui revient finalement à optimiser l'estimateur linéaire (4.25) en fonction des trois coefficients β_i , sans supposer qu'ils sont dépendants d'une variable commune ω_0 . Une telle optimisation étant longue et coûteuse en temps de calcul, nous ne la ferons pas dans ce document. Cependant, nous pouvons nous apercevoir en manipulant les coefficients β_i qu'il est possible d'améliorer grandement les performances de l'estimateur. De manière intuitive, on comprend qu'en augmentant β_1 et en diminuant β_3 , on augmente à la fois la rapidité de convergence de l'état en x_1 et la stabilité en x_3 . Nous allons donc tenter de donner forme à cette intuition mathématiquement parlant dans la prochaine section.

4.3.4 Décorrélation des pôles des états associés

Afin de mieux comprendre comment marche notre estimateur, il est intéressant d'obtenir les fonctions de transferts entre les estimations et leurs états associés, i.e., $\hat{x}_i(s)/x_i(s)$. Pour cela, nous allons appliquer la transformée de Laplace à notre système et à l'estimateur (ici nous ne considérons qu'une seule dimension à la fois comme les axes x , y et z sont indépendants) :

$$\begin{cases} \dot{x}_1 = x_2 \\ \dot{x}_2 = x_3 + bu \\ \dot{x}_3 = w \end{cases} \quad (4.46)$$

$$\begin{cases} \dot{\hat{x}}_1 = \hat{x}_2 + \beta_1(x_1 - \hat{x}_1) \\ \dot{\hat{x}}_2 = \hat{x}_3 + bu + \beta_2(x_1 - \hat{x}_1) \\ \dot{\hat{x}}_3 = \beta_3(x_1 - \hat{x}_1) \end{cases} \quad (4.47)$$

En supposant les conditions initiales nulles, le passage dans le domaine de Laplace nous donne :

$$\begin{cases} sx_1(s) = x_2(s) \\ sx_2(s) = x_3(s) + bu(s) \\ sx_3(s) = w(s) \end{cases} \quad (4.48)$$

$$\begin{cases} s\hat{x}_1(s) = \hat{x}_2(s) + \beta_1(x_1(s) - \hat{x}_1(s)) \\ s\hat{x}_2(s) = \hat{x}_3(s) + bu(s) + \beta_2(x_1(s) - \hat{x}_1(s)) \\ s\hat{x}_3(s) = \beta_3(x_1(s) - \hat{x}_1(s)) \end{cases} \quad (4.49)$$

En manipulant les équations (4.48) et (4.49), on obtient les fonctions de transfert \hat{x}_1/x_1 , \hat{x}_2/x_2 et \hat{x}_3/x_3 . Nous trouvons les relations suivantes :

$$\begin{cases} G_1(s) = \frac{\hat{x}_1(s)}{x_1(s)} = \frac{\beta_1 s^2 + \beta_2 s + \beta_3}{s^3 + \beta_1 s^2 + \beta_2 s + \beta_3} \\ G_2(s) = \frac{\hat{x}_2(s)}{x_2(s)} = \frac{\beta_2 s + \beta_3}{s^3 + \beta_1 s^2 + \beta_2 s + \beta_3} \\ G_3(s) = \frac{\hat{x}_3(s)}{x_3(s)} = \frac{\beta_3}{s^3 + \beta_1 s^2 + \beta_2 s + \beta_3} \end{cases} \quad (4.50)$$

Nous pouvons réécrire (4.50) afin d'y faire apparaître les trois pôles ω_1 , ω_2 et ω_3 définis tel que $s^3 + \beta_1 s^2 + \beta_2 s + \beta_3 = (s + \omega_1)(s + \omega_2)(s + \omega_3)$, ce qui nous donne :

$$\begin{cases} \beta_1 = \omega_1 + \omega_2 + \omega_3 \\ \beta_2 = \omega_1\omega_2 + \omega_3\omega_1 + \omega_2\omega_3 \\ \beta_3 = \omega_1\omega_2\omega_3 \end{cases} \quad (4.51)$$

La fonction de transfert $\hat{x}_3(s)/x_3(s)$ a trois pôles, $\hat{x}_2(s)/x_2(s)$ a 3 pôles et 1 zéro et $\hat{x}_1(s)/x_1(s)$ a trois pôles et deux zéros. En supposant que nous avons trois pôles distincts, la fonction de transfert a tendance à se comporter selon son pôle le plus lent. Ainsi donc, si nous choisissons judicieusement nos pôles, nous pouvons faire correspondre les zéros des fonctions de transfert G_1 et G_2 aux pôles les plus lents afin que les estimateurs associés soient plus rapides. Pour cela, considérons le cas $\omega_1 \ll \omega_2 \ll \omega_3$. Nous avons alors les approximations suivantes :

$$\begin{cases} \beta_1 \approx \omega_3 \\ \beta_2 \approx \omega_2\omega_3 \\ \beta_3 = \omega_1\omega_2\omega_3 \end{cases} \quad (4.52)$$

En exprimant les zéros des fonctions de transfert nous avons :

$$\frac{\beta_2 + \sqrt{\beta_2^2 - 4\beta_3\beta_1}}{2\beta_1} \approx \frac{\omega_2}{2} + \frac{\sqrt{\omega_2^2\omega_3^2}}{2\omega_3} = \omega_2 \quad (4.53)$$

car $\sqrt{\beta_2^2 - 4\beta_3\beta_1} = \sqrt{\omega_2^2\omega_3^2 - 4\omega_1\omega_2\omega_3^2} \approx \omega_2\omega_3$. En effet, une approximation du premier ordre suffit ici. En revanche pour le deuxième zéro de G_1 il nous faut faire une approximation du second ordre, c'est-à-dire $\sqrt{\beta_2^2 - 4\beta_3\beta_1} = \sqrt{\omega_2^2\omega_3^2 - 4\omega_1\omega_2\omega_3^2} \approx \omega_2\omega_3(1 - 2\omega_1/\omega_2) = \omega_2\omega_3 - 2\omega_1\omega_3$, ce qui nous donne le zéro suivant :

$$\frac{\beta_2 - \sqrt{\beta_2^2 - 4\beta_3\beta_1}}{2\beta_1} \approx \frac{\omega_2}{2} - \frac{\sqrt{\omega_2^2\omega_3^2 - 4\omega_1\omega_2\omega_3^2}}{2\omega_3} \approx \frac{\omega_2}{2} - \frac{\omega_2\omega_3 - 2\omega_1\omega_3}{2\omega_3} = \omega_1 \quad (4.54)$$

Le zéro de G_2 s'exprime plus facilement :

$$\frac{\beta_3}{\beta_2} \approx \omega_1 \quad (4.55)$$

Les fonctions de transfert de G_1 , G_2 et G_3 peuvent alors s'écrire de la manière suivante :

$$\begin{cases} G_1(s) = \frac{\beta_1(s + \frac{\beta_2 + \sqrt{\beta_2^2 - 4\beta_3\beta_1}}{2\beta_1})(s + \frac{\beta_2 - \sqrt{\beta_2^2 - 4\beta_3\beta_1}}{2\beta_1})}{(s + \omega_1)(s + \omega_2)(s + \omega_3)} \approx \frac{\beta_1(s + \omega_1)(s + \omega_2)}{(s + \omega_1)(s + \omega_2)(s + \omega_3)} \approx \frac{\omega_3}{s + \omega_3} \\ G_2(s) = \frac{\beta_2(s + \frac{\beta_3}{\beta_2})}{(s + \omega_1)(s + \omega_2)(s + \omega_3)} \approx \frac{\beta_2(s + \omega_1)}{(s + \omega_1)(s + \omega_2)(s + \omega_3)} \approx \frac{\omega_2\omega_3}{(s + \omega_3)(s + \omega_2)} \\ G_3(s) \approx \frac{\omega_1\omega_2\omega_3}{(s + \omega_1)(s + \omega_2)(s + \omega_3)} \end{cases} \quad (4.56)$$

Dans ces conditions nous nous retrouvons avec les comportements désirés :

- la fonction de transfert $\hat{x}_1(s)/x_1(s)$ a une fréquence de coupure à ω_3 , et le suivi est très rapide en position ;
- la fonction de transfert $\hat{x}_2(s)/x_2(s)$ a deux pôles (en ω_1 et ω_2) et a donc une rapidité en ω_2 (le moins rapide des deux), ce qui nous garantit une bonne vitesse de suivi et une estimation stable.
- la fonction de transfert $\hat{x}_3(s)/x_3(s)$ a les trois pôles ω_1 , ω_2 et ω_3 , ce qui nous amène une rapidité de suivi de l'ordre de ω_1 , plus lente mais qui nous permet d'avoir une estimation des forces beaucoup plus stable.

Il y a un grand intérêt à gagner en précision sur l'estimation en vitesse, car dans notre loi de commande, nous avons besoin de la dérivée de la position (que ce soit en contrôle

d'attitude ou de position). Or en supposant que nous n'avons pas accès à ses valeurs par simple dérivation (à cause du bruit par exemple), il nous faut la calculer. L'estimateur nous permet donc de fournir cette dérivée avec une rapidité de ω_2 , bien plus rapide que ω_1 .

Cette décorrélation des pôles pour chaque état nous permet donc de meilleures performances d'estimation. En effet, avec la technique d'un pôle triple utilisé précédemment, l'estimation de x_1 était fortement limitée par la nécessité d'avoir un pôle triple lent afin d'être stable. Ceci nous permet tout de même d'avoir une meilleure rapidité d'estimation de x_3 comme on peut le constater sur la simulation de la figure 4.10.

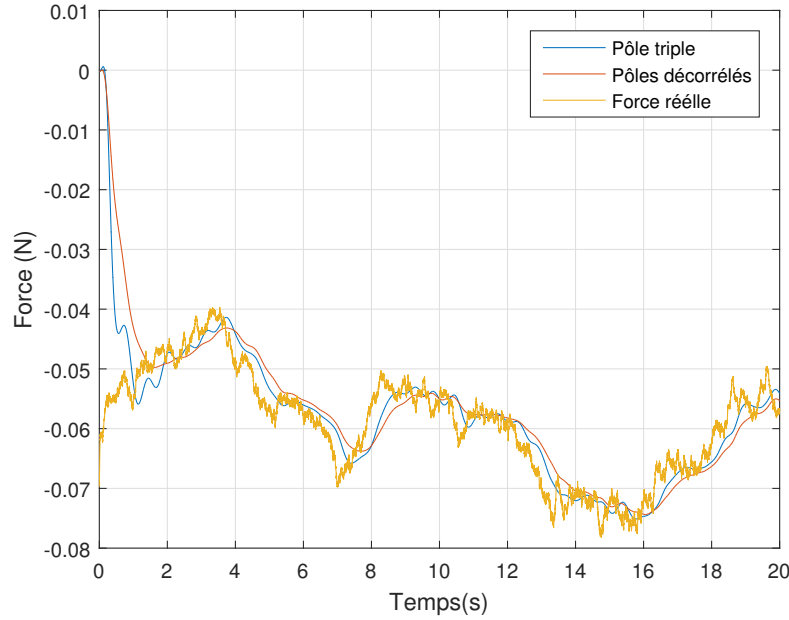


Figure 4.10 Comparaison d'estimations de x_3 avec un pôle triple $\omega_0 = 2\text{rad/s}$ et de trois pôles décorrélés $\omega_1 = 2\text{rad/s}$, $\omega_2 = 6\text{rad/s}$ et $\omega_3 = 20\text{rad/s}$

Il paraît cependant intéressant de se pencher sur l'optimisation de cet estimateur en essayant d'en comprendre mieux les mécanismes que ce soit dans le cas linéaire ou non linéaire, car plus cet estimateur est précis, meilleur est le contrôle associé. Nous allons donc présenter quelques candidats possibles susceptibles d'améliorer cet estimateur. Plusieurs méthodes d'estimation existent autres que celles présentées dans ce document. Parmi celles existantes nous en avons retenues quelques-unes qui pourraient potentiellement être de bonnes candidates :

- une méthode proposée par H.Khalil en 2013 se base sur une saturation de la commande via le calcul de l'estimation du bassin d'attraction du système (Khalil, 2008) ;
- une autre méthode nommée estimateur glissant (*Sliding Mode Observer*) (Edwards and Spurgeon, 1998) se base sur des gains du type bang-bang.

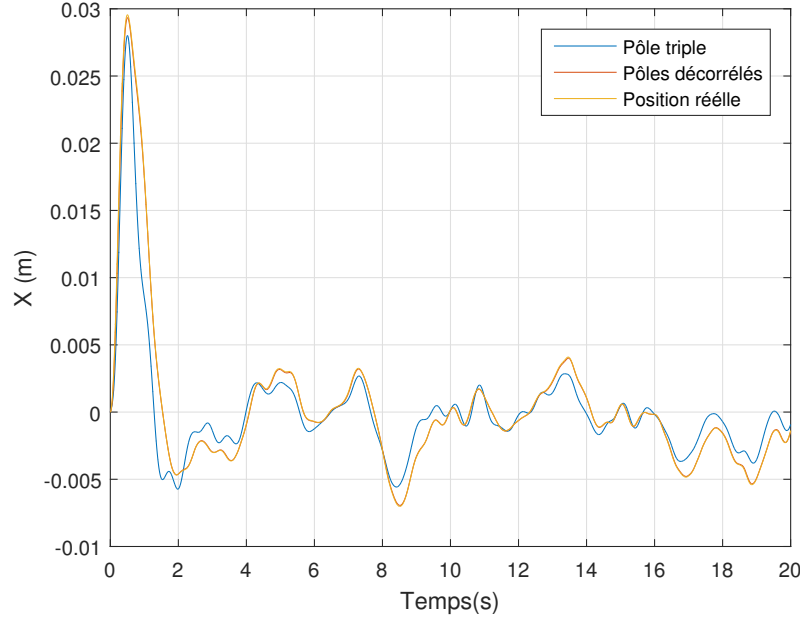


Figure 4.11 Comparaison d'estimations de x_1 avec un pôle triple $\omega_0 = 2\text{rad/s}$ et de trois pôles décorrés $\omega_1 = 2\text{rad/s}$, $\omega_2 = 6\text{rad/s}$ et $\omega_3 = 20\text{rad/s}$

- La dernière retenue et la plus prometteuse est un multi-estimateur (Bernat and Stepien, 2015), qui est une combinaison linéaire de plusieurs estimateurs linéaires à grands gains. Cet estimateur se base sur l'estimateur linéaire déjà présenté dans ce document (*High-Gain Observer* - HGO) que nous utilisons dans notre ADRC. Nous allons donc le présenter dans la sous-partie suivante comme moyen susceptible d'améliorer l'ESO.

4.3.5 Le multi-estimateur

Le multi-estimateur (Bernat and Stepien, 2015) a été proposé par Nagy, Marx, Mourot, Schutz, et Ragot (Nagy et al., 2010) comme technique permettant la diminution du phénomène de pic (*peaking phenomenon*) de l'estimateur HGO. Ce phénomène représente la partie transitoire de notre estimateur. Ainsi, si l'on impose une rapidité de convergence de l'ordre de ϵ à notre estimateur, nous allons avoir un pic d'erreur de l'ordre de ϵ^{-1} (Fig. 4.12).

Plus on cherche à converger rapidement, plus le phénomène de pic prendra de l'ampleur et risque de déstabiliser notre système en causant des commandes de trop grandes amplitudes. Afin d'éviter ce phénomène, plusieurs méthodes ont été explorées telle que la saturation de commande par Khalil (Esfandiari and Khalil, 1992), mais aussi l'estimation multiple. Le multi-estimateur est un estimateur comprenant deux sous-systèmes : le premier est un en-

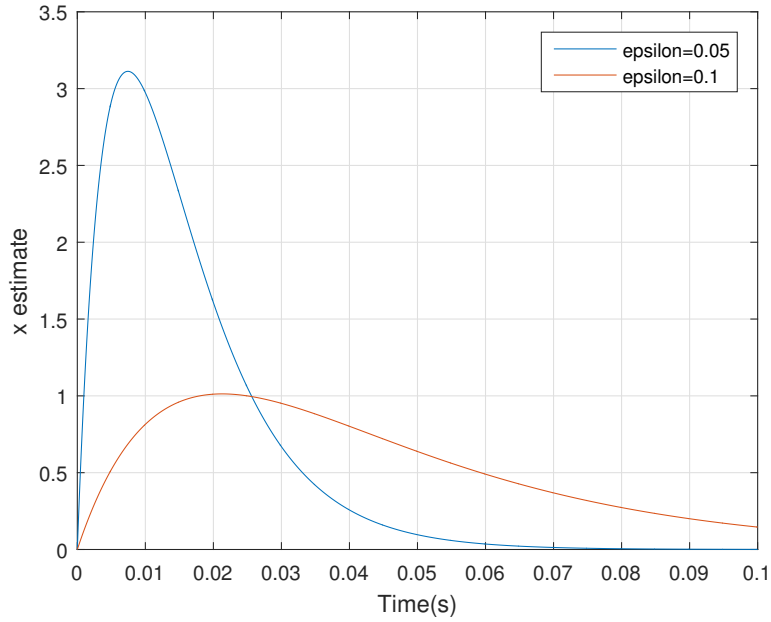


Figure 4.12 Pic d'erreur d'un estimateur HGO en fonction de ϵ

semble d'estimateurs HGO, et le second est un algorithme de régression récursive permettant de faire une combinaison linéaire des états estimés par les différents HGO. On peut représenter ce fonctionnement à l'aide du schéma de la figure 4.13, reprise de l'article expliquant le multi-estimateur (Bernat and Stepien, 2015).

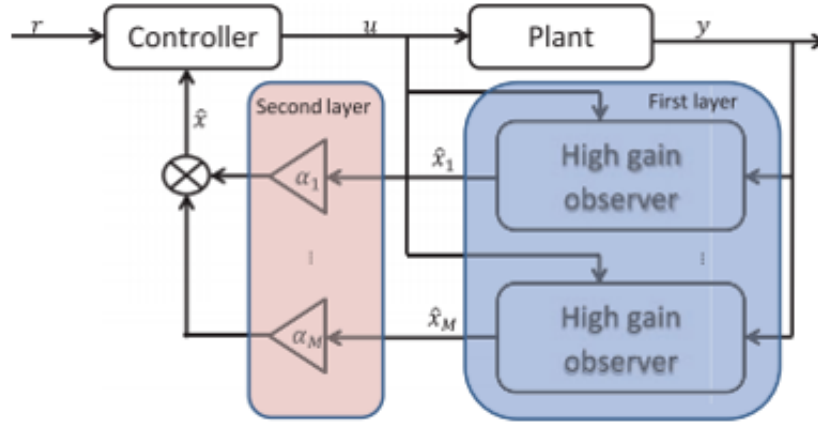


Figure 4.13 Fonctionnement du multi-estimateur (Bernat and Stepien, 2015)

Nous avons ainsi k estimateurs HGO dans le premier sous-système qui nous donnent les k différents états estimés \hat{x}_i , puis nous en faisons une combinaison linéaire afin d'obtenir l'état

estimé final $\hat{x}(t) = \sum_{i=1}^k \alpha_i(t) \hat{x}_i$. Les k différents estimateurs sont les mêmes estimateurs HGO avec les mêmes gains, à ceci près qu'ils ont des conditions initiales différentes. L'algorithme de récursion est ensuite utilisé afin de déterminer les coefficients α_i et d'en déduire l'état estimé final.

Ces coefficients doivent nous permettre de réduire l'erreur entre l'état estimé et l'état réel de manière significative dans les premiers instants. En effet, au bout d'un temps ϵ , les estimateurs auront tous convergé et le multi-estimateur n'aura donc plus d'intérêt à long terme. Cet ajout dans le contrôleur nous permet seulement de diminuer le phénomène de pic qui peut déstabiliser le système.

Nous allons donc exprimer les relations qui nous permettent de calculer les coefficients $\alpha_i(t)$. Pour cela, nous allons reprendre la démonstration effectuée par Bernat (Bernat and Stepien, 2015) , et nous allons donc chercher à produire l'égalité suivante :

$$\hat{x}(t) = \sum_{i=1}^4 \alpha_i(t) \hat{\mathbf{x}}_i \quad (4.57)$$

que l'on peut réécrire de la manière suivante :

$$\sum_{i=1}^4 \alpha_i(t) \mathbf{e}_i = 0 \quad (4.58)$$

Cependant les vecteurs d'erreurs $\mathbf{e}_i = \hat{\mathbf{x}}_i - \mathbf{x}_i$ ne sont pas entièrement connus, seule la première composante l'est (celle en position). Il nous faut donc travailler sur d'autres vecteurs connus dépendant seulement des erreurs en position. Supposons que nous disposons d'une telle variété ζ_i pour i allant de 1 à 4 et qui vérifie donc :

$$\sum_{i=1}^4 \alpha_i(t) \zeta_i = 0 \quad (4.59)$$

Nous rajoutons aussi la condition intuitive suivante :

$$\sum_{i=1}^4 \alpha_i(t) = 1 \quad (4.60)$$

On peut donc réécrire l'équation (4.59) sous la forme d'une régression linéaire :

$$-\zeta_4 = \begin{bmatrix} \zeta_1 - \zeta_4 & \zeta_2 - \zeta_4 & \zeta_3 - \zeta_4 \end{bmatrix} \begin{bmatrix} \alpha_1(t) \\ \alpha_2(t) \\ \alpha_3(t) \end{bmatrix} \quad (4.61)$$

Il nous faut donc trouver l'expression des ζ_i en fonction de l'erreur en position. Pour cela nous allons travailler sur notre système linéarisé comme démontré dans l'article (Bernat and Stepien, 2015) :

$$\begin{cases} \dot{\mathbf{x}} = \mathbf{A}\mathbf{x} + \mathbf{B}\phi(\mathbf{x}, u) \\ y = \mathbf{C}\mathbf{x} \end{cases} \quad (4.62)$$

avec

$$\begin{cases} \mathbf{A} = \begin{bmatrix} 0 & 1 & 0 \\ 0 & 0 & 1 \\ 0 & 0 & 0 \end{bmatrix} \\ \mathbf{B} = \begin{bmatrix} 0 \\ 1 \\ 0 \end{bmatrix} \\ \mathbf{C} = \begin{bmatrix} 1 & 0 & 0 \end{bmatrix} \end{cases} \quad (4.63)$$

La fonction ϕ représente la non-linéarité potentielle et l'influence de la commande sur le système. Nous désignons l'erreur en position de la manière suivante :

$$\eta_i(t) = \mathbf{C}\mathbf{e}_i(t) \quad (4.64)$$

En dérivant cette expression deux fois nous obtenons le système d'équations suivant :

$$\begin{cases} \eta_i(t) = \mathbf{C}\mathbf{e}_i(t) \\ \dot{\eta}_i(t) = \mathbf{C}\mathbf{A}\mathbf{e}_i(t) - \mathbf{C}\mathbf{L}\eta_i(t) \\ \ddot{\eta}_i(t) = \mathbf{C}\mathbf{A}^2\mathbf{e}_i(t) - \mathbf{C}\mathbf{A}\mathbf{L}\eta_i(t) - \mathbf{C}\mathbf{L}\dot{\eta}_i(t) \end{cases} \quad (4.65)$$

En effet comme $\mathbf{C}\mathbf{B} = 0$, cela nous permet d'éliminer les termes en ϕ . En intégrant le système précédent deux fois entre t_d et t avec t_d une constante, nous avons l'égalité suivante :

$$\begin{bmatrix} \mathbf{C} \\ \mathbf{C}\mathbf{A} \\ \mathbf{C}\mathbf{A}^2 \end{bmatrix} \int_{t_d}^t \int_{t_d}^t \mathbf{e}_i(\tau) d\tau = \begin{bmatrix} 1 & 0 & 0 \\ \mathbf{C}\mathbf{L} & 1 & 0 \\ \mathbf{C}\mathbf{A}\mathbf{L} & \mathbf{C}\mathbf{L} & 1 \end{bmatrix} \begin{bmatrix} \int_{t_d}^t \int_{t_d}^t \eta_i(\tau) d\tau \\ \int_{t_d}^t \eta_i(\tau) d\tau \\ \eta_i(t) - \eta_i(t - t_d) \end{bmatrix} \quad (4.66)$$

avec $\mathbf{L} = [\beta_1 \ \beta_2 \ \beta_3]^\top$. On pose :

$$\zeta_i = \begin{bmatrix} 1 & 0 & 0 \\ \mathbf{CL} & 1 & 0 \\ \mathbf{CAL} & \mathbf{CL} & 1 \end{bmatrix} \begin{bmatrix} \int_{t_d}^t \int_{t_d}^t \eta_i(\tau) d\tau \\ \int_{t_d}^t \eta_i(\tau) d\tau \\ \eta_i(t) - \eta_i(t - t_d) \end{bmatrix} \quad (4.67)$$

Nous obtenons alors la variété recherchée dépendant des erreurs \mathbf{e}_i tout en pouvant la calculer en utilisant seulement les erreurs en position η_i et ses intégrales. Ceci nous permet (en utilisant un algorithme de régression linéaire, le RLS (*Recursive Least Square* (Cioffi and Kailath, 1984) que l'on ne présentera pas dans ce document) de calculer les coefficients α_i , à condition que les vecteurs soient linéairement indépendants, c'est-à-dire que l'on choisit pour chaque estimateur des conditions initiales différentes. On peut maintenant calculer la taille du pic d'un multi-estimateur. La démonstration correspondante (Bernat and Stepien, 2015) sera reprise un peu plus loin, afin que l'on puisse y rajouter de nouveaux éléments déterminants pour notre estimateur. Nous en présentons simplement l'approximation finale de l'article sur les multi-estimateurs (Bernat and Stepien, 2015) :

$$\frac{\sup(\mathbf{e}_{simple})}{\sup(\mathbf{e}_{multi})} \approx \frac{1 + \frac{1}{\epsilon}}{\ln^{n-1}(\frac{1}{\epsilon})} \quad (4.68)$$

avec n la dimension du système, ici égale à 3 et ϵ la rapidité d'estimation. Cette variable détermine le temps que l'estimation met à converger vers l'état réel du système. Pour des variations aussi rapides que le vent, on aimerait que notre estimateur soit capable de suivre ces variations avec un retard de seulement $\epsilon = 0.1s$, ou moins si c'est possible. Il s'agit de l'ordre de grandeur typique du temps d'une variation significative de la vitesse du vent. La représentation graphique (Fig. 4.14) permet de mieux nous rendre compte de l'efficacité du multi-estimateur.

On remarque tout d'abord que plus la dimension du système est grande, moins le multi-estimateur est efficace. Il apparaît aussi que prendre un trop grand ϵ ne sera pas plus intéressant par rapport à l'intensification du phénomène de pic qui reste toujours proportionnel à ϵ^{-1} . Afin d'explorer l'amélioration offerte par notre nouvel estimateur, nous allons le tester sur un système plus simple que notre quadricoptère (exemple présenté dans (Bernat and Stepien, 2015)). Il s'agit du système suivant :

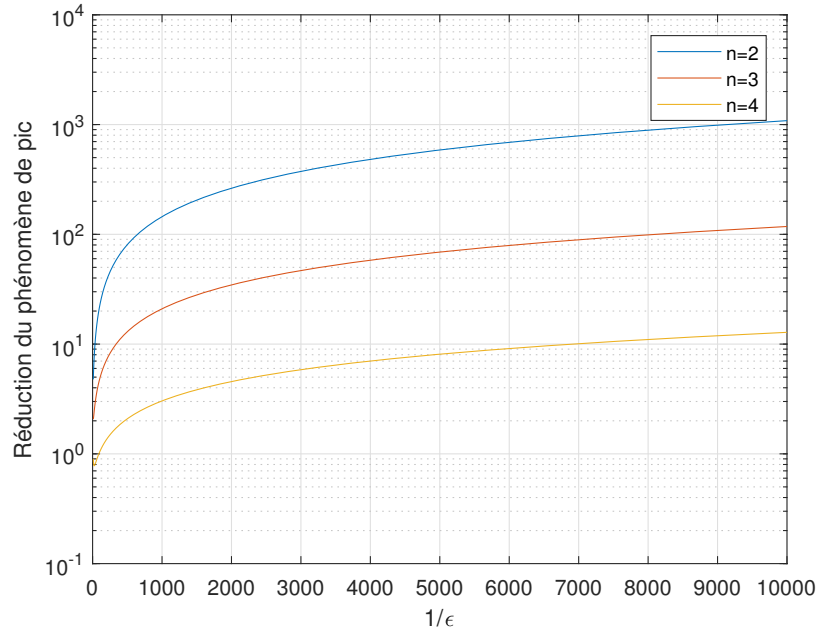


Figure 4.14 Réduction du phénomène de pic

$$\begin{cases} \dot{x}_1 = x_2 \\ \dot{x}_2 = x_2^3 + u(\hat{x}_1, \hat{x}_2) \\ y = x_1 \\ u = \text{sat}(-\hat{x}_2^3 - x_1 - \hat{x}_2) \end{cases} \quad (4.69)$$

Comme il s'agit ici d'un système de dimension 2, nous travaillons avec trois estimateurs explicites ci-dessous :

$$\begin{cases} \dot{\hat{x}}_1 = \hat{x}_2 + \frac{3}{\epsilon}(x_1 - \hat{x}_1) \\ \dot{\hat{x}}_2 = \hat{x}_3 + x_2^3 + u(\hat{x}_1, \hat{x}_2) + \frac{3}{\epsilon^2}(x_1 - \hat{x}_1) \\ \dot{\hat{x}}_3 = \frac{1}{\epsilon^3}(x_1 - \hat{x}_1) \end{cases} \quad (4.70)$$

Nous fixons aussi $\epsilon = 0.01$. Les figures 4.15, 4.16, 4.17 et 4.18 montrent les simulations effectuées sur Matlab avec des conditions initiales différentes pour chaque estimateur.

Les conditions initiales pour le système réel sont différentes des conditions initiales des trois autres estimateurs évidemment (sinon l'estimateur suivrait parfaitement l'état réel du système), mais doivent rester relativement proches afin qu'il y ait convergence des états estimés.

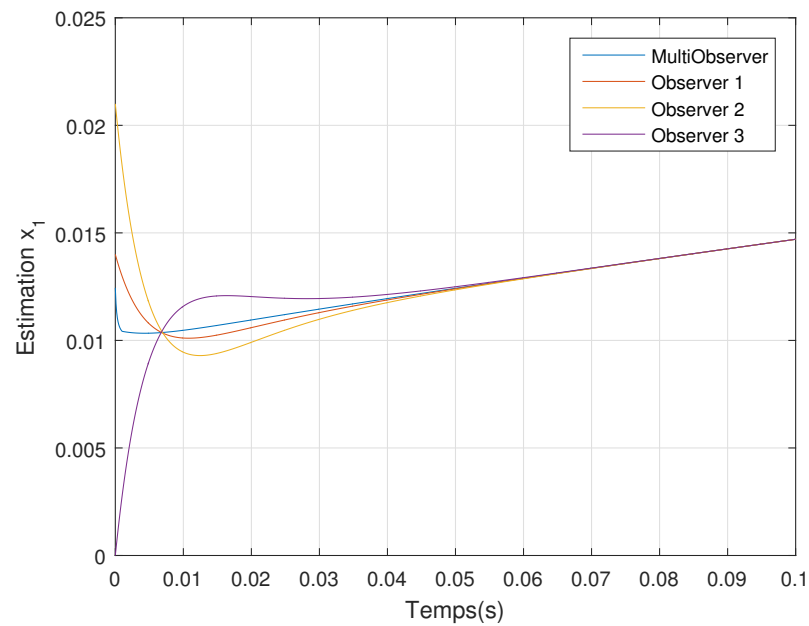


Figure 4.15 x_1 estimé en fonction du temps pour les différents estimateurs et le multi-estimateur

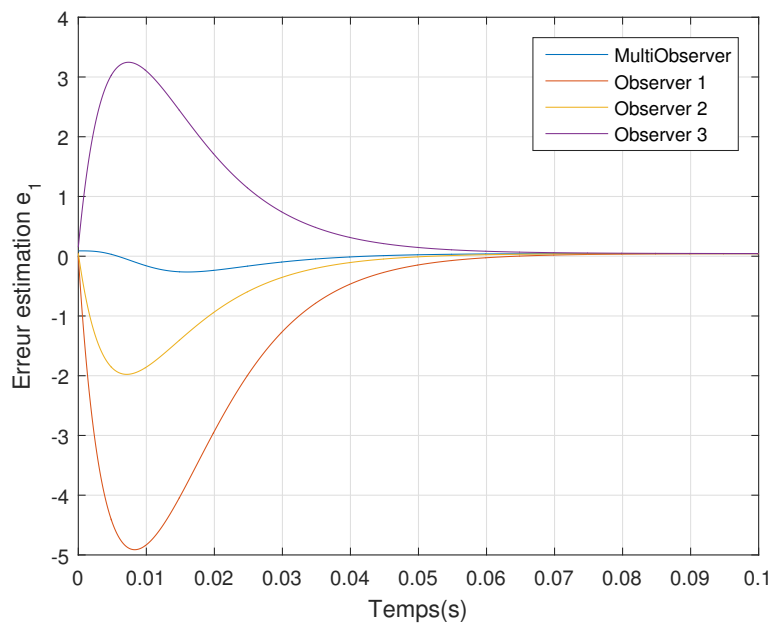


Figure 4.16 x_2 estimé en fonction du temps pour les différents estimateurs et le multi-estimateur

Il convient de faire quelques remarques à ce stade :

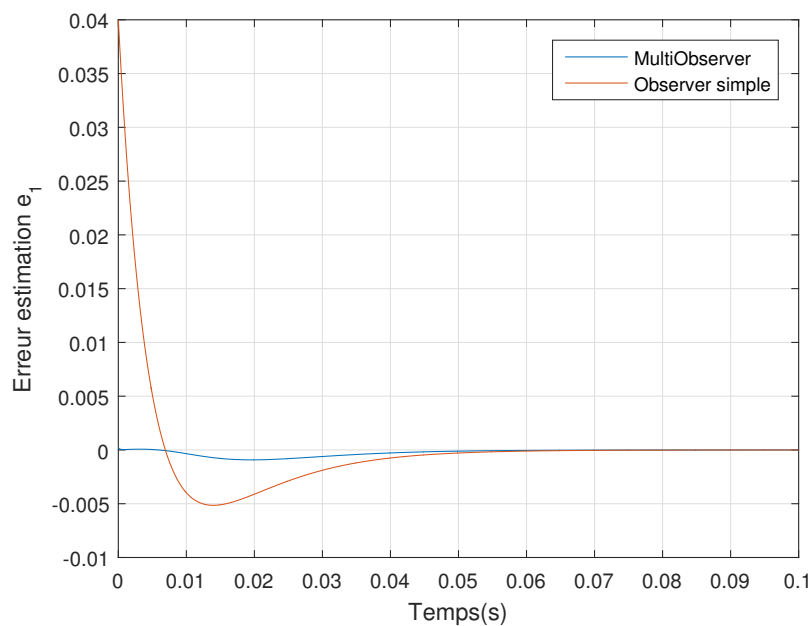


Figure 4.17 e_1 du meilleur estimateur simple et du multi-estimateur en fonction du temps

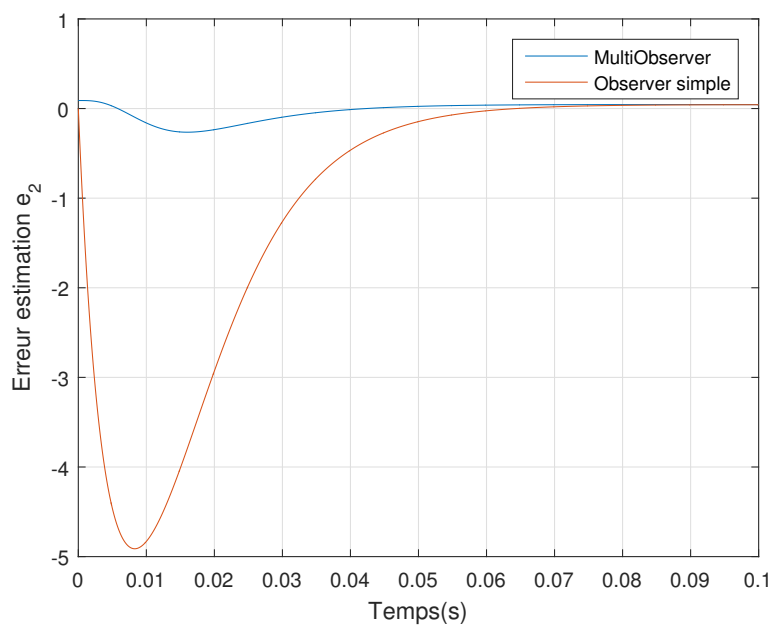


Figure 4.18 e_2 du meilleur estimateur simple et du multi-estimateur en fonction du temps

- nous observons bien une réduction du pic du multi-estimateur par rapport aux estimateurs simples de l'ordre de $\frac{1+\frac{1}{\epsilon}}{\ln(\frac{1}{\epsilon})}$;
- le pic d'un estimateur simple est aussi proportionnel à son erreur initial par rapport

- à l'état initial réel du système ;
- la réduction du pic est plus forte pour la première composante du vecteur d'état (x_1) du système que pour la seconde (x_2)

Ces remarques nous amènent à tester une seconde simulation avec cette fois-ci un estimateur dont les conditions initiales sont très proches des conditions initiales de l'état réel du système par rapport aux autres estimateurs. On peut exprimer cette condition de la façon suivante :

$$\mathbf{e}_1(0) = \hat{\mathbf{x}}_1(0) - \mathbf{x}_1(0) \ll \sum_{i=1}^3 \alpha_i(0) \hat{\mathbf{x}}_i(0) - \mathbf{x}(0) = \mathbf{e}_{multi}(0) \quad (4.71)$$

On obtient alors les figures 4.19 et 4.20.

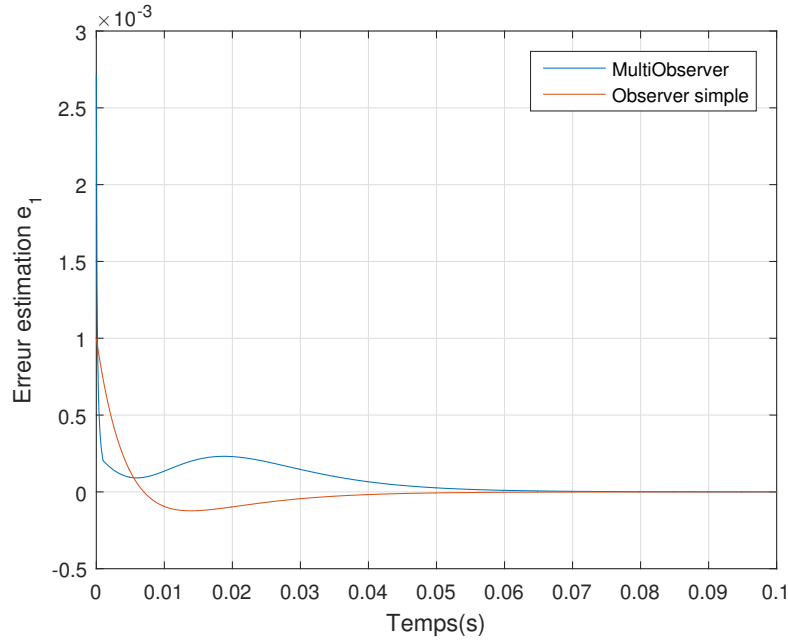


Figure 4.19 e_1 du meilleur estimateur simple et du multi-estimateur en fonction du temps

Il apparaît alors que le multi-estimateur est moins bon que l'estimateur simple qui est très proche de l'état réel à l'instant initial. Ceci paraît donc peu cohérent avec la démonstration de (Bernat and Stepien, 2015). Cependant en reprenant leur démonstration et en évitant la sursimplification finale de l'article, nous arrivons à la formule suivante :

$$\frac{\sup(\mathbf{e}_{simple})}{\sup(\mathbf{e}_{multi})} \approx \frac{\mathbf{e}_{simple}(0)}{\mathbf{e}_{multi}(0)} \frac{1 + \frac{1}{\epsilon}}{\ln^{n-1}(\frac{1}{\epsilon})} \quad (4.72)$$

Ainsi, le coefficient $\frac{\mathbf{e}_{simple}(0)}{\mathbf{e}_{multi}(0)}$ peut, selon les conditions initiales, compenser le coefficient d'amé-

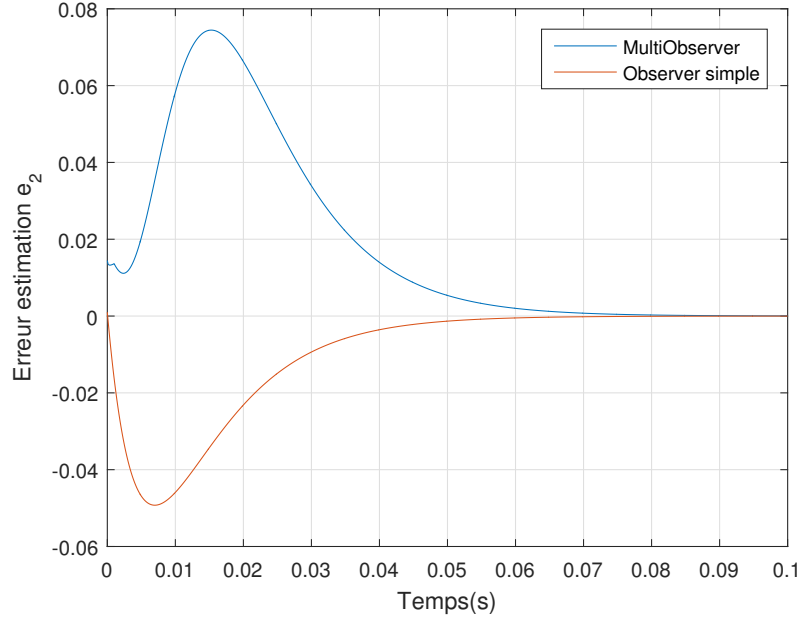


Figure 4.20 e_2 du meilleur estimateur simple et du multi-estimateur en fonction du temps

lioration de réduction de pic $\frac{1+\frac{1}{\epsilon}}{\ln^{n-1}(\frac{1}{\epsilon})}$. Dans notre exemple, ce dernier coefficient (avec $\epsilon = 0.01$ et $n = 2$) vaut $\frac{1+\frac{1}{\epsilon}}{\ln^{n-1}(\frac{1}{\epsilon})} \approx 22$. Afin d'observer cette compensation par les erreurs initiales, nous avons effectué deux simulations avec $\frac{\mathbf{e}_{simple}(0)}{\mathbf{e}_{multi}(0)} = \frac{1}{40}$ et $\frac{\mathbf{e}_{simple}(0)}{\mathbf{e}_{multi}(0)} = \frac{1}{10}$ (Figs 4.21 et 4.22).

Alors que le pic de l'estimateur simple est constant (l'erreur initiale $\mathbf{e}_{simple}(0)$ n'a pas été changé), le pic du multi-estimateur varie d'un facteur $\frac{40}{10} = 4$ passant d'une moins bonne performance par rapport à l'estimateur simple à une meilleure performance. Il paraît donc fondamental de soulever ce problème d'efficacité du multi-estimateur afin de choisir au mieux les conditions initiales des différents estimateurs simples. Il semble donc intuitif que meilleure sera notre connaissance a priori des conditions initiales du système, meilleure sera notre estimation grâce à un «encadrement» judicieux par les estimateurs simples. Nous ne pouvons pas cependant écarter la possibilité que les conditions initiales du système coïncident parfaitement ou presque avec les conditions initiales d'un des estimateurs, conduisant au phénomène expliqué précédemment. Il nous faut toutefois rajouter que le multi-estimateur restera alors meilleur que la moyenne de tous les estimateurs simples, ce qui implique que statistiquement, le multi-estimateur reste bien meilleur qu'un estimateur simple.

Maintenant que nous avons pu bien comprendre les mécanismes du multi-estimateurs, nous l'appliquons à notre quadricoptère. Le système étant de dimension 3 (positions, vitesses et forces), il nous faut quatre estimateurs, et ceci pour chaque direction (en x , y et z), ce qui

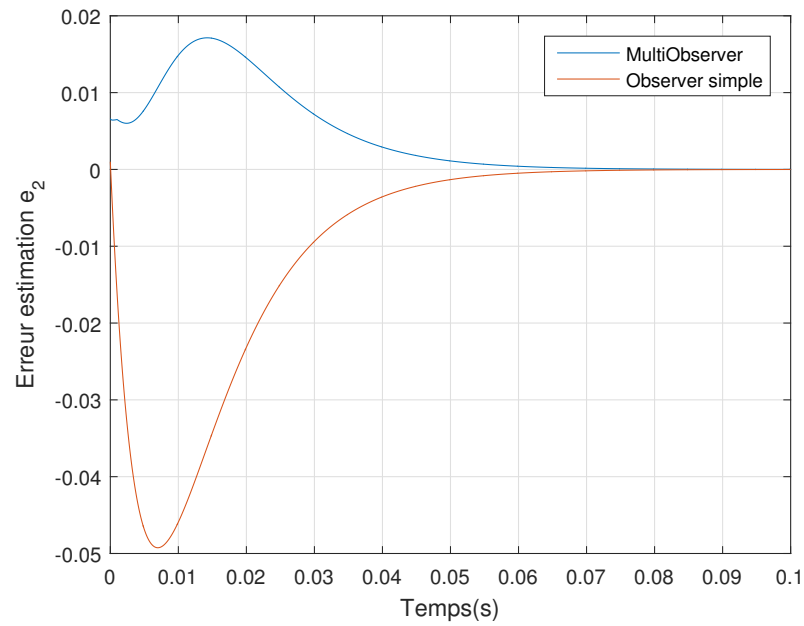


Figure 4.21 e_1 du meilleur estimateur simple et du multi-estimateur en fonction du temps avec $\frac{e_{simple}(0)}{e_{multi}(0)} = \frac{1}{10}$

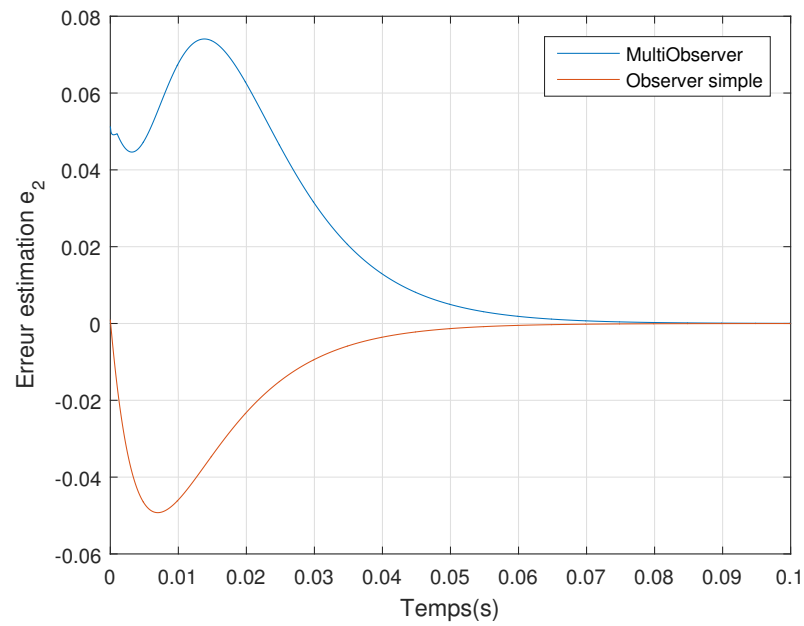


Figure 4.22 e_2 du meilleur estimateur simple et du multi-estimateur en fonction du temps avec $\frac{e_{simple}(0)}{e_{multi}(0)} = \frac{1}{40}$

nous amène à douze estimateurs. Nous aurons donc aussi trois algorithmes RLS permettant de calculer les coefficients α_i , un selon x , un selon y et un selon z . Le diagramme fonctionnel correspondant tiré de l'article de Bernat (Bernat and Stepien, 2015) est illustré sur la figure 4.13.

Un point important de ce multi-estimateur est le choix des conditions initiales de chaque estimateur simple. En connaissant a priori les conditions initiales (on suppose que le vent ne souffle pas à plus de 5m/s, que la vitesse du drone est inférieure à 0.1 m/s et que l'on connaît parfaitement la position initiale), nous pouvons quadriller l'espace en vitesse et en force autour de conditions initiales nulles (ni forces perturbatrices et une vitesse initiale nulle) afin que chaque estimateur ne soit pas trop loin des conditions initiales réelles, et ne soit donc pas instable. Les simulations nous donnent alors les résultats des figures 4.23, 4.24 et 4.25.

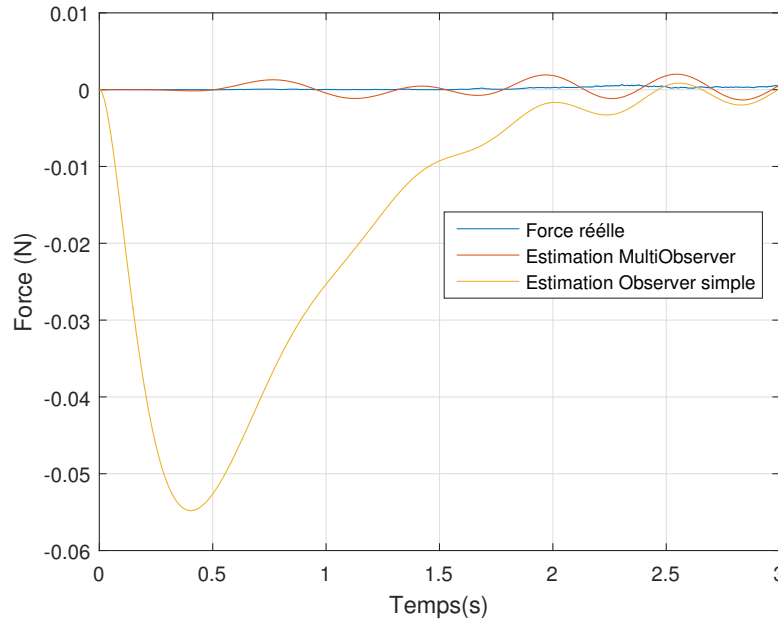


Figure 4.23 Comparaison de l'état réel, de l'état estimé par un estimateur simple et multiple de la force perturbatrice selon x

Comme nous pouvons le voir sur les figures précédentes, le multi-estimateur nous permet une diminution conséquente du pic d'instabilité dans les premières secondes de vol, garantissant de meilleures performances qu'avec un estimateur simple. Nous pouvons aussi constater une convergence des états estimés (et donc des performances du contrôleur) entre l'estimateur simple et multi-estimateur après quelques secondes, ce qui confirme bien l'intérêt du multi-estimateur seulement dans les premiers instants. Afin d'économiser de la puissance de calcul pour le reste du temps de vol, il serait intéressant de passer à un estimateur simple au bout

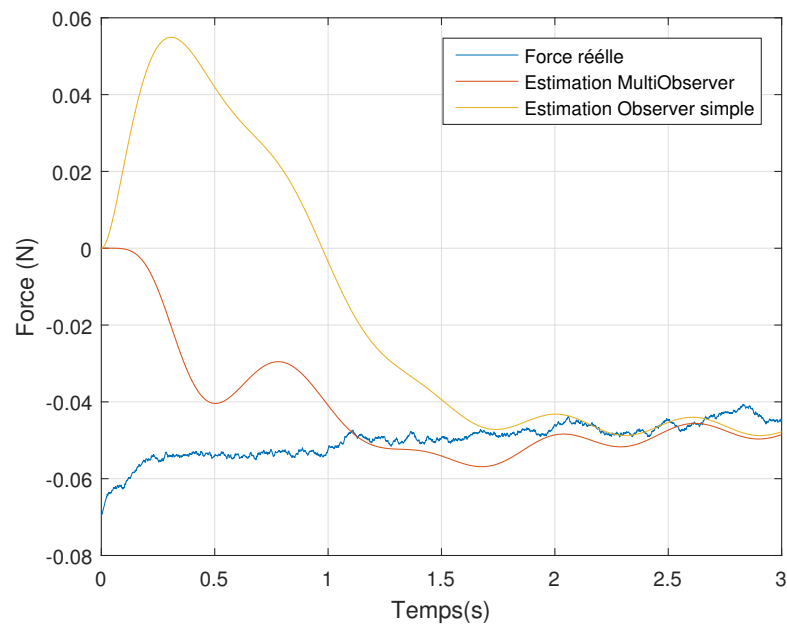


Figure 4.24 Comparaison de l'état réel, de l'état estimé par un estimateur simple et multiple de la force perturbatrice selon y

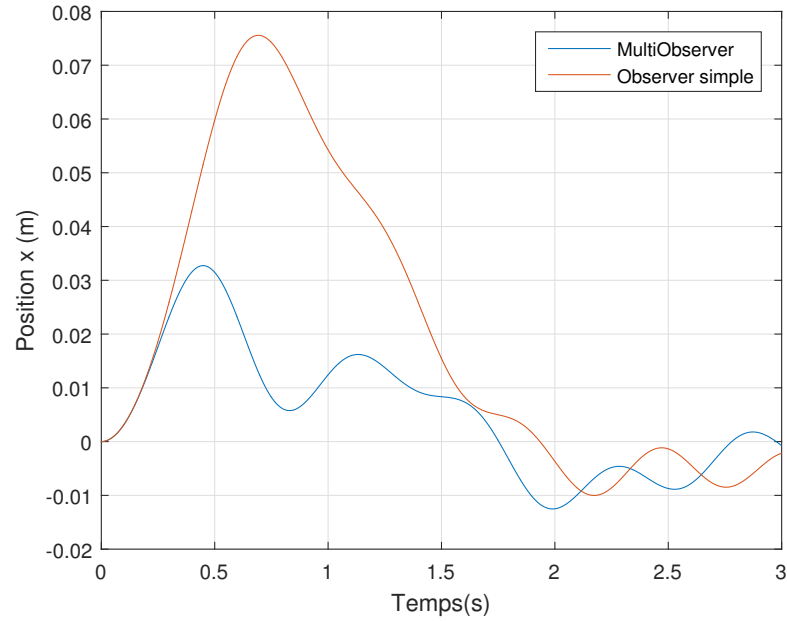


Figure 4.25 Comparaison des performances du contrôleur avec un estimateur simple et un multi-estimateur

de quelques secondes.

Comme montré dans cette partie, un bon estimateur est indispensable à l'amélioration du contrôleur. C'est pourquoi nous présenterons dans le chapitre suivant des résultats que l'on pourrait avoir si l'on arrivait à estimer parfaitement les perturbations.

CHAPITRE 5 COMPARAISON DES PERFORMANCES DE VOL DES DIFFÉRENTS CONTRÔLEURS EN SIMULATION

La comparaison des performances des différents contrôleurs a été réalisée en deux temps. Une première phase de validation préliminaire a été effectuée sur Matlab, puis, dans un second temps, nous avons réalisé des simulations similaires dans l’environnement ROS (*Robot Operating System*) et Gazebo nous permettant d’obtenir des résultats plus réalistes et complets. Nous avons de plus travaillé sur deux drones, i.e., le AscTec Pelican (Fig. 3.3) et le Crazyflie (Fig. 5.6). Pour toutes nos simulations, la position de référence est $x_d = y_d = z_d = 0\text{m}$ et le vent souffle selon l’axe x dans le sens positif à 5 m/s .

5.1 Résultats des simulations sur Matlab pour le drone AscTec Pelican

Nous allons présenter les résultats des simulations de manière analogue à l’article de référence (Tran et al., 2015), c’est-à-dire en prenant comme argument de performance les écarts-types. Afin de souligner l’importance de l’estimateur dans l’ADRC nous avons aussi fait la simulation d’un ADRC «parfait», c’est-à-dire que nous avons corrigé la commande avec les forces et moments réels plutôt que leurs estimés. Les figures 5.1 et 5.2 permettent de comparer qualitativement les performances des différents contrôleurs. Les contrôleurs utilisés pour ces simulations sont ceux présentés dans la partie 4.1 pour le LQR, 4.2 pour le PID et 4.3 pour l’ADRC, et les gains correspondants sont ceux donnés par le tableau 4.1 et le résultat 4.13. Nous donnons dans le tableau 5.1 les valeurs utilisées pour nos simulations.

Tableau 5.1 Valeurs des paramètres du drone AscTec utilisées pour les simulations

Paramètre	Valeur
m	1.7 kg
$I_x = I_y$	0.0364 kg.m ²
I_z	0.0486 kg.m ²

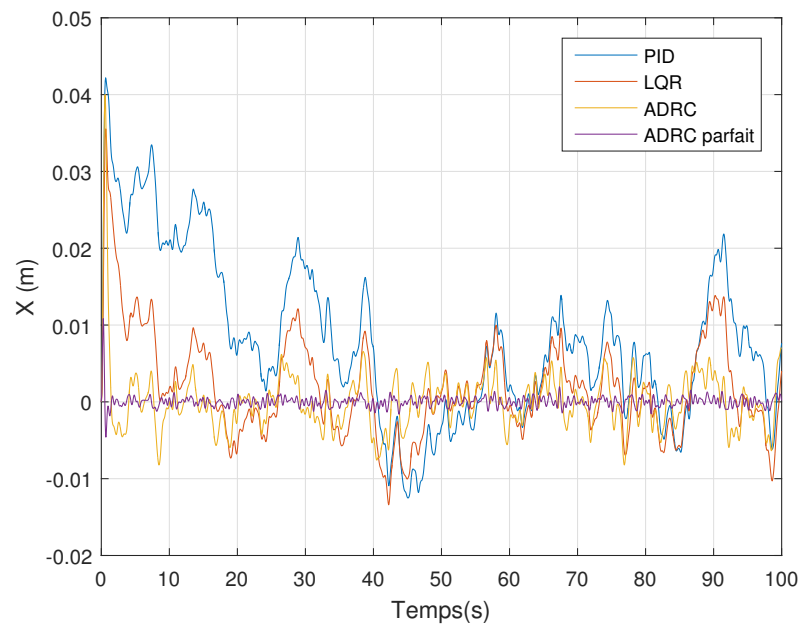


Figure 5.1 Position x du drone en fonction du temps pour les différents contrôleurs

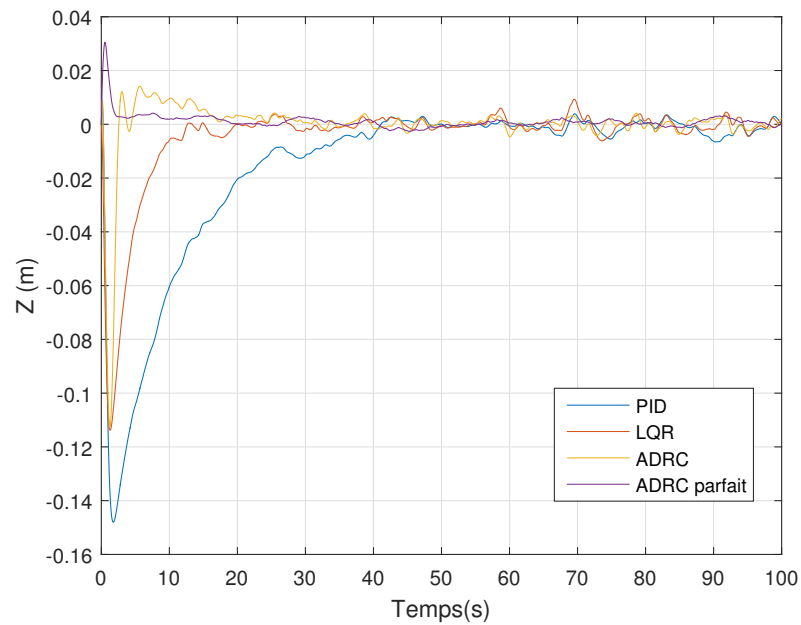


Figure 5.2 Position z du drone en fonction du temps pour les différents contrôleurs

L'objectif recherché est donc bien atteint pour le contrôleur ADRC qui arrive à mieux contrebalancer les forces du vent afin de bien rester à la position désirée. De plus le régime transitoire

est bien plus rapide avec l'ADRC.

En observant les angles d'Euler en fonction du temps (Figs. 5.3, 5.4 et 5.5), on s'aperçoit que l'ADRC et le PID nécessitent des variations plus rapides et de plus grandes amplitudes que le contrôleur LQR pour maintenir la position de référence. Ce résultat était attendu comme le contrôleur LQR cherche à économiser l'énergie utilisée par la commande de stabilisation du drone. Il s'agit donc d'un inconvénient en terme pratique car cela requiert des commandes plus grandes, donc plus énergivores et requiert potentiellement une fréquence de calcul plus rapide. On voit aussi que dans le cas de l'ADRC «parfait», le drone oscille presque autant que l'ADRC. Cette remarque confirme bien l'intuition que pour obtenir de meilleures performances nous devons dépenser plus d'énergie en calcul mais aussi en amplitude de commande. On pourrait penser que l'ADRC «parfait» devrait rester parfaitement immobile (vu qu'on compense exactement les forces perturbatrices), cependant la fonction de transfert de premier ordre des moteurs induit un léger retard sur ces compensations, ce qui explique les résultats observés.

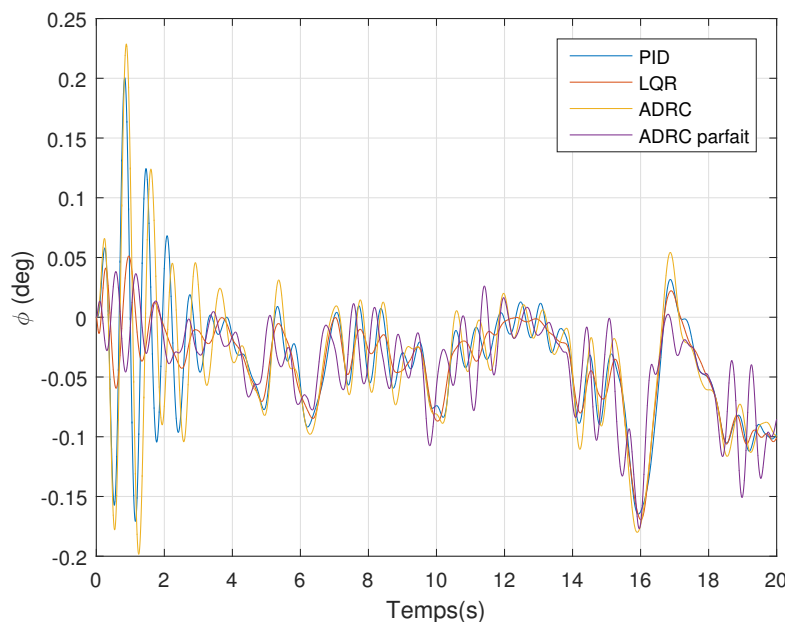


Figure 5.3 Angles d'Euler ϕ du drone en fonction du temps pour les différents contrôleurs

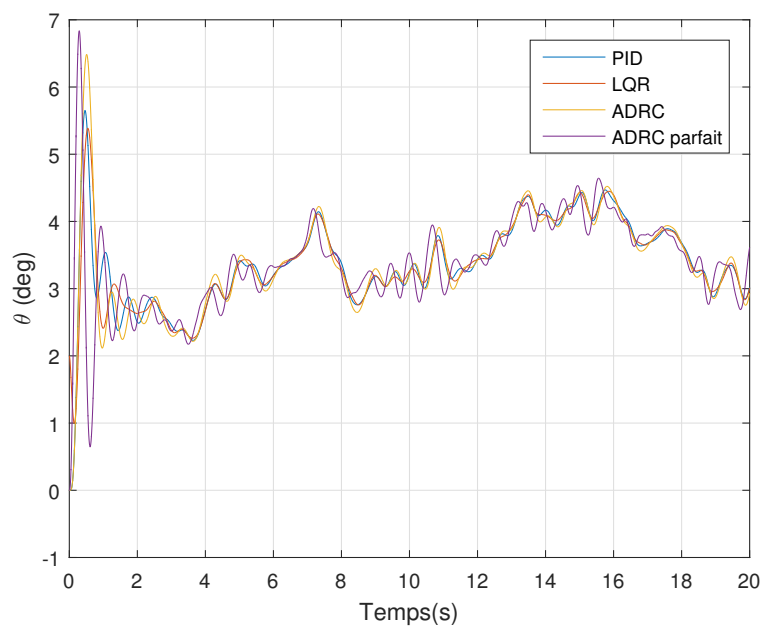


Figure 5.4 Angle d'Euler θ du drone en fonction du temps pour les différents contrôleurs

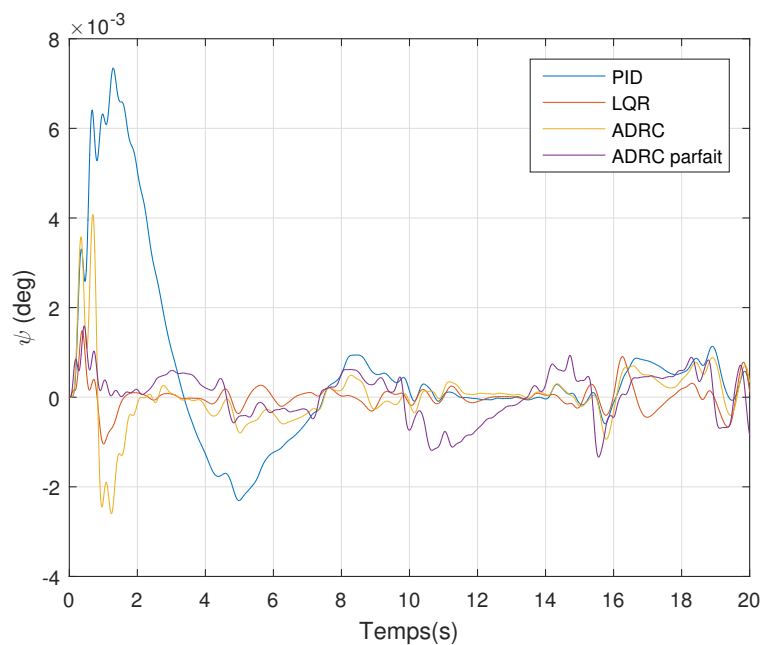


Figure 5.5 Angle d'Euler ψ du drone en fonction du temps pour les différents contrôleurs

On résume alors ces graphiques en les chiffrant via le calcul des écart-types. On ajoute aussi les performances d'un ADRC «parfait», qui utilise les forces perturbatrices réelles plutôt que

leurs estimées via l'estimateur ESO. On obtient alors le tableau 5.2.

Tableau 5.2 Écart-type selon les différents contrôleurs pour différentes grandeurs pour le drone AscTec Pelican

Variable	σ_x (m)	σ_y (m)	σ_z (m)	σ_ϕ (rad)	σ_θ (rad)	σ_ψ (rad)
LQR	5.4×10^{-3}	2.1×10^{-4}	2.5×10^{-3}	2.4×10^{-2}	0.80	1.1×10^{-4}
PID	7.6×10^{-3}	2.6×10^{-4}	3.7×10^{-3}	2.9×10^{-2}	0.80	3.1×10^{-4}
ADRC	3.1×10^{-3}	9.3×10^{-5}	1.8×10^{-3}	3.3×10^{-2}	0.81	2.6×10^{-4}
ADRC parfait	3.3×10^{-4}	5.8×10^{-5}	1.2×10^{-3}	2.5×10^{-2}	0.81	3.2×10^{-4}

L'ADRC «parfait» obtient encore de meilleurs résultats que l'ADRC que l'on a essayé d'optimiser. Il représente les performances maximales atteignables de l'ADRC basé sur le PID que l'on a présenté, en considérant que l'estimateur est parfait. On peut ainsi relier directement la performance de l'ADRC à son estimateur. Ceci nous montre l'importance cruciale de l'estimateur et de son optimisation afin de suivre au mieux le signal et ses dérivées (et donc les forces perturbatrices).

5.2 Résultats des simulations sur Matlab pour le drone Crazyflie

Afin de compléter les simulations faites pour le drone AscTec Pelican, nous avons réalisé une autre série de simulations pour le drone Crazyflie (Fig. 5.6), lui aussi soumis à un champ de vent de vitesse moyenne de 5 m/s selon x .



Figure 5.6 Drone Crazyflie

Les résultats des simulations sont présentés de manière similaires à ceux du drone AscTec Pelican, avec des graphiques montrant les positions en x et en z (Figs 5.7 et 5.8), ainsi que l'angle d'Euler θ en fonction du temps pour différentes vitesses d'estimation (Figs 5.9, 5.10

et 5.11). Ces figures nous permettent de nous rendre compte du retard des estimateurs en fonction de ω_0 . Les performances sont alors récapitulés dans le tableau 5.5, en différenciant plusieurs ADRC selon la vitesse du pôle le plus lent de son estimateur, en utilisant la technique de décorrélation des pôles. Nous donnons dans le tableau 5.3 les valeurs utilisées pour nos simulations.

Tableau 5.3 Valeurs des paramètres du drone Crazyflie utilisées pour les simulations

Paramètre	Valeur
m	0.027 kg
$I_x = I_y$	2.4×10^{-5} kg.m ²
I_z	3.2×10^{-5} kg.m ²

Les valeurs des gains PID utilisées pour les simulations avec le Crazyflie sont données par le tableau 5.4 :

Tableau 5.4 Valeurs des gains PID pour le Crazyflie

Coefficient	Valeur
k_{x,y_p}	0.02
k_{x,y_d}	0.05
k_{x,y_i}	0
k_{z_p}	3
k_{z_d}	0
k_{z_i}	2
k_{ϕ,θ_p}	3.5
k_{ϕ,θ_i}	2
k_{ψ_p}	0.04
k_{ψ_d}	0.04

Ceci nous permet à nouveau de nous rendre compte de l'importance du choix de la vitesse d'estimation de l'ADRC. On constate en effet des oscillations de plus grandes amplitudes si ω_0 est plus élevé, ce qui nous amène à des commandes plus gourmandes en énergie mais nous rapproche aussi de l'instabilité du drone. Il est à noter que les simulations pour $\omega_0 = 20$ donnent des oscillations en z et θ très grandes et rend les graphiques illisibles, c'est pourquoi la courbe associée n'y est pas affichée. Par souci de simplicité, nous avons choisi les mêmes vitesses d'estimation pour x , y et z . Il apparaît cependant que l'estimation en z devient instable plus rapidement qu'en x et y , comme le terme b est différent dans ces deux cas. Un ADRC ayant de bonnes performances nécessitent donc deux vitesses différentes pour les estimations en x et y , et celle en z . Les effets d'oscillations sont assez flagrants sur les courbes

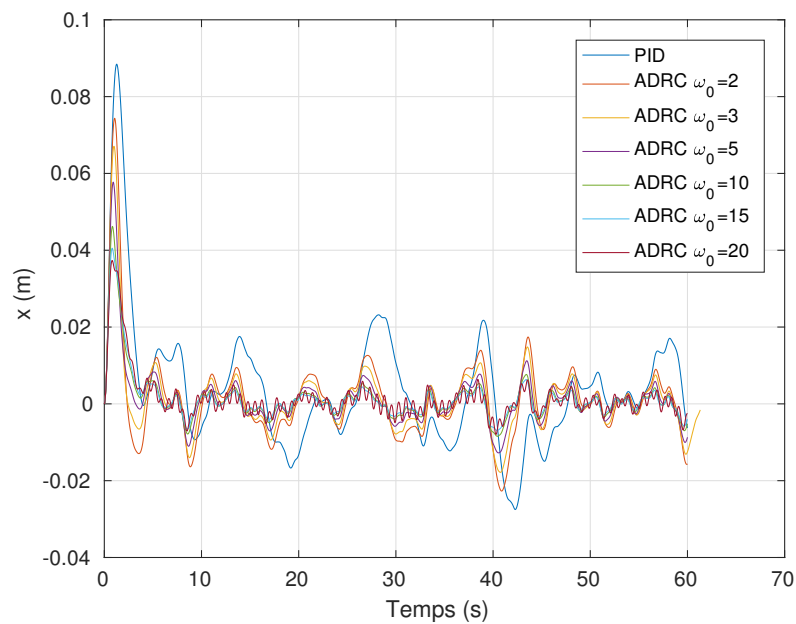


Figure 5.7 Position en x en fonction du temps pour un Crazyflie selon différentes vitesse d'estimation de l'ADRC

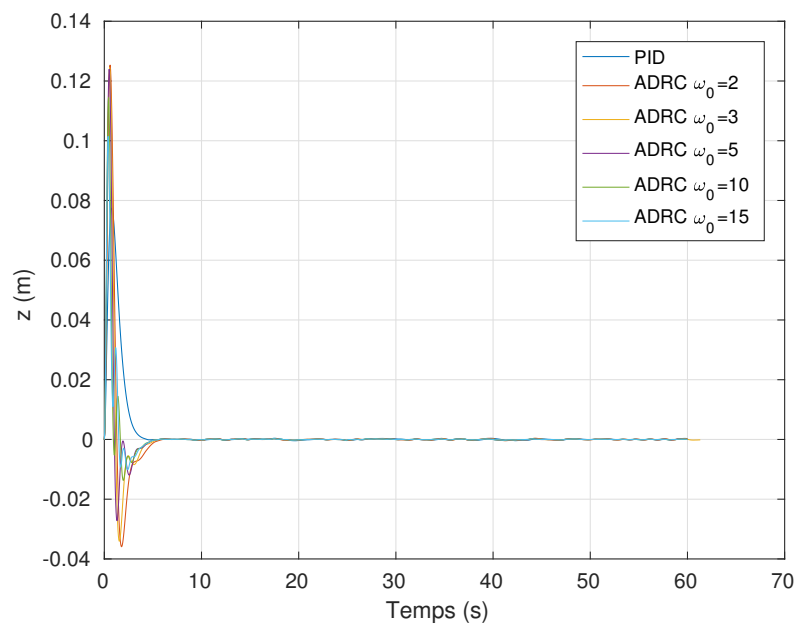


Figure 5.8 Position en z en fonction du temps pour un Crazyflie selon différentes vitesse d'estimation de l'ADRC

ci-dessous mais peuvent aussi être mesurés et exprimés via les écarts-types résumés dans le

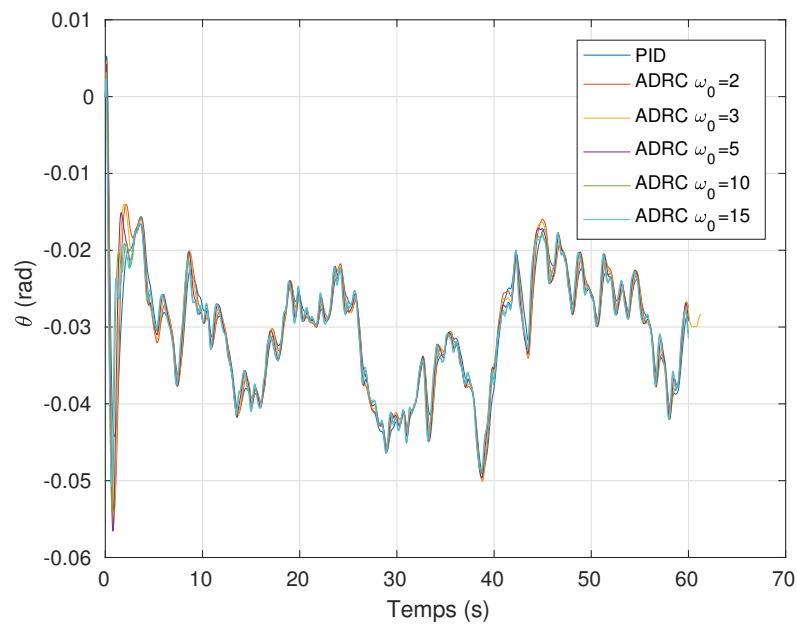


Figure 5.9 Angle θ en fonction du temps pour un Crazyflie selon différentes vitesse d'estimation de l'ADRC

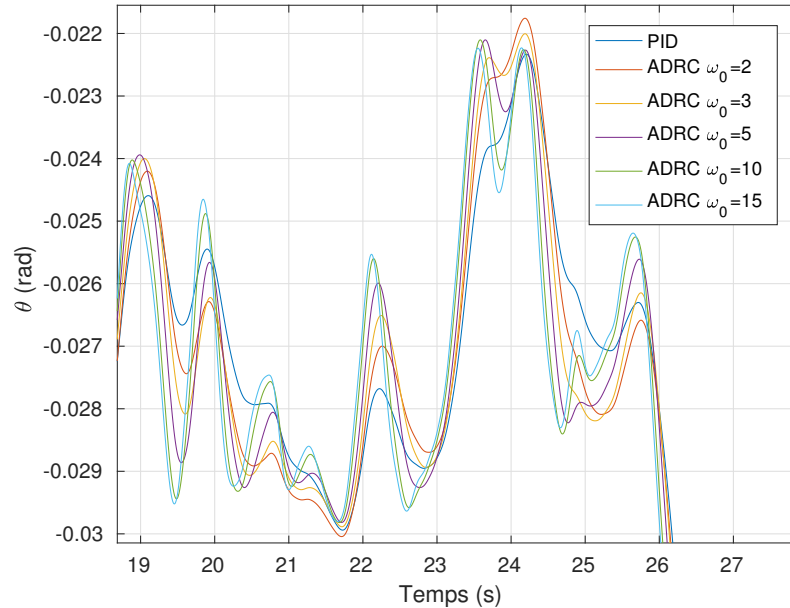


Figure 5.10 Zoom #1 sur la figure 5.9 représentant l'angle θ en fonction du temps pour un Crazyflie selon différentes vitesse d'estimation de l'ADRC

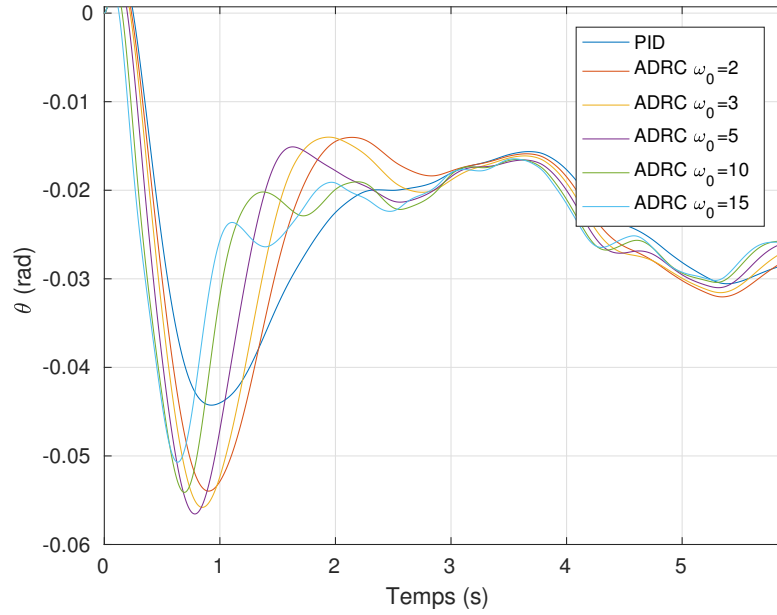


Figure 5.11 Zoom #2 sur la figure 5.9 représentant l'angle θ en fonction du temps pour un Crazyflie selon différentes vitesse d'estimation de l'ADRC

tableau 5.5.

Tableau 5.5 Écart-type selon les différents contrôleurs pour x , z et θ pour le drone Crazyflie

Variable	σ_x (m)	σ_z (m)	σ_θ (rad)
PID	1.1×10^{-2}	1.72×10^{-4}	7.6×10^{-3}
ADRC $\omega_0 = 2$	7.5×10^{-3}	1.68×10^{-4}	7.6×10^{-3}
ADRC $\omega_0 = 3$	6.0×10^{-3}	1.49×10^{-4}	7.6×10^{-3}
ADRC $\omega_0 = 5$	4.4×10^{-3}	1.24×10^{-4}	7.6×10^{-3}
ADRC $\omega_0 = 10$	3.0×10^{-3}	9.61×10^{-5}	7.5×10^{-3}
ADRC $\omega_0 = 15$	2.5×10^{-3}	8.69×10^{-5}	7.4×10^{-3}
ADRC $\omega_0 = 20$	9.5×10^{-3}	2.32×10^{-2}	8.9×10^{-3}

5.3 Résultats des simulation dans ROS et Gazebo

Nous présentons maintenant les résultats obtenus dans ROS et Gazebo, pour le drone Asc-Tec Pelican. Les simulations réalisées dans cet environnement (plus complet et précis) nous ont permis de trouver certains problèmes d'implémentation qui étaient passés inaperçus sur Matlab. En effet, bien que le contrôleur ADRC reste meilleur lorsqu'il doit maintenir une position constante malgré la présence de vent, il est moins bon lorsqu'on lui commande de se

déplacer. Pour deux versions différentes de l'ADRC et du PID sur lequel il est basé, la figure 5.12 donne les réponses temporelles suite à une commande de type échelon en position x .

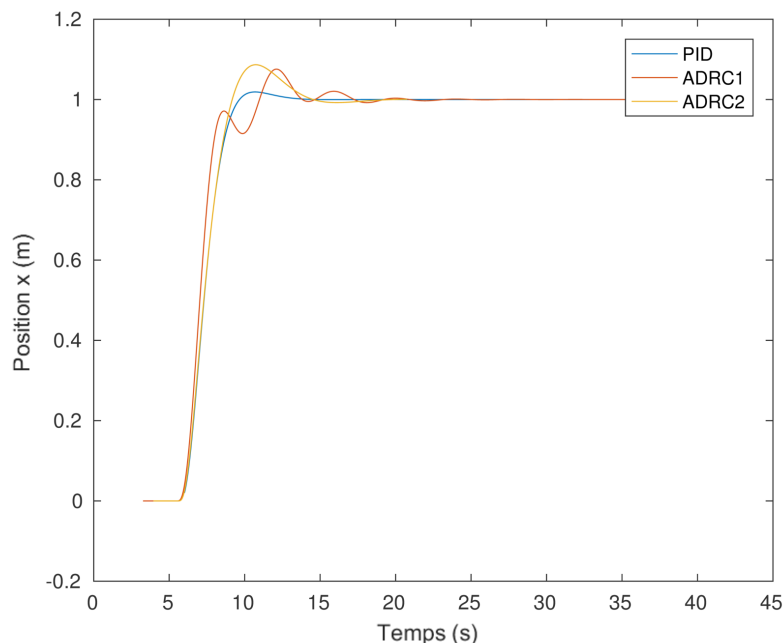


Figure 5.12 Comparaison de deux ADRC et du PID pour un échelon en x sans vent

Nous remarquons que la première version de l'ADRC utilisée (la même que pour les simulations Matlab) oscille fortement avant de se stabiliser. Nous avons pu déterminer que cette oscillation venait en fait de l'ESO qui estimait des forces perturbatrices inexistantes (car il n'y avait pas de vent dans cette simulation). Nous avons donc une erreur dans l'estimateur, qui provient a priori du terme \mathbf{Bu} de la commande.

En regardant le schéma fonctionnel utilisé jusqu'à présent (Fig. 5.14), on comprend facilement que le contrôle de position est erroné à cause de l'estimateur ESO qui prend comme entrée \mathbf{Bu} , non pas les angles d'Euler réels qui régissent la dynamique du quadricoptère mais les angles d'Euler désirés. Ainsi l'écart temporel entre ces deux grandeurs crée dans l'estimateur ESO une erreur d'intégration de la commande \mathbf{Bu} provoquant une erreur d'estimation des forces perturbatrices. La compensation de ces forces donnent alors lieu aux oscillations observées. La deuxième version de l'ADRC corrige ce défaut et envoie comme entrée \mathbf{Bu} de l'estimateur les angles d'Euler réels. Cette erreur de conception vient du fait que nous faisons du contrôle ADRC sur deux niveaux : le contrôle haut niveau, qui est le contrôle d'attitude, et le contrôle bas niveau qui est le contrôle de position. Afin de mieux comprendre l'erreur, rappelons la structure classique d'un ADRC (présenté ici sans TD) sur la figure 5.13.

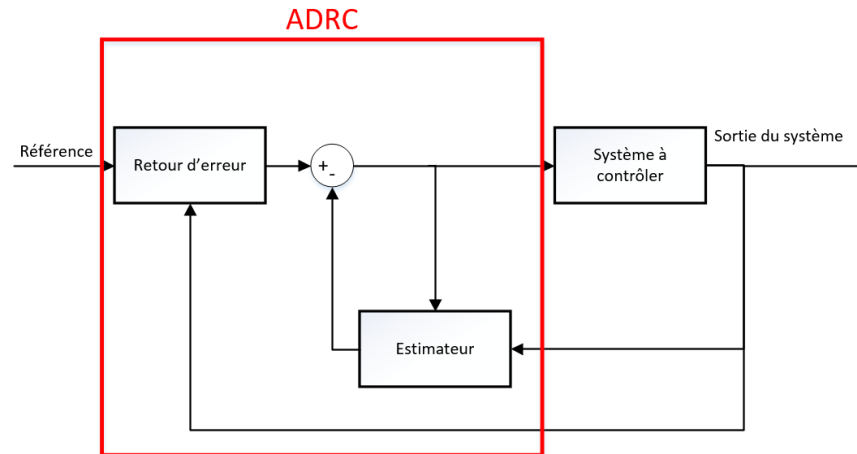


Figure 5.13 Structure classique d'un ADRC

Pour un contrôle à deux niveaux comme le nôtre, on considère généralement que le contrôleur haut niveau (ici contrôle en position) est bien plus lent que le bas niveau (contrôle d'attitude). Ceci nous permet de synthétiser le contrôleur haut niveau en considérant que le système associé qu'il contrôle est le système de base (ici notre quadricoptère) en rajoutant le contrôleur bas niveau (ce sous-système est en bleu dans la figure 5.14).

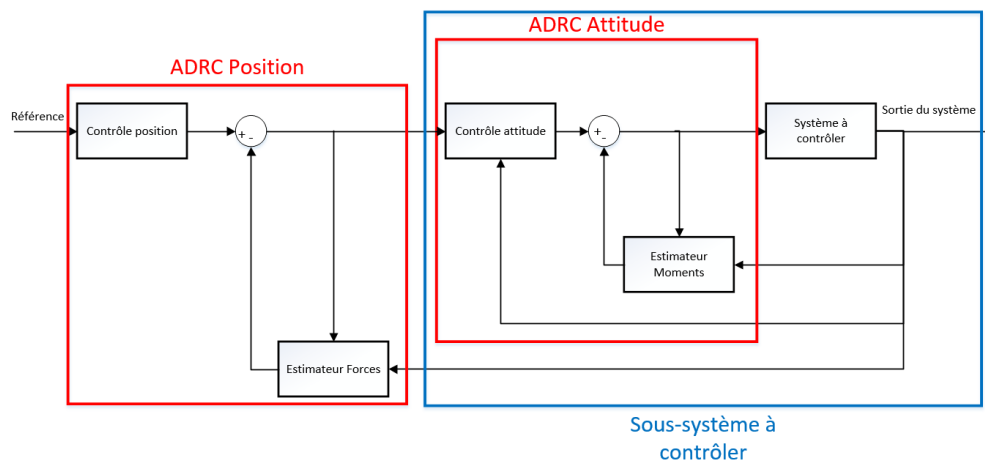


Figure 5.14 Structure intuitive d'un contrôle ADRC à deux niveaux

On voit bien ici que l'on prend comme entrée de l'ESO haut niveau les angles d'Euler désirés et non pas les angles d'Euler réels. Cette erreur est corrigée dans le schéma fonctionnel amélioré (Fig. 5.15).

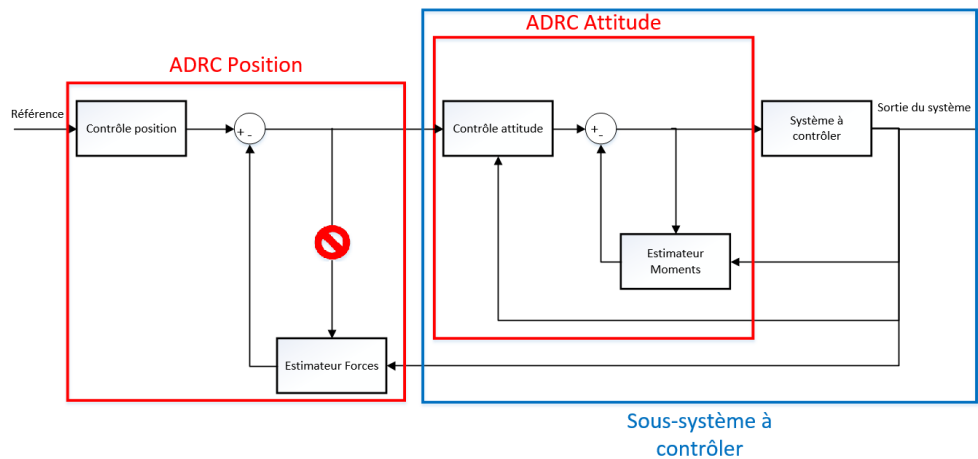


Figure 5.15 Structure corrigée d'un contrôle ADRC à deux niveaux

Ce problème ne s'était pas fait remarquer sur les simulations Matlab précédentes car le PID que nous avons utilisé pour contrôler l'attitude était bien plus rapide que le PID contrôlant la position. L'approximation $\theta_{\text{réel}} \approx \theta_{\text{ref}}$ était donc valable, mais elle ne l'était plus lorsque nous avons utilisé des PID différents sur ROS. En revanche une autre approximation que nous avons considérée comme valable mais qui aurait aussi pu créer des problèmes de commande se situe dans le contrôle d'attitude où l'on a supposé pour les vitesses de rotation des moteurs $\omega_{i,\text{réel}} \approx \omega_{i,\text{ref}}$. En effet on utilise comme entrée de l'estimateur de moments perturbateurs les vitesses de rotation des moteurs désirées. Cependant en pratique il y a un léger décalage entre les vitesses des moteurs voulues et les vitesses des moteurs réelles. Ce décalage est en général modélisé par une fonction de transfert du premier ordre induisant un certain retard. Dans notre cas, ce retard est seulement de l'ordre de 0.01 s, bien plus rapide que notre contrôleur PID d'attitude. L'approximation $\omega_{i,\text{réel}} \approx \omega_{i,\text{ref}}$ reste donc valable. Si cela n'était pas le cas, nous aurions dû prendre comme entrée pour l'estimateur des moments perturbateurs la vitesse angulaire réelle des moteurs, obtenue soit par mesure (ce qui est en général impossible), soit en la calculant à l'aide de la fonction de transfert du premier ordre mentionnée plus haut.

Malgré ce réarrangement, on observe que notre deuxième version de l'ADRC présente un plus grand dépassement de la position de référence que le PID associé. Encore une fois, nous pouvons constater que l'ESO estime une force perturbatrice lorsque le drone se déplace. Cette force perturbatrice est tout simplement la force de traînée (frottements avec l'air) de l'appareil. Cette force n'est pas prise en compte dans le modèle que l'on a construit pour l'estimateur. Ainsi le ralentissement provoqué par les frottements sont considérées par l'estimateur comme des forces perturbatrices et sont donc compensées en générant une force

de poussée supplémentaire, ce qui conduit à un plus grand dépassement de la position de référence. Afin d'éviter ce dernier inconvénient, nous pouvons, soit incorporer le modèle de frottement dans l'estimateur (ce qui nécessite une modélisation avancée du système), soit tout simplement régler à nouveau le PID afin de le suramortir. Comme nous avons accès au coefficient de frottement de l'appareil, nous pourrions éventuellement injecter les forces de frottements dans l'estimateur, mais il était plus rapide d'ajouter ce coefficient au terme $k_{x,y}$ et k_z , permettant d'éviter la surcompensation de l'estimateur :

$$\ddot{x} = w + Bu - C_x \dot{x} = w - k_{x_p}x - k_{x_i} \int x dt + k_{x_d} \dot{x} - C_x \dot{x} \quad (5.1)$$

avec C_x étant le coefficient de trainée selon x . En remplaçant le terme $k_{x,d}$ par $k_{x,d} + C_x$, les forces de frottements sont alors compensées directement dans la commande, et nous obtiendrons donc la dynamique désirée à la base en évitant que l'estimateur surcompense ces forces perturbatrices. On obtient avec cette configuration le même amortissement que pour le PID qui avait l'amortissement souhaité.

Après ces tests préliminaires sur ROS qui nous ont permis de nous rendre compte de certains phénomènes, nous avons pu refaire les mêmes simulations effectuées sur Matlab. La première étape était bien sûr de vérifier les meilleures performances de l'ADRC par rapport à son PID associé sur la figure 5.16.

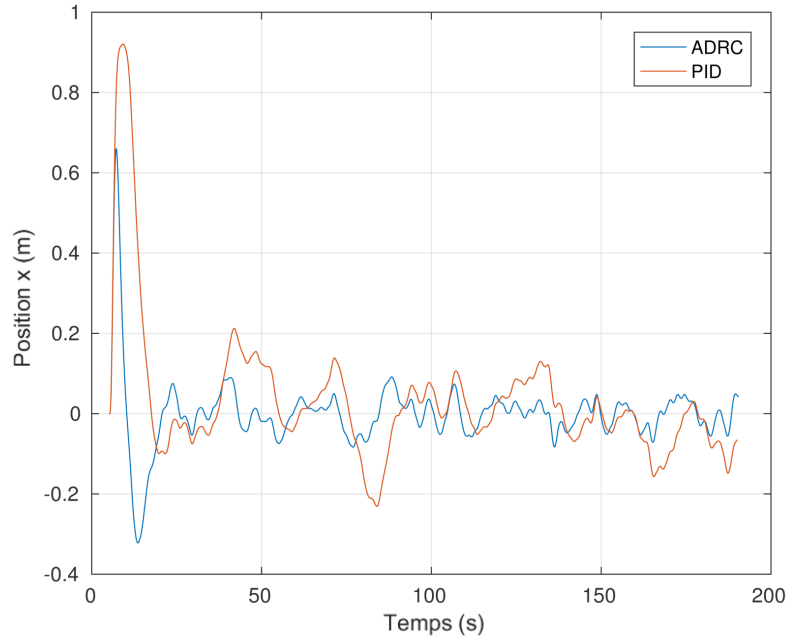


Figure 5.16 Comparaison du contrôleur PID et ADRC associé

Les résultats de cette simulation préliminaire nous donne les écart-types du tableau 5.6.

Tableau 5.6 Écart-type en position pour le PID et son ADRC associé

Variable	σ_x (m)
PID	0.1760
ADRC	0.0880

Nous avons ensuite fait différentes batteries de tests afin de comparer les différentes structures possibles pour le contrôleur ADRC. Nous avons d'abord observé les différences de performance de l'ADRC si l'on ajoute un estimateur dans le contrôle d'attitude afin de rejeter les moments induits par le vent. Une nette amélioration a été constatée sur la figure 5.17 et le tableau 5.7, bien que les moments que le vent induit sur le drone soient faibles.

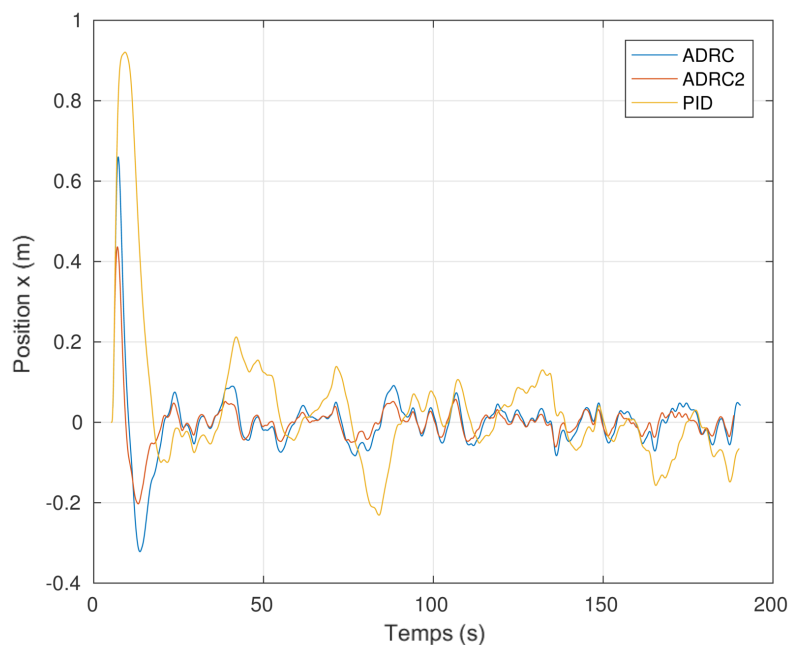


Figure 5.17 Comparaison du PID, de l'ADRC sans rejet des moments perturbateurs (ADRC1) et avec (ADRC2)

Nous avons aussi testé une version de l'ADRC basé sur un PD pour le contrôle de position et un PID pour le contrôle d'attitude pour le comparer à sa structure inversée (PID pour la position et PD pour l'attitude). Ces contrôleurs ainsi que les contrôleurs ADRC associés nous donnent les résultats présentés dans la figure 5.18.

Le contrôleur PD en position dérive naturellement et semble se stabiliser à une certaine distance de la position de référence. Ce comportement est dû à l'absence du terme intégral

Tableau 5.7 Écart-type en position du PID, de l'ADRC sans rejet des moments perturbateurs (ADRC1) et avec (ADRC2)

Variable	σ_x (m)
PID	0.1760
ADRC	0.0880
ADRC2	0.0559

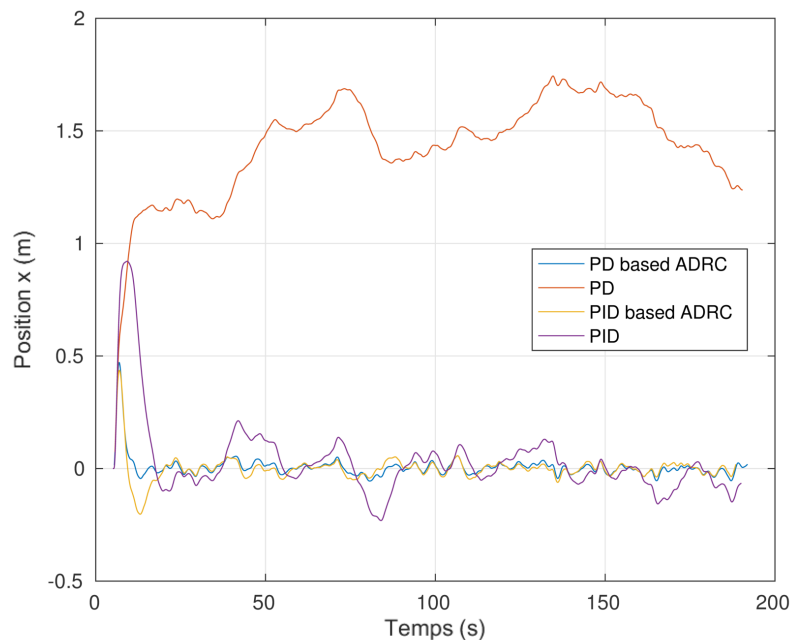


Figure 5.18 Comparaison des PD-PID et PID-PD ainsi que leur contrôleur ADRC associé

qui permet de compenser les forces du vent. Nous constatons en revanche que le contrôleur ADRC associé est meilleur que le PID en position. Intuitivement, cela s'explique par le fait que nous ajoutons deux types de compensations différentes pour les perturbations, alors que seul l'estimateur de l'ADRC suffit. Les écart-types correspondant sont présentés dans le tableau 5.8.

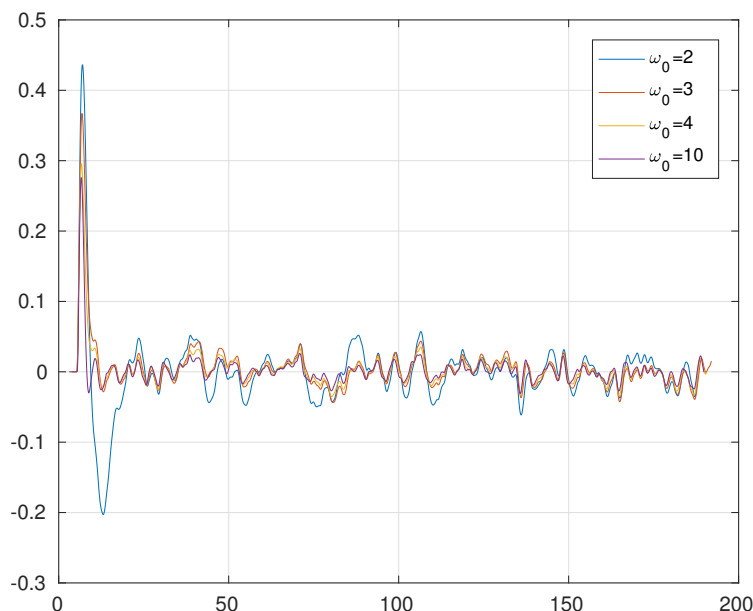
Nous avons aussi de nouveau vérifié l'influence de la rapidité de l'estimateur sur les performances du contrôleur, en faisant varier ω (Fig. 5.19).

Les écart-types calculés sont dans le tableau 5.9.

On voit donc bien que plus l'estimateur est rapide, meilleures sont les performances du contrôleur, mais il y a un léger inconvénient à le choisir trop rapide comme on peut s'en rendre compte dans la figure 5.20, car le quadricoptère oscille beaucoup plus, et consomme

Tableau 5.8 Écart-type en position pour le PID et son ADRC associé

Variable	$\sigma_x(\text{m})$
PID	0.1760
ADRC basé sur PID	0.0559
ADRC basé sur PD	0.0489

Figure 5.19 Comparaison des contrôleurs ADRC pour différents ω

donc plus.

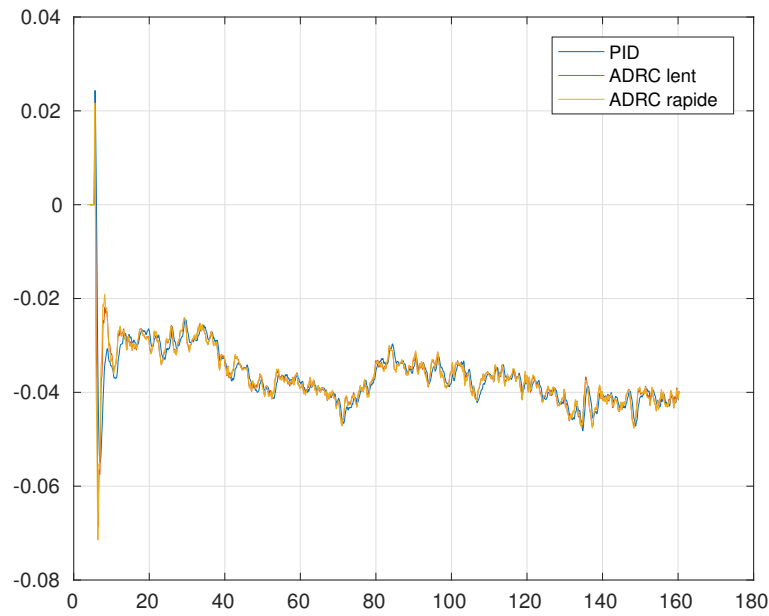
Les écart-types correspondants sont donnés dans le tableau 5.10.

On voit bien dans le tableau 5.10 qu'un estimateur trop rapide mène à des oscillations plus fortes que le PID de base. Nous avons donc là encore un compromis à trouver entre rapidité et stabilité. Avec tous les essais effectués, nous avons fixé $\omega = 5 \text{ rad/s}$, qui nous garantit de très bonnes performances face aux perturbations ainsi qu'une bonne stabilité. L'absence de fortes oscillations sur les angles d'Euler permet d'économiser de l'énergie (commandes d'amplitudes moins élevées) comparé à un estimateur plus rapide.

Le réalisme des simulations effectuées dans l'environnement ROS/Gazebo nous a permis de constater et de corriger quelques erreurs dans notre modèle, erreurs que nous n'avions pas pu observer sur Matlab. Nous allons maintenant pouvoir présenter les résultats expérimentaux obtenus au sein du laboratoire de drones pour différents réglages du contrôleur ADRC.

Tableau 5.9 Écart-type en position pour les différents ω

Variable	σ_x (m)
$\omega_0 = 2$	0.0489
$\omega_0 = 3$	0.0369
$\omega_0 = 10$	0.0240

Figure 5.20 Comparaison des contrôleurs ADRC en θ pour différents ω Tableau 5.10 Écart-type en θ pour les différents ω

Variable	θ (rad)
PID	0.0068
$\omega_0 = 5$	0.0066
$\omega_0 = 10$	0.0071

CHAPITRE 6 COMPARAISON EXPÉRIMENTALE DES PERFORMANCES EN VOL DES DIFFÉRENTS CONTRÔLEURS

Ce chapitre rassemble les résultats expérimentaux obtenus au sein du Laboratoire de Robotique Mobile et des Systèmes Autonomes (*Mobile Robotics and Autonomous Systems Laboratory* - MRASL). Nous décrivons d’abord le mode opératoire utilisé pour obtenir ces résultats ainsi que les différents drones utilisés, avec les spécificités d’implémentation pour chacun. Obtenir des séries de mesures sur un drone commercial permet de rendre compte de la difficulté pratique d’implémentation selon le drone, mais aussi d’évaluer l’efficacité de notre algorithme de contrôle comparé à celui du constructeur.

6.1 Mode opératoire des expériences

Nous avons cherché à reproduire les conditions de simulation présentées dans les parties précédentes. Pour cela, le drone à tester avait pour consigne de rester en vol stationnaire dans la volière prévue à cet effet. Nous y avons installé un ventilateur permettant de produire des vents de vitesse moyenne variant de 1 à 5 m/s dépendamment de la puissance utilisée et de la distance à laquelle nous disposons le drone. Un système de suivi de position, le Vicon Tracker

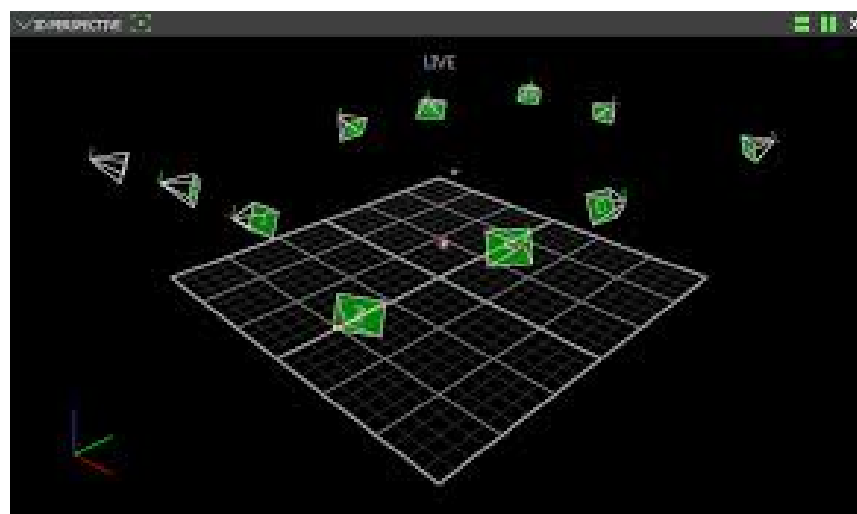


Figure 6.1 Vicon Tracker System

(Fig. 6.1) disposé tout autour de la volière et composé de douze caméras infrarouges (Fig. 6.2) permet d’avoir accès en temps réel à la position du drone avec une précision de l’ordre du millimètre, à l’aide de marqueurs collés sur ce dernier. Afin de comparer les performances des

contrôleurs, nous mesurons l'écart-type en position obtenu par chacun d'eux durant les vols en fonction de la vitesse du vent, mesurée par un anémomètre électronique. La vitesse du vent peut être modifiée en changeant la vitesse de rotation des pales (3 positions), ou en modifiant la distance entre le drone et le ventilateur. Le champs de vitesse du vent est certainement différent de celui du modèle de Dryden utilisé en simulation (avec, comme différence la plus notable, un profil non-homogène dans l'espace), mais nous aurons tout de même une bonne approximation des perturbations extérieures induites par le vent.



Figure 6.2 Caméra du Vicon Tracker System

6.2 Résultats obtenus avec le Rolling-Spider

Nous présentons ici les résultats expérimentaux obtenus sur le drone commercial Rolling-Spider de la compagnie Parrot (Fig. 6.3). Ce drone ne mesure que 8.5 cm d'envergure pour 65 g, et est donc extrêmement sensible au vent du fait de sa légèreté. Les vitesses de vent utilisées pour les expériences sont donc relativement faibles (de 1 à 3 m/s), afin d'éviter les crashes. Une autre spécificité de ce drone sont les capteurs dont il dispose. En effet, le Rolling-Spider possède un gyroscope à trois axes, un accéléromètre à trois axes, un sonar, un capteur de pression et une caméra. Le drone n'a donc pas accès à sa position exacte mais à l'estimation de sa position et de son orientation en utilisant un algorithme se basant sur la fusion des mesures obtenues des différents capteurs. Le contrôleur nominal du Rolling-Spider est un simple contrôleur PD pour maintenir la position, et un contrôleur PID pour le maintien en attitude. L'implémentation de l'ADRC paraît donc compromise étant donné que l'algorithme de contrôle que l'on a utilisé en simulation se basait sur un retour d'état comprenant la position exacte du drone. Il est cependant intéressant de connaître les performances du



Figure 6.3 Mini-drone Rolling-Spider de Parrot

contrôleur ADRC sur un drone ne disposant pas de GPS ou autre système permettant de faire un retour d'état en position et en orientation. En effet on peut s'attendre à une amélioration moindre qu'avec un retour d'état complet. On pourrait aussi imaginer un contrôleur qui estime les forces perturbatrices directement avec les accéléromètres plutôt que de repasser par une estimation de position, mais cette possibilité n'a pas été étudiée dans ce travail.

Nous présentons les résultats expérimentaux de la même manière que pour les simulations, avec les graphiques des trajectoires présentés sur les figures 6.4 et 6.5. On peut déjà remarquer à ce stade que les vols effectués avec l'ADRC compensent mieux les effets du vent.

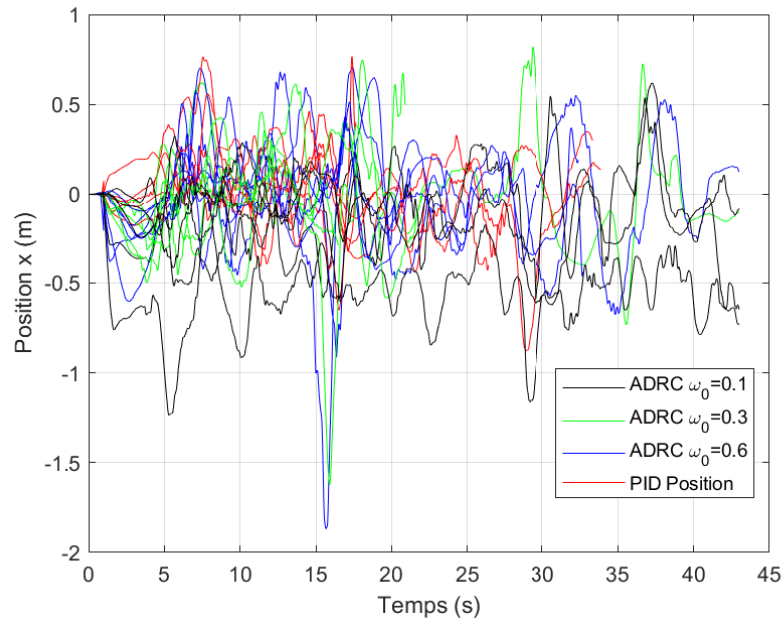


Figure 6.4 Position estimée en x du drone pour les différents contrôleurs

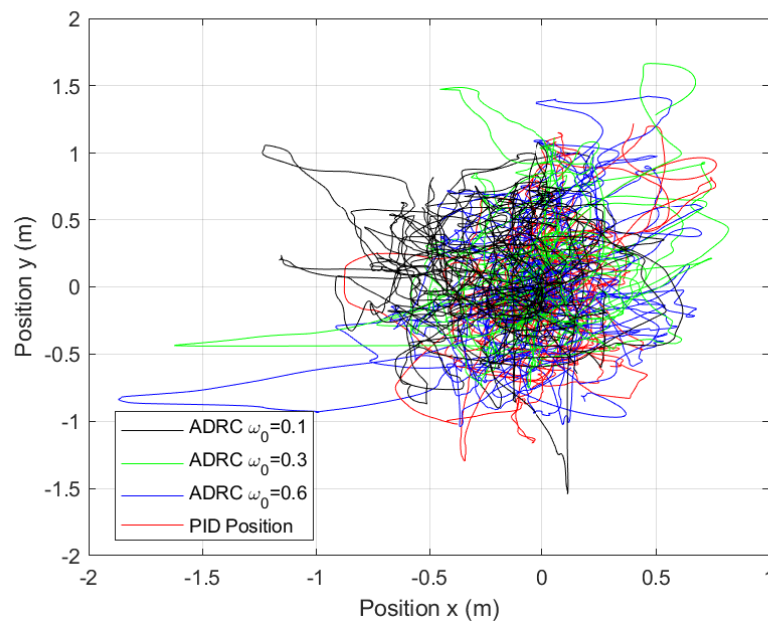


Figure 6.5 Trajectoires estimées du drone en x et y pour les différents contrôleurs

Il convient cependant de comparer ces données estimées avec des valeurs plus précises fournies par le Vicon (Figs 6.6 et 6.7).

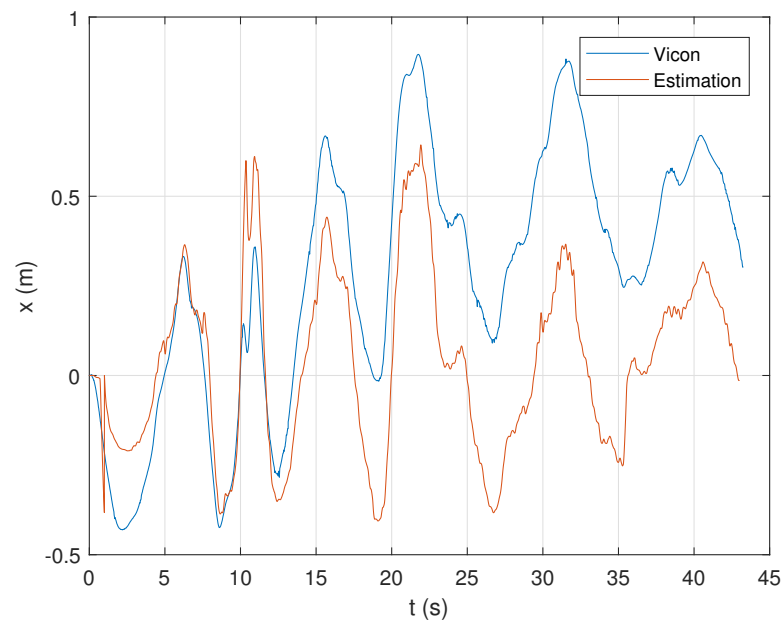


Figure 6.6 Comparaison de la position x estimée par le drone et par le Vicon

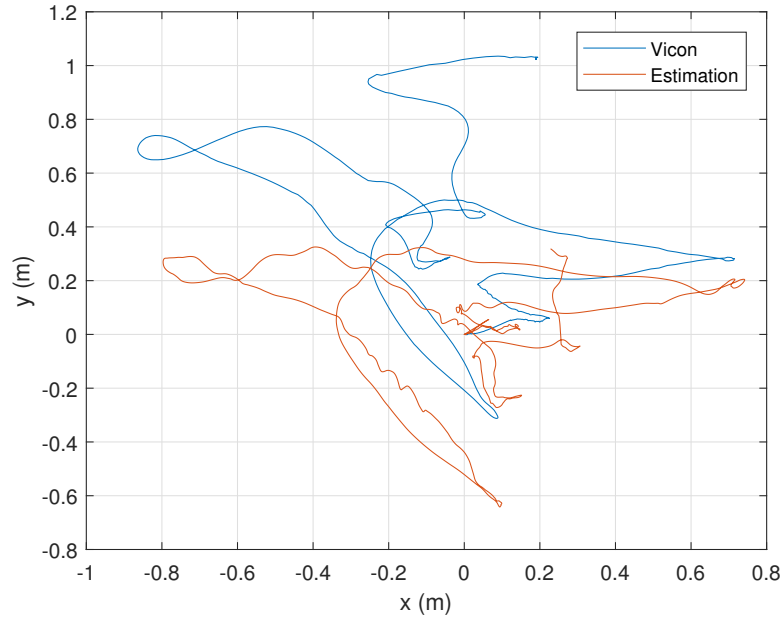


Figure 6.7 Comparaison de la trajectoire x et y estimée par le drone et par le Vicon

Comme on peut le voir sur les figures, les positions estimées par le Rolling-Spider et par le Vicon sont proches mais se décalent lorsque les variations de position sont grandes. Ce décalage occasionnel explique pourquoi on a un si grand écart entre les deux estimations après un certain temps de vol. Effectivement pendant les expériences on pouvait noter une faible dérive du drone Rolling-Spider dû à ce décalage d'estimation. L'ADRC se basant sur cette estimation, nous avons limité les temps de vol à 45 secondes chacun (afin de limiter ce décalage), puis nous en avons accumulé plusieurs (entre 6 et 10 de 45 secondes) afin d'avoir des résultats statistiques significatifs.

Les écarts-types ont ensuite été calculés sur les temps de vol cumulés d'environ 5 minutes en x - y et en z pour les différents contrôleurs. Les résultats sont présentés dans le tableau 6.1, pour les différents contrôleurs ADRC et le PID en fonction des estimations. Il convient de préciser que le contrôleur de base du Rolling-Spider (un PD en position) n'est absolument pas robuste au vent et a tendance à être instable (crash du drone systématique). On a donc rajouté un intégrateur afin que l'on ait un point de comparaison pour les ADRC. Aussi, dans cette section, les contrôleurs ADRC sont seulement munis d'un rejet des forces et non de moments. Nous avons modifié légèrement le critère de performance des contrôleurs. En effet, au lieu de simplement tenir compte de l'écart-type en position dans le sens du vent (ici en y) comme dans les simulations (où les dérives en x et y sont seulement corrélées aux vitesses de vent associées), nous avons tenu compte de l'écart-type en position en x et en y , combinés

sous la mesure $\sigma_{xy} = \sqrt{\sigma_x^2 + \sigma_y^2}$.

Tableau 6.1 Écart-type selon les différents contrôleurs pour différentes grandeurs pour le drone Rolling-Spider

Contrôleur	Estimateur interne		Vicon	
	σ_{xy} (m)	σ_z (m)	σ_{xy} (m)	σ_z (m)
PID	0.4981	0.2083	0.6550	0.2525
ADRC $\omega_0 = 0.1$ rad/s	0.4044	0.1940	0.5758	0.2072
ADRC $\omega_0 = 0.3$ rad/s	0.4246	0.1945	0.5557	0.2167
ADRC $\omega_0 = 0.6$ rad/s	0.4281	0.2229	0.4740	0.2548
ADRC $\omega_0 = 1$ rad/s	0.4675	0.1985	0.5580	0.2187

Afin de faciliter la lecture du tableau 6.1, nous en avons représenté son graphe, pour mieux comparer les valeurs d'écart-type en position estimées par le drone et par le Vicon :

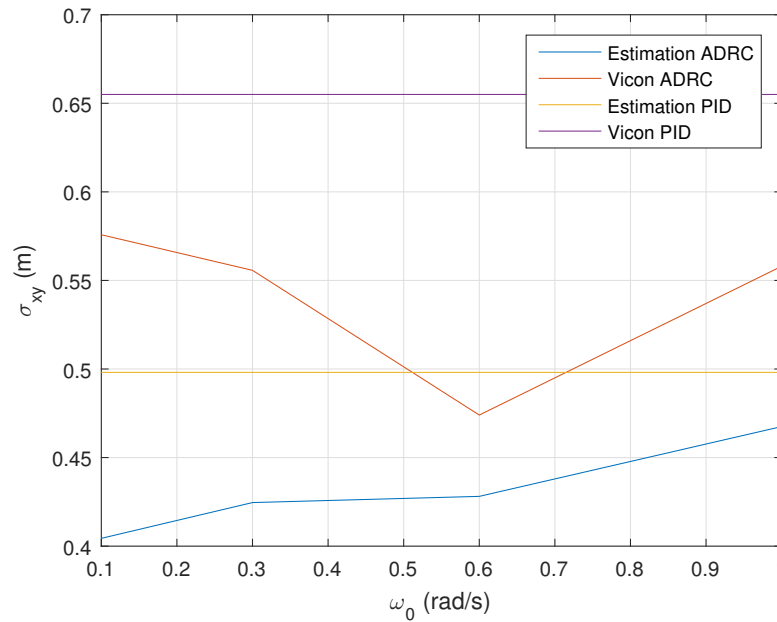


Figure 6.8 Comparaison des écarts-types en fonction de la vitesse d'estimation selon la méthode d'estimation de la position

Alors que les mesures de position par l'estimateur interne du drone et du Vicon ont été effectuées sur les mêmes vols, on voit bien une différence notable dans nos résultats. En effet l'estimateur interne a tendance à sous-estimer la dérive de la trajectoire du drone. On y voit aussi toute l'importance de la précision du Vicon Tracker. On peut dégager de ses mesures la vitesse d'estimation de l'ADRC optimale pour le Rolling-Spider, résultat qui n'aurait pas

été visible avec seulement l'estimation de position interne du drone.

Un autre critère important quant à la performance du contrôleur est sa consommation énergétique. Nous avons donc comparé les puissances consommées par les moteurs du drone selon les différents contrôleurs ainsi que l'énergie moyenne utilisée pour un vol de 30 secondes. Il est intuitif de penser qu'une meilleure performance se fait au détriment d'une augmentation de la consommation de la batterie, c'est pourquoi il est important de quantifier cet aspect des contrôleurs.

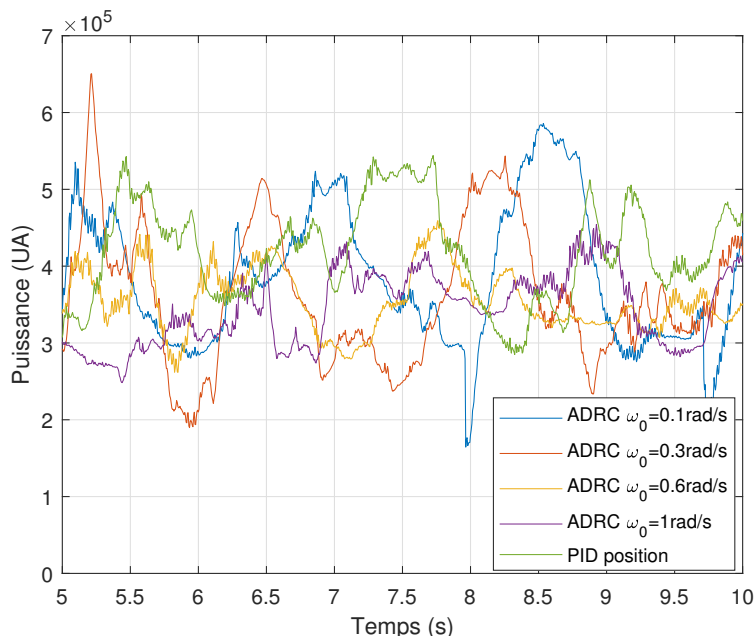


Figure 6.9 Comparaison des puissances (en unité arbitraire) consommées par les moteurs des différents contrôleurs au cours du temps

La figure 6.9 nous permet de visualiser les puissances délivrées par les moteurs au cours du temps, et donc la consommation de la batterie. Il est cependant difficile d'établir directement quel contrôleur est le plus efficace énergétiquement. C'est pourquoi la comparaison des énergies moyennes consommées du tableau 6.2 nous donne une comparaison statistique effectuée sur des temps de vols cumulés d'environ 5 minutes, évaluant ainsi la batterie consommée pour des conditions de vols similaires.

Le contrôleur ADRC qui donne les meilleures performances de vol (celui dont la vitesse d'estimation est de 0.6 rad/s) est aussi celui dont la consommation énergétique est minimale. Elle reste cependant légèrement supérieure à la valeur du PID. Il paraît évident qu'une amélioration des performances de vol nécessite une plus grande énergie pour mieux compenser les forces du vent. Le contrôleur ADRC avec la vitesse d'estimation optimale entraîne donc une

Tableau 6.2 Énergie consommée pour les différents contrôleurs pour 5 minutes de vol avec un vent de 3 m/s

Contrôleur	Énergie consommée (en unité arbitraire)
PID	3.64
ADRC $\omega_0 = 0.1$ rad/s	3.90
ADRC $\omega_0 = 0.3$ rad/s	3.80
ADRC $\omega_0 = 0.6$ rad/s	3.68
ADRC $\omega_0 = 1$ rad/s	4.05

augmentation des performances de 15% pour une augmentation de consommation d'énergie de seulement 1%. Le contrôleur ADRC remplit donc ses promesses en nous garantissant à la fois une nette amélioration de performance d'un contrôleur PID, tout en restant peu énergivore.

CHAPITRE 7 CONCLUSION ET RECOMMANDATIONS

Nous avons pu remplir notre objectif de concevoir un contrôleur ADRC plus robuste que le contrôleur PID nominal associé en simulation et en tests de vol. Ce résultat est très prometteur et nous laisse encore beaucoup de possibilités d'améliorations, dépendamment des objectifs que l'on veut se fixer. Un approfondissement des connaissances sur le fonctionnement du LESO paraît indispensable pour une meilleure optimisation de son estimation, et donc des performances de ce type de contrôleur.

L'ADRC parfait nous a permis d'observer la borne supérieure des performances atteignables par le LESO en simulation. Mais on peut encore imaginer plusieurs moyens d'améliorer ce contrôleur en utilisant les autres idées développées par J. Han (Han, 1994) comme des retours d'état non-linéaires plutôt qu'un PID standard, ou encore d'autres formes d'ESO. Le choix de la vitesse d'estimation de l'ADRC est apparue comme étant cruciale pour la performance et la robustesse du contrôleur. En réalisant des simulations et des expériences pour différentes structures de quadricoptère, nous avons pu déterminer la vitesse d'estimation optimale pour chaque drone (en simulation ou en laboratoire). Il pourrait cependant être intéressant de se pencher sur la prédiction du ω_0 optimal en fonction des paramètres du drone. Des graphes reliant la masse du drone et la vitesse d'estimation des forces, mais aussi en reliant son inertie à la vitesse d'estimation des moments, nous permettrait d'en savoir plus sur une éventuelle corrélation. Pour cela, il nous faudrait multiplier les mesures (ou les simulations) avec des drones de masse et d'inertie différentes, ce qui aurait été difficile de réaliser en laboratoire dans le cadre de ce projet. Une telle étude permettrait de se passer de cette phase de réglages préalable afin d'effectuer une implantation directe sur les drones. Il paraît aussi important de souligner que les quadricoptères commerciaux sont souvent munis d'un contrôleur PD en position, ce qui les rend très peu robustes, voire instables, pour des perturbations extérieures comme le vent. C'est pourquoi nous avons dû changer le PD en position en un PID, afin que l'on puisse avoir une performance de référence pour les ADRC. Le choix des constructeurs d'un PD plutôt qu'un PID en position s'explique facilement car il permet de faire un suivi de trajectoire. Un PID dans ces conditions, saturerait très rapidement à cause du terme intégral qui diverge si le drone est loin de sa position désirée, et s'écroulerait donc. L'ADRC se basant sur un PD en position comme nous l'avons conçu nous permet donc de palier la robustesse au vent mais aussi une éventuelle instabilité qui peut être présente en suivi de trajectoire.

Les vols en laboratoire ont validé les résultats obtenus en simulation, bien que les difficultés d'implémentation du contrôleur ADRC sur le Rolling-Spider (en particulier le fait que nous n'ayons pas accès directement à la position du drone mais à son estimation) ont fait en

sorte que nous ne soyons pas exactement dans les mêmes conditions que les simulations. Il serait donc aussi intéressant d'obtenir des résultats de vol avec des drones ayant un retour direct de position, soit par exemple en utilisant le système du Vicon Tracker (possible en laboratoire mais difficile, voire impossible à mettre en œuvre dans des conditions d'utilisation commerciale), ou encore, comme il se fait beaucoup dans les drones utilisés aujourd'hui, un système GPS (*Global Positioning System*), qui nous donnerait lui aussi une estimation de la position du drone, mais avec une meilleure précision, et une dérive moindre voire nulle. L'implantation de l'ADRC sur un drone commercial (comme nous l'avons fait pour le Rolling-Spider) reste donc très dépendante du hardware utilisé, et nécessite donc généralement une adaptation au système déjà mis en place dans le contrôleur. Cette adaptation indispensable se fait assez facilement et n'est qu'un petit prix à payer pour une nette amélioration des performances de vol des drones.

BIBLIOGRAPHIE

- A. N. Atassi et H. K. Khalil, “A separation principle for the stabilization of a class of nonlinear systems”, *IEEE Transactions on Automatic Control*, vol. 44, no. 9, pp. 1672–1687, 1999.
- J. Bernat et S. Stepien, “Multi-modelling as new estimation schema for high-gain observers”, *International Journal of Control*, vol. 88, no. 6, pp. 1209–1222, 2015.
- G. Bornard et H. Hammouri, “A high gain observer for a class of uniformly observable systems”, dans *Decision and Control, 1991., Proceedings of the 30th IEEE Conference on.* IEEE, 1991, pp. 1494–1496.
- W. Chenlu, C. Zengqiang, S. Qinglin, et Z. Qing, “Design of pid and adrc based quadrotor helicopter control system”, dans *Control and Decision Conference (CCDC), 2016 Chinese.* IEEE, 2016, pp. 5860–5865.
- J. Cioffi et T. Kailath, “Fast, recursive-least-squares transversal filters for adaptive filtering”, *IEEE Transactions on Acoustics, Speech, and Signal Processing*, vol. 32, no. 2, pp. 304–337, 1984.
- W. Dong, G.-Y. Gu, X. Zhu, et H. Ding, “A high-performance flight control approach for quadrotors using a modified active disturbance rejection technique”, *Robotics and Autonomous Systems*, 2016.
- J. C. Doyle, K. Glover, P. P. Khargonekar, et B. A. Francis, “State-space solutions to standard h_2 and h_∞ control problems”, *IEEE Transactions on Automatic control*, vol. 34, no. 8, pp. 831–847, 1989.
- H. L. Dryden et A. M. Kuethe, “Effect of turbulence in wind-tunnel measurements”, 1931.
- C. Edwards et S. Spurgeon, *Sliding mode control : theory and applications.* Crc Press, 1998.
- F. Esfandiari et H. K. Khalil, “Output feedback stabilization of fully linearizable systems”, *International Journal of control*, vol. 56, no. 5, pp. 1007–1037, 1992.
- Z. Gao, “Active disturbance rejection control : a paradigm shift in feedback control system design”, dans *2006 American control conference.* IEEE, 2006, pp. 7–pp.

- , “Scaling and bandwidth-parameterization based controller tuning”, dans *Proceedings of the American control conference*, vol. 6, 2006, pp. 4989–4996.
- Z. Gao, S. Hu, et F. Jiang, “A novel motion control design approach based on active disturbance rejection”, dans *Decision and Control, 2001. Proceedings of the 40th IEEE Conference on*, vol. 5. IEEE, 2001, pp. 4877–4882.
- Z. Gao, Y. Huang, et J. Han, “An alternative paradigm for control system design”, dans *Decision and Control, 2001. Proceedings of the 40th IEEE Conference on*, vol. 5. IEEE, 2001, pp. 4578–4585.
- B.-Z. Guo et Z.-L. Zhao, “The active disturbance rejection control”, *Active Disturbance Rejection Control for Nonlinear Systems : An Introduction*, pp. 53–92, 2016.
- B.-Z. Guo et Z.-l. Zhao, “On the convergence of an extended state observer for nonlinear systems with uncertainty”, *Systems & Control Letters*, vol. 60, no. 6, pp. 420–430, 2011.
- B.-Z. Guo et Z.-L. Zhao, “On convergence of the nonlinear active disturbance rejection control for mimo systems”, *SIAM Journal on Control and Optimization*, vol. 51, no. 2, pp. 1727–1757, 2013.
- Z. Z. L. Guo B Z, “On convergence of the nonlinear active disturbance rejection control for MIMO systems”, *SIAM Journal on Control and Optimization*, vol. Vol. 51, no. 2, pp. 1727–1757, 2013.
- J. Han, “Control theory : Model approach or control approach”, *Syst. Sci. Math*, vol. Vol. 9, no. 4, pp. 328–335, 1989.
- , “Nonlinear PID controller”, *J. Autom.*, vol. Vol. 20, no. 4, pp. 487–490, 1995.
- J. Han et W. Wang, “Nonlinear tracking-differentiator”, *Syst. Sci. Math*, vol. Vol. 14, no. 2, pp. 177–183, 1994.
- J. Han, “From pid to active disturbance rejection control”, *IEEE transactions on Industrial Electronics*, vol. 56, no. 3, pp. 900–906, 2009.
- , “Linearity and nonlinearity in feedback systems”, *Control and Decision*, vol. 3, no. 2, pp. 27–32, 1988.
- , “A class of extended state observers for uncertain systems”, *Control and Decision*, vol. 10, no. 1, pp. 85–88, 1995.

- , “The discrete form of the tracking differentiator”, *Systems Science and Mathematical Sciences*, vol. 19, pp. 268–273, 1999.
- , “Nonlinear design methods for control systems”, dans *Proc. of the 14th IFAC World Congress*, 1999, pp. 521–526.
- , “Active disturbance rejection control technique-the technique for estimating and compensating the uncertainties”, *National Defense Industry Press Beijing, China (in Chinese)*, 2008.
- , “A new type of controller : Nlpid [j]”, *Control and Decision*, vol. 6, 1994.
- , “Nonlinear state error feedback control law - nlsef [j]”, *Control and Decision*, vol. 3, 1995.
- Y. Huang et J. Han, “Analysis and design for the second order nonlinear continuous extended states observer”, *Chinese science bulletin*, vol. 45, no. 21, pp. 1938–1944, 2000.
- Y. Huang, K. Xu, J. Han, et J. Lam, “Flight control design using extended state observer and non-smooth feedback”, dans *Decision and Control, 2001. Proceedings of the 40th IEEE Conference on*, vol. 1. IEEE, 2001, pp. 223–228.
- H. K. Khalil, “High-gain observers in nonlinear feedback control”, dans *Control, Automation and Systems, 2008. ICCAS 2008. International Conference on*. IEEE, 2008, pp. xlvii–lvii.
- W. Khan et M. Nahon, “A propeller model for general forward flight conditions”, *International Journal of Intelligent Unmanned Systems*, vol. 3, no. 2/3, pp. 72–92, 2015.
- J. Li, R. Li, et H. Zheng, “Quadrotor modeling and control based on linear active disturbance rejection control”, dans *Control Conference (CCC), 2016 35th Chinese*. IEEE, 2016, pp. 10 651–10 656.
- C. Liu, Y. Chu, L. Wang, et Y. Zhang, “Application and the parameter tuning of adrc based on bfo-pso algorithm”, dans *Control and Decision Conference (CCDC), 2013 25th Chinese*. IEEE, 2013, pp. 3099–3102.
- D. Ma, Y. Xia, T. Li, et K. Chang, “Active disturbance rejection and predictive control strategy for a quadrotor helicopter”, *IET Control Theory & Applications*, vol. 10, no. 17, pp. 2213–2222, 2016.

- A. M. Nagy, G. Mourot, B. Marx, J. Ragot, et G. Schutz, “Systematic multimodeling methodology applied to an activated sludge reactor model”, *Industrial & Engineering Chemistry Research*, vol. 49, no. 6, pp. 2790–2799, 2010.
- G. P. Noone et D. C. Noone, “The debate over autonomous weapons systems”, *Case W. Res. J. Int’l L.*, vol. 47, p. 25, 2015.
- D. D. V. Pandit et A. Poojari, “A study on amazon prime air for feasibility and profitability—a graphical data analysis”, *IOSR Journal of Business and Management*, vol. 16, no. 11, pp. 06–11, 2014.
- B. Sun et Z. Gao, “A dsp-based active disturbance rejection control design for a 1-kw h-bridge dc-dc power converter”, *IEEE Transactions on Industrial Electronics*, vol. 52, no. 5, pp. 1271–1277, 2005.
- B. Theys, G. Dimitriadis, T. Andrianne, P. Hendrick, et J. De Schutter, “Wind tunnel testing of a vtol mav propeller in tilted operating mode”, dans *Unmanned Aircraft Systems (ICUAS), 2014 International Conference on*. IEEE, 2014, pp. 1064–1072.
- N. K. Tran, E. Bulka, et M. Nahon, “Quadrotor control in a wind field”, dans *Unmanned Aircraft Systems (ICUAS), 2015 International Conference on*. IEEE, 2015, pp. 320–328.
- G.-Y. Xia et Z.-H. Liu, “Quadrotor unmanned helicopter attitude control based on improved adrc”, dans *Guidance, Navigation and Control Conference (CGNCC), 2014 IEEE Chinese*. IEEE, 2014, pp. 916–921.
- Y. Xia, P. Shi, D. Rees, J. Han *et al.*, “Active disturbance rejection control for uncertain multivariable systems with time-delay”, *IET Control Theory & Applications*, vol. 1, no. 1, pp. 75–81, 2007.
- W. Xue et Y. Huang, “The active disturbance rejection control for a class of mimo block lower-triangular system”, dans *Control Conference (CCC), 2011 30th Chinese*. IEEE, 2011, pp. 6362–6367.
- W. X. Y. Huang et C. Zhao, “Active Disturbance Rejection Control Methodology And Theoretical Analysis”, *Journal of System Science and Mathematical Science*, vol. Vol. 31, no. 9, pp. 111–1129, 2011.
- P. C. Young et J. Willems, “An approach to the linear multivariable servomechanism problem”, *International journal of control*, vol. 15, no. 5, pp. 961–979, 1972.