

UNIVERSITÉ DE MONTRÉAL

COMMANDE ADAPTATIVE D'UN HÉLICOPTÈRE TANDEM SANS PILOTE À MOTEUR
QUATRE TEMPS

ADRIEL MUNGER
DÉPARTEMENT DE GÉNIE ÉLECTRIQUE
ÉCOLE POLYTECHNIQUE DE MONTRÉAL

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

UNIVERSITÉ DE MONTRÉAL

ÉCOLE POLYTECHNIQUE DE MONTRÉAL

Ce mémoire intitulé :

COMMANDE ADAPTATIVE D'UN HÉLICOPTÈRE TANDEM SANS PILOTE À MOTEUR
QUATRE TEMPS

présenté par : MUNGER Adriel

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

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

M. GOURDEAU Richard, Ph. D., président

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

M. SOFIANE Achiche, Ph. D., membre

DÉDICACE

*Aux personnes importantes dans ma vie,
Ainsi qu'à celle qui l'a été,
Merci.*

REMERCIEMENTS

J'aimerais tout d'abord remercier mon directeur de maîtrise, David Saussié, pour m'avoir donné l'opportunité d'entreprendre une maîtrise dans son domaine d'expertise, et pour son temps qu'il a consacré à m'aider dans cette tâche.

Je tiens à remercier la compagnie Laflamme Aéro Inc., et spécialement Enrick Laflamme, pour m'avoir permis de travailler sur le projet audacieux du LX300 et de m'avoir fait connaître la région de Thetford.

J'aimerais aussi remercier les partenaires industriels du projet CARIC, tel que NGC Aérospatiale Ltée et RAAS Inc., que j'ai pu côtoyer durant les essais avec le drone et qui m'ont été d'un précieux soutien lors de mes travaux.

Mes remerciements vont aussi à l'organisme MITACS, qui a grandement aidé à subventionner mes travaux de recherche et sans qui ce projet n'aurait pas été possible.

Je voudrais aussi remercier Frédéric Laliberté, qui a pu m'aider de nombreuses fois sur les travaux du drone et avec qui il fait toujours bon discuter.

Enfin je remercie mes parents et amis, qui ont été là pour moi et qui m'ont supporté tout au long de ma maîtrise. Vous m'avez aidé à maintenir l'enthousiasme qui m'a convaincu d'entreprendre ce projet.

RÉSUMÉ

Le développement d'applications utilisant des drones a grandement augmenté dans les dernières années. Ces drones, souvent des multicoptères, permettent par exemple de filmer des scènes avec des points de vue difficilement atteignables par d'autres moyens. Par contre, le transport d'équipements plus lourds demeure une tâche plus ardue et demande l'utilisation de drones plus puissants. Dans cette optique, le projet de recherche et de développement CARIC AUT703_TRL4+ répond à ce problème par la réalisation d'un hélicoptère tandem sans pilote. Cette configuration de rotors permet de transporter plus facilement des charges ayant différents gabarits pour différents centrages. Dans le cadre de ce projet CARIC, portant sur le drone LX300 de Laflamme Aéro Inc, le présent mémoire s'intéresse à la synthèse de différentes lois de commande.

La première partie de ce projet porte sur l'asservissement du moteur afin de maintenir la vitesse des rotors constante durant le vol. Une approximation de la dynamique du moteur par moindres carrés a été obtenue à partir de données expérimentales. Basé sur cette estimation, une loi de commande proportionnelle intégrale (PI) a été retenue et implémentée sur la plateforme embarquée du drone. Une attention a été portée sur la flexibilité et la robustesse de la loi de commande, entre autres par l'ajout d'un mode "manuel". Après la validation du contrôleur, un moyen de déterminer la force fournie par le moteur est développée au moyen d'un modèle dynamique. Cette estimation se base sur les forces de traînée théoriques provenant des rotors.

La seconde partie porte sur les lois de commande robustes du drone. Lors de précédents travaux, les lois de stabilisation en vol du drone LX300 ont été obtenues via une synthèse H_∞ structurée. Mais, suite à des limitations de l'autopilote retenu par la compagnie, il s'est avéré nécessaire d'apporter des modifications à l'architecture de la loi de commande, notamment par le positionnement de l'action proportionnelle des boucles PID sur le signal d'erreur. Une nouvelle synthèse H_∞ structurée est alors effectuée en considérant des gabarits fréquentiels supplémentaires sur le couplage entre états. Cette nouvelle loi de commande est ensuite évaluée sur un modèle non linéaire du drone et comparée avec la précédente architecture. Ces résultats démontrent les limitations de l'autopilote utilisé puisque l'on constate une diminution des performances possibles avec la nouvelle architecture.

Finalement, pour déterminer les changements dans la dynamique en vol du drone, un loi de commande adaptative est développée. Cette loi de commande se base sur le contrôleur robuste précédemment trouvé et utilise une approche indirecte. La dynamique du drone est donc estimée en vol via un algorithme de moindres carrés récursifs stabilisés, et basé sur cette estimation les gains du contrôleur sont mis à jour. Afin d'obtenir les performances en vol désirées, les gains sont modifiés via un placement de pôles adaptatif sur le modèle linéarisé du drone en boucle fermée. La stabilité de

l'estimateur est validée sur le modèle non linéaire du drone, puis les résultats de la loi de commande sont présentés et analysés.

ABSTRACT

The development in new drone applications has increased in the recent years. Often in a multicopter configuration, they allow for example to easily film viewpoints that would have been hard to reach otherwise. But, when it comes to carrying heavier equipments, a multicopter is often not adequate and a more powerful drone is needed. In this context, the CARIC AUT703_TRL4+ research and development project addresses this problem with the realization of a tandem helicopter drone. With this rotor configuration, the transportation of loads with different shapes and centerings is more easily achieved. Within this CARIC project, namely the LX300 drone of Laflamme Aéro Inc., work on different control laws has been done.

The first part of this project focuses on the engine control to maintain constant rotor speed during flight. The engine dynamics are obtained by using a least squares approximation from experimental data. Based on this estimate, a proportional-integral control (PI) law was selected and implemented on an embedded platform. Attention was paid to the flexibility and robustness of the control law used, among others by including the addition of a "manual" mode. After the controller has been validated, a means of determining the force supplied by the motor is approached by the development of a dynamic model. This estimate is based on the theoretical drag forces coming from the rotors.

The second part deals with the robust control laws of the drone. Previous results on the flight stabilization laws of the LX300 UAV have been obtained with a structured H_∞ synthesis. However, following the limitations of the autopilot chosen by the company, it has been proven necessary to make changes to the architecture of the control law, in particular by positioning the proportional action of the PID loops on the error signal. A new structured H_∞ synthesis is performed by considering additional frequency filters on interstate coupling. This new control law is then evaluated on a non-linear model of the drone and compared with the previous architecture. These results demonstrate the limitations of the autopilot used since there is a decrease in the available performance with the new architecture.

Finally, to determine possible changes in the flight dynamics of the drone, an adaptive control law is developed. This control law is based on the previously found robust controller and uses an indirect approach. The dynamics of the drone is therefore estimated inline using a stabilized recursive least-squares algorithm (MCRS), and based on this estimate the controller's gains are updated. In order to obtain the desired flight performance, the gains are modified via an adaptive pole placement on the linearized model of the closed loop drone. The stability of the estimator is validated on the non-linear model of the drone, then the results of the control law are presented and analyzed.

TABLE DES MATIÈRES

DÉDICACE	iii
REMERCIEMENTS	iv
RÉSUMÉ	v
ABSTRACT	vii
TABLE DES MATIÈRES	viii
LISTE DES TABLEAUX	x
LISTE DES FIGURES	xi
LISTE DES SIGLES ET ABRÉVIATIONS	xiii
LISTE DES ANNEXES	xiv
CHAPITRE 1 INTRODUCTION	1
1.1 Stabilisation du moteur	2
1.2 Stabilisation du drone	2
1.3 Plan du mémoire	4
CHAPITRE 2 REVUE DE LITTÉRATURE	5
2.1 Commande adaptative	5
2.2 Commande pour un tandem-rotor	8
CHAPITRE 3 CONTRÔLE EMBARQUÉ DU MOTEUR	10
3.1 Problématique	10
3.2 Modélisation du moteur Rotax	11
3.3 Limitations sur le contrôleur	17
3.4 Interface avec le monde réel	19
3.4.1 Encodage du throttle	20
3.4.2 Commande manuelle du pilote et transition vers contrôleur	20
3.5 Méthodologie de test	21
3.6 Résultats sur banc d'essai	22

3.7	Modélisation non linéaire du moteur	24
3.7.1	Débit d'air à l'entrée	25
3.7.2	Pression au collecteur d'admission	25
3.7.3	Génération du couple en sortie	26
3.7.4	Vitesse en sortie	28
3.7.5	Comparaison avec le modèle linéaire	28
CHAPITRE 4 COMMANDE ROBUSTE DU DRONE		30
4.1	Modélisation du drone	30
4.2	Synthèse H_∞ structurée initiale	39
4.3	Motivations d'une nouvelle synthèse	42
4.4	Résultats de la nouvelle architecture	43
CHAPITRE 5 COMMANDE ADAPTATIVE DU DRONE		50
5.1	Approche directe	50
5.2	Approche indirecte	52
5.2.1	Estimation du système	54
5.2.2	Limitations sur la matrice de covariance	58
5.2.3	Facteur d'oubli variable	60
5.3	Résultats d'estimation	62
5.4	Calcul du nouveau gain	69
5.4.1	Utilisation de la pseudoinverse de Moore-Penrose	70
5.4.2	Implémentation et résultats	73
CHAPITRE 6 CONCLUSION		80
6.1	Considérations sur les limitations matérielles	80
6.2	Implémentation des commandes robustes et adaptatives	80
6.3	Considérations sur le modèle utilisé	81
6.4	Implication des simplifications effectuées	81
6.5	Avenues possibles de la commande adaptative	82
RÉFÉRENCES		83
ANNEXES		92

LISTE DES TABLEAUX

Tableau 4.2	Requis de synthèse H_∞	41
Tableau 5.1	Position des pôles après adaptation, avec amélioration pour chaque pôle . .	77
Tableau 5.2	Position des pôles après adaptation, avec amélioration pour chaque pôle . .	78

LISTE DES FIGURES

Figure 1.1	Modèle du drone LX300 de Laflamme Aéro Inc. ¹	1
Figure 3.1	Schéma <i>haut-niveau</i> du contrôle moteur ²	11
Figure 3.2	Exemple de données utilisées pour l’approximation linéaire du moteur . . .	12
Figure 3.3	Mesure de la vitesse du moteur (mauve) et son approximation d’ordre 2 (bleue)	15
Figure 3.4	Schéma du contrôleur utilisé pour le moteur	15
Figure 3.5	Synthèse temporelle avec le contrôleur rebouclé sur le système	17
Figure 3.6	Schéma <i>Simulink</i> de la chaîne d’action du contrôleur avec ses limites . . .	17
Figure 3.7	Synthèse temporelle avec le contrôleur rebouclé sur le système	18
Figure 3.8	Plateforme embarquée pour le contrôle moteur	19
Figure 3.9	Encodage PWM du servo vers throttle	20
Figure 3.10	Circuit reliant le <i>manuel</i> et le <i>cruise</i>	21
Figure 3.11	Banc de test 4DOF du drone, © 2018 LAFLAMME AERO INC.	22
Figure 3.12	Premier test avec contrôleur de moteur	23
Figure 3.13	Second test avec contrôleur de moteur : rejet des perturbations	24
Figure 3.14	Modélisation nonlinéaire du moteur	25
Figure 3.15	Comparaison entre le modèle linéaire et celui non linéaire	29
Figure 4.1	Repère du drone, © 2018 LAFLAMME AERO INC.	30
Figure 4.2	Représentation des dynamiques du drone	33
Figure 4.3	Actionneurs du plateau cyclique d’un rotor du LX300, © 2018 LAFLAMME AERO INC. .	33
Figure 4.4	Boucle de stabilisation de l’attitude du drone	36
Figure 4.5	Architecture du contrôleur initial en roulis, utilisant la sortie	37
Figure 4.6	Problème de synthèse H_∞	40
Figure 4.7	Architecture du contrôleur en roulis utilisant l’erreur	42
Figure 4.8	Gains séquencés trouvés par H_∞ avec erreur en entrée	44
Figure 4.9	W_1 , Réponse fréquentielle d’un suivi de trajectoires	45
Figure 4.10	W_2 , Marge de stabilité du système	45
Figure 4.11	W_4 , Gain maximal sur commande latérale	46
Figure 4.12	W_5 , Gain maximal sur les commandes longitudinales et latérales	46
Figure 4.13	W_3 , Transitoires sur ψ pour un échelon de perturbation	47
Figure 4.14	Réponses temporelles avec utilisation de l’erreur en entrée	47
Figure 4.15	Réponse à l’échelon pour différentes vitesses et charges, avec retour de sortie	48
Figure 5.1	Architecture du contrôleur avec adaptation directe	52

Figure 5.2	Architecture du contrôleur avec adaptation indirecte	53
Figure 5.3	Estimation du modèle linéaire sans restriction sur \mathbf{P} , avec $\lambda = 0.99$	63
Figure 5.4	Estimation du modèle linéaire avec réinitialisation de la covariance et $\lambda = 0.99$	64
Figure 5.5	Estimation du modèle linéaire avec limite sur P^{-1} et $\lambda = 0.99$	65
Figure 5.6	Estimation du modèle linéaire avec limite sur P^{-1} et facteur d'oubli variable	66
Figure 5.7	Effet du paramètre Σ_0 sur la stabilité de l'estimation du sys. linéaire	67
Figure 5.8	Estimation des états du modèle non-linéaire avec restriction sur \mathbf{P} et FF variable	68
Figure 5.9	Réponse temporelle des angles ϕ, θ, ψ après l'adaptation du gain K_x	75
Figure 5.10	Réponse temporelle des vitesses angulaires p, q, r après l'adaptation du gain K_x	76
Figure 5.11	Repositionnement des pôles vers le modèle de référence en utilisant \mathbf{K}'_{ξ} . .	77
Figure 5.12	Repositionnement des pôles vers le modèle de référence en utilisant \mathbf{K}'_{\sim} . .	78

LISTE DES SIGLES ET ABRÉVIATIONS

CARIC	Consortium en aérospatiale pour la recherche et l'innovation au Canada
LTI	<i>Linear time-invariant</i> - Système linéaire invariant dans le temps
MCRS	Moindres Carrés Récurifs Stabilisés
MIMO	<i>Multiple Inputs Multiple Outputs</i> - Système à plusieurs entrées (et sorties)
MRAC	<i>Model Reference Adaptive Control</i> - Commande adaptative par un modèle de référence
PE	Persistence d'excitation
PI	Proportionnel-Intégral
PID	Proportionnel-Intégral-Dérivée
RLS	<i>Recursive Least Squares</i> - Méthode des moindres carrés récurifs
RPM	Révolutions par minutes
SISO	<i>Single Input Single Output</i> - Système à une entrée et une sortie
UAV	<i>Unmanned Aerial Vehicle</i> - Véhicule aérien sans pilote

LISTE DES ANNEXES

Annexe A	Convergence de l'estimateur RLS avec facteur d'oubli	92
Annexe B	Moindres carrés récurrents stabilisés	94

CHAPITRE 1 INTRODUCTION

La popularité des drones à vocation civile a grandement augmenté dans les dernières années, de par leur coût très abordable¹ et de l'évolution de l'électronique disponible sur le marché (de récents avancements pourraient même permettre à des drones d'effectuer des tâches de livraison²). La principale caractéristique d'un drone est bien entendu son absence de pilote, d'où son appellation "UAV" (*Unmanned Aerial Vehicle*, ou véhicule aérien sans pilote). Ce sont en général des multicoptères contrôlés à distance via une communication radio. Par exemple, les plus communs sont à l'heure actuelle des multiroteurs de type quadricoptère.

Pour les petits drones, leur faible inertie leur permet d'effectuer des manoeuvres acrobatiques très rapides, alors que les plus gros peuvent demeurer plus stables en vol et soulever une charge plus élevée. Lorsque la charge à transporter devient considérable, certaines configurations de rotors deviennent plus avantageuses (au détriment de la rapidité de leurs mouvements). L'une de ces topologies de rotors est en position tandem, soit deux rotors parallèles pointant vers le haut, et tournant en sens inverse. Un exemple connu d'hélicoptère de cette forme est certainement le CH-47 Chinook³ de Boeing, utilisé depuis 1962. Dans un projet d'innovation avec le CARIC, la compagnie Laflamme Aéro Inc. a développé un appareil similaire mais sans pilote, soit le LX300.



Figure 1.1 Modèle du drone LX300 de Laflamme Aéro Inc.⁴

Le LX300, présenté à la figure 1.1, est un drone tandem-rotor de 300kg pouvant transporter des

1. À ce jour, le prix peut être aussi bas que 30\$CAN

2. voir <https://www.amazon.com/Amazon-Prime-Air/b?ie=UTF8&node=8037720011>

3. voir <http://www.boeing.com/defense/ch-47-chinook/> pour un exemple du CH-47

4. Image tirée de XPONENTIAL 2017

charges allant théoriquement jusqu'à 90kg. Pour permettre une force de levage aussi grande, l'appareil est doté d'un moteur quatre temps de 100 HP de ROTAX, soit le moteur que l'on trouverait normalement dans un avion léger. À la différence d'un avion, la configuration des rotors permet au drone d'effectuer un vol stationnaire, ce qui facilite des manœuvres de précision. Outre sa charge utile, l'appareil peut aussi être doté de différents systèmes d'acquisition et d'une navigation automatisée.

Le projet CARIC concernant ce drone (AUT-703_TRL4+) fait intervenir plusieurs compagnies telles que Laflamme Aéro Inc, NGC Aérospatiale Ltée, RAAS Inc. et SINTERS AMERICA Inc., en plus de Polytechnique Montréal et l'École de Technologie Supérieure. Dans le cadre de cette maîtrise, une bourse MITACS a été obtenue pour mener des travaux de recherche en milieu industriel avec la compagnie Laflamme Aéro Inc. L'objectif visait notamment à assister la compagnie sur le travail nécessaire en contrôle pour permettre la tenue de tests en vol.

1.1 Stabilisation du moteur

Lorsque le moteur retenu est utilisé dans un avion, le contrôle de sa poussée se fait normalement par le pilote au moyen d'un levier relié à l'entrée d'air du moteur. Dans le cas du drone de Laflamme, l'absence de pilote nécessite que le contrôle se fasse à distance ou automatiquement. Un moyen possible est d'utiliser un système radio, où celui-ci communique avec un servomoteur pour actionner la manette des gaz. Cette méthode est néanmoins contrainte par la précision du pilote, et peut devenir contraignante lors de manœuvres délicates. Une solution simple est d'asservir le contrôle moteur au moyen d'un système embarqué, de sorte que le moteur se maintienne automatiquement à la même vitesse sans interaction nécessaire du pilote. Puisque ce système n'est pas déjà présent dans l'appareil, le contrôle moteur passe aussi par sa conception et son installation dans le drone.

Lors de l'élaboration du contrôleur, un moyen de tester tous ses modes d'opération sans risquer de l'endommager est de le simuler au moyen d'un simulateur temps réel. De cette façon, son temps de réponse dans des cas critiques peut être validé sécuritairement et plus rapidement que par tâtonnements. Une modélisation non linéaire du système permettra aussi de connaître la plage de performance du moteur asservi lors de la manipulation de différentes charges.

1.2 Stabilisation du drone

Un contrôle d'aéronef peut se faire sur deux niveaux, c'est-à-dire via une boucle de stabilisation interne ainsi qu'une boucle de stabilisation externe. La première vise à ce que l'appareil demeure stable en vol stationnaire ou pour différentes vitesses de croisière, alors que la seconde vise à ce que l'engin suive une trajectoire précise (ou encore se positionne à un point donné). Pour le drone de

Laflamme, un autopilote commercial a été utilisé pour contrôler la première boucle. Ce contrôleur, soit "Micropilot", permet d'appliquer des lois de commande sur les angles de roulis, tangage et lacet ainsi que sur sa vitesse verticale. Des travaux précédemment menés avec NGC Aérospatiale Ltée [42] en lien avec cet autopilote se sont conclus par un contrôleur à gains séquencés en vitesse. Bien que l'autopilote soit utilisé dans plusieurs projets d'aéronefs à travers le monde⁵, il se trouve que sa plateforme apportait beaucoup de limitations sur l'action du contrôleur conçu. La principale limitation venait d'une différence entre l'architecture des lois de contrôle implémentées, ce qui rajoutait plusieurs étapes de calcul supplémentaires pour se conformer au modèle théorique trouvé. Une solution envisagée était donc de mettre à jour la loi de commande synthétisée pour qu'elle soit davantage compatible avec l'autopilote utilisé.

De manière similaire aux précédentes lois de commande, le nouveau contrôleur en question se base sur la théorie de la commande robuste et est synthétisé avec un modèle dynamique du drone. Cette approche permet de garantir une stabilité même en présence de bruit de mesure ou de perturbations sur le système.

D'une autre part, les lois de commande robuste utilisés limitent les capacités du drone. Pour les différentes charges utiles transportées en vol, ce contrôleur doit toujours s'assurer de stabiliser le pire cas possible, ce qui se rapporte à dégrader ses performances possibles pour d'autres situations. Lors d'un vol, l'autopilote est capable de changer ses gains en fonction de ses vitesses de croisière, puisqu'il est capable de la mesurer. Par contre, s'il ne connaît pas la masse transportée, il doit assumer le cas extrême pour demeurer stable. Dans l'optique d'améliorer la tenue en vol du drone, l'utilisation d'une commande adaptative permet de régler ce problème. Ainsi, ce type de commande est en mesure de déterminer les dynamiques du système et de modifier le contrôleur en conséquence. Si la masse transportée varie (ou se déplace) durant le vol, la commande adaptative offre la flexibilité de garder le contrôleur robuste trouvé précédemment, tout en changeant son effet en vol pour atteindre les performances désirées.

La commande adaptative est un domaine de recherche vaste, où il existe des méthodes variées pour synthétiser une loi de commande pour une même application (voir Tao [84], Ioannou et Sun [34]). Le but de cette recherche est en premier lieu de cerner parmi celles disponibles la commande la plus adéquate à être appliquée au drone. Par exemple, certaines se prêtent mieux à l'implémentation sur une plateforme embarquée que d'autres, en ce qui concerne la puissance de calcul requise pour les effectuer. En second lieu, il est voulu que le contrôleur trouvé, une fois simulé, permette de déterminer l'amélioration possible en terme de performance et de stabilité du nouveau système.

5. Voir <https://www.micropilot.com/index.htm>

1.3 Plan du mémoire

Les chapitres de ce mémoire représentent les différents types de commandes utilisés pour le drone LX300. Avant d'entrer dans les détails de chacune d'elles, le chapitre 2 présente une revue de littérature succincte sur la commande adaptative et sur le contrôle moteur. L'ordre des chapitres suivra ensuite les étapes successives de conception des contrôleurs.

La première commande abordée est celle du moteur dans le chapitre 3. L'estimation de la dynamique du moteur, sa modélisation et son contrôle y sont décrites. Un modèle non linéaire équivalent, basé sur une approximation de la dynamique des rotors, est ensuite développé et comparé avec les résultats du premier modèle.

Le chapitre 4 traite ensuite de la commande robuste permettant de stabiliser le drone en vol. Le modèle utilisé fait l'hypothèse que le contrôle moteur est présent dans le drone et donc que la vitesse des rotors est maintenue constante. Les différents requis de performances y sont présentés, et un comparatif est fait entre l'ancienne architecture et la nouvelle pour l'autopilote.

En utilisant le résultat du chapitre précédent, le chapitre 5 utilise le contrôleur robuste comme point de départ pour stabiliser le drone avec une loi de commande adaptative. Les algorithmes d'estimation des paramètres inconnus du système sont développés, puis l'implémentation utilisée et les résultats sont présentés.

CHAPITRE 2 REVUE DE LITTÉRATURE

Cette section passe en revue la commande adaptative traitée dans la littérature et en lien avec ce mémoire. La première section effectue un survol des méthodes de contrôle adaptatives, puis la seconde section traite de l'application de ces méthodes pour un drone tandem-rotor. En ce qui concerne le contrôle linéaire d'un moteur, le lecteur peut se référer à l'ouvrage de Moudgalya [53] pour le contrôle de systèmes linéaires discrets.

2.1 Commande adaptative

Le terme *adaptatif* réfère à un contrôleur capable de modifier son comportement en fonction de changements dans les dynamiques du système auquel il est attaché [87]. Pour un système linéaire standard :

$$\begin{aligned}\dot{\mathbf{x}} &= \mathbf{Ax} + \mathbf{Bu} \\ \mathbf{y} &= \mathbf{Cx} + \mathbf{Du}\end{aligned}$$

le contrôleur habituellement implémenté utilise une matrice de gains \mathbf{K} , donnant la commande $\mathbf{u} = \mathbf{Hr} - \mathbf{Ky}$, soit la différence entre un signal de référence \mathbf{r} pondéré par \mathbf{H} et la mesure en sortie \mathbf{y} employée. En modifiant la commande pour que le gain du contrôleur soit maintenant fonction de différents paramètres du système ($\mathbf{K} = f(\alpha_1, \alpha_2, \dots)$), le contrôleur utilisé est maintenant capable de s'adapter pour que le temps de réponse et les transitoires d'une commande changent tel que désiré. La commande adaptative est habituellement implémentée en fonction d'un modèle de référence (*Model Reference Adaptive Control*), de sorte que les dynamiques de l'état \mathbf{x} tendent vers celles d'un modèle aux meilleures performances, connu de l'utilisateur.

Historiquement, la commande adaptative date des années 50, où les premières preuves de concepts pour ces contrôleurs furent publiés [7]. Les premiers algorithmes de contrôle utilisèrent une première méthode dite de *sensibilité*, basée sur une descente du gradient de l'erreur. Il s'agit d'une fonction de coût que l'algorithme minimise en variant la commande basé sur sa dérivée. Une méthode pour calculer ces fonctions de sensibilité est donnée par la règle dite *MIT* (*MIT rule*, [48]), ou par des algorithmes similaires (voir Margolis et Leondes [49] et Stromer [81]). Pour des systèmes stables et où les signaux d'intérêts peuvent facilement être générés (ex. l'état mesurable), c'est une méthode simple pouvant fonctionner. Par contre, lorsque la fonction de sensibilité doit être approximée numériquement, la méthode peut poser de sérieux problèmes pour un système instable en boucle ouverte (ex. un drone) et où une quantité non négligeable de bruit de mesure peut perturber la loi

de commande (voir [34]). Il est d'ailleurs dit que dans les débuts du contrôle adaptatif, le contrôle d'aéronefs était "caractérisé par beaucoup d'enthousiasme, des systèmes peu performants et une absence de théorie" (traduit de Åström [87]), en lien avec un tragique incident survenu lors d'essais en vol avec un contrôleur adaptatif [25, 59]. Successivement, la synthèse de nouveaux contrôleurs adaptatifs via une analyse de stabilité ont fait leur apparition. Cette analyse se base essentiellement sur la théorie de Lyapunov, où l'on s'assure que les signaux d'intérêts \mathbf{x} demeurent stable, tout comme la dynamique du contrôleur à synthétiser (Polycarpou et Ioannou [67] présentent la stabilité évaluée par une différente analyse). Le système 2.1 présente un exemple de ce type d'application où \mathbf{x} est l'état du système et $\boldsymbol{\theta}$ le "paramètre" à estimer.

$$\begin{aligned} V(\mathbf{x}, \boldsymbol{\theta}) &= \frac{1}{2} \mathbf{x}^\top \mathbf{P} \mathbf{x} + \frac{1}{2} \boldsymbol{\theta}^\top \boldsymbol{\Gamma}^{-1} \boldsymbol{\theta} > 0, \quad \mathbf{P}, \boldsymbol{\Gamma} > 0 \\ \dot{V}(\mathbf{x}, \boldsymbol{\theta}) &= \mathbf{x}^\top \mathbf{P} \dot{\mathbf{x}} + \boldsymbol{\theta}^\top \boldsymbol{\Gamma}^{-1} \dot{\boldsymbol{\theta}} \leq 0 \end{aligned} \quad (2.1)$$

On peut noter les travaux de Parks [65], Shackcloth [75] et Phillipson [66] à ce sujet.

Avec les différentes méthodes de contrôle adaptatives publiées par la suite, deux branches sont notables : la commande directe et la commande indirecte. La commande adaptative directe est l'équivalent du réglage automatisé que ferait manuellement un automaticien sur les gains *PID* d'une machine. Dans la commande directe, les gains sont les inconnus du système, et une analyse de stabilité est normalement faite sur la dynamique pour trouver la dynamique des gains à implémenter. Sur cette technique on cite notamment Monopoli [51], Narendra et al. [55] et Goodwin et al. [29]. La commande adaptative indirecte, quant à elle, se base sur la connaissance du système pour trouver le meilleur contrôleur. Ce type de commande effectue en premier lieu une étape d'identification des paramètres inconnus du système, puis met à jour la nouvelle loi de contrôle. L'étape d'identification peut par exemple se faire via une fonction de Lyapunov sur l'erreur des paramètres (Kreisselmeier [39]), un observateur d'état augmenté de l'erreur des paramètres (Marino [50]), ou par régression linéaire avec la méthode des moindres carrés :

$$\begin{aligned} \hat{\theta}_k &= \hat{\theta}_{k-1} + \mathbf{P}_k \mathbf{h}_k \epsilon_k \\ \mathbf{P}_k^{-1} &= \lambda \mathbf{P}_{k-1}^{-1} + \mathbf{h}_k \mathbf{h}_k^\top \end{aligned}$$

où le nouvel estimé θ_k est calculé basé sur la mise à jour d'une matrice de covariance \mathbf{P}_k (utilisant un facteur d'oubli λ), l'erreur du précédent estimé ϵ_k et le vecteur de signaux du système \mathbf{h}_k . D'autres implémentations similaires sont à noter, telle que celle de Clarke [17] ou Dasgupta et Huang [21]. Le contrôle adaptatif s'étend aussi aux systèmes non linéaires, où plusieurs lois de commandes non linéaires ont aussi vu le jour. Ces dernières se basent encore sur la théorie de Lyapunov pour caractériser le bassin de convergence d'une telle commande, que ce soit par backstepping (Krstic et

Kokotovic [40]), mode de glissement (Bartolini et al. [11]), etc. Avec la multitude de contrôleurs linéaires (Ioannou et Sun [34]) ou non linéaires présents dans la littérature, d'autres problèmes sur le *maintien* de la commande furent aussi abordés, tel que la *persistance d'excitation* (PE). De la même manière qu'un observateur, le contrôleur adaptatif converge vers sa valeur désirée en la présence de transitoires dans le système (voir Narendra et Annaswamy [58] et Anderson [5] pour une définition détaillée). Cet aspect de la commande adaptative est central dans le sens que les lois de contrôle synthétisées ainsi que les analyses de stabilité reposent toutes sur l'hypothèse que le système contient un signal d'entrée suffisamment riche (ou PE). L'inverse implique que le contrôleur adaptatif ne convergera pas et peut même déstabiliser le système si les conditions initiales de l'estimateur ne sont pas adéquates. Ce problème fut abordé de diverses façons dans la littérature par des algorithmes pouvant modifier leur vitesse d'adaptation et ainsi contourner le problème. On note par exemple l'utilisation d'un facteur d'oubli variable pour un estimateur par moindres carrés [3, 27, 63, 85] ou l'ajout d'une borne supérieure sur la matrice de covariance utilisée [14, 74]. D'autres méthodes proposées utilisent une méthode de projection pour restreindre la dynamique de l'estimateur dans un ensemble fermé de valeurs restreintes [4, 54].

Les techniques de commande adaptatives étant très variées et pouvant changer énormément d'un système à l'autre, l'aspect de la robustesse de tels systèmes a aussi été abordé dans plusieurs ouvrages. La définition de la robustesse d'un contrôleur adaptatif ici fait référence au système résultant lorsqu'il y a par exemple des perturbations ajoutées aux signaux mesurés (incertitude additive), ou encore lorsque les gains du contrôleur comportent une incertitude. La robustesse est aussi intimement liée à la notion de persistance d'excitation. On cite notamment sur l'étude de robustesse de tels systèmes les travaux de Ortega et Tang [61], Landau [43], Datta et Ioannou [23] et Ioannou et Sun [34].

En ce qui concerne les systèmes à plusieurs entrées (MIMO), les premières méthodes utilisées furent directement dérivées des systèmes à une seule entrée (SISO), soit le contrôle adaptatif dit *décentralisé*. Cette méthode émet l'hypothèse que les principales dynamiques d'un système peuvent être réduites aux fonctions de transfert présentes dans la diagonale de la matrice de transfert du système (entre ses entrées et ses sorties). Outre Ioannou, on trouve à ce sujet les travaux de Datta [22], Narendra et Oleng [56] et Kubalčík et al. [41]. Par contre, pour des systèmes avec beaucoup de couplage entre les états, cette méthode devient rapidement inadéquate. D'autres méthodes, telles que l'utilisation d'une *Left Interactor Matrix* (ou similairement une *Right Interactor Matrix*, représentant approximativement l'inverse de la matrice de transfert du modèle de référence, sous forme d'une matrice triangulaire) pour calculer algébriquement la matrice de transfert du système et les interactions entre états [76, 82, 83] a été utilisée. Cette méthode offre beaucoup de rigueur, mais demande une plus grande quantité de calculs en conséquence. D'autres algorithmes généralisent les analyses de stabilité utilisés précédemment, soit au sens de Lyapunov [2, 52], ou directement par le

calcul en parallèle de n estimateurs utilisant les moindres carrés [62] (n étant l'ordre du système). La revue de littérature sur les systèmes multivariables adaptatifs de Tao [84] donne d'ailleurs un survol détaillé de ces techniques.

Plus récemment, de nouveaux types de contrôleurs basés sur les réseaux de neurones ont aussi fait leur apparition, où les différents nœuds du réseau (inconnus) sont ajustés par rétropropagation du gradient de manière à faire corrélérer les états du système avec la sortie du réseau. Cette technique ne sera pas abordée dans le présent mémoire, mais le lecteur peut se référer aux articles de Spooner et Passino [79], Polycarpou et Ioannou [68] et Narendra et Parthasarathy [57] détaillant la stabilité et l'implémentation de tels systèmes.

2.2 Commande pour un tandem-rotor

Il existe beaucoup d'exemples dans la littérature de contrôle adaptatif pour des hélicoptères tandem. Par contre, les articles présents sont généralement basés sur de petits drones, où le contrôle s'apparente davantage à celui d'un multirotor standard¹. Un multicoptère, par exemple, se contrôle en variant les vitesses de ses rotors pour générer les forces et moments sur sa structure. Une relation linéaire peut donc directement être établie entre la force générée et la commande (en courant) sur les moteurs. Les modèles de drone à deux hélices souvent utilisés ont un mécanisme similaire, à la différence que les angles des rotors avec la structure sont aussi variables. L'allure générale du système s'apparente donc davantage à un pendule à deux hélices. En ajoutant une masse reliée au pendule, une simulation de la dynamique des états du système est possible (voir [12, 35, 64, 71, 77], etc.). Ces articles utilisent souvent des algorithmes non linéaires complexes et détaillés, mais s'appliquent sur un système aux dynamiques très rapides avec différentes équations de forces sur l'appareil. Le drone utilisé dans ce mémoire a quant à lui des dynamiques beaucoup plus lentes, provenant de sa masse et du mécanisme du plateau cyclique, utilisé pour générer les forces et les moments voulus. L'approche de modélisation précédemment faite sur le drone [42] se base sur une méthode d'éléments finis pour déterminer les forces générées à partir des angles de commande ainsi que de l'état en vol du système. Une relation linéaire ne peut donc pas directement se faire entre les forces/moments et les entrées du système.

Les algorithmes de contrôle utilisés se baseront donc principalement sur le modèle présent en [47] et [26], où un contrôle robuste et non linéaire ont été effectués sur un hélicoptère standard et tandem respectivement. Le choix d'un contrôle adaptatif indirect sera privilégié, de par sa souplesse de contrôle. Ce type de contrôleur permet une flexibilité sur la variation d'excitation en entrée et n'est pas restreint sur le degré relatif du système à contrôler (la différence entre l'ordre du

1. Omari [60] donne un bon exemple des dynamiques d'un multirotor

dénominateur et du numérateur d'une fonction de transfert). La commande indirecte peut aussi s'étendre à l'estimation de systèmes instables dont les entrées ne sont pas normalisées (ex. par placement de pôles adaptatifs, voir Ioannou [34]), mais il ne s'agit que d'un aspect de robustesse supplémentaire puisque ce ne sera pas le cas ici.

CHAPITRE 3 CONTRÔLE EMBARQUÉ DU MOTEUR

3.1 Problématique

Le drone de *Laflamme Aéro* comporte un puissant moteur¹ permettant de générer la force nécessaire pour faire décoller l'appareil. De la même façon qu'un hélicoptère standard, la vitesse de rotation du rotor est maintenue constante alors que le pas des pales est modifiée pour varier la force de portance. Étant donné les différentes forces s'exerçant sur les pales, il est nécessaire de réguler la vitesse du moteur pour éviter que celui-ci s'emballe ou ne génère une quantité insuffisante de portance. Il faut mentionner que le moteur utilisé comporte déjà un système gérant le mécanisme d'injection du carburant (ECU). Les entrées du système se résument donc au contrôle de l'ouverture d'entrée d'air au moteur (manette des gaz, ou *throttle*). Dû à la grande quantité de vibrations au sein de l'appareil durant un vol, il sera aussi important de concevoir le circuit du contrôleur de façon suffisamment robuste pour éviter des incidents potentiels (ex. empêcher que le moteur s'emballe ou tolérer des mauvaises mesures en entrée). L'approche utilisée ici pour concevoir le système nécessaire est en premier lieu d'approximer la dynamique du moteur pour synthétiser un premier contrôleur linéaire. Puis, faire une conception du circuit en fonction des requis du contrôleur, et finalement une validation avec le modèle approché et le drone lui-même. Une version non-linéaire du moteur est ensuite générée pour prédire son comportement dans un modèle haute-fidélité.

Le contrôle moteur se fait sur un système à priori stable, mais nécessite tout de même une grande attention sur sa robustesse et sur la stabilité des gains utilisés (surtout les gains en *hautes* fréquences), puisque tous les transitoires générées par le moteur se propageront nécessairement aux rotors, pouvant amener des perturbations sur l'attitude du drone (ses angles et sa vitesse). La commande en vitesse du moteur (ici constante) se fera donc à l'aide d'un contrôleur linéaire (proportionnel-intégral) embarqué. Les outils accessibles pour la conception sont principalement *Matlab* (et *Simulink*), *KiCAD* pour le circuit imprimé et *Eclipse IDE* pour la programmation en C du système embarqué. *Matlab* n'étant pas un simulateur temps-réel, les dynamiques du drone seront embarquées sur une seconde plateforme indépendante imitant le comportement du drone et permettant la validation du contrôleur.

1. Ce moteur permet de faire décoller un drone de plus de 300 kg, alors que la masse des multicoptères courants dépasse rarement 20 kg

3.2 Modélisation du moteur Rotax

Le moteur à asservir est un moteur quatre temps 100 HP de Rotax. Le système d'injection est géré automatiquement par l'*Engine Control Unit* (ECU), qui acquiert aussi diverses mesures du moteur et les envoie via un bus CANaero (bus CAN régi par certaines règles). De ce bus, nous pouvons donc accéder en temps réel aux données du moteur tel que l'ouverture de la soupape d'étranglement (entrée d'air), la pression dans le collecteur d'admission, la vitesse de l'arbre de sortie, etc. Le contrôle du *throttle* étant habituellement relié à un levier dans une cabine de pilotage, il s'agit ici d'un servomoteur modifiant le pourcentage d'ouverture via un signal électrique par modulation de largeur d'impulsion, appelé PWM (*Pulse Width Modulation*). Une plus grande entrée d'air apporte donc une plus grande quantité d'essence au moteur, générant un plus grand couple en sortie. Le système à contrôler peut ainsi être représenté par le schéma simplifié de la figure 3.1.

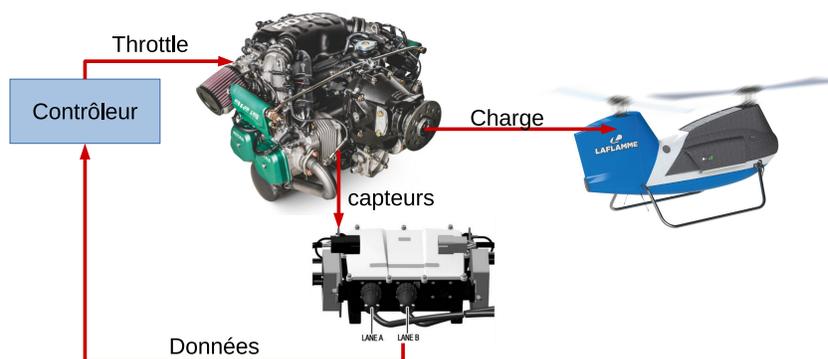


Figure 3.1 Schéma *haut-niveau* du contrôle moteur²

Les variations de l'angle d'incidence des pales ainsi que diverses perturbations en vol sont vues par le moteur comme une charge variable. Le but du contrôleur est de générer le couple nécessaire pour compenser ces dernières et stabiliser la vitesse mesurée à celle demandée en entrée (ici une vitesse en RPM). Le contrôleur doit donc lire les informations envoyées par l'ECU, calculer une nouvelle commande sous forme de largeur d'impulsion à chaque pas de temps et mettre à jour la position du servomoteur.

Pour trouver une première commande contrôlant le moteur, une estimation de la dynamique du moteur (comportement du moteur en terme de vitesse mesurée face à la variation de son entrée d'air en terme du pourcentage de *throttle* appliqué) est faite par régression linéaire sur les données prises d'essais faits par Laflamme sur leur banc de tests. Puisque le contrôleur est linéaire, la première

2. Images tirées de la fiche technique de ROTAX et du site web de Laflamme Aéro (<http://www.laflamme.aero/products/lx300/>)

estimation des dynamiques du moteur du drone sera faite dans les conditions nominales d'opération (en terme de portance mesurée par le drone et de la vitesse de rotation des pales durant un vol stationnaire *hover*). L'ensemble de données utilisé sera choisi pour contenir un moment où le moteur reçoit une commande brusque (semblable à un échelon), permettant facilement de voir la transition du moteur entre sa vitesse initiale et finale. Cette transition permettra d'approcher la fonction de transfert entre la position du *throttle* et la vitesse du moteur, c'est-à-dire sa dynamique.

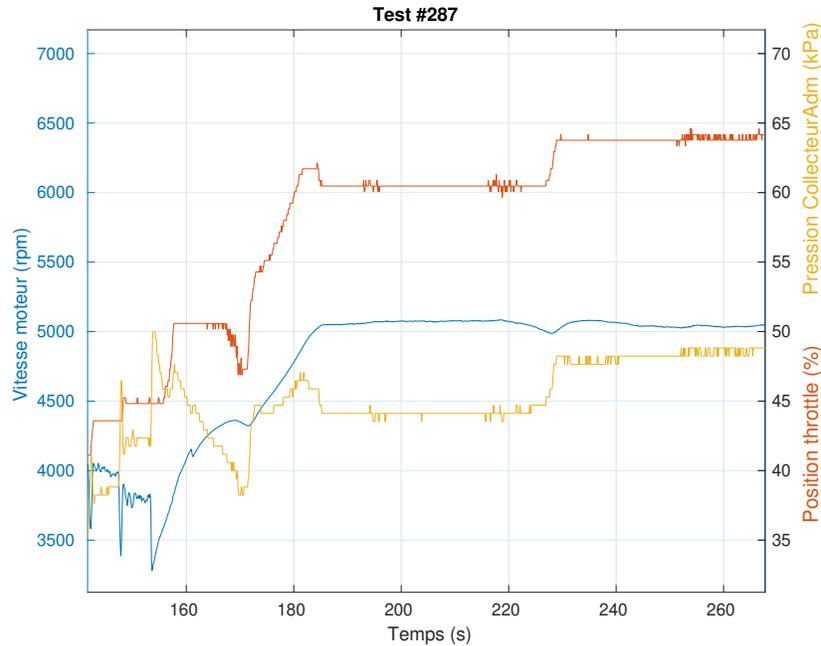


Figure 3.2 Exemple de données utilisées pour l'approximation linéaire du moteur

En partant du modèle d'état d'un système LTI (linéaire et invariant dans le temps) à une seule entrée et dont l'état est mesurable en sortie :

$$\begin{aligned}\dot{\mathbf{x}}(t) &= \mathbf{A}\mathbf{x}(t) + \mathbf{B}u(t) \\ y(t) &= \mathbf{C}\mathbf{x}(t)\end{aligned}$$

un modèle d'état discret équivalent peut être calculé via la matrice de transition Φ . Puisque les matrices \mathbf{A} et \mathbf{B} sont supposées constantes, la valeur de l'état \mathbf{x} au temps t_1 peut être calculé par :

$$\mathbf{x}(t_1) = e^{\mathbf{A}(t_1-t_0)}\mathbf{x}(t_0) + \int_{t_0}^{t_1} e^{\mathbf{A}(t_1-\tau)}\mathbf{B}u(\tau) d\tau \quad (3.1)$$

$$= \Phi(t_1, t_0)\mathbf{x}(t_0) + \Gamma(t_0)u(t_0) \quad (3.2)$$

où l'entrée u est maintenue constante entre deux pas de temps, ce qui la simplifie à $u(t_0)$. Les matrices du modèle d'état Φ et Γ se réduisent donc à :

$$\begin{aligned}\Phi(t_1, t_0) &= \Phi(t_1 - t_0, 0) = e^{\mathbf{A}(t_1 - t_0)} \\ \Gamma(t_0) &= \Gamma = \int_0^{t_1 - t_0} e^{\mathbf{A}\tau} d\tau \mathbf{B}\end{aligned}$$

Les matrices Φ et Γ seront représentées par les matrices \mathbf{F} et \mathbf{G} respectivement pour les prochaines équations. Étant donné que le pas de temps est le même pour chaque mise à jour de l'état, le système se ramène à la situation d'un bloqueur d'ordre zéro [53] où le temps t est maintenant exprimé par un nombre de pas de temps discret k :

$$t = kt_0 + T_s, \quad k \in \mathbb{R}^+$$

ce qui donne une équation matricielle aux différences finies, qui est beaucoup plus convenable pour une approximation numérique du système :

$$\begin{aligned}\mathbf{x}(k+1) &= \mathbf{F}\mathbf{x}(k) + \mathbf{G}u(k) \\ y(k) &= \mathbf{C}\mathbf{x}(k) + v(k)\end{aligned}\tag{3.3}$$

À noter que l'équation de sortie 3.3 donne la vitesse du moteur, où v est un bruit gaussien sur la mesure y . Avec l'hypothèse que la relation entre la manette des gaz $u(t)$ et la vitesse du moteur $y(t)$ peut être représentée par un modèle d'état d'ordre 2 dans le domaine fréquentiel, on obtient la fonction de transfert suivante :

$$\frac{Y(z)}{U(z)} = \frac{n_1 z + n_0}{z^2 + d_1 z + d_0}\tag{3.4}$$

où les coefficients n_i et d_j représentent les coefficients inconnus à estimer et z est l'opérateur d'avance d'un pas de temps T_s . La fonction de transfert 3.4, étant SISO, peut aisément être représentée par la réalisation suivante en utilisant la forme canonique commandable [72] :

$$\begin{aligned}\begin{bmatrix} x_1(k+1) \\ x_2(k+1) \end{bmatrix} &= \begin{bmatrix} 0 & 1 \\ -d_0 & -d_1 \end{bmatrix} \begin{bmatrix} x_1(k) \\ x_2(k) \end{bmatrix} + \begin{bmatrix} 0 \\ 1 \end{bmatrix} u(k) \\ y(k) &= \begin{bmatrix} n_0 & n_1 \end{bmatrix} \begin{bmatrix} x_1(k) \\ x_2(k) \end{bmatrix} + v(k)\end{aligned}$$

Une autre manière de représenter l'équation 3.4 est d'utiliser l'opérateur de retard z^{-1} , permettant

de montrer la relation entre la sortie actuelle et les précédentes entrées/sorties :

$$\begin{aligned}
 0 &= -Y(1 + d_1z^{-1} + d_0z^{-2}) + U(n_1z^{-1} + n_0z^{-2}) \\
 y(k) &= -d_1y(k-1) - d_0y(k-2) + n_1u(k-1) + n_0u(k-2)
 \end{aligned} \tag{3.5}$$

Afin d'estimer les éléments n_i, d_j , l'équation 3.5 est mise sous forme paramétrique de la forme :

$$\mathbf{z} = \mathbf{H}\theta + \mathbf{v}$$

avec

$$\begin{aligned}
 \mathbf{z} &= \begin{bmatrix} y(k) \\ y(k-1) \\ \dots \end{bmatrix} \\
 \mathbf{H} &= \begin{bmatrix} -y(k-1) & -y(k-2) & u(k-1) & u(k-2) \\ -y(k-2) & -y(k-3) & u(k-2) & u(k-3) \\ \dots & \dots & \dots & \dots \end{bmatrix} \\
 \theta &= [d_1 \quad d_0 \quad n_1 \quad n_0]^\top
 \end{aligned}$$

ce qui permet d'utiliser un algorithme de moindres carrés sur les mesures prises, minimisant l'impact de v [6]. La solution de ce modèle, présenté en [46], consiste à minimiser la fonction de coût suivante :

$$J = \frac{1}{2} \sum_{k=0}^n (z(k) - H(k)\hat{\theta})^2 = (\mathbf{z} - \mathbf{H}\hat{\theta})^\top (\mathbf{z} - \mathbf{H}\hat{\theta})$$

En annulant la dérivée de J , on peut isoler les paramètres θ pour en obtenir son minimum, ce qui permet d'obtenir le meilleur estimé $\hat{\theta}$:

$$\begin{aligned}
 \min J &\rightarrow \frac{dJ}{d\theta} = 0 \\
 \frac{dJ}{d\theta} &= -\mathbf{z}\mathbf{H}^\top + \mathbf{H}^\top\mathbf{H}\hat{\theta} \\
 \hat{\theta} &= (\mathbf{H}^\top\mathbf{H})^{-1}\mathbf{H}^\top\mathbf{z}
 \end{aligned} \tag{3.6}$$

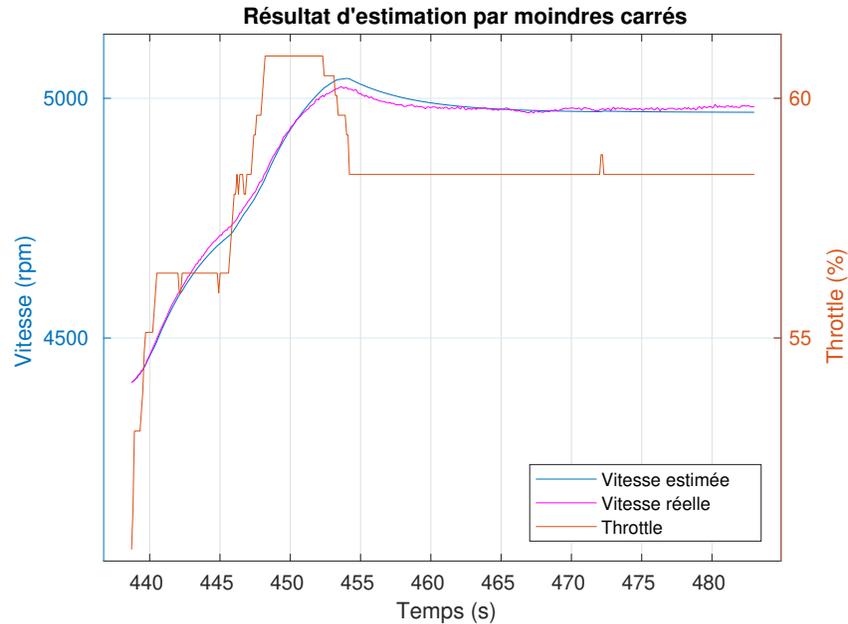


Figure 3.3 Mesure de la vitesse du moteur (mauve) et son approximation d'ordre 2 (bleue)

Parmi les tests disponibles lors de l'estimation du système, les données de la figure 3.3 (ligne magenta et orange) ont été utilisées avec l'algorithme 3.6. Cela représente 444 points pris entre 438.7s et 483s, et après calcul l'approximation suivante du comportement nominal du moteur est obtenue :

$$G(z) = \frac{1.523z + 0.4387}{z^2 - 0.717z - 0.2557} \quad (3.7)$$

La figure 3.3 montre aussi une comparaison entre les données prises du moteur et son approximation linéaire. Il demeure une légère différence sur l'estimation de la vitesse de pic, mais la différence est négligeable et ne justifie pas l'utilisation d'un modèle d'ordre supérieur. Un modèle d'ordre supérieur serait aussi davantage influencé par le bruit de mesure.

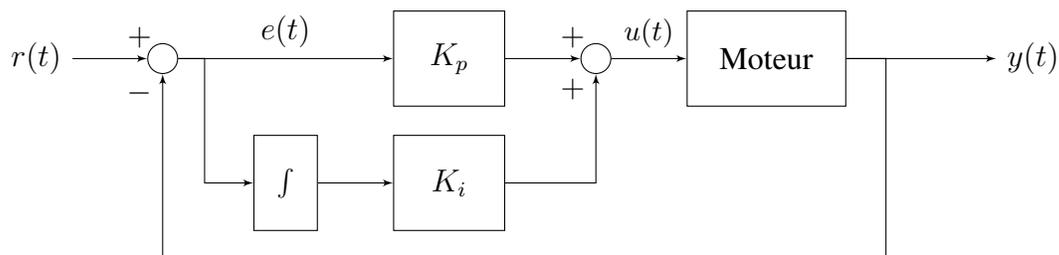


Figure 3.4 Schéma du contrôleur utilisé pour le moteur

Ayant une représentation du moteur fiable pour son mode d'opération nominal, nous pouvons synthétiser un contrôleur sur la vitesse du moteur. On effectue une synthèse dans le domaine temporel, ce qui permet d'analyser le dépassement de la commande et de facilement limiter l'effort du contrôleur pour éviter des saturations sur l'actionneur. La topologie Proportionnel-Intégral (PI, voir figure 3.4) est choisie ; la composante *dérivée* normalement présente dans un PID ne sera pas nécessaire. Étant donné qu'une mesure d'accélération du moteur n'est pas disponible, une méthode aurait été d'utiliser un *pseudo-dérivateur*, soit une approximation de la dérivée du signal en sortie et demandant une implémentation supplémentaire. L'utilité de la composante *dérivée* est principalement pour limiter le taux de variation du signal en sortie. Par contre, puisque le système est normalement stable et que les grandes variations en vitesse viennent essentiellement d'un changement brusque de la manette des gaz, un filtre sur la commande (vitesse demandée) sera utilisée. Sans ce filtre, une commande en échelon serait directement propagée dans le signal d'erreur $e(t)$ (différence entre un signal de référence $r(t)$ et la sortie mesurée $y(t)$), puis dans la sortie du contrôleur $u(t)$. La définition d'un contrôleur PI en temps continu est donc donnée comme suit :

$$\begin{aligned} e(t) &= r(t) - y(t) \\ u(t) &= K_p e(t) + K_i \int_{t_0}^t e(\tau) d\tau \end{aligned} \quad (3.8)$$

avec les gains K_p et K_i représentant les gains *proportionnel* et *intégral* respectivement. On peut représenter l'équation 3.8 dans le domaine fréquentiel via une transformée de Laplace, où les signaux fréquentiels $U(s)$, $E(s)$ correspondent aux signaux temporels $u(t)$, $e(t)$:

$$U(s) = \left(K_p + \frac{K_i}{s} \right) E(s)$$

de même que dans le domaine Z en représentant l'intégrale par son approximation numérique *backward Euler* (méthode d'Euler implicite) :

$$U(z) = \left(K_p + K_i \frac{zT_s}{z-1} \right) E(z)$$

En représentant le résultat dans le domaine temporel, on met bien en évidence la récursion de l'intégrateur sur l'erreur. Moyennant la mise en mémoire de la précédente erreur intégrée $u_i(t_n)$, l'équation de sortie du contrôleur est maintenant directement implémentable :

$$\begin{aligned} u(t_n) &= u_p(t_n) + u_i(t_n) \\ &= K_p e(t_n) + K_i T_s e(t_n) + u_i(t_{n-1}), \quad \text{où } u_i(t_0) = 0 \end{aligned} \quad (3.9)$$

Les différentes itérations pour trouver les gains du contrôleur a été faite avec la fonction `pidTuner`

de Matlab. Sur la figure 3.5, le même algorithme que `pidTuner` a été implémenté pour obtenir directement le transitoire du contrôleur pour une commande en vitesse augmentant de 1000 rpm. Il est à noter que les marges obtenues pour le contrôleur synthétisé sont de 25.303 pour la marge de gain et de $61.814T_s$ pour la marge de délai (un nombre élevé est meilleur).

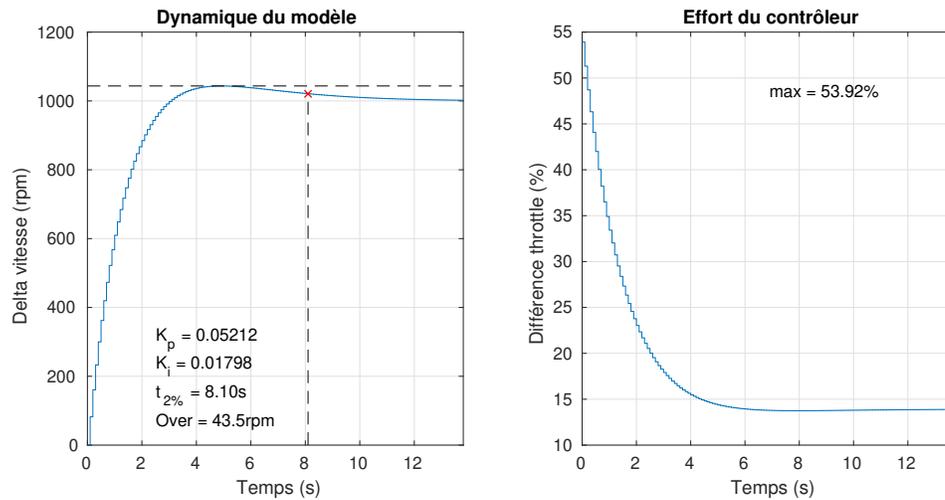


Figure 3.5 Synthèse temporelle avec le contrôleur rebouclé sur le système

Pour une vitesse initiale de 4000rpm à 40% d'ouverture du throttle, nous avons donc :

$$u_{max} = u_0 + \max(u(t)) = 93.92\%$$

Ceci justifie l'utilisation d'une étape de préfiltrage.

3.3 Limitations sur le contrôleur

Pour contrôler sécuritairement le moteur, les limitations de la figure 3.6 sont implémentées dans le contrôleur.

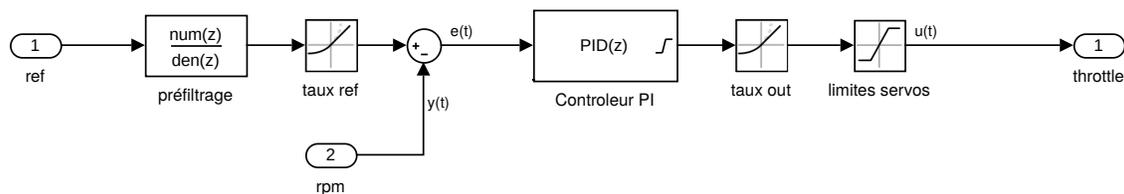


Figure 3.6 Schéma *Simulink* de la chaîne d'action du contrôleur avec ses limites

Le premier bloc (préfiltrage) est un filtre passe-bas sur la commande pour restreindre le transitoire

de la commande lors d'une grande variation. Le second bloc (*taux ref*) est une protection facultative ajoutant une restriction sur le débit de la commande en fonction du taux d'échantillonnage. Le bloc "Contrôleur PI" contient un *anti-windup*, ce qui permet à l'intégrateur de ne pas accumuler d'erreur lors d'une saturation de la commande. Enfin, les blocs "taux out" et "limites servos" réfèrent à la vitesse maximale d'actuation du servomoteur utilisé ainsi que les limites physiques en terme d'angle commandé que peut prendre le servomoteur. Le dernier point est particulièrement important puisque l'ouverture de la valve papillon ne peut excéder 95% d'ouverture ni être sous 0%. La limite supérieure assure que le ECU ne tente pas d'entrer en mode *puissance max* (ce qui pourrait occasionner un bris sur la transmission) et la limite inférieure assure que le moteur ne va pas caler. On utilise donc le filtre d'entrée suivant, ayant une constante de temps $\tau = 0.4s = 4T_s$:

$$F(s) = \frac{1}{0.4s + 1}$$

La discrétisation de ce filtre avec la méthode d'un bloqueur d'ordre zéro (*zero-order hold*) donne le filtre discret :

$$F(z^{-1}) = \frac{0.2212z^{-1}}{1 - 0.7788z^{-1}}$$

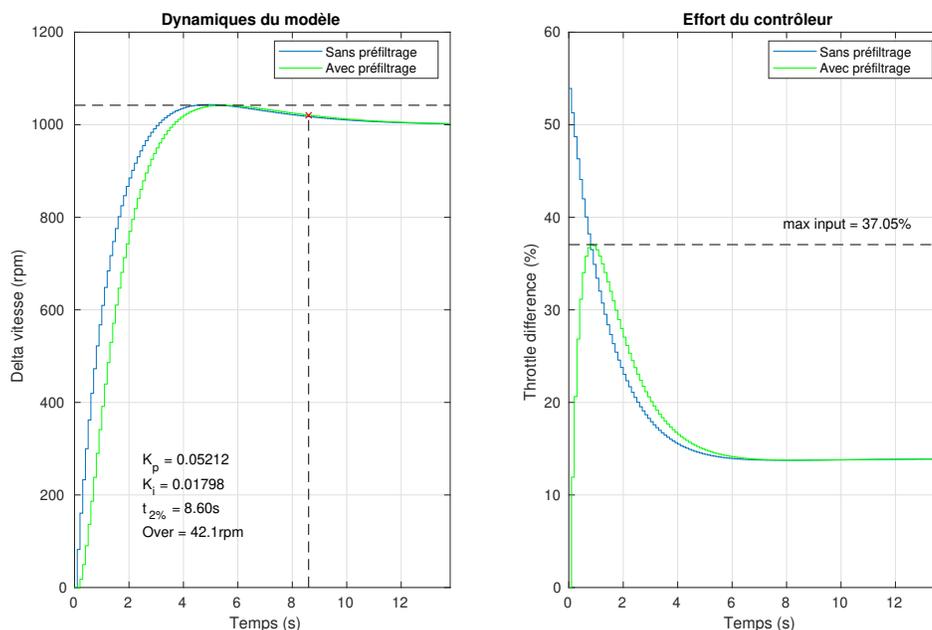


Figure 3.7 Synthèse temporelle avec le contrôleur rebouclé sur le système

Ce qui donne, en simulant de nouveau le transitoire en boucle fermée, le résultat de la figure 3.7.

L'amplitude maximale de la commande est grandement réduite, pour des transitoires presque identiques à la sortie du moteur.

3.4 Interface avec le monde réel

Les périphériques de la carte de contrôle doivent ensuite être ajustés pour que la communication entre l'ECU ou avec le servomoteur de la manette des gaz s'effectue correctement. Cette étape inclut l'implémentation des pilotes pour communiquer avec le bus CANaéro et pour communiquer avec la station de contrôle (via un câble USB). Puisque le temps de calcul du contrôleur doit être restreint à au maximum un temps d'échantillonnage T_s , un filtre matériel est utilisé pour la réception de messages CAN³. Une calibration des signaux à envoyer au servomoteur est faite et un circuit de transition entre le contrôle manuel du moteur et le contrôleur est conçu (permettant une isolation électrique des modes *manuel* et *cruise*). Enfin, le circuit électrique est réalisé à partir de composants robustes de classe automobile (Fig. 3.8) pour tolérer les grandes vibrations présentes dans le drone durant un test. Il est à noter que ce circuit a entièrement été conçu, assemblé et programmé dans le cadre de cette maîtrise, permettant de faciliter les tests en vol du drone.

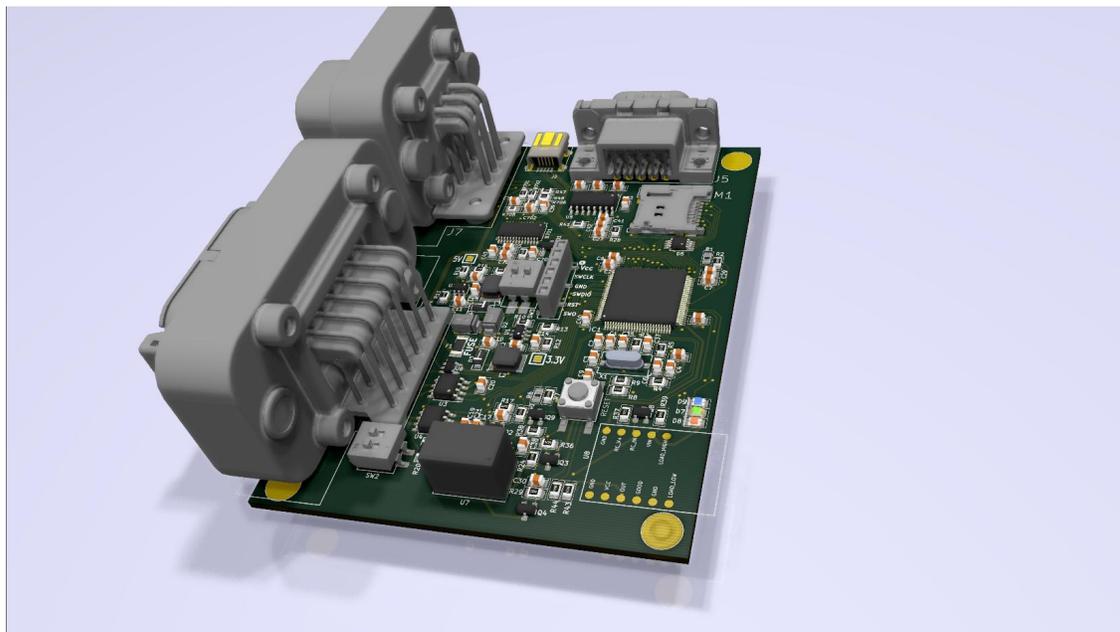


Figure 3.8 Plateforme embarquée pour le contrôle moteur

3. Le microcontrôleur choisi pour l'application comporte des *Filter banks*, évitant d'analyser chaque paquet reçu

3.4.1 Encodage du throttle

On peut voir à la figure 3.9 l'approximation polynomiale de l'encodage faite en deux sections pour relier une largeur d'impulsion du signal à envoyer (en nombre de cycle d'horloge) au *throttle* désiré (en %).

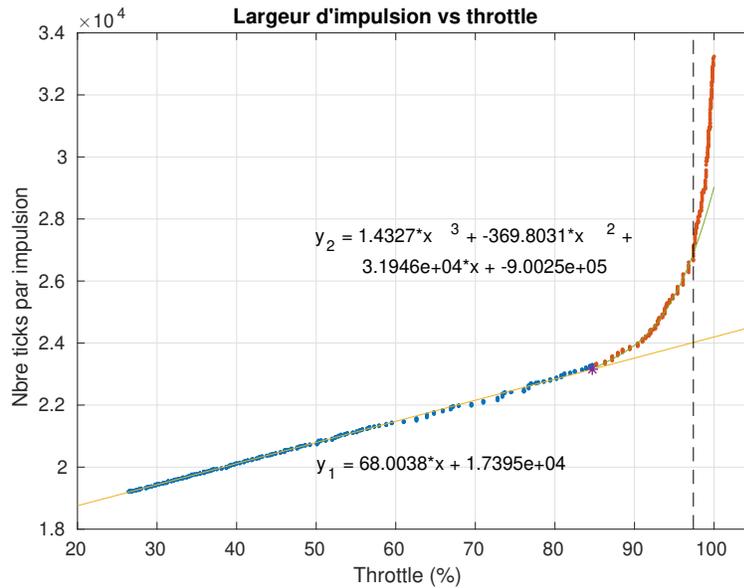


Figure 3.9 Encodage PWM du servo vers throttle

À partir de plusieurs valeurs envoyées au moteur avec la lecture de l'ECU de l'ouverture du *throttle*, la réciproque a été calculée pour connaître en boucle ouverte quel signal envoyer. Cet encodage utilise une courbe cubique (en points flottants) car le *throttle* résultant est très sensible à la position du servo. Dans le cas où il y a une grande charge sur le moteur (au-dessus de 90% d'ouverture), on désire avoir une plus grande précision sur l'encodage, car le moteur pourrait changer de mode d'opération s'il dépassait 97.5%. L'inverse de cet encodage est aussi calculé durant le mode manuel. En d'autres mots, la fonction prenant en entrée le nombre de *ticks* par impulsion et donnant le *throttle* correspondant est aussi implémentée, et ce pour connaître le premier des signaux à envoyer lors de la transition de manuel à croise.

3.4.2 Commande manuelle du pilote et transition vers contrôleur

Le pilote peut lui-même contrôler la manette des gaz et choisir le moment de la transition vers le *croise* pour stabiliser automatiquement le moteur à 5000 rpm (par exemple après que la transmission est complètement embrayée). Le circuit change électriquement l'entrée du throttle via un relais, ce qui permet de contrôler le moteur en cas de faute du contrôleur.

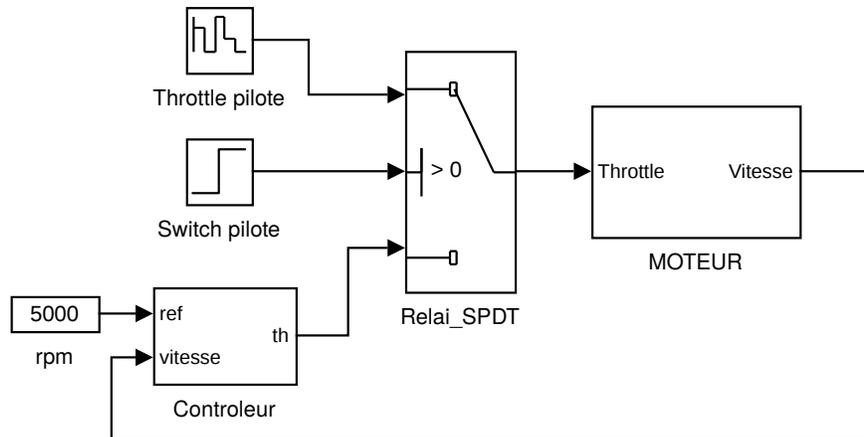


Figure 3.10 Circuit reliant le *manuel* et le *cruise*

3.5 Méthodologie de test

Le contrôleur PI de la figure 3.6 est mis à jour puis testé dans *Simulink*. Pour valider que les temps de réponse sont acceptables et que la simulation en continu pour différentes consignes garde les performances calculées, on teste avec un simulateur ayant la même interface électrique et les mêmes temps de réponse du moteur. Cela permet ensuite de valider l'*anti-windup* et les transitions du mode d'opération du contrôleur.

- Test avec simulateur "temps-réel"

Programmation du modèle linéaire du moteur dans une plateforme embarquée aux contraintes de temps identiques au drone. La communication CAN est simulée via un périphérique dédié et la valeur de throttle est envoyée par CAN. Le port UART⁴ du contrôleur est lu par un ordinateur, permettant d'afficher à chaque mise à jour de la manette des gaz l'état du système et d'envoyer de nouvelles commandes :

[temps | throttle envoyé | throttle CAN | vitesse | vitesse cible | mode]

- Test du *anti-windup* (vitesse bloquée du rotor)

À des moments aléatoires, le simulateur temps-réel gèle la vitesse de sortie, permettant de vérifier si un effet de *windup* est présent sur la commande.

- Test des transitions *manuel* ↔ *cruise* pour différentes vitesses/charges initiales

La valeur de *throttle* manuel est contrôlée en software par la plateforme de test et est lue par un *capture-compare-PWM*⁵ côté contrôleur pour connaître l'ancienne valeur et avoir une transition sans saut.

4. *Universal Asynchronous Receiver Transmitter*, un bus de communication série standard

5. Périphérique mesurant une largeur d'impulsion en chaque période

3.6 Résultats sur banc d'essai

Le moteur étant déjà installé dans le drone, une charge réaliste, provenant des pales, est déjà présente pour les tests. Le banc d'essai auquel sera fixé le contrôleur est visible à la figure 3.11. Ce banc d'essai a quatre degrés de liberté (trois angles d'Euler et le déplacement vertical) et permettra de valider la robustesse du contrôleur en situation de perturbations. Le test de la figure 3.12 montre



Figure 3.11 Banc de test 4DOF du drone, © 2018 LAFLAMME AERO INC.

qu'après que le moteur est embrayé (visible dans les grandes oscillations sur la vitesse initiale), le contrôleur passe en mode *cruise* à 113.9 s et se stabilise sans dépassement en 4.2 s. La courbe en rouge montre que l'effort du contrôleur ne dépasse pas 65% et est beaucoup plus lisse que la commande du pilote (avant 113 s). La courbe en jaune montre la pression du collecteur d'entrée, un indicateur de la charge vu du moteur.

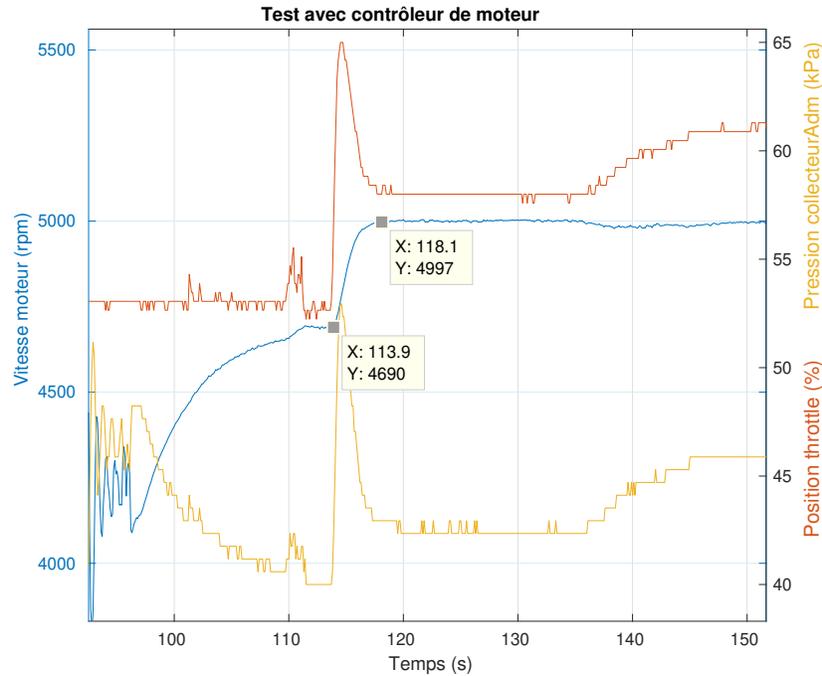


Figure 3.12 Premier test avec contrôleur de moteur

Le test de la figure 3.13 montre le retour à la valeur de référence lors de grandes perturbations. On remarque les plus grandes fluctuations de la pression du collecteur d'entrée lorsque le pilote effectue des commandes en tangage (430 s et 475 s) et le retour à la valeur nominale lorsque l'angle des pales retourne instantanément à zéro (portance nulle à 500 s). Les deux tests montrent aussi que le préfiltrage réduit les transitoires et maintient la commande en *throttle* à des valeurs sécuritaires.

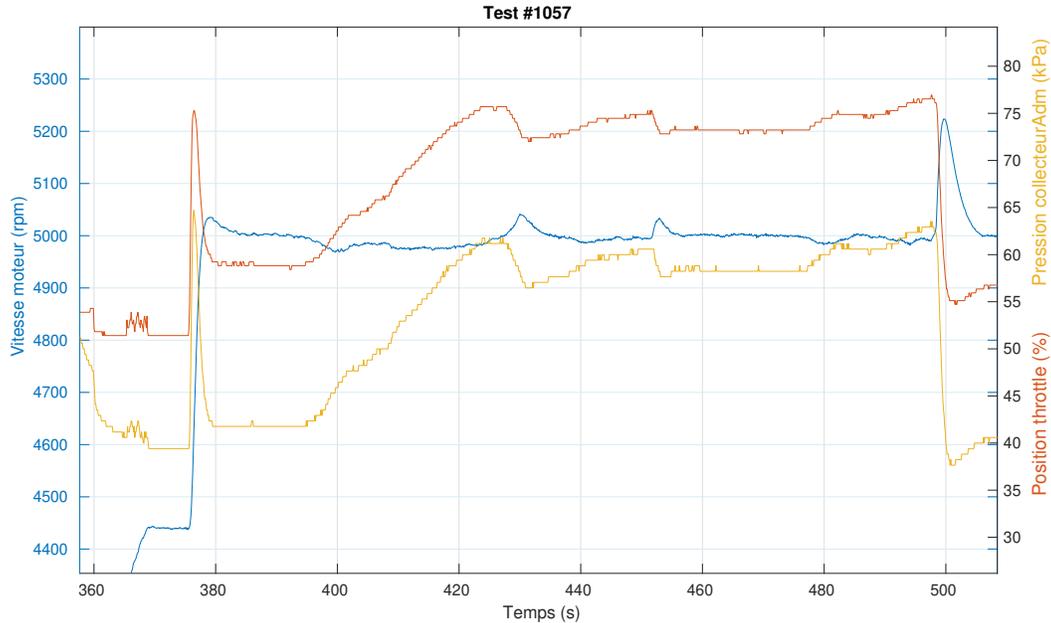


Figure 3.13 Second test avec contrôleur de moteur : rejet des perturbations

3.7 Modélisation non linéaire du moteur

Dans le projet entourant le drone tandem, un simulateur dit de *haute-fidélité* du drone a été fait pour prévoir sa dynamique en vol de façon précise. Afin d'ajouter à ce modèle le comportement du moteur, une modélisation non-linéaire est effectuée. Les équations dynamiques utilisées sont principalement basées sur les travaux de Crossley et Cook [19] et Hendricks [31], mais d'autres modèles similaires existent [8, 18, 30]. Les différentes dynamiques à modéliser sont constituées de :

1. Débit de l'air à l'entrée
2. Pression au collecteur d'admission (*intake manifold*)
3. Génération du couple en sortie
4. Vitesse en sortie

et sont identifiées dans le schéma de la figure 3.14. À noter qu'avec une fréquence de 10 Hz sur la mesure du CAN, il n'est pas possible d'observer la dynamique entre l'ouverture d'entrée d'air et le port d'admission, ni la dynamique d'explosion dans la chambre de combustion des pistons.

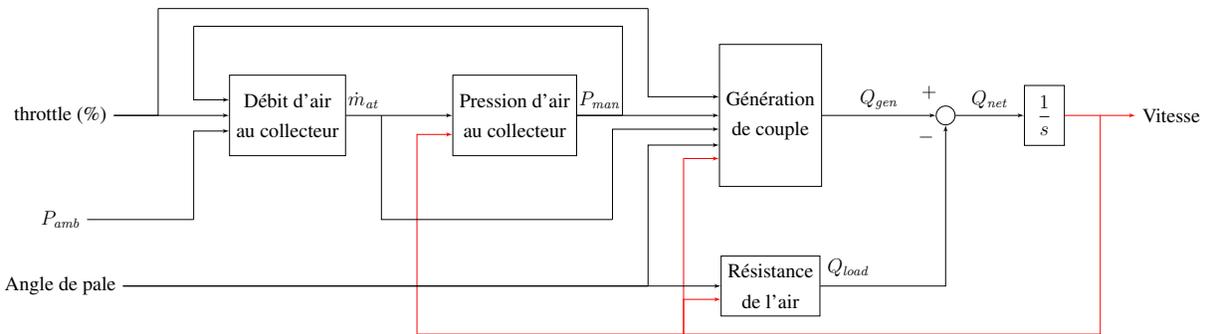


Figure 3.14 Modélisation nonlinéaire du moteur

3.7.1 Débit d'air à l'entrée

Une régression linéaire basée sur [19] est faite entre l'ouverture de la valve papillon (*throttle intake*) et la masse d'air entrante. L'équation 3.10 représente la relation entre l'angle de la valve et le flot d'air maximal théorique $\beta_1(\theta)$. L'équation 3.11, gère la différence de pression entre l'extérieur du moteur et celle du collecteur d'admission, où P_m est la pression du collecteur (*intake manifold*) et P_{amb} est la pression ambiante. Enfin l'équation 3.12 estime le débit d'air entrant dans le moteur (\dot{m}_{ap}). Les coefficients $\alpha_0, \alpha_1, \alpha_2$ ont été trouvés en se basant sur les mesures du *manifold pressure* et du *throttle* faites par l'ECU.

$$\theta = \text{throttle}[\%] \times \frac{\pi}{200} [\text{rad}/\%]$$

$$\beta_1(\theta) = \alpha_0 + \alpha_1 \cos(\theta) + \alpha_2 \cos^2(\theta) \quad (3.10)$$

$$\beta_2(P_m) = \begin{cases} 1, & P_m \leq \frac{P_{amb}}{2} \\ \frac{2}{P_{amb}} \sqrt{P_m P_{amb} - P_m^2}, & P_m > \frac{P_{amb}}{2} \end{cases} \quad (3.11)$$

$$\dot{m}_{at} = \beta_1(\theta) \beta_2(P_m) \quad (3.12)$$

$$\dot{m}_{ap} \approx \dot{m}_{at} \quad \text{car } T_s = 0.1s \quad (3.13)$$

3.7.2 Pression au collecteur d'admission

La dynamique de la pression d'air au collecteur d'admission vient normalement de la variation entre le débit d'air entrant \dot{m}_{at} et le débit d'air au port d'admission \dot{m}_{ap} (voir [18, 70]). Puisque la période d'échantillonnage couvre la dynamique entre \dot{m}_{ap} et \dot{m}_{at} , celle-ci est négligée et ces deux grandeurs sont considérées comme égales (Eq. 3.13). Puisque dans ces modèles une approximation d'ordre 2 est faite entre la vitesse N , la pression au collecteur P_{man} et le débit d'air \dot{m}_{at} , une équation

similaire est utilisée (eq. 3.14), où un élément " $z^{-1}\dot{m}_{at}$ " y est ajouté pour tenir compte de l'évolution du débit d'air.

$$P_{man} = p_1 + p_2N + p_3N^2 + p_4\dot{m}_{at} + z^{-1}p_5\dot{m}_{at} \quad (3.14)$$

Les termes p_n sont ensuite déterminés par régression linéaire sur les données des essais disponibles.

3.7.3 Génération du couple en sortie

Le calcul du couple généré est fait sur la base d'un bilan des moments à la sortie du moteur. En estimant les forces de traînée sur les rotors et en dérivant numériquement les vitesses mesurées du rotor, il est possible d'obtenir un estimé du couple généré. L'estimé des forces de traînées se base sur le *blade element method* présent dans le livre de R.W. Prouty [69]. En étant ensuite capable d'obtenir une approximation du couple généré durant une simulation, une régression linéaire est appliquée sur différentes grandeurs du système pour obtenir une relation avec ces dernières. Dans le cas où la fréquence des mesures est suffisamment haute, il est possible de modéliser les explosions (*spark ignitions*) au sein du moteur et d'utiliser le modèle de Crossley & Cook [19].

$$\begin{aligned} Q_{gen} &= Q_{net} - Q_{load} = \dot{N}I_{reflect} - Q_{load} \\ &\approx a_0 + a_1P_{man} + a_2\dot{m}_{ap} + a_3N + a_4\cos(\theta) \end{aligned}$$

avec

Q_{gen}	moment généré
Q_{net}	moment net à la sortie du moteur
Q_{load}	force de traînée convertie en moment
\dot{N}	accélération calculée du moteur
$I_{reflect}$	inertie des pales vu du moteur

Connaissant le profil de la pale, une interpolation 2D a été effectuée pour toutes les vitesses de Mach M et angles d'attaque (α) numériquement mesurés pour les coefficients de portance (C_L) et de traînée (C_D). L'interpolation effectuée permet d'obtenir un coefficient de détermination (R^2) de 0.9765 pour C_D et de 0.9942 pour C_L , selon les données de *Laflamme*⁶. Le calcul de la portance

6. La fonction `fit` de *Matlab* a ici été utilisée

permet de calculer le *drag-induced lift*, une force de traînée supplémentaire induite par la portance.

$$M = \Omega r$$

$$C_D = f(M, \alpha, \dots)$$

$$C_D(M, \alpha) \approx \sum_{i=0}^2 \sum_{j=0}^5 d_{ij} M^i \alpha^j = a_0(\alpha) + a_1(\alpha) \frac{\Omega r}{a} + a_2(\alpha) \frac{\Omega^2 r^2}{a^2}$$

où les coefficients a_0, a_1, a_2 sont une combinaison linéaire des coefficients d_{ij} et α^j . Le coefficient de portance est quant à lui donné par l'expression :

$$C_L = f(M, \alpha, \dots)$$

$$C_L(M, \alpha) \approx \sum_{i=0}^1 \sum_{j=0}^3 l_{ij} M^i \alpha^j = b_0(\alpha) + b_1(\alpha) \frac{\Omega r}{a}$$

où les coefficients b_0, b_1 sont une combinaison linéaire des coefficients l_{ij} et α^j . On peut donc calculer pour un élément de pale Δr le moment total ΔQ en fonction de la force de traînée ΔD_0 et de la traînée générée par la force de portance ΔL [69] :

$$\Delta Q = r(\Delta L \phi_s + \Delta D_0) = \Delta Q_{L \rightarrow D} + \Delta Q_D$$

$$\Delta L = \frac{\rho}{2} b_w c (\Omega r)^2 C_L \Delta r$$

$$\Delta D_0 = \frac{\rho}{2} b_w c (\Omega r)^2 C_D \Delta r$$

$$\Delta Q_{L \rightarrow D} = \frac{\rho}{2} b_w c (\Omega r)^2 \left[\frac{\theta_{tw} r}{R} + \alpha \right] C_L r \Delta r$$

En intégrant les équations précédentes, les deux résultats trouvés donnent le moment total s'opposant au moteur. Ces deux moments sont sommés et portés à la sortie du moteur, pour donner finalement :

$$Q_D(\Omega, \alpha) = 2\rho b_w \Omega^2 R^4 \left(a_0(\alpha) \frac{1}{4} + a_1(\alpha) \frac{\Omega R}{5a} + a_2(\alpha) \frac{\Omega^2 R^2}{6a^2} \right)$$

$$Q_{L \rightarrow D}(\Omega, \alpha) = 2\rho b_w \Omega^2 R^4 \left(b_0(\alpha) \left(\frac{\theta_{tw}}{5} + \frac{\alpha}{4} \right) + b_1(\alpha) \frac{\Omega R}{a} \left(\frac{\theta_{tw}}{6} + \frac{\alpha}{5} \right) \right)$$

$$Q_{load} = N_{ratio} (Q_D + Q_{L \rightarrow D})$$

avec

ΔD_0	traînée du profil de pale, parallèle au débit d'air local
ΔL	incrément de portée sur un élément de pale
ΔD_i	incrément de traînée induite par la portée

ΔQ	incrément de moment sur un élément de pale
ϕ_s	angle d'entrée d'air spécifique à un élément de pale
Ω	la vitesse d'une pale, en rad/s
a	vitesse de Mach, équivaut à $\sqrt{\gamma R_s T}$, en m/s
M	vitesse en Mach pour un élément de pale, équivaut à $\Omega r/a$
b_w	largeur de pale [m]
ρ	densité de l'air [kg/m ³]
c	nombre de pales (4)
α	angle d'attaque des pales (rad)
θ_{tw}	torsion (linéaire) de pale [rad]
Q_D	charge provenant de la traînée des pales (<i>drag</i>) [Nm]
$Q_{L \rightarrow D}$	charge provenant de la portance des pales (<i>lift</i>) [Nm]
N_{ratio}	ratio entre la vitesse angulaire des pales et du moteur

3.7.4 Vitesse en sortie

La vitesse du moteur est simplement obtenue en intégrant l'accélération du moteur.

$$N(t_n + \Delta t) = N(t_n) + \int_0^{\Delta t} \frac{Q_{net}}{I_{reflect}} dt, \quad N \geq 0$$

Le modèle non linéaire est bien entendu loin d'être parfait puisqu'il repose essentiellement sur la régression linéaire à basse fréquence de plusieurs paramètres internes du moteur. Il permettra néanmoins de valider le contrôleur linéaire et d'estimer le comportement du moteur pour des cas qui n'ont pas été mesurés.

3.7.5 Comparaison avec le modèle linéaire

Le modèle non linéaire est comparé avec l'approximation linéaire faite précédemment ainsi qu'avec des données prises lors de tests avec le moteur. Dans les trois cas, le même contrôleur PI est utilisé, et le temps initial de la simulation correspond au moment où le pilote effectue la transition de "manuel" à "cruise". Pour vérifier que le modèle puisse suivre le comportement du moteur pour différentes charges, on utilise les données d'un test où le moteur se stabilise à charge nominale, puis se stabilise de nouveau après une inclinaison rapide des pales créant une nouvelle charge demandant une ouverture du *throttle* à l'équilibre de 84% (Fig. 3.15). Puisque ces deux stabilisations se font à des moments écartés dans les tests, une partie des données a été retranchée, d'où la discontinuité à 15 secondes (en vert). Aussi, le modèle d'ordre 2 approxime seulement le système nominal (en rouge), alors il permettra seulement de valider la première étape de stabilisation du test. Même si les

données ne représentent pas exactement un échelon sur la charge du moteur tel que simulé, on peut tout de même remarquer que le temps de réponse, la valeur en régime permanent et l'effort lors du régime transitoire sont acceptables.

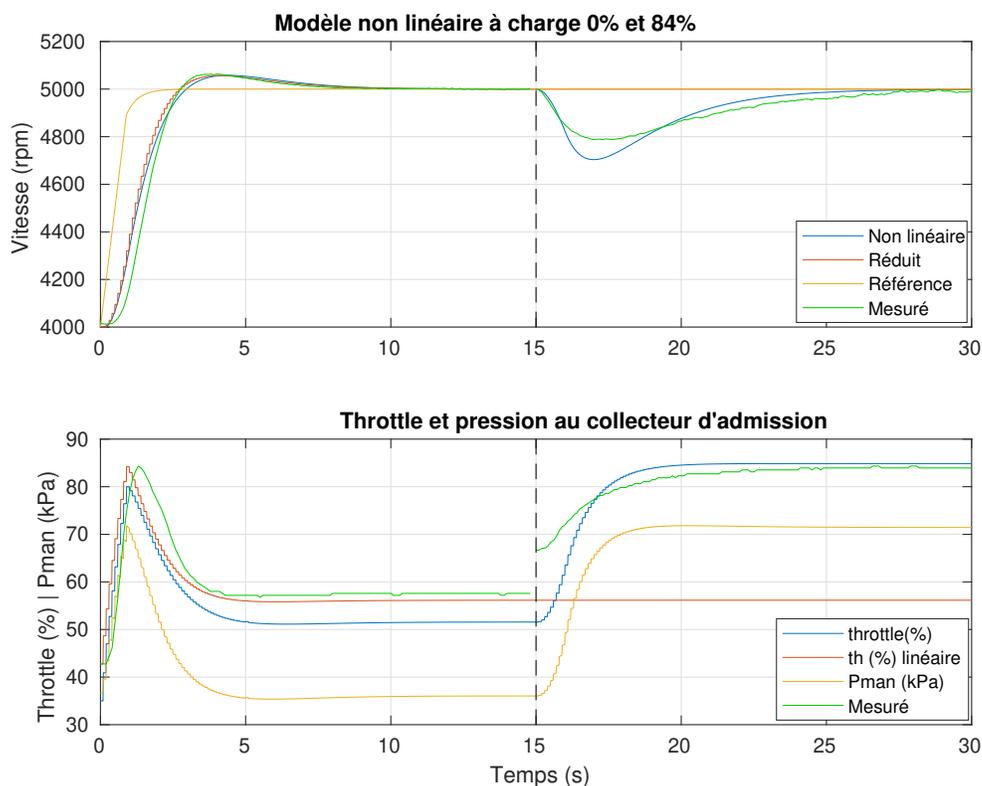


Figure 3.15 Comparaison entre le modèle linéaire et celui non linéaire

On peut donc émettre l'hypothèse que le moteur, une fois réglé avec le contrôleur synthétisé, ne provoquera pas de perturbations sur la tenue en vol du drone. Cette hypothèse repose sur le fait que l'action du contrôle moteur n'est pas saccadé lors de grandes variations de charge et que le dépassement sur la commande est très faible lors d'un transitoire. Le second point assure à l'autopilote que son action sur les pales du rotor génèrera des forces de valeurs proches à celles anticipées par sa loi de commande. D'une autre part, avec sa modélisation non linéaire, il est maintenant possible d'approcher son comportement dans un modèle haute-fidélité et de connaître sa demande en force lors de manoeuvres critiques. Pour raffiner le contrôleur utilisé, un gain *feed-forward* (valeur de *throttle* directement envoyée depuis $r(t)$) séquencée sur la pression à l'intérieur du collecteur d'entrée serait par exemple à considérer. Cet ajout aurait l'avantage d'obtenir de meilleures performances dans des cas extrêmes d'utilisation du moteur, soit loin du modèle nominal utilisé.

CHAPITRE 4 COMMANDE ROBUSTE DU DRONE

Ce chapitre présente les équations dynamiques du drone ainsi que le contrôleur robuste qui en découle. Ces équations seront évaluées en différents points d'opération, soit différentes vitesses de croisière et charges utiles. Une fois l'approximation linéaire du système obtenue, les réponses fréquentielles du système peuvent être analysées. Les bases de la commande robuste sont ensuite brièvement abordées, et une synthèse H_∞ structurée est effectuée. Cette synthèse cherchera à trouver un ensemble de gains pour chaque point d'opération du drone. Divers gabarits fréquentiels sont employés pour diriger la synthèse vers le contrôleur stabilisant, et les résultats associés y sont présentés. Ce chapitre présente donc le modèle du drone décrit par Laliberté [42], détaille la nouvelle loi de commande robuste et présente les résultats de synthèse en comparaison avec ceux obtenus via l'architecture précédente.

4.1 Modélisation du drone

Le drone est modélisé comme un corps rigide avec six degrés de liberté ; trois en position (x, y, z) et trois en orientation (ϕ, θ, ψ) . On note sur la figure 4.1 le repère fixe "origine", considéré comme inertiel et représenté par le symbole " \mathcal{O} ", et situé sur la surface de la Terre. La convention NED est utilisée pour son orientation (*North-East-Down*, soit l'axe x pointe vers le Nord et l'axe y pointe vers l'Est). Le second repère est le repère mobile "objet", représenté par le symbole " \mathcal{B} " et attaché au drone.

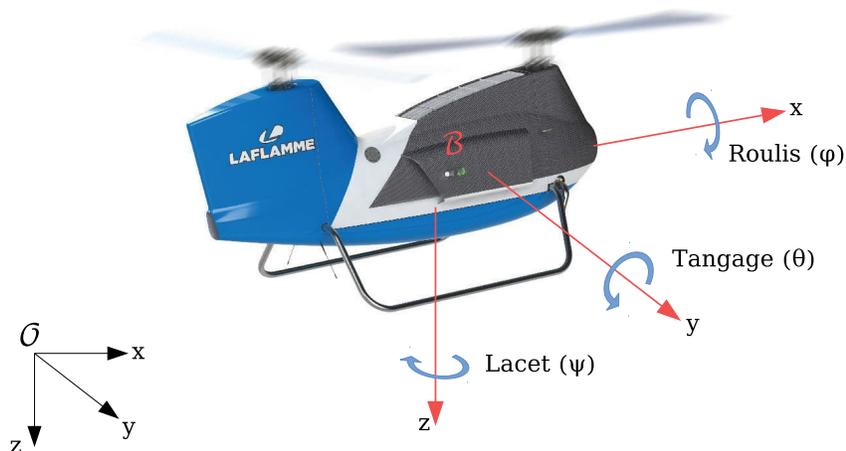


Figure 4.1 Repère du drone, © 2018 LAFLAMME AERO INC.

Pour représenter les dynamiques de l'appareil, les vecteurs suivants sont utilisés :

$$\begin{aligned}
\Phi &= [\phi \ \theta \ \psi]^\top && \text{angles d'Euler entre } \mathcal{O} \text{ et } \mathcal{B} \\
\omega_{b/o} &= [p \ q \ r]^\top && \text{vitesses angulaires de l'appareil dans le repère } \mathcal{B} \\
\mathbf{V}_{cm/o} &= [u \ v \ w]^\top && \text{vitesses du centre de masse par rapport à } \mathcal{O} \text{ dans le repère } \mathcal{B} \\
\mathbf{p}_{\text{NED}} &= [p_x \ p_y \ p_z]^\top && \text{position du centre de masse dans le repère } \mathcal{O}
\end{aligned}$$

Les équations du mouvement utilisées sont tirées de l'ouvrage de Stevens et al. [80]. Entre autres, les forces \mathbf{F} et moments \mathbf{M} agissant sur l'appareil sont représentés par les équations 4.1 et 4.2 respectivement :

$$\dot{\mathbf{V}}_{cm/o} = \frac{1}{m} \sum \mathbf{F} - \omega_{b/o} \times \mathbf{V}_{cm/o} \quad (4.1)$$

$$\dot{\omega}_{b/o} = \mathbf{I}_b^{-1} \left(\sum \mathbf{M} - \omega_{b/o} \times \mathbf{I}_b \omega_{b/o} \right) \quad (4.2)$$

où m est la masse de l'appareil et \mathbf{I}_b est la matrice d'inertie de l'appareil par rapport au repère \mathcal{B} . La matrice \mathbf{I}_b est symétrique et donnée par :

$$\mathbf{I}_b = \begin{bmatrix} I_{xx} & -I_{xy} & -I_{xz} \\ -I_{xy} & I_{yy} & -I_{yz} \\ -I_{xz} & -I_{yz} & I_{zz} \end{bmatrix}$$

Les équations 4.1 et 4.2 représentent le déplacement de l'appareil vis-à-vis du repère \mathcal{O} en considérant le changement de repères inertiels. L'équation de navigation (4.3) et l'équation de Coriolis (4.4) sont données comme suit :

$$\dot{\mathbf{p}}_{\text{NED}} = \mathbf{R}(\Phi) \mathbf{V}_{cm/o} \quad (4.3)$$

$$\dot{\Phi} = \mathbf{H}^{-1}(\Phi) \omega_{b/o} \quad (4.4)$$

permettant d'obtenir les vitesses cartésiennes et angulaires de l'appareil via une matrice de transformation sur les vitesses instantanées dans le repère \mathcal{B} . Ces matrices sont définies comme suit :

— $\mathbf{R}(\Phi)$ est la matrice d'Euler pour passer du repère \mathcal{O} au repère \mathcal{B} et donnée par :

$$\begin{aligned}
\mathbf{R}(\Phi) &= R_z(-\phi) R_y(-\theta) R_x(-\psi) \\
&= \begin{bmatrix} \cos \theta \cos \psi & \sin \phi \sin \theta \cos \psi - \cos \phi \sin \psi & \cos \phi \sin \theta \cos \psi + \sin \phi \sin \psi \\ \cos \theta \sin \psi & \sin \phi \sin \theta \cos \psi + \cos \phi \cos \psi & \cos \phi \sin \theta \sin \psi - \sin \phi \cos \psi \\ -\sin \theta & \sin \phi \cos \theta & \cos \phi \cos \theta \end{bmatrix}
\end{aligned}$$

— $\mathbf{H}(\Phi)$ est la matrice permettant d'exprimer les vitesses angulaires instantanées dans le repère \mathcal{B} en fonction de la dérivée des angles d'Euler, et donnée par :

$$\mathbf{H}(\Phi) = \begin{bmatrix} 1 & 0 & -\sin \theta \\ 0 & \cos \phi & \sin \phi \cos \theta \\ 0 & -\sin \phi & \cos \phi \cos \theta \end{bmatrix}$$

La somme des forces tient compte du poids de l'appareil, des forces aérodynamiques ainsi que des forces générées par les rotors (Eq. 4.5). Le calcul des forces générées est approché par la méthode des éléments de pales (*Blade element momentum theory*). Cette méthode se base sur le profil de la pale en chaque point de rayon en intégrant les forces aérodynamiques perçues en fonction de l'angle d'inclinaison de la pale (voir Prouty [69]). Il est donc plus simple de les représenter par ses paramètres influents (Eq. 4.6) :

$$\sum \mathbf{F} = m\mathbf{R}(\Phi)\mathbf{g} + \mathbf{F}_{wind} + \mathbf{F}_{drag} + \mathbf{F}_{gen} \quad (4.5)$$

$$\mathbf{F}_{gen,\{av/arr\}} = \mathbf{R}_y(\theta_{hub})f_{\mathbf{F}}(\mathbf{V}_{cm/o}, \omega_{b/o}, \theta_0, \theta_{1x}, \theta_{1y}) \quad (4.6)$$

avec

$\mathbf{F}_{gen,\{av/arr\}}$	force nette générée par le rotor avant (ou arrière)
\mathbf{F}_{wind}	force générée par le vent sur la surface de l'appareil
\mathbf{F}_{drag}	force de traînée aérodynamique de l'appareil en vol
$f_{\mathbf{F}}$	approximation du vecteur de force générée par le rotor
θ_{hub}	angle d'inclinaison du moyeu du rotor autour de l'axe y dans le repère \mathcal{B}
θ_0	commande du pas collectif du rotor (rad)
θ_{1x}	commande du pas cyclique longitudinal du rotor (rad)
θ_{1y}	commande du pas cyclique latéral du rotor (rad)

Les accélérations angulaires calculées proviennent des moments générés par le rotor avant (\mathbf{M}_{av}) et par le rotor arrière (\mathbf{M}_{arr}), et se calculent de façon similaire à l'équation 4.7

$$\begin{aligned} \sum \mathbf{M} &= \mathbf{M}_{av} + \mathbf{M}_{arr} \\ \mathbf{M}_{av} &= \mathbf{R}_y(\theta_{hub})\mathbf{M}_{av,z} + \mathbf{d}_{cm \rightarrow av} \times \mathbf{F}_{gen,av} \end{aligned} \quad (4.7)$$

avec

$\mathbf{M}_{av,z}$	moment généré par la rotation des pales
$\mathbf{d}_{cm \rightarrow av}$	distance entre le centre de masse de l'appareil et du rotor

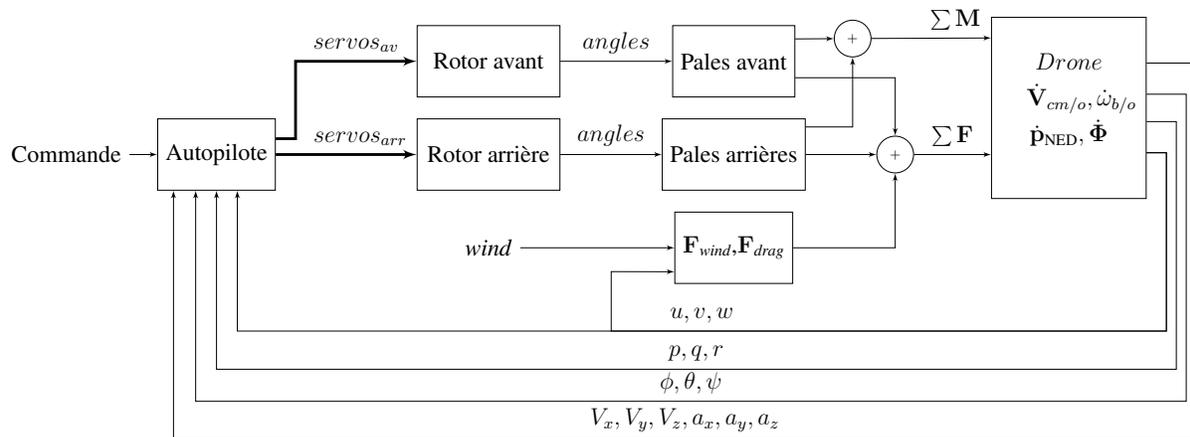


Figure 4.2 Représentation des dynamiques du drone

Le schéma bloc de la figure 4.2, implémenté dans *Simulink*, représente l'interaction des équations dynamiques du système avec le contrôleur utilisé. Le bloc *Drone* contient les équations dynamiques 4.1 à 4.4, et le calcul de la force de traînée du drone est calculée selon sa vitesse dans le repère objet.

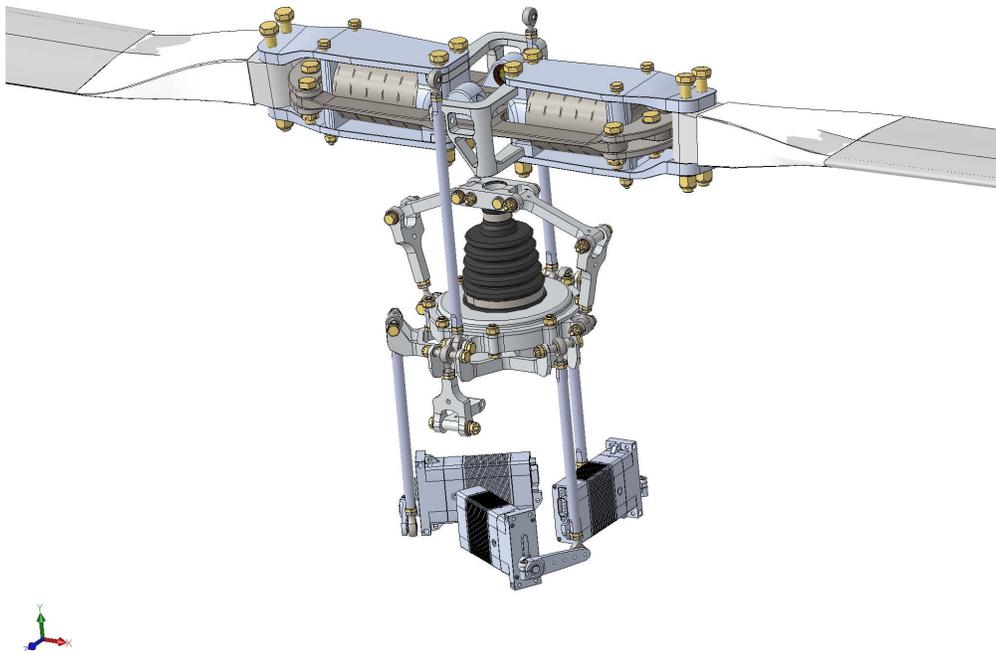


Figure 4.3 Actionneurs du plateau cyclique d'un rotor du LX300, © 2018 LAFLAMME AERO INC.

Le drone comporte des surfaces de contrôle pour modifier son état en vol. Les deux principales sont les deux plateaux cycliques¹, un sous chaque rotor, permettant de modifier l'angle des pales (Fig. 4.3). Chaque plateau comporte trois servomoteurs pour modifier son inclinaison, pouvant donc induire une force latérale, longitudinale, ou encore une force verticale lorsque les trois servomoteurs augmentent d'une même valeur.

Le positionnement des six servomoteurs du drone (trois au rotor avant et trois au rotor arrière) est géré par l'autopilote (bloc "Autopilote"). Il reçoit des mesures provenant d'un altimètre, de trois accéléromètres, d'un gyroscope, d'un GPS et d'un compas. Ces capteurs lui permettent d'obtenir les angles de roulis (ϕ), tangage (θ), lacet (ψ), leurs vitesses angulaires respectives (p, q, r), les positions cartésiennes (x, y, z) et leurs dérivées. Le contrôleur utilisé prend ces mesures et les compare avec un état de référence et applique la correction en envoyant une nouvelle commande sur les servomoteurs. Le contrôleur effectue en premier lieu la stabilisation des angles de roulis, tangage et de lacet (la boucle de stabilisation "interne"), puis effectue un suivi de trajectoire en modifiant la référence de ces angles (boucle de stabilisation "externe"). Pour simplifier la commande aux servomoteurs les signaux sont changés de repère via une fonction de *mixing*, pour que les commandes soient plutôt calculées en terme d'effort longitudinal, latéral et collectif (similaire à un avion). Ceci qui permet d'appliquer un contrôleur sur les états ϕ, θ, ψ, V_z séparément.

L'expression du *mixing*, tirée de [42], est représentée par :

$$\begin{bmatrix} servo_{av1} \\ servo_{av2} \\ servo_{av3} \\ servo_{arr1} \\ servo_{arr2} \\ servo_{arr3} \end{bmatrix} = \mathbf{B}_{mix_mec}(\theta_{av,0}, \theta_{av,1x}, \theta_{av,1y}, \theta_{arr,0}, \theta_{arr,1x}, \theta_{arr,1y}) \quad (4.8)$$

avec

$servo_{av3}$ angle du troisième servomoteur avant
 \mathbf{B}_{mix_mec} fonction reliant les angles collectif/cycliques aux angles des servomoteurs

L'équation 4.8 représente le lien mécanique du déplacement des servomoteurs sur le plateau cyclique. Il s'agit donc de la relation trigonométrique entre une inclinaison des servomoteurs et la résultante en terme d'angle collectif et cyclique durant la révolution de la pale. Les angles $\theta_0, \theta_{1x}, \theta_{1y}$ de chaque plateau cyclique, ayant des entrées redondantes, peuvent être réduites à quatre entrées découplées. Les termes de ces entrées sont empruntés de l'aviation, à savoir : une commande d'aileron (roulis,

1. il y a aussi la vitesse du moteur, mais pour des raisons de stabilité elle est maintenue à vitesse constante

δ_ϕ), d'élévateur (tangage, δ_θ), de direction (lacet, δ_ψ) et de poussée verticale / *thrust* (vitesse verticale, δ_w) par l'équation 4.9 :

$$\begin{bmatrix} \theta_{av,0} \\ \theta_{av,1x} \\ \theta_{av,1y} \\ \theta_{arr,0} \\ \theta_{arr,1x} \\ \theta_{arr,1y} \end{bmatrix} = \mathbf{B}_{mix_soft} \begin{bmatrix} \delta_\phi \\ \delta_\theta \\ \delta_\psi \\ \delta_w \end{bmatrix} \quad (4.9)$$

avec

$\delta_{\{\phi,\theta,\psi,w\}}$ commande d'aileron, élévateur, direction ou *thrust* respectivement
 \mathbf{B}_{mix_soft} matrice de *mixing* dont les gains ont été décidés pour avoir un découplage de la commande. Présentement, avec une architecture à points fixe, elle est donc donnée par l'équation 4.10, où le *bit-shift* post-calcul est explicité par le coefficient 2^{-14}

$$\mathbf{B}_{mix,soft} = 2^{-14} \begin{bmatrix} 0 & 4000 & 0 & 800 \\ 0 & 0 & 0 & 0 \\ 4000 & 0 & 600 & 0 \\ 0 & -4667 & 0 & 955 \\ 0 & 0 & 0 & 0 \\ 4800 & 0 & -700 & 0 \end{bmatrix} \quad (4.10)$$

Un contrôle Proportionnel-Intégral-Dérivée (PID) est appliqué sur les quatres états découplés (Fig. 4.4). Chaque PID suit un signal de référence pour l'état à asservir, et met à jour la commande de l'actionneur associé. Par exemple, le schéma bloc du PID pour le roulis est donné par la figure 4.5. Étant donné la nature hautement non-linéaire du système, les gains utilisés ne sont pas fixes. Les gains de chaque PID sont en fait séquencés en fonction de la vitesse longitudinale du drone (une vitesse latérale élevée est indésirable pour un drone tandem), ce qui lui permet d'utiliser par exemple des gains moins agressifs (et donc réagir plus lentement pour éviter des oscillations) à haute vitesse. Pour un contrôle séquencé en vitesse sur ϕ , la mise à jour de la commande en aileron est donnée par l'équation 4.11. On remarque que l'architecture utilise directement la sortie, puisque le gain proportionnel multiplie directement l'angle ϕ , et que le gain sur la dérivée utilise plutôt la vitesse

angulaire p , qui est directement accessible :

$$\delta_\phi(t) = K_p(V)\phi(t) + \int_{t_0}^t K_i(V) [r_\phi(\tau) - \phi(\tau)] d\tau + K_d(V)p(t) + Trim(V) \quad (4.11)$$

avec

$\delta_\phi(t)$	commande en aileron à l'instant t
$K_{\{p,i,d\}}(V)$	gain {Proportionnel / Intégral / Dérivée } séquencé en vitesse
$r_\phi(t)$	angle de roulis de référence à atteindre
$\phi(t)$	mesure de l'angle de roulis de l'appareil à l'instant t
$p(t)$	mesure de la vitesse angulaire p dans le repère objet à l'instant t
$Trim(V)$	Valeur d'aileron à l'équilibre pour une vitesse V

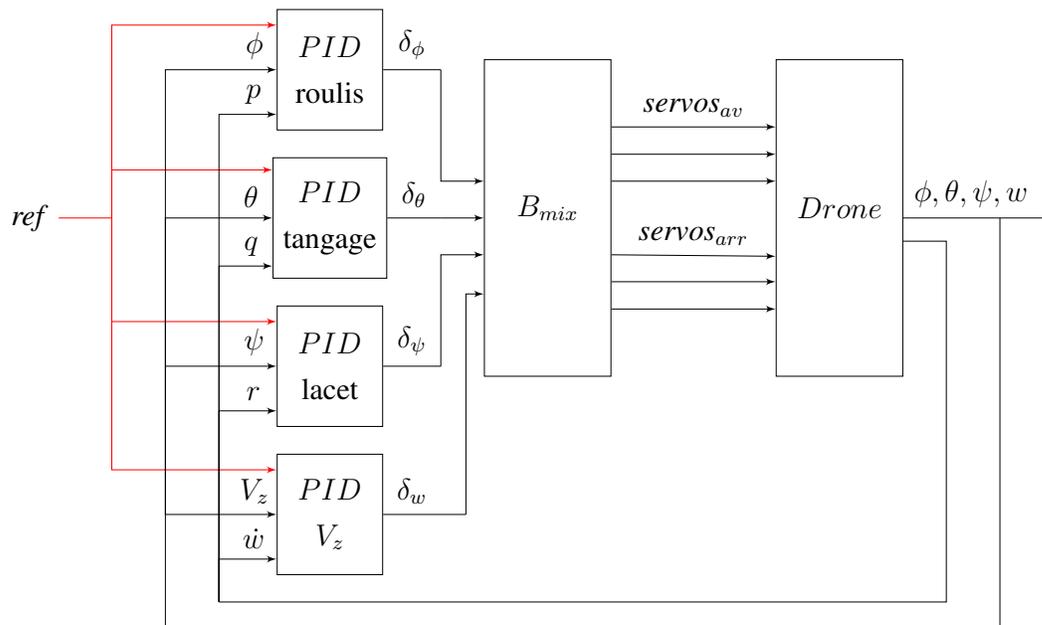


Figure 4.4 Boucle de stabilisation de l'attitude du drone

Avec des gains initiaux pour K_p , K_i , K_d , le modèle non-linéaire du drone peut être testé dans *Simulink*. Par contre, pour le calcul d'un nouveau contrôleur, le modèle linéarisé en une vitesse d'équilibre est utilisé. La méthode pour obtenir le modèle linéaire est, après avoir fixé la charge utile de l'appareil (en fixant une masse spécifique m) et une vitesse longitudinale V_x dans le modèle, d'utiliser la méthode `linearize` de *Matlab*. Par exemple, pour une linéarisation à un vol stable à

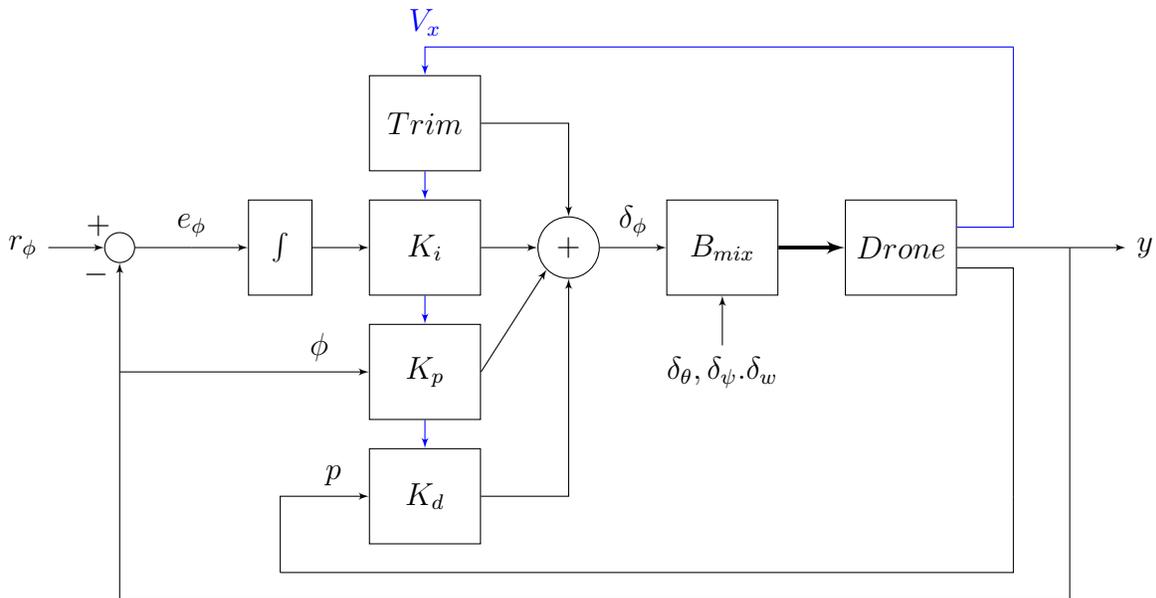


Figure 4.5 Architecture du contrôleur initial en roulis, utilisant la sortie

8 m/s, le modèle d'état synthétisé donnerait :

$$\begin{aligned} \mathbf{u} &= \mathbf{B}_{mix} \mathbf{u}_e + \Delta \mathbf{u} \\ \Delta \dot{\mathbf{x}} &= \mathbf{A} \Delta \mathbf{x} + \mathbf{B} \Delta \mathbf{u} \\ \Delta \mathbf{y} &= \mathbf{C} \Delta \mathbf{x} + \mathbf{D} \Delta \mathbf{u} \end{aligned}$$

où la commande envoyée au modèle complet est " \mathbf{u} ", comprenant les valeurs d'entrées " \mathbf{u}_e " pour maintenir le drone à l'équilibre à 8 m/s, soit les valeurs de *Trim* utilisées dans le contrôleur. Le point d'opération du modèle (valeurs de *trim*) a été trouvé via la fonction `findop` de *Matlab*. Le modèle linéaire présenté ci-haut est constitué des éléments suivants :

$$\begin{aligned} \Delta \mathbf{x} &= [\Delta \phi \quad \Delta \theta \quad \Delta \psi \quad \Delta p \quad \Delta q \quad \Delta r \quad \Delta u \quad \Delta v \quad \Delta w]^\top & (4.12) \\ \Delta \mathbf{u} &= [\theta_{av,0} \quad \theta_{av,1x} \quad \theta_{av,1y} \quad \theta_{arr,0} \quad \theta_{arr,1x} \quad \theta_{arr,1y}]^\top \\ \Delta \mathbf{y} &= [\Delta \phi \quad \Delta \theta \quad \Delta \psi \quad \Delta V_x \quad \Delta V_y \quad \Delta V_z \quad \Delta p \quad \Delta q \quad \Delta r \quad \Delta a_x \quad \Delta a_y \quad \Delta a_z]^\top \\ \mathbf{u}_e &= [0.1145 \quad -0.0463 \quad 0.7673 \quad 2.8731]^\top \end{aligned}$$

$$\mathbf{A} = \begin{bmatrix}
 0 & 0 & 0 & 1 & -0.0002 & 0.0384 & 0 & 0 & 0 \\
 0 & 0 & 0 & 0 & 1 & 0.0057 & 0 & 0 & 0 \\
 0 & 0 & 0 & 0 & -0.0057 & 1.0007 & 0 & 0 & 0 \\
 0 & 0 & 0 & 0.7499 & 0.1091 & -0.1027 & -0.0219 & -0.0617 & -0.0624 \\
 0 & 0 & 0 & 0.0638 & 1.6538 & -0.1932 & -0.0632 & -0.0151 & -0.3089 \\
 0 & 0 & 0 & -0.0106 & -0.0619 & 0.0388 & -0.0022 & 0.0228 & 0.0005 \\
 0 & -9.7994 & 0 & 0.0090 & -0.4738 & 0.0253 & -0.0060 & 0.0224 & 0.0162 \\
 9.7993 & 0.0021 & 0 & 0.4747 & 0.0215 & -8.0208 & -0.0055 & -0.0439 & -0.0150 \\
 0.0554 & -0.3766 & 0 & 0.0019 & 9.1111 & 0.0625 & -0.2870 & 0.0359 & -0.6244
 \end{bmatrix}$$

$$\mathbf{B} = \begin{bmatrix}
 0 & 0 & 0 & 0 & 0 & 0 \\
 0 & 0 & 0 & 0 & 0 & 0 \\
 0 & 0 & 0 & 0 & 0 & 0 \\
 -4.3156 & 4.2016 & 18.9064 & 2.7754 & -5.4295 & 23.5153 \\
 103.4557 & -11.5551 & 1.2987 & -105.8454 & -0.3354 & -1.4218 \\
 9.0742 & 1.4559 & 7.2545 & -9.6326 & 2.9137 & -9.1227 \\
 4.8211 & 4.6973 & -1.1176 & -0.7478 & 4.8031 & 1.0643 \\
 -0.3556 & 1.1024 & 5.0124 & 0.5187 & -1.1007 & 4.7859 \\
 -59.1701 & 3.7424 & -0.0978 & -58.3574 & 3.3213 & -0.0000
 \end{bmatrix}$$

Étant donné que les sorties a_x, a_y, a_z ne participent pas à la dynamique du système, ils proviennent simplement du calcul des forces nettes dans le repère objet. La matrice \mathbf{D} montre plus explicitement que les forces des pales viennent d'un calcul faisant intervenir tous les états ainsi que les entrées.

$$\mathbf{C} = \begin{bmatrix}
 1.0000 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
 0 & 1.0000 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 \\
 0 & 0 & 0 & 0 & 0 & 0 & 1.0000 & 0 & 0 \\
 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1.0000 & 0 \\
 0.0017 & -8.0059 & 0 & 0 & 0 & 0 & -0.0384 & -0.0057 & 0.9992 \\
 0 & 0 & 0 & 1.0000 & 0 & 0 & 0 & 0 & 0 \\
 0 & 0 & 0 & 0 & 1.0000 & 0 & 0 & 0 & 0 \\
 0 & 0 & 0 & 0 & 0 & 1.0000 & 0 & 0 & 0 \\
 0.0021 & -9.8066 & 0 & 0.0090 & -0.1236 & 0.0277 & -0.0171 & 0.0237 & -0.0078 \\
 9.7994 & 0.0021 & 0 & 0.4747 & 0.0731 & -8.0203 & -0.0072 & -0.0437 & -0.0186 \\
 0.0000 & 0.0000 & 0 & -0.0011 & 9.1223 & 0.1068 & -0.2866 & 0.0352 & -0.6245
 \end{bmatrix}$$

$$\mathbf{D} = \begin{bmatrix} 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 \\ 1.6559 & 4.9279 & -1.1036 & -1.6651 & 4.9024 & 1.0934 \\ -0.0806 & 1.0874 & 4.8481 & 0.0801 & -1.1063 & 4.9632 \\ -41.7870 & 0.1836 & -0.0704 & -42.4240 & -0.2345 & -0.0880 \end{bmatrix}$$

4.2 Synthèse H_∞ structurée initiale

L'ensemble de gains utilisés a été trouvé via une synthèse H_∞ structurée dans *Matlab* avec l'aide des travaux de Frédéric Laliberté [42]. Ce type de contrôleur se base sur la théorie de la commande robuste, où H_∞ réfère à la norme $\|n\|_\infty$ d'un système LTI dans le domaine fréquentiel. Par exemple, la norme H_2 d'un signal Y , noté $\|Y(s)\|_2$, est donné par :

$$\|Y(s)\|_2 = \sqrt{\frac{1}{2\pi} \int_{-\infty}^{+\infty} \text{trace}(Y^\top(-j\omega)Y(j\omega)) dt} \quad (4.13)$$

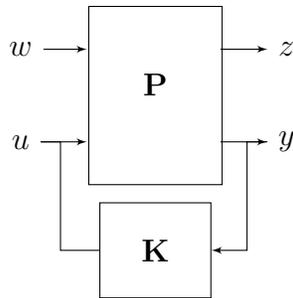
et représente l'énergie totale du signal fréquentiel Y . la norme H_∞ d'une matrice de transfert $\mathbf{G}(s)$, noté $\|\mathbf{G}(s)\|_\infty$, est donné par

$$\|\mathbf{G}(s)\|_\infty = \sup_{U(s) \in \mathcal{L}_2} \frac{\|Y(s)\|_2}{\|U(s)\|_2} = \sup_{\omega \in \mathcal{R}} \bar{\sigma}(\mathbf{G}(j\omega)) \quad (4.14)$$

représente le gain maximal que peut prendre une matrice de transfert dans le domaine fréquentiel (voir [16]) pour un système $\mathbf{G}(s)$ ayant la représentation suivante :

$$\left. \begin{array}{l} \dot{\mathbf{x}} = \mathbf{A}\mathbf{x} + \mathbf{B}\mathbf{u} \\ \mathbf{y} = \mathbf{C}\mathbf{x} + \mathbf{D}\mathbf{u} \end{array} \right\} G(s) = \frac{Y(s)}{U(s)} = \mathbf{C}(s\mathbf{I} - \mathbf{A})^{-1}\mathbf{B} + \mathbf{D}$$

Pour ω fixé, $\bar{\sigma}(\mathbf{G}(j\omega))$ est la plus grande valeur singulière de $\mathbf{G}(j\omega)$. Le problème de synthèse H_∞ est de trouver le gain \mathbf{K} stabilisant le système $\mathbf{G}(s)$ et minimisant sa norme H_∞ du transfert en z et w [24, 45].

Figure 4.6 Problème de synthèse H_∞

L'équation 4.15 montre la matrice de transfert \mathbf{P} divisée en sous-matrices de transfert $P_{a \rightarrow b}(s)$, où chacun contient les gabarits fréquentiels désirés (et possiblement la dynamique du système). Les éléments de \mathbf{P} sont représentés entre les entrées commandées (\mathbf{u}), les entrées exogènes (\mathbf{w}), et les sorties régulées (\mathbf{z}) ou les sorties mesurées (\mathbf{y}) (Fig. 4.6). Les entrées exogènes peuvent constituer, par exemple, un bruit de mesure entre certains blocs du système.

$$\begin{bmatrix} \mathbf{z} \\ \mathbf{y} \end{bmatrix} = \begin{bmatrix} \mathbf{P}_{11}(s) & \mathbf{P}_{12}(s) \\ \mathbf{P}_{21}(s) & \mathbf{P}_{22}(s) \end{bmatrix} \begin{bmatrix} \mathbf{w} \\ \mathbf{u} \end{bmatrix} \quad (4.15)$$

En utilisant la relation $\mathbf{u} = \mathbf{K}\mathbf{y}$, on obtient l'expression de la matrice de transfert entre \mathbf{w} et \mathbf{z} , notée $\mathbf{T}_{w \rightarrow z}$:

$$\begin{aligned} \mathbf{y} &= (\mathbf{I} - \mathbf{P}_{22}\mathbf{K})^{-1}\mathbf{P}_{21}\mathbf{w} \\ \rightarrow \mathbf{z} &= \left(\mathbf{P}_{11} + \mathbf{P}_{12}\mathbf{K}(\mathbf{I} - \mathbf{P}_{22}\mathbf{K})^{-1}\mathbf{P}_{21} \right) \mathbf{w} \\ \rightarrow \mathbf{z} &= \mathbf{T}_{w \rightarrow z} \mathbf{w} \end{aligned} \quad (4.16)$$

Avec la matrice de transfert de l'équation 4.16, le problème d'optimisation de la synthèse est de minimiser la norme H_∞ de $\mathbf{T}_{w \rightarrow z}$ pour que le système soit stable et se comporte conformément aux gabarits fréquentiels choisis (norme H_∞ inférieure à 1). La valeur minimale que peut prendre sa norme H_∞ , noté γ_{opt} , donne le \mathbf{K} optimal. L'objectif est toutefois considéré comme atteint pour le problème sous-optimal, soit en trouvant un gain \mathbf{K} donnant une norme H_∞ légèrement supérieur à γ_{opt} et inférieur à 1 :

$$\mathbf{K}(s) \text{ t.q. } \rightarrow \|\mathbf{T}_{w \rightarrow z}\|_\infty < \gamma < 1, \quad \gamma_{opt} \leq \gamma$$

La synthèse H_∞ effectuée est dite *structurée*, car certaines valeurs de \mathbf{K} sont imposées à 0. Puisque l'on cherche à synthétiser quatre contrôleurs PID, il n'y aura que douze gains à calculer par matrice

K. La loi de commande est alors :

$$\mathbf{u} = \mathbf{K}\mathbf{y}'$$

avec

$$\mathbf{K} = \begin{bmatrix} K_{p,\phi} & K_{i,\phi} & K_{d,\phi} & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & K_{p,\theta} & K_{i,\theta} & K_{d,\theta} & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & K_{p,\psi} & K_{i,\psi} & K_{d,\psi} & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & K_{p,Vz} & K_{i,Vz} & K_{d,Vz} & 0 \end{bmatrix}$$

et \mathbf{y}' , les signaux de sortie réordonnés, donné par :

$$\mathbf{y}' = \left[\phi \quad x_{i,\phi} \quad p \quad \theta \quad x_{i,\theta} \quad q \quad \psi \quad x_{i,\psi} \quad r \quad V_z \quad x_{i,w} \quad a_z \right]^\top$$

Les $x_{i,\{\phi,\theta,\psi,w\}}$ désignent l'état à la sortie de l'intégrateur de l'erreur de roulis, tangage, lacet ou vitesse verticale.

Le contrôleur de la figure 4.5 représente l'architecture initiale du contrôleur robuste utilisé. Les gabarits fréquentiels utilisés dans le modèle augmentée \mathbf{P} sont présentés dans le tableau 4.2. La seconde colonne réfère au nom des *Tuning goals* que la fonction `systeme` utilise pour représenter les différents objectifs de commande. Les requis durs (*hard*) sont des requis qui doivent être respectés durant l'optimisation ($\gamma_{hard} < 1$), tandis que les requis souples (*soft*) sont à minimiser, idéalement sous 1.

Tableau 4.2 Requis de synthèse H_∞

Requis	Objectif systeme	signal pondéré	Valeur
W_1	Tracking, Soft	$\phi, \theta, \psi, V_z \rightarrow \phi_c, \theta_c, \psi_c, V_{z,c}$	$\frac{1.2(s + 0.004167)}{s + 0.25}$
W_2	Margins, Soft	$\theta_{\sim,0}, \theta_{\sim,1y}, \phi, \theta, \psi, w, p, q, r, a_z$	$GM = 4 \text{ dB}$ $PM = 30 \text{ dB}$
W_3	StepRejection, Soft	$\phi_c, \theta_c \rightarrow \psi$	$\frac{0.077402s}{(s + 0.1139)^2}$
W_4	Gain, Hard	$\phi, p, q, r, a_z \rightarrow \theta_{\sim,y}$	Gain = 2 $f \in [0.1 \text{ } 100] \text{ Hz}$
W_5	MaxLoopGain, Hard	$\theta_{\sim,0}, \theta_{\sim,1y}$	$\frac{31.416}{s + 3.491}$

4.3 Motivations d'une nouvelle synthèse

Le résultat de la précédente synthèse donne d'excellentes performances sur le modèle basse-fidélité et haute-fidélité du drone. Malheureusement, le contrôleur n'a pas pu être physiquement testé dû à une incompatibilité avec l'architecture de la carte de contrôle *Micropilot*, ainsi qu'à certaines de ses limitations matérielles en terme de puissance de calcul. Le PID de cet autopilote ne prend en entrée que l'erreur avec un signal de référence, contrairement à la figure 4.5 utilisant directement la sortie. Puisque le contrôleur est linéaire, il est possible de propager l'inverse de la référence en entrée dans un *feed-forward* pour l'annuler en sortie, mais il se trouve que puisque *Micropilot* n'utilise pas de calcul en points flottants, plusieurs calculs intermédiaires doivent être effectués pour maintenir un maximum de résolution et rendent cette propagation inexacte. En effectuant une nouvelle synthèse avec une mise à jour de l'architecture du contrôleur, on laisse l'algorithme de la synthèse robuste déterminer de nouveaux gains pour les mêmes performances du drone, ce qui permet d'utiliser l'architecture native de l'autopilote².

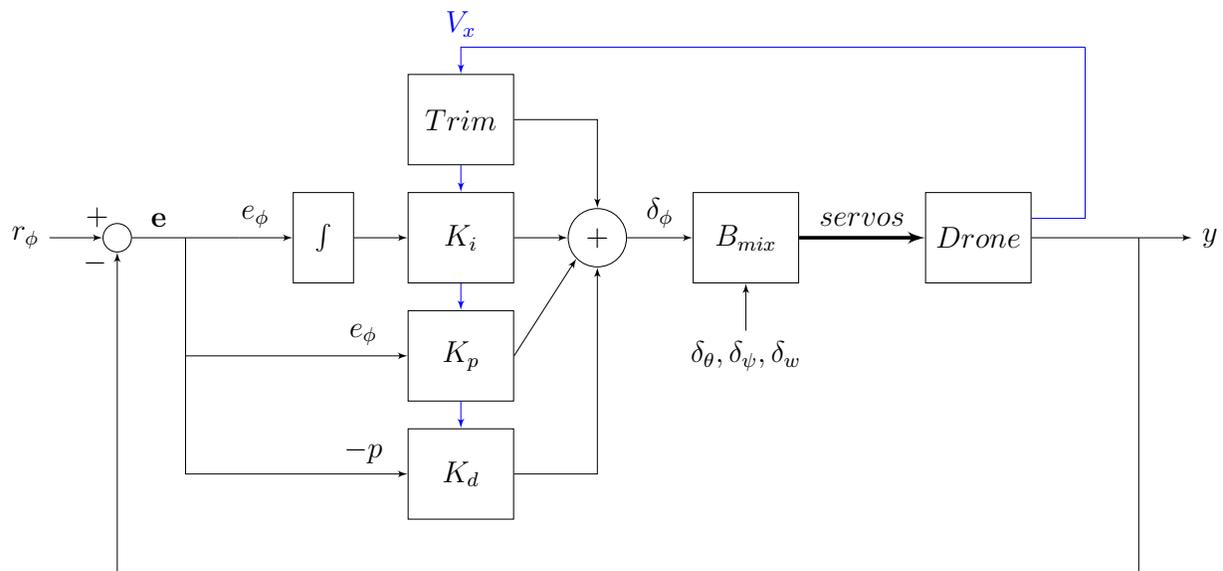


Figure 4.7 Architecture du contrôleur en roulis utilisant l'erreur

La figure 4.7 montre la modification du PID effectuée. On remarque que seul l'entrée du gain *proportionnel* est changée (la référence pour le gain *dérivée* est toujours zéro).

Dans l'optique d'utiliser l'architecture native des contrôleurs séquencés dans l'autopilote *Micropilot*, nous allons plutôt utiliser un contrôleur basée sur l'erreur entre le signal de référence et la sortie à

2. La plateforme embarquée et sa topologie sont un requis du client

contrôler. Le contrôle sur l'angle de roulis, en comparaison avec l'équation 4.11, donne maintenant :

$$\delta_\phi(t) = K_p(V) [r_\phi(t) - \phi(t)] + \int_{t_0}^t K_i(V) [r_\phi(\tau) - \phi(\tau)] d\tau - K_d(V)p(t) + Trim(V)$$

Remplacer l'architecture d'un retour de sortie vers l'utilisation de l'erreur pour le gain proportionnel du contrôleur n'est a priori qu'une question de signe sur le gain. En vol stationnaire, les angles de référence sont nuls et les gains par retour de sortie peuvent directement être utilisés, moyennant l'inversion des signes sur le *Proportionnel*. Par contre, pour des vitesses supérieures à 6 m/s (limite pour les gains du vol stationnaire), le drone n'est plus équilibré aux mêmes états en régime permanent, ce qui implique que la référence de l'erreur est non nulle et que les commandes envoyées au drone sont modifiées. D'une autre part, H. Lhachemi [44] a démontré que pour un contrôleur à gains séquencés, le profil des gains choisi est important pour un système non linéaire, car ce dernier est sensible aux taux de variation des gains selon la variable séquencée. Cela implique qu'un second terme multiplie la référence du contrôleur. La synthèse H_∞ structurée permet de prendre ce problème dans son ensemble et de trouver une nouvelle solution vers laquelle un ensemble de gains séquencés peuvent stabiliser le système.

4.4 Résultats de la nouvelle architecture

Les gains séquencés trouvés par la synthèse sont présentés à la figure 4.8. Pour le séquencement en vitesse des gains, ceux-ci étaient représentés par une équation quadratique en fonction de la vitesse de croisière. Pour un gain K_p , la synthèse modifiait donc les trois paramètres a_0 , a_1 et a_2 , avec

$$K_p(V) = a_0 + a_1V + a_2V^2$$

En bleu on peut voir le résultat des courbes quadratiques générées (la synthèse modifiait les trois coefficients d'une quadratique sur la vitesse), et en rouge l'interpolation linéaire par morceaux qui est implémentée dans *Micropilot*. À noter le plateau sous 6 m/s, qui permet de garder des gains constants en vol stationnaire.

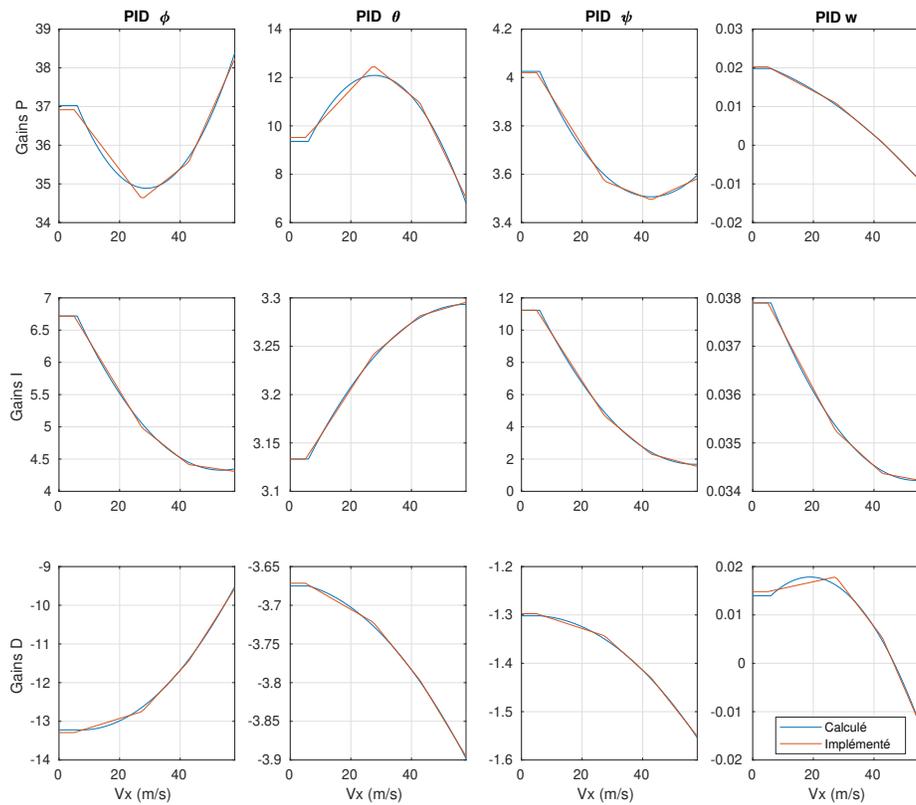


Figure 4.8 Gains séquencés trouvés par H_∞ avec erreur en entrée

Les figures 4.9 à 4.14 représentent les résultats des gabarits fréquentiels utilisés, où chaque figure contient la superposition de la réponse de chaque modèle linéarisé du drone pour différentes vitesses et charges utiles. Il est notable que ce ne sont pas tous les gabarits qui ont pu être respectés parfaitement pour l'architecture du contrôleur utilisant l'erreur en entrée, ce sont donc toutefois les meilleurs résultats qui ont pu être observés lors de la synthèse. Les vitesses longitudinales utilisées vont de 0 à 32m/s, ce qui est inférieur à la précédente synthèse. La raison est qu'avec cette architecture, $systeme$ essaiera de minimiser de plus en plus les coefficients *Proportionnel* puisque l'erreur implique des sauts sur la commande à haute vitesse, le rendant davantage oscillant. Les valeurs de γ associées à la synthèse sont les suivants :

$$\begin{aligned} \gamma_{hard} &: & 0.9993 \\ \gamma_{soft} &: & 38.6211 \end{aligned}$$

Le requis de la figure 4.9 montre que les quatre états éliminent effectivement l'erreur en régime

permanent, mais à différentes vitesses. Elles montrent aussi un léger mode de résonance des fréquences approchant 1 rad/s, soit une limite possible de la bande passante de la commande. En d'autres mots, cela implique qu'une commande trop brusque en roulis provoquerait une oscillation de 0.15 Hz sur ϕ pour certains points d'opération du drone.

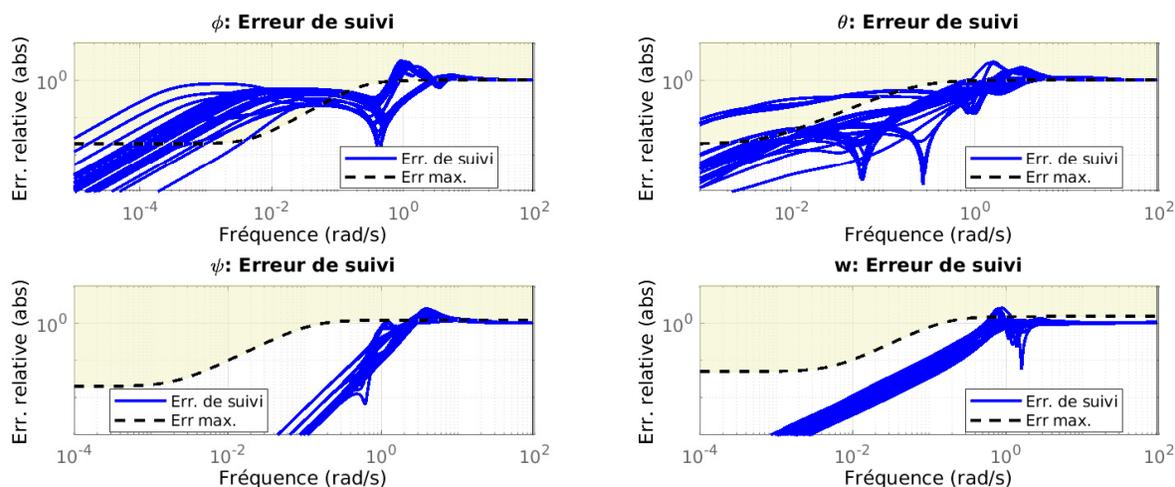


Figure 4.9 W_1 , Réponse fréquentielle d'un suivi de trajectoires

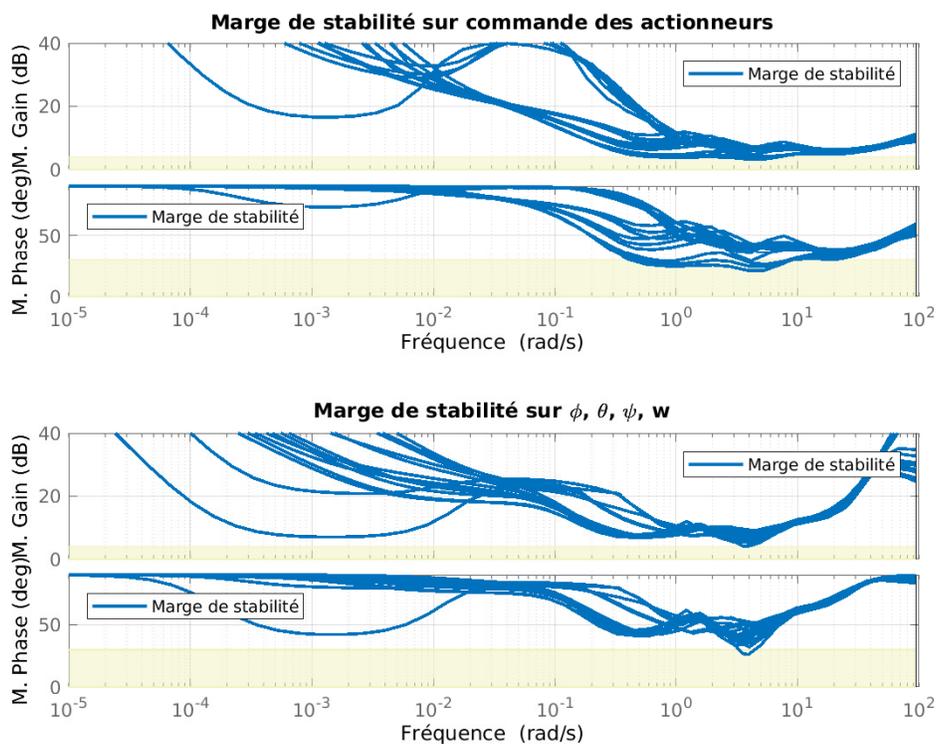


Figure 4.10 W_2 , Marge de stabilité du système

Les figures 4.11 et 4.12 montrent respectivement la réactivité du contrôleur sur la commande latérale, ainsi que la robustesse du contrôleur sur l'entrée, en termes de valeurs singulières de la fonction de transfert $T_{w \rightarrow z}$ résultante.

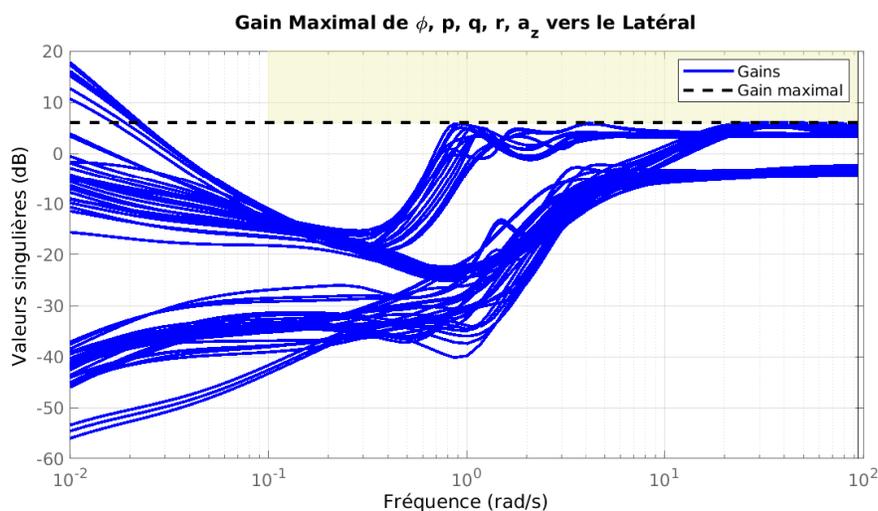


Figure 4.11 W_4 , Gain maximal sur commande latérale

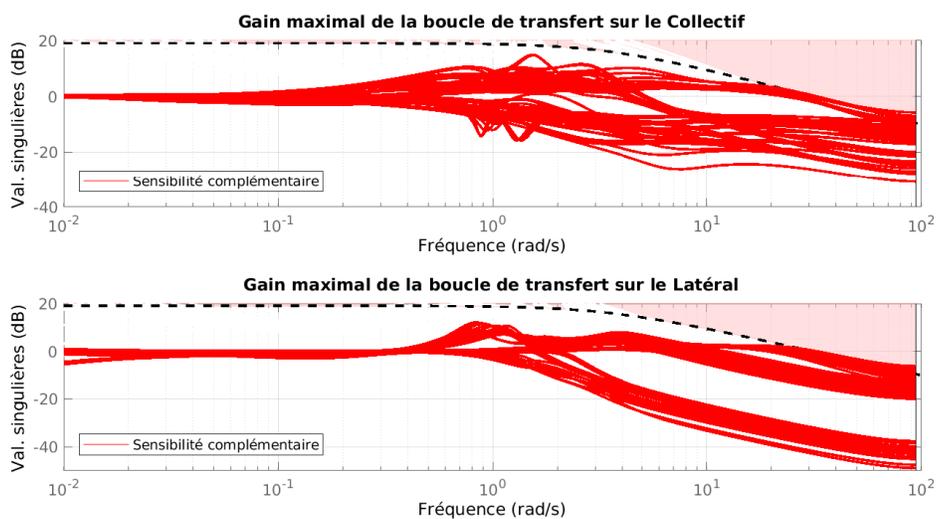


Figure 4.12 W_5 , Gain maximal sur les commandes longitudinales et latérales

La figure 4.13 montre le résultat de la propagation d'une commande échelon en ϕ et θ sur l'état ψ . Avec une architecture utilisant directement la sortie en entrée du contrôleur, cette perturbation n'est

pas présente puisque la nouvelle référence est filtrée par l'action intégrale et évite l'excitation de certains modes de résonance de l'appareil.

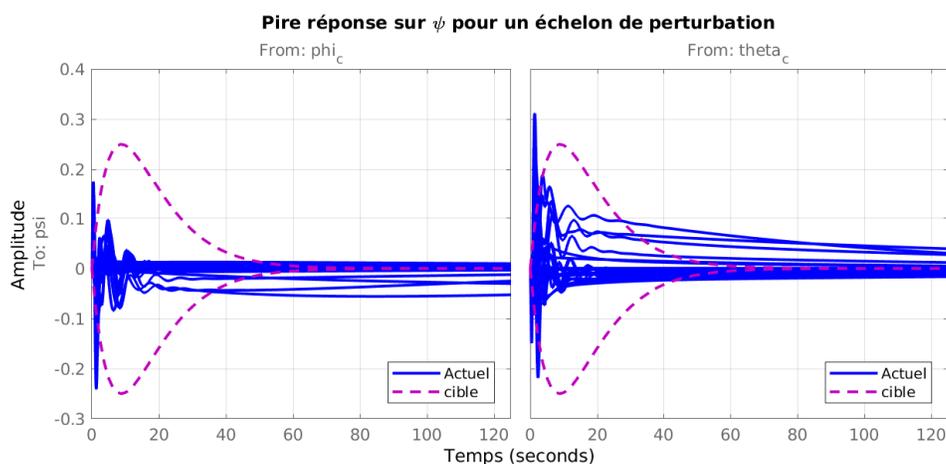


Figure 4.13 W_3 , Transitoires sur ψ pour un échelon de perturbation

On peut voir sur la figure 4.14 que les réponses temporelles comportent beaucoup d'oscillations.

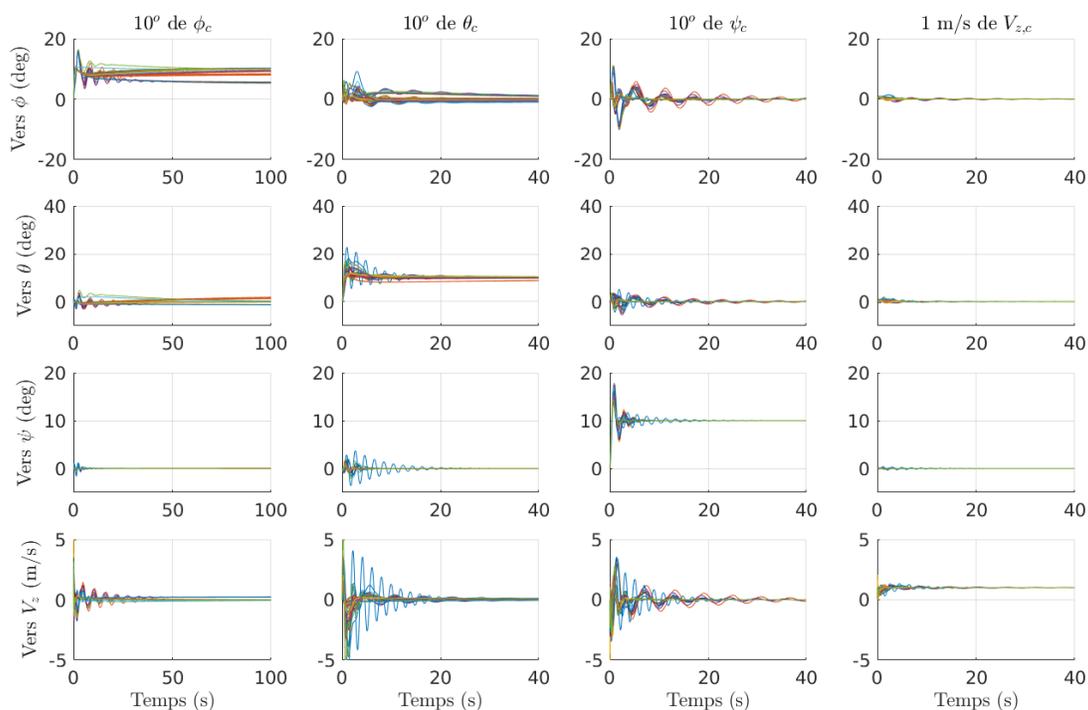


Figure 4.14 Réponses temporelles avec utilisation de l'erreur en entrée

En générant de nouveau la synthèse avec les mêmes gabarits fréquentiels pour un retour de sortie, nous pouvons grandement améliorer la réponse à l'échelon pour toute l'enveloppe de vol, visible en figure 4.15 :

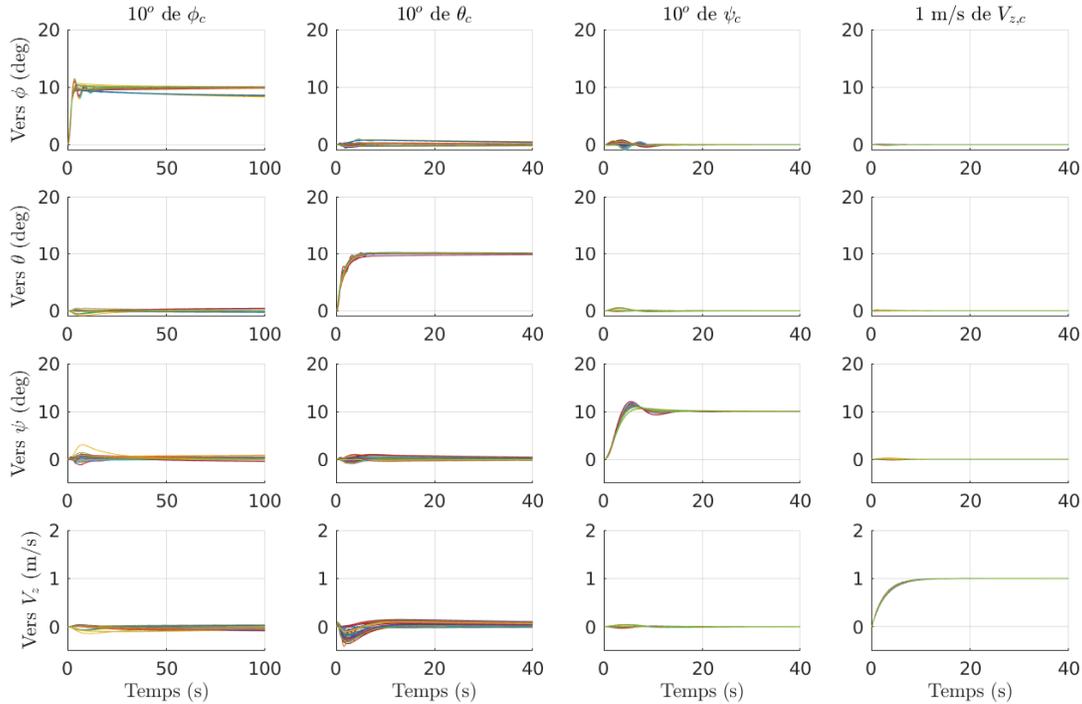


Figure 4.15 Réponse à l'échelon pour différentes vitesses et charges, avec retour de sortie

La figure 4.15 montre clairement une amélioration des transitoires sur les quatre états régulés du système. On peut voir qu'en utilisant l'erreur sur l'action proportionnelle des boucles PID, le signal de référence crée une perturbation mal amortie sur l'effort du contrôleur, qui est ensuite perçu en sortie. En utilisant un retour de sortie sur l'action proportionnelle, le signal de référence est filtré par l'action intégrale, ce qui permet un transitoire avec moins d'oscillations, un plus faible dépassement et moins de couplage entre états. La loi de commande précédemment faite pour le contrôle moteur utilisait une étape de préfiltrage sur le signal de référence, qui n'est malheureusement pas disponible ici sur la plateforme MicroPilot.

Puisque cette plateforme ne permet pas d'utiliser l'architecture de commande utilisant un retour de sortie, ni l'emploi d'une étape de préfiltrage, on peut en conclure que celle-ci ne convient pas à la stabilisation en vol du drone. D'une autre part, ces travaux ont été menés pour *limiter* les restrictions faites par MicroPilot. Celle-ci étant déjà restreinte par sa puissance de calcul et sa limite aux opérations à point fixe, son utilisation limiterait grandement les performances en vol du drone.

Après une synthèse robuste pour les différents points d'opération du modèle linéarisé du drone, on peut conclure de la meilleure architecture à utiliser pour le contrôle adaptatif du drone. L'ensemble de PID utilisant la sortie, de par leurs gains plus petits, leur meilleur découplage et l'absence d'oscillations transitoires, justifie de le prioriser aux gains de la figure 4.8 utilisant l'erreur sur l'action proportionnelle.

CHAPITRE 5 COMMANDE ADAPTATIVE DU DRONE

Il existe plusieurs méthodes de contrôle ayant un comportement adaptatif vis-à-vis de son système. La plus commune est probablement le séquençement de gains (*gain scheduling*) [73], où les gains du contrôleur évoluent en fonction d'un ou plusieurs paramètre(s) mesuré(s) à la sortie du système. Par exemple, les gains PID variant quadratiquement en fonction de la vitesse aérodynamique d'un aéronef, tel qu'utilisé avec le contrôleur robuste vu précédemment. Cette méthode est efficace, mais nécessite la connaissance du système à contrôler, ou encore des gains à utiliser. Dans le cas où les dynamiques du système sont inconnues, une méthode applicable est d'utiliser un contrôleur suffisamment conservateur pour englober toute incertitude possible du système, au détriment des performances possibles (commande robuste). Une troisième méthode, la commande adaptative, vient tenter de stabiliser les états du système en estimant en temps réel certaines caractéristiques du système et en modifiant les gains du contrôleur en fonction.

La méthode du *Model Reference Adaptive Control* (MRAC), utilisée ici, signifie que le contrôleur va modifier ses gains jusqu'à ce que le système se comporte comme un modèle de référence, souvent un système du premier ou du deuxième ordre. Par exemple, le comportement peut être défini par la position des pôles et des zéros de la fonction de transfert du système. Dans le cas d'un système non-linéaire, le modèle est le linéarisé tangent du système, variant selon le point de linéarisation (voir Khalil [37] pour différentes méthodes de contrôle non linéaires). Pour modifier les gains utilisés, le contrôleur peut être implémenté via une approche directe ou indirecte. La première signifie que les gains sont les inconnus du système et que la commande va judicieusement les modifier pour tendre vers les gains idéaux. Dans l'approche indirecte, ce sont les dynamiques (éléments de la matrice de transfert du système) qui sont inconnues et que le contrôleur doit estimer. Une fois estimés, les gains sont calculés en fonction des paramètres estimés. Dans ce chapitre, cette seconde approche sera utilisée.

5.1 Approche directe

La première classe de contrôleur adaptatif est dite directe car les inconnus du système sont les gains du contrôleur à utiliser. La synthèse de ce contrôleur vient principalement d'une analyse de stabilité au sens de Lyapunov. Avec une fonction de Lyapunov adéquate, on s'assure que les états du système en boucle fermée demeurent stables et que la dynamique de l'erreur des gains à estimer tende vers zéro. Un exemple avec une seule entrée, tiré de [34], représente bien l'approche directe. Soit le système d'ordre 1 :

$$\dot{x}(t) = ax(t) + u(t)$$

et le modèle de référence défini par :

$$\dot{x}_m(t) = -a_m x_m(t), \quad x_m(0) \neq x(0) \quad \text{et } a_m > 0$$

La commande idéale à utiliser pour que le système se comporte comme le modèle de référence serait :

$$u(t) = -k^* x(t)$$

où $k^* = a + a_m$ est le gain idéal inconnu (calculé en fonction de a). La commande réelle utilisée est plutôt $u = -kx$, ce qui amène à poser les variables \tilde{k} et ϵ représentant l'erreur sur le gain et l'erreur sur l'état respectivement :

$$\begin{aligned} \tilde{k}(t) &= k(t) - k^* \\ \epsilon(t) &= x(t) - x_m(t) \end{aligned}$$

En réarrangeant les équations pour faire apparaître \tilde{k} , on obtient la dynamique de l'erreur :

$$\dot{\epsilon}(t) = \dot{x}(t) - \dot{x}_m(t) = -a_m \epsilon(t) - \tilde{k}(t)x(t)$$

Un moyen de déterminer k est d'analyser son erreur avec une fonction de Lyapunov tenant compte de la stabilité du système ainsi que la convergence du gain k à choisir [37] :

$$\begin{aligned} V(x, \tilde{k}) &= \frac{\epsilon^2}{2} + \frac{\tilde{k}^2}{2\gamma} > 0, \quad \gamma > 0 \\ \dot{V}(x, \tilde{k}) &= -a_m \epsilon^2 - \tilde{k}\epsilon x + \frac{\tilde{k}\dot{\tilde{k}}}{\gamma} \leq 0 \end{aligned}$$

La première équation est définie positive avec une valeur de k non nulle, et la dérivée de k est trouvée en vérifiant que \dot{V} est semi-définie négative. Un exemple possible de la dynamique de k pouvant stabiliser le système au sens de Lyapunov est donnée par la loi de contrôle adaptative directe suivante :

$$\dot{\tilde{k}}(t) = \dot{k}(t) = \gamma \epsilon(t)x(t) \tag{5.1}$$

impliquant que la valeur de k provient d'un intégrateur ajouté au système.

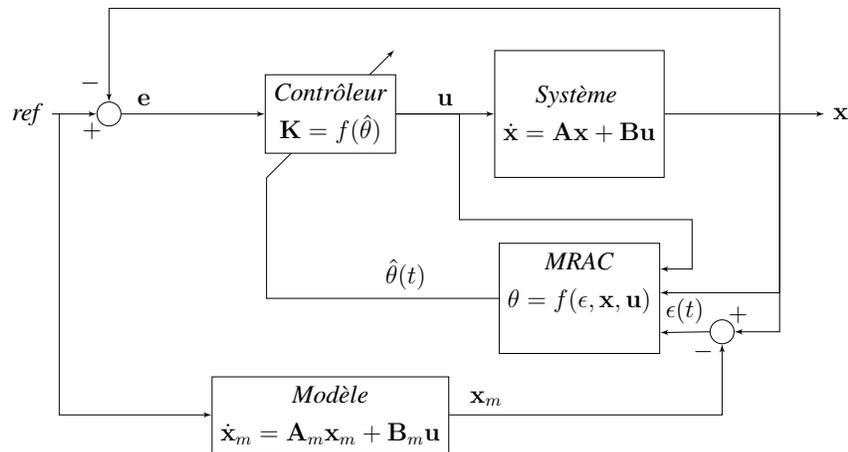


Figure 5.1 Architecture du contrôleur avec adaptation directe

Le schéma 5.1 montre une version générale d'une adaptation directe, où les paramètres inconnus sont contenus dans les matrices \mathbf{A} et \mathbf{B} et où la matrice de gain est contenue dans θ . Les dynamiques du système sont exprimées dans le domaine continu, mais pourraient très bien être représentées par un modèle d'état discret.

L'exemple précédent montre un contrôleur simple, mais trouvé sous l'hypothèse que le système est suffisamment excité pour que la dynamique du gain évolue vers k^* . Lorsque ce n'est pas le cas, il est dit qu'il n'y a pas une *persistente d'excitation*, et aucune convergence n'est garantie sur le système. De plus, pour des systèmes à plusieurs entrées et plusieurs sorties, le paramètre γ est en fait une matrice définie positive qui doit être choisie judicieusement car elle détermine la vitesse de convergence des dynamiques de la matrice de gains. Le choix de cette matrice peut devenir rapidement complexe s'il y a énormément de couplage entre les états.

5.2 Approche indirecte

L'approche indirecte est celle demandant davantage de calcul, mais donnant généralement les meilleurs résultats (voir Oudin [62]) en termes de délais sur la commande. Le processus consiste à estimer les dynamiques du système, puis de calculer le meilleur gain correspondant à la matrice de transfert trouvée. De manière analogue à l'exemple de l'approche directe, pour contrôler le système on utiliserait plutôt l'estimé de a (dénomé \hat{a}) dans le calcul du gain à utiliser :

$$k(t) = \hat{a}(t) + a_m$$

$$\tilde{a}(t) = \hat{a}(t) - a$$

La dynamique de l'erreur peut ensuite être représenté en fonction des paramètres estimés :

$$\dot{\epsilon}(t) = -a_m \epsilon(t) + \tilde{a}(t)x(t)$$

Ce qui permet d'obtenir la fonction de Lyapunov déterminant la stabilité de l'estimateur :

$$V(\epsilon, \tilde{a}) = \frac{\epsilon^2}{2} + \frac{\tilde{a}^2}{2\gamma} > 0, \quad \gamma > 0$$

$$\dot{V}(\epsilon, \tilde{a}) = -a_m \epsilon^2 - \tilde{a} \epsilon x + \frac{\tilde{a} \dot{\tilde{a}}}{\gamma} \leq 0$$

et où l'on peut déduire la dynamique de l'estimé :

$$\dot{\tilde{a}} = \hat{a} = \gamma \epsilon x$$

L'approche indirecte a l'avantage de donner une "fenêtre" au programme pour évaluer l'état de l'estimation, et de donner plus de flexibilité sur l'algorithme de calcul du gain adaptatif. L'exemple précédent utilise une fonction de Lyapunov pour déterminer l'algorithme de l'estimateur, et le gain est calculé de façon intermédiaire avec cet estimé. Cela veut aussi dire que dans le cas où \hat{a} a une représentation particulière dans le système, le calcul du gain garde un sens physique. À partir du moment où les dynamiques sont correctement estimées, le gain correcteur peut être mis à jour instantanément, permettant de garder le système en phase avec le modèle de référence.

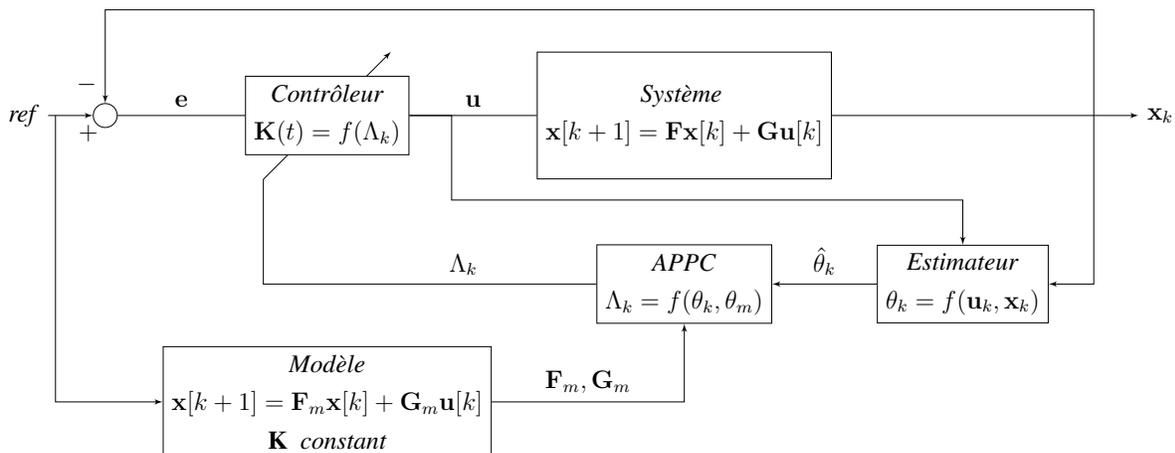


Figure 5.2 Architecture du contrôleur avec adaptation indirecte

On peut voir sur la figure 5.2 que le contrôleur est doté d'un estimateur déterminant les paramètres

inconnus de \mathbf{F} , \mathbf{G} , ainsi qu'un bloc correcteur *MRAC* calculant les nouveaux gains du contrôleur basé sur les dynamiques du modèle de référence ainsi que des dynamiques estimées $\hat{\theta}_k$. Les dynamiques du système et du modèle ont été mis dans le domaine discret pour montrer le lien avec l'estimateur discret présenté plus loin.

5.2.1 Estimation du système

Il existe plusieurs méthodes pour estimer les paramètres inconnus du système. L'utilisation d'une fonction de Lyapunov a été utilisée précédemment de par sa simplicité pour un système à une seule entrée (voir aussi [78], [86],[13] pour d'autres estimateurs utilisant une fonction de Lyapunov). Les méthodes généralement utilisées se basent sur des algorithmes de la famille des filtres de Kalman, soit les moindres carrés (dont la preuve de stabilité peut aussi se faire via une fonction de Lyapunov, voir annexe A). Un algorithme de moindres carrés purs permet de déterminer le meilleur estimé des paramètres d'un système, mais nécessite l'inversion de la matrice contenant toutes les données, ce qui ne le rend utile qu'après simulation. Une version récursive de l'algorithme est donc normalement utilisée, c'est-à-dire que l'estimateur calcule le nouvel estimé en ne tenant compte que de la valeur du précédent estimé ainsi que des nouvelles mesures.

Puisque l'on travaille dans le domaine discret, on cherche à estimer les matrices $\Phi(t_1, t_0)$ et $\Gamma(t_0)$ du modèle linéarisé du drone (Eq. 3.2). La matrice Φ correspond à la jacobienne du modèle non linéaire de l'instant t_0 à l'instant t_1 , et la matrice Γ représente l'interaction nette des entrées du système avec ses états. En se basant sur les travaux de Goodwin [28] et Oudin [62], il est possible d'obtenir Φ par un algorithme de moindres carrés récursifs sur chaque équation dynamique d'états. Le modèle dynamique MIMO linéarisé en un point d'opération est donné par le système suivant :

$$\begin{aligned}\mathbf{x}(t) &= \Phi(t, t_0)\mathbf{x}(t_0) + \Gamma(t_0)\mathbf{u}(t_0) \\ \mathbf{y}(t) &= \mathbf{C}\mathbf{x}(t) + \mathbf{D}\mathbf{u}(t)\end{aligned}$$

où au point de linéarisation on assume le système LTI et un pas de temps constant valant T_s , soit la période d'échantillonnage. La simplification suivante peut donc être faite :

$$\begin{aligned}\Phi(t + T_s, t) &= \Phi(T_s, 0) = \mathbf{F} \\ \Gamma(t) &= \mathbf{G}\end{aligned}$$

et le système peut être représenté dans le domaine discret, où la notation $\mathbf{x}[k]$ représente le k^e

échantillon du vecteur de signal \mathbf{x} :

$$\begin{aligned}\mathbf{x}[k+1] &= \mathbf{F}\mathbf{x}[k] + \mathbf{G}\mathbf{u}[k] \\ \mathbf{y}[k] &= \mathbf{C}\mathbf{x}[k] + \mathbf{D}\mathbf{u}[k]\end{aligned}$$

Étant donné que le vecteur d'états (Eq. 4.12) est entièrement mesuré à la sortie \mathbf{y} et que la matrice \mathbf{D} est nulle pour la mesure de tous les états, on ne considère que les éléments des matrices \mathbf{F} et \mathbf{G} dans la construction de l'estimateur. En représentant chacun des éléments des matrices \mathbf{F} et \mathbf{G} ,

$$\mathbf{F} = \begin{bmatrix} f_{11} & \cdots & f_{1n} \\ \vdots & & \vdots \\ f_{n1} & \cdots & f_{nn} \end{bmatrix}_{n \times n}$$

$$\mathbf{G} = \begin{bmatrix} g_{11} & \cdots & g_{1m} \\ \vdots & & \vdots \\ g_{n1} & \cdots & g_{nm} \end{bmatrix}_{n \times m}$$

on peut représenter chacune des équations du système individuellement. À noter qu'avec cette méthode, nous avons $n \times (n + m)$ éléments à estimer, ce qui est beaucoup plus élevé que le nombre de paramètres inconnu du modèle d'état continu. Cette méthode a l'avantage d'effectuer tous les calculs dans le domaine discret mais, puisqu'il y a sur-paramétrisation, les paramètres de \mathbf{F} et \mathbf{G} ne sont pas linéairement indépendants et la solution n'est plus unique. Le niveau d'excitation perçu par l'estimateur sera donc en réalité plus faible et une attention sera portée sur la valeur de l'estimé initial lors de la mise à jour des gains de la loi de commande. La dynamique du i^e état donne donc :

$$x_i[k+1] = f_{i1}x_1[k] + f_{i2}x_2[k] + \dots + f_{in}x_n[k] + g_{i1}u_1[k] + \dots + g_{im}u_m[k]$$

ce qui permet de découpler les paramètres inconnus des états et de représenter le système sous sa forme paramétrique [34] :

$$\begin{aligned}z_i[k] &= \mathbf{h}_i^\top[k]\theta_i^* & (5.2) \\ \mathbf{h}_i[k] &= \begin{bmatrix} x_1[k] & x_2[k] & \cdots & x_n[k] & u_1[k] & \cdots & u_m[k] \end{bmatrix}^\top \\ \theta_i^* &= \begin{bmatrix} f_{i1} & f_{i2} & \cdots & f_{in} & g_{i1} & \cdots & g_{im} \end{bmatrix}^\top\end{aligned}$$

L'équation 5.2 présente le modèle parfait de la dynamique de x_i au temps $[k+1]$. En réalité z_i sera

plutôt entaché d'un bruit blanc gaussien ¹ représenté par v .

$$z_i[k] = \mathbf{h}_i^\top[k] \theta_i^* + v[k] \quad (5.3)$$

$$= z_k = \mathbf{h}_k^\top \theta + v_k \quad \text{pour la } i^e \text{ colonne} \quad (5.4)$$

Pour les prochaines équations, la notation pour un pas de temps $[k]$ (eq. 5.3) sera remplacée par un indice k (eq. 5.4), étant donné que les équations développées se feront toujours sur la i^e colonne. Étant donné θ_i inconnu, l'algorithme à utiliser pour l'estimer peut être représenté par la minimisation de la fonction de coût J_k définie au pas de temps k suivante [14] :

$$J_k(\theta_k) = \sum_{\tau=1}^k \lambda^{k-\tau} \left(z_\tau - \mathbf{h}_\tau^\top \theta_k \right)^2, \quad 0 < \lambda \leq 1 \quad (5.5)$$

avec θ_k représentant la valeur de l'estimé au temps k . Le paramètre λ est nommé le facteur d'oubli, car dans la fonction de coût il modifie exponentiellement la pondération de l'erreur entre l'estimé et les mesures précédentes. L'algorithme des moindres carrés, pour déterminer le minimum de la fonction de coût, évalue la dérivée de la fonctionnelle 5.5 par rapport au vecteur d'intérêt θ_k :

$$\begin{aligned} \frac{\partial J_k}{\partial \theta_k} &= -2 \sum_{\tau=1}^k \lambda^{k-\tau} \left(z_\tau \mathbf{h}_\tau^\top - \mathbf{h}_\tau^\top \theta_k \mathbf{h}_\tau^\top \right) \\ \min J_k &\rightarrow \frac{\partial J_k}{\partial \theta_k} = 0 \end{aligned}$$

et le minimum local de la fonctionnelle correspond au point où la dérivée partielle est nulle. En isolant θ_k de l'équation, on obtient la forme des moindres carrés avec facteur d'oubli :

$$\begin{aligned} 0^\top &= \sum_{\tau=1}^k \lambda^{k-\tau} \mathbf{h}_\tau \left(z_\tau - \mathbf{h}_\tau^\top \theta_k \right), \quad \text{où } \theta_k^\top \mathbf{h}_\tau = \mathbf{h}_\tau^\top \theta_k \\ \theta_k &= \left(\sum_{\tau=1}^k \lambda^{k-\tau} \mathbf{h}_\tau \mathbf{h}_\tau^\top \right)^{-1} \left(\sum_{\tau=1}^k \lambda^{k-\tau} \mathbf{h}_\tau z_\tau \right) \end{aligned}$$

1. Dans le cas où le bruit n'est pas blanc, il est possible de le décolorer en augmentant l'état d'une dynamique du bruit avec un observateur [15]

Cette forme n'étant pas convenable au calcul *en ligne*, une forme récursive en est découlée. On note \mathbf{P}_k le premier élément de la solution, représentant aussi la covariance² des éléments de \mathbf{h}_k . On exprime donc \mathbf{P}_k selon sa précédente valeur au temps $[k - 1]$:

$$\begin{aligned}\mathbf{P}_k^{-1} &= \left(\sum_{\tau=1}^k \lambda^{k-\tau} \mathbf{h}_\tau \mathbf{h}_\tau^\top \right) \\ \mathbf{P}_k^{-1} &= \lambda \mathbf{P}_{k-1}^{-1} + \mathbf{h}_k \mathbf{h}_k^\top\end{aligned}\quad (5.6)$$

On développe ensuite $\mathbf{P}_k^{-1} \theta_k$, ce qui nous permet d'obtenir une relation récursive en fonction du dernier estimé, θ_{k-1} :

$$\begin{aligned}\mathbf{P}_k^{-1} \theta_k &= \left(\sum_{\tau=1}^k \lambda^{k-\tau} \mathbf{h}_\tau z_\tau \right) \\ &= \lambda \sum_{\tau=1}^{k-1} \lambda^{k-1-\tau} \mathbf{h}_\tau z_\tau + \mathbf{h}_k z_k \\ &= \lambda \mathbf{P}_{k-1}^{-1} \theta_{k-1} + \mathbf{h}_k z_k\end{aligned}$$

En isolant θ_k , on obtient l'algorithme de moindres carrés récursifs avec facteur d'oubli pour la i^e colonne de la dynamique du système au temps k :

$$\begin{aligned}\theta_k &= \mathbf{P}_k \left(\mathbf{P}_k^{-1} - \mathbf{h}_k \mathbf{h}_k^\top \right) \theta_{k-1} + \mathbf{P}_k \mathbf{h}_k z_k \\ &= \theta_{k-1} + \mathbf{P}_k \mathbf{h}_k (z_k - \mathbf{h}_k^\top \theta_{k-1}) \\ &= \theta_{k-1} + \mathbf{P}_k \mathbf{h}_k \epsilon_k\end{aligned}\quad (5.7)$$

avec ϵ_k valant l'erreur entre la mesure et le nouvel estimé de l'état au temps k . On remarque que l'équation de mise à jour de la covariance (eq. 5.6) utilise \mathbf{P}_{k-1}^{-1} , soit l'inverse de la matrice de la covariance, alors que l'équation de mise à jour de l'estimé (eq. 5.7) utilise directement \mathbf{P}_k . Cela veut aussi dire que le calcul de l'estimé utilise la covariance *a posteriori*³, soit que la propagation de la covariance doit être calculé avant θ_k . Afin d'éviter d'inverser cette matrice à chaque pas de temps, le lemme d'inversion matricielle est utilisé⁴ :

$$\begin{aligned}\mathbf{P}_k &= \lambda^{-1} \left(\mathbf{P}_{k-1}^{-1} + \mathbf{h}_k \lambda^{-1} \mathbf{h}_k^\top \right)^{-1} \\ \mathbf{P}_k &= \lambda^{-1} \left(\mathbf{P}_{k-1} - \mathbf{P}_{k-1} \mathbf{h}_k (\lambda + \mathbf{h}_k^\top \mathbf{P}_{k-1} \mathbf{h}_k)^{-1} \mathbf{h}_k^\top \mathbf{P}_{k-1} \right)\end{aligned}\quad (5.8)$$

2. Pour m signaux de moyenne nulle compris dans le vecteur \mathbf{x} , alors sa covariance sur un échantillon de n valeurs équivaut à $\sum_{i=1}^n \mathbf{x}_i \mathbf{x}_i^\top$

3. Il est possible d'estimer θ_k avec la covariance *a priori* en injectant l'équation 5.8 dans 5.7

4. Lemme d'inversion matricielle : $(\mathbf{F} + \mathbf{G}\mathbf{H}\mathbf{K})^{-1} = \mathbf{F}^{-1} - \mathbf{F}^{-1}\mathbf{G}(\mathbf{H}^{-1} + \mathbf{K}\mathbf{F}^{-1}\mathbf{G})^{-1}\mathbf{K}\mathbf{F}^{-1}$

On note le gain \mathbf{K}_k , utilisé lors de la mise à jour de θ_k avec la covariance *a priori* et pour simplifier l'équation de mise à jour de la covariance par l'équation 5.10 comme suit :

$$\mathbf{K}_k = \mathbf{P}_{k-1} \mathbf{h}_k (\lambda + \mathbf{h}_k^\top \mathbf{P}_{k-1} \mathbf{h}_k)^{-1} \quad (5.9)$$

$$\mathbf{P}_k = \lambda^{-1} (\mathbf{I} - \mathbf{K}_k \mathbf{h}_k^\top) \mathbf{P}_{k-1} \quad (5.10)$$

En choisissant judicieusement le facteur d'oubli et en implémentant l'algorithme pour les n états du système, il est possible de reconstruire le système. Dans un contexte où l'entrée du système varie suffisamment, cet algorithme est adéquat. Dans un autre cas, certaines méthodes doivent être mises en place pour garantir la convergence de l'algorithme.

5.2.2 Limitations sur la matrice de covariance

La convergence des moindres carrés récursifs avec facteur d'oubli est un algorithme dont la stabilité a été démontrée en [36],[21] et converge exponentiellement vers les paramètres dynamiques du système. Ceci est vrai lorsqu'il y a une excitation persistante dans l'entrée du système. Si tel est le cas, la matrice de covariance reste finie (l'inverse de la covariance ne tend pas vers zéro). Dans le cas contraire, les éléments de la matrice de covariance peuvent augmenter en flèche et éventuellement faire dépasser la limite physique de son conteneur (ex. un nombre point flottant, valant maximum 3.4×10^{38}) lors du calcul de la covariance *a posteriori*. Il est aisé de voir que de l'équation 5.6, lorsque les signaux contenus dans \mathbf{h}_n sont de faible amplitude et λ inférieur à 1, \mathbf{P}_n^{-1} va décroître exponentiellement et éventuellement atteindre zéro. Pour contrer ce phénomène, il existe plusieurs méthodes pour limiter la matrice de covariance (sa norme). Le plus simple est la réinitialisation de la covariance [9] [34], où la matrice \mathbf{P} est changée pour la matrice initiale \mathbf{P}_0 lorsque sa norme dépasse un certain seuil.

$$\mathbf{P}_n = \begin{cases} \lambda^{-1} (\mathbf{I} - \mathbf{K}_n \mathbf{h}_n^\top) \mathbf{P}_{n-1} & \text{si } \|\mathbf{P}_{n-1}\|_2 \leq P_{max} \\ \mathbf{P}_0 & \text{sinon} \end{cases} \quad (5.11)$$

D'autres méthodes impliquent un algorithme de projection sur la covariance seulement lorsque $\hat{\theta}$ est contenu dans un ensemble convexe fermé \mathcal{S} [34]. Enfin, l'algorithme de moindres carrés récursifs stabilisés (MCRS) [62] [14] rajoute une borne inférieure sur la matrice \mathbf{P}_n^{-1} (\mathbf{P} est semi-définie positive), l'empêchant d'atteindre zéro.

$$\mathbf{P}_n^{-1} = \lambda \mathbf{P}_{n-1}^{-1} + \mathbf{h}_n \mathbf{h}_n^\top + \delta(1 - \lambda) \mathbf{I} \quad (5.12)$$

Le faible élément δ dans 5.12 crée deux dynamiques dans la mise à jour de \mathbf{P}_n , soit avec ou sans persistance d'excitation (PE). Avec une entrée suffisamment excitante, l'élément δ est négligeable et l'algorithme se comporte comme précédemment. À l'inverse, sans PE, \mathbf{P}_n^{-1} va tendre vers zéro et ses valeurs propres tendront au plus bas vers $\delta + i0$.

$$\begin{aligned}\mathbf{P}_n^{-1} &= \lambda \mathbf{P}_{n-1}^{-1} + \mathbf{h}_n^\top \mathbf{h}_n + \delta(1 - \lambda)\mathbf{I} \\ \lim_{n \rightarrow \infty} \mathbf{P}_n^{-1} &= \mathbf{P}_{n-1}^{-1} \rightarrow \mathbf{P}_n^{-1} = \delta \mathbf{I}\end{aligned}$$

À noter qu'en modifiant l'équation de la covariance avec $\delta(1 - \lambda)\mathbf{I}$, la fonction de coût représentant l'algorithme de moindres carrés en est aussi modifiée, ce qui apporte l'équation de mise à jour de l'estimé (eq. 5.7) à :

$$\hat{\theta}_n = \theta_{n-1} + \mathbf{P}_n \mathbf{h}_n (z_n - \mathbf{h}_n^\top \theta_{n-1}) + \mathbf{P}_n \delta \lambda (\theta_{n-1} - \theta_{n-2})$$

L'algorithme des MCRS est développé en annexe B.

Variation de l'algorithme MCRS

L'algorithme de mise à jour de la covariance 5.12 ne peut être calculé via le lemme d'inversion matricielle. Les travaux de Oudin [62] et Bodson [14] montrent un algorithme permettant d'approcher l'inversion matricielle d'ordre n par la réduction de n inversions de matrices d'ordre 2 (ce qui réduit la complexité de $\mathcal{O}(n^{2.81})$ à $\mathcal{O}(mn)$, où m est le nombre d'opération pour l'inversion d'une matrice d'ordre 2)⁵.

L'idée est d'incorporer le facteur $\delta(1 - \lambda)$ colonne par colonne pour qu'en *moyenne* le calcul de la covariance se comporte comme si le facteur $\delta(1 - \lambda)$ était ajouté à chaque fois, tout en réduisant la complexité à l'inversion d'une matrice 2×2 . En partant de l'équation 5.12 de la covariance et en utilisant le lemme d'inversion matricielle [14] :

$$\mathbf{P}_k = (\mathbf{B}^{-1} + \mathbf{C}\mathbf{C}^\top)^{-1} = \mathbf{B} - \mathbf{B}\mathbf{C}(\mathbf{I} + \mathbf{C}^\top\mathbf{B}\mathbf{C})^{-1}\mathbf{C}^\top\mathbf{B}$$

avec

$$\begin{aligned}\mathbf{B} &= \lambda^{-1} \mathbf{P}_{k-1} \\ \mathbf{C} &= \begin{bmatrix} \mathbf{h}_k & \sqrt{\delta(1 - \lambda)} \mathbf{e}_i \end{bmatrix}\end{aligned}$$

5. $\mathcal{O}(n^{2.81})$ est obtenu avec un algorithme *LUP*, soit *LU* avec pivot, voir [1]

où \mathbf{e}_i est la i^e colonne de la matrice identité \mathbf{I}_n . À chaque calcul de \mathbf{P} , une colonne différente de \mathbf{I}_n est utilisée, représenté par le *modulo* du temps écoulé k par l'ordre n du système :

$$\mathbf{e}_i[k] = \mathbf{I}(:, k \bmod n)$$

Après une itérations, l'inverse de la covariance serait donc :

$$\begin{aligned} \mathbf{P}_k^{-1} &= \lambda \mathbf{P}_{k-1}^{-1} + \mathbf{h}_k \mathbf{h}_k^\top + \delta(1 - \lambda) \mathbf{e}_i \mathbf{e}_i^\top \\ \mathbf{P}_{k+1}^{-1} &= \lambda(\lambda \mathbf{P}_{k-1}^{-1} + \mathbf{h}_k \mathbf{h}_k^\top + \delta(1 - \lambda) \mathbf{e}_i \mathbf{e}_i^\top) + \mathbf{h}_{k+1} \mathbf{h}_{k+1}^\top + \delta(1 - \lambda) \mathbf{e}_{i+1} \mathbf{e}_{i+1}^\top \end{aligned}$$

Malheureusement, avec cette implémentation, des oscillations sur la covariance \mathbf{P} de fréquence $\frac{1}{(n+m)T_s}$ peuvent être présentes si le système est instable et très couplé. L'ajout ligne par ligne de la borne inférieure $\sqrt{\delta(1-\lambda)}$ n'empêche pas les lignes *individuelles* de \mathbf{P}^{-1} de tendre vers zéro en l'absence d'excitation, ce qui déstabiliserait l'estimation à partir du moment où l'un des pôles de \mathbf{P}_k^{-1} frôle zéro. Cette méthode ne sera donc pas utilisée ici.

5.2.3 Facteur d'oubli variable

Dans le cas d'une excitation qui n'est pas constante à l'entrée du système, un facteur d'oubli variable permet d'ajuster l'algorithme pour que la *quantité d'information* dans l'entrée reste environ constante. Comme le présente Goodwin [28], il existe un analogue entre un facteur d'oubli variable et les différentes stratégies de restrictions de la matrice de covariance (ex. forcer λ à zéro lors d'une estimation indésirable de θ revient utiliser un algorithme avec projection de $\hat{\theta}$ sur \mathcal{S}). Puisque le facteur d'oubli influence la vitesse de convergence de l'estimé, un facteur d'oubli très faible convergera rapidement mais contiendra d'énormes discontinuités [27], alors qu'un facteur très proche de 1 créera une convergence lente et monotone de l'estimé (donc plus robuste) [32]. L'utilisation d'un facteur d'oubli variable vient chercher un compromis de ces deux comportements, car l'utilisation d'un facteur d'oubli très faible est seulement pertinente à des instants de très grande excitation du système (en d'autres mots, lors d'une grande excitation, le système change de comportement, ce qui implique que le nombre d'échantillons représentant plus fidèlement sa dynamiques est réduit aux derniers mesurés). Lorsqu'il y a beaucoup de variation sur l'entrée du système, on considère qu'il y a une grande quantité d'*information* entrante. L'information est ici quantifiée en terme de son contenu fréquentiel, ou plus simplement de son énergie en tout instant t dans une fenêtre T_0 [34] :

$$\alpha_0 \mathbf{I} \leq \frac{1}{T_0} \int_t^{t+T_0} \mathbf{h}(\tau) \mathbf{h}^\top(\tau) d\tau \leq \alpha_1 \mathbf{I}, \quad \alpha_0, \alpha_1, T_0 > 0 \quad (5.13)$$

où α_0 définit le *niveau d'excitation*. Lorsque l'équation 5.13 est respectée, alors le signal \mathbf{h} est PE. L'idée du facteur d'oubli variable est donc d'avoir λ bas lorsque la variation dans \mathbf{h}_n est élevée, et de garder λ au maximum lorsqu'il y a peu d'excitation en entrée, pour réduire au maximum la croissance de $\|\mathbf{P}\|_2$. En d'autres mots, un facteur d'oubli variable est robuste à un changement soudain de la commande. Bien qu'un facteur d'oubli variable peut maintenir la norme de la covariance basse, il ne la borne pas au sens mathématique du terme [27]. C'est pourquoi il est intéressant d'utiliser la formulation des moindres carrés récurrents stabilisés avec une borne très élevée (ex. 10^7) et un facteur d'oubli variable. De cette façon, la dérivée de l'estimation reste bornée et les coefficients inconnus convergent aussi vite que le meilleur facteur d'oubli calculé. Il existe quelques implémentations possibles d'un facteur d'oubli variable, ici nous reprenons celle de Fortescue, de par sa simplicité et son faible coût de calcul. Pour partager le facteur d'oubli avec tous les états estimés, on utilise d'abord le produit scalaire de l'erreur d'estimation :

$$\begin{aligned}\mathbf{e}_k &= \mathbf{z}_k - \begin{bmatrix} \hat{\mathbf{F}}_k & \hat{\mathbf{G}}_k \end{bmatrix} \mathbf{h}_k \\ \epsilon_2 &= \mathbf{e}_k^\top \mathbf{e}_k\end{aligned}$$

Un calcul préliminaire λ' du facteur d'oubli est effectué en se basant sur la mesure \mathbf{h}_k et l'erreur d'approximation ϵ_2 :

$$\begin{aligned}\mathbf{K}_k &= \mathbf{P}_{k-1} \mathbf{h} \left(1 + \mathbf{h}^\top \mathbf{P}_{k-1} \mathbf{h}\right)^{-1} \\ \lambda'[k] &= 1 - \frac{(1 - \mathbf{h}_k^\top \mathbf{K}_k) \epsilon_2^2}{\Sigma_0}\end{aligned}$$

avec $\Sigma_0 = \sigma_v^2 N_0$ représentant la sensibilité du facteur d'oubli et \mathbf{K}_k le gain de mise à jour de la covariance (\mathbf{K}_k), sans considérer le facteur d'oubli (équation 5.9 légèrement modifiée). Le paramètre Σ_0 est calculé selon la variance du bruit de mesure σ_v et le paramètre N_0 équivaut approximativement au nombre d'échantillons précédent à utiliser, soit la vitesse d'adaptation de l'estimateur. Un facteur de sensibilité élevé fera donc fluctuer la valeur de λ plus facilement. Ici une valeur de $\Sigma_0 = 0.0025$ a été utilisée. Afin d'éviter des valeurs extrêmes, λ est borné par une valeur supérieure et inférieure :

$$\lambda[k] = \begin{cases} \lambda_{min}, & \lambda' < \lambda_{min} \\ \lambda_{max}, & \lambda' > \lambda_{max} \\ \lambda', & \text{sinon} \end{cases} \quad (5.14)$$

5.3 Résultats d'estimation

Pour vérifier la convergence de l'estimateur, ce dernier est d'abord testé sur le modèle linéarisé du drone avec le contrôleur y étant associé. Le modèle représente le drone faisant un vol stationnaire avec une charge et la moitié de son réservoir d'essence, et le contrôleur utilise les gains d'un retour de sortie trouvés précédemment par synthèse H_∞ (contrôle Proportionnel, Intégral et Dérivée sur les signaux régulés). On teste en premier lieu l'estimateur de moindres carrés récursifs sans restriction sur sa covariance, et pour assurer un minimum d'excitation du système une onde carrée filtrée est envoyée comme commande en roulis pour le système. Pour ne pas biaiser l'estimé, une matrice initiale unitaire pour \mathbf{F} et une matrice nulle pour \mathbf{G} sont utilisés.

L'estimateur est d'abord simulé dans des conditions ne tenant pas en compte la persistance d'excitation du système afin de souligner l'instabilité numérique possible lorsqu'aucune limitation sur l'algorithme n'est appliqué (Fig. 5.3). Un petit facteur d'oubli ($\lambda = 0.99$) permet en 90 s de montrer la convergence de l'estimateur ainsi que son comportement lorsqu'il atteint la limite de la précision machine ($\epsilon_m = 2^{-52} \approx 2.2 \times 10^{-16}$). En se référant à la définition de la persistance d'excitation (Eq. 5.13), le facteur d'oubli donne approximativement une fenêtre d'excitation T_0 de 23 s, si l'on considère la fin de la fenêtre à 10% de la valeur du signal :

$$T_0 = T_s \log_\lambda(0.10) \approx 23 \text{ s}$$

confirmant qu'il y a absence d'excitation en au moins trois points dans la figure 5.3 (la demi-période de l'onde carrée de référence dépasse T_0). On voit néanmoins que la norme de \mathbf{P}^{-1} ainsi que l'erreur des états diminuent à un rythme exponentiel ; ce test justifie l'ajout d'une restriction sur \mathbf{P} .

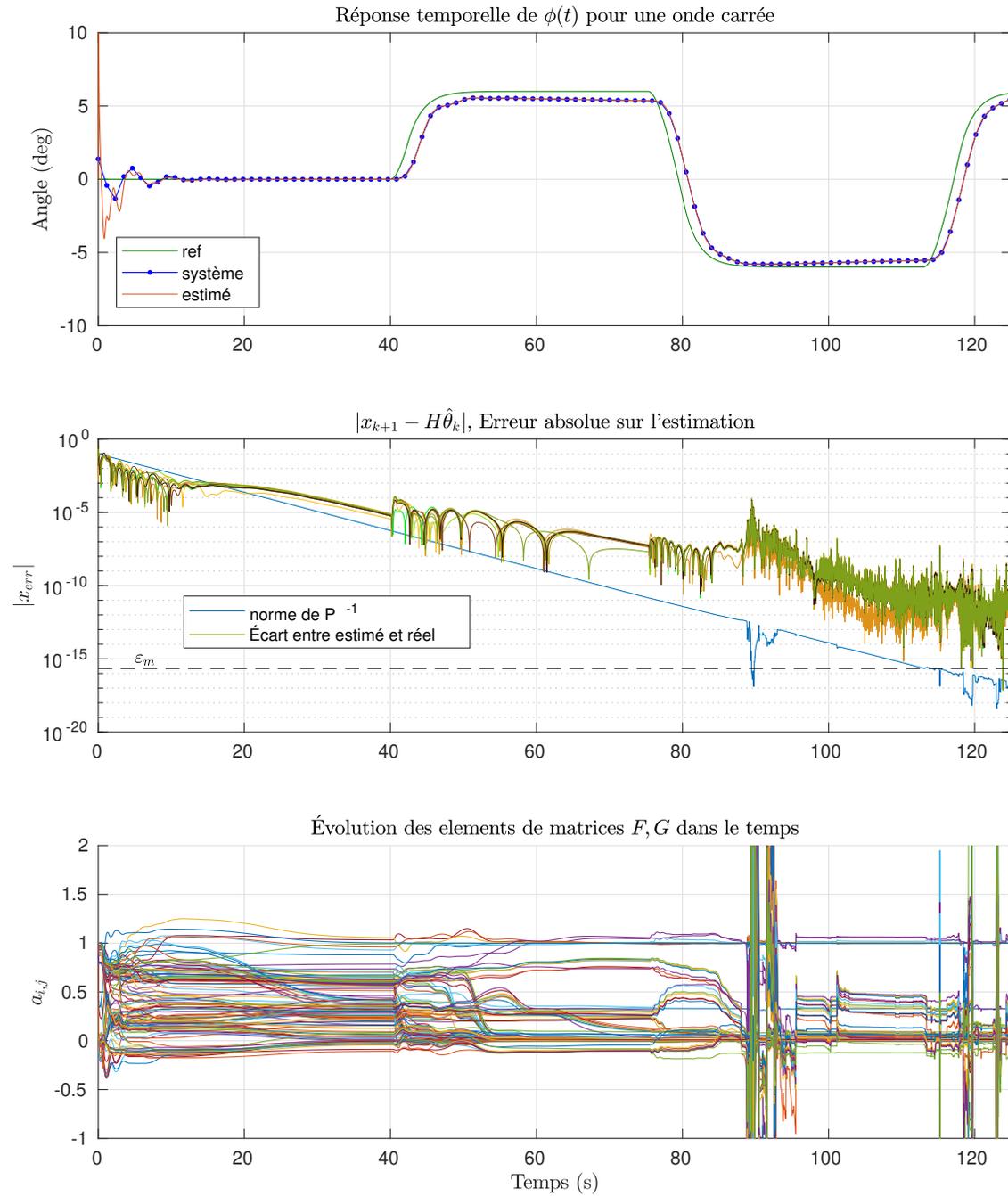


Figure 5.3 Estimation du modèle linéaire sans restriction sur \mathbf{P} , avec $\lambda = 0.99$

Pour les mêmes conditions de test, on simule ensuite l'estimateur utilisant une réinitialisation de la covariance (Fig. 5.4) et le MCRS (Fig. 5.5), montrant le comportement de ces derniers en l'absence d'excitation.

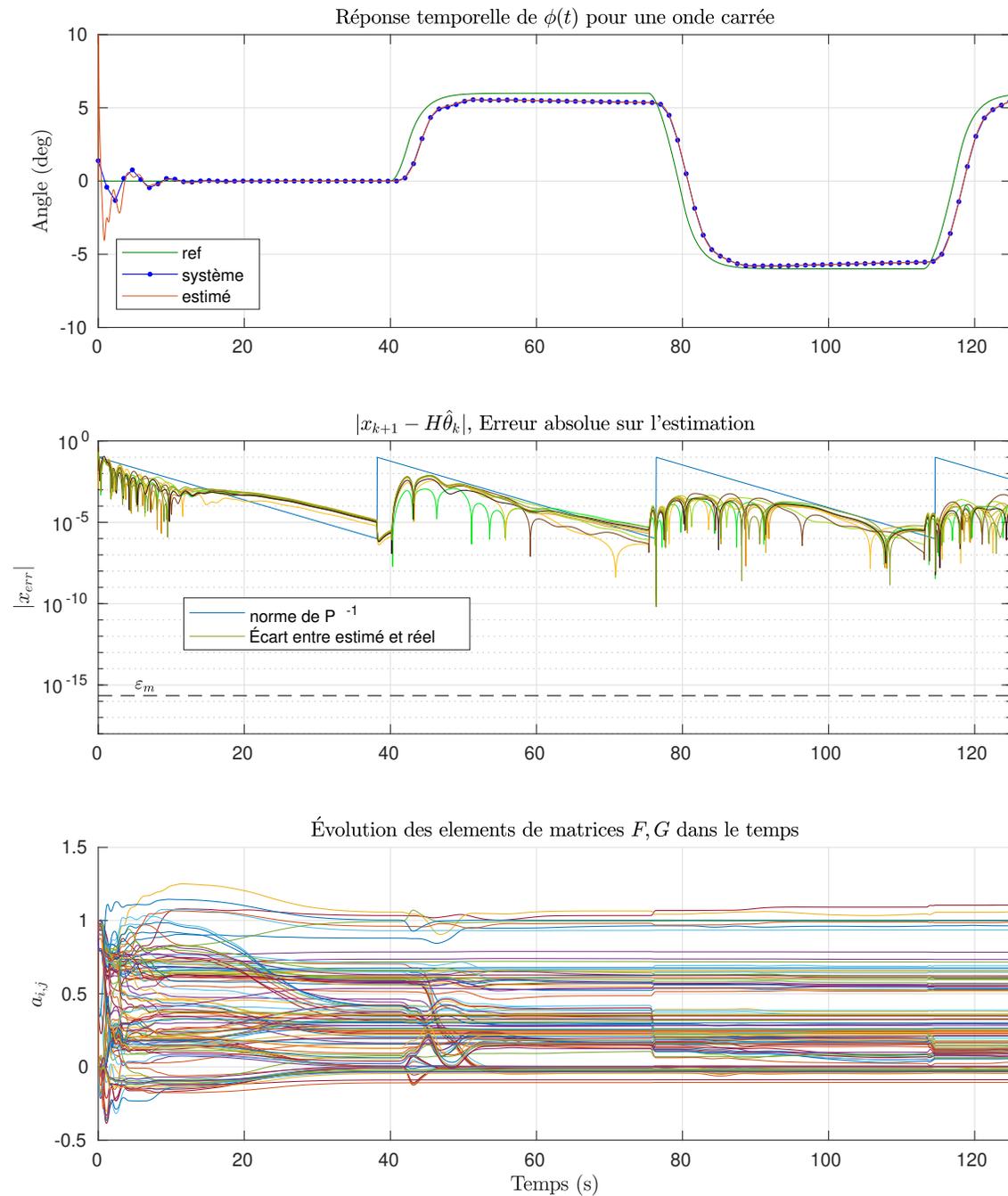


Figure 5.4 Estimation du modèle linéaire avec réinitialisation de la covariance et $\lambda = 0.99$

On peut voir sur la figure 5.4 qu'avec l'utilisation de l'algorithme de réinitialisation de la covariance il n'y a plus de perturbations sur l'évolution des dynamiques des états, mais aussi que l'évolution de l'erreur des états est principalement gouvernée par l'évolution de la covariance. Dans le cas où l'excitation du système n'est plus présente, la norme de \mathbf{P}^{-1} s'emballerait donc davantage, pour finir par osciller entre son maximum et son minimum. Cet algorithme offre l'avantage d'être simple

numériquement, mais la réinitialisation abrupte de la matrice P_k pourrait poser problème pour un système sensible tel le modèle non linéaire à contrôler.

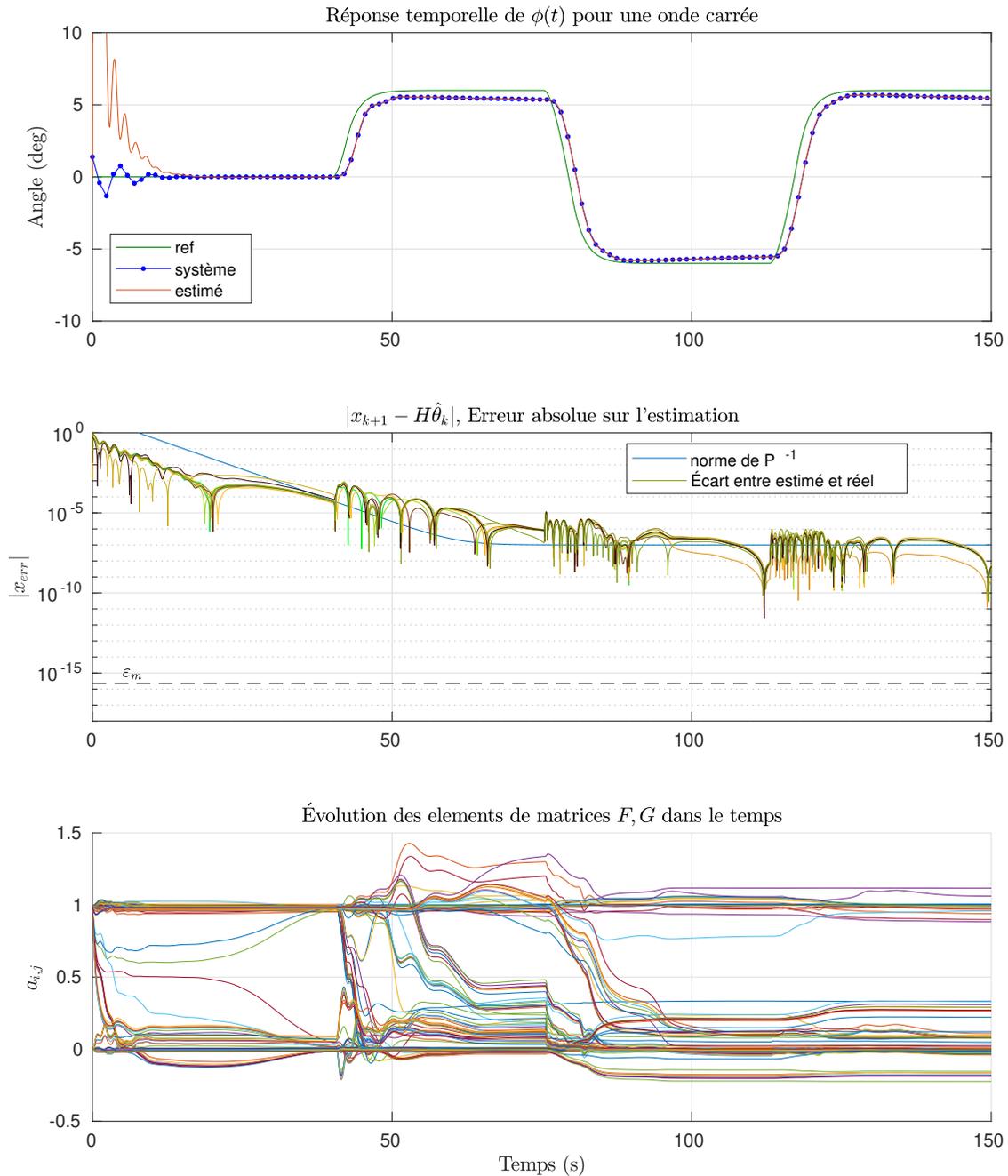


Figure 5.5 Estimation du modèle linéaire avec limite sur P^{-1} et $\lambda = 0.99$

La figure 5.5 présente l'utilisation des moindres carrés récurrents stabilisés avec facteur d'oubli fixe. On remarque qu'entre 0 et 50 s, l'évolution est similaire à l'algorithme de réinitialisation de la covariance, mais qu'ensuite la dynamique de $\delta(1 - \lambda)$ devient significatif et maintient l'évolution

de l'erreur sur x_k stable. L'évolution des éléments de \hat{F} et \hat{G} demeure lente et justifie l'ajout d'un facteur d'oubli variable.

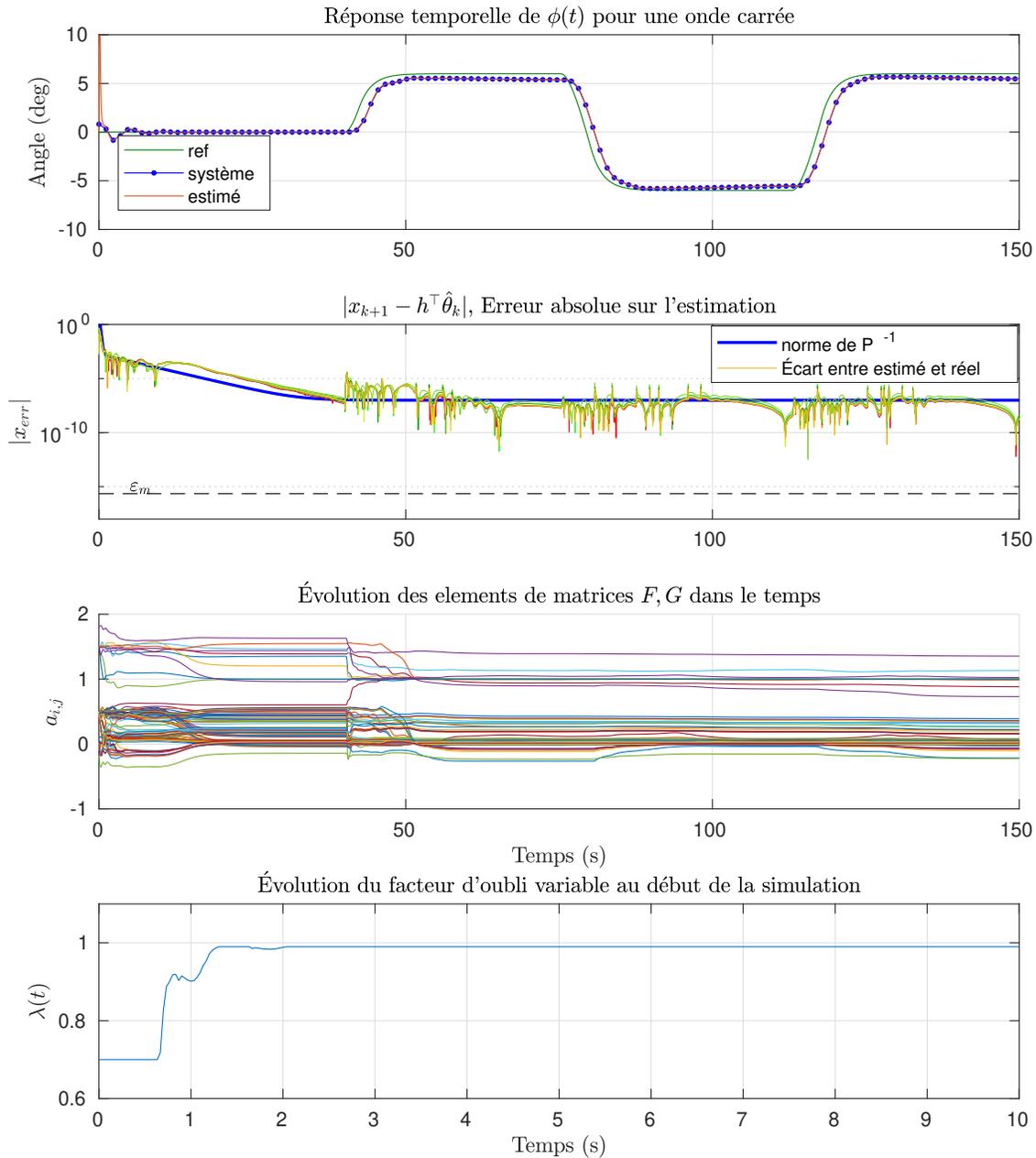


Figure 5.6 Estimation du modèle linéaire avec limite sur P^{-1} et facteur d'oubli variable

L'utilisation d'un facteur d'oubli variable est ensuite simulé, avec $0.7 \leq \lambda \leq 0.99$, permettant une fenêtre d'excitation aussi petite que 0.6 s, ce qui permettra de corriger rapidement l'estimé initial.

Le premier graphe de la figure 5.6 montre la nette amélioration de l'estimation (rouge) par l'utilisation de l'équation 5.14 pour la mise à jour de λ . Le quatrième graphe montre que pour les premiers

instants, le facteur d'oubli effectif est à son minimum de par le grand écart entre l'estimé et le système, puis se stabilise rapidement vers sa valeur maximale allouée.

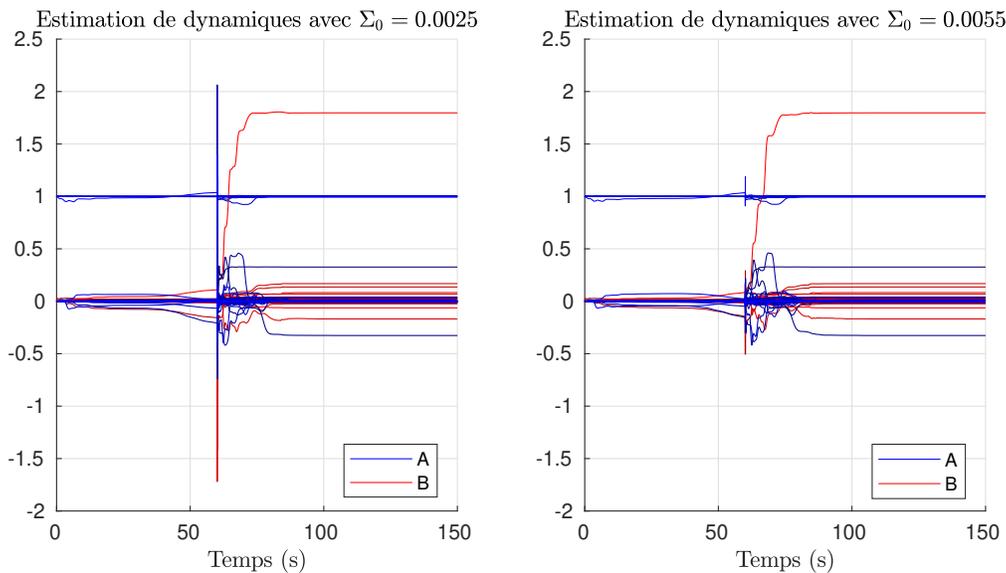


Figure 5.7 Effet du paramètre Σ_0 sur la stabilité de l'estimation du sys. linéaire

Parmi les paramètres à régler, le facteur de sensibilité du facteur d'oubli variable varie grandement les discontinuités dans la convergence des inconnus du système. De façon similaire à un facteur d'oubli plus faible, une faible sensibilité créera de larges dépassements sur les nouveaux estimés, ce qui est à éviter si ces paramètres sont rebouclés avec un contrôleur adaptatif. On peut voir sur la figure 5.7 le déroulement de l'estimation des dynamiques du système linéarisé. À gauche, pour un faible Σ_0 , un grand dépassement se produit lorsque l'onde carrée de référence en roulis commence (lorsque le système commence à être excité). On peut tout de même voir que pour une seconde valeur plus large de Σ_0 , la convergence est sensiblement la même (le temps où le facteur d'oubli n'est pas à son maximum est très faible).

Par la suite, les tests sur le système non linéaire montrent que la vitesse de convergence est similaire aux tests sur le modèle linéaire, et que la convergence n'est pas affectée par la restriction sur la norme de P.

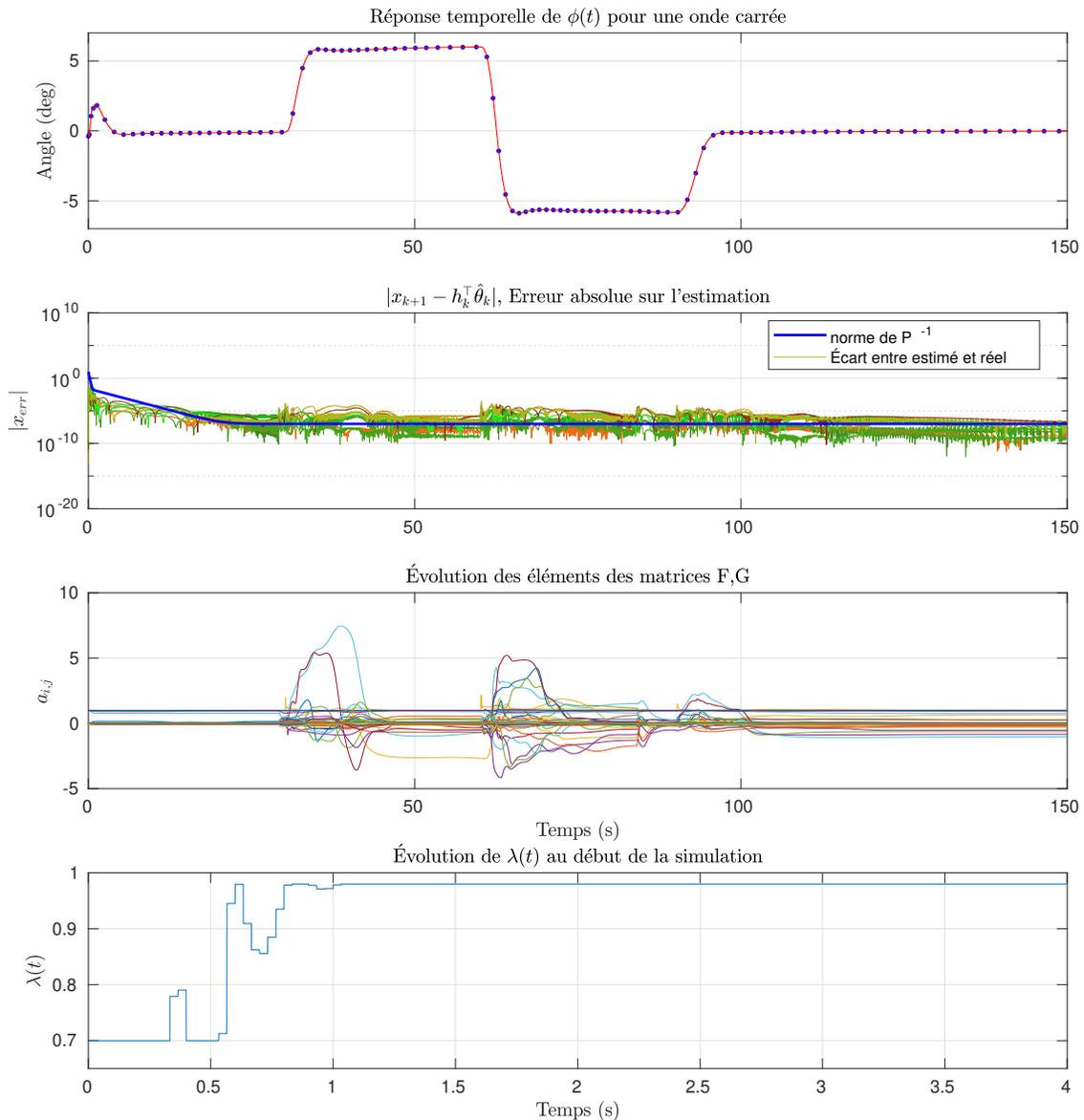


Figure 5.8 Estimation des états du modèle non-linéaire avec restriction sur P et FF variable

Il est à noter qu'un facteur d'oubli maximal de 0.99 a été utilisé dans les précédents tests pour souligner la relation de l'algorithme de l'estimateur avec le degré d'excitation en son entrée. Pour la suite, un facteur de 0.999 sera plutôt utilisé, puisque la constante de temps la plus lente du modèle linéarisé est d'environ 20 s et que l'on veut minimiser au maximum les fluctuations sur l'estimé lors de la mise à jour du gain du contrôleur.

5.4 Calcul du nouveau gain

Connaissant la dynamique du drone, il est alors possible de recalculer le gain du contrôleur pour placer les pôles du système en boucle fermée à l'emplacement du comportement idéal du drone. Ici, ce système correspond au linéarisé tangent du drone lorsqu'il ne contient pas de charge. Le modèle de référence est donné comme suit :

$$\begin{aligned}\mathbf{x}_m[k+1] &= \mathbf{F}_m \mathbf{x}_m[k] + \mathbf{G}_m \mathbf{u}[k] \\ \mathbf{u}[k] &= \mathbf{K}_x \mathbf{x}_m[k]\end{aligned}$$

où les matrices \mathbf{F}_m et \mathbf{G}_m sont connues et contiennent les pôles désirés du système. Le gain \mathbf{K}_x contient les gains *Proportionnel* et *Dérivée* qui seront aussi utilisé par le modèle réel lors de l'estimation. L'idée du nouveau gain à calculer est de trouver un nouveau \mathbf{K}_x plaçant les pôles du système vers ceux du modèle de référence en boucle fermée. Pour un contrôleur utilisant l'erreur avec un signal de référence, le système devient :

$$\begin{aligned}\mathbf{u}[k] &= \mathbf{K}_x (\mathbf{r}[k] - \mathbf{x}_m[k]) \\ \mathbf{x}_m[k+1] &= (\mathbf{F}_m - \mathbf{G}_m \mathbf{K}_x) \mathbf{x}_m[k] + \mathbf{G}_m \mathbf{K}_x \mathbf{r}[k]\end{aligned}\quad (5.15)$$

Avec un PID, on augmente en fait l'ordre du système avec quatre intégrateurs ($\phi_i, \theta_i, \psi_i, w_i$). En utilisant la définition d'un intégrateur discret *forward Euler* [53], la dynamique de ϕ_i est donnée comme suit :

$$\begin{aligned}\phi_i[k] &= \frac{T_s(r_\phi[k] - \phi[k])}{z - 1} \\ x_{1i}[k+1] &= x_{1i}[k] + T_s(r_1[k] - x_1[k])\end{aligned}$$

ce qui donne le modèle de référence augmenté suivant :

$$\begin{aligned}\begin{bmatrix} \mathbf{x}_m[k+1] \\ \mathbf{x}_{im}[k+1] \end{bmatrix} &= \begin{bmatrix} \mathbf{F}_m & \mathbf{0} \\ -T_s \mathbf{E} & \mathbf{I} \end{bmatrix} \begin{bmatrix} \mathbf{x}_m[k] \\ \mathbf{x}_{im}[k] \end{bmatrix} + \begin{bmatrix} \mathbf{G}_m \\ \mathbf{0} \end{bmatrix} \begin{bmatrix} \mathbf{K}_x & -\mathbf{K}_i \end{bmatrix} \left(\begin{bmatrix} \mathbf{r}[k] \\ \mathbf{0} \end{bmatrix} - \begin{bmatrix} \mathbf{x}_m[k] \\ \mathbf{x}_{im}[k] \end{bmatrix} \right) \\ \mathbf{E} &= \mathbf{I}([1 \ 2 \ 3 \ 9], :)\end{aligned}$$

où \mathbf{E} contient les quatre lignes de la matrice identité correspondant aux quatre intégrateurs (puisque l'on considère ici un retour d'états). À titre comparatif, le modèle estimé est plutôt le suivant :

$$\begin{aligned} \begin{bmatrix} \mathbf{x}[k+1] \\ \mathbf{x}_i[k+1] \end{bmatrix} &= \begin{bmatrix} \hat{\mathbf{F}} & \mathbf{0} \\ -T_s \mathbf{E} & \mathbf{I} \end{bmatrix} \begin{bmatrix} \mathbf{x}[k] \\ \mathbf{x}_i[k] \end{bmatrix} + \begin{bmatrix} \hat{\mathbf{G}} \\ \mathbf{0} \end{bmatrix} \begin{bmatrix} \mathbf{K}_x' & -\mathbf{K}_i' \end{bmatrix} \left(\begin{bmatrix} \mathbf{r}[k] \\ \mathbf{0} \end{bmatrix} - \begin{bmatrix} \mathbf{x}[k] \\ \mathbf{x}_i[k] \end{bmatrix} \right) \\ \mathbf{x}_a[k+1] &= \hat{\mathbf{F}}_a \mathbf{x}_a[k] + \hat{\mathbf{G}}_a \mathbf{K}_a' (\mathbf{r}_a[k] - \mathbf{x}_a[k]) \\ \begin{bmatrix} \mathbf{x}[k+1] \\ \mathbf{x}_i[k+1] \end{bmatrix} &= \begin{bmatrix} \hat{\mathbf{F}} - \hat{\mathbf{G}} \mathbf{K}_x' & \hat{\mathbf{G}} \mathbf{K}_i' \\ -T_s \mathbf{E} & \mathbf{I} \end{bmatrix} \begin{bmatrix} \mathbf{x}[k] \\ \mathbf{x}_i[k] \end{bmatrix} \end{aligned}$$

où les matrices \mathbf{K}_x' , \mathbf{K}_i' valent respectivement \mathbf{K}_x , \mathbf{K}_i avant l'adaptation, et l'indice a dénote le modèle augmenté. En émettant l'hypothèse que $\hat{\mathbf{F}}$ et $\hat{\mathbf{G}}$ ont été estimées correctement, le contrôleur adaptatif cherche à ce que le système ait la même matrice de transfert en boucle fermée pour \mathbf{x} que le modèle de référence, soit :

$$\begin{aligned} \hat{\mathbf{F}}_a - \hat{\mathbf{G}}_a \mathbf{K}_a' &= \mathbf{F}_{m,a} - \mathbf{G}_{m,a} \mathbf{K}_a \\ \hat{\mathbf{G}}_a \mathbf{K}_a' &= \hat{\mathbf{F}}_a - \mathbf{F}_{m,a} + \mathbf{G}_{m,a} \mathbf{K}_a \end{aligned} \quad (5.16)$$

ce qui amènera les états à tendre vers \mathbf{x}_m lorsque l'égalité est respectée. Puisque la matrice $\hat{\mathbf{G}}_a$ n'est pas carrée, Un moyen direct d'isoler le nouveau gain \mathbf{K}_a' à utiliser est de relier avec un facteur Λ :

$$\begin{aligned} \mathbf{K}_a' &= \mathbf{K}_a \Lambda_a \\ \Lambda_a &= (\hat{\mathbf{G}}_a \mathbf{K}_a)^{-1} \hat{\mathbf{F}}_a - \mathbf{F}_{m,a} + \mathbf{G}_{m,a} \mathbf{K}_a \end{aligned} \quad (5.17)$$

5.4.1 Utilisation de la pseudoinverse de Moore-Penrose

Le problème avec la méthode de l'équation 5.17 est que l'inversion de la matrice n'est pas stable numériquement. Lorsque le déterminant de la matrice à inverser frôle zéro, l'algorithme devient invalide. Il existe plusieurs méthodes pour inverser une matrice, telle que par la matrice des cofacteurs, par une élimination de Gauss-Jordan ou par une décomposition LU [20], mais ces dernières calculent la solution exacte (sur une matrice approximée), quitte à générer de grandes valeurs dans le résultat. Il est par exemple possible d'utiliser la méthode des moindres carrés pour obtenir une approximation de l'inverse, soit :

$$\begin{aligned} \mathbf{A} &= \hat{\mathbf{G}} \mathbf{K}_a \\ \mathbf{A}^{-1} &\approx (\mathbf{A}^\top \mathbf{A})^{-1} \mathbf{A}^\top \end{aligned}$$

mais cette méthode n'offre aucun moyen d'éviter une distortion du calcul lorsque $\mathbf{A}^\top \mathbf{A}$ contient des lignes nulles. Une seconde alternative est de calculer une approximation de son inverse, nommée la *pseudoinverse* de Moore-Penrose [10]. L'algorithme effectue la décomposition en valeurs singulières de la matrice (*SVD decomposition*), puis l'inversion se résume en une transposition et n inversions scalaires. Par exemple, pour une matrice $\mathbf{A}_{n \times m}$, sa i^e valeur singulière (σ_i) est définie comme la racine carrée de la i^e valeur propre de $\mathbf{A}^\top \mathbf{A}$:

$$\mathbf{M} = \mathbf{A}^\top \mathbf{A}$$

$$\sigma_i = \sqrt{\lambda_i(\mathbf{M})}$$

et pour chaque valeur singulière on peut lui associer des vecteurs singuliers $\mathbf{u}_i, \mathbf{v}_i$ (voir [38]) :

$$\sigma_i \mathbf{u}_i = \mathbf{A} \mathbf{v}_i, \quad \mathbf{v}_i \text{ le } i^e \text{ vecteur propre unitaire de } \mathbf{M}$$

$$\mathbf{u}_i = \sigma_i^{-1} \mathbf{A} \mathbf{v}_i$$

Les matrices \mathbf{U}, \mathbf{V} peuvent ensuite être construites via les vecteurs singuliers trouvés :

$$\mathbf{U} = \begin{bmatrix} \mathbf{u}_1 & \mathbf{u}_2 & \cdots & \mathbf{u}_m & \mathbf{0}_{n-m} \end{bmatrix}, \quad \mathbf{U} \mathbf{U}^\top = \mathbf{U}^\top \mathbf{U} = \mathbf{I}_n$$

$$\mathbf{V} = \begin{bmatrix} \mathbf{v}_1 & \mathbf{v}_2 & \cdots & \mathbf{v}_m & \mathbf{0}_{n-m} \end{bmatrix}, \quad \mathbf{V} \mathbf{V}^\top = \mathbf{V}^\top \mathbf{V} = \mathbf{I}_m$$

et les valeurs singulières construisent une matrice diagonale \mathbf{D} pour compléter la décomposition de \mathbf{A} :

$$\mathbf{D} = \begin{bmatrix} \sigma_1 & 0 & \cdots & 0 \\ 0 & \sigma_2 & & \\ \vdots & & \ddots & \\ 0 & & & \sigma_m \\ \hline & & & & \mathbf{0}_{(n-m) \times m} \end{bmatrix}_{n \times m}, \quad \sigma_1 > \dots > \sigma_m, \quad n > m$$

$$\mathbf{A} = \mathbf{U} \mathbf{D} \mathbf{V}^\top$$

où les valeurs singulières sont ordonnées par ordre décroissant le long de la diagonale de \mathbf{D} . La décomposition en valeurs singulières permet de retirer les éléments de faible valeur qui, lors de l'inversion d'une matrice, prendraient de très grandes valeurs. Par exemple, cela se ferait en annulant les valeurs propres inférieures à un certain seuil (ici 10^{-10}). Pour une matrice rectangulaire $\mathbf{A}_{n \times m}$ ayant sa dernière valeur singulière nulle, le calcul peut donc s'effectuer sur un système d'ordre

$m - 1$:

$$\begin{aligned} \mathbf{U} &= \begin{bmatrix} \mathbf{U}_1 & \mathbf{0} \end{bmatrix}_{(m-1) \times n} \\ \mathbf{V} &= \begin{bmatrix} \mathbf{V}_1 & \mathbf{0} \end{bmatrix}_{(m-1) \times n} \\ \mathbf{D} &= \begin{bmatrix} \mathbf{D}_1 & \mathbf{0} \\ \mathbf{0} & \mathbf{0} \end{bmatrix}, \quad \mathbf{D}_1 \text{ de dimension } (m-1) \times (m-1) \end{aligned}$$

ce qui réduirait la décomposition à :

$$\mathbf{A} = \mathbf{U}_1 \mathbf{D}_1 \mathbf{V}_1^\top$$

Puisque la matrice \mathbf{D}_1 est diagonale les matrices $\mathbf{U}_1, \mathbf{V}_1$ orthogonales unitaires, l'inversion se réduit à l'inversion des éléments de \mathbf{D}_1 :

$$\begin{aligned} \mathbf{A}^{-1} &= \mathbf{V}_1 \mathbf{D}_1^{-1} \mathbf{U}_1^\top \\ &= \mathbf{V}_1 \begin{bmatrix} \sigma_1^{-1} & 0 & \cdots & 0 \\ 0 & \sigma_2^{-1} & & \\ \vdots & & \ddots & \\ 0 & & & \sigma_{m-1}^{-1} \end{bmatrix} \mathbf{U}_1^\top \end{aligned}$$

Sous *Matlab*, le calcul de la *pseudoinverse* se fait avec la fonction `pinv`. Étant donné que la vitesse de la dynamique du système est beaucoup plus grande que celle des intégrateurs du système augmenté, on peut découpler les éléments de la matrice Λ_a entre les états et les intégrateurs :

$$\Lambda_a = \begin{bmatrix} \Lambda_x & 0 \\ 0 & \Lambda_i \end{bmatrix}$$

$$\mathbf{K}_x' = \mathbf{K}_x \Lambda_x$$

$$\mathbf{K}_i' = \mathbf{K}_i \Lambda_i$$

ce qui permet de trouver deux équations pour Λ_a à partir du modèle en boucle fermée du système, l'une pour les dynamiques des états et l'autre pour le couplage entre les états et les intégrateurs.

$$\begin{aligned} \Lambda_x &= \text{pinv}(\hat{\mathbf{G}}\mathbf{K}_x) (\hat{\mathbf{F}} - \mathbf{F}_m + \mathbf{G}_m \mathbf{K}_x) \\ \Lambda_i &= \text{pinv}(\hat{\mathbf{G}}\mathbf{K}_i) \mathbf{G}_m \mathbf{K}_i \end{aligned} \quad (5.18)$$

5.4.2 Implémentation et résultats

Obtention de \mathbf{K}'_x

Le calcul de la pseudoinverse peut être lourd vu la décomposition nécessaire. Puisque la décomposition par valeurs singulières peut s'effectuer sur une matrice rectangulaire, en modifiant 5.18 on peut directement obtenir \mathbf{K}'_x en n'inversant que $\hat{\mathbf{G}}$:

$$\mathbf{K}'_x = \text{pinv}(\hat{\mathbf{G}}) (\hat{\mathbf{F}} - \mathbf{F}_m + \mathbf{G}_m \mathbf{K}_x) \quad (5.19)$$

En connaissant la matrice \mathbf{K}_x , l'ordre des matrices à utiliser peut aussi être réduit. Par exemple, en utilisant les gains trouvés pour un vol stationnaire avec la synthèse robuste (Fig. 4.15), soit :

$$\mathbf{K}_x = \begin{bmatrix} -1.5516 & 0 & 0 & -1.0645 & 0 & 0 & 0 & 0 & 0 \\ 0 & -1.7345 & 0 & 0 & -0.2333 & 0 & 0 & 0 & 0 \\ 0 & 0 & -8.8482 & 0 & 0 & -12.7728 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1.2416 \end{bmatrix}$$

des simplifications sont possibles. Le gain précédent (pour le vecteur d'états donné en 4.12), montre que l'impact du gain *dérivée* sur la vitesse angulaire Δq est très faible (-0.2333), et que les états $\Delta u, \Delta v$ ne sont pas présents dans la commande. Les dynamiques de ces trois états peuvent donc être retirées des matrices de l'équation 5.19.

$$\begin{aligned} \boldsymbol{\xi} &= [\Delta\phi \quad \Delta\theta \quad \Delta\psi \quad \Delta p \quad \Delta r \quad \Delta w] \\ \mathbf{K}'_\xi &= \text{pinv}(\hat{\mathbf{G}}_\xi) (\hat{\mathbf{F}}_\xi - \mathbf{F}_{m\xi} + \mathbf{G}_{m\xi} \mathbf{K}_\xi) \end{aligned} \quad (5.20)$$

Dans le même ordre d'idées, le calcul de \mathbf{K}'_i est négligé, de par la dynamique très lente des états des intégrateurs et que les principaux pôles en boucle fermée sont dictés par l'équation 5.15 (la synthèse H_∞ a justement imposé de petits gains k_i pour minimiser ce couplage).

Calcul de \mathbf{K}'_x en maintenant sa structure particulière

Puisque le gain \mathbf{K}_x n'est constitué que d'un élément par colonne, ceux-ci peuvent être déterminés individuellement par moindres carrés ou par l'utilisation de la pseudo-inverse par décomposition en valeurs singulières. À partir de l'équation 5.16, on obtient le développement suivant en négligeant

les pôles des intégrateurs :

$$\begin{aligned}\hat{\mathbf{G}}\mathbf{K}'_x &= \hat{\mathbf{F}} - \mathbf{F}_m + \mathbf{G}_m\mathbf{K}_x \\ &= \left[\hat{\mathbf{g}}_1 K'_{p,\phi} \quad \hat{\mathbf{g}}_2 K'_{p,\theta} \quad \hat{\mathbf{g}}_3 K'_{p,\psi} \quad \hat{\mathbf{g}}_1 K'_{d,\phi} \quad \hat{\mathbf{g}}_2 K'_{d,\theta} \quad \hat{\mathbf{g}}_3 K'_{d,\psi} \quad \hat{\mathbf{g}}_4 K'_{p,w} \right]\end{aligned}$$

où chaque élément $\hat{\mathbf{g}}_i$ représente la i^e colonne de $\hat{\mathbf{G}}$. L'estimation de chaque nouveau gain K'_\sim peut ensuite être solution de l'équation paramétrique suivante :

$$\hat{\mathbf{g}}_i K'_\sim = \left(\hat{\mathbf{F}} - \mathbf{F}_m + \mathbf{G}_m\mathbf{K}_x \right)_j$$

avec l'indice j représentant la j^e colonne de la matrice $\left(\hat{\mathbf{F}} - \mathbf{F}_m + \mathbf{G}_m\mathbf{K}_x \right)$. Le nouveau gain peut ensuite se calculer par moindres carrés, ou par l'utilisation de la pseudoinverse, comme suit :

$$\begin{aligned}\hat{K}'_\sim &= \left(\hat{\mathbf{g}}_i^\top \hat{\mathbf{g}}_i \right)^{-1} \hat{\mathbf{g}}_i^\top \left(\hat{\mathbf{F}} - \mathbf{F}_m + \mathbf{G}_m\mathbf{K}_x \right)_j \\ &= \text{pinv}(\hat{\mathbf{g}}_i) \left(\hat{\mathbf{F}} - \mathbf{F}_m + \mathbf{G}_m\mathbf{K}_x \right)_j\end{aligned}\quad (5.21)$$

De cette façon, seul les éléments non nuls de \mathbf{K}_x seront modifiés par l'algorithme de placement de pôles adaptatif.

Résultat de simulation

Lors de la simulation du système, le modèle est d'abord estimé, puis le nouveau gain est injecté dans le contrôleur. Pour éviter des discontinuités dans la commande pouvant exciter l'estimateur, la transition des gains est ralentie avec un filtre passe-bas. Il est important de voir ici que ce sont les *gains* qui sont filtrés, et non les *entrées* résultantes, pour que seulement la transition vers le contrôleur adaptatif soit gérée par le filtre. Dans le second cas, le filtre travaillerait contre le contrôleur adaptatif, ce qui annulerait son action⁶. Le filtre utilisé est un passe-bas d'ordre 2 sans dépassement et ayant une fréquence de coupure en 1.76 rad/s, soit environs $10T_s$:

$$k'_i = \frac{0.003278}{z^2 - 1.9z + 0.9033} k_i, \quad k_i \in \mathbf{K}_x$$

Le contrôleur est ensuite testé sur le modèle linéaire pour valider la stabilité et la modification du placement de pôles effectuée. Les figures suivantes présentent les signaux mesurés des états $\phi, \theta, \psi, p, q, r$ avant et après la modification du gain par \mathbf{K}_x' à $t = 175$ s. Le calcul de mise à jour du gain s'est fait avec l'équation 5.20.

6. Il s'agit d'un des problèmes trouvés dans la formulation du contrôleur dit " L_1 -adaptatif" [33]

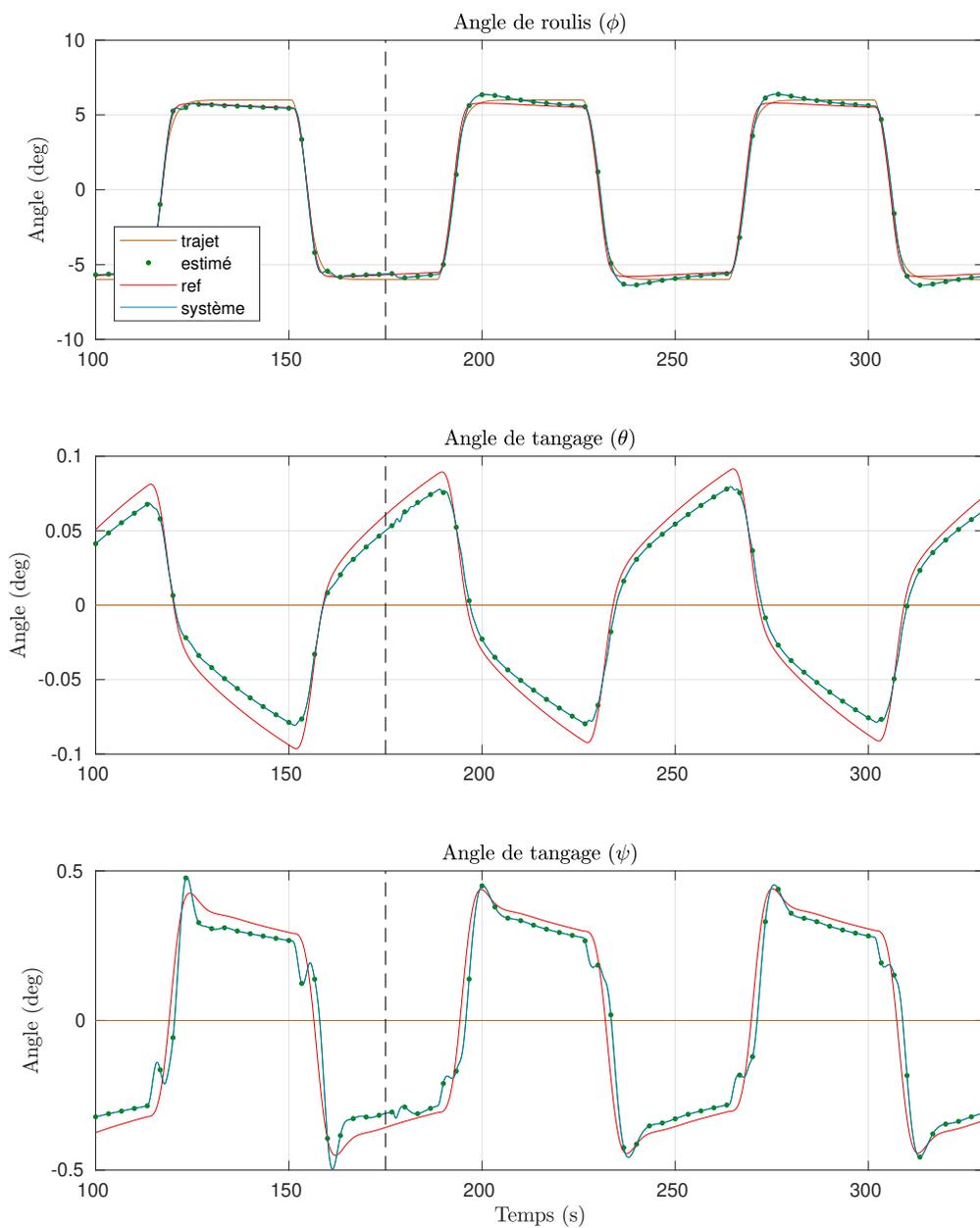


Figure 5.9 Réponse temporelle des angles ϕ , θ , ψ après l'adaptation du gain K_x

La figure 5.9 présente les angles du système pour l'entrée en onde carrée sur ϕ (en jaune foncé). Il est à noter les différentes échelles pour les trois angles. Après 175 s, les oscillations sur ψ ont été améliorées.

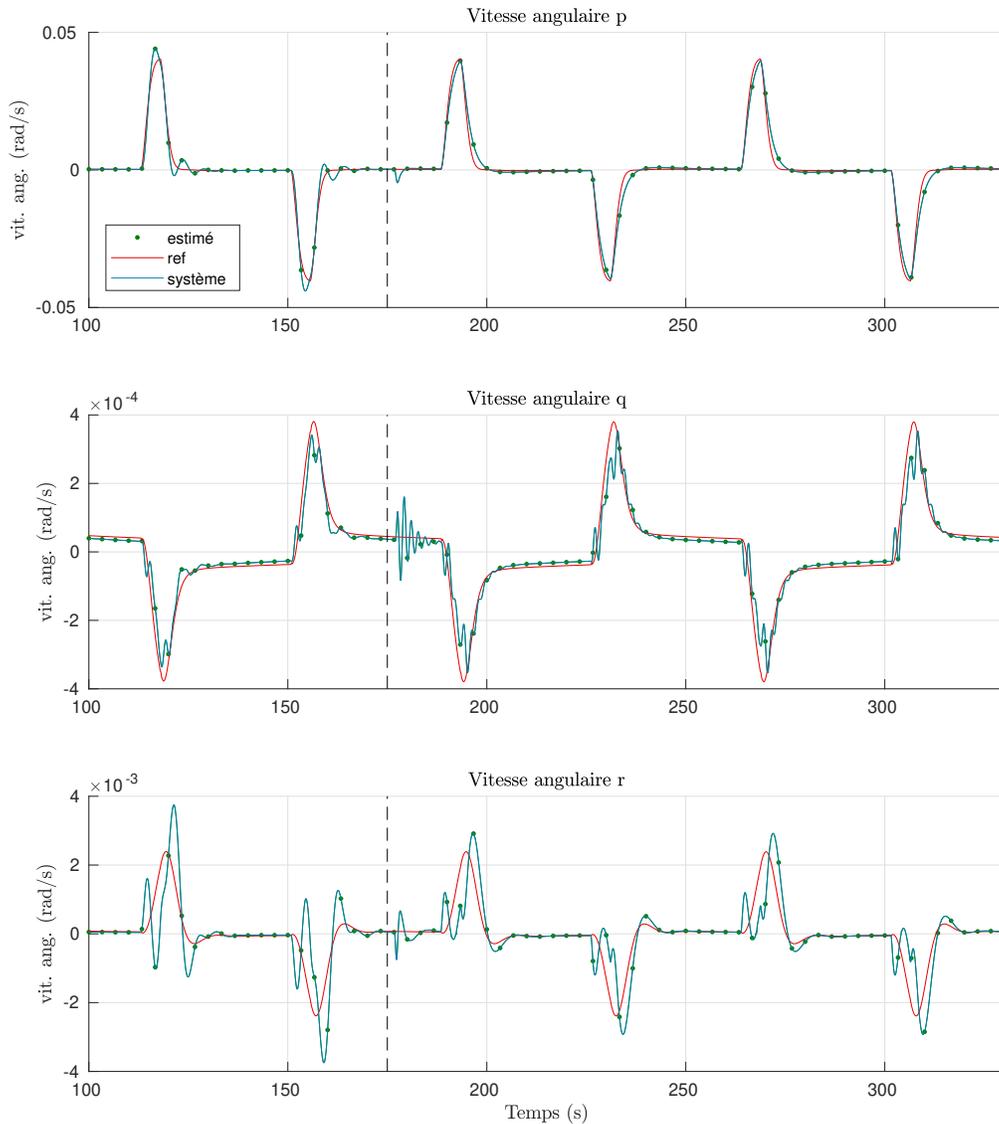


Figure 5.10 Réponse temporelle des vitesses angulaires p, q, r après l'adaptation du gain K_x

La figure 5.10 montre les signaux des vitesses angulaires du système. Après l'adaptation, on remarque une amélioration des oscillations sur p et r . L'état q , qui n'a pas été géré dans l'adaptation, garde une réponse similaire malgré un transitoire vers 180 s dû à la variation des gains du système. La figure 5.11 présente le résultat du système en boucle fermée en terme de position de pôles avant et après adaptation. Leurs valeurs numériques sont d'ailleurs présentées au tableau suivant.

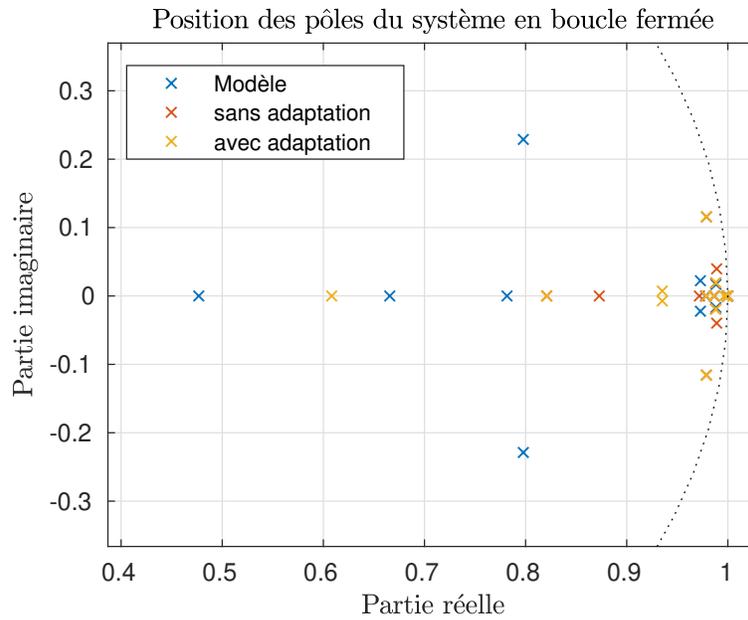


Figure 5.11 Repositionnement des pôles vers le modèle de référence en utilisant K'_ξ

Le tableau 5.1 présente l'emplacement des pôles du modèle de référence, du système initial (Réel) et du système après mise à jour des gains. La colonne "Distance" présente l'écart de chaque pôle entre le modèle de référence et son pôle correspondant du système après adaptation. L'amélioration de cette distance est affichée en pourcentage dans la colonne "Effet".

Tableau 5.1 Position des pôles après adaptation, avec amélioration pour chaque pôle

Modèle	Réel	Après adaptation	Distance	Effet (%)
0.4768	0.8209	0.8208	0.3440	+0.04
0.6657	0.8728	0.6083	0.0574	+72.28
0.7816	0.9718	0.9965	0.2148	-12.95
$0.7978 - 0.2288i$	$0.9879 - 0.0191i$	$0.9351 - 0.0073i$	0.2607	+7.91
$0.7978 + 0.2288i$	$0.9879 + 0.0191i$	$0.9351 + 0.0073i$	0.2607	+7.91
$0.9729 - 0.0223i$	$0.9788 - 0.1156i$	$0.9788 - 0.1161i$	0.0939	-0.46
$0.9729 + 0.0223i$	$0.9788 + 0.1156i$	$0.9788 + 0.1161i$	0.0939	-0.46
0.9780	0.9786	0.9786	0.0006	-7.04
0.9866	0.9864	0.9864	0.0002	~ 0
$0.9882 - 0.0169i$	$0.9889 - 0.0397i$	$0.9882 - 0.0193i$	0.0024	+89.56
$0.9882 + 0.0169i$	$0.9889 + 0.0397i$	$0.9882 + 0.0193i$	0.0024	+89.56
0.9994	0.9994	0.9993	0	~ 0
1.0000	1.0000	1.0000	0	0

En effectuant la simulation mais cette fois-ci avec une mise à jour des gains avec \mathbf{K}'_{\sim} , soit en préservant la structure du contrôleur, le nouveau lieu des racines du système est obtenu et présenté à la figure 5.12. Les valeurs numériques des pôles sont comparées dans le tableau 5.2.

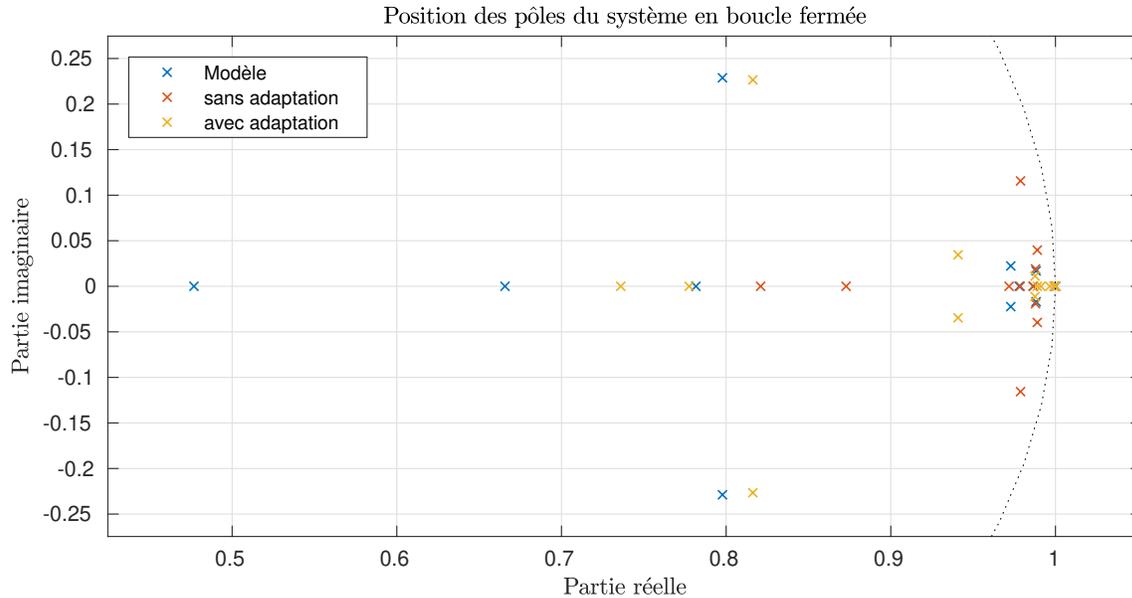


Figure 5.12 Repositionnement des pôles vers le modèle de référence en utilisant \mathbf{K}'_{\sim}

Tableau 5.2 Position des pôles après adaptation, avec amélioration pour chaque pôle

Modèle	Réel	Après adaptation	Distance	Effet (%)
0.4768	0.8209	0.7360	0.2592	+24.68
0.6657	0.8728	0.7776	0.1119	+45.98
0.7816	0.9718	0.9889	0.2073	-8.99
$0.7978 - 0.2288i$	$0.9879 - 0.0191i$	$0.8162 - 0.2266i$	0.0186	+93.43
$0.7978 + 0.2288i$	$0.9879 + 0.0191i$	$0.8162 + 0.2266i$	0.0186	+93.43
$0.9729 - 0.0223i$	$0.9788 - 0.1156i$	$0.9408 - 0.0346i$	0.0343	+63.33
$0.9729 + 0.0223i$	$0.9788 + 0.1156i$	$0.9408 + 0.0346i$	0.0343	+63.33
0.9780	0.9786	0.9911	0.0131	-2144
0.9866	0.9864	0.9966	0.0100	-5459
$0.9882 - 0.0169i$	$0.9889 - 0.0397i$	$0.9876 - 0.0112i$	0.0057	+75.00
$0.9882 + 0.0169i$	$0.9889 + 0.0397i$	$0.9876 + 0.0112i$	0.0057	+75.00
0.9994	0.9994	0.9993	0	0
1.0000	1.0000	1.0000	0	0

On remarque que ce sont principalement les pôles à proximité du point (1, 0), soit la dynamique des

intégrateurs, où la distance avec les pôles désirés ne s'est pas réduite.

En associant les pôles un à un pour comparer le lieu des racines du système estimé et celui avec le contrôleur modifié, il est possible de vérifier si la distance des pôles obtenus avec ceux désirés a été réduite. Dans la dernière colonne, on remarque que les pôles ayant eu un effet négatif sont principalement ceux n'ayant pas été modifiés par l'adaptation (la distance est relative aux pôles réels). Le contrôleur a néanmoins modifié le lieu des racines dans la direction du modèle de référence pour en améliorer les performances du système.

L'adaptation avec le modèle non linéaire donne des résultats différents, étant donné que le linéarisé tangent du système n'est pas constant. Une simulation avec l'implémentation du contrôleur démontre que l'adaptation se fait correctement, mais avec des transitoires beaucoup plus lentes.

CHAPITRE 6 CONCLUSION

Les développements de ce mémoire ont été présentés en trois temps. La première partie présentait une commande classique pour le contrôle moteur du drone. Ensuite était détaillée la commande robuste utilisée pour la stabilisation de la boucle interne du drone. Et enfin, une loi de commande adaptative a été développée pour compléter la commande robuste sur la boucle de stabilisation interne. Ces travaux se sont fait dans deux contextes différents, soit dans un contexte industriel avec la compagnie Laflamme Aéro Inc., puis dans un contexte académique avec Polytechnique Montréal. Ce chapitre effectue un survol de ce qui a été fait et apporte des détails quant aux améliorations possibles.

6.1 Considérations sur les limitations matérielles

Étant donné que le travail s'est effectué en grande partie en milieu industriel, le contrôle moteur fourni a occupé une grande partie des activités de recherche. Pour fournir un système robuste et fiable, un effort supplémentaire a été mis sur la conception de son circuit imprimé et de la programmation du microcontrôleur présent sur la carte. Ces aspects ont été abordés brièvement dans le chapitre du contrôle moteur, mais représentent un aspect important de la stabilité du système pris dans son ensemble. Par exemple, avant la réalisation du circuit imprimé de la figure 3.8, un circuit prototype avait été utilisé lors de la synthèse du contrôleur. Ce prototype avait des modes de vibration si grands que, pour un test sur banc d'essai, maintenir le contrôle moteur devenait difficile. Ceci souligne aussi l'importance à accorder sur le matériel physique à utiliser pour embarquer un contrôleur, au même titre que le choix de sa loi de commande.

Une réflexion similaire est possible pour le contrôle robuste, relativement à l'autopilote commercial qui a été choisi. La plateforme offrait une solution clé en main puisqu'aucune conception n'était nécessaire, mais en contrepartie sa faible puissance de calcul et son processeur limité au calcul point-fixe ont apporté beaucoup de restrictions à l'action possible de l'autopilote. La synthèse de la commande robuste a donc été effectuée de nouveau pour pallier à ce problème, mais les performances atteignables restent tout de même grandement limitées par le matériel choisi.

6.2 Implémentation des commandes robustes et adaptatives

Le travail sur la commande robuste a permis de souligner l'importance de l'architecture choisie pour les commandes de vol d'un hélicoptère tandem. Les résultats obtenus montrent les performances atteignables pour une architecture de contrôle utilisant l'erreur, comparativement à l'utilisation de la

mesure en sortie et ce, pour les mêmes critères d'une synthèse H_∞ structurée.

Le contrôleur développé dans le dernier chapitre a montré un exemple d'application d'une commande adaptative sur un système non linéaire à plusieurs entrées et plusieurs sorties. La commande présentée s'effectue sur un système normalement instable en boucle ouverte et se base sur les résultats d'une commande robuste pour calculer la mise à jour du contrôleur (le contrôleur initial est robuste). Dans cette optique, les travaux offrent un niveau de complexité supplémentaire à ce qui est présenté dans la littérature pour le contrôle adaptatif de drones tandems. Les choix d'implémentation pour un estimateur capable de demeurer robuste numériquement même en l'absence d'excitation en entrée ont été développés et justifiés par des résultats numériques. La loi de commande choisie, soit indirecte et utilisant un placement de pôles, présentait l'amélioration possible en recalculant *en ligne* le contrôleur avec le linéarisé tangent du système.

6.3 Considérations sur le modèle utilisé

Le modèle dynamique du drone utilisé est un modèle très complet et complexe en soi. De par les temps d'ajustements nécessaires pour effectuer de simples tests de stabilité, le modèle linéarisé du drone a été privilégié pour implémenter les lois de commande adaptatives. Par contre, le modèle non linéaire offre la flexibilité de considérer ou non certains éléments agissant avec le drone, tel le temps d'actuation des servomoteurs ou l'impact du vent avec la tenue en vol du drone.

Une des considérations à avoir avec les résultats obtenus se rapporte à la validité du modèle linéaire. Cet aspect vient de l'utilisation d'une approximation linéaire de la dynamique du drone comme système de référence. Étant donné que le contrôleur adaptatif choisi est indirect, un moyen de valider le modèle de référence aurait été de faire fonctionner l'estimateur par MCRS sur le drone en vol. De cette façon, un modèle de référence de la dynamique linéarisée de l'appareil aurait pu être disponible et être comparé au modèle théorique utilisé avec la commande robuste. Cette méthode aurait aussi permis de vérifier la tolérance de l'estimateur face aux vibrations de l'appareil.

6.4 Implication des simplifications effectuées

Les lois de commande présentées proviennent de divers choix de conception affectant leurs performances et leur réalisabilité. Par exemple, le manque d'une action dérivée du contrôle moteur impacte les temps de réponse possibles pour une même valeur de dépassement de la commande. Par contre, ce choix évite de devoir calculer numériquement la dérivée du signal d'intérêt, ce qui permet davantage de robustesse. Pour la loi de commande stabilisant le drone déterminée via une synthèse H_∞ structurée, les résultats sont essentiellement limités aux gabarits fréquentiels utilisés pour les points d'opération retenus. Considérer de plus basses vitesses de croisière aurait permis d'obtenir

de meilleurs résultats de synthèse, au détriment de l'enveloppe de vol possible pour le drone. En modifiant les gabarits fréquentiels utilisés, il aurait aussi été possible d'améliorer légèrement le couplage présents dans les résultats de la figure 4.14. La sélection des gabarits utilisés étant déjà une tâche délicate, il a été choisi de garder les gabarits utilisés précédemment lors de la synthèse H_∞ utilisant la mesure de sortie afin de mieux comparer les deux architectures de commande.

La loi de commande adaptative présentée comporte aussi plusieurs choix de conception affectant ses performances finales. L'utilisation du modèle dynamique discret du système augmente le nombre de paramètres à estimer, mais permet en contrepartie d'utiliser directement l'estimé dans la mise à jour des gains sans devoir effectuer de conversions vers le modèle à temps continu. Le placement de pôles est donc aussi limité par la fréquence d'échantillonnage choisie par l'estimateur. Afin de simplifier la mise à jour du gain, la dynamique des intégrateurs et les états non présents dans la commande ont été négligés, ce qui explique l'écart de certains pôles dans les résultats finaux. Sur ces résultats, les simulations se sont résumées en l'analyse de la réponse du système pour des commandes échelon en roulis. D'autres types de signaux auraient très bien pu être utilisés et auraient influencé la quantité d'excitation présente à l'entrée du modèle. Les simulations pour des commandes en roulis permettent néanmoins de rapidement valider la stabilité sur la dynamique latérale du système.

6.5 Avenues possibles de la commande adaptative

Avec la puissance de calcul des processeurs augmentant toujours de plus en plus, l'utilisation d'une commande adaptative évaluant en parallèle plusieurs modèles du système devient possible. Tel que présenté par Landau [43], la loi de commande adaptative pourrait avoir un comportement similaire au contrôleur robuste utilisé, soit par séquençement de gains en fonction des dynamiques estimées. L'estimateur calculerait donc, en tout temps, la proximité du système avec chacun des modèles connus et ajusterait ses gains pour que les dynamiques du système demeurent ceux d'un système de référence. Cette méthode offre l'avantage d'une robustesse accrue, étant donné que ce séquençement pourrait aussi être évalué a priori via une synthèse H_∞ sur l'ensemble des modèles linéaire considérés.

D'une autre part, l'utilisation d'un estimateur non linéaire comme l'algorithme de *Levenberg-Marquard* offre une possibilité prometteuse. En émettant l'hypothèse que l'estimateur non linéaire suit plus fidèlement les dynamiques du système, alors les variations sur les valeurs de l'estimé seront moindres lorsque l'estimateur converge. Celui-ci permettrait par exemple d'obtenir directement les relations trigonométriques de l'équation 4.4, soit la relation non linéaire entre certains états.

RÉFÉRENCES

- [1] A. V. Aho et J. E. Hopcroft, *The design and analysis of computer algorithms*. Pearson Education India, 1974.
- [2] S. Akhtar, R. Venugopal, et D. S. Bernstein, “Logarithmic lyapunov functions for direct adaptive stabilization with normalized adaptive laws”, *International Journal of Control*, vol. 77, no. 7, pp. 630–638, 2004.
- [3] F. Albu, “Improved variable forgetting factor recursive least square algorithm”, dans *Control Automation Robotics & Vision (ICARCV), 2012 12th International Conference on*. IEEE, 2012, pp. 1789–1793.
- [4] B. Aloliwi et H. K. Khalil, “Robust adaptive output feedback control of nonlinear systems without persistence of excitation”, *Automatica*, vol. 33, no. 11, pp. 2025–2032, 1997.
- [5] B. Anderson, “Exponential stability of linear equations arising in adaptive identification”, *IEEE Transactions on Automatic Control*, vol. 22, no. 1, pp. 83–88, 1977.
- [6] T. W. Anderson, “Least squares and best unbiased estimates”, *The Annals of Mathematical Statistics*, vol. 33, no. 1, pp. 266–272, 1962.
- [7] J. Aseltine, A. Mancini, et C. Sarture, “A survey of adaptive control systems”, *IRE Transactions on Automatic Control*, vol. 6, no. 1, pp. 102–108, December 1958. DOI : 10.1109/TAC.1958.1105168
- [8] B. Ashok, S. D. Ashok, et C. R. Kumar, “A review on control system architecture of a si engine management system”, *Annual Reviews in Control*, vol. 41, pp. 94–118, 2016.
- [9] M. H. Bahari, F. N. Moharrami, M. E. Ganjeh, et A. Karsaz, “Tracking a high maneuver target based on intelligent matrix covariance resetting”, dans *Image and Signal Processing and Analysis, 2007. ISPA 2007. 5th International Symposium on*. IEEE, 2007, pp. 35–40.
- [10] J. C. A. Barata et M. S. Hussein, “The moore–penrose pseudoinverse : A tutorial review of the theory”, *Brazilian Journal of Physics*, vol. 42, no. 1, pp. 146–165, Apr 2012. DOI : 10.1007/s13538-011-0052-z. En ligne : <https://doi.org/10.1007/s13538-011-0052-z>

- [11] G. Bartolini, A. Ferrara, et V. Utkin, “Adaptive sliding mode control in discrete-time systems”, *Automatica*, vol. 31, no. 5, pp. 769 – 773, 1995. DOI : [https://doi.org/10.1016/0005-1098\(94\)00154-B](https://doi.org/10.1016/0005-1098(94)00154-B). En ligne : <http://www.sciencedirect.com/science/article/pii/000510989400154B>
- [12] L. Belmonte, R. Morales, A. Fernandez-Caballero, et J. Somolinos, “Robust decentralized nonlinear control for a twin rotor mimo system”, *SENSORS*, vol. 16, no. 8, p. 1160, 2016.
- [13] K. Benzemrane, G. Damm, et G. L. Santosuosso, “Adaptive observer and kalman filtering”, dans *17th World Congress, International Federation of Automatic Control, IFAC*, vol. 17, no. 1 PART 1, 2008, pp. 3865–3870.
- [14] M. Bodson, “An adaptive algorithm with information-dependent data forgetting”, dans *American Control Conference, Proceedings of the 1995*, vol. 5. IEEE, 1995, pp. 3485–3489.
- [15] G. Chang, “On kalman filter for linear system with colored measurement noise”, *Journal of Geodesy*, vol. 88, no. 12, pp. 1163–1170, Dec 2014. DOI : [10.1007/s00190-014-0751-7](https://doi.org/10.1007/s00190-014-0751-7). En ligne : <https://doi.org/10.1007/s00190-014-0751-7>
- [16] . Chen, Ben M., *Robust and H [infinity] control : Ben M. Chen.* London : Springer, 2000.
- [17] D. W. Clarke, “Adaptive predictive control”, *Annual Reviews in Control*, vol. 20, pp. 83–94, 1996.
- [18] J. A. Cook et B. K. Powell, “Modeling of an internal combustion engine for control analysis”, *IEEE Control Systems Magazine*, vol. 8, no. 4, pp. 20–26, Aug 1988. DOI : [10.1109/37.7726](https://doi.org/10.1109/37.7726)
- [19] P. R. Crossley et J. A. Cook, “A nonlinear engine model for drivetrain system development”, dans *Control 1991. Control '91., International Conference on*, Mar 1991, pp. 921–925 vol.2.
- [20] J. J. D. CROZ et N. J. HIGHAM, “Stability of methods for matrix inversion”, *IMA Journal of Numerical Analysis*, vol. 12, no. 1, pp. 1–19, 1992. DOI : [10.1093/imanum/12.1.1](https://doi.org/10.1093/imanum/12.1.1). En ligne : <http://dx.doi.org/10.1093/imanum/12.1.1>
- [21] S. Dasgupta et Y.-F. Huang, “Asymptotically convergent modified recursive least-squares with data-dependent updating and forgetting factor for systems with bounded noise”, *IEEE Transactions on Information Theory*, vol. 33, no. 3, pp. 383–392, May 1987. DOI : [10.1109/TIT.1987.1057307](https://doi.org/10.1109/TIT.1987.1057307)

- [22] A. Datta, “Performance improvement in decentralized adaptive control : a modified model reference scheme”, *IEEE Transactions on Automatic Control*, vol. 38, no. 11, pp. 1717–1722, 1993.
- [23] A. Datta et P. A. Ioannou, “Performance improvement versus robust stability in model reference adaptive control”, dans [1991] *Proceedings of the 30th IEEE Conference on Decision and Control*, Dec 1991, pp. 748–753 vol.1. DOI : 10.1109/CDC.1991.261411
- [24] J. DOYLE, K. GLOVER, P. KHARGONEKAR, et B. FRANCIS, “State-space solutions to standard h-2 and h-infinity control-problems”, *IEEE TRANSACTIONS ON AUTOMATIC CONTROL*, vol. 34, no. 8, pp. 831–847, 1989.
- [25] Z. T. Dydek, A. M. Annaswamy, et E. Lavretsky, “Adaptive control and the nasa x-15 program : A concise history, lessons learned, and a provably correct design”, dans *2008 American Control Conference*, June 2008, pp. 2957–2962. DOI : 10.1109/ACC.2008.4586945
- [26] A. Dzul, T. Hamel, et R. Lozano, “Nonlinear control for a tandem rotor helicopter”, *IFAC Proceedings Volumes*, vol. 35, no. 1, pp. 229–234, 2002.
- [27] T. Fortescue, L. Kershenbaum, et B. Ydstie, “Implementation of self-tuning regulators with variable forgetting factors”, *Automatica*, vol. 17, no. 6, pp. 831 – 835, 1981. DOI : [https://doi.org/10.1016/0005-1098\(81\)90070-4](https://doi.org/10.1016/0005-1098(81)90070-4). En ligne : <http://www.sciencedirect.com/science/article/pii/0005109881900704>
- [28] G. Goodwin et K. S. Sin, “Adaptive filtering prediction and control”, *Englewood Cliffs : Prentice-Hall*, 1984.
- [29] G. C. Goodwin, P. J. Ramadge, et P. E. Caines, “Discrete time stochastic adaptive control”, *SIAM Journal on Control and Optimization*, vol. 19, no. 6, pp. 829–25, 11 1981, copyright - Copyright] © 1981 Society for Industrial and Applied Mathematics; Last updated - 2012-03-03. En ligne : <https://search.proquest.com/docview/926035205?accountid=40695>
- [30] B. Hariri, A. Shenton, et R. Dorey, “Parameter identification, estimation, and dynamometer validation of the nonlinear dynamics of an automotive spark-ignition engine”, *Journal of Vibration and Control*, vol. 4, no. 1, pp. 47–59, 1998.
- [31] E. Hendricks, “Engine modelling for control applications : A critical survey”, *Meccanica*, vol. 32, no. 5, pp. 387–396, Oct 1997. DOI : 10.1023/A:1004247514972. En ligne : <https://doi.org/10.1023/A:1004247514972>

- [32] T. Häggglund et K. J. Åström, “Supervision of adaptive control algorithms”, *IFAC Proceedings Volumes*, vol. 30, no. 6, pp. 609 – 616, 1997, iFAC Conference on Control of Industrial Systems "Control for the Future of the Youth", Belfort, France, 20-22 May. DOI : [https://doi.org/10.1016/S1474-6670\(17\)43431-8](https://doi.org/10.1016/S1474-6670(17)43431-8). En ligne : <http://www.sciencedirect.com/science/article/pii/S1474667017434318>
- [33] P. A. Ioannou, A. M. Annaswamy, K. S. Narendra, et S. Jafari, “L1-adaptive control : stability, robustness, and interpretations”, *IEEE Transactions on Automatic Control*, vol. 59, no. 11, pp. 3075–3080, 2014.
- [34] P. A. Ioannou et J. Sun, *Robust adaptive control*. PTR Prentice-Hall Upper Saddle River, NJ, 1996, vol. 1.
- [35] M. Ishitobi, M. Nishi, et K. Nakasaki, “Nonlinear adaptive model following control for a 3-dof tandem-rotor model helicopter”, *Control Engineering Practice*, vol. 18, no. 8, pp. 936–943, 2010.
- [36] R. M. Johnstone, C. R. Johnson, R. R. Bitmead, et B. D. O. Anderson, “Exponential convergence of recursive least squares with exponential forgetting factor”, dans *1982 21st IEEE Conference on Decision and Control*, Dec 1982, pp. 994–997. DOI : [10.1109/CDC.1982.268295](https://doi.org/10.1109/CDC.1982.268295)
- [37] H. K. Khalil, *Nonlinear systems; 3rd ed.* Upper Saddle River, NJ : Prentice-Hall, 2002, the book can be consulted by contacting : PH-AID : Wallet, Lionel. En ligne : <https://cds.cern.ch/record/1173048>
- [38] V. Klema et A. Laub, “The singular value decomposition : Its computation and some applications”, *IEEE Transactions on Automatic Control*, vol. 25, no. 2, pp. 164–176, 1980.
- [39] G. Kreisselmeier, “An indirect adaptive controller with a self-excitation capability”, *IEEE Transactions on Automatic Control*, vol. 34, no. 5, pp. 524–528, 1989.
- [40] M. Krstic et P. Kokotovic, “Control lyapunov functions for adaptive nonlinear stabilization”, *SYSTEMS & CONTROL LETTERS*, vol. 26, no. 1, pp. 17–23, 1995.
- [41] M. Kubalčík, V. Bobál, et P. Chalupa, “Adaptive control of twin rotor mimo system : Polynomial approach”, *IFAC Proceedings Volumes*, vol. 38, no. 1, pp. 892–897, 2005.
- [42] F. Laliberté, “Modélisation et commande d’un drone hélicoptère tandem”, Mémoire de maîtrise, École Polytechnique de Montréal, Montréal, Août 2017.

- [43] I. D. Landau, “From robust control to adaptive control”, *Control Engineering Practice*, vol. 7, no. 9, pp. 1113–1124, 1999.
- [44] H. Lhachemi, D. Saussié, et G. Zhu, “Explicit hidden coupling terms handling in gain-scheduling control design via eigenstructure assignment”, *Control Engineering Practice*, vol. 58, pp. 1–11, 2017.
- [45] H. Lhachemi, “Commande des systèmes aérospatiaux non linéaires et à dimension infinie”, Thèse de doctorat, École Polytechnique de Montréal, décembre 2017.
- [46] D. G. Luenberger, *Optimization by vector space methods*. John Wiley & Sons, 1997.
- [47] C.-C. Luo, R.-F. Liu, C.-D. Yang, et Y.-H. Chang, “Helicopter h-infinity control design with robust flying quality”, *Aerospace Science and Technology*, vol. 7, no. 2, pp. 159–169, 2003.
- [48] I. M. Mareels, B. D. Anderson, R. R. Bitmead, M. Bodson, et S. S. Sastry, “Revisiting the mit rule for adaptive control”, *IFAC Proceedings Volumes*, vol. 20, no. 2, pp. 161 – 166, 1987, 2nd IFAC Workshop on Adaptive Systems in Control and Signal Processing 1986, Lund, Sweden, 30 June-2 July 1986. DOI : [https://doi.org/10.1016/S1474-6670\(17\)55954-6](https://doi.org/10.1016/S1474-6670(17)55954-6). En ligne : <http://www.sciencedirect.com/science/article/pii/S1474667017559546>
- [49] M. Margolis et C. Leondes, “A parameter tracking servo for adaptive control systems”, *IRE Transactions on Automatic Control*, vol. 4, no. 2, pp. 100–111, November 1959. DOI : 10.1109/TAC.1959.1104854
- [50] R. Marino, “Adaptive control of nonlinear systems : Basic results and applications”, *Annual Reviews in Control*, vol. 21, pp. 55–66, 1997.
- [51] R. Monopoli, “Model reference adaptive control with an augmented error signal”, *IEEE Transactions on Automatic Control*, vol. 19, no. 5, pp. 474–484, 1974.
- [52] R. V. MONOPOLI et C. C. HSING, “Parameter adaptive control of multivariable systems”, *International Journal of Control*, vol. 22, no. 3, pp. 313–327, 1975. DOI : 10.1080/00207177508922087. En ligne : <https://doi.org/10.1080/00207177508922087>
- [53] K. M. Moudgalya, *Digital control : Kannan M. Moudgalya*. Chichester, England : Wiley, 2007.

- [54] S. M. Naik, P. R. Kumar, et B. E. Ydstie, “Robust continuous-time adaptive control by parameter projection”, *IEEE Transactions on Automatic Control*, vol. 37, no. 2, pp. 182–197, Feb 1992. DOI : 10.1109/9.121620
- [55] K. Narendra, Y.-H. Lin, et L. Valavani, “Stable adaptive controller design, part ii : Proof of stability”, *IEEE Transactions on Automatic Control*, vol. 25, no. 3, pp. 440–448, 1980.
- [56] K. S. Narendra et N. O. Olegg, “Exact output tracking in decentralized adaptive control systems”, *IEEE Transactions on Automatic Control*, vol. 47, no. 2, pp. 390–395, Feb 2002. DOI : 10.1109/9.983386
- [57] K. S. Narendra et K. Parthasarathy, “Identification and control of dynamical systems using neural networks”, *IEEE Transactions on Neural Networks*, vol. 1, no. 1, pp. 4–27, March 1990. DOI : 10.1109/72.80202
- [58] K. S. Narendra et A. M. Annaswamy, “Persistent excitation in adaptive systems”, *International Journal of Control*, vol. 45, no. 1, pp. 127–160, 1987. DOI : 10.1080/00207178708933715. En ligne : <https://doi.org/10.1080/00207178708933715>
- [59] NASA, “Experience with the x-15 adaptive flight control system”, NASA Flight Research Center, Rapp. tech. NASA-TN-D-6208, March 1971.
- [60] S. Omari, M.-D. Hua, G. Ducard, et T. Hamel, “Hardware and software architecture for nonlinear control of multicopter helicopters”, *IEEE/ASME Transactions on Mechatronics*, vol. 18, no. 6, pp. 1724–1736, 2013.
- [61] R. Ortega et Y. Tang, “Robustness of adaptive controllers—a survey”, *Automatica*, vol. 25, no. 5, pp. 651 – 677, 1989. DOI : [https://doi.org/10.1016/0005-1098\(89\)90023-X](https://doi.org/10.1016/0005-1098(89)90023-X). En ligne : <http://www.sciencedirect.com/science/article/pii/000510988990023X>
- [62] S. Oudin, “Commande adaptative pour avion de transport tolérante aux erreurs de modèle et aux pannes”, Thèse de doctorat, Université de Toulouse, Institut Supérieur de l’Aéronautique et de l’Espace (ISAE), novembre 2013.
- [63] C. Paleologu, J. Benesty, et S. Ciochina, “A robust variable forgetting factor recursive least-squares algorithm for system identification”, *IEEE Signal Processing Letters*, vol. 15, pp. 597–600, 2008. DOI : 10.1109/LSP.2008.2001559

- [64] V. K. Pandey, I. Kar, et C. Mahanta, “Control of twin-rotor mimo system using multiple models with second level adaptation”, *IFAC PapersOnLine*, vol. 49, no. 1, pp. 676–681, 2016.
- [65] P. Parks, “Liapunov redesign of model reference adaptive control systems”, *IEEE Transactions on Automatic Control*, vol. 11, no. 3, pp. 362–367, 1966.
- [66] P. Phillipson, “Design methods for model-reference adaptive systems”, *Proceedings of the Institution of Mechanical Engineers*, vol. 183, no. 1, pp. 695–706, 1968.
- [67] M. M. Polycarpou et P. A. Ioannou, “On the existence and uniqueness of solutions in adaptive control systems”, *IEEE Transactions on Automatic Control*, vol. 38, no. 3, pp. 474–479, 1993.
- [68] ———, *Identification and control of nonlinear systems using neural network models : Design and stability analysis*. University of Southern Calif., 1991.
- [69] R. W. Prouty, *Helicopter performance, stability, and control*. Phillips Pub Co, 1995.
- [70] P. Puleston, S. Spurgeon, et G. Monsees, “Automotive engine speed control : A robust nonlinear control framework”, *IEE Proceedings-Control Theory and Applications*, vol. 148, no. 1, pp. 81–87, 2001.
- [71] A. Rahideh, A. H. Bajodah, et M. H. Shaheed, “Real time adaptive nonlinear model inversion control of a twin rotor mimo system using neural networks”, *Engineering Applications of Artificial Intelligence*, vol. 25, no. 6, pp. 1289–1297, 2012.
- [72] W. J. Rugh, *Linear system theory : Wilson J. Rugh, 2e éd*. Upper Saddle River, N.J : Prentice Hall, 1996.
- [73] W. J. Rugh et J. S. Shamma, “Research on gain scheduling”, *Automatica*, vol. 36, no. 10, pp. 1401 – 1425, 2000. DOI : [https://doi.org/10.1016/S0005-1098\(00\)00058-3](https://doi.org/10.1016/S0005-1098(00)00058-3).
En ligne : <http://www.sciencedirect.com/science/article/pii/S0005109800000583>
- [74] M. E. SALGADO, G. C. GOODWIN, et R. H. MIDDLETON, “Modified least squares algorithm incorporating exponential resetting and forgetting”, *International Journal of Control*, vol. 47, no. 2, pp. 477–491, 1988. DOI : 10.1080/00207178808906026. En ligne : <https://www.tandfonline.com/doi/abs/10.1080/00207178808906026>
- [75] B. Shackcloth, “Design of model reference control systems using a lyapunov synthesis technique”, *Proceedings of the Institution of Electrical Engineers*, vol. 114, no. 2, p. 299, 1967.

- [76] R. Singh et K. Narendra, “Prior information in the design of multivariable adaptive controllers”, *IEEE Transactions on Automatic Control*, vol. 29, no. 12, pp. 1108–1111, 1984.
- [77] P. Sodhi et I. Kar, “Adaptive backstepping control for a twin rotor mimo system”, *IFAC Proceedings Volumes*, vol. 47, no. 1, pp. 740 – 747, 2014, 3rd International Conference on Advances in Control and Optimization of Dynamical Systems (2014). DOI : <https://doi.org/10.3182/20140313-3-IN-3024.00082>. En ligne : <http://www.sciencedirect.com/science/article/pii/S1474667016327379>
- [78] Y. Song, K. Zhao, et M. Krstic, “Adaptive control with exponential regulation in the absence of persistent excitation”, *IEEE Transactions on Automatic Control*, vol. 62, no. 5, pp. 2589–2596, May 2017. DOI : 10.1109/TAC.2016.2599645
- [79] J. T. Spooner et K. M. Passino, “Stable adaptive control using fuzzy systems and neural networks”, *IEEE Transactions on Fuzzy Systems*, vol. 4, no. 3, pp. 339–359, 1996.
- [80] . Stevens, Brian L., F. L. Lewis, et . Johnson, Eric N., *Aircraft control and simulation : dynamics, controls design, and autonomous systems ; Brian L. Stevens, Frank L. Lewis, Eric N. Johnson*. Hoboken, New Jersey : Wiley, 2016.
- [81] P. R. Stromer, “Adaptive or self-optimizing control systems — a bibliography”, *IRE Transactions on Automatic Control*, vol. AC-4, no. 1, pp. 65–68, May 1959. DOI : 10.1109/TAC.1959.6429404
- [82] G. Tao, “Model reference adaptive control of multivariable plants with unknown interactor matrix”, dans *29th IEEE Conference on Decision and Control*, Dec 1990, pp. 2730–2735 vol.5. DOI : 10.1109/CDC.1990.203274
- [83] G. Tao et P. Ioannou, “Robust model reference adaptive control for multivariable plants”, *International Journal of Adaptive Control and Signal Processing*, vol. 2, no. 3, pp. 217–248, 1988.
- [84] G. Tao, “Multivariable adaptive control : A survey”, *Automatica*, vol. 50, no. 11, pp. 2737 – 2764, 2014. DOI : <https://doi.org/10.1016/j.automatica.2014.10.015>. En ligne : <http://www.sciencedirect.com/science/article/pii/S0005109814003963>
- [85] W. W. Woo, S. A. Svoronos, et O. D. Crisalle, “A directional forgetting factor for single-parameter variations”, dans *Proceedings of 1995 American Control Conference - ACC’95*, vol. 2, June 1995, pp. 1149–1151 vol.2. DOI : 10.1109/ACC.1995.520928

- [86] S. Zarovy et M. Costello, “Extended state observer for helicopter mass and center-of-gravity estimation”, *Journal of Aircraft*, vol. 52, no. 6, pp. 1939–1950, 2015.
- [87] K. Åström, “Theory and applications of adaptive control—a survey”, *Automatica*, vol. 19, no. 5, pp. 471 – 486, 1983. DOI : [https://doi.org/10.1016/0005-1098\(83\)90002-X](https://doi.org/10.1016/0005-1098(83)90002-X).
En ligne : <http://www.sciencedirect.com/science/article/pii/S000510988390002X>

ANNEXE A Convergence de l'estimateur RLS avec facteur d'oubli

La preuve suivante est principalement la réécriture de l'article de Johnstone [36].

L'erreur de l'estimateur est défini comme suit :

$$\begin{aligned}
 \tilde{\theta}_k &= \theta^* - \hat{\theta}_k \\
 &= \theta^* - \left(\hat{\theta}_{k-1} + \frac{P_{k-1}H_k^T(z_k - H_k\hat{\theta}_{k-1})}{\lambda + H_kP_{k-1}H_k^T} \right) \\
 &= \tilde{\theta}_{k-1} - \left(\frac{P_{k-1}H_k^T(H_k\theta^* - H_k\hat{\theta}_{k-1})}{\lambda + H_kP_{k-1}H_k^T} \right) \\
 &= \tilde{\theta}_{k-1} - \left(\frac{P_{k-1}H_k^T H_k \tilde{\theta}_{k-1}}{\lambda + H_kP_{k-1}H_k^T} \right) \\
 &= \left(I - \frac{P_{k-1}H_k^T H_k}{\lambda + H_kP_{k-1}H_k^T} \right) \tilde{\theta}_{k-1}
 \end{aligned}$$

Puisque l'on considère que le signal d'entrée a une *persistence d'excitation*, c'est-à-dire qu'il a un contenu fréquentiel suffisamment riche pour distinguer chacun des inconnus du système, ce qui est représenté comme suit :

Une définition de persistence d'excitation pour un estimateur est donnée par [5],[34] :

$$0 < \rho_1 I < \sum_{k=j}^{j+S} H_k^T H_k < \rho_2 I < \infty$$

Un signal u *suffisamment riche d'ordre n* veut donc dire que le signal contient au moins $\frac{n}{2}$ sinusoïdes distincts [34]. Dans le cas où les éléments $x_i(k)$ sont bornés à priori, alors l'équation précédente est vraie pour :

$$0 < \rho_3 I < \sum_{k=j}^{j+S-n+1} \begin{bmatrix} u_{k+n} \\ \dots \\ u_{k-m+1} \end{bmatrix} \begin{bmatrix} u_{k+n} & \dots & u_{k-m+1} \end{bmatrix} < \rho_4 I$$

Définition stochastique de persistence d'excitation :

$$0 < \beta_1 I < E[H_k^T H_k]$$

Spécifier borne supérieure de P^{-1}

Considérons la fonction de Lyapunov suivante :

$$V_k = \tilde{\theta}_k^T P_{k-1}^{-1} \tilde{\theta}_k$$

Qui peut être associé à l'équation d'un hyper-ellipsoïde (loi normale multidimensionnelle) d'ordre n contenant θ^* (voir [21]). Dans le domaine discret, pour déterminer la stabilité au sens de Lyapunov on s'intéresse à " V_k " et " $V_k - V_{k-1}$ ".

$$\begin{aligned}
V_k - V_{k-1} &= \tilde{\theta}_k^T P_k^{-1} \tilde{\theta}_k - \tilde{\theta}_{k-1}^T P_{k-1}^{-1} \tilde{\theta}_{k-1} \\
&= \tilde{\theta}_{k-1}^T \left(I - \frac{P_{k-1} H_k^T H_k}{\lambda + H_k P_{k-1} H_k^T} \right)^T P_k^{-1} \left(I - \frac{P_{k-1} H_k^T H_k}{\lambda + H_k P_{k-1} H_k^T} \right) \tilde{\theta}_{k-1} - \tilde{\theta}_{k-1}^T P_{k-1}^{-1} \tilde{\theta}_{k-1} \\
&= \tilde{\theta}_{k-1}^T \left(I - \frac{H_k^T H_k P_{k-1}}{\lambda + H_k P_{k-1} H_k^T} \right) (\lambda P_{k-1}^{-1} + H_k^T H_k) \left(I - \frac{P_{k-1} H_k^T H_k}{\lambda + H_k P_{k-1} H_k^T} \right) \tilde{\theta}_{k-1} \\
&\quad - \tilde{\theta}_{k-1}^T P_{k-1}^{-1} \tilde{\theta}_{k-1} \\
&= \tilde{\theta}_{k-1}^T (\lambda P_{k-1}^{-1} + H_k^T H_k - \frac{H_k^T H_k (\lambda + P_{k-1} H_k^T H_k)}{\lambda + H_k P_{k-1} H_k^T}) \left(I - \frac{P_{k-1} H_k^T H_k}{\lambda + H_k P_{k-1} H_k^T} \right) \tilde{\theta}_{k-1} \\
&\quad - \tilde{\theta}_{k-1}^T P_{k-1}^{-1} \tilde{\theta}_{k-1} \\
&= \tilde{\theta}_{k-1}^T (\lambda P_{k-1}^{-1} + H_k^T H_k - \frac{H_k^T (\lambda + H_k P_{k-1} H_k^T) H_k}{\lambda + H_k P_{k-1} H_k^T}) \left(I - \frac{P_{k-1} H_k^T H_k}{\lambda + H_k P_{k-1} H_k^T} \right) \tilde{\theta}_{k-1} \\
&\quad - \tilde{\theta}_{k-1}^T P_{k-1}^{-1} \tilde{\theta}_{k-1} \\
&= \tilde{\theta}_{k-1}^T (\lambda P_{k-1}^{-1}) \left(I - \frac{P_{k-1} H_k^T H_k}{\lambda + H_k P_{k-1} H_k^T} \right) - \tilde{\theta}_{k-1}^T P_{k-1}^{-1} \tilde{\theta}_{k-1} \\
&= \tilde{\theta}_{k-1}^T \left(\lambda P_{k-1}^{-1} - \frac{\lambda H_k^T H_k}{\lambda + H_k P_{k-1} H_k^T} \right) - \tilde{\theta}_{k-1}^T P_{k-1}^{-1} \tilde{\theta}_{k-1} \\
&= \tilde{\theta}_{k-1}^T \left((\lambda - 1) P_{k-1}^{-1} - \frac{\lambda H_k^T H_k}{\lambda + H_k P_{k-1} H_k^T} \right) \tilde{\theta}_{k-1} \\
&\leq \tilde{\theta}_{k-1}^T \left((\lambda - 1) P_{k-1}^{-1} \right) \tilde{\theta}_{k-1} \\
&= (\lambda - 1) V_{k-1} < 0
\end{aligned}$$

ANNEXE B Moindres carrés récurrents stabilisés

En rajoutant le terme $\delta(1 - \lambda)I$ dans la mise à jour de l'inverse de la matrice de covariance, on se retrouve à limiter la norme de la matrice de covariance à $\frac{1}{\delta}$. Dans le cas où les états atteignent un équilibre, l'on obtient :

$$\begin{aligned} P_n^{-1} &= \lambda P_{n-1}^{-1} + h_n h_n^T + \delta(1 - \lambda)I \\ P_n h_n h_n^T &= I - \lambda P_n P_{n-1}^{-1} - P_n \delta(1 - \lambda)I \\ &\approx (1 - \lambda)I - \delta(1 - \lambda)P_n \end{aligned}$$

L'évolution de l'estimation des dynamiques devient donc :

$$\begin{aligned} \epsilon_n &= \theta_n - \theta^* \\ \epsilon_n &= \epsilon_{n-1} - P_n h_n h_n^T \epsilon_{n-1} + \delta \lambda (\epsilon_{n-1} - \epsilon_{n-2}) \\ \epsilon_n &= z^{-1} \epsilon_n - z^{-1} P_n h_n h_n^T \epsilon_n + \delta \lambda P_n (z^{-1} \epsilon_n - z^{-2} \epsilon_n) \\ 0 &= \epsilon_n (I + z^{-1}(-1 + (1 - \lambda)I - \delta(1 - \lambda)P_n) - \delta(1 - \lambda)P_n) + z^{-2} \delta \lambda P_n \\ 0 &= \epsilon_n (z^2 + z(-\lambda I - \delta P_n) + \delta \lambda P_n) \\ 0 &= \epsilon_n (z - \lambda I)(z - \delta P_n) \end{aligned}$$

Le déterminant de l'équation précédente donne donc le lieu des racines de l'estimateur : l'un à λI et l'autre à δP . Dans le plan complexe z , le deuxième lieu des racines se trouve dans le cercle unitaire proche de $(1, 0)$, en faisant des pôles très lents. En l'absence de perturbation à l'entrée du système, les pôles de l'estimateur tendent donc de λI vers cette section :

Par contre, en modifiant l'équation de la covariance, l'on se trouve aussi à changer le calcul du nouvel estimé θ_n . Forcer le terme $\delta(1 - \lambda)I$ sur la matrice de covariance signifie que la fonction de coût J utilisée est modifiée. On utilise donc maintenant :

$$J(\theta_n) = \sum_{k=1}^n \lambda^{n-k} (z_k - h_k^T \theta_n)^2 + \delta (\theta_n - \theta_{n-1})^2$$

Car sans le terme θ_{n-1} , la fonction de coût forcerait θ vers zéro

$$\begin{aligned}
\min J &\rightarrow \frac{\partial J}{\partial \theta} = 0 \\
0^\top &= \sum_{k=1}^n \lambda^{n-k} h_k (z_k - h_k^\top \theta_n) + \delta(\theta_n - \theta_{n-1}) \\
\left(\sum_{k=1}^n \lambda^{n-k} h_k h_k^\top + \delta I \right) \theta_n &= \sum_{k=1}^n \lambda^{n-k} h_k z_k + \delta \theta_{n-1} \\
P_n^{-1} &= \sum_{k=1}^n \lambda^{n-k} h_k h_k^\top + \delta I \\
&= \lambda \left(\sum_{k=1}^{n-1} \lambda^{n-1-k} h_k h_k^\top + \delta I \right) + h_n h_n^\top + \delta I - \lambda \delta I \\
&= \lambda P_{n-1}^{-1} + h_n h_n^\top + \delta(1 - \lambda) I \\
P_n^{-1} \theta_n &= \sum_{k=1}^n \lambda^{n-k} h_k z_k + \delta \theta_{n-1} \\
&= \lambda \left(\sum_{k=1}^{n-1} \lambda^{n-1-k} h_k z_k + \delta \theta_{n-2} \right) + h_n z_n + \delta \theta_{n-1} - \lambda \delta \theta_{n-2} \\
&= \lambda \left(P_{n-1}^{-1} \theta_{n-1} \right) + h_n z_n + \delta(\theta_{n-1} - \lambda \theta_{n-2}) \\
\theta_n &= P_n \left(P_n^{-1} - h_n h_n^\top - \delta(1 - \lambda) I \right) \theta_{n-1} + P_n h_n z_n + P_n \delta (\theta_{n-1} - \theta_{n-2}) \\
&= \theta_{n-1} + P_n h_n (z_n - h_n^\top \theta_{n-1}) + P_n \delta \lambda (\theta_{n-1} - \theta_{n-2}) \\
&= \theta_{n-1} + P_n h_n \epsilon_n + P_n \delta \lambda \tilde{\theta}_{n-1}
\end{aligned}$$