

REPUBLIQUE ALGERIENNE DEMOCRATIQUE ET POPULAIRE  
Ministère de l'Enseignement Supérieur et de la Recherche Scientifique  
Ecole Nationale Polytechnique



Département d'Automatique

Laboratoire de Commande des Processus

Mémoire de projet de fin d'études pour l'obtention du  
diplôme d'ingénieur d'état en Automatique

# Guidage adaptatif et commande FTC d'un mini drone de type quadri-rotor sous l'effet de perturbations

**Abdellah RAHMANI**

**Smail HOUTI**

Sous la direction de :

**Mr M. TADJINE Professeur**

**Mr T. SOUANEF Enseignant chercheur**

Présenté et soutenu publiquement le 24/06/2018

## Composition du Jury :

Président	Mr R. ILOUL	MCA	Ecole Nationale Polytechnique
Promoteurs	Mr M. TADJINE	Professeur	Ecole Nationale Polytechnique
	Mr T. SOUANEF	Enseignant chercheur	USTHB
Examineur	Mr H. CHEKIREB	Professeur	Ecole Nationale Polytechnique

**ENP 2018**



REPUBLIQUE ALGERIENNE DEMOCRATIQUE ET POPULAIRE  
Ministère de l'Enseignement Supérieur et de la Recherche Scientifique  
Ecole Nationale Polytechnique



Département d'Automatique

Laboratoire de Commande des Processus

Mémoire de projet de fin d'études pour l'obtention du  
diplôme d'ingénieur d'état en Automatique

# Guidage adaptatif et commande FTC d'un mini drone de type quadri-rotor sous l'effet de perturbations

**Abdellah RAHMANI**

**Smail HOUTI**

Sous la direction de :

**Mr M. TADJINE Professeur**

**Mr T. SOUANEF Enseignant chercheur**

Présenté et soutenu publiquement le 24/06/2018

## Composition du Jury :

Président	Mr R. ILOUL	MCA	Ecole Nationale Polytechnique
Promoteurs	Mr M. TADJINE	Professeur	Ecole Nationale Polytechnique
	Mr T. SOUANEF	Enseignant chercheur	USTHB
Examineur	Mr H. CHEKIREB	Professeur	Ecole Nationale Polytechnique

**ENP 2018**

# *Dédicace*

*Je dédie ce travail à,*

*Mes très chers parents, dont le sacrifice, la tendresse, l'amour, la patience, le soutien, et l'encouragement sont l'essentiel de ma réussite. Sans eux je ne serai pas à ce stade aujourd'hui,*

*Ma chère sœur pour son soutien et son aide continue durant mon parcours,*

*Ma grande famille,*

*A tous mes enseignants*

*Mon fidèle ami et binôme Smail, mon meilleur ami Diab, ainsi que tous mes camarades et toutes mes connaissances,*

*Abdellah*

# *Dédicace*

*Je dédie ce travail à mes chers parents, dont le sacrifice, la tendresse, l'amour, le soutien, l'aide et l'encouragement sont le secret de ma réussite*

*A mes frères et ma sœur*

*A ma grande famille*

*A mon binôme Abdellah pour son sérieux durant nos trois ans d'études en  
Automatique*

*A tous mes enseignants*

*A mes amis d'El-Biar*

*A toute personne très chère pour moi, qui a partagé ma joie et ma peine et  
m'a soutenu jusqu'au bout*

*Smail*

# *Remerciements*

*Nous remercions Dieu le tout puissant de nous avoir donné la force et le courage pour réaliser ce travail.*

*Ce travail a été effectué sous la direction de Monsieur M. Tadjine, professeur à l'Ecole Nationale Polytechnique, et Monsieur T. Souanef, enseignant chercheur à l'Université des Sciences et de la Technologie Houari-Boumediène, à qui nous tenons à exprimer nos vifs remerciements d'avoir suivi de très près ce travail et pour leurs aide et leurs précieux conseils tout au long de ce travail.*

*Nous remercions les membres du jury pour l'honneur qu'ils nous ont fait en acceptant d'examiner notre travail.*

*Et enfin, nous tenons à remercier Mlle N. GABOUR pour son aide à la réalisation de ce travail ainsi que toute personne ayant contribué à notre réussite tout au long de notre parcours d'étude.*

## ملخص:

يتعلق هذا العمل بتصميم قانون يسمح بالتعامل مع الاضطرابات والشكوك وأخطاء النمذجة والأعطاب التي يمكن أن تؤثر على أداء نظام الطائرات بدون طيار رباعية المروحيات. في البداية نقوم بعرض النتائج المتحصل عليها بعد تطبيق احدى طرق التحكم  $\mathcal{L}_1$  التكيفي و من ثم نقوم باستخراج نقاط ضعف هذه الطريقة لاقتراح قانون تحكم تكيفي  $\mathcal{L}_1$  معتمد على التحكم الإنزلاقي لمعالجة المشكلات التي تمت مواجهتها ثم نضيف نظام يسمح بالتعامل مع الاعطاب FTC التي يمكن ان تلحق الجهاز. و في الاخير, نقوم باجراء عمليات محاكاة واقعية باستعمال النظام ROS/GAZEBO

*الكلمات المفتاحية:* طائرة بدون طيار رباعية المروحيات ، التحكم  $\mathcal{L}_1$  التكيفي ، التحكم الإنزلاقي ، FTC ، ROS ، Gazebo

## Abstract:

This graduation project concerns the design of a command in order to deal with the disturbances, uncertainties, modeling errors and faults that can affect the performance of a quadri-rotor drone system. In a first place, we model the system and synthesize an  $\mathcal{L}_1$  adaptive control law based on the gradient descent, then we identify the weaknesses of the latter and propose a sliding mode  $\mathcal{L}_1$  adaptive control to remedy the problems we have encountered, and then we add to this approach a fault tolerant control to make the system robust against sensor, actuator or component faults, and finally, we will introduce the ROS / Gazebo simulator to perform realistic simulations.

**Keywords:** drone, quadri-rotor ,  $\mathcal{L}_1$  adaptive control, sliding modes, FTC, ROS, Gazebo

## Résumé :

Ce projet de fin d'études concerne la conception d'une commande permettant de faire face aux perturbations, aux incertitudes, aux erreurs de modélisation et aux défauts qui peuvent affecter les performances d'un système de drone de type quadri-rotor . Dans un premier lieu, nous modélisons le système et nous synthétisons une commande  $\mathcal{L}_1$  adaptative avec la descente du gradient, ensuite, nous relevons les points faibles de celle-ci et proposons une commande  $\mathcal{L}_1$  adaptative par modes glissants pour remédier aux problèmes rencontrés, et puis nous rajoutons à cette approche une commande tolérante aux défauts afin de rendre le système robuste vis-à-vis des défauts capteurs, actionneurs ou composants, et enfin, nous introduirons le simulateur ROS/Gazebo afin d'effectuer des simulations réalistes.

**Mots clés :** drone, quadri-rotor , commande  $\mathcal{L}_1$  adaptative, modes glissants, FTC, ROS, Gazebo

## Table des Matières :

### LISTE DES ILLUSTRATIONS

### LISTE DES ABREVIATIONS

### LISTE DES SYMBOLES

<b>Introduction Générale.....</b>	<b>14</b>
<b>Chapitre1. Généralités .....</b>	<b>17</b>
1.1 Introduction .....	18
1.2 Généralités sur les quadri-rotor s .....	18
1.2.1 Description du quadri-rotor .....	18
1.2.2 Avantages de la configuration quadri-rotor .....	19
1.3 Inconvénients de la configuration quadri-rotor : problématique .....	20
1.4 Principales méthodes de la commande des quadri-rotor s .....	20
1.5 Objectifs .....	22
1.6 La commande adaptative $\mathcal{L}1$ .....	23
1.7 Contribution du mémoire .....	24
1.8 Conclusion.....	24
<b>Chapitre 2. Modélisation du quadri-rotor .....</b>	<b>25</b>
2.1 Introduction .....	26
2.2 Repérage du quadri-rotor dans l'espace .....	26
2.3 Modèle dynamique du quadri-rotor .....	27
2.3.1 La dynamique de translation et de rotation.....	27
2.3.2 La relation entre les angles d'Euler et les vitesses angulaires .....	29
2.3.3 La dynamique des rotors.....	30
2.3.4 Modèle d'état de quadri-rotor .....	31
2.4 Conclusion.....	32
<b>Chapitre 3. La commande adaptative et la commande tolérante aux défauts (FTC) .....</b>	<b>34</b>
3.1 Introduction .....	35
3.2 La commande adaptative.....	35
3.3 Commande $\mathcal{L}1$ adaptative d'un système MIMO.....	36
3.3.1 Formulation du problème .....	36

3.3.2	Les hypothèses .....	36
3.3.3	Architecture de la commande.....	37
3.4	La commande $\mathcal{L}1$ avec la descente du gradient.....	37
3.4.1	Les lois d'adaptation .....	37
3.4.2	La loi de commande .....	38
3.4.3	Application au quadri-rotor .....	38
3.5	La commande $\mathcal{L}1$ adaptative avec modes glissants .....	41
3.5.1	La loi de commande .....	43
3.5.2	Application au quadri-rotor .....	43
3.6	Commande tolérante aux défauts .....	44
3.6.1	Défauts et défaillances.....	44
3.6.2	Classification des défauts .....	44
3.6.2.1	Classification selon la localisation .....	44
3.6.2.2	Classification selon la modélisation .....	45
3.6.2.3	Classification selon les caractéristiques temporelles .....	45
3.6.3	Classification des systèmes FTC.....	46
3.6.4	Commande $\mathcal{L}1$ adaptative tolérante aux défauts d'actionneurs .....	48
3.6.4.1	Formulation du problème .....	48
3.6.4.2	La loi de commande .....	48
3.6.5	Commande $\mathcal{L}1$ adaptative tolérante aux défauts actionneurs et capteurs.....	49
3.6.5.1	Formulation du problème .....	49
3.6.5.2	La loi de commande .....	50
3.7	Conclusion.....	51
<b>Chapitre 4. Implémentation de la commande <math>\mathcal{L}1</math> du Quadri-rotor sur l'environnement ROS.....</b>		<b>52</b>
4.1	Introduction .....	53
4.2	Robot Operating System (ROS).....	53
4.2.1	Les outils de ROS.....	53
4.2.2	La communication sous ROS .....	54
4.2.3	Hector_quadrotor .....	55
4.3	Robotics System Toolbox.....	56
4.3.1	Connexion Simulink/ ROS .....	56
4.3.2	Echange de données entre Simulink et ROS.....	57

4.3.3	Générer un nœud ROS à partir de Simulink.....	60
4.4	Conclusion.....	61
<b>Chapitre 5. Simulations et résultats.....</b>		<b>62</b>
5.1	Introduction .....	63
5.2	Résultats des simulations avec Descente du gradient .....	63
5.2.1	Simulations en l'absence des perturbations atmosphériques .....	63
5.2.2	Simulations sous l'effet du vent .....	65
5.2.3	La limite de la commande $\mathcal{L}1$ avec la descente du gradient .....	69
5.3	Résultats des simulations avec Modes Glissants.....	70
5.3.1	Simulations en l'absence des perturbations atmosphériques .....	70
5.3.2	Simulation sous l'effet du vent .....	73
5.4	Résultats des simulations de la commande $\mathcal{L}1$ adaptative tolérante aux défauts actionneurs .....	75
5.4.1	Simulations en l'absence des perturbations atmosphériques .....	76
5.4.2	Simulations sous l'effet du vent .....	77
5.5	Simulations de la commande $\mathcal{L}1$ adaptative tolérante aux défauts actionneurs et capteurs.....	78
5.5.1	Simulations en l'absence des perturbations atmosphériques .....	78
5.5.2	Simulation sous l'effet du vent .....	80
5.6	Simulations sous l'environnement Gazebo.....	83
5.7	Conclusion.....	87
<b>Conclusion générale .....</b>		<b>89</b>
<b>Bibliographie.....</b>		<b>91</b>
<b>Annexes.....</b>		<b>96</b>

## LISTE DES ILLUSTRATIONS :

Figure 1. 1 : Le quadri-rotor .....	18
Figure 1. 2 : Contrôle du quadri-rotor .....	19
Figure 1. 3: Architecture générale de la commande adaptative $\mathcal{L}1$ .....	23
Figure 2. 1 : Les repères utilisés.....	27
Figure 3. 1 : L'architecture de la commande L1. ....	43
Figure 3. 2 : Défauts additif et multiplicatif.....	45
Figure 3. 3 : Classification des FTC.....	47
Figure 3. 4 : Le schéma de la commande FTC .....	47
Figure 3. 5: Architecture de commande FTC capteurs.....	51
Figure 4. 1 : Schéma de fonctionnement du master sous ROS.....	55
Figure 4. 2 : Adresse IP de la machine virtuelle Linux .....	56
Figure 4. 3 : Interaction entre Simulink et Gazebo.....	58
Figure 4. 4: La publication de donnée sur ROS .....	59
Figure 4. 5: Abonnement à un topic ROS. ....	59
Figure 4. 6: Configuration du modèle Simulink.....	60
Figure 5. 1: Les réponses du système et du prédicteur.....	63
Figure 5. 2: Les erreurs de poursuite.....	64
Figure 5. 3: Les commandes appliquées.....	64
Figure 5. 4 : La trajectoire en 3D. ....	65
Figure 5. 5: Modèle du vent. ....	66
Figure 5. 6: Vitesse du vent.....	66
Figure 5. 7: Les réponses du système et du prédicteur.....	67
Figure 5. 8: Les erreurs de poursuite.....	67
Figure 5. 9 : Les commandes appliquées.....	68
Figure 5. 10: La trajectoire en 3D. ....	68
Figure 5. 11: Les réponses du système et du prédicteur.....	69
Figure 5. 12: Les erreurs de poursuites. ....	70
Figure 5. 13: Les réponses du système et du prédicteur.....	71
Figure 5. 14: Les erreurs de poursuite.....	71
Figure 5. 15: Les commandes appliquées.....	72
Figure 5. 16: La trajectoire en 3D. ....	72
Figure 5. 17: Vitesse du vent.....	73
Figure 5. 18: Les réponses du système et du prédicteur.....	73
Figure 5. 19: Les erreurs de poursuite.....	74
Figure 5. 20: Les commandes appliquées.....	74

## LISTE DES ILLUSTRATIONS

Figure 5. 21: La trajectoire en 3D. ....	75
Figure 5. 22: Les sorties du système. ....	76
Figure 5. 23: La commande u1. ....	76
Figure 5. 24: Les sorties du système sous l'effet du vent. ....	77
Figure 5. 25: La commande u1 ....	77
Figure 5. 26: Les sorties réelles et mesurées ....	78
Figure 5. 27: L'estimation du défaut des capteurs.....	79
Figure 5. 28: Les commandes appliquées.....	80
Figure 5. 29: Les sorties réelles et mesurées ....	81
Figure 5. 30: L'estimation du défaut des capteurs.....	82
Figure 5. 31: Les commandes appliquées.....	82
Figure 5. 32 : Positions de départ et d'arrivée du quadri-rotor ....	83
Figure 5. 33 : Réponses de l'Hector Quadrotor et du prédicteur.....	84
Figure 5. 34 : Les erreurs de poursuite. ....	84
Figure 5. 35 : Les signaux de commande. ....	85
Figure 5. 36 : La trajectoire 3D du quadri-rotor ....	85
Figure 5. 37 : Les retards de simulation. ....	87

## LISTE DES ABREVIATIONS :

AAP :	Algorithme d'Adaptation Paramétrique
AFS :	Académie Française des Sciences
AFTCS:	Active Fault Tolerant Control Systems
EKF :	Extended Kalman Filter
ESC :	Electronic Speed Controller
FDD:	Fault Detection and Diagnosis,
FDI:	Fault Detection and Isolation
FE :	Fault Estimation
FTC :	Fault Tolerant Control
GPS :	Global Positioning System
ILC :	Iterative Learning Control
IMU :	Inertial Measurement Unit
IP:	Internet Protocol
LBMPC:	Learning-Based Model Predictive Control
LMI:	Linear Matricial Inequality
<i>LQ :</i>	<i>Linéaire Quadratique</i>
LQR:	Linear Quadratic Regulator
MLI :	Modulation de Largeur d'Impulsion
MPC :	Mode Predictive Controller
MRAC :	Model Refrence Adaptive Control
PD :	<i>Proportionnelle Dérivée</i>
PFTC:	Passive Fault Tolerant Control
<i>PID :</i>	<i>Proportionnelle Intégrale Dérivée</i>
RC :	Radio Control
RM :	Reconfiguration Mechanism
ROS :	Robot Operating System
SISO :	Single Input Single Output
TLC:	Trajectory Linearisation Control
UAV :	Unmanned Aerial Vehicle
URDF :	Universal Robotic Description Format
VTOL :	Vertical Take Off and Landing

## LISTE DES SYMBOLES :

$\hat{L}_u$ :	Estimation de la borne des perturbations
$B_m, B_{um}$ :	Matrices constantes connues
$\hat{\theta}_i, \hat{\omega}_i, \hat{\eta}_i$ :	Estimations des paramètres inconnus
$\bar{\omega}_i$ :	Vitesse angulaire du rotor $i$
$\mathcal{M}_g$ :	Moment gyroscopique
$\delta_i(t)$ :	Coefficient d'efficacité du $i^{\text{ème}}$ actionneur
$A_m$ :	La matrice dynamique en boucle fermée
$C_v$ :	Coefficient du frottement visqueux
$C_\theta, S_\theta$ :	Cosinus et Sinus de l'angle $\theta$
$F_g$ :	Le poids dans l'axe du corps
$F_p$ :	Force de portance
$J_r$ :	Inertie des rotors
$K_d$ :	Matrice gain.
$K_s$ :	Matrice gain
$L_m, L_u$ :	Constantes réelles
$O_0$ :	Origine du repère $R_0$
$O_1$ :	Origine du repère $R_1$
$P_0$ :	Point d'équilibre
$R_0$ :	Repère lié au sol
$R_1$ :	Repère dont l'origine coïncide avec le centre de gravité du quadri-rotor .
$T_s$ :	Constante de temps de chaque capteur.
$b_{s_i}$ :	Biais, la dérive et la perte de précision du capteur.
$\hat{f}(t)$ :	Estimation de l'erreur de mesure.
$f_i$ :	Force de portance produite par la rotation de l'hélice $i$
$f_i(x, u, t)$ :	Fonction non linéaire du $i^{\text{ème}}$ défaut
$k_{a_i}$ :	Coefficient d'efficacité
$k_{s_i}$ :	Perte d'efficacité (erreur de calibrage) du capteur, modélise
$r_g$ :	Pré-compensateur
$t_{F_i}$ :	Instant d'occurrence du défaut du $i^{\text{ème}}$ actionneur.
$u_{c_i}(t)$ :	Signal de commande appliqué au $i^{\text{ème}}$ actionneur
$u_{i_{min}}, u_{i_{max}}$ :	Limites inférieure et supérieure du signal de sortie du $i^{\text{ème}}$ actionneur.
$u_{ad}(t)$ :	Commande adaptative.
$u_c$ :	Commande calculée.
$u_i(t)$ :	Signal de sortie de l'actionneur
$u_m(t)$ :	Commande en retour d'état.
$\hat{x}$ :	Etat prédit
$\tilde{x}$ :	Erreur d'estimation
$x_s$ :	Signaux des capteurs
$\alpha_i$ :	Constante positive qui représente le taux d'évolution du défaut
$\beta_i(t - t_{F_i})$ :	Profil temporel du défaut qui se produit à un instant inconnu $t_{F_i}$ ,

## LISTE DES SYMBOLES

$\delta_i(x, u)$ :	Changement de la dynamique du système au $i^{\text{ème}}$ défaut.
$\eta_m$ :	Perturbations en entrée
$\eta_u$ :	Erreur de modélisation et perturbation
$\tau_i$ :	Couple moteur
$\tau_{visq}$ :	Couple visqueux
$\dot{\phi}, \dot{\theta}, \dot{\psi}$ :	Variations des angles d'Euler
$G$ :	Centre de gravité du quadri-rotor
$\mathcal{M}$ :	Vecteur des moments
$R$ :	Matrice de passage
$V$ :	Vecteur des vitesses linéaires
$X, Y, Z$ :	Position du quadri-rotor
$\Gamma > 0$ :	Gain d'adaptation
$\Theta, \Omega$ :	Ensembles compacts convexes connus
$A$ :	La matrice dynamique
$B$ :	La matrice d'entrée du système d'état.
$C$ :	La matrice de sortie du système d'état
$C(s)$ :	Filtre stable strictement propre
$D(s)$ :	Matrice de transfert strictement propre
$F$ :	Ensemble des forces extérieures
$F(s)$ :	Matrice de transfert des capteurs
$J$ :	La matrice d'inertie
$L, M, N$ :	Moments agissant sur le centre de masse
$P$ :	Solution de l'équation algébrique de Lyapunov
$Q, R$ :	Matrices de pondération positives
$U, V, W$ :	Vitesses linéaires
$X$ :	Vecteur d'état
$f$ :	Erreur de mesure.
$k$ :	Le coefficient de trainée
$l$ :	La distance entre le centre de gravité du quadri-rotor et l'axe du rotor
$p, q, r$ :	Vitesses angulaires
$u(t)$ :	Commande appliquée au système
$\alpha$ :	Nombre positif arbitraire.
$\lambda$ :	Matrice constante arbitraire
$\sigma(t)$ :	Surface de glissement
$\omega$ :	Vecteur des vitesses angulaires
$\omega, \theta$ :	Matrices des paramètres
$\phi, \theta, \psi$ :	Angles d'Euler (Roulis, tangage, lacet)

# **Introduction Générale**

Le progrès de l'électronique embarquée a permis la miniaturisation des composants électroniques et l'amélioration des techniques de communication sans fil rendant ainsi la technologie UAV abordable pour un usage général. En effet, une myriade de prototypes, en particulier les UAV de petite taille, ont été construits au cours des deux dernières décennies.

Les drones, souvent appelés UAVs (de l'anglais : Unmanned Aerial Vehicles) deviennent omniprésents dans les secteurs militaires, industriels et universitaire. Ils ont été utilisés pour la première fois dans des applications militaires comme avions pilotés à distance afin de remplacer l'homme dans les missions risquées telles que la reconnaissance vidéo et les attaques aériennes. Mais de nos jours, ils sont de plus en plus populaires et mis en vente pour des applications civiles utilisant des configurations multi-rotors facilement déployables pour diverses missions telles que [6] la prise de vue aérienne, l'inspection des zones agricoles, des réseaux électriques, des chemins de fer, etc.

Dans ce travail, on s'intéresse à une configuration particulière des drones multi-rotors, notamment, les quadri-rotors. Ces derniers sont des véhicules aériens à décollage et atterrissage verticaux (VTOL) propulsés par quatre rotors [1]. Ils sont peu coûteux et mécaniquement simples ce qui permet leur construction à grande quantité. Leur habilité au décollage et à l'atterrissage vertical, au vol stationnaire et leur petite taille leur permet d'effectuer des vols dans des environnements restreints, et leur énergie cinétique réduite répartie sur les quatre rotors les rend plus sûrs à manipuler [1], [4].

Cependant, le guidage et la commande de ces engins constitue un grand défi en raison de leur dynamique inconnue et fortement non linéaire ainsi que leur nature instable et sous actionnée, de plus ; leur faible poids et leurs vitesses relativement basses les rendent extrêmement sensibles aux perturbations atmosphériques [6],[7], par conséquent ; la poursuite de trajectoire devient une tâche compliquée. En outre, la plupart des quadri-rotors sont conçus avec des composants à bas coût, ce qui augmente leur probabilité de défaillance ou de pannes, la chose qui n'est pas admissible surtout quand ces appareils survolent des zones urbaines.

On devra donc prendre en compte - en phase de conception des lois de commande - la qualité souvent médiocre des modèles disponibles pour synthétiser les correcteurs ainsi que les variations paramétriques qui peuvent être importantes. La théorie de la commande robuste nous apporte des réponses à ces problèmes, cependant ; on se retrouve face à la complexité parfois élevée des correcteurs et les problèmes de compromis performance/robustesse qui constituent une limitation de la commande robuste.

L'objectif de la commande robuste est de rendre la boucle fermée insensible aux perturbations et aux incertitudes liées à la modélisation, la commande adaptative a pour but de réduire les conséquences de ces dernières à travers leur identification ou l'identification d'un correcteur qui annule leurs effets [32], cependant ; elle est parfois limitée par des gains d'adaptation faibles pour des raisons de robustesse.

La commande  $\mathcal{L}_1$  adaptative qui fait l'objet de notre travail semble un choix pertinent pour repousser les limitations du compromis entre vitesse d'adaptation et robustesse, en effet, elle vient améliorer la commande MRAC en introduisant un filtre passe-bas permettant de découpler la vitesse d'adaptation et la robustesse [12]. Par conséquent, elle permet une adaptation rapide sans perdre en robustesse.

Ce manuscrit est composé de cinq chapitres organisés comme suit :

**Chapitre 1 – Généralités** : Ce premier chapitre donne un aperçu général sur les quadri-rotors, leurs avantages et inconvénients, il fait aussi un état de l'art sur les différentes techniques de commande des quadri-rotors, et se termine par une introduction à la commande  $\mathcal{L}_1$  adaptative utilisée dans ce travail.

**Chapitre 2 - Modélisation du quadri-rotor** : Ce deuxième chapitre est consacré à la modélisation du quadri-rotor ; il donne une description du quadri-rotor et ses mouvements de base, ensuite, il établit son modèle dynamique avec les hypothèses de départ, et enfin, il donne la représentation d'état du quadri-rotor .

**Chapitre 3 - La commande adaptative et La commande tolérante aux défauts (FTC)** : Le troisième chapitre est dédié à l'étude de deux approches de commande adaptative, une commande  $\mathcal{L}_1$  adaptative classique basée sur la descente du gradient, et puis nous proposons une nouvelle approche  $\mathcal{L}_1$  adaptative basée sur les modes glissent. Ensuite, les notions de base de la commande tolérante aux défauts, les différents types de défauts ainsi que les principales stratégies de la commande FTC sont présentées, et enfin, une application de la commande FTC avec la commande  $\mathcal{L}_1$  adaptative.

**Chapitre 4 – Implémentation de la commande L1 du Quadri-rotor sur l'environnement ROS** : Dans ce chapitre, nous présentons une méthode simple pour générer à partir d'un modèle Simulink ; un nœud ROS afin d'implémenter la commande  $\mathcal{L}_1$  sur l'environnement Gazebo.

**Chapitre 5 – Simulations et résultats** : Dans ce dernier chapitre, des résultats des simulations et des tests en l'absence/ présence des perturbations atmosphériques sont présentés et interprétés pour les deux approches avec descente du gradient et modes glissants ainsi que pour la commande FTC, et enfin, des simulations sous un environnement plus proche de la réalité ( Gazebo) sont présentées.

# **Chapitre 1.**

# **Généralités**

## 1.1 Introduction

Les drones constituent ces dernière années un domaine d'études en constante évolution grâce à l'avancement des connaissances dans le domaine de l'aéronautique et au développement technologique qui a donné naissance à des capteurs de plus en plus précis, et qui a amélioré la capacité de calcul des processeurs. Le quadri-rotor est un type de drones qui connaît un grand essor dans le monde de la recherche car il offre beaucoup d'avantages, ce qui permet de l'utiliser dans des applications différentes.

Dans ce premier chapitre, nous présenterons de façon générale les quadri-rotors, les avantages qu'ils offrent ainsi que leurs inconvénients, on fera aussi un revu de littératures sur les différentes techniques de commande des quadri-rotors, et enfin on introduit la technique de commande  $\mathcal{L}_1$  adaptative utilisée dans ce travail.

## 1.2 Généralités sur les quadri-rotors

### 1.2.1 Description du quadri-rotor

Le quadri-rotor est un véhicule aérien à décollage et atterrissage verticaux VTOL propulsé par quatre rotors. Il a deux paires de rotors contrarotatifs, le premier et le troisième rotor tournent dans le sens des aiguilles d'une montre tandis que le second et le quatrième rotor tournent dans le sens opposé afin de compenser le couple est généré par le mouvement de rotation des rotors autour de leurs axes (Figure 1.1). La force de poussée est contrôlée en modifiant la vitesse de rotation des rotors [1].

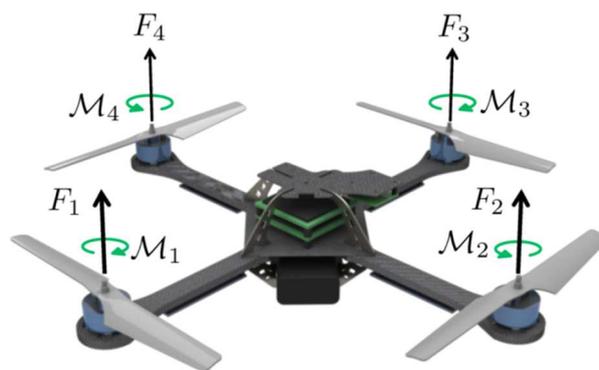


Figure 1. 1 : Le quadri-rotor (source: Journal of Mechanisms and Robotics)

En vol stationnaire ou en décollage/atterrissage vertical (figure 1.2 (a)), tous les rotors tournent à la même vitesse et en raison de la nature contrarotative des deux paires de rotors et la symétrie du quadri-rotor autour de son centre de masse, le couple net du quadri-rotor est nul, aucun mouvement de lacet n'est donc généré. Le contrôle du mouvement de lacet est réalisé par la différence des couples générés par une paire de rotors du même axe et les

couples générés par la paire de rotors restante. Par exemple, dans la figure 1.2 (c), un mouvement de lacet dans le sens antihoraire est généré en augmentant la vitesse de rotation du premier et du troisième rotor tout en diminuant les vitesses de rotation du second et du quatrième rotor [1].

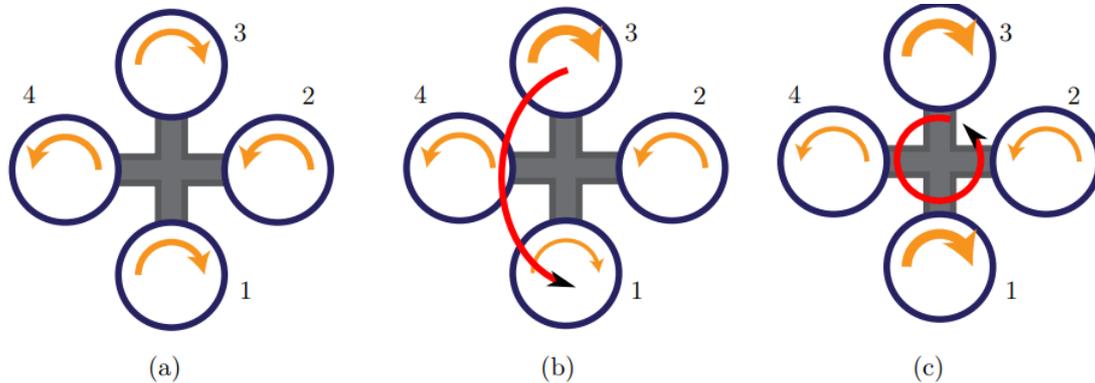


Figure 1. 2 : Contrôle du quadri-rotor . L'épaisseur de la flèche jaune est proportionnelle à la vitesse de rotation du moteur. La flèche rouge indique le mouvement du quadri-rotor [1]

Quant aux mouvements de roulis et de tangage, ils peuvent être effectués en augmentant la vitesse d'un rotor et en diminuant celle du rotor du même axe. Par exemple, dans la figure 1.6 (b), un mouvement de tangage vers l'avant est généré en augmentant la vitesse de rotation du troisième rotor tout en diminuant celle du premier rotor [1].

### 1.2.2 Avantages de la configuration quadri-rotor

Parmi les avantages des quadri-rotors, on peut citer [1], [4]:

- Les quadri-rotors sont mécaniquement moins complexes que les hélicoptères (absence du moyeu) ce qui simplifie leur conception ainsi que leur maintenance.
- Les quadri-rotors sont peu coûteux et peuvent être construits en grande quantité.
- L'utilisation de quatre rotors de petit diamètre réduit l'énergie cinétique des quadri-rotors par conséquent, ils sont plus sûrs à manipuler que les hélicoptères.
- Possibilité de décollage et d'atterrissage verticaux, vol stationnaire et leur petite taille leur permet d'effectuer des vols dans des environnements restreints.
- Effet gyroscopique réduit du fait que deux rotors tournent dans le sens des aiguilles d'une montre et les deux autres dans le sens contraire.
- Simplicité de la commande, en effet, le contrôle du quadri-rotor se fait en variant la vitesse de rotation des différents moteurs.

### 1.3 Inconvénients de la configuration quadri-rotor : problématique

Le drone de type quadri-rotor dont l'étude fait l'objet de ce mémoire, est un aéronef faisant partie de la famille des multi-rotors, il s'agit d'un appareil volant qui possède quatre rotors en forme de croix, commandé par la variation des vitesses angulaires de ces derniers, ce qui simplifie grandement sa conception en éliminant les pièces mécaniques complexes. Cependant, le guidage et la commande de ces engins constitue un grand défis en raison de leur dynamiques inconnues, nature instable et sous actionnée, en effet ; le quadri-rotor standard possède quatre entrées de commande indépendantes (vitesses des quatre rotors) contre six degrés de liberté paramétrant sa position et son orientation dans l'espace. De plus ; leurs faible poids et leurs vitesses relativement basses les rendent extrêmement sensibles aux perturbations atmosphériques [6], [7], par conséquent ; la poursuite de trajectoire devient une tâche compliquée. En outre, la plupart des quadri-rotor s sont conçu avec des composants à bas coût, ce qui augmente leur probabilité de défaillance ou de pannes, la chose qui n'est pas admissible surtout quand ces appareils survolent des zones urbaines, pour cette raison ; une commande tolérante aux défauts devient nécessaire.

### 1.4 Principales méthodes de la commande des quadri-rotor s

Les architectures de commandes utilisées pour les drones opérationnels ne sont pas documentées et les plateformes utilisées dans le domaine académique utilisent souvent des ressources open-source. Les micro-drones sont souvent utilisés par les laboratoires de recherche comme plateforme d'essai pour les architectures de commandes les plus variées. Si on prend l'exemple des quadri-rotor s, on peut établir la liste, non exhaustive, suivante :

#### **Commande PID [4] :**

Méthode de commande classique qui repose sur un modèle dynamique simplifié, les effets gyroscopiques dus à la rotation des hélices et à la rigidité du châssis sont négligés. Le régulateur PID a été implémenté sur le système réel (banc d'essais) et a permis la stabilisation en attitude et un bon contrôle sur l'angle de lacet. Un vol autonome a été réalisé par la suite avec succès, ce qui confirme la possibilité de commander le quadri-rotor avec cette approche classique.

#### **Commande LQ [4] :**

Technique de commande moderne, basée sur un modèle dynamique plus complet que celui de l'approche PID. Afin d'assurer la globalité de cette méthode, la linéarisation se fait autour de chaque point de fonctionnement mesuré. Après avoir réalisé des simulations sur un banc d'essais, les auteurs soulignent que cette approche donne des résultats moyens, en effet ; il y avait une erreur statique à cause des imperfections du model : les dynamiques des actionneurs n'étaient pas prises en compte, aussi ; ils disent qu'une action intégrale pourrait améliorer les résultats.

### **Backstepping :**

Cette technique a permis dans [8] de concevoir une commande non linéaire en prenant en compte les couples gyroscopiques et les couplages entre les axes mais le modèle ne fait pas intervenir les efforts aérodynamiques. Plusieurs simulations ont été réalisées et ont montré que les performances étaient satisfaisantes. Des tests réels en été effectués sur un banc d'essais, et les résultats qui en découlent confirment ceux des simulations, en effet ; l'attitude du quadri-rotor a été stabilisée même en présence de perturbations relativement élevées. Zuo a utilisé la technique de commande filtered Backsteeping pour stabiliser l'attitude du quadri-rotor . Le régulateur de suivi de trajectoire était basé sur la méthode PD linear feedback pour diviser les signaux de commande d'attitude [28]. Madani et Benallegue ont résolu aussi le problème de suivi de trajectoire via l'approche de Backsteeping d'état complet, cela a été fait en divisant la dynamique du quadri-rotor en trois sous-systèmes et en utilisant la même méthode de conception pour chaque sous-système [27].

### **Sliding-mode :**

Les auteurs dans [8] avaient pour objectif de stabiliser l'attitude du quadri-rotor (l'altitude étant fixée par le bac d'essai), les résultats de simulation suivis par des tests réels montrent que la commande permet de stabiliser les angles de roulis et de tangage, mais les effets du shattering affectent les mesures, celles de l'angle de lacet en particulier. Cela est dû en partie à la nature de commutation du contrôleur qui introduit des oscillations de haute fréquence et de faible amplitude provoquant ainsi la dérive du capteur. Dans [29] une stratégie de commande basée sur les modes glissants et la commande adaptative a été proposée pour faire face aux variations des perturbations du vent.

### **Commande adaptative basée sur les réseaux de neurones [9] :**

C'est une nouvelle méthode permettant de stabiliser le quadri-rotor contre les perturbations ainsi que les incertitudes sur les paramètres du système, cette méthode est basée sur l'utilisation d'un (Cerebellar Model Articulation Controller) ; un réseau de neurones dans lequel les techniques de contrôle neural-adaptatifs sont utilisées pour approximer les poids. Les résultats de simulation montrent la stabilité et la performance de cette méthode, elle permet de minimiser l'erreur et d'atteindre l'attitude souhaitée tout en réduisant également la dérive du poids lorsque de grandes perturbations sont présentes.

### **Commande prédictive [10] :**

Les détails sur la mise en œuvre en temps réel de la commande LBMPC (learning-based model predictive control) sont présentés, cette approche repose sur l'apprentissage statistique ainsi que les techniques de commande pour tirer les niveaux de robustesse et de convergence. Une architecture utilisant un filtre de Kalman étendu (EKF) pour estimer l'état ainsi que les paramètres du système est présentée. Les résultats expérimentaux montrent que cette approche améliore la réponse en régime transitoire.

### **MRAC [11] :**

Une commande adaptative par modèle de référence MRAC basée sur le théorème de Lyapounov a été appliquée dans [11] sur un quadri-rotor afin de faire face aux incertitudes et aux variations des moments d'inertie. Un modèle dynamique à base de quaternions a été utilisé durant les simulations qui ont abouti à des résultats satisfaisants, en effet ; les oscillations dues aux variations des moments d'inertie ont disparu, et l'auteur souligne que cette technique serait utile dans le cas où la charge du quadri-rotor est variable.

### **Trajectory Linearisation Control [26] :**

B. Zhu et W. Huo ont divisé le modèle du quadri-rotor en quatre sous-systèmes : cinématique de position, dynamique de position, cinématique d'attitude et dynamique d'attitude. Le régulateur de chaque système a été conçu avec la méthode TLC (Trajectory Linearisation Control).

### **Mode Predictive Controller [22] :**

Raffo et al. Ont réalisé un suivi de trajectoire en utilisant un schéma composé d'un régulateur MPC (Mode Predictive Controller) pour suivre la trajectoire de référence avec un régulateur  $H_\infty$  non-linéaire pour stabiliser les mouvements de rotation.

## **1.5 Objectifs**

La nature et comportement non-linéaire, la complexité des dynamiques du quadri-rotor et l'incertitude de son modèle qui a été fait en partant d'hypothèses qui peuvent changer en pratique et les perturbations stochastiques du vent nous poussent à utiliser la commande adaptative qui répond aux exigences de commande dans des situations pareilles.

Le principal objectif de ce travail est d'utiliser les outils modernes de la modélisation et de la simulation des drones quadri-rotor afin de développer une nouvelle approche de commande  $\mathcal{L}_1$  adaptative tolérante aux défauts. Les deux principaux aspects de ce projet sont la conception d'une nouvelle approche de commande, notamment la commande  $\mathcal{L}_1$  adaptative avec modes glissants en partant de l'approche basée sur la descente du gradient, et puis de lui associer une commande tolérante aux défauts.

La première étape consiste à reproduire la commande  $\mathcal{L}_1$  adaptative basée sur la descente du gradient afin de relever les faiblesses de cette approche, particulièrement ; les bornes de perturbations. La deuxième étape est de concevoir un contrôleur permettant de faire face aux problèmes relevés, et la commande de base par modes glissants semble un choix pertinent pour remédier ce genre de problèmes.

Ensuite, vient la partie où l'on associe à la commande  $\mathcal{L}_1$  par modes glissant, une commande tolérante aux défauts afin de faire face à certaines pannes qui peuvent atteindre

notre système. Et le travail s'achève par des simulations sous l'environnement Gazebo afin de tester notre commande dans un environnement réaliste. Lorsque le thème de ce travail a été défini; ayant très peu de connaissances préalables de l'environnement ROS, le principal problème rencontré dans cette phase était la complexité du schéma de la commande  $\mathcal{L}_1$ , ce qui rend difficile sa programmation sous ROS, c'est pourquoi nous avons opté pour une méthode novatrice permettant d'effectuer les simulations dans ROS directement à partir de Simulink.

## 1.6 La commande adaptative $\mathcal{L}_1$

La commande adaptative  $\mathcal{L}_1$  est une méthode qui a attiré beaucoup d'attention ces dernières années en raison de sa robustesse et sa capacité à fournir de hautes performances, à la fois en régime transitoire et en régime permanent et même en présence d'incertitudes, dynamiques inconnues ou perturbations. Le principe de la commande  $\mathcal{L}_1$  a été introduit en 2006 par Cao et Hovakimyan, et depuis, différentes commandes basées sur l'architecture  $\mathcal{L}_1$  adaptative ont fait l'objet de nombreuses publications [12].

Cette technique de commande est une version améliorée de la commande adaptative par modèle de référence MRAC avec prédicteur d'état. En effet ; la méthode adaptative  $\mathcal{L}_1$  introduit un filtre passe-bas dans la loi de commande ce qui permet de découpler la vitesse d'adaptation et la robustesse. Par conséquent, la commande  $\mathcal{L}_1$  permet une adaptation rapide sans perdre en robustesse, ce qui n'est pas le cas pour la commande MRAC. Le choix de la structure du filtre ainsi que ses paramètres joue un rôle primordial dans la performance de la commande. Cao et Hovakimyan ont conçu la commande pour un système SISO en considérant les paramètres inconnus et variant dans le temps avec une incertitude constante, et des perturbations bornées, de plus ; ils ont discuté la marge de stabilité de l'architecture  $\mathcal{L}_1$ . Comme le montre la Figure 1.9, l'architecture de commande  $\mathcal{L}_1$  adaptative se compose de trois blocs: le prédicteur d'état, la loi d'adaptation paramétrique, et la loi de commande avec un filtre passe-bas. La loi d'adaptation est utilisée pour mettre à jour les paramètres du système de manière à minimiser l'erreur entre l'état prédit et l'état réel du système [12].

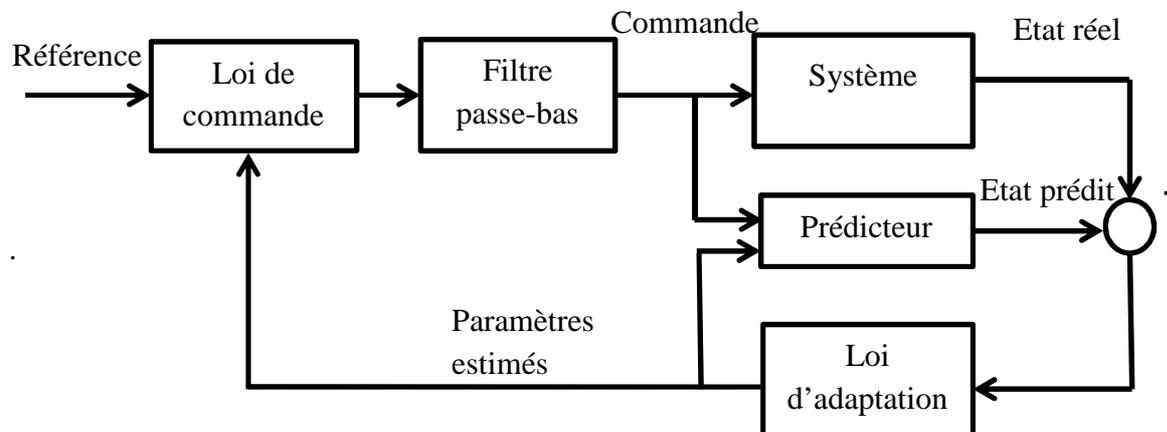


Figure 1. 3: Architecture générale de la commande adaptative  $\mathcal{L}_1$

La commande  $\mathcal{L}_1$  adaptative a été appliquée dans différents domaines et a démontré sa performance, on prend l'exemple des systèmes volant et plus particulièrement les quadri-rotors où bon nombre de travaux confirment l'efficacité de cette approche. Un état de l'art détaillé sur l'application de cette technique de commande au quadri-rotor est présenté en annexe B.

## 1.7 Contribution du mémoire

Ce mémoire porte sur l'utilisation d'outils modernes pour la modélisation et la commande des drones quadri-rotors, la contribution de ce travail est donnée ci-dessous :

- Modélisation dynamique du quadri-rotor : utiliser les outils mathématiques afin d'adapter un modèle du quadri-rotor à notre schéma de commande à l'aide des outils MATLAB / Simulink.
- Commande du quadri-rotor : comprendre, puis mettre en œuvre et maîtriser les techniques de commande d'un quadri-rotor , puis proposer une commande novatrice qui est la commande  $\mathcal{L}_1$  adaptative à base des modes glissants à laquelle est associée la commande tolérante aux défauts.
- Simulations virtuelles : en effectuant une connexion entre Matlab/Simulink et ROS, ce qui simplifie considérablement les manœuvres et simulations virtuelles en temps réel sur Gazebo.

## 1.8 Conclusion

Nous avons relevé les avantages de la configuration quadri-rotor mais surtout ses faiblesses qui ont motivé bon nombre de publications et de travaux, aussi ; nous avons pu constater la grande diversité des techniques de commande appliquées sur les quadri-rotors, et enfin ; nous avons établi les avantages de l'architecture de commande  $\mathcal{L}_1$  adaptative qui nous permettront de répondre à notre problématique.

# **Chapitre 2.**

## **Modélisation du quadri-rotor**

## 2.1 Introduction

La modélisation regroupe l'ensemble des techniques permettant de disposer d'une représentation mathématique du système à étudier. Elle requiert une connaissance précise des phénomènes intervenant dans le système et une aptitude à les représenter par des équations mathématiques.

La synthèse des lois de commande d'un système dynamique nécessite un modèle mathématique précis de ce dernier. Le modèle doit prévoir au mieux le comportement du système face aux différentes excitations (commandes, perturbations, etc). Ainsi, plus le modèle est précis, plus il est fidèle au système. Néanmoins, cela rend l'étude et la synthèse d'éventuelles lois de commande compliquées. Un compromis doit donc être fait en adoptant des hypothèses simplificatrices afin de pouvoir répondre aux contraintes pratiques.

Le quadri-rotor est classé dans la catégorie des systèmes volants les plus complexes, vu le nombre de phénomènes physiques qui affectent sa dynamique. Afin de concevoir un contrôleur de vol, on doit d'abord comprendre profondément les mouvements du système et sa dynamique. Cette compréhension est nécessaire pas simplement pour la conception du contrôleur, mais aussi pour s'assurer que les simulations de l'engin dépeindront un comportement aussi proche que possible de la réalité quand la commande est appliquée.

## 2.2 Repérage du quadri-rotor dans l'espace

Le quadri-rotor nécessite deux trièdres pour le repérer dans l'espace :

- **Le repère terrestre :**

Il est noté  $R_0(O_0, X_0, Y_0, Z_0)$ . C'est un repère lié au sol et supposé immobile.

- **Le repère lié au corps du quadri-rotor :**

Il est noté  $R_1(O_1, X_1, Y_1, Z_1)$ . C'est un repère dont l'origine  $O_1$  coïncide avec le centre de gravité  $G$  du quadri-rotor .

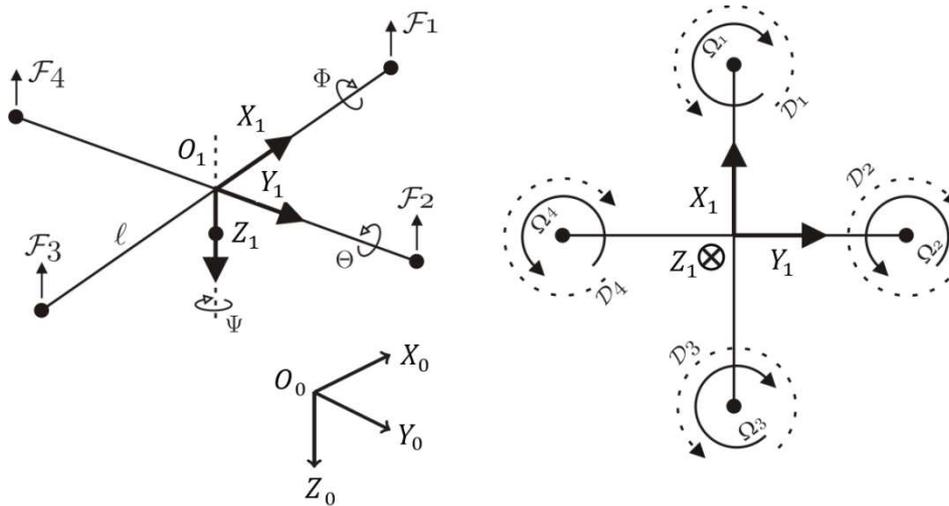


Figure 2. 1 : Les repères utilisés.

## 2.3 Modèle dynamique du quadri-rotor

Nous adaptons les hypothèses simplificatrices suivantes pour établir le modèle dynamique du quadri-rotor [16] :

- La structure du quadri-rotor est supposée rigide et symétrique.
- Les hélices sont supposées rigides pour pouvoir négliger l'effet de déformation lors de la rotation.
- La matrice d'inertie  $J$  est supposée constante (il n'y a pas de changement de masse).
- Les forces de portances et de trainée sont supposées proportionnelles au carré de la vitesse de rotation des rotors.

### 2.3.1 La dynamique de translation et de rotation

D'après la première loi de la dynamique de Newton, nous pouvons écrire :

$$\begin{cases} F = m \cdot \dot{V} + \omega \times m \cdot V \\ \mathcal{M} = J \cdot \dot{\omega} + \omega \times J \cdot \omega - \mathcal{M}_g \end{cases} \quad (2.5)$$

$F$  : Ensemble des forces extérieures appliquées sur le quadri-rotor [17].

$$F = F_g + F_p \quad (2.6)$$

$F_p$  : Force de portance, elle est perpendiculaire à l'écoulement de l'air, dirigée vers le haut, elle représente la force totale produite par les quatre hélices. Elle est donnée par :

$$F_p = -T \cdot z_1 \quad (2.7)$$

$$T = \left( \sum_{i=1}^4 f_i \right) z_i \quad (2.8)$$

Avec :  $f_i$  est la force de portance produite par la rotation de l'hélice  $i$ . Elle est donnée par :

$$f_i = b. \omega_i^2 \quad (2.9)$$

$F_g$  : Le poids dans l'axe du corps est donné par :

$$F_g = m. R^{-1}. \begin{bmatrix} 0 \\ 0 \\ g \end{bmatrix} = m. g. \begin{bmatrix} -S_\theta \\ C_\theta S_\phi \\ C_\theta C_\phi \end{bmatrix} \quad (2.10)$$

$V = [U \ V \ W]^T$  : Le vecteur des vitesses linéaires dans le repère lié au quadri-rotor.

$\omega = [p \ q \ r]^T$  : Le vecteur des vitesses angulaires dans le cadre du corps.

$J = \text{diag}(J_x, J_y, J_z)$  : La matrice d'inertie.

$\mathcal{M}$  : Le vecteur des moments agissant sur le centre de masse du quadri-rotor,  $\mathcal{M} = [L \ M \ N]^T$ , ils sont produits par les forces de trainée et de poussée créées par la rotation des quatre hélices :

$$\begin{aligned} L &= lb(\bar{\omega}_4^2 - \bar{\omega}_2^2) \\ M &= lb(\bar{\omega}_3^2 - \bar{\omega}_1^2) \\ N &= k(-\bar{\omega}_1^2 + \bar{\omega}_2^2 - \bar{\omega}_3^2 + \bar{\omega}_4^2) \end{aligned} \quad (2.11)$$

$l$  : La distance entre le centre de gravité du quadri-rotor et l'axe du rotor.

$k$  : Le coefficient de trainée.

$\bar{\omega}_i$  : Vitesses de rotation des moteurs.

$\mathcal{M}_g$  : Le moment gyroscopique, il se crée dans les systèmes physiques en mouvement avec des parties rotatoires. Lorsque la trajectoire du quadri-rotor change, le rotor subit des rotations du fuselage. Son expression générale est donnée par :

$$\mathcal{M}_g = \sum_{i=1}^4 \omega \times J_r (-1)^i. \bar{\omega}_i. z_1 \quad (2.12)$$

Où  $J_r$  est l'inertie des rotors [18],  $J_r$  est très petit, pour cette raison, les moments gyroscopiques sont supprimés dans la formulation du contrôleur. De plus, il existe de nombreux phénomènes aérodynamiques et aéroélastiques qui affectent le vol du quadri-rotor, tels que les effets du sol : en volant près du sol (ou pendant l'atterrissage), le flux d'air généré par les hélices perturbe la dynamique du quadri-rotor.

L'équation (2.5) donne :

$$\left\{ \begin{array}{l} \begin{bmatrix} \dot{U} \\ \dot{V} \\ \dot{W} \end{bmatrix} = \frac{1}{m} \begin{bmatrix} 0 \\ 0 \\ -T \end{bmatrix} + g \cdot \begin{bmatrix} -S_\theta \\ C_\theta S_\phi \\ C_\theta C_\phi \end{bmatrix} - \begin{bmatrix} qW - rV \\ rU - pW \\ pV - qU \end{bmatrix} \\ \begin{bmatrix} \dot{p} \\ \dot{q} \\ \dot{r} \end{bmatrix} = \begin{bmatrix} \frac{L}{J_x} \\ M \\ \frac{N}{J_y} \\ \frac{N}{J_z} \end{bmatrix} - \begin{bmatrix} \frac{J_z - J_y}{J_x} qr \\ \frac{J_x - J_z}{J_y} rp \\ \frac{J_y - J_x}{J_z} pq \end{bmatrix} \end{array} \right. \quad (2.13)$$

On utilise la matrice de rotation  $R$  pour exprimer les vitesses du véhicule dans l'axe de la terre, les vitesses centrées sur le corps sont connues :

$$\begin{bmatrix} \dot{X} \\ \dot{Y} \\ \dot{Z} \end{bmatrix} = R \cdot \begin{bmatrix} U \\ V \\ W \end{bmatrix} \quad (2.14)$$

$$\begin{bmatrix} \dot{X} \\ \dot{Y} \\ \dot{Z} \end{bmatrix} = \begin{bmatrix} C_\theta C_\psi & S_\phi S_\theta C_\psi - C_\phi S_\psi & C_\phi S_\theta C_\psi + S_\phi C_\psi \\ C_\theta S_\psi & C_\phi C_\psi + S_\phi S_\theta S_\psi & C_\phi S_\theta S_\psi - S_\phi C_\psi \\ -S_\theta & S_\phi C_\theta & C_\phi C_\theta \end{bmatrix} \cdot \begin{bmatrix} U \\ V \\ W \end{bmatrix} \quad (2.15)$$

### 2.3.2 La relation entre les angles d'Euler et les vitesses angulaires

Si un solide tourne à une vitesse constante, sa vitesse angulaire  $\omega$  est constante. Par contre, les variations des angles d'Euler seront variables car elles dépendent des angles instantanés entre les axes des deux repères. La séquence des angles d'Euler est obtenue à partir de trois rotations successives : roulis, tangage et lacet. La variation  $\dot{\phi}$  nécessite une rotation,  $\dot{\theta}$  nécessite deux rotations et  $\dot{\psi}$  nécessite trois rotations [19] :

$$\omega = R(X_0, \phi) \cdot \begin{bmatrix} \dot{\phi} \\ 0 \\ 0 \end{bmatrix} + R(X_0, \phi)R(Y_0, \theta) \cdot \begin{bmatrix} 0 \\ \dot{\theta} \\ 0 \end{bmatrix} + R(X_0, \phi)R(Y_0, \theta)R(Z_0, \psi) \cdot \begin{bmatrix} 0 \\ 0 \\ \dot{\psi} \end{bmatrix} \quad (2.16)$$

Ce qui donne :

$$\begin{bmatrix} p \\ q \\ r \end{bmatrix} = \begin{bmatrix} 1 & 0 & -S_\theta \\ 0 & C_\phi & S_\phi C_\theta \\ 0 & -S_\phi & C_\theta C_\phi \end{bmatrix} \cdot \begin{bmatrix} \dot{\phi} \\ \dot{\theta} \\ \dot{\psi} \end{bmatrix} \quad (2.17)$$

On trouve :

$$\begin{bmatrix} \dot{\phi} \\ \dot{\theta} \\ \dot{\psi} \end{bmatrix} = \begin{bmatrix} 1 & t_\theta S_\phi & t_\theta C_\phi \\ 0 & C_\phi & -S_\phi \\ 0 & \frac{S_\phi}{C_\theta} & \frac{C_\phi}{C_\theta} \end{bmatrix} \cdot \begin{bmatrix} p \\ q \\ r \end{bmatrix} \quad (2.18)$$

### 2.3.3 La dynamique des rotors

On peut exprimer la dynamique des rotors en utilisant la deuxième loi de Newton :

$$\frac{d(J_r \bar{\omega}_i)}{dt} = \sum \tau_{ext \rightarrow rotor_i} \quad (2.19)$$

La matrice  $J_r$  de chaque rotor est constante, et comme la vitesse angulaire  $\bar{\omega}_i$  est exprimée dans le repère lié aux rotors donc :

$$\frac{d(J_r \bar{\omega}_i)}{dt} = J_r \cdot \dot{\bar{\omega}}_i + \bar{\omega}_i \times J_r \cdot \bar{\omega}_i \quad (2.20)$$

Les couples (moments) extérieurs sont :

#### a. Couple réactif :

C'est un couple généré par l'hélice à une vitesse  $\bar{\omega}_i$  et opposant à sa rotation. Il est donné par :

$$\bar{Q}_i = -k |\bar{\omega}_i| \bar{\omega}_i \quad (2.21)$$

$k$  : Coefficient de traînée. Il dépend de plusieurs facteurs : la densité de l'air, le rayon de l'hélice, nombre de pales, ...

#### b. Couple moteur :

Il est noté  $\tau_i$ .

#### c. Couple de poussée :

Il est produit par la force de poussée générée par l'hélice. Comme cette force est perpendiculaire au plan de rotation de l'hélice, elle ne produit aucun couple.

#### d. Moment visqueux :

Résultat du frottement visqueux, qui s'oppose à la rotation de l'hélice. Il est donné par :

$$\tau_{visq} = -C_v \bar{\omega}_i \quad (2.22)$$

$C_v$  : Coefficient du frottement visqueux.

Comme les hélices ne tournent pas dans la même direction, le signe du couple moteur est toujours celui de la vitesse de rotation  $\bar{\omega}_i$ , tandis que le couple réactif et le moment visqueux sont toujours opposés à lui, donc on peut écrire :

$$\sum \tau_{ext \rightarrow rotor_i} = (-1)^i [\tau_i - k \cdot \bar{\omega}_i^2 - C_v \bar{\omega}_i] z_1 \quad (2.23)$$

L'équation (2.17) devient :

$$\frac{d(J_r (-1)^i \bar{\omega}_i)}{dt} z_1 = J_{rz} [(-1)^i \dot{\bar{\omega}}_i z_1 + (-1)^i \bar{\omega}_i \cdot (\omega \times z_1)] \quad (2.24)$$

Le terme  $(-1)^i \bar{\omega}_i \cdot (\omega \times z_1)$  est perpendiculaire à l'axe  $z_1$ , l'équation finale de la dynamique du rotor  $i$  sera donc :

$$J_{rz} \cdot \dot{\bar{\omega}}_i = \tau_i - \bar{\omega}_i^2 - C_v \bar{\omega}_i \quad (2.25)$$

On obtiendra le système suivant :

$$\begin{cases} J_{rz} \cdot \dot{\bar{\omega}}_1 = \tau_1 - \bar{\omega}_1^2 - C_v \bar{\omega}_1 \\ J_{rz} \cdot \dot{\bar{\omega}}_2 = \tau_2 - \bar{\omega}_2^2 - C_v \bar{\omega}_2 \\ J_{rz} \cdot \dot{\bar{\omega}}_3 = \tau_3 - \bar{\omega}_3^2 - C_v \bar{\omega}_3 \\ J_{rz} \cdot \dot{\bar{\omega}}_4 = \tau_4 - \bar{\omega}_4^2 - C_v \bar{\omega}_4 \end{cases} \quad (2.26)$$

### 2.3.4 Modèle d'état de quadri-rotor

Le système dynamique final du quadri-rotor donné par (2.13, 2.15, 2.18, 2.26) est subdivisé en quatre sous-systèmes (position/vitesse linéaire, angles d'Euler, vitesse angulaire et équations des rotors). Une architecture de ce type, où l'état du premier système est contrôlé par une commande indirecte, qui est en réalité l'état prochain du système, est appelée : structure en cascade.

$$\begin{aligned} \dot{X} &= (C_\theta C_\psi)U + (S_\phi S_\theta C_\psi - C_\phi S_\psi)V + (C_\phi S_\theta C_\psi + S_\phi C_\psi)W \\ \dot{Y} &= (C_\theta S_\psi)U + (S_\phi S_\theta S_\psi + C_\phi C_\psi)V + (C_\phi S_\theta S_\psi - S_\phi C_\psi)W \\ \dot{Z} &= -S_\theta U + (S_\phi C_\theta)V + (C_\phi C_\theta)W \\ \dot{U} &= -gS_\theta - (qW - rV) \\ \dot{V} &= gS_\phi C_\theta - (rU - pW) \\ \dot{W} &= -\frac{T}{m} + gC_\phi C_\theta - (pV - qU) \\ \dot{\phi} &= p + (S_\phi t_\theta)q + (S_\phi t_\theta)r \\ \dot{\theta} &= C_\phi q - S_\phi r \\ \dot{\psi} &= \frac{S_\phi}{C_\psi} q + \frac{C_\phi}{C_\theta} r \end{aligned} \quad (2.27)$$

$$\begin{aligned}\dot{p} &= \frac{L}{J_x} + \left(\frac{J_y - J_z}{J_x}\right)qr \\ \dot{q} &= \frac{M}{J_y} + \left(\frac{J_z - J_x}{J_y}\right)rp \\ \dot{r} &= \frac{N}{J_z} + \left(\frac{J_x - J_y}{J_z}\right)pq\end{aligned}$$

L'écriture des entrées de commande  $T, L, M,$  et  $N$  en fonction des vitesses de rotation des rotors est la suivante :

$$\begin{bmatrix} T \\ L \\ M \\ N \end{bmatrix} = \begin{bmatrix} b & b & b & b \\ 0 & -lb & 0 & lb \\ -b & 0 & b & 0 \\ -k & k & -k & k \end{bmatrix} \begin{bmatrix} \omega_1^2 \\ \omega_2^2 \\ \omega_3^2 \\ \omega_4^2 \end{bmatrix} \quad (2.28)$$

On choisit comme vecteur d'état :

$$X = [X Y Z U V W \phi \theta \psi p q r]^T$$

On aura le système suivant :

$$\begin{aligned}\dot{x}_1 &= (C_{x_8}C_{x_9})x_4 + (S_{x_7}S_{x_8}C_{x_9} - C_{x_7}S_{x_9})x_5 + (C_{x_7}S_{x_8}C_{x_9} + S_{x_7}C_{x_9})x_6 \\ \dot{x}_2 &= (C_{x_8}S_{x_9})x_4 + (S_{x_7}S_{x_8}S_{x_9} + C_{x_7}C_{x_9})x_5 + (C_{x_7}S_{x_8}S_{x_9} - S_{x_7}C_{x_9})x_6 \\ \dot{x}_3 &= -S_{x_8}x_4 + (S_{x_7}C_{x_8})x_5 + (C_{x_7}C_{x_8})x_6 \\ \dot{x}_4 &= -gS_{x_8} - (x_{11}x_6 - x_{12}x_5) \\ \dot{x}_5 &= gS_{x_7}C_{x_8} - (x_{12}x_4 - x_{10}x_6) \\ \dot{x}_6 &= -\frac{T}{m} + gC_{x_7}C_{x_8} - (x_{10}x_5 - x_{11}x_4) \\ \dot{x}_7 &= x_{10} + (S_{x_7}t_{x_8})x_{11} + (S_{x_7}t_{x_8})x_{12} \\ \dot{x}_8 &= C_{x_7}x_{11} - S_{x_7}x_{12} \\ \dot{x}_9 &= \frac{S_{x_7}}{C_{x_9}}x_{11} + \frac{C_{x_7}}{C_{x_8}}r \\ \dot{x}_{10} &= \frac{L}{J_x} + \left(\frac{J_y - J_z}{J_x}\right)x_{11}x_{12} \\ \dot{x}_{11} &= \frac{M}{J_y} + \left(\frac{J_z - J_x}{J_y}\right)x_{10}x_{12} \\ \dot{x}_{12} &= \frac{N}{J_z} + \left(\frac{J_x - J_y}{J_z}\right)x_{10}x_{11}\end{aligned} \quad (2.29)$$

## 2.4 Conclusion

Dans ce chapitre, nous avons décrit les mouvements de base du quadri-rotor . Puis, nous avons donné son modèle dynamique par la loi fondamentale de la dynamique (Newton). Ce modèle montre la nature couplée, complexe, non linéaire, multi-variable et sous-actionnée du système. La dynamique est exprimée en deux parties : dans le repère du corps et dans le repère fixe qui nous intéresse le plus. Cette modélisation est très proche du comportement réel du quadri-rotor car nous n'avons pas considéré les premières dérivées des angles d'Euler

comme vitesses angulaires du quadri-rotor , de plus, les déplacements  $X, Y$  sont commandés respectivement par  $\theta$  et  $\phi$ .

# **Chapitre 3. La commande adaptative et la commande tolérante aux défauts (FTC)**

### 3.1 Introduction

Le quadri-rotor a fait l'objet de nombreux travaux de recherche ces dernières années, et qui portent sur la conception d'un régulateur assurant la stabilité et le suivi de trajectoire désirée en présence de perturbations (le vent en particulier) qui affecte la stabilité et les performances du système. La nature et le comportement non-linéaire, la complexité et l'incertitude du modèle du quadri-rotor s'appuyant sur des hypothèses qui peuvent changer en pratique et les perturbations stochastiques du vent nous poussent à utiliser la commande adaptative qui répond aux exigences de commande dans des situations pareilles.

Dans ce chapitre, nous allons faire la synthèse d'une commande adaptative qui assure la stabilité et le suivi de trajectoire en présence de rafales du vent, deux approches seront présentées : la commande  $\mathcal{L}_1$  adaptative avec descente du gradient en premier lieu, et puis la commande  $\mathcal{L}_1$  adaptative avec modes glissants en deuxième lieu. Ensuite, nous allons présenter les concepts fondamentaux de la commande tolérante aux défauts FTC afin de l'appliquer au quadri-rotor. Dans cette partie, nous nous intéresserons à la commande FTC appliquée avec la commande  $\mathcal{L}_1$  adaptative où nous allons évaluer son aptitude à maintenir la robustesse et les performances en présence de différents défauts.

### 3.2 La commande adaptative

La commande adaptative est apparue dans les années cinquante à cause de l'insuffisance des commandes classiques, en effet ; pour certains systèmes ; aucune combinaison des paramètres du régulateur n'était capable de les stabiliser globalement, la solution était donc l'adaptation de ces paramètres en les faisant varier par rapport au temps de manière à optimiser certains critères.

La commande adaptative a permis d'obtenir un régulateur dynamique et non-linéaire qui n'est pas aussi universel mais qui convient à une large classe de systèmes [31]. La commande adaptative a une littérature très riche pour la conception, l'analyse, l'exécution et l'application. Plusieurs approches de commandes adaptatives ont été créées comme MRAC directe et indirecte, mais aussi, la commande adaptative a été combinée avec des commandes classiques et modernes améliorant ainsi les performances recherchées, nous citons par exemple la commande avec PID adaptative, mode glissant adaptatifs,  $H_\infty$  adaptative etc.

La commande adaptative apporte différents avantages au système de commande, nous pouvons citer les principaux points suivants [32] :

- Le régulateur peut s'adapter en temps réel au comportement du processus.
- La commande adaptative est une solution adéquate pour la commande des processus non-linéaires dépendant d'un point de consigne, car la commande de ce type de systèmes avec un régulateur à paramètres fixes est souvent difficile vu leur comportement variable.

- Les régulateurs adaptatifs peuvent souvent éliminer les erreurs plus rapidement et avec moins de fluctuations, permettant au processus d'être exploité plus près de ses contraintes où la rentabilité est la plus élevée. Ceci est particulièrement avantageux dans les industries telles que la pétrochimie et l'aérospatiale où la performance compte pour des raisons de rentabilité ou de sécurité.

### 3.3 Commande $\mathcal{L}_1$ adaptative d'un système MIMO

Nous avons déjà présenté cette commande dans le premier chapitre. Dans cette partie, nous allons faire le développement théorique et une application sur le quadri-rotor et puis nous allons analyser ses avantages et inconvénients en faisant des tests de performances et de robustesse.

#### 3.3.1 Formulation du problème

Soit la classe de systèmes non-linéaires suivante :

$$\begin{aligned} \dot{x}(t) &= A_m x(t) + B(\omega u(t) + \theta^T x(t) + \eta_m(t, x)) + \eta_u(t, x) \\ y(t) &= Cx(t) \end{aligned} \quad (3.1)$$

$A_m \in R^{n \times n}$  : la matrice de Hurwitz qui est la matrice de dynamique désirée en boucle fermée.

$B \in R^{n \times m}, C \in R^{m \times n}$  : matrices constantes connues.

$\omega$ : matrice constante inconnue (paramètres du système).

$\theta^T \in R^{m \times n}$  : matrice des paramètres inconnus, elle représente les incertitudes de modélisation.

$\eta_m(t, x) \in R^m$  : perturbation en entrée.

$\eta_u(t, x) \in R \times R^n \rightarrow R^n$  : Erreur de modélisation et perturbation.

#### 3.3.2 Les hypothèses

- Les fonctions non-linéaires  $\eta_m(t, x)$  et  $\eta_u(t, x)$  sont uniformément bornées :  
 $\exists L_m, L_u > 0$  tel que pour  $\forall t \geq 0$  :  $\|\eta_m(t, x)\| \leq L_m$  et  $\|\eta_u(t, x)\| \leq L_u$ .  
 $\|\cdot\|$  : la norme 2 d'une matrice.  
 $L_m, L_u$  : constantes réelles.
- Les incertitudes de modélisation sont bornées :  
 $\theta \in \Theta$ , où  $\Theta$  est un ensemble compact convexe connu.
- $\omega$  est une matrice diagonale strictement dominante ( $\forall i \in [1, n], |\omega_{i,i}| > \sum_{j=1, j \neq i}^n |\omega_{i,j}|$  ce qui signifie que les couplages sont faibles), non singulière avec  $\text{sgn}(\omega_{ii})$  connu. Et nous supposons qu'il existe un ensemble compact convexe connu  $\Omega$  tel que  
 $\omega \in \Omega \subset R^{m \times n}$ .

### 3.3.3 Architecture de la commande

La commande  $\mathcal{L}_1$  adaptative se compose d'un prédicteur d'état, algorithme d'adaptation paramétrique AAP, filtre passe bas et calculateur de la commande.

Pour estimer les paramètres, nous allons présenter deux approches différentes : la descente du gradient et les modes glissants, et nous allons comparer ces deux méthodes à travers les résultats de simulations obtenus.

## 3.4 La commande $\mathcal{L}_1$ avec la descente du gradient

D'après [34] et [35] pour la conception de la commande  $\mathcal{L}_1$ , les bornes des perturbations  $L_m, L_u$  sont supposés connues.

Aussi, le prédicteur d'état est défini par :

$$\begin{aligned} \dot{\hat{x}}(t) = & A_m \hat{x}(t) + B_m \left( \hat{\omega}(t)u(t) + \hat{\theta}_1(t)\|x_t\|_{\mathcal{L}_\infty} + \hat{\eta}_1(t) \right) \\ & + B_{um} \left( \hat{\theta}_2(t)\|x_t\|_{\mathcal{L}_\infty} + \hat{\eta}_2(t) \right), \hat{x}(0) = x_0 \end{aligned} \quad (2.2)$$

$\hat{x}(t)$  est l'état prédit et  $\hat{\theta}_i(t), \hat{\omega}(t), \hat{\eta}_i(t)$  sont les estimations des paramètres inconnus du système et des perturbations.

$B_m \in R^{n \times m}; B_{um} \in R^{n \times (n-m)}$  : deux matrices constantes connues tel que :

$$\begin{aligned} B_m^T B_{um} &= 0 \text{ et } \text{rang}([B_m \ B_{um}]) = n \\ \|x_t\|_{\mathcal{L}_\infty} &= \max_{0 \leq \tau \leq t} (\max_{i=1, \dots, n} |x_i(\tau)|) \end{aligned} \quad (3.3)$$

### 3.4.1 Les lois d'adaptation

$$\begin{aligned} \dot{\hat{\omega}}(t) &= \Gamma \text{Proj}(\hat{\omega}(t), -(\tilde{x}^T(t) P B_m)^T u^T(t)), \quad \hat{\omega}(0) = \hat{\omega}_0 \\ \dot{\hat{\theta}}_1(t) &= \Gamma \text{Proj}(\hat{\theta}_1(t), -(\tilde{x}^T(t) P B_m)^T \|x_t\|_{\mathcal{L}_\infty}), \quad \hat{\theta}_1(0) = \hat{\theta}_{10} \\ \dot{\hat{\eta}}_1(t) &= \Gamma \text{Proj}(\hat{\eta}_1(t), -(\tilde{x}^T(t) P B_m)^T), \quad \hat{\eta}_1(0) = \hat{\eta}_{10} \\ \dot{\hat{\theta}}_2(t) &= \Gamma \text{Proj}(\hat{\theta}_2(t), -(\tilde{x}^T(t) P B_{um})^T \|x_t\|_{\mathcal{L}_\infty}), \quad \hat{\theta}_2(0) = \hat{\theta}_{20} \\ \dot{\hat{\eta}}_2(t) &= \Gamma \text{Proj}(\hat{\eta}_2(t), -(\tilde{x}^T(t) P B_{um})^T), \quad \hat{\eta}_2(0) = \hat{\eta}_{20} \end{aligned} \quad (3.4)$$

$\tilde{x}(t) = \hat{x}(t) - x(t)$  est l'erreur d'estimation,  $\Gamma > 0$  est le gain d'adaptation.

$P = P^T > 0$  est la solution de l'équation algébrique de Lyapunov :

$$A_m^T P + P A_m = -Q, \quad Q = Q^T > 0 \quad (3.5)$$

L'opérateur  $\text{Proj}(\dots)$  (défini en annexe) assure le maintien des paramètres dans leurs domaines définis.

### 3.4.2 La loi de commande

La loi de commande est donnée par :

$$u(s) = -K_d D(s) \hat{v}(s) \quad (3.6)$$

$D(s)$  : matrice de transfert strictement propre d'ordre  $m$ .

$K_d \in R^{m \times m}$  : matrice gain.

$\hat{v}(s)$  est la transformée de Laplace du signal :

$$\hat{v}(t) \triangleq \hat{\omega}(t)u(t) + \hat{v}_1(t) + \hat{v}_{2m}(t) - r_g \quad (3.7)$$

Avec  $r_g \triangleq K_g r(s)$ ,  $K_g = -(CA_m^{-1}B_m)^{-1}$  : le pré-compensateur qui assure la poursuite en régime statique

$$\begin{aligned} \hat{v}_{2m}(s) &\triangleq H_m^{-1}(s)H_{um}(s)\hat{v}_2(t) \\ \hat{v}_1(t) &\triangleq \hat{\theta}_1(t)\|x_t\|_{\mathcal{L}_\infty} + \hat{\eta}_1(t) \\ \hat{v}_2(t) &\triangleq \hat{\theta}_2(t)\|x_t\|_{\mathcal{L}_\infty} + \hat{\eta}_2(t) \end{aligned} \quad (3.8)$$

Avec  $H_{um}(s) = C(sI - A_m)^{-1}B_{um}$  et  $H_m(s) = C(sI - A_m)^{-1}B_m$ .

La conception de  $D(s)$  et  $K_d$  est faite pour avoir un filtre stable strictement propre  $C(s)$  :

$$C(s) = \omega K_d D(s) (I + \omega K D(s))^{-1}, C(0) = I_m \quad (3.9)$$

L'analyse de robustesse de la commande est présentée dans [36].

### 3.4.3 Application au quadri-rotor

Notre modèle d'état (2.26) est de la forme :

$$\dot{x}(t) = F(x(t)) + B\omega u(t) \quad (3.10)$$

Avec :

$$F(x) = \begin{cases} (C_{x_8} C_{x_9})x_4 + (S_{x_7} S_{x_8} C_{x_9} - C_{x_7} S_{x_9})x_5 + (C_{x_7} S_{x_8} C_{x_9} + S_{x_7} C_{x_9})x_6 \\ (C_{x_8} S_{x_9})x_4 + (S_{x_7} S_{x_8} S_{x_9} + C_{x_7} C_{x_9})x_5 + (C_{x_7} S_{x_8} S_{x_9} - S_{x_7} C_{x_9})x_6 \\ -S_{x_8} x_4 + (S_{x_7} C_{x_8})x_5 + (C_{x_7} C_{x_8})x_6 \\ -g S_{x_8} - (x_{11} x_6 - x_{12} x_5) \\ g S_{x_7} C_{x_8} - (x_{12} x_4 - x_{10} x_6) \\ g C_{x_7} C_{x_8} - (x_{10} x_5 - x_{11} x_4) \\ x_{10} + (S_{x_7} t_{x_8})x_{11} + (S_{x_7} t_{x_8})x_{12} \\ C_{x_7} x_{11} - S_{x_7} x_{12} \\ \frac{S_{x_7}}{C_{x_9}} x_{11} + \frac{C_{x_7}}{C_{x_8}} x_{12} \\ \left(\frac{J_y - J_z}{J_x}\right) x_{11} x_{12} \\ \left(\frac{J_z - J_x}{J_y}\right) x_{10} x_{12} \\ \left(\frac{J_x - J_y}{J_z}\right) x_{10} x_{11} \end{cases} \quad (3.11)$$

$$B = \begin{bmatrix} 0 & 0 & 0 & 0 \\ -1 & 0 & 0 & 0 \\ 0_{3 \times 1} & 0_{3 \times 1} & 0_{3 \times 1} & 0_{3 \times 1} \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \quad (3.12)$$

$$\omega = \begin{bmatrix} 1/m & 0 & 0 & 0 \\ 0 & 1/J_x & 0 & 0 \\ 0 & 0 & 1/J_y & 0 \\ 0 & 0 & 0 & 1/J_z \end{bmatrix} \quad (3.13)$$

Linéarisation du système autour du point de d'équilibre  $P_0 = \begin{Bmatrix} x_0 \\ u_0 \end{Bmatrix}$ :

$$\sum_L x_0 = \begin{cases} \dot{x}(t) = Ax(t) + B_l u(t) \\ y(t) = Cx(t) \end{cases} \quad (3.14)$$

Le point d'équilibre est le point correspond à  $\dot{x}(t) = 0$

On trouve  $P_0 = \begin{cases} x_0 = 0_{12 \times 1} \\ u_0 = [mg \ 0 \ 0 \ 0]^T \end{cases}$

A : la matrice dynamique de dimension  $n \times n$ , elle est donnée par :

$$A = \frac{dF(x(t))}{dx(t)} \quad \text{La matrice Jacobienne de F en } x = x_0$$

$$A = \begin{bmatrix} 0_{3 \times 3} & I_3 & 0 & 0 & 0 & 0_{3 \times 3} \\ 0_{1 \times 3} & 0_{1 \times 3} & 0 & -g & 0 & 0_{1 \times 3} \\ 0_{1 \times 3} & 0_{1 \times 3} & g & 0 & 0 & 0_{1 \times 3} \\ 0 & 0 & 0 & 0 & 0 & 0 \\ 0_{3 \times 3} & 0_{3 \times 3} & 0_{3 \times 1} & 0_{3 \times 1} & 0_{3 \times 1} & I_3 \\ 0_{3 \times 3} & 0_{3 \times 3} & 0_{3 \times 1} & 0_{3 \times 1} & 0_{3 \times 1} & 0_{3 \times 3} \end{bmatrix} \quad (3.15)$$

$$B_l = B\omega = \begin{bmatrix} 0 & 0 & 0 & 0 \\ -1/m & 0 & 0 & 0 \\ 0_{3 \times 1} & 0_{3 \times 1} & 0_{3 \times 1} & 0_{3 \times 1} \\ 0 & 1/J_x & 0 & 0 \\ 0 & 0 & 1/J_y & 0 \\ 0 & 0 & 0 & 1/J_z \end{bmatrix} \quad (3.16)$$

La matrice dynamique en boucle fermée  $A_m$  :

$$A_m = A - B_l K \quad (3.17)$$

La matrice de gain  $K$  est calculée par la méthode  $LQR$  qui se base sur l'optimisation du critère quadratique suivant :

$$J = \int_0^{\infty} (x^T(t)Qx(t) + u^T(t)Ru(t)) \quad (3.18)$$

$Q, R$  : matrices de pondération positives et  $Q^T = Q$

$$K = R^{-1}B^T P \quad (3.19)$$

$P = P^T \geq 0$  est la solution de l'équation de Lyapunov :

$$A^T P + PA + Q - PBR^{-1} = 0 \quad (3.20)$$

La commande appliquée sur le système non-linéaire :

$$u(t) = u_m(t) + u_{ad}(t) \quad (3.21)$$

$u_m(t) = -Kx(t) + u_0$  : la commande en retour d'état.

$u_{ad}(t)$  : la commande adaptative.

On ajoute et soustrait  $Ax(t)$  au système non-linéaire (3.14) :

$$\dot{x}(t) = Ax(t) + F(x(t)) + B\omega u(t) - Ax(t)$$

On remplace la commande  $u(t)$  par  $u_m(t) + u_{ad}(t)$  on trouve :

$$\dot{x}(t) = A_m x(t) + B\omega u_{ad}(t) + \eta_u(t, x) \quad (3.3)$$

$$A_m = A - B\omega K$$

$$\eta_u(t, x) = F(x(t)) - Ax(t) + B\omega u_0$$

$$\eta_u(t, x) = \left\{ \begin{array}{l} (C_{x_8} C_{x_9} - 1)x_4 + (S_{x_7} S_{x_8} C_{x_9} - C_{x_7} S_{x_9})x_5 + (C_{x_7} S_{x_8} C_{x_9} + S_{x_7} C_{x_9})x_6 \\ (C_{x_8} S_{x_9})x_4 + (S_{x_7} S_{x_8} S_{x_9} + C_{x_7} C_{x_9} - 1)x_5 + (C_{x_7} S_{x_8} S_{x_9} - S_{x_7} C_{x_9})x_6 \\ -S_{x_8} x_4 + (S_{x_7} C_{x_8})x_5 + (C_{x_7} C_{x_8} - 1)x_6 \\ (x_8 - S_{x_8})g - (x_{11}x_6 - x_{12}x_5) \\ (S_{x_7} C_{x_8} - x_7)g - (x_{12}x_4 - x_{10}x_6) \\ (C_{x_7} C_{x_8} - 1)g - (x_{10}x_5 - x_{11}x_4) \\ (S_{x_7} t_{x_8})x_{11} + (S_{x_7} t_{x_8})x_{12} \\ (C_{x_7} - 1)x_{11} - S_{x_7} x_{12} \\ \frac{S_{x_7}}{C_{x_9}} x_{11} + \left( \frac{C_{x_7}}{C_{x_8}} - 1 \right) x_{12} \\ \left( \frac{J_y - J_z}{J_x} \right) x_{11} x_{12} \\ \left( \frac{J_z - J_x}{J_y} \right) x_{10} x_{12} \\ \left( \frac{J_x - J_y}{J_z} \right) x_{10} x_{11} \end{array} \right. \quad (3.4)$$

Le prédicteur d'état est défini par :

$$\hat{x}(t) = A_m \hat{x}(t) + B_m (\hat{\omega}(t)u(t) + \hat{\eta}_1(t)) + B_{um} \hat{\eta}_2(t), \quad \hat{x}(0) = x_0 \quad (5.24)$$

$$B_m = \begin{bmatrix} 0_{5 \times 1} & 0_{5 \times 3} \\ -1 & 0_{1 \times 3} \\ 0_{3 \times 3} & 0_{3 \times 3} \\ 0_{3 \times 3} & I_3 \end{bmatrix}$$

$$\eta_u(t, x) = B_m \eta_1(t, x) + B_{um} \eta_2(t, x) \quad (3.25)$$

$$B_{um} \in R^{12 \times 8}, B_m^T B_{um} = 0, \text{rang}([B_m \ B_{um}]) = 12$$

$$B_{um} = \begin{bmatrix} I_5 & 0_{5 \times 3} \\ 0_{1 \times 5} & 0_{1 \times 3} \\ 0_{3 \times 5} & I_3 \\ 0_{3 \times 5} & 0_{3 \times 3} \end{bmatrix}$$

### 3.5 La commande $\mathcal{L}_1$ adaptative avec modes glissants

En raison des limitations présentées par l'approche  $\mathcal{L}_1$  adaptative avec descente du gradient, et plus particulièrement, la sensibilité aux variations des bornes de perturbations (Voir chapitre 5), une approche basée sur les modes glissants permettant de dépasser ces limitations en estimant les bornes des perturbations est proposée dans cette partie. Cette approche a la même structure que celle avec la descente du gradient mais elles diffèrent dans les lois d'adaptation. Les éléments communs entre ces deux approches déjà définis précédemment ne seront pas redéfinis dans cette partie.

Le prédicteur d'état est défini par [20] :

$$\begin{aligned}\hat{\dot{x}}(t) &= A_m \hat{x}(t) + B \left( \hat{\omega}(t)u(t) + \hat{\theta}^T(t)x(t) + \hat{\eta}_m(t) \right) + \hat{\eta}_u(t) \\ \hat{x}(0) &= x_0\end{aligned}\quad (3.66)$$

$\hat{x}(t)$  est l'état prédite et  $\hat{\theta}(t)$ ,  $\hat{\omega}(t)$ ,  $\hat{\eta}_m(t)$  et  $\hat{\eta}_u(t)$  sont les estimations des paramètres inconnus du système et les perturbations.

Soit la surface de glissement :

$$\sigma(t) = \lambda \tilde{x}(t) \quad (3.27)$$

$\lambda \in R^{m \times n}$  est une matrice constante arbitraire, elle est choisie de telle façon que la matrice  $\lambda B$  soit non-singulière (inversible) et les coefficients  $\lambda(i, j)$  forment un hyperplan stable.

L'estimation de  $\hat{\eta}_m(t)$  est donnée par :

$$\hat{\eta}_m(t) = -(\lambda B)^{-1}(\lambda A_m \tilde{x}(t) + \alpha \sigma(t)) - \hat{L}_m(t) \frac{B^T \lambda^T \sigma(t)}{\|B^T \lambda^T \sigma(t)\|} \quad (3.28)$$

$\alpha$ : un nombre positif arbitraire.

L'estimation de la borne  $\hat{L}_m(t)$  de  $\eta_m(t, x)$  est donnée par :

$$\hat{L}_m(t) = \Gamma \|\sigma^T(t) \lambda B\| - \varepsilon_m \hat{L}_m(t), \hat{L}_m(0) = L_{m0}, 0 < \varepsilon_m < 1 \quad (3.29)$$

L'estimation de  $\hat{\eta}_u(t)$  est donnée par :

$$\hat{\eta}_u = -\hat{L}_u(t) \frac{\lambda^T \sigma(t)}{\|\lambda^T \sigma(t)\|} \quad (3.30)$$

L'estimation de la borne  $\hat{L}_u(t)$  de  $\eta_u(t, x)$  est donnée par :

$$\hat{L}_u(t) = \Gamma \|\lambda^T \sigma(t)\| - \varepsilon_u \hat{L}_u(t), \hat{L}_u(0) = L_{u0}, 0 < \varepsilon_u < 1 \quad (3.31)$$

Les paramètres inconnus  $\theta$  et  $\omega$  sont estimés par :

$$\begin{aligned}\hat{\omega}(t) &= -\Gamma(u(t)\sigma^T(t)\lambda B)^T \\ \hat{\theta}(t) &= -\Gamma(x(t)\sigma(t)\lambda B)\end{aligned}\quad (3.32)$$

Soient  $H_m(s) = C(sI_n - A_m)^{-1}B$ ,  $H_0(s) = C(sI_n - A_m)^{-1}$  et  $H_u(s) = (sI_n - A_m)^{-1}B$

### 3.5.1 La loi de commande

$$u(s) = K_d D(s)(K_g r(s) - \hat{v}_1(s) - \hat{v}_2(s)) \quad (3.33)$$

$\hat{v}_1(s)$  : la transformée de Laplace du terme :  $\hat{\omega}(t)u(t) + \hat{\theta}^T(t)x(t) + \hat{\eta}_m(t)$ .

$$\hat{v}_2(s) = H_m^{-1}(s)H_0(s)\hat{\eta}_u(s)$$

L'architecture de la commande :

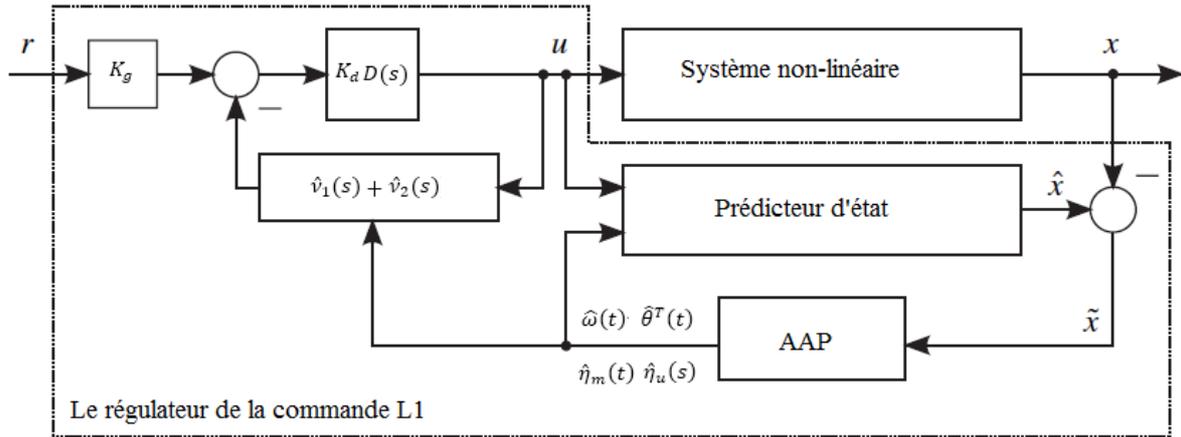


Figure 3. 1 : L'architecture de la commande L1.

### 3.5.2 Application au quadri-rotor

Le prédicteur d'état dans ce cas s'écrit comme suit :

$$\dot{\hat{x}}(t) = A_m \hat{x}(t) + B \omega u_{ad}(t) + \hat{\eta}_u(t) \quad (3.34)$$

La surface de glissement :

$$\sigma(t) = \lambda \tilde{x}(t); \lambda \in R^{4 \times 12} \quad (3.35)$$

Ici, s'intéresse seulement à l'estimation de  $\hat{\eta}_u(t)$  pour compenser les perturbations du vent et maintenir la poursuite.

$$\hat{\eta}_u = -\hat{L}_u(t) \frac{\lambda^T \sigma(t)}{\|\lambda^T \sigma(t)\|}$$

$$\hat{L}_u(t) = \Gamma \|\lambda^T \sigma(t)\| - \varepsilon_u \hat{L}_u(t), \hat{L}_u(0) = L_{u0}, 0 < \varepsilon_u < 1$$

## 3.6 Commande tolérante aux défauts

Plusieurs travaux sur la commande tolérante aux défauts (FTC) appliqués sur les multi-rotors ont été déjà réalisés, nous citons par exemple Mueller et D'Andrea ainsi que Lanzon et al qui ont résolu le problème de défaillance en perdant la commande d'un degré de liberté (l'angle de lacet) pour maintenir le vol stationnaire [47] [48]. Adîr et al., Marks et al., Hamayun et al. Ainsi qu'Alwi et Edwards qui ont utilisé les méthodes d'attribution de commande qui reposent sur la redondance des actionneurs dans les arrangements hexrotor ou octorotor [49] [50] [51] [52].

### 3.6.1 Défauts et défaillances

Un défaut est un dysfonctionnement d'un ou plusieurs éléments du système qui peut dégrader ses performances, mais il peut être tolérable et n'affectant pas le bon fonctionnement du système. Le cas extrême d'un défaut est la défaillance totale, c'est une panne catastrophique d'un composant ou d'un système, pour faire face à une telle situation, une commande reconfigurable est nécessaire et une redondance matérielle doit être disponible.

### 3.6.2 Classification des défauts

#### 3.6.2.1 Classification selon la localisation

Un défaut peut se produire dans les trois parties suivantes :

- **Défauts d'actionneurs :**

Les défauts actionneurs représentent une perte totale ou partielle de l'action de commande. Un actionneur complètement perdu est un actionneur bloqué qui ne fonctionne pas. L'actionneur partiellement défaillant ne produit qu'une partie de l'action nominale. Dans le cas d'un quadri-rotor , les actionneurs sont les moteurs équipés avec des hélices, une perte d'efficacité d'un moteur ou dégradation d'une hélice (impact, changement de forme) affecte directement le comportement du système.

- **Défauts de capteurs :**

Les défauts capteurs représentent des mesures incorrectes d'un signal. Ces défauts peuvent dégrader la robustesse et les performances du système en boucle fermée. Les défauts capteurs les plus connus sont : le biais, la dérive, la perte d'efficacité, le blocage et l'erreur de calibrage.

- **Défauts de composants :**

Les défauts composants sont les défauts qui proviennent du système lui-même, souvent ces défauts n'appartiennent pas à la catégorie des défauts capteurs et actionneurs. Ils représentent des changements dans les paramètres du système. Les défauts composants se traduisent par un changement dans l'équation d'état.

### 3.6.2.2 Classification selon la modélisation

Les défauts peuvent être modélisés comme défauts additifs ou multiplicatifs.

- **Défauts additifs :**

Les défauts additifs sont considérés comme des signaux externes supplémentaires, qui correspondent à des changements constatés indépendamment des entrées connues. Ces défauts sont appropriés pour représenter des défauts de composants dans le système à commander.

- **Défauts multiplicatifs :**

Les défauts multiplicatifs correspondent à des changements de paramètres qui affectent l'évolution des entrées et/ou des sorties, où l'amplitude de ces défauts dépend des entrées connues. Notons que les défauts d'actionneurs et les défauts capteurs sont le plus souvent multiplicatifs par nature.

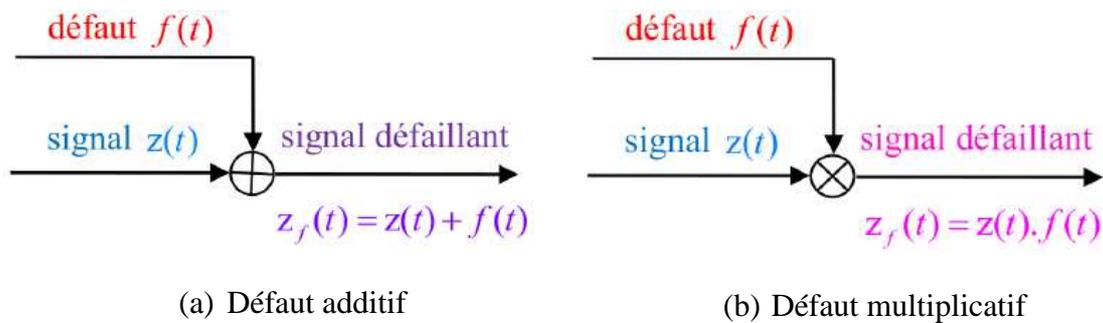


Figure 3. 2 : Défauts additif et multiplicatif.

### 3.6.2.3 Classification selon les caractéristiques temporelles

En fonction de la caractéristique temporelle, les défauts se divisent en défauts graduels et abrupts [43], [44] :

- **Défauts graduels :**

Ils représentent des changements paramétriques lents, qui sont souvent dus au vieillissement. Ils sont difficiles à détecter à cause de leur caractéristique temporelle lente, mais ils ne sont pas sévères. Cependant, si ces défauts ne sont pas pris en charge rapidement, ils peuvent conduire à une situation grave. Les défauts graduels peuvent être modélisés comme suit :

$$f_i(x, u, t) = \beta_i(t - t_{F_i})\delta_i(x, u) \quad (4.3)$$

$$\beta_i(t - t_{F_i}) = \begin{cases} 0, & \text{pour } t < t_{F_i} \\ 1 - e^{-\alpha_i(t-t_{F_i})}, & \text{pour } t \geq t_{F_i} \end{cases} \quad \begin{array}{l} \text{Absence de défaut} \\ \text{Présence de défaut} \end{array} \quad (4.4)$$

Où  $f_i(x, u, t)$  représente la fonction non linéaire du  $i^{\text{ème}}$  défaut,  $\beta_i(t - t_{F_i})$  caractérise le profil temporel de ce défaut qui se produit à un instant inconnu  $t_{F_i}$ ,  $\alpha_i$  est une constante positive qui représente le taux d'évolution du défaut,  $\delta_i(x, u)$  désigne le changement de la dynamique du système au  $i^{\text{ème}}$  défaut. Cette caractéristique permet au même temps d'avoir des défauts additifs et multiplicatifs ( $\delta_i$  peut-être en fonction du vecteur d'état  $x$  et/ou du vecteur de commande  $u$ ). Il est à noter que les défauts graduels sont caractérisés par des faibles valeurs de  $\alpha_i$ .

- **Défauts abrupts :**

Ils se reproduisent instantanément souvent à la suite d'un dommage matériel. Ces défauts peuvent être très sévères car, s'ils affectent les performances et/ou la stabilité du système bouclé, une réaction rapide du système FTC est nécessaire. Mathématiquement, ces défauts peuvent être modélisés comme un cas particulier des défauts graduels, lorsque la constante  $\alpha_i$  prend des grandes valeurs, ce qui rend le profil temporel  $\beta_i(t - t_{F_i})$  proche d'un échelon unitaire.

$$\beta_i(t - t_{F_i}) = \begin{cases} 0, & \text{pour } t < t_{F_i}, & \text{Absence de défaut} \\ 1, & \text{pour } t \geq t_{F_i}, & \text{Présence de défaut} \end{cases} \quad (4.5)$$

- **Défauts intermittents :**

Ce sont des défauts qui apparaissent et disparaissent à plusieurs reprises. Ils représentent également un cas particulier de défauts abrupt, avec la propriété particulière qu'ils reviennent d'une façon aléatoire à leurs valeurs normales. Ces défauts peuvent être causés par un contact intermittent ou un câblage partiellement endommagé.

### 3.6.3 Classification des systèmes FTC

La commande FTC est une stratégie dans l'architecture et la conception des systèmes de commande permettant de s'assurer qu'un système en boucle fermée peut continuer à fonctionner de manière tolérante face aux défauts de l'actionneur, du capteur ou du processus lié. Lorsque la stabilité prescrite et les indices de performance en boucle fermée sont maintenus malgré l'action des défauts, le système est dit "tolérant aux défauts" et le schéma de commande qui assure la tolérance aux pannes est dit: contrôleur tolérant aux pannes. L'objectif de la conception d'un système FTC est de garantir que le système en boucle fermée conserve la stabilité et une performance acceptable pendant une ou plusieurs actions de panne.

Selon la manière avec laquelle le problème est abordé, les systèmes FTC sont généralement divisés en deux classes : **passive** et **active**. Le choix de l'approche à utiliser dépend de la capacité à déterminer les défauts qu'un système peut subir à la phase de conception, des changements induits par le défaut et du type de redondance utilisé dans le système.

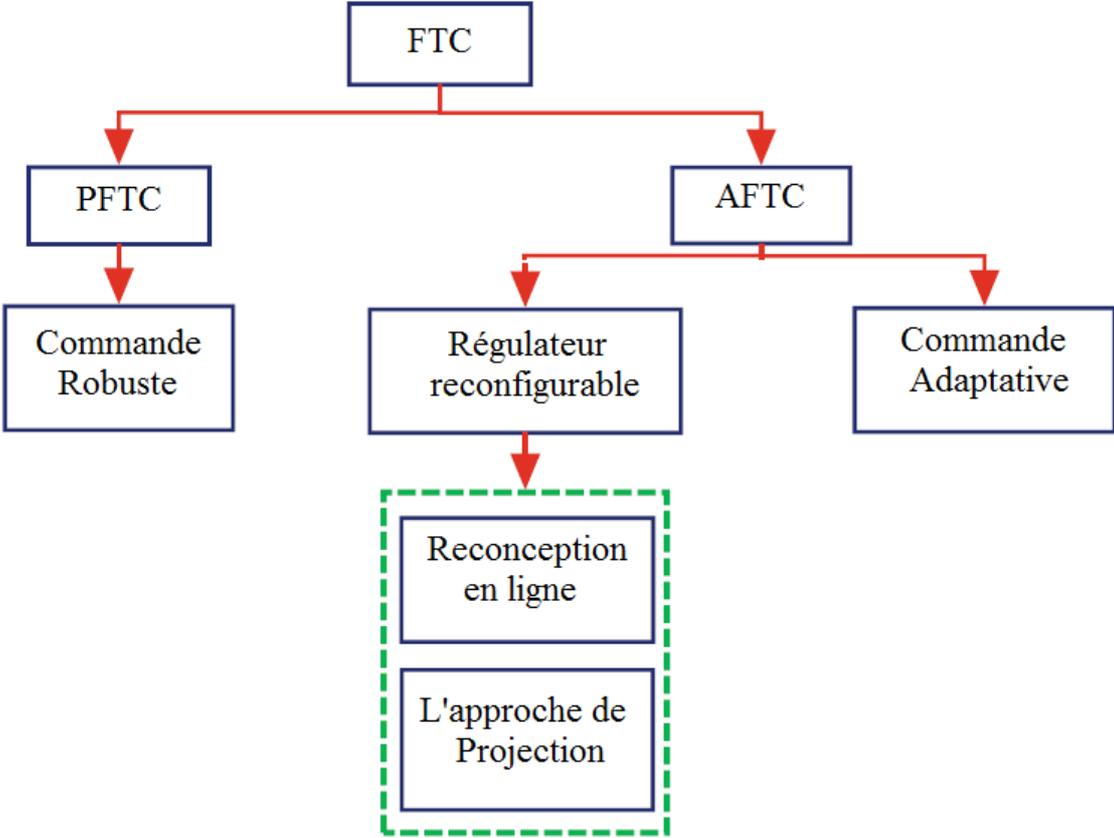


Figure 3. 3 : Classification des FTC.

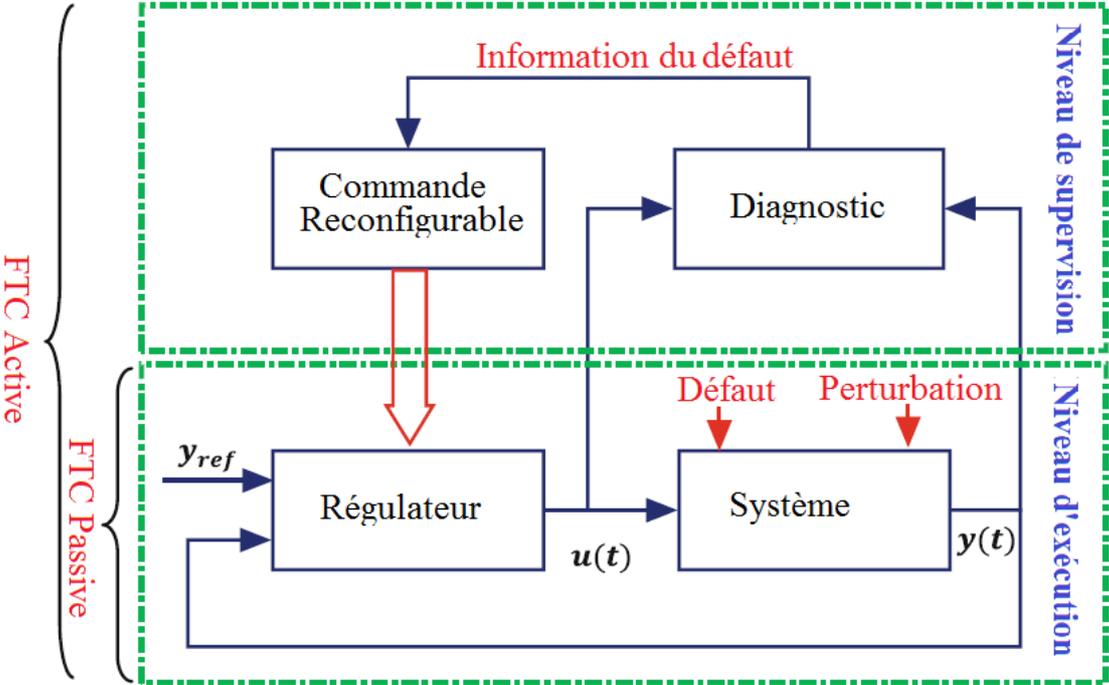


Figure 3. 4 : Le schéma de la commande FTC [46].

Dans ce qui suit, nous allons appliquer la commande  $\mathcal{L}_1$  adaptative sur le quadri-rotor comme une approche AFTC (Active Fault Tolerant Control ) adaptative pour les défauts actionneurs et capteurs.

### 3.6.4 Commande $\mathcal{L}_1$ adaptative tolérante aux défauts d'actionneurs

#### 3.6.4.1 Formulation du problème

Soit le modèle du quadri-rotor (3.10) :

$$\dot{x}(t) = F(x(t)) + B\omega u(t)$$

La relation entre la commande  $u$  et les vitesses (l'effet de l'actionneur) des moteurs (actionneurs)  $\bar{\omega}_i$  (2.27) est :

$$u = \begin{bmatrix} b & b & b & b \\ 0 & -lb & 0 & lb \\ -lb & 0 & lb & 0 \\ -k & k & -k & k \end{bmatrix} \begin{bmatrix} \bar{\omega}_1^2 \\ \bar{\omega}_2^2 \\ \bar{\omega}_3^2 \\ \bar{\omega}_4^2 \end{bmatrix}$$

Soit :

$$\Delta = \begin{bmatrix} b & b & b & b \\ 0 & -lb & 0 & lb \\ -lb & 0 & lb & 0 \\ -k & k & -k & k \end{bmatrix}, \bar{\omega}^2 = [\bar{\omega}_1^2 \bar{\omega}_2^2 \bar{\omega}_3^2 \bar{\omega}_4^2]^T \quad (4.6)$$

Le défaut rencontré par les actionneurs est la perte d'efficacité (loss of effectiveness), tel que  $\bar{\omega} \neq \bar{\omega}_d$ ,  $\bar{\omega}_d$ : la vitesse désirée.

Ce défaut est modélisé par :

$$u = \delta(t)\Delta\bar{\omega}^2 = \delta u_c, \delta(t) = \text{diag}[\delta_1(t) \delta_2(t) \delta_3(t) \delta_4(t)] \quad (4.7)$$

Où  $\delta_i(t) \in [\underline{\delta}_i \ 1]$  est le coefficient d'efficacité du  $i^{\text{ème}}$  actionneur,  $u_c = -KX + u_{ad}$  est la commande calculée.

Le système (3.10) devient :

$$\dot{x}(t) = F(x(t)) + \bar{B}\delta(t)u_c(t); \bar{B} = B\omega \quad (4.8)$$

#### 3.6.4.2 La loi de commande

Soit le prédicteur d'état :

$$\hat{\dot{x}}(t) = A_m\hat{x}(t) + \bar{B}\hat{\delta}(t)u_{ad}(t) + \hat{\eta}_u(t) \quad (4.9)$$

La surface de glissement :

On prend la surface de glissement (3.35) :  $\sigma(t) = \lambda\tilde{x}(t)$ ;  $\lambda \in R^{4 \times 12}$

Les lois de commande :

$$\hat{\eta}_u = -\hat{L}_u(t) \frac{\lambda^T \sigma(t)}{\|\lambda^T \sigma(t)\|}$$

$$\hat{L}_u(t) = \Gamma \|\lambda^T \sigma(t)\| - \varepsilon_u \hat{L}_u(t), \hat{L}_u(0) = L_{u0}, 0 < \varepsilon_u < 1$$

$$\hat{\delta}(t) = -\Gamma(u(t)\sigma^T(t)\lambda\bar{B})^T \quad (4.10)$$

La loi de commande (3.33) :

$$u_{ad}(s) = K_d D(s)(K_g r(s) - \hat{v}_1(s) - \hat{v}_2(s))$$

$\hat{v}_1(s)$  : la transformée de Laplace du terme :  $\hat{\delta}(t)u(t) + \hat{\eta}_m(t)$ .

$$\hat{v}_2(s) = H_m^{-1}(s)H_0(s)\hat{\eta}_u(s)$$

Dans ce cas la commande adaptative va maintenir les performances, en estimant les coefficients d'efficacité de chaque moteur et en ajustant les gains de la commande automatiquement pour compenser les défauts.

On va simuler le système sans et avec les effets du vent. On prend  $\delta_1 = \delta_2 = \delta_3 = \delta_4 = cte$ , dans ce cas les défauts n'influent que sur la position "z", les commandes  $u_2, u_3$  et  $u_4$  tendent rapidement vers 0 en régime statique.

On suppose que les capteurs de vitesse et les pré-actionneurs des quatre moteurs sont parfaits.

### 3.6.5 Commande $\mathcal{L}_1$ adaptative tolérante aux défauts actionneurs et capteurs

Dans cette partie nous considérons les hypothèses suivantes :

- Tous les états du système sont mesurables.
- Les défauts sont seulement dans les capteurs des sorties.
- Le système suit parfaitement le prédicteur d'état.

#### 3.6.5.1 Formulation du problème

Soit le modèle des capteurs suivant :

$$x_s = F(s).Y \quad (4.11)$$

Où  $x_s \in R^{4 \times 1}$  délivré par les capteurs,  $F(s) = \frac{1}{T_s s + 1} I_4$  est la matrice de transfert des capteurs,  $T_s$  est la constante de temps de chaque capteur.

Le modèle des capteurs dans l'espace d'état :

$$x_s(s) = \frac{1}{T_s s + 1} I_4 . Y(s)$$

Après développement :

$$\dot{x}_s = -A_s x_s + A_s Y, \quad A_s = \frac{1}{T_s} I_4 \quad (4.12)$$

Si le défaut existe [38] :

$$\dot{x}_s(t) = -A_s x_s(t) + A_s Y(t) + f \quad (4.13)$$

$f \in R^{4 \times 1}$  : L'erreur de mesure.

### 3.6.5.2 La loi de commande

Pour le système de commande, on garde la même commande appliquée dans la partie précédente (4.5.1).

Dans le but d'estimer l'erreur de mesure, nous ajoutons un sous-système de commande basé sur la commande  $\mathcal{L}_1$  adaptative, cette erreur est prise en considération dans le calcul de la commande en l'ajoutant au retour de sortie.

- **Le sous-système de commande :**

Soit le prédicteur d'état suivant :

$$\dot{\hat{x}}_s(t) = A_{ms} \hat{x}_s(t) + A_s u_{ads}(t) + \hat{f}(t) \quad (4.14)$$

$A_{ms} = -A_s(I_4 + K_s)$ ,  $K_s$  matrice gain calculée qui minimise le critère quadratique :

$$J_s = \int_0^{+\infty} (x_s^T Q_s x_s + u_s^T R_s u_s) dt \quad (4.15)$$

- **La loi d'adaptation :**

La surface de glissement :

$$\sigma_s(t) = \lambda_s \tilde{x}_s(t); \lambda_s \in R^{4 \times 4}, \quad \tilde{x}_s(t) = \hat{x}_s(t) - x_s(t) \quad (4.16)$$

$$\hat{f}(t) = \hat{\eta}_s(t) = -\hat{L}_s(t) \frac{\lambda_s^T \sigma_s(t)}{\|\lambda_s^T \sigma_s(t)\|} \quad (4.17)$$

$$\hat{L}_s(t) = \Gamma_s \|\lambda_s^T \sigma_s(t)\| - \varepsilon_s \hat{L}_s(t), \hat{L}_s(0) = L_{s0}, 0 < \varepsilon_s < 1$$

$$\hat{f}(t) = \int \hat{\eta}_s(t) dt \quad (4.18)$$

L'estimation  $\hat{f}(t)$  est ajoutée au retour de sortie  $y(t)$  pour la compenser.

$$y(t) = C x_s(t) + \hat{f}(t) \quad (4.19)$$

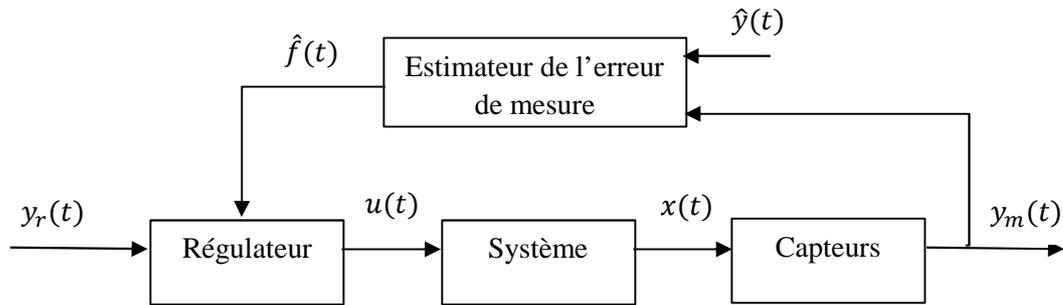


Figure 3. 5: Architecture de commande FTC capteurs

### 3.7 Conclusion

Nous avons conçu deux architectures de commandes  $\mathcal{L}_1$  adaptatives différentes, une première approche classique basée sur la descente du gradient limitée par sa sensibilité vis-à-vis des bornes de perturbations, et une seconde approche basée sur les modes glissants et qui permet d'estimer les bornes des perturbations. Aussi, nous avons développés deux structures de commandes AFTC basée sur la commande  $\mathcal{L}_1$  adaptative afin de faire face aux défauts actionneurs ou capteurs qui peuvent atteindre notre système rajoutant ainsi un autre niveau de robustesse. Les résultats des simulations sont présentés au chapitre 5.

# **Chapitre 4. Implémentation de la commande $\mathcal{L}_1$ du Quadri-rotor sur l'environnement ROS**

## 4.1 Introduction

L'apprentissage et la conception de nouvelles commandes de vol pour les drones est parfois difficile à cause des pertes récurrentes de composants coûteux durant les tests, pour répondre à cette difficulté, l'équipe Hector Darmstadt a développé l'Hector Quadrotor ; un drone de type quadri-rotor simulé sous l'environnement ROS Gazebo et permettant de tester de nouveaux algorithmes de commande et d'enregistrer et analyser les données de capteurs avant de passer aux tests réels.

Dans ce chapitre, nous allons présenter une nouvelle méthode, simple permettant de simuler et générer un " Standalone ROS node " directement à partir du modèle de commande Simulink afin d'implémenter la commande  $\mathcal{L}_1$  adaptative sur le drone Hector Quadrotor ce qui nous permettra par la suite de tirer des conclusions sur les performances de notre commande et le comportement du système dans un environnement réaliste tel que Gazebo.

## 4.2 Robot Operating System (ROS)

ROS est une plateforme flexible de développement de logiciels collaboratifs pour robots. Il s'agit d'un ensemble d'outils et de bibliothèques permettant de simplifier la conception de robots à comportement complexe et robuste. La nature distribuée de ROS favorise la contribution de la communauté de ses utilisateurs pour le développement de nouveaux packages ajoutant ainsi beaucoup de valeur au système ROS de base [53].

### 4.2.1 Les outils de ROS

- **Rviz**

C'est un outil de ROS permettant une visualisation tridimensionnelle de nombreux types de données. Rviz peut visualiser la plupart des types de messages courants disponibles dans ROS, tels que les scans laser, les nuages de points tridimensionnels et les images de caméra. La visualisation de toutes vos données dans la même application vous permet de voir ce que voit votre robot instantanément et donc, d'identifier les problèmes tels que les défauts d'alignement des capteurs ou les inexactitudes des modèles de robots [53].

- **Rqt**

*Rqt* est une plateforme ROS pour le développement d'interfaces graphiques pour votre robot, elle permet de créer des interfaces personnalisées en composant et en configurant la vaste bibliothèque de plugins *rqt* intégrés, elle permet aussi à l'utilisateur d'écrire ses propres plugins *rqt*.

Parmi les plugins que fournit *rqt*, on trouve le *rqt\_graph* qui permet de visualiser les nœuds en cours d'exécution ainsi que les connections entre eux, *rqt\_plot* qui permet de représenter

les données numériques variant dans le temps par des courbes, *rqt\_topic* et *rqt\_publisher* permettent de surveiller et de publier des messages sur les topics [53].

- **Gazebo**

Gazebo, c'est est un simulateur 3D multi-robots et open source, il offre la possibilité de simuler avec précision et efficacité des populations de robots dans des environnements intérieurs et extérieurs complexes, il génère des informations réalistes. Gazebo fournit un robuste moteur physique, des graphiques de haute qualité et des interfaces programmatiques et graphiques pratiques. L'intégration entre ROS et Gazebo est assurée par un ensemble de plugins Gazebo qui prennent en charge de nombreux robots et capteurs existants [53], [55].

#### 4.2.2 La communication sous ROS

L'échange d'informations est un élément clé dans la commande des systèmes, pour cela; un système de communication est un besoin impérieux lors de la mise en œuvre d'une nouvelle application de robot. Le système de messagerie intégré de ROS permet de gérer les détails de la communication entre les nœuds répartis via le mécanisme de publication / abonnement anonyme. Dans ce qui suit, on définira quelques notions de base de ROS [53], [54] :

- **Le master**

Le master est le processus cœur de ROS, il fournit un serveur de paramètres ainsi qu'un serveur de noms et d'enregistrement aux autres nœuds du système ROS. Le rôle du master est de permettre aux nœuds de se localiser les uns par rapport aux autres afin qu'ils puissent communiquer entre eux (Figure 4.1). Le master est le plus souvent exécuté en utilisant la commande *roscore*, qui charge le master ROS avec d'autres composants essentiels.

- **Le nœud**

Un nœud est une instance d'un exécutable, il peut correspondre à un capteur, un moteur, un algorithme... Chaque nœud qui se lance se déclare au Master. Les nœuds sont combinés dans un graphique et communiquent entre eux à l'aide des topics et services. Un système de commande comprend généralement de nombreux nœuds. L'utilisation des nœuds dans ROS offre plusieurs avantages à l'ensemble du système tel que la réduction de la complexité du code par rapport aux systèmes monolithiques.

- **Les topics**

Les topics sont des bus asynchrones sur lesquels les nœuds échangent des messages. Ils se basent sur le principe de publication / abonnement anonyme (subscribe / publish). Il peut y avoir plusieurs nœuds qui publient et qui s'abonnent au même topic, c'est le principe de la communication many-to-many.

- **Le service**

Le service assure une communication synchrone entre deux nœuds, il peut être défini comme une paire de messages : une requête et une réponse. Les services sont déclarés dans des fichiers *srv* précisant les types des entrées et des sorties.

- **Le message**

Un message est une structure de données comprenant des champs typés (entier, booléen, réel...), des tableaux et des structures. Les messages utilisés sur les topics sont déclarés dans des fichiers *msg* tandis que les messages échangés sur un service sont définis dans des fichiers *srv*.

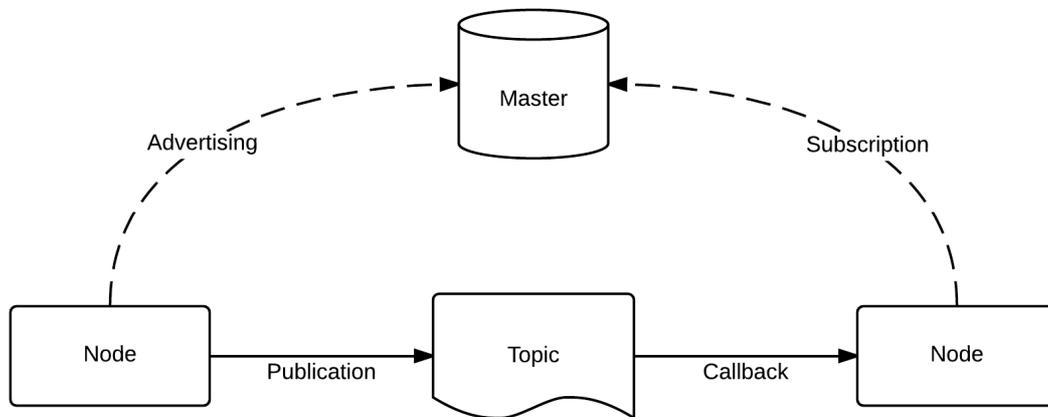


Figure 4. 1 : Schéma de fonctionnement du master sous ROS

### 4.2.3 Hector\_quadrotor

Hector\_quadrotor contient des packages liés à la modélisation, au contrôle et à la simulation des systèmes de drones quadri-rotors sous Gazebo. Les packages suivants sont disponibles [53] :

- **Hector\_quadrotor\_description** fournit un modèle URDF (Universal Robotic Description Format) quadri-rotor générique ainsi que des variantes avec divers capteurs.

- **Hector\_quadrotor\_gazebo** contient les fichiers de lancement nécessaires et les informations de dépendance pour la simulation du modèle de quadri-rotor dans Gazebo

- **Hector\_quadrotor\_teleop** contient un nœud qui permet le contrôle du quadri-rotor à l'aide d'une manette.

- **Hector\_quadrotor\_gazebo\_plugins** fournit des plugins spécifiques à la simulation de drones quadri-rotors dans la simulation sous Gazebo.

## 4.3 Robotics System Toolbox

La Robotics System Toolbox fournit une interface entre MATLAB / Simulink et Robot Operating System (ROS) qui permet de tester et simuler des applications sur des robots fonctionnant sous ROS, ou bien sur des simulateurs tels que Gazebo. Elle contient des algorithmes pour le développement d'applications de robotique pour les véhicules autonomes aériens et terrestres.

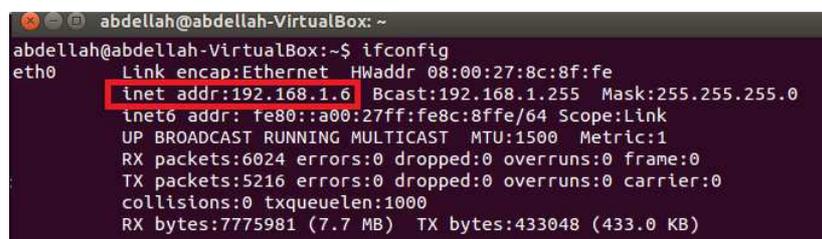
La Robotics System Toolbox peut générer du code C++ à partir d'un modèle Simulink, ce qui permet de générer un nœud ROS à partir du modèle de commande Simulink et de le déployer sur ROS. Elle permet de visualiser les signaux provenant du robot et en temps réel à partir de Simulink et de modifier les paramètres du modèle de commande même pendant son exécution. La Robotics System Toolbox a été lancée en premier sur la version 2015b de Matlab [56].

### 4.3.1 Connexion Simulink/ ROS

Pour que la connexion ROS s'établisse correctement, il faut s'assurer que tous les nœuds peuvent communiquer avec le master et entre eux. Les nœuds doivent communiquer avec le master pour enregistrer les *subscribers*, les *publishers* et les services. Ils doivent également être en mesure de communiquer entre eux pour envoyer et recevoir des données. Comme la communication fonctionne de cette manière, il est possible d'envoyer des données et de ne pas pouvoir les recevoir (ou vice versa) si votre réseau ROS n'est pas configuré correctement, pour éviter ces problèmes, il faut suivre les étapes 1 à 7 décrites dans le paragraphe suivant.

Matlab/ Simulink fonctionne sur notre machine Windows, afin de pouvoir les connecter à ROS, on a choisi d'installer ROS sur une machine virtuelle Linux, Ubuntu (ROS n'est pas compatible avec Windows). On doit en premier lieu effectuer une configuration sur les deux machines en suivant les étapes ci-dessous :

- Récupérer les adresses IP de la machine Windows (avec *ipconfig* à partir de l'invite de commande) ainsi que celle de la machine virtuelle Linux (avec la commande *ifconfig* qu'on lance sur le terminal Linux).



```
abdellah@abdellah-VirtualBox: ~
abdellah@abdellah-VirtualBox:~$ ifconfig
eth0      Link encap:Ethernet  HWaddr 08:00:27:8c:8f:fe
          inet addr:192.168.1.6  Bcast:192.168.1.255  Mask:255.255.255.0
          inet6 addr: fe80::a00:27ff:fe8c:8ffe/64  Scope:Link
          UP BROADCAST RUNNING MULTICAST  MTU:1500  Metric:1
          RX packets:6024 errors:0 dropped:0 overruns:0 frame:0
          TX packets:5216 errors:0 dropped:0 overruns:0 carrier:0
          collisions:0 txqueuelen:1000
          RX bytes:7775981 (7.7 MB)  TX bytes:433048 (433.0 KB)
```

Figure 4. 2 : Adresse IP de la machine virtuelle Linux

- Sur la machine Ubuntu, accéder au fichier *hosts* via la commande `sudo gedit /etc/hosts` et rajouter les hostnames ainsi que les adresses IP correspondantes aux deux machines.

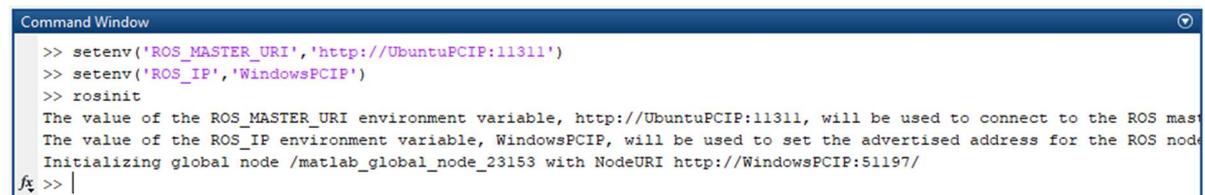
- Sur la machine Windows, ouvrir le Bloc-notes en tant qu'administrateur et puis accéder au fichier C:\Windows\System32\drivers\etc\hosts (choisir « Tous les fichiers » pour que le fichier hosts apparaisse) et faire la même chose qu'en l'étape 2.
- Désactiver le Pare-feu sur les deux machines.
- Redémarrer les deux machines pour que les changements soient pris en compte.
- Accéder au *bashrc* via la commande *sudo gedit ~/.bashrc* et rajouter les lignes suivantes :

```
export ROS_MASTER_URI=http://UbuntuPCIP:11311  
  
export ROS_IP=UbuntuPCIP
```

- Sur Matlab, définir les variables ROS via les commandes suivantes :

```
setenv('ROS_MASTER_URI','http://UbuntuPCIP:11311')  
  
setenv('ROS_IP','WindowsPCIP')  
  
rosinit
```

Le message ci-dessous indiquant que la communication au ROS master a été effectuée s'affiche.



```
Command Window  
>> setenv('ROS_MASTER_URI','http://UbuntuPCIP:11311')  
>> setenv('ROS_IP','WindowsPCIP')  
>> rosinit  
The value of the ROS_MASTER_URI environment variable, http://UbuntuPCIP:11311, will be used to connect to the ROS master.  
The value of the ROS_IP environment variable, WindowsPCIP, will be used to set the advertised address for the ROS node.  
Initializing global node /matlab_global_node_23153 with NodeURI http://WindowsPCIP:51197/  
fx >> |
```

**NB :**

- A la place de « UbuntuPCIP » et de « WindowsPCIP », il faut mettre les hostnames choisis.
- Les étapes 1, 2 et 3 sont à refaire dans le cas où les adresses IP changent, sinon ; un message d'erreur s'affiche au lancement de ROS.
- L'étape 7 doit être effectuée à chaque fois qu'on veut connecter Matlab avec ROS.

### 4.3.2 Echange de données entre Simulink et ROS

Le contrôleur (nœud correspondant à la commande  $\mathcal{L}_1$ ) reçoit les informations relatives à la position du quadri-rotor simulé sous Gazebo et envoie des commandes en vitesse pour ramener le quadri-rotor à la position désirée. Le diagramme suivant (Figure 4.3) montre l'interaction entre Simulink et le simulateur 3-D Gazebo.

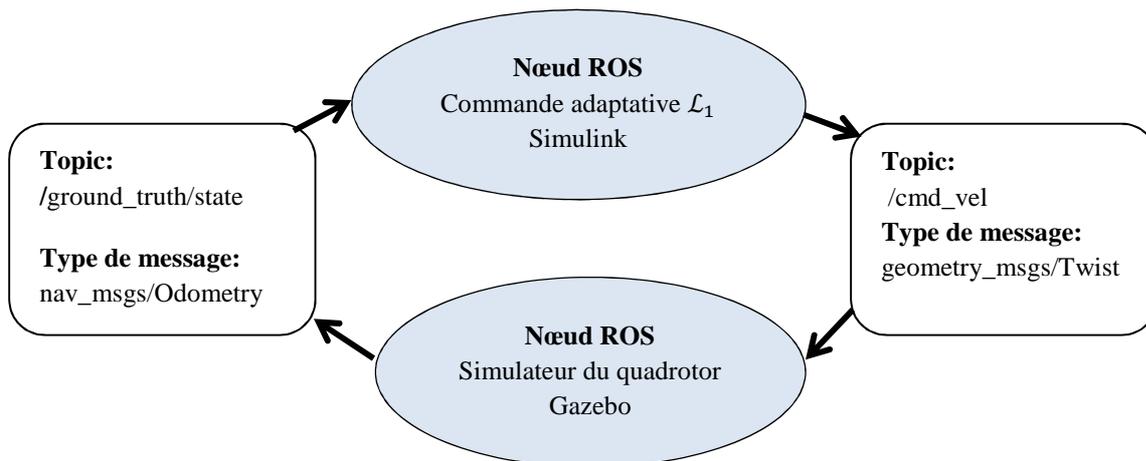


Figure 4. 3 : Interaction entre Simulink et Gazebo

Les flèches dans le diagramme indiquent la transmission de messages ROS. Le topic `"/ odom"` transmet les informations de localisation tandis que le topic `"/ cmd_vel"` transmet les commandes en vitesses.

- **Publication de commandes Simulink sur ROS (Gazebo)**

La transmission de commandes de Simulink vers Gazebo (ROS) passe par un bloc Publish (Figure 4.4), ce dernier reçoit en entrée un bus correspondant au message spécifié, et le publie sur le topic choisi. Un bloc Blank Message doit être connecté au bloc publish via un Bus Assignment. Le bloc blank message crée un bus de données correspondant au type de message sélectionné. Le Bus Assignment permet de sélectionner les éléments du bus de donnée auxquels seront affectées les nouvelles valeurs présentes en entrée [56].

Le vecteur commande d'un quadri-rotor est composé de la portance et de trois couples, cependant ; sur Gazebo, l'Hector Quadrotor ne peut être commandé qu'en publiant des messages "Twist" sur le topic "cmd\_vel". Ce message contient une composante linéaire pour les vitesses de translation selon les axes (x, y, z), et une composante angulaire pour les vitesses de rotation autour des axes (x, y, z).

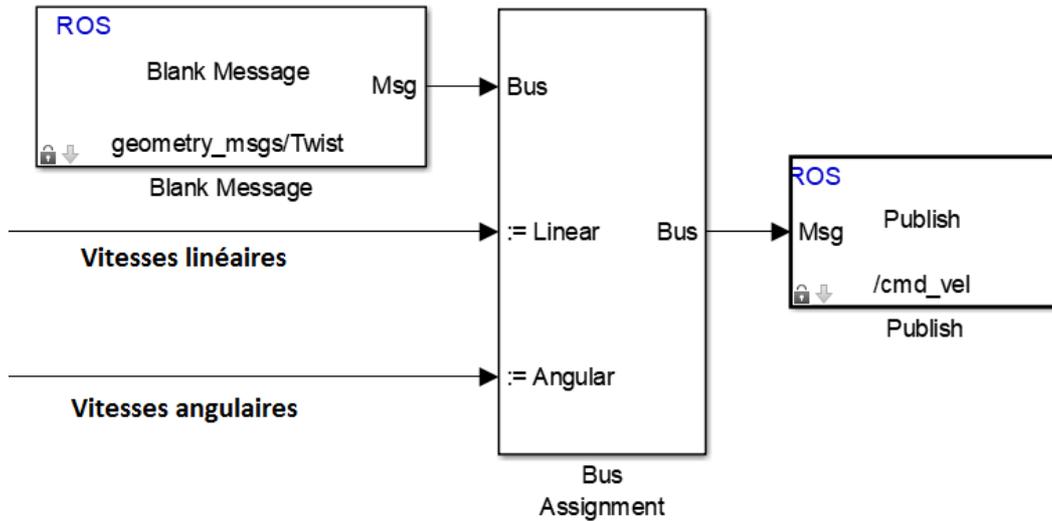


Figure 4. 4: La publication de donnée sur ROS

- **Réception de donnée à partir de ROS**

Pour avoir une boucle de commande fermée, un retour de données est nécessaire, pour cela ; on doit reconstruire le vecteur d'état à partir des mesures qu'on peut recevoir de l'Hector Quadrotor via le bloc "Subscribe" (Figure 4.5). Ce dernier permet de s'abonner à un topic donné selon le message ou l'information que nous souhaitons récupérer.

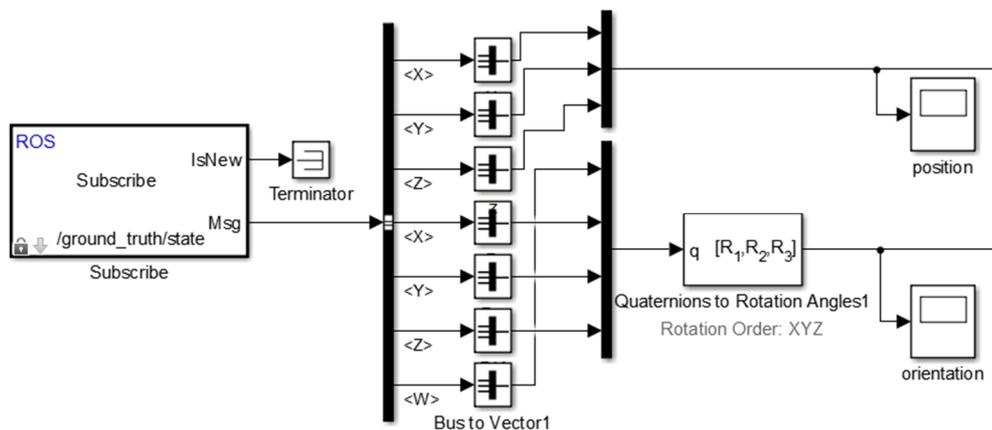


Figure 4. 5: Abonnement à un topic ROS.

Dans notre simulation, nous avons utilisé le bloc "Subscribe" pour s'abonner au topic "ground\_truth/state" pour récupérer des messages de type "Odometry". Ce message contient la position (x, y, z) du quadri-rotor, son orientation (x, y, z, w) donnée en quaternions, ainsi que ses vitesses linéaires et angulaires. On utilise aussi un bloc "Quaternions to Rotation Angles" afin d'exprimer l'orientation du quadri-rotor avec les angles d'Euler.

### 4.3.3 Générer un nœud ROS à partir de Simulink

Générer un nœud ROS (standalone ROS node) à partir de Simulink consiste à générer le code C++ du nœud ROS à partir du modèle de commande Simulink. Les étapes à suivre sont [56] :

- Copier le modèle Simulink dans un nouveau fichier puis aller sur Model Configuration Parameters et exécuter les trois étapes suivantes.
- Activer la fonction "variable-sized arrays" sur Model Configuration Parameters > Code Generation > Interface > Software environment > Support > variable-size signals.
- Dans le volet Hardware Implementation, définir "Robot Operating System (ROS)" Comme Hardware board, puis cliquer sur Build options et choisir Build and run, cette dernière action permet de lancer automatiquement le nœud correspondant au model Simulink sur ROS une fois sa construction est terminée (Figure 4.6).
- Utiliser un pas de simulation fixe (fixed-step time).

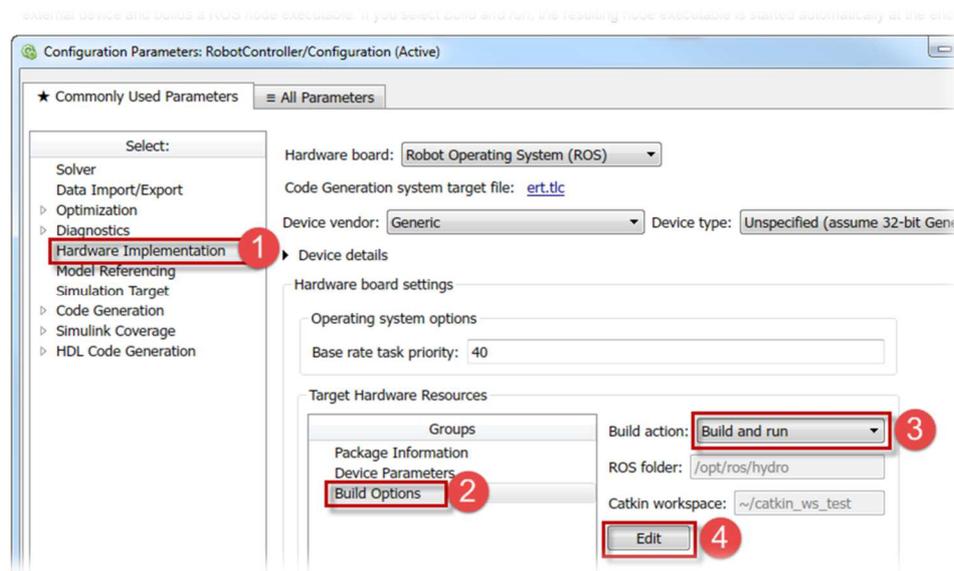


Figure 4. 6: Configuration du modèle Simulink

Ensuite, il faut vérifier la connexion entre Matlab/Simulink et ROS, et puis sur Sur Matlab, changer le dossier de travail vers un autre emplacement temporaire, et la dernière étape sera de générer le code C++, pour cela, il faut aller dans Code> C/C++ Code > Build Model. Des erreurs de compatibilité entre les bus de données vont probablement apparaître, une fois corrigées, refaire l'opération.

Une fois le code généré avec succès, le nœud ROS est transféré sur le Catkin workspace, se connecte au ROS master et se lance automatiquement. Suite à cette opération, Simulink crée deux fichiers dans le dossier de travail : un fichier archive contenant le code source C ++ (Nom\_du\_modèle\_Simulink.tgz) et un script *shell* pour l'extraction et la construction du code C ++ manuellement (build\_ros\_model.sh).

## **4.4 Conclusion**

Nous avons vu comment générer un nœud ROS directement à partir d'un modèle de commande Simulink, cette nouvelle méthode est plus pratique que l'ancienne approche qui nécessitait de réécrire la commande manuellement en code C++ ou Python dans ROS, elle permet donc de gagner du temps et d'agir sur la simulation en temps réel en modifiant les paramètres de commande à partir de Simulink. Cependant ; les retards de simulation et les problèmes de synchronisation (Voir chapitre 5) doivent être pris en compte car ils risquent de nuire aux résultats de la simulation.

# **Chapitre 5.**

## **Simulations et résultats**

## 5.1 Introduction

Afin de valider les différentes architectures et configurations de commandes précédemment conçues, nous avons consacré ce chapitre pour présenter les résultats des simulations effectuées sur MATLAB/ Simulink en l'absence et en présence des perturbations atmosphériques, aussi, les résultats des simulations réalistes effectuées sous l'environnement Gazebo sont présentées à la fin.

## 5.2 Résultats des simulations avec Descente du gradient

### 5.2.1 Simulations en l'absence des perturbations atmosphériques

Afin de valider et tester la loi de commande synthétisée, nous présentons dans cette partie les résultats des simulations effectuées sur la commande  $\mathcal{L}_1$  adaptative par descente du gradient en l'absence des perturbations. Dans les figures suivantes, le réglage à une référence en échelon unitaire a été effectué pour les coordonnées  $x$ ,  $y$ ,  $z$  et à une référence nulle pour l'angle  $\Psi$ . Les erreurs de réglage et les signaux de commande sont présentés aussi.

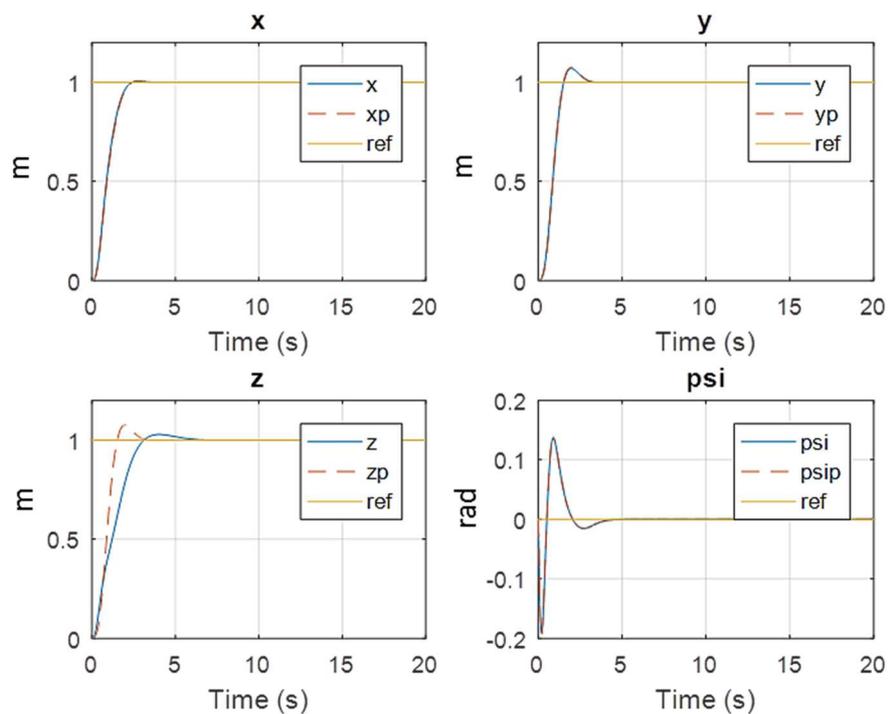


Figure 5. 1: Les réponses du système et du prédicteur.

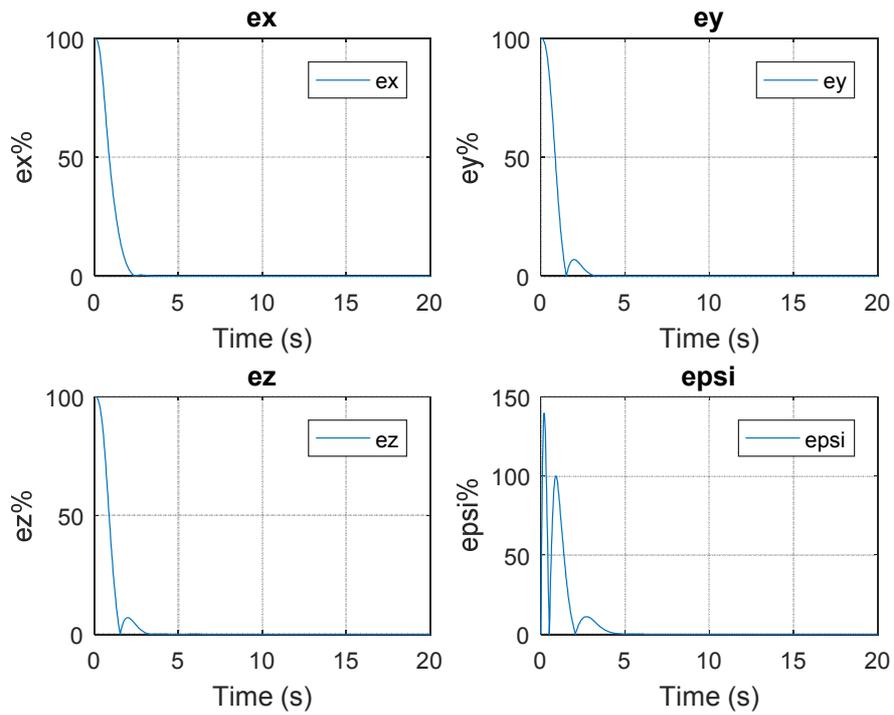


Figure 5. 2: Les erreurs de poursuite.

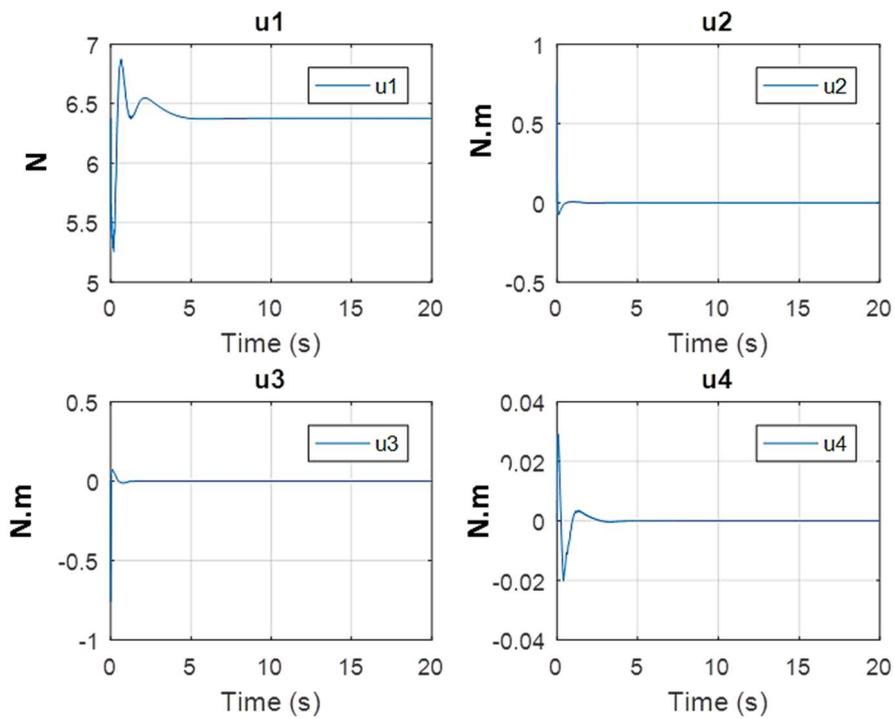


Figure 5. 3: Les commandes appliquées.

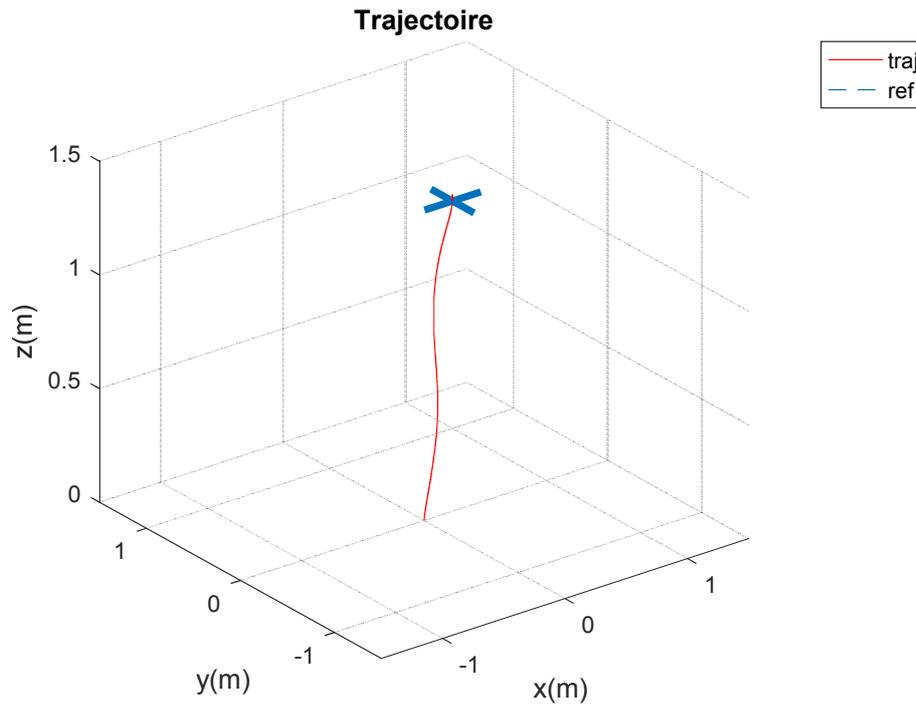


Figure 5. 4 : La trajectoire en 3D.

Nous remarquons que le quadri-rotor est stable et suit bien la trajectoire désirée, les erreurs de réglage convergent vers zéros. De plus le système suit le modèle de référence, nous pouvons voir que les deux sorties sont confondues. De ce fait, nous pouvons conclure que la commande adaptative est robuste, le système considéré inconnu est stable. Les paramètres d'adaptation sont bien choisis pour obtenir un bon résultat, un mauvais choix de paramètres peut nuire à la stabilité du système et/ou les performances.

### 5.2.2 Simulations sous l'effet du vent

En pratique, le quadri-rotor fonctionne sous l'effet du vent qui a une grande influence sur sa dynamique, en particulier les vitesses linéaires et angulaires. Pour être proche de la réalité, nous utilisons le bloc (Dryden Wind Turbulence Model) de Simulink. Ce modèle utilise la représentation spectrale de Dryden pour ajouter de la turbulence au modèle aérospatial en passant le bruit blanc limité par la bande à travers des filtres de formation appropriée. La turbulence est un processus stochastique défini par les spectres de vitesse. Pour un aéronef volant à une vitesse  $V$  à travers un champ de turbulence gelé avec une fréquence spatiale de  $\Omega$  radians par mètre, la fréquence circulaire  $\omega$  est calculée en multipliant  $V$  par  $\omega$  [37].

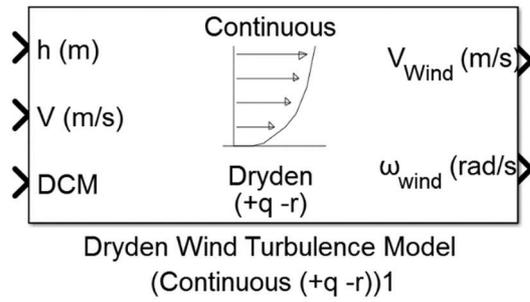


Figure 5. 5: Modèle du vent.

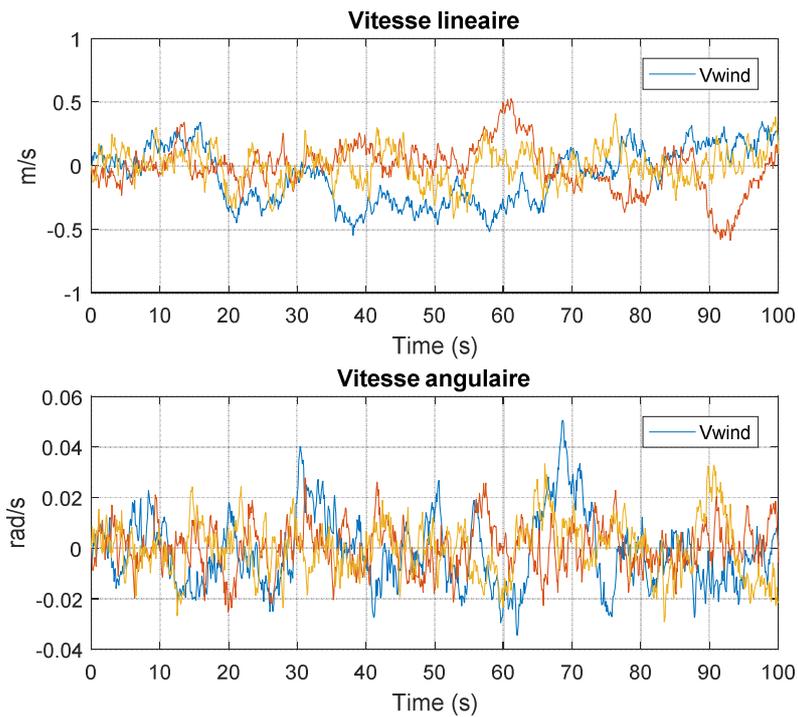


Figure 5. 6: Vitesse du vent.

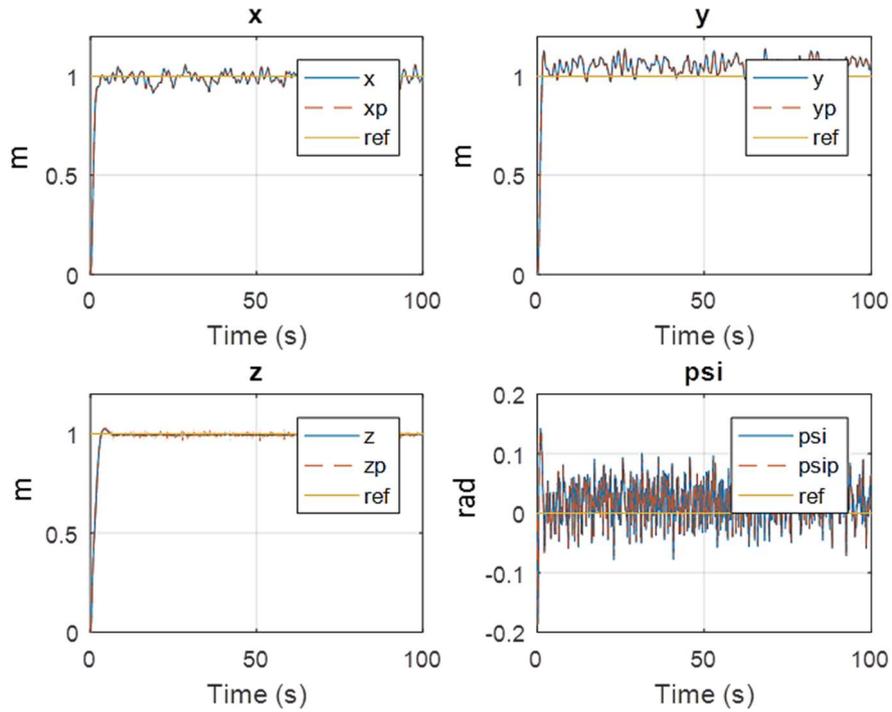


Figure 5. 7: Les réponses du système et du prédicteur.

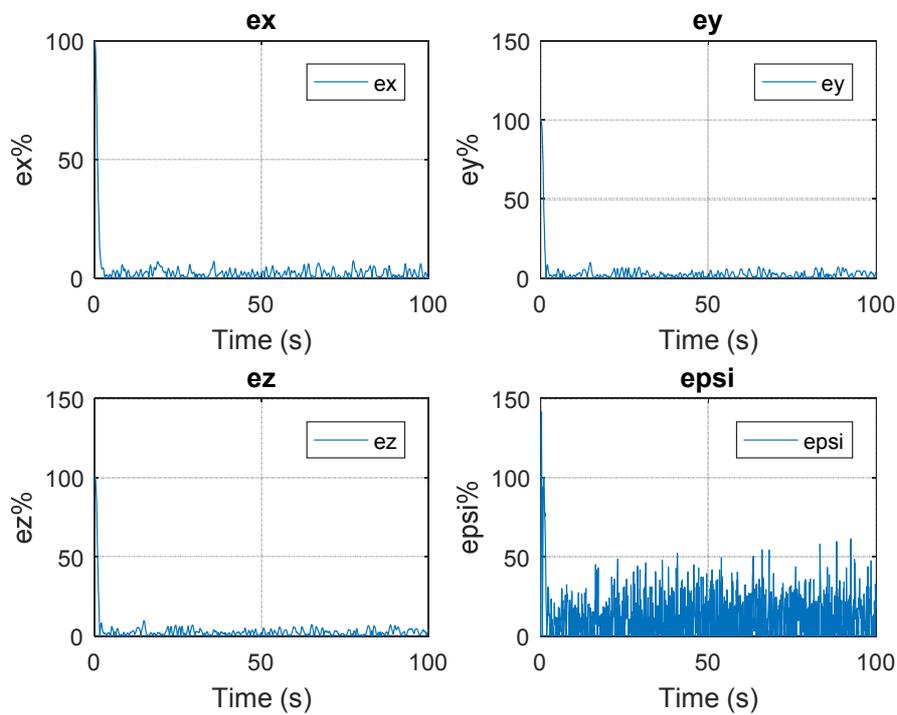


Figure 5. 8: Les erreurs de poursuite.

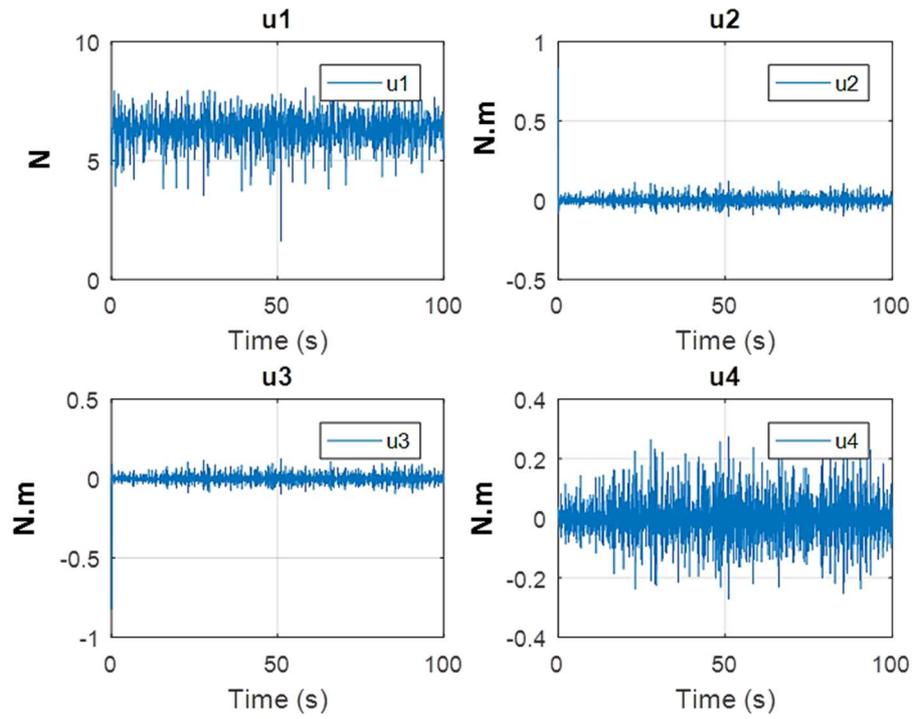


Figure 5. 9 : Les commandes appliquées.

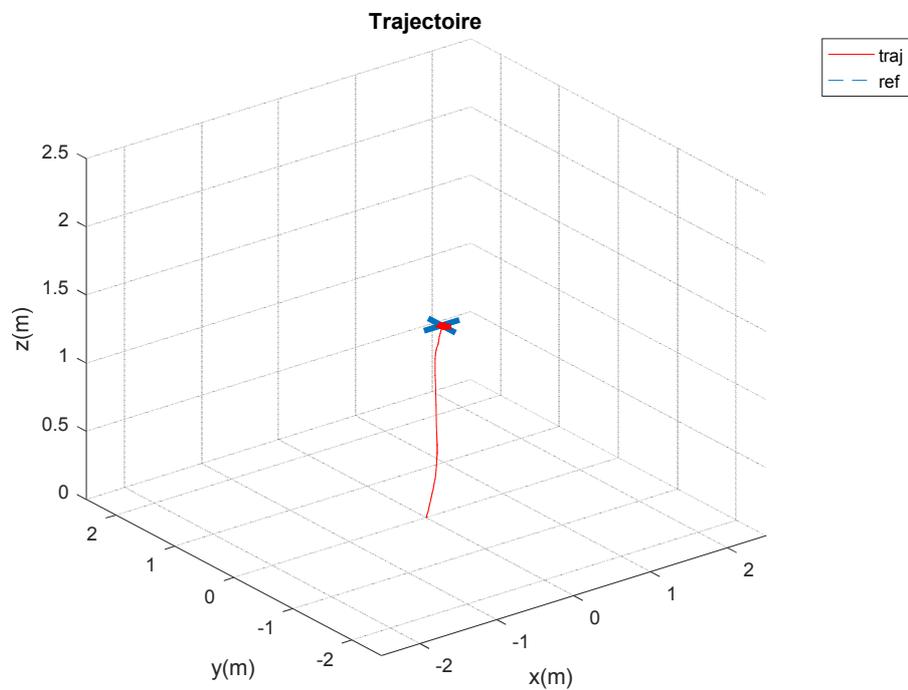


Figure 5. 10: La trajectoire en 3D.

Dans un environnement exposé aux rafales de vent, la commande adaptative a pu maintenir le quadri-rotor au voisinage de la référence avec de faibles erreurs de poursuites. On voit que les commandes oscillent pour compenser les perturbations. En pratique le quadri-rotor vibre sous l'effet du vent, mais il faut noter qu'en pratique, pour un certain seuil de l'amplitude des perturbations atmosphériques, la commande ne peut pas compenser ces perturbations. La commande LQR (placement de pôles) de la boucle interne a stabilisé le système et la commande adaptative compense les perturbations et assure le suivi de trajectoire.

### 5.2.3 La limite de la commande $\mathcal{L}_1$ avec la descente du gradient

La commande  $\mathcal{L}_1$  adaptative est limitée par la connaissance des bornes des perturbations, ce qui n'est pas toujours vérifiée en pratique, ce problème peut nuire à la stabilité et aux performances du système (rejet de perturbation et poursuite). Les simulations suivantes ont été effectuées après avoir modifié les bornes des perturbations de l'opérateur de projection. Afin de bien voir l'effet de ces dernières, tous les autres paramètres ont été maintenus.

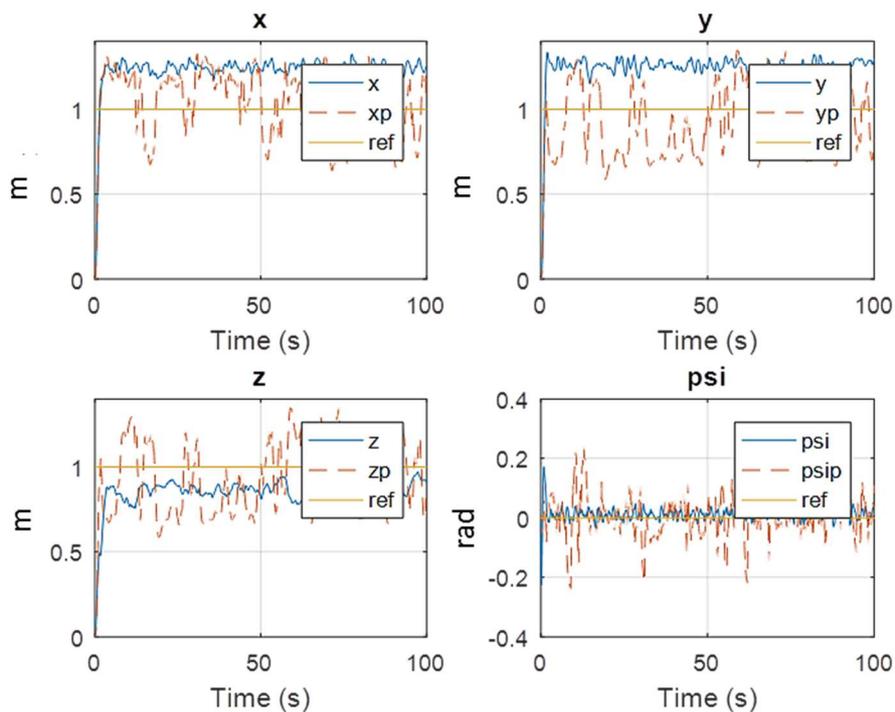


Figure 5. 11: Les réponses du système et du prédicteur.

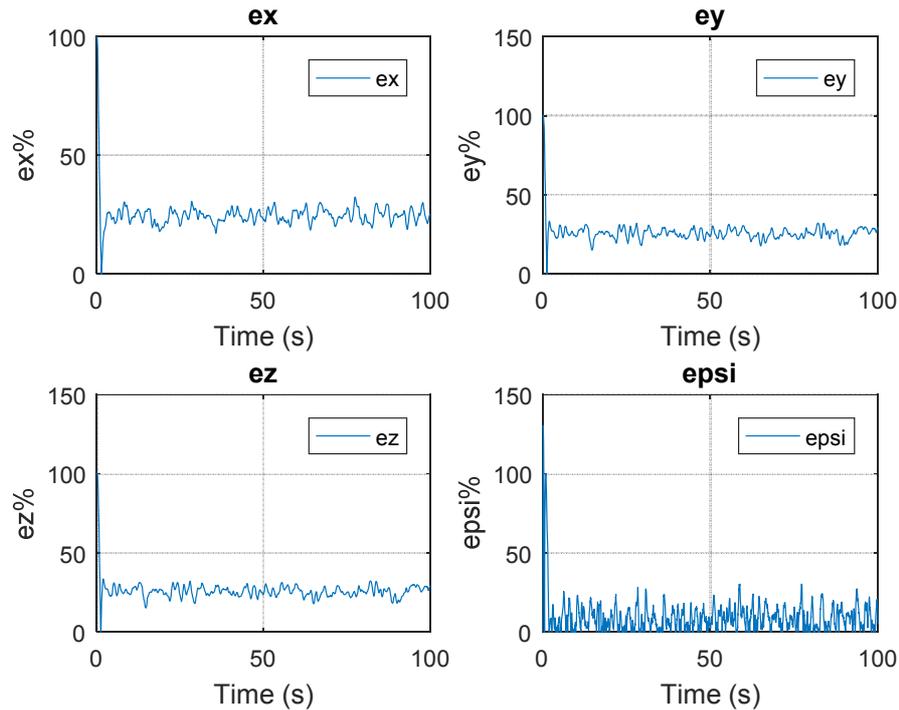


Figure 5. 12: Les erreurs de poursuites.

En changeant les bornes des perturbations le quadri-rotor ne suit plus la trajectoire désirée, donc la commande a perdu sa performance. Pour remédier à ce problème, une autre approche de la commande  $\mathcal{L}_1$  adaptative se basant sur les modes glissants permettra d'estimer les perturbations et leurs bornes.

### 5.3 Résultats des simulations avec Modes Glissants

#### 5.3.1 Simulations en l'absence des perturbations atmosphériques

De même que pour la commande avec descente du gradient, nous présentons ici les résultats de simulations effectuées sur la commande  $\mathcal{L}_1$  adaptative par modes glissants et en l'absence des perturbations. Dans les figures qui suivent, le réglage de position à une référence en échelon unitaire a été effectuées pour les coordonnées x, y, z et d'une référence nulle pour l'angle  $\psi$ . Les erreurs de réglage et les signaux de commande sont présentés aussi.

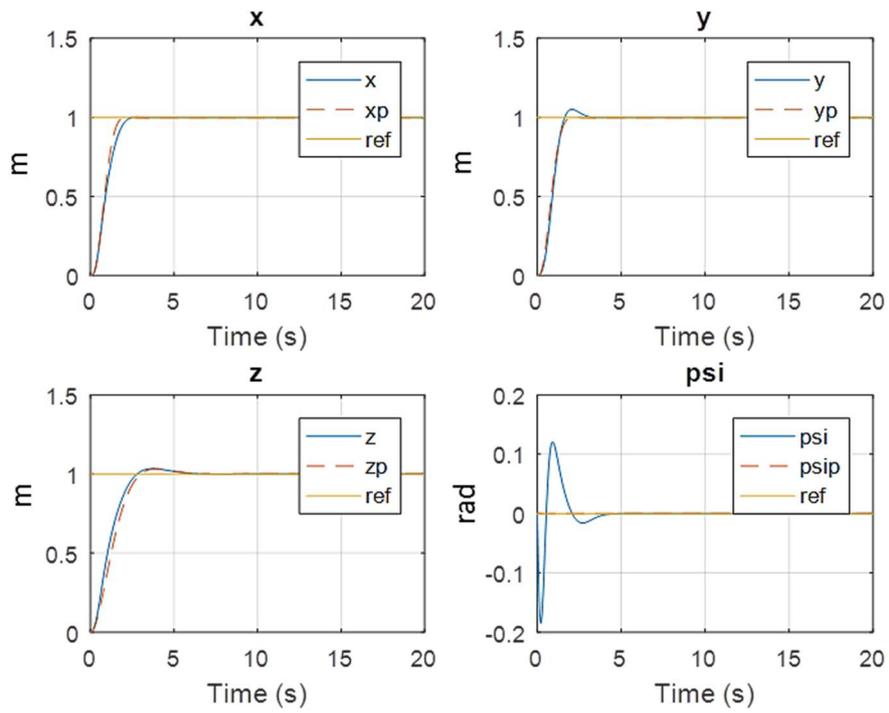


Figure 5. 13: Les réponses du système et du prédicteur.

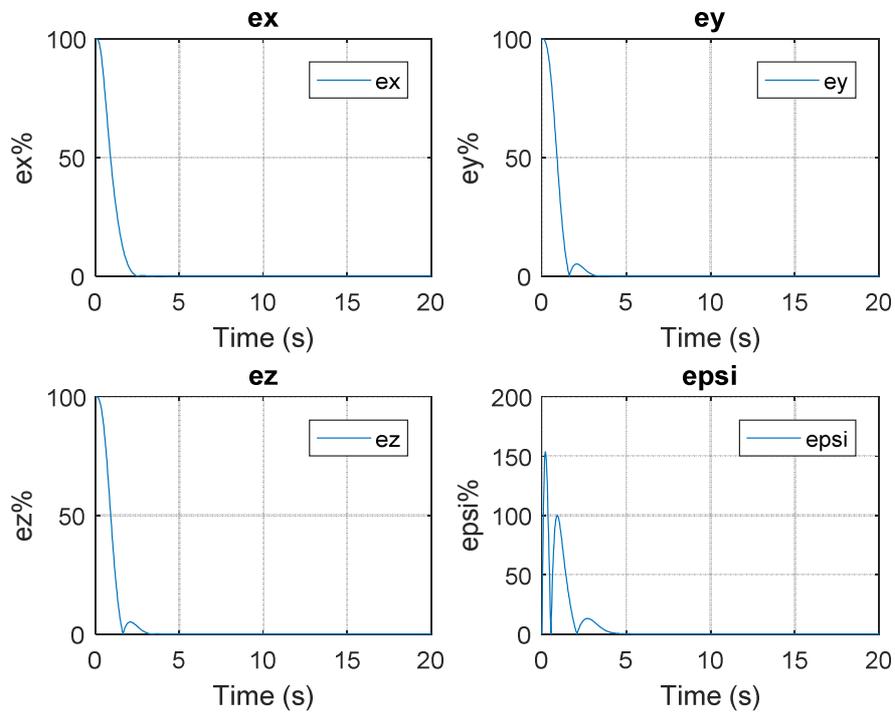


Figure 5. 14: Les erreurs de poursuite.

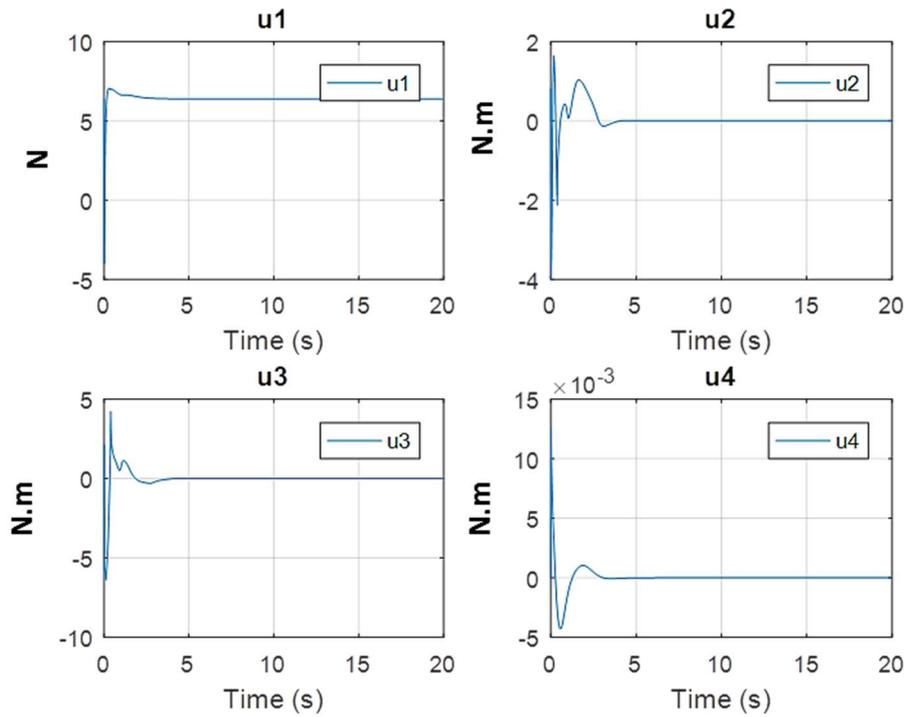


Figure 5. 15: Les commandes appliquées.

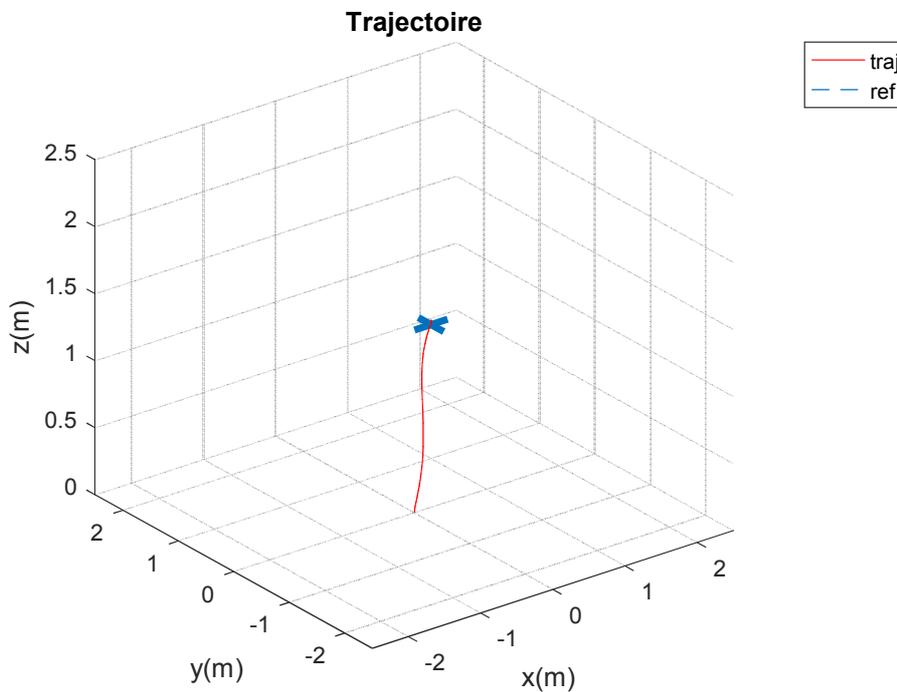


Figure 5. 16: La trajectoire en 3D.

Le quadri-rotor suit la trajectoire désirée, nous remarquons que les erreurs de réglage convergent vers l'origine en régime statique, notre système suit aussi le prédicteur d'état, donc les incertitudes sont bien estimées par l'AAP.

### 5.3.2 Simulation sous l'effet du vent

Nous allons utiliser le même environnement exposé aux rafales du vent de la partie précédente, pour pouvoir comparer entre les deux approches.

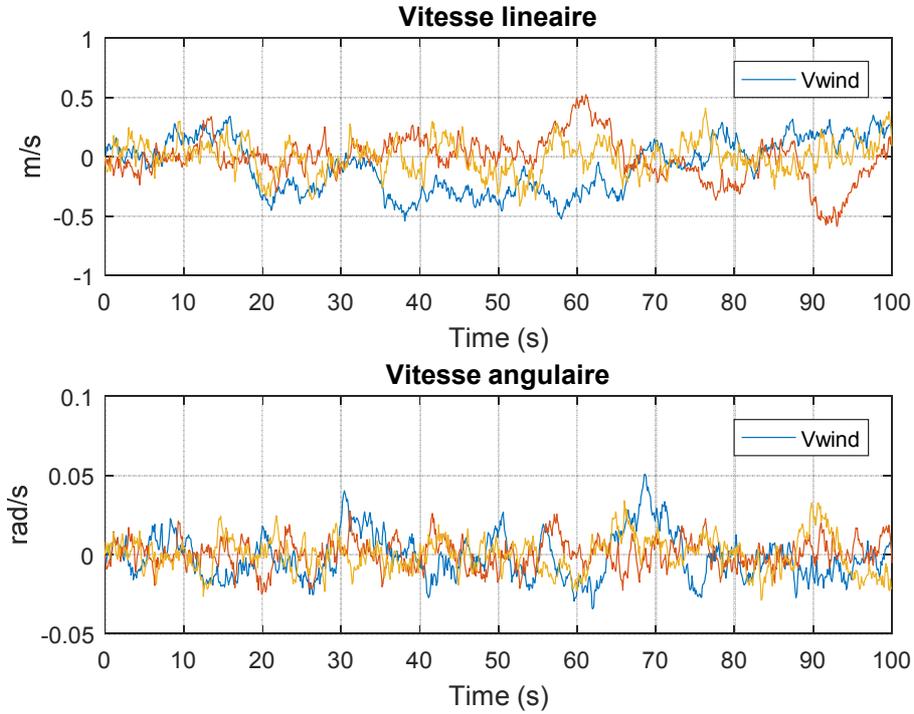


Figure 5. 17: Vitesse du vent.

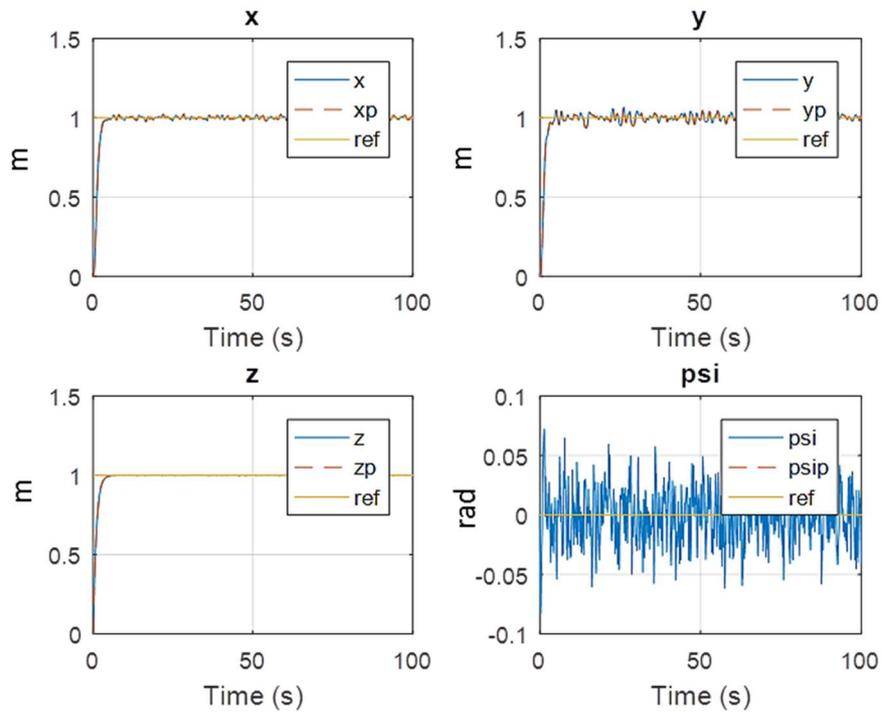


Figure 5. 18: Les réponses du système et du prédicteur.

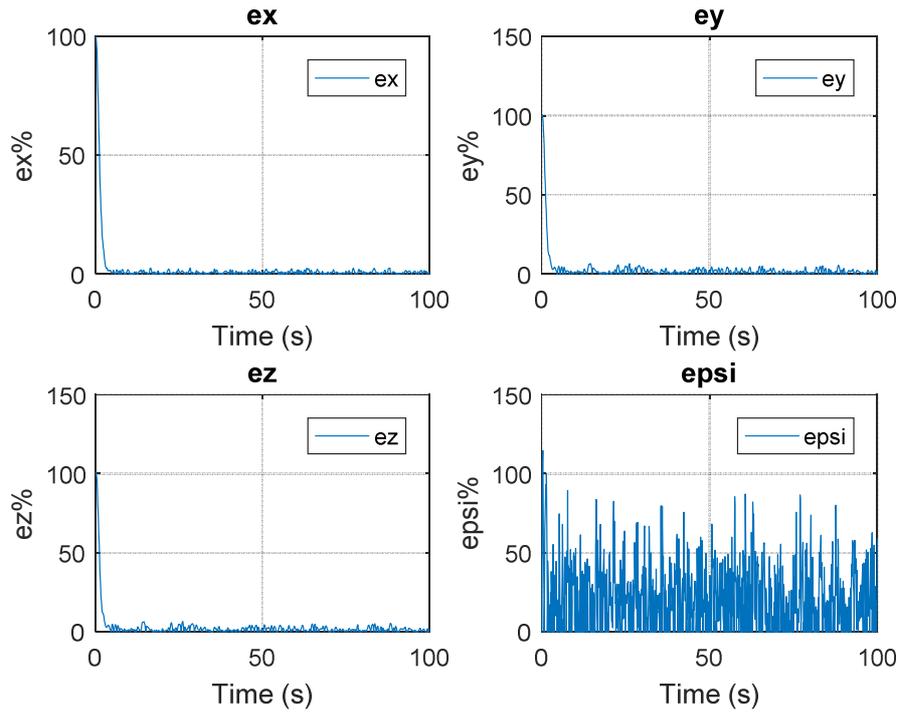


Figure 5. 19: Les erreurs de poursuite.

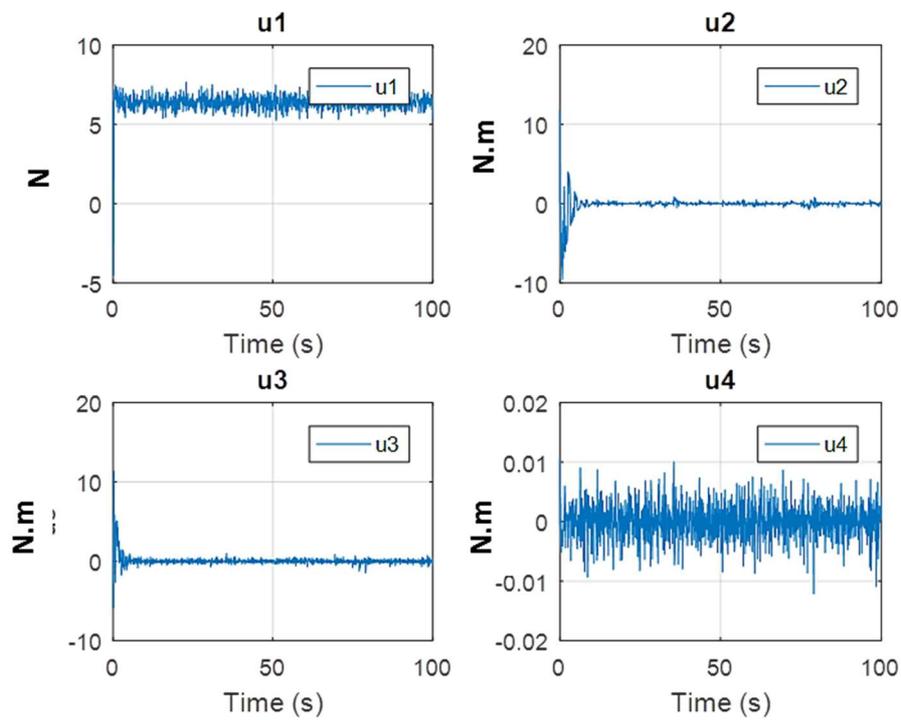


Figure 5. 20: Les commandes appliquées.

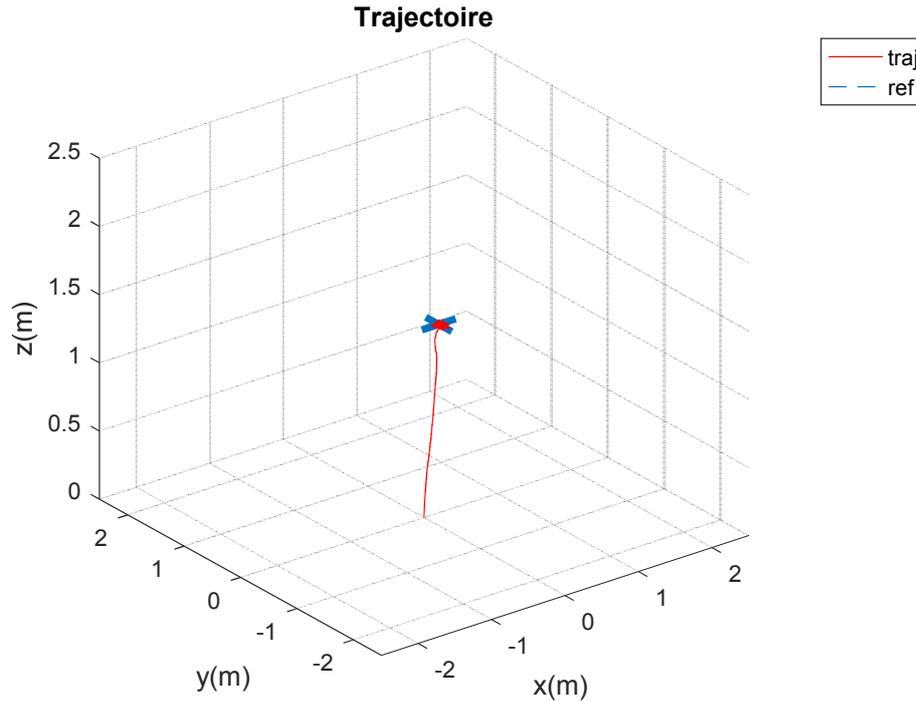


Figure 5. 21: La trajectoire en 3D.

La commande par l'approche des modes glissants a pu compenser les perturbations créées par le vent et maintenir la stabilité et le réglage de position avec de faibles erreurs en régime statique.

#### 5.4 Résultats des simulations de la commande $\mathcal{L}_1$ adaptative tolérante aux défauts actionneurs

Pour valider la loi de commande calculée, nous considérons  $\delta(t)$  comme suit :

$$\delta(t) = \begin{cases} 0, & t < 10s \\ \delta, & t \geq 10s \end{cases}$$

Nous allons utiliser une référence constante et voir si la commande garde les performances et la robustesse en présence d'un défaut d'actionneur. Nous présentons les sorties du système et la commande  $u_1$ .

### 5.4.1 Simulations en l'absence des perturbations atmosphériques

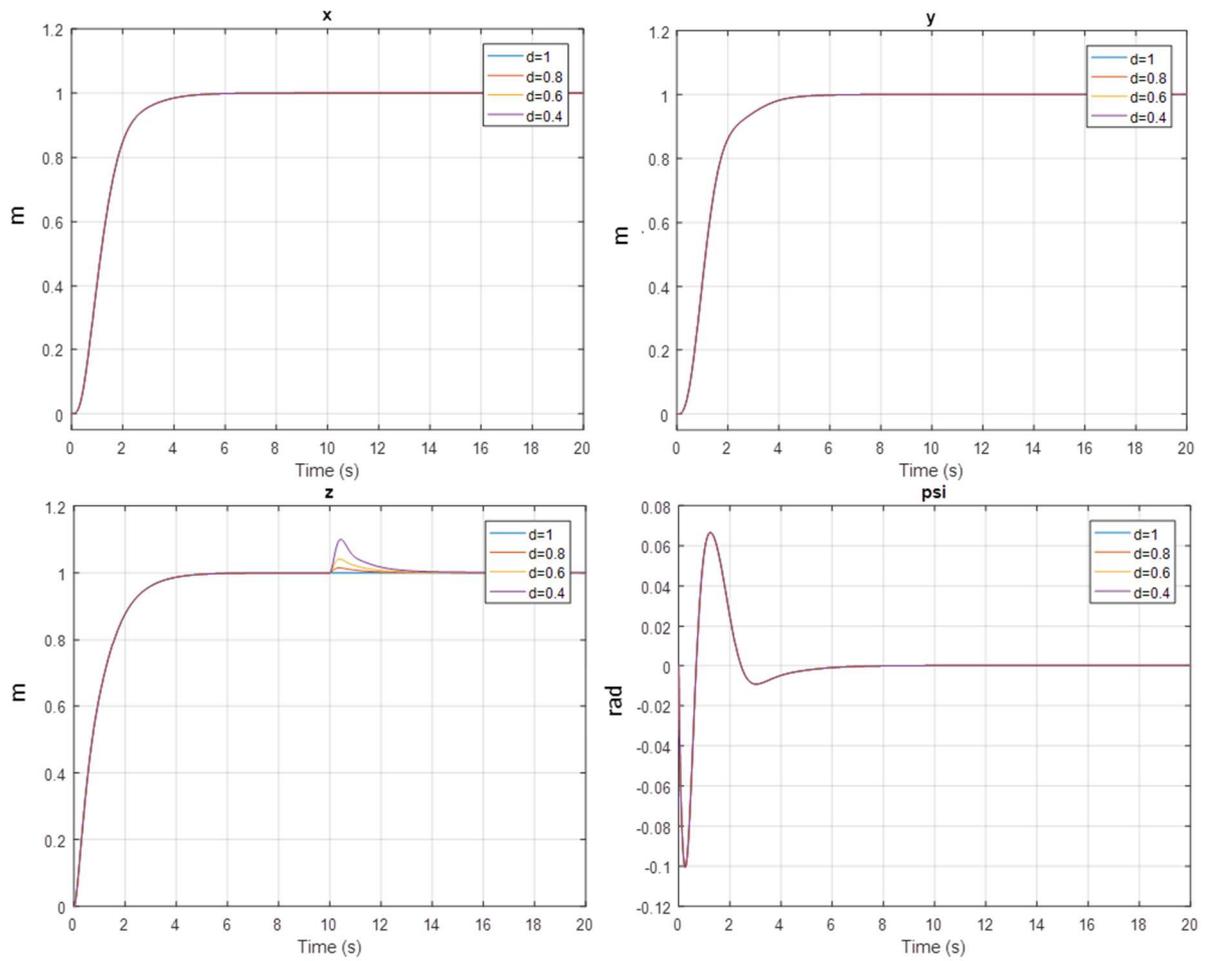


Figure 5. 22: Les sorties du système.

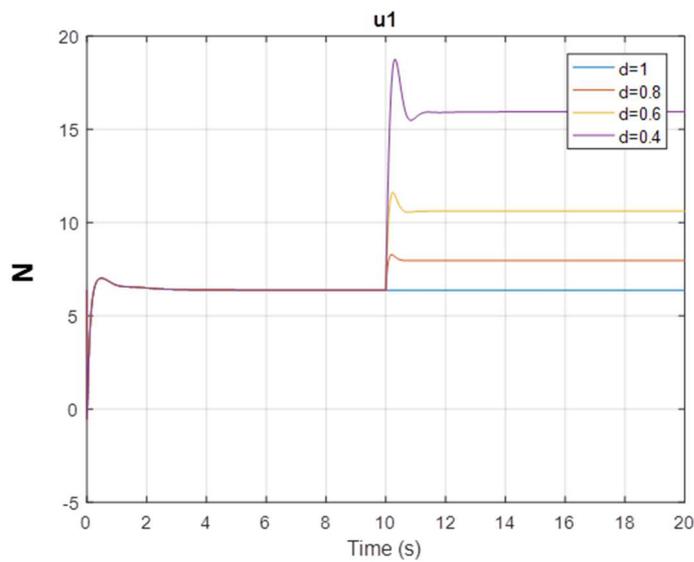


Figure 5. 23: La commande  $u_1$ .

### 5.4.2 Simulations sous l'effet du vent

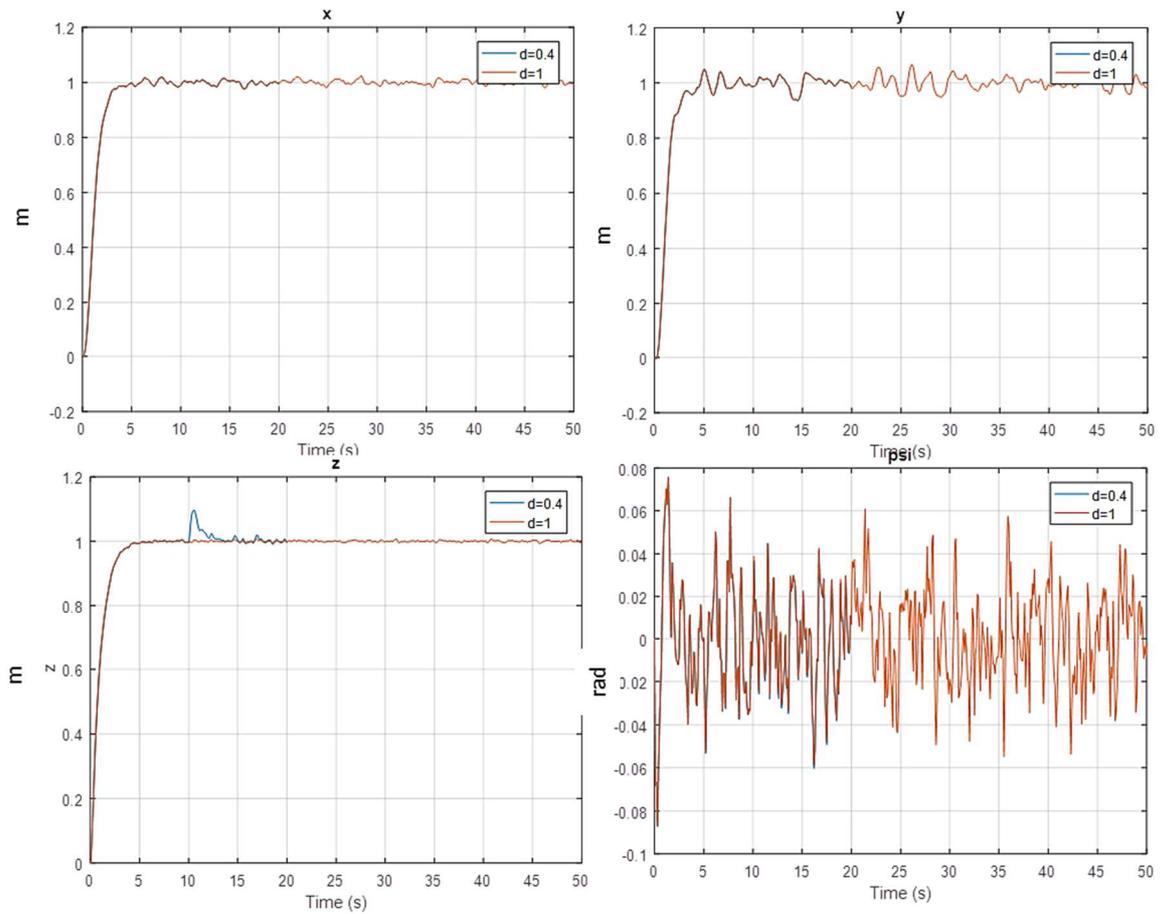


Figure 5. 24: Les sorties du système sous l'effet du vent.

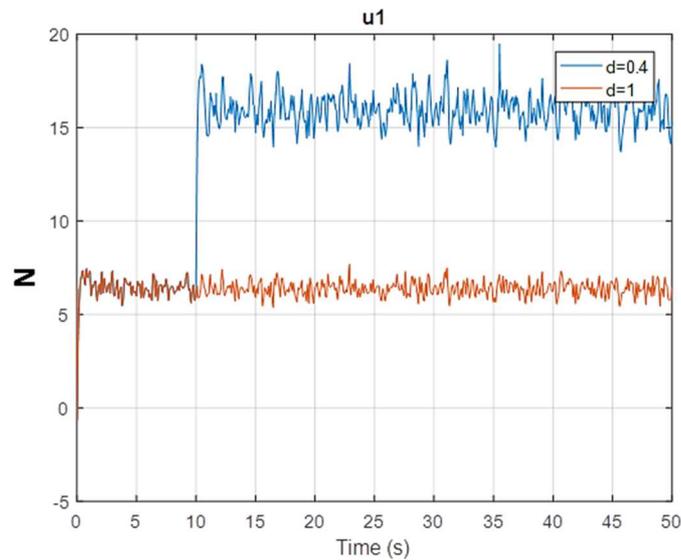


Figure 5. 25: La commande  $u_1$

Dans le cas d'une perte d'efficacité des actionneurs, la commande  $\mathcal{L}_1$  adaptative a pu compenser le défaut, maintenir la stabilité du système, offrir des bonnes performances et assurer le bon fonctionnement du système.

A l'instant 10s, la commande a augmenté au triple de la valeur nominale. En pratique il faut vérifier le seuil de défaut supporté par le système.

A travers ces résultats obtenus, on peut conclure que la commande  $\mathcal{L}_1$  adaptative est robuste vis-à-vis des défauts d'actionneurs.

### 5.5 Simulations de la commande $\mathcal{L}_1$ adaptative tolérante aux défauts actionneurs et capteurs

Soit : 
$$\delta(t) = \begin{cases} 0, & t < 10s \\ 0.6, & t \geq 10s \end{cases}; f(t) = \begin{cases} 0, & t < 10s \\ f, & t \geq 15s \end{cases}$$

Nous ajoutons un défaut constant aux capteurs défaillants et nous allons visualiser la sortie réelle du système, les sorties mesurées par les capteurs, l'estimation du défaut et les commandes appliquées.

#### 5.5.1 Simulations en l'absence des perturbations atmosphériques

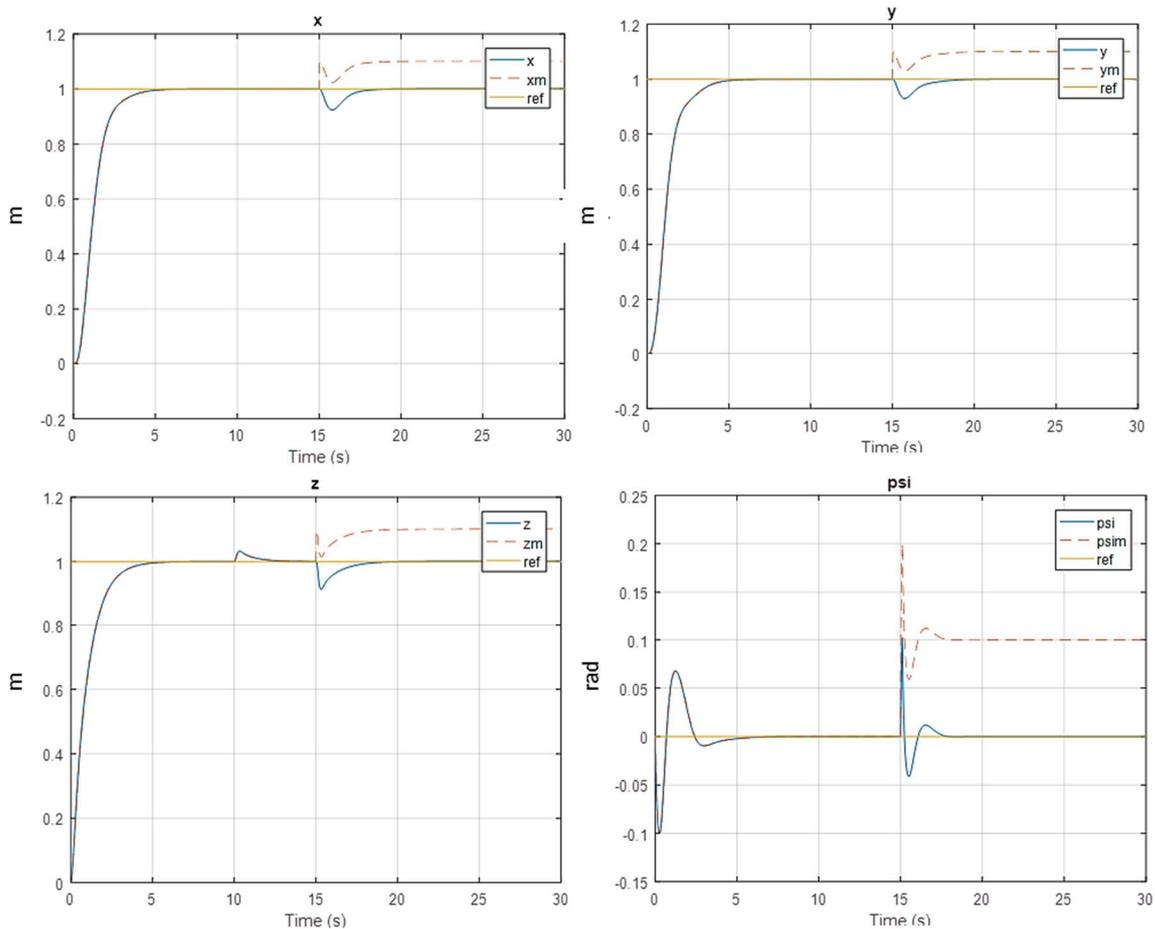


Figure 5. 26: Les sorties réelles et mesurées

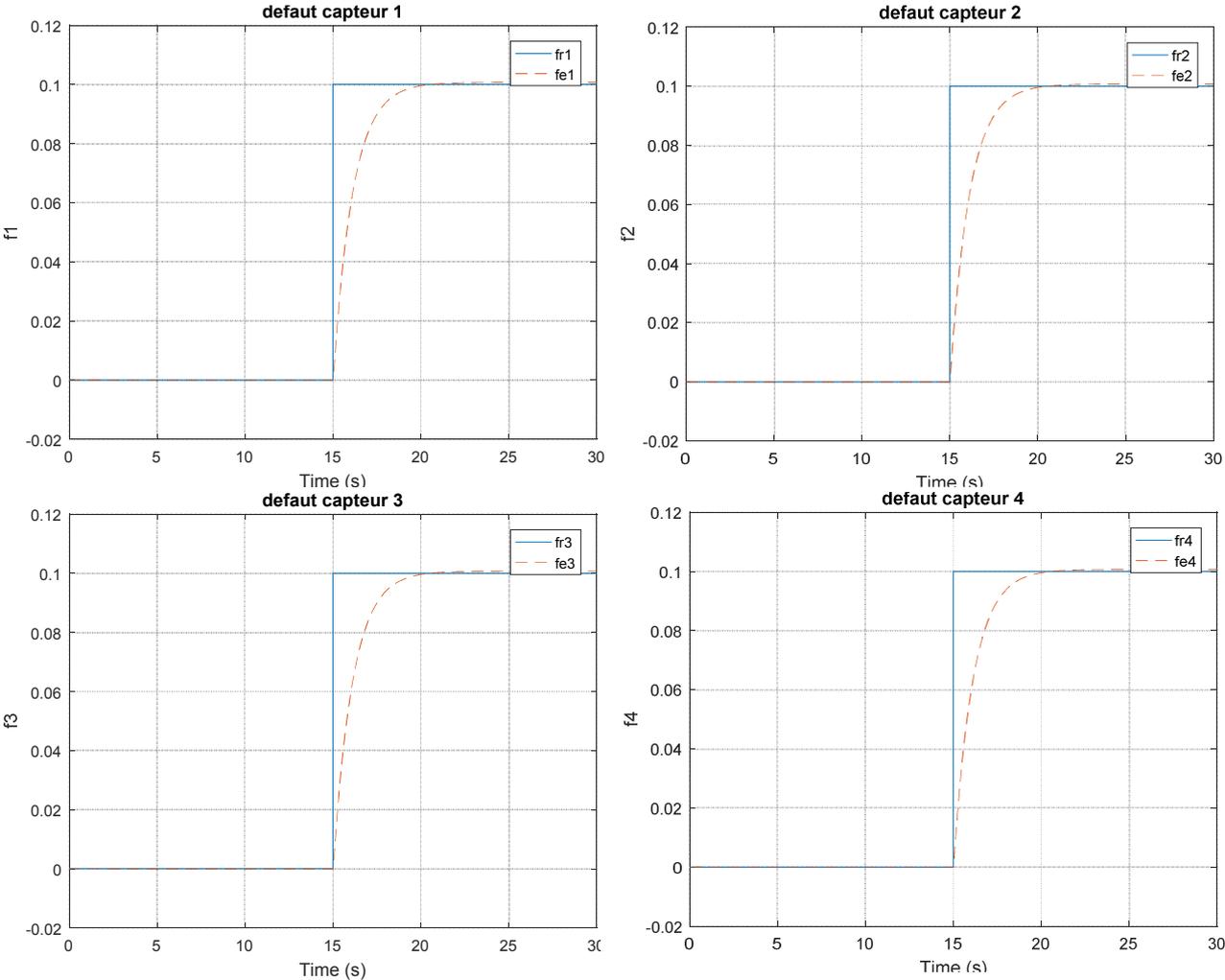


Figure 5. 27: L'estimation du défaut des capteurs

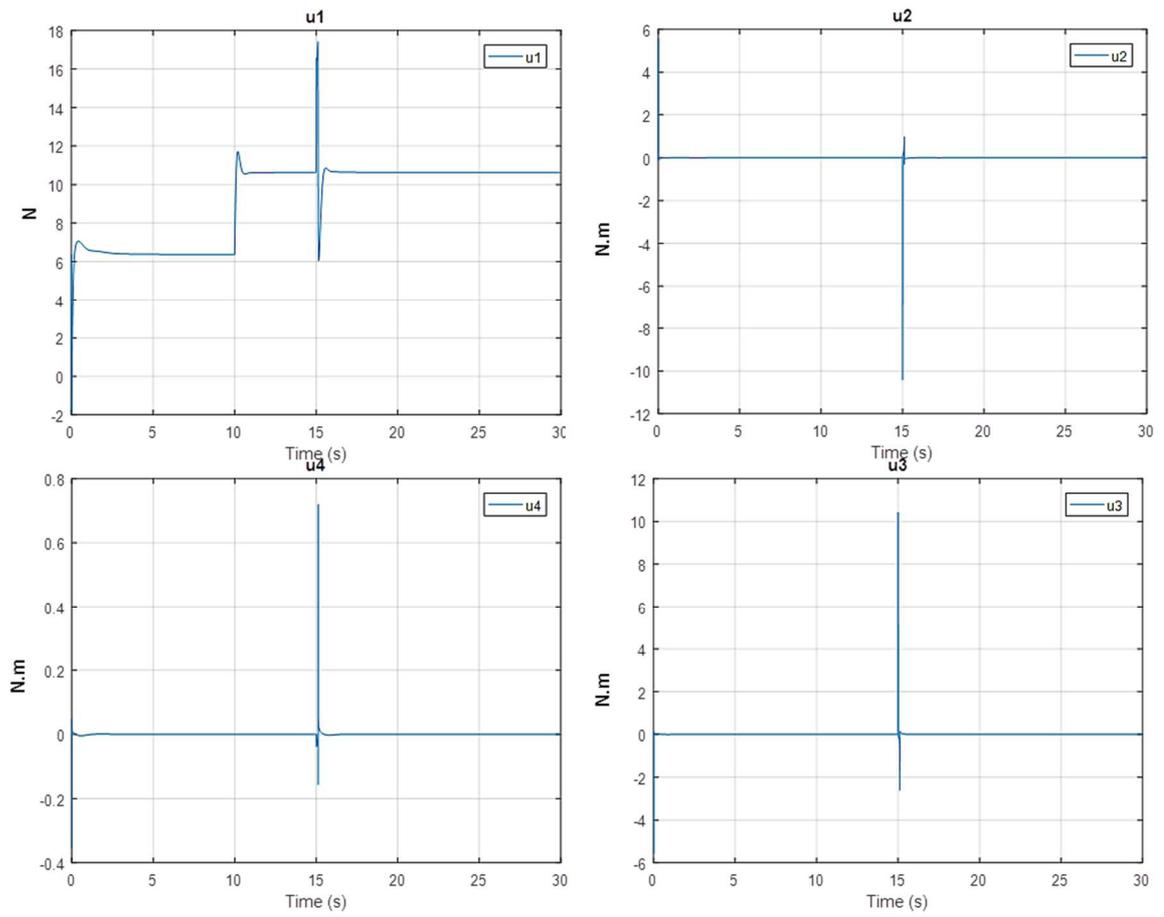


Figure 5. 28: Les commandes appliquées

### 5.5.2 Simulation sous l'effet du vent

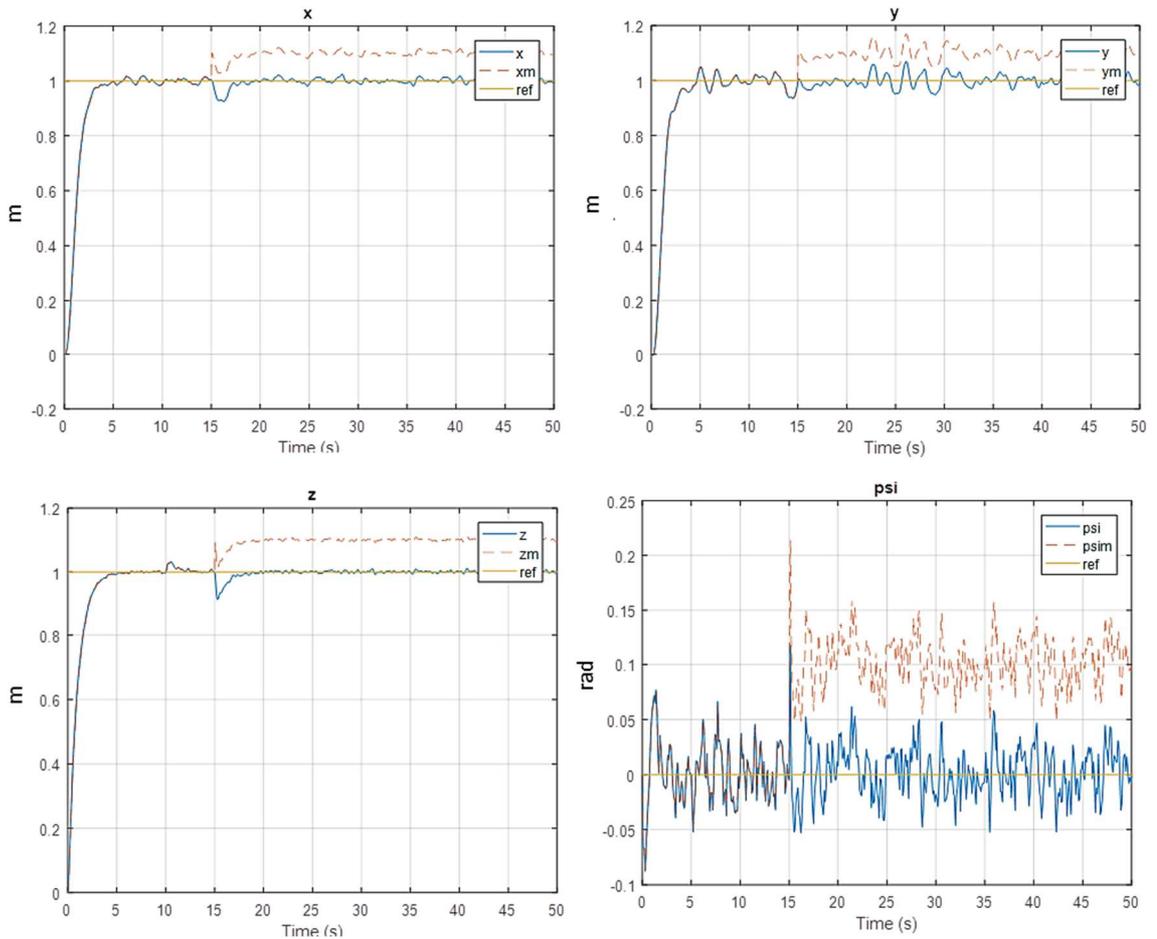
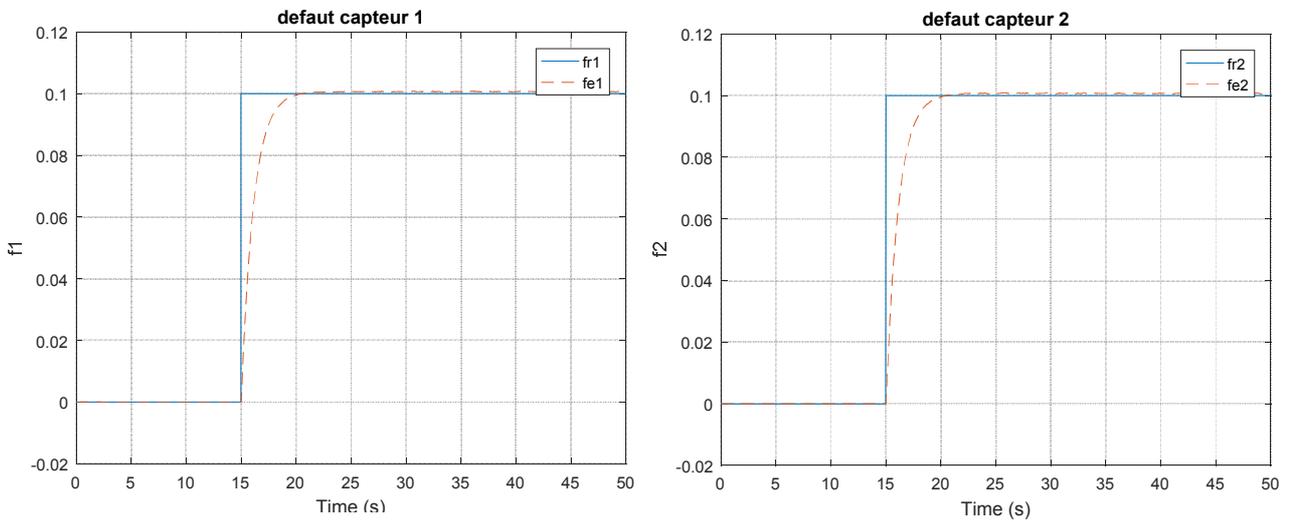


Figure 5. 29: Les sorties réelles et mesurées



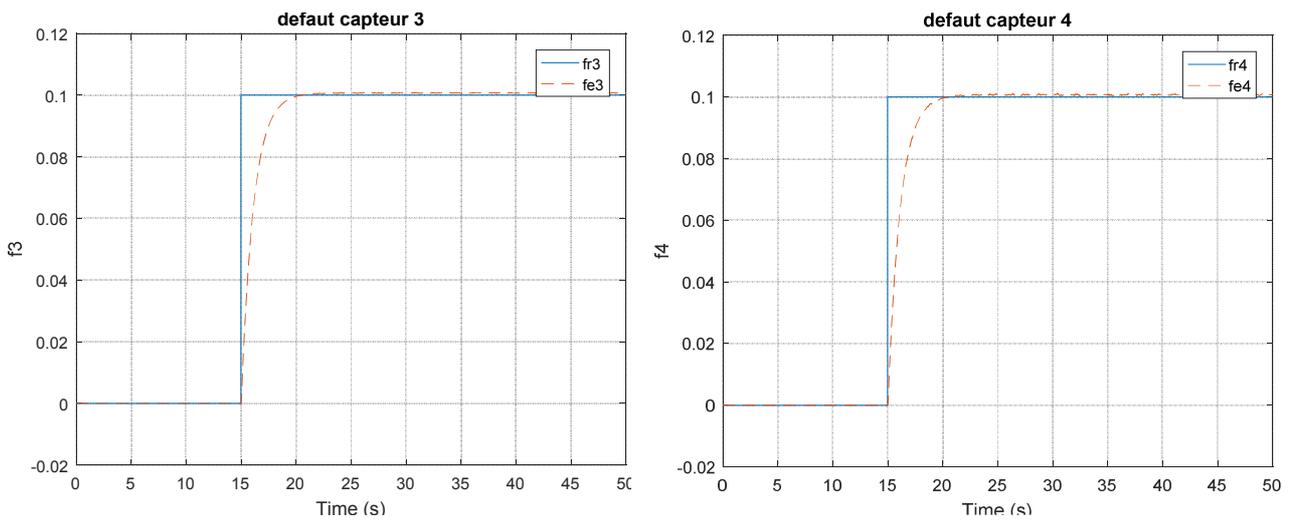


Figure 5. 30: L'estimation du défaut des capteurs

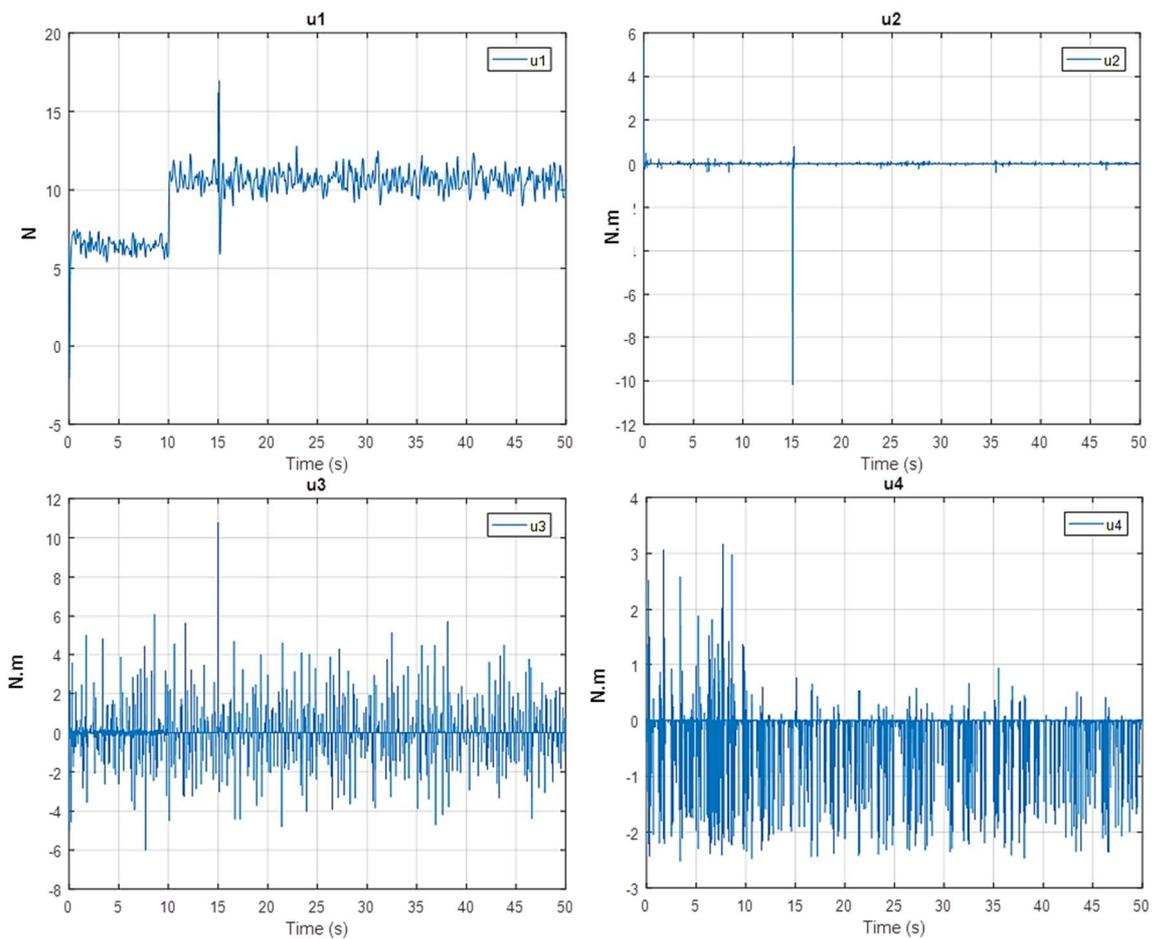


Figure 5. 31: Les commandes appliquées

Nous remarquons que le défaut du système est compensé malgré la défaillance du capteur et le système maintient sa robustesse et ses performances. Le sous-système basé sur la commande  $\mathcal{L}_1$  adaptative a pu estimer le défaut du capteur et l'ajouter au retour de sortie.

A l'instant 15s, les commandes ont varié à cause du défaut, le régulateur considère ce défaut comme une perturbation en sortie et dès que l'estimation du défaut converge vers sa valeur réelle, les commandes reviennent à leurs vraies valeurs.

Un mauvais choix des paramètres de commande peut conduire le système vers l'instabilité ou la perte de performance.

## 5.6 Simulations sous l'environnement Gazebo

On démarre Gazebo ainsi que Matlab/Simulink Puis on les connecte comme décrit précédemment. Afin de synchroniser Simulink et Gazebo, après plusieurs essais par tâtonnement, nous avons constaté qu'un pas de 0.003 s sur Simulink correspond à un pas d'environ 0,00018 s sur Gazebo. Le bloc qui devrait assurer cette tâche sur Simulink (Real Time Synchronization bloc) n'est actuellement pas opérationnelle selon MathWorks.

Le Hector Quadrotor est initialement à la position  $x=0,0048$  m,  $y=0,0046$ m,  $z=0,27$  m,  $\Psi=-0,0045$  rad,

Pour la simulation, on donne comme référence un échelon unitaire pour les trois coordonnées  $x$ ,  $y$ ,  $z$  et on laisse l'angle  $\Psi$  (angle de rotation autour de l'axe  $Z$ ) à zéro par exemple. On démarre la simulation sur Simulink. Le quadri-rotor effectuera le déplacement suivant:

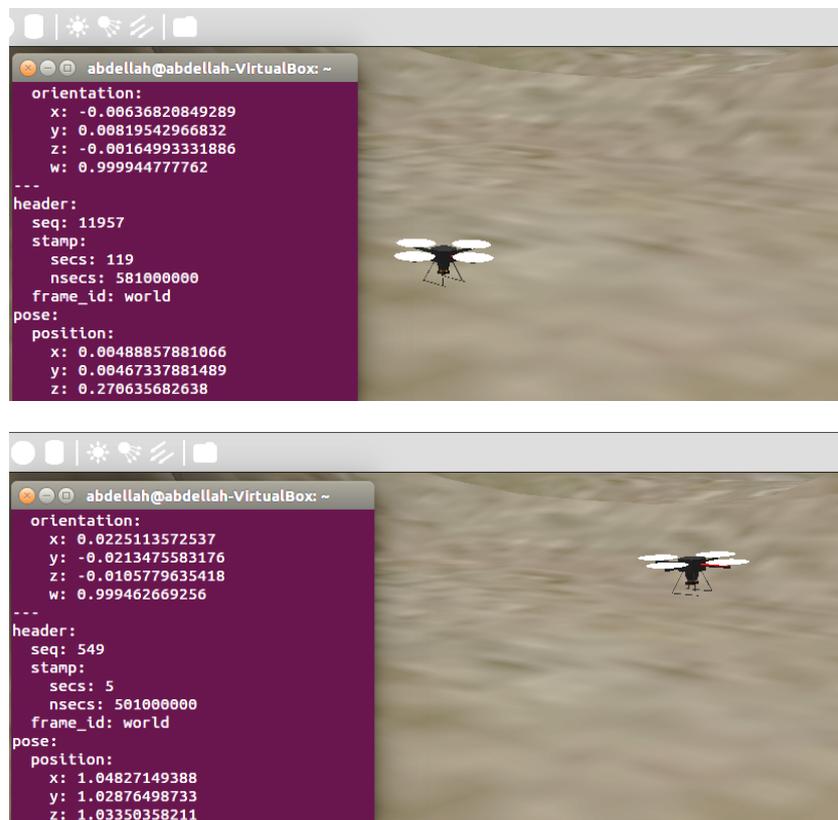


Figure 5. 32 : Positions de départ et d'arrivée du quadri-rotor .

Et à partir de Simulink, on reçoit les signaux suivants :

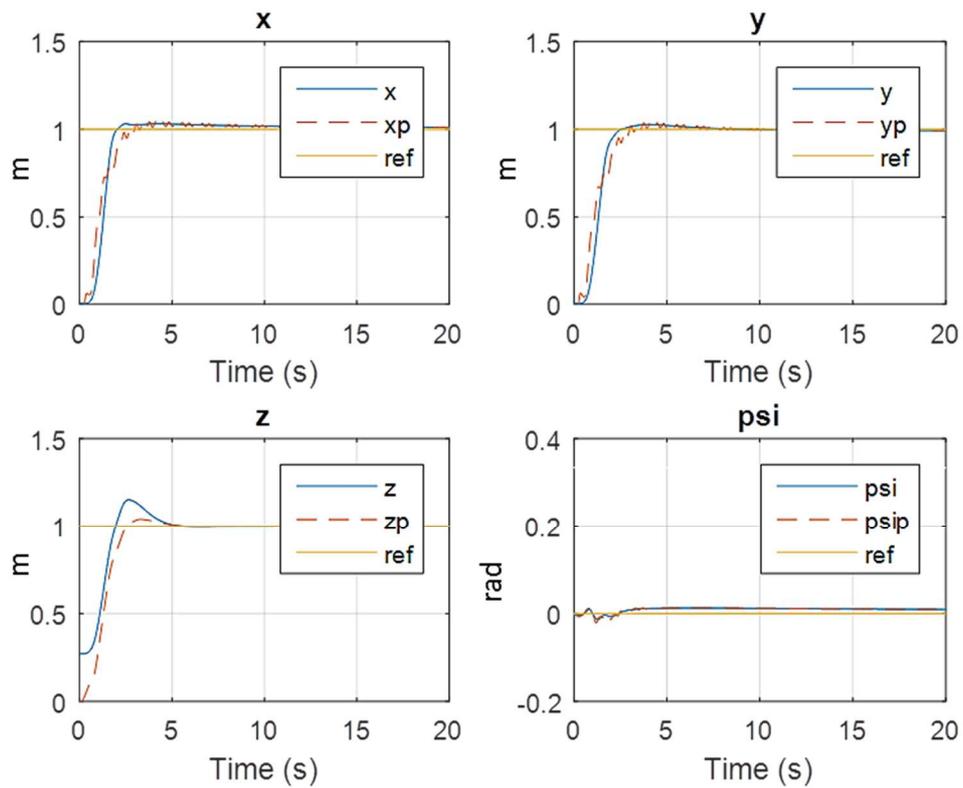


Figure 5.33 : Réponses de l'Hector Quadrotor et du prédicteur

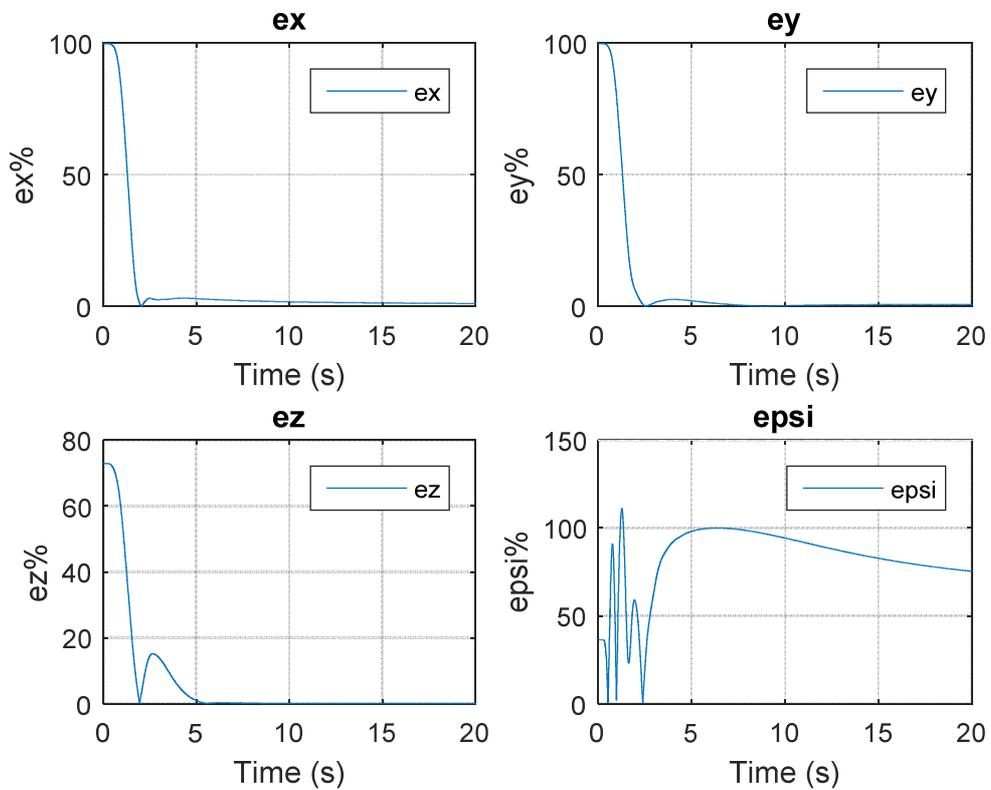


Figure 5.34 : Les erreurs de poursuite.

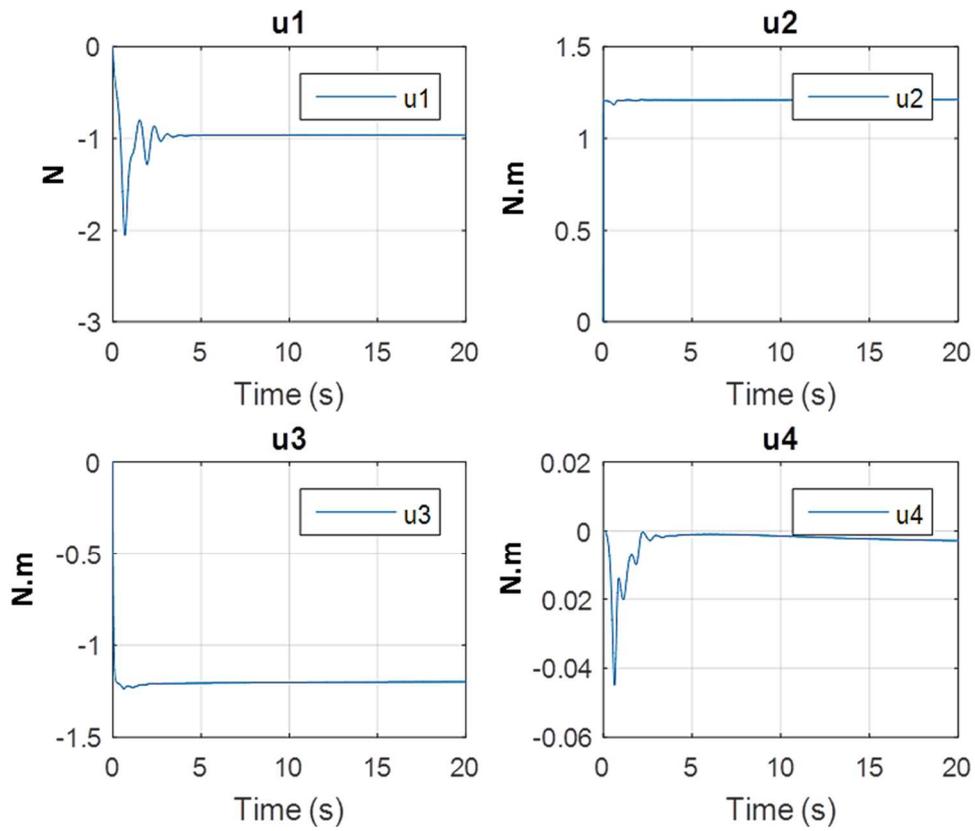


Figure 5. 35 : Les signaux de commande.

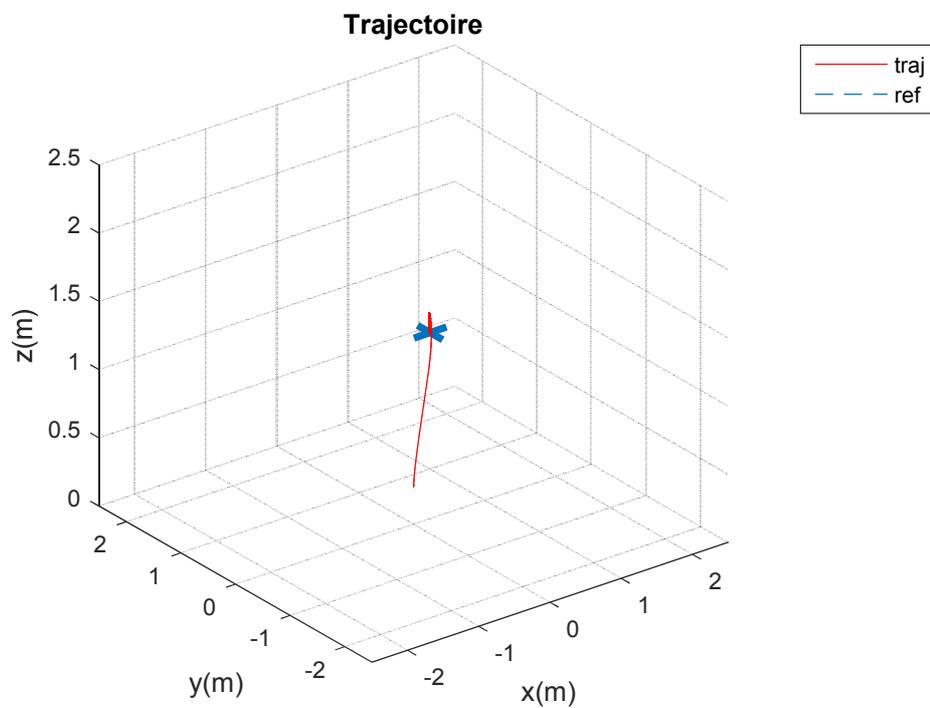


Figure 5. 36 : La trajectoire 3D du quadri-rotor

A partir des figures précédentes, on voit bien que le quadri-rotor est stable et suit la trajectoire désirée, de plus, le système suit le modèle de référence, de ce fait ; on peut conclure que la commande  $\mathcal{L}_1$  adaptative est robuste, d'autant plus qu'elle ait pu stabiliser le quadri-rotor du simulateur Gazebo dont on ne connaît pas le modèle. Cependant les performances ne sont pas parfaites, en effet ; les coordonnées  $x$ ,  $y$  et  $\Psi$  sont entachées d'une erreur statique négligeable pour les deux premières mais un peu plus importante pour  $\Psi$ , aussi, on remarque un dépassement d'environ 15% au régime transitoire pour la coordonnée  $z$ . On voit bien aussi que le quadri-rotor n'atteint le régime permanent qu'à partir de la cinquième seconde de la simulation.

Ces imperfections sont principalement dues à l'ensemble des retards ; en effet, la transmission de signaux entre Simulink (machine Windows) et ROS (machine virtuelle Linux) est sujette à des retards de communication. D'autre part, la commande de l'Hector Quadrotor se fait impérativement par la publication des vitesses, ceci nous amène à rajouter un bloc permettant le passage des signaux de commandes  $[T, L, M, N]$  vers les vitesses linéaires et angulaires, ce qui retarde encore plus la commande. Et nous savons bien que les retards sur la commande peuvent avoir une forte influence sur les performances et même la stabilité du système.

La figure 5.37 illustre mieux le problème de retards, on voit bien l'instant auquel les vitesses sont générées par le modèle de commande Simulink, et l'instant auquel le quadri-rotor (Gazebo) répond à la commande.

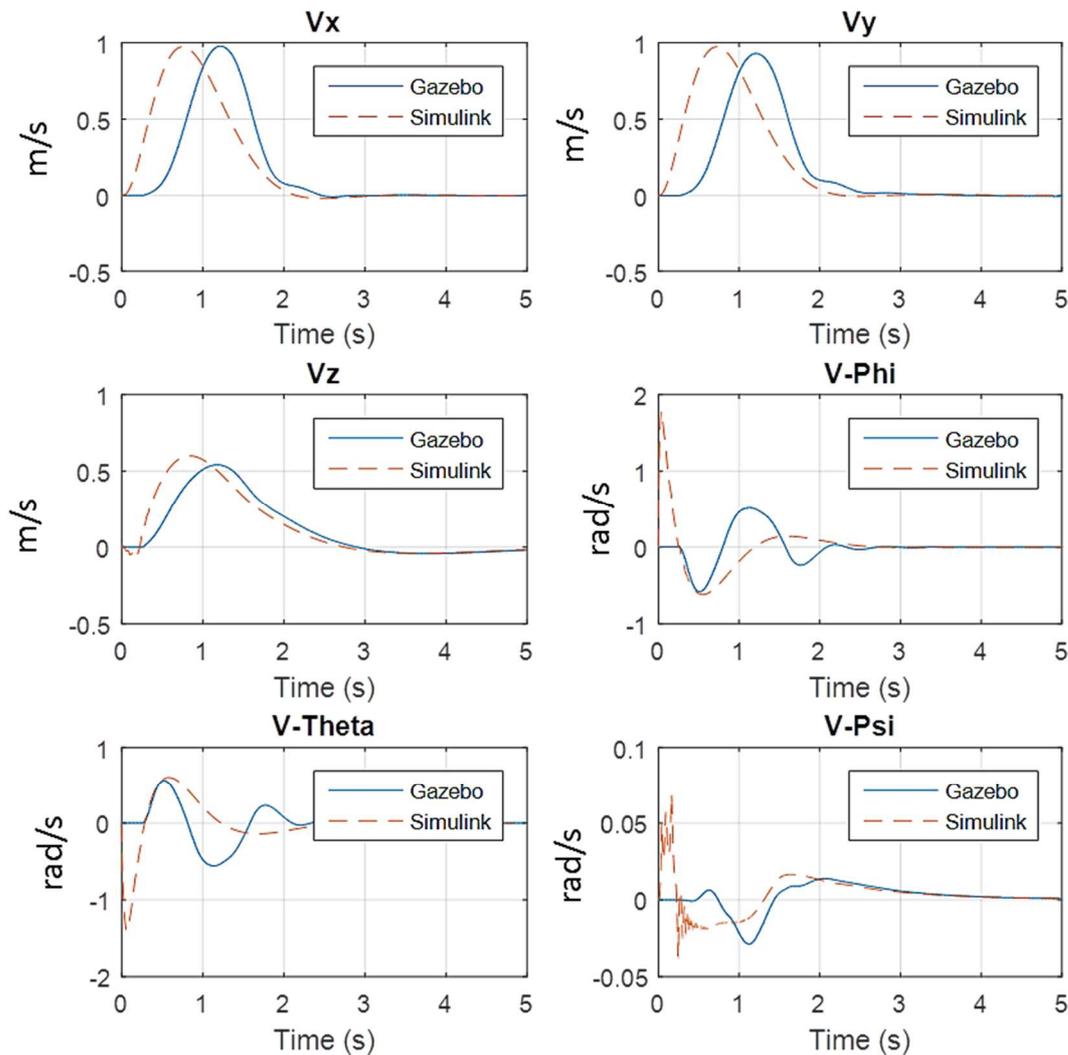


Figure 5. 37 : Les retards de simulation.

## 5.7 Conclusion

La configuration de la commande  $\mathcal{L}_1$  adaptative par ses deux approches a pu résoudre les problèmes de perturbations du vent qui ont une grande influence sur le quadri-rotor . De plus, l'approche par modes glissants rend la commande plus robuste en estimant les bornes des perturbations dépassant ainsi les limites de la commande adaptative  $\mathcal{L}_1$  classique. D'après ces résultats, le quadri-rotor peut fonctionner dans divers environnements et résister aux effets du vent. De plus, les défauts actionneurs et capteurs deviennent tolérables grâce à la commande FTC  $\mathcal{L}_1$  qui a permis de les estimer et les compenser. La commande  $\mathcal{L}_1$  par modes glissants tolérante aux défauts est simple à synthétiser et très robuste, elle a amélioré considérablement le système AFTC. Et enfin, les simulations sous Gazebo viennent confirmer encore une fois la robustesse et la performance de la commande  $\mathcal{L}_1$  adaptative testée sur l'Hector Quadrotor

dont on ne connaît pas le modèle, mais qui a pu stabiliser le quadri-rotor et assurer la poursuite.

# **Conclusion générale**

Ce travail a été dédié à l'étude et l'application d'une commande adaptative tolérante aux défauts appliquée sur le quadri-rotor, notamment, la commande  $\mathcal{L}_1$  adaptative. Cette méthode permet à la fois de faire face aux incertitudes et aux perturbations qui peuvent affecter le système, et au même temps, d'assurer une adaptation rapide sans perdre en robustesse, remédiant ainsi au problème de compromis adaptation/robustesse.

Deux approches ont été présentées dans ce travail ; la première approche s'est basée sur la descente du gradient, les résultats des simulations nous ont permis de déduire les faiblesses de cette approche fondée sur l'hypothèse que les bornes des perturbations sont connues, mais ce n'est pas toujours le cas, d'ailleurs, les variations de ces bornes affecte directement les performances du système, une approche alternative était donc nécessaire afin de remédier à ce problème.

Notre proposition était d'appliquer la commande  $\mathcal{L}_1$  adaptative mais en se basant sur les modes glissants, tout en estimant les bornes des perturbations, ce choix justifié vu que l'approche par modes glissants offre une robustesse à la commande, elle permet d'estimer les perturbations et leurs bornes, ce qui permet de repousser les limites de la commande adaptative  $\mathcal{L}_1$  classique. Les résultats des simulations confirment la pertinence de cette approche pour la résolution de ce problème, en effet ; le quadri-rotor peut fonctionner dans divers environnements et résister aux effets du vent.

L'aspect tolérant aux défauts de notre commande lui rajoute un autre niveau de robustesse. En estimant les défaut capteur, actionneurs et les erreurs de modélisation qui peuvent atteindre le système, la commande tolérante aux défauts a pu compenser leurs effets et maintenir la robustesse et les performances du quadri-rotor même en présence de ces défaillances, cette approche permet donc de rendre le quadri-rotor un engin plus sûr répondant ainsi aux besoins de sécurité élevés vu l'usage vaste de ces appareils en zones urbaines.

## Bibliographie

- [1] PHUNG, Duc Kien. *Conception, modélisation et commande d'un mini-drone convertible* [en ligne]. Thèse de doctorat : Robotique. Paris : Université Pierre et Marie Curie, 2015, 173p. [Consultée le 17 mars 2018]. Disponible à l'adresse : <https://tel.archives-ouvertes.fr/tel-01261345>
- [2] *All the world's Rotorcraft*, [www.aviastar.org](http://www.aviastar.org).
- [3] Philippe Giguère, *Introduction à la robotique mobile: les capteurs*. Université LAVAL. Disponible à l'adresse : <http://www2.ift.ulaval.ca/~pgiguere/cours/IntroRobotique/notes/02-Capteurs.pdf>
- [4] BOUABDALLAH, Samir. NOTH, Andre. SIEGWART, Roland. *PID vs LQ Control Techniques Applied to an Indoor Micro Quadri-rotor* . Autonomous Systems Laboratory Swiss Federal Institute of Technology.
- [5] Krossblade Aerospace. HISTORY OF QUADCOPTERS AND OTHER MULTIROTORS [en ligne]. Disponible sur: <http://www.krossblade.com/history-of-quadcopters-and-multirotors/>
- [6] AMEHO, Yann. *Du pilotage d'une famille de drones à celui d'un drone hybride via la commande adaptative*. [en ligne]. Thèse de doctorat : Automatique. Université de Toulouse: Institut Supérieur de l'Aéronautique et de l'Espace, 2013, 144 p. Disponible à l'adresse: <https://tel.archives-ouvertes.fr/tel-01020984/document>
- [7] ZEMALACHE MEGUENNI, Kadda. *Commande d'un système sous-actionné : Application à un drone à Quatre Hélices*. [en ligne]. Thèse de doctorat : Robotique. Université d'Evry Val d'Essonne, 2006, 119 p.
- [8] BOUABDALLAH, Samir. SIEGWART, Roland. *Backstepping and Sliding-mode Techniques Applied to an Indoor Micro Quadri-rotor* . Autonomous Systems Lab Ecole Polytechnique Fédérale de Lausanne (EPFL).
- [9] C. Nicol, C.J.B. Macnab, A. Ramirez-Serrano. *ROBUST NEURAL NETWORK CONTROL OF A QUADRI-ROTOR HELICOPTER*. University of Calgary.
- [10] Patrick Bouffard, Anil Aswani, Claire Tomlin. *Learning-Based Model Predictive Control on a Quadri-rotor : Onboard Implementation and Experimental Results*.
- [11] Matthias Schreier. *Modeling and Adaptive Control of a Quadri-rotor* . Institute of Automatic Control and Mechatronics .Control Theory and Robotics Lab, TU Darmstadt, Landgraf-Georg-Str. 4, 64283 Darmstadt, Germany.

- [12] Ahsene Boubakir, Toufik SouaneF, Salim Labiod, Franck Plestan, Fares Boudjema. *An LI fuzzy adaptive controller for a class of SISO nonaffine nonlinear systems: application to the control of an electropneumatic actuator*. J Systems and Control Engineering. 2016, Vol. 230(8) 736–748 IMechE 2016
- [13] BREZOESCU, Cornel-Alexandru. *Navigation d'un avion miniature de surveillance aérienne en présence de vent*. [en ligne]. Thèse de doctorat : Laboratoire HEUDIASYC, Université de Technologie de Compiègne, 2013, 165 p. [Consultée le 24 mars 2018]. Disponible à l'adresse : <https://tel.archives-ouvertes.fr/tel-01060415>
- [14] ZEMALACHE MEGUENNI, Kadda. *Commande d'un système sous-actionné : Application à un drone à Quatre Hélices*. [en ligne]. Thèse de doctorat : Robotique, Université d'Evry Val d'Essonne, 2006, 119 p. Disponible à l'adresse : <http://www.biblio.univ-evry.fr/theses/2006/Interne/2006EVRY0039.pdf>
- [15] A. Kadi. "Mécanique Rationnelle Cours et exercices résolus". (pages 222-225). Université M'hamed Bougara, Boumerdès.
- [16] S. Bouabdallah. "*Design and Control of Quadri-rotor s with Application to Autonomous Flying*". Thèse de Doctorat. Ecole Polytechnique Fédérale de Lausanne, Suisse, 2007.
- [17] Francesco Sabatino. "Quadri-rotor control: modeling, nonlinear control design, and simulation". Thèse de Master Stockholm, Suède juin 2015.
- [18] D. Mellinger, Q. Lindsey, M. Shomin, and V. Kumar. Design, modeling, estimation and control for aerial grasping and manipulation. In Intelligent Robots and Systems (IROS), 2011 IEEE/RSJ International Conference on, pages 2668–2673, sept. 2011.
- [19] J. Miguel Brito Domingues. "Quadri-rotor prototype". Thèse de Master. Université technique de Lisbonne, Portugal, 2009.
- [20] SOUANEF, Toufik, BOUBAKIR, Ahsene, et FICHTER, Walter.  $\mathcal{L}_1$  Adaptive Control of Systems with Disturbances of Unknown Bounds. In : Advances in Aerospace Guidance, Navigation and Control. Springer, Cham, 2015. p. 151-165.—
- [21] G. Antonelli, F. Arrichiello, S. Chiaverini, and P. R. Giordano, "Adaptive trajectory tracking for quadri-rotor MAVs in presence of parameter uncertainties and external disturbances," IEEE/ASME International Conference on Advanced Intelligent Mechatronics, pp. 1337–1342, July 2013.
- [22] G. V. Raffo, M. G. Ortega, and F. R. Rubio, "An integral predictive/ nonlinear control structure for a quadri-rotor helicopter," Automatica, vol. 46, pp. 29–39, Jan. 2010.
- [23] D. Mellinger and V. Kumar, "Minimum snap trajectory generation and control for quadri-rotor s," in IEEE International Conference on Robotics and Automation, pp. 2520–2525, Ieee, May 2011
- [24] P. D. Monte and B. Lohmann, "Position trajectory tracking of a quadri-rotor helicopter

- based on L1 adaptive control,” in European Control Conference (ECC), pp. 3346–3353, 2013.
- [25] S. Zhao, W. Dong, and J. Farrell, “Quaternion-based Trajectory Tracking Control of VTOL-UAVs using Command Filtered Backstepping,” in American Control Conference (ACC), pp. 1018–1023, 2013.
- [26] B. Zhu and W. Huo, “Trajectory Linearization Control for a quadri-rotor helicopter,” in IEEE International Conference on Control and Automation ICCA, pp. 34–39, Ieee, June 2010.
- [27] T. Madani and A. Benallegue, “Control of a Quadri-rotor Mini-Helicopter via Full State Backstepping Technique,” in Proceedings of the 45th IEEE Conference on Decision and Control, pp. 1515–1520, Ieee, 2006
- [28] Z. Zuo, “Trajectory tracking control design with command-filtered compensation for a quadri-rotor ,” IET Control Theory & Applications, vol. 4, pp. 2343–2355, Nov. 2010.
- [29] J. Escareño, S. Salazar, H. Romero, and R. Lozano, “Trajectory Control of a Quadri-rotor Subject to 2D Wind Disturbances,” Journal of Intelligent & Robotic Systems, vol. 70, pp. 51–63, Aug. 2012.
- [30] Ioannou PA and Sun J. Robust adaptive control. Englewood Cliffs, NJ: Prentice Hall, 1996.
- [31] Jean-Baptiste POMET ‘‘ Sur la commande adaptative des systèmes non linéaire’’ thèse de doctorat, Ecole des mines de paris, septembre 1989.
- [32] Vance J. VanDoren, Introduction, Editor(s): Vance J. VanDoren, Techniques for Adaptive Control, Butterworth-Heinemann,2003, Pages 1-21, ISBN 9780750674959, <https://doi.org/10.1016/B978-075067495-9/50002-1>. (<http://www.sciencedirect.com/science/article/pii/B9780750674959500021>)
- [33] A.Boubakir et al. ‘‘ An L1 fuzzy adaptive controller for a class of SISO nonaffine nonlinear systems: application to the control of an electropneumatic actuator’’J Systems and Control Engineering 2016, Vol. 230(8) 736–748, DOI:10.1177/0959651816652023.
- [34] Cao, C. and Hovakimyan, N. (2008a), “L1 adaptive controller for multi-input multi-output systems in the presence of unmatched disturbances”, Proceedings of the 2008 American Control Conference (ACC), Seattle, WA, June, pp. 4105-4110. doi: 10.1109/ACC.2008.4587136
- [35] Cao, C. and Hovakimyan, N. (2008b), “L1 adaptive controller for systems with unknown time-varying parameters and disturbances in the presence of non-zero trajectory initialization error”, International Journal of Control, Vol. 81 No. 7, pp. 1148-1162. doi: 10.1080/00207170701670939
- [36] HOVAKIMYAN, Naira et CAO, Chengyu. L1 adaptive control theory: guaranteed robustness with fast adaptation (page145). SIAM-Society for Industrial and Applied Mathematics, 2010.

- [37] <https://fr.mathworks.com/help/aeroblks/drydenwindturbulencemodelcontinuous.html>
- [38] EDWARDS, Christopher et TAN, Chee Pin. A comparison of sliding mode and unknown input observers for fault reconstruction. *European Journal of control*, 2006, vol. 12, no 3, p. 245-260.
- [39] Hicham Khebach, "Contribution à la commande tolérante aux défauts : théorie et application". Thèse de doctorat : Automatique, Ecole Nationale Polytechnique, Alger, 2017.
- [40] GERTLER, Janos. *Fault Detection and Diagnosis*. Springer London, 2015.
- [41] BOŠKOVIĆ, Jovan D. et MEHRA, Raman K. Failure detection, identification and reconfiguration in flight control. In: *Fault Diagnosis and Fault Tolerance for Mechatronic Systems: Recent Advances*. Springer, Berlin, Heidelberg, 2003. p. 129-167.
- [42] BOSKVIC, J. D. et MEHRA, Raman K. Stable adaptive multiple model-based control design for accommodation of sensor failures. In : *American Control Conference, 2002. Proceedings of the 2002*. IEEE, 2002. p. 2046-2051.
- [43] DEMETRIOU, Michael A. et POLYCARPOU, Marios M. Incipient fault diagnosis of dynamical systems using online approximators. *IEEE Transactions on Automatic Control*, 1998, vol. 43, no 11, p. 1612-1617.
- [44] ZHANG, Xiaodong, POLYCARPOU, Marios M., et PARISINI, Thomas. A robust detection and isolation scheme for abrupt and incipient faults in nonlinear systems. *IEEE transactions on automatic control*, 2002, vol. 47, no 4, p. 576-593.
- [45] PATTON, Ron J. Fault-tolerant control: the 1997 situation. *IFAC Proceedings Volumes*, 1997, vol. 30, no 18, p. 1029-1051.
- [46] BLANKE, Mogens, KINNAERT, Michel, LUNZE, Jan, et al. *Diagnosis and fault-tolerant control*. Berlin : springer, 2006.
- [47] Mueller, M.W. and D'Andrea, R. (2014), "Stability and control of a quadcopter despite the complete loss of one, two, or three propellers", *IEEE International Conference on Robotics and Automation (ICRA)*, Hong Kong, May, pp. 45-52. doi: 10.1109/ICRA.2014.6906588
- [48] Lanzon, A., Freddi, A. and Longhi, S. (2014), "Flight control of a quadri-rotor vehicle subsequent to a rotor failure", *Journal of Guidance Control and Dynamics*, Vol. 37 No. 2, pp. 580-591. doi: 10.2514/1.59869
- [49] Adîr, V.G., Stoica, A.M., Marks, A. and Whidborne, J.F. (2011), "Modelling, stabilization and single motor failure recovery of a 4Y octorotor", *Proceedings of the 13th IASTED International Conference on Intelligent Systems and Control (ISC 2011)*, Cambridge, July, pp. 82-87. doi: 10.2316/P.2011.744-070.
- [50] Marks, A., Whidborne, J.F. and Yamamoto, I. (2012), "Control allocation for fault tolerant control of a VTOL octorotor", *UKACC International Conference on Control 2012*, Cardiff, September, pp. 357-362. doi: 10.1109/CONTROL.2012.6334656.

- [51] Hamayun, M.T., Edwards, C., Alwi, H. and Bajodah, A. (2015), “A fault tolerant direct control allocation scheme with integral sliding modes”, *International Journal of Applied Mathematics and Computer Science*, Vol. 25 No. 1, pp. 93-102. doi: 10.1515/amcs-2015-0007.
- [52] Alwi, H. and Edwards, C. (2015), “Sliding mode fault-tolerant control of an octorotor using linear parameter varying-based schemes”, *IET Proceedings – Control Theory and Applications*, Vol. 90 No. 4, pp. 618-636. doi: 10.1049/iet.cta.2014.0215.
- [53] Robot Operating System ROS. [en ligne]. Disponible sur : <http://www.ros.org>
- [54] Willow Garage. [en ligne]. Disponible sur : <http://www.willowgarage.com/>
- [55] Gazebo. [en ligne]. Disponible sur : <http://gazebosim.org/>
- [56] MathWorks. [en ligne]. Disponible sur : <https://fr.mathworks.com/>
- [57] Yang, G.-H., Wang, J. L., and Soh, Y. C. Reliable controller design for linear systems. *Automatica* 37, 5 (2001), 717–725. 3

# **Annexes**

# Annexes A

## Historique et généralités

### A.1 Les premiers quadri-rotors

Les premiers concepteurs d'aéronefs ont expérimenté la configuration à quatre rotors, car l'alternative consistant à utiliser un seul rotor principal avec un rotor de queue semblait complexe et inefficace. En effet, la fabrication de grandes pales de rotor était un énorme problème, de plus, les grands rotors étaient proportionnellement beaucoup plus lourds que les petits rotors [5].

Le premier quadri-rotor était le Gyroplane n°1 « Breguet-Richet », conçu par les frères scientifiques français Louis et Jaque Breguet en collaboration avec le professeur Charles Richet en 1907. Ce Gyroplane de 578 kg réussit tout de même à décoller de 60 cm au-dessus du sol, son vol fut cependant si instable que quatre hommes furent nécessaires pour le tenir. Néanmoins, l'Académie Française des Sciences (AFS) accepta de valoriser cet essai comme le premier décollage motorisé à la verticale d'un homme. Un an plus tard, un autre quadri-rotor Gyroplane n° 2 a été construit. Il était équipé d'un moteur Renault, plus puissant. Malheureusement, l'aéronef fut détruit lors d'un atterrissage un peu brusque. Le quadri-rotor fut réparé, mais la suite des essais des frères Breguet ne déboucha sur aucun résultat significatif [2].

En 1920, Etienne Oemichen a commencé ses expériences dans la conception des aéronefs à voilures tournantes. Six modèles ont été conçus par ce jeune ingénieur de la compagnie Peugeot. Parmi lesquels nous constatons un grand quadri-rotor de 800 kg construit en 1922, avec quatre rotors de diamètre (2x7.60 m + 2x6.40 m) et huit hélices tournées par un seul moteur Gnome Rhône de 135 kW (Figure 1.2). Ce quadri-rotor a montré un degré considérable de stabilité et de contrôlabilité. En 1924, Oemichen a fait un premier vol réussi de 360 m.

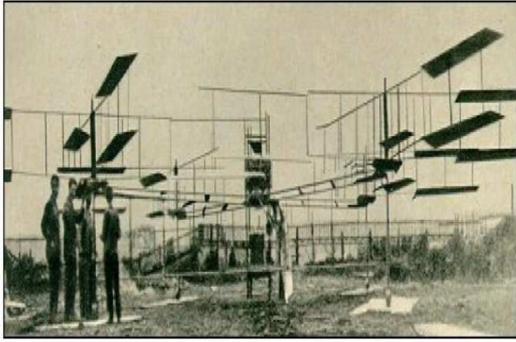


Figure A. 4 : Gyroplane de Bréguet Richet[2]

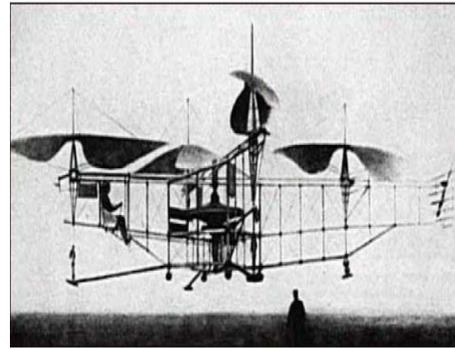


Figure A. 5 : Le quadri-rotor d'Oemichen[2]

Une autre plateforme à vol vertical a été construite pour le service aérien de l'armée américaine à Dayton, Ohio en 1921 par les ingénieurs George de Bothezat et Ivan Jerome . C'était une structure en croix de 1678 kg portant six hélices de diamètre 8.1 m sur chaque axe de longueur 9 m. Plus de 100 essais de vols ont été exécutés sur ce grand quadri-rotor , mais la commande en vol était très difficile et le prototype n'a pas répondu aux exigences d'exécution de l'armée américaine [2].

Dans les années cinquante, Convertawings a construit un quadri-rotor disposant de rotors de diamètre 5.92 m et des ailes pour générer une portance vers l'avant. Deux moteurs Continental de 90 CV ont été utilisés et le véhicule a été commandé en changeant la poussée fournie par chaque rotor. Le quadri-rotor de Convertawings a été piloté avec succès, mais la production a été arrêtée en raison d'un manque d'intérêt commercial pour cet avion.

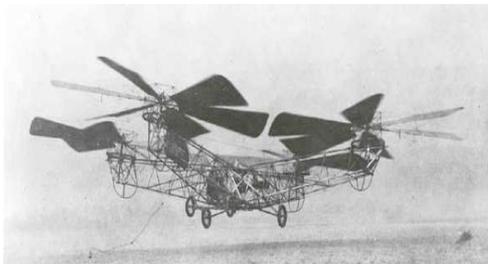


Figure A.6 : Quadri-rotor de Bothezat[5]

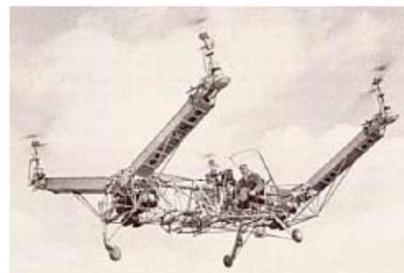


Figure A. 7 : Convertawings[2]

Ces anciens designs avaient un moteur situé au centre du quadri-rotor , entraînant les 4 rotors par l'intermédiaire de courroies ou d'arbres. Les arbres étaient cependant lourds et susceptibles de se briser. Les 4 rotors étaient tous légèrement différents les uns des autres, provoquant un problème d'instabilité dans le quadri-rotor , ce dernier devait être constamment stabilisé- en l'absence d'ordinateurs – ce qui présentait une charge de travail monumentale pour le pilote, en conséquence, les conceptions multi-rotors ont été abandonnées [5].

Au début des années quatre-vingts, avec l'avènement des moteurs électriques et en particulier de la microélectronique, il est devenu possible de construire des multi-rotors fiables et performants. Les quadri-rotors modernes ont un moteur électrique accouplé à chaque rotor, un ordinateur de vol qui ajuste en permanence l'orientation du quadri-rotor en changeant simplement la vitesse de rotation des différents moteurs. Depuis, plusieurs

chercheurs ont commencé à s'intéresser à cette configuration dans des applications mini drones à cause de sa simplicité, sa capacité à supporter une charge utile accrue et son coût réduit [5].

## A.2 La technologie du quadri-rotor

Le quadri-rotor est en général, constitué par les éléments suivants [6], [7] :

- **La carte de commande :**

C'est le cerveau du quadri-rotor, elle est faite à base d'un microcontrôleur. Cette carte est destinée à être embarquée sur le quadri-rotor, elle assure sa commande et son pilotage; elle reçoit de manière permanente les mesures délivrées par les capteurs. La loi de commande calculée à base du modèle du quadri-rotor est implémentée sur le microcontrôleur en utilisant le langage de programmation approprié, ce qui permet à cette carte de traiter les informations, exécuter ces lois stabilisantes et générer des signaux MLI pour commander les vitesses des moteurs.

- **Le système de propulsion :**

- Les moteurs :

Ils permettent de faire tourner les hélices, et ils sont sous différents types mais le plus souvent utilisé est le moteur brushless. Ce dernier se comporte comme un moteur à courant continu traditionnel. Il présente des caractéristiques semblables à celles des moteurs à courant continu et alternatif sans les inconvénients : une forte dynamique de vitesse et d'accélération sans l'usure mécanique des moteurs courant continu ; la commutation électronique se substituant à la commutation mécanique. Ils entrent dans la catégorie des moteurs synchrones.

- Les hélices :

Il existe différents types d'hélices en fonction de leur diamètre et de leur inclinaison. Le choix de l'hélice doit se faire en fonction de la taille du châssis et du moteur. Il faut noter que la force de poussée générée par les propulseurs est proportionnelle au diamètre et à l'inclinaison de l'hélice. Cependant, de grandes hélices nécessitent plus de puissance. Aussi, une hélice de petite ou moyenne taille doit être utilisée si les moteurs tournent à une vitesse élevée. Mais en utilisant des moteurs à bas régime, les hélices doivent être grandes pour qu'elles puissent soulever le quadri-rotor.

- Les variateurs de vitesse :

Les contrôleurs de vitesse ESC (de l'Anglais : Electronic Speed Controller) sont employés afin de faire varier la vitesse du moteur électrique, sa direction et agir comme frein si nécessaire. L'ESC indique au moteur à un instant « t » à quelle vitesse il devrait tourner et ceci en régulant la tension fournie au moteur ainsi que son ampérage. Comme il y a quatre moteurs dans un quadcoptère, on associe à chacun un ESC indépendamment.

- **Les capteurs :**

La prise en compte de la technologie et de la performance des capteurs est très importante dans la conception des micro-drones. Les capteurs doivent permettre de déterminer l'état du drone (position, vitesse, orientation, vitesse de rotation avec un niveau de précision satisfaisant afin d'assurer de bonnes performances. Les capteurs sont généralement classés en deux familles principales [3] : les capteurs proprioceptifs et les capteurs extéroceptifs.

Les capteurs proprioceptifs permettent de mesurer le déplacement du mobile entre deux instants, Tandis que les capteurs extéroceptifs mesurent la situation absolue du mobile par observation de points du repère dont la situation est connue dans un référentiel attaché à l'environnement. Le tableau A.1 répertorie les différents types de capteurs qu'on peut utiliser dans la conception d'un quadri-rotor ainsi que leurs caractéristiques [3].

<b>Familles de capteurs</b>	<b>Capteurs</b>	<b>Caractéristiques</b>
<b>Proprioceptifs</b>	Accéléromètres	<ul style="list-style-type: none"> <li>- Détermination de la position du mobile par double intégration</li> <li>- Peuvent être utilisés pour fournir la direction de la gravité</li> </ul>
	Gyroscopes	<ul style="list-style-type: none"> <li>- <i>Utilisés pour la</i> mesure de la rotation absolue</li> <li>- Utilisés en composants intégrés de centrales inertielles</li> <li>- Il en existe 2 types : gyroscopes mécaniques et gyroscopes à laser.</li> </ul>
	Centrales inertielles IMU	<ul style="list-style-type: none"> <li>- Systèmes complexes et chers</li> <li>- Composées au minimum de 3 accéléromètres et de 3 gyroscopes.</li> <li>- Mesurent l'accélération non-gravitationnelle et la vitesse de rotation instantanées selon les trois directions.</li> </ul>
<b>Extéroceptifs</b>	Compas magnétique	<ul style="list-style-type: none"> <li>- Indique la direction du nord magnétique</li> <li>- Sensibles aux perturbations magnétiques environnantes</li> </ul>
	Gyrocompas	<ul style="list-style-type: none"> <li>- Indicateurs de direction</li> <li>- lourds, onéreux, mais insensibles aux perturbations magnétiques</li> </ul>
	GPS	<ul style="list-style-type: none"> <li>- Système de positionnement par satellite</li> <li>- Utilisés en mode positionnement absolu du mobile (erreur jusqu'à 20 m) ou bien en mode positionnement relatif (erreur réduite aux centimètres).</li> </ul>
	Capteurs télémétriques	<ul style="list-style-type: none"> <li>- Permettent de mesurer la distance par rapport à un obstacle par émission/réception de signaux.</li> <li>- Types : à ultrasons, laser à balayage, radar Ultras-Large-Bande</li> </ul>
	Caméra	<ul style="list-style-type: none"> <li>- Parmi les capteurs extéroceptifs les plus performants</li> <li>- Transmission d'image vue par le mobile</li> <li>- Nécessitent un temps de calcul important, très</li> </ul>

		sensibles au changement d'éclairage
--	--	-------------------------------------

Table A.1 : Les capteurs

- **La communication :**

Un échange de données avec la station de control est requis pour un bon fonctionnement du quadri-rotor . La communication entre le véhicule aérien et la station de control permet d'un côté, la commande manuelle (pilotage) du quadri-rotor afin de mettre à jour la trajectoire de vol, et la réception des données de navigation en temps réel d'un autre côté. La communication peut être réalisée en utilisant [13] :

- Radio control :

Le quadri-rotor peut être commandé à distance en utilisant un émetteur / récepteur RC ordinaire. Le récepteur, monté sur le quadri-rotor , relie le microprocesseur au train d'impulsions provenant de l'émetteur. Le processeur traite les informations et les convertit en signaux de contrôle qui sont envoyés aux actionneurs. La communication avec le récepteur s'effectue via le port de capture d'entrée du microprocesseur.

- Modem radio :

Il permet de créer un réseau privé entre le quadri-rotor et la station de control. Le modem est connecté au processeur à travers un port série et permet un échange en temps réel des données de navigation (position, trajectoire parcourue, ...) et en quantité importante .



Figure A.5 : Système de Radio control



Figure A.6: Xbee Pro 2.4 GHz, [13]

- **Le châssis :**

C'est un élément très important, car c'est la structure qui logera toutes les autres parties du quadri-rotor . Le choix de cette partie influe grandement sur les performances et la stabilité du système. En effet, pour avoir de bonnes caractéristiques de vol, il faut choisir un châssis qui soit symétrique et qui offre le moins de déformation et de flexion possibles et qui soit en mesure d'absorber les vibrations provenant des moteurs. Le poids et la taille du châssis doivent être pris en compte aussi. Selon les matériaux utilisés, on trouve des châssis en aluminium, en contreplaqué ou bien en fibre de carbone.

- **La batterie :**

Le développement des batteries a permis de réduire considérablement leurs poids pour une autonomie plus importante, cependant ; un des grands problèmes rencontrés dans la navigation des drones est la durée de vol qui est limitée par l'autonomie de la batterie, il a été donc donné une grande importance à l'aspect énergétique et l'économie de celle-ci. En effet, il faut faire une étude complète sur les actionneurs à utiliser et leur consommation en énergie ainsi que la durée de vol nécessaire pour un bon dimensionnement [14].

## Annexes B

### Lois de commandes

#### B.1 Matrices de rotation

On considère que les centres  $O_0$  et  $O_1$  des deux repères sont confondus, ce qui signifie que le repère  $R_1$  ne fait que des rotations par rapport au repère  $R_0$ . Trois paramètres indépendants sont nécessaires pour décrire complètement l'orientation du repère  $R_1$  par rapport à celle de  $R_0$ .

Le passage du repère  $R_1$  vers le repère  $R_0$  se fera par trois rotations en utilisant deux repères intermédiaires  $R_i$  et  $R_j$  [15].

- **Passage du repère  $R_0$  vers  $R_i$  :**

La rotation se fait autour de l'axe  $x_i = x_0$ . On passe du repère  $R_0$  vers  $R_i$  en faisant une rotation d'un angle  $\phi$  (l'angle de roulis). La représentation se fait par des figures planes, à partir desquelles nous construisons les matrices de passages (Figure 2.2).

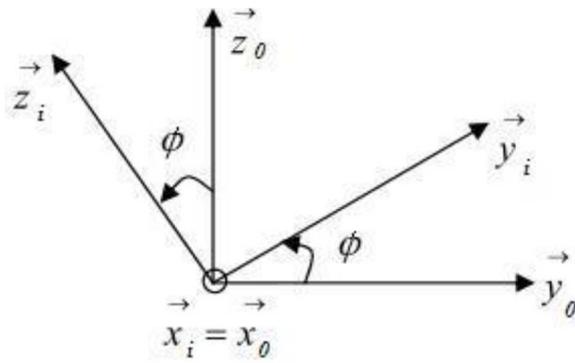


Figure B.1 : Rotation autour de l'axe X (Roulis)

Nous obtenons ainsi la matrice :

$$R(X_0, \phi) = \begin{bmatrix} 1 & 0 & 0 \\ 0 & \cos(\phi) & -\sin(\phi) \\ 0 & \sin(\phi) & \cos(\phi) \end{bmatrix} \quad (2.1)$$

- **Passage du repère  $R_j$  vers le repère  $R_i$  :**

La rotation se fait autour de l'axe  $y_j = y_i$ . On passe du repère  $R_j$  vers le repère  $R_i$  en faisant une rotation d'un angle  $\theta$  (l'angle de tangage).

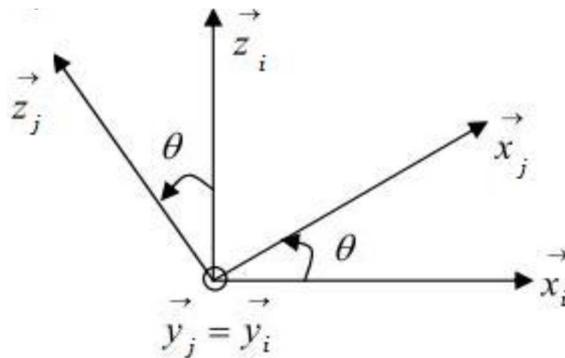


Figure B.2 : Rotation autour de l'axe Y (Tangage)

Nous avons ainsi la matrice :

$$R(Y_0, \theta) = \begin{bmatrix} \cos(\theta) & 0 & \sin(\theta) \\ 0 & 1 & 0 \\ -\sin(\theta) & 0 & \cos(\theta) \end{bmatrix} \quad (2.2)$$

- **Passage du repère  $R_1$  vers le repère  $R_j$  :**

La rotation se fait autour de l'axe  $z_1 = z_j$ . On passe du repère  $R_1$  vers le repère  $R_j$  en faisant une rotation d'un angle  $\psi$  (l'angle de lacet).

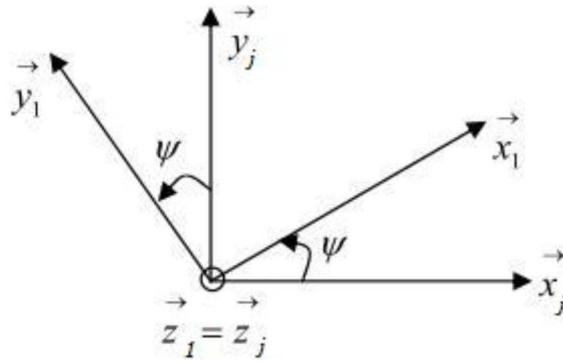


Figure B.3 : Rotation autour de l'axe Z (Lacet)

Et enfin, on obtient la matrice :

$$R(Z_0, \psi) = \begin{bmatrix} \cos(\psi) & -\sin(\psi) & 0 \\ \sin(\psi) & 0 & 0 \\ 0 & 0 & 1 \end{bmatrix} \quad (2.3)$$

Le passage du repère  $R_1$  vers le repère  $R_0$  ou inversement se fait par trois rotations successives de telle sorte que tous les axes de du repère  $R_1$  occupent des positions différentes de celles du repère  $R_0$ . La matrice de passage de  $R_1$  vers  $R_0$  est donnée par le produit des trois matrices successives, on obtient donc :

$$R = R(\phi, \theta, \psi) = R(Z_0, \psi) * R(Y_0, \theta) * R(X_0, \phi)$$

$$R = \begin{bmatrix} C_\theta C_\psi & S_\phi S_\theta C_\psi - C_\phi S_\psi & C_\phi S_\theta C_\psi + S_\phi C_\psi \\ C_\theta S_\psi & C_\phi C_\psi + S_\phi S_\theta S_\psi & C_\phi S_\theta S_\psi - S_\phi C_\psi \\ -S_\theta & S_\phi C_\theta & C_\phi C_\theta \end{bmatrix} \quad (2.4)$$

Avec :  $S_\theta = \sin(\theta)$  ,  $C_\theta = \cos(\theta)$ .

## B.2 Application de la commande $\mathcal{L}_1$ aux quadri-rotors (Etat de l'art)

L'architecture de commande  $\mathcal{L}_1$  a été appliquée dans différents domaines et a démontré sa performance, on prend l'exemple des systèmes volant et plus particulièrement les quadri-rotors où bon nombre de travaux publiés confirment l'efficacité de cette approche. [33] présente une procédure de conception d'une commande  $\mathcal{L}_1$  adaptative par retour de sortie et démontre sa performance en l'appliquant à un quadri-rotor . [34] utilise la même approche de commande pour une application d'inspection des turbines avec quadri-rotor , plusieurs simulations et tests en vol ont été effectués dans les deux travaux et démontrent la robustesse de cette méthode. Dans [35] une approche mixte-déterministe-aléatoire pour la conception de loi de commande  $\mathcal{L}_1$  adaptative est proposée. Le contrôleur est testé en simulation et montre de bonnes performances. [36] s'intéresse à la méthode de synthèse d'un filtre optimal pour la commande  $\mathcal{L}_1$  adaptative. Les résultats expérimentaux démontrent la robustesse aux retards, au bruit, aux perturbations et aux incertitudes de modélisation. Dans [37] un contrôleur  $\mathcal{L}_1$

adaptatif est associé à un régulateur PID afin de compenser les incertitudes dans la boucle de commande des vitesses angulaires de roulis et de tangage. Les performances de ce régulateur ont été vérifiées sous différentes conditions par simulations et tests en vol, la commande  $\mathcal{L}_1$  a permis de diminuer les retards et les dépassements.

La commande par backstepping a été aussi largement associée à la commande  $\mathcal{L}_1$ , en effet ; [38] présente deux architectures de commande en attitude  $\mathcal{L}_1$  adaptatives par backstepping, la première utilise une représentation par angles d'Euler, tandis que la deuxième utilise les quaternions. Ces architectures viennent améliorer la commande par backstepping conventionnelle qui représente des faiblesses face aux incertitudes et qui génère des signaux de commande bruités. Les résultats de simulations ainsi que les tests en vol ont montré que l'architecture qui utilise les quaternions était la meilleure car elle assure la poursuite tout en éliminant les singularités. [39] combine également une approche de commande par backstepping avec un contrôleur  $\mathcal{L}_1$  adaptatif pour compenser les erreurs non linéaires provenant de la conception de la commande par backstepping. De même, [40] utilise la configuration en deux boucles ; la boucle externe consiste en un régulateur  $\mathcal{L}_1$  adaptatif par backstepping assurant la poursuite de trajectoire. Des résultats de simulations ainsi que des tests de vol réel ont été représentés afin d'appuyer cette approche, le système suit la trajectoire même en présence de perturbations inconnues avec une erreur de poursuite tolérable.

Le schéma de commande  $\mathcal{L}_1$  a été utilisée également pour la conception de commandes tolérantes aux défauts ; [41] étudie les performances de la commande  $\mathcal{L}_1$  adaptative pour la commande tolérante aux défauts d'un quadri-rotor . L'architecture de commande proposée consiste en un régulateur LQR utilisé dans une boucle interne pour stabiliser le véhicule, et un régulateur  $\mathcal{L}_1$  adaptatif appliqué dans la boucle externe afin d'estimer et de compenser les incertitudes et les défauts. Les résultats de simulation ont montré que la commande  $\mathcal{L}_1$  a permis d'améliorer les performances en cas de pannes partielles des actionneurs. [42] présente un régulateur  $\mathcal{L}_1$  adaptatif augmenté par feed-forwarding, cette approche a été validée par simulations. Cette commande est aussi présentée dans [43] où une commande tolérante aux défauts a été conçue pour la poursuite de trajectoire à partir d'un modèle ayant des paramètres incertains et en présence de perturbations extérieures. Les résultats de simulation montrent la robustesse de la boucle fermée.

D'autres schémas de commande avec la méthode  $\mathcal{L}_1$  ont été utilisés comme dans [44] où une approche qui consiste à combiner une commande linéaire robuste  $H_\infty$  avec une commande adaptative a été proposée. Deux architectures ont été utilisées, la première suivant la configuration MRAC et la seconde en utilisant l'architecture  $\mathcal{L}_1$  adaptative. Les résultats de simulations montrent que la commande adaptative réagit aux perturbations et les compense de manière plus rapide que le régulateur  $H_\infty$  utilisé seul. Des tests réels ont été effectués sur un quadri-rotor fixé sur un banc d'essais (tous les degrés de liberté sont fixés sauf l'angle de tangage), les résultats expérimentaux montrent une amélioration des performances en poursuite dans le cas où la commande adaptative est introduite. [45] propose un schéma de commande qui combine entre la commande  $\mathcal{L}_1$  adaptative et l'ILC (Iterative Learning Control) afin d'améliorer la poursuite de trajectoire. La commande  $\mathcal{L}_1$  compense les

perturbations et les dynamiques inconnues tandis que l'ILC permet d'améliorer la poursuite en se basant sur l'expérience obtenue des itérations précédentes, mais les performances obtenues restent limitées. Les résultats expérimentaux ont montré une amélioration significative des performances de la configuration  $\mathcal{L}_1$ -ILC comparée à l'approche PD-ILC.

Les travaux précédemment mentionnés ainsi que d'autres qu'on n'a pas cités, viennent confirmer l'efficacité de la commande  $\mathcal{L}_1$  adaptative appliquée aux systèmes volants.

### B.3 L'opérateur *Proj* [36]:

Considérons un ensemble compact convexe avec une limite par la borne suivante:

$$\Omega_c \cong \{\theta \in R^n | f(\theta) \leq c\}, 0 \leq c \leq 1$$

Avec  $f: R^n \rightarrow R$  est donnée par :

$$f(\theta) = \frac{\theta^T \theta - \theta_{max}^2}{\epsilon_\theta \theta_{max}^2}$$

Avec  $\theta_{max}$  est la norme liée au vecteur des paramètres  $\theta$ , et  $\epsilon_\theta$  désigne la tolérance de la convergence de notre choix. Soit  $\theta^*$  la vraie valeur du paramètre  $\theta$ , et qui appartient à  $\Omega_0$ ,

L'opérateur de projection peut donc être défini par :

$$Proj(\theta, y) \cong \begin{cases} y & \text{si } f(\theta) \leq 0 \\ y & \text{si } f(\theta) \geq 0 \text{ et } \nabla f^T y \leq 0, \\ y - \frac{\nabla f}{\|\nabla f\|} \left\langle \frac{\nabla f^T}{\|\nabla f\|}, y \right\rangle f(\theta) & \text{si } f(\theta) \geq 0 \text{ et } \nabla f^T y > 0 \end{cases}$$

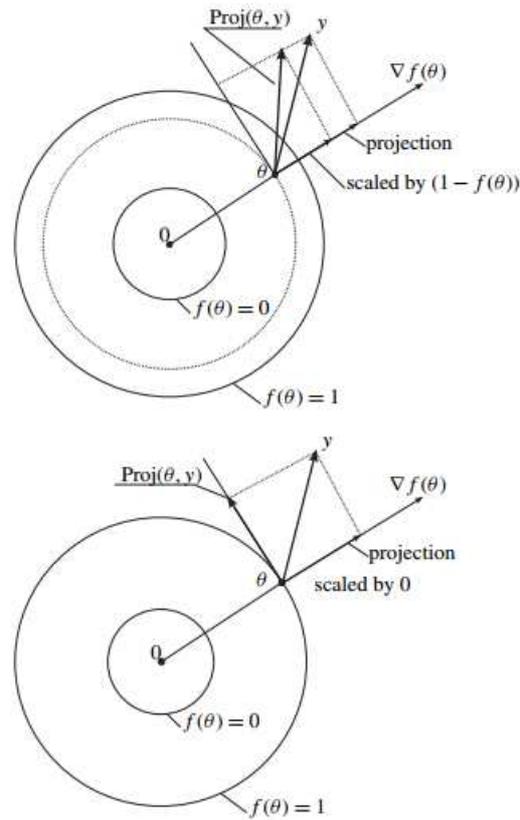


Figure B.4 : Illustration de l'opérateur Proj(.,.)

## B.4 Analyse de commande :

L'analyse de l'architecture de commande est une extension aux systèmes MIMO des résultats montrés pour les systèmes SISO.

Soit :

$$L = \max_{\theta \in \Theta} \|\theta\|_1,$$

$$G(s) = H(s)(\mathbb{I} - C(s)).$$

La commande adaptative  $\mathcal{L}_1$  est définie par les équation (2.51) -(2.58) et est soumise à la condition suivant de la norme  $\mathcal{L}_1$

$$\|G(s)\|_{\mathcal{L}_1} L < 1.$$

De plus, la conception de  $C(s)$  doit garantir que la matrice de transfert

$$G_u(s) = (s\mathbb{I} - A_m)^{-1} - H(s)C(s)H_m^{-1}(s)H_0(s)$$

Est une matrice propre et stable.

Une autre condition est d'assurer aussi que  $C(s)H_m^{-1}(s)$  est une matrice de transfert propre et stable.

### Système de référence en boucle fermée :

Le système à référence est défini par :

$$\begin{aligned}\dot{x}_r(t) &= A_m x_r(t) + B(\omega u_r(t) + \theta^T x_r(t) + \eta_m(t, x_r)) + \eta_u(t, x_r), \\ y_r(t) &= C x_r(t).\end{aligned}$$

La loi de commande de référence est donnée par :

$$u_r(s) = \omega^{-1}C(s) \left( K_g r(s) - v_{1r}(s) - v_{2r}(s) \right),$$

Où  $v_{1r}(s)$  est la transformée de Laplace du terme  $\theta^T x_r(t) + \eta_m(t, x_r)$  ;  $\eta_{ur}(s)$  est la transformée de Laplace de  $\eta_u(t, x_r)$ , et  $v_{2r} = H_m^{-1}(s)H_0(s)\eta_{ur}(s)$ .

**Lemme 2.3** si le filtre  $C(s)$  est conçu de telle sorte qu'il vérifie la condition de la norme  $\mathcal{L}_1$  dans l'équation (2.60), alors le système de référence en boucle fermée de l'équation (2.62) et (2.63) est BIBS stable par rapport à l'entrée de référence et aux conditions initiales.

### Démonstration

Le système de référence en boucle fermée (2.62) et (2.63) peut s'écrire comme suit :

$$x_r(s) = H(s)C(s)K_g r(s) + G(s)\theta^T x_r(s) + G(s)\eta_{mr}(s) + G_u(s)\eta_{ur}(s) + x_{in}(s),$$

Où  $x_{in}(s) = (s\mathbb{I} - A_m)^{-1}x_0$ .

Alors, pour tout  $t \in [0, \tau]$  nous avons

$$\|x_{r\tau}\|_{\mathcal{L}_\infty} \leq \frac{\|H(s)C(s)\|_{\mathcal{L}_1} K_g \|r_\tau\|_{\mathcal{L}_\infty} + \|G(s)\|_{\mathcal{L}_1} L_m + \|G_u(s)\|_{\mathcal{L}_1} L_u + \|x_{in}\|_{\mathcal{L}_\infty}}{1 - \|G(s)\|_{\mathcal{L}_1} L}$$

Si la condition de la norme  $\mathcal{L}_1$  dans (2.12) est vérifiée, alors  $\|x_{r\tau}\|_{\mathcal{L}_\infty}$  est uniformément bornée pour tout  $\tau > 0$ , et la démonstration est complétée.

### Performance transitoire et statique :

Le Lemme suivant indique que l'erreur de prédiction  $\tilde{x}(t)$  et les erreurs d'estimation des perturbations  $\eta_m(t, x_r)$  et  $\eta_u(t, x_r)$  sont bornées.

**Lemme 2.4** la borne suivante tient pour la norme de l'erreur de prédiction

$$\|\tilde{x}\|_{L_\infty} \leq \rho,$$

Où  $\rho > 0$  est arbitrairement petit.

En outre, si le système en boucle fermée est stable alors l'erreur de prédiction  $\tilde{x}(t)$  converge vers zéro, i.e.,

$$\lim_{t \rightarrow \infty} \tilde{x}(t) = 0.$$

### Démonstration

De (2.50) et (2.51), les dynamiques de l'erreur de prédiction peuvent s'écrire

$$\dot{\tilde{x}} = A_m \tilde{x} + B(\tilde{\omega}u + \tilde{\theta}^T x + \tilde{\eta}_m) + \tilde{\eta}_u.$$

On considère une fonction de Lyapunov candidate

$$V = \frac{1}{2} \sigma^T \sigma + \frac{1}{2} \Gamma^{-1} (tr(\tilde{\theta}^T \tilde{\theta}) + tr(\tilde{\omega}^T \tilde{\omega}) + \tilde{L}_m^2 + \tilde{L}_u^2).$$

$$\dot{V} = \sigma^T \dot{\sigma} + \Gamma^{-1} (tr(\tilde{\theta}^T \dot{\tilde{\theta}}) + tr(\tilde{\omega}^T \dot{\tilde{\omega}}) + \tilde{L}_m \dot{\tilde{L}}_m + \tilde{L}_u \dot{\tilde{L}}_u).$$

De (2.52) et (2.69) la dérivée de la surface de glissement peut s'écrire

$$\dot{\sigma} = \lambda A_m \tilde{x} + \lambda B(\tilde{\theta}^T x + \tilde{\omega}u + \tilde{\eta}_m) + \lambda \tilde{\eta}_u.$$

En remplaçant dans (2.71), il vient que

$$\begin{aligned} \dot{V} = & \sigma^T \left( \lambda A_m \tilde{x} + \lambda B(\tilde{\theta}^T x + \tilde{\omega}u + (\hat{\eta}_m - \eta_m)) + \lambda(\hat{\eta}_u - \eta_u) \right) \\ & + \Gamma^{-1} \left( tr(\tilde{\theta}^T \dot{\tilde{\theta}}) + tr(\tilde{\omega}^T \dot{\tilde{\omega}}) + \tilde{L}_m \dot{\tilde{L}}_m + \tilde{L}_u \dot{\tilde{L}}_u \right). \end{aligned}$$

Etant donné que pour tout scalaire  $s$ ,  $tr(s) = s$ , alors

$$\begin{aligned} \dot{V} = & \sigma^T \lambda A_m \tilde{x} + tr(\sigma^T \lambda B \tilde{\theta}^T x) + tr(\sigma^T \lambda B \tilde{\omega}u) + \sigma^T \lambda B(\hat{\eta}_m - \eta_m) + \sigma^T \lambda(\hat{\eta}_u - \eta_u) \\ & + \Gamma^{-1} \left( tr(\tilde{\theta}^T \dot{\tilde{\theta}}) + tr(\tilde{\omega}^T \dot{\tilde{\omega}}) + \tilde{L}_m \dot{\tilde{L}}_m + \tilde{L}_u \dot{\tilde{L}}_u \right). \end{aligned}$$

En utilisant la propriété  $tr(M1 M2) = tr(M2 M1)$  pour n'importe quelles matrices  $M1, M2$ , nous obtenons

$$\begin{aligned} \dot{V} = & \sigma^T \lambda A_m \tilde{x} + tr(\tilde{\theta}^T x \sigma^T \lambda B) + tr(\tilde{\omega}u \sigma^T \lambda B) + \sigma^T \lambda B(\hat{\eta}_m - \eta_m) + \sigma^T \lambda(\hat{\eta}_u - \eta_u) \\ & + \Gamma^{-1} \left( tr(\tilde{\theta}^T \dot{\tilde{\theta}}) + tr(\tilde{\omega}^T \dot{\tilde{\omega}}) + \tilde{L}_m \dot{\tilde{L}}_m + \tilde{L}_u \dot{\tilde{L}}_u \right). \end{aligned}$$

$\hat{\eta}_m$  et  $\hat{\eta}_u$  étant donnés par (2.53) et (2.55) et la loi d'adaptation (2.57) ça peut s'écrire

$$\dot{V} = -\alpha \sigma^T \sigma - \sigma^T \lambda B \eta_m - \sigma^T \lambda \eta_u - \|\sigma^T \lambda B\| \hat{L}_m - \|\sigma^T \lambda\| \hat{L}_u + \Gamma^{-1} (\tilde{L}_m \dot{\tilde{L}}_m + \tilde{L}_u \dot{\tilde{L}}_u).$$

Par conséquent, la borne supérieure suivante peut être dérivée

$$\begin{aligned} \dot{V} \leq & -\alpha \|\sigma\|^2 + \|\sigma^T \lambda B\| \|\eta_m\| + \|\sigma^T \lambda\| \|\eta_u\| - \|\sigma^T \lambda B\| \hat{L}_m - \|\sigma^T \lambda\| \hat{L}_u \\ & + \Gamma^{-1} (\tilde{L}_m \hat{L}_m + \tilde{L}_u \hat{L}_u). \end{aligned}$$

En utilisant la supposition 2.5, il en suit que

$$\dot{V} \leq -\alpha \|\sigma\|^2 - \|(\lambda B)^T \sigma\| \hat{L}_m - \|\lambda^T \sigma\| \hat{L}_u + \Gamma^{-1} (\tilde{L}_m \hat{L}_m + \tilde{L}_u \hat{L}_u).$$

En prenant en compte la loi d'adaptation (2.6) et (2.56), il vient que

$$\dot{V} \leq -\alpha \|\sigma\|^2.$$

Par conséquent, la surface de glissement  $\sigma$ , les erreurs d'estimation  $\tilde{\theta}, \tilde{\omega}$  ; et les erreurs de perturbations  $\tilde{L}_m$  et  $\tilde{L}_u$  sont uniformément bornées.

De même pour la démonstration du Lemme 2.2, puisque les coefficients de la surface de glissement forment un collecteur stable et  $\tilde{x}(0) = 0$ , i.e., le système est initialisé sur la surface de glissement, et étant donné que sur la surface de glissement les trajectoires sont gouvernées par  $\sigma(\tilde{x}, t) = 0$ , il existe toujours un réel arbitrairement petit  $\rho > 0$  vérifiant

$$\|\tilde{x}\|_{L_\infty} \leq \rho.$$

Ce résultat vient de la propriété fondamentale de la commande par modes glissants, stipulant que le système est sur la surface de glissement, il reste à proximité de la surface de glissement malgré les perturbations [57].

De plus, de (2.79) on peut écrire

$$\int_0^t \|\sigma(t)\|^2 dt \leq \frac{1}{\alpha} (V(0) - V(t)).$$

Puisque  $V(0)$  est bornée et  $V(t)$  est bornée et n'est pas croissante, alors

$$\lim_{t \rightarrow \infty} \int_0^t \sigma(t)^2 dt$$

Est bornée.

Si le système en boucle fermée est stable, i.e.,  $u(t)$  et  $x(t)$  sont bornés alors  $\dot{\sigma}(t)$  dans (2.72) est bornée. En appliquant le Lemme de Barbalat il vient que

$$\lim_{t \rightarrow \infty} \|\sigma(t)\|^2 = 0 \text{ et } \lim_{t \rightarrow \infty} \|\sigma(t)\| = 0.$$

Par conséquent

$$\lim_{t \rightarrow \infty} \tilde{x}(t) = 0,$$

Et la démonstration est complète.

Par la suite, dans le théorème suivant, les limites de performance de la commande adaptative  $\mathcal{L}_1$  sont indiquées.

**Théorème 2.2** étant donné le système (2.50), le système de référence (2.62) et (2.63) et la commande adaptative  $\mathcal{L}_1$  (2.58), (2.51), (2.57) et (2.54), nous avons

$$\|x_r - x\|_{\mathcal{L}_\infty} \leq \gamma_1,$$

$$\|u_r - u\|_{\mathcal{L}_\infty} \leq \gamma_2,$$

Où

$$\gamma_1 = 2 \frac{\|G(s)\|_{\mathcal{L}_1}}{1 - \|G(s)\|_{\mathcal{L}_1} L} L_m + 2 \frac{\|G(s)\|_{\mathcal{L}_1}}{1 - \|G(s)\|_{\mathcal{L}_1} L} L_u + \|\omega^{-1} C(s)\|_{\mathcal{L}_1} \|C(s) H_m^{-1}(s) C\|_{\mathcal{L}_1} \rho,$$

Et

$$\gamma_2 = \|\omega^{-1} C(s)\|_{\mathcal{L}_1} (L\gamma_1 + 2(L_m + \|H_m^{-1}(s) H_0(s)\|_{\mathcal{L}_1} L_u) + C(s) H_m^{-1}(s) \rho).$$

### Démonstration :

La loi de commande dans (2.58) peut s'écrire

$$u(s) = KD(s) \left( K_g r(s) - \omega u(s) - \theta^T x(s) - \eta_m(s) - H_m^{-1}(s) H_0(s) (\eta_u(s) + \tilde{\eta}_u(s)) - \tilde{v}(s) \right),$$

Où  $\tilde{v}(s)$  est la transformée de Laplace de la fonction  $\tilde{\omega}u + \tilde{\theta}x(t) + \tilde{\eta}_m(t)$  et  $\tilde{\eta}_u(s)$  est la transformée de Laplace de  $\tilde{\eta}_u(t)$ . Par conséquent

$$u(s) = KD(s) (\mathbb{I} + \omega KD(s))^{-1} \left( K_g r(s) - \theta^T x(s) - \eta_m(s) - H_m^{-1}(s) H_0(s) (\eta_u(s) + \tilde{\eta}_u(s)) - \tilde{v}(s) \right).$$

Qui induit à

$$u(s) = \omega^{-1}C(s) \left( K_g r(s) - \theta^T x(s) - \eta_m(s) - H_m^{-1}(s)H_0(s)(\eta_u(s) + \tilde{\eta}_u(s)) - \tilde{v}(s) \right).$$

Par conséquent, la transformée de Laplace du système en boucle fermée (2.50) et (2.89) peut s'écrire

$$\begin{aligned} x(s) &= H(s)C(s)K_g r(s) + G(s)\theta^T x(s) + G(s)\eta_m(s) + G_u(s)\eta_u(s) \\ &\quad - H(s)C(s)(\tilde{v}(s) + H_m^{-1}(s)H_0(s)\tilde{\eta}_u(s)) + x_{in}(s). \end{aligned}$$

En prenant la différence de (2.64) et (2.90) il suit que

$$\begin{aligned} x_r(s) - x(s) &= G(s)\theta^T(x_r(s) - x(s)) + G(s)(\eta_m(s) - \eta_{mr}(s)) + G_u(s)(\eta_u(s) - \eta_{ur}(s)) \\ &\quad + H(s)C(s)(\tilde{v}(s) + H_m^{-1}(s)H_0(s)\tilde{\eta}_u(s)). \end{aligned}$$

De (2.69), la transformée de Laplace des dynamiques de l'erreur de prédiction peut s'écrire

$$\tilde{x}(s) = H(s)\tilde{v}(s) + (s\mathbb{I} - A_m)^{-1}\tilde{\eta}_u(s).$$

En multipliant les termes de (2.92) par  $H_m^{-1}(s)C$  on obtient

$$H_m^{-1}(s)C\tilde{x}(s) = \tilde{v}(s) + H_m^{-1}(s)H_0(s)\tilde{\eta}_u(s).$$

En substituant dans (2.91) nous aurons

$$\begin{aligned} x_r(s) - x(s) &= G(s)\theta^T(x_r(s) - x(s)) + G(s)(\eta_m(s) - \eta_{mr}(s)) + G_u(s)(\eta_u(s) - \eta_{ur}(s)) \\ &\quad + H(s)C(s)H_m^{-1}(s)C\tilde{x}(s). \end{aligned}$$

En résolvant pour  $x_r(s) - x(s)$ , la borne suivante tient pour  $t \in [0, \tau]$

$$\begin{aligned} \|(x_r - x)_\tau\|_{\mathcal{L}_\infty} &\leq \frac{\|G(s)\|_{\mathcal{L}_1}}{1 - \|G(s)\|_{\mathcal{L}_1}L} \|(\eta_m - \eta_{mr})_\tau\|_{\mathcal{L}_\infty} \\ &\quad + \frac{\|G(s)\|_{\mathcal{L}_1}}{1 - \|G(s)\|_{\mathcal{L}_1}L} \|(\eta_u - \eta_{ur})_\tau\|_{\mathcal{L}_\infty} \\ &\quad + \frac{\|H(s)C(s)H_m^{-1}(s)C\|_{\mathcal{L}_1}}{1 - \|G(s)\|_{\mathcal{L}_1}L} \|\tilde{x}_\tau\|_{\mathcal{L}_\infty}. \end{aligned}$$

Etant donné la borne supérieure de  $\tilde{x}(t)$  du Lemme 2.4, et les bornes de perturbation de la supposition 2.5, in vient que

$$\|(x_r - x)_\tau\|_{\mathcal{L}_\infty} \leq 2 \frac{\|G(s)\|_{\mathcal{L}_1}}{1 - \|G(s)\|_{\mathcal{L}_1}L} L_m + 2 \frac{\|G(s)\|_{\mathcal{L}_1}}{1 - \|G(s)\|_{\mathcal{L}_1}L} L_u + \frac{\|H(s)C(s)H_m^{-1}(s)C\|_{\mathcal{L}_1}}{1 - \|G(s)\|_{\mathcal{L}_1}L} \rho,$$

Ce qui conduit à la borne dans (2.85).

Pour montrer la deuxième borne dans (2.86), en prenant la différence de (2.63) et (2.89), la borne de commande peut s'écrire

$$\begin{aligned} u_r(s) - u(s) &= -\omega^{-1}C(s)\theta^T(x_r(s) - x(s)) - \omega^{-1}C(s)(\eta_m(s) - \eta_{mr}(s)) \\ &\quad - \omega^{-1}C(s)H_m^{-1}(s)H_0(s)(\eta_u(s) - \eta_{ur}(s)) \\ &\quad + \omega^{-1}C(s)(\tilde{v}(s) + H_m^{-1}(s)H_0(s)\tilde{\eta}_u(s)). \end{aligned}$$

Par conséquent

$$\begin{aligned} u_r(s) - u(s) &= -\omega^{-1}C(s)\theta^T(x_r(s) - x(s)) - \omega^{-1}C(s)(\eta_m(s) - \eta_{mr}(s)) \\ &\quad - \omega^{-1}C(s)H_m^{-1}(s)H_0(s)(\eta_u(s) - \eta_{ur}(s)) + \omega^{-1}C(s)H_m^{-1}(s)C\tilde{x}(s). \end{aligned}$$

Et (2.97) peut être bornée supérieurement par

$$\begin{aligned} \|(u_r - u)_\tau\|_{\mathcal{L}_\infty} &\leq \|\omega^{-1}C(s)\|_{\mathcal{L}_1} L \|(x_r - x)_\tau\|_{\mathcal{L}_\infty} \\ &\quad + 2\|\omega^{-1}C(s)\|_{\mathcal{L}_1} (L_m + \|H_m^{-1}(s)H_0(s)\|_{\mathcal{L}_1} L_u) \\ &\quad + \|\omega^{-1}C(s)\|_{\mathcal{L}_1} \|C(s)H_m^{-1}(s)C\|_{\mathcal{L}_1} \|\tilde{x}_\tau\|_{\mathcal{L}_\infty}, \end{aligned}$$

Qui est uniforme pour tout  $\tau \geq 0$ , conduisant à la borne (2.86).

## B.5 Commande tolérante aux défauts

- **Défauts et défaillances**

Un défaut est un dysfonctionnement d'un ou plusieurs éléments du système qui peut dégrader ses performances, mais il peut être tolérable et n'affectant pas le bon fonctionnement du système. Le cas extrême d'un défaut est la défaillance totale, c'est une panne catastrophique d'un composant ou d'un système, pour faire face à une telle situation, une commande reconfigurable est nécessaire et une redondance matérielle doit être disponible.

- **Classification des défauts**

- **Classification selon la localisation**

Un défaut peut se produire dans les trois parties suivantes :

**a/ Défauts d'actionneurs :**

Les défauts actionneurs représentent une perte totale ou partielle de l'action de commande. Un actionneur complètement perdu est un actionneur bloqué qui ne fonctionne pas. L'actionneur partiellement défaillant ne produit qu'une partie de l'action nominale. Dans le cas d'un quadri-rotor, les actionneurs sont les moteurs équipés avec des hélices, une perte d'efficacité d'un moteur ou dégradation d'une hélice (impact, changement de forme) affecte directement le comportement du système.

Les différents types de défauts actionneurs peuvent être représentés par [41] :

$$u_i(t) = \begin{cases} u_{c_i}(t), & \text{pour } t \geq t_0, & \text{Sans défaut} \\ 0, & \text{pour } t \geq t_{F_i}, & \text{Oscillation} \\ u_{c_i}(t_{F_i}), & \text{pour } t \geq t_{F_i}, & \text{Blocage} \\ k_{a_i}(t)u_{c_i}(t), 0 < \underline{k_{a_i}} \leq k_{a_i}(t) \leq 1 & \text{pour } t \geq t_{F_i}, & \text{Perte d'efficacité} \\ u_{i_{min}} \vee u_{i_{max}}(t), & \text{pour } t \geq t_{F_i}, & \text{Saturation} \end{cases} \quad (4.8)$$

Où  $u_i(t)$  est le signal de sortie de l'actionneur,  $u_{c_i}(t)$  est le signal de commande appliqué au  $i^{\text{ème}}$  actionneur,  $t_{F_i}$  est l'instant d'occurrence du défaut du  $i^{\text{ème}}$  actionneur,  $k_{a_i}$  est le coefficient d'efficacité,  $u_{i_{min}}$  et  $u_{i_{max}}$  représentent respectivement les deux limites inférieure et supérieure du signal de sortie du  $i^{\text{ème}}$  actionneur.

**b/ Défauts de capteurs :**

Les défauts capteurs représentent des mesures incorrectes d'un signal. Ces défauts peuvent dégrader la robustesse et les performances du système en boucle fermée.

Les défauts capteurs les plus connus sont : le biais, la dérive, la perte d'efficacité, le blocage et l'erreur de calibrage et ils peuvent être modélisés comme suit [41], [42] :

$$y_i(t) = \begin{cases} x_i(t), & \text{pour } t \geq t_0, & \text{Sans défaut} \\ x_i(t) + b_{s_i}, \dot{b}_{s_i} = 0, b_{s_i}(t_{F_{s_i}}) \neq 0 & \text{pour } t \geq t_{F_i}, & \text{Biais} \\ x_i(t) + b_{s_i}(t), |b_{s_i}(t)| = \lambda_i t, 0 < \lambda_i \ll 1 & \text{pour } t \geq t_{F_i}, & \text{Dérive} \\ x_i(t) + b_{s_i}(t), |b_{s_i}(t)| < \bar{b}_{0_i}, \dot{b}_{s_i}(t) \rightarrow 0 & \text{pour } t \geq t_{F_i}, & \text{Perte de précision} \\ x_i(t_{F_{s_i}}), & \text{pour } t \geq t_{F_i}, & \text{Blocage} \\ k_{s_i} x_i(t), 0 < \underline{k}_{s_i} \leq k_{s_i}(t) \leq 1 & \text{pour } t \geq t_{F_i}, & \text{Erreur de calibrage} \end{cases} \quad (4.2)$$

Où  $t_{F_i}$  représente l'instant d'occurrence d'un défaut au  $i^{\text{ème}}$  capteur,  $k_{s_i}$  modélise la perte d'efficacité (erreur de calibrage) du capteur,  $b_{s_i}$  modélise le biais, la dérive et la perte de précision du capteur.

### c/ Défauts de composants :

Les défauts composants sont les défauts qui proviennent du système lui-même, souvent ces défauts n'appartiennent pas à la catégorie des défauts capteurs et actionneurs. Ils représentent des changements dans les paramètres du système. Les défauts composants se traduisent par un changement dans l'équation d'état.

- **Classification selon la modélisation**

Les défauts peuvent être modélisés comme défauts additifs ou multiplicatifs.

#### - Défauts additifs :

Les défauts additifs sont considérés comme des signaux externes supplémentaires, qui correspondent à des changements constatés indépendamment des entrées connues. Ces défauts sont appropriés pour représenter des défauts de composants dans le système à commander.

#### - Défauts multiplicatifs :

Les défauts multiplicatifs correspondent à des changements de paramètres qui affectent l'évolution des entrées et/ou des sorties, où l'amplitude de ces défauts dépend des entrées connues. Notons que les défauts d'actionneurs et les défauts capteurs sont le plus souvent multiplicatifs par nature.

- **Classification selon les caractéristiques temporelles**

En fonction de la caractéristique temporelle, les défauts se divisent en défauts graduels et abrupts [43], [44] :

#### - Défauts graduels :

Ils représentent des changements paramétriques lents, qui sont souvent dus au vieillissement. Ils sont difficiles à détecter à cause de leur caractéristique temporelle lente,

mais ils ne sont pas sévères. Cependant, si ces défauts ne sont pas pris en charge rapidement, ils peuvent conduire à une situation grave.

Les défauts graduels peuvent être modélisés comme suit :

$$f_i(x, u, t) = \beta_i(t - t_{F_i})\delta_i(x, u) \quad (4.3)$$

$$\beta_i(t - t_{F_i}) = \begin{cases} 0, & \text{pour } t < t_{F_i} \\ 1 - e^{-\alpha_i(t-t_{F_i})}, & \text{pour } t \geq t_{F_i} \end{cases} \quad \begin{array}{l} \text{Absence de défaut} \\ \text{Présence de défaut} \end{array} \quad (4.4)$$

Où  $f_i(x, u, t)$  représente la fonction non linéaire du  $i^{\text{ème}}$  défaut,  $\beta_i(t - t_{F_i})$  caractérise le profil temporel de ce défaut qui se produit à un instant inconnu  $t_{F_i}$ ,  $\alpha_i$  est une constante positive qui représente le taux d'évolution du défaut,  $\delta_i(x, u)$  désigne le changement de la dynamique du système au  $i^{\text{ème}}$  défaut. Cette caractéristique permet au même temps d'avoir des défauts additifs et multiplicatifs ( $\delta_i$  peut-être en fonction du vecteur d'état  $x$  et/ou du vecteur de commande  $u$ ). Il est à noter que les défauts graduels sont caractérisés par des faibles valeurs de  $\alpha_i$ .

#### - Défauts abrupts :

Ils se reproduisent instantanément souvent à la suite d'un dommage matériel. Ces défauts peuvent être très sévères car, s'ils affectent les performances et/ou la stabilité du système bouclé, une réaction rapide du système FTC est nécessaire. Mathématiquement, ces défauts peuvent être modélisés comme un cas particulier des défauts graduels, lorsque la constante  $\alpha_i$  prend des grandes valeurs, ce qui rend le profil temporel  $\beta_i(t - t_{F_i})$  proche d'un échelon unitaire.

$$\beta_i(t - t_{F_i}) = \begin{cases} 0, & \text{pour } t < t_{F_i}, \\ 1, & \text{pour } t \geq t_{F_i}, \end{cases} \quad \begin{array}{l} \text{Absence de défaut} \\ \text{Présence de défaut} \end{array} \quad (4.5)$$

#### - Défauts intermittents :

Ce sont des défauts qui apparaissent et disparaissent à plusieurs reprises. Ils représentent également un cas particulier de défauts abrupt, avec la propriété particulière qu'ils reviennent d'une façon aléatoire à leurs valeurs normales. Ces défauts peuvent être causés par un contact intermittent ou un câblage partiellement endommagé.

### • Classification des systèmes FTC

La commande FTC est une stratégie dans l'architecture et la conception des systèmes de commande permettant de s'assurer qu'un système en boucle fermée peut continuer à fonctionner de manière tolérante face aux défauts de l'actionneur, du capteur ou du processus lié. Lorsque la stabilité prescrite et les indices de performance en boucle fermée sont maintenus malgré l'action des défauts, le système est dit "tolérant aux défauts" et le schéma de commande qui assure la tolérance aux pannes est dit: contrôleur tolérant aux pannes. L'objectif de la conception d'un système FTC est de garantir que le système en boucle fermée conserve la stabilité et une performance acceptable pendant une ou plusieurs actions de panne.

Selon la manière avec laquelle le problème est abordé, les systèmes FTC sont généralement divisés en deux classes : **passive** et **active** . Le choix de l'approche à utiliser dépend de la capacité à déterminer les défauts qu'un système peut subir à la phase de conception, des changements induits par le défaut et du type de redondance utilisé dans le système.

#### - Systèmes FTC passifs

Les systèmes PFTC (Passive Fault Tolerant Control systems) sont basés sur des commandes robustes, avec une structure fixe et sans aucun ajustement en ligne, afin de rendre le système en boucle fermée insensible à certains défauts présumés. Cette approche n'exige pas un module de diagnostic de défauts, elle traite les défauts comme étant des sources d'incertitude, ce qui la rend sur le plan des calculs moins complexe et plus attractive. Cependant, elle est très restrictive à cause des inconvénients suivants :

- Afin d'assurer la robustesse aux défauts, généralement un sous-ensemble très restreint des défauts possibles peut être considéré. Souvent, seulement les défauts qui ont un "petit effet" sur le comportement du système qui peuvent être traités de cette manière.
- L'obtention d'une bonne robustesse vis-à-vis de certains défauts n'est possible, qu'au prix d'une diminution des performances nominales. Puisque le système est affecté très rarement par les défauts, il n'est pas raisonnable de dégrader considérablement les performances du système en absence de défauts, uniquement pour obtenir une certaine insensibilité à une classe restreinte de défauts.

Les approches FTC passives les plus répandues dans la littérature sont : la commande linéaire quadratique (LQ), la commande robuste  $H_\infty$  et la commande basée sur les inégalités matricielles linéaires (LMIs).

#### - Systèmes FTC actifs

Les AFTCS (Active Fault Tolerant Control Systems) sont conçus en se basant sur les informations en temps réel des défauts fournis par des systèmes de détection et de diagnostic de défauts (Fault Detection and Diagnosis, FDD). Plus précisément, les systèmes FTC actifs réagissent activement aux défauts/défaillances, en reconfigurant les lois de commande de manière à ce que la stabilité et les performances du système en boucle fermée restent maintenues. Dans certains cas critiques, des performances dégradées peuvent être acceptées.

La structure d'un système AFTC est généralement plus compliquée par rapport à celle des systèmes PFTC, mais elle peut traiter une classe plus large avec des types plus intensifs de défauts.

Les systèmes FTC actifs peuvent être distingués des schémas PFTC par deux aspects :

- **Niveau de supervision (supervision level) :** Un bloc de diagnostic de défauts FDD, détection et isolation de défauts (Fault Detection and Isolation, FDI), estimation de défauts (Fault Estimation, FE).
- **Niveau d'exécution (execution level) :** Un mécanisme de reconfiguration/ restructuration (Reconfiguration Mechanism, RM).

Les méthodes AFTC peuvent être divisées en trois classes :

- **Les méthodes à base de projection :** Dans ces méthodes, l'un des contrôleurs précalculés hors ligne pour chaque scénario de défaut spécifique est sélectionné, en fonction des informations de défaut fournies par le schéma FDD.
- **Les méthodes de reconception en ligne :** Dans ces méthodes, en fonction des informations de défauts fournies par le bloc FDD, le nouveau contrôleur est synthétisé en ligne.
- **La commande adaptative :** Elle est utilisée pour l'ajustement automatique des gains de commande pour maintenir les performances en boucles fermées.

Les différentes stratégies FTC actives :

- Méthode de pseudo inverse (Pseudo-inverse method, PIM).
- Placement de structure propre (Eigenstructure assignment, EA).
- Poursuite de modèle (Model following, MF).
- Séquencement de gain (Gain-scheduling, GS).
- Commande pour les systèmes Linéaires à Paramètres Variants (Linear parameter varying control, LPV control).
- Multi-modèles (Multiple Model, MM).
- Commande prédictive à base de modèle (Model predictive control, MPC).
- Diagnostic et commande intégrées (Integrated diagnostics and control, IDC).
- Commande intelligente (Intelligent control, IC).
- Linéarisation par bouclage (Feedback linearisation, FL).
- Commande à structure variable/Commande par mode glissant (Variable structure control/Sliding mode control, VSC/SMC).
- Backstepping /Commande par surface dynamique /Backstepping filtré (Backstepping/ Dynamic surface control /Commande filtrée backstepping, BS /DSC /CFB).

- **AFTC adaptative**

Dans cette approche de commande, il existe deux méthodes : adaptation *directe* et *indirecte*. Dans l'approche *directe*, le régulateur est conçu sans estimer les paramètres du système. Tandis que dans l'approche *indirecte*, la conception du régulateur passe par deux étapes. Tout d'abord, en estimant les paramètres du système. Ensuite, en utilisant ces informations pour synthétiser le régulateur. Grâce à leur capacité à s'adapter automatiquement aux changements des paramètres du système, les approches FTC adaptative peuvent être appelées "auto-reconfigurables", où lors de la conception d'un système FTC adaptatif, l'existence des composants "FDD" et "mécanisme de reconfiguration" n'est pas forcément nécessaire.

## B.6 Paramètres utilisés :

Les paramètres physiques utilisés pour les simulations du quadri-rotor sont résumés dans le tableau suivant :

Paramètre	Symbole	Valeur	Unité
Masse du quadrirotor	$m$	0.650	$kg$
Distance entre le centre d'un moteur et le centre de gravité	$l$	0.230	$m$
Moment d'inertie du quadrirotor par rapport à son axe 'X'	$J_x$	$7.7 \times 10^{-3}$	$kg.m^2$
Moment d'inertie du quadrirotor par rapport à son axe 'Y'	$J_y$	$7.7 \times 10^{-3}$	$kg.m^2$
Moment d'inertie du quadrirotor par rapport à son axe 'Z'	$J_z$	$1.3 \times 10^{-2}$	$kg.m^2$
Moment d'inertie du rotor par rapport à son axe 'Z'	$J_r$	$6 \times 10^{-6}$	$kg.m^2$
Coefficient de portance	$b$	$3.13 \times 10^{-5}$	$N.s^2$
Coefficient de traînée	$k$	$7.5 \times 10^{-5}$	$kg.m^2$
Constante de gravité	$g$	9.81	$m.s^{-2}$

Table B.1 : Les paramètres physiques utilisés pour le système.

Les paramètres utilisés dans la commande  $\mathcal{L}_1$  adaptative par la descente du gradient synthétisée au troisième chapitre sont résumés dans le tableau suivant :

Symbole	Valeur
Q	$100 \times I_{12}$
R	$1000 \times I_4$
$\Gamma$	8000

$K_d$	$200 \times I_4$
-------	------------------

Table B.2 : Les paramètres de commande  $\mathcal{L}_1$  DG.

Les paramètres utilisés dans la commande  $\mathcal{L}_1$  adaptative par mode glissant synthétisée au troisième chapitre sont résumés dans le tableau suivant :

Symbole	Valeur
Q	$1000 \times I_{12}$
R	$200 \times I_4$
$\Gamma$	1000
$K_d$	$200 \times I_4$
$\lambda$	$0.1 \times [I_4 \ 0_{4 \times 8}]$
$\varepsilon_u$	0.09

Table B.3 : Les paramètres de commande  $\mathcal{L}_1$  MG.

Les paramètres utilisés dans la commande FTC capteur synthétisée au quatrième chapitre sont résumés dans le tableau suivant :

Symbole	Valeur
$Q_s$	$900 \times I_4$
$R_s$	$10 \times I_4$
$\Gamma_s$	100
$K_{ds}$	$100 \times I_4$
$\lambda_s$	$0.1 \times I_4$
$\varepsilon_s$	0.09
$T_s$	0.001

Table B.4 : Les paramètres de commande FTC capteur.