

وزارة التعليم العالي والبحث العلمي

MINISTERE DE L'ENSEIGNEMENT SUPERIEUR ET DE LA RECHERCHE SCIENTIFIQUE

Université Mouloud Mammeri de Tizi-Ouzou
FACULTE DU GENIE ELECTRIQUE ET DE
L'INFORMATIQUE
DEPARTEMENT D'AUTOMATIQUE



وزو جامعة مولود معمري - تيزي

Mémoire en vue de l'obtention
Du diplôme de master II

DOMAINE : Sciences et Technologies

FILIERE : Génie électrique

SPECIALITE : Commande des systèmes

Titre

**Modélisation et Commande Linéaire et non Linéaires d'un
Drone a Voleur Tournante Type : Quadri-rotor**

Présenté par :
LARBI MEZIANE
MEBREK SAMIR

Encadré par :
M Mohand Achour TOUAT

Jury de soutenance :

M DJENOUNE S.
Professeur, UMMTO, Président
M Maidi A.
Professeur, UMMTO, Examineur
Mme Yacine.
MC, UMMTO, Examineur
M TOUAT M A.
MC B, UMMTO, Encadreur

Promotion : 2016/2017



Remerciements

*Au terme de ce mémoire, nous remercions tout d'abord
D.F.E.U., le clément, le miséricordieux, le tout puissant qui
nous a donné durant toutes ces années la santé, le courage et la
foi en nous même pour arriver à ce jour.*

*Nous exprimons nos sincères remerciements à l'égard de
notre promoteur M. Touat pour son aide et ses
encouragements précieux, durant les travaux pour la mise en
œuvre de ce modeste travail*

*Que les membres de jury trouvent nos remerciements les plus
vifs pour avoir accepté d'honorer par leur jugement notre
travail.*

*Enfin, nous remercions toutes les personnes ayant
contribué d'une façon ou d'une autre à réaliser ce projet de fin
d'études.*

Dédicaces

Je dédie ce modeste travail :
A mes très chers parents qui
Ont su être toujours présent à
Mes cotés, tout au long de ma
vie pour me soutenir.

A mes très chers frères.

Et à mes deux très chères
Sœurs Hayat, Fazia que j'aime
Beaucoup.

A mes amis que j'admire
Beaucoup.

A mon binôme ainsi toute sa
Famille.

Et à tous ceux qui se
Reconnaîtront en ce mot
« AMI »...

MEZIANE.

Dédicaces

Je dédie ce modeste travail en premier titre :

*À la mémoire de ma grande mère "DJOUHER". Que Dieu l'accueille
dans son vaste paradis.*

*À mon grand père. Que Dieu vous bénisse et soit avec vous tout au long de
votre vie.*

*Nulle dédicace ne saurait exprimer mes sentiments , mes reconnaissances et mon
profond attachement , pour tout l'amour que vous m'avez donné, pour tous les
sacrifices que vous avez consentis pour moi.*

*En témoignage de mon amour éternel et ma gratitude infinie, je dédie ce mémoire
À mon exceptionnelle mère.
À mon extraordinaire père .*

Que dieu vous prête une bonne santé et longue vie .

*Je dédie aussi ce mémoire ,
À mon adorable et unique frère Ghiles
À ma chère bien aimée Faiza.*

Que dieu vous protège pour moi et pour ceux qui vous aiment.

*Je dédie aussi ce travail ,
À toutes les personnes qui m'ont toujours aimé, supporté, concilié, poussé à
m'améliorer, dont le mérite, les sacrifices et les qualités humaines, ont fait de moi
ce que je suis aujourd'hui.*

*À mes oncles et mes tantes et leurs enfants ;
À mes cousins et mes cousines et leurs enfants ;
À mes amis et mes camarades ;*

Et à toute personne qui me connaisse de loi ou de près.

Samir

Table des matières

Table des matières	I
Liste des figures.....	V
Liste des tableaux	VII
Liste des symboles	VIII
Introduction générale	1
Chapitre I: Modélisation d'un quadri-rotor à six degrés de liberté.....	3
1. Introduction:	3
2. Présentation et principe du vol d'un quadri-rotor:	3
3. Sens de rotation des rotors:	4
4. Mouvements possibles:.....	4
4.1 Mouvement vertical :.....	4
4.2 Mouvement de roulis :.....	5
4.3 Mouvement de tangage :	5
4.4 Mouvement de lacet :	6
4.5 Mouvements de translation:	6
5. Mode de vol:.....	6
5.1 Vol vertical:.....	6
5.2 Vol Stationnaire	7
5.3 Vol de translation:.....	7
6. Modèle mathématique de la dynamique de vol d'un quadri-rotor:	7
6.1 Hypothèses du modèle :	7
6.2 Correspondance forces/moments – vitesses de moteur :.....	11
6.3 La dynamique du rotor:	12
7. Les contraintes non holonomie :	14
8. Commande des drones :.....	15

8.1 Les techniques de commande linéaires :	15
8.2 Les techniques de commande non linéaires :	15
9. Conclusion:	15
Chapitre II: Commande par PID.....	16
1. Introduction:	16
2. Principe :.....	16
2.1 Action proportionnelle :	16
2.2 Action intégrale:	17
2.3 Action dérivée:	17
2.4 Action proportionnelle-intégrale-dérivée (PID) :	17
3. Optimisation :.....	18
3.1 Optimisation sans contraintes:	18
3.2 Choix des paramètres initiaux :.....	19
4. Résultat de simulation :	20
5. Conclusion :.....	25
Chapitre III: Commande par retour d'état linéarisant.....	26
1. Introduction :	26
2. Géométrie différentielle :	26
2.1 Outils mathématiques :	26
2.2 Opérateurs de Lie (dérivée et crochet de Lie)	26
3. Difféomorphisme :	27
4. Commande par linéarisation entrée-sortie (E/S):.....	27
4.1 Système SISO :	28
4.2 Système (MIMO) :	31
4.3 Le contrôle externe :	34
4.3.1 Notion de pôles	34
5. Application de la technique de linéarisation par bouclage au quadri-rotor :	35
5.1 Stratégie de commande :	35

5.2 Sous-système d'orientation :	36
5.3 Sous système de translation :	39
6. Résultats de simulation :	43
6.1 Interprétations sur les résultats :	47
7. Conclusion :	48
Chapitre IV: Commande PDC basée sur les modèles Takagi-Sugeno	48
1. Introduction	48
2. Définition des modèles flous TS:	48
3. Construction des modèles flous T-S :	50
3.1 Identification des modèles TS :	50
3.2 Linéarisation d'un modèle non linéaire :	50
3.3 Décomposition en secteurs non linéaires :	51
4. Exemple :	52
5. Stabilité et Stabilisation et suivi de trajectoire des modèles flous T-S :	55
5.1 Théorème de Lyapunov :	55
5.2 L'approche de la Compensation Parallèle Distribuée (PDC)	56
5.3 Suivi de trajectoire des modèles flous TS :	59
5.4 Application de la technique PDC au quadri-rotor :	63
6. Résultats de simulation :	65
6.1 Interprétation des résultats :	69
7. Comparaison entre les différentes commandes étudiées:	69
8. Conclusion :	69
Chapitre V: Réalisation du quadri-rotor	70
1. Introduction:	70
2. Les éléments indispensables :	70
2.1 Composants mécaniques	70
2.2 Composants électriques	71
2.3 Capteurs	74

2.4 La carte de programmation	76
3. matériels utilisés (configuration matérielle):.....	77
3.1 Châssis:.....	77
3.2 Hélices:	77
3.3 Moteurs et ECS:	78
3.4 Batterie/Alimentation:	79
3.5 La carte de commande (processeur):	79
3.6 La centrale inertielle:	80
3.7 Capteur Ultrason:	81
3.8 La station de base:	82
3.9 Communication avec le quadri-rotor.....	82
4. Montage du quadri-rotor:	82
4.1 Mise en place du contrôleur de vol openpilot	82
4.2 Remplacement du contrôleur avec la carte Arduino	83
Conclusion générale et perspective	85
Bibliographie	87

Liste des figures

Figure I.1 Quad type + (à gauche); Quad type x (à droite) [1]	3
Figure I.2 Sens de rotation des rotors	4
Figure I.3 Le mouvement vertical	5
Figure I.4 Le mouvement de roulis	5
Figure I.5 Le mouvement de tangage	5
Figure I.6 Le mouvement de lacet	6
Figure I.7 Le repère du quadri-rotor	7
Figure I.8- Rotation suivant le roulis, le tangage et le lacet.	9
Figure I.9- Rotation angulaire, et translation linéaire.	14
Figure II.1 Structure de commande par PID	16
Figure II.2 Schéma synoptique de commande par PID	18
Figure II.3 Résultats de poursuite, dans le cas des trajectoires de références en échelon, selon les axes (x, y, z, ψ)	21
Figure II.4 Trajectoire globale en 3D	21
Figure II.5 L'angle de tangage et de roulis	22
Figure II.6 Les erreurs de déplacement selon les axes (x, y, z, ψ)	22
Figure II.7 Signaux de commande de la rotation	23
Figure II.8 Signaux de commande de la translation	24
Figure III.1 Linéarisation exacte par retour d'état (Forme canonique de Brunovsky).....	31
Figure III.2 Synoptique de linéarisation entrée/sortie des systèmes MIMO	33
Figure III.3 Dynamique de sorties y_i	34
Figure III.4 Schéma bloc du système linéarisé en boucle fermée.....	35
Figure III.5 Schéma synoptique de commande	35
Figure III.6 Résultats de poursuite, dans le cas des trajectoires de références en échelon, selon les axes (X, Y, Z, ψ)	44
Figure III.7 Trajectoire globale du quadri-rotor en 3D.....	44
Figure III.8 les erreur de déplacement selon les 3 axes	45
Figure III.9 L'angle de tangage et de roulis	45
Figure III.10 signaux de commande des rotations	46
Figure III.11 signaux de commande des translations.....	47
Figure IV.1 Secteur global.....	51
Figure IV.2 Secteur local	52

Figure IV.3 Structure et validation d'un modèle T-S	52
Figure IV.4 Simulation du système (NL) et (TS)	54
Figure IV.5 Principe des régulateurs flous PDC	57
Figure IV.6 a stabilisation du l'état $\mathbf{x1}(t)$ et $\mathbf{x2}(t)$	59
Figure IV.7 Réponse de l'état $\mathbf{x1}(t)$ et $\mathbf{x2}(t)$	63
Figure IV.8 Schéma synoptique de commande	63
Figure IV.9 Résultats de poursuite, dans le cas des trajectoires de références en échelon, selon les axes (X,Y,Z, ψ)	66
Figure IV.10 Trajectoire globale en 3D	66
Figure IV.11 les erreurs de déplacement selon les 3 axes	67
Figure IV.12 L'angle de tangage et de roulis	67
Figure IV.13 signaux de commande des rotations	68
Figure IV.14 signaux de commande de translation	68
Figure V.1 Caractéristiques de l'hélice	71
Figure V.2 Emax MT1806 2280kv	71
Figure V.3 Contrôleur électronique de vitesse Robo India ESSC30A	72
Figure V.4 Système de distribution de l'alimentation [36]	73
Figure V.5 Centrale inertielle associant les circuits MEMS	74
Figure V.6 Capteur Ultrason HC-SR04	75
Figure V.7 Capteur de pression BMP180	75
Figure V.8 Module GPS GYGPSV5-NEOM8N	75
Figure V.9 Module NRF24 L01	76
Figure V.10 Comosition du châssis WMC Alien X-500	77
Figure V.11 Les 4 hélices du quadri-rotor	78
Figure V.12 Moteur readytosky 920kv	78
Figure V.13 ESC Simonk 30A	78
Figure V.14 Batterie Lithium-polymère 3S	79
Figure V.15 Caractéristiques de la batterie	79
Figure V.16 Arduino Due	80
Figure V.17 Module MPU6050	81
Figure V.18 Capteur ultrason HC-SR04	82
Figure V.19 Capture d'écran du logiciel openpilot	83
Figure V.20 Liaison du quadri-rotor au pc via le port USB	83
Figure V.21Schéma de montage du quadri-rotor	84

Liste des tableaux

Tableau II.1 Paramètres initiaux du pid.....	19
Tableau II.2 Paramètres du pid après simulation.....	20
Tableau V.1 Choix des moteurs d'un quadri-rotor.....	72
Tableau V.2 Caractéristiques du châssis WMC Alien X-500.....	77
Tableau V.3 Caractéristique des ESC Simonk 30A.....	79
Tableau V.4 Caractéristique de la carte Arduino Due.....	80
Tableau V.5 Caractéristique du module MPU6050.....	81
Tableau V.6 Caractéristique du capteur ultrason HC-SR04.....	82

Liste des symboles

SYMBOLE	SIGNIFICATION	UNITE
x, y, z	Coordonnées cartésiennes	m
$\dot{x}, \dot{y}, \dot{z}$	Vitesse de translation	m/s^2
$\ddot{x}, \ddot{y}, \ddot{z}$	Accélération de translation	m/sec^2
φ, θ, ψ	Angle d'Euler (Roulis, Tangage, Lacet)	rad
$\dot{\varphi}, \dot{\theta}, \dot{\psi}$	Vitesse angulaire	rad/sec
$\ddot{\varphi}, \ddot{\theta}, \ddot{\psi}$	Accélération angulaire	rad/sec^2
ξ	Vecteur position	-----
v	Vecteur des vitesses linéaire	-----
m	Masse de quadri-rotor	Kg
d	Distance entre le centre de gravité du quadri-rotor et l'axe de rotation des hélices	m
g	Accélération de la pesanteur	m/sec^2
w_i	Vitesse angulaire de rotor i	rad/sec
F_i	Force développer par rotor i	N
Γ_i	Moment applique à la structure de quadri-rotor	$N.m$
K_p	Coefficient de portance	$N.sec^2/rad^2$
K_{tx}, K_{ty}, K_{tz}	Coefficients de trainée selon 3 axes	$N.m.sec^2/rad^2$
K_{ax}, K_{ay}, K_{az}	Coefficients frottement aérodynamique	-----
J_r	L'inertie de rotor	$Kg.m^2$
Ω	Vitesse angulaire exprimer aux repères lie au centre quadri-rotor	$Kg.m^2$

Introduction générale

Depuis les années 60 les véhicules aériens sans pilotes (Unmanned Aerial Véhicule) ont été conçus dans le domaine militaire, dont l'objectif était de remplacer l'être humain lorsque la tâche à réaliser devenait pénible ou répétitive et surtout lorsqu'il s'agissait d'un environnement hostile où la sécurité du pilote n'était plus assurée. Les premières recherches dans le domaine des UAV datent de la seconde guerre mondiale, les premiers appareils conçus avaient des dynamiques et des dimensions proches de celles des avions et ils volaient à une très haute altitude.

Aujourd'hui, avec l'augmentation de la puissance des calculateurs et la miniaturisation des capteurs et les actionneurs, plusieurs applications civiles ont investi dans la conception des UAV de différentes tailles. L'évolution de la commande automatique a ouvert la voie à plusieurs applications comme la surveillance des grandes infrastructures telles que les lignes haute tension, les barrages et les ouvrages d'art. Dans l'environnement, ces UAV peuvent être très utiles pour la détection des zones contaminées avec des agents radioactifs, chimiques ou biologiques, la mesure de la pollution de l'air, la surveillance des forêts..., etc.

La recherche dans le domaine des drones fait appel à plusieurs disciplines scientifiques à savoir l'aérodynamique, la mécanique, l'électronique, l'automatique, la communication..., etc.

Le travail présenté dans ce mémoire s'inscrit dans le cadre de la modélisation et le développement des lois de commandes linéaire et non linéaire pour un drone type quadri-rotor. Le but attendu est d'abord de déterminer un modèle compatible décrivant la dynamique de ce dernier et d'arriver enfin à concevoir des commandes linéaire et non linéaire réalisables, qui permettent au drone quadri-rotor la poursuite d'une trajectoire de référence désirée.

Ce mémoire s'articule autour de cinq chapitres :

Le premier chapitre présentera une modélisation mathématique du quadri-rotor, qui consistera en une description de ce système et de son principe de vol, puis sous certaines hypothèses simplificatrices, on va aboutir à un modèle dynamique basé sur le formalisme de Newton-Euler valables pour le cas de petits angles pour enfin proposer une représentation d'état qui est utilisée par la suite dans les lois de commande.

Le deuxième chapitre exposera une technique de commande linéaire par PID et une optimisation des paramètres de cette dernière par minimisation de l'erreur quadratique.

Le troisième chapitre, est consacré à la linéarisation du modèle du quadri-rotor en utilisant la linéarisation par bouclage (feedback linéarisation) [13]. Cette technique est basée sur le contrôle interne et externe. Le contrôle interne du modèle du quadri-rotor est réalisé par bouclage statique

(commande par placement de pôles). Cette commande non linéaire va être appliquée au drone à quatre hélice afin d'assurer la poursuite de trajectoire.

Dans le quatrième chapitre, L'approche de modélisation floue basée sur les modèles de type Takagi-Sugeno (T-S) à partir des équations d'état qui décrivent la dynamique du quadri-rotor est présentée, qui permet donc de représenter un système non linéaire par une interconnexion de modèles affines et linéaires autour de différents points de fonctionnement à travers des fonctions de pondération normalisées appelées fonctions d'activation. Ces fonctions peuvent être soit identifiées à partir d'un processus réel [6], soit sur la base d'un modèle de connaissance afin d'aboutir à une représentation exacte de celui-ci dans un espace compact de l'espace d'état [7], [6]. Par la suite, on aborde quelques concepts généraux sur la stabilité et la stabilisation des modèles TS par la technique PDC ainsi que le problème de suivi de trajectoire.

Dans le cinquième chapitre, nous présentons le matériel utilisé pour la réalisation ainsi que les démarches prises pour la mise en marche du quadri-rotor.

Enfin, la conclusion générale présente le bilan de ce travail et les perspectives envisagées.

Chapitre I: Modélisation d'un quadri-rotor à six degrés de liberté

Chapitre I: Modélisation d'un quadri-rotor à six degrés de liberté

1. Introduction:

Afin de concevoir un contrôleur de vol, on doit d'abord comprendre profondément les mouvements de l'avion, sa dynamique et par conséquent ses équations dynamiques. Cette compréhension est nécessaire non simplement pour la conception du contrôleur, mais aussi pour s'assurer que les simulations du comportement de véhicule sont plus proche que possible de la réalité lorsque la commande est appliquée.

En utilisant le formalisme de Newton, nous présentons, dans ce chapitre, la description et la modélisation dynamique d'un hélicoptère de type quadri-rotor à six degré de liberté ainsi que ses mouvements de base.

2. Présentation et principe du vol d'un quadri-rotor:

Le quadri-rotor est un aéronef soulevé et propulsé par quatre rotors, comme son nom l'indique, il se compose de quatre moteurs situés aux extrémités d'une armature qui n'est autre que deux axes en croix ou en plus, comme le montre la Figure I.1.

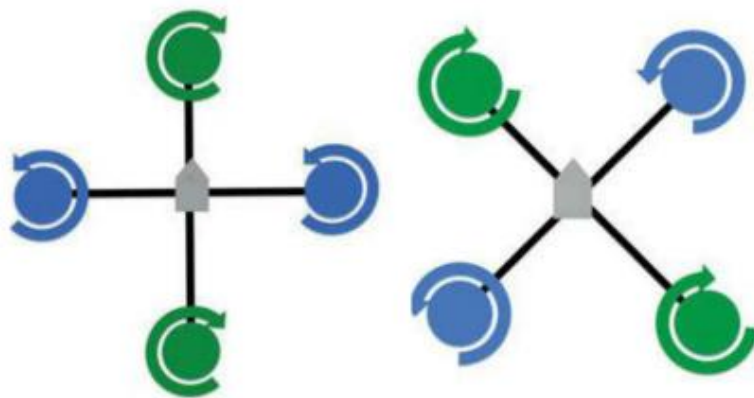


Figure I.1 Quad type + (à gauche); Quad type x (à droite) [1]

Son mouvement est contrôlé par la variation de vitesse de chaque rotor pour changer la force de portance et le couple créé par chacun.

Le fonctionnement d'un quadri-rotor est assez particulier. En effet, comme son nom l'indique, c'est un drone muni de quatre rotors définis dans l'espace par 6 DDL [2]. Le mouvement vertical de montée et de descente est assuré par les quatre rotors à la fois. La différence de poussée entre les moteurs avant et arrière produit un couple de tangage qui contrôle le mouvement de translation (avant/arrière). Le déplacement gauche/droite, quant à lui, est assuré par la différence de poussée entre les rotors situés sur la gauche et sur la droite.

Enfin pour le contrôle en lacet du système, c'est la somme des anti-couples produits par les quatre rotors qui définit le sens de rotation. La conception du quadri-rotor offre de réels avantages par rapport à d'autres configurations d'hélicoptères et ouvre de nombreuses perspectives:

- La taille réduite et la manœuvrabilité qui leur permettent de voler dans des environnements fermés (Indoor) ou ouvert (Outdoor) et près des obstacles à l'opposition des hélicoptères conventionnels.
- La simplicité de la mécanique du quadri-rotor facilite sa maintenance.
- Quatre petits rotors remplacent le grand rotor de l'hélicoptère ce qui réduit énormément l'énergie cinétique stockée et minimise les dégâts en cas d'accidents [3].

3. Sens de rotation des rotors:

Afin d'éviter un mouvement de lacet (l'appareil tourne sur lui-même), il est nécessaire que deux rotors tournent dans un sens et que les deux autres tournent dans l'autre sens. En plus, pour diriger l'appareil, les rotors tournant dans le même sens doivent être placés l'un en face de l'autre (Figure I.2). De plus, pour faciliter la gestion des commandes et les calculs, le rotor numéro 1 est choisi comme l'avant du quadri-rotor.

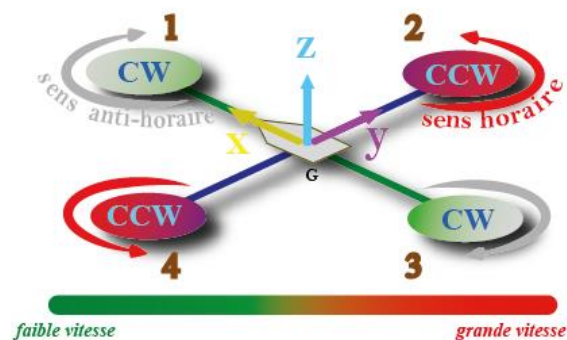


Figure I.2 Sens de rotation des rotors

4. Mouvements possibles:

4.1 Mouvement vertical :

Le mouvement vertical (Sustentation) s'obtient de la contribution des quatre hélices au même temps comme la montre la Figure I.3.

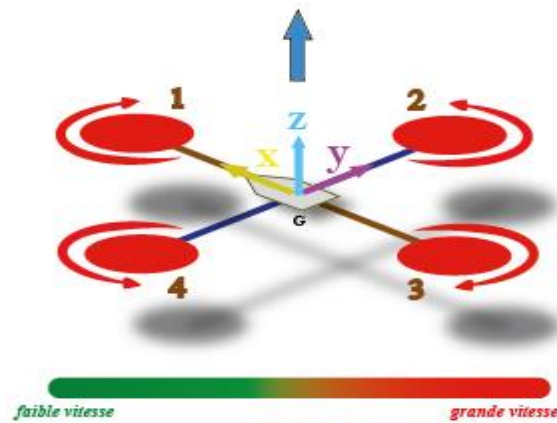


Figure I.3 Le mouvement vertical

4.2 Mouvement de roulis :

Le déplacement suivant l'axe y se produit suite à une rotation autour de l'axe x cette dernière se crée à cause de la différence de portance des rotors 2-4 (Roulis ϕ).

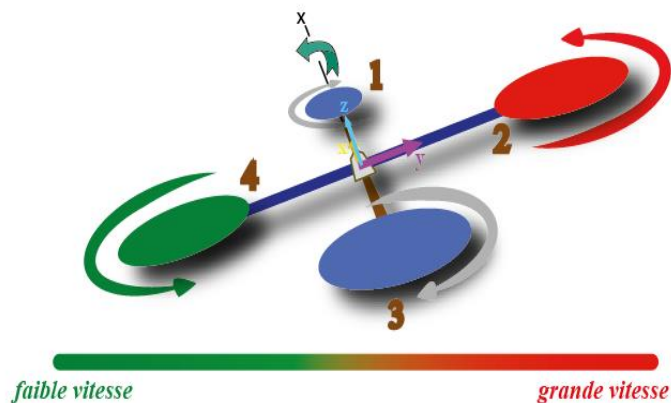


Figure I.4 Le mouvement de roulis

4.3 Mouvement de tangage :

Le déplacement suivant l'axe x se produit suite à une rotation autour de l'axe y , cette dernière se crée à cause de la différence de portance des rotors 1-3 (Tangage θ).

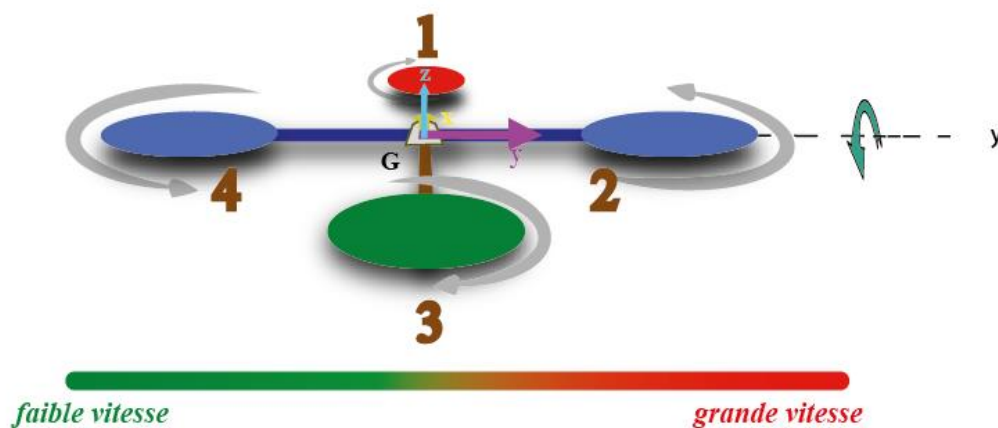


Figure I.5 Le mouvement de tangage

4.4 Mouvement de lacet :

Le mouvement en lacet nécessite que deux rotors du même axe tournent dans un sens tandis que les deux autres dans l'autre sens (Lacet ψ).

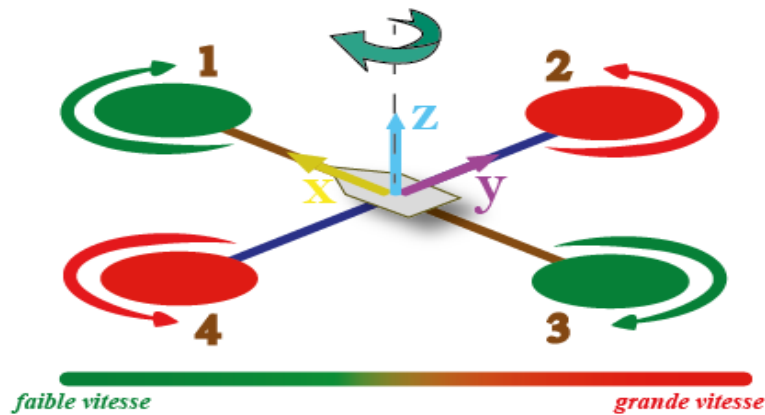


Figure I.6 Le mouvement de lacet

4.5 Mouvements de translation:

4.5.1 Translations sur l'axe X :

Pour réaliser une translation sur l'axe X, nous devons réaliser une rotation autour de l'axe Y comme expliqué précédemment tout en respectant les conditions nécessaires pour ne pas avoir de rotation de Lacet.

4.5.2 Translations sur l'axe Y :

Pour réaliser une translation sur l'axe Y, nous devons réaliser une rotation autour de l'axe X tout en respectant les conditions citées précédemment pour ne pas avoir de rotation de Lacet.

5. Mode de vol:

En se basant sur les mouvements possibles cités précédemment, le quadri-rotor peut effectuer trois modes de vol :

- Vol vertical
- Vol stationnaire
- Vol de translation

5.1 Vol vertical:

Dans le vol vertical, la résultante aérodynamique et le poids total sont deux forces ayant la même direction mais de sens opposé [4]. Le quadri-rotor peut monter ou descendre, suivant l'effet aérodynamique soit supérieur ou inférieur au poids de l'appareil.

5.2 Vol Stationnaire

Quand la force de Portance et celle de pesanteur sont égales et opposées, le quadri rotor reste immobile. On parle de vol stationnaire.

5.3 Vol de translation:

Le vol de translation correspond à la navigation de quadri-rotor sur un plan horizontal. Il est assuré en se basant sur les mouvements d'inclinaison tangage et roulis.

6. Modèle mathématique de la dynamique de vol d'un quadri-rotor:

Pour étudier les mouvements du quadri-rotor, on utilise deux repères (Figure I.7) : le repère fixe $R_e = (O_e, \vec{X}_e, \vec{Y}_e, \vec{Z}_e)$ lié à la terre et supposé galiléen et le repère mobile $R = (O, \vec{x}, \vec{y}, \vec{z})$ lié au corps du quadri-rotor de centre de gravité G.

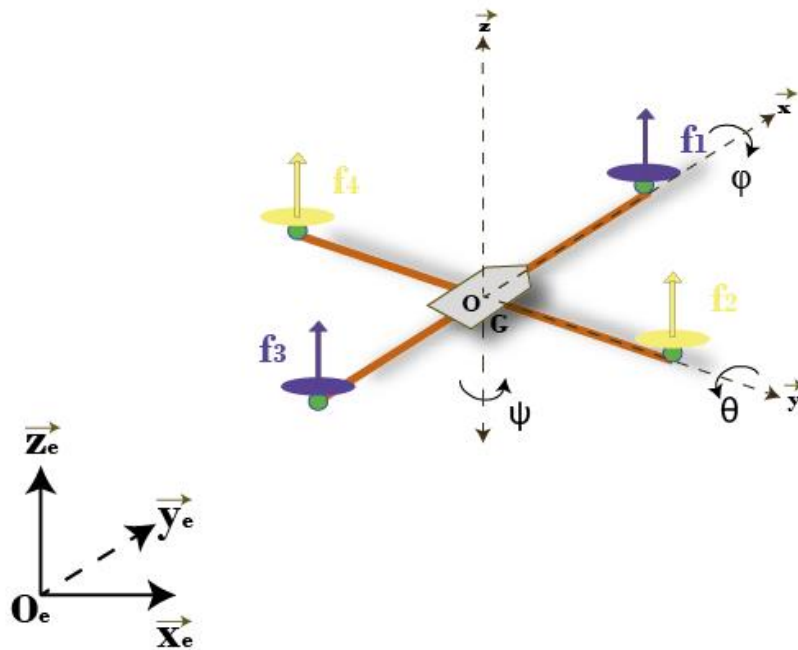


Figure I.7 Le repère du quadri-rotor

Afin de pouvoir comprendre au mieux le modèle dynamique développé ci-dessous, voilà les différentes hypothèses du travail:

6.1 Hypothèses du modèle :

Pour pouvoir déterminer le modèle dynamique, le quadri-rotor est assimilé à une structure rigide et symétrique d'où l'hypothèse que la matrice d'inertie est diagonale. Les hélices sont également supposées rigides pour pouvoir négliger l'effet de leur déformation lors de la portance et la trainée de chaque moteur qui sont proportionnelles au carré de la vitesse de rotation des rotors, ce qui est une approximation très proche du comportement aérodynamique et le repère lié à cette structure est généralement supposé confondu avec son centre de gravité. Cela nous emmène à considérer la dynamique du quadri-rotor comme celle d'un corps solide dans l'espace [5] [6]. Les

conditions atmosphériques sont la condition standard de pression et de température. Sous ces hypothèses, il est possible de décrire la dynamique du quadri-rotor comme celle d'un corps rigide dans l'espace à laquelle viennent s'ajouter les forces aérodynamiques provoquées par la rotation du rotor. En utilisant le formalisme de Newton-Euler, les équations de la dynamique s'écrivent sous la forme suivante [7]:

$$\begin{cases} \dot{\xi} = v \\ m \ddot{\xi} = F_f + F_t + F_g \\ \dot{R} = RS(\Omega) \\ J\dot{\Omega} = -(\Omega \wedge J\Omega) + \Gamma_f - \Gamma_a - \Gamma_g \end{cases} \quad (I-1)$$

Où:

R : La matrice de rotation

S : est la matrice antisymétrique; pour un vecteur de vitesse

ξ: représente la position du centre de masse du quadri-rotor par rapport au repère inertiel $\{R_e\}$.

J: est la matrice d'inertie au centre de masse, exprimé dans le repère mobile $\{R\}$, considérée diagonale, car la structure du quadri-rotor est supposée symétrique.

$$J = \begin{pmatrix} I_x & 0 & 0 \\ 0 & I_y & 0 \\ 0 & 0 & I_z \end{pmatrix} \quad (I-2)$$

Le quadri-rotor est un robot volant à six degrés de liberté, trois translations $(O, \vec{x}, \vec{y}, \vec{z})$ et trois rotations (φ, θ, ψ) . Les translations du quadri-rotor sont les coordonnées cartésiennes $(O_e, \vec{X}_e, \vec{Y}_e, \vec{Z}_e)$ du centre de masse G dans le repère inertiel. Les rotations du quadri-rotor sont décrites par trois rotations consécutives à travers trois angles qui s'appellent les angles d'Euler, et ce, en utilisant la séquence d'axe x - y - z (utilisée par la communauté aéronautique). Les angles s'appellent alors roulis autour de l'axe x, tangage autour de l'axe y et lacet autour de l'axe z. Dans notre travail, cette séquence est utilisée vu qu'elle est la plus adaptée par la plupart de centrales inertielles existantes sur le marché. Pour cet ensemble particulier d'angles d'Euler, l'orientation du système de référence attaché au quadri-rotor $(O, \vec{x}, \vec{y}, \vec{z})$ par rapport au système de référence inertiel lié à la terre $(O_e, \vec{X}_e, \vec{Y}_e, \vec{Z}_e)$ est décrite suivant la combinaison des trois rotations consécutives suivantes (Figure I.8) :

- Une rotation autour de l'axe z d'un angle ψ avec : $-\pi < \psi < \pi$
- Une rotation autour de l'axe y d'un angle θ avec : $-\frac{\pi}{2} < \theta < \frac{\pi}{2}$
- Une rotation autour de l'axe x d'un angle φ avec : $-\frac{\pi}{2} < \varphi < \frac{\pi}{2}$

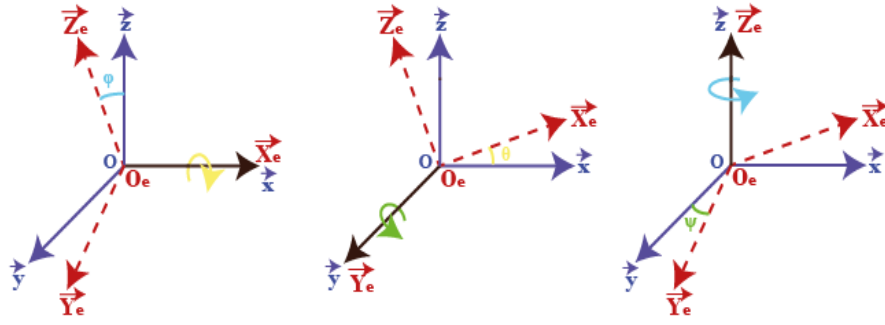
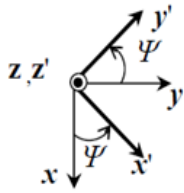


Figure I.8- Rotation suivant le roulis, le tangage et le lacet.

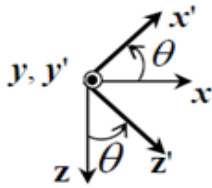
Les matrices relatives à ces transformations sont :

- Rotation autour de l'axe z :



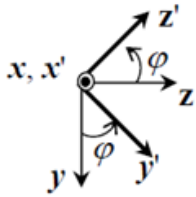
$$Rot(z, \psi) = \begin{bmatrix} \cos(\psi) & -\sin(\psi) & 0 \\ \sin(\psi) & \cos(\psi) & 0 \\ 0 & 0 & 1 \end{bmatrix} \quad (I-3)$$

- Rotation autour de l'axe y :



$$Rot(y, \theta) = \begin{bmatrix} \cos(\theta) & 0 & \sin(\theta) \\ 0 & 1 & 0 \\ -\sin(\theta) & 0 & \cos(\theta) \end{bmatrix} \quad (I-4)$$

- Rotation autour de l'axe x :



$$Rot(x, \varphi) = \begin{bmatrix} 1 & 0 & 0 \\ 0 & \cos(\varphi) & -\sin(\varphi) \\ 0 & \sin(\varphi) & \cos(\varphi) \end{bmatrix} \quad (I-5)$$

On a donc :

La transformation qui permet de passer du repère local vers le repère global est donnée par [5][8]:

$$R(\psi, \theta, \varphi) = Rot(z, \psi) * Rot(y, \theta) * Rot(x, \varphi)$$

$$R(\psi, \theta, \varphi) = \begin{bmatrix} \cos\theta \cos\psi & \cos\psi \sin\theta \sin\varphi - \sin\psi \cos\varphi & \cos\psi \sin\theta \cos\varphi + \sin\psi \sin\varphi \\ \cos\theta \sin\psi & \sin\psi \sin\theta \sin\varphi + \cos\psi \cos\varphi & \sin\psi \sin\theta \cos\varphi - \cos\psi \sin\varphi \\ -\sin\theta & \sin\varphi \cos\theta & \cos\varphi \cos\theta \end{bmatrix} \quad (I-4)$$

✚ $S(\Omega)$ est la matrice antisymétrique; pour un vecteur donné $\Omega = [\Omega_1 \quad \Omega_2 \quad \Omega_3]^T$ elle est définie comme suit:

$$S(\Omega) = \begin{bmatrix} 0 & -\Omega_3 & \Omega_2 \\ \Omega_3 & 0 & -\Omega_1 \\ -\Omega_2 & \Omega_1 & 0 \end{bmatrix} \quad (I-5)$$

✚ F_f est la résultante des forces de poussée générées par les quatre rotors. Elle est donnée par :

$$F_f = \begin{pmatrix} \cos(\varphi)\cos(\psi)\sin(\theta) + \sin(\varphi)\sin(\psi) \\ \cos(\varphi)\sin(\theta)\sin(\psi) - \sin(\varphi)\cos(\psi) \\ \cos(\varphi)\cos(\theta) \end{pmatrix} \sum_{i=1}^4 F_i \quad (I-6)$$

Avec:

$$F_i = K_p \omega_i^2 \quad (I-7)$$

Où K_p désigne le coefficient de portance et ω_i désigne la vitesse angulaire du rotor en question.

✚ F_t est la résultante des forces de trainée selon $(\overrightarrow{X_e}, \overrightarrow{Y_e}, \overrightarrow{Z_e})$. Elle est donnée par:

$$F_t = \begin{pmatrix} -K_{ftx} & 0 & 0 \\ 0 & -K_{fity} & 0 \\ 0 & 0 & -K_{ftz} \end{pmatrix} \dot{\xi} \quad (I-8)$$

Où K_{ftx} , K_{fity} , K_{ftz} sont les coefficients des forces de trainée selon les trois axes.

✚ F_g regroupe les forces liées à la gravité :

$$F_g = \begin{pmatrix} 0 \\ 0 \\ -mg \end{pmatrix} \quad (I-9)$$

✚ Γ_f représente le vecteur résultant des moments appliqués sur la structure du quadri-rotor:

$$\Gamma_f = \begin{bmatrix} d(F_3 - F_1) \\ d(F_4 - F_2) \\ K_d(\omega_1^2 - \omega_2^2 + \omega_3^2 - \omega_4^2) \end{bmatrix} \quad (I-10)$$

Où d est la distance entre le centre de masse du quadri-rotor et l'axe de rotation du rotor et K_d le coefficient de trainée.

✚ Γ_a représente le vecteur résultant des frottements dus aux couples aérodynamique :

$$\Gamma_a = \begin{pmatrix} K_{fax} & 0 & 0 \\ 0 & K_{fay} & 0 \\ 0 & 0 & K_{faz} \end{pmatrix} \Omega^2 \quad (I-11)$$

Où K_{fax} , K_{fay} , K_{faz} sont les coefficients des frottements aérodynamiques selon les trois axes.

✚ Γ_g représente l'ensemble des couples dus aux effets gyroscopiques :

$$\Gamma_g = \sum_{i=1}^4 \Omega \wedge J_r \begin{pmatrix} 0 \\ 0 \\ (-1)^{i+1} \omega_i \end{pmatrix} \quad (I-12)$$

Où J_r et ω_i représentent l'inertie et la vitesse angulaire, respectivement, du rotor en question.

A partir des relations précédentes on obtient le modèle dynamique complet suivant qui régit le quadri rotor:

$$\left\{ \begin{array}{l} \ddot{\phi} = \frac{1}{I_x} [\dot{\theta} \dot{\psi} (I_y - I_z) - K_{fax} \dot{\phi}^2 - J_r \bar{\Omega} \dot{\theta} + dU_2] \\ \ddot{\theta} = \frac{1}{I_y} [\dot{\phi} \dot{\psi} (I_z - I_x) - K_{fay} \dot{\theta}^2 + J_r \bar{\Omega} \dot{\phi} + dU_3] \\ \ddot{\psi} = \frac{1}{I_z} [\dot{\theta} \dot{\phi} (I_x - I_y) - K_{faz} \dot{\psi}^2 + K_d U_4] \\ \ddot{x} = \frac{1}{m} \{ (\cos \phi \sin \theta \cos \psi + \sin \phi \sin \psi) U_1 - K_{f_{tx}} \dot{x} \} \\ \ddot{y} = \frac{1}{m} \{ (\cos \phi \sin \theta \sin \psi - \sin \phi \cos \psi) U_1 - K_{f_{ty}} \dot{y} \} \\ \ddot{z} = \frac{1}{m} \{ (\cos \phi \cos \theta) U_1 - K_{f_{tz}} \dot{z} \} - g \end{array} \right. \quad (I-13)$$

Où U_1 , U_2 , U_3 et U_4 sont les entrées de commande du système.

6.2 Correspondance forces/moments – vitesses de moteur :

En se basant sur les équations (I-6), (I-7) et (I-10), nous pouvons formuler une correspondance permettant de calculer la vitesse des moteurs à partir des forces et des moments appliqués au quadri-rotor.

Cette correspondance est essentielle lors de l'implémentation du contrôleur au niveau de l'appareil. Nous pouvons réécrire les équations sous la forme matricielle suivante :

$$\begin{bmatrix} U_1 \\ U_2 \\ U_3 \\ U_4 \end{bmatrix} = \begin{bmatrix} K_p & K_p & K_p & K_p \\ 0 & -K_p & 0 & K_p \\ -K_p & 0 & K_p & 0 \\ -K_d & K_d & -K_d & K_d \end{bmatrix} \begin{bmatrix} \omega_1^2 \\ \omega_2^2 \\ \omega_3^2 \\ \omega_4^2 \end{bmatrix} \quad (I-14)$$

En inversant cette matrice, nous obtenons la correspondance désirée, notée $\mathbf{M}$. Il est intéressant de noter que le déterminant de celle-ci est égal à $8K_p^3 K_d d^2$ qui est toujours non nul de telle sorte que l'inversion est toujours valide.

$$\begin{bmatrix} \omega_1^2 \\ \omega_2^2 \\ \omega_3^2 \\ \omega_4^2 \end{bmatrix} = \begin{bmatrix} \frac{1}{4 k_p} & 0 & \frac{1}{2 k_p d} & -\frac{1}{4 k_d} \\ \frac{1}{4 k_p} & -\frac{1}{2 k_p d} & 0 & \frac{1}{4 k_d} \\ \frac{1}{4 k_p} & 0 & -\frac{1}{2 k_p d} & -\frac{1}{4 k_d} \\ \frac{1}{4 k_p} & \frac{1}{2 k_p d} & 0 & \frac{1}{4 k_d} \end{bmatrix} \begin{bmatrix} U_1 \\ U_2 \\ U_3 \\ U_4 \end{bmatrix} = \mathbf{M} \quad (\text{I-15})$$

6.3 La dynamique du rotor:

Le rotor est un ensemble constitué d'un moteur à courant continu entrainant une hélice via un réducteur. Le moteur à courant continu est régi par la dynamique suivante:

$$\begin{cases} V = ri + L \frac{di}{dt} + K_e \omega \\ K_m i = J_r \frac{d\omega}{dt} + C_s + K_r \omega^2 \end{cases} \quad (\text{I-16})$$

Les différents paramètres du moteur sont définis comme suit :

V: est la tension d'entrée du moteur.

K_e , K_m : sont des constantes des couples électrique et mécanique respectivement.

K_r : est la constante du couple de charge.

r : désigne la résistance interne du moteur.

J_r : désigne l'inertie du rotor.

C_s : frottement.

En négligeant l'effet inductif des moteurs à cause de leur taille réduite, le modèle dynamique des moteurs est approximé par:

$$\dot{\omega}_i = bV_i - \beta_0 - \beta_1 \omega_i - \beta_2 \omega_i^2 \quad i \in [1,4] \quad (\text{I-17})$$

Avec:

$$\beta_0 = \frac{C_s}{J_r}, \quad \beta_1 = \frac{K_e K_m}{r J_r}, \quad \beta_2 = \frac{K_r}{J_r} \quad \text{et} \quad b = \frac{K_m}{r J_r}$$

Le modèle (I-13) développé précédemment peut être réécrit dans l'espace d'état sous la forme $\dot{X} = f(x) + g(x)U$.

Considérant $X = [x_1 \dots x_{12}]^T$ comme vecteur d'état du système.

Soit:

$$\begin{aligned} X &= [\varphi \quad \dot{\varphi} \quad \theta \quad \dot{\theta} \quad \psi \quad \dot{\psi} \quad x \quad \dot{x} \quad y \quad \dot{y} \quad z \quad \dot{z}]^T \quad i \in [1,4] \quad (\text{I-18}) \\ x_1 &= \varphi & x_7 &= x \\ x_2 &= \dot{x}_1 = \dot{\varphi} & x_8 &= \dot{x}_7 = \dot{x} \\ x_3 &= \theta & x_9 &= y \end{aligned}$$

$$x_4 = \dot{x}_3 = \dot{\theta}$$

$$x_{10} = \dot{x}_9 = \dot{y}$$

$$x_5 = \psi$$

$$x_{11} = z$$

$$x_6 = \dot{x}_5 = \dot{\psi}$$

$$x_{12} = \dot{x}_{11} = \dot{z}$$

Et U comme vecteur de commande :

$$U = [U_1 \quad U_2 \quad U_3 \quad U_4]^T$$

- La commande U_2 (moment de roulis) va orienter le quadrotor a la direction de l'axe y et U_1 va le propulser.
- La commande U_3 (moment de tangage) va orienter le quadrotor a la direction de l'axe x et U_1 va le propulser vers cette orientation.
- La commande U_4 (moment de lacet) fait tourner le quadrotor autour de l'axe z avec un angle ψ .

$$\begin{cases} U_1 = k_p(\omega_1^2 + \omega_2^2 + \omega_3^2 + \omega_4^2) \\ U_2 = k_p(-\omega_2^2 + \omega_4^2) \\ U_3 = k_p(\omega_1^2 - \omega_3^2) \\ U_4 = b(\omega_1^2 - \omega_2^2 + \omega_3^2 - \omega_4^2) \end{cases} \quad (I-19)$$

En se basant sur le modèle (I-13) et l'équation (I-18), on peut écrire :

$$\left\{ \begin{array}{l} \dot{x}_1 = x_2 \\ \dot{x}_2 = a_1 x_2^2 + a_2 x_4 x_6 + a_3 \bar{\Omega} x_4 + b_1 U_2 \\ \dot{x}_3 = x_4 \\ \dot{x}_4 = a_4 x_4^2 + a_5 x_2 x_6 + a_6 \bar{\Omega} x_2 + b_2 U_3 \\ \dot{x}_5 = x_6 \\ \dot{x}_6 = a_7 x_6^2 + a_8 x_2 x_4 + b_3 U_4 \\ \dot{x}_7 = x_8 \\ \dot{x}_8 = a_9 x_8 + U_x \frac{U_1}{m} \\ \dot{x}_9 = x_{10} \\ \dot{x}_{10} = a_{10} x_{10} + U_y \frac{U_1}{m} \\ \dot{x}_{11} = x_{12} \\ \dot{x}_{12} = a_{11} x_{12} + \frac{\cos(x_7) \cos(x_9)}{m} U_1 - g \end{array} \right. \quad (I-20)$$

Avec:

$$\begin{aligned} a_1 &= -\frac{k_{fax}}{I_x}, & a_2 &= +\frac{(I_y - I_z)}{I_x}, & a_3 &= -\frac{J_r}{I_x} \\ a_4 &= -\frac{k_{fay}}{I_y}, & a_5 &= +\frac{(I_z - I_x)}{I_y}, & a_6 &= +\frac{J_r}{I_y} \\ a_7 &= -\frac{k_{faz}}{I_z}, & a_8 &= +\frac{(I_x - I_y)}{I_z}, & a_9 &= -\frac{k_{ftx}}{m} \end{aligned}$$

$$a_{10} = -\frac{k_{fty}}{m}, \quad a_{11} = -\frac{k_{ftz}}{m}$$

$$b_1 = \frac{d}{I_x}, \quad b_2 = \frac{d}{I_y}, \quad b_3 = \frac{1}{I_z}$$

$$u_x = (\cos x_1 \sin x_3 \cos x_5 + \sin x_1 \sin x_5)$$

$$u_y = (\cos x_1 \sin x_3 \sin x_5 - \sin x_1 \cos x_5)$$

On remarque bien, que les angles d'Euler, et leurs dérivées temporelles sont indépendants des composantes de translation, contrairement aux translations qui dépendent des angles d'Euler. On pourra donc diviser notre système, en deux sous-systèmes :

- Sous-système des rotations angulaires.
- Sous-système des Translations linéaires.

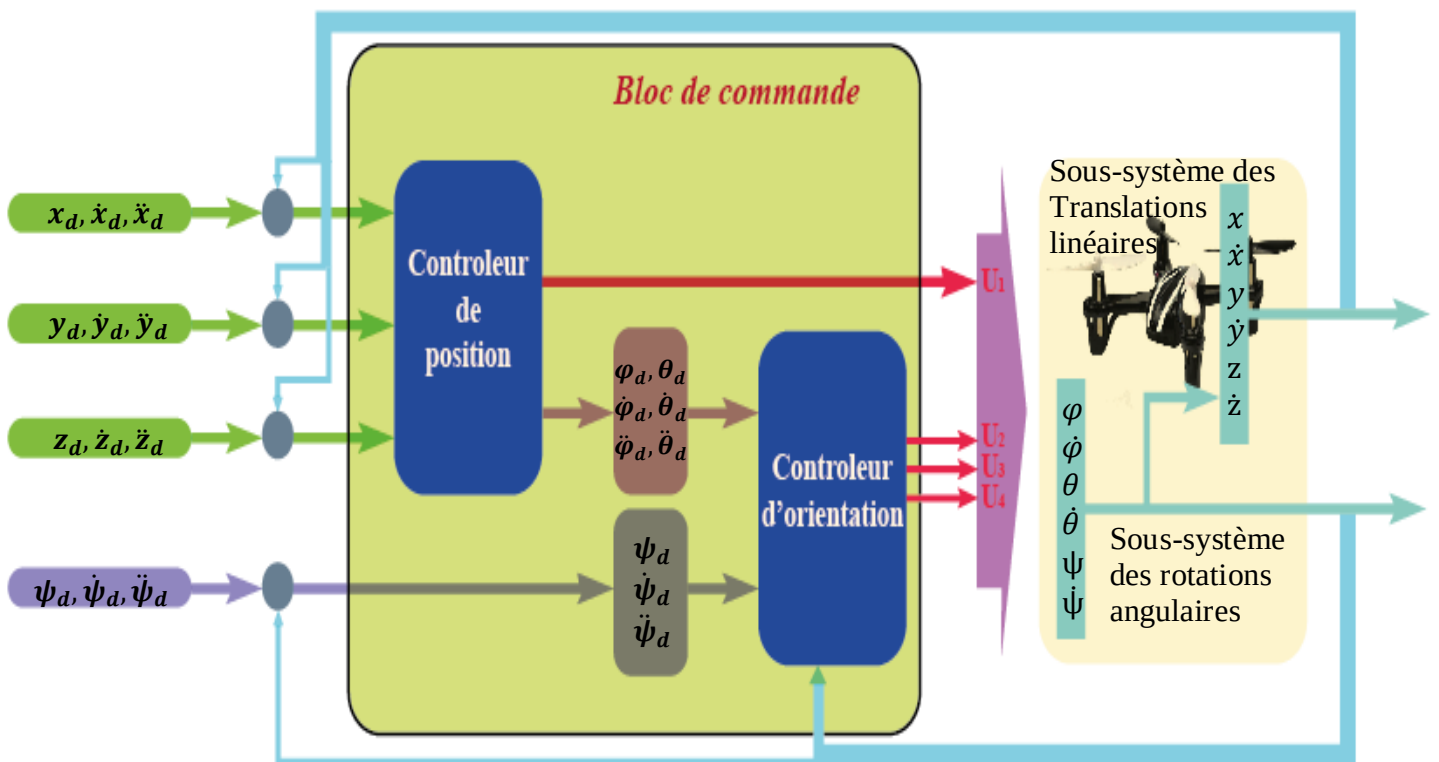


Figure I.9- Rotation angulaire, et translation linéaire.

7. Les contraintes non holonomie :

Le développement des contraintes non holonomes permettent d'explicitier le couplage entre les différents états du système. A partir des équations de la dynamique de translation $\ddot{z}, \ddot{y}$ et $\ddot{x}$ on peut extraire les expressions des contraintes non holonomes comme suit:

$$\begin{cases} \tan \theta = \frac{(\ddot{x}) \cos \psi + (\ddot{y}) \sin \psi}{\ddot{z} + g} \\ \sin \varphi = \frac{-(\ddot{x}) \cos \psi + (\ddot{y}) \sin \psi}{\sqrt{(\ddot{x})^2 + (\ddot{y})^2 + (\ddot{z} + g)^2}} \end{cases} \quad (I-23)$$

D'où:

$$\begin{cases} \theta_d = \arctan\left(\frac{(\ddot{x}_d) \cos \psi + (\ddot{y}_d) \sin \psi}{\ddot{z}_d + g}\right) \\ \varphi_d = \arcsin\left(\frac{-(\ddot{x}_d) \cos \psi + (\ddot{y}_d) \sin \psi}{\sqrt{(\ddot{x}_d)^2 + (\ddot{y}_d)^2 + (\ddot{z}_d + g)^2}}\right) \end{cases} \quad (I-24)$$

8. Commande des drones :

8.1 Les techniques de commande linéaires :

Des approches de commandes linéaires telles que les correcteurs PID peuvent alors être utilisés. Cette démarche a été adoptée dans plusieurs travaux antérieurs. Par exemple dans [9], la commande d'un hélicoptère Yamaha R-50 a été considérée en utilisant pour chacune des boucles (mouvement longitudinal, mouvement latéral, altitude et lacet) un correcteur de type (PID).

L'élaboration des lois de commandes linéaires pour contrôler le mouvement d'un hélicoptère pose de nombreux problèmes car les modèles linéaires simplifiés sont généralement loin de la réalité du système physique. Le modèle dynamique complet d'un hélicoptère engendre en réalité des incertitudes qui constituent des erreurs de la dynamique par rapport au modèle linéaire et, par conséquent, rend l'élaboration de lois de commandes linéaires très difficile. Donc la meilleure solution est de synthétiser des commandes non linéaires basées sur une modélisation complète du système physique.

8.2 Les techniques de commande non linéaires :

Les conceptions de contrôleurs non linéaires sont généralement basées sur la notion de la linéarisation de la boucle fermée du modèle non linéaire [10] [11]. L'idée est de transformer la dynamique non-linéaire en forme linéaire en utilisant le retour d'état.

9. Conclusion:

L'élaboration d'un modèle mathématique de la dynamique du vol d'un quadri-rotor est une tâche difficile. Nous avons procédé à sa modélisation en utilisant le formalisme Newton-Euler, après avoir posé quelques hypothèses simplificatrices. Le modèle développé dans ce chapitre tiens compte de tous les phénomènes agissants sur le quadri-rotor.

Nous avons établi un modèle non linéaire qui montre la nature couplée et complexe de ce système. Nous avons aussi remarqué que le système est sous actionné.

Chapitre II: Commande par PID

Chapitre II: Commande par PID

1. Introduction:

La conception d'un contrôleur de type proportionnel intégral dérivé (PID), exige la spécification des trois paramètres : le gain proportionnel (K_p), le gain intégrale (K_i) et le gain dérivé (K_d). Dans la Littérature, il existe plusieurs méthodes pour déterminer les gains, parmi lesquelles, nous pouvons citer celle de Ziegler Nicols [12]. L'inconvénient major de ces techniques est que les gains du PID sont obtenus pour les systèmes linéaires. Donc ce type de réglage du contrôleur ne convient pas à notre cas. Cela revient au modèle du quadri-rotor qui est non linéaire.

L'objectif de ce chapitre consiste à trouver les paramètres du régulateur PID en minimisant le critère de performance de l'intégrale de l'erreur quadratique et d'asservir selon le schéma de la Figure II.2 le comportement du notre système.

2. Principe :

La stratégie de commande par PID (Proportionnel-Intégral-Dérivé) est certainement l'approche la plus intuitive et la plus simple à embarquer sur un processeur. Elle permet de comprendre facilement le rôle physique de chacun des termes de la commande, ce qui permet de régler les gains en conséquence.

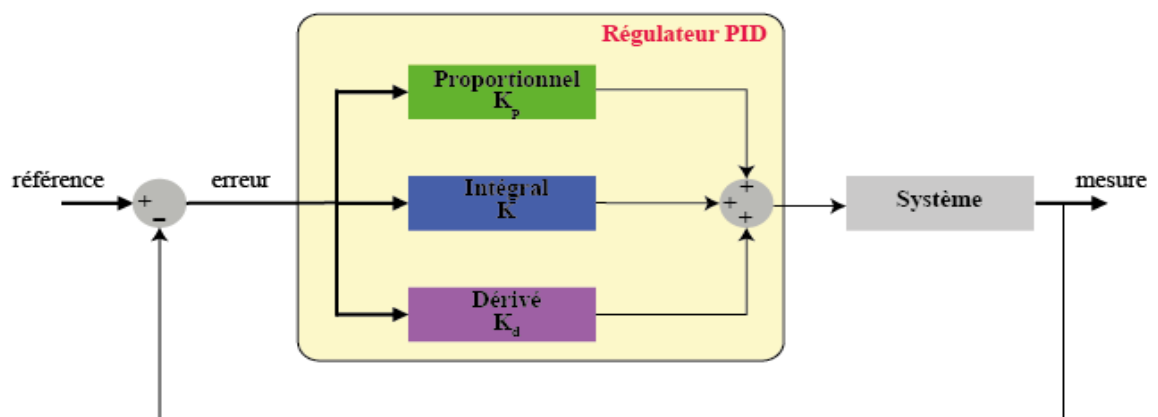


Figure II.1 Structure de commande par PID

2.1 Action proportionnelle :

Le coefficient K_p présente l'avantage de produire un large changement en sortie par rapport à l'erreur, cependant il peut aussi rendre le système instable s'il est trop grand. Cette action proportionnelle est représentée comme suit :

$$C(t) = K_p \cdot \varepsilon(t) \quad (\text{II-1})$$

2.2 Action intégrale:

L'action intégrale a pour but de réduire voire d'éliminer l'erreur statique en régime permanent. Pour réaliser cela, le régulateur intègre l'erreur par rapport au temps et multiplie le résultat par une constante K_i comme suit :

$$C(t) = K_i \cdot \int_0^t \varepsilon(t) \cdot dt \quad (\text{II-2})$$

Plus la valeur de K_i est grande plus l'erreur statique ne sera vite corrigée mais nous perdons un peu en stabilité et il y a un risque de dépassement qui subvient.

2.3 Action dérivée:

Enfin, le gain K_d permet d'atténuer le dépassement et les oscillations du système, ainsi il permet de corriger le défaut du terme intégral, mais il ralentit également le système dans l'atteinte de la consigne sa relation est donnée comme suit :

$$C(t) = K_d \cdot \frac{d\varepsilon(t)}{dt} \quad (\text{II-3})$$

2.4 Action proportionnelle-intégrale-dérivée (PID) :

En vertu de ce qui a été dit précédemment, Un régulateur PID est obtenu par l'association de ces trois actions et il remplit essentiellement les trois fonctions suivantes :

- Fonction proportionnelle donne un système plus précis, plus rapide
- Fonction intégrateur élimine l'erreur statique.
- Fonction dérivée accélère la correction

L'expression d'un régulateur PID est donnée comme suit :

$$U = K_p \cdot e(t) + K_i \cdot \int_0^t e(t) \cdot dt + K_d \cdot \left(\frac{de(t)}{dt} \right) \quad (\text{II-4})$$

Le schéma de la Figure II.2 illustre la stratégie de commande adoptée :

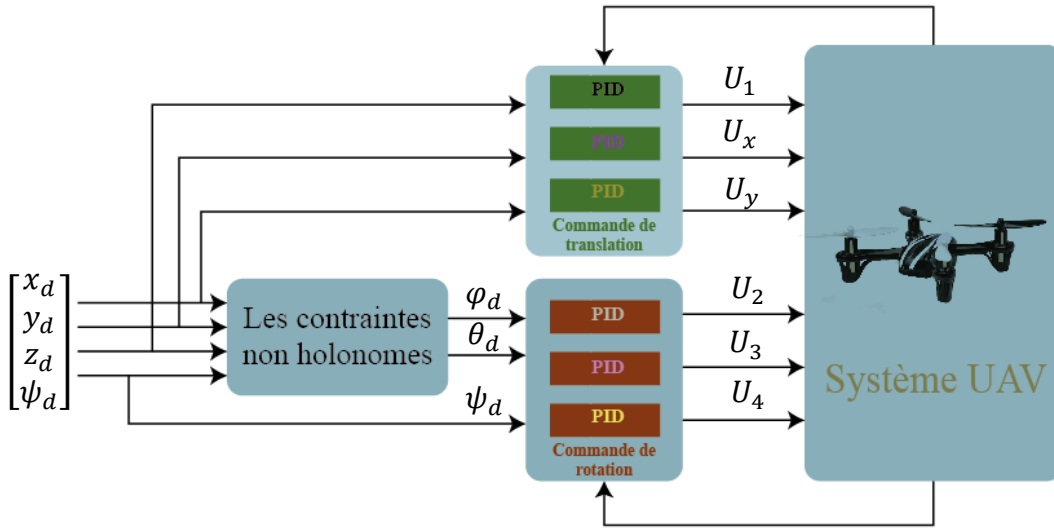


Figure II.2 Schéma synoptique de commande par PID

3. Optimisation :

L'optimisation est une branche des mathématiques cherchant à modéliser, à analyser et à résoudre analytiquement ou numériquement les problèmes qui consistent à minimiser ou maximiser une fonction sur un ensemble.

3.1 Optimisation sans contraintes:

3.1.1 Formulation mathématique:

Un problème d'optimisation est formulé mathématiquement comme suit :

$$\begin{cases} \min J(x) \\ x \in R^n \end{cases} \quad (\text{II-5})$$

$J: R^n \rightarrow R$ est une fonction de plusieurs variables ($x = (x_1, \dots, x_n)$) à valeurs réelles. Cette fonction (que l'on minimise) est appelée indifféremment fonction coût, objectif ou critère.

Maintenant on va réaliser une optimisation des paramètres des régulateurs vu précédemment pour notre système. Nous avons donc posé le problème d'optimisation comme suit:

$$\min_{pid} E(t) = \sum_{i=1}^6 \int_0^t e_i^2(t) \quad (\text{II-6})$$

Avec:

$$pid = [K_{pz}, K_{iz}, K_{dz}, K_{p\varphi}, K_{i\varphi}, K_{d\varphi}, K_{p\theta}, K_{i\theta}, K_{d\theta}, K_{p\psi}, K_{i\psi}, K_{d\psi}, K_{px}, K_{ix}, K_{dx}, K_{py}, K_{iy}, K_{dy}]$$

Et

$e_i(t)$ représente l'erreur des différentes variable $\{Z, \varphi, \theta, \psi, X, Y\}$ tel que :

$$e_i(t) = Y_{di} - Y_i = \{1, 2, 3, 4, 5, 6\}$$

3.2 Choix des paramètres initiaux :

Nous avons déterminé des paramètres initiaux permettant d'assurer une certaine stabilité, ces paramètres sont résumés dans le tableau suivant :

Régulateur	Paramètres
Régulateur Z	$K_{pz} = 3.8$
	$K_{iz} = 1.5$
	$K_{dz} = 4$
Régulateur ψ	$K_{p\psi} = 2$
	$K_{i\psi} = 0.5$
	$K_{d\psi} = 1.9$
Régulateur θ	$K_{p\theta} = 1.8$
	$K_{i\theta} = 0.9$
	$K_{d\theta} = 2$
Régulateur φ	$K_{p\varphi} = 2$
	$K_{i\varphi} = 6$
	$K_{d\varphi} = 0.6$
Régulateur x	$K_{px} = 6$
	$K_{ix} = 0.6$
	$K_{dx} = 3$
Régulateur y	$K_{py} = 6$
	$K_{iy} = 0.6$
	$K_{dy} = 3$

Tableau II.1 Paramètres initiaux du pid

Ainsi les résultats obtenus après la simulation sont résumés dans le tableau suivant :

Régulateur	Paramètres
Régulateur z	$K_{pz} = 124$
	$K_{iz} = 7$
	$K_{dz} = 35$
Régulateur ψ	$K_{p\psi} = 5.3620$
	$K_{i\psi} = 0.1000$
	$K_{d\psi} = 0.3352$
Régulateur θ	$K_{p\theta} = 7.2036$
	$K_{i\theta} = 0.6861$
	$K_{d\theta} = 0.5974$
Régulateur φ	$K_{p\varphi} = 7.1957$
	$K_{i\varphi} = 0.6841$
	$K_{d\varphi} = 0.5974$
Régulateur x	$K_{px} = 2.8594$
	$K_{ix} = 0.01$
	$K_{dx} = 2.2697$
Régulateur y	$K_{py} = 2.8594$
	$K_{iy} = 0.01$
	$K_{dy} = 0.01$

Tableau II.2 Paramètres du pid après simulation

4. Résultat de simulation :

Dans cette simulation nous avons utilisé, pour calculer les différentes lois de commande développées dans la section précédente, l'ensemble des paramètres du modèle du quadri-rotor qui sont donnés dans le tableau II.2. Nous avons également considéré des trajectoires de référence en échelon. Les performances de commande obtenues sont illustrées par les figures (II-3) à (II-8).

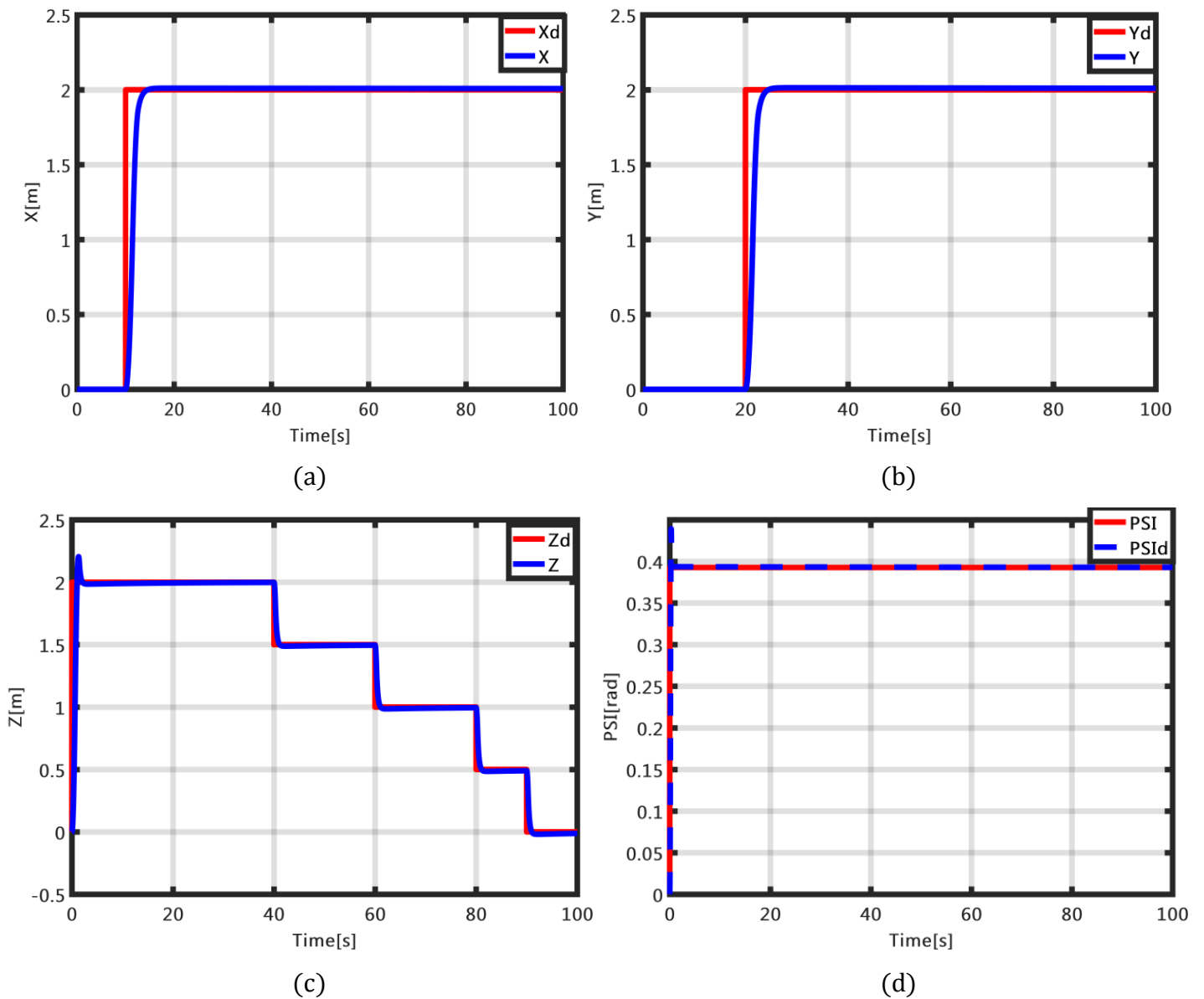


Figure II.3 Résultats de poursuite, dans le cas des trajectoires de références en échelon, selon les axes (x, y, z, ψ)

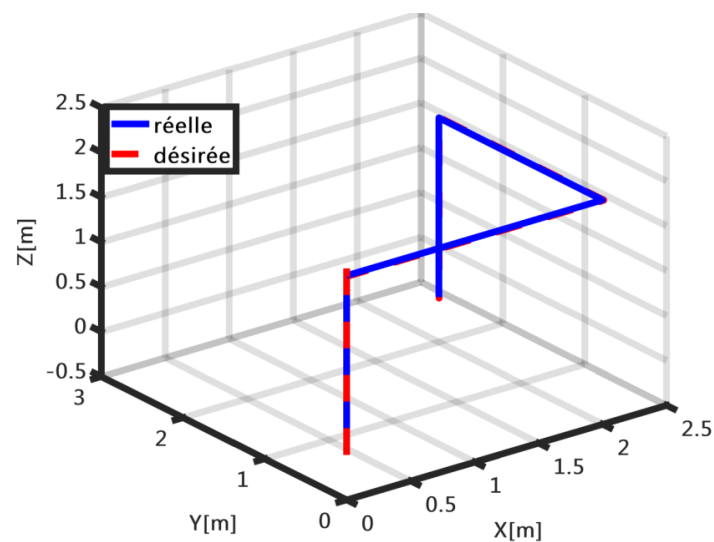


Figure II.4 Trajectoire globale en 3D

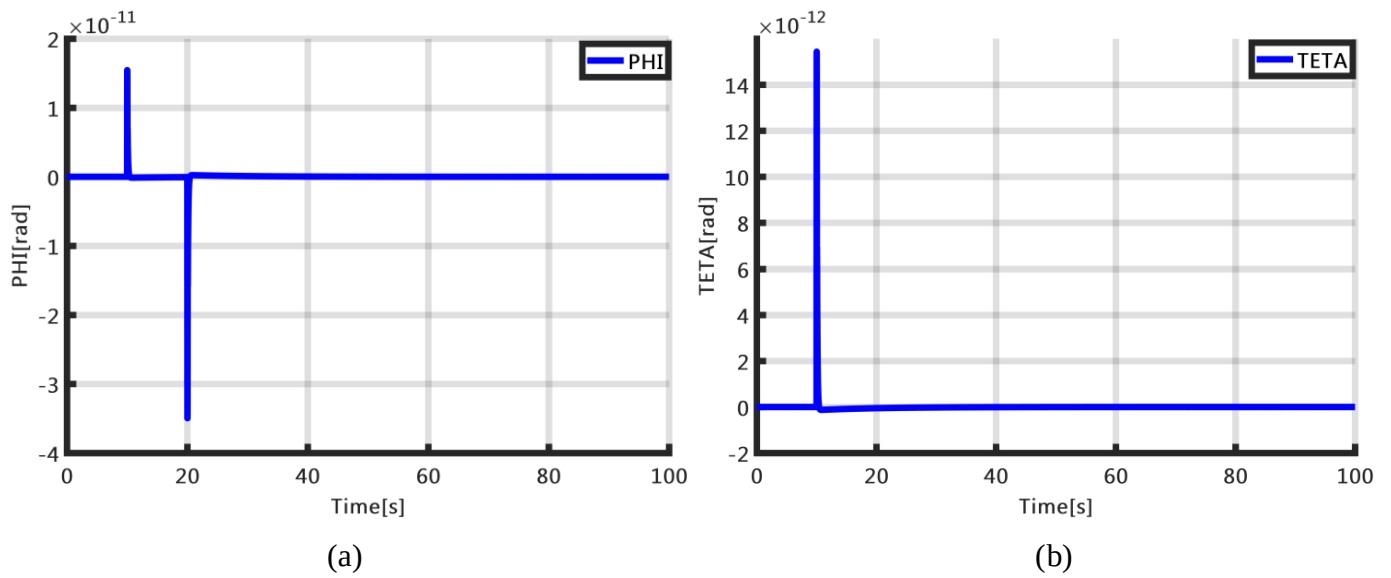
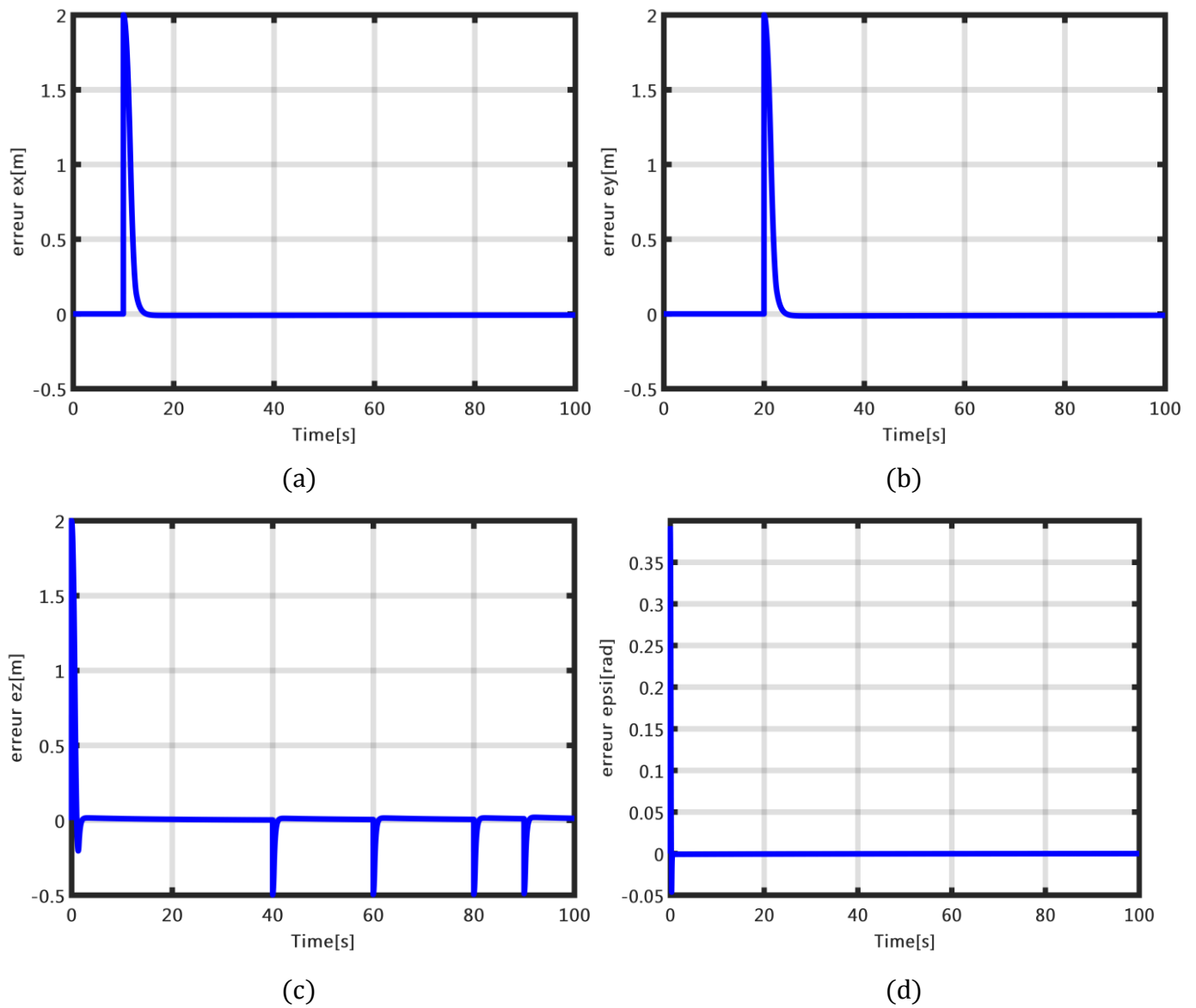
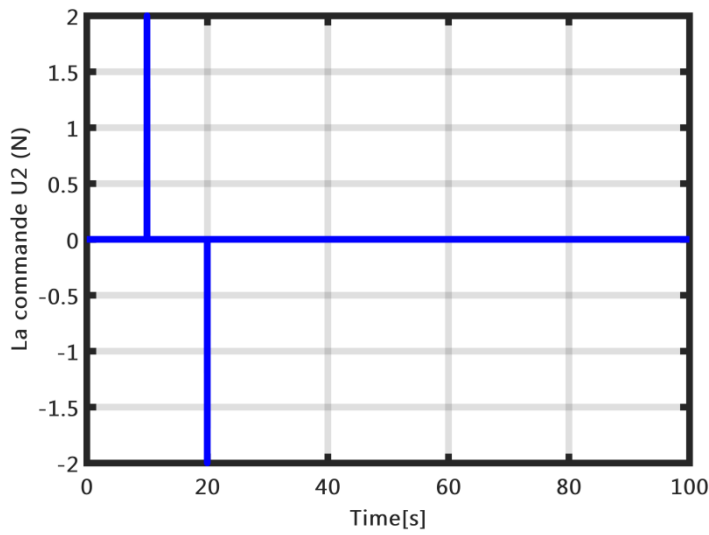
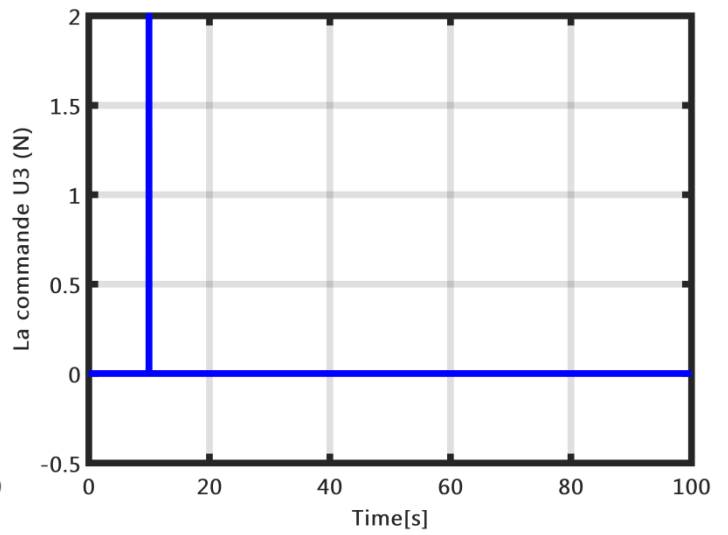


Figure II.5 L'angle de tangage et de roulis

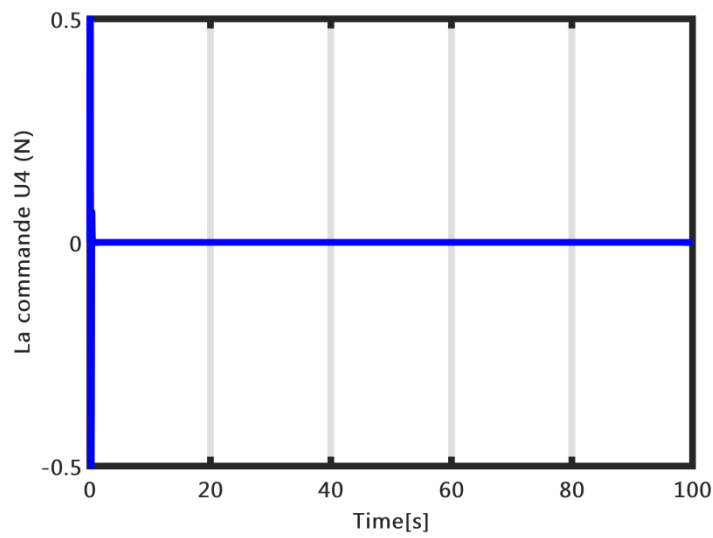
Figure II.6 Les erreurs de déplacement selon les axes (x, y, z, ψ)



(a)

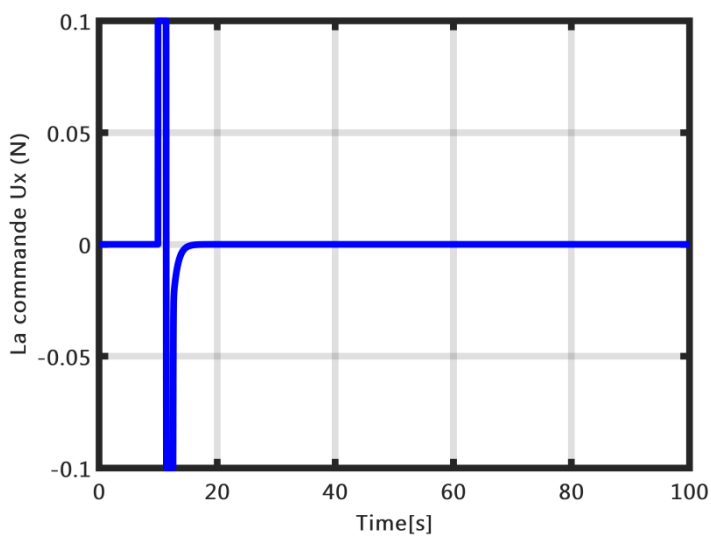


(b)

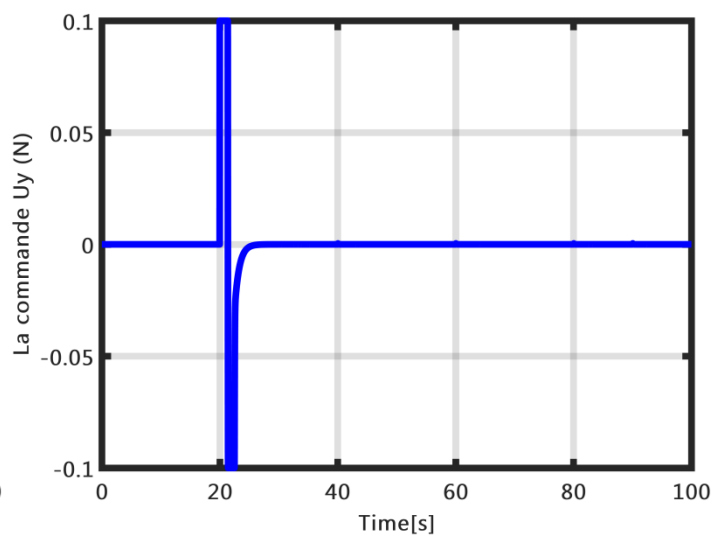


(c)

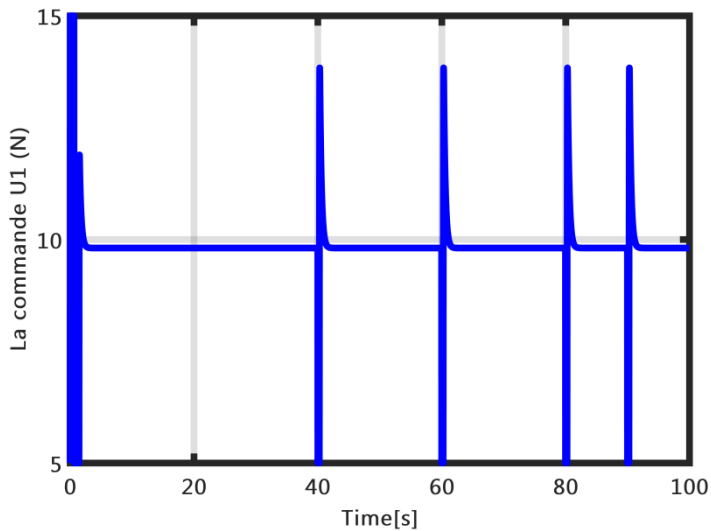
Figure II.7 Signaux de commande de la rotation



(a)



(b)



(c)

Figure II.8 Signaux de commande de la translation

En étudiant ces réponses nous avons extraits les temps de réponse à 5% de chaque variable :

- Le temps de réponse en Z est de 6.48s.
- Le temps de réponse en X est de 8s
- Le temps de réponse en Y est de 7.93s
- Le temps de réponse en ψ est de 0.42s.

■

4.1.1 Interprétation des résultats de simulation :

Les figures (II-3) à (II-8) montrent les performances de commande obtenues dans le cas des trajectoires en échelon. Le drone doit effectuer un mouvement vertical de 2m puis un mouvement suivant x d'une distance de 2m ensuite un mouvement suivant y de la même distance. Nous remarquons qu'il y a une bonne poursuite des trajectoires de référence en échelon (Figure II.3). Cependant, de légers dépassements sont observés (Figure II.3 (a), (b) et (c)). Ces dépassements ne sont pas tolérés et peuvent être dangereux surtout lors d'un atterrissage. La Figure II.5 présente l'évolution des angles de tangage et de roulis (les sorties du bloc non holonomie). Nous avons remarqué qu'il y a un accouplement entre ces deux angles (φ, θ), c'est à dire la variation de l'un influe sur l'autre angle par une petite variation. Nous avons constaté aussi que ces angles tendent vers zéro après avoir effectué leurs orientations désirées qui ne dépassent pas les domaines de variation de ces angles. A partir de la Figure II.7, on remarque que les commandes U_2 , U_3 et U_4 produisent des moments corrigeant les erreurs du suivi des angles (φ, θ) en assurant la poursuite de la trajectoire. Nous remarquons aussi qu'il y a un accouplement entre les commandes

U_2 , U_3 due aux accouplements des angles (φ, θ) désirés. La Figure II.8 montre que la commande U_1 effectue le déplacement vertical, et satisfait la condition d'équilibre ($U_1 = m \cdot g$). Nous avons constaté aussi qu'il y a un accouplement entre les commandes qui assurent le déplacement du drone suivant z, y et x, ce qui est interprété par les deux battements aux instants 10s et 20s. Les commandes U_x et U_y agissent d'une manière similaire afin de maintenir le déplacement suivant les axes x et y.

5. Conclusion :

Dans ce chapitre nous avons brièvement présenté le principe de fonctionnement d'un régulateur PID puis nous avons conçu plusieurs régulateur PID pour la commande de notre système.

Ensuite nous avons procédé à une optimisation de l'erreur quadratique qui consiste à minimiser l'erreur entre les entrées désirées et les sorties mesurée des différentes variables du système, puis nous avons effectué une série de tests de simulation en utilisant Matlab-Simulink, et enfin nous avons présenté les résultats de simulation ainsi que leurs interprétations.

Chapitre III: Commande par retour d'état linéarisant

Chapitre III: Commande par retour d'état linéarisant

1. Introduction :

Les techniques de commande non-linéaires basées sur la théorie de la géométrie différentielle ont été introduites principalement pour remédier aux problèmes rencontrés avec les commandes linéaires.

Actuellement, plusieurs méthodes traitant le sujet sont disponibles, parmi lesquelles, on trouve la technique de linéarisation au sens des entrées-sorties appelée aussi la linéarisation par bouclage (feedback linéarisation) [13].

Dans ce chapitre, nous présentons, dans une première étape, le principe de la technique de linéarisation au sens des entrées-sorties, puis dans une deuxième étape, nous utilisons cette technique pour concevoir un contrôleur non linéaire pour le modèle du quadri-rotor présenté dans le chapitre précédent. A la fin de ce chapitre nous présenterons les résultats de simulation en utilisant l'environnement SIMULINK/MATLAB.

2. Géométrie différentielle :

La géométrie différentielle est une discipline mathématique récente qui permet de faire de l'analyse sur des surfaces ou objets géométriques généraux.

2.1 Outils mathématiques :

Soit le système non linéaire :

$$\begin{cases} \dot{x} = f(x(t)) + g(x(t))u(t) \\ y = h(x(t)) \end{cases} \quad (\text{III-1})$$

Où l'état $x \in \mathbb{R}^n$, l'entrée $u \in \mathbb{R}^m$, la sortie $y \in \mathbb{R}^p$, $f(x)$, $g(x)$ et $h(x)$ sont des fonctions non linéaires.

2.2 Opérateurs de Lie (dérivée et crochet de Lie)

2.2.1 Dérivées de Lie :

Soient: $f: \mathbb{R}^n \rightarrow \mathbb{R}^n$ un champ de vecteurs et $h: \mathbb{R}^n \rightarrow \mathbb{R}$ une fonction scalaire. On introduit la dérivée de Lie comme étant une nouvelle fonction scalaire, notée $L_f h$ donnant la dérivée $h(x)$ dans la direction de $f(x)$, tel que:

$$L_f h(x) = \sum_{i=1}^n \frac{\partial h(x)}{\partial x_i} f_i \quad (\text{III-2})$$

Alors :

$$L_f h(x) = \begin{bmatrix} \frac{\partial h}{\partial x_1} & \frac{\partial h}{\partial x_2} & \dots & \frac{\partial h}{\partial x_n} \end{bmatrix} \begin{bmatrix} f_1 \\ f_2 \\ \vdots \\ f_n \end{bmatrix} \quad (\text{III-3})$$

De plus, si $g(x)$ est un autre champ de vecteurs alors la fonction scalaire $L_g L_f h(x)$ est donnée par:

$$L_g L_f h(x) = \frac{\partial L_f h(x)}{\partial x_i} g(x) \quad (\text{III-4})$$

3. Difféomorphisme :

Un difféomorphisme généralise la notion de changement de coordonnées au cadre non linéaire. Une fonction $\Phi: \mathbb{R}^n \rightarrow \mathbb{R}^n$, définie dans la variété M , est nommée un difféomorphisme si elle est lisse (*de classe C^∞*) et si sa fonction inverse Φ^{-1} existe est aussi lisse [14], [15], [16]

Le difféomorphisme est utilisé pour transformer un système non linéaire en un autre système linéaire en effectuant un changement de variables de la forme:

$$z = \Phi(x)$$

Où $\Phi(x)$ représente n variables:

$$\Phi(x) = \begin{bmatrix} \Phi_1(x) \\ \Phi_2(x) \\ \vdots \\ \Phi_n(x) \end{bmatrix}, \quad x = \begin{bmatrix} x_1 \\ x_2 \\ \vdots \\ x_n \end{bmatrix} \quad (\text{III-5})$$

Lorsque cette transformation existe, elle permet en particulier de stabiliser le système exactement comme s'il s'agissait d'un système linéaire, en utilisant les méthodes classiques des systèmes linéaires.

4. Commande par linéarisation entrée-sortie (E/S):

Le principe de la linéarisation E/S consiste à trouver une transformation qui permet de compenser les non-linéarités du modèle et ainsi rendre la relation entre la sortie d'un système et son entrée complètement linéaire. Cela revient à un contrôle interne et externe du système.

L'idée de base du contrôleur interne (boucle interne) est la manipulation algébrique de la dynamique complète ou partielle du système non linéaire, et cela afin de lui transformer en un système linéaire et par la suite de pouvoir lui appliquer des contrôleurs linéaires ce qui présente un contrôle externe (par exemple polynomial control). Cette technique est appelée « feedback linearization controller » [17][18].

La conception de la commande basée sur la linéarisation entrée-sortie comprend trois étapes [19]:

- Différencier la sortie $y^{(r)}$ 'r' fois jusqu'à l'apparition de l'entrée u , avec 'r' appelé le degré relatif.
- Choisir la commande u pour éliminer les non-linéarités et garantir la convergence de poursuite de trajectoire de la référence.
- Etudier la stabilité de la dynamique interne.

4.1 Système SISO :

On considère le système non linéaire avec une seule entrée et une seule sortie (SISO) décrit par le modèle non linéaire suivant :

$$\begin{cases} \dot{x} = f(x(t)) + g(x(t))u(t) \\ y = h(x(t)) \end{cases} \quad (\text{III-6})$$

x est un vecteur de $\mathbb{R}^n$. (Le vecteur d'état du système)

u est un scalaire représentant l'entrée. (La commande)

y est un scalaire représentant la sortie.

$f(x)$, $g(x)$ et $h(x)$ sont des fonctions non linéaires.

4.1.1 Degré relatif :

Le degré relatif r d'un système SISO (single input single output) peut être défini comme étant, le nombre minimum de fois qu'il faut dériver par rapport au temps l'expression de la sortie ($y(t)$) pour faire apparaître explicitement l'entrée ($u(t)$).

Le système (III-6) est dit de degré relatif r si :

$$\begin{cases} L_g L_f^i h(x) = 0 & 0 \leq i < r - 1 \\ L_g L_f^{r-1} h(x) \neq 0 \end{cases} \quad (\text{III-7})$$

Si nous dérivons la sortie du système d'équation (III-6), on obtient:

$$\dot{y} = \frac{dh(x)}{dx_i} \frac{dx}{dt} = \frac{dh(x)}{dx_i} (f(x) + g(x)u) = L_f h(x) + L_g h(x)u \quad (\text{III-8})$$

- a) Si le degré relatif est supérieur à 1, on a : $L_g h(x) = 0$. Et la deuxième dérivation de y conduit à:

$$\begin{cases} \ddot{y} = \frac{d(L_f h(x))}{dx} \frac{dx}{dt} = \frac{d(L_f h(x))}{dx} (f(x) + g(x)u) \\ \ddot{y} = L_f^2 h(x) + L_g h(x)u \end{cases} \quad (\text{III-9})$$

- b) Si encore le degré relatif supérieur à deux, on a : $L_g L_f h(x) = 0$, d'où:

$$\ddot{y} = L_f^2 h(x) \quad (\text{III-10})$$

Ce qui montre que :

$$\begin{cases} y^i = L_f^i h(x) & 0 \leq i < r \\ y^i = L_f^r h(x) + L_g L_f^{r-1} u \end{cases} \quad (\text{III-11})$$

Remarques: [24,25]

- Pour un système commandable, on a toujours $r \leq n$. Si l'entrée n'apparaît pas après n dérivations de la sortie, le système est non commandable.
- Pour $r = n$, le système admet une linéarisation exacte.
- Pour $r < n$, le système admet une linéarisation partielle. Et l'ordre du sous système linéarisé est égal à r .

Dans notre cas on s'intéresse à la linéarisation exacte.

4.1.2 Linéarisation par bouclage statique:

La régulation d'état statique consiste à appliquer au système une entrée ne dépendant que de son vecteur d'état actuel (x) et d'une entrée de référence(v). Cette linéarisation se fait en deux étapes :

- La mise du système sous la forme normale.
- Le retour d'état linéarisant.

On peut donc trouver la forme normale du système en appliquant le changement de Coordonnées (un difféomorphisme) .

4.1.2.1 Forme normale :

La notion de forme normale d'un système non-linéaire est basée sur la définition d'un nouveau vecteur d'état au moyen d'un changement de variables. Ce nouveau vecteur d'état permet d'exprimer les équations d'état du système sous une forme considérablement plus simple que celle de départ.

On peut donc trouver la forme normale du système en appliquant le changement de Coordonnées (un difféomorphisme) suivant :

$$\phi = [\phi_1(x) \dots \dots \phi_n(x)]$$

Tel que:

$$\begin{cases} z_1 = \phi_1(x) = h(x) \\ z_2 = \phi_2(x) = L_f h(x) \\ \vdots \\ z_n = \phi_n(x) = L_f^{n-1} h(x) \end{cases} \quad (\text{III-12})$$

Dans les nouvelles coordonnées $(z_1, z_2, \dots, z_n)$, le système (III-6) s'écrit :

$$\begin{cases} \dot{z}_1 = z_2 \\ \dot{z}_2 = z_3 \\ \vdots \\ \dot{z}_n = b(z) + a(z)u \end{cases} \quad (\text{III-13})$$

Avec:

$$\begin{cases} a(z) = L_g L_f^{n-1} h(x) \\ b(z) = L_f^n h(x) \end{cases} \quad (\text{III-14})$$

4.1.2.2 Retour d'état linéarisant:

Pour pouvoir utiliser les techniques de commande des systèmes linéaires, nous avons à travers la relation qui lie la commande u à l'état z , supposée mesurable et à une entrée de référence v , pour obtenir un comportement entre/sortie linéaire. Cette relation est donnée par la fonction des états suivants: [20,21]

$$u = \alpha(z) + \beta(z)v \quad (\text{III-15})$$

Où: v désigne la nouvelle entrée pour laquelle le système est linéaire.

On choisit α et β pour que la dernière équation de (III-13) Devient :

$$\dot{z}_n = v \quad (\text{III-15})$$

Par identification, on trouve :

$$\begin{cases} \alpha(z) = \frac{1}{a(z)} \\ \beta(z) = -\frac{b(z)}{a(z)} \end{cases} \quad (\text{III-16})$$

La forme de l'équation (III-17) correspond à un système linéaire contrôlable; elle est appelée la forme canonique de Brunovsky sous forme réduite. On a [20,21] :

$$\dot{z}_n = Az + bv \quad (\text{III-17})$$

Avec:

$$A = \begin{bmatrix} 0 & 1 & 0 & \dots & \dots & 0 \\ 0 & 0 & 1 & 0 & \dots & 0 \\ \vdots & \vdots & \vdots & \vdots & \vdots & \vdots \\ \vdots & \vdots & \vdots & \vdots & \vdots & 1 \\ 0 & 0 & \dots & \dots & \dots & 0 \end{bmatrix}; \quad b = \begin{bmatrix} 0 \\ 0 \\ \vdots \\ 0 \\ 1 \end{bmatrix}$$

L'équation (III-17) représente un intégrateur d'ordre n .

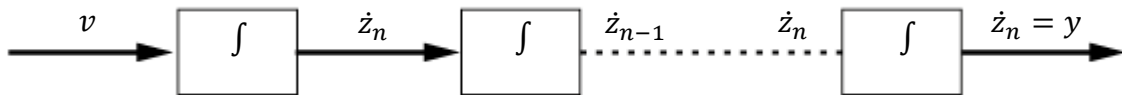


Figure III.1 Linéarisation exacte par retour d'état (Forme canonique de Brunovsky)

Nous avons r intégrateurs, on obtient un système linéaire de la forme:

$$\frac{Y(s)}{V(s)} = \frac{1}{s^r} \quad (\text{III-18})$$

Donc, nous pouvons concevoir maintenant un contrôleur pour ce système en utilisant n'importe quelle méthode de conception des contrôleurs linéaires.

4.2 Système (MIMO) :

Les notions utilisées pour les systèmes SISO peuvent être étendues au cas des systèmes MIMO [21,22,23]. Les systèmes sont considérés carrés, c'est-à-dire possédant le même nombre d'entrée que de sortie, de ce fait un grand nombre de résultats issus des systèmes monovariabiles peuvent faire l'objet d'une extension au cas multivariable (la transformation locale des coordonnées, dynamique des zéros ...etc.), les systèmes dynamiques non linéaires, multi-entrées/multi-sorties sont donnés sous la forme suivante :

$$\begin{cases} \dot{x} = f(x) + \sum_{i=1}^p g_i(x)u_i & i = 1, 2, \dots, p \\ y_i = h_i(x) \end{cases} \quad (\text{III-19})$$

Où:

$x = [x_1, x_2, \dots, x_n]^T$ est le vecteur d'état.

$u = [u_1, u_2, \dots, u_p]^T$ Est le vecteur des commandes.

$y = [y_1, \dots, y_p]^T$ Représente le vecteur des sorties.

f, g_i Sont des champs de vecteurs lisses.

h_i est une fonction lisse.

4.2.1 Degré relatif :

On dit que le système (III-19) à pour degré relatif (r) s'il vérifié:

$$\begin{cases} L_{g_j} \left(L_f^k h_j(x) \right) u_i = 0 & 0 < k < r_j - 1 & 1 \leq j \leq p \\ L_{g_j} \left(L_f^k h_j(x) \right) u_i \neq 0 & k = r_j - 1 \end{cases} \quad (\text{III-20})$$

Sachant que:

$$L_{g_j} L_f^k h_j(x) = \sum_{j=1}^k \frac{\partial L_f h(x)}{\partial x_j} g_{ij}(x) \quad (\text{III-21})$$

4.2.2 Linéarisation par bouclage statique:

Le problème de la linéarisation par bouclage statique consiste à trouver une relation linéaire entre l'entrée et la sortie en dérivant la sortie jusqu'à ce qu'au moins une entrée apparaisse en utilisant l'expression suivante [33]:

$$y_j^{(r_j)} = L_f^{r_j} h_j(x) + \sum_{i=1}^p L_{g_i} \left(L_f^{r_j-1} h_j(x) \right) u_i \quad j = 1, 2, \dots, p \quad (\text{III-22})$$

Où: $L_f^i h_j$ et $L_{g_i}^i h_j$ sont dérivées de h_j respectivement dans la direction de f et g_i , r_j est le nombre de dérivées nécessaires pour qu'au moins une des entrées apparaisse dans l'expression (III-22). Ce ' r_j ' est connu sous le nom de degré relatif correspondant à la sortie y_j . Le degré relatif total ' r ' obtenu à l'aide de (III-22) doit être inférieur ou égal à l'ordre du système ' n ' :

$$r = \sum_{j=1}^p r_j \leq n$$

Pour trouver l'expression de la loi linéarisante $u(x, v)$ qui permet de rendre la relation entre l'entrée et la sortie linéaire, on réécrit l'expression (III-22) sous forme matricielle, si nous avons ' p ' Sorties, nous obtenons l'expression suivante [24]:

$$[y_1^{r_1} \dots y_p^{r_p}]^T = A_0(x) + B_0(x)u \quad (\text{III-23})$$

Avec:

$$A_0(x) = \begin{bmatrix} L_f^{r_1} h_1(x) \\ \dots \\ L_f^{r_p} h_p(x) \end{bmatrix} \quad \text{et} \quad B_0(x) = \begin{bmatrix} L_{g_1} L_f^{r_1-1} h_1(x) & L_{g_2} L_f^{r_1-1} h_1(x) & \dots & L_{g_p} L_f^{r_1-1} h_1(x) \\ L_{g_1} L_f^{r_2-1} h_2(x) & L_{g_2} L_f^{r_2-1} h_2(x) & \dots & L_{g_p} L_f^{r_2-1} h_2(x) \\ \dots & \dots & \dots & \dots \\ L_{g_1} L_f^{r_p-1} h_p(x) & L_{g_2} L_f^{r_p-1} h_p(x) & \dots & L_{g_p} L_f^{r_p-1} h_p(x) \end{bmatrix}$$

La loi de linéarisation est donnée donc sous la forme :

$$u = B_0^{-1}(x)[-A_0(x) + V] \quad (\text{III-24})$$

$B_0(x)$ doit être une matrice inversible.

Le vecteur $V = [v_1 v_2 \dots v_i]$ représente les nouvelles commandes conçues afin d'imposer une nouvelle dynamique.

La figure ci-dessous illustre la linéarisation par bouclage statique [21].

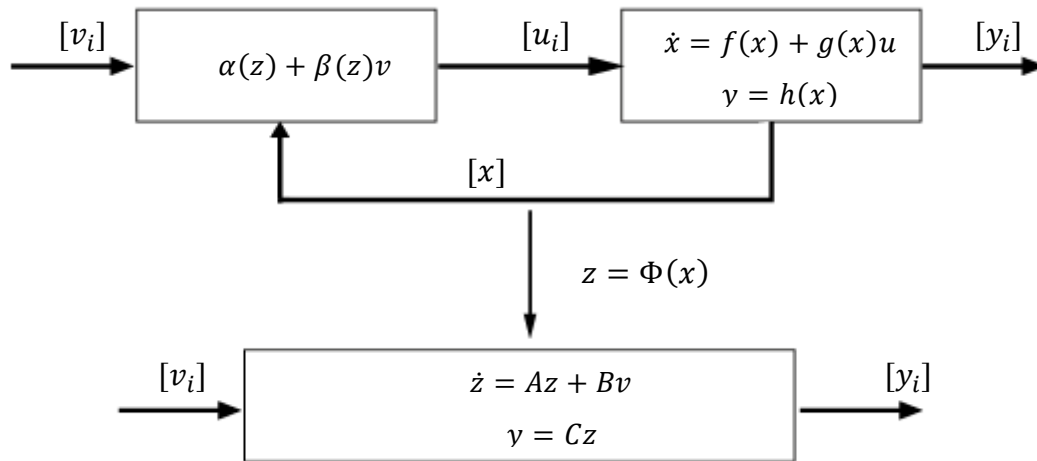


Figure III.2 Synoptique de linéarisation entrée/sortie des systèmes MIMO

En substituant (III-24) dans (III-23), le système équivalent devient linéaire et découplé sous la forme:

$$y_i^{r_j} = v_i \quad j = 1, 2, \dots, p \quad \text{et} \quad j = 1, 2, \dots, p \quad (\text{III-25})$$

La dynamique de la sortie y est alors ramenée à une chaîne d'intégrateurs. Le système équivalent est formé de p (p : nombre de sorties) chaînes d'intégrateurs en cascade comme le montre la figure suivante (Figure III.3):

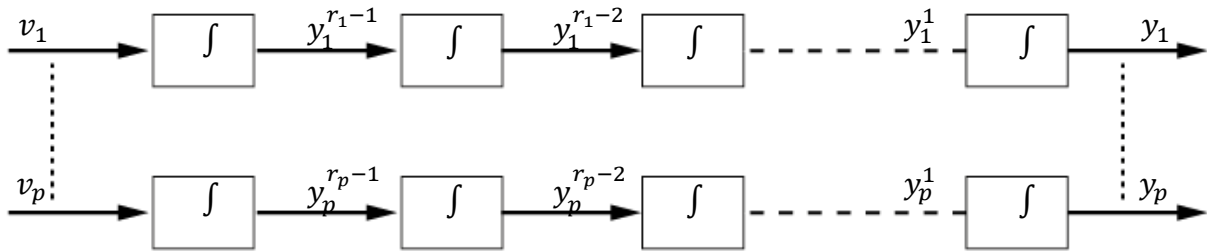


Figure III.3 Dynamique de sorties y_i

4.3 Le contrôle externe :

La linéarisation par bouclage (retour d'état) nous amène à un système formé de plusieurs chaînes d'intégrateurs découplés entre elles. Cette configuration rend le système instable en boucle ouverte, donc un second bouclage est nécessaire pour commander et stabiliser le système [24].

4.3.1 Notion de pôles :

Les pôles d'un système caractérisent le comportement dynamique du processus, alors si on change les coefficients de l'équation caractéristique, on cherche donc à modifier le

comportement propre du système commandé en jouant avec la position des pôles, ceci est effectué par un système bouclé par placement de pôle [25].

Prenons notre système linéarisé de la forme canonique de Brunovsky, on peut déduire que chaque sortie est reliée à une seule entrée par une chaîne d'intégrateur en cascade. On peut donc réaliser une commande par un placement de pôle pour chaque chaîne d'intégrateur. La bonne conception des nouvelles entrées internes (v_j) a une très grande importance. Elles sont Calculées en imposant un régime statique ($y_j \text{ référence} = y_j \text{ mesuré}$) et une dynamique sur l'erreur. Pour un problème de poursuite de trajectoire, on a [25] :

$$v_j = y_j^r - \sum_{i=1}^r k_{j(r+1-i)} e_j^{r-i} \quad j = 1, 2 \dots r ; i = 1, 2 \dots r \quad (\text{III-26})$$

Où $k_{j(r+1-i)}$ sont des constantes positives et $e_j = y_j \text{ mesuré} - y_j \text{ référence}$.

Et

$$p^r + \sum_{i=1}^r k_{j(r+1-i)} p^{r-i} = 0 \quad (\text{III-27})$$

4.3.1.1 Choix des pôles:

Pour déterminer les gains du correcteur $k_{j(r+1-i)}$. On utilise (III-27) et on impose des pôles de tels sorts que le système bouclé soit [16]:

- Stable et que sa réponse ne présente aucun dépassement.

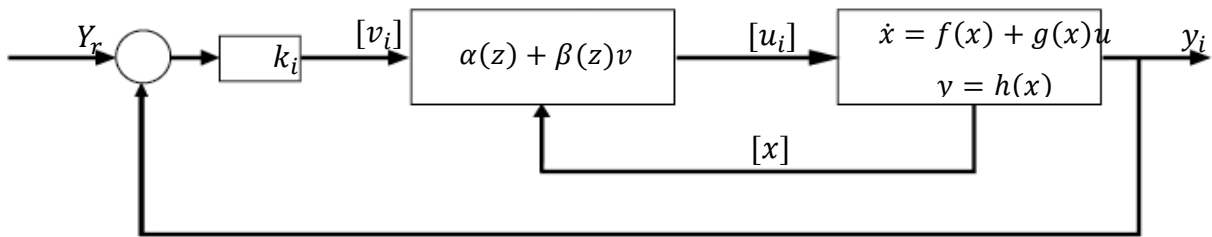


Figure III.4 Schéma bloc du système linéarisé en boucle fermée

5. Application de la technique de linéarisation par bouclage au quadri-rotor :

5.1 Stratégie de commande :

La stratégie de commande adoptée est récapitulée en deux sous-systèmes ; le premier est consiste en boucle interne lié à la commande des mouvements de rotation, et le deuxième est lié au commande des mouvements de translation, comme il est exposé sous le schéma synoptique de la figure (Figure III.5) [11]:

Cette stratégie se fait en deux étapes :

- Une transformation de coordonnées de chaque boucle
- Un retour d'état non linéaire pour chaque boucle.

Et suivant le choix de sortie du système, la linéarisation est soit exacte ou partielle [6], Afin d'assurer la poursuite des trajectoires désirées suivant les trois axes (X,Y,Z) et l'angle du lacet ψ .

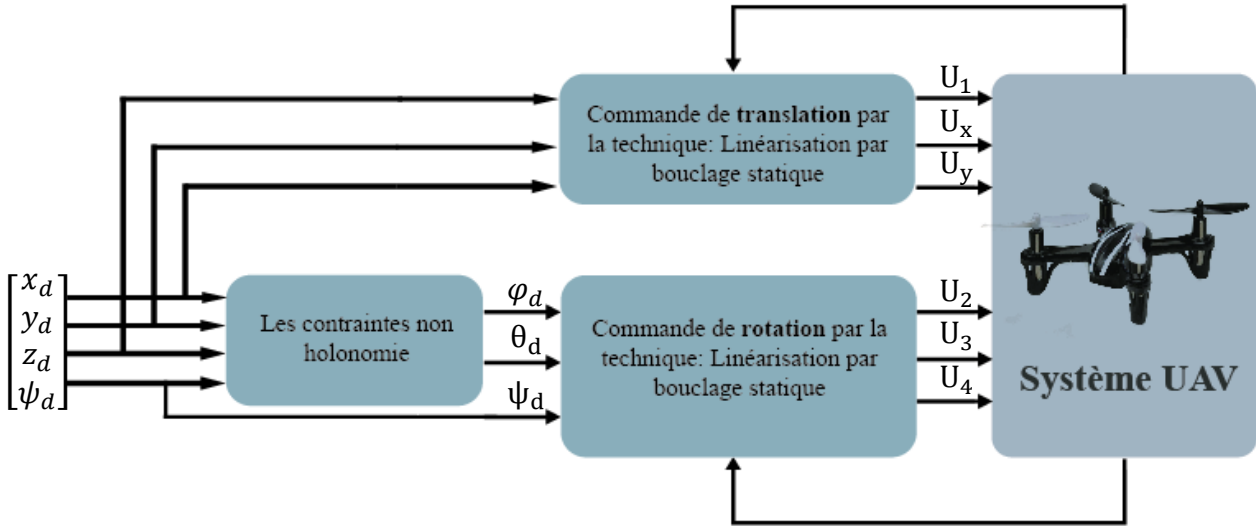


Figure III.5 Schéma synoptique de commande

Etant donné le modèle dynamique du système sous sa forme d'état (I-20), les fonctions $f(x)$, $g(x)$ et $h(x)$ sont alors données par :

$$f(x) = \begin{pmatrix} x_2 \\ a_1 x_2^2 + a_2 x_4 x_6 + a_3 \bar{\Omega} x_4 \\ x_4 \\ a_4 x_4^2 + a_5 x_2 x_6 + a_6 \bar{\Omega} x_2 \\ x_6 \\ a_7 x_6^2 + a_8 x_2 x_4 \\ x_8 \\ a_9 x_8 \\ x_{10} \\ a_{10} x_{10} \\ x_{12} \\ a_{11} x_{12} \end{pmatrix} = \begin{pmatrix} f_1(x) \\ f_2(x) \end{pmatrix} \quad (\text{III-28})$$

Pour procéder à une commande par linéarisation entrée/sortie, il faut définir tout d'abord le vecteur de sortie Y , et qui se présente comme suit [6]:

$$Y = \begin{pmatrix} x_1 \\ x_3 \\ x_5 \\ x_7 \\ x_9 \\ x_{11} \end{pmatrix} = \begin{pmatrix} Y_1 \\ Y_2 \end{pmatrix} \quad (\text{III-29})$$

5.2 Sous-système d'orientation :

Pour le contrôle de l'attitude on utilise le sous-système du mouvement de rotation donné comme suit :

$$\begin{cases} \ddot{\phi} = \frac{1}{J_x} [\dot{\theta} \dot{\psi} (J_z - J_y) + d \cdot U_2 - K_{ax} \dot{\phi}^2 - J_r \bar{\Omega} \dot{\theta}] \\ \ddot{\theta} = \frac{1}{J_y} [\dot{\phi} \dot{\psi} (J_z - J_x) + d \cdot U_3 - K_{ay} \dot{\theta}^2 - J_r \bar{\Omega} \dot{\phi}] \\ \ddot{\psi} = \frac{1}{J_z} [\dot{\phi} \dot{\theta} (J_x - J_y) + U_4 - K_{az} \dot{\psi}^2] \end{cases} \quad (\text{III-30})$$

Et sa représentation d'état :

$$\begin{cases} \dot{x}_1 = x_2 \\ \dot{x}_2 = a_1 x_4 x_6 + a_2 x_2^2 + a_3 x_4 \bar{\Omega} + b_1 U_2 \\ \dot{x}_3 = x_4 \\ \dot{x}_4 = a_4 x_2 x_6 + a_5 x_4^2 + a_6 x_2 \bar{\Omega} + b_2 U_3 \\ \dot{x}_5 = x_6 \\ \dot{x}_6 = a_7 x_2 x_4 + a_8 x_6^2 + b_3 U_4 \end{cases} \quad (\text{III-31})$$

Avec:

$$f_1(x) = \begin{pmatrix} x_2 \\ a_1 x_2^2 + a_2 x_4 x_6 + a_3 \bar{\Omega} x_4 \\ x_4 \\ a_4 x_2 x_6 + a_5 x_4^2 + a_6 \bar{\Omega} x_2 \\ x_6 \\ a_7 x_2 x_4 + a_8 x_6^2 \end{pmatrix} \quad (\text{III-32})$$

Et le vecteur de sortie Y_1 pour ce système est défini par :

$$Y_1 = \begin{bmatrix} x_1 \\ x_3 \\ x_5 \end{bmatrix} \quad (\text{III-33})$$

5.2.1 Degré relatif :

Soit r_0 le degré relatif du sous-système d'orientation:

On calcul le degré relatif associe à chaque sortie :

$$\begin{cases} y_1 = x_1 = L_f^0 h_1(x) = h_1(x) \\ \dot{y}_1 = \dot{x}_1 = x_2 = L_{f1} h_1(x) \\ \ddot{y}_1 = \dot{x}_2 = a_1 x_4 x_6 + a_2 x_2^2 + a_3 x_4 \bar{\Omega} + b_1 U_2 = L_{f1}^2 h_1(x) + L_{g1} L_{f1} h_1(x) U_2 \end{cases} \quad (\text{III-34})$$

Le degré relatif correspondant à (III-34) : $r_1 = 2$

$$\begin{cases} y_2 = x_3 = L_{f1} h_2(x) = h_2(x) \\ \dot{y}_2 = \dot{x}_3 = x_4 = L_{f1} h_2(x) \\ \ddot{y}_2 = \dot{x}_4 = a_4 x_2 x_6 + a_5 x_4^2 + a_6 x_2 \bar{\Omega} + b_2 U_3 = L_{f1}^2 h_2(x) + L_{g1} L_{f1} h_2(x) U_3 \end{cases} \quad (\text{III-35})$$

Le degré relatif correspondant à (III-35): $r_2 = 2$

$$\begin{cases} y_3 = x_5 = L_{f1} h_3(x) = h_3(x) \\ \dot{y}_3 = \dot{x}_5 = x_6 = L_{f1} h_3(x) \\ \dot{y}_3 = \dot{x}_6 = a_7 x_2 x_4 + a_8 x_6^2 + b_3 U_4 = L_{f1} h_3(x) + L_{g1} L_{f1} h_3(x) U_4 \end{cases} \quad (\text{III-36})$$

Le degré relatif correspondant à (III-36): $r_3 = 2$

D'où le degré relatif du sous-système d'orientation est: $r_0 = r_1 + r_2 + r_3 = 6 = n_1$

n_1 Étant l'ordre du sous-système d'orientation ; cela montre qu'il n'y a pas une dynamique de zéros. Dans ce cas on est dans la linéarisation par bouclage statique.

5.2.2 Le difféomorphisme :

$$\begin{cases} z_1^1 = L_{f0} h_1(x) = y_1 = x_1 \\ z_2^1 = L_f h_1(x) = L_f(x_1) = x_2 \\ z_1^2 = h_2(x) = x_3 \\ z_2^2 = L_f h_2(x) = L_f(x_3) = x_4 \\ z_1^3 = h_3(x) = x_5 \\ z_2^3 = L_f h_3(x) = L_f(x_5) = x_6 \end{cases} \quad (\text{III-37})$$

Alors

$$\begin{cases} z_1^1 = x_1 \\ z_2^1 = x_2 \\ z_1^2 = x_3 \\ z_2^2 = x_4 \\ z_1^3 = x_5 \\ z_2^3 = x_6 \end{cases} \quad (\text{III-38})$$

5.2.3 La forme normale

$$\begin{cases} \dot{z}_1^1 = \dot{x}_1 = x_2 = z_2^1 \\ \dot{z}_2^1 = \dot{x}_2 = a_1 x_2^2 + a_2 x_4 x_6 a_3 + \bar{\Omega} x_4 + b_1 U_2 = v_1 = \ddot{y}_1 \\ \dot{z}_1^2 = \dot{x}_3 = x_4 = z_2^2 \\ \dot{z}_2^2 = \dot{x}_4 = a_4 x_4^2 + a_5 x_2 x_6 + a_6 \bar{\Omega} x_2 + b_2 U_3 = v_2 = \ddot{y}_2 \\ \dot{z}_1^3 = \dot{x}_5 = x_6 = z_2^3 \\ \dot{z}_2^3 = \dot{x}_6 = a_7 x_6^2 + a_8 x_2 x_4 + b_3 U_4 = v_3 = \ddot{y}_3 \end{cases} \quad (\text{III-39})$$

Alors:

$$\begin{cases} \dot{z}_1^1 = z_2^1 \\ \dot{z}_2^1 = v_1 \\ \dot{z}_1^2 = z_2^2 \\ \dot{z}_2^2 = v_2 \\ \dot{z}_1^3 = z_2^3 \\ \dot{z}_2^3 = v_3 \end{cases} \quad (\text{III-40})$$

5.2.3.1 Ecriture matricielle :

$$\dot{z}_n = Az + bv \quad (\text{III-41})$$

Avec:

$$A = \begin{bmatrix} 0 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 \\ 0 & 0 & 0 & 0 & 0 & 0 \end{bmatrix}; \quad b = \begin{bmatrix} 0 & 0 & 0 \\ 1 & 0 & 0 \\ 0 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 0 \\ 0 & 0 & 1 \end{bmatrix}$$

5.2.4 Boucle externe :

Notons que la synthèse des différentes lois de commandes stabilisantes (boucle externe) se fait par placement de pôles comme suit :

- Calcul des lois de commandes stabilisantes :

Tel que:

$$\begin{cases} V_2 = -k_1(e_1) - k_2(-\dot{e}_1) + \ddot{x}_{1d} \\ V_3 = -k_3(e_2) - k_4(-\dot{e}_2) + \ddot{x}_{3d} \\ V_4 = -k_5(e_3) - k_6(-\dot{e}_3) + \ddot{x}_{5d} \end{cases} \quad (\text{III-42})$$

Avec :

$$\begin{cases} e_1 = (x_1 - x_{1d}) = (\varphi - \varphi_d) \\ e_2 = (x_3 - x_{3d}) = (\theta - \theta_d) \\ e_3 = (x_5 - x_{5d}) = (\psi - \psi_d) \end{cases}$$

Les pôles (Valeurs propres) choisis sont comme suit :

$$p_1 = -0.685$$

$$p_2 = -1.315$$

$$p_3 = -0.582$$

$$p_4 = -1.1717$$

$$p_5 = -1.06$$

$$p_6 = -2.74$$

Après calcul des différents polynômes caractéristiques et identification des coefficients k_i , les valeurs de ces derniers sont comme suit :

$$\begin{cases} k_1 = 2.3 \\ k_2 = 3.2 \\ k_3 = 2 \\ k_4 = 3 \\ k_5 = 2.9 \\ k_6 = 3.6 \end{cases}$$

5.2.5 Boucle interne :

5.2.5.1 Retour d'état non linéaire

Par identification terme à terme entre le système d'équations (III-39) et le système d'équations (III-42), il en résulte que les trois lois de commandes du sous-système d'orientation (retour d'état non linéaire) sont comme suit :

$$\begin{cases} U_4 = \frac{1}{b_3}(-k_1(x_1 - x_{1d}) - k_2(x_2 - \dot{x}_{1d}) + \ddot{x}_{1d} - a_8 x_6^2 - a_7 x_2 x_4) \\ U_3 = \frac{1}{b_2}(-k_3(x_3 - x_{3d}) - k_4(x_4 - \dot{x}_{3d}) + \ddot{x}_{3d} - a_4 x_4^2 - a_5 x_2 x_5 - a_6 \bar{\Omega} x_2) \\ U_2 = \frac{1}{b_1}(-k_5(x_5 - x_{5d}) - k_6(x_6 - \dot{x}_{5d}) + \ddot{x}_{5d} - a_1 x_2^2 - a_2 x_4 x_6 - a_3 \bar{\Omega} x_4) \end{cases} \quad (\text{III-43})$$

5.3 Sous système de translation :

Pour le contrôle de position on utilise le sous-système du mouvement de translation donné comme suit :

$$\begin{cases} \ddot{x} = \frac{1}{m} \{(\cos \varphi \cos \psi \sin \theta + \sin \psi \sin \varphi) U_1 - k_{ftx} \dot{x}\} \\ \ddot{y} = \frac{1}{m} \{(\cos \varphi \sin \psi \sin \theta + \cos \psi \sin \varphi) U_1 - k_{fity} \dot{y}\} \\ \ddot{z} = \frac{1}{m} \{(\cos \varphi \cos \theta) U_1 - k_{ftz} \dot{z}\} - g \end{cases} \quad (\text{III-44})$$

Et sa représentation d'état :

$$\begin{cases} \dot{x}_7 = x_8 \\ \dot{x}_8 = a_9 x_8 + U_x \frac{U_1}{m} \\ \dot{x}_9 = x_{10} \\ \dot{x}_{10} = a_{10} x_{10} + U_y \frac{U_1}{m} \\ \dot{x}_{11} = x_{12} \\ \dot{x}_{12} = a_{11} x_{12} + \frac{\cos x_7 \cos x_9}{m} U_1 - g \end{cases} \quad (\text{III-45})$$

Et le vecteur de sortie Y_2 pour ce système est défini par :

$$Y_2 = \begin{bmatrix} x_7 \\ x_9 \\ x_{11} \end{bmatrix} \quad (\text{III-46})$$

5.3.1 Degré relatif :

Soit r_t le degré relatif du sous-système de translation

On calcul le degré relatif associe a chaque sortie :

$$\begin{cases} y_4 = x_7 = L_f h_4(x) = h_4(x) \\ \dot{y}_4 = \dot{x}_7 = x_8 = L_f h_4(x) \\ \ddot{y}_4 = \ddot{x}_8 = a_9 x_8 + U_x \frac{U_1}{m} = L_{f^2} h_4(x) + L_{g^2} L_{f^2} h_4(x) U_x \end{cases} \quad (\text{III-47})$$

Le degré relatif correspondant à (III-47) : $r_1 = 2$

$$\begin{cases} y_5 = x_9 = L_{f_2} h_5(x) = h_5(x) \\ \dot{y}_5 = \dot{x}_9 = x_{10} = L_{f_2} h_5(x) \\ \ddot{y}_5 = \dot{x}_{10} = \dot{x}_{10} = a_{10} x_{10} + U_y \frac{U_1}{m} = L_{f_2} h_5(x) + L_{g_2} L_{f_2} h_5(x) U_y \end{cases} \quad (\text{III-48})$$

Le degré relatif correspondant à (III-48): $r_2 = 2$

$$\begin{cases} y_6 = x_{11} = L_{f_2} h_6(x) = h_6(x) \\ \dot{y}_6 = \dot{x}_{11} = x_{12} = L_{f_2} h_6(x) \\ \ddot{y}_6 = \dot{x}_{12} = a_{11} x_{12} + \frac{\cos(x_7) \cos(x_9)}{m} = L_{f_2} h_6(x) + L_{g_2} L_{f_2} h_6(x) U_1 \end{cases} \quad (\text{III-49})$$

Le degré relatif correspondant à (III-49): $r_3 = 2$

D'où le degré relatif du sous-système d'orientation est: $r_0 = r_1 + r_2 + r_3 = 6 = n_2$

n_2 Étant l'ordre du sous-système d'orientation ; cela montre qu'il n'y a pas une dynamique de zéros. Dans ce cas on est dans la linéarisation par bouclage statique.

5.3.2 Le difféomorphisme :

$$\begin{cases} z_1^4 = L_{f^0_2} h_4(x) = y_4 = x_7 \\ z_2^4 = L_{f_2} h_4(x) = L_{f_2}(x_7) = x_8 \\ z_1^5 = h_5(x) = x_9 \\ z_2^5 = L_{f_2} h_5(x) = L_{f_2}(x_9) = x_{10} \\ z_1^6 = h_6(x) = x_{11} \\ z_2^6 = L_{f_2} h_6(x) = L_{f_2}(x_{11}) = x_{12} \end{cases} \quad (\text{III-50})$$

Alors:

$$\begin{cases} z_1^4 = x_7 \\ z_2^4 = x_8 \\ z_1^5 = x_9 \\ z_2^5 = x_{10} \\ z_1^6 = x_{11} \\ z_2^6 = x_{12} \end{cases} \quad (\text{III-51})$$

5.3.3 La forme normale:

$$\begin{cases} \dot{z}_1^4 = \dot{x}_7 = x_8 = z_2^4 \\ \dot{z}_2^4 = \dot{x}_8 = a_9 x_8 + U_x \frac{U_1}{m} = v_4 = \ddot{y}_4 \\ \dot{z}_1^5 = \dot{x}_9 = x_{10} = z_2^5 \\ \dot{z}_2^5 = \dot{x}_{10} = a_{10} x_{10} + U_y \frac{U_1}{m} = v_5 = \ddot{y}_5 \\ \dot{z}_1^6 = \dot{x}_{11} = x_{12} = z_2^6 \\ \dot{z}_2^6 = \dot{x}_{12} = a_{11} x_{12} + \frac{\cos(x_7) \cos(x_9)}{m} = v_6 = \ddot{y}_6 \end{cases} \quad (\text{III-52})$$

Alors:

$$\begin{cases} \dot{z}_1^4 = z_2^4 \\ \dot{z}_2^4 = v_4 \\ \dot{z}_1^5 = z_2^5 \\ \dot{z}_2^5 = v_5 \\ \dot{z}_1^6 = z_2^6 \\ \dot{z}_2^6 = v_6 \end{cases} \quad (\text{III-53})$$

5.3.3.1 Ecriture matricielle :

$$\dot{z}_n = Az + bv \quad (\text{III-54})$$

Avec:

$$A = \begin{bmatrix} 0 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 \\ 0 & 0 & 0 & 0 & 0 & 0 \end{bmatrix}; \quad b = \begin{bmatrix} 0 & 0 & 0 \\ 1 & 0 & 0 \\ 0 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 0 \\ 0 & 0 & 1 \end{bmatrix}$$

5.3.4 Boucle externe :

Notons que la synthèse des différentes lois de commandes stabilisantes (boucle externe) se fait par placement de pôles comme suit :

- Calcul des lois de commandes stabilisantes :

Tel que:

$$\begin{cases} V_4 = -k_7(e_4) - k_8(\dot{e}_4) + \ddot{x}_{7d} \\ V_5 = -k_9(e_5) - k_{10}(\dot{e}_5) + \ddot{x}_{9d} \\ V_6 = -k_{11}(e_6) - k_{12}(\dot{e}_6) + \ddot{x}_{11d} \end{cases} \quad (\text{III-55})$$

Avec :

$$\begin{cases} e_4 = (x_7 - x_{7d}) = (x - x_d) \\ e_5 = (x_9 - x_{9d}) = (y - y_d) \\ e_6 = (x_{11} - x_{11d}) = (z - z_d) \end{cases}$$

Les pôles (Valeurs propres) choisis sont comme suit :

$$p_7 = -0.46$$

$$p_8 = -6.54$$

$$p_9 = -5.66$$

$$p_{10} = -0.4$$

$$p_{11} = -0.59$$

$$p_{12} = -3.41$$

Après calcul des différents polynômes caractéristiques et identification des coefficients K_i , les valeurs de ces derniers sont comme suit :

$$\begin{cases} k_7 = 1.9 \\ k_8 = 3 \\ k_9 = 0.2 \\ k_{10} = 0.1 \\ k_{11} = 0.5 \\ k_{12} = 0.4 \end{cases}$$

5.3.5 Boucle interne :

5.3.5.1 Retour d'état non linéaire

Par identification terme à terme entre le système d'équations (III-52) et le système d'équations (III-55), il en résulte que les trois lois de commande du sous-système d'orientation (retour d'état non linéaire) sont comme suit :

$$\begin{cases} U_x = \frac{m}{U_1} (-k_7(x_7 - x_{7d}) - k_8(x_8 - \dot{x}_{7d}) - a_9 x_8) \\ U_y = \frac{m}{U_1} (-k_9(x_9 - x_{9d}) - k_{10}(x_{10} - \dot{x}_{9d}) - a_{10} x_4) \\ U_1 = \frac{m}{\cos x_9 \cos x_{11}} (-k_{11}(x_{11} - x_{11d}) - k_6(x_{12} - \dot{x}_{11d}) + g - a_{11} x_{12}) \end{cases} \quad (\text{III-56})$$

Tel que: $U_1 \neq 0$ et $\cos x_9 \cos x_{11} \neq 0$

6. Résultats de simulation :

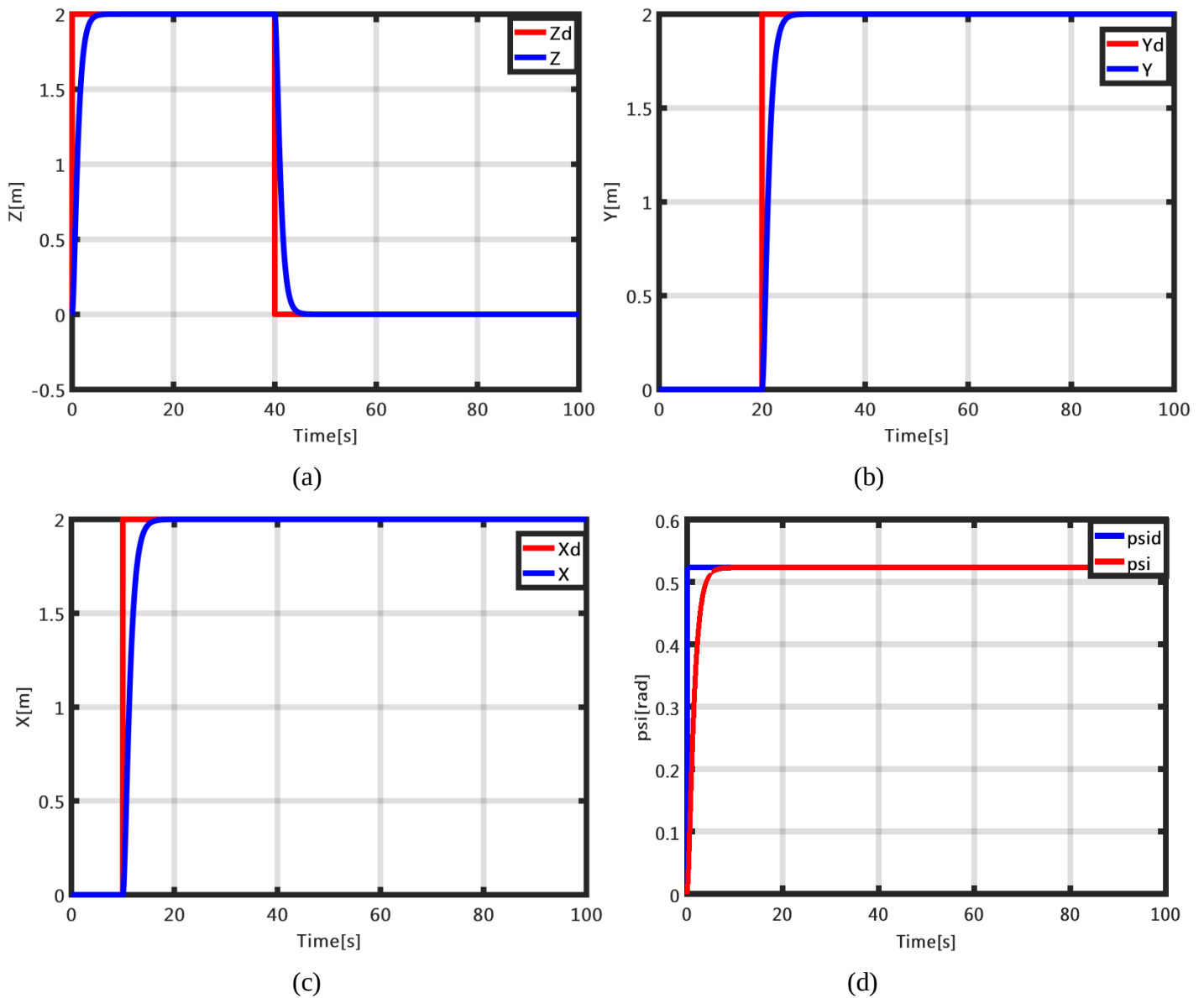


Figure III.6 Résultats de poursuite, dans le cas des trajectoires de références en échelon, selon les axes (X, Y, Z, ψ)

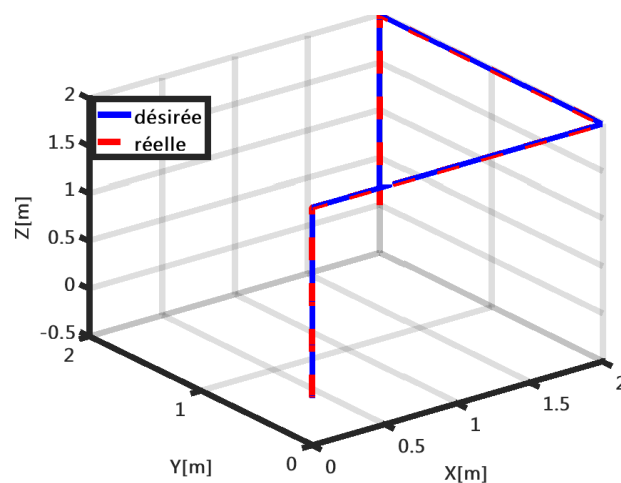


Figure III.7 Trajectoire globale du quadri-rotor en 3D

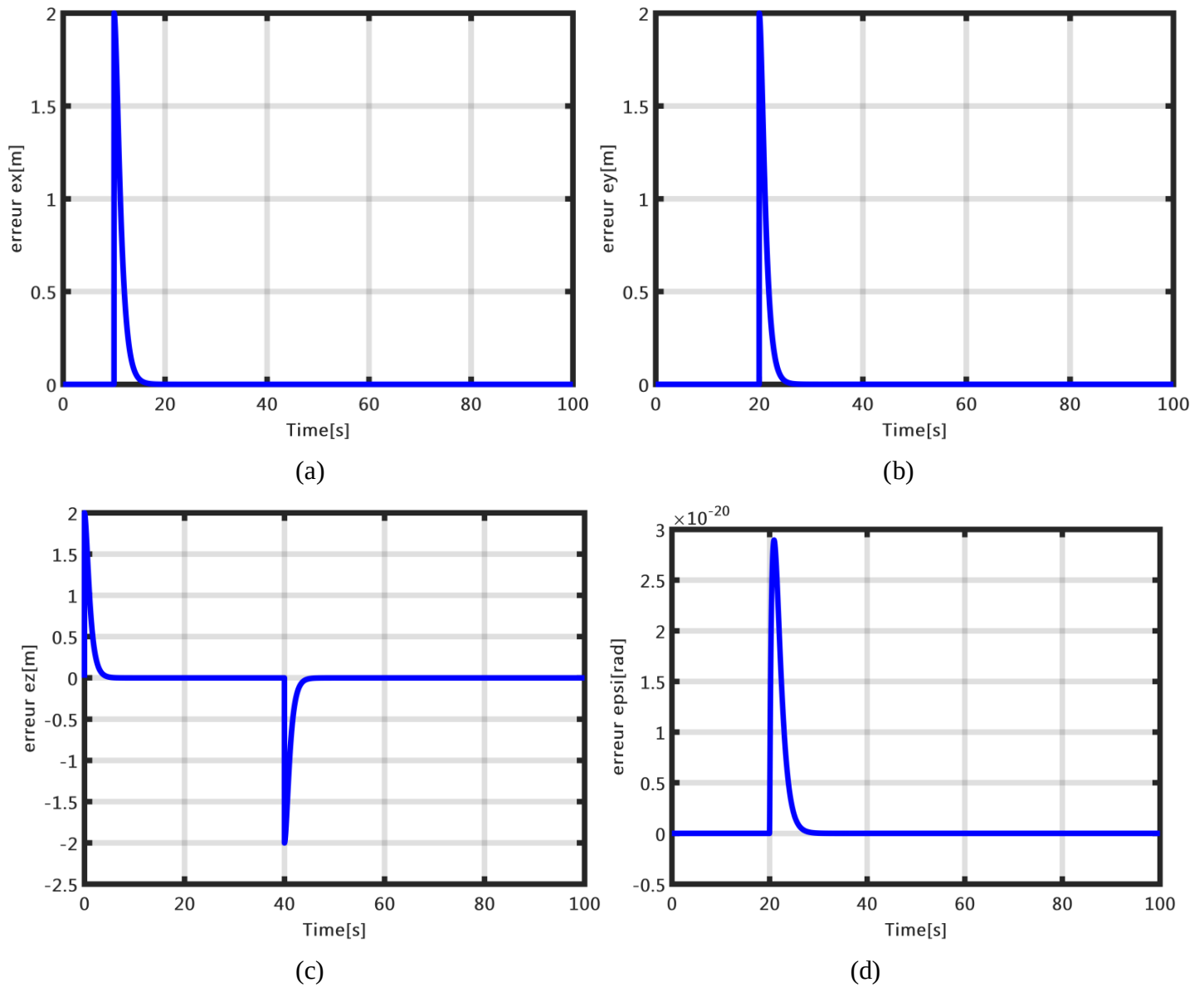


Figure III.8 les erreur de déplacement selon les 3 axes

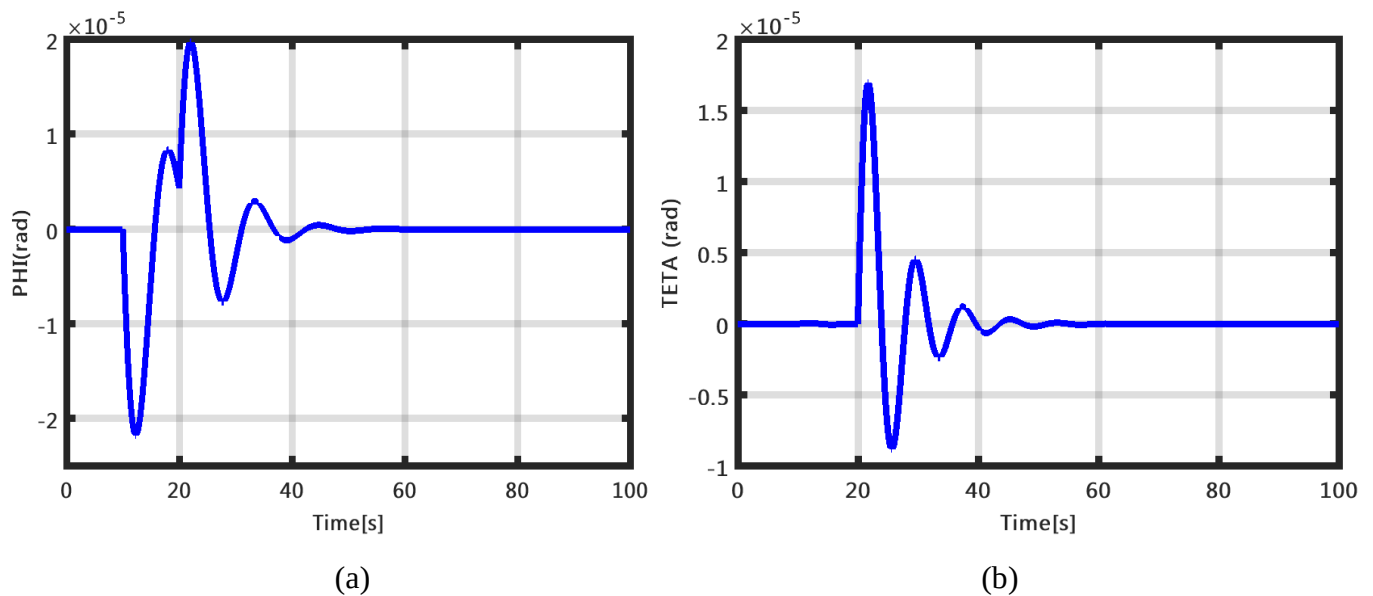


Figure III.9 L'angle de tangage et de roulis

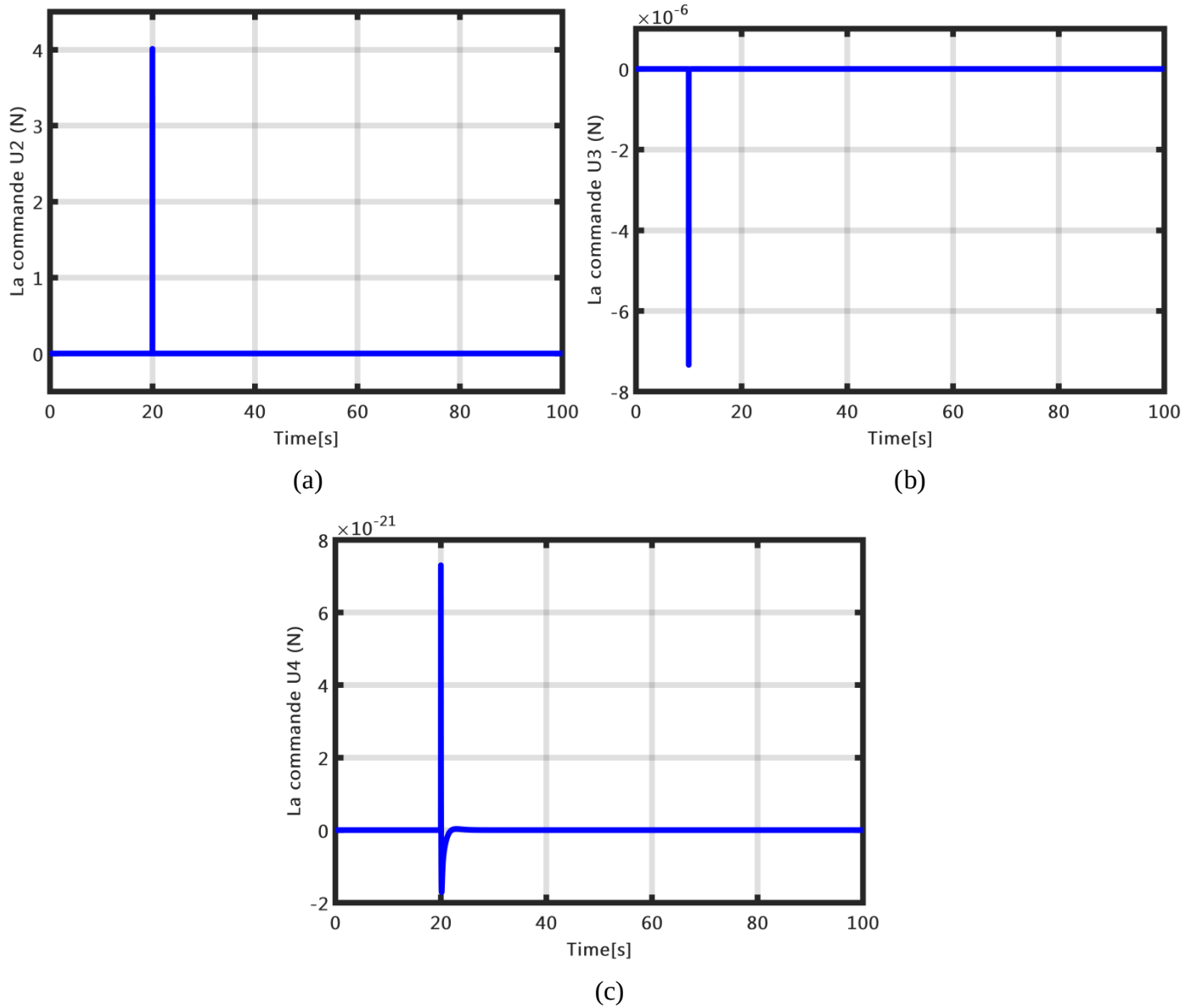
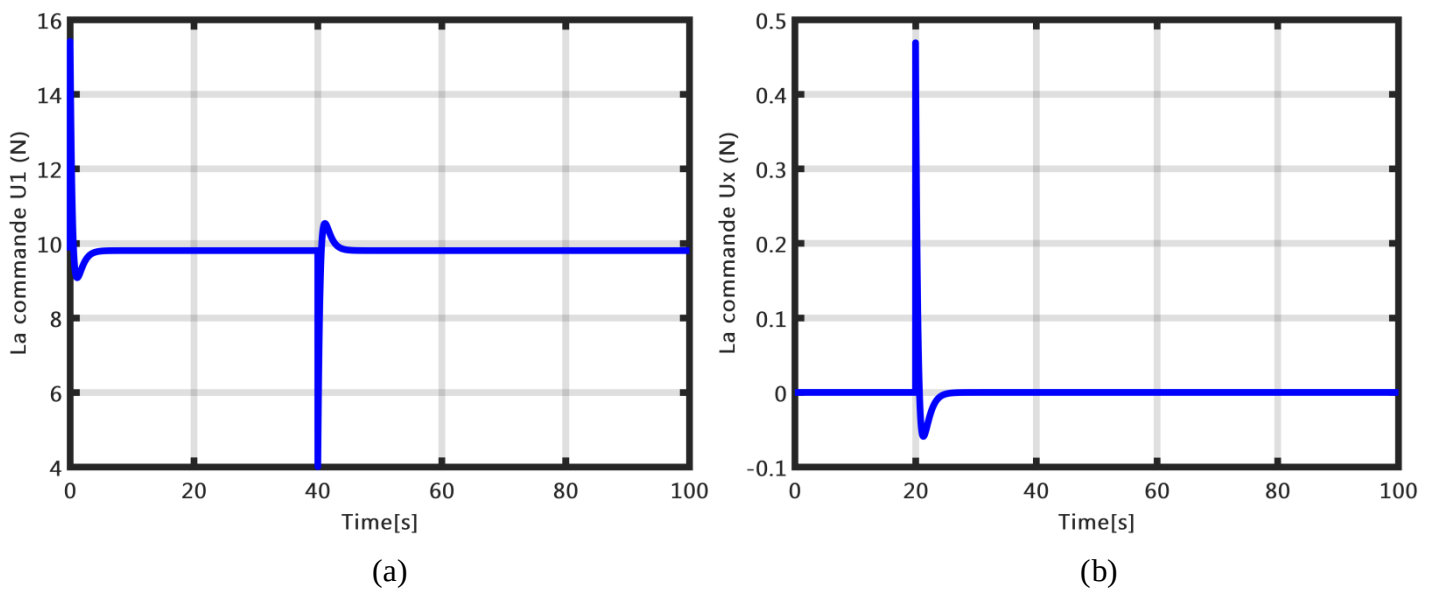


Figure III.10 signaux de commande des rotations



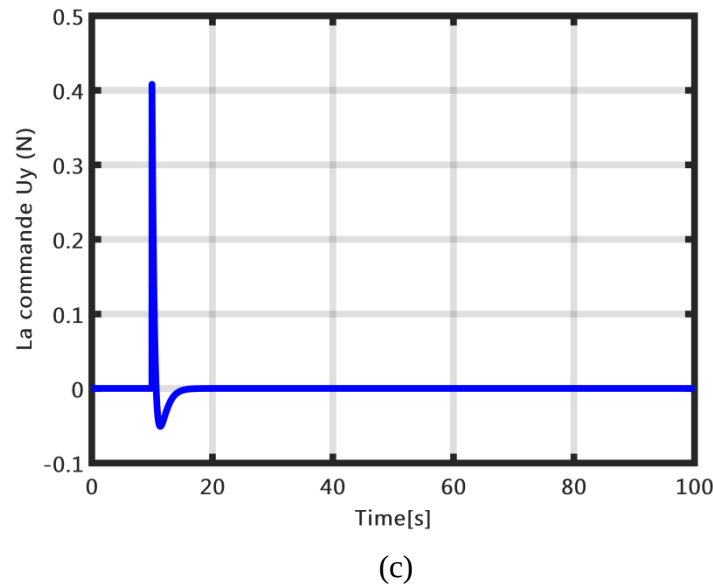


Figure III.11 signaux de commande des translations

En étudiant ces réponses nous avons extraits les temps de réponse à 5% de chaque variable :

- Le temps de réponse en Z est de 4.48s.
- Le temps de réponse en X est de 5.7s
- Le temps de réponse en Y est de 7.93s
- Le temps de réponse en Psy est de 4.5s.

6.1 Interprétations sur les résultats :

Les figures (III-6) à (III-11) montrent les performances de commande obtenues dans le cas des trajectoires en échelon. Le drone doit effectuer un mouvement vertical de 2m puis un mouvement suivant x d'une distance de 2m ensuite un mouvement suivant y de la même distance. Nous obtenons, une bonne précision de poursuite et nous remarquons que, dans ce cas, la poursuite s'effectue sans aucun dépassement (Figure III.6 (a)– (d)). Les mêmes figures montrent bien le découplage des sorties (z, x et y), c'est-à-dire la variation de chaque sortie n'influe pas sur les autres. A partir de la Figure III.7 on constate que l'erreur converge vers zéro, cela prouve que la trajectoire parcourue par le drone suit la valeur désirée. Les figures (III-9) (a) et (b) montrent la convergence des angles (φ, θ) vers zéro. Egalement, les figures (III-10) et (III-11) montrent les signaux de commandes obtenus.

7. Conclusion :

Dans ce chapitre, nous avons présenté essentiellement le concept de la linéarisation par bouclage entrée/sortie (feedback linearization), puis nous avons utilisé le modèle découplé en deux sous-systèmes (horizontal et vertical) du quadri-rotor donné dans le chapitre précédent. En se basant sur la technique de commande par linéarisation entrée-sortie, nous avons développé pour chaque sous-système une loi de commande. Les résultats de simulation des lois de commandes développées et appliquées sur le modèle du quadri-rotor ont montré une bonne stabilité du système et aussi une bonne poursuite de trajectoire. Cela prouve l'efficacité de la loi de commande synthétisée.

Chapitre IV:Commande PDC (Compensation parallèle distribuer) basée sur les modèles Takagi-Sugeno

Chapitre IV: Commande PDC basée sur les modèles Takagi-Sugeno

1. Introduction

Il est clair que, d'une part, les systèmes linéaires bénéficient d'une abondance d'outils d'analyse et de synthèse au détriment de la précision de représentation, et d'autre part, les systèmes non linéaires sont plus complexes certes mais plus précis. Afin d'exploiter l'avantage caractérisant les systèmes linéaires et au même temps la qualité de représentation des systèmes non linéaires, le formalisme Takagi-Sugeno (T-S) sera utilisé.

Dans ce chapitre, nous présenterons, dans une première étape les différentes techniques d'obtention d'un modèle T-S et donnerons un exemple illustratif sur la méthode la plus utilisée pour y aboutir, ainsi que la représentation des notions de stabilité, stabilisation et de suivi de trajectoire de ce type de modèle par une loi de commande classiquement développée pour cette famille de modèle nommée PDC (Parallel Distributed Compensation).

Puis dans une seconde étape, nous utiliserons cette technique pour concevoir un contrôleur non linéaire pour le modèle du quadri-rotor présenté dans le chapitre précédent.

A la fin de ce chapitre, nous présenterons les résultats de simulation en utilisant l'environnement SIMULINK/MATLAB.

2. Définition des modèles T-S:

Considérons l'équation générale d'un système non linéaire dynamique :

$$\begin{cases} \dot{x}(t) = f(x(t)) + g(x(t))u(t) \\ y(t) = h(x(t)) \end{cases} \quad (\text{IV-1})$$

Où $x(t) \in \mathbb{R}^n$ est le vecteur d'état, $u(t) \in \mathbb{R}^m$ est le vecteur de commande, $y(t) \in \mathbb{R}^q$ est le vecteur de sortie. $f(x)$, $g(x)$ et $h(x)$ sont des fonctions non linéaires avec une dimension appropriée. Le système non linéaire (IV-1) est décrit par un ensemble de règles floues "si-alors" qu'il représente localement les relations entrées/sorties d'un système en exprimant chaque conclusion par un système linéaire.

La $i^{\text{ème}}$ règle du modèle flou s'écrit :

$$R^i: \text{Si } z_1(t) \text{ est } F_1^i(z_1(t)) . \text{ ET } z_2(t) \text{ est } F_2^i(z_2(t)) \dots \dots \dots z_p(t) \text{ est } F_p^i(z_p(t))$$

Alors:

$$\begin{cases} \dot{x}(t) = A_i x(t) + B_i u(t) \\ y(t) = C_i x(t) + D_i u(t) \end{cases} \quad (\text{IV-2})$$

Où: R^i représente la $i^{\text{ème}}$ règle floue, $i = 1, 2, \dots, r$, $F_i^j(z_i(t))$ pour $j = 1, 2, \dots, p$ sont les sous-ensembles flous, r le nombre de règles floues, $z_j(t)$ sont les variables de prémisses qui dépendent de l'entrée et/ou de l'état du système, $A_i \in \mathbb{R}^{n \times n}$ est la matrice d'état, $B_i \in \mathbb{R}^{n \times m}$ est la matrice d'entrée, $C_i \in \mathbb{R}^{q \times n}$ est la matrice de sortie.

A chaque règle R^i est attribuée un poids noté $w_i(z(t))$. Ce poids dépend du degré d'appartenance des variables de prémisses $z_j(t)$ aux sous ensembles flous $F_i^j(z_i(t))$ et du connecteur « **ET** » reliant les prémisses choisies telles que :

$$w_i(z(t)) = \prod_{j=1}^p F_i^j(z_i(t)) \quad (\text{IV-3})$$

Pour $i = 1, 2, \dots, r$

$F_i^j(z_i(t))$ représente la valeur de la fonction d'appartenance $z_i(t)$ à l'ensemble flou F_i^j . On a alors les propriétés suivantes :

$$\begin{cases} \sum_{i=1}^r w_i(z(t)) > 0 \\ w_i(z(t)) \geq 0, \forall t \end{cases} \quad (\text{IV-4})$$

On pose:

$$h_i(z(t)) = \frac{w_i(z(t))}{\sum_{i=1}^r w_i(z(t))} \quad (\text{IV-5})$$

$h_i(z(t))$ représente donc la fonction d'activation (pondération) de la $i^{\text{ème}}$ règle du modèle flou. $i = 1, 2, \dots, r$, ces fonctions vérifient la propriété d'une somme convexe, c'est-à-dire:

$$\sum_{i=1}^r h_i(z(t)) = 1 \quad (\text{IV-6})$$

Avec:

$$h_i(z(t)) \geq 0 \quad (\text{IV-7})$$

Finalement, la défuzzification du modèle permet d'obtenir la représentation d'état d'un modèle non linéaire par l'interconnexion de modèles locaux invariants dans le temps par des fonctions d'activation non linéaires. On obtient alors :

$$\begin{cases} \dot{x}(t) = \sum_{i=1}^r h_i(z(t))(A_i x(t) + B_i u(t)) \\ y(t) = \sum_{i=1}^r h_i(z(t))(C_i x(t) + D_i u(t)) \end{cases} \quad (\text{IV-8})$$

Dans le cadre de la modélisation par modèles Takagi-Sugeno, on rencontre souvent les termes [14][15]:

- variables de prémisses
- fonctions d'activation
- zones de fonctionnement (sous espaces)
- règles floues

 **Règles floues :** dénombrées par $r \in \mathbb{N}$ dans la représentation d'état d'un modèle flou T-S. Elles correspondent au nombre de modèles locaux LTI.

 **variables de prémisses:** notées $z(t) \in \mathbb{R}^j$ grandeurs connues et accessibles permettent l'évaluation des fonctions d'appartenance. Elles dépendent éventuellement des variables d'état mesurables et/ou de la commande.

 **Fonctions d'activation :** notées $h_i(z(t)) \in \mathbb{R}^i \rightarrow \mathbb{R}$, ce sont des fonctions non linéaires dépendant des variables de prémisses associées aux différentes zones de fonctionnement. Elles permettent de traduire la contribution d'un modèle local LTI correspondant à un point de fonctionnement par rapport à la zone de fonctionnement du système. Ainsi, elles assurent le passage progressif d'un modèle local LTI aux modèles locaux voisins.

3. Construction des modèles T-S :

Dans la littérature, trois approches sont couramment décrites pour l'obtention d'un modèle TS :

3.1 Identification des modèles T-S :

À partir de mesures acquises des entrées et sorties d'un Système physique. Dans ce cas, le problème d'identification du modèle non linéaire se réduit à l'identification de modèles locaux autour de points de fonctionnement préalablement choisis d'un système non linéaire. Cette méthode est utilisée dans le cas de systèmes dotés d'une dynamique difficile à décrire à l'aide d'un modèle analytique.

3.2 Linéarisation d'un modèle non linéaire :

Autour d'un ensemble fini de points de Fonctionnement. Un représentant TS est obtenu par l'interconnexion des dynamiques locales à l'aide de fonctions d'appartenance judicieusement choisies (gaussiennes, triangulaires, trapézoïdales, etc.). Notons que cette approche constitue dans la plupart des cas une approximation du modèle non linéaire considéré.

3.3 Décomposition en secteurs non linéaires :

Basée sur des transformations polytopiques convexes des termes non linéaires d'un système dynamique. Cette méthode garantit la construction d'un modèle TS représentant exactement un modèle non linéaire sur un espace compact des variables d'état. Dans notre travail, notre intérêt s'est porté sur la troisième méthode dans la mesure où elle présente l'avantage de s'affranchir d'approximations de modélisation lors du passage du modèle analytique non linéaire au modèle TS (hormis une réduction éventuelle de l'espace de validité du modèle TS). Par ailleurs, notons que de manière générale, les transformations polytopiques convexes sont injectives. En effet, une infinité de modèles TS peuvent être associés à un système non linéaire donné. Néanmoins, une approche systématique d'obtention des modèles TS à partir d'un système non linéaire affine en la commande peut être utilisée [16]. Celle-ci est résumée par le lemme suivant :

Lemme [20] :

Si $x \in [-b \ a]$ borné, $a, b \in \mathbb{R}^+$, $f(x): \mathbb{R} \rightarrow \mathbb{R}$ borné sur $[-b \ a]$ alors :

il existe deux fonctions sur $w_1(x)$ et $w_2(x)$ ainsi que deux réels α et β tels que :

$$\begin{cases} f(x) = \alpha w_1(x) + \beta w_2(x) \\ w_1(x) + w_2(x) = 1, w_1(x) \geq 0, w_2(x) \geq 0 \end{cases} \quad (\text{IV-9})$$

▪ **Démonstration:**

Considérons la fonction $f(x)$ bornée tel que $\underline{f} \leq f(x) \leq \bar{f}$, on peut alors toujours écrire :

$$f(x) = \alpha w_1(x) + \beta w_2(x) \text{ avec } \alpha = \underline{f}, \beta = \bar{f}, w_1 = \frac{f(x) - \underline{f}}{\bar{f} - \underline{f}}, w_2 = \frac{\bar{f} - f(x)}{\bar{f} - \underline{f}} \quad (\text{IV-10})$$

Notons enfin que, selon les propriétés des termes non linéaires rencontrés dans le modèle mathématique non linéaire, on distingue deux types de représentants TS, En effet :

- Si toutes les non-linéarités du système sont continues et bornées, alors le modèle TS représente de manière exacte le système non linéaire sur l'intégralité de l'espace des variables d'état. Dans ce cas, le découpage en secteur non linéaire est dit « global » (Figure IV.1)

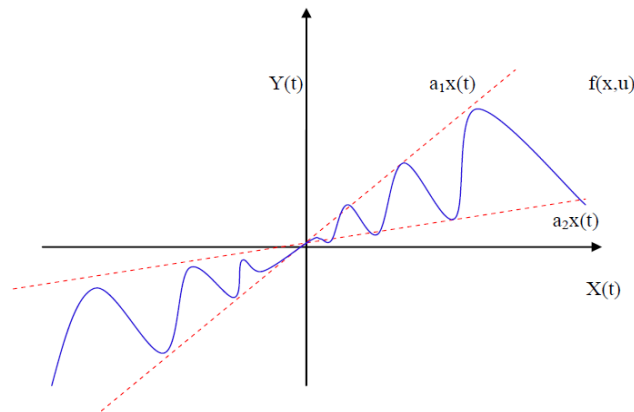


Figure IV.1 Secteur global

- Si tout ou partie des non-linéarités du système sont continues mais non bornées, alors le modèle TS représente de façon exacte le système non linéaire sur un sous-espace compact de l'espace des variables d'état. Dans ce cas, le découpage en secteur non linéaire est dit « local » (Figure IV.2).

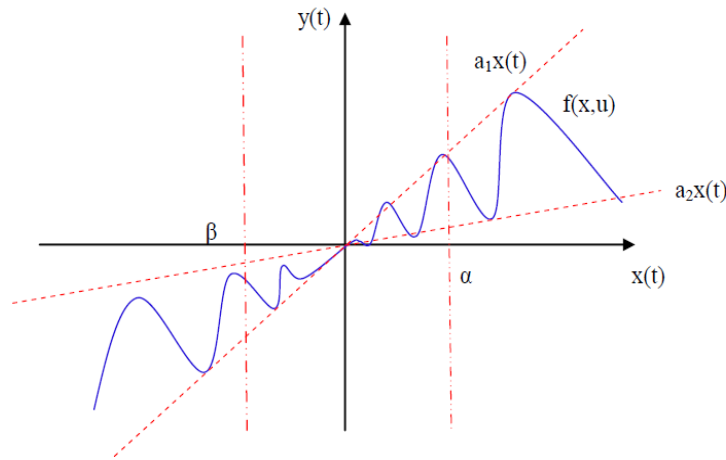


Figure IV.2 Secteur local

La Figure IV.3 illustre le schéma détaillé d'un modèle T-S standard. En effet, ils permettent de diminuer la complexité d'un problème non linéaire à traiter (stabilité, stabilisation, observation, diagnostic,...etc.) en le décomposant en un ensemble de problèmes linéaires locaux. L'ensemble des solutions locales correspondant à ces derniers constitue alors la solution globale du problème non linéaire initial.

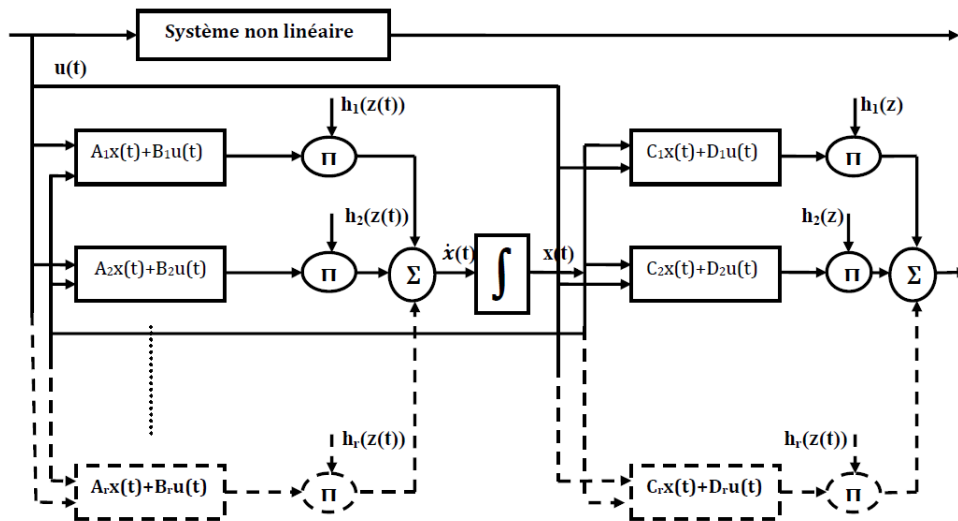


Figure IV.3 Structure et validation d'un modèle T-S

4. Exemple :

Dans cet exemple, nous expliquons comment arriver à obtenir le modèle flou T-S d'un système non linéaire en utilisant l'approche par secteur non-linéaire décrit précédemment. Une simulation est donnée pour montrer qu'un système non linéaire peut être approximé exactement par un ensemble des règles floues de type si-alors.

On considère le système non linéaire décrit par les équations suivantes :

$$\begin{cases} \dot{x}_1(t) = -x_1(t) + x_1(t)x_2^3(t) + u_1(t) \\ \dot{x}_2(t) = -x_2(t) + (3 + x_2(t))x_1^3(t) + u_2(t) \end{cases} \quad (\text{IV-11})$$

Pour la simplicité, nous supposons que x_1 et $x_2 \in [-1 \ 1]$, le système peut être réécrit comme suit :

$$\dot{x}(t) = \begin{bmatrix} -1 & x_1(t)x_2^2(t) \\ (3 + x_2(t))x_1^2(t) & -1 \end{bmatrix} x(t) + \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix} u(t) \quad (\text{IV-12})$$

Où: $x(t) = [x_1(t) \ x_2(t)]^T$, $x_1(t)x_2^2(t)$ et $(3 + x_2(t))x_1^2(t)$ sont les termes non linéaires, alors on définit les variables de prémisse comme suit :

$$\begin{cases} z_1(t) \equiv x_1(t)x_2^2(t) \\ z_2(t) = (3 + x_2(t))x_1^2(t) \end{cases} \quad (\text{IV-13})$$

Alors, on obtient :

$$\dot{x}(t) = \begin{bmatrix} -1 & z_1(t) \\ z_2(t) & -1 \end{bmatrix} x(t) + \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix} u(t)$$

Nous calculons les valeurs minimales et maximales de z_2 et z_1 :

$$\text{Max}(z_1(t)) = 1, \quad \text{Min}(z_1(t)) = -1, \quad \text{Max}(z_2(t)) = 4, \quad \text{Min}(z_2(t)) = 0.$$

A partir de ces valeurs, nous pouvons écrire :

$$\begin{cases} z_1(t) = F_{11}(z_1(t)) \cdot (1) + F_{12}(z_1(t)) \cdot (-1) \\ z_2(t) = F_{21}(z_2(t)) \cdot (4) + F_{22}(z_2(t)) \cdot (0) \end{cases}$$

Où:

$$\begin{cases} F_{11}(z_1(t)) + F_{12}(z_1(t)) = 1 \\ F_{21}(z_2(t)) + F_{22}(z_2(t)) = 1 \end{cases} \quad (\text{IV-14})$$

Par conséquent, les fonctions d'appartenance peuvent être calculées comme suit :

$$\begin{cases} F_{11}(z_1(t)) = \frac{(z_{1\max} - z_1)}{(z_{1\max} - z_{1\min})} \\ F_{12}(z_1(t)) = \frac{(z_1 - z_{1\min})}{(z_{1\max} - z_{1\min})} \\ F_{21}(z_2(t)) = \frac{(z_{2\max} - z_2)}{(z_{2\max} - z_{2\min})} \\ F_{22}(z_2(t)) = \frac{(z_2 - z_{2\min})}{(z_{2\max} - z_{2\min})} \end{cases} \quad (\text{IV-15})$$

Le système non linéaire (IV-1) peut être représenté par l'ensemble des règles floues suivantes:

Si z_1 est F_{11} et z_2 est F_{21} Alors $\dot{x}(t) = A_1(t)x(t) + B_1(t) u(t)$

Si z_1 est F_{11} et z_2 est F_{22} Alors $\dot{x}(t) = A_2(t)x(t) + B_2(t) u(t)$

Si z_1 est F_{12} et z_2 est F_{21} Alors $\dot{x}(t) = A_3(t)x(t) + B_3(t) u(t)$

Si z_1 est F_{12} et z_2 est F_{22} Alors $\dot{x}(t) = A_4(t)x(t) + B_4(t) u(t)$

Avec :

$$A_1 = \begin{bmatrix} -1 & 1 \\ 4 & -1 \end{bmatrix}, A_2 = \begin{bmatrix} -1 & 1 \\ 0 & -1 \end{bmatrix}, A_3 = \begin{bmatrix} -1 & -1 \\ 4 & -1 \end{bmatrix}, A_4 = \begin{bmatrix} -1 & -1 \\ 0 & -1 \end{bmatrix}$$

$$B_1 = B_2 = B_3 = B_4 = \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix}$$

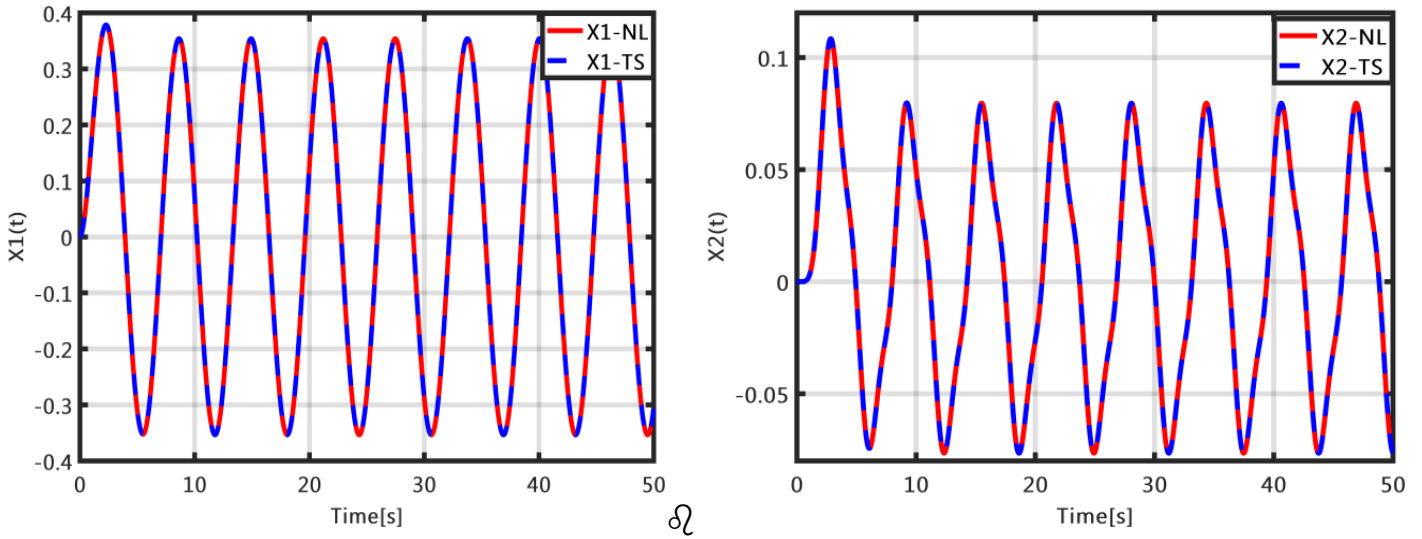
L'inférence du système flou est donnée par :

$$\begin{cases} \dot{x}(t) = \sum_{i=1}^4 h_i(z(t))(A_i x(t) + B_i(t)) \\ y(t) = \sum_{i=1}^4 h_i(z(t))(C_i x(t)) \end{cases} \quad (IV-16)$$

Avec:

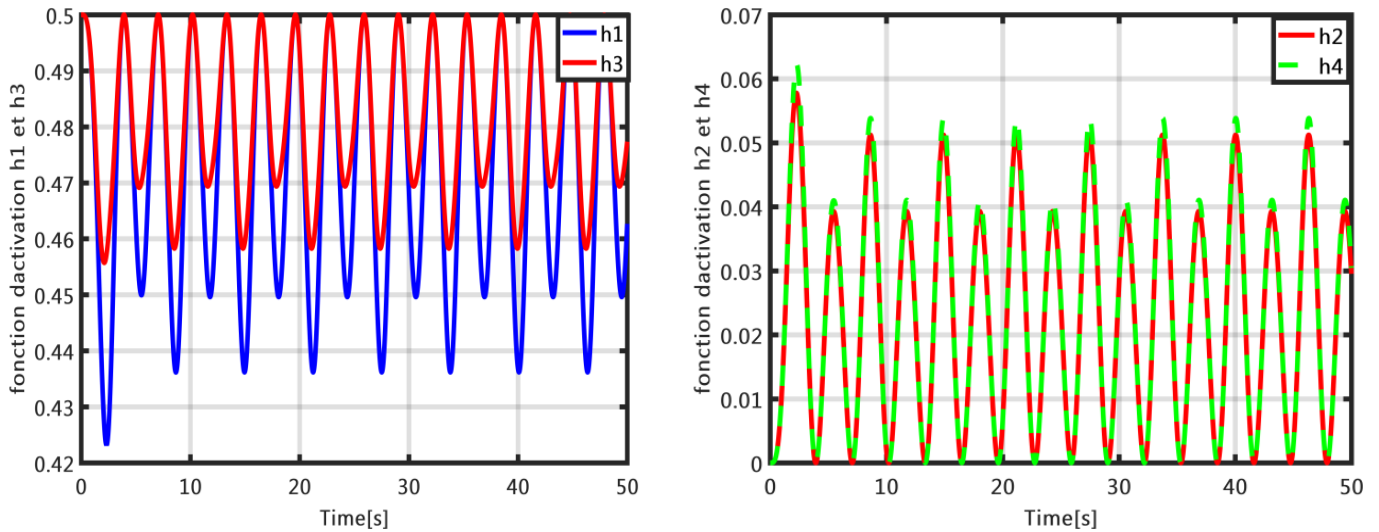
$$\begin{cases} h_1(z(t)) = F_{11}(z_1(t)) \times F_{21}(z_2(t)) \\ h_2(z(t)) = F_{11}(z_1(t)) \times F_{22}(z_2(t)) \\ h_3(z(t)) = F_{12}(z_1(t)) \times F_{21}(z_2(t)) \\ h_4(z(t)) = F_{12}(z_1(t)) \times F_{22}(z_2(t)) \end{cases} \quad (IV-17)$$

Les résultats de la simulation du modèle (T-S) et réel pour $x(0) = \begin{bmatrix} 0 \\ 0 \end{bmatrix}$. Et $u_1(t) = u_2(t) = 0.5 * \sin(t)$ sont montrés dans la Figure IV.4. Il est évident que les deux modèles sont identiques. Cet exemple montre bien que le système non linéaire (IV-1) peut être représenté exactement par un modèle flou de type T-S dans la région $[-1 \ 1] \times [-1 \ 1]$



a. La réponse de l'état $x_1(t)$ b. La réponse de l'état $x_2(t)$

Figure IV.4 Simulation du système (NL) et (TS)



c. Fonction d'activation $h_1(t)$ et $h_3(t)$ d. Fonction d'activation $h_2(t)$ et $h_4(t)$

Figure IV-5 Fonctions d'activations

▪ **Remarque:**

Le nombre de règles des modèles TS obtenus via une transformation poly topique convexe dépend directement du nombre de non-linéarités à découper. Ainsi, lorsque l'on a l termes non linéaires pour un système dynamique donné, un représentant TS obtenu par décomposition en secteurs non linéaires est constitué de 2^l règles floues.

5. Stabilité et Stabilisation et suivi de trajectoire des modèles T-S :

Notre objectif est d'assurer la stabilité des systèmes de Takagi-Sugeno, et nous avons privilégié l'utilisation de la stabilisation quadratique du système par la seconde méthode de Lyapunov.

5.1 Théorème de Lyapunov :

Un modèle $\dot{x}(t) = f(x(t))$ est asymptotiquement stable au voisinage de l'origine si et seulement si, il existe une fonction candidate de Lyapunov V vérifiant [20] :

$$\forall x \in V_X - \{0\}, \frac{dV(x)}{dx} < 0 \quad (\text{IV-18})$$

Pour un certain voisinage V_X de l'origine, la fonction candidate de Lyapunov la plus couramment utilisée est la fonction quadratique donnée par :

$$V(x) = x^T p x, p = p^T > 0 \quad (\text{IV-19})$$

Si on étudie la stabilité du système avec une fonction de Lyapunov de ce type, on parlera de stabilité quadratique.

▪ **Preuve : Le modèle T-S autonome :**

$$\dot{x}(t) = \sum_{i=1}^r h_i(z(t))(A_i x(t)) \quad (IV-20)$$

La stabilité quadratique s'étudie en calculant la dérivée de la fonction V :

$$\frac{dV(x)}{dx} = \frac{d}{dx} (x^T(t) p x(t)) = \dot{x}^T(t) p x(t) + x^T(t) p \dot{x}(t) \quad (IV-21)$$

En utilisant (IV-20) :

$$\frac{dV(x)}{dx} = \left(\sum_{i=1}^r h_i(z(t))(A_i x(t)) \right)^T p x(t) + p x(t) \left(\sum_{i=1}^r h_i(z(t))(A_i x(t)) \right)^T \quad (IV-22)$$

En posant :

$$A_k = \sum_{i=1}^r h_i(z(t))(A_i) \quad (IV-23)$$

On obtient :

$$\frac{dV(x)}{dx} = x(t)^T (A_k^T P + P A_k) x(t) \quad (IV-24)$$

Prouver la stabilité quadratique du modèle (IV-20) revient à résoudre le problème suivant :

- Trouver une matrice $P > 0$ tel que $(A_k^T P + P A_k) < 0 \forall x(t) \in R^n$

Théorème 1[23] : S'il existe $P > 0$ telle que :

$$(A_k^T P + P A_k) < 0 \quad i = 1, \dots, r \quad (IV-25)$$

Alors le système (IV-20) est quadratiquement stable.

5.2 L'approche de la Compensation Parallèle Distribuée (PDC)

Une loi de commande fréquemment utilisée pour stabiliser les modèles T-S est basée sur le concept PDC (Parallel Distributed Compensation). L'idée principale de ce concept est de créer un compensateur pour chaque règle du modèle flou T-S. La loi de commande résultante est non linéaire, en général. La partie conclusion de chaque régulateur local utilise un gain de commande par retour d'état. La Figure IV.5 montre le principe de régulateur PDC.

Le régulateur flou PDC utilise la même base de règle que le système flou. La $i^{\text{ème}}$ règle du régulateur flou s'écrit :

$$\begin{cases} \text{Si } z_1(t) \text{ est } F_1^i(z_1(t)) \text{ ET } \dots \dots z_p(t) \text{ est } F_p^i(z_p(t)) \\ \text{alors } U(t) = -K_i x(t), i = 1, \dots, r \end{cases} \quad (IV-26)$$

Où K_i sont les gains de retour d'état local. L'inférence du régulateur flou global est donnée par :

$$U(t) = - \sum_{i=1}^r h_i(z(t)) K_i x(t), \quad i = 1, \dots, r \quad (\text{IV-27})$$

La stabilisation des modèles T-S par une loi de commande PDC se base sur la fonction de Lyapunov pour montrer la convergence du modèle en boucle fermée, en l'occurrence déterminer les retours d'états correspondants à chaque modèle local composant le modèle T-S. Ainsi, à partir d'une telle fonction, on peut trouver une matrice définie positive P commune entre tous les modèles locaux en boucle fermée.

Pour obtenir le modèle en boucle fermée, on applique la commande PDC au modèle TS (IV-20). Ainsi le modèle obtenu est le suivant :

$$\dot{x}(t) = \sum_{i=1}^r \sum_{j=1}^r h_i(z(t)) h_j(z(t)) (A_i - B_i K_j) x(t) \quad (\text{IV-28})$$

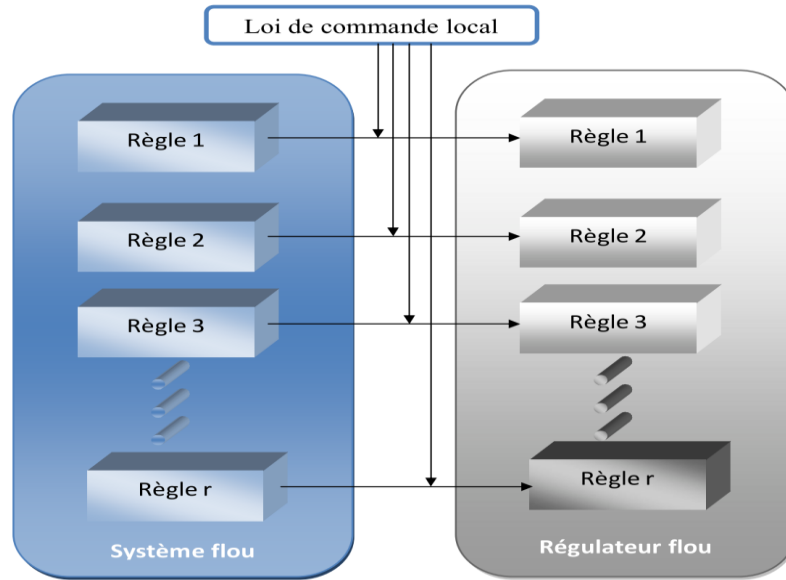


Figure IV.5 Principe des régulateurs flous PDC

Posant $G_{ij} = (A_i - B_i K_j)$

L'équation (IV-28) peut s'écrire comme suit :

$$\dot{x}(t) = \sum_{i=1}^r \sum_{j=1}^r h_i(z(t)) h_j(z(t)) G_{ij} x(t) \quad (\text{IV-29})$$

▪ **Remarque:**

Le premier résultat concernant la stabilisation des modèles TS par la commande PDC a été donné par Wang et al [24] [27] énoncé dans le théorème suivant :

Théorème 2: L'équilibre de boucle fermée continue du système flou (IV-29) est asymptotiquement stable s'il existe une matrice symétrique $P > 0$, une matrice diagonale F et des matrices Q_{ij} avec $Q_{ii} = Q_{ii}^T$ et $Q_{ji} = Q_{ij}^T$ pour $i \neq j$ tel que :

$$G_{ii}^T P + P G_{ii} + Q_{ii} + F P F < 0, i = 1, \dots, r \quad (\text{IV-30})$$

$$\frac{(G_{ij} + G_{ji})^T}{2} P + \frac{(G_{ij} + G_{ji})}{2} P + Q_{ij} \leq 0 \quad i < j \leq r \quad (\text{IV-31})$$

$$\begin{bmatrix} Q_{11} & \cdots & Q_{1r} \\ \vdots & \ddots & \vdots \\ Q_{r1} & \cdots & Q_{rr} \end{bmatrix} > 0 \quad (\text{IV-32})$$

Pour $i, j = 1, \dots, r$, S.T. les paires (i, j) de sorte que :

▪ **Remarque**

Si le nombre de règles est assez grand, il est difficile de trouver la matrice commune P . L'idée est de relâcher les conditions de stabilité et elle est basée sur les propriétés suivantes[24] [27]:

$$\begin{cases} \sum_{i=1}^r h_i^2(z(t)) - \frac{1}{r-1} \sum_{i < j}^r 2h_i(z(t))h_j(z(t)) \geq 0 \\ \sum_{i=1}^r h_i^2(z(t)) = 1, h_i(z(t)) \geq 0 \end{cases}$$

Vérifiée par :

$$\sum_{i=1}^r h_i^2(z(t)) - \frac{1}{r-1} \sum_{i < j}^r 2h_i(z(t))h_j(z(t)) = \frac{1}{r-1} \sum_{i < j}^r (h_i(z(t)) - h_j(z(t)))^2 \geq 0$$

Les conditions du **théorème 2** peuvent être transformées en un problème équivalent sous forme de LMIs, qui peuvent être résolues efficacement par des outils d'optimisation convexe. C'est pour cela qu'on effectue les changements de variables bijectifs usuels suivants : $P = X^{-1}$ et $K_i = M_i X^{-1}$ et l'utilisation d'une congruence des inégalités (IV-30)(IV-31)(IV-32). Les LMIs en M_i et X peuvent être obtenus [21]:

$$\exists X = X^T > 0, \quad \exists Y_{ii} = Y_{ii}^T, \quad \exists Y_{ij} = Y_{ji}^T, \quad \exists M_i : \quad (\text{IV-33})$$

$$\begin{bmatrix} X A_i^T + A_i X - M_i^T B_i^T - B_i M_i + Y_{ii} & X F^T \\ F X & -X \end{bmatrix} < 0 \quad (\text{IV-34})$$

$$X A_i^T + A_i X + X A_j^T + A_j X - B_i M_j - M_j^T B_i^T - M_i^T B_j^T - B_j M_i + 2Y_{ij} \leq 0, \quad j < i \leq r \quad (\text{IV-35})$$

$$\begin{bmatrix} Y_{11} & \cdots & Y_{1r} \\ \vdots & \ddots & \vdots \\ Y_{1r} & \cdots & Y_{rr} \end{bmatrix} > 0 \quad (\text{IV-36})$$

▪ **Exemple :**

Dans ce qui suit nous reprenons l'exemple de la section précédente afin de prouver la stabilité des quadratiques des systèmes TS développés. Pour stabiliser le système dans son point d'équilibre, on utilise une loi de commande PDC donnée dans l'équation (IV-27):

La résolution des LMIs (IV-35), nous donne les résultats suivants :

$$P_{stab} = \begin{bmatrix} 0.2738 & -0.0008 \\ -0.0008 & 0.0896 \end{bmatrix}$$

$$K_1 = [-8.0301 \quad 0.0238]$$

$$K_2 = [-4.6341 \quad 0.0137]$$

$$K_3 = [-4.8239 \quad 0.0143]$$

$$K_4 = [-3.3278 \quad -2.114]$$

La figure suivante montre la stabilité asymptotique des états $x_1(t)$ et $x_2(t)$ pour les conditions initiales $x_0 = [0.5 \ 0.5]^T$

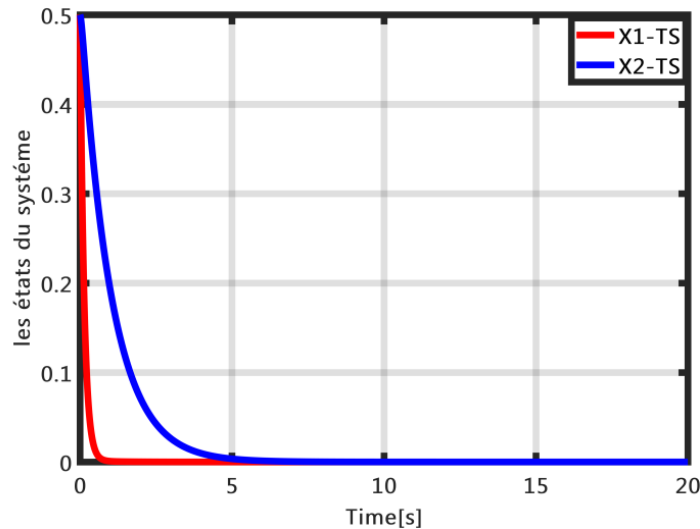


Figure IV.6 a stabilisation du l'état $x_1(t)$ et $x_2(t)$

5.3 Suivi de trajectoire des modèles T-S :

Dans cette section, nous présentons la loi de commande qui garantit le suivi de trajectoire ainsi que l'analyse de la stabilité du suivi de trajectoire, nous supposons que tous les états sont mesurables.

Le problème de suivi de trajectoire peut être considéré comme une généralisation du problème de stabilisation.

Considérons le système décrit par [28] :

$\begin{cases} \dot{x}(t) = f(x(t)) + g(x(t))u(t) \\ y(t) = h(x(t)) \\ \bar{y}(t) = \varphi(x(t)) \end{cases}$	(IV-37)
--	---------

Où $x(t) \in \mathbb{R}^n$ est le vecteur d'état, $u(t) \in \mathbb{R}^m$ est le vecteur de commande, $y(t), \bar{y}(t) \in \mathbb{R}^q$ sont respectivement les vecteurs de sorties mesurées et contrôlées, $f(x), g(x), h(x), \varphi(x)$ et $\theta(x)$ sont des fonctions non linéaires avec une dimension appropriée.

Le système non-linéaire (IV-37) peut être exprimé par le modèle flou suivant :

$$\left\{ \begin{array}{l} \text{Si } z_1(t) \text{ est } F_{1i}(z_1(t)) \text{ ET } z_2(t) \text{ est } F_{2i}(z_2(t)) \dots \dots z_g(t) \text{ est } F_{gi}(z_g(t)) \\ \text{Alors} \quad \begin{cases} \dot{x}(t) = A_i x(t) + B_i u(t) \\ y(t) = C_i x(t) + D_i u(t) \end{cases} \end{array} \right. \quad (\text{IV-38})$$

Où :

$F_{gi}(z_g(t))$ Pour $i = 1, 2, \dots, g$ sont les sous-ensembles flous, r le nombre de règles floues, $z_g(t)$ sont les variables de prémisses qui dépendent de l'entrée et/ou de l'état du système, $A_i \in \mathbb{R}^{n \times n}$ est la matrice d'état, $B_i \in \mathbb{R}^{n \times m}$ est la matrice de commande.

5.3.1 Synthèse de la loi de commande :

L'objectif d'un contrôleur est de satisfaire la condition suivante:

$$y(t) - r(t) \rightarrow 0, t \rightarrow \infty$$

Où: $r(t)$ représente la trajectoire désirée ou le signal de référence. Afin de convertir le problème de suivi de trajectoire en un problème de stabilisation, on introduit un ensemble de variables virtuelles désirées (VDVs) x_d qui doivent être suivies par les variables d'états x telle que

$$\begin{aligned} \bar{y}(t) &= \varphi(x) \\ r(t) &= \varphi(x_d) \end{aligned}$$

On posant $\tilde{x}(t) = x(t) - x_d(t)$ qui désigne l'erreur de poursuite, la dérivée de $\tilde{x}(t)$ s'écrit comme suit :

$$\begin{aligned} \dot{\tilde{x}}(t) &= \dot{x}(t) - \dot{x}_d(t) \\ \dot{\tilde{x}}(t) &= \sum_{i=1}^r h_i(z(t)) (A_i x(t) + B_i u(t)) - \dot{x}_d(t) \\ \dot{\tilde{x}}(t) &= \sum_{i=1}^r h_i(z(t)) (A_i x(t)) - \sum_{i=1}^r h_i(z(t)) (A_i x_d(t)) + \sum_{i=1}^r h_i(z(t)) (A_i x_d(t)) \\ &\quad + \sum_{i=1}^r h_i(z(t)) (B_i u(t)) - \dot{x}_d(t) \end{aligned}$$

En posant :

$$\sum_{i=1}^r h_i(z(t)) (B_i u(t)) = \sum_{i=1}^r h_i(z(t)) (B_i u(t)) + \sum_{i=1}^r h_i(z(t)) (A_i x_d(t)) - \dot{x}_d(t) \quad (\text{IV-39})$$

Alors on peut écrire l'erreur de poursuite comme suit :

$$\dot{\tilde{x}}(t) = \sum_{i=1}^r h_i(z(t)) (A_i \tilde{x}(t) + B_i \omega(t)) \quad (\text{IV-40})$$

Selon la description ci-dessus, on constate que le problème de suivi $\tilde{x}(t)$ de trajectoire est similaire au problème de la stabilisation de l'erreur de poursuite. Dans ce qui suit notre but est d'obtenir un contrôleur pour rendre le nouvel état $\tilde{x}(t) \rightarrow 0$. Alors on cherche à déterminer le nouveau contrôleur flou.

- Règle du contrôleur :

$$\text{Règle } i: \begin{cases} \text{Si } z_1(t) \text{ est } F_{1i}(z_1(t)) \text{ ET } z_2(t) \text{ est } F_{2i}(z_2(t)) \dots \dots z_g(t) \text{ est } F_{gi}(z_g(t)) \\ \text{Alors: } \omega(t) = -M_i \tilde{x}(t) \end{cases} \quad i = 1, 2, \dots, r$$

Où M_i sont les gains de retour d'état. L'inférence du régulateur flou global est donnée par:

$$\omega(t) = - \sum_{i=1}^r h_i(z(t)) M_i \tilde{x}(t) \quad (\text{IV-41})$$

En utilisant (IV-39) et (IV-40) on obtient le système en boucle fermée suivant :

$$\dot{\tilde{x}}(t) = \sum_{i=1}^r \sum_{j=1}^r h_i(z(t)) h_j(z(t)) G_{ij} \tilde{x}(t) \quad (\text{IV-42})$$

Avec :

$$G_{ij} = A_i - B_i K_i$$

▪ *Remarque:*

Les conditions de stabilité pour le suivi de trajectoire sont les mêmes que le problème de stabilisation des modèles flous TS par une loi de commande type PDC. Cela signifie que les gains de retour d'état K_i peuvent être obtenus en faisant directement la résolution du problème de stabilisation.

Il nous reste alors qu'à déterminer les variables désirées $x_d(t)$ et obtenir ensuite la loi de commande $u(t)$. Pour atteindre cet objectif, on utilise l'expression suivante [29]:

$$g(x) = \sum_{i=1}^r h_i(z(t)) B_i$$

Et on réécrit (IV-39) comme suit:

$$g(x)(u(t) - \omega(t)) = -A(x)x_d(t) + \dot{x}_d(t) \quad (\text{IV-43})$$

Avec:

$$A(x) = \sum_{i=1}^r h_i(z(t)) A_i$$

L'existence de la commande $u(t)$ dépend de la forme $\deg(x)$. La matrice d'entrée $g(x)$ est supposée de plein rang colonne.

De (IV-43) on peut déduire la commande $u(t)$ comme suit:

$$u(t) = - \sum_{i=1}^r h_i(z(t)) M_i \tilde{x} + B^{-1}(x) [\dot{x}_d(t) - A(x)x_d] \quad (\text{IV-44})$$

➤ **Reprenons l'exemple précédent :**

Les variables désirées x_d sont nécessaires pour satisfaire (IV-43), qui est réécrite ci-dessous:

$$g(x)(u(t) - \omega(t)) = -A(x)x_d(t) + \dot{x}_d(t)$$

On obtient alors la forme matricielle suivante:

$$\begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix} (u(t) - \omega(t)) = - \begin{bmatrix} -1 & x_1(t)x_2^2(t) \\ (3 + x_2(t))x_1^2(t) & -1 \end{bmatrix} \begin{bmatrix} x_{1d}(t) \\ x_{2d}(t) \end{bmatrix} + \begin{bmatrix} \dot{x}_{1d}(t) \\ \dot{x}_{2d}(t) \end{bmatrix}$$

Avec:

$$u(t) = [u_1(t) \quad u_2(t)], \quad \omega(t) = [\omega_1(t) \quad \omega_2(t)]$$

$\omega_1(t)$, $\omega_2(t)$ sont les nouvelles commandes, La commande $\omega(t)$ est donnée par :

$$\omega(t) = \begin{bmatrix} \omega_1(t) \\ \omega_2(t) \end{bmatrix} = - \sum_{i=1}^r h_i(z(t)) M_i (x(t) - x_d(t))$$

La résolution des LMIs (IV-35), nous a donné les résultats suivants :

$$P_{suivi} = \begin{bmatrix} 0.2738 & -0.006 \\ -0.006 & 0.0296 \end{bmatrix}$$

$$M_1 = [5.0301 \quad 0.0238]$$

$$M_2 = [3.6341 \quad 0.0137]$$

$$M_3 = [4.8239 \quad -0.0143]$$

$$M_4 = [-3.3278 \quad -2.114]$$

Les résultats de la simulation de suivi de trajectoire de l'état $x_1(t)$ et $x_2(t)$ sont montrés sur la Figure IV.7.

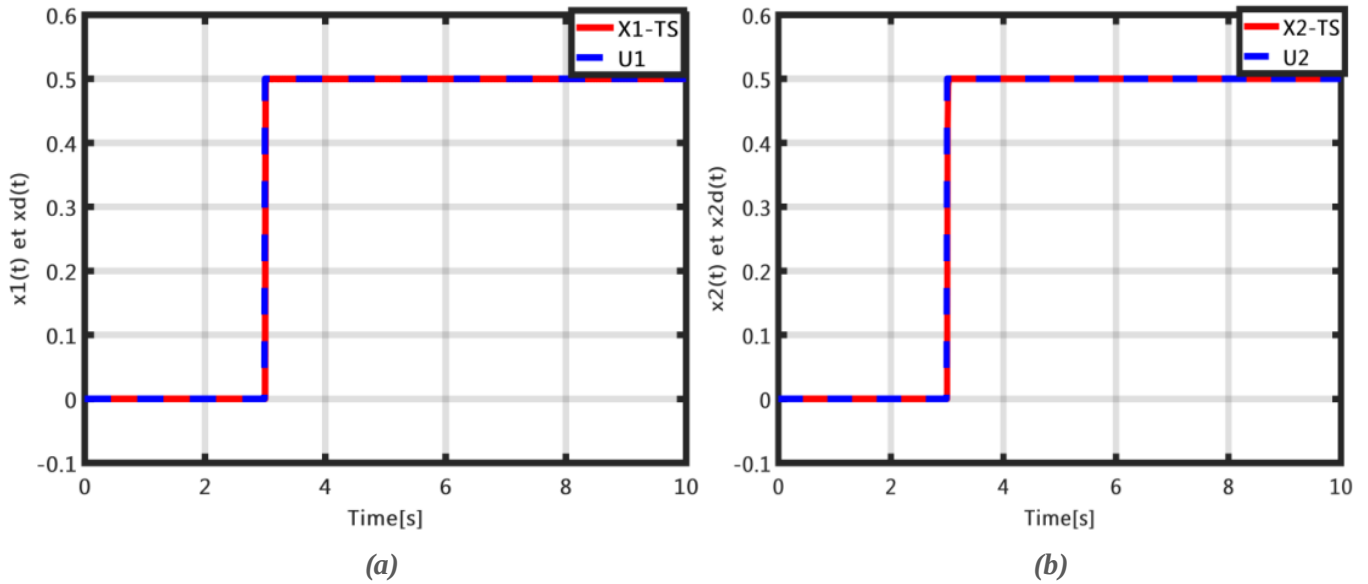


Figure IV.7 Réponse de l'état $x_1(t)$ et $x_2(t)$

5.4 Application de la technique PDC au quadri-rotor :

5.4.1 Stratégie de commande :

Dans cette partie nous exposons l'application de la technique de commande PDC présenté au début de ce chapitre pour la commande du vol d'un drone. La dynamique de ce modèle a été présentée dans le chapitre I. La stratégie de commande adoptée est résumée en deux sous-systèmes en cascade; le premier est lié au contrôle de la rotation, et le deuxième est lié au contrôle de translation, comme il est donné sur le schéma synoptique suivant [29]:

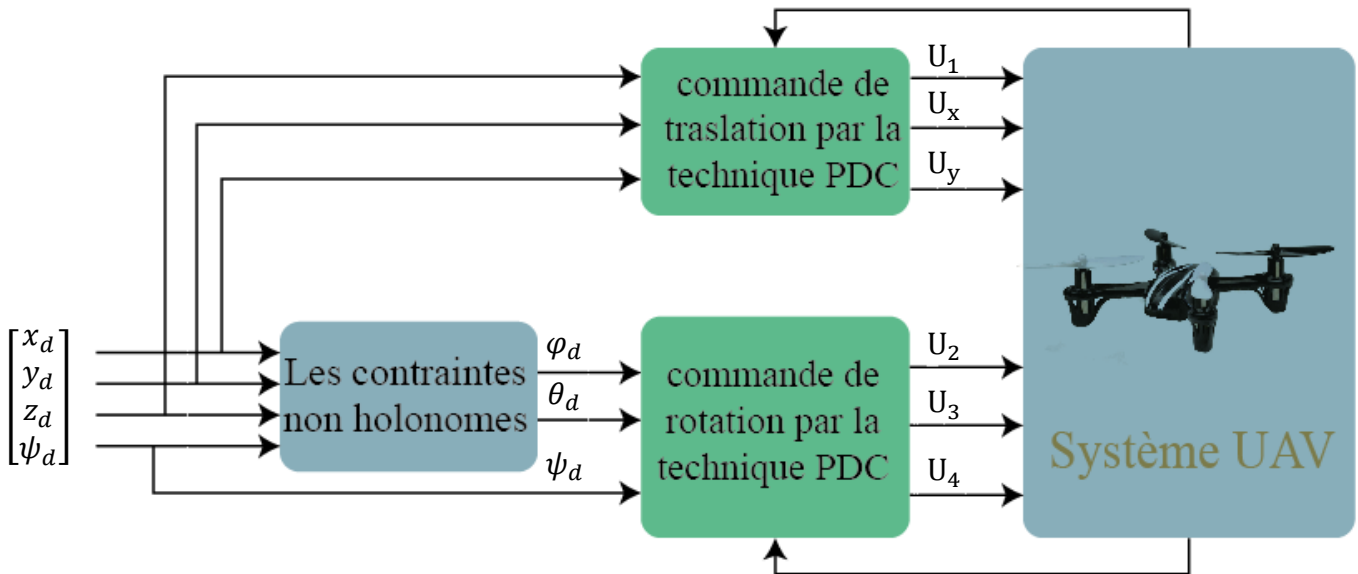


Figure IV.8 Schéma synoptique de commande

La résolution des LMIs donnés en (IV-35), nous donne les résultats suivants :

a) Pour le sous système d'orientation :

$$\begin{aligned}
M_1 &= \begin{bmatrix} 0.1954 & 0.0738 & 0.0004 & -0.0001 & 0 & 0 \\ -0.0017 & 0.0001 & 0.3730 & 0.1398 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0.6291 \end{bmatrix} M_2 \\
&= \begin{bmatrix} 0.19 & 0.078 & 0.0004 & -0.0005 & 0 & 0 \\ -0.0022 & -0.0003 & 0.346 & 0.1297 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0.5615 \end{bmatrix} \\
M_3 &= \begin{bmatrix} 0.1821 & 0.0688 & 0.0012 & 0.0002 & 0 & 0 \\ -0.001 & 0.0001 & 0.3163 & 0.1186 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0.5615 \end{bmatrix} \\
M_4 &= \begin{bmatrix} 0.1753 & 0.0662 & -0.0006 & -0.0005 & 0 & 0 \\ -0.003 & 0.0006 & 0.2861 & 0.1072 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0.5193 \end{bmatrix} \\
M_5 &= \begin{bmatrix} 0.1792 & 0.0677 & -0.0006 & -0.0005 & 0 & 0 \\ 0.0020 & -0.0003 & 0.2981 & 0.1117 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0.5407 \end{bmatrix} \\
M_6 &= \begin{bmatrix} 0.1739 & 0.0657 & -0.0004 & -0.0004 & 0 & 0 \\ -0.0017 & -0.0002 & 0.2779 & 0.1042 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0.5193 \end{bmatrix} \\
M_7 &= \begin{bmatrix} 0.1738 & 0.0657 & -0.0004 & -0.0004 & 0 & 0 \\ -0.0017 & -0.0002 & 0.2779 & 0.1142 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0.4193 \end{bmatrix} \\
M_8 &= \begin{bmatrix} 0.1718 & 0.0649 & 0.0024 & 0.0006 & 0 & 0 \\ 0 & 0.0004 & 0.1639 & 0.0989 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0.1718 \end{bmatrix}
\end{aligned}$$

Avec:

$$P_{Orientation} = \begin{bmatrix} 0.0055 & 0.0051 & 0 & 0 & 0 & 0 \\ 0.0051 & 0.0093 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0.0037 & 0.0038 & 0 & 0 \\ 0 & 0 & 0.0038 & 0.0087 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0.5693 & 0.1887 \\ 0 & 0 & 0 & 0 & 0.1887 & 0.0743 \end{bmatrix}$$

b) Pour le sous système de translation :

$$\begin{aligned}
M_1 &= \begin{bmatrix} 4.0480 & 7.3539 & 0 & 0 & 0 & 0 \\ 0 & 0 & 2.9161 & 6.7154 & 0 & 0 \\ 0 & 0 & 0 & 0 & 966.3292 & 380.3663 \end{bmatrix} \\
M_2 &= \begin{bmatrix} 3.9820 & 7.2339 & 0 & 0 & 0 & 0 \\ 0 & 0 & 2.8241 & 6.5035 & 0 & 0 \\ 0 & 0 & 0 & 0 & -3.29.068 & -129.527 \end{bmatrix} \\
M_3 &= \begin{bmatrix} 3.9563 & 7.1871 & 0 & 0 & 0 & 0 \\ 0 & 0 & 2.8241 & 6.5035 & 0 & 0 \\ 0 & 0 & 0 & 0 & -591.05 & -232.6491 \end{bmatrix}
\end{aligned}$$

$$M_4 = \begin{bmatrix} 3.7480 & 6.7529 & 0 & 0 & 0 & 0 \\ 0 & 0 & 2.5916 & 5.9154 & 0 & 0 \\ 0 & 0 & 0 & 0 & 257.3292 & 101.3963 \end{bmatrix}$$

$$M_5 = \begin{bmatrix} 0.29820 & 0.2339 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0.2024 & 10.48035 & 0 & 0 \\ 0 & 0 & 0 & 0 & 437.418 & 172.177 \end{bmatrix}$$

$$M_6 = \begin{bmatrix} 0.2563 & 0.51871 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0.2824 & 10.5035 & 0 & 0 \\ 0 & 0 & 0 & 0 & -217.05 & -85.6491 \end{bmatrix}$$

$$M_7 = \begin{bmatrix} 0.285 & 0.5190 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0.1997 & 0.4600 & 0 & 0 \\ 0 & 0 & 0 & 0 & -378.177 & -148.8582 \end{bmatrix}$$

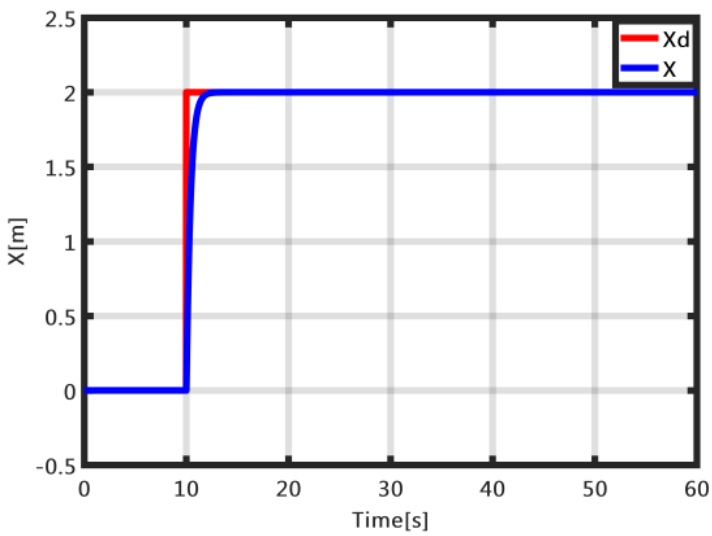
$$M_8 = \begin{bmatrix} 0.3396 & 0.54 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0.2353 & 0.4707 & 0 & 0 \\ 0 & 0 & 0 & 0 & 78.8527 & 32.0119 \end{bmatrix}$$

Avec :

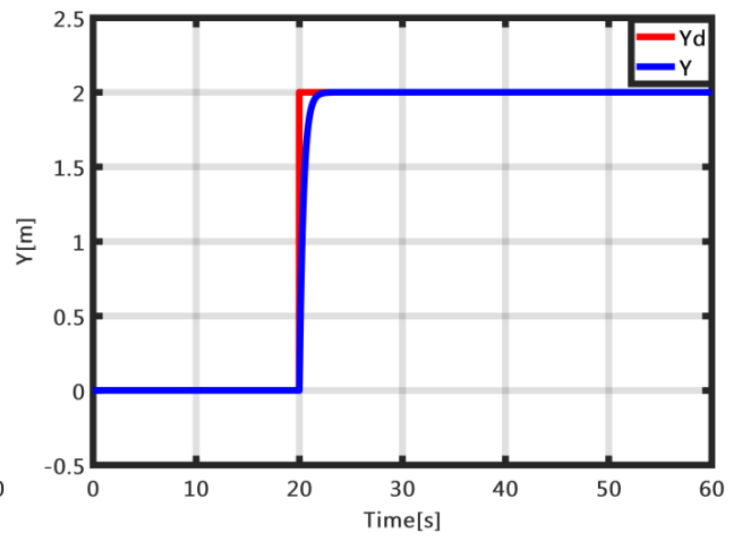
$$P_{translation} = \begin{bmatrix} 0.0045 & 0.00321 & 0 & 0 & 0 & 0 \\ 0.00321 & 0.0093 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0.0037 & 0.0218 & 0 & 0 \\ 0 & 0 & 0.0218 & 0.0087 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0.5093 & 0.1087 \\ 0 & 0 & 0 & 0 & 0.1087 & 0.043 \end{bmatrix}$$

6. Résultats de simulation :

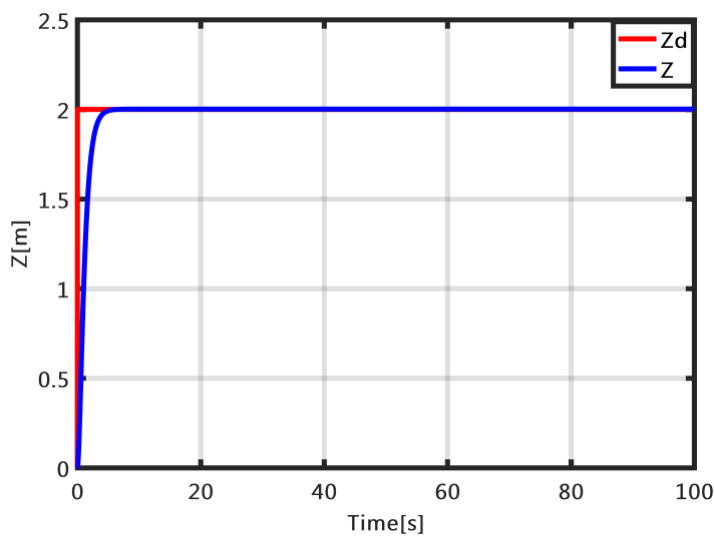
Dans cette simulation nous avons utilisé, pour calculer les différentes lois de commande, l'ensemble des paramètres du modèle du quadri-rotor cités dans le premier chapitre, donné précédemment. Nous avons également considéré des trajectoires de référence en échelon.



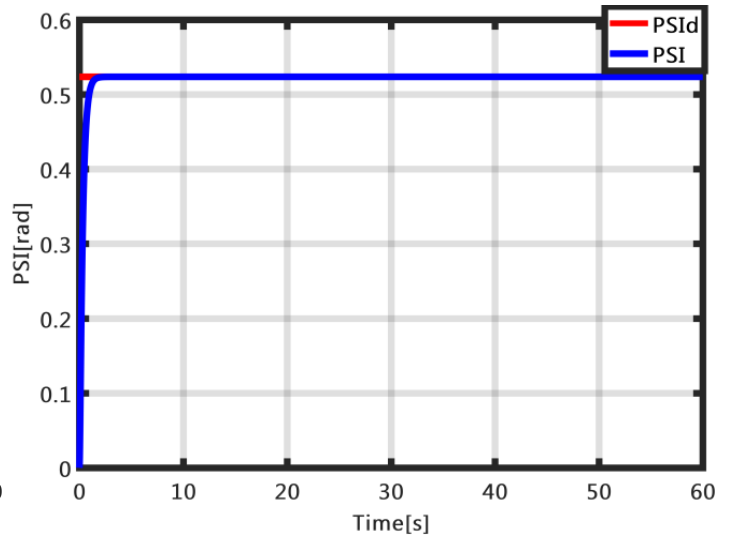
(a)



(b)



(c)



(d)

Figure IV.9 Résultats de poursuite, dans le cas des trajectoires de références en échelon, selon les axes (X, Y, Z, ψ)

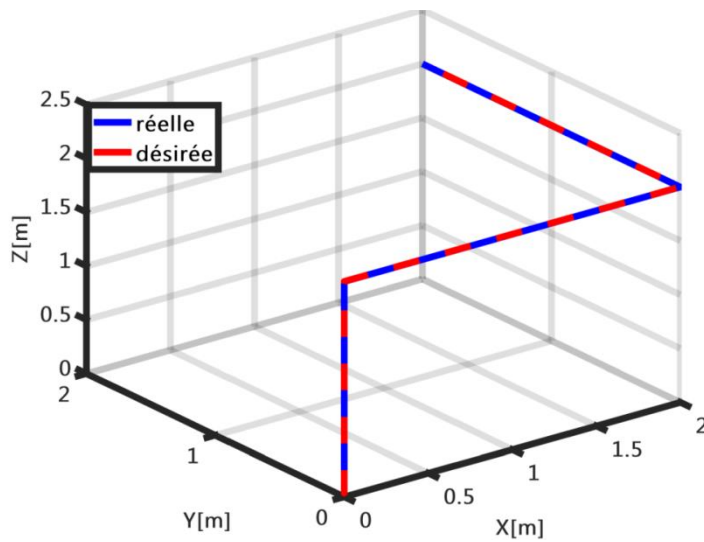
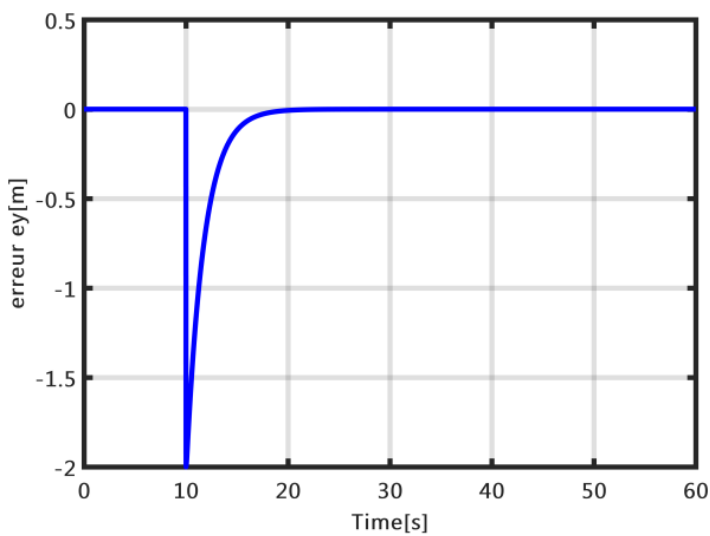
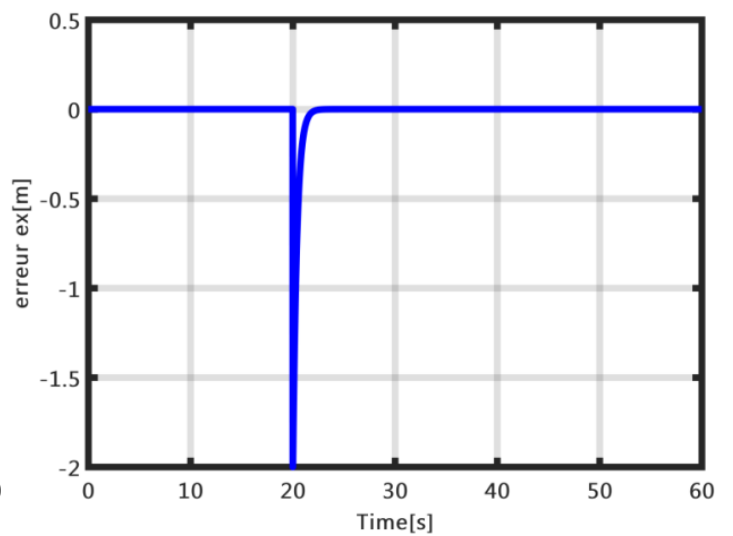


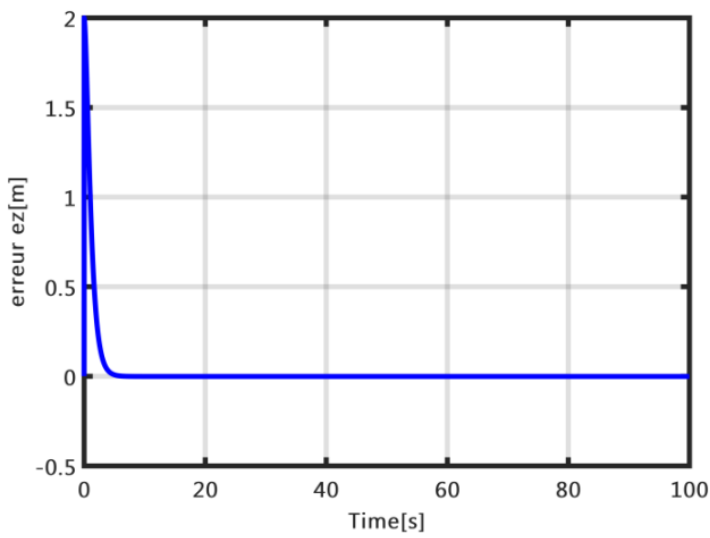
Figure IV.10 Trajectoire globale en 3D



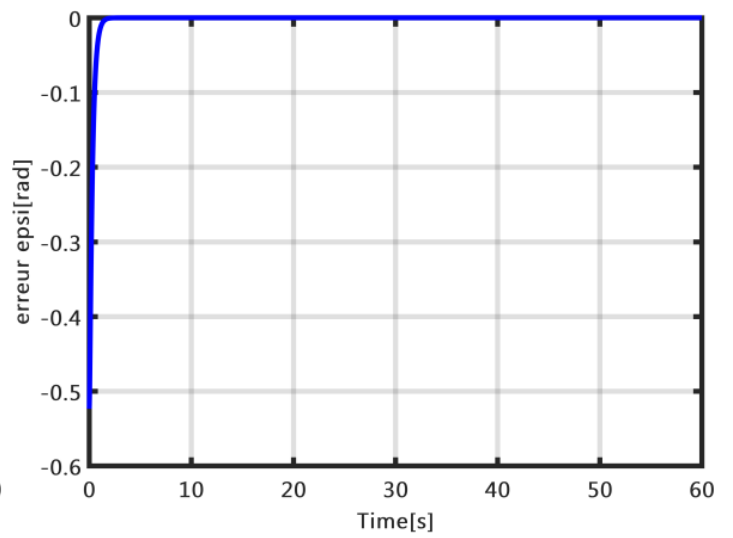
(a)



(b)

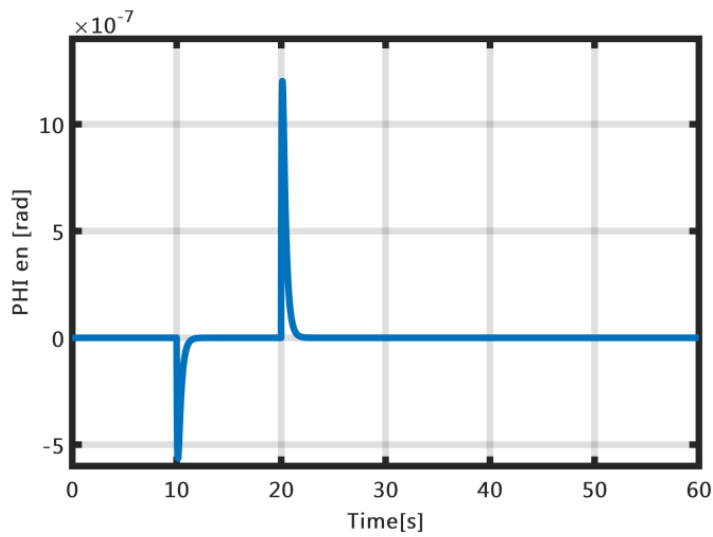


(c)

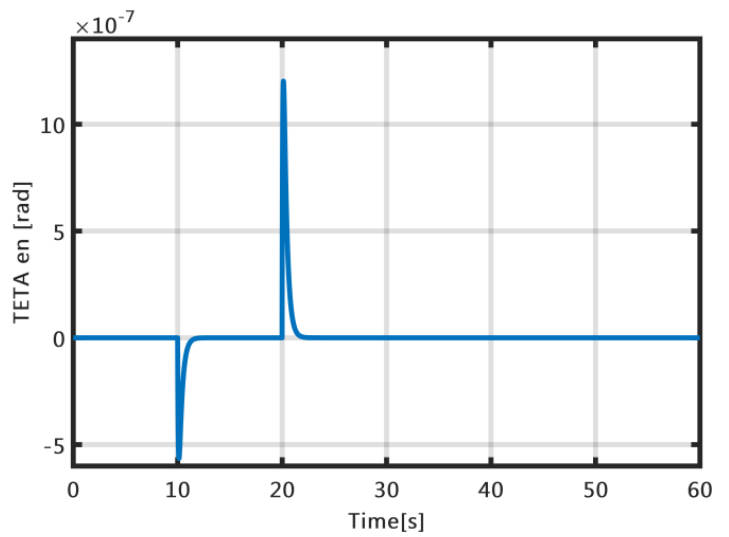


(d)

Figure IV.11 les erreurs de déplacement selon les 3 axes

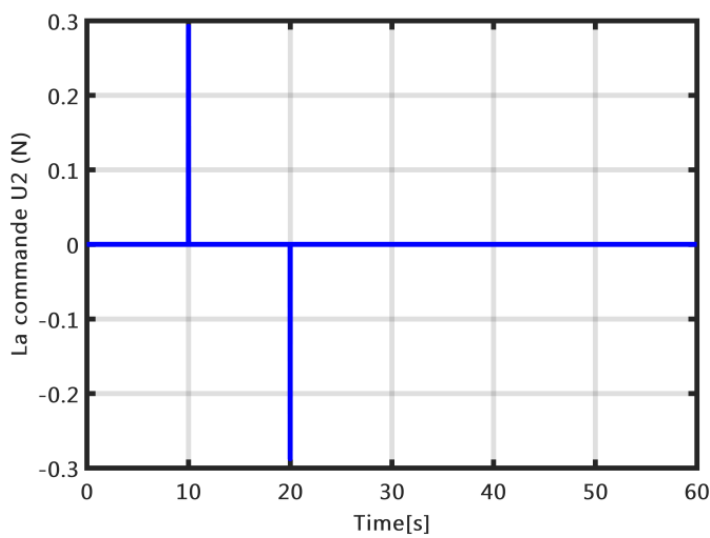


(a)

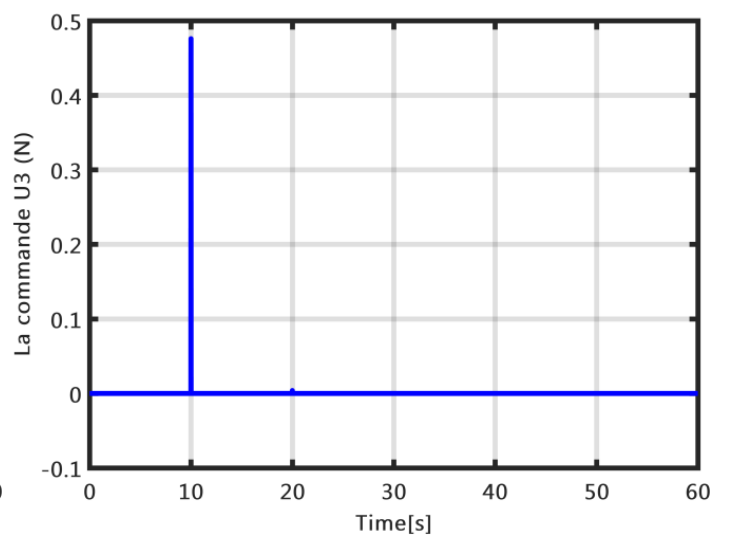


(b)

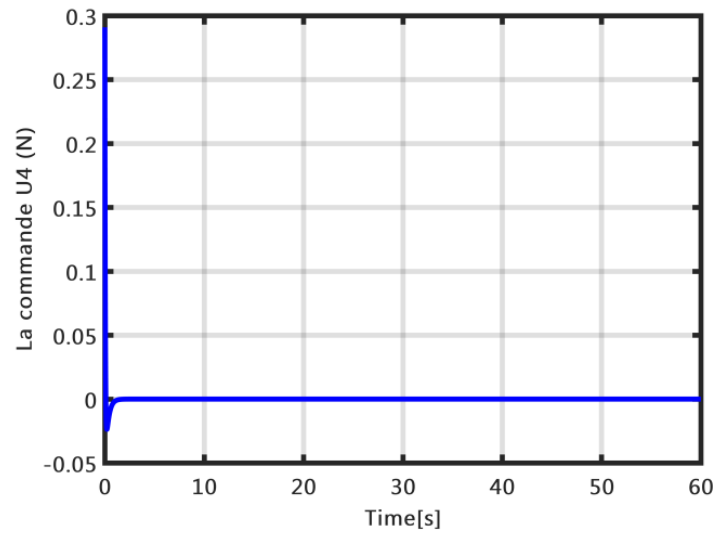
Figure IV.12 L'angle de tangage et de roulis



(a)

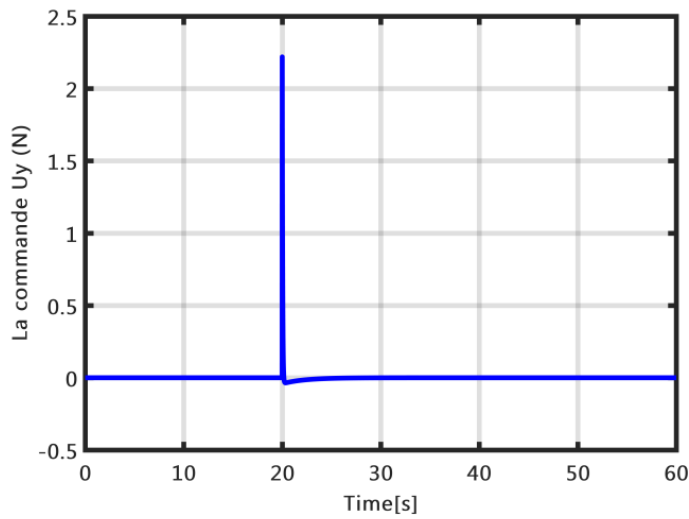


(b)

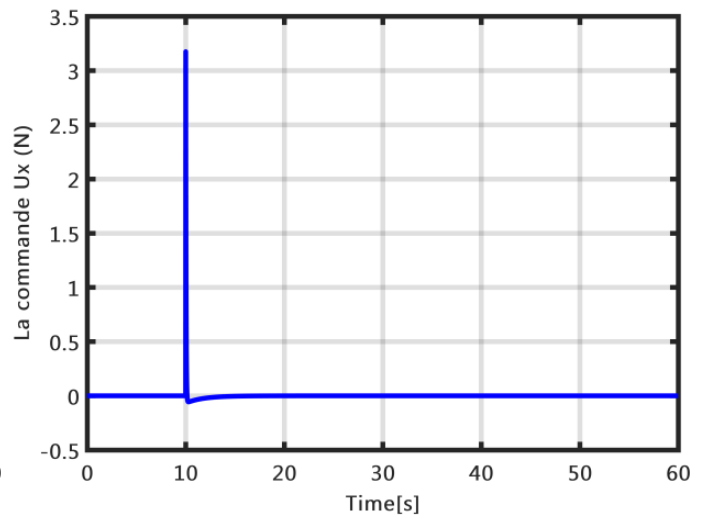


(c)

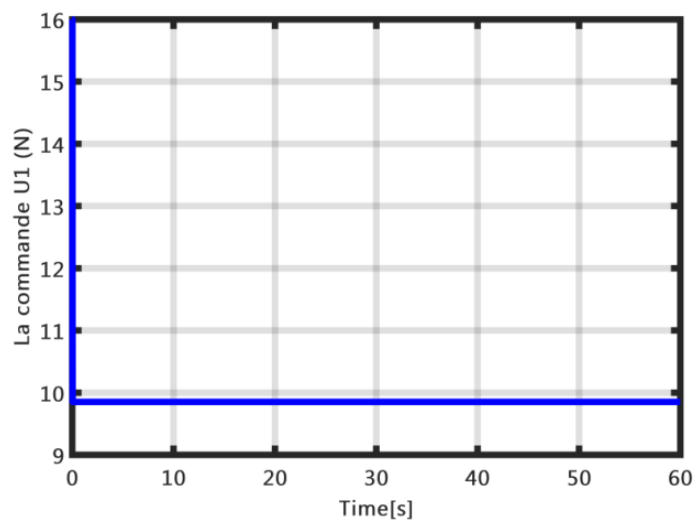
Figure IV.13 signaux de commande des rotations



(a)



(b)



(c)

Figure IV.14 signaux de commande de translation

Nous observons une poursuite quasi parfaite sur toutes les variables étudiées et une rapidité de la réponse avec un temps de réponse comme suit :

- Le temps de réponse en Z est de 2.1s.
- Le temps de réponse en X est de 1.55s.
- Le temps de réponse en Y est de 1.55s.
- Le temps de réponse en ψ est de 0.9s.

6.1 Interprétation des résultats :

Les figures (IV.9) à (IV.14) montrent les performances de commande obtenues dans le cas des trajectoires en échelon. Le drone doit effectuer un mouvement vertical de 2m puis un mouvement suivant x d'une distance de 2m ensuite un mouvement suivant y de la même distance. Nous remarquons que, dans ce cas, la poursuite s'effectue sans aucun dépassement (Figure IV.9 (a)– (d)). Les figures (VI.13) et (VI.14) montrent les signaux de commande obtenus. Figure IV.11 montre que l'erreur suivant les trois axes converge vers zéro. Cela prouve que la trajectoire parcourue par le drone suit la trajectoire désirée. À partir de la Figure IV.14, nous remarquons que la commande U_1 dépasse les limites physiques des rotors du drone pour cela nous l'avons saturée. La somme de ces forces satisfait la condition d'équilibre ($\sum_i^4 F_i = m \cdot g$)

7. Comparaison entre les différentes commandes étudiées:

Commande par PID	Commande par linéarisation E/S	Commande par PDC
Système lent et présente un important dépassement.	Système lent mais précis et ne présente pas de dépassement	Système rapide et précis et ne présente pas de dépassement

8. Conclusion :

Dans ce chapitre, nous avons présenté essentiellement le concept de la linéarisation par secteurs non linéaires puis nous avons utilisé le modèle découplé en deux sous-systèmes (horizontal et vertical) de quadri-rotor, donné dans le chapitre précédent, et en se basant sur la technique de commande PDC, nous avons développé pour chaque sous-système une loi de commande. Les résultats de simulation des lois de commande développées et appliquées sur le modèle du quadri-rotor ont montré la stabilité du système et aussi une bonne poursuite de trajectoire. Cela prouve l'efficacité de la stratégie de commande proposée.

Chapitre V: Réalisation du quadri-rotor

Chapitre V: Réalisation du quadri-rotor

1. Introduction:

Après l'utilisation des drones dans le domaine militaire, les drones civils ont vu le jour et intègrent les différentes technologies mises au point depuis leur naissance. La pluparts de ces drones sont commandés manuellement avec une radiocommande ou avec le téléphone en utilisant les applications android. L'usage de ce type d'engin volant est très limité. Dans le cadre de notre projet de fin d'étude, nous nous intéressons à la commande à distance des quadri-rotor basés sur l'une des approches de commande citées dans les chapitres précédents. L'ampleur de ce travail est d'envoyer la trajectoire désirée au quadri-rotor afin que ce dernier puisse la suivre et faire des vols automatiques en utilisant le logiciel matlab/simulink.

Ce travail sera réalisé en plusieurs étapes. Nous commencerons par la présentation des éléments indispensables pour la réalisation du quadri-rotor et présenter le kit HJ Alien X-500. Par la suite, nous remplacerons la carte de contrôle préprogrammée avec la carte ARDUINO reliée à un ordinateur qui constitue la station de base du quadri-rotor. Avant d'injecter l'une des commandes étudiées au préalable, nous effectuerons un test basique pour la commande des moteurs via matlab/simulink. Dans la quatrième étape, nous ajouterons le capteur et la centrale inertielle MPU6050 afin de faire la régulation et ainsi stabiliser le quadri-rotor à une altitude désirée et contrôler le cap et l'attitude.

2. Les éléments indispensables :

2.1 Composants mécaniques

2.1.1 Châssis

C'est la structure qui va comporter tous les éléments du quadri-rotor. C'est de ce châssis que dépend le comportement du drone. Un châssis ultra léger permet de gagner en autonomie. Des bras longs amélioreront la stabilité alors que des bras plus courts permettront des figures plus acrobatiques.

Les quadri-rotors sont dotés de quatre bras, en forme de « + » ou en « x » (Figure I.1), la seule différence entre ces deux modèles est le fait que celui en croix a une vue plus dégagée à l'avant et à l'arrière. De ce fait, il réside une différence de la manière d'actionner les moteurs.

Les matériaux utilisés pour la construction des châssis sont la fibre de carbone, l'aluminium, le bois, le métal et le plastique. Ce dernier est beaucoup utilisé en raison de sa polyvalence, poids léger, prix bas et solidité.

2.1.2 Hélices

L'hélice est un organe mécanique entraîné par le moteur et assure la propulsion de l'aéronef. Lorsqu'elle se met en rotation, les pales forment un disque rotor. C'est alors leur forme aérodynamique qui produit entre les deux faces du disque une différence de pression perpendiculaire au disque et centrée sur son axe [30].

La poussée produite par une hélice dépend de la densité de l'air, de son diamètre, de la forme et de la surface des pales et de son pas. C'est pourquoi il est très judicieux de choisir une hélice adéquate à notre quadri-rotor selon le type de vol que nous voulons réaliser.

La figure ci-dessous (Figure V.1) illustre les deux caractéristiques de l'hélice et leurs effets sur le comportement du quadri-rotor.

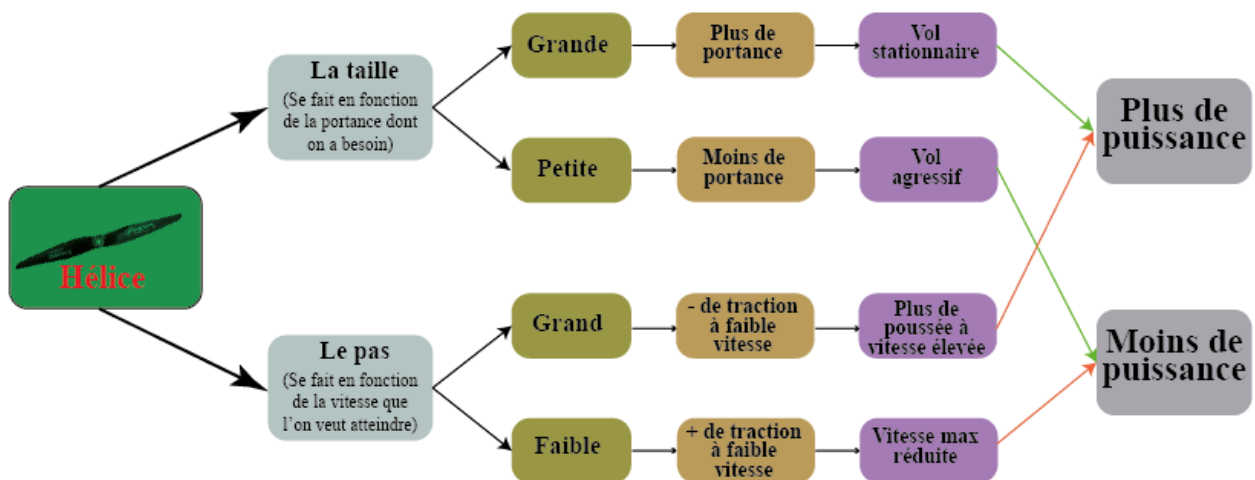


Figure V.1 Caractéristiques de l'hélice

En plus de ça, il faut penser à équilibrer les hélices. Une hélice non équilibrée crée des vibrations supplémentaires qui sont à éviter. Pour en finir, les hélices ont un sens de rotation, soit dans le sens horaire (CW) soit dans le sens anti horaire (CCW).

Un autre critère sur le choix des hélices est sa composition. Il existe plusieurs matériaux:

- Le plastique: Peu coûteux et durable mais moins rigide.
- Le bois: Robuste mais très cher et difficile à trouver.
- La fibre de carbone: Plus flexible et très disponible mais coûteuse.

2.2 Composants électriques

2.2.1 Moteurs



Figure V.2 Emax MT1806 2280kv

Le moteur électrique brushless est un moteur sans balais qui est utilisé pour faire tourner les hélices. Il est constitué d'aimants permanents sur le rotor et d'un bobinage triphasé sur le stator [31][32].

La caractéristique principale des moteurs brushless est le nombre de KV. C'est une caractéristique physique qui indique la vitesse de rotation d'un moteur. Elle correspond au nombre de tours par minutes pour 1 volt (RPM: rotation par minutes). Plus le KV est grand, plus on a moins de couple.

Ce type de moteur offre un rapport poids/puissance très favorable et ont un rendement proche de 90% et ne demandent pas d'entretien, donc ils ont une durée de vie plus grande du fait par l'absence d'usure des balais ainsi qu'une augmentation de l'efficacité du moteur, de son couple et un meilleur rendement électrique[33].

En résumé, pour choisir un moteur, il faut prendre en compte à la fois le nombre de KV, la puissance et le poids. Le tableau suivant illustre quelques paramètres sur le choix du matériel selon le cahier de charge spécifié.

Spécification Matériel	Vol en immersion (de portée)	Vol nerveux	Vol stable
	Avoir plus d'autonomie	Avoir plus de vitesse	Avoir plus de couple
Moteurs	Faible KV	- KV important (1000 - 1500 KV) - Grande puissance des moteurs	Faible KV (<1000 KV)
Batterie	Batterie plus importante (plus de mAh)	Le "C" des Lipos plus élevé	
Hélice	Grande taille	- Petite taille (8-9") - grand pas (>4.5")	- Grande taille (>9") - petit pas (4-4.5")

Tableau V.1 Choix des moteurs d'un quadri-rotor

2.2.2 ESC

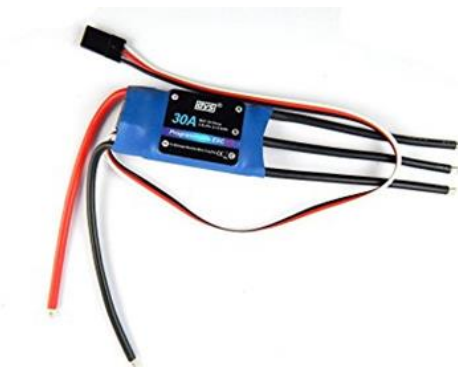


Figure V.3 Contrôleur électronique de vitesse Robo India ESSC30A

« Electronic Speed Controller » ou « *Contrôleur électronique de vitesse* en français»: C'est un dispositif qui permet de commander la vitesse des moteurs brushless en transformant le courant continu de la batterie en impulsion précises pour faire fonctionner les moteurs brushless.

Pour changer le sens de rotation, il suffit de permuter deux quelconques des trois connecteurs entre l'ESC et le moteur sans balais. Pratiquement, on ne devra pas aussi oublier de configurer les ESC avant leur utilisation, cette opération s'appelle "Calibration des ESC". Elle permet de faire travailler les 4 moteurs dans une plage commune à bas régime comme à haut régime [34].

Le choix des contrôleurs électroniques de vitesse se rapporte sur sa caractéristique qui est la puissance qu'il peut fournir (10A, 12A, 20A, 30A etc.), une valeur choisie en fonction du moteur avec une marge pour que l'ESC ne risque pas de bruler.[35]

2.2.3 Batterie

La plupart des batteries utilisées pour la réalisation des quadri-rotors sont de type Lithium Polymères (LiPo). C'est un choix de prédilection en raison d'une meilleure performance comparées aux d'autres types de batterie (NiCd, etc.); elles offrent un meilleur rapport puissance/masse.

Les batteries sont classées selon:

- Le nombre de cellules: 3S, 4S, 5S etc.
- Le taux de décharge qui est le nombre de 'C': 35C, 65C, 75C.
- La capacité en mAh : 850mAh, 1800mAh, 2200mAh etc. plus la capacité est élevée plus le vol est long.

En pratique, les Lipos doivent être capables de fournir en continu l'équivalent du double de la consommation max de nos 4 moteurs.

Cependant, les LiPos présentent quelques inconvénients qui sont le prix et le danger d'explosion notamment dans le cas d'un crash ou en raison d'une surcharge.

Hormis la batterie, il faut un système de distribution du courant vu qu'on dispose de 4 moteurs.

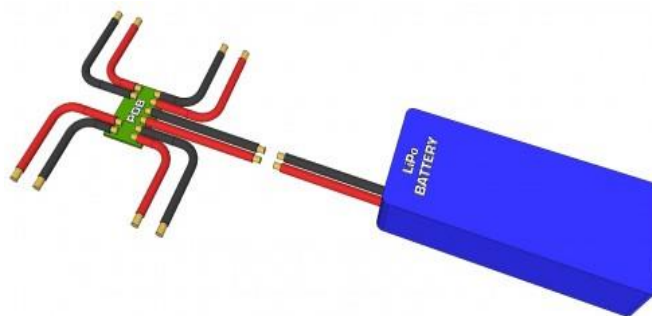


Figure V.4 Système de distribution de l'alimentation [36]

2.3 Capteurs

2.3.1 Gyroscope & Accéléromètre

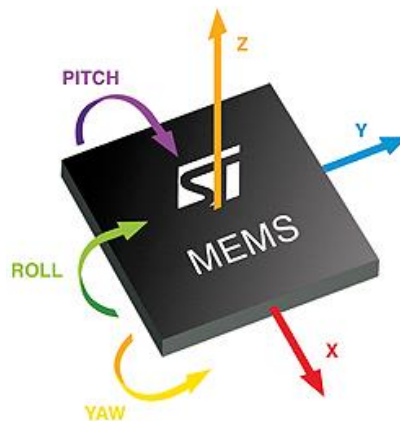


Figure V.5 Centrale inertielle associant les circuits MEMS

Un accéléromètre est le capteur le plus important. Lorsqu'il est fixé à un mobile ou tout autre objet, il permet de mesurer les accélérations statiques (la gravité) et dynamiques (choc, mouvement).

On parle de gyromètre lorsqu'on cherche à détecter une rotation ou vitesse angulaire. Le gyromètre ne donne que des données relatives (ex : rotation de 30° vers la droite) et il n'est pas influencé par la gravité contrairement à l'accéléromètre.

Plus généralement on parle de centrale inertielle (Figure V.5) lorsqu'on cherche à mesurer l'ensemble des 6 accélérations. [37]

Il existe différentes technologies de centrale inertielle mais l'une des plus répandue est la centrale inertielle capacitive pour le rapport performance/prix qui est excellent. Lire l'accélération et la vitesse angulaire est une chose très facile, le problème réside plutôt dans la façon de traiter ces deux informations pour en déduire la position. Le traitement numérique de ces deux informations est très lourd à effectuer sous un microcontrôleur vu les erreurs de dérives. Pour cela, certains constructeurs ont intégré un calculateur de mouvement qu'on appelle DMT (Digital Motion Processor)

2.3.2 Altimètre

Un altimètre est un capteur qui renseigne l'altitude si la mesure est effectuée par rapport au niveau de la mer, la hauteur si c'est par rapport au niveau du sol local et le niveau de vol si c'est par rapport à une surface isobare théorique. Le niveau choisi est appelé calage altimétrique. Ce dernier est le facteur du choix d'un altimètre:

2.3.2.1 Capteur à ultrason



Figure V.6 Capteur Ultrason HC-SR04

Les deux façons les plus courantes de mesurer la distance sont les échos du son et la réflexion de la lumière. Nous intéressons au premier cas. On appelle ultrason le son à haute fréquence; c.à.d. la fréquence utilisée est en générale si élevée que les êtres humains ne peuvent pas l'entendre afin d'éviter d'ennuyer les gens avec ses signaux sonores. Ce capteur peut fournir des mesures exactes de la distance. [38]

Un capteur à ultrason (Figure V.6) envoie un son puis mesure le temps que l'écho met à revenir. Comme le son se déplace à environ 330 mètres par seconde, notre programme peut calculer la distance.

2.3.2.2 Baromètre

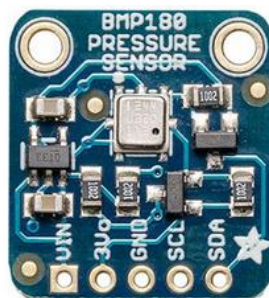


Figure V.7 Capteur de pression BMP180

Le baromètre illustré dans la Figure V.7 est un instrument de mesure qui est utilisé pour gérer le maintien d'altitude. Son fonctionnement est basé sur le fait que la pression atmosphérique diminue, lorsque l'altitude augmente, de l'ordre d' $1/1000^e$ tous les 30m [33].

2.3.3 GPS:



Figure V.8 Module GPS GYGPSV5-NEOM8N

La figure ci-dessus montre un module GPS (Global Positioning system en anglais). C'est un système de positionnement par satellites conçus initialement pour des applications militaires. Désormais, des millions d'applications dans le monde utilisent ce système avec un codage (C/A) réservé au domaine civil contrairement au codage P(Y) réservé au domaine militaire [39].

2.3.4 La communication:



Figure V.9 Module NRF24 L01

Afin de pouvoir communiquer avec le quadri-rotor en vol et lui envoyer des signaux de commande pour le piloter ou recevoir des signaux pour le suivi, il est nécessaire d'avoir une radiocommande ou une station de base équipée d'antennes et d'écran. Cependant, s'équiper de cette station demande plus de budget. Pour y remédier à cela, il est possible d'en fabriquer une en utilisant des modules de communications Radio ou XBee ou le bluetooth qui sont pas cher et facile à trouver sur le marché. La Figure V.9 montre l'un des composants le plus utilisé pour piloter un quadri-rotor sans fil avec une fréquence de 2,4GHz.

2.4 La carte de programmation

La carte de programmation se charge de la réception des valeurs des différents capteurs, les comparer avec des valeurs désirées puis l'émission des ordres de pilotage. Dans le monde des UAVs, on parle de contrôleurs de vols, ces derniers sont équipés en plus des broches d'entrées/sorties, des capteurs et d'un processeur comportant le programme qui traite les informations reçues. La configuration de ce contrôleur est assurée par un logiciel spécifique selon le type du processeur. Plutôt que d'utiliser ces contrôleurs qui sont tout prêt, nous optons pour une réalisation de la carte de programmation d'open source pour charger un programme propre à nous qui va assurer une des commandes étudiées dans ce projet.

Les cartes de programmation sont à choix multiple:

- Arduino: Arduino Uno, Mega, Due ...etc
- Raspberry Pi: Raspberry Pi 2 Model B ...etc
- STM32: STM32 NucleoF103RB ...etc
- FPGA: FPGA Cyclon II, Cyclone V ...etc

La différence entre ces cartes réside dans le processeur, la fréquence opérationnelle, la mémoire programme, la mémoire de calcul, le convertisseur A/N que disposent et les nombre/types de broches.

3. matériels utilisés (configuration matérielle):

3.1 Châssis:

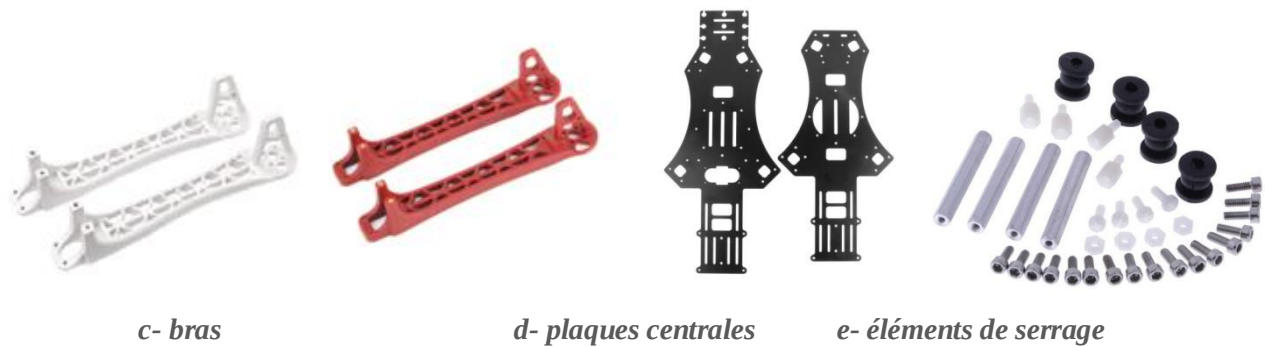


Figure V.10 Comosition du châssis WMC Alien X-500

Le premier composant choisi est le châssis MWC Alien X-500. Il est à la fois léger et rigide. Il est composé de 4 bras de couleur rouge et blanc (Figure V.10-a), 2 plaques centrales inférieure et supérieure (Figure V.10-b) et des éléments de serrage (Figure V.10-c). Ses caractéristiques sont résumées dans le tableau ci-dessus:

Model	Forme X
Empattement	500 mm
Matériaux	Fibre de Verre + Acrylonitrile butadiène styrène (ABS)
Poids	485g
Taille max de l'hélice supportée	12"

Tableau V.2 Caractéristiques du châssis WMC Alien X-500

3.2 Hélices:

Selon le type de vol choisi dans ce projet de fin d'étude, nous avons opté pour des hélices de grandes tailles et d'un pas géométrique petit. La Figure V.11 montre les 4 hélices en nylon de taille 9,4" avec un pas géométrique de 4,3". L'image à droite montre les deux hélices de sens horaire (CW). A gauche, nous disposons de deux hélices de sens antihoraire (CCW). Les hélices en plastique sont très répandues dans le marché des UAV pour son coût très réduit et sa durabilité.

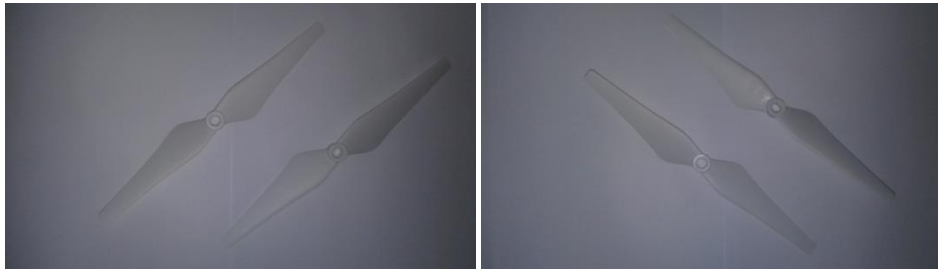


Figure V.11 Les 4 hélices du quadri-rotor

3.3 Moteurs et ECS:



Figure V.12 Moteur readytosky 920kv

La motorisation de notre quadri-rotor est assurée par 4 moteurs brushless de marque ReadyTosky (Figure V.12) qui peuvent être alimentés avec une tension de 7 à 12 V et ayant une vitesse de rotation de 920 Tours/minute/volt. Pour un vol plus stable, nous avons choisi auparavant des hélices de grande taille. Pour tourner ces hélices, nous devons avoir plus de puissance, d'où le choix d'un faible KV selon le résonnement établi dans la section 2.2.1 de ce présent chapitre. Le moteur utilisé est contrôlé par un contrôleur électronique de vitesse (ESC) montré dans la suivante:



Figure V.13 ESC Simonk 30A

Ces ESC ont une plus grande précision avec son oscillateur en cristal et un taux de rafraîchissement élevé. Les valeurs qui caractérisent ces ESC se résument dans le tableau suivant:

Tension d'entrée	2S-4S batterie au lithium/polymer ou 5S-12S NiCd/NIMh batterie
BEC	1A/5V (mode linéaire)
Sortie	30A En continu, 40A Burst jusqu'à 10sec.

Vitesse Max	210 000 rpm pour 2 pôles BLM, 70 000 rpm pour 6 pôles BLM, 35 000 rpm pour 12 pôles BLM
Taille	45mm (L) * 24mm (W) * 11mm (H)
Poids	25g

Tableau V.3 Caractéristique des ESC Simonk 30A

3.4 Batterie/Alimentation:



Figure V.14 Batterie Lithium-polymère 3S

La Figure V.14 illustre la batterie utilisée pour notre quadri-rotor. Elle est à la fois puissante et légère. Les moteurs choisis s'alimentent avec une tension d'alimentation de 11,1 V d'où le choix d'une batterie avec 3 cellules, chacune d'elle correspond à 3,7 V.

Les spécifications de la batterie choisie sont résumées dans le tableau suivant:

Marque	FLOUREON
Chimie	Li-Polymère
Tension	11,1 V
Capacité	2800 mAh
Prise de charge	JST-XH
Bouchon de décharge	XT60 connecteur
Dimensions	110*37*30 (mm)
Poids	213 g
Taux de décharge	35C

Figure V.15 Caractéristiques de la batterie

3.5 La carte de commande (processeur):

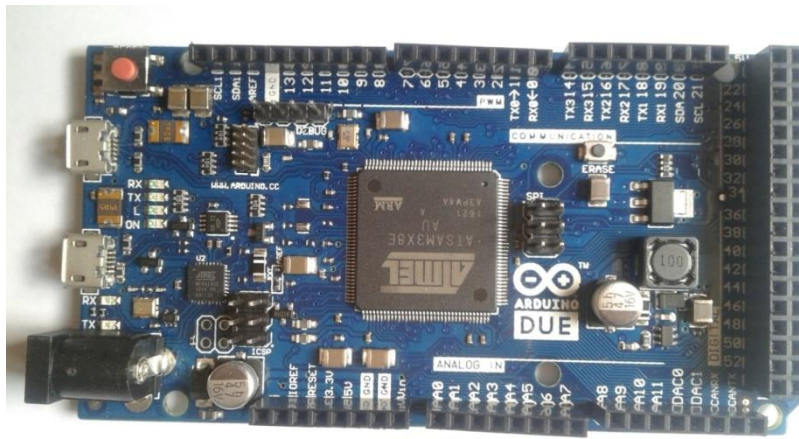


Figure V.16 Arduino Due

Notre choix de la carte de commande s'est porté sur une carte Arduino Due (Figure V.16) du fait que le matériel et le logiciel de programmation sont en open-source, elle est peu onéreuse et multiplateforme ainsi que l'environnement de programmation est simple et clair. La première carte arduino utilisant l'architecture ARM 32bit est la Due contrairement aux autres cartes basées sur l'architecture ATmega 328. L'avantage de l'ARM cortex-M3 est qu'elle possède un noyau 32bits qui autorise des opérations sur 4 octets dans un seul cycle d'horloge ainsi qu'un contrôleur DMA qui peut soulager la CPU lors des tâches intensives en mémoire. Les caractéristiques techniques de la carte Arduino Due sont reportées au tableau ci-dessous:

Microcontrôleur	AT91SAM3X8E
Tension de fonctionnement	3,3 V
Tension d'alimentation (recommandée)	7-12 V
Tension d'alimentation (limites)	6-20 V
Nombre d'entrées/sorties	54 (dont 12 pouvant générer un signal PWM)
Nombre de ports "Analogiques/Numériques"	12
Nombre de ports "analogiques"	2 (DAC)
Courant maximal par E/S	130 mA
Courant pour broches 3,3 V	800 mA
Courant pour broches 5 V	800 mA
Mémoire Flash	512 KB
SRAM	96 KB (2 banques 64 KB et 32 KB)
Vitesse d'horloge	84 MHz

Tableau V.4 Caractéristique de la carte Arduino Due

3.6 La centrale inertielle:

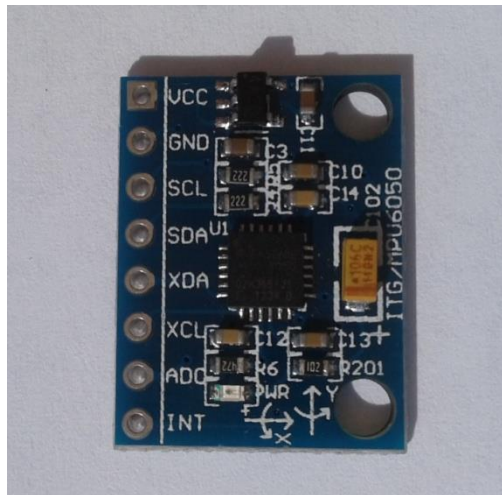


Figure V.17 Module MPU6050

La Figure V.17 montre le capteur MPU6050 qui combine un gyroscope à 3 axes et un accéléromètre à 3 axes sur la même matrice de silicium, ainsi qu'un processeur numérique embarqué qui traite des algorithmes complexes de 6 axes. Les caractéristiques technique de ce module sont données dans le tableau suivant:

Alimentation	2,375V - 3,46V
Consommation	3,9 mA
Interface	I ² C
Accéléromètre	
Plages de mesures (g)	Sensibilité (LSB/g)
± 2	16384
± 4	8192
± 8	4096
± 16	2048
Gyroscope	
Plages de mesures (°/s)	Sensibilité (LSB/°/s)
± 250	131
± 500	65.5
± 1000	32.8
± 2000	16.4

Tableau V.5 Caractéristique du module MPU6050

3.7 Capteur Ultrason:

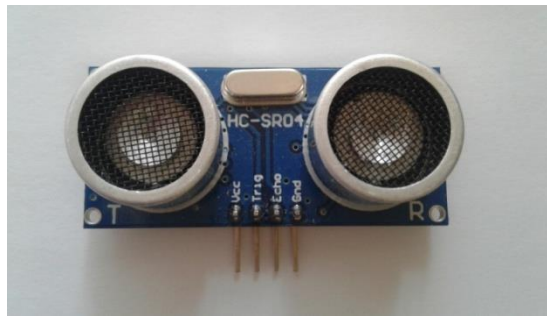


Figure V.18 Capteur ultrason HC-SR04

Afin de mesurer l'altitude de notre quadri-rotor, nous avons utilisé le capteur ultrason HC-SR04 (Figure V.18). Le tableau suivant montre les caractéristiques de ce capteur.

Alimentation	5 V
Consommation	15 mA
Fréquence	40 KHz
Portée	de 6cm à 4m
Déclenchement	impulsion TTL positive de 10µs
Signal écho	Impulsion positive TTL proportionnelle à la distance
Dimension	45*21*18 mm

Tableau V.6Caractéristique du capteur ultrason HC-SR04

3.8 La station de base:

La station de base avec laquelle aura lieu la communication de notre quadri-rotor est constituée d'un ordinateur portable équipé du logiciel matlab/Simulink. On a installé sur ce dernier un support packages qui contient une librairie dans laquelle nous trouvons des blocks pour communiquer et programmer la carte arduino.

3.9 Communication avec le quadri-rotor.

Dans ce projet, nous avons limité le travail avec une communication par câble USB. Toutefois, il est possible de remplacer par la suite le câble avec une communication radio vu précédemment en utilisant le composant NRF24.

4. Montage du quadri-rotor:

Le montage de la structure est très aisé. Les quatre moteurs sont vissés sur les bras, puis ces derniers aux deux plaques centrales. Les câbles d'alimentation des ESC sont soudés sur la carte de distribution de l'alimentation. Cette dernière est ensuite fixée au centre de la plaque du bas du châssis. Au début, les trois câbles de chaque moteur sont reliés aléatoirement aux trois câbles de chaque ESC. Ensuite, si un de ces moteurs ne tourne pas dans le bon sens, il suffit de permuter deux câbles quelconques.

4.1 Mise en place du contrôleur de vol openpilot

Le but de l'utilisation de ce contrôleur dans le début de l'expérience est de calibrer les ESC et tester le sens de rotation des moteurs vu que ce contrôleur dispose d'un logiciel open source et facile à utiliser et ne nécessite aucune programmation. Voici à quoi ressemble le logiciel:

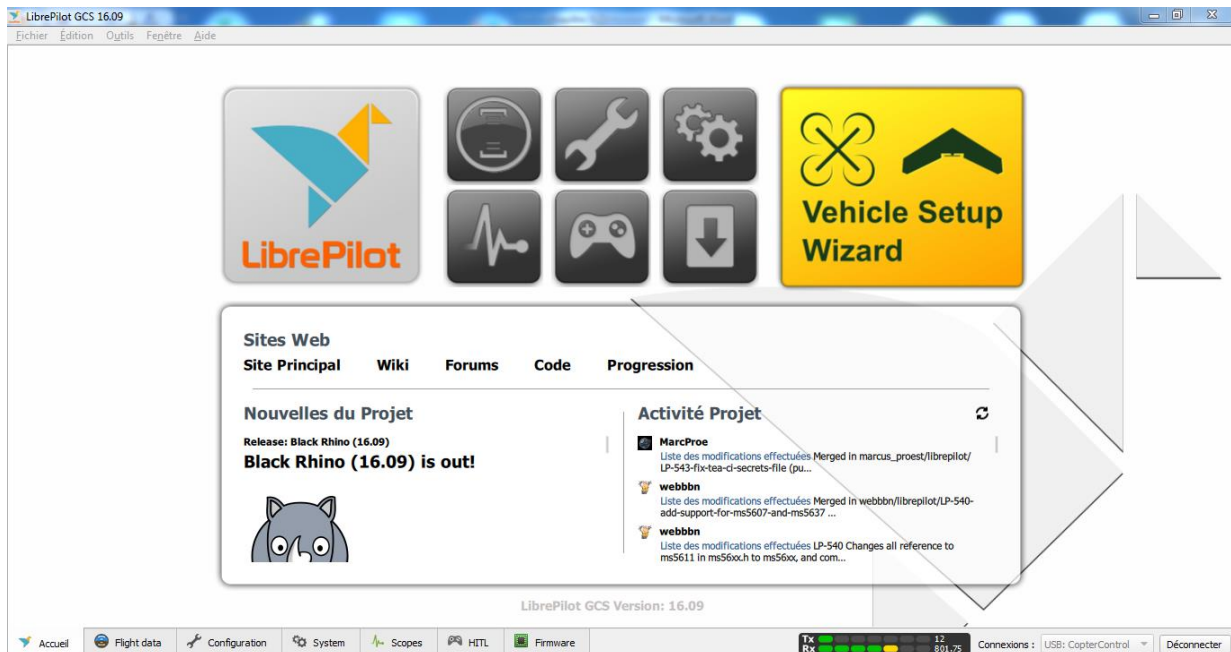


Figure V.19 Capture d'écran du logiciel openpilot



Figure V.20 Liaison du quadri-rotor au pc via le port USB

Après connexion du contrôleur de vol au pc avec un câble USB (Figure V.20), nous apercevons un signal de transmission et de réception en couleur verte et orange en bas de la fenêtre comme montré dans la Figure V.19.

Un premier test a été fait. Nous avons calibré les ESC pour que les moteurs fonctionnent sur une plage commune à bas régime comme à haut régime. Nous avons constaté que le sens de rotation des quatre moteurs était le même. Pour cela, nous avons permuté deux câbles des moteurs 2 et 4.

4.2 Remplacement du contrôleur avec la carte Arduino

Afin de charger le programme propre à nous fait avec matlab, nous avons donc remplacé ce contrôleur avec la carte arduino qui va contenir le programme et communiquer avec la station de base. Le schéma de réalisation de notre quadri-rotor est montré sur la figure suivante:

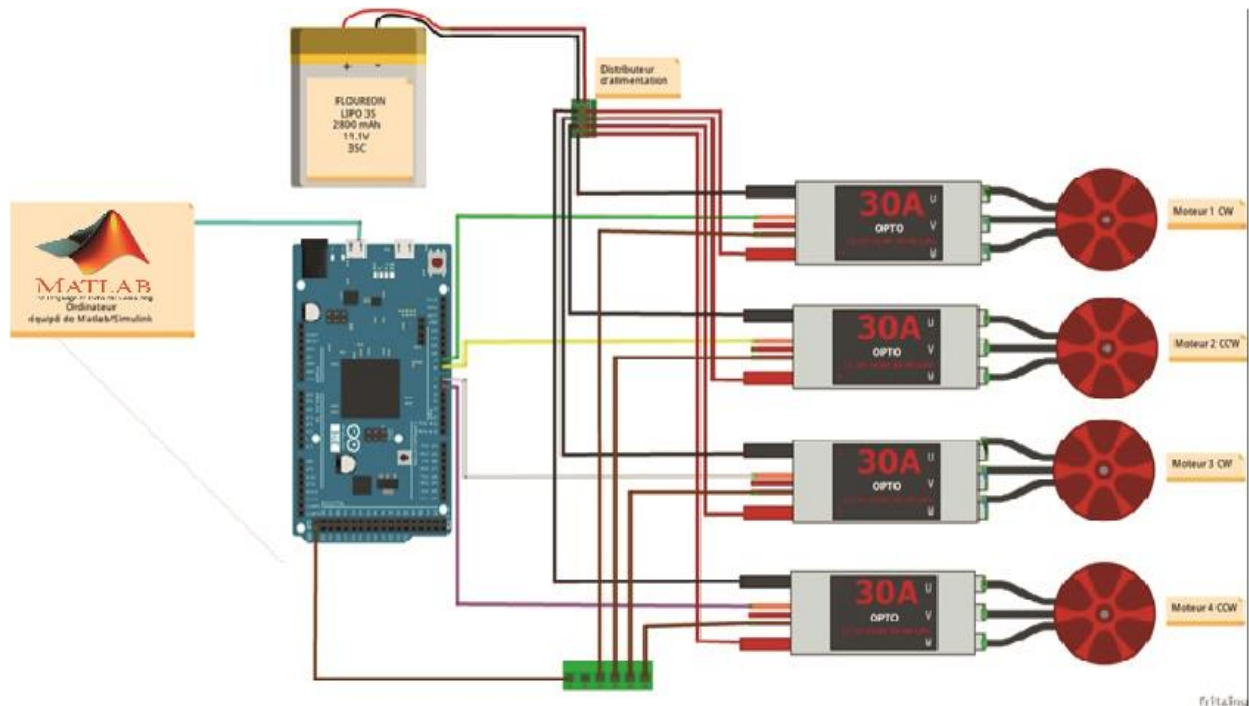


Figure V.21 Schéma de montage du quadri-rotor

La démarche suivie pour commander les moteurs est d'envoyer un signal sur les broches d'arduino via le bus série en utilisant le logiciel matlab/simulink. Ce signal correspond au rapport cyclique compris entre 1 et 255 qui n'est autre que le signal pwm exprimé en binaire au lieu d'un pourcentage. L'ESC se charge de délivrer la tension nécessaire selon le signal reçu sur le fil orange.

Lors du premier essai, nous avons pu commander les moteurs avec matlab/simulink en temps réel en envoyant des signaux pwm. Dans la quatrième partie qui consiste à injecter l'une des commandes étudiées, nous avons rencontré un problème due au solveur de matlab. En effet l'arduino accepte comme solveur "Fixed variable step" alors que notre programme contient des variables continues.

En raison du retard de la réception du matériel et le manque de quelques composants, nous n'avons pas pu arriver à intégrer le capteur ultrason et la centrale inertielle MPU6050 qui sont obligatoires pour faire la régulation et transmettre les informations utiles au programme matlab conçu pour stabiliser l'altitude et l'attitude. Ces deux derniers composants sont testés individuellement avec matlab et les résultats s'avèrent être intéressants et leur intégration au programme précédent ne serait pas une tâche vraiment difficile si on arrive à régler le problème rencontré précédemment.

Conclusion générale et perspective

Le drone est un système complexe non linéaire à plusieurs variables, instable notamment en mode de vol, et représente une dynamique fortement couplée. Le problème traité dans ce manuscrit consiste à garantir la réalisation du suivi de trajectoire en utilisant des contrôleurs linéaires et non linéaires.

Le travail présenté dans ce mémoire consiste d'abord à attribuer au drone un modèle représentatif qui reflète sa dynamique de translation et celle de rotation. Ce modèle a été élaboré en utilisant la méthode Newton-Euler. Nous avons ensuite appliqué des lois de commande linéaire et non linéaire afin d'assurer le suivi du drone aux trajectoires de référence désirées.

Pour cela nous avons procédé en premier lieu à une commande PID qui est portée sur la réalisation de régulateurs PID par la technique d'optimisation de l'erreur quadratique entre la sortie mesurée et la référence. Puis nous avons réalisé une simulation pour pouvoir tester les régulateurs synthétisés, ces tests ont été visualisés et observés.

Les résultats de simulation obtenus ont montré que cette commande présente des dépassements qui ne sont pas tolérés et peuvent être dangereux lorsqu'il s'agit d'un atterrissage.

Pour remédier à cet inconvénient, on a opté pour l'application d'autres commandes non linéaires.

Cette méthode consiste à compenser les termes non linéaires pour obtenir un modèle linéaire découplé présentant des intégrations pures (linéarisation par bouclage). Le système linéarisé obtenu nécessite une boucle externe pour le stabiliser qui a été représenté par un contrôleur linéaire (commande par placement de pôles).

Les résultats de simulation obtenus ont montré que cette commande a linéarisé et découplé le modèle du drone en réalisant la poursuite des références adoptées. Mais lorsqu'on a étudié la rapidité de cette commande, on n'a pas abouti à des résultats satisfaisants. Pour remédier à cet inconvénient, on a opté pour l'application d'une autre commande non linéaire par technique du PDC.

Dans une première étape, nous avons présenté d'abord la technique d'obtention d'un modèle T-S à partir d'un ensemble des équations non linéaires. Ensuite, nous avons abordé le concept de stabilisation de base des modèles TS par interpolation de retour d'état appelée commande PDC (Parallel Distributed Compensation). Les conditions de la stabilisation des modèles flous TS nous a aboutit à un ensemble d'inégalités matricielles linéaires (LMI) qui peuvent être résolues à l'aide d'outils issus du domaine de l'optimisation convexe. Nous avons également abordé le problème de poursuite de trajectoire des modèles TS avec la même stratégie de

commande que celle adoptée dans la stabilisation. Par la suite, une application à un drone type quadri-rotor a été présentée. Dans un premier temps, les résultats de la simulation ont montré que le quadri-rotor a bien été stabilisé du fait que toutes les trajectoires convergent vers les valeurs désirées. Dans un deuxième temps, nous avons obtenus des résultats montrant également la validité de contrôleur PDC dans le cas de suivi de trajectoire.

Afin de vérifier les résultats précédents, nous avons opté pour une réalisation réelle en se basant sur un kit HJ Alien X-500 qui répond à nos besoins. Nous avons remplacé le contrôleur de vol existant avec une carte arduino Due qui sert de moyen de communication avec le drone et la station de base constituée d'un pc portable équipé du logiciel matlab/simulink. Avec matlab, nous pouvons faire des systèmes très complexes et faire des commandes plus élaborées, nous pouvons imposer des trajectoires de notre choix et faire des simulations pour voir le résultat. Notre choix c'est justifié sur le fait que l'arduino ne peut pas être utilisée seule pour faire de telles commandes complexes, il est donc nécessaire d'avoir d'un système de calcul très performant comme le pc portable. La communication est faite avec un câble USB et notre modèle présente un petit qui se résume sur le choix d'un système continu.

Ainsi les perspectives futures sont dans un premier temps concevoir un modèle discret de notre quadri-rotor qui est une tâche pratiquement faisable.

Dans un second lieu, nous intégrerons la centrale inertielle et le capteur ultrason pour pouvoir permettre au modèle de recevoir les informations nécessaires pour la régulation.

Ajouter à cela, nous pouvons remplacer le câble avec une communication à distance en utilisant des modules spécifiques comme le NRF24...etc. afin de pouvoir le piloter avec une plus grande portée.

Il est possible d'ajouter le module GPS vu dans le cinquième chapitre afin d'augmenter la portée du quadri-rotor et faire des vols autonomes en se référant sur des positions bien déterminées.

Pour des raisons plus avancées sur l'intérêt de ce type d'engin, il est intéressant d'intégrer une caméra pour des utilisations de télésurveillance ou dans le contrôle ou d'inspection de zones dangereuses.

Bibliographie

- [1] : K.M.Zemalache, "Commande d'un système sous-actionné : Application à un drone à Quatre Hélices " Thèse de Doctorat, Université d'Evry Val d'Essonne, France.
- [2] T. Hamel, R. Mahony, R. Lozano and J.P. Ostrowski, "Dynamic modeling and configuration stabilization for an X4-flyer", Proceedings of the Triennial World Congress on Automatic Control, Barcelona, Spain, 21-26 Jul 2002, pp. 846-851.
- [3] C.A. Patel, "Building a Test Bed for mini Quadrotor Unmanned Aerial Vehicle with protective Shroud", Master of Science thesis, Graduate School of Wichita, USA, 2006.
- [4]:H.Bouadi, «Synthèse de lois de commande stabilisantes pour unUAV type Quadrotor étude et application », Thèse de Magister, Ecole Militaire Polytechnique,2007.
- [5] H. Khebbache "Tolérance aux défauts via la méthode backstepping des systèmes non linéaires Application : Système UAV de type Quadrirotor", universite Ferhat Abbas de Setif, 2012.
- [6] H. Bouadi, M. Bouchoucha, M. Tadjine "Modelling and Stabilizing Control Laws Design Based on Sliding Mode for an UAV Type-Quadrotor ", Engineering Letters, 15:2, EL_15_2_24, 2007.
- [7]: H. Bouadi, M. Bouchoucha, and M. Tadjine "Sliding Mode Control based on Backstepping Approach for an UAV Type-Quadrotor" World Academy of Science, Engineering and Technology 26, 2007.
- [8] P. Castillo, R. Lozano and A.E. Dzul, "Modelling and Control of Mini-Flying Machines", Springer-Verlag London, 2005.
- [9] H. J. Kim and D. H. Shim, "A Flight Control System for Aerial Robots : Algorithms and Experiments", Control Engineering Practice, 11(12), 2003, pp. 1389-1400.
- [10] J.J.E. Slotine and W. Li, "Applied nonlinear control", Prentice-Hall, Inc, 1991.
- [11]: M. Benyahia, "commande non linéaire et prédictive application à la machine asynchrone" thèse de magister, université de Batna, 2001.
- [12]
- [13]: E.J. Slotine, W. Li, "Applied nonlinear control". Prentice-Hall International Editions,1991.
- [14]: Z. Saida "Etude d'une commande non linéaire d'un moteur à induction" université de Batna.
- [15] Lalili, Djafer, " Les commandes non linéaires adaptatives de la machine asynchrone", Mémoire de Magistère, Ecole Nationale Polytechnique, Alger 1999.

[16] Lamine Kistrane, "Commande non linéaire de la machine à induction «aspect expérimental», Mémoire de Magister en Électrotechnique, Université de Batna, 2008. Algérie.

[17] Meziane.S, Toufouti.R and Benalla.H. "Applied input-output linearizing control for highperformance induction motor", JATIT, Journal of Theoretical and Applied Information Technology, Vol. 4, Number1, pp. 7-15, January, 2008.

[18]: L.Derafa, A.Ouldali "Feedback Linearization and Pole Placement Law Control for A four Rotors Helicopter" Control and Command Laboratory EMP, CGE-06, Alger 13-14 April 2009.

[19]: N. Guenrad, T. Hamel, V. Moreau, " Modélisation et élaboration de commande de stabilisation de vitesse et de correction d'assiette pour un drone de type X4-Flyer ", UNSA-CNRS, 2004.

[20]: H. Bouadi, M. Bouchoucha, and M. Tadjine "Sliding Mode Control based on Backstepping Approach for an UAV Type-Quadrotor" World Academy of Science, Engineering and Technology 26, 2007.

[21]:H.Bouadi, «Synthèse de lois de commande stabilisantes pour unUAV type Quadrotor étude et application », Thèse de Magister, Ecole Militaire Polytechnique, 2007.

[22] T. Bouarar, Contribution à la synthèse de lois de commande pour les descripteurs de type Takagi–Sugeno incertains et perturbés, Thèse de Doctorat, Université Reims Champagne-Ardenne.2009

[23] S. Aouaouda, Modélisation multimodale et commande prédictive d'une station d'épuration , Thèse de Doctorat, Université Badji Mokhtar-Annaba (UBMA) ,2012

[24] K . Guelton, Contribution a la Commande des systèmes non linéaires par l'approche TAKAGI-SUGENO, Thèse de Doctorat , Université Reims Champagne-Ardenne,2010

[25] M.Dideir,Contribution à l'étude d e la stabilité des systèmes électrotechniques, Thèse de Doctorat , école Doctorale D Lorraine ,2009.

[26] S. Chaabani , Dynamique non-linéaire d'un dirigeable flexible , Thèse deDoctorat,Université d'Evry ,École Polytechnique Val d'Essonne de Tunisie,2014

[27] D. Ounnas et T. Bouktir M. Ramdani S. Chenikher , A Fuzzy Tracking ControlDesign Strategy for Wind Energy Conversion System .

[28] D. Ounnas et T. Bouktir M. Ramdani S. Chenikher , A Fuzzy Tracking ControlDesign Strategy for Wind Energy Conversion System .

[29] D. Ounnas et T. Bouktir M. Ramdani S. Chenikher , Tracking Control ForPermanent Magnet Synchronous Machine Based On Tak:agi-Sugeno FuzzyModels,2013.

[30] Calcul et contruction d'un avion [en ligne]. Disponible sur:
http://www.gilbert-pernot.fr/helice_calcul_fabrication.html

[31] Moteur brushless, le principe [en ligne]. Disponible sur: <http://www.dronekas.com/drone-principe-du-moteur-brushless/>

[32] Le brushless ou "Moteur synchrone triphasé" [en ligne]. Disponible sur: <http://ericaeromodelisme974.unblog.fr/2011/11/20/le-brushless-ou-moteur-synchrone-triphas/>

[33] Rodolphe Jobard, Les drones fonctionnement, télépilotage, applications, réglementation. Paris, Eyrolles (2016), 2ème édition. p48-53. ISBN: 978-2-212--14189-4

[34] Calibration des ESC, calibrer vos ESC sous CleanFlight et BLHeliSuite [en ligne]. Disponible sur: <http://www.dronekas.com/calibration-esc/>

[35] Comment choisir les composants d'un quadricoptère [en ligne]. Disponible sur: https://projetdroneamateur.wordpress.com/choix_composant/

[36] Comment fabriquer un drone UAV [en ligne]. Disponible sur: <http://www.robotshop.com/blog/fr/comment-fabriquer-un-droneuav-lecon-5-assemblage-4839>

[37] Accéléromètre, site du wikipédia .

[38] Tero Karvinen, Kimmo Karvinen, Ville Valtokari, Les capteurs pour Arduino et Raspberry pi. Paris, Dunod (2014). p31-37. ISBN: 978-2-10-071793-4.

[39] Jean Marc Piéplu, GPS et Galileo, Systèmes de navigation par satellites. Paris, Eyrolles (2006). ISBN: 978-2-212-11947-3