

بِسْمِ اللَّهِ الرَّحْمَنِ الرَّحِيمِ

République Algérienne Démocratique et Populaire
Ministère de l'Enseignement Supérieur et de la Recherche Scientifique
Université des Sciences et de la Technologie d'Oran Mohamed Boudiaf
Faculté de Génie Electrique
Département d'Automatique



ECOLE DOCTORAL DE GÉNIE ÉLECTRIQUE

E.D.G.E

Laboratoire de Développement des Entrainement Electrique

Thèse Présentée pour l'obtention du

DIPLOME DE MAGISTER EN AUTOMATIQUE

Par

BELLAHCENE ZAKARIA

Ingénieur d'Etat en Génie Electrique/Automatique

Titre:

Synthèse de lois de commande robuste pour un hélicoptère à quatre hélices

Soutenue le: 6/3/2013

Membres du jury:

Président:	M. Omari Abdelhafid	Professeur "USTO"
Encadreur:	M. Bouhamida Mohamed	Professeur "USTO"
Examineur:	M. Mokhtari Abdellah	Professeur "USTO"
Examineur:	M.Mansouri Abdellah	Professeur "E.N.P.O"
Examineur:	M. Zemalache Meguenni Kadda	Maître de conférences «A» "USTO"

REMERCIEMENTS

Je tiens surtout à assurer ma sincère reconnaissance, mon plus grand respect et ma totale gratitude à Mr. Bouhamida Mohamed, mon encadreur et Professeur à l'Université des Sciences et de la Technologie d'Oran, pour tous les efforts, l'aide, le soutien et les encouragements qu'il a montré tout au long de ce travail et aussi pour tant de compréhension, et disponibilité dont il a fait preuve.

C'est un honneur pour moi que Mr. Omari Abdelhafid Professeur à l'USTO ait accepté de participer à ce jury d'en assumer la tâche de président.

Mrs. Mokhtari Abdellah , Mansouri Abdellah et Zemalache Meguenni Kadda me font l'honneur de participer à mon jury, de juger mon travail, le commenter et l'enrichir par ses remarques, je les en suis profondément reconnaissant.

Enfin, j'adresse mes remerciements à tous ceux que j'aurais pu omettre de citer et qui de près ou de loin m'ont aidé et soutenu.

Dédicace

Je dédie cette thèse :

A mes chers parents

A mes chers frères et sœurs

Et à tous mes ami(e)s.

Zakaria



Sommaire



SOMMAIRE

LISTE DES FIGURES	i
LISTE DES TABLEAUX.....	iii
LISTE DES ABRÉVIATIONS	iii
Introduction générale.....	1

Chapitre I : Description et état de l'art

I.1. Introduction	3
I.2. Les catégories de drones.....	3
I.3. Présentation et principe du vol.....	5
I.4. L'histoire des quadrirotors	6
I.5. Les travaux de recherche récents sur les quadrirotors.....	8
I.5.1. Travaux portant sur la commande des quadrirotor.....	8
I.6. Les projets existants sur le quadrirotor	10
I.6.1. Le projet Mesicopter	10
I.6.2 Le projet STARMAC	10
I.6.3 Le projet OS4	11
I.7. Conclusion	12

Chapitre II : Modélisation dynamique du quadrirotor

II.1. Introduction	13
II.2. Modèle de quadrirotor.....	13
II.2.1. Les repères.....	14
II.2.2. Les angles d'Euler.....	14
II.2.3. Vitesses angulaires	15
II.2.4. Vitesses linéaires	16
II.2.5. Les effets physiques agissants sur le quadrirotor.....	16
A. Les forces	16
B. Les moments.....	17
C. Effet gyroscopique	18
II.2.6. Développement du modèle mathématique(Newton-Euler).....	19
A. Equations de mouvement de translation	21
B. Equations de mouvement de rotation	21
C. Dynamique du rotor	22
II.2.7. La représentation d'état du système	23
II.3. Conclusion... ..	25

Chapitre III : Synthèse de lois de commande robuste

<i>III.1. Introduction.....</i>	<i>26</i>
<i>III.2. La commande par mode de glissement</i>	<i>27</i>
<i>III.2.1. Principe de la commande par mode de glissement</i>	<i>27</i>
<i>III.2.2. Notions de base de la commande mode de glissement</i>	<i>28</i>
<i>A. Surface de glissement.....</i>	<i>28</i>
<i>B. Régime glissant idéal.....</i>	<i>29</i>
<i>C. Régime glissant réel.....</i>	<i>29</i>
<i>D. Condition d'existence et d'unicité du régime glissant</i>	<i>29</i>
<i>III.2.3. Méthodes de synthèse classiques de la commande SMC</i>	<i>29</i>
<i>A. Méthode de Philipov</i>	<i>29</i>
<i>B. Méthode d'Utkin (méthode de la commande équivalente)</i>	<i>31</i>
<i>III.2.4. La réticence dans le mode de glissement (chattering).....</i>	<i>32</i>
<i>A. La proposition de Slotine</i>	<i>33</i>
<i>B. La proposition de Harshima</i>	<i>33</i>
<i>III.2.5. Structures de contrôle par mode de glissement.....</i>	<i>33</i>
<i>III.3. Synthèse de la commande par mode de glissement.....</i>	<i>34</i>
<i>III.3.1. Démonstration.....</i>	<i>34</i>
<i>III.3.2. Les Résultats de simulation.....</i>	<i>36</i>
<i>III.3.3. Etude de la robustesse.....</i>	<i>40</i>
<i>III.4. La commande par mode flou glissant</i>	<i>41</i>
<i>III.4.1. Quelques définitions.....</i>	<i>41</i>
<i>a) Ensemble flou.....</i>	<i>41</i>
<i>b) Noyau.....</i>	<i>41</i>
<i>c) Support.....</i>	<i>41</i>
<i>d) Hauteur.....</i>	<i>42</i>
<i>e) Ensemble normalisé</i>	<i>42</i>
<i>III.4.2. Opérations.....</i>	<i>42</i>
<i>a) L'intersection</i>	<i>42</i>
<i>b) L'union.....</i>	<i>42</i>
<i>c) La complémentation.....</i>	<i>43</i>
<i>III.4.3. Structure générale d'un contrôleur flou</i>	<i>43</i>
<i>a) Interface de fuzzification.....</i>	<i>43</i>
<i>b) Base de règles</i>	<i>44</i>
<i>c) Mécanisme d'inférence</i>	<i>44</i>
<i>1) Méthode d'inférence MAX-MIN</i>	<i>44</i>
<i>2) Méthode d'inférence Max-Produit.....</i>	<i>44</i>
<i>3) Méthode d'inférence Somme-Produit</i>	<i>45</i>
<i>d) Interface de déffuzzification.....</i>	<i>46</i>
<i>III.4.4. Synthèse de la commande par mode flou glissant</i>	<i>46</i>
<i>III.4.5. Les résultats de simulation.....</i>	<i>48</i>
<i>III.4.6. Etude de la robustesse.....</i>	<i>50</i>
<i>III.5. La commande par le PI mode flou glissant avec optimisation par l'algorithme génétique</i> <i>« PIFSMC-GA »</i>	<i>51</i>

III.5.1. Les algorithmes génétiques	52
III.5.2. Notions fondamentales sur les algorithmes génétiques	52
1) Individu (chromosome / séquence)	52
2) Population.....	52
3) Environnement	52
4) Fonction de fitness	52
III.5.3. Principaux étapes des algorithmes génétiques	53
III.5.4. Codage et population initiale.....	53
III.5.5. Opérateurs génétiques	53
a) Le croisement	53
b) La mutation.....	54
III.5.6. Différentes étapes de mise en œuvre	55
a) Croisement	55
b) Mutation.....	55
c) Sélection	55
III.5.7. Fonctionnement de l'AG	56
III.5.8. Synthèse de la commande par PI flou glissant avec optimisation par l'AG.....	57
III.5.9. Démonstration.....	58
III.5.10. Résultats d'optimisation.....	61
III.5.11. Résultats de simulation.....	61
III.5.12. Etude de la robustesse.....	64
III.6. La commande par backstepping.....	67
III.6.1. Définition	68
a) Point d'équilibre	68
b) Stabilité	68
c) Stabilité de Lyapunov	68
III.6.2. Méthodes d'analyse de la stabilité des systèmes	68
a) Méthode direct de Lyapunov	69
b) La synthèse de la commande par la méthode directe de Lyapunov	70
III.6.3. Technique de commande par le backstepping	71
a) Algorithme de base	71
b) Cas des systèmes d'ordre n	74
III.7. Synthèse de la commande par backstepping	75
III.7.1. Démonstration.....	75
III.7.2. Résultats de simulation.....	78
III.7.3. Etude de la robustesse.....	80
III.8. Synthèse de la commande par l'intégrale backstepping	81
III.8.1. Démonstration.....	81
III.8.2. Résultats de simulation.....	83
III.8.3. Etude de la robustesse.....	85
III.9. Etude comparative.....	89
III.10. Conclusion... ..	90

Chapitre IV: Synthèse d'un observateur non linéaire

<i>IV.1. Introduction</i>	<i>91</i>
<i>IV.2. Observabilité des systèmes non linéaire.....</i>	<i>92</i>
<i>IV.3. Observateur à mode glissant.....</i>	<i>93</i>
<i>IV.3.1. Observateur classique à mode glissant</i>	<i>93</i>
<i>IV.3.2. Observateur à mode glissant étape par étape</i>	<i>94</i>
<i>IV.4. Synthèse de l'observateur.....</i>	<i>97</i>
<i>IV.4.1. Résultats de simulation.....</i>	<i>98</i>
<i>IV.4.2. Etude de la robustesse</i>	<i>102</i>
<i>IV.5. Conclusion.....</i>	<i>103</i>
<i>Conclusion générale.....</i>	<i>105</i>
<i>Annexe</i>	<i>107</i>
<i>Bibliographie.....</i>	<i>108</i>

LISTE DES FIGURES

- Figure I.1.** Les catégories des drones.
- Figure I.2.** Quadrirotors.
- Figure I.3.** Illustration du mouvement de quadrirotor.
- Figure I.4.** Bréguet Richet Gyro1907.
- Figure I.5.** Oemichen 1920.
- Figure I.6.** Le quadrirotor de Bothezat 1922.
- Figure I.7.** Convertawings Model A 1956.
- Figure I.8.** Le Mesicopter.
- Figure I.9.** Le quadrirotor STARMACII.
- Figure I.10.** Le Quadri-rotors OS4.
- Figure II.1.** Vue générale du quadri-rotors.
- Figure II.2.** Les trois rotations roulis tangage, et lacet.
- Figure III.1.** Schéma synoptique de la commande proposée.
- Figure III.2.** Mode de glissement.
- Figure III.3.** Illustration de la résolution de Philipov.
- Figure III.4.** Commande équivalente.
- Figure III.5.** Fonction de saturation *sat*.
- Figure III.6.** Fonction de saturation *cont*.
- Figure III.7.** Structure de régulation par ajout de la commande équivalente.
- Figure III.8.** Trajectoire de référence.
- Figure III.9.** Résultats de poursuite des trajectoires désirées selon (x, y, z, ψ) .
- Figure III.10.** Les commandes $(u_1 u_2 u_3 u_4)$.
- Figure III.11.** Les erreurs de déplacements suivant (x, y, z) et l'angle ψ .
- Figure III.12.** Schéma global de la commande en présence de perturbation
- Figure III.13.** L'influence de la perturbation selon (z) .
- Figure III.14.** Format d'un ensemble flou normalisé.
- Figure III.15.** Union et intersection de deux sous ensemble flous.
- Figure III.16.** Structure d'un système de contrôle flou.
- Figure III.17.** Méthode d'inférence Max-Min.
- Figure III.18.** Méthode d'inférence Max-Prod.
- Figure III.19.** Méthode d'inférence floue Somme-Produit.
- Figure III.20.** Fonction d'appartenance de l'entrée du contrôleur FSMC.
- Figure III.21.** Fonction d'appartenance de la sortie du contrôleur FSMC.
- Figure III.22.** Signal de commande du contrôleur flou-glissant.
- Figure III.23.** Schéma bloc du contrôleur flou-glissant.
- Figure III.24 :** Résultats de poursuite des trajectoires désirées selon (x, y, z, ψ) .
- Figure III.25.** Les commandes $(u_1 u_2 u_3 u_4)$
- Figure III.26.** Les erreurs de déplacements suivant (x, y, z) et l'angle ψ .
- Figure III.27.** L'influence de la perturbation selon (z) .
- Figure III.28.** Croisement en codage binaire.
- Figure III.29.** Opérateur de mutation.

Figure III.30. Cycle d'un algorithme génétique.

Figure III.31. Organigramme de l'algorithme génétique.

Figure III.32. Schéma bloc du contrôleur par PI flou glissant avec optimisation par l'AG « PIFSMC-GA ».

Figure III.33. Représentation de la fonction objective.

Figure III.34. Résultats de poursuite des trajectoires désirées selon (x, y, z, ψ) .

Figure III.35. Les commandes (u_1, u_2, u_3, u_4)

Figure III.36. Les erreurs de déplacements suivant (x, y, z) et l'angle ψ .

Figure III.37. L'influence de la perturbation selon (z) pour le contrôleur « PIFSMC-GA ».

Figure III.38. L'influence de la perturbation selon (x, y, z) avec des incertitudes paramétriques +20% de (I_x, I_y, I_z) pour le contrôleur « PIFSMC-GA ».

Figure III.39. Les signaux de commandes et des erreurs selon (x, y, z) avec des incertitudes paramétriques +20% de (I_x, I_y, I_z) pour le contrôleur « PIFSMC-GA ».

Figure III.40 : l'influence de la perturbation selon (z) , du contrôleur « FSMC » et « SMC », « PIFSMC-GA » avec des incertitudes paramétriques +20% de (I_x, I_y, I_z) .

Figure III.41. Interprétation géométrique du théorème de Lyapunov.

Figure III.42. Résultats de poursuite des trajectoires désirées selon (x, y, z, ψ) .

Figure III.43. Les commandes (u_1, u_2, u_3, u_4)

Figure III.44. Les erreurs de déplacements suivant (x, y, z) et l'angle ψ .

Figure III.45. L'influence de la perturbation selon (z) .

Figure III.46. L'approche de commande proposée.

Figure III.47. Résultats de poursuite des trajectoires désirées selon (x, y, z, ψ) .

Figure III.48. Les commandes (u_1, u_2, u_3, u_4)

Figure III.49. Les erreurs de déplacements suivant (x, y, z) et l'angle ψ .

Figure III.50. L'influence de la perturbation selon (z) pour le contrôleur intégrale backstepping.

Figure III.51. L'influence de la perturbation selon (x, y, z) avec incertitudes paramétriques +20% de (I_x, I_y, I_z) pour le contrôleur intégrale backstepping.

Figure III.52. Les signaux de commandes et des erreurs selon (x, y, z) avec incertitudes paramétriques +20% de (I_x, I_y, I_z) pour le contrôleur intégrale backstepping.

Figure III.53. L'influence de la perturbation selon (z) , du contrôleur backstepping et intégrale backstepping avec des incertitudes paramétriques +20% de (I_x, I_y, I_z) .

Figure IV.1. Schéma de principe d'un observateur.

Figure IV.2. Commande avec Observateur.

Figure IV.3. Résultats de poursuite des trajectoires désirées selon le lacet ψ et les axes (x, y, z)

Figure IV.4. Erreurs entre les états réels et leurs estimés.

Figure IV.5. Erreurs entre les états estimés et les états désirés.

Figure IV.6. Résultats de poursuite des trajectoires désirées et estimées pour les variables non mesurées par les capteurs $(\dot{x}, \dot{y}, \dot{z}, \dot{\psi})$ et $(\ddot{x}, \ddot{y}, \ddot{z})$.

Figure IV.7. L'influence de la perturbation selon (x, y, z) avec incertitudes paramétriques +20% de (I_x, I_y, I_z) .

Figure IV.8. Les erreurs entre les états estimés et les états désirés selon (x, y, z) avec incertitudes paramétriques +20% de (I_x, I_y, I_z) .

LISTE DES TABLEAUX

Tableau I.1. Quelques projets sur les quadrirotors.

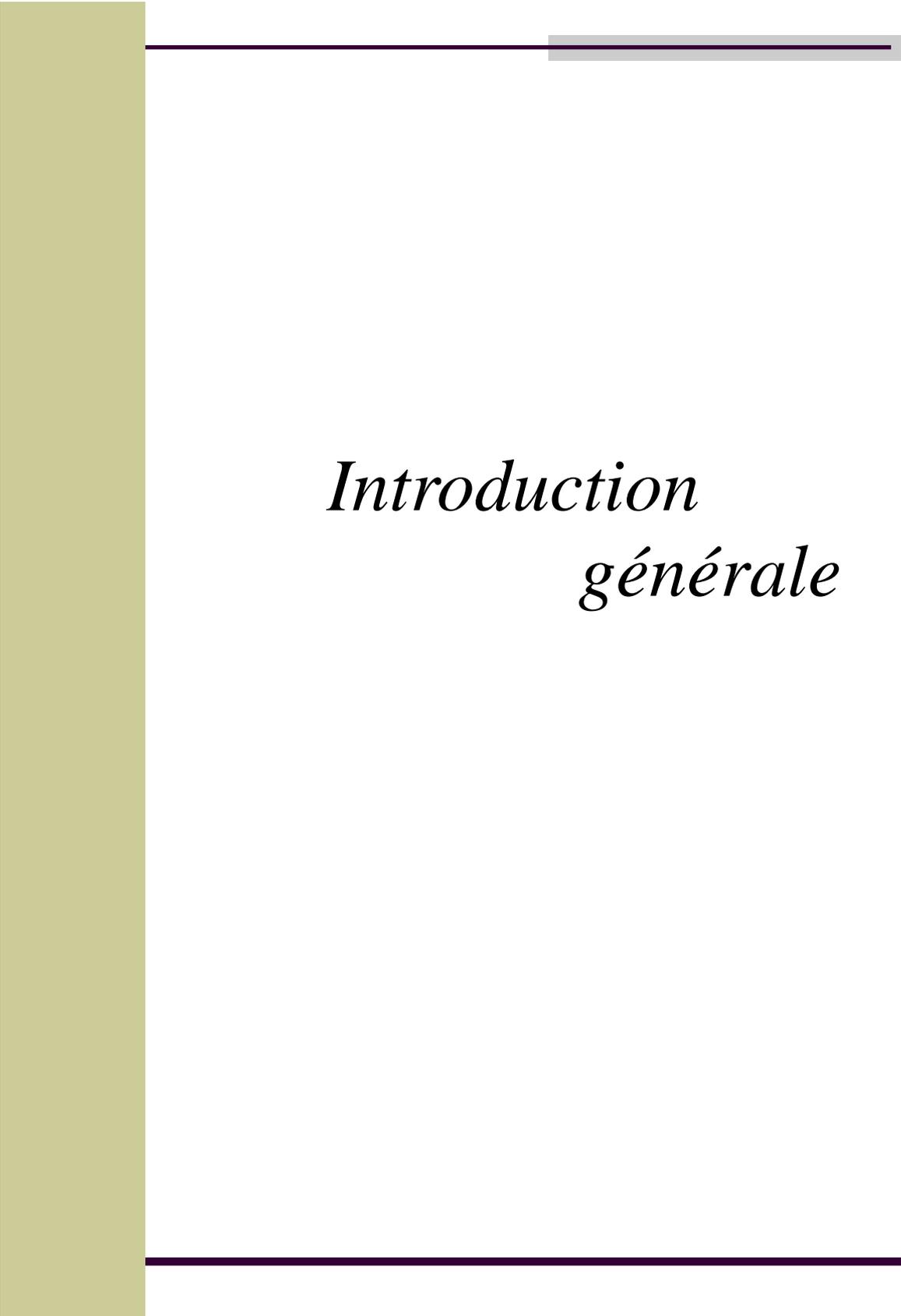
Tableau III.1. Paramètres du contrôleur PI optimisés par l'AG.

Tableau III.2. Étude comparative entre les commandes développées pour le quadrirotor.

Tableau III.3: Etude comparative entre les deux techniques de commande avec des perturbations et des incertitudes paramétriques.

LISTE DES ABRÉVIATIONS

UAV	Unmanned Aerial Vehicle.
VTOL	Vertical Take-off and Landing.
OS4	Omnidirectional Stationary Flying Outstretched Robot.
LQ	Linear Quadratic.
PID	Proportional Integral Derivative.
SMC	Sliding Mode Control.
FLC	Fuzzy Logic Control.
GA	Genetic algorithms.



Introduction
générale

Introduction générale

Les drones ou UAV (Unmanned Aerial Vehicles) sont des engins volants sans pilote capables de mener à bien une mission, leur utilisation est d'abord militaire pour des missions de reconnaissances ou de surveillance. Ils sont bien adaptés pour la réalisation de missions qui mettraient potentiellement un équipage en danger ou qui nécessitent une permanence sur zone. Des applications civiles font leur apparition comme la surveillance du trafic autoroutier, la prévention des feux de forêts et l'intervention dans des environnements hostiles comme les milieux radioactifs, la récolte de données météorologiques ou bien encore l'inspection d'ouvrages d'art.

Le développement de plates-formes robotiques volantes connaît un essor croissant depuis quelques années en raison de la miniaturisation toujours plus poussée des capteurs et des actionneurs, mais surtout grâce à la possibilité d'embarquer des cartes de commande toujours plus performantes capables d'exécuter la masse de calcul nécessaire au contrôle des robots volants.

Quelle que soit la méthode utilisée pour obtenir un modèle mathématique d'un système physique, il existe toujours un compromis entre la simplicité du modèle et son aptitude à décrire l'ensemble des phénomènes qui le caractérise. Ces écarts entre le modèle et le système réel sont souvent modélisés par des grandeurs d'erreurs (erreur d'entrée, erreur de sortie, erreur sur l'état,..., erreur de modèle) car les systèmes réels sont la plupart du temps non linéaires, mal définis, ont des paramètres variables et sont soumis à des perturbations externes. La modélisation de ces systèmes relève très souvent d'une approximation des phénomènes physiques mis en jeu. Or, c'est à partir de cette représentation approximative des systèmes que l'on souhaite construire une commande robuste dans le sens où elle devra assurer une faible sensibilité aux incertitudes sur les paramètres, à leurs variations et aux perturbations externes. L'objectif que nous nous sommes fixé pour ce travail de recherche peut donc se résumer à déterminer des commandes robustes et performantes appliquée a notre système non linéaire.

Le but de notre étude se situe dans le cadre de la synthèse de lois de commande stabilisantes pour un mini drone de type quadrirotor. Les problèmes auxquels nous nous intéressons sont décrits comme suit :

La modélisation dynamique de système et la synthèse de lois de commande stabilisantes pour un système non linéaire couplé, ainsi que la poursuite (suivi) de trajectoires. Enfin le

problème de la synthèse d'un observateur non linéaire dans la mesure où le milieu de navigation présente des perturbations extérieures et notamment pour pouvoir contourner le problème des erreurs de modélisation.

Organisation de la thèse

Cette thèse est divisée en quatre chapitres résumés ci-dessous.

Chapitre I : est consacré à la description et au fonctionnement du quadrirotor. Nous donnons un bref historique sur la conception des quadrirotors et présentons quelques travaux récents sur la commande utilisant cette plateforme.

Chapitre II : Dans ce chapitre une étude détaillée de modèle dynamique ainsi que le modèle des moteurs utilisés. Des hypothèses sont établies afin d'obtenir quelques simplifications du modèle.

Chapitre III : Après avoir établi le modèle dynamique, Nous concevons des lois de commande basées sur différentes techniques. Des résultats en simulation sont présentés pour ce modèle afin d'illustrer les développements théoriques. Une analyse de robustesse est effectuée pour montrer l'efficacité de chaque commande et enfin une étude comparative entre les différentes techniques de commande utilisées en terme de précision suivant un critère.

Chapitre IV : Afin de pouvoir estimer les états non mesurables en présence des perturbations extérieures et parer au problème des erreurs de modélisation La synthèse d'un observateur non linéaire est nécessaire et fera l'objet du dernier chapitre.

Conclusion générale : sur le travail effectué et les résultats obtenus et évoquons quelques perspectives.



Chapitre I

*Description et état
de l'art*

I.1 Introduction

Un UAV ou « Drone » est un aéronef inhabité qui utilise les forces aérodynamiques pour produire un vol vertical. Il peut être piloté à distance, autonome ou semi autonome [Pat.06]. Il est susceptible d'emporter différentes charges utiles, le rendant capable d'effectuer des tâches spécifiques, pendant une durée de vol qui peut varier en fonction de ses capacités. Son utilisation a d'abord été connue dans les applications militaires, comme la surveillance et la reconnaissance et comme plateforme de désignation de cible ou comme arme. Puis, plusieurs applications civiles sont devenues concurrentes, notamment dans l'observation des phénomènes naturels (Avalanches, volcans...), la pulvérisation des pesticides sur les surfaces agricoles, la surveillance de l'environnement (exemple : mesures de la pollution) et des réseaux routiers, la maintenance des infrastructures...etc.

Les deux principaux avantages d'un UAV par rapport à un aéronef standard sont :

- la miniaturisation possible ;
- l'évolution dans un environnement «3D», «Dull, Dirty and Dangerous» (monotone, sale et dangereux), inadapté aux pilotes humains.

Ces avantages ont été reconnus très tôt dans le domaine militaire mais sont également valables dans le domaine civil. La miniaturisation permet de réduire les coûts de fabrication et de fonctionnement. Par exemple une photo aérienne prise par un ballon captif de 2.4m de diamètre (société Phodia ou Aerovue par exemple) revient plus de 10 fois moins cher qu'une photo prise depuis un hélicoptère traditionnel. La réduction de la taille permet également une mise en oeuvre plus simple qui ne nécessite ni installations au sol ni expert en pilotage.

I.2. Les catégories de drones

La classification des drones peut se faire selon plusieurs paramètres : Les performances, la taille, l'altitude ou selon la mission. De ce fait on trouve la classification suivante :

Les micro UAV : (Micro Air Vehicule ou MAV), pouvant être contenu dans une sphère de 30 cm. Ces micro-drones sont différenciées par la miniaturisation des cartes électroniques et le choix des matériaux des capteurs et des systèmes embarqués.

Mini-UAV : Ce sont des drones plutôt légers et de taille réduite (jusqu'à quelques kilogrammes et d'une envergure jusqu'à 1 à 2 mètre) facilitant la mise en oeuvre et généralement utilisés pour l'observation de zones difficiles d'accès.

Les drones MALE (Medium Altitude High Endurance) : sont utilisé pour des vols de plus longue durée à une altitude opérationnelle de 10 à 15 Km. Avec une taille comparable aux avions traditionnels. Ils sont généralement voués à l'observation, mais certains peuvent embarquer des armes.

Les drones HALE (High Altitude High Endurance) volent à une altitude d'environ 20 km. Il s'agit de drones de grande envergure volant à très haute altitude. Ils sont majoritairement utilisés pour des communications, comme alternative aux satellites.

Ces deux types de drones font partie de la classe de grande taille pouvant accomplir des missions atteignant 2 jours. [Ped.05]

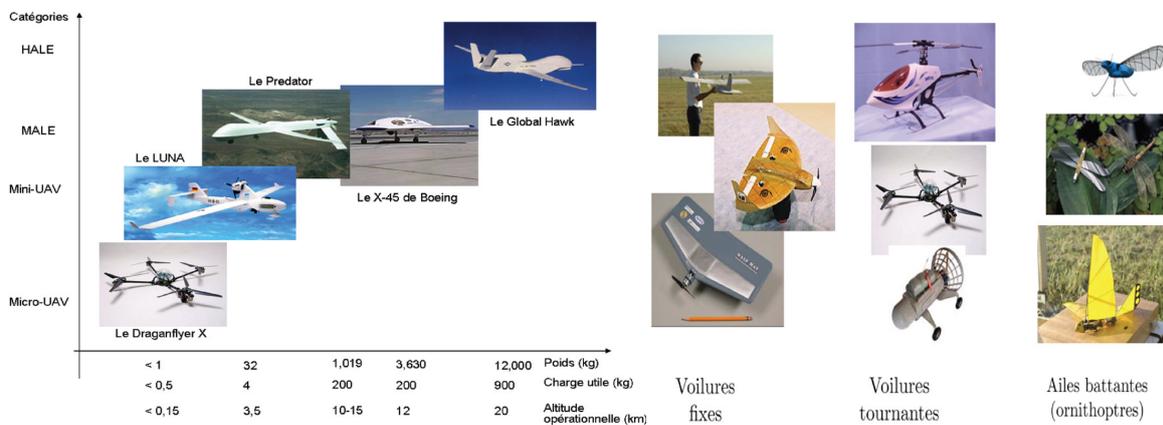


Figure I.1 Les catégories des drones. [Ped.05]

Le fonctionnement aérodynamique fournit une autre possibilité de classification. Ainsi, les drones peuvent être structurés principalement en trois familles (Figure.I.1) :

- ceux à voilures fixes.
- ceux à voilures tournantes.
- les aéronefs à ailes battantes.

Une étude bibliographique montre que la recherche en robotique aériennes est principalement focalisé sur des drones à voilures tournantes : des hélicoptères et des architectures "VTOL".

Dans notre projet, on s'intéresse en particulier aux quadrirotors.

I.3. Présentation et principe du vol

Le quadrirotor 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, comme le montre la figure I.2: [Jor.09]



Figure I.2 Quadrirotors.

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

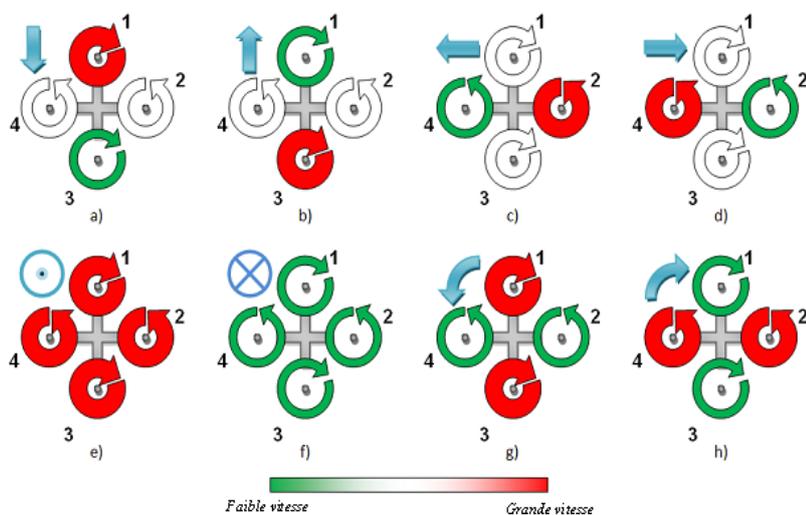


Figure I.3. Illustration du mouvement de quadrirotor.

Le fonctionnement d'un quadri-rotors 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 [Tay.04]; le mouvement vertical de montée et de descente est assuré par les quatre (04) 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, quand

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 quadrirotor offre de réels avantages par rapport à d'autres configurations d'hélicoptères et ouvre de nombreuses perspectives.

- Leurs tailles réduites et leur manoeuvrabilité 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 quadrirotor facilite sa maintenance.
- Aucun embrayage n'est exigé entre le moteur et le rotor et aucune exigence n'est donnée sur l'angle d'attaque des rotors [A.S1.08].
- 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 [A.S1.08].

I.4. L'histoire des quadrirotors

La conception des quadrirotors a évolué dans deux grandes générations. La première génération était conçue pour transporter un ou plusieurs passagers. Ces véhicules étaient parmi les premiers véhicules aériens à décollage et atterrissage vertical réussis. Caractérisés par leur grande taille et souffrent de mauvaises performances (encombrement, qualité des moteurs, matériau de conception,..).

Le premier quadrirotor était le Gyroplane n°1 « Breguet-Richet », conçu par les frères scientifiques français Louis et Jaque Breguet en collaboration avec le professeur Charles Richet en 1907 (figure.I.4). Un an plus tard, un autre quadrirotor « Gyroplane n°2 » a été construit. Le Gyroplane n°2 est équipé d'un moteur Renault, plus puissant, qui atteignit 55 CV. Malheureusement, l'aérodyne fut détruit lors d'un atterrissage un peu brusque. Le quadrirotor fut réparé mais, la suite des essais des frères Breguet ne déboucha sur aucun résultat significatif. En 1920, Etienne Oemichen a commencé ses expériences dans la conception des aéronefs à voilures tournantes. Six modèles ont été conçus par ce jeune ingénieur de la compagnie Peugeot. Parmi les quels nous constatons un grand quadrirotor de

800kg construit en 1922, avec quatre rotors de diamètre (2 x 7.60 mètres + 2 x 6.40 mètres) et huit hélices tournées par un seul moteur Gnome Rhône de 135kw (Figure I.5), [Avi.08]

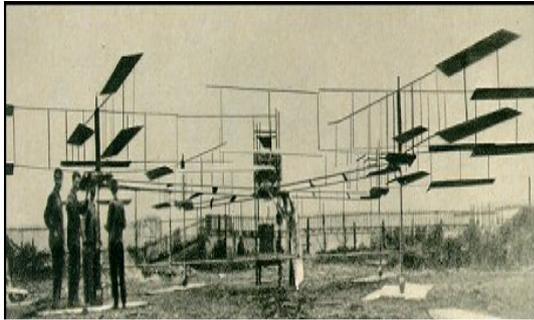


Figure I.4. Bréguet Richet Gyro1907.

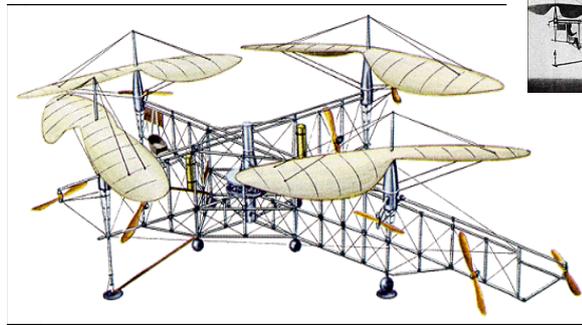


Figure I.5. Oemichen 1920.

Une autre plateforme à vol vertical été construit pour le service aérien de l'armée américaine à Dayton, Ohio en 1921 par les ingénieurs George de Bothezat et Ivan Jérôme [Avi.08]. C'était une structure en croix de 1678Kg portant six hélices de diamètre 8.1 mètres sur chaque axe de longueur 9m.

Plus de 100 essais de vols ont été exécutés sur ce grand quadrirotor, mais la commande en vol était très difficile et le prototype n'a pas répondu aux exigences d'exécution de l'armée américaine

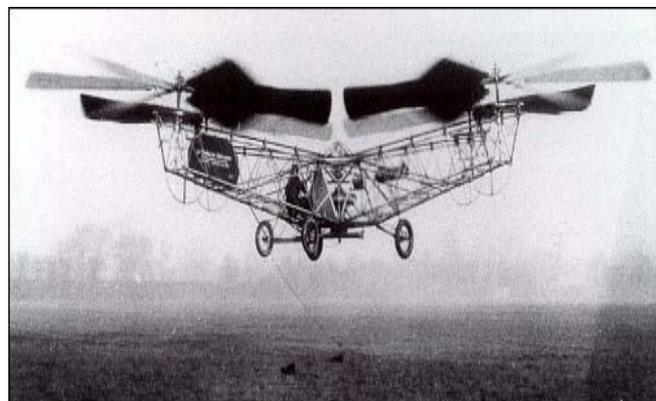


Figure I.6. Le Quadrirotor de Bothezat1922.

Dans les années 50, Amityville (New York), Convertawings a construit un quadrirotor disposant de rotors de diamètre 5.92 mètres et des ailes pour générer une portance vers l'avant. Deux moteurs « Continental » de 90 CV ont été utilisés et le véhicule a été commandé en changeant la poussée fournie par chaque rotor. Le quadrirotor de Convertawings a été piloté avec succès, mais la production a été arrêtée en raison d'un manque d'intérêt commercial pour cet avion, [Avi.08].



Figure I.7. Convertawings Model A 1956.

La configuration de quadrirotor n'a pas obtenu beaucoup d'attention jusqu'au début des années 80. Depuis, plusieurs chercheurs ont commencé à s'intéresser à la configuration « Quadrirotor » dans des applications mini drones à cause de sa simplicité, sa capacité à supporter une charge utile accrue et son coût réduit.

I.5. Les travaux de recherche récents sur les quadrirotors

Au fil du temps et grâce à l'avancement des techniques de production et l'innovation connu dans la technologie des capteurs et des calculateurs numériques, qui tend particulièrement vers la miniaturisation et la haute précision, une nouvelle génération de quadrirotors a vu le jour. Cette nouvelle génération englobe des prototypes conçus pour être des véhicules aériens sans pilote (Unmanned Aerial Vehicle, UAV), appelés également Drones.

Ces drones sont équipés d'une électronique de commande et des capteurs pour assurer leur autonomie et leur stabilité. Par conséquent, le problème de la commande de cette nouvelle génération d'UAV a connu un énorme progrès et de nouveaux axes de recherches ont été créés pour résoudre cette problématique.

I.5.1 Travaux portant sur la commande des quadrirotors

Les différents travaux de recherche effectués, dans le monde, ont montré que la configuration du quadrirotor est meilleure que d'autres configurations d'hélicoptère pour les applications UAV. Plusieurs architectures et techniques de contrôle ont été développées sur cette plateforme [Pat.06].

Un grand nombre de travaux de recherche ont été effectués sur le développement des techniques de commande sur les quadrirotors. Nous présentons ici quelques techniques utilisées dans la littérature :

- Commande utilisant le théorème de Lyapunov : Cette technique de commande permet de démontrer que le quadrirotor est asymptotiquement stable sous certaines conditions ([Bou.07] [Dzu.04], [Loz.04]).
- Le contrôleur PID : cette commande classique n'a aucune exigence sur les paramètres du modèle commandé et elle est très simple à mettre en oeuvre [Bou.04].
- Le contrôleur PD² : la puissance de ce contrôleur réside dans sa propriété de la convergence exponentielle, essentiellement due à la compensation des moments de Coriolis et des moments gyroscopiques [MCG.04].
- La commande adaptative : les techniques de commande adaptative offrent de bonnes performances, notamment avec les dynamiques non modélisables et l'incertitude des paramètres [Bre.08].
- La commande LQR : cette loi de commande a donné des résultats dans la stabilisation d'attitude du quadrirotor OS4 dans les travaux de S. Bouabdallah et André Noth. Ces résultats ont été comparés avec ceux obtenus par le contrôleur PID [Bou.04].
- L'approche Backstepping : de meilleurs résultats ont été obtenus avec cette technique de commande non linéaire, la convergence des états internes du quadrirotor a été garantie quelque soit les états initiaux. Cette technique de commande a été renforcée par la suite dans les travaux de [Bou.07] par l'ajout de l'action intégrale. Cette approche a été validée sur OS4 dans diverses expériences de vol.
- La commande « Dynamic Feedback » : cette technique a été appliquée dans quelques projets sur le quadrirotor, l'objectif est de transformer le système en boucle fermée en sous-systèmes linéaires, contrôlables et découplés [Bre.08], [Mok.04].
- La Commande par Vision : cette technique est basée sur la commande visuelle utilisant soit une caméra miniature embarquée à bord du quadrirotor, ou une caméra externe [Alt.03].

D'autres techniques de commande ont été implémentées pour la commande des quadrirotors parmi lesquelles on peut citer : Feedback Linearization, la logique floue, les réseaux de neurones [Dun.04], et l'apprentissage par renforcement dans la stabilisation et la navigation du quadrirotor [Bre.08].

I.6. les projets existants sur le quadrirotor

Le nombre de projets portant sur le problème de la conception et la commande des quadrirotors ne cesse d'augmenter. Ils existent des projets qui portent sur les problèmes de la modélisation et la commande en se basant sur des plateformes commerciales comme le Draganflyer ,UFO4, ...etc. L'objectif est de doter ces quadrirotors avec plus de capteurs et d'intelligence pour réaliser un certain degré d'autonomie. Tandis que d'autres projets ont abordé le problème de conception et à ce niveau réside un grand compromis entre la conception et la commande [Han.05].

I.6.1. Le projet Mesicopter (1999-2001)

Le Mesicopter, appelé également « Meso-Scale » est un nano quadrirotor électrique de 1.5cm d'envergure, (Figure I.8). Ce quadrirotor est capable de voler en portant sa propre alimentation fournie par des batteries miniatures et porte des capteurs dédiés à la recherche atmosphérique ou l'exploration planétaire.

Ce travail est une collaboration d'une équipe de chercheurs du département d'Aéronautique et d'Astronautique et le département de mécanique à l'université de Stanford aux USA, avec l'appui des associés industriels Intel et SRI qui est un leader dans la fabrication de batteries et de capteur miniatures de la haute technologie ainsi que la NASA.



Figure I.8. Le Mesicopter.

I.6.2. Le projet STARMAC 2007

STARMAC (Stanford Testbed of Autonomous Rotorcraft for Multi-Agent Control), un autre projet très intéressant qui étudie la commande multi – agents à l'université de Stanford d'Aéronautique et d'Astronautique, USA, [Was.07].

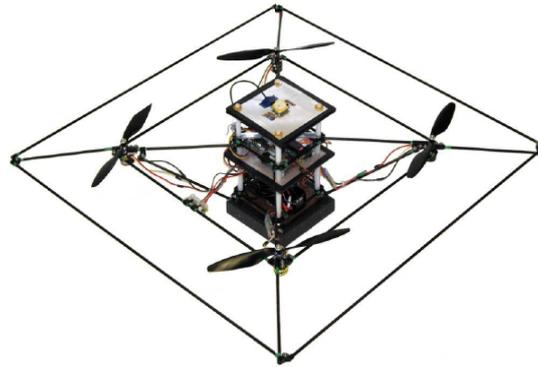


Figure I.9. Le quadrirotor STARMACII

Les quadrirotors conçus pour ce projet (figure I.9) sont des plateformes autonomes dédiés pour des applications « Outdoor » idéales pour valider de nouveaux algorithmes de coordination multi- véhicules.

I.6.3 Le projet OS4 (2003-2007)

Au Laboratoire des Systèmes Autonomes (ASL) de l'Ecole Polytechnique Fédérale de Lausanne, EPFL, Suisse, de nombreux projets ont étudié des Hélicoptères existants dans le commerce. Un modèle propre à l'EPFL a été développé. : L'Omnidirectional Stationary Flying Outstreched Robot (OS4). Le développement de OS4 a eu pour but la réalisation d'un quadrirotor avec une pleine autonomie et capable de voler dans un environnement encombrés. Dans ce projet, l'équipe a travaillé simultanément sur les aspects de la conception et de la commande. Ceci a permet de simplifier la commande par changement de conception et vice-versa [Bou.07].



Figure I.10. Le Quadri-rotors OS4.

Le quadrirotor OS4 inclut tous les dispositifs nécessaires de l'avionique et d'énergie pour un vol entièrement autonome. Le tableau ci-dessous résume les différents projets concernant la conception des quadrirotors [Bou.07] :

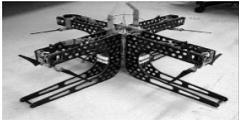
Projets	Université	Plateforme
Le X4-Flyer [Pou.04]	Université National Australienne	
Le X4-Flyer [Gue.04]	CEA France	
Le Quadrirotor du projet Quentin	Ecole Pour l'Informatique et Techniques Avancées France	
Le Quadrirotor Scott D. Hanford 2005	Université de Pennsylvanie Australie	
P. Tournier's Thèse	Institute de technique Massachusetts USA	

Tableau I.1. Quelques projets sur les quadrirotors.

I.7. Conclusion

En conclusion à ce chapitre, on constate que le monde des drones en général et celui des quadrirotors en particulier a connu ces dernières années un intérêt croissant dans la communauté scientifique vu les développements et les avancées technologiques surtout en matière d'instrumentation, ces derniers ont rendu possible la conception de systèmes performants (mini drones) doués de réelles capacités de navigation autonome à des coûts non prohibitifs ,cet avantage était l'appât qu'il fallait présenter aux investisseurs et aux constructeurs aéronautique afin qu'ils produisent ces derniers sous multiples formes dédiés à de multiples applications civiles que militaires. Cependant, les chercheurs sont toujours contraints à respecter les difficultés et les contraintes imposées par les phénomènes atmosphériques et les effets aérodynamiques qui sont généralement difficiles à estimer et à modéliser sans oublier la complexité du problème de navigation nécessitant la perception d'un environnement souvent contraint et évolutif, en particulier dans le cas des vols à basse altitude.



Chapitre II

Modélisation dynamique du quadrirotor

II.1. Introduction

La commande des robots aériens nécessite un modèle dynamique afin de pouvoir tenir en compte les effets gyroscopiques ainsi que les forces aérodynamiques [Gue.04]. Dans notre cas on s'intéresse de près au monde des drones et plus précisément à celui du quadrirotor qui est en autre terme un robot mobile aérien, En réalité les modèles dynamiques proposés pour le quadrirotor dans la littérature ne sont pas tous les mêmes et cela en fonction des tâches planifiées, Ce système est très compliqué et difficile à commander. La difficulté de commande de cet appareil vient de leur sensibilité à l'environnement extérieur et leur nature instable définis à priori par les opérateurs cela en fonction des tâches planifiées. Dans ce chapitre, nous présenterons le modèle dynamique en vue de la commande du système quadrirotor tenant en compte les différents paramètres qui peuvent affecter la dynamique d'une structure volante tel que les frottements dus aux moments aérodynamiques, les forces de traînée et les effets gyroscopiques identifiés pour un prototype expérimental existant au niveau du LISV de France (Laboratoire d'Ingénierie des Systèmes de Versailles) [Der.06].

II.2. Modèle de quadrirotor

La modélisation des robots volant est une tâche délicate puisque la dynamique du système est fortement non linéaire et pleinement couplée et afin de pouvoir comprendre au mieux le modèle dynamique développé ci-dessous, voila les différentes hypothèses de travail :

- La structure du quadrirotor est supposée rigide et symétrique d'où l'hypothèse que la matrice d'inertie est diagonale.
- Les hélices sont supposées rigides pour pouvoir négliger l'effet de leur déformation lors de la rotation.
- Le repère lié à cette structure est généralement supposé confondu avec son centre de gravité, cela nous emmène à considérer la dynamique du quadrirotor comme celle d'un corps rigide dans l'espace.
- Les forces de portance et de traînée sont proportionnelles aux carrés de la vitesse de rotation des rotors ce qui est une approximation très proche du comportement aérodynamique.
- Les conditions atmosphériques sont les conditions standard de pression et de température.

II.2.1. Les repères

Pour étudier le modèle mathématique du quadrirotor on utilise deux repères : le repère Local (figure II.1) est attaché au centre de masse $B(o', x, y, z)$ du corps du quadrirotor situé dans l'intersection des deux barres. Le repère inertiel ou global lié à la terre et supposé Galiléen défini par $E(O, X, Y, Z)$ [Bou.05], pour définir l'orientation au cours de temps du repère quadrirotor par rapport au repère inertiel on utilise les angles d'Euler (ϕ, θ, ψ) qui représentent respectivement le roulis le tangage et le lacet.

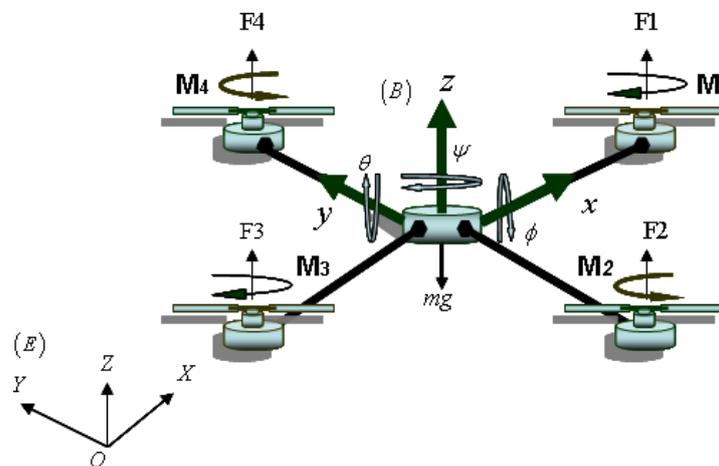


Figure II.1. Vue générale du quadrirotors.

II.2.2. Les angles d'Euler

L'orientation entre les repères peut être donnée par une matrice de rotation orthogonale, la paramétrisation de la matrice de rotation par les angles d'Euler est souvent utilisée dans les applications robotiques [Kha.02]. Comme nous sommes en présence de forces conservatives (le poids est toujours dans la direction Z du repère fixe de la Terre $E(O, X, Y, Z)$), nous avons donc besoin de la matrice de rotation 3D pour les ramener dans le repère mobile $B(o', x, y, z)$. En effet les forces de poids seront ainsi exprimées à l'aide des angles qui représentent respectivement :

- L'angle du Roulis ϕ tel que $-\frac{\pi}{2} < \phi < \frac{\pi}{2}$
- L'angle de Tangage θ tel que $-\frac{\pi}{2} < \theta < \frac{\pi}{2}$
- L'angle du Lacet tel que $-\pi < \psi < \pi$

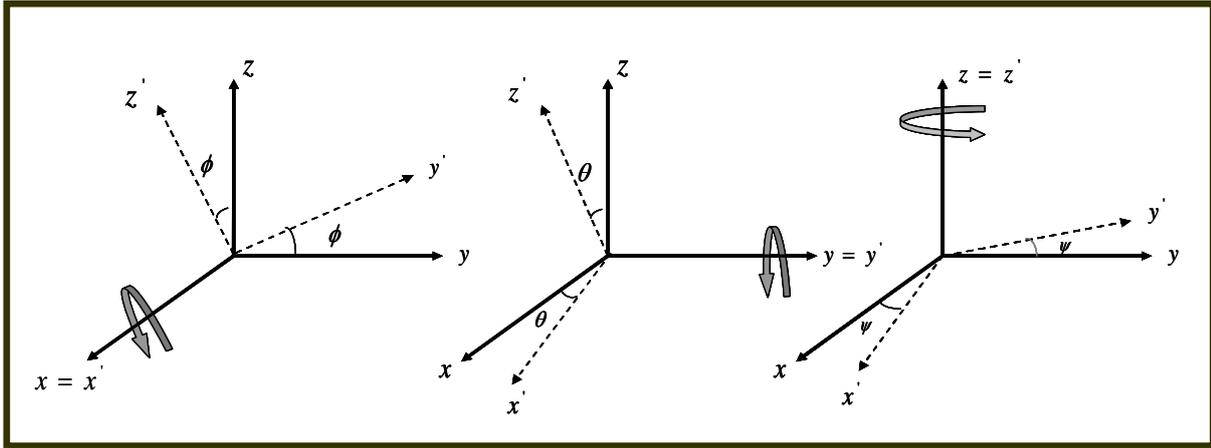


Figure II.2. Les trois rotations roulis, tangage, et lacet.

On peut facilement établir les trois sous matrices élémentaires et effectuer la multiplication suivante :

$$R = Rot_z(\psi) \times Rot_y(\theta) \times Rot_x(\phi) \quad (\text{II.1})$$

$$R = \begin{bmatrix} c\psi & -s\psi & 0 \\ s\psi & c\psi & 0 \\ 0 & 0 & 1 \end{bmatrix} \times \begin{bmatrix} c\theta & 0 & s\theta \\ 0 & 1 & 0 \\ -s\theta & 0 & c\theta \end{bmatrix} \times \begin{bmatrix} 1 & 0 & 0 \\ 0 & c\phi & -s\phi \\ 0 & s\phi & c\phi \end{bmatrix} \quad (\text{II.2})$$

$$R = \begin{bmatrix} c\psi c\theta & s\psi s\theta c\psi - s\psi c\phi & c\psi s\theta c\psi + s\psi s\phi \\ s\psi c\theta & s\psi s\theta s\psi + c\psi c\theta & c\psi s\theta s\psi - s\psi c\psi \\ -s\theta & s\phi c\theta & c\phi c\theta \end{bmatrix} \quad (\text{II.3})$$

Avec : $c = \cos$ et $s = \sin$

II.2.3. Vitesses angulaires

Les vitesses $\Omega_1, \Omega_2, \Omega_3$ sont les vitesses angulaires exprimées dans le repère fixe tel que :

$$\Omega = \begin{bmatrix} \Omega_1 \\ \Omega_2 \\ \Omega_3 \end{bmatrix} = \begin{bmatrix} \dot{\phi} \\ 0 \\ 0 \end{bmatrix} + Rot_x(\phi)^{-1} \begin{bmatrix} 0 \\ \dot{\theta} \\ 0 \end{bmatrix} + (Rot_y(\theta) Rot_x(\phi))^{-1} \begin{bmatrix} 0 \\ 0 \\ \dot{\psi} \end{bmatrix} \quad (\text{II.4})$$

En effet, la rotation en roulis a lieu lorsque les repères sont encore confondus. Puis, en ce qui concerne le tangage, le vecteur représentant la rotation doit être exprimé dans le repère fixe.

Il est donc multiplié par $Rot_x(\phi)^{-1}$. De même, le vecteur représentant la rotation en lacet doit être exprimé dans le repère fixe qui a déjà subi deux rotations. On arrive ainsi à:

$$\Omega = \begin{bmatrix} \dot{\phi} \\ 0 \\ 0 \end{bmatrix} + \begin{bmatrix} 0 \\ \dot{\theta}c\phi \\ -\dot{\theta}s\phi \end{bmatrix} + \begin{bmatrix} -\dot{\psi}s\theta \\ \dot{\psi}s\phi c\theta \\ \dot{\psi}c\phi c\theta \end{bmatrix} = \begin{bmatrix} \dot{\phi} - \dot{\psi}s\theta \\ \dot{\theta}c\phi + \dot{\psi}s\phi c\theta \\ \dot{\psi}c\phi c\theta - \dot{\theta}s\phi \end{bmatrix} \quad (\text{II.5})$$

$$\Omega = \begin{bmatrix} 1 & 0 & -s\theta \\ 0 & c\phi & s\phi c\theta \\ 0 & -s\phi & c\phi c\theta \end{bmatrix} \times \begin{bmatrix} \dot{\phi} \\ \dot{\theta} \\ \dot{\psi} \end{bmatrix} \quad (\text{II.6})$$

Dans le cas ou le quadrirotor réalise des mouvements angulaires de faibles amplitudes :

$$\begin{cases} c\phi = c\theta = c\psi = 1 \\ s\phi = s\theta = s\psi = 0 \end{cases}$$

Le vecteur Ω peut être assimilé à $[\dot{\phi} \ \dot{\theta} \ \dot{\psi}]^T$.

II.2.4. Vitesses linéaires

Les vitesses linéaires v_x^b, v_y^b, v_z^b dans le repère fixe sont en fonction des vitesses linéaires v_x^m, v_y^m, v_z^m dans le repère mobile tel que :

$$v = \begin{bmatrix} v_x^b \\ v_y^b \\ v_z^b \end{bmatrix} = R \times \begin{bmatrix} v_x^m \\ v_y^m \\ v_z^m \end{bmatrix} \quad (\text{II.7})$$

R : la matrice de rotation.

II.2.5. Les effets physiques agissants sur le quadrirotor

A. Les forces

Les forces agissant sur le système sont:

- **Le poids du quadri-rotors** : il est donné par

$$P = mg$$

m : la masse totale et g : l'accélération de la pesanteur.

- **Les forces de poussée « portance »:** qui sont des forces provoquées par la rotation des moteurs, elles sont perpendiculaires sur le plan des hélices. Ces forces sont proportionnelles au carré de la vitesse de rotation des moteurs :

$$F_i = b\omega_i^2 \quad (\text{II.8})$$

Avec $i = \overline{1:4}$, b : est le coefficient de portance, il dépend de la forme et le nombre des pales et la densité de l'air.

- **Les forces de traînée « résistance »:** la force de traînée est le couplage entre une force de pression et la force de frottement visqueux, dans ce cas on a deux forces de traînée agissant sur le système :

- La traînée dans les hélices : elle agisse sur les pales, elle est proportionnelle à la densité de l'air, à la forme des pales et au carré de la vitesse de rotation de l'hélice, elle est donnée par la relation suivante :

$$T_i = d\omega_i^2 \quad (\text{II.9})$$

Avec $i = \overline{1:4}$ d : est le coefficient de traînée.

- La traînée selon les axes (x, y, z) : elle est due au mouvement du corps du quadrirotor

$$F_t = K_{ft}v \quad (\text{II.10})$$

K_{ft} : Le coefficient de force de traînée de translation.

v : La vitesse linéaire.

B. Les moments

Il y a plusieurs moments agissants sur le quadrirotor, ces moments sont dus aux forces de poussée et de traînée et aux effets gyroscopiques.

- **Moments dus aux forces de poussée des hélices**

- La rotation autour de l'axe x : elle est due au moment créé par la différence entre les forces de portance des rotors 2 et 4, ce moment est donné par la relation suivante :

$$M_x = l(F_4 - F_2) = lb(\omega_4^2 - \omega_2^2) \quad (\text{II.11})$$

l : est la distance entre le centre de masse du quadrirotor et l'axe de rotation du rotor.

- La rotation autour de l'axe y : elle est due au moment créé par la différence entre les forces de portance des rotors 1 et 3, ce moment est donné par la relation suivante :

$$M_y = l(F_3 - F_1) = lb(\omega_3^2 - \omega_1^2) \quad (\text{II.12})$$

- **Moments dus aux forces de traînée des hélices**

- La rotation autour de l'axe z : elle est due à un couple réactif provoqué par les couples de traînée dans chaque hélice, ce moment est donné par la relation suivante :

$$M_z = d(\omega_1^2 - \omega_2^2 + \omega_3^2 - \omega_4^2) \quad (\text{II.13})$$

- Moment résultant des frottements aérodynamiques, il est donné par :

$$M_a = K_{fa} \Omega^2 \quad (\text{II.14})$$

K_{fa} : Le coefficient des frottements aérodynamiques et Ω est la vitesse angulaire.

C. Effet gyroscopique

L'effet gyroscopique se définit comme la difficulté de modifier la position ou l'orientation du plan de rotation d'une masse tournante [Ped.05]. L'effet gyroscopique est ainsi nommé en référence au mode de fonctionnement du gyroscope, appareil de contrôle de mouvement utilisé dans l'aviation (du grec *gyro* qui signifie rotation et *scope*, observer).

Physiquement, l'effet gyroscopique est une conséquence directe de l'énergie cinétique de rotation, cette grandeur étant proportionnelle à la masse, au carré de son rayon et au carré de sa vitesse de rotation. Pour faire simple, on peut comparer l'énergie cinétique d'une masse en rotation avec l'énergie cinétique d'une masse en translation et la difficulté de modifier l'orientation du plan de rotation avec la difficulté d'arrêter une masse en mouvement.

L'effet gyroscopique peut être mis en évidence en tenant une roue de vélo à bout de bras : on constate qu'il est plus facile de modifier le plan de rotation de la roue si celle-ci ne tourne pas.

Inversement, plus la roue tourne vite, plus il est difficile d'opérer cette modification. La même observation serait faite si on augmentait la masse de la roue.

Dans notre cas il y a deux moments gyroscopiques, le premier est dans le moment gyroscopique des hélices et l'autre le moment gyroscopique dû aux mouvements de quadrirotor.

- Moment gyroscopique des hélices : il est donné par la relation suivante :

$$M_{gh} = \sum_{i=1}^4 \Omega \wedge J_r \left[0 \quad 0 \quad (-1)^{i+1} \omega_i \right]^T \quad (\text{II.15})$$

avec : J_r est l'inertie des rotors.

- Moment gyroscopique dû aux mouvements de quadrirotor : il est donné par la relation suivante :

$$M_{gm} = \Omega \wedge J \Omega \quad (\text{II.16})$$

avec : J est l'inertie du système.

II.2.6. Développement du modèle mathématique selon Newton-Euler

Après avoir décrit les différentes équations on peut maintenant élaborer le modèle mathématique en utilisant la formulation de Newton-Euler, les équations sont écrites sous la forme suivante:

$$\begin{cases} \dot{\zeta} = v \\ m\ddot{\zeta} = F_f + F_t + F_g \\ \dot{R} = RS(\Omega) \\ J\dot{\Omega} = -\Omega \wedge J\Omega + M_f - M_a - M_{gh} \end{cases} \quad (\text{II.17})$$

Avec :

ζ : Est le vecteur de position du quadrirotor $\zeta = [x \quad y \quad z]^T$,

m : la masse totale du quadrirotor,

J : matrice d'inertie symétrique de dimension (3x3), elle est donnée par :

$$J = \begin{bmatrix} I_x & 0 & 0 \\ 0 & I_y & 0 \\ 0 & 0 & I_z \end{bmatrix}, \quad (\text{II.18})$$

Ω : La vitesse angulaire exprimée dans le repère inertiel « fixe ».

R : La matrice de rotation,

\wedge : Le produit vectoriel,

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

$$S(\Omega) = \begin{bmatrix} 0 & -\Omega_3 & \Omega_2 \\ \Omega_3 & 0 & -\Omega_1 \\ -\Omega_2 & \Omega_1 & 0 \end{bmatrix} \quad (\text{II.19})$$

F_f : est la force totale générée par les quatre rotors, elle est donnée par :

$$F_f = R \times \left[0 \quad 0 \quad \sum_{i=1}^4 F_i \right]^T \quad (\text{II.20})$$

$$F_i = b\omega_i^2$$

F_t : la force de traînée selon les axes (x, y, z) , elle est donnée par :

$$F_t = \begin{bmatrix} -K_{f_{tx}} & 0 & 0 \\ 0 & -K_{f_{ty}} & 0 \\ 0 & 0 & -K_{f_{tz}} \end{bmatrix} \dot{\zeta} \quad (\text{II.21})$$

$K_{f_{tx}}, K_{f_{ty}}, K_{f_{tz}}$: Les coefficients de traînée de translation,

F_g : force de gravité, elle est donnée par :

$$F_g = \begin{bmatrix} 0 \\ 0 \\ -mg \end{bmatrix}, \quad (\text{II.22})$$

M_f : moment provoqué par les forces de poussée et de traînée.

$$M_f = \begin{bmatrix} l(F_4 - F_2) \\ l(F_3 - F_1) \\ d(\omega_1^2 - \omega_2^2 + \omega_3^2 - \omega_4^2) \end{bmatrix} \quad (\text{II.23})$$

M_a : moment résultant des frottements aérodynamiques, il est donnée par :

$$M_a = \begin{bmatrix} K_{f_{ax}} \dot{\phi}^2 \\ K_{f_{ay}} \dot{\theta}^2 \\ K_{f_{az}} \dot{\psi}^2 \end{bmatrix} \quad (\text{II.24})$$

$K_{f_{ax}}, K_{f_{ay}}, K_{f_{az}}$: Les coefficients des frottements aérodynamiques.

A. Equations de mouvement de translation

On a :

$$m\ddot{\zeta} = F_f + F_t + F_g$$

$$m \begin{bmatrix} \ddot{x} \\ \ddot{y} \\ \ddot{z} \end{bmatrix} = \begin{bmatrix} c\phi c\psi s\theta + s\phi s\psi \\ c\phi s\theta s\psi - s\phi c\psi \\ c\phi c\theta \end{bmatrix} \sum_{i=1}^4 F_i - \begin{bmatrix} K_{ftx} \dot{x} \\ K_{fty} \dot{y} \\ K_{ftz} \dot{z} \end{bmatrix} - \begin{bmatrix} 0 \\ 0 \\ mg \end{bmatrix} \quad (\text{II.25})$$

On obtient alors les équations différentielles qui définissent le mouvement de translation :

$$\begin{cases} \ddot{x} = \frac{1}{m} (c\phi c\psi s\theta + s\phi s\psi) \left(\sum_{i=1}^4 F_i \right) - \frac{K_{ftx}}{m} \dot{x} \\ \ddot{y} = \frac{1}{m} (c\phi s\theta s\psi - s\phi c\psi) \left(\sum_{i=1}^4 F_i \right) - \frac{K_{fty}}{m} \dot{y} \\ \ddot{z} = \frac{1}{m} (c\phi c\theta) \left(\sum_{i=1}^4 F_i \right) - \frac{K_{ftz}}{m} \dot{z} - g \end{cases} \quad (\text{II.26})$$

B. Equations de mouvement de rotation

On a :

$$J\dot{\Omega} = -\Omega \wedge J\Omega + M_f - M_a - M_{gh} \quad (\text{II.27})$$

Si on remplace par les équations on obtient la forme suivante :

$$\begin{bmatrix} I_x & 0 & 0 \\ 0 & I_y & 0 \\ 0 & 0 & I_z \end{bmatrix} \begin{bmatrix} \ddot{\phi} \\ \ddot{\theta} \\ \ddot{\psi} \end{bmatrix} = - \begin{bmatrix} \dot{\phi} \\ \dot{\theta} \\ \dot{\psi} \end{bmatrix} \wedge \left(\begin{bmatrix} I_x & 0 & 0 \\ 0 & I_y & 0 \\ 0 & 0 & I_z \end{bmatrix} \begin{bmatrix} \dot{\phi} \\ \dot{\theta} \\ \dot{\psi} \end{bmatrix} \right) - \begin{bmatrix} K_{fax} \dot{\phi}^2 \\ K_{fay} \dot{\theta}^2 \\ K_{faz} \dot{\psi}^2 \end{bmatrix} \\ + \begin{bmatrix} lb(\omega_4^2 - \omega_2^2) \\ lb(\omega_3^2 - \omega_1^2) \\ d(\omega_1^2 - \omega_2^2 + \omega_3^2 - \omega_4^2) \end{bmatrix} - \begin{bmatrix} J_r \Omega_r \dot{\theta} \\ -J_r \Omega_r \dot{\phi} \\ 0 \end{bmatrix} \quad (\text{II.28})$$

ça conduit aux équations différentielles définissant le mouvement de rotation :

$$\begin{cases} I_x \ddot{\phi} = -\dot{\theta} \dot{\psi} (I_z - I_y) - K_{fax} \dot{\phi}^2 - J_r \Omega_r \dot{\theta} + lb(\omega_4^2 - \omega_2^2) \\ I_y \ddot{\theta} = \dot{\phi} \dot{\psi} (I_z - I_x) - K_{fay} \dot{\theta}^2 + J_r \Omega_r \dot{\phi} + lb(\omega_3^2 - \omega_1^2) \\ I_z \ddot{\psi} = -\dot{\phi} \dot{\theta} (I_y - I_x) - K_{faz} \dot{\psi}^2 + d(\omega_1^2 - \omega_2^2 + \omega_3^2 - \omega_4^2) \end{cases} \quad (\text{II.29})$$

À signaler que $u_1 u_2 u_3 u_4$ sont les entrées de commande (de translation selon l'axe « z, x, y » et la rotation selon lacet). du système et qui s'écrivent en fonction des vitesses angulaires des quatre rotors comme suit [Non.10] :

$$\begin{bmatrix} u_1 \\ u_2 \\ u_3 \\ u_4 \end{bmatrix} = \begin{bmatrix} b & b & b & b \\ 0 & -lb & 0 & lb \\ -lb & 0 & lb & 0 \\ d & -d & d & -d \end{bmatrix} \begin{bmatrix} \omega_1^2 \\ \omega_2^2 \\ \omega_3^2 \\ \omega_4^2 \end{bmatrix} \quad (\text{II.30})$$

$$\text{Et : } \Omega_r = \omega_1 - \omega_2 + \omega_3 - \omega_4$$

En conséquence le modèle dynamique complet qui régit le quadrirotor est le suivant :

$$\left\{ \begin{array}{l} \ddot{\phi} = \frac{(I_y - I_z)}{I_x} \dot{\theta} \dot{\psi} - \frac{J_r}{I_x} \Omega_r \dot{\theta} - \frac{K_{fax}}{I_x} \dot{\phi}^2 + \frac{1}{I_x} u_2 \\ \ddot{\theta} = \frac{(I_z - I_x)}{I_y} \dot{\phi} \dot{\psi} + \frac{J_r}{I_y} \Omega_r \dot{\phi} - \frac{K_{fay}}{I_y} \dot{\theta}^2 + \frac{1}{I_y} u_3 \\ \ddot{\psi} = \frac{(I_x - I_y)}{I_z} \dot{\theta} \dot{\phi} - \frac{K_{faz}}{I_z} \dot{\psi}^2 + \frac{1}{I_z} u_4 \\ \ddot{x} = -\frac{K_{ftx}}{m} \dot{x} + \frac{1}{m} u_x u_1 \\ \ddot{y} = -\frac{K_{fty}}{m} \dot{y} + \frac{1}{m} u_y u_1 \\ \ddot{z} = -\frac{K_{ftz}}{m} \dot{z} - g + \frac{(c\phi c\theta)}{m} u_1 \end{array} \right. \quad (\text{II.31})$$

Avec :

$$\begin{cases} u_x = (c\phi c\psi s\theta + s\phi s\psi) \\ u_y = (c\phi s\theta s\psi - s\phi c\psi) \end{cases} \quad (\text{II.32})$$

C. Dynamique du rotor

Le rotor est un ensemble constitué d'un moteur à courant continu entraînant une hélice via un réducteur ; le moteur à courant continu est régit par les équations dynamiques suivantes :

$$\begin{cases} V = ri + L \frac{di}{dt} + k_e \omega \\ k_m i = J_r \frac{d\omega}{dt} + C_s + k_r \omega^2 \end{cases} \quad (\text{II.33})$$

Les différents paramètres du moteur sont définis comme suit :

V : est la tension d'entrée du moteur.

ω_i : est la vitesse angulaire du rotor.

k_e, k_m : représentent respectivement les constantes des couples électrique et mécanique.

k_r : Est la constante du couple de charge.

r, L : La résistance du moteur et l'inductance.

J_r : L'inertie du rotor.

C_s : représente le frottement sec.

Sur ce ; le modèle choisit pour le rotor est le suivant :

$$\dot{\omega}_i = bV_i - \beta_0 - \beta_1 \omega_i - \beta_2 \omega_i^2 \quad i \in [1, 4] \quad (\text{II.34})$$

Avec :

$$\beta_0 = \frac{C_s}{J_r}, \beta_1 = \frac{k_e k_m}{r J_r}, \beta_2 = \frac{k_r}{J_r} \text{ et } b = \frac{k_m}{r J_r}.$$

II.2.7. La représentation d'état du système

Pour un système physique il existe une multitude de représentations d'état, dans notre cas on

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

$$\text{Soit : } \underline{X} = [\phi \ \dot{\phi} \ \theta \ \dot{\theta} \ \psi \ \dot{\psi} \ x \ \dot{x} \ y \ \dot{y} \ z \ \dot{z}]^T$$

On obtient la représentation d'état sous la forme $\dot{X} = f(X) + g(X, u)$:

$$\begin{cases}
 \dot{x}_1 = x_2 \\
 \dot{x}_2 = f_1(X) + g_1(X)u_2 \\
 \dot{x}_3 = x_4 \\
 \dot{x}_4 = f_2(X) + g_2(X)u_3 \\
 \dot{x}_5 = x_6 \\
 \dot{x}_6 = f_3(X) + g_3(X)u_4 \\
 \dot{x}_7 = x_8 \\
 \dot{x}_8 = f_4(X) + g_4(X)u_x \\
 \dot{x}_9 = x_{10} \\
 \dot{x}_{10} = f_5(X) + g_5(X)u_y \\
 \dot{x}_{11} = x_{12} \\
 \dot{x}_{12} = f_6(X) + g_6(X)u_1
 \end{cases} \quad (\text{II.35})$$

avec :

$$\begin{aligned}
 f_1(X) &= \frac{(I_y - I_z)}{I_x} x_4 x_6 - \frac{J_r}{I_x} \Omega_r x_4 - \frac{K_{fax}}{I_x} x_2^2, & g_1(X) &= \frac{1}{I_x} \\
 f_2(X) &= \frac{(I_z - I_x)}{I_y} x_2 x_6 + \frac{J_r}{I_y} \Omega_r x_2 - \frac{K_{fay}}{I_y} x_4^2, & g_2(X) &= \frac{1}{I_y} \\
 f_3(X) &= \frac{(I_x - I_y)}{I_z} x_2 x_4 - \frac{K_{faz}}{I_z} x_6^2, & g_3(X) &= \frac{1}{I_z} \\
 f_4(X) &= -\frac{K_{fzx}}{m} x_8, & g_4(X) &= \frac{u_1}{m} \\
 f_5(x) &= -\frac{K_{fzy}}{m} x_{10}, & g_5(x) &= \frac{u_1}{m} \\
 f_6(X) &= -\frac{K_{fz}}{m} x_{12} - g, & g_6(X) &= \frac{\cos \phi \cos \theta}{m}
 \end{aligned}$$

II.3. Conclusion

Tous les travaux sur les mini-drones nécessitent une modélisation du système, L'utilisation du formalisme de Newton-Euler nous a permis d'établir le modèle dynamique, où nous avons constatés que le quadrirotor est un système sous actionné. De plus, la complexité du modèle, la non linéarité, l'interaction entre les états du système, peuvent se voir clairement.

Les algorithmes de contrôle pour stabiliser ce type d'engin continuent à se développer. Nous supposons que beaucoup d'entre eux seront appliqués un jour car les applications pour ces véhicules aériens augmentent de jour en jour. En plus, tous ces travaux nécessitent la modélisation du système [Zem.06].

Dans le chapitre suivant, nous présenterons une structure de commande basée sur la commande par mode de glissement et la commande backstepping avec des approches d'hybridation comme la logique floue le PID et l'optimisation par l'algorithme génétique.



Chapitre III

Synthèse de lois de commande robuste

III.1. Introduction

La présence des organes non linéaires dans les boucles de contrôle; nous incite à dépendre le comportement non linéaire du système global. [Bel.07] Il s'agit de nouveaux comportements, comparativement aux systèmes linéaires dont la mise en évidence des outils d'analyse et des théories de stabilité permettent la mise en place de nombreux outils et techniques de contrôle, cependant, le contrôle non linéaire nécessite d'autres outils d'analyse et de nouvelles théories de stabilité. L'objectif principal de ce chapitre est la synthèse de lois de commande stabilisantes en termes de translation et d'orientation pour le quadrirotor ou nous avons fait appel à plusieurs techniques de commande non linéaire. [Bou.05] En ce qui concerne notre travail nous avons adopté une stratégie de commande basée principalement sur la décomposition du système d'origine en deux sous systèmes; le premier concerne la commande en position tandis que le second est celui de la commande en orientation. Le schéma ci-dessous présente la stratégie de commande adoptée [Bel.12.a] :

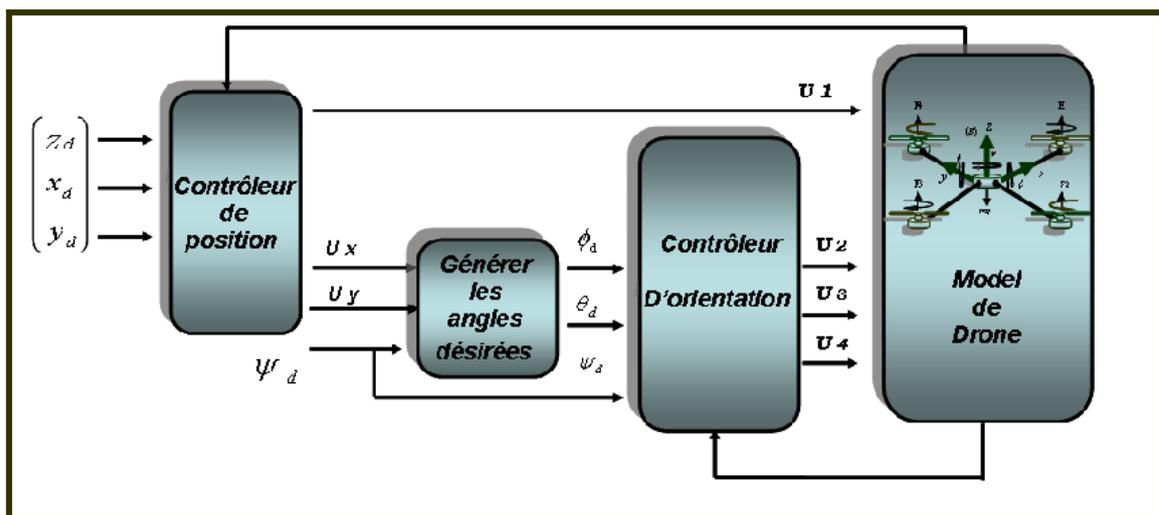


Figure III.1. Schéma synoptique de la commande proposée.

Les différentes techniques de commande non linéaire utilisées sont les suivantes :

- Le mode de glissement.
- Le mode flou glissant
- Le PI mode flou glissant avec optimisation par l'algorithme génétique « PIFSMC-GA »
- Le Backstepping.
- L'intégrale Backstepping.

Le choix des ces différentes techniques de commande n'est pas fortuit considérant les avantages majeurs qu'elles peuvent nous assurer à savoir :

- La stabilité au sens de Lyapunov.
- La robustesse et la dynamique des trajectoires désirées.
- la prise en considération de tous les non linéarités du système.

III.2. La commande par mode de glissement

La théorie des systèmes à structure variable (SSV) et les modes glissants associés a fait l'objet d'études détaillées au cours des trente dernières années [Eme.67] [Utk.77], aussi bien par les chercheurs soviétiques, que par les chercheurs des autres pays. La discussion qui a eu lieu en 1960 entre Neimark et Philipov a été concluante dans la mesure où un nouveau problème sur la théorie des modes glissants a été posé.

III.2.1. Principe de la commande par mode de glissement

L'idée de base de la commande par régime glissant est premièrement d'attirer les états du système dans une région convenablement sélectionnée, puis de concevoir une loi de commande qui maintiendra toujours le système dans cette région. En résumé, une commande par régime glissant est divisée en deux parties [Dec.88] [Utk.92] [Slo.91]:

$$u = u_{eq} + u_{gliss} \quad (\text{III.1})$$

u_{eq} : La commande équivalente ou nominale est déterminée par le modèle du système, dans ce cas il s'agit d'un modèle en immersion linéaire ou non linéaire. Cette partie est conçue avec la méthode de la commande équivalente, dont le principe est basé sur la détermination du comportement du système lorsqu'il est sur la surface de glissement s .

u_{gliss} : Le glissement est utile pour compenser les incertitudes du modèle. Il est constitué de la fonction signe « sign » de la surface de glissement s , multipliée par une constante K_{gliss} . La surface de glissement est définie dans l'espace d'état des erreurs afin de garantir la convergence des états.

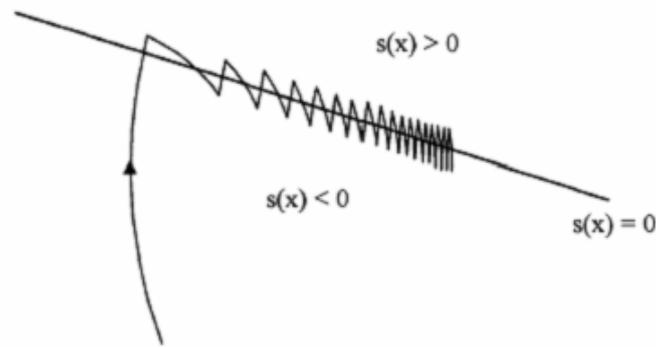


Figure III.2. Mode de glissement.

III.2.2. Notions de base de la commande mode de glissement

Soit le système donné par :

$$\dot{x} = f(x) + g(x)u \quad (\text{III.2})$$

Où $x = (x_1, \dots, x_n)^T \in X$ une variété différentiable, u représente la commande du système, f et g sont des champs de vecteur, définis sur X .

A. Surface de glissement

Pour des raisons de stabilisation et de définition d'une dynamique désirée du système dans le mode de glissement, la surface de glissement $s(x)$ peut être choisie en générale comme étant un hyperplan passant par l'origine de l'espace.

La surface s est donnée par :

$$s_i = \left[\frac{d}{dt} + \lambda_i \right]^{r-1} e_i \quad (\text{III.3})$$

Avec :

$$e_i = x_i - x_{id} \quad (\text{III.4})$$

λ_i : paramètre de la surface de glissement.

x_i : état du système.

x_{id} : état désiré.

r : degré relatif.

B. Régime glissant idéal

Ce régime correspond à une oscillation de fréquence infinie et d'amplitude nulle, d'où le point représentatif du mouvement du système glisse parfaitement sur l'hyperplan de commutation ($s(x) = 0$).

C. Régime glissant réel

La trajectoire d'état dans ce régime reste autour de l'hyperplan de glissement ($s(x) = 0$) jusqu'à au point d'équilibre.

D. Condition d'existence et d'unicité du régime glissant

Un régime glissant existe sur une surface de glissement si et seulement si, dans un voisinage de la surface de glissement toutes les trajectoires du système sont dirigées vers elle. En d'autres termes :

$$\lim_{s \rightarrow 0^-} \dot{s}(x) > 0 \quad \text{et} \quad \lim_{s \rightarrow 0^+} \dot{s}(x) < 0 \quad (\text{III.5})$$

III.2.3. Méthodes de synthèse classiques de la commande SMC

Dans ce qui suit nous allons présenter deux expressions équivalentes pour la détermination de l'état $x(t)$, qui est la solution du système en mode de glissement.

La commande u est construite de façon à ce que les trajectoires du système soient amenées vers la surface de glissement et soient ensuite maintenues dans un voisinage de celle-ci.

u est une loi de commande à structure variable définie comme suit :

$$\begin{cases} u^+(x) & \text{si } s(t,x) > 0 \\ u^-(x) & \text{si } s(t,x) < 0 \end{cases} \quad u^+ \neq u^- \quad (\text{III.6})$$

u^+ et u^- étant des fonctions continues. Il est à noter que le caractère discontinu de la loi de commande qui permet d'obtenir une convergence en temps fini sur la surface ainsi que des propriétés de robustesse vis-à-vis de certaines perturbations.

A. Méthode de Philipov

L'approche de Philipov est l'une des premières approches. Cette méthode est basée sur les résultats des travaux effectués par ce chercheur sur les équations différentielles à second membre discontinu.

Considérons le système non linéaire suivant :

$$\dot{x}(t) = f(x) + g(x)u \quad (\text{III.7})$$

Avec la stratégie de commande suivante :

$$u = \begin{cases} u^+(x) & \text{si } s(t, x) > 0 \\ u^-(x) & \text{si } s(t, x) < 0 \end{cases} \quad (\text{III.8})$$

On peut montrer à partir des travaux de Philipov que les trajectoires d'états (III.7), et (III.8) à $s(x) = 0$ sont les solutions de l'équation :

$$\dot{x} = \alpha f^+ + (1-\alpha) f^- = f^0, \text{ avec } 0 \leq \alpha \leq 1 \quad (\text{III.9})$$

où, $f^+ = f(x) + g(x)u^+$, $f^- = f(x) + g(x)u^-$ et f^0 est le vecteur de vitesse résultant de la trajectoire d'état et qui est tangentiel à la surface de glissement, il est obtenu par la moyenne géométrique comme le montre la figure suivante :

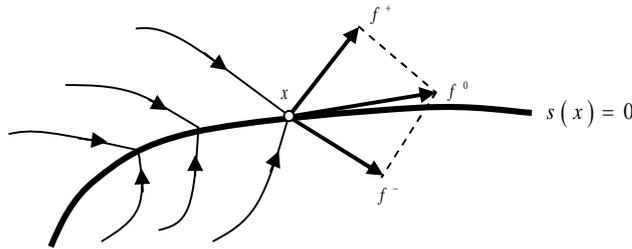


Figure III.3. Illustration de la résolution de Philipov.

En mode de glissement : $s(x) = 0$,

En dérivant $s(x) = 0$ par rapport au temps, on aura :

$$\frac{ds(x)}{dt} = \left(\frac{\partial s}{\partial x} \right)^T f^0 + \frac{\partial s}{\partial t} = \langle \nabla s, f^0 \rangle + \frac{\partial s}{\partial t} = 0 \quad (\text{III.10})$$

d'où :

$$\alpha = \frac{\langle \nabla s, f^- \rangle + \frac{\partial s}{\partial t}}{\langle \nabla s, f^- - f^+ \rangle} \quad (\text{III.11})$$

Avec :

∇s : est le gradient de s .

\langle, \rangle : est le produit scalaire.

En utilisant l'expression de α dans (III.11), l'équation caractérisant la trajectoire d'état du système en régime glissant est comme suit :

$$\frac{dx}{dt} = \left[\frac{\langle \nabla s, f^- \rangle + \frac{\partial s}{\partial t}}{\langle \nabla s, f^- - f^+ \rangle} \right] f^+ - \left[\frac{\langle \nabla s, f^+ \rangle + \frac{\partial s}{\partial t}}{\langle \nabla s, f^- - f^+ \rangle} \right] f^- = f^0 \quad (\text{III.12})$$

La méthode de Philipov est l'une des possibilités permettant la détermination de la trajectoire d'état en mode de glissement, une autre méthode, appelée méthode de la commande équivalente, a été proposée par Utkin.

B. Méthode d'Utkin (méthode de la commande équivalente)

La méthode de la commande équivalente est un moyen pour déterminer le mouvement sur la surface de glissement.

Soit le système donné par :

$$\dot{x} = f(x) + g(x)u \quad (\text{III.13})$$

en régime de glissement

$$s(x) = 0$$

et

$$\dot{s} = \frac{ds}{dt} = \left(\frac{\partial s}{\partial x} \right)^T \frac{dx}{dt} + \frac{\partial s}{\partial t} = \left(\frac{\partial s}{\partial x} \right)^T (f(x) + g(x)u_{eq}) + \frac{\partial s}{\partial t} = 0 \quad (\text{III.14})$$

Où u_{eq} est la commande équivalente. Elle est donc définie par l'équation suivante :

$$u_{eq} = - \left(\left(\frac{\partial s}{\partial x} \right)^T g(x) \right)^{-1} \left(\left(\frac{\partial s}{\partial x} \right)^T f(x) + \frac{\partial s}{\partial t} \right) \quad (\text{III.15})$$

Avec la condition :

$$\left(\left(\frac{\partial s}{\partial x} \right)^T g(x) \right) \neq 0$$

En remplaçant l'expression de u_{eq} dans (III.13), on obtient la trajectoire d'état en mode de glissement.

$$\frac{dx}{dt} = f(x) - g(x) \left(\left(\frac{\partial s}{\partial x} \right)^T g(x) \right)^{-1} \left(\left(\frac{\partial s}{\partial x} \right)^T f(x) + \frac{\partial s}{\partial t} \right) \quad (\text{III.16})$$

La commande équivalente peut être interprétée comme la valeur moyenne que prend la grandeur de commande discontinue lors de la commutation rapide entre u_{\max} et u_{\min} et comme il est représenté sur la figure (III.4)

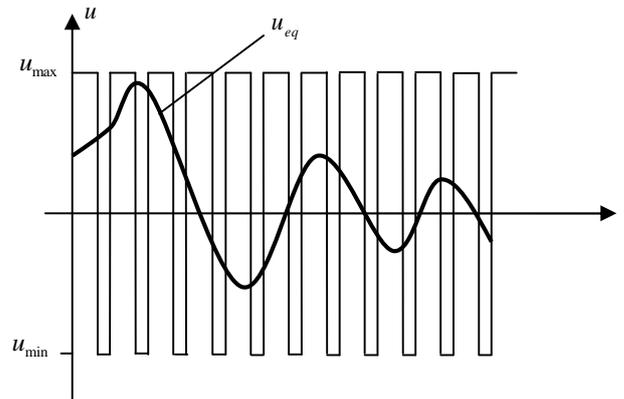


Figure III.4. Commande équivalente.

III.2.4. La réticence dans le mode de glissement (chattering)

La commande des systèmes à structure variable peut être commutée d'une valeur à une autre suivant le signe d'une fonction de commutation, avec une fréquence infinie. Cependant, dans les systèmes réels, il est impossible de réaliser une telle commutation de la commande pour des raisons technologiques telles que : l'hystérésis, la présence de retard, limitation des actionneurs...etc.

La commande discontinue engendre des oscillations du vecteur d'état et de la commande à une fréquence finie. Ce phénomène est appelé « *le phénomène de réticence* » (chattering en anglais).

Ce phénomène a plusieurs effets indésirables sur le comportement du système. Il peut :

- Exciter des dynamiques non modélisées,
- Diminuer la précision,
- Produire une grande perte d'énergie,
- Créer une fatigue des parties mécaniques mobiles.

Plusieurs techniques ont alors été proposées pour réduire ou éliminer ce phénomène. Parmi ces techniques on présente deux solutions :

A. La proposition de Slotine

Elle consiste à approximer la commande discontinue par une loi continue au voisinage de la surface, cette fonction à le nom « *sat* ». Dans ce cas la commande devient :

$$u_n = -k \text{sat}(s) \quad (\text{III.17})$$

La fonction « *sat* » est donnée par :

$$\text{sat}(s) = \begin{cases} \frac{s}{\Phi} & \text{si } |s| \leq \Phi \\ \text{sign}(s) & \text{si } |s| > \Phi \end{cases} \quad (\text{III.18})$$

qui peut être illustrée par la figure (III.5).

B. La proposition de Harshima

Dans celle-ci, la fonction signe est remplacée par une fonction de lissage appelée « *cont* ». Dans ce cas la commande devient :

$$u_n = -k \text{cont}(s) \quad (\text{III.19})$$

Avec :

$$\text{cont}(s) = \begin{cases} \frac{s}{|s| + \delta} & \text{si } |s| \leq \Phi \\ \text{sign}(s) & |s| > \Phi \end{cases} \quad \text{avec } \delta > 0 \quad (\text{III.20})$$

La figure (III.6) représente la forme de cette fonction.

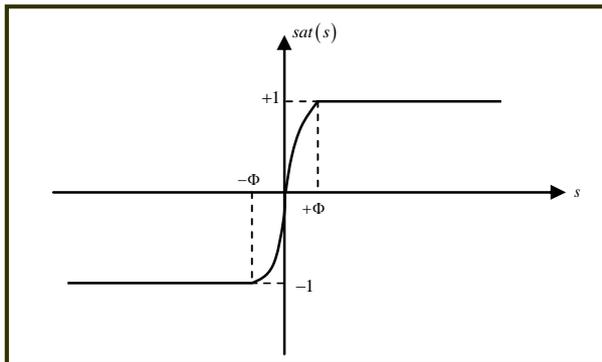


Figure III.5. Fonction de saturation *sat*.

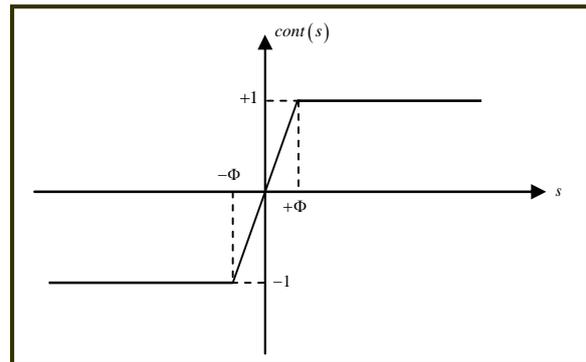


Figure III.6. Fonction de saturation *cont*.

III.2.5. Structures de contrôle par mode de glissement

Dans les systèmes à structure variable utilisant la commande par mode de glissement, on peut trouver trois configurations de base pour la synthèse des différentes commandes. La première correspond à la structure la plus simple où la commutation a lieu au niveau de

l'organe de commande lui-même. On l'appellera, structure par commutation au niveau de l'organe de commande. La deuxième structure fait intervenir la commutation au niveau d'une contre-réaction d'état. Enfin, la dernière structure est la structure par commutation au niveau de l'organe de commande avec ajout de la « *commande équivalente* ». On s'intéresse dans notre étude par cette dernière structure.

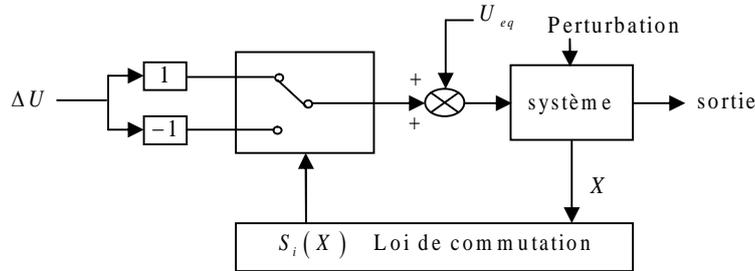


Figure III.7. Structure de régulation par ajout de la commande équivalente.

Dans ce qui suit, nous proposons un contrôleur classique à base des modes glissants, la synthèse de ce dernier est élaborée en deux étapes ; la première consiste en le choix des surfaces de glissement (s) fonctions des erreurs de poursuite, tandis que la deuxième consiste en la synthèse des lois de commande stabilisantes à partir des fonctions de Lyapunov choisies et qui peuvent assurer la condition nécessaire de glissement à savoir ($s \dot{s} < 0$).

III.3. Synthèse de la commande par mode de glissement

III.3.1. Démonstration : soit le sous système suivant pour la commande de l'angle de roulis :

$$\begin{cases} \dot{x}_1 = x_2 \\ \dot{x}_2 = \frac{(I_y - I_z)}{I_x} x_4 x_6 - \frac{J_r}{I_x} \Omega_r x_4 - \frac{K_{fax}}{I_x} x_2^2 + \frac{1}{I_x} u_2 \end{cases} \quad (\text{III.21})$$

Le degré relatif $r_2 = 2$, pour cela nous choisissons la surface de glissement comme suit :

$$s_2 = \dot{e}_2 + \lambda_2 e_2 \quad (\text{III.22})$$

Avec $\lambda_2 > 0$, $e_2 = \phi - \phi_d = x_1 - x_{1d}$, l'erreur entre l'angle de roulis ϕ et celle désirée et \dot{e}_2 sa dérivée, en utilisant la théorie de la commande par mode de glissement on trouve :

$$u_2 = u_{2eq} + u_{2gliss} \quad (\text{III.23})$$

Soit la fonction de Lyapunov suivante :

$$V(s_2) = \frac{1}{2} s_2^2 \quad (\text{III.24})$$

On présente ici une approche basée sur la fonction de Lyapunov. Le principe consiste à définir une fonction scalaire $V(s)$ positive pour chaque sortie y_i et construire une loi de commande afin que cette fonction décroît, et garantir ainsi la stabilité de la sortie et de système.

Si $\dot{V}(s_2) < 0$, alors $s_2 \dot{s}_2 < 0$, on peut dire alors que la condition nécessaire de glissement est vérifiée et la stabilité au sens de Lyapunov est garantie.

On a la dérivée temporelle de s_2 , tel que: [Hun.93]

$$\begin{cases} \dot{s}_2 = -k_2 \text{sign}(s_2) - q_2 s_2 \\ \dot{s}_2 = \ddot{e}_2 + \lambda_2 \dot{e}_2 \end{cases} \quad \text{Avec : } k_2, q_2 > 0 \quad (\text{III.25})$$

Il en résulte :

$$\begin{aligned} \dot{s}_2 &= -k_2 \text{sign}(s_2) - q_2 s_2 \\ &= \ddot{e}_2 + \lambda_2 \dot{e}_2 = \dot{x}_2 - \ddot{x}_{1d} + \lambda_2 (x_2 - \dot{x}_{1d}) \end{aligned} \quad (\text{III.26})$$

Alors :

$$u_2 = I_x \left(\ddot{x}_{1d} - \left(\frac{I_y - I_z}{I_x} \right) x_4 x_6 + \frac{J_r}{I_x} \Omega_r x_4 + \frac{K_{fax}}{I_x} x_2^2 - \lambda_2 (x_2 - \dot{x}_{1d}) - k_2 \text{sign}(s_2) - q_2 s_2 \right) \quad (\text{III.27})$$

Par conséquent :

$$\begin{cases} u_{2equ} = I_x \left(\ddot{x}_{1d} - \left(\frac{I_y - I_z}{I_x} \right) x_4 x_6 + \frac{J_r}{I_x} \Omega_r x_4 + \frac{K_{fax}}{I_x} x_2^2 - \lambda_2 (x_2 - \dot{x}_{1d}) \right) \\ u_{2gliss} = I_x (-k_2 \text{sign}(s_2) - q_2 s_2) \end{cases} \quad (\text{III.28})$$

Les mêmes étapes sont reprises pour les autres commandes :

u_1 : Commande de translation selon l'axe z.

u_3 : Commande de l'angle de tangage.

u_4 : Commande de l'angle de lacet.

u_x, u_y : Deux commandes virtuelles, pour la commande en translation suivant x et y et qui seront utilisées ensuite pour générer les consignes de rotation ϕ_d et θ_d suivant l'axe x et l'axe y et ceci pour atteindre la position désirée (x_d, y_d) .

$$\begin{aligned}
u_1 &= \frac{m}{(\cos(\phi)\cos(\theta))} \left(\ddot{x}_{11d} + \frac{K_{ftz}}{m} x_{12} + g - \lambda_1 (x_{12} - \dot{x}_{11d}) - k_1 \text{sign}(s_1) - q_1 s_1 \right) \\
u_3 &= I_y \left(\ddot{x}_{3d} - \left(\frac{I_z - I_x}{I_y} \right) x_2 x_6 - \frac{J_r}{I_y} \Omega_r x_2 + \frac{K_{fay}}{I_y} x_4^2 - \lambda_3 (x_4 - \dot{x}_{3d}) - k_3 \text{sign}(s_3) - q_3 s_3 \right) \\
u_4 &= I_z \left(\ddot{x}_{5d} - \left(\frac{I_x - I_y}{I_z} \right) x_2 x_4 + \frac{K_{faz}}{I_z} x_6^2 - \lambda_4 (x_6 - \dot{x}_{5d}) - k_4 \text{sign}(s_4) - q_4 s_4 \right) \\
u_x &= \frac{m}{u_1} \left(\ddot{x}_{7d} + \frac{K_{ftx}}{m} x_8 - \lambda_x (x_8 - \dot{x}_{7d}) - k_x \text{sign}(s_x) - q_x s_x \right) \\
u_y &= \frac{m}{u_1} \left(\ddot{x}_{9d} + \frac{K_{fty}}{m} x_{10} - \lambda_y (x_{10} - \dot{x}_{9d}) - k_y \text{sign}(s_y) - q_y s_y \right) \quad \text{avec: } (k_i, q_i) > 0
\end{aligned} \tag{III.29}$$

Dont les erreurs de poursuite sont écrites sous la forme ci-dessous :

$$\begin{aligned}
e_1 &= z - z_d = x_{11} - x_{11d} \\
e_3 &= \theta - \theta_d = x_3 - x_{3d} \\
e_4 &= \psi - \psi_d = x_5 - x_{5d} \\
e_x &= x - x_d = x_7 - x_{7d} \\
e_y &= y - y_d = x_9 - x_{9d}
\end{aligned} \tag{III.30}$$

La commande des angles de roulis et de tangage c'est une étape intermédiaire pour réaliser la commande en translation suivant l'axe x et l'axe y , les deux commandes u_x et u_y seront utilisées pour générer les angles désirées ϕ_d et θ_d .

D'après le modèle du système on a :

$$\begin{cases} u_x = (c\phi c\psi s\theta + s\phi s\psi) \\ u_y = (c\phi s\theta s\psi - s\phi c\psi) \end{cases} \tag{III.31}$$

à partir de l'équation (III.31), on trouve :

$$\begin{aligned}
\phi_d &= \arcsin(u_x \sin(\psi_d) - u_y \cos(\psi_d)) \\
\theta_d &= \arcsin((u_x \cos(\psi_d) + u_y \sin(\psi_d)) / \cos(\phi_d))
\end{aligned} \tag{III.32}$$

III.3.2. Les résultats de simulation

Dans cette section, on va présenter les résultats de simulation issus de l'application de la structure de commande mode de glissement, les trajectoires de référence utilisées sont du type cycloïdale. Avec l'état initial $X = [0, 0, 0, 0, 0, 0]^T$,

Avec : $1=100$; $2=100$; $3=100$; $4=50$; $x=0.02$; $y=0.02$;

et $k_1=0.5; k_2=0.95; k_3=0.1; k_4=0.1; k_x=0.1; k_y=0.1;$

$q_1=0.4; q_2=0.95; q_3=0.4; q_4=0.1; q_x=0.01; q_y=0.01;$

$$\eta_i(t) = \begin{cases} \eta_i(0) + \frac{D_i}{2\pi} \left[2\pi \frac{t}{t_f} - \sin\left(2\pi \frac{t}{t_f}\right) \right] & \text{pour } 0 \leq t \leq t_f \\ \eta_i(t_f) & \text{pour } t_f < t \end{cases} \quad (\text{III.33})$$

$\eta_i = (\eta_1, \eta_2, \eta_3, \eta_4) = (x, y, z, \psi)$

Avec : $D_i = \eta_i(t_f) - \eta_i(0)$: le déplacement, et t_f est l'instant final du mouvement.

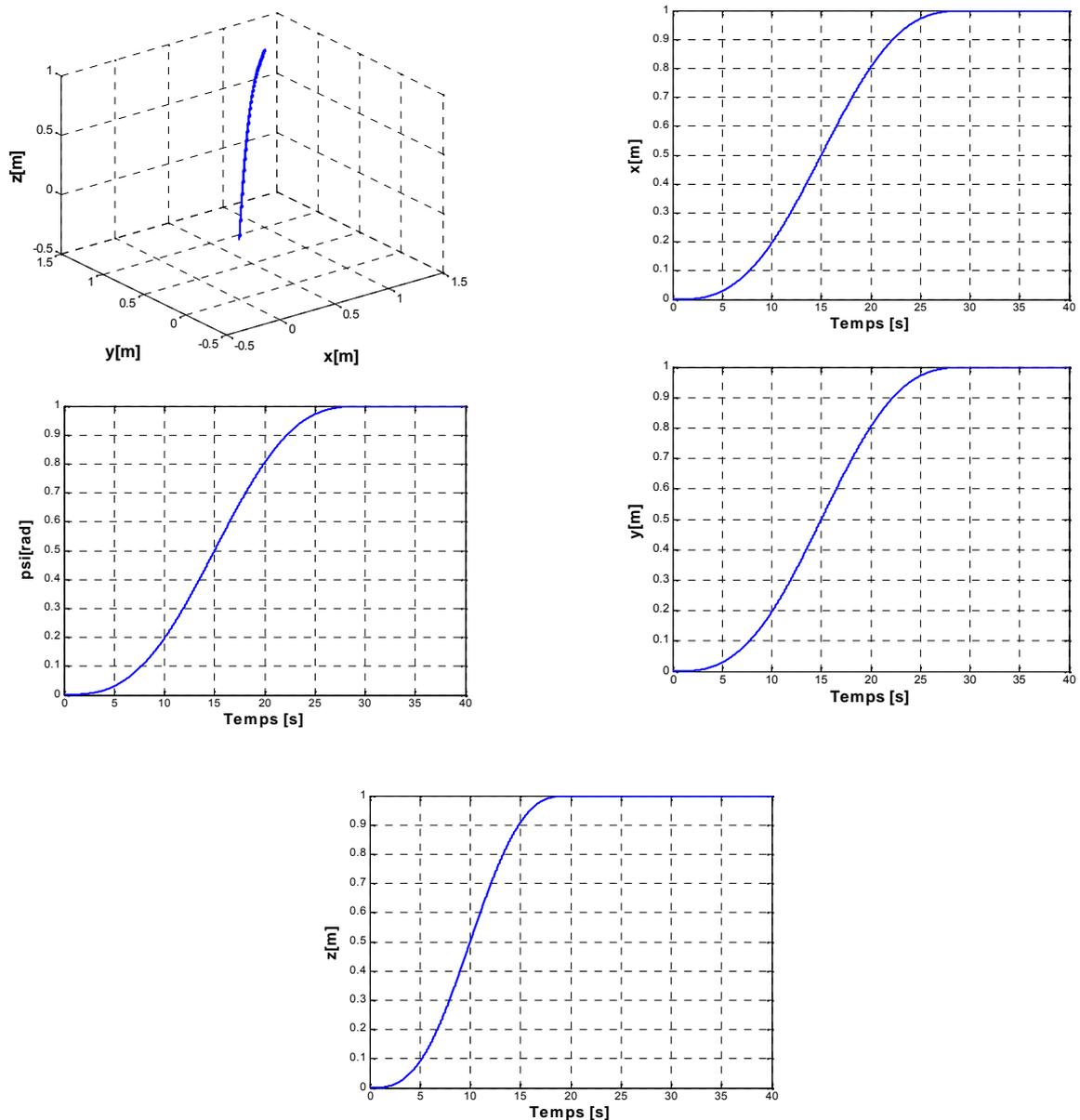


Figure III.8. Trajectoire de référence.

La figure III.9 présente les trajectoires désirées et réelles ainsi que la réalisation de mouvements z_x et y en 3D, on remarque que la commande par mode de glissement assure la stabilité et donne de très bons résultats de poursuite avec des erreurs minimale selon la figure III.11.

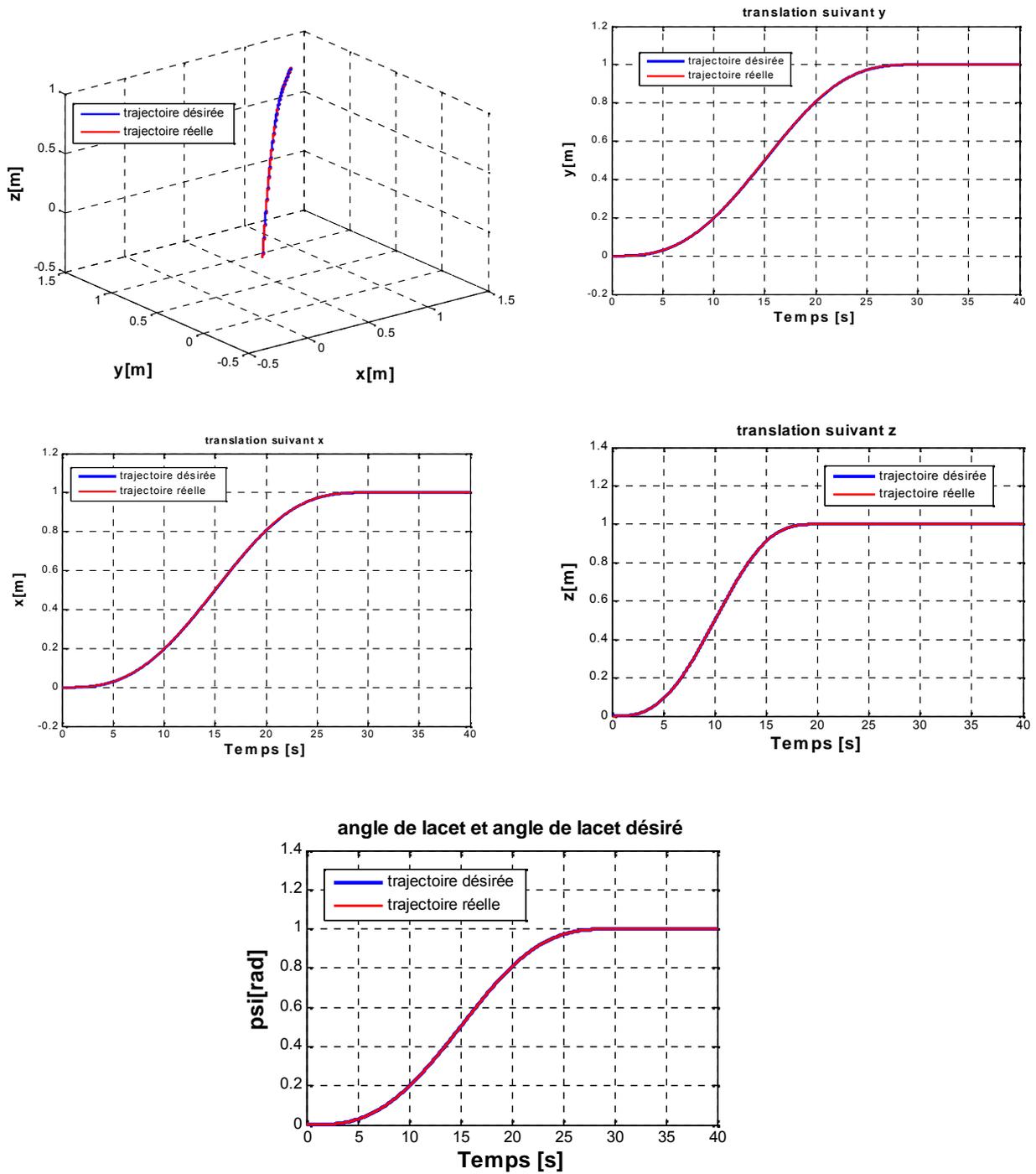


Figure III.9. Résultats de poursuite des trajectoires désirées selon (x, y, z, ψ) .

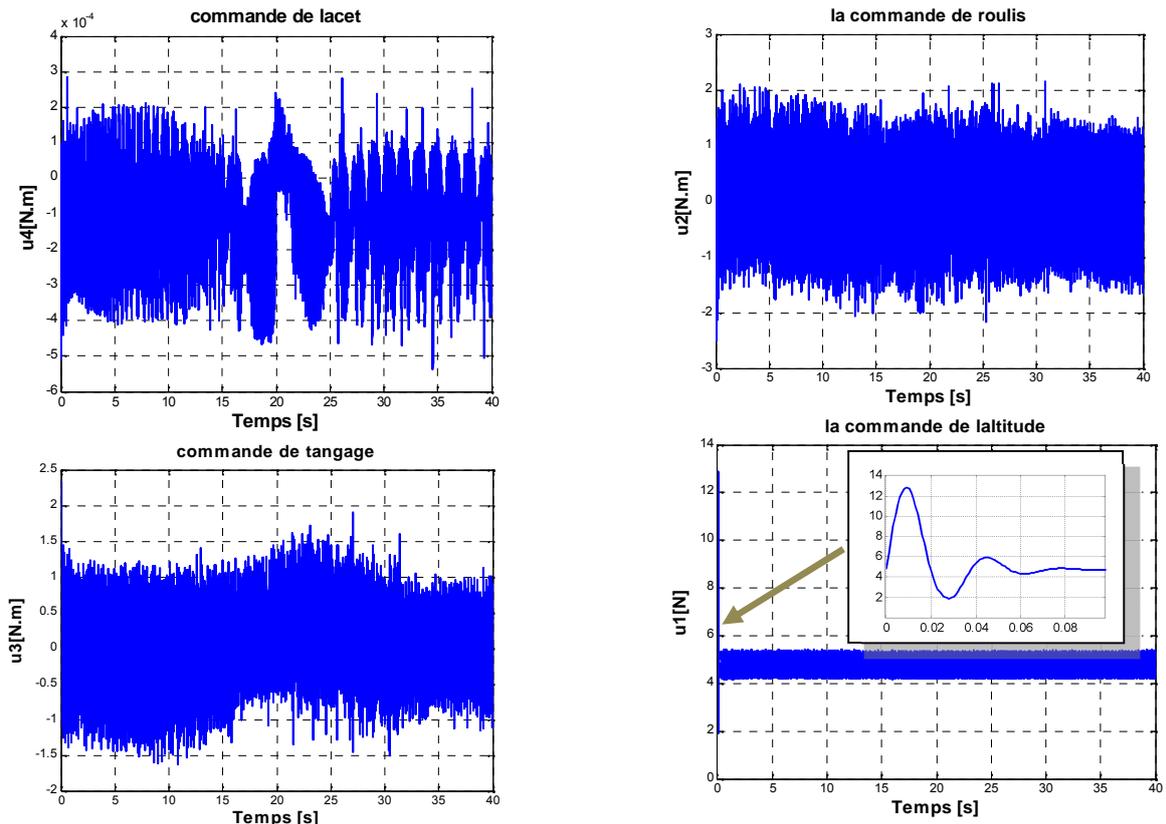


Figure III.10. Les commandes (u_1 u_2 u_3 u_4).

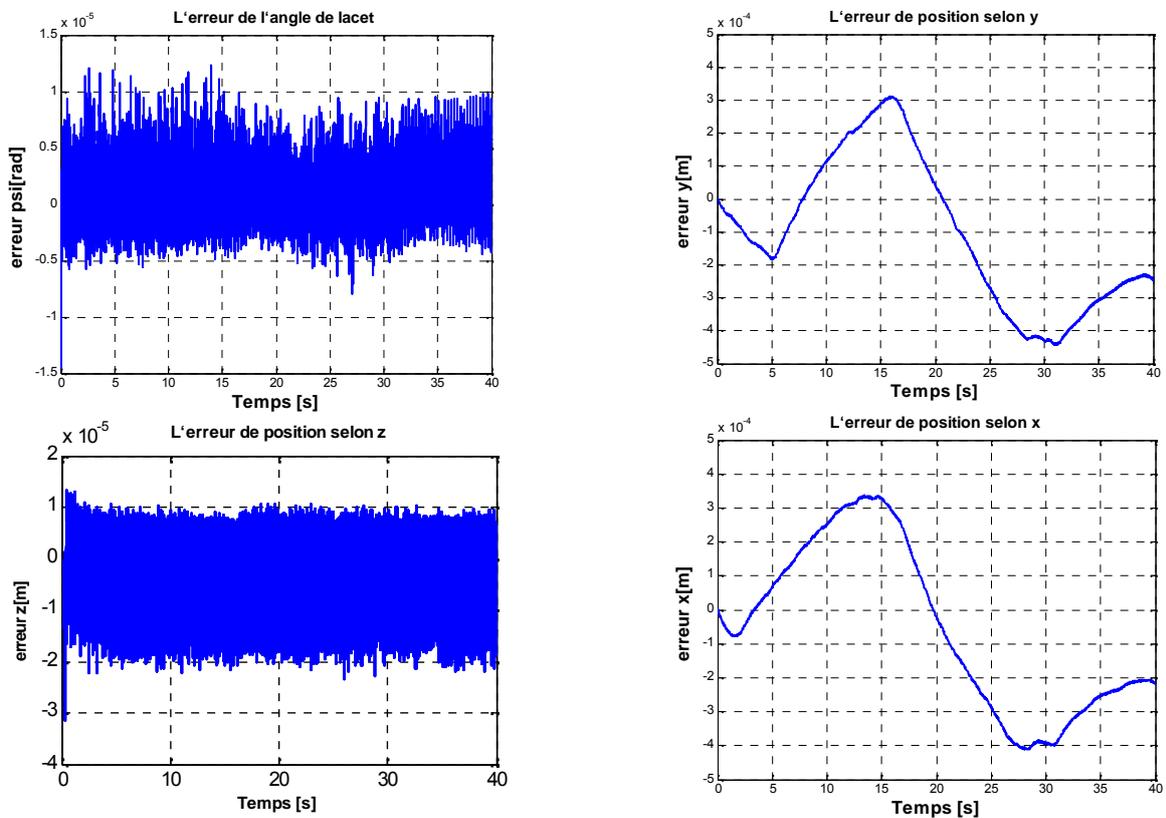


Figure III.11. Les erreurs de déplacements suivant (x , y , z) et l'angle ψ .

La figure **III.10** qui représente les commandes $(u_1 u_2 u_3 u_4)$ montre que le phénomène du chattering est toujours présent surtout dans les signaux de commande ce qui influe sur le comportement du système, pour surmonter le problème du chattering, on va utiliser un contrôleur flou.

III.3.3. Etude de la robustesse

Nous nous intéressons dans cette partie, au test de la robustesse de la commande par mode de glissement développée où nous allons imposer une force de traînée ou une résistance « vent » au mouvement de drone dans le cas de la commande du vol vertical qui entraînera une dégradation de la performance avec une consommation supplémentaire d'énergie aux niveaux des actionneurs.

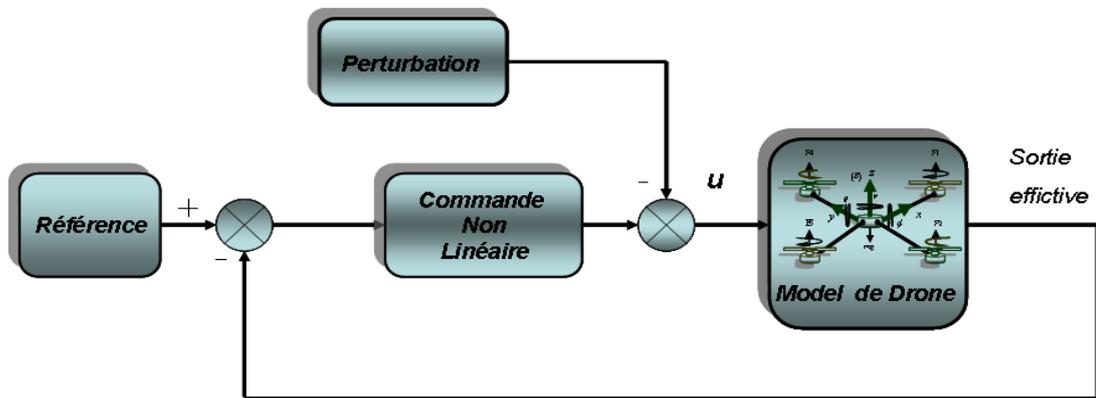


Figure III.12. Schéma global de la commande en présence de perturbation

Si on applique une force perturbatrice $D_z = 7 \sin(0.1t)$ à l'instant $t=25\text{sec}$ suivant (z) on obtient les figures suivantes:

La figures **III.13**, illustre les résultats de simulation dans le cas d'une force de traînée, nous remarquons que le contrôleur ne rejette pas la perturbation car on voit clairement une grande influence de la perturbation selon la figure de l'erreur de position (z) . Le zoom sur la figure montre très bien l'influence de la perturbation à $t=25\text{sec}$.

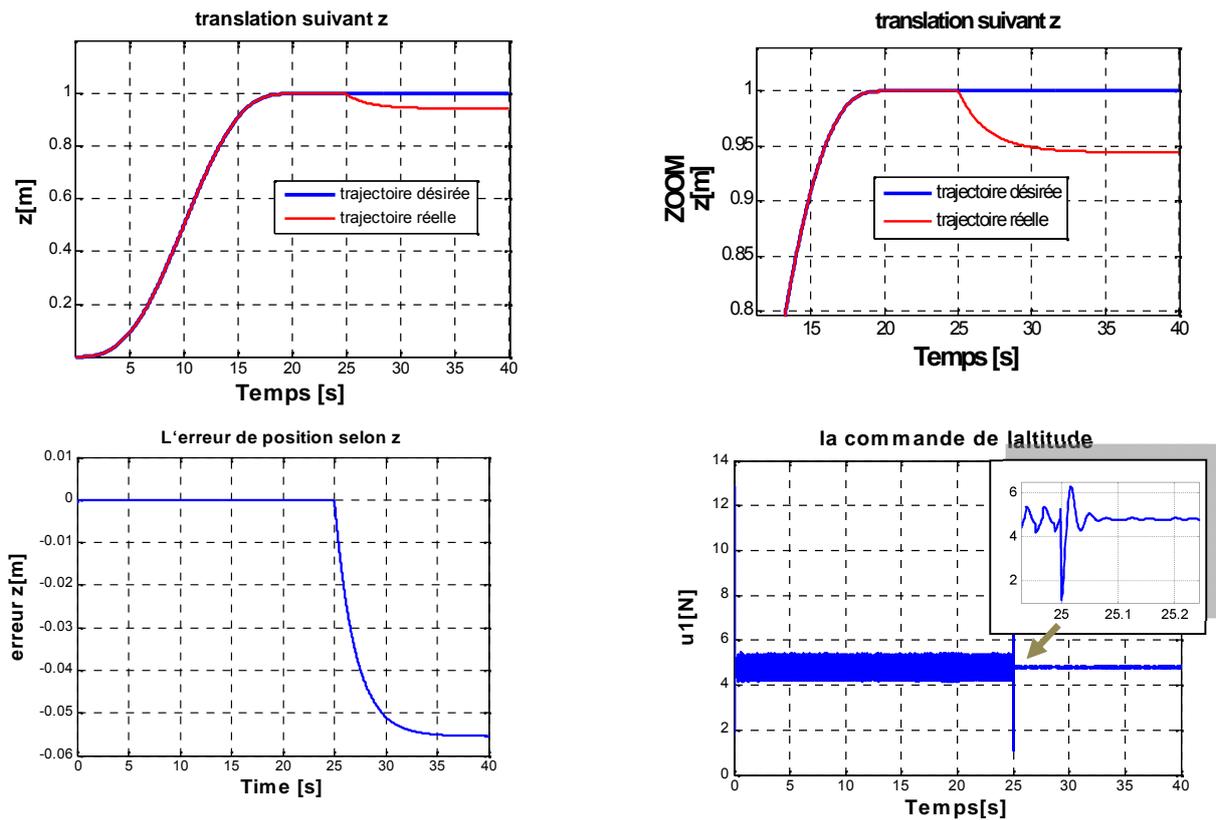


Figure III.13. L'influence de la perturbation selon (z).

III.4. La commande par mode flou glissant

La logique floue est une technique de traitement des incertitudes et a pour objet : la représentation des connaissances imprécises, elle est basée sur des termes linguistiques courants comme petit, grand, moyen... etc.

III.4.1. Quelques définitions

a) Ensemble flou

Dans un ensemble de référence E , un sous ensemble flou de ce référentiel E est caractérisé par une fonction d'appartenance μ de E dans l'intervalle des nombres réels $[0 \ 1]$ qui indique avec quel degré un élément appartient à cette classe. Un sous ensemble flou est caractérisé par un noyau, un support et une hauteur [Vil.02].

b) Noyau

C'est l'ensemble des éléments qui sont vraiment dans E : $Noy(E) = \{x / \mu_E(x) = 1\}$

c) Support

C'est l'ensemble des éléments qui sont dans E à des degrés divers.

d) Hauteur

C'est la borne supérieure de la fonction d'appartenance : $ht(E) = \text{Sup}_{(x \in E)} \mu_E(x)$

e) Ensemble normalisé

Un ensemble est dit normalisé s'il est de hauteur 1.

Exemple : Dans la figure III.14 nous indiquons un exemple de sous ensemble normalisé ainsi que son noyau, son support et sa hauteur.

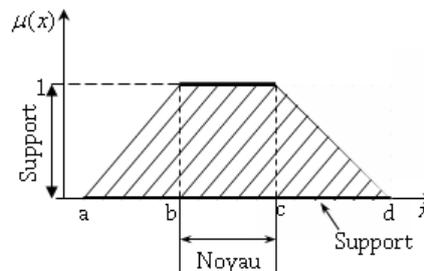


Figure III.14. Format d'un ensemble flou normalisé

III.4.2. Opérations

Les opérations possibles sur les ensembles flous sont des opérations de base existant déjà en logique booléenne en respectant quelques propriétés. Soient A et B un couple d'univers de discours, une relation floue R entre A et B est définie par : $A \times B \rightarrow [0,1]$, $(x, y) \rightarrow \mu_R(x, y)$

a) L'intersection

L'intersection de deux ensembles flous est le plus grand ensemble flou contenu dans A et dans B : $\mu_{A \cap B}(x, y) = \mu_{AND}(x, y) = \min(\mu_A, \mu_B)$

b) L'union

Une union de deux ensembles flous A et B est le plus petit ensemble flou contenant A et B : $\mu_{A \cup B}(x, y) = \mu(x, y) = \max(\mu_A, \mu_B)$

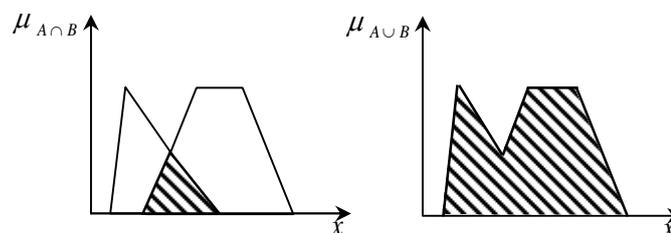


Figure III.15. Union et intersection de deux sous ensemble flous

c) La complémentation

Le complémentaire d'un sous ensemble flou A dans un ensemble de référence E est défini par la relation : $\mu_{\neg A} = 1 - \mu_A$.

III.4.3. Structure générale d'un contrôleur flou

L'avantage de la commande floue par comparaison avec les commandes classiques est qu'elle ne nécessite pas la connaissance des modèles mathématiques du système. Par contre elle a besoin d'un ensemble de règles basées essentiellement sur la connaissance d'un opérateur qualifié manipulant le système [Kac.97, Kim.98]

La conception du contrôleur flou (FLC) passe par quatre principales parties distinctes, comme le montre la figure III.16

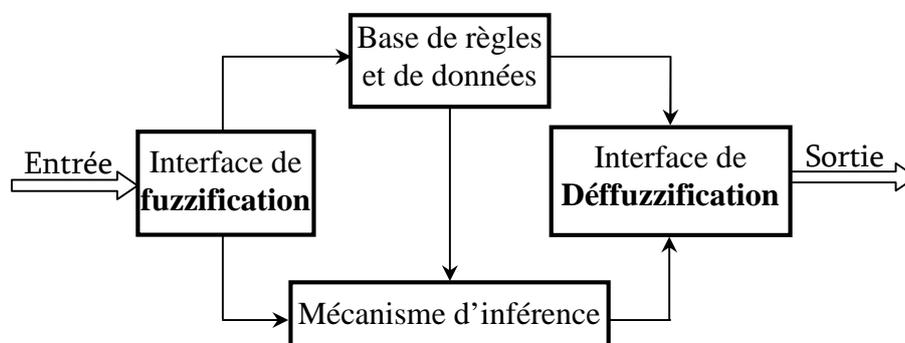


Figure III.16. Structure d'un système de contrôle flou.

a) Interface de fuzzification

Dans le domaine du contrôle, les données observées sont des grandeurs physiques générées par des capteurs. Il est nécessaire de convertir ces grandeurs réelles en des variables floues. Pour cela, on fait appel à une opération dite fuzzification, qui permet de fournir les degrés d'appartenance de la variable floue à ses sous ensembles flous en fonction de la valeur réelle de la variable d'entrée.

Chaque grandeur physique y utilisée doit être normalisée entre -1 et +1 en la divisant par $\max(y_{\max} - y_{\min})$.

On utilise en général des fonctions d'appartenance de forme triangulaire, trapézoïdale et/ou gaussienne bien qu'il en existe d'autres. Quelque soit la forme choisie, il faut prendre certaines précautions dans la construction et la disposition des fonctions d'appartenance :

Pour la variable linguistique « environ zéro » On veillera à éviter un plat au sommet (entraînant d'une indétermination du réglage).

On évite les recouvrements trop importants ou trop faibles de deux fonctions d'appartenance contiguës. On préfère les triangles et les trapèzes pour définir les fonctions d'appartenances pour gagner de l'espace mémoire et minimiser le temps de calculs.

b) Base de règles

Le système de contrôle flou comprend un nombre de règles d'inférence reliant les variables floues d'entrée d'un système aux variables floues de sortie de ce système. Ces règles se présentent sous la forme usuelle suivante :

Si condition 1 Et/Ou condition 2 (Et/Ou...) alors action sur les sorties.

L'établissement de ces règles est généralement basé sur la connaissance du problème et sur l'expérience de l'opérateur qui peut fixer le nombre de sous-ensembles, leurs fonctions d'appartenance ainsi que les variables linguistiques. Ils existent plusieurs présentations de la base de règles telles que la description linguistique, symbolique ou par une matrice d'inférence. [Kac.97]

c) Mécanisme d'inférence

Dans cette étape, il s'agit de déterminer comment le système interprète les variables linguistiques floues. Les variables linguistiques (entrées et sorties) sont liées par les règles d'inférence. Les variables sont liées par l'opérateur "ET", tandis que les variables de sortie des différentes règles sont liées par l'opérateur "OU" et l'ensemble des règles sont liées par les connecteurs tels que "ET" et "Alors". La conséquence d'une règle floue est inférée par l'emploi de règle de composition, en utilisant les fonctions d'implications floues et les connecteurs "ET" et "Alors".

Les méthodes d'inférences se différencient selon la combinaison et l'utilisation des opérateurs (ET et OU) dans les règles d'inférence. Parmi ces méthodes on trouve [Bag.99, Gou.04, Tab.04, Sab.04] :

1) Méthode d'inférence MAX-MIN

Cette méthode représente l'opérateur "ET" par la fonction "Min", la conclusion "ALORS" par la fonction "Max" et l'opérateur "OU" par la fonction "Min". La représentation graphique de cette méthode d'inférence est illustrée par la Figure III.17.

2) Méthode d'inférence Max-Produit

Cette méthode présente l'opérateur "ET" par la fonction "Min", l'opérateur "OU" par la fonction "Max" et la conclusion "Alors" par la fonction "Prod", d'où la représentation graphique de cette méthode est schématisée par la Figure III.18.

3) Méthode d'inférence Somme-Produit

Dans cette méthode, l'opérateur "ET" est représenté par la fonction "Prod", l'opérateur "OU" est représenté par la fonction "Somme" et la conclusion "Alors" est représentée par la fonction "Prod", sa représentation graphique est illustrée par la Figure III.19.

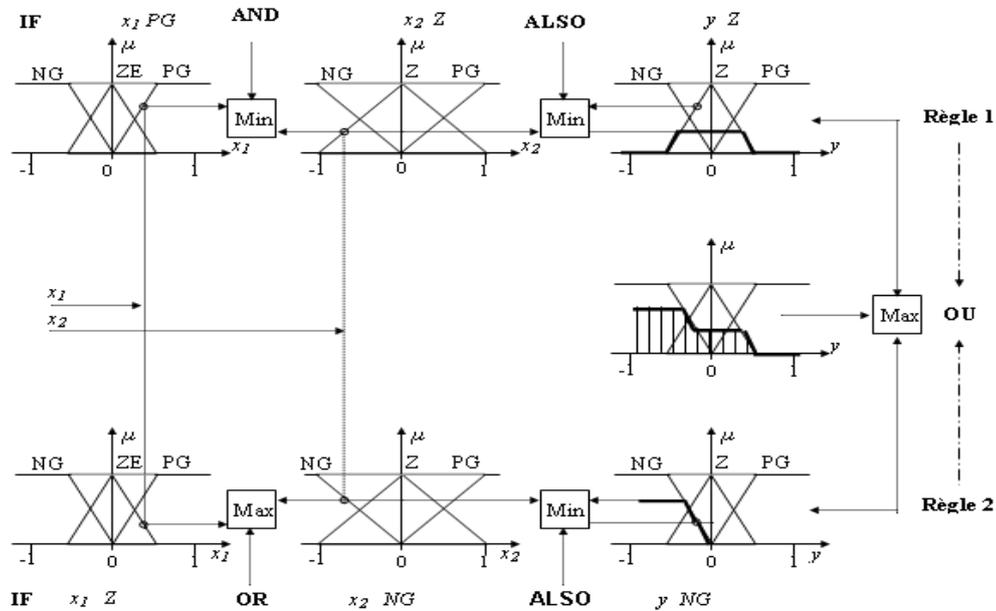


Figure III.17. Méthode d'inférence Max-Min.

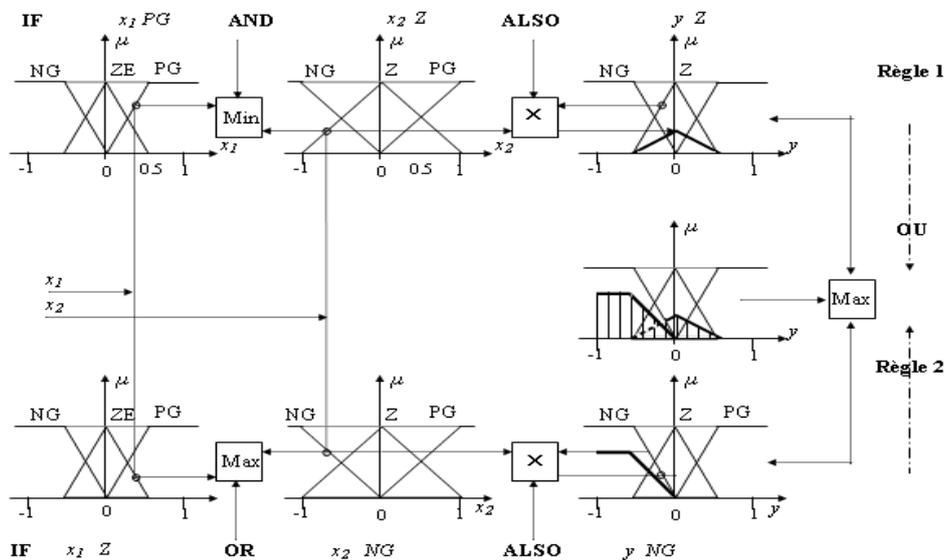


Figure III.18. Méthode d'inférence Max-Prod.

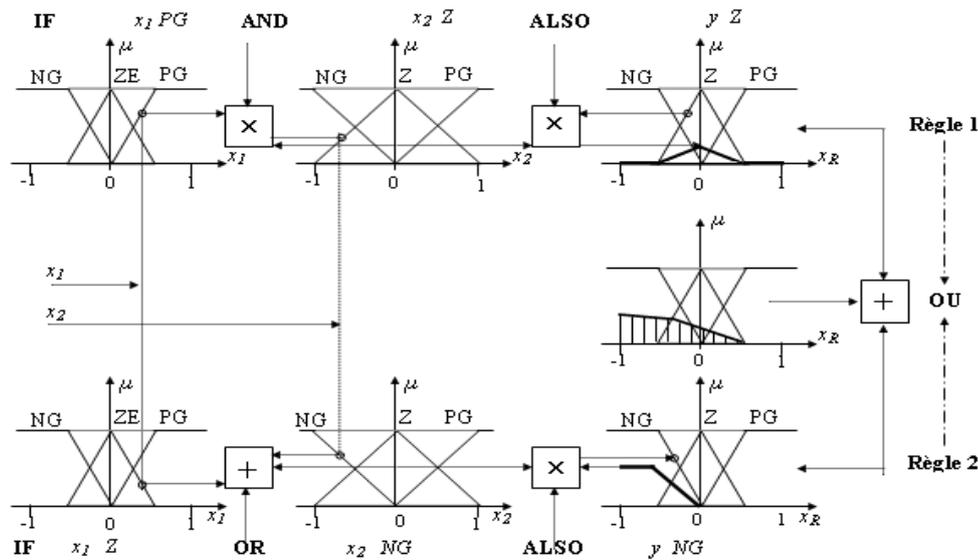


Figure III.19. Méthode d'inférence floue Somme-Produit.

d) Interface de déffuzzification

Les méthodes d'inférence génèrent une fonction d'appartenance, il faut transformer cette grandeur floue en grandeur physique réelle. L'opération de déffuzzification permet de calculer à partir des degrés d'appartenance à tous les sous-ensembles flous de la variable de sortie, la valeur de sortie à appliquer au système. Il y a plusieurs méthodes de déffuzzification à savoir la méthode du maximum, la méthode des hauteurs pondérées et la méthode du centre de gravité, cette dernière est la plus utilisée dans plusieurs travaux L'expression de la sortie dans cette méthode donnée par l'équation.

$$u = \frac{\int x \cdot u_R(x) \cdot dx}{\int u_R(x) \cdot dx} \quad (\text{III.34})$$

III.4.4. Synthèse de la commande par mode flou glissant

Dans cette approche, la logique floue et le mode de glissement sont combinés entre eux pour donner naissance à un nouveau concept de contrôleurs. Le contrôleur ainsi obtenu est fait partie de la famille des contrôleurs flou-glissant. Celui-ci présente la même structure de commande du SMC donnée dans la partie mode glissant, mis à part le terme $k \text{ sat}(s)$ qui sera remplacé par un contrôleur flou.

Il est clair que la partie discontinue de la loi de commande par mode de glissement explique la stratégie de contrôle suivante :

« **Si** l'erreur **est** négative, **alors** la sortie du système est poussée vers la direction positive ».

Pour cela, le terme $k \text{sat}(s)$ peut être remplacé par un contrôleur flou. Ce contrôleur possède une entrée et une sortie, et la base de règle sert à établir une connexion entre s et u_n « la commande discontinue ». Ceci est interprété par des règles de la forme :

R1 : Si s est GN **Alors** u_n est TG

R2 : Si s est PN **Alors** u_n est G

R3 : Si s est ZE **Alors** u_n est M

R4 : Si s est PP **Alors** u_n est P

R5 : Si s est GP **Alors** u_n est TP

Avec :

TG : Très grand.

M : Moyen.

P : Petit.

G : Grand.

TP : Très petit.

Le contrôleur flou ayant une entrée s de cinq fonctions d'appartenance et une sortie u_n de cinq fonctions d'appartenance qui sont représentées dans les figures III.20 et III.21 respectivement.

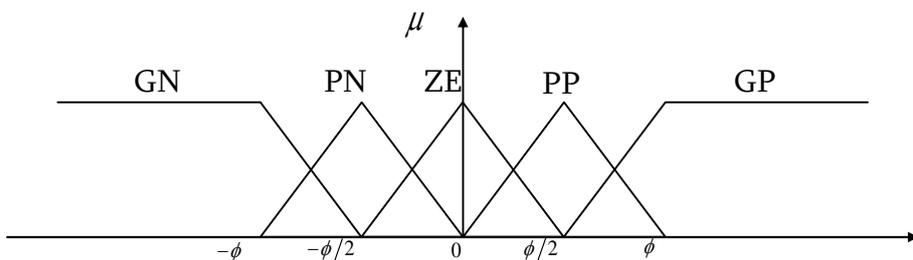


Figure III.20. Fonction d'appartenance de l'entrée du contrôleur FSMC.

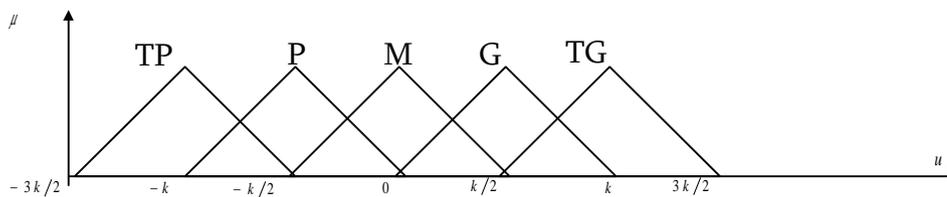


Figure III.21. Fonction d'appartenance de la sortie du contrôleur FSMC.

Le résultat de défuzzification de la sortie u_n pour une entrée s est présenté comme le montre la figure III.22.

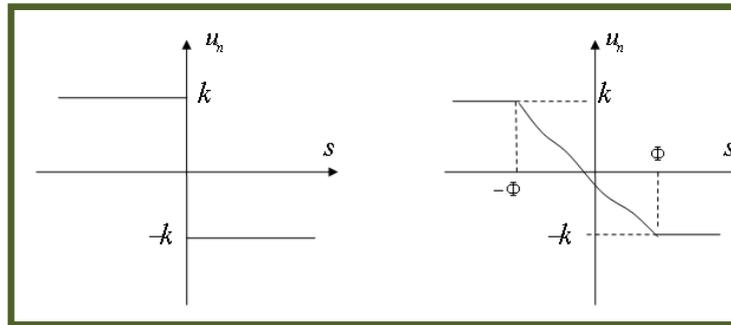


Figure III.22. Signal de commande du contrôleur flou-glissant.

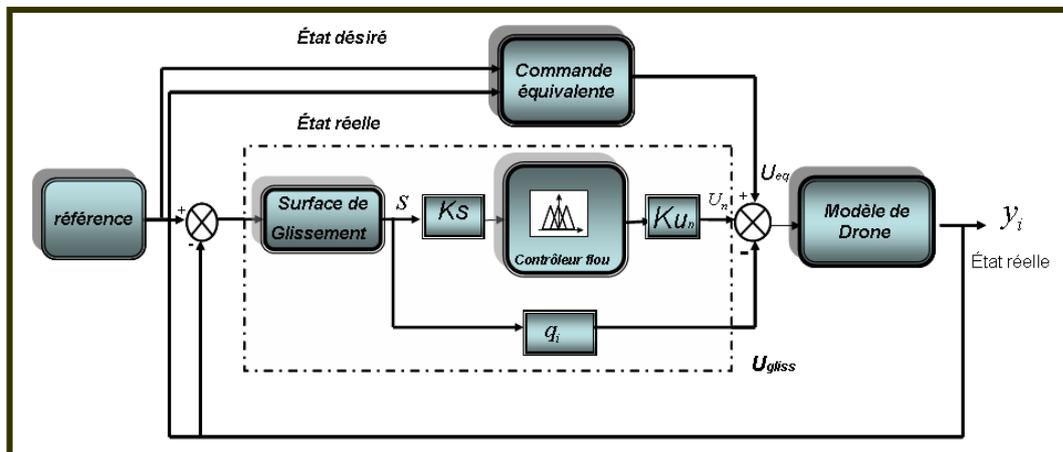


Figure III.23. Schéma bloc du contrôleur flou-glissant

III.4.5. Les résultats de simulation

En se référant aux résultats de simulation on remarque que :

Les figures III.24 et III.26, montrent une poursuite de trajectoire satisfaisante avec des erreurs de très faibles valeurs donc le drone tend vers les positions désirées.

La figure III.25, qui représente les signaux de commandes $(u_1 \ u_2 \ u_3 \ u_4)$ démontre que les commandes u_4, u_3, u_2 tendent vers zéros à partir de $t=30\text{sec}$ et $u_1 \approx mg = 4.7657$ après avoir effectué un manœuvre. Nous remarquons aussi que l'effet du phénomène de réticence est exclu et les commandes sont devenues lisses.

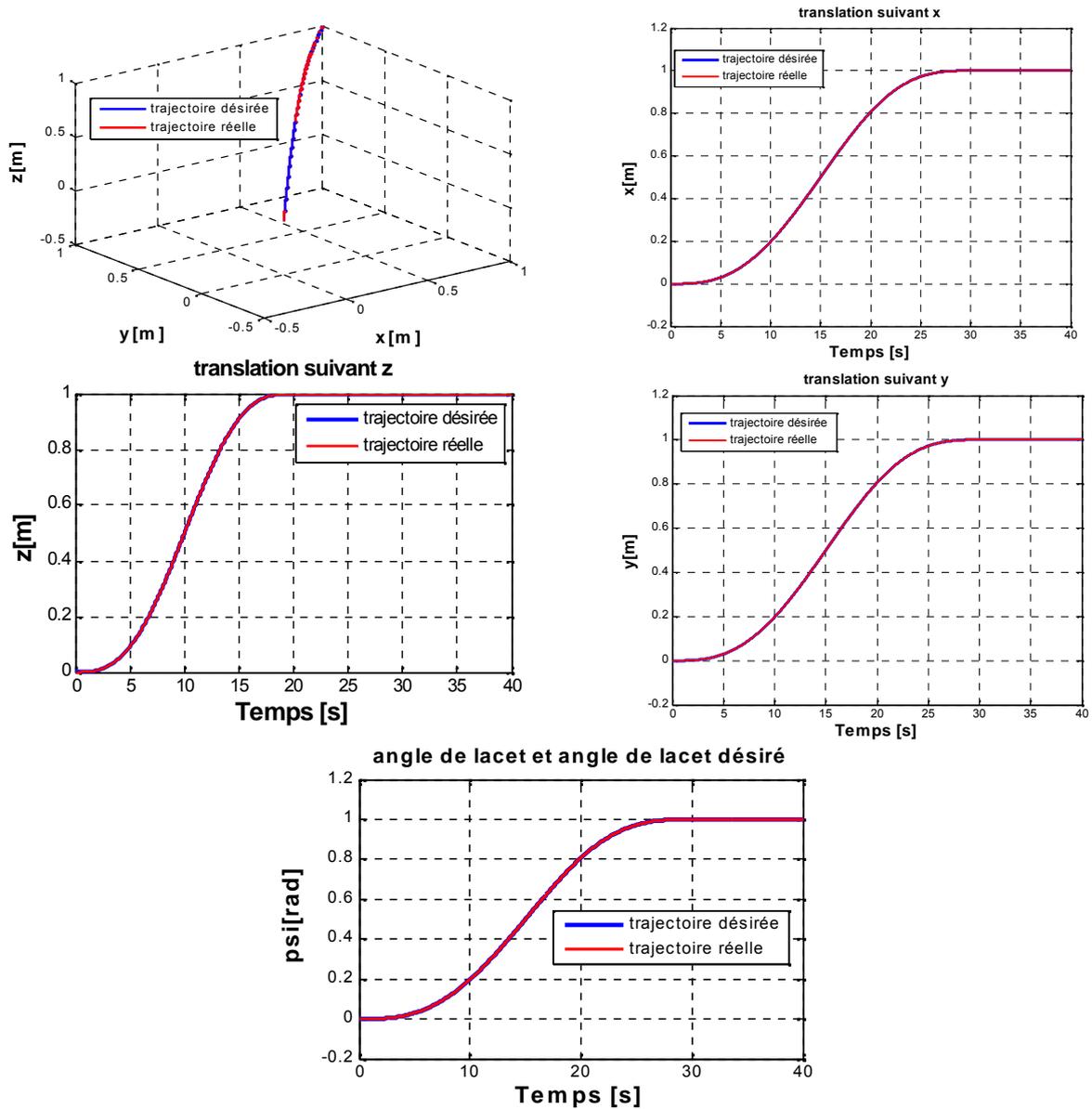
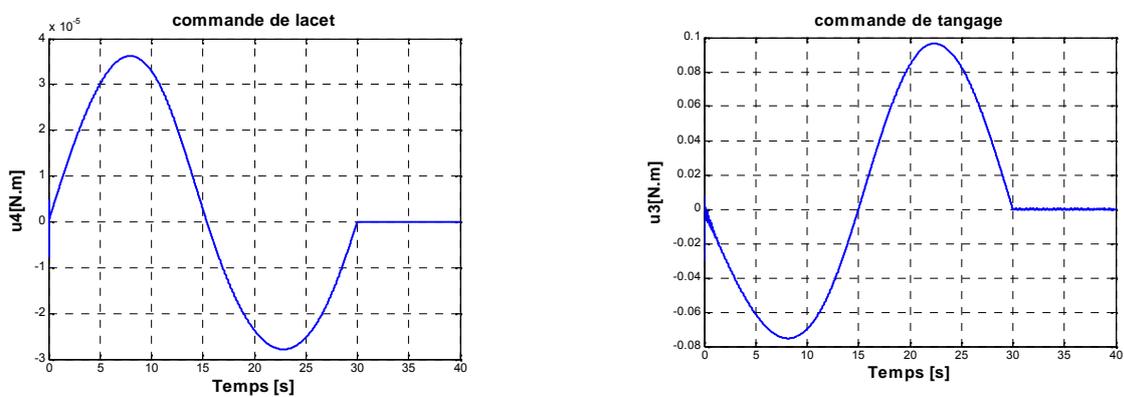


Figure III.24. Résultats de poursuite des trajectoires désirées selon (x, y, z, ψ) .



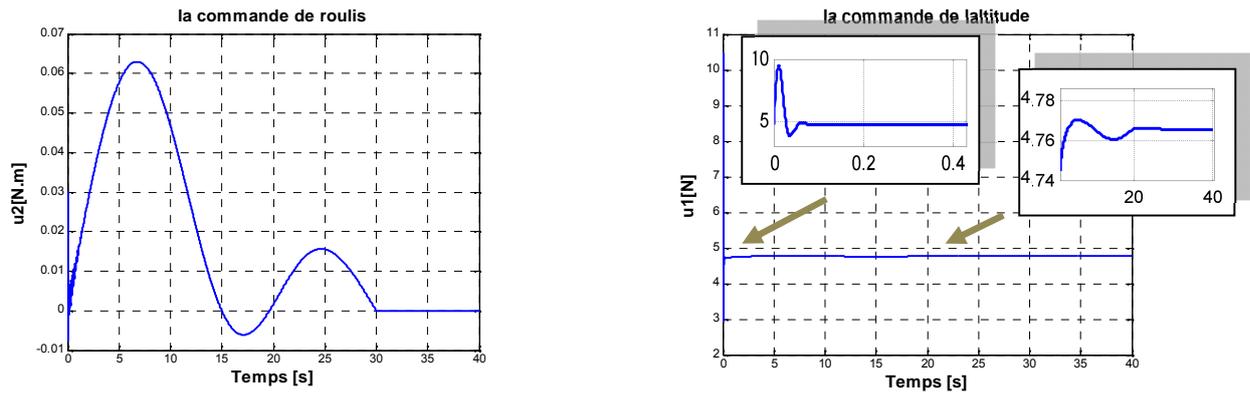


Figure III.25. Les commandes (u_1, u_2, u_3, u_4)

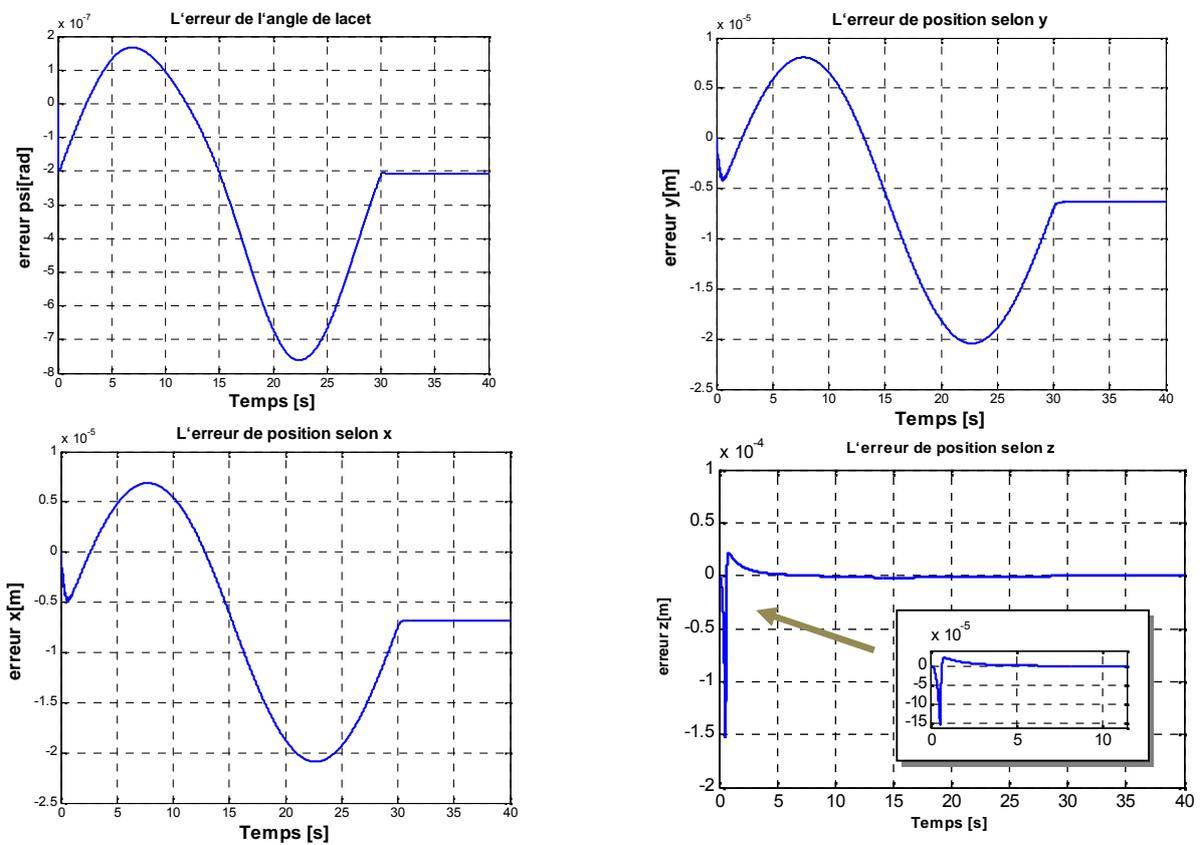


Figure III.26. Les erreurs de déplacements suivant (x, y, z) et l'angle ψ .

III.4.6. Etude de la robustesse

Si on applique une force perturbatrice $D_z = 7 \sin(0.1t)$ à l'instant $t=25s$ suivant (z) on obtient les figures suivantes:

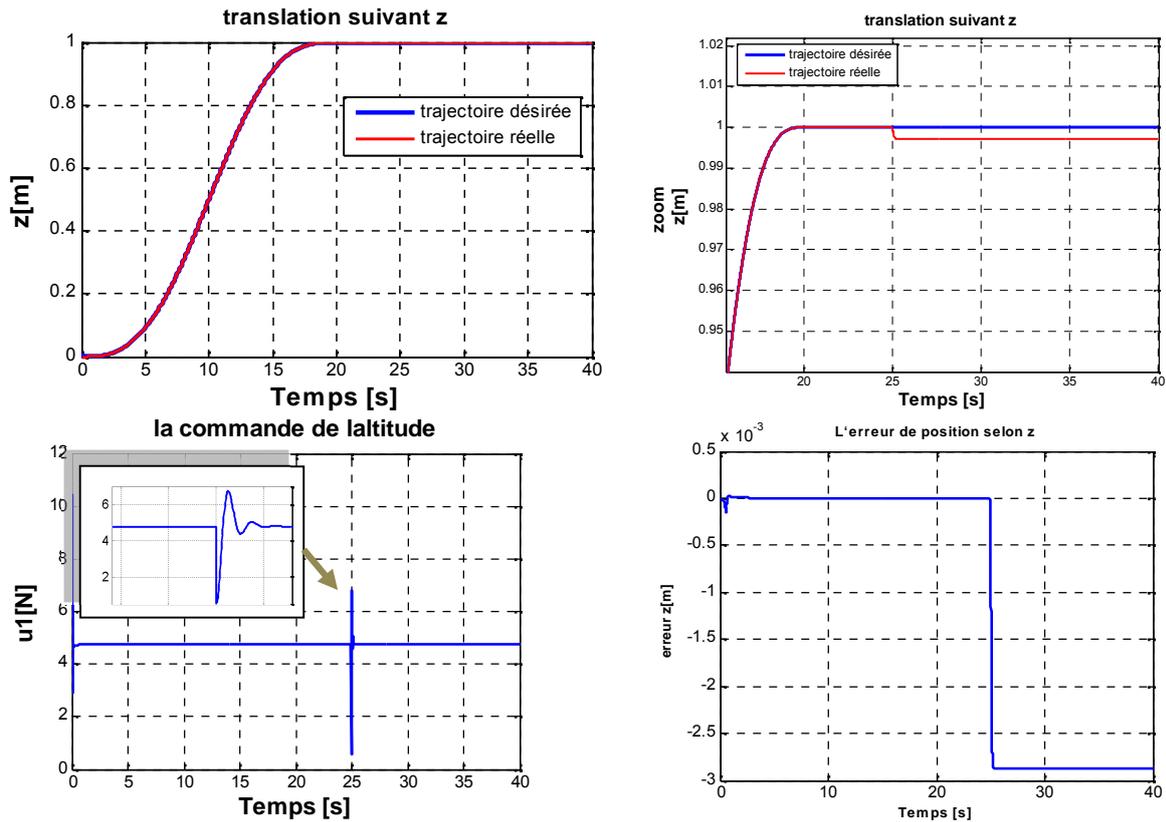


Figure III.27. L'influence de la perturbation selon (z)

D'après les résultats de simulation (figure III.27), nous remarquons que le contrôleur ne rejette pas la perturbation, le zoom sur la figure selon le vol vertical « z » montre très bien l'influence de la perturbation à $t=25$ sec car le contrôleur flous-glissant génère une erreur statique, pour surmonté ce problème on essaye de concevoir un contrôleur plus robuste vis-à-vis à la perturbation.

III.5. La commande par le PI mode flou glissant avec optimisation par l'algorithme génétique « PIFSMC-GA »

Dans le but d'améliorer la robustesse de la commande flous-glissant déterminée au paragraphe précédent, un contrôleur PI basé sur la théorie des modes de glissement est présenté ou nous allons introduire un algorithme d'optimisation pour trouvés les gains idéaux en prenant en considération toutes les non linéarités de notre système [Bel.12.a]. En effet, pour mémoire, il est connu que le point faible de la commande que nous avons élaborée est lié à la robustesse.

Cette section de chapitre est consacrée à la présentation des mécanismes qui constituent l'algorithme. Ces mécanismes s'inspirent directement des mécanismes de l'évolution naturelle. nous allons utiliser ce type d'algorithme d'optimisation pour la détermination

d'une structure optimale de contrôleurs PI utilisés. L'hybridation de l'AG avec les différentes structures de commande déjà étudiées donne naissance au contrôleur PIFSMC-GA.

III.5.1. les algorithmes génétiques

Les algorithmes génétiques (AG) sont des algorithmes d'optimisation stochastiques fondés sur les mécanismes de la sélection naturelle et de la génétique. Leur fonctionnement est extrêmement simple. On peut avec une population de solutions potentielles (chromosomes) initiales arbitrairement choisies évaluer leur performance (fitness) relative. Sur la base de ces performances, on crée une nouvelle population de solutions potentielles en utilisant des opérateurs évolutionnaires simples : la sélection, le croisement et la mutation. On recommence ce cycle jusqu'à ce que l'on trouve une solution satisfaisante [Cor.95].

III.5.2. Notions fondamentales sur les algorithmes génétiques

Un algorithme génétique est défini par :

1) Individu (chromosome / séquence)

Nous appelons une séquence (chromosome, individu) A de longueur $l(A)$ une suite $A = \{a_1, a_2, \dots, a_l\}$ avec $\forall i \in [1, l]$; $a_i \in V = \{0, 1\}$. Un chromosome est donc une suite de bits en codage binaire, appelé aussi chaîne binaire, tel que le codage réel, la suite A ne contient qu'un point, nous avons $A = \{a\}$ avec $a \in \mathbb{R}$.

2) Population

C'est un ensemble de chromosomes ou de points de l'espace de recherche.

3) Environnement

Un environnement est défini comme étant l'espace de recherche.

4) Fonction de fitness

Nous appelons fitness d'une séquence toute valeur positive que nous noterons $f(A)$, où f est typiquement appelée fonction de fitness. La fitness (efficacité) est donc donnée par une fonction à valeurs positives réelles. Dans le cas d'un codage binaire, nous utiliserons souvent une fonction de décodage qui permettra de passer d'une chaîne binaire à un chiffre à valeur réelle : $d : \{0, 1\}^l \rightarrow \mathbb{R}$ (où l est la longueur de la chaîne). La fonction de fitness est alors choisie telle qu'elle transforme cette valeur en valeur positive. Soit $f : d(\{0, 1\}^l) \rightarrow \mathbb{R}_+^*$. Le but

d'un algorithme génétique est alors simplement de trouver la chaîne qui maximise cette fonction f [gol.94].

III.5.3.Principaux étapes des algorithmes génétiques

Les algorithmes génétiques sont alors basés sur les phases suivantes :

- **Initialisation** : Une population initiale de N chromosomes est tirée aléatoirement.
- **Evaluation** : Chaque chromosome est décodé.
- **Sélection** : Création d'une nouvelle population N chromosomes par l'utilisation d'une méthode de sélection appropriée.
- **Reproduction** : Possibilité de croisement et mutation au sein de la nouvelle population.

Retour à la phase d'évaluation jusqu'à l'arrêt de l'algorithme.

III.5.4.Codage et population initiale

Il existe trois principaux types de codage : binaire, gray ou réel. Nous pouvons facilement passer d'un codage à l'autre. Rappelons que la transformation la plus simple (fonction de décodage d) d'une chaîne binaire A en nombre entier n s'opère par la règle suivante:

$$x = d(A) = \sum_{i=1}^l a_i 2^{l-i-1}. \quad (\text{III.35})$$

Ainsi le chromosome $A = \{1, 0, 1, 1\}$ vaut :

$$1 \times 2^3 + 0 \cdot 2^2 + 1 \cdot 2^1 + 1 \cdot 2^0 = 8 + 2 + 1 = 11$$

Evidemment la fonction d sera modifiée selon le problème. Ainsi nous cherchons à maximiser une fonction $f : [0,1] \rightarrow [0,1]$ une méthode possible sera la suivante (la taille du chromosome dépendant bien évidemment de la précision voulue).

$$x = d(A) = \sum_{i=1}^l a_i 2^{l-i-1}.$$

III.5.5.Opérateurs génétiques :

Les opérateurs génétiques les plus fréquemment utilisés sont le croisement et la mutation.

a) Le croisement :

L'opérateur de croisement permet la création de nouveaux individus selon un processus très simple. Il permet l'échange des informations entre les chromosomes (individus). Deux individus formant un couple sont tirés au sein de la nouvelle population issue de la reproduction. Puis un site de croisement est tiré aléatoirement (chiffre entre 1 et $l-1$). Enfin le

croisement s'effectue, les segments finaux des deux parents sont alors échangés autour de ce site. Cet opérateur permet la création de deux nouveaux individus. Un individu sélectionné lors de la reproduction ne subit pas nécessairement l'action d'un croisement. Ce dernier ne s'effectue qu'avec une certaine probabilité (P_c). Plus cette probabilité est élevée et plus la population subira de changements. Il se peut que l'action conjointe de la reproduction et du croisement soit insuffisante pour assurer la réussite de l'algorithme génétique. Ainsi le cas du codage binaire, certaines informations peuvent disparaître de la population. Ainsi aucun individu de la population initiale ne contient de un en dernière position de la chaîne et que ce un fasse partie de la chaîne optimale à trouver, tous les croisements possibles ne permettrons pas de faire apparaître ce un initialement inconnu. En codage réel, une telle situation peut arriver si la population initiale ne contient pas la valeur optimale. Pour remédier à ce problème l'opérateur de mutation est utilisé.

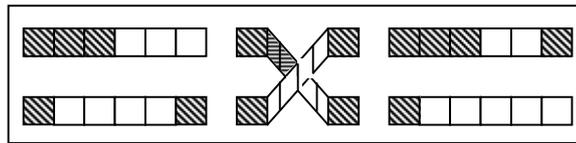


Figure III.28. Croisement en codage binaire

b) La mutation

Le rôle de cette opération est de modifier aléatoirement, avec une certaine probabilité, la valeur d'un composant de l'individu. Dans le cas du codage binaire, chaque bit $a_i \in \{0,1\}$ est remplacé selon une probabilité P_m par son inverse $a'_i = 1 - a_i$. C'est ce qu'illustre la figure suivante. Comme plusieurs lieux de croisement peuvent être possibles, nous pouvons très bien admettre qu'une même chaîne puisse subir plusieurs mutations.

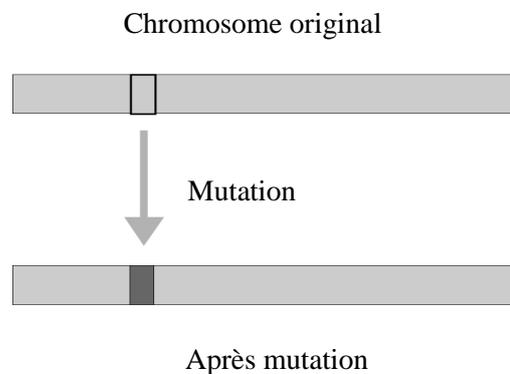


Figure III.29. Opérateur de mutation

III.5.6. Différentes étapes de mise en œuvre

a) Croisement

S'applique sur deux individus différents. Son résultat est un chromosome formé à partir des gènes de ses deux parents. Deux enfants sont 'produits' pour la génération suivante. Et Un pourcentage de croisement est fixé.

Exemple de croisement d'un point

	x	$y \downarrow$	z	t	u
Parent 1	$\underbrace{1110}_{14}$	$\underbrace{1100}_{12}$	$\underbrace{0010}_2$	$\underbrace{0111}_7$	$\underbrace{1001}_9$
Parent 2	$\underbrace{0101}_5$	$\underbrace{1111}_{15}$	$\underbrace{1011}_{11}$	$\underbrace{0000}_0$	$\underbrace{1010}_{10}$
Enfant 1	$\underbrace{1110}_{14}$	$\underbrace{1101}_{13}$	$\underbrace{1011}_{11}$	$\underbrace{0000}_0$	$\underbrace{1010}_{10}$
Enfant 2	$\underbrace{0101}_5$	$\underbrace{1110}_{14}$	$\underbrace{0010}_2$	$\underbrace{0111}_7$	$\underbrace{1001}_9$

b) Mutation

S'applique sur un seul individu pour la modification d'un ou plusieurs gènes du parent choisi (s) aléatoirement. Un seul nouvel enfant est fourni. Un pourcentage de mutation est fixé.

Exemple de mutation un point

	x	$y \downarrow$	z	t	u
Parent	$\underbrace{1110}_{14}$	$\underbrace{1100}_{12}$	$\underbrace{0010}_2$	$\underbrace{0111}_7$	$\underbrace{1001}_9$
Enfant	$\underbrace{1110}_{14}$	$\underbrace{1100}_{12}$	$\underbrace{0010}_2$	$\underbrace{0111}_7$	$\underbrace{1001}_9$

c) Sélection

Elle permet de choisir les individus sur lesquels s'appliqueront les opérations de reproduction pour la création de la future génération (création du mating pool). Elle doit favoriser les meilleurs individus. Elle doit permettre d'explorer les différentes parties de l'ensemble de recherche.

III.5.7.Fonctionnement de l'AG

Les algorithmes génétiques travaillent sur une population composée d'individus, tous différents, qui sont des solutions potentielles du problème à résoudre. Dans un premier temps, chaque individu est évalué ce qui permet de juger de la pertinence des solutions par rapport au problème considéré et conduit à éliminer les solutions jugées inutiles ou très mauvaises d'où mise à l'écart des individus les plus faibles pour favoriser les plus performants.

Les gènes des solutions sélectionnées sont combinés pour obtenir une nouvelle population qui doit être mieux adaptée au problème que la population précédente. La nouvelle population est alors soumise à des mutations et reproductions par hybridation. L'encouragement des éléments les plus aptes a pour résultat que les générations successives sont de plus en plus adaptées à la résolution du problème. Le processus est réitéré jusqu'à l'obtention d'une solution jugée satisfaisante [gol.94].

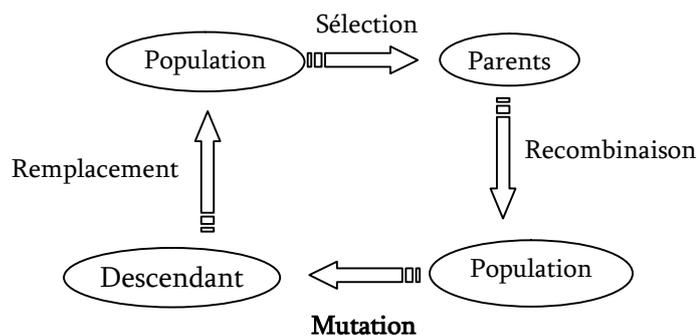


Figure III.30. Cycle d'un algorithme génétique

Il faut combiner les meilleurs individus de la population actuelle identifiée et sélectionnée. Ce phénomène s'appelle le crossover et permet d'exploiter l'ensemble des solutions possibles. La mutation est une modification qui intervient de manière aléatoire sur le génome d'un individu, elle a pour rôle de maintenir une certaine diversité dans la population. Elle n'intervient que sur une partie suffisamment petite pour ne pas détruire les caractéristiques sélectionnées mais suffisent généralement pour apporter des éléments nouveaux à un individu. L'algorithme génétique utilisant ce type de croisement est appelé algorithme génétique simple (AGS), cet algorithme sera utilisé dans la suite de notre travail.

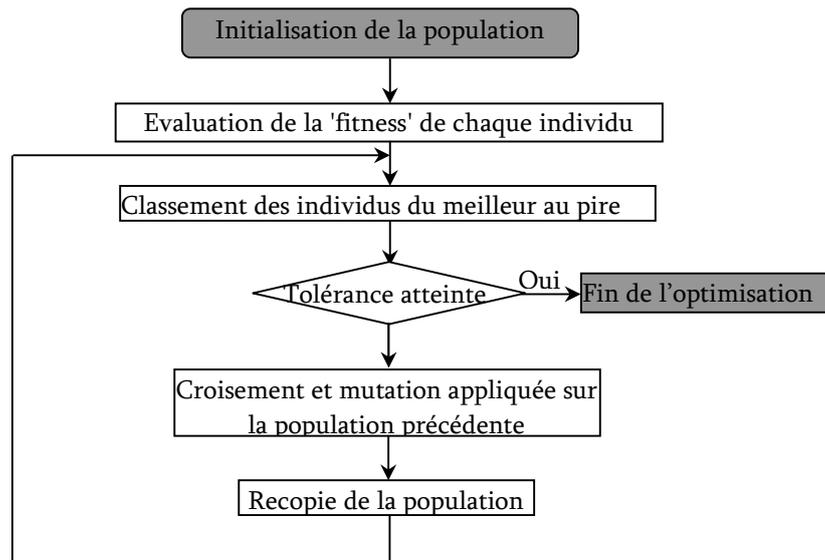


Figure III.31 : Organigramme de l'algorithme génétique.

III.5.8. Synthèse de la commande par PI flou glissant avec optimisation par l'AG

La commande développée dans cette section se compose de deux termes: [Far.06]

- un premier terme correspondant à la commande équivalente.
- un deuxième terme correspondant à un PI synthétisé par la méthode du mode glissant, avec des gains optimaux en prenant en considération toutes les non linéarités et les caractéristiques du processus supplémentaire. L'algorithme génétique peut être appliqué pour réaliser le réglage optimal.

c'est à dire:

$$u = u_{eq} + u_{sm} \quad (\text{III.36})$$

Avec : $u_{sm} = u_n + u_{PI}$

u_n correspond à la commande flou du terme $\underbrace{k \operatorname{sgn}(s)}_{\text{fuzzy}}$, u_{PI} désigne le contrôleur PI. On a:

$$u_{PI} = - \left[K_p s + K_I \int s \, d\tau \right] \quad (\text{III.37})$$

où K_p et K_I désignent comme à l'accoutumée et respectivement le gain d'intégration et le gain proportionnel.

La surface de glissement (s) est en général un hyperplan dans l'espace d'état global, et représente le comportement dynamique désiré donnée par l'expression (III.3).

Pour analyser la stabilité de ce contrôleur, nous choisissons une fonction de Lyapunov candidate définie positive basée sur la dynamique de l'erreur du système, c'est à dire:

$$V(s) = \frac{1}{2} s^2 \quad (\text{III.38})$$

sa dérivée par rapport au temps est:

$$\dot{V} = s \dot{s} \quad (\text{III.39})$$

L'objectif du contrôleur est de réaliser la condition de glissement. Ceci est exprimé par l'équation suivante:

$$s \dot{s} < 0 \quad (\text{III.40})$$

Supposons que, pour le gain proportionnel donné K_p , il existe respectivement un gain intégral K_I tels que la condition soit satisfaite. Par conséquent, la loi de commande u_{PI} est choisie de telle sorte à ce que la trajectoire soit dirigée vers la surface de glissement.

III.5.9. Démonstration : soit le sous système suivant :

$$\begin{cases} \dot{x}_1 = x_2 \\ \dot{x}_2 = \frac{(I_y - I_z)}{I_x} x_4 x_6 - \frac{J_r}{I_x} \Omega_r x_4 - \frac{K_{fax}}{I_x} x_2^2 + \frac{1}{I_x} u_2 \end{cases} \quad (\text{III.41})$$

Le degré relatif $r_2 = 2$, pour cela nous choisissons la surface de glissement comme suit :

$$s_2 = \dot{e}_2 + \lambda_2 e_2 \quad (\text{III.42})$$

Avec $\lambda_2 > 0$, $e_2 = \phi - \phi_d = x_1 - x_{1d}$, l'erreur entre l'angle de roulis ϕ et celle désirée et \dot{e}_2 sa dérivée, en utilisant la théorie de la commande par mode de glissement on trouve :

$$u_2 = u_{2eq} + u_{2sm} \quad (\text{III.43})$$

On a la dérivée temporelle de s_2 , tel que:

$$\begin{cases} \dot{s}_2 = -\underbrace{k_2 \operatorname{sgn}(s_2)}_{\text{fuzzy}} - \left[K_{2p} s_2 + K_{2I} \int s_2 d\tau \right] \\ \dot{s}_2 = \ddot{e}_2 + \lambda_2 \dot{e}_2 \end{cases} \quad \text{Avec : } k_2, K_{2p}, K_{2i} > 0 \quad (\text{III.44})$$

Il en résulte :

$$\begin{aligned}\dot{s}_2 &= -\underbrace{k_2 \operatorname{sgn}(s_2)}_{fuzzy} - \left[K_{2p}s_2 + K_{2I} \int s_2 d\tau \right] \\ &= \ddot{e}_2 + \lambda_2 \dot{e}_2 = \dot{x}_2 - \ddot{x}_{1d} + \lambda_2 (x_2 - \dot{x}_{1d})\end{aligned}\quad (\text{III.45})$$

Alors :

$$u_2 = I_x \left(\begin{array}{l} \ddot{x}_{1d} - \left(\frac{I_y - I_z}{I_x} \right) x_4 x_6 + \frac{J_r}{I_x} \Omega_r x_4 + \frac{K_{fax}}{I_x} x_2^2 - \lambda_2 (x_2 - \dot{x}_{1d}) \\ -\underbrace{k_2 \operatorname{sgn}(s_2)}_{fuzzy} - \left[K_{2p}s_2 + K_{2I} \int s_2 d\tau \right] \end{array} \right) \quad (\text{III.46})$$

Par conséquent :

$$\left\{ \begin{array}{l} u_{2equ} = I_x \left(\ddot{x}_{1d} - \left(\frac{I_y - I_z}{I_x} \right) x_4 x_6 + \frac{J_r}{I_x} \Omega_r x_4 + \frac{K_{fax}}{I_x} x_2^2 - \lambda_2 (x_2 - \dot{x}_{1d}) \right) \\ u_{2sm} = I_x \left(-\underbrace{k_2 \operatorname{sgn}(s_2)}_{fuzzy} - \left[K_{2p}s_2 + K_{2I} \int s_2 d\tau \right] \right) \end{array} \right. \quad (\text{III.47})$$

Les mêmes étapes sont reprises pour les autres commandes :

$$\begin{aligned}u_1 &= \frac{m}{(\cos(\phi)\cos(\theta))} \left(\ddot{x}_{11d} + \frac{K_{fz}}{m} x_{12} + g - \lambda_1 (x_{12} - \dot{x}_{11d}) - k_1 \operatorname{sgn}(s_1) - \left[K_{1p}s_1 + K_{1I} \int s_1 d\tau \right] \right) \\ u_3 &= I_y \left(\ddot{x}_{3d} - \left(\frac{I_z - I_x}{I_y} \right) x_2 x_6 - \frac{J_r}{I_y} \Omega_r x_2 + \frac{K_{fay}}{I_y} x_4^2 - \lambda_3 (x_4 - \dot{x}_{3d}) - k_3 \operatorname{sgn}(s_3) - \left[K_{3p}s_3 + K_{3I} \int s_3 d\tau \right] \right) \\ u_4 &= I_z \left(\ddot{x}_{5d} - \left(\frac{I_x - I_y}{I_z} \right) x_2 x_4 + \frac{K_{faz}}{I_z} x_6^2 - \lambda_4 (x_6 - \dot{x}_{5d}) - k_4 \operatorname{sgn}(s_4) - \left[K_{4p}s_4 + K_{4I} \int s_4 d\tau \right] \right) \\ u_x &= \frac{m}{u_1} \left(\ddot{x}_{7d} + \frac{K_{fx}}{m} x_8 - \lambda_x (x_8 - \dot{x}_{7d}) - k_x \operatorname{sgn}(s_x) - \left[K_{xp}s_x + K_{xI} \int s_x d\tau \right] \right) \\ u_y &= \frac{m}{u_1} \left(\ddot{x}_{9d} + \frac{K_{fy}}{m} x_{10} - \lambda_y (x_{10} - \dot{x}_{9d}) - k_y \operatorname{sgn}(s_y) - \left[K_{yp}s_y + K_{yI} \int s_y d\tau \right] \right) \text{ avec: } (k_i, K_{ip}, K_{iI}) > 0\end{aligned}\quad (\text{III.48})$$

Remarque 1 : On remplace le terme $k_i \operatorname{sgn}(s_i)$ par un contrôleur flou.

Le schéma de principe de l'optimisation du contrôleur PIFSMC-GA est illustré dans la figure III.32

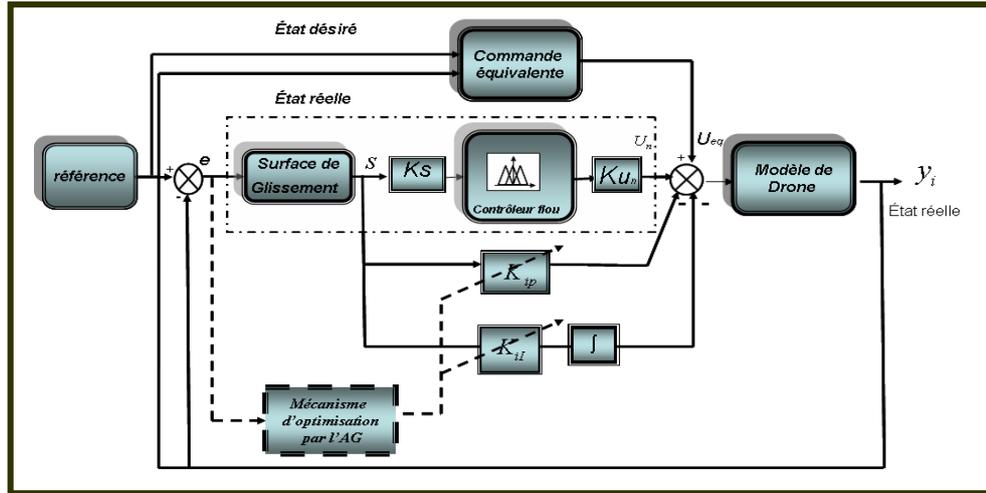


Figure III.32. Schéma bloc du contrôleur par PI flou glissant avec optimisation par l'AG « PIFSMC-GA ».

La fonction objective utilisée dans l'optimisation des deux gains K_{ip} , K_{ii} est donnée par l'expression suivante :

$$\text{Fitness} = \int_0^t (e_T(t)) dt \quad (\text{III.49})$$

$$\text{Avec : } e_T = |e_1| + |e_2| + |e_3| + |e_4| + |e_x| + |e_y|$$

Dont les erreurs de poursuite sont écrites sous la forme ci-dessous :

$$\begin{aligned} e_1 &= z - z_d = x_{11} - x_{11d} \\ e_2 &= \phi - \phi_d = x_1 - x_{1d} \\ e_3 &= \theta - \theta_d = x_3 - x_{3d} \\ e_4 &= \psi - \psi_d = x_5 - x_{5d} \\ e_x &= x - x_d = x_7 - x_{7d} \\ e_y &= y - y_d = x_9 - x_{9d} \end{aligned} \quad (\text{III.50})$$

Les paramètres de l'algorithme génétique choisis sont :

- population initiale = 100 ;
- probabilité de croisement = 0.9 ;
- probabilité de mutation = 0.1 ;
- nombre maximal de génération = 50 ;

III.5.10. Résultats d'optimisation

Avec les paramètres suscités de l'AG, l'optimisation du contrôleur PI par l'AG nous a permis d'avoir les valeurs optimaux des K_{ip} , K_{il} , ces valeurs sont présentées dans le tableau III.1 :

Tableau III.1 : paramètres du contrôleur PI optimisés par l'AG.

K_{ip}					
K_{1p}	K_{2p}	K_{3p}	K_{4p}	K_{xp}	K_{yp}
15.2877	15.2118	13.0688	10.3426	15.0194	12.5349
K_{il}					
K_{1l}	K_{2l}	K_{3l}	K_{4l}	K_{xl}	K_{yl}
12.8339	8.7609	10.85	5.35	19.5784	11.6

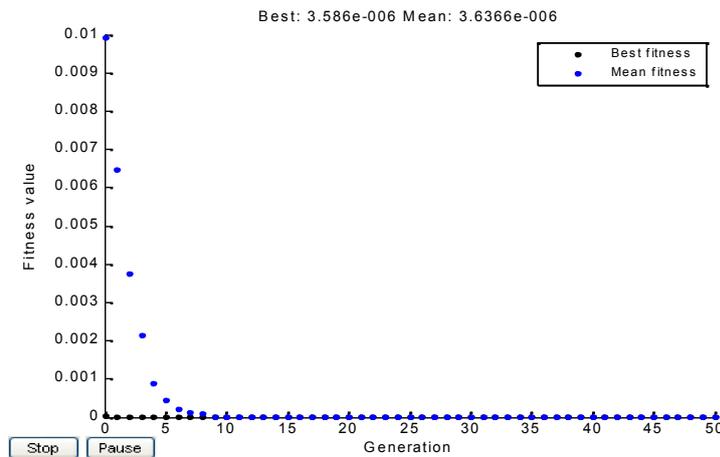


Figure III.33. Représentation de la fonction objective.

III.5.11. Résultats de simulation

Pour illustrer les performances du réglage avec le PIFSMC optimisé par l'AG, nous avons effectué les tests suivants :

Avec : $x_1=25$; $x_2=25$; $x_3=25$; $x_4=25$; $x_x=2$; $x_y=2$;

et $k_1=0.01$; $k_2=0.95$; $k_3=0.01$; $k_4=0.01$; $k_x=0.01$; $k_y=0.01$;

Les figures III.34, III.35, III.36, représentent respectivement la réalisation en 3D selon (x, y, z) de trajectoire de type cycloïdale et l'angle ψ , la commande et les erreurs de

déplacements. Nous remarquons qu'à l'équilibre u_1 est approximativement égale à $mg = 4.7657$ avec un bon suivi de consigne. Les erreurs tendent vers zéro donc le drone tend vers les positions désirées. Dans la section suivante nous allons tester la robustesse de notre contrôleur vis-à-vis à la perturbation et la variation paramétrique.

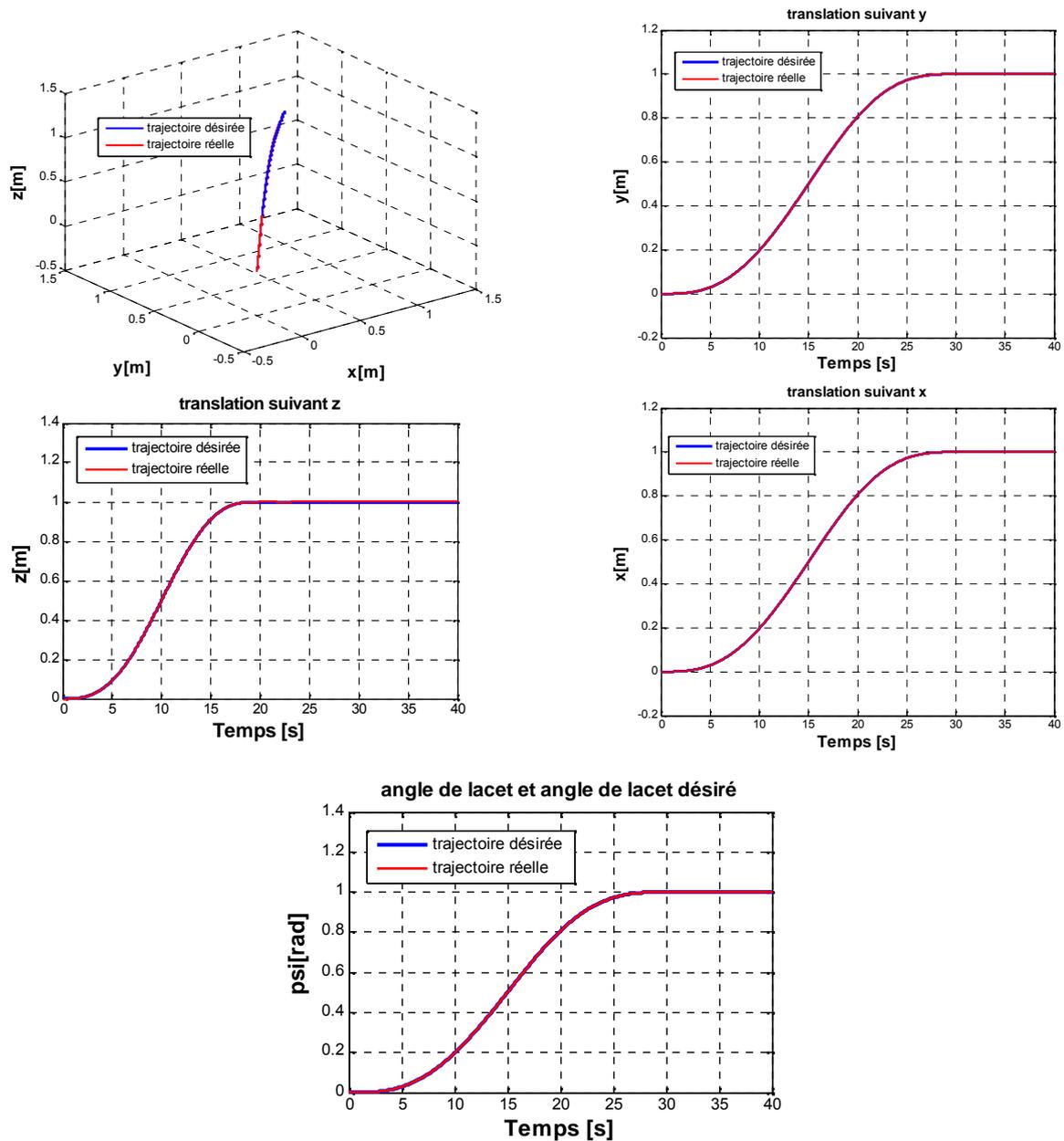


Figure III.34. Résultats de poursuite des trajectoires désirées selon (x, y, z, ψ) .

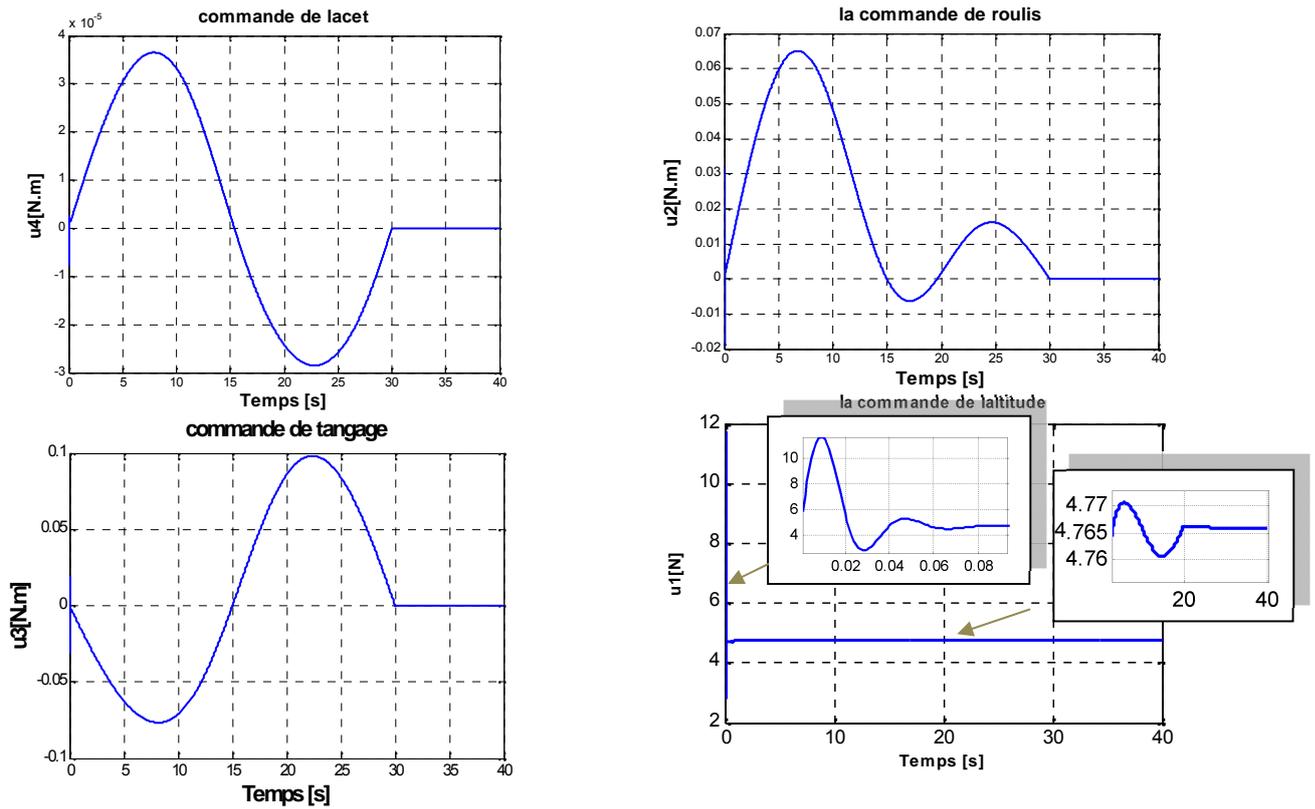


Figure III.35. Les commandes (u_1, u_2, u_3, u_4)

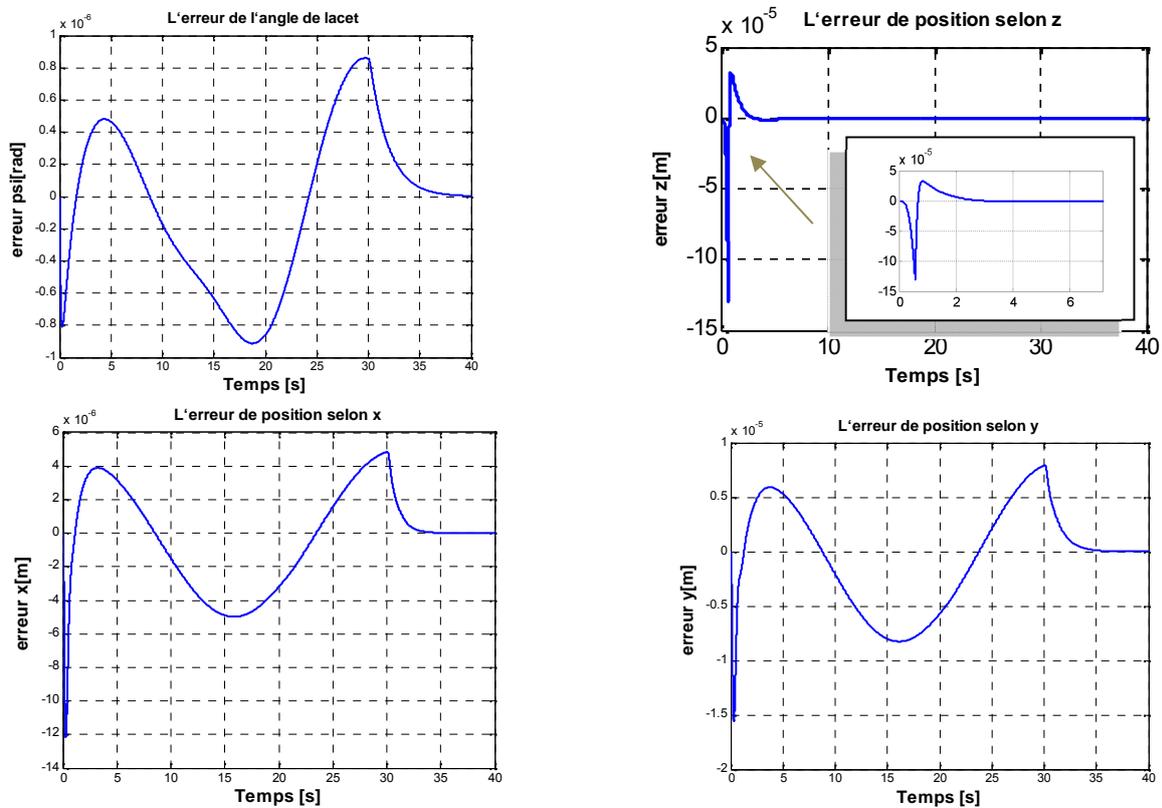


Figure III.36. Les erreurs de déplacements suivant (x, y, z) et l'angle ψ .

III.5.12. Etude de la robustesse

Pour illustrer les performances de notre nouvelle commande on effectue les testes suivants :

- On applique une force perturbatrice $D_z = 7 \sin(0.1t)$ à l'instant $t=25\text{sec}$ suivant (z).
- On applique des forces perturbatrices : $d_x = 2 \sin(0.1t)$; $d_y = 1.5 \sin(0.1t)$; $d_z = 2 \sin(0.1t)$ respectivement à l'instant $t=25\text{sec}$, $t=35\text{sec}$ et $t=20\text{sec}$ et des incertitudes paramétriques +20% de (I_x, I_y, I_z) .
- On compare entre les trois commandes suivant (z) avec : $D_z = 7 \sin(0.1t)$ à l'instant $t=25\text{sec}$ et des incertitudes paramétriques +20% de (I_x, I_y, I_z) .

Les résultats de simulations sont illustrés dans les figures suivantes:

a)

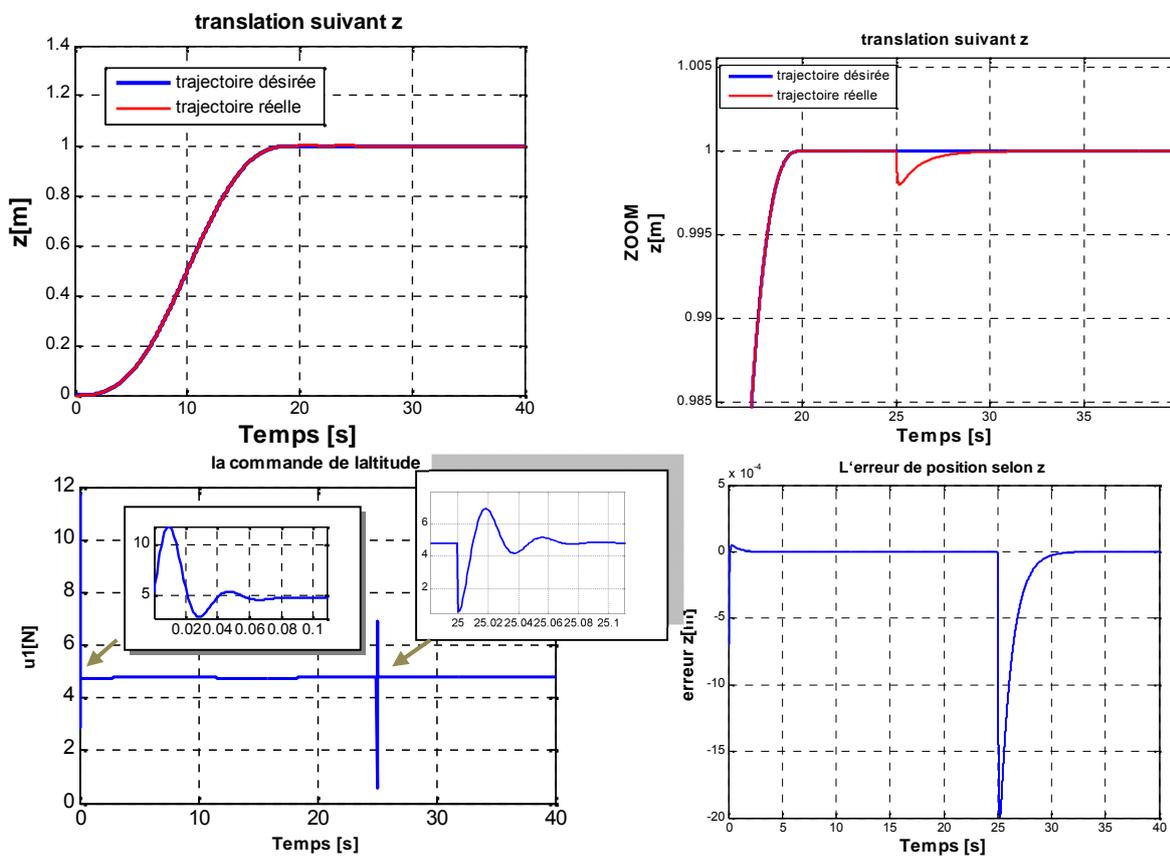


Figure III.37. L'influence de la perturbation selon (z) pour le contrôleur «PIFSMC-GA».

b)

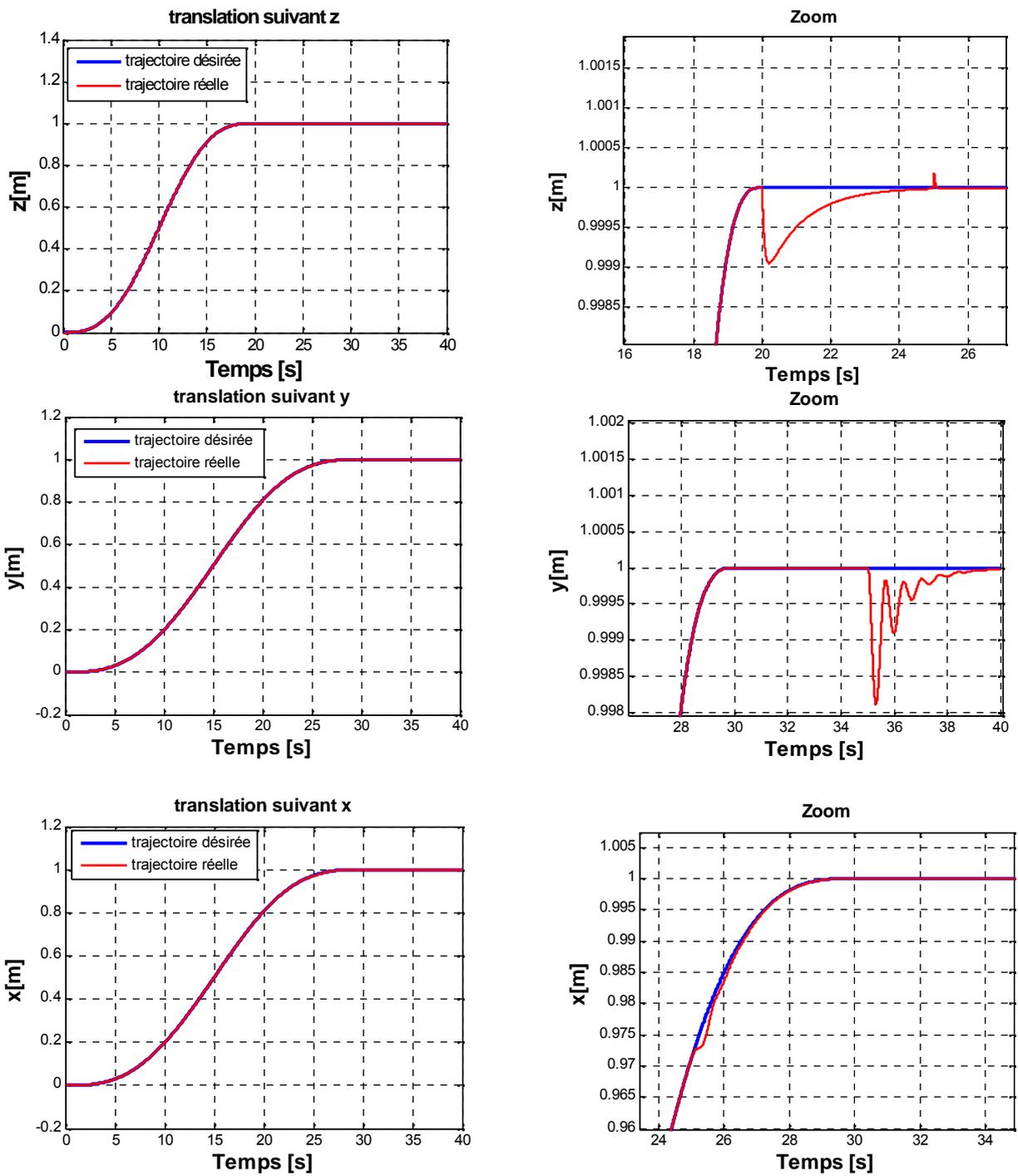


Figure III.38. L'influence de la perturbation selon (x, y, z) avec des incertitudes paramétriques +20% de (I_x, I_y, I_z) pour le contrôleur «PIFSMC-GA».

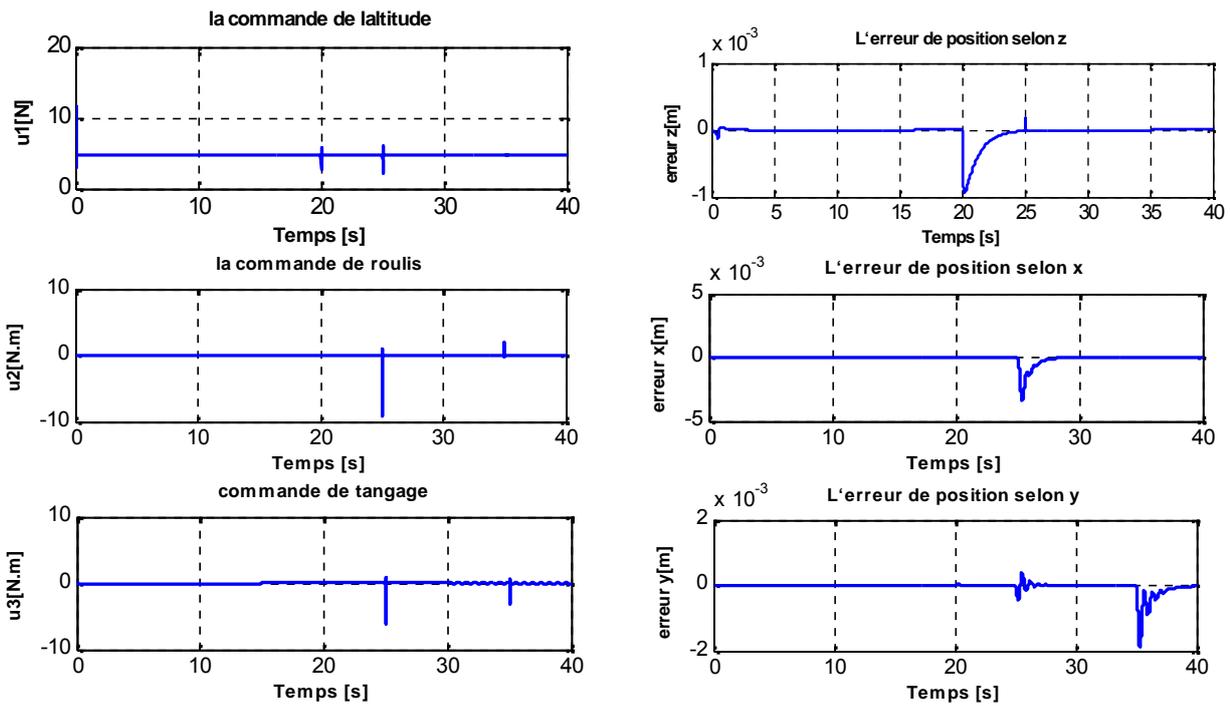
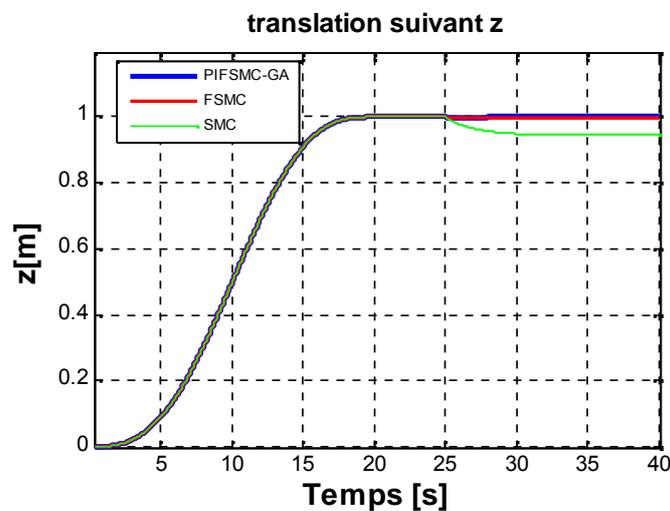


Figure III.39. Les signaux de commandes et des erreurs selon (x, y, z) avec des incertitudes paramétriques +20% de (I_x, I_y, I_z) pour le contrôleur «PIFSMC-GA»

D’après les figures III.37 III.38 III.39, nous remarquons que la commande PIFSMC-GA présente une robustesse vis-à-vis aux perturbations externes et aux incertitudes paramétriques cela montre clairement la performance de ce contrôleur et sa robustesse, nous constatons aussi que puisque notre système est un système non linéaire couplé, les commandes sont influencées par les perturbations externes selon les axes (x,y,z) .

c) comparaison des trois commandes suivant (z) :



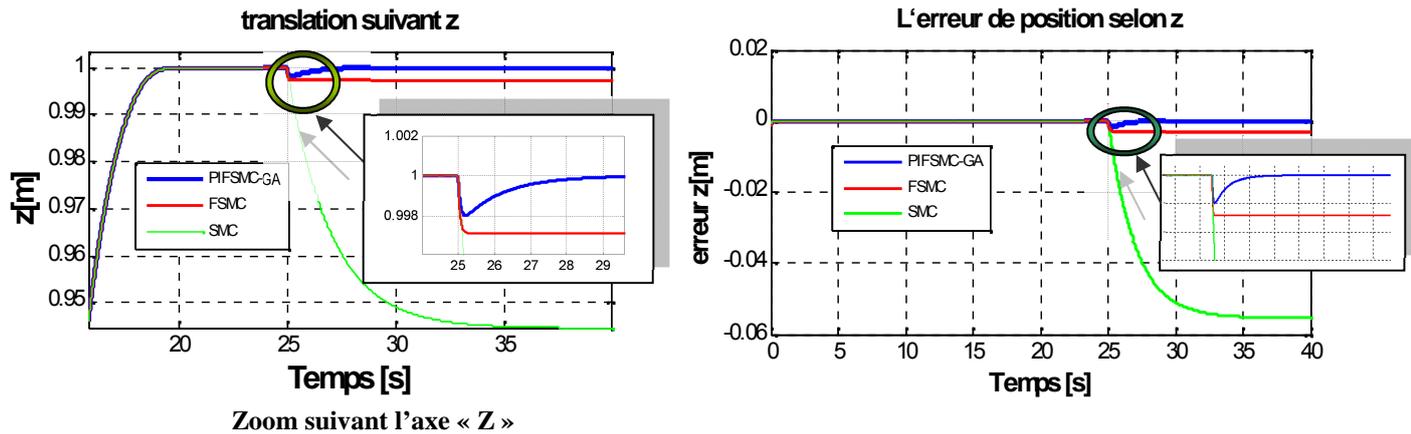


Figure III.40 : l'influence de la perturbation selon (z), du contrôleur « FSMC » et « SMC », « PIFSMC-GA » avec des incertitudes paramétriques +20% de (I_x, I_y, I_z) .

D'après les résultats de simulations (figure. III.40) nous remarquons clairement la différence entre les trois contrôleurs vis-à-vis au rejet de perturbation car le PIFSMC-GA a donné de meilleurs résultats avec une poursuite de trajectoire satisfaisante contrairement aux deux contrôleurs SMC et FSMC donc l'étude de la robustesse en stabilité et en performance montre que le correcteur établi par la PIFSMC-GA offre les meilleures qualités.

Dans cette partie de chapitre nous avons étudié la commande par mode de glissement. Ce choix a été justifié par la simplicité de la conception de cette commande. Pour améliorer davantage les performances de la commande SMC nous avons proposé des approches d'hybridation entre le mode glissant et la logique floue et le PI avec optimisation par l'algorithme génétique. En générale les résultats de simulations sont satisfaisants par rapport à la stabilité, mais au perturbation nous avons vu que PIFSMC-GA a donné de meilleurs résultats avec une très grande robustesse vis-à-vis des incertitudes paramétriques et des perturbations externes. Il faut cependant souligner que le fait d'introduire l'action intégrale et proportionnelle et la logique floue dans la commande par mode glissant nous a permis d'obtenir de meilleurs résultats.

III.6. la commande par backstepping

La technique de commande backstepping a été développée par « Kanellakopoulos » en 1991 et inspirée par les travaux de Feurer & Morse (1978) d'une part et Tsiniias (1989) et Kokotovic et Sussmann (1989) d'autre part. Cette technique offre une méthode systématique pour effectuer la conception d'un contrôleur pour les systèmes non linéaire, l'idée consiste à calculer une loi de commande afin de garantir pour une certaine fonction (*Lyapunov*) définie positive que sa dérivée soit toujours négative. L'objectif de cette technique est de calculer, en

plusieurs étapes, une commande qui garantit la stabilité globale du système [Ben.00]. Contrairement à la plupart des autres méthodes, le backstepping n'a aucune contrainte de non linéarité.

III.6.1. Définition

a) Point d'équilibre : physiquement, un système est en équilibre quand il conserve son état en absence de forces externes. Mathématiquement, cela équivaut à dire que la dérivée \dot{x} de son état est nulle.

$$\dot{x} = f(x) = 0 \quad (\text{III.51})$$

b) Stabilité : on dit qu'un système est stable lorsque il revient à son état d'équilibre si l'on excite par une impulsion, il est instable lorsque il s'éloigne.

c) Stabilité de Lyapunov : [Esh.05]

Considérons le système invariable suivant :

$$\dot{x} = f(x) \quad (\text{III.52})$$

commençons à l'état initial $x(0)$. Supposons que x_e est un point d'équilibre du système, $f(x_e) = 0$. Nous disons que le point d'équilibre est :

- Stable, si pour chaque $\varepsilon > 0$ existe $\delta(\varepsilon) > 0$, tel que :

$$\|x(0) - x_e\|^2 < \delta \Rightarrow \|x(t) - x_e\|^2 < \varepsilon, \text{ pour tout } t \geq 0$$

- Asymptotiquement stable, s'il est stable et en plus il existe $r > 0$ tels que

$$\|x(0) - x_e\|^2 < r \Rightarrow x(t) \rightarrow x_e, \text{ lorsque } t \rightarrow \infty$$

- Globalement asymptotiquement stable, s'il est asymptotiquement stable pour tous les états initiaux.

III.6.2. Méthodes d'analyse de la stabilité des systèmes

L'étude de la stabilité des systèmes non linéaire est très complexe. L'approche de Lyapunov est l'approche la plus utilisée pour étudier ce problème. Cette approche a été introduite au 19^{ième} siècle par le mathématicien russe *Alexandre Mikhailovich Lyapunov* dans son travail intitulé « *The général problem of the motion stability* ».

On distingue deux méthodes de Lyapunov pour l'analyse de la stabilité :

- La méthode de linéarisation,
- La méthode directe.

La première concerne la stabilité locale autour d'un point d'équilibre. Par contre, la deuxième détermine la stabilité des systèmes en construisant une fonction scalaire des états du système dont on examinera la variation temporelle.

On s'intéresse par la méthode directe de Lyapunov car le backstepping est basé sur cette méthode.

a) Méthode direct de Lyapunov

Son objectif, est de définir une méthode permettant d'analyser la stabilité d'un système linéaire sans connaître explicitement les solutions des équations différentielles qui le décrivent. La philosophie de cette méthode n'est que l'extension mathématique d'un phénomène physique observé, car les systèmes mécaniques et électriques perdent de l'énergie pour se stabiliser au point d'équilibre.

➤ Théorème de Lyapunov

La méthode directe de Lyapunov (ou la méthode des fonctions de Lyapunov) est dérivée du critère énergétique de stabilité en appliquant ce critère indépendamment du concept d'énergie, on remplace alors l'énergie du système par une « fonction de Lyapunov » qui est définie positive (comme l'énergie). Soit le système autonome :

$$\dot{x} = f(x) \quad , x_e = 0$$

Ce système aura un point d'équilibre $x_e = 0$, globalement asymptotiquement stable, s'il existe une fonction scalaire $V(x)$ continue avec une dérivée partielle par rapport au temps $\dot{V}(x)$ continue ayant les propriétés suivantes :

1. $V(0) = 0$,
2. $V(x) > 0, \quad \forall x \neq 0$,
3. $\lim_{\|x\| \rightarrow \infty} V(x) = \infty$ (radialement non bornée),
4. $\dot{V} < 0 \quad \forall x \neq 0$.

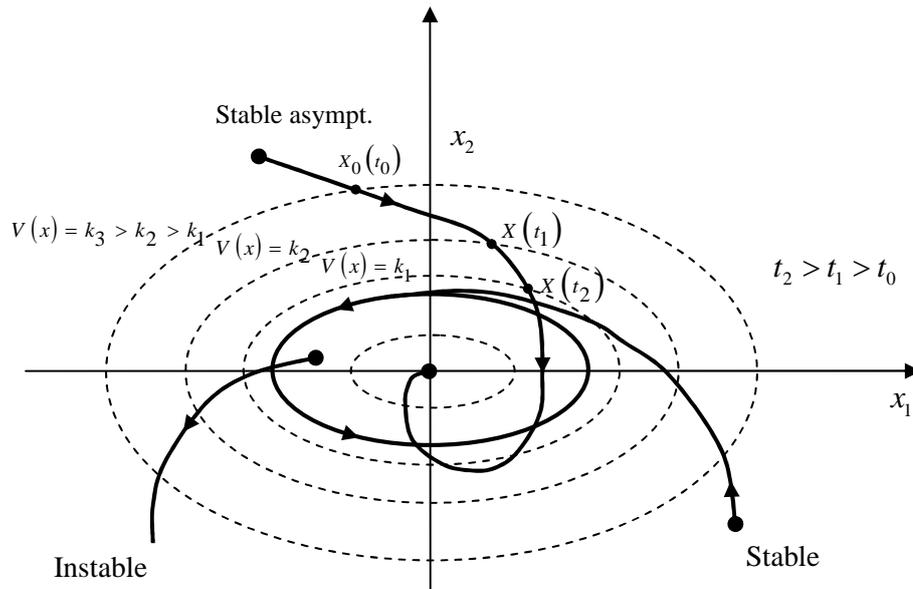


Figure III.41. Interprétation géométrique du théorème de Lyapunov.

b) La synthèse de la commande par la méthode directe de Lyapunov

Dans les paragraphes précédents, nous avons étudié la stabilité des systèmes où on a supposé implicitement que la loi de commande a été choisie et notre but était de vérifier la stabilité du système avec cette loi de commande, mais le problème dans cette synthèse est comment trouver cette commande qui stabilisera le système.

Nous allons présenter une méthodologie qui combine entre la recherche de la fonction de Lyapunov et la loi stabilisante. En général, il existe deux concepts pour l'application de la méthode directe de Lyapunov pour la synthèse d'une commande stable :

1^{ère} concept : On suppose que la loi de commande existe et on cherche la fonction de Lyapunov.

2^{ème} concept : Cette fois si, on fait un choix sur $V(x)$, la fonction de Lyapunov candidate, et on cherche la loi de commande qui rend cette fonction candidate la fonction de Lyapunov réelle.

Dans la plupart des théorèmes et lemmes de la stabilité au sens de Lyapunov, l'existence de la fonction de Lyapunov était assumée et l'objectif était de déterminer la stabilité de ces systèmes. Mais dans la plupart de cas, la recherche de cette fonction est très difficile. Dans le paragraphe suivant, nous présenterons la notion du « *Backstepping* » qui nous offre une solution à ce problème.

III.6.3. Technique de commande par le backstepping

a) Algorithme de base

Afin d'illustrer le principe de la méthode du backstepping, on considère le cas des systèmes non linéaires de la forme [Ben.00] :

$$\begin{aligned}\dot{x}_1 &= f_1^T(x_1)\mathcal{G} + g_1(x_1)x_2 \\ \dot{x}_2 &= f_2^T(x_1, x_2)\mathcal{G} + g_2(x_1, x_2)x_3 \\ \dot{x}_3 &= f_3^T(x_1, x_2, x_3)\mathcal{G} + g_3(x_1, x_2, x_3) \cdot u\end{aligned}\quad (\text{III.53})$$

Le vecteur des paramètres \mathcal{G} est supposé connu. On désire faire suivre à la sortie $y = x_1$ le signal de référence y_r , où y_r , \dot{y}_r , \ddot{y}_r et $y_r^{(3)}$ sont supposées connues et uniformément bornées.

Le système étant du troisième ordre, le design s'effectue en trois étapes.

Étape1 : On considère d'abord le premier sous système

$$\dot{x}_1 = f_1^T(x_1)\mathcal{G} + g_1(x_1)x_2 \quad (\text{III.54})$$

la variable d'état x_2 est traitée comme une commande et l'on définit la première valeur désirée

$$(x_1)_d \triangleq \alpha_0 = y_r \quad (\text{III.55})$$

la première variable d'erreur se définit par

$$z_1 = x_1 - \alpha_0 \quad (\text{III.56})$$

avec ces variables, le système (III.54) s'écrit :

$$\begin{aligned}\dot{z}_1 &= \dot{x}_1 - \dot{\alpha}_0 \\ &= f_1^T\mathcal{G} + g_1x_2 - \dot{\alpha}_0\end{aligned}\quad (\text{III.57})$$

pour un tel système, la fonction quadratique,

$$V_1(z_1) = \frac{1}{2}z_1^2 \quad (\text{III.58})$$

Sa dérivée le long de la solution de (III.57), est donnée par :

$$\begin{aligned}\dot{V}_1 &= z_1\dot{z}_1 \\ &= z_1 \cdot [f_1^T\mathcal{G} + g_1x_2 - \dot{\alpha}_0]\end{aligned}\quad (\text{III.59})$$

un choix judicieux de x_2 rendrait \dot{V}_1 négative et assurerait la stabilité de l'origine du sous système décrit par (III.54). Prenons comme valeur de x_2 , la fonction α_1 , telle que :

$$f_1 \alpha_1 + g_1^T \mathcal{G} - \dot{\alpha}_0 = -c_1 z_1 \quad (\text{III.60})$$

où $c_1 > 0$ est un paramètre de design. Cela donne

$$(x_2)_d \triangleq \alpha_1 = \frac{1}{g_1} [-c_1 z_1 - f_1^T \mathcal{G} + \dot{\alpha}_0] \quad (\text{III.61})$$

et la dérivée s'écrit

$$\dot{V}_1 = -c_1 z_1^2 \leq 0 \quad (\text{III.62})$$

d'où la stabilité asymptotique de l'origine de (III.54).

Etape2 : On considère, dans ce cas, le deuxième sous-système

$$\dot{x}_2 = f_2^T(x_1, x_2) \mathcal{G} + g_2(x_1, x_2) x_3 \quad (\text{III.63})$$

on définit la nouvelle variable d'erreur

$$z_2 = x_2 - \alpha_1 \quad (\text{III.64})$$

qui représente l'écart entre la variable d'état x_2 et sa valeur désirée α_1 . A cause du fait que x_2 ne peut être forcée à prendre instantanément une valeur désirée, en l'occurrence α_1 , l'erreur z_2 n'est pas, instantanément, nulle. Le design dans cette étape consiste, alors, à la forcer à s'annuler avec une certaine dynamique, choisie au préalable.

Les équations du système à commander, dans l'espace (z_1, z_2) , s'écrivent

$$\begin{aligned} \dot{z}_1 &= f_1^T \cdot \mathcal{G} - \dot{\alpha}_0 + g_1 \cdot (z_2 + \alpha_1) \\ \dot{z}_2 &= f_2^T \cdot \mathcal{G} - \dot{\alpha}_1 + g_2 \cdot x_3 \end{aligned} \quad (\text{III.65})$$

pour lequel on choisit comme fonction de Lyapunov

$$V_2(z_1, z_2) = V_1 + \frac{1}{2} z_2^2 \quad (\text{III.66})$$

cette dernière a pour dérivée, le long de la solution de (III.65).

$$\begin{aligned}
\dot{V}_2(z_1, z_2) &= \dot{V}_1 + z_2 \dot{z}_2 \\
&= z_1 \cdot [f_1^T \mathcal{G} + g_1 \cdot (\alpha_1 + z_2) - \dot{\alpha}_0] + z_2 \cdot [f_2^T \mathcal{G} + g_2 x_3 - \dot{\alpha}_1] \\
&= z_1 \cdot [f_1^T \mathcal{G} + g_1 \alpha_1 - \dot{\alpha}_0] + z_2 \cdot [f_2^T \mathcal{G} + g_1 z_1 + g_2 x_3 - \dot{\alpha}_1] \\
&= -c_1 z_1^2 + z_2 \cdot [f_2^T \mathcal{G} + g_1 z_1 + g_2 x_3 - \dot{\alpha}_1]
\end{aligned} \tag{III.67}$$

le choix de la valeur désirée (la fonction stabilisante) de x_3 devient évident. Ce dernier est donné par :

$$(x_3)_d \triangleq \alpha_2 = \frac{1}{g_2} \cdot [\dot{\alpha}_1 - g_1 z_1 - f_2^T \mathcal{G} - c_2 z_2] \tag{III.68}$$

où $c_2 > 0$, avec $\dot{\alpha}_1$ calculée analytiquement

$$\dot{\alpha}_1 = \frac{\partial \alpha_1}{\partial x_1} \dot{x}_1 + \frac{\partial \alpha_1}{\partial y_r} \dot{y}_r + \frac{\partial \alpha_1}{\partial \dot{y}_r} \ddot{y}_r \tag{III.69}$$

un tel choix permet de réduire la dérivée à

$$\begin{aligned}
\dot{V}_2 &\leq -k_1 z_1^2 - k_2 z_2^2 \\
&\leq 0
\end{aligned} \tag{III.70}$$

ce qui assure la stabilité asymptotique de l'origine de (III.65).

Etape3 : le système (III.54) est maintenant considéré dans sa globalité. La variable d'erreur

$$z_3 = x_3 - \alpha_2 \tag{III.71}$$

est définie, ce qui permet d'écrire les équations du système, dans l'espace des erreurs

(z_1, z_2, z_3)

$$\begin{aligned}
\dot{z}_1 &= f_1^T \mathcal{G} - \dot{\alpha}_0 + g_1 \cdot (z_2 + \alpha_1) \\
\dot{z}_2 &= f_2^T \mathcal{G} - \dot{\alpha}_1 + g_2 \cdot (z_3 + \alpha_2) \\
\dot{z}_3 &= f_3^T \mathcal{G} - \dot{\alpha}_2 + g_3 \cdot u
\end{aligned} \tag{III.72}$$

avec comme fonction de Lyapunov

$$V_3(z_1, z_2, z_3) = V_2 + \frac{1}{2} z_3^2 \tag{III.73}$$

la dérivée, le long de la solution de (III.72), devient

$$\begin{aligned} \dot{V}_3(z_1, z_2, z_3) &= \dot{V}_2 + z_3 \dot{z}_3 \\ &= -c_1 z_1^2 - c_2 z_2^2 + z_3 \cdot [g_3 u + g_2 z_2 + f_3^T \mathcal{G} - \dot{\alpha}_2] \end{aligned} \quad (\text{III.74})$$

à présent, on est en présence de la vraie commande u . Un bon choix de celle-ci est donné par

$$u = \frac{1}{g_3} [\dot{\alpha}_2 - g_2 z_2 - f_3^T \mathcal{G} - c_3 z_3] \quad (\text{III.75})$$

où $c_3 > 0$ et $\dot{\alpha}_2$ est également calculée analytiquement,

$$\dot{\alpha}_2 = \frac{\partial \alpha_2}{\partial x_1} \dot{x}_1 + \frac{\partial \alpha_2}{\partial x_2} \dot{x}_2 + \frac{\partial \alpha_2}{\partial y_r} \dot{y}_r + \frac{\partial \alpha_2}{\partial \dot{y}_r} \ddot{y}_r + \frac{\partial \alpha_2}{\partial \ddot{y}_r} y_r^{(3)} \quad (\text{III.76})$$

avec ce choix, on a

$$\dot{V}_3(z_1, z_2, z_3) \leq -c_1 z_1^2 - c_2 z_2^2 - c_3 z_3^2 \leq 0 \quad (\text{III.77})$$

d'où la stabilité asymptotique de l'origine de (III.72). Ceci se traduit par la stabilité, en boucle fermée, du système originel (III.53) et la régulation à zéro de l'erreur de poursuite $y - y_r$. Les deux principaux objectifs du design sont alors atteints.

Remarque 2 : les paramètres de design c_i sont directement liés à la position de pôles de la boucle fermée. Leur choix permet de faire un placement des pôles, fixant ainsi la dynamique en régulation de cette boucle.

b) Cas des systèmes d'ordre n

L'application récursive du backstepping permet l'extension de la procédure de design aux systèmes triangulaires de la forme

$$\begin{aligned} \dot{x}_1 &= f_1^T(x_1) \mathcal{G} + g_1(x_1) x_2 \\ \dot{x}_2 &= f_2^T(x_1, x_2) \mathcal{G} + g_2(x_1, x_2) x_3 \\ &\vdots \\ \dot{x}_{n-1} &= f_{n-1}^T(x_1, x_2, \dots, x_{n-1}) \mathcal{G} + g_{n-1}(x_1, x_2, \dots, x_{n-1}) x_n \\ \dot{x}_n &= f_n^T(x_1, x_2, \dots, x_{n-1}, x_n) \mathcal{G} + g_n(x_1, x_2, \dots, x_{n-1}, x_n) u \end{aligned} \quad (\text{III.78})$$

où $f_i(0) = 0$, $g_i \neq 0$ pour $1 \leq i \leq n$

La procédure de design commence avec la première équation. Le changement de variable adéquat à chaque étape i permet d'appliquer le backstepping récursivement, en rajoutant l'équation $i+1$. Partant de α_0 , on construit les différents α_i et V_i . Ce qui résulte en

$$\begin{aligned} (x_1)_d &\triangleq \alpha_0 = y_r \\ (x_{i+1})_d &\triangleq \alpha_i = \frac{1}{g_i} \left[\sum_{k=1}^{i-1} \left(\frac{\partial \alpha_{i-1}}{\partial x_k} g_k x_{k+1} + \frac{\partial \alpha_{i-1}}{\partial y_r^{(k-1)}} y_r^{(k)} \right) - g_{i-1} z_{i-1} - c_i z_i - \omega_i^T \mathcal{G} \right] \end{aligned} \quad (\text{III.79})$$

où

$$\begin{cases} i = 1, \dots, n \\ \omega_i = f_i - \sum_{k=1}^{i-1} \frac{\partial \alpha_{i-1}}{\partial x_k} f_k \\ z_i = x_i - \alpha_{i-1} \end{cases} \quad (\text{III.80})$$

les différentes fonctions de Lyapunov sont données par

$$V_i = \frac{1}{2} \sum_{j=1}^i [x_j - \alpha_{j-1}]^2 \quad (\text{III.81})$$

La commande u , qui permet d'atteindre les objectifs du design pour le système global, est donnée par la dernière commande virtuelle α_n .

III.7. Synthèse de la commande par backstepping

Comme le backstepping se présente comme étant un algorithme de commande récursif, les étapes de calcul sont comme suit :

III.7.1. Démonstration : soit le sous système suivant :

$$\begin{cases} \dot{x}_1 = x_2 \\ \dot{x}_2 = \frac{(I_y - I_z)}{I_x} x_4 x_6 - \frac{J_r}{I_x} \Omega_r x_4 - \frac{K_{fax}}{I_x} x_2^2 + \frac{1}{I_x} u_2 \end{cases} \quad (\text{III.82})$$

la synthèse de la commande u_2 se fait en deux étapes :

Étape 1 :

On prend le premier sous système

$$\dot{x}_1 = x_2 \quad (\text{III.83})$$

On définit l'erreur z_1 entre ϕ et ϕ_d par : $z_1 = \phi - \phi_d = x_1 - x_{1d}$.

dont la dynamique peut être dérivée comme suit :

$$\dot{z}_1 = x_2 - \dot{x}_{1d} \quad (\text{III.84})$$

cette définition indique explicitement notre objectif de commande : l'erreur z_1 doit converger asymptotiquement vers zéro.

on choisit la première fonction de Lyapunov candidate de la forme suivante:

$$V(z_1) = \frac{1}{2} z_1^2 \quad (\text{III.85})$$

le calcul de la dérivée de la fonction de Lyapunov le long de la trajectoire est résolu en employant (III.84):

$$\dot{V}(z_1) = z_1 \dot{z}_1 = z_1 (x_2 - \dot{x}_{1d}) \quad (\text{III.86})$$

pour assurer la stabilité il faut que $\dot{V}(z_1) \leq 0$, pour cela on prend comme commande virtuelle \tilde{x}_2 avec :

$$\tilde{x}_2 = \dot{x}_{1d} - c_1 z_1, \quad c_1 > 0 \quad (\text{III.87})$$

ce choix permet d'obtenir :

$$\dot{V}(z_1) = -c_1 z_1^2 \quad (\text{III.88})$$

Etape 2 :

Comme la commande virtuelle ne peut pas prendre instantanément sa valeur désirée, on cherche dans ce qui suit à stabiliser l'erreur entre la commande virtuelle et la fonction stabilisante. Le deuxième sous système est défini par :

$$\dot{x}_2 = \frac{(I_y - I_z)}{I_x} x_4 x_6 - \frac{J_r}{I_x} \Omega_r x_4 - \frac{K_{fax}}{I_x} x_2^2 + \frac{1}{I_x} u_2 \quad (\text{III.89})$$

la nouvelle variable d'erreur, entre la commande virtuelle et la fonction stabilisante, est donnée par :

$$z_2 = x_2 - \tilde{x}_2 \quad (\text{III.90})$$

$$z_2 = x_2 - \dot{x}_{1d} + c_1 z_1 \quad (\text{III.91})$$

en utilisant (III.84) et (III.91), on trouve :

$$\begin{cases} \dot{z}_1 = z_2 - c_1 z_1 \\ \dot{z}_2 = \dot{x}_2 - \ddot{x}_{1d} + c_1 (z_2 - c_1 z_1) \end{cases} \quad (\text{III.92})$$

la nouvelle fonction de Lyapunov du système augmenté est définie comme suit:

$$V(z_1, z_2) = \frac{1}{2} z_1^2 + \frac{1}{2} z_2^2 \quad (\text{III.93})$$

sa dérivée est donnée par :

$$\dot{V}(z_1, z_2) = z_1 \dot{z}_1 + z_2 \dot{z}_2 \quad (\text{III.94})$$

on remplace \dot{z}_1 et \dot{z}_2 dans (III.94), on aura:

$$\dot{V}(z_1, z_2) = z_1 (z_2 - c_1 z_1) + z_2 (\dot{x}_2 - \ddot{x}_{1d} + c_1 (z_2 - c_1 z_1)) \quad (\text{III.95})$$

le choix de la loi de commande doit mener à une dérivée négative de la fonction de Lyapunov, $\dot{V}(z_1, z_2) \leq 0$, comme suit :

$$\dot{V}(z_1, z_2) = -c_1 z_1^2 - c_2 z_2^2 \quad (\text{III.96})$$

avec $c_1, c_2 > 0$

en utilisant (III.95), on trouve :

$$-c_2 z_2 = z_1 + \frac{(I_y - I_z)}{I_x} x_4 x_6 - \frac{J_r}{I_x} \Omega_r x_4 - \frac{K_{fax}}{I_x} x_2^2 + \frac{1}{I_x} u_2 - \ddot{x}_{1d} + c_1 (z_2 - c_1 z_1) \quad (\text{III.97})$$

donc la commande u_2 qui assure la poursuite est :

$$u_2 = I_x \left[\ddot{x}_{1d} - z_1 - c_2 z_2 - c_1 (z_2 - c_1 z_1) - \frac{(I_y - I_z)}{I_x} x_4 x_6 + \frac{J_r}{I_x} \Omega_r x_4 + \frac{K_{fax}}{I_x} x_2^2 \right] \quad (\text{III.98})$$

Les mêmes étapes sont reprises afin d'extraire, u_1, u_3, u_4, u_x, u_y

$$\begin{aligned} u_1 &= \frac{m}{(\cos(\phi)\cos(\theta))} \left(\ddot{x}_{11d} - z_3 - c_3 (z_4 - c_3 z_3) - c_4 z_4 + g + \frac{K_{fz}}{m} x_{12} \right) \\ u_3 &= I_y \left(\ddot{x}_{3d} - z_5 - c_5 (z_6 - c_5 z_5) - c_6 z_6 - \left(\frac{I_z - I_x}{I_y} \right) x_2 x_6 - \frac{J_r}{I_y} \Omega_r x_2 + \frac{K_{fay}}{I_y} x_4^2 \right) \\ u_4 &= I_z \left(\ddot{x}_{5d} - z_7 - c_7 (z_8 - c_7 z_7) - c_8 z_8 - \left(\frac{I_x - I_y}{I_z} \right) x_2 x_4 + \frac{K_{faz}}{I_z} x_6^2 \right) \\ u_x &= \frac{m}{u_1} \left(\ddot{x}_{7d} - z_9 - c_9 (z_{10} - c_9 z_9) - c_{10} z_{10} + \frac{K_{fzx}}{m} x_8 \right) \\ u_y &= \frac{m}{u_1} \left(\ddot{x}_{9d} - z_{11} - c_{11} (z_{12} - c_{11} z_{11}) - c_{12} z_{12} + \frac{K_{fzy}}{m} x_{10} \right) \text{ avec: } c_3, \dots, c_{12} > 0 \end{aligned} \quad (\text{III.99})$$

III.7.2 Résultats de simulation :

Pour illustrer les performances du réglage avec la commande backstepping, nous avons effectué les tests suivants :

Avec : $c_1=c_2=c_3=c_4=c_5=c_6=c_7=c_8=c_9=c_{10}=c_{11}=c_{12}=11$.

D'après les figures III.42 et III.44, nous remarquons que la trajectoire simulée suit bien la référence avec des erreurs de très faibles valeurs et pour les commandes, on remarque d'après la figure III.43 que u_1 est approximativement égale à mg le cas d'équilibre de notre drone et les commandes u_4, u_3, u_2 tendent vers zéro après $t=30$ sec.

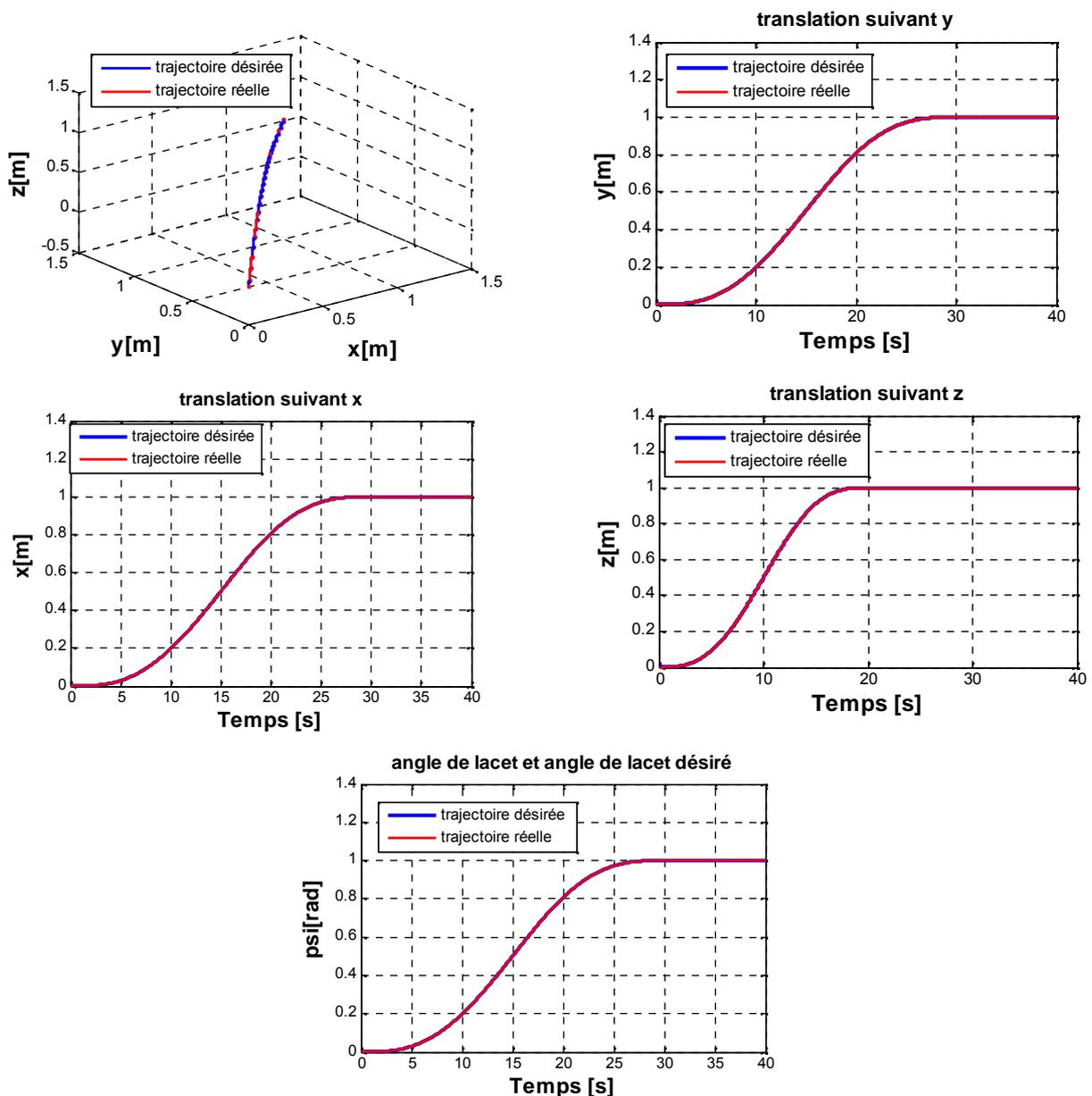


Figure III.42. Résultats de poursuite des trajectoires désirées selon (x, y, z, ψ) .

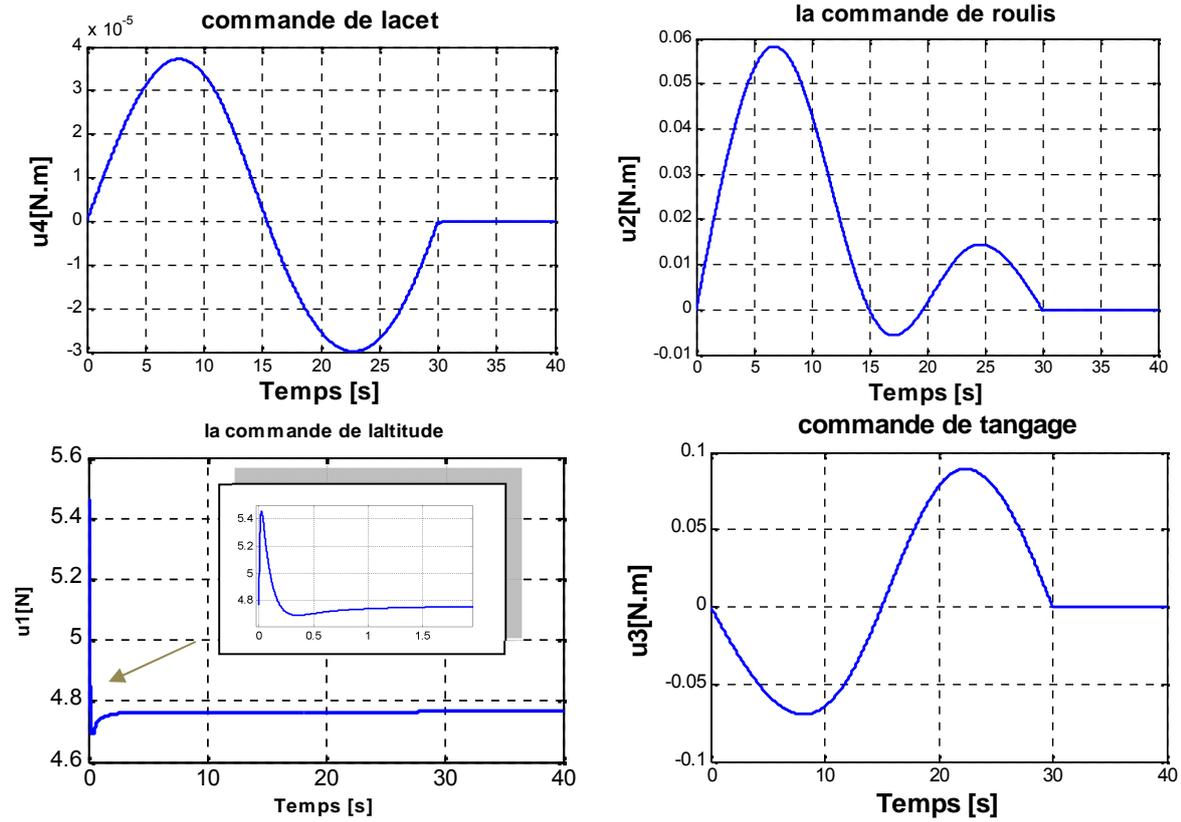


Figure III.43. Les commandes (u_1 u_2 u_3 u_4)

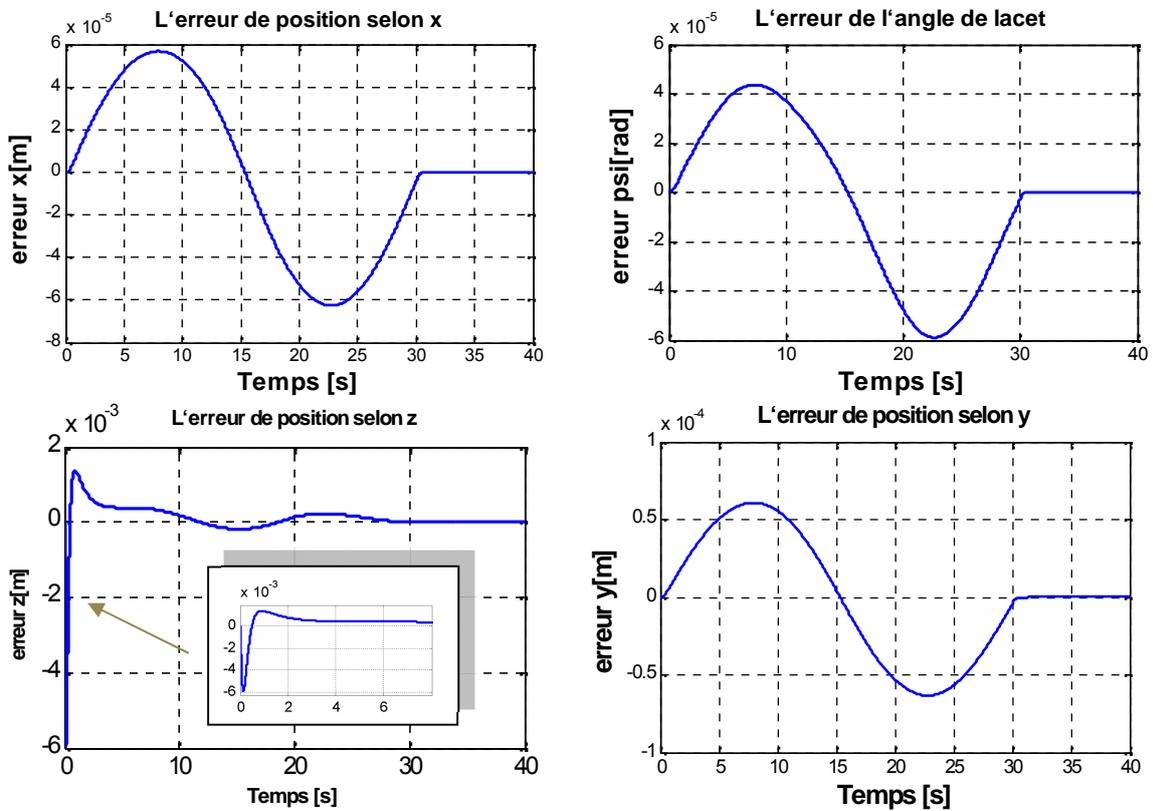


Figure III.44. Les erreurs de déplacements suivant (x , y , z) et l'angle ψ .

III.7.3. Etude de la robustesse

Si on applique une force perturbatrice $D_z = 7 \sin(0.1t)$ à l'instant $t=25s$ suivant (z) on obtient les figures suivantes:

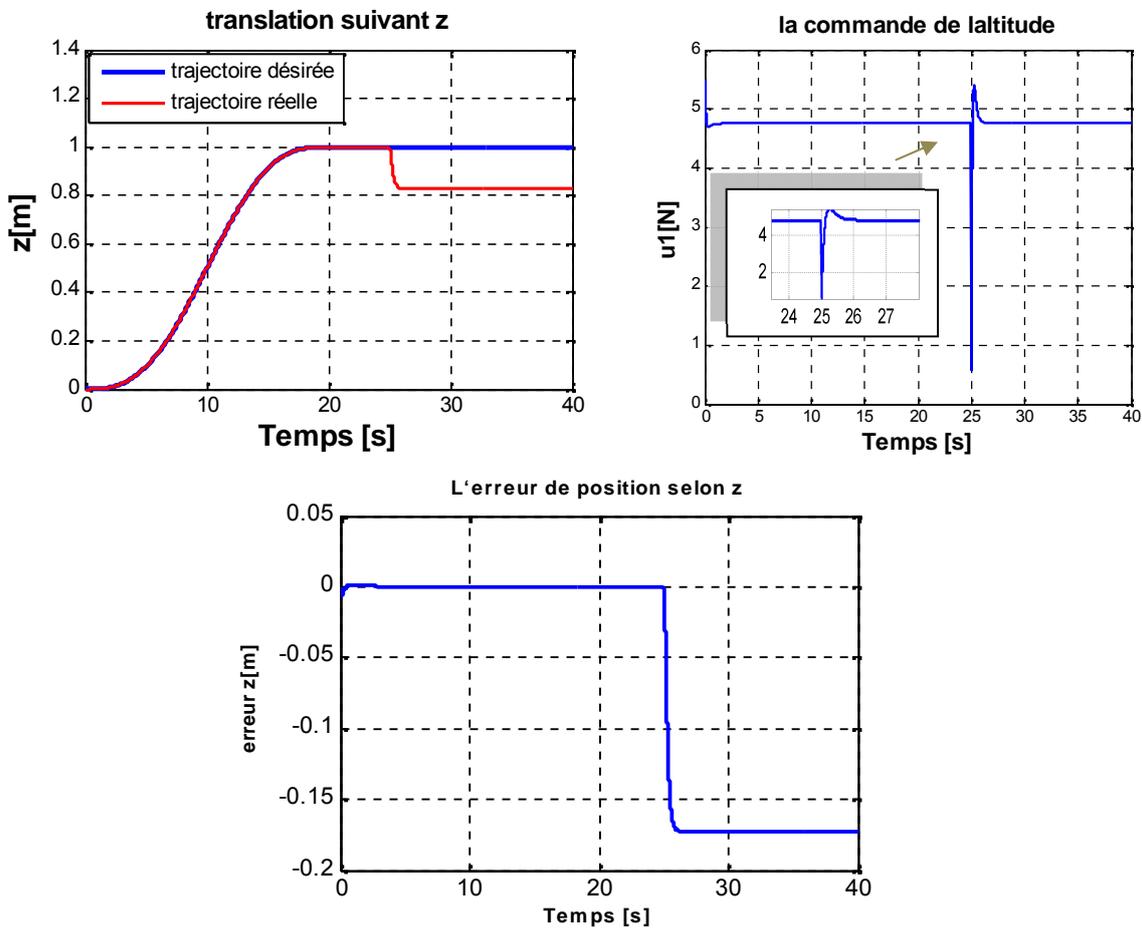


Figure III.45. L'influence de la perturbation selon (z)

D'après la figure III.45, nous remarquons que la commande backstepping génère une erreur statique après avoir appliqué une force de traînée « vent » au mouvement de drone dans le cas de la commande du vol vertical à l'instant $t=25s$ donc finalement on peut dire que la commande n'assure pas le rejet de perturbation, le système est stabilisé, mais l'erreur n'est pas nulle afin d'améliorer cette commande on essaye d'introduire une action intégrale pour éliminer l'erreur statique.

III.8. Synthèse de la commande par l'intégrale backstepping

Après avoir étudié la commande backstepping, il est apparu clairement que pour obtenir un contrôleur robuste vis-à-vis aux perturbations externes, nous allons essayer de combiné entre le PID et la commande backstepping. Ceci permettra d'avoir la commande intégrale backstepping [Bel.12.b].

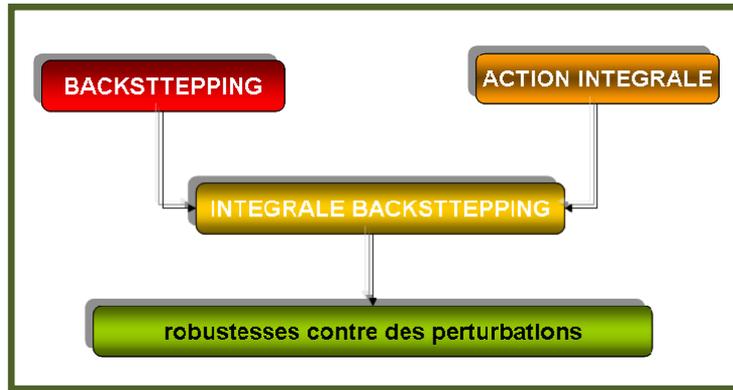


Figure III.46. L'approche de commande proposée.

III.8.1.Démonstration : soit le sous système suivant pour la commande de l'angle de roulis :

$$\begin{cases} \dot{x}_1 = x_2 \\ \dot{x}_2 = \frac{(I_y - I_z)}{I_x} x_4 x_6 - \frac{J_r}{I_x} \Omega_r x_4 - \frac{K_{fax}}{I_x} x_2^2 + \frac{1}{I_x} u_2 \end{cases} \quad (\text{III.100})$$

Etape1 : la première étape dans la commande backstepping avec action intégrale c'est définir l'erreur à partir de premier sous système :

$$\dot{x}_1 = x_2 \quad (\text{III.101})$$

avec :

$$e_1 = x_1 - x_{1d} \quad (\text{III.102})$$

dont la dynamique peut être dérivée comme suite :

$$\dot{e}_1 = \dot{x}_1 - \dot{x}_{1d} = x_2 - \dot{x}_{1d} \quad (\text{III.103})$$

pour assurer la stabilité il faut que de la dérivée de la fonction de Lyapunov $\dot{V}(e_1) \leq 0$ pour cela on prend comme commande virtuelle \tilde{x}_2 :

$$\tilde{x}_2 = \dot{x}_{1d} - \beta_1 e_1 - \lambda_1 \chi_1 \quad (\text{III.104})$$

avec : $\beta_1 > 0$ et $\lambda_1 > 0$, dont $\chi_1 = \int_0^t e_1(\tau) d\tau$, c'est l'intégrale de l'erreur e_1 ainsi le terme χ_1 sera

introduit pour assurer la convergence de l'erreur vers zéro.

Etape 2: on cherche maintenant à stabiliser l'erreur entre la commande virtuelle et la fonction stabilisante où la nouvelle variable d'erreur est définie par :

$$e_2 = x_2 - \tilde{x}_2 \quad (\text{III.105})$$

$$e_2 = x_2 - \dot{x}_{1d} + \beta_1 e_1 + \lambda_1 \chi_1 \quad (\text{III.106})$$

dont la dynamique peut être dérivée comme suit :

$$\dot{e}_2 = \dot{x}_2 - \ddot{x}_{1d} + \beta_1 \dot{e}_1 + \lambda_1 \dot{\chi}_1 \quad \text{avec: } \dot{\chi}_1 = e_1 \quad (\text{III.107})$$

on utilisant l'équation (III.104) et (III.105) et (III.103) on trouve :

$$\dot{e}_1 = -\beta_1 e_1 - \lambda_1 \chi_1 + e_2 \quad (\text{III.108})$$

on remplace \dot{x}_2 dans (III.108), par son expression correspondante on aura :

$$\dot{e}_2 = \frac{(I_y - I_z)}{I_x} x_4 x_6 - \frac{J_r}{I_x} \Omega_r x_4 - \frac{K_{fax}}{I_x} x_2^2 + \frac{1}{I_x} u_2 - \ddot{x}_{1d} + \beta_1 \dot{e}_1 + \lambda_1 e_1 \quad (\text{III.109})$$

ainsi, on utilisant l'équation (III.108) et (III.109) on trouve :

$$\dot{e}_2 = \beta_1 (-\beta_1 e_1 - \lambda_1 \chi_1 + e_2) - \ddot{x}_{1d} + \lambda_1 e_1 + \frac{(I_y - I_z)}{I_x} x_4 x_6 - \frac{J_r}{I_x} \Omega_r x_4 - \frac{K_{fax}}{I_x} x_2^2 + \frac{1}{I_x} u_2 \quad (\text{III.110})$$

la nouvelle fonction de Lyapunov du système augmenté est définie comme suit:

$$V = \frac{\lambda_1}{2} \chi_1^2 + \frac{1}{2} e_1^2 + \frac{1}{2} e_2^2 \quad (\text{III.111})$$

incluant l'erreur e_1 et son intégrale et l'erreur e_2 .

le choix de loi de commande doit mener à une dérivée négative de la fonction de Lyapunov :

$$\dot{V}(e_1, e_2) = -\beta_1 e_1^2 - \beta_2 e_2^2 \leq 0 \quad (\text{III.112})$$

avec $\beta_1, \beta_2 > 0$

la dynamique souhaitable pour l'erreur est définie comme suit :

$$\dot{e}_2 = -\beta_2 e_2 - e_1 \quad (\text{III.113})$$

donc la commande u_2 qui assure la poursuite est :

$$u_2 = I_x \left[\ddot{x}_{1d} + (\lambda_1 + \beta_1^2 - 1)e_1 - (\beta_2 + \beta_1)e_2 + \beta_1 \lambda_1 \chi_1 - \frac{(I_y - I_z)}{I_x} x_4 x_6 + \frac{J_r}{I_x} \Omega_r x_4 + \frac{K_{fax}}{I_x} x_2^2 \right] \quad (\text{III.114})$$

Les mêmes étapes sont reprises afin d'extraire, u_1, u_3, u_4, u_x, u_y

$$\begin{aligned}
u_1 &= \frac{m}{(\cos(\phi)\cos(\theta))} \left(\ddot{x}_{11d} + (\lambda_2 + \beta_3^2 - 1)e_3 - (\beta_4 + \beta_3)e_4 + \beta_3\lambda_2\chi_2 + \frac{K_{fz}}{m}x_{12} + g \right) \\
u_3 &= I_y \left(\ddot{x}_{3d} + (\lambda_3 + \beta_5^2 - 1)e_5 - (\beta_6 + \beta_5)e_6 + \beta_5\lambda_3\chi_3 - \left(\frac{I_z - I_x}{I_y} \right) x_2 x_6 - \frac{J_r}{I_y} \Omega_r x_2 + \frac{K_{fay}}{I_y} x_4^2 \right) \\
u_4 &= I_z \left(\ddot{x}_{5d} + (\lambda_4 + \beta_7^2 - 1)e_7 - (\beta_8 + \beta_7)e_8 + \beta_7\lambda_4\chi_4 - \left(\frac{I_x - I_y}{I_z} \right) x_2 x_4 + \frac{K_{faz}}{I_z} x_6^2 \right) \quad (III.115) \\
u_x &= \frac{m}{u_1} \left(\ddot{x}_{7d} + (\lambda_x + \beta_9^2 - 1)e_9 - (\beta_{10} + \beta_9)e_{10} + \beta_9\lambda_x\chi_x + \frac{K_{fzx}}{m}x_8 \right) \\
u_y &= \frac{m}{u_1} \left(\ddot{x}_{9d} + (\lambda_y + \beta_{11}^2 - 1)e_{11} - (\beta_{12} + \beta_{11})e_{12} + \beta_{11}\lambda_y\chi_y + \frac{K_{fzy}}{m}x_{10} \right) \\
\text{avec: } & (\beta_3 \dots \beta_{12}) > 0 \text{ et } (\lambda_2, \lambda_3, \lambda_4, \lambda_x, \lambda_y) > 0
\end{aligned}$$

Dont les erreurs et l'intégrale des erreurs de poursuite sont écrites sous la forme ci-dessous :

$$\begin{aligned}
e_3 &= x_{11} - x_{11d} & \chi_2 &= \int_0^t e_3(\tau) d\tau \\
e_5 &= x_3 - x_{3d} & \chi_3 &= \int_0^t e_5(\tau) d\tau \\
e_7 &= x_5 - x_{5d} & \chi_4 &= \int_0^t e_7(\tau) d\tau \\
e_9 &= x_7 - x_{7d} & \chi_x &= \int_0^t e_9(\tau) d\tau \\
e_{11} &= x_9 - x_{9d} & \chi_y &= \int_0^t e_{11}(\tau) d\tau
\end{aligned} \quad (III.116)$$

III.8.2 Résultats de simulation

Pour illustrer les performances du réglage avec la commande intégrale backstepping, nous avons effectué les tests suivants :

Avec : $1= 2= 3= 4= 5= 6= 7= 8= 9= 10= 11= 12=7$

et $2= 3= 4= x= y=10$;

D'après les résultats obtenus des figures III.47, III.48, III.49 on remarque que notre contrôleur assure la stabilité avec une amélioration dans la précision de poursuite « erreur de l'ordre 10^{-6} » à cause de l'action intégrale introduite dans la commande backstepping. Les signaux de commandes $(u_1 u_2 u_3 u_4)$ démontrent que la condition l'état d'équilibre est toujours vérifiée après avoir effectué une manœuvre.

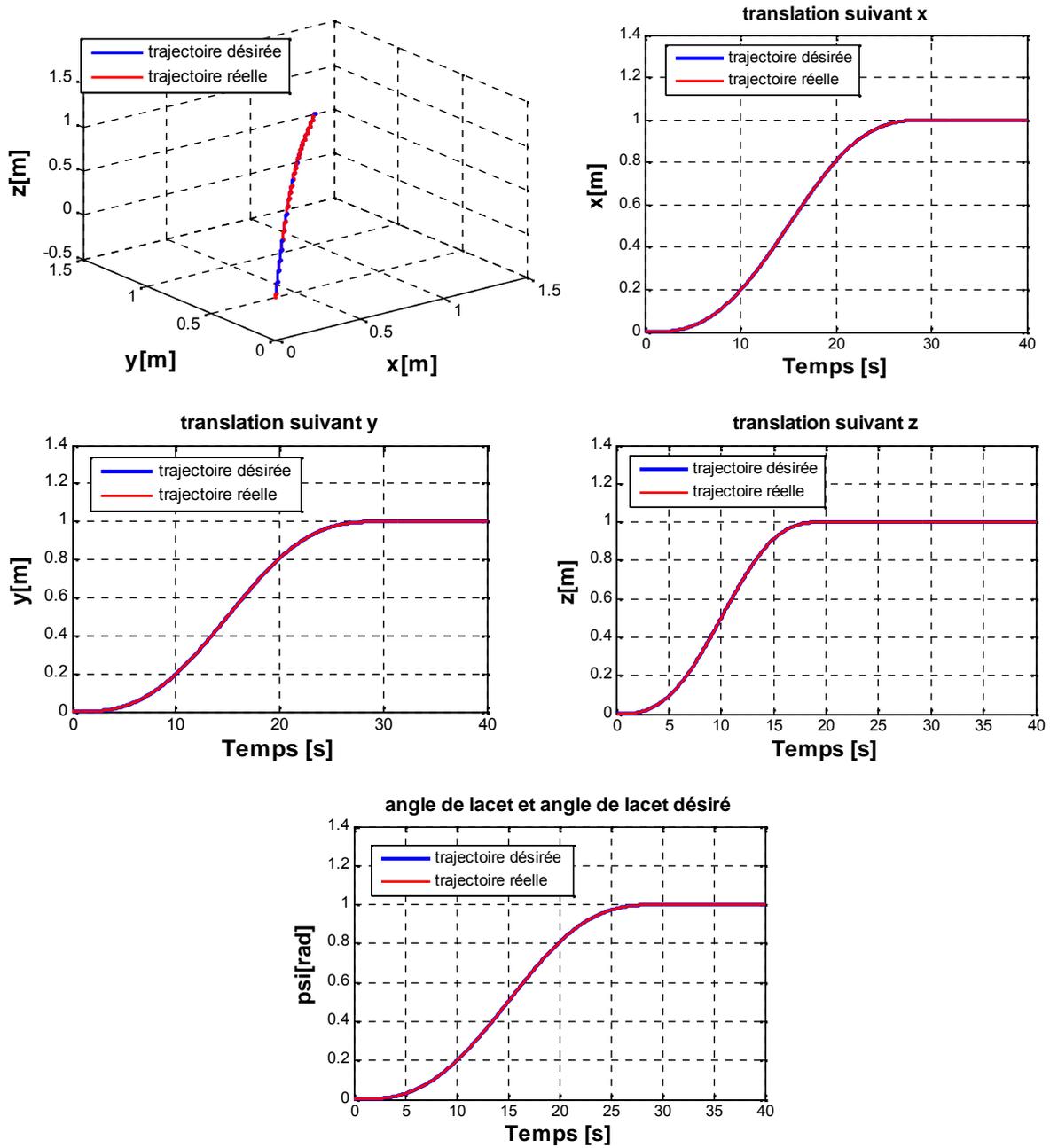
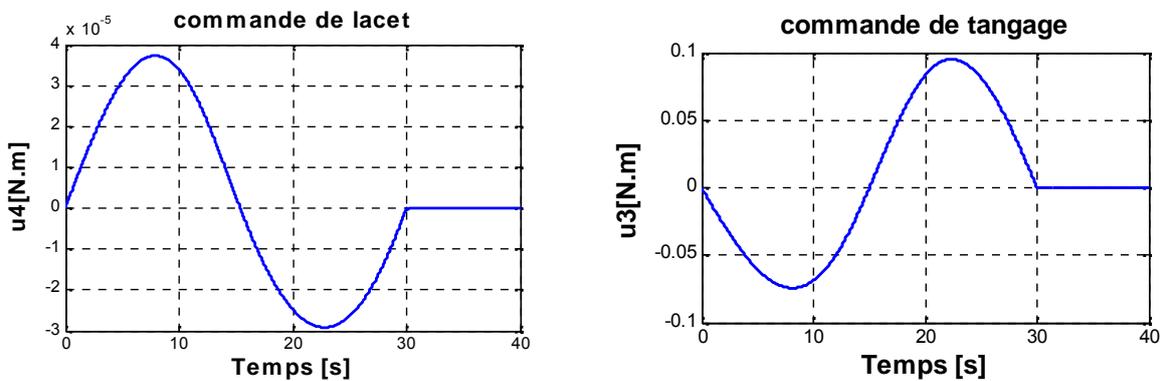
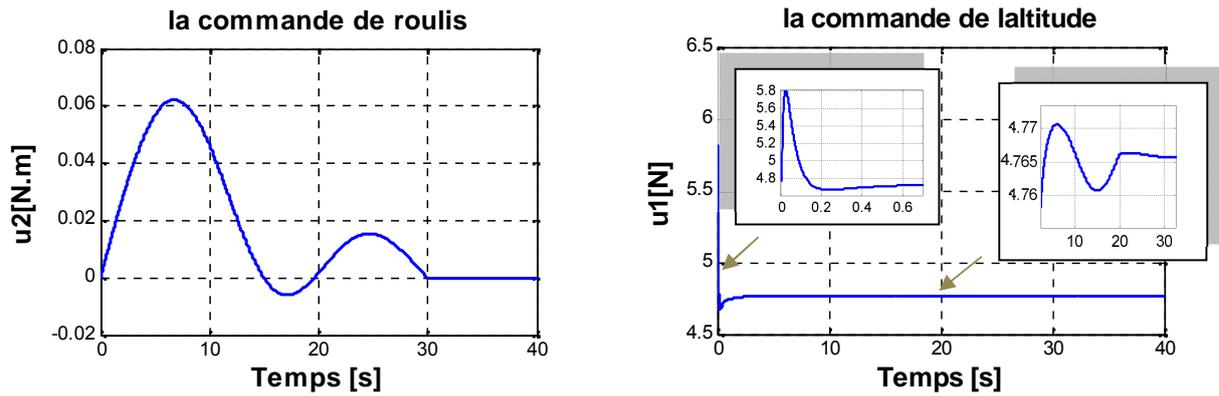
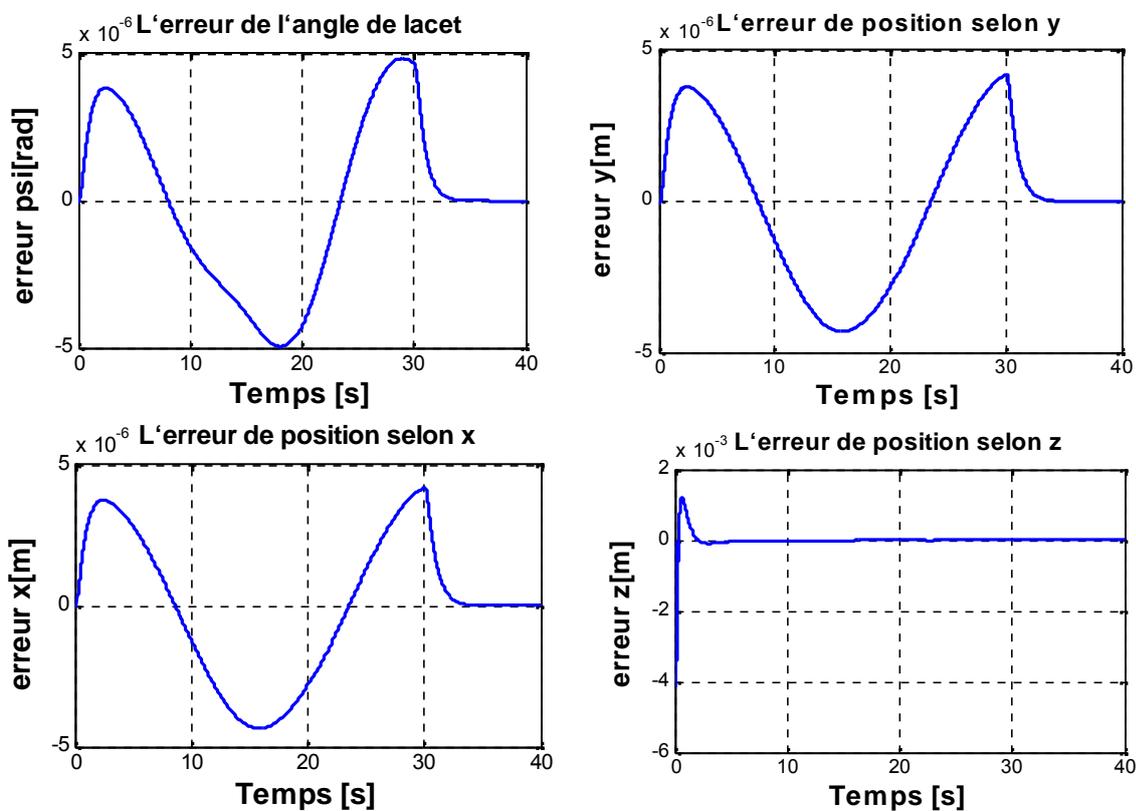


Figure III.47. Résultats de poursuite des trajectoires désirées selon (x, y, z, ψ) .



Figure III.48. Les commandes (u_1, u_2, u_3, u_4) Figure III.49. Les erreurs de déplacements suivant (x, y, z) et l'angle ψ .

III.8.3. Etude de la robustesse

Afin de tester la robustesse de notre nouvelle commande nous avons effectués des tests suivants :

- On applique une force perturbatrice $D_z = 7 \sin(0.1t)$ à l'instant $t=25\text{sec}$ suivant (z) .
- On applique des forces perturbatrices : $d_x = 2 \sin(0.1t)$; $d_y = 1.5 \sin(0.1t)$; $d_z = 2 \sin(0.1t)$ respectivement à l'instant $t = 25\text{sec}$, $t=35\text{sec}$ et $t=20\text{sec}$ avec des incertitudes paramétriques +20% de (I_x, I_y, I_z) .

c) On compare entre les deux commandes suivant (z) avec : $D_z = 7 \sin(0.1t)$ à l'instant $t=25\text{sec}$ et des incertitudes paramétriques +20% de (I_x, I_y, I_z) .

D'après les résultats des figures III.50, III.51 et III.52 III.53 on voit clairement que la commande par intégrale backstepping est plus performante. Elle montre une meilleure robustesse vis-à-vis aux incertitudes paramétriques et aux perturbations externe par rapport à la commande backstepping. Cela est dû au fait que nous avons introduit une action intégrale qui à permet d'annuler l'erreur statique.

Les résultats de simulations sont illustrés dans les figures suivantes:

a)

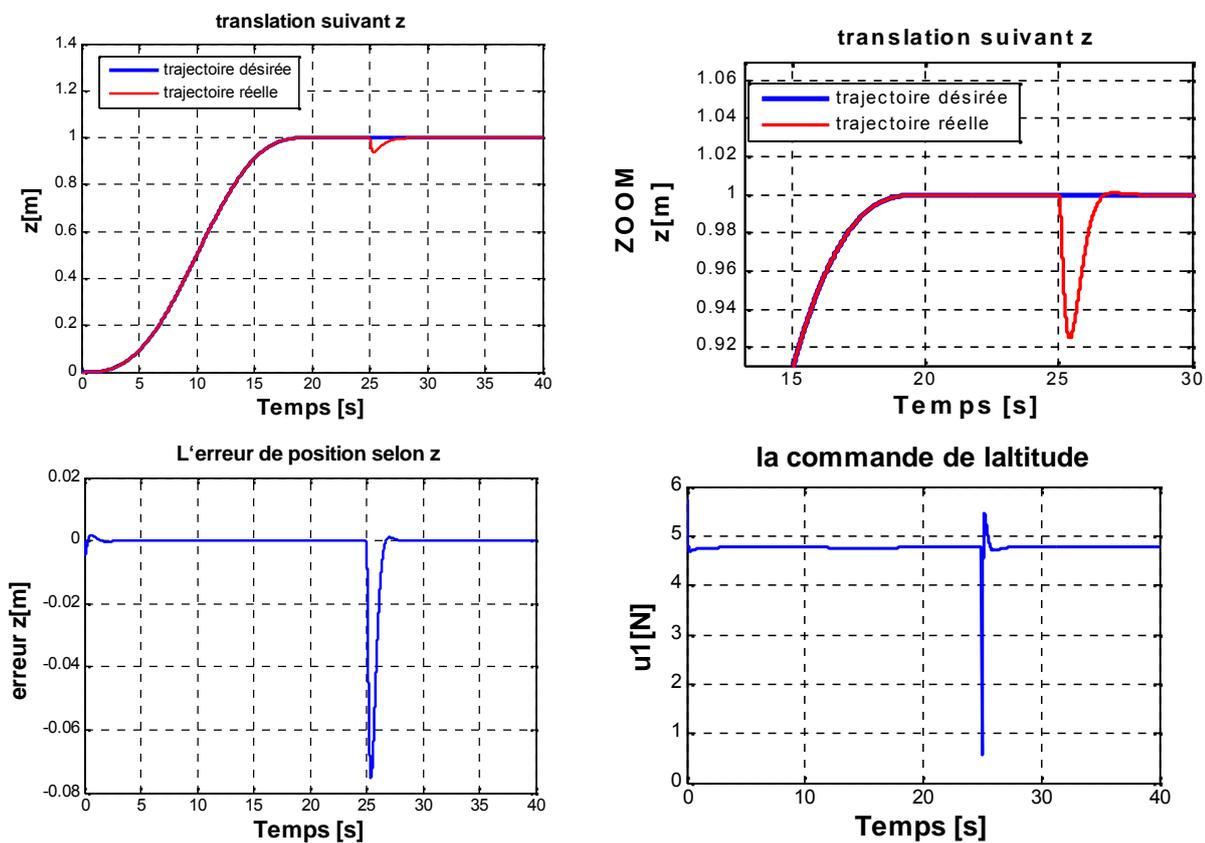


Figure III.50. L'influence de la perturbation selon (z) pour le contrôleur intégrale backstepping.

b)

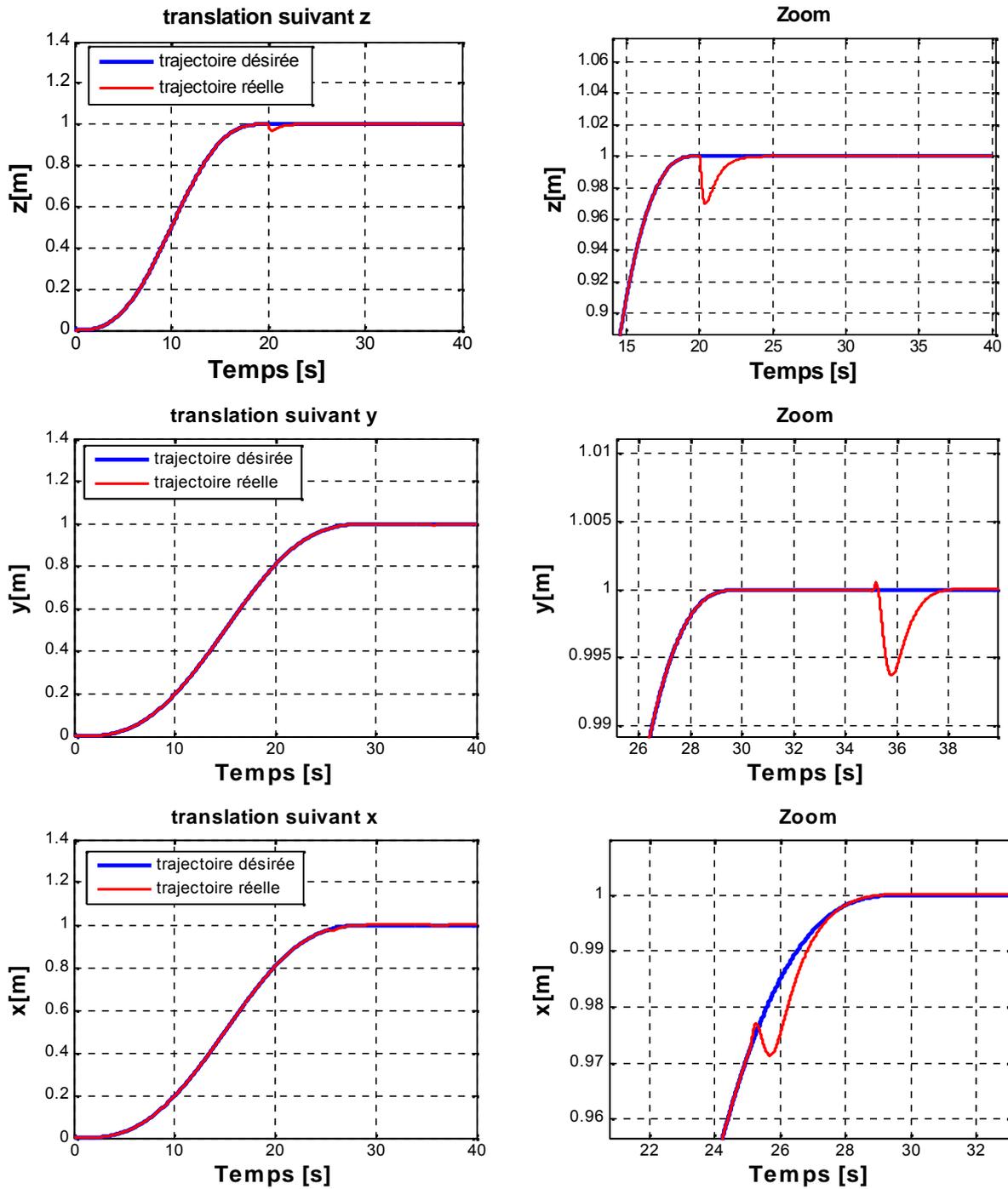


Figure III.51. L'influence de la perturbation selon (x, y, z) avec incertitudes paramétriques +20% de (I_x, I_y, I_z) pour le contrôleur intégrale backstepping.

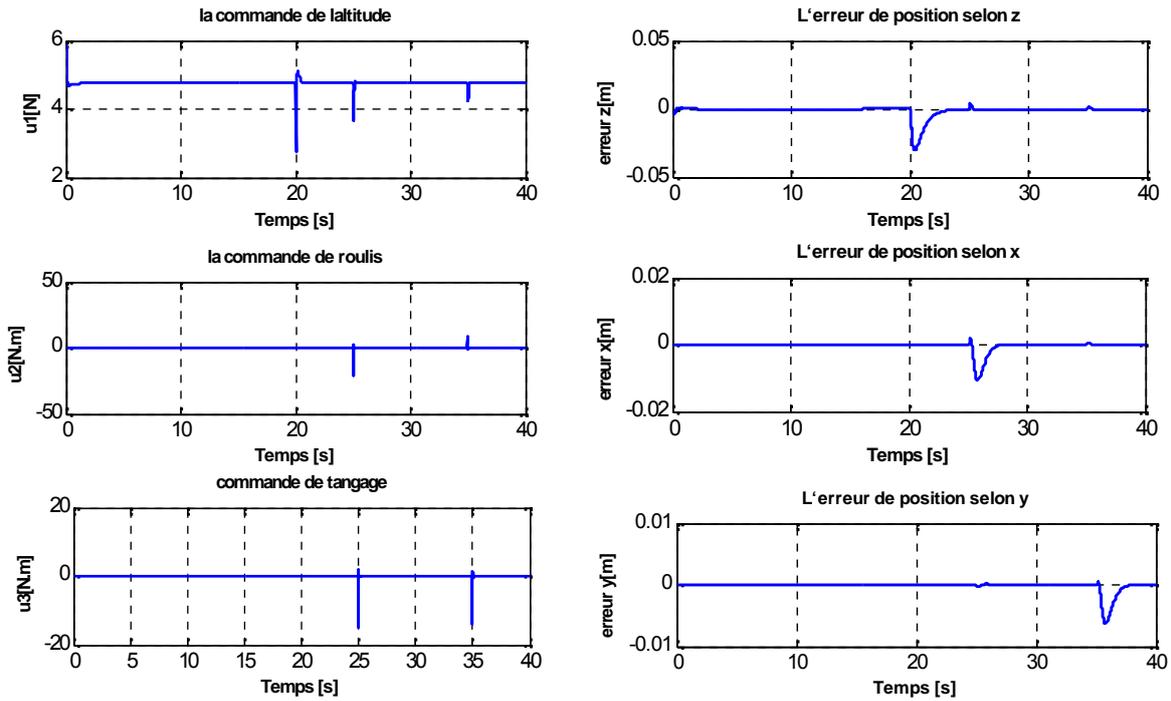


Figure III.52. Les signaux de commandes et des erreurs selon (x, y, z) avec incertitudes paramétrique +20% de (I_x, I_y, I_z) pour le contrôleur intégrale backstepping

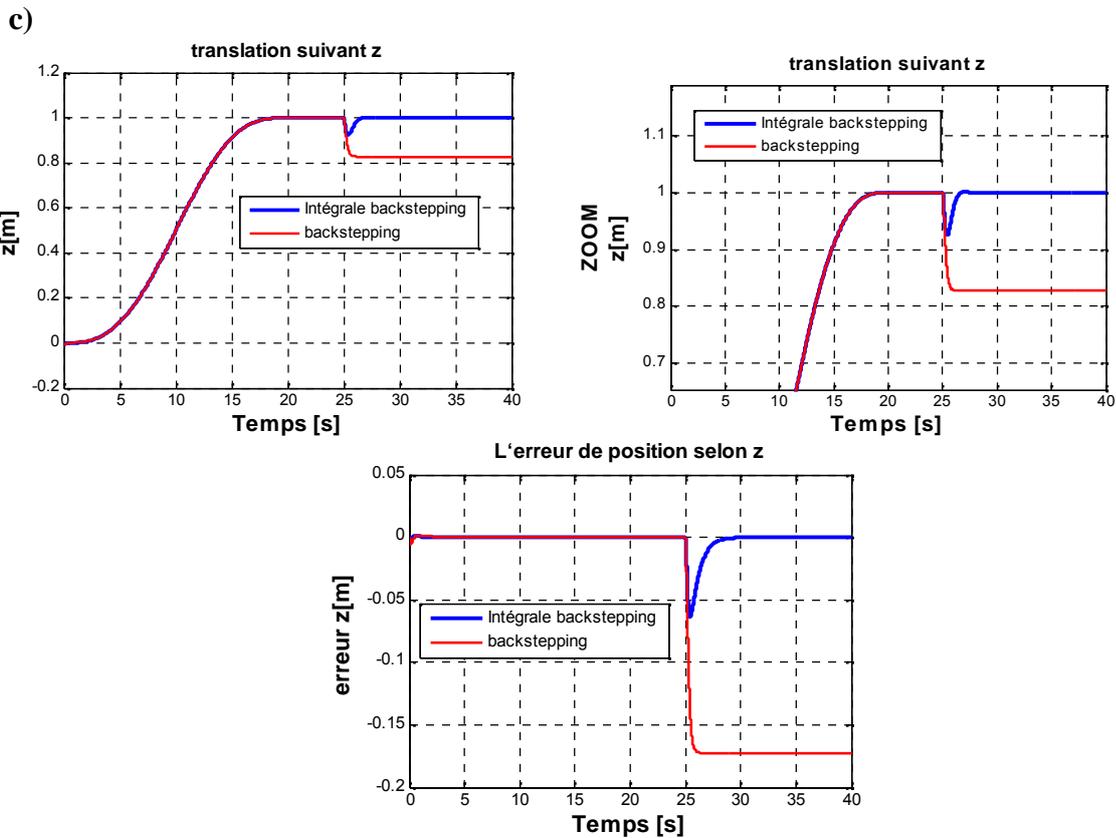


Figure III.53. L'influence de la perturbation selon (z) , du contrôleur backstepping et intégrale backstepping avec des incertitudes paramétriques +20% de (I_x, I_y, I_z) .

Après avoir vu ces résultats, nous pouvons conclure ce qui suit :

Le backstepping est une technique de commande qui assure la stabilité au sens de Lyapunov.

Les lois de commande synthétisées par cette technique prennent en considération toutes les non linéarités du système ainsi que la dynamique des trajectoires désirées.

La technique présente une souplesse de manipulation et l'introduction de l'action intégrale nous a permis d'avoir une commande plus robuste.

III.9. Etude comparative

Pour examiner les différentes lois de commande, développées pour le quadrirotor, nous avons opté pour une étude comparative entre ces différentes commandes. Pour cela, nous définissons un critère, en fonction de l'erreur résultante.

Pour bien examiner ces commandes, les résultats ont été pris dans les mêmes conditions, à titre d'exemple : la période d'échantillonnage, le temps de simulation, ...etc.

Critère	Commande par mode de glissement	Commande par mode flou glissant	Commande par PIFSMC-GA	Commande par le backstepping	Commande par l'intégrale backstepping
$J_z = \sum_{i=1}^n e_z^t e_z$	$3.0081 \cdot 10^{-5}$	$2.383 \cdot 10^{-5}$	$1.418 \cdot 10^{-5}$	0.0090	0.0031
$J_y = \sum_{i=1}^n e_y^t e_y$	0.0060	$4.397 \cdot 10^{-6}$	$3.783 \cdot 10^{-7}$	$2.998 \cdot 10^{-5}$	$2.603 \cdot 10^{-7}$
$J_x = \sum_{i=1}^n e_x^t e_x$	0.0044	$4.670 \cdot 10^{-6}$	$1.479 \cdot 10^{-7}$	$3.192 \cdot 10^{-5}$	$2.596 \cdot 10^{-7}$
$J_\psi = \sum_{i=1}^n e_\psi^t e_\psi$	$5.549 \cdot 10^{-4}$	$8.418 \cdot 10^{-6}$	$1.122 \cdot 10^{-8}$	$3.855 \cdot 10^{-5}$	$3.328 \cdot 10^{-7}$

Tableau III.2 : Étude comparative entre les commandes développées pour le quadrirotor.

Pour mieux éclaircir cette comparaison, on s'est intéressé à l'évolution de l'erreur résultante entre la commande « PIFSMC-GA » et « Intégrale backstepping ». L'étude des deux techniques de commande illustrées précédemment est effectuée par rapport à la robustesse vis-à-vis des perturbations extérieures suivant (x,y,z) et des incertitudes paramétriques (I_x, I_y, I_z) :

Remarque 2: Ces deux contrôleurs ont présenté un rejet relatif des perturbations dans l'étude précédente.

Critère	Commande par PIFSMC-GA	Commande par l'intégrale backstepping
$J_z = \sum_{i=1}^n e_z^t e_z$	0.00069	0.7266
$J_y = \sum_{i=1}^n e_y^t e_y$	0.0013	0.0283
$J_x = \sum_{i=1}^n e_x^t e_x$	0.0046	0.0811

Tableau III.3: Etude comparative entre les deux techniques de commande avec des perturbations et des incertitudes paramétriques.

Dans les deux tableaux, nous avons présenté une comparaison quantitative entre les différentes approches développées précédemment pour mieux éclaircir les résultats obtenus par des graphes.

En conclusion à cette étude comparative entre les différentes techniques de commande utilisées pour le contrôle du quadrirotor ; nous constatons ce qui suit :

- la commande par mode de glissement représente les performances les moins appréciées par rapport aux autres techniques.
- la commande « PIFSMC-GA » représente les meilleures performances car les critères prennent les valeurs les plus faibles malgré les perturbations et les incertitudes paramétriques (Tableau III.3).

III.10. Conclusion

En se basant sur le modèle dynamique présenté dans le chapitre II nous avons utilisé deux techniques de commande pour le control du quadrirotor avec des approches d'hybridation comme la logique flou et le PID et l'optimisation par l'algorithme génétique, ces derniers ont donné des résultats de simulation assez acceptables vu les performances réalisées, plusieurs tests de robustesse ont été effectués et cela en ajoutant des perturbations de nature externe comme le vent et des incertitudes paramétriques . Les deux contrôleurs « PIFSMC-GA » et « l'intégrale backstepping » ont présenté un rejet relatif à ces perturbations et ils ont su garder la stabilité du système.

Le quatrième chapitre sera consacré à la synthèse d'un observateur non linéaire afin d'estimer les états non accessibles.



Chapitre IV

Synthèse d'un observateur non linéaire

IV.1. Introduction

Il est clair que toute notre étude précédente est faite sur la base que les états de la sortie sont accessibles pour l'utilisateur, la question qui se pose d'elle même est : comment faire si quelques états ne sont pas accessibles ou en autre terme ne sont pas mesurables et comment faire si nous avons des sorties trop bruitées ?

Bien que les signaux non-mesurables puissent être obtenus par une dérivée successive des équations, cependant, ils sont souillés par le bruit de mesure à un tel degré que la dérivée ne peut plus être employée.

La réponse à cette question qui revient souvent dans le domaine de la commande des systèmes dans les milieux contraints et évolutifs est la synthèse d'un observateur non linéaire. Dans plusieurs applications dans le domaine de la commande, il est souhaitable d'avoir des estimateurs pour certains variables qui ne sont pas mesurables en raison des contraintes techniques ou économique, Par conséquent, le recours à un observateur est indispensable.

Un observateur est un système dynamique que l'on peut appeler capteur informatique, puisqu'il est souvent implanté sur ordinateur afin de reconstituer ou d'estimer en temps réel l'état courant d'un système, à partir des mesures disponibles, des entrées du système et une connaissance à priori du modèle. Il nous permet alors de suivre l'évolution de l'état en tant qu'information sur le système pour les systèmes non linéaires, la synthèse d'observateurs est encore un problème ouvert.

Une des classes les plus connues des observateurs robustes est celle des observateurs par modes glissants. Ce type d'observateur est basé sur la théorie des systèmes à structure variable. Dans ce chapitre, nous présentons une structure de commande à travers la synthèse d'un observateur non linéaire.

Le schéma de principe d'un observateur est montré sur la figure (IV.1) avec $y(t)$ représente les sorties mesurées et $u(t)$ l'entrée de commande et $\hat{x}(t)$ est l'état estimé.

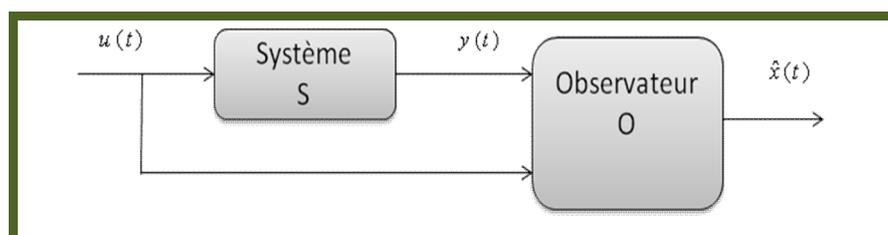


Figure IV.1. Schéma de principe d'un observateur.

IV.2.Observabilité des systèmes non linéaire [Aev.79, Her.77, Lan.01]

Pour les systèmes non linéaires, étant donné l'espace de l'état $X \subseteq R^n$ et l'ensemble U des entrées, la notion d'observabilité est basée sur la possibilité de différencier deux conditions initiales distinctes.

Soit un système non linéaire tel que :

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

Où:

- L'état $x(t) \in X$ ouvert de R^n
- L'entrée $u(t) \in U$ ensemble mesurable de R^m
- Les sorties $y(t) \in R^p$

Définition 1 : On dit que les deux états initiaux $x_0^1, x_0^2 \in X$ sont indistinguables si $\exists t \geq 0$ $\exists u : [0, t] \rightarrow U$ une entrée admissible et notés (x_0^1, x_0^2) si les deux fonctions $y(t, x_0^1, u(t)), y(t, x_0^2, u(t)), \forall t \geq 0$ sont identiques, où $y(t, x_0^i, u(t))$ est la sortie du système pour une condition initiale x_0^i . Ils sont dits distinguables sinon.

Définition 2 : Un système est observable en $x_0^1 \in X$ si tout autre état $x_0^2 \neq x_0^1$ est distinguable de x_0^1 dans X . Un système est observable s'il est observable en tout point $x_0^1 \in X$. Le système est localement faiblement observable en $x_0^1 \in X$, s'il existe un voisinage ouvert $X'(x_0^1) \subset X$ contenant x_0^1 , tel que pour tout voisinage $X'' \subset X'(x_0^1)$ de x_0^1 , pour tout point $x_0^2 \in X''(x_0^1)$ les couples (x_0^1, x_0^2) sont distinguables et les trajectoires $y(t, x_0^1, u(t))$ et $y(t, x_0^2, u(t))$ évoluent à l'intérieur de $X''(x_0^1)$.

Intuitivement, le système est localement observable si on peut instantanément distinguer chaque état de ses voisins en choisissant judicieusement l'entrée u .

Définition 3 (Observabilité au sens du rang) : On dit que la paire (f, h) est observable au sens du rang si :

$$\text{Rang} \{dh, dL_f h, \dots, dL_f^{n-1} h\}^T = n \quad (\text{IV.2})$$

Pour ce qui est de notre cas ; il a été vérifié que le rang de la matrice d'observabilité de notre système est $r = 12$, ce qui nous ramène à confirmer que ce dernier est faiblement localement observable.

IV.3. Observateur à mode glissant

IV.3.1 Observateur classique à mode glissant

La mise en œuvre des lois de commandes basées sur le modèle non linéaire du système, nécessite la connaissance du vecteur d'état complet du système à chaque instant. Cependant, dans la plupart des cas, seule une partie de l'état est accessible à l'aide de capteurs. Ici, il s'agit de développer une certaine robustesse vis-à-vis d'incertitudes paramétriques. Une des classes les plus connues des observateurs robustes est celle des observateurs par modes glissants [Kri.96, Sir.94, Slo.87, Utk.95].

Pour le système (IV.1), une structure d'observateur par modes glissants classique s'écrit :

$$\begin{cases} \dot{\hat{x}} = f(\hat{x}, u) - K\Gamma_s \\ \hat{y} = h(\hat{x}) \end{cases} \quad (\text{IV.3})$$

Où :

K : est la matrice de gain de dimension $(n \times p)$.

Γ_s : est un vecteur de dimension $(p \times 1)$ défini tel que $\Gamma_s = [\text{sign}(y_1 - \hat{y}_1) \dots \text{sign}(y_p - \hat{y}_p)]^T$.

Nous définissons également les vecteurs relatifs aux erreurs d'observation tel que :

$e = x - \hat{x}$: est le vecteur d'état des erreurs d'observation.

$S = e_y = y - \hat{y}$: est la surface de glissement.

Le terme de correction utilisé est proportionnel à la fonction discontinue *signe* appliquée à l'erreur de sortie. L'étude de la stabilité et de la convergence pour de tels observateurs, est basée sur l'utilisation des fonctions de Lyapunov.

L'étude de la stabilité et de convergence utilise les concepts de résolutions de Filippov [Fil.60], ainsi que les méthodes dites du vecteur équivalent développées par Utkin pour la commande, et par Drakunov [Utk.92, Dra.92, Dra.95] pour les observateurs.

Notre choix de ce type d'observateurs se justifie par les bonnes propriétés qui peuvent être obtenues et parmi lesquelles nous citerons :

➤ Une convergence en temps fini vers la surface de glissement si $S = e_y = y - \hat{y} = 0$, entre autre les conditions d'attractivité vers cette surface sont vérifiées.

➤ Une réduction de dimension du système d'observation à $(n-p)$ états avec n la dimension de l'état x , et p la dimension du vecteur de sortie.

➤ Comme la fonction *signe* est équivalente à un grand gain au voisinage de l'origine, nous obtenons une certaine robustesse vis-à-vis d'erreurs de modèle et de perturbations bornées.

L'exploitation de ces propriétés, et l'utilisation d'observateurs à modes glissant à structures particulières, nous permettent d'obtenir des dynamiques d'erreurs d'observation plus simple et plus facilement stabilisables pour des formes de système plus large que la forme injection de sortie [Kre.83]. Ainsi, pour des systèmes qui se mettent sous forme d'observation triangulaire [Bou.97], de manière générale pour des systèmes quelconques, il existe différentes classes d'observateurs, nous avons choisi d'étudier des observateurs par modes glissants pour les avantages qu'ils présentent en simplicité de mise en œuvre.

IV.3.2. Observateur à mode glissant étape par étape

L'observateur à mode glissant étape par étape a été développé pour des systèmes pouvant se mettre sous la forme, appelée forme triangulaire d'observation, suivante [Bou.97] :

$$\begin{cases} \dot{x}_1 = x_2 + g_1(x_1, u) \\ \dot{x}_2 = x_3 + g_2(x_1, x_2, u) \\ \vdots \\ \dot{x}_{n-1} = x_n + g_{n-1}(x_1, x_2, \dots, x_{n-1}, u) \\ \dot{x}_n = f_n(x_1, x_2, \dots, x_n) + g_n(x_1, x_2, \dots, x_n, u) \\ y = x_1 \end{cases} \quad (\text{IV.4})$$

Où f_n et g_i , pour $i = 1, 2, \dots, n$ sont des fonctions analytiques, $x = [x_1 x_2 \dots x_n]^T \in R^n$ l'état du système, $u \in R^m$ est le vecteur d'entrée et y est la sortie. La structure de l'observateur proposé est :

$$\begin{cases} \dot{\hat{x}}_1 = \hat{x}_2 + g_1(x_1, u) + \lambda_1 \text{sign}_1(x_1 - \hat{x}_1) \\ \dot{\hat{x}}_2 = \hat{x}_3 + g_2(x_1, \bar{x}_2, u) + \lambda_2 \text{sign}_2(\bar{x}_2 - \hat{x}_2) \\ \vdots \\ \dot{\hat{x}}_{n-1} = \hat{x}_n + g_{n-1}(x_1, \bar{x}_2, \dots, \bar{x}_{n-1}, u) + \lambda_{n-1} \text{sign}_{n-1}(\bar{x}_{n-1} - \hat{x}_{n-1}) \\ \dot{\hat{x}}_n = f_n(x_1, \bar{x}_2, \dots, \bar{x}_n) + g_n(x_1, \bar{x}_2, \dots, \bar{x}_n, u) + \lambda_n \text{sign}_n(\bar{x}_n - \hat{x}_n) \\ \hat{y} = \hat{x}_1 \end{cases} \quad (\text{IV.5})$$

Où les variables \bar{x}_i sont données par :

$$\begin{cases} \bar{x}_1 = x_1 \\ \bar{x}_i = \hat{x}_i + \lambda_{i-1} \text{sign}_{\text{moy}, i-1}(\bar{x}_{i-1} - \hat{x}_{i-1}) \end{cases} \quad (\text{IV.6})$$

Avec $sign_{moy}$ désigne la fonction $sign(\cdot)$ classique filtrée par un filtre passe bas; la fonction $sign_i(\cdot)$ est mise à zéro s'il existe $j \in \{1, i-1\}$ tel que $\bar{x}_j - \hat{x}_j \neq 0$, si non la fonction $sign_i(\cdot)$ est prise égale à la fonction $sign(\cdot)$ classique. Grâce à cela nous imposons que le terme correcteur ne soit "actif" que si la condition $\bar{x}_j - \hat{x}_j = 0$ pour $j = 1, 2, \dots, i$ est vérifiée.

Pour ce type d'observateur, nous avons une convergence en temps fini (par étapes) de l'erreur d'observation. En outre, il permet la séparation de la synthèse de l'observateur et de la commande; il facilite l'étude de stabilité en boucle fermée. La convergence en temps fini de l'observateur est assurée par le théorème suivant :

Théorème 1 [Man.98] : Considérons le système (IV.4) et l'observateur (IV.5). Si le système est à entrées bornées et à états bornés (BIBS) pour une durée finie $[0, T]$, alors pour tout état initial $x(0)$, $\hat{x}(0)$ et toute entrée bornée u , il existe un choix de λ_i tel que l'état de l'observateur \hat{x} converge en un temps fini vers l'état x du système.

Cependant cette convergence se fait par étapes :

Étape 1 : Dans cette étape on assure la convergence de $e_1 = x_1 - \hat{x}_1$ vers zéro dans un temps $t < t_1$. Pour $i > 1$ toutes les fonctions $sign_i$ sont égales à zéro, les dynamiques des erreurs d'observation $e = x - \hat{x}$ sont donc :

$$\begin{cases} \dot{e}_1 = e_2 - \lambda_1 sign(x_1 - \hat{x}_1) \\ \dot{e}_2 = e_3 + g_2(x_1, x_2, u) - g_2(x_1, \hat{x}_2, u) \\ \vdots \\ \dot{e}_{n-1} = e_n + g_{n-1}(x_1, x_2, \dots, x_{n-1}, u) - g_{n-1}(x_1, \hat{x}_2, \dots, \hat{x}_{n-1}, u) \\ \dot{e}_n = f_n(x_1, x_2, \dots, x_n) - f_n(x_1, \hat{x}_2, \dots, \hat{x}_n) + g_n(x_1, x_2, \dots, x_n, u) \\ \quad \quad \quad - g_n(x_1, \hat{x}_2, \dots, \hat{x}_n, u) \end{cases} \quad (IV.7)$$

L'entrée u et les états sont bornés. Par conséquent les états du système ne divergent pas et les erreurs d'observation sont aussi bornées. On considère la fonction de Lyapunov $V_1 = \frac{e_1^2}{2}$, alors :

$$\dot{V}_1 = e_1 \dot{e}_1 = e_1 (e_2 - \lambda_1 sign(e_1)) \quad (IV.8)$$

En choisissant $\lambda_1 > |e_2|_{\max}$, l'erreur d'observation e_1 converge vers zéro en un temps fini t_1 .

Après cet instant, e_1 reste égale à zéro et on obtient alors $e_2 = \lambda_1 sign(x_1 - \hat{x}_1)$ ce qui implique que $\bar{x}_2 = x_2$.

Étape 2 : L'objectif dans cette étape est d'atteindre la surface de glissement $e_2 = x_2 - \hat{x}_2 = 0$.

Pour rester sur la surface $e_1 = 0$, il faut que $\lambda_1 > |e_2|_{\max}$, mais cela est vérifié de part le fait que e_2 est strictement décroissante après t_1 . Les dynamiques des erreurs d'observation sont alors :

$$\begin{cases} \dot{e}_1 = e_2 - \lambda_1 \text{sign}(x_1 - \hat{x}_1) = 0 \\ \dot{e}_2 = e_3 + g_2(x_1, x_2, u) - g_2(x_1, \hat{x}_2, u) - \lambda_2 \text{sign}(e_2) = e_3 - \lambda_2 \text{sign}(e_2) \\ \vdots \\ \dot{e}_{n-1} = e_n + g_{n-1}(x_1, x_2, \dots, x_{n-1}, u) - g_{n-1}(x_1, \hat{x}_2, \dots, \hat{x}_{n-1}, u) \\ \dot{e}_n = f_n(x_1, x_2, \dots, x_n) - f_n(x_1, \hat{x}_2, \dots, \hat{x}_n) + g_n(x_1, x_2, \dots, x_n, u) \\ \quad - g_n(x_1, \hat{x}_2, \dots, \hat{x}_n, u) \end{cases} \quad (\text{IV.9})$$

En choisissant la fonction de Lyapunov : $V_2 = \frac{e_1^2}{2} + \frac{e_2^2}{2}$ on aura :

$$\dot{V}_2 = e_1(e_2 - \lambda_1 \text{sign}(e_1)) + e_2(e_3 - \lambda_2 \text{sign}(e_2)) = e_2(e_3 - \lambda_2 \text{sign}(e_2)) \quad (\text{IV.10})$$

Si $\lambda_2 > |e_3|_{\max}$ alors e_2 converge vers zéro après un temps fini $t_2 > t_1$. De plus $\dot{V}_2 < 0$, et l'erreur d'observation est strictement décroissante durant la période $[t_1, t_2]$ ce qui implique que la condition imposée dans la première étape sur λ_1 doit être vérifiée aussi après t_1 . En fin, après un temps fini t_2 $\bar{x}_3 = x_3$. Ainsi étape par étape nous obtenons la convergence des composantes de l'erreur d'observation vers zéro et celles de \hat{x} vers x , sous les conditions que $\lambda_i > |e_{i+1}|_{\max}$ durant $[t_i, t_{i+1}]$.

Étape n : cette étape commence à l'instant t_{n-1} , à cet instant $e_k = 0$ pour tous $k < n$ Alors, la dynamique de l'erreur d'observation devient :

$$\begin{cases} \dot{e}_1 = e_2 - \lambda_1 \text{sign}(x_1 - \hat{x}_1) = 0 \\ \dot{e}_2 = e_3 - \lambda_2 \text{sign}(e_2) = 0 \\ \vdots \\ \dot{e}_{n-1} = e_n - \lambda_{n-1} \text{sign}(e_{n-1}) = 0 \\ \dot{e}_n = f_n(x_1, x_2, \dots, x_n) - f_n(x_1, \hat{x}_2, \dots, \hat{x}_n) + g_n(x_1, x_2, \dots, x_n, u) \\ \quad - g_n(x_1, \hat{x}_2, \dots, \hat{x}_n, u) - \lambda_n \text{sign}(\bar{x}_n - \hat{x}_n) = -\lambda_n \text{sign}(e_n) \end{cases} \quad (\text{IV.11})$$

De la même façon on choisit la fonction de Lyapunov $V_n = \frac{e_1^2}{2} + \frac{e_2^2}{2} + \dots + \frac{e_n^2}{2}$. On obtient donc

$\dot{V}_n = e_n(-\lambda_n \text{sign}(e_n))$. Ainsi, e_n converge vers zéro en un temps fini $t_n > t_{n-1}$ pour toutes

valeurs de $\lambda_n > 0$, si évidemment toutes les conditions sur $\lambda_k, k < n$ sont elles aussi, vérifiées.

Ces conditions resteront satisfaites après t_{n-1} . Ainsi le théorème est vérifié.

IV.4. Synthèse de l'observateur

Considérons le modèle du système (II.35), et soit le vecteur $\hat{\underline{X}}$ l'estimé du vecteur \underline{X} , le modèle de l'observateur est une copie du modèle d'origine (celui du système) et qui se présente comme suit :

$$\hat{\underline{X}} = [\hat{x}_1 \quad \hat{x}_2 \quad \hat{x}_3 \quad \hat{x}_4 \quad \hat{x}_5 \quad \hat{x}_6 \quad \hat{x}_7 \quad \hat{x}_8 \quad \hat{x}_9 \quad \hat{x}_{10} \quad \hat{x}_{11} \quad \hat{x}_{12}]^T \quad (\text{IV.12})$$

Le vecteur de sortie est défini par :

$$\underline{Y} = h(\underline{X}) = [x_1, x_3, x_5, x_7, x_9, x_{11}]^T$$

$$\left\{ \begin{array}{l} \dot{\hat{x}}_1 = \hat{x}_2 + \lambda_1 \text{sign}_1(x_1 - \hat{x}_1) \\ \dot{\hat{x}}_2 = \frac{(I_y - I_z)}{I_x} \bar{x}_4 \bar{x}_6 - \frac{J_r}{I_x} \Omega_r \bar{x}_4 - \frac{K_{fax}}{I_x} \bar{x}_2^2 + \frac{1}{I_x} u_2 + \lambda_2 \text{sign}_2(\bar{x}_2 - \hat{x}_2) \\ \dot{\hat{x}}_3 = \hat{x}_4 + \lambda_3 \text{sign}_3(x_3 - \hat{x}_3) \\ \dot{\hat{x}}_4 = \frac{(I_z - I_x)}{I_y} \bar{x}_2 \bar{x}_6 + \frac{J_r}{I_y} \Omega_r \bar{x}_2 - \frac{K_{fay}}{I_y} \bar{x}_4^2 + \frac{1}{I_y} u_3 + \lambda_4 \text{sign}_4(\bar{x}_4 - \hat{x}_4) \\ \dot{\hat{x}}_5 = \hat{x}_6 + \lambda_5 \text{sign}_5(x_5 - \hat{x}_5) \\ \dot{\hat{x}}_6 = \frac{(I_x - I_y)}{I_z} \bar{x}_2 \bar{x}_4 - \frac{K_{faz}}{I_z} \bar{x}_6^2 + \frac{1}{I_z} u_4 + \lambda_6 \text{sign}_6(\bar{x}_6 - \hat{x}_6) \\ \dot{\hat{x}}_7 = \hat{x}_8 + \lambda_7 \text{sign}_7(x_7 - \hat{x}_7) \\ \dot{\hat{x}}_8 = -\frac{K_{fix}}{m} \bar{x}_8 + \frac{1}{m} u_x u_1 + \lambda_8 \text{sign}_8(\bar{x}_8 - \hat{x}_8) \\ \dot{\hat{x}}_9 = \hat{x}_{10} + \lambda_9 \text{sign}_9(x_9 - \hat{x}_9) \\ \dot{\hat{x}}_{10} = -\frac{K_{fxy}}{m} \bar{x}_{10} + \frac{1}{m} u_y u_1 + \lambda_{10} \text{sign}_{10}(\bar{x}_{10} - \hat{x}_{10}) \\ \dot{\hat{x}}_{11} = \hat{x}_{12} + \lambda_{11} \text{sign}_{11}(x_{11} - \hat{x}_{11}) \\ \dot{\hat{x}}_{12} = -\frac{K_{fyz}}{m} \bar{x}_{12} - g + \frac{1}{m} u_z u_1 + \lambda_{12} \text{sign}_{12}(\bar{x}_{12} - \hat{x}_{12}) \end{array} \right. \quad (\text{IV.13})$$

$$\text{Où : } \left\{ \begin{array}{l} \bar{x}_1 = x_1 \\ \bar{x}_2 = \hat{x}_2 + \lambda_1 \text{sign}_{moy,1}(\bar{x}_1 - \hat{x}_1) \\ \bar{x}_3 = x_3 \\ \bar{x}_4 = \hat{x}_4 + \lambda_3 \text{sign}_{moy,3}(\bar{x}_3 - \hat{x}_3) \\ \bar{x}_5 = x_5 \\ \bar{x}_6 = \hat{x}_6 + \lambda_5 \text{sign}_{moy,5}(\bar{x}_5 - \hat{x}_5) \\ \bar{x}_7 = x_7 \\ \bar{x}_8 = \hat{x}_8 + \lambda_7 \text{sign}_{moy,7}(\bar{x}_7 - \hat{x}_7) \\ \bar{x}_9 = x_9 \\ \bar{x}_{10} = \hat{x}_{10} + \lambda_9 \text{sign}_{moy,9}(\bar{x}_9 - \hat{x}_9) \\ \bar{x}_{11} = x_{11} \\ \bar{x}_{12} = \hat{x}_{12} + \lambda_{11} \text{sign}_{moy,11}(\bar{x}_{11} - \hat{x}_{11}) \end{array} \right. \quad (\text{IV.14})$$

La démonstration de la convergence de l'observateur par mode glissant sous forme triangulaire en utilisant la stabilité de la fonction de Lyapunov, s'effectue en 2 étapes :

Étape 1 :

Pour la convergence de e_1 (respectivement $e_3, e_5, e_7, e_9, e_{11}$) en un temps fini, on considère la

fonction de Lyapunov : $V_1 = \frac{e_1^2}{2}$ (respectivement $V_3, V_5, V_7, V_9, V_{11}$)

$$\dot{V}_1 = e_1 \dot{e}_1 = e_1 (e_2 - \lambda_1 \text{sign}(e_1)) \quad (\text{IV.15})$$

Par le choix $\lambda_1 > |e_2|_{\max}$ nous obtenons la convergence de e_1 vers zéro après le temps t_1 l'état atteint la surface de glissement et sur cette surface nous avons :

$$e_1 = \dot{e}_1 = 0 \quad (\text{IV.16})$$

alors on a : $\bar{x}_2 = x_2$. Notons que la même procédure de démonstration est utilisée pour montrer respectivement la convergence en temps fini t_1 de $e_3, e_5, e_7, e_9, e_{11}$.

Étape 2 : Dans cette étape on assure la convergence de e_2 (respectivement pour $e_4, e_6, e_8, e_{10}, e_{12}$) tout en respectant les conditions de convergence de la première étape.

En remplaçant \bar{x}_2 par x_2 (respectivement \bar{x}_4 par x_4, \bar{x}_6 par x_6, \bar{x}_8 par x_8, \bar{x}_{10} par x_{10}, \bar{x}_{12} par x_{12}) et en considérant la fonction de Lyapunov : $V_2 = \frac{e_1^2}{2} + \frac{e_2^2}{2}$ (resp. $V_4, V_6, V_8, V_{10}, V_{12}$) on aura :

$$\dot{V}_2 = e_1 \dot{e}_1 + e_2 \dot{e}_2 = e_2 (-\lambda_2 \text{sign}(e_2)) \quad (\text{IV.17})$$

Il suffit donc de choisir des petites valeurs pour λ_2 pour garantir la convergence de toutes les erreurs d'estimation après un temps $t_2 > t_1$. (resp. $\dot{V}_4, \dot{V}_6, \dot{V}_8, \dot{V}_{10}, \dot{V}_{12}$)

Notons que la même procédure de démonstration est utilisée pour montrer respectivement la convergence ($e_4, e_6, e_8, e_{10}, e_{12}$)

IV.4.1. Résultats de simulation

Les simulations présentées ci-dessous représentent la combinaison de l'observateur avec la commande par intégrale backstepping. L'objectif de la commande est de suivre la trajectoire désirée.

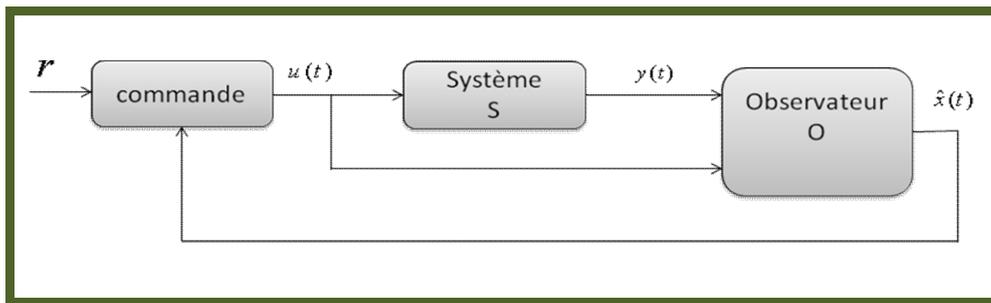


Figure IV.2. Commande avec Observateur.

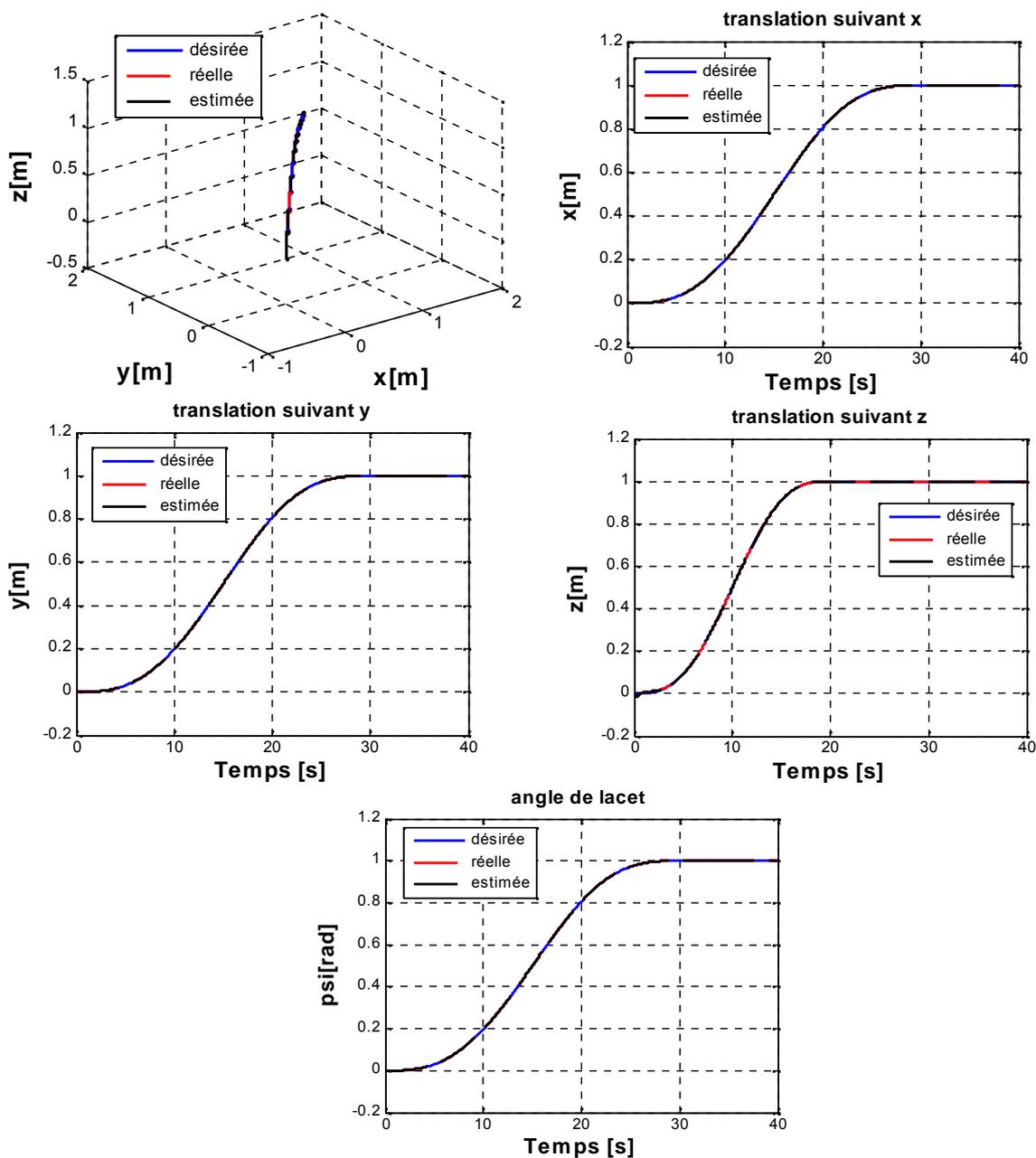


Figure IV.3. Résultats de poursuite des trajectoires désirées selon le lacet ψ et les axes (x, y, z) .

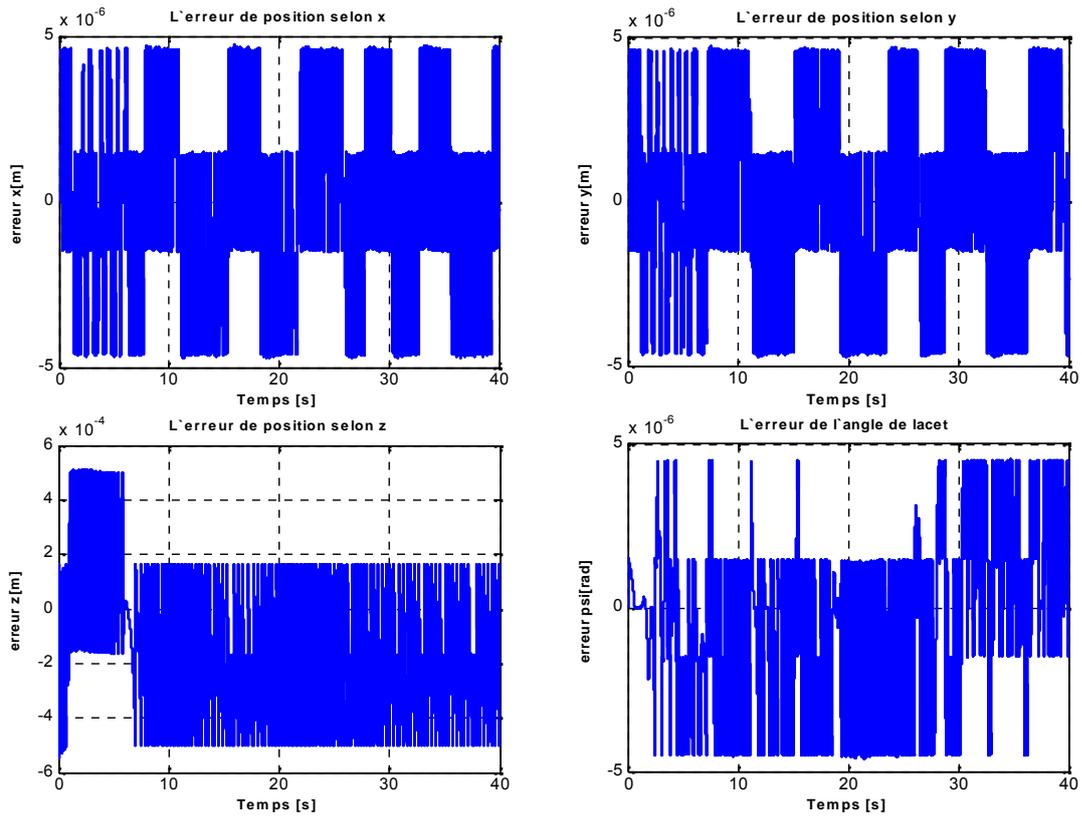


Figure IV.4. Erreurs entre les états réels et leurs estimés.

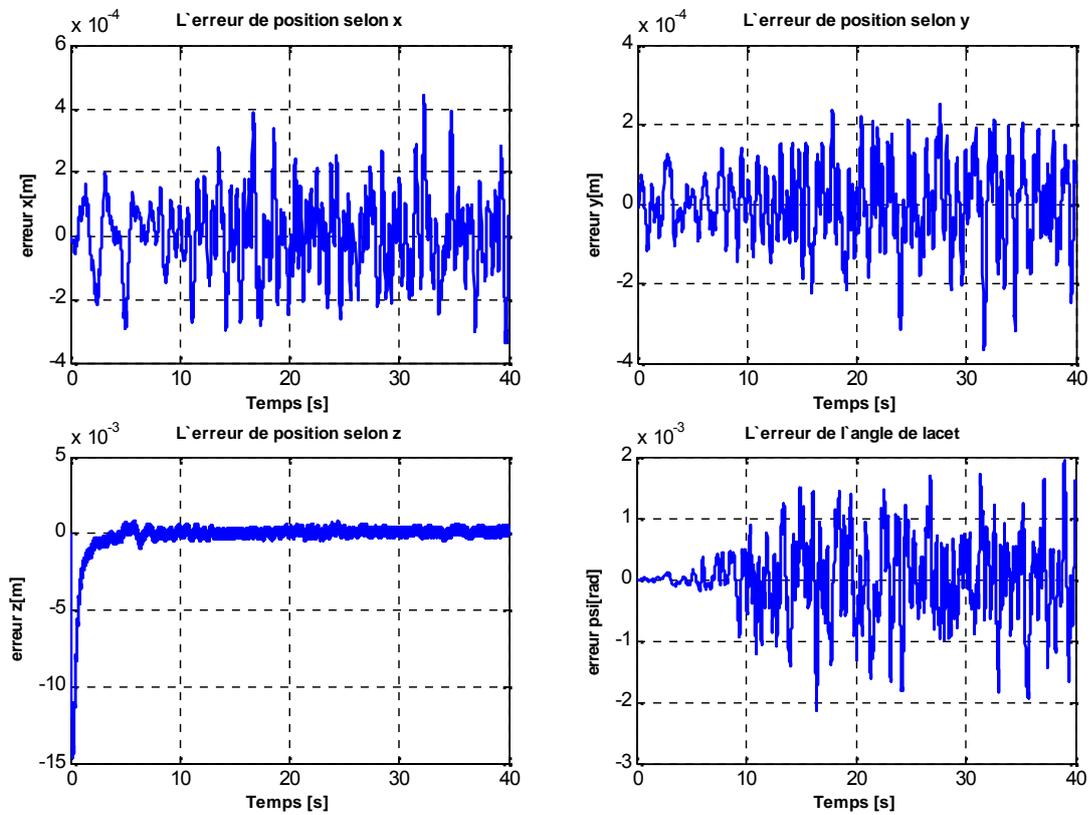


Figure IV.5. Erreurs entre les états estimés et les états désirés.

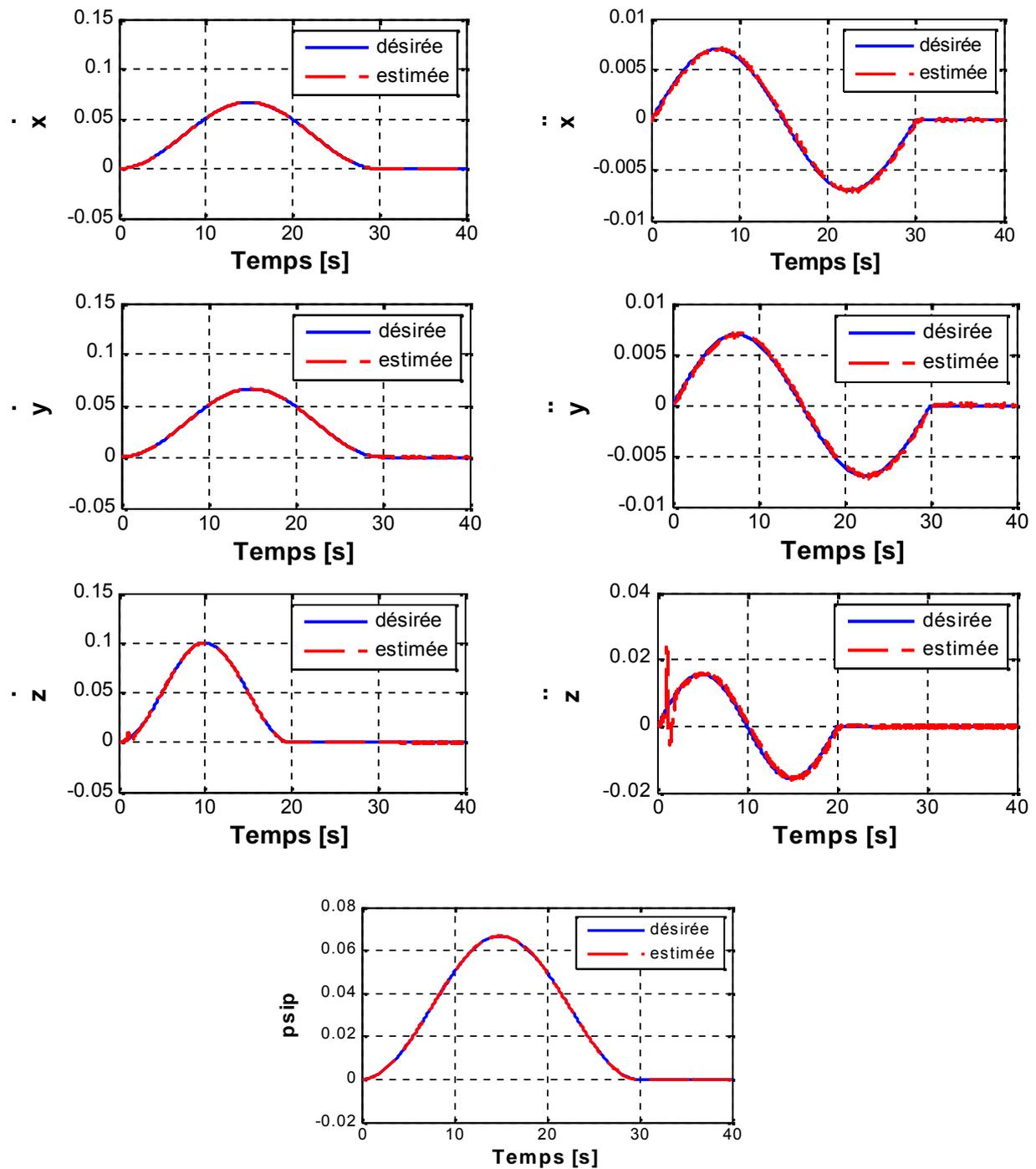


Figure IV.6. Résultats de poursuite des trajectoires désirées et estimées pour les variables non mesurées par les capteurs $(\dot{x}, \dot{y}, \dot{z}, \dot{\psi})$ et $(\ddot{x}, \ddot{y}, \ddot{z})$.

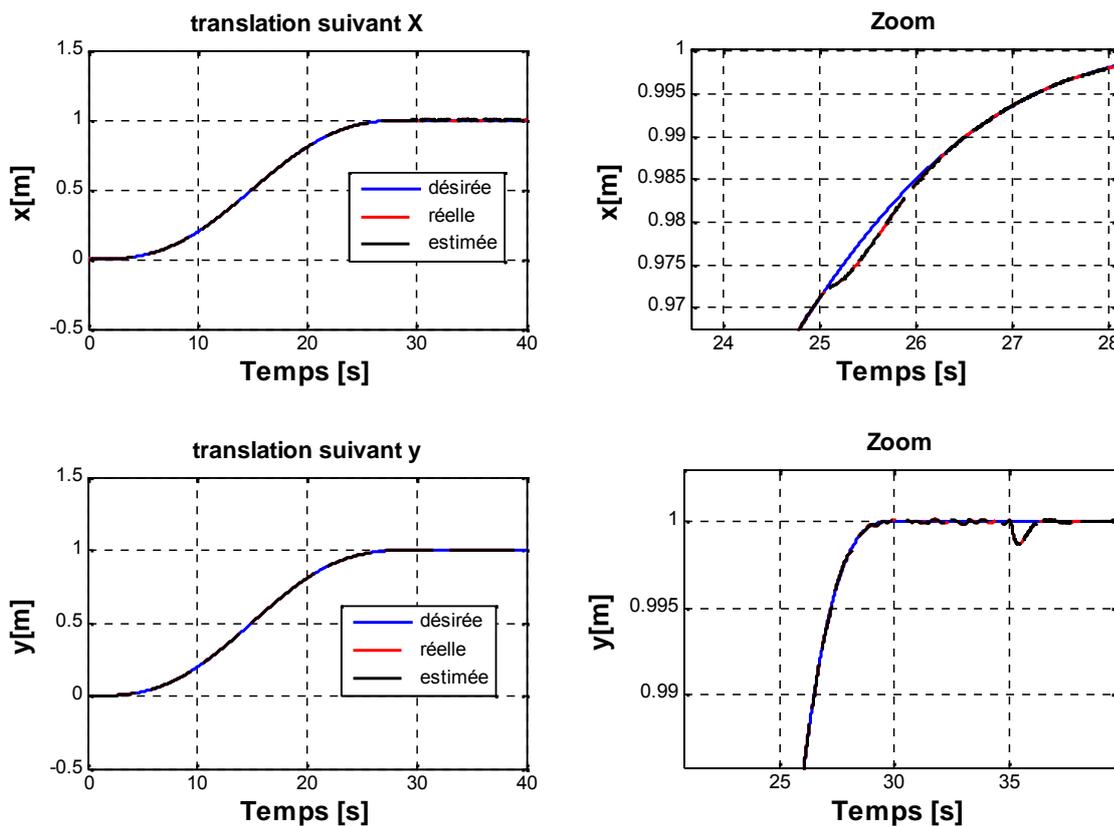
La simulation présentée concerne les trois axes (x, y, z) et le lacet ψ du quadrirotor, permettant ainsi de valider le schéma de commande et d'observation proposé dans ce chapitre. Le comportement de la commande par intégrale Backstepping et de l'observateur par modes glissants a été testé en simulation pour des trajectoires cycloïdale (figure IV.3). Les résultats

des figures IV.4, IV.5 montrent que le système complet, commande et observateur, est stable avec un bon suivi de trajectoire et de petites erreurs malgré le phénomène de réticence. La figure IV.6 montre l'efficacité de notre observateur à obtenir des signaux non bruités des états estimés.

IV.4.2. Etude de la robustesse

Afin de tester la robustesse de notre commande avec observateur nous avons effectués des tests suivants, On applique des forces perturbatrices : $d_x = 2\sin(0.1t)$; $d_y = 1.5\sin(0.1t)$; $d_z = 3\sin(0.1t)$ respectivement à l'instant $t = 25\text{sec}$, $t = 35\text{sec}$ et $t = 20\text{sec}$ avec des incertitudes paramétriques +20% de (I_x, I_y, I_z) .

D'après les figures IV.7, IV.8 on constate que notre système manifeste un comportement stable, c'est à dire qu'il converge vers les positions désirées malgré l'introduction des perturbations et des variations paramétriques de l'inertie sur les trois axes (x, y, z) .



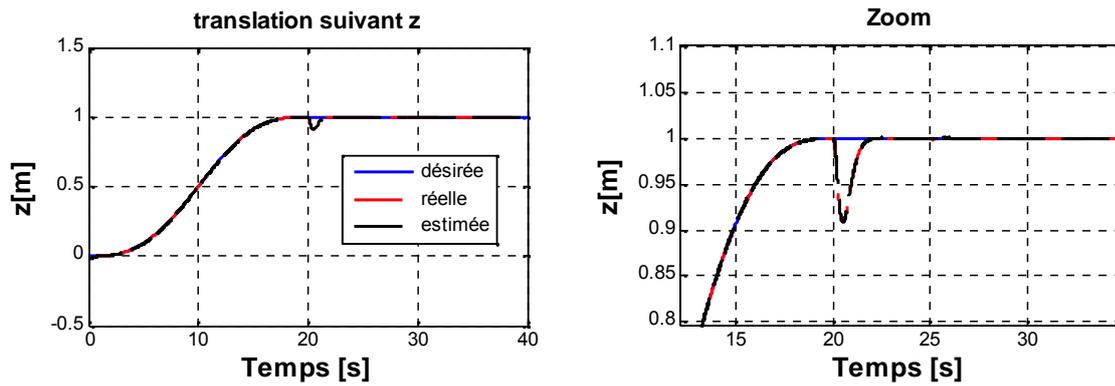


Figure IV.7. L'influence de la perturbation selon (x, y, z) avec incertitudes paramétriques $+20\%$ de (I_x, I_y, I_z) .

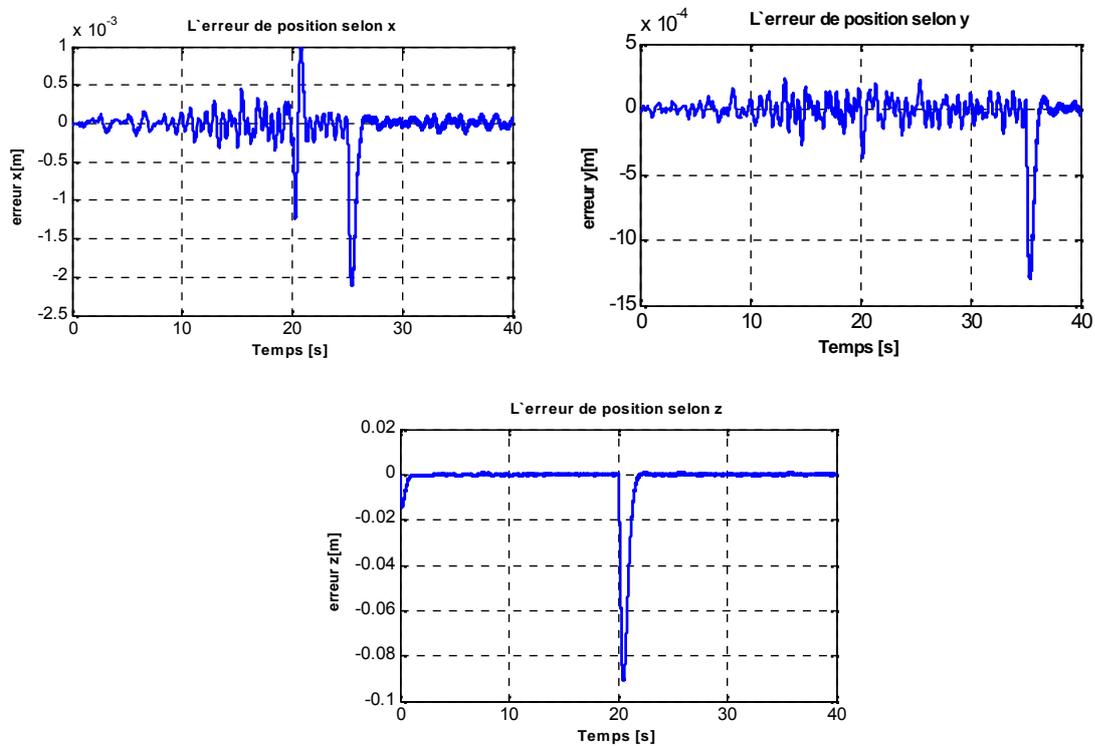


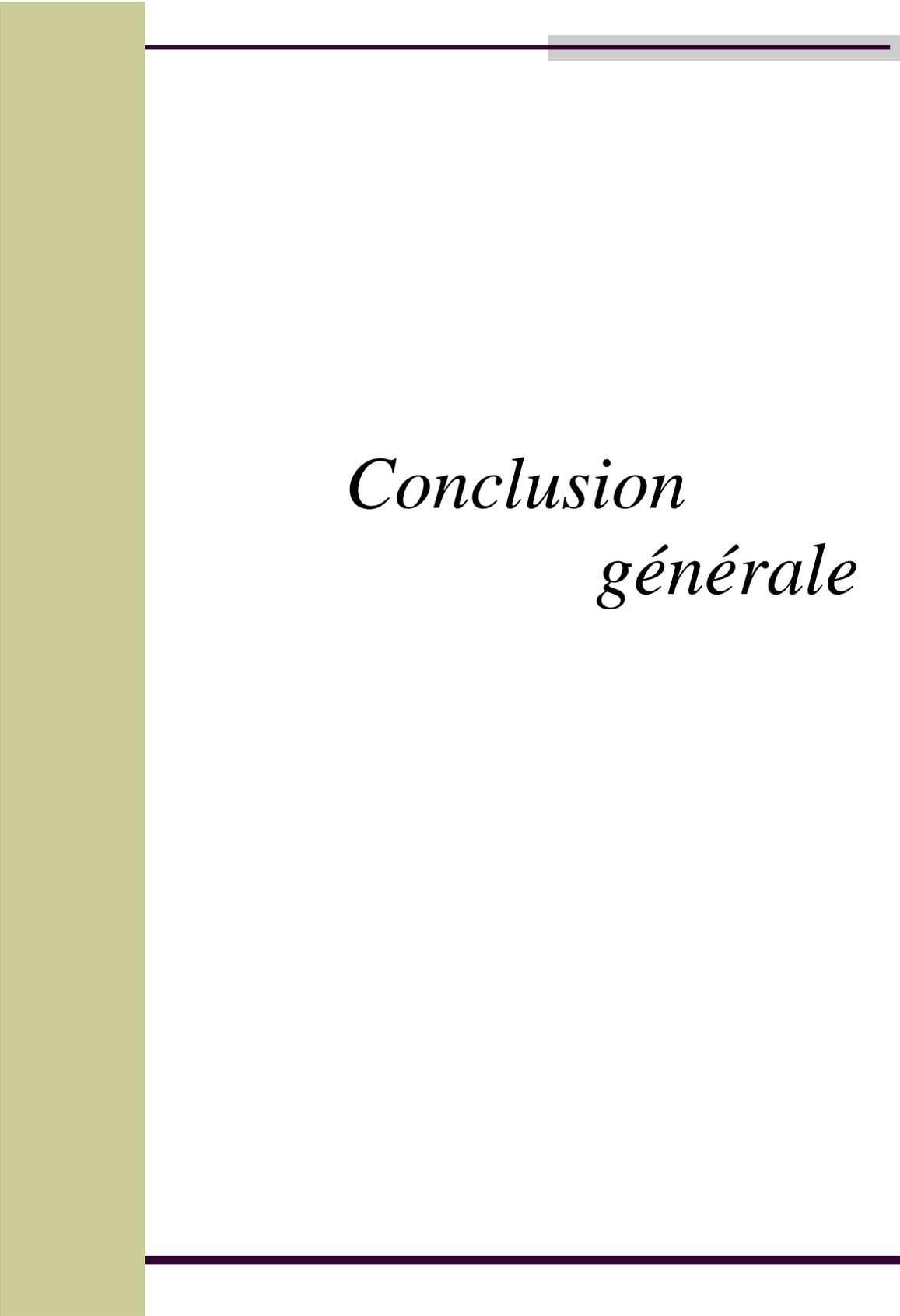
Figure IV.8. Les erreurs entre les états estimés et les états désirés selon (x, y, z) avec incertitudes paramétriques $+20\%$ de (I_x, I_y, I_z) .

IV.5.Conclusion

Dans ce chapitre nous avons donné la théorie nécessaire pour la synthèse d'un observateur par mode glissant, et nous avons combiné la commande intégrale Backstepping avec cet observateur. Les résultats de simulation ont permis d'évaluer les performances de cet observateur.

De façon générale la connaissance entière ou partielle de l'état d'un système est une exigence importante qui intervient dans les domaines de la commande, du diagnostic et de la surveillance des systèmes. Sur le plan pratique, cette exigence s'avère difficile à satisfaire directement dans la plupart des cas. Ceci est dû d'une part, au fait que les variables d'état n'ont pas toujours une signification physique et leur mesure directe est impossible à réaliser. D'autre part, lorsqu'une variable d'état existe physiquement, sa mesure peut être délicate à effectuer d'un point de vue technique (capteur nécessaire indisponible ou de précision insuffisante ...).

De plus d'un point de vue économique, il est souvent souhaitable d'installer un minimum de capteurs afin de réduire les coûts d'instrumentation et de maintenance. Par conséquent, dès qu'une stratégie de commande, de diagnostic ou de surveillance demande l'utilisation de variables d'état non mesurées, il est indispensable, de reconstruire entièrement ou partiellement, le vecteur d'état du système. Ce problème peut être résolu en utilisant un système dynamique auxiliaire, appelé observateur d'état dont le rôle est de fournir en temps réel une estimation du vecteur d'état du système étudié en fonction des entrées connues, des sorties et du modèle dynamique de celui-ci.



*Conclusion
générale*

Conclusion générale

Pendant la réalisation de ce travail, nous avons acquis une expérience très enrichissante dans le plan théorique de la commande de quadrirotor, l'intelligence artificielle et les techniques de commandes avancées. Notre travail nous a permis d'une part de développer plusieurs techniques de commandes avancées, et d'autre de proposer des méthodes d'hybridation entre ces techniques de commande.

Le quadrirotor est un système complexe non linéaire, multi variables, instable notamment et présente une dynamique fortement couplée. Le problème traité consiste à garantir en premier lieu la stabilité de ce dernier ainsi que la poursuite de trajectoire avec plus au moins des performances acceptables vis-à-vis le milieu de navigation.

Après avoir présenté la modélisation de quadrirotor nous avons synthétisé différentes lois de commande en utilisant plusieurs techniques. Notre première démarche a été l'étude de la commande à structure variable (par mode glissant) qui est une loi facile à mettre en œuvre. Cependant le signal de commande obtenu par un « SMC », présente des variations brusques dues au phénomène de broutement. Afin de réduire les effets du phénomène de broutement et d'améliorer davantage les performances de contrôle de quadrirotor, une hybridation entre la logique floue et le mode de glissement a été proposé. Cette hybridation a donné naissance « FSMC », celui-ci présente la même structure de commande « SMC », mis à part le deuxième terme « u_n », qui sera remplacé par un contrôleur flou. On a proposé aussi, un nouveau contrôleur PI basé sur la théorie des modes de glissement « PIFSMC-GA » qui a donné de meilleurs résultats contrairement au « FSMC » vis-à-vis aux rejets des perturbations.

Nous avons utilisé l'algorithme génétique pour la détermination des gains optimaux de contrôleur « PI », afin d'avoir des performances meilleures qu'avec celles obtenus avec des gains non optimisés.

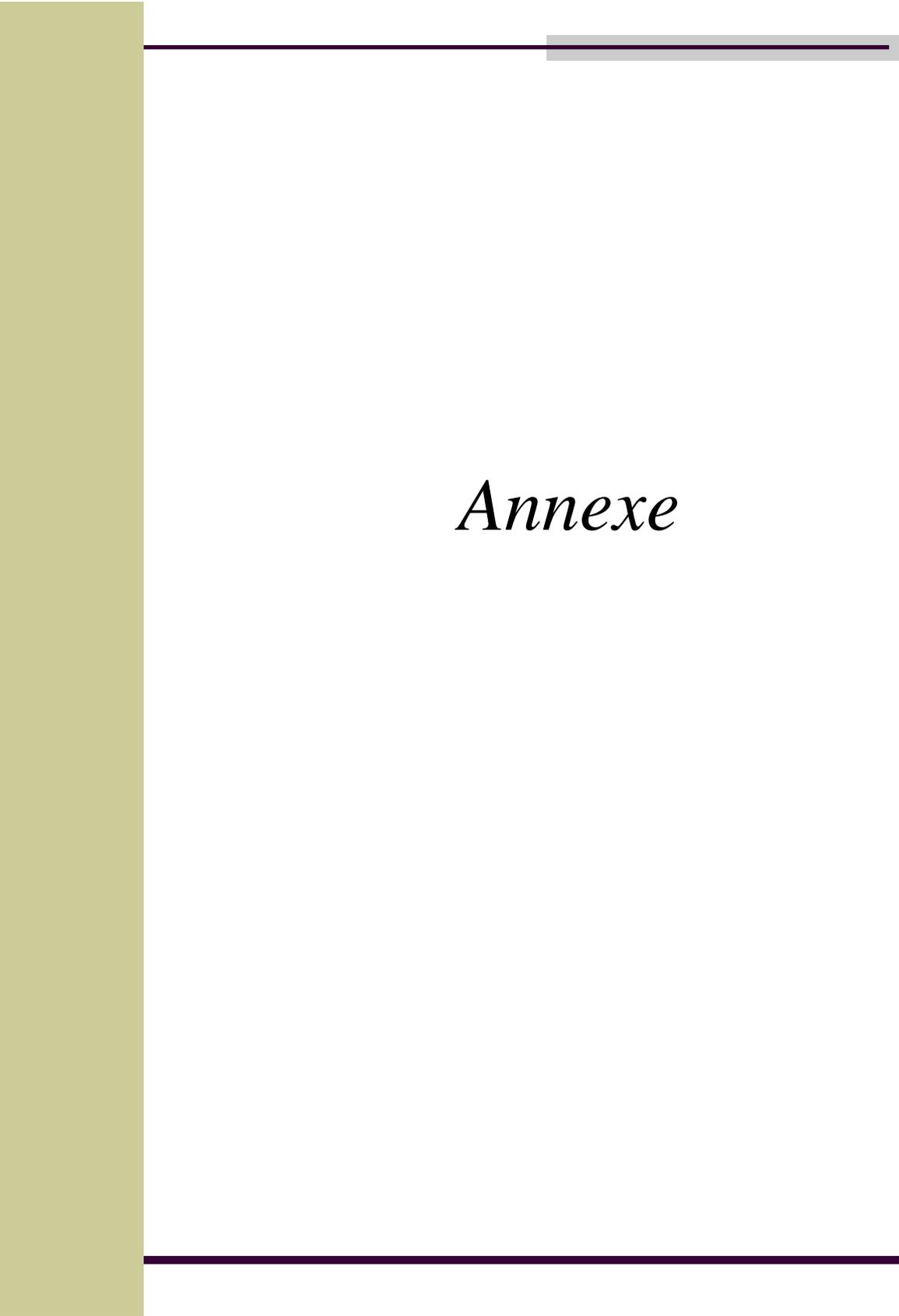
Par la suite, nous avons présenté la technique de commande par backstepping et intégrale backstepping cette approche d'hybridation entre le backstepping et le PID a été pour but d'augmenter davantage les performances de contrôleur. Dans la majorité des cas nous avons réussi à assurer en général la stabilité au sens de Lyapunov, on en déduit que quelques contrôleurs présentent de meilleures performances que d'autres.

En réalité l'idée de base de la conception des UAV était la navigation dans des milieux hostiles à l'homme tels les zones de conflits, d'industrie militaire ou de fortes turbulences

atmosphérique , ceci peut rendre la navigation très délicate voir impossible dans ces milieux si contraints et si évolutifs, d'où la nécessité de synthétiser des observateurs qui peuvent reconstruire le vecteur d'état d'une façon partielle ou d'une façon complète et qui peuvent aussi parer et remédier aux problèmes des erreurs de modélisation et ceux dus aux états non mesurables, ceci a fait l'objet du dernier chapitre.

A l'issue de ces travaux, ce mémoire ouvre de nouvelles perspectives de recherche parmi les quelles nous citons :

- Mise en oeuvre expérimentale des lois de commande développées sur un système réel.
- Utilisation d'autres algorithmes d'optimisations pour la détermination des différents paramètres de la loi de commande.
- La reformulation des méthodes développées dans un contexte adaptatif.
- Validation des commandes proposées sur d'autres systèmes, tels que le robot manipulateur, la machine asynchrone ...etc.



Annexe

ANNEXE A

Tableau. A.1 : les paramètres de quadrirotor.

Symbole	description	valeurs et unité
m	Masse du quadri-rotors	0.486 kg
l	Distance entre le moteur et C.G	0.255 m
I_x	Moment d'inertie suivant l'axe X	$3.8278 \times 10^{-3} \text{ kg} \cdot \text{m}^2$
I_y	Moment d'inertie suivant l'axe Y	$3.8278 \times 10^{-3} \text{ kg} \cdot \text{m}^2$
I_z	Moment d'inertie suivant l'axe Z	$7.6566 \times 10^{-3} \text{ kg} \cdot \text{m}^2$
b	Le coefficient de portance	$2.9842 \times 10^{-5} \text{ N/rad/s}$
d	Le coefficient de drag	$3.2320 \times 10^{-6} \text{ N/rad/s}$
J_r	L'inertie du rotor	$2.8385 \times 10^{-5} \text{ kg} \cdot \text{m}^2$
g	La gravité	9.806 m/s^2
k_e	Constante de couple électrique	0.0216
C_s	Frottement sec	5.386×10^{-3}
k_r	Constante de couple de charge	3.4629×10^{-7}
k_m	Constante couple mécanique	4.333×10^{-3}
r	Résistance moteur	0.67 Ω

A.2. Représentation des forces (Zoom) :

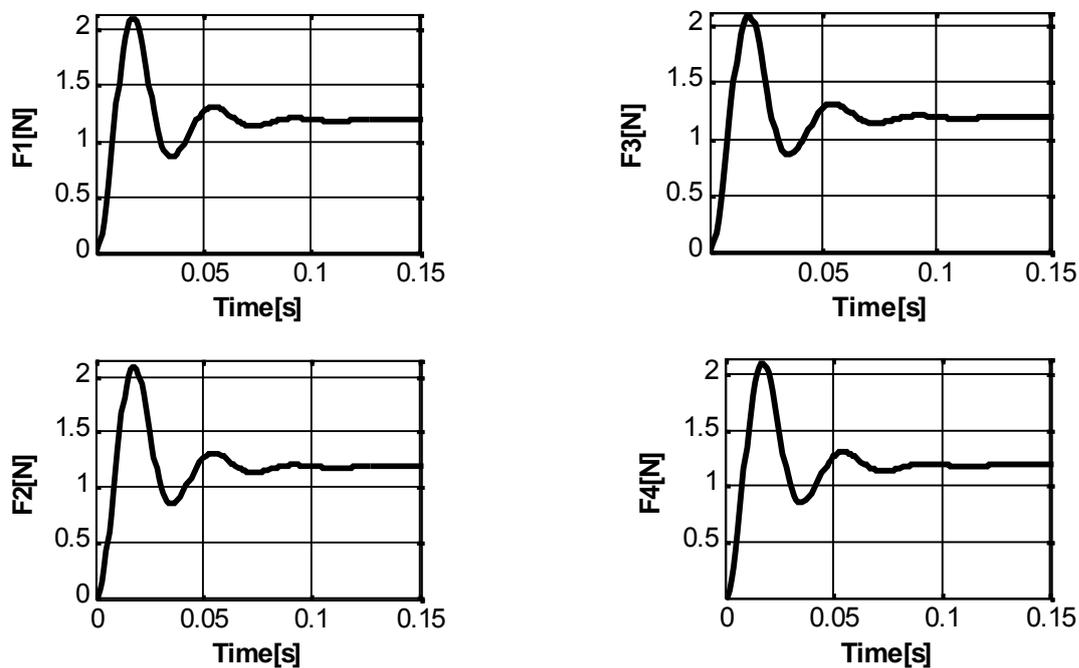
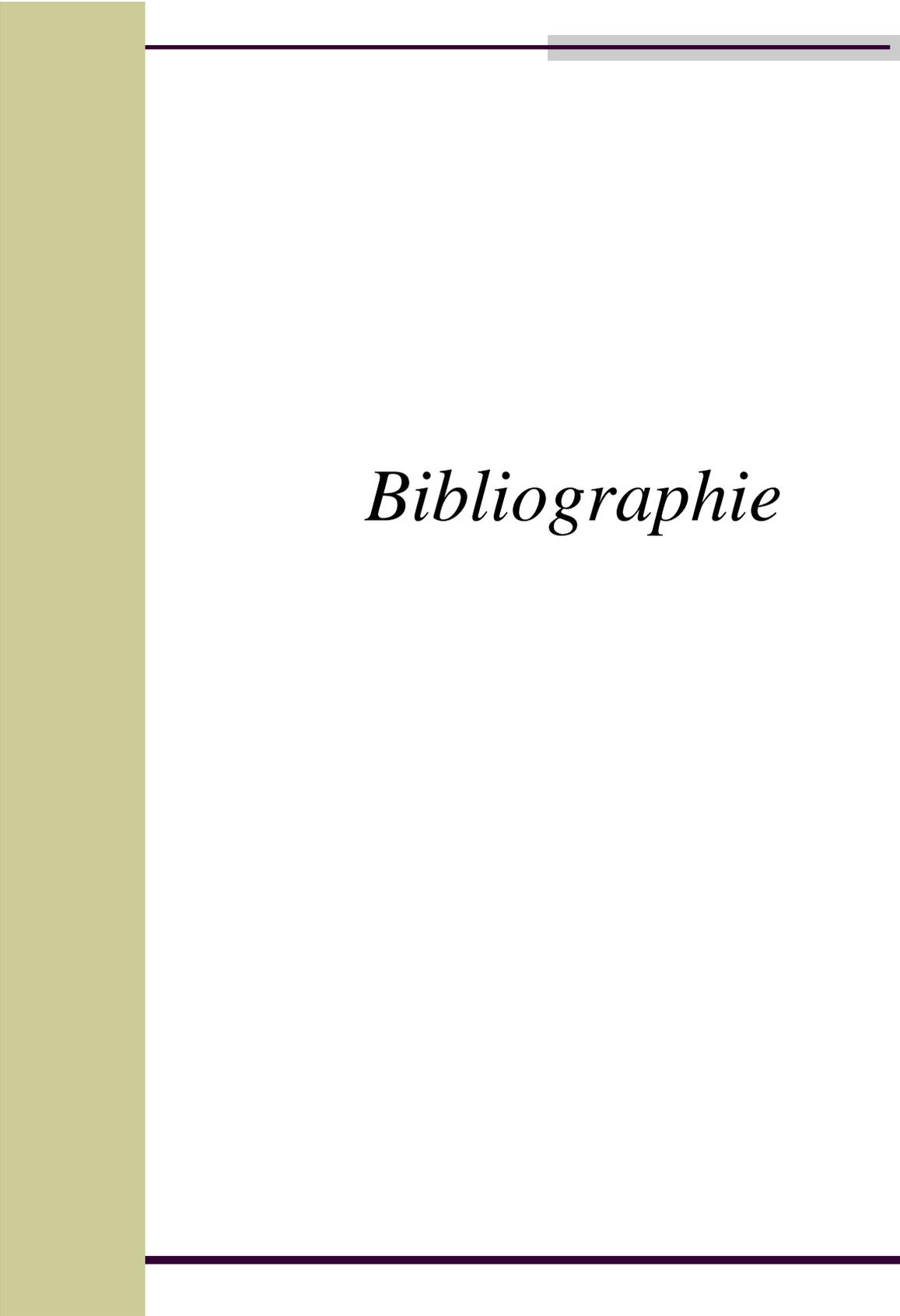


Fig. A.1. Les forces (F_1 F_2 F_3 F_4)

A decorative vertical bar in a muted olive green color is positioned on the left side of the page. Two horizontal lines, one dark purple and one grey, extend from the right edge of this bar across the top and bottom of the page respectively.

Bibliographie

- [Aev.79] Aeyels, D. « **Local and global observability of non linear differential equations** ». Delft, The Netherlands, MTNL (1979).
- [Alt.03] Erdinç Altu , « **Vision based control of unmanned aerial vehicles with applications to an autonomous four rotor helicopter, quadrotor** », A dissertation in Mechanical Engineering and applied Mechanics, presented to the faculties of the University of Pennsylvania, in partial fulfilment of the requirements for the degree of Doctor of Philosophy,(2003).
- [A.SI.08] Rami Abou Sleiman, Dan Korff, Ermal Gjioni, and Hong Chul Yang , « **The Oakland University Unmanned Aerial Quadrotor System** », for the AUVSI UAS Student Competition (2008).
- [Avi.08] All the world's Rotorcraft, www.aviastar.org.
- [Bel.12.a] Z.Bellahcene, M.Bouhamida, A.Mokhtari, « **PI Fuzzy Sliding Mode Control with Genetic Algorithm Optimization Applied to Unmanned Aerial Quadrotor Vehicle** »,PRZEGL D ELEKTROTECHNICZNY (Electrical Review), publication en cours (2012).
- [Bel.12.b] Z.Bellahcene, M.Bouhamida, M. Benghanem, A. Laidani ,« **La Commande Intégrale Backstepping Appliquée à un Hélicoptère à quatre hélices** », 2^{ème} conférence internationale sur la maintenance, la gestion, la logistique et l'électrotechnique, ENSET Oran, 19-21 Novembre (2012).
- [Bou.97] T. Boukhobza. « **Observateurs à modes glissants et formes d'observabilité et analyse et synthèse des commandes par ordres supérieurs** ». PhD thesis, Université de Paris-Sud Orsay, Laboratoire des Signaux et Systèmes, (1997).
- [Bag.99] L. Baghli, « **Contribution à la commande de la machine asynchrone, utilisation de la logique floue, des réseaux de neurones et des algorithmes génétiques** », Thèse de Doctorat de l'université Henri Poincaré, janvier (1999).
- [Ben.00] A.R. Benaskeur, «**Aspects de l'application du backstepping adaptatif à la commande décentralisée des systèmes non linéaires** », Thèse doctorat, Université Laval, Québec, Février (2000).
- [Bou.04] S. Bouabdallah, A. Noth and R. Siegwart, «**PID vs LQ Control Techniques Applied to an Indoor Micro Quadrotor** », Autonomous Systems Laboratory Swiss Federal Institute of Technology Lausanne, Switzerland, (2004).
- [Bou.05] Samir Bouabdallah and Roland Siegwart «**Backstepping and Sliding mode Techniques Applied to an Indoor Micro Quadrotor**» IEEE international conference on robotics and automation ,Barcelona,spain,april,(2005).
- [Bou.07] Samir Bouabdallah, « **Design and control of quadrotors with to application autonomous flying** », thèse de Doctorat, Ecole Polytechnique Fédérale de Lausanne, EPFL (2007).

- [Bel.07] A. Belhani « **Conception des Contrôleurs Flous Multivariables** » Présentée pour l'obtention de diplôme de doctorat en science « UNIVERSITE MENTOURI-CONSTANTINE ». 09-juillet (2007).
- [Bre.08] Tommaso Bresciani, « **Modelling, Identification and Control of a Quadrotor Helicopter**», Master Thesis, Department of Automatic Control Lund University, October (2008).
- [Cor.95] O. Cordon and F. Herrera, « **A General Study on Genetic Fuzzy Systems** », Editor J. Periaux and G. Winter, M. Galan, P. Custa (Eds.), Genetic algorithms in engineering and computer science, John Wiley and Sons, (1995).
- [Dec.88] R.A. Decarlo, S.H. Zak, G.P. Matthews, « **Variable structure control of nonlinear multivariable systems**», Proc. IEEE. CODEN IEEPAD, Vol.76, pp.212-232, (1988).
- [Der.06] Derafa L. Madani t. and Benallegue A, « **dynamic modelling and experimental identification of four rotor helicopter parameters** » IEEE-ICIT Mumbai, India, (2006).
- [Dzu.04] A. Dzul P. Castillo and R. Lozano « **Real-time stabilization and tracking of a four-rotor mini rotorcraft** », IEEE Transaction on Control System Technology, 12(4):510 – 516, July (2004).
- [Dun.04] J. Dunfield, M. Tarbouchi, G. Labonte, « **Neural Network Based control of a Four Rotor Helicopter** », IEEE International Conference on Industrial Technology (ICIT) 2004.
- [Dra.95] S. Drakunov and V. Utkin. « **Sliding mode observer** ». Tutorial. IEEE Conf. on Dec. and Cont., pages 3376.3379, (1995).
- [Dra.92] S.V. Drakunov. « **Sliding mode observer based on equivalent control method** ». In IEEE Conf. on Dec. and Cont., pages 2368.2369, (1992).
- [Esh.05] F. Eshbair, « **Modélisation et commande d'un système multi-moteur par la technique de commande backstepping** », Thèse doctorat, université du Québec à Trois-Rivières, Août (2005).
- [Eme.67] S. V. Emelyanov, “**Variable structure control systems**”, Moscow. Nauka, 1967.
- [Far.06] F. Abdulgalil « **Commande non linéaire dans les systèmes de forage pétrolier : contribution a la suppression du phénomène de « STICK-SLIP »** » Université PARIS XI UFR SCIENTIFIQUE D'ORSAY, Thèse de Doctorat Février (2006).
- [Fil.60] A. F.Fillipov. « **Differential equations with discontinuous right-hand side** ». Amer.Math. Soc.Trans. 62. p:199-231, (1960).
- [gol.94] D. E. Goldberg, « **Genetic algorithms, Exploration, Optimization and Machine Learning** », Addison-Wesley Edition, France, SA (1994).

- [Gou.04] R. Gouri, « **Commande par mode flou glissant et par backstepping d'une machine asynchrone** », Mémoire de Magister, CU Béchar (2004).
- [Gue.04] Guenard N. Hamel t. Moreau V « **modélisation et élaboration de commande de stabilisation de vitesse et de correction d'assiette pour un drone** » CIFA. ,(2004).
- [Hun.93] Hung, J.Y., Gao,W., Hung, J.C. « **Variable structure control** » A survey. IEEE Transaction on Ind. Electronics 40(1), 2–22 (1993).
- [Han.05] Scott D. Hanford, « **A small semi-autonomous rotary-wing unmanned air vehicle** », The Pennsylvania State University The Graduate School, A Thesis in Aerospace Engineering, (2005).
- [Her.77] Hermann, R. and Krener A.J. « **Nonlinear controllability and observability** ». IEEE Trans. Automat. Contr. 5:728-740, (1977).
- [Jor.09] Jorge Miguel Brito Domingues, « **Quadrotor prototype** », Thèse master mécanique, institue de technique supérieure Lisboa, octobre (2009).
- [Kha.02] W.Khalil, Dombre « **modelling, identification and control of robots** » HPS edition, (2002).
- [Kac.97] J. Kacprzyk, « **Multistage Fuzzy control** », Ed. Wiley, (1997).
- [Kim.98] C. N. Kim et M. Trivedi, « **A neuro – fuzzy controller for mobile robot navigation and multirobot convoying** », IEE Transactions on fuzzy systems, Man and cybernetics, Vol. 5, No. 4, pp.345-371, (1998).
- [Kri.96] V. Krishmaswami, C. Siviero, F. Cabognani, and V Utkin. « **Application of sliding mode observers to automobile power-train** ». In : Proc. of the IEEE int. Conf. On Cont. Appli., pages 355.360, (1996).
- [Kre.83] A. Krener and A. Isidori. « **Linearization by output injection and nonlinear observers** ». Syst. & Cont. Lett., 3 :47.52, (1983).
- [Loz.04] R. Lozano P. Castillo and A. Dzul. « **Stabilization of a mini rotorcraft having four rotors** ». Proceedings of 2004 IEEE/RSJ International Conference on Intelligent Robots and Systems, pages 2693 – 2698, (2004).
- [Lan.01] Landau I.D. « **Identification des systèmes** ». HERMES Science Europe Publications, Paris. pp.167-173. (2001).
- [Man.98] N. Manamani. « **Commandes et Observateurs pour systèmes pneumatique Méthodologie pour la locomotion à pattes** ». PhD thesis, Université de Paris VI, Laboratoire de Robotique de Paris France, (1998).
- [MCG.04] S.J.MC Gilvray, « **Attitude stabilization of a quadrotor aircraft** », A thesis submitted in partial fulfilment of the requirements for the degree of Master Science, in control Engineering, Lakhead University, Thunder Bay, Ontario, Canada , June (2004).

- [Mok.04] Abdellah. Mokhtari and A. Benallegue, « **Dynamic Feedback Controller of Euler Angles and Wind parameters estimation for a Quadrotor Unmanned Aerial Vehicle** », Proceedings of the 2004 IEEE, International Conference on Robotics and Automation, New Orleans, LA. April 2004.
- [Non.10] K. Nonami , F. Kendoul ,S S W. Wang .D. Nakazawa « **Unmanned Aerial Vehicles and Micro Aerial Vehicles** » Book Springer Tokyo Dordrecht Heidelberg London New York, Springer(2010).
- [Ped.05] Pedro Castillo, Rogelio Lozano and Alejandro E. Dzul « **Modelling and Control of Mini-Flying Machines** » ,Book Springer-Verlag London Limited (2005).
- [Pat.06]. C.A. Patel, « **Building a Test Bed for mini Quadrotor Unmanned Aerial Vehicle with protective Shroud** », a thesis submitted to the department of Mechanical Engineering and the faculty of the Graduate School of Wichita State University in partial fulfilment the requirements for the degree of Master of science, July (2006) .
- [Sab.04] E. Sabouni, « **Contrôleurs par mode glissant flou et prédictif flou optimisés par l'AG pour la commande de MSAP** », Mémoire de Magister, CU Béchar (2004).
- [Sir.94] H. Sira-Ramirez and S.K. Spurgeon. « **On the robust design of sliding observers for linear systems** ». Syst. & Cont. Lett., 23 : 9.14, (1994).
- [Slo.87] J.J.E. Slotine and W. Li. « **On the adaptive control of robot manipulators. Int. Jour.of Robot** ». Research, 6(3): 49.59, (1987).
- [Slo.91] J.E. Slotine and W. Li, « **Applied nonlinear control**», Prentice Hall, Englewood Cliffs, NJ, (1991).
- [Tay.04] A.Tayebi, S.Mcgilvray« **Attitude stabilisation of a four rotor aerial robot** », IEEE conference on decision and control, December 14-17, Atlantis Paradise Island, Bahamas 1216-1217, (2004).
- [Tab.04] S. Tab, « **Contrôleurs flous et par mode flou glissant optimisés par l'AG pour la commande d'un bras manipulateur** », Mémoire de Magister, CU Béchar (2004).
- [Utk.77] V. I. Utkin, “**Variable structure systems with sliding modes**”, IEEE Transactions on automatic control, vol. 22, No. 2, pp.212-222, (1977).
- [Utk.92] V.I. Utkin, « **Sliding modes in control optimization**», Communication and control Engineering series, Springer Verlag, (1992).
- [Utk.95] V. I. Utkin and S. Drakunov. « **Sliding mode observer** ». tutorial IEEE Conf.On Dec.and Cont., pages 3376.3378, (1995).

- [Vil.02] P. Viljamaa, «**Fuzzy gain scheduling and tuning of multivariable fuzzy control methods of fuzzy computing in control systems** », Thèse de Doctorat University of Technology Finland, (2002).
- [Was.07] S. L. Waslander, « **Multi-Agent systems design for aerospace applications**», a dissertation submitted to the department of aeronautics and astronautics and the committee on graduate studies of Stanford University in partial fulfilment of the requirements for the degree of doctor of philosophy, June (2007).
- [Zem.06] K.Zemalache Meguenni,«**Commande d'un système sous-actionné : application à un drone à quatre hélices** », Thèse de doctorat, Université d'Evry Val d'Essone, Décembre (2006).

Résumé

Dans ce travail, on s'intéresse à la modélisation dynamique du quadrirotor tenant compte les divers phénomènes physiques, qui peuvent influencer la dynamique d'une structure volante.

Par la suite, nous avons étudiés le contrôleur par mode glissant (SMC). Le signal de commande obtenu par le SMC, présente des variations brusques dues au phénomène de broutement. Afin de réduire les effets du phénomène de broutement et d'améliorer davantage les performances de contrôle de quadrirotor, une hybridation, entre la logique floue et le mode de glissement et le PI a été proposée. Ces contrôleurs sont appliqués à la commande de quadrirotor.

Dans la partie qui suit, nous avons présenté la technique de commande par backstepping. Une hybridation avec le PID (Intégrale backstepping) a été proposée afin d'améliorer la robustesse vis-à-vis des perturbations.

Les algorithmes génétiques (AG) sont des algorithmes de recherche efficaces qui utilisent les opérations découvertes dans les génétiques naturelles pour la détermination des extrêmes d'une fonction définie sur un espace de données. Dans ce contexte nous avons utilisé ce type d'algorithme d'optimisation pour trouver des gains optimaux. En suite, un observateur non linéaire est synthétisé pour reconstruire le vecteur d'état d'une façon complète ou d'une façon partielle et d'estimer les états non mesurables et les effets des perturbations extérieures tels que le vent.

Mots clés

quadrirotor, logique floue, mode glissant, backstepping, optimisation, Algorithme Génétique, PID.

Abstract

In this thesis, we are interested in dynamic modeling of quadrotor while taking into account the various physical phenomena, which can influence the dynamics of a flying structure.

Thereafter, we are leaning on the synthesis of a controller by sliding mode (SMC). The control signal obtained by the SMC, present abrupt variations due to the phenomena of chattering, In order to reduce the effects of the phenomena of chattering and to more improve the performances of control of the quadrotor, an hybridization, between fuzzy logic and the sliding mode and PI was proposed. These controllers are applied to the quadrotor control.

In the next part, we presented the technique of control by backstepping. The hybridization between the PID (Integral backstepping) was proposed to more improve the performances of the quadrotor control.

The genetic algorithm are efficient research algorithm with use the method find in natural genetics for find the extremes of function on data space. In this frame work, we have used this kind of the optimization algorithm to tune the PI controller gains to ensure optimal performance. Nonlinear observer is then synthesized in order to reconstruct completely or partially the state vector and to estimate the unmeasured states and the effects of the external disturbances such as wind.

Key words

quadrotor, fuzzy logic, sliding mode, backstepping, optimization, Genetic Algorithm, PID.