

وزارة التعليم العالي والبحث العلمي
MINISTÈRE DE L'ENSEIGNEMENT SUPÉRIEUR ET DE LA RECHERCHE SCIENTIFIQUE
ⵎⵓⵎⵎⵓⵏⵉ ⵏ ⵓⵎⵎⵓⵏⵉ ⵏ ⵜⵉⵣⵓⵣⵓ
X.ⵓⵓⵎⵎⵓⵏⵉ ⵏ ⵓⵎⵎⵓⵏⵉ ⵏ ⵜⵉⵣⵓⵣⵓ
X.ⵓⵓⵎⵎⵓⵏⵉ ⵏ ⵓⵎⵎⵓⵏⵉ ⵏ ⵜⵉⵣⵓⵣⵓ

UNIVERSITE MOULOUD MAMMERI DE TIZI-OUZOU
FACULTE DU GENIE ELECTRIQUE ET DE L'INFORMATIQUE
DEPARTEMENT D'AUTOMATIQUE



Mémoire en vue de l'obtention Du diplôme de master

DOMAINE : Sciences et Technologies

FILIERE : Génie électrique

SPECIALITE : Commande des systèmes

Thème

**Modélisation et commande d'un quadrirotor :
Etude comparative de la commande floue et PID**

Présenté par :

Azouz Mustapha

Encadré par :

M Mohand Achour TOUAT

Mémoire soutenu publiquement le 29/09/2016 devant le jury composé de :

M DJENOUNE S.

Professeur, UMMTO, Président

M Maidi A.

Professeur, UMMTO, Examineur

M Kara R.

MC A, UMMTO, Examineur

M TOUAT M A.

MC B, UMMTO, Encadreur

Remerciements

*J*e tiens à exprimer mes vifs remerciements à toutes les personnes qui m'ont apporté de l'aide dans l'élaboration de ce Travail.

Je tiens à remercier aussi mon promoteur M. Touat.

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

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

Dédicaces

*J*e dédie ce travail :

En premier lieu à mes parents et ma famille qui m'ont soutenu.

Ainsi qu'à tous mes proches et à mes amis pour leurs encouragements et leurs soutiens qui se reconnaîtront.

Table des matières

Introduction générale

Chapitre I : Modélisation d'un drone de type quadrirotor

1. Introduction	3
2. Description du Quadrirotor.....	3
3. Les mouvements du quadrirotor	5
3.1 Mouvement vertical.....	5
3.2 Rotation de roulis	6
3.3 Rotation de tangage	6
3.4 Rotation de lacet.....	7
3.5 Translations sur l'axe X.....	8
3.6 Translations sur l'axe Y.....	8
4. Modèle dynamique d'un quadrirotor.....	9
4.1 Angles d'Euler.....	11
4.2 Vitesses angulaires	11
4.3 Effets physiques agissants sur le quadrirotor	12
4.3.1 Les forces :.....	12
4.3.1.1 La gravité	12
4.3.1.2 Les forces de poussées	12
4.3.1.3 La forces de trainée des hélices.....	13
4.3.1.4 La trainée selon les axes.....	13
4.3.2 Les moments	13
4.3.2.1 Moments dus aux forces de trainée :.....	13
4.3.2.2 Moments dus aux forces de poussées	14
4.3.3 L'effet gyroscopique	14
4.3.3.1 Moment gyroscopique du quadrirotor :.....	14
4.3.3.2 Moment gyroscopique des hélices :.....	14
4.4 Modèle mathématique selon Newton-Euler	15
4.4.1 Equation de mouvement de translation	16
4.4.2 Equations de rotation.....	17

4.5	Représentation d'état	18
4.6	Dynamique des moteurs	19
4.7	Valeurs de simulation	20
5.	Conclusion.....	21

Chapitre II : Réalisation d'une commande floue pour un quadrirotor

1.	Introduction	22
2.	Présentation	22
3.	Principe.....	22
3.1	Univers de discours	22
3.2	Fonctions d'appartenances	23
3.3	Operateurs flous.....	23
3.4	Règles d'inférences floues.....	23
3.5	Structure d'un régulateur flou.....	23
3.5.1	La fuzzification	24
3.5.2	L'inférence et règles floue	24
3.5.3	Défuzzification	24
4.	Application de la commande floue a l'UAV	25
4.1	Axe de translation Z	26
4.2	Axe de rotation φ	27
4.3	Axe de rotation θ	29
4.4	Axe de rotation ψ	30
5.	Simulation de la commande sur Matlab-Simulink	31
5.1	Simulation de la dynamique de vol	31
5.2	Simulation des boucles de régulation	31
6.	Résultats de la simulation.....	34
6.1	Poursuite d'un échelon	34
6.2	Poursuite de consignes aléatoire.....	35
6.3	Poursuite de consigne sinusoïdale	36

6.4	Régulation face à des perturbation	36
7.	Régulation des axes X et Y	38
7.1	Axe de translation X.....	38
7.2	Axe de translation Y.....	40
8.	Simulation sur Matlab-Simulink	41
9.	Résultats de la simulation.....	43
9.1	Poursuite d'un échelon	43
9.2	Poursuite de consignes aléatoire.....	44
9.3	Poursuite de consigne sinusoïdale.....	44
9.4	Régulation face à des perturbation	45
10.	Conclusion.....	46

Chapitre III : Réalisation d'une commande par PID pour un drone de type quadrirotor

1.	Introduction	48
2.	Principe.....	48
2.1	Action proportionnelle.....	48
2.2	Action intégrale	48
2.3	Action dérivée	49
2.4	Action PID.....	49
3.	Choix des paramètres et optimisation.....	49
3.1	Schéma de régulation	49
3.2	Choix de paramètres initiaux.....	50
3.3	Optimisation	51
4.	Simulation des régulateurs	52
5.	Résultats des simulations.....	55
5.1	Régulateur PID non optimisé	55
5.1.1	. Poursuite d'un échelon.....	55
5.1.2	Poursuite de consigne aléatoire.....	57

5.1.3	Poursuite de consigne sinusoïdale	59
5.1.4	Régulation face à des perturbation	61
5.2	Régulateur PID optimisé	63
5.2.1	Poursuite d'un échelon	63
5.2.2	Poursuite de consigne aléatoire	65
5.2.3	Poursuite de consigne sinusoïdale	67
5.2.4	Régulation face à des perturbation	69
6.	Comparaison entre les différentes commandes étudiées	71
7.	Conclusion	71

Chapitre IV : Réalisation d'un quadrirotor

1.	Introduction	72
2.	Les composants	72
2.1	Composants mécaniques	72
2.1.1	Le châssis	72
2.1.2	Les hélices	73
2.2	Composants électriques	74
2.2.1	Les moteurs brushless	74
2.2.2	Les variateurs de vitesse électronique (ESC)	75
2.2.3	Batterie	76
2.3	Les capteurs	76
2.3.1	Capteur gyroscopique et accéléromètre	76
2.3.2	Capteurs à ultrason	76
2.3.3	Capteurs GPS	77
2.3.4	Caméra	77
2.4	La carte programmable	77
3.	Montage	78
4.	Conclusion	79

Conclusion générale

Liste des figures

Figure I. 1 : Direction de rotation des rotor	4
Figure I. 2 : les six degrés de libertés du quadrirotor	4
Figure I. 3 : Mouvement de translation verticale	5
Figure I. 4 : Mouvement de roulis	6
Figure I. 5 : Mouvement de tangage	7
Figure I. 6 : Mouvement de lacet	7
Figure I. 7 : Mouvement de translation sur l'axe X	8
Figure I. 8 : Mouvement de translation sur l'axe Y	8
Figure I. 9 : Repères fixe et mobile du quadrirotor	10
Figure II. 1 : structure d'un régulateur Mamdani	24
Figure II. 2 schéma de régulation flou Z, φ , θ , ψ	25
Figure II. 3 : entrées du régulateur flou de la position Z	26
Figure II. 4 sortie du régulateur flou de la position Z	27
Figure II. 5 entrées du régulateur flou de l'angle Phi	28
Figure II. 6 sortie du régulateur flou de l'angle Phi	28
Figure II. 7 : entrées du régulateur flou de l'angle theta	29
Figure II. 8 : sortie du régulateur flou de l'angle theta	29
Figure II. 9 : entrées du régulateur flou de l'angle Psy	30
Figure II. 10 : sortie du régulateur flou de l'angle Psy	30
Figure II. 11 : simulation Simulink de la dynamique du quadrirotor	31
Figure II. 12 : simulation sur Simulink de la régulation flou Z, φ , θ , ψ	33
Figure II. 13 : Réponse à un échelon de la régulation flou sur les variable Z, Phi, Theta, Psy	34
Figure II. 14 : Réponse à des consignes aléatoires de la régulation flou sur les variable Z, Phi, Theta, Psy	35
Figure II. 15 : Réponse à une consigne sinusoïdale de la régulation flou sur les variable Z, Phi, Theta, Psy	36

Figure II. 16 : Réponse face à une perturbation de la régulation floue sur les variable Z, Phi, Theta, Psy	37
Figure II. 17 : schéma de régulation flou X, Y	38
Figure II. 18 : entrées du régulateur floue de la position X	39
Figure II. 19 sortie du régulateur floue de la position X	39
Figure II. 20 : entrées du régulateur floue de la position Y	40
Figure II. 21 : sortie du régulateur floue de la position Y	41
Figure II. 22 : simulation sur Simulink de la régulation flou X, Y	42
Figure II. 23 : Réponse à un échelon de la régulation floue sur les variables X et Y	43
Figure II. 24 : Réponse à des consignes aléatoires de la régulation floue sur les variables X et Y	44
Figure II. 25 : Réponse à une consigne sinusoïdale de la régulation floue sur les variables X et Y	45
Figure II. 26 : Réponse face à une perturbation de la régulation floue sur les variables X et Y	46
Figure III. 1 : schéma de régulation PID.	50
Figure III. 2 simulation Simulink de la régulation PID Z, phi, theta, psy.	53
Figure III. 3 simulation sur Simulink de la régulation PID X, Y.	54
Figure III. 4 : Réponse à un échelon de la régulation PID non optimisée sur toutes les variables	56
Figure III. 5 : Réponse à des consignes aléatoires de la régulation PID non optimisée sur toutes les variables	58
Figure III. 6 : Réponse à une consigne sinusoïdale de la régulation PID non optimisée sur toutes les variables	60
Figure III. 7 : Réponse face à une perturbation de la régulation PID non optimisé sur toutes les variables	62
Figure III. 8 : Réponse à un échelon de la régulation PID optimisé sur toutes les variables	64
Figure III. 9 : Réponse à des consignes aléatoires de la régulation PID optimisé sur toutes les variables	66
Figure III. 10 : Réponse à une consigne sinusoïdale de la régulation PID optimisée sur toutes les variables	68

Figure III. 11 : Réponse face à une perturbation de la régulation PID optimisé sur toutes les variables	70
Figure IV. 1 : châssis en aluminium	73
Figure IV. 2 : hélices de quadrirotor	74
Figure IV. 3 : moteur brushless	75
Figure IV. 4 : variateur de vitesse	75
Figure IV. 5 capteur gyroscopique	76
Figure IV. 6 : capteur ultrason	77
Figure IV. 7 : schéma de branchement	79

Liste de tableaux

Tableau I. 1 : paramètre du modèle du quadrirotor simulé	20
Tableau III. 1 paramètres du régulateur PID avant optimisation	51
Tableau III. 2 paramètres du régulateur PID après optimisation	52
Tableau III. 3 : tableau comparatif	71

Liste des abréviations

PI : régulateur proportionnel intégrale

PID : régulateur proportionnel intégrale dérivé

VTOL : décollage et atterrissage verticale

DDL : degré de liberté

R^b : repère fixe lié a la terre

R^m : repère mobile lié au centre de gravité du quadrirotor

T : matrice de transformation

R : matrice de rotation

ζ : vecteur de position

φ : angle de roulis

θ : angle de tangage

ψ : angle de lacet

Ω_i : vitesse de rotation dans le repère fixe

P : force de la gravité sur le quadrirotor

m : masse du quadrirotor

g : accélération de pesanteur

F_i : force de poussée de l'hélice i

b : coefficient de portance des hélices

ω : vitesse de rotation des hélices

T_{hi} : force de trainée de l'hélice i

d : coefficient de drag

K_{fti} : coefficient de trainée de translation

M_z : moment autour de l'axe Z

M_a : moment des frottements aérodynamiques

K_{fai} : coefficients des frottement aérodynamiques

M_x : moment autour de l'axe X

M_y : moment autour de l'axe Y

l : longueur d'un bras

M_{gm} : moment gyroscopique du quadrirotor

M_{gh} : moment gyroscopique des hélices

J_r : inertie des rotors

J : matrice d'inertie symétrique

F_f : force de poussée totale des hélices

F_t : force de traînée

F_g : la force de gravité

M_f : moment autour des axe X, Y et Z

V : la tension d'alimentation du moteur

R : la résistance du moteur.

L : l'inductance du moteur.

K_e : la constante de couple mécanique.

K_m : la constante de couple électrique.

i : le courant de l'induit.

C_s : frottement

K_r : la constante de couple de charge.

$\mu_E(x)$: fonction d'appartenance a E

K_p : gain proportionnel

K_i : gain intégrateur

K_d : gain proportionnel

ε : erreur statique

ESC : variateur de vitesse électronique

eFLL : bibliothèque Arduino d'implémentation de logique floue

GPS : Géopositionnement par Satellite

Li-Po : Accumulateur lithium-ion polymère

Introduction générale

Avec la démocratisation des drones et leur grande accessibilité, ces appareils sont de plus en plus utilisés dans un grand nombre de domaines, nous citerons comme exemple la surveillance du trafic routier, l'exploration environnementale, la cartographie et prise de vue aérienne sans compter toutes les utilités militaires. La capacité des drones de type quadrirotor à décoller et atterrir verticalement et leur grande agilité, ainsi que leur capacité à faire du vol stationnaire en font des appareils d'une grande versatilité, et a l'utilité limitée seulement par l'imagination [1].

La modélisation d'un quadrirotor n'est pas une tâche aisée, cela s'explique par la complexité et le nombre importants d'effets agissant sur le système comme les effets gyroscopique, aérodynamiques, la force de gravité les frottements et les forces d'inertie. Tous ces effets font que notre système est fortement non linéaire, rajoutée à ça une forte interaction entre les états et nous obtenons un système difficile à modéliser [2].

De nombreux travaux ont été fait sur la modélisation et la commande d'un quadrirotor, de nombreuses méthodes ont été utilisé pour la commande tel les régulateur PI et PID, la commande à base de backstepping [3] [4], la commande par mode glissant ou encore la commande par réseaux de neurones. Ces travaux en tous en commun le but de stabiliser et d'asservir un quadrirotor et de lui faire suivre une trajectoire donnée.

Vue la difficulté de trouver un modèle mathématique fiable, la commande floue constitue une alternative au contrôle de ces appareils, ce type de régulation est non linéaire et se base sur les connaissances de l'expert qui réalise la commande. Le principe de cette régulation repose sur le choix de fonctions d'appartenances et de règles régissant le comportement du système.

Ce mémoire s'articule autour de quatre chapitres :

Le premier chapitre présentera une modélisation mathématique du quadrirotor, qui consistera en une description de ce système et de son principe de vol puis nous développeront les différentes équations qui régissent ce système pour enfin proposer un modèle d'état en vue de réaliser une simulation de ce système sous Matlab-Simulink.

Le second chapitre introduira la logique floue et son principe en proposant une brève présentation de la régulation floue, des fonctions d'appartenances des règles floues et de la structure de régulation choisie qui est de type Mamdani. Puis proposera une commande à l'aide de régulateurs flous qui aura pour but de stabiliser et de commander un quadrirotor, ainsi nous présenterons le schéma de régulation et les règles floues utilisées lors de la conception de cette commande. Enfin nous verrons les différents résultats obtenus lors des simulations effectuées grâce au modèle présenté dans le premier chapitre.

Le troisième chapitre exposera une technique de régulation par PID et une optimisation des paramètres du PID par minimisation de l'erreur quadratique. Nous présenterons le résultat obtenu lors de l'optimisation puis nous verrons les résultats obtenus lors des simulations et nous les comparerons à ceux obtenus lors du chapitre précédent.

Le quatrième chapitre portera sur les différentes étapes à suivre pour pouvoir réaliser une maquette physique d'un quadrirotor. Nous verrons ainsi les différents choix disponibles pour la sélection des composants et le schéma de câblage proposé pour cette réalisation.

Ce mémoire s'achèvera sur une conclusion générale.

Chapitre I

Modélisation d'un drone de type quadrirotor

1. Introduction

La première étape dans la conception d'un système autonome est la modélisation, pour cela, il faut comprendre et maîtriser les équations qui régissent la dynamique de ce système. De la précision de cette modélisation découlera la justesse de la simulation et la pertinence de la commande.

Etant donné le nombre important d'effets qui affectent la dynamique du quadrirotor (les effets aérodynamiques, la gravité, les effets gyroscopiques, les frottements et le moment d'inertie) il fait partie des systèmes volants les plus complexes [1].

2. Description du Quadrirotor

Un Quadrirotor est un drone de type VTOL (vertical take-off and landing), ce qui veut dire que l'appareil est capable de décoller et d'atterrir verticalement, (contrairement aux avions par exemple qui ont besoin de prendre de l'élan). Un Quadrirotor est composé : d'un châssis en croix(X) ou en plus (+), de quatre hélices attachées à des moteurs (le plus souvent des moteurs à courant continu brushless) qui sont placés aux extrémités de la structure, d'une carte de commande située au milieu de la structure, qui sert à récupérer les informations des capteurs et de commander les moteurs, d'une ou plusieurs batteries et des différents capteurs (accéléromètre, capteur ultra son...etc.) [2-3].

Dans les hélicoptères classiques, le rotor principal produit un couple réactif en tournant qui incite l'appareil à tourner sur lui-même, pour contrer ce couple, les hélicoptères sont généralement équipés d'un rotor de queue qui produit une poussée latérale ou encore d'un deuxième rotor associé au premier qui tourne dans le sens inverse.

Dans le cas d'un quadrirotor, afin d'éviter à l'appareil de tourner sur lui-même, il est indispensable qu'une paire d'hélice opposée sur un bras tourne dans un même sens et que l'autre paire tourne dans le sens inverse ceci pour contrer l'effet du couple réactif.

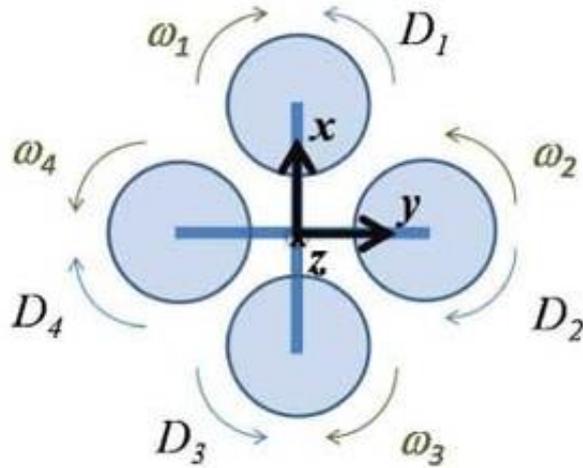


Figure I.1 : Direction de rotation des rotors

Un quadrirotor est défini dans l'espace par six degrés de libertés « 6 DDL » (trois rotations et trois translations). Pour agir sur ces six états il est nécessaire de faire varier astucieusement la puissance des moteurs, le quadrirotor s'incline vers le rotor le plus lent, grâce à cette propriété il est possible d'agir sur les angles de l'appareil et de le faire incliner autour d'un des axes et ainsi réaliser la translation désirée. Etant donné que le quadrirotor ne se compose que de quatre actionneurs et est défini par six états c'est donc un système sous actionné [2].

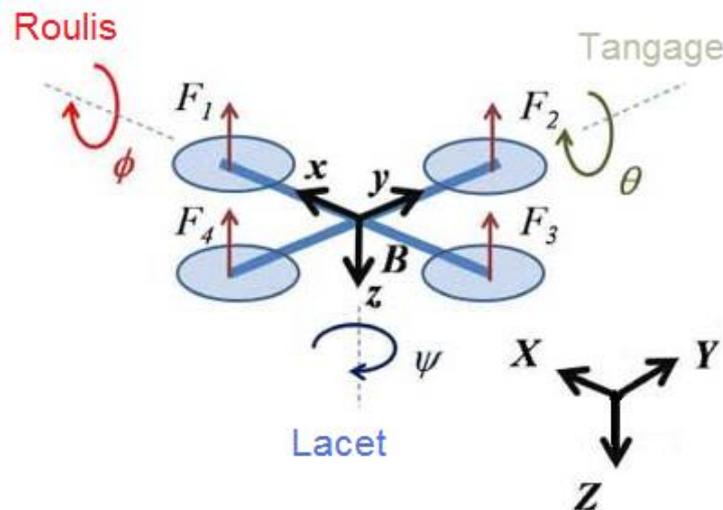


Figure I.2 : les six degrés de libertés du quadrirotor

3. Les mouvements du quadrirotor

Pour réaliser les six mouvements du quadrirotor nous n'avons à notre disposition que quatre actionneurs, par conséquent les mouvements de l'appareil sont couplés ce qui signifie que nous ne pouvant pas effectuer un des mouvements sans en impliquer d'autres. Par exemple en réduisant la vitesse du rotor droit, le quadrirotor s'incline de ce côté et l'équilibre entre les rotors qui tourne dans le sens des aiguilles d'une montre et les rotors qui tourne dans le sens inverse est perturbé, ce qui implique deux mouvements de rotation appelés roulis et lacet. Le mouvement de rotation de roulis implique un mouvement de translation sur l'axe Y. C'est cette particularité qui permet de commander le quadrirotor sur six degrés de liberté avec seulement quatre actionneurs [4]. Ainsi le quadrirotor a les six mouvements suivants :

- Mouvement vertical
- Rotation de roulis
- Rotation de tangage
- Rotation de lacet
- Translations sur l'axe X
- Translations sur l'axe Y

3.1 Mouvement vertical

La force combinée des quatre rotors, projetés sur l'axe Z, s'oppose à la force de pesanteur. C'est la somme de ces deux forces qui détermine si le quadrirotor s'élève ou descend, si la force exercée par les rotors est supérieure à la force de pesanteur alors l'appareil s'élève sinon l'appareil descend, si ces deux forces sont d'intensité égale alors le quadrirotor planera.

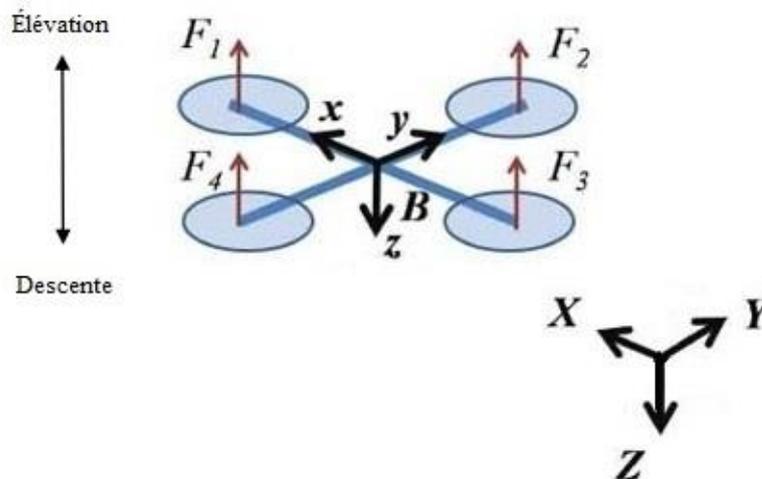


Figure I. 3 : Mouvement de translation verticale

3.2 Rotation de roulis

Le mouvement de roulis est obtenu en appliquant une différence de poussé entre le rotor 2 et 4, ce qui implique une rotation autour de l'axe X couplé à une translation sur l'axe Y. Si le changement de poussé est appliqué sur un seul rotor alors l'équilibre général des rotors est rompu et l'appareil effectue une rotation sur l'axe Z (Lacet).

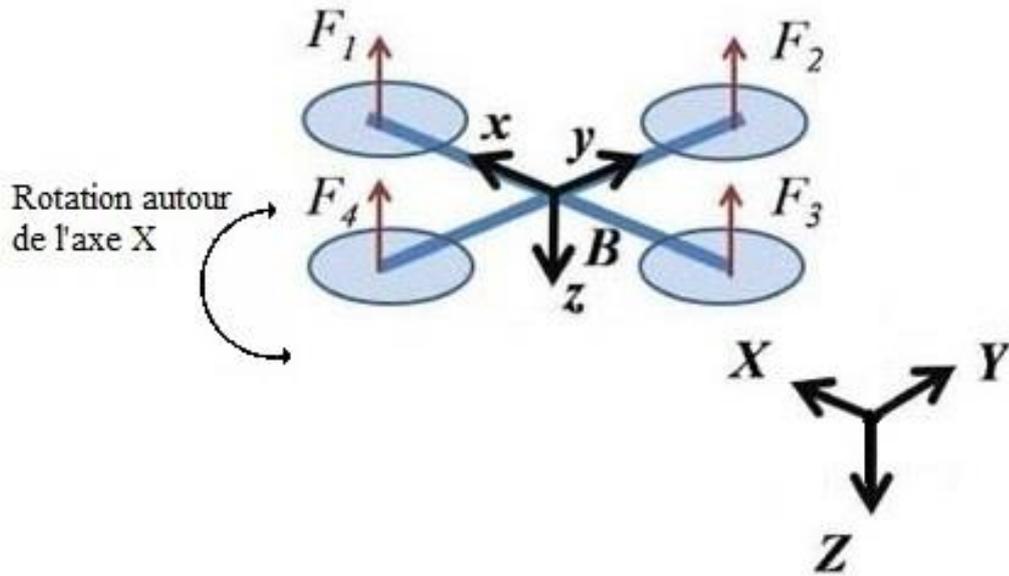


Figure I. 4 : Mouvement de roulis

3.3 Rotation de tangage

Le mouvement de Tangage est obtenu en appliquant une différence de poussé entre le rotor 1 et 3, ce qui implique une rotation autour de l'axe Y couplé à une translation sur l'axe X. Si le changement de poussé est appliqué sur un seul rotor alors l'équilibre générale des rotors est rompu et l'appareil effectue une rotation sur l'axe Z (Lacet).

Généralement ce mouvement de Lacet est non désiré. Pour pallier à cela, il est nécessaire d'équilibrer les différences de poussé, c'est-à-dire, lorsqu'on augmente la vitesse de rotation d'un rotor d'une certaine valeur (le rotor 2 par exemple) il faut diminuer celle du rotor opposé de la même valeur (le rotor 4), de cette façon l'équilibre entre les quatre rotors est respecté et l'appareil n'effectue pas de mouvement de rotation Lacet, ceci permet aussi de garder la somme des forces des rotors égales et ainsi de ne pas perdre en force de portance.

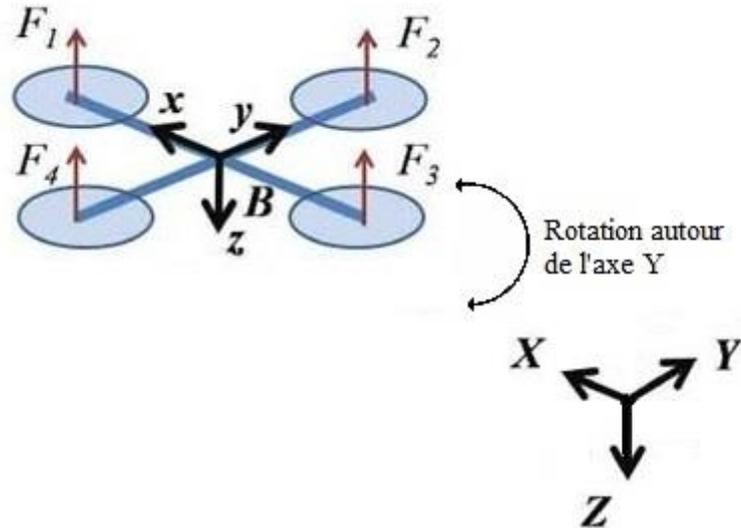


Figure I. 5 : Mouvement de tangage

3.4 Rotation de lacet

Pour réaliser ce mouvement, il est nécessaire d'appliquer un couple autour de l'axe Z, pour ce faire, il suffit d'appliquer une différence de vitesse entre les rotors (1-3) et les rotors (2-4). Pour ne pas coupler ce mouvement de rotation avec les autres mouvements de rotation, il est nécessaire que les vitesses des paires de rotors sur le même bras soient égales. Et pour garder la force de poussée constante, il est indispensable que l'augmentation de la vitesse de rotation sur une paire de rotors soit égale à la diminution de la vitesse de rotation sur l'autre paire.

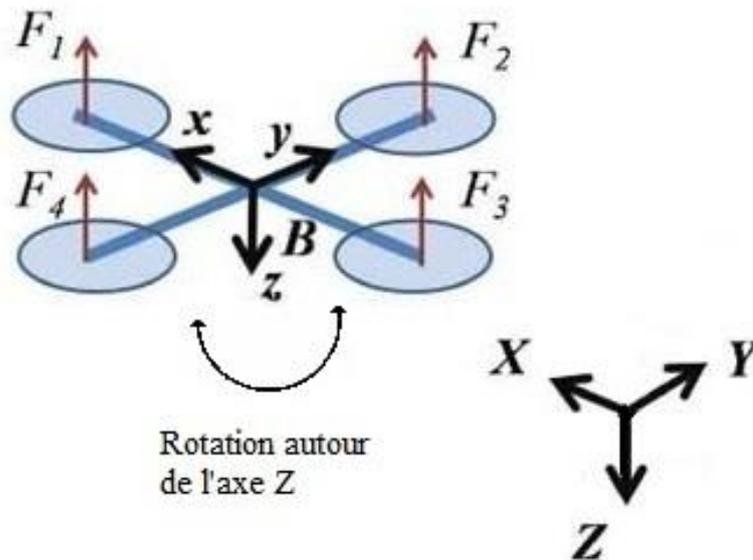


Figure I. 6 : Mouvement de lacet

3.5 Translations sur l'axe X

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

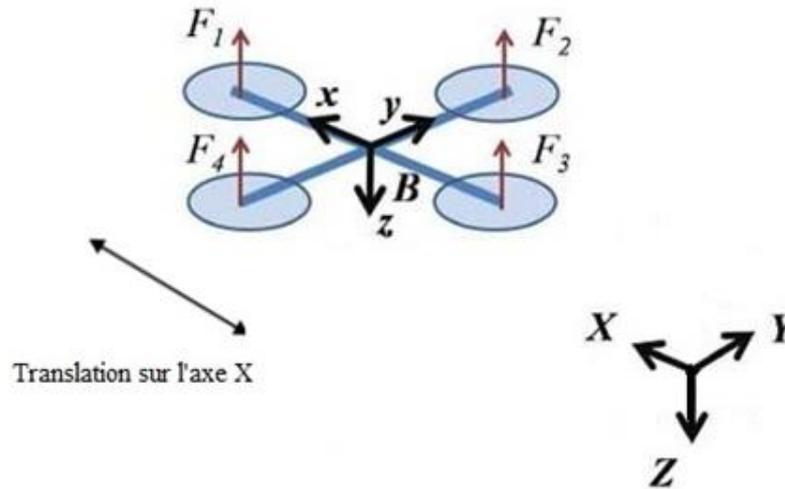


Figure I. 7 : Mouvement de translation sur l'axe X

3.6 Translations sur l'axe Y

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

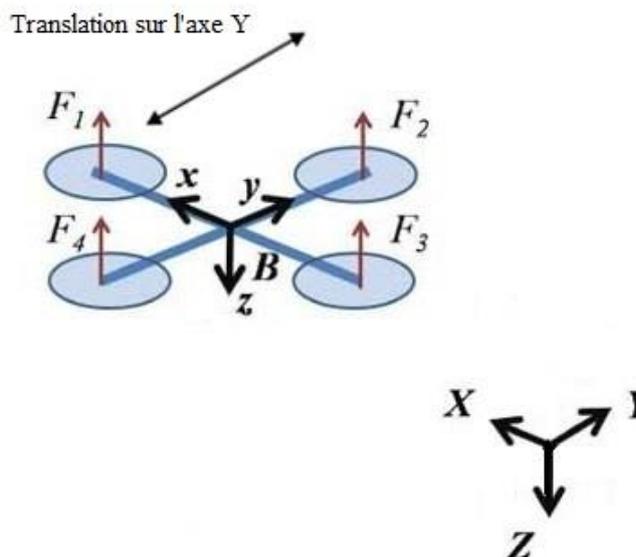


Figure I. 8 : Mouvement de translation sur l'axe Y

4. Modèle dynamique d'un quadrirotor

La dynamique fortement non linéaire du quadrirotor et le couplage des différents états rendent la modélisation de son vol difficile. Pour réaliser cette tâche plusieurs hypothèses ont été proposées dans la littérature tels que [5], [6] :

- Le centre de gravité de l'appareil est situé au centre de celui-ci et est considéré confondu avec le repère lié à la structure.
- La structure composant le quadrirotor est supposé rigide et symétrique.
- Les hélices sont supposées rigides pour pouvoir négliger l'effet de leurs déformations lors de la rotation.
- Les forces de portance et de traînée sont proportionnelles aux carrés de la vitesse de rotation des rotors.

Nous allons proposer dans ce qui suit un modèle qui nous permettra la simulation de notre système.

Pour ce faire nous allons définir deux repères, un repère fixe lié à la terre R^b et un repère mobile lié au centre de gravité du quadrirotor R^m .

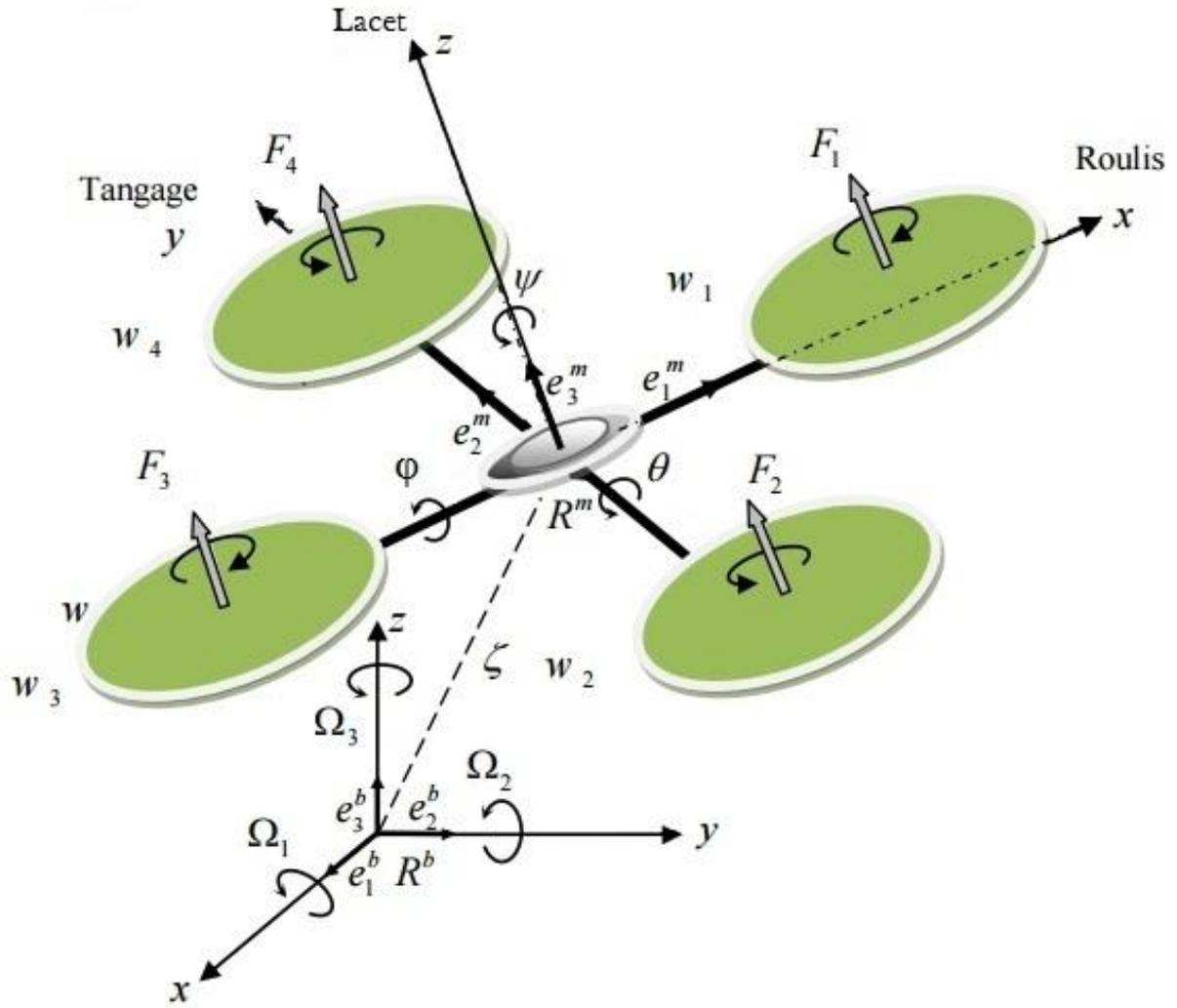


Figure I.9 : Repères fixe et mobile du quadrirotor

Pour passer du repère fixe au repère mobile nous définissons une matrice de transformation T qui contient la position et l'orientation du repère mobile par rapport au repère fixe.

$$T = \begin{bmatrix} R & \zeta \\ 0 & 1 \end{bmatrix} \quad (1.1)$$

Avec R la matrice de rotation de l'objet mobile.

$\zeta = [x \ y \ z]^T$ le vecteur de position.

Pour déterminer les éléments de la matrice R nous allons utiliser les angles d'Euler

4.1 Angles d'Euler

Au début le repère mobile coïncide avec le repère fixe, puis celui-ci effectue trois mouvements de rotation, une rotation autour de l'axe X avec un angle de Roulis φ , une autre rotation autour de l'axe Y avec un angle de Tangage θ , et enfin une rotation autour de l'axe Z avec un angle de Lacet ψ .

Nous supposons que $(-\frac{\pi}{2} < \varphi < \frac{\pi}{2})$ et $(-\frac{\pi}{2} < \theta < \frac{\pi}{2})$ et $(-\pi < \psi < \pi)$.

Nous obtenons R comme suit [3][5] :

$$R = \text{Rot}_z(\psi) \times \text{Rot}_y(\theta) \times \text{Rot}_x(\varphi) =$$

$$\begin{bmatrix} \cos(\psi) & -\sin(\psi) & 0 \\ \sin(\psi) & \cos(\psi) & 0 \\ 0 & 0 & 1 \end{bmatrix} \cdot \begin{bmatrix} \cos(\theta) & 0 & \sin(\theta) \\ 0 & 1 & 0 \\ -\sin(\theta) & 0 & \cos(\theta) \end{bmatrix} \cdot \begin{bmatrix} 1 & 0 & 0 \\ 0 & \cos(\varphi) & -\sin(\varphi) \\ 0 & \sin(\varphi) & \cos(\varphi) \end{bmatrix} \quad (\text{I.2})$$

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

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

4.2 Vitesses angulaires

Nous posons les vitesses de rotations Ω_1 , Ω_2 et Ω_3 dans le repère fixe, Ces vitesses sont exprimées en fonction des vitesses $\dot{\varphi}$, $\dot{\theta}$ et $\dot{\psi}$ dans le repère mobile. Au début les deux repères sont confondus et la rotation de Roulis a lieu, viens ensuite une rotation de Tangage le vecteur de vitesse doit être exprimé dans le repère fixe, on multiplie donc le vecteur $\dot{\theta}$ par $\text{Rot}_x(\varphi)^{-1}$, et enfin nous avons la rotation de Lacet et pour exprimer le vecteur de vitesse $\dot{\psi}$ dans le repère fixe on le multiplie par $[\text{Rot}_y(\theta) \cdot \text{Rot}_x(\varphi)]^{-1}$ [5].

Nous avons donc :

$$\Omega = \begin{bmatrix} \Omega_1 \\ \Omega_2 \\ \Omega_3 \end{bmatrix} = \begin{bmatrix} \dot{\varphi} \\ 0 \\ 0 \end{bmatrix} + \text{Rot}_x(\varphi)^{-1} \begin{bmatrix} 0 \\ \dot{\theta} \\ 0 \end{bmatrix} + [\text{Rot}_y(\theta) \text{Rot}_x(\varphi)]^{-1} \begin{bmatrix} 0 \\ 0 \\ \dot{\psi} \end{bmatrix} \quad (\text{I.4})$$

Finalement nous obtenons :

$$\begin{aligned} \Omega = \begin{bmatrix} \Omega_1 \\ \Omega_2 \\ \Omega_3 \end{bmatrix} &= \begin{bmatrix} \dot{\varphi} \\ 0 \\ 0 \end{bmatrix} + \begin{bmatrix} 0 \\ \dot{\theta} \cos(\varphi) \\ -\dot{\theta} \sin(\varphi) \end{bmatrix} + \begin{bmatrix} -\dot{\psi} \sin(\theta) \\ \dot{\psi} \sin(\varphi) \cos(\theta) \\ \dot{\psi} \cos(\varphi) \cos(\theta) \end{bmatrix} \\ &= \begin{bmatrix} \dot{\varphi} - \dot{\psi} \sin(\theta) \\ \dot{\theta} \cos(\varphi) + \dot{\psi} \sin(\varphi) \cos(\theta) \\ \dot{\psi} \cos(\varphi) \cos(\theta) - \dot{\theta} \sin(\varphi) \end{bmatrix} \end{aligned} \quad (I.5)$$

Il est supposé que le quadrirotor fait de petites rotations de roulis et de tangage nous pouvons approximer comme suit : $\cos(\varphi)=\cos(\theta)=1$ et $\sin(\varphi)=\sin(\theta)=0$ la vitesse angulaire pourra donc s'écrire comme suit :

$$\Omega = \begin{bmatrix} \dot{\varphi} \\ \dot{\theta} \\ \dot{\psi} \end{bmatrix} \quad (I.6)$$

4.3 Effets physiques agissants sur le quadrirotor

4.3.1 Les forces :

Les forces agissant sur le système sont :

4.3.1.1 La gravité

La gravité agit sur le quadrirotor proportionnellement à sa masse et est donnée par :

$$P = m * g \quad (I.7)$$

Avec : m la masse du quadrirotor, $g= 9.81\text{m/s}^2$.

4.3.1.2 Les forces de poussées

Les hélices entraînées par les moteurs provoquent une force de poussé perpendiculaire au plan des hélices, ces forces sont proportionnelles au carrée de la vitesse de rotation des moteurs :

$$F_i = b\omega_i^2 \quad (I.8)$$

Avec $i= \{1,2,3,4\}$

Et b le coefficient de portance des hélices, il dépend de leurs formes, du nombre de pales et de la densité de l'air.

4.3.1.3 La forces de trainée des hélices

La trainée est la force qui s'oppose au mouvement d'un corps dans un liquide ou un gaz, elle agit comme un frottement. Dans le cas des hélices cette force est proportionnelle au carré de la vitesse de rotation de ces dernières et est donnée par la relation

$$T_{hi} = d\omega^2_i \quad (I.9)$$

Avec d le coefficient de drag, il dépend de la fabrication de l'hélice.

Et $i = \{1,2,3,4\}$

4.3.1.4 La trainée selon les axes

Elle est due au mouvement du quadrirotor et est donnée comme suit :

$$F_t = \begin{bmatrix} -K_{ftx} & 0 & 0 \\ 0 & -K_{fty} & 0 \\ 0 & 0 & -K_{ftz} \end{bmatrix} V \quad (I.10)$$

Avec $K_{ftx}, K_{fty}, K_{ftz}$ les coefficients de trainée de translation et V la vitesse linéaire.

4.3.2 Les moments

Plusieurs moments agissent sur le quadrirotor, ils sont dus aux différentes forces de poussées et de trainées et aux effets gyroscopiques.

4.3.2.1 Moments dus aux forces de trainée :

- Les couples de trainée de chaque hélice induisent un couple réactif qui agit sur la rotation autour de l'axe z , ce moment est donné par :

$$M_z = d(\omega_1^2 - \omega_2^2 + \omega_3^2 - \omega_4^2) \quad (I.11)$$

- Les frottements aérodynamiques induisent des moments :

$$M_a = \begin{bmatrix} K_{fax}\dot{\phi}^2 \\ K_{fay}\dot{\theta}^2 \\ K_{faz}\dot{\psi}^2 \end{bmatrix} \quad (I.12)$$

Avec $K_{fax}, K_{fay}, K_{faz}$ les coefficients de frottement aérodynamiques et $\dot{\phi}, \dot{\theta}$ et $\dot{\psi}$ les vitesses angulaires.

4.3.2.2 Moments dus aux forces de poussées

- Lorsqu'il y a une différence entre les forces de portances des rotors 2 et 4, cela implique une rotation autour de l'axe X. Ce moment est donné par :

$$M_x = l.(F_4 - F_2) = l.b(\omega_4^2 - \omega_2^2) \quad (I.13)$$

Avec l la longueur du bras où est fixé le quadrirotor.

- Lors d'une rotation autour de l'axe Y, qui est due à une différence entre les forces de portances des rotors 1 et 3, nous obtenons un moment :

$$M_y = l.(F_3 - F_1) = l.b(\omega_3^2 - \omega_1^2) \quad (I.14)$$

4.3.3 L'effet gyroscopique

Lorsqu'un objet tourne il est sujet à l'effet gyroscopique, c'est-à-dire que l'objet oppose une résistance au changement d'orientation de son plan de rotation. Nous avons donc deux moments gyroscopiques :

4.3.3.1 Moment gyroscopique du quadrirotor :

Ce sont les moments dus aux rotations du quadrirotor sur lui-même, ils sont décrits par la relation suivante :

$$M_{gm} = \Omega \wedge J . \Omega \quad (I.15)$$

Avec J la matrice d'inertie du système.

4.3.3.2 Moment gyroscopique des hélices :

Ce sont les Moments dus aux rotations des hélices sur leurs axes, ils sont définis comme suit [6] :

$$M_{gh} = \sum_{i=1}^4 \Omega \wedge J_r [0 \quad 0 \quad (-1)^{i+1} \omega_i] = \begin{bmatrix} J_r \dot{\theta} \bar{\Omega} \\ -J_r \dot{\phi} \bar{\Omega} \\ 0 \end{bmatrix} \quad (I.16)$$

Avec : J_r l'inertie des rotors et $\bar{\Omega} = \omega_1 \omega_2 \omega_3 \omega_4$

4.4 Modèle mathématique selon Newton-Euler

La méthode Newton-Euler est une des méthodes les plus utilisées dans la modélisation de robots, elle consiste en les équations suivantes [6] :

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

Avec :

- ζ : le vecteur de position du quadrirotor.
- m : la masse totale du quadrirotor.
- Ω : le vecteur de vitesse angulaire exprimé dans le repère fixe
- R : la matrice de rotation
- J : la matrice d'inertie symétrique :

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

- $S(\Omega)$: la matrice antisymétrique associé au vecteur Ω , elle est donnée par

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

- F_f : la force de poussée totale générée par les quatre rotors :

$$F_f = R \times \begin{bmatrix} 0 & 0 & \sum_{i=1}^4 F_i \end{bmatrix}^T \quad (\text{I.20})$$

- F_t : la force de trainée selon les axes (x.y.z) est donnée comme suit :

$$F_t = \begin{bmatrix} -K_{ftx}\dot{x} \\ -K_{fity}\dot{y} \\ -K_{ftz}\dot{z} \end{bmatrix} \quad (I.21)$$

- F_g est donnée par :

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

- M_f : moment provoqué par les forces de poussée et de trainée.

$$M_f = \begin{bmatrix} M_x \\ M_y \\ M_z \end{bmatrix} \quad (I.23)$$

4.4.1 Equation de mouvement de translation

D'après le modèle de Newton-Euler nous avons :

$$m\ddot{\zeta} = F_f + F_t + F_g \quad (I.24)$$

Nous réalisons les transformations nécessaires pour aboutir à :

$$M \begin{bmatrix} \ddot{x} \\ \ddot{y} \\ \ddot{z} \end{bmatrix} = \begin{bmatrix} c(\varphi) \cdot s(\theta) \cdot c(\psi) + s(\psi) \cdot s(\varphi) \\ c(\varphi) \cdot s(\theta) \cdot s(\psi) - s(\varphi) \cdot c(\psi) \\ c(\varphi) \cdot c(\theta) \end{bmatrix} \sum_{i=1}^4 F_i + \begin{bmatrix} -K_{ftx}\dot{x} \\ -K_{fity}\dot{y} \\ -K_{ftz}\dot{z} \end{bmatrix} + \begin{bmatrix} 0 \\ 0 \\ -mg \end{bmatrix} \quad (I.25)$$

Nous obtenons donc les équations différentielles qui définissent le mouvement de translation comme suit :

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

4.4.2 Equations de rotation

L'équation de rotation est donnée par :

$$J\dot{\Omega} = M_{gm} + M_f + M_a + M_{gh} \quad (I.27)$$

$$\begin{bmatrix} I_x & 0 & 0 \\ 0 & I_y & 0 \\ 0 & 0 & I_z \end{bmatrix} \begin{bmatrix} \dot{\varphi} \\ \dot{\theta} \\ \dot{\psi} \end{bmatrix} = - \begin{bmatrix} \dot{\varphi} \\ \dot{\theta} \\ \dot{\psi} \end{bmatrix} \wedge \begin{bmatrix} I_x & 0 & 0 \\ 0 & I_y & 0 \\ 0 & 0 & I_z \end{bmatrix} \begin{bmatrix} \dot{\varphi} \\ \dot{\theta} \\ \dot{\psi} \end{bmatrix} + \begin{bmatrix} M_x \\ M_y \\ M_z \end{bmatrix} - \begin{bmatrix} K_{fax} \dot{\varphi}^2 \\ K_{fay} \dot{\theta}^2 \\ K_{faz} \dot{\psi}^2 \end{bmatrix} - \begin{bmatrix} J_r \dot{\theta} \bar{\Omega} \\ -J_r \dot{\varphi} \bar{\Omega} \\ 0 \end{bmatrix} \quad (I.28)$$

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

Nous pouvons maintenant poser un modèle complet régissant un quadrirotor :

$$\begin{cases} \ddot{\varphi} = \dot{\theta} \dot{\psi} \frac{(I_y - I_z)}{I_x} - \frac{J_r}{I_x} \bar{\Omega} \dot{\theta} - \frac{K_{fax}}{I_x} \dot{\varphi}^2 + \frac{U_2}{I_x} \\ \ddot{\theta} = \dot{\varphi} \dot{\psi} \frac{(I_z - I_x)}{I_y} - \frac{J_r}{I_y} \bar{\Omega} \dot{\varphi} - \frac{K_{fay}}{I_y} \dot{\theta}^2 + \frac{U_3}{I_y} \\ \ddot{\psi} = \dot{\theta} \dot{\varphi} \frac{(I_x - I_y)}{I_z} - \frac{K_{faz}}{I_z} \dot{\psi}^2 + \frac{U_4}{I_z} \\ \ddot{x} = \frac{(c(\varphi) \cdot s(\theta) \cdot c(\psi) + s(\psi) \cdot s(\varphi))}{m} U_1 - \frac{K_{ftx}}{m} \dot{x} \\ \ddot{y} = \frac{(c(\varphi) \cdot s(\theta) \cdot s(\psi) - s(\varphi) \cdot c(\psi))}{m} U_1 - \frac{K_{fity}}{m} \dot{y} \\ \ddot{z} = \frac{(c(\varphi) \cdot c(\theta))}{m} U_1 - \frac{K_{ftz}}{m} \dot{z} - g \end{cases} \quad (I.30)$$

Avec :

$$\begin{bmatrix} U_1 \\ U_2 \\ U_3 \\ U_4 \end{bmatrix} = \begin{bmatrix} b & b & b & b \\ 0 & -lb & 0 & lb \\ -lb & 0 & lb & 0 \\ d & -d & d & -d \end{bmatrix} \begin{bmatrix} \omega_1^2 \\ \omega_2^2 \\ \omega_3^2 \\ \omega_4^2 \end{bmatrix} \quad (I.31)$$

4.5 Représentation d'état

Nous allons réécrire notre modèle mathématique sous forme de modèle d'état, pour cela nous posons :

$$\begin{aligned} X &= [\varphi \quad \dot{\varphi} \quad \theta \quad \dot{\theta} \quad \psi \quad \dot{\psi} \quad x \quad \dot{x} \quad y \quad \dot{y} \quad z \quad \dot{z}]^T \\ &= [x_1 \quad x_2 \quad x_3 \quad x_4 \quad x_5 \quad x_6 \quad x_7 \quad x_8 \quad x_9 \quad x_{10} \quad x_{11} \quad x_{12}]^T \end{aligned} \quad (I.32)$$

Finalement nous obtenons la représentation d'état suivante

$$\left\{ \begin{array}{l} \dot{x}_1 = x_2 \\ \dot{x}_2 = a_1 x_4 x_6 + a_2 x_2^2 + a_3 \bar{\Omega} x_4 + b_1 U_2 \\ \dot{x}_3 = x_4 \\ \dot{x}_4 = a_4 x_2 x_6 + a_5 x_4^2 + a_6 \bar{\Omega} x_2 + b_2 U_3 \\ \dot{x}_5 = x_6 \\ \dot{x}_6 = a_7 x_2 x_4 + a_8 x_6^2 + b_3 U_4 \\ \dot{x}_7 = x_8 \\ \dot{x}_8 = a_9 x_8 + \frac{\cos(x_1) \cos(x_5) \sin(x_3) + \sin(x_1) + \sin(x_5)}{m} U_1 \\ \dot{x}_9 = x_{10} \\ \dot{x}_{10} = a_{10} x_{10} + \frac{\cos(x_1) \sin(x_3) \sin(x_5) - \sin(x_1) + \sin(x_5)}{m} U_1 \\ \dot{x}_{11} = x_{12} \\ \dot{x}_{12} = a_{11} x_{12} + \frac{\cos(x_1) \cos(x_3)}{m} U_1 - g \end{array} \right. \quad (I.33)$$

Avec :

$$\begin{aligned} a_1 &= \frac{I_y - I_z}{I_x}, a_2 = -\frac{K_{fax}}{I_x}, a_3 = -\frac{J_r}{I_x}, a_4 = \frac{I_z - I_x}{I_y}, a_5 = -\frac{K_{fay}}{I_y}, a_6 = \frac{J_r}{I_y}, \\ a_7 &= \frac{I_x - I_y}{I_z}, a_8 = -\frac{K_{faz}}{I_z}, a_9 = -\frac{K_{ftx}}{m}, a_{10} = -\frac{K_{fty}}{m}, a_{11} = -\frac{K_{fxy}}{m}, \\ b_1 &= \frac{1}{I_x}, b_2 = \frac{1}{I_y}, b_3 = \frac{1}{I_z}, \end{aligned} \quad (I.34)$$

4.6 Dynamique des moteurs

Avant de pouvoir commander les moteurs il faut trouver la vitesse désirée de chaque moteur, pour cela nous utiliserons la formule suivante [6] :

$$\omega_d = M^{-1}U \quad (\text{I.35})$$

Avec $\omega_d = (\omega_{d1}^2, \omega_{d2}^2, \omega_{d3}^2, \omega_{d4}^2)^T$ et $U = (U_1, U_2, U_3, U_4)^T$ et M une matrice inversible obtenu à partir de (I.31) tel que

$$M^{-1} = \begin{bmatrix} b & b & b & b \\ 0 & -lb & 0 & lb \\ -lb & 0 & lb & 0 \\ d & -d & d & -d \end{bmatrix}^{-1} \quad (\text{I.36})$$

Les moteurs utilisés pour ce type d'appareil sont des moteurs à courant continu brushless, la dynamique est donc donnée comme suit :

$$\begin{cases} V = Ri + L \frac{di}{dt} + k_e \omega \\ k_m i = J_r \frac{d\omega}{dt} + C_s + k_r \omega^2 \end{cases} \quad (\text{I.37})$$

Avec :

- V : la tension d'alimentation du moteur
- R : la résistance du moteur.
- L : l'inductance du moteur.
- K_e : la constante de couple mécanique.
- ω : la vitesse de rotation du moteur.
- K_m : la constante de couple électrique.
- i : le courant de l'induit.
- J_r : l'inertie du rotor
- C_s : frottement
- K_r : la constante de couple de charge.

Comme les moteurs utilisés sont petits, ils ont donc une inductance faible. Par conséquent la dynamique des moteurs peut être approximé comme suit [2] :

$$V = \frac{C_s R}{K_m} + K_e \omega + \frac{K_r R}{K_m} \omega^2 + \frac{K_m}{J_r R} \dot{\omega} \quad (I.38)$$

4.7 Valeurs de simulation

Le tableau suivant représente les valeurs utilisées lors de la simulation de la dynamique du système sur Matlab-Simulink :

Paramètre	Valeur	Paramètre	Valeur
m	1 kg	Iz	$7,6566 \times 10^{-3}$ kg.m2
g	9,806 m/s2	Kfax	$5,5670 \times 10^{-4}$ N/rad/s
l	0,25 m	Kfay	$5,5670 \times 10^{-4}$ N/rad/s
b	$2,9842 \times 10^{-5}$ N/rad/s	Kfaz	$6,3540 \times 10^{-4}$ N/rad/s
d	$3,2320 \times 10^{-7}$ N.m/rad/	Kftx	$5,5670 \times 10^{-4}$ N/m/s
Jr	$2,8385 \times 10^{-5}$ kg.m2	Kfty	$5,5670 \times 10^{-4}$ N/m/s
Ix	$3,8278 \times 10^{-3}$ kg.m2	Kftz	$6,3540 \times 10^{-4}$ N/m/s
Iy	$3,8288 \times 10^{-3}$ kg.m2		

Tableau I. 1: paramètre du modèle du quadrirotor simulé [5]

5. Conclusion

Dans ce chapitre, nous nous sommes proposés de réaliser une modélisation complète d'un quadrirotor. Nous avons commencé par la description du fonctionnement général d'un quadrirotor et les différents mouvements qu'il peut effectuer lors d'un vol, nous avons ensuite défini les effets physiques qui affectent le vol d'un quadrirotor, pour enfin poser un modèle mathématique décrivant la dynamique du quadrirotor selon le formalisme de Newton-Euler que nous avons représenté dans un modèle d'état.

A partir de la modélisation obtenue nous avons pu en conclure que le quadrirotor est un système non linéaire, sous actionné et à forte interaction entre les états. Dans le chapitre suivant nous allons réaliser une commande par logique floue de type Mamdani pour le quadrirotor.

Chapitre II

Réalisation d'une commande floue pour un quadrirotor

1. Introduction

Lors du chapitre précédent, nous avons abouti à un model dynamique du quadrirotor, nous allons à présent concevoir un contrôleur flou et réaliser des tests pour vérifier son efficacité.

2. Présentation

Les bases de la logique floue ont été établies en 1965 par le professeur Lotfi Zadeh de l'université de Berkeley en Californie, mais ce n'est qu'en 1974 que le professeur Mamdani propose la première application industrielle en définissant la structure de base des régulateurs flous et les principes de la commande floue, il commande ainsi un générateur de vapeur [7][8].

Les avantages de la logique floue sont nombreux :

- La robustesse de la commande.
- La possibilité de commande non linéaire.
- La réduction du temps de développement et de maintenance.
- La non nécessité d'une modélisation mathématique rigoureuse du processus.
- La simplicité de définition et de conception.

3. Principe

Dans la logique conventionnelle, l'appartenance d'un élément à un objet est décrite de manière binaire, c'est-à-dire que l'élément peut soit appartenir entièrement ou ne pas appartenir du tout à l'ensemble. On a donc deux possibilités $\{0,1\}$. Dans le cas de la logique floue, un élément appartient à un ensemble avec un degrés d'appartenance, qui est compris dans l'intervalle $[0,1]$. Ces deux logiques ne s'opposent pas, au contraire la logique floue est une extension de la logique booléenne pour laquelle les degrés de vérité (degrés d'appartenance que l'on note μ), au lieu d'être vrai ou faux peuvent prendre des valeurs comprises entre 0 et 1[7].

3.1 Univers de discours

L'univers de discours est l'ensemble des valeurs réelles que peut prendre la variable floue x et $\mu_E(x)$ est le degré d'appartenance de l'élément x à l'ensemble floue E , cet ensemble flou est un sous ensemble de l'univers de discours [8].

3.2 Fonctions d'appartenances

A chaque degré d'appartenance est associée une valeur linguistique (par exemple : petit, moyen, grande). Chaque valeur linguistique est définie par une fonction d'appartenance, ces fonctions peuvent avoir plusieurs formes nous citerons pour l'exemple la forme triangulaire, sigmoïdale et trapézoïdale qui sont les plus utilisées [7].

3.3 Operateurs flous

Les opérateurs de la logique floue sont au nombre de trois, le ET le OU et le NON. Le ET peut être exprimé par la fonction Minimum ou la fonction multiplication. Le OU peut être exprimé par la fonction Maximum ou la fonction addition. Le NON est exprimé par l'expression :

(II.1)

$$\mu_{\bar{E}}(x) = 1 - \mu_E(x)$$

Avec $\mu_{\bar{E}}(x)$ la fonction d'appartenance de l'opérateur NON

3.4 Règles d'inférences floues

La prise de décision est basée sur les connaissances humaine ou de l'expert et est présentée sous forme de règles d'inférence floue qui s'expriment comme suit :

SI (prémisse) **ALORS** (conclusion)

Où

SI (prémisse) **ET** (prémisse) **ALORS** (conclusion)

Les prémisses sont des conditions (entrées) et la conclusion est l'action à effectuer (sorties).

3.5 Structure d'un régulateur flou

La structure conventionnelle d'un régulateur flou est représentée sur la figure (II.1), elle est proposée par Mamdani et se base sur une interface de fuzzification, inférence floue et défuzzification.

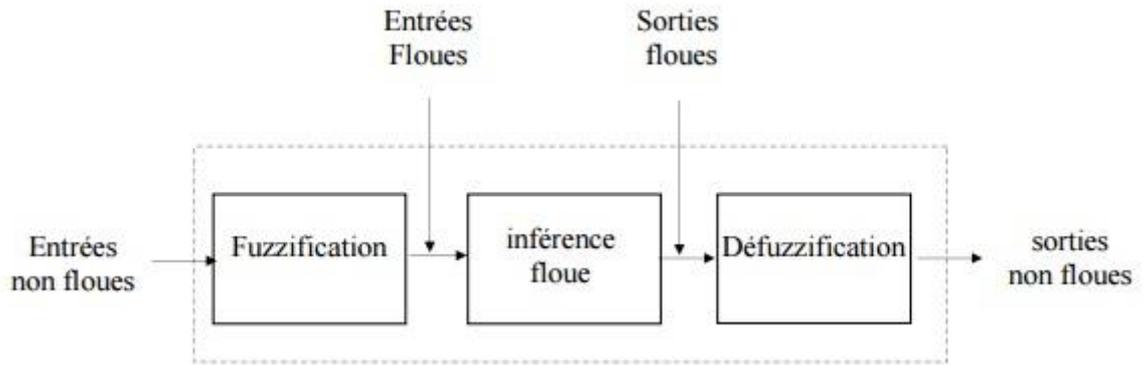


Figure II. 1 : structure d'un régulateur Mamdani

3.5.1 La fuzzification

Le but du module de fuzzification est la transformation des variables d'entrée en variable linguistique. Cette action s'effectue en définissant des fonctions d'appartenance pour les variables d'entrée.

3.5.2 L'inférence et règles floue

C'est là que sont appliquées les règles floues définies par le concepteur, elles se basent sur les connaissances et l'expérience du concepteur et sont la partie la plus importante. Le concepteur choisit aussi une méthode d'inférences parmi :

- La méthode max-min.
- La méthode max-produit.
- La méthode somme-produit.

Cette inférence sera appliquée aux différentes règles floues que le concepteur aura posées pour aboutir à une conclusion qui sera transmise au module de défuzzification.

3.5.3 Défuzzification

La défuzzification est l'étape de déduction d'une valeur numérique exploitable à partir de la conclusion obtenue grâce à la règle floue. Pour cela nous emploierons la méthode du centre de gravité ou méthode du centroid qui est donnée par la formule suivante [7] :

$$x_R = \frac{\int x \cdot \mu(x) dx}{\int \mu(x) dx} \quad (\text{II.2})$$

4. Application de la commande floue a l'UAV

En observant le modèle d'état, nous pouvons remarquer que certains états sont sensibles à certaines commandes. Par exemple : \dot{x}_2 dépend de U_2 , \dot{x}_4 dépend de U_3 , \dot{x}_6 dépend de U_4 , quant à \dot{x}_8, \dot{x}_{10} et \dot{x}_{12} ils dépendent tous de U_1 . De plus nous savons que $\dot{x}_2, \dot{x}_4, \dot{x}_6$ représentent respectivement $\dot{\varphi}$, $\dot{\theta}$ et $\dot{\psi}$ et que \dot{x}_{12} représente \dot{z} .

De plus nous savons que :

- U_1 Représente la puissance totale des moteurs.
- U_2 Représente la différence de puissance entre le moteur 2 et le moteur 4.
- U_3 Représente la différence de puissance entre le moteur 3 et le moteur 4.
- U_4 Représente la différence de puissance entre les moteurs tournant dans le sens des aiguilles d'une montre « horaire » et ceux tournant dans le sens opposé « anti horaire ».

En exploitant ces informations nous pouvons dès à présent concevoir un régulateur floue pour chacun des états $\dot{x}_8, \dot{x}_{10}, \dot{x}_{12}$ et \dot{x}_{12} . Le diagramme de commande est représenté par la figure 2.2

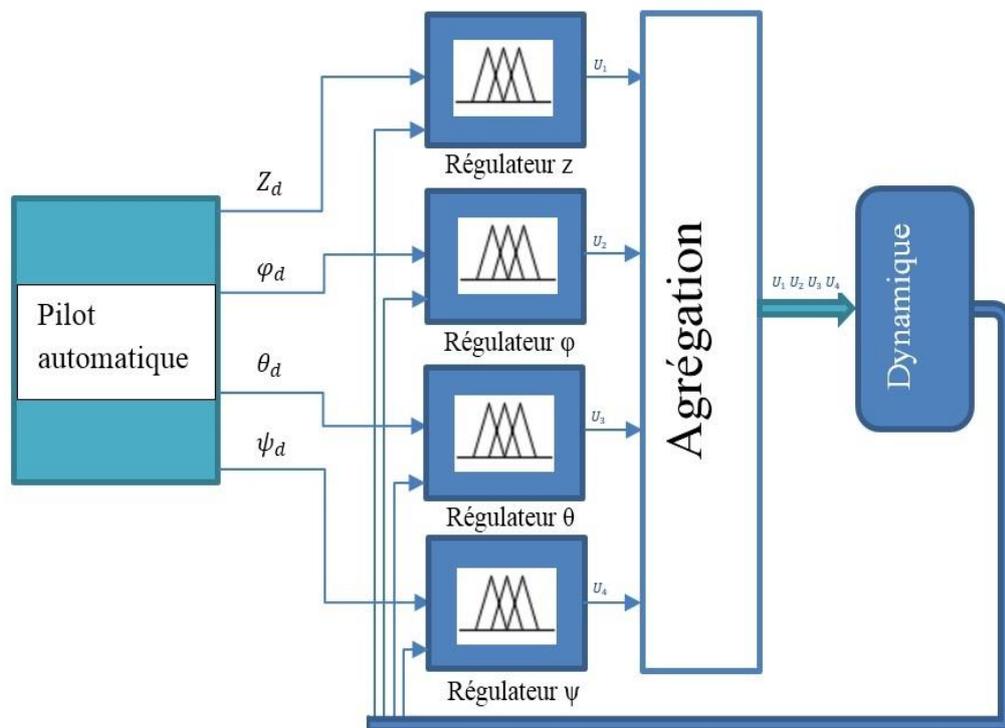


Figure II. 2 schéma de régulation flou Z, φ, θ, ψ

Nous avons opté pour trois fonctions d'appartenance triangulaires par variable d'entrée (pour permettre l'implémentation sur un système à faible puissance de calcul comme la carte Arduino Méga). Chaque régulateur reçoit en entrée l'erreur de position et la vitesse de déplacement (ou de rotation) et grâce à neuf règles d'inférences nous obtenons, en appliquant une inférence de type max-min, une sortie du régulateur à laquelle nous appliquons une défuzzification utilisant La méthode de centre de gravité.

4.1 Axe de translation Z

Pour la régulation sur l'axe de translation Z, nous avons pris en entrée la différence entre la consigne et la position (en mètre) sur cet axe et la vitesse de déplacement (en mètre par seconde). Après plusieurs tests et ajustements nous avons abouti aux fonctions d'appartenances pour les entrées représentées sur la figure suivante.

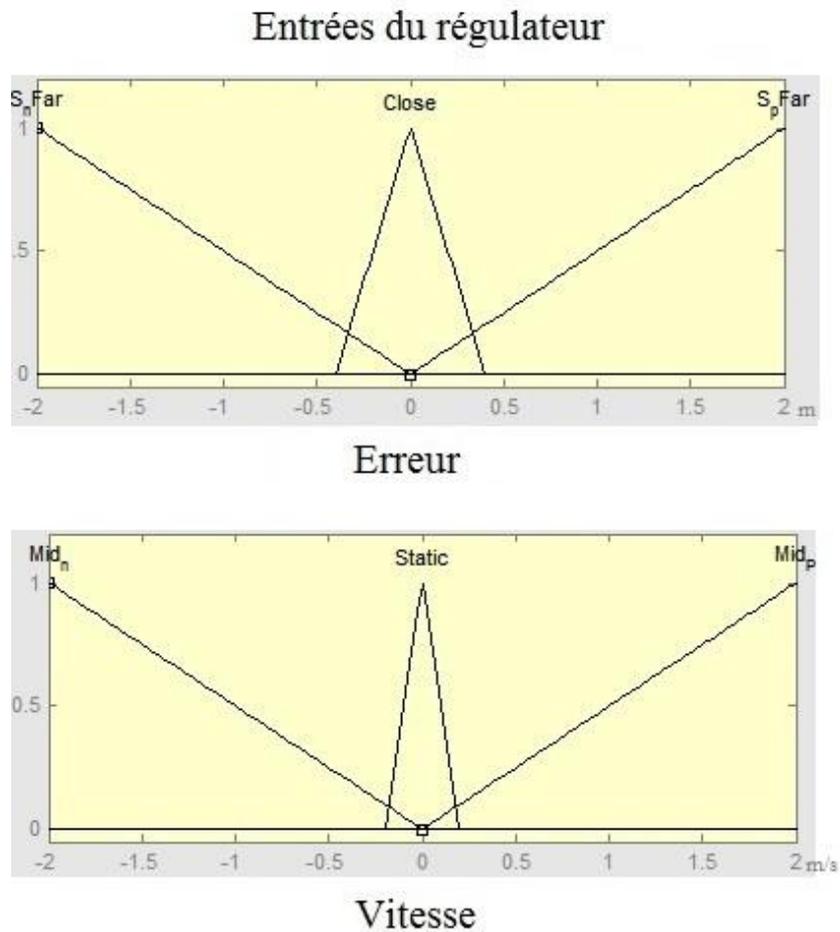


Figure II. 3 : entrées du régulateur floue de la position Z

Une fois les lois de commande floues appliquées et l'inférence appliqué nous obtenons une valeur floue en sorti, nous effectuons donc une défuzzification. La figure suivante représente la fonction d'appartenance de la sortie.

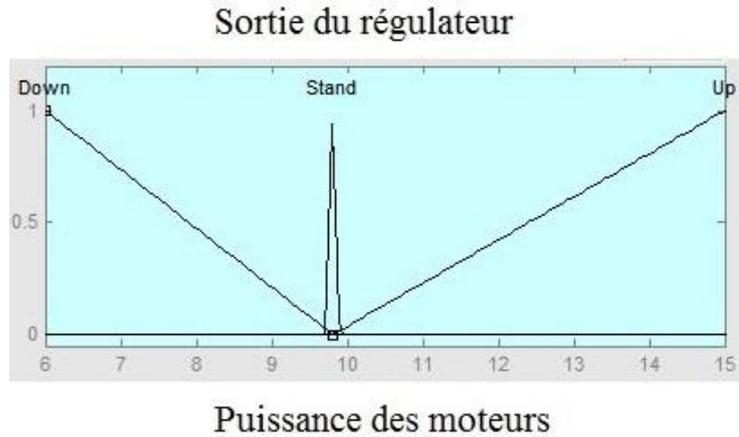


Figure II. 4 sortie du régulateur floue de la position Z

4.2 Axe de rotation φ

Pour la régulation sur l'axe de rotation φ , nous avons pris en entrée la différence entre la consigne et l'angle sur cet axe (en radian) et la vitesse de déplacement (en radian par seconde). Après plusieurs tests et ajustements nous avons abouti aux fonctions d'appartenances pour les entrées représentées sur la figure suivante.

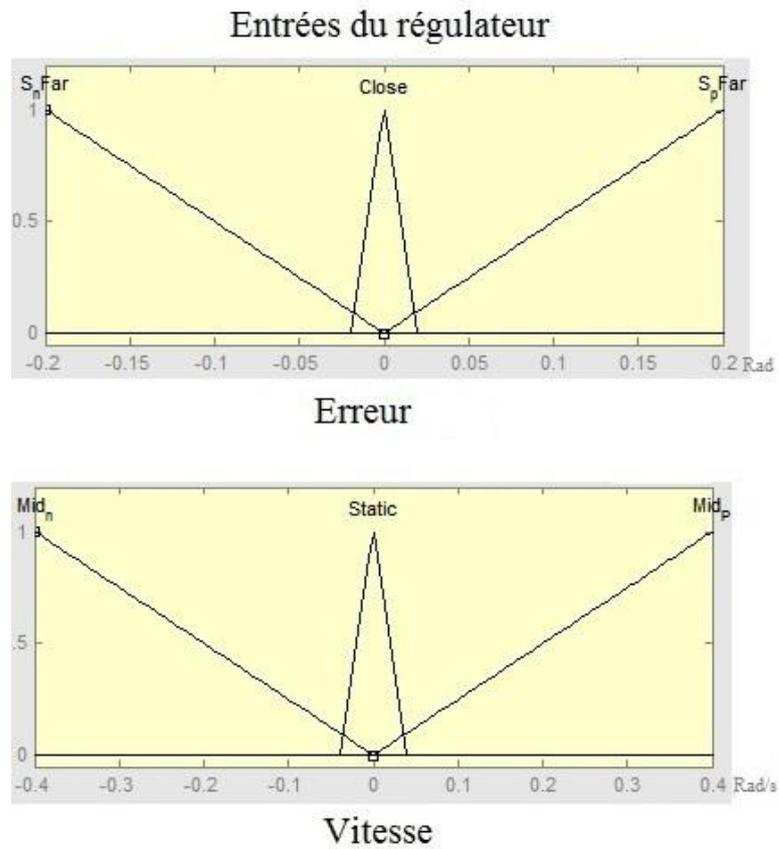


Figure II. 5 entrées du régulateur floue de l'angle Phi

Une fois les lois de commande floues appliquées et l'inférence appliqué nous obtenons une valeur floue en sorti, nous effectuons donc une défuzzification. La figure suivante représente la fonction d'appartenance de la sortie.

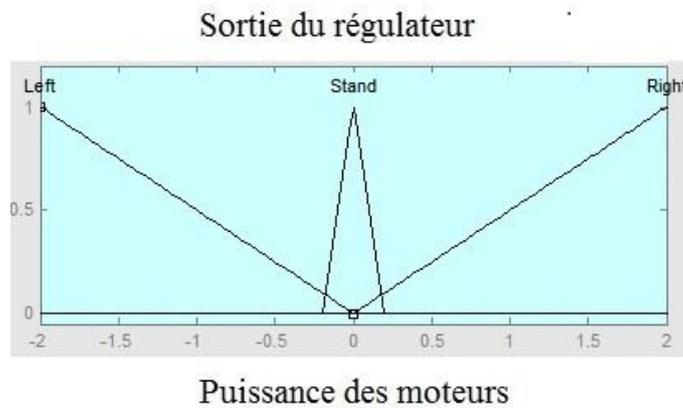


Figure II. 6 sortie du régulateur floue de l'angle Phi

4.3 Axe de rotation θ

Pour la régulation sur l'axe de rotation θ , nous avons pris en entrée la différence entre la consigne et l'angle sur cet axe (en radian) et la vitesse de déplacement (en radian par seconde). Après plusieurs tests et ajustements nous avons abouti aux fonctions d'appartenance pour les entrées représentées sur la figure suivante.

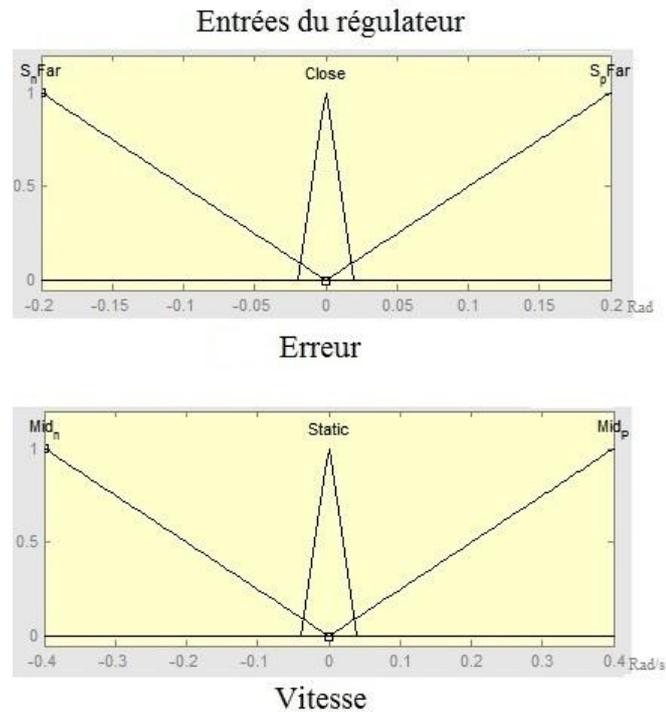


Figure II. 7 : entrées du régulateur floue de l'angle theta

Une fois les lois de commande floues appliquées et l'inférence appliqué nous obtenons une valeur floue en sorti, nous effectuons donc une défuzzification. La figure suivante représente la fonction d'appartenance de la sortie.

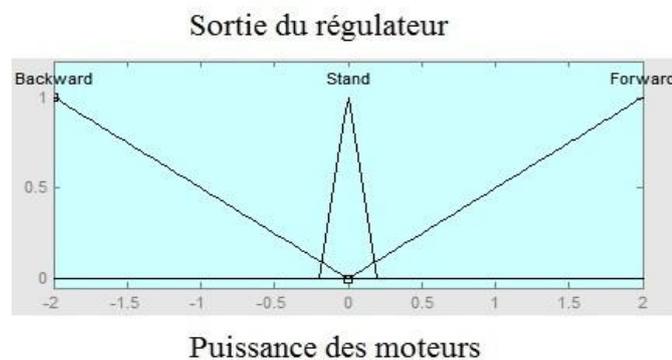


Figure II. 8 : sortie du régulateur floue de l'angle theta

4.4 Axe de rotation ψ

Pour la régulation sur l'axe de rotation ψ , nous avons pris en entrée la différence entre la consigne et l'angle sur cet axe (en radian) et la vitesse de déplacement (en radian par seconde). Après plusieurs tests et ajustements nous avons abouti aux fonctions d'appartenance pour les entrées représentées sur la figure suivante.

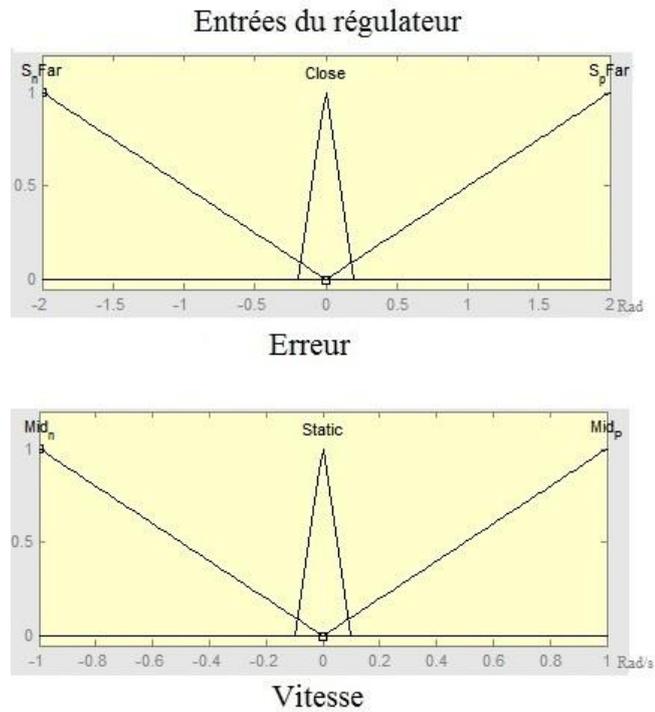


Figure II. 9 : entrées du régulateur floue de l'angle Ψ

Une fois les lois de commande floues appliquées et l'inférence appliqué nous obtenons une valeur floue en sorti, nous effectuons donc une défuzzification. La figure suivante représente la fonction d'appartenance de la sortie.

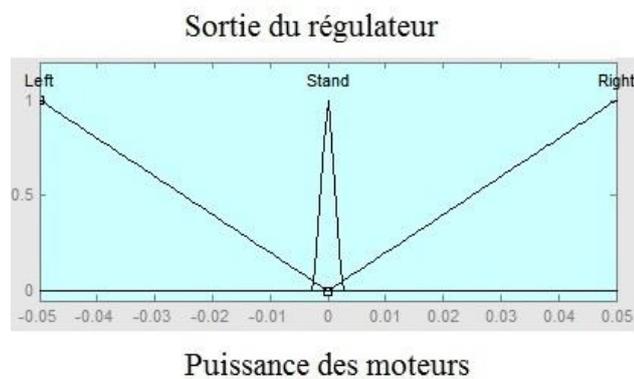


Figure II. 10 : sortie du régulateur floue de l'angle Ψ

5. Simulation de la commande sur Matlab-Simulink

Grace à une simulation, qui se base sur le modèle obtenu lors du chapitre précédent, nous pouvons effectuer des tests qui sont représentatifs de la réalité.

5.1 Simulation de la dynamique de vol

La figure suivante représente le sous-système qui simule la dynamique de vol. Nous avons utilisé un bloc « Matlab function » pour regrouper toutes les lois régissant le système, nous avons en entrée du bloc l'état précédent ($t-1$) et la commande actuel (t) et en sortie nous avons W , un vecteur qui représente la vitesse de rotation des rotors et dx que nous injectons dans un bloc intégrateur (nous avons séparé les variables pour pouvoir les étudier individuellement).

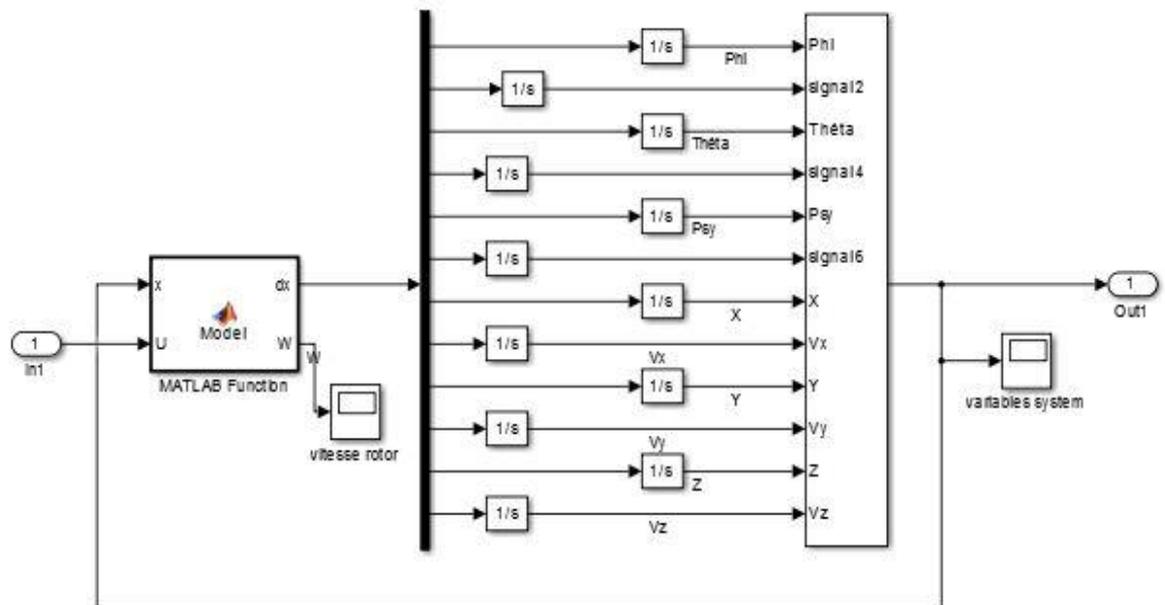


Figure II. 11 : schéma Simulink de la dynamique du quadrirotor

5.2 Simulation des boucles de régulation

Nous avons commencé par créer les régulateurs flous grâce à l'outil « fuzzy » de Matlab, puis nous avons réalisé le schéma sur Simulink. Nous avons donc intégré le sous-système de la dynamique de vol que nous avons mis dans un bloc de sous-système, en entrée du sous-système nous avons les sorties des régulateurs flous et en sortie nous avons les douze variables de notre représentation d'état. Nous observons les angles (après les avoir convertie en degré) et les positions. En entrée des régulateurs nous injectons l'erreur statique et la vitesse instantanée.

Chapitre II : Réalisation d'une commande floue pour un drone de type quadrirotor.

Pour générer les consignes nous utilisons en fonction du besoin, un générateur de signal aléatoire ou un générateur d'échelon. Ces signaux sont utilisés pour calculer l'erreur statique et est observé sur le même graphe que l'état correspondant.

Nous récupérons ensuite les données puis nous les exploitons sur office-Excel pour dessiner les graphes illustrant les résultats.

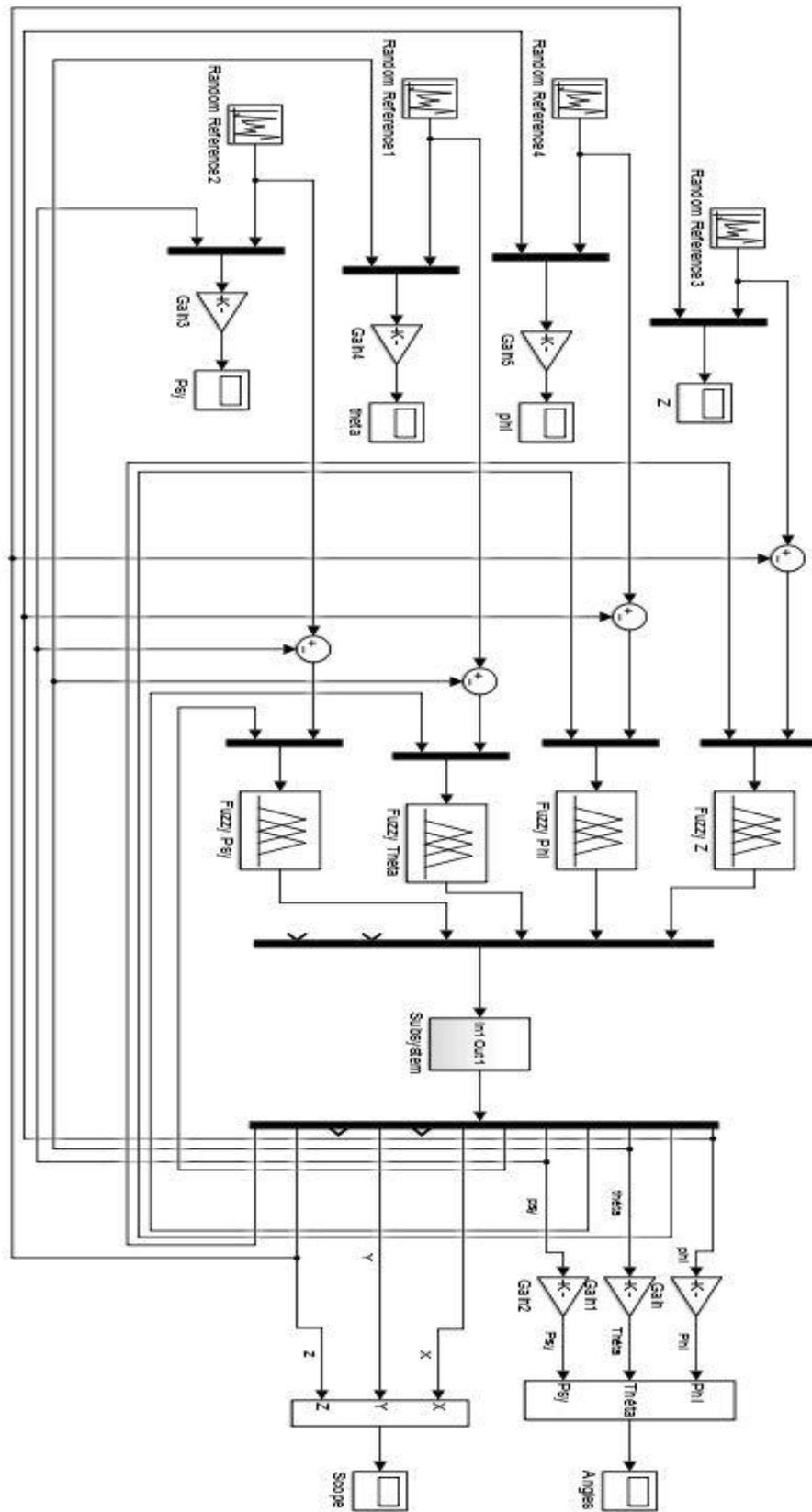


Figure II. 12 : simulation sur Simulink de la régulation floue Z, ϕ, θ, ψ

Nous allons présenter dans ce qui suit les résultats des tests effectués à l'aide de Matlab Simulink.

6. Résultats de la simulation

Nous avons réalisé plusieurs tests de poursuites de consigne et de robustesse face aux perturbations, les résultats obtenus sont présentés dans ce qui suit.

6.1 Poursuite d'un échelon

Le premier test auquel nous soumettons notre système est un test de poursuite d'échelon. Les résultats pour les boucles de régulation sont présentés sur la figure suivante.

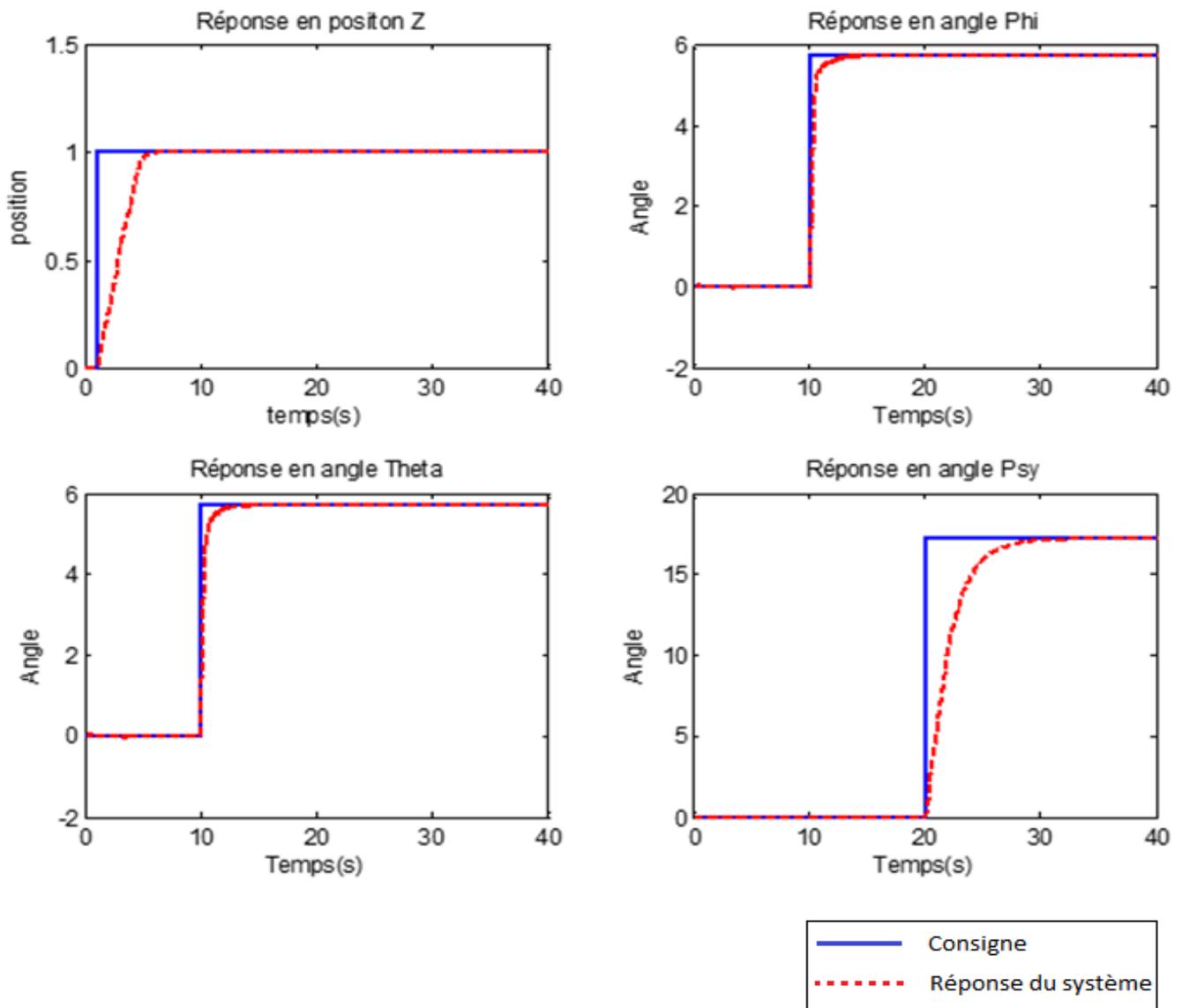


Figure II. 13 : Réponse à un échelon de la régulation floue sur les variables Z, Phi, Theta, Psi

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

- Le temps de réponse en Z est de 6.63s.
- Le temps de réponse en Phi est de 1.15s.
- Le temps de réponse en Thêta est de 1.2s.
- Le temps de réponse en psy est de 5.85s.

Nous observons aussi une poursuite précise de la consigne et aucun dépassement.

6.2 Poursuite de consignes aléatoire

Nous soumettons ensuite le système à une série de consignes aléatoires. Pour réaliser cette simulation, nous avons utilisé un générateur de signal délivrant un signal aléatoire dans un intervalle que l'on spécifie et avec une fréquence réglable. Ainsi nous obtenons les résultats suivant :

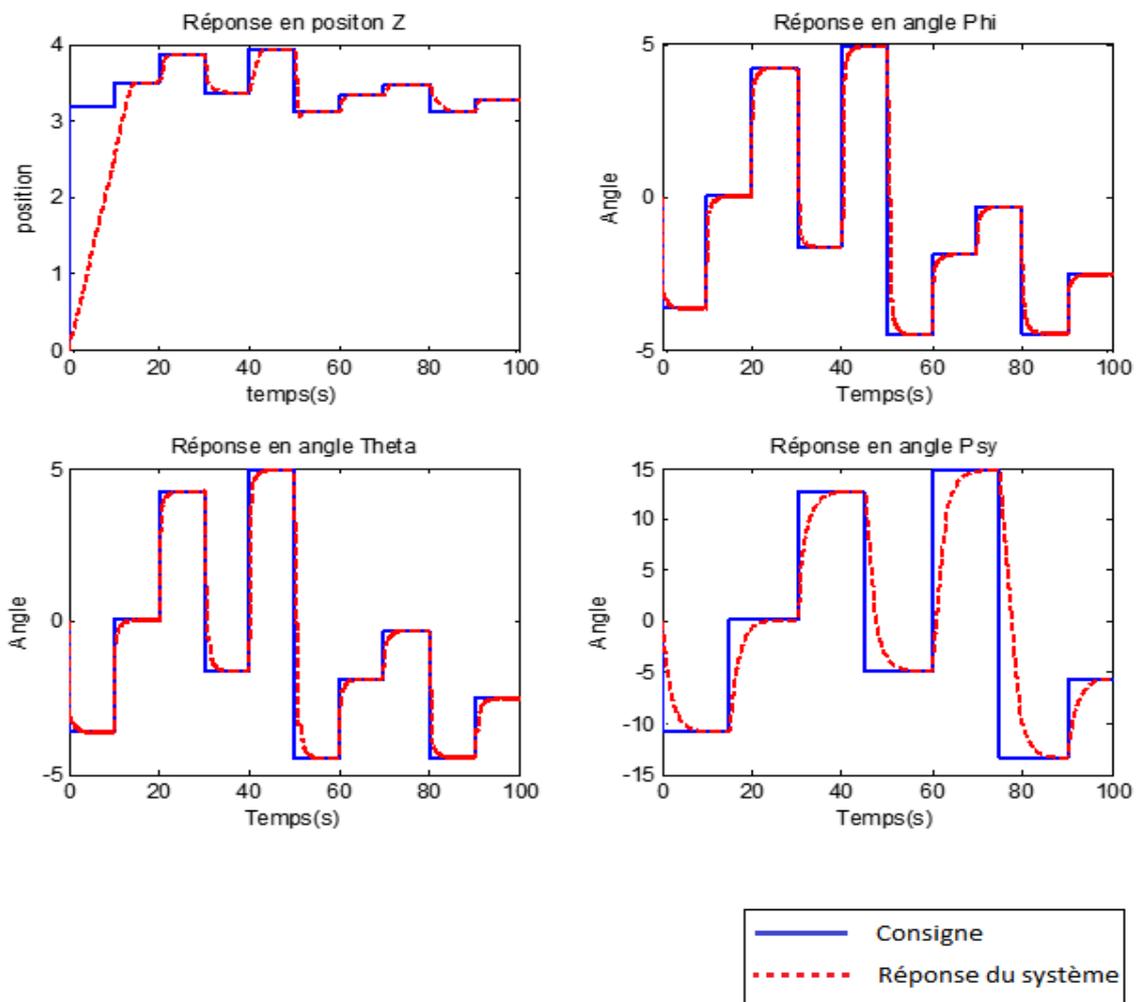


Figure II. 14 : Réponse à des consignes aléatoires de la régulation floue sur les variables Z, Phi, Theta, Psy

6.3 Poursuite de consigne sinusoïdale

Dans le prochain test nous mettons à l'épreuve notre système commandé en lui donnant en consigne un signal sinusoïdal. Les résultats sont représentés sur la figure suivante :

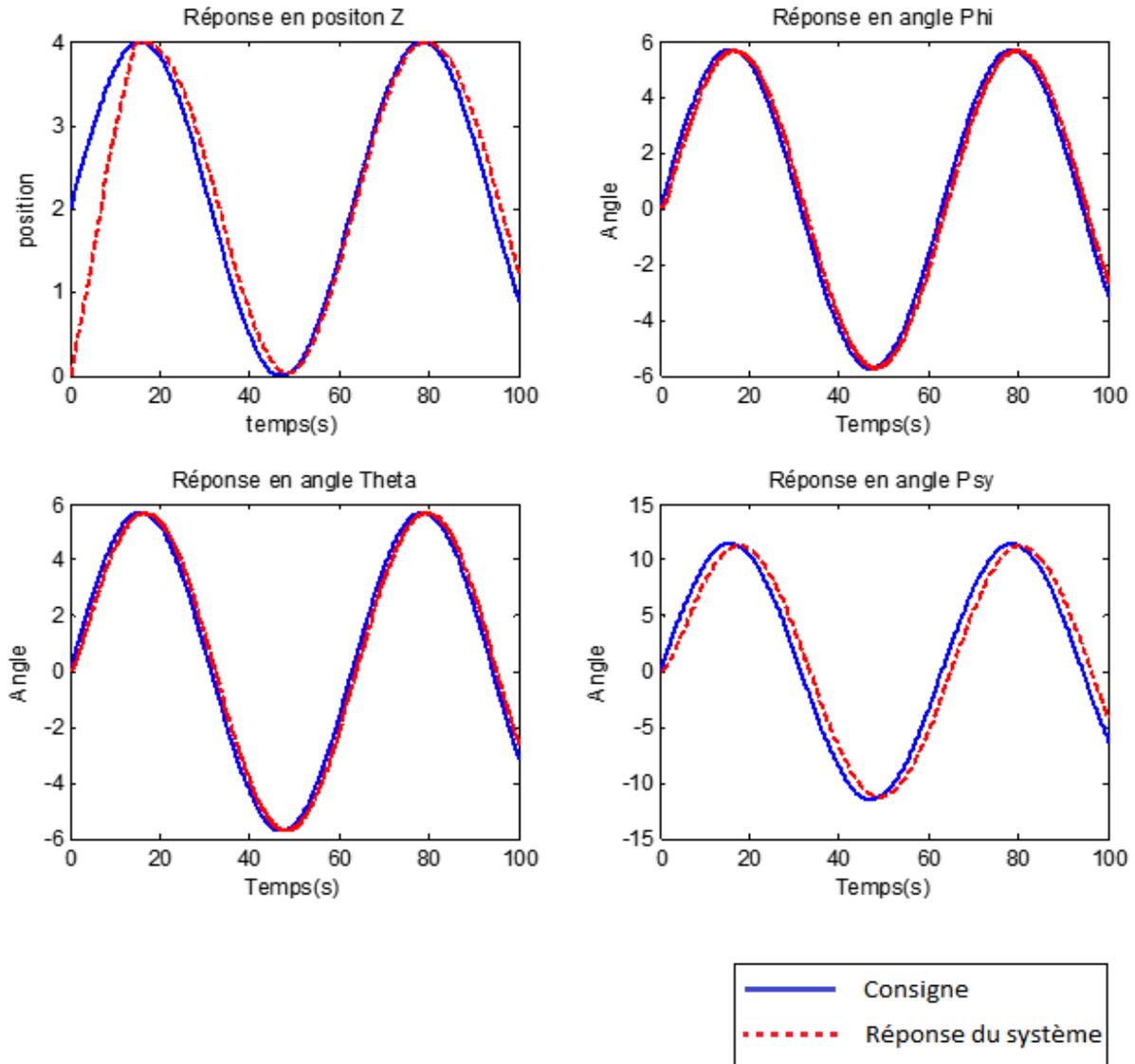


Figure II. 15 : Réponse à une consigne sinusoïdale de la régulation floue sur les variables Z, Phi, Theta, Psi

Nous observons une bonne poursuite de la consigne avec toutefois un petit décalage.

6.4 Régulation face à des perturbation

Pour réaliser ce dernier test nous allons fixer la consigne a une valeur constante et nous allons injecter des perturbations instantanées et a valeurs aléatoires. Les résultats sont présentés sur la figure suivante :

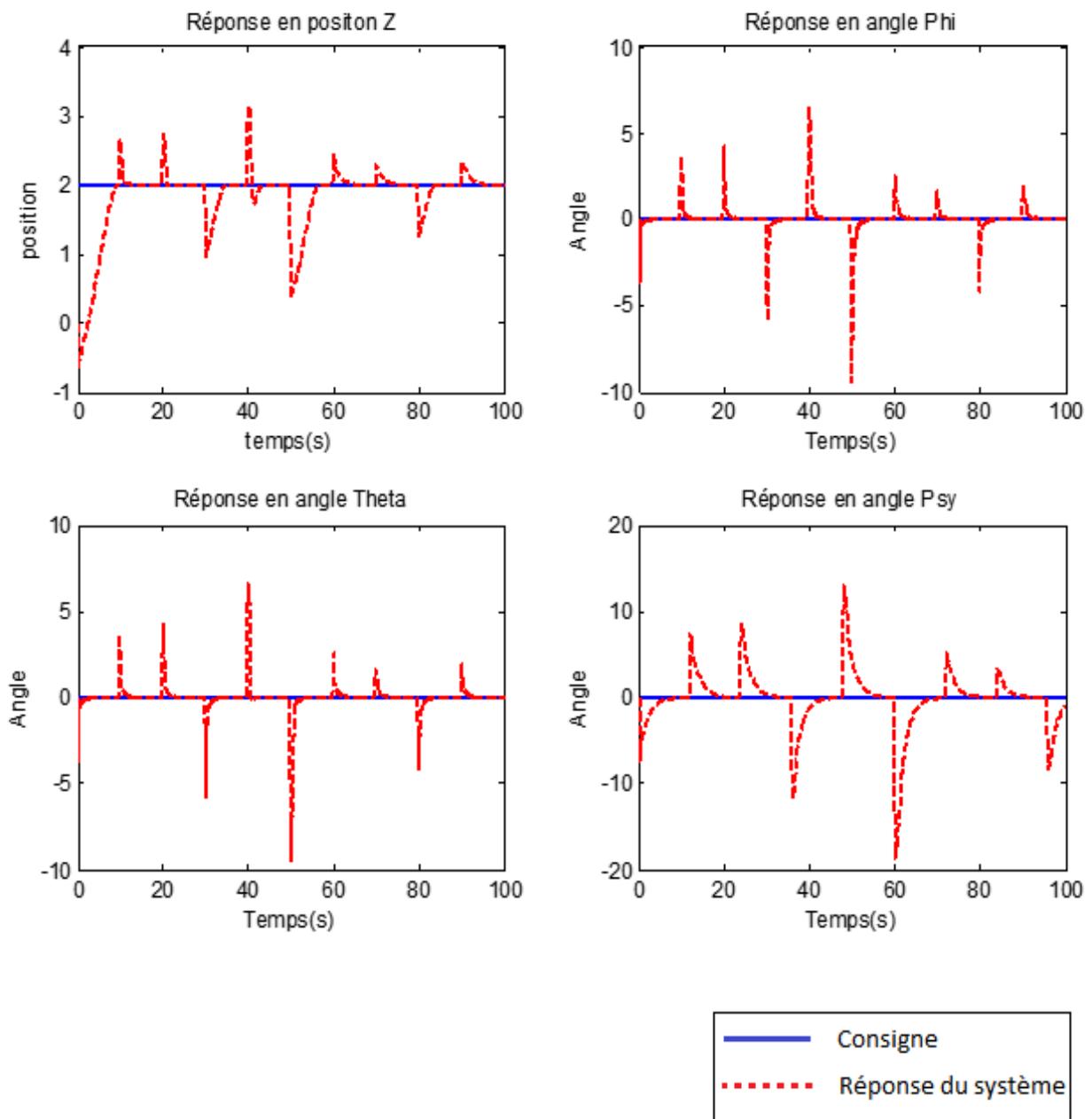


Figure II. 16 : Réponse face à une perturbation de la régulation floue sur les variables Z, Phi, Theta, Psi

Nous observons clairement que le système rejette toutes les perturbations.

7. Régulation des axes X et Y

Après avoir réalisé une boucle de régulation pour les deux axes de rotation θ et φ et en ayant observé des résultats très satisfaisant, nous avons décidé de réaliser une double boucle de régulation sur les deux axes de translation X et Y, en prenant comme première boucle la régulation des angles respectivement θ et φ , nous obtenons le schéma de régulation suivant.

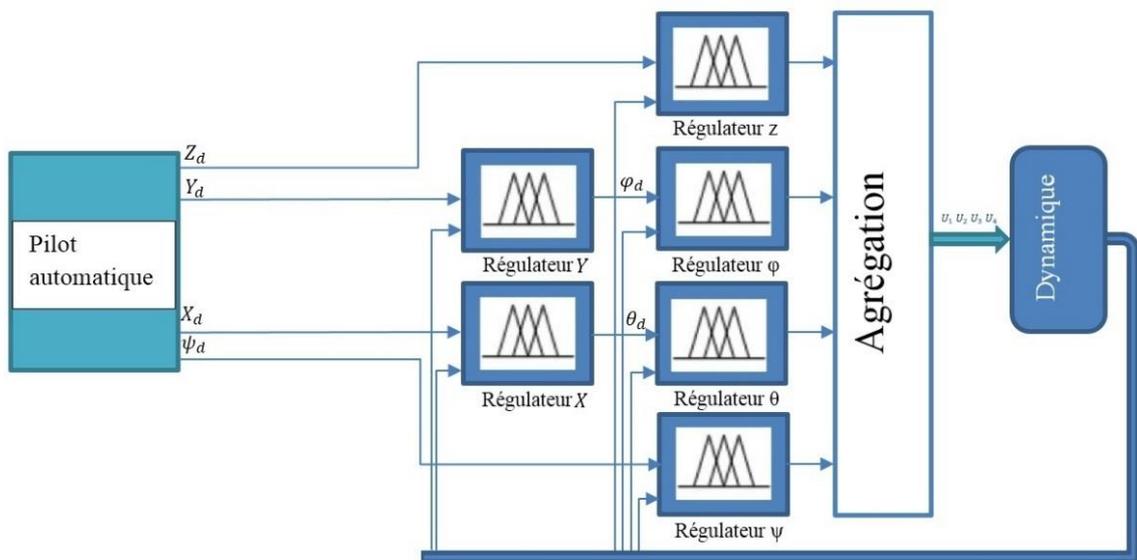


Figure II. 17 : schéma de régulation flou X, Y

7.1 Axe de translation X

Comme pour la régulation précédente, nous avons opté pour l'erreur statique (mètre) et la vitesse de déplacement (mètre/seconde) comme entrées du régulateur. A bout de plusieurs tests et réglages, nous avons fixé les fonctions d'appartenances suivantes.

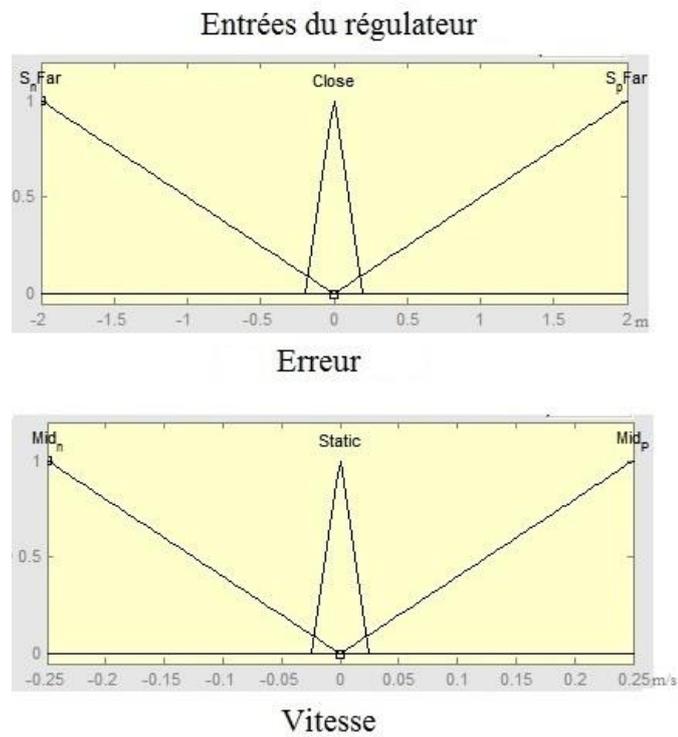


Figure II. 18 : entrées du régulateur floue de la position X

Une fois les lois de commande floues appliquées et l'inférence appliqué nous obtenons une valeur floue en sorti, nous effectuons donc une défuzzification. La figure suivante représente la fonction d'appartenance de la défuzzification.

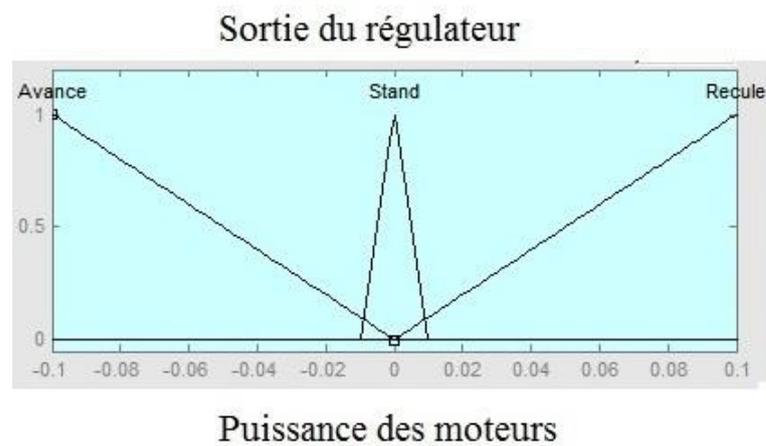


Figure II. 19 sortie du régulateur floue de la position X

7.2 Axe de translation Y

Encore une fois, nous avons opté pour l'erreur statique(mètre) et la vitesse de déplacement (mètre/seconde) comme entrées du régulateur. A bout de plusieurs tests et réglages, nous avons fixé les fonctions d'appartenances suivantes.

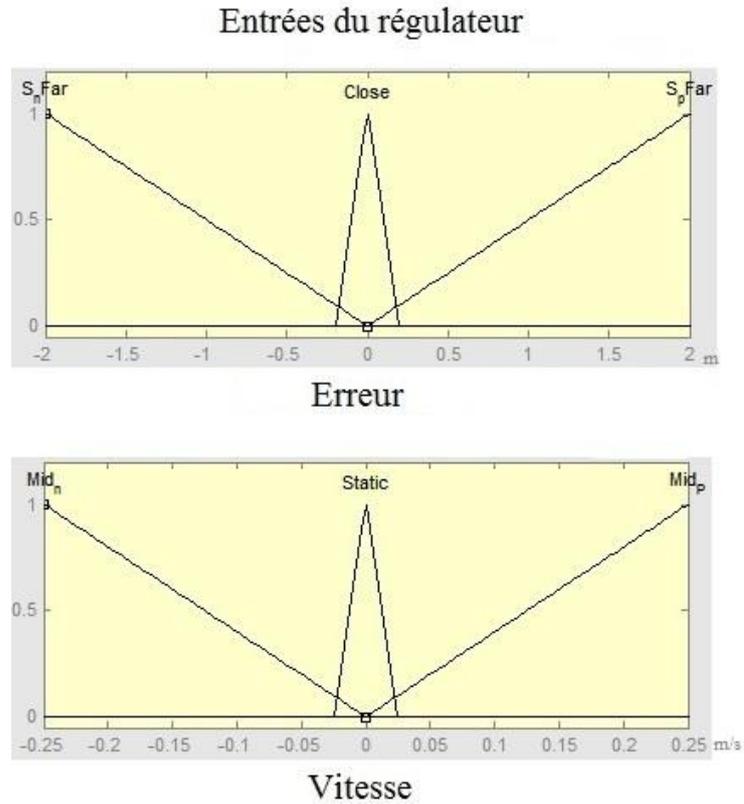


Figure II. 20 : entrées du régulateur floue de la position Y

Une fois les lois de commande floues appliquées et l'inférence appliqué nous obtenons une valeur floue en sorti, nous effectuons donc une défuzzification. La figure suivante représente la fonction d'appartenance de la défuzzification.

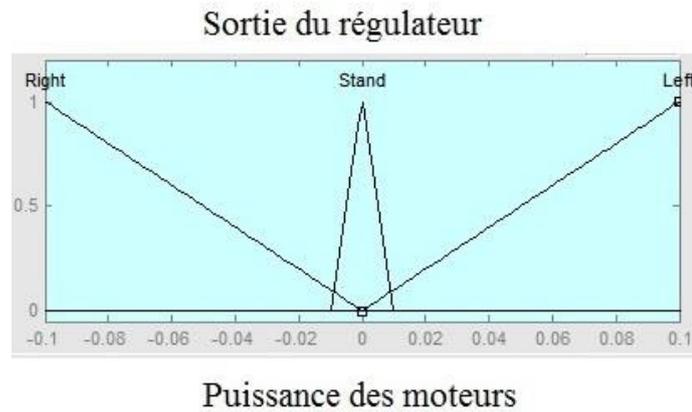


Figure II. 21 : sortie du régulateur floue de la position Y

8. Simulation sur Matlab-Simulink

Pour réaliser cette seconde simulation, nous nous sommes basé sur le système réaliser précédemment auquel nous avons intégré deux nouvelles boucles de régulation. Ces boucles sont des doubles boucles de régulation ayant pour boucles internes la régulation des angles correspondant, ainsi le boucle de régulation de la position X a pour boucle interne la régulation de l'angle θ et le boucle de régulation de la position Y a pour boucle interne la régulation de l'angle φ .

Nous pouvons observer sur la figure suivante la structure de cette régulation sur Simulink

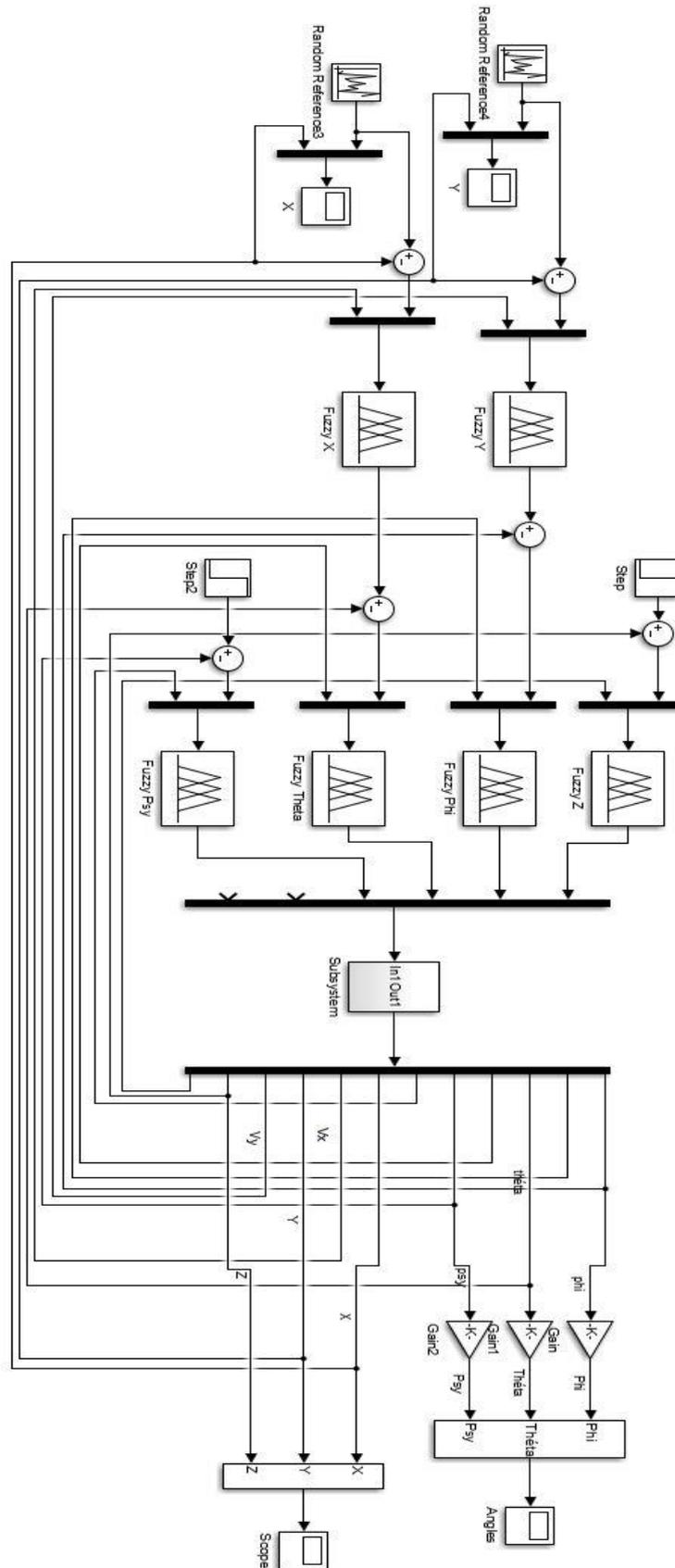


Figure II. 22 : simulation sur Simulink de la régulation flou X, Y

9. Résultats de la simulation

Nous avons réalisé plusieurs tests de poursuites de consigne et de robustesse face aux perturbations, les résultats obtenus sont présentés dans ce qui suit.

9.1 Poursuite d'un échelon

Le premier test auquel nous soumettons notre système c'est un test de poursuite d'échelon. Les résultats pour les boucles de régulation sont présentés sur la figure suivante.

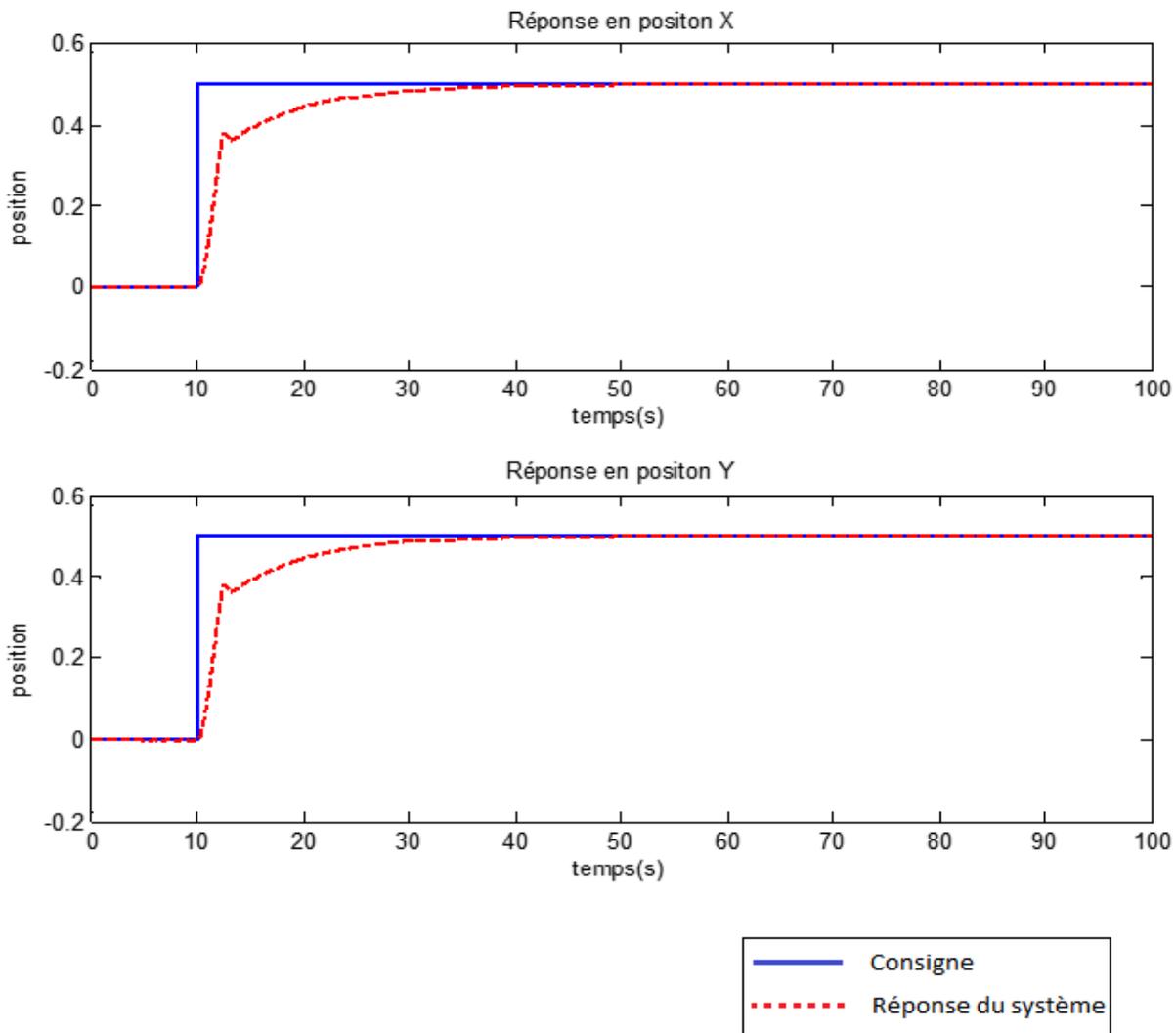


Figure II. 23 : Réponse à un échelon de la régulation floue sur les variables X et Y

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

- Le temps de réponse en X est de 13.3s.
- Le temps de réponse en Y est de 13.9s.

Nous observons aussi une poursuite précise de la consigne et aucun dépassement mais un temps de réponse relativement lent.

9.2 Poursuite de consignes aléatoire

Nous soumettons ensuite le système a une série de consignes aléatoires. Pour réaliser cette simulation, nous avons utilisé un générateur de signal délivrant un signal aléatoire dans un intervalle de $[-1 ; 1]$ et avec une fréquence de $1/30\text{Hz}$. Ainsi nous obtenons les résultats suivant :

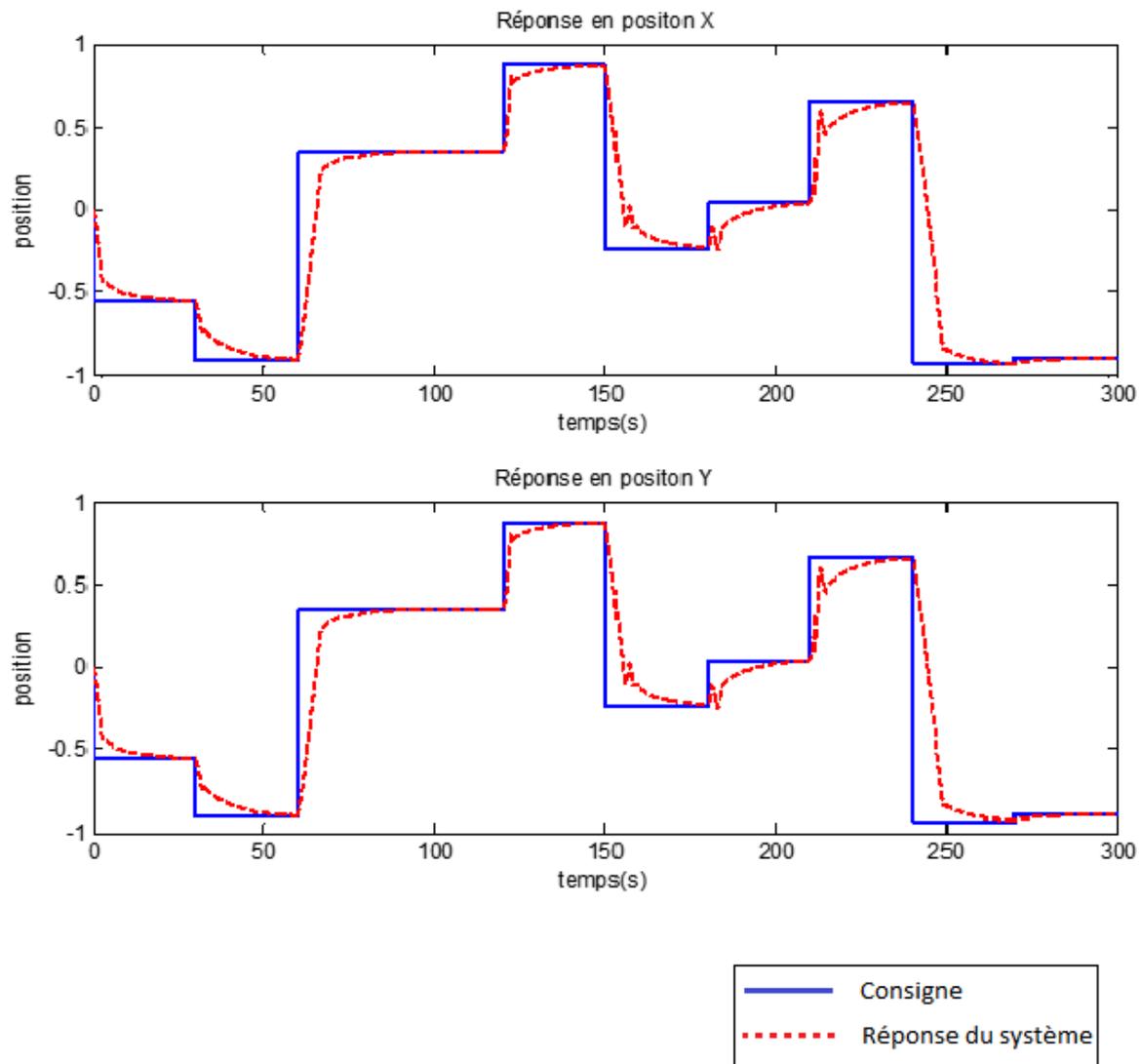


Figure II. 24 : Réponse à des consignes aléatoires de la régulation floue sur les variables X et Y

9.3 Poursuite de consigne sinusoïdale

Dans le prochain test nous mettons à l'épreuve notre système commandé en lui donnant en consigne un signal sinusoïdal. Les résultats sont représentés sur la figure suivante :

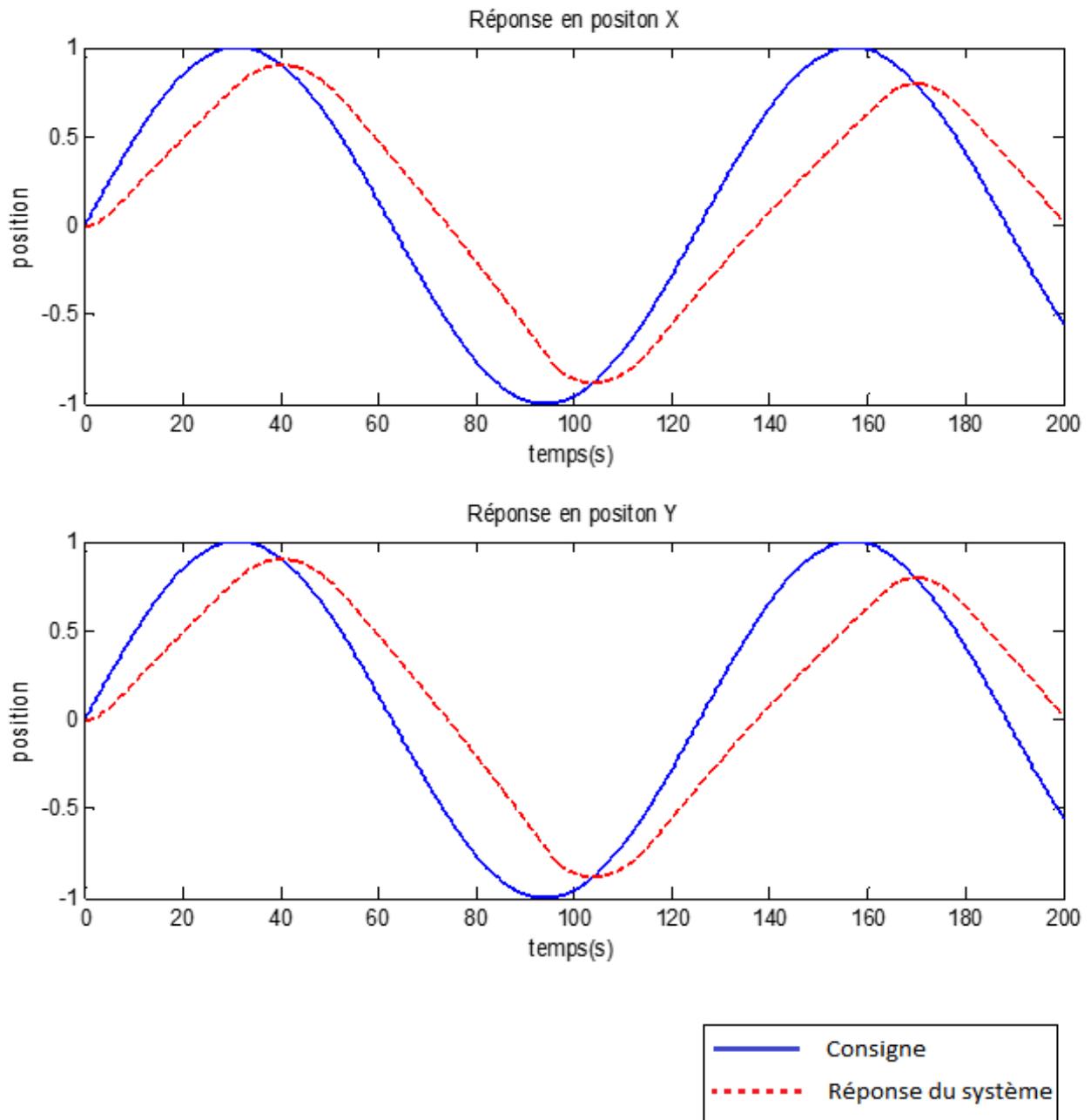


Figure II. 25 : Réponse à une consigne sinusoïdale de la régulation floue sur les variables X et Y

Nous observons une poursuite de la consigne mais avec un décalage non négligeable.

9.4 Régulation face à des perturbation

Pour réaliser ce dernier test nous allons fixer la consigne a une valeur constante et nous allons injecter des perturbations instantanées et a valeurs aléatoires. Les résultats sont présentés sur la figure suivante :

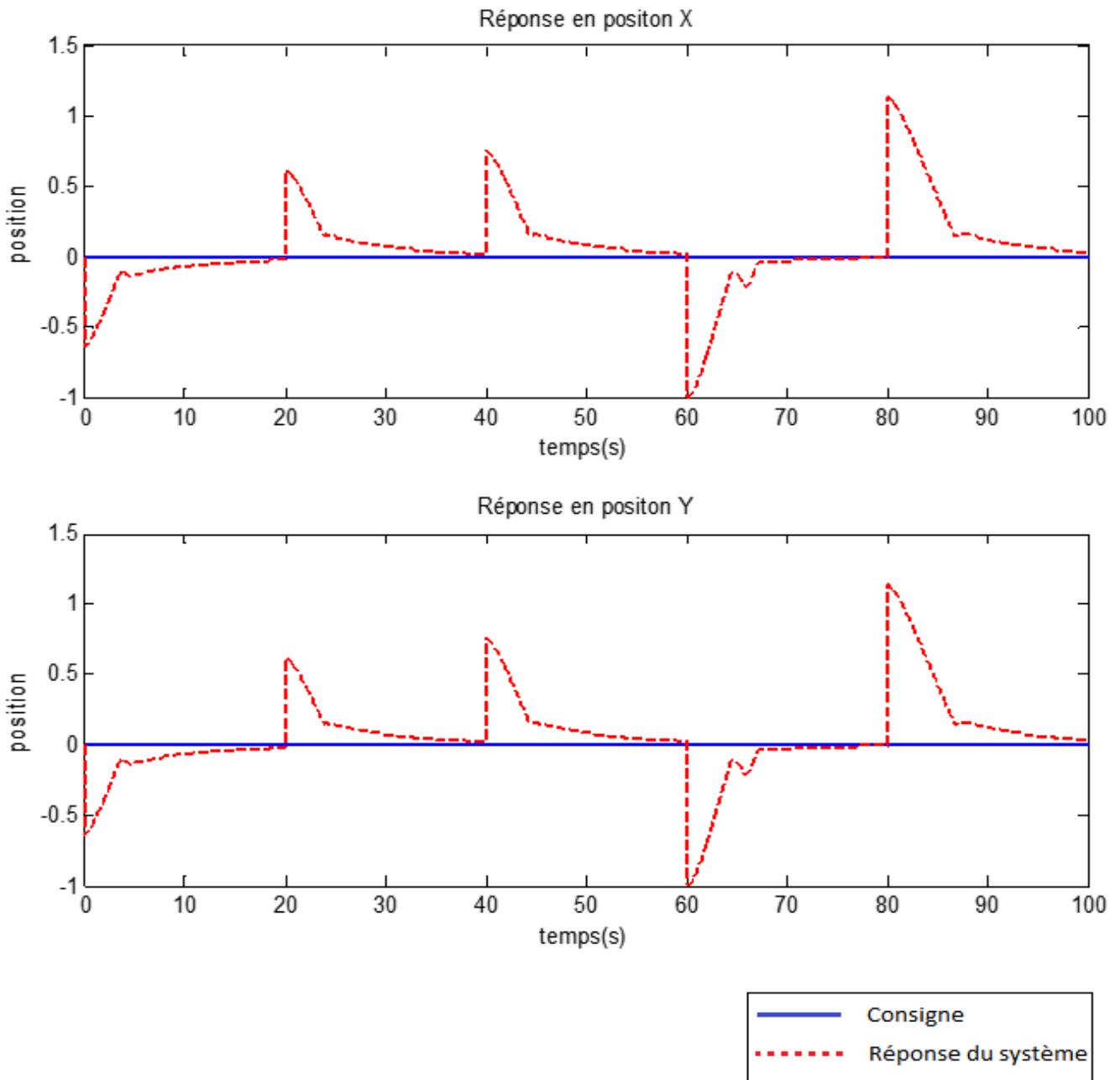


Figure II. 26 : Réponse face à une perturbation de la régulation floue sur les variables X et Y

Nous voyons bien que le système rejette toutes les perturbations.

10. Conclusion

Lors de ce chapitre nous avons présenté le principe utiliser par la logique floue, puis nous avons entreprit de concevoir une commande pour notre système. Tout d'abord nous avons réalisé des

commandes sur une seule boucle pour les quatre paramètres principaux, Z , φ , θ et ψ . Chacune de ces commandes à donner des résultats amplement satisfaisant, autant en asservissement et poursuite de consigne qu'en régulation et robustesse face aux perturbations. Nous avons donc poussé la conception plus loin en concevant pour les paramètres X et Y une double boucle de commande exploitant respectivement les boucles des paramètres θ et φ . Ces commandes ont montré de bons résultats en asservissement et poursuite de consigne et tout autant en matière de régulation et robustesse face aux perturbations. Lors du chapitre suivant nous allons tenter de réaliser une commande pour notre système à l'aide de régulateur PID et nous essayerons d'optimiser les paramètres de ces régulateurs afin d'obtenir les meilleurs résultats possibles.

Chapitre III

Réalisation d'une commande par PID pour un drone de type quadrirotor

1. Introduction

Lors des chapitres précédents, nous avons abouti à un model dynamique du quadrirotor, en exploitant les lois physiques régissant la dynamique de vol de ce système et nous avons conçu une régulation à base de règles floues que nous avons simulé et tester. Le but de ce chapitre est d'asservir ce système en utilisant une autre méthode de régulation à l'aide d'un contrôleur PID.

2. Principe

La régulation par contrôleur PID exploites trois actions principales, l'action proportionnelle, l'action intégrale et l'action dérivée. Ces actions ont chacune leurs avantage et inconvénient et leur action combinée permet d'avoir une certaine compensation des inconvénients tout en gardant les avantages et ainsi aboutir à des performances intéressantes.

2.1 Action proportionnelle

L'effet de l'action proportionnelle consiste à amplifier l'erreur d'un gain constant afin que le système réagisse plus rapidement aux changements de consigne. Cette action proportionnelle est représentée comme suit :

$$C(t) = K_p \cdot \varepsilon(t) \quad (\text{III.1})$$

Plus la valeur de K_p est grande plus la réponse est rapide mais au détriment d'une détérioration de la stabilité du système allant jusqu'à l'instabilité pour de grandes valeurs.

2.2 Action intégrale

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

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

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

2.3 Action dérivée

Pour obtenir une action dérivée nous multiplions la dérivée de l'erreur par un coefficient K_d cette action permet d'éliminer le dépassement de la réponse et d'améliorer la stabilité du système. Sa relation est donnée comme suit :

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

Plus la valeur de K_d est grande plus le dépassement est atténué mais si elle est trop grande le système est ralenti jusqu'à risquer de devenir instable pour de très grandes valeurs.

2.4 Action PID

Le régulateur PID combine les trois actions vues précédemment et permet ainsi d'avoir de bonnes performances aussi bien en vitesse en stabilité qu'en précision. L'expression d'un régulateur PID est donné comme suit :

$$C(t) = K_p \cdot \varepsilon(t) + K_i \cdot \int_0^t \varepsilon(t) dt + K_d \cdot \frac{d\varepsilon(t)}{dt} \quad (\text{III.4})$$

Ainsi ce régulateur permet d'atteindre les objectifs fixés en termes de vitesse de stabilité et de précision et ce en trouvant la configuration optimale des valeurs des différents gains K_p , K_i et K_d .

3. Choix des paramètres et optimisation

Le choix des paramètres est crucial pour le bon fonctionnement de notre système nous avons donc commencer par chercher des paramètres en effectuant plusieurs essais pour amener notre système à un état stable puis nous avons effectué une optimisation de ces paramètres pour minimiser l'erreur.

3.1 Schéma de régulation

Nous avons choisi d'utiliser la même structure de régulation que celui utilisé lors de la régulation par contrôleur flou, nous avons ainsi deux boucles de régulation simple utilisées pour commander les variables Z et ψ et deux doubles boucles de régulation qui commandent la position X à l'aide de l'angle θ et la position Y à l'aide de l'angle φ . Le schéma est donné comme suit :

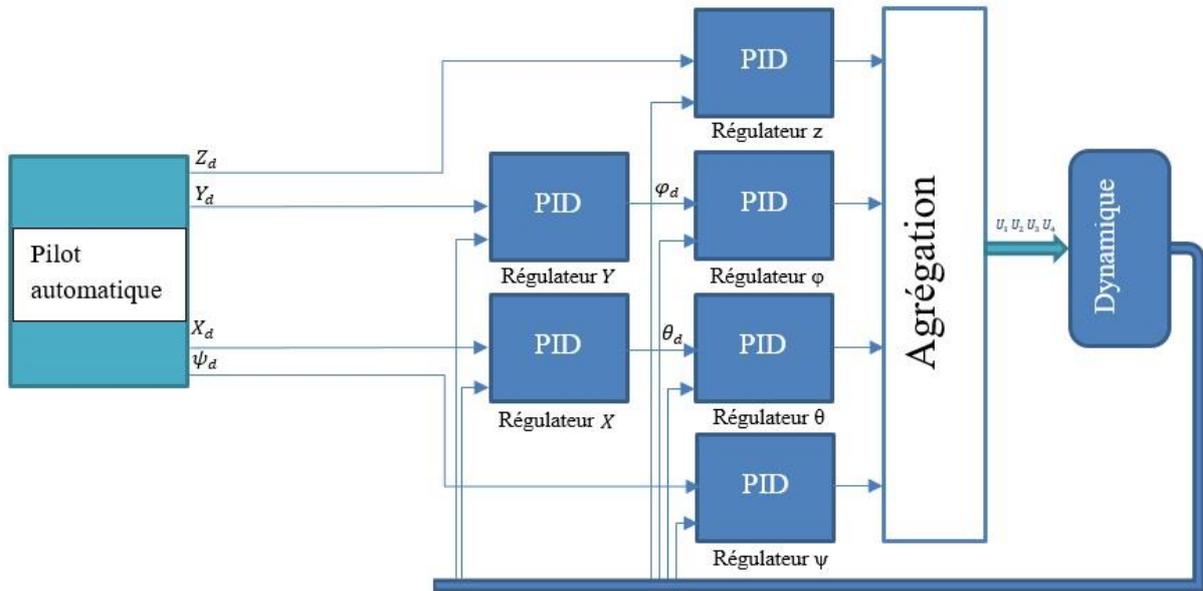


Figure III. 1 : schéma de régulation PID.

Nous avons donc au totale six régulateurs PID ce qui fait 18 paramètres à définir.

3.2 Choix de paramètres initiaux

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

Régulateur	Paramètres
Régulateur Z	$K_{pz} = 5$
	$K_{iz} = 2$
	$K_{dz} = 4,5$
Régulateur ϕ	$K_{p\phi} = 2,8$
	$K_{i\phi} = 0,7$
	$K_{d\phi} = 2,9$
Régulateur θ	$K_{p\theta} = 2,8$
	$K_{i\theta} = 0,7$
	$K_{d\theta} = 2,9$

Régulateur ψ	$K_{p\psi} = 2,8$
	$K_{i\psi} = 0,7$
	$K_{d\psi} = 2,9$
Régulateur X	$K_{px} = 6,7$
	$K_{ix} = 0,6$
	$K_{dx} = 3,4$
Régulateur Y	$K_{py} = 6,7$
	$K_{iy} = 0,6$
	$K_{dy} = 3,4$

Tableau III. 1 paramètres du régulateur PID avant optimisation

3.3 Optimisation

Pour réaliser une optimisation des paramètres des régulateurs vu précédemment nous avons à notre disposition plusieurs méthodes d'optimisation, nous avons choisis de réaliser une minimisation de l'erreur quadratique. Nous avons donc posé le problème d'optimisation comme suit :

$$\min_{pid} E(t) = \sum_{i=1}^6 \int_0^t e_i^2(t) \quad (III.5)$$

Avec

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

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

$$e_i(t) = Y_{di} - Y_i \quad (III.5)$$

Avec $i = \{1,2,3,4,5,6\}$.

Ainsi les résultats obtenus sont résumés dans le tableau suivant :

Régulateur	Paramètres
Régulateur Z	$K_{pz} = 128,6283$
	$K_{iz} = 10,8637$
	$K_{dz} = 44,9601$
Régulateur ϕ	$K_{p\phi} = 7,1957$
	$K_{i\phi} = 0,6841$
	$K_{d\phi} = 0,5974$
Régulateur θ	$K_{p\theta} = 7,2036$
	$K_{i\theta} = 0,6861$
	$K_{d\theta} = 0,5973$
Régulateur ψ	$K_{p\psi} = 5,3620$
	$K_{i\psi} = 0,1000$
	$K_{d\psi} = 0,3352$
Régulateur X	$K_{px} = 6,8594$
	$K_{ix} = 0,1146$
	$K_{dx} = 2,2697$
Régulateur Y	$K_{py} = 6,85890$
	$K_{iy} = 0,1274$
	$K_{dy} = 2,2695$

Tableau III. 2 paramètres du régulateur PID après optimisation

4. Simulation des régulateurs

Nous avons utilisé le model vu lors du chapitre précédent pour réaliser une simulation des régulateurs PIDs. Notre simulation se divise en deux parties :

- Simulation des boucles de régulation simples lié aux variables Z, ϕ , θ , ψ .
- Simulation des doubles boucles de régulation lié aux variable X, Y.

Les simulations sont réaliser sous Matlab-Simulink et sont donné comme suit :

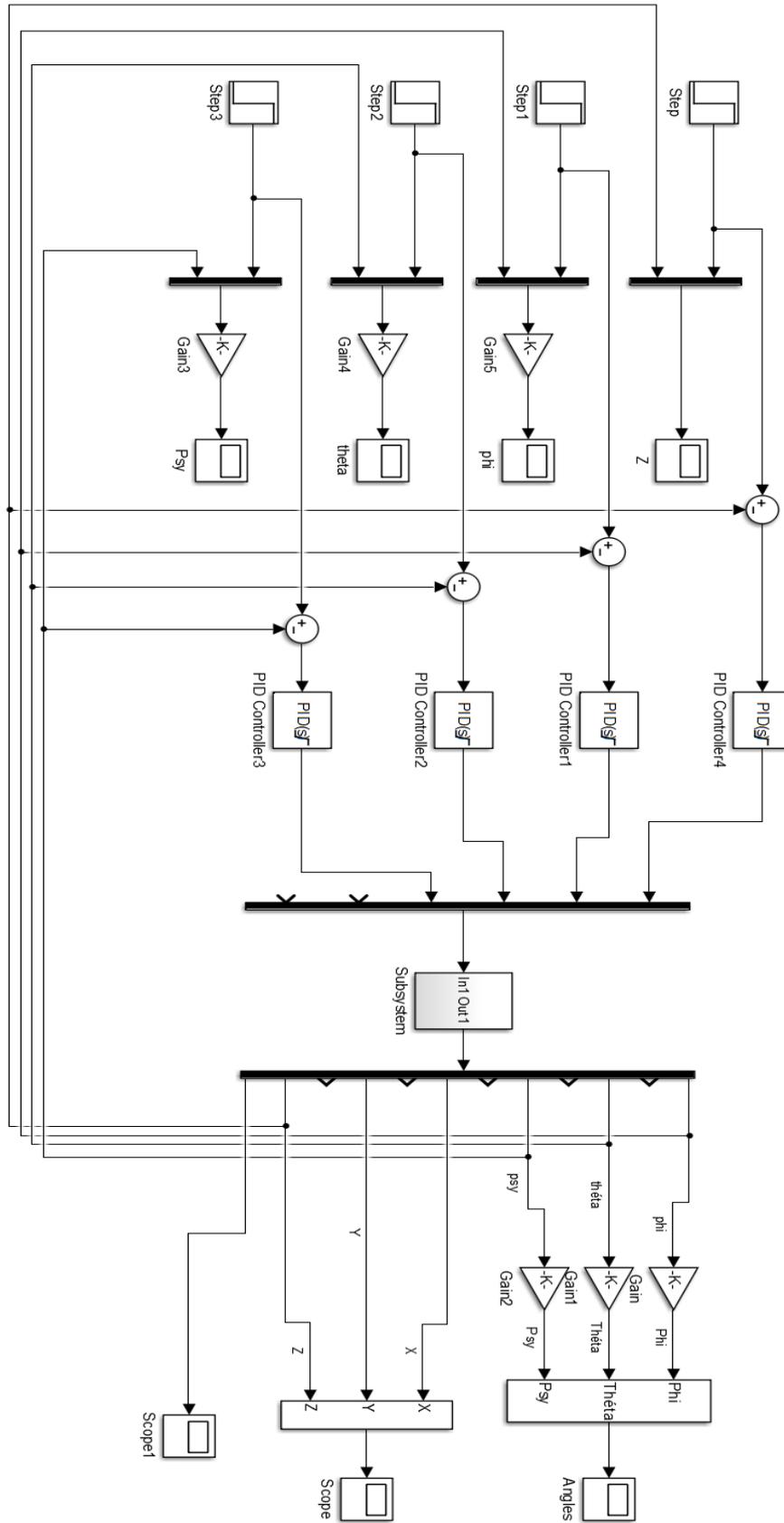


Figure III. 2 simulation Simulink de la régulation PID Z, phi, theta, psy.

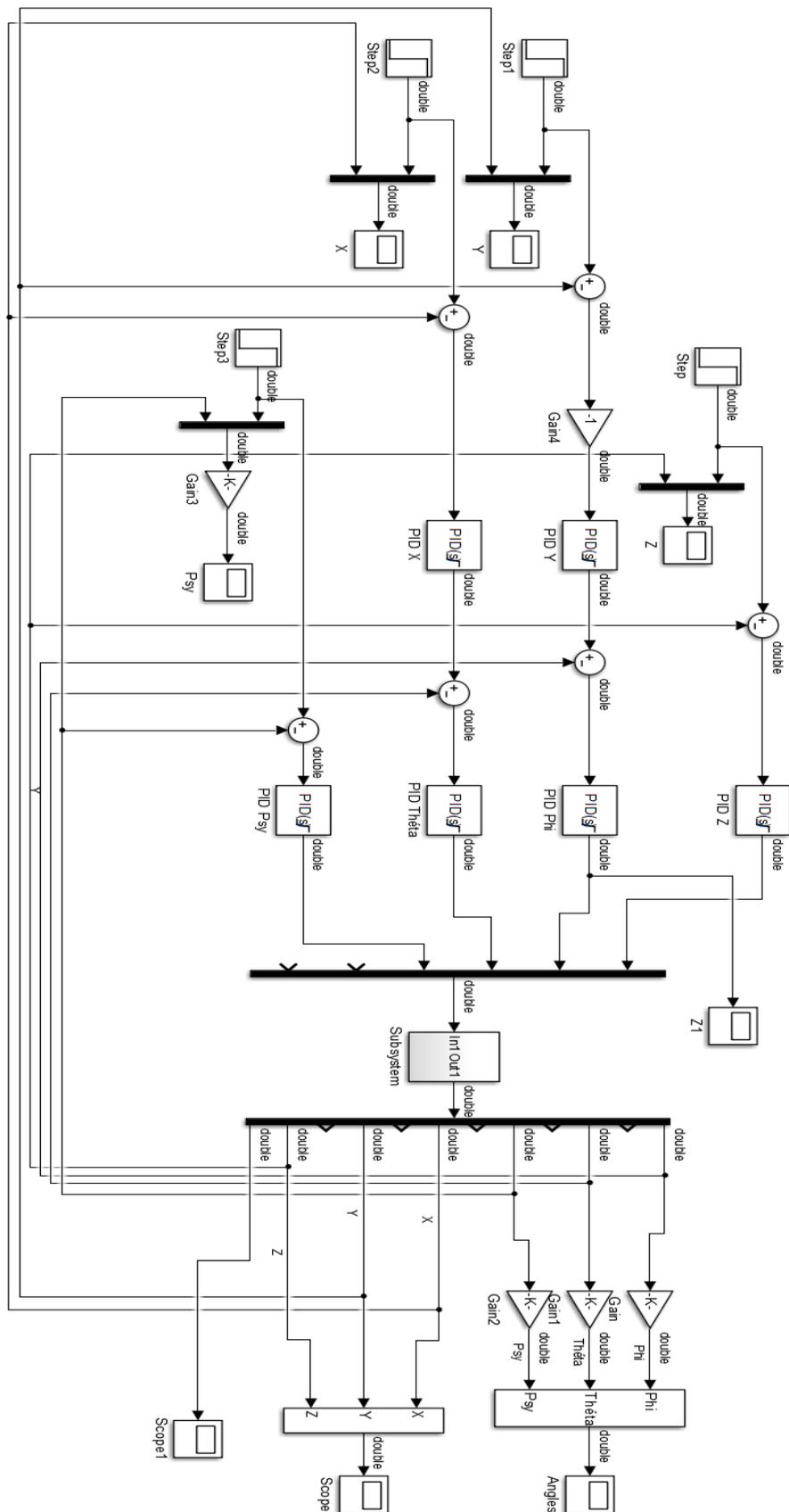


Figure III. 3 simulation sur Simulink de la régulation PID X, Y.

5. Résultats des simulations

Nous allons maintenant voir les résultats obtenus lors des différentes simulations qui consistent en asservissement/poursuite de trajectoire et en régulation/robustesse face à de simples perturbations.

5.1 Régulateur PID non optimisé

5.1.1 . Poursuite d'un échelon

Le premier test auquel nous soumettons notre système c'est un test de poursuite d'échelon Les résultats pour les boucles de régulation sont présenté sur la figure suivante.

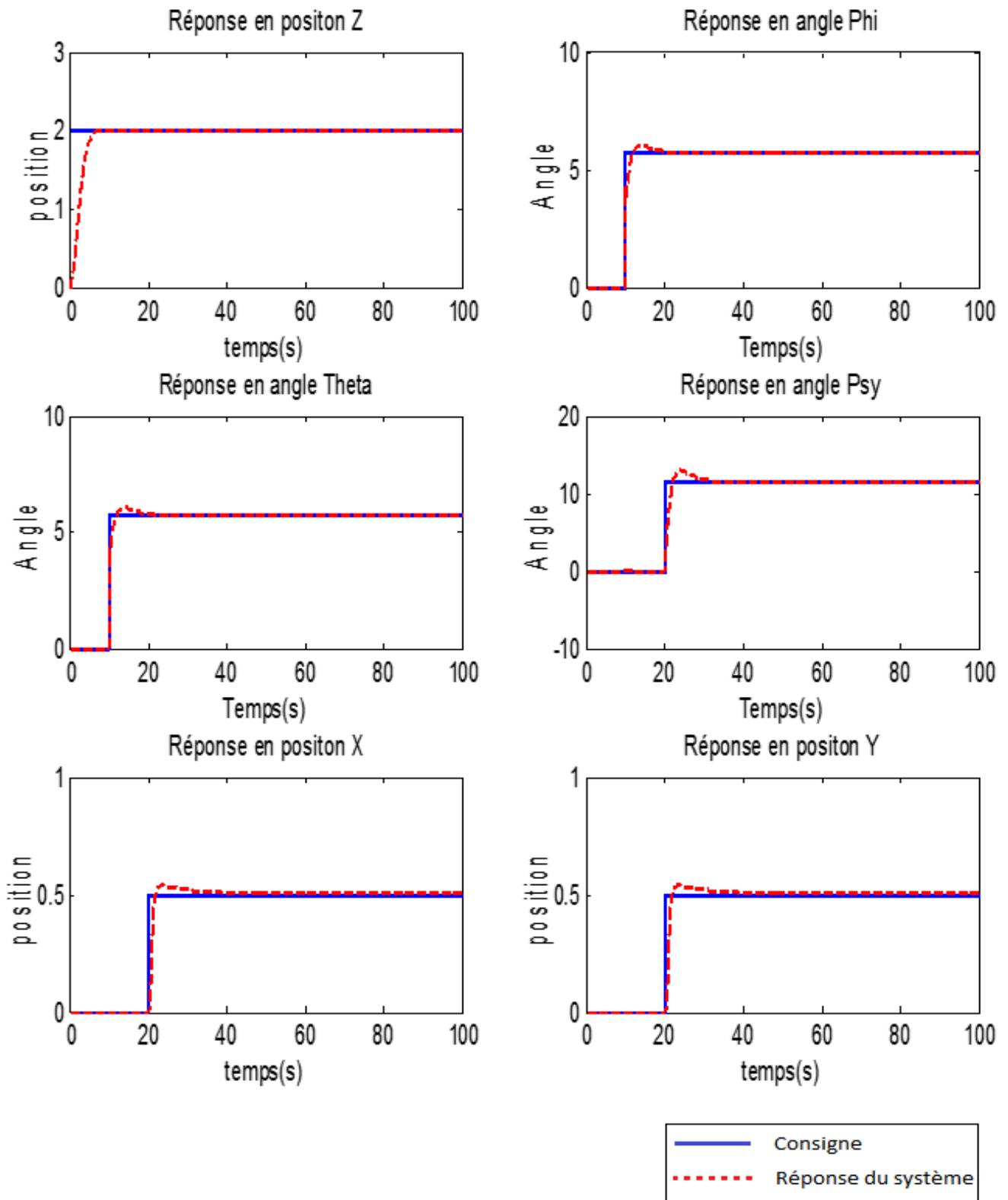


Figure III. 4 : Réponse à un échelon de la régulation PID non optimisée sur toutes les variables

Nous observons une bonne poursuite sur toutes les variables étudiées et bonne vitesse de réponse avec un temps de réponse comme suit :

- Le temps de réponse en Z est de 6.48s.
- Le temps de réponse en Phi est de 5.36s.
- Le temps de réponse en Thêta est de 5.36s.
- Le temps de réponse en psy est de 8.42s.
- Le temps de réponse en X est de 8s
- Le temps de réponse en Y est de 7.93s

Mais nous observons que le système présente des dépassements.

5.1.2 Poursuite de consigne aléatoire

Nous soumettons ensuite le système a une série de consignes aléatoires. Pour réaliser cette simulation, nous avons utilisé un générateur de signal délivrant un signal aléatoire dans un intervalle que l'on spécifie et avec une fréquence réglable. Ainsi nous obtenons les résultats suivant :

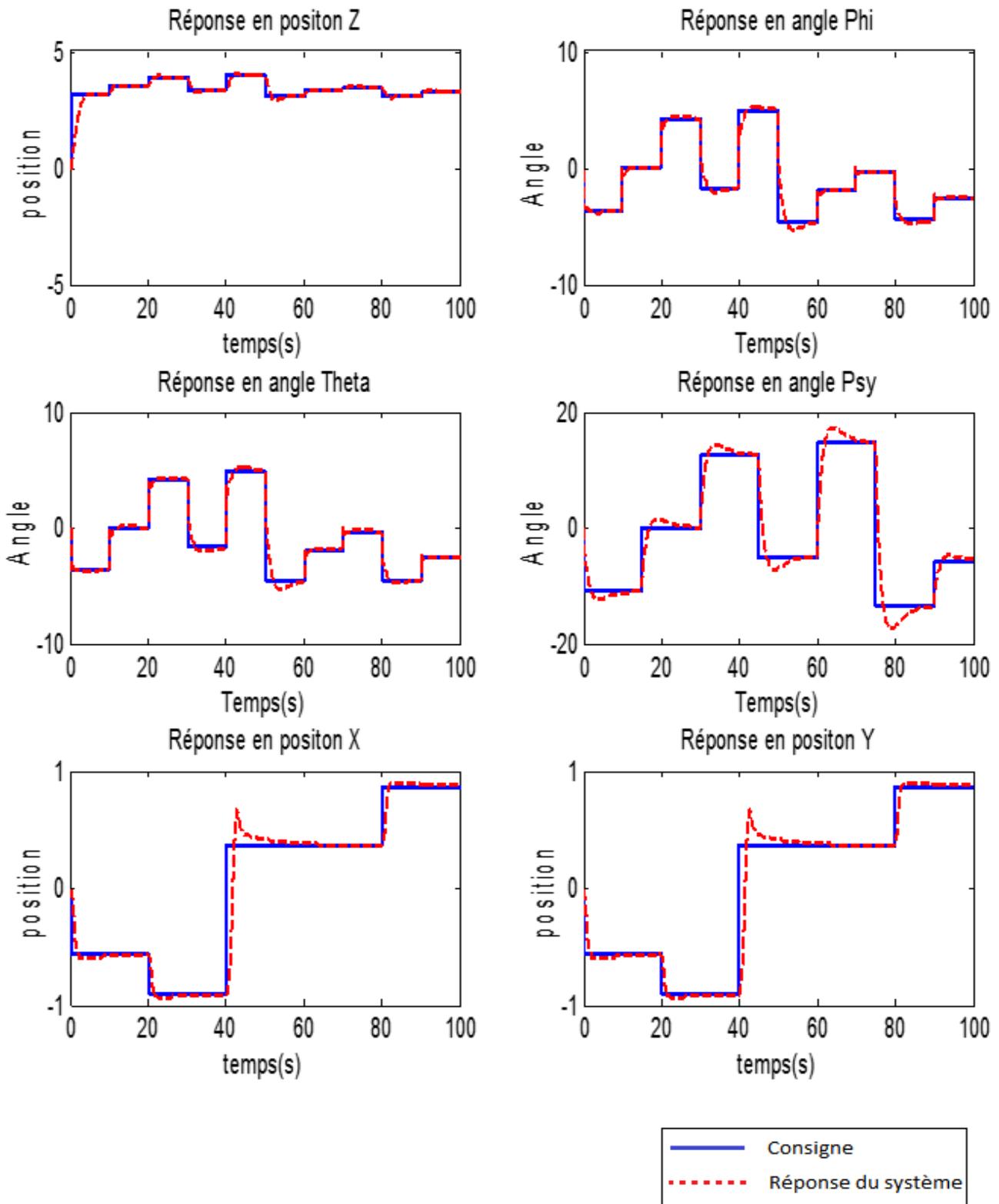


Figure III. 5 : Réponse à des consignes aléatoires de la régulation PID non optimisée sur toutes les variables

Nous observons que le système suit la consigne avec toutefois quelques dépassements, le système a néanmoins une réponse rapide.

5.1.3 Poursuite de consigne sinusoïdale

Dans le prochain test nous mettons à l'épreuve notre système commandé en lui donnant en consigne un signal sinusoïdal. Les résultats sont représentés sur la figure suivante :

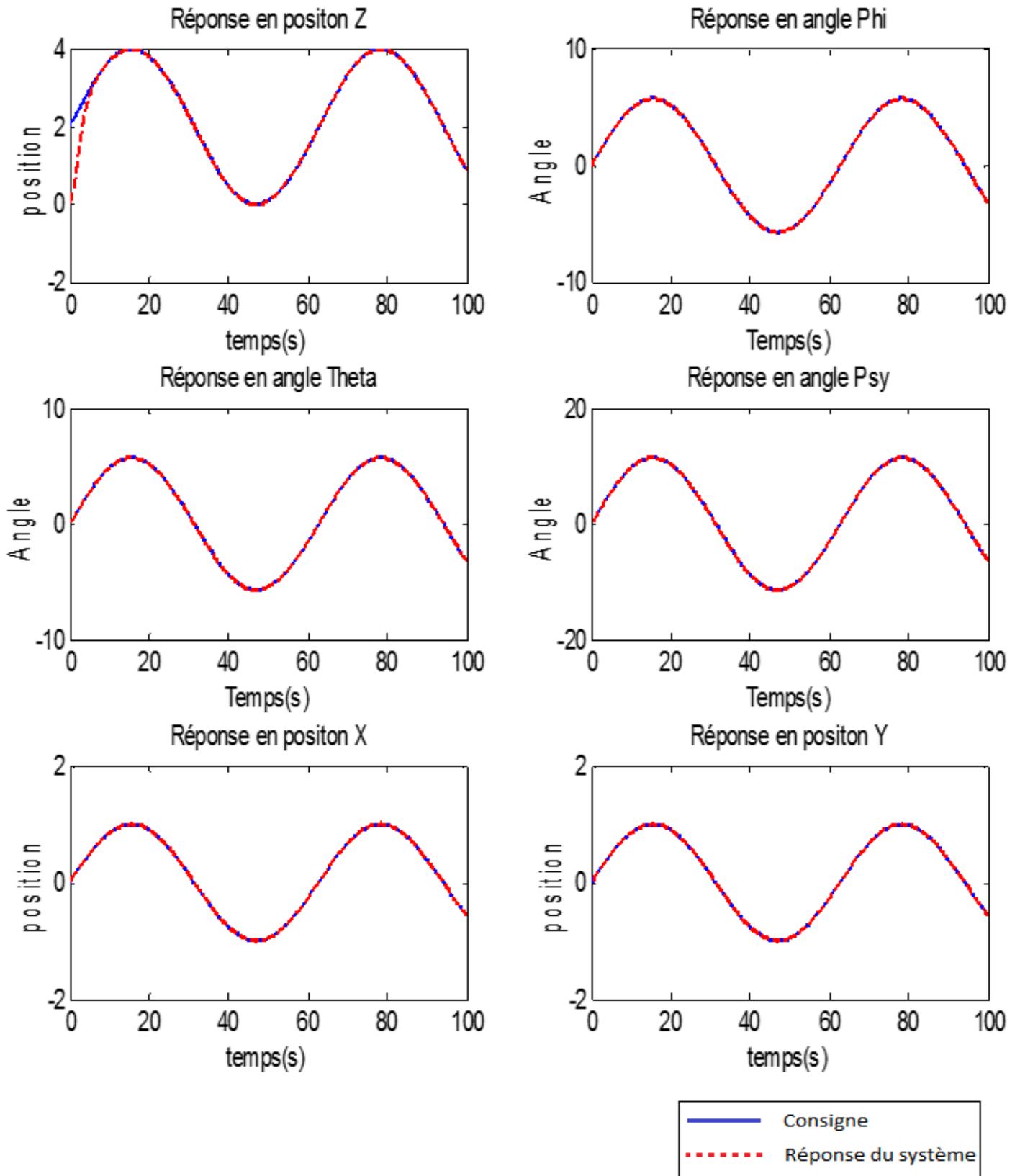


Figure III. 6 : Réponse à une consigne sinusoïdale de la régulation PID non optimisée sur toutes les variables

Le système suit parfaitement la consigne sans aucun déphasage.

5.1.4 Régulation face à des perturbation

Pour réaliser ce dernier test nous allons fixer la consigne a une valeur constante et nous allons injecter des perturbations instantanées et a valeurs aléatoires. Les résultats sont présentés sur la figure suivante :

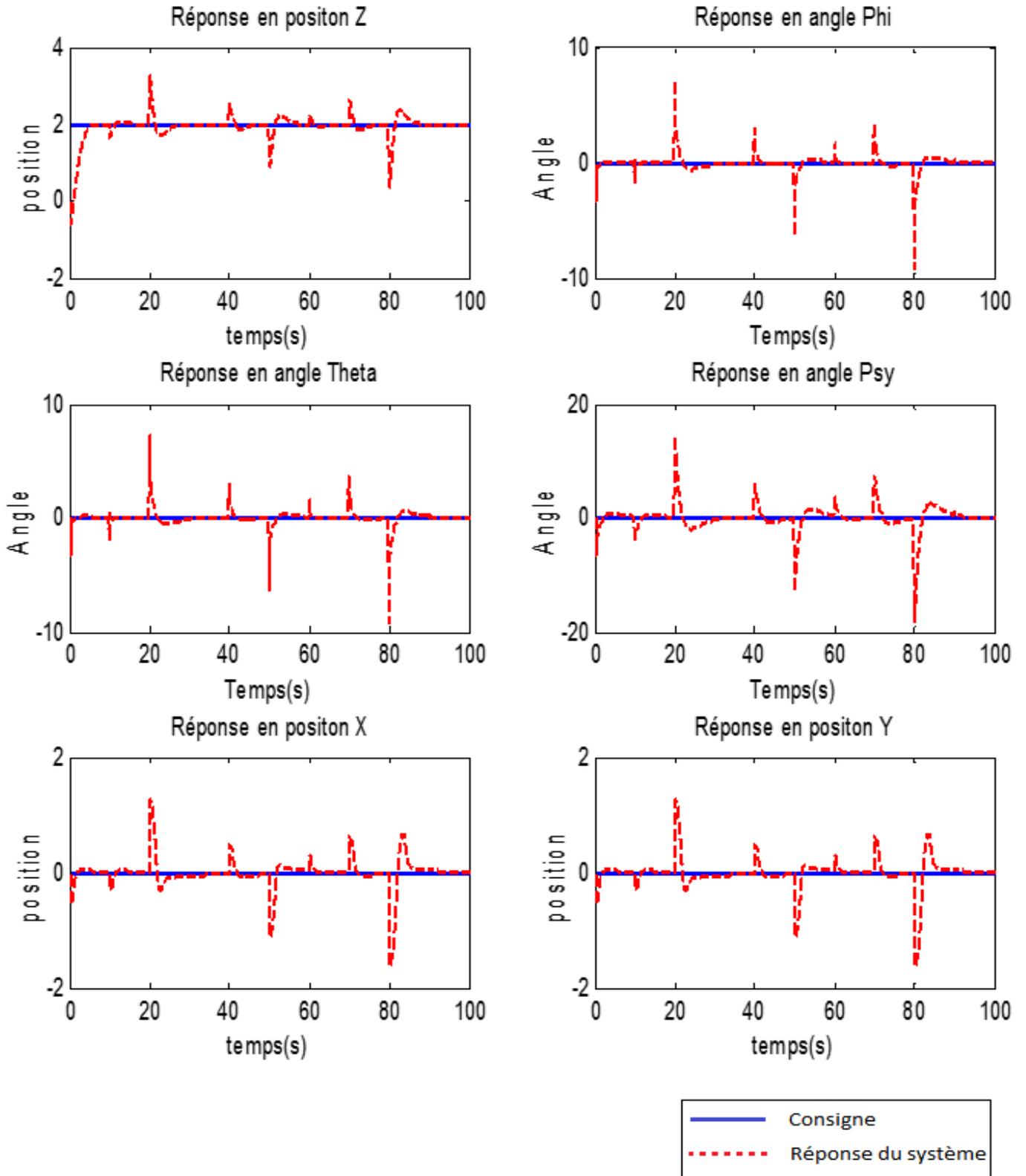


Figure III. 7 : Réponse face à une perturbation de la régulation PID non optimisé sur toutes les variables

Nous remarquons que le système rejette les perturbations avec néanmoins quelques dépassements.

5.2 Régulateur PID optimisé

5.2.1 Poursuite d'un échelon

Le premier test auquel nous soumettons notre système c'est un test de poursuite d'échelon Les résultats pour les boucles de régulation sont présenté sur la figure suivante.

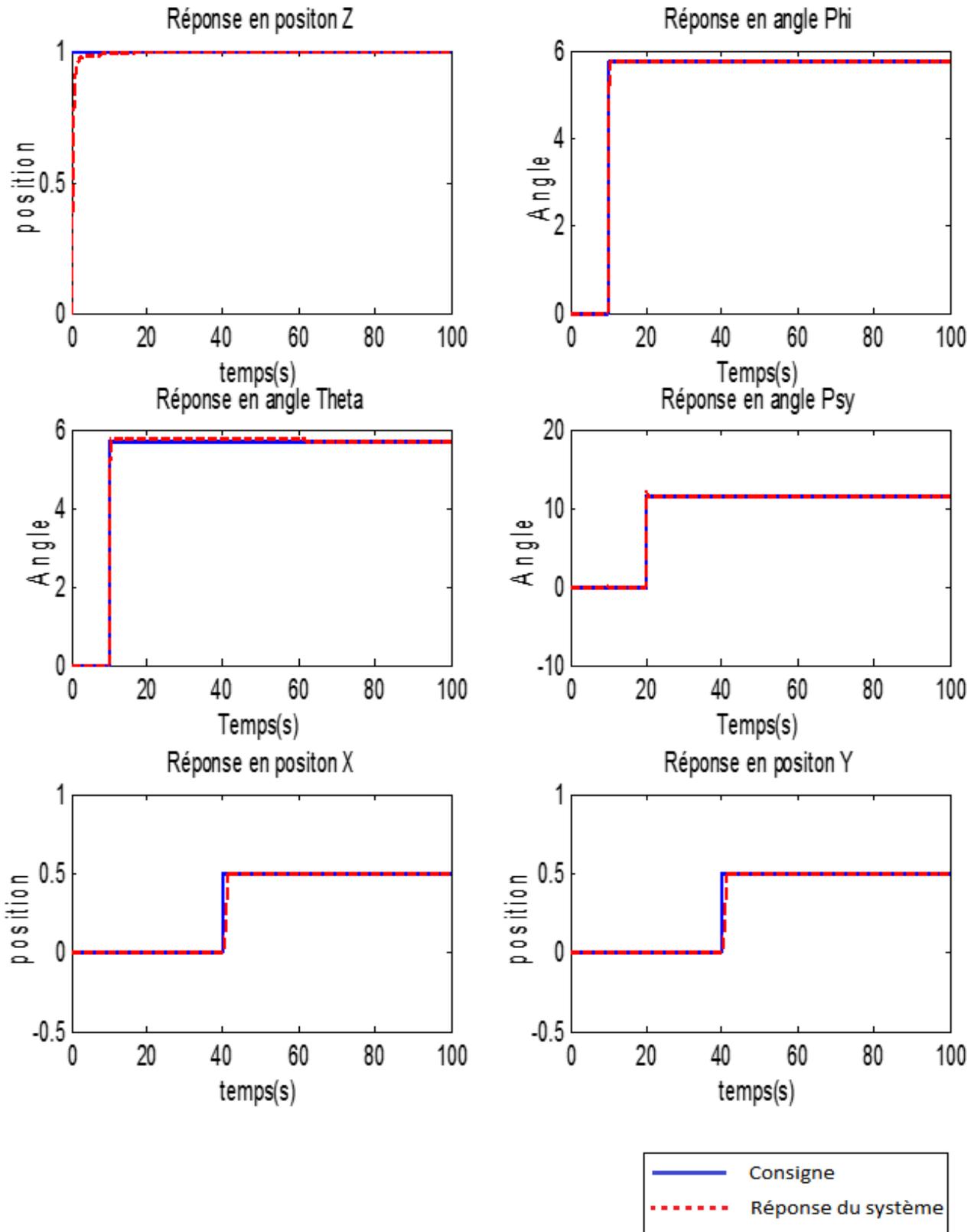


Figure III. 8 : Réponse à un échelon de la régulation PID optimisé sur toutes les variables

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

- Le temps de réponse en Z est de 1.12s.
- Le temps de réponse en Phi est de 0.17s.
- Le temps de réponse en Thêta est de 0.17s.
- Le temps de réponse en psy est de 0.19s.
- Le temps de réponse en X est de 1.55s
- Le temps de réponse en Y est de 1.55s

5.2.2 Poursuite de consigne aléatoire

Nous soumettons ensuite le système a une série de consignes aléatoires. Pour réaliser cette simulation, nous avons utilisé un générateur de signal délivrant un signal aléatoire dans un intervalle que l'on spécifie et avec une fréquence réglable. Ainsi nous obtenons les résultats suivant :

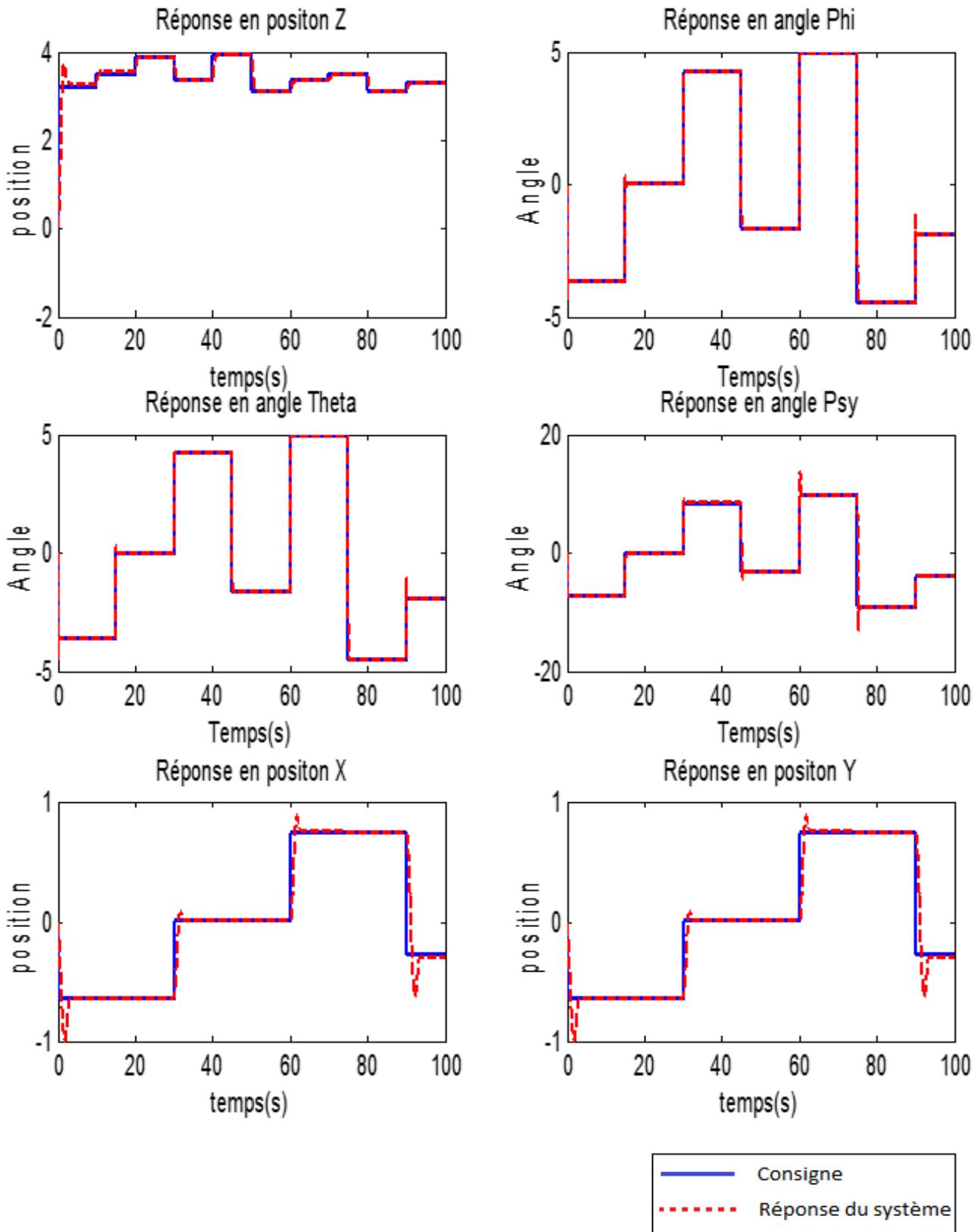


Figure III. 9 : Réponse à des consignes aléatoires de la régulation PID optimisé sur toutes les variables

Nous observons que le système suit parfaitement la consigne avec toutefois quelques dépassements, le système a néanmoins une réponse très rapide.

5.2.3 Poursuite de consigne sinusoïdale

Dans le prochain test nous mettons à l'épreuve notre système commandé en lui donnant en consigne un signal sinusoïdal. Les résultats sont représentés sur la figure suivante :

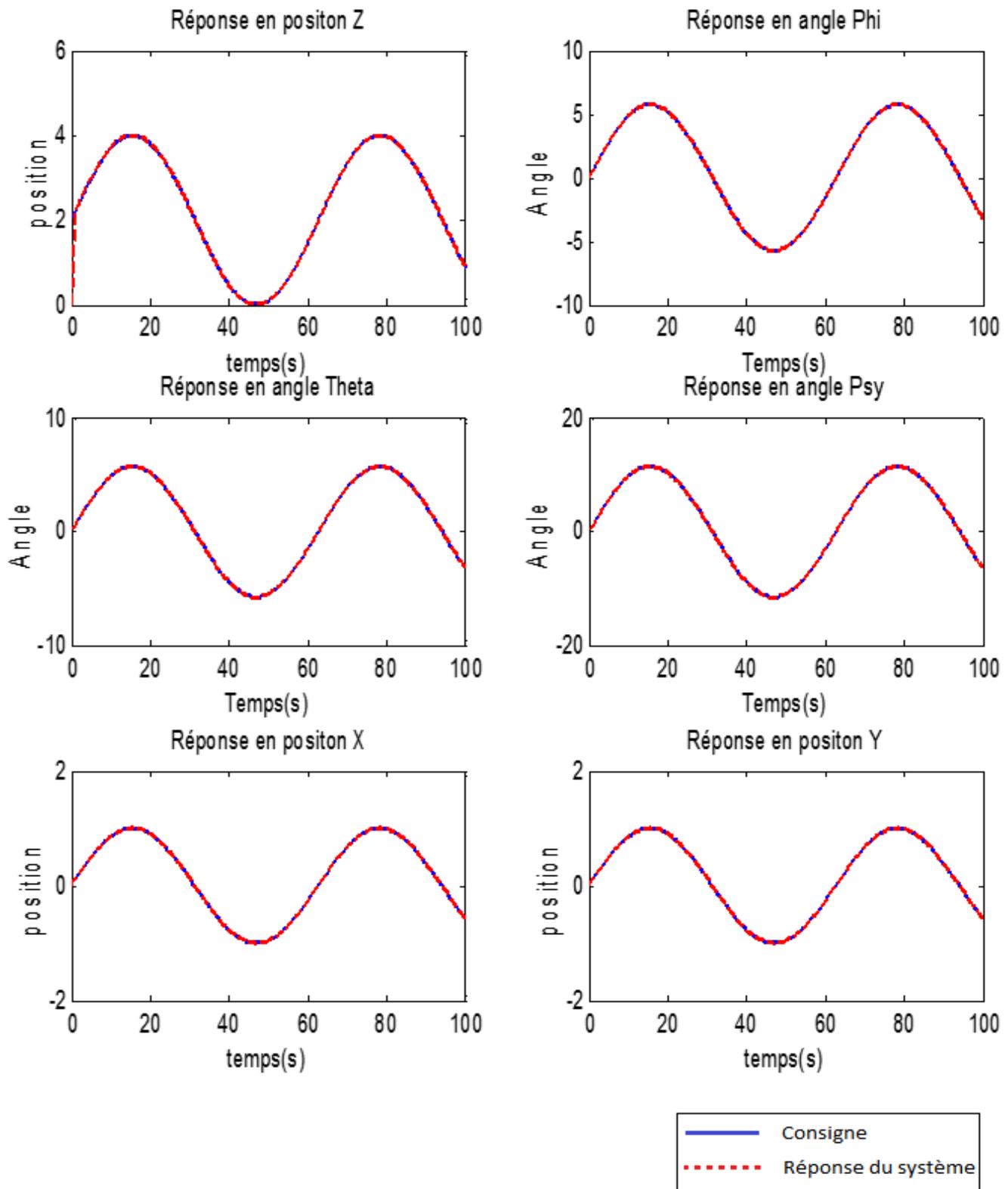


Figure III. 10 : Réponse à une consigne sinusoïdale de la régulation PID optimisée sur toutes les variables

Le système suit parfaitement la consigne sans aucun déphasage.

5.2.4 Régulation face à des perturbation

Pour réaliser ce dernier test nous allons fixer la consigne a une valeur constante et nous allons injecter des perturbations instantanées et a valeurs aléatoires. Les résultats sont présentés sur la figure suivante :

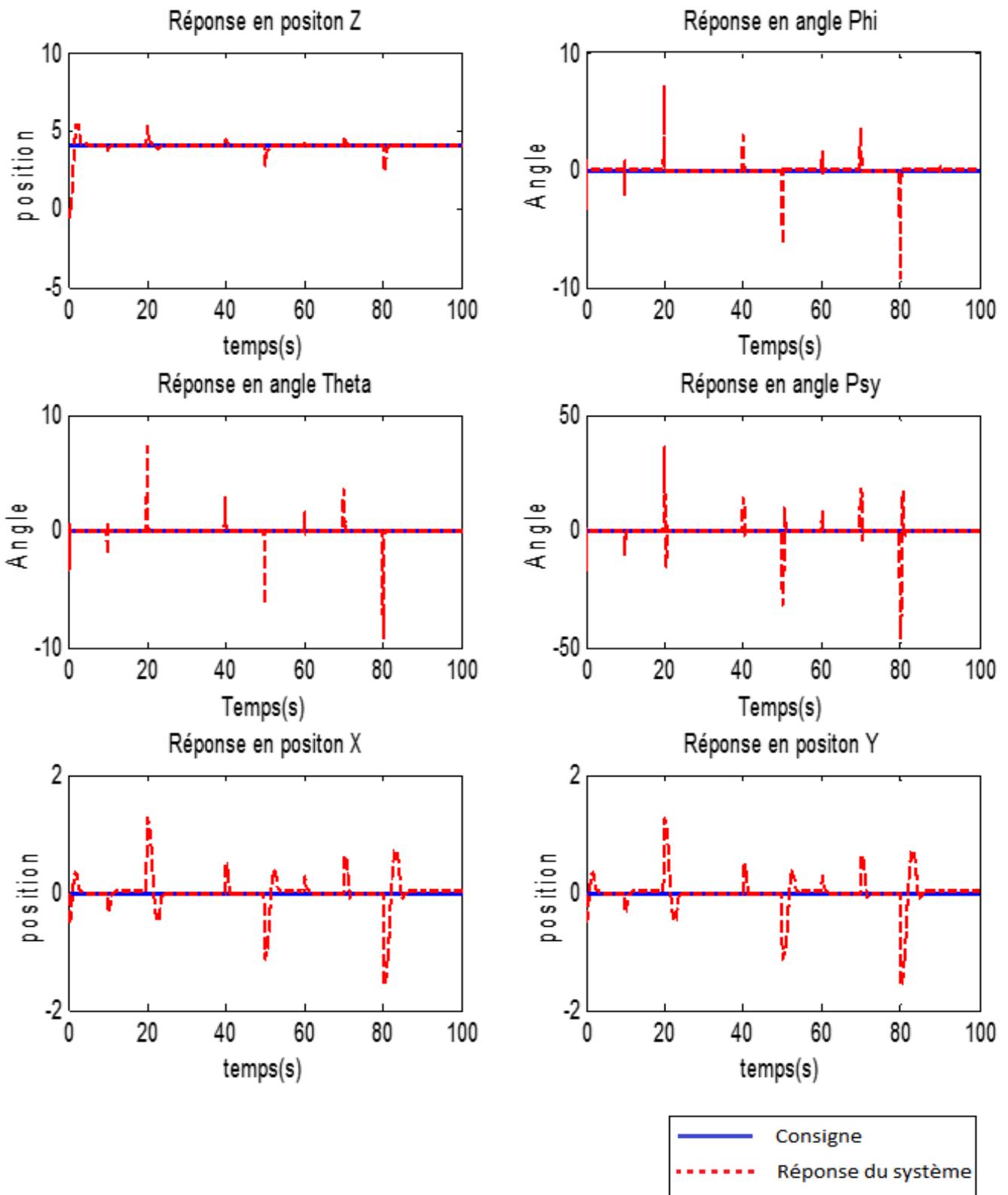


Figure III. 11 : Réponse face à une perturbation de la régulation PID optimisé sur toutes les variables

Nous remarquons que le système rejette toutes les perturbations en un temps minimale mais avec quelques dépassements.

6. Comparaison entre les différentes commandes étudiées

Commande floue.	Commande PID non optimisée.	Commande PID optimisée.
Système lent mais précis et ne présentant pas de dépassement	Système lent et présentant un important dépassement.	Système rapide et précis mais présentant un léger dépassement

Tableau III. 3 : tableau comparatif

7. Conclusion

Dans ce chapitre nous avons brièvement présenté le principe de fonctionnement d'un régulateur PID puis nous avons conçu plusieurs régulateur PID pour la commande de notre système ensuite nous avons procédé à une optimisation quadratique qui consisté à minimiser l'erreur des différentes variables du système. Nous avons ainsi présenté les résultats de simulation obtenus puis nous avons effectué une série de tests grâce à une simulation que nous avons réalisé sur Matlab-Simulink, nous avons ainsi pu observer le comportement du système commandé lors de poursuite de consigne pour l'étude de l'asservissement et de résistances face à des perturbations instantanées pour l'étude de la régulation.

Chapitre IV

Réalisation d'un quadrirotor

1. Introduction

Lors de ce chapitre nous allons voir les éléments indispensables à la réalisation pratique d'un quadrirotor. Nous verrons ainsi la partie électromécanique composé des moteurs, des contrôleurs (ESCs), du châssis, des hélices et des branchements. Ensuite nous verrons la partie acquisition des données et programmation qui consiste en la carte de commande et les différents capteurs.

2. Les composants

2.1 Composants mécaniques

Les composants mécaniques sont les composants qui ne nécessitent pas d'alimentation électrique. Ils sont liés au reste des composant mécaniquement, par entrainement par exemple.

2.1.1 Le châssis

Le choix du châssis (ou sa construction) est l'étape la plus importante dans la réalisation d'un drone, c'est sur ce châssis que seront montés tous les autres composants.

Il existe plusieurs types de châssis pour drone à hélices [9] :

- Le châssis en forme de Y.
- Le châssis en forme de X.
- Le châssis en forme de +.
- Le châssis à six bras.
- Le châssis à huit bras.
- Etc.

Pour notre réalisation nous avons opté pour un châssis de types +, la différence majeure entre ce type et le type X c'est la façon d'actionner les moteurs, en effet pour le châssis de type X la rotation de tangage et de roulis implique toutes les deux d'agir sur les quatre moteurs alors que pour le type + il suffit d'agir sur deux moteurs pour chaque mouvement.

La prochaine étape dans le choix du châssis est de choisir un matériel qui répond aux critères choisis. Il existe une grande quantité de matériaux aux différentes qualités et inconvénients différents [9] :

- Le bois : peu coûteux mais fragile.

- Le plastique : le matériau le plus utilisé pour sa polyvalence qui mêle légèreté solidité et faible coût.
- La fibre de carbone : sans aucun doute le matériau le plus adapté à ce genre de réalisation en raison de sa solidité et sa légèreté, mais c'est aussi le plus coûteux.
- L'aluminium : un matériau léger, solide, facile à travailler et peu onéreux. C'est le matériau que nous avons choisi pour construire notre châssis.
- Il existe aussi différents alliages qu'il est possible d'utiliser.

Nous avons donc réalisé un châssis en forme de + avec de l'aluminium, nous avons aussi mis une plaque centrale pour placer la carte de commande ainsi que les autres composants. La taille de chaque bras est de 20cm et la plaque centrale qui est un carré a une longueur de 15cm.



Figure IV. 1 : châssis en aluminium

2.1.2 Les hélices

Le choix des hélices est très important dans la mesure où il faut des hélices adaptées au type de vol que nous voulons réaliser, de grandes hélices (entre 9 et 11 pouces) assure un vol plus stable et pouvant soulever plus de poids et de petites hélices (entre 5 et 8 pouces) permettent d'avoir un vol « nerveux » plus rapide et nécessite un châssis plus léger [9].

Les matériaux utilisés pour construire des hélices sont nombreux :

- Le plastique : le plus répandu des matériaux de fabrication d'hélices qui permet d'avoir des hélices rigides solide et peu coûteuse.

- Le bois : bien moins répandu les hélices en bois existent et ont des caractéristiques intéressantes en durabilité et en solidité, mais elles sont difficiles à trouver
- La fibre de carbone : le matériau le plus fiable mais aussi le plus couteux.

Nous voulons réaliser un vol stable, nous avons donc opté pour des grandes hélices de 10 pouces en plastique.



Figure IV. 2 : hélices de quadrirotor

2.2 Composants électriques

2.2.1 Les moteurs brushless

Les moteurs brushless sont des moteurs a courant continue qui n'utilisent pas de balais collecteurs car les bobines sont alimentées directement et que les aimants permanent sont placés sur le rotor [9].

Les moteurs brushless sont caractérisés par leurs ratio KV, il représente le nombre de tour par minute que réalise le moteur pour un volt 1V et est proportionnel au voltage d'entrée. Par exemple un moteur de 800KV a une entrée de 12V fera 9600 tours par minutes. Plus le ratio KV d'un moteur est grand plus ce dernier sera rapide mais moins il aura de couple moteur qu'un moteur a ratio KV plus petit.

Le choix du moteur se base donc sur le ratio KV et prend en considération le type de vol et la taille des hélices, pour un vol stable il faut de grandes hélices et des moteurs a fort couple et a petit ratio KV, pour un vol « nerveux » ou rapide il faut de petites hélices et des moteurs a fort ratio KV.

Dans notre cas nous voulons un vol stable nous optons donc pour des moteurs à 900KV.



Figure IV. 3 : moteur brushless

2.2.2 Les variateurs de vitesse électronique (ESC)

Les variateurs de vitesse sont des petites cartes électroniques qui permettent de commander la vitesse des moteurs brushless. Ils acceptent en entrée une tension d'alimentation (de 12V généralement) et une tension de commande envoyée par la carte de commande variant de 0V à 5V, en sortie nous avons deux fils qui fournissent du 5V et du 0V et qui permettent d'alimenter la carte de commande et 3 fils qui permettent d'alimenter et de commander le moteur à la vitesse désirée. Ces variateurs permettent ainsi de commander en vitesse les moteurs brushless grâce à la tension de sortie modulable d'une carte programmable.



Figure IV. 4 : variateur de vitesse

2.2.3 Batterie

Les batteries utilisées pour ce genre de réalisation sont des batteries au lithium polymer (Li-Po) 3S (3 cellules équivalent de 11.1V) qui ont une grande capacité (entre 2000mAh et 3000mAh) ce sont des batteries fiables rechargeable et plus durable que les petites batteries.

2.3 Les capteurs

Nous avons besoin de plusieurs capteurs pour récupérer les données nécessaires au bon fonctionnement de notre drone.

2.3.1 Capteur gyroscopique et accéléromètre

Il représente le capteur le plus important, il permet de récupérer la valeur des angles de rotation autour des axes du quadrirotor à tout moment et ainsi de réaliser une boucle de régulation. Il est accompagné d'un accéléromètre qui permet de récupérer la valeur des accélérations sur les axe de translation. Nous avons opté pour un MPU6050



Figure IV. 5 capteur gyroscopique

2.3.2 Capteurs à ultrason

Ces capteurs permettent de détecter des obstacles dans l'environnement d'évolution de l'appareil. Nous l'utilisons aussi pour récupérer l'altitude de l'appareil lors du décollage. Nous avons opté pour un US-016.

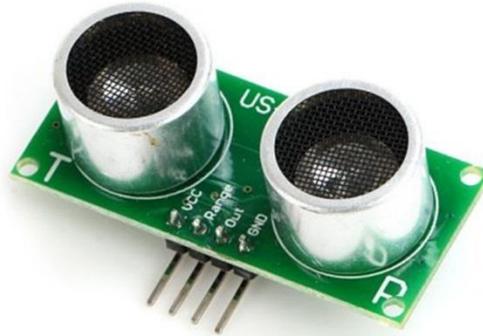


Figure IV. 6 : capteur ultrason

2.3.3 Capteurs GPS

Nous pouvons ajouter à notre quadrirotor un capteur GPS qui permettra de récupérer les coordonnées GPS de l'appareil et ainsi de le suivre à distance à partir d'un ordinateur.

2.3.4 Caméra

Nous pouvons ajouter une caméra à ce système afin de récupérer des données vidéo et ainsi permettre de réaliser plusieurs applications.

2.4 La carte programmable

Pour le choix de la carte de commande plusieurs possibilités s'offrent à nous :

- Cartes Raspberry.
- Cartes STM.
- Cartes Arduino.

Nous avons besoin d'une carte qui dispose de nombreuses entrées digitales et analogiques et de nombreuses sorties digital ainsi que de sortie en PWM (pulse width modulation) et d'une facilité de prise en main. Nous avons choisi la carte « Arduino Mega » car elle répond à toutes les exigences.

La carte Arduino est appréciée pour sa facilité de prise en main, pour sa disponibilité en « open source » (tout le monde peut accéder au plan de fabrication et en fabriquer) et aussi pour sa communauté très active sur le net.

La carte Arduino se programme en langage C et fait appel à de nombreuses bibliothèques incluse dans le logiciel et d'autres qui ont été faites par la communauté pour faciliter la programmation comme

la bibliothèque « eFLL » qui permet d'utiliser des variables et des règles floues à l'intérieur de notre programme.

3. Montage

Dans ce qui suit nous allons voir comment réaliser le montage de notre quadrirotor. Pour commencer nous avons accrocher les moteurs aux extrémités de chaque bras puis nous les avons reliés aux ESCs, nous devons avoir deux moteurs qui tourne dans le sens horaire et deux autre qui tourne dans le sens anti horaire, si après branchement il s'avère qu'un des moteurs tourne dans le mauvais sens il suffit d'inverser deux des trois câbles le reliant à son ESC. La prochaine étape consiste à fixer les ESCs et la carte au châssis puis à fixer le gyroscope bien au centre du châssis pour avoir la plus grande justesse possible. Ensuite il faut placer les hélices, les systèmes diffèrent d'un modèle a un autre mais en générale il suffit de visser. Il faut aussi placer tout autres composants comma la camera ou le GPS pour enfin procéder au branchement comme présenté dans la figure IV.7.

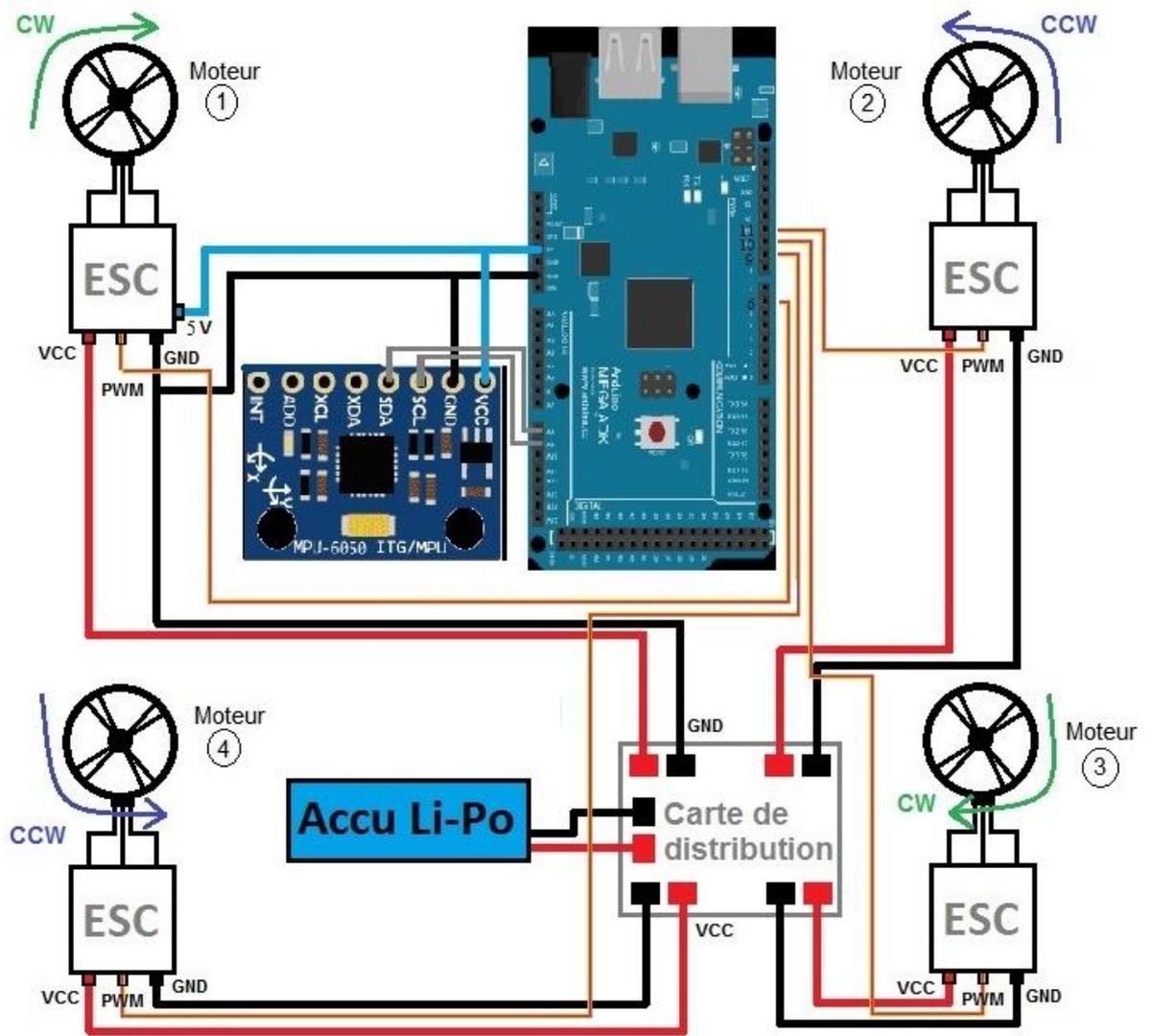


Figure IV. 7 : schéma de branchement

Pour le branchement, nous avons choisi les sorties 11-10-9 et 6 (qui sont des sorties en PWM) pour relier les ESCs à la carte de contrôle et les entrées analogiques A8 et A9 pour le gyroscope. Les ESCs disposent de fils d'alimentation de 5V et de GND qui servent à alimenter la carte programmable. Nous alimentons la carte gyroscope grâce à l'alimentation des ESCs ou de la carte et enfin nous alimentons tous les ESCs grâce à une carte de distribution qui est elle-même alimentée par la batterie Li-Po.

4. Conclusion

A l'issue de ce chapitre nous avons présenté les étapes à suivre pour le choix des composants d'un drone quadrirotor, nous avons présenté les différents composants utilisés lors de notre réalisation puis nous avons expliqué les étapes à suivre lors du montage de l'appareil. Nous n'avons

malheureusement pas réussi à réaliser une maquette fonctionnelle, les causes étant le mauvais choix initial des moteurs et des hélices qui n'était pas compatible en termes de couple mécanique ainsi faute de possibilité de changer le matériel et étant soumis à des contraintes de temps et au vu de la non disponibilité des pièces nécessaire, nous avons été obligés de laisser cette réalisation. Néanmoins nous avons acquis de l'expérience qui nous sera, sans aucun doute, bien utile dans nos projets futurs.

Conclusion générale

Le travail effectué lors de ce mémoire consiste en la modélisation et la commande d'un drone de type quadrirotor dans le but de le stabiliser, de l'asservir et de le réguler. Nous avons aussi tenté de réaliser une maquette physique d'un quadrirotor dans le but de tester les commandes réalisées lors de ce travail. Malheureusement nous avons rencontré de nombreuses difficultés lors du choix du matériel, nous nous sommes retrouvé avec des hélices et des moteurs incompatibles en termes de couples mécanique. La non disponibilité du matériel de rechange nous a conduit à abandonner notre projet de réalisation de maquette, néanmoins nous avons pu, grâce à l'expérience acquise lors de cette tentative, donner quelque conseils relatifs au choix du matériel.

A l'issue du premier chapitre nous avons pu aboutir à un modèle dynamique du système étudié grâce à l'étude des équations agissants sur ce système et en utilisant les équations de Newton-Euler qui décrivent la dynamique de translation et de rotation d'un corps rigide. Nous avons obtenu un modèle d'état non linéaire et nous avons remarqué de fortes interactions entre les états.

Lors du second chapitre nous avons présenté le principe de fonctionnement de la logique floue et la structure du régulateur de Mamdani, que nous avons utilisé pour réaliser une commande de notre système puis nous avons présenté le modèle Simulink qui nous a permis de réaliser plusieurs tests dont les résultats ont été observé et commenté.

Le troisième chapitre a porté sur la réalisation de régulateurs PID pour un quadrirotor et sur une optimisation des paramètres de ces régulateurs. Puis nous avons réalisé une simulation pour pouvoir teste les régulateurs synthétisés, ces tests ont été visualisé et observé. Enfin nous avons comparé les résultats obtenus lors de ce chapitre avec ceux obtenu lors du chapitre précédent, nous avons donc conclu que la régulation par PID optimisé donne de meilleurs résultats en termes de rapidité alors que la régulation floue permet d'éviter les dépassements.

Le quatrième chapitre a été consacré à la réalisation de la maquette et aux différentes étapes à suivre ainsi nous avons décrit les différents types de quadrirotor et le types de matériel nécessaire pour réaliser chaque type. Nous avons présenté tous les composants nécessaires à cette réalisation et nous avons proposé un schéma de branchement.

Références Bibliographiques

- [1] S. Bouabdallah “Design and control of quadrotors with application to autonomous flying”, école Polytechnique Fédérale de Lausanne, 2007.
- [2] L. Derafa, T. Madani, A. Benallegue “Dynamic Modelling and Experimental Identification of Four Rotors Helicopter Parameters”, IEEE 1-4244-0726-5/06, 2006, pp 1834-1839.
- [3] P. Pounds, R. Mahony, P. Hynes, J. Roberts “Design of a four rotor aerial robot”, Australasian conference on robotics and automation, 2002, pp 145-150.
- [4] T. Hamel, R. Mahony, R. Lozano, J. Ostrowski “Dynamic modelling and configuration stabilization for an x4-flyer.”, 15th Triennial World Congress, Barcelona, Spain, 2002.
- [5] H. Khebbache “Tolérance aux défauts via la méthode backstepping des systèmes non linéaires Application : Système UAV de type Quadrirotor“, universite Ferhat Abbas de Setif, 2012.
- [6] H. Bouadi, M. Bouchoucha, M. Tadjine “Modelling and Stabilizing Control Laws Design Based on Sliding Mode for an UAV Type-Quadrotor “, Engineering Letters, 15:2, EL_15_2_24, 2007.
- [7] T. Laamayad “Commande Optimale d’une Machine Asynchrone Apport de La Logique Floue“, université de Batna, 2008.
- [8] B. Toual “Modélisation et Commande Floue Optimisée d’une Génératrice à Double Alimentation, Application à un Système Eolien à Vitesse Variable”, Université de Batna, 2010.
- [9] Jean-Louis Naudin “Construire un Quadricoptère“, Vers 1.2, 2011.