République Algérienne Démocratique et Populaire

Ministère de l'Enseignement Supérieur et de la Recherche Scientifique

Université Abderrahmane Mira de Béjaïa

Faculté des Sciences Exactes

Département d'Informatique

Mémoire de fin de cycle

En vue de l'obtention du diplôme master professionnel informatique

Option : Génie logiciel

Thème

Simulation d'un drone sous MATLAB

Cas d'étude : Quad-copter

Présenté par :

MOHAMEDI Fatima & SACI Nassim

Soutenu devant le jury composé de :

Rapporteur	M_{me} GHIDOUCHE Kahina	M.A.A	Université A.Mira Béjaia
Président	M_r KHENOUS Lachemi	M.A.A	Université A.Mira Béjaia
Examinateur	M_r AMROUN Kamel	M.C.B	Université A.Mira Béjaia

Promotion 2015 – 2016

Remerciements

Nous tenons à remercier notre encadreur M_{me} GHIDOUCHE Kahina, qui nous a aidé à progresser dans notre réflexion grâce à ses conseils, son esprit critique et son soutien tout au long de la réalisation de ce mémoire.

Nous remercions également les membres de jury qui ont accepté d'examiner notre modeste travail.

Sans oublier de fortement remercier M_r CHAIBI Gaya, BENYAHIA Amar des étudiant en master 2 génie mécanique qui nous ont aidé à comprendre les différentes notions de la mécanique, spécialement leur encadreur BELAMRI Abdelatif.

Enfin, nous remercions tous ceux qui ont contribué de près ou de loin à la réalisation de ce travail.

Dédicace

C'est avec profonde gratitude et sincéres mots, que nous dédions ce modeste travail de fin d'étude

à nos trés chers parants ; qui ont sacrifié leur vie pour notre réusite et

nous ont éclairé le chemin par leurs conseils judicieux.

Nous dédions aussi ce travail à nos fréres, soeurs et leurs époux.

Nos grands parents.

Nos tantes, nos oncles et leurs familles .

Nos cousins et cousines.

Tous nos adorables amis.

Ainsi qu'à toute personne qui nous ont soutenue.

Liste des Abréviations

AC Alternating Current

DC Direct Current

ESC Electronic Speed Controller

GPS Global Position System

HMI Human Machine Interface

IMU Inertial Measurement Unit

IR InfraRed

IRLED IR Light Emitting Diode

LED Light Emitting Diode

P Proportionnel

PI Proportionnel Intégrale

PID Proportionnel Intégrale Dérivée

PSD Position-Sensitive Device

RPAS Remotely Piloted Aircraft Systems

SONAR SOund Navigation And Ranging

UAV Unmanned Aerial Vehicle

Table des matières

C Annexe

Table des figures

Liste des tableaux

Introduction générale

Durant les dix dernières années, un intérêt croissant est porté aux engins volants sans pilote humain à bord que l'on appelle drone. Un drone est un véhicule aérien robotique capable de mener une mission de façon autonome.

Au début, les applications du drone ont été orientées uniquement vers le domaine militaire. Aujourd'hui, ils sont aussi utilisés dans divers domaines tel que la recherche, sauvetage, cartographie, etc...

Dans le domaine civil, l'utilisation des drones est envisageable pour un grand nombre de missions délicates ou coûteuses telles que l'exploration d'un environnement inconnu, la surveillance ou l'intervention sur des zones potentiellement dangereuses, l'évaluation des dommages, la surveillance des feux de forêt, des lignes électriques à haute tension, du trafic routier, le survol des régions montagneuses et peu accessibles.

Les fortes évolutions technologiques ont permis le développement des drones de plus en plus miniaturisés, du coût modéré et facile à utiliser. Un intérêt particulier est destiné aux quad-copters de petite taille pour leurs forte demande d'utilisation dans plusieurs domaines. Les quad-copters sont actionnés par des moteurs électriques qui peuvent être sans balais (brushless) alimentés par des courants continus.

La conception d'un robot aérien tel qu'un quad-copter est une tâche complexe, nécessitant des coûts très élevées et des connaissances dans divers domaines, impliquant ainsi l'utilisation d'une plate-forme virtuelle de simulation.

Dans cette optique nous allons traité la problématique suivante :
Comment réaliser une plate-forme virtuelle de simulation afin d'étudier les comportements et la stabilité d'un quad-copter ?

L'objectif principal de ce travail porte sur :

- L'exploitation d'un modèle mathématique d'une complexité minimale qui définit la dynamique du quad-copter.
- Transcrire ce modèle sous forme d'algorithmes qui seront implémentés sous Simulink.
- Représentation des régulateurs PID sous forme de blocs Simulink, afin de varier et simuler avec différentes valeurs des paramètres du régulateur.
- Réalisation d'une plate-forme virtuelle qui permet d'exécuter les algorithmes de contrôle et visualiser les résultats de simulation.

En effet le véhicule aérien sans pilote connaît un essor croissant ces dernières années grâce à leurs différents domaines d'application, ces appareils ont notamment l'avantage de moins exposer le personnel au dangers tout en couvrant une large gamme de missions. La réalisation de tel véhicule apparaît très coûteuse, pour éviter cela, un logiciel de simulation est indispensable afin d'avoir une vision sur les différents comportements du quad-copter dans le monde réel.

Afin d'atteindre l'objectif cité ci-dessus nous avons opter pour le plan de travail qui s'articule autour de quatre chapitres :

Chapitre I : Généralités sur les quad-copters

Dans ce chapitre, on présente les drones en général et les quad-copters en particulier. Voir leur différents domaines d'application et leur fonctionnement. Ensuite, une étude approfondie et détaillée sur les différentes parties mécanique, électronique et informatique d'un quad-copter.

Chapitre II : Modélisation mathématique d'un quad-copter

Ce chapitre porte sur la modélisation mathématique de la dynamique du quad-copter en se basant sur les équations de mouvement de Newton-Euler. Dans un premier temps, nous définissons les repères de référence nécessaires pour exprimer l'orientation quad-copter. Dans un deuxième temps nous exposant le modèle mathématique complet sous forme d'équations. Pour finir, nous présenterons les moteurs à courant continu (Brushless).

Chapitre III : Algorithme de contrôle

Ce chapitre s'articule sur trois sections, la premier section consiste a présenter les différents régulateurs existants afin de choisir un régulateur convenable à ce système. Dans la deuxième section nous allons voir comment contrôler les différents mouvements en utilisant la technique

PID tout en testant le modèle de simulation réalisé sous MATLAB, où il est facile d'évaluer les performances du système avec une approche mathématique. La dernière section consiste à réaliser une plate-forme du quad-copter pour évaluer le comportement du système réel.
Chapitre IV : Simulation d'un modèle

Dans ce chapitre, nous allons simuler le modèle expliqué dans le chapitre III, et discuter des différents résultats qu'on a obtenu avec différentes valeurs.

Ce chapitre a pour but d'illustrer la déstabilisation et la stabilisation du quad-copter selon les valeurs du régulateur PID. Pour le contrôle du mouvement d'altitude, roulis, lacet ou tangage. Les résultats sont présentés à l'aide de l'outil Scope ou la commande plot sous simulink.

I

Généralités sur les Quad-copters

Introduction

Un drone est un engin volant sans pilote à bord, commandé à distance et réutilisable. Il peut se définir comme un système mécanique, électronique et informatique contrôlé ou programmé pour effectuer des tâches trop répétitives, dangereuses ou difficiles pour être faites directement par des êtres humains. Aujourd'hui les drones jouent un rôle très important dans divers domaines et qui accomplies plusieurs missions. ce qui a mener pas mal de pays à les développer.

Dans ce chapitre nous allons définir les drones en général et les quad-copters en particulier. Voir leur différents domaine d'applications et fonctionnalités. Ensuite, Une étude approfondie et détaillée sur les différentes parties d'un quad-copter.

I.1 Définition d'un drone

Un drone de l'anglais faux-bourdon est un appareil volant sans pilote, semi autonome ou complètement autonome qui peut emporter une charge utile est destinée à des missions spécifiques [1]. En France, le terme drone désigne aussi un aéronef sans pilote destiné à un usage public et on emploie drone militaire ou drone de combat pour désigner un matériel équipé d'armements. Les Anglo-Saxons nomment ainsi un appareil non militaire sans personne à bord, télécommandé ou autonome, par *UAV*, ou encore *RPAS* [2].

Le véhicule aérien sans pilote à long terme décrit les drones volants et les hélicoptères. S'il fonctionne dans l'air, nous l'appellerons un UAV. Le drone amateur le plus populaire est le quad-rotor ou quad-copter [3].

Enfin, le caractéristique essentielle des drones est qu'ils sont récupérables, ce qui permet de les réutiliser. Cela les différencie des missiles, auxquels on aurait pu être tenté de les assimiler [4].

I.2 Définition d'un quad-copter

Un quad-copter est un aéronef à voilure tournante comportant quatre rotors pour sa sustentation. Les rotors sont généralement placés aux extrémités d'une croix. Afin d'éviter à l'appareil de tourner sur lui-même sur son axe de lacet, il est nécessaire que deux hélices

tournent dans un sens et les deux autres dans l'autre sens. Pour pouvoir diriger l'appareil, il est nécessaire que chaque couple d'hélices tournant dans le même sens soit placé aux extrémités opposées d'une branche de la croix.

I.3 Domaines d'applications

Nous évoquons ici quelques missions des drones :

1) **Recherche et sauvetage :** Aider l'équipe de sauvetage à localiser les victimes d'une manière rapide en cas d'un accident ou une catastrophe naturelle.

2) **Cinématographie :** Les drones peuvent être la caméra ou l'œil dans le ciel d'un directeur artistique.

3) **Inspection :** Les drones peuvent inspecter les grandes structures, bâtiments, routes et les lignes électriques à haute tension dans des sites lointains ou difficiles d'accès, afin de localiser les matériaux dangereux par le transfert d'images aériennes ou par l'identification automatique de matériaux.

4) **Cartographie :** Les drones peuvent construire des cartes topologiques plus précises que l'avion conventionnel. ils peuvent voler près de la terre tout en portant des appareils photographiques ou des sondes pour construire des cartes en *3D* de haute résolution [5].

On peut utiliser les drones dans plusieurs autres domaines, comme la télécommunication, la livraison, la publicité, la médecine, etc...

I.4 Fonctionnement

Le fonctionnement d'un quad-copter est assez particulier. On distingue quatre mouvements possibles : vertical, le lacet, le roulis et le tangage. Pour garder le contrôle du lacet, cela implique que deux hélices tournent dans le sens horaire (hélice à pas normal) et les deux autres dans le sens antihoraire (hélices à pas inversé).

Le vertical

Le mouvement de vertical correspond tout simplement à la montée/descente du quad-copter. La montée est obtenue en augmentant la vitesse des quatre moteurs. La descente, qui elle est plus difficile à doser, s'obtient par la réduction de la vitesse des moteurs.

Le lacet

Le mouvement de lacet sert à faire tourner le quad-copter sur lui-même. Il est obtenu en augmentant la vitesse des hélices à pas normal et en diminuant proportionnellement la vitesse des hélices à pas inversé.

Le roulis et le tangage

Le roulis et le tangage sont des mouvements assez similaires visant à pencher le quad-copter sur un axe ou sur un autre. Ce mouvement est obtenu en augmentant la vitesse d'une hélice et en abaissant proportionnellement la vitesse de l'hélice opposée (hélice du même couple).

I.5 Partie Mécanique

Dans la première partie nous évoquons les composants essentiels d'un quad-copter. La deuxième partie consiste à présenté les capteurs qui servent à mesuré la hauteur et les angles de rotation.

I.5.1 Composants

La figure suivante montre les composants essentiels d'un quad-copter :

FIGURE I.1 – Les composants d'un quad-copter [3]

Hélices

Un quad-copter est constitué de deux hélices standard (sens horaire) et deux hélices pusher (sens anti horaire) rotation contraire [3].

Moteur électrique

Quatre moteurs sans balai (brushless) sont nécessaires pour le quad-copter. Il en existe de différentes tailles et puissances. Ils sont caractérisés par le diamètre de leur cage tournante et par le nombre de $tours/volt$ ou KV. Un moteur ayant un KV de 1000 $tours/V$ fonctionnera à 12000 $tours/min$ s'il est alimenté en 12 V. Sur les moteurs sans balai utilisés en modélisme, les bobinages en cuivre sont montés sur le stator et les aimants sur le rotor, à l'inverse des moteurs électriques conventionnels [6].

Contrôleur de vitesse

Les *ESCs* convertissent les *DC* en *AC* pour les sans balai (brushless) moteur, et ils permettent aussi d'alimenter les moteurs. Chaque moteur a besoin d'un *ESC*. On peut modifier le firmware d'un ESC pour créer un comportement diffèrent d' un moteur. Par exemple, *ESC* est souvent configuré pour diminuer la vitesse d'un moteur plutôt de l'arrêter brusquement [3].

Contrôleur de vol

C'est une carte électronique, équipée de capteurs très précis, qui va traiter les consignes du pilote envoyées à l'émetteur ainsi que les informations envoyées par ses capteurs et va transmettre des impulsions électriques aux contrôleurs des moteurs pour faire varier leur vitesse. Ces cartes sont équipées de gyroscopes et d'accéléromètres pour mesurer et compenser les déplacements. Certaines cartes évoluées sont équipées de *GPS* et d'altimètre afin de maintenir un point fixe ou une altitude ou même de retourner au point de départ du drone en cas de perte de signal radio [6].

Cadre

Sur les quad-copters, le cadre comporte quatre bras en forment une croix. Suivant les règles de la construction mécanique. il existe deux configurations possibles en X et en H [3].

Batteries

Les batteries utilisées sur un quad-copter sont essentiellement des « Lithium Polymère ». Elles sont issues d'une technologie qui permet d'avoir un très bon rapport *poids/puissance*. Un élément Li-Po ($1S$) fournit une tension de $3,7v$. Sur un drone, on utilise en général des batteries à 3 ou 4 éléments ($3Sou4S$). L'intensité est aussi un critère de choix. Une batterie de 3000 mAH aura une meilleure autonomie qu'une batterie de 2200 mAH.

Un chargeur spécifique est à prévoir ainsi que des règles de sécurité très strictes car ces batteries peuvent exploser en cas de mauvaise manipulation [6].

Camera

C'est des cameras à faible résolution qui envoie des images à des station terre via des ondes radio.

Radio-commande

Pour piloter le quad-copter, il faut un émetteur radio pour le pilote et un récepteur dans le drone. Il existe plusieurs technologies pour les radio-commandes, les radios FM en $41MHz$ de moins en moins utilisées et les radios en $2,4\ GHz$, souvent programmables pour s'adapter à chaque appareil radio-commandé. Une radio-commande doit idéalement comporter 6 voies pour piloter un quad-copter, 4 voies sur les manettes et 2 voies sur des interrupteurs pour actionner certaines fonctions. Il existe deux modes de configuration des manettes, le premier mode dans lequel les gaz sont à droite et le deuxième mode où les gaz sont à gauche. Certaines radios fonctionnent dans les deux sens, c'est-à-dire qu'elles peuvent envoyer des ordres à l'émetteur mais aussi recevoir des informations de celui-ci...) [6].

I.5.2 Capteurs

Les capteurs sont fondamentaux pour identifier et calculer l'altitude et la hauteur d'un corps. Grâce à ces capteurs il est possible d'accomplir l'évitement des obstacles et la planification de la trajectoire lors d'un contrôle du haut niveau [7].

SONAR

SONAR est un capteur capable de mesurer la distance en utilisant des ondes ultra-son. Dans un quad-copter on le combine avec le module *IR* pour estimer la hauteur du quad-copter par rapport au sol. Il est monté sur la face de dessous et il pointe vers le bas. Pour calculer la distance (Z), des données mesurées par ce capteur $SONAR$ doivent être multipliées par une consigne de l'angle de roulis (ϕ) et l'angle de lacet (θ) mesuré par l'IMU présenté ci-après [7].

IMU

IMU est un ensemble de capteurs capables de mesurer l'altitude, l'accélération et l'orientation par rapport un point fixe d'un corps. Il est composé de plusieurs capteurs Accéléromètre,

Baromètre, gyroscope [8].

Module IR

IR sert à identifier l'altitude grâce à des ondes lumineuses Avec une certaine plage de fréquence. Ces ondes ne sont pas visibles par l'être humain. Le module ou le système *IR* est composé d'un *IRLED* chargé de la conversion d'un signal électrique en onde *IR*, et un *PSD* qui est chargé de la conversion inverse [7].

I.6 Partie électronique

La partie électronique d'un quad-copter est constituée d'une carte Arduino. Cette carte est programmable à partir d'un ordinateur via l'outil Arduino IDE.

I.6.1 Arduino

Comment faire des montages électroniques simplement en utilisant un langage de programmation ? La réponse, c'est le projet Arduino qui l'apporte. Celui-ci a été conçu pour être accessible à tous par sa simplicité, mais il peut également être d'usage professionnel. Ces cartes polyvalentes sont donc parfaites pour notre projet [9].

I.6.2 Qu'est ce que c'est ?

Arduino est un projet créé par une équipe de développeurs, composée de six individus : Massimo Banzi, David Cuartielles, Tom Igoe, Gianluca Martino, David Mellis et Nicholas Zambetti. C'est un outil qui va permettre aux débutants, amateurs ou professionnels de créer des systèmes électroniques plus ou moins complexes. Le système Arduino, nous donne la possibilité d'allier les performances de la programmation à celles de l'électronique [9].

I.6.3 Les outils Arduino

Il est composé de deux principaux outils, qui sont : le matériel et le logiciel.

Le matériel Il s'agit d'une carte électronique basée autour d'un microcontrôleur Atmega du fabricant Atmel, dont le prix est relativement bas pour l'étendue possible des applications. Il y a trois types de carte : officielles, compatibles, Seeduino. la figure (I.2) montre à quoi ressemble une carte Arduino [9] :

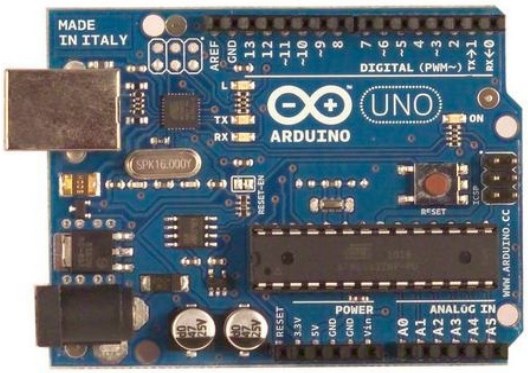


FIGURE I.2 – Carte Arduino Uno

le logiciel Le logiciel va nous permettre de programmer la carte Arduino. Il nous offre une multitude de fonctionnalités. Il est gratuit et open source, développé en Java, dont la simplicité d'utilisation relève du savoir [9].

I.7 Partie informatique

La partie informatique consiste à construire des algorithmes qui contrôlent la dynamique d'un quad-copter, qui seront implémenté sous Simulink. Afin de réaliser une plate-forme virtuelle de simulation qui étudie les mouvements et la stabilité du système, après avoir exploiter le modèle mathématique Newton-Euler d'une complexité minimale qui décrit les comportements et les différents mouvements, qui est représenté sous forme d'équations qui seront transcrites en algorithmes. Ensuite, ces derniers seront représentés sous forme de blocs Simulink qui forment un diagramme de contrôle. Ce système est contrôler par un régulateur PID, les paramètres de ce régulateur sont paramétrables à partir de la plate-forme générale.

Conclusion

Ce chapitre nous a permis d'avoir une vision globale des quad-copters, ce qui nous a permit de constater que le système du quad-copter est divisé en trois parties : mécanique, électronique, et informatique, cette dernière fait l'objet de notre étude dans ce mémoire.

II

Modélisation d'un quad-copter

Introduction

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

Ce chapitre porte sur la modélisation mathématique de la dynamique du quad-copter en se basant sur les équations de mouvement de Newton. Dans un premier temps, nous présentons les repères de référence nécessaires pour exprimer l'orientation du quad-copter. Dans un deuxième temps nous donnons la définition de toutes les grandeurs physiques nécessaires à la modélisation. Pour finir, nous identifions toutes les forces et les moments ayant un impact sur le modèle du quad-copter.

II.1 Modélisation

II.1.1 C'est quoi une modélisation ?

La modélisation : Un problème bien posé, un résultat concret. Les mathématiques consistent d'abord en un langage, qui permet de transcrire des problèmes de nature quantitative : c'est la modélisation. Une fois cette transcription est faite, des outils sont disponibles pour résoudre ces problèmes, partiellement ou complètement. On ramène ensuite la solution dans son contexte d'origine [10].

II.1.2 Pourquoi vouloir modéliser ?

Réaliser une modélisation signifie avant tout chercher à comprendre ce qui se passe, ne pas se contenter d'une solution empirique. Modéliser un processus, c'est le décrire de manière scientifique, quantitative, par exemple en termes d'équations (physiques, chimiques, etc). Cela permet d'en étudier l'évolution, d'en simuler des variantes, en modifiant certains paramètres [10].

II.2 Description générale du quad-copter

Les quatre rotors d'un quad-copter sont généralement placés aux extrémités d'une croix, et l'électronique de contrôle est habituellement placé au centre de la croix. Afin d'éviter à l'appareil de tourner sur lui même ou autour de son axe z (lacet), il est nécessaire que deux hélices tournent dans un sens, et les deux autres dans l'autre sens. Pour pouvoir diriger l'appareil, il est nécessaire que chaque couple d'hélices tourne dans le même sens, soit placé aux extrémités opposées d'une branche de la croix.

Le fonctionnement d'un quad-copter est assez particulier. En faisant varier astucieusement la puissance des moteurs, il est possible de le faire monter/descendre, de l'incliner à gauche/-droite (roulis) ou en avant/arrière (tangage) ou encore de le faire pivoter sur lui-même (lacet), le quad-copter a six *DDL*, trois mouvements de rotation et trois mouvements de translation, ces six degrés doivent être commandés à l'aide de quatre déclencheurs seulement [11].

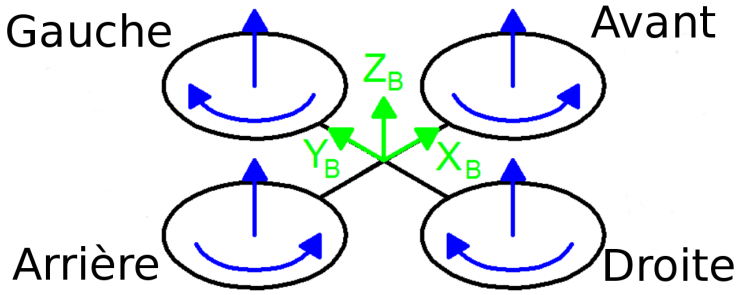


FIGURE II.1 – Structure générale d'un quad-copter [7]

La figure (II.1) montre le modèle dans un état stationnaire, où tous les hélices ont la même vitesse.

La structure d'un quad-copter est représentée en noir, le corps fixe *B-frame* est représenté en vert et le bleu représente les vitesses angulaires des hélices et les noms des variables de vitesse. Pour chaque hélice, deux flèches sont dessinées : les courbes représentent les sens de rotation, les autres représentent les vitesses. Ces derniers vecteurs pointes toujours vers le haut par conséquent, il ne suit pas le sens horaire [11].

II.3 Les mouvements du quad-copters

Dans les hélicoptères classiques, quand le rotor principal tourne, il produit un couple réactif qui inciterait le corps de l'hélicoptère à tourner dans la direction opposée si ce couple n'est pas contrarié. Ceci est habituellement fait en ajoutant un rotor de queue qui produit une poussée dans une direction latérale. Cependant, ce rotor avec son alimentation électrique associée ne fait aucune contribution à la poussée. Par contre, en cas de quad-copter, le rotor droit et le rotor gauche tournent dans le sens des aiguilles d'une montre et dans la direction opposée les rotors avant et arrière, ceci permet au véhicule de planer sans tourner hors de la commande.

Les mouvements de base de quad-copter sont réalisés en variant la vitesse de chaque rotor, de ce fait la poussée est produite. Le quad-copter incline vers la direction du rotor plus lent, qui tient compte alors de la translation le long de cet axe. Par conséquent, le quad-copter ne peut pas réaliser la translation sans roulis ou tangage, ce qui signifie qu'un changement de la vitesse d'un rotor se traduit dans un mouvement en au moins trois degrés de liberté. Par exemple, l'augmentation de la vitesse du propulseur gauche aura comme conséquence un mouvement de roulis. Nous pouvons commander les six degrés de liberté de quad-copter avec seulement quatre commandes, ce qui implique les quatre mouvements principaux suivants [11] :

- Mouvement vertical
- Mouvement de roulis
- Mouvement de tangage
- Mouvement de lacet

II.3.1 Mouvement vertical

Afin de planer, toute la force de portance devrait seulement être le long de l'axe z avec une grandeur exactement opposée à la force de pesanteur. Par conséquent, la poussée produite par chaque rotor doit être identique.

Les mouvements ascendant et descendant sont obtenus par la variation de la vitesse de rotation des moteurs par conséquence la poussée produite, si la force de portance est supérieure au poids du quad-copter le mouvement est ascendant, et si la force de portance est inférieure au poids du quad-copter le mouvement est descendant [11].

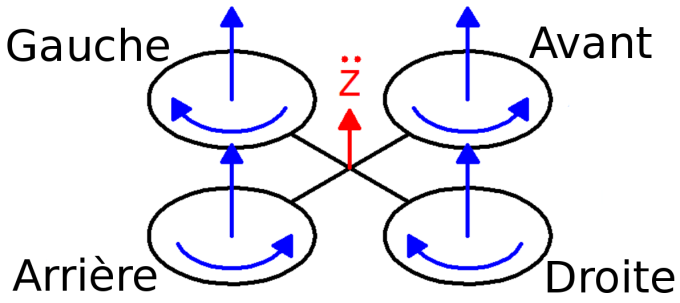


FIGURE II.2 – Illustration du mouvement vertical [7]

Les flèches bleu, montrent les vitesses des hélices, $\ddot{Z}$ représente l'accélération par rapport à l'axe z [11].

II.3.2 Mouvement de roulis

Le mouvement de roulis est obtenu en augmentant (ou en diminuant) la vitesse d'hélice gauche et en diminuant (ou en augmentant) la vitesse d'hélice droit. Elle conduit à un couple qui fait tourner le quad-copter par rapport à l'axe x_B . L'ensemble des poussées verticales est le même que du vol stationnaire, donc cette commande conduit seulement à une accélération angulaire de roulis [11].

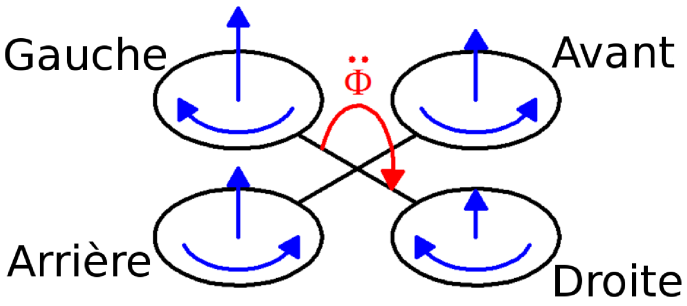


FIGURE II.3 – Illustration du mouvement de roulis [7]

II.3.3 Mouvement de tangage

Ce mouvement est très ressemblant à ce lui du roulis et il est obtenu en augmentant (ou en diminuant) la vitesse d'hélice arrière et en diminuant (ou en augmentant) la vitesse d'hélice d'avant. Elle conduit à un couple par rapport à l'axe y_B qui fait tourner le quad-copter [11].

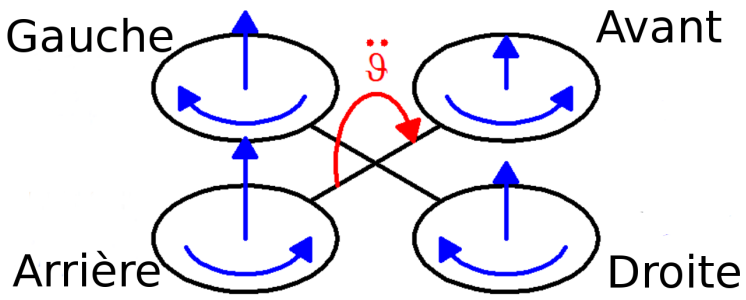


FIGURE II.4 – Illustration du mouvement de tangage [7]

$\ddot{\theta}$ représente l'accélération selon par rapport à l'axe y [11].

II.3.4 Mouvement de lacet

Le mouvement de lacet est obtenu en augmentant (ou en diminuant) la vitesse des hélices avant-arrière et en diminuant (ou en augmentant) la vitesse des hélices gauche-droit. Elle conduit à un couple qui a un rapport avec l'axe z_B qui fait tourner le quad-copter. Le mouvement de lacet est produit grâce au fait que les hélices gauche-droit tournent dans le sens horaire tandis que les hélices avant-arrière tournent dans le sens antihoraire.

Ainsi, lorsque le couple global est déséquilibré, le quad-copter tourne sur lui-même autour de z_B. Cette commande elle conduit seulement à une accélération de l'angle de lacet [11].

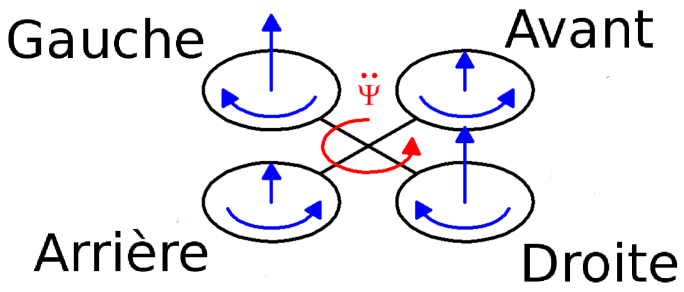


FIGURE II.5 – Illustration du mouvement de Lacet [7]

II.4 Le modèle dynamique

Le modèle dynamique d'un quad-copter est important pour la simulation du mouvement et pour analyser la structure d'un robot afin de développer un algorithme de contrôle. Les

chercheurs ont découvert des approches différentes pour calculer la dynamique d'un robot, mais en général il y a deux méthodes pour déterminer les équations du mouvement d'un quad-copter [12] :

- Modèle d'Euler-Lagrange.
- Modèle Newton-Euler.

II.4.1 Euler-Lagrange versus Newton-Euler

Dans la formulation standard d'Euler-Lagrange le quad-copter est traité dans son ensemble, et le système est analysé a base de son énergie cinétique et potentielle. La formulation de Newton-Euler est tout à fait différente. Il y a d'abord une récursion vers l'avant décrivant son mouvement linéaire et angulaire, puis une récursion en arrière pour calculer les forces et les couples.

Les deux formulations sont dérivées des premiers principes dans [13], [14] et [12], Ces derniers contiennent des exemples de la façon dont les méthodes peuvent être appliquées.

Il y a aucune réponse claire à la question quelle méthode est meilleure que l'autre. L'objectif principal est de dériver le modèle dynamique rapidement, chaque méthode dépend de plusieurs facteurs dans la chaîne cinématique.

Le modèle dynamique Newton-Euler est le choix préférable des quad-copters avec beaucoup de degrés de liberté. L'avantage de ce modèle est qu'il produit un modèle de façon récursive qui est en général plus rapide en calcul et en commande [12].

II.4.2 Modèle dynamique Newton-Euler

La modélisation des robots volants est une tache délicate puisque la dynamique du système est fortement non linéaire. Cette section fournit des informations sur le quad-copter à partir du modèle Newton-Euler [7].

Définition des repères

Pour décrire la position et l'orientation du quad-copter, nous avons besoin de deux repères. Le premier est nommé le repère inertiel (*E-frame*). Il s'agit d'un référentiel orthogonal fixe de

type galiléen, au sens que celui-ci n'accélère pas et ne tourne pas par rapport à un observateur. La définition d'un deuxième repère est nécessaire pour décrire l'orientation du quad-copter. Celui-ci est attaché au châssis du quad-copter et se déplace donc avec celui-ci. Il est dénommé le repère du corps-fixe (*B-frame*).

Modèle mathématique

Le système dynamique est écrit dans cadre de référence hybride H-frame, il est composer d'équation de vitesse linéaire dans E-frame et d'équation angulaire dans B-frame.

Cette référence est adoptée car il est facile d'exprimer la dynamique d'un système combiner avec le contrôle dans une seule référence.

l'équation II.1 montre le vecteur vitesse généralisé ζ dans H-frame [7].

$$\zeta \;=\; \begin{bmatrix} \dot{\Gamma^E} & \omega^B \end{bmatrix}^T \;=\; \begin{bmatrix} \dot{X} & \dot{Y} & \dot{Z} & p & q & r \end{bmatrix}^T \tag{II.1}$$

Où,

$\dot{\Gamma^E}$ Est le vecteur vitesse linéaire dans E-frame.

ω^B Est le vecteur de vitesse angulaire dans B-frame.

$\dot{X}$ Vitesse linéaire autour de x_E.

$\dot{Y}$ Vitesse linéaire autour de y_E.

$\dot{Z}$ Vitesse linéaire autour de z_E.

p Vitesse angulaire autour de x_B.

q Vitesse angulaire autour de x_B.

r Vitesse angulaire autour de x_B.

La dynamique du système dans *H-frame* peut être récrite sous une forme matricielle selon l'équation (II.2) [1].

$$M_H\;\dot{\zeta} \;+\; C_H(\zeta)\;\zeta \;=\; G_H \;+\; O_H(\zeta)\;\Omega \;+\; E_H(\xi)\;\Omega^2 \tag{II.2}$$

Où,

M_H Est la matrice d'inertie.

1. Pour plus de détail Voir *T. Bresciani, Modeling, Identification and control of quadrotor helicopter, pages 12-21* [7]

$\dot{\zeta}$ Est le vecteur d'accélération .

$C_H(\zeta)\ \zeta$ Est la matrice de coriolis.

G_H Présente le vecteur de gravitation il affecte juste sur les équations linéaire et non pas sur les équation angulaires car c'est une force et non pas un couple.

$O_H(\zeta)$: Représente la matrice gyroscopique, elle est produite par la rotation des hélices.

$E_H(\xi)$ Représente les forces qui agissent sur le système (le poids du quad-copter, Les forces de poussée et Les forces de traînée).

G_H, $O_H(\zeta)$ et $E_H(\xi)$ Représentent les effets physiques qui agissent sur le quad-copter [7].

En réarrangeant l'équation ((II.2)), il est possible d'isoler la dérivé du vecteur vitesse généralisé $\dot{\zeta}$.

$$\dot{\zeta} = M_H^{-1}\ (-\ C_H(\zeta)\ \zeta\ +\ G_H\ +\ O_H(\zeta)\ \Omega\ +\ E_H(\xi)\ \Omega^2) \tag{II.3}$$

On peut récrire l'équation((II.3)) sous forme d'un système d'équations, selon L'équation((II.4)).

$$\begin{cases} \ddot{X} &= (\sin\psi\sin\phi + \cos\psi\sin\theta\cos\phi)\frac{U_1}{m} \\ \ddot{Y} &= (-\cos\psi\sin\phi + \sin\psi\sin\theta\cos\phi)\frac{U_1}{m} \\ \ddot{Z} &= -g + (\cos\theta\cos\phi)\frac{U_1}{m} \\ \dot{p} &= \frac{I_{YY}-I_{ZZ}}{I_{XX}}\ q\ r\ -\ \frac{J_{TP}}{I_{XX}}\ q\ \Omega\ +\frac{U_2}{I_{XX}} \\ \dot{q} &= \frac{I_{ZZ}-I_{XX}}{I_{YY}}\ p\ r\ -\ \frac{J_{TP}}{I_{YY}}\ P\ \Omega\ +\frac{U_3}{I_{YY}} \\ \dot{r} &= \frac{I_{XX}-I_{YY}}{I_{ZZ}}\ p\ q\ +\frac{U_4}{I_{ZZ}} \end{cases} \tag{II.4}$$

Où,

$\ddot{X}$ Accélération linéaire autour de x_E dans E-frame.

$\ddot{Y}$ Accélération linéaire autour de y_E dans E-frame.

$\ddot{Z}$ Accélération linéaire autour de z_E dans E-frame.

$\dot{p}$ Accélération angulaire autour de x_B dans B-frame.

$\dot{q}$ Accélération angulaire autour de y_B dans B-frame.

$\dot{r}$ Accélération angulaire autour de z_B dans B-frame.

Le second système d'équation explique comment les mouvements basiques sont reliés à la vitesse carré des hélices.

$$\begin{cases} U_1 &= b\ (\Omega_1^2 + \Omega_2^2 + \Omega_3^2 + \Omega_4^2) \\ U_2 &= l\ b\ (-\Omega_2^2 + \Omega_4^2) \\ U_3 &= l\ b\ (-\Omega_1^2 + \Omega_3^2) \\ U_4 &= d\ (-\Omega_1^2 + \Omega_2^2 - \Omega_3^2 + \Omega_4^2) \\ \Omega &= -\Omega_1^2 + \Omega_2^2 - \Omega_3^2 + \Omega_4^2 \end{cases} \tag{II.5}$$

U_1 La poussée vertical dans B-frame.

U_2 Couple de roulis dans B-frame.

U_3 Couple de tangage dans B-frame.

U_4 Couple de lacet dans B-frame.

Avec cette approche, en théorie il est possible de déterminer la position du quad-copter par un double intégral de ses accélérations linéaire et angulaire. Pour effectuer cette opération, il faut gérer le voltage des quatre moteur. Ce processus est appelé cinématique directe ou dynamique directe.

Le but de la stabilisation est de trouver des voltages adéquat pour maintenir le quad-copter dans certains positions demandé par le contrôle distant. Ce processus est appelé cinématique inverse et dynamique inverse. Contrairement aux cinématique directe et dynamique directe, les opérations inverses ne sont pas toujours possible est unique. Pour ces raisons il est beaucoup plus compliqué.

La dynamique du quad-copter doit être simplifié au maximum pour qu'elle nous fournit un modèle inverse facile, qui est implémenté dans les algorithmes de contrôle. La figure suivante montre l'architecture complète du système de contrôle de position et d'altitude [7].

II.5 Moteur à courant continu (Brushless)

Un moteur DC est tout simplement un actionneur qui convertit l'énergie électrique en une énergie mécanique. Il est formé de deux circuits électromagnétiques interactifs, le premier (appelé le rotor) est libre de tourner autour du second (appelé le stator) qui est fixé à la place.

Dans le rotor, plusieurs groupes d'enroulement de cuivre sont connectés en série et sont

accessibles de l'extérieur grâce à un dispositif appelé collecteur. Dans le stator, deux ou plusieurs aimants permanents imposent un champ magnétique qui affecte le rotor.

En appliquant un flux de courant continu dans les enroulements, le rotor tourne en raison de la force générée par l'interaction électrique et magnétique.

Le moteur à courant continu a un modèle bien connu qui lie la partie électrique à la partie mécanique. Ce modèle est composé d'une résistance $R(\omega)$, d'une inductance $L(h)$ et d'un générateur $E(V)$ [7].

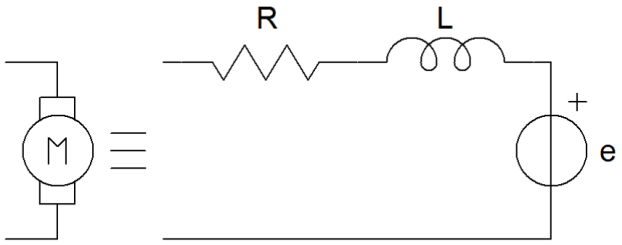


FIGURE II.6 – Le modèle du moteur DC

L'équation (II.6) montre la dynamique du moteur et présente la relation entre la vitesse des hélices et le voltage du moteur.

$$\dot{\Omega} \;=\; A_p \;\Omega \;+\; B_p \;\upsilon \;+\; C_p \tag{II.6}$$

Où,

$\dot{\Omega}$ Est le vecteur d'accélération des hélices.

Ω Est le vecteur vitesse des hélices.

υ Vecteur de tension d'entrée.

A_p Est le coefficient de vitesse de l'hélice. B_p Est le coefficient de tension d'entrée.

C_p Est un coefficient constant [2].

2. Pour plus de détail Voir *T. Bresciani, Modeling, Identification and control of quadrotor helicopter, pages 22-27* [7]

Conclusion

Dans ce chapitre nous avons effectué la modélisation dynamique d'un quad-copter. Dans les sections (II.2) et (II.3) nous avons établi une description générale sur les quad-copters et une étude détaillée sur ses différents mouvements. Dans la section (II.4), nous avons élaboré le modèle dynamique Newton-Euler qui nous a aidé à avoir un modèle général simple à implémenté. Dans la section (II.5) nous avons présenté les moteurs à courant continu et leur utilité dans la composition d'un quad-copter.

Dans le chapitre suivant nous allons étudier les différents algorithmes de contrôle pour évaluer les performances et les différents comportements d'un quad-copter.

III

Algorithmes de contrôle

Introduction

A fin de pouvoir piloter et gérer un quad-copter, plusieurs algorithmes de contrôle peuvent être appliqués pour construire des lois de commande. Cependant, ces approches souffrent de dégradation des performances lorsque l'aéronef s'éloigne de ces points d'équilibres. De plus, la présence de perturbation peut déstabiliser ces véhicules. Pour cela, on peut employer des techniques adaptatives pour avoir un système stable.

Ce chapitre s'articule sur trois sections, la premier section consiste a présenter les différents régulateurs qu'ils existent, afin de choisir un régulateur convenable a ce système. Dans la deuxième section nous allons voir comment contrôler les différents mouvements en utilisant la technique PID tout en testent le modèle de simulation réalisé sous MATLAB, où il facile d'évaluer les performances du système avec une approche mathématique. Dans la dernière section consiste à réaliser une plate-forme du quad-copter pour évaluer le comportement du système réel.

III.1 Asservissement et régulation

Un asservissement est un algorithme dont l'objet principal est d'atteindre le plus rapidement possible et de limiter l'écart par rapport à sa valeur de consigne, quelles que soient les perturbations externes. Le principe général est de comparer la consigne et l'état du système de manière à le corriger efficacement. La régulation (ou asservissement) consiste à agir de façon à ce que une mesure soit égale à une consigne. Si l'on cherche à atteindre une consigne, on parlera de poursuite ou asservissement ; si l'on cherche à éliminer des perturbations pour qu'une valeur reste constante, on parlera de régulation [15]. Il existe plusieurs types de régulateur :

III.1.1 Régulateur proportionnel P

Dans le cas d'un contrôle proportionnel, l'erreur est virtuellement amplifiée d'un certain gain constant qu'il conviendra de déterminer en fonction du système.

$$Consigne(t) = K_p.\varepsilon(t)$$

Ce qui en Laplace donne :

$$Consigne(p) = K_p.\varepsilon(p)$$

L'idée étant d'augmenter l'effet de l'erreur sur le système afin que celui-ci réagisse plus rapidement aux changements de consignes. Plus la valeur deK_p est grande, plus la réponse l'est aussi. En revanche, la stabilité du système s'en trouve détériorée et dans le cas d'un K_p démesuré le système peut même diverger. L'action proportionnelle applique une correction instantanée pour tout écart entre la mesure et la consigne, plus la perturbation est grande, plus la correction apportée est grande. Cette composante seule ne permet pas une grande précision surtout dans les systèmes à faible inertie [16].

III.1.2 Régulateur proportionnel-intégrale PI

Au contrôle proportionnel, nous pouvons ajouter l'intégration de l'erreur. Dans ce cas nous obtenons une régulation PI (proportionnelle et intégré). L'erreur entre la consigne et la mesure est ici intégrée par rapport au temps et multipliée par une constante qu'il faudra aussi déterminer en fonction du système.

$$Consigne(t) = K_p.\varepsilon(t) + K_i \int_0^t \varepsilon(\tau)d\tau$$

Ce qui en Laplace donne :

$$Consigne(p) = K_p.\varepsilon(p) + K_i + K_i.\frac{\varepsilon(p)}{p}$$

Pourquoi a-ton besoin de rajouter cette fonctionnalité à notre organe de contrôle ? Et bien, lors d'un simple contrôle proportionnel, il subsiste une erreur statique. Lorsque le système s'approche de sa consigne, l'erreur n'est plus assez grande pour faire avancer le moteur. Le terme intégral permet ainsi de compenser l'erreur statique et fournit, par conséquent, un système plus stable en régime permanent. Plus K_i est élevé l'erreur statique est élevé. Cette composante apporte une notion de temps d'intégration à la correction, cette notion de temps s'exprime généralement en seconde. Cette action est complémentaire à l'action proportionnelle, elle permet de stabiliser dans le temps l'action proportionnelle, plus l'erreur mesurée est constante plus la correction est constante [16].

III.1.3 Régulateur proportionnel-intégrale-dérivée PID

Pour obtenir un contrôle en PID, il nous faut encore rajouter un terme. Celui-ci consiste à dériver l'erreur entre la consigne et la mesure par rapport au temps et a le multiplier lui aussi par une constante [16].

III.2 Régulation PID

Dans le domaine de l'industrie, la plupart des régulateurs utilisés sont des régulateurs PID, Les raisons de ce succès sont [7] :

- Structure simple.
- Hautes performances dans plusieurs processus.

Dans la robotique, les techniques PID représentent les bases du contrôle. La structure traditionnelle PID est composé de trois facteurs, présentés dans la figure III.1 et l'équation (III.1)

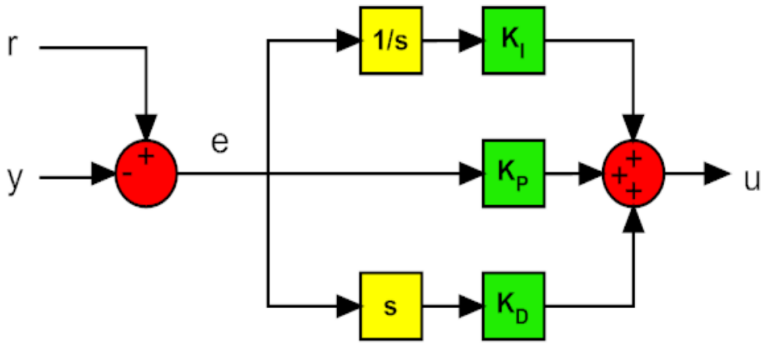


FIGURE III.1 – Structure du PID traditionnelle [7]

Les blocs "1/s" et "s" représente respectivement les opérations d'intégration et de dérivation [7].

$$u(t) = K_P e(t) + K_I \int_0^t e(\tau)d\tau + K_Z \frac{de(t)}{dt} \tag{III.1}$$

Où :

u : La variable générique contrôlée.

e : L'erreur entre la tâche r et la sortie du processus y.

K_P : Coefficient proportionnel.

K_I : Coefficient intégra.

K_D : Coefficient dérivé.

Le premier facteur (P) est proportionnel à l'erreur il définit la bande passante proportionnelle. À l'intérieur de cet interval, la sortie doit être proportionnelle à l'erreur, tandis que à l'extérieur la sortie doit être au minimum ou au maximum. Le deuxième facteur (I) varie selon l'intégral de l'erreur. Bien que ce composant augmente le dépassement et le temps de réglage, il a une propriété unique c'est que il élimine l'erreur de l'état de stabilité. Le troisième facteur (D) varie selon la dérivé de l'erreur, ce composant aide à réduire les dépassements et le temps de réglage [7].

Dans le domaine de Laplace, la structure PID traditionnelle peut s'écrire sous la forme suivante :

$$u(s) = \Big(K_P + \frac{K_I}{s} + sK_D\Big)e(s)$$

Vu que cette fonction est impropre, physiquement ce n'est pas faisable à cause de la dérivabilité du terme. Après une certaine fréquence, la contribution du D doit être atténuée pour filtrer le bruit de la bande passante. Pour cette raison, Dans le dérivateur réel un pôle est ajouté comme le montre l'équation ci-dessous [7] :

$$u(s) = \Big(K_P + \frac{K_I}{s} + \frac{sK_D}{1 + sK_D/(kK_p)}\Big)e(s)$$

La structure *PID* traditionnelle présente deux principaux inconvénients :

- L'action de dérivation est calculée à partir de l'erreur. Si la tâche ajoute une étape dans la référence, la sortie du dérivateur peut présenter une impulsion. Le *mouvement pointu* peut saturer l'actionneur et propager loin du système par rapport à la zone linéaire. Pour cette raison la plupart des architectures *PID* présentent l'action de dérivé de la sortie du processus seulement.

- L'action de l'intégration combiné avec un actionneur de saturation peut provoquer un effet non linéaire, celui-ci peut diminuer la performance du système de contrôle. Lorsque la valeur de l'intégrale est assez grande et l'erreur change de signe il est nécessaire d'attendre beaucoup de temps pour restaurer le comportement en tant que système linéaire. Ce phénomène est appelé Intégrante wind-up. pour l'éviter, on ajoute un saturateur à

l'intégrale pour limiter les valeurs maximales et minimales. La figure(III.2) montre la structure *PID* améliorée [7].

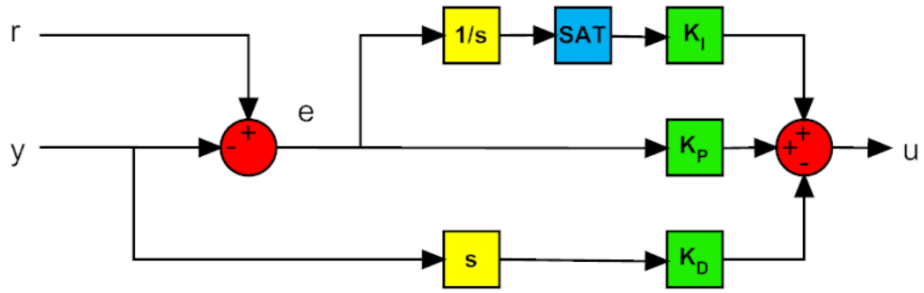


FIGURE III.2 – Structure du PID amélioré [7]

Le bloc "SAT" représente le saturateur.

La description de quatre algorithmes internes à l'aide des diagramme de bloc de contrôle pour la stabilisation de la hauteur et d'altitude sont présentés ci-après :

III.2.1 Contrôle de roulis

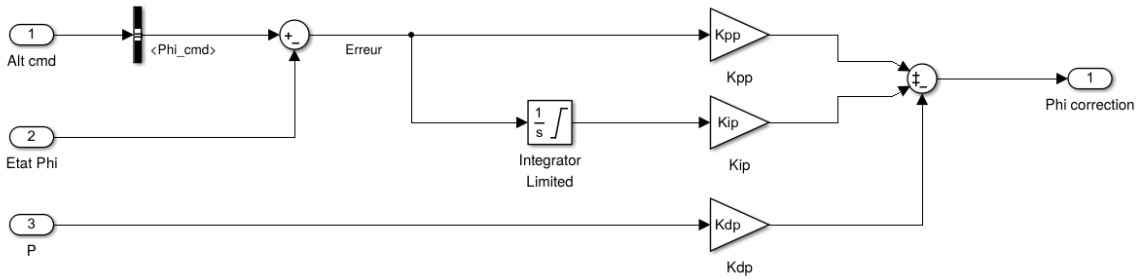


FIGURE III.3 – Diagramme de contrôle de roulis

où :

Alt_cmd Contient la valeur de l'angle ϕ_cmd désiré.

Etat phi L'angle de roulis mesuré.

P La vitesse angulaire du quad-copter suivant l'axe x.

$Phi_correction$ Vitesse de roulis corrigé.

K_{pp} , K_{ip} , K_{dp} Les trois paramètre de contrôle.

III.2.2 Contrôle de lacet

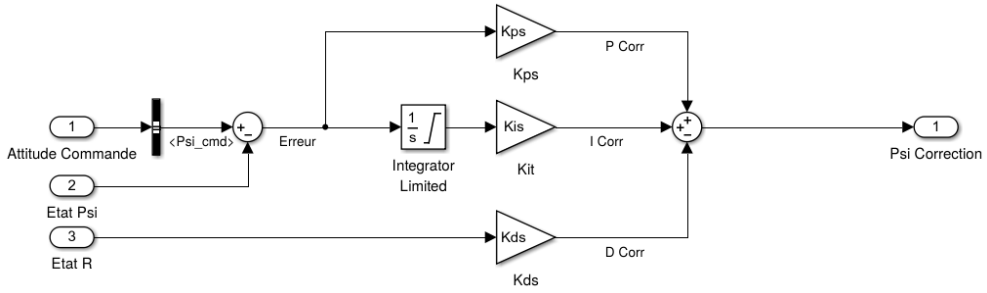


FIGURE III.4 – Diagramme de contrôle de lacet

Où :

AltitudeCommande Contient la valeur de l'angle ψcmd désiré.

Etat psi L'angle de lacet mesuré.

Etat R La vitesse angulaire du quad-copter suivant l'axe y.

Psi correction Vitesse de lacet corrigé.

K_{ps} , K_{is} , K_{ds} Les trois paramètre de contrôle.

III.2.3 Contrôle de tangage

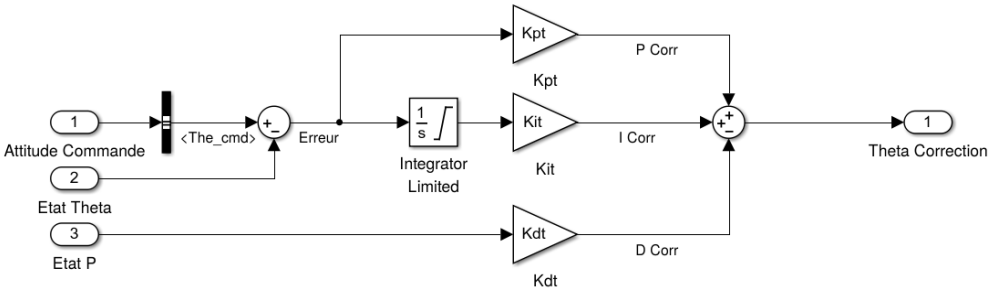


FIGURE III.5 – Diagramme de contrôle de tangage

où :

Altitude Comande Contient la valeur de l'angle *the cmd* désiré.

Etat theta L'angle de tangage mesuré.

Etat P La vitesse angulaire du quad-copter suivant l'axe z.

Theta correction Vitesse de tangage corrigé.

K_{pt} , K_{it} , K_{dt} Les trois paramètre de contrôle.

III.2.4 Contrôle de hauteur

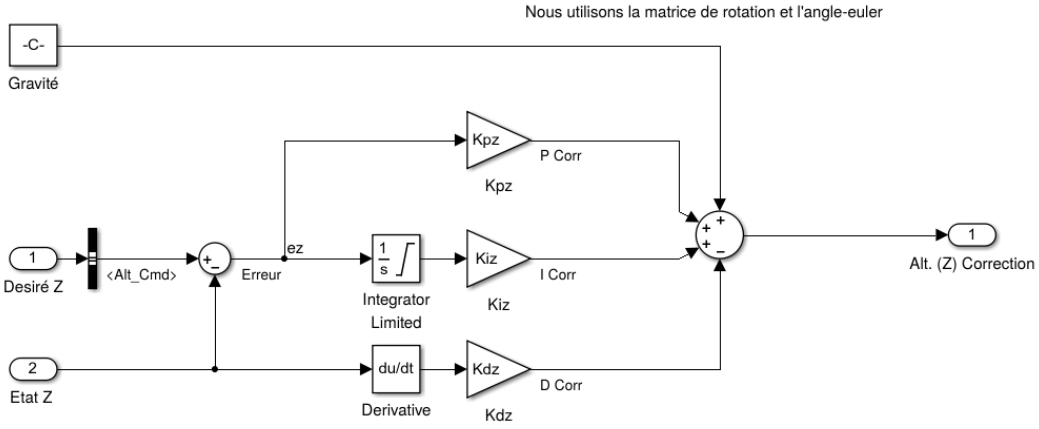


FIGURE III.6 – Diagramme de contrôle de hauteur

où :

Altitude Comande Contient la valeur de l'angle *Alt Cmd* la hauteur désiré.

Etat Z Hauteur mesuré par les capteurs.

C La force de gravité .

Theta correction Vitesse de roulis requis.

K_{pt} , K_{it} , K_{dt} Les trois paramètre de contrôle.

III.3 Structure du système

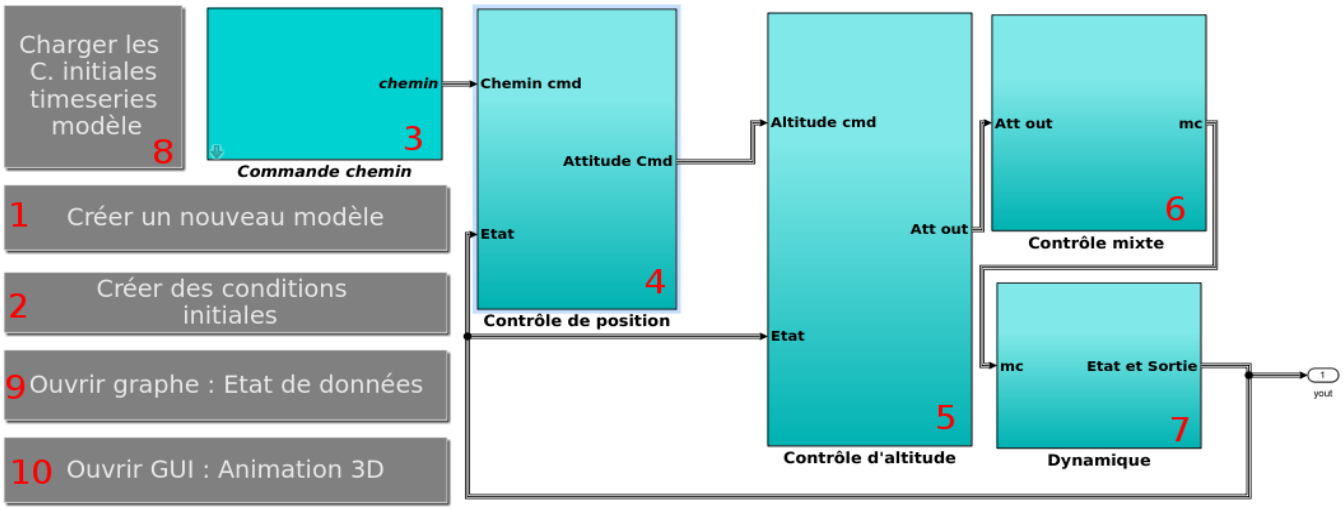


FIGURE III.7 – Plate-forme générale sous simulink

III.3.1 Créer un nouveau modèle

(Numéro 1 sur la figure (III.7)) Est un bouton qui ouvre l'interface qui sert à créer un nouveau modèle d'un quad-copter. La figure suivante est une capture de cette interface :

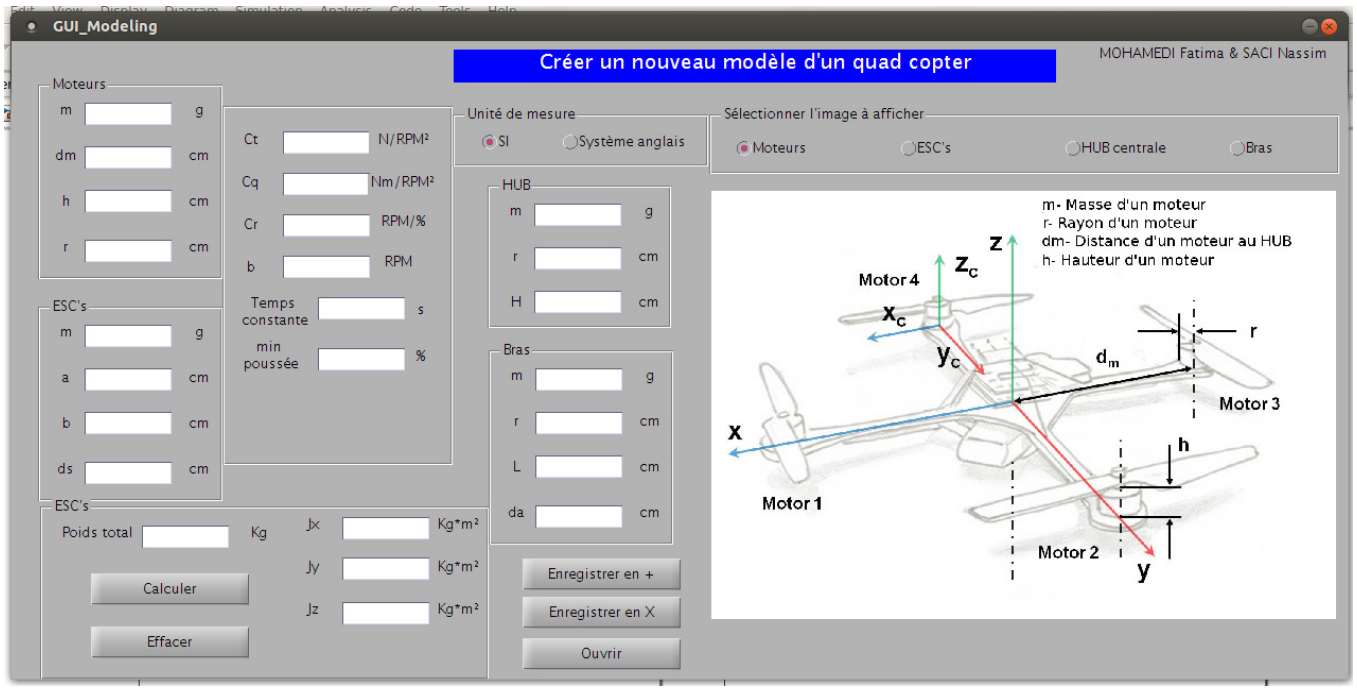


FIGURE III.8 – Interface “Créer un nouveau modèle”

Elle contient des champs de saisie pour introduire les différentes valeurs du modèle à créer tel que la masse, le rayon et la hauteur d'un moteur elle permet de calculer la masse totale du quad-copter et les valeurs de la matrice d'inertie $J_x,\ J_y,\ J_z$ selon [7]. Elle contient trois boutons pour ouvrir un modèle, enregistrer en '+' ou 'X' la structure du modèle sous un fichier « .mat ».

III.3.2 Créer des conditions initiales

(Numéro 2 sur la figure (III.7)) Est un bouton qui ouvre l'interface qui sert à créer des conditions initiales. La figure suivante est une capture de cette interface :

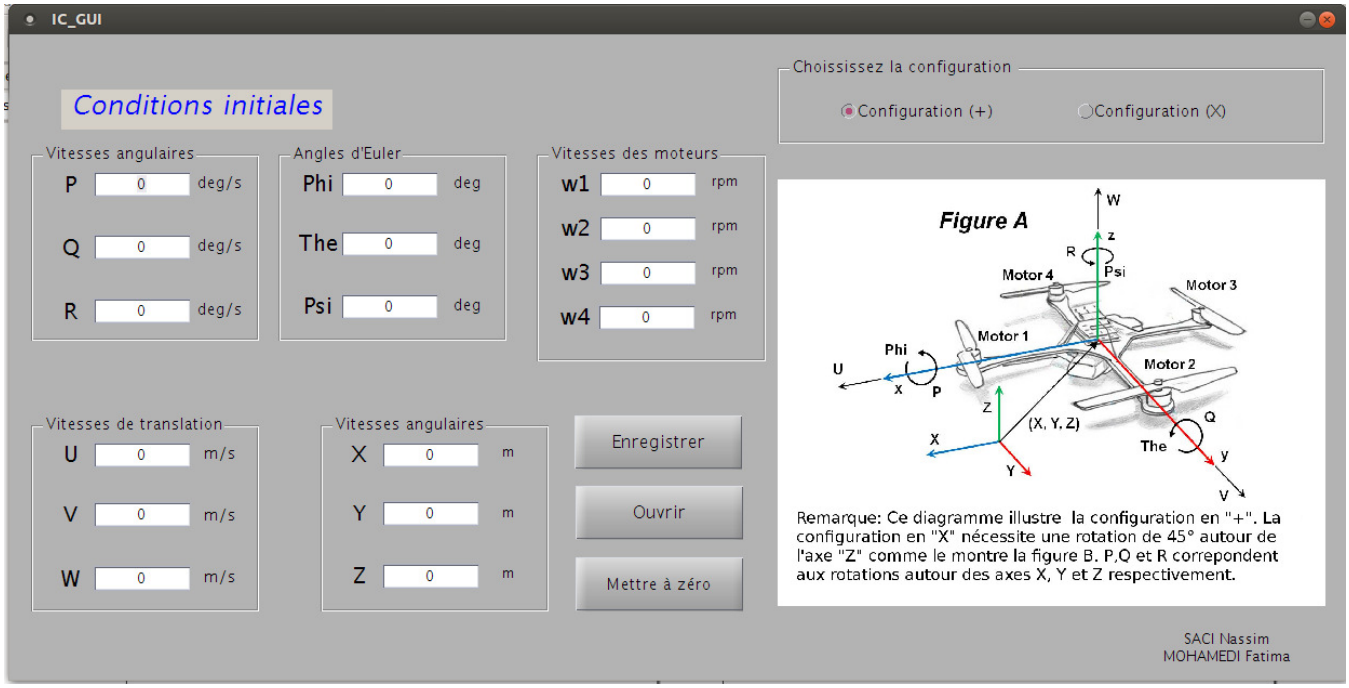


FIGURE III.9 – Interface “créer des conditions initiales”

Cette interface permet de créer des conditions initiales de la simulation tel que les vitesses angulaires des hélices initiales, les vitesses de translation initiales, les positions par rapport aux axes x, y, z, et les angles d’Euler initiales et de les enregistrer sous un fichier « .mat »

III.3.3 Commande chemin

(Numéro 3 sur la figure (III.7)) Est le premier bloc dans la chaîne de contrôle. De type struct il associe les valeurs de x, y, z à chaque instant, à l’aide du type (classe) timeseries, les données sont stockées sous la forme d’un attribut (Data) et le temps dans le attribut (Time).

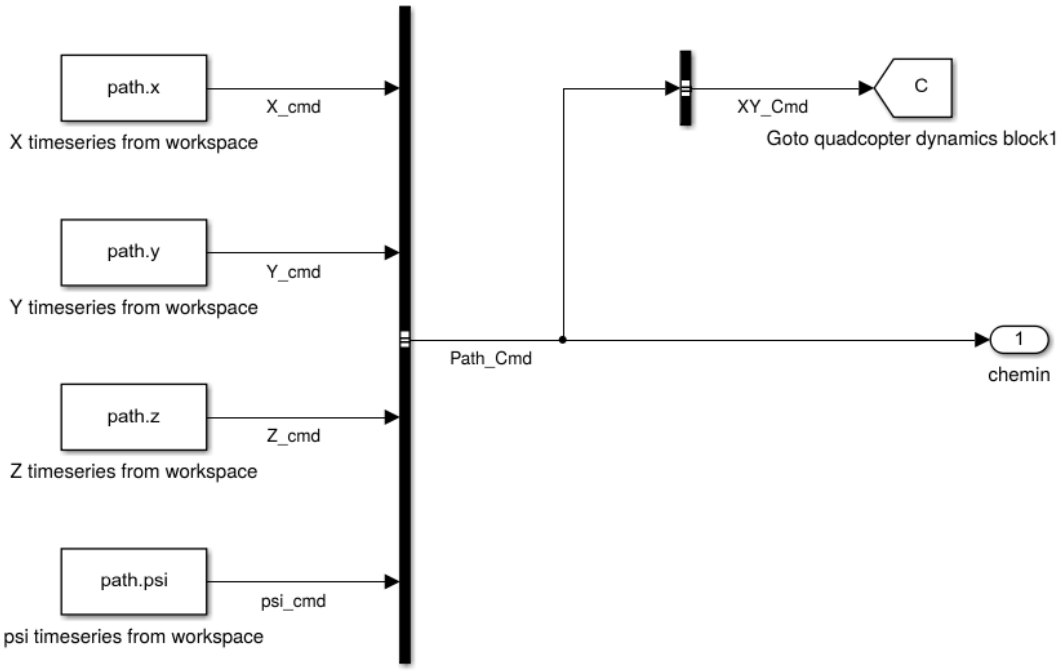


FIGURE III.10 – Diagramme de contrôle

Ce type de données est généré à partir d'un script MATLAB et stocké sous un fichier « .mat ».

III.3.4 Contrôle de position

(Numéro 4 sur la figure (III.7)) Est le deuxième bloc dans la chaîne de contrôle. Il fournit la commande de l'altitude des deux valeurs des angles θ, ϕ à partir de l'angle ψ donné par le bloc précédent. Cette commande sera traitée dans le bloc suivant. Grâce au régulateur PID on peut trouver les valeurs exactes de ces deux angles selon [7] et [8].

III.3.5 Contrôle d'altitude

(Numéro 5 sur la figure (III.7)) Chargé de calculer et de corriger l'erreur des différents mouvements. Il a comme entrée la commande d'altitude et l'état, il fournit en sortie les quatre correction des quatre différents mouvements roulis, tangage, lacet, altitude.

III.3.6 Contrôle mixte

(Numéro 6 sur la figure (III.7)) Il existe deux configurations possibles :

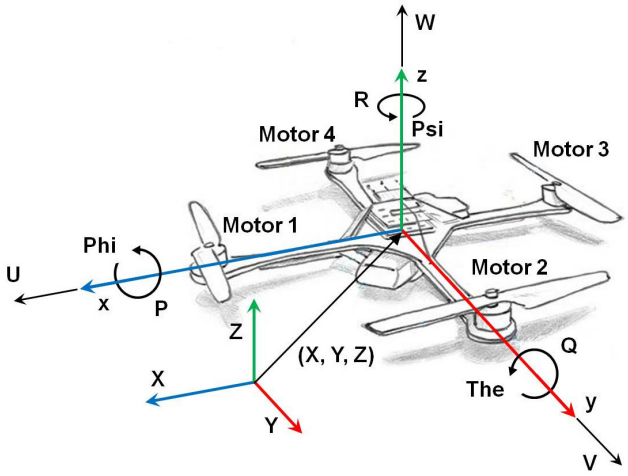


FIGURE III.11 – Configuration en '+'

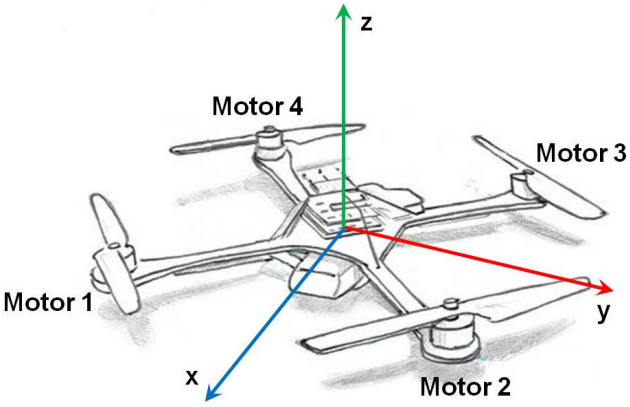


FIGURE III.12 – Configuration en 'X'

Configuration en '+'

On place l'axe x en parallèle avec le bras qui porte le moteur 1. La rotation du quad-copter autour de l'axe $x,\ y,\ z$ correspond aux angles d'Euler $\phi,\ \theta,\ \psi$ en degrés et leurs vitesses angulaires sont $P,\ Q,\ R$ en degrés/seconde respectivement. $U,\ V,\ W$ en mètre/seconde est la vitesse de translation par rapport aux axes $x,\ y,\ z$ respectivement.

Les deux blocs ont pour entrée la correction de l'altitude Z et les trois corrections $\psi,\ \theta,\ \phi$, et il fournit en sortie le pourcentage de force qui doit produire chaque moteur. L'équation qui définit cette configuration est la suivante selon [7] :

$$\begin{cases} mc1 = Altitude_{correction} - \psi_{correction} - \theta_{correction} \\ mc2 = Altitude_{correction} - \psi_{correction} - \theta_{correction} \\ mc3 = Altitude_{correction} - \psi_{correction} - \theta_{correction} \\ mc4 = Altitude_{correction} - \psi_{correction} - \theta_{correction} \end{cases} \tag{III.2}$$

Configuration en 'X'

La configuration en 'X' nécessite une rotation de 45° autour de l'axe z. C'est à dire en plaçant l'axe x entre le moteur 1 et 2.

L'équation qui définit cette configuration est la suivante selon [7] :

$$\begin{cases} mc1 = Altitude_{correction} - \theta_{correction} - \phi_{correction} - \psi_{correction} \\ mc2 = Altitude_{correction} + \theta_{correction} + \phi_{correction} - \psi_{correction} \\ mc3 = Altitude_{correction} + \theta_{correction} - \phi_{correction} + \psi_{correction} \\ mc4 = Altitude_{correction} - \theta_{correction} + \phi_{correction} + \psi_{correction} \end{cases} \tag{III.3}$$

III.3.7 Dynamique

(Numéro 7 sur la figure (III.7)) C'est l'implémentation de la dynamique du quad-copter. Il est composé de trois sous systèmes le premier à pour but de définir la dynamique des moteur c'est à dire la représentation mathématique du moteur selon [7] et [8], et l'équation principale (Équation d'état) qui a comme entrée les vitesses des quatre moteurs en rotation/-minute(RPM), et le bruit généré hors hasard par le bloc disturbance produit du bruit.

III.3.8 Charger les C.initiales

(Numéro 8 sur la figure (III.7)) Est un bouton qui charge les fichiers « .mat »sur le système de contrôle générale.

III.3.9 Ouvrir graphe : Etat des données

(Numéro 9 sur la figure (III.7)) Est un bouton qui ouvre l'interface des différents graphes qu'on va utiliser dans le chapitre 4.

III.3.10 Ouvrir GUI : Animations 3D

(Numéro 10 sur la figure (III.7)) Est un bouton qui ouvre l'interface qui piste le comportement et dessine le quad-copter, pendant toute la période de la simulation en utilisant les données enregistrées dans la variable "yout". Grâce à l'outil 3D animation de simulink.

Conclusion

Au cours de ce chapitre, nous avons évoqué les algorithmes de contrôle qui gèrent les quad-copters. Dans la section (III.1) nous avons vu les différents types de régulateur. Dans la section (III.2) nous avons implémenter le régulateur PID sous matlab qui permet de générer des lois de commande pour la stabilisation et qui contrôle les mouvements. Dans la section (II.4.2) nous avons présenté le programme principal implémentés sous Simulink pour évaluer ses comportements.

Dans le chapitre qui suit, nous présentons les résultats de simulations menés sur le modèle dynamique complet, toute en faisons variées les valeurs des paramètres PID.

IV

Simulation d'un quad-copter

Introduction

La simulation va nous permettre d'étudier les résultats et les performances d'un quad-copter de façon virtuelle, sans réaliser l'expérience et le testé sur un quad-copter réel.

Dans ce chapitre, nous allons simuler le modèle expliqué dans le chapitre précédent, et discuter les différents résultats qu'on a obtenu avec les différentes valeurs.

Ce chapitre a pour but d'illustrer la déstabilisation et la stabilisation du quad-copter selon les valeurs du régulateur PID, pour le contrôle du mouvement d'altitude, roulis, lacet ou tangage. Les résultats sont présentées à l'aide de l'outil Scope sous simulink ou la commande plot dans un script matlab.

IV.1 Outils de développement

IV.1.1 MATLAB

Le logiciel MATLAB est un logiciel de manipulation de données numériques et de programmation dont le champ d'application est essentiellement les sciences appliquées. Son objectif, par rapport aux autres langages, est de simplifier au maximum la transcription en langage informatique d'un problème mathématique, en utilisant une écriture la plus proche possible du langage naturel scientifique. Le logiciel fonctionne sous Windows et sous Linux. Son interface de manipulation *HMI* utilise les ressources usuelles du multi-fenêtrage. Son apprentissage n'exige que la connaissance de quelques principes de base à partir desquels l'utilisation des fonctions évoluées est très intuitive grâce à l'aide intégrée aux fonctions [17].

IV.1.2 SIMULINK

SIMULINK est une extension de MATLAB qui permet aux ingénieurs de construire des modèles dynamiques rapidement et avec une haute précision en utilisant les notations des diagrammes de bloc. En utilisant SIMULINK, il est facile de modéliser des systèmes complexes non linéaires. Un modèle SIMULINK peut inclure des composants des systèmes continus, hybrides et discrets, ce modèle peut produire des animations et des graphes qui montrent le progrès de la simulation [17].

IV.2 Caractéristiques du quad-copter

Dans ce travail, nous allons utiliser un modèle qui définit un quad-copter durant toutes les simulations. le tableau suivant montre les valeurs utilisées :

Variables	**Valeurs**	**Unité**	**Description**
motor_m	0.0730	g	Masse d'un moteur avec l'hélice
motor_dm	0.223	cm	La distance du moteur par rapport au centre du quad-copter
motor_h	0.0318	cm	Hauteur d'un moteur par rapport au bras qu'il le porte
motor_r	0.140	cm	Rayon d'un moteur
ESC_m	0.0300	g	Masse d'un ESC
ESC_a	0.0254	cm	Largeur d'un ESC
ESC_b	0.0572	cm	Longueur d'un ESC
ESC_ds	0.0826	cm	Distance d'un ESC par rapport au centre du quad-copter
HUB_m	0.4310	g	Masse du HUB
HUB_r	0.0564	cm	Rayon du HUB
HUB_H	0.0429	cm	Hauteur du HUB
arms_m	0.0450	g	Masse d'un bras
arms_r	0.0325	cm	Rayon d'un bras
arms_l	0.1857	cm	Longueur d'un bras
arms_da	0.0508	cm	Distance entre le centre du quad-copter au debut d'un bras
Jx	0.0095	$Kg\ m^2$	Valeurs de la matrice d'inertie
Jy	0.0095	$Kg\ m^2$	
Jz	0.0186	$Kg\ m^2$	
g	9.8100	$m.\ s^{-2}$	Force de gravité
mass	1.0230	Kg	Masse totale
plusconfig	1	int	Sert à déterminer la configuration utilisé

TABLE IV.1 – Les caractéristiques du quad-copter [13]

IV.3 Conditions initiales

Nous allons créer des conditions initiales à partir de l'interface illustrée dans la figure (III.9). Le tableau suivant montre :

Variables	**Valeurs**	**Unité**	**Description**
P	20	deg/s	Vitesse angulaire par rapport à l'axe x
Q	20	deg/s	Vitesse angulaire par rapport à l'axe y
R	0	deg/s	Vitesse angulaire par rapport à l'axe z
U	0	m/s	Vitesse de translation par rapport à l'axe x
V	0	m/s	Vitesse de translation par rapport à l'axe y
W	0	m/s	Vitesse de translation par rapport à l'axe z
ϕ	0	deg	Angle d'Euler par rapport à l'axe x
θ	0	deg	Angle d'Euler par rapport à l'axe y
ψ	0	deg	Angle d'Euler par rapport à l'axe z
X	0	m	La position du quad-copter par rapport l'axe x dans le repère E-Frame
Y	0	m	La position du quad-copter par rapport l'axe y dans le repère E-Frame
Z	0	m	La position du quad-copter par rapport l'axe z dans le repère E-Frame
ω_1	0	rpm	La vitesse de rotation su moteur 1
ω_2	0	rpm	La vitesse de rotation su moteur 2
ω_3	0	rpm	La vitesse de rotation su moteur 3
ω_4	0	rpm	La vitesse de rotation su moteur 4

TABLE IV.2 – Les conditions initiales [13]

IV.4 Simulation et interprétation des résultats

Dans ce qui suit, nous présenterons les résultats de simulation obtenues. Nous intéressons aux performances dynamiques en faisant varier les valeurs des paramètres P,I,D, afin de trouver les valeurs convenables pour assurer un système stable, une étude comparative sera menée pour chaque test.

IV.4.1 Le premier test

Le tableau suivant montre les valeurs des trois paramètres P, I, D des quatre mouvements qu'on a choisi aléatoirement pour un premier test.

Mouvement	**P**	**I**	**D**
Roulis	1.8	1.5	1.5
Tangage	1.8	1.5	1.5
Lacet	3	1	3
Altitude	1.8	1.5	3.5

TABLE IV.3 – Le premier test

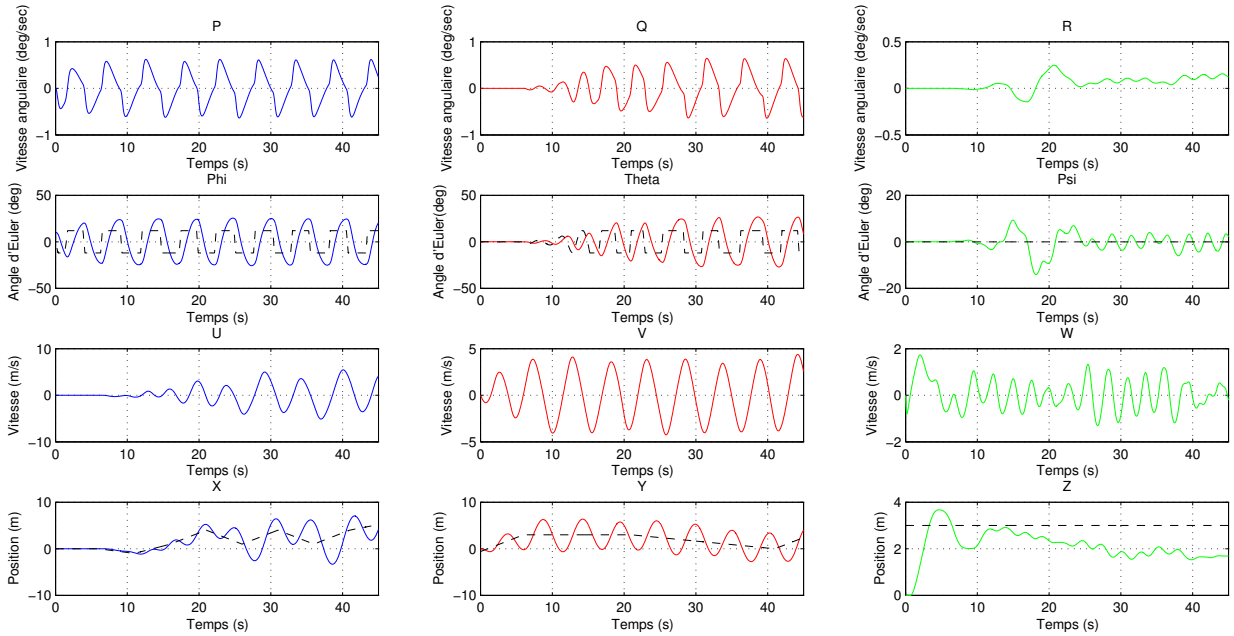


FIGURE IV.1 – Résultats du premier test

D'après la figure (IV.1)on remarque que :

les vitesses angulaires (P, Q, R), les angles d'Euler (ϕ,ψ,θ), les vitesses (U,V,W) sont oscillantes à l'infini donc le système est instable par rapport à la référence désirée qui est représenté par des lignes discontinues.

On remarque aussi que la position (X, Y, Z) ne suit pas la référence désirée qui est représentée par une ligne discontinue. Donc les valeurs du régulateur PID sont incorrect.

IV.4.2 Le deuxième test

Le tableau suivant montre les valeurs des trois paramètre P, I, D des quatre mouvements qu'on choisit pour le deuxième test.

Mouvement	**P**	**I**	**D**
Roulis	2.0	1.1	1.2
Tangage	1.8	1.5	1.5
Lacet	3	1	3
Altitude	1.8	1.5	3.5

TABLE IV.4 – Le deuxième test

La figure suivante montre les résultats du deuxième test :

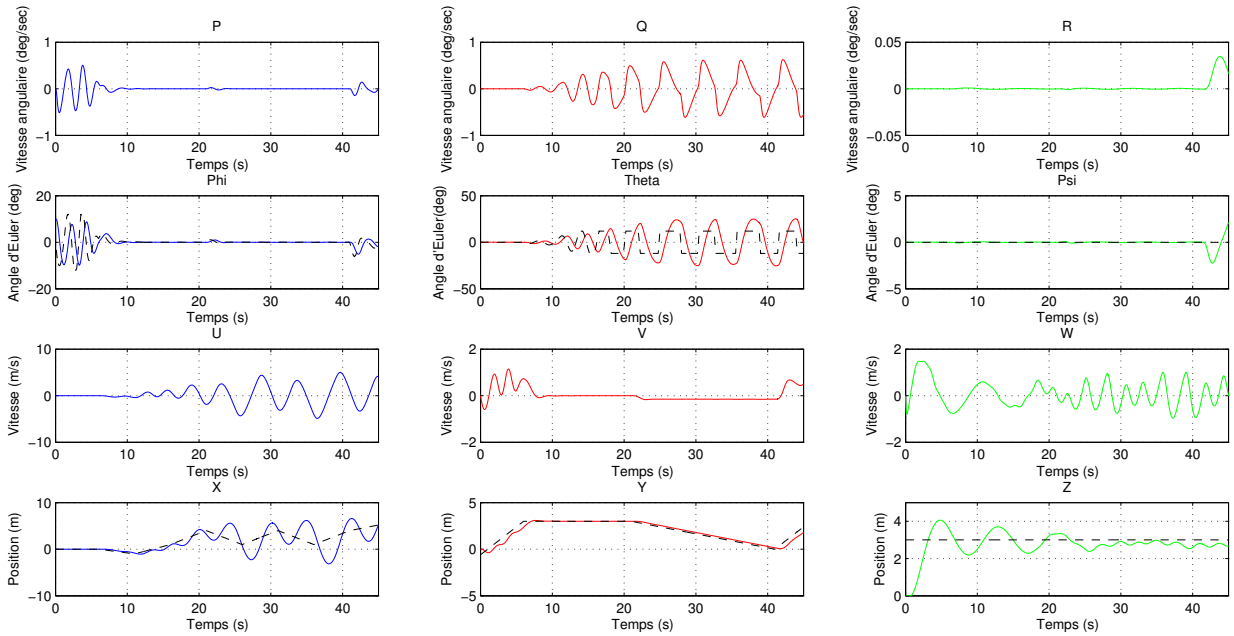


FIGURE IV.2 – Résultats du deuxième test

D'après la figure (IV.2)on remarque que :

Les graphes qui représentent la vitesse angulaire par rapport à l'axe (x) p, angle d'Euler ϕ se stabilise vers l'infini, ils suivent la référence désirée. Ce qui implique que les graphes de position X, Y ce rapprochent de la stabilité ils suivent la trajectoire désirée, d'où on conclue que le mouvement roulis se stabilise pour les valeurs des paramètres P=2.0, I=1.1, D=1.2

IV.4.3 Le troisième test

Le tableau suivant montre les valeurs des trois paramètres P, I, D des quatre mouvements qu'on choisit pour un troisiéme test.

Roulis	2.0	1.1	1.2
Tangage	2	1.1	1.2
Lacet	3	1	3
Altitude	1.8	1.5	3.5

TABLE IV.5 – Le troisième test

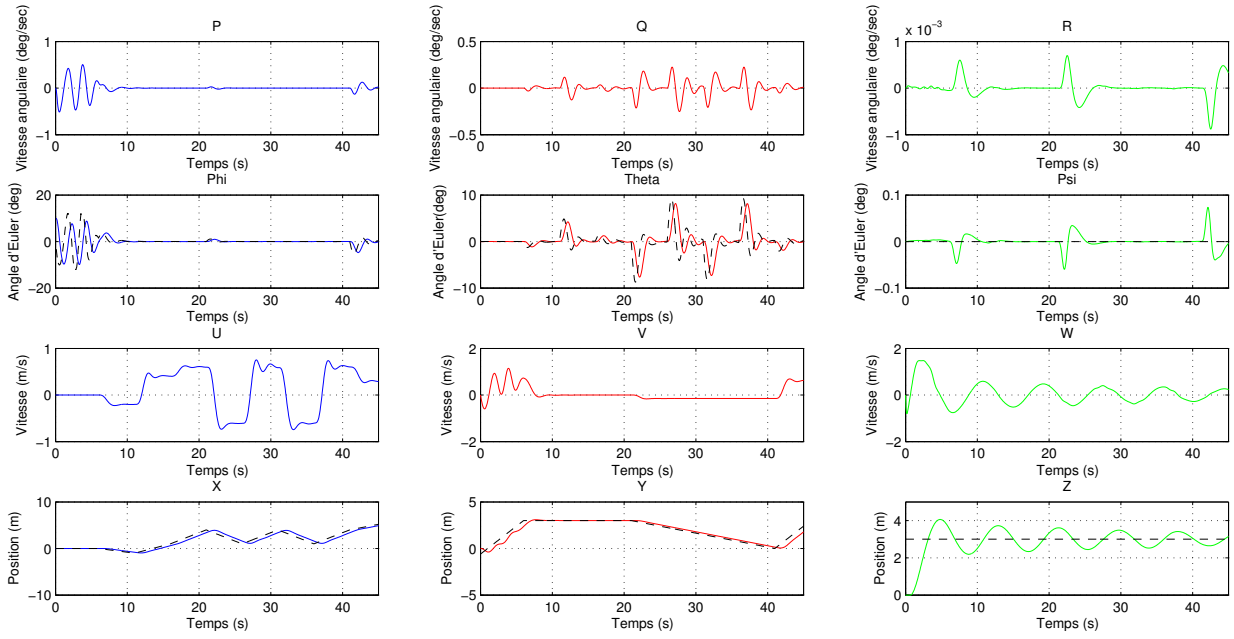


FIGURE IV.3 – Résultats du troisième test

D'après la figure (IV.3)on remarque que :

Les graphes qui représente la vitesse angulaire par rapport à l'axe (y) Q, angle d'Euler θ se stabilise vers l'infini ils suivent la référence désirée, ce qui implique que les graphes de position X, Y ce rapprochent de la stabilité, ils suivent la trajectoire désirée, d'où on conclue que le mouvement tangage se stabilise pour les valeurs des paramètres P=2.0, I=1.1, D=1.2

et puisque il exerce une influence sur le mouvement roulis vis-versa ce dernier devient plus stable.

IV.4.4 Le quatrième test

Mouvement	**P**	**I**	**D**
Roulis	2.0	1.1	1.2
Tangage	2.0	1.1	1.2
Lacet	4	0.5	3.5
Altitude	1.8	1.5	3.5

TABLE IV.6 – Le quatrième test

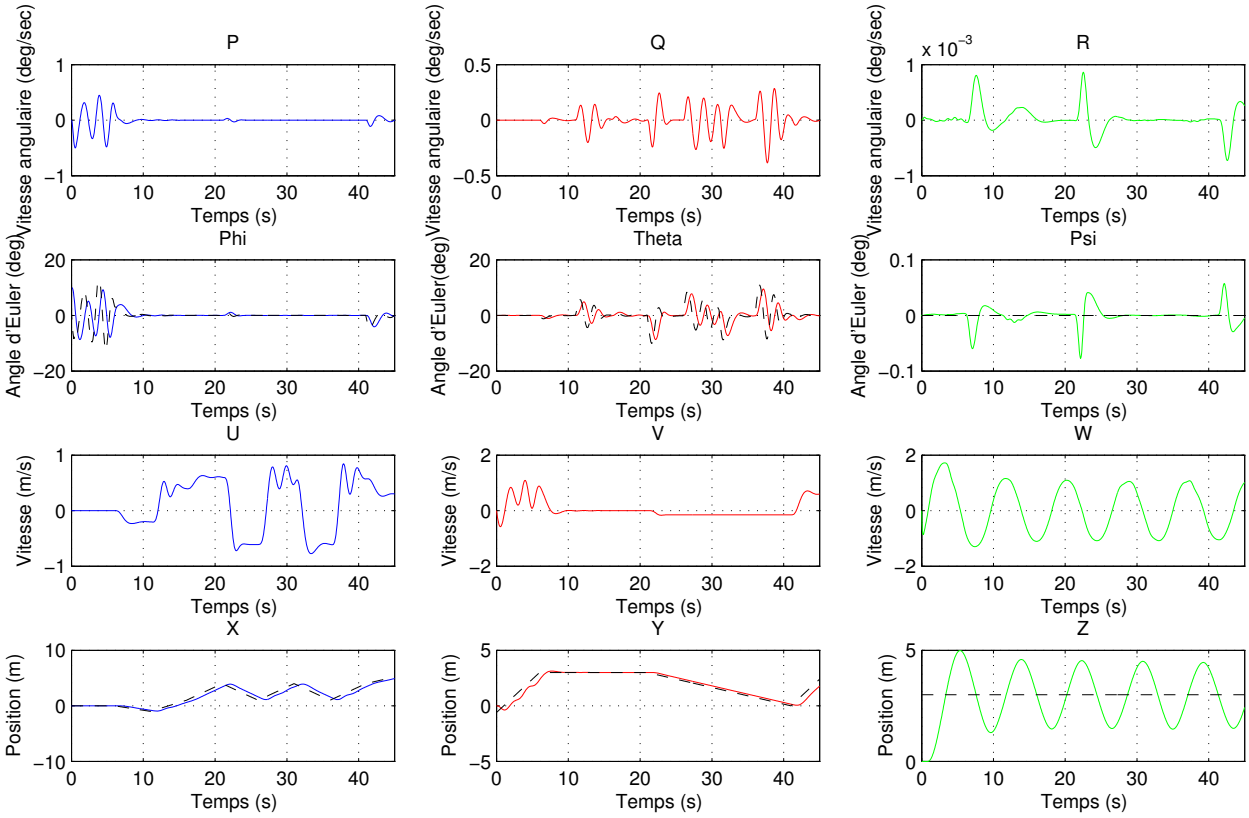


FIGURE IV.4 – Résultats du quatrième test

D'après la figure (IV.4), on remarque que les graphes qui représentent la vitesse angulaire par rapport à l'axe z et l'angle d'Euler ψ se stabilisent par rapport à la référence désirée. d'où la stabilité du mouvement du lacet pour les valeurs P=4, I=0.5 et D=3.5. La vitesse par rapport à l'axe z et la position Z est instable ce qui explique l'instabilité du mouvement vertical(Altitude).

IV.4.5 Le cinquième test

Mouvement	**P**	**I**	**D**
Roulis	2.0	1.1	1.2
Tangage	2	1.1	1.2
Lacet	4	0.5	3.5
Altitude	2	1.1	3.3

TABLE IV.7 – Le cinquième test

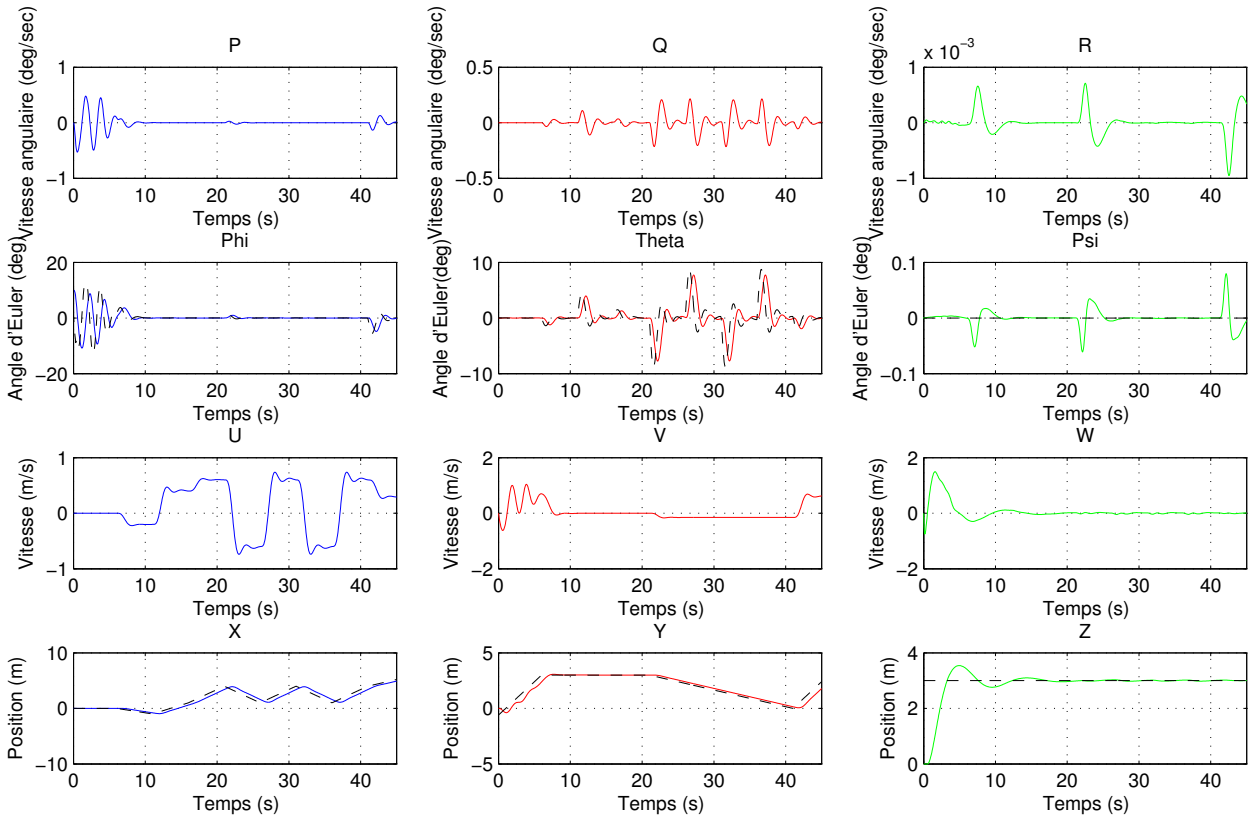


FIGURE IV.5 – Résultats du cinquième test

D'après la figure (IV.5) on remarque que le système est plus stable en utilisant les valeurs P, I et D qui sont adéquates pour les quatre mouvements (Altitude, roulis, lacet, tangage).

En conclusion, les valeurs du régulateur PID proposée dans les cinq testes sont convenables pour les quatre régulateurs de chaque mouvements d'où la stabilité du système en boucle fermée.

Conclusion

L'étude menée dans ce chapitre sur les mouvements d'un quad-copter nous a permis de construire un algorithme de contrôle plus performant. En effet, il augmente la rapidité du système ainsi que sa précision en changeant les valeurs des gains qui présente les paramètre PID. Chaque paramètre du régulateur influence directement sur la stabilisation du système, Ce qui nous a mené à varier les valeurs de ces paramètres jusqu'à l'obtention d'un système

stable avec des valeurs convenables.

Nous avons constaté que toute perturbation agissant directement sur la grandeur réglante est fortement atténuée avant qu'elle agisse sur la grandeur réglée.

Conclusion générale et perspectives

Cette note finale clôture ce mémoire et contient deux parties. La première partie concerne les conclusions qui peuvent être faites pour chacune des principales tâches de ce projet par rapport à l'objectif principal. La deuxième partie utilise les conclusions comme base pour des considérations de développements futures du projet.

Conclusion générale

L'objectif principal du travail est d'adopter un modèle mathématique de complexité minimal et d'avoir des lois de commande de vol qui permettent d'assurer la stabilité d'un quad-copter. Cette expérience était extrêmement enrichissante et formatrice puisque les connaissances nécessaires à la résolution de notre problématique dépassent largement le cadre de l'informatique, vu qu'elle inclue d'autres domaines tel que la mécanique et l'automatique qui nous ont permis d'enrichir nos connaissances.

Le but de la première partie est à la fois de présenter les drones plus précisément les quad-copters, mais aussi d'exposer leurs fonctionnements, leurs domaine d'applications et leurs différentes composantes.

La modélisation d'un quad-copter est une tâche complexe. Il s'agit d'une étape indispensable à la bonne compréhension des lois de la physique qui régissent de tels systèmes. Pour cela nous avons exploités un modèle dynamique, pour le but de décrire la dynamique d'un quad-copter. Le modèle utilisé est celui de Newton-Euler car il produit un modèle de façon récursive qui est en général plus rapide en calcul et en commande avec six degrés de liberté. Le processus de modélisation est constitué de plusieurs équations qui identifient toutes les forces et les couples ayant un impact sur le quad-copter. Ce dernier était implémenté sous Simulink

et ensuite assemblé dans un modèle complet.

Les algorithmes présentés dans ce mémoire offrent des performances de stabilisation et de suivi de trajectoires satisfaisante. La commande utilisée est la commande par régulation PID, cette technique est simple, efficace et de haute performance. Elle nous a permis de bien contrôler les différents comportements d'un quad-copter.

Le dernier chapitre, nous permis de produire de façon virtuelle la simulation d'un quad-copter, ce qui nous a mené à implémenter le modèle sous Simulinlk, ce dernier nous a permis de voir les différentes performances d'un quad-copter le cas de stabilisation et de déstabilisation. Les résultats ont était présenté à l'aide de l'outil Scope ou la commande plot.

Perspectives

Pour développer et améliorer ce travail, nous comptons réaliser les tâches suivantes :

- Améliorer la génération des perturbations dans la plate-forme Simulink pour avoir des résultats plus proches de la réalité.
- Implémenter les algorithmes de contrôle étudiés dans le chapitre III sous une carte Arduino en utilisant l'extension Ardu-pilot sous MATLAB.
- Améliorer la plate-forme en ajoutant l'interface 3D sous simulink pour pouvoir visualiser et étudier la trajectoire du quad-copter.
- Réaliser la configurations en "X".

Bibliographie

[1] K. adi, "Conférence," in *tutorial complet sur la monté des drones*, 2015.

[2] Wikipédia, "Drone." `https://fr.wikipedia.org/wiki/Drone#D.C3.A9finition`. Connecté ; dernière visite le 02 Février 2016.

[3] J. Baichtal, *Building Your Own Drones : A Beginners Guide to Drones, UAVs, and ROVs*. quepublishing.com, 2015.

[4] Onera, "Conférence," in *Mieux connaître les drones*, 2015.

[5] K. Saber, *Modélisation et commande d'un mini quad-copter*. PhD thesis, UNIVERSITE DU 20 AOUT 1955 SKIKDA, 2013.

[6] afcadillac, "Généralités." `http://drone.afcadillac.net/composition_dun_drone.html`. Connecté ; dernière visite le 02 Février 2016.

[7] T. Bresciani, *Modelling, Identification and Control of a Quadrotor Helicopter*. Lund University, October 2008.

[8] yvind Magnussen and K. Skjnhaug, *Modeling, Design and Experimental Study for a Quad-copter System Construction*. University of Agder, 2011.

[9] Astalaseven and E. etolyte, *Arduino pour bien commencer en électronique et en programmation*. Site de zéro, 2014.

[10] R. Taillet, L. Villain, and P. Febvre, *Dictionnaire de physique*. Bruxelles, De Boeck, 2013.

[11] K. Hicham, "Tolérance aux défauts via la méthode backstepping des systèmes non linéaires," Master's thesis, UNIVERSITE FERHAT ABBAS DE SETIF, 2012.

[12] H. Hoifodt, "Dynamic modeling and simulation of robot manipulators, the newton-euler formulation," Master's thesis, University of science and technology Norwegian, 2011.

[13] L. Gautier, B. Hamilton, J. Hazebrouck, and T. Tourrette, *Pilotage d'un quadri-rotor via un FPGA*. Tuteur ESIEE engineering , Journées pédagogiques, 2008.

[14] M. Spong, S. Hutchinson, and M. Vidyasagar, *Robot modeling and control*. Wiley New Jersey, 2006.

[15] K. Saber, "Modélisation et commande d'un mini-hélicopter drone," Master's thesis, Université de SKIKDA, 2013.

[16] D. Ross, E. Deguine, and M. Camus, *Asservissement par PID*. Université de lille 1, 2010.

[17] P. Bonnet, *Outils de simulation*. Université Lille 1, 2010.

A

Annexe

Liste des variables

Symbole	Unité	Description
e	$+$	erreur générique
e_z	m	erreur vertical
e_θ	rad	erreur de tangage
e_ϕ	rad	erreur de roulis
e_ψ	rad	erreur de lacet
p	$rads^{-1}$	vitesse angulaire autour de x_B dans B-frame
$\dot{p}$	$rads^{-2}$	accélération angulaire autour de x_B dans B-frame
q	$rads^{-1}$	vitesse angulaire autour de y_B dans B-frame
$\dot{q}$	$rads^{-2}$	accélération angulaire autour de y_B dans B-frame
r	$rads^{-1}$	vitesse angulaire autour de z_B dans B-frame
$\dot{r}$	$rads^{-2}$	accélération angulaire autour de z_B dans B-frame
u	ms^{-1}	vitesse linéaire autour de x_B dans B-frame
$\dot{u}$	ms^{-2}	accélération linéaire autour de x_B dans B-frame
v	ms^{-1}	vitesse linéaire autour de y_B dans B-frame
$\dot{v}$	ms^{-2}	accélération linéaire autour de y_B dans B-frame
w	ms^{-1}	vitesse linéaire autour de z_B dans B-frame
$\dot{w}$	ms^{-2}	accélération linéaire autour de z_B dans B-frame
y	$+$	processus générique de sortie
z^d	m	la hauteur désiré
z_{IR}	m	la hauteur mesuré par le module IR
z_{SONAR}	m	la hauteur mesuré par le module SONAR
$C_H(\zeta)$	$+$	matrice de Coriolis-centripètes dans H-frame
$E_H(\xi)$	$+$	matrice du mouvement dans H-frame
$G_H(\xi)$	$+$	vecteur gravitationnel dans H-frame

Symbole	Unité	Description
K_D	$+$	coefficient dérivé
K_{Dz}	s^{-1}	coefficient dérivé vertical
$K_{D\theta}$	s^{-1}	coefficient dérivé tangage
$K_{D\phi}$	s^{-1}	coefficient dérivé roulis
$K_{D\psi}$	s^{-1}	coefficient dérivé lacet
K_I	$+$	coefficient intégral
K_{Iz}	s^{-3}	coefficient intégral vertical
$K_{I\theta}$	s^{-3}	coefficient intégral tangage
$K_{I\phi}$	s^{-3}	coefficient intégral roulis
$K_{I\psi}$	s^{-3}	coefficient intégral lacet
K_P	$+$	coefficient proportionnel
K_{Pz}	s^{-3}	coefficient proportionnel vertical
$K_{P\theta}$	s^{-2}	coefficient proportionnel tangage
$K_{P\phi}$	s^{-2}	coefficient proportionnel roulis
$K_{P\psi}$	s^{-2}	coefficient proportionnel lacet
$O_H(\zeta)$	$+$	matrice gyroscopique des hélices dans H-frame
U_1	N	poussée vertical dans B-frame
U_2	Nm	un torque de roulis dans B-frame
U_3	Nm	un torque de tangage dans B-frame
U_4	Nm	un torque de lacet dans B-frame
U	$+$	vecteur basique de mouvement
X	m	position linéaire autour de x_E dans E-frame
$\dot{X}$	ms^{-1}	vitesse linéaire autour de x_E dans E-frame
$\ddot{X}$	ms^{-2}	accélération linéaire autour de x_E dans E-frame
Y	m	position linéaire autour de y_E dans E-frame
$\dot{Y}$	ms^{-1}	vitesse linéaire autour de y_E dans E-frame
$\ddot{Y}$	ms^{-2}	accélération linéaire autour de y_E dans E-frame

Symbole	Unité	Description
Z	m	position linéaire autour de Z_E dans E-frame
$\dot{Z}$	ms^{-1}	vitesse linéaire autour de Z_E dans E-frame
$\ddot{Z}$	ms^{-2}	accélération linéaire autour de Z_E dans E-frame
ζ	$+$	vecteur de vitesse généralisé dans H-frame
$\dot{\zeta}$	$+$	vecteur d'accélération généralisé dans H-frame
Ω_1	$rads^{-1}$	vitesse d'hélice d'avant
Ω_2	$rads^{-1}$	vitesse d'hélice droit
Ω_3	$rads^{-1}$	vitesse d'hélice arrière
Ω_4	$rads^{-1}$	vitesse d'hélice gauche
Ω	$rads^{-1}$	vecteur vitesse des hélices
Ω	$rads^{-1}$	vitesse des hélices en général
$\dot{\Omega}$	$rads^{-2}$	vecteur d'accélération des hélices
ϕ	rad	position angulaire autour de x_2 E-frame(roulis)
$\dot{\phi}$	$rads^{-1}$	vitesse angulaire autour de x_2 E-frame(roulis)
$\ddot{\phi}$	$rads^{-2}$	accélération angulaire autour de x_2 E-frame(roulis)
ϕ^d	rad	angle de roulis désiré
ψ	rad	position angulaire autour de z_2 E-frame(lacet)
$\dot{\psi}$	$rads^{-1}$	vitesse angulaire autour de z_2 E-frame(lacet)
$\ddot{\psi}$	$rads^{-2}$	accélération angulaire autour de z_2 E-frame (lacet)
$\dot{\psi}^d$	$rads^{-1}$	vitesse angulaire de lacet désiré
ψ^d	rad	angle de lacet désiré
Γ^E	m	vecteur de position linéaire E-frame
$\dot{\Gamma^E}$	ms^{-1}	vecteur de vitesse linéaire E-frame
$\ddot{\Gamma^E}$	ms^{-2}	vecteur d'accélération linéaire E-frame
υ	V	vecteur de tension d'entrée

B

Annexe

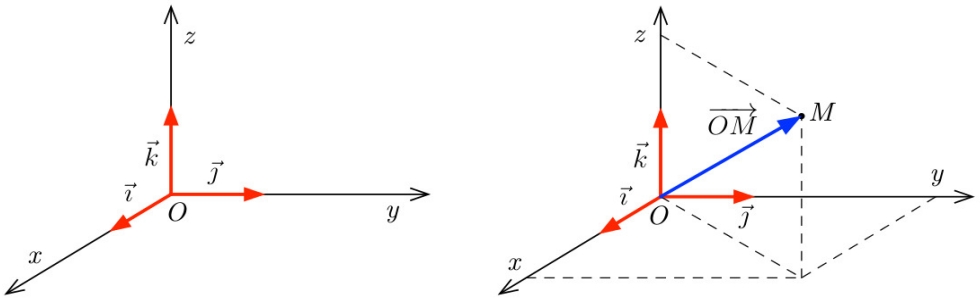
Cinématique et dynamique

Grandeurs cinématiques

En classe de 2e nous avons introduit les grandeurs cinématiques utilisées pour décrire le mou- vement d'un point matériel : l'abscisse curviligne, les vecteurs position, vitesse et accélération. Les vecteurs sont exprimés dans la base d'un repère, le plus souvent orthonormée. Le choix de la base est arbitraire mais, en pratique, est guidé par la trajectoire et les forces qui agissent sur le mobile. Nous allons utiliser la base cartésienne et la base de Frenet.

Base cartésienne

À un référentiel galiléen (par exemple le référentiel terrestre) nous pouvons attacher un repère cartésien $(O, \overrightarrow{i}, \overrightarrow{j}, \overrightarrow{k})$ dont les vecteurs unitaires de base sont fixes par rapport au référentiel figure (A.1).



(a) base cartésienne (b) vecteur position

FIGURE A.1 – Repère orthonormé à 3 dimensions

position d'un mobile

Dans la base cartésienne, le vecteur position du point mobile M s'exprime figure (A.1b) :

$$\overrightarrow{OM} = x\,\overrightarrow{i} + y\,\overrightarrow{j} + z\,\overrightarrow{k} \tag{B.1}$$

Une autre façon de repérer la position d'un mobile M sur sa trajectoire est d'utiliser l'abscisse curviligne. Pour cela, on choisit arbitrairement figure (A.2) :

- Un point A sur la trajectoire (l'origine).

- Un sens positif.

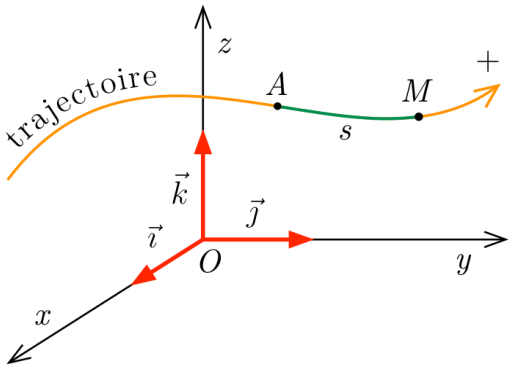


FIGURE A.2 – Abscisse curviligne

L'abscisse curviligne s est la mesure algébrique de l'arc $\widehat{AM}$. Il est à noter que pour pouvoir utiliser l'abscisse curviligne, il faut connaître la trajectoire du mobile.

Vecteur vitesse

Le vecteur vitesse v du mobile M à l'instant t nous renseigne sur la rapidité du changement du vecteur position à cet instant. Il est défini par figure (A.3) :

$$\vec{v} = \lim_{t \to t'} \frac{\overrightarrow{MM'}}{t' - t} = \frac{d\overrightarrow{OM}}{dt} \tag{B.2}$$

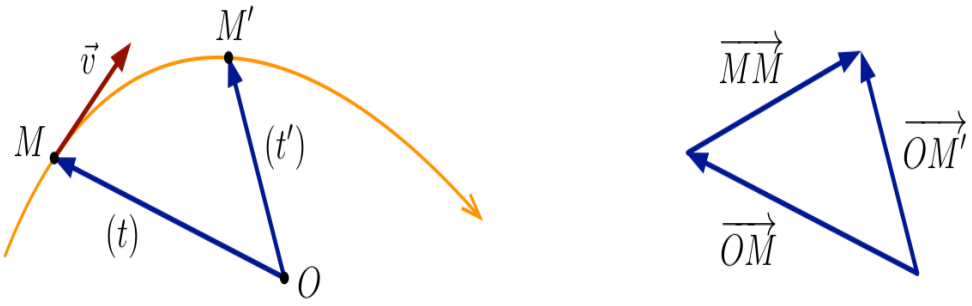


FIGURE A.3 – Vecteur vitesse

En effet :

$$\overrightarrow{MM'} = \overrightarrow{MO} + \overrightarrow{OM'} = \overrightarrow{OM'} - \overrightarrow{OM} = \delta\overrightarrow{OM} \tag{B.3}$$

Vecteur accélération

Le vecteur accélération a à l'instant t indique la rapidité de la variation du vecteur vitesse. Il est défini par figure (A.4) :

$$\overrightarrow{a} = \lim_{t' \to t} \frac{v' - v}{t' - t} = \frac{d\overrightarrow{v}}{dt} \tag{B.4}$$

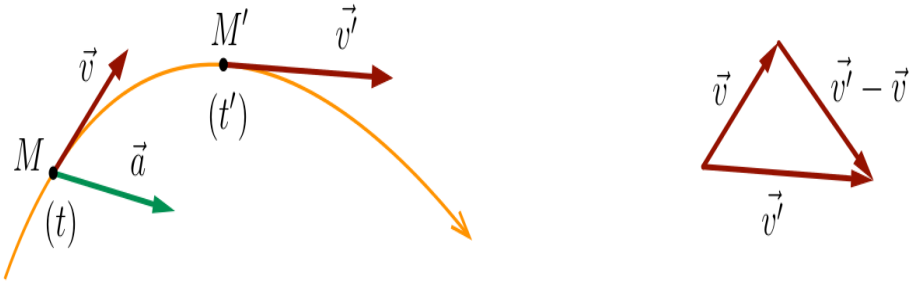


FIGURE A.4 – Vecteur accélération

Étude dynamique

Le système étudié est un projectile ponctuel de masse m. L'étude de son mouvement se fera dans le référentiel terrestre supposé galiléen.

La seule force appliquée est le poids $P = m.g$ du projectile. Nous négligeons ici le frottement de l'air et la poussée d'Archimède.

Dans le référentiel terrestre, la relation fondamentale de la dynamique s'applique :

$$\sum_i \overrightarrow{F}_i = \overrightarrow{P} = \overrightarrow{m}.a$$

L'accélération du projectile est donnée par :

$$\overrightarrow{a} = \frac{\overrightarrow{P}}{m} = \frac{m\overrightarrow{g}}{m} = \overrightarrow{g}$$

Le vecteur accélération est indépendant de la masse du projectile et égal au vecteur champ de pesanteur. C'est un vecteur constant [10].

La force d'interaction gravitationnelle

Selon la loi de gravitation de Newton, deux corps A et B quasi ponctuels ou à symétrie sphérique, de masses M et m et dont les centres $\overrightarrow{OA}$ et $\overrightarrow{OB}$ sont distants de r, exercent l'un

sur l'autre des forces attractives $\overrightarrow{F}_{A/B}$ et $\overrightarrow{F}_{B/A}$ de même direction $O_A\ O_B$, de même intensité mais de sens opposés figure (A.5) :

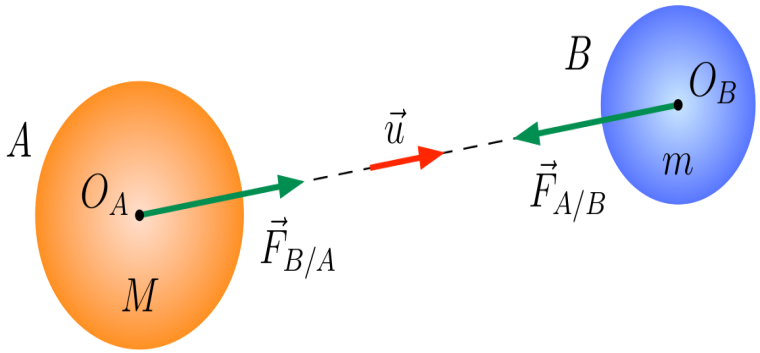


FIGURE A.5 – Forces d'interaction gravitationnelle

La constante K est appelée constante de gravitation. Sa valeur dans le Système international d'unités est :

$$K = 6,67.10^{-11} N.kg^{-2}.m^2$$

Une expression vectorielle de la force gravitationnelle s'obtient en définissant un vecteur unitaire u, directeur de la droite O_AO_B et orienté de O_A vers O_B figure (A.5) [10] :

$$\overrightarrow{F}_{A/B} = -\overrightarrow{F}_{B/A} = -K\frac{mM}{r^2}\overrightarrow{u} = -F\overrightarrow{u}$$

Définition du champ de gravitation

Lorsqu'une masse ponctuelle m subit les forces d'attraction d'un ensemble de masses, chaque terme de la somme vectorielle qui représente la résultante $\overrightarrow{F}$ est proportionnelle à m ; il en suit que la résultante est également proportionnelle à m. La grandeur vectorielle $\overrightarrow{F}/m$ est donc indépendante de m et appelée vecteur champ de gravitation [10].

Vitesse angulaire

La vitesse angulaire Ω, aussi appelée fréquence angulaire ou pulsation, est une mesure de la vitesse de rotation et une caractéristique d'une oscillation sinusoïdale.

Elle s'exprime dans le système international en radians par seconde ($rad.s^{-1}$). On peut aussi utiliser des degrés/seconde et des tours/seconde. Comme les angles sont des grandeurs

sans dimension, on pourrait la communiquer simplement en s^{-1}, mais cette pratique est à éviter, à moins que l'unité d'angle soit parfaitement claire. Dans les domaines de la mécanique industrielle et de la vie courante, on l'exprime souvent en tours par minute (tr/min). [11]

Vitesses linéaires

Les vitesses linéaires v_x^b, v_b^y, v_z^b dans le repère fixe en fonction des vitesses linéaires v_x^m, v_y^m, v_z^m dans le repère mobile sont données par [11] :

$$v = \begin{bmatrix} v_x^b \\ v_b^y \\ v_z^b \end{bmatrix} = R \times \begin{bmatrix} v_x^m \\ v_y^m \\ v_z^m \end{bmatrix}$$

Dynamique

La dynamique est une branche de mécanique qui étudie les effets des forces et couples sur le mouvement d'un corps ou d'un système des corps. Il y a plusieurs techniques qui peuvent être employées pour dériver les équations d'un corps rigide avec 6 DDF.La formulation de Newton-Euler a été adoptée dans ce travail.

Les forces

Le poids du quad-copter

Il est donné par $P = mg$, où : m est la masse totale et g la gravité.

Les forces de poussée

Qui sont des forces provoquées par la rotation des moteurs, elles sont perpendiculaires sur le plan des hélices. Ces forces sont proportionnelles au carrée de la vitesse de rotation des moteurs :

$$F_i \ = \ b\, \omega_i^2 \tag{B.5}$$

Les forces de traînée

La force de traînée est le couplage entre une force de pression et la force de frottement visqueux, dans ce cas on a deux forces de traînée agissant sur le système qu'elles sont :

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

$$T_h \;=\; d\,\omega^2 \tag{B.6}$$

avec d est le coefficient de drag il dépend de la fabrication de l'hélice.

- a traînée selon les axes (x, y, z) : elle est due au mouvement du corps du quad-copter.

$$F_t \;=\; K_{ft}\,v \tag{B.7}$$

avec : K_{ft} le coefficient de traînée de translation et v la vitesse linéaire.

Les moments

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

Moments dus aux forces de poussée

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

$$M_x \;=\; l\,(F_4 \;-\; F_2) \;=\; l\;b\,(\omega_4^2 \;-\; \omega_2^2) \tag{B.8}$$

avec l est la longueur du bras entre le rotor et le centre de gravité du quad-copter. La rotation autour de l'axe y : elle est due au moment crié par la différence entre les forces de portance des rotors 1 et 3, ce moment est donné par la relation suivante :

$$M_y \;=\; l\,(F_3 \;-\; F_1) \;=\; l\;b\,(\omega_3^2 \;-\; \omega_1^2) \tag{B.9}$$

Moments dus aux forces de traînée

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

$$M_z \;=\; dl\,(\omega_1^2 \;+\; \omega_2^2 \;+\; \omega_3^2 \;+\; \omega_4^2) \tag{B.10}$$

Moment résultant des frottements aérodynamiques, il est donné par :

$$M_a \;=\; K_{fa}\,\Omega^2 \tag{B.11}$$

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

C

Annexe

Code source

quadPlots.m

```
function [] = quadPlots(yout,tout)
close all; clc;
%fatima c quoi caaaaaa
[~,column] = size(yout);
if column==26
    PC = true;
else PC = false;
end
A = yout;T = tout;t = T;
% lunite rad/s sera converti apres le dessin
P = A(:,1); Q = A(:,2);R = A(:,3);
% en radians  sera converti apres le dessin
Phi   = A(:,4); Theta = A(:,5);Psi= A(:,6);
Phi_cmd = A(:,21);Theta_cmd = A(:,22);Psi_cmd = A(:,23);
U = A(:,7); V = A(:,8);W = A(:,9);
X = A(:,10);Y = A(:,11);Z = A(:,12);
Z_cmd = A(:,24);
w1 = A(:,13);w2 = A(:,14);w3 = A(:,15);w4 = A(:,16);
mc1 = A(:,17);mc2 = A(:,18);mc3 = A(:,19);mc4 = A(:,20);
if (PC==true)
X_cmd = A(:,25);
Y_cmd = A(:,26);
end
% Dessin des graphs _____________________________

figure
subplot(4,3,1)
plot(T,P,'b')
```

```
29 xlabel('Temps (s)')
30 ylabel('Vitesse angulaire (deg/sec)')
31 xlim([min(t) max(t)])
32 title('P')
33 grid on
34
35 subplot(4,3,2)
36 plot(T,Q,'r')
37 xlabel('Temps (s)')
38 ylabel('Vitesse angulaire (deg/sec)')
39 xlim([min(t) max(t)])
40 title('Q')
41 grid on
42
43 subplot(4,3,3)
44 plot(T,R,'g')
45 xlabel('Temps (s)')
46 ylabel('Vitesse angulaire (deg/sec)')
47 xlim([min(t) max(t)])
48 title('R')
49 grid on
50
51 subplot(4,3,4)
52 plot(T,Phi*180/pi,'b')
53 hold on
54 plot(T,Phi_cmd*180/pi,'k--')
55 hold off
56 xlabel('Temps (s)')
57 ylabel('Angle d''Euler (deg)')
58 xlim([min(t) max(t)])
59 title('Phi')
```

```
60 grid on
61
62 subplot(4,3,5)
63 plot(T,Theta*180/pi,'r')
64 hold on
65 plot(T,Theta_cmd*180/pi,'k--')
66 hold off
67 xlabel('Temps (s)')
68 ylabel('Angle d''Euler(deg)')
69 xlim([min(t) max(t)])
70 title('Theta')
71 grid on
72
73 subplot(4,3,6)
74 plot(T,Psi*180/pi,'g')
75 hold on
76 plot(T,Psi_cmd*180/pi,'k--')
77 hold off
78 xlabel('Temps (s)')
79 ylabel('Angle d''Euler (deg)')
80 xlim([min(t) max(t)])
81 title('Psi')
82 grid on
83
84 subplot(4,3,7)
85 plot(T,U,'b')
86 xlabel('Temps (s)')
87 ylabel('Vitesse (m/s)')
88 xlim([min(t) max(t)])
89 title('U')
90 grid on
```

```
91
92  subplot(4,3,8)
93  plot(T,V,'r')
94  xlabel('Temps (s)')
95  ylabel('Vitesse (m/s)')
96  xlim([min(t) max(t)])
97  title('V')
98  grid on
99
100 subplot(4,3,9)
101 plot(T,W,'g')
102 xlabel('Temps (s)')
103 ylabel('Vitesse (m/s)')
104 xlim([min(t) max(t)])
105 title('W')
106 grid on
107
108 subplot(4,3,10)
109 plot(T,X,'b')
110 if (PC==true)
111 hold on
112 plot(T,X_cmd,'k--')
113 hold off
114 end
115 xlabel('Temps (s)')
116 ylabel('Position (m)')
117 xlim([min(t) max(t)])
118 title('X')
119 grid on
120
121 subplot(4,3,11)
```

```
122 plot(T,Y,'r')
123 if (PC==true)
124 hold on
125 plot(T,Y_cmd,'k--')
126 hold off
127 end
128 xlabel('Temps (s)')
129 ylabel('Position (m)')
130 xlim([min(t) max(t)])
131 title('Y')
132 grid on
133
134 subplot(4,3,12)
135 plot(T,Z,'g')
136 hold on
137 plot(T,Z_cmd,'k--')
138 hold off
139 xlabel('Temps (s)')
140 ylabel('Position (m)')
141 xlim([min(t) max(t)])
142 title('Z')
143 grid on
```

SFunction.m

```
1 function quadcopterDynamicsSFunction(block)
2 setup(block);
3 function setup(block)
4   % nombre de port.
5   %------
6   block.NumInputPorts  = 5;
7   %------
```

```
8   block.NumOutputPorts = 12;
9   for i = 1:4; % Les entree du moteurs
10  block.InputPort(i).Dimensions        = 1;
11  block.InputPort(i).DirectFeedthrough = false;
12  block.InputPort(i).SamplingMode      = 'Sample';
13  end
14  %-----
15  % Le bruit
16  block.InputPort(5).Dimensions        = 6; % couple x,y,z; forces
        x,y,z.
17  block.InputPort(5).DirectFeedthrough = false;
18  block.InputPort(5).SamplingMode      = 'Sample';
19  %-----
20  for i = 1:12;
21  block.OutputPort(i).Dimensions       = 1;
22  block.OutputPort(i).SamplingMode     = 'Sample';
23  end
24  % enregistrement des parametres
25  block.NumDialogPrms     = 2;
26  block.NumContStates = 12;
27  block.SampleTimes = [0  0];
28  block.SetAccelRunOnTLC(false);
29  block.SimStateCompliance = 'DefaultSimState';
30  block.RegBlockMethod('CheckParameters', @CheckPrms);
31  block.RegBlockMethod('InitializeConditions',
        @InitializeConditions);
32  block.RegBlockMethod('Outputs', @Outputs);
33  block.RegBlockMethod('Derivatives', @Derivatives);
34
35
36  function CheckPrms(block)
```

```
37        quad    = block.DialogPrm(1).Data;
38        IC      = block.DialogPrm(2).Data;
39
40  function InitializeConditions(block)
41  % Initialiser les 12 etats
42  IC = block.DialogPrm(2).Data;
43  % IC.P, IC.Q, IC.R en deg/s convertir rad/s
44  P = IC.P*pi/180; Q = IC.Q*pi/180; R = IC.R*pi/180;
45  % IC.Phi, IC.The, IC.Psi en deg convertir rads
46  Phi = IC.Phi*pi/180; The = IC.The*pi/180; Psi = IC.Psi*pi/180;
47  U = IC.U; V = IC.V; W = IC.W;
48  X = IC.X; Y = IC.Y; Z = IC.Z;
49  init = [P,Q,R,Phi,The,Psi,U,V,W,X,Y,Z];
50  for i=1:12
51  block.OutputPort(i).Data = init(i);
52  block.ContStates.Data(i) = init(i);
53  end
54
55  function Outputs(block)
56  for i = 1:12;
57    block.OutputPort(i).Data = block.ContStates.Data(i);
58  end
59
60  function Derivatives(block)
61  quad = block.DialogPrm(1).Data;
62  % P Q R rad/sec
63  P = block.ContStates.Data(1);
64  Q = block.ContStates.Data(2);
65  R = block.ContStates.Data(3);
66  % Phi radians
67  Phi = block.ContStates.Data(4);
```

```
68 The = block.ContStates.Data(5);
69 Psi = block.ContStates.Data(6);
70 % U V W m/s
71 U = block.ContStates.Data(7);
72 V = block.ContStates.Data(8);
73 W = block.ContStates.Data(9);
74 % X Y Z m
75 X = block.ContStates.Data(10);
76 Y = block.ContStates.Data(11);
77 Z = block.ContStates.Data(12);
78 % w  radians/s
79 w1 = block.InputPort(1).Data;
80 w2 = block.InputPort(2).Data;
81 w3 = block.InputPort(3).Data;
82 w4 = block.InputPort(4).Data;
83 w  = [w1; w2; w3; w4];
84 Dist_tau = block.InputPort(5).Data(1:3);
85 Dist_F   = block.InputPort(5).Data(4:6);
86 tau_motorGyro = [Q*quad.Jm*2*pi/60*(-w1-w3+w2+w4);
87      P*quad.Jm*2*pi/60*(w1+w3-w2-w4);  0];
88 Mb = (quad.dctcq*(w.^2))+ tau_motorGyro + (Dist_tau);
89 Fb = [0;  0;  sum(quad.ct*(w.^2))];
90 omb_bi = [P; Q; R];
91 OMb_bi = [ 0,-R, Q;
92              R, 0,-P;
93             -Q, P, 0];
94 b_omdotb_bi = quad.Jbinv*(Mb-OMb_bi*quad.Jb*omb_bi);
95 H_Phi = [1,tan(The)*sin(Phi), tan(The)*cos(Phi);
96          0,          cos(Phi),          -sin(Phi);
97          0,sin(Phi)/cos(The),cos(Phi)/cos(The)];
98 Phidot = H_Phi*omb_bi;
```

```
99  Rib = [cos(Psi)*cos(The)  cos(Psi)*sin(The)*sin(Phi)-sin(Psi)*cos(
        Phi)  cos(Psi)*sin(The)*cos(Phi)+sin(Psi)*sin(Phi);
100         sin(Psi)*cos(The)  sin(Psi)*sin(The)*sin(Phi)+cos(Psi)*cos(
              Phi)  sin(Psi)*sin(The)*cos(Phi)-cos(Psi)*sin(Phi);
101         -sin(The)           cos(The)*sin(Phi)
                                          cos(The)*cos(Phi)];
102 Rbi = Rib';
103 ge = [0;  0; -quad.g];
104 gb = Rbi*ge;
105 Dist_Fb = Rbi*Dist_F;
106 vb = [U;V;W];
107 b_dv = (1/quad.mass)*Fb+gb+Dist_Fb-OMb_bi*vb;
108 i_dp = Rib*vb;
109 dP = b_omdotb_bi(1);
110 dQ = b_omdotb_bi(2);
111 dR = b_omdotb_bi(3);
112 dPhi = Phidot(1);
113 dTheta = Phidot(2);
114 dPsi = Phidot(3);
115 dU = b_dv(1);
116 dV = b_dv(2);
117 dW = b_dv(3);
118 dX = i_dp(1);
119 dY = i_dp(2);
120 dZ = i_dp(3);
121 if ((Z<=0) && (dZ<=0))
122     dZ = 0;
123     block.ContStates.Data(12) = 0;
124 end
125 f = [dP dQ dR dPhi dTheta dPsi dU dV dW dX dY dZ].';
126 block.Derivatives.Data = f;
```

Résumé

De nos jours les systèmes embarqués joue un rôle très important dans développement de l'industrie. C'est pour cela que nous nous sommes intéressés à réaliser une plate-forme virtuelle qui permet d'analyser les comportements et les différents mouvements d'un quad-copter en exploitant un modèle mathématique.

Cette plate-forme est composer de plusieurs algorithmes de contrôle implémenté sous l'environnement simulink, qui permet de création des modèles et conditions initiales, ces dernier, peuvent être simuler, tester et visualiser les résultats sous forme des graphe.

Mots clés : Simulink, MATLAB, Quad-copter, Régulation, PID.

Abstract

Nowadays embedded systems play a very important role in development of the industry. this is why we are interested to make a virtual platform which allows to analyze the behavior and the different movements of a quad-copter by exploiting an existing mathematical model.

This platform consists of several control algorithms implemented in the Simulink environment, which allows the creation of models and initial conditions, these latter can be simulate, test and view the results in the form of graph.

Keywords : Simulink, MATLAB, Quad-copter, Regulation, PID.