

REPUBLIQUE ALGERIENNE DEMOCRATIQUE ET POPULAIRE

الجمهورية الجزائرية الديمقراطية الشعبية

MINISTRY OF HIGHER EDUCATION
AND SCIENTIFIC RESEARCH

HIGHER SCHOOL IN APPLIED SCIENCES
--T L E M C E N--



المدرسة العليا في العلوم التطبيقية
École Supérieure en
Sciences Appliquées

وزارة التعليم العالي والبحث العلمي

المدرسة العليا في العلوم التطبيقية
-تلمسان-

Mémoire de fin d'étude

Pour l'obtention du diplôme de Master

Filière : Génie électrique

Spécialité : Automatique

Présenté par : BOUCHOUK Meroua

Thème

**Estimation de l'attitude d'un UAV à base
d'observateur d'état**

Soutenu publiquement, le 29 / 09 / 2020, devant le jury composé de :

M. MERAD Lotfi	Professeur	ESSA. Tlemcen	Président
M.TAHOUR Ahmed	Professeur	ESSA. Tlemcen	Directeur de mémoire
M.TADJINE Mohamed	Professeur	ENP. Alger	Co- Directeur de mémoire
M.MOKHTARI Mohamed Rida	MCB	ESSA. Tlemcen	Examineur 1
M. ABDI Sidi Mohamed	MCB	ESSA. Tlemcen	Examineur 2

Année universitaire : 2019/2020

DÉDICACE

Je dédie ce travail

*A mes chers parents qui m'ont toujours apporté leur amour et leur soutien
pour affronter les difficultés de la vie.*

A mes chers frères, et à ma chère sœur.

A tous les membres de ma famille

A tous les automaticiens et automaticienne de promo 2015/2020.

A tous mes amis

*A tous ceux qui m'ont connu et aidé de près ou de loin pour réaliser ce
modeste travail.*

*A tous ceux qui liront ce travail. N'hésitez pas à m'écrire si vous avez besoin
d'aide.*

REMERCIEMENTS

Celui qui ne remercie pas pour la petite chose, ne remercie pas pour la grande chose et celui qui ne remercie pas les gens, ne remercie pas Allah. Prophète Mohammed que la prière d'Allah et son salut soient sur lui.

Je tiens à remercier très vivement mes encadreur Monsieur TADJINE Mohamed Professeur à l'École Nationale Polytechnique, Alger d'avoir proposé le sujet sur lequel j'ai travaillé, et qui a assuré la direction et l'encadrement du travail présenté dans ce mémoire. C'est avec beaucoup de chance que j'ai eu l'honneur et le plaisir de travailler sous sa direction. Monsieur TAHOUR Ahmed Professeur à Ecole Supérieure en Science Appliquée, Tlemcen qui m'a fait profiter de ses larges connaissances et ses précieux conseils. J'ai beaucoup apprécié sa méthode de travail, son enthousiasme et sa qualité de correction de documents scientifiques.

Je tiens mes plus vifs et profonds remerciements sont adressés à Monsieur ALLAM Ahmed de École Nationale Polytechnique, Bordj El-Bahri, Alger pour ses efforts, ses encouragements et son soutien permanent et qui a été toujours à l'écoute et prêt pour donner des conseils et pour sa pédagogie à diriger mon travail.

Mes remerciements s'adressent aussi à Monsieur Mehdi Zareb, Docteur à l'université Mustapha Stambouli Mascara et Monsieur BOUZID Yasser de École Nationale Polytechnique, Bordj El-Bahri, Alger. Pour l'intérêt particulier qu'ils ont porté à mon travail, leurs conseils et leur temps perdu pour répondre a mes questions.

Mes remerciements vont également a Monsieur KECHIDA Ahmed, Maitre de recherche au Centre de technologies industrielles CRTI, ex-CSC de m'avoir accueillie au centre. Ainsi, le temps, l'attention, l'intérêt qui il avait bien voulu me témoigner n'ont pas été perdus. Ils m'ont donné envie de persévérer dans ce domaine de robotique et d'aviation.

Je remercie très chaleureusement Monsieur GOUBAA Abderrahim Ingénieur diplômée de l'institut des sciences appliqués et des technologies (INSAT), Tunisie pour tous ces efforts et son temps.

Mes remerciements s'adressent également à Monsieur le président du jury Monsieur MERAD Lotfi Professeur et chef de département du second cycle à l'ESSA de Tlemcen et à mes examinateurs Monsieur MOKHTARI Mohamed Rida Maître de conférence B et Monsieur ABDI Sidi Mohamed Maître de conférence B qui m'ont fait l'honneur d'évaluer ce travail.

ملخص

يندرج العمل في هذه الأطروحة في اطار التحكم، ويدرس مشاكل تتبع المسار والتحكم في التنسيق لأنظمة الطائرات بدون طيار الفردية و طرق التتبع لكل عناصرها. تتم معالجة مشاكل التحكم هذه من أجل نوع الدرون رباعي المحركات. لذا كخطوة أولى في الأطروحة الحالية، تم تصميم نموذج رباعي الدوران بالإضافة إلى التحكم فيه باستعمال تحكم خطي، ثم نقوم بدراسة مقارنة لثلاث انواع من المرقبات(فلتر كلمان، فلتر كلمان ممدد، لينبيرجي) لغرض تقدير عناصر النموذج باستخدام دمج لمستشعرات مختلفة. ينتهي العمل بمحاكاة ثلاثية الأبعاد تحت بيئة تحكم الروبوتات. الكلمات الرئيسية: الطائرات بدون طيار، طائرات رباعية المحرك، فلتر كلمان، فلتر كلمان ممدد، لينبيرجي، متلاب، تحكم خطي .

ABSTRACT

The work presented in this thesis concerns unmanned aerial vehicle (UAV) systems, and studies trajectory tracking, control and observability problems. These control problems are addressed for quadrotor type. Thus, as a first step, the quadrotor is modeled as well as controlled. In a second step, we will do a study of the observers we had chosen (Luenberger observer, Kalman filter, extended Kalman filter (EKF)) as part of an attitude state estimate for a quadrotor using a data fusion of different sensors of an IMU. The work ends with a 3D simulation under the ROS environment.

Keywords: UAV, KF, EKF, quadrirotor, Luenberger observer, state estimate, data fusion, IMU, PID, ROS, MATLAB, Gazebo.

RÉSUMÉ

Les travaux présentés dans cette thèse concernent les systèmes de véhicules aériens sans pilote (UAV) et étudient les problèmes de suivi de trajectoire, de contrôle et d'observabilité. Ces problèmes de contrôle sont abordés pour le type quadrotor.

Ainsi, dans un premier temps, le quadrirotor est modélisé et contrôlé. Dans un deuxième temps, nous ferons une étude des observateurs que nous avons choisis (observateur Luenberger, filtre de Kalman, filtre de Kalman étendu (EKF)) dans le cadre d'une estimation de l'état d'attitude d'un quadrotor à l'aide d'une fusion de données de différents capteurs d'une IMU. Le travail se termine par une simulation 3D sous l'environnement ROS.

Mots-clés : UAV, quadrirotor, KF, EKF, l'observateur de Luenberger, la fusion des données, PID, ROS, MATLAB, Gazebo.

TABLE DES MATIÈRES

REMERCIEMENTS	3
ARABIC ABSTRACT	5
ABSTRACT	6
RÉSUMÉ	7
TABLE DES MATIÈRES	8
LISTE DES TABLEAUX	11
LISTE DES FIGURES	12
LISTE DES SIGLES ET ABRÉVIATIONS	14
LISTE DES ANNEXES	15
CHAPITRE 1 REVUE DE LITTÉRATURE	3
1.1 Intoduction	3
1.2 L'EVOLUTION DES DRONES AU FILS DU TEMPS	3
1.2.1 les premiers drones	3
1.2.2 L'entre-deux guerres	3
1.2.3 LA GUERRE FROIDE	5
1.3 CLASSIFICATION DES DRONES	7
1.4 Technologie des capteurs des drones	8
1.4.1 Capteur ultrason	8
1.4.2 Caméra	9
1.4.3 Système de positionnement général (GPS)	9
1.4.4 Centrale inertielle (IMU)	9
1.5 Mission et Application des drones	9
1.6 Conclusion	10
CHAPITRE 2 COMMANDE ET OBSERVATION D'UN DRONE TYPE QUADRIROTOR	11
2.1 Intoduction	11

2.1.1	Modélisation et commande d'un Quadrotor	11
2.1.2	Modèle Cénimatique	12
2.1.3	Modilisation des forces Dynamique	13
2.1.4	Relation entre les vitesses angulaires et les angles d'Euler	15
2.1.5	Modèle dynamique non linéaire en utilisant Newton Euler	16
2.2	Commande d'un Quadrotor	18
2.2.1	Approches linéaires	18
2.2.2	Approches non linéaires	21
2.3	Attitude Estimation pour un Quadrotor	23
2.3.1	La notion d'observateur	24
2.3.2	Observateurs des systèmes linéaires	24
2.3.3	Observateurs des systèmes non linéaires	26
2.4	Conclusion	30
CHAPITRE 3 Test et résultats de simulation		31
3.1	Introduction	31
3.2	Simulation d'une commande PID d'un seul quadrirotor	31
3.2.1	Résultat de simulation	32
3.2.2	Interprétation des résultats	37
3.3	Estimation d'attitude d'un quadrotor a l'aide d'observateur Luenberger	37
3.3.1	Résultat de simulation	39
3.3.2	Interprétation des résultats	44
3.4	Estimation d'attitude d'un quadrotor a base d'un filtre de kalman	44
3.4.1	Filtre de Kalman étendu (Extended Kalman Filter, EKF)	45
3.4.2	Résultat de simulation	45
3.4.3	Interprétation des résultats	47
3.5	Estimation d'attitude d'un quadrotor a base d'un filtre de kalman sous ROS	48
3.5.1	Technique d'estimation d'angle ϕ, θ	49
3.5.2	Estimation d'angle ψ	50
3.6	Simulation d'un quadrirotor en utilisant ROS	51
3.6.1	Resultat d'une commande et implimentation d'un estimation d'angle d'un seul quadrotor	52

3.6.2	Interprétation des résultats	60
3.7	Conclusion	61
	RÉFÉRENCES	63
	ANNEXES	65

LISTE DES TABLEAUX

2.1	Comparaison des commandes quadrotor existante [Zulu, 2016]	23
3.1	Comparaison entre les observateurs	47
3.2	les distributions du ROS	52
D.1	Tableaux des paramètres physique	70
D.2	Tableaux des gains de réglage pour la commande PD	70

LISTE DES FIGURES

1.1	the flying bomb	4
1.2	Kettering Bug	4
1.3	queen	5
1.4	BoeingRB 47	6
1.5	espion U2	6
1.6	SA2	6
1.7	graph de classification des drones	8
2.1	Exemple d'un quadrirotor	11
2.2	Crazyflie dans le système de référence body-frame (OABC) et fixed-frame (OFI). Forces.[De Dinechin und Melquiond, 2015]	12
2.3	les configurations des quadrirotor [Adventures, 2012]	13
2.4	Mouvement d'un quadrirotor [M.GUETTACHE, 2019]	15
2.5	schéma de contrôleur PID d'un quadrotore [Zulu, 2016]	19
2.6	schéma de contrôleur LQR d'un quadrotor [Zulu, 2016]	20
2.7	Schéma de contrôle adaptatif pour la commande du quadrotor [Zulu, 2016]	21
2.8	Schéma de commande par mode glissant pour le contrôle du quadrotor [Zulu, 2016]	22
2.9	Schéma d'observateur Luenberger	25
2.10	Formulation mathématique du filtre de Kalman	26
2.11	Schéma d'observateur mode glissant	27
2.12	Schéma d'estimation d'attitude a l'aide d'un filtre linéaire complémentaire	29
3.1	Résultat de simulation de vecteur d'orientation	32
3.2	Résultat de simulation d'altitude avec PID	32
3.3	Angle θ avec présence de bruit	33
3.4	Angle ϕ avec présence de bruit	33
3.5	Angle ψ avec présence de bruit	34
3.6	vecteur de position	34
3.7	Résultat de simulation d'angle θ et la commande U3	35
3.8	Résultat de simulation d'angle ϕ et la commande U2	35
3.9	Résultat de simulation d'angle ψ et la commande U4	36
3.10	Résultat de simulation d'altitude Z et la commande U1	36

3.11	Angle θ observé	39
3.12	Angle ϕ observé	40
3.13	Angle ψ observé	40
3.14	erreur entre la valeur mesuré et observé	41
3.15	Angle θ observé avec présence de bruit	41
3.16	Angle ϕ observé avec présence de bruit	42
3.17	Angle ψ observé avec présence de bruit	42
3.18	erreur entre la valeur observé d'angle θ et l'entré avec présence de bruit	43
3.19	erreur entre la valeur observé d'angle ϕ et l'entré avec présence de bruit	43
3.20	erreur entre la valeur observé d'angle ψ et l'entré avec présence de bruit	44
3.21	Angle θ observé	46
3.22	Angle ϕ observé	46
3.23	Angle ψ observé	47
3.24	IMU	48
3.25	Schéma d'estimation d'attitude	51
3.26	IRIS quadrirotor	52
3.27	Rqt Graph d'implémentation d'estimateur d'angle ϕ	53
3.28	Rqt Graph d'implémentation d'estimateur d'angle θ	53
3.29	signal bruité d'accéléromètre	54
3.30	vitesse angulaire de gyromètre	54
3.31	champ magnétique de magnomètre	55
3.32	Résultats de l'implémentation d'un filtre kalman sur l'angle θ	55
3.33	Résultats de l'implémentation d'un filtre kalman sur l'angle ϕ	56
3.34	Un schéma descriptif de la compilation du programme entre ROS et gazebo	56
3.35	Graphe explique le passage des commandes au IRIS	57
3.36	RQT graph de la commande d'IRIS	58
3.37	RQT graph d'application de filtre de kalman sur IRIS	58
3.38	L'appel de quadrotor iris avec le fichier launch	59
3.39	Appel de node de commande PID	59
3.40	Appel depuis le nœud d'estimation qui publie dans Gazebo	60
A.1	schéma descriptif de traitement des paquets	67
C.1	Stabilité au sens de lyapunov	69

LISTE DES SIGLES ET ABRÉVIATIONS

UAV	Unmanned Aerial Vehicle
PID	Proportional Integral Derivative
LQR	Linear Quadratic Regulator
LQG	Linear Quadratic Gaussian
RPC	Linear Quadratic Estimator
PD	Proportional Derivative
3D	3 Dimensional cartesian space
ROS	Robot Operating System
TCP	Très Courte Portée
MALE	Moyenne Altitude et Longue Endurance
HALE	Haute Altitude et Longue Endurance
SLAM	Simultaneous Localization And Mapping
ROS	Robot Operating System
GPS	Global Positioning System
IMU	International Mathematical Union
RGB	red, green, blue
MAV	Micro Air Vehicule

LISTE DES ANNEXES

Annexe A	Robot Operating System/ROS	65
Annexe B	Gazebo	68
Annexe C	Notion sur la stabilité au sens de Lyapunov	69
Annexe D	Paramètres utilisés	70

Introduction

Afin d'assurer un haut niveau d'autonomie ce qui permet au quadrotor de mener à bien sa mission. L'estimation de l'orientation d'un drone aérien est essentielle, il est donc nécessaire d'incorporer une estimation robuste de l'orientation dans les algorithmes de contrôle pour garantir de bons résultats. Pour cela, il existe deux branches d'estimation dans la littérature. Estimation de l'orientation d'un véhicule à partir de mesures corporelles et d'observations de référence connues. Filtrage des mesures bruité en combinant les mesures avec des modèles. Une large variété de capteurs peut être utilisée dans la 2^{ème} branche telle que les capteurs inertiels (gyroscopes, accéléromètres), les magnétomètres, les capteurs de pression et les systèmes de vision. Les informations provenant des différents capteurs sont ensuite fusionnées en temps réel avec un modèle dynamique du système par l'algorithme d'estimation, afin de fournir une estimation de l'orientation du drone.

Le quadrotor est un UAV de la famille multirotor utilisé dans les domaines de la surveillance et de la reconnaissance à distance des réseaux routiers et attaques aériennes. Vu la faible précision (biais, bruit) des capteurs embarqués utilisés dans la conception de ce type de drone, il est donc nécessaire de disposer d'algorithmes d'estimation qui permettent la reconstruction robuste de l'orientation avec un minimum de capteurs disponibles.

Pour l'estimation de l'orientation d'un corps rigide, on trouve un nombre important de méthodes basées sur différentes paramétrisations possibles (Matrices de rotation, quaternion, angles d'Euler, paramètres ...). Néanmoins, dans des scénarios réalistes, les mesures sont biaisées et bruitées, ce n'est donc plus possible de reconstruire l'orientation. Par exemple, le magnétomètre est très facilement bruité par des perturbations électromagnétiques et des masses ferromagnétiques. De même, un accéléromètre triaxial idéal accroché au centre de masse mesure la projection dans le repère local de l'accélération spécifique et non juste l'accélération propre du drone, ce qui ne permet pas de connaître la direction de la gravité sauf dans des cas d'accélération faible. Pour cette raison, il semble intéressant d'utiliser un gyroscope afin de compenser la basse précision de mesures vectorielles disponibles. Si on cherche dans la littérature d'estimation de l'orientation à base de mesures vectorielles on trouve deux branches. Le filtrage stochastique sauf que cette approche ne peut pas garantir une convergence globale à cause de l'inconvénient de reconstruction de vecteur l'orientation qui est reconstruite algébriquement juste à partir des mesures à l'instant t , sans filtrer et sans exploiter l'information par le gyro. Les observateurs avec fusion de donner cette approche sont les plus récents et les plus prometteurs, proposent des observateurs non linéaires avec un large domaine d'attraction et un réglage de gain simple, L'approche habituelle dans la

littérature à base d'observateurs non linéaires est d'estimer l'orientation et les biais gyro à partir d'un modèle dynamique, de 2 mesures vectorielles (accélérations, mesures du champ magnétique) et de mesures gyro. Les hypothèses principales dans la plupart des travaux étant que les mesures vectorielles sont non biaisées et que le biais gyro est constant, afin d'assurer des meilleures performances, il est nécessaire de considérer le problème de l'estimation de l'orientation en présence de biais de mesures vectorielles mais aussi le cas où une seule mesure vectorielle est fiable.

Tout au long de notre travail, dans un premier temps on réalise un état de l'art sur cette problématique puis le développement d'une technique d'estimation à l'aide d'un filtre de Kalman. La validation des travaux sera réalisée à partir de données expérimentales obtenues en utilisant des bags sous ROS. Donc notre mémoire s'articule autour de ces trois chapitres :

— chapitre I :

Résume toutes les notions de base nécessaires à la compréhension du domaine des drones.

— chapitre II :

Tout d'abord, une modélisation du drone quadrirotor est introduit avec un modèle dynamique basé sur les équations de Newton-Euler. Nous élaborerons ensuite, un état de l'art sur les différentes techniques de commande de position et d'orientation du quadrirotor et aussi sur la problématique d'estimation afin de sélectionner des méthodes de référence qui serviront de comparaison.

— chapitre III :

Comporte plusieurs simulations en boucle fermée montrant la stabilité et la robustesse des commandes d'un quadrirotor, avec des interprétations des résultats pour chaque partie. Par la suite, les techniques d'estimation choisissent avec leurs résultats. Nous discutons également de l'influence des paramètres sur les résultats obtenus. Nous allons aussi présenter l'environnement de simulation ROS ou nous avons testé toutes les commandes en temps réel.

— Enfin, nous concluons le manuscrit par une synthèse des résultats obtenus ainsi que par la proposition de quelques perspectives.

CHAPITRE 1 REVUE DE LITTÉRATURE

1.1 Introduction

Au début du nouveau millénaire, les progrès réalisés dans de nombreuses technologies comme : les matériaux, l'électronique, les ordinateurs, les capteurs, les instruments de navigation ont contribué à l'avènement des drones, les hélicoptères et d'autres configurations de véhicules aériens pour assurer des services civils comprennent la surveillance, le sauvetage (maritime et en montagne), l'ambulance aérienne, la lutte contre les incendies, etc... et des fonctions militaires comprennent le déminage, la surveillance du champ de bataille, etc...

Aujourd'hui, les drones UAV jouent un rôle vraiment important dans l'aviation moderne. Les générations des drones en vol stationnaire, sont conçus pour être petits, plus légers et dotés de certaines fonctions d'autonomie.

Au cours de ce chapitre, nous présentons de manière générale ce qu'est un drone, les types de drones, puis nous présentons les capteurs de situation des drones et nous terminons par les missions et les applications des drones (civils et militaires).

1.2 L'EVOLUTION DES DRONES AU FILS DU TEMPS

1.2.1 les premiers drones

Les tous premiers drones commencent en 1883 lorsque Douglas Archibald attacha un anémomètre à un jouet léger qui s'élève en l'air à l'aide du vent en étant retenu par une ficelle. La mesure de la vitesse du vent qu'il effectua s'éleva à 400m. Après 5 ans Arthur battu un cerf-volant d'une appareil photo et réussit la première photo aérienne le 20 juin 1888 à Paris. Ce furent alors les premiers engins volants équipés pour la surveillance.

1.2.2 L'entre-deux guerres

La première guerre mondiale nous a donné idée de développement de deux véhicules volants sans pilotes à bords « the flying bomb » (Figure.1.1) de la Navy et le « kettering bug » (Figure.1.2) de Charles kettering.

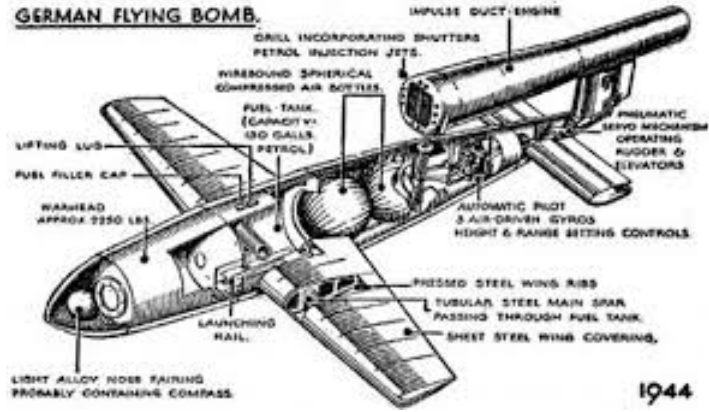


Figure 1.1 the flying bomb



Figure 1.2 Kettering Bug

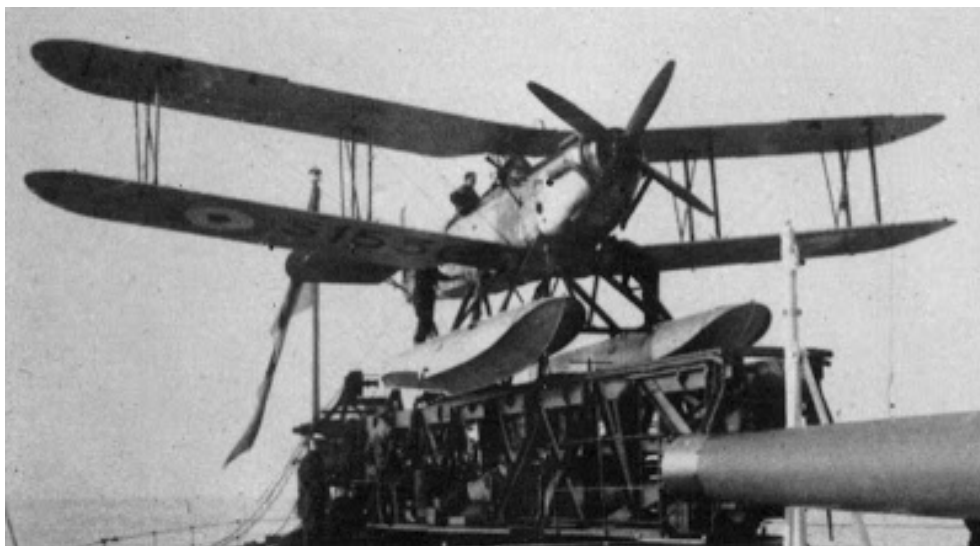


Figure 1.3 queen

De l'autre côté, les anglais développèrent les premiers appareils radiocommandés sans pilotes : les « Fairey Queens » (reine des fées). Ces modèles laissèrent leurs places dans les années 30 aux modèles « Queen Bee » (reine des abeilles) qui sont à l'origine du terme « drone » (faux bourdon ou abeille mâle) (Figure.1.3).

1.2.3 LA GUERRE FROIDE

Le premier événement majeur pour le développement de drones plus intense s'est produit dans le premier Mai 1960. Un avion (espion U-2) (Figure.1.5) de l'armée américaine fut abattu par l'armée soviétique alors qu'il survolait le territoire de l'URSS pour y prendre des photographies et son pilote fut capturé. Le premier juillet de la même année, un (BoeingRB-47) (Figure.1.4) de reconnaissance fut à son tour abattu par l'armée soviétique alors qu'il survolait les frontières de l'URSS. Quatre passagers furent tués, et les deux autres capturés.

Ces incidents avec l'URSS en pleine guerre froide posèrent des questions de remplacé les avions par les drones ou des satellites. En 1962 les drones de reconnaissances furent définitivement mirent en route après qu'un (U-2) fut abattu par un (SA-2) (Figure.1.6) SAM (surface-to-air missile) soviétique alors qu'il survolait Cuba pour déterminer les sites des missiles soviétiques. Cet avion-espion laissa son nom à un célèbre groupe irlandais en 1978.



Figure 1.4 BoeingRB 47



Figure 1.5 espion U2



Figure 1.6 SA2

1.2.3.1 Les années 1990

Après que l'emploi du drone connut un succès lors des guerres la DARPA à lancer plusieurs programmes pour financer le développement d'autres drones comme les drones tactiques pour le développement de la première ligne du front, les drones de moyenne portée pour surveiller la zone des opérations, les drones de grandes portées et de grande autonomie pour rechercher les cibles plus lointaines et les drones discrets pour surveiller les zones très bien gardées.

1.3 CLASSIFICATION DES DRONES

Les normes de classification des drones diffèrent selon les pays et leurs besoins. Cependant, ils peuvent être classés selon trois critères :

- * L'altitude de croisière
- * L'endurance en terme de temps de vol
- * La dimension du drone

Mais le domaine opérationnel des drones peut se décomposer en 3 segments :

- les drones tactiques qui se décomposent eux-mêmes en six segments :
 1. les micro-drones (Micro Air Vehicle ou MAV), pouvant être contenu dans une sphère de 30 cm
 2. les mini-drones (Mini Air Vehicle ou MAV également), pouvant être contenu dans une sphère de 70 cm ;
 3. les drones de très courte portée (TCP)
 4. les drones moyenne portée lents (multicharges multimiSSIONS ou MCMM lents)
 5. les drones rapides basse altitude (MCMM rapides)
 6. les drones maritimes tactiques (DMT)
- les drones de Moyenne Altitude et Longue Endurance (MALE) permettant d'utiliser une charge utile de l'ordre de 100kg.
- les drones de Haute Altitude et Longue Endurance (HALE)

Là (Figure.1.7) peut résumer un cas des classifications mondial des drones.

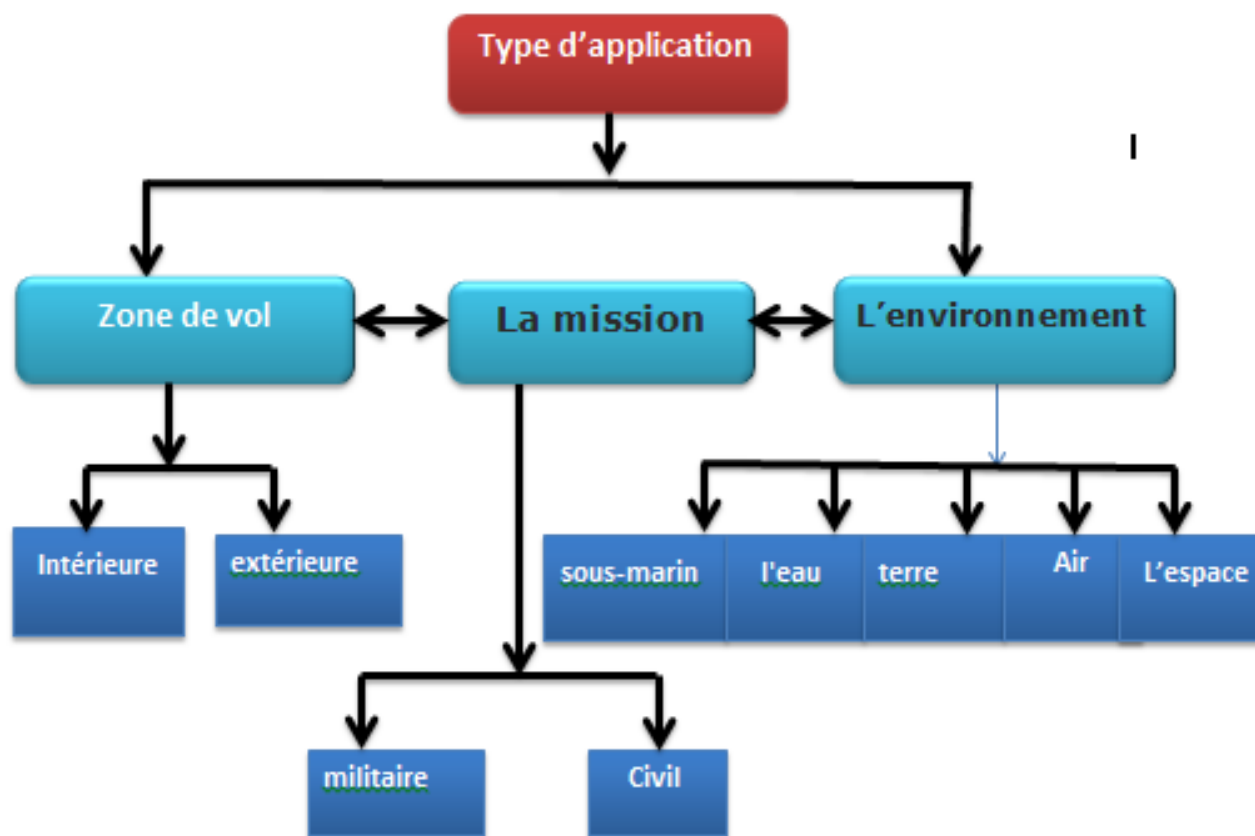


Figure 1.7 graph de classification des drones

1.4 Technologie des capteurs des drones

Les capteurs servent principalement à effectuer des mesures tridimensionnelles (position, vitesse, orientation, accélération ...) permettant d'alimenter des entrées en temps réel pour les lois de commande. Ces mesures sont effectuées par différents types de capteurs :

1.4.1 Capteur ultrason

Il sert à détecter l'obstacle et à mesurer la distance de séparation. Se munir uniquement d'un capteur à ultrason pour la mesure de position peut donner des résultats erronés. En effet, en couvrant les lieux, l'onde peut tomber sur différents obstacles à chaque instant et des distances différentes. Il est donc primordial de l'associer à un autre capteur de position.

1.4.2 Caméra

Le capteur le plus riche en données pouvant exister est une caméra. En général, nous pouvons classer les caméras en deux catégories :

- * Caméra numérique.
- * Caméra analogique.

Le choix de la caméra dépend de son embarquabilité, ses capacités de calculs et d'application elle-même. Les modèles les plus utilisés pour SLAM sont : les caméras monoculaires, les caméras stéréos, les caméras RGB-D....

1.4.3 Système de positionnement général (GPS)

Le GPS est un système de positionnement par satellites assez précis "la précision pouvant atteindre le mètre en présence de filtrage". L'utilisation de la localisation par le système **GPS!** est très vaste, elle touche plusieurs domaines : le transport (aérien, maritime. . .), étude géographique, assistance aux équipes de secours ...

1.4.4 Centrale inertielle (IMU)

La centrale inertielle est une carte intégrant trois types de capteurs

- Gyroscopes : un appareil qui exploite le principe de la conservation du moment angulaire en mécanique des solides
- Magnétomètres : un appareil qui sert à mesurer l'aimantation d'un système. Dans notre cas le magnétomètre sert à calculer le champ magnétique terrestre.
- Accéléromètres : un capteur qui, fixé à un mobile, permet de mesurer l'accélération de ce dernier.

1.5 Mission et Application des drones

Les drones sont développés à l'origine pour remplacer l'homme dans des environnements hostiles ou des situations dangereuses. Ces engins sans pilotes présentent de nombreux avantages, c'est très probablement dans le domaine civil comme dans le domaine militaire que les drones sont appelés à jouer le plus grand rôle, et ce du fait de leur souplesse, et de leur polyvalence d'emploi, les exemples d'applications potentielles sont divers.

- Dans le domaine de la sécurité : surveillance de l'espace aérien, du trafic urbain et interurbain .

- Dans la gestion des risques naturels : surveillance de l'activité des volcans .
- La protection de l'environnement : mesure de la pollution de l'air, surveillance des forêts.
- L'intervention dans des sites hostiles : milieux radioactifs, déminage des terrains (cartographie de terrains minés).
- La gestion des grandes infrastructures : barrages, lignes à haute tension, pipelines.
- L'agriculture : détection et traitement des cultures
- La prise de vue aérienne dans la production des films
- Télécommunications mobiles, publicité et radiodiffusion (télévision, ...).
- Géodésie et mesures atmosphériques.

1.6 Conclusion

Nous avons présenté une grande diversité de familles de drones à voilure fixe et tournantes, qui ont chacune leurs spécifications techniques.

le chapitre suivant fait l'objet de la modélisation d'un drone de type quadrotor.

CHAPITRE 2 COMMANDE ET OBSERVATION D'UN DRONE TYPE QUADRIROTOR

2.1 Introduction

Le Quadrotor est un robot commandé par une unité centrale, équipé de quatre rotors. C'est un robot aérien qui utilise des capteurs comme l'IMU et des télémètres pour estimer son altitude et son attitude.

Pour une localisation précise, il utilise le GPS dans des scénarios en extérieur ou des systèmes de capture de mouvement dans des scénarios de vol en intérieur. Ce chapitre traite de la modélisation mathématique du quadrotor et les types de commande de position et attitude ainsi les algorithmes d'estimation.

2.1.1 Modélisation et commande d'un Quadrotor

Quand on parle d'un quadrotor on parle d'un corps rigide avec quatre rotors aux quatre coins qui forment un carré. Ce corps rigide a six degrés de liberté dans l'espace 3D. Cependant, sa configuration d'actionneur ne permet de contrôler que quatre degrés de liberté, à savoir la position le long de x, y et z et le lacet. Il s'agit donc d'un véhicule sous-actionné. (le nombre des entrées inférieur au nombre des sorties)



Figure 2.1 Exemple d'un quadrirotor

2.1.2 Modèle Cénématique

Il existe deux cadres de référence de coordonnées différents associés à la modélisation d'un quadrotor :

- **Le repère terrestre(inertielle ou fixe)** : est noté $O_{FI}(X, Y, Z)$ C'est un repère d'origine O_{FI} et d'axes X , Y et Z lié à la terre supposée immobile
- **Le repère lié au corps de quadrotor** : Le cadre $O_{ABC}(e_x, e_y, e_z)$ est le cadre qui est fixé au quadrotor de telle manière que son origine coïncide avec le centre de gravité du véhicule. Les quatre rotors se trouvent tous sur le plan $O_x - O_y$. L'axe z positif est orienté vers le haut, à l'opposé du sol, comme le montre (Figure.2.2).

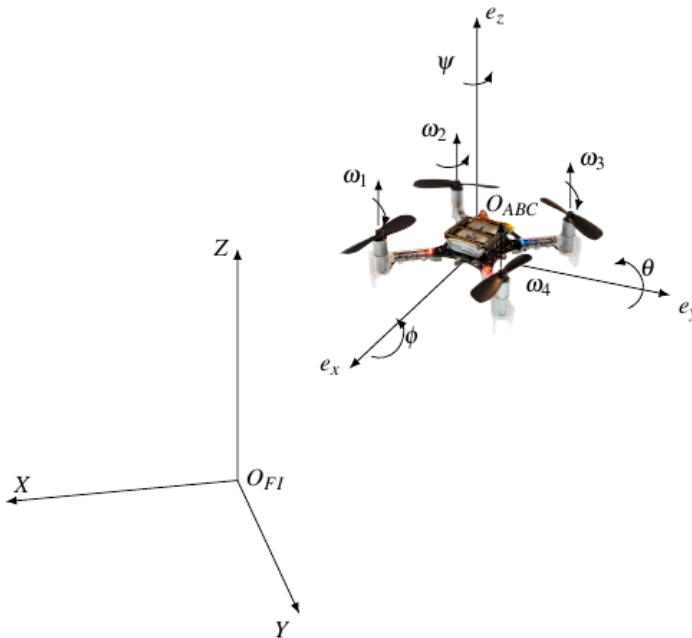


Figure 2.2 Crazyflie dans le système de référence body-frame (O_{ABC}) et fixed-frame (O_{FI}). Forces.[De Dinechin und Melquiond, 2015]

La rotation du cadre inertielle par rapport aux axes (x,y,z) , est représentée par la matrice de rotation R utilisant les angles d'Euler.

$$\phi = Phi \quad \theta = Theta \quad \psi = Psi$$

$$R = \begin{bmatrix} \cos \theta \cos \psi & \sin \phi \cos \psi - \cos \phi \sin \psi & \cos \phi \sin \theta \cos \psi + \sin \phi \sin \psi \\ \cos \theta \sin \psi & \sin \phi \sin \theta \sin \psi + \cos \phi \cos \psi & \cos \phi \sin \theta \sin \psi - \sin \phi \cos \psi \\ -\sin \theta & \sin \phi \cos \theta & \cos \phi \cos \theta \end{bmatrix}. \quad (2.1)$$

Les quadrotors sont généralement conçus dans deux configurations à savoir "X" et "+". La plupart des chercheurs utilisent la configuration (X) pour s'assurer que le centre de masse est situé au centre du drone.

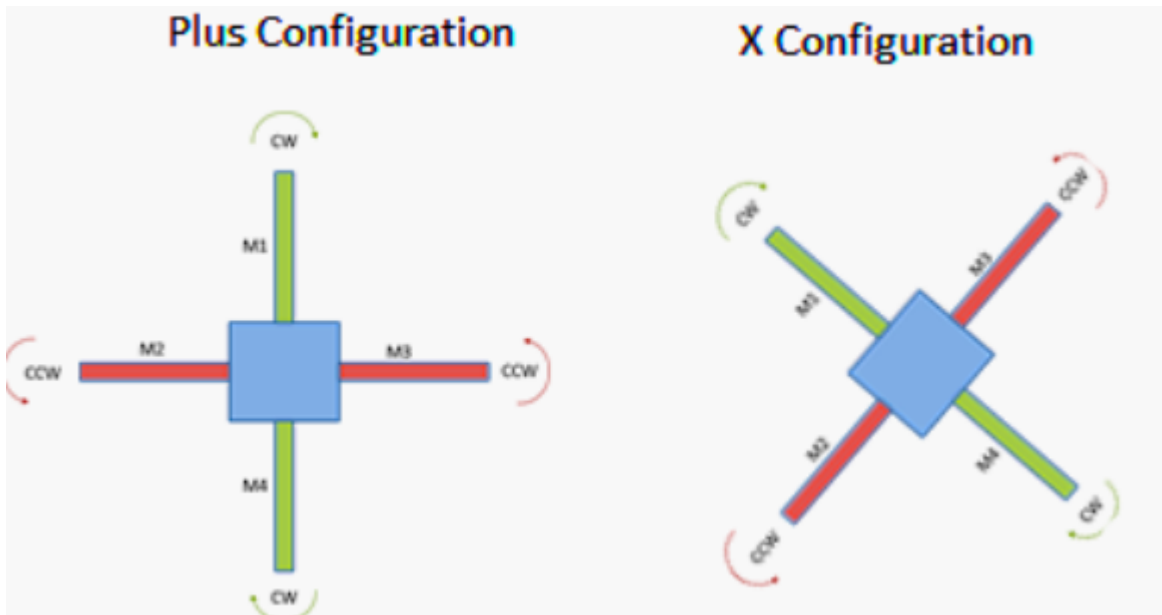


Figure 2.3 les configurations des quadrirotor [Adventures, 2012]

2.1.3 Modilisation des forces Dynamique

2.1.3.1 Hypothèses

Afin de modéliser les forces dynamique de quadrotor, on suppose les hypothèses suivantes :

- La structure du quadrotor est supposée rigide et symétrique.
- La matrice d'inertie J est supposée constante
- Les forces de portance et de trainée sont supposées proportionnelles au carré de la vitesse de rotation des rotors.
- Le centre de gravité . est supposé confondu avec Le repère liée au corps du quadrotor

2.1.3.2 force de poussée

La force de poussée en régime permanent générée par un rotor en vol stationnaire (c'est-à-dire un rotor qui ne se déplace pas horizontalement ou verticalement elle a tendance à faire élever le quadrirotor) peut être modélisé en pratique par (après la simplification de la surface du disque du rotor,...)

$$F_i = C_T \Omega_i^2 \quad (2.2)$$

où $i^{\text{ème}}$ représente le rotor, Ω_i la vitesse angulaire (ou vitesse de rotation) et C_T représente le coefficient de portance déterminée à partir de tests statiques. [M.GUETTACHE, 2019]

2.1.3.3 force de trainée réaction

C'est la réaction créée par la rotation des quatre hélices :

$$M_i = C_D \Omega_i^2 \quad (2.3)$$

. C_D représente la constante de trainée et peut être déterminée par des essais statiques de poussée.

La force totale générée dans la direction z est donnée par la somme des poussées individuelles générées par chacun des quatre rotors. La force générée par les rotors (la Figure.2.4) est donnée par :

$$F = \sum_{i=1}^4 F_i \quad (2.4)$$

$$F = C_T(\Omega_1^2 + \Omega_2^2 + \Omega_3^2 + \Omega_4^2) \quad (2.5)$$

2.1.3.4 Les couples aérodynamiques actif

Les couples aérodynamiques appliqués au quadrirotor s'écrivent alors (2.14)

$$\tau = \begin{bmatrix} \tau_\phi \\ \tau_\theta \\ \tau_\psi \end{bmatrix} = \begin{bmatrix} l.C_T(F_2 - F_4) \\ l.C_T(F_3 - F_1) \\ C_D \cdot [-(M_1 + M_3) + (M_2 + M_4)] \end{bmatrix} \quad (2.6)$$

Où :

- $L \in R^+$, est la distance entre le centre de gravité du quadrirotor et l'axe de rotation de l'un des rotors

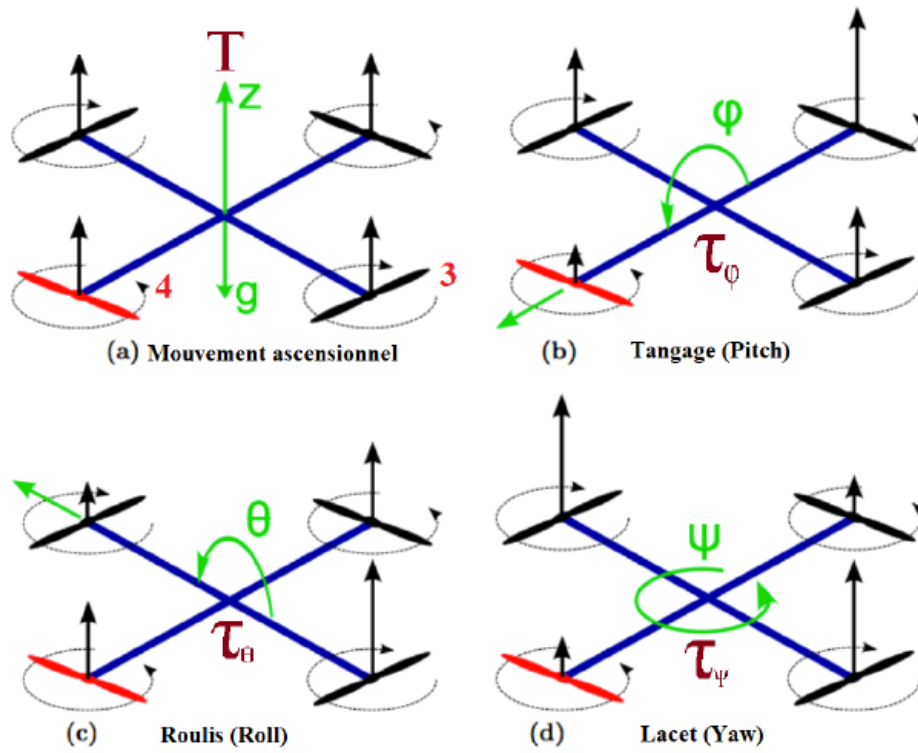


Figure 2.4 Mouvement d'un quadrirotor [M.GUETTACHE, 2019]

D'après les équations ci-dessus (2.14) et (2.5) on obtient une matrice qui relie la poussée et les couples aérodynamiques aux vitesses des rotors :

$$\begin{bmatrix} F \\ \tau_\phi \\ \tau_\theta \\ \tau_\psi \end{bmatrix} = \begin{bmatrix} C_T & C_T & C_T & C_T \\ 0 & lC_T & 0 & -lC_T \\ -lC_T & 0 & lC_T & 0 \\ C_D & -C_D & C_D & -C_D \end{bmatrix} \begin{bmatrix} \Omega_1^2 \\ \Omega_2^2 \\ \Omega_3^2 \\ \Omega_4^2 \end{bmatrix} \quad (2.7)$$

2.1.4 Relation entre les vitesses angulaires et les angles d'Euler

Lorsque un solide tourne à une vitesse constante, sa vitesse angulaire est constante, par contre les variations des angles d'Euler seront variables car elles dépendent des angles instantanés entre les axes des deux repères. La séquence des angles d'Euler est obtenue à partir de trois

rotations successives : lacet, tangage et roulis.

$$\begin{bmatrix} p \\ q \\ r \end{bmatrix} = R_\phi R_\theta \begin{bmatrix} 0 \\ 0 \\ \dot{\psi} \end{bmatrix} + R_\phi \begin{bmatrix} 0 \\ \dot{\theta} \\ 0 \end{bmatrix} + \begin{bmatrix} \dot{\phi} \\ 0 \\ 0 \end{bmatrix} \quad (2.8)$$

Pour des raisons de simplification et comme la plupart des cas étudiés dans la littérature travaillent avec un modèle simplifié où p , q et r représentent les vitesses angulaires autour des trois axes principaux du quadrotor. donc l'équation devient :

$$\begin{bmatrix} p \\ q \\ r \end{bmatrix} = \begin{bmatrix} \dot{\phi} \\ \dot{\theta} \\ \dot{\psi} \end{bmatrix} \quad (2.9)$$

2.1.5 Modèle dynamique non linéaire en utilisant Newton Euler

La dynamique des quadrotors est modélisée à l'aide des équations de mouvement de Newton Euler pour un corps rigide dans l'espace libre .

En utilisant les hypothèses de la (section.2.1.3.1) les équations des mouvements sont les suivantes :

Mouvement de translation

$$m\ddot{r} = \begin{bmatrix} 0 \\ 0 \\ -mg \end{bmatrix} + \mathbf{R} \begin{bmatrix} 0 \\ 0 \\ F_1 + F_2 + F_3 + F_4 \end{bmatrix} \quad (2.10)$$

Mouvement de rotation

$$I \begin{bmatrix} \dot{p} \\ \dot{q} \\ \dot{r} \end{bmatrix} = \tau - \mathbf{R} \begin{bmatrix} p \\ q \\ r \end{bmatrix} I \begin{bmatrix} p \\ q \\ r \end{bmatrix} \quad (2.11)$$

où p , q et r représentent les vitesses angulaires autour des trois axes principaux du quadrotor et τ est défini dans l'équation(2.14)

Le modèle dynamique du quadrotor est donc donné par le système d'équations suivant :

$$\begin{cases} \ddot{x} = \left(\frac{F}{m}\right)(\cos \phi \sin \theta \cos \psi + \sin \phi \sin \psi) \\ \ddot{y} = \left(\frac{F}{m}\right)(\cos \phi \sin \theta \cos \psi - \sin \phi \cos \psi) \\ \ddot{z} = -g + \cos \phi \cos \psi \left(\frac{F}{m}\right) \\ \ddot{\phi} = \left(\frac{I_y - I_z}{I_x}\right) \dot{\theta} \dot{\psi} - I_p \Omega \dot{\theta} \left(\frac{1}{I_x}\right) + \left(\frac{1}{I_x}\right) \tau_\phi \\ \ddot{\theta} = \left(\frac{I_z - I_x}{I_y}\right) \dot{\phi} \dot{\psi} - I_p \Omega \dot{\phi} \left(\frac{1}{I_y}\right) + \left(\frac{1}{I_y}\right) \tau_\theta \\ \ddot{\psi} = \left(\frac{I_x - I_y}{I_z}\right) \dot{\phi} \dot{\theta} + \left(\frac{1}{I_z}\right) \tau_\psi \end{cases} \quad (2.12)$$

I_x, I_y, I_z représente le moment d'inertie autour des trois axes principaux du quadrotor et I_p représente le moment d'inertie de l'hélice autour de son propre axe.

2.1.5.1 Modèle d'état

Pour construire un modèle d'état, nous choisissons comme vecteur état :

$$[x, y, z, \dot{x}, \dot{y}, \dot{z}, \phi, \theta, \psi, \dot{\phi}, \dot{\theta}, \dot{\psi}]^T = [x_1, x_2, x_3, x_4, x_5, x_6, x_7, x_8, x_9, x_{10}, x_{11}, x_{12}]^T \quad (2.13)$$

et les commandes :

$$\begin{bmatrix} U_1 \\ U_2 \\ U_3 \\ U_4 \end{bmatrix} = \begin{bmatrix} T \\ \tau_\phi \\ \tau_\theta \\ \tau_\psi \end{bmatrix} = \begin{bmatrix} C_T \cdot (\Omega_1^2 + \Omega_3^2 + \Omega_2^2 + \Omega_4^2) \\ l \cdot C_T \cdot (\Omega_2^2 - \Omega_4^2) \\ l \cdot C_T \cdot (\Omega_3^2 - \Omega_1^2) \\ C_D \cdot [(\Omega_1^2 + \Omega_3^2) - (\Omega_2^2 + \Omega_4^2)] \end{bmatrix} \quad (2.14)$$

La représentation d'état obtenue on utilisant (2.13) et (2.14) :

$$\left\{ \begin{array}{l} \dot{x}_1 = x_4 \\ \dot{x}_2 = x_5 \\ \dot{x}_3 = x_6 \\ \dot{x}_4 = \frac{U_1}{m} \cdot (\cos(x_7) \sin(x_8) \cos(x_9) + \sin(x_7) \sin(x_9)) \\ \dot{x}_5 = \frac{U_1}{m} \cdot (\cos(x_7) \cdot \sin(x_8) \cdot \sin(x_9) - \sin(x_7) \cdot \cos(x_9)) \\ \dot{x}_6 = \frac{U_1}{m} \cdot (\cos(x_7) \cos(x_8)) - g \\ \dot{x}_7 = x_4 \\ \dot{x}_8 = x_4 \\ \dot{x}_9 = x_4 \\ \dot{x}_{10} = \left(\frac{I_y - I_z}{I_x} \right) x_5 x_6 - I_p \Omega x_5 \left(\frac{1}{I_x} \right) + \left(\frac{1}{I_x} \right) U_1 \\ \dot{x}_{11} = \left(\frac{I_z - I_x}{I_y} \right) x_4 x_6 - I_p \Omega x_4 \left(\frac{1}{I_y} \right) + \left(\frac{1}{I_y} \right) U_2 \\ \dot{x}_{12} = \left(\frac{I_x - I_y}{I_z} \right) x_4 x_5 + \left(\frac{1}{I_z} \right) U_3 \end{array} \right. \quad (2.15)$$

2.2 Commande d'un Quadrotor

Il existe différentes techniques de contrôle pour contrôler les quadrirotors (linéaire et non linéaire) du au instabilité de quadrirotor. Dans la littérature, les algorithmes proposés : PID, régulateur quadratique linéaire (LQR), mode glissant, Backstepping, Feedback Linearization, contrôle adaptatif, algorithme de contrôle hybride et algorithmes de contrôle optimal.

2.2.1 Approches linéaires

Les commandes linéaires examinées considèrent des systèmes linéaires et invariants dans le temps (LTI), qui peuvent être représentés par un modèle d'état de la forme :

$$\begin{aligned} \dot{X} &= AX + BU \\ y &= CX + DU \end{aligned}$$

où x est le vecteur des états du système, u est le vecteur des commandes, y est le vecteur des sorties, et les matrices A , B , C et D caractérisent le système.

2.2.1.1 Proportionnel, Intégral et Dérivé

Cette approche est la plus intuitive et la plus simple à embarquer sur un processeur. Elle permet de comprendre facilement le rôle physique de chacun des termes de la commande, ce qui permet de régler les gains en conséquence. L'un de ses avantages aussi est qu'il est facile de régler les gains et sa robustesse, mais L'inconvénient principal associé au contrôle PID est les approximations faites lors de la linéarisation du modèle non linéaire. Cela limite les performances du contrôle ([Li und Li, 2011]). Comme indique (Figure.2.5) à partir de l'erreur existence entre la référence et la mesure, de son intégrale et de sa dérivée. Bien que l'analyse d'un système piloté avec un PID soit simple, sa conception peut être difficile, car il n'existe pas de méthode unique pour déterminer le gain intervenant dans le processus.

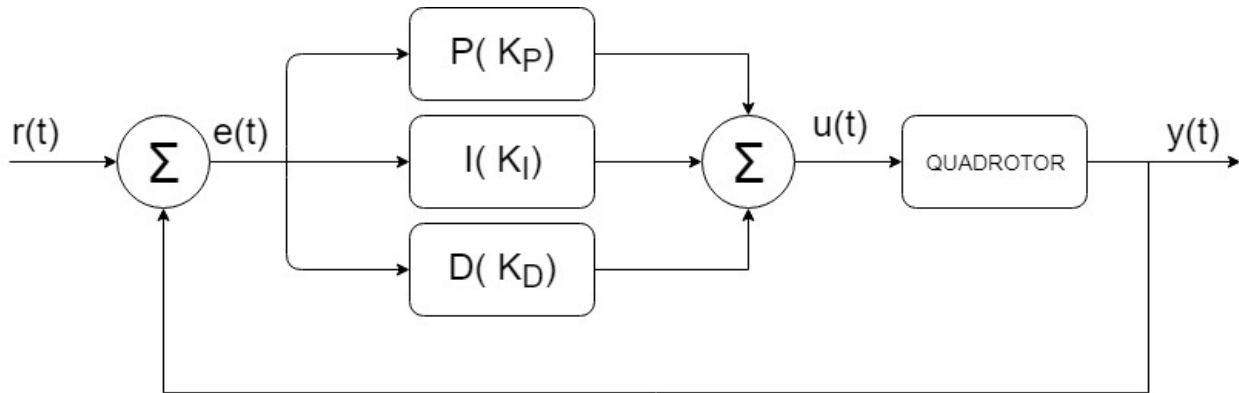


Figure 2.5 schéma de contrôleur PID d'un quadrotore [Zulu, 2016]

2.2.1.2 Régulateur quadratique linéaire

Dans cette méthode de synthèse qui permet de déterminer la commande optimale d'un système qui minimise (ou maximise) un critère quadratique de performance (2.16) le schéma de contrôle de cet algorithme peut être vu dans (Figure.2.6).

$$J_{LQR} = \int_0^{\infty} (x_i^T(t)Qx_i(t) + u_i^T(t)Ru_i(t))dt \quad (2.16)$$

Q et R sont les matrices de pondérations. L'avantage de la commande linéaire quadratique est qu'elle possède intrinsèquement des très bonnes propriétés de robustesse. Toutefois, une telle stabilité n'est assurée que dans l'hypothèse où le modèle est parfaitement connu, où la totalité de l'état est disponible et où les signaux ne sont pas bruités ([Palunko und Fierro, 2011]).

La gaussien quadratique linéaire (LQG) est une combinaison de l'estimateur quadratique

linéaire (LQE) et d'un filtre de Kalman. Dans [Minh und Ha, 2010], le LQG avec une action intégrale a été mis en oeuvre pour la stabilisation de l'attitude en vol stationnaire et de bonnes performances ont été observées. L'avantage de cette technique est qu'il n'est pas nécessaire de disposer d'une information complète sur l'état.

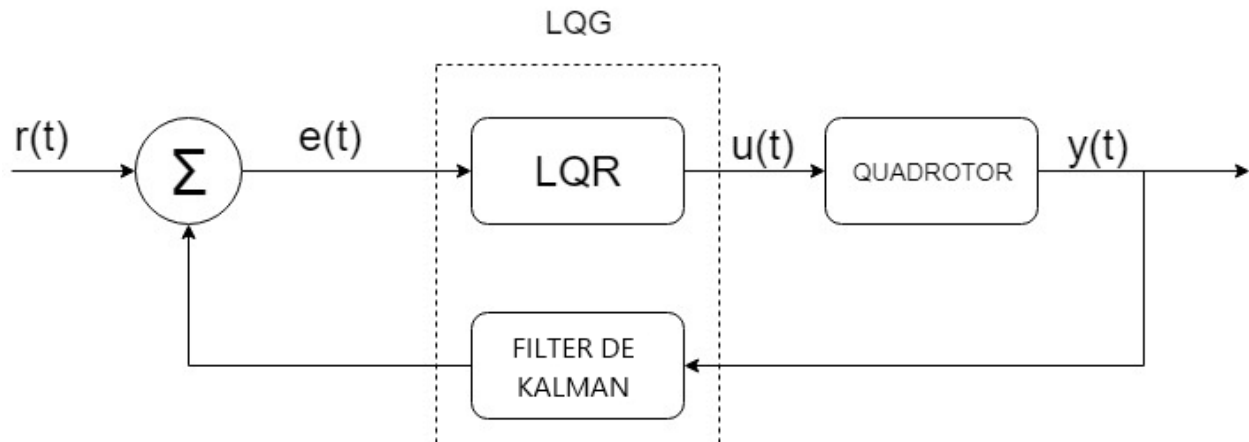


Figure 2.6 schéma de contrôleur LQR d'un quadrotor [Zulu, 2016]

2.2.1.3 Contrôle Adaptatif

Cet algorithme de contrôle est conçu pour s'adapter aux paramètres variables dans le temps et aux incertitudes du modèle donc la nécessité de contrôler des systèmes inconnus et des systèmes variantes dans le temps elle consiste à rejeter la perturbation une fois qu'elle a été estimée ou mesurée. est disponible. cette idée nous a permis de créer une telle commande ([M.Mokhtari, 2015]), [Palunko und Fierro, 2011], [Palunko und Fierro, 2011]. (Figure.2.7) montre un schéma général d'un système de contrôle adaptatif.

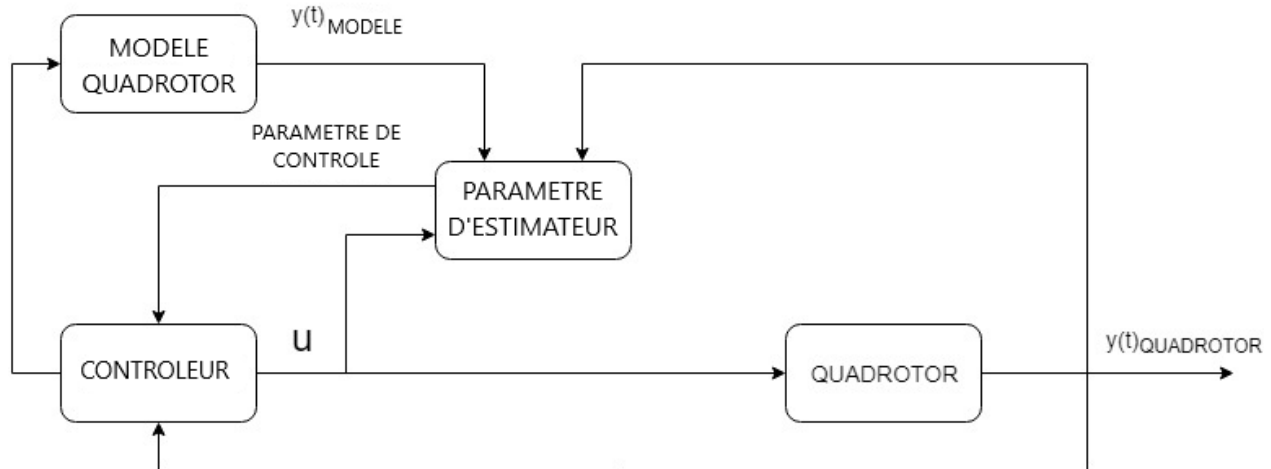


Figure 2.7 Schéma de contrôle adaptatif pour la commande du quadrotor [Zulu, 2016]

2.2.1.4 Linéarisation par retour d'état

Par le changement des variables, cette méthode transforme un modèle de système non linéaire en un système linéaire équivalent. Le problème de cette méthode est qu'elle nécessite un modèle exact du système et qu'il existe des approximations dans la linéarisation du modèle non linéaire.

2.2.2 Approches non linéaires

La commande des systèmes non linéaires est beaucoup plus complexe que celle des systèmes linéaires. Plusieurs méthodes de synthèse de contrôleurs non linéaires existent, mais requièrent généralement une connaissance précise du modèle et ne sont généralement pas aussi systématiques que les méthodes de synthèse de contrôleurs linéaires.

2.2.2.1 Commande par mode glissant

Le principe de la méthode des modes glissants est basé sur l'idée de considérer une surface S et un système dynamique tel que :

$$\dot{x} = f(x) = \begin{cases} f^+(x) & \text{si } S(x) > 0 \\ f^-(x) & \text{si } S(x) < 0 \end{cases} \quad (2.18)$$

On dit que le système est en régime glissant si toutes les trajectoires convergent vers la surface S et y demeurent. Les principaux avantages de cette méthode sont un bon suivi et le fait que le contrôle est appliqué directement au modèle non linéaire. Il n'est pas nécessaire de linéariser le modèle autour d'un point d'équilibre.

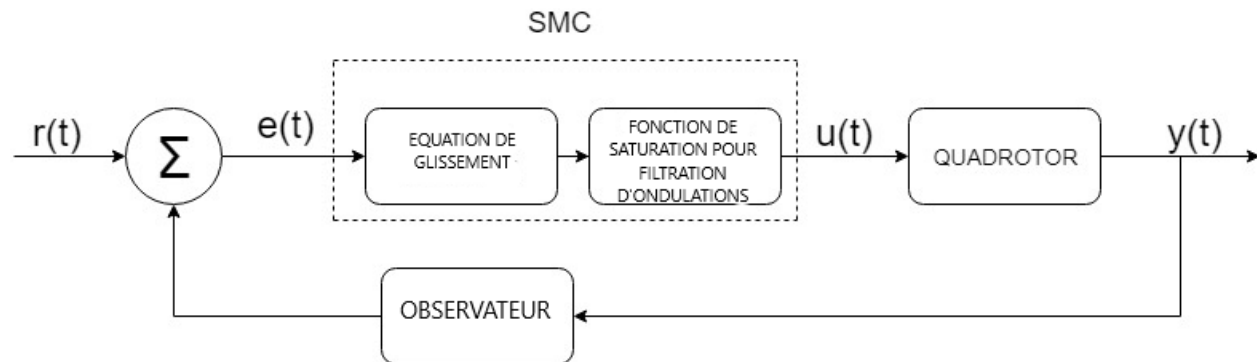


Figure 2.8 Schéma de commande par mode glissant pour le contrôle du quadrotor [Zulu, 2016]

2.2.2.2 Commande par backstepping

La commande par Backstepping est une loi de commande pour les systèmes non linéaires basée sur le théorème de Lyapunov¹. Chaque état du système commande l'état précédent et s'appelle « Commande Virtuelle », jusqu'au dernier état qui est commandé par la commande réelle.

L'objectif de cette technique est de calculer en plusieurs étapes une loi de commande qui assure la stabilité globale du système. Le backstepping est utilisé pour obtenir un bon suivi de la position et du lacet avec une stabilité en roulis et en tangage du quadrotor.[Madani, 2006]

2.2.2.3 La commande par logique floue

Les modèles flous peuvent être considérés comme des modèles logiques qui utilisent des règles du type « Si... Alors... ». Pour établir des relations qualitatives entre les variables du modèle. Les systèmes flous peuvent être classés en trois groupes : les systèmes flous linguistiques ou systèmes de Mamdani, les systèmes flous relationnels et les systèmes à conséquence fonctionnelle ou encore connus sous le nom de systèmes flous de type Takagi-Sugeno-Kang. (Tableau.D.1) Un tableau résumé quelque performances assuré par chaque commande²

1. Notion sur la stabilité au sens de Lyapunov (C)

2. 0 \mapsto faible, 1 \mapsto moyenne, 2 \mapsto assuré.

Tableau 2.1 Comparaison des commandes quadrotor existante [Zulu, 2016]

Commande	robuste	adaptative	précis	intelligent	optimal	perte d'énergie
PID	1	0	1	0	0	0
LQR	0	2	0	0	1	0
LQG	0	2	0	0	2	0
BACKSTEPPING	0	0	1	0	2	0
LOGIQUE FLOU	1	1	1	2	1	0
RÉSEAUX NORAUX	1	1	1	2	2	0
H_∞	2	2	1	0	1	0
FDL	1	1	2	0	0	0
SMC	1	2	2	0	1	2

2.3 Attitude Estimation pour un Quadrotor

L'estimation de l'état le processus d'extraction d'informations sur des variables d'intérêt a partir de mesures bruitées et qui peuvent être liées à ces variables par des modèles mathématiques complexes. Il assure une autonomie augmentée qui permet au drone de mener avec succès sa mission (suivi de cible, évitement d'obstacles, exploration d'espaces etc.). On peut résumer l'estimation dans deux branches :

- Estimation de l'orientation d'un véhicule à partir de mesures corporelles et d'observations de référence connues.
- Filtrage de mesures bruitées en combinant des mesures avec des modèles à l'aide de différentes manières. On peut citer l'utilisation d'un modèle cinématique propagé par des gyroscopes à intégration de vitesse à trois axes. Cependant, les taux mesurés par les gyroscopes dérivent dans le temps. Par conséquent, le vecteur d'état d'attitude est généralement complété par trois autres états pour déterminer cette dérive. Cela conduit à une approche complémentaire, dans laquelle les gyroscopes sont utilisés pour filtrer les mesures de corps bruyants et les mesures sont utilisées pour déterminer la dérive inhérente aux gyroscopes. Une autre méthode consiste à combiner le modèle cinématique avec un modèle dynamique pour la vitesse angulaire.

Dans la littérature, le terme observateur est remplacé par le terme d'algorithme d'estimation par lots, c'est-à-dire exécuté en temps réel, pour fournir de meilleures estimations en temps réel de l'algorithme de filtrage.

2.3.1 La notion d'observateur

Pour un système :

$$\begin{cases} \dot{x} = f(x, u) \\ y = h(x, u) \end{cases} \quad (2.19)$$

Les signaux y, u sont supposés connus jusqu'à l'instant courant t . La sortie y est mesurée par des capteurs, l'entrée u est mesurée et/ou connue en tant que signal de commande.

De manière très générale, un estimateur est un processus qui produit à tout instant t une estimation \hat{x} tendant vers $x(t)$ à partir des signaux connus $y(\tau), u(\tau)$, $0 \leq \tau \leq t$

Dans la suite on se restreindra à la notion d'observateur, à savoir un système dynamique :

$$\begin{cases} \dot{z} = \mathbf{F}(z, u, y) \\ \hat{x} = G(z) \end{cases} \quad (2.20)$$

L'observateur vérifié les deux propriétés suivantes :

si

$$y = h(\hat{x}, u) \quad (2.21)$$

dans (2.20), alors (\hat{x}, u) est solution de (2.19), c'est-à-dire :

$$G(z) \cdot \mathbf{F}(z, u, h(G(z), u)) = f(G(z), u) \quad (2.22)$$

$$\hat{x}(t) \rightarrow x(t) \quad \text{quand } t \rightarrow +\infty \quad (2.23)$$

La première condition dit que si l'estimation est exacte à un certain instant, alors elle l'est aussi à tous les instants suivants, la deuxième dit que l'estimation se rapproche asymptotiquement de la vraie valeur.

2.3.2 Observateurs des systèmes linéaires

Dans la littérature, il existe plusieurs types d'observateurs pour les systèmes linéaires : Observateur à entrée inconnue, observateur réduit de Luenberger, l'observateur de Luenberger, le filtre de Kalman. Mais l'une des solutions les plus simple et optimale au problème de l'estimation de l'état pour systèmes linéaires sont l'observateur de Luenberger dans le cadre déterministe, et par Kalman dans le cadre stochastique [Larbi, 2015], [Chekakta, 2016].

2.3.2.1 l'observateur de Luenberger

l'observateur de Luenberger repose essentiellement sur des techniques de placement de poles. On se place dans le cas deterministe, Luenberger propose l'observateur suivant :

$$\begin{cases} \dot{\hat{x}} = A\hat{x} + Bu + L(y - \hat{y}). \\ \hat{y} = C\hat{x}. \end{cases} \quad (2.24)$$

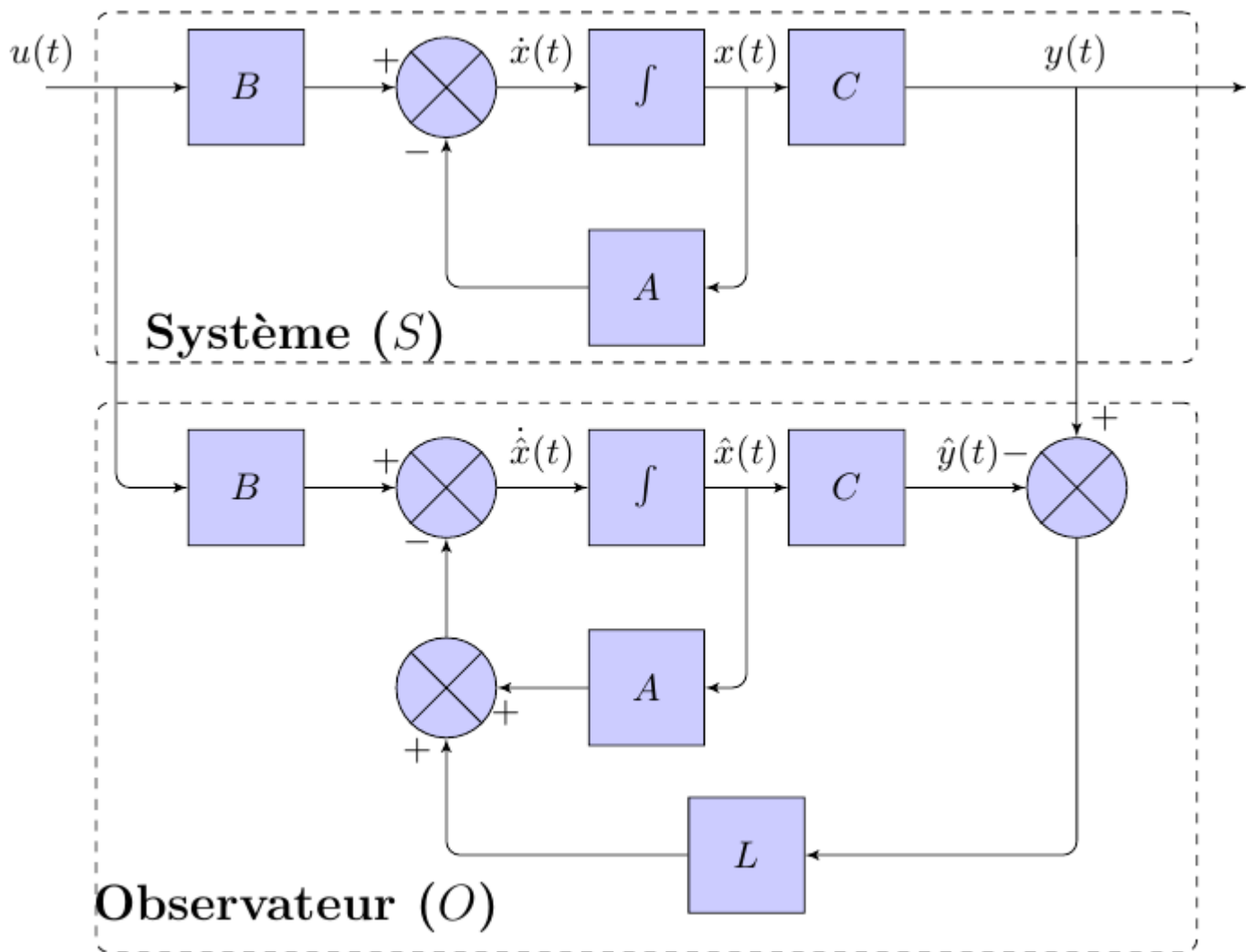


Figure 2.9 Schéma d'observateur Luenberger

Il suffit alors de choisir le gain L de l'observateur en utilisant une technique de placement de

poles, de telle sorte que la matrice (A-LC) soit Herwitz.

2.3.2.2 Le filtre de Kalman (KF)

Le filtre de Kalman est un filtre à réponse impulsionnelle infinie qui estime les états d'un système dynamique à partir d'une série de mesures incomplètes ou bruitées. Le filtre a été nommé d'après le mathématicien et informaticien américain d'origine hongroise Rudolf Kalman. IL est également connu sous le nom d'estimateur quadratique linéaire. Il s'agit d'un type d'observateur ou d'estimateur d'état qui est optimal dans le sens où il tente de minimiser une fonction de coût quadratique. Le filtre de Kalman se compose de six étapes, exécutées récursivement avec un temps d'échantillonnage entre les exécutions (Figure. 2.10).

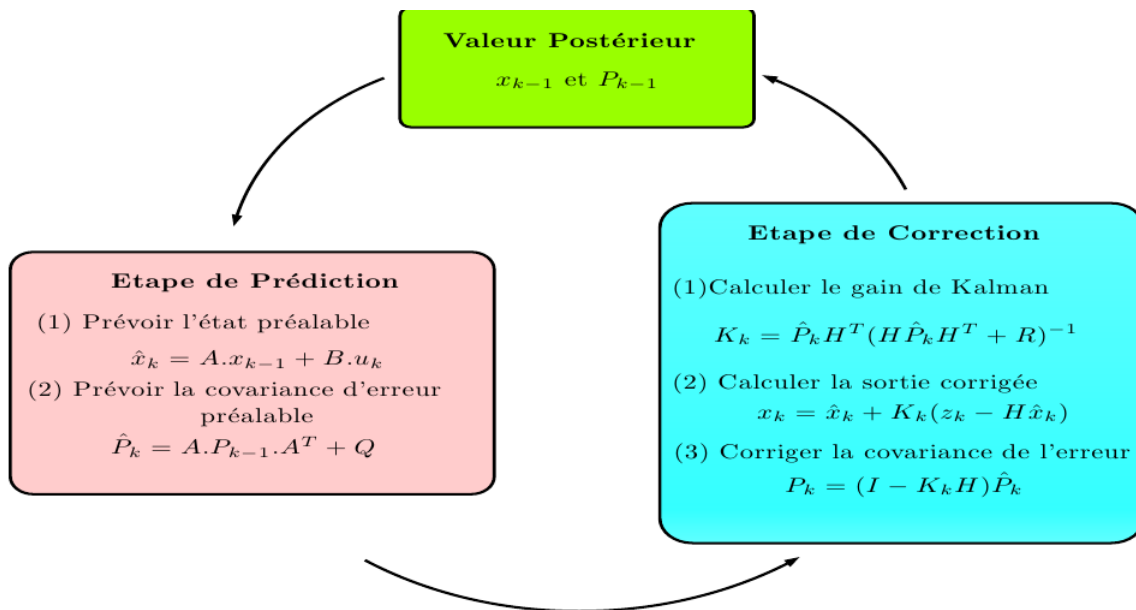


Figure 2.10 Formulation mathématique du filtre de Kalman

2.3.3 Observateurs des systèmes non linéaires

2.3.3.1 Observateurs a mode glissant

Dans tout les methodes des observateur linéaire le modèle dynamique du système étudié était supposé parfaitement connu. Le principe des observateurs à modes glissants consiste

à contraindre, à l'aide de fonctions discontinues, les dynamiques d'un système d'ordre n à converger vers une variété S de dimension $(n-p)$ dite surface de glissement (p étant la dimension du vecteur de mesure). L'attractivité de cette surface est assurée par des conditions appelées conditions de glissement. Si ces conditions sont vérifiées, le système converge vers la surface de glissement et y évolue selon une dynamique d'ordre $(n-p)$. [Larbi, 2015]

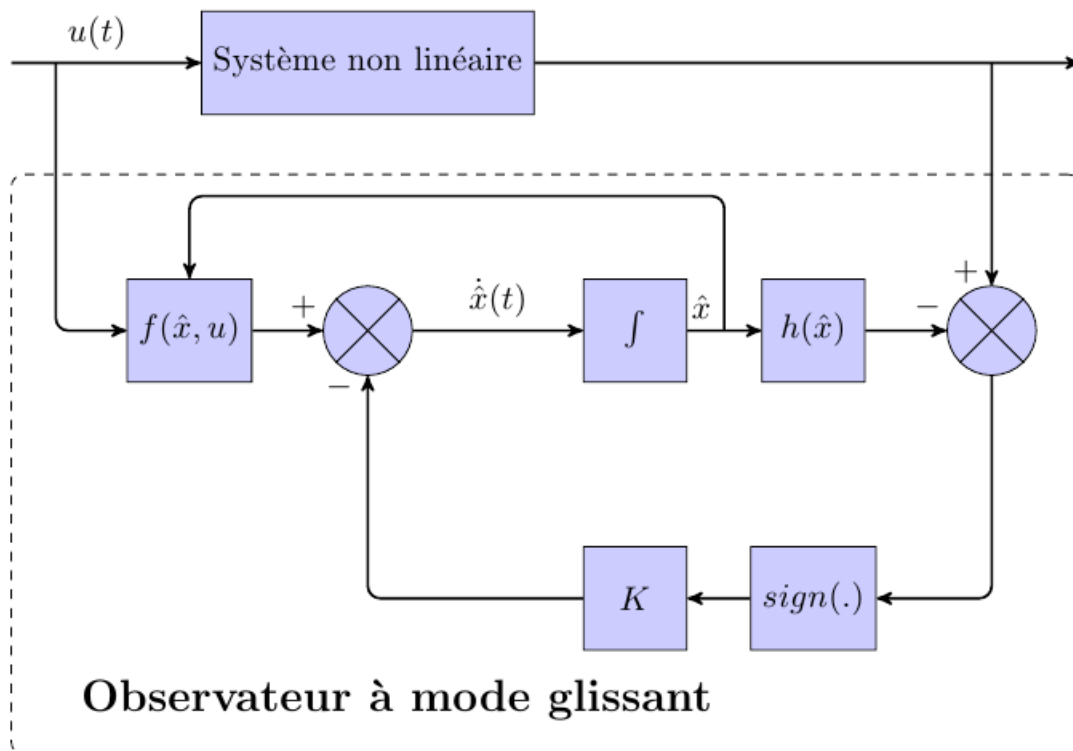


Figure 2.11 Schéma d'observateur mode glissant

2.3.3.2 Le filtre de Kalman étendu (EKF)

Le filtre Kalman étendu est l'une des techniques d'observation les plus populaires et largement étudié dans le domaine de l'observation d'état des systèmes non linéaires dynamiques. Ce filtre étendu consiste à utiliser les équations du filtre de Kalman standard au modèle non linéaire linéarisé par la formule de Taylor au premier ordre. Ce filtre étendu a été appliqué avec succès sur divers types de processus non linéaires. Malheureusement, les preuves de stabilité et de convergence établie dans le cas des systèmes linéaires, ne peut être générale dans le cas des systèmes non linéaires.

2.3.3.3 Filtre particulière

Le filtre particulière fonctionne en combinant les caractéristiques de basse fréquence souhaitables de l'accéléromètre avec les caractéristiques de haute fréquence souhaitables du gyroscope. Donc ils sont des techniques sophistiquées d'estimation de modèles fondées sur la simulation. S'ils sont conçus correctement, ils peuvent être rendus plus précis que les filtres de Kalman. Quan [2017] **Technique d'estimation d'angle θ , ϕ a l'aide d'un filtre particulière**

$$\begin{cases} \phi_{ACC}^m = \phi + n_\phi \\ \theta_{ACC}^m = \theta + n_\theta \end{cases} \quad (2.25)$$

Considérant que l'angle de tangage estimé en intégrant la vitesse angulaire a un peu de bruit mais un grande déphasage, l'intégration est modélisée comme :

$$\begin{cases} \frac{GYR_x^m(s)}{s} = \phi(s) + c\frac{1}{s} \\ \frac{GYR_y^m(s)}{s} = \theta(s) + c\frac{1}{s} \end{cases} \quad (2.26)$$

où $\frac{GYR_x^m(s)}{s}$ est la transformée de Laplace de l'intégration de la vitesse angulaire $GYR_x^m(s)$, $c\frac{1}{s}$ représente la transformée de Laplace de la dérive constante, et $GYR_x^m(s)$ est la mesure gyroscopique. Par conséquent la forme standard d'un filtre linéaire complémentaire est exprimée par :

$$\begin{cases} \phi_{Estimé} = \frac{1}{\tau.s+1} \phi_{ACC}^m + \frac{\tau}{\tau.s+1} \left(\frac{GYR_x^m(s)}{s} \right) \\ \theta_{Estimé} = \frac{1}{\tau.s+1} \theta_{ACC}^m + \frac{\tau}{\tau.s+1} \left(\frac{GYR_y^m(s)}{s} \right) \end{cases} \quad (2.27)$$

où $\frac{1}{\tau.s+1}$ est la fonction de transfert d'un filtre passe-bas, et $\frac{\tau}{\tau.s+1}$ est la fonction de transfert d'un filtre passe-haut.

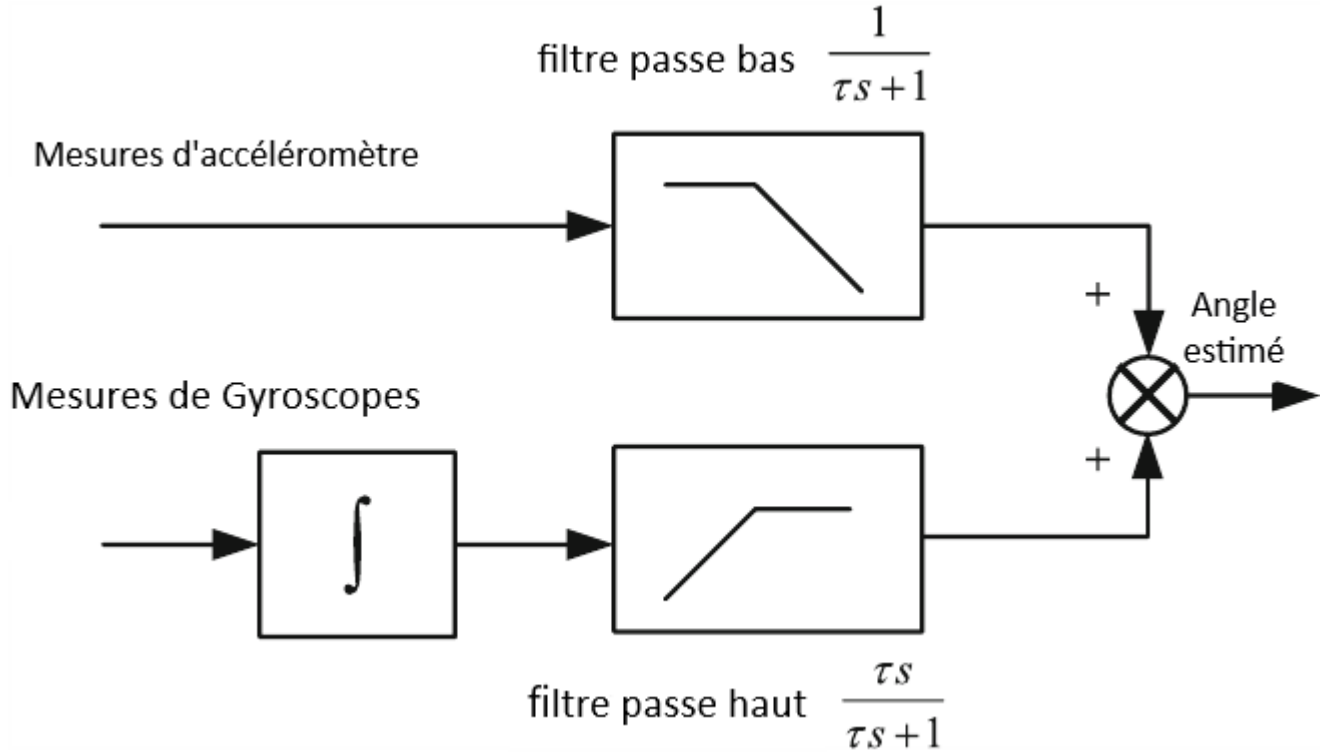


Figure 2.12 Schéma d'estimation d'attitude a l'aide d'un filtre linéaire complémentaire

Estimation d'angle ψ l'angle de lacet est mesuré à la fois par une fusion des données d'GPS et un magnomètre désignés par ψ_{GPS} et ψ_{MAG} , respectivement. Une méthode simple pour obtenir les données consiste à additionner la mesure pondérée des deux capteurs :

$$\{\psi_m = (1 - \alpha) + \alpha \cdot \psi_{MAG} \quad (2.28)$$

Où α est un facteur de pondération. Dans ce cas la valeur estimée d'angle ψ est donnée par :

$$\psi_{Estimé}(k) = \frac{\tau}{\tau \cdot s + T_s} (\psi_{Estimé}(k-1) + T_s \cdot GYR_z^m(k)) + \frac{T_s}{\tau \cdot s + T_s} \cdot \psi_m(k) \quad (2.29)$$

Dans cette approche nous avons utilisé les mesures d'un gyroscope et un l'accéléromètre pour estimer l'attitude d'un qudrirator. Notre approche suit le filtre complémentaire linéaire de Quan [2017]. Nous supposons que ϕ et θ sont petits pour assurer la linéarité de notre système.

En d'autres termes

$$\begin{bmatrix} \dot{\phi} \\ \dot{\theta} \\ \dot{\psi} \end{bmatrix} \approx \begin{bmatrix} p \\ q \\ r \end{bmatrix} \quad (2.30)$$

Ce qui est contraire des filtres complémentaires non linéaires malgré que ils bénéficient de l'avantage des caractéristiques complémentaires des accéléromètres et des gyroscopes. La seule différence est que les filtres complémentaires non linéaires sont basés sur une relation non linéaire entre la vitesse angulaire et l'angle de rotation le filtre passe bas, et le signal basse fréquence c/s peut être éliminé après avoir passé le filtre passe haut. En conséquence, les angles convergent avec une erreur nulle.

2.4 Conclusion

Dans cette section, nous avons élaboré un modèle dynamique complet du quadrotor. Ce modèle montre une nature couplée, complexe, non linéaire, multi variable et sous actionné. Ce qui rend son contrôle relativement difficile. D'autre part, nous avons également donné une idée globale de l'estimation des attitudes en introduisant la notion d'observateurs.

Le chapitre suivant sera consacré à la simulation de notre modèle en utilisant MATLAB ainsi d'appliquer les commandes. Puis en utilisant la simulation en temps réel pour estimer l'attitude d'un drone de type quadrirotor par la plateforme ROS.

CHAPITRE 3 Test et résultats de simulation

3.1 Introduction

Dans ce chapitre, nous avons testé le modèle mathématique et la commande PID développé du quadrotor. La dynamique de ces derniers est présentée par des équations différentielles dérivées de l'utilisation du formalisme de Newton Euler à l'aide de logiciel Matlab. Puis, nous utilisons l'environnement «ROS robot operating systems» pour récupérer les mesures et envoyer les commandes du drone, afin d'intégrer les techniques d'estimation que nous avons proposées pour améliorer notre commande et de maîtriser l'environnement ROS.

3.2 Simulation d'une commande PID d'un seul quadrotor

A base des équations de dynamique du quadrotor mentionnées dans le chapitre précédent, un contrôleur PID linéarisé est utilisé pour contrôler le quadrotor. La linéarisation est effectuée autour du point d'équilibre qui est le point stationnaire. La loi de commande est encore simplifiée en utilisant l'hypothèse du petit angle de :

$$\begin{cases} \sin(\alpha) \approx (\alpha) \\ \cos(\alpha) \approx 1 \end{cases} \quad (3.1)$$

Les angles de roulis et de tangage souhaités sont calculés en fonction de la position souhaitée dans le plan x-y en utilisant les équations :

$$\begin{cases} m(v_x \cdot \cos(\psi) + v_y \cdot \sin(\psi)) = \sin(\theta) \cdot U_1 \\ m(v_x \cdot \sin(\psi) - v_y \cdot \cos(\psi)) = \sin(\phi) \cos(\theta) \cdot U_1 \\ m(v_z + g) = \cos(\phi) \cdot \cos(\theta) \cdot U_1 \end{cases} \quad (3.2)$$

La commande reçoit la position souhaitée du quadrotor dans le plan x-y et calcul les angles comme suit :

$$\phi_d = \arctan\left(\frac{v_x \cdot \sin(\psi_d) - v_y \cdot \cos(\psi_d)}{v_z + g}\right) \quad (3.3)$$

$$\theta_d = \arctan\left(\frac{v_x \cdot \cos(\psi_d) + v_y \cdot \sin(\psi_d)}{\sqrt{(v_x \cdot \sin(\psi_d) - v_y \cdot \cos(\psi_d))^2 + (v_z + g)^2}}\right) \quad (3.4)$$

3.2.1 Résultat de simulation

Un contrôleur d'attitude et d'altitude est introduit pour chaque angle d'orientation, donc une commande d'attitude fixe est testée. Nous choisissons la référence fixe $\phi_d = 20^\circ$, $\theta_d = 10^\circ$, $\psi_d = 10^\circ$ et $Z_d = 4$

Nous pouvons voir les résultats de la simulation dans les figures : (Figure.3.1), (Figure.3.2).

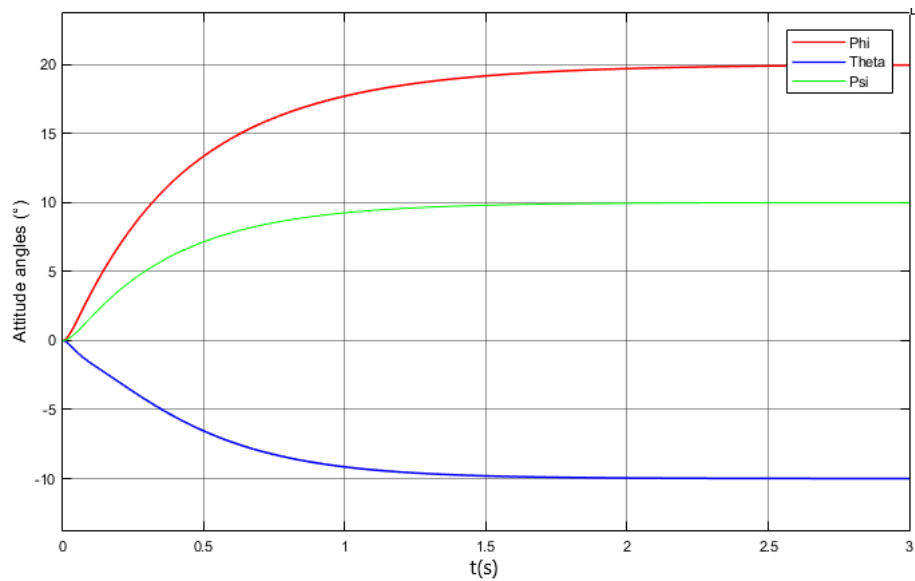


Figure 3.1 Résultat de simulation de vecteur d'orientation

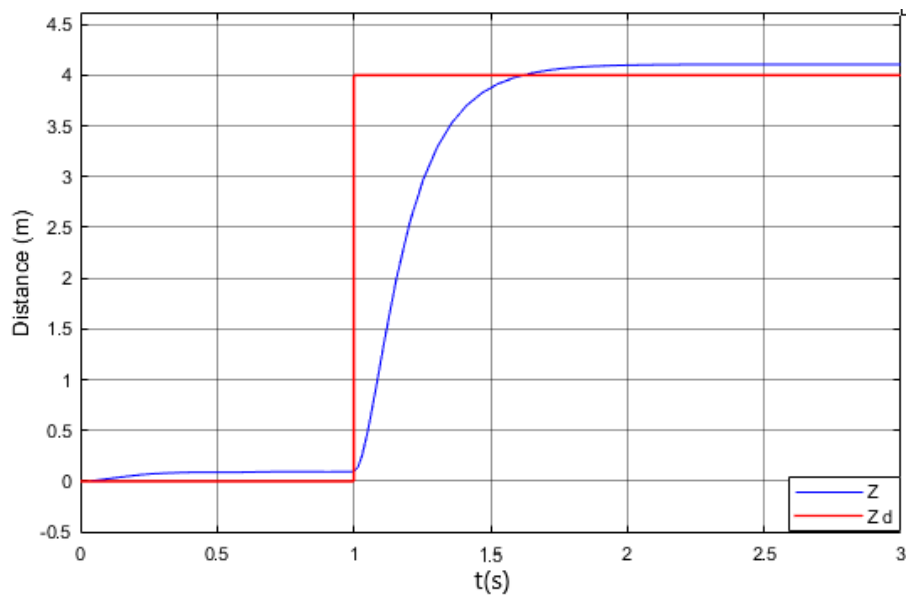
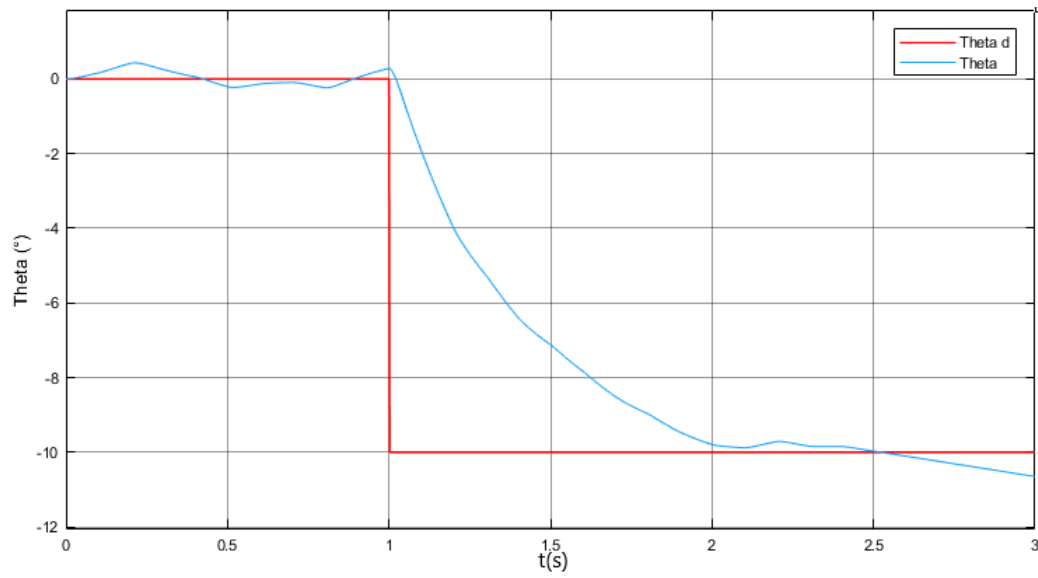
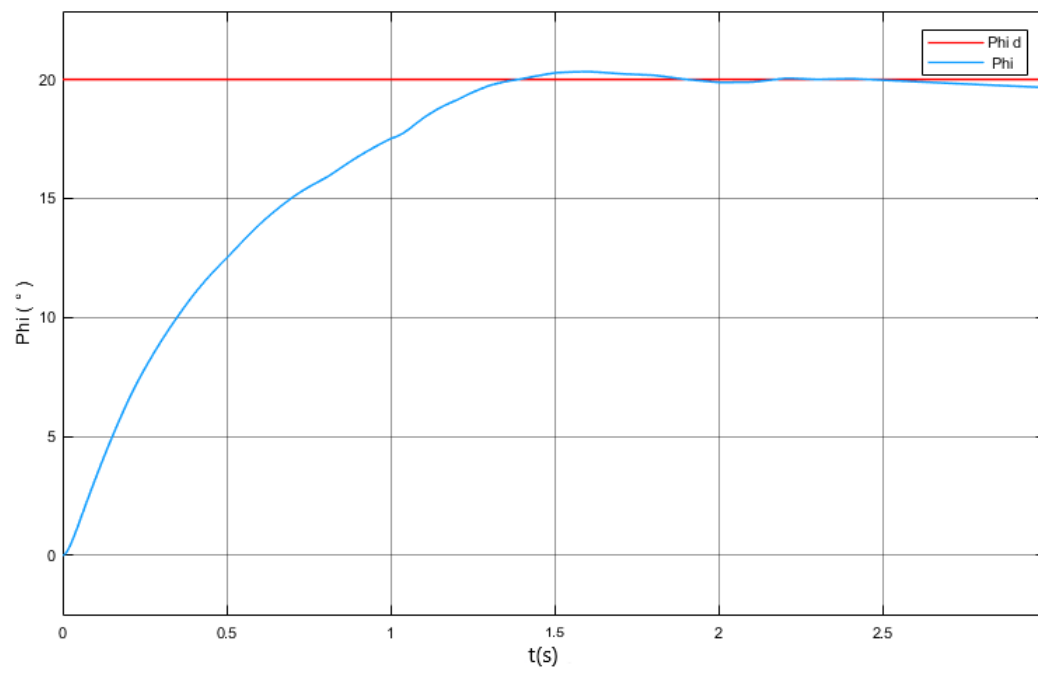


Figure 3.2 Résultat de simulation d'altitude avec PID

Figure 3.3 Angle θ avec présence de bruitFigure 3.4 Angle ϕ avec présence de bruit

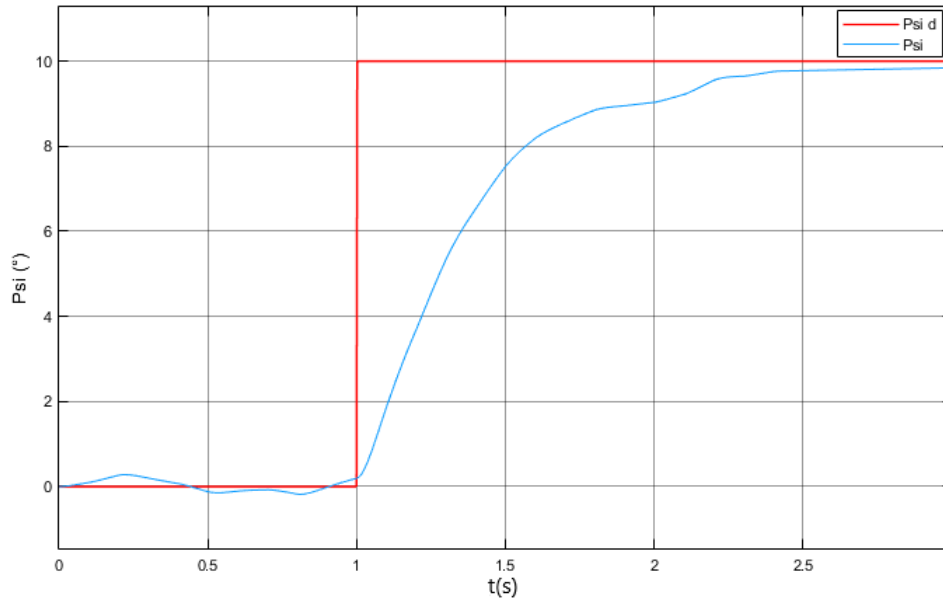


Figure 3.5 Angle ψ avec présence de bruit

Un contrôleur pour tout les états du système (Vecteur d'attitude et vecteur de position).est aussi introduit :

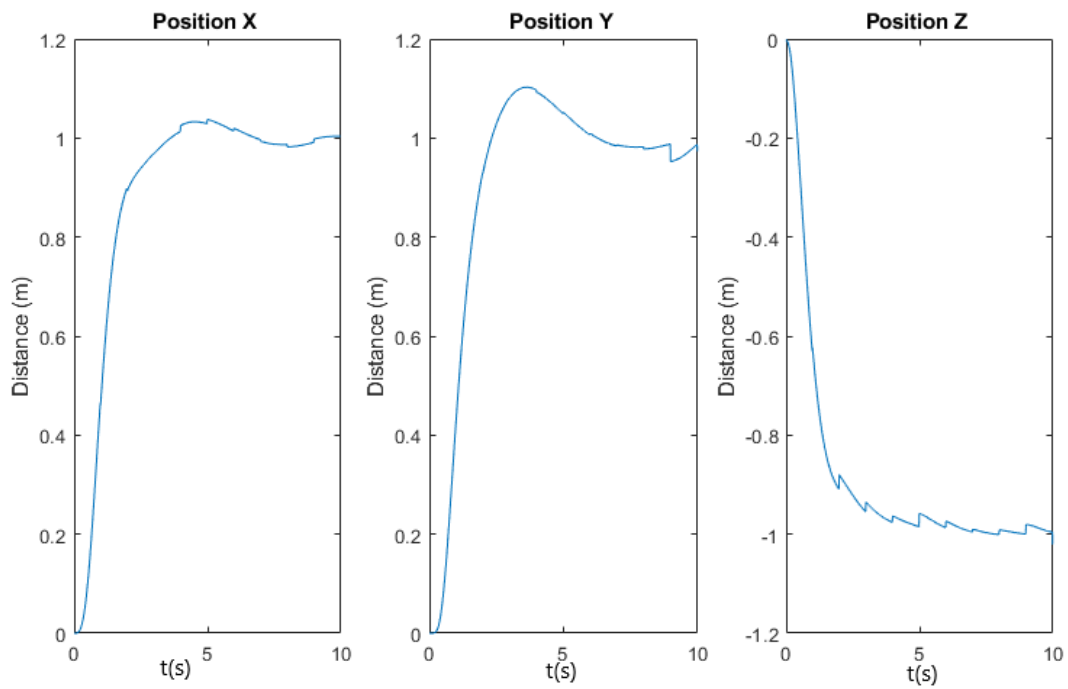


Figure 3.6 vecteur de position

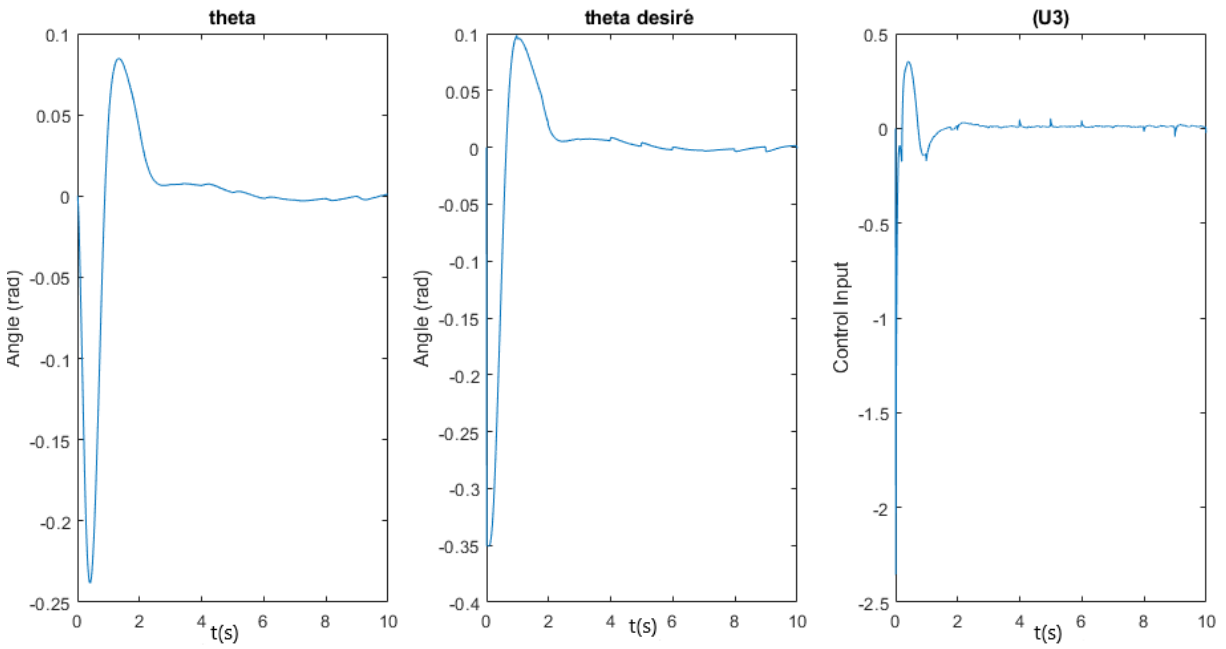


Figure 3.7 Résultat de simulation d'angle θ et la commande U3

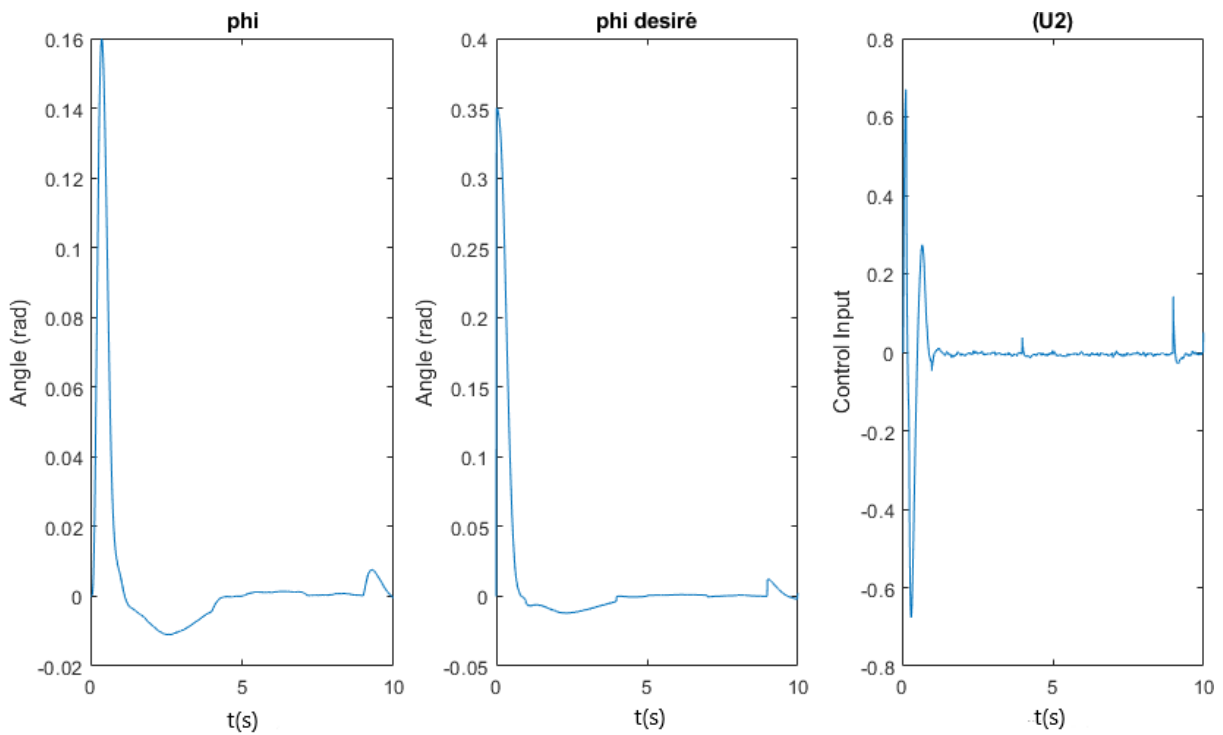


Figure 3.8 Résultat de simulation d'angle ϕ et la commande U2

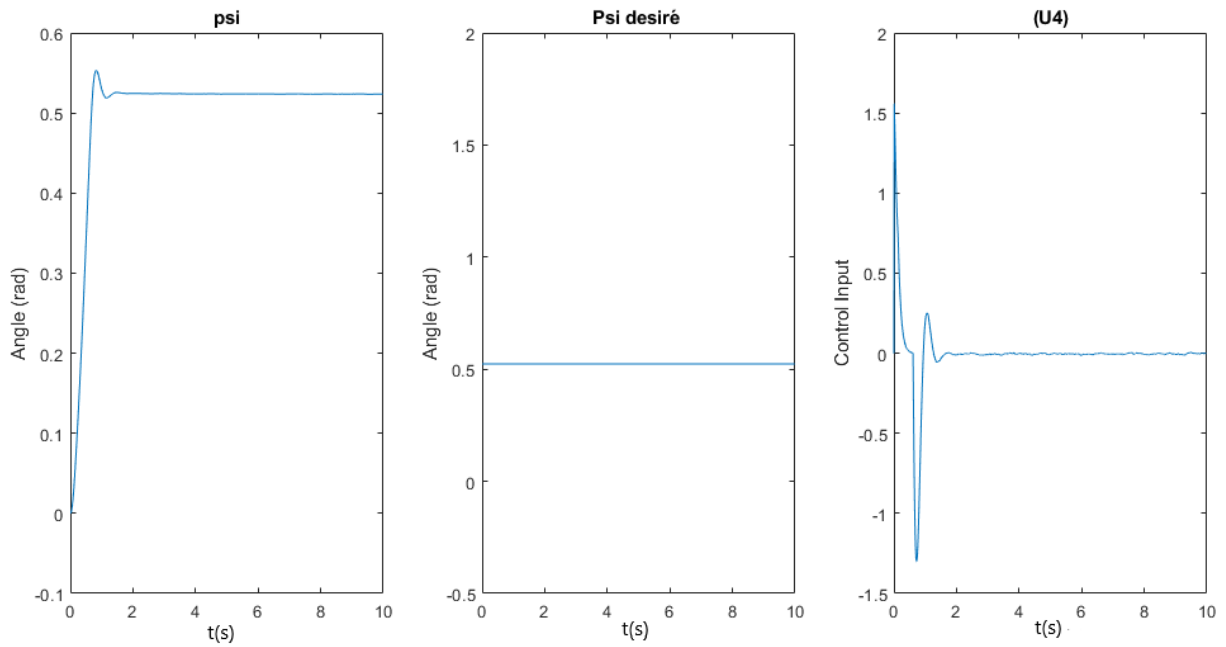


Figure 3.9 Résultat de simulation d'angle ψ et la commande U_4

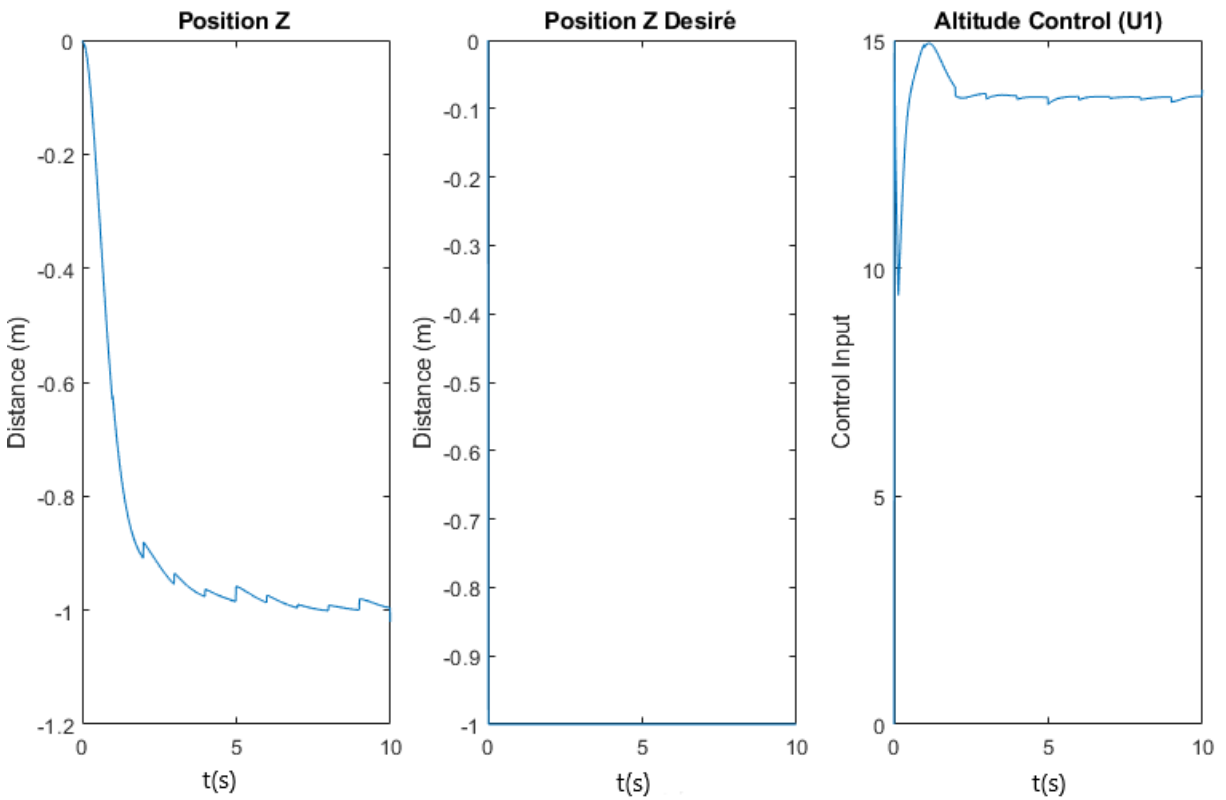


Figure 3.10 Résultat de simulation d'altitude Z et la commande U_1

3.2.2 Interprétation des résultats

On voit que notre commande arrive à stabiliser le quadrotor à la hauteur et angles désirées. Donc la commande PID est très efficace dans le cas des systèmes linéaires à paramètres constants ce qui correspond au vol stationnaire pour le quadrotor. Nous pouvons alors dans ce cas réaliser un vol autonome avec succès mais dans le cas des systèmes non linéaires ayant des variations paramétriques (Quelques perturbations ont été introduites) ces lois peuvent être insuffisantes (côté robustesse) car le régulateur perd son fonctionnement naturel (Figure. ??) à cause des bruits donc il y a une apparition d'une erreur statique en cas de mauvais choix des gains pour des perturbations assez grand.

3.3 Estimation d'attitude d'un quadrotor a l'aide d'observateur Luenberger

Comme nous l'avons mentionné, l'observateur de Luenberger est un observateur qui s'applique aux systèmes de dynamique suivants :

$$\begin{cases} \dot{x} = Ax + Bu \\ y = Cx + Du \end{cases} \quad (3.5)$$

Cela crée le besoin d'utiliser la linéarisation jacobienne afin d'analyser notre système non linéaire qui est approché par un système linéaire autour du point d'équilibre \hat{x} :

$$\begin{cases} A = \frac{\partial f}{\partial x} \Big|_{x=\hat{x}, u=\hat{u}} \\ B = \frac{\partial f}{\partial u} \Big|_{x=\hat{x}, u=\hat{u}} \end{cases} \quad (3.6)$$

Pour le cas de quadrirotor le point d'équilibre :

$$\hat{x} = [x, y, z, 0, 0, 0, 0, 0, 0, 0, 0, 0]^T \quad (3.7)$$

$$\hat{u} = [mg, 0, 0, 0, 0]^T \quad (3.8)$$

À partir des équations (3.6), (3.7), (3.8) les matrices A et B, C peuvent être déduits. Nous avons effectué des tests avec le modèle linéarisés ce modèle est commandable, observable et

detectable

Sachant que le quadrirotor est un système sous-actionné, c'est-à-dire, les six degrés de liberté dans l'espace sont contrôlés seulement avec quatre moteurs. Par conséquent, la commande de notre drone doit être établie pour un sous-ensemble de quatre de ses degrés de liberté. Ainsi nous allons commander et observé 4 paramètres : la position en Z , les angles d'attitude suivant les axes X , Y , Z .

3.3.1 Résultat de simulation

À partir du modèle de contrôle que nous avons développé dans la section précédente, nous ajoutons un observateur Luenberger à notre système.

La simulation commence par l'introduction des entrées souhaitées, puis l'ensemble des données est dégradé avec le retard et le bruit blanc. Il est ensuite utilisé dans le bloc d'observateur qui générera les signaux d'état estimés à sa sortie.

Nous pouvons voir les résultats de la simulation dans les figures : (Figure.3.11), (Figure.3.12), (Figure.3.13).

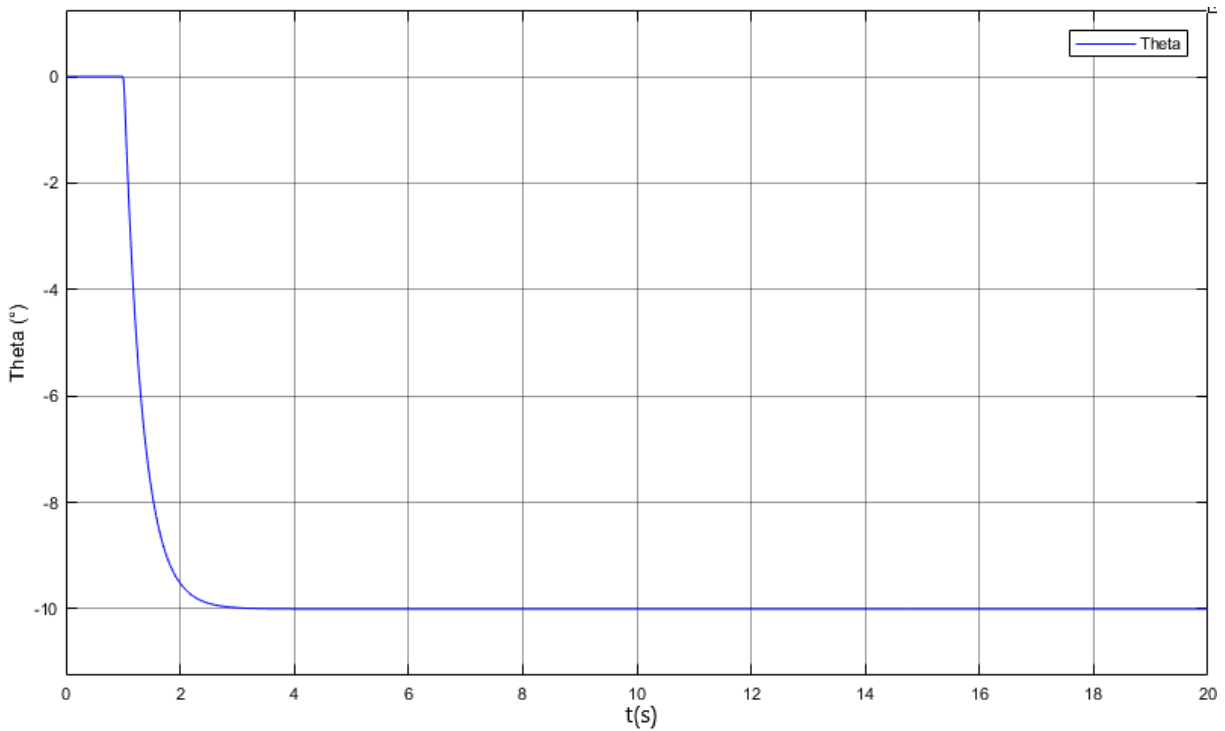
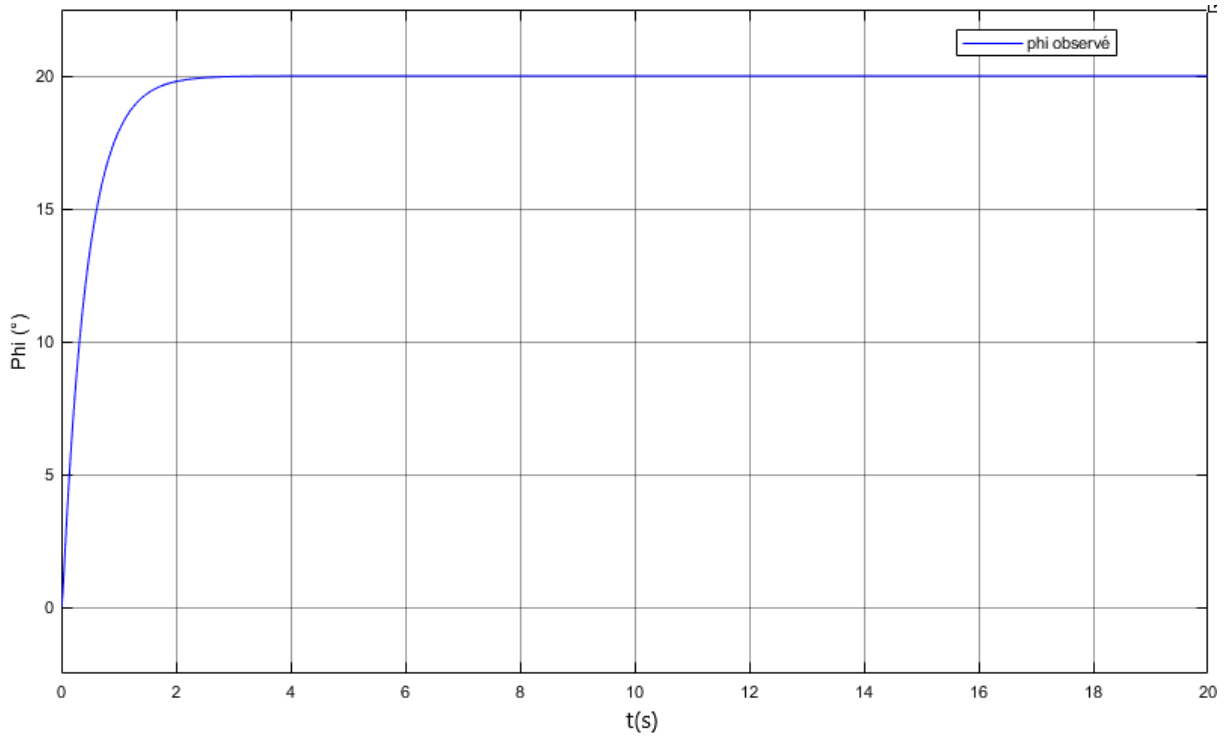
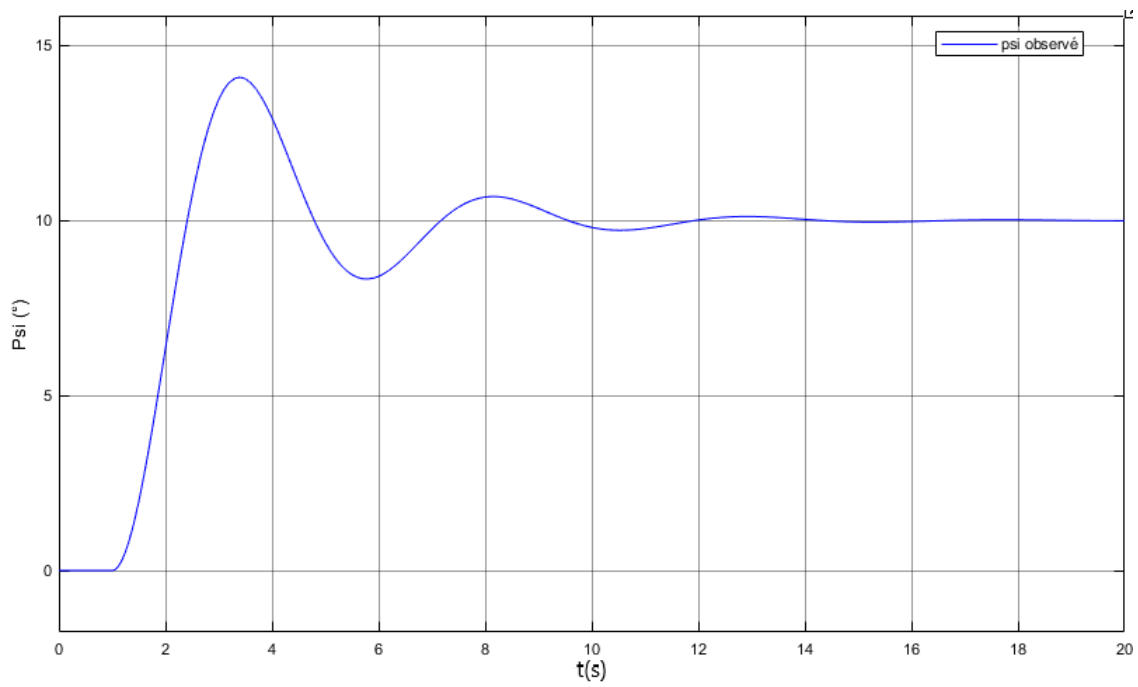


Figure 3.11 Angle θ observé

Figure 3.12 Angle ϕ observéFigure 3.13 Angle ψ observé

La (Figure.3.14) exprime l'erreur entre les états des sorties de notre système et d'observateur

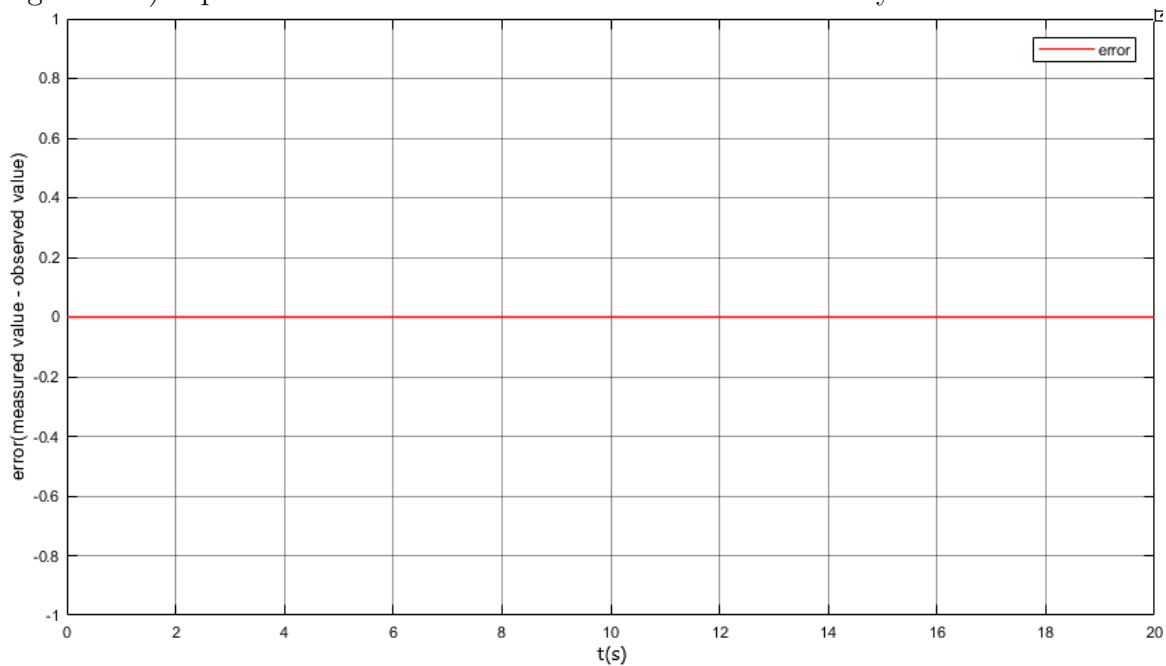


Figure 3.14 erreur entre la valeur mesuré et observé

Ensuite en introduisant des perturbations a l'entrée de système :

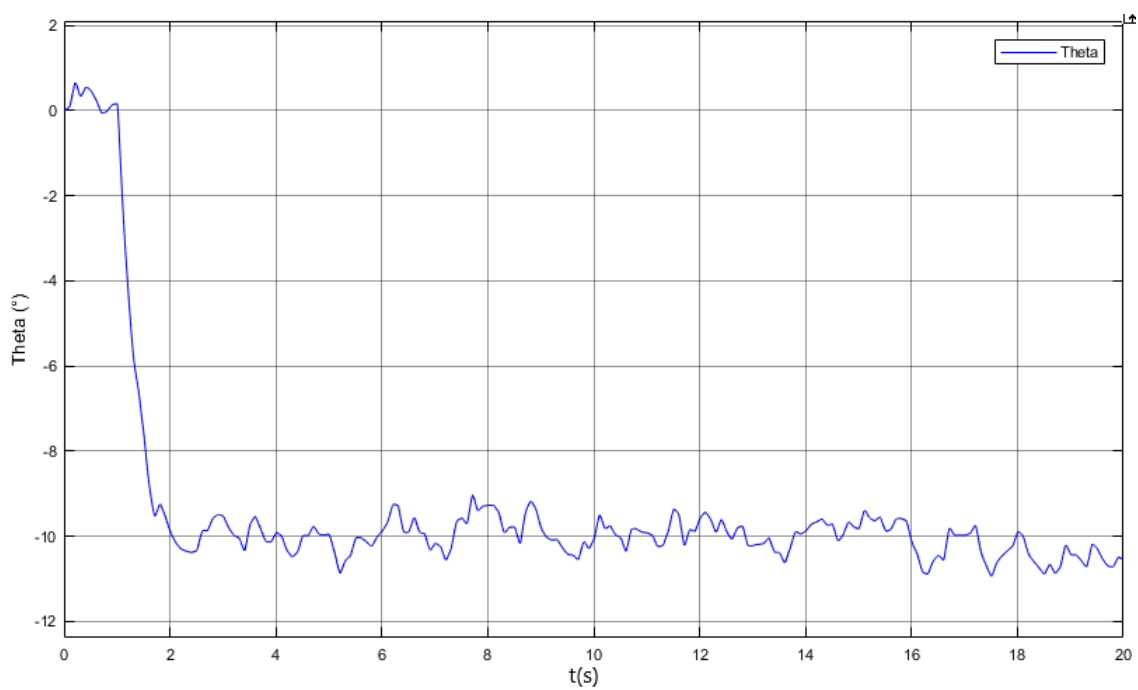


Figure 3.15 Angle θ observé avec présence de bruit

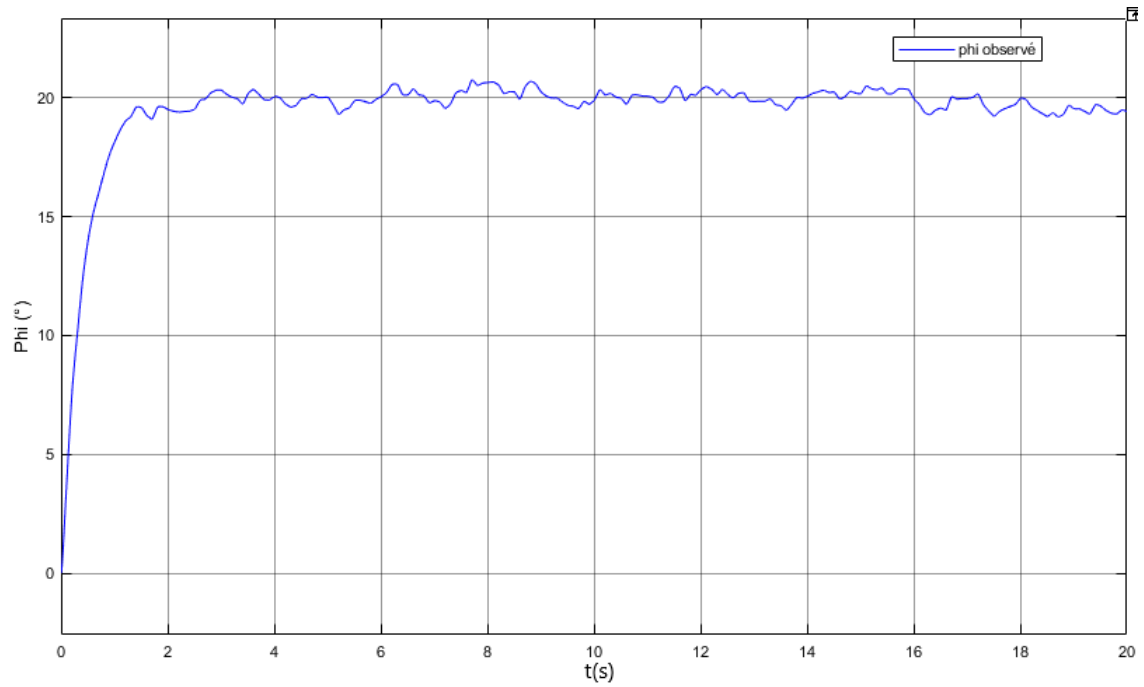


Figure 3.16 Angle ϕ observé avec présence de bruit

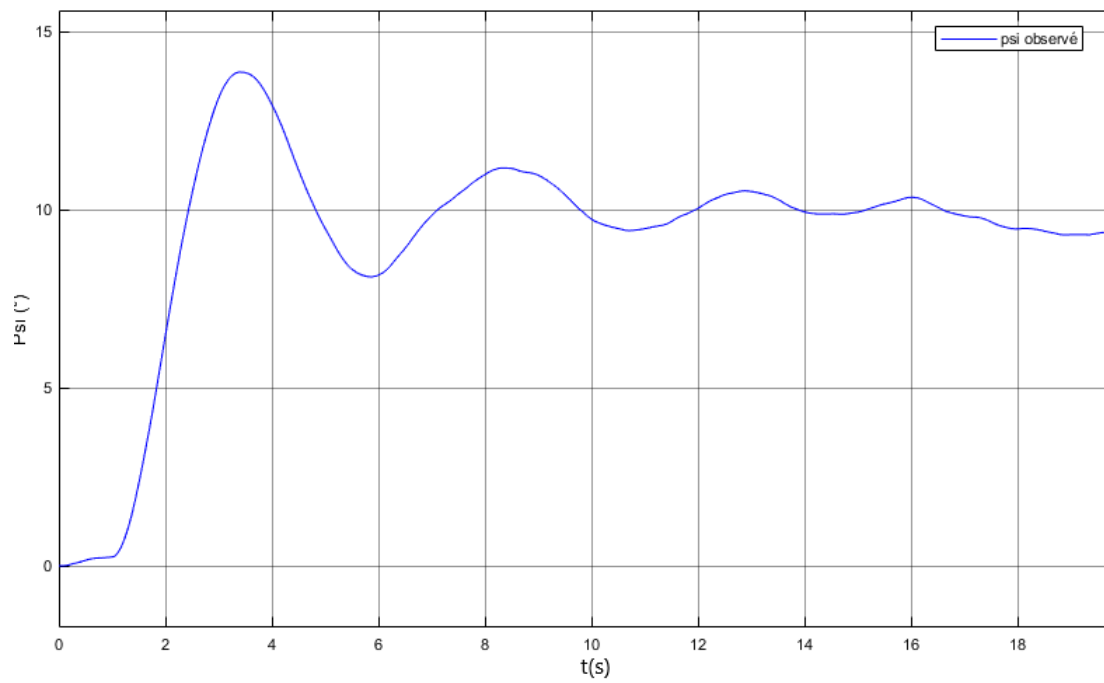


Figure 3.17 Angle ψ observé avec présence de bruit

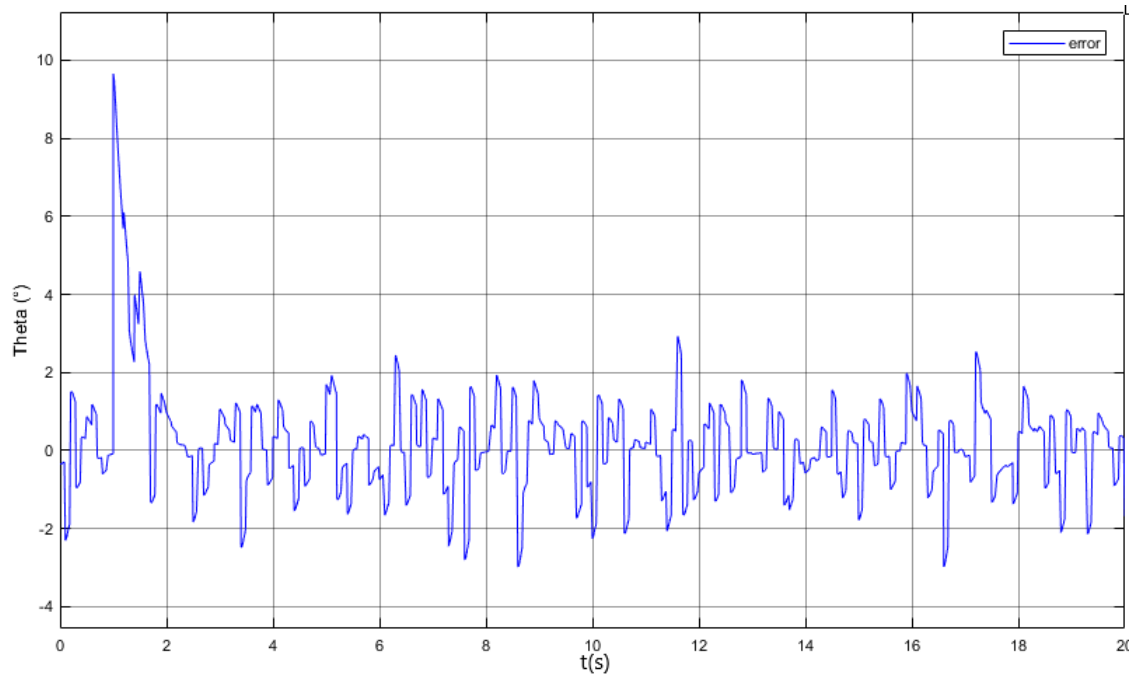


Figure 3.18 erreur entre la valeur observé d'angle θ et l'entré avec présence de bruit

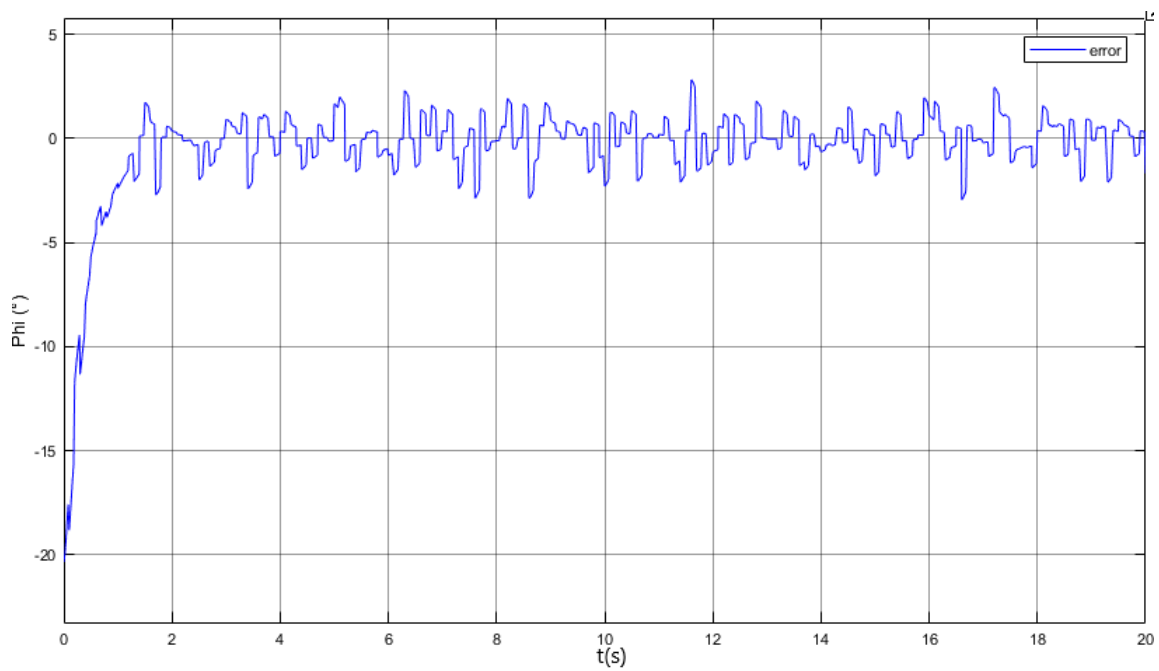


Figure 3.19 erreur entre la valeur observé d'angle ϕ et l'entré avec présence de bruit

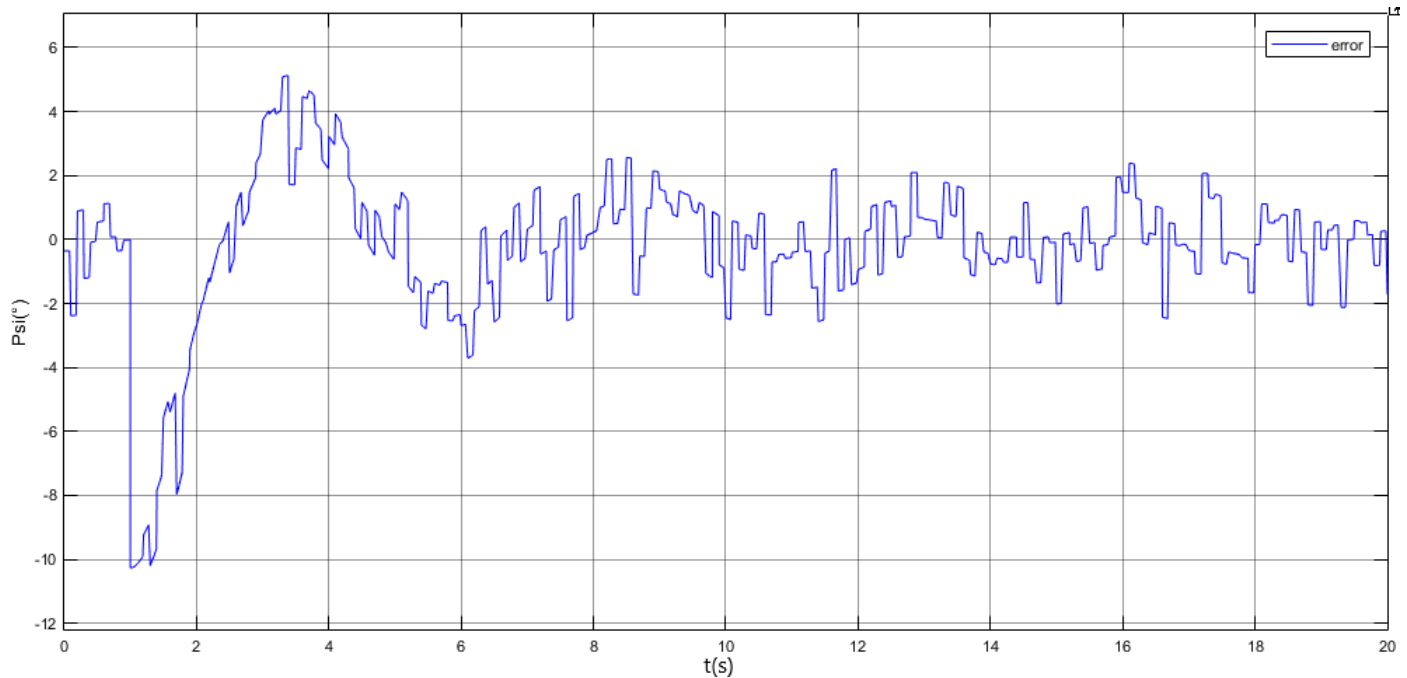


Figure 3.20 erreur entre la valeur observé d'angle ψ et l'entré avec présence de bruit

3.3.2 Interprétation des résultats

Nous avons présenté l'observateur de Luenberger à entrée connue et dégradé. Cette méthode est appliquée pour la construction d'un observateur d'ordre complet dans le cas de systèmes linéaires. Il est basé exclusivement sur le gain L qui joue le rôle de régulateur de vitesse de convergence et un élément de base dans la synthèse de cet observateur pour garantir la convergence exacte des états observés.

L'observateur a donné presque les mêmes résultats des états, les erreurs d'estimations sont nulles comme il est montré sur la (Figure.3.14). Nous avons constaté que nos résultats sont fiable par rapport aux résultats mesurés. Mais en raison de la complexité de la mise en œuvre de cet observateur sur des systèmes avec un nombre d'états assez élevé, l'observateur de Luenberger est limité, ce qui a créé le besoin de synthèse d'un filtre kalman et un filtre kalman étendu afin d'assurer plus de robustesse contre les perturbations, un filtrage ou une correction des données.

3.4 Estimation d'attitude d'un quadrotor a base d'un filtre de kalman

Le filtre de Kalman est un observateur/prédicteur optimal il permettra d'estimer l'état du système linéaire et son imprécision associée. Pour ce faire, le filtre de Kalman intègre de

manière itérative des observations au cours du temps.

Le filtre kalman s'applique en utilisant les équations du modèle d'évolution du système et les équations de mesure :

$$\begin{cases} \dot{x} = Ax + Bu + Mw \\ z_k = Cx_k + Du + Nv \end{cases} \quad (3.9)$$

w et v exprimer respectivement le bruit (Gaussien) du système et de la mesure avec des covariances MM^T et NN^T . L'observateur est donné par :

$$\begin{cases} \dot{\hat{x}} = A\hat{x} + Bu + (PC^T(NN^T)^{-1}(y - (C\hat{x} + Du))). \\ \dot{P} = AP + PA^T + MM^T - PC^T(NN^T)^{-1}CP. \end{cases} \quad (3.10)$$

Notons que le filtre kalman a la même structure qu'un observateur lunberger sauf que la matrice de gain est variable $L = PC^T(NN^T)^{-1}$

3.4.1 Filtre de Kalman étendu (Extended Kalman Filter, EKF)

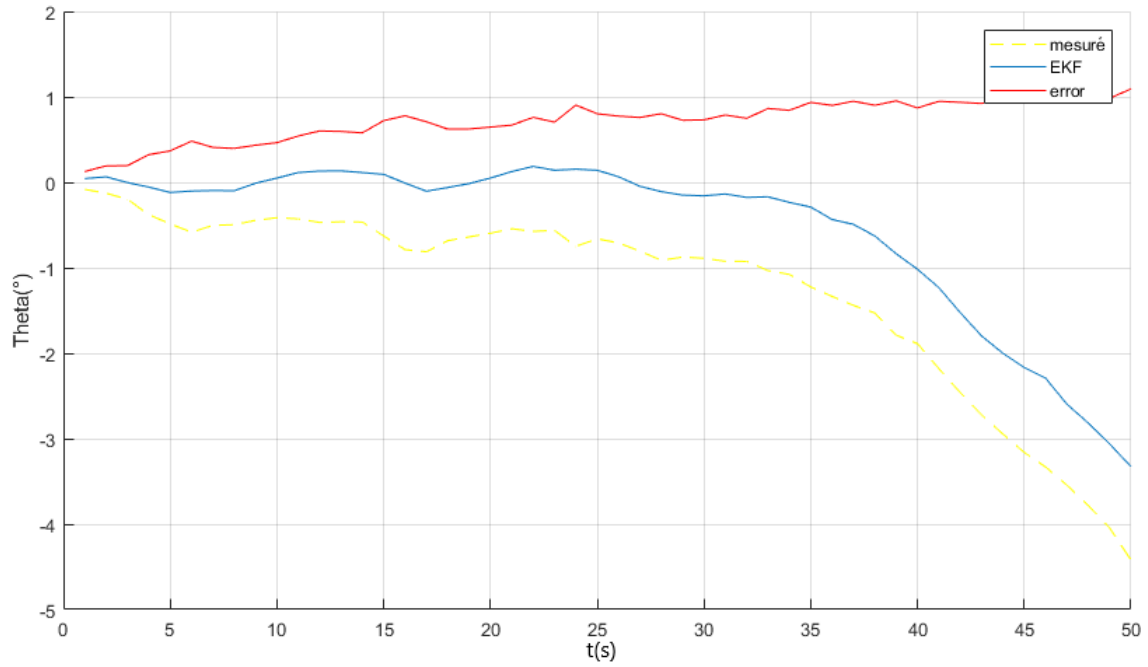
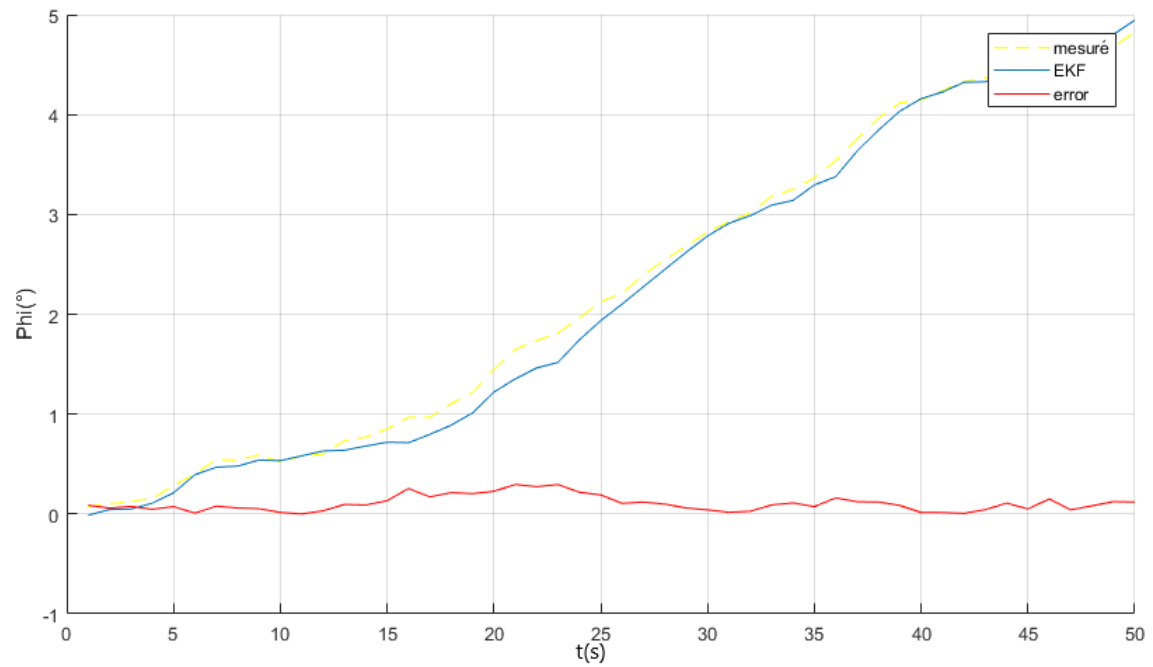
C'est le cas général d'un filtre kalman qui s'applique aux systèmes non linéaires par analogie avec le système (3.10) le filtre kalman étendu peut être défini par le système des equations suivant :

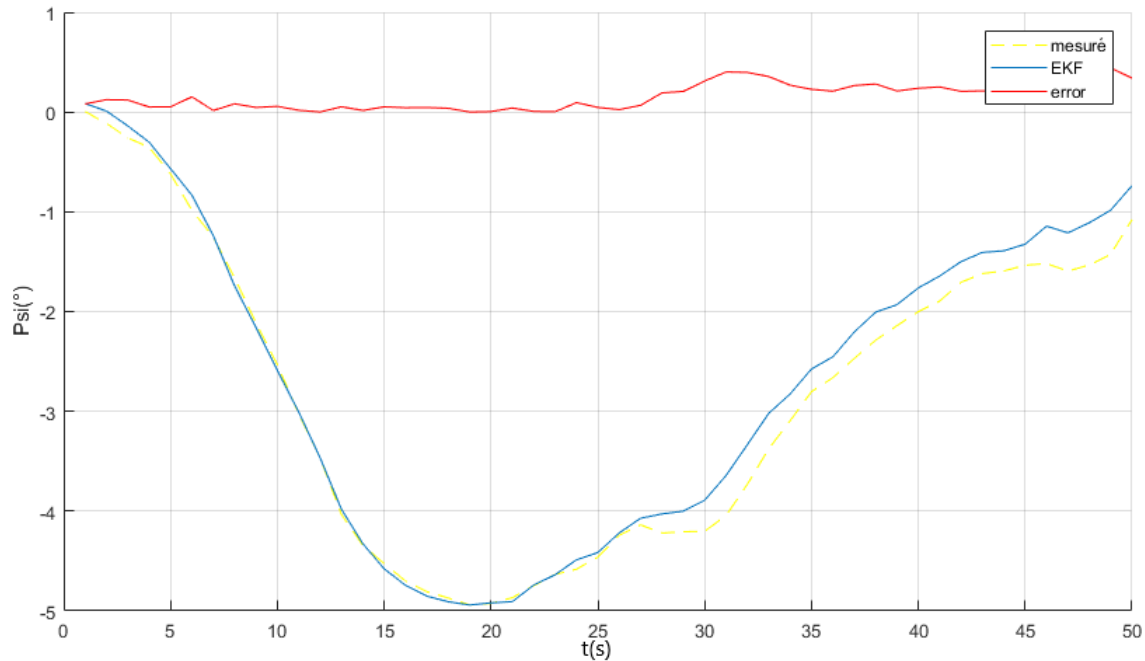
$$\begin{cases} \dot{\hat{x}} = f(\hat{x}, u) + (PC^T(NN^T)^{-1}(y - h(\hat{x}, u))). \\ \dot{P} = AP + PA^T + MM^T - PC^T(NN^T)^{-1}CP. \end{cases} \quad (3.11)$$

Contrairement au filtre kalman dans un filtre kalman étendu, les matrices A et C ne sont pas constante elles seront calculées à chaque fois.

3.4.2 Résultat de simulation

Nous étudions dans cette section l'implémentation du filtre kalman étendu dans un modèle d'attitude simplifié. Nous pouvons voir les résultats de la simulation dans les figures : (Figure.3.21), (Figure.3.22), (Figure.3.23).

Figure 3.21 Angle θ observéFigure 3.22 Angle ϕ observé

Figure 3.23 Angle ψ observé

3.4.3 Interprétation des résultats

Nous avons constaté que le filtre de Kalman est un estimateur d'état qui fonctionne sur une base de prédiction et de correction. Il converge exponentiellement mieux que la méthode précédente et on s'aperçoit que malgré le bruit important, le filtre de Kalman a réussi à estimer assez correctement et donne une meilleure estimation à partir de données bruyantes. Mais ce dernier peut être appliqué sur des systèmes linéaires observables ce qui crée le besoin d'utiliser un EKF qui linéarise le modèle autour d'une trajectoire estimée \hat{x} .

Tableau 3.1 Comparaison entre les observateurs

Observateur	système	robustess	précis	Temp reel	Nombre de paramètres
Luenberger	linéaire	non robuste	précis	non	grand
KF	linéaire	robuste	précis	oui	moyenne
EKF	non linéaire	robuste	très précis	oui	moyenne

3.5 Estimation d'attitude d'un quadrotor a base d'un filtre de kalman sous ROS

La centrale inertielle (IMU) est utilisée pour fournir une information sur l'orientation du drone sur les trois axes, alors la sortie du filtre de Kalman sera un vecteur composé des trois angles $(\hat{\phi}, \hat{\theta}, \hat{\psi})$ suivant les trois axes x, y et z respectivement et les biais de mesure sur l'axe x, y.

Configuration classique d'IMU

1. 3 axes Accéléromètres pour le calcul d'accélération
2. 3 axes Gyromètres pour calcule les vitesses angulaires
3. 3 axes Magnétomètres pour calculer la hauteur
4. Baromètres pour les mesures de pression

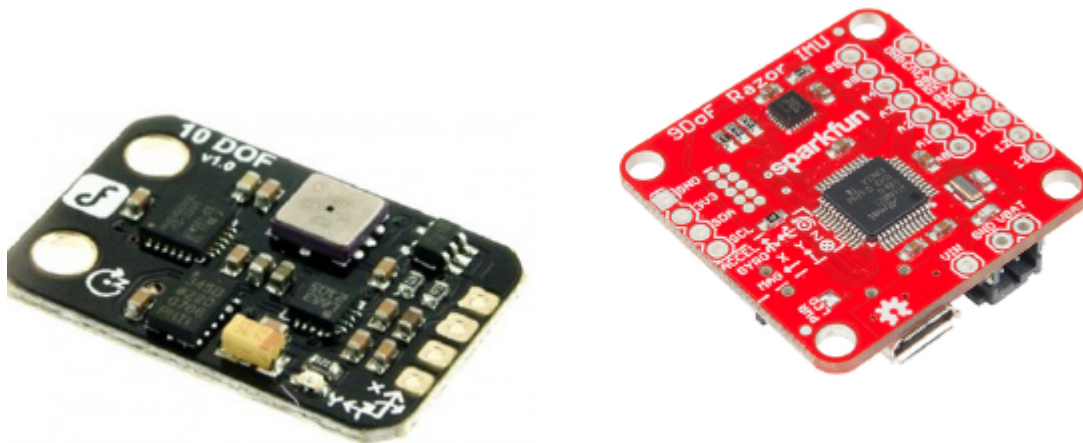


Figure 3.24 IMU

Comme nous avons mentionné l'idée principale du filtre est de faire une bonne estimation en minimisant l'erreur par fusionnement de plusieurs paramètres ou plusieurs informations délivrées au système de contrôle à partir des différents types de capteurs.

L'algorithme de fusion de capteurs mis en œuvre dans notre cas utilise le gyroscope en tant que source principale d'orientation. Malheureusement, les gyroscopes sont soumises à la déviation d'orientation estimée après un certain temps et s'éloigne de celle réelle. Ce problème est résolu en fusionnant le gyroscope à l'accéléromètre, ce qui compense la déviation des angles de roulis et de tangage, et le magnétomètre compense la déviation suivant le lacet. L'idée de base est

d'ajouter la gravité et le nord magnétique comme point de référence ou à base de Kalman vecteur de correction.

3.5.1 Technique d'estimation d'angle ϕ, θ

Le gyroscope est la source principale d'orientation et d'entrée a notre systeme :

$$\begin{cases} GY R_x^m(k) = \dot{\phi} + n_x + b_x \\ GY R_y^m(k) = \dot{\theta} + n_y + b_y \end{cases} \quad (3.12)$$

On prend

$$\begin{cases} \dot{\phi}(t) = GY R_x^m(k) - n_x - b_x \\ \dot{b}_x(t) = n_x^b \\ \dot{\theta}(t) = GY R_y^m(k) - n_y - b_y \\ \dot{b}_y(t) = n_y^b \end{cases} \quad (3.13)$$

Après discrétisation d'une période T_s :

$$\begin{cases} \phi(k+1) = \phi(k) + T_s \cdot GY R_x^m(k) - T_s \cdot n_x(k) - T_s \cdot b_x(k). \\ b_x(k+1) = b_x(k) \\ \theta(k+1) = \theta(k) + T_s \cdot GY R_y^m(k) - T_s \cdot n_y(k) - T_s \cdot b_y. \\ b_y(k+1) = b_y(k) \end{cases} \quad (3.14)$$

Donc notre système peut être mis sous la forme : $x(k+1) = Ax(k) + Bu(k) + v(k)$. Telque

le vecteur d'état :

$$\begin{bmatrix} \phi(k) \\ b_x(k) \\ \theta(k) \\ b_y(k) \end{bmatrix}$$

Le vecteur d'information délivrés par l'accéléromètre est donné par les équations suivants :

$$\begin{cases} \tan \phi_{ACC}^m = \frac{ACC_y^m}{ACC_z^m} \\ \tan \theta_{ACC}^m = \frac{ACC_x^m}{\sqrt{(ACC_y^m)^2 + (ACC_z^m)^2}} \end{cases} \quad (3.15)$$

L'équation (3.15) exprime les équations de mesure mais ces équations ne sont pas valides sauf si on a un système stable et une faible accélération (le lancement non agressif).

L'équation de mesure peut s'écrire sous la forme $y = Cx(k) + W(k)$.

À ce stade toutes les données de filtre de kalamn sont prêtes. la matrice d'estimation de la covariance de l'erreur préalable $v(k)$, la matrice $W(k)$ la matrice de la covariance de l'erreur du processus, les matrices de A , B , C de modèle de notre système. Toutes ces données sont des données non changables par contre $u(k)$, $y(k)$ ce sont des données changable à chaque fois dans la boucle de calcul $u(k)$ pour la prédiction, et $y(k)$ pour la mise a jour des mesures.

3.5.2 Estimation d'angle ψ

Pour l'estimation d'angle ψ en utilisent les données de magnétomètre qui va compenser la déviation suivant le lacet. L'angle ψ peut être estimé en utilisant les équations suivantes :

Dans le cas ou n'avons pas des mesures de ϕ et θ

$$\tan \psi = \frac{MAg_y^m}{MAg_x^m} \quad (3.16)$$

Dans le cas contraire nous devons introduire la notion de système de compensation ou

$$\begin{aligned} MAg^h &= R_{\phi,\theta}.MAg^m \\ \tan \psi &= \frac{MAg_y^h}{MAg_x^h} \end{aligned} \quad (3.17)$$

La matrice $R_{\phi,\theta}$ est la matrice de rotation.

$$R_{\phi,\theta} = \begin{bmatrix} \cos \theta & \sin \phi \sin \theta & \cos \phi \sin \theta \\ 0 & \cos \phi & -\sin \theta \\ -\sin \theta & \sin \phi \cos \theta & \cos \phi \cos \theta \end{bmatrix}. \quad (3.18)$$

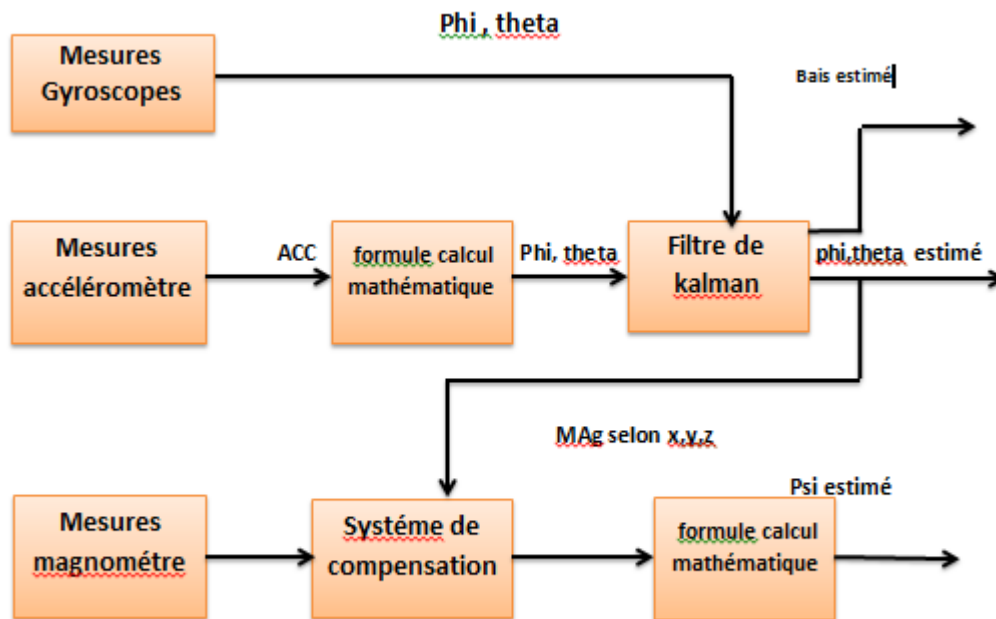


Figure 3.25 Schéma d'estimation d'attitude

3.6 Simulation d'un quadrirotor en utilisant ROS

Ces dernières années, le traitement en temps réel s'est imposée comme une discipline importante des sciences et du génie informatiques.

ROS est l'abréviation de « Robot Operating System », comme son nom l'indique ROS est un système d'exploitation pour robot, il a été créé en 2007 sous le nom de « ligne d'interconnexion » par le Laboratoire d'intelligence artificielle de Stanford avec l'appui du projet de l'AI Robot Stanford « STAIR ». En 2008, le développement s'est poursuivi principalement au Willow Garage (créé et financé par les dirigeants de Google). Cet organisme est un institut de recherche en robotique/incubateur ou plus de vingt institutions collaborent dans le sens d'un modèle de développement fédéré et ouvert (aux grandes universités et aux grands industriels de la robotique).

De ce fait, l'outil ROS est classé open source afin d'accélérer les recherches et d'améliorer les robots.

Il existe plusieurs distributions qui ont été améliorées et dont les quatre dernières années :

Tableau 3.2 les distributions du ROS

Distribution	Date
ROS Noetic Ninjemys	Mai, 2020
ROS Melodic Morenia	Mai 2018
ROS Lunar Loggerhead	Mai, 2017
ROS Kinetic Kame	Mai, 2016

Dans notre cas, nous avons travaillé avec la version *Melodic Morenia* de la distribution ROS sous un système d'exploitation Linux (Ubuntu 18.04).

Vous trouvez Une explication du concept et du système de fichiers du ROS (A)

3.6.1 Resultat d'une commande et implimentation d'un estimation d'angle d'un seul quadrotor

Comme nous l'avons mentionné précédemment l'outil ROS est classé open source dont il n'est pas nécessaire de refaire tout le travail (création d'environnement ,modélisation des moteurs des drones ...). Il suffit juste de choisir un type de drone, ajouter des capteurs de mesure, utiliser un package et de tester la commande qui a été programmée en python. Nous avons validé nos commandes avec un drone IRIS. Le quadcopter IRIS est un tout-en-un autonome, élégant et facile à utiliser est parfait pour toute application d'imagerie aérienne.



Figure 3.26 IRIS quadrirotor

3.6.1.1 Résultat d'estimation d'attitude en temps réel

Les mesures fournies par une unité de mesure inertielle (IMU) ont été enregistrées dans des Bags (vecteurs d'enregistrements sous ROS). Nous avons calculé les angles de roulis, de tangage et de lacet à partir des mesures des capteurs. Les angles estimés seront comparés aux angles déduits du quaternion calculé par le traitement interne de l'IMU, qui sera supposé suffisamment précis pour être considéré comme une vérité-terrain. Il faut motioneer qu'on peut séparer le système d'estimation de roulis et de tangage car les deux angles ont des paramètres indépendants.

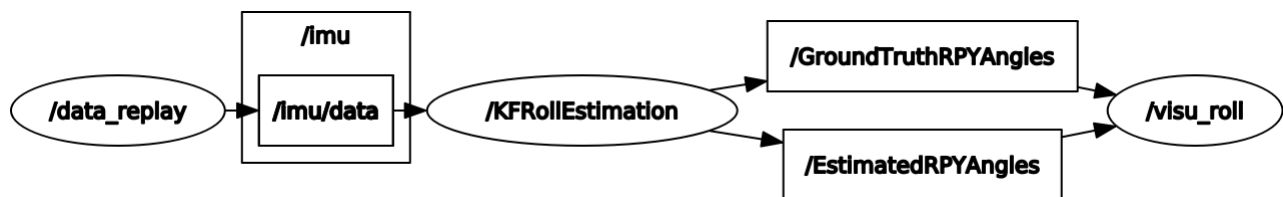


Figure 3.27 Rqt Graph d'implémentation d'estimateur d'angle ϕ

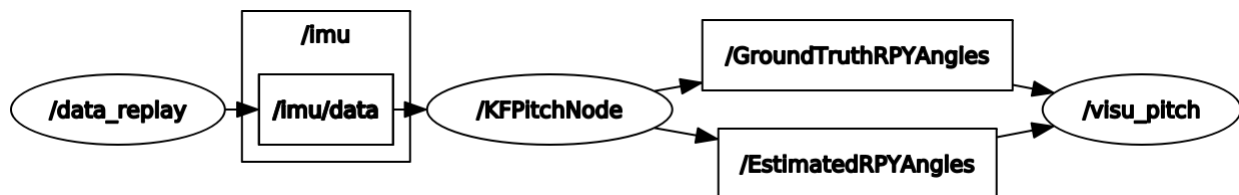


Figure 3.28 Rqt Graph d'implémentation d'estimateur d'angle θ

(Figure.3.27), (Figure.3.28) expriment l'architecture d'estimation que nous utilisons dans nos travaux avant de l'implémenter sur IRIS.

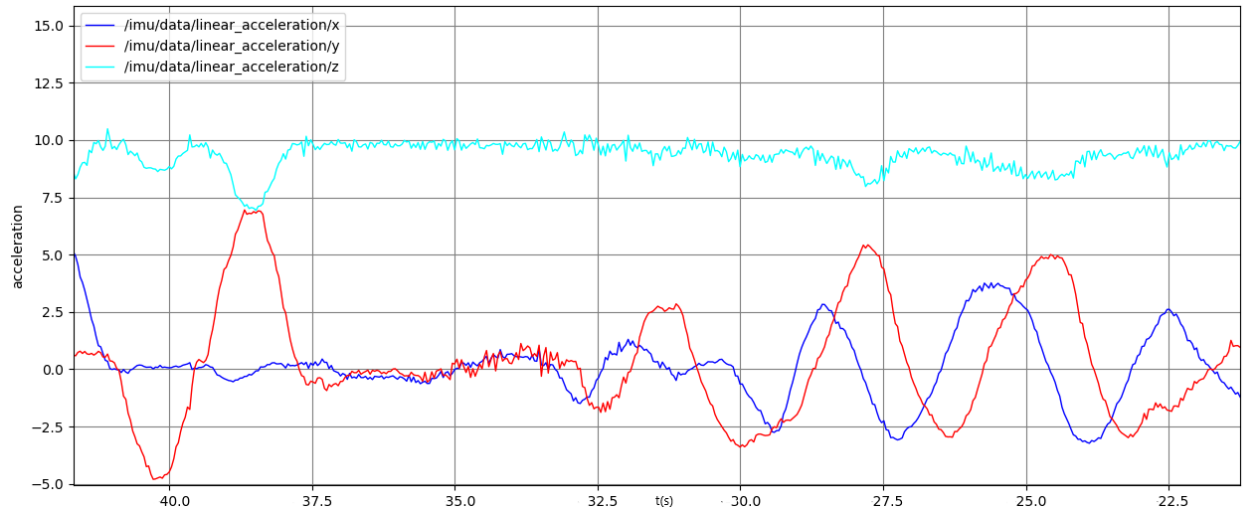


Figure 3.29 signal bruité d'accéléromètre

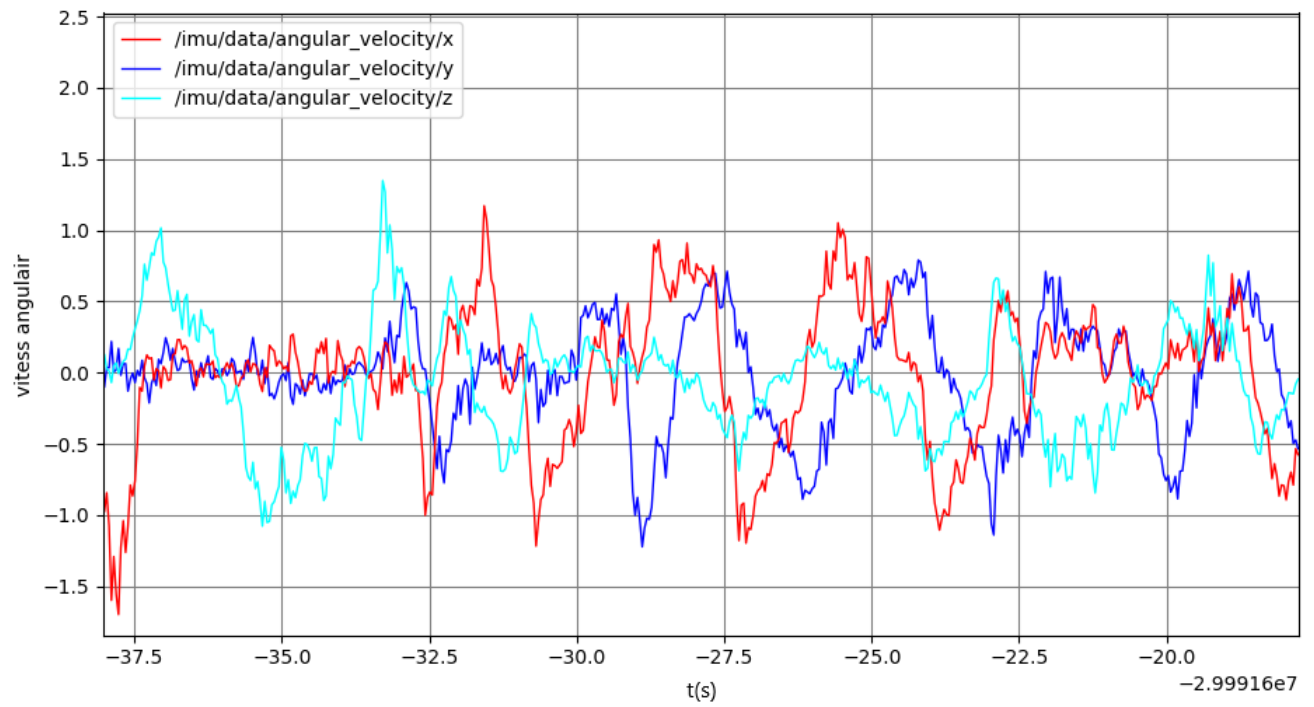


Figure 3.30 vitesses angulaire de gyromètre

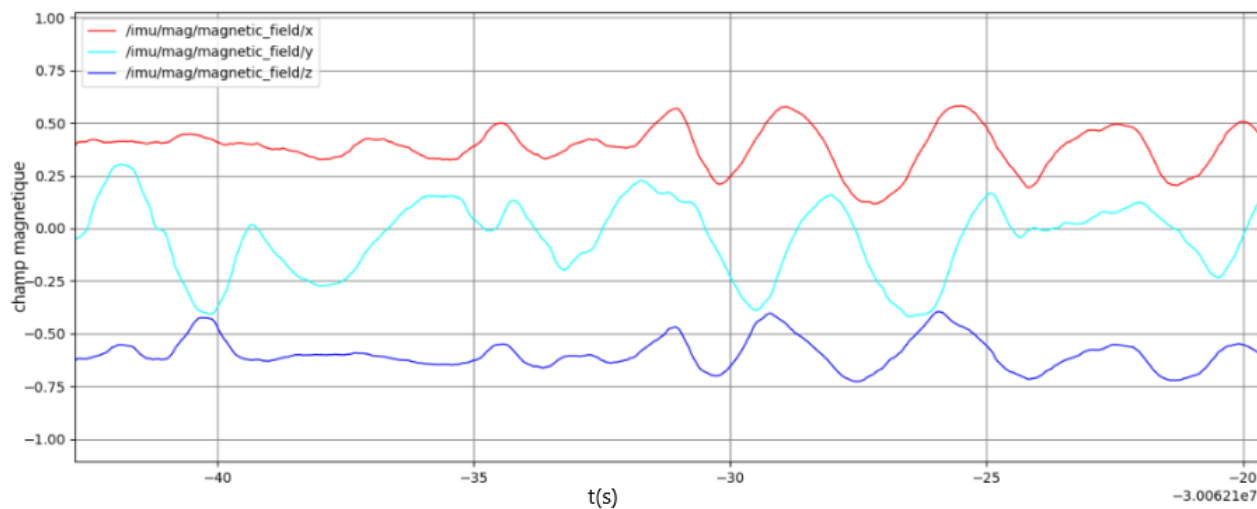


Figure 3.31 champ magnetique de mangométre

Nous présentons d'abord les mesures fournies par une unité de mesure inertielle (IMU) dans les figures (Figure.3.29), (Figure.3.30), (Figure.3.31). Les angles estimés (Figure.3.32), (Figure.3.33) seront comparés aux angles calculés par le traitement de l'IMU, qui seront supposés précis.

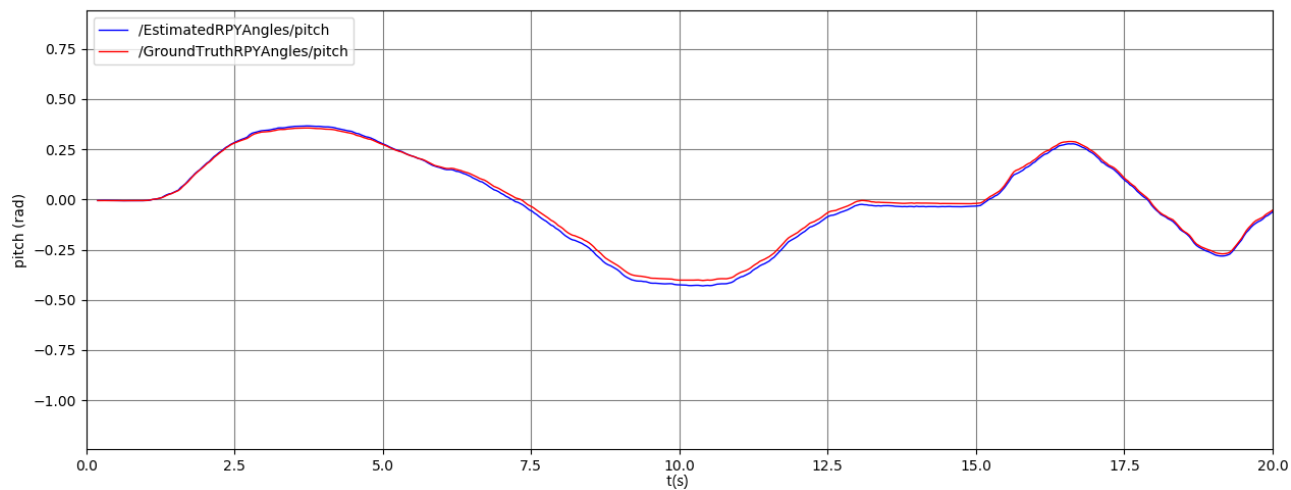


Figure 3.32 Résultats de l'implémentation d'un filtre kalman sur l'angle θ

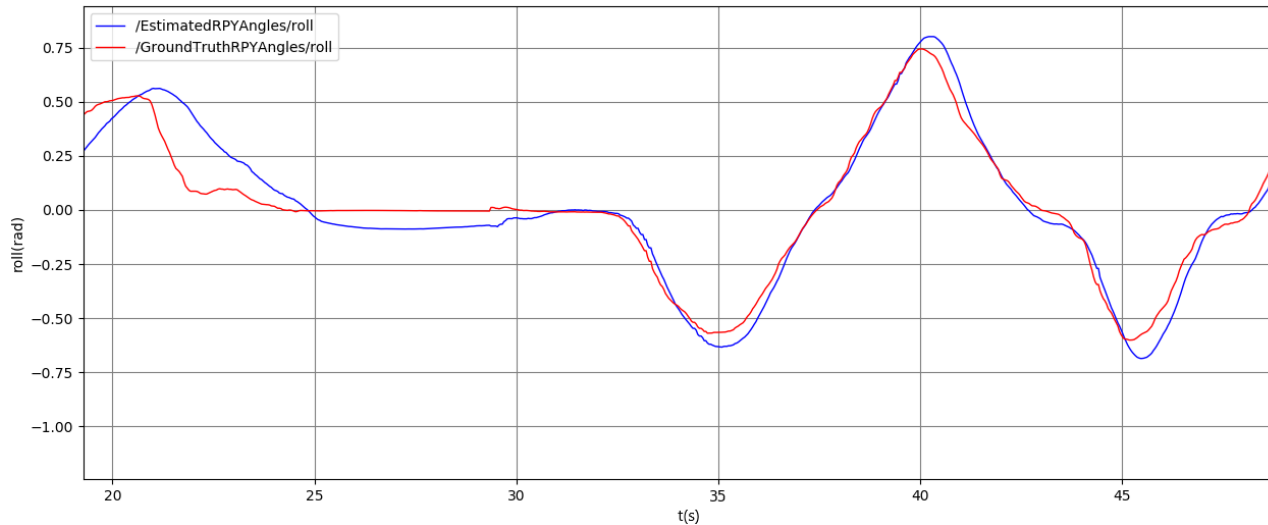


Figure 3.33 Résultats de l'implémentation d'un filtre kalman sur l'angle ϕ

Le nœud d'estimation KF (/KFPitchNode)/(/KFPitchNode) dans la (Figure.3.27), (Figure.3.28) reçoit l'accélération et la vitesse angulaire fournies par l'IMU à l'aide de Topic (/imu/data) ensuite chaque nœud publie l'angle estimé.

Dans ce qui suit, nous prenons les mêmes nœuds de la section précédente et nous ajoutons l'estimation d'angle ψ tel que nous l'avons mentionné dans (Figure.3.25) et nous l'impliquons tout le code sur le quadrotor IRIS.

Le schéma (3.34) explique le déroulement du process entre ROS et gazebo

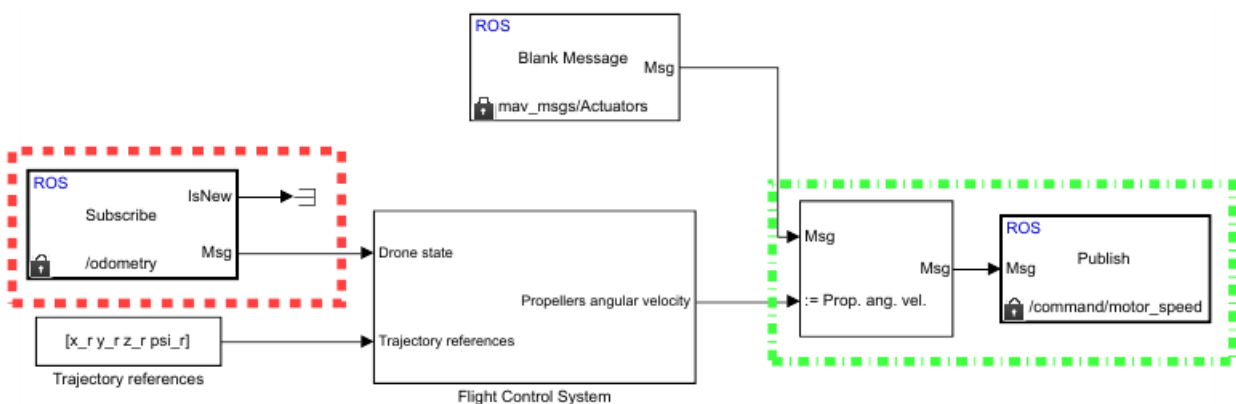


Figure 3.34 Un schéma descriptif de la compilation du programme entre ROS et gazebo

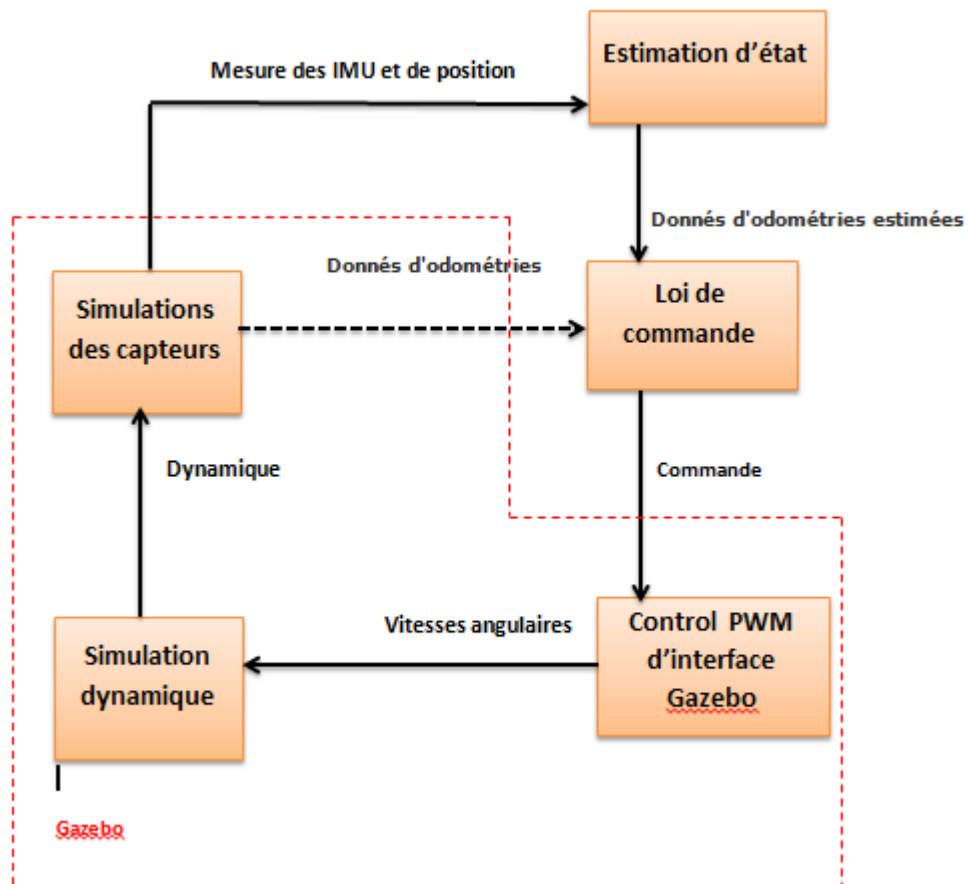


Figure 3.35 Graphe explique le passage des commandes au IRIS

La (Figure.3.36) exprime l'une des méthodes de modélisation graphique du ROS. Il fournit un plugin GUI pour visualiser le graphe de calcul ROS. Les node (ellipse) et les topics (carrés). Les flèches continues sont les topics actifs qui ont une direction allant du subscriber node au publisher node.

Nous avons développé un nouveau nœud pour l'estimation d'attitude d'IRIS en utilisant un filtre de Kalman en implémentant la commande PID que nous avons testée.

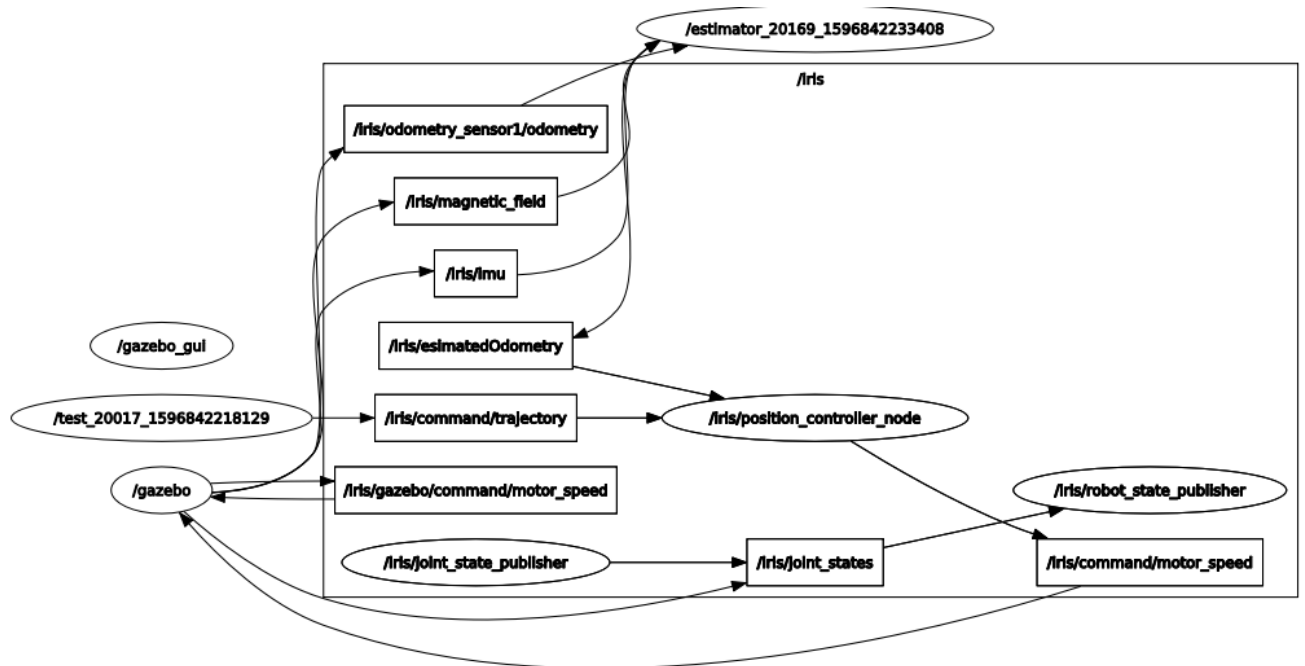


Figure 3.36 RQT graph de la commande d'IRIS

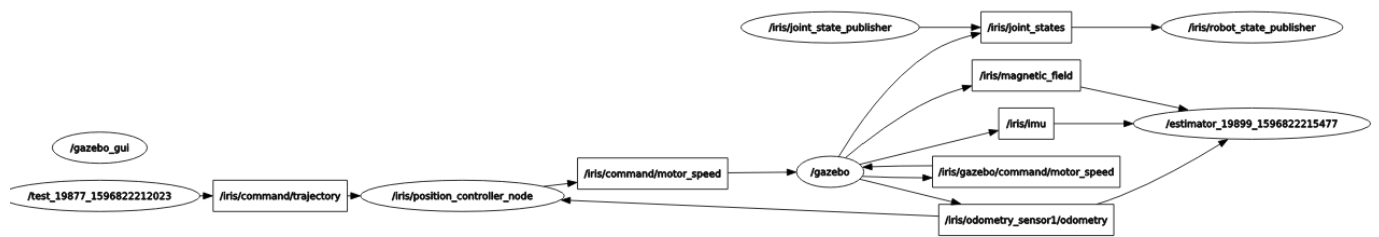


Figure 3.37 RQT graph d'application de filtre de kalman sur IRIS

(Estimator) c'est le bloc responsable de l'exécution l'algorithmes KF présentés. Ce bloc recoit en entrée les signaux issues des capteurs (signaux bruts de l'accéléromètre et du gyroscope et de magnomètre). Sa sortie est l'orientation du quadrirotor estimé. Le bloc (position controller node) est un régulateur PID présenté dans ce rapport. Il recoit en entrée les valeurs estimées de la pose du quadcopter, En sortie, il donne des commandes au moteur de quadrotor

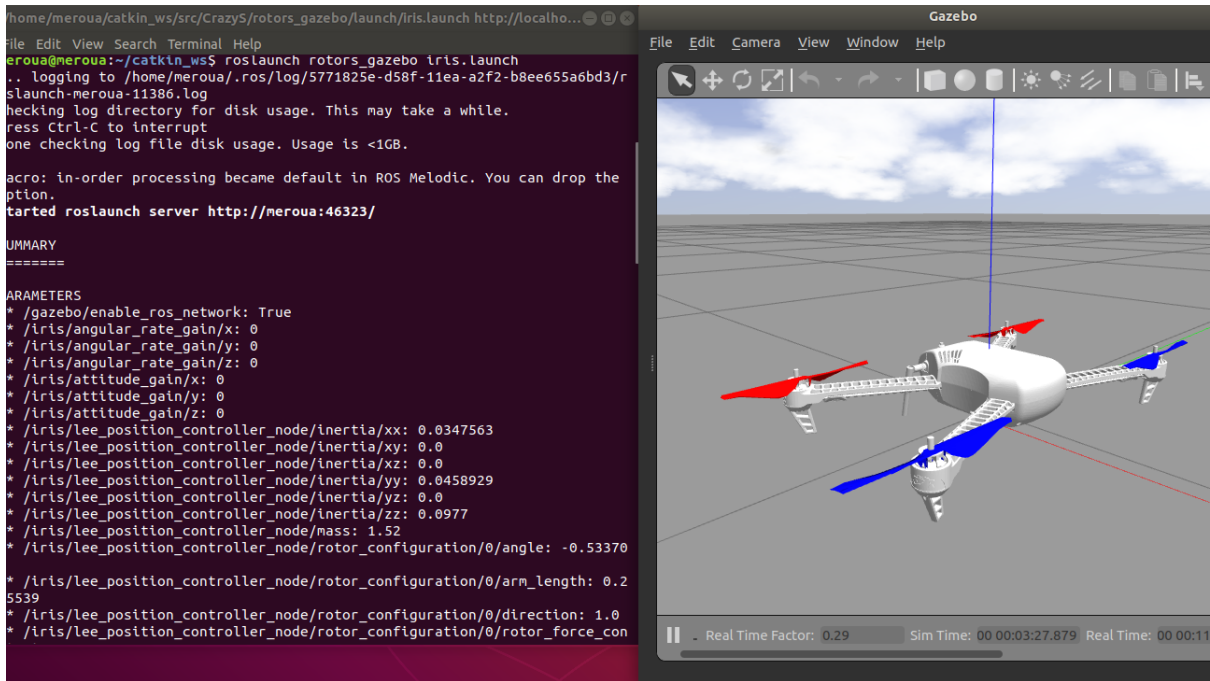


Figure 3.38 L'appel de quadrotor iris avec le fichier launch

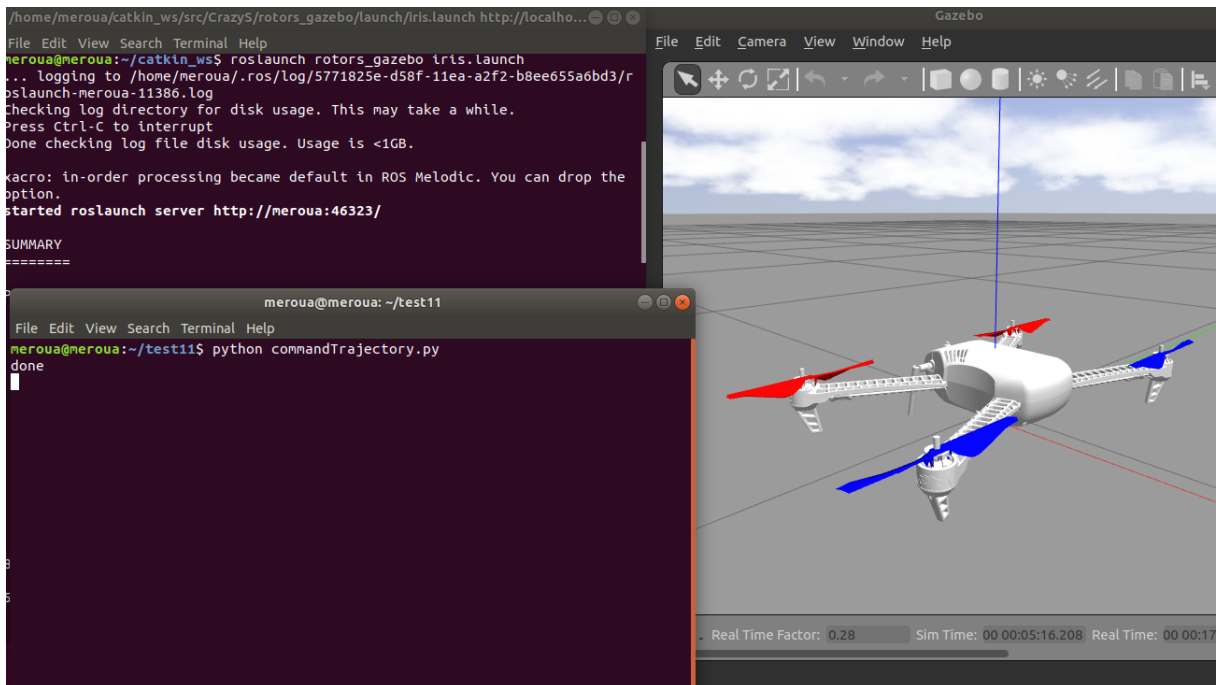


Figure 3.39 Appel de node de commande PID

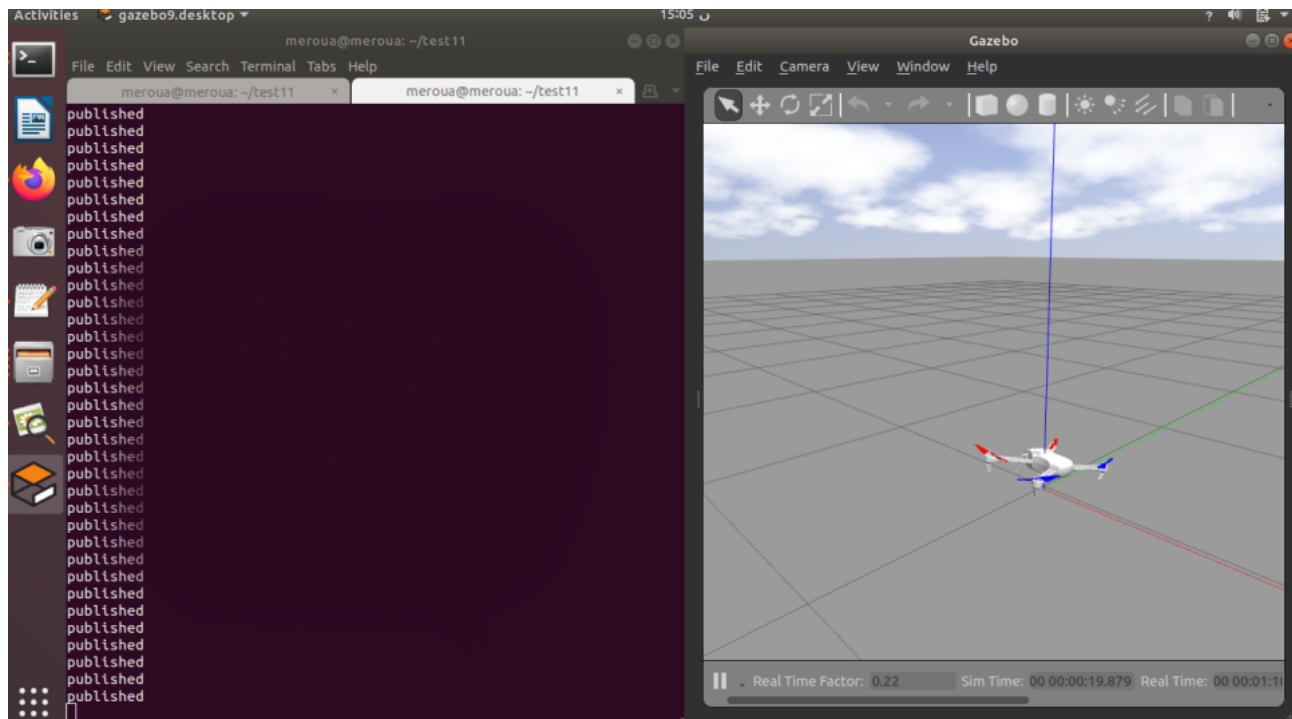


Figure 3.40 Appel depuis le nœud d'estimation qui publie dans Gazebo

3.6.2 Interprétation des résultats

Dans cette approche nous avons utilisé les mesures d'IMU pour estimer l'attitude d'un quadrotor. L'utilisation de cette technique nécessite des petits angles et aussi une petite accélération, c'est pourquoi le bon choix de commande appliquée sur le quadrotor influence les résultats. L'avantage de cette technique qu'elle est facile à mettre en œuvre, le filtre de Kalman utile pour corriger l'existence d'incertitude et des bruits qui agissent sur les différents capteurs mais elle n'est pas robuste surtout dans le cas de l'estimation de l'angle ψ parce que cet angle utilise les valeurs de l'angle ϕ , θ estimés alors pour les cas où il y a un échec dans l'estimation cela peut provoquer une instabilité de notre système. Pour renforcer notre technique, il faut donc soit ajouter d'autres capteurs (GPS, caméra, etc.) dans IMU car la fusion de capteurs améliore le processus d'estimation en combinant les mesures de différents types de capteurs et compenser les inconvénients de certains capteurs par les avantages de certains d'autre, soit utiliser des observateurs non linéaire comme EKF, l'observateur à mode glissant...

3.7 Conclusion

Dans cette section, nous avons implémenté la commande PID synthétisée au chapitre 2 et nous avons interprété les résultats qu'elle fournit. Puis les différents types d'observateurs sont évalués par des simulations avec des scénarios spécifiques. Les résultats montrent une amélioration considérable de précision. Enfin nous avons présenté ROS, ses notions de bases. Ensuite, nous avons implémenté une méthode d'estimation d'attitude pour le quadrotor IRIS basée sur le filtre de Kalman sous ROS et Gazebo.

CONCLUSION

Les quadrotors sont l'un des mini drones les plus populaires dans le domaine d'aviation et de la robotique à cause des modèles dynamiques précis et caractéristiques de stabilité favorables ainsi le vol stationnaire à proximité des sites spécifiés qui offrent par rapport aux autres drones à voilure tournante et fixe.

Ce drone est un système complexe, non linéaire, multivariables, instable et présente une dynamique fortement couplée, ce qui a fait de sa commande un grand défi. Par conséquent, il est devenu un sujet de recherche de plusieurs équipes et laboratoires ces dernières années.

Nous avons commencé le document avec les aperçus des quadrotors avec leur histoire. Ensuite, nous avons décrit les mouvements de base du quadrotor, et donné son modèle dynamique par le formalisme de Newton-Euler. Puis le modèle était modifié en adoptant des hypothèses simplificatrices afin de faciliter la synthèse de loi de commande. Concernant le contrôle, nous présenterons les différents schémas de contrôle appliqué au quadrotor : PID, mode glissant et contrôle adaptatif ... et nous choisissons d'utiliser le contrôle PID pour assurer la stabilité du quadrotor. Une synthèse des observateurs est aussi évaluée par des simulations avec des scénarios et testée sur le quadrotor IRIS sous ROS et Gazebo.

L'un des problèmes majeurs de la solution finale que nous avons présentée est qu'elle n'est pas robuste surtout dans le cas de l'estimation de l'angle ψ . Parce que cet angle utilise les valeurs de l'angle ϕ , θ estimés alors pour les cas où il y a un échec dans l'estimation cela peut provoquer une instabilité de notre système.

Les futures recherches pourraient renforcer notre technique, soit en ajoutent d'autres capteurs (GPS, caméra, etc.) dans IMU car la fusion de capteurs améliorer le processus d'estimation en combinant les mesures de différents types de capteurs et compenser les inconvénients de certains capteurs par les avantages de certains autres (bruit/ biais/ précision/ fréquence...) et aussi améliorer la robustesse en cas de perte des capteurs et de perturbations (environnement), soit en utilisant des observateurs non linéaires comme EKF que nous avons détaillé dans ce mémoire.

Il est également possible d'ajouter des capteurs pour la prévention des obstacles, et intégrer un algorithme lié à l'intelligence artificielle qui rendrait IRIS totalement autonome pour des missions plus spécialisées.

RÉFÉRENCES

- [Adventures 2012] ADVENTURES, Technical : *QuadCopter Stabilization Control System "X Plus Configuration*. 2012. – URL <http://technicaladventure.blogspot.com/2012/09/quadcopter-stabilization-control-system.html>
- [Chekakta 2016] CHEKAKTA, Abdellah : *Conception, Modélisation et Commande d'un UAV de type Quadrirotor*, École Nationale Polytechnique d'Oran, Dissertation, 2016
- [De Dinechin und Melquiond 2015] DE DINECHIN, Lauter C. ; MELQUIOND, Guillaume : Mixed reality for robotics, IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS). In : *IEEE* (2015)
- [Khalil 2011] KHALIL, Hassan K. : Nonlinear Systems. In : *Nonlinear Systems*. Upper Saddle River, N.J, 2011, S. 565 – 569
- [Larbi 2015] LARBI, Mohamed E. : *Synthèse des Observateurs pour un Dorne Hélicoptère*, UNIVERSITE DES SCIENCES ET DE LA TECHNOLOGIE D'ORAN MOHAMED BOUDIAF, Dissertation, 2015
- [Li und Li 2011] LI, Jun ; LI, Yuntang : Dynamic analysis and PID control for a quadrotor. In : *Mechatronics and Automation (ICMA), 2011 International Conference on IEEE* (Veranst.), 2011, S. 573–578
- [Madani 2006] MADANI, Abdelaziz : Backstepping control for a quadrotor helicopter. In : *Intelligent Robots and Systems, 2006 IEEE/RSJ International Conference on IEEE* (Veranst.), 2006, S. 3255–3260
- [Makarov 2019] MAKAROV, M. : Sensor fusion and state estimation, 2019, S. 16
- [M.GUETTACHE 2019] M.GUETTACHE, Y.BENAMEUR : *Perception et Navigation visuelle d'un quadrirotor en utilisant des techniques d'apprentissage profond*, École Polytechnique d'Alger, Dissertation, 2019
- [Minh und Ha 2010] MINH, Ly D. ; HA, Cheolkeun : Modeling and control of quadrotor MAV using vision-based measurement. In : *Strategic Technology (IFOST), 2010 International Forum on IEEE* (Veranst.), 2010, S. 70–75
- [M.Mokhtari 2015] M.MOKHTARI : *Observation et Commande de Drones Miniatures à voilures tournantes*, Université Aboubekr Belkaid Tlemcen, Dissertation, 2015
- [Palunko und Fierro 2011] PALUNKO, Ivana ; FIERRO, Rafael : Adaptive control of a quadrotor with dynamic changes in the center of gravity. In : *Proceedings 18th IFAC World Congress Bd. 18*, 2011, S. 2626–2631

- [Quan 2017] QUAN, Quan : *Introduction to multicopter design and control*. Springer, 2017
- [Zulu 2016] ZULU, Samuel : A review of control algorithms for autonomous quadrotors.
In : *arXiv preprint arXiv :1602.02622* (2016)

ANNEXE A Robot Operating System/ROS

Comme son nom l'indique, ROS (Robot Operating System) est un système d'exploitation pour robots. De même que les systèmes d'exploitation pour PC, serveurs ou appareils autonomes, ROS est un système d'exploitation complet pour la robotique de service.

ROS en quelques mots Le principe de base d'un OS robotique est de faire fonctionner en parallèle un grand nombre d'exécutables qui doivent pouvoir échanger de l'information de manière synchrone ou asynchrone. Par exemple, un OS robotique doit interroger à une fréquence définie les capteurs du robot (capteur de distance à ultrasons ou infrarouge, capteur de pression, capteur de température, gyroscope, accéléromètre, caméras, microphones...), récupérer ces informations, les traiter (faire ce que l'on appelle la fusion de données), les passer à des algorithmes de traitement (traitement de la parole, vision artificielle, localisation et cartographie simultanée...) et enfin contrôler les moteurs en retour. Tout ce processus s'effectue en continu et en parallèle. D'autre part, l'OS robotique doit assurer la gestion de la concurrence afin d'assurer l'accès efficace aux ressources du robot. Nous décrivons ci-dessous les concepts regroupés dans ROS sous le nom de « ROS Computation Graph » et qui permettent d'atteindre ces objectifs.

— Les paquets :

sont l'unité principale pour l'organisation de logiciels, ils sont l'élément le plus atomique dans ROS. un paquet peut contenir des processus d'exécution ROS (noeuds - nodes), une bibliothèque dépendant de ROS, des données, des fichiers de configuration, ou tout autre élément organisé de manière utile

— Traitement des paquets :

1. Le Master :

Le Master est un service de déclaration et d'enregistrement des noeuds qui permet ainsi à des noeuds de se connaître et d'échanger de l'information.

2. Les noeuds :

Les noeuds (nodes) sont des processus qui effectuent le traitement. par exemple, un noeud contrôle le capteur laser, un autre noeud contrôle les moteurs des roues, un noeud effectue la localisation, un noeud effectue la planification du chemin, un noeud fournit une vue graphique du système, etc. Un noeud ROS est écrit avec l'utilisation d'une bibliothèque client ROS, comme roscpp ou rospy.

3. Les topics :

L'échange de l'information s'effectue soit de manière asynchrone via un topic ou de manière synchrone via un service. Un topic est un système de transport de l'information basé sur le système de l'abonnement / publication (subscribe / publish). Un ou plusieurs nœuds pourront publier de l'information sur un topic et un ou plusieurs nœuds pourront lire l'information sur ce topic. Le topic est en quelque sorte un bus d'information asynchrone un peu comme un flux RSS. Cette notion de bus many-to-many asynchrone est essentielle dans le cas d'un système distribué. Le topic est typé, c'est-à-dire que le type d'information qui est publiée (le message) est toujours structuré de la même manière. Les nœuds envoient ou reçoivent des messages sur des topics.

4. Les services :

Le topic est un mode de communication asynchrone permettant une communication many-to-many. Le service en revanche répond à une autre nécessité, celle d'une communication synchrone entre deux nœuds. Cette notion se rapproche de la notion d'appel de procédure distante (remote procedure call).

5. Les messages :

Un message est une structure de donnée composite. Un message est composé d'une combinaison de types primitifs (chaînes de caractères, booléens, entiers, flottants...) et de message (le message est une structure récursive). Par exemple un nœud représentant un servomoteur du robot, publiera certainement son état sur un topic (selon ce que vous aurez programmé) avec un message contenant par exemple un entier représentant la position du moteur, un flottant représentant sa température, un autre flottant représentant sa vitesse.

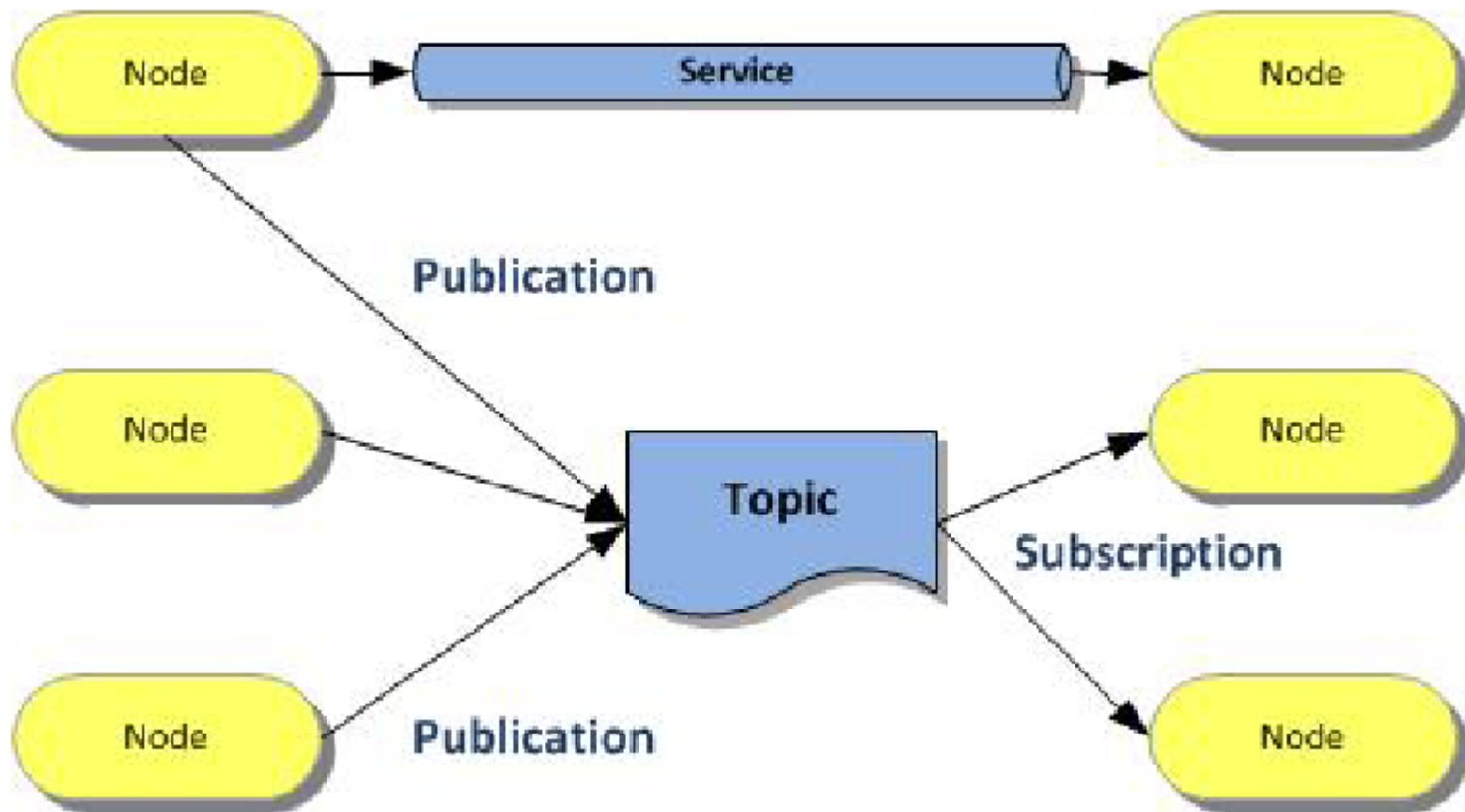


Figure A.1 schéma descriptif de traitement des paquets

ANNEXE B Gazebo

Pourquoi Gazebo ? La simulation de robot est un outil essentiel dans la boîte à outils de chaque roboticien. Un simulateur bien conçu permet de tester rapidement des algorithmes, de concevoir des robots, d'effectuer des tests de régression et de former un système d'IA à l'aide de scénarios réalistes. Gazebo offre la possibilité de simuler avec précision et efficacité des populations de robots dans des environnements intérieurs et extérieurs complexes.

À portée de main est un moteur physique robuste, des graphiques de haute qualité et des interfaces programmatiques et graphiques pratiques. Le meilleur de tous, Gazebo est gratuit avec une communauté dynamique.

Caractéristiques Gazebo est un logiciel libre financé en partie par Willow Garage et peut être reconfiguré, développé et modifié. Il est compatible avec ROS et Player. On peut exécuter Gazebo à partir de ROS et utiliser les API de ces dernières pour contrôler les robots dans les simulations, c'est-à-dire envoyer et recevoir des données de ceux-ci.

Ce logiciel permet de faire des simulations réalistes de la physique des corps rigides. Les robots peuvent interagir avec le monde (ils peuvent ramasser et pousser des objets, rouler et glisser sur le sol) et inversement (ils sont affectés par la gravité et peuvent se heurter à des obstacles dans le monde). Pour ce faire, Gazebo utilise de multiples moteurs physiques tels que Open Dynamics Engine (ODE) ou Bullet.

Premiers pas avec Gazebo et ROS Vous devez avoir préalablement installé Gazebo et ROS. Vous êtes maintenant prêt à découvrir le monde de la simulation. :

ANNEXE C Notion sur la stabilité au sens de lyapunov

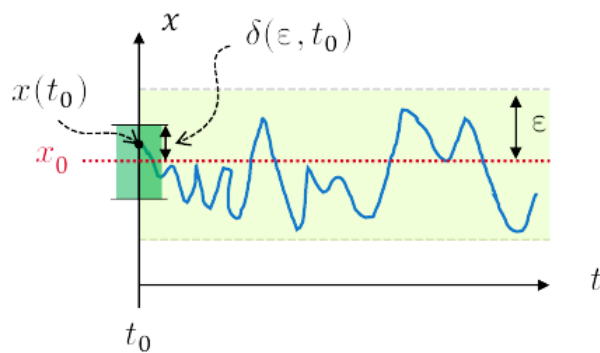
([Khalil, 2011]),([Makarov, 2019])

Lyapunov stability

$\forall \varepsilon > 0, \forall t_0, \exists \delta(\varepsilon, t_0) \text{ s. t. :}$

$$\|\mathbf{x}(t_0) - \mathbf{x}_0\| < \delta(\varepsilon, t_0)$$

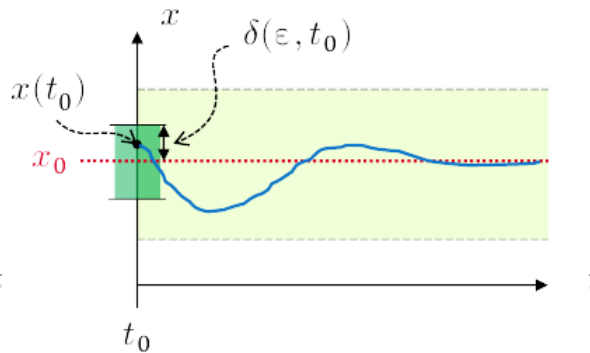
$$\Rightarrow \|\mathbf{x}(t) - \mathbf{x}_0\| < \varepsilon, \forall t \geq t_0$$



Asymptotic stability

i. Lyapunov stability

ii. $\lim_{t \rightarrow \infty} \|\mathbf{x}(t) - \mathbf{x}_0\| = 0$



Exponential stability

i. Asymptotic stability

ii. $\exists M, \alpha > 0 \text{ s. t. :}$

$$\|\mathbf{x}(t) - \mathbf{x}_0\| \leq M \|\mathbf{x}(t_0) - \mathbf{x}_0\| e^{-\alpha(t-t_0)}$$

Figure C.1 Stabilité au sens de lyapunov

ANNEXE D Paramètres utilisés

Tableau D.1 Tableaux des paramètres physique

Paramètre	Symbole	Valeur	Unité
Masse du quadrirotor	m	0.468	kg
Distance entre le centre d'un moteur et le centre de gravité	l	0.225	m
Moment d'inertie du quadrirotor par rapport à son axe 'X'	J_x	$4.856 \cdot 10^{-3}$	kg.m^2
Moment d'inertie du quadrirotor par rapport à son axe 'Y'	J_y	$4.856 \cdot 10^{-3}$	kg.m^2
Moment d'inertie du quadrirotor par rapport à son axe 'Z'	J_z	$8.801 \cdot 10^{-3}$	kg.m^2
Moment d'inertie du rotor par rapport à son axe 'Z'	J_r	0.00006	kg.m^2
Coefficient de portance	b	$1.14 \cdot 10^{-7}$	N.s^2
Coefficient de traînée	k	$2.98 \cdot 10^{-6}$	N.m.s^2
Constante de gravité	g	9.81	m.s^{-2}

Tableau D.2 Tableaux des gains de réglage pour la commande PD

Constant	Valeur
$k_{p\phi}$	1
$K_{p\theta}$	1.5
$k_{p\psi}$	1.8
k_{pz}	120
$k_{d\phi}$	0.5
$k_{d\theta}$	0.5
$k_{d\psi}$	0.5
k_{dz}	25