N° Ordre...../IGEE/UMBB/2023

REPUBLIQUE ALGERIENNE DEMOCRATIQUE ET POPULAIRE

MINISTERE DE L'ENSEIGNEMENT SUPERIEUR ET DE LA RECHERCHE SCIENTIFIQUE

UNIVERSITE M'HAMED BOUGARA -BOUMERDES

Institut de **G**énie **E**lectrique et **E**lectronique

Thèse de **D**octorat

Présenté par:

Mr Saibi Ali

En vue d'obtention de diplôme de **Doctorat** en Sciences

Filière: Génie Electrique

Option: Automatique

TITRE: NEW APPROACH OF DRONE'S CONTROL

Soutenue le 25 /06/2023 Devant le jury :

Mr DAAMOUCHE Abdelhamid	Professeur	UMBB	Président
Mr BOUZOUIA Brahim	Professeur	CDTA	Examinateur
Mr ALLAL Abderrahim	MCA	U.El-oued	Examinateur
Mr IDIR Abdelhakim	MCA	U.M'sila	Examinateur
Mme BOUSHAKI Razika	Professeur	UMBB	Rapporteur
Mme BELAIDI hadjira	MCA	UMBB	Co- Rapporteur

Année Universitaire 2022/2023

Acknowledgment

The work presented in this thesis under the direction of **Pr. BOUSHAKI Razika,** professor at IGEE/UMBB, and **Dr.** B**ELAIDI Hadjira**, MCA at the same institution, to whom I express my gratitude for their constant follow-up and their sustained encouragement that they have not ceased to lavish on me until completion of this work.

I would like to warmly thank **Pr. DAAMOUCHE Abdelhamid,** professor at IGEE/UMBB who did me the honor of chairing the jerry. That my **Sirs**, **Pr. BOUZOUIA Brahim**, Professor at CDTA and **Dr. ALLAL Abderrahim**, MCA at Univ. El-oued and **Dr. IDIR Abdelhakim**, MCA at Univ M'sila, are warmly thanked for the interest they have shown in this work by agreeing to examine it.

Abstract

In this work, the control of a quadrotor was studied. After having found its mathematical model which makes it possible to simulate its behavior, three nonlinear controls were used: the proportional derivative controller (PDC), the backstepping control (BSC), and the sliding mode control (SMC); in order to study the performance of each of them in quadrotor trajectory tracking. Thus, by comparing the obtained results, it has been demonstrated that with all controllers, the position, orientation, and attitude route following errors can fastly converge to minor levels. In the presence of non-external disturbances, BSC controls the yaw angle and altitude of the quadrotor better than the other two controllers (SMC and PDC).

Furthermore, in the presence of disturbances, each controller's steady state error maintained the same order as in the absence of any disturbance. However, as the disturbance increased, the controllers were unable to keep the quadrotor on course. The numerical and simulation findings show that BSC is the last one to collapse, confirming the robustness and efficacy of our built-enhanced control technique.

As the next step, the Extended Kalman Filtre (EKF) was used to estimate the states of the system and control it without using angular and linear speed sensors. Moreover, the robustness of the controls in the face of the disturbance of the wind force was studied and estimated using the EKF and was compensated in real-time. It has been proved that the quadrotor returns to the target trajectory after a given amount of time (depending on the dynamics of the EKF estimator). This method's resilience is obvious, and it fills the gaps missed by controllers.

الملخص

في هذا العمل ، تمت دراسة التحكم في المحرك الرباعي. بعد العثور على نموذجها الرياضي الذي يجعل من الممكن محاكاة سلوكها ، تم استخدام ثلاثة عناصر تحكم غير خطية: المتحكم الاشتقاقي النسبي (PDC) ، والتحكم الخلفي (BSC) ، والتحكم في الوضع المنزلق (SMC) ؛ من أجل دراسة أداء كل منهم في تتبع المسار الرباعي. وبالتالي ، من خلال مقارنة النتائج التي تم الحصول عليها ، فقد ثبت أنه مع جميع وحدات التحكم ، يمكن أن يتقارب الموقف والتوجيه والمسار الذي يتبع الأخطاء بسرعة إلى مستويات ثانوية. في حالة وجود اضطرابات غير خارجية ، تتحكم BSC بزاوية الانعراج والارتفاع للرباعي بشكل أفضل من المتحكمات الأخريين (SMC و PDC).

علاوة على ذلك ، في حالة وجود اضطرابات ، حافظ خطأ الحالة الثابت لكل وحدة تحكم على نفس الترتيب كما هو الحال في حالة عدم وجود أي اضطراب. ومع ذلك ، مع زيادة الاضطراب ، لم تتمكن وحدات التحكم من الحفاظ على المحرك الرباعي في مساره. تظهر النتائج العددية والمحاكاة أن BSC هو آخر من ينهار ، مما يؤكد متانة وفعالية تقنية التحكم المدمجة لدينا.

كخطوة تالية ، تم استخدام Extended Kalman Filtre (EKF) لتقدير حالات النظام والتحكم فيه دون استخدام مستشعرات السرعة الزاوية والخطية. علاوة على ذلك ، تمت دراسة متانة أدوات التحكم في مواجهة اضطراب قوة الرياح وتقديرها باستخدام EKF وتم تعويضها في الوقت الفعلي. لقد ثبت أن الرباعي يعود إلى المسار المستهدف بعد فترة زمنية معينة (اعتمادًا على ديناميكيات مقدر EKF). مرونة هذه الطريقة واضحة ، وهي تملأ الفجوات التي فاتها المتحكمون.

Resumé

Dans ce travail, le contrôle d'un quadrirotor a été étudié. Après avoir trouvé son modèle mathématique permettant de simuler son comportement, trois commandes non linéaires ont été utilisées : la commande proportionnelle dérivée (PDC), la commande backstepping (BSC) et la commande par mode glissant (SMC) ; afin d'étudier les performances de chacun d'eux en suivi de trajectoire quadrirotor. Ainsi, en comparant les résultats obtenus, il a été démontré qu'avec tous les contrôleurs, les erreurs de suivi de route de position, d'orientation et d'attitude peuvent rapidement converger vers des niveaux mineurs. En présence de perturbations non externes, le BSC contrôle mieux l'angle de lacet et l'altitude du quadrirotor que les deux autres contrôleurs (SMC et PDC).

De plus, en présence de perturbations, l'erreur de régime permanent de chaque contrôleur conservait le même ordre qu'en l'absence de toute perturbation. Cependant, à mesure que la perturbation augmentait, les contrôleurs étaient incapables de maintenir le quadrirotor sur la bonne voie. Les résultats numériques et de simulation montrent que le BSC est le dernier à s'effondrer, confirmant la robustesse et l'efficacité de notre technique de contrôle amélioré.

Dans l'étape suivante, le filtre de Kalman étendu (EKF) a été utilisé pour estimer les états du système et le contrôler sans utiliser de capteurs de vitesse angulaire et linéaire. De plus, la robustesse des commandes face à la perturbation de la force du vent a été étudiée et estimée à l'aide de l'EKF et a été compensée en temps réel. Il a été prouvé que le quadrirotor revient à la trajectoire cible après un laps de temps donné (selon la dynamique de l'estimateur EKF). La résilience de cette méthode est évidente et comble les lacunes manquées par les contrôleurs.

Contents

List of Figures

List of Tables

List of symbols

b	thrust factor
c	propeller chord C propulsion group cost factor
d	drag factor
g	acceleration due to gravity
h	vertical distance: Propeller center to CoG H hub force
i	motor current
Ixx,yy,zz	inertia moments
Jr	rotor inertia
m	the quadrotor mass
l	horizontal distance: propeller center to CoG
x,y,z	position in the body coordinate frame
X,Y,Z	position in earth coordinate frame
θ	pitch angle
φ	roll angle
ψ	yaw angle
Ω	propeller angular rate
Ωr	overall residual propeller angular speed
$\dot{\eta}$	Euler rates
ω	angular body rates
R	the rotation matrix
τ	the net torque
I_{d3}	3 × 3 identity matrix
α	the angular acceleration
a	the net force acting on the quadrotor
I	is the inertia matrix of the quadrotor
M	are all the moments acting on the quadrotor in the body frame
ρ	is the air density
A	blade area
C_D, C_T	aerodynamic coefficients.

r	the radius of blade
K_f	are the aerodynamic force
K_M	moment constants
Ω_i	angular velocity of rotor i.
F	the net force acting on the quadrotor,
$(x^{"}, y^{"}, z^{"})$	represents the second derivative of position vector

Introduction

Recently, UAVs (Unmanned Aerial Vehicles), especially quadrotors, are largely used in different fields such as: military and civil rescue and surveillance tasks, reconnaissance applications, industry pipe maintenance, agriculture field monitoring, and several amateur tasks. This growing attractiveness is due to their simple structure and good stability, low cost and maneuverability [1].

Due to quadrotor's complicated mathematical modeling and the system non-linearity owing to its six degrees of freedom (translational and rotational motion) with only four control inputs, several researches were investigated to develop an efficient quadrotors' controller. Starting from traditional PID controller which is commonly used to preserve the equilibrium or a desired attitude of the drone; however, it does not ensure the robustness of the quadrotor, whatever is the controlling target Euler angle or angular rate, which leads to the cascade PID algorithm to provide better performance and motion stabilization [2-3]. However, the implemented system still needs to tackle the drone system non-linearity. In addition to adaptive finite-time control [4], feedback control based on the representation in quaternions [5] and LQR based controllers [6].

According to various existing research works, backstepping is the best solution for disturbances rejection because it improves the transient performance [7-10]. Moreover, Different techniques were established to estimate the required quadrotor's position and altitude measurements such as: Kalman filtering (KF) [11], the complementary filter (CF) [12] and Extended Kalman Filters (EKF) [13] estimators. Kalman filter is a well-known recursive algorithm that takes the stochastics states spaces model of the system together with measured outputs to achieve the optimal estimation states. The optimality of the state's estimation is achieved with the minimization of the mean estimation error. Therefore, EKF and backstepping were investigated in this work to estimate the angular velocity and the speed of our quadrotor; hence, track the desired trajectory and control the altitude.

However, the presence of external disturbances such as wind decreases the efficiency of the existing controllers [14-15]; therefore, environmental uncertainties such as turbulence influences and wind disturbances present additional forces and moments on the quadrotor dynamics which were ignored in the previous works [16-18]. Hence, the design of more reliable controller for quadrotor is

a challenging task which require an accurate wind forces estimation proposed in the last part of this paper using EKF. Simulation results are shown to verify the efficiency of the presented model which can considerably advance the trajectory tracking feature of the quadrotor under wind disturbances.

The work of this thesis focuses on new drone control strategies and more particularly on non-linear controls. In the first place, we studied the effect of external disturbances on the trajectory of the drone (wind force). In second place, our work will be devoted to the estimation of the states of the system using the extended Kalman filter (EKF), the latter will be used to estimate the force of the wind and compensate to reach the desired trajectory in the most unfavorable conditions.

In the first chapter, the kinematics and dynamics of the quadrotors are described in detail to aid in understanding the behavior of the drone. we will extract the nonlinear mathematical model of a quadrotor.

In the second chapter, nonlinear control (BSC **(**Backstepping Control) and SMC (Sliding Mode Control)) theories are used to control the quadrotors. In this part, an easy-to-implement enhanced BSC is developed; then, tested beside two other controllers: sliding mode (SMC) and proportional derivative controller (PDC) to keep the quadrotor tracking the desired trajectory both in a steady state and in the existence of outside instabilities. Finally, the three controllers' results are compared to determine which is the best.

Chapter three will be devoted initially to the estimation of the twelve states of the quadrotor using the extended Kalman filter, after which the nonlinear control will be carried out without angular and linear speed sensors.

Chapter four presents the estimation of the system states (quadrotors) where we assume that the wind force is a state of the system and try to compensate for it and minimize its impact on the system. A robustness test of this method will be carried out to shed light on the contribution that this technique can make to the control of the quadrotor.

Finally, a general conclusion on this work is given and some perspectives are proposed.

Chapter 1: Mathematical Model of Quadrotors

A drone, also known as an unmanned aerial vehicle (UAV), is an aerial vehicle without a human operator inside. It can be remotely or autonomously flown. It is either flown autonomously on a computer using pre-programmed flight plans or remotely by a pilot on the ground.

The first sections of this chapter start with a brief description of the drones, their various categories, and their main configurations. Then, detailed mathematical modeling of the quadrotors will be explained in the remaining sections of this chapter.

I.1. Drones' Classification

UAVs can be categorized in a variety of ways, such as by their range of operation, aerodynamic configuration, size, payload, or degrees of autonomy, among other factors.

I.1.1. Classification according to operation's range

According to their maximum altitude, resilience, and sizes, which range from large drones to micro drones, UAVs can be divided into seven distinct categories as follows:

- High-Altitude Long-Endurance (HALE) aircraft: these aircraft are perfect for offering surveillance, remote sensing, and communication relay capabilities for both military and civilian uses. HALE UAVs usually fly between 9 and 20 kilometers above the ground while circling particular points of interest at a slow speed, with a 24-hour resilience or longer.
- Medium-Altitude Long-Endurance (MALE): these UAVs have a range of 3 to 9 km and can travel for long periods of time, usually 24 to 48 hours. These drones can be used for law enforcement, boarder and coastal patrol, disaster survey, information relays, and environmental protection.
- Medium-Range or Tactical UAV (TUAV): Capable of altitudes between 100 and 300 km foe an endurance of approximately 10-12 hours. Compared to HALE and MALE, which are currently used mainly to support military applications, they are smaller and work with simpler systems.

- Close Range UAV: their operational range is 100 km. They are primarily used in civil applications like powerline inspection, crop spraying, traffic monitoring, national security, etc.…
- Mini UAV (MUAV): Within a range of up to 10 km, they can complete their tasks with a short endurance (30 minutes). They are mainly made for gathering information during close-quarters battles. Their ability to lift off and land vertically gives them a significant application in crowded environments.
- Micro UAV (MAV): Their wingspan can extend up to 150 millimeters. They are primarily utilized indoors, where they must hover and fly gently.
- Nano Air Vehicles (NAV): They are only about 10 mm in dimension. They are primarily used in groups for purposes like radar interference. If they have a tiny camera, they can also be used for close-range surveillance.

I.1.2. Aerodynamic configuration's classification

Based on their aerodynamic configuration, UAVs can be divided into two major categories as follows:

- Fixed-wing UAVs: need a runway to launch and land. They are able to cruise at high speeds for extended periods of time. They are primarily utilized for scientific purposes such as environmental tracking and meteorological reconnaissance.
- Rotary-wing UAVs: they are able to take off and land vertically. They have excellent maneuverability when hover and fly. Four additional groups can be created from the Ro-tary-wing UAVs which are:
  - Single-rotor: for stability, they have a primary rotor on top and a second rotor at the tail, similar to the helicopter configuration.
  - Coaxial: They have a single shaft with two rotors mounted on it that rotate in opposing orientations.
  - Quadrotor: they have four rotors fitted in a cross-like configuration. Whereas, our thesis is concerned with this drone's configuration.

- Multi-rotor: eight or six-rotor UAVs. Due to the large numbers of rotors, they have redundancy and are able to fly even when a motor breaks.

I.1.3. Quadrotor's description

Quadrotors consist of four actuators that are individually controlled to produce a relative thrust. Two of the rotors rotate clockwise, and the other two rotors rotate in counter-clockwise direction. Figure 1.1 shown below depicts that the rotors on the same axis rotate in the same direction.

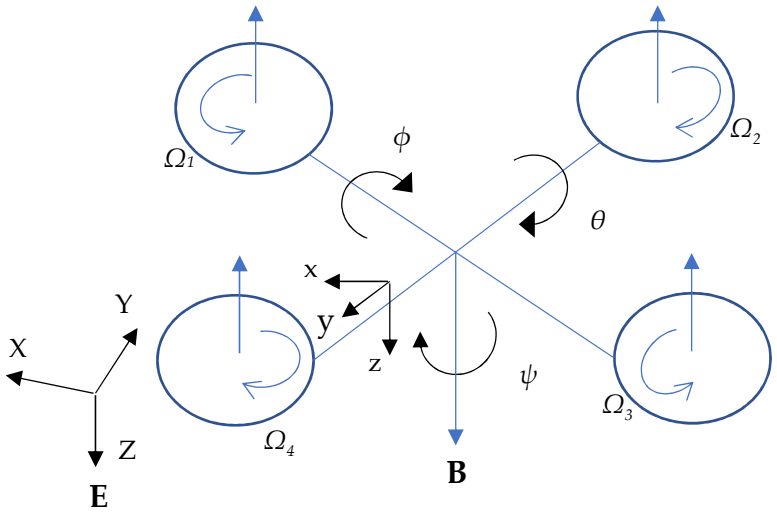


Fig. 1.1. Rotors rotation on quadrotor.

Even though the quadrotor has 6 DOF, it is equipped just with four propellers; hence it is not possible to reach a desired set-point for all the DOF, but at maximum four. However, thanks to its structure, it is quite easy to choose the four best controllable variables and to decouple them to make the controller easier. The four quadrotor targets are thus related to the four basic movements which allow the helicopter to reach a certain height and attitude. It follows the description of these basic movements: Throttle, Roll, Pitch and Yaw.

- Throttle is provided by increasing (or decreasing) all the propeller speeds by the same amount. It leads to a vertical force which raises or lowers the quadrotor.
- Roll is provided by increasing (or decreasing) the left propeller speed and by decreasing (or increasing) the right one. The motion in the 'x simply describes the quadrotor moving forward/backward.

- Pitch is very similar to the roll and is provided by increasing (or decreasing) the rear propeller speed and by decreasing (or increasing) the front one. The motion in the 'y' describes the quadrotor moving sideways.
- Yaw This command is provided by increasing (or decreasing) the front-rear propellers' speed and by decreasing (or increasing) that of the left-right couple. The motion in the 'z' direction describes the quadrotor turning left/right.

Two configurations generally exist within Quadrotors which are categorized as the '+' or 'X' sets, a comparison between the two suggests that the overall control authority from both configurations shows that the performance is identical.

I.2. State of the art

UAVs have become increasingly widely used in the civil and military fields and are the subject of concurrent scientific research, especially with the development of estimation and observation theory and artificial intelligence.

Among these works which have been published, we find: The paper presented in [1] applied LQ and LQG methodologies for quadcopter control systems. The article [2] has focused on optimizing the quadcopter system by implementing a controller that provides motion stabilization. In [6] an optimal control is developed for the position and yaw control of the quadcopter, based on linear quadratic regulator (LQR). The work in [7] investigates the problem of a robust path following control of a quadcopter unmanned aerial vehicle (UAV). In the article [8], the backstepping control strategy was devised to control the positions and orientations of the quadrotor subsystem. Several scenarios were performed to examine the performance of the backstepping strategy. To control UAVs, a simple control strategy based on a PD controller in the inner loop and a PI controller in the outer loop is proposed and tested in simulation and experimentation [9]. The authors of [10] used a robust back-stepping (RIB) integral control method to control a prototype quadcopter, and they concluded that the controller provides attitude and motion stabilization under various assumptions. Authors of [11] opted for the Aircraft position, speed, and attitude estimates can be obtained

by GPS/INS integration using Kalman filter implementation. [12]presents the proposal of a composition of a Complementary Filter used in orientation estimation and a Linear Kalman Filter (LKF) in position estimation of the UAVs. The data evaluated in the experiments were acquired post flight from the flight controller embedded in a UAV. The article [13] ackles the problem of constant positioning and collision avoidance on UAVs in outdoor (wildness) search scenarios by using received signal strength (RSS) from the onboard communication module. Colored noise is found in the RSS, which invalidates the unbiased assumptions in the least quares (LS) algorithms that are widely used in RSS-based position estimation. A colored noise model is thus proposed and applied in the extended Kalman filter (EKF) for distance estimation. The paper [14] proposed a novel adaptive formation control architecture for a group of quadrotor systems, under line-of-sight (LOS) distance and relative distance constraints as well as attitude constraints, where the constraint requirements can be both asymmetric and time-varying in nature. The authors of the article [15] used a steady-state identification method to estimate the parameters of the mathematical model of the variable-pitch electric propeller, and after adopted a control and optimization strategy for the variable-pitch propeller, mainly including the steady-state control and optimization scheme with minimum power consumption, [16] and numerical simulations. In [17], different types of wind fields which easily affect the UAV are summarized; furthermore, the mechanism of their wind fields affecting the UAV is first strictly analyzed. Next, a novel “reject external disturbance” flight mode for UAV is put forward to offset the trajectory deviation caused by side wind, which makes use of the wind speed information obtained by airspeed and ground speed of UAV. In order to implement the “reject external disturbance” flight mode, the Lyapunov stability theory-based variable model reference adaptive control (VMRAC) system is proposed, and it could also deal with the adverse effects of wind shear and turbulence on UAV flight.

The characteristics and mathematical models of wind, which have a great influence on unmanned aerial vehicles in the low-altitude environment, are summarized in [18]; constant wind, turbulent flow, many types of wind shear, and propeller vortex were considered in this work. Moreover, the mathematical model of the unmanned aerial vehicle was combined with the mechanism of the unmanned aerial vehicle motion in the wind field and was illustrated from three different types of viewpoints, including the velocity viewpoint, the force viewpoint, and the energy viewpoint.

Some simulation tests were implemented to show the effects of different types of wind on the trajectory and flight states of the unmanned aerial vehicle.

The authors of the paper [19] discussed wind formation and quadcopter kinematics and provided cost functions for A* based on distance and wind information. In addition, a collision control method is incorporated into a flight scenario in outdoor terrain under various curvy formations. In [20] a new UAV trajectory planning technique has been proposed, based on the artificial potential field (APF) for following GMTs in windy environments, to provide steady and continuous coverage over the GMT. Therefore, a new modified attractive force to enhance the UAV's sensitivity to wind speed and direction was proposed. The modified wind resistance attractive force function accommodates any small variation of relative displacement caused by wind leading the UAV to drift in a certain direction. This enables the UAV to maintain its position by tilting (i.e., changing its roll and pitch angles) against the wind to retain the camera aim point on the GMT. The proposed path-planning technique is hardware-independent, does not require an anemometer for measuring wind speed and direction, and can be adapted for all types of multirotor UAVs equipped with basic sensors and an autopilot flight controller.

In [21] a new UAV trajectory planning technique has been proposed, based on the artificial potential field (APF) for following GMTs in windy environments, to provide steady and continuous coverage over the GMT, by proposing a new modified attractive force to enhance the UAV's sensitivity to wind speed and direction. The modified wind resistance attractive force function accommodates for any small variation of relative displacement caused by wind leading the UAV to drift in a certain direction. This enables the UAV to maintain its position by tilting (i.e., changing its roll and pitch angles) against the wind to retain the camera aim point on the GMT. The proposed path-planning technique is hardware-independent, does not require an anemometer for measuring wind speed and direction, and can be adopted for all types of multirotor UAVs equipped with basic sensors and an autopilot flight controller. This paper [22] presents a novel UAV path-planning technique, based on the artificial potential field (APF) for following GMTs in windy environments, to provide steady and continuous coverage over the GMT, by proposing a new modified attractive force to enhance the UAV's sensitivity to wind speed and direction .

In the thesis presented in [23], a new bio-inspired algorithm suitable for finding solutions in optimization problems is developed. The Ecological Systems Algorithm (ESA) mimics ecological rules to iteratively find the maximum of a function. This algorithm can be used in static and dynamic search environments. It is used in our thesis to find the gains of different fault-tolerant controllers designed for drones.Sliding mode control is then used to develop two passive fault-tolerant controllers (PFTC) for quadcopters.Because passive fault-tolerant controls have reduced robustness and because they can only handle a small number of faults, an active sliding mode control using a Kalman filter is developed for quadcopters. The fault is estimated in real time and the control is reconfigured accordingly. To deal with extreme situations, an emergency controller is developed for quadcopters subject to such faults. The emergency control device is based on the conversion of the quadcopter into a trirotor, the fault diagnosis and identification unit which estimates faults and serious failures online is based on the estimates of a Kalman filter.

In [24] a control strategy was presented using Proportional Derivative (PD) Controller for the attitude and trajectory control of the Quadrotor (UAV). Whereas the authors in [25] explained the control architecture including vision-based control of the quadrotor; thus, the proposed dynamical model comprised gyroscopic effects and its control strategies.

In [29], the authors do a theoretical analysis of the dynamics of the Draganflyer in order to extract its model which can be used as a computer control system to ensure its stable hovering and indoor flight. The author of article [31] opted to use a control strategy that includes feedback linearization coupled with a PD controller for the translational subsystem and a backstepping-based PID nonlinear controller for the rotational subsystem of the quadrotor. In [32] the results of two nonlinear control techniques applied to a quadrotor, a backstepping, and sliding mode control techniques were presented. After that, the design of an optimal model-free backstepping controller for a MIMO quadrotor helicopter, perturbed by unknown external disturbances, was described in [33]. The proposed method consists of using a model-free-based backstepping controller optimized by a cuckoo search algorithm. The work exposed in [34] was devoted to designing and optimizing an autonomous quadrotor UAV controller. First, the aerial vehicle's dynamic model was presented; then, an optimal backstepping controller (OBC) was suggested. Traditionally, backstepping controller (BC) parameters are often selected arbitrarily. The gravitational search algorithm (GSA) was used in that

work to determine the BC parameter optimum values. In [39] a Backstepping approach was used for the synthesis of tracking errors and Lyapunov functions, a sliding mode controller was developed in order to ensure Lyapunov stability, handle all system nonlinearities, and track the desired trajectories. In [40], a Sliding Mode Control (SMC) approach was designed for the quadrotor in order to stabilize its vertical flight dynamics. The tracking of a helical desired trajectory was investigated for the SMC-controlled Quad rotorcraft.

While in [41], two types of nonlinear controllers for an autonomous quadrotor helicopter were studied. The first one proposed a feedback linearization controller involving high-order derivative terms and turns out to be quite sensitive to sensor noise as well as modeling uncertainty. The second involved a new approach to an adaptive sliding mode controller using input augmentation in order to account for the underactuated property of the helicopter, sensor noise, and uncertainty without using control inputs of large magnitude.

A direct adaptive sliding mode control was developed in [51] for the quadrotor attitude stabilization and altitude trajectory tracking. First, the developed controller was applied without considering disturbances and parameter uncertainties. After that, a centered white Gaussian noise with some parameter uncertainties was added to the considered output vectors, mass, and inertia matrix, respectively. The synthesis of the adaptation laws was based on the positivity and Lyapunov design principle.

Hence, the article [53] proposed employing Wolfram Mathematica to find the optimal PID settings for a quadcopter by minimizing the error integral. The quadcopter control system used four PID controllers: one to control the altitude and three to control the attitude, i.e., the roll, pitch and yaw. The minimization functions available in Wolfram Mathematica are ‘FindMinimum’ and ‘NMinimize’. The main difference between them lies in the fact that one searches for a local minimum and the other is responsible for finding a global minimum of a constrained function.

The authors of [54] constructed the relationship between the attitude and linear acceleration of a small quadrotor unmanned aircraft and proposed a trajectory tracking control design algorithm, based on the relationship, using a new command-filtered backstepping technique to stabilize the attitude; and a linear tracking differentiator to eliminate the classical inner/outer-loop structure. In

[55], a new approach was proposed, by using extended Kalman filter (EKF)—linear Kalman filter (LKF), in a cascaded form, to couple the GPS with INS. GPS raw data were fused with noisy Euler angles coming from the inertial measurement unit (IMU) readings, in order to produce more consistent and accurate real-time navigation information.

In [56], a novel robust backstepping-based approach combined with sliding mode control is proposed for trajectory tracking of a quadrotor UAV subject to external disturbances and parameter uncertainties associated with the presence of aerodynamic forces and possible wind force.

[57] the dynamic model of a 4Y octorotor is derived, and a suitable feedback controller is designed. Moreover, a simple control reconfiguration scheme is proposed to handle the failure of one rotor. Simulation results show the feasibility of the proposed controller.

In [58], a cascaded SMC is developed for quadrotor control. The inner loop responsible of stabilizing altitude and attitude variables is based on feedback linearization control, while the outer loop that controls the position of the quadrotor is based on SMC. Simulation results show good performance of the proposed controller despite the presence of external disturbances.

In [59], a nonlinear adaptive estimator is proposed to improve robustness in the velocity estimation, when only the linear acceleration, the angles and the angular velocity are available for measurement.

I.3. Quadrotor's Configuration

The quadrotor consists of four rotors in a cross-configuration, as shown in Figure I.2. These four actuators are individually controlled to produce a relative thrust. Two of the rotors rotate clockwise, and the other two rotors rotate in counter clockwise direction. This configuration allows the quadrotor to be relatively simple in design yet highly reliable and maneuverable. The dynamic equation of movement of the attitude could be deduced from the Euler equation. The quadrotor mathematical and state-space models are explained in the following subsections.

Quadrotor has six degrees of freedom (DOF), it only has four rotors, making it impossible to achieve a desired set-point for all six DOF. However, because of its structure, it is quite simple to select the four best controllable variables and decouple them to make the controller simpler. The

four quadrotor targets are thus linked to the four fundamental movements that enable the helicopter to achieve a specific height and attitude. To describe the motion of a 6 DOF rigid body it is usual to define two reference frames:

- Earth inertial reference (E-frame)
- Body-fixed reference (B-frame).

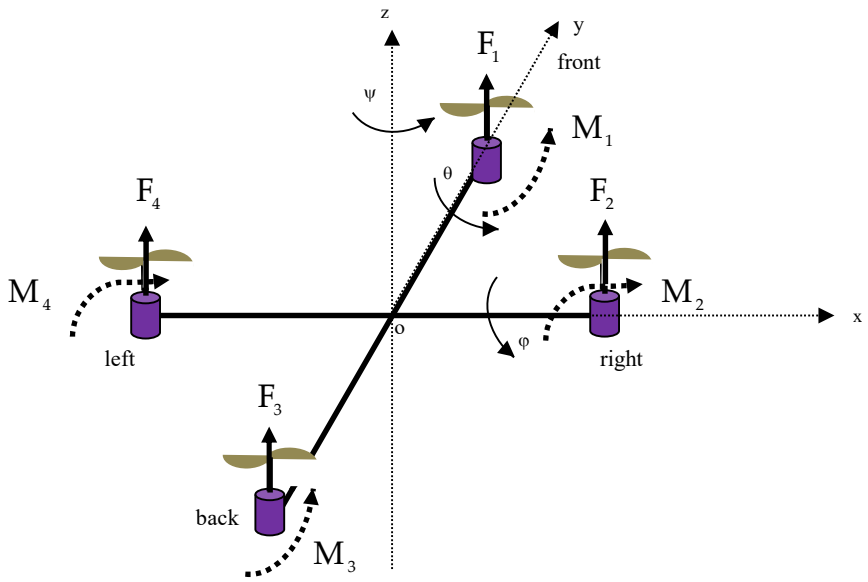


Fig. 1.2. Quadrotor configuration.

The distance between the Earth frame and the body frame indicates the exact position of the center of mass of the quadrotor $r = [x\ y\ z]^T$.The orientation of the quadrotor is indicated by the rotation R from the body frame to the inertial one, and it is described by roll (ϕ), pitch (θ) and yaw (ψ) angles representing rotations about the X, Y and Z-axes respectively. Assuming the order of rotation is pitch, roll then yaw; thus, the rotation matrix R is given by Eq. (1.1) [39]:

$$R = \begin{bmatrix} \cos(\theta)\cos(\psi) & \cos(\psi)\sin(\theta)\sin(\phi) - \cos(\phi)\sin(\psi) & \cos(\phi)\sin(\theta)\cos(\psi) + \sin(\phi)\sin(\psi) \\ \cos(\theta)\sin(\psi) & \sin(\phi)\sin(\theta)\sin(\psi) + \cos(\theta)\cos(\psi) & \cos(\phi)\sin(\theta)\sin(\psi) - \sin(\theta)\cos(\psi) \\ -\sin(\theta) & \sin(\phi)\cos(\theta) & \cos(\phi)\cos(\theta) \end{bmatrix} \quad (1.1)$$

To link between the Euler rates $\dot{\eta} = [\dot{\phi} \quad \dot{\theta} \quad \dot{\psi}]^T$ that are measured in the inertial frame and angular body rates $\omega = [p \quad q \quad r]^T$. The following transformation is needed: $\omega = R_r\dot{\eta}$

Where:

$$R_r = \begin{bmatrix} \cos(\theta) & 0 & -\cos(\phi)\sin(\theta) \\ 0 & 1 & \sin(\phi) \\ \sin(\theta) & 0 & \cos(\phi)\cos(\theta) \end{bmatrix} \quad (1.2)$$

Around the hover position, a small angle assumption is made where $\cos(\phi) \approx \cos(\theta) \approx 1$, and $\sin(\phi) \approx \sin(\theta) \approx 0$. Thus, R_r can be simplified to an identity matrix I [27].

I.3.1. Quadrotor Dynamics

The quadrotor's motion may be split into two subsystems: a rotating subsystem (roll, pitch and yaw) and a translating subsystem (x, y and z coordinates). The revolving subsystem is completely actuated whilst the translating subsystem is under-actuated [27]. Newton Euler Equation can be formulated by Eq. 1.3:

$$\begin{bmatrix} F \\ \tau \end{bmatrix} = \begin{bmatrix} mI_{d3} & O_3 \\ O_3 & I_3 \end{bmatrix} \begin{bmatrix} \mathrm{a} \\ \alpha \end{bmatrix} + \begin{bmatrix} 0 \\ \omega \times I_3\omega \end{bmatrix}, \quad (1.3)$$

Knowing that:

$$I_3 = \begin{bmatrix} I_{xx} & 0 & 0 \\ 0 & I_{yy} & 0 \\ 0 & 0 & I_{zz} \end{bmatrix}$$

With:

F: the net force acting on the quadrotor,

τ: the net torque

I_{d3}: 3 × 3 identity matrix,

I_3: the moment of inertia,

m: the quadrotor mass

a: the linear acceleration of the center of mass,

α: the angular acceleration

Based on the Newton-Euler method, the rotational equations of motion are derived in the body frame with the general formalism Eq. 1.4:

$$I\dot{\omega} + \omega \times I\omega = M_B \quad (1.4)$$

Where:

I: is the inertia matrix of the quadrotor.

ω: is the angular body rate

M_B: are all the moments acting on the quadrotor in the body frame.

$I\dot{\omega}\ and\ \omega \times I\omega$: Represent the rate of change of angular momentum in the body frame.

I.3.2. Moments acting on the quadrotors

Before we define the last term of the equation (1.4) - M_B- , two physical effects have to be defined: the aerodynamic forces and moments generated by a rotor. As an effect of rotation, each rotor generates a force called the aerodynamic force $\boldsymbol{F_i}$ and a moment called the aerodynamic moment $\boldsymbol{M_i}$. And are given by:

$$F_i = \frac{1}{2}\rho A C_T r^2 \Omega_i^2 \tag{1.5}$$

$$M_i = \frac{1}{2}\rho A C_D r^2 \Omega_i^2 \tag{1.6}$$

Where:

ρ: is the air density.

A: blade area.

C_D, C_T: aerodynamic coefficients.

r: the radius of blade.

Ω_i: angular velocity of rotor i.

The geometry of the propeller and the density of the air are the major parameters that the aerodynamic force and moment depends on. In case of a quadrotor the maximum altitude is limited, thus the air density can be considered as constant. As a result, Eqs. (1.5) and (1.6) can be simplified to the following [8]:

$$F_i = K_f \Omega_i^2 \tag{1.7}$$

$$M_i = K_M \Omega_i^2 \tag{1.8}$$

Where K_f and K_M are the aerodynamic force and moment constants respectively and Ωi is the angular velocity of rotor i.

Figure I.2 shows the forces and moments acting and produced on each rotor of the quadcopter.

It is clear that each rotor produces an upwards thrust force F_i and generates a moment M_i that has an opposite direction to the directions of the corresponding rotor.

Starting with the moments about the body frame's x-axis. By using the right-hand rule in association with the axes of the body frame, $F2$ multiplied by the moment arm l generates a negative moment about the x-axis, in the same manner, $F4$ generates a positive moment. Hence, the total moment about the x-axis can be expressed as [1]:

$$M_x = -F_2 l + F_4 l = -\left(K_f \Omega_2^2\right) l + \left(K_f \Omega_4^2\right) l$$

$$M_x = l K_f (-\Omega_2^2 + \Omega_4^2) \tag{1.9}$$

Where l is the arm length or the distance between the center of mass of the quadrotor and the axis of rotation of each rotor.

With the same manner the moments about the y-axis are expressed as:

$$M_y = -F_1 l + F_3 l = -\left(K_f \Omega_1^2\right) l + \left(K_f \Omega_3^2\right) l$$

$$M_y = K_f (\Omega_1^2 - \Omega_3^2) \tag{1.10}$$

About the z-axis, using the right-hand rule, the moments are expressed as:

$$M_z = -M_1 + M_2 - M_3 + M_4$$

$$M_z = -(K_M \Omega_1^2) + (K_M \Omega_2^2) - (K_M \Omega_3^2) + (K_M \Omega_4^2)$$

$$M_z = K_M (-\Omega_1^2 + \Omega_2^2 - \Omega_3^2 + \Omega_4^2) \tag{1.11}$$

Finally, by combining the equations M_B can be easily found and is given by following:

$$M_B = \begin{bmatrix} l K_f (-\Omega_2^2 + \Omega_4^2) \\ l K_f (\Omega_1^2 - \Omega_3^2) \\ K_M (-\Omega_1^2 + \Omega_2^2 - \Omega_3^2 + \Omega_4^2) \end{bmatrix} \tag{1.12}$$

Where: l:represents the distance from the center of the quadcopter to any of the propellers.

The dynamic model of quadrotor can be defined in terms of the position vector and forces expressions as given in Eqs. (1.13) and (1.14); knowing that:

$$m\begin{bmatrix} x'' \\ y'' \\ z'' \end{bmatrix} = \begin{bmatrix} 0 \\ 0 \\ -mg \end{bmatrix} + \begin{bmatrix} (\sin(\phi)\sin(\psi) + \cos(\phi)\cos(\psi)\sin(\theta))U_1 \\ (\cos(\phi)\sin(\psi)\sin(\theta) - \cos(\psi)\sin(\phi))U_1 \\ \cos(\phi)\cos(\theta)\,U_1 \end{bmatrix}$$

Thus:

$$\begin{cases} x^{"} = -\frac{U_1}{m}(\sin(\varphi)\sin(\psi) + \cos(\varphi)\cos(\theta)\cos(\psi)) \\ y^{"} = -\frac{U_1}{m}(\cos(\varphi)\sin(\theta)\sin(\psi) - \sin(\psi)\cos(\theta)) \\ z^{"} = g - \frac{U_1}{m}(\cos(\theta)\cos(\varphi)) \end{cases} \quad (1.13)$$

Where, $(x^{"}, y^{"}, z^{"})$ represents the second derivative of position vector, Where U_1 is the resulting upward thrust of the four propellers and m denotes the mass.

Input terms are defined as U_1 is the normalized total lift force, and $U_\varphi, U_\theta,$ and U_ψ correspond to the control inputs of roll, pitch and yaw moments, respectively (Eq. (1.14)).

$$\begin{cases} u_1 = b(F_1 + F_2 + F_3 + F_4) \\ u_\varphi = lb(\ F_2 - F_4\ ) \\ u_\theta = lb(\ F_3 - F_1\ ) \\ u_\psi = d(F_1 - F_2 + F_3 - F_4) \end{cases} \quad (1.14)$$

The moment equations can be expressed in terms of the orientation angles (φ, θ, ψ): Roll, Pitch and Yaw respectively as given in Eq. (1.15) and Eq. (1.16).

$$\begin{cases} p' = \frac{I_z - I_y}{I_x} qr - \frac{J_r}{I_x} q\Omega + \frac{1}{I_x} U_\varphi \\ q' = \frac{I_z - I_x}{I_y} pr - \frac{J_r}{I_y} p\Omega + \frac{1}{I_y} U_\theta \\ r' = \frac{I_y - I_x}{I_z} pq + \frac{1}{I_z} U_\psi \end{cases} \quad (1.15)$$

$$\begin{cases} \varphi' = p + qsin(\varphi)tg(\theta) + rcos(\varphi)tg(\theta) \\ \theta' = qcos(\varphi) - rsin(\varphi) \\ \psi' = q\frac{\sin(\varphi)}{\cos(\theta)} + r\frac{\cos(\varphi)}{\cos(\theta)} \end{cases} \quad (1.16)$$

$U_\varphi, U_\theta, U_\psi$ represents the total rolling, pitching and yawing torques; whereas, p, q and r represent the angular velocities in the body frame.

I.4. State-space Model

A state space representation is a mathematical model of a physical system as a set of inputs, outputs and state variables related by first order differential equations. "State space" refers to the space whose axes are the state variables. The state of the system can be represented as a vector within that space.

In this work the state space model of the quadrotor in the inertial frame is developed. Thus, the dynamic model of the quadrotor in the inertial frame can be expressed in the form:

$$\dot{X} = f(X, U)$$

By the system referred as Eq. (1.17):

$$\begin{cases} x_1' = x_2 \\ x_2' = a_1 x_6 x_4 + \Omega a_3 x_4 + b_1 U_\varphi \\ x_3' = x_4 \\ x_4' = a_4 x_2 x_6 + \Omega a_6 x_2 + b_2 U_\theta \\ x_5' = x_6 \\ x_6' = a_7 x_2 x_4 + b_3 U_\psi \\ x_7' = x_8 \\ x_8' = a_9 x_8 + \frac{\cos(x_1)\cos(x_3)}{m} U_1 - g \\ x_9' = x_{10} \\ x_{10}' = a_{10} x_{10} + \frac{U_1}{m} U_y \\ x_{11}' = x_{12} \\ x_{12}' = a_{11} x_{12} + \frac{U_1}{m} U_x \end{cases} \quad (1.17)$$

The parameters a_1, a_3, a_4, a_6, b_1, b_2 and b_3 can be calculated as follow:

$$\begin{cases} a_1 = \dfrac{I_y - I_z}{I_x} \\ a_3 = \dfrac{J_r}{I_x} \\ a_4 = \dfrac{I_z - I_x}{I_y} \\ a_6 = \dfrac{J_r}{I_y} \\ b_1 = \dfrac{I_x - I_y}{I_z} \\ b_2 = \dfrac{d}{I_y} \\ b_3 = \dfrac{d}{I_z} \end{cases}$$

Ix, Iy and Iz denote the inertias of the x-axis, y-axis and z-axis of the Quadrotor, respectively, Jr denotes the z-axis inertia of the propellers' rotors.

Whereas,

$$\begin{cases} U_x = \cos(x_1)\cos(x_3)\cos(x_5) + \sin(x_1)\sin(x_5) \\ U_y = \cos(x_1)\sin(x_3)\sin(x_5) - si\,\mathrm{n}(x_5)\ \cos(x_3) \end{cases}$$

I.5. Conclusion

In order to find the equation of state of the quadrotor, newton's first and second law are used to found six differential equations that manage the operation of quadrotor (system has six degrees of freedom 6 DOF); at the end, the equation of quadrotor is obtained which is constituted of twelve states. The latter will be used to simulate the operation of the quadrotor.

Chapter 2: Nonlinear control of quadrotor

In this chapter, three nonlinear controllers are implemented in Matlab Simulink starting by proportional derivative controller (PDC) beside two other controllers: backstepping controller (BSC) sliding mode controller (SMC); in order to study the performance of each of them in quadrotor trajectory tracking. Thus, the obtained results are compared and conclusions are extracted.

II.1. PID Control of Quadrotor

From the model obtained in the previous chapter, we will realize the control of the quadrotor using the PID regulators. To overcome the problem of nonlinearity of the system and to control the positions x,y through the attitudes φ and θ, we use the PI-D regulator to control the angles and PD regulator to control the positions. The general control strategy is shown in the figure 2.1.

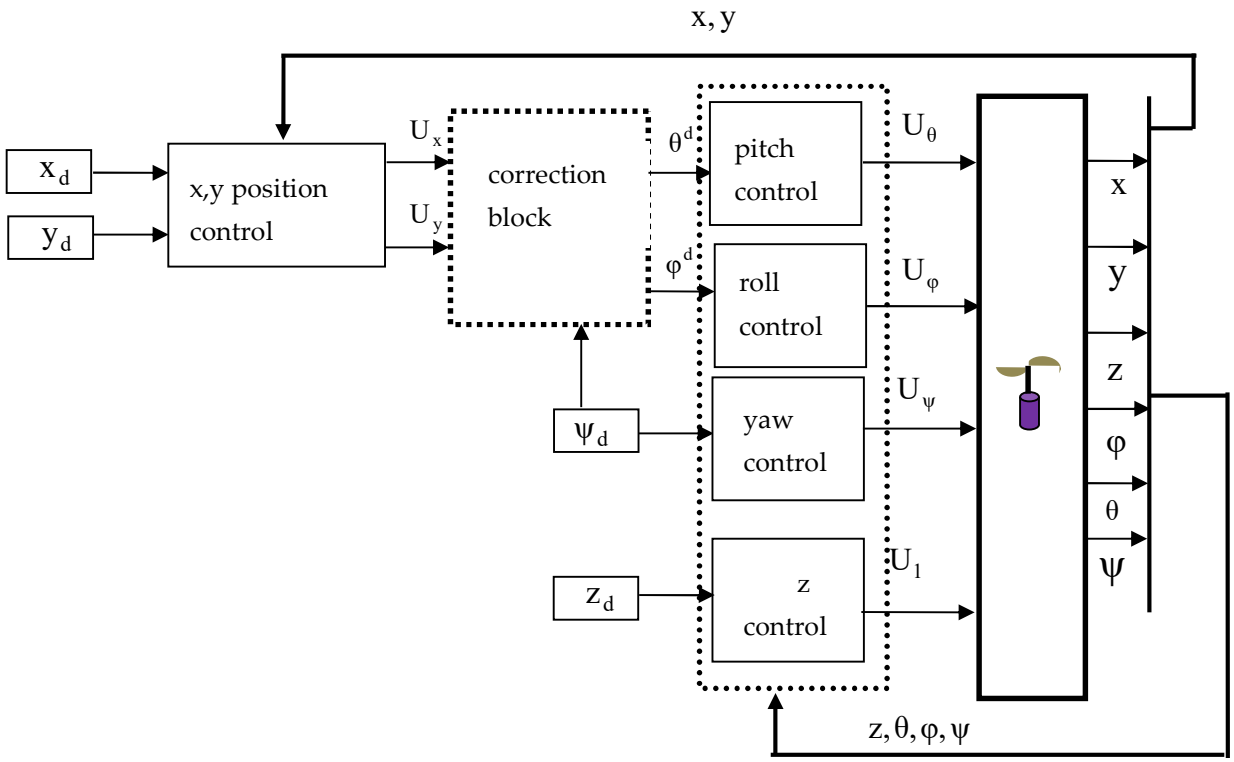


Fig. 2.1. Quadrotor control strategy.

The desired angles θ^d and φ^d are given by the relations (2.1) and (2.3) indicated in [54] as follows:

$$\theta^d = arctg\left(\frac{U_x\cos(\psi)+U_y\sin(\psi)}{U_z+g}\right) \quad (2.1)$$

$$\varphi^d = arsin\left(m\frac{U_x\sin(\varphi)-U_y\cos(\psi)}{U_1}\right) \quad (2.2)$$

$$U_1 = \sqrt{U_x^2+U_y^2+U_z^2} \quad (2.3)$$

II.1.1. Control design

II.1.1.1. Altitude z-stabilization

A PD controller is used to the z-position control. The vertical input position will be obtained from the following relationship:

$$U_z = K_{pz}(z^d - z) - K_{dz}(z'^d - z') \quad (2.4)$$

II.1.1.2. Yaw (ψ)-controller

To control the yaw angle we use a PI-D controller, and the torque is given by the relation given in Eq. 2.5:

$$U_\psi = K_{p\psi}(\psi^d - \psi) + K_{i\psi}\int(\psi^d - \psi)dt - K_{d\psi}\psi' \quad (2.5)$$

II.1.1.3. Roll attitude hold

Using the equation of motion $\varphi'' = \frac{1}{I_x}U_\varphi$ and the structure of the PI-D controller shown in figure 2.2, the transfer function can be extracted:

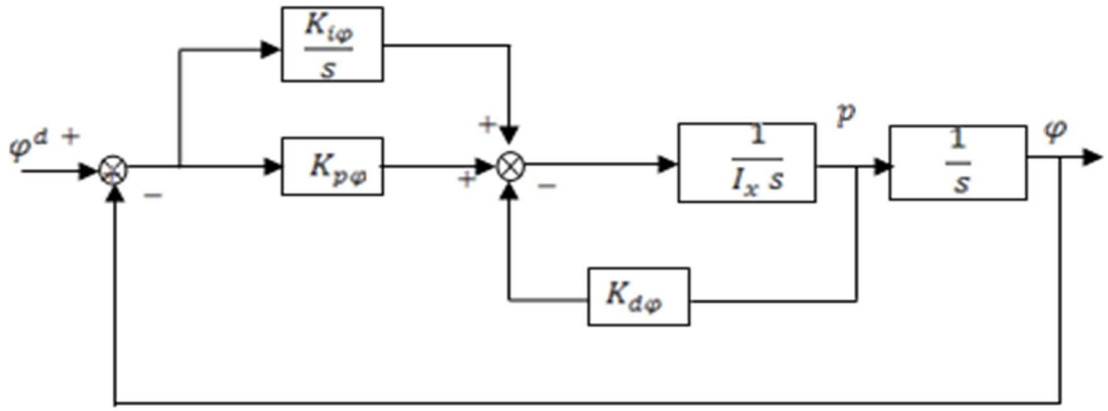


Fig. 2.2. A block diagram of the roll attitude hold loop.

Therefore, the transfer function of this structure is expressed as:

$$TF = \frac{\frac{K_{i\varphi}}{I_x}}{s^3+\frac{K_{d\varphi}}{I_x}s^2+\frac{K_{p\varphi}}{I_x}s+\frac{K_{i\varphi}}{I_x}} \tag{2.6}$$

Which can have the same response with ordinary second order transfer function $\frac{\omega_n^2}{(s^2+2\xi\omega_n s+\omega_n^2)}$ .his characteristic equation can multiplied by $(s-(-p))$

$(s+p)(s^2+2\xi\omega_n s+\omega_n^2) = s^3+(2\xi\omega_n+p)s^2+(2\xi\omega_n p+\omega_n^2)s+p\omega_n^2$

and identified with the characteristic equation of controller as follows:

$$s^3+(2\xi\omega_n+p)s^2+(2\xi\omega_n p+\omega_n^2)s+p\omega_n^2 = s^3+\frac{K_{d\varphi}}{I_x}s^2+\frac{K_{p\varphi}}{I_x}s+\frac{K_{i\varphi}}{I_x}$$

$$\text{With } \begin{cases} K_{i\varphi} = p\omega_n^2 I_x; \\ K_{p\varphi} = (2\xi\omega_n p+\omega_n^2)I_x; \\ K_{d\varphi} = (2\xi\omega_n+p)I_x \end{cases} \tag{2.7}$$

For $\omega_n = \frac{10rd}{s}$, $\xi = 0.9$, and p=20

$$\begin{cases} K_{i\varphi} = 2000\ I_x; \\ K_{p\varphi} = 460\ I_x \\ K_{d\varphi} = 38 I_x \end{cases}$$

II.1.1.4. Pitch attitude control

Using the equation of motion $\theta'' = \frac{1}{I_y} U_\theta$ and the structure of the PI-D controller, the same results of controller are obtained:

$$\begin{cases} K_{i\theta} = p\omega_n^2 I_y \ ; \\ K_{p\theta} = (2\xi\omega_n p + \omega_n^2) I_y; \\ K_{d\theta} = (2\xi\omega_n + p) I_y \end{cases} \quad (2.8)$$

II.1.1.5. Positions x and y control

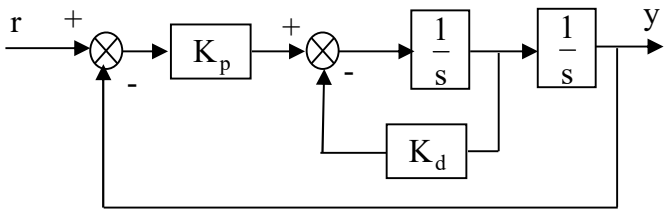


Fig. 2.3. Standard PD loop.

Using the transfer function $\frac{r}{y} = \frac{K_p}{s^2 + K_d s + K_p}$, which can identify with a second order system $\frac{\omega_n^2}{s^2 + 2\xi\omega_n s + \omega_n^2}$, we deduce that $K_p = \omega_n^2$ and $K_d = 2\xi\omega_n$.

For example $\omega_n = 10 rd/s \ and \ \xi = 0.9$, we obtain $K_p = 100 \ \ and \ \ K_d = 18$. The controller gains are shown in Table 2.1.

Table 2.1. Controller gains (PID-C)

	x	y	z	φ	θ	ψ
K_p	100	100	100	$460 I_x$	$460 I_y$	$460 I_z$
K_i	0	0	0	$2000 I_x$	$2000 I_y$	$2000 I_z$
K_d	18	18	18	$38 I_x$	$38 I_y$	$38 I_z$

II.1.2. PID controller results and discussions

The simulation results are shown in Fig. 2.4 - Fig. 2.7, the control inputs of rotors are presented in Fig. 2.4. The obtained control inputs commands could easily be applied to the real model. , it can be seen from Fig. 2.5 that the speeds of the rotors are of the order of 250rd/s. Fig. 2.6 show the rotational speed of the quadrotor and the responses of attitudes .in Fig.2.7 show that the quadrotor must follow a square trajectory at a height of ten meters by setting the yaw angle at 0 radium.

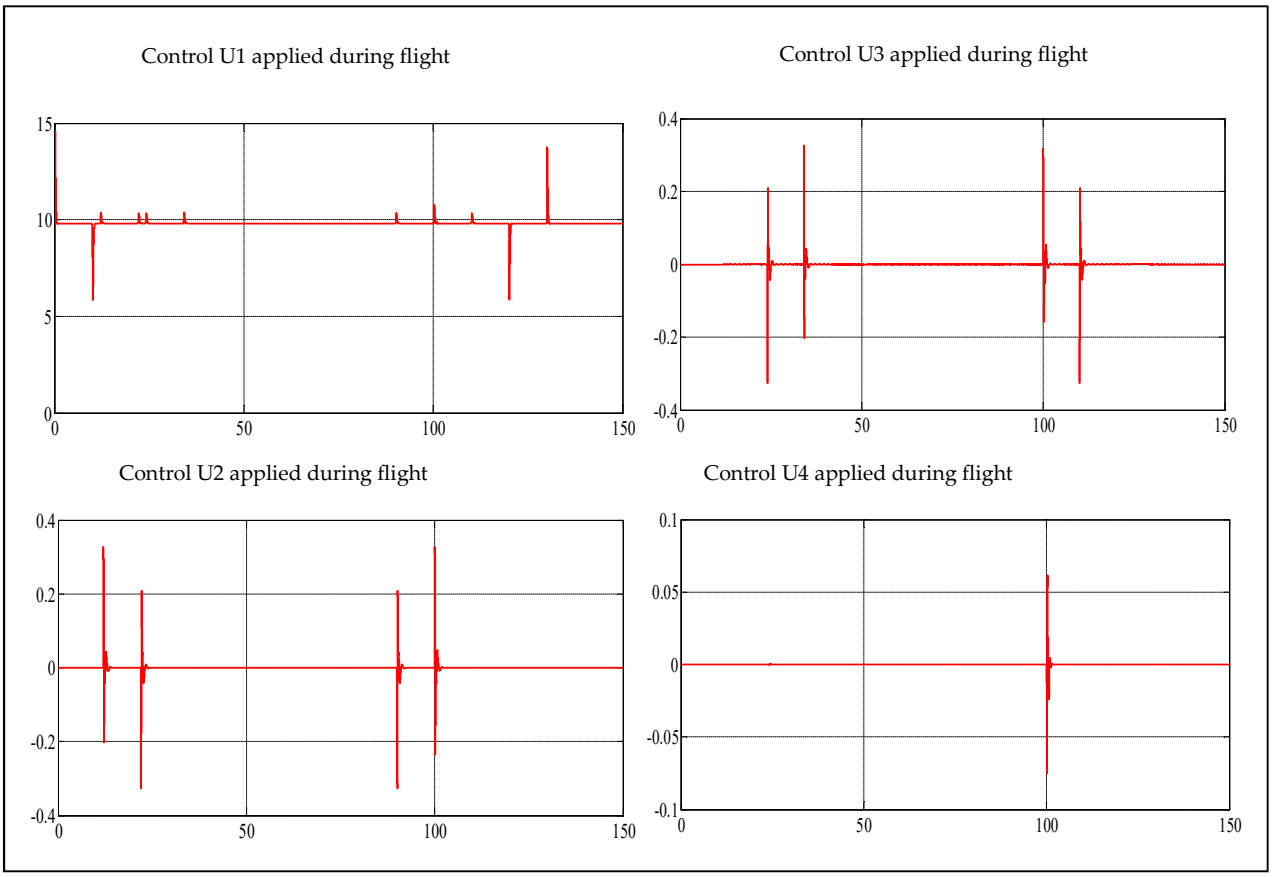


Fig.2.4. Inputs generated by controllers during simulation.

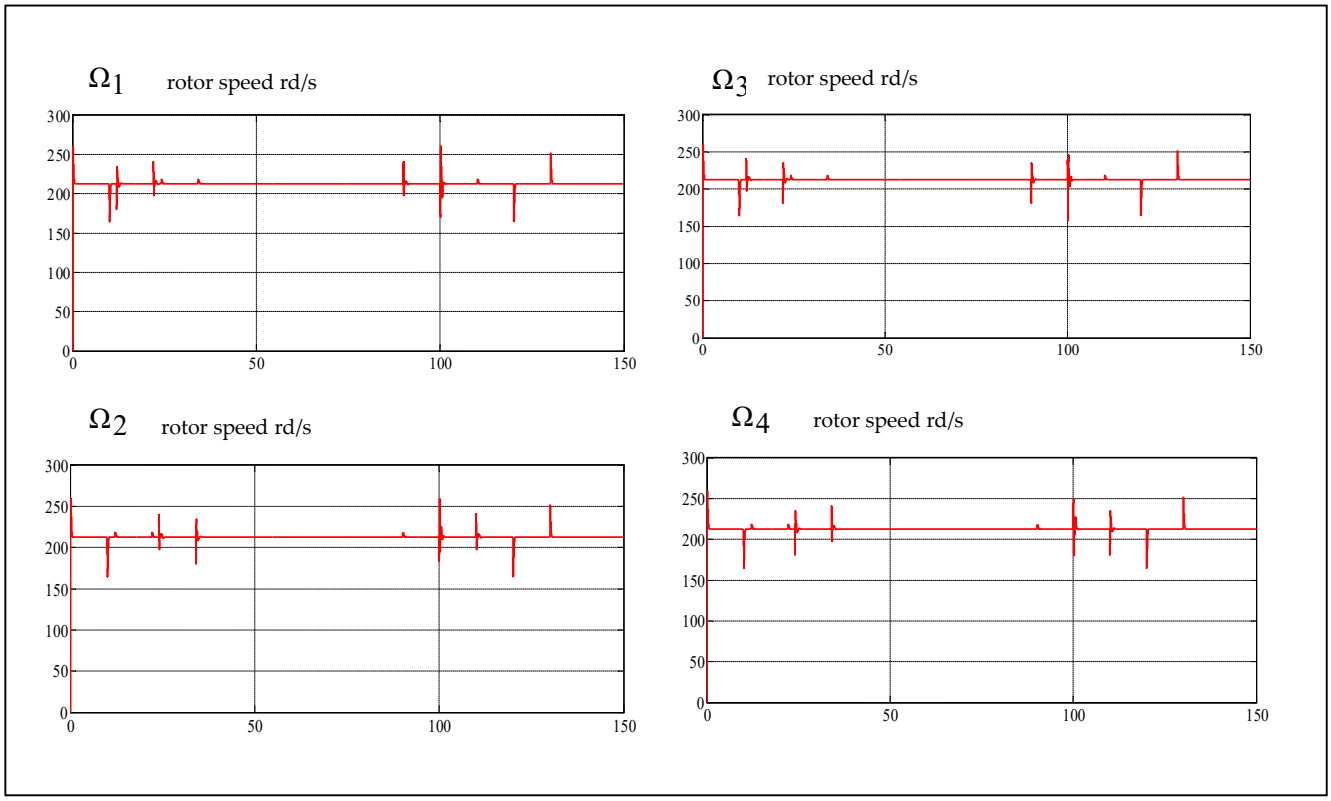


Fig. 2.5. Rotor speeds.

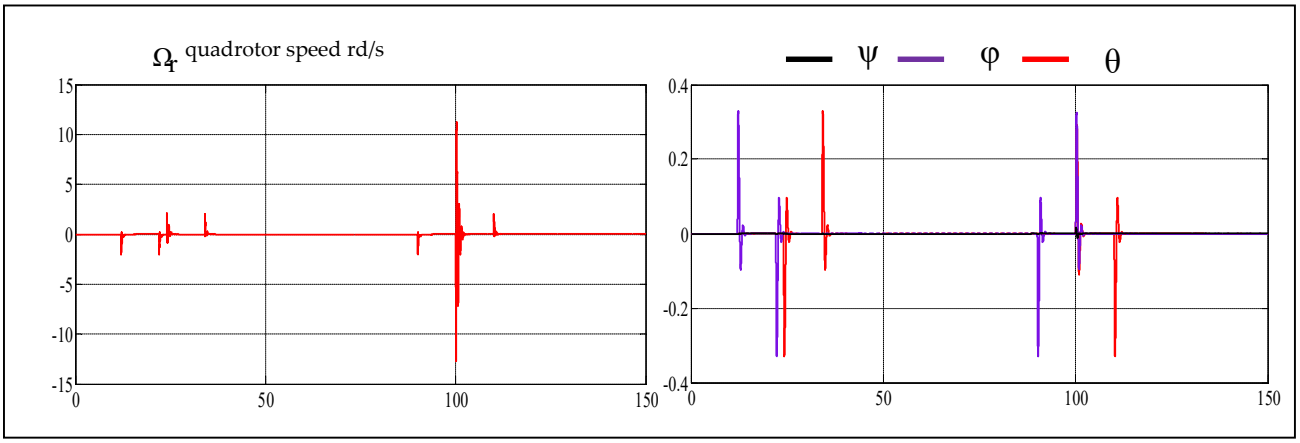


Fig. 2.6. Quadrotor speed and angles responses.

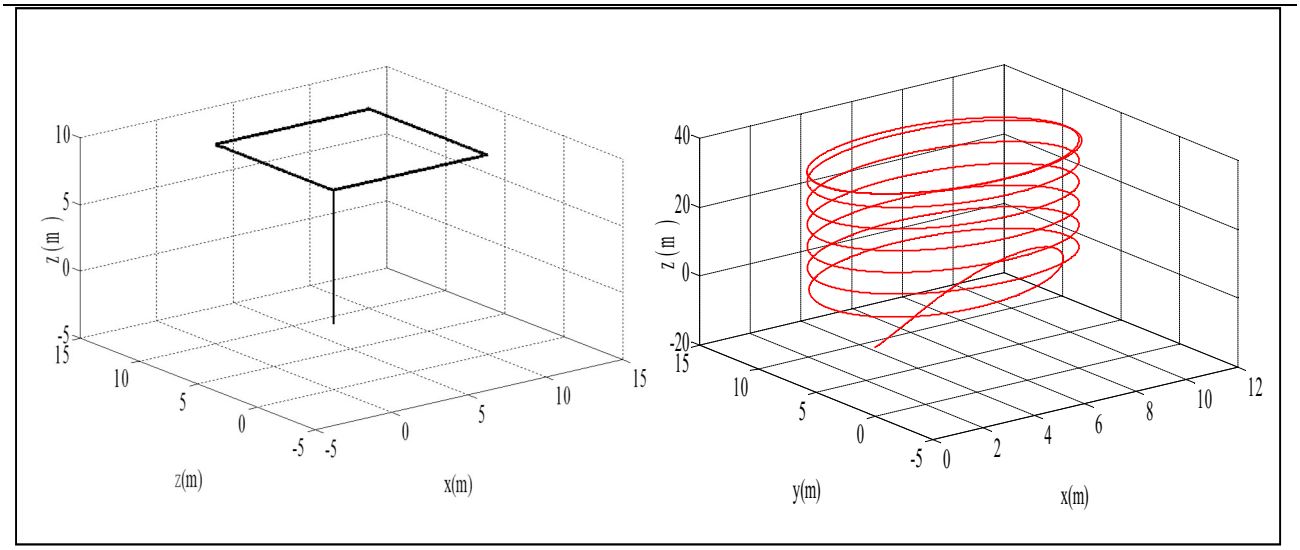


Fig. 2.7. Path of quadrotor controlled with PID (in left square path, in the right helical path).

In this section, the control of quadrotor using PID regulators was realized. The simulation results show the effectiveness of this control technique despite the fact that is difficult to size the regulators because of the nonlinearity of the quadrotor model. In order to overcome this problem and enrich our research topic, nonlinear commands will be used in the coming sections.

II.2. Backstepping Control

Backstteping is a nonlinear control method, in backstepping design, no constraint is imposed on the nonlinear characteristic of the system, which must be in the pure parametric form. The basic of backstepping algorithmes are explained below.

II.2.1. Backstepping algorithm

Considering the case of nonlinear systems of the form presented by Eq. 2.9:

$$\begin{cases} x_1' = \varphi_1^t(x_1)\theta + \psi_1(x_1)x_2 \\ x_2' = \varphi_2^t(x_1, x_2)\theta + \psi_2(x_1x_2,)x_3 \\ x_3' = \varphi_3^t(x_1, x_2, x_3)\theta + \psi_3(x_1x_2, x_3)u \end{cases} \tag{2.9}$$

The vector of parameters θ is assumed to be known, the number of steps of backstepping algorithm is equal to the number of states of the system. The corresponding steps are given below:

step1

We first consider equation (2.8)

$$x_1' = \varphi_1^t(x_1)\theta + \psi_1(x_1)x_2 \tag{2.10}$$

Knowing that, the state variable x_2 is treated as a command. Thus, the first desired value is defined as:

$$\alpha^0 = y^r = x_1^d \tag{2.11}$$

The first error variable is defined by ε_1 defined by Eq. 2.12

$$\varepsilon_1 = \alpha^0 - x_1 = x_1^d - x_1 \tag{2.12}$$

Which implies:

$$\varepsilon_1' = \alpha^{0\prime} - x_1' = x_1^{d'} - x_1' = -\varphi_1^t(x_1)\theta - \psi_1(x_1)x_2 \tag{2.13}$$

The quadratic function $V(\varepsilon_1) = \frac{1}{2}\varepsilon_1^2$ constitutes a good choice of the Lyapunov candidate function. Hence, its derivative, yields the solution of Eq. 2.13, and is given by :

$$V'(\varepsilon_1) = \varepsilon_1\varepsilon_1' = \varepsilon_1(\alpha^{0'} - \varphi_1^t(x_1)\theta - \psi_1(x_1)x_2) \tag{2.14}$$

A wise choice of x_2 would make $V'(\varepsilon_1)$ negative and ensure the stability of the origin of the subsystem:

$$\alpha^{0'} - \varphi_1^t(x_1)\theta - \psi_1(x_1)x_2 = -K_1\varepsilon_1 \tag{2.15}$$

Therefore:

$$x_2{}^d = \frac{1}{\psi_1(x_1)}(\alpha^{0'} - \varphi_1^t(x_1)\theta + K_1\varepsilon_1) = \alpha^1 \tag{2.16}$$

step2

We first consider the subsystem

$$\begin{cases} x_1' = \varphi_1^t(x_1)\theta + \psi_1(x_1)x_2 \\ x_2' = \varphi_2^t(x_1, x_2)\theta + \psi_2(x_1x_2,)x_3 \end{cases} \quad (2.17)$$

Where the state variable x_3 is treated as a command. Thus, the second error variable is defined by ε_2 as follows:

$$\varepsilon_2 = \alpha^1 - x_2 = x_2^d - x_2 \quad (2.18)$$

Consequently, its derivative can be written as:

$$\varepsilon_2' = \alpha^{1\prime} - x_2' = x_2^{d\prime} - x_2' = \alpha^{1'} - \varphi_2^t(x_1, x_2)\theta - \psi_2(x_1x_2,)x_3 \quad (2.19)$$

The Lyapunov candidate function is

$$V(\varepsilon_1, \varepsilon_2) = V_1 + \frac{1}{2}\varepsilon_2^2 \quad (2.20)$$

and its derivate can calculated as follows:

$V'(\varepsilon_1, \varepsilon_2) = \varepsilon_1\varepsilon_1' + \varepsilon_2\varepsilon_2'$ from Eq.2.13 and Eq.2.19 we obtain Eq.2.21

$$V'(\varepsilon_1, \varepsilon_2) = -K_1\varepsilon_1^2 + \varepsilon_2\left(\alpha^{1'} - \varphi_2^t(x_1, x_2)\theta - \psi_2(x_1x_2,)x_3 - \varepsilon_1\right) \quad (2.21)$$

The choice of the desired value of x_3 becomes obvious. The latter is given by Eq. 2.22 as explained below:

$$\left(\alpha^{1'} - \varphi_2^t(x_1, x_2)\theta - \psi_2(x_1x_2,)x_3 - \varepsilon_1\right) = -K_2\varepsilon_2$$

$$x_3{}^d = \frac{1}{\psi_2(x_1x_2,)}(\alpha^{1'} - \varphi_2^t(x_1, x_2)\theta + \psi_1(x_1)\varepsilon_1 + K_2\varepsilon_2) = \alpha^{(2)} \quad (2.22)$$

With k2 > 0, and α_1' calculated analytically such that:

$$\alpha_1' = \frac{\partial\alpha}{\partial x_1}\frac{dx_1}{dt}$$

step3

The system is now considered as a whole. The error variable ε_3 is defined as:

$$\varepsilon_3 = \alpha^2 - x_3 = x_3^d - x_3 \tag{2.23}$$

Thus, makes it possible to write the equations of the system, in the error space as explained in the following equations:

$$\varepsilon_1' = \alpha^{0'} - \varphi_1^t(x_1)\theta - \psi_1(x_1)(\alpha^1 - \varepsilon_2) \tag{2.24}$$

$$\varepsilon_2' = \alpha^{1'} - \varphi_2^t(x_1, x_2)\theta - \psi_2(x_1 x_2,)(\alpha^2 - \varepsilon_3) \tag{2.25}$$

$$\varepsilon_3' = \alpha^{2'} - \varphi_3^t(x_1, x_2, x_3)\theta - \psi_3(x_1 x_2, x_3)u \tag{2.26}$$

With as lyapunov function defined as:

$$V(\varepsilon_1, \varepsilon_2, \varepsilon_3) = V_2 + \frac{1}{2}\varepsilon_3^2 \tag{2.27}$$

Its derivative, along the solutions of (2.25) and (2.26), becomes:

$$V'(\varepsilon_1, \varepsilon_2, \varepsilon_3) = V_2' + \varepsilon_3\varepsilon_3' = -K_1\varepsilon_1^2 - K_2\varepsilon_2^2 + \varepsilon_3\big(\alpha^{2'} - \varphi_3^t(x_1, x_2, x_3)\theta + \psi_2(x_1 x_2,)\varepsilon_2 - \psi_3(x_1 x_2, x_3)u\big) = -K_1\varepsilon_1^2 - K_2\varepsilon_2^2 - K_3\varepsilon_3^2$$

which leads to $$\alpha^{2'} - \varphi_3^t(x_1, x_2, x_3)\theta + \psi_2(x_1 x_2,)\varepsilon_2 - \psi_3(x_1 x_2, x_3)u = -K_3\varepsilon_3 \tag{2.28}$$

Now, we are in the presence of the real order . A good choice of it is given by Eq. 2.29 as shown below:

$$u = \frac{1}{\psi_3(x_1 x_2, x_3)}(\alpha^{2'} - \varphi_3^t(x_1, x_2, x_3)\theta + \psi_2(x_1 x_2,)\varepsilon_2 + K_3\varepsilon_3) \tag{2.29}$$

With α_2'calculated analytically as follow:

$$\alpha_2' = \frac{\partial \alpha_2}{\partial x_1}.\frac{dx_1}{dt} + \frac{\partial \alpha_2}{\partial x_2}.\frac{dx_2}{dt} \tag{2.30}$$

II.2.2. Backstepping control of quadrotors

The principle of backstepping is to divide the system into several sub-systems in cascade. The control laws are then made for each subsystem, in a decreasing manner, until a global control law for the whole system is generated.

II.2.2.1. Roll angle Control φ

Considering the first subsystem mentioned below:

$$\begin{cases} x_1' = x_2 \\ x_2' = a_1 x_6 x_4 - \Omega a_2 x_4 + b_1 U_\varphi \end{cases} \quad (2.31)$$

step1

The error ε_1 between the desired and actual roll angle is expressed as follow:

$$\varepsilon_1 = x_1^d - x_1 \quad (2.32)$$

Consider the Lyapunov function $V_1 = \frac{1}{2}\varepsilon_1^2$ (2.33)

The derivate of V_1 along x_1 trajectory, V', is computed as follow:With:

$$V_1' = \varepsilon_1 \varepsilon_1' \quad (2.34)$$

$$\varepsilon_1' = x_1^{d\prime} - x_1' = \varepsilon_1' = x_1^{d\prime} - x_2 \quad (2.35)$$

Choosing:

$\varepsilon_1^{'} = -K_1 \varepsilon_1$ (where: $K_1 \varepsilon_1^2$ positive definite function).

The desired X_2^d is extracted:

$$x_2^d = x_1^{d\prime} + K_1 \varepsilon_1 \quad (2.36)$$

step2

Denoting ε_2 the error between desired and actual roll angle rate, so that:

$$\varepsilon_2 = x_2^d - x_2 \quad (2.37)$$

its derivate is $\varepsilon_2' = x_2^{d'} - a_1 x_6 x_4 + \Omega a_2 x_4 - b_1 U_\varphi$ (2.38)

Using candidate Lyapunov function

$$V_2 = \frac{1}{2}\varepsilon_2^2 + V_1$$

$$V_2' = V_1' + \varepsilon_2 \varepsilon_2'$$

$$V_2' = \varepsilon_1 \varepsilon_1' + \varepsilon_2 \varepsilon_2' \quad (2.39)$$

using the Eq.2.35 and Eq.2.38 , the expression of the derivate of the Lyapunov candidate function will be:

$$V_2' = \varepsilon_1 (x_1^{d\prime} - x_2) + \varepsilon_2 (x_2^{d'} - a_1 x_6 x_4 + \Omega a_2 x_4 - b_1 U_\varphi \ )$$

$$= \varepsilon_1 (x_1^{d\prime} + \varepsilon_2 - x_2^{d}) + \varepsilon_2 (x_2^{d'} - a_1 x_6 x_4 + \Omega a_2 x_4 - b_1 U_\varphi \ )$$

$$= -k_1 \varepsilon_1^2 + \varepsilon_2 (\varepsilon_1 + x_2^{d'} - a_1 x_6 x_4 + \Omega a_2 x_4 - b_1 U_\varphi \ )$$

to satisfy Lyaponv's stability condition we take

$$\varepsilon_2 (\varepsilon_1 + x_2^{d'} - a_1 x_6 x_4 + \Omega a_2 x_4 - b_1 U_\varphi \ ) = -K_2 \varepsilon_2^2 \quad (2.40)$$

Which gives:

$$(\varepsilon_1 + x_2^{d'} - a_1 x_6 x_4 + \Omega a_2 x_4 - b_1 U_\varphi \ ) = -K_2 \varepsilon_2 \quad (2.41)$$

Where: k_2 is a positive constant

$x_2^{d'} = -K_1 x_2$

Thus, the control law is: $U_\varphi = \frac{1}{b_1}(\varepsilon_1 - a_1 x_6 x_4 + K_2 \varepsilon_2 - K_3 x_4 + \Omega a_2 x_4)$ (2.42)

II.2.2.2.Pitch angle control θ

Considering the second subsystem mentioned below:

$$\begin{cases} x_3' = x_4 \\ x_4' = a_4 x_2 x_6 + \Omega a_6 x_2 + b_2 U_\theta \end{cases} \quad (2.43)$$

step1

Considering ε_3 is the error between the desired and actual angle θ and which can be found by:

$$\varepsilon_3 = x_3^d - x_3 \quad (2.44)$$

and

$$\varepsilon_3 = x_3^d - x_3 \quad (2.44)$$

$$\varepsilon_3' = x_3^{d\prime} - x_3' = x_3^{d\prime} - x_4 \quad (2.45)$$

Using Lyapunov stability by choosing $V_3 = \frac{1}{2}\varepsilon_3^2$.

If *V'* is negative; then, the system trajectory is ensured to verify this condition:

$$V_3' = \varepsilon_3 \varepsilon_3' = \varepsilon(x_3^{d'} - x_4) < 0 \quad (2.46)$$

$$x_4^d = x_3^{d\prime} + K_3 \varepsilon_3 \quad (2.47)$$

step2

The error $\varepsilon_4 = x_4^d - x_4$ gives

$$\varepsilon_4' = x_4^{d\prime} - x_4' \quad (2.48)$$

The lyapunov function is defined as: $V_4 = \frac{1}{2}\varepsilon_4^2 + V_3$

Thus: $V_4' = V_3' + \varepsilon_4 \varepsilon_4' = \varepsilon_3 \varepsilon_3' + \varepsilon_4 \varepsilon_4'$

from where $V_{4=}' \varepsilon_3 (x_3^{d\prime} - x_4) + \varepsilon_4 (x_4^{d'} - a_3 x_2 x_6 - \Omega a_4 x_2 - b_2 U_{\theta_1})$

Therefore:

$$U_\theta = \frac{1}{b_2}(\varepsilon_3 - a_3 x_6 x_2 + K_4 \varepsilon_4 - K_3 x_4 - \Omega a_4 x_2) \tag{2.49}$$

II.2.2.3 Yaw angle control ψ

Now, consider the third subsystem mentioned below:

$$\begin{cases} x_5' = x_6 \\ x_6' = a_7 x_2 x_4 + b_3 U_\psi \end{cases} \tag{2.50}$$

step1

Let's ε_5 the error between the desired and actual angle ψ

$\varepsilon_5 = x_5^d - x_5$ and $\varepsilon_5' = x_5^{d\prime} - x_6' = x_5^{d\prime} - x_4$

With, Lyapunov function is $V_5 = \frac{1}{2}\varepsilon_5^2$

where $V_5' = \varepsilon_5 \varepsilon_5' = \varepsilon_5(x_5^{d'} - x_6) < 0$

Thus; $x_6^d = x_5^{d\prime} + K_5 \varepsilon_5$

step2

The error $\varepsilon_6 = x_6^d - x_6$ and $\varepsilon_6' = x_6^{d\prime} - x_6'$ and the candidate lyapunov function $V_6 = \frac{1}{2}\varepsilon_6^2 + V_5$

Thus:

$V_6' = V_5' + \varepsilon_6 \varepsilon_6'$

$= \varepsilon_5 \varepsilon_5' + \varepsilon_6 \varepsilon_6'$

$= \varepsilon_5(x_5^{d\prime} - x_6^d + \varepsilon_6) + \varepsilon_6(x_6^{d'} - a_7 x_2 x_4 - b_3 U_\psi)$

$= -K_6 \varepsilon_6^2$

$$U_\psi = \frac{1}{b_3}(\varepsilon_5 - a_7 x_2 x_4 + K_6\varepsilon_6 - K_5 x_6) \quad (2.51)$$

II.2.2.4. Control of the position z

The fourth subsystem is:

$$\begin{cases} x_7' = x_8 \\ x_8' = a_9 x_8 + \frac{cos(x_1)cos(x_3)}{m} U_1 - g \end{cases} \quad (2.52)$$

step1

ε_7 is the error between the desired and actual position z: $\varepsilon_7 = x_7^d - x_7$

and $\varepsilon_7' = x_7^{d\prime} - x_7' = x_7^{d\prime} - x_8$

The Lyapunov function is $V_7 = \frac{1}{2}\varepsilon_7^2$ and its derivate is:

$V_7' = \varepsilon_7\varepsilon_7' = \varepsilon_7(x_7^{d'} - x_8) < 0,$

Then $\left(x_7^{d'} - x_8\right) = -K_7\varepsilon_7$

The desired value: $x_8^d = x_7^{d\prime} + K_7\varepsilon_7$

step2

The error $\varepsilon_8 = x_8^d - x_8$ gives $\varepsilon_8' = x_8^{d\prime} - x_8'$, and the lyapunov function is:

$V_8 = \frac{1}{2}\varepsilon_8^2 + V_7$

Thus $V_8' = V_7' + \varepsilon_8\varepsilon_8' = \varepsilon_7\varepsilon_7' + \varepsilon_8\varepsilon_8'$

$$= \varepsilon_7\left(x_7^{d'} - x_8^d\right) + \varepsilon_8(x_8^{d'} + \varepsilon_7 - (a_9 x_8 + \frac{\cos(x_1)\cos(x_3)}{m} U_1 - g\ )$$

Then:

$$\left(x_8^{d'}+\varepsilon_7-\left(a_9x_8+\frac{\cos(x_1)\cos(x_3)}{m}U_1-g\right)\right)=-K_8\varepsilon_8$$

The control law:

$$U_1=\frac{m}{\cos(x_1)\cos(x_3)}(\varepsilon_7-a_9x_8+g+K_8\varepsilon_8-K_7x_8) \qquad (2.53)$$

II.2.2.5. Control of the position y

The fifth subsystem is:

$$\begin{cases} x_9'=x_{10} \\ x_{10}'=\dfrac{\mathrm{U}_y}{m}U_1 \end{cases} \qquad (2.54)$$

step1

ε_9 is the error between the desired and actual position y such that:

$$\varepsilon_9=x_9^d-x_9 \text{ and}$$

$$\varepsilon_9'=x_9^{d\prime}-x_9'=x_9^{d\prime}-x_9$$

The Lyapunov function is $V_9=\frac{1}{2}\varepsilon_9^2$; whereas, its derivate is:

$$V_9'=\varepsilon_9\varepsilon_9'=\varepsilon_9(x_9^{d'}-x_9)<0$$

Then,

$$\left(x_9^{d'}-x_{10}\right)=-K_9\varepsilon_9$$

So the desired value is:

$$x_{10}^d=x_9^{d\prime}+K_9\varepsilon_9$$

step2

The error will be: $\varepsilon_{10}=x_{10}^d-x_{10}$ give $\varepsilon_{10}'=x_{10}^{d\prime}-x_{10}'$

The lyapunov function $V_{10} = \frac{1}{2}\varepsilon_{10}^2 + V_9$; so,

$$V'_{10} = V'_9 + \varepsilon_{10}\varepsilon'_{10}$$

$$= \varepsilon_9\varepsilon'_9 + \varepsilon_{10}\varepsilon'_{10}$$

$$= \varepsilon_9(x_9^{d'} - x_{10}^d + \varepsilon_{10}) + \varepsilon_{10}(x_{10}^{d'} - (\frac{U_y}{m}U_1))$$

$$= \varepsilon_9\left(x_9^{d'} - x_{10}^d\right) + \varepsilon_{10}(x_{10}^{d'} + \varepsilon_9 - (\frac{U_y}{m}U_1))$$

$$\left(x_{10}^{d'} + \varepsilon_9 - \left(\frac{U_y}{m}U_1\right)\right) = -K_{10}\varepsilon_{10}$$

The control law:

$$U_y = \frac{m}{U_1}(\varepsilon_9 + K_{10}\varepsilon_{10} - K_9 x_{10}) \qquad (2.56)$$

II.2.2.6.Control of the position x

The last subsystem is represented by the equation below:

$$\begin{cases} x'_{11} = x_{12} \\ x'_{12} = \dfrac{\mathrm{U}_x}{m}U_1 \end{cases} \qquad (2.57)$$

step1

Where, ε_{11} is the error between the desired and actual position x

$$\varepsilon_{11} = x_{11}^d - x_{11} \quad \text{and} \quad \varepsilon'_{11} = x_{11}^{d\prime} - x'_{11} = x_{11}^{d\prime} - x_{11}$$

The Lyapunov function is $V_{11} = \frac{1}{2}\varepsilon_{11}^2$; its derivate

$$V'_{11} = \varepsilon_{11}\varepsilon'_{11} = \varepsilon_{11}(x_{11}^{d'} - x_{11}) < 0$$

Then, $\left(x_{11}^{d'} - x_{12}\right) = -K_{11}\varepsilon_{11}$ and the desired value will be:

$$x_{12}^{d} = x_{11}^{d'} + K_{11}\varepsilon_{11}$$

step2

The error is given by: $\varepsilon_{12} = x_{12}^{d} - x_{12}$; the derivate is $\varepsilon'_{12} = x_{12}^{d'} - x'_{12}$

The lyapunov function $V_{12} = \frac{1}{2}\varepsilon_{12}^2 + V_{11}$ its derivate:

$$V'_{12} = V'_{11} + \varepsilon_{12}\varepsilon'_{12}$$

$$= \varepsilon_{11}\varepsilon'_{11} + \varepsilon_{12}\varepsilon'_{12}$$

$$= \varepsilon_{11}(x_{11}^{d'} - x_{12}^{d} + \varepsilon_{12}) + \varepsilon_{12}(x_{12}^{d'} - (\frac{U_x}{m}U_1))$$

$$= \varepsilon_{11}\left(x_{11}^{d'} - x_{12}^{d}\right) + \varepsilon_{12}(x_{12}^{d'} + \varepsilon_{11} - (\frac{U_x}{m}U_1))$$

Then:

$$\left(x_{12}^{d'} + \varepsilon_{11} - \left(\frac{U_x}{m}U_1\right)\right) = -K_{12}\varepsilon_{12}$$

The control law:

$$U_x = \frac{m}{U_1}(\varepsilon_{11} + K_{12}\varepsilon_{12} - K_{11}x_{12}) \qquad (2.58)$$

All the previous steps of backstepping control used to generate a global control law for the whole system are summarized in the block diagram shown in Figure 2.8. Whereas, the control gains are given in Table 2.2.

Table 2.2: controller gains (BSC)

m	1	3	5	7	9	11
	$\boldsymbol{\varphi}$	$\boldsymbol{\theta}$	$\boldsymbol{\psi}$	$\boldsymbol{z}$	$\boldsymbol{y}$	$\boldsymbol{x}$
$\boldsymbol{K_m}$	15	15	45	5	10	10
$\boldsymbol{K_{m+1}}$	15	15	45	5	5	10

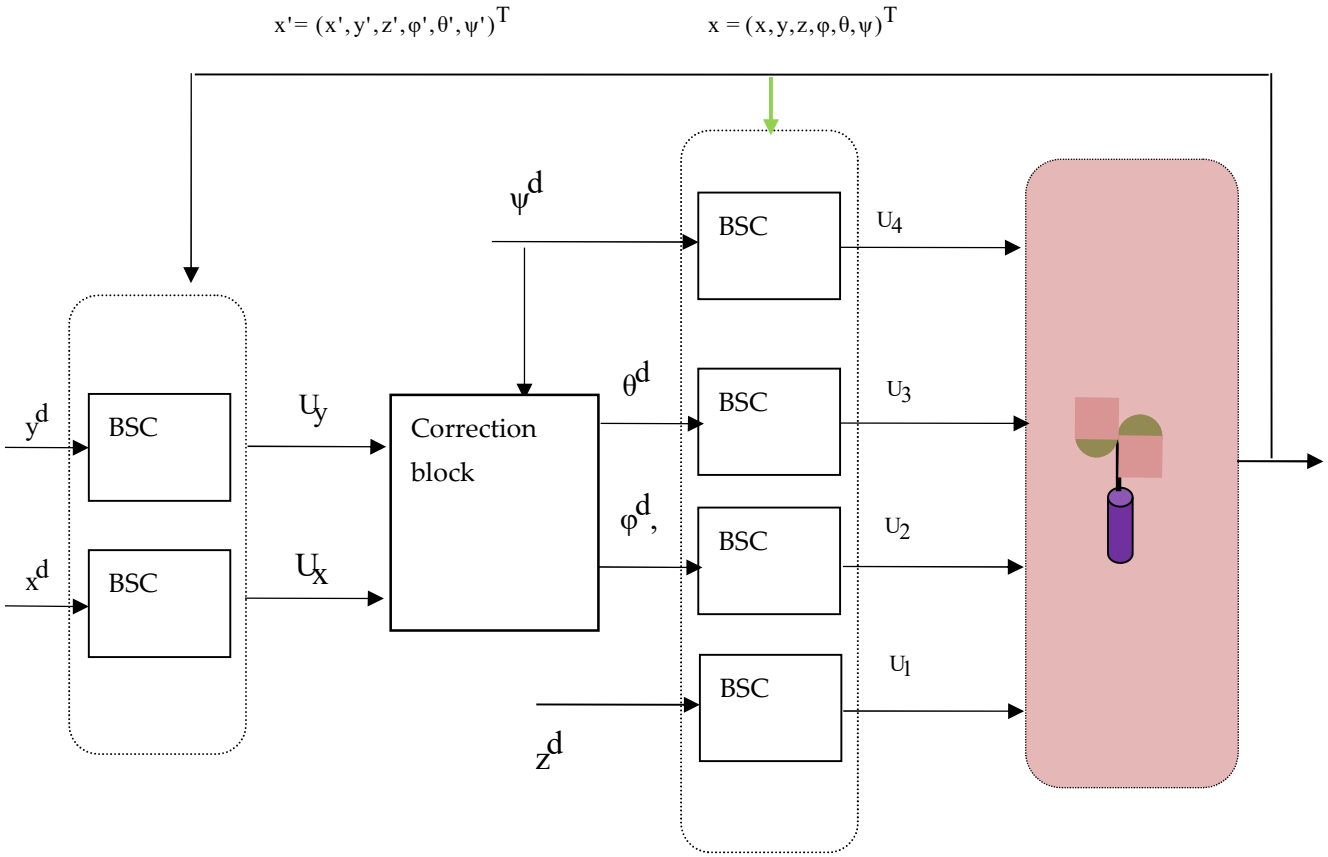


Fig. 2.8. Quadrotor backstepping control structure

II.2.3. BSC results and discussions

In order to validate our proposed control solution, the model is simulated under Matlab Simulink software. For that purpose, the results are obtained based on the application of the real parameters summarized on Table 2.1 [1]. In this scenario, it is desired to follow a circular trajectory in XY

plane, centered in the origin. The height z increases uniformly from zero to 15 meters where the drone stabilizes.

NW : Because the speeds of the real quadrotor motors might not be accessible during the flight, the term Ω_r *might not be performed. Consequently, this term cannot be used in the calculation of the control laws and may cause stability issues.*

Figure 2.10 shows the response of rotor speeds. Figure 2.11 illustrates the inputs generated by controllers during flight. Figure 2.12 illustrates that the real and the desired positions exactly meet each other in three-dimensional space. Figure 2.11.d illustrates the response of orientation angles (roll, pitch, and yaw), where the dotted lines denote the desired values and continues-lines shows the estimated values. It is clearly demonstrated that the estimated values track the desired trajectories with an acceptable dynamic.

In this part we realize the control by backstepping of the quadrotor, this control technique makes possible to control firstly a state of the system and secondly the successive derivatives of the state, we noticed that simulation results showed the effectiveness of the proposed control.

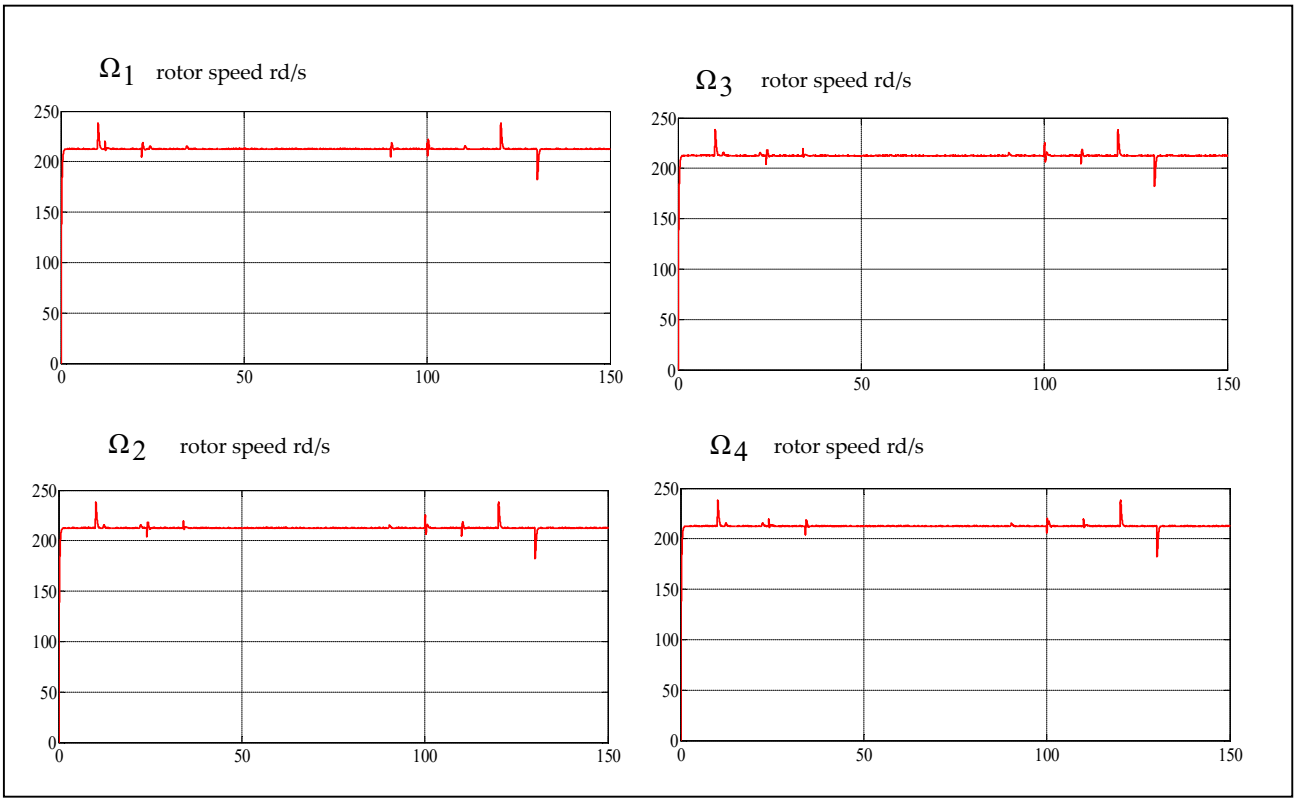


Fig. 2.9. Rotor speeds using BSC.

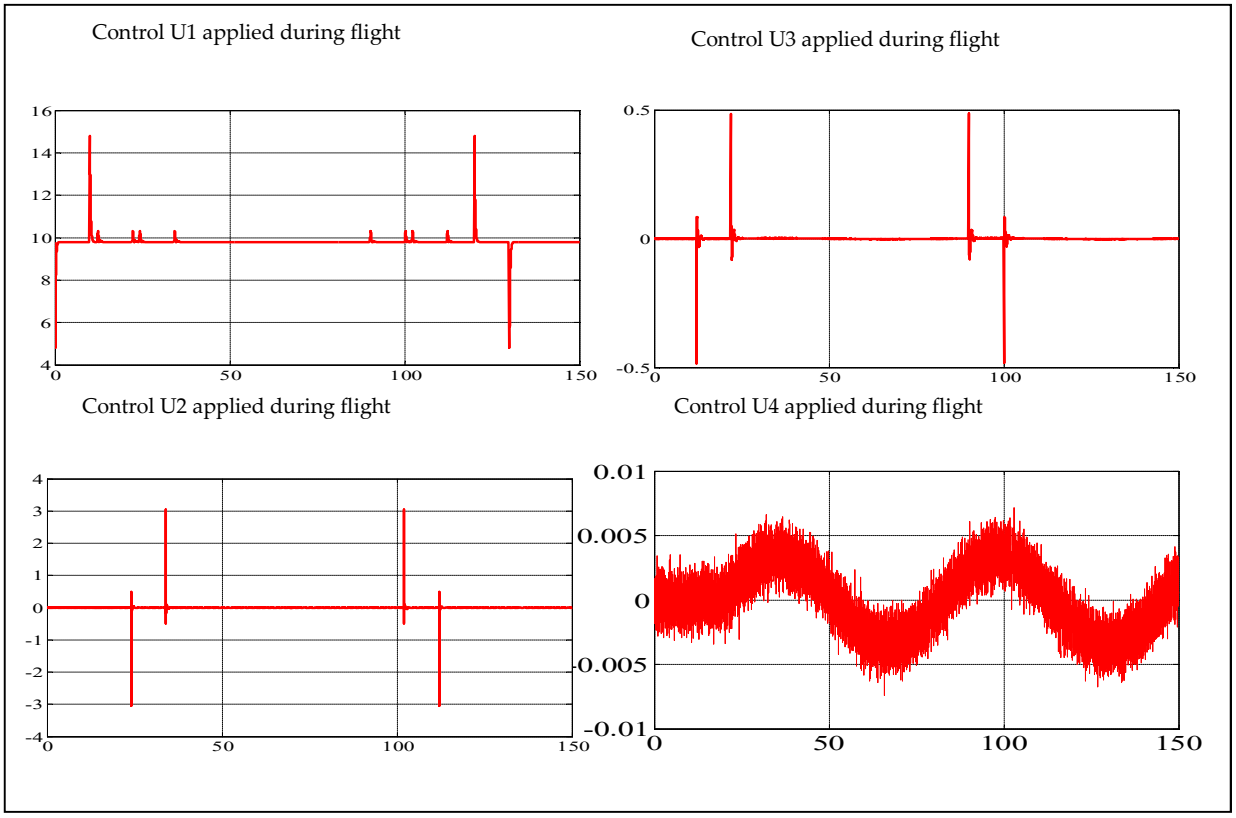


Fig.2.10. Inputs generated by controllers during BSC simulation.

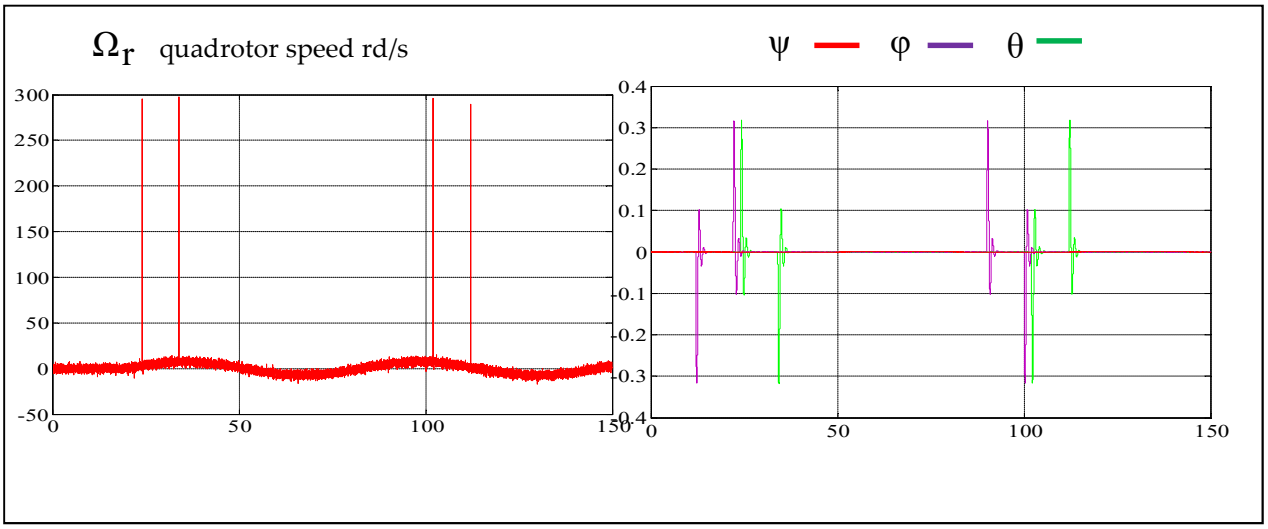


Fig.2.11. Quadrotor speed and angles responses (BSC).

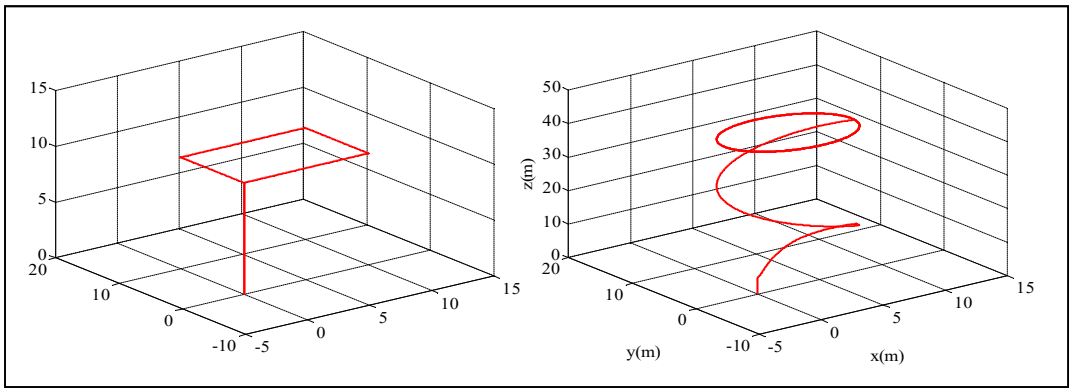


Fig. 2.12. Path of quadrotor controlled with BSC (in left square path, in the right helical path).

II.3. Sliding Mode Control of Quadrotors

This part describes the following points: First, the theory of sliding mode control (**SMC)** is presented, and then the control of quadrotor by **SMC** is treated. Finally, we interpret obtained the results.

II.3.1. Design of sliding mode control

The design of sliding mode controllers considers stability and good performance issues systematically in its approach, which is divided into three main stages:

1- Choice of surfaces,

2- The establishment of the conditions of existence and convergence,

3- Determination of control law.

II.3.1.1 Choice of sliding surfaces

The choice of sliding surface concerns the number needed as well as the shape, depending on the application and the intended purpose. In general, for a system defined by the following state equation:

$$x'(t) = f(x) + g(x)U \tag{2.59}$$

As for the general form, we propose a form of general equation to determine the sliding surface which ensures the convergence of a state variable **x** towards its set value .

$$s(x) = \left(\frac{d}{dt} + \lambda\right)^{r-1} \mathrm{e(x)} \tag{2.60}$$

e(x): the error of the variable to be regulated,

λ: positive constant,

r: relative degree is the smallest positive integer such that

II.3.1.2 Conditions of existence of convergence

The convergence conditions allow the dynamics of the system, in the phase plane, to converge towards the sliding surface, we cite two conditions

The direct switching condition

This is the first convergence condition, it is in the form: $s(x)s'(x) < 0$

The Lyapunov function,This involves formulating a positive scalar function for the state variables of the system and choosing a control law that will cause this function to decrease $V'(x) < 0$.

By defining the Lyapunov function:

$$V(s,x,t) = \frac{1}{2}\, s^2(x,t) \tag{2.61}$$

Its derivative will be: $V'(x) = S(x).S'(x)$

For the Lyapunov function to decrease, it suffices to ensure that: $\dot{V}(s,x,t) = \dot{V}(s) = s.\dot{s} < 0$

it is used to estimate the performance of the control, the study of robustness and guarantees the stability of the nonlinear system

II.3.1.3. - Determination of control law

The sliding mode control includes two terms which are equivalent control term and switching control term:

$$U(t) = U_s(t) + U_{eq}(t) \tag{2.62}$$

II.3.1.3.a. Equivalent control

The equivalent control is a control which, applied to the system, produces the movement of the system on the sliding surface whenever the initial state is on the surface. Suppose that the trajectory

of the state meets the surface of the commutation at time t1 and that a sliding mode exists. The existence of a sliding mode implies that, for all

The system is represented in the following form

$$x'(t) = f(x) + g(x)U \tag{2.63}$$

$U_{eq}(t)$ is the equivalent part of the sliding mode control, i.e., the necessary known part of the control system when $\dot{s} = 0$.

We define the equivalent control U_{eq} as the vector that satisfies

$$\dot{s} = \frac{dS}{dt} = \frac{\partial S}{\partial x}.\frac{dx}{dt} = \frac{\partial S}{\partial x}(f(x) + g(x)U) = \frac{\partial S}{\partial x} f(x) + \frac{\partial S}{\partial x} g(x)U = 0 \tag{2.64}$$

the equivalent control is:

$$U_{eq} = -\left(\frac{\partial S}{\partial x} g(x)\right)^{-1}\left(\frac{\partial S}{\partial x} f(x)\right) \tag{2.65}$$

II.3.1.3.b. Sliding control

$U_s(t)$ is the sliding control mode defined as:

$$\dot{V}(s,x,t) = \dot{V}(s) = s.\dot{s} = s\left(\frac{\partial S}{\partial x} f(x) + \frac{\partial S}{\partial x} g(x)\left(U_{eq} + U_s\right)\right) < 0$$

$$s\left(\frac{\partial S}{\partial x} f(x) + \frac{\partial S}{\partial x} g(x)U_{eq} + \frac{\partial S}{\partial x} g(x)U_s\right) < 0 \tag{2.66}$$

With eq(2.64) Then: $s\left(\frac{\partial S}{\partial x} g(x)U_s\right) < 0$

Where $sgn(s)$ is the mathematical signum function defined as:

$$U_s = -Ksgn(s) * sgn\left(\frac{\partial S}{\partial x} g(x)\right) \tag{2.67}$$

$$U_s = \begin{cases} -Ksgn(s) & if \ \frac{\partial S}{\partial x} g(x) > 0 \\ Ksgn(s) & if \ \frac{\partial S}{\partial x} g(x) < 0 \end{cases} \tag{2.68}$$

II.3.2. Sliding mode control of quadrotors

Considering the following system:

$$\begin{cases} x_1' = x_2 \\ x_2' = a_1 x_6 x_4 + \Omega a_3 x_4 + b_1 U_\varphi \\ x_3' = x_4 \\ x_4' = a_4 x_2 x_6 + \Omega a_6 x_2 + b_2 U_\theta \\ x_5' = x_6 \\ x_6' = a_7 x_2 x_4 + b_3 U_\psi \\ x_7' = x_8 \\ x_8' = a_9 x_8 + \dfrac{cos(x_1)\, cos(x_3)}{m} U_1 - g \\ x_9' = x_{10} \\ x_{10}' = a_{10} x_{10} + \dfrac{U_1}{m} U_y \\ x_{11}' = x_{12} \\ x_{12}' = a_{11} x_{12} + \dfrac{U_1}{m} U_x \end{cases} \tag{2.68}$$

II.3.2.1. Control of the x position

The degree r equal 2:

$$s(x) = \left(\frac{d}{dt} + \lambda\right)^{r-1} e(x) = e'(x) + \lambda e(x), e(x) = x - x^d \tag{2.69}$$

$$s(x) = = x' - x^{d'} + \lambda(x - x^d) = x_2 - x^{d\prime} + \lambda(x - x^d) \tag{2.70}$$

The control U_{eff} using eq(2.65)

$$\frac{\partial s}{\partial x} g(x) = [\lambda \quad 1] \begin{bmatrix} 0 \\ \frac{1}{m} \end{bmatrix} = \frac{1}{m}$$

and $\frac{\partial s}{\partial x} f(x) = [\lambda \quad 1] \begin{bmatrix} x_2 \\ \frac{-f_x}{m} x_2 \end{bmatrix} = \lambda x_2 - \frac{f_x}{m} x_2$

$$U_{eq} = m\left(-\lambda x_2 + \frac{f_x}{m} x_2\right) \tag{2.71}$$

The Uatt $U_s = -Ksgn(s(x)) \ because \ \frac{\partial s}{\partial x} g(x) > 0$

$$\text{The control} \quad U_{x-control} = m\left(-\lambda x_2 + \frac{f_x}{m} x_2\right) - K_x sgn\big(s(x)\big) \tag{2.72}$$

II.3.2.2. Control of the y position

The degree r equal 2,

$$s(y) = \left(\frac{d}{dt} + \lambda\right)^{r-1} e(y) = e'(y) + \lambda e(y), e(y) = y - y^d \tag{2.73}$$

$$s(y) = = y' - y^{d\prime} + \lambda(y - y^d) = y - y^{d\prime} + \lambda(y - y^d)$$

The control Ueff

$$U_{eq} = -\left(\frac{\partial s}{\partial x} g(y)\right)^{-1} \left(\frac{\partial s}{\partial x} f(y)\right)$$

$\frac{\partial s}{\partial x} g(y) = [\lambda \quad 1] \begin{bmatrix} 0 \\ \frac{1}{m} \end{bmatrix} = \frac{1}{m}$

and $\frac{\partial s}{\partial x} f(y) = [\lambda \quad 1] \begin{bmatrix} y_2 \\ \frac{-f_y}{m} y_2 \end{bmatrix} = \lambda y_2 - \frac{f_y}{m} y_2$

$$U_{eq} = m\left(-\lambda y_2 + \frac{f_y}{m} y_2\right)$$

The Uatt :

$$U_s = -Ksgn(s(y)\ because\ \frac{\partial s}{\partial x} g(y) > 0$$

The control law:

$$U_{y-control} = m\left(-\lambda y_2 + \frac{f_y}{m} y_2\right) - K_y sgn\big(s(y)\big) \tag{2.74}$$

II.3.2.3. Control of the z position

The degree r equal 2,

$$s(z) = \left(\frac{d}{dt} + \lambda\right)^{r-1} e(z) = e'(z) + \lambda e(z), e(z) = z - z^d \tag{2.75}$$

$$s(z) = z' - z^{d\prime} + \lambda(z - z^d) = z_2 - z^{d\prime} + \lambda(z - z^d)$$

The control Ueff:

$$U_{eq} = -\left(\frac{\partial s}{\partial x} g(z)\right)^{-1} \left(\frac{\partial s}{\partial x} f(z)\right) \tag{2.76}$$

$\frac{\partial s}{\partial x} g(z) = [\lambda \quad 1] \begin{bmatrix} 0 \\ \frac{1}{m} \end{bmatrix} = \frac{1}{m}$

and $\frac{\partial s}{\partial x} f(z) = [\lambda \quad 1] \begin{bmatrix} z_2 \\ \frac{-f_z}{m} z_2 \end{bmatrix} = \lambda z_2 - \frac{f_z}{m} z_2$

$U_{eq} = m\left(-\lambda z_2 + \frac{f_y}{m} z_2 - g\right)$

The Uatt $U_s = -Ksgn(s(z)\ because\ \frac{\partial s}{\partial x} g(z) > 0$

The control law: $U_{z-control} = -m\left(-\lambda z_2 + \frac{f_z}{m} z_2 - g\right) - K_z sgn(s(z))$ (2.77)

II.3.2.4. Control of the φ direction

Considering these equations: $\begin{cases} \varphi_1' = \varphi_2 \\ \varphi_2' = a_1 \theta' \psi' + a_3 \Omega \theta' - a_{13} \varphi_2^2 + b_1 U_2 \end{cases}$ (2.78)

The degree r equal 2,

$$s(\varphi) = \left(\frac{d}{dt} + \lambda_\varphi\right)^{r-1} e(\varphi) = e'(\varphi) + \lambda_\varphi e(\varphi), e(\varphi) = \phi^d - \varphi$$

$$s(\varphi) = \varphi^{d'} - \varphi' + \lambda_\varphi(\varphi^d - \varphi)$$

$$= \varphi^{d\prime} - \varphi_2 + \lambda_\varphi(\varphi^d - \varphi_1)$$

Thus: $U_{eq} = -\left(\frac{\partial s}{\partial x} g(\varphi)\right)^{-1} \left(\frac{\partial s}{\partial x} f(\varphi)\right)$

$$\frac{\partial s}{\partial x} g(\varphi) = [-\lambda_\varphi \quad -1] \begin{bmatrix} 0 \\ \frac{-1}{I_x} \end{bmatrix} = \frac{1}{I_x}$$

and $\frac{\partial s}{\partial x} f(\varphi) = [-\lambda_\varphi \quad -1] \begin{bmatrix} \varphi_2 \\ a_1 \theta' \psi' - a_{13} \varphi_2^2 + a_3 \Omega \theta' \end{bmatrix} = -\lambda_\varphi \varphi_2 - a_1 \theta' \psi' - a_3 \Omega \theta' + a_{13} \varphi_2^2$

$$U_{eq} = -I_x\left(-\lambda_\varphi \varphi_2 - a_1 \theta' \psi' - a_3 \Omega \theta' + a_{13} \varphi_2^2\right)$$

The Uatt $U_s = -Ksgn(s(\varphi)\ because\ \frac{\partial s}{\partial x} g(\varphi) > 0$

$$U_{\varphi-control} = -I_x\left(-\lambda_\varphi \varphi_2 - a_1 \theta' \psi' - a_3 \Omega \theta'\right) - K_\varphi sgn(s(\varphi)) \quad (2.79)$$

II.3.2.5. Control of the θ direction

$$\begin{cases} \theta_1' = \theta_2 \\ \theta_2' = a_4\varphi'\psi' + a_6\Omega\varphi' - a_{14}\theta_2^2 + b_2U_3 \end{cases} \quad (2.80)$$

The degree r equal 2,

$$s(\theta) = \left(\frac{d}{dt} + \lambda_\theta\right)^{r-1} e(\theta) = e'(\theta) + \lambda_\theta e(\theta), e(\theta) = \theta^d - \theta$$

$$s(\theta) = \theta^{d\prime} - \theta' + \lambda_\theta(\theta^d - \theta) = \theta^{d\prime} - \theta_2 + \lambda_\theta(\theta^d - \theta_1)$$

$$U_{eq} = -\left(\frac{\partial s}{\partial x}g(\theta)\right)^{-1}\left(\frac{\partial s}{\partial x}f(\theta)\right)$$

$$\frac{\partial s}{\partial x}g(\theta) = [-\lambda_\theta \quad -1]\begin{bmatrix} 0 \\ \frac{-1}{I_y} \end{bmatrix} = \frac{1}{I_y}$$

and $\frac{\partial s}{\partial x}f(\theta) = [-\lambda_\theta \quad -1]\begin{bmatrix} \varphi_2 \\ a_4\varphi'\psi' - a_{14}\theta_2^2 + a_6\Omega\varphi' \end{bmatrix} = -\lambda_\theta\theta_2 - a_4\varphi'\psi' - a_6\Omega\varphi' + a_{14}\theta_2^2$

The Uatt: $U_s = -Ksgn(s(\theta)\ because\ \frac{\partial s}{\partial x}g(\theta) > 0$

$$U_{\theta-contro} = -I_y(-\lambda_\theta\theta_2 - a_4\varphi'\psi' - a_6\Omega\varphi' + a_{14}\theta_2^2) - K_\theta sgn(s(\theta) \quad (2.81)$$

II.3.2.6. Control of the ψ direction

$$\begin{cases} \psi_1' = \psi \\ \psi_2' = a_7\varphi'\theta' - a_{15}\psi_2^2 + b_3U_4 \end{cases} \quad (2.82)$$

The degree r equal 2,

$$s(\psi) = \left(\frac{d}{dt} + \lambda_\psi\right)^{r-1} e(\psi) = e'(\psi) + \lambda_\psi e(\psi), e(\psi) = \psi^d - \psi$$

$$s(\psi) = = \psi^{d\prime} - \psi' + \lambda_\psi(\psi^d - \psi) = \psi^{d\prime} - \psi_2 + \lambda_\psi(\psi^d - \psi_1)$$

$$U_{eq} = -\left(\frac{\partial s}{\partial x}g(\psi)\right)^{-1}\left(\frac{\partial s}{\partial x}f(\psi)\right)$$

$$\frac{\partial s}{\partial x}g(\psi) = [-\lambda_\psi \quad -1]\begin{bmatrix} 0 \\ \frac{-1}{I_z} \end{bmatrix} = \frac{1}{I_z}$$

and $\frac{\partial s}{\partial x} f(\psi) = [-\lambda_\psi \quad -1] \begin{bmatrix} \psi_2 \\ -a_{15}\psi_2^2 + a_7 \varphi' \theta' \end{bmatrix} = -\lambda_\psi \psi_2 - a_7 \varphi' \theta' + a_{15}\psi_2^2$

The Uatt: $U_s = -Ksgn(s(\psi)\ because\ \frac{\partial s}{\partial x} g(\psi) > 0$

$$U_{\psi-control} = -I_z\left(-\lambda_\psi \psi_2 - a_7 \varphi' \theta' + a_{15}\psi_2^2\right) - K_\psi sgn(s(\psi) \qquad (2.83)$$

All the previous steps of SMC used to generate a global control law for the whole system are summarized in the block diagram shown in Figure 2.13. Whereas, the control gains are given in Table 2.3.

Table 2.3. Used gains of the SMC controllers

	x	y	z	φ	θ	ψ
K	10	10	15	20	20	20
λ	2	2	2	0.5	0.9	0.9
ε	0.5	0.5	0.5	0.5	0.5	0.5

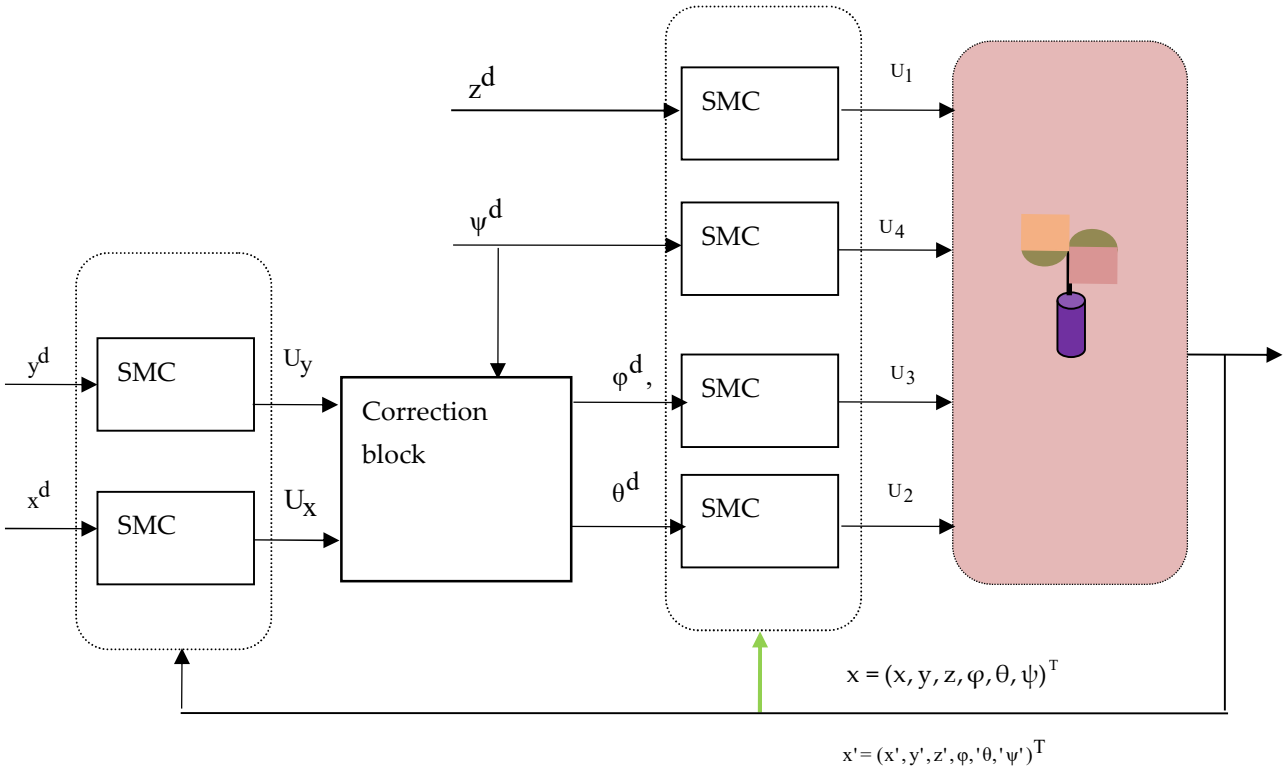


Fig. 2.13. Quadrotor sliding mode control structure.

II.3.3. SMC results and discussions

To test the control theory applied to the quadrotor we chose a square trajectory at an altitude of ten meters and we fixed the yaw angle at zero radium.

The simulation parameters are given in Table 2.3.

Figure 2.14 shows the simulation results of the four inputs, where U1 represents the pushing force it is of the order of ten Newtons and the other inputs the torques are a round zero Nm.

Figure 2.15 illustrates the speeds of the four rotors they are almost equal to 215rd/s. In Figure 2.16 the speed of rotation of the quadrotor is almost zero for a square trajectory and on the left; in addition, the attitudes coincide with the requested movement.

To test the robustness of the control structure, a helical trajectory was applied (see Figure 2.17), and the quadrotor follows the desired trajectory successfully.

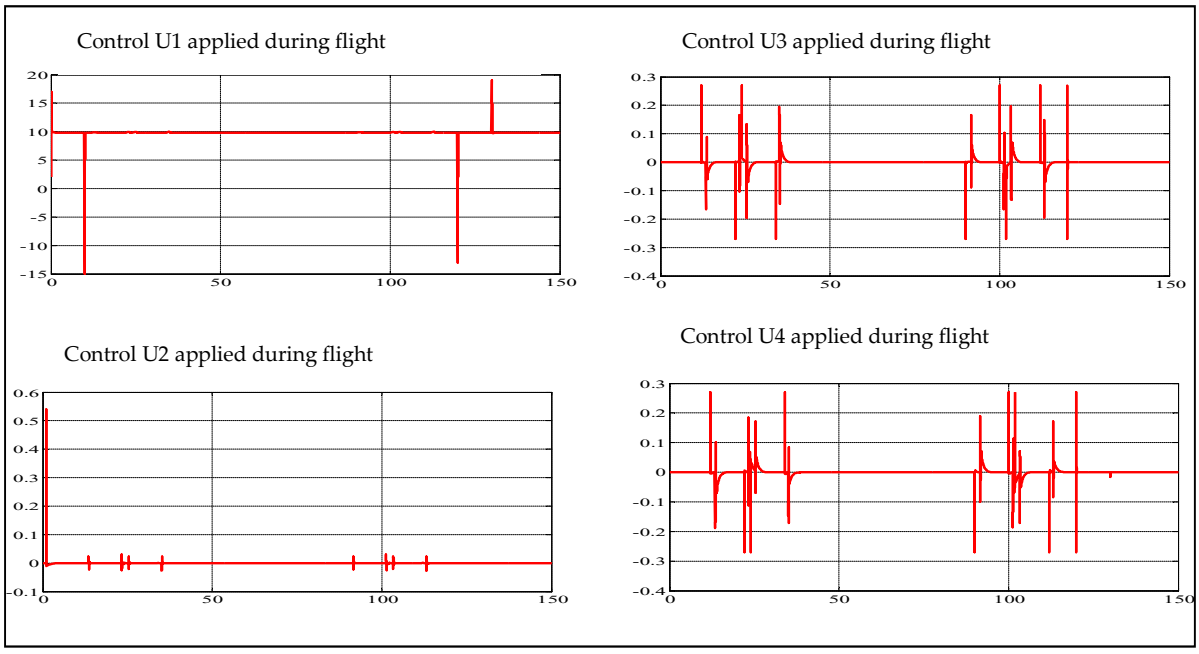


Fig. 2.14. Inputs generated by controllers during simulation (SMC)

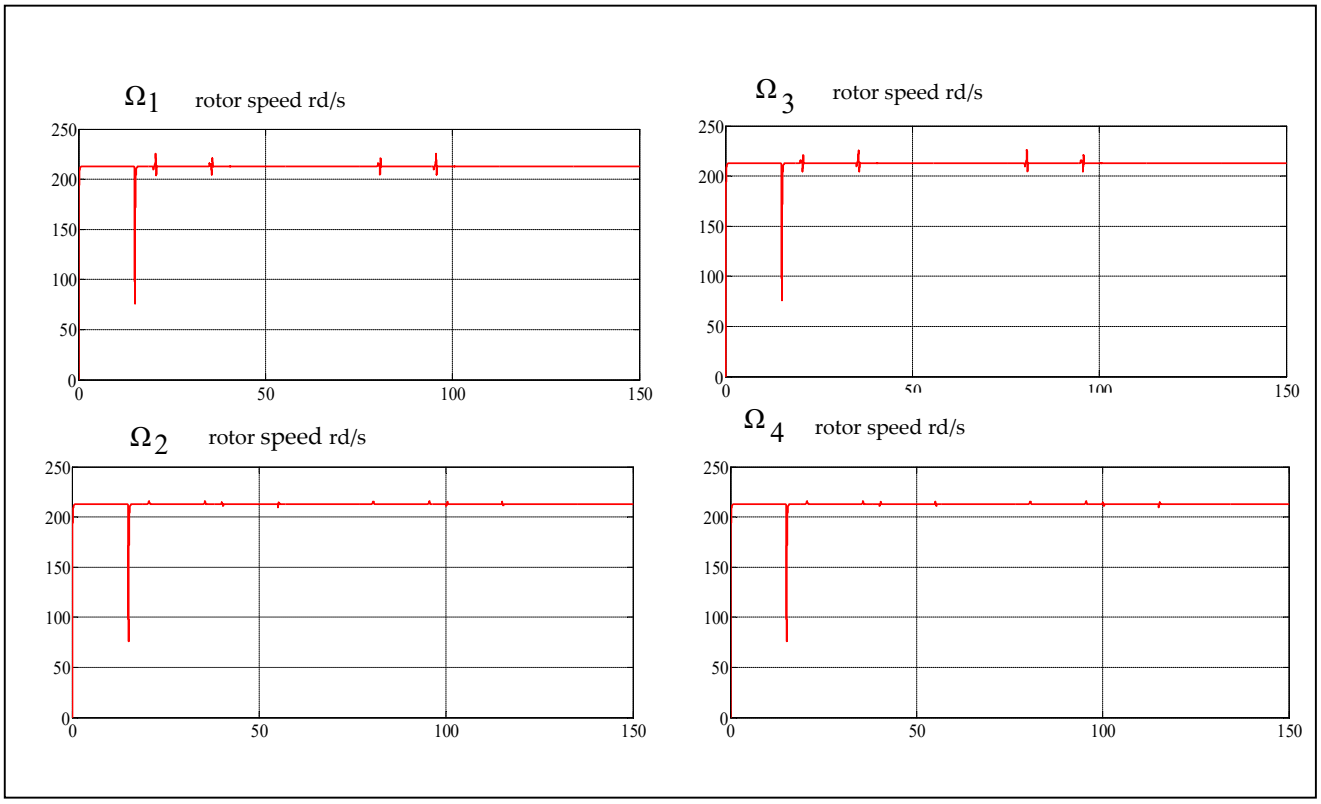


Fig. 2.15. Rotor speeds (SMC)

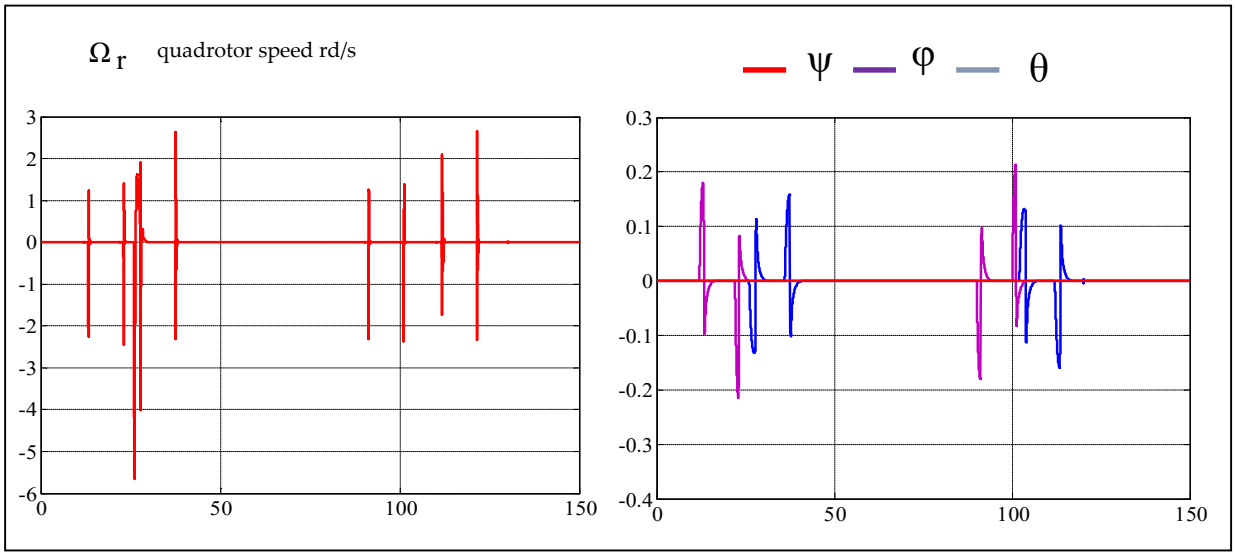


Fig. 2.16. Quadrotor speed and angles responses (SMC).

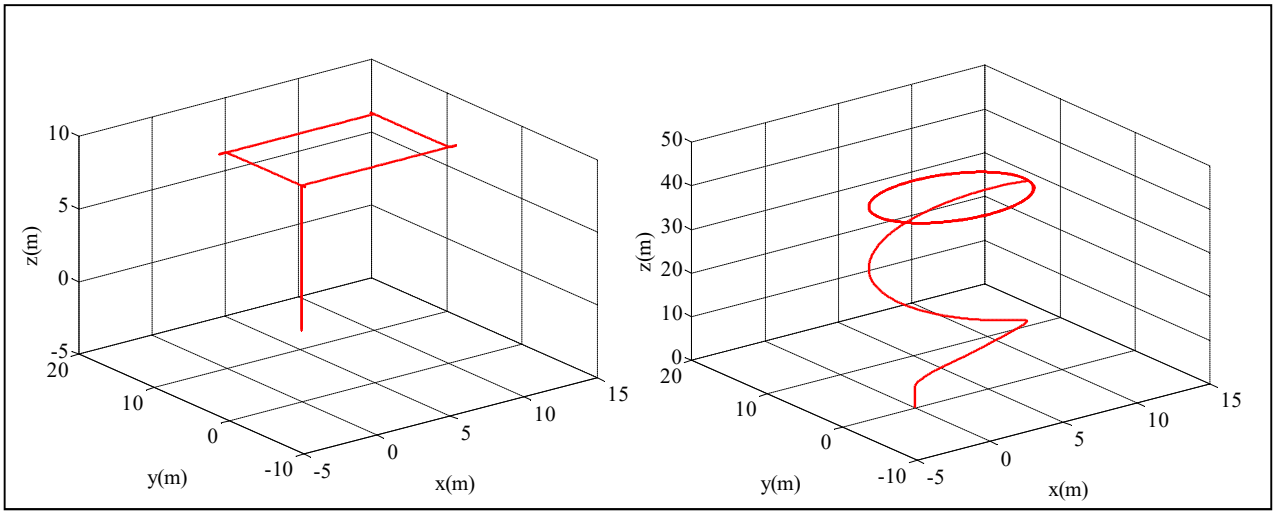


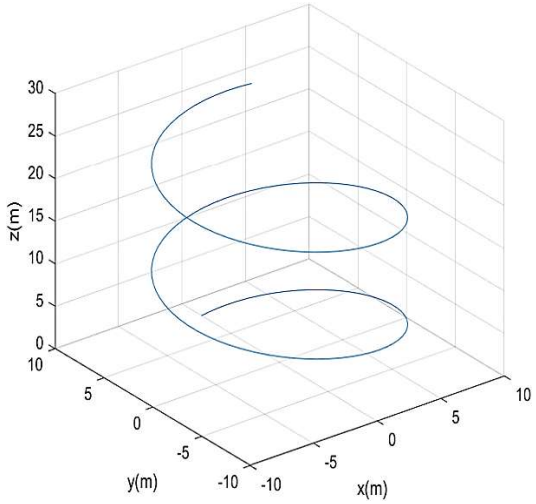
Fig. 2.17. Path of quadrotor controlled with SMC (in left square path, in the right helical path).

In this part of the chapter 2, sliding mode control of quadrotor was realized following the same scenario as the previous part. The obtained simulation results showed the good performances and robustness of this control technique.

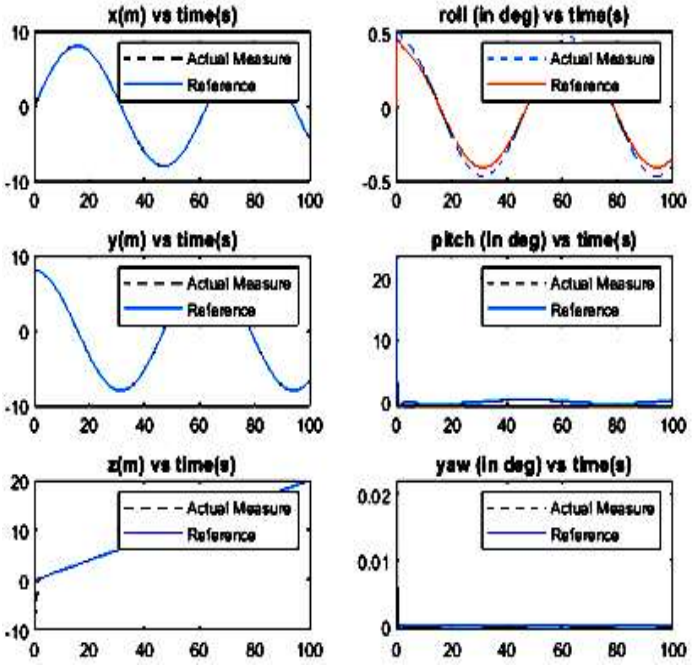
In order to make a comparison between the three control techniques chosen in this chapter, disturbance will be applied to the three control structures, in the coming part, and extract the reaction of each one to this disturbance.

II.4. Nonlinear control for disturbances rejection in quadrotors

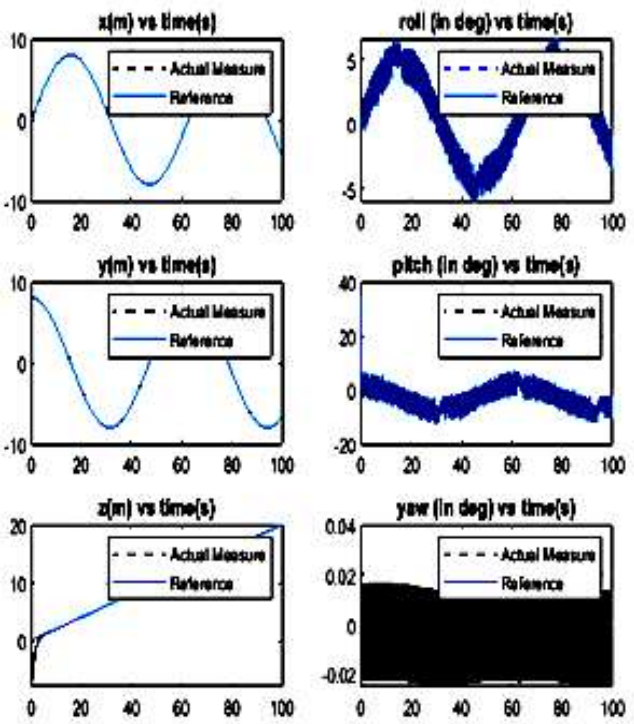
Matlab simulink is the most useful software used to test the behaviour of nonlinear systems and to validate the results of the recently developed controllers [38], [40], [41]. Therefore, model simulation is done using Matlab Simulink program to verify our suggested control approach. To achieve that, a quadrotor model is designed and controlled by three different controllers which are: PD, sliding mode and the BSC. Hence, each one of them is tested with and without disturbances to track a quadrotor's trajectory of radians R=8 shown in Figure 2.18. The quadrotor has to track the trajectory defined by the time functions: $X = 8\sin(0.1t)$, $Y = 8\cos(0.1t)$ and $Z = 0.2t$.



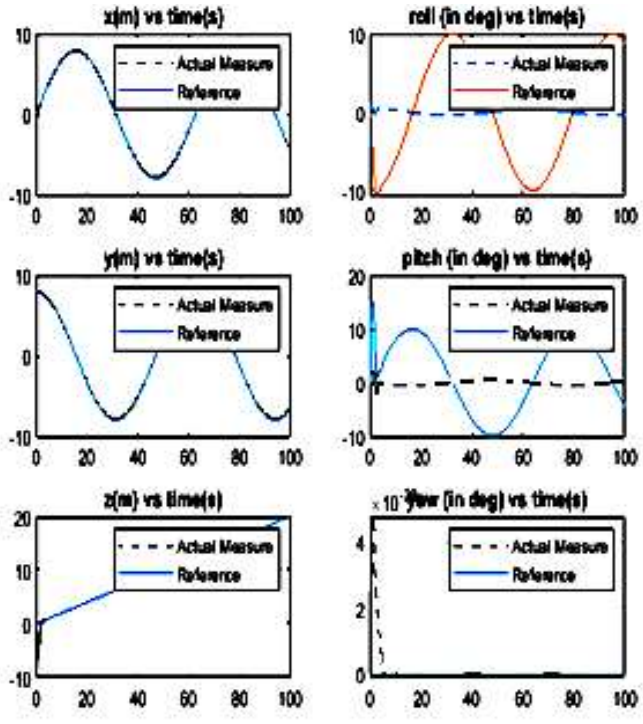
(a)



(b)



(c)



(d)

Fig. 2.18. Attitude, heading and position reference, (a) trajectory tracked by the Quadrotor. (b) PDC, (c) SMC, and (d) BSC.

II.4.1. Without disturbance

Figure 2.18(a) is an illustration of the trajectory tracked by the quadrotor. The next Figures show the position, orientation (see Figure 2.18(b), Figure 2.18(c), and Figure 2.18(d)), trajectory errors (illustrated in Figure 2.19(a), Figure 2.19(b) and Figure 2.19(c)) and control inputs in the absence of disturbances generated by the three controllers.

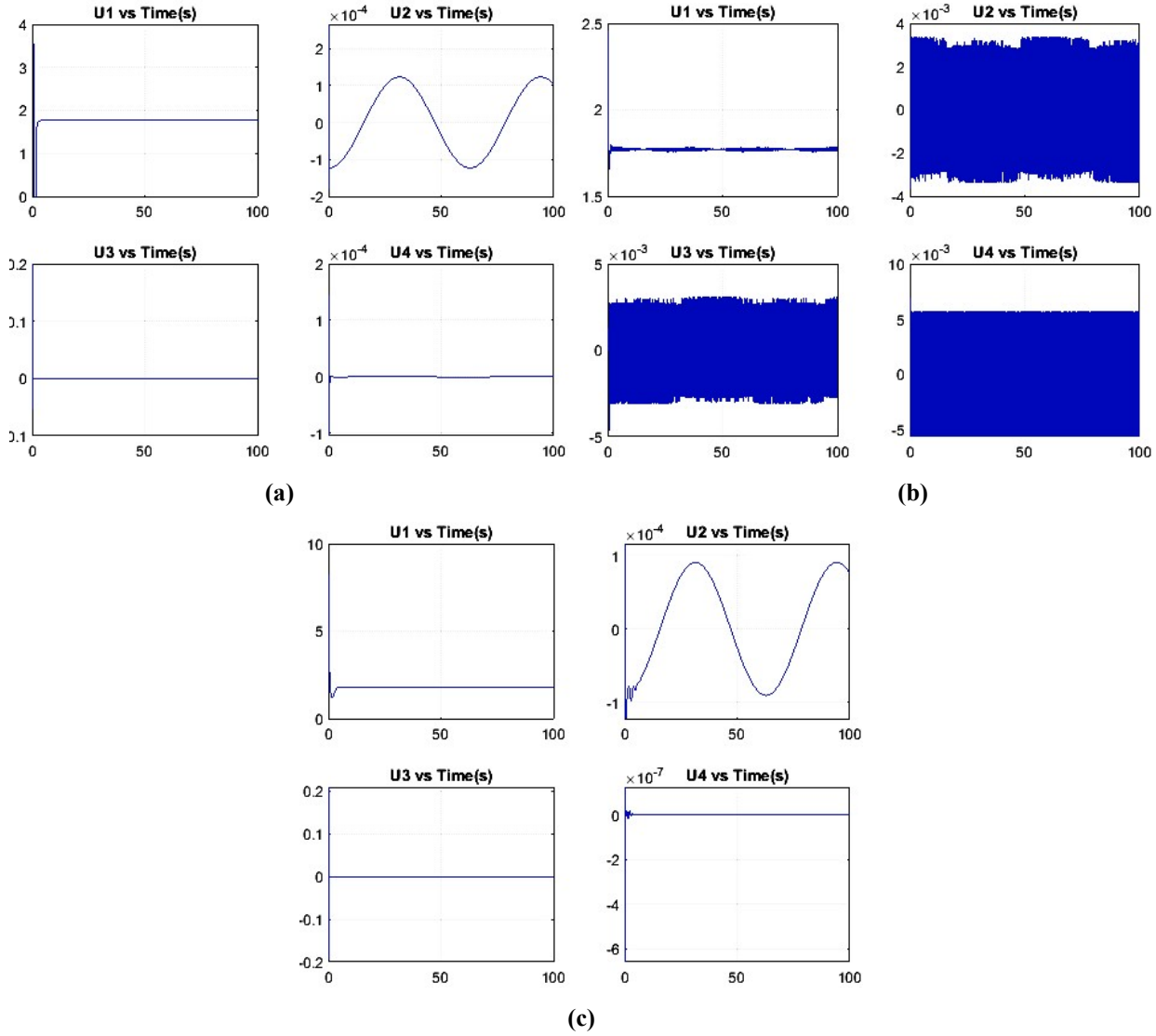
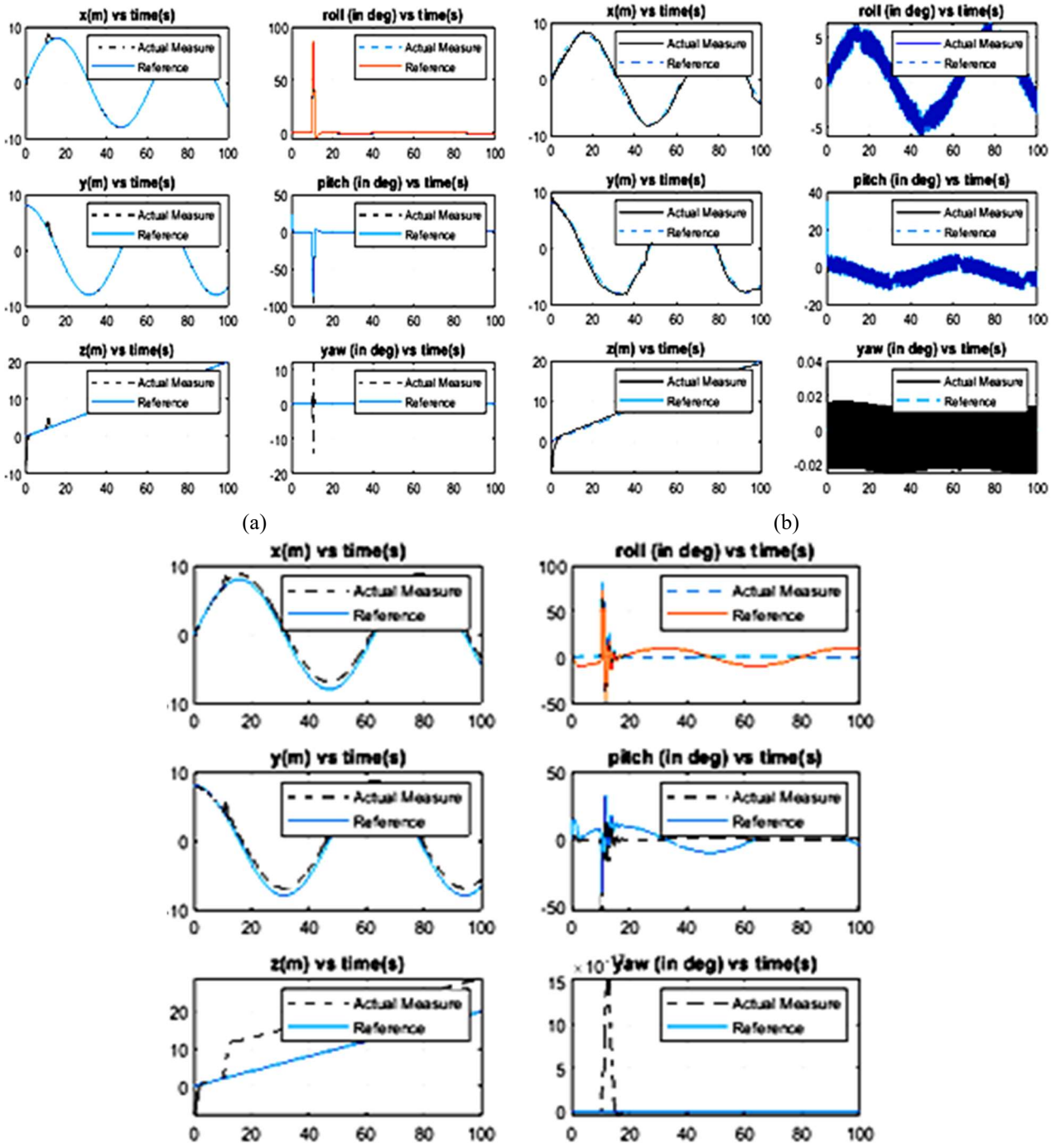


Fig. 2.19. U1, U2, U3 and U4 vs. time, (a) PDC, (b) SMC, and (c) BSC

Discussion: without any disturbance, the steady state error from the SMC was the smallest followed by PDC then the BSC for the displacement across both the X and Y axes. In the event that no disturbances, SMC has proven its efficiency even in the previous work [4], [20], [42]. However, for the displacement on the Z axis, the SMC did 10^{12} times worse than the BSC, with the PDC doing the best here. And finally, for the steady state error for yaw angle, the BSC also was 10^{17} better than the PDC, and the SMC and being lasting this category.

II.4.2. With disturbance

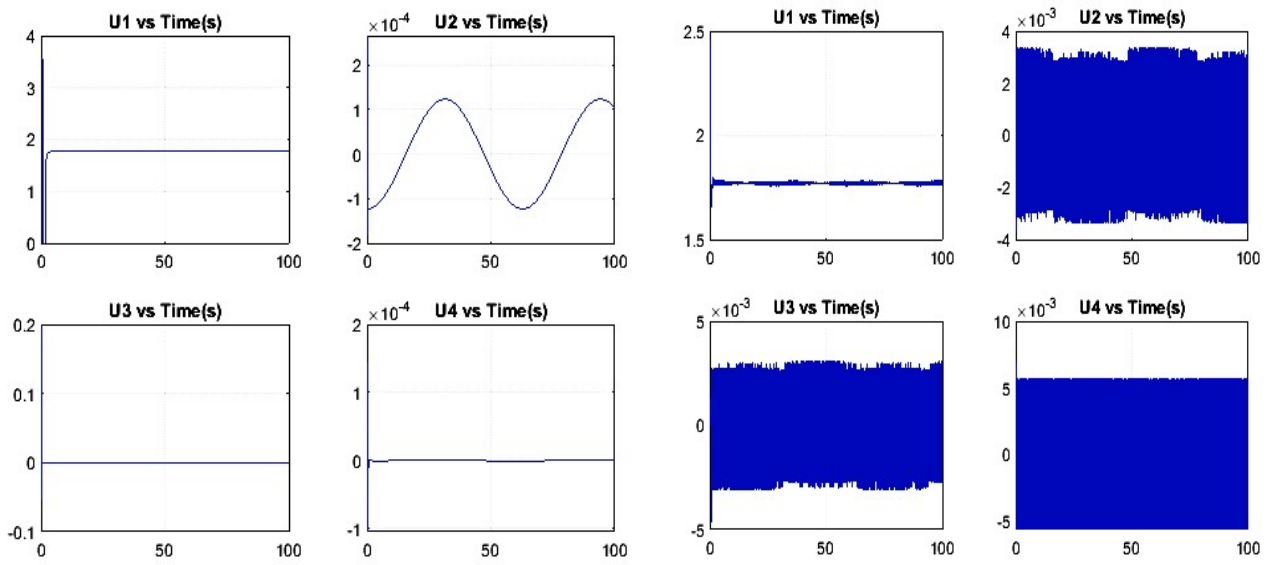
The Next figures show the position and orientation (see Figures 2.20(a), 2.20(b) and 2.20(c)), and control inputs (shown in Figures 2.21(a), 2.21(b) and 2.21(c)) in the presence of a ramp disturbance of the vector F=9t i+9t j+ 9t k (N), starting from the 10th second of the simulation with a force limit of 9 N.



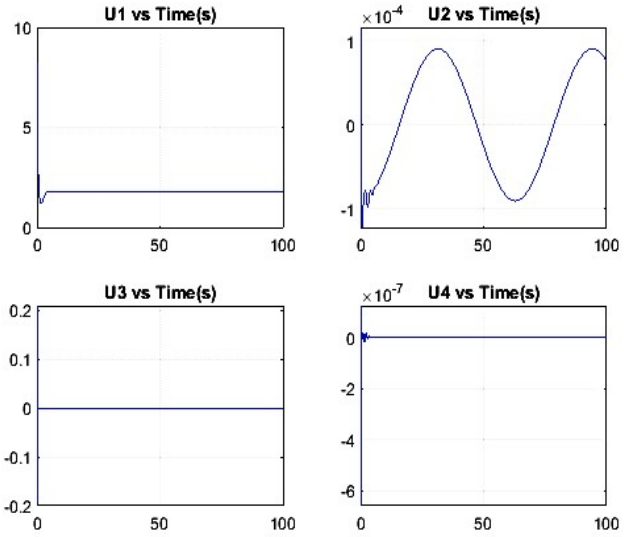
(a) (b)

(c)

Fig.2.20. Altitude, heading and position reference measurement vs. actual measurement, (a) PDC, (b) SMC, and (c) BSC.



(a) (b)



(c)

Fig.2.21. U1, U2, U3 and U4 vs. time, (a) PDC, (b) SMC, and (c) BSC

Discussion: now, in the case of disturbance. For small disturbance, the steady state error for every controller kept the same order as in case of no disturbance. But as the disturbance increased the controllers could not keep the quadrotor in trajectory anymore. The first controller that collapsed was the SMC. Making the PDC the best controller of the three under disturbance, but even this last one also collapsed after adding 133.33% of the first disturbance force. Then, BSC proves that it is the best to resist for all kind of disturbances; but, after adding about 50% of the previous disturbance force, the BSC collapsed too. However, the simulation result obtained from the implementation of the proposed BSC is very satisfactory compared to the previous works which show some complexity of

the analytical inference and a considerable dynamic error, especially in presence of noise [4], [5], [43]

II.5. Conclusion

It is established that the position, orientation and attitude path following errors can rapidly converge to slight values with all controllers. In case of non-external disturbance, BSC shows good control of the yaw angle and the altitude of the quadrotor comparing to the two other controllers (SMC and PDC).

Moreover, in the case of presence of disturbances, for small disturbance, each controller's steady state error maintained the same order as in the absence of any disturbance. However, as the disturbance increase the controllers could not keep the quadrotor in trajectory anymore. Numerical and simulation results confirm that BSC is the last one that collapsed which confirm the robustness and efficacy of our constructed enhanced control strategy .

Chapter 3: Extended Kalman Filter (EKF) estimation

Different techniques were established to estimate the required quadrotor's position and altitude measurements such as: Kalman filtering (KF) [11], the complementary filter (CF) [12] and Extended Kalman Filters (EKF) [13] estimators. Kalman filter is a well-known recursive algorithm that takes the stochastics states spaces model of the system together with measured outputs to achieve the optimal estimation states. The optimality of the state's estimation is achieved with the minimization of the mean estimation error. Therefore, EKF and backstepping were investigated in this chapter to estimate the angular velocity and the speed of our quadrotor; hence, track the desired trajectory and control the altitude.

III.1. Quadrotors state model

The estimated orientation of the quadrotor is defined by the Euler angles: $\{\hat{\varphi}, \hat{\theta}, \hat{\psi}\}$, pitch, roll, and yaw, respectively. The total estimated angular velocity of the quadrotor represented by $\{\widehat{\varphi'}, \widehat{\theta'}, \widehat{\psi'}\}$, is seen in the axes (X, Y, Z) of the reference frame. Therefore, the total estimated speed of the quadrotor is presented by $\{\widehat{x'}, \widehat{y'}, \widehat{z'}\}$. Hence, in this work EKF, is used for the estimation of: $\hat{X} = \left[\hat{x}, \hat{y}, \hat{z}, \hat{\varphi}, \hat{\theta}, \hat{\psi}, \widehat{x'}, \widehat{y'}, \widehat{z'}, \widehat{\varphi'}, \widehat{\theta'}, \widehat{\psi'}\right]^T$.

The state space representation of the quadrotor is defined as:

$$\begin{cases} x_1' = x_2 \\ x_2' = a_1 x_6 x_4 + \Omega a_3 x_4 - a_{13} x_2^2 U_\varphi \\ x_3' = x_4 \\ x_4' = a_4 x_2 x_6 + \Omega a_6 - a_{14} x_4^2 + b_2 U_\theta \\ x_5' = x_6 \\ x_6' = a_7 x_2 x_4 - a_{15} x_6^2 + b_3 U_\psi \\ x_7' = x_8 \\ x_8' = a_9 x_8 + \frac{\cos(x_1)\cos(x_3)}{m} U_1 - g \\ x_9' = x_{10} \\ x_{10}' = a_{10} x_{10} + \frac{U_1}{m} U_y \\ x_{11}' = x_{12} \\ x_{12}' = a_{11} x_{12} + \frac{U_1}{m} U_x \end{cases} \tag{3.1}$$

Where: $x_1 = \varphi, x_2 = p, x_3 = \theta, x_4 = q, x_5 = \psi, x_6 = r,\ x_7 = z, x_8 = z', x_9 = y, x_{10} = y', x_{11} = x, x_{12} = x',\ a1 = \frac{Iy-Iz}{Ix}, a2 = \frac{Jr}{Ix}, a4 = \frac{Jr}{Iy}, a3 = \frac{Iz-Ix}{Iy}, a5 = \frac{Iy-Ix}{Iz}\ ,\quad b1 = \frac{1}{Ix}, b2 = \frac{1}{Iy},$

$b3 = \frac{1}{Iz}, a9 = -\frac{kftz}{m}, a10 = -\frac{kfty}{m},\quad a13 = \frac{Kfax}{Ix}, a14 = \frac{Kfay}{Iy}, a15 = \frac{Kfa}{Iz}.$

In the vector form as $x'(t) = f(x) + g(x)U + g_r(x)$, the system is represented in the following form:

$$\begin{bmatrix} x_1' \\ x_2' \\ x_3' \\ x_4' \\ x_5' \\ x_6' \\ x_7' \\ x_8' \\ x_9' \\ x_{10}' \\ x_{11}' \\ x_{12}' \end{bmatrix} = \begin{bmatrix} x_2 \\ a_1 x_6 x_4 + \Omega a_3 x_4 - a_{13} x_2^2 \\ x_4 \\ a_4 x_2 x_6 + \Omega a_6 x_2 - a_{14} x_4^2 \\ x_6 \\ a_7 x_2 x_4 - a_{15} x_6^2 \\ x_8 \\ a_9 x_8 \\ x_{10} \\ a_{10} x_{10} \\ x_{12} \\ a_{11} x_{12} \end{bmatrix} + \begin{bmatrix} 0 \\ b_1 U_\varphi \\ 0 \\ b_2 U_\theta \\ 0 \\ b_3 U_\psi \\ 0 \\ \frac{\cos(x_1)\cos(x_2)}{m} U_1 \\ 0 \\ \frac{U_1}{m} U_y \\ 0 \\ \frac{U_1}{m} U_x \end{bmatrix} - \begin{bmatrix} 0 \\ 0 \\ 0 \\ 0 \\ 0 \\ g \\ 0 \\ 0 \\ 0 \\ 0 \end{bmatrix} \tag{3.2}$$

$$\begin{bmatrix} x_1' \\ x_2' \\ x_3' \\ x_4' \\ x_5' \\ x_6' \\ x_7' \\ x_8' \\ x_9' \\ x_{10}' \\ x_{11}' \\ x_{12}' \end{bmatrix} = \begin{bmatrix} x_2 \\ a_1 x_6 x_4 + \Omega a_3 x_4 - a_{13} x_2^2 \\ x_4 \\ a_4 x_2 x_6 + \Omega a_6 x_2 - a_{14} x_4^2 \\ x_6 \\ a_7 x_2 x_4 - a_{15} x_6^2 \\ x_8 \\ a_9 x_8 \\ x_{10} \\ a_{10} x_{10} \\ x_{12} \\ a_{11} x_{12} \end{bmatrix} + \begin{bmatrix} 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & b_1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & b_2 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & b_3 \\ 0 & 0 & 0 & 0 & 0 & 0 \\ \frac{1}{m} & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & \frac{1}{m} & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & \frac{1}{m} & 0 & 0 & 0 \end{bmatrix} \begin{bmatrix} U_z' \\ U_y' \\ U_x' \\ U_\varphi \\ U_\theta \\ U_\psi \end{bmatrix} + \begin{bmatrix} 0 \\ 0 \\ 0 \\ 0 \\ 0 \\ g \\ 0 \\ 0 \\ 0 \\ 0 \end{bmatrix} \tag{3.3}$$

Thus

$$f(\mathrm{x}) = \begin{bmatrix} x_2 \\ a_1 x_6 x_4 + \Omega a_3 x_4 - a_{13} x_2^2 \\ x_4 \\ a_4 x_2 x_6 + \Omega a_6 x_2 - a_{14} x_4^2 \\ x_6 \\ a_7 x_2 x_4 - a_{15} x_6^2 \\ x_8 \\ a_9 x_8 \\ x_{10} \\ a_{10} x_{10} \\ x_{12} \\ a_{11} x_{12} \end{bmatrix}, g(x) = \begin{bmatrix} 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & b_1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & b_2 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & b_3 \\ 0 & 0 & 0 & 0 & 0 & 0 \\ \frac{1}{m} & 0 & 0 & 0 & 0 & 0 \\ 0 & \frac{1}{m} & 0 & 0 & 0 & 0 \\ 0 & 0 & \frac{1}{m} & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 \end{bmatrix},$$

$$g_r(x) = \begin{bmatrix} 0 \\ 0 \\ 0 \\ 0 \\ 0 \\ g \\ 0 \\ 0 \\ 0 \\ 0 \end{bmatrix} \tag{3.4}$$

Taking the derivate of *f(x)* with respect to *x*:

$$\frac{\partial f}{\partial x} = \begin{bmatrix} 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & -2a_{13} & 0 & \Omega a_2 + a_1 x_6 & 0 & a_1 x_4 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & -\Omega a_4 + a_3 x_6 & 0 & -2a_{14} & 0 & a_3 x_2 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & a_3 x_4 & 0 & a_5 x_2 & 0 & -2a_{15} & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & a_9 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & a_{10} & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & a_{11} \end{bmatrix} \tag{3.5}$$

The discrete formulation of the state model is obtained by discrediting the continuous solution between two sampling times T_s, by considering $t(k+1) - t(k) = Ts$ and $t(k) = kTs \ (k \in N)$.

Thus:

$$\mathrm{F} = \mathrm{e}^{dF * Ts} \approx I + dF * T_S, G = g(x) * T_S, \ G_r = g_r(x) * T_S$$

Where I is the identity matrix and $dF = \frac{\partial f}{\partial x}$. Therefore, F can be calculated from Eq. (3.5) as follow:

$$I + \left(\frac{\partial f}{\partial x}\right) T_S$$

$$= \begin{bmatrix} 1 & T_S & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 1-2a_{13}T_S & 0 & \Omega a_2 T_S + a_1 x_6 T_S & 0 & a_1 x_4 T_S & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & T_S & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & -\Omega a_4 T_S + a_3 x_6 T_S & 0 & 1-2a_{14}T_S & 0 & a_3 x_2 T_S & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & T_S & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & a_3 x_4 T_S & 0 & a_5 x_2 T_S & 0 & 1-2a_{15}T_S & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 1 & T_S & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1+a_9 T_S & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & T_S & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1+a_{10}T_S & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & T_S \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1+a_{11}T_S \end{bmatrix} \tag{3.6}$$

In the same way, we get:

$$G = g(x)T_S = \begin{bmatrix} 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & b_1 T_S & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & b_2 T_S & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & b_3 T_S \\ 0 & 0 & 0 & 0 & 0 & 0 \\ -T_S & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & -T_S & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & -T_S & 0 & 0 & 0 \end{bmatrix}, \; G_r = g_r(x)T_S = \begin{bmatrix} 0 \\ 0 \\ 0 \\ 0 \\ 0 \\ 0 \\ 0 \\ gT_S \\ 0 \\ 0 \\ 0 \\ 0 \end{bmatrix} \tag{3.7}$$

III.2. Extended Kalman Filter (EKF) estimation

The algorithm of the Extended Kalman filter (EKF) can be summarized in nine steps: first, the state vector must be initialized followed by the acquisition of the data; then, the state must be predicted. After that, matrix of the covariance error must be estimated and the gain of the Kalman filter must be calculated. Therefore, the state-vector can be estimated at time (k+1) and the estimation covariance error must be updated. At the end, the final results must be stored and repeated from step 3. These steps are explained in detail in the following subsections. Start by defining:

States $X = [x, y, z, \varphi, \theta, \psi, x', y', z', \varphi', \theta', \psi']^T$,

Input $U = \left[U_z, U_y, U_{x,} U_\varphi, U_\theta, U_\psi\right]^T$, and

Output $y = [x, y, z, \varphi, \theta, \psi]^T$

III.2.1. Initialize the state vector:

$\tilde{x}(0/0), P(0/0), R$ and Q must be initialized. With P is the covariance matrix of states estimation, R is the covariance matrix of the output noise, and Q is the covariance matrix of the system noise.

III.2.2. Data acquisition

At this step, the process starts the acquisition of the data: $u(k/k), y(k+1)$

III.2.3. Prediction of the state

Prediction of the state vector at sampling time (k+1) from the input u (k), state vector at previous sampling time $\tilde{x}(^k/_k)$, by using F given by Eq.(3.6) and G given by Eq.(3.7), is obtained from:

$$\tilde{x}\left(^{k+1}/_k\right) = F * \tilde{x}(k/k) + G * u(k)) \tag{3.8}$$

$$\tilde{y}(^{k+1}/_k) = g(\tilde{x}(k/k), u(k)) \tag{3.9}$$

The notation $\tilde{x}(^{k+1}/_k)$ means that it is a predicted value at the $(k+1)^{th}$ instant, and it is based on the measurements up to k^{th} instant. In the following step of the recursive EKF computation starts.

III.2.4. Estimation of the covariance error matrix

The covariance error matrix can be recursively estimated using the following equation:

$$P(^{k+1}/_k) = F(k)P(k/k)\,F^T(k) + Q(k) \tag{3.10}$$

III.2.5. Calculation of the gain of the Kalman filter

The Kalman filter gain $K(k+1)$ is computed as;

$$K(k+1) = P(^{k+1}/_k)G^T(k)(GP(^{k+1}/_k)G^T + R(k))^{-1} \tag{3.11}$$

Where $G = \frac{\partial h}{\partial x}$,

With h represents the output of the system.

III.2.6. State estimation

The state-vector estimation at time (k+1) then is determined as:

$$\tilde{x}(k+1/k+1) = \tilde{x}(k+1/k) + K(k+1)\big(y(k+1) - \tilde{y}(k+1)\big) \quad (3.12)$$

When calculating the new state value $\tilde{x}(k+1/k+1)$, the Kalman filter gain $K(k+1)$ is multipliedwith the error of the output (named the innovation). The innovation process $(y(k+1) - \tilde{y}(k+1))$ has an important impact in improving the results of our work; where, $y(k+1)$ is the real output of the process and $\tilde{y}(k+1)$ is the estimated output of the process.

III.2.7. Updating the matrix of the estimation covariance error

The last step of the EKF algorithm is to estimate the covariance computation as:

$$P(k+1/k+1) = F(k)(I - K(k+1)G)P(k+1/k) \quad (3.13)$$

III.2.8. The storage of the final value

The obtained results must be stored using the following formula:

$$\tilde{x}(k/k) = \tilde{x}\,(k+1/k+1)$$

$$P(k/k) = P(k+1/k+1) \quad (3.14)$$

III.2.9. For the next step (k+1),

Repeat the steps from step 2.

III.3. Results and discussions

In order to validate our proposed control solution, the model is simulated under Matlab/Simulink software as shown in Fig. 3.1.

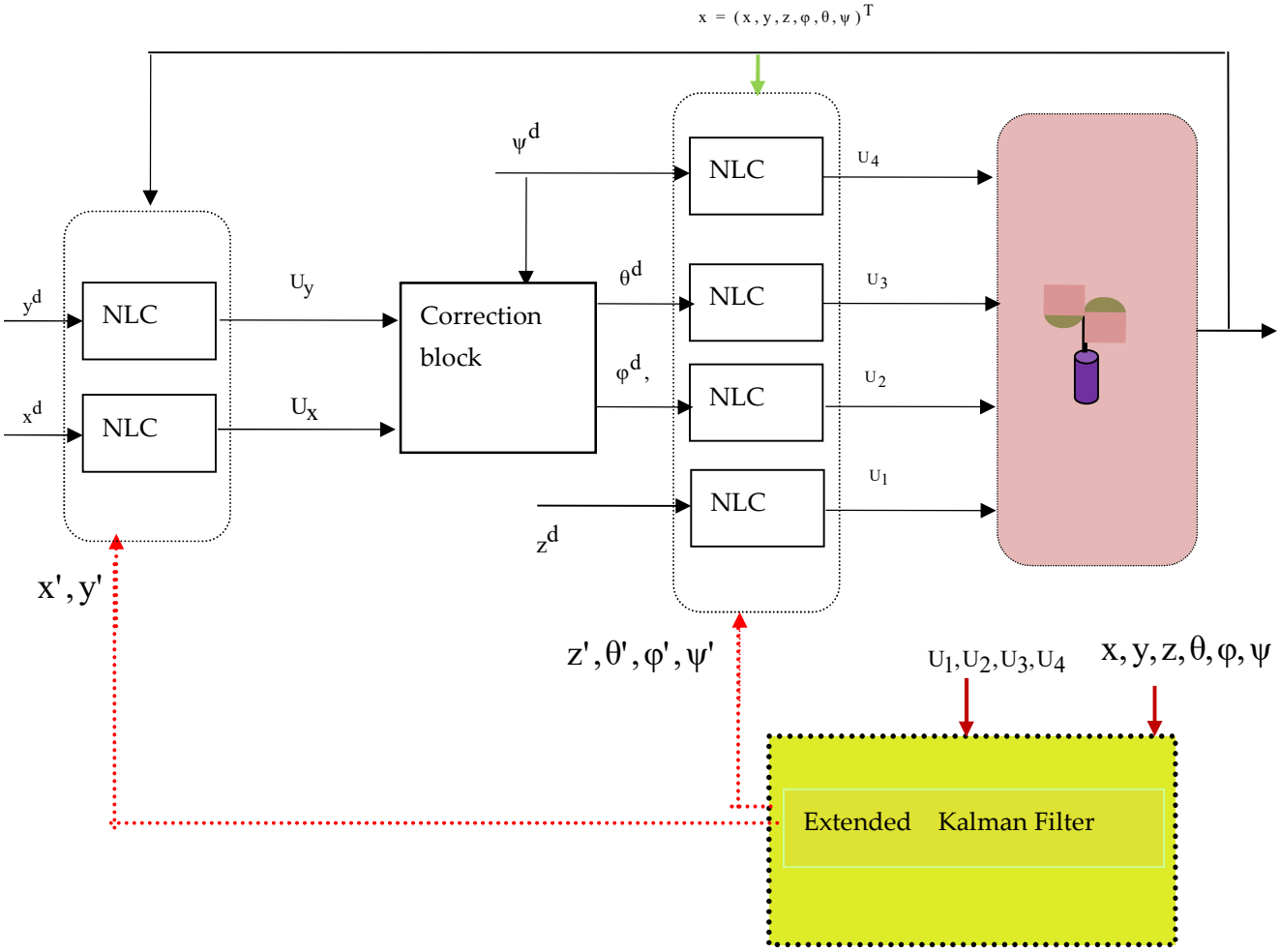


Fig. 3.1. Block diagram Nonlinear Control of Quadrotor using EKF.

For that purpose, the results are obtained based on the application of the real parameters summarized on Table 3.1 [23].

Table 3.1. Quadrotor parameter used in our simulation

Ix	Body inertia respect to x axis	8. e^{-3} Kg. m^2		Friction aerodynamics coefficients
Iy	Body inertia respect to y axis	8. e^{-3} Kg. m^2	Kfax	5.5670.e^{-6} N/rd/s
Iz	Body inertia respect to z axis	14.2.e^{-6} Kg. m^2	Kfay	5.5670.e^{-6} N/rd/s
Jr	Rotor inertia	104 e^{-6} Kg. m^2	Kfaz	6.3540.e^{-4} N/rd/s
m	Mass of the quadrotor	1 Kg		
g	Gravitational constant	9.81 m. s^{-2}		Translation drag coefficients
b	Thrust factor	54.2 e^{-6}	Kftx	5.5670.e^{-6} N/m/s
d	Grag factor	1.1e^{-6} m	Kfty	5.5670.e^{-6} N /m/s
l	Horizontal distance: propeller center to CoG	0.24 m	Kftz	6.3540.e^{-4} N/m/s

In this part of or work we simulated the nonlinear control without linear and angular sensors using the extended kalman filter according to the following scenario, who it is desired to follow a rectangular trajectory in *XY* plane. The height *z* increases uniformly from zero to 15 meters where the drone stabilizes as shown in Figure 3.2.

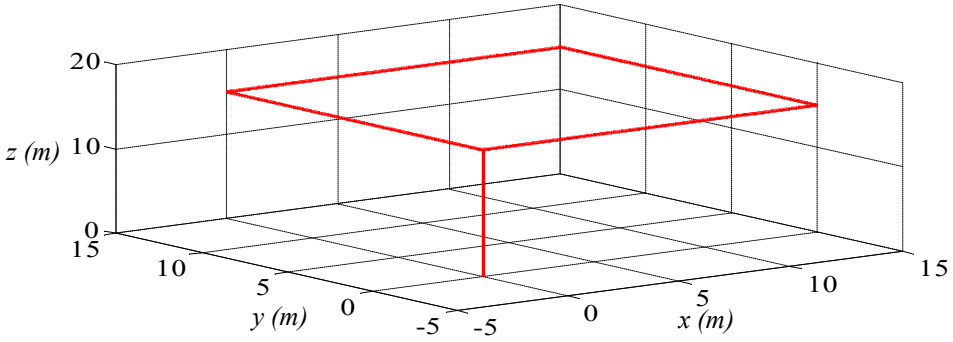
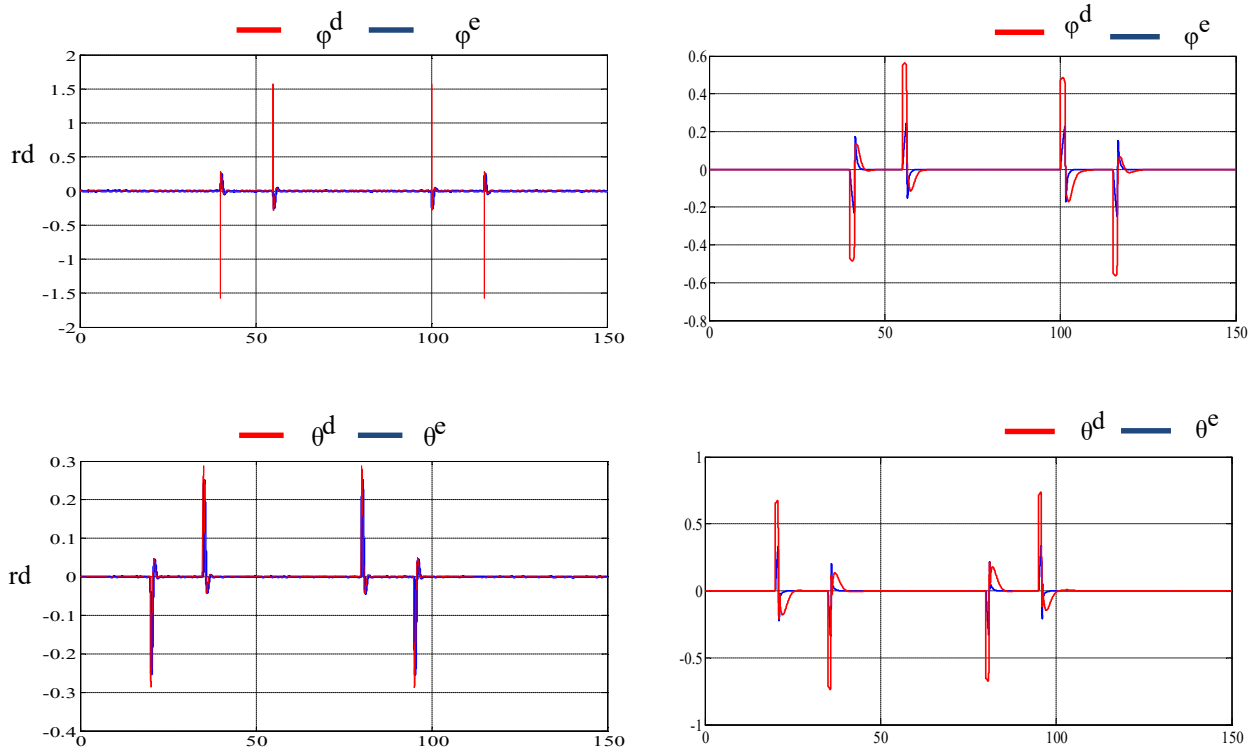


Fig. 3.2. The desired rectangular trajectory to be followed by the quadrotor.

In Fig 3.3, we find the simulation results of the roll angle ϕ, pitch angle θ and yaw angle ψ using nonlinear control, where we notice that the difference between the desired and estimated quantities is small in the case of the backstepping control.



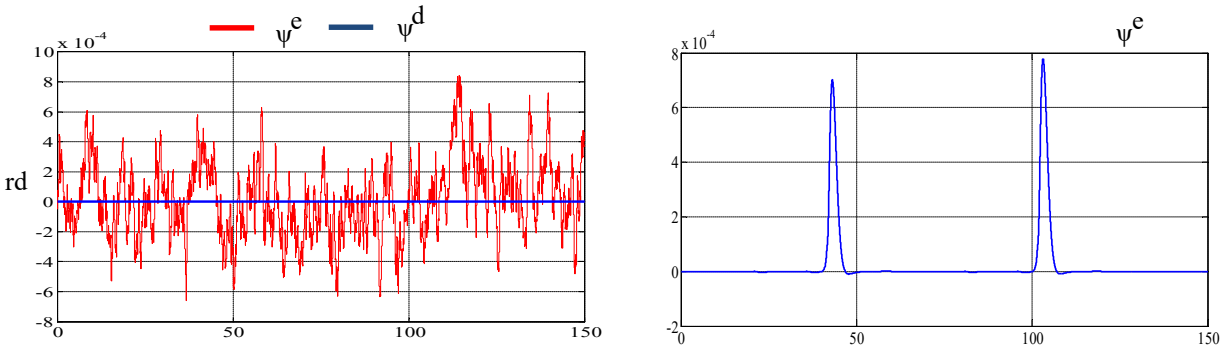


Fig. 3.3. Simulation results of the roll angle ϕ, pitch angle θ and yaw angle ψ using nonlinear control. (a) BSC, (b) SMC

In Fig.3.4, we find the estimated linear velocities where we notice that the results given by BSC are relatively good compared to those given by SMC.

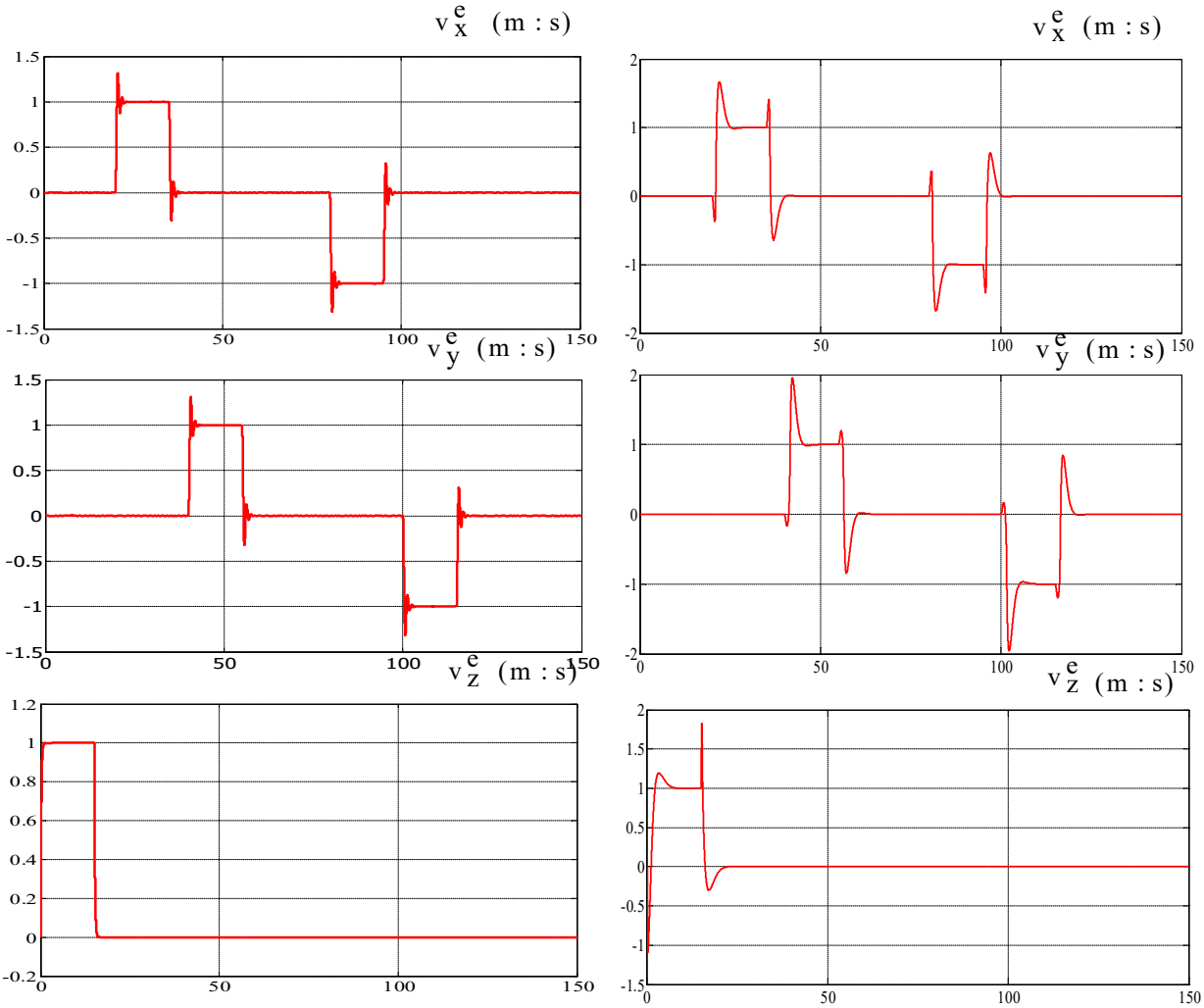


Fig. 3.4. Simulation results of the *x* velocity, *y* velocity and *z* velocity using EKF, (a) BSC, (b) SMC

Fig. 3.5 illustrates the simulation results of the angular velocities estimated using EKF where notice that the velocities given by the sliding mode control are more than the double of the velocities given by the backstepping control during the transitional regime.

Fig 3.6 shows the simulation results of the positions estimated using the extended kalman filter where we notice that the estimated states follow the desired instructions in the two control cases with an acceptable overshoot in the case of sliding mode control.

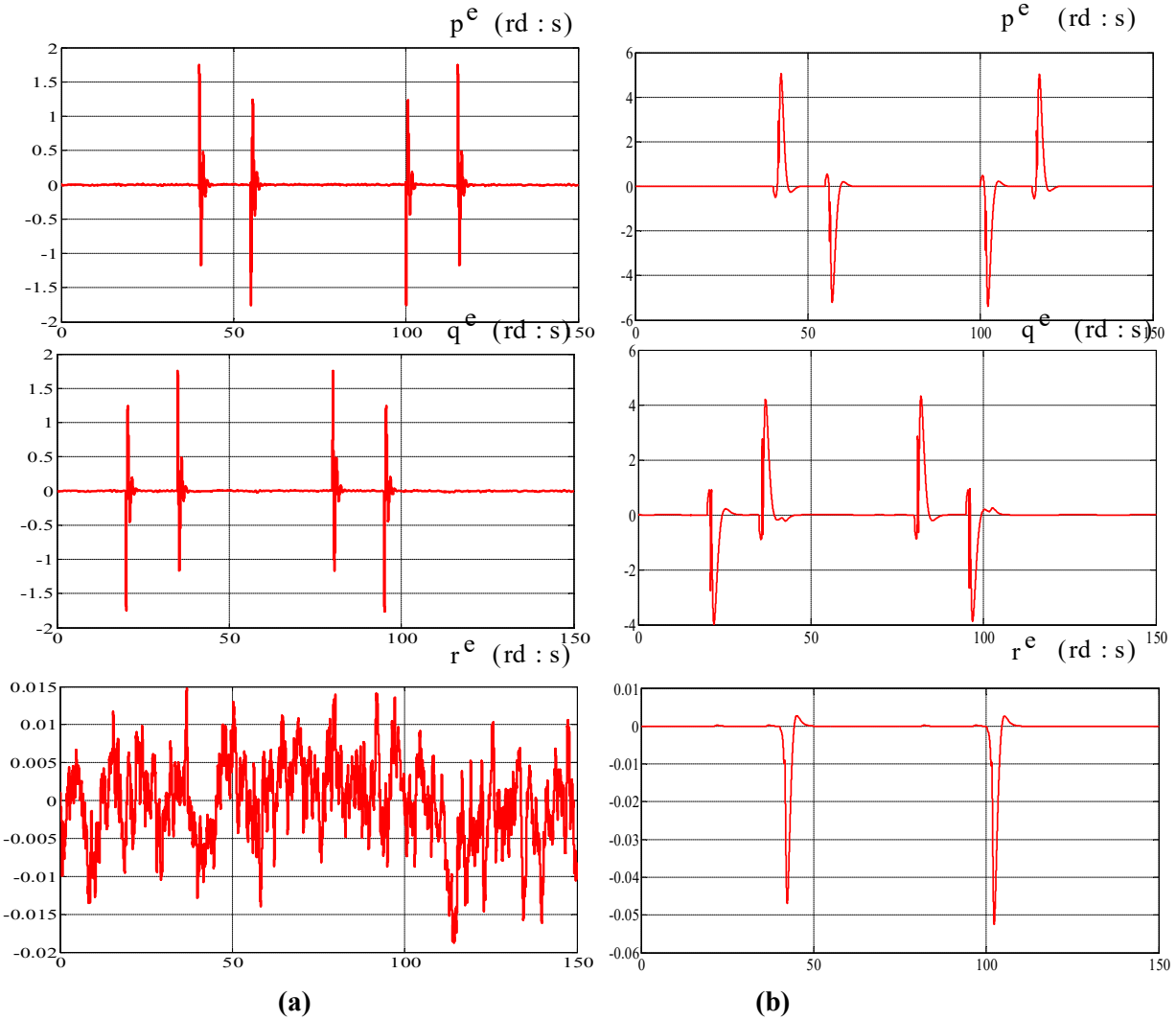


Fig. 3.5 the estimated results using EKF of: the pitch rate *q*, the roll rate *p* and the yaw rate *r*, (a) BSC, (b) SMC

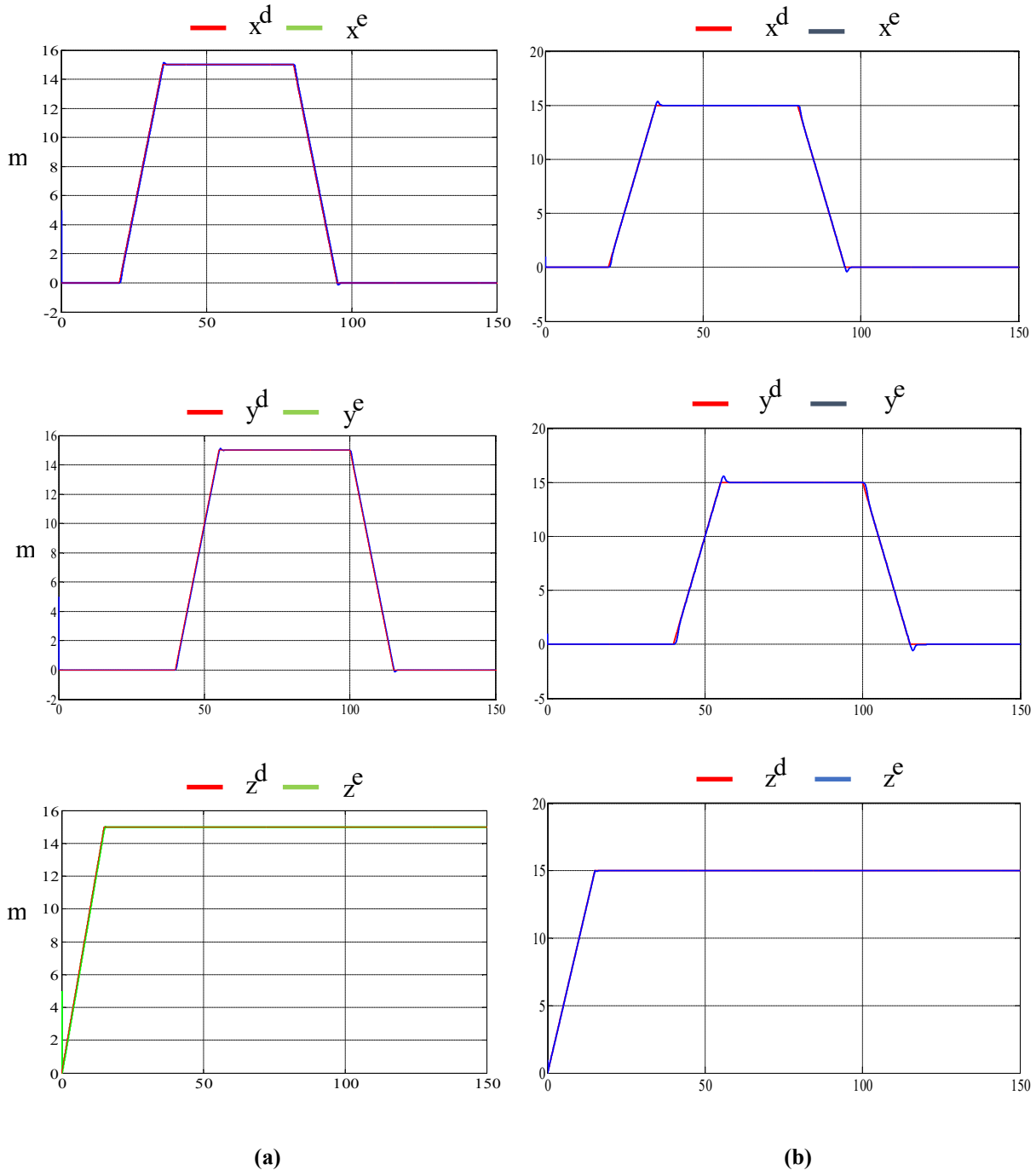
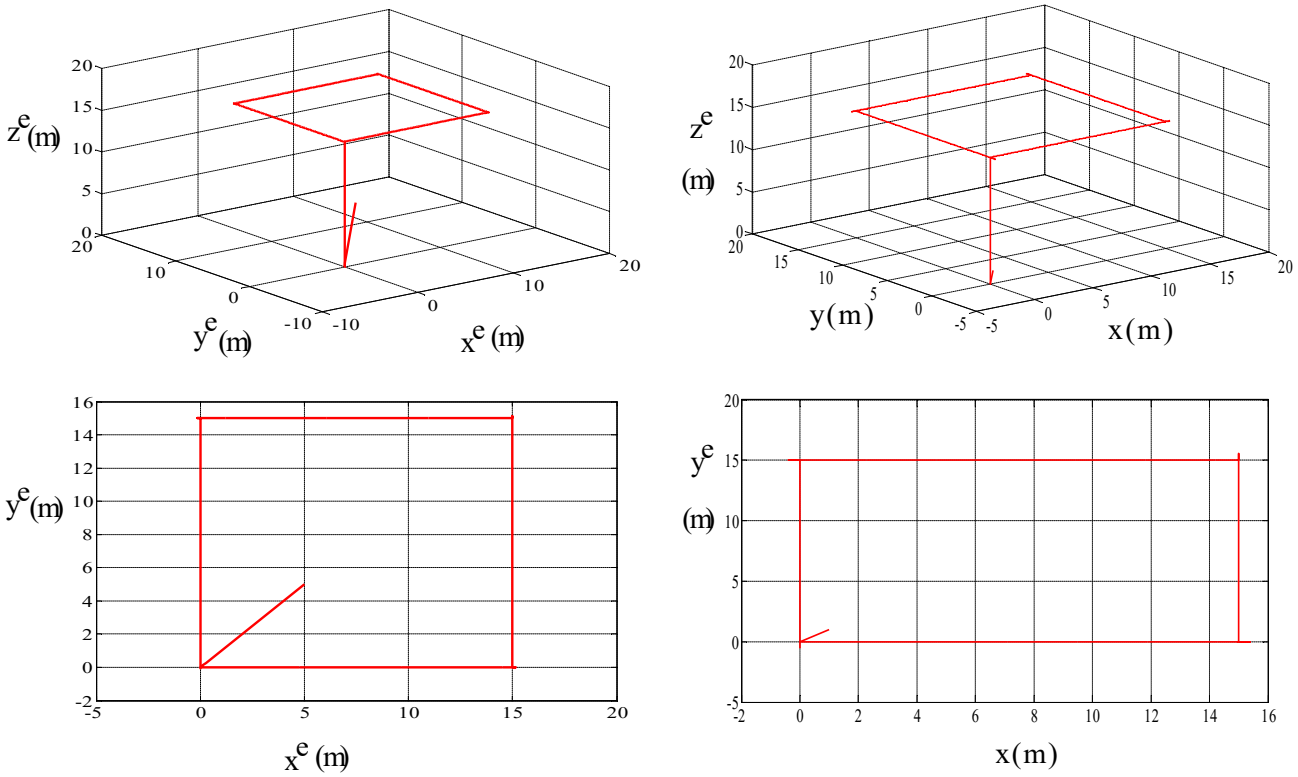


Fig. 3.6. Simulation results of the x position and y position and z position using backstepping control, (a) BSC, (b) SMC

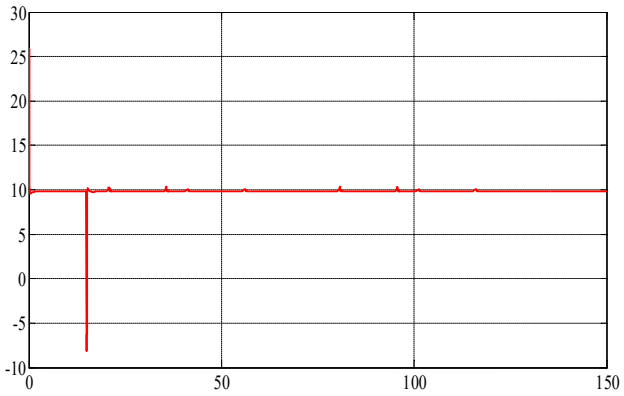
In Fig.3.7 we find the path followed by the quadrotor estimated using extended Kalman filter, where we notice that the quadrotor successfully follows the desired path for the two commands (BSC) and (SMC).



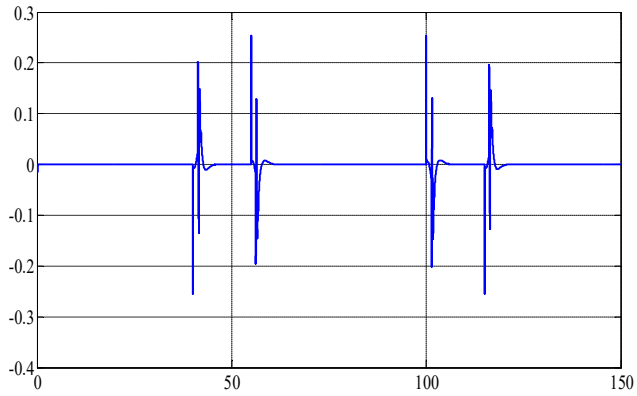
(a) **(b)**

Fig. 3.7. Response of quadrotors model. (a) BSC, (b) SMC.

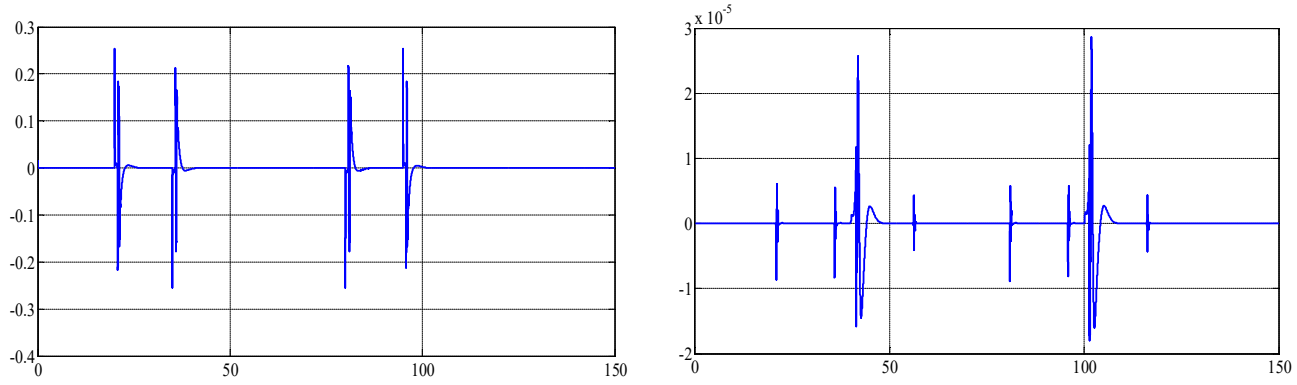
Fig. 3.8 shows the altitude and attitude control inputs of the quadrotor. As shown in the simulation result, the altitude control input u1 is a positive value and close to about 10 N, we see also that the inputs u2, u3 and u4 are reasonable.



Control input U1 for the position



Control input φ for the attitude



Control input Θ for the attitude

Control input ψ for the attitude

Fig. 3.8. Simulation of the control inputs.

The previously discussed results were obtained from Matlab Simulink. The models created in the second chapter were modified by adding EKF block to the already implemented controllers' blocks. The EKF are the blue boxes shown in the following Simulink blocks. Figures 3.9, 3.10, and 3.11 correspond to the modified PDC, SMC and BSC controllers respectively by considering the EKF.

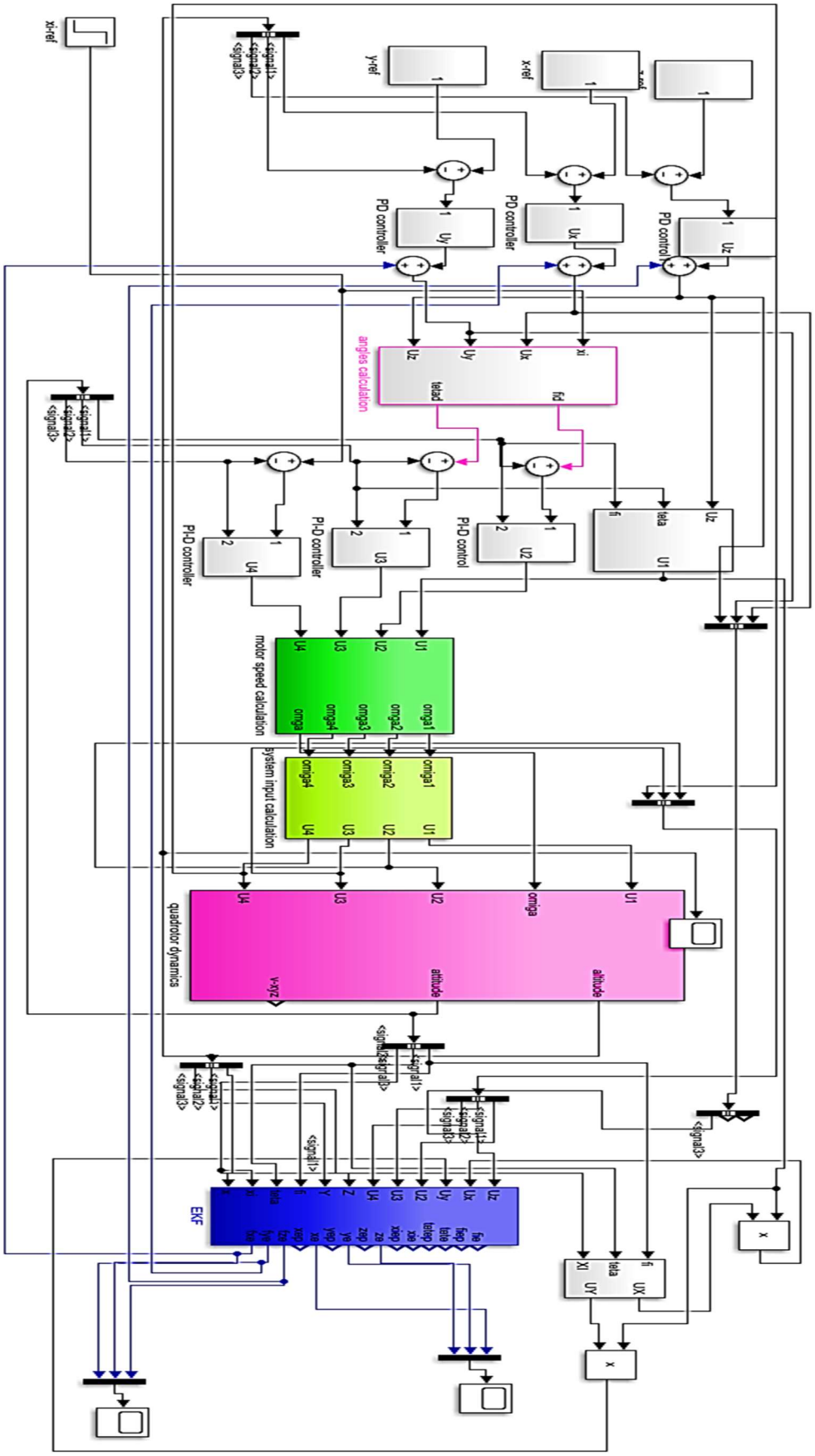


Fig. 3.9. PID with EKF Simulink block.

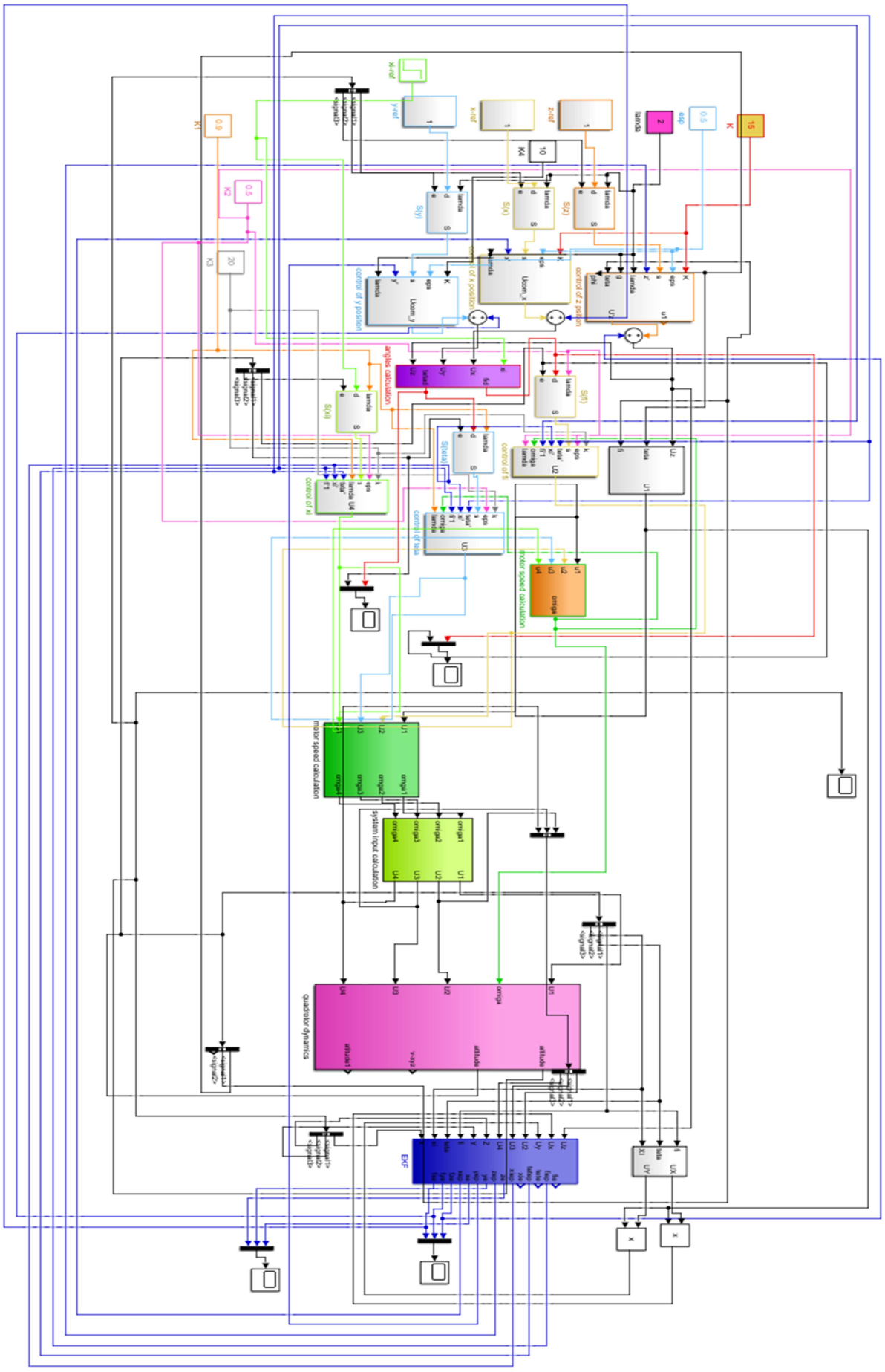


Fig. 3.10. SMC with EKF Simulink model.

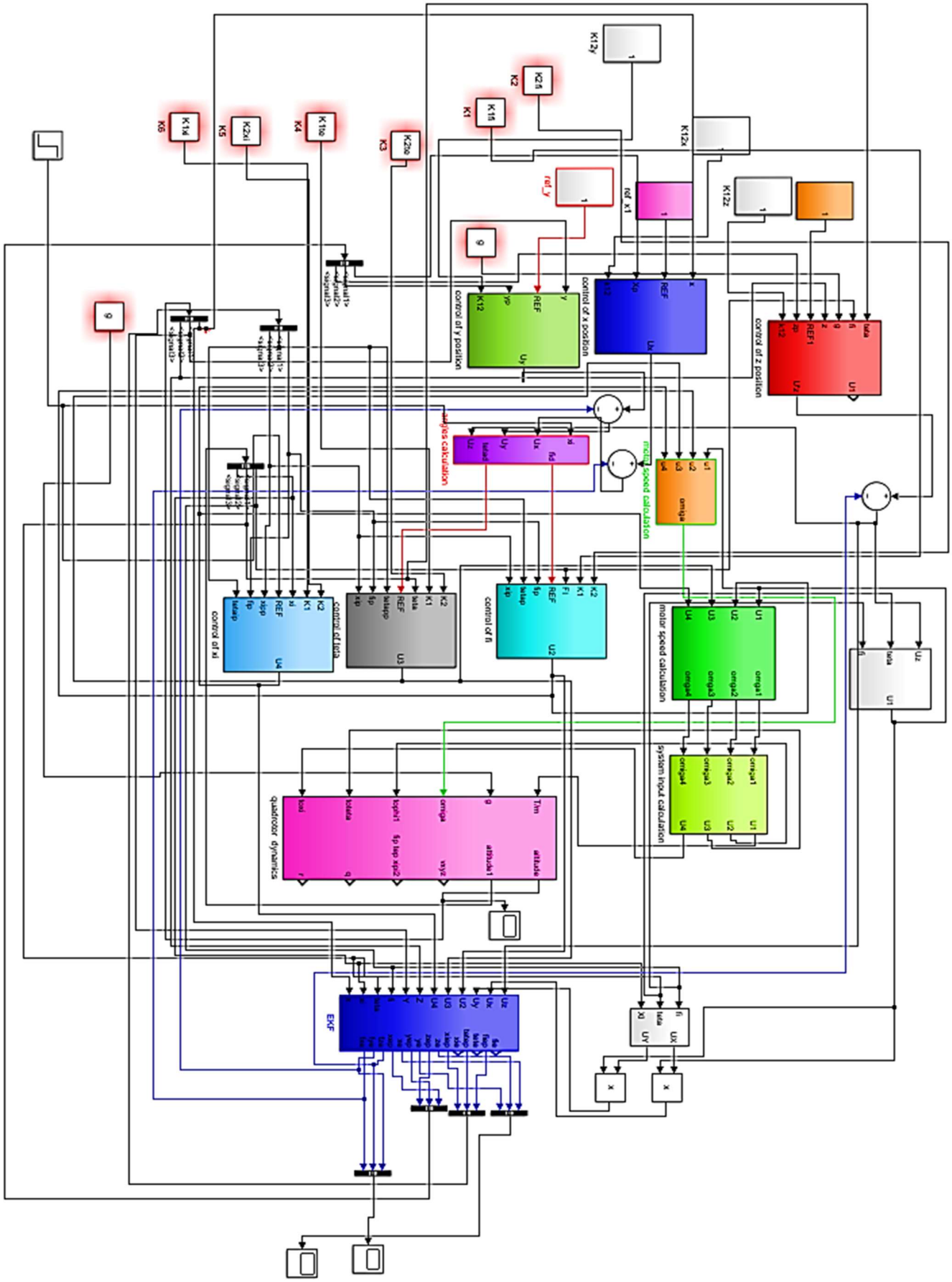


Fig. 3.11. BSC with EKF Simulink model.

III.4. Conclusion

In this chapter we controlled a drone using a nonlinear control method with the aim of keeping all the information on the system and without linearization (backstepping control method, sliding mode control), which is generally used to control high order nonlinear systems. In the control structure the angles (φ, θ and ψ) and the positions (x, y and z) have been picked up directly from the system; while, the angular and linear velocities (p, q, r and x', y', z') were estimated using the extended kalman filter.

The results obtained in this chapter are very successful which has urged us to use the extended kalman filter to estimate the wind disturbance in the next part of our thesis.

Chapter 4: Wind's Force Estimation

The presence of external disturbances such as wind decreases the efficiency of the existing controllers [14-15]; therefore, environmental uncertainties such as turbulence influences and wind disturbances present additional forces and moments on the quadrotor dynamics which were ignored in the previous works [16-18]. Hence, the design of more reliable controller for quadrotor is a challenging task which requires an accurate wind forces estimation using EKF that is proposed in this chapter. Simulation results are shown to verify the efficiency of the presented model which can considerably advance the trajectory tracking feature of the quadrotor under wind disturbances.

IV.1. Wind's force estimation using EKF

The success and efficient quadrotor's trajectory tracking in presence of wind disterbances, model depends on the density of measurements and the wind prediction map accuracy [19]. The wind forecast maps provided present high uncertainty and the wind distribution is really stochastic [19]; this can cause Fixed-wing UAVs with rigid wings and airfoils, for example, drift in the wind direction due to wind thrust [20]. Therefore, in this section of our work, wind's force estimation using EKF and compensated it in real time is provided. The wind's force is break down in three directions x, y and z as shown in Fig. 4.1.

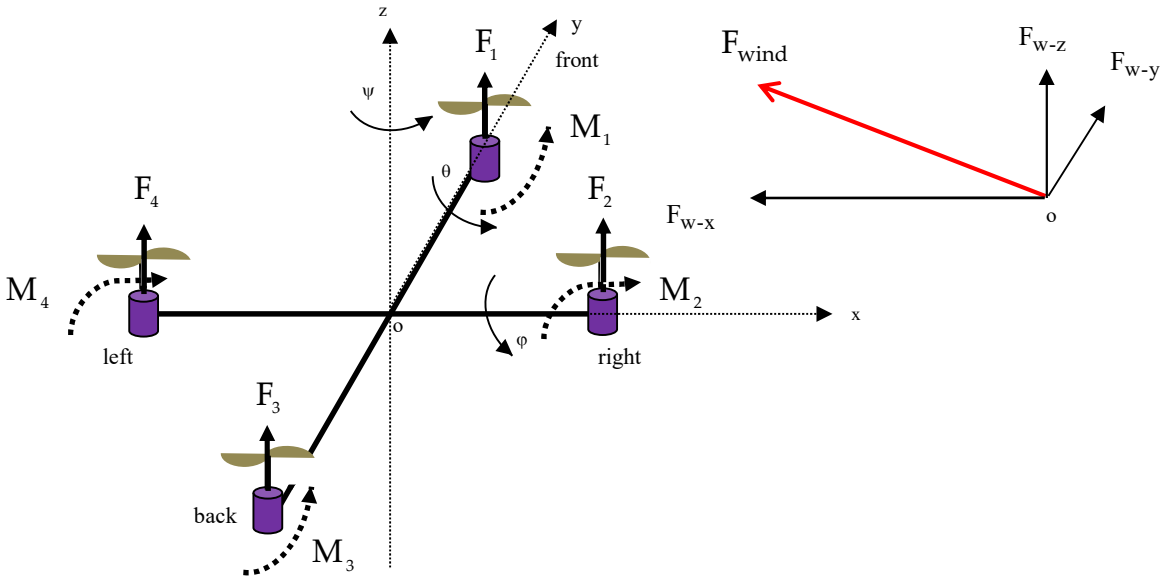


Fig. 4.1. Quadrotor state model including wind's force.

Starting from the newton's second rule $m\vec{\gamma} = \sum \vec{F_i}$, we can get the following equations' system:

$$\begin{cases} \mathrm{m}x^{"} = U_x - f_x x' - F_{w-x} \\ \mathrm{m}y^{"} = U_y - f_y y' - F_{w-y} \\ \mathrm{m}z^{"} = U_z - f_z z' - F_{w-z} \end{cases} \quad (4.1)$$

Therfore, the linear accelerations is expressed by eq. (4.2) as follow:

$$\begin{cases} x^{"} = \frac{1}{m}(U_x - f_x x' - F_{w-x}) \\ y^{"} = \frac{1}{m}(U_y - f_y y' - F_{w-y}) \\ z^{"} = \frac{1}{m}(U_z - f_z z' - F_{w-z}) \end{cases} \quad (4.2)$$

Where:

$\mathrm{F_{w-x}}$ is the wind force in *x* direction

$\mathrm{F_{w-y}}$ is the wind force in *y* direction

$\mathrm{F_{w-z}}$ is the wind force in *z* direction

In the vector form as: $x'(t) = f(x) + g(x)U + g_r(x)$, and in the aim to introduce the wind force in the system, Eq. (4.2) is modified such that: $x_1 = \varphi,\ x_2 = p,\ x_3 = \theta,\ x_4 = q,$

$x_5 = \psi,\ \ x_6 = r,\ \ x_7 = z,\ \ x_8 = z',\ \ x_9 = y,\ \ x_{10} = y',\ \ x_{11} = x,\ \ x_{12} = x',$ and

$x_{13} = F_z,\ x_{14} = F_y,\ x_{15} = F_x.$

Therefore, the system is expressed by Eq. (4.3) as follow:

$$\begin{bmatrix} x_1' \\ x_2' \\ x_3' \\ x_4' \\ x_5' \\ x_6' \\ x_7' \\ x_8' \\ x_9' \\ x_{10}' \\ x_{11}' \\ x_{12}' \\ x_{13}' \\ x_{14}' \\ x_{15}' \end{bmatrix} = \begin{bmatrix} x_2 \\ a_1 x_4 x_6 + a_3 \Omega x_4 - a_{13} x_2^2 \\ x_4 \\ a_4 x_2 x_6 + a_6 \Omega x_2 - a_{14} x_4^2 \\ x_6 \\ a_7 x_2 x_4 - a_{15} x_6^2 \\ x_8 \\ a_9 x_8 + \frac{x_{13}}{m} \\ x_{10} \\ a_{10} x_{10} + \frac{x_{14}}{m} \\ x_{12} \\ a_{11} x_{12} + \frac{x_{15}}{m} \\ 0 \\ 0 \\ 0 \end{bmatrix} + \begin{bmatrix} 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & b_1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & b_2 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & b_3 \\ 0 & 0 & 0 & 0 & 0 & 0 \\ \frac{1}{m} & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & \frac{1}{m} & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & \frac{1}{m} & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 \end{bmatrix} \begin{bmatrix} U_z{}' \\ U_y{}' \\ U_x{}' \\ U_\varphi \\ U_\theta \\ U_\psi \end{bmatrix} + \begin{bmatrix} 0 \\ 0 \\ 0 \\ 0 \\ 0 \\ 0 \\ 0 \\ g \\ 0 \\ 0 \\ 0 \\ 0 \\ 0 \\ 0 \\ 0 \end{bmatrix} \quad (4.3)$$

Such that:

$$f(x) = \begin{bmatrix} x_2 \\ a_1 x_4 x_6 + a_3 \Omega x_4 - a_{13} x_2^2 \\ x_4 \\ a_4 x_2 x_6 + a_6 \Omega x_2 - a_{14} x_4^2 \\ x_6 \\ a_7 x_2 x_4 - a_{15} x_6^2 \\ x_8 \\ a_9 x_8 + \frac{x_{13}}{m} \\ x_{10} \\ a_{10} x_{10} + \frac{x_{14}}{m} \\ x_{12} \\ a_{11} x_{12} + \frac{x_{15}}{m} \\ 0 \\ 0 \\ 0 \end{bmatrix}, \; g(x) = \begin{bmatrix} 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & b_1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & b_2 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & b_3 \\ 0 & 0 & 0 & 0 & 0 & 0 \\ \frac{1}{m} & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & \frac{1}{m} & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & \frac{1}{m} & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 \end{bmatrix} \text{ and } g_r(x) = \begin{bmatrix} 0 \\ 0 \\ 0 \\ 0 \\ 0 \\ 0 \\ 0 \\ g \\ 0 \\ 0 \\ 0 \\ 0 \\ 0 \\ 0 \\ 0 \end{bmatrix}$$

The derivate of *f(x)* with respect to *x* is expressed as:

$$\left[\frac{\partial f}{\partial x}\right] =$$

$$\begin{bmatrix}
0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & -2a_{13} & 0 & \Omega a_2 + a_1 x_6 & 0 & a_1 x_4 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & -\Omega a_4 + a_3 x_6 & 0 & -2a_{14} & 0 & a_3 x_2 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & a_3 x_4 & 0 & a_5 x_2 & 0 & -a_{15} & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & a_9 & 0 & 0 & 0 & 0 & a_{20} & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & a_{10} & 0 & 0 & 0 & a_{20} & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & a_{11} & 0 & 0 & a_{20} \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0
\end{bmatrix}$$

(4.4)

Therefore, $F = I + \left(\frac{\partial f}{\partial x}\right) T_s$ can be calculated from Eq. (4.5) as follow:

$$F =$$

$$\begin{bmatrix}
1 & T_S & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 1 - 2a_{13}T_S & 0 & \Omega a_2 T_S + a_1 x_6 T_S & 0 & a_1 x_4 T_S & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 1 & T_S & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & -\Omega a_4 T_S + a_3 x_6 T_S & 0 & 1 - 2a_{14}T_S & 0 & a_3 x_2 T_S & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 1 & T_S & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & a_3 x_4 T_S & 0 & a_5 x_2 T_S & 0 & 1 - 2a_{15}T_S & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 1 & T_S & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 + a_9 T_S & 0 & 0 & 0 & 0 & a_{20}T_S & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & T_S & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 + a_{10}T_S & 0 & 0 & 0 & a_{20}T_S & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & T_S & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 + a_{11}T_S & 0 & 0 & a_{20}T_S \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1
\end{bmatrix}$$

(4.5)

Whereas, $G = g(x)T_S = \begin{bmatrix} 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & b_1T_S & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & b_2T_S & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & b_3T_S \\ 0 & 0 & 0 & 0 & 0 & 0 \\ \frac{T_S}{m} & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & \frac{T_S}{m} & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & \frac{T_S}{m} & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 \end{bmatrix}$ and $G_r = g_r(x)T_S = \begin{bmatrix} 0 \\ 0 \\ 0 \\ 0 \\ 0 \\ 0 \\ 0 \\ gT_S \\ 0 \\ 0 \\ 0 \\ 0 \\ 0 \\ 0 \\ 0 \end{bmatrix}$

IV.2. Results and discussions

In order to validate our proposed control solution, the previously created block diagram given in Fig. 3.1 is modified such that the wind's force was injected to the system and simulated under Matlab/Simulink software. The modified block is shown in Fig .4.1. Therefore, to test the efficiency of our estimator, a wind force expressed by $\vec{F}_w = d_{1x}\vec{\imath} + d_{1y}\vec{\jmath} + d_{1z}\vec{k}$ is applied at t=50s then another force expressed by $\vec{F}_w = d_{2x}\vec{\imath} + d_{2y}\vec{\jmath} + d_{2z}\overrightarrow{k}$ is applied at t=60s and from t=70s until the end of the simulation (150s) we applied the force $\vec{F}_w = d_{3x}\vec{\imath} + d_{3y}\vec{\jmath} + d_{3z}\vec{k}$. The injected and estimated *x*, *y* and *z* wind's force components are shown in Fig. 4.2.

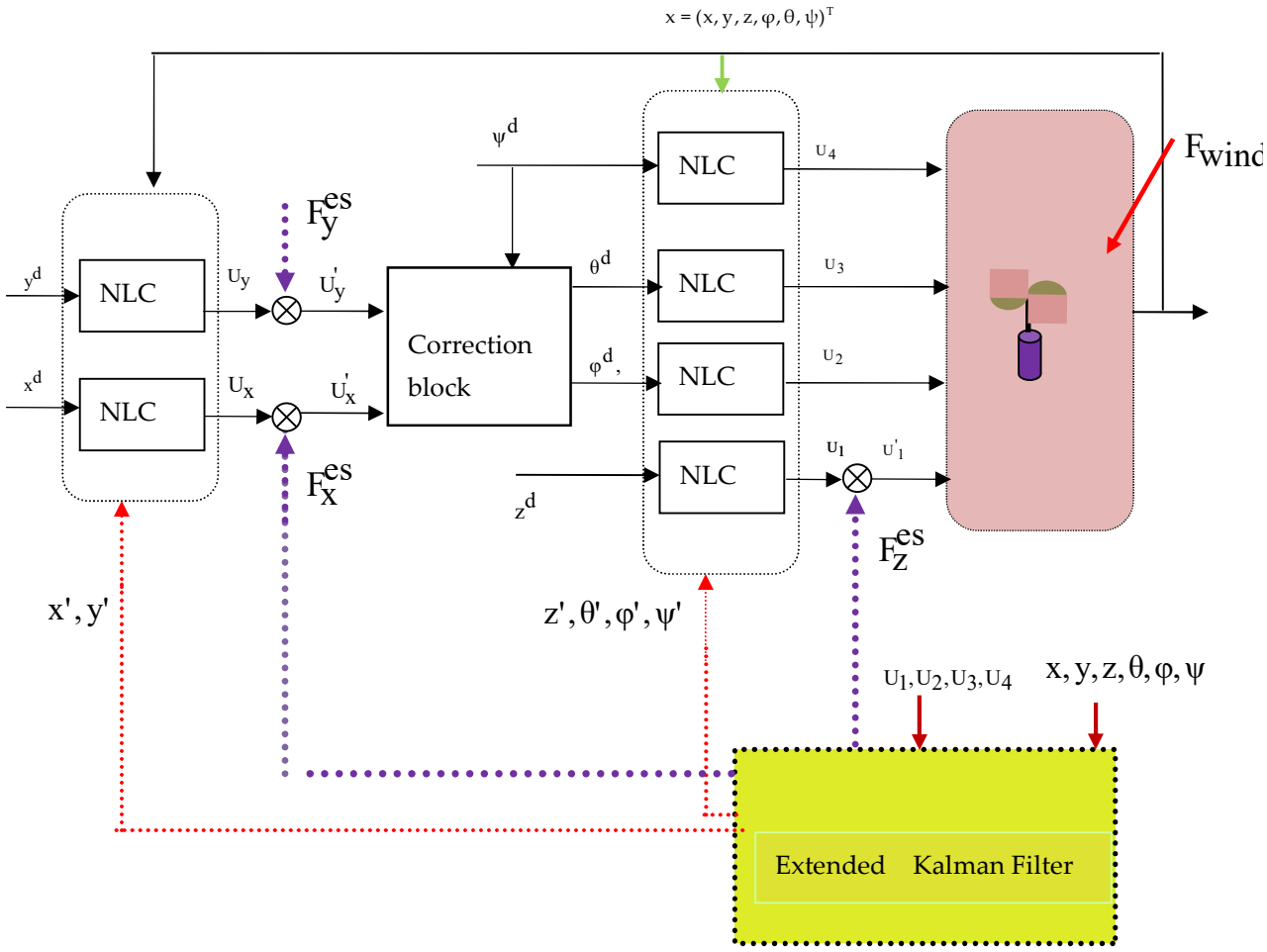


Fig. 4.2. Block diagram of Nonlinear Control of Quadrotor Using the extended Kalman filter for the estimation and compensation of wind's force.

$d_1 = [d_{1x}, d_{1y}, d_{1z}]$ =[5+0.2sin (4t), 5+0.2sin (4t), 5+0.2sin (4t)]

$d_2 = [d_{2x}, d_{2y}, d_{2z}]$= [7+0.2sin (4t), 3+0.2sin (4t), 5+0.2sin (4t)]

$d_3 = [d_{3x}, d_{3y}, d_{3z}]$ =[7+0.2sin (4t), 3+0.2sin (4t), 7+0.2sin (4t)]

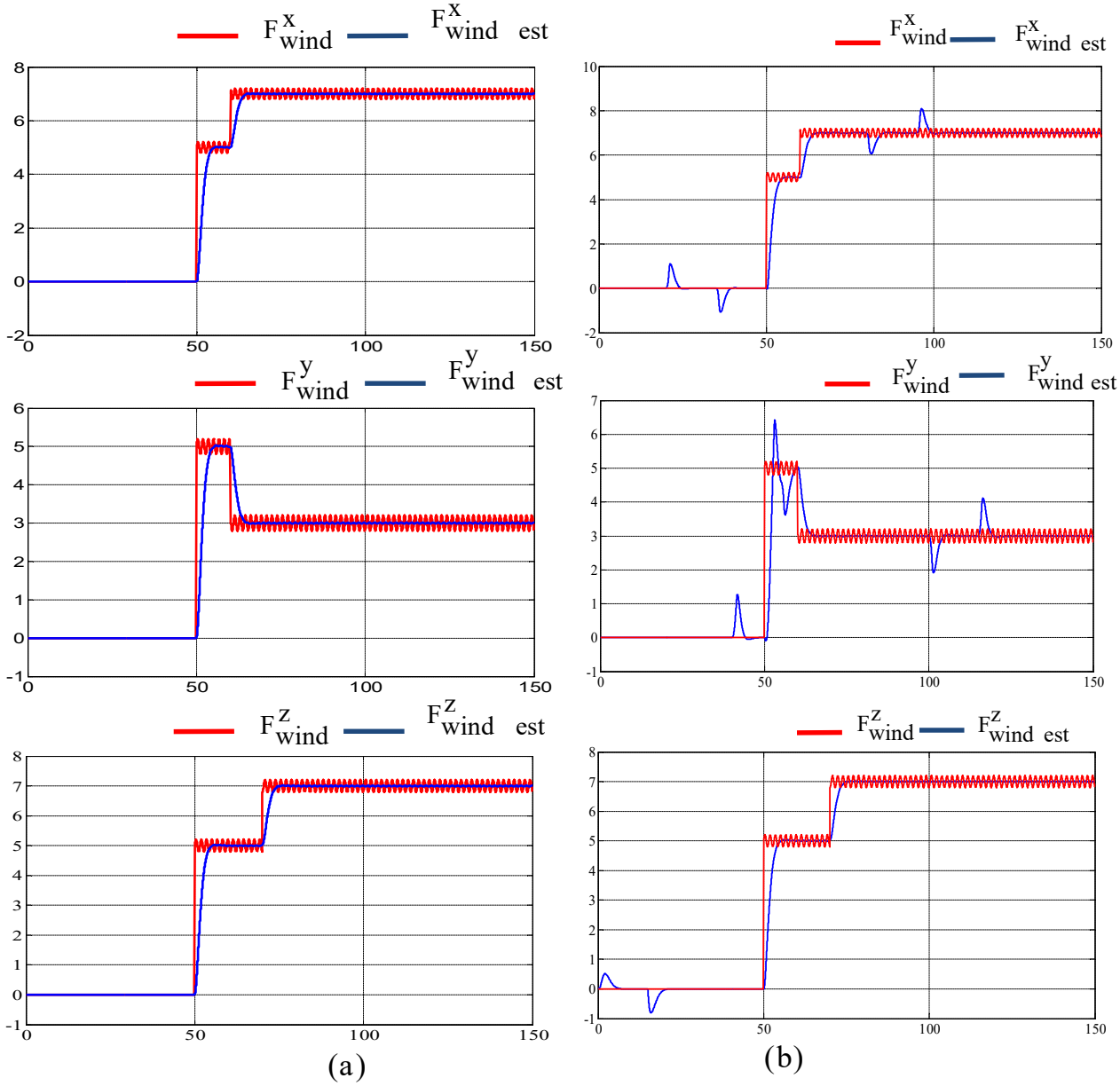


Fig. 4.3. The injected and estimated wind's forces. (a) BSC, (b) SMC.

It can be noticed that $\hat{F}_{Wz}$ successfully follows the component of F_{wz}, and $\hat{F}_{Wx}$ and $\hat{F}_{Wy}$ follow F_{wx} and F_{wy} respectively but with oscillations.

Therefore, the work carried out allows us to estimate the amplitude and the direction of the wind's force at any time. Hence, this makes it possible the compensation the wind's force in real time. The presence of the wind's force disturbs the quadrotor response and causes it to shift from the desired trajectory as it is illustrated in Fig. 4.4. The disturbance error depends on the applied wind's force amplitude.

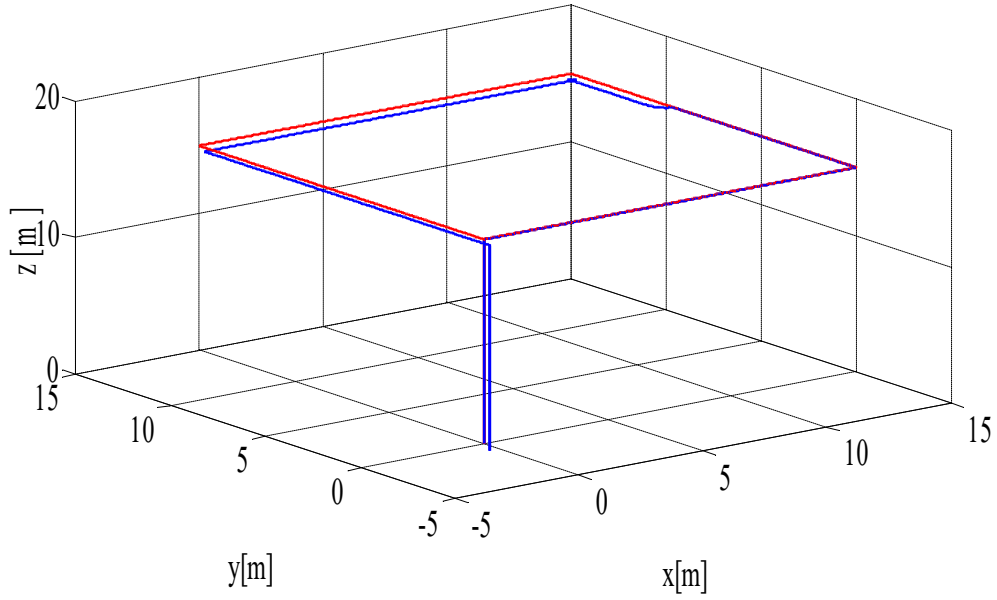


Fig. 4.4. Response of quadrotors model to the wind force without compensation.

Thus, our work objective in this part consists in estimating and compensating this force in order to reach the desired position even in the hazardous conditions. Fig. 4.5 illustrates the wind force compensation. It is clearly shown that the drone position is disturbed for short time (from 50s to 54s); then, the quadrotor will be able to track its desired trajectory perfectly once the wind force is compensated as shown in Fig. 4.5 (b).

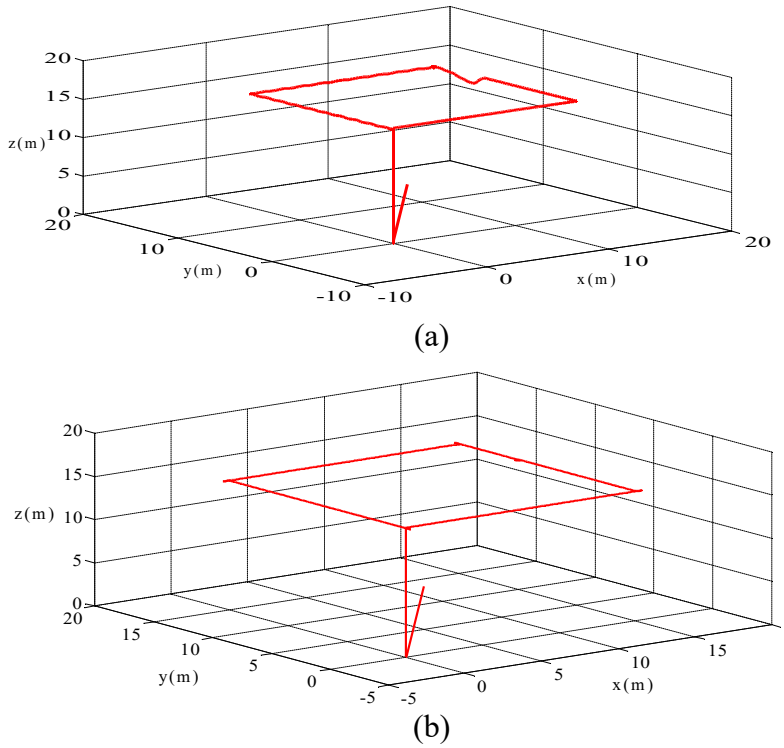
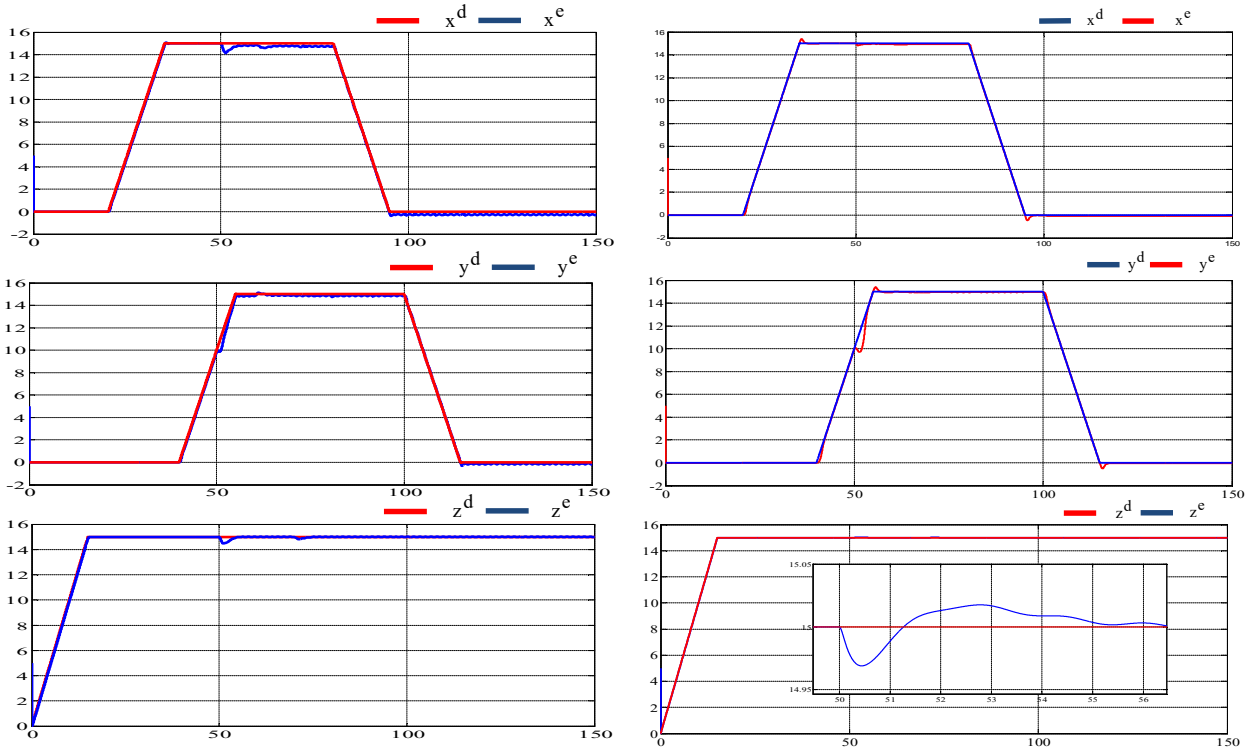
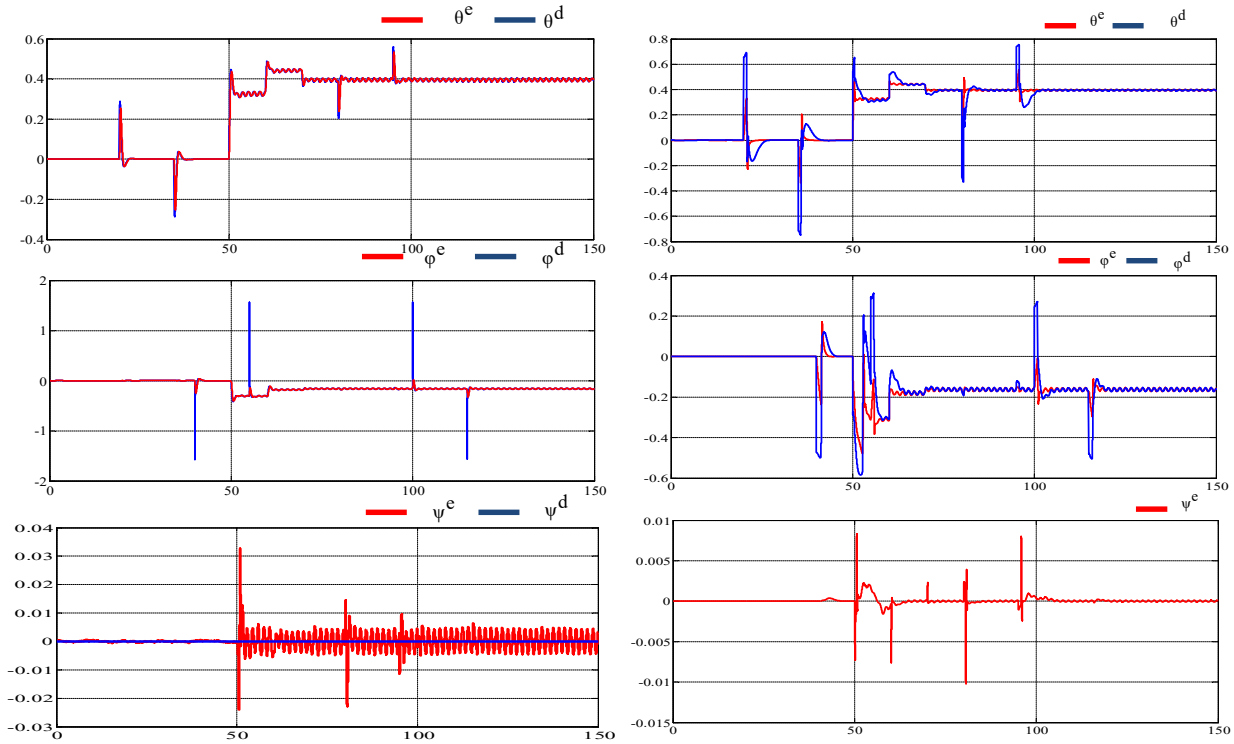


Fig. 4.5. Response of quadrotor model to the wind force with compensation (a) BSC, (b) SMC.



(a) **(b)**

Fig. 4.6. Altitude, heading and position reference measurement vs. actual measurement. (a)BSC, (b) SMC.



(a) **(b)**

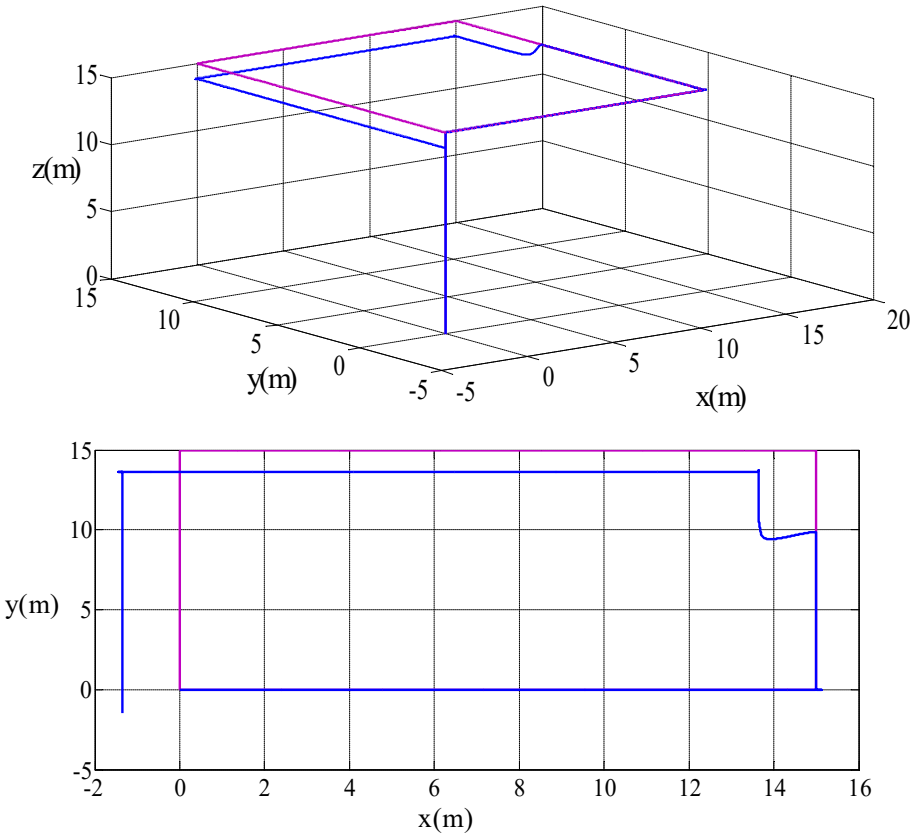
Fig. 4.7.Attitude tracking (φ,θ,ψ). Graph legend: blue Reference Trajectory; Red - Trajectory estimated Using Extended Kalman Filter, (a) BSC, (b) SMC

IV.3. Robustness test of the compensation method:

In order to test the robustness of the compensation method we applied a wind force of ten newton, the test is done in two stages:

First stage : we apply this force at t=50s and see the behavior of the two control methods , in the Fig.4.8 we find the response of the quadrotor model to the disturbance of the wind force without compensation .

Fig.4.8.(a) shows that for the control by backstepping the quadrotor moves away from the desired trajectory but remains contollable.in the other hand for the sliding mode control (fig4.8.(b)), the quadrotor becomes uncontrollable and risks being destroyed.



(a)

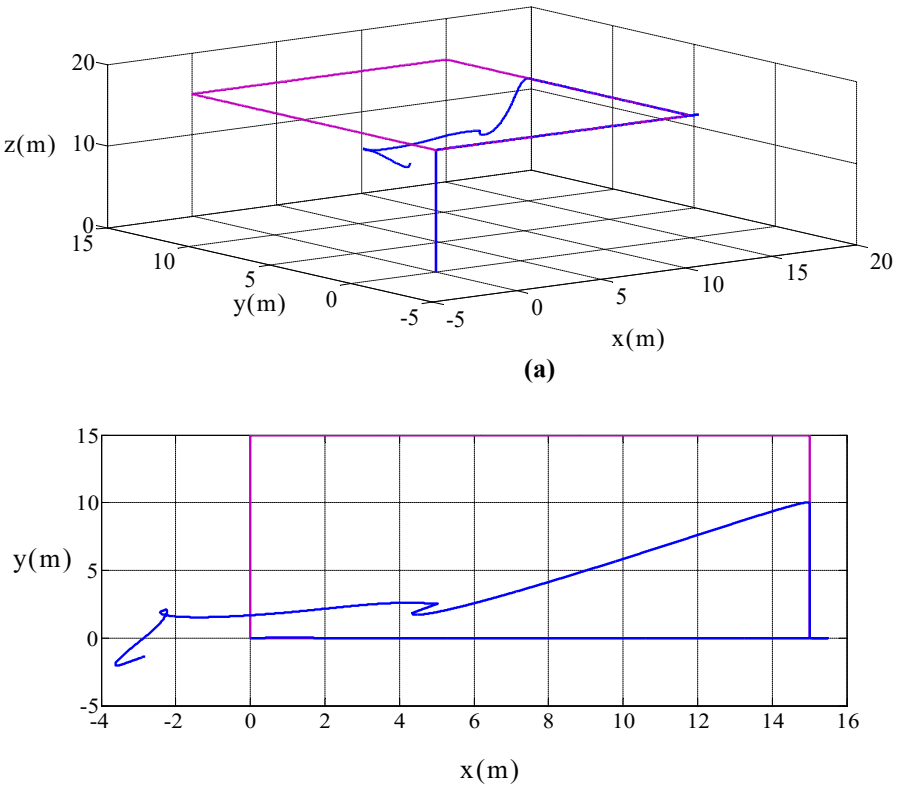


Fig. 4.8 Response of quadrotors model to the wind force without compensation ,(a)BSC, (b) SMC

In the fig.4.9 we find the thrust force U1 and the speed of the first rotor where we notice that they are proportional and that during the application of the wind force the speed increases and generates the increases in the thrust force to overcome the force of the wind.

Second stage: in this step we have compensated the wind force by the force estimated by EKF. In fig. 4.10, we noticed that quadrotor resumes the desired trajectory after a certain time (which depends on the dynamics of the estimator EKF). The robustness of this method is clear and fills the gap left by the sliding mode control.

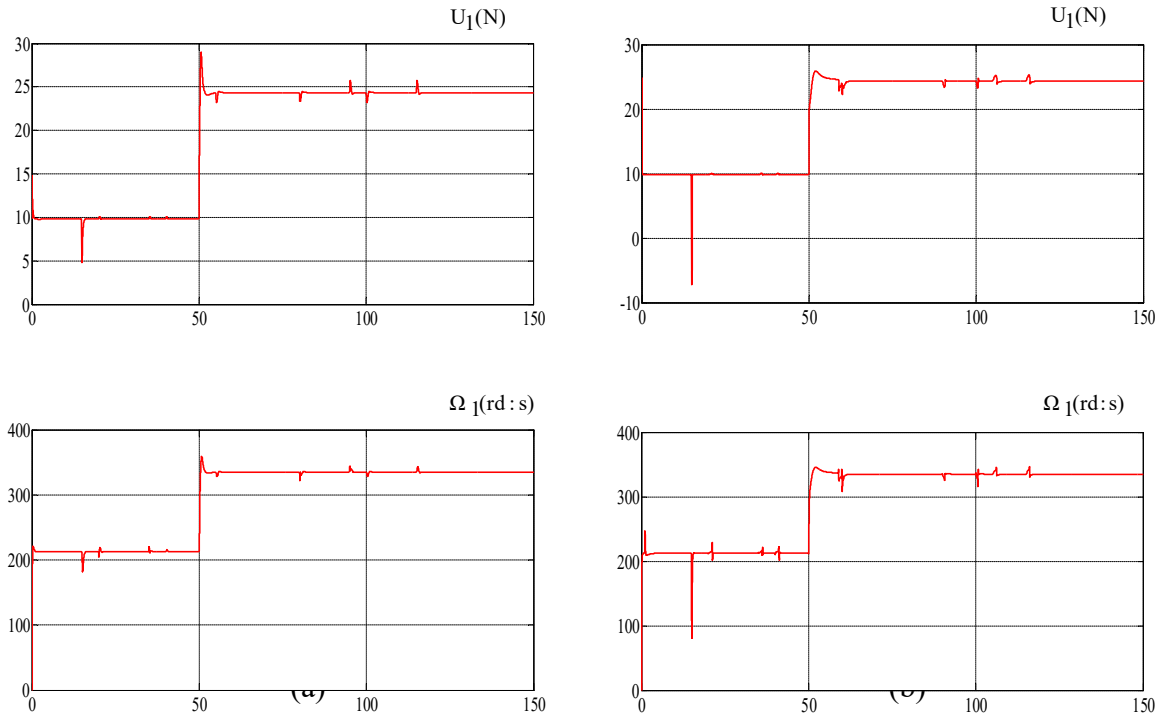


Fig. 4.9 Control U1 and Rotor speed, (a) BSC, (b) SMC.

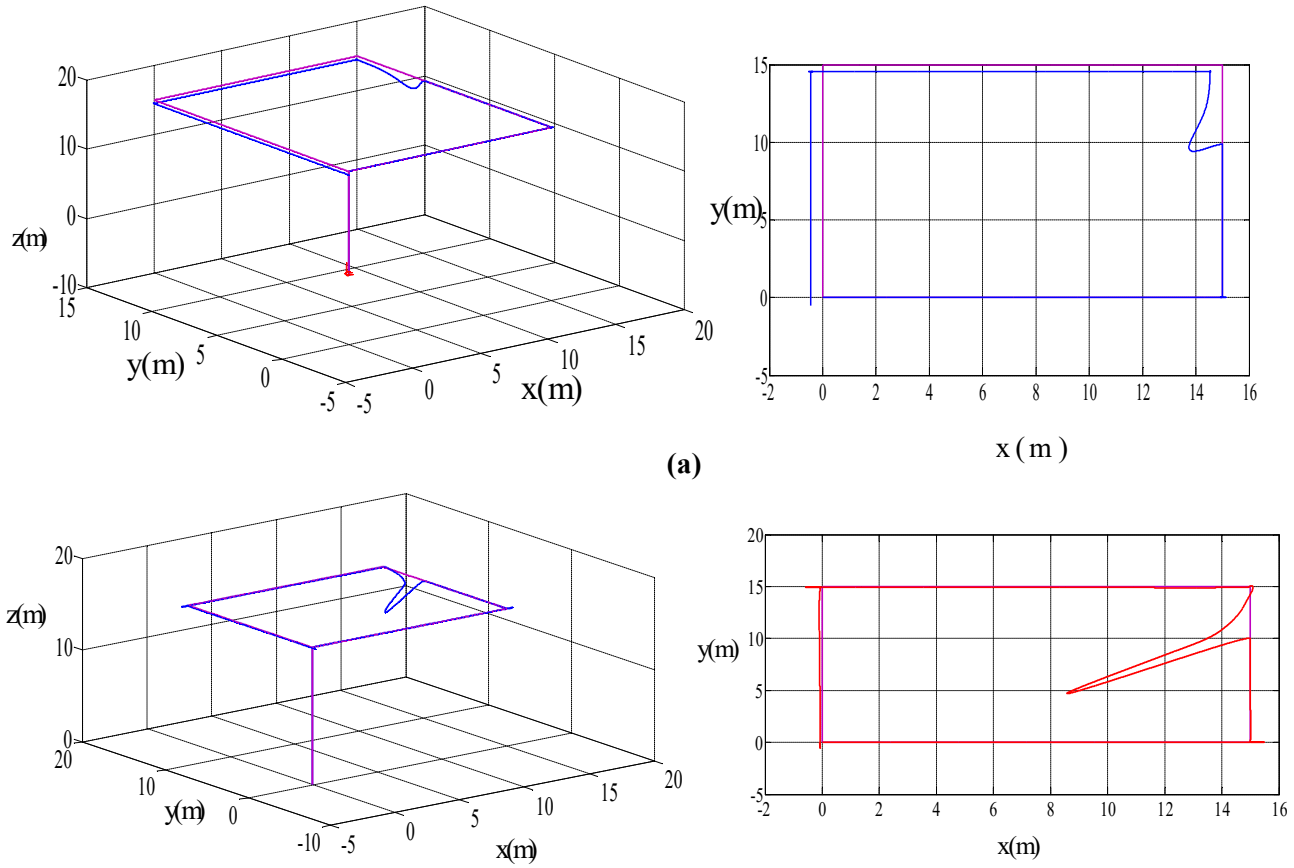


Fig. 4.10 Response of quadrotors model to the wind force with compensation, (a) BSC, (b) SMC

Fig.4.11 shows the attitudes responses, where we notice that the simulation results of the roll angle is the image of the component of the wind force according to the y direction (F_{wy}), and the simulation results of the pitch angle is the image of the component of the wind force according to the x direction($-F_{wx}$).

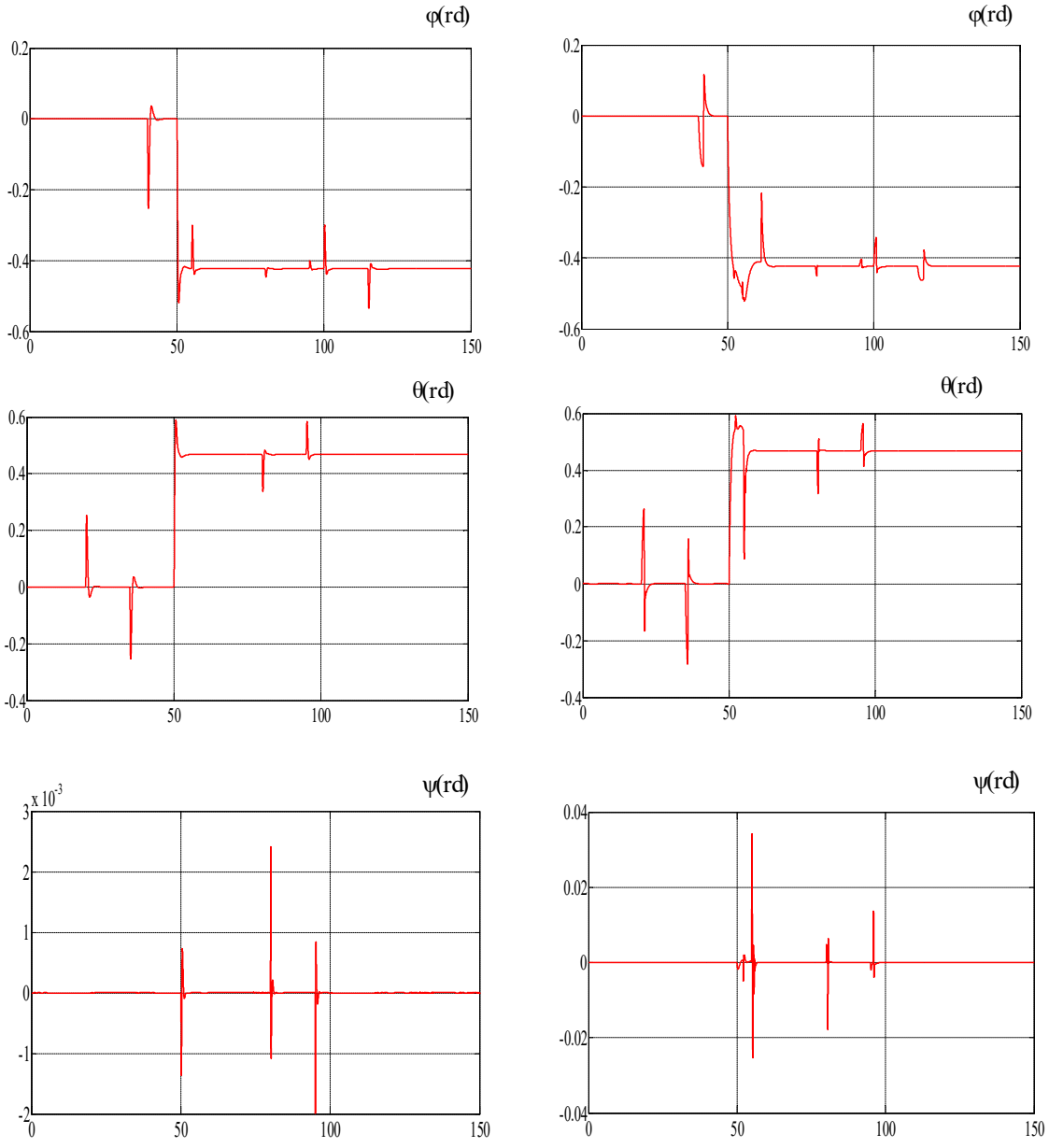


Fig. 4.11 Attitude tracking (φ,θ,ψ). (a)BSC, (b) SMC

IV.4. Conclusion

Kalman filter was used to estimate the wind force; satisfactory results were obtained which allowed us to compensate instantly this force and to reach the desired position with great precision even in unfavorable conditions due to presence of wind disturbances. We concluded that this method is beneficial in the case where the quadrotor undergoes a disturbance of wind force and can bring a complement to the control methods .but like all control methods this technique is limited by the power of the quadrotor and the strength of wind.

Conclusion and Perspectives

The work presented in this thesis aimed to show that it is possible to achieve the desired trajectory by several control and estimation methods even under unfavorable conditions, such is the wind.

In the first chapter we developed the mathematical model of the quadrotors in space(x,y,z) and by using the first and second law of Newton we have obtained the six main equations which the movement of translation and the movement of rotation of quadrone. we ended up with a system of equations of twelve states ,six of which represent the positions (x,y,z) and directions (φ, θ, ψ).,the other six equations represent the linear velocities (x', y', z').and angular velocities $(\varphi', \theta', \psi')$.. This system of equations has been exploited in the remaining chapters.

In the second chapter we realized the back stepping control of the quadrotors, for this we first developed the basic algorithm of the control by back stepping, in the second place we applied this control theory to the model of the quadrone obtained in the first chapter.

To test the reliability of the control structure, we chose a helical trajectory; we have noticed from the results of simulations that the quadrotors quickly reaches the desired trajectory.

Third, we announced the theory of sliding mode control and its application to the quadrotors model. The results of simulations have shown the effectiveness of this control method in achieving the desired trajectory. In order to test the robustness of the two control methods, we injected a disturbance (wind force) and we noticed that the quadrotors moved away from the trajectory under the effect of this force and successfully returned to the desired position after a while hence the robustness of these methods to disturbance rejection.

In the last chapter, we used the extended Kalman filter to estimate the linear and angular velocities to control the quadrotors.

The control structure consists of three position sensors and three direction sensors, the angular and linear velocities were estimated by the extended Kaman filter.

The results showed that the Kalman filter converges rapidly with faster dynamics then that of the system (quadrotors).

In a new approach we assumed that the wind force is a state of the system and we have extended the system states equation to fifteen states, whose three additional states represent the components of the wind force($F_{w-x}, F_{w-y}, F_{w-z}$).

The results showed that the Kalman filter allowed to instantaneously estimating the components of the wind force with an acceptable error, and then we proposed in a new initiative to compensate this force and allow the quadrotors to reach its desired trajectory successfully.

In the future work we hope to practically realize the compensation of the force of the wind; as well as, a tolerant control of the loss of one of four motors and to realize the control of the quadrotor through a vision camera to avoid unexpected obstacles.

References

[1] Kurak, S., Hodzic, M., “Control and Estimation of a Quadcopter Dynamical Model”, Periodicals of Engineering and Natural Sciences, Vol.6, No.1, March 2018, pp. 63–75. DOI: 10.21533/pen.v6i1.164.

[2] N. Hanani, F. Syazwanadira, N. A. Fakharulrazi, F. Yakub, Z. A. Rasid and S. Sarip, "Full Control of Quadrotor Unmanned Aerial Vehicle using Multivariable Proportional Integral Derivative Controller", 2019 IEEE 9th International Conference on System Engineering and Technology (ICSET), 2019, pp. 447-452, doi: 10.1109/ICSEngT.2019.8906418.

[3] H. S. Paing, A. V. Schagin, K. S. Win and Y. H. Linn, "New Designing Approaches for Quadcopter Using 2D Model Modelling a Cascaded PID Controller," 2020 IEEE Conference of Russian Young Researchers in Electrical and Electronic Engineering (EIConRus), 2020, pp. 2370-2373, doi: 10.1109/EIConRus49466.2020.9039395.

[4] B. L. Tian, J. Cui, H. C. Lu, Z. Y. Zuo, and Q. Zong, “Adaptive finite-time attitude tracking of quadrotors with experiments and comparisons”, IEEE Trans. Ind. Electron., vol. 66, no. 12, pp. 9428–9438, Dec. 2019, doi: 10.1109/TIE.2019.2892698.

[5] J. Cariño, H. Abaunza and P. Castillo, "Quadrotor quaternion control," *2015 International Conference on Unmanned Aircraft Systems (ICUAS)*, 2015, pp. 825-831, doi: 10.1109/ICUAS.2015.7152367.

[6] Faraz Ahmad, Pushpendra Kumar, Anamika Bhandari, Pravin P. Patil, “Simulation of the Quadcopter Dynamics with LQR based Control”, Materials Today: Proceedings, Volume 24, Part 2, 2020, Pages 326-332, https://doi.org/10.1016/j.matpr.2020.04.282.

[7] L. -X. Xu, H. -J. Ma, D. Guo, A. -H. Xie and D. -L. Song, "Backstepping Sliding-Mode and Cascade Active Disturbance Rejection Control for a Quadrotor UAV," in IEEE/ASME Transactions *on Mechatronics*, vol. 25, no. 6, pp. 2743-2753, Dec. 2020, doi: 10.1109/TMECH.2020.2990582.

[8] Saibi, A.; Boushaki, R.; Belaidi, H., “Backstepping Control of Drone”. Eng. Proc. 2022, 14, 4. https://doi.org/10.3390/engproc2022014004.

[9] Brice J. Njinwoua, Alain Vande Wouwer, “Cascade attitude control of a quadcopter in presence of motor asymmetry”, University of Mons, Mons, Belgium, IFAC-PapersOnLine, Volume 51, Issue 4, 2018, Pages 113-118, DOI: 10.1016/j.ifacol.2018.06.055.

[10] Cardimen, S., Zohdy, M.A., Sababha, M. and Rusek, A. (2018) “Robust integral back-stepping control for a quadrotor: experiments and simulation”, Int. J. Intelligent Systems Design and Computing, Vol. 2, Nos. 3/4, pp.297–312. 10.1504/IJISDC.2018.097473.

[11] Vasko Sazdovski, Tatjana Kolemishevska-Gugulovska, Mile Stankovski, “Kalman Filter Implementation For Unmanned Aerial Vehicles Navigation Developed Within a Graduate Course”, IFAC Proceedings Volumes, Volume 38, Issue 1, 2005, Pages 12-17, https://doi.org/10.3182/20050703-6-CZ-1902.02265.

[12] Gabriel Schmitz, Tiago Alves, Renato Henriques, Edison Freitas, Ebrahim El'Youssef, “A simplified approach to motion estimation in a UAV using two filters”, IFAC-PapersOnLine, Volume 49, Issue 30, 2016, Pages 325-330, https://doi.org/10.1016/j.ifacol.2016.11.156.

[13] C. Luo, S. I. McClean, G. Parr, L. Teacy and R. De Nardi, "UAV Position Estimation and Collision Avoidance Using the Extended Kalman Filter," in IEEE Transactions on Vehicular Technology, vol. 62, no. 6, pp. 2749-2762, July 2013, doi: 10.1109/TVT.2013.2243480.

[14] H. Liu, T. Ma, F. L. Lewis, and Y. Wan, “Robust formation control for multiple quadrotors with nonlinearities and disturbances”, IEEE transactions on cybernetics, vol. 50, no. 4, pp. 1362–1371, 2020.

[15] Sheng, S.; Sun, C., “Control and Optimization of a Variable-Pitch Quadrotor with Minimum Power Consumption”, Energies 2016, 9, 232. https://doi.org/10.3390/en9040232.

[16] Yao Lei, Yiyong Huang, Hengda Wang, "Effects of Wind Disturbance on the Aerodynamic Performance of a Quadrotor MAV during Hovering", Journal of Sensors, vol. 2021, Article ID 6681716, 13 pages, 2021. https://doi.org/10.1155/2021/6681716.

[17] Bohang Wang, Zain Anwar Ali, Daobo Wang, "Controller for UAV to Oppose Different Kinds of Wind in the Environment", Journal of Control Science and Engineering, vol. 2020, Article
ID 5708970, 10 pages, 2020. https://doi.org/10.1155/2020/5708970.

[18] Bo Hang Wang, Dao Bo Wang, Zain Anwar Ali, Bai Ting Ting, Hao Wang, "An overview of various kinds of wind effects on unmanned aerial vehicle", Measurement and Control 2019, Volume: 52 issue: 7-8, page(s): 731-739, https://doi.org/10.1177/0020294019847688.

[19] Guo, Y., Zhang, R. & Li, H. (2016, July). Robust trajectory control for quadrotors with disturbance observer. 2016 35th Chinese Control Conference (CCC), pp. 10788-10794. doi: 10.1109/ChiCC.2016.7555067.

[20] K. Guo, J. Jia, X. Yu et al "Multiple observers based anti-disturbance control for a quadrotor UAV against payload and wind disturbances". Control Engineering Practice 102 (2020) 104560

[21] Georgios A. Thanellas, Vassilis C. Moulianitis, Nikos A. Aspragathos, "A spatially wind aware quadcopter (UAV) path planning approach", IFAC-PapersOnLine, Volume 52, Issue 8, 2019, Pages 283-288, https://doi.org/10.1016/j.ifacol.2019.08.084.

[22] Jayaweera, H.M.P.C.; Hanoun, S. "Path Planning of Unmanned Aerial Vehicles (UAVs) in Windy Environments", Drones 2022, 6, 101. https://doi.org/10.3390/ drones6050101.

[23] Abdel-Razzak MERHEB, "Diagnostic and Fault-Tolerant Control Applied to an Unmanned Aerial Vehicle", PhD thesis, Aix Marseille Université ED 184, École Doctorale En Mathematiques Et Informatique, 2016.

[24] Faizan Shahid NUST PNEC Pakistan Muhammad Bilal Kadri, Nasir Aziz Jumani, Zaid Pirwani PAF KIET Pakistan, "Dynamical Modeling and Control of quadrotor,Transactions on Machine Design ", Volume 4 Number 2 August 2016

[25] Erginer, B. and Altug, E. (2007). "Modeling and pd control of a quadrotor vtol vehicle". In IEEE Intelligent Vehicles Symposium, 894–899. IEEE.

[26] S. Ahmed d, M. Dubakh, and A. Abbas. « Design of Optimal Pid Controller Using Genetic Algorithm. » In: Al-Mansour 15 (2011)

[27] ArmanKiani-B and NaserPariz. « Fractional PID Controller Design based on Evolutionary Algorithms for Robust two-inertia Speed Control. » In: First Joint Congress on Fuzzy and Intelligent Systems. Ferdowsi University of Mashhad, Iran. Iran, Aug. 2007.

[28] A. Bagis. « Determination of the PID Controller Parameters by Modified Genetic Algorithm for Improved Performance. » In: Journal Of Information Science And Engineering 23 (2007), pp. 1469–1480.

[29] McKerrow P. Modelling the Draganflyer four-rotor helicopter". Proceedings of the IEEE International Conference on Robotics and Automation. 2004; 3596-3601.

[30] P. Wang, Z. Man, Z. Cao, J. Zheng and Y. Zhao, "Dynamics modelling and linear control of quadcopter," 2016 International Conference on Advanced Mechatronic Systems (ICAMechS), Melbourne, VIC, Australia, 2016, pp. 498-503.

[31] Mian, A. and Daobo, W. (2008). " Modeling and backstepping-based nonlinear control strategy for a 6 dof quadrotor helicopter". Chinese Journal of Aeronau- tics, 21(3), 261–268.

[32] Bouabdallah, S. and Siegwart, R. (2005). "Backstepping and sliding mode techniques applied to an indoor mi- cro quadrotor". In IEEE International Conference on Robotics and Automation, 2259–2264. IEEE.

[33] Glida, H.E., Abdou, L., Chelihi, A. et al. "Optimal model-free backstepping control for a quadrotor helicopter". Nonlinear Dyn 100, 3449–3468 (2020).

[34] Basri M. A. M., Noordin A. "Optimal backstepping control of quadrotor UAV using gravitational search optimization algorithm", Bulletin of Electrical Engineering and Informatics, Vol. 9, No. 5, October 2020, pp. 1819-1826

[35] Wang,J., He, S. & Lin, D. (2016). Robust backstepping control for a class of nonlinear systems using generalized disturbance observer. International Journal of Control, Automation and Systems, 14(6), 1475–1483. doi: 10.1007/s12555-014-0401-

[36] Obaid, M. A. M., Husain, A. R. & Al-Kubati, A. A. M. (2016). Robust Backstepping Tracking Control of Mobile Robot Based on Nonlinear Disturbance Observer. International Journal of Electrical and Computer Engineering, 6(2), 901

[37] Madani, T. & Benallegue, A. (2006, Oct). Backstepping Control for a Quadrotor Helicopter. 2006 IEEE/RSJ International Conference on Intelligent Robots and Systems, pp. 3255- 3260. doi: 10.

[38] Ali Saibi, Hadjira Belaidi, Razika Boushaki, Recham Zine Eddine, Amrouche Hafid "Enhanced backstepping control for disturbances rejection in quadrotors". Bulletin of Electrical Engineering and Informatics Vol. 11, No. 6, December 2022, pp. 3201~3216

[39] H. Bouadi, M. Bouchoucha, and M. Tadjine "Sliding Mode Control based on Backstepping Approach for an UAV Type-Quadrotor",World Academy of Science, Engineering and Technology International Journal of Mechanical and Mechatronics Engineering Vol:1, No:2, 2007

[40] Nour BEN AMMAR, Soufiene BOUALL`EGUE and Joseph HAGG`EGE , "Modeling and Sliding Mode Control of a Quadrotor Unmanned Aerial Vehicle " 3rd International Conference on Automation, Control, Engineering and Computer Science (ACECS'16)

[41] Daewon Lee, H. Jin Kim*, and Shankar Sastry, "Feedback Linearization vs. Adaptive Sliding Mode Control for a Quadrotor Helicopter", international conferance an automation, control; International Journal of Control, Automation, and Systems (2009).

[42] H. Mohamed, S. Yang, and M. oghavvemi, "Sliding mode controller design for a flying quadrotor with simplified action planner," in Iccas-sice, 2009. IEEE, 2009, pp. 1279 – 1283.

[43] Luque-Vega, L., Castillo-Toledo, B. & Loukianov, A. G. (2012). Robust block second sliding mode control for a quadrotor. Journal of the Franklin Institute, 349(2), 719 - 739. doi: https://doi.org/10.1016/j.jfranklin.2011.10.017. Advances in Guidance and of Aerospace Vehicles using Sliding Mode Control and Observation Techniques.

[44] Levant,A. (2003). Higher-order sliding modes, differentiation and outputfeedback control. International Journal of Control, 76(9-10), 924-941. doi: 10.1080/0020717031000099029.

[45] Ginoya, D., Shendge, P. D. & Phadke, S. B. (2014). Sliding Mode Control for Mismatched Uncertain Systems Using an Extended Disturbance Observer. IEEE Transactions on Industrial Electronics, 61(4), 1983-1992. doi: 10.1109/TIE.2013.2271597

[46] Moreno, J. A. & Osorio, M. (2008, Dec). A Lyapunov approach to second-order sliding mode controllers and observers. 2008 47th IEEE Conference on Decision and Control, pp. 2856-2861. doi: 10.1109/CDC.2008.4739356.

[47] Runcharoon, K. & Srichatrapimuk, V. (2013, May). Sliding Mode Control of quadrotor. 2013 The International Conference on Technological Advances in Electrical, Electronics and Computer Engineering (TAEECE), pp. 552-557. doi: 10.1109/TAEECE.2013.6557334.

[48] X. Yu, S. Wei, and L. Guo. « Cascade Sliding Mode Control for Bicycle Robot. » In: International Conference on Artificial Intelligence and Computational Intelligence (Oct. 2010), pp. 62–66

[49] Yang, J., Li, S. & Yu, X. (2013). Sliding-Mode Control for Systems With Mismatched Uncertainties via a Disturbance Observer. IEEE Transactions on Industrial Electronics, 60(1), 160-169. doi: 10.1109/TIE.2012.2183841

[50] Wadoo, S. A. (2013). Sliding Mode Control of Crowd Dynamics. IEEE Transactions on Control Systems Technology, 21(3), 1008-1015. doi: 10.1109/TCST.2012.2196700.

[51] H. Bouadi, S. Cunha, A. Drouin, and F. Mora, "Adaptive sliding mode control for quadrotor attitude stabilization and altitude tracking," in Computational Intelligence and Informatics (CINTI), 2011 IEEE 12th International Symposium on. IEEE, 2011, pp. 449 – 455.

[52] L. Tuan and S. Won, "PID based sliding mode controller design for the micro quadrotor," in Control, Automation and Systems (ICCAS), 2013 13th International Conference on. IEEE, 2013, pp. 1860 – 1865.

[53] L Cedro, K Wieczorkowski , "optimizing PID controller gains to model the performance of quadrotor", 13th International Conference on. Sustainable, Modern and Safe Transport (TRANSCOM 2019), Transportation Research Procedia 40 (2019) 156-169.

[54] Z Zuo,"Trajectory tracking control design with command filtered compensation for quadrotor ", IET Control theory appl , 2010,vol 4,iss.11 ,pp.2343-2355.

[55] Nagui, N., Attallah, O., Zaghloul, M.S. et al. Improved GPS/IMU Loosely Coupled Integration Scheme Using Two Kalman Filter-based Cascaded Stages. Arab J Sci Eng 46, 1345–1367 (2021). https://doi.org/10.1007/s13369-020-05144-8.

[56] Fethalla, N,Saad M,Michalska ,H&Ghommam,J.(2018).Robust Observer-Based Dynamic Sliding Mode Controller UAV.IEEE Access, 6 ,45846-45859. Doi: 10 .1109/ACCESS .2018.2866208.

[57] V.G. Adir, A.M. Stoica, A. Marks, et al. « Modelling, stabilization and single motor failure recovery of a 4Y octorotor. » In: In Proc. 13th IASTED International Conference on Intelligent Systems and Control (ISC 2011). Cambridge, U.K., July 2011

[58] A.Das , F.L.Lewis , and K.Subbarao," Sliding Mode Approach to Control Using Dynamic Iversion " in:challenges and paradigms in Applied Robust Control , Robust Control , Book 3 ed .A.Lazinica inTech publishing .Rijecka, Croatia ,2011.

[59] K.Benzemrane, G.L.Santosuosso, and G Damm "Unmznned aerial vehicle speed estimation via nonlinear adaptive observers ". In Proc.of the 2007 american control conference ACC'2007, pages 985-990, New York City, USA, July 11-13 2007.

Appendices

Appendix A:

Correction block

The desired angles θ^d and φ^d are given by the relations () and () indicated in [38]

$$\theta^d = arctg\left(\frac{U_x\cos(\psi) + U_y\sin(\psi)}{U_z + g}\right)$$

$$\varphi^d = arctg\left(m\frac{U_x\sin(\varphi) - U_y\cos(\psi)}{U_1}\right)$$

With $\left(\frac{U_1}{m}\right)^2 = U_x^2 + U_y^2 + (U_z + g)^2$

$$U_z + g = \frac{U_1}{m}\cos(\varphi)\cos(\theta)$$

$$U_x = \frac{U_1}{m}(\cos(\varphi)\sin(\theta)\cos(\psi) + \sin(\varphi)\sin(\psi))$$

$$U_y = \frac{U_1}{m}(\cos(\varphi)\sin(\theta)\sin(\psi) - \cos(\psi)\sin(\varphi))$$

We multiply the expression of U_x by $\cos(\psi)$ and we obtain

$$U_x\cos(\psi) = \frac{U_1}{m}(\cos(\varphi)\sin(\theta)\cos^2(\psi) + \sin(\varphi)\sin(\psi)\cos(\psi))$$

We multiply the expression of U_y by $\sin(\psi)$ and we obtain

$$U_y\sin(\psi) = \frac{U_1}{m}(\cos(\varphi)\sin(\theta)\sin^2(\psi) - cos(\psi)\,sin(\varphi)\sin(\psi))$$

the sum of the two expressions gives

$$U_x\cos(\psi) + U_y\sin(\psi) = \frac{U_1}{m}\cos(\varphi)\sin(\theta)$$

Dividing by $U_z + g$ we obtain

$$\frac{U_x\cos(\psi) + U_y\sin(\psi)}{U_z + g} = \frac{\cos(\varphi)\sin(\theta)}{\cos(\varphi)\cos(\theta)} = tg(\theta)$$

and the desired pitch angle will be expressed by

$$\theta^d = arctg\left(\frac{U_x\cos(\psi) + U_y\sin(\psi)}{U_z + g}\right)$$

We multiply the expression of U_x by $\sin(\psi)$ and we obtain

$$U_x\sin(\psi) = \frac{U_1}{m}(\cos(\varphi)\sin(\theta)\cos(\psi)\sin(\psi) + \sin(\varphi)\sin^2(\psi))$$

We multiply the expression of U_y by $\cos(\psi)$ and we obtain

$$U_y\cos(\psi) = \frac{U_1}{m}\left(\cos(\varphi)\sin(\theta)\sin(\psi)\cos(\psi) - \cos^2(\psi)\sin(\varphi)\right)$$

subtracting the two expressions gives

$$\left(U_x \sin(\psi) - U_y \cos(\psi)\right) = \frac{U_1}{m}\sin(\varphi)$$

and the desired roll angle will be expressed by

$$\varphi^d = arsin\left(m\frac{U_x \sin(\varphi) - U_y \cos(\psi)}{U_1}\right) \qquad \text{with } U_1 = \sqrt{U_x^2 + U_y^2 + U_z^2}$$

Appendix B:

The EKF has been started with the following initial conditions:

$$X_0 = [0 \quad 0 \quad 0 \quad 0 \quad 0 \quad 0 \quad 1 \quad 0 \quad 1 \quad 0 \quad 1 \quad 0 \quad 0 \quad 0 \quad 0]^t$$

$$P = \begin{bmatrix} 3 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 3 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 3 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 3 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 3 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 3 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 3 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 3 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 3 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 3 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 3 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 3 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 3 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 3 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 3 \end{bmatrix}$$

$$C = \begin{bmatrix} 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 \\ 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 \end{bmatrix}$$

The system noise covariance matrix Q is 15x15, and the measurement noise covariance matrix R is 6x6,

Q and R are diagonal, and only 15 elements must be known in Q and 6 in R.

System noise covariance matrix Q:

$$Q = \begin{bmatrix} 0.5 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0.5 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0.5 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0.5 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0.5 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0.5 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0.5 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0.5 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0.5 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0.5 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0.5 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0.5 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0.5 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0.5 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0.5 \end{bmatrix}$$

Measurement noise covariance matrix R

$$R = \begin{bmatrix} 0.2 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0.2 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0.2 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0.2 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0.2 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0.2 \end{bmatrix}$$

Appendix C:

Control of quadrotor