UNIVERSITY OF THESSALY

DEPARTMENT OF MECHANICAL ENGINEERING

DIPLOMA THESIS

Trajectory tracking for an unmanned quadrotor using model predictive control

KAPNOPOULOS ARISTOTELIS

Thesis Supervisor: Dr. Alex Alexandridis, Professor

Volos, February 2019

Members of the Examination Committee:

First Examiner (Supervisor)	Dr. Alex Alexandridis Professor, Department of Electrical and Electronic Engineering, University of West Attica
Second Examiner	Dr. Nikolaos Andritsos Professor, Department of Mechanical Engineering, University of Thessaly
Third Examiner	Dr. Alexandros Kermanidis Assistant Professor, Department of Mechanical Engineering, University of Thessaly

AKNOWLEDGEMENTS

First of all I would like to thank my family and especially my mother and father for their patience and support of my choices over the years.

Secondly, I would like to express my deepest gratitude to my supervisor, Dr. Alex Alexandridis. His valuable guidance and encouragement made this Thesis a rich experience.

Thirdly, I would like to thank Dr. Nikolaos Andritsos and Dr.Alexandros Kermanidis for their participation in the examination committee.

Last but not least, I would like to thank my friends for the great time spent together and the experience shared.

Abstract

In this Thesis a model predictive control strategy is presented to solve the trajectory tracking problem for an unmanned quadcopter. The control of a quadcopter is a difficult task as its dynamical behavior, which in this work is obtained via Newton Euler formalism, exhibits nonlinear, under-actuated and strongly coupled terms, as well as multi input - multi output features. The proposed controller is divided into two sub-control schemes: the first scheme is responsible for the position, whereas the second one for the attitude control of the quadrotor. The tasks of successfully reaching a certain set point in space and tracking a reference trajectory are performed with two different kinds of tuning by the predictive controller. The simulation results validate the effectiveness of the proposed control strategy.

Keywords: Automatic control, Model predictive control, Quadcopter, Trajectory Tracking, Unmanned vehicles

ΠΕΡΙΛΗΨΗ

Σε αυτή τη διπλωματική εργασία παρουσιάζεται μία μέθοδος ελέγχου μέσω προβλεπτικού μοντέλου για την επίλυση του προβλήματος της παρακολούθησης τροχιάς στο χώρο από ένα μη επανδρωμένο τετρακόπτερο. Ο έλεγχος ενός τετρακόπτερου αποτελεί ένα δύσκολο έργο καθώς η δυναμική του συμπεριφορά, η οποία σε αυτό την εργασία περιγράφεται μέσω εξισώσεων Newton Euler, επιδεικνύει μη γραμμικούς και ισχυρά συζευγμένους όρους όπως και χαρακτηριστικά πολλαπλών εισόδων - εξόδων. Ο προτεινόμενος ελεγκτής χωρίζεται σε δύο στρατηγικές ελέγχου. Η μία είναι υπεύθυνη για τη θέση του, και η δεύτερη για τον έλεγχο της στάσης του στο χώρο. Τα προβλήματα της επιτυχούς άφιξης σε συγκεκριμένο σημείο του χώρου και της παρακολούθησης πηγαίας τροχιάς εκτελούνται με δύο διαφορετικές βαθμονομήσεις των παραμέτρων του ελεγκτή. Τα αποτελέσματα των προσομοιώσεων επιβεβαιώνουν την αποδοτικότητα της προτεινόμενης μεθοδολογίας.

Λέξεις Κλειδιά: Αυτόματος έλεγχος, Έλεγχος προβλεπτικού μοντέλου, Τετρακόπτερο, Παρακολούθησης τροχιάς, Μη επανδρωμένο όχημα

Contents

1 Introduction 1
2 Quadcopter dynamics and system 4
2.1 The concept of UAV 4
2.2Basic commands 4
2.3 Mathematical Model 8
2.3.1 Kinematics 9
2.3.2 Newton-Euler model 11
2.4 State Space Modeling 12
3 PID Control 15
3.1Theory 15
3.2 Altitude and height control 17
3.3 PID Algorithm 21
4 Model Predictive Control 25
4.1 The idea of Model Predictive Control 25
4.2 Controller Design based on E-SSPC 29
4.2.1 Position Control 29
4.2.2 Attitude Control 36
5 Case study: Control of a simulated quadcopter 40
5.1 Open loop simulations for the basic movements 41
5.1 .1 Throttle Command 41
5.1 .2 Roll Command 44
5.1 .3 Pitch Command 46
5.1.4 Yaw Command 49
5.2 Closed loop simulations using PID Controllers 51
5.2.1 Reaching a certain altitude z 53

5.2.2 Reaching a certain Euler angle φ 55
5.2.3 Reaching a certain Euler angle θ 57
5.2.4 Reaching a certain Euler angle ψ 59
5.2.5 Reaching a complex set of desired values 61
5.3 Closed loop simulations using MPC 63
5.3.1Reaching a certain point in space 63
5.3.2Task of tracking a reference trajectory in space 67
5.3.2.1Task of tracking a reference circle trajectory 68
5.3.2.2Tracking a reference spiral trajectory 72
6 Conclusions 77

Chapter 1

Introduction

In the last decades great interest has been raised around vertical take-off and landing unmanned aerial vehicles (UAV).A quadrotor helicopter is a vehicle equipped with four propellers which makes it possible to reduce the size of each rotor and to maintain or to increase the total load capacity, when compared with a helicopter with one main rotor. As a result the design and maintenance cost are reduced allowing flight in many different environments.

Some of the advantages of the quadrotor are related to its high maneuverability, its agility, its stationary flight (hovering) and its ability for vertical take-off and landing. The ample set of abilities that the quadrotor possess has led to a growing implementation in several industries (surveillance, rescue, research area, photography, etc).

The control of a quadrotor is not an easy task due to its under actuated nature and strongly coupled dynamics. From the six outputs of its dynamic system only a maximum of four can be controlled, as it has only four control inputs. A variety of control strategies have been proposed in order to deal with the problem of attitude and altitude stabilization.

The first control methods that were implemented to solve the above problem made use of linear control algorithms such as Proportional-Integral-Derivate (PID)[1-4]and Linear Quadratic Regulator (LQR)control[5].The disadvantage of such control strategies is that stability is only limited to a certain domain during flights. However, the use of nonlinear control methods has expanded the controllability and stability of quadrotors.

Nonlinear control methods were used to control the system such as linearization, saturation, integral back stepping,H_∞ control, sliding model control.Back-stepping controller generates commands to the four rotors to drive the quadrotor to track the desired values. Integral back-stepping approach was applied into the autonomous

flight of the quadrotor to solve the nonlinear control problem[5-8].Sliding mode control method alter the dynamics of the quadrotor by application of a discontinuous control signal that forces the system to slide along it's behavior[9-12].In linearization methods a nonlinear controller based on a decomposition into a nested structure and feedback linearization is implemented[13-15]. Another effective nonlinear control method dealing with the tracking a trajectory tracking is via H infinitive control[16, 17].

In this Thesis the solution of the path tracking problem was solved with the help of a different kind of control strategy, namely Model Predictive Control (MPC)[18-21].The MPC controller makes predictions by using a dynamical model. This is possible by solving an optimization problem which gilds an optimal sequence of inputs that brings the model predictions as close as it gets to the desired values. The preview capability and the fact that it can handle constraints, makes MPC a very effective control strategy in dealing with nonlinear systems.

In this work a model predictive controller is used to track a reference trajectory for a quadrotor. The main work presented in this Thesis focuses on solving the problem of tracking a desired path and attitude stabilization of the quadcopter. The system is decoupled into two subsystems, namely one dealing with the problem of path following and another one dealing with attitude stabilization. An error-state space predictive controller is used to solve the position control, while attitude stabilization is achieved via PID controllers.

In order to successfully track a reference trajectory the right tuning must be implemented to the predictive controller. This is not an easy task as it involves tuning in two different levels, one for the PID and one for the MPC parameters as bothe sets have a strong impact on the close loop performance.

The rest of this Thesis is organized as follows:

In Chapter 2 the quadrotor's dynamic model is presented. The basic movements of the quadrotor are shown and the system's dynamic behavior is explained with the help of the Newton Euler form.

In Chapter 3 the concept of the PID control algorithms is explained. The implementation of such controllers on the quadrotor model is performed in order to achieve a set of reference values.

In Chapter 4 the control strategy behind trajectory tracking is developed. Position control and attitude control structures show the way of tracking the desired set of values at each time instant.

In Chapter 5 the results of a case study on quadrotor control are presented. The efficiency and the robustness of the MPC-PID controllers is shown through simulations performed in Matlab/Simulink .

In Chapter 6, we draw conclusions and set directions for future work.

Chapter 2

Quadcopter dynamics and system

2.1 The concept of UAV

An unmanned aerial vehicle (UAV) is an aircraft without the presence of a pilot. Although the initial use of UAVs was for military purposes soon the use expanded in various sectors such as scientific, surveillance, aerial photography and product deliveries. This Thesis focuses on the analysis of rotary wings UAV .These kind of vehicles can hover, take off and land vertically and fly with high maneuverability.

A quadrotor, also called as quadcopter belongs to the family of rotary wing UAV. It uses four motors with four propellers that create the necessary thrust in order to lift the aircraft. Each rotor has a propeller fitted to an independent dc motor which converts electrical energy to mechanical. Two motors of the quad rotate clockwise and the two other counter clock wise. As a result of the upper rotation of the motors the torque that is created from each rotor is cancelled by the torque of the opposite corresponding one. This configuration of pairs rotating in opposite direction eliminates the need of a tail rotor which counterbalances the torque created by the rotation of the main rotor in the conventional helicopter.

2.2 Basic commands

The four electrical motors that are responsible for the motion of the aircraft are limiting the number of variables that can be controlled during the flight. That means that the quadrotor is an under-actuated system with 6 degrees of freedom. Thus, from the six degree of freedom (D.O.F.) it can reach a desired set point to a maximum of four. The 4 variables that are chosen to be controlled are related to the four basic movements that ensure attitude and altitude stabilization.

As it was mentioned above the control of the quadrotor is obtained by changing the angular velocities $\Omega_i (i=1,2,3,4)$ of the propellers . Each rotor creates thrust and

torque about its center of location. The proper change of the propellers speed leads to smooth movement of the vehicle in space.

In hovering condition all the propellers rotate with the same angular velocity in order to counterbalance the force due to gravity. In this state the quadrotor performs stationary flight and no forces or torques move it from its position.

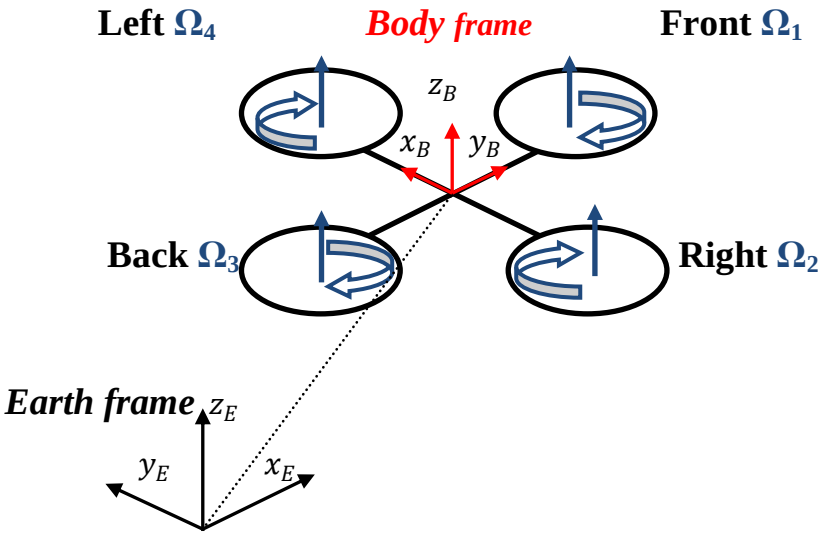


Figure 2.1:*Simplified quadrotor during hovering state*

In order to make the quadrotor fly,4 variables should be chosen to be controlled. The best four variables to control the aircraft are related to the four basic movement of the helicopter, which are:

Throttle (U_1)

This command is generated by increasing (or decreasing) all of the angular velocities $\Omega_1, \Omega_2, \Omega_3, \Omega_4,$ by the same amount. This action leads to a vertical force with respect to the body frame that raises or lower the quadcopter. In this case the speed of each propeller is equal to $\Omega_H + \Delta\omega$where Ω_H is the hovering angular speed and $\Delta\omega$is an positive variable which represent an increase in lift.$\Delta\omega$ cannot be to large so that the model won't be affected by strong non linearities.

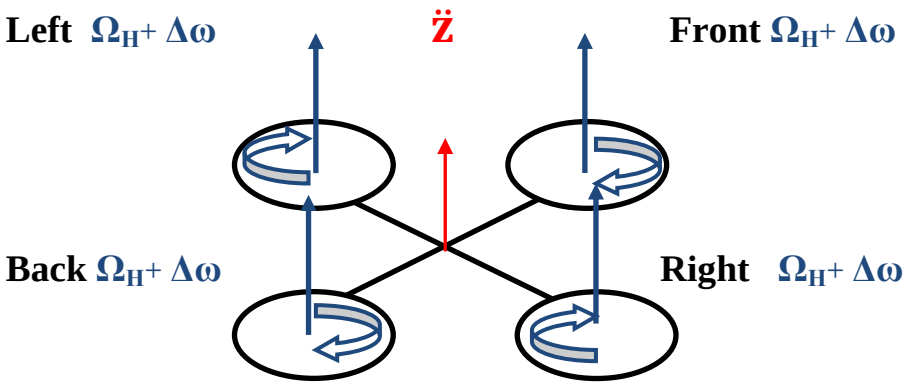


Figure 2.2:*Throttle Movement*

Roll (U_2)

This command is provided by increasing (or decreasing) the left propeller's speed and by decreasing (or increasing) the right one. This leads to a torque along the X_{BODY} axis which makes the quadrotor turn. The overall vertical thrust is the same as in the hovering, hence this command leads only to a roll angle acceleration. Likewise with Throttle movement $\Delta\omega$ cannot be too large in order to not be influenced by non linearities.

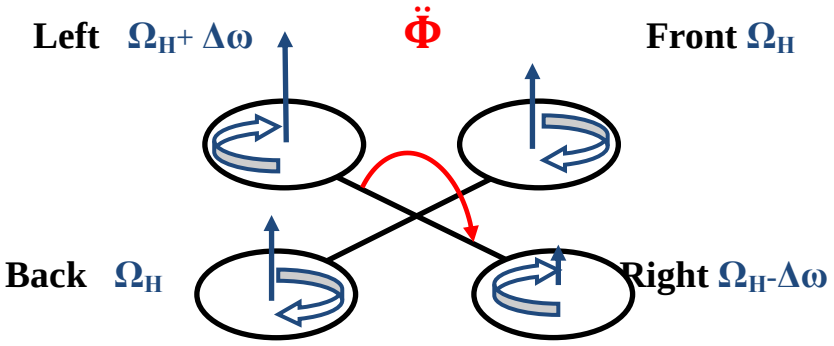


Figure 2.3:*Roll Movement*

Pitch (U_3)

This command is similar to the roll and it is provided by increasing (or decreasing) the eat propeller speed and by decreasing (or increasing) the front one. It leads to a torque along the Y_{BODY} axis which makes the quadrotor turn. The overall vertical trust is the same as in the hovering, hence this command leads only to a pitch angle acceleration. As in the previous command $\Delta\omega$ are chosen small enough in order to maintain an unchanged vertical thrust.

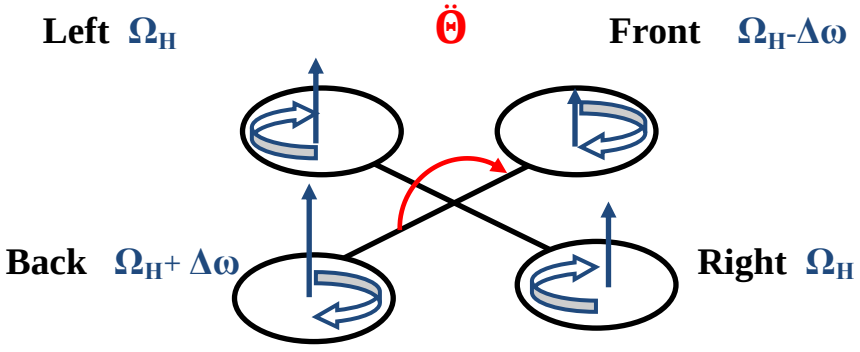


Figure 2.4:*Pitch Movement*

Yaw (U_4)

This command is provided by increasing(or decreasing)the front and back propellers' speed simultaneously and by decreasing (or decreasing) the left-right ones at the same time. This leads to a torque along the Z_{BODY} axis which makes the quadrotor turn. The yaw movement is possible due to the fact that the left-right propellers rotate counter-clockwise while the front -back couple rotate clockwise. Hence, when the overall torque is unbalanced, the helicopter spins around Z_{BODY} .The total vertical thrust is the same as in the hovering, hence this command leads only to a yaw angle acceleration. The positive variable Δω is chosen small enough so that the vertical thrust will remain unchanged as in the previous movements.

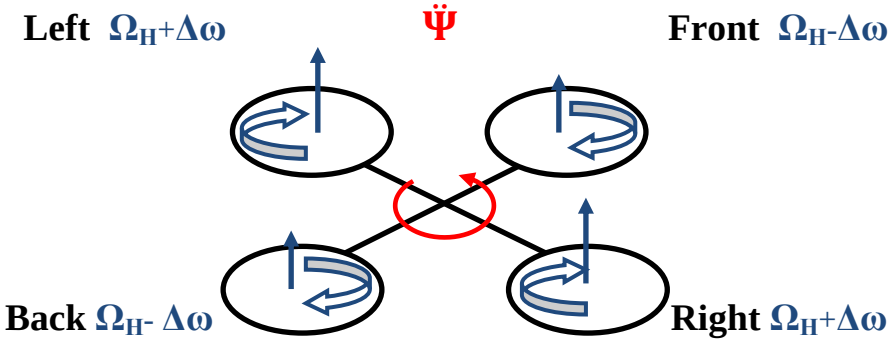


Figure 2.5:*Yaw Movement*

2.3 Mathematical Model

The mathematical model of the quadrotor describes the link between the movement and attitude with the external influences and input values. Knowing the 4 angular velocities of the propellers it is possible to predict the attitude and altitude of the quadcopter. The present model is based on the following assumptions:

- Quadrotor is a rigid body.
- Quadrotor has a symmetrical structure (the inertia matrix is diagonal).
- The center of mass and the body fixed frame origin coincide.
- The propellers are rigid.

To describe the motion of a 6 D.O.F. rigid body two reference frames are used:

- The earth inertial frame (*E* frame)
- The body fixed frame (*B* frame)

2.3.1 Kinematics

The linear position of the quadrotor is defined in the inertial frame x-y-z axes with $\boldsymbol{\xi}$. The angular position is described in the inertial frame with three Euler angles φ-θ-ψ ($\boldsymbol{\eta}$).Vector $\boldsymbol{q}$ contains the linear and angular position vectors.

$$\boldsymbol{\xi} = [\, x \;\; y \;\; z \,]^T \tag{2.1}$$

$$\boldsymbol{\eta} = [\, \varphi \;\; \theta \;\; \psi \,]^T \tag{2.2}$$

$$\boldsymbol{q} = [\; \boldsymbol{\xi} \;\; \boldsymbol{\eta} \;\;]^T \tag{2.3}$$

In the body frame the linear velocities are defined by V_B and the angular velocities by *v*.

$$\boldsymbol{V_B} = [\, u \;\; v \;\; w \,]^T \tag{2.4}$$

$$\boldsymbol{v} = [\, p \;\; q \;\; r \,]^T \tag{2.5}$$

The rotational matrix from the body frame to the inertial frame is :

$$\boldsymbol{R} = \begin{bmatrix} cos\psi\, cos\theta & cos\psi\, sin\theta\; sin\varphi - sin\varphi\, cos\varphi & cos\psi\, sin\theta\; cos\varphi + sin\psi\, sin\varphi \\ sin\psi\, cos\theta & sin\psi\, sin\theta\; sin\varphi + cos\psi\, cos\varphi & sin\psi\, sin\theta\; cos\varphi - cos\psi\, sin\varphi \\ -sin\theta & cos\theta\, sin\varphi & cos\theta\, cos\varphi \end{bmatrix} \tag{2.6}$$

With the above matrix is possible the transformation of the measured linear velocities from the one coordinate system to the other. Matrix $\boldsymbol{R}$ is orthogonal thus $R^{-1} = R^T$.

Respectively angular velocities are transformed from the inertia to the body frame with the transformation matrix $\boldsymbol{W_\eta}$.From body frame to the inertial frame the transformation matrix is $\boldsymbol{W^{-1}_\eta}$.

$$\dot{\boldsymbol{\eta}} = \boldsymbol{W}_{\boldsymbol{\eta}}^{-1}\ \boldsymbol{v} \begin{bmatrix} \dot{\varphi} \\ \dot{\theta} \\ \dot{\psi} \end{bmatrix} = \begin{bmatrix} 1 & sin\theta\ tan\theta & cos\theta\ tan\theta \\ 0 & cos\varphi & -sin\theta \\ 0 & sin\varphi/cos\theta & sin\varphi/cos\theta \end{bmatrix} \begin{bmatrix} p \\ q \\ r \end{bmatrix} \quad (2.7)$$

$$\boldsymbol{v} = \boldsymbol{W}_{\boldsymbol{\eta}}\, \dot{\boldsymbol{\eta}} \quad \begin{bmatrix} p \\ q \\ r \end{bmatrix} = \begin{bmatrix} 1 & 0 & -sin\theta \\ 0 & cos\varphi & cos\theta sin\varphi \\ 0 & -sin\varphi & cos\theta\ cos\varphi \end{bmatrix} \begin{bmatrix} \dot{\varphi} \\ \dot{\theta} \\ \dot{\psi} \end{bmatrix} \quad (2.8)$$

As it was mentioned above the quadrotor has a symmetric structure with four arms aligned with the body x and y axis. Thus the inertia matrix $\boldsymbol{I}$ is a diagonal one :

$$\boldsymbol{I} = \begin{bmatrix} I_{xx} & 0 & 0 \\ 0 & I_{yy} & 0 \\ 0 & 0 & I_{zz} \end{bmatrix} \quad (2.9)$$

Each rotor *i,* with angular velocity ω_i creates force $\boldsymbol{f_i}$ in the direction of the rotor axis .The angular velocity of the rotor also create torque $\boldsymbol{\tau_{Mi}}$**.** Where *k* is the lift constant and *b* is the drag constant.

$$\boldsymbol{f_i = b\ \Omega^2{}_i} \quad (2.10)$$

$$\boldsymbol{\tau_{Mi} = d\ \Omega^2{}_i} \quad (2.11)$$

The combination of forces $\boldsymbol{f_i}$ creates thrust $\boldsymbol{U_1}$in the direction z of the body frame. Torque $\boldsymbol{\tau_B}$ consists of the pitch torque $\boldsymbol{\tau_\theta}$ around Y_B axis, the roll torque $\boldsymbol{\tau_\varphi}$ around X_B axis, the yaw torque $\boldsymbol{\tau_\psi}$around Z_B axis.

$$\boldsymbol{U_1} = \sum_{i=1}^{4} f_i = k \sum_{i=1}^{4} \Omega_i^2 \quad (2.12)$$

$$\boldsymbol{T_B} = \begin{bmatrix} 0 \\ 0 \\ U_1 \end{bmatrix} \quad (2.13)$$

$$\boldsymbol{\tau_B} = \begin{bmatrix} \tau_\varphi \\ \tau_\theta \\ \tau_\psi \end{bmatrix} = \begin{bmatrix} l\ b\ (-\Omega_2^2 + \Omega_4^2) \\ l\ b\ (-\Omega_1^2 + \Omega_3^2) \\ d\ (+\Omega_2^2 + \Omega_4^2 - \Omega_1^2 - \Omega_3^2) \end{bmatrix} \quad (2.14)$$

2.3.2 Newton-Euler model

Using the Newton Euler equations the translational and rotational dynamics of the quadcopter are described. In the body frame, the force required for acceleration of mass and the centrifugal force are equal to the gravity and total thrust of rotors.

$$m\,\dot{\boldsymbol{V}}_{\boldsymbol{B}} + v \times (m\,\boldsymbol{V}_{\boldsymbol{B}}) = \boldsymbol{R}^{T}\boldsymbol{G} + \boldsymbol{T}_{B} \qquad (2.15)$$

In the inertial frame the centrifugal force is nullified. As a result, the acceleration of the quadrotor derives from the gravitational force the magnitude and the direction of the thrust.

$$m\,\ddot{\boldsymbol{\xi}} = \boldsymbol{G} + \boldsymbol{R}^{T}\boldsymbol{T}_{B} \qquad (2.16)$$

$$\begin{bmatrix}\ddot{\boldsymbol{x}}\\ \ddot{\boldsymbol{y}}\\ \ddot{\boldsymbol{z}}\end{bmatrix} = -g\begin{bmatrix}0\\0\\1\end{bmatrix} + \frac{U_1}{m}\begin{bmatrix}cos\psi sin\theta cos\varphi + sin\psi sin\varphi\\ sin\psi sin\theta cos\varphi - cos\psi sin\varphi\\ cos\theta cos\varphi\end{bmatrix} \qquad (2.17)$$

In the inertial frame the angular acceleration of the inertia the centrifugal forces and the gyroscopic forces are equal to the external torque.

$$I\,\dot{\boldsymbol{v}} + \boldsymbol{v} \times (\boldsymbol{I}\,\boldsymbol{v}) + \boldsymbol{\Gamma} = \boldsymbol{\tau} \qquad (2.18)$$

$$\dot{\boldsymbol{v}} = \boldsymbol{I}^{-1}\left(\begin{bmatrix}p\\q\\r\end{bmatrix} \times \begin{bmatrix}I_{xx}\ p\\ I_{yy}\ q\\ I_{zz}\ r\end{bmatrix} - I_r\begin{bmatrix}p\\q\\r\end{bmatrix} \times \begin{bmatrix}0\\0\\1\end{bmatrix}\omega_{\Gamma} + \boldsymbol{\tau}\right) \qquad (2.19)$$

$$\begin{bmatrix}\dot{\boldsymbol{p}}\\ \dot{\boldsymbol{q}}\\ \dot{\boldsymbol{r}}\end{bmatrix} = \begin{bmatrix}(I_{yy} - I_{zz})\,q\,r/I_{xx}\\ (I_{zz} - I_{xx})\,p\,r/I_{yy}\\ (I_{xx} - I_{yy})\,q\,r/I_{zz}\end{bmatrix} - I_r\begin{bmatrix}q/I_{xx}\\ -p/I_{yy}\\ 0\end{bmatrix}\omega_{\Gamma} + \begin{bmatrix}\tau_{\varphi}/I_{xx}\\ \tau_{\theta}/I_{yy}\\ \tau_{\psi}/I_{zz}\end{bmatrix} \qquad (2.20)$$

In which $\omega_{\Gamma}=\omega_1 - \omega_2 + \omega_3 - \omega_4$ .The angular accelerations move from the inertial frame to the body one with the transformational matrix $\boldsymbol{W^{-1}_{\eta}}$.

$$\ddot{\boldsymbol{\eta}} = \frac{d}{dt}\left(\boldsymbol{W}_{\boldsymbol{\eta}}^{-\mathbf{1}}\ \boldsymbol{v}\right) = \frac{d}{dt}\left(\boldsymbol{W}_{\boldsymbol{\eta}}^{-\mathbf{1}}\right)\boldsymbol{v} + \ \dot{\boldsymbol{v}} \qquad (2.21)$$

Hence the mathematical model that describes the translational dynamics for the helicopter in a system of equations is:

$$\begin{cases} \ddot{x} = \frac{1}{m}(\cos\psi \sin\theta \cos\varphi + \sin\psi sin\varphi)U_1 \\ \ddot{y} = \frac{1}{m}(sin\psi \sin\theta \cos\varphi - \cos\psi sin\varphi)U_1 \\ \ddot{z} = -g + \frac{1}{m}(cos\theta\ cos\varphi)U_1 \end{cases} \quad (2.22)$$

2.4 State Space Modeling

The mathematical model that describes the translational (2.22) and rotational (2.21) dynamics of the system will be defined by using a state space representation. This new model of the system's dynamics is a set of first-order differential equations.

A general state-space representation of a linear system can be written in the following form:

$$\dot{x}(t) = A(t)\,x(t) + B(t)u(t) \quad (2.23)$$

where x is the state vector , u is called the input vector A is the state matrix B is the input or control matrix.

The state space form is created by writing equations (2.21) and (2.22) in the form of (2.23) and results to the following equations:

$$x_1 = x$$

$$x_2 = \dot{x_1}$$

$$x_3 = y$$

$$x_4 = \dot{x_3}$$

$$x_5 = z$$

$$x_6 = \dot{x_5} \quad (2.24)$$

$$x_7 = \varphi$$

$$x_8 = \dot{x_7}$$

$$x_9 = \theta$$

$$x_{10} = \dot{x_9}$$

$$x_{11} = \psi$$

$$x_{12} = \dot{x_{11}}$$

$$\dot{x}_1 = x_2$$

$$\dot{x}_2 = \frac{1}{m}\ (\cos x_7 \sin x_9 \cos x_{11} + \sin\ x_{11} \sin\ x_7)U_1$$

$$\dot{x}_3 =\ x_4$$

$$\dot{x}_4 = \frac{1}{m}\ (sin\ x_7 \sin x_9 \cos x_{11} - \cos\ x_{11} sin\ x_7)U_1$$

$$\dot{x}_5 =\ x_6$$

$$\dot{x}_6 = -g\ +\ \frac{1}{m}\ (cos\ x_7\ cos\ x_9)U_1 \qquad (2.25)$$

$$\dot{x}_7 =\ x_8$$

$$\dot{x}_8 =\ x_{12}\ x_{10}\ \frac{I_{yy} - I_{zz}}{I_{xx}}\ -\frac{\Omega_r\ I_r}{I_{xx}}\ x_4 + \frac{l}{I_{xx}} U_2$$

$$\dot{x}_9 =\ x_{10}$$

$$\dot{x}_{10} =\ x_{12}\ x_8\ \frac{I_{zz} - I_{xx}}{I_{yy}}\ +\frac{\Omega_r\ I_r}{I_{yy}}\ x_8 + \frac{l}{I_{yy}} U_3$$

$$\dot{x}_{11} =\ x_{12}$$

$$\dot{x}_{12} =\ x_{10}\ x_8\ \frac{I_{xx} - I_{yy}}{I_{zz}}\ + \frac{l}{I_{zz}} U_4$$

$$\text{where } \begin{cases} U_1 = k\,(+\Omega_2^2 + \Omega_4^2 + \Omega_1^2 + \Omega_3^2) \\ U_2 = l\,b\,(-\Omega_2^2 + \Omega_4^2) \\ U_3 = l\,b\,(-\Omega_1^2 + \Omega_3^2) \\ U_4 = d\,(+\Omega_2^2 + \Omega_4^2 - \Omega_1^2 - \Omega_3^2) \\ \Omega_r = (+\Omega_2 + \Omega_4 - \Omega_1 - \Omega_3) \end{cases} \tag{2.26}$$

The above set of twelve first order nonlinear differential equations represents the mathematical model of the system. This system is solved with the help of Runge-Kutta numerical methods. The solution of this system in each time instant gives the position of the quadrotor in space.

Chapter 3

PID Control

3.1 Theory

PID control systems have emerged from the early 1920s. The first practical application of these controllers was for the automated steering of ships. Since then PID control techniques have been implemented widely in the manufacturing industry with success. The reasons behind the common and universal use of such controllers include its simple structure, its good performance for a variety of processes and the fact that they are tunable without a specific model of the controlled system.

The acronym PID stands for the Proportional, Integral and Derivative actions of the controller. This three-term controller is a control loop feedback mechanism that calculates an error value *e(t)* as the difference between a desired set point and a measured process variable. Based on the error, a correction is implemented based on the proportional, integral and derivative terms.

A block diagram of a PID controller in feedback loop is shown in figure 3.1 in which *r(t)* is the desired value or set point ,and *y(t)* is the measured process value.

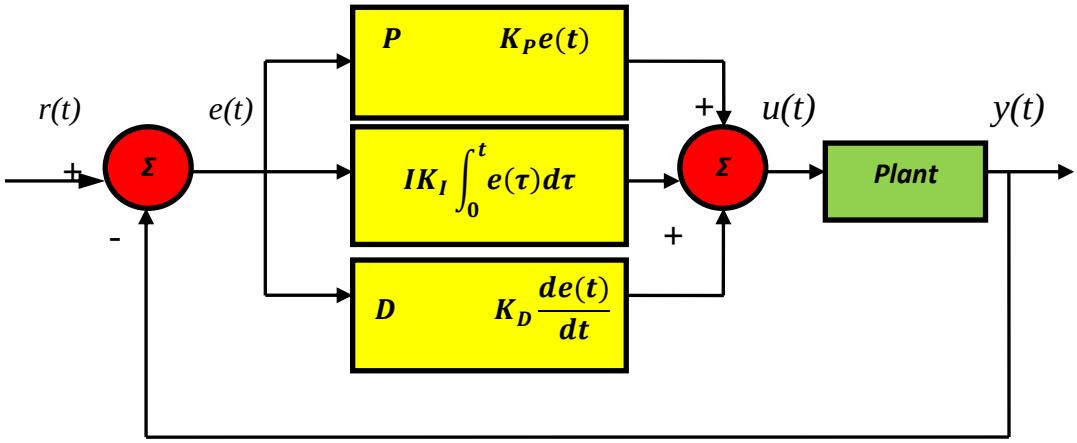


Figure 3.1:*Block diagram of PID controller*

In the above scheme *u(t)* is the manipulated variable, *e(t)* the error between the desired value *r(t)* and the plant output *y(t).*K_P ,K_I ,K_D denote the coefficients for the proportional ,integral and derivative terms, respectively.

The basic idea of a PID controller is to continuously evaluate an error over time *e(t)* as the difference between the desired value *r(t)* and the process value *y(t)* and to apply a correction based on proportional integral and derivative terms. As the time progresses the control variable *u(t)* is adjusted in a way that the error is minimized.

The three main terms of the PID model are defined below.

- ***P***

  With the help of the above term a correction is applied to the control variable which is proportional to the error .A main drawback of the proportional control is that it cannot eliminate the residual between the set point and the actual value .

- ***I***

  Term I integrates the past error values over time. After the application of the proportional control the integral control comes in order to eliminate the residual error by adding up the past values of errors. Even though this component increases the overshoot and the settling time it can eliminate the steady error completely.

- ***D***

  This term estimates the future behavior of the error based on its current rate of change. In a way derivative terms tends to reduce the effect of the produced error using a control based on the rate of error change. This part helps to decrease the overshoot, the settling time and controls the dampening effect in systems.

The right selection of the proportional integral and derivative gain constants K_P,K_I ,K_D steers the process value to the desired set point given, with the minimal overshoot, settling time and rise time. The selection of the above components can be quite challenging in order to achieve optimal behavior. Optimal behavior is possible, by reserving the requirements of regulation and command tracking. The first requirement is obtained by staying at the given set point despite external disturbance, and the second by implementing different set point values.

The control function of the PID controller can be expressed mathematically in the time domain as :

$$\boldsymbol{u(t) = K_P e(t) + K_I \int_0^t e(\tau) d\tau + K_D \frac{de(t)}{dt}} \qquad (3.1)$$

In the Laplace domain the above PID structure is written in the form**:**

$$\boldsymbol{u(s) = e(s)\left(K_P + K_I \frac{1}{s} + K_D\, s\right)} \qquad (3.2)$$

3.2 Altitude and height control

The quadrotor is an under actuated system which means that from the six D.O.F. can reach only a maximum amount of desired values even to the number of inputs. As the number of inputs is only four, the same number of values can be controlled. The selection of the controlled values are related to the four basic movements that are mentioned in the chapter 2.

Thus the four main values that are chosen to be controlled will be $\boldsymbol{z}$ height, pitch angle $\boldsymbol{\theta}$**,**roll angle $\boldsymbol{\varphi}$, yaw angle $\boldsymbol{\psi}$**.**These four values are the key to establish altitude stabilization and height stabilization for the quadrotor.

In order to control the quadrotor and maintain it in a certain position the values of the propellers rotational speed have to be found. This process is also known inverse dynamics .These kind of operation is not always possible and in many cases not

unique. In order to create an inverse model for the quadrotor some simplifications should be done.

The most important concepts of the dynamics are summarized in equations (2.20) and (2.22) in the chapter 2 .By taking into account these two equations (3.3) is created. Equation (3.3) shows the relation between the quadrotor accelerations according to the basic movements.

$$\begin{cases} \ddot{x} = \frac{1}{m}\,(\cos\psi \sin\theta \cos\varphi + \sin\psi sin\varphi)\; U_1 \\ \ddot{y} = \frac{1}{m}\,(sin\psi \sin\theta \cos\varphi - \cos\psi sin\varphi)\; U_1 \\ \ddot{z} = -g \; + \frac{1}{m}\,(cos\theta\; cos\varphi)\; U_1 \\ \dot{p} = \frac{(I_{yy} - I_{zz})}{I_{xx}}\, q\; r \; - \frac{I_r}{I_{xx}}\, q\; \omega_\Gamma + \frac{\tau_\varphi}{I_{xx}} \\ \dot{q} = \frac{(I_{zz} - I_{xx})}{I_{yy}}\, p\; r \; - \frac{I_r}{I_{yy}}\; p\; \omega_\Gamma + \frac{\tau_\theta}{I_{yy}} \\ \dot{r} = \frac{(I_{xx} - I_{yy})}{I_{zz}}\; r\; q \; + \frac{\tau_\psi}{I_{zz}} \end{cases} \qquad (3.3)$$

Another system of equations that relates basic movements with the propellers' squared speed is described via (3.4).

$$\begin{cases} U_1 = b\;\; (+\Omega_2^2 + \Omega_4^2 + \omega_1^2 + \omega_3^2) \\ U_2 = l\;\; b\; (-\Omega_2^2 + \Omega_4^2) \\ U_3 = l\;\; b\; (-\Omega_1^2 + \Omega_3^2) \\ U_4 = d\;\; (+\Omega_2^2 + \Omega_4^2 - \Omega_1^2 - \Omega_3^2) \\ \Omega_\Gamma \; = \; \Omega_1 - \Omega_2 + \Omega_3 - \Omega_4 \end{cases} \qquad (3.4)$$

The quadrotor dynamics must be simplified a lot in order to provide a model which can be implemented in the control. Equation (3.2) will be rearranged according to some considerations.

Firstly it is assumed that the motion of the quadrotor is near to the hovering condition, as result small angular changes occur especially for the roll and pitch angles.

The angular accelerations refer to its body fixed frame .They are different from the accelerations of the Euler angles which determine the attitude in the earth frame. After the hovering assumption for the quadrotor the acceleration equations have been refer directly to the Euler angle accelerations.

The number of the propellers shows the number of variables that can be controlled during the flight. Since there are four, only four can be controlled. As it has been stated the Euler angles φ,θ, $\psi$and height z will be controlled eliminating the equations which describe the x and y positions.

After the above assumptions the equation (3.5) will describe the quadrotor dynamics which will be used in control.

$$\begin{cases} \ddot{z} = -g + \dfrac{1}{m}\,(cos\theta\ cos\varphi)\ U_1 \\ \ddot{\varphi} = \dfrac{U_2}{I_{xx}} \\ \ddot{\theta} = \dfrac{U_3}{I_{yy}} \\ \ddot{\psi} = \dfrac{U_4}{I_{zz}} \end{cases} \tag{3.5}$$

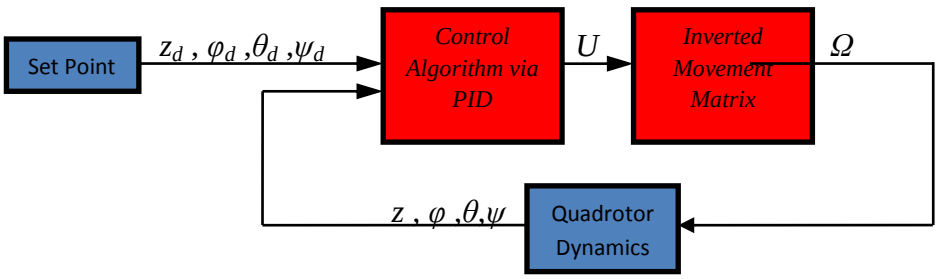


Figure 3.2:*Block diagram of attitude and height control*

The above block describes the control loop of a quadrotor that performs the task of reaching a set of four desired values. The block control algorithm receives the desired values from the task and the measured values from the block quadrotor dynamics. The output of the control continues to the inverted movement matrix which relates the four basic movements with the rotational speeds of the propellers. In particular:

- Control Algorithm via PID block:
  This block receives the values from the set point and the dynamics block and provides a signal for each basic movement *U*. Basic movements are transformed from acceleration commands in this block with the use of equation (3.5).The estimation of the accelerations command is possible with PID control.

- Inverted Movement Matrix:
  This Block computes the propellers' speed from the four basic movements. The computations needed for this process are shown in equation (3.6).

$$\begin{cases} \Omega_1^2 = \frac{1}{4\,b}U_1 - \frac{1}{2\,b\,l}U_3 - \frac{1}{4\,d}U_4 \\ \Omega_2^2 = \frac{1}{4\,b}U_1 - \frac{1}{2\,b\,l}U_2 + \frac{1}{4\,d}U_4 \\ \Omega_3^2 = \frac{1}{4\,b}U_1 + \frac{1}{2\,b\,l}U_3 - \frac{1}{4\,d}U_4 \\ \Omega_4^2 = \frac{1}{4\,b}U_1 + \frac{1}{2\,b\,l}U_2 + \frac{1}{4\,d}U_4 \end{cases} \quad (3.6)$$

3.3 PID Algorithm

In this section the control algorithm is analyzed in detail. The control algorithm block is the heart of the control loop.PID structure is used here to calculate the acceleration commands in order to estimate the basic movements using equations (3.5) .Each acceleration command is estimated individually based on the desired and measured values .Thus the number of PID blocks needed are equal to the number of acceleration commands.

Control Algorithm via PID block consists from four inner control algorithms that ensure height and attitude stabilization. The structure of these control algorithms is shown below:

- Roll Control φangle
-

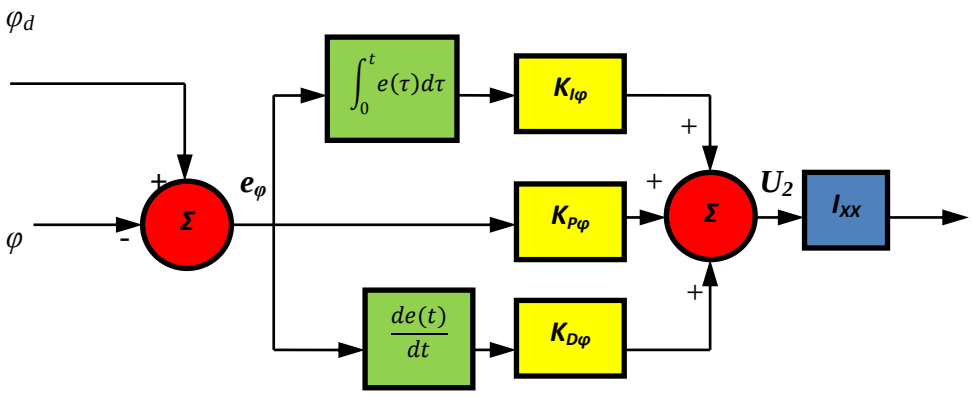


Figure 3.3:*Block diagram of Roll control*

In the above figure the sequence of action in order to produce the roll command $U_2$is shown.φ_d(rad) represents the desired roll angle value given by the user and φ(rad) the measured roll angle .e_φ(rad) is the roll error and U_2 describes the roll torque command (N m).$K_{P\varphi}$,$K_{I\varphi}$,$K_{D\varphi}$ (s^{-2}) are the three control gain parameters. Finally I_{xx}(N m)is the moment of inertia around x axis. The contribution of $I_{xx}$is mandatory to the roll command U_2 based on the equation (3.5).

- Pitch Control for θ angle

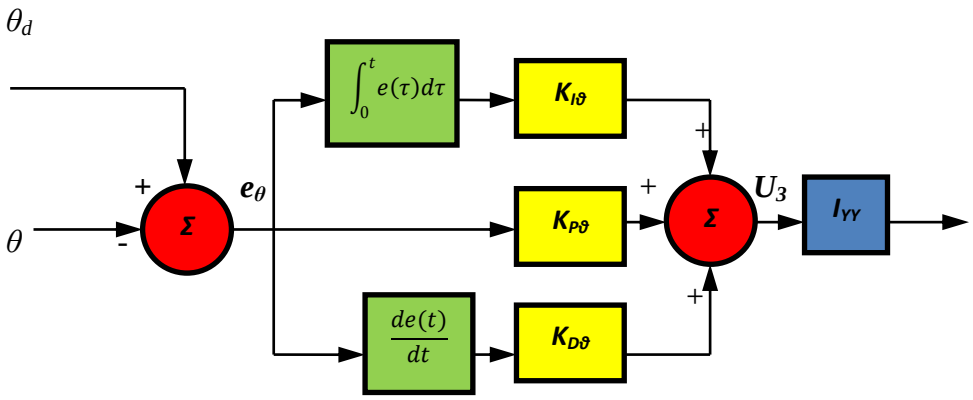


Figure 3.4: *Block diagram of Pitch control*

In figure (3.4) the sequence of action in order to produce the pitch command U_3it is shown.θ_d(rad) represents the desired roll angle value given by the user and θ(rad) the measured roll angle .e_θ(rad) is the roll error and U_3describes the pitch torque command (N m).$K_{P\theta}$,$K_{I\theta}$,$K_{D\theta}$ (s^{-2}) are the three control gain parameters. Finally I_{yy}(N m)is the moment of inertia around y axis. The only difference between roll and pitch commands it that the pitch one acts around y axis .The contribution of $I_{yy}$is mandatory to produce the pitch command U_3 based on the equation (3.5).

- Yaw Control for ψangle

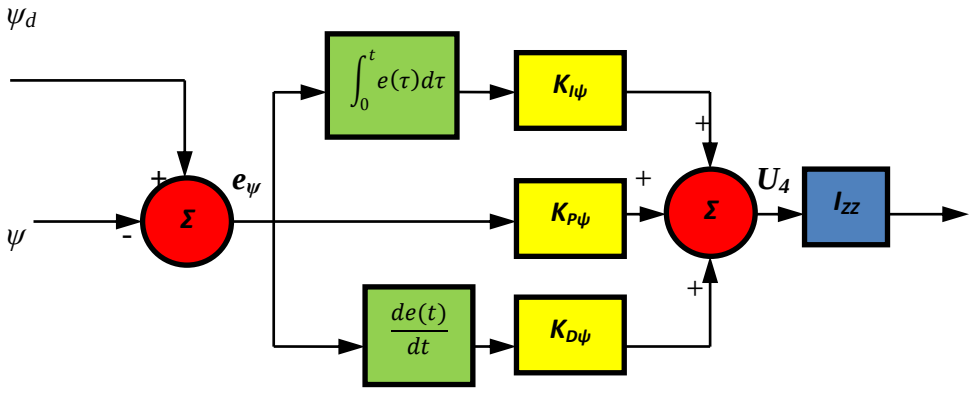


Figure 3.5:*Block diagram of Yaw control*

Figure (3.5) shows the sequence of action in order to produce the yaw command U_4.ψ_d(rad) represents the desired yaw angle value given by the user and ψ(rad) the measured yaw angle .e_ψ(rad) is the yaw error and U_4describes the yaw torque command (N m). $K_{P\psi}$, $K_{I\psi}$, $K_{D\psi}$ (s^{-2}) are the three control gain parameters. Finally I_{zz} (N m)is the moment of inertia around z axis. The contribution of I_{zz} is mandatory to produce the yaw command U_4based on the equation (3.5).

- Height Control for *z* altitude

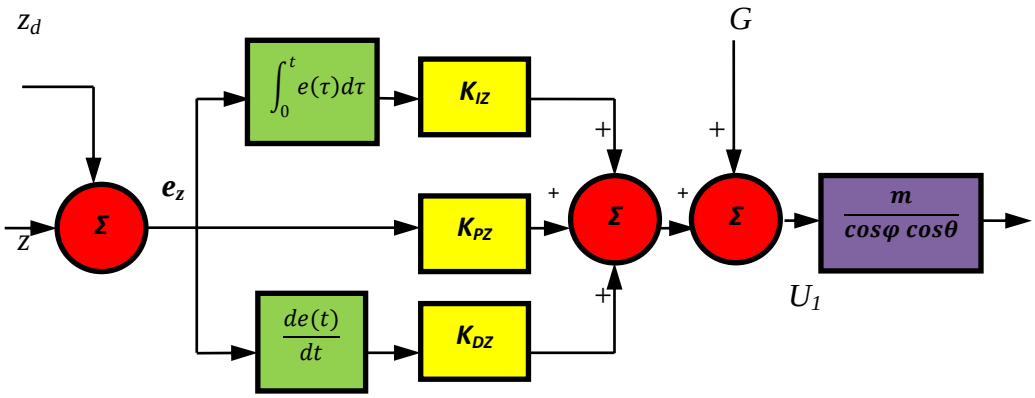


Figure 3.6: *Block diagram of Height control*

Finally, figure (3.6) shows the sequence of action in order to produce the altitude command U_1. z_d (m) represents the desired height value given by the user and z(m) the measured height value. e_Z (m) is the height error and U_1 describes the thrust command (N). K_{PZ}, K_{IZ}, K_{DZ} (s^{-2}) are the three control gain parameters. G (m s^{-2}) is the acceleration due to gravity and m (kg) the mass of the quadrotor. Part $\frac{m}{cos\varphi\ cos\theta}$completes the transformation of the altitude acceleration to the thrust command U_1based on the equation (3.5).

The above 4 control algorithms are necessary in order to reach a desired set point ensuring altitude and attitude stabilization.PID is a simple and effective controller which deals sufficiently with the nonlinear dynamics of our model. The four above control algorithms deal with reaching a constant set point which does not change over time. In the next chapter a new controller will be introduced for the purpose of controlling the position of the quadrotor. This new controller called MPC provides the capability of tracking a reference trajectory. The position control is achieved by controlling not only altitude z but also the x, y directions who change over time. In this thesis the PID algorithms that will be used deal only with attitude control, controlling the task of reaching the desired values for the Euler Angles φ, θ, ψ.

Chapter 4

Model Predictive Control

In this chapter a technique for automatic control which makes predictions about future outputs of a process will be introduced. This controller named MPC (Model Predictive Controller) solves an on line optimization problem on each time step using the plants' model. The implementation of this control algorithm settles with the difficult task of non linear and time variant system control with success.

4.1 The idea of Model Predictive Control

The use of predictive control has appeared in automatic control systems since the early 1980s. Since then, these methodologies have been widely implemented in the process industries as well as for academic purposes. The unique ability of making a prediction on the future outputs and optimizing the current timeslot is the reason of its wide utilization in industries.

The basic concept behind MPC control is to make predictions for the future outputs of a dynamic model using the current measurements and the model. These predictions are made by making the appropriate changes on the input variables.

The main advantages of the MPC controllers follow:

- MPC can handle Multi-input Multi-output systems.
- MPC can handle constraints.
- They can be used in many different kind of processes, linear and nonlinear.
- They are easy to be implemented by the working payroll.

As it is shown, the implementation of MPC controllers offers a wide variety of advantages to the industrial and academic use. However these methodologies also present some drawbacks. Firstly, the augmented complexity of an MPC controller creates difficulties in solving the optimization problem. Another serious drawback has to do with the extraction of the system's model which in a lot of cases is a very

difficult task. Last, in order to use these controllers for industrial purposes a PC-based control system is needed.

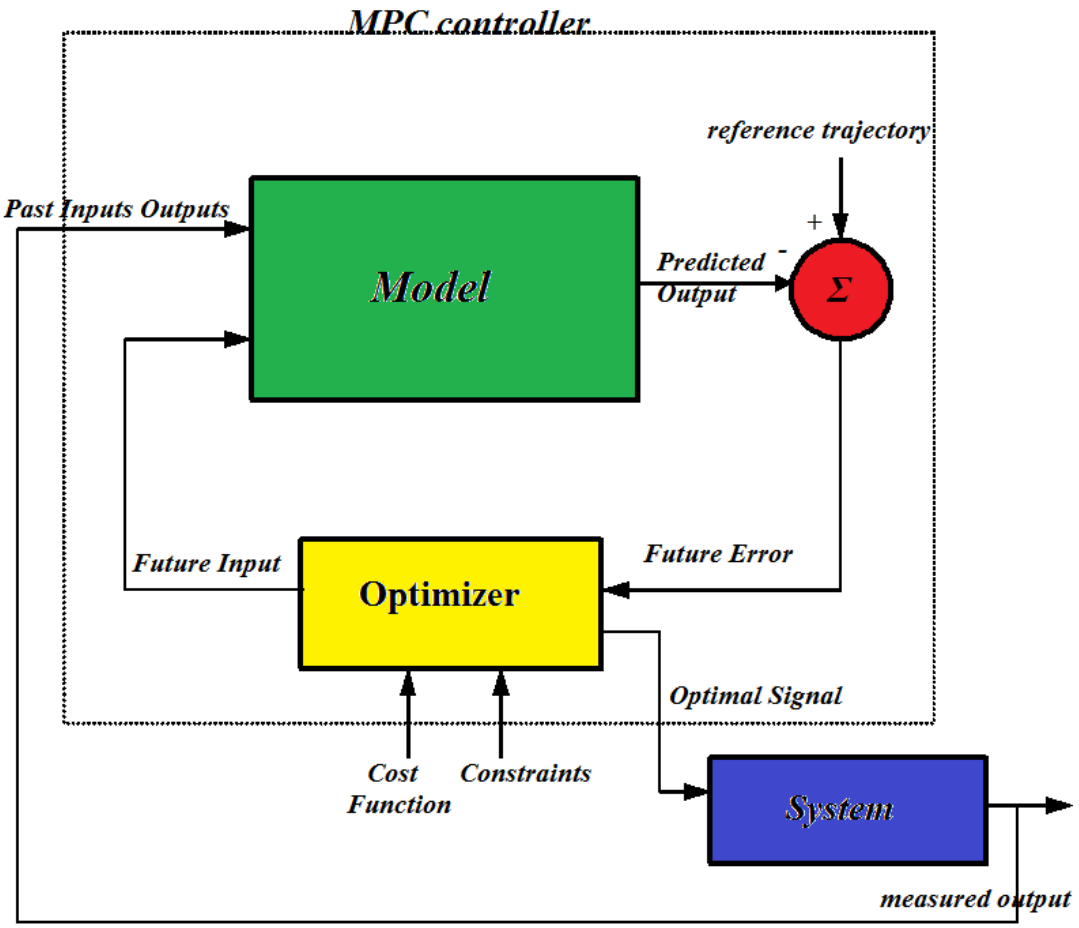


Figure 4.1: *Block diagram of MPC controller*

Figure 4.1 shows the basic structure of an MPC system.

A detailed sequence of moves of an MPC controller will be referred in order to analyze the way that this type of controller reaches a desired given task.

At each time step k the predictions about the future plant outputs are calculated based on a model of the system for a predetermined horizon h_p. The prediction horizon h_p is the number of the predicted future time steps and shows how far the controller predicts in to the future. The predicted outputs $\hat{y}(k+i)$ where $i=1,..,h_p$ are based on the past input outputs of the system and the future control moves $u(k+i-1)$,$i=1,2...,h_c-1$.

The number of control moves to until the future time step h_c-1 is called control horizon and is symbolized by h_c. Inside the control horizon, the control moves change, while outside they remain constant. The control horizon moves u are estimated by optimizing a cost function J. The cost function that is often selected is a summation of quadratic errors in order to minimize the difference between the predicted future outputs and the desired given set points. The control horizon is chosen always smaller than the prediction horizon .Usually only the first couple of control moves have a significant effect on the predicted output behavior.

After the estimation of control moves $u(k+i-1)$ for $i=1,...,h_c-1$ at time instant, only the first one $u(k)$ is implemented in the system and the remaining ones are rejected. After applying $u(k)$ the real desired output $y(k)$ is calculated. Now the prediction and the control horizon shifts forward by one time step and the MPC controller repeats the same cycle of calculations to compute the optimal u for the next time step $k+1$.

Because of the forward moving nature of the prediction horizon, MPC is also referred to as receding horizon control

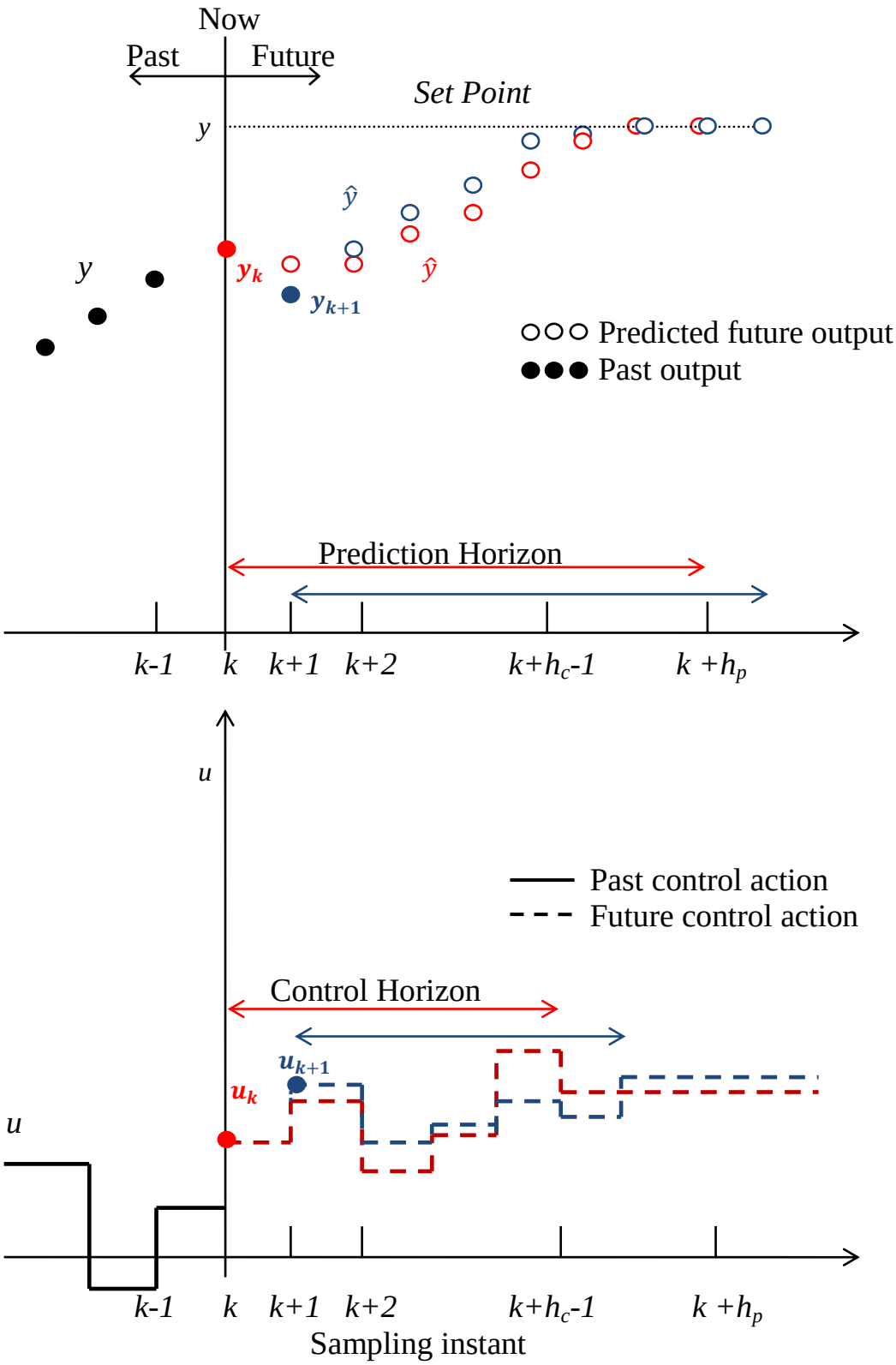


Figure 4.2: *MPC methodology for a SISO system*

The previous figure shows the sequence of moves followed by a single input single output (SISO) system to reach a certain set point. At time instant $\boldsymbol{k}$ the system is at position $\boldsymbol{y_k}$(recent output). Taking into account the past inputs-outputs and the future inputs the MPC controller produces the predicted future outputs $\hat{\boldsymbol{y}}$for a prediction horizon $\boldsymbol{h_p}$**.** In order to achieve these predicted future outputs a set of future control action $\boldsymbol{u_{k+1}}$are estimated for a control horizon $\boldsymbol{h_c}$**.** From the set of future control actions only the first is implemented to the real system and the rest are rejected. After applying control action $\boldsymbol{u_k}$the system move to a new position $\boldsymbol{y_{k+1}}$for time ***k+1.*** The prediction horizon together with the control horizon shift forward to time step ***k+1*** and the MPC controller calculates a new set of future control action $\boldsymbol{u_{k+1}}$ **.**

In the next section, an MPC algorithm named Error State Space Predictive Control (E-SSPC) will be introduced in order to complete the task of tracking a reference trajectory.

4.2 Controller Design based on E-SSPC

4.2.1 Position Control

A augmented vector $\bar{\xi}(t) = [x(t)\ u(t)\ y(t)\ v(t)\ z(t)\ w(t)]^T$defines the state of the system, where *u(t),v(t),w(t),* are the linear velocities of the quad rotor's body frame. The position subsystem in equation (2.22) can be rewritten as

$$\dot{\bar{\xi}}(t) = \begin{bmatrix} u(t) \\ u_x(t)\frac{U_1(t)}{m} \\ v(t) \\ u_y(t)\frac{U_1(t)}{m} \\ w(t) \\ -g + cos\theta(t)cos\varphi(t)\frac{U_1(t)}{m} \end{bmatrix} \quad (4.1)$$

Where

$$u_x(t) = \cos\psi(t)\sin\theta(t)\cos\varphi(t) + \sin\psi(t)sin\varphi(t) \quad (4.2)$$

$$u_y(t) = \sin\psi(t)\sin\theta(t)\cos\varphi(t) - \cos\psi(t)sin\varphi(t)$$

In the above subsystem it is assumed that there are no external disturbances.

System (4.1) can be decoupled into two subsystems which control altitude *z* and motion in *x*-*y* plane, respectively.

For the **altitude *z*** subsystem the model is given by :

$$\dot{\vec{\xi}}_z(t) = \begin{bmatrix} w(t) \\ -g + cos\theta(t)cos\varphi(t)\frac{U_1(t)}{m} \end{bmatrix} (4.3)$$

where $\vec{\xi}(t) = [z(t)\ \ w(t)]^T$

The reference trajectory is provided to the controller off-line. The trajectory is time-varying, thus a virtual reference model with same dynamics as the quadrotor (virtual quadrotor) is created. It is assumed that there is no external disturbance to the virtual reference which results to the following reference for the quadrotor dynamics:

$$\dot{\vec{\xi}}_{r_z}(t) = \begin{bmatrix} w_r(t) \\ -g + cos\theta(t)cos\varphi(t)\frac{U_{1_r}(t)}{m} \end{bmatrix} (4.4)$$

with the augmented vector given by $\bar{\xi}_{r_z}(t) = [z_r(t)\ \ w_r(t)]^T$ which represents the reference state.

The reference control input can be obtained:

$$U_{1_r}(t) = \frac{m\,(\ddot{z}_r(t) + g)}{cos\theta(t)cos\varphi(t)} \qquad (4.5)$$

By subtracting the virtual system (4.4) from the position system (4.3) a position error model can be described as:

$$\dot{\tilde{\xi}}_z(t) = \begin{bmatrix} w(t) - w_r(t) \\ cos\theta(t)cos\varphi(t)\frac{(U_1(t)-U_{1_r}(t))}{m} \end{bmatrix} \qquad (4.6)$$

where $\tilde{\xi}_z(t) = \bar{\xi}_z(t) - \bar{\xi}_{r_z}(t)$ is the position error vector ,$\bar{w}(t) = w(t) - w_r(t)$ is the velocity error in direction z ,$\bar{u}_z(t) = U_1(t) - U_{1_r}(t)$ is the control input error.

$$\dot{\tilde{\xi}}_z(t) = \begin{bmatrix} \bar{w}(t) \\ cos\theta(t)cos\varphi(t)\frac{\bar{u}_z(t)}{m} \end{bmatrix} \qquad (4.7)$$

Now the state error vector is defined:

$$x_z(t) = \tilde{\xi}_z(t) = \begin{bmatrix} \bar{z}(t) \\ w(t) \end{bmatrix} = \begin{bmatrix} z(t) - z_r(t) \\ w(t) - w_r(t) \end{bmatrix} \qquad (4.8)$$

Using Euler's method system (4.7) can be rewritten for the time instant *k+1* based on the initial current time *k*. Δt is the sampling time where $\Delta t = t_{k+1} - t_k$.

$$\begin{bmatrix} \tilde{z}(k+1) \\ \tilde{w}(k+1) \end{bmatrix} = \begin{bmatrix} z(k) - z_r(k) \\ w(k) - w_r(k) \end{bmatrix} + dt \begin{bmatrix} w(k) - w_r(k) \\ cos\theta(k)cos\varphi(t)\frac{\bar{u}_z(k)}{m} \end{bmatrix}$$

$$x_z(k+1) = \begin{bmatrix} 1 & dt \\ 0 & 1 \end{bmatrix} x_z(k) + \begin{bmatrix} 0 \\ \frac{dt}{m} cos\theta(k)cos\varphi(k) \end{bmatrix} \bar{u}_z(k)$$

$$x_z(k+1) = A_z\, x_z(k) + B_z(k)\tilde{u}_z(k) \qquad (4.9)$$

where matrices A_z and $B_z(k)$ are in the following form :

$$A_z = \begin{bmatrix} 1 & dt \\ 0 & 1 \end{bmatrix}, B_z(k) = \begin{bmatrix} 0 \\ \frac{dt}{m} cos\theta(k)cos\varphi(k) \end{bmatrix} (4.10)$$

Euler angles θ,φ ,$\psi$are regarded as time varying parameters .

The control law is designed to minimize the cost function J_Z and is defined by:

$$J_z = \left[\hat{x}_z - \hat{x}_{r_z}\right]^T Q_z \left[\hat{x}_z - \hat{x}_{r_z}\right] + \left[\hat{\hat{u}}_z - \hat{\hat{u}}_{r_z}\right]^T R_z \left[\hat{\hat{u}}_z - \hat{\hat{u}}_{r_z}\right] + \Omega\left[\hat{x}_z(k+h_p \ / k) - \right.$$
$$\left._{xrz(k+hp\ /k)}\right. \quad (4.11)$$

whereQ_z and R_z are diagonal positive weight matrices h_p is the prediction horizon, h_c is the control horizon. $\hat{x}_z$and$\hat{\hat{u}}_z$are in the following form:

$$\hat{x}_z = \begin{bmatrix} \hat{x}_z(k+1\ /k) \\ . \\ . \\ . \\ \hat{x}_z(k+h_p\ /k) \end{bmatrix}, \hat{\hat{u}}_z = \begin{bmatrix} \hat{\hat{u}}_z(k\ /k) \\ . \\ . \\ . \\ \hat{\hat{u}}_z(k+h_c-1\ /k) \end{bmatrix} \quad (4.12)$$

where$\hat{x}_z(k+j\ /k)$is the prediction of the plant output for the time instant k+j at time k .

The reference vectors are:

$$\hat{x}_{r_z} = \begin{bmatrix} x_{r_z}(k+1\ /\ k) - x_{r_z}(k/\ k) \\ . \\ . \\ . \\ x_{r_z}(k+h_p\ /k) - x_{r_z}(k/\ k) \end{bmatrix} \quad (4.13)$$

$$\hat{\hat{u}}_{r_z} = \begin{bmatrix} U_{1_r}(k\ /\ k) - U_{1_r}(k-1/\ k) \\ . \\ . \\ . \\ U_{1_r}(k+h_c-1\ /\ k) - U_{1_r}(k-1/\ k) \end{bmatrix} \quad (4.14)$$

the terminal state cost is defined as :: $\Omega\left[\hat{x}_z(k+h_p \ / k) - \hat{x}_{r_z}(k+h_p \ / k)\right]$

$$\Omega\left[\hat{x}_z(k+h_p \ / k) - \hat{x}_{r_z}(k+h_p \ / k)\right] =$$

$$\left[\hat{x}_z(k+h_p \ / k) - \hat{x}_{r_z}(k+h_p \ / k)\right] G_z \left[\hat{x}_z(k+h_p \ / k) - \hat{x}_{r_z}(k+h_p \ / k)\right] G_z \quad (4.15)$$

where G_Z is a diagonal positive weight matrix of terminal states.

The prediction of the plant output is computed using the equations (4.9) and we obtain

$$\hat{x}_z(k+1) = P_z(k/k)\, x_z(k) + H_z(k/k)\vec{u}_z(k) \qquad (4.9)$$

where matrices H_z and P_z are written as follows :

$$P_z\left(\frac{k}{k}\right) = \begin{bmatrix} A_z \\ A^2{}_z \\ . \\ . \\ . \\ A^{h_p}{}_z \end{bmatrix} \qquad (4.10)$$

$$H_z(k/k) = \begin{bmatrix} B_z(k/k) & 0 & \dots & 0 \\ A_z\, B_z(k/k) & B_z(k+1/k) & \dots & 0 \\ \vdots & \vdots & \ddots & \vdots \\ A_z{}^{h_p-2} B_z(k/k) & A_z{}^{h_p-3} B_z(k+1/k) & \dots & 0 \\ A_z{}^{h_p-1} B_z(k/k) & A_z{}^{h_p-2} B_z(k+1/k) & \dots & B_z(k+h_c-1/k) \end{bmatrix} \qquad (4.11)$$

For simplicity a vector F_Z:

$$F_z = [A_z{}^{h_p-1} B_z(k/k) \quad A_z{}^{h_p-2} B_z(k+1/k) \quad \dots \quad B_z(k+h_c-1/k)] \qquad (4.12)$$

where $\hat{x}_z\left(k+h_p/k\right) = A_z{}^{h_p} x_z + F_z \hat{\vec{u}}_z$

Minimizing the cost function J_Z the control law is constructed as :

$$\hat{\vec{u}}_{z_0} = \left[H_z{}^T Q_z H_z + R_z + F_z{}^T G_z F_z\right]^{-1} *$$

$$\left[H_z{}^T Q_z(\hat{x}_{r_z}(k) - P_z x_z(k)) + R_z \hat{\vec{u}}_{r_z} + F_z{}^T G_z(\hat{x}_{r_z}(k+h_p/k) - A_z{}^{h_p} x_z(k))\right] \qquad (4.13)$$

At each time instant *k* only $\hat{\vec{u}}_z(k/k)$needs to be estimated. Thus the control input for the altitude *z* at time *k* is:

$$U_1(k) = \hat{\vec{u}}_z(k/k) + U_{1_r}(k) \qquad (4.14)$$

The **motion in *x-y*** plane can be designed on the basis of E-SSPC method. The dynamic model is given by :

$$\dot{\bar{\xi}}_{xy}(t) = \begin{bmatrix} u(t) \\ u_x(t)\frac{U_1(t)}{m} \\ v(t) \\ u_y(t)\frac{U_1(t)}{m} \end{bmatrix} \qquad (4.15)$$

where$\bar{\xi}_{xy}(t) = \lceil x(t) \quad u(t) \quad y(t) \quad v(t) \rceil$ and the reference control inputs are:

$$u_{x_r}(t) = \frac{\ddot{x}_r(t)\,m}{U_1(t)}\,, u_{y_r}(t) = \frac{\ddot{y}_r(t)\,m}{U_1(t)} \qquad (4.16)$$

The position error can be described as :

$$\dot{\tilde{\xi}}_{xy}(t) = \begin{bmatrix} \tilde{u}(t) \\ \tilde{u}_x(t)\frac{U_1(t)}{m} \\ \tilde{v}(t) \\ \tilde{u}_y(t)\frac{U_1(t)}{m} \end{bmatrix} \qquad (4.17)$$

where$\tilde{\bar{\xi}}_{xy}(t) = \bar{\xi}_{xy}(t) - \bar{\xi}_{xy_r}(t)$ is the position error vector ,$\tilde{u}(t)$ and $\tilde{v}(t)$ stands for velocity errors ,$\tilde{u}_x(t)$ and $\tilde{u}_y(t)$are the control input errors.

Define $x_{xy}(t) = \tilde{\bar{\xi}}_{xy}(t)$ the nominal system is :

$$\dot{x}_{xy}(t) = \begin{bmatrix} \tilde{u}(t) \\ \tilde{u}_x(t)\frac{U_1(t)}{m} \\ \tilde{v}(t) \\ \tilde{u}_y(t)\frac{U_1(t)}{m} \end{bmatrix} \qquad (4.18)$$

The system can be discretized using Euler's method into the following form:

$$x_{xy}(k+1) = A_{xy}\, x_{xy}(k) + B_{xy}(k)\tilde{u}_{xy}(k) \qquad (4.19)$$

where matrices A_{xy} and $B_{xy}(k)$are in the following form:

$$A_{xy} = \begin{bmatrix} 1 & \Delta t & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & \Delta t \\ 0 & 0 & 0 & 1 \end{bmatrix}, \quad B_{xy}(k) = \begin{bmatrix} 0 & 0 \\ \frac{\Delta t}{m} U_1(k) & 0 \\ 0 & \frac{\Delta t}{m} U_1(k) \\ 0 & 0 \end{bmatrix} \qquad (4.20)$$

and $\hat{\tilde{u}}_{xy}(k) = [\tilde{u}_x(k) \quad \tilde{u}_y(k)]^T$

Following the same procedure with the control law for the altitude subsystem by minimizing the cost function J_{XY} the control law is constructed:

$$\hat{\tilde{u}}_{xy_0} = \left[{H_{xy}}^T Q_{xy} H_{xy} + R_{xy} + {F_{xy}}^T G_{xy} F_{xy}\right]^{-1} *$$

$$\left[{H_{xy}}^T Q_{xy} (\hat{x}_{r_{xy}}(k) - P_{xy} x_{xy}(k)) + R_{xy} \hat{\tilde{u}}_{r_{xy}} + {F_{xy}}^T G_{xy} (\hat{x}_{r_{xy}}(k + h_p/k) - Azhpxxy_{(k))} \right. \quad (4.21)$$

whereQ_{xy}, R_{xy} and G_{xy} are diagonal positive weight matrices. The control input is $\hat{\tilde{u}}_{xy_0} = \hat{\tilde{u}}_{xy}$. the control inputs at time k can be written as:

$$\begin{bmatrix} u_x(k) \\ u_y(k) \end{bmatrix} = \begin{bmatrix} \tilde{u}_x(k/k) \\ \tilde{u}_y(k/k) \end{bmatrix} + \begin{bmatrix} u_{x_r}(k) \\ u_{y_r}(k) \end{bmatrix} \qquad (4.22)$$

From equation (4.2) it is obtained that:

$$u_x(k) = \cos\psi(k) \sin\theta_r(k) \cos\varphi_r(k) + \sin\psi(k) sin\varphi_r(k) \qquad (4.23)$$

$$u_y(k) = \sin\psi(k) \sin\theta_r(k) \cos\varphi_r(k) - \cos\psi(k) sin\varphi_r(k)$$

Combining equations (4.21),(4.22),(4.23) the reference angles φ_r , θ_r can be computed which is necessary for the inner loop. Reference Euler's angles φ_r , θ_r are time varying angles which change at each time *k* according to input control.

In order to calculate these two angles we set $\boldsymbol{\psi_r = 0}$ in order to solve a 2 x 2 system.

$$\varphi_r = \sin^{-1}(\, u_x(k) \sin\psi_r - \, u_y(k) \cos\psi_r) (4.24)$$

$$\theta_r = \sin^{-1}(\frac{u_x(k) \cos\psi_r + u_y(k) \sin\psi_r}{\cos\varphi_r}) (4.25)$$

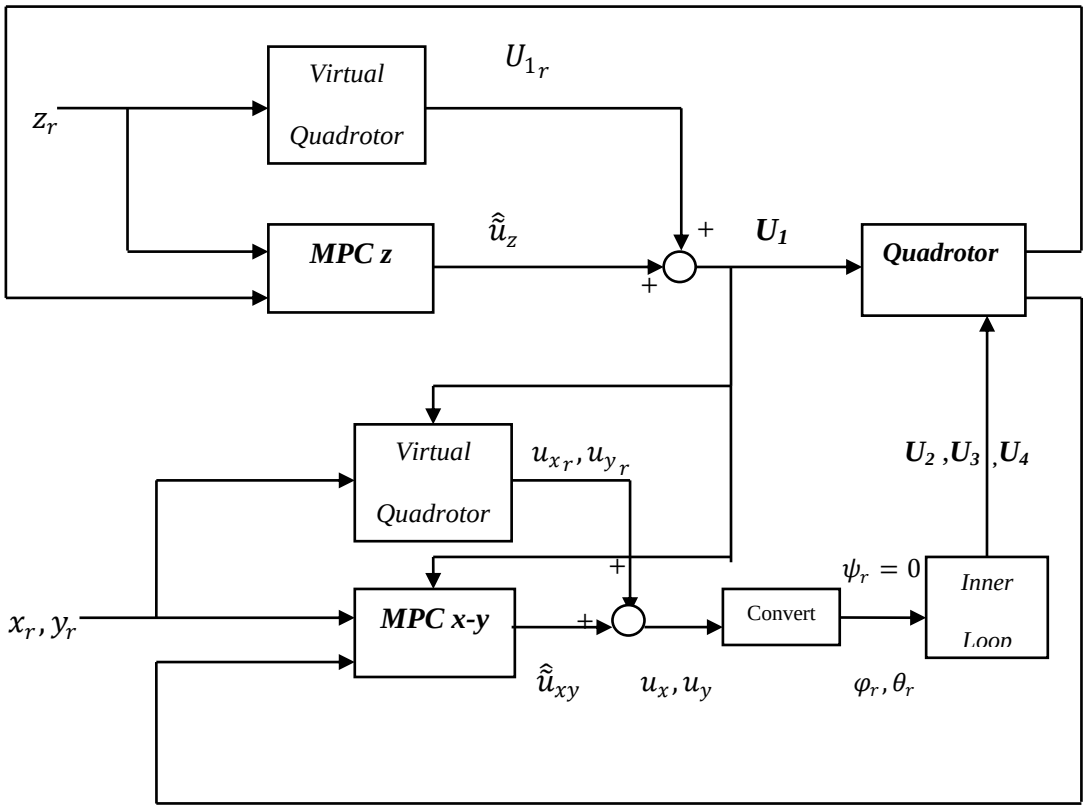


Figure 4.3: *MPC diagram for position control*

4.2.2 Attitude Control

The attitude control is the key in order to control the quadrotor with success. In order to develop the attitude controller the rotational dynamics are considered. The system of equations are expressed based on (3.3):

$$\begin{aligned}
\dot{p} &= \frac{\left(I_{yy} - I_{zz}\right)}{I_{xx}} q\, r \; - \frac{I_r}{I_{xx}} q\, \omega_\Gamma + \frac{\tau_\varphi}{I_{xx}} \\
\dot{q} &= \frac{(I_{zz} - I_{xx})}{I_{yy}} p\, r \; - \frac{I_r}{I_{yy}}\, p\, \omega_\Gamma + \frac{\tau_\theta}{I_{yy}} \\
\dot{r} &= \frac{\left(I_{xx} - I_{yy}\right)}{I_{zz}}\, r\, q \; + \frac{\tau_\psi}{I_{zz}}
\end{aligned} \qquad (4.26)$$

Under the assumptions that were mentioned at chapter 3 the above system of equations leads to the following form:

$$\begin{aligned}
\ddot{\varphi} &= \frac{U_2}{I_{xx}} \\
\ddot{\theta} &= \frac{U_3}{I_{yy}} \\
\ddot{\psi} &= \frac{U_4}{I_{zz}}
\end{aligned} \qquad (4.27)$$

The objective of the attitude controller is to ensure that the attitude of the quadrotor described by its Euler's angles φ,θ ,ψtracks the desired trajectory values φ_r,θ_r ,ψ_r asymptotically. In order to achieve the above goal the control inputs U_2 ,U_3 ,U_4 should be estimated in such a way that Euler's angles will follow the desired trajectory attitude angles which are derived from the position controller by relations (4.24) and (4.25).

PID controllers provide a simple and thus agood response for controlling the attitude of the quadrotor ,and to design the control inputsU_2 ,U_3 ,U_4 .

Three independent channels are considered one for each of Euler's angle. As a result 3 PID blocks will be created one for each Euler angle, respectively.

PID controllers:

$$\begin{cases}
u_\theta(t) = K_{P\theta}(\theta_r(t) - \theta(t)) + K_{D\theta}(\dot{\theta}_r(t) - \dot{\theta(t)}) \\
u_\varphi(t) = K_{P\varphi}(\varphi_r(t) - \varphi(t)) + K_{D\varphi}(\dot{\varphi}_r(t) - \dot{\varphi(t)}) \\
u_\psi(t) = K_{P\psi}(\psi_r(t) - \psi(t)) + K_{D\psi}(\dot{\psi}_r(t) - \dot{\psi(t)})
\end{cases} (4.28)$$

Where $K_{P\theta}$ $K_{D\theta}$, $K_{P\varphi}$, $K_{D\varphi}$, $K_{P\psi}$, $K_{D\psi}$ are positive gain values .

By estimating the virtual control inputs $u_\theta(t), u_\varphi(t), u_\psi(t)$ with the help of PID controllers the Roll command U_2 ,Pitch command U_3 and Yaw command U_4can be estimated using (4.28) and (4.27).

$$\begin{aligned} U_2 &= I_{xx}\, u_\varphi \\ U_3 &= I_{yy}\, u_\theta \\ U_4 &= I_{zz}\, u_\psi \end{aligned} \quad (4.29)$$

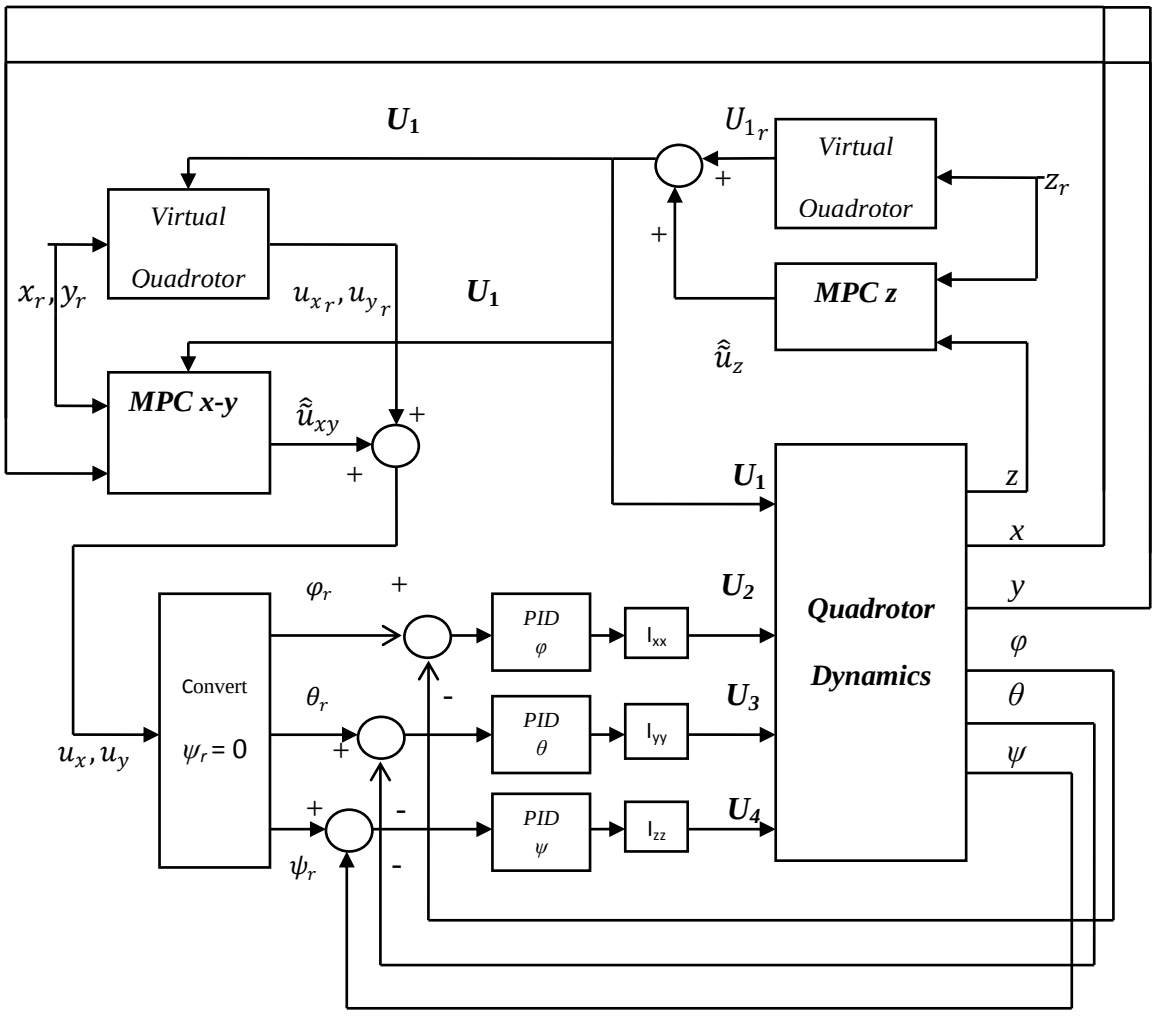


Figure 4.4: *Control Structure of Quadrotor*

Figure 4.4 depicts the overall strategy in order to control the position and attitude of the quadrotor with the MPC and PID controllers which were described in chapters 3 and 4.

Chapter 5

Case study: Control of a simulated quadcopter

This chapter aims in demonstrating the simulation results of the quadrotor flight. The proposed control scheme was fully programmed in Matlab and the simulations where performed in Matlab/Simulink using the solver ode45, which solves a system of ODEs at each time instant to calculate the quadrotor's position and attitude.

In order to perform the simulations the values shown in Table 5.1were assigned to the quadrotor parameters.

Symbol	Description	Value	Unit
l	Arm Length	0.24	m
m	Mass of the quadrotor	1	kg
I_x	Body Moment of Inertia around x axis	$8 \cdot 10^{-3}$	N m s^2
I_Y	Body Moment of Inertia around y axis	$8 \cdot 10^{-3}$	N m s^2
I_Z	Body Moment of Inertia around z axis	$14.2 \cdot 10^{-3}$	N m s^2
I_r	Rotational Moment of Inertia	$1.08 \cdot 10^{-6}$	N m s^2
b	Thrust Coefficient	$54.2 \cdot 10^{-6}$	N s^2
d	Drag Coefficient	$1.1 \cdot 10^{-6}$	N m s^2
g	Acceleration due to Gravity	9.81	m s^{-2}

Table 5.1: *Parameters for Simulation*

5.1 Open loop simulations for the basic movements

In this section the four basic movements of the quadrotor will be simulated. The purpose of this simulation is to show how the quadrotor moves in space by changing its angular velocities. In the following tests the results are based on a open loop system of the quadrotor. This means that the quadrotor does not reach a desired value by feedback control .Alternating the four angular velocities by the right values leads to a different command. The commands that will be demonstrated have been mentioned in chapter 2 and include: *throttle* ,*roll* ,*pitch* and *yaw* command.

5.1 .1 Throttle Command

This command is generated by increasing all the angular velocities of the quadrotor by the same amount. This leads to an angular acceleration around z axis which raises the quadrotor higher .

In order to produce the throttle command the angular velocities should all have same value, e.g.$\boldsymbol{\Omega_1 = \Omega_2 = \Omega_3 = \Omega_4 = 250}$ $(\frac{rad}{s})$

The simulation results are shown in the figs 5.2, 5.3, 5.4

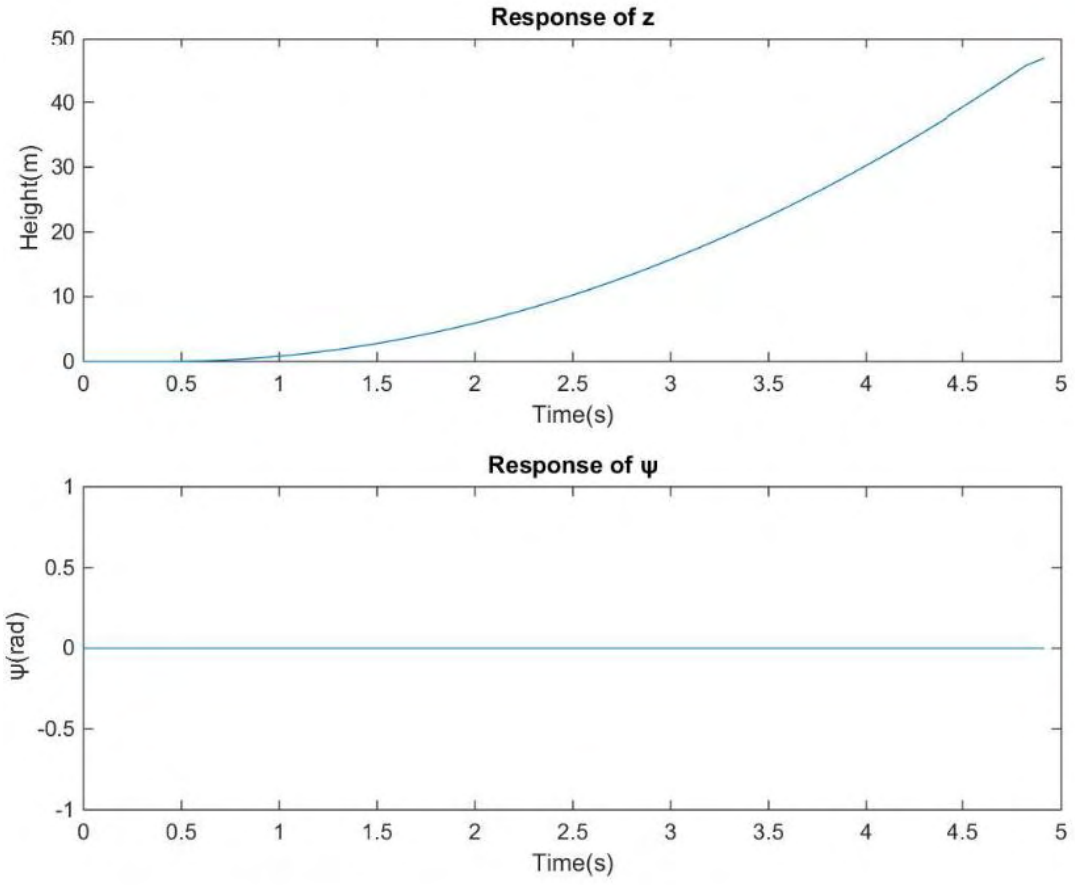


Figure 5.2: *Response of z and ψparameters for Throttle Command*

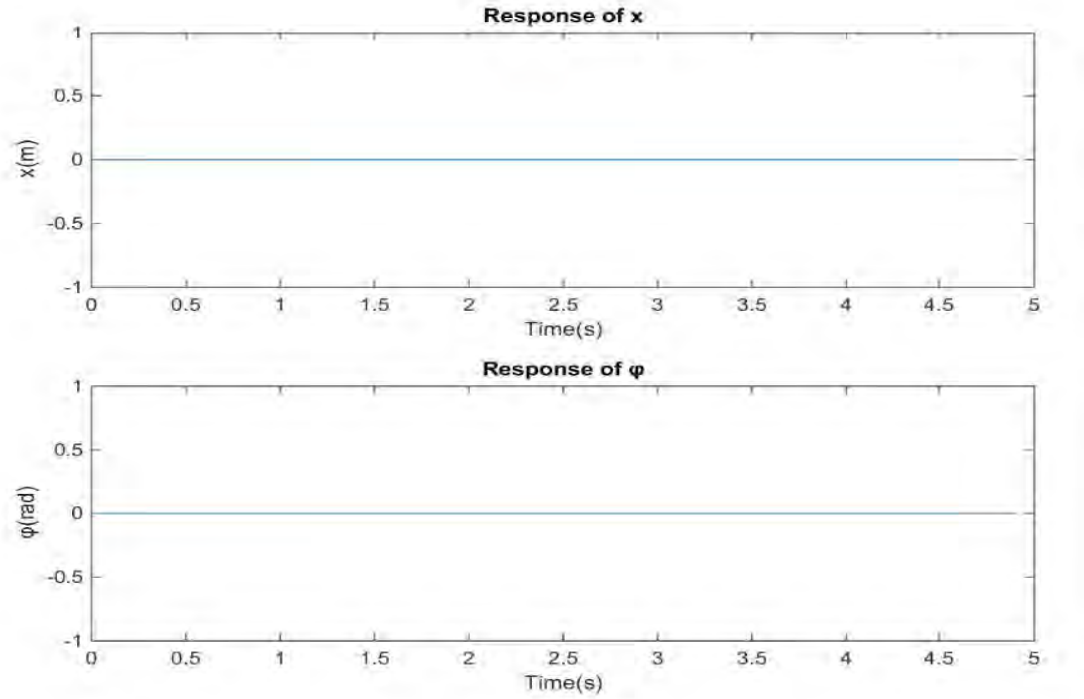


Figure 5.3: *Response of x and φparameters for throttle Command*

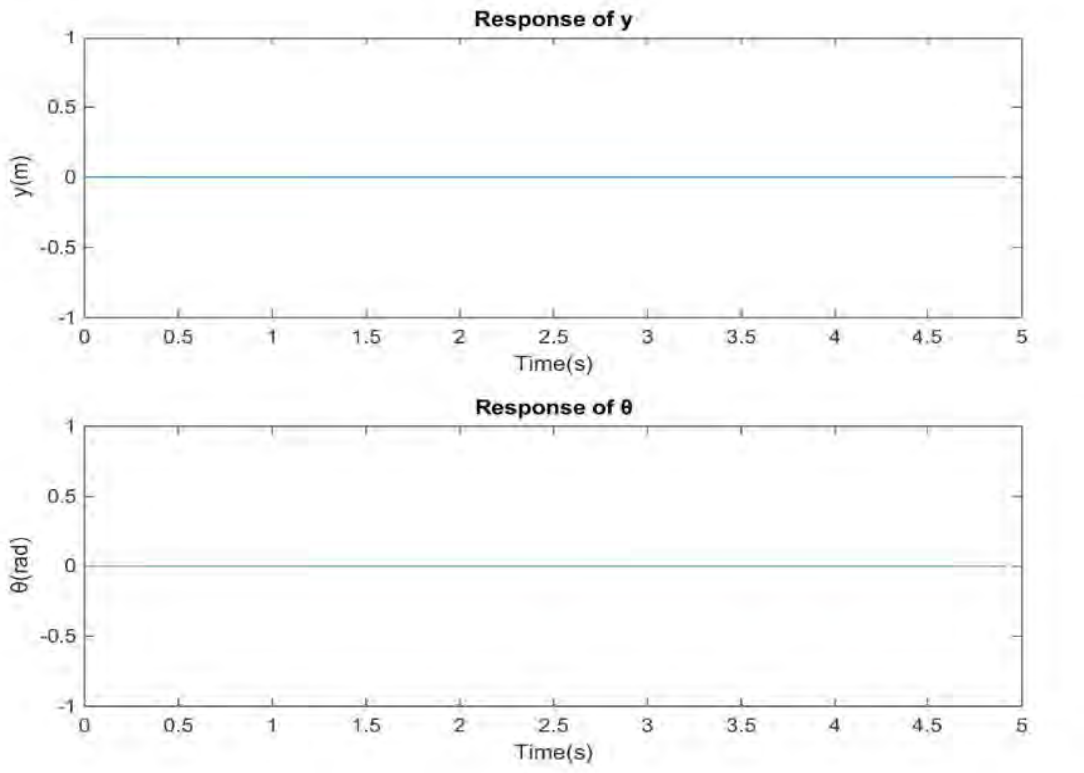


Figure 5.4: *Response of y and θ parameters for throttle Command*

The above 6 diagrams show the response of the quadrotor when the 4 angular velocities take the value 250 rad/s. The only parameter that change over time is the height z .The 5 remaining parameters x, y, φ, θ, ψ stay unchanged. As it is shown in figure 5.2by applying the throttle movement only the height increases over time which means that the quadcopter moves upwards .

5.1 .2 Roll Command

This command is generated by increasing the angular velocity $\Omega_4$by a small amount $\Delta\Omega$ and by decreasing by the same amount the angular velocity Ω_2 while maintaining the rest angular velocities Ω_1 and Ω_3 at hovering state. This leads to torque around x axis and an angular acceleration of φ. .

In order to produce the Roll command the angular velocities should be set in the following way:

$$\boldsymbol{\Omega_1 = \Omega_3 = \Omega_{hovering} = 212.718}\left(\frac{rad}{s}\right),$$

$$\boldsymbol{\Omega_4 = 215}\left(\frac{rad}{s}\right),$$

$$\boldsymbol{\Omega_2 = 211}\left(\frac{rad}{s}\right)$$

The simulation of Roll command are is shown in figures 5.5, 5.6, 5.7

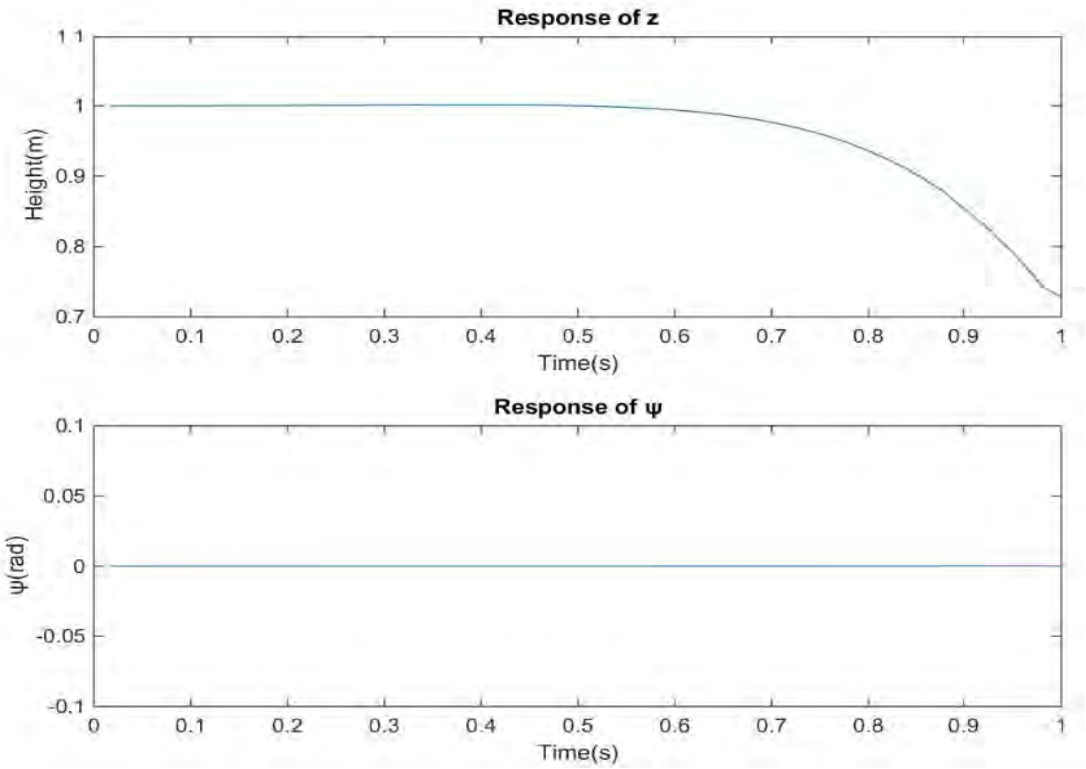


Figure 5.5: *Response of z and ψ parameters for Roll Command*

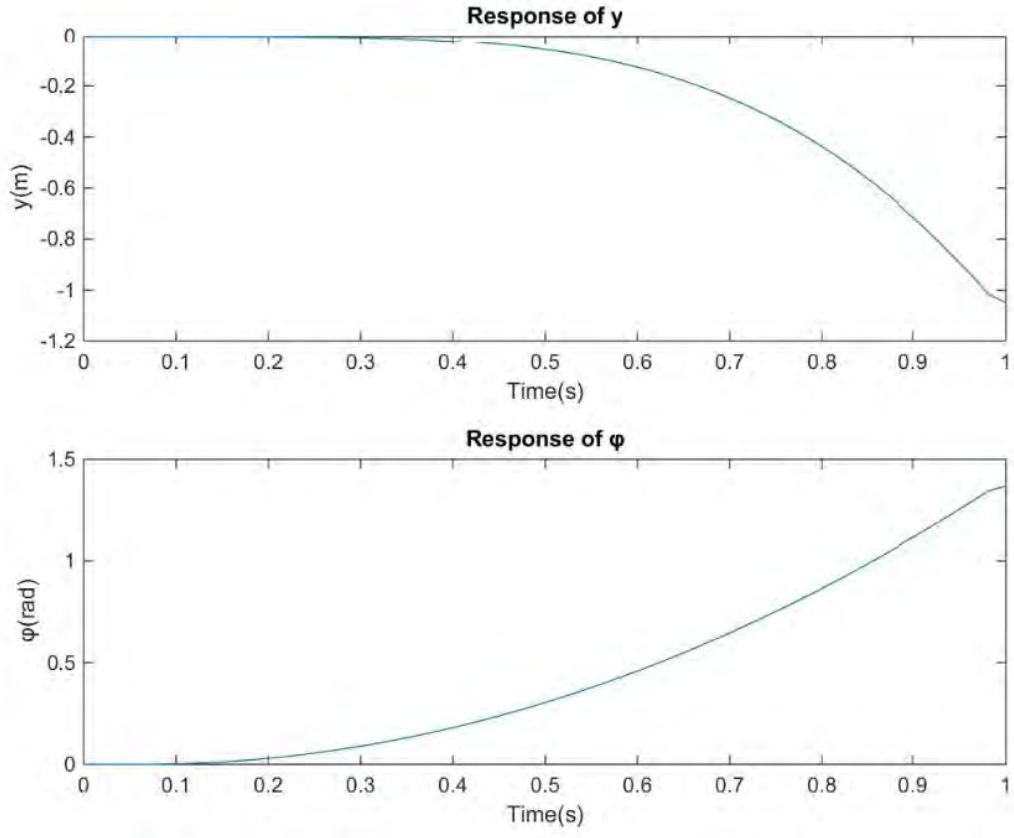


Figure 5.6: *Response of y and φ parameters for Roll Command*

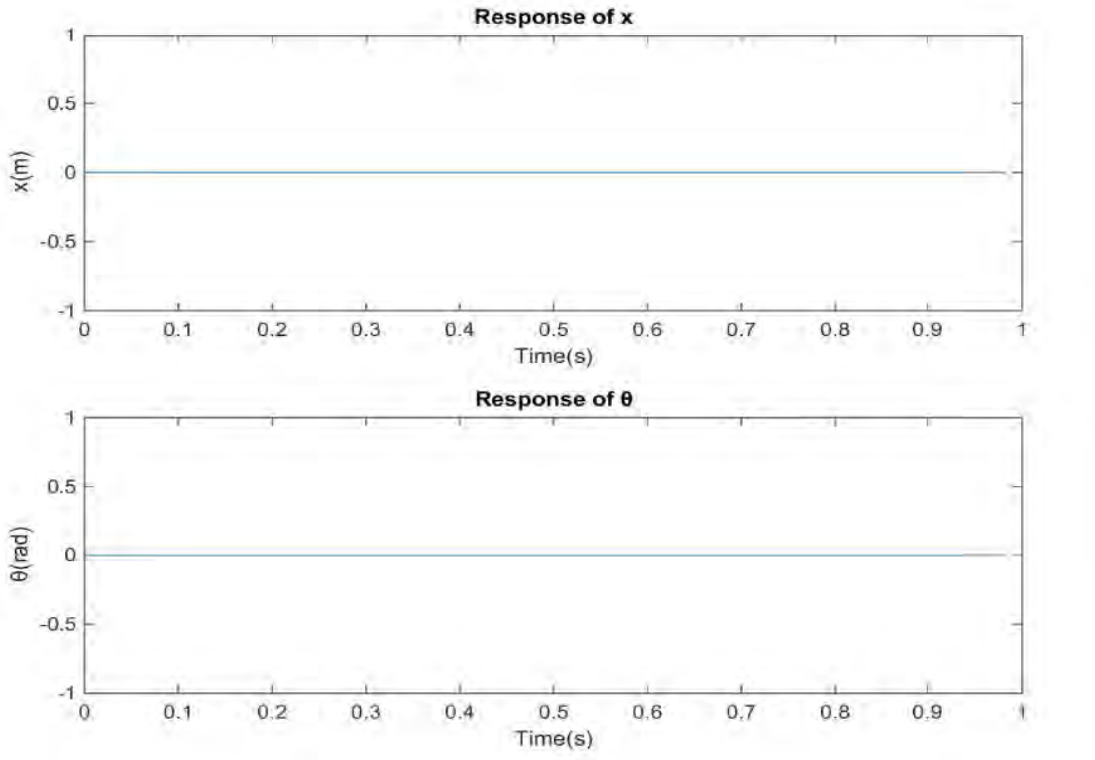


Figure 5.7: *Response of x and θ parameters for Roll Command*

The above six diagrams show the movement of the quadrotor while it performs the roll command. As it is shown form the diagrams the Euler's angle θ,ψremain almost unchanged . The height *z* remains unchanged . The main change by applying the Roll command is that the Euler angle φincreases its value over time. This change in angle φresults in motion in y-plane as it is shown from figure 5.6.

5.1 .3 Pitch Command

This command is generated by increasing the angular velocity Ω_3 by as small amount $\Delta\Omega$ and by decreasing by the same amount the angular velocity $\Omega_1$while maintaining the remaining angular velocities Ω_2 and Ω_4 at hovering state. This leads to torque around y axis and an angular acceleration of θ. .

In order to produce the Pitch command the angular velocities should be set in the following way:

$$\boldsymbol{\Omega_2 = \Omega_4 = \Omega_{hovering} = 212.718}\left(\frac{rad}{s}\right),$$

$$\boldsymbol{\Omega_3 = 216}\left(\frac{rad}{s}\right), \boldsymbol{\Omega_1 = 210}\left(\frac{rad}{s}\right)$$

The simulation of Pitch command are shown in figures 5.8, 5.9, 5.10:

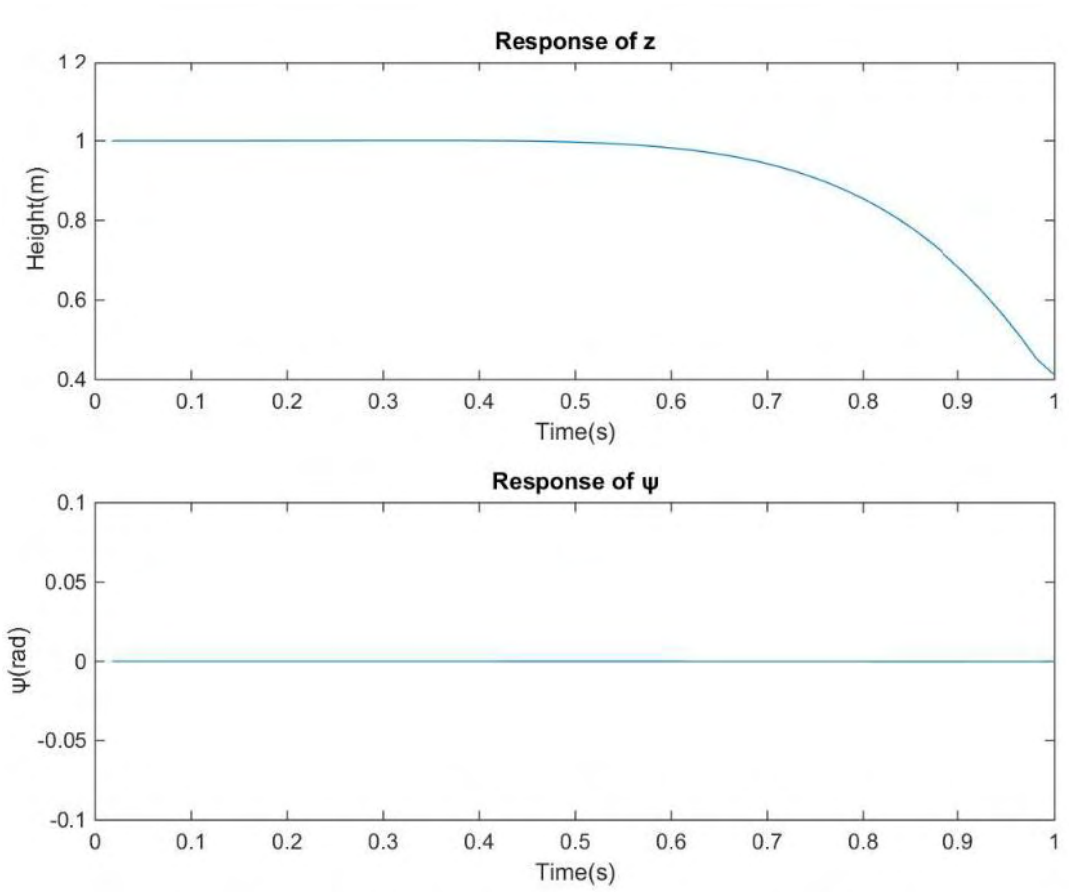


Figure 5.8: *Response of z and ψ parameters for Pitch Command*

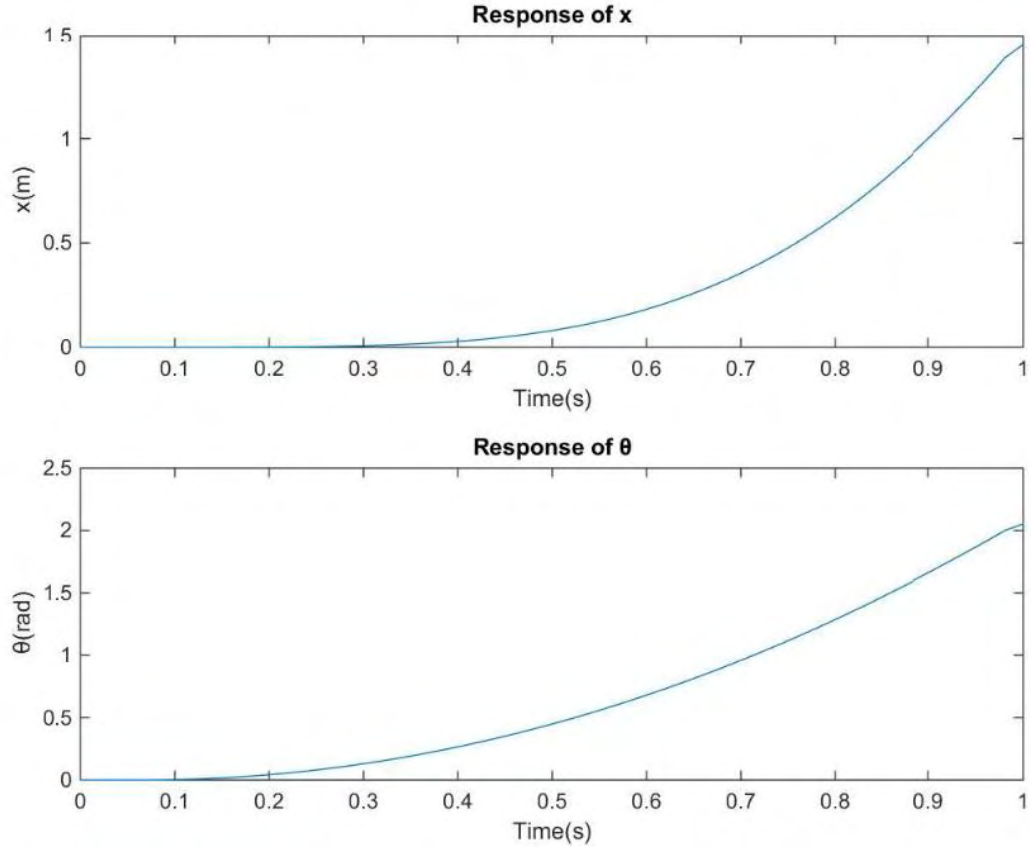


Figure 5.9: *Response of x and θ parameters for Pitch Command*

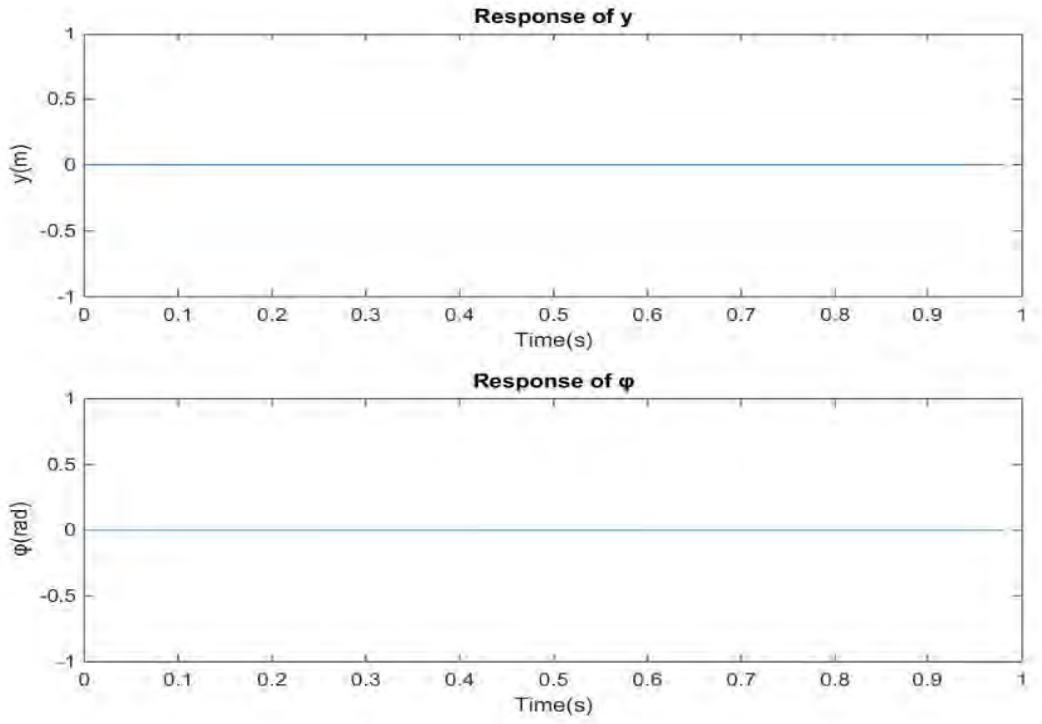


Figure 5.10: *Response of y and φ parameters for Pitch Command.*

The above six diagrams show the movement of the quadrotor while it performs the pitch command. As it is shown form the diagrams the Euler's angle φ,ψremain unchanged . The height z is remaining unchanged . The main change by applying the Pitch command is that the Euler angle θincreases its value over time. This change in angle θresults in motion in x-plane as its shown from figure 5.9.

5.1.4 Yaw Command

This command is generated by increasing the angular velocities$\Omega_4$and $\Omega_2$by a small amount $\Delta\Omega$ and simultaneously decreasing by the same amount the angular velocities$\Omega_1$and Ω_3. This leads to torque around z axis and an angular acceleration of ψ. As a result of this action the quadrotor turns around its self in the z axis .

In order to produce the Yaw command the angular velocities should be set in the following way:

$$\boldsymbol{\Omega_2 = \Omega_4 = 215}\left(\frac{rad}{s}\right)$$

$$\boldsymbol{\Omega_3 = \Omega_1 = 211}\left(\frac{rad}{s}\right)$$

The simulation for the Yaw command are shown in figures 5.11, 5.12, 5.13:

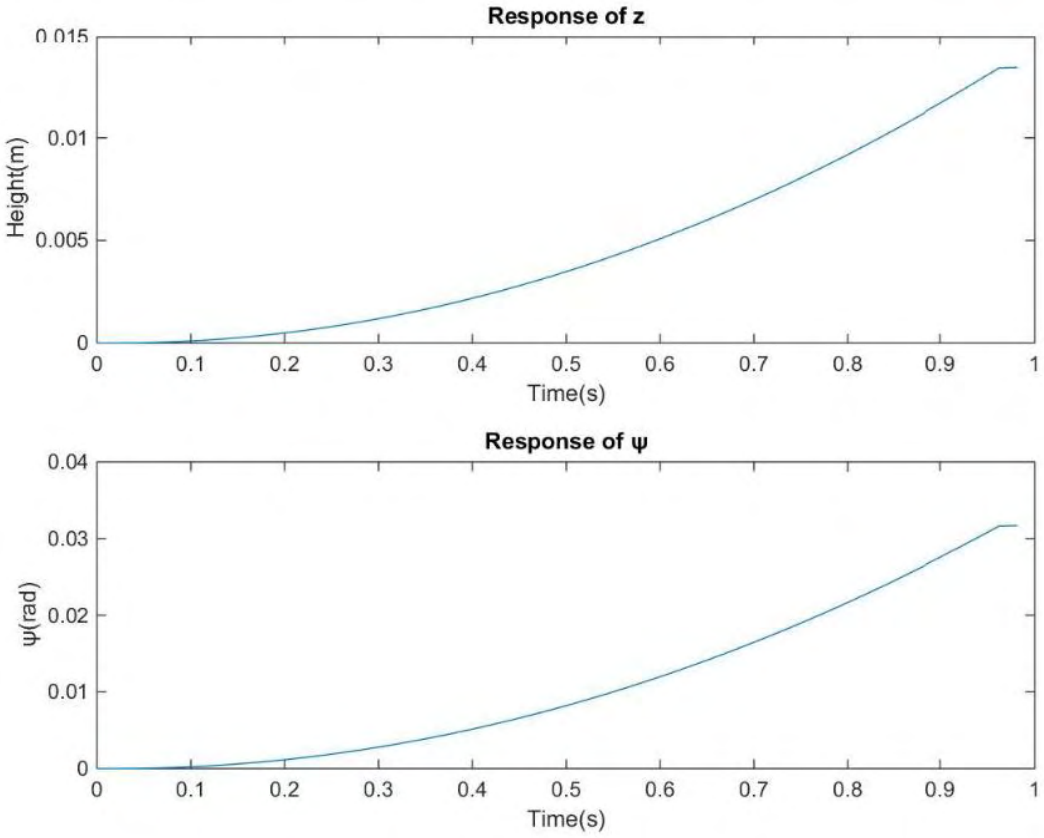
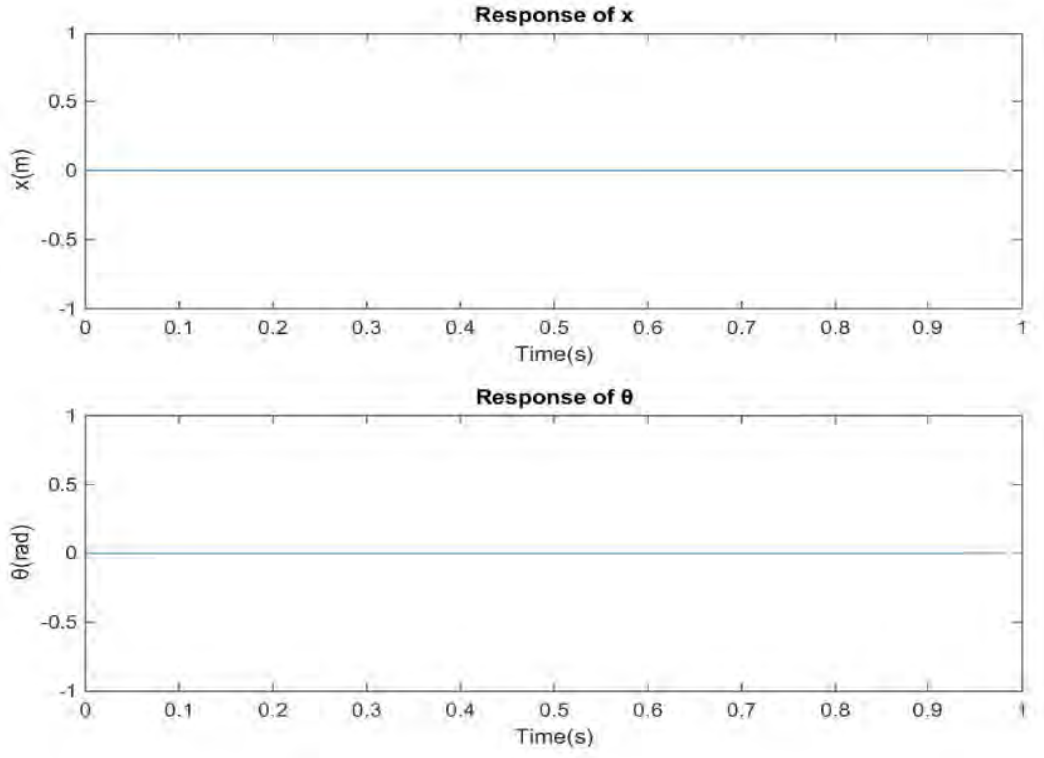


Figure 5.11: *Response of z and ψ parameters for Yaw Command*



5.12: *Response of x and θ parameters for Yaw Command*

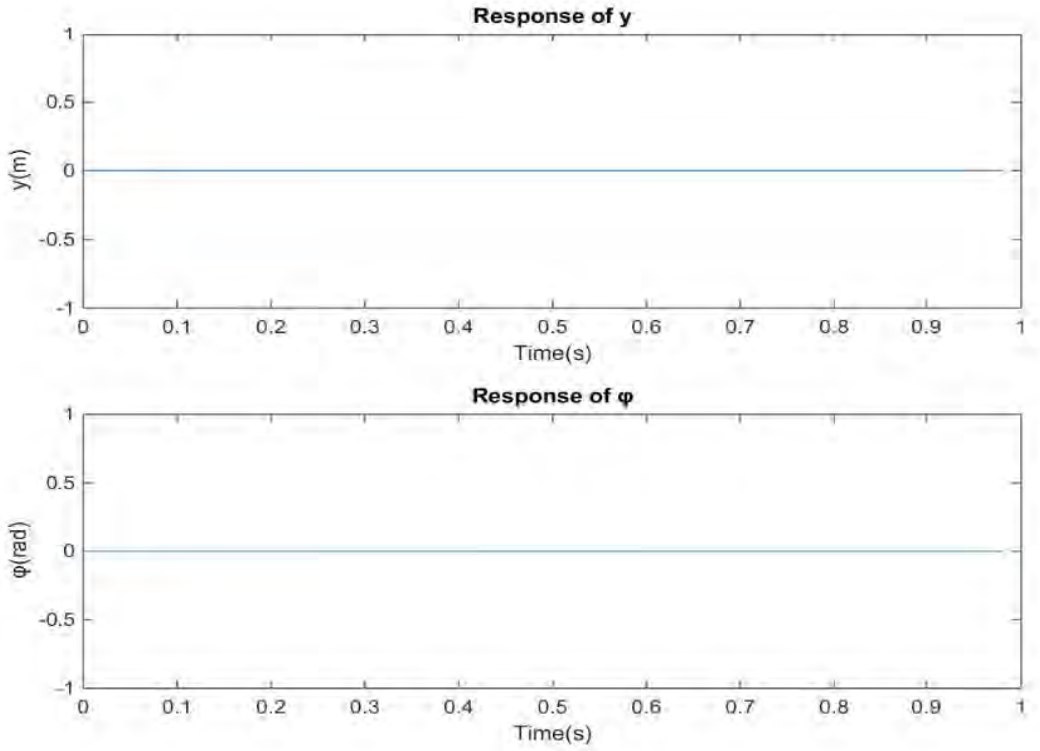


Figure 5.13: *Response of y and φ parameters for Yaw Command.*

The above six diagrams show the movement of the quadrotor while it performs the yaw command. The goal by performing this command is to change its direction in ψangle while maintaining the other unchanged. As it is shown from figure 5.10 the quadrotor change its angle ψ over time . This leads to a change in altitude z by performing this change in angle. The other parameters *x, y, φ, θ*remain unchanged as the diagrams 5.12 and 5.13 show.

5.2 Closed loop simulations using PID Controllers

Because of the under-actuated nature of the quadrotor from the 6 D.O.F. only a maximum of four can be controlled successfully. The parameters chosen for control are related to the four basic movement that the quadrotor performs in order to flight

in 3-D space. The parameters altitude z and Euler's angles φ,θ,ψwere chosen to be controlled based on the Throttle U_1, Roll U_2 ,Pitch U_3 and Yaw U_4 commands .

PID controllers were used in these simulations in order to achieve the requested set points of z ,φ, θ ,ψ .Four PID controllers where used one for each parameter. The controllers where tuned manually with the criterion of minimizing the sum of square error (SSE), which can be calculated as follows:

$$SSE = \sum_{i=1}^{n}(x_i - x_d)^2 \qquad (5.1)$$

where n is the number of observations x_i is the i-th observation and $x_d$the desired set point value.

Choosing the parameters of the PID controllers manually is difficult task as it needs a great deal of trial and error tries. The process of the tuning starts by picking 3 values for the P, I and D gains. Then the SSE is measured. By changing the gains by a little the new SSE is measured. If the new SSE is smaller than the previous one the last values of the PID controller are kept and a new set of gains are chosen. This procedure continues until the SSE is considered small enough.

A selected set of tries is shown in the following Table by performing manual tuning of the four PID controllers based on minimizing the SSE:

Height					*Roll-Pitch*					*Yaw*				
P_z	I_z	D_z	N_z	SSE_z	P_φ	I_φ	D_φ	N_φ	SSE_φ	P_ψ	I_ψ	D_ψ	N_ψ	SSE_ψ
1	1	1	20	780.8	0.19	0.009	0.97	6	0.425	0.8	0.05	4.1	8	2.842
1	0.5	0.9	20	451.2	0.2	0.009	1	6	0.422	0.6	0.05	3.5	8	2.792
1	0.1	0.9	20	368.2	0.25	0.009	1	5	0.4123	0.3	0.001	3.5	8	2.744
0.5	0.1	0.9	10	319.2	0.26	0.01	1.2	5	0.3978	0.3	0	3.5	4	2.331
0.4	0.01	0.8	6	277.6	0.28	0.01	1.5	4	0.377	0.27	0	3.1	2	2.192
0.4	0.001	0.78	2	232.4	0.33	0.01	1.8	4	0.366	0.26	0	3.15	2	2.185
0.39	**0.001**	**0.78**	**2**	**232**	**0.39**	**0.012**	**1.9**	**5**	**0.3634**	**0.25**	**0**	**3.15**	**2**	**2.184**

Table 5.2: *Tuning for PID parameters*

As Roll and Pitch are symmetrical motions, the same tuning was used for both PID controllers.

The best tuning parameters where given in bold for the 4 PID controllers.

The initial conditions of the quadrotor 's position are [x y z φ θ ψ] = [0 0 1 0 0 0].

The quadrotor at time t=0 (s) is performing a stationary flight at altitude z=1(m).

5.2.1 Reaching a certain altitude z

At the first set of simulations a desired height of z = 5 (m) is given while the rest of the Euler angles are set to 0, thus performing the Throttle movement in space.

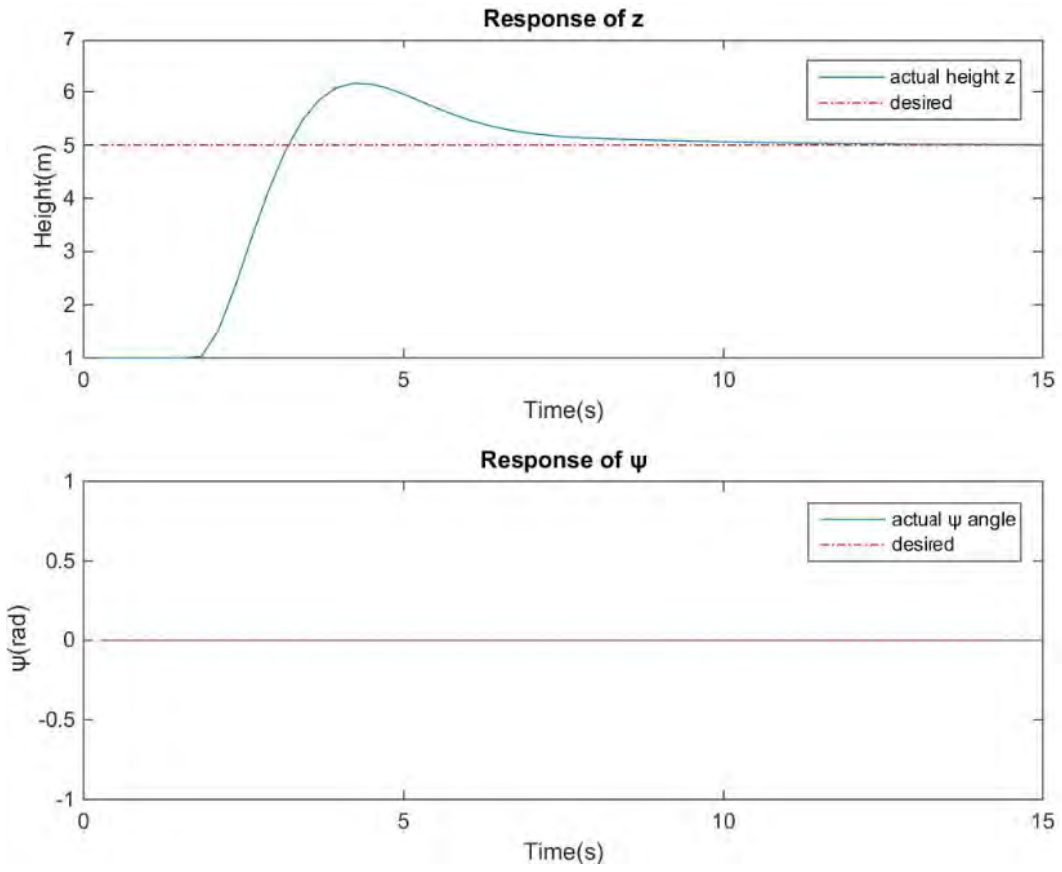


Figure 5.14: *Response of z and ψparameters*

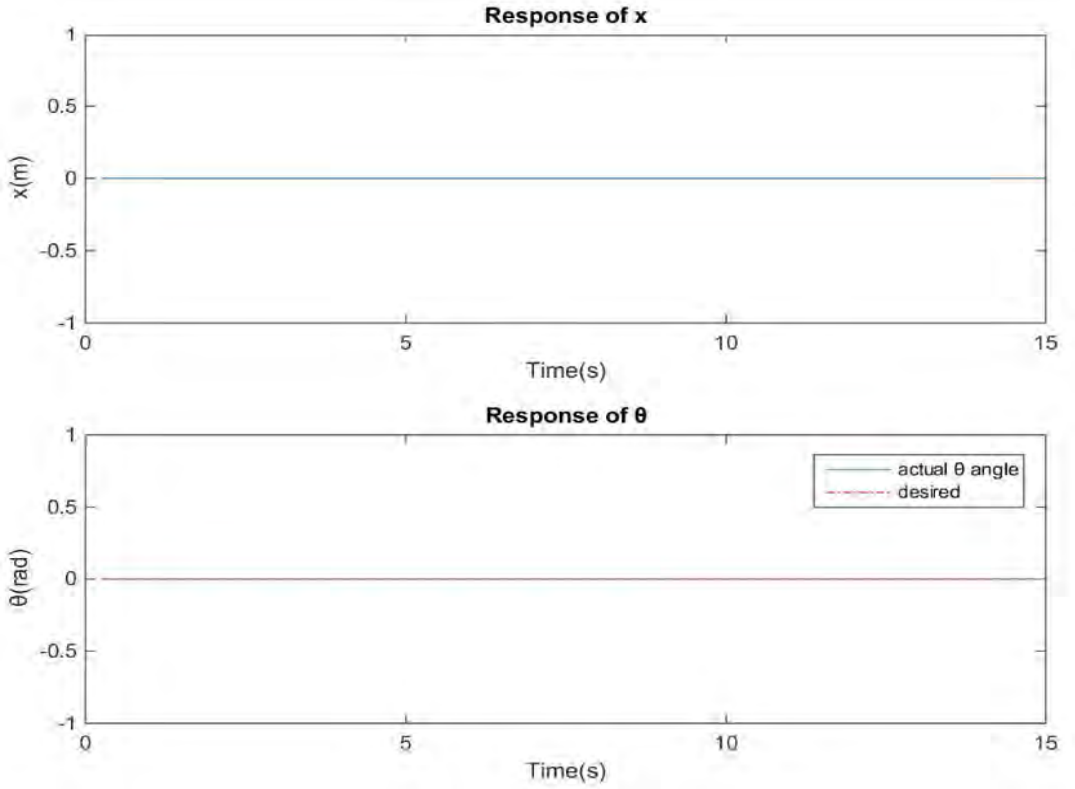


Figure 5.15: *Response of x and θparameters*

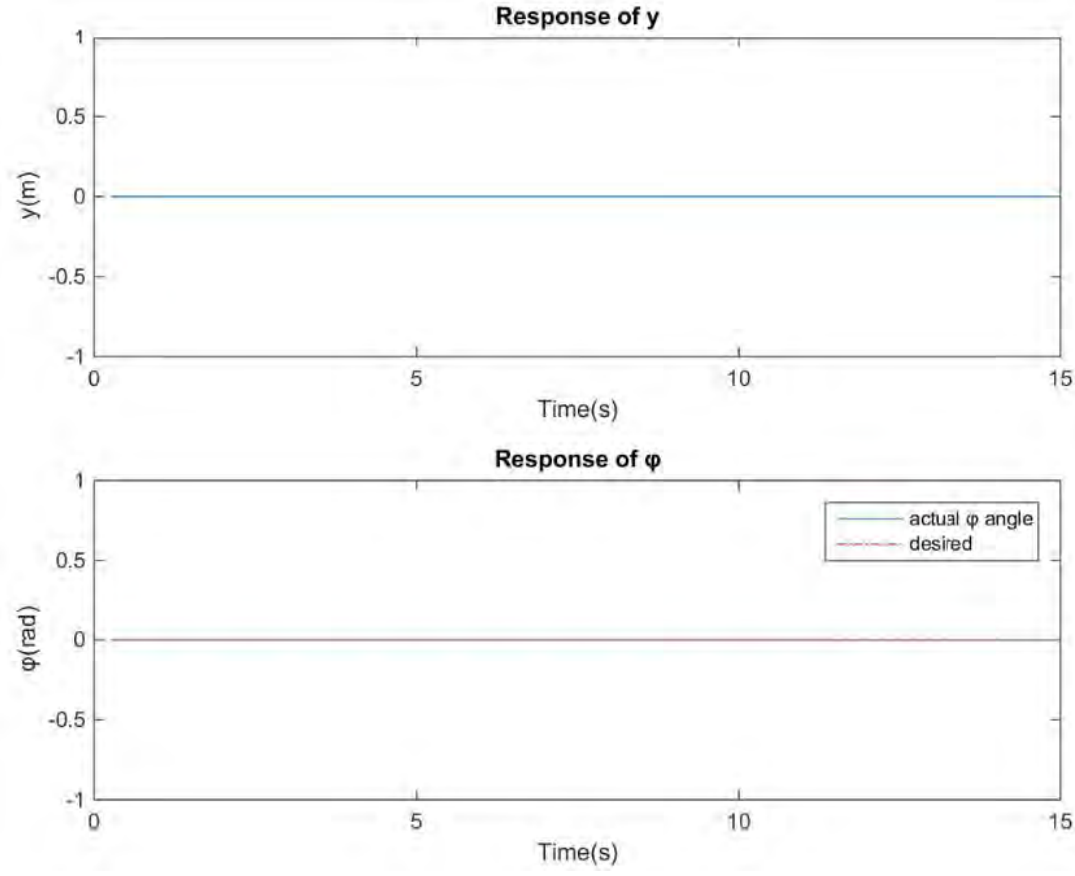


Figure 5.16: *Response of y and φparameters*

5.2.2 Reaching a certain Euler angle φ

At the second set of simulations a desired Euler angle of φ = 0.2 (rad) is given, while the rest of the Euler angles are set to 0 at altitude z=1 (m),thus performing the Roll movement.

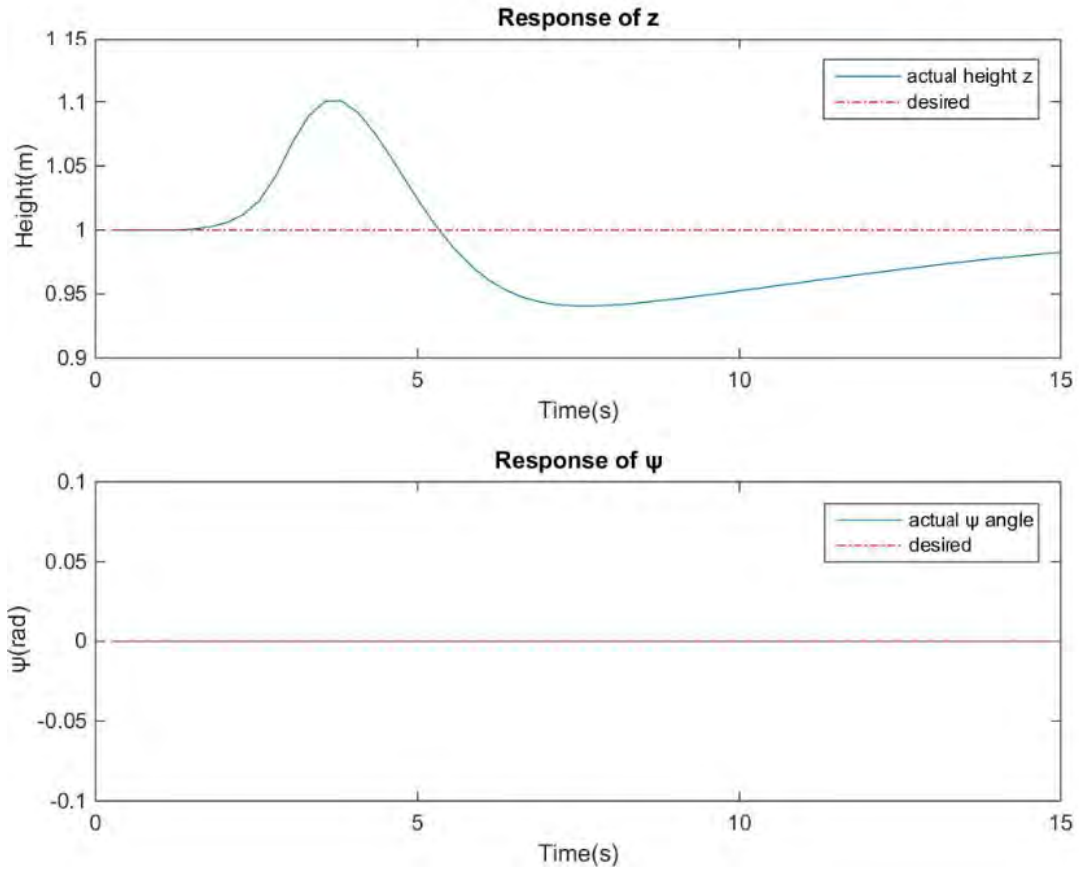


Figure 5.17: *Response of z and ψparameters*

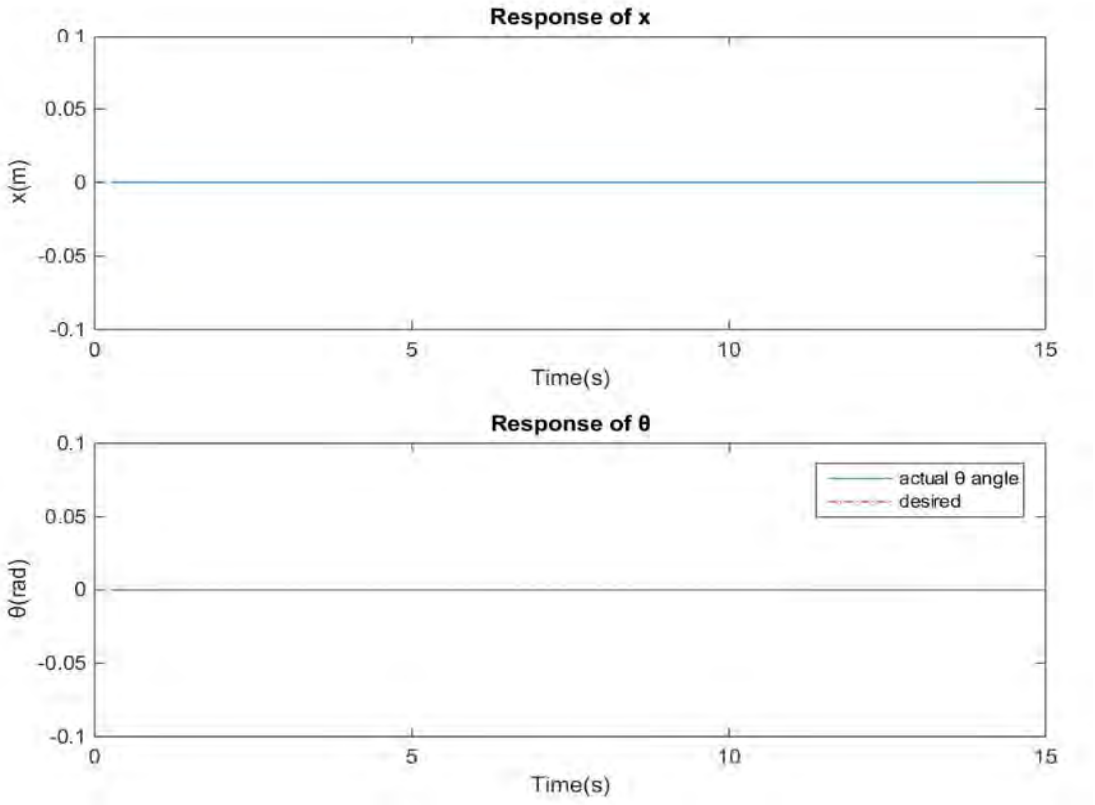


Figure 5.18: *Response of x and θparameters*

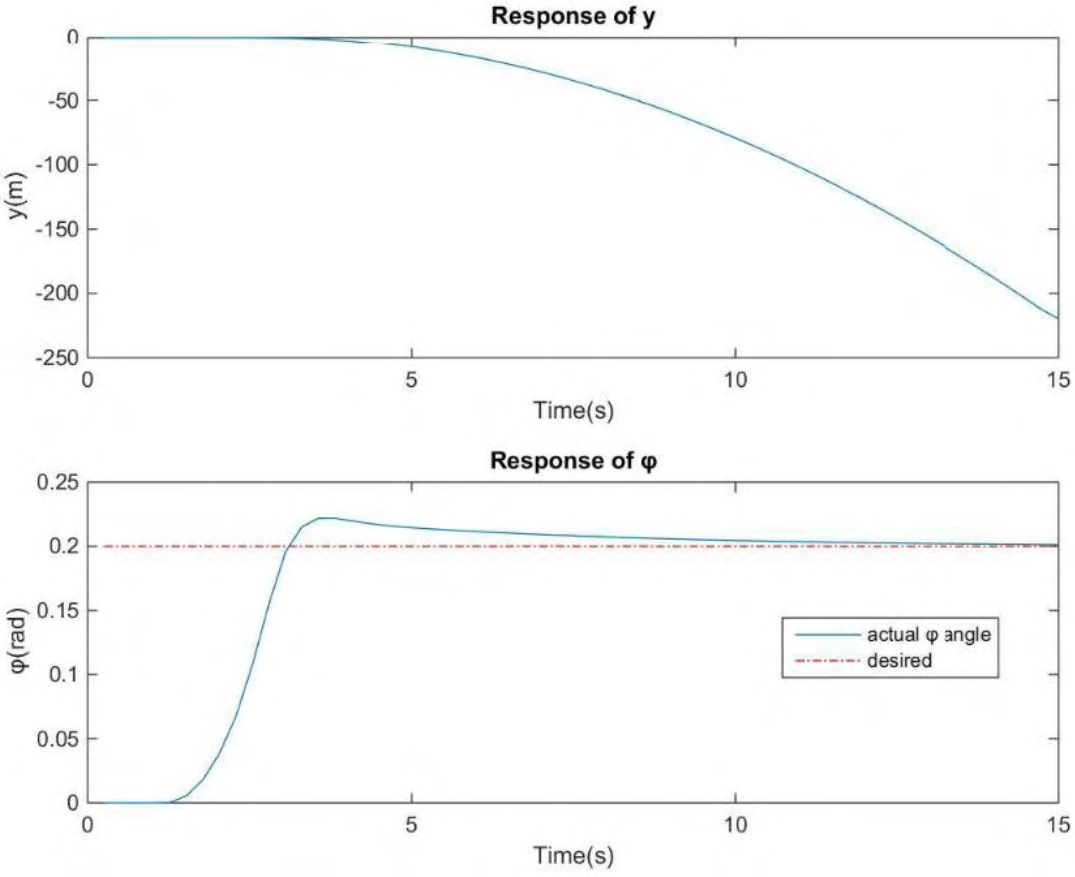


Figure 5.19: *Response of y and φparameters*

5.2.3 Reaching a certain Euler angle θ

At the next set of simulations a desired Euler angle of θ = 0.3 (rad) , the rest of the Euler angles are set 0 at altitude z=1 (m)performing Pitch movement.

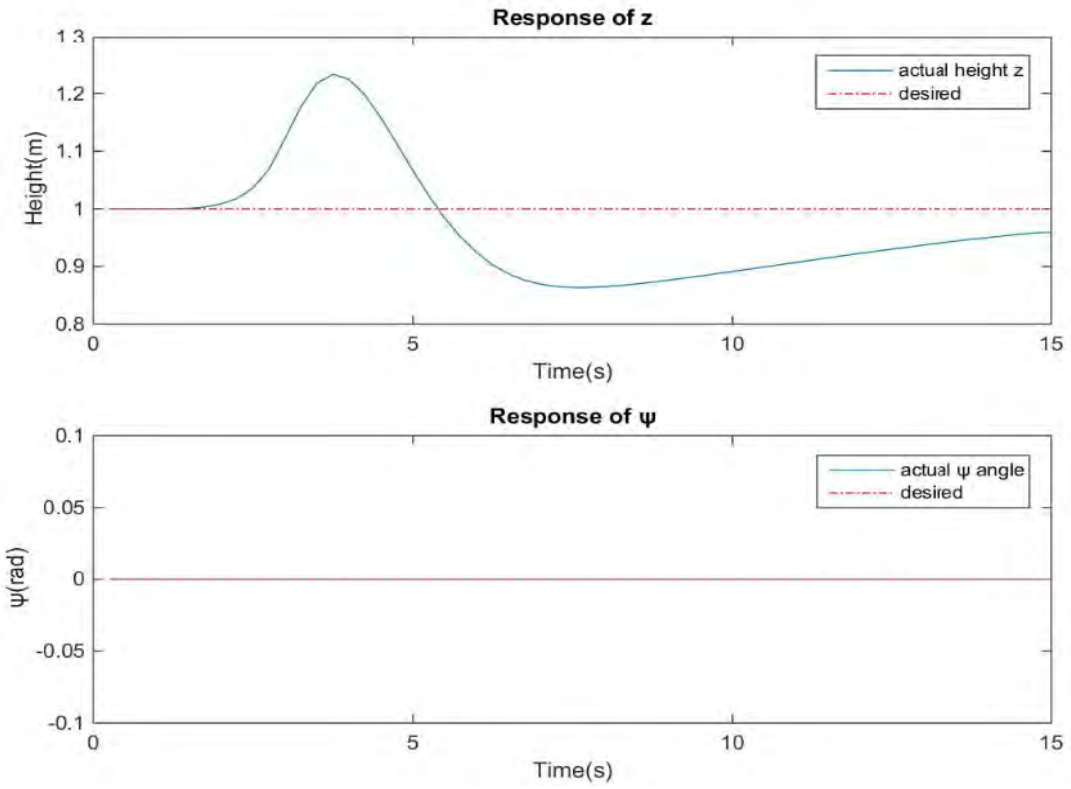


Figure 5.20: *Response of z and ψparameters*

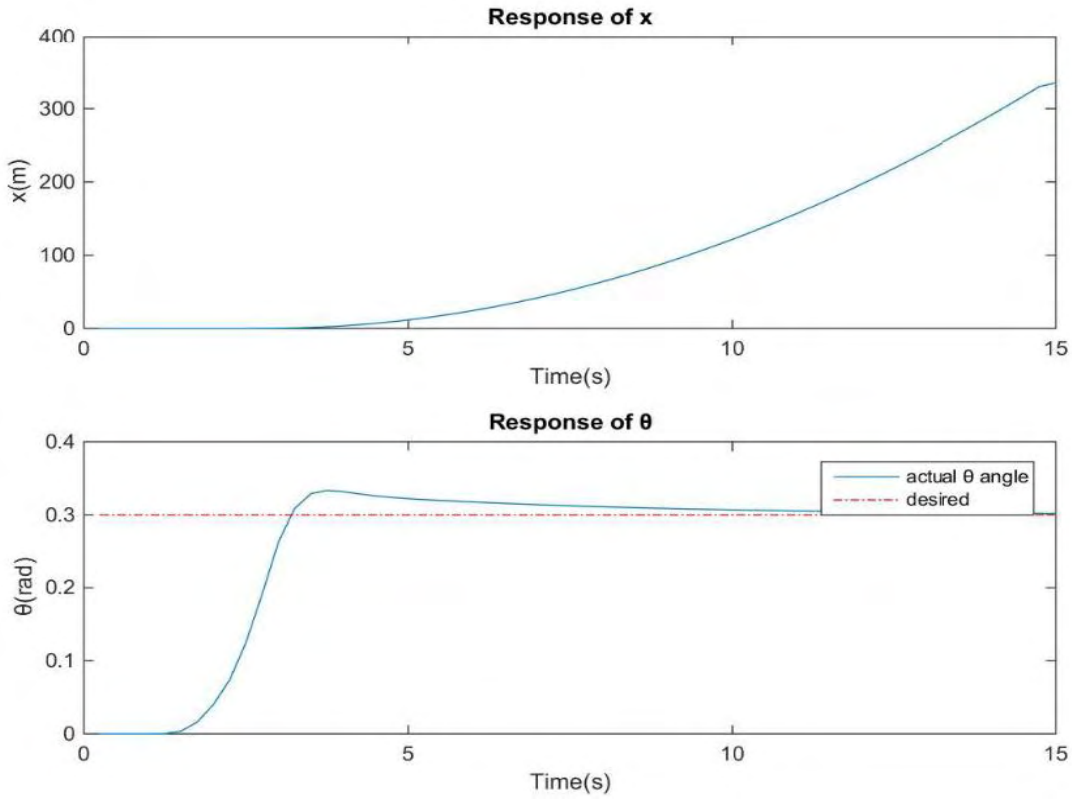


Figure 5.21: *Response of x and θparameters*

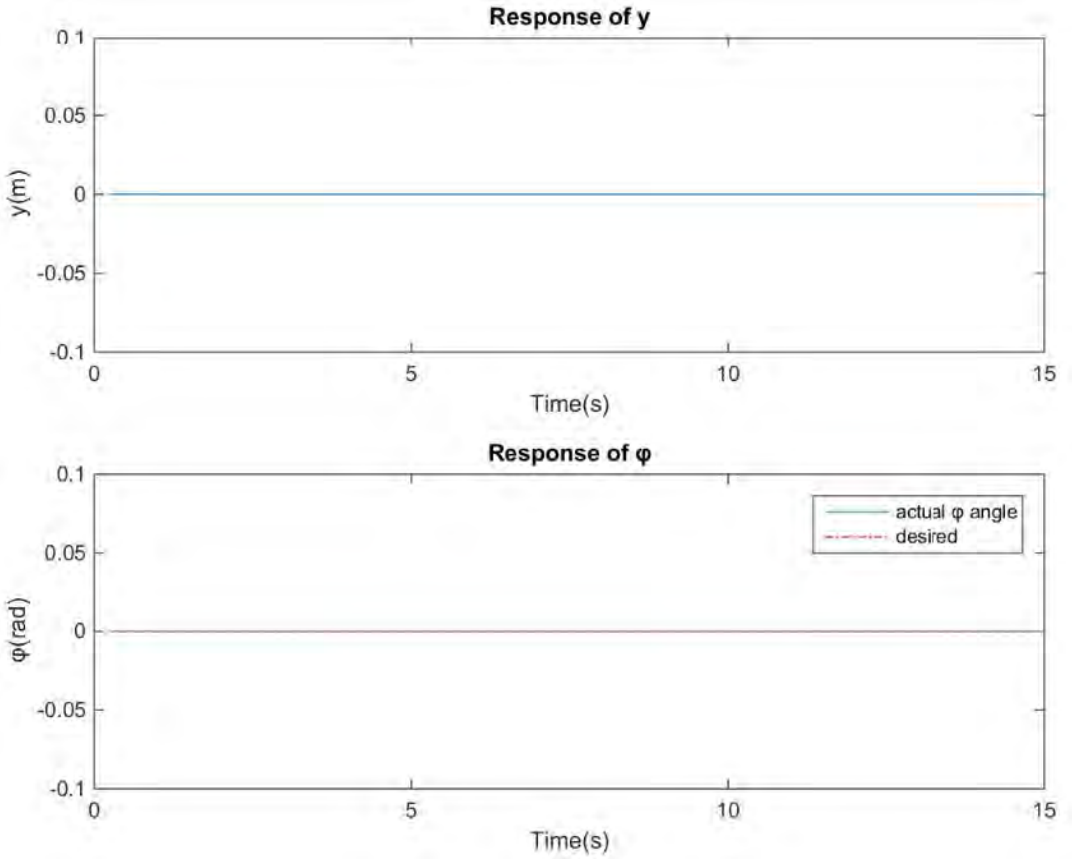


Figure 5.22: *Response of y and φparameter*

5.2.4 Reaching a certain Euler angle ψ

A desired Euler angle of ψ = 1 (rad) is given, while the rest of the Euler angles are set 0 at altitude z=1 (m) performing only yaw movement.

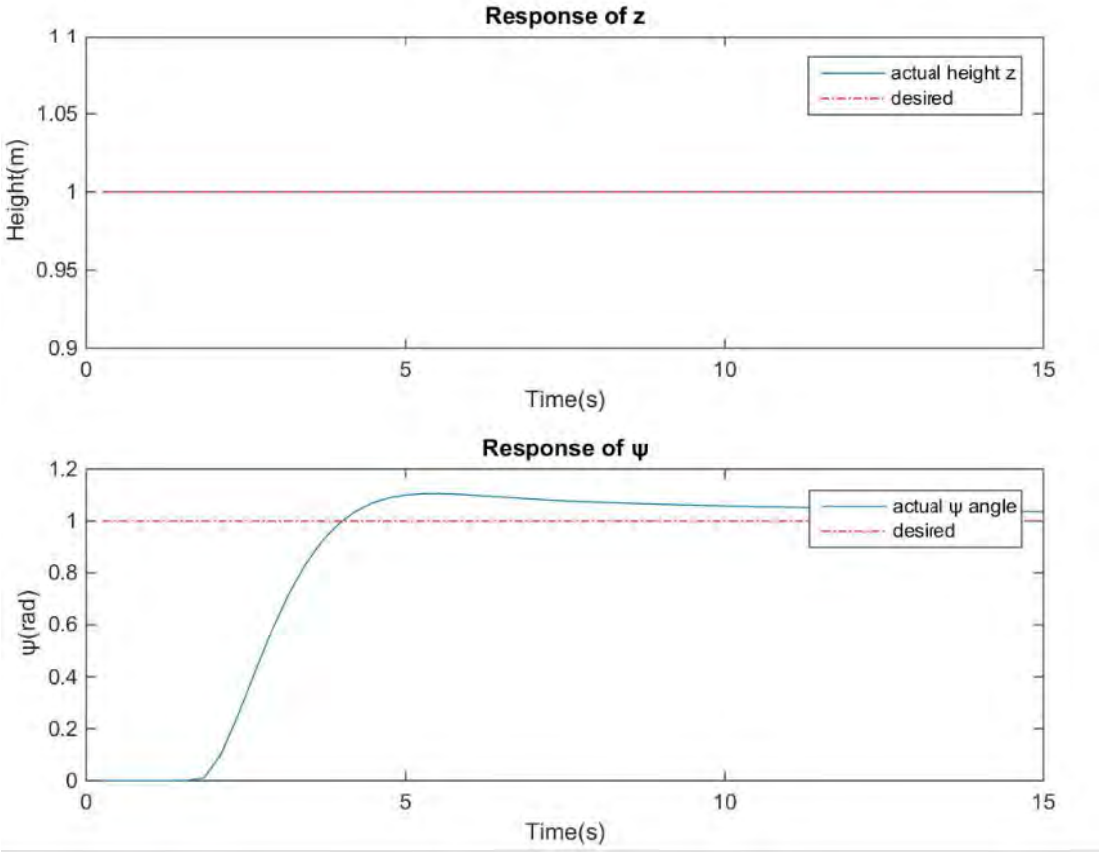


Figure 5.23: *Response of z and ψparameters*

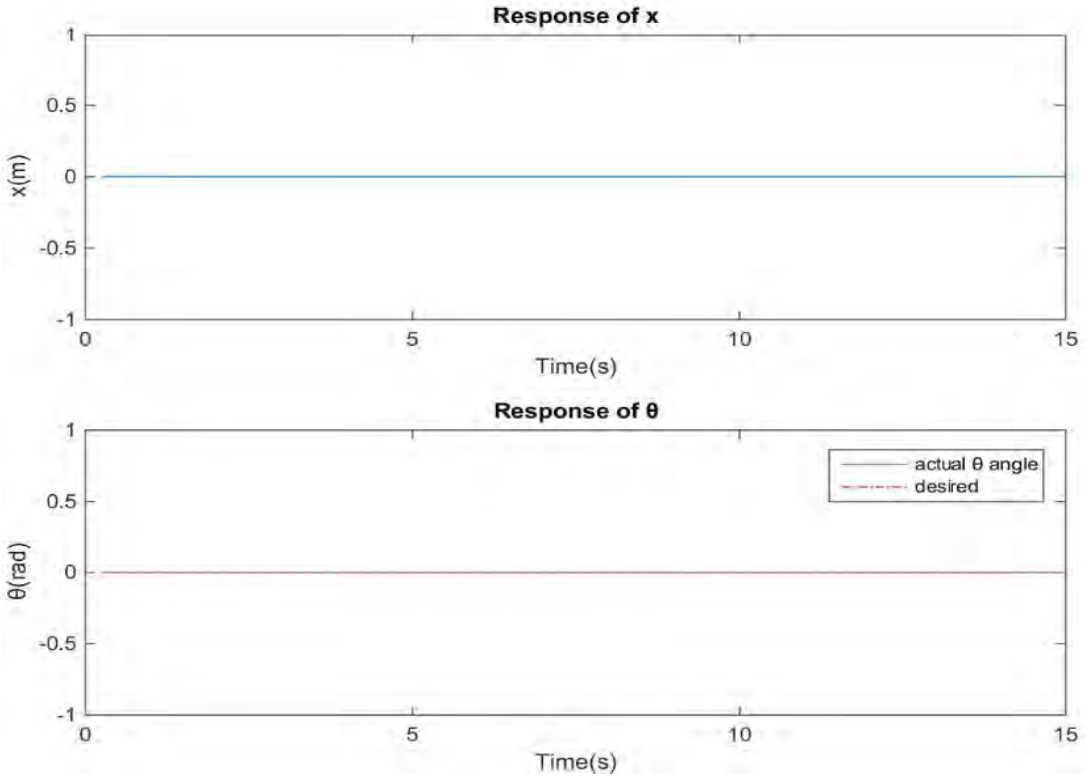


Figure 5.24: *Response of x and θparameters*

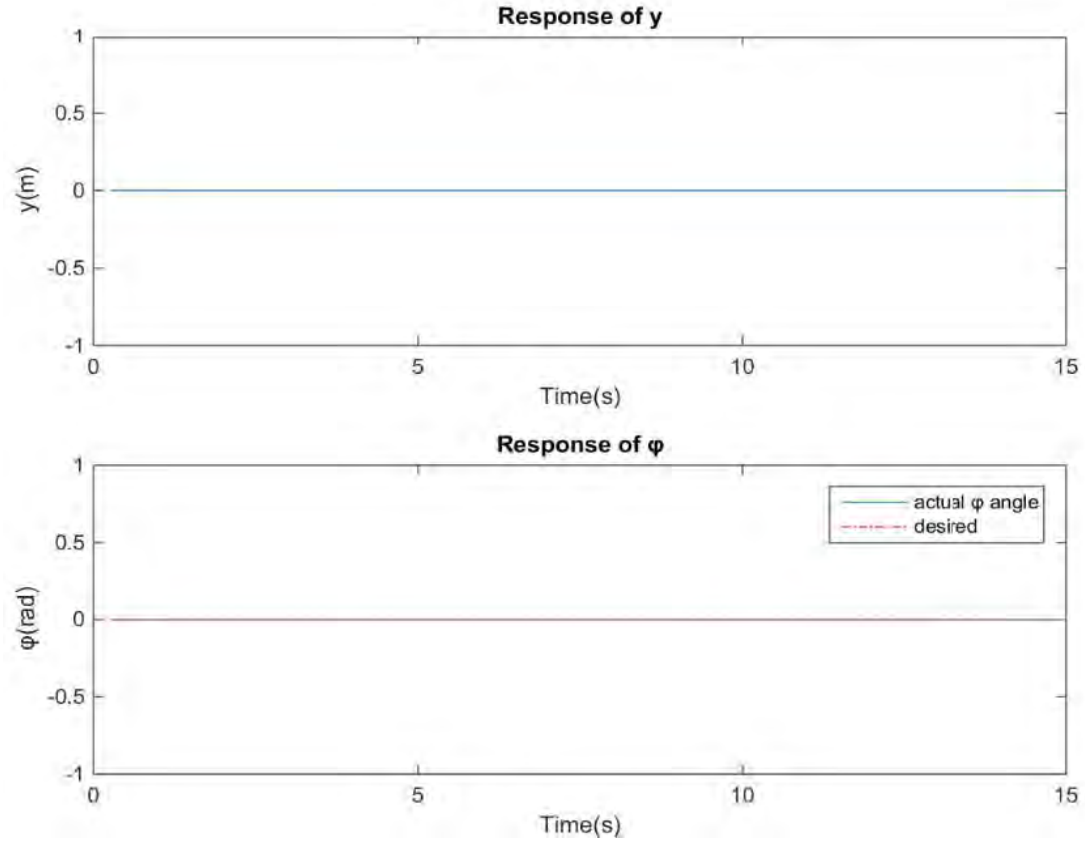


Figure 5.25: *Response of y and φparameter*

5.2.5 Reaching a complex set of desired values

The following simulations test the task of reaching a multiple set of set point values.

The reference values set were: *z=4 (m),φ=0.1(rad), θ=0.13(rad),ψ=0.3 (rad).*

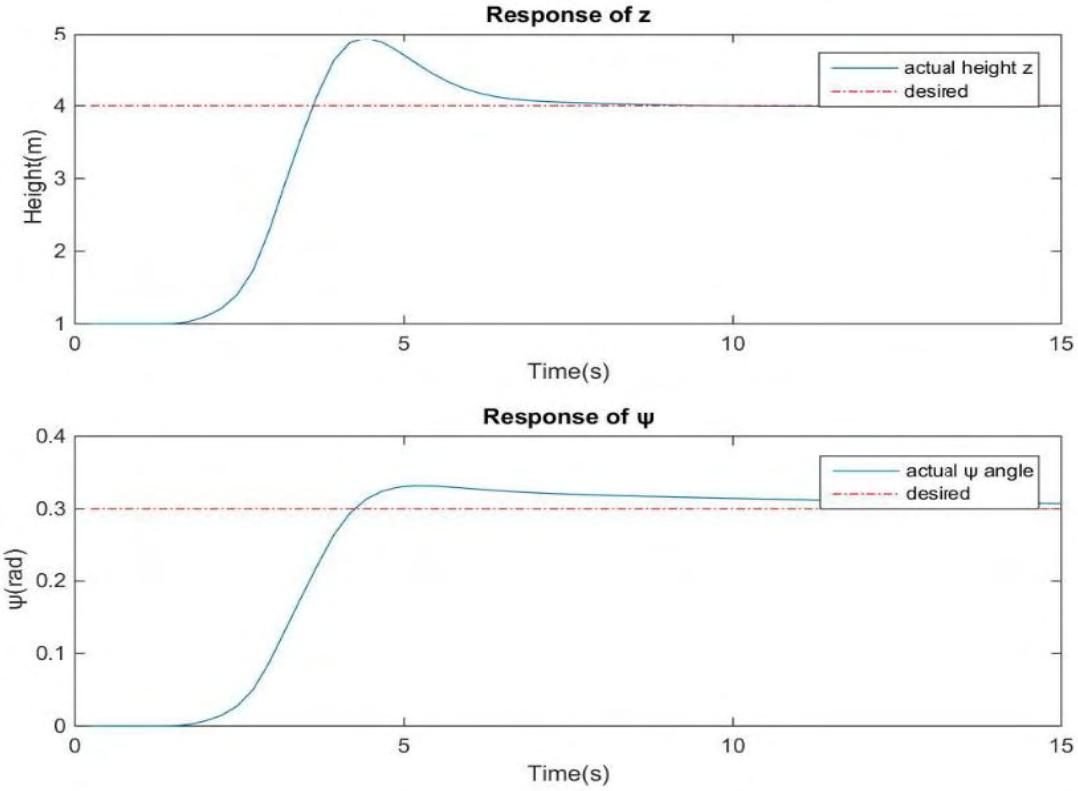


Figure 5.26: *Response of z and ψparameters*

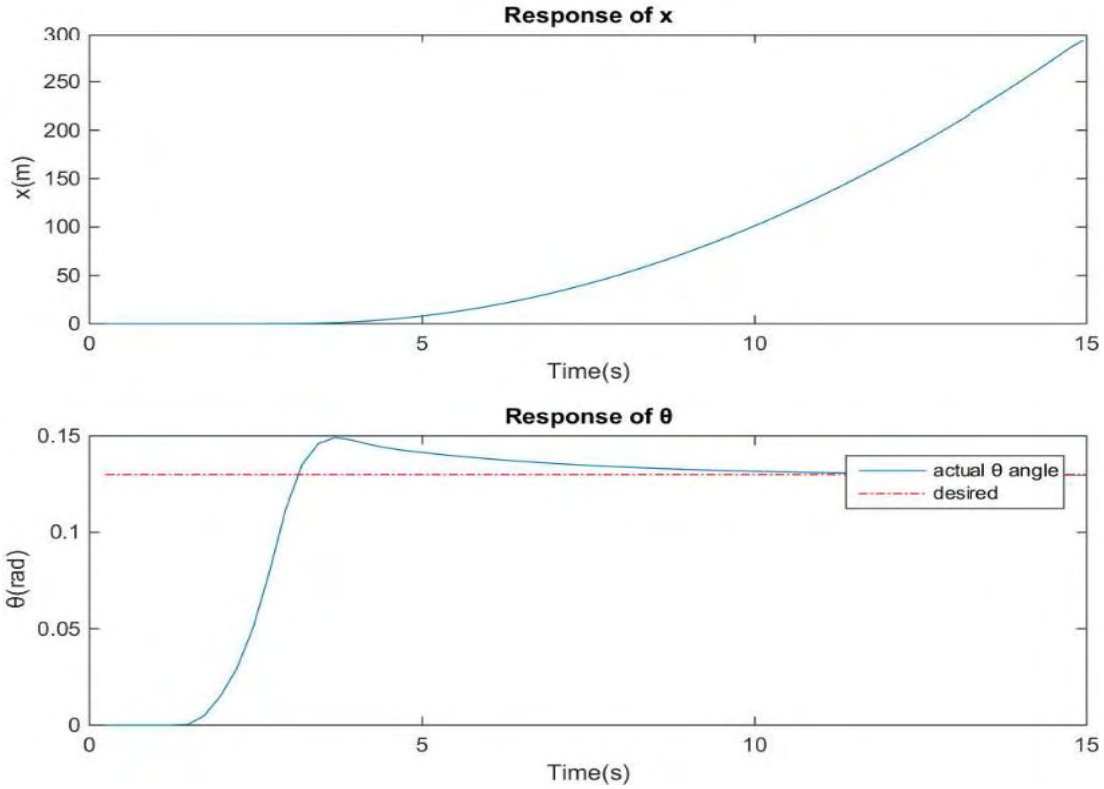


Figure 5.27: *Response of x and θparameters*

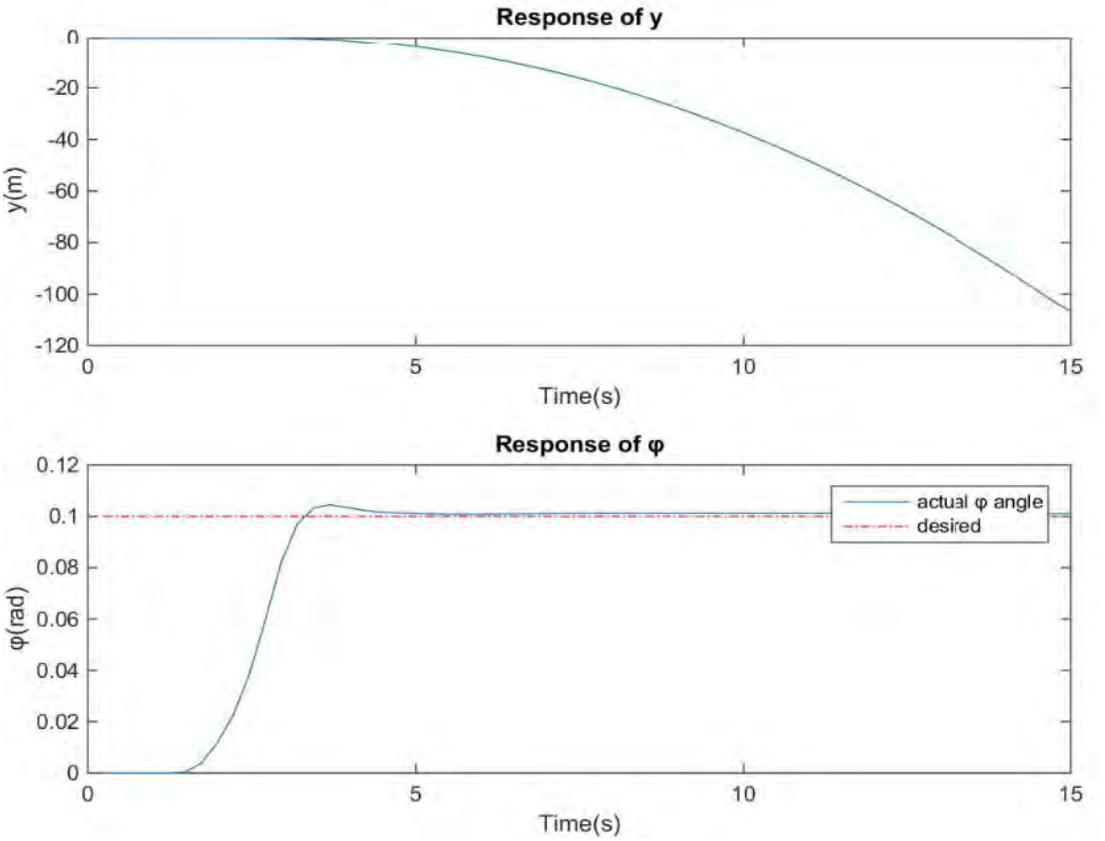


Figure 5.28: *Response of φ and y parameters*

5.3 Closed loop simulations using MPC

In this section the quadrotor will perform a group of simulations in order to evaluate the efficiency of the predictive controllers. While the predictive controller is responsible for the position control, PID controllers are used in order to control the attitude of the quadrotor. Two kind of simulations will be examined. In the first set of simulations the quadrotor will try reaching a desired set point in space while in the second the objective will be to track a certain path.

5.3.1 Reaching a certain point in space

The initial conditions of the quadrotor 's position are [x y z φ θ ψ] = [0 0 0 0 0 0] .The weight matrices of the Model Predictive Controllers are adjusted in the following way:

$$h_p = 18\,, h_c = 18\,,$$

$$Q_z = diagonal(15, \dots, 15) \text{where} Q_z \text{is a } h_p \; by \; h_p \text{ matrix}$$

$$R_z = diagonal(1, \dots, 1) \text{where} R_z \text{is a } h_c \; by \; h_c \text{ matrix}$$

$$G_z = diagonal(30, \dots, 30) \text{where} G_z \text{is a } 2 \; by \; 2 \text{ matrix}$$

$$Q_{xy} = diagonal(15, \dots, 15) \text{where} Q_{xy} \text{is a } 2 \cdot h_p \; by \; 2 \cdot h_p \text{ matrix}$$

$$R_{xy} = diagonal(5, \dots, 5) \text{where} R_{xy} \text{is a } 2 \cdot h_c \; by \; 2 \cdot h_c \text{ matrix}$$

$$G_{xy} = diagonal(20, \dots, 20) \text{where} G_{xy} \text{is a } 4 \; by \; 4 \text{ matrix}$$

The gains of the three PID controllers responsible for the attitude control we tuned manually with the following values:

$$N = 16$$

$$K_{P\Phi} = 1\,, K_{D\Phi} = 19$$

$$K_{P\theta} = 1\,, K_{D\theta} = 19$$

$$K_{P\psi} = 1\,, K_{D\psi} = 18$$

The matrices where tuned manually with trial and error procedure, where the objective was to minimize the sum of square errors between the desired values and the predicted outputs for x y and z parameters.

The reference desired coordinates for the simulation in space are:

$$x_r = 9\ (m), y_r = 8\ (m), z_r = 10(m)$$

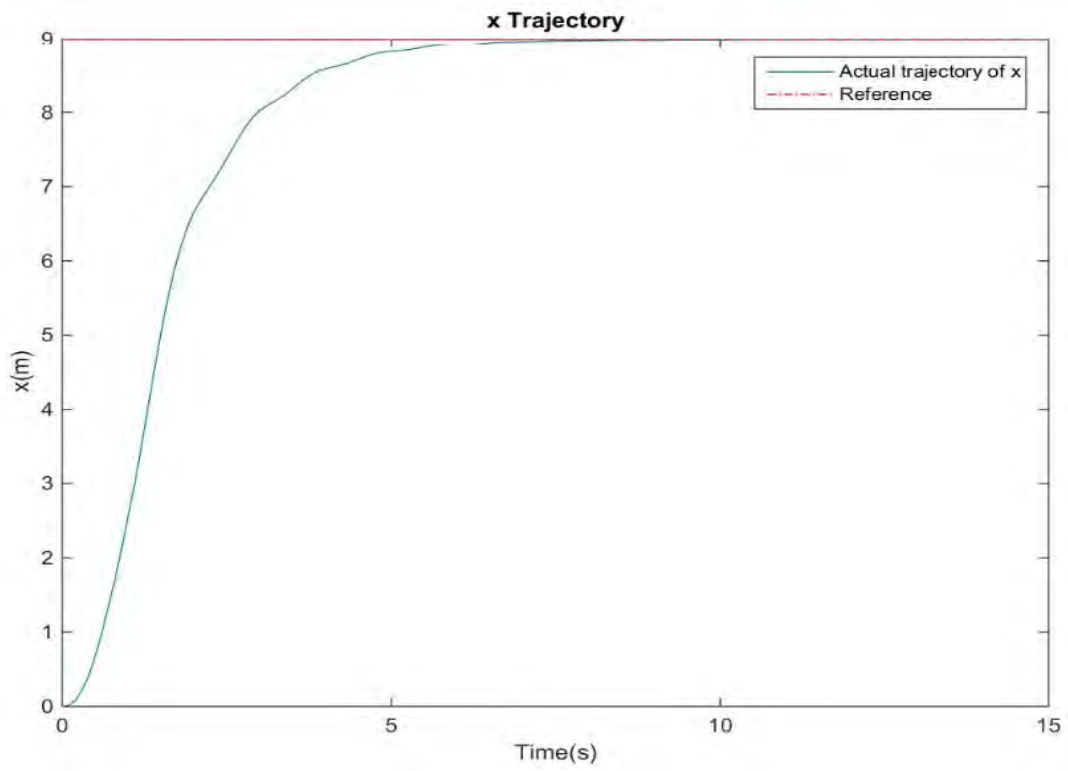


Figure 5.29: *Response of x parameter*

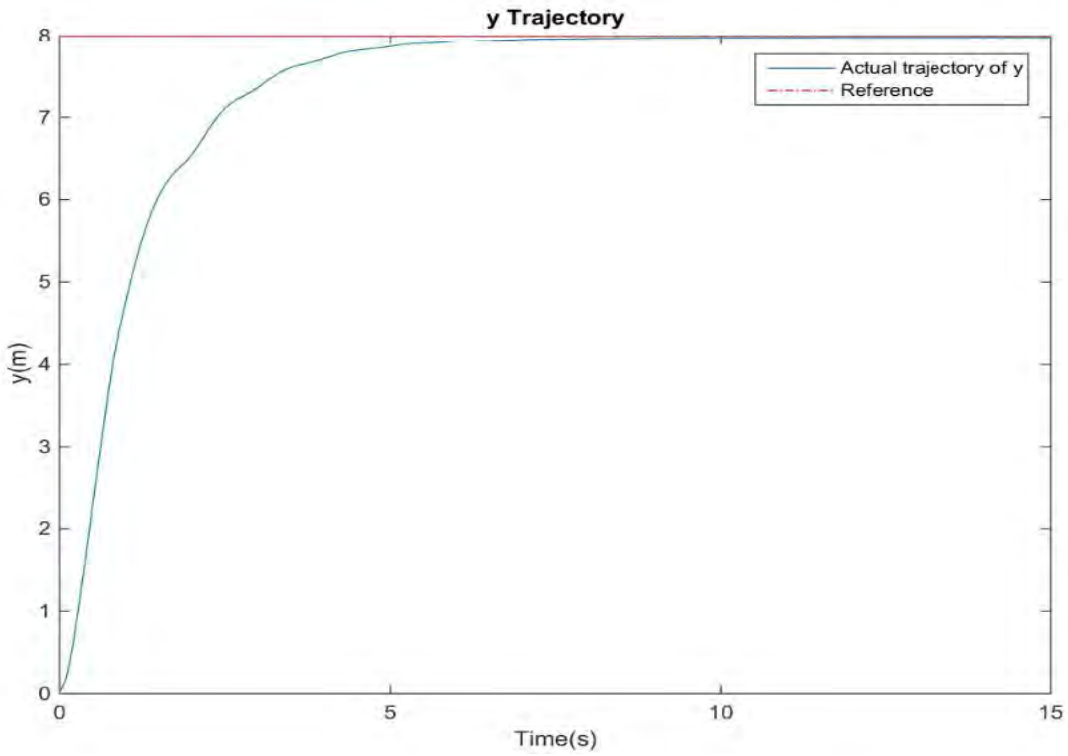


Figure 5.30: *Response of y parameter*

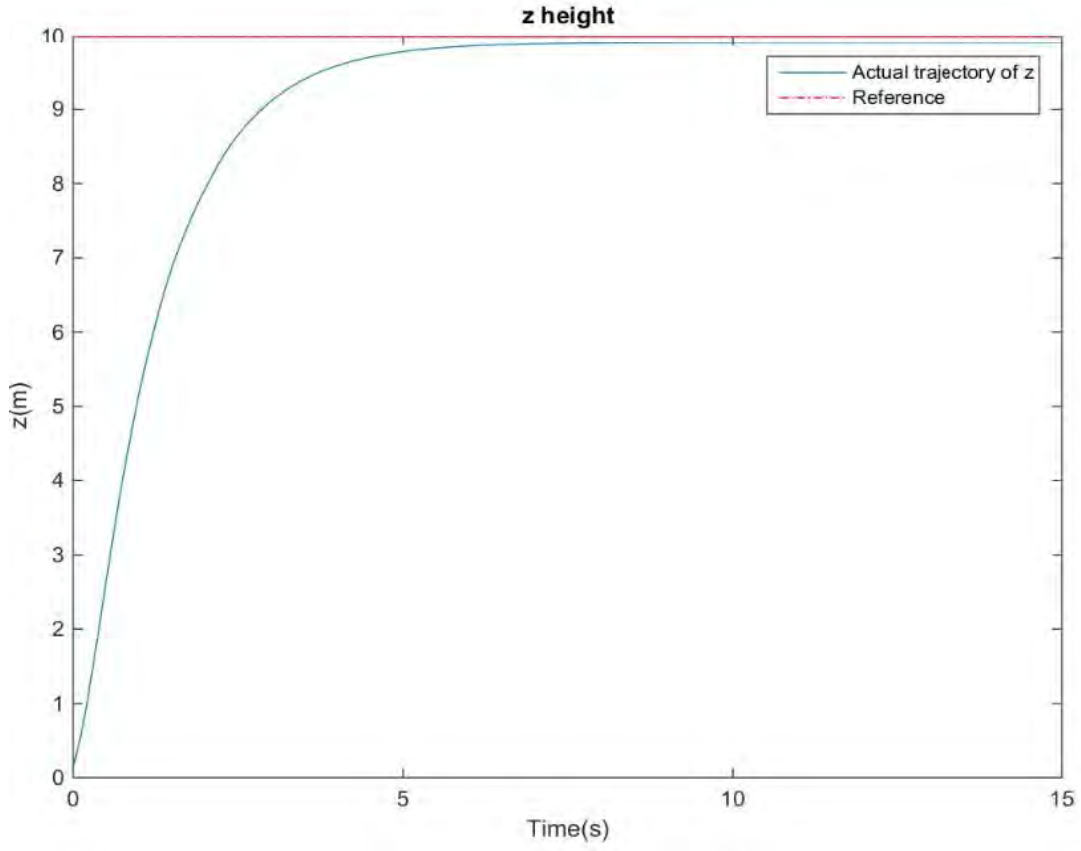


Figure 5.31: *Response of z parameter*

The goal of achieving a certain set of coordinates on 3-D space is managed with success. In the three previous diagrams all the parameters reach the set point value. The responses of all three parameters share some characteristics in common. First, there is no maximum swing above the given value eliminating the overshoot phenomenon. Second, there are no oscillations of the responses eliminating the ringing effect . Finally, the time for reaching the final value ,known as settling time is quite small.

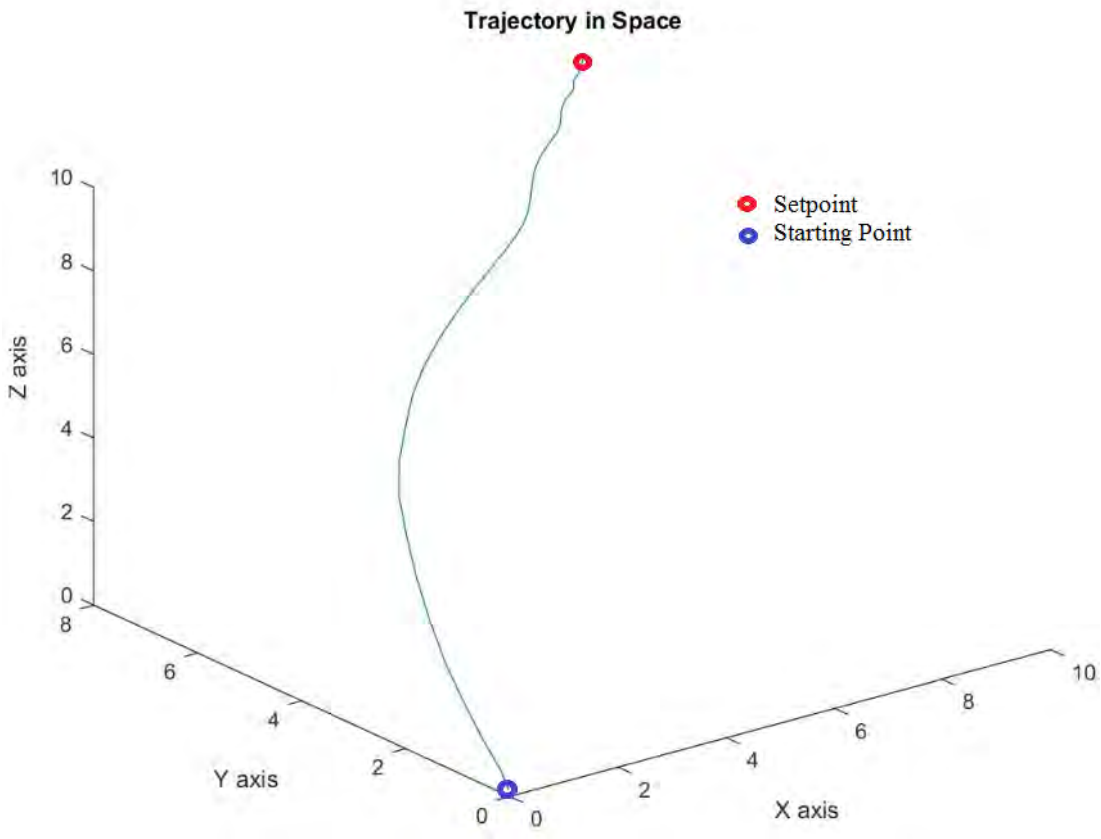


Figure 5.32: *Response for the task of reaching a certain point in space*

5.3.2 Task of tracking a reference trajectory in space

The weight matrices of the Model Predictive Controllers are adjusted in the following way:

$$h_p = 18\,, h_c = 18\,,$$

$$Q_{z1} = diagonl(\ 35\ 5\ ) \text{where} Q_z \text{is a } 2\ by\ 2 \text{ matrix}$$

$$Q_z = diagonal(Q_{z1}, \dots, Q_{z1}) \text{where} Q_z \text{is a } h_p\ by\ h_p \text{ matrix}$$

$$R_z = diagonal(0.01, \dots, 0.01) \text{where} R_z \text{is a } h_c\ by\ h_c \text{ matrix}$$

$$G_z = diagonal(50\ \ 5) \text{where} G_z \text{is a } 2\ by\ 2 \text{matrix}$$

$$Q_{xy} = diagonal(Q_{xy\,1}, \dots, Q_{xy\,1}) \text{where} Q_{xy} \text{is a } 2 \cdot h_p\ by\ 2 \cdot h_p \text{ matrix}$$

$$Q_{xy\,1} = diagonal(\ 40\ 8\ ) \text{where} Q_{xy\,1} \text{is a } 2\ by\ 2 \text{ matrix}$$

$$R_{xy} = diagonal(5, \dots, 5) \text{where} R_{xy} \text{ is a } 2 \cdot h_c\ by\ 2 \cdot h_c \text{ matrix}$$

$$G_{xy} = diagonal(100\ 10\ 100\ 10) \text{where} G_{xy} \text{ is a } 4\ by\ 4 \text{matrix}$$

The gains of the three PID controllers responsible for the attitude control we tuned manually with the following values:

$$N = 16$$

$$K_{P\Phi} = 1\,, K_{D\Phi} = 19$$

$$K_{P\theta} = 1\,, K_{D\theta} = 19$$

$$K_{P\psi} = 1\,, K_{D\psi} = 18$$

5.3.2.1 Task of tracking a reference circle trajectory

The reference trajectory used in the simulation is a circle in the Euclidean space

$$x_r = \frac{1}{2}\cos(\frac{\pi\, t}{20})\ m\ , y_r = \frac{1}{2}\sin(\frac{\pi\, t}{20})\ m\ , z_r = 3 - 2\cos(\frac{\pi\, t}{20})\ m\ \ , \psi_r = 0\ rad$$

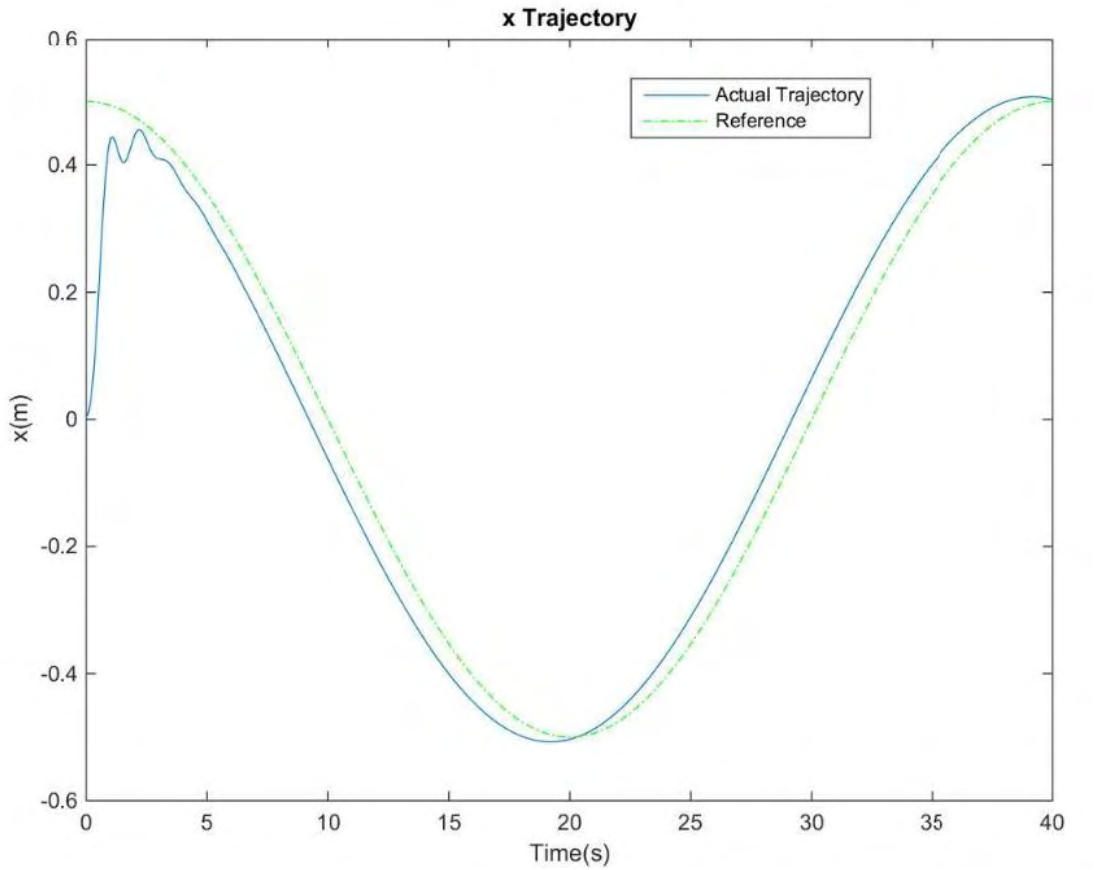


Figure 5.33: *Response of x parameter*

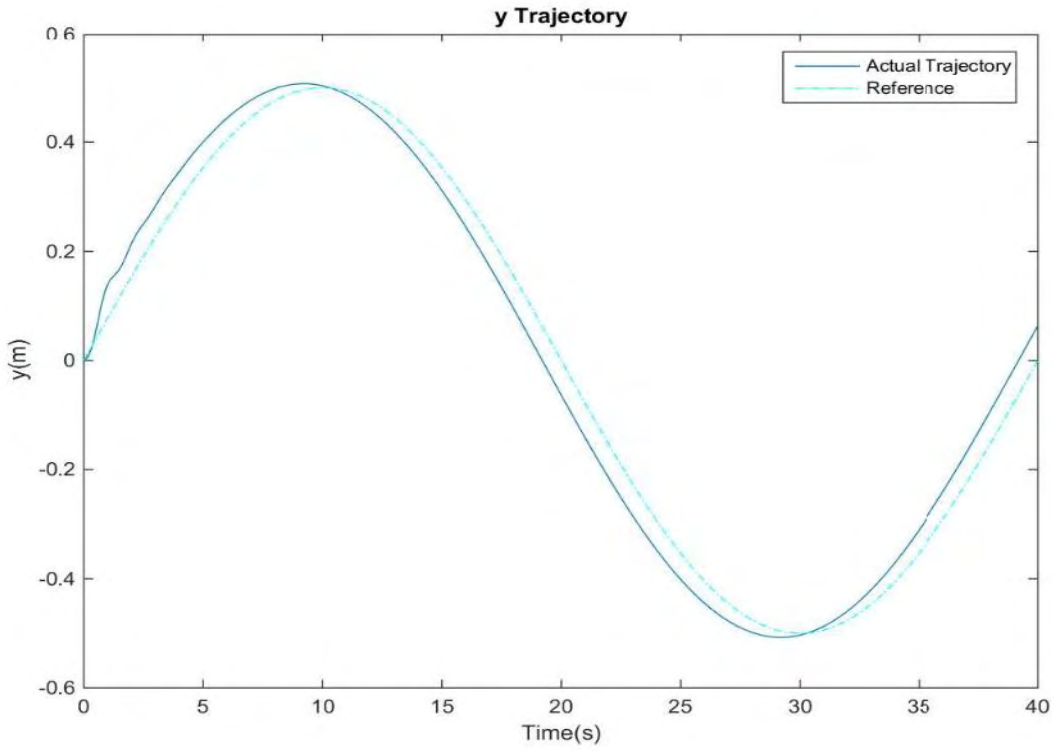


Figure 5.34: *Response of y parameter*

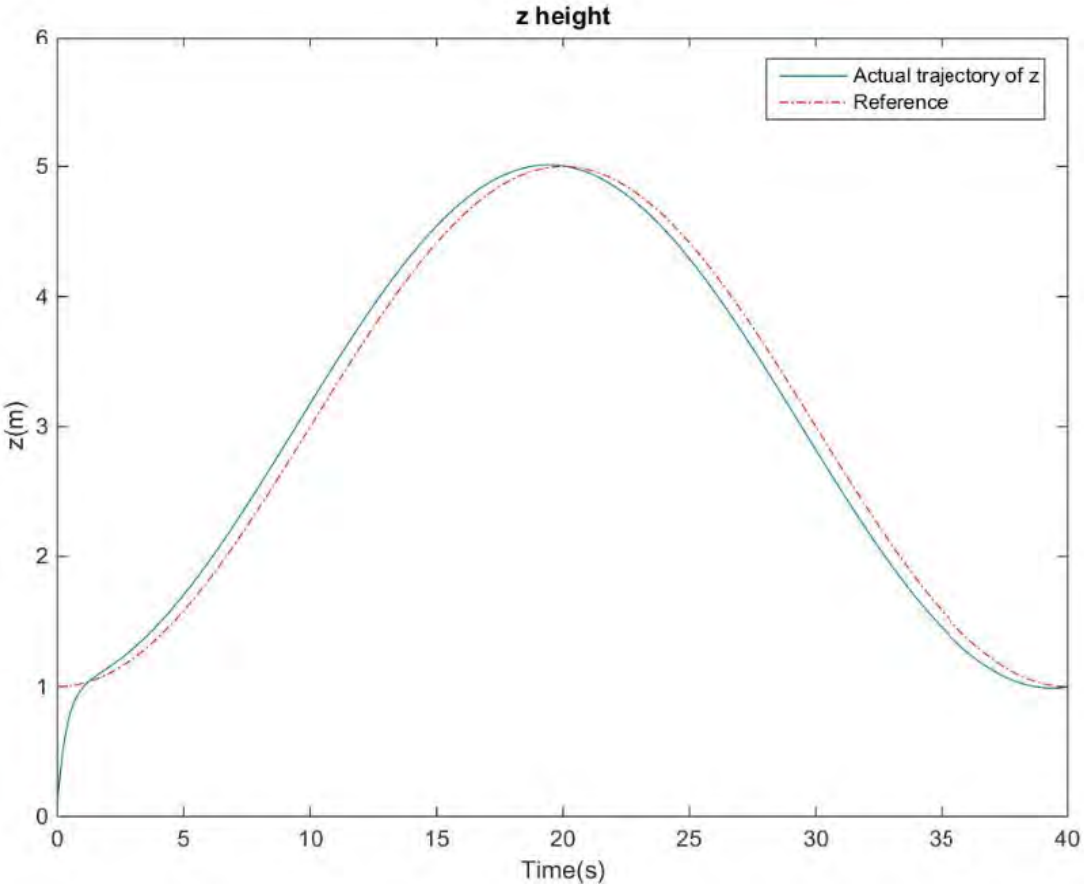


Figure 5.35: *Response of z parameter*

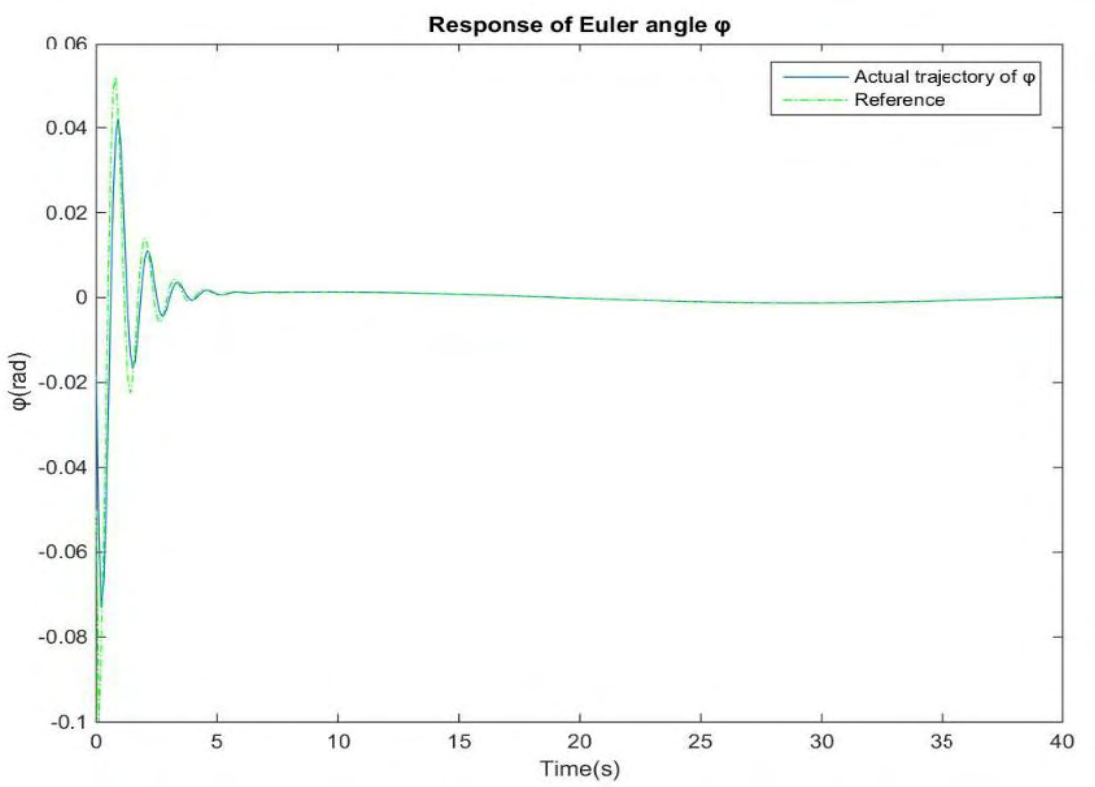


Figure 5.36: *Response of Euler angleφ*

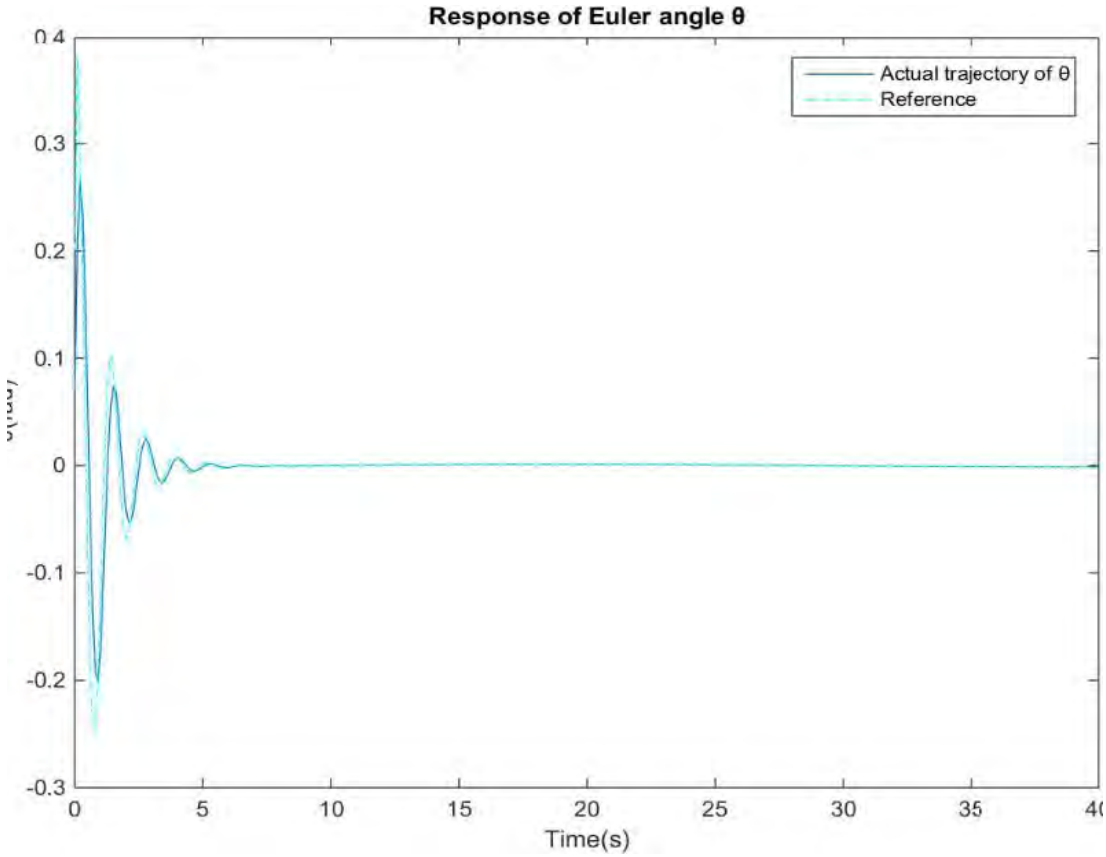


Figure 5.37: *Response of Euler angleθ*

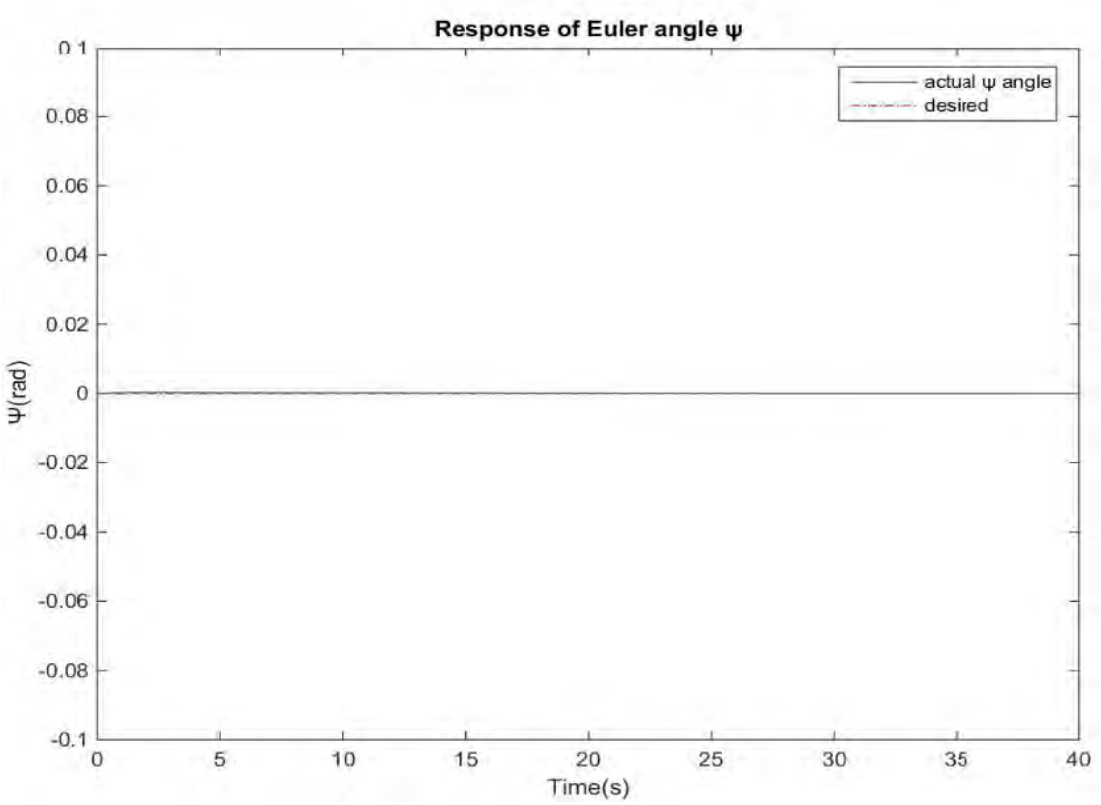


Figure 5.38: *Response of Euler angleψ*

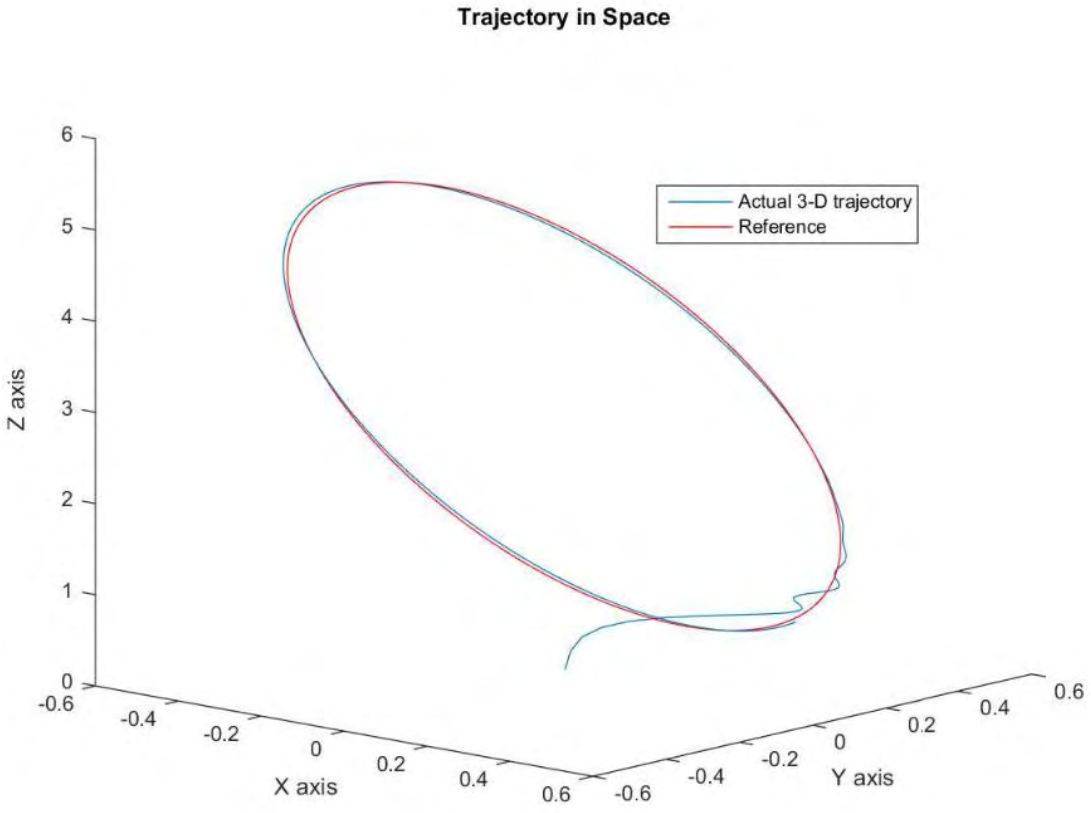


Figure 5.39: *Response for the following a circle Trajectory in Space*

Figures 5.33 ,5.34, 5.35 and 5.38 exhibit the ability of the system to track the reference trajectory given. As the simulation progresses in time the position errors between the actual and the reference values is minimized. After a short time window the trajectory of quadrotor movement is stabilized and it follows smoothly the reference one . From the figures 5.36, 5.37 the efficiency of the PID controllers to track the reference Euler's is shown. Angles φand θ display a ringing effect at the first seconds of their simulation. This is caused until the quadrotor enters the trajectory of the circle. Angle ψremains close to reference value during the simulation. Lastly figure 5.36 display the vehicle' s circle movement in 3 D space.

5.3.2.2 Tracking a reference spiral trajectory

The reference trajectory used in the simulation is a helix in the Euclidean space

$$x_r = \frac{1}{2}\cos\frac{t}{2}\ m\ , y_r = \frac{1}{2}\sin\frac{t}{2}\ m\ , z_r = 1 + \frac{t}{10}\ m\ \ , \psi_r = 0\ rad$$

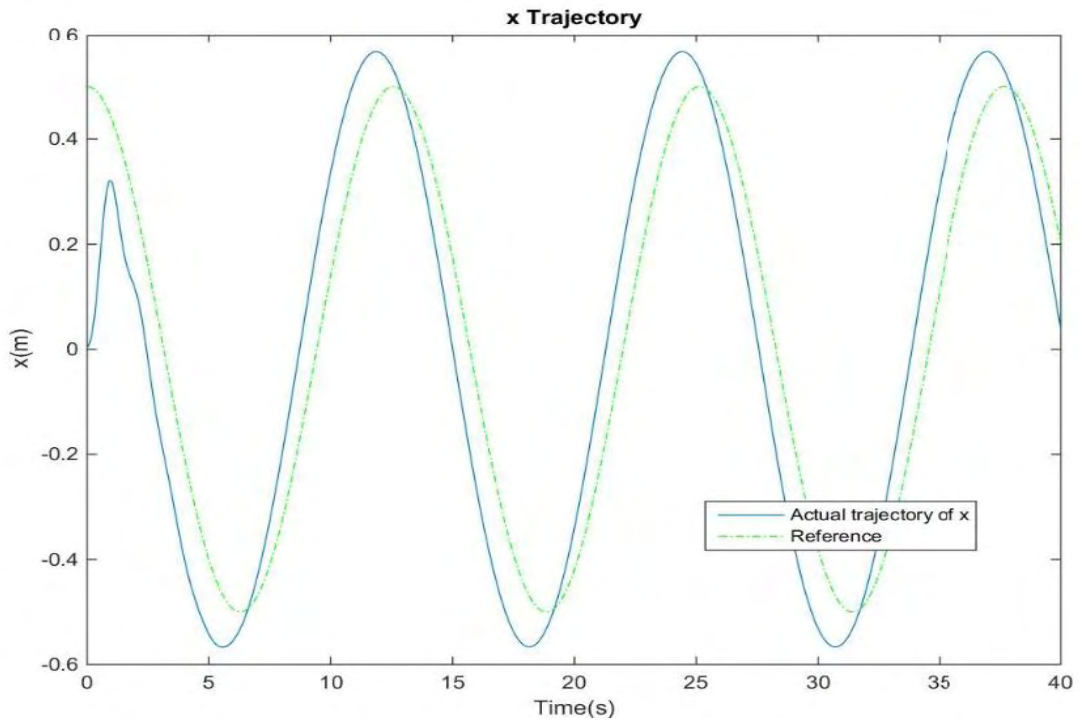


Figure 5.40: *Response of x parameter*

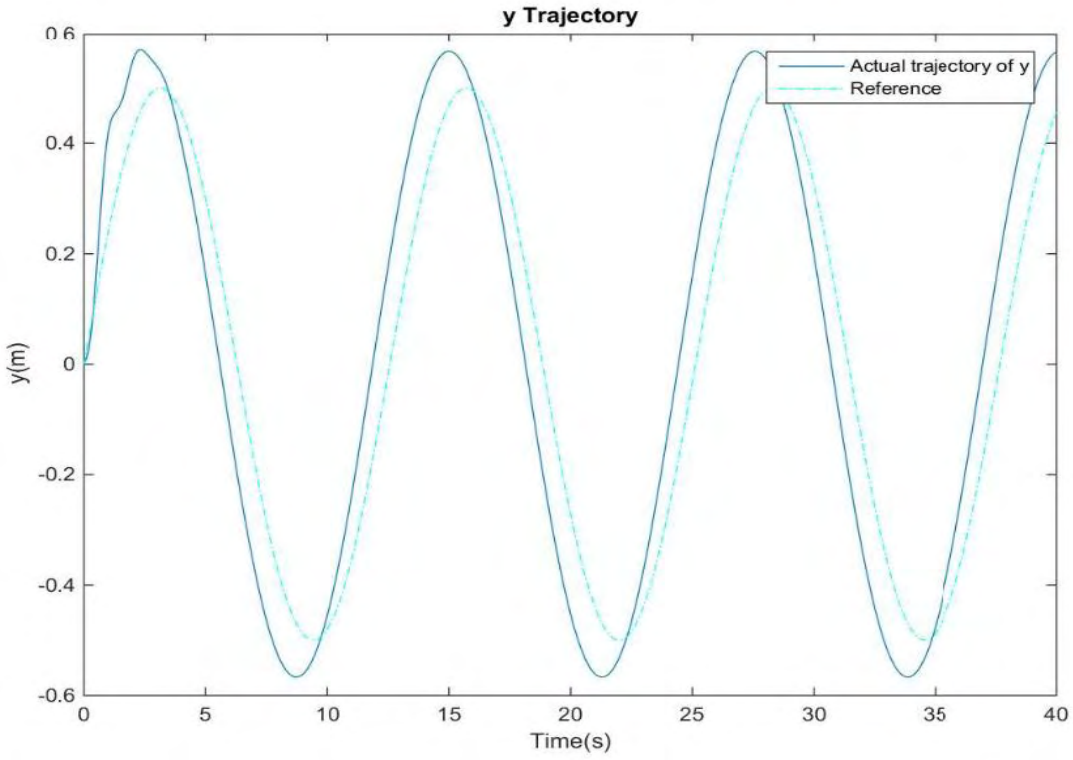


Figure 5.41: *Response of y parameter*

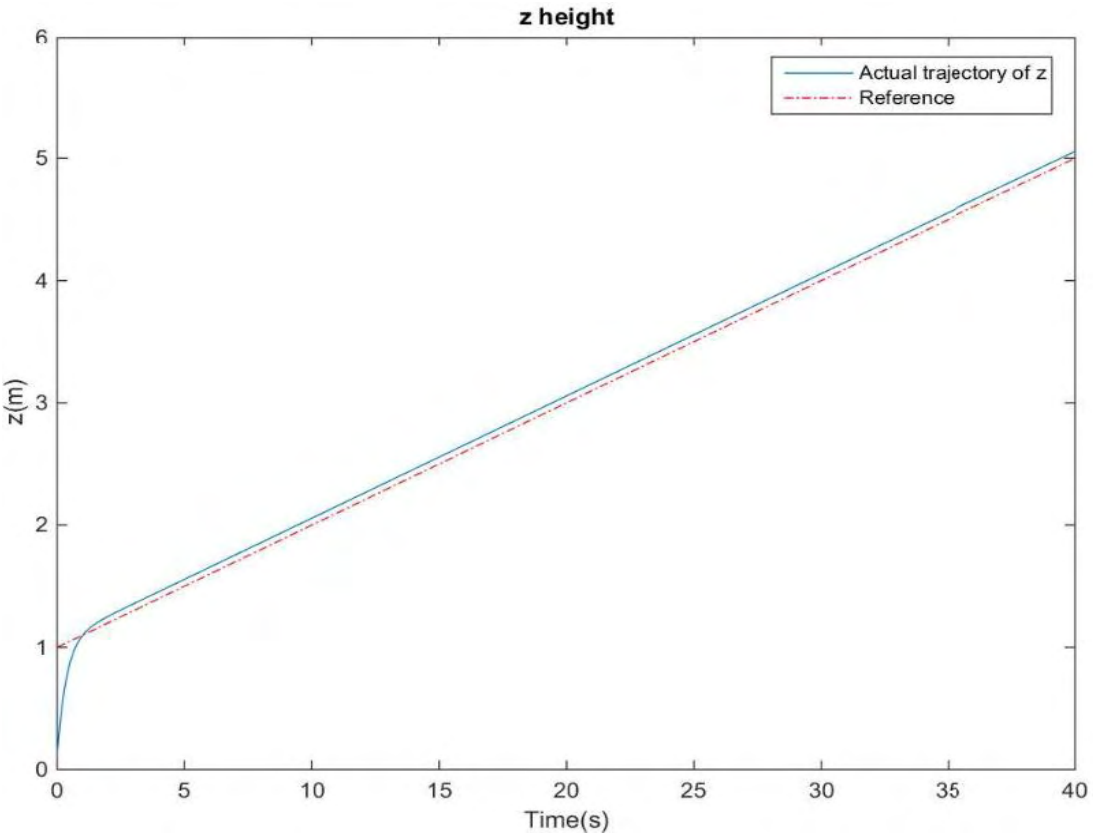


Figure 5.42: *Response of z parameter*

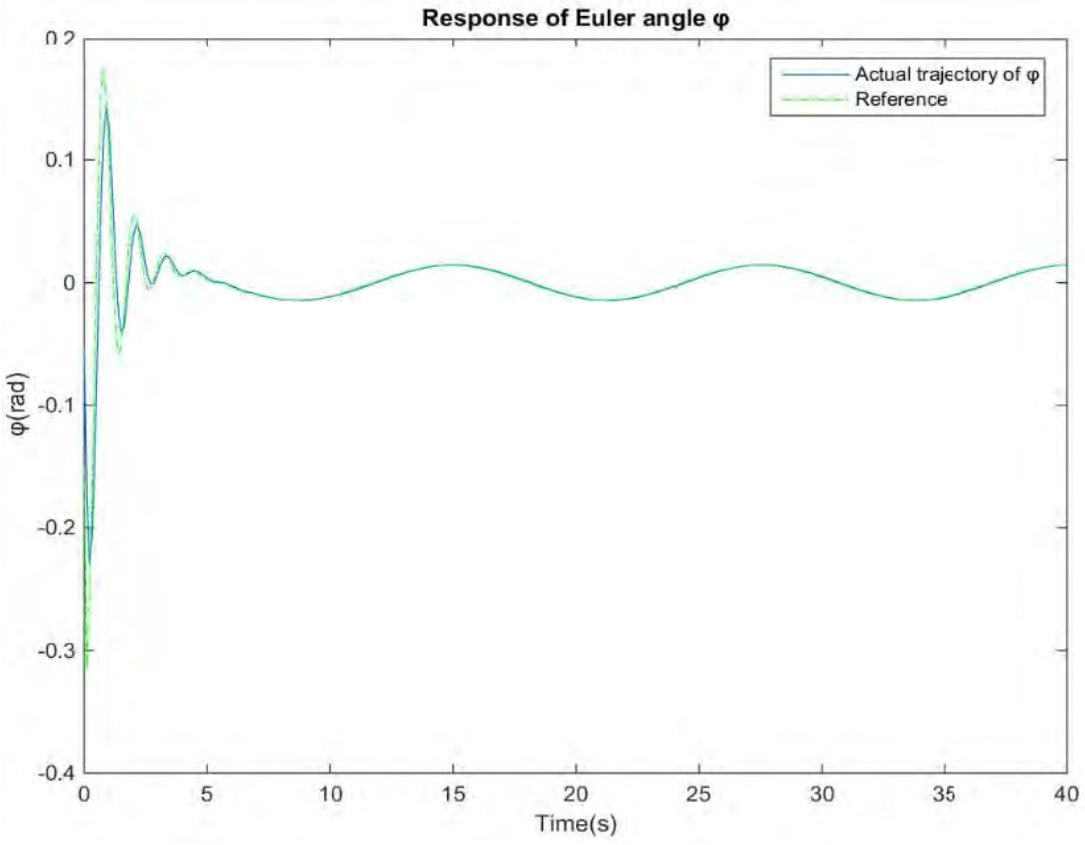


Figure 5.43: *Response of Euler angleφ*

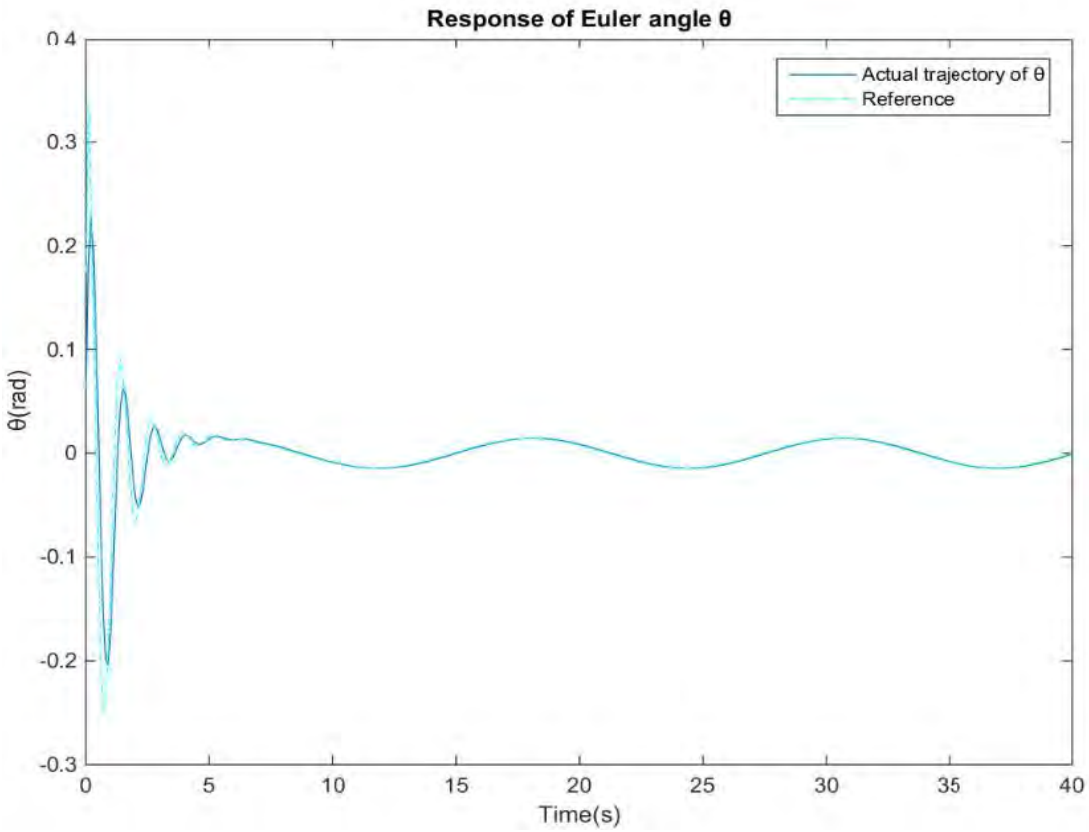


Figure 5.44: *Response of Euler angleθ*

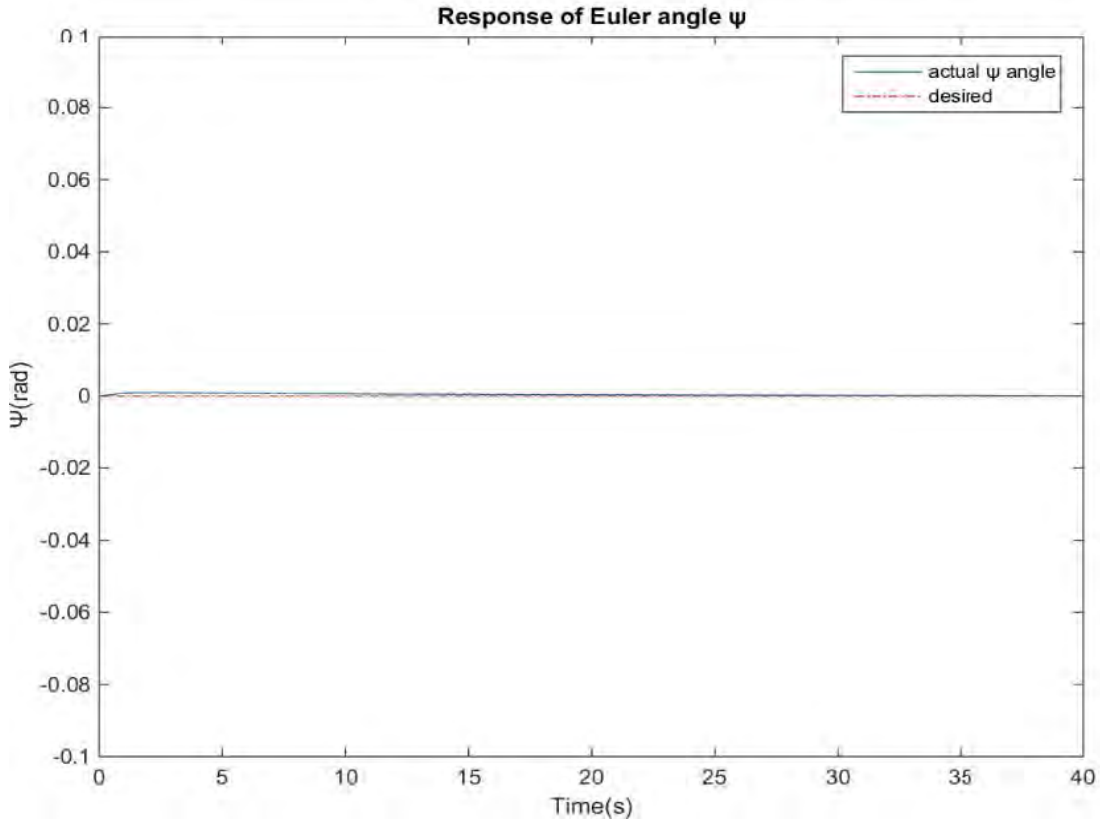


Figure 5.45: *Response of Euler angleψ*

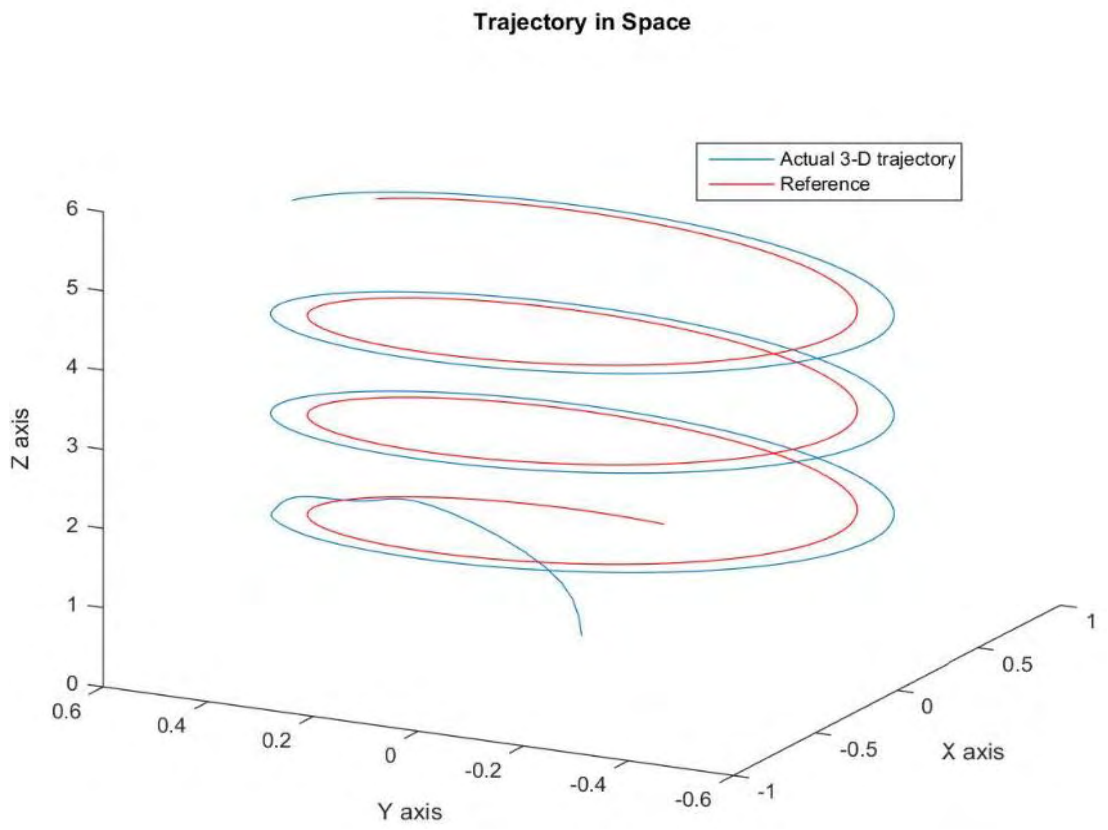


Figure 5.46: *Helix Trajectory in Space*

Figures 5.40, 5.41, 5.42 and 5.45 show the ability of the system 's tracking the reference trajectory given. The predictive controller performs a quick and smooth tracking as it is shown in figures 5.40 ,5.41, 5.42. Figures 5.43, 5.44, 5.45 depict the efficiency of the attitude controllers to track the reference Euler's angles. The figures show that the Euler's angles follow the reference trajectory with ease. Finally figure 5.46 display the vehicle's helix movement in 3 D space.

Chapter 6

Conclusions

The goal of this Thesis was to derive a mathematical model for an unmanned quadrotor, and develop a control algorithm to track a reference trajectory in space and verify the performance of this controller via simulations results.

First, a control strategy was applied to solve the problem of stabilizing the desired states which include quadrotor's altitude and Euler's Angles. This was possible with the help of four PID controllers which were tuned manually. The simulations of the closed loop system show the efficiency of the tuning performed in chapter 5.2.

An important objective of this Thesis was to develop a control scheme that enables the quadrotor to track a reference trajectory. To be more specific, a model predictive control strategy was proposed in order to follow the desired path. Simulation results implemented in Matlab/Simulink verify that the control strategy designed in this Thesis can smoothly and effectively track time-varying quadrotor trajectories.

Tuning the parameters and the constants of both model predictive and PID controllers was a challenging task. In the case of the predictive control, it was shown that 2 different kind of tunings were needed in order to perform with success the tasks of reaching a certain set point in space and tracking a reference trajectory.

Future research plans, include replacing the automatic tuning procedure with optimization methods, which are expected to improve the performance and tracking ability of the reference path. This is possible using meta-heuristic search methods as particle swarm optimization (PSO). Another interesting direction is to integrate in the MPC scheme nonlinear modeling techniques, e.g. neural networks.

References

1. Kadri, H.S.K.M.B., *Attitude and altitude control of quadrotor by discrete PID control and non-linear model predictive control*, in *International Conference on Information and Communication Technologies (ICICT)*. 2015, IEEE: Karachi, Pakistan.
2. Chang, C.L.J.P.Y., *PID and LQR trajectory tracking control for an unmanned quadrotor helicopter: Experimental studies*, in *PID and LQR Trajectory Tracking Control for An Unmanned Quadrotor Helicopter: Experimental Studies*. 2016, IEEE: Chengdu, China.
3. Ling, G.B.L.X.Z.H.W., *Quadrotor helicopter Attitude Control using cascade PID* in *Chinese Control and Decision Conference (CCDC)*. 2016, IEEE: Yinchuan, China.
4. Zhao, Y.F.Y.C.Y., *Design of the nonlinear controller for a quadrotor trajectory tracking*, in *29th Chinese Control And Decision Conference (CCDC)*. 2017, IEEE: Chongqing, China.
5. R.Rubio, G.V.R.G.O., *Path Tracking of a UAV via an Underactuated Control Strategy.* European Journal of Control, 2011. **17**(2): p. 194-213.
6. Young-Cheol Choi , H.-S.A., *Nonlinear Control of Quadrotor for Point Tracking: Actual Implementation and Experimental Tests.* IEEE, 2014. **20**(3): p. 1179-1192.
7. Hassam, D.M.O.G.F.A.A., *Quadrotor Position and Attitude Control via Backstepping Approach* in *8th International Conference on Modelling, Identification and Control (ICMIC)*. 2016, IEEE: Algiers, Algeria.
8. ZeFang He, L.Z., *Internal Model Control /Backstepping Sliding Model Control for Quadrotor Trajectory Tracking* in *2nd Information Technology, Networking, Electronic and Automation Control Conference (ITNEC)*. 2017, IEEE: Chengdu, China.
9. Tzes, K.A.G.N.A., *Model Predictive Control Scheme for the Autonomous Flight of an Unmanned Quadrotor*, in *International Symposium on Industrial Electronics*. 2011, IEEE: Gdansk, Poland.
10. Saadia, A.G.M.H.N., *A Robust Adaptive Nonlinear Control Design for Quadrotor* in *8th International Conference on Modelling, Identification and Control (ICMIC)*. 2016, IEEE: Algiers, Algeria.
11. Yu, W.W.X., *Chattering free and nonsingular terminal sliding mode control for attitude tracking of a quadrotor*, in *Chinese Control And Decision Conference (CCDC)*. 2017, IEEE: Chongqing, China.
12. Huaman-Loayza, A.S., *Path-Following of a Quadrotor Using Fuzzy Sliding Mode Control*, in *XXV International Conference on Electronics, Electrical Engineering and Computing (INTERCON)*. 2018, IEEE: Lima, Peru.
13. Voos, H., *Nonlinear control of a quadrotor micro-uav using feedback-linearization*, in *Int. Conf. Mechatronics*. 2009, IEEE.
14. Lincheng, Z.S.A.H.Z.D.S., *A new feedback linearization LQR control for attitude of quadrotor*, in *13th International Conference on Control Automation Robotics & Vision (ICARCV)*. 2014, IEEE: Singapore, Singapore.
15. Dhillon, A.J.M.C.E.Y.B.S., *Feedback linearization approach to fault tolerance for a micro quadrotor*, in *International Conference on Industrial Technology (ICIT)*. 2018, IEEE: Lyon, France.
16. Lee, T.L.W.R.R.K.D., *Model linearization and H∞, controller design for a quadrotor unmanned air vehicle: Simulation study*, in *13th International Conference on Control Automation Robotics & Vision (ICARCV)*. 2014, IEEE: Singapore, Singapore.
17. Li, L.Z.J.W.S.L.C.W.A., *Adaptive backstepping sliding mode controller design based on "H" type quadrotor*, in *36th Chinese Control Conference (CCC)*. 2017: Dalian, China.

18. Dailiang Ma, Y.X., Tianya Li, Kai Chang, *Active Disturbance Rejection and Predictive Control Strategy for a Quadrotor Helicopter.* IET Control Theory & Applications, 2016. **10**(17): p. 2213-2222.
19. Dharmana, G.G.M.M., *MPC controller for trajectory tracking control of quadcopter*, in *International Conference on Circuit ,Power and Computing Technologies (ICCPCT)*. 2017, IEEE: Kollam, India.
20. Yang, H.C.Y., *Model predictive control and PID for path following of an unmanned quadrotor helicopter*, in *12th IEEE Conference on Industrial Electronics and Applications (ICIEA)*. 2017: Siem Reap, Cambodia.
21. Schoellig, M.G.A.P., *Flatness-Based Model Predictive Control for Quadrotor Trajectory Tracking*, in *International Conference on Intelligent Robots and Systems (IROS)*. 2018, IEEE: Madrid, Spain.