Politecnico di Torino

Faculty of Engineering
Master's Degree in Aerospace Engineering

Master's Degree Thesis

GPS-based trajectory tracking technique for an unmanned hexacopter in fire-detection missions

Thesis Advisor:
Professor Manuela Battipede

Candidate:
Luca Lorusso

19th September, 2018

GPS-based trajectory tracking technique for an unmanned hexacopter in fire-detection missions

Luca Lorusso

Acknowledgements

Desidero ringraziare in primis la Prof. Battipede, che si è dimostrata sempre disponibile e gentile. La sua passione per la materia, l'entusiasmo e la fiducia che mi sono stati trasmessi hanno reso questo lavoro un percorso avvincente e interessante.

Ringrazio tutti i miei compagni di corso che ho conosciuto, in particolare Marco P., Marco L., Elio, Felix e Luca, con cui ho legato di più; ci siamo aiutati a vicenda e insieme siamo riusciti a superare gli ostacoli, senza mai arrenderci. Un ringraziamento speciale a chi mi è stato maggiormente vicino, dandomi sostegno e aiuto e spronandomi a fare sempre meglio. Ringrazio anche Fausto, amico di una vita, e Simone e Andrea, amici sin dall'infanzia.

Ringrazio in particolar modo la mia fidanzata Orsola, che è stata fondamentale in tutto questo percorso. Mi ha sempre dato grinta, sostegno ed una solida spalla su cui contare ogni giorno. Nei momenti difficili c'è sempre stata, facendo in modo che tutto andasse per il verso giusto. Grazie anche ai suoi genitori Flora e Giuseppe e a suo fratello Antonio che hanno sempre creduto in me.

Ringrazio i miei nonni Tommaso, Lina, Pietro e Pina, che mi hanno cresciuto come se fossi stato un figlio per loro. La loro saggezza mi ha trasmesso principi e valori sani ed i loro insegnamenti sono stati indispensabili per affrontare al meglio la vita. Ringrazio tutti i miei zii e cugini, senza eccezioni, che hanno sempre contato su di me.

Infine, ma non da ultimi, ringrazio i miei genitori Maria Rosa e Gianfranco, le persone più importanti della mia vita, che più di tutti hanno saputo guidarmi, da quando ero piccolo, verso la strada giusta. Presenti in ogni momento, la loro parola è stata fondamentale per ogni mia scelta di vita. Nel percorso universitario, la loro fiducia nei miei confronti è stata fondamentale per permettere il raggiungimento di questo primo traguardo.

Contents

List of Figures

Abstract

This master thesis focuses on modeling, control and trajectory tracking of a hexacopter UAV (Unmanned Aerial Vehicle) used in fire-detection missions. A set of non-linear dynamic equations describing the motion of the hexacopter were derived. These equations were then implemented in Matlab/Simulink, which became a good simulation environment for further studies. A PID controller was successfully implemented in simulation; a linear model was used to tune the parameters of the inner PID controllers.

Chapter 1

Introduction

The use and development of small unmanned aerial vehicles (UAVs) has increased significantly over the last few decades. UAVs can be used in several field and applications. People in possession of large grounds, such as farmers and forest owners, can use UAVs to remotely inspect their properties. Being able to replace human pilots with autonomous aerial vehicles is obviously a tremendous advantage. The goal of this master thesis is modeling, control and trajectory tracking of a hexacopter UAV (Unmanned Aerial Vehicle) used in fire-detection missions. A set of non-linear dynamic equations describing the motion of the hexacopter were derived. These equations were then implemented in Matlab/Simulink, which became a good simulation environment for further studies. A PID controller was successfully implemented in simulation; a linear model was used to tune the parameters of the inner PID controllers.

1.1 System modeling

The trajectory tracking system is based on a 4-dimension model, which needs the desired position (x, y, z) and the time between every waypoint. Throughout trajectory generation, the simulator prefers linear trajectories between one waypoint and another, but it takes into account the presence of potential No-Fly Zones; in addition, it chooses paths which guarantee a good GPS signal.

Once the desired position has been calculated, the control system generates the desired velocities in order to reach that point, by the given time. In the lateral-directional plane, the controller also generates the desired pitch and roll angles (considering operating limits). The controller outputs are used to calculate forces and moments (one vertical force and three moments around the three axes); those ones are the input commands that the model needs to compute the actual position, speed and orientation. At the end of the simulation, it is possible to assess the results with plots and animations of the flying drone.

Chapter 2

Model

2.1 Reference frame

Let $E = \{x_E, y_E, z_E\}$ be an inertial frame with origin on the surface of the earth. This earth fixed frame is a north-east-down-system (NED-system) with x_E pointing to the north, y_E pointing to the east and z_E pointing downwards.

Also, introduce $B = \{x_B, y_B, z_B\}$ as a body fixed frame with origin at the hexarotor's center of gravity. x_B is equivalent with the forward direction of the hexarotor, y_B with right and z_B with down. This choice of body fixed frame has the advantage that the inertia tensor is time-invariant and that the body symmetry will simplify the equations.

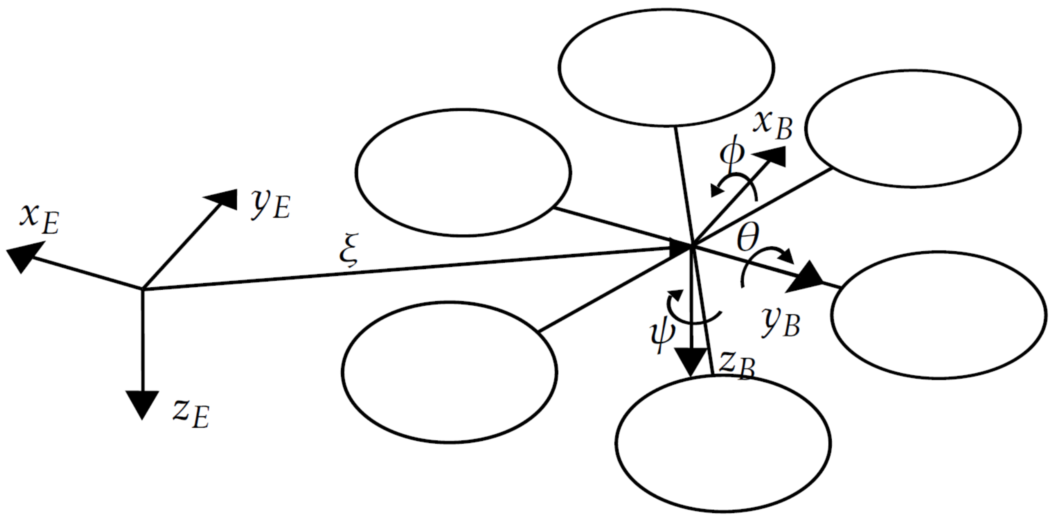

Figure 2.1: The two frames of reference and their relation

2.2 Kinematics

The body fixed frame's position in the earth fixed frame can be described by the vector $\boldsymbol{\xi} = [x,\, y,\, z]^T$ and its orientation, attitude and heading, by the vector $\boldsymbol{\eta} = [\phi,\, \theta,\, \psi]^T$. The angles used to represent the orientation are defined using Tait-Bryan formalism. This type of formalism differs from proper Euler angles by using three different axes when forming the rotation. Within aerospace literature this representation is often referred to as Euler angles, which may cause some confusion.

The adopted order of rotation is commonly used when describing aircraft motion. To bring the body fixed frame into coincidence with the earth fixed frame the following rotations are considered:

- First, rotate the body-fixed frame about the x_B-axis by the roll angle ϕ, resulting in a new frame of reference called B^1.
- Then, rotate the new frame B^1 about the new axis y_B^1 by the pitch angle θ, resulting in a new frame of reference called B^2.
- Lastly, rotate the new frame B^2 about the new axis z_B^2 (which coincide with z_E) by the yaw angle ψ, resulting in a new frame aligned with the earth fixed frame.

In order to transform any linear quantity from earth frame to body frame rotation matrices are used. In order to simplify the notation $\sin(\cdot)$ and $\cos(\cdot)$ are abbreviated $s\cdot$ and $c\cdot$ respectively. The relation between B and B^1 after the rolling can be described by

$$\begin{bmatrix} x_B \\ y_B \\ z_B \end{bmatrix} = \begin{bmatrix} 1 & 0 & 0 \\ 0 & c\phi & s\phi \\ 0 & -s\phi & c\phi \end{bmatrix} \begin{bmatrix} x_B^1 \\ y_B^1 \\ z_B^1 \end{bmatrix} \tag{2.1}$$

where the rotation matrix is denoted $R(x, \phi)$. In similar manner, after the pitching B^1 is related to B^2 via

$$\begin{bmatrix} x_B^1 \\ y_B^1 \\ z_B^1 \end{bmatrix} = \begin{bmatrix} c\theta & 0 & s\theta \\ 0 & 1 & 0 \\ -s\theta & 0 & c\theta \end{bmatrix} \begin{bmatrix} x_B^2 \\ y_B^2 \\ z_B^2 \end{bmatrix} \tag{2.2}$$

with the rotation matrix $R(y, \theta)$. After the final yawing B^2 and E are related by

$$\begin{bmatrix} x_B^2 \\ y_B^2 \\ z_B^2 \end{bmatrix} = \begin{bmatrix} c\psi & s\psi & 0 \\ -s\psi & c\psi & 0 \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} x_E \\ y_E \\ z_E \end{bmatrix} \tag{2.3}$$

using the rotation matrix $R(z,\psi)$. The total rotational matrix, to transform any quantity from earth frame to body frame, is obtained by multiplying $R(x,\phi)$, $R(y,\theta)$ and $R(z,\psi)$. That yields

$$\begin{bmatrix} x_B \\ y_B \\ z_B \end{bmatrix} = \begin{bmatrix} c\theta c\psi & c\theta s\psi & -s\theta \\ s\phi s\theta c\psi - c\phi s\psi & s\phi s\theta s\psi + c\phi c\psi & s\phi c\theta \\ c\phi s\theta c\psi + s\phi s\psi & c\phi s\theta s\psi - s\phi c\psi & c\phi c\theta \end{bmatrix} \begin{bmatrix} x_E \\ y_E \\ z_E \end{bmatrix} \tag{2.4}$$

and the total rotational matrix is denoted R_L. This matrix is sometimes referred to as the direction cosine matrix. One convenient feature of R_L is that its inverse is equal to the transpose, that is $R_L^{-1} = R_L^T$, since $R_L \in SO(3)$. This is useful when transforming quantities in the body fixed frame to the earth fixed frame.

Denote the linear velocity of the hexarotor expressed in the body fixed frame $V = [u,\, v,\, w]^T$. Then the second time derivative of the position is

$$\dot{\boldsymbol{\xi}} = R_L^T \boldsymbol{V} \tag{2.5}$$

In order to relate the change of attitude with the body angular velocities the different steps of the rotation have to be considered. First, denote the body angular velocities $\omega = [p,\, q,\, r]^T$, where p is rotation around the x_B-axis, q is rotation around the y_B-axis and r is rotation around the z_B-axis.

The first rotation applied to the earth fixed frame, the yawing, is subject to three successive angular transformation: rotation around z_E, y_B^1 and x_B. The second rotation, the pitching, is subject to two successive angular transformation: rotation around y_B^1 and x_B. Lastly, the rolling is only subject to one attitude transformation: rotation around x_B. That gives the relation

$$\begin{aligned} \begin{bmatrix} p \\ q \\ r \end{bmatrix} &= R(x,\phi)R(y,\theta)R(z,\psi) \begin{bmatrix} 0 \\ 0 \\ \dot{\psi} \end{bmatrix} + R(x,\phi)R(y,\theta) \begin{bmatrix} 0 \\ \dot{\theta} \\ 0 \end{bmatrix} + R(x,\phi) \begin{bmatrix} \dot{\phi} \\ 0 \\ 0 \end{bmatrix} = \\ &= \begin{bmatrix} 1 & 0 & -s\theta \\ 0 & c\phi & s\phi c\theta \\ 0 & -s\phi & c\phi c\theta \end{bmatrix} \begin{bmatrix} \dot{\phi} \\ \dot{\theta} \\ \dot{\psi} \end{bmatrix} \end{aligned} \tag{2.6}$$

where the final transformation matrix is denoted R_A. Then the time derivative of the hexarotor's attitude $\boldsymbol{\eta}$ is

$$\dot{\boldsymbol{\eta}} = R_A^{-1} \omega \tag{2.7}$$

and with $t\cdot$ short for $\tan(\cdot)$ the inverse can be written

$$R_A^{-1} = \begin{bmatrix} 1 & s\phi t\theta & c\phi t\theta \\ 0 & c\phi & -s\phi \\ 0 & \frac{s\phi}{c\theta} & \frac{c\phi}{c\theta} \end{bmatrix} \tag{2.8}$$

2.3 Dynamics

To begin with, the general equations of linear and angular motion of a rigid body must be derived. To do this, Newton-Euler formalism is used. In order to distinguish between quantities expressed in different frames of reference, the following notation is used:

- A quantity expressed in the inertial earth frame is denoted X^E
- A quantity expressed in the body fixed frame is denoted X^B
- A quantity already defined in a frame of reference, e.g. $\boldsymbol{\omega}$, will not have the notation above.

2.3.1 Translation dynamics

In the earth fixed inertial frame, Newton's second law can be applied, giving

$$\boldsymbol{F}^E = m\ddot{\boldsymbol{\xi}} = m\boldsymbol{a}^E \tag{2.9}$$

where the time derivative is with respect to the inertial frame and $\boldsymbol{a}^E$ is the acceleration of the body fixed frame expressed in the inertial frame, m is the mass of the body which is constant and $\boldsymbol{F}^E$ is the sum of all external forces applied to the body expressed in the inertial frame. However, it would be convenient to express the dynamics in the body fixed frame. By using the rotational matrix from equation (2.5) a change of basis can be obtained and the expression becomes

$$R_L\boldsymbol{F}^B = mR_L\boldsymbol{a}^B = mR_L\left(\frac{d\boldsymbol{V}}{dt}\right)_E \tag{2.10}$$

where the time derivative is still with respect to the inertial frame and $\boldsymbol{V}$ is the velocity of the body fixed frame expressed in the body fixed frame. Computing that derivative is rather cumbersome, but using a well-known relation, sometimes called the transport theorem, yields

$$\boldsymbol{F}^B = m\left(\left(\frac{d\boldsymbol{V}}{dt}\right)_B + \boldsymbol{\omega}\times\boldsymbol{V}\right) \tag{2.11}$$

Let the time derivative with respect to the body fixed frame be denoted by a dot, then the final equation of the translational dynamics expressed in the body fixed frame is

$$\boldsymbol{F}^B = m\dot{\boldsymbol{V}} + \boldsymbol{\omega}\times m\boldsymbol{V} \tag{2.12}$$

2.3.2 Rotational dynamics

Again starting in the inertial frame, Euler's second axiom is

$$\boldsymbol{M}^E = \dot{\boldsymbol{L}}^E \tag{2.13}$$

where the time derivative is with respect to the inertial frame, $\boldsymbol{L}^E$ is the angular momentum of the body expressed in the earth frame and $\boldsymbol{M}^E$ is the sum of all external torques applied to the body expressed in the earth frame. The quantities can be expressed in the body fixed frame by applying the rotational matrix for angular velocities from equation (2.7), giving

$$R_A \boldsymbol{M}^B = R_A \dot{\boldsymbol{L}}^B = R_A \left(\frac{dJ\boldsymbol{\omega}}{dt} \right)_E \tag{2.14}$$

where $\boldsymbol{L} = J\boldsymbol{\omega}$ and J is the inertia matrix expressed in the body frame. Differentiating J with respect to the earth frame is rather difficult since it will be time-dependent. Instead, using the transport theorem again, the expression becomes

$$\boldsymbol{M}^B = \left(\frac{dJ\boldsymbol{\omega}}{dt} \right)_B + \boldsymbol{\omega} \times J\boldsymbol{\omega} \tag{2.15}$$

Since the choice of body fixed frame ensured that the inertia matrix is time invariant, the final equation is

$$\boldsymbol{M}^B = J\dot{\boldsymbol{\omega}} + \boldsymbol{\omega} \times J\boldsymbol{\omega} \tag{2.16}$$

2.4 Applied forces and torques

The relations in equation (2.12) and (2.16) are general equations of motion for a rigid body. Now, it is time to apply them to the hexarotor system by finding the different components of the external forces $\boldsymbol{F}$ and torques $\boldsymbol{M}$.

The forces and torques acting on the hexacopter are gravity, air friction aerodynamic forces and torques produced by the propellers and the gyroscopic effects from the rotation of the propellers. The torque caused by the angular acceleration of the propeller has been neglected. All quantities are expressed in the body fixed frame unless anything else is stated.

2.4.1 Gravity

First off is the gravity which is the only force or torque which is naturally expressed in the earth frame. The gravitational force is acting on the hexacopter's center of gravity according to Euler's first axiom and is directed along the z_E-axis. In the body fixed frame the contribution of the gravitational force F_G^B is

$$F_G^B = R_L \begin{bmatrix} 0 \\ 0 \\ mg \end{bmatrix} = mg \begin{bmatrix} s\theta \\ s\phi c\theta \\ c\phi c\theta \end{bmatrix} \tag{2.17}$$

where g is the acceleration due to gravity.

2.4.2 Thrust and torque from propellers

The second contribution is the thrust produced by the aerodynamics of the propellers and reaction torque from the rotation of the rotors. Figure 2.2 shows where the forces are applied on the air frame.

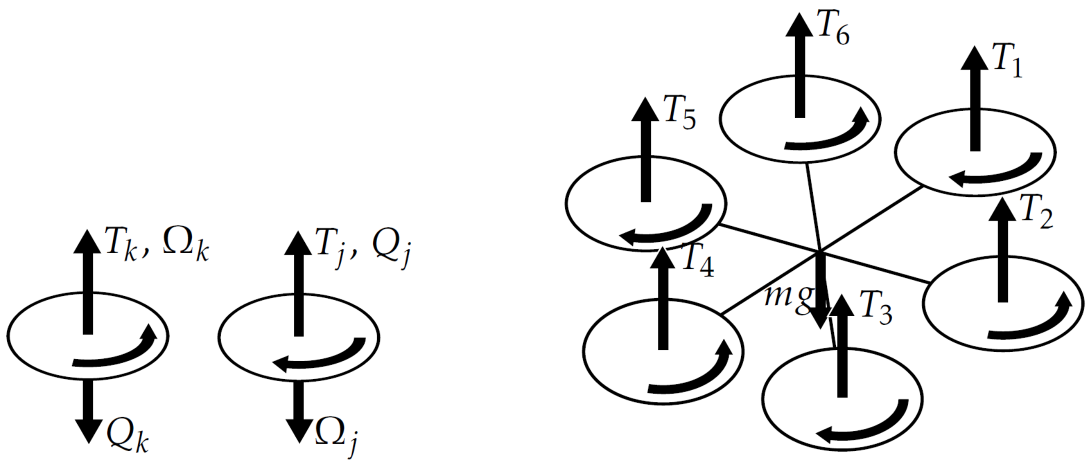

Figure 2.2: Forces and torques produced by rotors $k = 2, 4, 6$ and $j = 1, 3, 5$

The sum of the thrust from the different propellers is total lift force and is always directed along the negative z_B-axis. If the thrust from propeller i is denoted T_i, then the lift force F_w is

$$F_w = \sum_{i=1}^{6} T_i = T_1 + T_2 + T_3 + T_4 + T_5 + T_6 \tag{2.18}$$

Since the sources of thrust, i.e. the propellers, are not located in the center of gravity, they will create torques around the different axes of rotation. With some basic geometry seen in Figure 2.3, it's easy to find the produced torque. Around the x_B-axis the torque M_p from propeller thrust is

$$M_p = -\frac{l}{2}T_1 - lT_2 - \frac{l}{2}T_3 + \frac{l}{2}T_4 + lT_5 + \frac{l}{2}T_6 \tag{2.19}$$

where l is the length of each arm. Around the y_B-axis the torque M_q is

$$M_q = \frac{l\sqrt{3}}{2}T_1 - \frac{l\sqrt{3}}{2}T_3 - \frac{l\sqrt{3}}{2}T_4 + \frac{l\sqrt{3}}{2}T_6 \tag{2.20}$$

The torque around the z_B-axis is a result of Newton's third law. When the DC-motor accelerates and keeps the propeller rotating, it exerts a torque on the propeller shaft. The motor will be subject to an equally sized torque in opposite direction from the propeller shaft. Since the motor is mounted to the airframe, the torque will propagate to the airframe. This torque is often called reaction torque. If the reaction torque from propeller i is called τ_i the total torque around the z_B-axis denoted M_r is

$$M_r = -\tau_1 + \tau_2 - \tau_3 + \tau_4 - \tau_5 + \tau_6 \tag{2.21}$$

The reaction torque is produced by two different sources. When the motors and propellers accelerate, they will exert a torque on the air frame. Also, pressing the propeller through the air creates friction. This friction is called aerodynamic torque. The contribution from accelerating the propeller is neglected since its duration will be very quick, and thus the reaction torque will be equal to the aerodynamic torque.

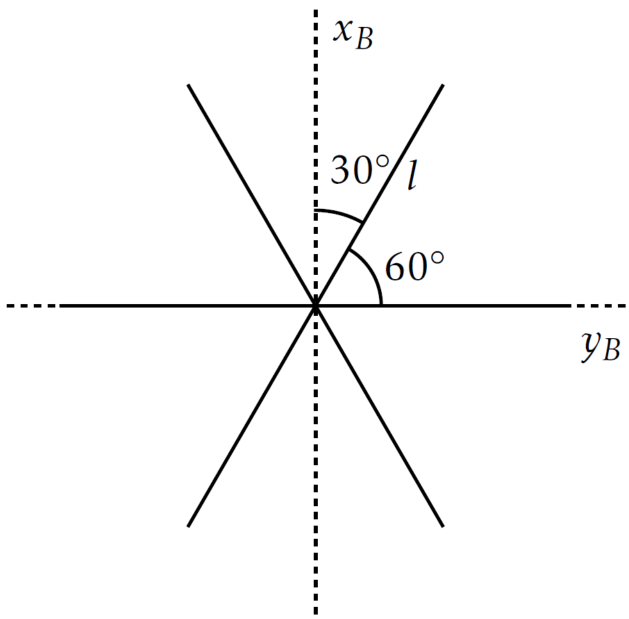

Figure 2.3: The geometry of the hexarotor seen from above

The thrust and aerodynamic torque produced by a propeller can be related to the rotational speed of the propeller blades. If the rotational speed of propeller i

is denoted Ω_i, then the generated thrust T_i is

$$T_i = k_T \Omega_i^2 \tag{2.22}$$

where k_T is a propeller specific constant explained more thoroughly later. In the same way, the aerodynamic torque Q_i of propeller i becomes

$$Q_i = k_Q \Omega_i^2 \tag{2.23}$$

where k_Q is a propeller specific constant.

Now, the total torque $\boldsymbol{M}_A$ from the aerodynamic effects of the propeller can be written

$$\boldsymbol{M}_A = \begin{bmatrix} -\frac{l}{2}k_T\Omega_1^2 - lk_T\Omega_2^2 - \frac{l}{2}k_T\Omega_3^2 + \frac{l}{2}k_T\Omega_4^2 + lk_T\Omega_5^2 + \frac{l}{2}k_T\Omega_6^2 \\ \frac{l\sqrt{3}}{2}k_T\Omega_1^2 - \frac{l\sqrt{3}}{2}k_T\Omega_3^2 - \frac{l\sqrt{3}}{2}k_T\Omega_4^2 + \frac{l\sqrt{3}}{2}k_T\Omega_6^2 \\ -k_Q\Omega_1^2 + k_Q\Omega_2^2 - k_Q\Omega_3^2 + k_Q\Omega_4^2 - k_Q\Omega_5^2 + k_Q\Omega_6^2 \end{bmatrix} \tag{2.24}$$

and the total force $\boldsymbol{F}_A$ becomes

$$\boldsymbol{F}_A = \begin{bmatrix} 0 \\ 0 \\ -k_T\left(\Omega_1^2 + \Omega_2^2 + \Omega_3^2 + \Omega_4^2 + \Omega_5^2 + \Omega_6^2\right) \end{bmatrix} \tag{2.25}$$

2.4.3 Gyroscopic effects from propellers

A propeller is a mass which rotates both around the propeller shaft, but since it is connected to the airframe, it will follow the rotations of the airframe. If the rotation around the shaft is the spin and the rotation of the airframe is the precession, the gyroscopic torque produced by the propeller can be found. The spin quantity is already defined as Ω_i and the precession vector as ω. The gyroscopic torque $\boldsymbol{M}_{G_i}$ from propeller i is

$$\boldsymbol{M}_{G_i} = \omega \times J_P \begin{bmatrix} 0 \\ 0 \\ (-1)^i\Omega_i \end{bmatrix} \tag{2.26}$$

where J_P is the inertia matrix of the propeller around the propeller axis and the factor $(-1)^i$ comes from the fact that the propellers rotate in opposite directions. Using the fact that the inertia matrix is a diagonal matrix the expression can be simplified to

$$\boldsymbol{M}_{G_i} = \omega \times \begin{bmatrix} 0 \\ 0 \\ J_{P,zz}(-1)^i\Omega_i \end{bmatrix} = \begin{bmatrix} qJ_{P,zz}(-1)^i\Omega_i \\ -pJ_{P,zz}(-1)^i\Omega_i \\ 0 \end{bmatrix} \tag{2.27}$$

Summing up all propellers, the total gyroscopic torque $\boldsymbol{M}_G$ is

$$\boldsymbol{M}_G = \begin{bmatrix} qJ_{P,zz}(-\Omega_1 + \Omega_2 - \Omega_3 + \Omega_4 - \Omega_5 + \Omega_6) \\ -pJ_{P,zz}(-\Omega_1 + \Omega_2 - \Omega_3 + \Omega_4 - \Omega_5 + \Omega_6) \\ 0 \end{bmatrix} \tag{2.28}$$

2.4.4 Air friction

The air frames movement through the air will cause friction. Because of the shape of the air frame, the air friction is assumed to be low and a simple model is sufficient. The model of the air friction is given by

$$\boldsymbol{F}_R = -A_T \cdot \boldsymbol{V} \tag{2.29}$$

$$\boldsymbol{M}_R = -A_R \cdot \boldsymbol{\omega} \tag{2.30}$$

where A_T and A_R are diagonal matrices with diagonal elements a_T and a_R respectively.

2.5 Rotor model

The rotor model can be separated into two subsystems: the propeller and the electric motor. The propeller converts rotational speed into thrust and the electrical motor converts voltage into rotational speed. This part of the modeling is quite often neglected, but can have a big impact of the performance. The most important part is the time delay of the propulsion system which is the time between the PWM-signal is calculated and the propeller reaching the desired speed.

2.5.1 Propeller model

A proper, accurate propeller model includes studying of momentum theory, blade element theory, relative airflow, blade flapping and the bending and twisting of the propeller blade; it is a quite complicated subject, and much of the result comes out as empirical data. A basic propeller model states that the generated thrust T is

$$T = C_T \rho r_P^4 \pi \Omega^2 \tag{2.31}$$

where ρ is the air density, r_P is the propeller radius and C_T is the thrust coefficient of the propeller. Similarly, the aerodynamic torque Q can be expressed as

$$Q = C_Q \rho r_P^5 \pi \Omega^2 \tag{2.32}$$

where C_Q is the propeller torque constant. Modeling C_T and C_Q is the complicated part of the propeller model and is mostly done empirically. The constants will depend on the geometry of the propeller as well as the direction the propeller is traveling in.

2.5.2 Electric motor model

The electrical motors used on the platform are rather complex to model in detail. The ESC receives a PWM-signal which it decodes to a desired rotor speed. The motor contains permanent magnets. To make the magnets rotate, a magnetic field is moved around the housing of the motor. The magnets are trying to follow that magnetic field, and a rotation is created. The speed of this rotation depends on how fast the magnetic field is moved around which is made by turning on and off the poles in the housing.

The interesting part of the dynamics of the rotor is the time constant and the PWM to rotor speed relation. Because of that, a PWM to voltage mapping and a classic model of a dc-motor is considered sufficient for the purpose of this thesis. By assuming that the inductance and the friction caused by the moving parts of the motor are zero the model becomes a first order model which is considered enough. The model is defined by

$$I = \frac{V_{in} - K_v \Omega}{R} \tag{2.33}$$

$$\dot{\Omega} = \frac{K_t}{J_P} + I - \frac{\tau}{J_P} \tag{2.34}$$

$$\tau = Q \tag{2.35}$$

where I is the current, V_{in} is the voltage input, K_v and K_t are speed and torque constants, Ω is the motor speed, τ is the reaction torque and Q is the aerodynamic torque.

2.6 Final model

Combining the first derivative of the position equation (2.5), the first derivative of the orientation equation (2.7), the translational dynamics equation (2.12), the rotational dynamics equation (2.16), and the external forces equations (2.17), (2.24), (2.25) and (2.28) the final model becomes

2.6.1 System input

The input to the system can be chosen as several physical quantities. To fulfill the requirement of platform independence, or at least make the model as modular as possible, one can choose the system input as total aerodynamic force and torque produced by the propellers and the gyroscopic torque as a disturbance.

To extend the model, the rotational speed of the propellers can be chosen as input to the system. Then the model can include a complex relation between rotational speed of the propellers and generated trust and reaction torque. Going even further, one can include the electrical motors in the model. These are controlled by the PWM-input to the ESC, and then the duty cycle of the PWM-signals could be the input signal.

In this model, the rotational speed of the propellers will be chosen as input. To simplify notation, these will be mapped to the virtual input signals by

$$F_w = k_T\left(\Omega_1^2 + \Omega_2^2 + \Omega_3^2 + \Omega_4^2 + \Omega_5^2 + \Omega_6^2\right) \tag{2.36a}$$

$$M_p = -\frac{l}{2}k_T\Omega_1^2 - lk_T\Omega_2^2 - \frac{l}{2}k_T\Omega_3^2 + \frac{l}{2}k_T\Omega_4^2 + lk_T\Omega_5^2 + \frac{l}{2}k_T\Omega_6^2 \tag{2.36b}$$

$$M_q = \frac{l\sqrt{3}}{2}k_T\Omega_1^2 - \frac{l\sqrt{3}}{2}k_T\Omega_3^2 - \frac{l\sqrt{3}}{2}k_T\Omega_4^2 + \frac{l\sqrt{3}}{2}k_T\Omega_6^2 \tag{2.36c}$$

$$M_r = -k_Q\Omega_1^2 + k_Q\Omega_2^2 - k_Q\Omega_3^2 + k_Q\Omega_4^2 - k_Q\Omega_5^2 + k_Q\Omega_6^2 \tag{2.36d}$$

$$W_G = -\Omega_1 + \Omega_2 - \Omega_3 + \Omega_4 - \Omega_5 + \Omega_6 \tag{2.36e}$$

where F_w corresponds to the third component of aerodynamic force $\boldsymbol{F}_A$ from the propellers defined by equation (2.25) and M_p, M_q and M_r to the components in the aerodynamic torques $\boldsymbol{M}_A$ defined by equation (2.24) and where W_G is the input to the disturbance caused by gyroscopic torques.

2.6.2 Final system model

For convenience, the system model is written in component form where the inputs have been included

$$\dot{x} = c\theta c\psi u + (s\phi s\theta c\psi - c\phi s\psi)v + (c\phi s\theta c\psi + s\phi s\psi)w \tag{2.37a}$$

$$\dot{y} = c\theta s\psi u + (s\phi s\theta s\psi + c\phi c\psi)v + (c\phi s\theta s\psi - s\phi c\psi)w \tag{2.37b}$$

$$\dot{z} = -s\theta u + s\phi c\theta v + c\phi c\theta w \tag{2.37c}$$

$$\dot{u} = rv - qw + s\theta g - \frac{a_T}{m}u \tag{2.37d}$$

$$\dot{v} = pw - ru + s\phi c\theta g - \frac{a_T}{m}v \tag{2.37e}$$

$$\dot{w} = qu - pv + c\phi c\theta g - \frac{1}{m}F_w - \frac{a_T}{m}w \tag{2.37f}$$

$$\dot{\phi} = p + s\phi t\theta q + c\phi t\theta r \tag{2.37g}$$

$$\dot{\theta} = c\phi q - s\phi r \tag{2.37h}$$

$$\dot{\psi} = \frac{s\phi}{c\theta}q + \frac{c\phi}{c\theta}r \tag{2.37i}$$

$$\dot{p} = \frac{J_{yy} - J_{zz}}{J_{xx}}qr + \frac{1}{J_{xx}}M_p + \frac{J_{P,zz}}{J_{xx}}qW_G - \frac{a_R}{J_{xx}}p \tag{2.37j}$$

$$\dot{q} = \frac{J_{zz} - J_{xx}}{J_{yy}}rp + \frac{1}{J_{yy}}M_q - \frac{J_{P,zz}}{J_{yy}}pW_G - \frac{a_R}{J_{yy}}q \tag{2.37k}$$

$$\dot{r} = \frac{J_{xx} - J_{yy}}{J_{zz}}pq + \frac{1}{J_{zz}}M_r - \frac{a_R}{J_{zz}}r \tag{2.37l}$$

Chapter 3

Control

3.1 The PID controller

Position, velocity and angle controllers are initially implemented using PID techniques. PID controllers are very popular, easily implemented and can be tuned by common users.

$$y(t) = K_P e(t) + K_I \int_{t_0}^{t} e(\tau) d\tau + K_D \frac{\mathrm{d}e(t)}{\mathrm{d}t} \tag{3.1}$$

where K_P, K_I and K_D are proportional, integral and derivative gains respectively, and $e(t)$ is the error between reference signal and measured value. The controllers were implemented on the autopilot and tuned empirically. They consist of position, velocity and attitude control.

3.2 Position control

The position of the hexarotor is controlled by the following equation:

$$V_d(t) = K_P e_p(t) + K_I \int_{t_0}^{t} e_p(\tau) d\tau + K_D \frac{\mathrm{d}e_p(t)}{\mathrm{d}t} \tag{3.2}$$

with

$$e_p(t) = p_d(t) - \hat{p}(t) \tag{3.3}$$

where e_p is the position error, p_d is the desired position and $\hat{p}$ is the measured one, whereas V_d is the desired velocity of the hexarotor. A block diagram of this controller is shown in Figure 3.1.

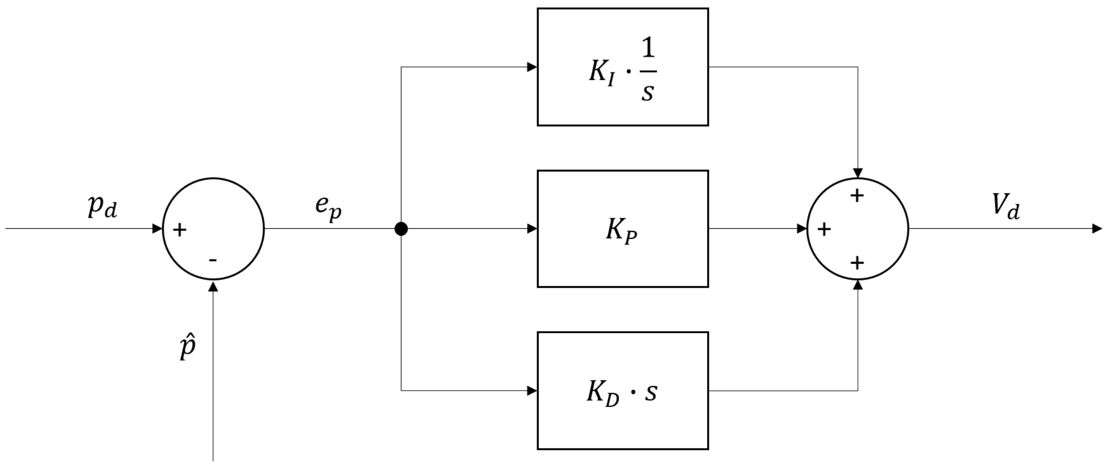

Figure 3.1: Block diagram of the position controller

The velocity along x-axis, u, will be given by the equation (3.2) using $e_{p_x}(t) = x_d(t) - \hat{x}(t)$, where x_d and $\hat{x}$ are the desired and measured position. The same procedure can be repeated for velocities v and w, using y and z positions respectively, thus obtaining the following equations:

$$u_d(t) = K_P e_{p_x}(t) + K_I \int_{t_0}^{t} e_{p_x}(\tau) d\tau + K_D \frac{\mathrm{d}e_{p_x}(t)}{\mathrm{d}t} \tag{3.4}$$

$$v_d(t) = K_P e_{p_y}(t) + K_I \int_{t_0}^{t} e_{p_y}(\tau) d\tau + K_D \frac{\mathrm{d}e_{p_y}(t)}{\mathrm{d}t} \tag{3.5}$$

$$w_d(t) = K_P e_{p_z}(t) + K_I \int_{t_0}^{t} e_{p_z}(\tau) d\tau + K_D \frac{\mathrm{d}e_{p_z}(t)}{\mathrm{d}t} \tag{3.6}$$

with

$$e_{p_x}(t) = x_d(t) - \hat{x}(t) \tag{3.7}$$

$$e_{p_y}(t) = y_d(t) - \hat{y}(t) \tag{3.8}$$

$$e_{p_z}(t) = z_d(t) - \hat{z}(t) \tag{3.9}$$

3.3 Velocity control

This controller needs to be split into two different blocks: the first one controls u and v velocities, in order to generate pitch and roll angles, respectively, as outputs; the second one controls the w velocity, and this gives the desired vertical force as output.

3.3.1 Velocities along x and y axes

The velocities u and v of the hexarotor are controlled by the following equations:

$$\phi_d(t) = K_P e_{V_u}(t) + K_I \int_{t_0}^{t} e_{V_u}(\tau) d\tau + K_D \frac{\mathrm{d}e_{V_u}(t)}{\mathrm{d}t} \tag{3.10}$$

$$\theta_d(t) = K_P e_{V_v}(t) + K_I \int_{t_0}^{t} e_{V_v}(\tau) d\tau + K_D \frac{\mathrm{d}e_{V_v}(t)}{\mathrm{d}t} \tag{3.11}$$

with

$$e_{V_u}(t) = u_d(t) - \hat{u}(t) \tag{3.12}$$

$$e_{V_v}(t) = v_d(t) - \hat{v}(t) \tag{3.13}$$

where where e_{V_u} and e_{V_v} are the velocities error along x and y axes respectively, u_d and v_d are the desired velocities, $\hat{u}$ and $\hat{v}$ are the measured ones. The block diagrams of these controllers are shown in Figures 3.2 and 3.3.

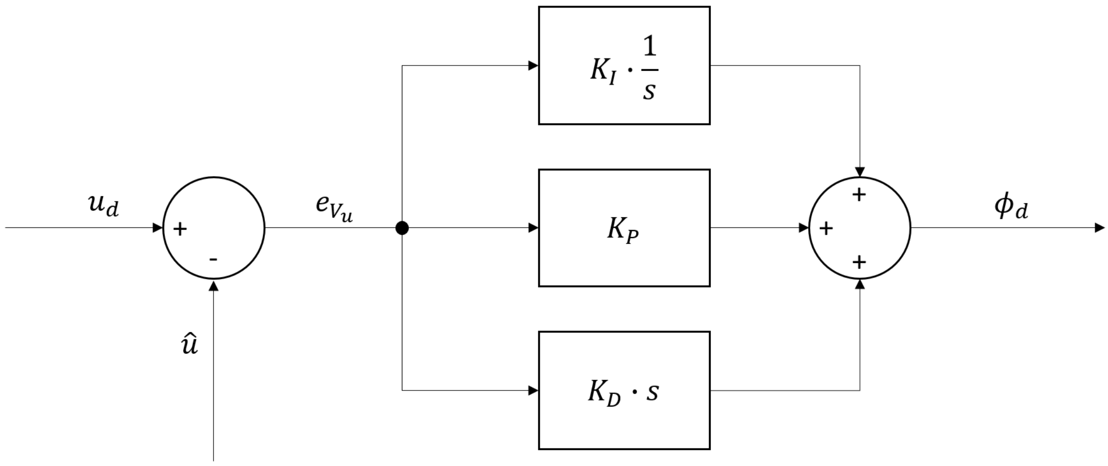

Figure 3.2: Block diagram of the u velocity controller

It is necessary to limit these outputs, so that desired angles will never exceed ±90°. In order to avoid large values in pitch and roll rates, a filter block has been implemented to limit ϕ_d and θ_d at ±45°

3.3.2 Velocity along z axis

The velocity w of the hexarotor is controlled by the following equation:

$$U_1(t) = K_P e_{V_w}(t) + K_I \int_{t_0}^{t} e_{V_w}(\tau) d\tau + K_D \frac{\mathrm{d}e_{V_w}(t)}{\mathrm{d}t} \tag{3.14}$$

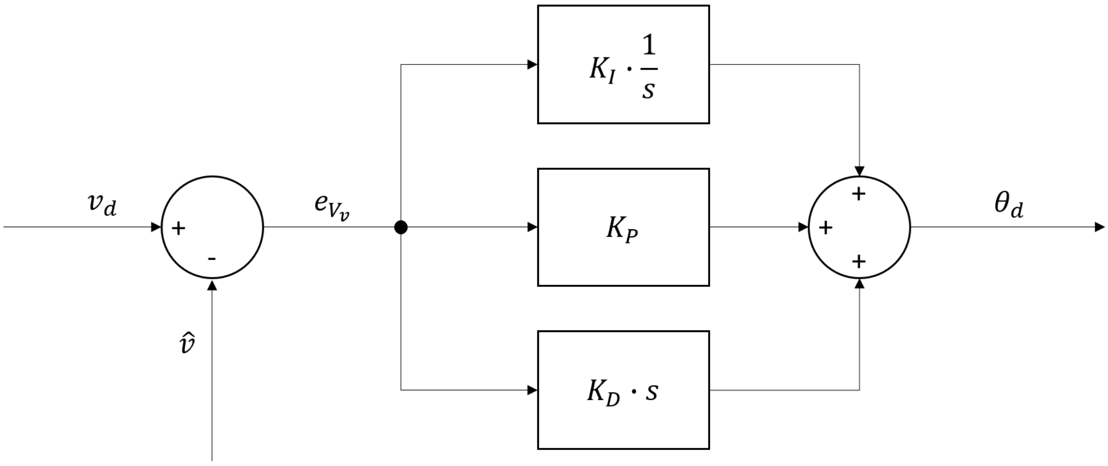

Figure 3.3: Block diagram of the v velocity controller

with

$$e_{V_w}(t) = w_d(t) - \hat{w}(t) \tag{3.15}$$

where where e_{V_w} is the velocity error along z axis, w_d is the desired velocity, $\hat{w}$ is the measured one and U_{1_d} is the desired vertical force. A block diagram of this controller is shown in Figure 3.4.

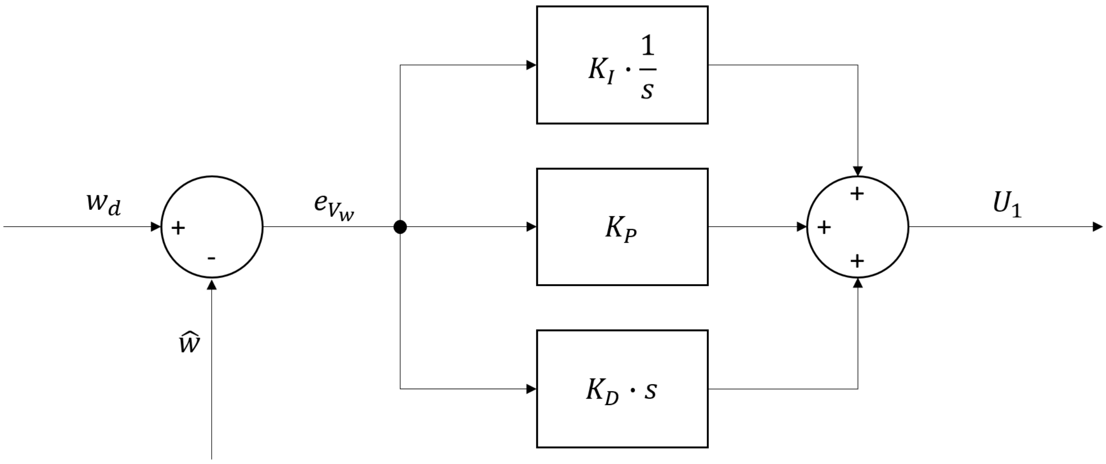

Figure 3.4: Block diagram of the w velocity controller

3.4 Attitude control

Pitch, roll and yaw controllers generate the remaining outputs that the motor model needs to assess the right angular velocities of the rotors, as shown in the following equations:

$$U_2(t) = K_P e_\phi(t) + K_I \int_{t_0}^{t} e_\phi(\tau) d\tau + K_D \frac{\mathrm{d}e_\phi(t)}{\mathrm{d}t} \tag{3.16}$$

$$U_3(t) = K_P e_\theta(t) + K_I \int_{t_0}^{t} e_\theta(\tau) d\tau + K_D \frac{\mathrm{d}e_\theta(t)}{\mathrm{d}t} \tag{3.17}$$

$$U_4(t) = K_P e_\psi(t) + K_I \int_{t_0}^{t} e_\psi(\tau) d\tau + K_D \frac{\mathrm{d}e_\psi(t)}{\mathrm{d}t} \tag{3.18}$$

with

$$e_\phi(t) = \phi_d(t) - \hat{\phi}(t) \tag{3.19}$$

$$e_\theta(t) = \theta_d(t) - \hat{\theta}(t) \tag{3.20}$$

$$e_\psi(t) = \psi_d(t) - \hat{\psi}(t) \tag{3.21}$$

where e_ϕ is the error between the desired pitch command U_2 and the actual one; the same notation is applied to the roll command U_3 and the yaw command U_4. The block diagram of these controllers are shown in Figure 3.5-3.6.

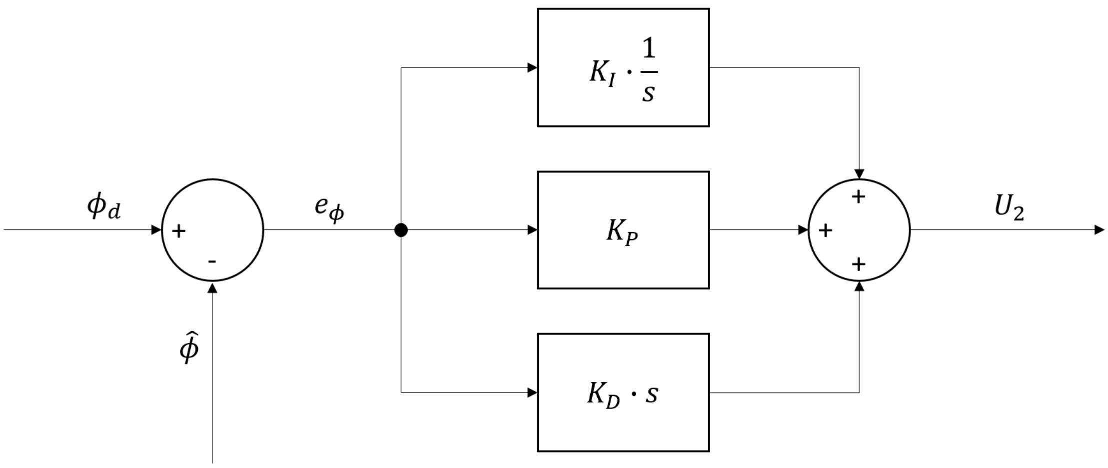

Figure 3.5: Block diagram of the pitch angle controller

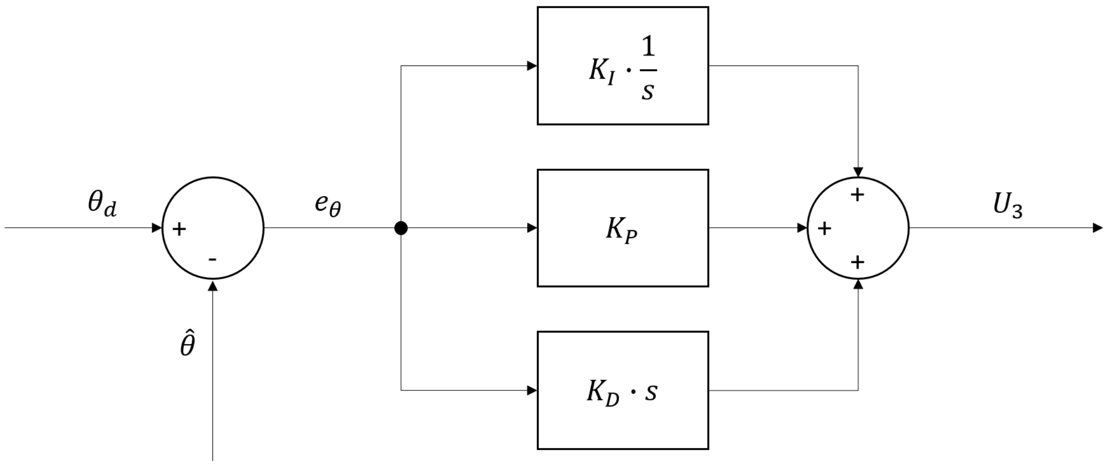

Figure 3.6: Block diagram of the roll angle controller

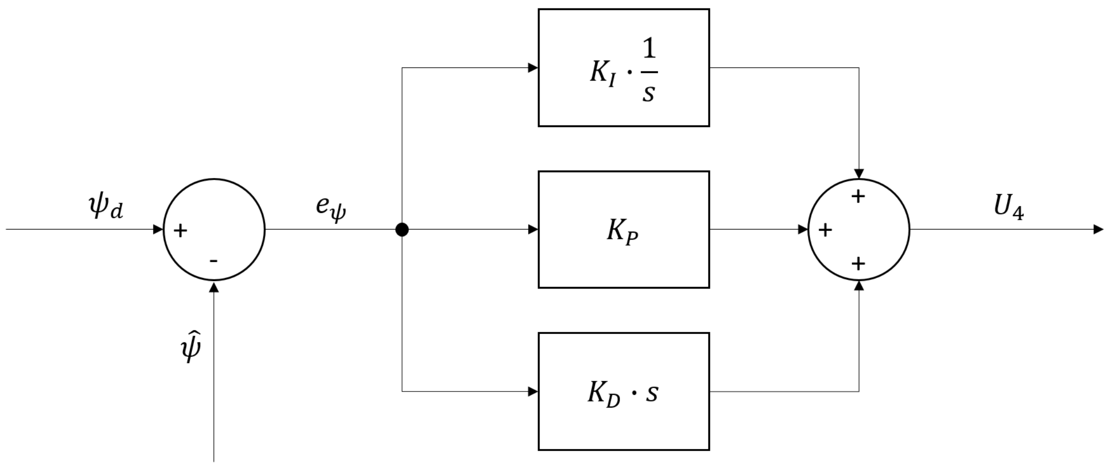

Figure 3.7: Block diagram of the yaw angle controller

Chapter 4

Simulation and results

Once the model of the hexarotor is complete and the control laws are successfully defined, the system needs something that can easily generate the desired position or trajectory to follow. The simulation has been implemented in Simulink, linking all blocks together, as shown in Figure 4.1

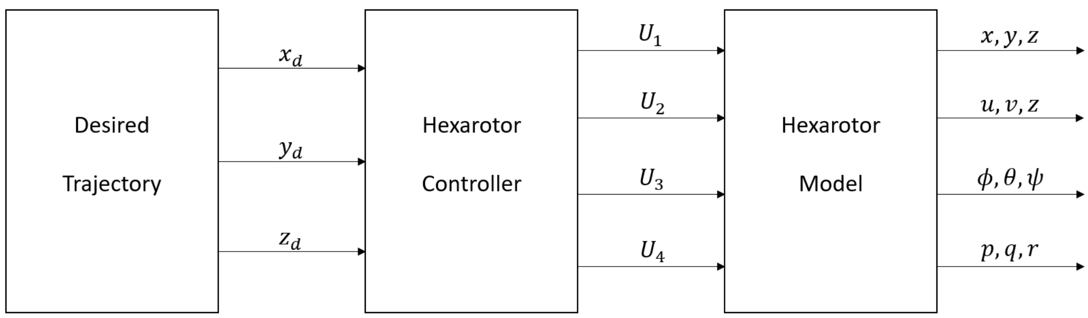

Figure 4.1: Overview of the complete system

The trajectory generator block has the purpose to provide the desired coordinates (x, y, z) to the controller block; those coordinates are time-dependent and each one of them can be easily linked to the previous one and to the next one, in order to shape any type of trajectory, both static and dynamic.

4.1 Linear trajectory along x and y axes

The first trajectory that has been tried is the linear one, in the xy plane. It is represented by the following equation:

$$\begin{cases} x = 5t \\ y = 5t \\ z = 10(e^{-0.1t} - 1) \end{cases} \tag{4.1}$$

The trajectory along the z axis simply represents a soft ascent until the hexarotor reaches 10 m (the value on the graph is negative because of the z axis pointing toward the ground). The trajectories along x and y axes are linear and the derivative of the trajectory equation defines the speed at which the UAV has to reach: 5 m/s along x and 5 m/s along y.

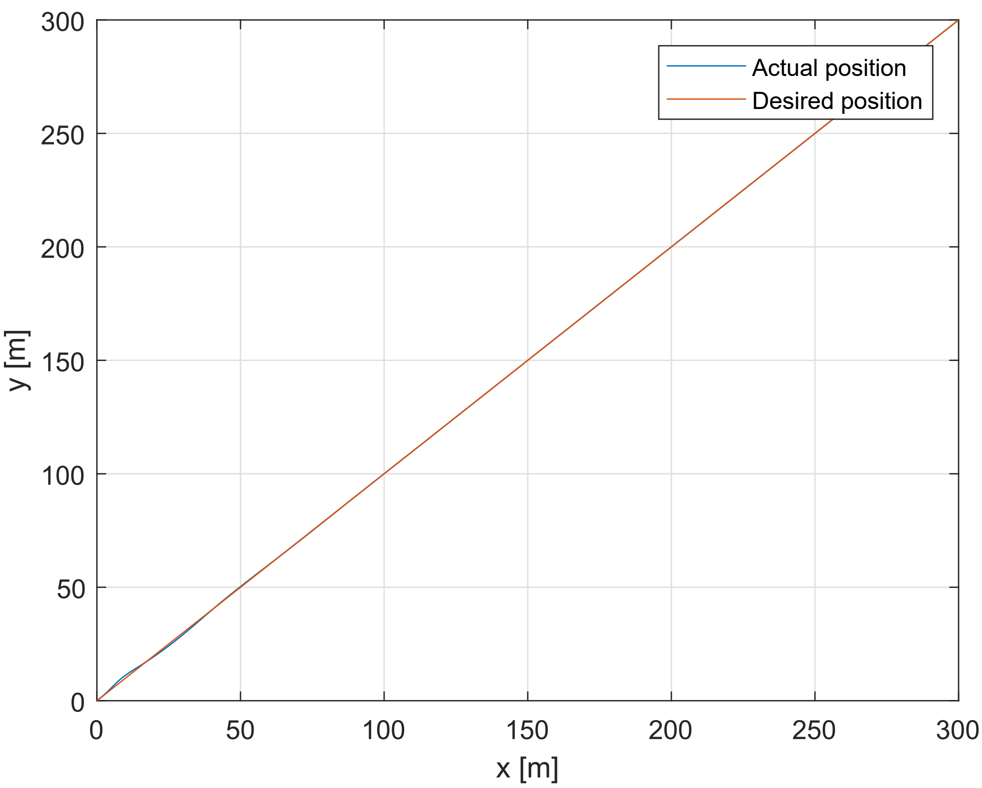

Figure 4.2: Trajectory of the hexarotor in xy plane

The UAV is capable of following perfectly this type of trajectory; the error between the desired and the actual position is quickly canceled in the first seconds of flight.

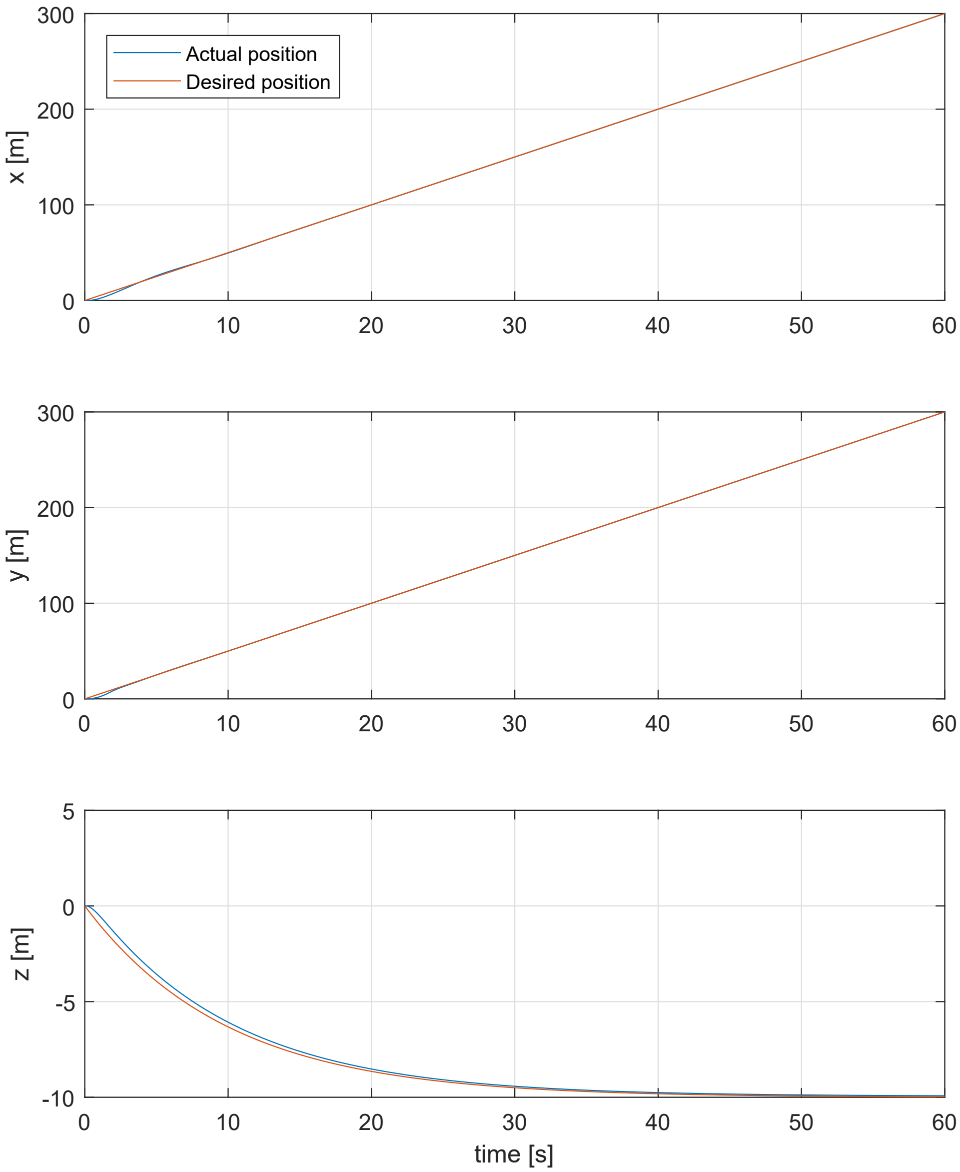

Figure 4.3: Trajectory of the hexarotor along the three axes

The maximum absolute values of the roll and pitch angles are 25°, and then, after few seconds, the attitude of the drone stabilizes and low value of those angles are sufficient to maintain the desired velocity.

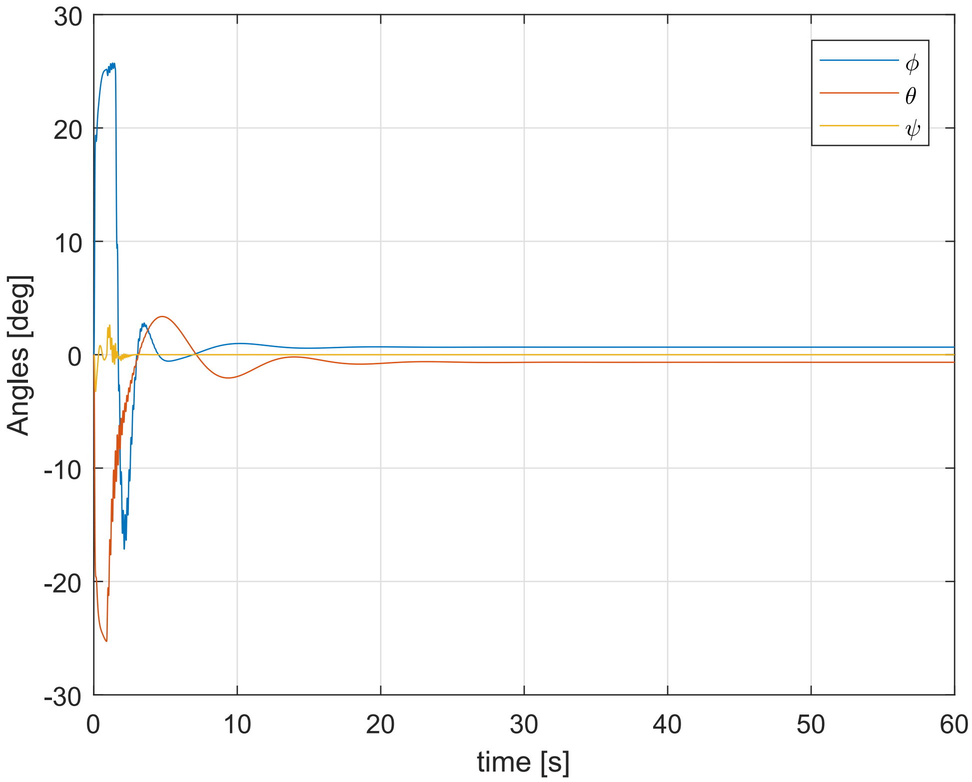

Figure 4.4: Roll, pitch and yaw angles of the hexarotor

4.2 Linear trajectory along y axis

The next step is to add a non-linearity on one axis, the x one in this case. This trajectory is really quick and harder to follow, and it introduces some disturbance in the attitude controller.

$$\begin{cases} x = 5(1 - \cos(0.9t)) \\ y = 5t \\ z = 10(e^{-0.1t} - 1) \end{cases} \tag{4.2}$$

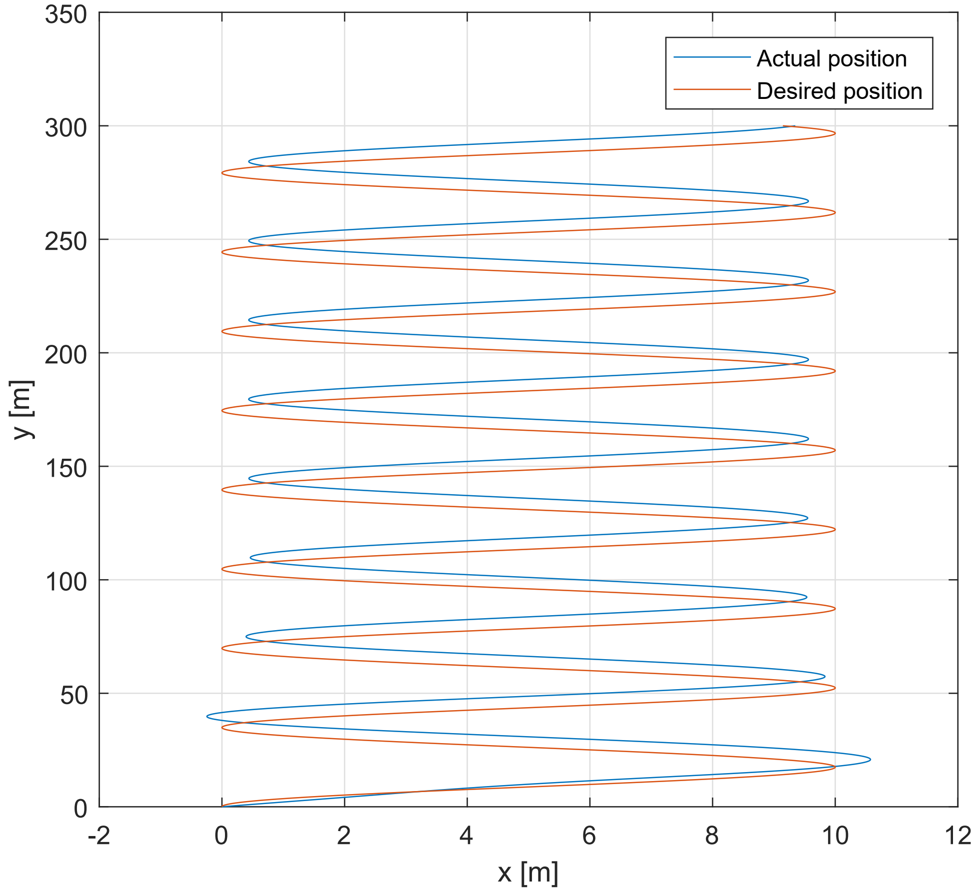

Figure 4.5: Trajectory of the hexarotor in xy plane

It is evident that the UAV can not follow perfectly the desired trajectory, because it can not predict the right path. However, this is a good result, since the error on the x axis is no more than 5%

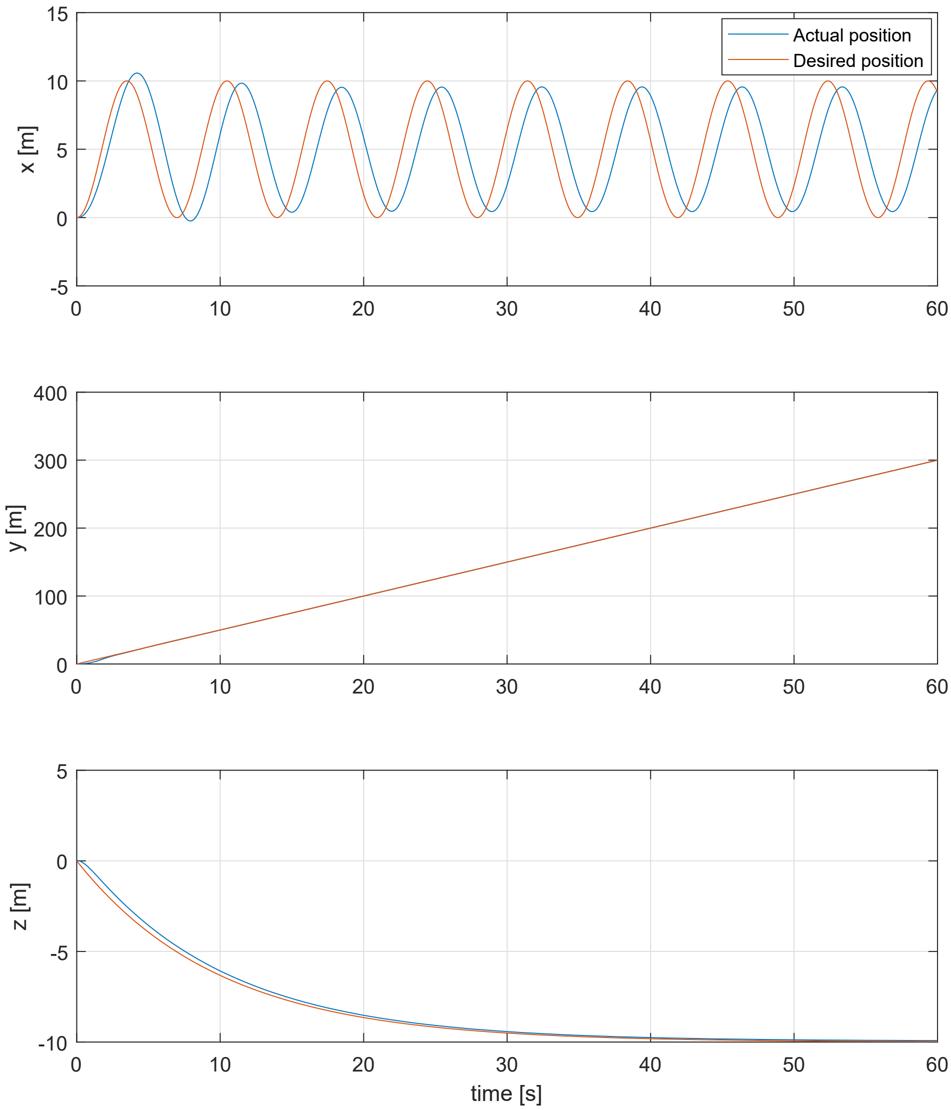

Figure 4.6: Trajectory of the hexarotor along the three axes

The disturbance in pitch and yaw angles in the very first seconds are caused by the speed of the desired trajectory, that makes the system behave as there be a step input. Despite this disturbance, the system is able to stabilize quickly and keeps oscillating between $\pm 20°$.

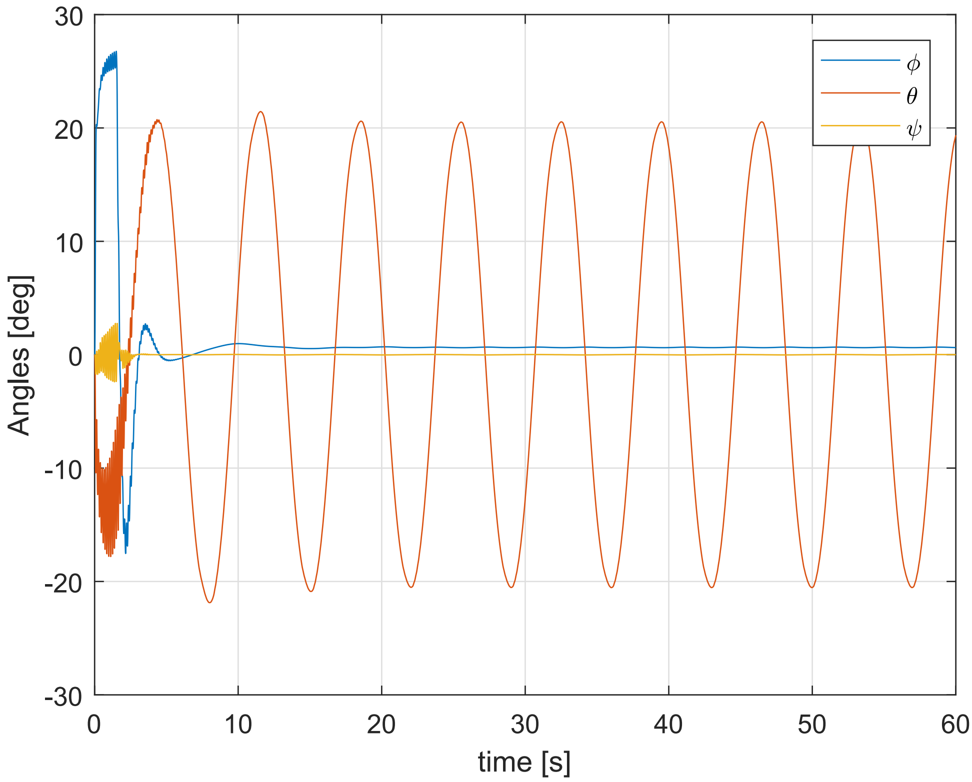

Figure 4.7: Roll, pitch and yaw angles of the hexarotor

4.3 Linear trajectory along x axis

This time the non-linearity has been added on y axis. The behavior of the system is similar to the previous one: position error in y axis doesn't exceed 5% and linear trajectory on x axis is successfully obtained.

$$\begin{cases} x = 5t \\ y = 5sin(0.9u) \\ z = 10(e^{-0.1t} - 1) \end{cases} \tag{4.3}$$

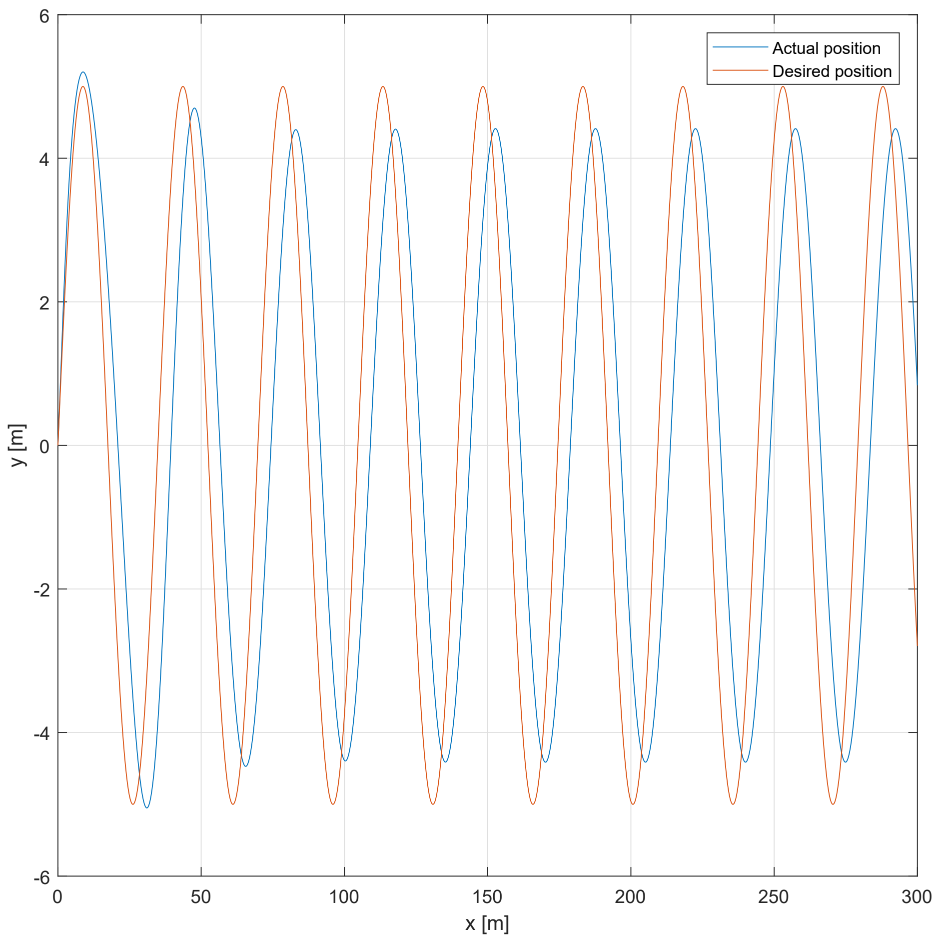

Figure 4.8: Trajectory of the hexarotor in xy plane

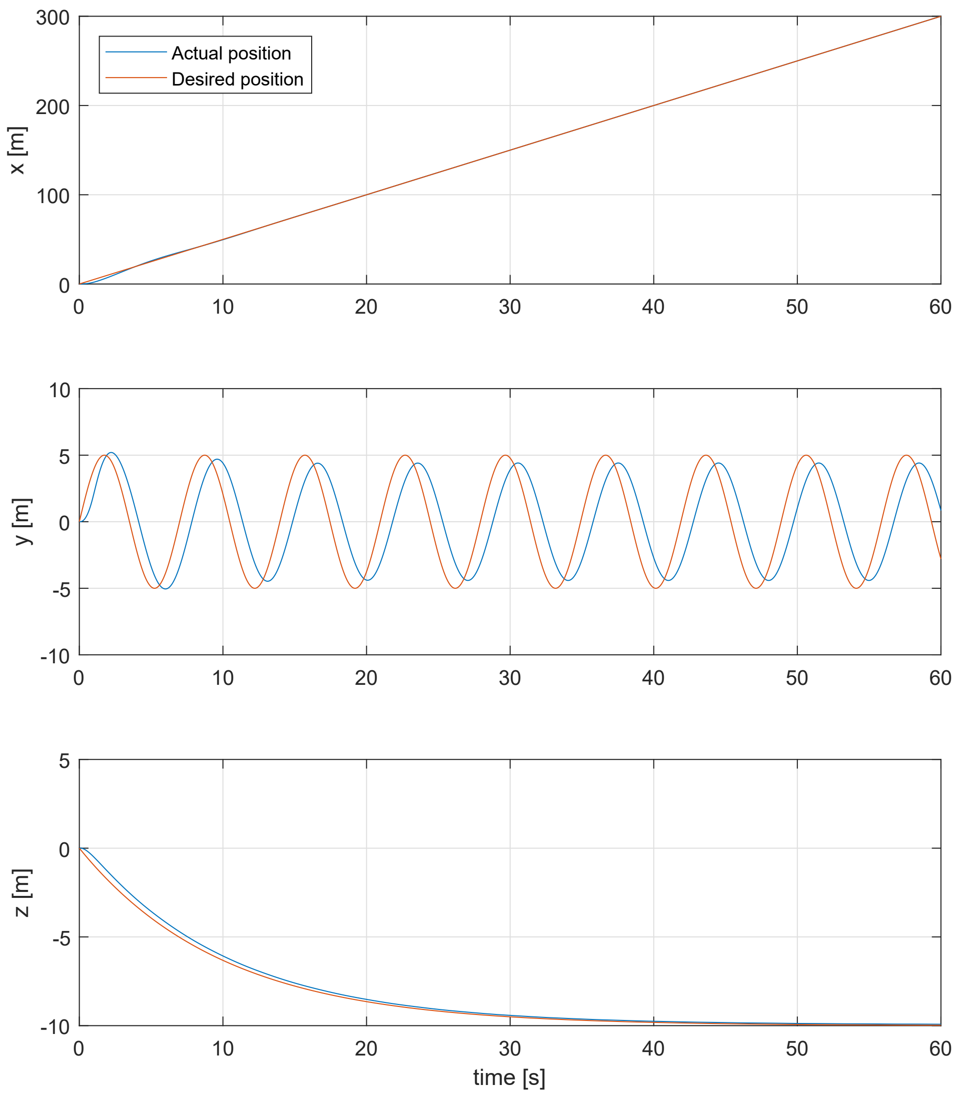

Figure 4.9: Trajectory of the hexarotor along the three axes

It is notable that less disturbance is present in pitch and yaw angles: this is caused by the non-symmetry between x and y axes. In fact, the higher inertia about y axis, guarantee that the system is less unstable when changing the attitude quickly.

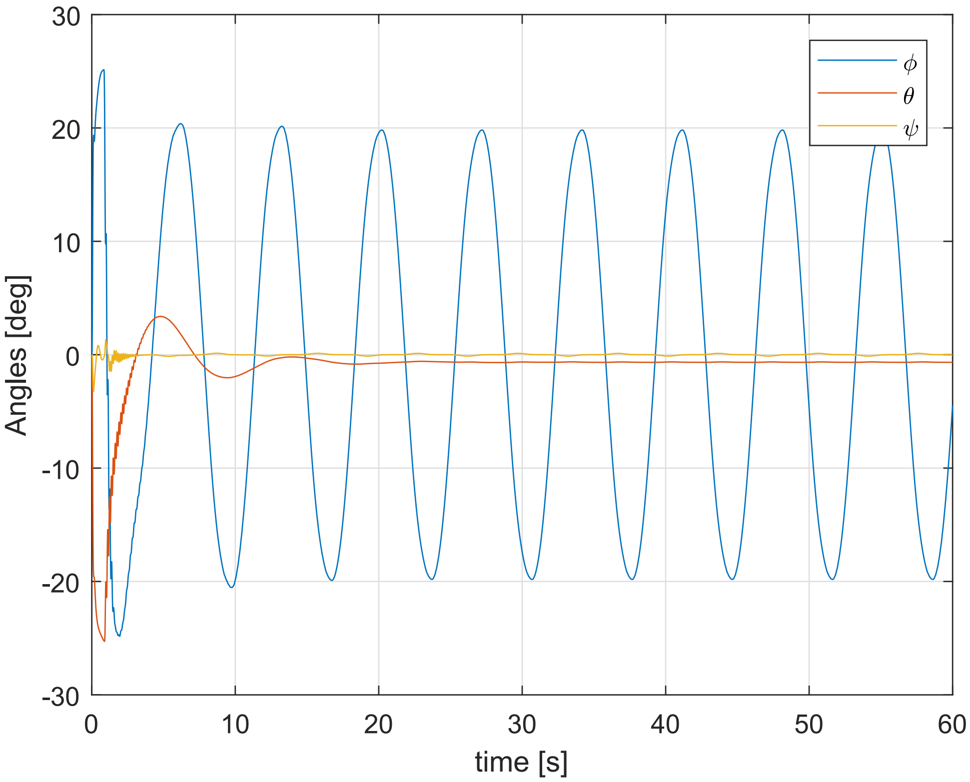

Figure 4.10: Roll, pitch and yaw angles of the hexarotor

4.4 Non-linear trajectory

In this case, the trajectory is non-linear along all the three axes. It consists of a helicoidal trajectory, that climbs until it reaches 10 m.

$$\begin{cases} x = 3(1 - \cos(0.9t) \\ y = 3sin(0.9u) \\ z = 10(e^{-0.1t} - 1) \end{cases} \tag{4.4}$$

This can be considered a good result, since on the x axis the error never exceed 5%; on the y axis a slightly higher error can be observed, but it never goes beyond 10%.

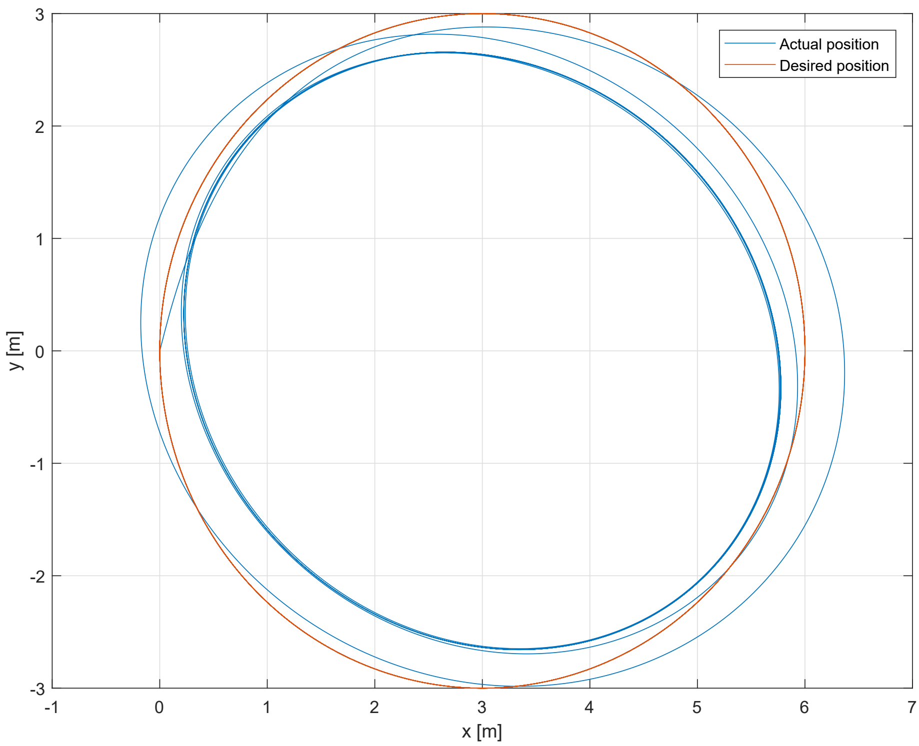

Figure 4.11: Trajectory of the hexarotor in xy plane

A light disturbance on pitch angle is present. However, this never cause instability, because the controller is able to easily and quickly adjust those values, granting stability after about 1 second.

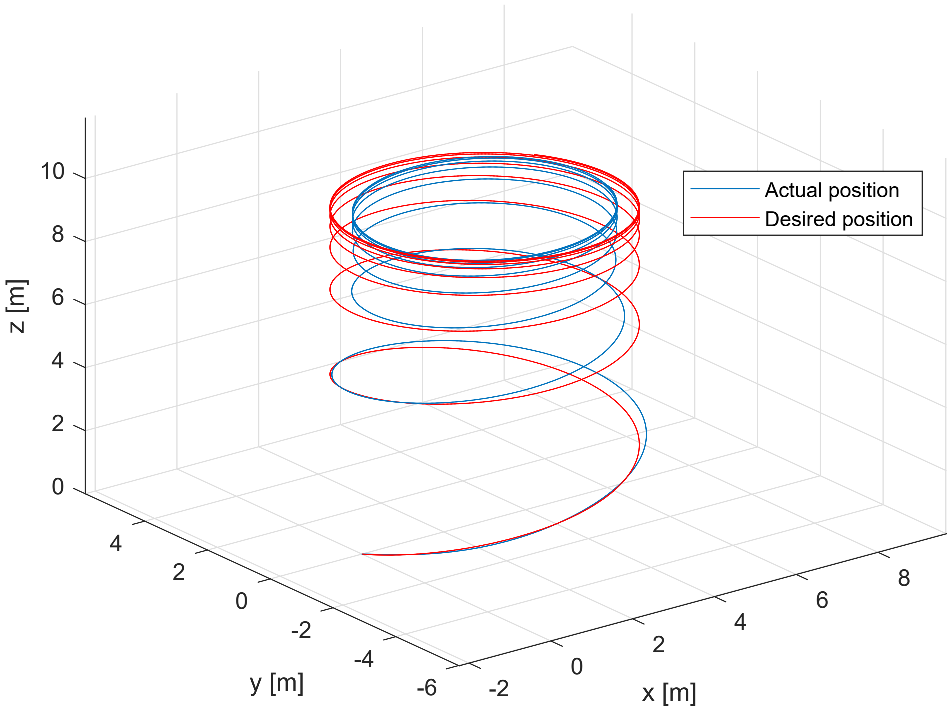

Figure 4.12: Trajectory of the hexarotor in a 3D plot

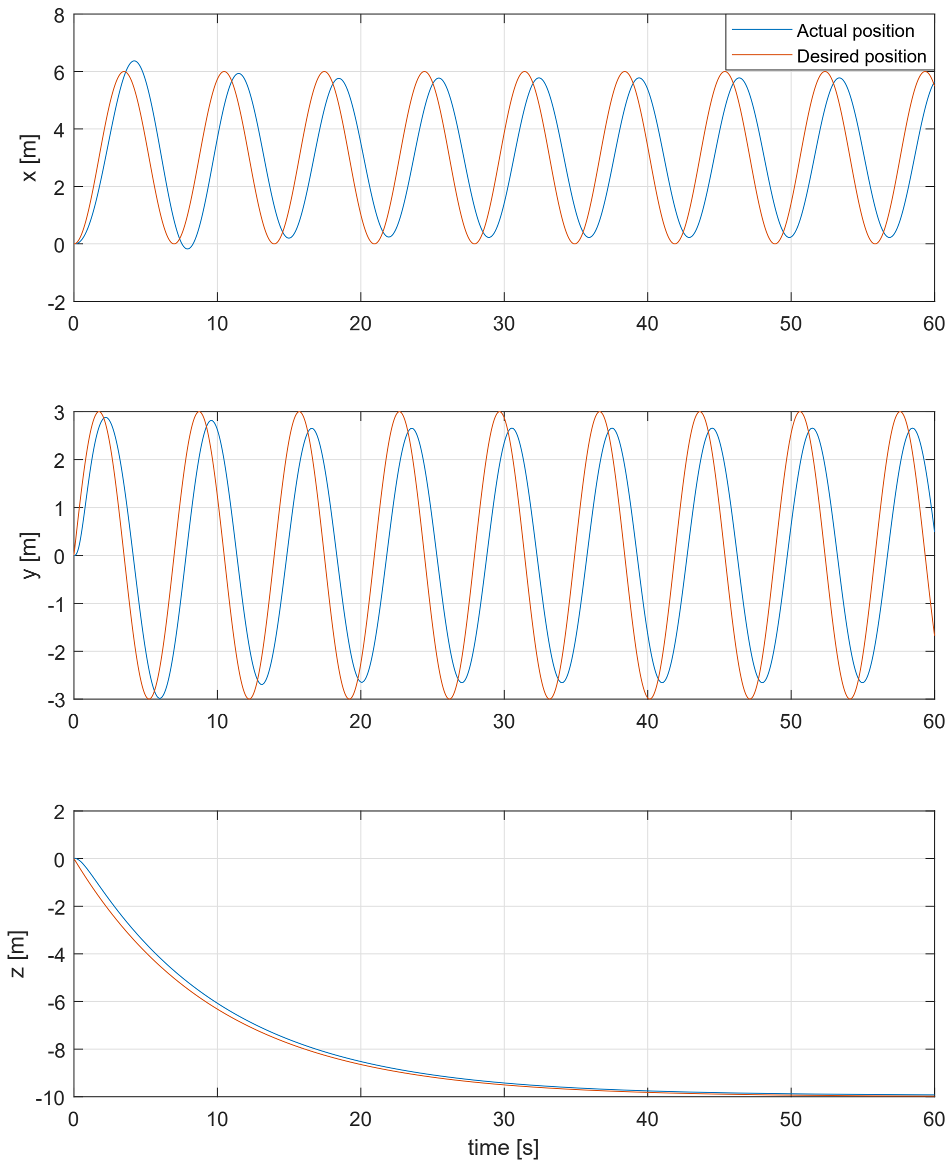

Figure 4.13: Trajectory of the hexarotor along the three axes

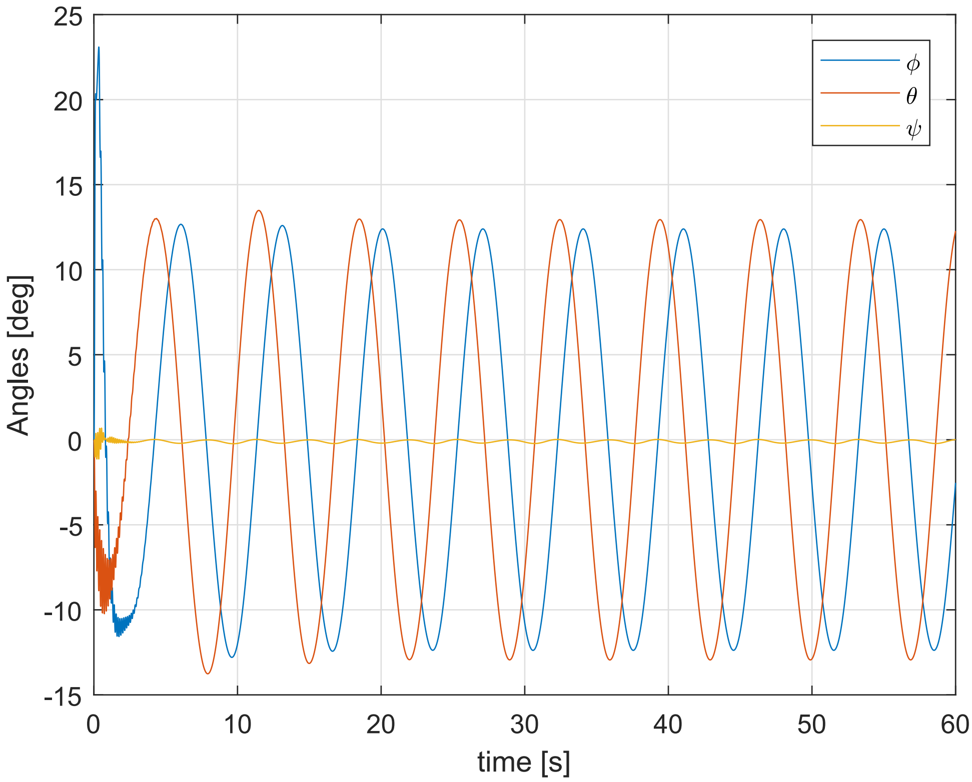

Figure 4.14: Roll, pitch and yaw angles of the hexarotor

4.5 Non-linear dynamic trajectory

In this case, the trajectory is non-linear along all the three axes and the radius of the circle expands constantly. It consists of a dynamic helicoidal trajectory, that climbs until it reaches 10 m.

$$\begin{cases} x = 3(1 - \cos(0.9t) \\ y = 3sin(0.9u) \\ z = 10(e^{-0.1t} - 1) \end{cases} \tag{4.5}$$

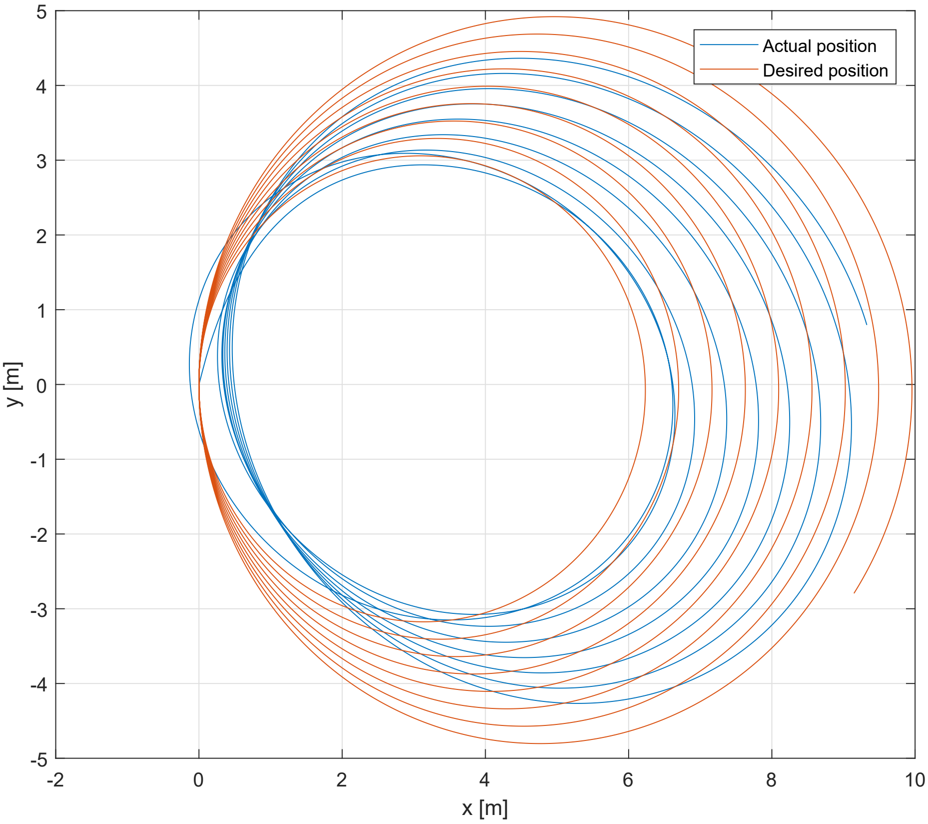

Figure 4.15: Trajectory of the hexarotor in xy plane

This case can be considered in an application of a fire-detection mission, where the trajectory changes his path dynamically. It is a good result, since on the x and y axes the error never exceed 5%.

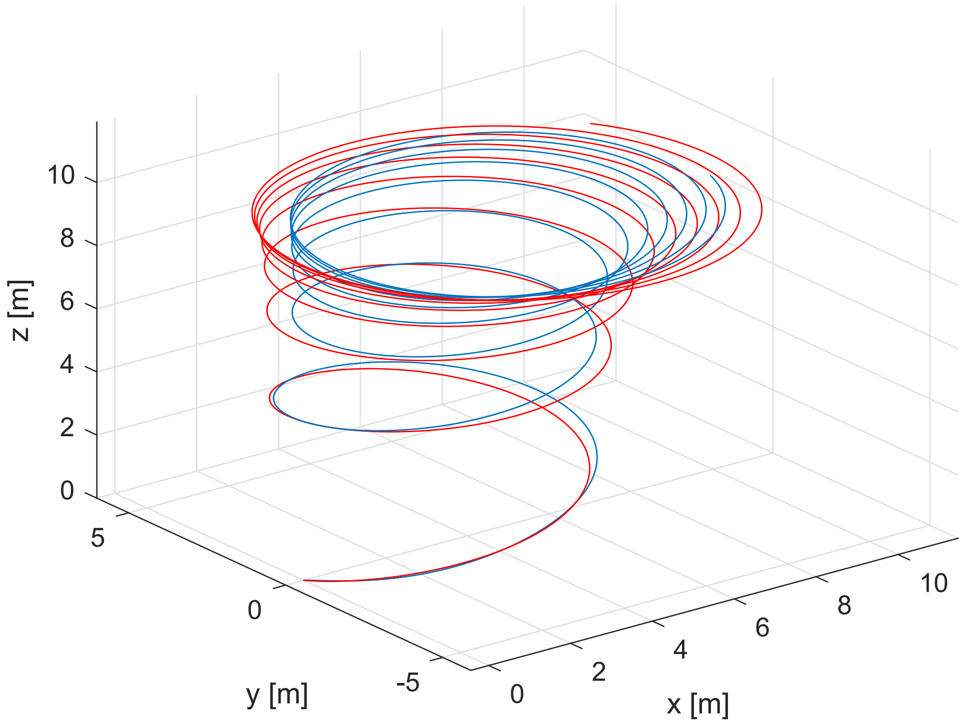

Figure 4.16: Trajectory of the hexarotor in a 3D plot

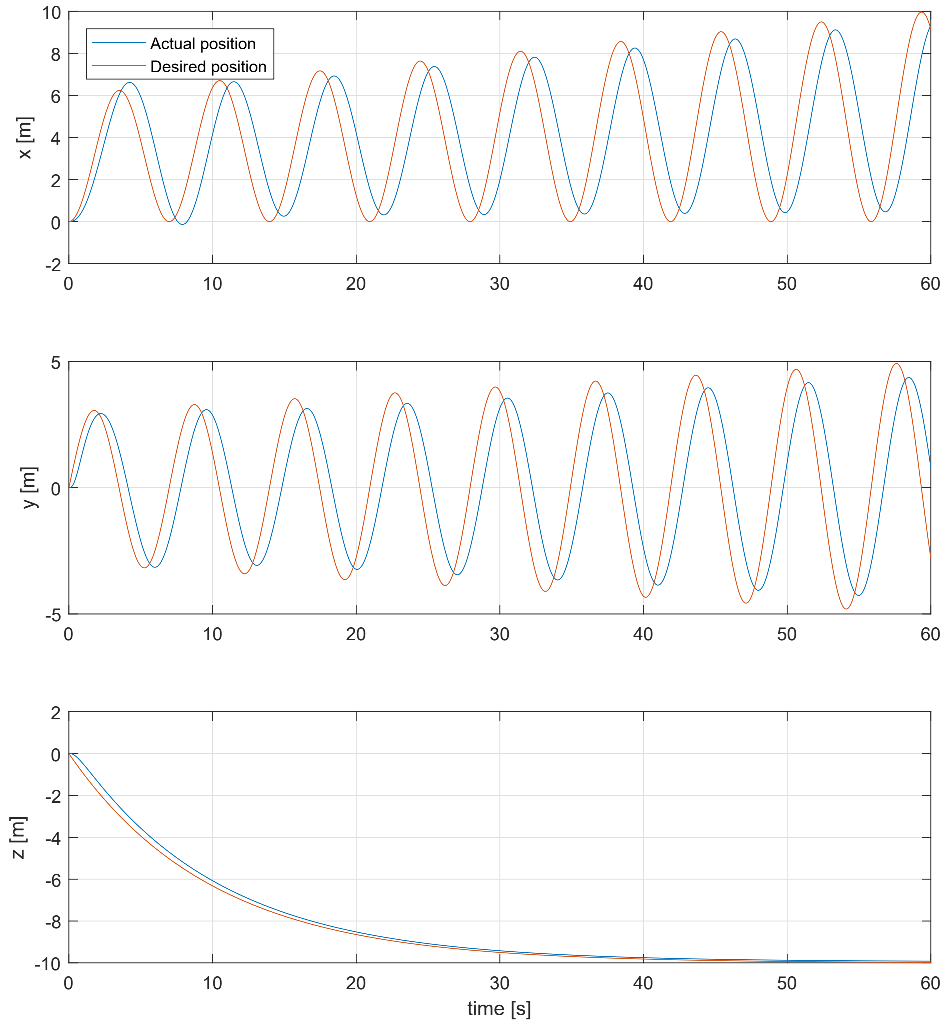

Figure 4.17: Trajectory of the hexarotor along the three axes

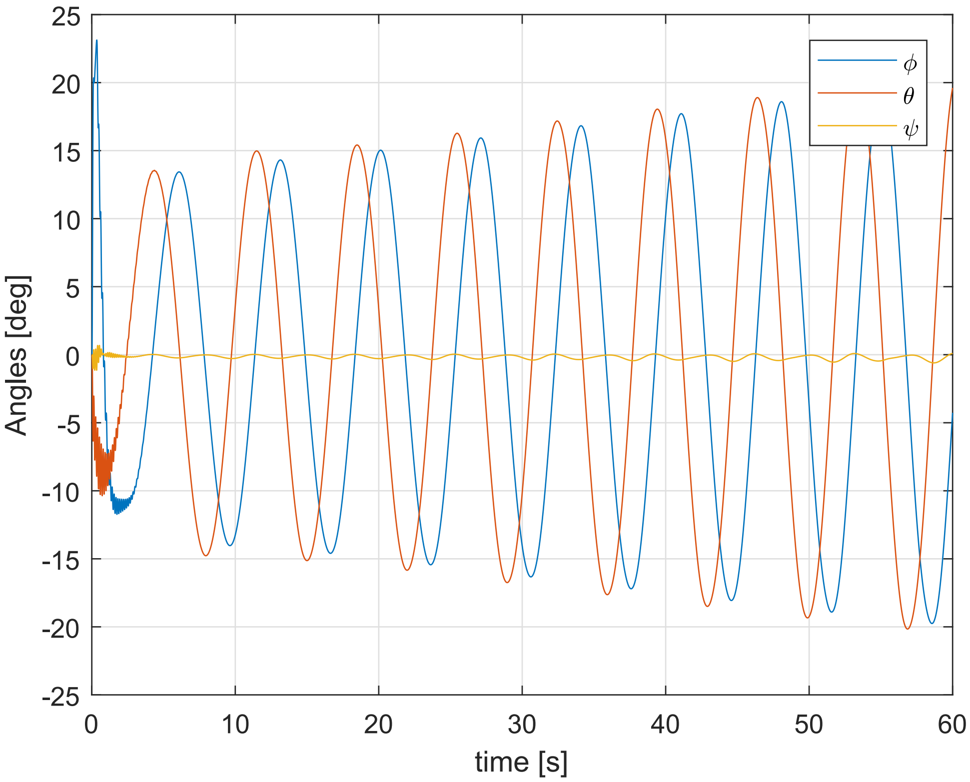

Figure 4.18: Roll, pitch and yaw angles of the hexarotor

Chapter 5

Conclusion

The model implemented is capable of fairly well simulating the behavior of a hexarotor during different flight conditions. Several PID blocks have been used to lower the error between position, velocity and attitude and they have been tuned so that the stability of the UAV is granted.

The trajectory block is able to generate any path in any shape, and the system proved that it is able to follow any of them, with a small error in the worst case. Above all, the UAV demonstrated that it is capable of following a dynamic trajectory as well, and this can be considered the best result in a fire-detection mission.

5.1 Further work

The system is ready to follow dynamic paths and moving waypoints, without excessive error in position and velocity. A future work may have as purpose the installation of a thermal camera, that takes information from the burning ground. The flames keep moving if not extinguished, so it would be a good idea to track at which speed they move and adjust velocity and position.

It would be also a great work trying to swap the PID controller with another one, for example with the Backstepping method, that has a good application in non-linear models.

Finally, building the UAV so that it would be possible to validate the modeled system with a real one could be useful to have a good gain tuning during the controller development.

Bibliography

[1] Alaimo Andrea et al. *PID Controller Applied to Hexacopter Flight*. 2013.

[2] Stevens Brian L., Lewis Frank L., and Johnson Eric N. *Aircraft control and simulation, third edition*. 2016.

[3] Mansson Christian and Stenberg Daniel. “Model-based Design Development and Control of a Wind Resistant Multirotor UAV”. Lunds Universitet, 2014.

[4] Wei Zhong Fum. “Implementation of Simulink controller design on Iris quadrotor”. Naval Postgraduate School, 2015.

[5] Homann Gabriel M. and Waslandery Steven L. “Quadrotor Helicopter Trajectory Tracking Control”. Stanford University.

[6] Pugliese Carratelli Giovanni. “Controllo di un Quadcopter”. Univesita Federico II Napoli, 2012.

[7] Eswarmurthi Gopalakrishnan. “Quadcoptor flight mechanics model and control algorithms”. Czech Technical University, 2016.

[8] Fogelberg Johan. “Navigation and Autonomous Control of a Hexacopter in Indoor Environments”. Lund University, 2013.

[9] Teppo Luukkonen. *Modelling and control of quadcopter*. Aalto University, 2011.

[10] Moussid1 Mostafa, Sayouti Adil, and Hicham Medromi. “Dynamic Modeling and Control of a HexaRotor using Linear and Nonlinear Methods”. In: *International Journal of Applied Information Systems (IJAIS)* (2015).

[11] Moussid1 Mostafa et al. “Autonomous HexaRotor Arial Dynamic Modeling and a Review of Control Algorithms”. In: *International Research Journal of Engineering and Technology (IRJET)* (2015).

[12] *Quadcopter Dynamics, Simulation, and Control*.

[13] Bose1 Simanti, Bagchi Adrija, and Dave Naisargi. “Hexacopter using MATLAB Simulink and MPU Sensing”. In: *International Research Journal of Engineering and Technology (IRJET)* (2017).

[14] Magnusson Tobias. “Attitude Control of a Hexarotor”. Linkopings universitet, 2014.

[15] Artale V., Milazzo C.L.R., and Ricciardello A. *Mathematical Modeling of Hexacopter*. 2013.

[16] Adriana Gabriela Zsurzsan et al. *General model and control of an n rotor helicopter*. 2015.