Fractional Order Controller for Quadcopter Subjected to Ground Effect

A master`s thesis submitted to the

University of Ottawa

in partial fulfilment of the requirements

for the degree of

Master of Applied Science

in

Mechanical Engineering

By

Seyed Alireza Mirghasemi

Under the Supervision of

Prof. Dan Necsulescu

Ottawa-Carleton Institute for Mechanical and Aerospace Engineering

University of Ottawa

Ottawa, Ontario, Canada, 2019

Abstract

Although the concept of fractional calculus was known for centuries, it was not considered in engineering due to the lack of implementation tools and acceptable performance of integer order models and control. However, recently, engineers and researchers started to investigate the potentially high performance of fractional calculus in various fields among which are acoustics, conservation of mass, diffusion equation and specifically in this thesis control theory. The intention of this thesis is to analyze the relative performance of fractional versus integer order PID controller for a quadcopter. Initially, the dynamics of the quadcopter is presented with additional consideration of the ground effect and torque saturation. Then, are introduced the concept of fractional calculus and the mathematical tools to be used for modeling fractional order controller. Finally, the performance of the fractional order controller is evaluated by comparing it to an integer order controller.

Acknowledgement

While doing my research, I have enjoyed the support and encouragement of my family members and friends and I am thankful to all of them. I would like to dedicate the next few paragraphs to mention the most important ones, so that maybe I can partially compensate their kindness and trust, they have had toward me.

First, I would like to thank my parents. Without their spiritual and financial support, it would have been almost impossible for me to continue my studies.

Second, I would like to thank my supervisor, Prof. Dan Necsulescu. His knowledge, guidance and patience facilitated the process of post-graduation study.

Next, I am grateful to two of my closest friends, Mr. Kheiri and Mrs. Haeri, whom I view them as my inseparable siblings, for their encouraging support.

At the end, I want to thank all of my friends who happened to be also my classmates. Their aptitude for science and scientific projects has always been an excellent motivation for me.

Contents

List of Figures

List of Tables

Nomenclature

C	Calibrator of weight
d	Perpendicular distance of motors to center of gravity
F	Force
g	Gravitational Acceleration
H	State-Space Variables
I_x	Moment of Inertia around *x*-axis
I_y	Moment of Inertia around *y*-axis
K_p	Proportional gain
K_d	Derivative gain
K_i	Integral gain
K	Oustaloup filter gain
m	Mass of Quadcopter
m	Slope of the Ramp Function
N	Filter Coefficient
N	Determinant of order of Oustaloup filter
R	Radius of Propellers
u_z	Output of controller responsible for controlling height
u_{θ_x}	Output of controller responsible for controlling pitch angle
u_{θ_y}	Output of controller responsible for controlling roll angle
T	Time margin
ω_i	Angular velocity of i^{th} motor
ω_b	Lower bound frequency of Oustaloup Filter
ω_h	Higher bound frequency of Oustaloup Filter
θ_x	Pitch Angle

θ_y Roll Angle

θ_z Yaw Angle

γ Order of Differentiation

μ Order of Integration

v_i Input to the i^{th} motor

Chapter 1

Introduction

For many years, researchers and engineers have used integer order differential equations, whether linear or nonlinear, in order to model different phenomena or systems. These integer order differential equations have done a satisfactory performance of modeling and describing the characteristic behavior of these systems such that they could be used for a reliable design.

However, in the recent decades, researchers have found out that some phenomena such as conservation of mass [1], groundwater flow [2], advection dispersion equation [3], time-space diffusion equation [4], structural damping model [5] and acoustical wave equation for complex media [6] can be modeled using fractional order differential equation with more accurate results.

The term “fractional” refers to the mathematical concept of fractional calculus. Fractional calculus generalizes and unifies the definition of derivation and integration of a function.

Among the aforementioned applications of fractional calculus, control theory has not been an exception. In general, there are two goals expected to be achieved from a controller. First goal is to prevent the output of a system, which may have a tendency to diverge, from diverging. This is mostly studied in stability analysis of system. After making sure that the output of system converges to a desired output, the second goal is to see how well the output converges based on certain criteria such as speed of response or minimizing oscillations.

A controller affects the dynamics of the system and every system has its own unique dynamics. Therefore, the overall performance of system is evaluated when system and controller are combined together. Additionally, the determination of parameters of the controller is related to the dynamics of system. So far, many proposed controllers were based on integer order differential equations, including conventional proportional-integral-derivative (PID) controllers. The objective of this thesis is to see whether a better result can be obtained when a PID controller is remodeled using fractional differential equations. In fact, many studies have shown that fractional order controllers can perform better, compared to integer order systems. References [7-8] are examples of such studies.

The evaluation of the performance of a controller is impossible without selecting a system. The system of interest in this thesis is a quadcopter. Quadcopters are very important since they are capable of moving in three-dimensional space unlike cars and trains. One may argue that planes move within three-dimensional space as well which is a correct argument however, planes are not operational when they have low translational speed. Helicopters have also the ability to move in three-dimensions and they can be categorized in multirotor family as a single rotor. The advantage of quadcopter compared to helicopter is in maneuverability since quadcopters have a higher number of actuators.

The fact that a quadcopter can operate in three-dimensional space easily makes it desirable for a large range of applications. For instance, depending on the type of camera installed on quadcopter, like standard or night vision or thermal camera, it can be used for video recording, photography, surveillance and scouting in both civil and military purposes. Another example is when quadcopters are designed to be able to carry more weight so that by installing a basket or a container, they can transport different products.

Chapter 2 is dedicated to the dynamics of quadcopter. One of the easiest and basic ways of modeling a quadcopter is to exclude the effect of environment and the limitation of motor power. However, this thesis has considered multiple external parameters that affects the performance of quadcopter in order to add certain degrees of complexity such as ground effect based on Cheeseman model [9] and saturation of actuators so that the model will be more realistic.

Chapter 3 is focused on fractional order controller and the topics necessary to implement it. It begins with the introduction of fractional calculus and then discusses the numerical methodology which approximates fractional differential equation which is used to remodel the integer order characteristic equation of conventional PID controller. Later, the approach to tuning the fractional controller and the issue of the performance of filters are mentioned, comprehensively.

In the end, chapter 4 presents numerous results of the simulations that evaluates the performance of fractional order controller in comparison with integer order controller for

different possible scenarios. It also adds the technique of saturated ramp input instead of step input to enhance the performance of both type of controller.

Chapter 2

Dynamics of Quadcopter

This chapter discusses the dynamic of quadcopter while considering internal and external parameters that affect the overall performance of the system. This is starting with the framework since it is required to define a coordinate system. After establishing framework, constraints that are considered to describe the behavior of quadcopter in a more accurate way, are presented. Later, the model of ground effect is introduced so that in the final step the equations of motion with consideration of all assumptions will be derived. Here, a thought process of this chapter is presented.

- Establishing the frame work and coordinate system
- Deriving equations of motion and the state space model of quadcopter regardless of external parameters.
- Linearization of quadcopter itself
- Understanding how motors generate thrust force which are used in equations of motion and how motors are modeled.
- Understanding how ground affects the motors, and brings nonlinearity to the linearized quadcopter.

2.1 The Framework

Figure 1 shows a schematic sketch of a quadcopter. The quadcopter is represented by two perpendicular rigid rods. A body coordinate frame is attached to the center of quadcopter such that the x axis passes through one rod and y axis passes through the other rod, based on the right-hand rule, while the z axis is perpendicular to the surface of the page outward. In order to describe the orientation of the quadcopter, θ_x is defined as the angular displacement around x axis and θ_y is defined as the angular displacement around y axis and θ_z is defined as the angular

displacement around z axis. All of these variables can be presented in a vector form as $p = [x\ y\ z\ \theta_x\ \theta_y\ \theta_z]^T$. Thrust forces generated by motors, are denoted F_i , $i = 1,2,3,4$ where i indicates the number of the motor. A diagonal matrix of inertia has also been considered with the elements I_x, I_y, I_z respectively.

A complete analysis of quadcopter`s framework is presented in [10].

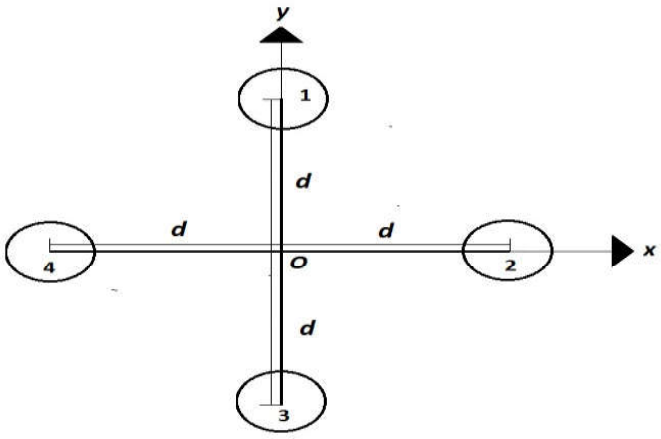

Figure 1. A schematic sketch of quadcopter

2.2 Equations of Motion

By using Newton`s second law of motion, the equation of translational motion along X axis would be

$$\sum_{i=1}^{4} F_i \left(\cos\theta_x \sin\theta_y \cos\theta_z - \sin\theta_x \sin\theta_z\right) = m\ddot{x} \tag{1}$$

and along Y axis would be

$$\sum_{i=1}^{4} F_i \left(\cos\theta_x \sin\theta_y \sin\theta_z - \sin\theta_x \cos\theta_z\right) = m\ddot{y} \tag{2}$$

while along Z axis is

$$\cos\theta_x \cos\theta_y \sum_{i=1}^{4} F_i - mg = m\ddot{z} \tag{3}$$

The equation of rotational motion around X axis is

$$(F_1 - F_3)d = I_x\ddot{\theta}_x - (I_y - I_z)\dot{\theta}_y\dot{\theta}_z \tag{4}$$

and around Y axis is (5)

$$(F_4 - F_2)d = I_y\ddot{\theta}_y - (I_z - I_x)\dot{\theta}_x\dot{\theta}_z \tag{5}$$

while around Z axis is (6)

$$0 = I_z\ddot{\theta}_z - (I_x - I_y)\dot{\theta}_x\dot{\theta}_y \tag{6}$$

In order to linearize and simplify the six equations above, several assumptions are made. By looking at equation (6), due to symmetry of quadcopter one can assume $I_x = I_y$ simplifying the equation (6) to be $I_z\ddot{\theta}_z = 0$ which means angular velocity around Z axis never changes. Since the initial conditions for $\dot{\theta}_z = \theta_z = 0$, then equations (4) and (5) can also be simplified further. Another assumption which has been considered is the small angle oscillation allowing us to approximate $\sin\theta \cong \theta$ and $\cos\theta \cong 1$ for values of $\theta \approx 0$. These assumptions simplify equations (1)-(6). The simplified equations of motion will be in terms of equation (7).

$$\begin{Bmatrix} \sum_{i=1}^{4} F_i(\theta_y) = m\ddot{x} \\ \sum_{i=1}^{4} F_i(-\theta_x) = m\ddot{y} \\ \sum_{i=1}^{4} F_i - mg = m\ddot{z} \\ (F_1 - F_3)d = I_x\ddot{\theta}_x \\ (F_4 - F_2)d = I_y\ddot{\theta}_y \\ I_z\ddot{\theta}_z = 0 \end{Bmatrix} \quad (7)$$

Now that the equations of motion are simplified, we try to obtain the state space model by considering the state vector $H = [H_1\ H_2\ H_3\ H_4\ H_5\ H_6\ H_7\ H_8\ H_9\ H_{10}\ H_{11}\ H_{12}]^T$ such that each of these states are defined in (8) .

$$\begin{Bmatrix} H_1 = x \rightarrow \dot{H}_1 = \dot{x} \\ H_2 = y \rightarrow \dot{H}_2 = \dot{y} \\ H_3 = z \rightarrow \dot{H}_3 = \dot{z} \\ H_4 = \dot{x} \rightarrow \dot{H}_4 = \ddot{x} \\ H_5 = \dot{y} \rightarrow \dot{H}_5 = \ddot{y} \\ H_6 = \dot{z} \rightarrow \dot{H}_6 = \ddot{z} \\ H_7 = \theta_x \rightarrow \dot{H}_7 = \dot{\theta}_x \\ H_8 = \theta_y \rightarrow \dot{H}_8 = \dot{\theta}_y \\ H_9 = \theta_z \rightarrow \dot{H}_9 = \dot{\theta}_z \\ H_{10} = \dot{\theta}_x \rightarrow \dot{H}_{10} = \ddot{\theta}_x \\ H_{11} = \dot{\theta}_y \rightarrow \dot{H}_{11} = \ddot{\theta}_y \\ H_{12} = \dot{\theta}_z \rightarrow \dot{H}_{12} = \ddot{\theta}_z \end{Bmatrix} \quad (8)$$

The equations of motion in (7) can be written in terms of states in (8) in matrix format presented in equation (9).

$$\begin{bmatrix}\dot{H}_1\\ \dot{H}_2\\ \dot{H}_3\\ \dot{H}_4\\ \dot{H}_5\\ \dot{H}_6\\ \dot{H}_7\\ \dot{H}_8\\ \dot{H}_9\\ \dot{H}_{10}\\ \dot{H}_{11}\\ \dot{H}_{12}\end{bmatrix} = \begin{bmatrix}0&0&0&1&0&0&0&0&0&0&0&0\\ 0&0&0&0&1&0&0&0&0&0&0&0\\ 0&0&0&0&0&1&0&0&0&0&0&0\\ 0&0&0&0&0&0&0&0&0&0&0&0\\ 0&0&0&0&0&0&0&0&0&0&0&0\\ 0&0&0&0&0&0&0&0&0&0&0&0\\ 0&0&0&0&0&0&0&0&0&1&0&0\\ 0&0&0&0&0&0&0&0&0&0&1&0\\ 0&0&0&0&0&0&0&0&0&0&0&1\\ 0&0&0&0&0&0&0&0&0&0&0&0\\ 0&0&0&0&0&0&0&0&0&0&0&0\\ 0&0&0&0&0&0&0&0&0&0&0&0\end{bmatrix}\begin{bmatrix}H_1\\ H_2\\ H_3\\ H_4\\ H_5\\ H_6\\ H_7\\ H_8\\ H_9\\ H_{10}\\ H_{11}\\ H_{12}\end{bmatrix} + \begin{bmatrix}0&0&0&0\\ 0&0&0&0\\ 0&0&0&0\\ H_8/m&0&0&0\\ -H_7/m&0&0&0\\ 1&0&0&-1\\ 0&0&0&0\\ 0&0&0&0\\ 0&0&0&0\\ 0&0&d/I_x&0\\ 0&d/I_y&0&0\\ 0&0&0&0\end{bmatrix}\begin{bmatrix}\sum_{i=1}^{4} F_i\\ F_4-F_2\\ F_1-F_3\\ mg\end{bmatrix}$$

(9)

Table 2.1 provides a list and definition of all geometrical and non-geometrical parameters used in the modeling of dynamic of quadcopter based on a custom design.

Variable (Unit)	Definition	Value
m (Kg)	mass	1
d (m)	Perpendicular distance to origin	0.17
D (N/rad^2)	Thrust Coefficient	10^{-6}
I_x (Kg.m^2)	Inertia around X	0.5
I_y (Kg.m^2)	Inertia around Y	0.5
R (m)	Radius of Blades	0.08

Table 1. Dynamic Parameters of Quadcopter

2.3 Actuators

Each brushless DC motor is equipped with a blade and, depending of the sign of voltage terminals, they can either rotate counter-clockwise or clockwise. The blades are designed in a way that if motors rotate counter-clockwise, they generate force upward and if they rotate clockwise, they generate force downward.

Reference [11] uses the fact that the force generated by motors are a parabolic function of angular velocity as it is shown in equation (10).

$$F_{nominal} = sgn(\omega_i) D \omega_i^2 \qquad i = 1,2,3,4 \tag{10}$$

where *D* is the thrust coefficient which is a function of air density and the geometry of blades while ω_i is the angular velocity of the motors. It is very unrealistic to assume that motors can rotate at any arbitrary speed so that they can produce any amount of force. In a practical example, the battery has a given charge which limits the amount of energy that can be delivered to motors in a specified interval of time. Therefore, instead of designing an electrical circuit and conducting an analysis to determine the power, we simply set a limit as a constraint on the

maximum amount of force that can be generated by motors, as shown in equation (1) for a 10 (N) limit.

$$\left|F_{nominal_i}\right| \leq 10\ [N] \tag{11}$$

An additional constraint is that the polarity of the voltage applied to the motors cannot change. Hence, the direction of rotation is only counter clockwise to generate force only upward as shown in equation (12).

$$\omega_i \geq 0\ \ (\frac{rad}{s}) \tag{12}$$

The brushless DC motors have a gain of $K_v = 2400\ (\frac{rpm}{V})$ and a time constant of $\tau_m = 0.1\ (\mathrm{s})$. Therefore, the transfer function of each motor can be written as equation (13)

$$G_m(s) = \frac{\omega(s)}{v(s)} = \frac{2400}{\tau_m s+1} \tag{13}$$

where $\omega(s)$ is the Laplace transformation of angular velocity of motors and $v(s)$ is the input voltage applied to the motors.

2.4 Ground Effect

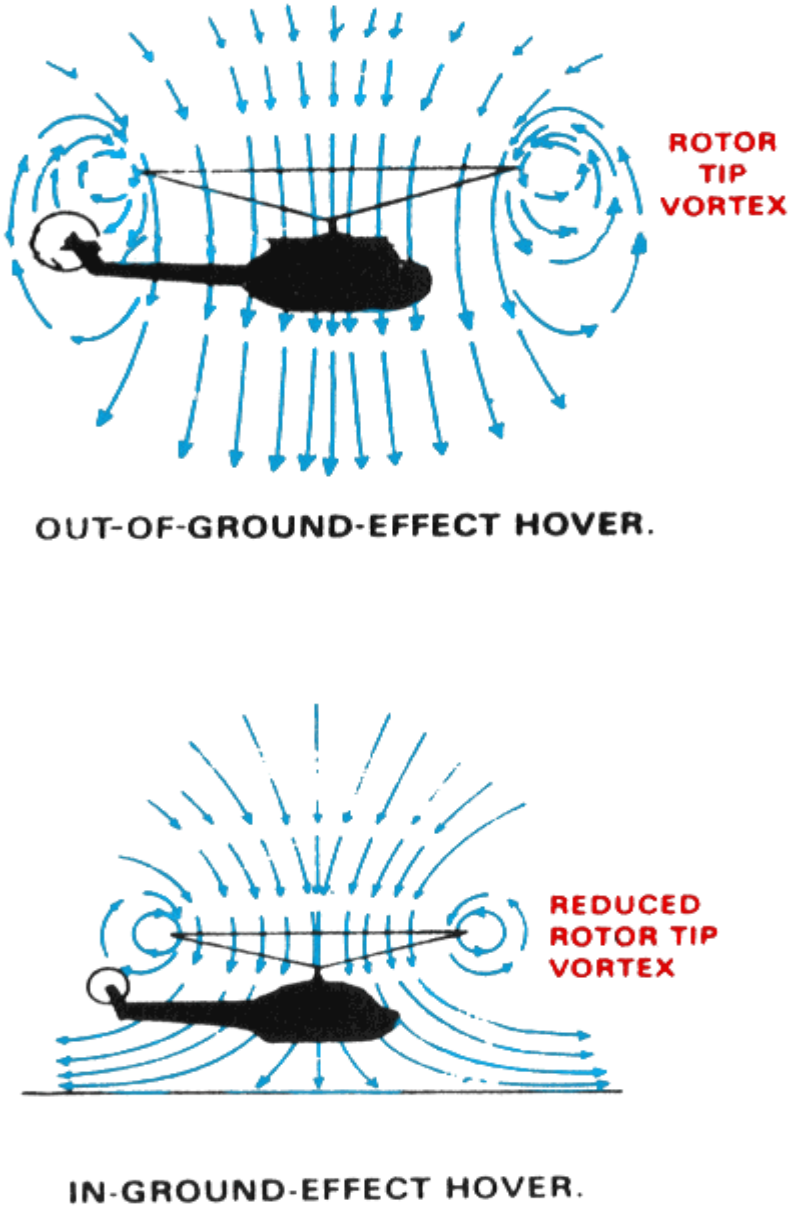

Figure 2. Airflow Field with and without Ground Effect for Helicopter [13]

When the blades of the motor rotate, an air flow is created downward as a result of the reaction force exerted on air by blades. When quadcopter is operating at low altitudes, the ground acts as an obstacle that hinders the movement of the airflow. As a result, the pressure near the ground will increase causing the quadcopter to lift easier. Such observation was

originally done in [12] for a helicopter and is adopted for the quadcopter. Figure 2 portraits the effect of ground for a helicopter.

Later, Cheeseman and Bennette [9] presented a model based on an experimental study that described how the force, generated by a helicopter`s motor, varies as the altitude changes. Equation (19) is the Cheeseman model which is independent from specifications of a particular helicopter. Therefore, by assuming that quadcopter`s motors are sufficiently far away from one another, such that there will not be any cross-interaction of airflows, equation (14) can be applicable for each motor individually.

$$F_i = F_{nominal_i}\left(\frac{1}{1-\frac{R^2}{16z^2}}\right) \tag{14}$$

where *R* is the radius of blades and *z* is the altitude. Equation (14) also indicates that for large values of *z,* which means the quadcopter is far away from the ground, the actual thrust force is equal to nominal force. However, for small values of altitude including zero, the fraction in equation (14) can become infinite and, hence, Cheeseman and Bennette added that equation (14) is applicable whenever the inequality (15) is maintained

$$\frac{Z}{R} \leq 0.6 \tag{15}$$

otherwise a large amount of error is introduced.

Equation (14) shows that the thrust force affected by ground is a nonlinear function of altitude. Therefore, by substituting (10) to (14) and then substituting to linearized equations of motion in (7), the entire system becomes nonlinear again.

Chapter 3

Fractional Controller

This chapter explains the type of controller which has been used for controlling the quadcopter. As it was stated earlier, the term “fractional” refers to mathematical concept of fractional calculus. Therefore, it is essential to introduce fractional calculus initially. In the second step, the methodology and the assumptions of how fractional calculus is being applied to a conventional PID controller, in order to obtain a fractional controller, is presented. Later, the numerical method of Oustaloup filter [14] is presented and how it has been used. Here is an outline of this chapter

- Introducing the concept of fractional calculus
- Numerical methods and why they are needed
- Introducing fractional PID controllers and how they are connected based on designer`s choice
- Method of tuning fractional PID controllers
- Setting a criterion that fractional and integer PID become comparable.

3.1 Fractional Calculus

The concept of fractional calculus was presented first by Leibniz and later it was developed by Liouville [15]. It states that continuous and differentiable functions such as *f(x)* can be differentiated not only to an integer order but also to a non-integer order. Additionally, differentiating a function to a negative order is actually integrating.

Such definition of differentiating operations extends and covers both concepts of differentiation and integration. The general notion used to show fractional differentiations is

$$D^{\gamma}[f(x)] = \begin{cases} \gamma \in \mathbb{N} & \frac{d^{\gamma}(f(x))}{dx^{\gamma}} \\ \gamma = 0 & f(x) \\ \gamma \in -\mathbb{N} & \int \underbrace{\cdots\cdots\cdots}_{|\gamma| \; times} \int f(x)\,(dx)^{-\gamma} \\ \gamma \in \mathbb{R} - \mathbb{Z} & D^{\gamma}[f(x)] \end{cases} \tag{16}$$

where γ is the order of differentiation, in this case a real number representing the fractional order.

There have been several definitions for fractional derivatives in time domain such as Grunwald-Letnikov [17], Riemann-Liouville [20] and Caputo [21], all of which have been presented in detail by Pudlubny [16].

The Grunwald-Letnikov definition in equation (17) is obtained by writing the integer order definition of standard derivatives in a consecutive form such that a general pattern is derived, then it tries to generalize the obtained pattern by changing domain of the order of differentiation from an integer to a real number.

$$D^{\gamma}[f(x)] = \lim_{h \to 0} \frac{1}{h^{\gamma}} \sum_{r=0}^{\infty} (-1)^{r} \binom{\gamma}{r} f(x + (\gamma - r)h) \tag{17}$$

Reimann-Liouville approach initially defines a fractional integration as in (18). In order to define fractional differentiation in equation (19), the function *f(x)* is initially integrated to the order of the difference between *γ* and the nearest integer greater than *γ*. Then, it is differentiated using integer definition as many times as the nearest integer greater than *γ* such as *n*+1. For example, in order to differentiate to the order of *γ* =2.6, the Reimann-Liouville presentation performs a fractional integration to the order of 0.4, then the resulting function is differentiated 3 times using integer order definitions.

$$ {}_aD^{\gamma}[f(x)] = \frac{1}{\Gamma(-\gamma)} \int_a^x (x-\tau)^{-\gamma-1} f(\tau) d\tau \qquad \gamma < 0 \tag{18} $$

Where *Γ* is the Gamma function.

$$ D^{\gamma}[f(x)] = (\frac{d}{dx})^{n+1} * {}_aD^{\gamma-n}[f(x)] \qquad m < \gamma < m+1 \tag{19} $$

Unlike Reiman-Liouville, Caputo`s definition tries to integrate function *f(x)* fractionally which had been differentiated *n* times (20).

$$ D^{\gamma}_{Caputo}[f(x)] = \frac{1}{\Gamma(\gamma-n)} \int_a^t \frac{f^{(n)}(\tau)d\tau}{(t-\tau)^{\gamma+1-n}} \quad n-1 < \gamma < n \tag{20} $$

Putting aside the mathematical formulation of fractional calculus temporarily, the realization of fractional calculus is relatively more complicated compared to integer calculus due to the fact that the values returned by fractional derivatives of a function, are not indicators of a geometrical and local property of a point in the domain of the function. However, they may be indicators of the behavior of the function globally. For instance, it is known that for an arbitrary point (such as x_0) on a function such as *f(x),* the first order derivative of *f(x)* shows the slope or steepness at that point (x_0) or the second derivative determines the convexity or concavity of function *f(x).* Since these geometrical properties belong to one point like x_0, they are considered to be local. However, this is not the case for fractional derivatives. let`s take to the consideration that *f(x) = sin(x)*. Therefore, by taking integer derivatives, we can obtain the pattern below

$$ \frac{d(f(x))}{dx} = \cos(x) = \sin\left(x + \frac{\pi}{2}\right) \tag{21} $$

$$ \frac{d^2(f(x))}{dx^2} = -\sin(x) = \sin(x+\pi) \tag{22} $$

$$ \frac{d^n(f(x))}{dx^n} = \sin(x + n\frac{\pi}{2}) \tag{23} $$

The pattern in (23) shows that every time the sine function is differentiated, it is shifted by 90 degrees or $\frac{\pi}{2}$. In [22] it has been shown that the pattern in (23) is correct even if the order of differentiation *n* is a real number. Therefore, by taking half derivative, the sine function shifts

by 45 degrees or $\frac{\pi}{4}$. While $D^{\frac{1}{2}}[f(x) = \sin(x)]$ at point x_0 may not return any value that can be interpreted as a geometrical property. Thus, fractional derivatives can shift the entire sine function to a specified angle. It should be noted that the angle in which the function has shifted is not limited to a point, so it is not local. The shifting interpretation is only true for sine function. Now by using this interpretation and Fourier series, the geometrical analysis of fractional derivatives of functions can be taken one step further. Since Fourier series approximates a periodic function within a specified interval using a series of sine and cosine functions, applying fractional derivative shifts each of the sine and cosine terms of Fourier series as well.

In Pudlubny`s work, the Laplace transformation of Reimann-Liouville and Caputo`s derivatives are documented. The only difference, in representation of Laplace transformation of these two derivatives, is the way that initial conditions are presented. However, if all initial conditions are considered zero, then the Laplace representation of both becomes equal as in (24).

$$L\big[D^{\gamma}[f(x)]\big] = s^{\gamma} * F(s) \tag{24}$$

From (24), one can expect that whenever fractional derivatives exist within a differential equation, the term s^{γ} will appear if Laplace transformation is taken.

3.2 Numerical Methods

Finding the analytical solution of fractional order differential equations is relatively more complicated compared to integer order differential equations. Therefore, numerical tools become handy in order to implement the fractionality on a system. There have been multiple numerical methods to approximate fractional derivative operation such as truncated Grunwald-Letnikov method [17] and Pudlubny`s matrix approach [16] and Oustaloup filter [14].

In this thesis the Oustaloup filter has been used since it is relatively easy to implement it on MATLAB™ SIMULNK environment and it provides sufficient accuracy, as shown in [18]. Additionally, Oustaloup filter can be presented in a way that can be compared with the standard first order filter of a conventional PID controller which is explained later.

The transfer function of the Oustaloup Filter is shown in (25) and its parameters are presented in (26), (27) and (28). Oustaloup filter approximates the variable *s* which has been risen to the power of *γ* for the range of frequencies of (ω_b, ω_h). $2N+1$ is the order of the filter. For the values of $N \geq 2$ the results are sufficiently accurate [18]. There are $2N+1$ poles and zeros in Oustaloup filter which are $-\omega_i$ and $-\omega_i'$ respectively. Due to its form of representation, it is easy to use MATLAB™ zero-pole transfer function in SIMULINK to create Oustaloup block for a given *γ*.

$$s^{\gamma} \cong K * \prod_{i=-N}^{N} \frac{(s+\omega_i')}{(s+\omega_i)} \tag{25}$$

where

$$K = \omega_h^{\gamma} \tag{26}$$

and

$$\omega_i' = \omega_b \left(\frac{\omega_h}{\omega_b}\right)^{\frac{i+N+0.5-0.5\gamma}{2N+1}} \tag{27}$$

and

$$\omega_i = \omega_b \left(\frac{\omega_h}{\omega_b}\right)^{\frac{i+N+0.5+0.5\gamma}{2N+1}} \tag{28}$$

3.3 Implementation of Fractional Controller

The transfer function of an ideal conventional PID controller is given in (29) and its characteristic equation in time domain is in (30)

$$G_{ic}(s) = K_p + K_d * s + \frac{K_i}{s} \tag{29}$$

$$u(t) = K_p * e(t) + K_d * \dot{e}(t) + K_i \int e(t)dt \tag{30}$$

where K_p, K_d and K_i are the proportional, derivative and integral gains in the order they have been mentioned. In order to generalize the standard PID controller transfer function, two main questions can be posed. First one is why is not possible to use fractional derivatives or fractional integration instead of integer order derivative and integer order integration. The second question is why there has to be only one fractional derivative and only one fractional integration. These issues can be presented in the form of the transfer function for multiple fractional derivatives and integrations as in (31)

$$G_{fc}(s) = K_p + \sum_{a=1}^{A} K_{d_a} * s^{\gamma_a} + \sum_{b=1}^{B} K_{i_b} * \frac{1}{s^{\mu_b}} \tag{31}$$

where K_{d_a}, K_{i_b} represents various derivative gains and various integral gains that can exist within the structure of the controller as γ_a, μ_b which are the orders of fractional differentiation and integration respectively. Analyzing the fractional order controller in this comprehensive form of transfer function (31) can be a topic of interest for future studies.

The focus of this study has been put only on derivative part of the transfer function in (29) and (30) and that replacing integer order derivative with a fractional order derivative as in (32) and (33). Although it is logical and useful to investigate the fractional integration as well, it has not been considered since the effect of solely derivative part on the performance of system is of interest here.

$$G_{fc}(s) = K_p + K_d * s^{\gamma} + \frac{K_i}{s} \tag{32}$$

The time domain equation for the fractional order controller is

$$u(t) = K_p * e(t) + K_d * D^{\gamma}[e(t)] + K_i \int e(t)dt \tag{33}$$

In order to control the quadcopter, three fractional controllers have been used. One for controlling the altitude with the command $u_z(t)$, another for controlling the pitch angle with the command $u_{\theta_x}(t)$ and the third one for adjusting roll angle with the command $u_{\theta_y}(t)$

By looking at the equations of motion (7), we can see that controllers are responsible to alter F_1, F_2, F_3, F_4 such that the desired output is achieved. For instance, a closer look shows that in order to generate a positive torque around *x* axis, either F_1 can be increased or F_3 can be decreased or a combination of both since the system is over-actuated. Therefore, it is up to designer to make the decision how one may proceed. In this thesis, we have proceeded as it is presented below.

When a reference input is set for altitude, it is expected that all four motors respond simultaneously and operate in a synchronous manner. Therefore, the signal generated by the controller responsible for adjusting altitude, must be fed to all four motors, equally. In the next step, when a reference input is set for pitch angle, the signal generated by controller responsible for adjusting pitch angle can be added to the input of first motor and subtracted from third motor. Similarly, for controlling roll angle, the output signal of controller responsible for controlling roll angle can be added to the input signal of forth motor and subtracted from second motor. Motors are numbered based on figure 1.

In another word, if $v_1(t)$, $v_2(t)$, $v_3(t)$ and $v_4(t)$ are voltage commands to the input signals of motors, then equations (34)-(37) show the input signals of motors as a function of output signals generated by three controllers.

$$v_1(t) = u_z(t) + u_{\theta_x}(t) + C \quad (34)$$

$$v_2(t) = u_z(t) - u_{\theta_y}(t) + C \quad (35)$$

$$v_3(t) = u_z(t) - u_{\theta_x}(t) + C \quad (36)$$

$$v_4(t) = u_z(t) + u_{\theta_y}(t) + C \quad (37)$$

Where C is the constant calibrator providing a constant voltage to motors. Its role is to account for the effect of weight.

3.4 Tuning Method

By looking at equations (7), we can see the lack of damping element and spring-like element or stiffness within the equations of motion of quadcopter. Therefore, neither open loop nor closed loop methods of standard Ziegler-Nichols are applicable.

The open loop method of Ziegler-Nichols requires the output of system, without controller, to be a *S*-shape curve for a unit step input. Then, parameters of controller can be determined based on the graphical plot of output. However, the quadcopter acts as a second order system without damping and spring element. Hence, its output is a parabolic shape curve rather than a *S*-shape curve.

The closed loop Ziegler-Nichols method requires that after all gains are being set to zero, initially, then the proportional gain must be increased to the level that the output of system becomes critically oscillatory. The proportional gain correspondent to this condition is called ultimate gain K_u and the desired proportional, derivative and integral gain are determined based on the value of ultimate gain. However, the quadcopter, which does not have damping element or friction, has a critically oscillatory response from the beginning. Hence for preserving stability, the derivative gain must be non-zero and positive which violates the first assumption of closed loop Ziegler-Nichols method.

In this study, the criterion which is used for tuning the controllers is *Integral of Absolute Error (IAE)* in (38)

$$IAE = \int_0^\infty |e(t)|dt \tag{38}$$

The *IAE* calculates the area under the curve of error function which is the area between the reference input and system output, if these two are plotted on one graph. A small value of *IAE* indicates that the output is following the reference input closely. It must be mentioned that the integration of error function in equation (38) is done over a time domain from 0 to ∞, theoretically. However, in practice we cannot conduct a simulation from time 0 which is initiation time to ∞ which is eternity. In order to limit the time domain of this integration, we assume that there exists a time margin as *T,* such that the output has to settle within *T* (sec) otherwise the results are not acceptable. Therefore, the equation (38) can be rewritten as

$$IAE = \int_0^\infty |e(t)|dt = \int_0^T |e(t)|dt + \int_T^\infty |e(t)|dt \tag{39}$$

Since the output has settled within *T* seconds, the error will definitely remain 0 after *T* (sec). Consequently, integration after *T* (sec) will be also zero which leaves us with the finite integration part.

The way that this thesis has approached the tuning of fractional PID controller is that initially the minimum value of *IAE* is found when a conventional PID controller is considered. Therefore, proportional and integral and derivative gains are obtained. Then, by preserving the gains and changing the value of gamma, it is checked to see whether *IAE* can be further minimized or not.

Although there are other parameters that can determine the performance of the controller such as Integral of Squared Error *(ISE)* in equation (40) or Integral of Time-Weighted Error *(ITWE)* in equation (41), The integral of absolute error has been chosen as the main criterion since it gives a fair share to all parts of the error function while the other criteria are slightly biased.

$$ISE = \int_0^\infty e^2(t)dt \tag{40}$$

$$ITWE = \int_0^\infty t|e(t)|dt \tag{41}$$

For instance, *ISE* signifies the places where error is large while relatively ignores the places where error is small due to the fact that the error has been risen to power two. Therefore, it is useful to consider *ISE* when peak value or overshoot have high priority rather than settling time. Additionally, *ITWE* ignores the error at the beginning of the response since *t* has a very small value while it signifies the effect of error in later times. *ITWE* can be used efficiently to determine the performance of controller if settling time is a priority.

3.5 Standard Filter Vs. Oustaloup Filter

In MATLAB™ SIMULINK, the derivative part of conventional PID controller block is combined with a first order filter to process signals. The transfer function of this PID controller is given in (42) MATLAB™ SIMULINK as [19]

$$G_{MATLAB_{PID}}(s) = K_p + \frac{K_i}{s} + K_d \frac{Ns}{s+N} \tag{42}$$

where the *N* is the filter coefficient, and its value has an impact on the accuracy of output. The larger *N* is, the more accurate results are.

On the other side, the fractional order controller uses Oustaloup filter in its derivative part and its parameters also have an impact on the accuracy of result. Therefore, the question may raise “how the accuracy of both filters could have been effective in obtaining final results?”. For example, if fractional order controller outperforms the integer order controller, how can one make sure that these results are obtained because of the fractionality of controller and not better accuracy of Oustaloup filter?

In order to answer the question above, let us consider Oustaloup filter and its parameters mentioned previously. By expanding the product form of Oustaloup filter in (25) we get

$$s^{\gamma} \cong K * \frac{(s+\omega'_{-N})(s+\omega'_{-N+1})\cdots(s+\omega'_{N-1})(s+\omega'_{N})}{(s+\omega_{-N})(s+\omega_{-N+1})\cdots(s+\omega_{N-1})(s+\omega_{N})} \tag{43}$$

By inserting *γ*=1 in (26), (27) and (28) we get

$$K = \omega_h^1 \quad and \quad \omega_i = \omega'_{i+1} \qquad i = -N, -N+1, \ldots, N-1 \tag{44}$$

Equation (44) shows that when *γ*=1, pole-zero cancelation occurs in Oustaloup filter and what remains, regardless of the order of the Oustaloup filter, is always equation (45)

$$s^1 \cong \omega_h * \frac{(s+\omega'_{-N})}{(s+\omega_N)} \tag{45}$$

For -*N* into (27) and *N* into (28), the equation (45) can be rewritten as (46)

$$s \cong \omega_h * \frac{(s+\omega_b)}{(s+\omega_h)} = \frac{s\omega_h+\omega_b\omega_h}{s+\omega_h} \tag{46}$$

By assuming that ω_b is small enough such that the term $\omega_b\omega_h \cong 0$, then equation (46) can be rewritten as (47)

$$s \cong \frac{s\omega_h}{s+\omega_h} \tag{47}$$

Comparing ω_h in (47) to *N* in derivative part of (42), it can be observed that they have the same role and effect. This allows us to establish this criterion that whenever a conventional PID controller equipped with a first order filter, with filter coefficient N, is considered, the Oustaloup filter has to be designed with $\omega_h = N$ and $\omega_b \cong 0$ so that whenever the Oustaloup filter differentiates to first order, it provides the same performance as a standard first order filter.

3.6 Summary

The block diagram in Figure 3 shows a brief summary of the entire system and how each of its elements are connected to one another.

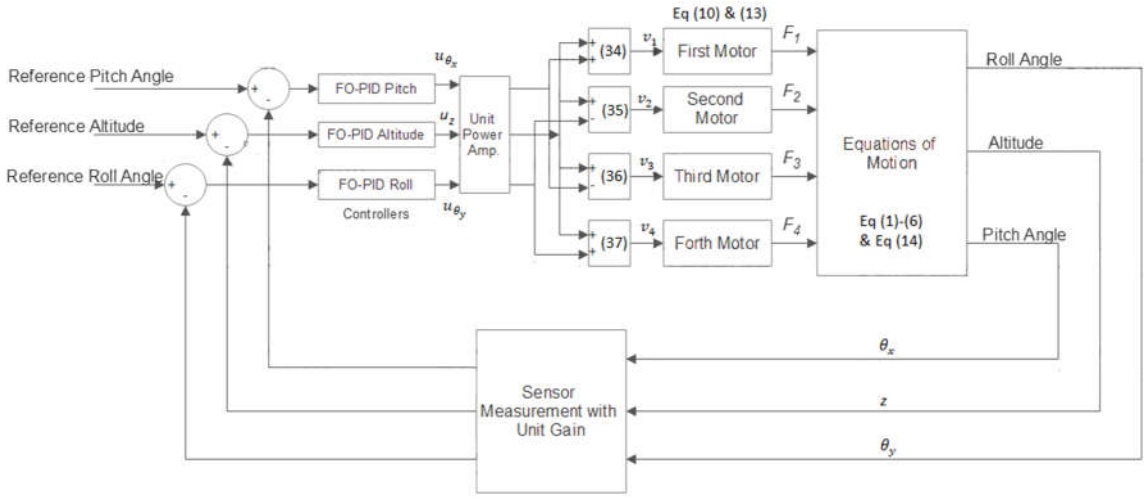

Figure 3. Block Diagram of Entire System

When reference inputs are given by operator or pilot, sensors will measure the current output. Therefore, the error signals are generated and fed to each controller respectively. Then the output of controllers is amplified so that they can deliver power to motors. However, these amplified signals are delivered to motors based on equations (34)-(37). One should keep in mind that the amplification and the process of equations (34)-(37) are interchangeable. Within the blocks of motors, now that motors receive signals, they start rotating with an angular velocity of ω_i based on transfer function in equation (13) and consequently generation force $F_{nominal_i}$ based on equation (10). Inside equations of motion block, the nominal forces are affected by ground nonlinearly based on the value of altitude, so that the actual forces are exerted on quadcopter using linearized equations of motion obtained in (7).

Chapter 4

Results

As it was mentioned when presenting equations (11) and (12), there are constraints on actuators such that motors can only generate forces limited upward. If a quadcopter is set to elevate from height $z_1$to z_2, the profile of generated forces will differ from the case where a quadcopter is set to descend form height $z_2$to z_1. Hence, the best set of gains and γ obtained for ascending motion, using *IAE* for fractional PID controller, are not necessarily the best gains for descending motion. Therefore, descending is evaluated separately from ascending. Due to the symmetry of quadcopter it is expected that the results of rolling and pitching motion will be similar.

4.1 Ascending

The ascending motion has been considered when the quadcopter is positioned initially on the ground and it is intended to raise to altitude of one meter.

Table 4.1 shows optimal values of the gains which belongs to the minimum value of *IAE* for conventional PID controllers and the value of *IAE* itself.

IAE (m.s)	**Gains**		
	K_p	K_d	K_i
0.375	1.3	0.65	4.5

Table 2. Optimal gains of IO-PID controller for ascending

Figure 4 presents the results of the quadcopter for ascending motion when equipped with a conventional PID controller, using the gains provided in Table 4.1, while Figure 5 presents the case of ascending motion of a quadcopter equipped with a fractional order controller, using gains in Table 4.1. However, by examining different values of γ, it was observed that $\gamma = 1.02$ has caused the value of *IAE* to be further minimized which became 0.265 [m.s]. Figure 5 also demonstrated that fractional order controller has reduced the initial overshoot by 4% since the peak value of altitude in Fig.3 is 1.08 (m) and the peak value of altitude in Figure 5 is 1.04 (m).

By considering the Equation (14), it is expected that ground effect only increases the net lifting force at the beginning of transient part where quadcopter is in low altitude regardless of the type of the controller.

Figure 6 shows the variation of integral of absolute error with respect to gamma considering the *T* in equation (39) to be 10 (sec). The values of gamma less than 0.8 or greater than 1.16 has caused the output to either diverge, which returns a value if infinity for *IAE*, or not settle within 10 seconds, which is not acceptable and returns relatively large values. It can be seen that the minimum value of integral of absolute error is obtained when $\gamma = 1.02$. One may argue that such value of gamma is close to 1, which is the characteristic of a conventional PID controller, resulting in fractional controller not being necessary however, it must be taken into the account that this small difference has reduced the *IAE* by 29.3%. Since the output is oscillatory, it is probable that it can be described with Fourier series and applying the shifting interpretation on its terms, can justify the considerable change of *IAE* with respect to small change in gamma.

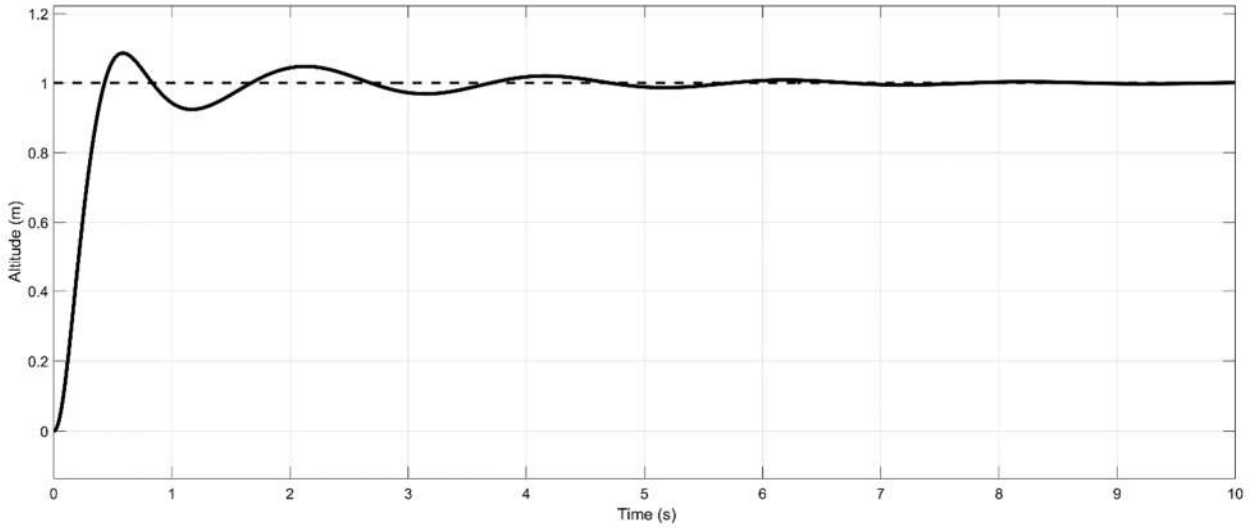

Figure 4. Altitude of quadcopter for ascending with IO-PID

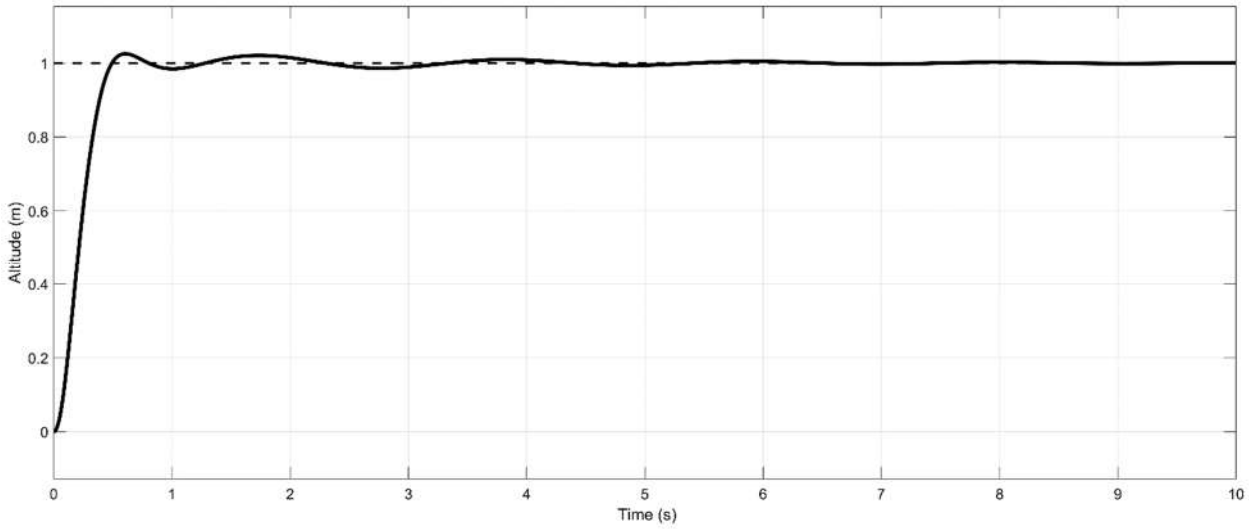

Figure 5. Altitude of quadcopter for ascending with FO-PID

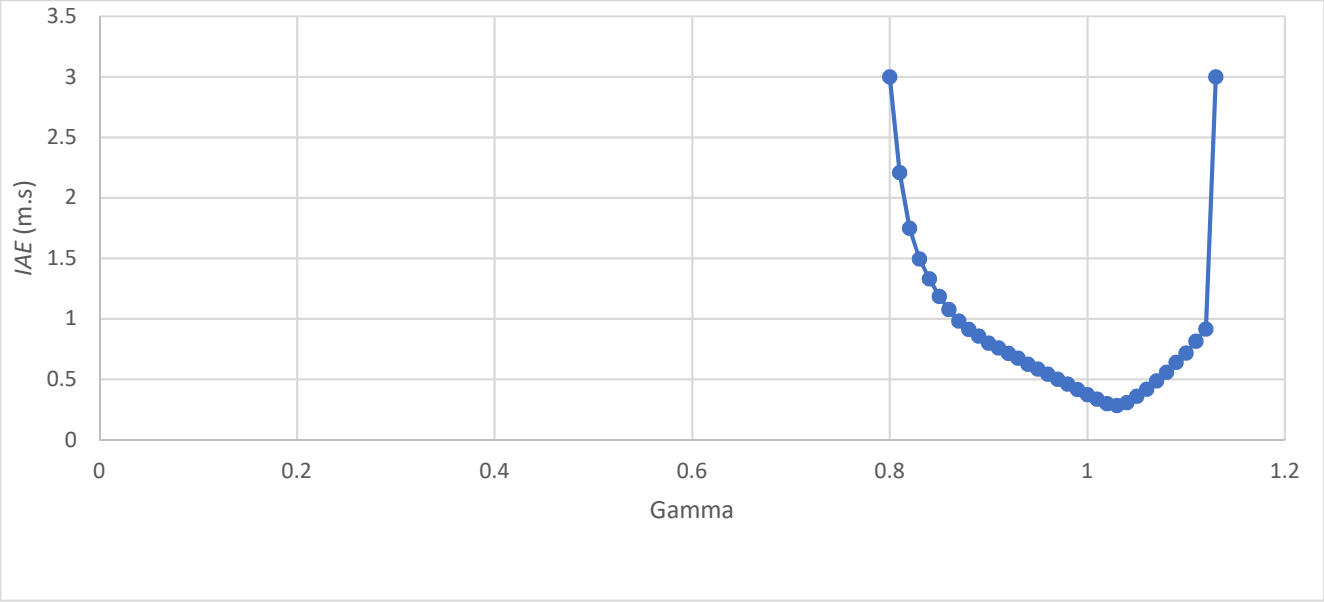

Figure 6. Gamma vs IAE for Ascending Motion

4.2 Descending Motion

In order to simulate descending motion, additional considerations have been taken into account. For instance, due to existence of overshoot, the final point or the reference input for descending motion has been set to 0.2 (m) instead of 0 (m) because descending to altitude less than zero means hitting the ground which is an undesirable condition while the initial point or the initial condition of altitude is 1 (m).

Table 4.2 provides the optimal gains for IO-PID controller with the minimized value of *IAE* when the quadcopter descends.

IAE (m.s)	**Gains**		
	K_p	K_d	K_i
0.624	9.4	3	5.5

Table 3. Optimal gains of IO-PID for descending

Figure 7 shows the altitude for descending motion of quadcopter equipped with integer order PID while Figure 8 and Figure 9 show the altitude of quadcopter when fractional order controller with two different values have been used. It must be mentioned that although there had been other sets of gains which provided smaller values of *IAE*, they have been ignored because of the large overshoot bringing quadcopter to altitude less than zero.

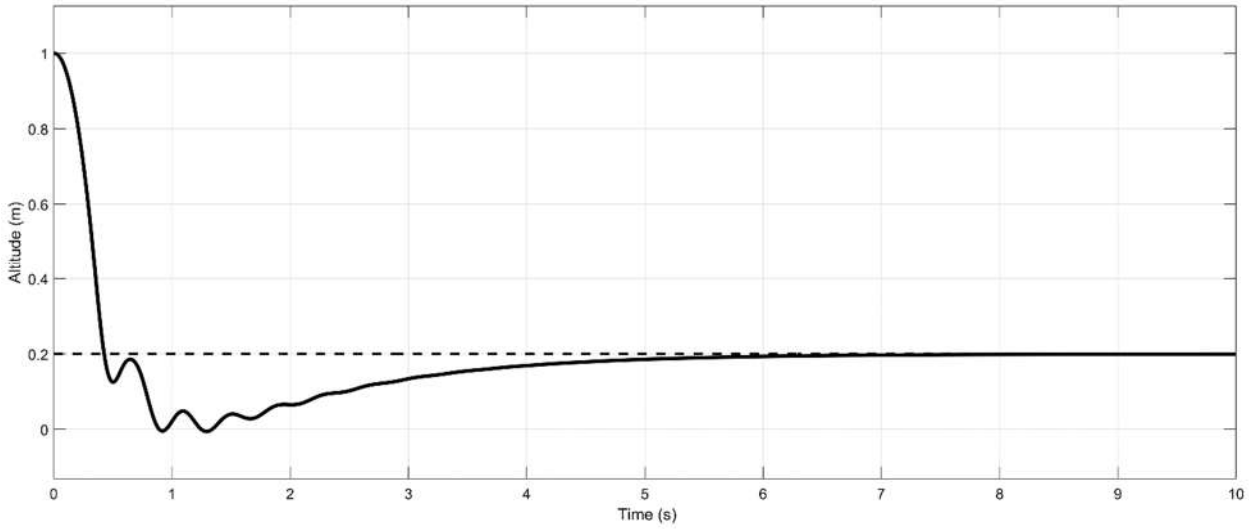

Figure 7. Altitude of quadcopter when descending with IO-PID controller

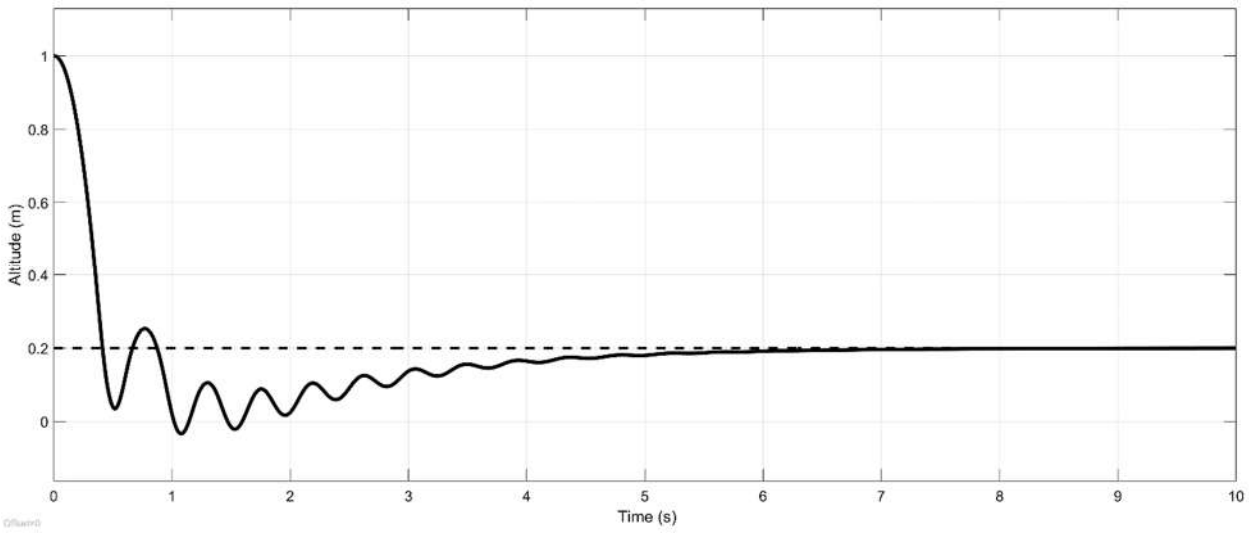

Figure 8. Altitude of quadcopter when descending with FO-PID controller with γ =0.94

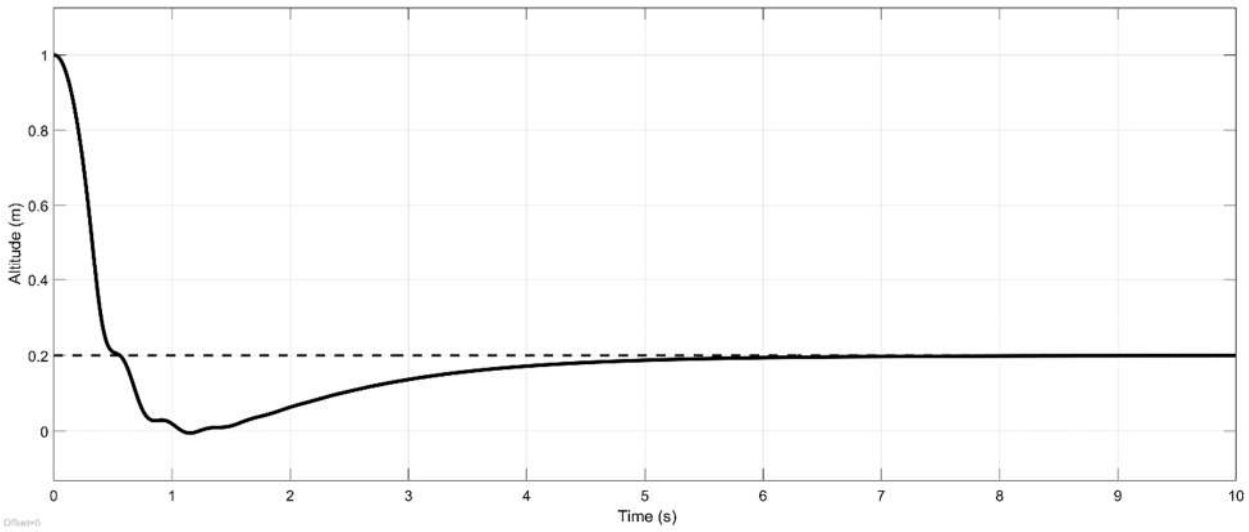

Figure 9. Altitude of quadcopter when descending with FO-PID controller with γ=1.06

By changing the values of γ, two interesting results were observed. One is shown in Figure 8 when $\gamma = 0.94$ which allowed the value of *IAE* to be further minimized slightly to 0.615 (m.s) although it may seem that the amplitude of oscillations has increased a bit.

The other result is depicted in Figure 9. It shows when $\gamma = 1.06$ the value of *IAE* is not further minimized, nor increased but remained same as given in table 4.2. However, the oscillation has reduced significantly and the response is smooth. It seems that increasing γ to more than one has increased the overall damping of system such that it could not have been done simply by increasing K_d. Otherwise, a better set of gains would have been obtained when IO-PID was being tuned. However, this statement must be further validated through mathematical verification or experimental testing.

Figure 10 is the plot of integral of absolute error with respect to gamma. Applying the same *T* = 10(sec), results in values of integral of absolute error not being acceptable for $\gamma \leq 0.8$ or $\gamma \geq 1.2$. Additionally, $1.1 < \gamma \leq 1.2$ has a large overshoot such that brings the quadcopter to altitude less than zero which is not acceptable. However, it is still depicted in Figure 10.

The effect of the ground can be more significant in this motion since as the quadcopter approaches the ground, based on (14) the lifting force generated by motors increases parabolically.

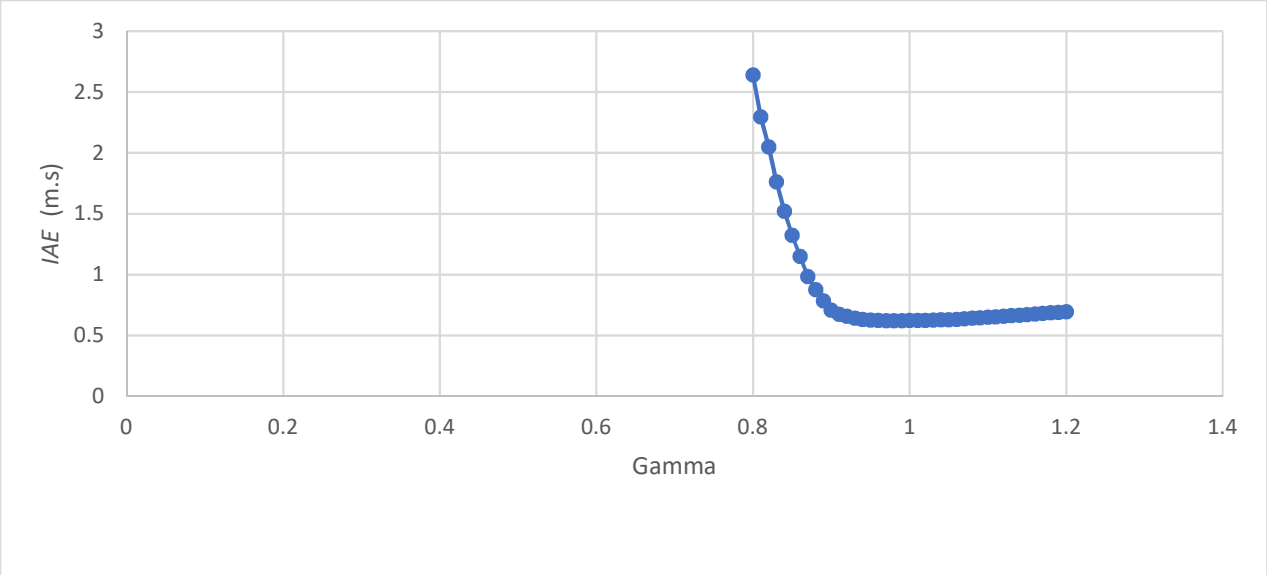

Figure 10. Gamma vs IAE for Descending Motion

4.3 Pitching Motion

As it was stated earlier, the performance of a quadcopter is expected to be similar for pitching and rolling motion due to the geometrical symmetry of quadcopter. The reference input considered for pitching motion was chosen ±0.35 (rad) or ±20˚ (degree). It is introduced after 3 seconds assuming that quadcopter maintains its altitude which is 1 meter. Therefore, the effect of ground is negligible at this altitude.

Table 4 shows the optimal gains obtained of method integral o absolute error and the *IAE* itself for pitching motion.

IAE (rad.s)	**Gains**		
	K_p	K_d	K_i
0.234	7.4	3	6.3

Table 4. Optimal gains of IO-PID for pitching motion

Figure 11 shows the response of a quadcopter equipped with integer order PID controller using the gains in Table 4.3 while Figure 12 shows the response of quadcopter for pitching when fractional order controller is used with the same gains in Table 4.3 but with $\gamma = 1.15$.

It must be mentioned that no value of γ was able to reduce the value of *IAE* any further. However, when $\gamma = 1.15$, the response has less overshoot while the value of γ is preserved. Such conclusion can also be derived from looking at Figure 13 which shows the *IAE* as gamma varies. The value of *IAE* for pitching motion does not change significantly when $0.8 \leq \gamma \leq 1.2$.

It may seem that the response of the integer order controller has an unacceptable amount of overshoot with the optimal obtained gains. However, it should be taken into consideration that the method of integral of absolute error does not focuses on overshoot. It is merely a general criterion to compare and evaluate performances. In this case, the reduction of overshoot has been achieved at the expense of a bit longer settling time.

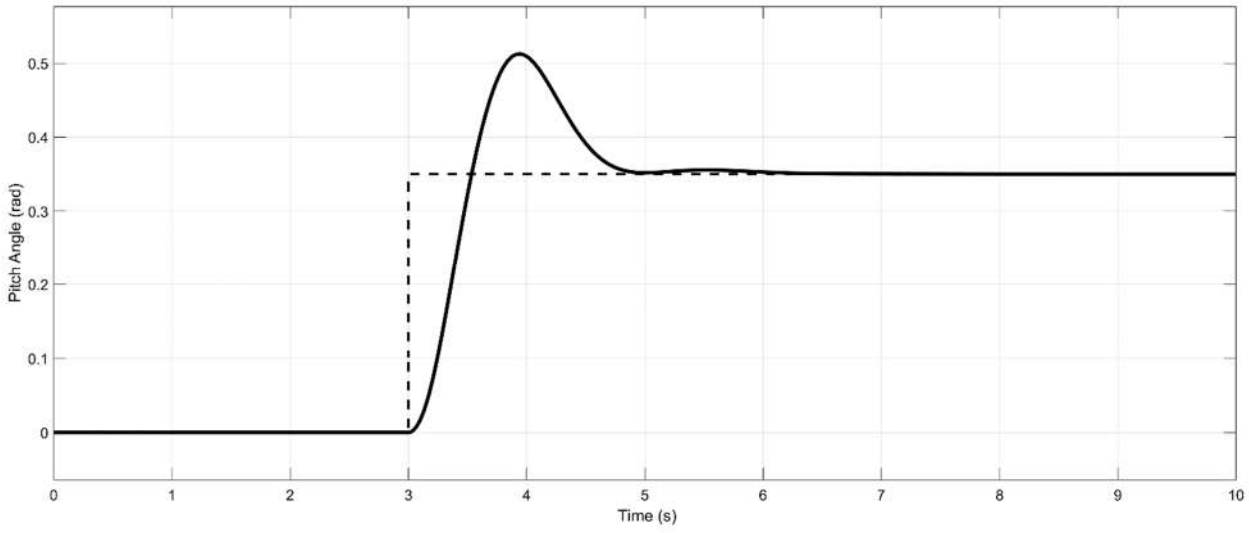

Figure 11. Pitch angle of quadcopter with IO-PID

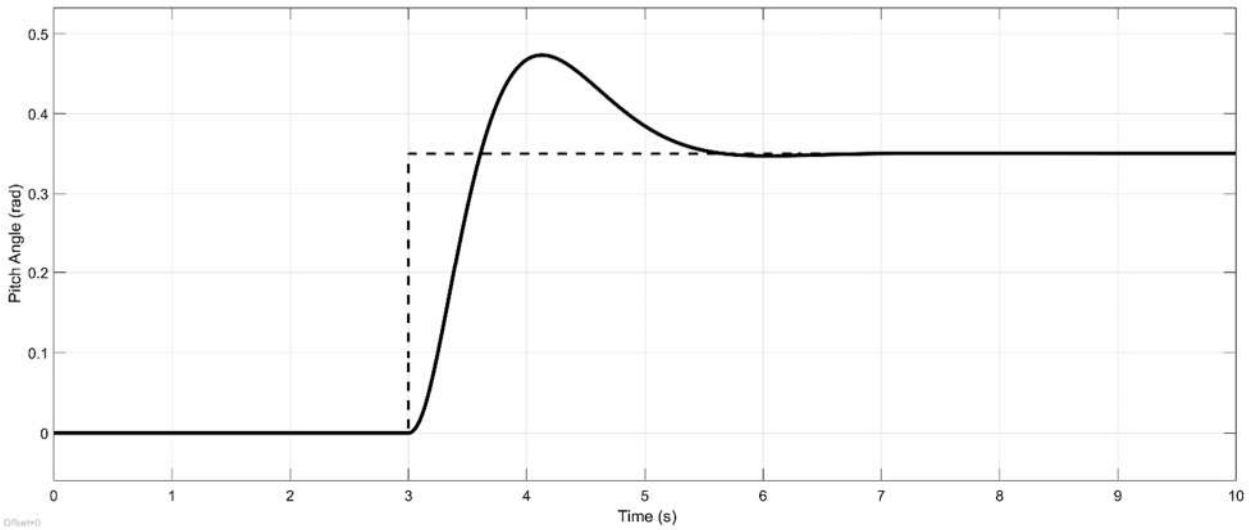

Figure 12. Pitch angle of quadcopter with FO-PID when γ=1.15

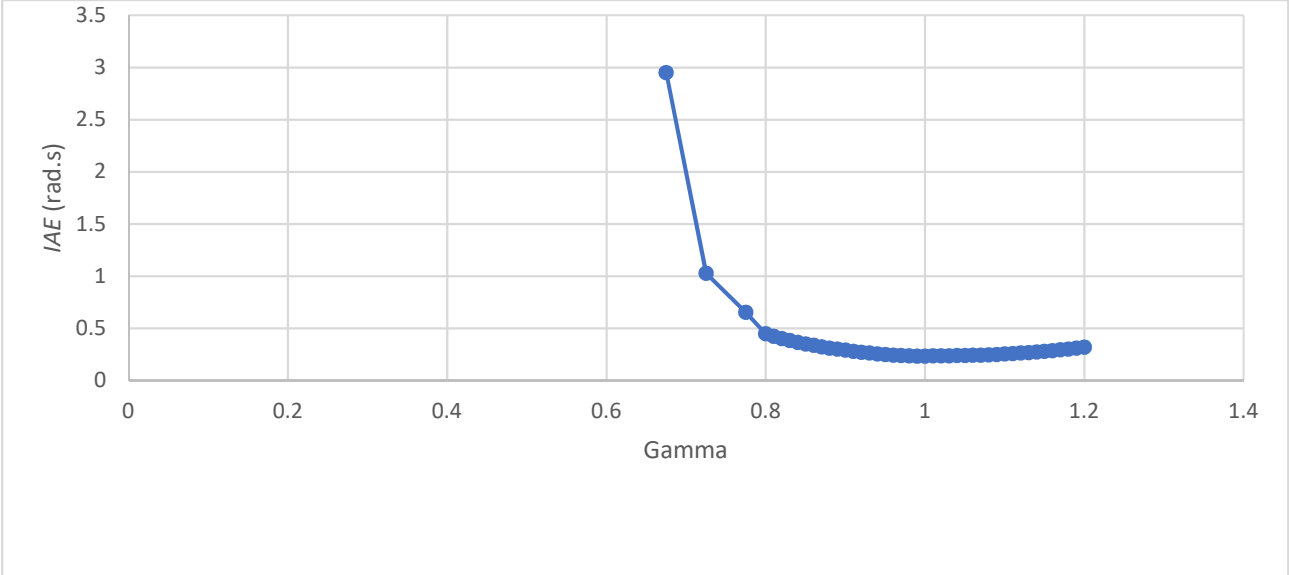

Figure 13. Gamma vs IAE for Pitching Motion

As it was stated earlier, the pitching command is given to controller assuming the quadcopter maintain its altitude at 1 (m). However, when quadcopter pitches, the angle of generated forces changes as well. Therefore, the vertical component of thrust forces which is responsible for adjusting altitude varies which affects the altitude. Figure 14 shows how altitude is affected when pitching command is given to conventional PID controller after 3 seconds. It must be added that

since pitching command is given with a delay, the time margin for calculation *IAE* has been extended to 15 (sec). It can be seen from Figure 12 that quadcopter experiences a critical undershoot of almost 30% and a critical overshoot of 22% while the value of *IAE* = 1.081 (m.s) which is not a good performance.

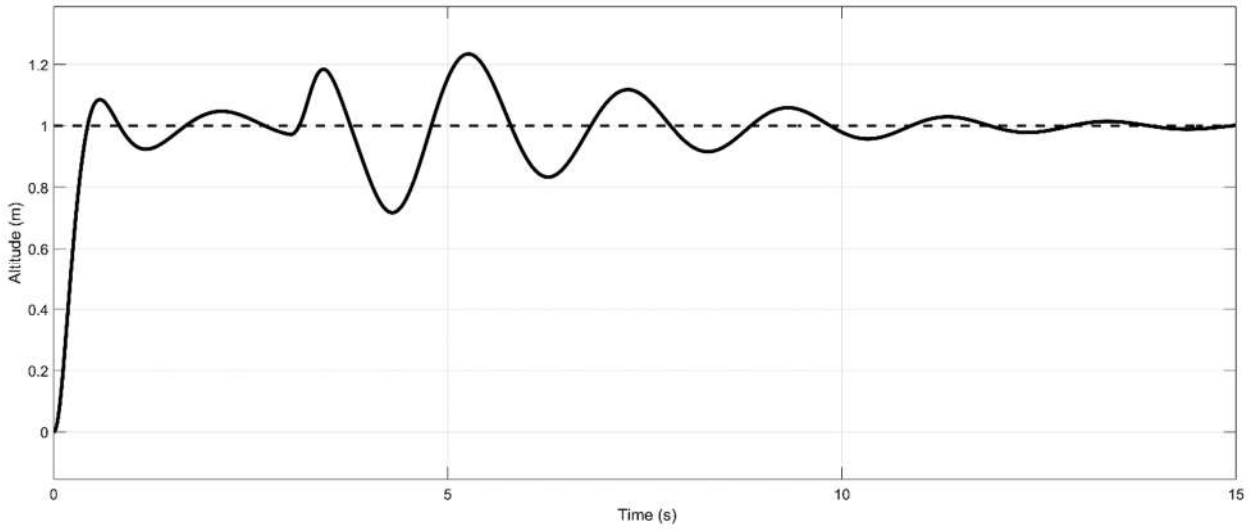

Figure 14. Variation of Altitude when Pitching Command is Given to IO-PID

However, Figure 14, which shows the same scenario for fractional order controller, has relatively better results. In this case, the critical undershoot is 20% and the critical overshoot is almost 18% while the *IAE* = 0.845 (m.s)

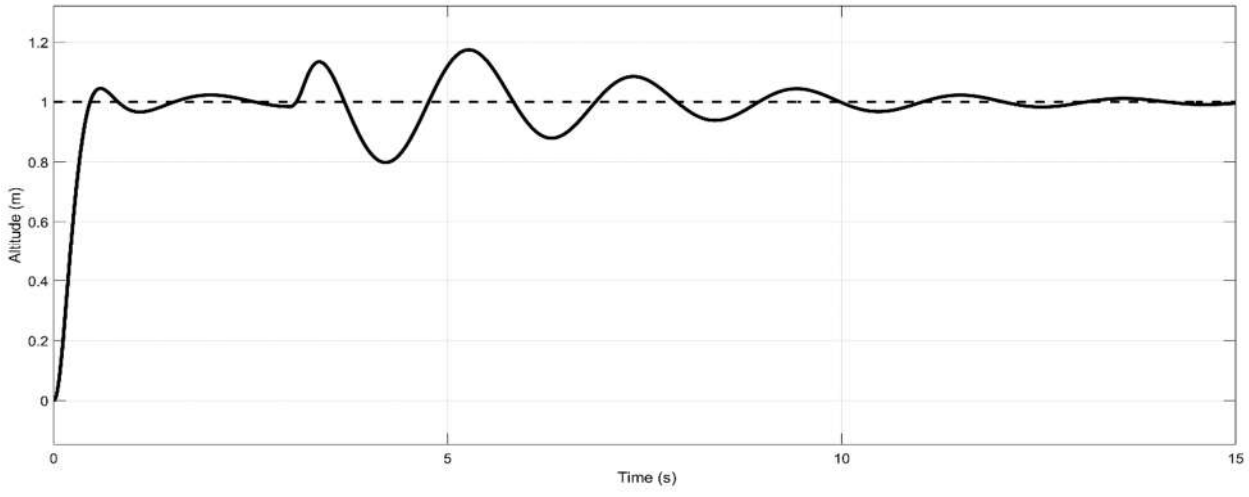

Figure 15. Variation of Altitude when Pitching Command is Given to FO-PID

Although fractional order controller in Figure 14 demonstrates better result compared to integer order controller in Figure 14, its performance of handling the altitude may not be satisfactory. Therefore, instead of a step input for pitching motion, a saturated ramp is considered to enhance the performance such that the output of this ramp for pitching input cannot exceed 0.35 (rad) however it may have different slopes *m*. Table 5 presents the values of *IAE* for both ascending and pitching motion of both integer and fractional controller while a variety of slopes are considered.

m (rad/s) input ramp slope	***IAE* (m.s) for Ascending with IO-PID**	***IAE* (rad.s) for Pitching with IO-PID**	***IAE* (m.s) for Ascending with FO-PID**	***IAE* (rad.s) for Pitching with FO-PID**
0.1	0.405	0.033	0.334	0.033
0.2	0.446	0.065	0.379	0.068
0.3	0.464	0.094	0.417	0.096
0.4	0.467	0.123	0.433	0.119
0.5	0.453	0.17	0.475	0.147
0.6	0.589	0.209	0.580	0.189
0.7	0.777	0.225	0.759	0.185
0.8	0.924	0.236	0.952	0.193
0.9	1.03	0.243	1.096	0.198
1	1.11	0.249	1.214	0.205
10	1.082	0.224	0.863	0.266
100	1.081	0.233	0.848	0.278

Table 5. Variations of IAE for Ascending and Pitching for both IO-PID and FO-PID with respect to slope of saturated ramp of pitch reference input

By looking at the trend of how the values of *IAE*, for both ascending and pitching motion, a few interesting key points can be concluded. First, for small slopes, which means slow variation of reference pitch input, the *IAE* for pitching motion for both controllers presented in 3rd and 5th columns are close to zero which indicates that both IO-PID and FO-PID have no problem following

input at this pace. Second, for large slopes like 10 (rad/s) and 100 (rad/s), the value of *IAE* for ascending motion presented in 2nd and 4th columns are similar to what was presented for step input. Third, by examining the 2nd column which is for IO-PID and 4th which is for FO-PID column, a nonlinear trend can be observed such that the smaller the slope is, the smaller *IAE* would be. For instance, the slope of 0.6 (rad/s) makes the value of *IAE* for ascending motion with IO-PID to reduce by 45.5% compared to a step input while with FO-PID the reduction is 31.6%. It should be mentioned that for the reference input to be saturated at 0.35 (rad) with the slope of 0.6 (rad/s), it only takes almost half a second.

Figure 16 shows the variation of altitude when a saturated ramp with the slope of 0.6 (rad/s) is given as a reference input of IO-PID controller responsible for pitching motion while Figure 17 shows the same except it is for FO-PID controller.

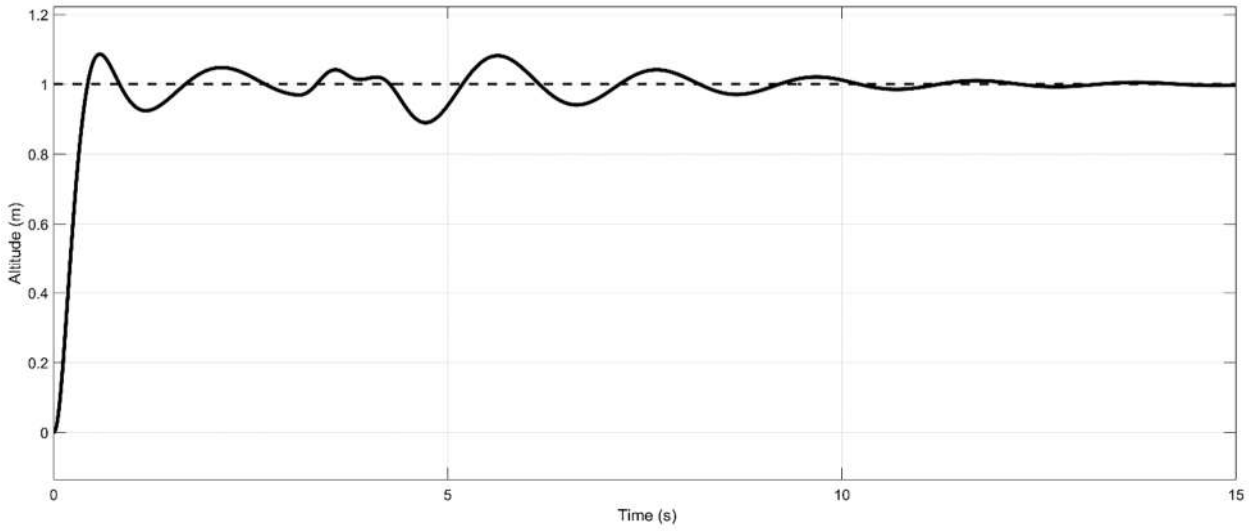

Figure 16. Altitude Variation for Saturated Ramp with the Slope of 0.6 (rad/s) using IO-PID Controller

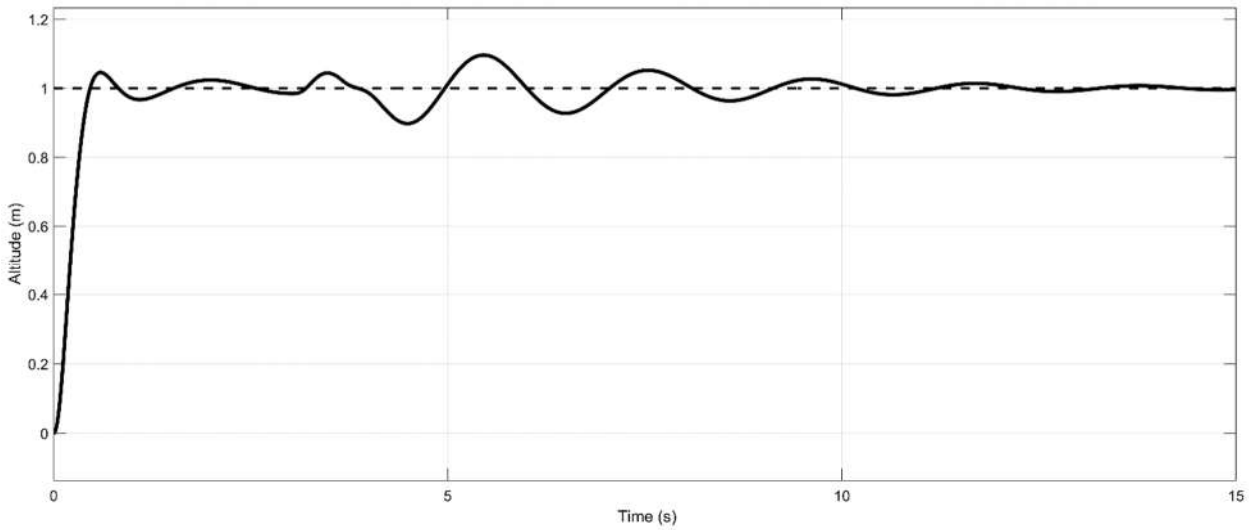

Figure 17. Altitude Variation for Saturated Ramp with the Slope of 0.6 (rad/s) using FO-PID Controller

4.4 Cross-Coupled Action

In previous section, it was assumed that quadcopter stays in altitude of 1 (m) then the pitching command is given. Therefore, the reference inputs for adjusting the altitude and pitch angle are not provided simultaneously. However, in this section it is assumed that both reference inputs are given at the same time such that quadcopter pitches as it ascends and in the other case descends.

Figure 18 shows the response of quadcopter when the reference inputs for altitude and pitch angle are set to 1 (m) and 0.35 (rad) respectively, at the beginning, using IO-PID controllers. The value of *IAE* for altitude, in this case, is 0.85 (m.s) while the *IAE* for pitching is 0.27 (rad.s). It can be seen that the response has a 20% undershoot in the first cycle.

Figure 19 shows the response of quadcopter under the same circumstances as in Figure 18 except that FO-PID controllers are used. The gains are same as in Table 2 and Table 4 while γ is 1.02 for fractional controller responsible for adjusting the altitude and 1.15 for fractional controller for adjusting pitch angle as in earlier sections. The value of *IAE* for altitude is 0.456 (m.s) which is 46% reduction and the overshoot is less than 10%, while the *IAE* for pitching is 0.27 (rad.s), same as in the case with IO-PID controller.

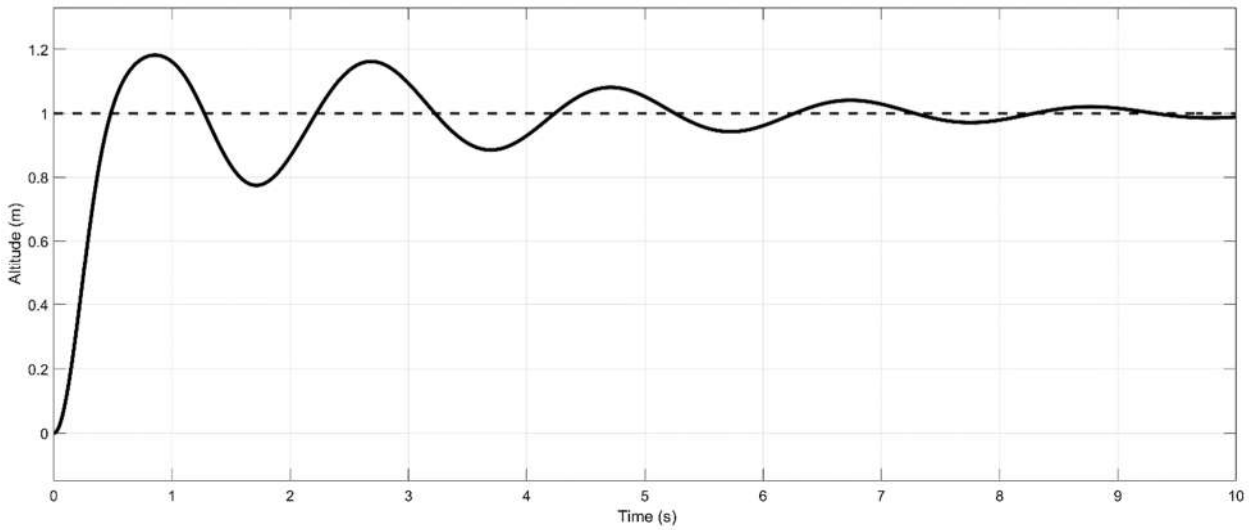

Figure 18. Altitude Variation when Quadcopter Ascends and Pitches Simultaneously Using IO-PID Controller

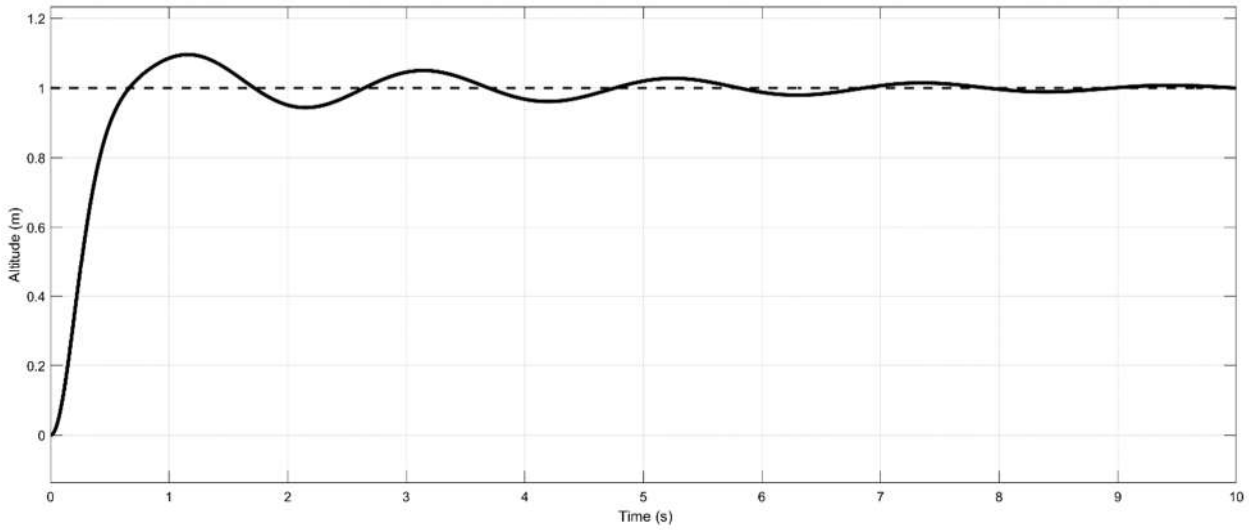

Figure 19. Altitude Variation when Quadcopter Ascends and Pitches Simultaneously Using FO-PID Controller

The case for pitching and descending is that quadcopter descends from altitude 1 (m) to 0.2 (m) while a reference input of 0.35 (rad) is given. The gains are same as in Table 3 and Table 4. Figure 20 shows how quadcopter descends while pitching using IO-PID controllers. The value of *IAE* for altitude is 0.65 (m.s) while the *IAE* for pitching is 0.439 (rad.s).

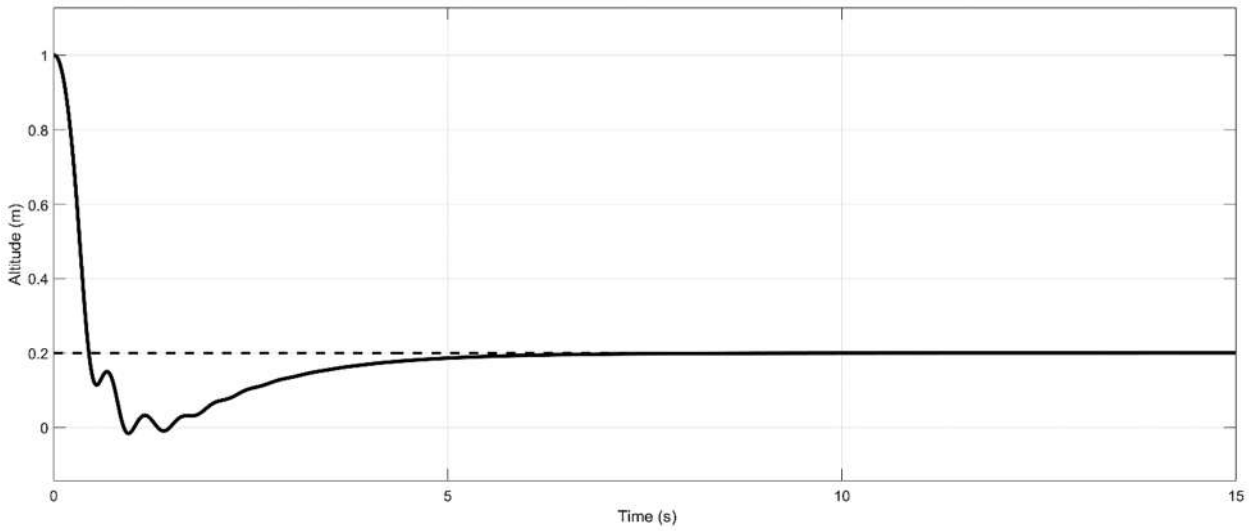

Figure 20. Altitude Variation when Quadcopter Descends and Pitches Simultaneously Using IO-PID Controller

Figure 21 shows how quadcopter descends while pitching using FO-PID controller. It should be mentioned that γ is 1.06 for fractional controller adjusting altitude and 1.15 for fractional controller adjusting the pitch angle. The value of *IAE* for altitude is 0.65 (m.s) which is same as with IO-PID controller while *IAE* for pitch angle is 0.469 (rad.s). It can be observed that the response with fractional controllers is smoother however integral of absolute error is not a precise indicator of smoothness.

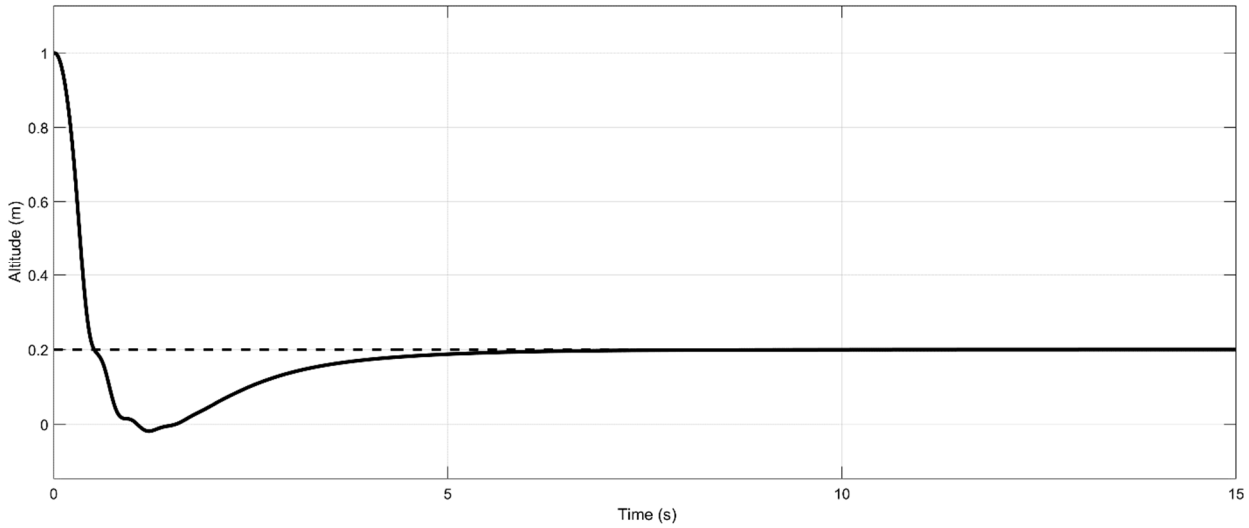

Figure 21. Altitude Variation when Quadcopter Descends and Pitches Simultaneously Using FO-PID Controller

By comparing Figure 20 to Figure 8 and by comparing Figure 21 to Figure 9, it can be seen that applying a reference input to adjust the pitch angle has not had a major effect on the way that quadcopter descends. The reason could be that motors are not capable of generating downward forces and regardless of the signals generated by either type of controllers, motors will not act.

Another set of cross-coupled action results could be considered when quadcopter experiences ascending and pitching and rolling simultaneously. Figure 22 shows the altitude variation of quadcopter using an IO-PID when a step reference input of 1 (m) is given and a step reference input for pitching and rolling of magnitude 0.35 (rad) are provided. Figure 23 shows the same scenario but for a FO-PID using $\gamma = 1.15$.

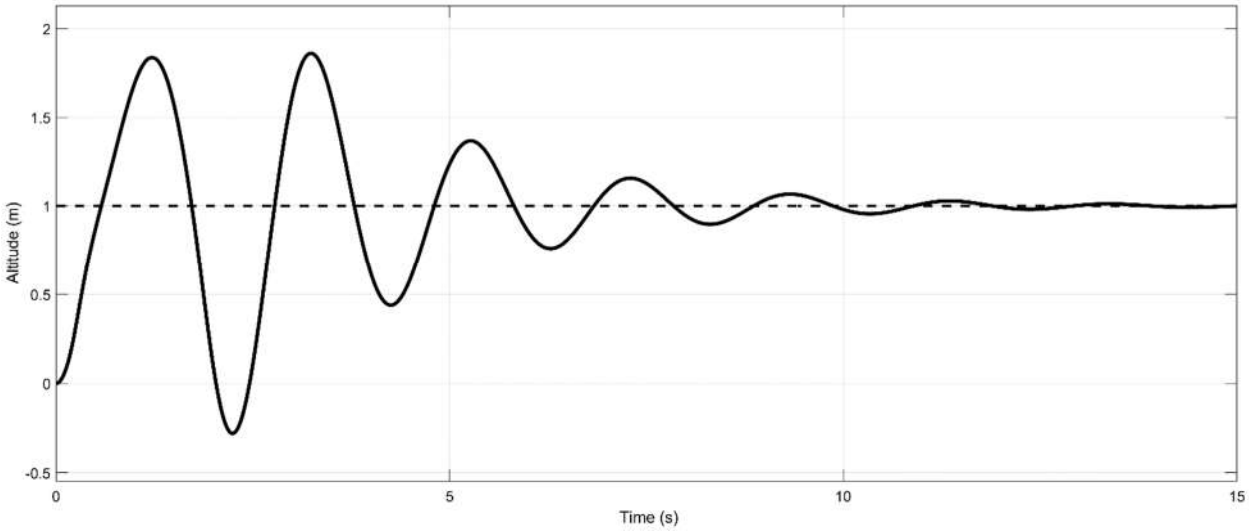

Figure 22. Altitude Variation for Ascending and Pitching and Rolling using IO-PID

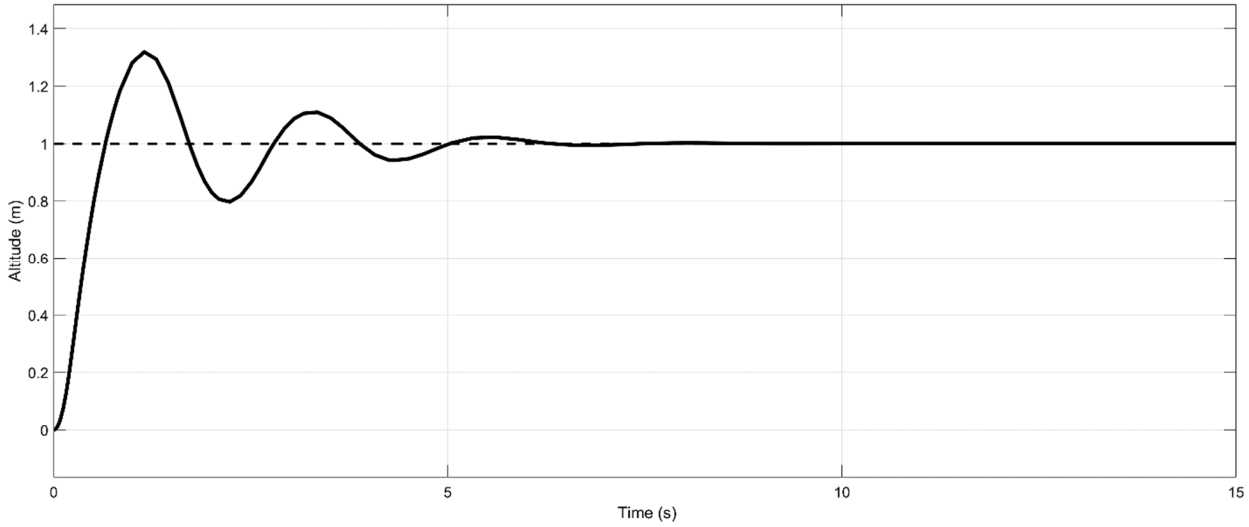

Figure 23. Altitude Variation for Ascending and Pitching and Rolling using FO-PID

As one can see, the performance of IO-PID in Figure 22 is very poor. It returns a value of $IAE = 3.35$ (m.s) and it has 86% overshoot and more than a 100% undershoot. To the contrary, FO-PID returns a value of $IAE = 0.857$ (m.s) and a 30% overshoot and 20% undershoot. Although, a 30% overshoot is not acceptable, it is expected to obtain better results if saturated ramp functions are given as reference input instead of step.

4.5 Conclusion

As the results showed in previous section, the fractional order controller is capable of enhancing the performance of the quadcopter compared to a conventional PID controller in various scenarios. It was also shown that the saturated ramp function can increase the performance of both integer order controller and fractional order controller in a nonlinear way. In conclusion section, it is beneficial to mention some points which could be investigated further regarding fractional order controller.

At first, due to the difficulty of providing a geometrical interpretation of fractional derivatives, intuitive approaches are relatively more cumbersome. For instance, it is known that first derivative of a function means the instantaneous slope of that function or the rate of change. Therefore, by studying the conventional controller comprehensively, it can be understood that how inputs, which are the error signals, make the system to behave in a certain way and how the changes of inputs affect it. However, when fractional derivatives are involved, analyzing the behavior of system gets more complicated.

Secondly, the computation of fractional derivatives is relatively costly compared to integer derivatives. Therefore, in design problems, financial resources must also be taken into account. In industry, a designer should always investigate whether a conventional PID controller can provide a satisfactory performance or not. If not, then fractional order controller can be taken into consideration.

Thirdly, the standard methods and criteria such as integral of absolute error (IAE) or integral of squared error (ISE) can be useful in order to determine the additional parameters regarding fractional order controller. However, they do not guarantee the best performance. Therefore, a verity of approaches toward designing a fractional order controller is recommended for further investigation.

References

[1] S. Wheatcraft, M. Meerschaert, “Fractional Conservation of Mass”, Elsevier Journal, 2008.

[2] A. Atangana, N. Bildik, “The Use of Fractional Order Derivative to Predict Groundwater Flow”, Mathematical Problems in Engineering, Hindawi Publishing Corporation, 2013.

[3] D. Benson, S. Wheatcraft, M. Meerschaert, “Application of a fractional advection dispersion equation”, Water Resource Research, Vol. 36, 2000.

[4] R. Metzler, J. Klafter, “The Random Walks Guide to Anomolous Diffusion: a Fractional Dynamics Approach”, ELSEVIER Journal, Physics Report 339, 2000.

[5] F. Mainardi, “Fractional Calculus and Waves in Linear Viscoelasticity”, Imperial College Press, 2010.

[6] S. Holm, S. P. Näsholm, "A causal and fractional all-frequency wave equation for lossy media". Journal of the Acoustical Society of America, 2011.

[7] D. Xue, C. Zhao, Y. Chen,” Fractional Order PID Control of A DC-Motor with Elastic Shaft: A Case Study”, Proceedings of the 2006 American Control Conference, June 2006.

[8] S. Dadras, H. R. Momeni, “Fractional terminal sliding mode control design for a class of dynamical systems with uncertainty”, ELSEVIER Journal, Communication in Nonlinear Science and Numerical Simulation, 2011.

[9] I.C. Cheeseman, W. E. Bennett, “The Effect of Ground on a Helicopter Rotor in Forward Flight”, Aeronautical Research Council Reports and Memoranda, Reports and Memoranda No. 3021, September 1955.

[10] F. Sabatino, “Quadrotor control: modeling, nonlinear control design, and simulation”, Master`s Degree Project, KTH Electrical Engineering, Sweden, June 2015.

[11] A. Katiar, R. Rashdi, Z. Ali, U. Baig, “Control and stability analysis of quadcopter”, International Conference on Computing, Mathematics and Engineering, 2018.

[12] J. K. Zbrozek, “Ground effect on the lifting rotor”, Aeronautical Research Council Reports and Memoranda, Reports and Memoranda No. 2347, June 1947.

[13] http://www.copters.com/aero/ground_effect.html.

[14] D. Xue, Y. Chen, Derek P. Atherton, “Linear Feedback Control: Analysis and Design with MATLAB”, Society for Industrial and Applied Mathematics, 2007, pp. 292-294.

[15] Stéphane Dugowson, Les différentielles métaphysiques (histoire et philosophie de la généralisation de l'ordre de dérivation), Thèse, Université Paris Nord (1994).

[16] I. Podlubny, “Fractional Differential Equations,” vol. 198, Academic Press, 1999.

[17] R. Scherer, S. Kalla, Y. Tang, J. Huang, “The Grünwald–Letnikov method for fractional differential equations”, ELSVIER Journal, Computer and Mathematics with Application, 2011.

[18] Y. Marushchak, B. Kopchak, “Approximation of Fractional Order Differential-Integral Controllers by Integer Order Transfer Functions”, Computational Problems of Electrical Engineering, Vol. 4, No. 1, 2014.

[19] MATLAB Software →Help → MATLAB Documentation →Simulink → Blocks → PID Controller

[20] Mehmet. Z. Sarikaya, Z. Dahmani, M. E. Kiris, F. Ahmed, “(k,s)-Riemann-Liouville Fractional Integral and Applications”, Hacettepe Journal of Mathematics and Statistics, Vol. 45, 2016, Page 77-89.

[21] M. Caputo, M. Fabrizio, “A New Definition of Fractional Derivative without Singular Kernel”, Progress in Fractional Differentiation and Application: An International Journal, No. 2, year 2015, page 73-85.

[22] K. B. Oldham, J. Spanier, “The Fractional Calculus: Theory and Applications of Differentiation and Integration to Arbitrary Order”, Academic Press: Dover Edition, 2006.

[23] S. Kurak, M. Hodzic, “Control and Estimation of Quadcopter Dynamical Model”, Periodicals of Engineering and Natural Science, Vol.6, No.1, March 2018, pp. 63-75.