IMPLEMENTATION OF FRACTIONAL ORDER CONTROLLER FOR STABILIZED VISUAL TRACKING SYSTEM IN REAL MILITARY PLATFORMS

A THESIS SUBMITTED TO
THE GRADUATE SCHOOL OF NATURAL AND APPLIED SCIENCES
OF
MIDDLE EAST TECHNICAL UNIVERSITY

BY

MEHMET BAYINDIR KARASOY

IN PARTIAL FULFILLMENT OF THE REQUIREMENTS
FOR
THE DEGREE OF MASTER OF SCIENCE
IN
ELECTRICAL AND ELECTRONICS ENGINEERING

JULY 2018

Approval of the thesis:

IMPLEMENTATION OF FRACTIONAL ORDER CONTROLLER FOR STABILIZED VISUAL TRACKING SYSTEM IN REAL MILITARY PLATFORMS

submitted by **MEHMET BAYINDIR KARASOY** in partial fulfillment of the requirements for the degree of **Master of Science in Electrical and Electronics Engineering Department, Middle East Technical University** by,

Prof. Dr. Halil Kalıpçılar
Dean, Graduate School of **Natural and Applied Sciences** _____

Prof. Dr. Tolga Çiloğlu
Head of Department, **Electrical and Electronics Engineering** _____

Prof. Dr. Aydan Erkmen
Supervisor, **Electrical and Electronics Engineering Department, METU** _____

Examining Committee Members:

Prof. Dr. Aydan Erkmen
Electrical and Electronics Engineering Department, METU _____

Prof. Dr. Kemal Leblebicioğlu
Electrical and Electronics Engineering Department, METU _____

Assist. Prof. Dr. Mustafa Mert Ankaralı
Electrical and Electronics Engineering Department, METU _____

Assoc. Prof. Dr. Uluç Saranlı
Computer Engineering Department, METU _____

Prof. Dr. Ömer Morgül
Electrical and Electronics Engineering Department, Bilkent University _____

Date: _____

I hereby declare that all information in this document has been obtained and presented in accordance with academic rules and ethical conduct. I also declare that, as required by these rules and conduct, I have fully cited and referenced all material and results that are not original to this work.

Name, Last Name: MEHMET BAYINDIR KARASOY

Signature :

ABSTRACT

IMPLEMENTATION OF FRACTIONAL ORDER CONTROLLER FOR STABILIZED VISUAL TRACKING SYSTEM IN REAL MILITARY PLATFORMS

Karasoy, Mehmet Bayındır
M.S., Department of Electrical and Electronics Engineering
Supervisor : Prof. Dr. Aydan Erkmen

July 2018, 179 pages

Main tasks of the defence systems are mainly built on the stabilization and target tracking features of the platforms. The servo controllers in these platforms are responsible to obtain desired performance over challenging environments. In this thesis, Fractional Order Controller is developed and implemented to stabilized, visual tracking real military system. Aim of the study is to improve stabilization and target tracking performances of the gimbal. The system identification procedure is applied and system model is obtained to simulate real system conditions. The proposed controllers and their realization procedures with their optimal tuning studies are discussed. The comparison tests of integer order and fractional order controllers are performed both in simulations and real system. For real system tests, stewart platform is used to simulate external disturbances and stabilization accuracy and target tracking performances are measured.

Keywords: Fractional order control, stabilization, video tracking, disturbance rejection, optimal tuning, system identification

ÖZ

KESİR DERECELİ KONTROLCÜLERİN GERÇEK ASKERİ PLATFORMLARDA STABİLİZE GÖRSEL TAKİP SİSTEMLERİNE UYGULANMASI

Karasoy, Mehmet Bayındır
Yüksek Lisans, Elektrik ve Elektronik Mühendisliği Bölümü
Tez Yöneticisi : Prof. Dr. Aydan Erkmen

Temmuz 2018 , 179 sayfa

Savunma sistemlerinin temel görevleri, temel olarak platformların istikrar ve hedef izleme özellikleri üzerine kurulmuştur. Bu platformlardaki servo kontrol cihazları zorlu ortamlarda istenen performansı elde etmekten sorumludur. Bu tezde Kesirli Dereceli Kontrolörü, stabilize, görsel takip gerçek askeri sistem için geliştirilmiş ve uygulanmıştır. Çalışmanın amacı gimbalin stabilizasyon ve hedef izleme performanslarını iyileştirmektir. Sistem tanımlama prosedürü uygulanmış ve gerçek sistem koşullarını simüle etmek için sistem modeli elde edilmiştir. Önerilen kontrolörler ve bunların ideal ayarlamaları ile gerçekleşme prosedürleri tartışılmıştır. Tamsayılı sıralama ve kesirli dereceli denetleyicilerinin karşılaştırma testleri hem simülasyonlarda hem de gerçek sistemde gerçekleştirilmiştir. Gerçek sistem testleri için, harici bozucu etkileri simüle etmek için stewart platformu kullanılmış ve stabilizasyon hassasiyeti ile hedef takip performansları ölçülmüştür.

Anahtar Kelimeler: Fractional order control, stabilization, video tracking, disturbance rejection, optimal tuning, system identification

To my beloved love

ACKNOWLEDGMENTS

I would like to express my deepest gratitude to my supervisor Prof.Dr.Aydan Erkmen for her guidance, attention and valuable critics , which helped me enhance my research.

I would like to thank Deniz Mutlu and Hayrettin Ulaş Akova for sharing their experience and guidance on the topic. I would like express my special gratitude to Mustafa Burak Gürcan for granting me the opportunity to write this thesis and thank ASELSAN A.Ş for providing me all the resources I used throughout my thesis

I am very grateful to have my sister, Gökçe, my parents, Hatice and Yakup. I always feel their endless support, love and patience to me.

Last but not least, I would like to thank my wife, Ayşe. I am thankful for her eternal love and continuous encouragement in my life. Without her support, I could not finish this thesis.

TABLE OF CONTENTS

LIST OF TABLES

TABLES

LIST OF FIGURES

FIGURES

LIST OF ABBREVIATIONS

HASG	High-Accuracy Stabilized Gimbal
EOS	Electro-Optical System
FO	Fractional Order
IO	Integer Order
FOC	Fractional Order Control
IOC	Integer Order Control
ORA	Oustaloup Recursive Approximation
CFE	Continuous Fraction Expansion
PSE	Power Series Expansion
P	Proportional
PI	Proportional-Integral
PID	Proportional-Integral-Derivative
LTI	Linear Time Invariant
ITAE	Integral Time Absolute Error
ITSE	Integral Time Square Error
IAE	Integral Absolute Error
ISE	Integral Square Error
GA	Genetic Algorithm
GS	Global Search
FPGA	Field Programmable Gate Array
UAV	Unmanned Aerial Vehicle
HDD	Hard Disc Drive
LOS	Line of Sight
FOV	Field of View
DOF	Degree of Freedom
DSP	Digital Signal Processor
FRF	Frequency Response Function
GUI	Graphical User Interface

CHAPTER 1

INTRODUCTION

1.1 Motivation

Although the main tasks of defense systems varies from each other for different environments, the common requirements of the majority for these systems are that they must be equipped with navigation and surveillance functions. In general, electro-optic platforms are responsible to fulfill these functions. Regardless of environmental conditions and the place where electro-optic platforms are mounted, these systems must provide full functionality. Depending on the usage, these functionalities are target identification, target tracking, kill assessment and missile guidance. Therefore, with the need for defense systems, electro-optic platforms become more important, and their performances, capabilities play crucial role on battlefield.

An electro-optic platform is responsible for detecting any threat in its field of view. These threats can be in different forms such as vessels, aircrafts or missiles. The main aim of the electro-optic platform is to track these targets. In order to be able to track, electro-optic platforms generally includes a tracker and mission computer unit to measure line of sight position error of the platform with respect to target. The servo unit of the platform calculates required torque values with these errors. In these calculations, the controllers in the servo unit are used. The obtained performance from these controllers are directly related to system's hit performance. Therefore, having a great performance in target tracking is one of

the main foci of the servo units in electro-optic systems.

Most of the electro-optic systems have a compact structure, so they are widely used in various platforms. They can take place as commander periscope in a tank or surveillance sighting unit in a vessel. Acquired view from these electro-optics play crucial role on defense missions of these platforms. Besides, they may accompany weapon system to provide sight and threats locations to weapon triggering. Instead of stabilizing the entire heavy and large gun system, it is generally preferable to stabilize the optical sensors which take up less space and assume higher precision. Platform disturbances, external disturbances, fire shocks are the main disturbance sources of the operating environment. Therefore, holding line of sight stationary and obtaining clear image from electro-optics without any jitter against all kind of disturbances is one most crucial capabilities of the electro-optic platforms. Stabilization performance is the another key criterion to fulfill defensive requirements of electro-optic platforms.

In order to provide required stabilization and tracking performance for these tasks, proportional, integral and derivative type (PID) controllers are generally used in the control structure of electro-optic platform since PID controllers are still most popular compared to other control types: PID controllers are easy to compute, implement, tune and numerous analysis tools exist for these controllers. These controllers are not perfect and have robustness issues when confronted by high level of disturbances and time delays. Therefore, better target tracking performance and stabilization accuracy require better controller design compared to that of classical PID controllers.

Hence, high stabilization and tracking performance is always a desired features in electro-optic platforms. In order to overcome disturbances, controllers must be well designed and must have better disturbance rejection characteristics. The main aim of this thesis is to design a controller such that electro-optic platforms would have better performance in terms of stabilization and target tracking performances, and achieve enhanced system effectiveness in the defense industry.

1.2 Problem Definition

In motivation section, it is emphasized that having good stabilization accuracy and target tracking performance are key features of stabilized electro-optic platforms. These systems are used in different environmental conditions containing different disturbances. The main objective of the servo system is to overcome these problems and have desired stabilization and target tracking accuracies.

The performance of the servo subsystem under the given commands in eliminating the disruptive effects is directly related to the increase in the bandwidth of this subsystem. In order to increase the bandwidth of the system, the natural frequencies of the mechanical structure should be controlled, nonlinearities like backlash and friction should be minimized, the delays in the sensors and communication infrastructure should be removed and the sampling time should be increased. The bandwidth of the control loop determines how quickly the servo system respond to inputs such as position, velocity and torque demands. Higher bandwidth generally provides better motor performance, decreases errors and improves response times.

The most challenging problem in servo control of electro-optic platforms is generally the non-linearities in the system such as backlash, friction, and servo limitations. As most of the electro-optic platforms has compact structure with very low inertia, friction would cause enormous problems in stabilization and target tracking. When the static and dynamic friction levels differs from each other, the control of the low-inertia system is a very challenging task with classical PID controllers. Servo limitations the another source of non-linearity in servo systems. Because of the used hardware, maximum torque, maximum speed and acceleration values have limitations, which sometimes prevent servo system to operate in desired performance.

Target tracking is designed to minimize the error occurring due to its nature which is caused during fusion of information obtained from different sensors and tracking algorithms. These informations are the position of the servo system and line of sight position of the target obtained from tracker algorithms. When the bandwidth and the gains of servo systems are low, position errors are high enough to deteriorate the required tracking task. In order to diminish the level of this error, better accuracy feed-forwarding target velocity is an applicable solution. The success of this way depends on the correctness of the target velocity. If the estimated velocity and actual target velocity are close enough, or equal, theoretically , tracking error would not occur, tracking errors are minimized with the designed controller. Therefore, higher bandwidth in control loops is required to have better target tracking performances, and servo controllers must be designed in such way to obtain highest closed loop bandwidth.

High stabilization accuracy and target tracking performances require well-designed servo controller to cope with the explained challenges. The servo controller should minimize settling time, decrease the overshoot and oscillations of the response. The controller should be reliable enough to respond to changes in the environmental and system dynamics. The controller need to be designed to improve performances of the platform in the light of these requirements.

1.3 Methodology

When using fractional order calculus and fractional order controllers under the problems introduced in previous section, we aim at improving current performance of the system where classical integer order PID controllers are used. Using fractional differential equations in PID instead of classical integration and derivation actions, the obtained fractional order $PI^{\lambda}D^{\mu}$ controller is proposed for the stabilized visual tracking system.

The fractional order controller brings two more tuning nodes to the controller which are the integration order λ and derivation order μ. Having more parameters to tune gives fractional order controllers flexibility to adjust gain and phase characteristics. This flexibility makes fractional order controller more powerful than the integer order controllers in terms of robustness. Fractional order controllers requires system's whole state history, however with proper approximation and realization methods, these controllers are implemented with considerable low complexities.

The fractional order controller offer tuning flexibility to fulfill more design specifications. However, fractional order controllers must be tuned in a systematic way. Although some classical order controller tuning rules were tried to be adopted for fractional order controllers, the preferred way to tune fractional order controllers is generally using optimization procedures which enable finding an optimal set for $PI^{\lambda}D^{\mu}$ parameters towards meeting the requirements.

The purpose of designing $PI^{\lambda}D^{\mu}$ to improve stabilization and target tracking performances of the electro-optic platforms rely on harnessing the advantage of the two extra tuning nodes that provide more flexibilities in fulfilling design specifications. The resulting controller with fractional order derivation and integration ensure more reliable performance when confronting gain variations and any kind of disturbance and noise.

1.4 Contribution

In the literature, several applications of fractional calculus can be found, especially in the field of control theory where plenty of studies introduce better functionality of fractional order controller over integer order controllers. Most of these works are focused on obtaining theoretical and analytical results,very few studies focus on real system applications. In this thesis, the main concern

is to illustrate the advantages of the fractional order controller in real system applications and especially with the military precision.

To best of our knowledge, fractional order controllers are implemented for stabilized visual tracking system in real military platform for the first time. Implementation on real military platform requires many design concerns; the response speed of the controller, the memory constraints of the module which controller is implemented, the performances of the system which enables platform to perform its tasks. In this thesis, all implementation procedures are covered in detail, and analyzed performance wise.

The military platform is an electro-optic platform used in a vessel for surveillance and defense tasks. Therefore, performance of the designed controller is tested under disturbances which represent the actual sea environment. Detailed stabilization tests are performed in order to compare integer and fractional order controllers. After stabilization tests, the real target tracking tests are performed. 6-DOF Stewart platform and recorded real aircraft video is used for stabilization and video tracking comparison tests. In these comparisons, the content of comparison of fractional and integer order controllers is extended and unlike other works in literature, stabilization accuracy and the target tracking performance values are measured and represented for the first time. This thesis provides a first full analytical design procedures of fractional order controller concerning performance criteria on real stabilized visual tracking military system.

1.5 Outline of Thesis

The fractional calculus is a relatively popular topic that gets attention of the many researchers working control theory in last years. However, there are still many people unfamiliar with fractional calculus and its applications. Therefore, the literature survey on fractional calculus and target tracking and the mathematical background of the topic is covered first in the thesis as chapter two. Literature survey includes the historical development of the fractional calculus, its applications to various fields, the fractional systems, fractional order controllers, the approximation and realization methods of these controllers, tuning strategies, real system implementations, and main aspects of stabilization platforms with target tracking tasks.

In chapter three, the fractional calculus, the mathematical aspects of the topic are given first. The definition of the fractional integration and derivation are studied and compared. In order to visualize these new concepts geometrically and physically, several examples are studied. Later, the system modeling examples are covered, mathematical models of the systems are obtained by using fractional order integration and derivation. The approximation and realization methods are covered, their definitions are given. At the end of chapter, the stability properties o fractional order systems and the fractional order controllers are covered.

In chapter four, the real hardware system is introduced together with all the hardware tests to be performed. With this system, the test platform is also introduced to show the capabilities of the our tests. The system identification is explained in detail for the simulation tests, which will construct the first part of the results of the our thesis.

In chapter five, the control structure in the hardware is explained first. Then, the controller approximation and realizations are discussed towards the design

of the controller including the approximation and tuning parts.

In chapter six and seven, the simulation and hardware experimental tests are presented respectively. These tests are performed for integer and fractional order controllers. The comparison of the controllers are performed in terms of speed loop responses, disturbance rejection characteristics and reference tracking tests in simulations. Similarly, the comparison in hardware tests includes the stabilization accuracy comparison, and the target tracking performances. After each results are presented, they are discussed and detailed comparison is made for both controllers.

At the end of thesis, conclusive remarks are given opening the horizon to possible future works. All the references used in the thesis are given in the references section.

CHAPTER 2

LITERATURE SURVEY

Fractional Calculus was mentioned first time in a letter by L'Hospital to Leibniz in 1695 [8],[13]. Leibniz introduced the definition for the nth order derivation as follow,

$$\frac{d^n}{dx^n} f(x) \tag{2.1}$$

L'Hospital's letter was specifically asking the meaning of derivative action if n=1/2. The idea of order of differentiation being a fraction, opened new field in mathematics and it is called as Fractional Calculus. Fractional calculus is based on a differentiation and integration to an arbitrary order [12].

Since the early 1800s, the several studies are done to express fractional derivative and integral operators. In [16], different formulations of these operators are given. In [15], [6], the most commonly accepted and used definitions of fractional integration and derivations are addressed and given. These definitions dated from the 19th century are called as Grunwald-Letnikov, Riemann-Liouville and Caputo definitions. For example, Abey has discovered that the solution of the integral equation for the tautochrone could be achieved by semi-derivative [15]. Similarly, Heaviside's works on electromagnetic theory, transmission line theory, and elasticity with Gemant's extension were the other early applications. Later, the fractional calculus has been used in several applications of mechanics, physics, chemistry, control, signal processing and economics.

While the fractional calculus becomes more widespread and popular, the ap-

proximations and realization methods became more important. In [22] , several approximations and realization methods are presented and their examples are given. In [23] CRONE Toolbox on MATLAB is introduced. The Crone Toolbox is mainly focuses on system identification and control of these systems via fractional calculus phenomena. The other toolboxes in the literature are FOTF by Chen [7], Ninteger by Valerio [24], FOMCON by Tepljakov [25]. All of these toolboxes provides MATLAB implementations to fractioal order derivative and integral, and tools for control.

In [6] the analog implementations of the fractional derivative is given as fractance circuits. These circuits includes both capacitors and resistors modeled in classical sense, however the behavior of the circuit may show a fractional characteristics., resistor models. The more information about the characteristics of the fractance element is covered in these works [17],[18]

As digital implementation of fractional operators, in [26] pic microprocessors are used. In this paper, it is shown that pic microprocessors could be used as hardware realization tool in order to implement numerical approximations of fractional operators. In [27], the fractional order controllers are implemented to DSP module via using software environment called Code Composer Studio. There are several other similar works for hardware implementations of fractional operators, such as microcontrollers [28], FPGA [29].

The fractional calculus is used to model semi-infinite lossy transmission line in [5], to model diffusion of heat through semi-infinite solid in [6], to model the suspension of automotive in [20], to propose new electrical circuit element called fractance in [17],[18],[19], to model dynamic backlash in [21], and many others. In [8], the authors points out that "Real objects are generally fractional." with the emphasize of fractional calculus could model and describe objects better than the integer order case. The fractional order system modeling and their features can be found in [10],[11],[12] with the definitions of Laplace Transform

of fractional operators, impulse responses, numerical methods to calculate fractional operators.

One of the strongest aspects of the fractional calculus is to control nonlinearities, anomalies, complex dynamic systems [7],[11]. In [11], fractional order chaotic systems are given with mathematical models, and control strategies to these models are covered. In paper [30], the control of a dc motor with elastic shaft is studied, and illustrated that fractional order controller gives better result than the integer order controller. Similar studies are mode for backlash vibration [31], for cogging effect compensation in permanent magnetic synchronous motors [32], for dynamic friction [33].

The fractional order $PI^{\lambda}D^{\mu}$ controller is defined and proposed in [34], the general structure of it is constructed and represented. In [7],[35], it is illustrated that the fractional order $PI^{\lambda}D^{\mu}$ controller outperforms the classical PID controller. This comparison is done for both fractional order plants and integer order plants. The research shows that 90 percent of the controllers in the industry are PID type [37]. Therefore, the success of fractional order $PI^{\lambda}D^{\mu}$ controller over classical integer order controller becomes more important. In addition to PID controllers, In [36] the fractional fuzzy adaptive sliding mode controller is designed for 2-DOF robot arm. With the novel parameter adjustment scheme introduced in the paper, better tracking performance and very high degree of robustness to disturbances are achieved.

In order to make a comparison between fractional and integer order controllers, the tuning of the controllers must be done systematically. In [38], the procedures for tuning fractional order controllers are introduced. The tuning procedure given in [39] provides specifically basing it upon on the given stability measures and damping ratio of the closed control loop. The tuning rules proposed in [41] are inspired from Ziegler-Nichols rule for integer order PID controllers. In [40],[7],[42], optimization procedures are used for tuning the fractional order

controller parameters. In these studies, both fractional and integer order PID controllers are tuned with optimization routines, and it is observed that fractional order controllers had better response.

Servo systems are one of the most popular areas that fractional order controllers implemented for. In book [8], there are several works are given for velocity control, position control, and stability, feasibility features of fractional order systems. Controlling DC-motor, small fixed-wing UAV, HDD servo systems are studied and results are presented with controller design procedures and comparisons of these controller via step responses. In [44] the application of fractional order control to motion control is given with experimental torsional system. It is illustrated that the derivative order could change the characteristics of the control and provides more flexibility to design controllers.

In servo platforms, one of the main task is to stabilize payload which can be a gun or electro optics. For electro optic platform, this task is to isolate optical instruments from the platform and other external disturbances and prevent image blur [61]. For this purpose, the gyros are installed on the platform to sense inertial angular motion. Stabilizing the platform establishes a relation with the vehicle's spatial motion and the earth [62] so, recent developments in optical sensors, lasers the electro optical platforms became more important in defense industry to identify targets from long range, track and locate their position on earth. These applications require more stringent performance for positioning and stabilizing the platform under disturbances [63]. Target tracking from a long distance require ultra-low speed requirements for servo system without any vibration, and pointing accuracy must be as low as possible such as few pixels.

The stabilization and pointing or video tracking requires two cascaded loop. The inner control loop is named as speed loop and is used for stabilization, the outer loop is called as position loop and is used for aiming purposes for tracking a target. [1]. The speed loop must have high enough bandwidth to reject external

and internal disturbances to stabilize the line of sight which is shown in figure 2.1.

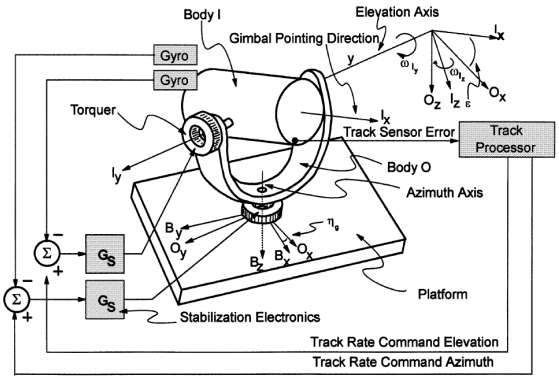

Figure 2.1: Two-axis tracker configuration [1]

In figure, two-axis gimbal is represented. Two-axis gyroscope is used for stabilizing elevation and traverse axis. The gimbal's elevation and traverse axis position values gives the pointing direction. Target's position in the line of sight is captured and track sensor errors are generated with using angles of gimbal and the field of view of the optic sensors by tracker processor. The position errors are sent to the servo controller to be minimized. Having all steps completed, one can achieve target tracking.

The position loop, where tracking is performed by, operates with LOS angle. Tracker loop accepts the commands from the target tracker unit which processes the optical image fed by the optical sensors and determines the target location in the field of view. Succeeding the determination of target position on the field of view, the orientation error of the optical platform is generated and sent to the servo system. In target tracking, sensor noise, process noise while finding the target position on the image play a crucial role on the performance. If tracker's sampling rate is lower than the servo loops, tracking errors must be interpolated and then fed to position loop in order to avoid discontinuous movements in position. For better performance, the tracker subsystems at least must work in similar sampling rates with servo subsystem.

The most important and indicative parameter to show how well and effective disturbance rejection is performed and the system is stabilized, is the closed loop

bandwidth. The ability to follow commands are directly related with bandwidth. The structural resonances and the dynamics of the system determine the bandwidth limit of the system [9].

The control of the LOS orientation plays a tremendous role on every kind of electro optic related defense applications in terms of orienting the optical sensors without any jitter, and getting clear image. Such systems with greater stabilization and tracking performance which is a multidisciplinary work and requires a full understanding of operating environments, mechanics, controls, system integrations and test procedures [2].

CHAPTER 3

FRACTIONAL CALCULUS AND CONTROL APPLICATIONS

As fractional calculus got more and more attention in last decades, there are numerous dynamical systems modeled using fractional approaches which are generally based upon using fractional integration and derivation where the order of the systems does not have to be integer compared to the old, classical calculus that requires integer.

Both system modeling and controller design by fractional order calculus require understanding well the theory of fractional calculus. In this chapter, we aim to make the reader familiar with fractional calculus and its usage in control applications.

The chapter begins with the mathematical descriptions of fractional derivations and integrations. In order to visualize these descriptions and distinguish them from the integer order cases, several examples are provided in the second part of chapter. The chapter continues with fractional order system modeling examples which illustrate that some specific systems have in their motive fractional order degree rather than integer order. Later, the realization and approximation methods of fractional order controllers are discussed. The fractional order control applications are mentioned at the end of chapter, the well-known concepts of control theories are discussed here in terms of fractional order systems.

3.1 Mathematical Background

3.1.1 Fractional Derivative and Integral Definitions

Initially, Leibniz and L'Hopital in 1695 mentioned about half-order derivative and since then, there has been numerous works on fractional order operators leading to various definitions [8],[13]. This section will overview the commonly used definitions of fractional order integral and derivative.

At first, we will define a general operator as a fractional integral and derivative operator D_t^r, called integro-differential operator [7] that is:

$$ {}_a\mathrm{D}_t^r = \begin{cases} \frac{d^r}{dt^n} & Re(r) > 0 \\ 1 & Re(r) = 0 \\ \int_a^t (d\tau)^{-r} & Re(r) > 0 \end{cases} \tag{3.1} $$

This general expression emerged from the need of the notation for fractional order operator. In this definition, r is a fractional number that specifies the degree of integral or derivation depending on its sign. a and t determine the interval of operator takes place.

Integro-differential operator takes different forms based on several leading definitions of derivation and integrations;

- **Riemann-Liouville Definition**

  This definition is one of the most popular definitions in the field of fractional calculus. The definition is proposed by Bernhard Riemann and Joseph Liouville [3]. In this definition, fractional order integral is defined as,

  $$ {}_a\mathrm{D}_t^{-\alpha} f(t) = \frac{1}{\Gamma(\alpha)} \int_a^t (t-\tau)^{\alpha-1} f(\tau) d\tau \tag{3.2} $$

  where, a is the initial time instance and $0 < \alpha < 1$. Fractional order derivation is defined as,

  $$ {}_a\mathrm{D}_t^{\alpha} f(t) = \frac{d^n}{dt^n} [{}_a\mathrm{D}_t^{-(n-\alpha)} f(t)] = \frac{1}{\Gamma(n-\alpha)} \frac{d^n}{dt^n} \left[\int_a^t \frac{f(\tau)}{(t-\tau)^{\alpha-n+1}} d\tau \right]. \tag{3.3} $$

where $n-1 < \alpha < n$. In this case, some of the properties of integro-differential operators are used to define fractional order derivative. The Riemann-Liouville definition is based on integration, therefore in order to do fractional derivation operations, the fractional integration and integer order derivations are used. The properties of *Integer Order Equivalence* and *Commutativity* of integro-differential operator are used for this purpose. In figure 3.4, these features can be visualized for differentiation of order α=1.5,

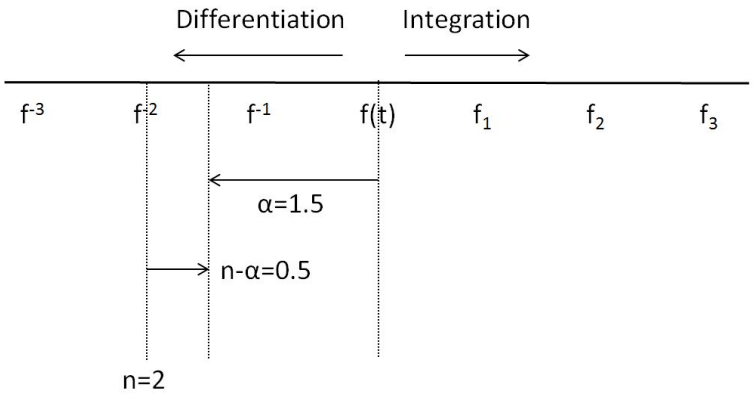

Figure 3.1: Riemann-Liouville Definition of Fractional Derivation [3]

In the figure, derivation is represented with left arrows where the integration is the right arrow. In classical integration and derivations, the length of these arrows are integer. In our case, the derivation of order 1.5 is shown in two parts. Derivation the function f(t) by order of 2, then integrating it by order 0.5 will give the solution as below,

$$ {}_a\mathrm{D}_t^{1.5} f(t) =_a \mathrm{D}_t^2 [{}_a\mathrm{D}_t^{-0.5} f(t)] \tag{3.4} $$

The properties of the integro-differential operator ${}_a\mathrm{D}_t^{-\alpha}$ are given in the Appendix A with more detailed explanations.

- **Grünwald-Letkinov Definition**

  Another popular definition that commonly uses in fractional calculus is proposed by Anton Karl Grünwald followed by Aleksey Vasilievich Letnikov [11]. Unlike Riemann-Liouville definitions, Grünwald-Letnikov definitions approaches the problem from the basic concept of derivation.

Lets begin with the definition of a derivative of first order;

$$f^{'}(x) = \lim_{h \to 0} \frac{f(x+h) - f(x)}{h} \tag{3.5}$$

The second derivation;

$$f^{''}(x) = \lim_{h \to 0} \frac{f(x+2h) - 2f(x+h) + f(x)}{h^2} \tag{3.6}$$

Considering first and second derivative, for the n^{th} derivative, the following definition can be generated,

$$d^n f(x) = \lim_{h \to 0} \frac{1}{h^n} \sum_{m=0}^{n} (-1)^n \binom{n}{m} f(x - mh) \tag{3.7}$$

Generalizing this expression for non-integer case,

$${}_a\mathrm{D}_x^{\alpha} f(x) = \lim_{h \to 0} \frac{1}{h^{\alpha}} \sum_{m=0}^{[(x-a)/h]} (-1)^m \binom{\alpha}{m} f(x - mh) \tag{3.8}$$

We obtain the Grünwald-Letnikov fractional derivation.

- **Caputo Definition**

  At this point, two different formulation is given for fractional integral and derivative definitions. In the book [6], these definitions are covered in detail and it is shown that Riemann-Liouville and Grünwald-Letnikov definitions are identical. It is stated that with the enhancements in technology, the problem of initial states and derivative values at limit values emerged the need for well-established mathematical solution. Solution to the problem is proposed by Caputo in 1967[52].

  Caputo's definition for fractional order derivation is the extended version of Riemann-Liouville definition. This extension makes this definition able to handle initial states. When solving differential equations using this definition, defining initial states are not necessary. Caputo's fractional order derivative is defined as,

$${}_0\mathrm{D}_t^{\alpha} y(t) = \frac{1}{\Gamma(1-\gamma)} \int_0^t \frac{y^{m+1}(\tau)}{(t-\tau)^{\gamma}} d\tau, \tag{3.9}$$

  and fractional order integration is,

$${}_0\mathrm{D}_t^{\gamma} = \frac{1}{\Gamma(1-\gamma)} \int_0^t \frac{y(\tau)}{(t-\tau)^{1+\gamma}} d\tau, \tag{3.10}$$

  for $\alpha = m + \gamma$, m is an integer and $0 < \gamma \leq 1$.

3.1.2 Fractional Derivative and Integral Examples

Having reviewed the fractional order derivative and integral approaches, this section provides examples demonstrating the fractional integral and derivative concepts. Since, the previous section, mentioned that various definitions leads identical results. Therefore, in the examples will not use different definitions but proceed by implementing Grünwald-Letnikov definition for demonstration purposes.

- Sine Function

  In figure 3.2, the fractional derivative of sine function is given.

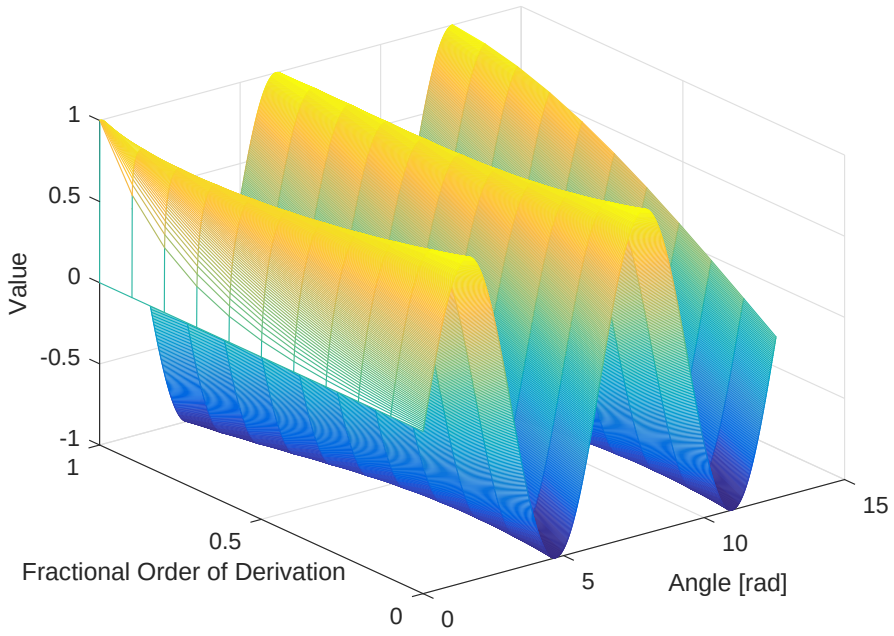

Figure 3.2: Fractional Derivative of sin(t) for α=0:0.1:1

So, the 3D figure is given with angle of the sin function, order of the fractional derivation and the function value representing three axis. Setting the order of derivation 0, the derivation of sine function is equal to itself.

$$ {}_0\mathrm{D}_t^0 sin(x) = sin(x) \tag{3.11}$$

This is called as identity property of the integro-differential operator. With

α becoming closer to 1, we see that ${}_0\mathrm{D}_t^0 sin(x)$ gets closer to cos(x) function.

$$ {}_0\mathrm{D}_t^1 sin(x) = cos(x) \tag{3.12}$$

This is exactly equal to integer order derivation of order one.

- Function Square

  In figure 3.3, the fractional derivative of $f(x) = x^2$ is given.

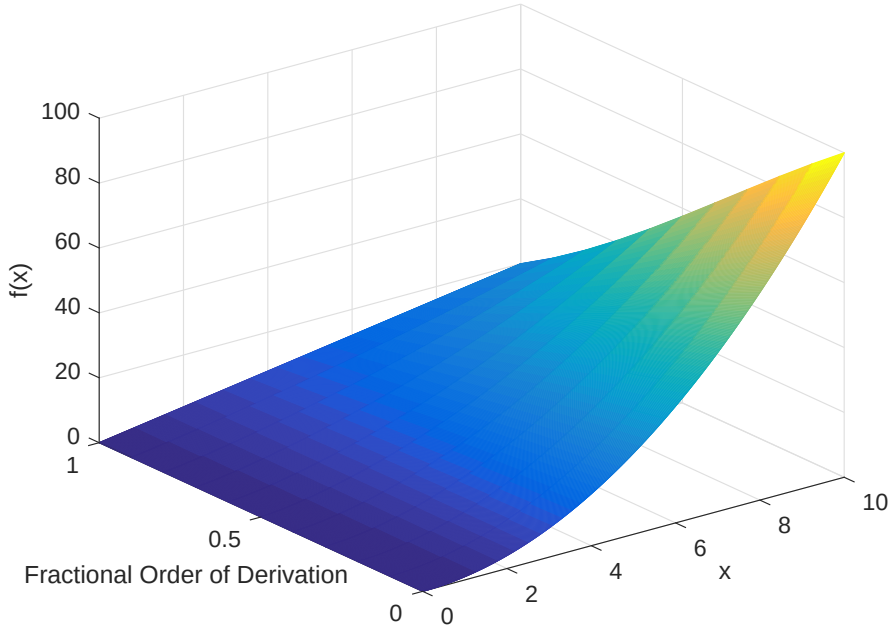

Figure 3.3: Fractional Derivative of x^2 for α=0:0.1:1

When order of the derivative is one, we get the integer order derivation of function which is $2x$, this is the straight line on the 3D plot where order of derivation is fixed to one. When the order is zero, the identity operator is obtained and resulting in x^2 which is the exponential edge of plot. For the fractional orders, as the order increases, the result gets closer to that of first order derivative.

Although meaning of fractional derivative and integral is not very clear, and is not well-explained in the literature, there are some leading author's views that need to be mentioned here. Podlubnly proposes two different approach for the

meaning of fractional integration, [4]. These are as follows,

- **The Fence and Shadow** To begin with Riemann-Liouville fractional integral of order α,

$$ {}_0\mathrm{D}_t^{\alpha} f(t) = \frac{1}{\Gamma(\alpha)} \int_t^0 f(\tau)(t-\tau)^{\alpha-1} d\tau \tag{3.13} $$

with $t > 0$, and we write this equation in the form,

$$ {}_0\mathrm{D}_t^{\alpha} f(t) = \int_t^0 f(\tau) dg_t(\tau) \tag{3.14} $$

where

$$ g_t(\tau) = \frac{1}{\Gamma(\alpha+1)} \left(t^{\alpha} - (t - tau)^{\alpha}\right) \tag{3.15} $$

so, if we take $t_1 = kt$ and $\tau_1 = k\tau$, we obtain following,

$$ g_{t_1}(\tau_1) = g_{kt}(k\tau) = k^{\alpha} g_t(\tau). \tag{3.16} $$

Drawing f(τ) on axes (τ,g,f) one can obtain the three-dimensional f(τ) curve with having different heights on the (τ,g) plane. The author calls this f(τ) as "fence", and points out that the fence has two projections on two different surfaces. These are,

  - The projection on (τ,f) surface which is,

$$ {}_0\mathrm{D}_t^{1} = \int_0^t f(\tau) d\tau \tag{3.17} $$

  - The projection on (g,f) surface which is the Reimann-Liouville fractional integral itself.

Therefore, the fence puts out two shadow on the wall. The first one, which is on the plane (τ,f) is the classical integral of the function f given in equation 3.16. The second one is the "shadow" on the plane (g,f) which meaning of the fractional order integral. In the figure below, the visualizations for the fence and shadows are given.

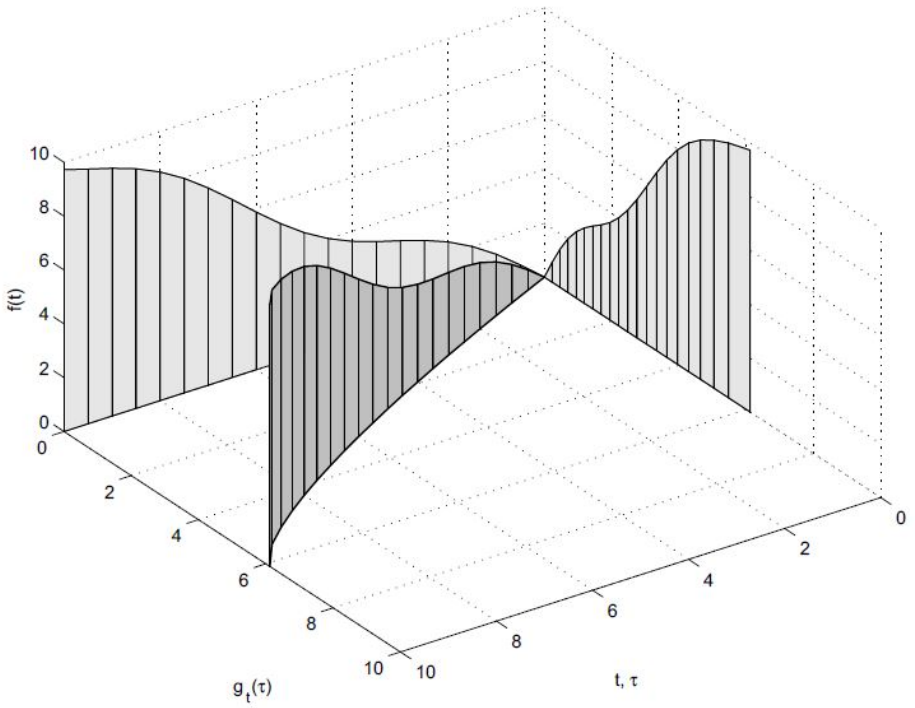

Figure 3.4: The fence and its shadows[4]

- **Two Kinds of Time - I**

  The classical calculusthat is to say the integer order calculus, is based on the acceptance of having time flowing equably. This assumption make us believe that time axis can be divided into equal intervals, scaled with equal time intervals. On the other hand, It is not possible to prove or reject this assumption by experiment. The measurement of time depends on counting ticks in spesific interval. This is the only observation of some process and there might be possible inhomogeneity of the time scale. In the figure below, the homogeneous and inhomogeneous time scales are given.

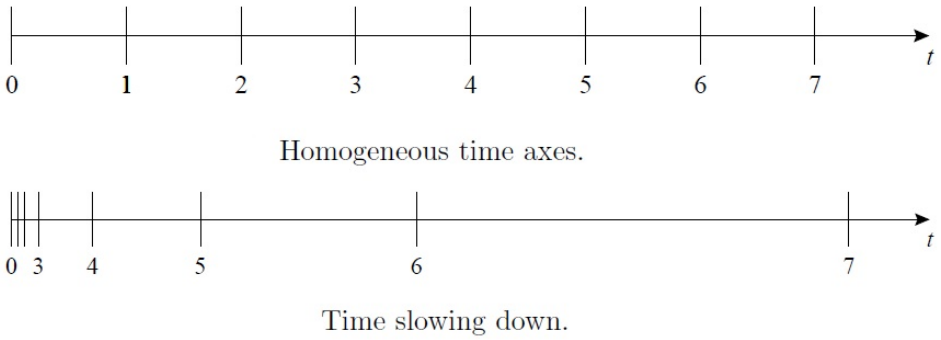

Figure 3.5: Homogeneous and Slowing Time illustration

Imagine a car with two different devices to measure velocity v(τ), and time τ. Also, assume that the driver in the car measures time incorrectly where the real time is function of incorrect measured time, $T = g(\tau)$. This means that the time interval measured by the driver is $d\tau$, and the real time interval is $dT = dg(\tau)$. If we consider the distance traveled by the car, according to driver the traveled distance will be defined as,

$$S_A(t) = \int_0^t v(\tau) d\tau \tag{3.18}$$

and the observer outside of the car will know about the real time and real distance traveled as,

$$S_O(t) = \int_0^t v(\tau) dg(\tau) \tag{3.19}$$

In these equations,$S_O(t)$ represents the real distance passed, $g(\tau)$ is the inhomogeneous time scale, and $v(\tau)$ is the local individual speed of moving object. At this point the author explains the physical interpretation of the Riemann-Liouville fractional derivative.

$$v(t) =_0 \mathrm{D}_t^{\alpha} S_O(t) \tag{3.20}$$

where ${}_0\mathrm{D}_t^{\alpha}$ denotes the Riemann-Liouville fractional derivative of the real distance passed.

Differentiating the equation 3.18, we can obtain the velocity of the car by observer, $v_O(t) = S'_O(t)$. If we define it with the velocity measured by the car,

$$v_O(t) = \frac{d}{dt}\, {}_0\mathrm{D}_t^{\alpha} v(t) =_0 \mathrm{D}^{(}1-\alpha)_t v(t) \tag{3.21}$$

which means the $(1-\alpha)$-th order derivative of the velocity $v(t)$ is equal to $v_O(t)$. If we assume that there are no inhomogeneity in the time scale and the function $T = g_T(\tau)$ and for the $\alpha = 1$, the velocities become equal to each other: $v_O(t) = v(t)$.

3.2 Fractional Order Systems

Having an understanding over fractional order differential equations, there are two ways to model systems. The comparison of integer order and fractional order modeling systems are worked in these papers [17],[46],[34]. All of the papers showed that fractional order models illustrates closer characteristics to the real system. Besides, in the paper [44] it is stated that integer order modeled systems are just some good, unconscious approximations of fractional order system. However, it must be noted that these unconscious approximations are valid only for small portions of the real systems.
Motivation behind using fractional order system modeling is that complex dynamical features of systems can be better modeled . Towards a demonstration of what is better we proceed by giving some examples of fractional order system models obtained with fractional order derivatives or integrals.

3.2.1 Examples

- Semi-Infinite Transmission Line
  Manabe first stated that the current flowing through semi-infinite cable can be modeled by non-integer order differential equation. The voltage $v(t)$ is applied to the one end of cable and, current $i(t)$ is measured over the semi-infinite cable.

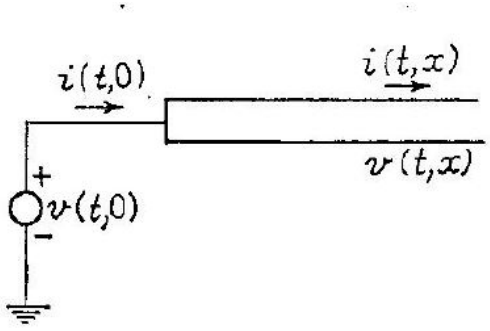

Figure 3.6: Semi-Infinite Transmission Line [5]

$$-\frac{\partial v}{\partial x} = Ri \qquad -\frac{\partial i}{\partial x} = C\frac{\partial v}{\partial t} \tag{3.22}$$

where, R and C are the resistances and capacitances of cable per unit length respectively. By doing some calculus and taking laplace transform, we obtain following equalities,

$$-\frac{\partial v^2}{\partial x^2} = R\frac{\partial i}{\partial x} \tag{3.23}$$

$$-\frac{\partial v^2}{\partial x^2} = RC\frac{\partial v}{\partial t} \tag{3.24}$$

$$-\frac{\partial V^2}{\partial x^2} = RCsV \tag{3.25}$$

and, the boundedness of voltage of semi-infinite cable while $x \to \infty$ is given as,

$$V(x) = V_0 e^{-x/\lambda} \tag{3.26}$$

then, using these equations one can obtain the current as

$$I(s,0) = \sqrt{\frac{C}{R}}\sqrt{s}V(s,0) \tag{3.27}$$

so, it is found that the current is half derivative of applied voltage times $\sqrt{C/R}$.

So, at this point, the full expression for $s^{\frac{1}{2}}$ is required to move on. In other works, fractional order impedance named as Fractance is presented [17],[18],[19]. The fractional order differential equation of the electrical circuit element fractance is the same as the Manabe's equation of the current flowing into semi-infinite cable.

- Heat Flux

  In the Fractional Differential Equations book [6], Podlubny gives Heat Flux as an example for calculating fractional order derivatives. Considering heat conduction for a semi-infinite body:

$$\hat{c}\hat{p}\frac{\partial T}{\partial t} = \hat{\lambda}\frac{\partial^2 T}{\partial x^2} \tag{3.28}$$

  where $t > 0$ and $-\infty < x < 0$.

$$T(0, x) = T_0 \tag{3.29}$$

$$T(t, 0) = T_{surf}(t) \tag{3.30}$$

$$\left|\lim_{x\to\infty} T(t, x)\right| < \infty \tag{3.31}$$

  where,

  - t : Time (s),
  - x : Spatial coordinate in the direction of the heat conduction (m),
  - $\hat{c}$: Heat Capacity ($\mathrm{J}kg^{-1}K^{-1}$),
  - $\hat{p}$: Mass Density ($\mathrm{kg}m^{-3}$),
  - T(t,x) : Temperature (K) ,
  - λ : Coefficient of the heat conductivity ($\mathrm{W}m^{-1}K^{-1}$)

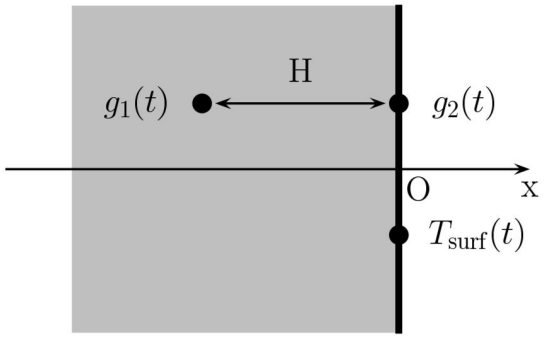

Figure 3.7: Blast Furnace Wall [6]

Define u(t,x) such that,

$$u(t, x) = T(t, x) - T_0 \tag{3.32}$$

which is the solution of the problem given in eqn 3.28 . Taking the Laplace transform the equation,

$$\hat{c}\hat{p}U(s,x) = \hat{\lambda}\frac{d^2U(s,x)}{dx^2} \tag{3.33}$$

the solution to this equation is,

$$U(s,x) = U(s,0)exp\left(x\sqrt{\frac{\hat{c}\hat{p}s}{\hat{\lambda}}}\right) \tag{3.34}$$

after taking its derivative with respect to x, we obtain,

$$\frac{dU}{dx}(s,x) = U(s,0)\sqrt{\frac{\hat{c}\hat{p}s}{\hat{\lambda}}}exp\left(x\sqrt{\frac{\hat{c}\hat{p}s}{\hat{\lambda}}}\right) \tag{3.35}$$

So, from these two equations, it is obtained that,

$$\frac{1}{\sqrt{s}}\frac{dU}{dx}(s,0) = \sqrt{\frac{\hat{c}\hat{p}s}{\hat{\lambda}}}U(s,0) \tag{3.36}$$

taking the inverse Laplace transform,

$${}_0\mathrm{D}_t^{-1/2}\frac{\partial u}{\partial x}(t,0) = \sqrt{\frac{\hat{c}\hat{p}s}{\hat{\lambda}}}u(t,0) \tag{3.37}$$

using the linearity of the fractional operator,

$$\frac{\partial u}{\partial x}(t,0) = \sqrt{\frac{\hat{c}\hat{p}s}{\hat{\lambda}}}{}_0\mathrm{D}_t^{1/2}u(t,0) \tag{3.38}$$

so, using the previously defined u(t,x), the relation for calculation of the heat flux at the point x=0 is found as,

$$q_A(t) = \sqrt{\hat{c}\hat{p}\hat{\lambda}}{}_0\mathrm{D}_t^{1/2}g(t) \tag{3.39}$$

where $g(t) = T_{surf}(t) - T_0$ and $q_A(t) = \hat{\lambda}\frac{\partial T}{\partial x}(t,0)$ gives the resulting heat flux.

- CRONE Suspension

  CRONE Suspension model is obtained by replacing traditional suspension model with a model that includes non-integer derivation. In this model, mechanical and hydropneumatic system is defined using fractional order operators [20].

  First, consider one tire of the car and its suspension system,

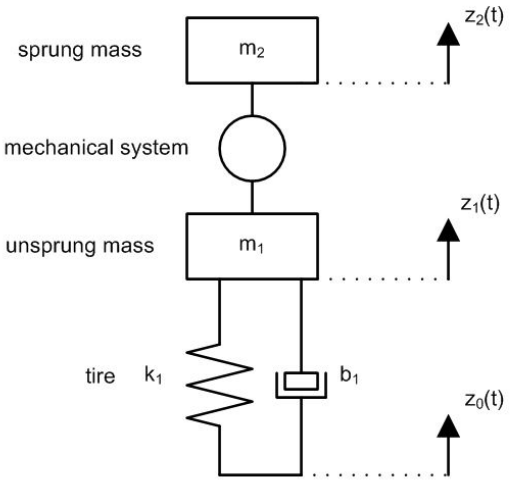

Figure 3.8: Suspension System on a tire

where,

- m_i : Mass carried by each wheel, quarter of total mass
- k_i : Stiffness of the spring
- b_i : Damping coefficient

for i=1, CRONE Suspension and for i=2, the traditional suspension model are considered. In the figure 3.8, $z_0(t)$ shows the deflection of the road,$z_1(t)$ and $z_2(t)$ show the displacement of wheel and the body respectively.
Let us define the relation between CRONE suspension force with the relative displacement between $z_1(t)$ and $z_2(t)$, calling it $z_{12}(t)$,

$$F_2(s) = C(s)Z_{12}(s) \tag{3.40}$$

where,

$$Z_{12}(s) = Z_1(s) - Z_2(s) \tag{3.41}$$

From the fundamental laws of dynamics,

$$m_1\dot{z}_1(t) = f_1(t) - f_2(t) \tag{3.42}$$

$$m_2\dot{z}_2(t) = f_2(t) \tag{3.43}$$

$$f_1(t) = k_1[z_0(t) - z_1(t)] + b_1[\dot{z}_0(t) - \dot{z}_1(t)] \tag{3.44}$$

with $f_2(t)$ is the force developed by suspension.
Taking the Laplace Transforms of the equations 3.42, 3.43, 3.44

$$m_1 s^2 Z_1(s) = k_1(Z_0(s) - Z_1(s)) + b_1 s(Z_0(s) - Z_1(s)) - F_2(s) \tag{3.45}$$

$$m_2 s^2 Z_2(s) = F_2(s) \tag{3.46}$$

replacing $F_2(s)$ with equation 3.40,

$$m_1 s^2 Z_1(s) = k_1(Z_0(s) - Z_1(s)) + b_1 s(Z_0(s) - Z_1(s)) - C(s)(Z_1(s) - Z_2(s)) \tag{3.47}$$

$$m_2 s^2 Z_2(s) = C(s)(Z_1(s) - Z_2(s)) \tag{3.48}$$

Transmittance are defined as follow to analyze the vibration insulation,

$$T_2(s) = \frac{Z_2(s)}{Z_1(s)} = \frac{C(s)}{m_2 s^2 + C(s)} \tag{3.49}$$

and,

$$S_2(s) = \frac{Z_{12}(s)}{Z_1(s)} = \frac{m_2 s^2}{m_2 s^2 + C(s)} \tag{3.50}$$

CRONE Suspension method is based on the interpretation of the transmittances $T_2(s)$ and $S_2(s)$. This synthesis method is given as,

$$T_2(s) = \frac{\beta(s)}{1 + \beta(s)} \tag{3.51}$$

$$S_2 s = \frac{1}{1 + \beta(s)} \tag{3.52}$$

where,

$$\beta(s) = \frac{C(s)}{m_2 s^2} \tag{3.53}$$

β(s) is defined as open-loop transmittance here. In the CRONE suspension, transmittance is defined as,

$$\beta(s) = \left(\frac{\omega_u}{s}\right)^n \tag{3.54}$$

where the C(s) is defined as,

$$C(s) = \left(\frac{s}{\omega_0}\right)^m \tag{3.55}$$

with m=2-n and 0<n<1. Then, equation 3.19 becomes,

$$F_2(s) = \left(\frac{s}{\omega_0}\right)^m [Z_1(s) - Z_2(s)] \tag{3.56}$$

in the time domain,

$$f_2(t) = \frac{1}{\omega_0^m}\left(\frac{d}{dt}\right)^m [z_1(t) - z_2(t)] \tag{3.57}$$

so, it is obtained that the force generated by CRONE Suspension is proportional to the fractional derivative of relative displacement. m is determined by constrained optimization and found as 0.8, and other parameters can be found in the article.

3.3 Realization and Approximation Methods

When the definitions of fractional order derivatives and integrals are considered, it is seen that as t grows, more and more terms are added to the computation. This means that for solving fractional differential equations, an infinite memory is required. In the literature, *The Short Memory Principle* is proposed [10],[53]. This principle basically approximates the numerical solution by taking only the "recent past". The time interval is restricted as [t-L,t] with L being the length of memory.

Consequently, many approximation approaches can be implemented. Thus, in the literature there are several methods proposed for the approximation of fractional order operator and finite dimensional models are proposed for infinite dimensional systems. When real applications of fractional order operators are taken into account, these finite models with limited memory must be within realizable form.

The essential step for implementing fractional operator is the discretization condition. In the literature, several discretization rules are proposed. These methods can be classified in two distinct types called as Direct and Indirect

Realization methods as listed in table 3.1 . Among direct Discretization Methods,power series expansion and continuous fractional expansion are applied to the fractional operator by generating functions which directly replace s^α. Depending on the method itself, these functions takes different forms. In Indirect Discretization Methods, first, the fractional operator is approximated in continuous time domain, then this approximated function is discretized. The details of these methods will be given in following subsections.

Direct Discretization Methods	Indirect Discretization Methods
Forward Euler Rule	Oustaluop's Methdod
Al-Alaoui Rule	General CFE
Tustin Rule	Carlson's Method
Backward Euler Rule	Matsuda's Method
Implicit Adams Second Rule	

Table 3.1: Realization and Approximation Methods

3.3.1 Direct Discretization Methods

In general, the fractional operator s^α is represented by a generating function. Generating functions are varying according to the methods used. The general form of the generating function is given as,

$$H\alpha(z) = (\frac{1}{T}\frac{1-z^{-1}}{\gamma+(1-\gamma)z^{-1}}) \tag{3.58}$$

where γ is the fractional constant between 0 and 1 that determines the final form of generating functions. Here are the generating functions of direct discretization methods,

The generating function $w(z^{-1})$ is obtained and there is a need for discretizating this function which has a fractional exponent. In direct discretization methods, there are mainly two methods called Power Series Expansion and Continuous Fraction Expansion.

Discrete Approximation using Power Series Expansion: Fractional deriva-

Direct Discretization Methods	γ	s→z
Forward Euler Rule	0	$s^\alpha \approx (\frac{1}{T}\frac{1-z^{-1}}{z^{-1}})^\alpha$
Al-Alaoui	7/8	$s^\alpha \approx (\frac{8}{7T}\frac{1-z^{-1}}{1+z^{-1}/7})^\alpha$
Tustin	1/2	$s^\alpha \approx (\frac{2}{T}\frac{1-z^{-1}}{1+z^{-1}})^\alpha$
Backward Euler Rule	1	$s^\alpha \approx (\frac{1-z^{-1}}{T})^\alpha$
Implicit Adams Second Rule	3/2	$s^\alpha \approx (\frac{2}{3T}\frac{1-z^{-1}}{z^{-1}/3})^\alpha$

Table 3.2: Direct Discretization Methods

tive of order α can be expressed as,

$$\mathrm{D}^{\pm\alpha}(z) = \frac{Y(z)}{F(z)} = T^{\pm\alpha} PSE\{(w(z^{-1}))^{\pm\alpha}\} \tag{3.59}$$

where T is sample period and $\mathrm{D}^{\pm\alpha}$ is discrete equivalent of fractional order operator. $w(z^{-1})$ can be determined according to generating functions as seen in table 3.2.

Discrete Approximation using Continuous Fraction Expansion: Fractional derivative of order α can be expressed as,

$$\mathrm{D}^{\pm\alpha}(z) = \frac{Y(z)}{F(z)} = T^{\pm\alpha} CFE\{(w(z^{-1}))^{\pm\alpha}\}_{p,q} \tag{3.60}$$

$$\mathrm{D}^{\pm\alpha}(z) = T^{\pm\alpha} \frac{P_p(z^{-1})}{Q_q(z^{-1})} \tag{3.61}$$

where P and Q are polynomials with degree p and q. In the table 3.2, The Tustin Rule is defined when choosing γ=1/2. The paper [54],[55],[56]have shown that the fractional order differentiator can be obtained by a recursive formula. In the following equations, the recursive formula is explained.
By taking γ=1/2, we obtain the generating function as,

$$(w(z^{-1}))^\alpha = \left(\frac{2}{T}\right)^\alpha \left(\frac{1-z^{-1}}{1+z^{-1}}\right)^\alpha \tag{3.62}$$

which can also be defined as,

$$(w(z^{-1}))^\alpha = \left(\frac{2}{T}\right)^\alpha \lim_{n\to\infty} \frac{A_n(z^{-1},\alpha)}{A_n(z^{-1},-\alpha)} \tag{3.63}$$

where $A_0(z^{-1},\alpha)$=1 and,

$$A_n(z^{-1},\alpha) = A_{n-1}(z^{-1},\alpha) - c_n z^n A_{n-1}(z,\alpha) \tag{3.64}$$

and

$$c_n = \begin{cases} r/n & n \text{ is odd} \\ 0 & n \text{ is even} \end{cases} \tag{3.65}$$

Thus,

$$s^\alpha \approx \left(\frac{2}{T}\right)^\alpha \frac{A_n(z^{-1}, \alpha)}{A_n(z^{-1}, -\alpha)} \tag{3.66}$$

The recursive computation of A_n leads to the computation of s^α under a defined order of approximation n.

3.3.2 Indirect Discretization Methods

Firstly, approximation of fractional operator in continuous time is obtained then 's' are converted into discrete one for implementation purposes.

- **Carlson's Method** by Carlson [22] is defined as;

$$H(s) = G(s)^\alpha \tag{3.67}$$

$$H_i(s) = H_{i-1}(s)\frac{(q-m)(H_{i-1}(s))^2 + (q+m)G(s)}{(q+m)(H_{i-1}(s))^2 + (q-m)G(s)} \tag{3.68}$$

  where α=1/q, m=q/2, and initial condition $H_0 = 1$.

- **Oustaloup's Method** is given below [57],[58]

$$H(s) = s^\alpha \tag{3.69}$$

$$\hat{H}(s) = C \prod_{k=-N}^{N} \frac{s + \omega_k}{s + \omega_k^{'}} \tag{3.70}$$

  where,

$$\omega_k^{'} = \omega_b \left(\frac{\omega_h}{\omega_b}\right)^{\frac{k+N+1/2(1-\alpha)}{2N+1}}, \quad \omega_k = \omega_b \left(\frac{\omega_h}{\omega_b}\right)^{\frac{k+N+1/2(1+\alpha)}{2N+1}}, \quad K = \omega_h^\alpha. \tag{3.71}$$

  where $0 < \alpha < 1$ and $s^\alpha = s^n s^\alpha$, equation holds for any n integer value.

- **General Continuous Fraction Expansion** is a method for computing functions that are irrational and their approximation G(s) can be expressed in the following form [22],

$$G(s) \approx a_0(s) + \cfrac{b_1(s)}{a_1(s) + \cfrac{b_2(s)}{a_2(s) + \cfrac{b_3(s)}{a_3(s) + \ldots}}} \tag{3.72}$$

$$G(s) = a_0(s) + \frac{b_1(s)}{a_1(s)} + \frac{b_2(s)}{a_2(s)} + \frac{b_3(s)}{a_3(s)} + \ldots \tag{3.73}$$

where $a_i(s)$ and $b_i(s)$ are rational functions of s. When the method is applied, the approximation of the fractional operator can be obtained as G(s). So, in general, the approximation of the function s^α can be obtained by performing the CFE of the functions:

$$G_h(s) = \frac{1}{(1+sT)^\alpha} \tag{3.74}$$

$$G_l(s) = (1 + \frac{1}{s})^\alpha \tag{3.75}$$

where G_h is the approximations for higher frequencies, and G_l is the approximation for lower frequencies.

- **Matsuda's Method** is based on the approximation of the function obtained from Continuous Fraction Expansion such as[22];

$$H(s) = a_0 + \frac{s - s_0}{a_1} + \frac{s - s_1}{a_2} + \frac{s - s_2}{a_3} + \ldots \tag{3.76}$$

where

$$a_i = v_i(s) \tag{3.77}$$

$$v_0(s) = H(s) \tag{3.78}$$

$$v_{i+1}(s) = \frac{s - s_i}{v_i(s) - a_i} \tag{3.79}$$

3.4 Fractional Order Control Applications

In this section, we will define the fractional order systems by using fractional order differential equations. Control theory and applications are one of the areas

where fractional calculus is used. General formulations of the systems that are defined with fractional order differential equations, their general stability theorem, general controller structures are covered in this section.
A general formulation of the fractional order differential equation can be expressed as,

$$a_n D^{\alpha_n} y(t) + a_{n-1} D^{\alpha_{n-1}} y(t) + \cdots + a_0 D^{\alpha_0} y(t) = \\ b_m D^{\beta_m} u(t) + b_{m-1} D^{\beta_{m-1}} u(t) + \cdots + b_0 D^{\beta_0} u(t) \quad (3.80)$$

where a_n and b_m are constants for $a_k (k = 0, \ldots, n)$ and $b_k (k = 0, \ldots, m)$. By taking the Laplace transform, one can the obtain the transfer function in the following format

$$G(s) = \frac{Y(s)}{U(s)} = \frac{b_m s^{\beta_m} + b_{m-1} s^{\beta_{m-1}} + \cdots + b_0 s^{\beta_0}}{a_n s^{\alpha_n} + a_{n-1} s^{\alpha_{n-1}} + \cdots + a_0 s^{\alpha_0}} \quad (3.81)$$

The discretized z-domain representation of the transfer function is then obtained as,

$$G(z) = \frac{b_m (\omega(z^{-1}))^{\beta_m} + b_{m-1} (\omega(z^{-1}))^{\beta_{m-1}} + \cdots + b_0 (\omega(z^{-1}))^{\beta_0}}{a_n (\omega(z^{-1}))^{\alpha_n} + a_{n-1} (\omega(z^{-1}))^{\alpha_{n-1}} + \cdots + a_0 (\omega(z^{-1}))^{\alpha_0}} \quad (3.82)$$

where $(w(z^{-1}))$ denotes the discrete equivalent of the Laplace operator s which is described in the previous section. Thus, choosing $w(z^{-1})$ and using it in the equation 3.82, givens the fully discretized form of the transfer function. At this stage, this function is in the form that it can directly be implemented.

3.4.1 Stability of Fractional Order Systems

Matignon's Stability Theorem [49] states that the fractional transfer function G(s) is stable if and only if the following condition holds,

$$|arg(\sigma)| > q\frac{\pi}{2} \quad (3.83)$$

for every σ in complex plane C, and $\sigma := s^q$. This theorem changes the well-known stability condition for integer order LTI systems which is determined by the roots of the characteristic polynomial being in the left half plane. However, in the fractional order case, the stability definition expands to the right half plane. The figure 3.9 illustrates the fractional order stability conditions derived by Matignon's Theorem.

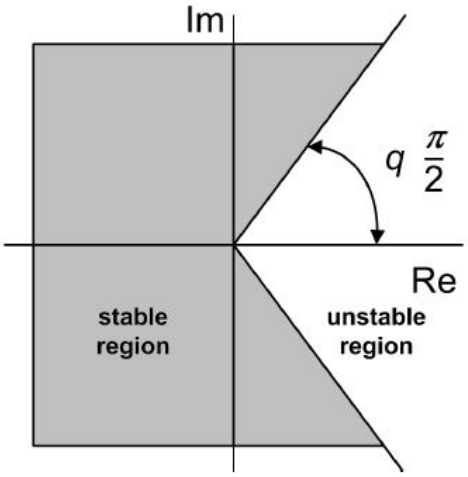

Figure 3.9: Stability Region for Fractional Order Systems [7]

Consider teh fractional order system in the following form,

$$D^q\omega = f(\omega) \tag{3.84}$$

where q is the fractional order between (0,1) and $\omega \in \mathbb{R}^n$. The equilibrium points of the system can be found solving following equation,

$$f(\omega) = 0 \tag{3.85}$$

The equilibrium points are asymptotically stable if all eigenvalues λ_j satisfy following condition,

$$|\angle(eig(\mathbf{J}))| = |\angle(\lambda_j)| > q\frac{\pi}{2}, \quad j = 1, 2, \ldots, n \tag{3.86}$$

where $\mathbf{J} = \partial f/\partial\omega$ evaluated at the equilibrium.

3.4.2 Fractional Order Controllers

PID Controllers, which are commonly used in control systems, can also be defined fractionally as $PI^{\lambda}D^{\mu}$ [34]. These controllers undergo the following parametrization,

$$G_c(s) = \frac{U(s)}{E(s)} = K_p + K_I s^{-\lambda} + K_D s^{\mu} \tag{3.87}$$

and, their output can be defined in terms of the error function in time domain as,

$$u(t) = K_p e(t) + K_I D^{-\lambda} e(t) + K_D D^{\mu} e(t). \tag{3.88}$$

In these equations, if we let $\lambda = 1$ and $\mu = 1$, we get the classical PID controllers.

In addition to PID controllers, in fractional order control history, the tilted integral derivative (TID) controller was proposed in [45] the CRONE Controllers are proposed with three generations [46], lead and lag compensator design is covered in [47].

- TID Controller

  In TID controller, the proportional unit of the classical PID Controller is replaced with transfer function of $s^{\frac{-1}{n}}$. The resulting transfer function of the entire system was able to illustrate improved performance in terms of simpler tuning, better disturbance rejection ratio and more robustness to plant parameter variations on closed loop response.

  The block diagram of the TID control scheme is given in figure 3.10,

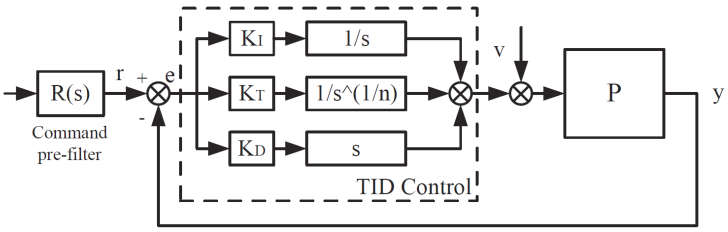

Figure 3.10: TID Controller Scheme [8]

In [45] and [8], more detailed analysis of the controller is given.

- CRONE Controller

  CRONE stands for french abbreviation of "Contrôle Robuste d'Ordre Non Entier". The CRONE control was introduced early 90s, and have three generations. The first generation of CRONE Controller is based on a constant phase of the controller around open loop gain crossover frequency ω_{cg} [23]. The band-limited real fractional differentiator transfer function is,

$$C(s) = C_0 \left(\frac{1 + s_{\omega_l}}{1 + s_{\omega_h}} \right)^n \tag{3.89}$$

  The perfect robustness of the phase margin is achievable if ω_{cg} is within constant phase frequency range of the plant.

  The second generation of CRONE Control arise from the problem of choosing ω_{cg} within an asymptotic behavior frequency band of the plant. The fractional C(s) is defined as,

$$C(s) = \beta(s)/G_0(s) \tag{3.90}$$

  where $G_0(s)$ is the plant frequency response and $\beta(s)$ is the nominal open-loop function defined as,

$$\beta(s) = K \left(\frac{\omega_l}{s} + 1 \right)^{n_l} \left(\frac{1 + \frac{s}{\omega_h}}{1 + \frac{s}{\omega_l}} \right)^n \left(1 + \frac{s}{\omega_h} \right)^{-n_h} \tag{3.91}$$

  The third generation of the CRONE Control is defined for the more general cases where there are phase variations of the plant. For the third generation CRONE Controller, the following open loop transfer function is introduced [49],

$$\beta(s) = K \left(\frac{\omega_0}{s} + 1 \right)^{n_1} \frac{1}{(1 + s/\omega_1)^{n_h}} \left[\left(\frac{1 + s/\omega_1}{1 + s/\omega_0} \right)^{a_0} Re_{/i} \left(\left(C_0 \frac{1 + s/\omega_1}{1 + s/\omega_0} \right)^{ib_0} \right)^{-sign(b_0)} \right] \tag{3.92}$$

  In this study [49], it is shown that the CRONE control ensures better robustness to uncertainties over classical PID and H_∞ controllers.

- Fractional Lead-Lag Compensator

  With the use of fractional integration and derivation, it is possible to

extend the classical lead-lag compensator to the fractional order case [48]. The fractional lead-lag compensator can be defined as,

$$C_r(s) = C_0 \left(\frac{1 + s/\omega_b}{1 + s/\omega_h} \right)^r \tag{3.93}$$

In the study [50] the CRONE controller, fractional order $PI^{\lambda}D^{\mu}$, TID controller and the fractional lead-lag compensator are compared. It is stated that fractional order $PI^{\lambda}D^{\mu}$ controller has a potential to achieve better performances over classical PID controllers. As PID control is very popular in many industry, development of $PI^{\lambda}D^{\mu}$ is strongly desired.

In our studies, we will work on fractional order $PI^{\lambda}D^{\mu}$ controllers. In the Chapter 5, the design and implementation procedures of these controller are discussed. The tuning procedures of fractional $PI^{\lambda}D^{\mu}$ controllers are covered in detail with illustrations. In the chapter 6, the closed loop responses of designed controllers are discussed with the identified system model.

CHAPTER 4

PAN-TILT PLATFORMS

4.1 Introduction to Pan-Tilt Platforms

Although defense systems can work in various environments leading to different main objectives, most of the systems have components aimed to surveillance and navigation. For example, modern gun-turret systems would achieve high shooting success if these systems have an integrated target tracking system. Similarly, the rate of success of guidance techniques crucially depends on the performance of the target tracking and navigation systems.
Acquiring high performance on target tracking systems is directly related with ensuring the tight requirements which are defined for several other disciplines such as;

- Structural stiffness of mechanical body must be good enough to allow system work properly.
- Controllers must be well designed and tuned such that the system works under any disturbances and keeps line of sight steady.
- The choice of tracking algorithms are crucial so that the system does not lose the track of target in working environment such as sea, ground or air targets.
- The electronic hardware and drivers must be capable of running all the algorithms with the smallest amount of delay.

- Test Scenarios, and effective testing of hardware and software plays tremendous role on systems' performance on the field. All the hardwares and softwares in the system must be tested and validated in order to avoid failures.

- Production techniques should be well established and described in detail for mass production.

Therefore, systems are expected to fully satisfy the abilities affected by above listed requirements from the very early stages of system design.
A pan-tilt platform is a mechanical structure that helps satisfy and test the requirements of surveillance, navigation and target tracking. In order to provide abilities to such tasks, these platforms are composed of bearings, motors, gyroscope(s), and optical elements as load. A pan-tilt platform can be inertially stabilized using gyroscopes. Inertial stabilization is a must-have feature for almost all the applications that the platform is used for. Line of sight stabilization is a common goal for all inertially stabilized platforms. Depending on the applications, line of sight can vary as aimpoint of laser beam, center of the field of view where target tracking occurs, or the direction that a optical sensor is pointed.
Figure 4.1 illustrates the line of sight (LOS) stabilization with a moving target when the platform itself has a dynamics.

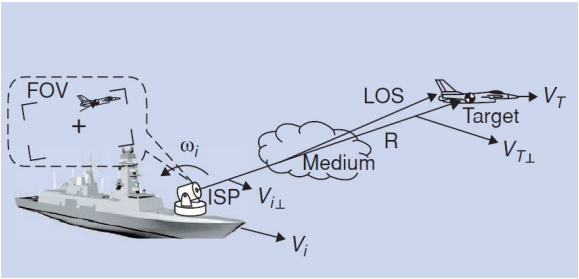

Figure 4.1: A Typical LOS application [9]

The main aim is to hold LOS stationary under the platform dynamics and disturbances. For pointing applications, the line of sight is defined by the field of view of the sensors that includes the target.

When the applications of pan-tilt platforms are considered, there are several requirements that have to be satisfied to obtain the desired performance.

- In most of the LOS applications, two-axis gimbal is required. These two angles are pitch and yaw angle which are necessary to put target in our line of sight. Stabilization of the pitch and yaw disturbances are needed in these applications, the roll angle does not have to be implemented.However, if there is a need for orientation of the image, or the rotation of the image about LOS causes excessive motion at the edges of the FOV, the third axis of the control is needed and must be implemented. These angles are shown on figure 4.2.

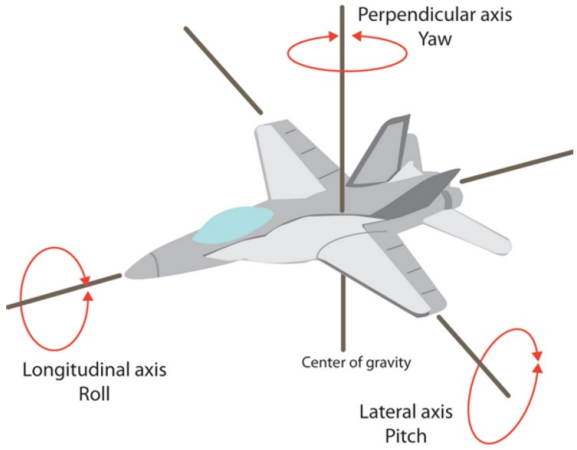

Figure 4.2: Yaw, Pitch and Roll angles [9]

Two axes must be orthogonal to each other in order to manage isolation of base motions. The inertial sensors, such as gyroscope must be placed on the point of gimbal of which the LOS is stabilized. This mounting must be also orthogonal to moving axes but needed to be stabilized. For example, if a gimbal is wanted to have two axes of stabilization, it should have traverse and elevation actuator, and the gyroscope has to be aligned to these axes.

- One of the most important parameter is the closed-loop bandwidth. This parameter determines the performance of the system on how well disturbances are rejected or how well the target is tracked, in other words how

well the desired inputs are followed by control commands. The closed loop bandwidth depends on dynamical characteristics of the system, the torsional responses of the structure, its equipped abilities based on the inertial sensors, the actuators and the control algorithms used. So, both mechanical and programming issues play tremendous role on determining the closed loop bandwidth.

- As pointed out before,the structural dynamics of the pan-tilts affect almost all outputs of the system shaping all performance related criteria. There is an important issue to keep in mind while designing pan-tilt platforms: The first natural resonance plays a crucial role on bandwidth. With higher resonance frequencies, we obtain higher closed loop bandwidths. As much as the mechanical structure of the gimbal, the place where it is mounted is also critical towards better performance and higher closed loop bandwidth. The structure of the system should be mounted on a platform that does not have any base vibration that can excite jitters on the gimbal. Especially, the frequency of the resonance which is caused by the base or connection elements should be lower than the system's first natural resonance frequency.

- The alignment of the optical sensors, encoders, gyroscopes, gun-turret subsystems must be integrated without any mistakes. The smallest error in alignment would cause the whole system to fail, making the objectives of target detection, target tracking or guiding a missile, impossible to reach.

4.2 Introduction to Hardware

The test platform has designed to do evaluations on stabilization and video tracking performance. It is composed of a High Accuracy Stabilized Gimbal, 6-DOF stewart platform for simulating external disturbances, power supply, human control unit, monitor. Test platform is equipped with projector to project real target videos on the curved screen so that one can have a full view of the real life target videos recorded for the region surveyed together with video tracking ability where performance tests can be applied.

In figure 4.3, test platform is depicted which ables us to compare different controllers in terms of stabilization and video tracking performances.

Figure 4.3: HASG Simulation Platform (Photo Courtesy of ASELSAN Inc.)

During the tests, in order to compare results, a real time data must be collected and processed. We use MATLAB's real-time windows target to collect data from Servo Control Unit which can provide any information we desire. The test serial channel has RS422 interface with 1khz data rate and 921600 bits/seconds baud-rate. For the serial channel data collection the following structure is used,

Figure 4.4: Data collection procedure

The servo control unit sends the required informations on RS422 interface. Then, we use the BrainBox RS422 to USB converter to be able to collect datas on the computer. The MATLAB R2011b is used for data collection. In the following sections whenever we call collected data, those are collected from the system with this serial channel.

- **High Accuracy Stabilized Gimbal (HASG)**
  ASELSAN uses a two-axes gimbal named HASG-High Accuracy Stabilized Gimbal which is also used in my thesis work for which we will design a fractional order controller. All of experimental results are analyzed based on this hardware system. In this section, the system will be introduced in detail with its features.

Figure 4.5: HASG:High-Accuracy Stabilized Gimbal (Photo Courtesy of ASELSAN Inc.)

High-Accuracy Stabilized Gimbal is used in navigation,surveillance, aerial defense, hunter-killer applications, missile guidance and many more applications. Any platform equipped with such a gimbal can carry different payloads according to a desired mission. In figure 4.6, the mechanical and electrical interface of HASG, the one used in both simulations and hardware tests, is given. There two direct drive motors mounted on each axis

and aligned with encoders. The gyroscope is mounted on the elevation axis, where optical sensors are mounted. Therefore, we will be able to stabilize the line of sight using gyroscope feedback.

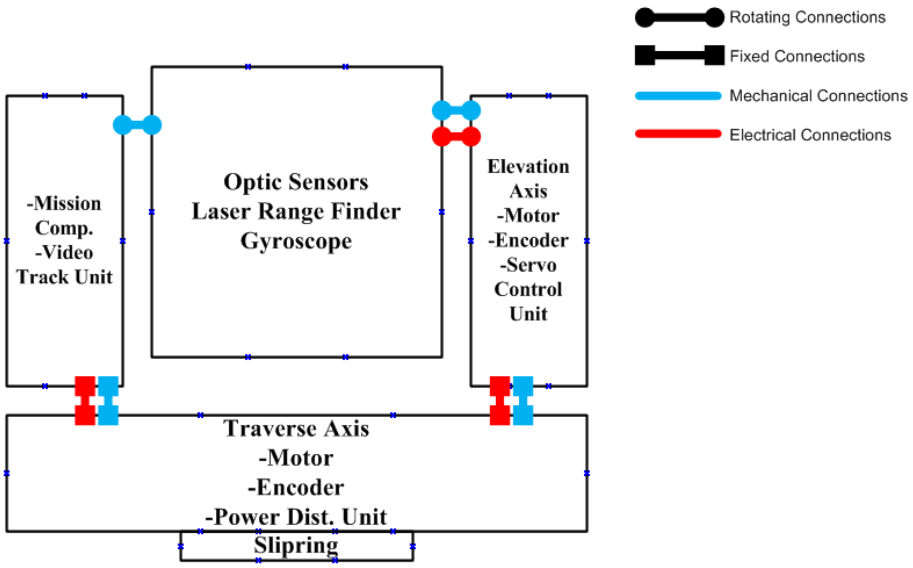

Figure 4.6: HASG:High-Accuracy Stabilized Gimbal and its Electrical and Mechanical Interfaces

The workflow of the system begins with the Human Control Unit. As it can be seen on the figure 4.7, the ON/OFF rotary switch turns the system on and off. Firstly,the Mission Computer is triggered ensuring that all the other units are accessible. All the commands to other units are determined in this computer. These commands include optic sensor related commands, servo commands, tracker related commands and all others such as built-in tests.

The general work-flow of the system is given in figure 4.7,

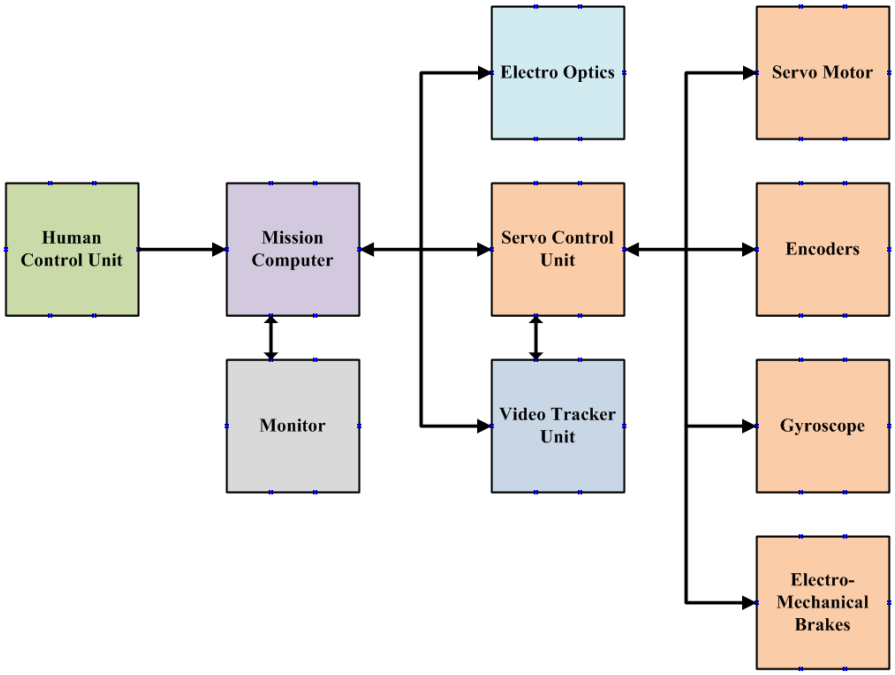

Figure 4.7: Workflow of High-Accuracy Stabilized Gimbal System

- **Mission Computer:** The Mission Computer includes the main processor which is responsible for the communication between all components managing every states of the whole system. The opening and closing scenarios are also managed by this computer by monitoring certain enable signals coming through switches on the human control unit.
  The Mission computer takes all the CAN messages from the human control unit, takes all the feedbacks messages from optical sensors, servo control unit and video track unit and process these messages to determine the next state of the system depending on user input commands. For example, If user is controlling HASG in surveillance mode, the joystick commands coming to mission computer are shaped and sent to the servo control unit as specific velocity demands in degree/sec. The shaping depends on the how hard user press the turret aiming button on the joystick. Mission computer can also control optical sensors while turning the system. Zoom in and out, or switching between cameras are independent of moving turret. In all of these functions, the user observes the current state on the

monitor.

- **Electro Optics:** Electro-optics provide the user with the *picture* of the region surveyed that lead the process to find targets location. In High-Accuracy Stabilized Gimbal (HASG), there are two cameras working with different wavelengths such as day and thermal camera. These cameras allow to work on different conditions, such as asymmetric targets in sea environment or air environment. In addition to these two cameras, HASG has one laser range finder unit and one laser pointer. These are used to measure the target distance.

  Communication with optical sensors and laser range finder is done with RS422 interface. Mission computer directly communicates with these units. The video outputs coming out of optics are sent to video tracker unit to be processed. These videos outputs are analog that carried in specific cable designed for this purpose.

- **Video Tracker Unit:** This unit processes the picture obtained by electro-optics with the aim of finding the target on the video outputs. The algorithms may vary for different environment and target types. The High-Accuracy Stabilized Gimbal uses mainly the sea and air algorithms as it is designed to be used on a vessel. The choices of detection and tracking algorithm are critical to ensure that video tracking works properly.

  Video tracker unit has an ethernet interface with mission computer, and an RS422 interface with the servo controller unit. It takes all the camera related commands from the mission computer and by RS422, it communicates with the servo control unit and sends the required information for video tracking. These informations are the bore-sight errors, target velocity, the tracker mode and tracking delay. Bore-sight errors are measured with using gimbal's line of sight position and the target's position. Having zero bore-sight error means gimbal is directly aims to the target and, the target in centralized on the monitor.

  The outputs of video track unit are directly used in orienting the platform towards the target. Servo controller takes the bore-sight error generated by video tracking unit in order to use as position error in the position loops

of the controller. As controller minimizes the error, turret rotates to the target. The delays in this generation of the error process has a decisive role in target tracking performance. Thus, delay should be minimized and measured correctly. If camera and video tracking delays exceed certain amount, the turret might have oscillations on steady state.

- **Servo Control Unit:** Servo controller gets commands from the mission computer and video tracking unit depending on the task. If the task is the tracking, the command is taken as error from the video track unit. For other tasks like home position or surveillance mode, commands are taken as speed or position commands from the mission computer. Servo controllers are responsible for computing the required current to drive motors to fulfill the desired command. While doing this, there are other challenges such as non-linearities in system,external disturbances, time delays, quantization errors, etc. that servo control unit must handle.
  Servo software is coded and run in DSP module. As motor driver and controller, ASELSAN's own design Herkul-04D is used given in figure 4.8. The fractional order controllers are implemented on this controller unit. Details about designing of fractional order controllers will be given in following chapter.

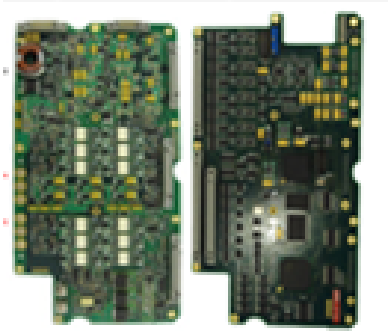

Figure 4.8: Herkul-04D Servo Controller (Photo Courtesy of ASELSAN Inc.)

We have also Herkul GUI to communicate with servo control unit directly. In order to send some special commands or parameter change, we use this GUI. It must be noted that when we use this infrastructure, the mission

computer is turned off. Servo unit takes all the commands via CAN bus. This enables us to listen command while system is operating, however, in order to make changes at servo control unit, mission computer must be turned off and we are needed to be host to send commands. The basic functions of this GUI is to change any parameter in system including controller parameters, sending torque, speed or positio commands, configuring test serial channel to choose which data to collect and etc. In the following sections, for example, we will compare step responses of real system and simulations. All the real system related commands are sent through this GUI except video tracking. In order to send video tracking commands, all system units must work together.

- **Direct Drive Motors:** Direct drive motors are used as actuators in HASG for performance requirements. Their high torque outputs and high efficiencies are the main reason for selecting these motors. There are two direct drive motor, one for traverse and for elevation axis. These motors are driven by the servo control unit and their own motor driver.

Figure 4.9: Direct Drive Motor

- **Encoders:** Encoders are used for position feedback sensor. In order to obtain high accuracy in positioning, high resolution encoders are used for both axes. These encoders can provide 0.000171^o resolution. Servo control unit reads the data signals coming from the encoder synchronously.

Figure 4.10: Encoder

- **Gyroscope:** Gyroscope is used for speed feedback in inertia frame. Gyroscope is mounted on elevation axis. For stabilization performance, this unit plays tremendous role. The noise level on the gyroscope is the determinant factor on the performance.
  There is an asynchronous RS422 interface between gyroscope and the servo control unit. Servo control unit is responsible for reading gyro signals.

- **Power Distributor Unit:** This unit basically distributes the power for all electronic devices in the HASG.

- **Electromechanic Brake:** Electromechanic brakes are used to prevent HASG from moving while non-operating. Their working principle is simple so that, when voltage difference is applied to their pins, they releases the brake.

- **Human Interface Unit:** Human interface unit ensures the interface between the system and human. The interface has several switches to enable system, servo unit, laser unit, and to change system operating mode such as speed mode, tracking mode or park position mode.

Figure 4.11: Human Interface Unit (Photo Courtesy of ASELSAN Inc.)

The usage of Human Interface Unit is given below as an example scenario;

- User turns the Electro Optical System (EOS) rotary switch to on position to enable system.
- User turns the EOS Turret switch to on position to enable servo unit.
- User turns the EOS Turret Mode switch to "Stab On" position to make stabilization on. It must be noted that stabilization is a feature in the system that can be turned on and off.
- User turns the Camera switch to IR position to have thermal image on the monitor.
- User uses the right joystick to move turret in two axes. The image of the line of sight is seen on the monitor.
- User finds a target to track, and with left joystick , starts the video tracking. From now on, automatic video tracking starts, and it continues until target is lost due to range, or obstacle blocking the line of sight.

In the system, the communication protocol between human control unit and the mission computer in gimbal is CANBus. All the commands applied on human control unit by pressing buttons or using joystick are sent as CAN messages.

4.3 System Identification

This section develops the open loop plant model of HASG which will be used for our simulations in this thesis.

In order to obtained the open loop model of the gimbal, an input should be applied to the system and the system's response to input is observed as output. High-Accuracy Stabilized Gimbal was explained with its components in section 4.2. There, it was mentioned that servo control unit and its driver are responsible for measuring the required current to fulfill user command. So, the controller in the control unit computes the required torque, and finds the current information that should be sent to motor driver board. It is know that the current and torque relationship is such as,

$$\tau_{cmd} = i_{cmd}\frac{K_t}{\sqrt{2}} \tag{4.1}$$

so, for open loop system identification, torque input will be given to the system and speed feedback will be collected from the gyroscope. Both of these informations are collected from the test serial channel of the HASG system. In figure 4.12, open torque loop is shown and the plant that we want to identify is given.

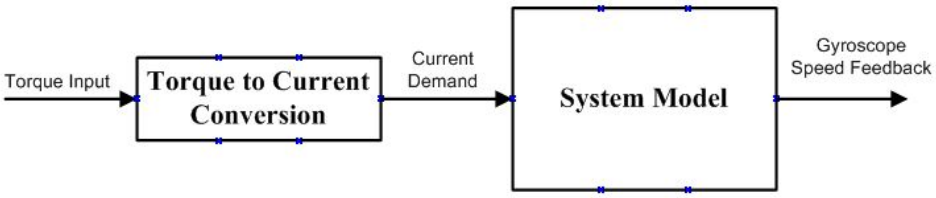

Figure 4.12: Open Torque Loop

The system identification is done in frequency domain. The input torques are applied to the system as a sine wave covering a wide range of frequencies.

In order to stay within the scope of the thesis, system identification and system modeling is done using classical integer order differential equations. All of the simulations and hardware experimental results are done for the comparison between integer and fractional order controllers.

For frequency sweep implementation on the real system, we have used the following Simulink block,

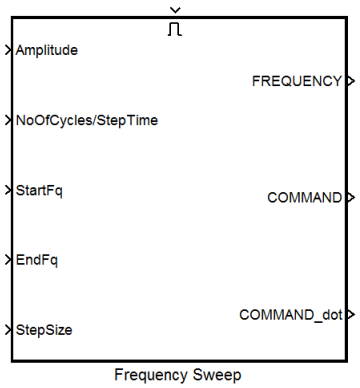

Figure 4.13: Frequency Sweep Block

where the block can be enabled by user input. When the user initiates the frequency sweep, the torque outputs of this block are fed into the system. Before frequency sweep test is enabled, the following parameters must be set,

Frequency Sweep Parameter	Value
Start Frequency	5 Hz
End Frequency	200 Hz
Frequency Step	0.25 Hz
Number of Cycle per Frequency	25
Sweep Amplitude	7.5 Nm

Table 4.1: Traverse Axis - Frequency Sweep Parameters

Frequency Sweep Parameter	Value
Start Frequency	10 Hz
End Frequency	200 Hz
Frequency Step	0.25 Hz
Number of Cycle per Frequency	25
Sweep Amplitude	2.5 Nm

Table 4.2: Elevation Axis - Frequency Sweep Parameters

Note that the frequency sweeps are applied to the traverse and elevation axis independently. In table 4.1 and 4.2, start and end frequency represents the frequencies that torque inputs will sweep. All the control algorithms run in the servo control unit are working in 1 kHz. Therefore, we consider 200 Hz as a sufficient frequency limit as it gets closer to the bandwidth of the torque loop.

In figures 4.14 and 4.15 , the torque output of frequency sweep block and the torque feedback are shown together, and it is seen that as frequency increases as the amplitude of the torque feedback decreases. The frequency step defines the increment size of the sweep that will be covered. Choosing the increment size small makes the frequency sweep more accurate as it covers wider range of frequencies. For each frequency, there are certain number of cycles of sine torque inputs applied, and this is determined by the number of cycle per frequency. For example, after 25 full cycle of the sine wave, the frequency increases with the amount of step which is 0.25 . The sweep amplitude is the last parameter that should be entered. Depending on the motor features, the sweep amplitude plays a crucial role on the result of the frequency sweep test. Sweep amplitude should be chosen as high as possible so that it excites all the nodes on the system. Thus, all the jitter caused by these nodes on the system become observable. The sweep amplitude is generally chosen as %40 of the maximum torque values of the motor. Applying high amount of torques in lower frequencies may cause the gimbal to turn with very high accelerations and velocities. In order to avoid such high accelerations and velocities, starting frequency of the sweep is choosen as 5 Hz for traverse, 10 Hz as elevation axis.
Using the parameters as defined above, the frequency sweep as torque input and feedback is drawn in the figure 4.14, 4.15,

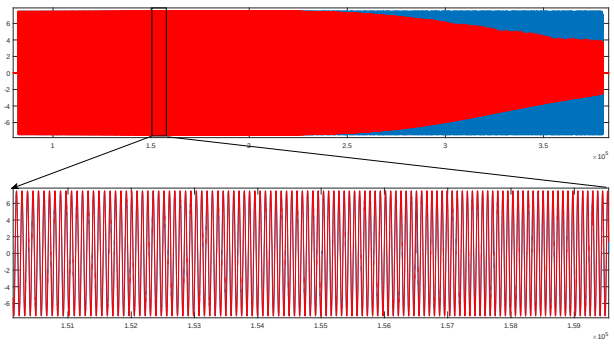

Figure 4.14: Traverse Axis Torque Command and Feedback

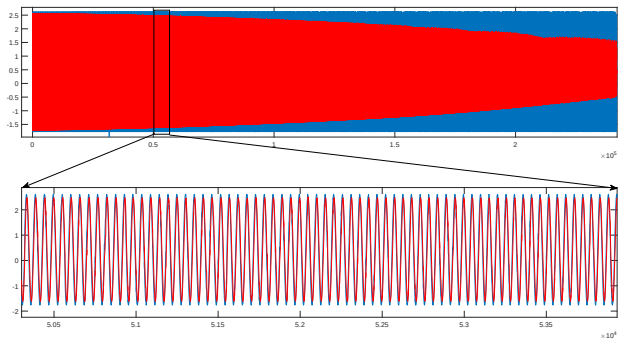

Figure 4.15: Elevation Axis Torque Command and Feedback

From figures 4.14 and 4.15, It is observed that as frequencies increase, the system becomes unable to follow the command signals. The red signal represents the torque feedback, and the blue is the torque reference. Torque feedback is obtained through current information on current sensors on both drivers, and multiplying them by $\frac{K_t}{\sqrt{2}}$

In the figure 4.16, system's gyroscope velocity feedbacks are given as response to torque inputs versus time. Note that upper plot is for traverse axis, lower plot is for elevation axis.

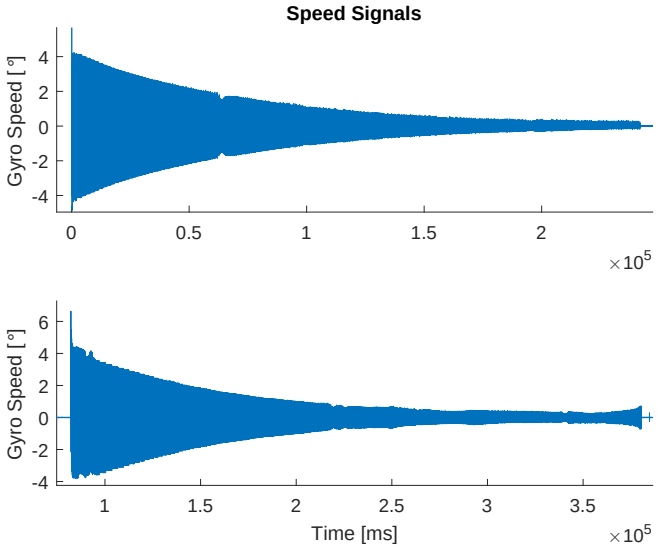

Figure 4.16: Gyroscope Speed Feedback

At this point, after the frequency sweeps are applied to the system, we have the

data including the torque inputs, current inputs, gyroscope velocity feedbacks in the frequency domain. For open loop system identification, we take torque inputs and gyroscope velocity feedbacks and obtain system open loop model. After the collected data are processed and following frequency response functions are obtained, both traverse and elevation axes given in figures 4.17 and 4.18.

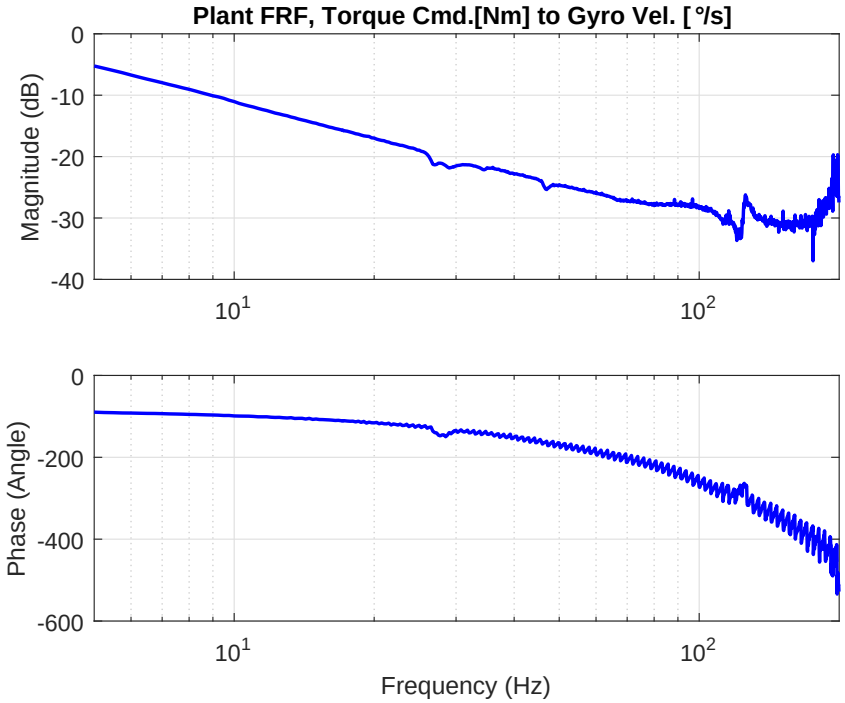

Figure 4.17: Traverse Axis Torque Command and Feedback

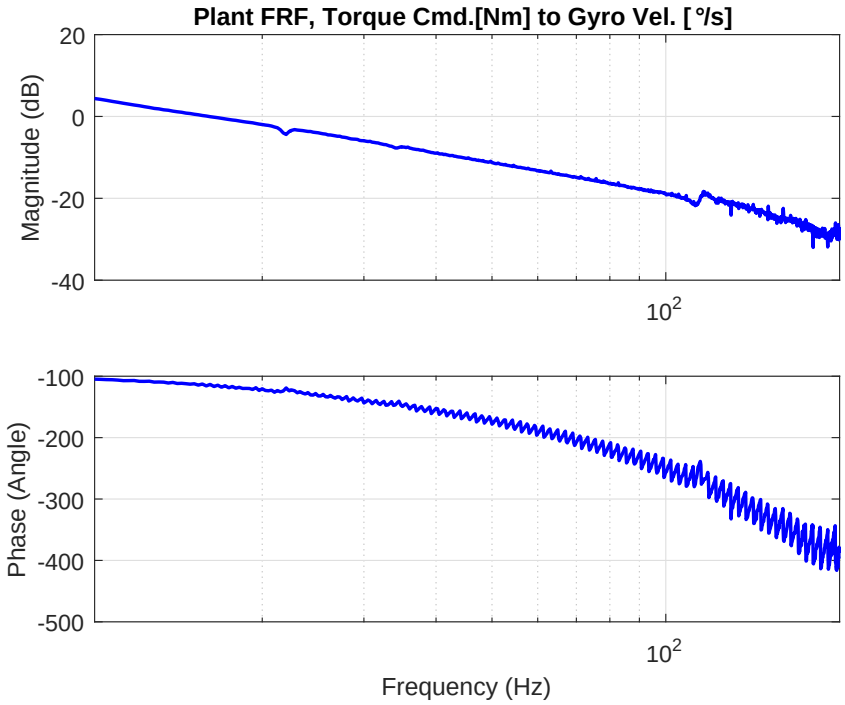

Figure 4.18: Elevation Axis Torque Command and Feedback

The frequency response functions of the plants demonstrates that in low frequencies they behave like an inertia term. However in higher frequencies the impacts of the mechanical structures, time delays and the sensor noises enhance resonance and anti-resonance peaks.

In these work [60] several system identification methods are discussed. We will use the method which is defined in the work [59]. In this work, the plant model of another stabilized gimbal of ASELSAN is derived. The obtained model and real model have an almost same responses. Therefore, the same procedure of the system identification will be adopted in my thesis. To stay in the scope of the thesis, the other identification methods are not tried here. We chose the method which was working great in similar systems.

As seen in the real system frequency response function in figure 4.17 and 4.18, there will be three dominant components of the model: The G_{plant} will have the following form:

$$G_{plant}(s) = \frac{1}{Js} e^{-st} G(s) \tag{4.2}$$

- The inertia term, which can be seen on lower frequencies as 20dB decrease

per decade that is $1/Js$.

- Resonances and anti-resonances, there will be 3-4 resonance and anti-resonance pairs in the given frequency range. These will be take place in G(s).
- Delay term, there should be certain amount of time delay in real system. So, we will add this term to G_{plant} as e^{-st}.

The G(s) is composed of four bi-quad filter where these filters represents the resonance and anti-resonance pairs. Therefore, the method is able to perfectly identify all the resonance and anti-resonance peaks. In the equation below, the form of the bi-quad filter is given:

$$G_{bi-quad}(s) = \frac{\omega_{nD}^2(s^2 + 2\xi_N\omega_{nN}s + \omega_{nN}^2)}{\omega_{nN}^2(s^2 + 2\xi_D\omega_{nD}s + \omega_{nD}^2)} \tag{4.3}$$

Choosing bi-quad filter and the applying restriction to damping ratio and natural frequencies having positive value will ensure that we will have all the poles and zeros on the left hand plane. By the Routh Criterion, we will have stable and minimum phase system having all the poles on the left half plane. The filter parameters will be found through optimization routine. Non-linear least square optimization will be applied to find all parameters for the four bi-quad filter. For this purpose, MATLAB's Optimization Toolbox and the function *lsqcurvefit* is used. The identified model has three main part; the inertia term, delay term and the cascaded bi-quad filter which represents the resonance and the anti-resonances. The first step in the identification process that has to be determining cascaded bi-quad filters from the collected data. The collected data is the frequency sweep data of real system. We want our model to fit real plant with all three components explained above. In order to do this, we first need to fit G(s) with resonance and anti-resonance frequencies. Doing system identification in frequency domain requires to remove inertia term.
In order to remove inertia term on the plant FRF, we first need to measure it from the collected data. Here is the procedure;

- First, we will select a point from the Bode plot where it behaves like $\frac{1}{Js}$. Lets select 10Hz.

- We know that $|H(jw)|_{dB}$ will give us magnitude of transfer function in dB where $H(s) = \frac{1}{Js}$. We pick a point from the bode plot of the real ssytem, where it behaves exactly like $\frac{1}{Js}$. At 10 Hz,

$$20log(H(jw)) = -11.04 \tag{4.4}$$

in dB where f=20Hz, and $\omega = 2\pi f$.

$$20log(\frac{1}{J}\frac{1}{j2\pi f}) = -11.04 \tag{4.5}$$

so, from the equation above, J is found as 0.0567 kg/m^2 for traverse axis. For elevation axis, same calculation made at 20Hz, and it is measured as 0.0098 kg/m^2. In figure 4.19, these points are shown,

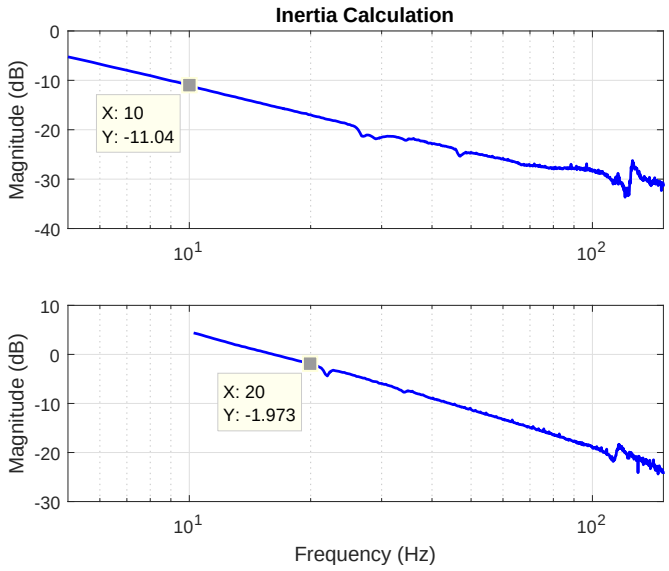

Figure 4.19: Inertia Calculation on Bode Plot

The frequency sweep for elevation axis was starting at 10Hz, in order to avoid contact with mechanical limits. We are performing these test in open loop, so we don't have control on these limits, end damping blocks only work in closed speed loops and outer loops.

- After inertia has found, in order to remove inertia from the collected plot, we multiply the plant with the inertia term.

$$G_{plantW/oInertia} = G_{plant}Js = e^{-st}G(s) \tag{4.6}$$

In the figure 4.20 and 4.21 the inertia removed frequency response functions of the system are shown.

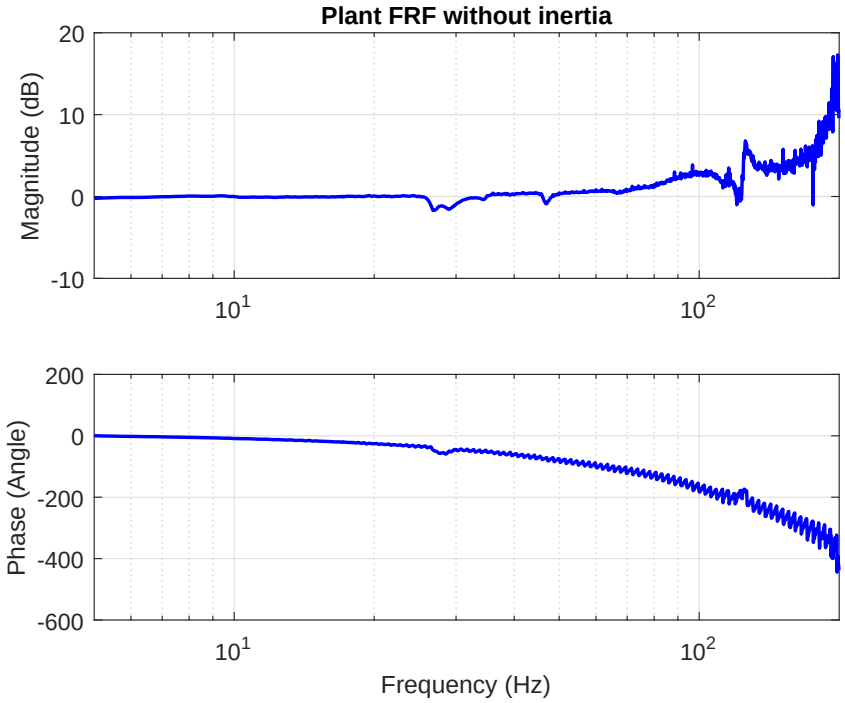

Figure 4.20: Traverse Axis Bode without Inertia

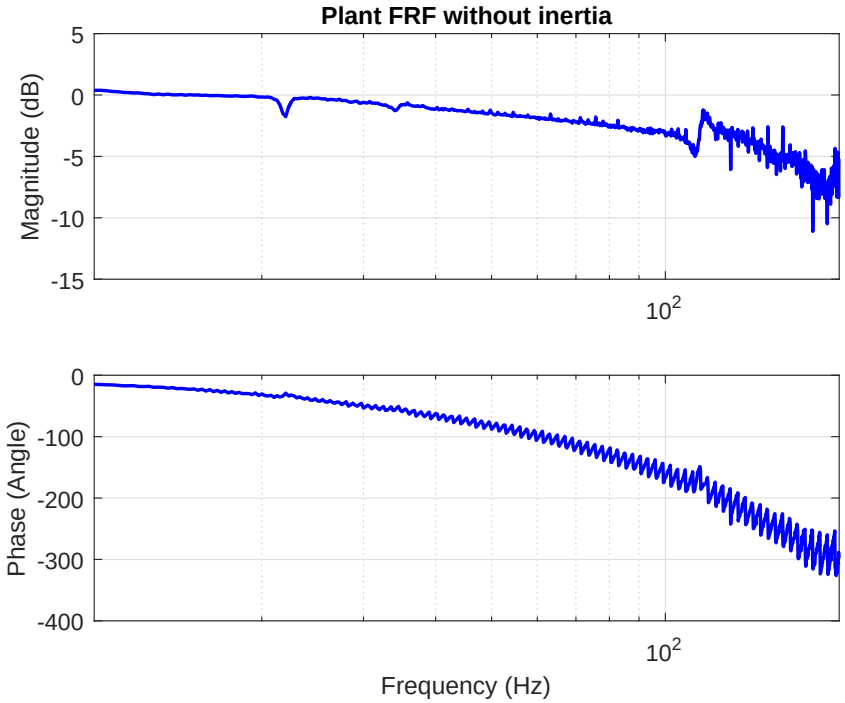

Figure 4.21: Elevation Axis Bode without Inertia

At this stage of system identification procedure, we will find the G(s) and will be ensure that our magnitude plot is same as in the plots 4.20 and 4.21. The optimization routine needs initial points and some constraints. As we mentioned before, we have added constraints on damping ratio and frequency positive. We also added another constraint that the maximum value for the resonance and antiresonance frequency is 200 Hz which was our upper bound for the frequency sweep. For four bi-quad filters each having four parameters, in the optimization routine, we will obtain 16 parameters.
In table 4.3, these 16 parameters are given with their initial starting point with their range where optimization will be done.

Opt.Par	L.B.	U.B.	Tra.Axis	Ele.Axis
f_{1N}	0	200	26	22
ξ_{1N}	0	1	0.5	0.5
f_{1D}	0	200	32	24
ξ_{1D}	0	1	0.5	0.5
f_{2N}	0	200	112	34
ξ_{2N}	0	1	0.5	0.5
f_{2D}	0	200	115	36
ξ_{2D}	0	1	0.5	0.5
f_{3N}	0	200	122	112
ξ_{3N}	0	1	0.5	0.5
f_{3D}	0	200	125	116
ξ_{3D}	0	1	0.5	0.5
f_{4N}	0	200	160	190
ξ_{4N}	0	1	0.5	0.5
f_{4D}	0	200	195	196
ξ_{4D}	0	1	0.5	0.5

Table 4.3: Optimization constraints and initial values for both axis

In figure 4.22 and 4.23, the resonance and anti-resonance frequencies are shown. These frequencies are taken as initial point in these optimization procedures.

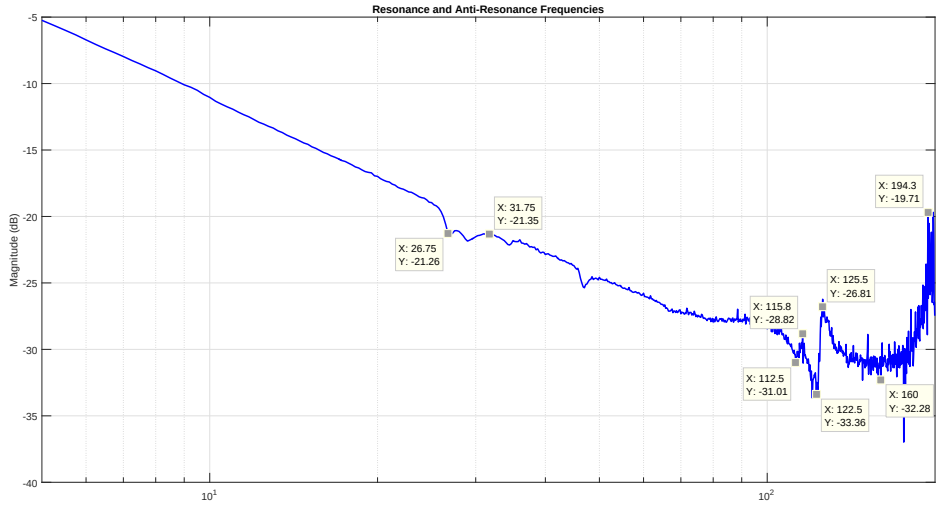

Figure 4.22: Traverse Axis Resonance and Anti-Resonance Frequencies

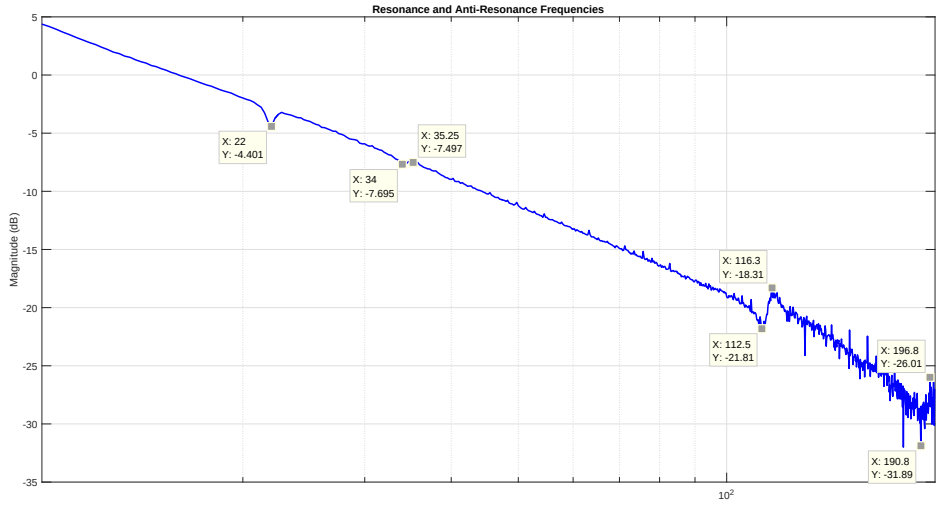

Figure 4.23: Elevation Axis Resonance and Anti-Resonance Frequencies

Using MATLAB's function *lsqcurvefit*, cascaded bi-quad filter is obtained which is the only G(s) part of the identified model. The curve fit is applied to only the magnitude response of the system, so that the minimum phase assumption is done. In the figure 4.24, the poles and zeros of the biquad filter G(s) can be seen,

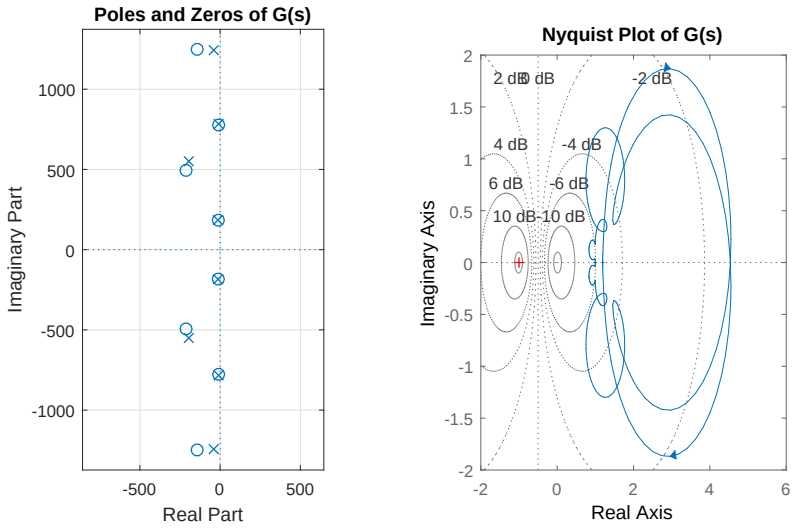

Figure 4.24: Traverse Axis, Poles and Zeros of the Identified G(s)

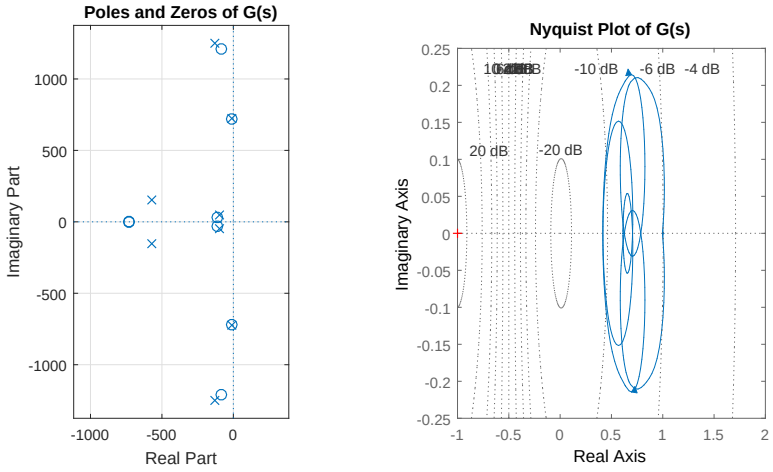

Figure 4.25: Elevation Axis, Poles and Zeros of the Identified G(s)

As seen in figures 4.24 and 4.25, all the poles and zeros are on the left half plane. In the Nyquist plot, we observe that there no encirclement of zero in clockwise direction around the contour, and we also know that there no unstable poles and zeros, we can conclude that system is stable and minimum phase assumption is achieved.

In the figure 4.26 and 4.27, it is observed that the magnitude responses are very closer to each other. However, the phase responses are very different, because the fitting process is made for only the magnitude response. The reason for the

difference is the certain amount of time delay in the real system, and the delay will be found next.

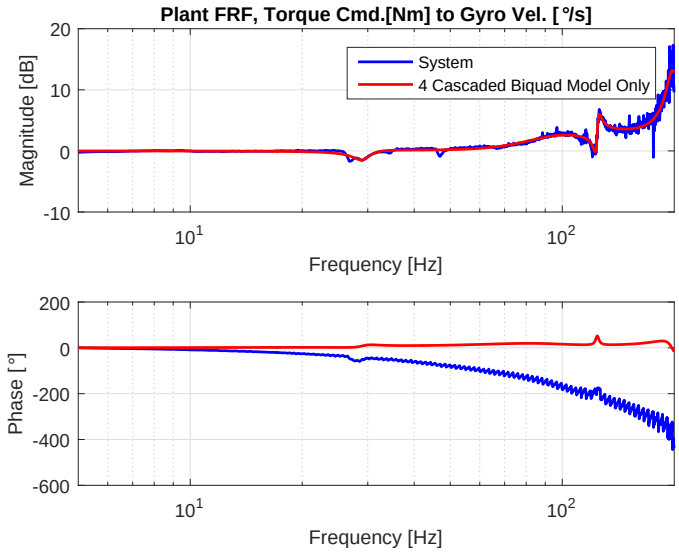

Figure 4.26: Traverse Axis Bode without Inertia with Delay

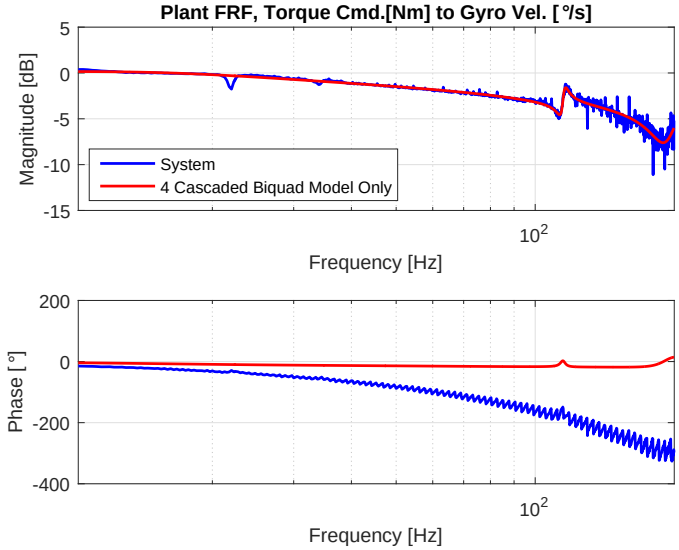

Figure 4.27: Traverse Axis Bode without Inertia with Delay

In this model, ignoring the delay term completely changes the phase of the bode plot. The collected real system data and the plant have a time delay which must be found. The next process in the system identification is to find this time delay. Once the G(s) is fixed with resonance and anti-resonance frequencies, and

structured, the e^{-st} is added to the model.

This time, curve fit is run for phase component of the frequency response function in order to find amount of delay. The output of the this process is illustrated in figures 4.28 and 4.29 . In these figures, the delay component of the response is affected by adding time delay, the difference in the phase plots is now diminished and these plots shows similar responses.

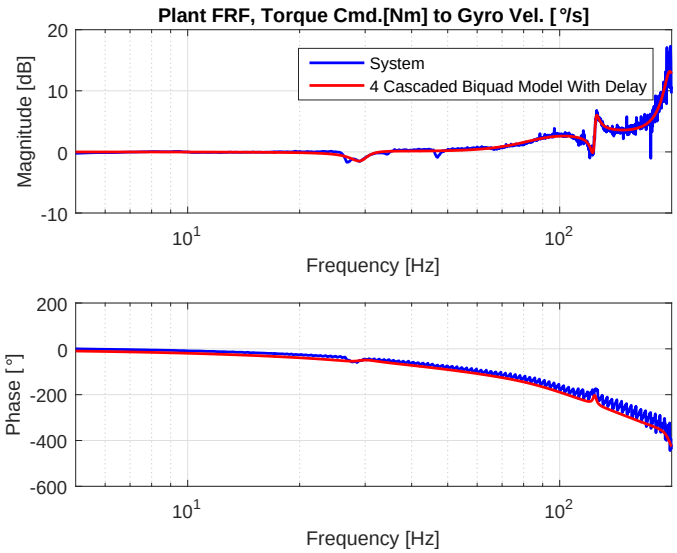

Figure 4.28: Traverse Axis Bode without Inertia with Delay

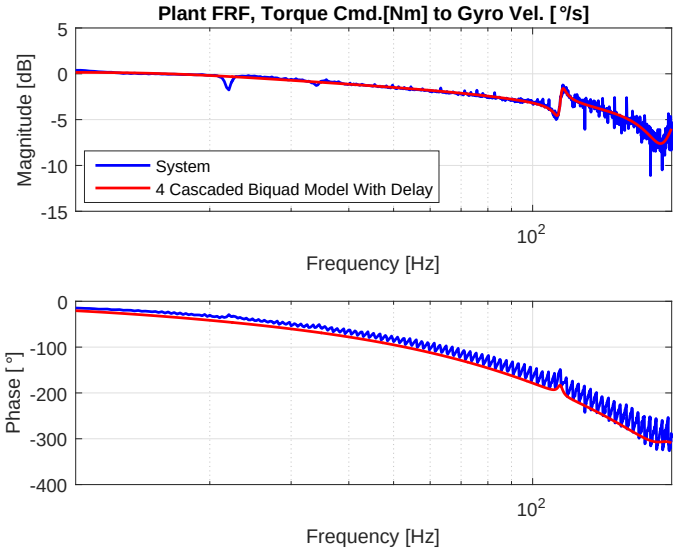

Figure 4.29: Elevation Axis Bode without Inertia with Delay

Through the fitting process, the delay is found as 6.402 ms for traverse axis and 4.4986 ms for elevation axis. This is the amount of the time that the system waits for the response to the command that is coming from the user. In order to illustrate the time delay, the following step response data taken from the system and the figures are shown,

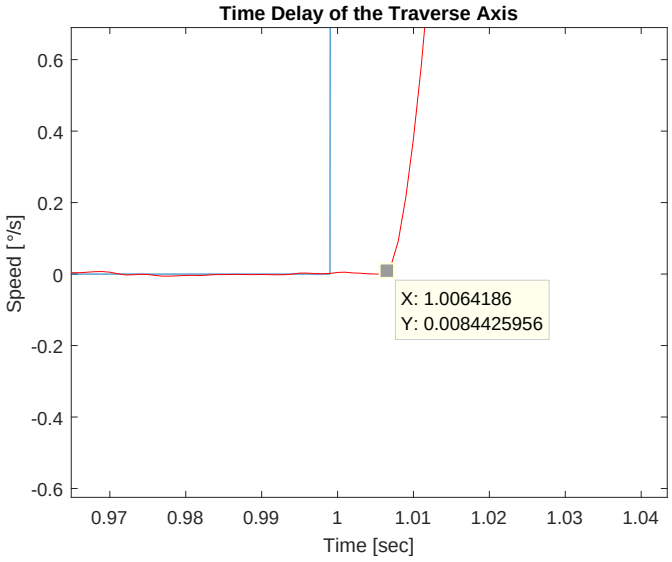

Figure 4.30: Time Delay in Traverse Axis

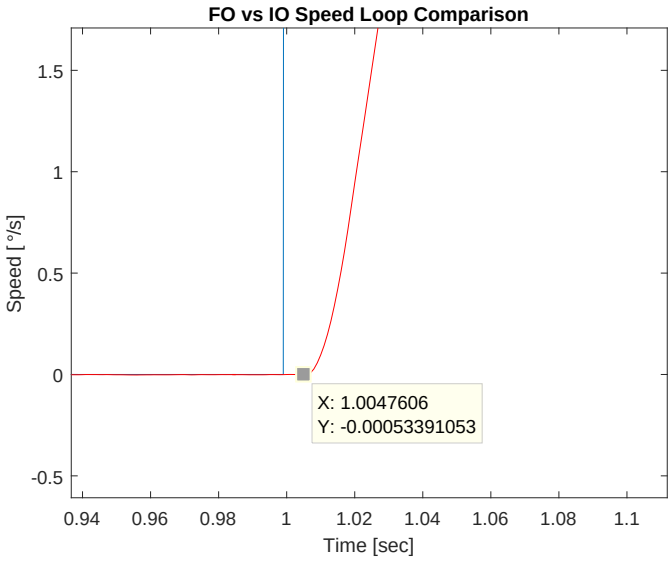

Figure 4.31: Time Delay in Elevation Axis

In figures 4.30 and 4.31, the time delays are shown approximately, in the identified model, the result of the curve fit process for the time delay will be used. Note that, the delay in the real system and the ones that we obtain by optimization are very close to each other. Therefore, we verify that we have estimated the delay correctly and the identified system's bode plots gets closer to real system. Until this point, the first two part of the transfer function of the model are constructed and the responses are shown. Next, the inertia term will be added to this model. After adding 1/Js to the model, the identification is completed and the identified model shows the exactly same response with the real system.

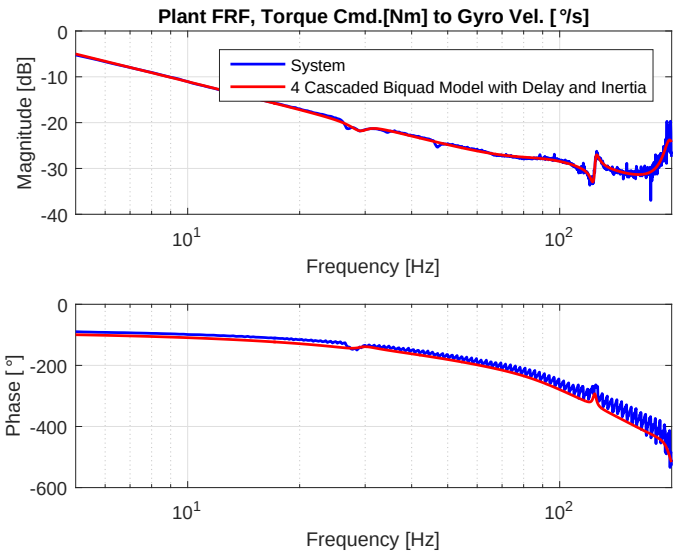

Figure 4.32: Traverse Axis Bode with Delay and Inertia

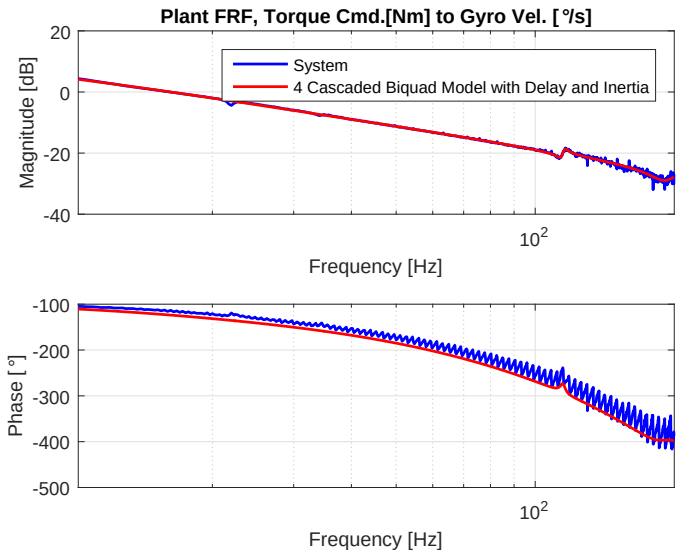

Figure 4.33: Elevation Axis Bode with Delay and Inertia

In the frequency response function of the real plant, it is observed that the phase part of the response has increasing wave magnitudes in saw-tooth shape with higher frequencies. When this behavior in the phase plots were obtained first, we have reapplied frequency sweep tests and in all of the tests similar results are obtained. After trying several changes in the system, we have obtained the result that when we use different kind of gyroscope, the shape of the phase plot changes. Then, it is found that the reason for behavior is due to the asynchronous communication between gyroscope and the servo control unit which reads the gyroscope data. This communication is held on RS422 protocol with 921600 kbps baudrate. The gyroscope is supposed to send data in 1kHz, however, the gyroscope is not able to send data with required sampling rate. We have performed several tests with changing the sampling rate of the gyroscope and similar results are observed, and the amount of the error that sampling rate changes affect the frequency of the peaks. The more detailed analysis of this problem and the illustration of the aforementioned tests and the results are given in the next section.

4.3.1 Asynchronous Communication with Gyroscope and its Effect on System Identification

4.3.1.1 Problem Definition

We have faced with the problem of saw-tooth like triangular jumps on phase plot of the bode of the identified system. The open loop system identification in frequency domain directly uses the gyroscope's speed feedbacks for given sinusoidal torque inputs. This procedure has repeated many times to be sure that obtained bode plot has no mistake. All the results for frequency domain system identification tests were same, all of them has same pattern on the phase component.

In figure 4.34, the bode plot of the system is given. In this figure, the response at the higher frequencies are focused to demonstrate the pattern on the phase plot. As frequencies gets higher, the amplitude of these frequencies gets bigger.

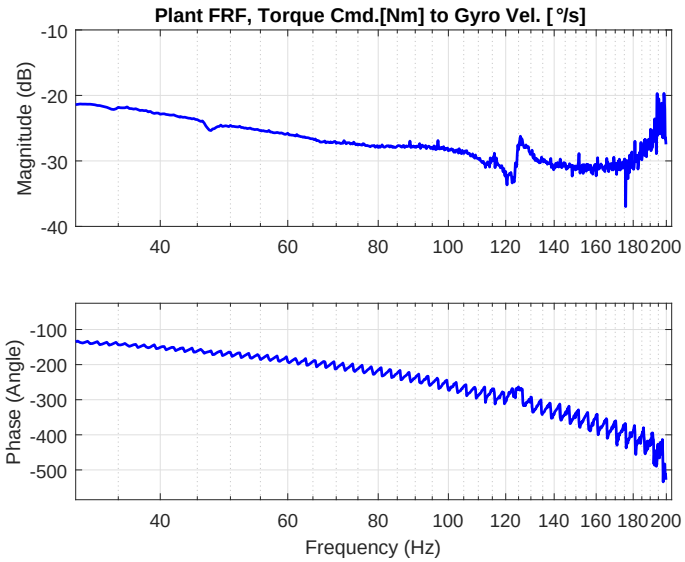

Figure 4.34: Traverse Axis Torque Command and Feedback

In order to specify the problem, many frequency sweeps are applied at some other systems, with same and different gyroscopes. It is found that the similar patterns occurs every time with different shapes.

Theoretically, the servo controller module expects gyroscope data from serial

RS422 channel at baud rate 921600 and with sampling 1KHz. When we first obtained that the shape of pattern changes with using different gyroscope, we have considered that the problem in the serial communication might cause these results.

In order to simulate this the asynchronous communication between servo controller unit and the gyroscope, we have build a MATLAB Simulink model. This model basically creates the frequency sweep torque signals at sampling rate 1μ sec, and applies rate transition to send these commands to the current controller at 1 KHz. The simulink model is given in figure 4.35,

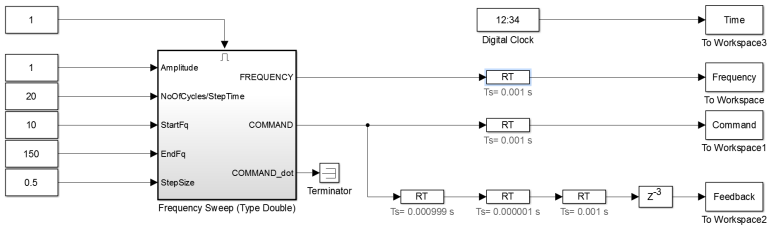

Figure 4.35: Simulink Model to simulate effect of asynchronous communication

From the simulink model, it is observable that, the whole model runs with sampling rate $1e^{-6}$sec. The outputs are saved to workspace with sampling rate $1e^{-3}$sec, in order to make computations easier. The asynchronous sampling is applied to the command signal and it is saved as feedback.

4.3.1.2 Results

The bode plots are drawn for these feedback and command signals in order to simulate asynchronous communication between gyroscope and servo controller.

- Bode plot for sampling rate $T_s = 0.001$

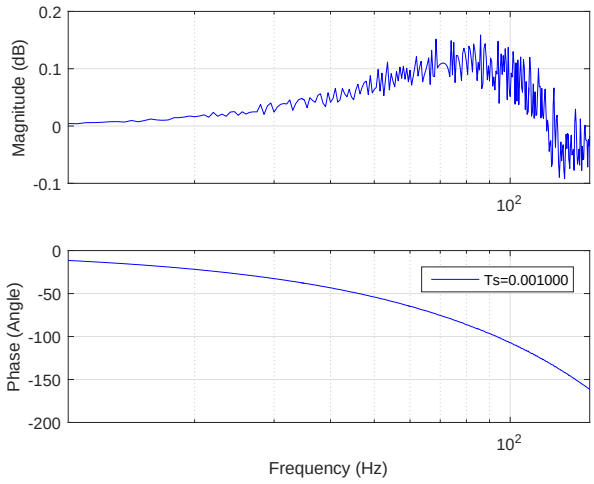

Figure 4.36: The Bode Diagram with Gyroscope running in 0.001 sec sampling

It is observable that the triangular patterns are not observable here. Because, the sampling rate is determined for ideal case as 1Khz.

- Bode plot for sampling rate $T_s = 0.000990$

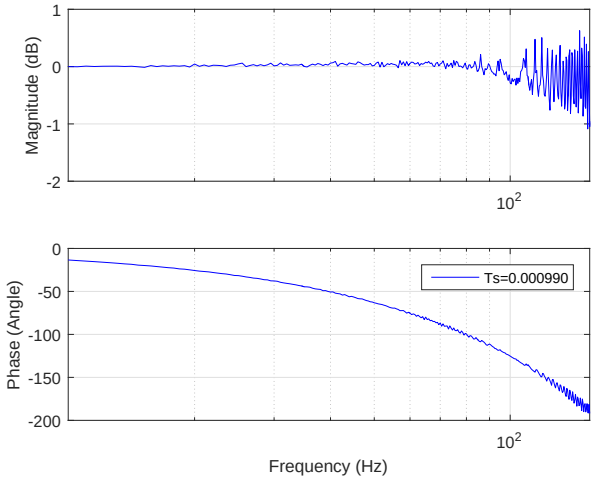

Figure 4.37: The Bode Diagram with Gyroscope running in 0.00099 sec sampling

In figure 4.37, the triangular pattern became to be observable at higher frequencies.

- Bode plot for sampling rate $T_s = 0.000998$

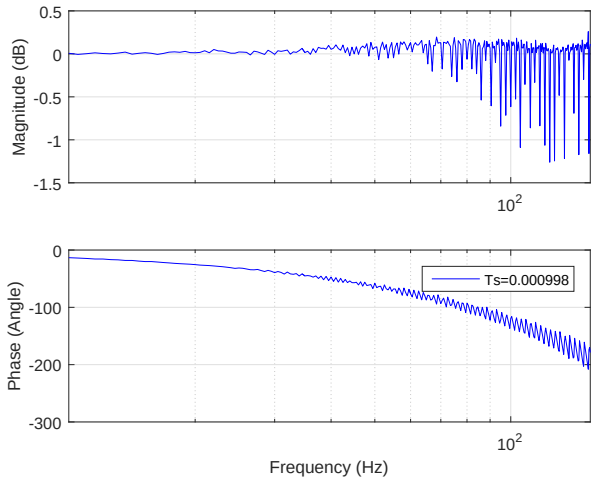

Figure 4.38: The Bode Diagram with Gyroscope running in 0.00998 sec sampling

As sampling rate gets closer to 0.001, the patterns became more observable as in figure 4.38.

- Bode plot for sampling rate $T_s = 0.000999$

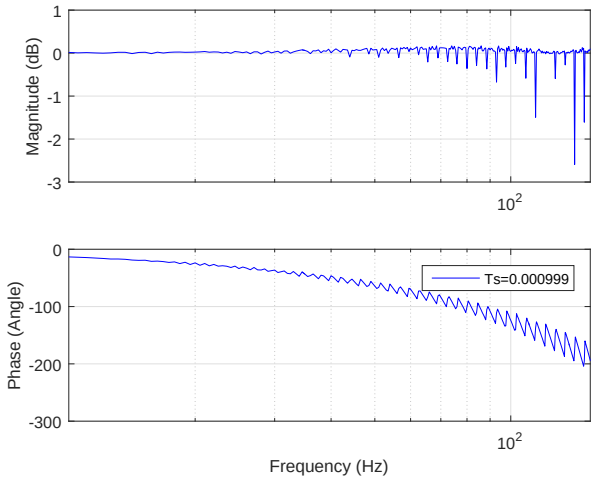

Figure 4.39: The Bode Diagram with Gyroscope running in 0.00999 sec sampling

In figure 4.39, triangular patterns are clearly illustrated, this result is very similar to the one represented in figure 4.34.

- Bode plot for sampling rate $T_s = 0.001001$

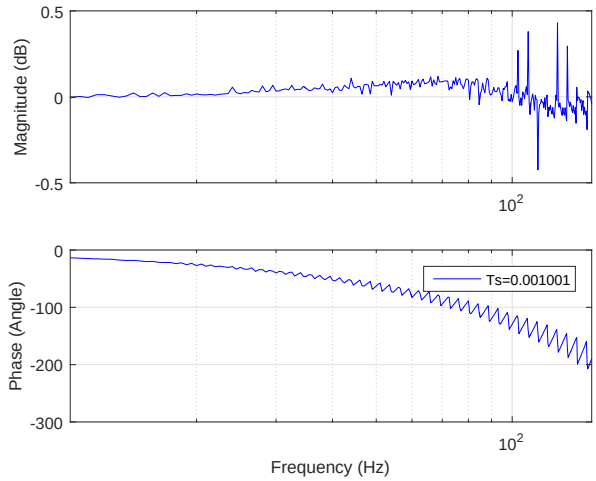

Figure 4.40: The Bode Diagram with Gyroscope running in 0.001001 sec sampling

In figure 4.40, sampling of gyroscope becomes slower than the system, and the direction of the triangular shapes changes. Compared to figure 4.39, figure 4.40 is more closer to the real system bode illustrated on figure 4.34.

- Bode plot for sampling rate $T_s = 0.001002$

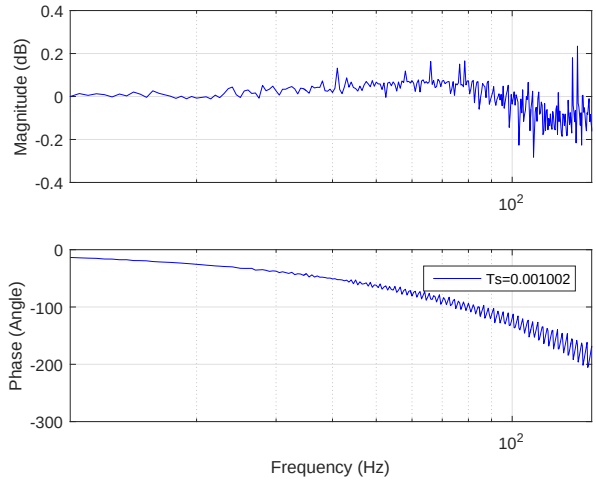

Figure 4.41: The Bode Diagram with Gyroscope running in 0.001002 sec sampling

- Bode plot for sampling rate $T_s = 0.001010$

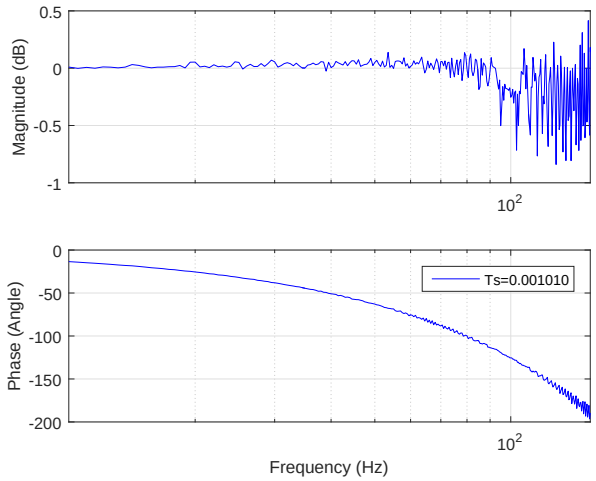

Figure 4.42: The Bode Diagram with Gyroscope running in 0.00101 sec sampling

From the figures, it is obtained that, whenever the sampling rate of the signal differs from the expected rate and it is collected at this rate, the shape of the bode plot starts to deteriorate. When the difference is small, the frequency of the triangular pattern on the phase plot decreases. The analysis showed that when this difference goes to zero, these patterns will not occur. It must also be noted that the direction of these triangular patterns changes with the sign of the difference in sampling rates. When sampling is faster than the ideal, the direction of these triangles are downward. In this case, after downward phase drop, there occurs phase jump in positive way. When sampling is slower than the ideal case, the direction becomes upward. In this case, the negative drops occurs at phase plot.

As it is explained in the system identification section 4.3, we have chosen the worst case time delay parameter to fit phase plot of the model with the real system. In both axis, the response of the axes became more similar to the real system by choosing higher time delay for the plant.

4.4 Verification of System Identification

In this section, the obtained open loop plant model is verified. The verification is done using the model of the plant where the step response of this model is compared with the step response of the real system.

In order to compare the real system and the identified plant model, firstly, we construct a simulink model. In this model, we have added all the components which are currently exist in the DSP software of the real system. The general view of the model is shown in figures 4.43 and 4.44.

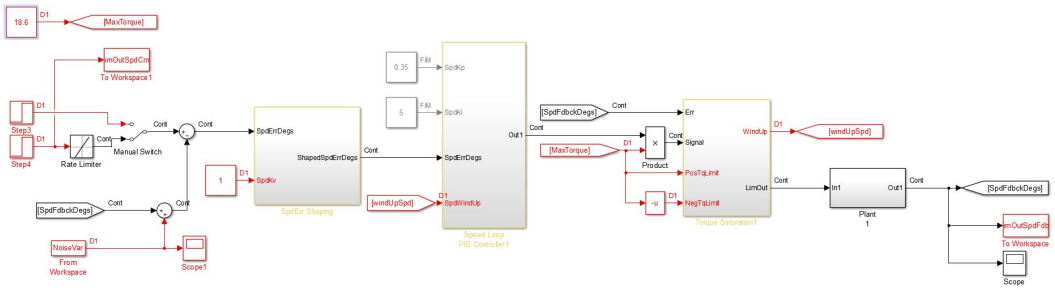

Figure 4.43: Simulink Model for Tests for Traverse Axis

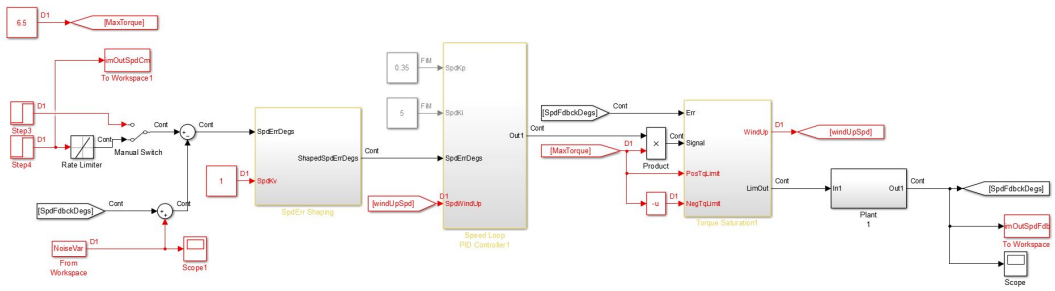

Figure 4.44: Simulink Model for Tests for Elevation Axis

The only difference for traverse and elevation axis models will be the constraints regarding motors used as actuator. These figures are the implementation of the block diagrams represented in figure 5.2 section 5.1.

The model is basically composed of cascaded three loops which are torque, speed and position loops. The nonlinear components which exist in the current torque and speed loops are added to the simulink model. The physical restrictions of

the real system and implementation methods are the main reasons for these nonlinear components. The physical restrictions occur due to the torque limits for the motor, the mechanical properties, and safety concerns. There is a saturation block in the torque loop, which basically limits the maximum torque output. The saturation block has an windup output, which discards the integral output of the controller by switching integral input to zero when torque saturation occurs.
The open loop plant is modeled as,

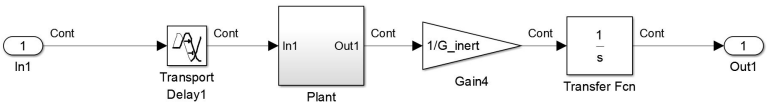

Figure 4.45: Plant model on Simulink

In figure 4.45, the plant subsystem refers to the cascaded bi-quad filters as mentioned in previous section that we use four cascaded bi-quad filters to simulate resonance and anti-resonance peaks. Transport delay is added to model with the measured time delay as e^{-st}. G_{inert} is the inertia of the system which was calculated in previous section and with $\frac{1}{s}$, they represents the inertia term on the identified model. Lastly, the G(s) is modeled as,

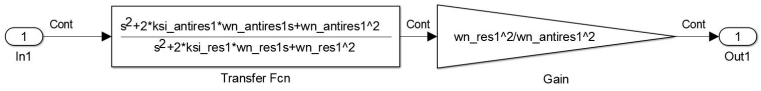

Figure 4.46: Bi-quad filters on Simulink

having the parameters found from the optimization routine.

Step responses of the real system and the identified model is compared in the figure 4.47 and 4.48.

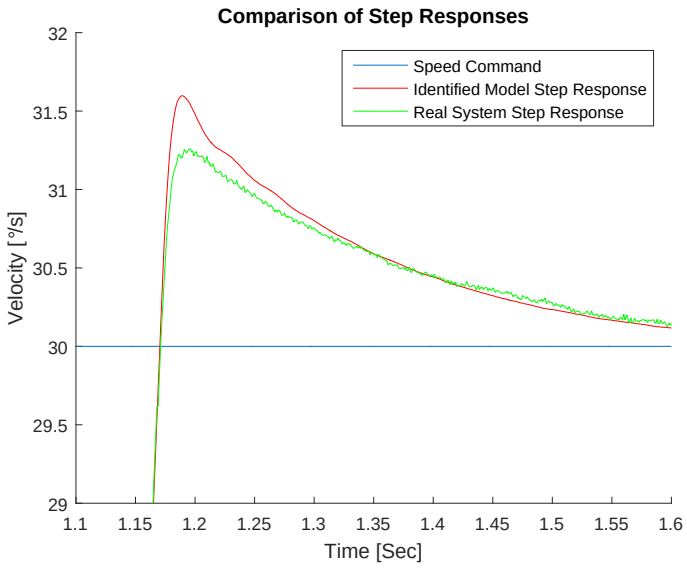

Figure 4.47: Traverse Axis Step Response Comparisons

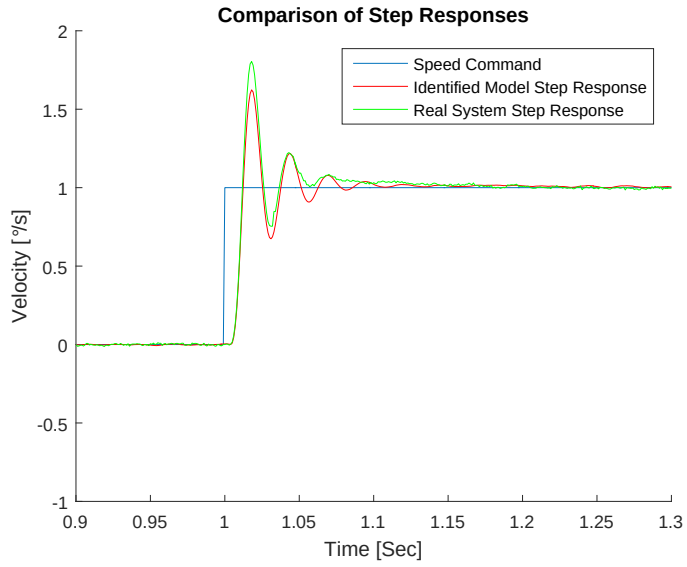

Figure 4.48: Elevation Axis Step Response Comparisons

When the responses of the identified model and the real system are compared, there are similarities in time delay, rising time, the acceleration, the maximum overshoot and the settling time. The differences in terms of the comparison parameters are negligibly small. In figures 4.49 and 4.50, the error plots of the step responses are given,

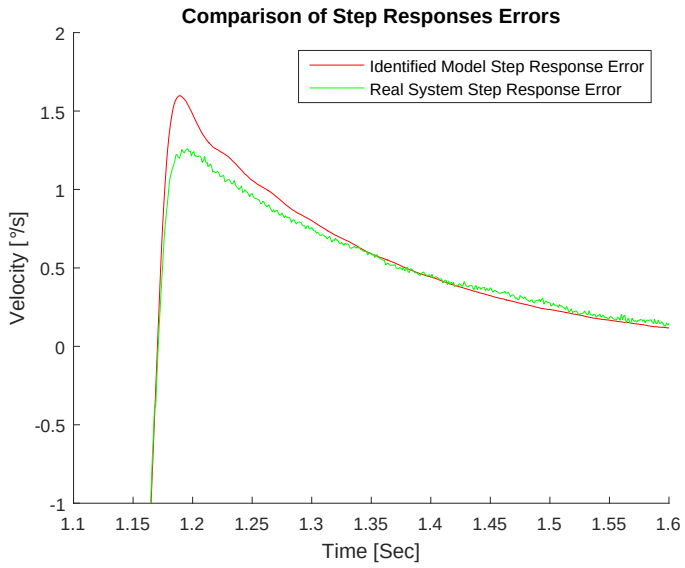

Figure 4.49: Traverse Axis Step Response Error Comparisons

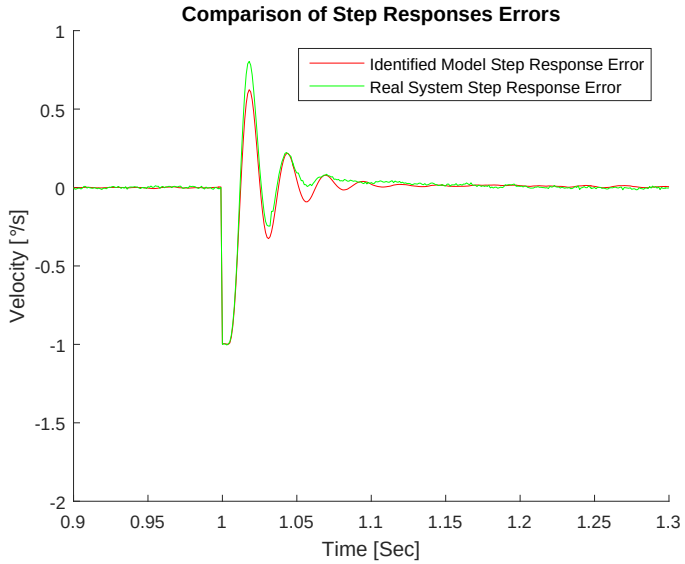

Figure 4.50: Elevation Axis Step Response Error Comparisons

It can be stated that, model and the system's step responses are very similar, and the identified model can simulate the real system. It is critical to use a validated model for the simulations of the plant where fractional order controllers are applied. The validation errors directly affected by the identified model to be controlled by fractional order controller. The aim of the detailed system identification studies was to create a place where we can perform controlled

studies on fractional order controllers. In the following chapter, we will study fractional order controllers, the effects of the controller parameters on response of the system which is identified in this chapter.

CHAPTER 5

FRACTIONAL ORDER CONTROLLER DESIGN AND IMPLEMENTATION

In this chapter, we develop the control architecture that will be applied to the simulation tests introduced in Chapter 6 and to the hardware tests introduced in Chapter 7.

The aim of the control architecture developed in this chapter is to replace integer order controllers with the fractional order controllers with enhancing the performance criteria of the system compared to the ones controlled by integer order controller. In Chapter 2, it is explained that the fractional order PID controller outperforms the integer order controller. The aim of the this chapter is to perform several tests for fractional controllers and justify the selected accurate design procedure for the system and demonstrate the level of outperformance over classical controllers.

Fractional order PID controller design and implementation on MATLAB is introduced in this chapter. We will initiate this chapter by the approximation methodologies we adopted in our design . These are the Oustaloup's Recursive Approximation, Direct Tustin Discretization and the general CFE method. These three methodologies are the most commonly used methods in the literature. There are many reasons for these methodologies to become most commonly used among other methods. These methods are easily implementable in MATLAB, and are used in existing Fractional Order Controller Toolboxes created by the community. The most used and commonly admitted toolboex are CRONE [23] by Oustaloup et. al, FOTF by Chen and Petras, FOMCON [25] by Tepljakov. Therefore, we will implement all of these three methods and compare

them. Evaluating the results with the performances of the controllers will lead us to select the most accurate methodology for our system.
The approximated and then realized controller will be studied after appropriate tuning in order to evaluate the fractional order controller used for the High-Accuracy Stabilized Gimbal. The tuning nodes of the fractional controller are first discussed in section 5.4.1 in order to illustrate the effects of these parameters. These parameters and their effects on controller responses will be given in plots.
In order to compare integer and fractional order controllers, the tuning and design procedures are considered with some criteria which will let us conduct a suitable comparison controllers without any bias. While comparing these controllers, their step responses with identified system plant model, their disturbance rejection performance and reference tracking performances will be considered. Before making any comparison, we will introduce the optimization procedures in section 5.4.2, so that controllers will be tuned optimally. Therefore, the comparison of the integer and the fractional order controllers will be done under optimized performance indexes. These optimization procedures are held for both elevation and traverse axis using the identified model in previous chapter.

5.1 Control Structure in Hardware

Control structure mainly consists of three loops which are position, speed and torque loops respectively. In figures 5.1, 5.2 and 5.3 the overview of these control loops are given. There are some additional elements used in these loops such as torque limiter, speed limiter, acceleration limiter. Torque limiter limits the torque output of the motor depending on its capacity. Similarly, the speed and the acceleration properties of the system are also limited.
The torque loop is actually a current loop in controlling DC motors. DC motors, torque is directly proportional to currents flowing in the windings of motor. In our system, we have the current sensors used as feedback for torque loop. The relation between torque and current is torque equals $\sqrt{2}/K_t$ times current. Torque

demand is changing according to current demand with this relationship and send to the motor driver as in figure 5.1.

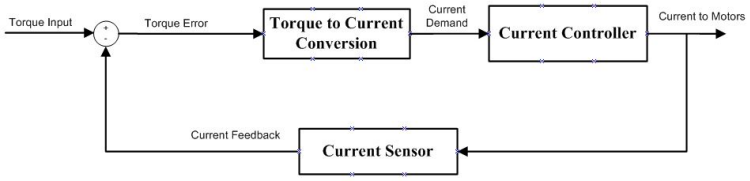

Figure 5.1: Closed Torque Loop Overview

Speed loop takes place is the outer loop of the torque loop as seen in figure 5.2. Speed loop performs the line of sight velocity control with PID Controller. The velocity feedbacks are obtained from the gyroscope.

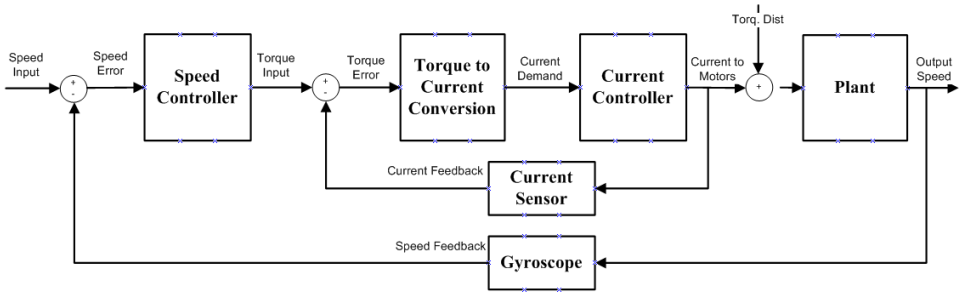

Figure 5.2: Closed Speed Loop Overview

For target tracking performances and position demands, there exists a position loop above the speed loop shown in figure 5.3. The position inputs are the references coming by system control unit, as position demands. Depending on the demand type such as stabilized position reference or non-stabilized position reference, the position sensor switches between gyroscopes integral and encoder respectively.

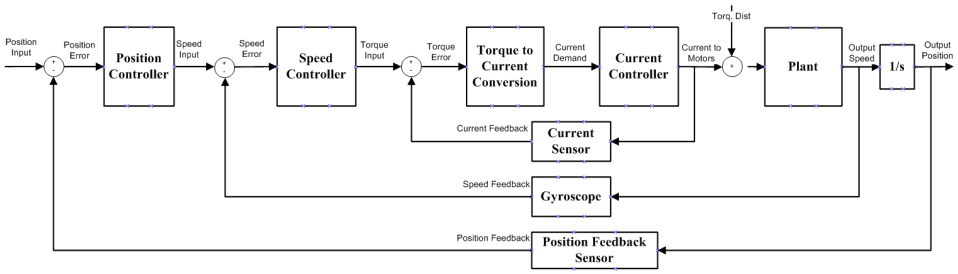

Figure 5.3: Closed Position Loop Overview

The position of target and the speed of the target are provided to the controller by the video tracker unit. In video tracking tasks, the position loop takes the bore-sight error as input error, and target speed is added to the PID output of the controller as feedforward speed. The position loop for the video tracking performance is given in the figure 5.4.

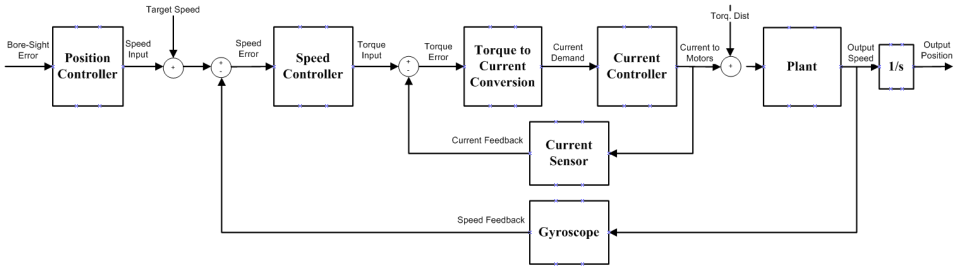

Figure 5.4: Closed Position Loop Overview while Video Tracking

5.2 Controller Realization

In this section, the results of the three selected approximation methods are compared in terms of performance. These methods are,

- Oustaloup's Recursive Approximation
- General Continued Fraction Expansion
- Recursive Tustin Discretization

The aim of the comparison is to obtain the best one, and this controller will be realizable for hardware implementation purposes. The criteria to find the

best controller is the ability of being realizable and the performance outcomes. We run the comparison after all of the designed controllers are optimally tuned. These results will be discussed in Section 5.4.2.3

The fractional order controller approximation methods are implemented on MATLAB. In order to validate these implementations, we have chosen to implement fractional integrator of order $\lambda = -1$ regarding the feature of the integer equivalence of the fractional order controller. We know that, the approximated controllers must give the same response with standard integral term s^{-1}. So, the first order integral term as the designed controller s^{λ} with $\lambda = -1$ is obtained in the 3 form of approximations,

- Oustaloup's Recursive Approximation

$$G_{C_{ORA}} = \frac{0.001s^5 + 1.067s^4 + 67.61s^3 + 269.2s^2 + 67.34s + 1}{s^5 + 67.34s^4 + 269.2s^3 + 67.61s^2 + 1.067s + 0.001} \tag{5.1}$$

  its discretized, approximated version is,

$$G_{c_d} = \frac{0.0015z^5 - 0.006402z^4 + 0.01067z^3 - 0.008608z^2 + 0.003304z - 0.0004674}{z^5 - 4.935z^4 + 9.739z^3 - 9.608z^2 + 4.739z - 0.9349} \tag{5.2}$$

- General Continued Fraction Expansion

$$G_{C_{CFE}} = \frac{0.001}{z^5 - z^4} \tag{5.3}$$

- Recursive Tustin Discretization

$$G_{C_{Tustin}} = \frac{0.0005z^5 + 0.0005z^4 + 0.0002z^3 + 0.0002z^2 + 0.0001z^1 + 0.0001}{z^5 - z^4 + 0.4z^3 - 0.4z^2 + 0.2z^1 - 0.2} \tag{5.4}$$

In figure 5.5, the bode plot of these controllers are given for comparison. It is seen that,for a certain range of frequencies, their responses are similar. However, some of the methods takes lower and higher frequency values as input to their process to limits their working range. The lower bound for the working range is chosen as 0.001 Hz, and the upper bound is selected as 1000 Hz. When bode plots are compared, the results obtained that closer to these ranges, their responses become different. The differences in the responses will play a crucial role for deciding which realization method to choose for both simulation tests

and real system tests.

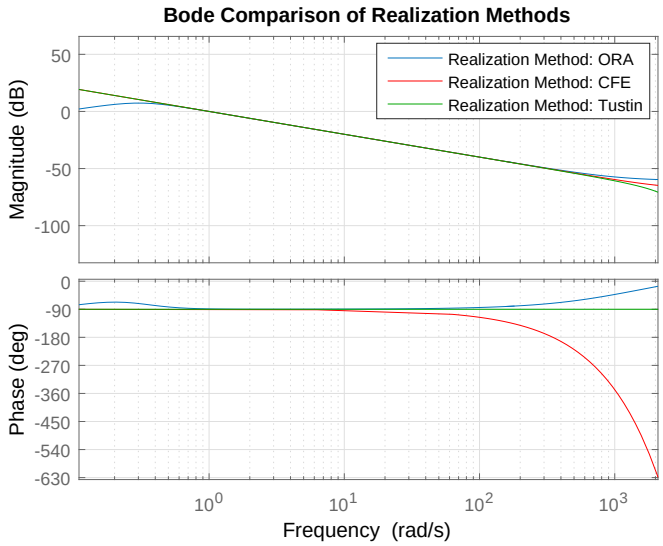

Figure 5.5: Bode plots for controllers approximated by the 3 different methods

In the figures 5.6, 5.7 and 5.8 Bode plots of these functions are compared with first order integral. The error plots are given in both magnitude and phase domain.

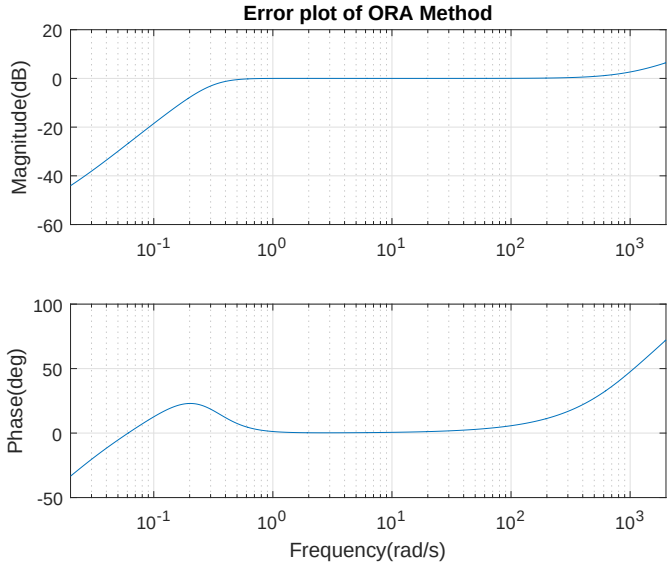

Figure 5.6: Error of ORA Method compared to first order integral

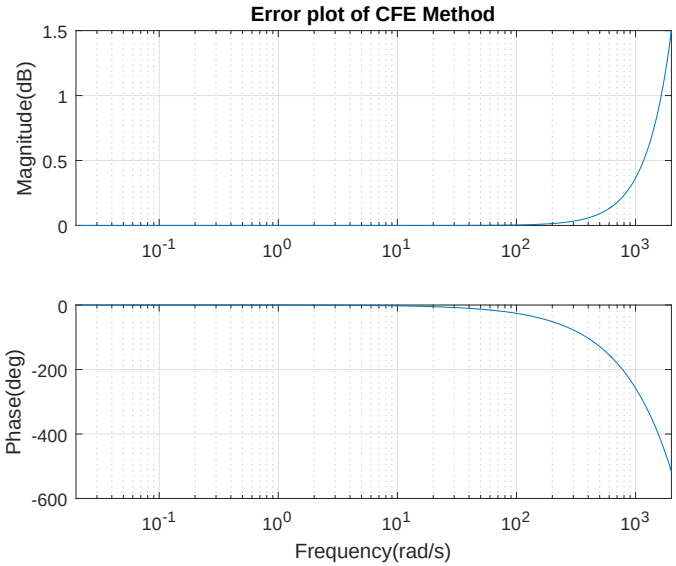

Figure 5.7: Error of CFE Method compared to first order integral

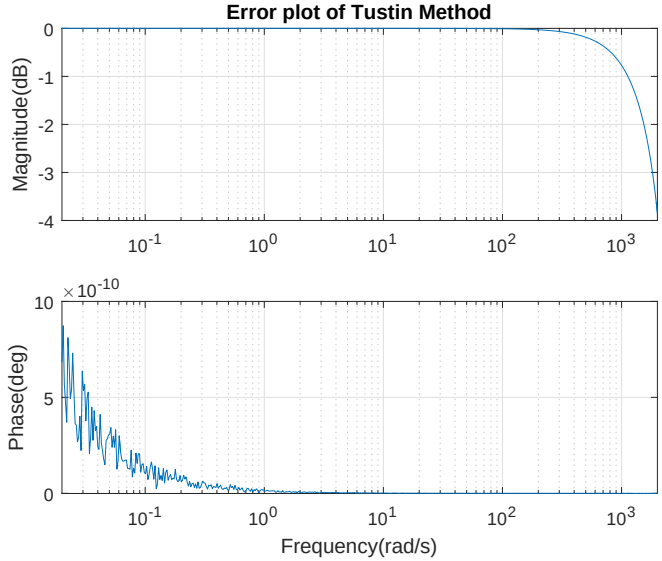

Figure 5.8: Error of Tustin Method compared to first order integral

The error plots demostrate that, the ORA method which takes frequency lower and upper bound is actually have less error in between these ranges. The CFE and Tustin methods show less error in lower frequencies, however in higher frequencies, these methods also starts to generate error. In terms of the error in phase domains, we observe that the Tustin method give less error compared to others. We know that, the servo controller in driver has sampling rate of 1KHz,

and the in torque loop the maximum bandwidth we can obtain is around 200-300 Hz. Therefore, the errors obtained in high frequencies using these approximation methods are negligible in our application.
In figure 5.9, bode plots for different orders of integrations are given.

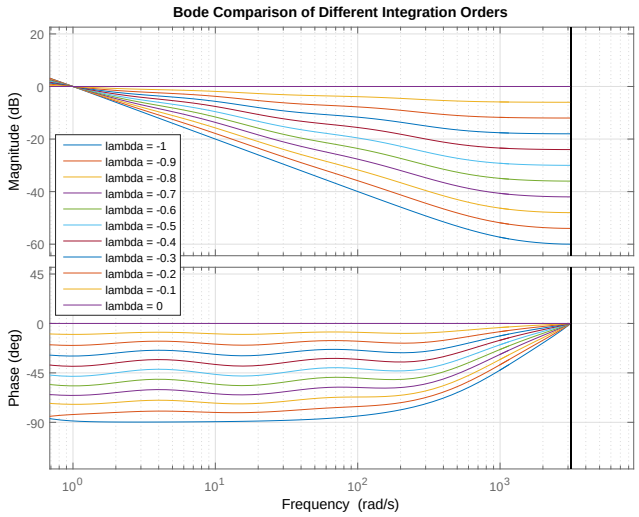

Figure 5.9: Bode Plots for different order Integrations

For different fractional orders we observe that there occurs change in the slope of magnitude plot of the bode. Where classical integral yields 20dB decrease per decade, the fractional order integration yields 20*λ decrease per decade. Therefore, the fractional order controllers will give large flexibility to design with ability to change in integration and derivation order. Depending on the activity, one could select appropriate integration order. For the bode plots of fractional order integrations, the result gets closer to classical integration as order increases up to -1.
Next, the integer controller is designed with first order integral actions approximated by the 3 different aforementioned methods, and the proportional term of the controller is considered as $K_p = 0.25$. The integral coefficient K_i is chosen as 5. It must be noted that these values are selected only for illustration and comparison purposes. Here, we aim to justify that our approximation methods with same controller parameters give the similar responses. Therefore, these parameters are selected same for all three methods.

While approximating these controllers it has been observed that the controller $G_{C_{ORA}}$ makes system unstable after approximation order N=3. The reason makes system unstable is that the gain component of H(s) in (3.70) becomes considerably high. Therefore, for $G_{C_{ORA}}$, N is picked as two. For other methods $G_{C_{Tustin}}$ and $G_{C_{CFE}}$ there was no issue for higher realization orders. In figure 5.10, the step response of the identified plant is given with these controllers.

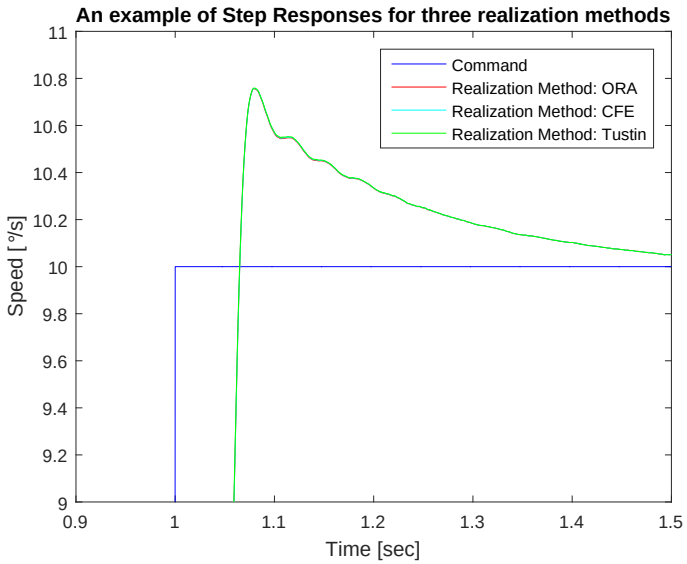

Figure 5.10: Speed Step Responses for Approximated PI Controllers by 3 Different Methods

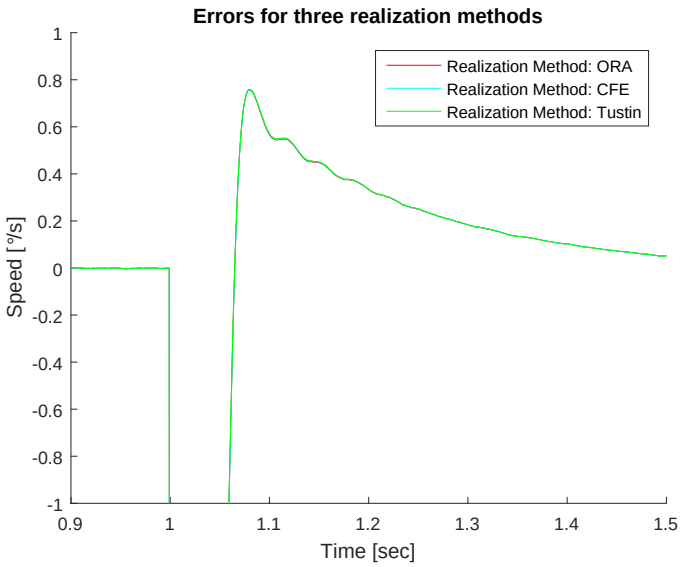

Figure 5.11: Error for Approximated PI Controllers by 3 Different Methods

In our trials, we obtained similar results with different parameters, so that we have chosen to give response of the controller with these values as an example. So, it can be stated that all controllers gave the almost same result with same controller parameters for the 3 different realization methods. These realizations will be used in all of the simulation tests and hardware experimental studies.

5.3 Controller Design and Tuning Procedures

In this section, fractional order controller design together with its tuning procedures will be discussed. Firstly, the tuning nodes of the fractional order controller and their effect to the system's response will be investigated. Then, we will use MATLAB Optimization toolbox to optimize fractional order PID controller parameters for specific performance indexes given in section 5.4.2.

5.3.1 Tuning Rules

As a controller, $PI^{\lambda}D^{\mu}$ controller has five parameter to be tuned which are K_P,K_I,K_D,λ and μ. The fractional order controller will be in the form:

$$G_c(s) = K_p[1 + K_I s^{-\lambda}] + K_D s^{\mu} \tag{5.5}$$

Comparing the classical PID controller, there are extra two parameters to be tuned: Both integer and fractional order controllers have K_P, K_I and K_D in common to be tuned, but fractional order controllers have the as derivation and integration orders μ and λ to be tuned as well. In this section, the effects of these five parameters are studied. When studying the effects of each parameters, the other parameters are kept constant. In the following subsections effect of these parameters are illustrated with their effect on the step response of the system. In order to compare effects of each parameter, several simulation are done. These simulations include both integer and fractional order controllers. In the table below, the simulations are presented which are performed and their results are presented in upcoming sections. In table 5.1, the parameters which are kept constant are shown with the numerical value which they took in the simulations.

The remaining parameter is variable, and its effect to step response is presented. All of the simulations are using the simulink model validated before in section

#	K_P*	K_I	K_D	λ	μ	Controller Type
1	K_P*	0	0	-1	1	P
2	K_P*	2	0	-1	1	PI
3	K_P*	2	0	-0.5	1	PI^{λ}
4	K_P*	2	0.01	-1	1	PID
5	K_P*	2	0.01	-0.5	0.5	$PI^{\lambda}D^{\mu}$
6	0.25	K_I*	0	-1	1	PI
7	0.25	K_I*	0	-0.5	1	PI^{λ}
8	0.25	K_I*	0.01	-1	1	PID
9	0.25	K_I*	0.01	-0.5	0.5	$PI^{\lambda}D^{\mu}$
10	0.25	1	0	$\lambda*$	1	PI^{λ}
11	0.25	1	0.01	$\lambda*$	0.5	$PI^{\lambda}D^{\mu}$
12	0.25	2	K_D*	-1	1	PID
13	0.25	2	K_D*	-0.5	0.5	$PI^{\lambda}D^{\mu}$
14	0.25	5	0.01	-0.5	$\mu*$	$PI^{\lambda}D^{\mu}$

Table 5.1: Simulations Overview

4.4. The results for these simulations are given in following sections with simulation numbers. In table, the constants are shown with their numerical values and the variables are shown with the star mark on them. In each simulation, we present the response of the system for the changes on these variables.

5.3.1.1 Effect of Kp

When the effect of controller parameter K_p is considered, it is expected that for classical integer order controller;

- High values of gain makes system more insensitive to the disturbances,
- Too large gain makes system more sensitive to the sensor noises, measurement noises,
- Steady state error decreases with higher gain,
- Oscillation increases with higher gain,

In figure 5.12 , the step response of the plant with $G_C = P$ is given with different K_P values.

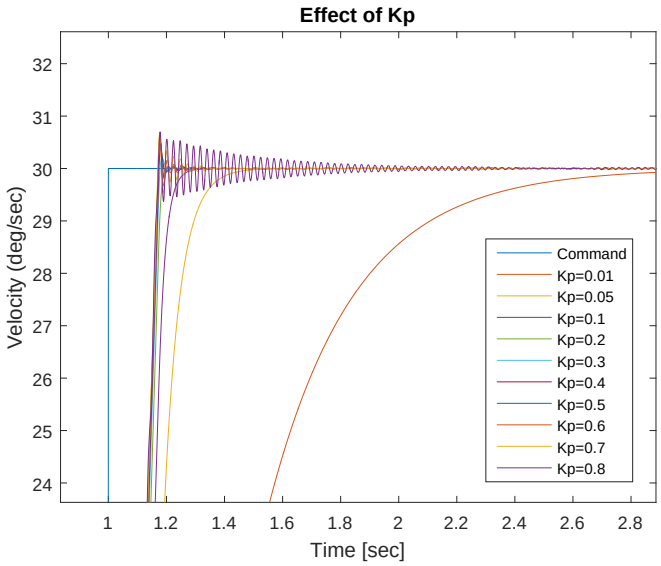

Figure 5.12: Effect of K_p in Simulation 1

The following performance values are obtained from the test results.

K_P Value	Rise Time(sec)	Maximum Overshoot(deg)
0.01	N.A	N.A
0.05	1.758	0.001
0.1	1.400	0.002
0.2	1.200	0.008
0.3	1.180	0.201
0.4	1.176	0.391
0.5	1.174	0.529
0.6	1.173	0.618
0.7	1.172	0.656
0.8	1.174	0.699

Table 5.2: Performance Values for Simulation 1

So, the settling time decreases with increase in K_P as seen on figure 5.12. Similarly, the steady state error also decreases as K_P increases. It must be noted that for $K_P = 0.8$ system became unstable and illustrates oscillatory behavior. The performance values measured for $K_P = 0.8$ should not be taken into consideration. Excluding unstable behavior at higher value of K_P, the responses obtained with different K_P's is expected, as there are no fractional order component.

In figure 5.13 , the step response of the plant with $G_C = PI$ is given with different K_P values. In this case, K_I value is constant and equal to 2.

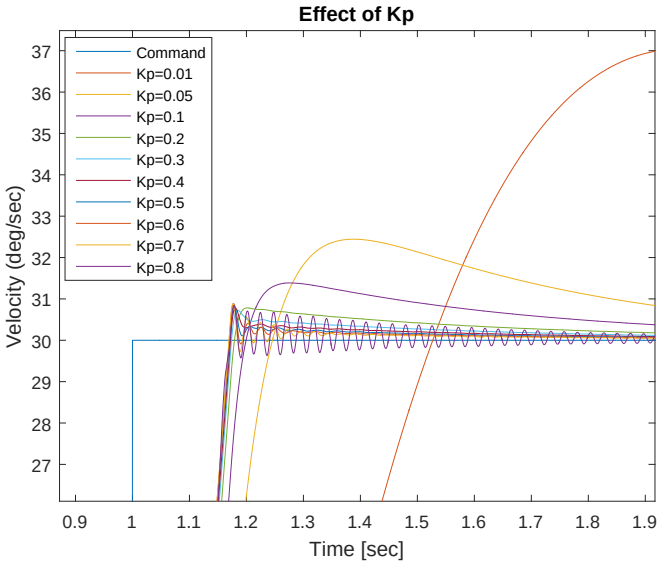

Figure 5.13: Effect of K_p in Simulation 2

The following performance values are obtained from the test results.

K_P Value	Rise Time(sec)	Maximum Overshoot(deg)
0.01	1.529	7.051
0.05	1.247	2.443
0.1	1.201	1.386
0.2	1.181	0.782
0.3	1.175	0.729
0.4	1.174	0.790
0.5	1.172	0.852
0.6	1.171	0.888
0.7	1.171	0.881
0.8	1.173	0.842

Table 5.3: Performance Values for Simulation 2

With integer order integrator, the increase in K_P shows the expected results. The response becomes faster, however at some point oscillations start on the system.

In figure 5.14, the step response of the plant with $G_C = PI^{\lambda}$ is given with different K_P values. In this case, K_I and λ values are constant and equal to 2 and -0.5 respectively.

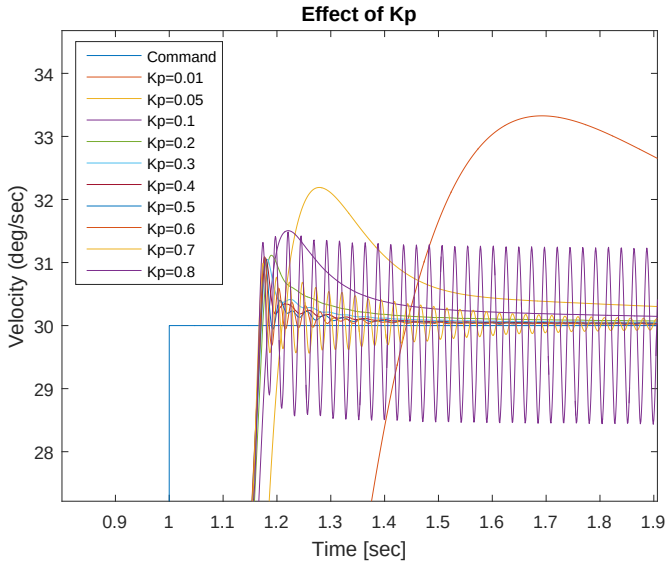

Figure 5.14: Effect of K_p in Simulation 3

About the results, following performance values are obtained:

K_P Value	Rise Time(sec)	Maximum Overshoot(deg)
0.01	1.442	3.326
0.05	1.211	2.191
0.1	1.185	1.506
0.2	1.175	1.115
0.3	1.173	1.049
0.4	1.172	1.062
0.5	1.171	1.076
0.6	1.170	1.092
0.7	1.168	0.986
0.8	1.169	1.485

Table 5.4: Performance Values for Simulation 3

In this case, responses are very similar to integer order PI case. However, at higher gain, the oscillations have higher amplitudes. In fractional order controller, the integration order is chosen as -0.5 for comparison purposes. We know that as integration order goes to zero, the integration becomes an identity operator from the features of integro-differential operator. In this case we obtain it as gain factor to system's response, so the response have amplified oscillation

at higher gains.

In figure 5.15, the step response of the plant with $G_C = PID$ is given with different K_P values. In this case, K_I and K_D values are constant and equal to 2 and 0.01 respectively.

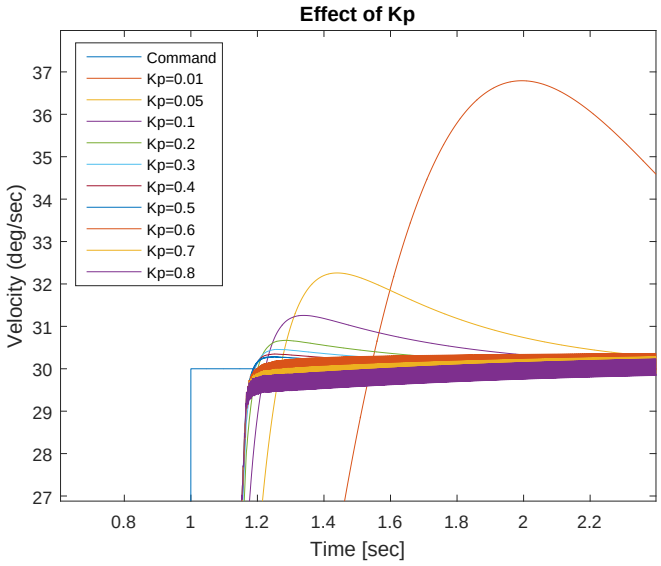

Figure 5.15: Effect of K_p in Simulation 4

The following performance values are obtained from the results.

K_P Value	Rise Time(sec)	Maximum Overshoot(deg)
0.01	1.542	6.792
0.05	1.270	2.259
0.1	1.225	1.258
0.2	1.202	0.670
0.3	1.195	0.456
0.4	1.191	0.348
0.5	1.188	0.287
0.6	1.198	0.397
0.7	1.280	0.393
0.8	1.570	0.390

Table 5.5: Performance Values for Simulation 4

The results shows that increasing K_P after 0.5, makes the system unstable. The settling time decreases as K_P increases until system becomes unstable.
In this analysis, in order to get reasonable results, we have discarded the gyro noise term in the model. The output of derivative term with gyro noise makes the system unstable. The result is expected as a derivative action generates a problematic behavior when there is a high level of noise and a delay in the system. In figure 5.16, the exact simulation is performed while there is a gyro noise in the model. The effect of derivative action and the problematic behavior is illustrated in the figure. System was not responding the changes in K_P.

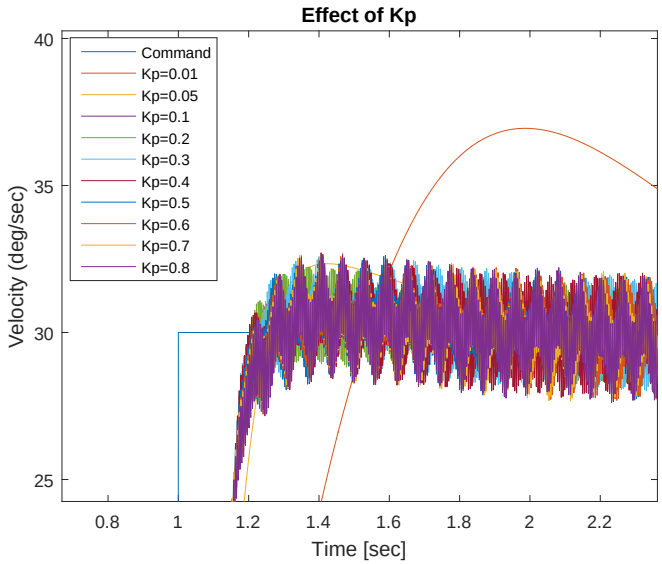

Figure 5.16: Effect of K_p in Simulation 4 with gyro noise in the model

In the following simulations, when there is derivation term in the controller,it must be noted that the gyro noise term is discarded from the model. Gyro noise level was high enough to make system unstable and uncontrollable with derivative action in the controller. Therefore, reasonable results are obtained with removing derivation in the controller. It must be noted that, in real system, the derivative term is also not used in the controllers because of the similar problems. All of the controllers in our applications are in the form of PI for speed loops.

In figure 5.17, the step response of the plant with $G_C = PI^{\lambda}D^{\mu}$ is given with

different K_P values. In this case, K_I, K_D, λ and μ values are constant and equal to 2,0.01,-0.5,0.5 respectively.

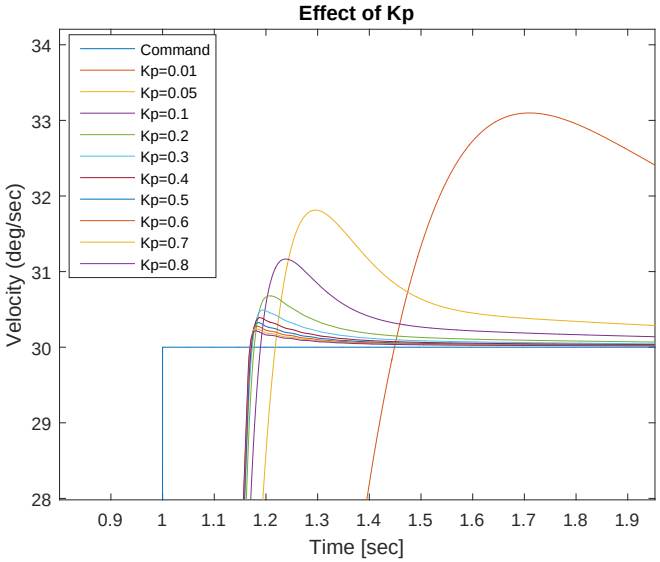

Figure 5.17: Effect of K_p in Simulation 5

The following performance values are obtained for the curves of figure 5.17:

K_P Value	Rise Time(sec)	Maximum Overshoot(deg)
0.01	1.450	3.098
0.05	1.219	1.813
0.1	1.191	1.166
0.2	1.179	0.679
0.3	1.175	0.491
0.4	1.173	0.392
0.5	1.172	0.326
0.6	1.171	0.280
0.7	1.171	0.247
0.8	1.171	0.221

Table 5.6: Performance Values for Simulation 5

In the figure 5.17 and table 5.6, we obtain better responses than integer order PID case. We observe that with higher gain, system does not oscillate and does not become unstable. We are able to increase the gain without disturbing the stability of the system via fractional order integration and differentiation.

5.3.1.2 Effect of Ki

When the effect of controller parameter K_i is considered, it is expected that for classical integer order controller;

- Steady state error is removed with integral term
- Short integration time leads to oscillation

In figure 5.18, the step response of the plant with $G_C = PI$ is given with different K_I values. In this case, K_P value is constant and equal to 0.25.

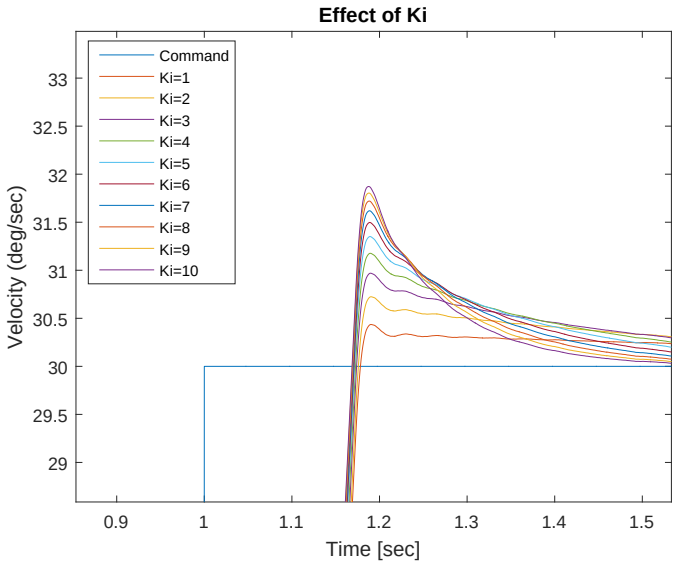

Figure 5.18: Effect of K_I in Simulation 6

The following performance values are calculated:

K_I Value	Rise Time(sec)	Maximum Overshoot(deg)
1	1.180	0.436
2	1.177	0.724
3	1.175	0.970
4	1.174	1.177
5	1.173	1.351
6	1.172	1.498
7	1.172	1.619
8	1.171	1.720
9	1.171	1.804
10	1.170	1.872

Table 5.7: Performance Values for Simulation 6

These results are from the integer order PI controller. As K_I increases, steady state error decreases, and settling time decreases.

In figure 5.19, the step response of the plant with $G_C = PI^{\lambda}$ is given with different K_I values. In this case, K_P and λ values are constant and equal to 0.25 and -0.5 respectively.

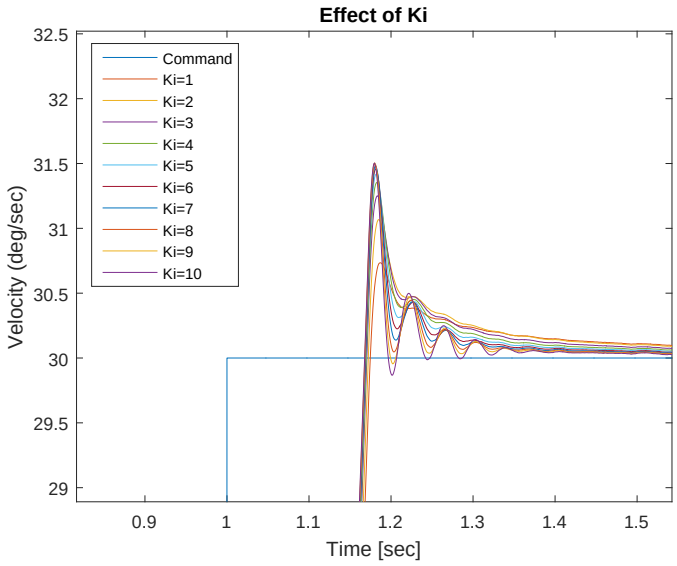

Figure 5.19: Effect of K_I in Simulation 7

The following performance values are obtained.

K_I Value	Rise Time(sec)	Maximum Overshoot(deg)
1	1.176	0.735
2	1.174	1.068
3	1.172	1.250
4	1.171	1.359
5	1.171	1.419
6	1.170	1.457
7	1.170	1.484
8	1.170	1.498
9	1.169	1.505
10	1.169	1.504

Table 5.8: Performance Values for Simulation 7

In this simulation, for the spesific fractional integration order, increasing the K_I integral gain illustrates the similar responses with integer order case. The overshoot increases, however, the settling time decreases, rising time decreases.

In figure 5.20, the step response of the plant with $G_C = PID$ is given with different K_I values. In this case, K_P and K_D values are constant and equal to 0.25 and 0.01 respectively.

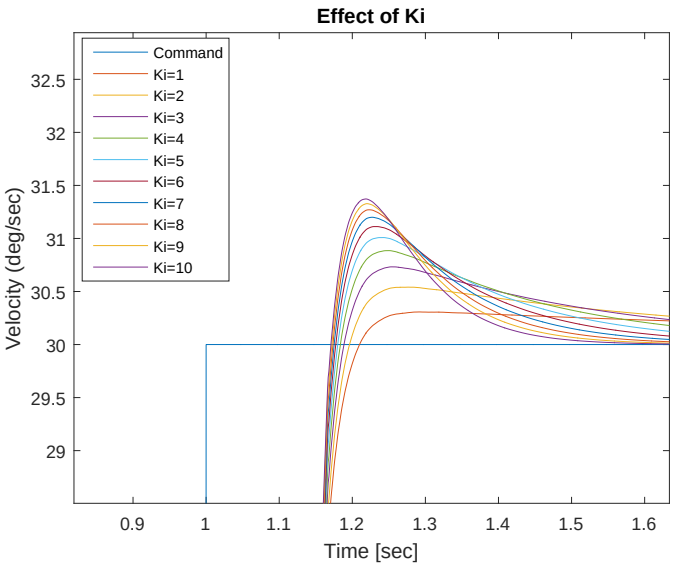

Figure 5.20: Effect of K_I in Simulation 8

The following performance values are obtained from figure 5.20.

K_I Value	Rise Time(sec)	Maximum Overshoot(deg)
1	1.211	0.307
2	1.198	0.541
3	1.190	0.732
4	1.185	0.885
5	1.182	1.009
6	1.179	1.112
7	1.177	1.200
8	1.175	1.270
9	1.174	1.328
10	1.172	1.372

Table 5.9: Performance Values for Simulation 8

For the integer order PID case, the changes in K_I are similar to the previous integer order PI case in simulation 6. The settling time diminishes with the increase in integral gain, and settling time decreases too.

In figure 5.21, the step response of the plant with $G_C = PI^{\lambda}D^{\mu}$ is given with different K_I values. In this case, K_P, K_D, λ and μ values are constant and equal to 0.25, 0.01, -0.5 and 0.5 respectively.

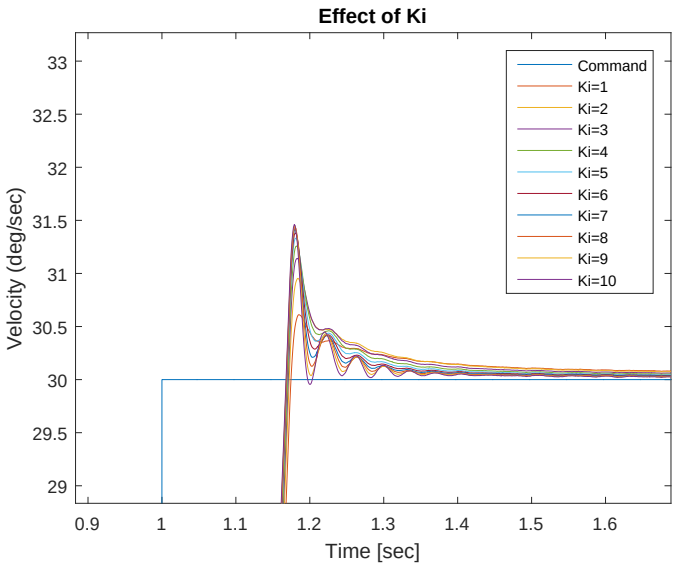

Figure 5.21: Effect of K_I in Simulation 9

About the results, the following performance values are obtained:

K_I Value	Rise Time(sec)	Maximum Overshoot(deg)
1	1.211	0.307
2	1.198	0.541
3	1.190	0.732
4	1.185	0.885
5	1.182	1.009
6	1.179	1.112
7	1.177	1.200
8	1.175	1.270
9	1.174	1.328
10	1.172	1.372

Table 5.10: Performance Values for Simulation 9

The responses for the fractional order $PI^{\lambda}D^{\mu}$ case is also similar with the PI^{λ} case. Rise time decreases, maximum overshoot increases and more oscillatory behavior is obtained as integral coefficient increases.

5.3.1.3 Effect of λ

In figure 5.22, the step response of the plant with $G_C = PI^{\lambda}$ is given with different λ values. In this case, K_P and K_I values are constant and equal to 0.25 and 2 respectively.

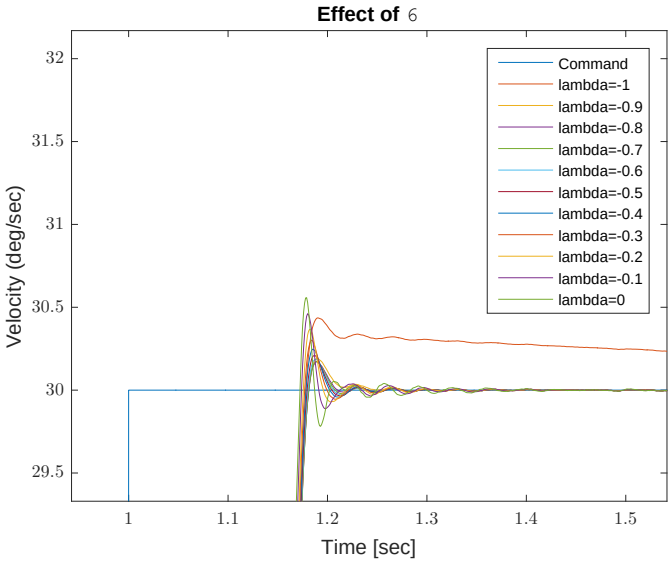

Figure 5.22: Effect of λ in Simulation 10

About the above results, the following performance values are obtained:

λ Value	Rise Time(sec)	Maximum Overshoot(deg)
-1	1.180	0.436
-0.9	1.183	0.192
-0.8	1.183	0.171
-0.7	1.183	0.174
-0.6	1.182	0.186
-0.5	1.181	0.209
-0.4	1.180	0.247
-0.3	1.179	0.302
-0.2	1.177	0.367
-0.1	1.175	0.461
0	1.174	0.558

Table 5.11: Performance Values for Simulation 10

In the figure 5.23, the integral output of the controller is given separately in order to observe results.

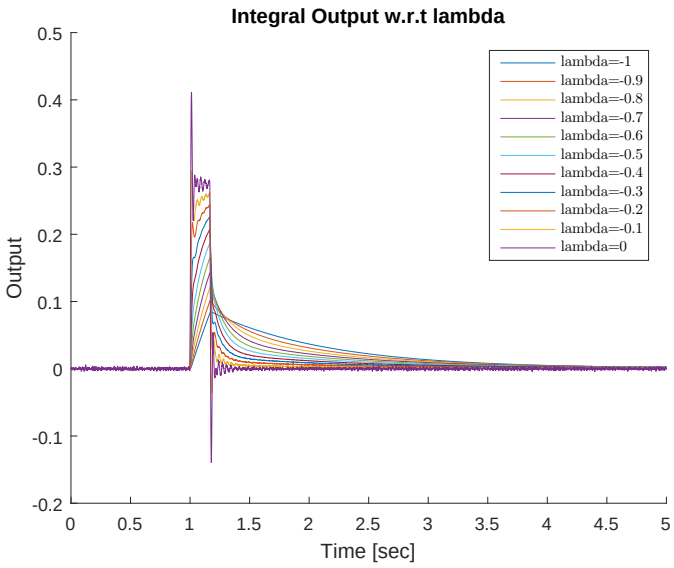

Figure 5.23: Integral output of different order of integrations

In the figure 5.23, the step response of the controller with different fractional order integral action is given. When $\lambda = 0$, the integral action vanishes, and it becomes classical proportional action. As λ gets closer to -1, the classical integral, its response gets closer to the integer order integral action. We state that Integral effect of controller gets stronger as λ becomes higher, and gets closer to the integer order integral value -1. For fractional values of λ, output behaves like both proportional and integral action but with diminished effects.

In figure 5.24 , the step response of the plant with $G_C = PI^{\lambda}D^{\mu}$ is given with different λ values. In this case, K_P, K_I, K_D and μ values are constant and equal to 0.25, 1, 0.01 and 0.5 respectively.

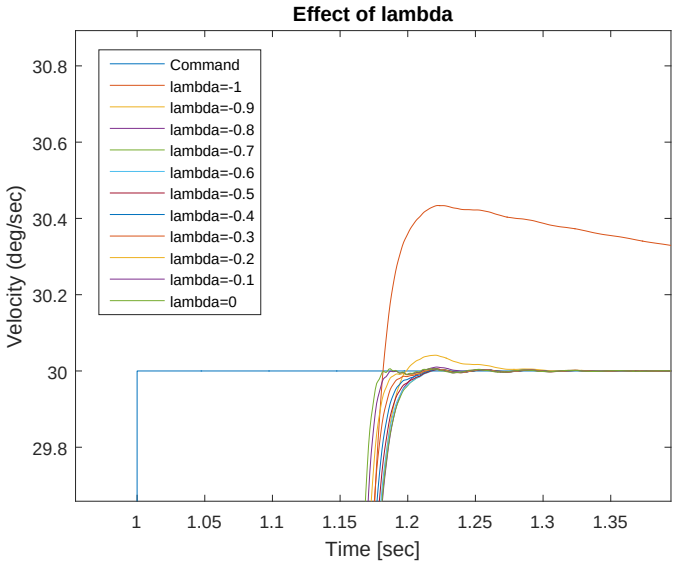

Figure 5.24: Effect of λ in Simulation 11

About the above results, the following performance values are obtained as:

λ Value	Rise Time(sec)	Maximum Overshoot(deg)
-1	1.183	0.4338
-0.9	1.200	0.0413
-0.8	1.213	0.0103
-0.7	1.219	0.0040
-0.6	1.220	0.0029
-0.5	1.219	0.0035
-0.4	1.217	0.0045
-0.3	1.215	0.0057
-0.2	1.213	0.0063
-0.1	1.212	0.0064
0	1.187	0.0060

Table 5.12: Performance Values for Simulation 11

These results illustrates that fractional order $PI^{\lambda}D^{\mu}$ with lower λ behaves like a controller whose proportionality term is stronger. On the other hand, as λ gets closer to -1, the classical integer order integration, the integral term of the controller gets stronger. Therefore, the order of the integration must be determined depending on the systems where fractional order controller to be

implemented.

5.3.1.4 Effect of Kd

When the effect of controller parameter K_d is considered, it is expected that for classical integer order controller;

- Derivative term can predict output
- Response will be faster and stable
- Noises in the system will make control problematic when there are delays
- Fast changes in the reference signal will result in control signal saturation

In figure 5.25, the step response of the plant with $G_C = PID$ is given with different K_D values. In this case, K_P and K_I values are constant and equal to 0.25 and 2 respectively.

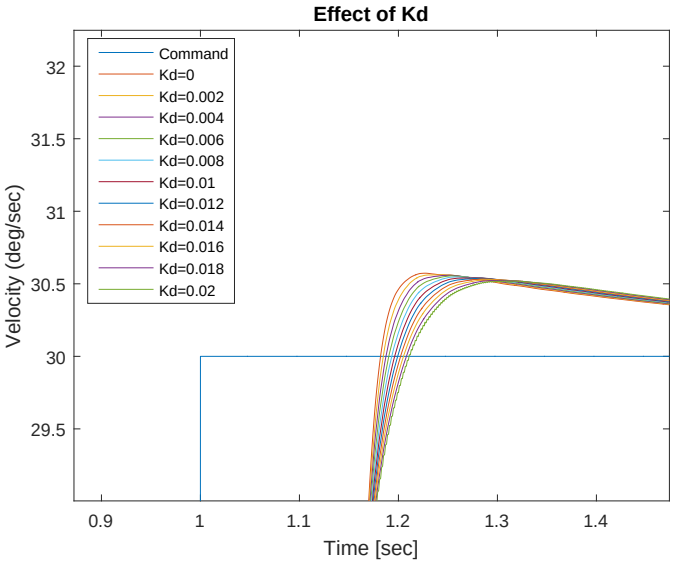

Figure 5.25: Effect of K_D in Simulation 12

About the results, the following performance values are obtained:

K_D Value	Rise Time(sec)	Maximum Overshoot(deg)
0	1.1830	0.5731
0.002	1.186	0.562
0.004	1.189	0.560
0.006	1.192	0.555
0.008	1.195	0.548
0.01	1.198	0.541
0.012	1.201	0.538
0.014	1.204	0.533
0.016	1.207	0.528
0.018	1.210	0.522
0.02	1.212	0.521

Table 5.13: Performance Values for Simulation 12

In figure 5.25, the derivative term makes system faster as its coefficient K_D increases. However, the rising time is also increases with K_D.
In figure 5.26 , the step response of the plant with $G_C = PI^{\lambda}D^{\mu}$ is given with different K_D values. In this case, K_P, K_I, λ and μ values are constant and equal to 0.25, 2, -0.5 and 0.5 respectively.

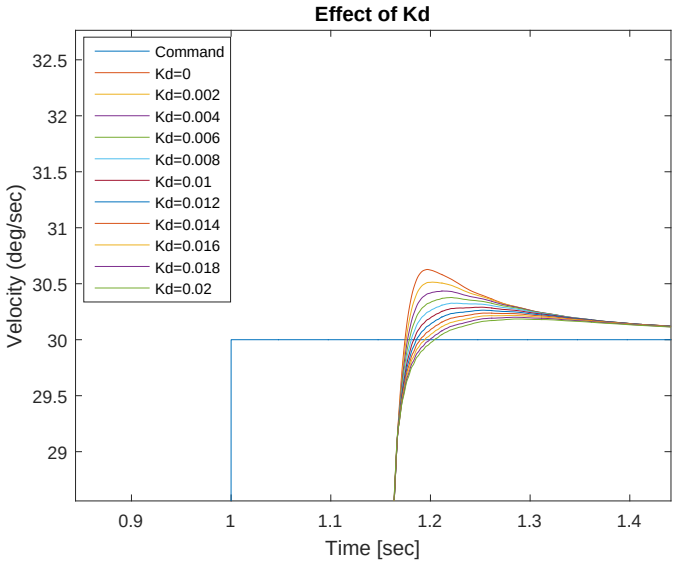

Figure 5.26: Effect of K_D in Simulation 13

About the results, the following performance values are obtained:

K_D Value	Rise Time(sec)	Maximum Overshoot(deg)
0	1.176	0.627
0.02	1.177	0.514
0.04	1.178	0.435
0.06	1.180	0.377
0.08	1.182	0.326
0.1	1.185	0.291
0.12	1.187	0.264
0.14	1.191	0.238
0.16	1.196	0.217
0.18	1.201	0.201
0.2	1.205	0.185

Table 5.14: Performance Values for Simulation 13

When the effect of K_D is considered, it is shown that, higher K_D is seems to yield lower overshoot, and shorter settling time. However, the rise time also increases as K_D increases. We see that the response gets more stable and closer to the command input.

5.3.1.5 Effect of μ

In figure 5.27, the step response of the plant with $G_C = PI^{\lambda}D^{\mu}$ is given with different μ values. In this case, K_P, K_I, K_D and λ values are constant and equal to 0.25, 5, 0.01 and -0.5 respectively.

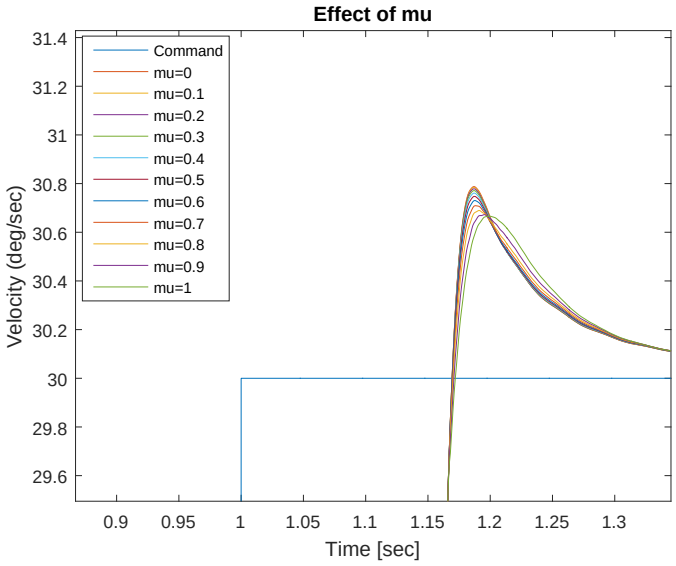

Figure 5.27: Effect of μ in Simulation 14

About the results, the following performance values are obtained:

μ Value	Rise Time(sec)	Maximum Overshoot(deg)
0	1.171	0.788
0.1	1.171	0.784
0.2	1.171	0.779
0.3	1.171	0.771
0.4	1.171	0.761
0.5	1.171	0.748
0.6	1.171	0.730
0.7	1.171	0.709
0.8	1.172	0.690
0.9	1.172	0.673
1	1.173	0.666

Table 5.15: Performance Values for Simulation 14

In the figure 5.27, we can see the effect of derivative term with different orders of μ. As μ gets closer to zero, the response gets close to the classical proportional controller. As μ increases we see faster and more overdamped response. These are also expected results from derivative action. It must be noted that in order to add derivative action to the model, the gyro noise and the time delay of

the plant are removed from the model. Otherwise, we were not able to get the meaningful results as systems gets unstable. In the figure 5.28 the output of the derivative term is drawn separately. The result shows that as μ goes to 1, the effect of the derivative terms gets bigger. Having derivation order μ less than 1 leads responses weaker than the classical integer order derivation depending on the order level. These results demonstrates that fractional order derivation is not classical derivation however, it is a weaker version of it depending on the order μ.

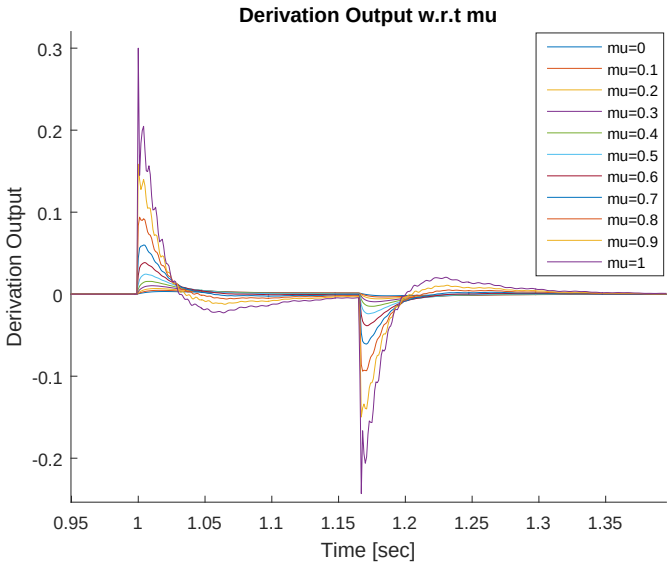

Figure 5.28: Derivative output of different order of derivations

5.3.2 Controller Design by Optimization

In the previous sections, the effects of the controller parameters are discussed. Tuning these parameters at the same time by hand would have been a difficult and non-systematic task. Therefore a systematic way of tuning these parameters must be developed.It should be again mentioned that adding derivative term were problematic due to high gyroscope noise level, in these analysis we optimized integer order PI controller with fractional order PI^{λ} controller.
Optimizations procedure mainly optimizes the controller parameters by minimizing the selected cost function. In the analysis here, we optimized the speed loop controller parameters. MATLAB's Optimization Toolbox, Global Opti-

mization Toolbox and Parallel Computing Toolbox are used to find the best controller parameters. The results that we obtain must give the global optimum solution in the given interval for the optimization parameters to validate the comparison between integer order and fractional order controllers. To obtain the global optimum solution, we have used two different algorithm, which are Genetic Algorithm and the Global Search. Both of these algorithms are implemented using MATLAB's toolboxes.

In order to tune these parameters in the simulation tests, we will make use of the MATLAB's Optimization toolbox. We will use commonly used cost functions, which depends on the error which are all listed below. For speed loop tuning, the PI controller parameters will be tuned. The error will be calculated and fed into these cost functions;

- ITAE:Integral Time Absolute Error

$$J = \int_0^t t|e(t)|dt \tag{5.6}$$

- ITSE:Integral Time-Square Error

$$J = \int_0^t te(t)^2 dt \tag{5.7}$$

- IAE:Integral Absolute Error

$$J = \int_0^t |e(t)|dt \tag{5.8}$$

- ISE:Integral Square Error

$$J = \int_0^t e^2(t)dt \tag{5.9}$$

These cost functions are calculated on the simulink model, and used as performance criteria measurements.

The Simulink implementation of the cost functions are shown in the figure 5.29.

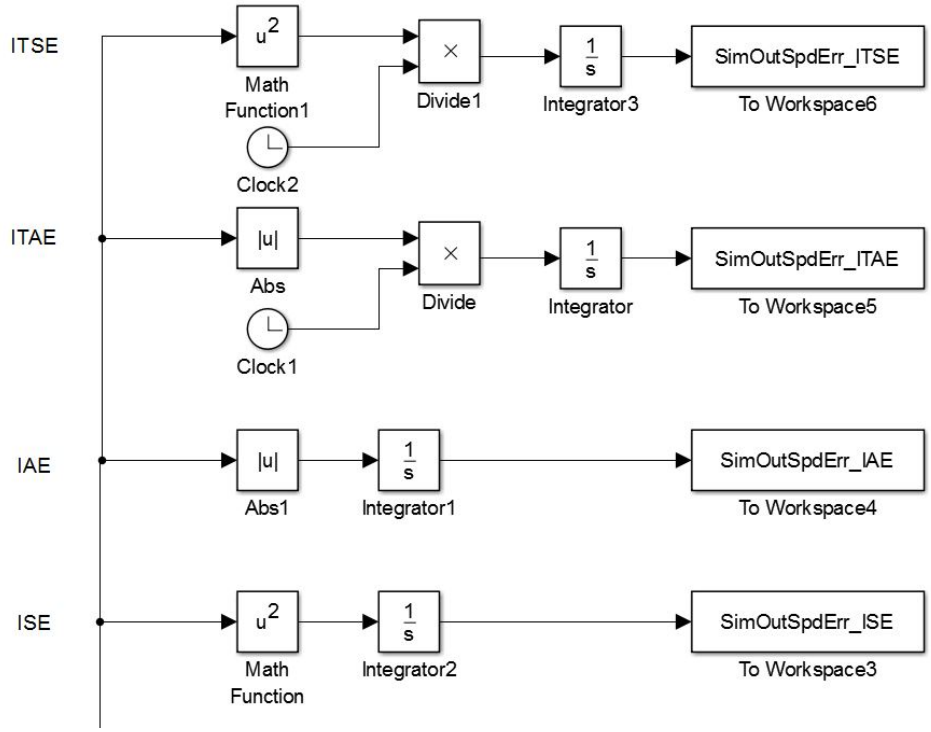

Figure 5.29: Performance indexes on Simulink Model

In addition to these cost functions, we have tried some other custom cost evaluations. However, results were skipped here as they did not give any different results than the previously defined four cost functions. These custom cost evaluations were;

- Integral Square Error plus Square of Derivative of Error

$$J = \int_0^t e^2(t) + \dot{e}^2(t)dt \tag{5.10}$$

- Integral Absolute Error plus Absolute of Derivative of Error

$$J = \int_0^t |e(t)| + |\dot{e}(t)|dt \tag{5.11}$$

- ITSE:Integral Time-Square Error with t_0 equals to first crossing of speed feedback and command. In this cost function, the first rising triangular part was ignored.

$$J = \int_{t_0}^t te(t)^2dt \tag{5.12}$$

The details for the result of the optimization procedures and obtained parameters are given in the following section 5.4.2.1.

5.3.2.1 Optimization Results

Genetic Algorithm and Global Search are both used for determining optimized parameters for the given cost evaluation. In our problem, these optimization routines are performed for both elevation and traverse axis separately to find the best controllers in the form of PI for integer order case and PI^{λ} for fractional order case.
For fractional order controller, three controllers approximated by the 3 aforementioned different methods with fixed realization order N depending on method. Optimization routines are performed for four different performance indexes which were Integral Time Square Error, Integral Square Error, Integral Absolute Error and Integral Time Absolute Error. For integer order controller, there were only one optimization routines for different axes.
The optimization procedure requires some inputs to run method and to find the opptimal points. The required information to run optimization procedures are given as,

- **The cost function**: It is the function where optimization procedure will try to minimize. In our case, we use four different cost evaluation functions. In following chapters, we recall these cost evaluation functions as performance indexes.

- K_P**,** K_I **and** λ: These are the outputs of the optimization procedure and will be the optimal controller parameters. These parameters are bounded within specific ranges. The integration order must be bounded in between 0 and -1, and search ranges are selected as $K_P \in$ [0,2] and as $K_I \in$ [0,20].

- **The speed error,** $\triangle V$: The speed error(shown as e(t) in cost evaluation functions) measured from the simulink model will feed into optimization procedure.

The following simulink model is used for the optimization procedures;

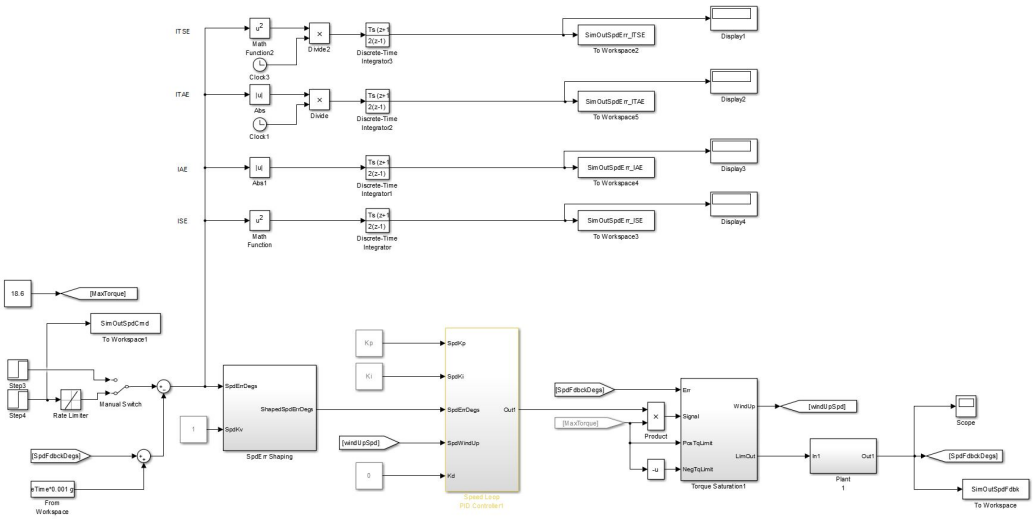

Figure 5.30: Performance indexes on Simulink Model

All of these optimization routines are executed in a workstation with Intel(R) Core(TM) i7-6820HQ CPU @2.7GHz, 32 Gb RAM with 64-bit operating system and MATLAB R2015b. Parallel computing is activated in these tests and all cores were used by simulations. When execution time of the optimization procedure is measured, a dramatic difference between Genetic Algorithm and Global Search was obtained. A single Genetic algorithm solution is found approximately in 40 minutes to 80 minutes where Global Search is completed in approximately in 3-4 hours per routine. In Genetic Algorithm, we have tried to maximize population size and stall generations. We have not used mutations due to our linear bounds on the controller parameters as they may procedure new generations outside of our bounds.

In the next part, the optimization parameters which are found through optimization routines are given. These parameters are found for each axis traverse and elevation, with two different optimization procedure with four cost functions and with three different approximations methods.

- **Traverse Axis:** The results are given for different performance indexes. There performance indexes are Integral Time Absolute Error (ITAE), In-

tegral Time Square Error (ITSE), Integral Absolute Error (IAE) and Integral Square Error (ISE). These optimization routines are applied to three different approximation method which were Oustaloup Recursive Approximation (ORA), Continuous Fraction Expansion (CFE) and Tustin's direct approximation method. In table 5.16, the optimization results are

	ITAE		
	ORA	CFE	Tustin
Kp	0.5343	0.6269	0.5756
Ki	18.9317	18.5387	19.9946
λ	-0.8703	-0.996	-0.9972
Fval	0.1461	0.1567	0.1529
	ITSE		
	ORA	CFE	Tustin
Kp	0.5726	0.6305	0.6510
Ki	17.9648	19.8966	19.2280
λ	-0.822	-0.9564	-0.9754
Fval	0.0616	0.0598	0.0613
	IAE		
	ORA	CFE	Tustin
Kp	0.5329	0.6209	0.6490
Ki	19.7198	19.9867	19.9749
λ	-0.8785	-0.9935	-0.9981
Fval	0.1019	0.1080	0.1081
	ISE		
	ORA	CFE	Tustin
Kp	0.5672	0.6282	0.6548
Ki	16.1737	16.2816	18.3463
λ	-0.7873	-0.9566	-0.9740
Fval	0.0563	0.0574	0.0565

Table 5.16: Genetic Algorithm Results for Performance Indexes for Traverse Axis

given for Genetic Algorithm for different cost functions and approximations methods. In these results, it is observed that the integration order changes in the range of [-1,-0.8]. K_P and K_I values are very similar in

all simulations. These results show that there are only small differences between the results of optimization routines based on different previously defined cost functions. The integration order in most of the simulations is found to be closer to integer order -1. In order to minimize the error for the step response of the system for simulation time, the controller needs integral action. In section 5.4.1, we illustrated that the lower order of integration makes controller behave more like proportional. Therefore, in these optimization procedures, finding the λ value higher proves that these kind of applications needs integral action on controller.
Global Search results are given in tables 5.17.

	ITAE		
	ORA	CFE	Tustin
Kp	0.5288	0.5765	0.5852
Ki	18.6786	19.1558	19.0056
λ	-0.8650	-0.995	-0.9963
Fval	0.1468	0.1557	0.1579
	ITSE		
	ORA	CFE	Tustin
Kp	0.5848	0.6291	0.6395
Ki	19.9959	19.2331	19.6296
λ	-0.8743	-0.9541	-0.9799
Fval	0.0612	0.0605	0.0612
	IAE		
	ORA	CFE	Tustin
Kp	0.6527	0.6475	0.6412
Ki	19.2720	18.9515	19.2202
λ	-0.9984	-0.998	-0.9974
Fval	0.1120	0.1120	0.1106
	ISE		
	ORA	CFE	Tustin
Kp	0.5622	0.6171	0.6483
Ki	18.9246	19.9535	19.0092
λ	-0.8511	-0.9507	-0.9768
Fval	0.0559	0.0547	0.0560

Table 5.17: Global Search Results for Performance Indexes for Traverse Axis

The results obtained from Global Search method was similar to Genetic Algorithm results. The integration order stays in similar range, K_P and K_P values are similar. In these simulations, firstly, Genetic Algorithm was implemented. However, there was a doubt about that could we get global optimum solution. Therefore, second method Global Search method is implemented. When these results are compared, its observed that Global Search cost function value is always better than Genetic algorithm ones.

Hence, under these optimization routines and optimization parameters, Global Search always give more optimum solution.
The step responses of the controllers obtained by Global Search routine and the corresponding error plots are given in figure 5.31.

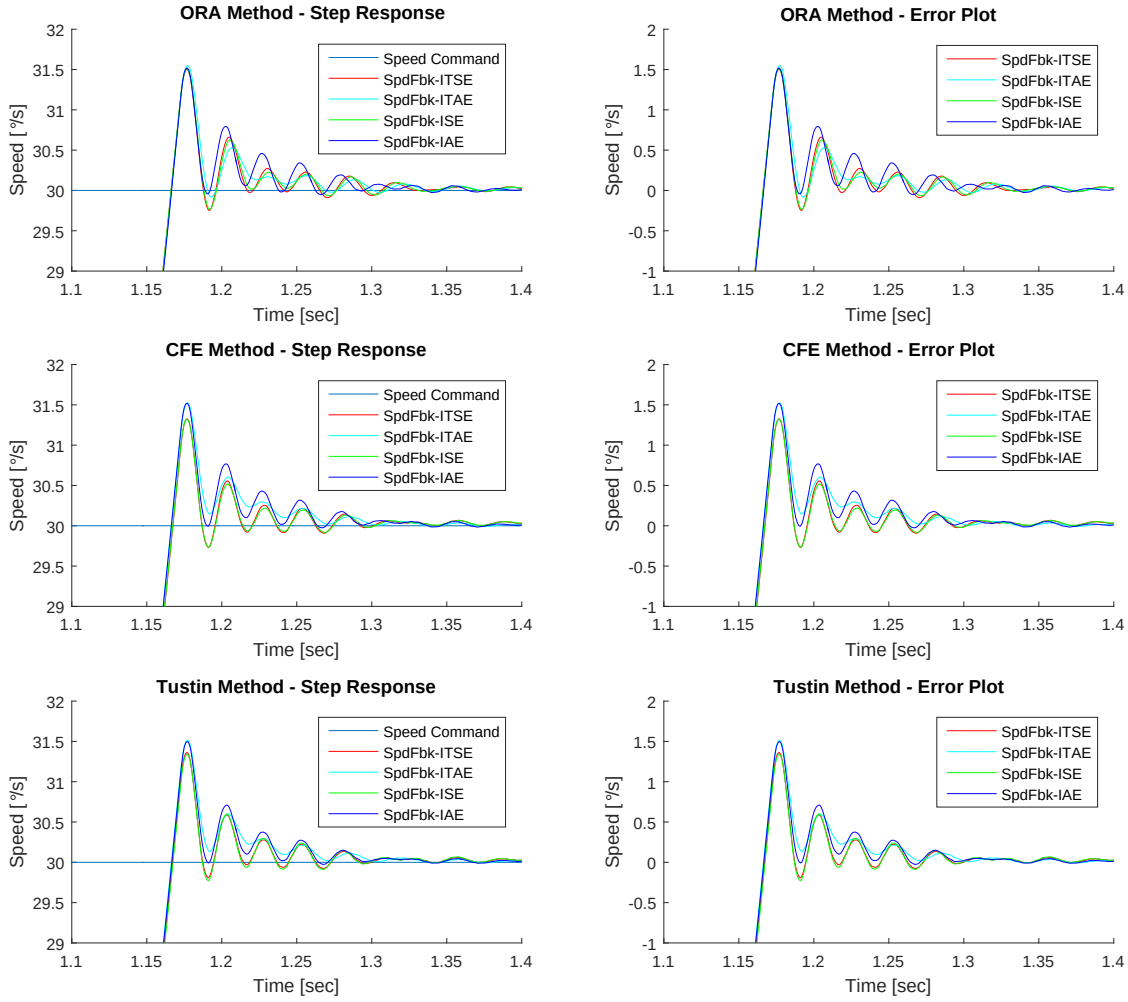

Figure 5.31: Step Responses of Best PI controllers

When these plots are compared, the obtained step responses are very similar for different performance indexes. We have chosen the Integral-Time Square Error performance index to take reference while comparing the

approximation methods. The results of the step responses with ITSE parameters are better than other performance indexes comparing maximum overshoot, oscillations, and settling time. In the figure below, best controllers with ITSE performance index parameters are realized by different methods and compared,

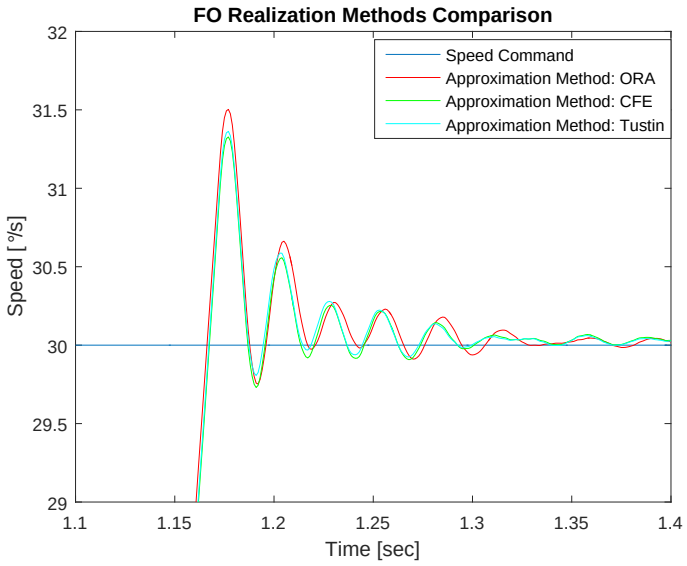

Figure 5.32: Step Responses of Best PI controllers

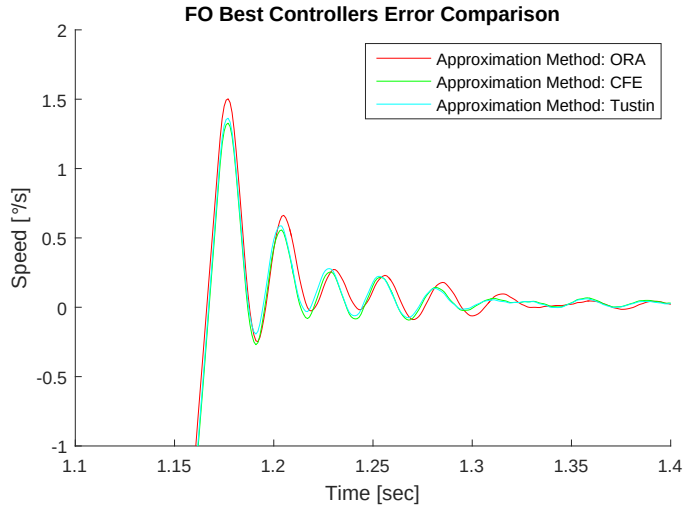

Figure 5.33: Step Responses of Best PI controllers

When realization methods are compared, Recursive Tustin method and

CFE gave similar results and better than Oustaloup Recursive Approximation method. For the analyses in Chapter 6 the controller realized by Recursive Tustin Method will be used.

- **Elevation Axis:**

  Genetic Algorithm results are given in tables 5.18 below for different performance indexes. There were same as traverse axis as Integral-Time Absolute Error(ITAE), Integral-Time Square Error(ITSE), Integral Absolute Error(IAE) and Integral Square Error(ISE).

	ITAE		
	ORA	CFE	Tustin
Kp	0.3232	0.3891	0.414
Ki	19.9922	18.5896	16.9454
λ	-0.8885	-0.994	-0.9968
Fval	0.1253	0.1332	0.1382
	ITSE		
	ORA	CFE	Tustin
Kp	0.3438	0.3812	0.4059
Ki	19.9298	18.890	17.0237
λ	-0.8922	-0.9645	-0.9790
Fval	0.0442	0.0416	0.0427
	IAE		
	ORA	CFE	Tustin
Kp	0.3327	0.3826	0.3882
Ki	19.992	19.9621	19.9903
λ	-0.8994	-0.9944	-0.9975
Fval	0.0848	0.0881	0.0878
	ISE		
	ORA	CFE	Tustin
Kp	0.3471	0.3778	0.3867
Ki	19.9521	19.179	19.9722
λ	-0.9026	-0.9677	-0.9810
Fval	0.0399	0.0376	0.0376

Table 5.18: Genetic Algorithm Results for Performance Indexes

In tables 5.18, the optimization results are given for Genetic Algorithm for the Elevation axis. Similar to Traverse axis, it is observed that the integration order changes in the range of [-1,-0.8]. Its observed that the K_I parameter converges the upper limit in some of the simulations. Global Search results are given in tables 5.19 below for different performance indices, Similar to traverse axis, the results of Global Search are better

	ITAE		
	ORA	CFE	Tustin
Kp	0.3368	0.3246	0.3773
Ki	19.3915	14.4069	19.0039
λ	-0.9374	-0.9652	-0.9959
Fval	0.1287	0.1844	0.1323
	ITSE		
	ORA	CFE	Tustin
Kp	0.3217	0.320	0.4141
Ki	13.7352	18.8846	14.5674
λ	-0.7251	-0.9215	-0.9481
Fval	0.0517	0.0477	0.0453
	IAE		
	ORA	CFE	Tustin
Kp	0.3344	0.3888	0.3911
Ki	19.9295	18.7861	19.7425
λ	-0.9414	-0.9794	-0.9990
Fval	0.0873	0.0958	0.0893
	ISE		
	ORA	CFE	Tustin
Kp	0.330	0.4131	0.3844
Ki	19.01	19.9302	19.0074
λ	-0.8465	-0.9546	-0.9777
Fval	0.0407	0.0399	0.0381

Table 5.19: Global Search Results for Performance Indexes

than Genetic Algorithm. The step responses of the system and the error plots of these step responses are given in figures 5.34.

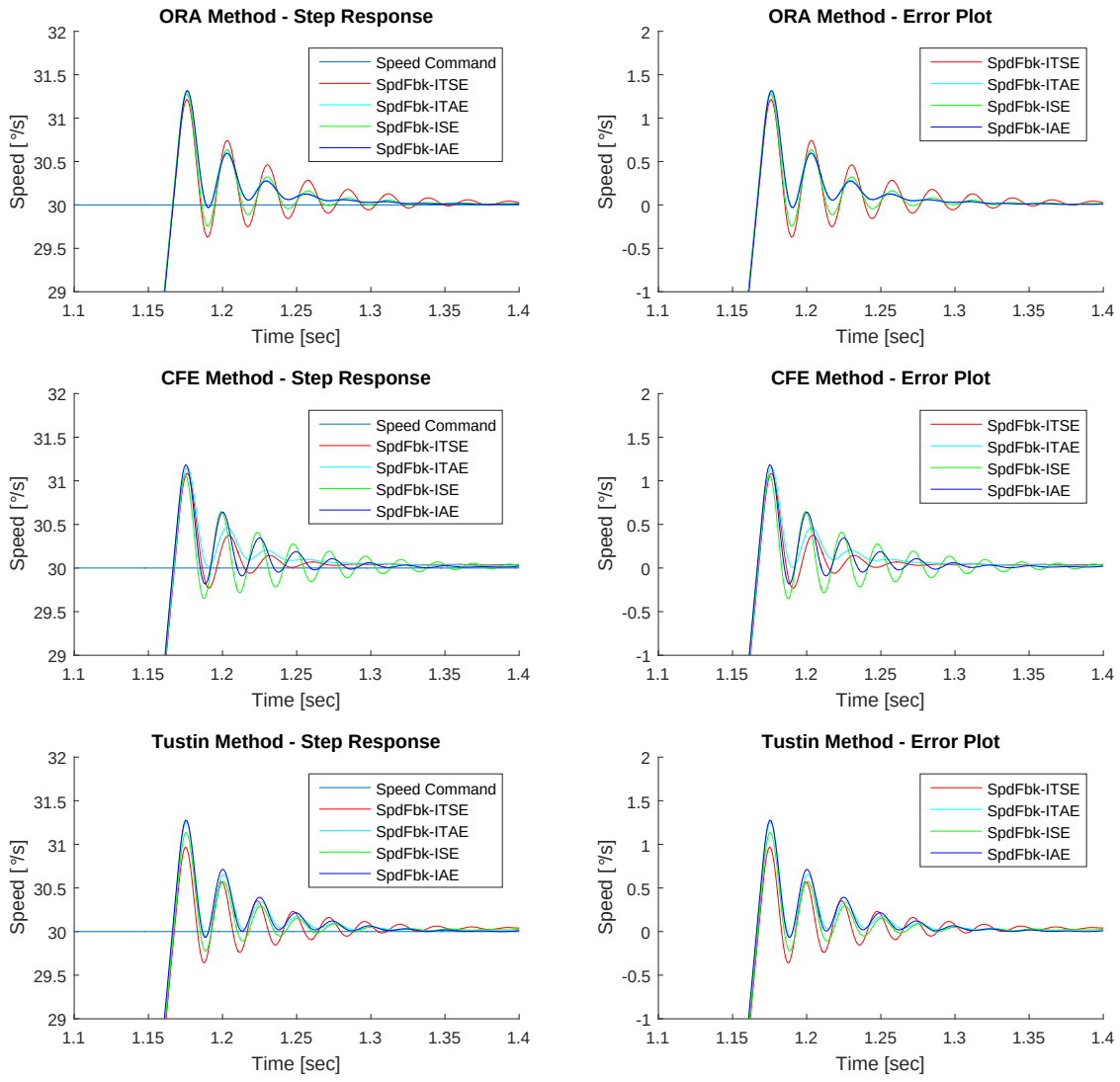

Figure 5.34: Step Responses of Best PI controllers

Similar to traverse axis, the controller parameters found with ITSE cost function gave the better responses in terms of oscillation, overshoot and settling time. Therefore, for the three different approximation method, the best controller are compared in figure 5.35, and the error comparisons of these responses are given in figure 5.36. For the elevation axis, The CFE and Tustin approximation methods gave better response than ORA method.

These results illustrated that the controller parameters optimized by Global

Search method using ITSE cost function and approximated by Tustin Recursive Algorithm gave the best result among other options. Therefore, in Chapter 6, when Integer and Fractional order controllers are compared, these parameters will be used.

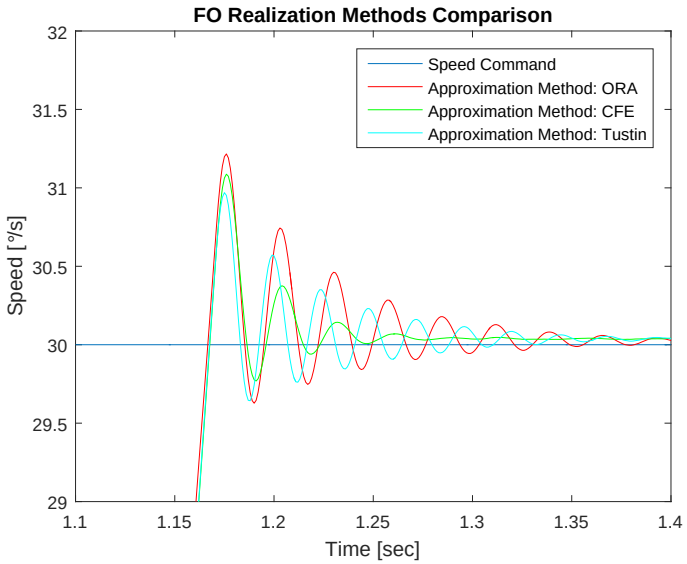

Figure 5.35: Step Responses of Best PI controller Realized by Recursive Tustin Method

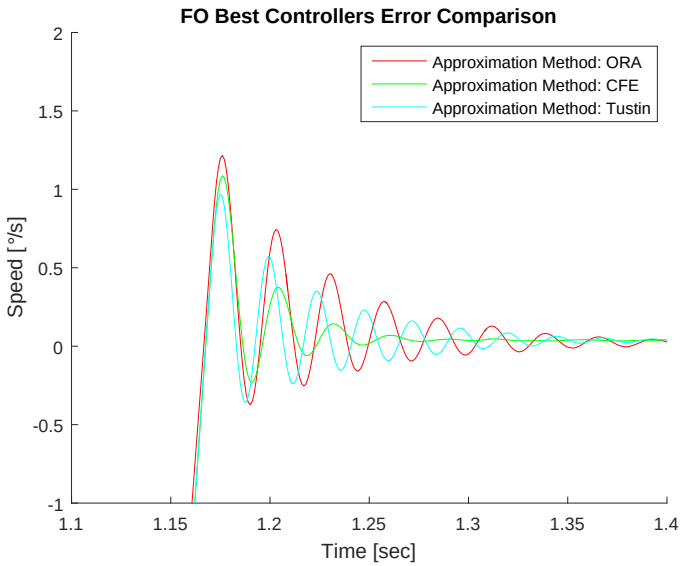

Figure 5.36: Step Responses of Best PI controller Realized by Recursive Tustin Method

The results are similar to the ones obtained for traverse axis. Therefore, for elevation axis, same discretization methods will be used for simulation and hardware tests.

CHAPTER 6

SIMULATION RESULTS AND DISCUSSION

In this chapter, we present the results of the simulation tests which we performed for fractional order controllers and integer order controllers comparison. Before moving into the real system, these simulations are performed. In these simulations, we compare FOC and IOC in terms of their responses in speed loop, their disturbance rejection characteristics and the performance of reference tracking. In section 4.3, We have introduced our plant model using frequency domain system identification. Then, we have covered how to approximate fractional order controller and realize it in our simulink model using the approximation methods covered in chapter 3. After applying approximation methods and realizing fractional order controller, we have introduced the procedure for optimally tuning the controller in chapter 5 and in this chapter, we will use all previously obtained results to compare integer order and fractional order controllers.

In order to validate this comparison, optimal controller design procedure are kept same at both controllers. Both controllers are optimally tuned using same performance index. The used performance index is Integral-Time Square Error and it was chosen in section 5.3.2.1. These controllers are added to the simulink model defined in chapter 5. In this simulink model, we aim to compare speed loop responses, disturbance rejection characteristics and reference tracking accuracy.

Firstly, the closed loop speed responses are compared in terms of step responses and disturbance rejection characteristics. For integer order case, only K_P and K_I are optimized, for the fractional order case, K_p, K_i and λ are optimized. As it was stated in the Chapter 5, the derivative term were causing trouble with

using gyro noise. In these tests, we prefer to run simulations to a model which is the closest to the real system. In section 5.4.1.1, it was explained that the derivative term in the controller when there is a noise in the system was making the system unstable. Similarly, the real system has similar noises sources and we were not using the derivative term on the real systems. Hence, we have discarded the derivative term from the controller, and choose to implement PI^{λ} controller.

Finally, a position loop is added to the model. Reference tracking test is done by applying real aircraft data which is collected on the field to the model. We use it for video tracking tests in our test platform. The errors are compared for both type of FO and IO controllers in section 6.3.

6.1 Speed Loop Comparison and Discussion

In this section, closed loop speed responses are compared. For a given reference signal, optimally tuned controllers are compared in terms of their response.

For fractional order case, as implied in chapter 5, Tustin's Recursive formula is used for realization of fractional order controller. Because, the response of the controller approximated by Tustin's recursive formula was better in terms of maximum overshoot, settling time and rise time performance values. The fractional order controller is realized with approximation order N=5 and it's parameters are found with optimization routine as in section 5.4.2.1 These parameters are repeated in table 6.1.

	FOC	IOC
Speed Loop Kp	0.6395	0.6562
Speed Loop Ki	19.6296	19.998
λ	-0.9799	-1
Fval	0.0612	0.0678

Table 6.1: FO and IO Controllers Used for Comparison in Traverse Axis

In equation 6.1, integer order PI controller is given for the traverse axis,

$$Gc_{IO} = K_p(1 + K_i s^{-1}) = 0.6562(1 + 19.998 s^{-1}) = 0.6562 + 13.1227 s^{-1} \quad (6.1)$$

for the fractional order case, the PI^{λ} controller is given in equation 6.2 as,

$$Gc_{FO} = K_p(1 + K_i s^{\lambda}) = 0.6395(1 + 19.6296 s^{-0.9799}) \tag{6.2}$$

in this controller s^{λ} is realized by Recursive Tustin algorithm. s^{λ} is discretized as below,

$$s^{\lambda} = s^{-0.9799} = 1e^{-3}\frac{0.742z^5 + 0.7034z^4 + 0.2668z^3 + 0.2766z^2 + 0.1334z + 0.14007}{z^5 - 0.9481z^4 + 0.3595z^3 - 0.3728z^2 + 0.1798z - 0.1896} \tag{6.3}$$

The controller parameters for the elevation axis is given as,

	FOC	IOC
Speed Loop Kp	0.4141	0.4111
Speed Loop Ki	14.5674	19.9922
λ	-0.9481	-1
Fval	0.0453	0.0473

Table 6.2: FO and IO Controllers Used for Comparison in Elevation Axis

In the equations 6.4, the integer order PI controller is given for the elevation axis,

$$Gc_{IO} = K_p(1 + K_i s^{-1}) = 0.4111(1 + 19.9922 s^{-1}) = 0.4111 + 8.2188 s^{-1} \tag{6.4}$$

for the fractional order case, the PI^{λ} controller is given in equation 6.5 as,

$$Gc_{FO} = K_p(1 + K_i s^{\lambda}) = 0.4141(1 + 14.5674 s^{-0.9481}) = 0.4141 + 6.0324 s^{-0.9481} \tag{6.5}$$

in this controller, similar to traverse axis, s^{λ} is realized by Recursive Tustin algorithm. s^{λ} is discretized as below,

$$s^{\lambda} = s^{-0.9481} = 1e^{-3}\frac{0.5825z^5 + 0.5708z^4 + 0.2237z^3 + 0.2268z^2 + 0.1119z + 0.1142}{z^5 - 0.9799z^4 + 0.3841z^3 - 0.3894z^2 + 0.192z - 0.196} \tag{6.6}$$

Both controller's parameters are found by optimizing ITSE, Integral Time Square Error, given in chapter 5.4.2. The cost function values named as Fval, illustrates that fractional order controller has performed better than integer order controller as having lower value in the cost function of the optimization procedure. This is the first result we obtain that fractional order controller is better than integer

order controller.

In figure 6.1 and 6.2, the step responses of the plant with integer and fractional order controllers and the error plots are given for traverse axis,

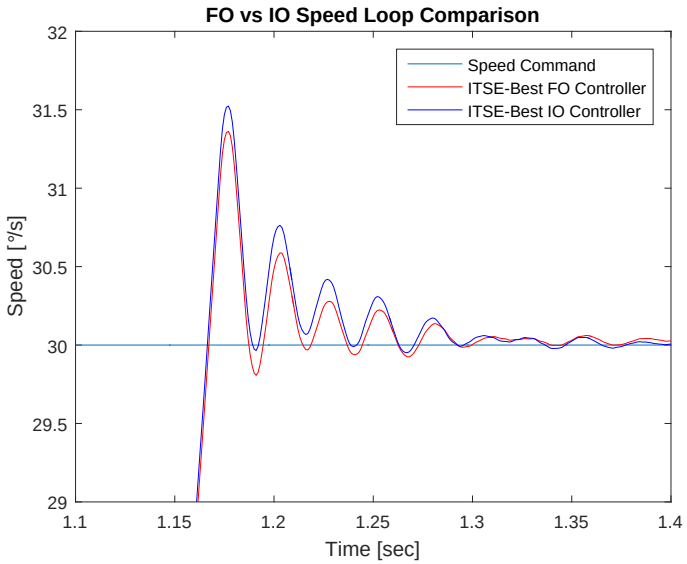

Figure 6.1: Traverse Axis FO and IO Controllers' Step Responses Comparison

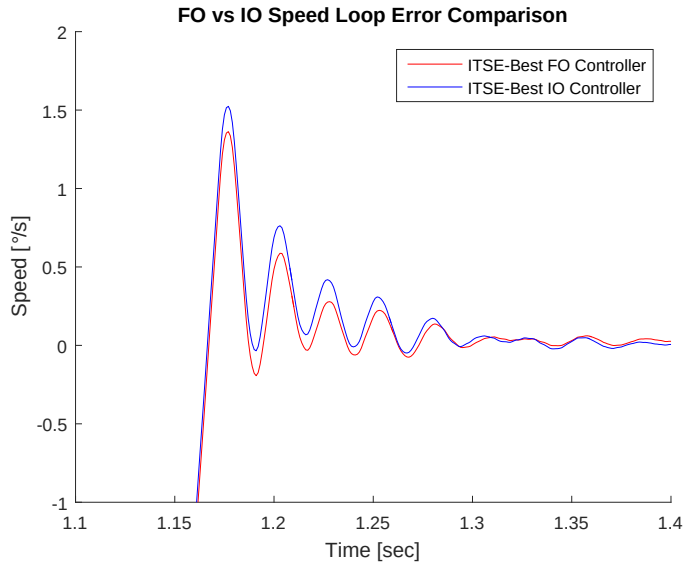

Figure 6.2: Traverse Axis FO and IO Controllers' Errors Comparison

In these figures, the step response with fractional order controller has less overshoot, and faster settling time than the integer order case. The integer order controller has approximately %11.8 more overshoot and settles 4-5 millisecond

later. In figure 6.3 and 6.4, the step responses of the plant with integer and fractional order controllers and the error plots are given for elevation axis,

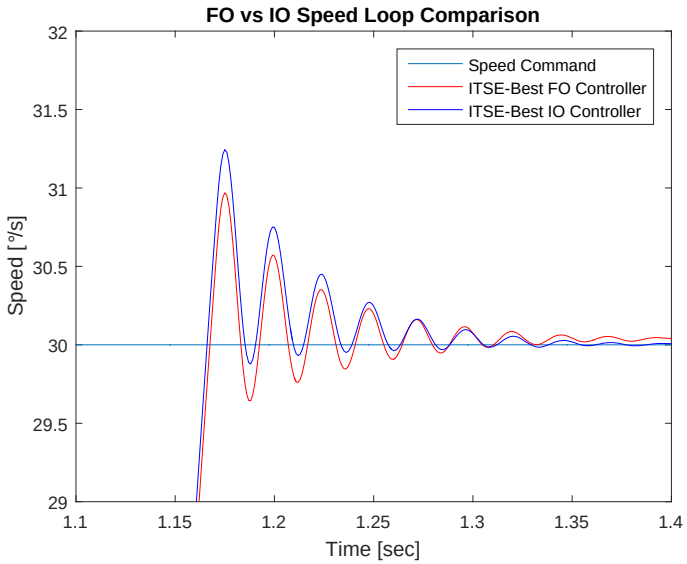

Figure 6.3: Elevation Axis FO and IO Controllers' Step Responses Comparison

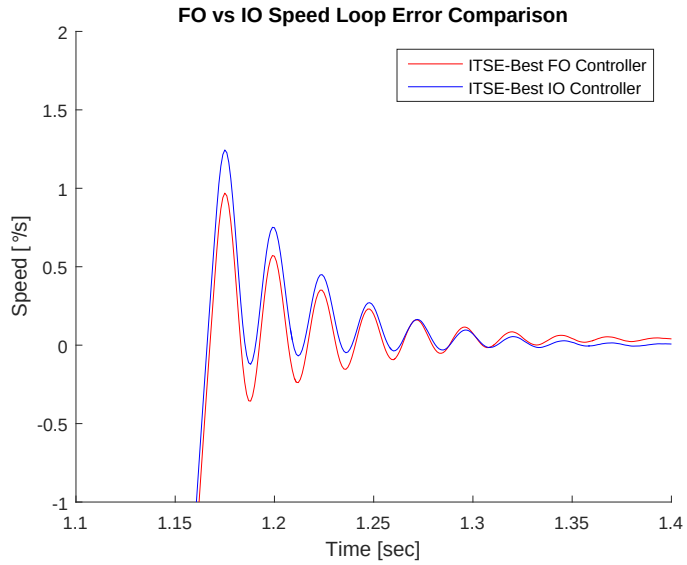

Figure 6.4: Elevation Axis FO and IO Controllers' Errors Comparison

In these figures, when the responses of both controllers are compared, we have obtained that the rising time of the response of controllers are similar, the maximum overshoot is %28.3 bigger for integer order case, and fractional order controller has shorter settling time as 2-3 millisecond. Therefore, depending on

these performance criteria, the fractional order controller has better response than integer order controller.

6.2 Disturbance Rejection Comparison and Discussion

In this part, disturbance rejection characteristics of the designed fractional and integer order controllers are studied. In order to measure the disturbance rejection characteristics the schematic in figure 6.5 is used. In this figure, the speed reference is taken as zero.

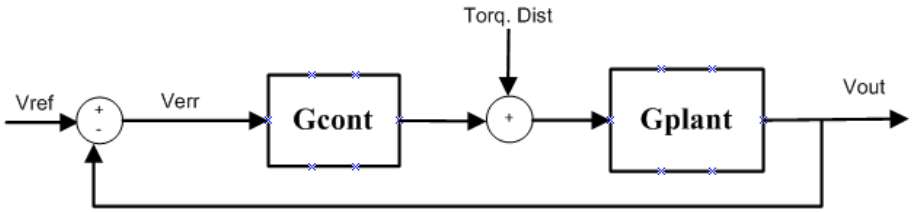

Figure 6.5: Disturbance to the Torque Loop

The transfer function from torque disturbance to the output speed is derived as in equation 6.7,

$$G_{dist} = \frac{G_{plant}}{1 + G_{plant}G_{cont}} \tag{6.7}$$

where G_{cont} is Gc_{FO} and Gc_{IO} in different cases. In figure 6.6 and 6.7, torque disturbance rejection characteristics are given with both fractional order controller and integer order controller for both traverse and elevation axis.

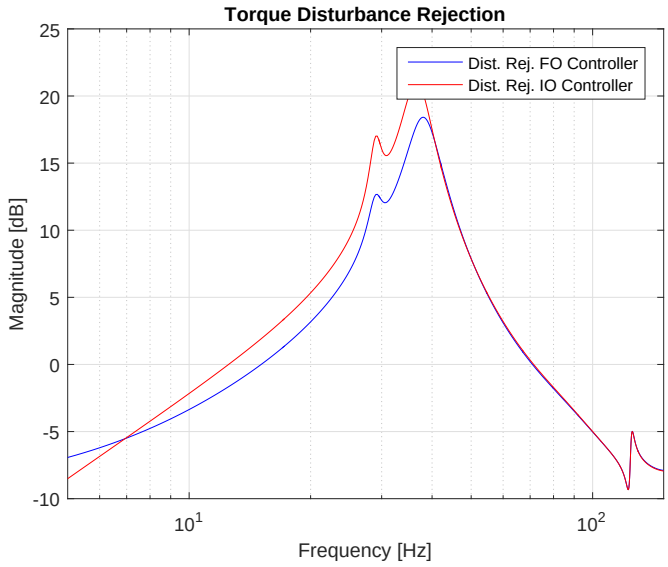

Figure 6.6: Traverse Disturbance Rejection Characteristics Comparisons

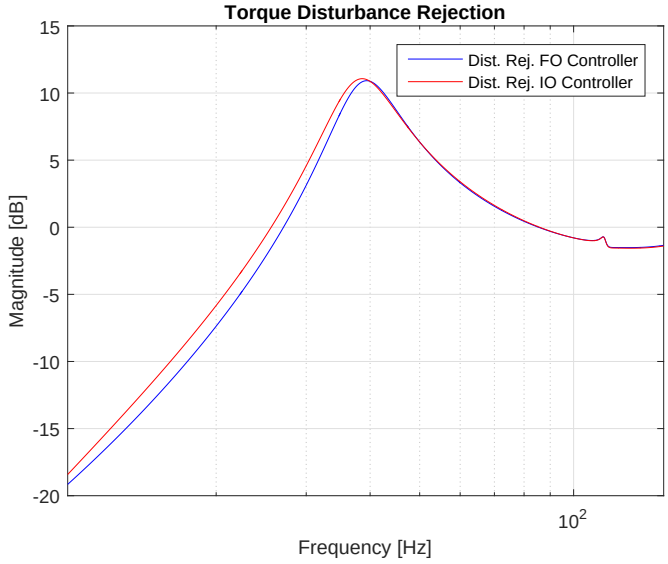

Figure 6.7: Elevation Disturbance Rejection Characteristics Comparisons

Fractional order controller gave better torque disturbance rejection characteristics so that The fractional order controller attenuates the disturbance more than integer order controller in certain range of frequencies. However, in high frequencies above 40 Hz, we observe the similar rejection characteristic of fractional and integer order controllers.

In order to simulate disturbance rejection characteristics in time-domain, sinu-

soidal torque disturbance is added to torque output of the speed controller in simulink model.

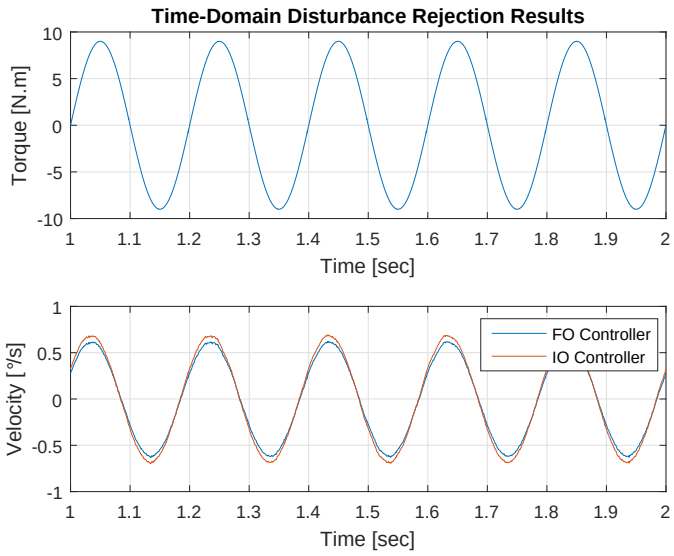

Figure 6.8: Elevation Disturbance Rejection Characteristics Comparisons

Given sinusoidal disturbance and zero reference speed demand, from figure 6.8, the fractional order controller has better performance keeping velocity close to the demand. The figure is given for one second time-lapse, it must be noted that similar periodic response is obtained in rest of the simulation run-time. In higher frequencies, we observe similar performance values. Table 6.3 provides the stabilization accuracy performance values for different frequency values. The values shows that fractional order controller is more successful to perform under disturbances based on lower error values.

Frequency	FOC (mrad)	IOC (mrad)
5Hz	0.266	0.294
10Hz	0.182	0.186
20Hz	0.130	0.134
30Hz	0.109	0.113
40Hz	0.117	0.118

Table 6.3: FO and IO Controllers Used for Comparison in Elevation Axis

In order to simulate disturbance rejection characteristics in elevation axis, disturbance torque is added to the simulink model.

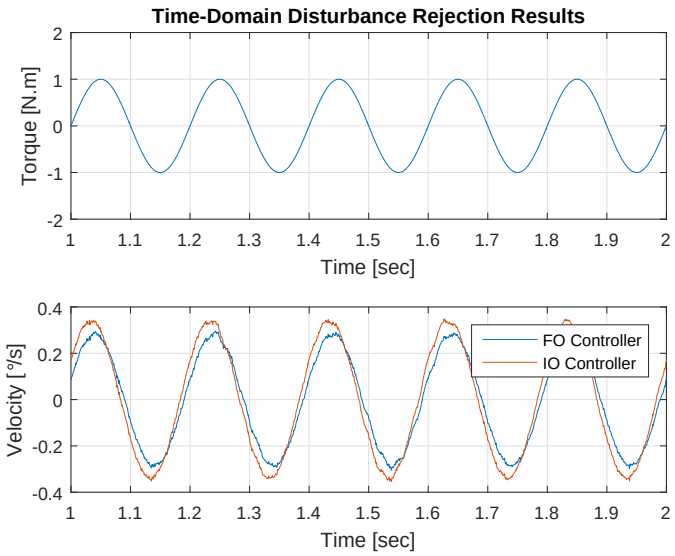

Figure 6.9: Elevation Disturbance Rejection Characteristics Comparisons

Given sinusoidal disturbance and zero reference speed demand, from figure 6.9, the fractional order controller has better performance keeping velocity close to the demand. In higher frequencies, we observe similar performance values. In table 6.4 gives the stabilization accuracy performance values for different frequency values are given. In elevation axis, similar results are obtained as in traverse axis, the fractional order controller has better performance in terms of disturbance rejection obtained with lower error values.

Frequency	FOC (mrad)	IOC (mrad)
5Hz	0.125	0.146
10Hz	0.082	0.091
20Hz	0.056	0.059
30Hz	0.060	0.062
40Hz	0.126	0.125

Table 6.4: FO and IO Controllers Used for Comparison in Elevation Axis

6.3 Reference Tracking Comparison and Discussion

In this part, the reference tracking comparison is made for the integer and fractional order controllers. In order to obtain the simulation model, several improvements and additions are made to the simulink model given in chapter 5. These changes are,

- The position loop is added to the model. The position loop contains the P controller, speed limiter and the wrapper for position input to be sure in it is in between $[-180°, 180°]$.
- The real aircraft data collected from the open field tests is added. Data is recorded with thermal camera, and the target in the camera's field of view is not centralized. Therefore, we will have target position information as it is moving around our field of view.
- The video of the aircraft is projected on the screen in our test setup defined in section 4.1. The angular displacements of the aircraft with respect to our system's position is obtained and saved and it can be considered as the target is moving around our field of view. In the projected video, aircraft moves around in our field of view, the motion of the aircraft in terms of the angular displacement is given in figure 6.10 from our system's perspective.
- The angular displacement of the aircraft is recorded and is the position input to our new simulink model. The traverse and elevation axis movements are separated from each other and simulations are run separately. Both axis have different motors and drivers, so we always consider these axis as independent of each other, and there are no coupling between each other.

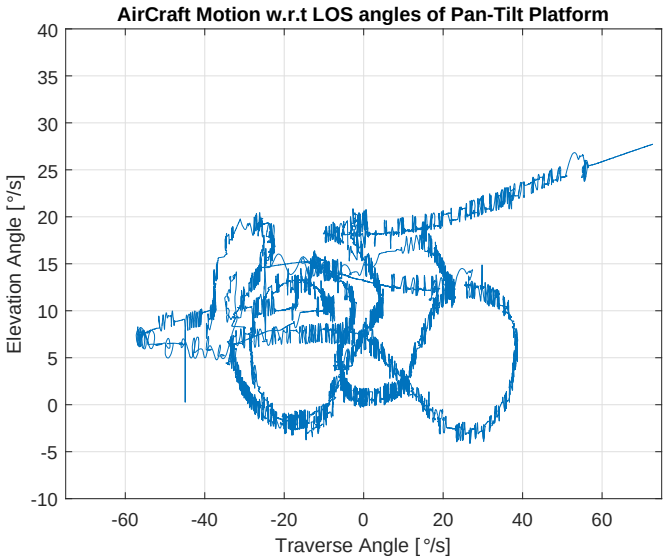

Figure 6.10: Aircraft Motion in terms of LOS angles

In real system we only use the proportional controller for position loops. The reason for using only proportional controller is that the integral term on position loops causes oscillation around the target. High Accuracy Stabilized Gimbal is used for many applications, such as guiding missile, tracking targets including drones, aircrafts. Therefore the pointing accuracy is the most important performance criteria among these applications. Having oscillations around the target completely causes our system to fail in these real applications. The system needs to point the target with highest accuracy as possible. For target tracking applications, the target velocity from tracker is used to track moving targets. Using the velocity information as feed-forward velocity to speed loop makes system able to track targets with less error even if they maneuver frequently. Therefore, the controller used in position loop does not have integral part, and in these simulations we set integral gain to zero. Reference tracking tests are performed by using the simulink model in figure 6.11 which is the implementation of block diagram given in section section 5.1 and figure 5.3,

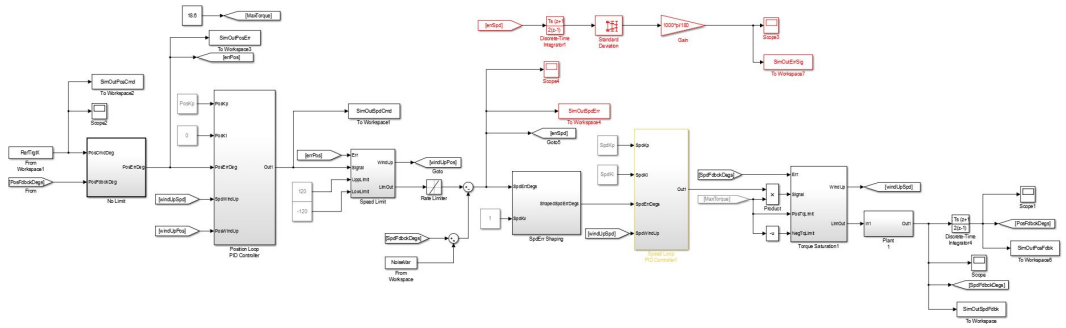

Figure 6.11: Reference Tracking Tests Simulink Model

In figures 6.12 and 6.13 the traverse and elevation axis position demands and the position feedbacks are drawn. In these figures, we can observe the responses of the fractional and integer order controllers to the reference position input.

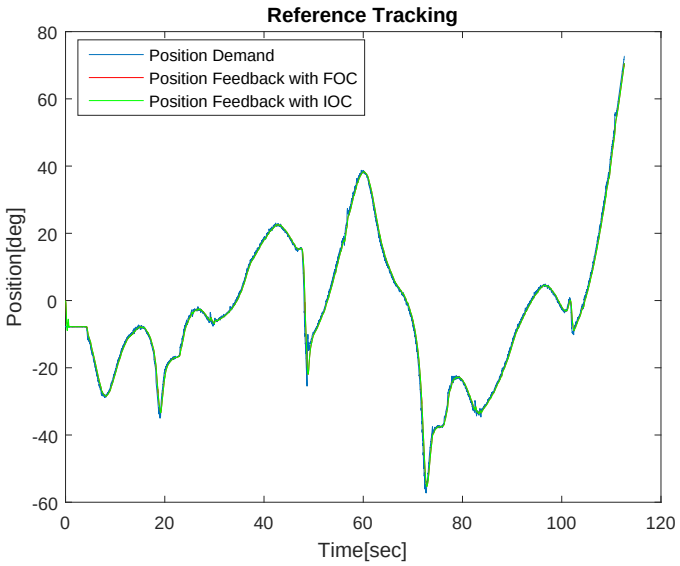

Figure 6.12: Traverse Axis Reference Tracking Position Feedbacks

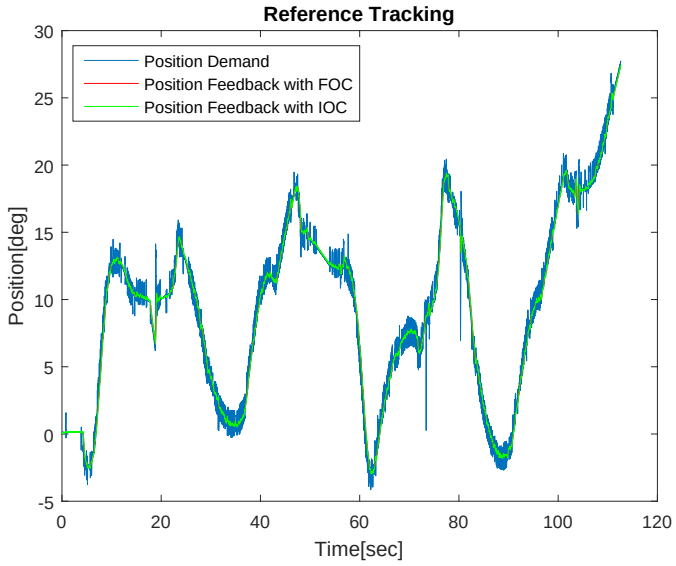

Figure 6.13: Elevation Axis Reference Tracking Position Feedbacks

In figures 6.12-6.13, the integer and fractional order controller responses are very similar. Among the simulation run time, their responses differentiates at the sharp edges on the trajectory. In these parts, we observe fractional order controller illustrates the faster and better response. In figure 6.14 and 6.15 the reference tracking errors are drawn in mrads with 1σ.

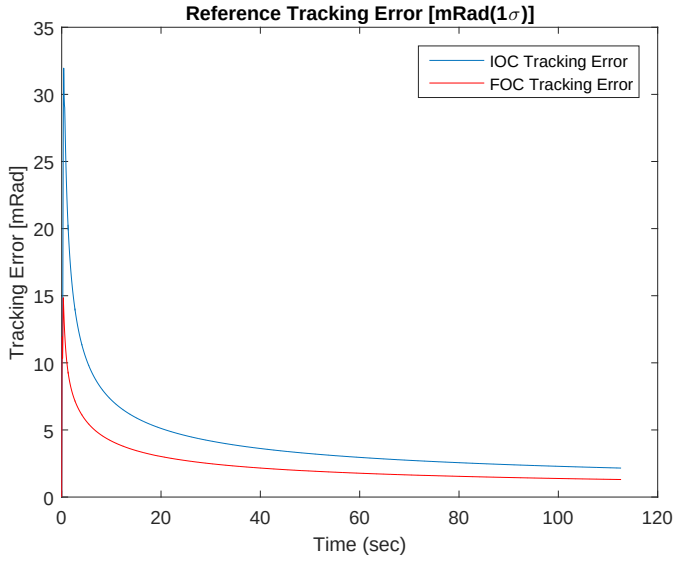

Figure 6.14: Traverse Axis Reference Tracking Tests Simulink Model

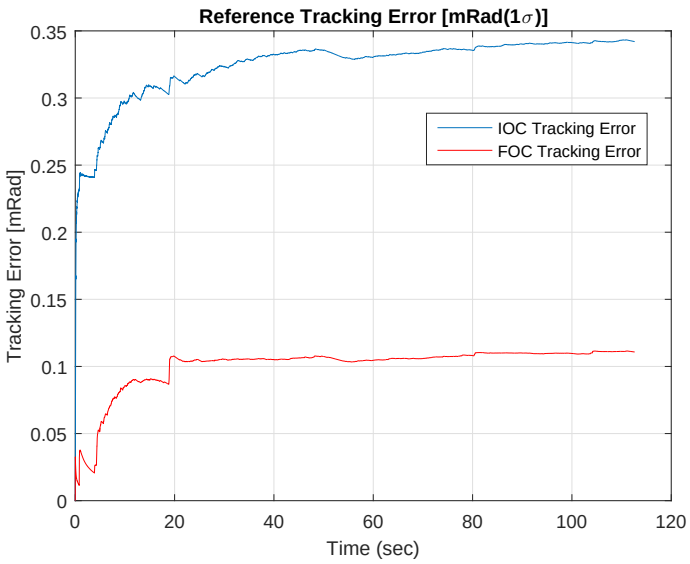

Figure 6.15: Elevation Axis Reference Tracking Tests Simulink Model

Target tracking error levels show that the fractional order controller have less error than the integer order controller. In traverse axis, the target maneuvers more frequently and faster than elevation axis. Therefore, the reference tracking error for traverse axis is measured higher than elevation axis.

In this chapter, we have compared integer order controller with fractional controller in terms of their speed loop responses, stabilization accuracies and target tracking performances. In all of the simulations, we observed that FOC has better response than IOC.

CHAPTER 7

HARDWARE EXPERIMENTAL RESULTS AND DISCUSSION

The comparison of fractional order controller and integer order controller is conducted by performing several simulations on MATLAB in the previous chapter(Chapter 6). In this chapter (Chapter 7), FOC and IOC will be compared on the real system. High-Accuracy Stabilized Gimbal of ASELSAN will be used in hardware experimental studies. The detailed information about the system were given in section 4.2.

In chapter 5, the approximation methods of the fractional differential operators were compared, and we decided to use Tustin's recursive formula for the comparison tests because of its better performance in step response to velocity demand. In chapter 6, same method is used for FOC implementation and all results illustrated that FOC is better than IOC in terms of speed loop responses, disturbance rejection characteristics and reference tracking. We will implement FOC using Tustin's Recursive Formula as approximation method in our hardware tests as well.

In real system tests, after Fractional order controller is implemented on DSP module, the previously found best controllers in chapter 5 are tried first. However, parameters found on simulation via optimization routines could not manage to control the real system. When tried, system has gone unstable with these parameters, and brought system to resonance. The reasons that optimized parameters on simulation could not control to the real system are as follows;

- Identified system model on simulink is not exactly same as with real sys-

tem. There are some unmodeled variables of the system.

- Some of the nonlinearities in the real system are not modeled in simulation. These are friction, backlash between motor and encoder,coupling with other axis, and quantization errors etc.
  Modeling these nonlinearities requires different efforts and would be studied as future work. In order to stay in the scope of the thesis, the simulation tests are performed without these parts.

- Step responses with the optimized parameters are oscillatory. In order to minimize the error, the optimization procedure finds the oscillastory behaviors as better solution. These oscillations mean that system would vibrate and make noise in real system, so these parameters are not acceptable. Even smallest vibrations on the system is not acceptable due to video tracking tasks. When electro-optics are in minimum FOV, these vibrations are observable on the screen, and it causes blur on the screen and makes impossible perform video tracking performance.

Therefore, the hardware tests presented in this chapter will have different controller parameters than the simulations. In these tests, the FOC and the IOC are tuned manually online, by adjusting gains and observing resulting performances.
In following sections, the speed loop responses of the real system with FOC and IOC, the stabilization performances and target tracking performances are presented with hardware test results.

7.1 Speed Loop Comparison and Discussion

In this section, we compare step responses of the controllers with integer order and fractional order of integration. For the fractional order case, the integration order must be determined by ourself, as there are not any well defined tuning strategy for the tuning of integration order. In order to choose the integration order we have selected several values, and tried to tune system, which results are then obtained.

- The first value of the integration order is 0.5. In table 7.1, the tuned speed loop PI^{λ} controller parameters are given.

	Traverse Axis FOC	Elevation Axis FOC
Speed Loop Kp	0.09	0.12
Speed Loop Ki	15	10
λ	-0.5	-0.5

Table 7.1: FO Controller Parameters for Integration Order $\lambda = -0.5$

In figures 7.1 and 7.2, the step responses of the controller are given for traverse and elevation axis.

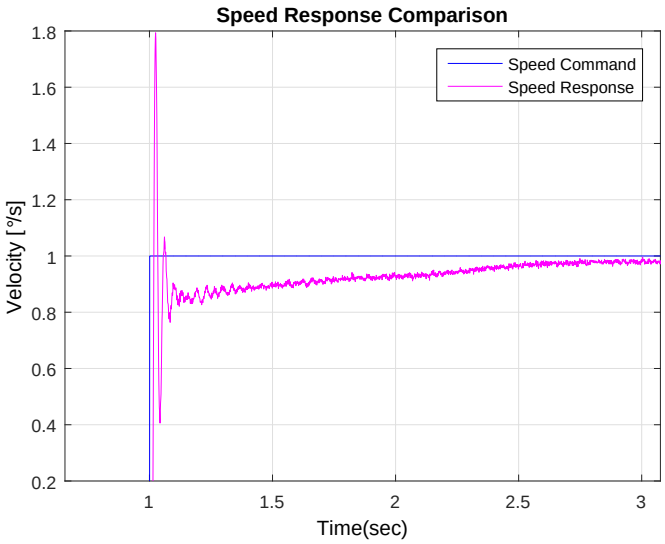

Figure 7.1: Traverse Axis FOC Response with $\lambda = -0.5$

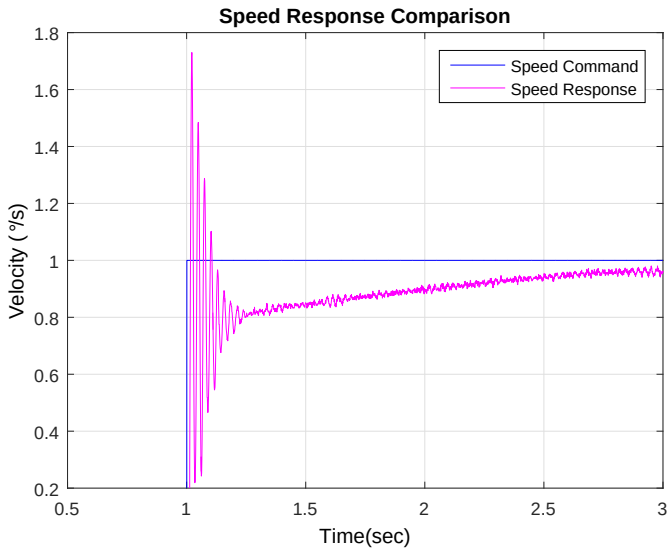

Figure 7.2: Elevation Axis FOC Response with $\lambda = -0.5$

For $\lambda = -0.5$, the integration action is not enough, the controller can not eliminate error fast enough, and the rising time is considerably high. When the integral gain is increased, the response gets unstable due to proportional effect of the fractional integration. Next, the integration order will be increased.

- The value of integration order is selected as 0.8. In table 7.2, the controller parameters are given for both axis.

	Traverse Axis FOC	Elevation Axis FOC
Speed Loop Kp	0.1	0.14
Speed Loop Ki	60	30
λ	-0.8	-0.8

Table 7.2: FO Controller Parameters for Integration Order $\lambda = -0.8$

In figures 7.3 and 7.4, the step responses of the controller are given for traverse and elevation axis.

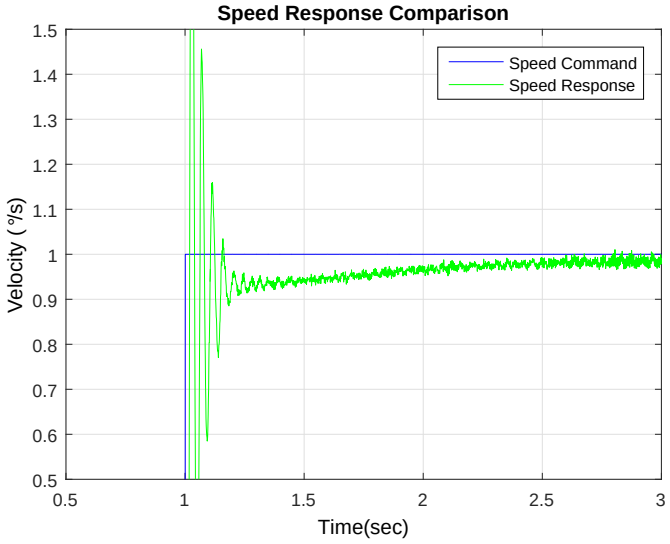

Figure 7.3: Traverse Axis FOC Response with $\lambda = -0.8$

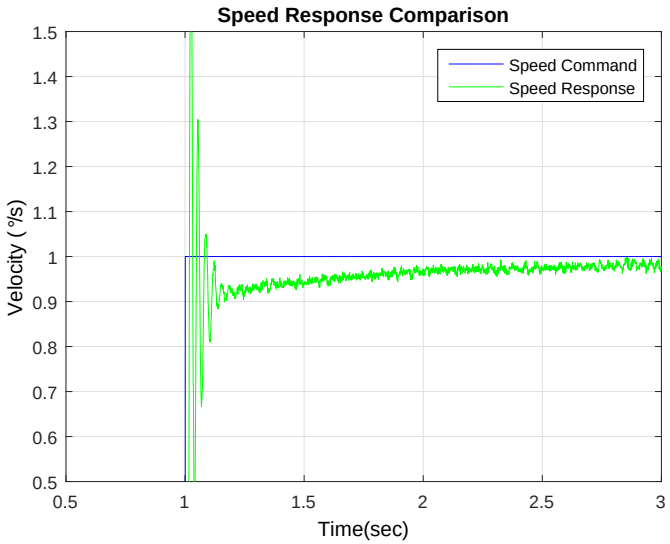

Figure 7.4: Elevation Axis FOC Response with $\lambda = -0.8$

For $\lambda = -0.8$, the integration action has become more effective, and this can be seen on the results. The settling time decreases and oscillations increase compared to order of $\lambda = -0.5$. However, the responses are still lack of integral action, the responses should not have high settling time. In next test, the integration order in increased and chosen as 0.9.

- For this test, the value of integration order is selected as 0.9. In table 7.3, the controller parameters are given for both axis.

	Traverse Axis FOC	Elevation Axis FOC
Speed Loop Kp	0.11	0.18
Speed Loop Ki	50	40
λ	-0.9	-0.9

Table 7.3: FO Controller Parameters for Integration Order $\lambda = -0.9$

In figures 7.5 and 7.6, the step responses of the controller are given for traverse and elevation axis.

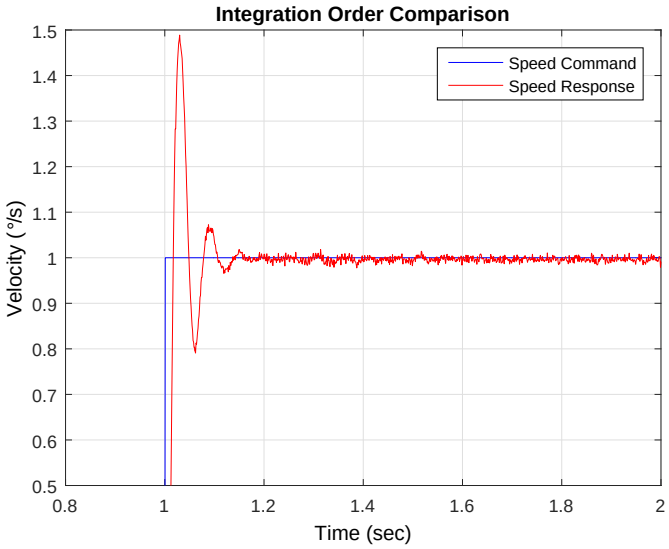

Figure 7.5: Traverse Axis FOC Response with $\lambda = -0.9$

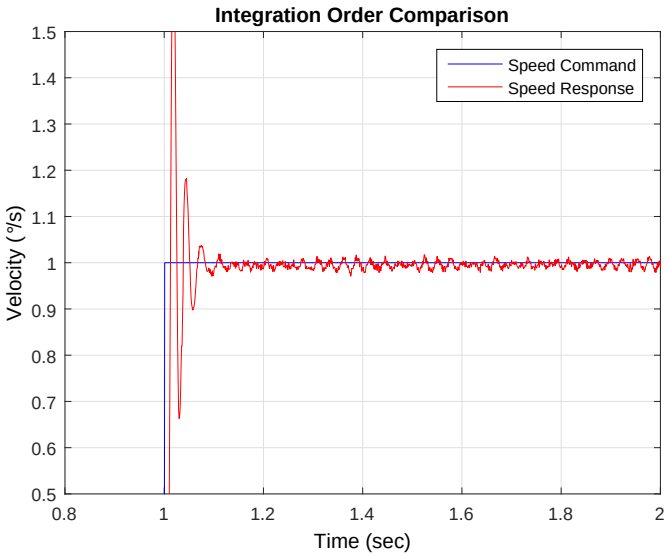

Figure 7.6: Elevation Axis FOC Response with $\lambda = -0.9$

These responses illustrates that higher integration order has faster response and shorter settling time compared to lower integration order. The lower integration order does not able to meet the requirements of the system towards higher performances.

In figure 7.7, the all responses are given together to see the effect of integration order in real system.

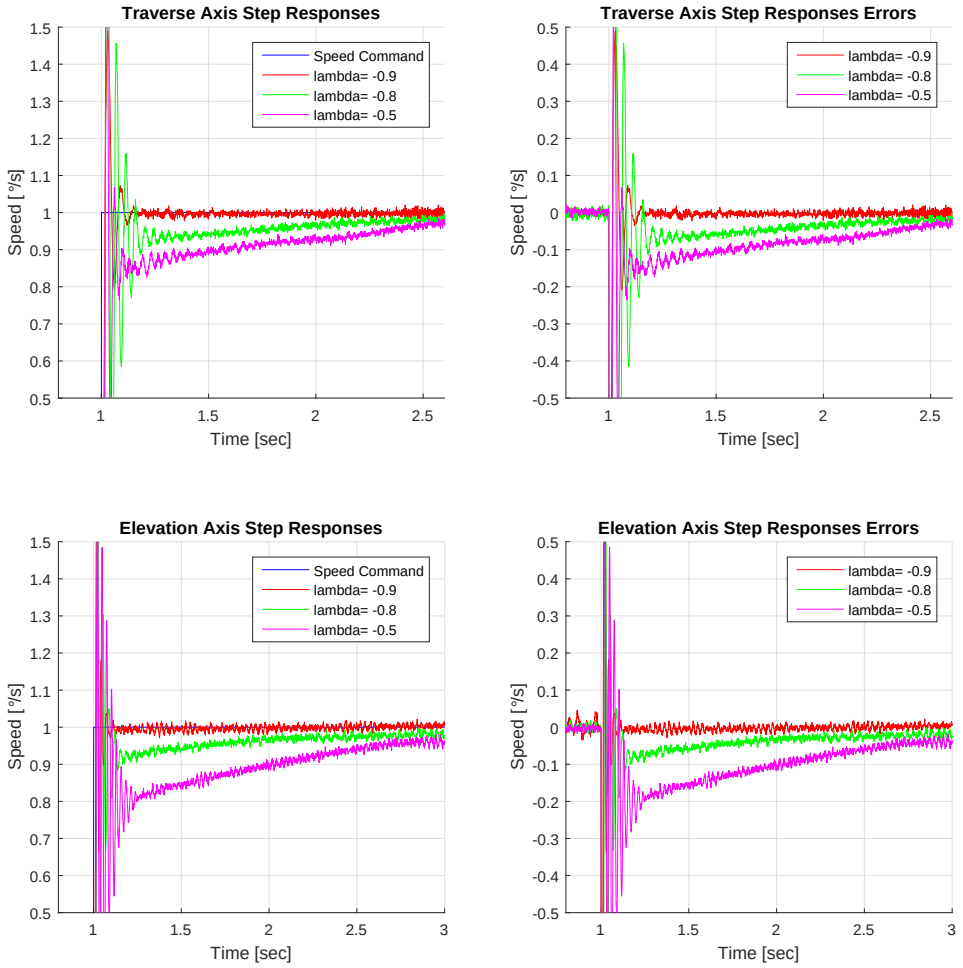

Figure 7.7: Integration order comparisons

In this figure, as integration order increases, the performance of controller gets faster, the steady state error decreases and the settling time also decreases. So, it is observed that system need high order of integration to have faster and steady responses.

In hardware tests, we have to choose the integration order as there is no systematic way to tune it. Besides, both in optimization simulations made in chapter 5 and the first hardware tests, we found that, the higher integration order gives the better result. In the integer order and fractional order controller comparison, we select the integration order as 0.9. The optimally tuning studies also resulted in similar range for integration order as $-0.8 > \lambda > -1$.

After the integration order is fixed, we performed tuning studies manually and find controller parameters for both fractional and integer order controller. In table 7.4, integer order PI controller is given for the traverse axis.

	FOC	IOC
Speed Loop Kp	0.11	0.12
Speed Loop Ki	50	5
λ	-0.9	-1

Table 7.4: FO and IO Controllers Used for Comparison in Elevation Axis

$$Gc_{IO} = K_p(1 + K_i s^{-1}) = 0.12(1 + 5s^{-1}) \tag{7.1}$$

for the fractional order case, the PI^{λ} controller is given in equation 6.2 as,

$$Gc_{FO} = K_p(1 + K_i s^{\lambda}) = 0.11(1 + 50s^{-0.9}) \tag{7.2}$$

The controllers used in elevation axis speed loop comparison tests are given in table 7.5:

	FOC	IOC
Speed Loop Kp	0.18	0.13
Speed Loop Ki	40	30
λ	-0.9	-1

Table 7.5: FO and IO Controllers Used for Comparison in Elevation Axis

In the equations 6.3, integer order PI controller is given for the elevation axis,

$$Gc_{IO} = K_p(1 + K_i s^{-1}) = 0.13(1 + 30s^{-1}) \tag{7.3}$$

for the fractional order case, the PI^{λ} controller is given in equation 6.4 as,

$$Gc_{FO} = K_p(1 + K_i s^{\lambda}) = 0.18(1 + 40^{-0.9}) \tag{7.4}$$

In figures 7.8 and 7.9 step responses of the plant with integer and fractional order controllers and the error plots are given for traverse axis for the high speed demand;

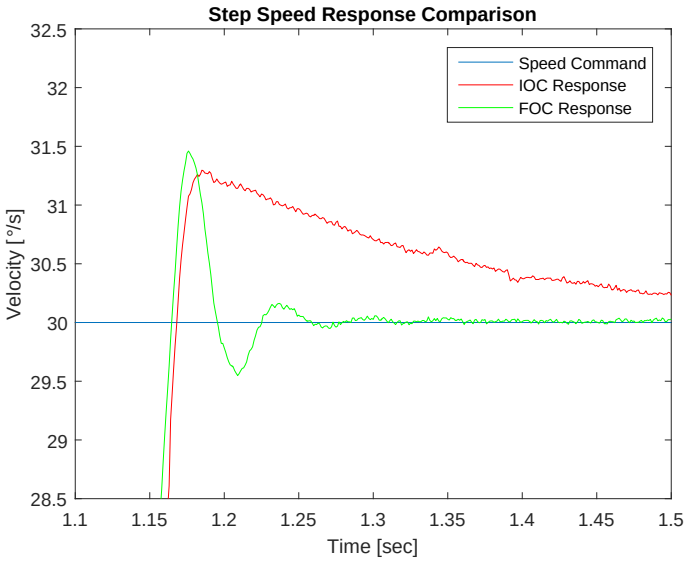

Figure 7.8: Traverse Axis FO and IO Controllers' Step Responses Comparison

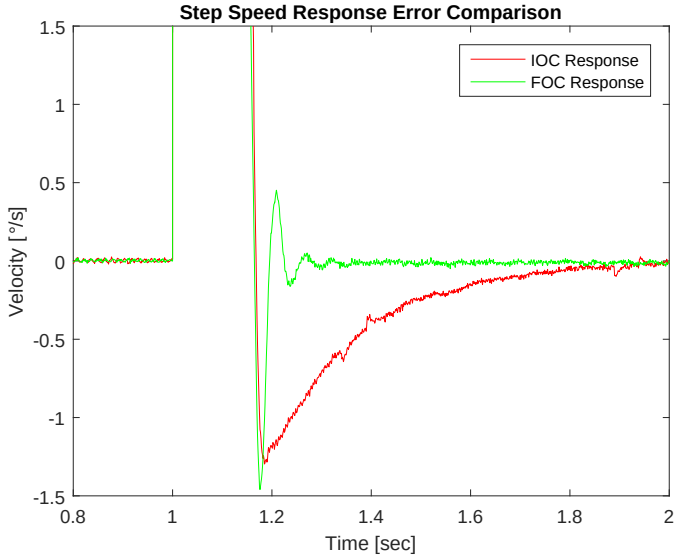

Figure 7.9: Traverse Axis FO and IO Controllers' Errors Comparison

In these responses, the integer order controller fails to respond and eliminate error faster. The fractional order controller has very shorter settling time, and faster response compared to integer order controller. The stabilized visual tracking systems mostly operate in lower speeds. Therefore, in the figure 7.10 and 7.11, we compare same controller responses with lower velocity demands.

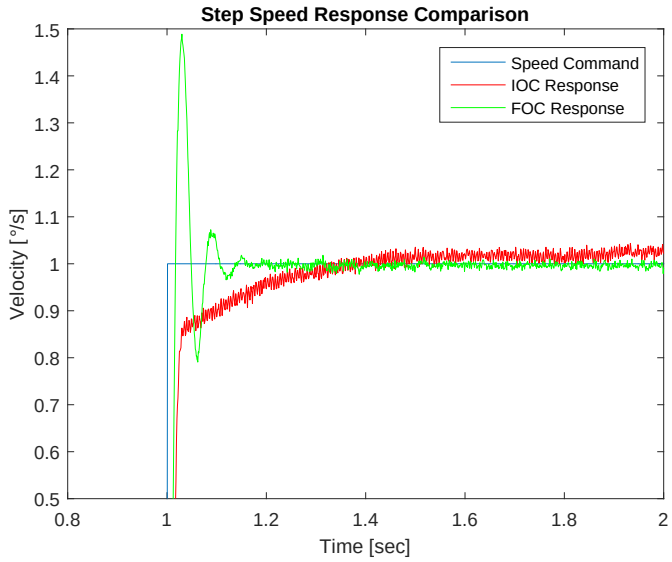

Figure 7.10: Traverse Axis FO and IO Controllers' Step Responses Comparison

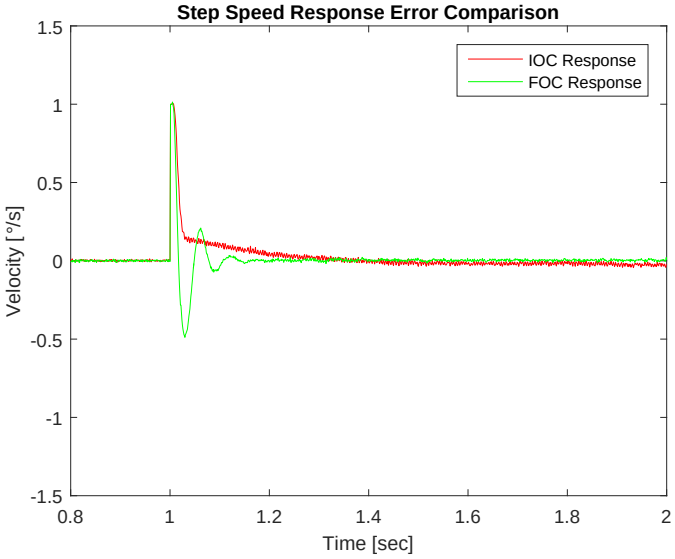

Figure 7.11: Traverse Axis FO and IO Controllers' Errors Comparison

In these figures, the fractional order controller has faster response, its rising time shorter than integer order controller. Similarly, the settling time is also shorter for fractional order controller. However, it is observed that, the fractional order controller has overshoot compared to integer order controller. The fractional order integration also adds to the proportional response. As covered in the section 5.3.1.3, the proportional gain effect of fractional integration gets lower

as order increases. However, in these tests, we observe the beneficial effects of the fractional order controllers.

In figures 7.12 and 7.13 step responses of the plant with integer and fractional order controllers and the error plots are given for elevation axis for the high speed demand;

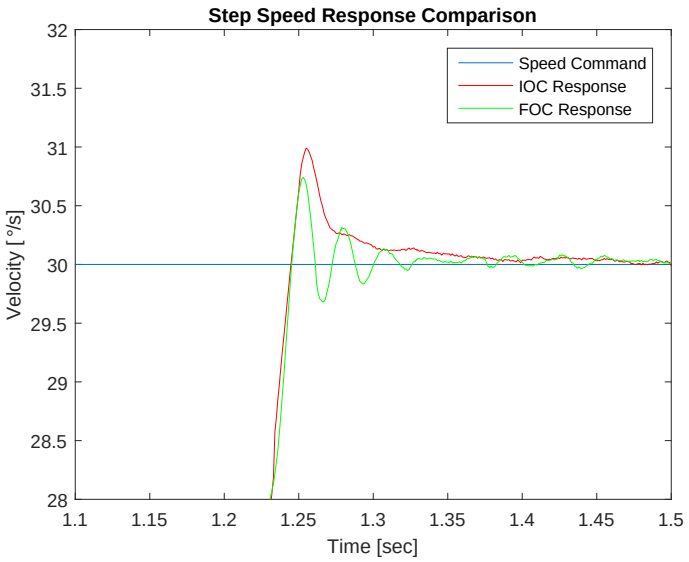

Figure 7.12: Elevation Axis FO and IO Controllers' Step Responses Comparison

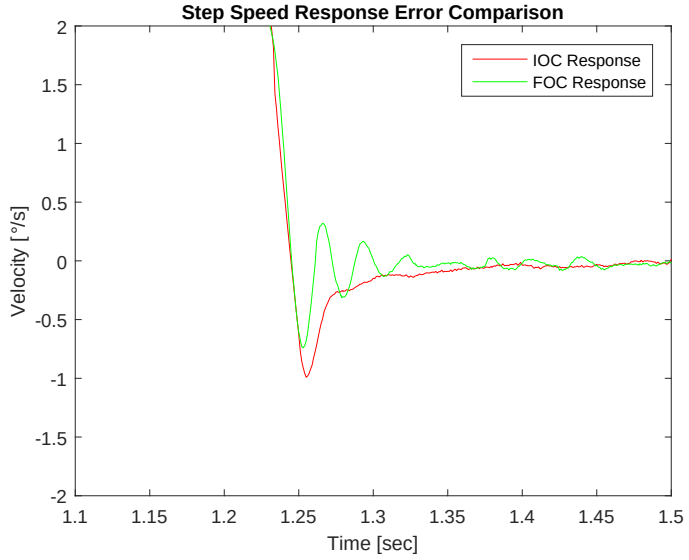

Figure 7.13: Elevation Axis FO and IO Controllers' Errors Comparison

In higher step demands, the fractional order controller is better in terms of settling time, and steady state error. The rising time and maximum overshoot values are similar. In lower velocity demands, we observe that in terms of rising time, steady state error, and settling time, the FOC is much more better than IOC case.

In figures 7.14 and 7.15 step responses of the plant for the low speed demands are given;

Figure 7.14: Elevation Axis FO and IO Controllers' Step Responses Comparison

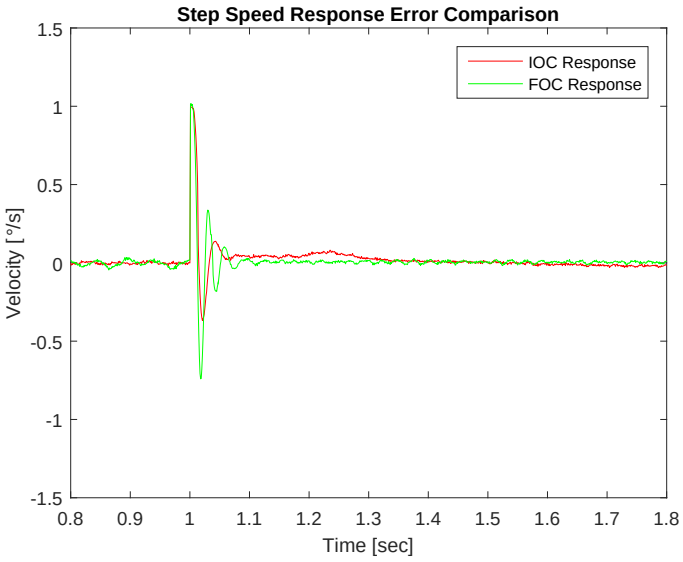

Figure 7.15: Elevation Axis FO and IO Controllers' Errors Comparison

For lower speed step demands, we observe that the fractional order controller has lesser steady state error, faster settling time. It is able to follow low speed demands better than the integer order controller. The most of the surveillance systems, guidance devices requires smooth responses for lower velocities. Therefore, the result we obtained from these hardware tests is important. These results will play crucial effect on stabilization accuracy performances. These kind of systems requires faster response to demands and better disturbance rejection characteristics. Having better speed loop controller will make us able to improve stabilization accuracy and reference tracking performances. These tests are performed in following sections.

7.2 Stabilization Accuracy Comparison and Discussion

In the test platform, as described in the Chapter 4, we have 6-DOF Stewart platform. In the tests given in this section, we will use stewart platform to measure stabilization accuracy of the system separately with FOC and IOC.
We measure both traverse and elevation axis stabilization performances by applying roll, yaw and pitch disturbances all together to the system. Our system, High-Accuracy Stabilized Gimbal is two-axis gimbal, so , yaw and pitch disturbances will be eliminated and system will be stable in these two axis.

In figure 7.16, the applied disturbances are given,

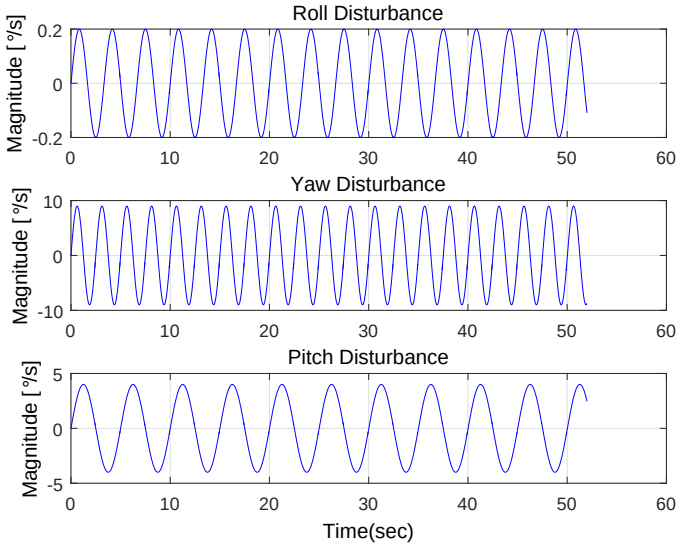

Figure 7.16: Disturbances applied to the system

In table 7.6, the properties of disturbances are given,

	Frequency (Hz)	Magnitude ($\circ/s$)
Roll	0.3	0.2
Yaw	0.4	9
Pitch	0.3	4

Table 7.6: The properties of Disturbances in sine waveform

In figure 7.17 the results of the test are given for traverse axis,

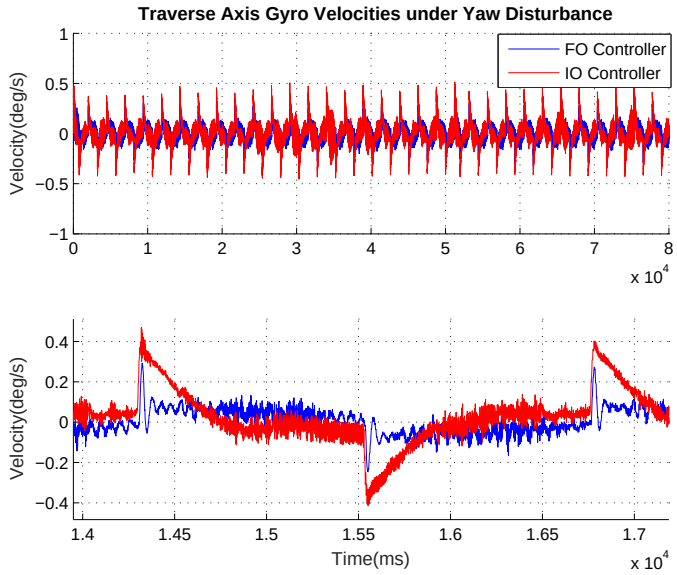

Figure 7.17: Traverse Axis FO and IO Controllers' Errors Comparison

For the elevation axis, the results of the second test is given in figure 7.18,

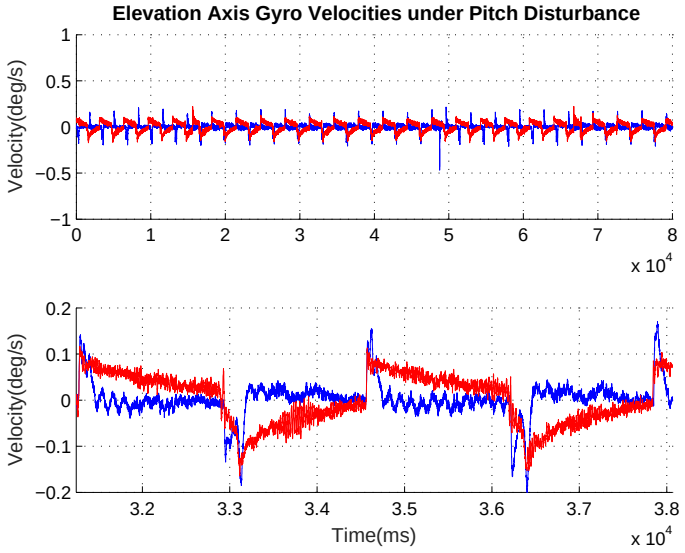

Figure 7.18: Elevation Axis FO and IO Controllers' Errors Comparison

In these figures, upper and lower plots represents same gyro speed. In order to analyze response in smallar time scale, lower plots are added. We observe that the velocity measured from the gyroscope on the system was closer to zero in fractional order controller case. In order words,FOC was able to stabilize system

better than the IOC case. In table 7.7, the measured stabilization accuracies are given; For the stabilization tests we can observe that the FOC outperforms

Given Disturbance	FOC (mrad)	IOC (mrad)
Yaw(Traverse Axis Accuracy)	0.402	0.728
Pitch(Elevation Axis Accuracy)	0.159	0.473
All(Traverse Axis Accuracy)	0.442	0.843
All(Elevation Axis Accuracy)	0.147	0.434

Table 7.7: Measured Stabilization accuracies of the system

the IOC. These accuracies are measured by using following procedure;

- The gyroscope drift comes from its nature, so the drift of the gyroscope must be neutralized. In order to measure accurately, we detrend the measured gyroscope velocity to remove drift on it.
- Then, the integral of the gyroscope velocity must be taken and it's standard deviation must be calculated.
- Finally, the obtained result is converted into radians.

7.3 Target Tracking Comparison and Discussion

In this section, the target tracking tests are performed in two way. In first one, the Stewart platform was turned off, and tests are performed under no disturbance. The other tests was performed under the same disturbances we used in previous section.
In target tracking tests, we projected the real aircraft video to the curved screen of which our system stays in the center. As described in chapter 4, the video tracking unit calculates the tracking errors then, sends these information to the servo unit. The projected video was taken from the real aerial target and recorded on the field. Therefore, for detection and tracking algorithm, the ones for the air targets were chosen. In our system, there are many tracking and detection algorithms installed, for the best performance, the algortihm must be chosen for the right target.

The measured tracking errors are illustrated in figures 7.19 and 7.20. It must be observed that, the resolution of the errors generated by algorithm used in tracker unit and the tracking accuracy measured from these data are higher than the stabilization tests.

Figure 7.19: Video Tracking Errors with FOC under disturbance (Upper plot: Elevation Axis, Lower plot: Traverse axis)

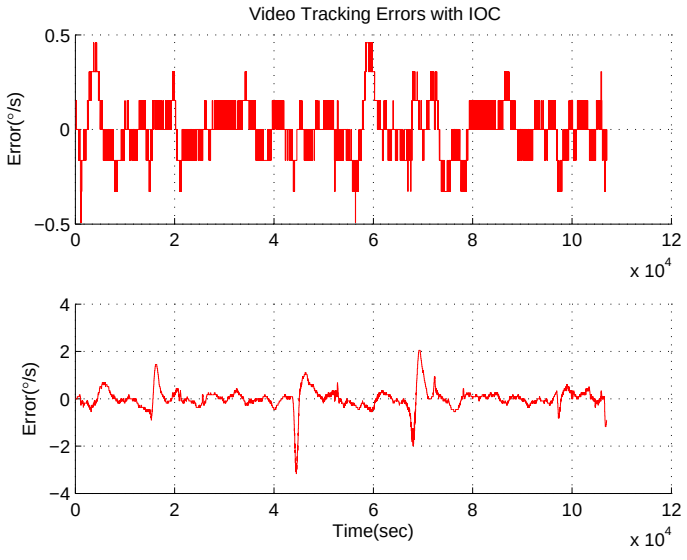

Figure 7.20: Video Tracking Errors with IOC under disturbance (Upper plot: Elevation Axis, Lower plot: Traverse axis)

The reason for having such bigger step size for position errors in video tracking is directly related with field of view of the camera we have used. As it is explained in section 4.3, and in figure 4.3, HASG is directly mounted on a center of the curved screen. In order to track the aerial target on this screen, we have worked with maximum field of view. We also know that our camera is giving outputs in 720x288p. In figure 7.20, the error degrees in steps corresponds to one pixel so we easily claim that we are able to track the target with one pixel error. One pixel corresponding to angle errors in these figures is directly related to camera's field view, and it illustrates that, we are able to track the target within 2-3 pixel range and generally around one pixel range.
In the table 7.8 the measurement of tracking accuracies are given,

	FOC (mrad)	IOC (mrad)
Traverse(With Disturbance)	7.029	7.536
Elevation(With Disturbance)	2.238	2.527
Traverse(Without Disturbance)	7.046	7.089
Elevation(Without Disturbance)	1.945	2.288

Table 7.8: Measured tracking accuracies of the system

Another reason for these accuracy tests to have high values is that the tracker unit does not provide the servo unit with the specified target speed. Therefore, in sharp movements of the aircraft, error increases and the servo undergoes delays, falling behind the tracker.
In video tracking tests, the results were not clear as the stabilizations tests in showing that FOC outperforms IOC. In order to execute the video tracking, we implement the position loop, and as it is pointed out in previous chapter, we always use proportional term only in position loop controller. Therefore, in this part of chapter, we actually compared the results of $P-PI$ with $P-PI^{\lambda}$ controller. This is the reason for us to get similar results for IOC and FOC cases. However, numerical results still shows that the FOC outperforms IOC.

7.4 Adding Fractional Order Derivative to the Controller

In section 5.3, while demonstrating the effects of the parameters in fractional order $PI^{\lambda}D^{\mu}$ controller, it was stated that due the higher level of noise in the model, the use of fractional order derivative was not possible. The responses obtained from controllers with fractional order derivative was unstable, and showing high frequency oscillatory behaviors on steady state. In this section, adding derivate term to real system controller and its effects are demonstrated and discussed.

After illustrating fractional order PI^{λ} controller outperforms integer order PI controller, we decided to add fractional order derivative term to the controller which will be in form $PI^{\lambda}D^{\mu}$.

In order to decide the order of derivation, we first implemented $PI^{\lambda}D^{\mu}$ with $\mu = 1$. Similar to the results in section 5.3, we have obtained unstable response with very low K_d value such as 0.001. In figure 7.21, the response of the system in traverse axis is given,

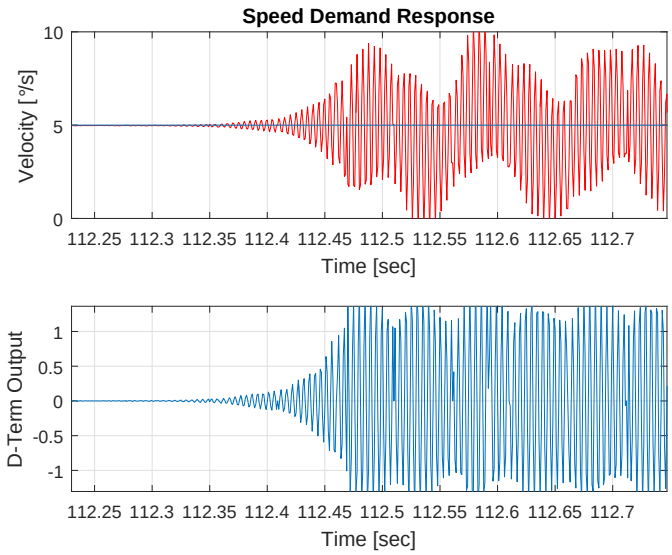

Figure 7.21: Upper plot: Speed demand and Speed response, Lower plot: Derivative action's output in controller

The derivative output in the controller gets bigger with time and causes the system to unstable even with such small K_d value. Higher order derivation is

not desired in our system as pointed out in the table 7.9, we were not even able to increase the derivative gain. Thus, we changed the order of derivation to 0.2 and 0.1. In each of these cases, we were able to increase K_d more than the higher order case. However, at lower derivative gain, the response was not showing the effect of adding the derivative term similar to that of zero derivative. When we increase K_d more than 0.5, the results were similar to figure 7.21.
In table 7.9, the derivative order and K_d values are given.
After setting K_d to given values and letting the system step response stabilize

μ	K_dt
1	0.001
0.2	0.2
0.1	0.5

Table 7.9: K_d values for different μ

with zero velocity demand the following figure 7.22 is obtained. In figure 7.22, the systems response under the disturbance is given,

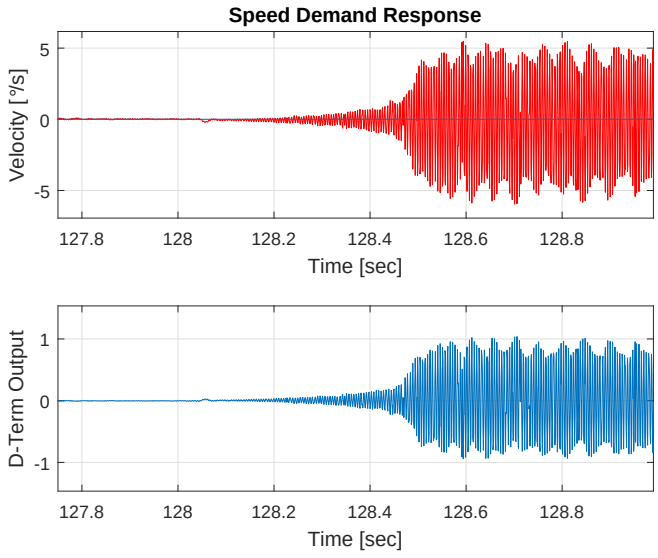

Figure 7.22: Upper plot: Speed response, Lower plot: Derivative action's output in controller

The disturbance used here is the same used in section 7.2, and its magnitude and frequency values are given in table 7.6. These experimental studies illus-

trated that, we are not able to use integer and fractional order derivatives in our system. Although we were able to increase the gains for fractional order derivation more than integer order case, beyond some point they caused system to go unstable. And this point, their effects on the response is not observable. When derivation outputs exceeds certain level where we observe the effect of derivation on response, system goes unstable which can be seen on lower plot in figure 7.22. Sensor noises, time delays, quantization errors and all other noises in the system prevent us from using both integer and fractional order derivation. The closed loop speed controller will stay in the from of PI and PI^{λ}.

CHAPTER 8

CONCLUSION AND FUTURE WORK

8.1 Conclusion

The main objective of this thesis is to improve the servo performances of the stabilized visual tracking system. The stabilization accuracy and the target tracking performance are the our two main criteria of evaluating the performance of the servo system. These systems are used for target detection and tracking, missile guidance, master-slave applications with weapon platforms etc. The success on these tasks depend on the performance of the system, stabilization accuracy and the target tracking performance. Regardless of the environment and the vehicle or platform system mounted, these performances must be achieved at certain level against external disturbances.

In order to obtain high performance in these systems, mechanics, hardware, controllers, electro optics, all these elements must satisfy strict requirements. In our study, we change the controller structure in ASELSAN's High-Accuracy Stabilized Gimbal (HASG) with proposed controller. The proposed new controller must provide better servo performances such as better disturbance rejection characteristics, faster response to references, shorter settling time. The controller is implemented on DSP module in the servo controller is a new type of PID controllers which is Fractional Order $PI^{\lambda}D^{\mu}$ Controller. The mathematics of this controller comes from the fractional differential equations where the order of the integration and the derivation could be fractional.

The frequency domain system identification is performed in real system in order to obtain Simulink model of the plant which will be used in simulations. In these simulations, early fractional order implementations are done, their tuning methods are studied. In these tuning studies, the optimization methods are compared, their results are discussed. All of these studies are done for different fractional order controllers approximated with different methods. The reason to compare approximation methods, optimization methods in these simulations to acquire the best way implement fractional order controller in real system.

The implementation of fractional order controller to DSP module was not a straightforward task. The implementation studies showed that realization order of the integration and derivation plays crucial on realizing the controller. Some methods failed with higher approximation orders. Therefore, approximation and realization method must be chosen with proper order.

After we obtain the implementation methods for approximation and optimally tuning method of controller, we proceed to the real system, and implemented same controllers. The only difference in real system is, we have to tune the system by hand. The controller parameters found by optimization procedure in simulation were not able to stabilize the system and work properly. Because of the other noises which are not included in the model, the controller parameters found by optimization were comparatively high to the real system parameters. In real system comparisons, we tuned the integer and fractional order controllers by hand, and performed several tests.

The comparison tests in both simulation and hardware illustrated that fractional order controller outperforms integer order controller in all aspects. As fractional order controller have two more tuning node, the tuning studies for $PI^{\lambda}D^{\mu}$ was harder than the integer order case. The order of the fractional integration and derivation must be chosen depending on application. In our application, our

system was linear-like and both optimal controllers and hand-tuned controllers had integration order close to first order integration.

The controller used in our system for integer order case would not accept any derivation due to noises in our system. Moreover, we were not able to increase the derivation gain. However, when fractional order derivation is added to the controller, it is observed that, depending on the order, we were able to increase the derivation gain until certain level.

In hardware tests, a 6-DOF stewart platform is used for stabilization tests. The real target data collected from the field is used for target tracking tests with stewart platform. These tests are for the first time in literature are used for the comparison of fractional and integer order controllers.

We illustrated that fractional order $PI^{\lambda}D^{\mu}$ controllers outperforms the integer order controllers both in simulation tests and the hardware tests. Fractional order controller have had better response on lower speed demands, which very challenging problem in control literature with having very low inertia. The response showed that fractional order controller gives faster and stable response with less error. In target tracking tests, the another challenging problem occurs when target maneuvers fast. Similar to previous results, the fractional order controller outperformed integer order controller and give better response especially at these maneuvering points.

To conclude, we observed that the fractional order controller have better response in all tests. In tests involving more challenges, the superiority of the fractional order controller is obtained and illustrated. Without using any advanced control techniques, we have achieved to increase classical PID performance even further. The stabilization accuracy and the target tracking performance in enhanced without increasing the complexity of the controller too much. The betterness of the proposed controller is proved with hardware and simulation experimental tests.

8.2 Future Work

The fractional order controller is implemented for specifically for this system, and it offered a solution to improve the performance of these kind of systems. The implementation of such controllers would be done in other kind of systems. Especially, the system we used for these comparisons is very linear-like system, with very few nonlinear components. Performances of the systems with high nonlinearities, with more backlash, more friction, or with gun-turret platforms, can be investigated with fractional order controllers.

In addition to controller structure used in this thesis, a disturbance observer can be added to the controller to improve stabilization and tracking performances. Similarly, depending on the plant characteristics, notch filters, low-pass filters would be added to the torque loop. The cut-off frequencies of these filters should be around the first resonance frequency. Adding these filters would enable us to increase gains further, and obtain greater performances.

Complex control structures can also be tried such as adaptive controllers, fractional order sliding mode controllers. Depending on the application, and nonlinearities that need to be handled the control structure may change. Kalman filters, disturbance observers may be added to model to overcome sensor noises and other non-linearities in the system. In all of these advancements, we have to consider the complexity and applicability. As we are increasing the performance of the real system with new techniques, one of the main concern is to preserve the complexity at certain level, and we always want sensitive, sound controllers to work in different environments, different external disturbances.

REFERENCES

[1] P. J. Kennedy and R. L. Kennedy, "Direct versus indirect line of sight (los) stabilization," *IEEE Transactions on control systems technology*, vol. 11, no. 1, pp. 3–15, 2003.

[2] M. K. Masten, "Inertially stabilized platforms for optical imaging systems," *IEEE Control Systems*, vol. 28, no. 1, pp. 47–64, 2008.

[3] A. Loverro, "Fractional calculus: history, definitions and applications for the engineer," *Rapport technique, Univeristy of Notre Dame: Department of Aerospace and Mechanical Engineering*, pp. 1–28, 2004.

[4] I. Podlubny, "Geometric and physical interpretation of fractional integration and fractional differentiation," *arXiv preprint math/0110241*, 2001.

[5] S. Manabe, "The non-integer integral and its application to control systems," *Journal of Institute of Electrical Engineers of Japan*, vol. 80, no. 860, pp. 589–597, 1960.

[6] I. Podlubny, *Fractional differential equations: an introduction to fractional derivatives, fractional differential equations, to methods of their solution and some of their applications*. Elsevier, 1998, vol. 198.

[7] Y. Chen, I. Petras, and D. Xue, "Fractional order control-a tutorial," in *American Control Conference, 2009. ACC'09.* IEEE, 2009, pp. 1397–1411.

[8] Y. Luo and Y. Chen, *Fractional order motion controls*. John Wiley & Sons, 2012.

[9] J. Hilkert, "Inertially stabilized platform technology concepts and principles," *IEEE Control Systems*, vol. 28, no. 1, pp. 26–46, 2008.

[10] C. A. Monje, Y. Chen, B. M. Vinagre, D. Xue, and V. Feliu-Batlle, *Fractional-order systems and controls: fundamentals and applications*. Springer Science & Business Media, 2010.

[11] I. Petráš, *Fractional-order nonlinear systems: modeling, analysis and simulation*. Springer Science & Business Media, 2011.

[12] R. Caponetto, *Fractional order systems: modeling and control applications*. World Scientific, 2010, vol. 72.

[13] B. Ross, "The development of fractional calculus 1695–1900," *Historia Mathematica*, vol. 4, no. 1, pp. 75–89, 1977.

[14] K. Diethelm, *The analysis of fractional differential equations: An application-oriented exposition using differential operators of Caputo type.* Springer, 2010.

[15] K. Oldham and J. Spanier, *The fractional calculus theory and applications of differentiation and integration to arbitrary order.* Elsevier, 1974, vol. 111.

[16] E. C. De Oliveira and J. A. Tenreiro Machado, "A review of definitions for fractional derivatives and integral," *Mathematical Problems in Engineering*, vol. 2014, no. 1940, 2014.

[17] M. Nakagawa and K. Sorimachi, "Basic characteristics of a fractance device," *IEICE Transactions on Fundamentals of Electronics, Communications and Computer Sciences*, vol. 75, no. 12, pp. 1814–1819, 1992.

[18] K. B. Oldham and C. G. Zoski, "Analogue instrumentation for processing polarographic data," *Journal of electroanalytical chemistry and interfacial electrochemistry*, vol. 157, no. 1, pp. 27–51, 1983.

[19] L. Debnath, "Recent applications of fractional calculus to science and engineering," *International Journal of Mathematics and Mathematical Sciences*, vol. 2003, no. 54, pp. 3413–3442, 2003.

[20] A. Oustaloup, X. Moreau, and M. Nouillant, "The crone suspension," *Control Engineering Practice*, vol. 4, no. 8, pp. 1101–1108, 1996.

[21] J. T. Machado, "Fractional order modelling of dynamic backlash," *Mechatronics*, vol. 23, no. 7, pp. 741–745, 2013.

[22] B. Vinagre, I. Podlubny, A. Hernandez, and V. Feliu, "Some approximations of fractional order operators used in control theory and applications," *Fractional calculus and applied analysis*, vol. 3, no. 3, pp. 231–248, 2000.

[23] A. Oustaloup, P. Melchior, P. Lanusse, O. Cois, and F. Dancla, "The crone toolbox for matlab," in *Computer-Aided Control System Design, 2000. CACSD 2000. IEEE International Symposium on.* IEEE, 2000, pp. 190–195.

[24] D. Valerio, "Ninteger v. 2.3 fractional control toolbox for matlab," *Lisbon Technical University*, 2005.

[25] A. Tepljakov, E. Petlenkov, and J. Belikov, "Fomcon: Fractional-order modeling and control toolbox for matlab," in *Mixed Design of Integrated Circuits and Systems (MIXDES), 2011 Proceedings of the 18th International Conference.* IEEE, 2011, pp. 684–689.

[26] I. Petras, S. Grega, and L. Dorcak, "Digital fractional order controllers realized by pic microprocessor: Experimental results," *arXiv preprint math/0306016*, 2003.

[27] W. Zheng, Y. Luo, X. Wang, Y. Pi, and Y. Chen, "Fractional order PI D controller design for satisfying time and frequency domain speci fi cations simultaneously," *ISA Transactions*, vol. 68, pp. 212–222, 2017. [Online]. Available: http://dx.doi.org/10.1016/j.isatra.2017.02.016

[28] A. Tepljakov, E. Petlenkov, and J. Belikov, "Embedded system implementation of digital fractional filter approximations for control applications," in *Mixed Design of Integrated Circuits & Systems (MIXDES), 2014 Proceedings of the 21st International Conference*. IEEE, 2014, pp. 441–445.

[29] C. I. Muresan, S. Folea, G. Mois, and E. H. Dulf, "Development and implementation of an fpga based fractional order controller for a dc motor," *Mechatronics*, vol. 23, no. 7, pp. 798–804, 2013.

[30] D. Xue, C. Zhao, and Y. Chen, "Fractional order pid control of a dc-motor with elastic shaft: a case study," in *American Control Conference, 2006*. IEEE, 2006, pp. 6–pp.

[31] C. Ma and Y. Hori, "Backlash vibration suppression control of torsional system by novel fractional order pidk controller," *IEEJ Transactions on Industry Applications*, vol. 124, no. 3, pp. 312–317, 2004.

[32] Y. Luo, Y. Chen, H.-S. Ahn, and Y. Pi, "Fractional order robust control for cogging effect compensation in pmsm position servo systems: stability analysis and experiments," *Control Engineering Practice*, vol. 18, no. 9, pp. 1022–1036, 2010.

[33] I. J. Rudas, J. K. Tar, and B. Pátkai, "Compensation of dynamic friction by a fractional order robust controller," in *Computational Cybernetics, 2006. ICCC 2006. IEEE International Conference on*. IEEE, 2006, pp. 1–6.

[34] I. Podlubny, "Fractional-order systems and pi/sup/spl lambda//d/sup/spl mu//-controllers," *IEEE Transactions on automatic control*, vol. 44, no. 1, pp. 208–214, 1999.

[35] Y. Chen, "Ubiquitous fractional order controls?" *IFAC Proceedings Volumes*, vol. 39, no. 11, pp. 481–492, 2006.

[36] M. Ö. Efe, "Fractional fuzzy adaptive sliding-mode control of a 2-dof direct-drive robot arm," *IEEE Transactions on Systems, Man, and Cybernetics, Part B (Cybernetics)*, vol. 38, no. 6, pp. 1561–1570, 2008.

[37] S. Yamamoto, "Present status and future needs: the view from japanese industry," in *Proc. of 4th International Conference on Chemical Process Control-CPC IV, CACHE-AIChE*, 1991, pp. 1–28.

[38] C. A. Monje, B. M. Vinagre, V. Feliu, and Y. Chen, "Tuning and auto-tuning of fractional order controllers for industry applications," *Control engineering practice*, vol. 16, no. 7, pp. 798–812, 2008.

[39] I. Petras, "The fractional-order controllers: Methods for their synthesis and application," *arXiv preprint math/0004064*, 2000.

[40] A. Tepljakov, E. Petlenkov, and J. Belikov, "Fopid controller tuning for fractional fopdt plants subject to design specifications in the frequency domain," in *Control Conference (ECC), 2015 European*. IEEE, 2015, pp. 3502–3507.

[41] D. Valério and J. S. Da Costa, "Tuning-rules for fractional pid controllers," *IFAC Proceedings Volumes*, vol. 39, no. 11, pp. 28–33, 2006.

[42] J. Sabatier, O. P. Agrawal, and J. T. Machado, *Advances in fractional calculus*. Springer, 2007, vol. 4, no. 9.

[43] A. Tepljakov, E. Petlenkov, J. Belikov, and S. Astapov, "Digital fractional-order control of a position servo," in *Mixed Design of Integrated Circuits and Systems (MIXDES), 2013 Proceedings of the 20th International Conference*. IEEE, 2013, pp. 462–467.

[44] C. Ma and Y. Hori, "Fractional-order control: Theory and applications in motion control [past and present]," *IEEE Industrial Electronics Magazine*, vol. 1, no. 4, pp. 6–16, 2007.

[45] Z. Lv, "Time-domain simulation and design of siso feedback control systems," *National Cheng kung University, Taiwan*, 2004.

[46] A. Oustaloup, *La dérivation non entière*. Hermes, 1995.

[47] I. J. Rudas, J. K. Tar, and B. Pátkai, "Compensation of dynamic friction by a fractional order robust controller," in *Computational Cybernetics, 2006. ICCC 2006. IEEE International Conference on*. IEEE, 2006, pp. 1–6.

[48] H.-F. Raynaud and A. Zergaınoh, "State-space representation for fractional order controllers," *Automatica*, vol. 36, no. 7, pp. 1017–1021, 2000.

[49] N. Monot, X. Moreau, A. Benine-Neto, A. Rizzo, and F. Aioun, "Dynamic stability control system: the crone approach," *IFAC-PapersOnLine*, vol. 50, no. 1, pp. 13 822–13 827, 2017.

[50] D. Xue and Y. Chen, "A comparative introduction of four fractional order controllers," in *Intelligent Control and Automation, 2002. Proceedings of the 4th World Congress on*, vol. 4. IEEE, 2002, pp. 3228–3235.

[51] D. Matignon, "Stability properties for generalized fractional differential systems," in *ESAIM: proceedings*, vol. 5, 1998, pp. 145–158.

[52] M. Caputo, "Linear models of dissipation whose q is almost frequency independent—ii," *Geophysical Journal International*, vol. 13, no. 5, pp. 529–539, 1967.

[53] I. Podlubny, "Numerical solution of ordinary fractional differential equations by the fractional difference method," in *Proc. of the 2nd International Conf. in Diffence Equations (Gordon and Berach, London, 1997)*, 1997.

[54] Y. Q. Chen and K. L. Moore, "Discretization schemes for fractional-order differentiators and integrators," *IEEE Transactions on Circuits and Systems I: Fundamental Theory and Applications*, vol. 49, no. 3, pp. 363–367, 2002.

[55] J. Machado, "Analysis and design of fractional-order digital control systems," *SAMS*, vol. 27, pp. 107–122, 1997.

[56] B. Vinagre, I. Petras, P. Merchan, and L. Dorcak, "Two digital realizations of fractional controllers: Application to temperature control of a solid," in *Control Conference (ECC), 2001 European.* IEEE, 2001, pp. 1764–1767.

[57] A. Oustaloup, J. Sabatier, and P. Lanusse, "From fractal robustness to the crone control," *FCAA*, vol. 2, no. 1, pp. 1–30, 1999.

[58] Y. Chen, D. P. Atherton *et al.*, *Linear feedback control: analysis and design with MATLAB.* Siam, 2007, vol. 14.

[59] D. Mutlu, "Platform motion disturbances decoupling by means of inertial sensors for a motion stabilized gimbal," *Middle East Technical University*, 2015.

[60] G. Özdoğan, "System identification and modeling of gyro-stabilized ir/eo gimbal system in frequency domain," *Middle East Technical University*, 2014.

[61] Y. Han, Y. Lu, and H. Qiu, "An improved control scheme of gyro stabilization electro-optical platform," in *Control and Automation, 2007. ICCA 2007. IEEE International Conference on.* IEEE, 2007, pp. 346–351.

[62] T. Mitsutomi, "Characteristics and stabilization of an inertial platform," *IRE Transactions on Aeronautical and Navigational Electronics*, no. 2, pp. 95–105, 1958.

[63] A. Rue, "Stabilization of precision electrooptical pointing and tracking systems," *IEEE Transactions on Aerospace and Electronic Systems*, no. 5, pp. 805–819, 1969.

APPENDIX A

FRACTIONAL CALCULUS

A.1 Fractional Calculus

In section, three mainly used fractional order derivative and integral definitions are given. These functions will be studied and their main properties will be given. Firstly, special functions of fractional calculus will be discussed and then, properties of fractional calculus will be studied.

A.1.1 Definitions

After the special functions which are used in the fractional order derivative and integral definitions are introduced, fractional order derivative and integral definitions will be given in detail regarding their features.

A.1.1.1 Gamma Function

Euler's $\Gamma(z)$ function is one of the fundamental functions of fractional calculus. It generalizes the factorial (n!) definition, and allows n to have non-integer or complex values. $\Gamma(z)$ is defined by following integral given in [3],

$$\Gamma(z) = \int_0^\infty e^{-u} u^{z-1} dz \tag{A.1}$$

for all z, $z \in \mathbb{R}$. Below, main property of Gamma function is given,

$$\Gamma(z+1) = z\Gamma(z) \tag{A.2}$$

This property may be proved easily through integration by parts method. As a result of this property, integer values of z gives the result of known factorial, z!. You can see the plot of $\Gamma(z)$ drawn below on MATLAB to visualize function.

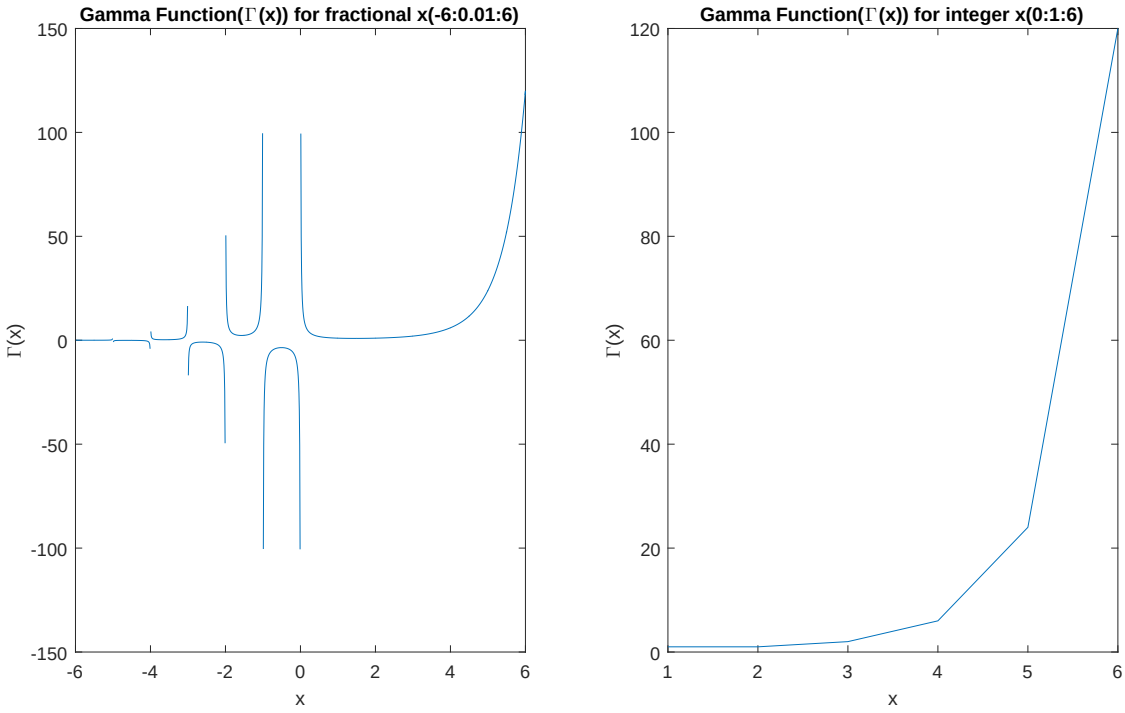

Figure A.1: Gamma Function with fractional and integer input

A.1.1.2 Beta Function

Another important function in fractional Calculys is Beta function which is also called as Euler Integral of the First Kind [**?**]. The definition of Beta function and its solution in terms of Gamma function is given below.

$$B(p,q) := \int_0^1 (1-u)^{p-1} u^{q-1} du = \frac{\Gamma(p)\Gamma(q)}{\Gamma(p+q)} \tag{A.3}$$

besides, it can easily be showed that $B(p,q) = B(q,p)$. In the figure below, Beta function is drawed for $0 \leqslant p \leqslant 4$ and $0 \leqslant q \leqslant 4$,

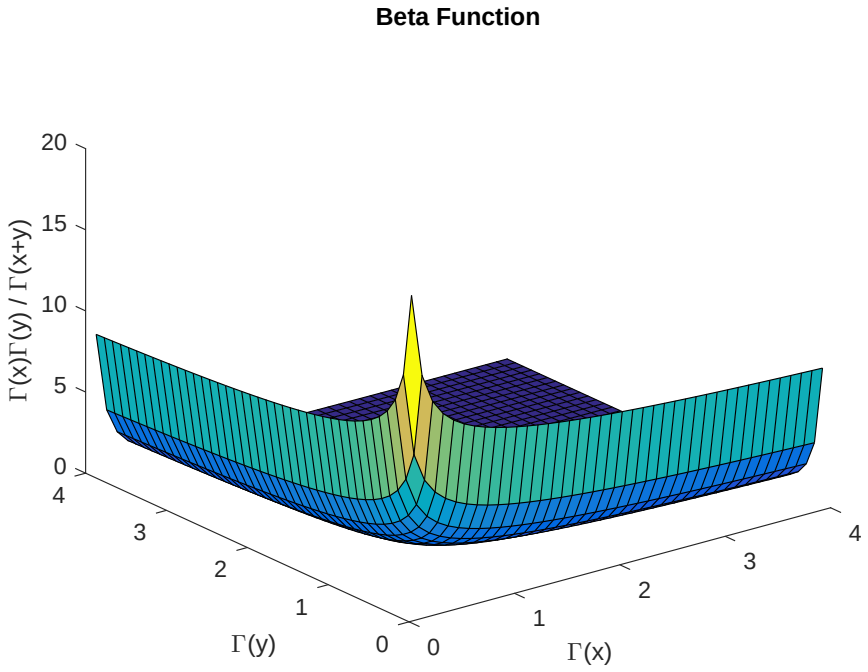

Figure A.2: Beta Function

A.1.1.3 Mittag-Leffler Function

Mittag-Leffler function is the another important function is fractional calculus. It plays an analogous role in solving fractional order differential equations. The definition of Mittag-Leffler function is given below;

$$E_\alpha(z) = \sum_{k=0}^{\infty} \frac{z^k}{\Gamma(\alpha k + \beta)} \tag{A.4}$$

where $\alpha > 0$ and $\beta > 0$.

A.1.2 Properties

After formulation of fractional order derivative and integral are given, lets move on its properties. Main properties of fractional order operators are listed below.

A.1.2.1 Identity Operator

When order of Differintegral term is taken by 0, as ${}_0\mathrm{D}_t^0$, the operator becomes an identity operator.

$$ {}_0\mathrm{D}_t^0 f(t) = f(t) \tag{A.5}$$

A.1.2.2 Commutativity

Differintegral operators satisfy commutativity law,

$$ {}_0\mathrm{D}_t^{\alpha}[{}_0\mathrm{D}_t^{\beta} f(t)] = {}_0\mathrm{D}_t^{\beta}[{}_0\mathrm{D}_t^{\alpha} f(t)] = {}_0\mathrm{D}_t^{\alpha+\beta} f(t) \tag{A.6}$$

A.1.2.3 Linearity

Differintegral operator is linear operator. For any a and b constants following equality holds,

$$ {}_0\mathrm{D}_t^{\alpha}(af(t) + bf(t)) = a\ {}_0\mathrm{D}_t^{\alpha} f(t) + b\ {}_0\mathrm{D}_t^{\alpha} g(x) \tag{A.7}$$

A.1.2.4 Integer Order Equivalence

When order of Differintegral term is taken as integer value, as ${}_0\mathrm{D}_t^r$, the operator becomes an exact integer order derivative or integral. Some examples are given below.

$$ {}_0\mathrm{D}_t^1 f(t) = \frac{\mathrm{d}}{\mathrm{d}t} f(t) \tag{A.8}$$

$$ {}_0\mathrm{D}_t^2 f(t) = \frac{\mathrm{d}^2}{\mathrm{d}t^2} f(t) \tag{A.9}$$

$$ {}_0\mathrm{D}_t^{-1} f(t) = \int_0^t f(t) \tag{A.10}$$

A.1.2.5 Laplace Transform

The general definition of Laplace transform of fractional order differentiation is given below,

$$\mathcal{L}[{}_0\mathrm{D}_t^{\alpha} f(t)] = s^{\alpha}\mathcal{L}[f(t)] \; - \; \sum_{k=1}^{n-1} s^k [{}_0\mathrm{D}_t^{\alpha-k-1} f(t)]_{t=0}. \tag{A.11}$$

However, if the derivatives of function f(t)'s are equal to 0 at t=0, this definition becomes much more simpler,

$$\mathcal{L}[{}_0\mathrm{D}_t^{\alpha} f(t)] = s^{\alpha}\mathcal{L}[f(t)] \tag{A.12}$$